

## 1. Einleitung

\*\*\* Vorläufige Fassung für Version 1.44 \*\*\*

\*\*\* Änderungen vorbehalten \*\*\*

Diese Systemspezifikation beschreibt den Entwurf und die Realisierung des Programmes für den Gebäude-Automatisierungs-Knoten (GA-Knoten) und dient der System-Betreuung.

Vorausgesetzt werden grundlegende Kenntnisse zur Gebäudeleittechnik (**GLT**) und Gebäude-Automatisation (**GA**) gemäß VDI 3814, zum FND gemäß "Planung und Ausführung von firmenneutralen Datenübertragungssystemen in öffentlichen Gebäuden und Liegenschaften" Teil 1 (FND-Spezifikation).

Begriffe und Abkürzungen werden, soweit nicht nachfolgend definiert, in Anlehnung an die zuvor genannten Dokumente verwendet und als bekannt vorausgesetzt.

Nachfolgend wird der "Standard"-GA-Knoten beschrieben. Auf Abweichungen spezifischer Varianten wird hingewiesen.

Die Standard-Schnittstellen-Adapter (SSA) für Insel- und Leitzentralen (IZ- bzw. LZ-SSA) sind aus dem GA-Knoten abgeleitet, so dass sie in dieser Systemspezifikation enthalten sind. Auf spezifische Eigenschaften wird hingewiesen.

## 2. Änderungen und Erweiterungen

Die Erfahrungen beim Einsatz des GA-Knotens in der Version 1.3 und die erweiterten Anforderungen für den Einsatz in zukünftigen Projekten führten zur Überarbeitung des GA-Knotens. Nachfolgend werden die durchgeführten Änderungen und Erweiterungen auf dem Weg zur Version 1.4 als Überblick kurz beschrieben, wobei die Systemspezifikation zum GA-Knoten in der Version 1.3 als bekannt vorausgesetzt wird.

**Hinweis: Nachfolgende Änderungen und Erweiterungen führen zu einem erhöhten Bedarf an Hauptspeicher. Unter DOS reduziert sich dadurch die Anzahl ladbaren Datenpunkte.**

Aus diesem Grund wird empfohlen, ein Update des GA-Knotens unter DOS mit einem Wechsel auf das Betriebssystem LINUX zu verbinden.

Abhängig von der LINUX-Version, der Anzahl und Größe der gestarteten Prozesse und Parametrierung des GA-Knotens kann eine Erweiterung des Hauptspeichers erforderlich sein. Stellen Sie sicher, dass keine rechenbereiten Prozesse auf Festplatte ausgelagert werden (z.B. Überwachung des LOAD-Wertes und des Prozesses KSWAPD mittels TOP). Desweiteren muss ausreichend Hauptspeicher zum Zwischenspeichern von Dateien bereitgestellt werden. Es wird empfohlen dass die Angaben "free" (Mem), "buff" und "cached" zusammen mindestens 50% des verfügbaren Hauptspeichers ergeben.

Die Festplatte, auf der das Arbeitsverzeichnis des GA-Knotens eingerichtet ist, muss über einen ausreichenden freien Speicherplatz verfügen und sollte einen Füllungsgrad von max. 60% haben.

### 2.1. Anschluss der DEMA

Die Dezentrale Erfassungs- und Melde-Anlage (DEMA) wird mittels des LON und ggf. über Modem an den GA-Knoten angeschlossen. Hierzu wurde der IZ-Treiber für LON und nachfolgende Programm-Module dahingehend erweitert, dass Rufnummer und Neuron-Id. zugeordnet und zwischengespeicherte Werte (Iz\_subid=7) mit dem übertragenen Zeitstempel weiterverarbeitet werden.

Unter LINUX und generell ab Version 1.3a muss, abweichend von der Spezifikation "Anschluß des LON an den GA-Knoten", als Link-Protokoll statt "ALERT/ACK" nun "Buffered" am SLTA eingestellt werden.

## **2.2. Sortierung der ESn-Dateien für GEN\_CFG und Vorbelegungen**

Das Programm GEN\_CFG zur Generierung der Konfigurationsdateien wurde so erweitert, dass nun die Vorlagedateien für die ES-Zeilen, die spezifisch für den Datenpunkt-Typ gemäß einem bis zu 16-stelligen Schlüssel parametrisiert werden können (\*.ESn), nun nach einem festgelegten Verfahren sortiert werden. Dadurch wird die Bearbeitungsreihenfolge eindeutig festgelegt, so dass trotz Überschneidungen nun eine bestimmte Vorlagedatei verwendet wird.

**Hinweis: In Verbindung mit dieser Maßnahme wurden die ESn-Dateien für die Landeshauptstadt München für einen 11-stelligen Schlüssel überarbeitet. Stellen Sie vor Verwendung des Programmes GEN\_CFG den aktuellen Stand der ESn-Dateien ihres Auftraggebers sicher.**

In der Datei GEN\_CFG.LOG wird protokolliert, welche Vorlagedatei für welchen Datenpunkt verwendet wurde.

Zusätzlich werden die Flags für das Meldeverhalten und die Protokollierung abhängig von der ATB\_ID automatisch besetzt, wenn diese in der Vorlagedatei nicht angegeben sind (Leerzeichen).

## **2.3. Konvertierung der ESn-Dateien für GEN\_CFG unter DOS**

Mit Erweiterung des Schlüssels auf bis zu 16 Stellen werden nun auch die Namen der ESn-Dateien bis zu 16-stellig. Da unter DOS jedoch max. 8-stellige Dateinamen unterstützt werden, wird der Dateiname ggf. in einen 8-stelligen Anteil für das Unterverzeichnis und der verbleibende Teil für den Dateinamen aufgeteilt - z.B. wird die Datei ABCDEFGHIJK.es1 unter DOS in ABCDEFGHIJK.es1 gespeichert. Um eine einmalige Pflege der ESn-Dateien unter LINUX zu ermöglichen können die dort gespeicherten ESn-Dateien mittels des Programmes ESN2DOS in das DOS-Format konvertiert werden.

## **2.4. Vorrangige Belegung von Anlagenbildern**

Zusätzlich zu dem in der ES-Zeile konfigurierten Abfrageintervall werden alle die Mess-, Stell- und Zählpunkte vorrangig in der IZ abgefragt, für die eine zyklische Istwertabfrage (IWA) parametrisiert ist. Dadurch werden die in Anlagenbildern dargestellten Werte in kürzeren Intervallen als bisher aktualisiert. Das Verfahren passt sich automatisch an die herstellerspezifischen Antwortzeiten der IZ an und stellt auch die Behandlung der anderen Programm-Module sicher.

Zusätzlich enthalten die Datagramme zur Belegung der Anlagenbilder nun als Zeitstempel den Zeitpunkt, an dem der übermittelte Zustand bzw. Wert letztmalig aus der IZ gelesen wurde.

Konnten Datenpunkte nicht gelesen werden, so wurde bisher von einem unveränderten Zustand für die Belegung der Anlagenbilder ausgegangen. Zukünftig wird in diesem Fall der Zustand bzw. Wert UNDEFINIERT gemäß FND-Spezifikation mit dem zugehörigen Zeitstempel analog geliefert.

## **2.5. Zyklische Überwachung von Melde- und Schaltpunkten**

Melde- und Schaltpunkte wurden bisher nur aufgrund ihrer Spontanmeldungen gemäß FND-Spezifikation (USM) aktualisiert. Bleiben diese aufgrund von Ausblendungen, Flimmern, Abschaltung bzw. Neustart einer Unterstation oder Störung der Kommunikation innerhalb bzw. mit der MSR-Insel aus, so konnten Wertänderungen bisher nicht angezeigt werden. Analog zu den Mess-, Stell- und Zählpunkten werden zukünftig auch die Melde- und Schaltpunkte zyklisch abgefragt und auf zu meldende Wertänderungen überwacht. Das Verfahren passt sich automatisch an die herstellerspezifischen Antwortzeiten der IZ an und stellt auch die Behandlung der anderen Programm-Module sicher.

## **2.6. Vorrangige Übertragung zwischengespeicherter Werte**

Bisher wurden Meldungen und Werte objekt-spezifisch zwischengespeichert und nur dann übertragen, wenn der Verbindungs-Aufbau durch den GA-Knoten für das jeweilige Objekt erfolgte. Zukünftig erfolgt die Übertragung auch dann, wenn eine berechtigte Leitzentrale beim GA-Knoten anruft - z.B. beim Lesen eines einzelnen Datenpunktes. Dadurch wird die Datenbank der Leitzentrale bevorzugt aktualisiert.

## **2.7. Erweiterung auf max. 128 IZ-Treiber (GAK XXL.EXE)**

Um das Erzeugen hersteller- bzw. projektspezifischer Varianten zu vermeiden wurde der GA-Knoten unter LINUX so erweitert, dass max. 128 IZ-Treiber eingebunden werden können.

Neben dem bisherigen "Standard"-GA-Knoten (GAK.EXE mit IZ01-IZ03) wird zusätzlich eine Variante bereitgestellt, die alle bisher frei verfügbaren IZ-Treiber in der benötigten Anzahl enthält (GAK\_XXL.EXE).

**Hinweis:** Beim Einsatz der GAK\_XXL.EXE in bestehenden Installationen kann eine Neuordnung der Datenpunkte zum jeweiligen IZ-Treiber erforderlich sein (vgl. Treiber-Id. in den IZ- und IS-Zeilen der Konfigurationsdatei GAK.CFG) und davon abhängiger Datenpunkte wie "%IZmmnnSTAT-oooo".

## **2.8. Unterstützung der HRP- und MRP-Regler der Fa. Kieback&Peter**

Die HRP- und MRP-Regler der Fa. Kieback&Peter verwenden eigene Varianten des P90-Protokolls. Der IZ-Treiber für P90 wurde so erweitert, dass diese Variante wahlweise unterstützt werden.

## **2.9. Mehrstufige Melde-/Schaltpunkte für MILES-Arbeitsstationen der Fa. Messner**

Technisch bedingt unterstützen MILES-Systeme für digitale Ein-/Ausgänge oder Pseudo-Datenpunkte nur 1-stufige Zustände mit je einer zugeordneten Maschinen-Adresse. Sollen mehrstufige Zustände abgebildet werden, so wird jeder Stufe in aufsteigender Reihenfolge eine eigene Maschinen-Adresse zugeordnet und im IZ-Treiber anhand des Attributs zugeordnet.

## **2.10. Unterstützung der Systeme IQ3 und ANC der Fa. Trend**

Zusätzlich zu den Systemen der Produktfamilie CNC wird ab der Version 1.3e auch die neue Produktfamilie IQ3 unterstützt. Dabei werden auch von der IQ3 invertierte Melde- und Schaltpunkte unterstützt.

Gleichzeitig wurde der IZ-Treiber dahingehend erweitert, dass externe Trend-Netzwerke mittels eines ANC aufgeschaltet werden können. Die Verwaltung der Rufnummern erfolgt durch den Treiber und ermöglicht somit den Anschluss von bis zu 255 externen Trend-Netzwerken.

## **2.11. Bedingte Umlenkungen**

Ab der Version 1.3f werden bedingte Umlenkungen unterstützt mit folgenden Bestandteilen:

- Invertierung (Schaltpunkte) bzw.
- Korrektur-Faktor (Stellpunkte)
- Wartezeit
- IF-Datenpunkt und -Wert
- ELSE-Datenpunkt

## **2.12. Aktualisierung von ES-Zeilen in der Konfigurationsdatei**

Ab der Version 1.3g bietet das Programm GEN\_CFG die Möglichkeit, die ES-Zeilen einer bestehenden Konfigurationsdatei mit den neuesten Voreinstellungen zu aktualisieren. Wird eine Objekt-Kennung angegeben, so wird diese verwendet, sonst wird die Objekt-Kennung aus der vorhandenen ES-Zeile übernommen.

## **2.13. Plausibilitätsprüfung beim IZ-Zeitabgleich**

Ab der Version 1.3h erfolgt der Zeitabgleich zu den angeschlossenen IZ's (vgl. ZA-Zeile in der Konfigurationsdatei) nur dann, wenn die Systemzeit des GA-Knotens neuer ist, als Datum/Zeit der aufgerufenen EXE-Datei (z.B. gak.exe, iz\_test.exe). Dadurch bleibt die IZ-interne Zeit z.B. nach einem Neustart des GA-Knoten mit einer defekten BIOS-Uhr/leere Batterie unverändert erhalten.

## **2.14. Zyklische Überwachung und Übernahme der IM-Dateien**

Ab der Version 1.3i werden die IM-Dateien von ES-Modul zyklisch überwacht. Änderungen in der Parametrierung als auch Störungen beim Zugriff auf die IM-Dateien und die darin enthaltenen Arbeitssätze werden auf gesonderte Meldepunkte abgebildet.

Desweiteren wird geprüft, ob im Unterverzeichnis "es" eine oder mehrere Dateien mit dem Namen "IM-  
jjjjmmthhmmss.INIT.tgz" oder "IM-  
jjjjmmthhmmss.CHANGE.tgz" ("jjjj..." Datum/Zeit) vorliegen. Falls JA, so werden die darin enthaltenen IM-Dateien wie folgt übernommen:

- Sichern der aktuellen IM-Dateien als TGZ-Datei im ARCHIV
- Für "mmmm"=INIT Löschen der aktuellen IM-Dateien

- Entpacken der IM-Dateien aus der TGZ-Datei ins ES-Verzeichnis
- Verschieben der TGZ-Datei ins ARCHIV

### **2.15. Zutrittskontrolle mit PIN-Eingabe**

Ab der Version 1.3i unterstützt das ES-Modul die Zutrittskontrolle mit Eingabe einer Persönlichen Identifikations-Nummer (PIN). Unterstützt werden Kombigeräte (IME und PIN-Eingabe in einem Gerät) als auch gesonderte Geräte für die PIN-Eingabe (P-IME), die einer IME zugeordnet werden.

### **2.16. Optimierungen Sauter EY 3600**

Ab der Version 1.3i unterstützt der Treiber für Sauter EY 3600 Optimierungen beim Polling der Datenpunkte. Dadurch werden die Reaktionszeiten beim Melden von Ereignissen wesentlich verkürzt.

### **2.17. Wartezeit beim Zeitabgleich**

Ab der Version 1.3i verwendet der Zeitabgleich über ISDN nach einem fehlerhaften Versuch die Wartezeit aus der VBP-Datei des ersten Objektes.

### **2.18. Abkündigung DOS-Version**

Ab der Version 1.3i werden GA-Knoten unter DOS nur noch auf besondere Anfrage unterstützt. Zu diesem Zweck bleiben die zugehörigen Ausführungen nachfolgend auf dem bis dahin bestehenden Stand.

## **2.19. TSYNC: Stoppen des Rechners nach ausgewiesenen Fehlern**

Ab Version 1.3i wird der Kommunikations-Prozess (KP) inkl. des Betriebssystems/Rechners nach ausgewiesenen Fehler gestoppt. Grund ist das Verhalten der TK-Anlage (TENOVIS) im Technischen Rathaus (BREF) der Landeshauptstadt München. Diese stürzt seit einem Update ab, da abgewiesene Anrufe protokolliert werden und der Puffer überläuft.

Gestoppt wird, wenn :

- der Kind-Prozess beim Eintragen von Messwerten mit Fehler beendet wird
- bei einem eingehenden ISDN-Ruf das Lesen der Tabelle GAKNOTEN zu einem anderen Fehler führt, als das kein Eintrag gefunden wurde.

Dazu wird mittels InterProcessCommunication (IPC) eine Nachricht an den Prozess TSYNC gesendet, der den privilegierten System-Aufruf zum Stoppen des Rechners (init 0) ausführt. TSYNC muss somit auf diesem Rechner mit den entsprechenden Rechten gestartet werden (analog zum GA-Knoten).

## **2.20. Fremd-Generierung mit GEN\_CFG**

Ab Version 1.3j unterstützt GEN\_CFG auch die Fremd-Generierung (vgl. Schalter "/f"). Damit können z.B. unter DOS Konfigurationsdateien für ein Fremd-System wie LINUX erzeugt werden.

## **2.21. IZ-Treiber für BACnet/IP (BACnet-Client)**

Ab Version 1.3j unterstützt ein IZ-Treiber den Anschluss von MSR-Systemen auf Basis von BACnet/IP.

## **2.22. IZ-Treiber für AS511**

Ab Version 1.3j unterstützt ein IZ-Treiber den Anschluss von MSR-Systemen auf Basis des Protokolls AS511 (z.B. SIEMENS S5).



### **2.23. LZ-Treiber mit FND/LON-Kernel**

Ab Version 1.3j unterstützt ein LZ-Treiber alternativ zum Standard-LZ-Treiber den Anschluss an andere GA-Knoten (Kaskadierung) mit FND über LON (FND/LON-Kernel, d.h. LonTalk mit FND 1.0).

### **2.24. SAIA-Bus/GfR-Treiber mit CMSPAR-Flag**

Aufgrund der spezifischen Parity-Behandlung musste unter LINUX der GA-Knoten mit Superuser-Rechten gestartet werden, wenn der IZ-Treiber für SAIA-Bus bzw. GfR verwendet wurde und war aufgrund des System-Aufruf IOPERM begrenzt auf serielle Schnittstellen mit Port-Adressen  $\leq 0x3FF$  (vgl. "man ioperm").

Ab Version 1.3k unterstützt der IZ-Treiber alternativ ein Verfahren unter Verwendung des CMSPAR-Flags. Dieses MUSS vom LINUX-Treiber "serial.o" unterstützt werden, ermöglicht dann aber höhere Port-Adressen (z.B. PCI-Steckkarten) und erfordert keine Superuser-Rechte.

### **2.25. GfR-Treiber mit SBUS+ und automatische REF-Generierung**

Die Fa. GfR verwendet alternativ zum SBUS-Protokoll die spezifische Variante SBUS+.

Ab Version 1.3k unterstützt der IZ-Treiber den erweiterten Adressbereich für Register sowie die Formate Integer, Float, Date (TTMM) und Time (SSMM).

### **2.26. P90-Treiber mit Datums/Zeit-Formaten**

Ab Version 1.3k unterstützt der IZ-Treiber für das P90-Protokoll der Fa. Kieback & Peter zusätzlich die Datums-/Zeit-Formate für Stellpunkte die folgenden Parametertypen :

- Datum (TTMMJJ)

- Zeit (SSMM)

- Nutzzeit                      VON/BIS-Stellpunkte    (jeweils SSMM)
  
- Sondernutzzeit              DATUM/VON/BIS-Stellpunkte    (TTMMJJ bzw. jeweils SSMM)

### **2.27. Automatisches Löschen von IMS-Dateien**

Ab Version 1.3k wurde die Zutrittskontrolle um das Hilfsprogramm CLR\_IMS ergänzt. Dieses kann z.B. mittels eines CRON-Jobs zyklisch gestartet werden und dient zum Löschen der IMS-Dateien unter folgenden Bedingungen :

- 1.) die in der IMS-Datei vermerkte Raumzone "RZ\_Kenn" kann wegen abgelaufener Gültigkeit nicht mehr betreten bzw. verlassen werden
  
- 2.) die in der IMS-Datei vermerkte Raumzone "RZ\_Kenn" kann wegen nicht mehr bestehender Zutrittsberechtigung gemäß IM-Datei nicht mehr betreten bzw. verlassen werden

Der Aufruf des Programmes erfolgt im Arbeits-Verzeichnis des GA-Knotens.

Jeder Programmablauf wird in der Datei CLR\_IMS.LOG protokolliert. Dabei werden Zeitpunkt und Grund der Löschung für jede IMS-Dateien angegeben.

### **2.28. LZ-Treiber für BACnet/IP (BACnet-Server)**

Ab Version 1.3k unterstützt ein LZ-Treiber (LZBN) den Anschluss von Leitzentralen/Management-Stationen auf Basis von BACnet/IP. Damit können alle IZ-seitig unterstützten MSR-Systeme nun auch auf BACnet-Managementstationen (Leitzentralen) aufgeschaltet werden.

### **2.29. IZ-Treiber für ILC 350, FC 350 o.ä. über PCOS (Fa. PHOENIX Contact)**

Ab Version 1.4a unterstützt ein IZ-Treiber den Anschluss von MSR-Systemen ILC 350 ETH, FC 350 ETH o.ä. auf Basis des Protokolls "ProConOS online communication services" (PCOS; Fa.. PHOENIX Contact).

### **2.30. IZ-Treiber für MODBUS/IP und MODBUS/RTU**

Ab Version 1.4a unterstützt ein IZ-Treiber den Anschluss von MSR-Systemen auf Basis des Protokolls MODBUS wahlweise über TCP/IP-Netzwerk (MODBUS/IP; z.B. WAGO 750-841) oder serielle Schnittstelle (MODBUS/RTU; z.B. WAGO 750-814).

### **2.31. GA-Knoten auf Basis des VPN-Routers GenuBOX (GenuGAK, Fa. GeNUA)**

Ab Version 1.4a ist der GA-Knoten auch lauffähig auf dem VPN-Router GenuBOX der Gesellschaft für Netzwerk- und Unix-Administration mbH (GeNUA). Dadurch kann insbesondere bei kleineren und mittleren Installationen auf den Einsatz eines gesonderten PC's verzichtet werden.

### **2.32. Swapped Formate für MODBUS/IP und MODBUS/RTU**

Ab Version 1.42a unterstützt der GA-Knoten auch register-übergreifende Formate, bei denen die Reihenfolge der Register im MODBUS-Datagramm umgekehrt (swapped) zugeordnet wurde.

### **2.33. Start von GEN\_CFG im Arbeitsverzeichnis und gak\_OOOO.cfg**

Ab Version 1.42d ist GEN\_CFG auch im Arbeitsverzeichnis der GA-Knotens ablauffähig und nutzt die vorhandene Unterverzeichnis-Struktur. Zusätzlich wird eine Kopie von GEN\_CFG.CFG im Unterverzeichnis CFG mit dem Namen gak\_OOOO.cfg angelegt, wobei OOOO der beim Aufruf angegebenen Objekt-Kennung entspricht.

### **2.34. Kommando-Datei GAK.CMD**

Ab Version 1.43c bearbeitet der GA-Knoten auch Kommandos, die mittels der Datei GAK.CMD im Arbeitsverzeichnis übergeben wurden.

### 2.35. GfR-Treiber mit SBUS+ und Zugriff auf RealTimeClock (RTC)

Die Fa. GfR verwendet alternativ zum SBUS-Protokoll die spezifische Variante SBUS+.

Ab Version 1.43c unterstützt der IZ-Treiber unter SBUS+ den lesenden und modifizierenden Zugriff auf die Real Time Clock einer jeden angeschlossenen Unterstation.

## 3. DV-Architektur

Der GA-Knoten verwendet das Betriebssystem MS-DOS oder LINUX. Als Entwicklungs- und Einsatz-Umgebung wurden bisher verwendet:

- MS-DOS Version 6.22 (bis GA-Knoten Version 1.3h)
- SuSE-LINUX in den Versionen 6.0, 6.1, 6.4, 7.3, 8.1 und 10.1
- OpenBSD in der Version 3.7 (nur GenuGAK)

**Hinweis: Nachfolgend sind Verzeichnis- und Dateinamen i.a. in Großbuchstaben dargestellt, um sie aus dem Text optisch hervorzuheben. Unter LINUX sind diese in der Einsatzumgebung stets in Kleinbuchstaben anzugeben, während in der Entwicklungsumgebung auch die Schreibweise in Großbuchstaben auftreten kann.**

**Ab der Version 1.3i werden GA-Knoten unter DOS nur noch auf besondere Anfrage unterstützt. Zu diesem Zweck bleiben die zugehörigen Ausführungen nachfolgend auf dem bis dahin bestehenden Stand.**

### 3.1. Software-Einheiten

Der Rechner für den GA-Knoten verwendet das Betriebssystem MS-DOS oder LINUX. Für DOS werden alle Programme (inkl. Systemtreiber) so installiert, dass auch der Speicherbereich oberhalb von 640 KB genutzt wird. Für LINUX entfällt die Speicherbegrenzung, so dass eine praktisch unbegrenzte Anzahl von Datenpunkten unterstützt wird.

Nach einem Neustart des Systems werden die Programme und Systemtreiber automatisch durch Aufruf aus den Dateien CONFIG.SYS und AUTOEXEC.BAT (für DOS) bzw. /sbin/init.d/boot.local oder /etc/init.d/rc (für LINUX) des Boot-Mediums gestartet. Die angeschlossenen IZ's und peripheren Geräte (z.B. Protokolldrucker) werden initialisiert.

Normalerweise läuft auf dem Rechner das Programm für den GA-Knoten **permanent** ab, d.h. es wird einmalig angestartet und nur dann beendet, wenn ein Fehler im System auftritt oder ein externes Ereignis ein Weiterarbeiten verhindert (z.B. Stromausfall, Reset, ...). Bei Bedarf läßt sich das Programm durch Eingabe eines STOP-Kommandos an der Tastatur durch den Bediener gezielt beenden.

Um eine Fernwartung des GA-Knotens durch eine LZ zu ermöglichen, wurde der Schalterpunkt "%GAKnnn\_START" eingeführt. Dabei gilt: 001 <= nnn <= 127, falls die Kennung des GA-Knotens in der KK-Zeile der Konfigurationsdatei definiert wurde, sonst entfällt nnn. Nach Empfang des Schaltbefehls mit dem Vorgabewert STOP wird der Programmablauf beendet und in den aufrufenden Batch rückgekehrt. Hier kann nun ein Hilfsprogramm für die Fernwartungs-Dienste (z.B. FWD) gestartet werden. Zusätzlich meldet dieser Datenpunkt jeden Programmstart/-stop spontan zur LZH.

Der ISDN/X.25-Router kann durch die LZ gestoppt werden, indem über den ISDN/X.25-Router ein ISDN-Verbindungsaufbau auf sich selbst versucht wird. Die Behandlung der Fernwartungs-Dienste erfolgt analog zum GA-Knoten.

Nach Beendigung der Fernwartung kann durch Aufruf des Programmes REBOOT (für DOS) bzw. INIT 6 (für LINUX) ein Neustart (wahlweise Warm- oder Kaltstart) durchgeführt werden, so dass auch Änderungen an der Systemkonfiguration (z.B. CONFIG.SYS, AUTOEXEC.BAT, ...) berücksichtigt werden.

Ruft ein für die Fernwartung berechtigter Teilnehmer bei einem GA-Knoten, SSA oder ISDN/X.25-Router mit aktiviertem IZ03-Treiber an, so beendet der GA-Knoten den Programmablauf und kehrt in den aufrufenden Batch zurück.

Durch Austausch des Boot-Mediums (z.B. Einlegen einer Floppy-Disk) kann erreicht werden, dass andere Programme gestartet werden (z.B. Test- oder Parametrierprogramme, ...).

Das Programm für den GA-Knoten verwendet zur Bildschirmsteuerung die **ANSI-ESCAPE-Zeichenfolgen**, so dass es auch ablauffähig ist, wenn statt einer im Rechner integrierten Bildschirmparte (HERCULES, EGA, VGA,...) ein ANSI-kompatibler Bildschirm (Terminal) über eine serielle Schnittstelle angeschlossen wird.

**Hinweis: Zur Unterstützung der ANSI-ESCAPE-Sequenzen auf der Bildschirmparte muß auf dem Rechner der DOS-Treiber ANSI.SYS und die Codeseite 850 installiert sein. Unter LINUX wird der Consolefont "cp850-8x16" empfohlen.**

### **3.2. Hardware-Einheiten**

Als Rechner für den GA-Knoten wird ein marktüblicher PC (Desktop, Tower, Laptop, Notebook,...) verwendet. Zum Anschluß der IZ's dienen die seriellen Schnittstellen (COM1 - COM4) oder spezifische Schnittstellenkarten. Der Anschluß an die LZH und den BWD wird über die in den Rechner eingebaute ISDN- oder Netzwerk-Karte realisiert.

Für den Rechner unter DOS werden minimal die folgenden Komponenten verwendet :

- 486 DX (IBM AT kompatibel, inkl. Co-Prozessor)
- 8 MByte Hauptspeicher
- >200 MByte Festplatte (Laufwerk C:)
- 3.5" Floppy-Disk mit 1.44 MByte (Laufwerk A:)
- MF-102-Tastatur
- VGA-Karte mit Monitor
- 2 serielle Schnittstellen (COM1, COM2)
- 1 parallele Schnittstelle (LPT1)
- (inter)-aktive ISDN-Karte (d.h. mindestens Behandlung der unteren Protokoll-Schichten auf der ISDN-Karte)

Für den Rechner unter LINUX werden minimal die folgenden Komponenten verwendet, soweit nicht seitens der verwendeten LINUX-Distribution weitergehende Anforderungen bestehen :

- Pentium mit 800MHz (oder kompatibel)
- >= 64 MByte Hauptspeicher (>=128 MB ab SuSE 8.1)
- >= 8 GByte Festplatte
- 3.5" Floppy-Disk mit 1.44 MByte (Laufwerk A:)
- MF-102-Tastatur
- VGA-Karte mit Monitor
- 2 serielle Schnittstellen (COM1, COM2)
- 1 parallele Schnittstelle (LPT1)
- (inter)-aktive ISDN-Karte (d.h. mindestens Behandlung der unteren Protokoll-Schichten auf der ISDN-Karte)

- alternativ Netzwerk-Karte

Für den GenuGAK unter OpenBSD werden minimal die folgenden Komponenten verwendet, soweit nicht seitens des Herstellers weitergehende Anforderungen bestehen :

- GenuBOX basierend auf Pentium mit  $\geq 400$ MHz (oder kompatibel)
- $\geq 512$ MByte flüchtiger Hauptspeicher (auch als RAM-Disk für Arbeitsverzeichnis)
- $\geq 4$ GByte wiederbeschreibbarer, nicht flüchtiger Hauptspeicher (Flash-Disk)
- 2 serielle Schnittstellen (/dev/cua01, /dev/cua02 auch als console)
- 3 Netzwerk-Schnittstellen (RJ45, mind. 10 und 100 Mbit/s)

**Hinweis: Abhängig von projektspezifischen Anforderungen müssen für den GA-Knoten ggf. andere Hardware-Komponenten konfiguriert werden. Dieses betrifft die Bauart, die benötigte Rechnerleistung (CPU), die Speicherkapazität (Hauptspeicher, Festplatte) und die Schnittstellen für die anzuschließenden IZ. Beachten Sie auch die Empfehlungen des LINUX-Distributors.**

**Hinweis: Werden bei Projekten unter DOS die seriellen Schnittstellen COM3 und COM4 verwendet, so ist sicherzustellen, dass diese die Interrupt-Leitungen IRQ 10 und IRQ 11 bedienen.**

**Hinweis: Für die kontinuierliche Eigenüberwachung des Hauptspeichers unter DOS ist ein spezifischer Befehl des Prozessortyps 486 zum Leeren des Cache-Speichers erforderlich (WBINVD). Als Folge davon ist der GA-Knoten in der gelieferten Form nur ablauffähig auf diesem oder kompatiblen Prozessoren.**

Bei Rechnern mit PCI-Bus führte dieser Befehl zum Absturz/Aufhänger, so dass er bis auf weiteres deaktiviert bleibt (vgl. Routine SY\_MEMCHK).

**Hinweis: GA-Knoten auf Basis des GenuGAK unterstützen ISDN derzeit nicht**

Als Protokoll-Drucker kann jeder handelsübliche Zeilendrucker mit paralleler Schnittstelle verwendet werden, soweit er 132 Zeichen/Zeile ausgeben kann. Vorzugsweise sollte dieser mit einer Vorrichtung zum Drucken auf Endlospapier ausgestattet sein. Der GA-Knoten sendet keine Initialisierungs-Sequenzen, so dass der Drucker nach jedem Einschalten automatisch auf die Ausgabe von 132 Zeichen/Zeile eingestellt sein muß. Die Anzahl der Zeilen/Seite können in der Konfigurationsdatei (vgl. DR-Zeile) eingestellt werden.

### 3.3. Systeminterne Schnittstellen

Die Bedienung des GA-Rechners (z.B. Installation und Tests bei der Inbetriebnahme oder später) erfolgt über den angeschlossenen Bildschirm und die Tastatur. Nachfolgend wird der GA-Knoten in der Standard-Konfiguration (**Standard-GA-Knoten**) der Landeshauptstadt München beschrieben.

**Hinweis:** Für DOS-basierte Projekte mit ANSI-Bildschirm statt Bildschirm-Karte mit Monitor und Tastatur bzw. zum Anschluß zusätzlicher Eingabemedien (z.B. Maus) bleibt die erste serielle Schnittstelle **COM1** für diese Aufgaben reserviert. Der an dieser Schnittstelle ggf. anzuschließende Bildschirm mit Tastatur muß die **ANSI-ESCAPE-Zeichenfolgen für die Bildschirmsteuerung** unterstützen. Wird an dieser Schnittstelle eine Maus betrieben, so ist der zugehörige DOS-Treiber zu laden. Nur in Ausnahmefällen, wenn keine andere serielle Schnittstelle verfügbar ist, kann COM1 zum Anschluß einer IZ verwendet werden.

Bei LINUX-basierten GA-Knoten wird ggf. die Unterbrechungsfreie Stromversorgung (USV) an die erste serielle Schnittstelle **/dev/ttyS0** angeschlossen.

Zum Versenden von Short-Message-Services (SMS) als auch zum Melden von Ruf-Ereignissen wird der GA-Knoten an das ISDN angeschlossen. Die zugehörigen Datenpunkte werden dem Treiber IZ01 zugeordnet. Unter LINUX können zusätzlich Fax und eMail versandt werden.

Die aus der SPS (AEG/Schneider Groupe Modicon) aufgebaute IZ wird über eine serielle Schnittstelle an den GA-Knoten angeschlossen und dem Treiber IZ02 zugeordnet. Die Datenübertragung erfolgt mit den folgenden Schnittstellenparametern :

- 9600 Baud
- 1 Startbit
- 8 Datenbits
- 1 Stopbit
- ODD Parity

Als Übertragungsprotokoll werden die KS-Funktionen gemäß "A120 KS-Funktionen Benutzerhandbuch " zur Produktfamilie Modicon der Firma AEG/Schneider Groupe (Dokumentennummer A91M.271974.20-0791) realisiert, soweit sie zur Realisierung des Projektes erforderlich sind.



Die IZ mit FND-Protokoll gemäß Version 1.0 (z.B. DDC diverser Hersteller) wird über die V.24-, ISDN- oder Netzwerk-Schnittstelle angeschlossen und dem Treiber IZ03 zugeordnet. Die Datenübertragung über V.24 erfolgt mit den hersteller- bzw. projektspezifisch vereinbarten Schnittstellenparametern, die vor Start des GA-Knotens mittels des MODE-Befehls (für DOS) bzw. setserial-Befehls (für LINUX) gesetzt werden müssen. Für Verbindungen über ISDN- oder Netzwerk müssen die zugehörigen Rufnummer bzw. IP-/Port-Adresen der Verbindungspartner eingestellt werden.

Der Protokolldrucker wird an die parallele Schnittstelle LPT1 angeschlossen.

### **3.4. Systemexterne Schnittstellen**

Die Übertragung der Daten vom GA-Knoten zur bzw. von der LZH und den BWD für Gefahr und Einbruch erfolgt über die in den Rechner eingebaute ISDN- oder Netzwerk-Karte. Hierzu ist der zur ISDN- bzw. Netzwerk-Karte zugehörige Treiber - bei ISDN inkl. CAPI-Schnittstelle in der Version 2.0 - erforderlich, auf den das Programm des GA-Knotens mittels des in der CAPI-Spezifikation festgelegten Verfahrens zugreift. Unter LINUX wird momentan nur die CAPI 2.0 durch die FritzCard-, B1-, C1- und T1-Karten der Fa. AVM unterstützt.

Als Verbindungsprotokoll zur LZH wird das FND-Protokoll mit Ergänzungen ("te"-Bit und Karteipunkte) verwendet.

Zu den BWD's werden die Verbindungsprotokolle TSS 13a und FND 1.0 realisiert, die über das Verbindungsprofil bzw. die Anruferberechtigung (vgl. ANRUFER.CHK) für den jeweiligen Verbindungs-Partner parametrisiert werden.

### **3.5. Übersicht**

Die Einbindung des GA-Knotens in das Gesamtsystem, das aus ISDN-Netz und/oder lokalem Netzwerk (LAN), LZH und BWD's besteht, ist der allgemeinen Beschreibung im Leistungsverzeichnis zu entnehmen.

## **4. Software-Architektur**

### **4.1. Entwicklungsumgebung**

Die Software-Entwicklung erfolgt auf einem ausreichend ausgebauten separaten DOS-Rechner, der ansonsten keine besonderen Merkmale aufweist. Die Übergabe der Programme erfolgt mittels 3.5"-Floppies mit einer Kapazität von 1.44 MB, CD, DVD oder als gepackte Datei, z.B. mittels eMail.

Für die Software-Entwicklung unter DOS werden folgende Software-Werkzeuge verwendet :

- Bildschirm-Editor
- Microsoft Visual C/C++ Version 1.5
- Microsoft MASM Version 6.1
- EICON X.25 Network-Level Developer's Toolkit (falls X.25-Schnittstellen nicht deaktiviert sind)
- CAPI 2.0

Für die Software-Entwicklung unter LINUX werden die auf der DOS-Entwicklungs-Umgebung erzeugten Programm-Quellen auf die Entwicklung-Umgebung unter LINUX kopiert. Ggf. sind in den Programm-Quellen systemspezifische Schalter (Define) für den Pre-Compiler eingefügt. Es werden folgende Software-Werkzeuge verwendet :

- Bildschirm-Editor
- GNU C/C++
- Efence
- CAPI 2.0

Für die Software-Entwicklung unter OpenBSD (nur für GenuGAK) werden die auf der DOS-Entwicklungs-Umgebung erzeugten Programm-Quellen auf die Entwicklung-Umgebung unter OpenBSD kopiert. Ggf. sind in den Programm-Quellen systemspezifische Schalter (Define) für den Pre-Compiler eingefügt. Es werden folgende Software-Werkzeuge verwendet :

- Bildschirm-Editor
- GNU C/C++

Die zu den Software-Werkzeugen gehörigen Dokumentationen enthaltenen Informationen werden in diese Systemspezifikation nicht aufgenommen und werden als bekannt vorausgesetzt.

**Hinweis: Die Entwicklungswerkzeuge (Hard- und Software) und die zugehörigen Dokumentationen sind nicht Bestandteil des Lieferumfangs und müssen ggf. vom Auftraggeber gesondert beschafft werden !**

Für die Generierung der Programm-Quellen wird das Hilfsprogramm CHANGE aus dem Unterverzeichnis QUELLEN\BAT benötigt. Soweit noch nicht geschehen, generieren Sie dieses durch Aufruf von "ch\_comp" und "c\_link @change.lnk" (DOS) bzw. "change.lnk" (LINUX).

Installieren Sie dieses dann, indem Sie die Datei "change.com" (DOS) bzw. "change.exe" (LINUX) in ein Verzeichnis kopieren, das über PATH erreichbar ist. Unter LINUX muss die Datei dabei nach "change" umbenannt werden.

#### **4.1.1. Generierung aus den Programm-Quellen**

Für die Generierung aller zum GA-Knoten gehörenden Module und Programme steht im Unterverzeichnis QUELLEN die Batchdatei GEN\_GAK.BAT (DOS) bzw. GEN\_GAK (LINUX) zur Verfügung, die wie folgt aufzurufen ist:

- Wechseln Sie mittels CD-Kommando in das Unterverzeichnis QUELLEN
- Starten Sie die Generierung durch Eingabe von GEN\_GAK bzw. für Sondersysteme mit zusätzlicher, zeichengenaue Angabe dessen Namens (z.B. "gen\_gak GenuGAK").
- In den jeweiligen Unterverzeichnissen finden sie nun die jeweiligen Programme (z.B. GAK\GAK.EXE, SSA\SSA.EXE, FWD\FWD.EXE,...). Die verfügbaren IZ-Treiber sind in den jeweiligen Unterverzeichnissen von QUELLEN\IZ in Form von den Routinen-Bibliotheken IZnnLIB.LIB (mit nn=01, 02, 03, 04, ...) abgelegt und sind fest den jeweiligen IZnn-Treiber zugeordnet.

**Hinweis:** Die Programm-Quellen werden aufgrund unterschiedlicher Nutzungsrechte in verschiedenen Konfigurationen ausgeliefert. Alternativ sind DUMMY-Routinen, nur die Routinen-Bibliothek oder die ablauffähigen Programme (EXE-Dateien) verfügbar.

GEN\_GAK erkennt die vorhandene Konfiguration und generiert die Programm-Module anhand der verfügbaren Bestandteile.

#### 4.1.2. Einbinden spezifischer IZ-Treiber unter DOS

Soll eine Version des GA-Knotens mit spezifischen IZ-Treibern gebunden werden, so können die im nachfolgenden Beispiel *kursiv* markierten Zeilen angepaßt bzw. entfernt werden. Dabei kann nur jeweils eine Routinen-Bibliothek IZnnLIB.LIB eingebunden werden.

Beispiel einer LNK-Datei (QUELLEN\GAK\GAK\_LON.LNK):

```
GAK+
INIT+EXIT+LOOP+
IIZ+ILZ+ISP+IES+OIZ+SYSCHK+ITAST+
DPP_INIT+DPP_EXIT+DPP_CMD+DPP_POLL+
FLIMMCHK+ROLLMELD+START_DP
GAK_LON.EXE/NOE/NOD/CP:1/STACK:5632/SEG:256
GAK_LON.MAP/MAP:FULL
..\SPASP_LIB.LIB+
..\ES\ES_DLIB.LIB+
..\LZ\LZ_LIB.LIB+
..\IZ\IZ_LIB.LIB+
..IZ\FND_10\IZ01LIB.LIB+
..IZ\AEGKS120\IZ02LIB.LIB+
..IZ\FND_CAP1\IZ03LIB.LIB+
..IZ\LON\IZ04LIB.LIB+
..\IZ\IZ_DTRB.LIB+
..\CAPI_20\CAPILIB.LIB+
..\X25\X25_LIB.LIB+
..\UT\UT_LIB.LIB+
..\BS\BS_LIB.LIB+
..\SY\SY_LIB.LIB+
C:\EICON\X25DOS\LIB\X25MVL.LIB+
C:\MSVC_15\LIB\OLDNAMES.LIB+
C:\MSVC_15\LIB\LLIBC7.LIB
```

Dabei stehen unter DOS folgende IZ-Treiber als Routinen-Bibliotheken (IZnnLIB.LIB) zur Verfügung:

<u>QUELLEN-Unterverzeichnis</u>	<u>Protokoll</u>	<u>Anbieter</u>
..\IZ\AEGKS120	KS 120 (V.24)	Schneider Groupe Modicon
(c) Daten - Technik Fries,	Hochstr. 25, D-85221 Dachau	16.Juni 2010

..\IZ\AND_INF	Infinity (V.24)	Andover
..\IZ\DEKATEL	DEKATEL (V.24)	Viessmann
..\IZ\EIB_221	EIB 2.21 (V.24)	diverse
..\IZ\EY_2400	EY 2400 (V.24)	Sauter
..\IZ\EY_3600	EY 3600 (V.24)	Sauter
..\IZ\FND_10	FND 1.0 (V.24)	diverse
..\IZ\FND_CAPI	FND 1.0 (V.24 / ISDN)	diverse
..\IZ\GFR	SBus+ (V.24)	GfR
..\IZ\KP_P90	P90 (V.24)	Kieback & Peter
..\IZ\KP_SUCOM	SUCOM (V.24)	(Klöckner-)Moeller
..\IZ\LON	LonTalk mit FND 1.0 (V.24)	diverse
..\IZ\MK_DC97	Supramat DC97 (V.24)	Fröling
..\IZ\MILES_AS	MILES-Arbeitssation (V.24)	Messner
..\IZ\MULT_A3K	Access 3000 (V.24)	Multitone
..\IZ\RAMCC600	CC 600 (V.24)	RAM
..\IZ\SAIA_BUS	SAIA-Bus (V.24)	SAIA
..\IZ\SE_IPC	IPC (V.24)	SE-Electronik GmbH (Straub)
..\IZ\SMS	D1-, D2-, ePlus (ISDN)	diverse
..\IZ\TRENDNCNC	CNC+ (V.24)	Caradon TREND

Für die Generierung eines GA-Knotens mit spezifischen IZ-Treibern gehen sie wie folgt vor:

- Wechseln Sie mittels CD-Kommando in das Unterverzeichnis QUELLEN\IZ
- Geben Sie "DIR IZ??LIB.LIB /s /p" ein und vermerken Sie das Unterverzeichnis und Dateinamen der gewünschten IZnnLIB.LIB
- Wechseln Sie mittels CD-Kommando in das Unterverzeichnis QUELLEN\GAK
- Kopieren Sie z.B. die Datei GAK\_LON.LNK auf GAK\_XXX.LNK (mit "XXX" als neuer Kennung)
- Ersetzen Sie in GAK\_XXX.LNK die Zeichenfolgen "\_LON" durch "\_XXX"
- Passen Sie in GAK\_XXX.LNK die Unterverzeichnisse und "nn" für die einzubindenden IZnnLIB.LIB an
- Starten Sie den Bindevorgang durch Aufruf von "C\_LINK @GAK\_XXX.LNK". Als Ergebnis wird GAK\_XXX.EXE und GAK\_XXX.MAP geliefert.

#### **4.1.3. Einbinden spezifischer IZ-Treiber unter LINUX**

Soll eine Version des GA-Knotens mit spezifischen IZ-Treibern gebunden werden, so erfolgt dieses analog zu DOS. Als Beispiel einer LNK-Datei dient QUELLEN/GAK/gak\_lon.lnk:

```
rm gak_lon.exe
```

```
cc -o gak_lon.exe gak.o init.o exit.o loop.o \
    iiz.o oiz.o ilz.o ies.o isp.o start_dp.o \
    dpp_init.o dpp_exit.o dpp_cmd.o dpp_poll.o \
```

```
rollmeld.o flimmchk.o itast.o syschk.o \  
../SP/sp_lib.a \  
../ES/es_dlib.a \  
../LZ/lz_lib.a \  
../IZ/iz_lib.a \  
../IZ/SMS/iz01lib.a \  
../IZ/AEGKS120/iz02lib.a \  
../IZ/FND_CAPI/iz03lib.a \  
../IZ/LON/iz04lib.a \  
../IZ/iz_dtrb.a \  
../IZ/iz_lib.a \  
../CAPI_20/capilib.a \  
../IP_4/ip_lib.a \  
../UT/ut_lib.a \  
../BS/bs_lib.a \  
../SY/sy_lib.a \  
../IPC/ipc_lib.a \  
/usr/lib/libcapi20.a \  
/usr/lib/libm.a
```

Dabei stehen unter LINUX folgende IZ-Treiber als Routinen-Bibliotheken (IZnnLIB.LIB) zur Verfügung:

<b>QUELLEN-Unterverzeichnis</b>	<b>Protokoll</b>	<b>Anbieter</b>
..\IZ\AEGKS120	KS 120 (V.24)	Schneider Groupe Modicon
..\IZ\AND_INF	Infinity (V.24)	Andover
..\IZ\AS511	AS511 (V.24)	diverse
..\IZ\BNC	BACnet/IP (LAN)	diverse
..\IZ\DEKATEL	DEKATEL (V.24)	Viessmann
..\IZ\EY_2400	EY 2400 (V.24)	Sauter
..\IZ\EY_3600	EY 3600 (V.24)	Sauter
..\IZ\FND_10	FND 1.0 (V.24)	diverse
..\IZ\FND_CAPI	FND 1.0 (V.24 / ISDN / LAN)	diverse
..\IZ\GFR	SBus+ (V.24)	GfR
..\IZ\KP_P90	P90 (V.24)	Kieback & Peter
..\IZ\KP_SUCOM	SUCOM (V.24)	(Klößner-)Moeller
..\IZ\LON	LonTalk mit FND 1.0 (V.24)	diverse
..\IZ\MK_DC97	Supramat DC97 (V.24)	Fröling
..\IZ\MILES_AS	MILES-Arbeitssation (V.24)	Messner
..\IZ\MULT_A3K	Access 3000 (V.24)	Multitone
..\IZ\RAMCC600	CC 600 (V.24)	RAM
..\IZ\SAIA_BUS	SAIA-Bus (V.24)	SAIA
..\IZ\SE_IPC	IPC (V.24)	SE-Electronik GmbH (Straub)
..\IZ\SMS	D1-, D2-, ePlus (ISDN)	diverse
..\IZ\SMS	Fax, eMail (ISDN/LAN)	diverse
..\IZ\TRENDNCNC	CNC+ (V.24)	Caradon TREND

**Hinweis:** Unter LINUX stehen zusätzlich IZ-Treiber für folgende Systeme zur Verfügung :

- PRV/EKL (V.24) Landis & Staefa

Diese IZ-Treiber sind bisher nur zu Testzwecken freigegeben. Nähere Informationen erhalten Sie bei:

**Daten-Technik FRIES**

**Hochstr. 25**

**85221 Dachau**

**Tel.: 08131 / 80770**

**Fax: 08131 / 80772**

**Mail: info@DT-Fries.de**

#### **4.1.4. Einbinden spezifischer LZ-Treiber unter DOS**

Soll eine Version des GA-Knotens mit spezifischem LZ-Treiber gebunden werden, so können die im nachfolgenden Beispiel *kursiv* markierte Zeile angepaßt werden. Dabei kann nur eine Routinen-Bibliothek LZxxLIB.LIB eingebunden werden.

Beispiel einer LNK-Datei (QUELLEN\GAK\GAK.LNK):

```
GAK+
INIT+EXIT+LOOP+
IIZ+ILZ+ISP+IES+OIZ+SYSCHK+ITAST+
DPP_INIT+DPP_EXIT+DPP_CMD+DPP_POLL+
FLIMMCHK+ROLLMELD+START_DP
GAK.EXE/NOE/NOD/CP:1/STACK:5632/SEG:256
GAK.MAP/MAP:FULL
..\SP\SP_LIB.LIB+
..\ES\ES_DLIB.LIB+
..\LZ\LZ_LIB.LIB+
..\IZ\IZ_LIB.LIB+
..\IZ\FND_10\IZ01LIB.LIB+
..\IZ\AEGKS120\IZ02LIB.LIB+
..\IZ\FND_CAPI\IZ03LIB.LIB+
..\IZ\IZ_DTRB.LIB+
..\CAPI_20\CAPILIB.LIB+
..\X25\X25_LIB.LIB+
..\UT\UT_LIB.LIB+
..\BS\BS_LIB.LIB+
..\SY\SY_LIB.LIB+
C:\EICON\X25DOS\LIB\X25MVL.LIB+
C:\MSVC_15\LIB\OLDNAMES.LIB+
C:\MSVC_15\LIB\LLIBC7.LIB
```

Dabei stehen folgende LZ-Treiber als Routinen-Bibliotheken zur Verfügung:

<b>Routinen-Bibliothek</b>	<b>Protokoll</b>	<b>Anbieter</b>
..\LZ\LZ_LIB.LIB	FND 1.0 + Ergänzungen (ISDN)	LZH der LH München
	FND 1.0 (ISDN)	GA-Knoten der LH München
	TSS 13a (ISDN)	diverse
..\LZARS232\LZRSLIB.LIB	FND 1.0 (V.24)	diverse
..\LZ EYLZEYLIB.LIB	3964R / RK 512 (V.24)	diverse
..\LZLONLZLNLIB.LIB	LonTalk mit FND 1.0 (V.24)	diverse

Für die Generierung eines GA-Knotens mit spezifischem LZ-Treibern gehen sie wie folgt vor:

- Wechseln Sie mittels CD-Kommando in das Unterverzeichnis QUELLEN\LZ
- Geben Sie "DIR \*.LIB /s /p" ein und vermerken Sie das Unterverzeichnis und Dateinamen der gewünschten LZxxLIB.LIB
- Wechseln Sie mittels CD-Kommando in das Unterverzeichnis QUELLEN\GAK
- Kopieren Sie z.B. die Datei GAK.LNK auf GAK\_xxx.LNK (mit "xxx" als neuer Kennung)
- Ersetzen Sie in GAK\_xxx.LNK die Zeichenfolgen "GAK." durch "GAK\_xxx."
- Passen Sie in GAK\_xxx.LNK die Unterverzeichnisse für die einzubindene LZxxLIB.LIB an



- Starten Sie den Bindevorgang durch Aufruf von "C\_LINK @GAK\_xxx.LNK". Als Ergebnis wird GAK\_xxx.EXE und GAK\_xxx.MAP geliefert.

**4.1.5. Einbinden spezifischer IZ-Treiber unter LINUX**

Soll eine Version des GA-Knotens mit spezifischen IZ-Treibern gebunden werden, so erfolgt dieses analog zu DOS.

**4.1.6. GAK\_XXL.EXE unter LINUX**

Um das Erzeugen hersteller- bzw. projektspezifischer Varianten zu vermeiden, wurde der GA-Knoten unter LINUX so erweitert, dass max. 128 IZ-Treiber eingebunden werden können.

**Hinweis:** Die große Anzahl der IZ-Treiber führen zu einem erhöhten Bedarf an Hauptspeicher.

Neben dem bisherigen "Standard"-GA-Knoten (GAK.EXE mit IZ01-IZ03) wird zusätzlich eine Variante bereitgestellt, die alle bisher frei verfügbaren IZ-Treiber in der benötigten Anzahl enthält (GAK\_XXL.EXE).

**Hinweis:** Beim Einsatz der GAK\_XXL.EXE in bestehenden Installationen kann eine Neuordnung der Datenpunkte zum jeweiligen IZ-Treiber erforderlich sein (vgl. Treiber-Id. in den IZ- und IS-Zeilen der Konfigurationsdatei GAK.CFG) und davon abhängiger Datenpunkte wie "%IZmmnnSTAT-oooo".

Dabei ist zu beachten, dass die Numerierung der IZ-Treiber aus Kompatibilitätsgründen HEXADEZIMAL (01-80), die Treiber-Id. jedoch DEZIMAL (1-128) angegeben werden muss.

Die Aufteilung der IZ-Treiber auf die einzelnen Treiber-Id. erfolgte unter dem vorrangigen Gesichtspunkt, die Anpassungen bei bestehenden Installationen zu minimieren und trotzdem eine sinnvolle Zusammenfassung sowie eine ausbaufähige Struktur zu erreichen. Es gelten folgende Bereiche :

<b><u>IZnn (hex.)</u></b>	<b><u>Treiber-Id. (dez.)</u></b>	<b><u>Bemerkung</u></b>
00	0	* reserviert *
01 - 0F	1 - 15	Standard-GA-Knoten, erweitert in Anlehnung an Techn. Rathaus (BREF)
10 - 3F	16 - 63	Offene Protokolle wie FND, BACnet
40 - 7F	64 - 127	Herstellerspez. Protokolle

80 - EF	128 - 239	* reserviert *
F0	240	interne Datenpunkte
F1 - FF	241 - 255	* reserviert *

Bei der Aufteilung wurde bereits berücksichtigt, dass einige IZ-Treiber mehrfach eingebunden werden müssen.

Bei der nachfolgenden Konfiguration für die GAK\_XXL.EXE wurde die Anzahl für in bestehenden Installationen bereits mehrfach eingebundene IZ-Treiber erhöht. Zusätzlich wurde darauf geachtet, dass zwischen den einzelnen IZ-Treibern freie Bereiche verbleiben, die für spätere Mehrfach-Verwendungen der benachbarten IZ-Treiber oder zusätzliche Treiber genutzt werden können.

LNK-Datei für GAK\_XXL.EXE (QUELLEN\GAK\gak\_xxl.lnk; ergänzt um IZnn, Treiber-Id. und das verwendete Protokoll bzw. zugehörige System):

rm gak\_xxl.exe

```
cc -o gak_xxl.exe gak.o init.o exit.o loop.o \
    iiz.o oiz.o ilz.o ies.o isp.o start_dp.o \
    dpp_init.o dpp_exit.o dpp_cmd.o dpp_poll.o \
    rollmeld.o flimmchk.o itast.o syschk.o \
    ../SP/sp_lib.a \
    ../ES/es_lib.a \
    ../LZ/lz_lib.a \
    ../IZ/iz_lib.a \
    # IZnn(hex.) Treiber-Id.(dez.) Protokoll/System
    # -----
    ../IZ/SMS/iz01lib.a \      # 01      1      SMS (D1, D2, E+), Fax, Mail
    ../IZ/AEGKS120/iz02lib.a \  # 02      2      AEG KS120 (A120)
    ../IZ/FND_CAPI/iz03lib.a \  # 03      3      FND 1.0 (V.24, ISDN, UDP/IP)
    ../IZ/KP_P90/iz04lib.a \   # 04      4      Kieback&Peter P90 (DDC 3000,
    #                                     HRP,MRP)
    ../IZ/MULT_A3K/iz05lib.a \  # 05      5      Multitone (ACCESS
3000)
    ../IZ/LON/iz06lib.a \      # 06      6      LON mit FND/LON-Kernel
    ../IZ/LON/iz07lib.a \      # 07      7
    ../IZ/LON/iz08lib.a \      # 08      8
    ../IZ/LON/iz09lib.a \      # 09      9
    ../IZ/LON/iz0Alib.a \      # 0A     10
    ../IZ/LON/iz0Blib.a \      # 0B     11
    ../IZ/LON/iz0Clib.a \      # 0C     12
    ../IZ/FND_10/iz10lib.a \    # 10     16      FND 1.0 (V.24)
    ../IZ/FND_10/iz11lib.a \    # 11     17
    ../IZ/FND_10/iz12lib.a \    # 12     18
    ../IZ/FND_10/iz13lib.a \    # 13     19
    ../IZ/FND_10/iz14lib.a \    # 14     20
    ../IZ/FND_10/iz15lib.a \    # 15     21
    ../IZ/FND_10/iz16lib.a \    # 16     22
    ../IZ/FND_10/iz17lib.a \    # 17     23
    ../IZ/BNC/iz20lib.a \      # 20     32      BACnet/IP (Vendor: AMEV)
    ../IZ/BNC/iz21lib.a \      # 21     33      BACnet/IP (Vendor: DTF)
    ../IZ/BNC/iz28lib.a \      # 28     40      MODBUS (IP od.. RTU)
    ../IZ/BNC/iz29lib.a \      # 29     41
    ../IZ/BNC/iz2Aib.a \       # 2A     42
    ../IZ/BNC/iz2Blib.a \      # 2B     43
    ../IZ/KM_SUCOM/iz40lib.a \   # 40     64      Moeller SUCOM (PS...)
    ../IZ/KM_SUCOM/iz41lib.a \   # 41     65
    ../IZ/KM_SUCOM/iz42lib.a \   # 42     66
    ../IZ/KM_SUCOM/iz43lib.a \   # 43     67
    ../IZ/MILES_AS/iz48lib.a \   # 48     72      MESSNER Miles
    ../IZ/MILES_AS/iz49lib.a \   # 49     73
    ../IZ/MILES_AS/iz4Alib.a \   # 4A     74
    ../IZ/MILES_AS/iz4Blib.a \   # 4B     75
    ../IZ/AND_INF/iz60lib.a \    # 60     96      ANDOVER Infinity
    ../IZ/DEKATEL/iz62lib.a \    # 62     98      VIESSMANN Dekatel
    ../IZ/EY_2400/iz64lib.a \    # 64    100      Sauter EY 2400
    ../IZ/EY_3600/iz66lib.a \    # 66    102      Sauter EY 3600
    ../IZ/GFR/iz68lib.a \       # 68    104      GfR (S-Bus / S-Bus+)
    ../IZ/MK_DC97/iz6Alib.a \    # 6A    106      Froeling DC97
    ../IZ/RAMCC600/iz6Clib.a \   # 6C    108      RAM CC600
    ../IZ/SAIA_BUS/iz6Elib.a \   # 6E    110      SAIA S-Bus (PCD...)
```

	../Z/SE_IPC/iz70lib.a \	# 70	112	Straub Electronic
	../Z/TRENDNCN/iz72lib.a \	# 72	114	TREND CNC
	../Z/LS_PRV/iz74lib.a \	# 74	116	L&S EKL/PRV1/PRV2/BSG
Zugriffe)	../Z/AS511/iz76lib.a \	# 76	118	AS511 (nur lesende
	../Z/PC_ILC/iz78lib.a \	# 78	120	Phoenix Contact ILC od. FC 350
	../Z/iz_dtrb.a \			
	../Z/iz_lib.a \			
	../CAPI_20/capilib.a \			
	../IP_4/ip_lib.a \			
	../UT/ut_lib.a \			
	../BS/bs_lib.a \			
	../SY/sy_lib.a \			
	../IPC/ipc_lib.a \			
	/usr/lib/libcapi20.a \			
	/usr/lib/libm.a			

## 4.2. Einsatzumgebung

Der Installationsort des GA-Knotens muß den herstellereigenen Anforderungen des einzusetzenden Rechners und der anderen Komponenten (z.B. Steckkarten, Drucker,...) genügen und über die Anschlüsse für die systeminternen und -externen Schnittstellen, sowie die Stromversorgung verfügen.

**Hinweis:** Empfohlen wird die Installation aller Komponenten in einem abschließbaren Stahlschrank mit ausreichender Lüftung und Erdung.

Auf dem Laufwerk C: wird unter DOS das Unterverzeichnis \ISDN\xxxx ("xxxx" Bezeichnung der ISDN-Karte, z.B. DIVA, SXN, SCOM, ...) eingerichtet, in dem alle erforderlichen für die ISDN-Karte erforderlichen Dateien (Treiber usw.) gespeichert werden.

Ebenfalls auf dem Laufwerk C: wird das Unterverzeichnis \GAK eingerichtet. Dieses dient als aktuelles Arbeitsverzeichnis, d.h. das Programm für den GA-Knoten wird aus diesem Verzeichnis heraus gestartet, so dass alle vom GA-Knoten erzeugten Dateien hier erzeugt werden. Die einzelnen Dateien und deren Funktion werden im Datenkatalog beschrieben.

Unter diesem Unterverzeichnis wird das Unterverzeichnis \GAK\EXE eingerichtet und in den Suchpfad (vgl. Umgebungs-Variable PATH) eingefügt. Hier werden alle zum Ablauf des GA-Knotens erforderlichen EXE- und BAT-Dateien abgelegt.

Desweiteren müssen folgende Unterverzeichnisse und Dateien eingerichtet sein (vgl. auch die Datenkataloge zu den Modulen GAK, IZ, SP und LZ):

\GAK\CFG	Unterverzeichnis GA-Knoten-Konfiguration
\GAK\CFG\GAK.CFG	GA-Knoten-Konfiguration
\GAK\CFG\DP_ATB.TXT	FND-Attribut-Texte
\GAK\CFG\DP_DIM.TXT	FND-Dimensions-Texte
\GAK\CFG\ANRUFER.CHK	Anrufer-Berechtigungen
\GAK\IZ	Unterverzeichnis IZ-Konfiguration
\GAK\IZ\IZ03CONN.VBP	Verbindungs-Profil für die am IZ03-Modul angeschlossenen IZ's
\GAK\IZ\A3K	Unterverzeichnis Konfiguration für Multitone Access 3000
\GAK\IZ\SMS	Unterverzeichnis Konfiguration für SMS
\GAK\SP\CFG	Unterverzeichnis Schaltprogramm-Konfiguration
\GAK\SP\CFG\SPP.CFG	Schaltprogramm-Parameter (ggf. leer)
\GAK\SP\CFG\KAL.CFG	Schaltprogramm-Kalender (ggf. leer)
\GAK\SP\PRG	Unterverzeichnis Schaltprogramme
\GAK\SP\PRG\*.ZSP	Zeit-Schaltprogramme
\GAK\SP\PRG\*.ESP	Ereignis-Schaltprogramme
\GAK\LZ\VBP	Unterverzeichnis Verbindungs-Profile
\GAK\LZ\VBP\VBP_oooo.nnn	Verbindungs-Profil zum Objekt "oooo" in der Version "nnn"
\GAK\LZ\IWA	Unterverzeichnis Istwert-Abfrage
\GAK\LZ\IWA\IWA_oooo.nnn	Istwert-Abfrage zum Objekt "oooo" mit der "Iz_subid" "nnn"

Neben den üblichen von DOS verwendeten I/O-Adreßbereichen und Interrupts (IRQ) werden vom GA-Knoten zusätzlich folgende verwendet bzw. reserviert:

<b>I/O-Adresse (hex.)</b>	<b>IRQ</b>	<b>Bezeichnung</b>
3F8 - 3FF	4	COM1: reserviert zum Anschluß eines externen ANSI-Bildschirms als Statusbildschirm und Tastatur
2F8 - 2FF	3	COM2: AEG SPS (vgl. "IZ02")
3E8 - 3EF	10	COM3: reserviert für zukünftige IZ
2E8 - 2EF	11	COM4: reserviert für zukünftige IZ
3BC- 3BF	7	LPT1: Protokoll-Drucker
380 - 387	5	EICON-Karte für X.25-Schnittstelle mit Memory-Bereich D000 - D1FF (reserviert für zukünftige IZ oder Erweiterungen)
240 - 24F	5	ISDN-Karte DIVA als PCMCIA-Steckkarte mit Treiber in der Version 5.02 und spezifischen Ladeprogramm DLOAD_I5, falls keine EICON-Karte installiert ist
240 - 24F	12	ISDN-Karte DIVA als ISA-Bus-Steckkarte
D000 - D1FF EICON- Memory-Bereich	5	ISDN-Karte SXn, SCOM als ISA-Bus-Steckkarte, falls keine Karte installiert ist (statt I/O-Adresse wird hier der benötigt)
D200 - D3FF Karte	12	ISDN-Karte SXn, SCOM als ISA-Bus-Steckkarte, falls EICON- installiert ist

Als letztes auszuführendes Kommando in der Datei AUTOEXEC.BAT wird der GA-Knoten mit Ausgabe der aktuellen Werte und Status am Bildschirm gestartet. Ist im Rechner eine Bildschirnkarte (z.B. VGA) integriert, so kann der daran angeschlossene Monitor nach dem erfolgreichen Start abgeschaltet werden, da der Bildschirminhalt in der Bildschirnkarte gespeichert wird.

Als Betriebssystem wird MS-DOS in der Version 6.22 (deutsch) installiert. Die Konfigurationsdateien haben z.B. folgenden Aufbau :

**CONFIG.SYS**

```

REM    MS-DOS im oberen Speicherbereich (HMA) installieren
REM
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH,UMB
STACKS=9,256

REM    DIEHL-ISDN-Treiber im Protected Mode starten
REM
DEVICE=C:\ISDN\DPMS.EXE
DEVICEHIGH=C:\ISDN\DIDD.SYS

REM    Bildschirm- und Tastatur-Treiber starten
REM
DEVICEHIGH=C:\DOS\ANSI.SYS
COUNTRY=49,437,C:\DOS\COUNTRY.SYS

REM    DOS-Parameter setzen
REM
BUFFERS = 50
FILES = 50
SHELL=C:\DOS\COMMAND.COM /E:2048 /P

```

**AUTOEXEC.BAT**

```

@ECHO OFF

REM    Cache für Festplattenzugriffe und Tastatortreiber starten
REM
LOADHIGH C:\DOS\SMARTDRV.EXE
LOADHIGH C:\DOS\KEYB GR,437,C:\DOS\KEYBOARD.SYS

REM    Zugriffspfade und Eingabe-Prompt definieren
REM
PATH=C:\GAK\EXE;C:\DOS;
PROMPT $p $T$H$H$H$H$H$H $g

REM    DIEHL-ISDN-Treiber im Protected Mode (DPMS wird in CONFIG.SYS geladen)
REM    ( mit Version 5.03 BETA mit Stand vom 10.01.96 ff. !!!! )
REM
C:\ISDN\TEDETSI -i12 -a240
LOADHIGH C:\ISDN\CAPI20

REM    GA-Knoten starten wahlweise mit folgenden Schaltern :
REM
REM    /s Status-Bildschirm
REM    /lr Ausgabe von REJECT-Datagrammen in Datei REJECT.LOG
REM    /le Ausgabe von ERROR-Datagrammen in Datei ERROR.LOG
REM    /li Ausgabe von ungültigen Datagrammen in Datei INVALID.LOG
REM
cd \GAK
GAK /s

```

**Hinweis:** Der Aufbau der Dateien CONFIG.SYS und AUTOEXEC.BAT ist abhängig von den verwendeten Schnittstellen und anderen eingesetzten System-Komponenten. Beim Einsatz des GA-Knotens in anderen Projekten oder Änderung der oben beschriebenen Konfiguration können erhebliche Abweichungen auftreten.

Insbesondere muß die eindeutige Verwendung der Interrupts (IRQ), I/O- und Hauptspeicher-Adreßbereiche sichergestellt sein. Desweiteren kann die Reihenfolge, in der die Treiber geladen werden, von erheblicher Bedeutung für die einwandfreie Funktion des GA-Knotens sein.

**Hinweis:** Eine Änderung der Einsatzumgebung kann den einwandfreien Ablauf des GA-Knotens beeinträchtigen (z.B. Start des GA-Knotens mit Ausgabe von Trace-Meldungen auf eine Datei; vgl. unten). Der Betreiber der GA-Knoten muß sicherstellen, dass nur qualifizierte Personen Änderungen an der Einsatzumgebung vornehmen.

Die Einsatz-Umgebung unter LINUX basiert auf der Distribution der Fa. SuSE - derzeit Version 8.1 mit Kernel 2.4.19. Als Arbeitsverzeichnis wird das Home-Verzeichnis des Benutzers "gak" (/home/gak) verwendet. Die davon ausgehende Verzeichnisstruktur und das Format der Dateien entspricht weitestgehend der unter DOS. Im wesentlichen sind nur die Bezeichnungen der Schnittstellen anzupassen (COM1 => /dev/ttyS0). Detaillierte Informationen zur Installation von LINUX und dem GA-Knoten finden sich im zugehörigen Installations-Handbuch.

**Hinweis:** Beachten Sie bitte, dass Verzeichnis- und Dateinamen unter LINUX zeichensensitiv sind, d.h. nach Gross-/Kleinschreibung unterschieden werden und in den Konfigurationsdateien übereinstimmend angegeben sein müssen.

Voreinstellung ist die Kleinschreibung. Dateinamen, die z.B. Objekt-Kennungen enthalten, enthalten diese entsprechend der Schreibweise in der CFG-Datei - z.B. "vbp\_BREF.001".

Abhängig von der LINUX-Version, der Anzahl und Größe der gestarteten Prozesse und Parametrierung des GA-Knotens kann eine Erweiterung des Hauptspeichers erforderlich sein. Stellen Sie sicher, dass keine rechenbereiten Prozesse auf Festplatte ausgelagert werden (z.B. Überwachung des LOAD-Wertes und des Prozesses KSWAPD mittels TOP). Desweiteren muss ausreichend Hauptspeicher zum Zwischenspeichern von Dateien bereitgestellt werden. Es wird empfohlen dass die Angaben "free" (Mem), "buff" und "cached" zusammen mindestens 50% des verfügbaren Hauptspeichers ergeben.

Die Festplatte, auf der das Arbeitsverzeichnis des GA-Knotens eingerichtet ist, muss über einen ausreichenden freien Speicherplatz verfügen und sollte einen Füllungsgrad von max. 60% haben.

### **4.3. Prozesse**

Das Programm des GA-Knotens (GAK.EXE) läuft exklusiv ab, d.h. mit Ausnahme der erforderlichen Treiber und Systemprozesse für die Schnittstellen zu den IZ und den Leitzentralen (V.24, ISDN, Netzwerk), laufen keine konkurrierenden Prozesse ab.



Unter LINUX können jedoch optional weitere Programme ablaufen. Dabei ist jedoch sicherzustellen, dass keine Beeinträchtigung des GA-Knotens auftritt.

In einer Programmschleife werden alle Programm-Module des GA-Knotens zyklisch aktiviert (Polling).

Für den Ablauf des GA-Knotens wesentliche Module sind der Programm-Kern (GAK), die Kommunikation zu den IZ's (IZ und Untermodule IZnn), die Kommunikation zu den Leitzentralen inkl. Bearbeitung der zugehörigen Karteipunkte (LZ), der elektronische Schlüsslersatz (ES) und die Schaltprogramme (SP). Die Module kommunizieren untereinander durch Versenden von FND-Datagrammen (Version 1.0 und Ergänzungen). Jedes Modul kennzeichnet die von ihm erzeugten Datagramme durch eine eigene "Iz\_subid" im APCI des Datagrammes, die wie folgt kodiert werden:

- 0:        allgemeines Datagramm, d.h. dieses betrifft alle Module
- 1:        internes Datagramm des Programmkerns (GAK)
- 2:        internes Datagramm der Schaltprogramme (SP)
- 3:        Datagramm der Leitzentrale Haustechnik (LZ)
- 4:        Datagramm des BWD-Gefahr
- 5:        Datagramm des BWD-Einbruch
- 6:        Datagramm des elektr. Schlüsslersatzes (ES)
- 7:        Datagramm mit zwischengespeichertem Wert von IZ (z.B. DEMA)
- 8-9:     \*\*\* reserviert für zukünftige Erweiterungen \*\*\*
- 10-255:  Behandlung von komplexeren Abfragen

#### **4.4. Daten- und Funktionssicherung**

Der Betreiber sichert die Stromversorgung des GA-Knotens z.B. durch eine unterbrechungsfreie Stromversorgung (USV). IZ's und GA-Knoten überwachen ihre Funktion gegenseitig.

Für SPS- und LON-basierte Systeme bestehen entsprechende Programmier-Richtlinien der Landeshauptstadt München. In diesem Fall führt der IZ-Treiber die erforderlichen Maßnahmen automatisch durch.

Für alle anderen Systeme wird empfohlen, innerhalb der IZ einen Datenpunkt einzurichten, auf den der GA-Knoten mittels Schaltprogramm minütlich einen Schaltbefehl ausgibt. Den Schaltbefehl interpretiert die IZ als Rücksetz-Befehl für einen Watchdog-Timer. Bleibt der Schaltbefehl z.B. 5 mal hintereinander aus, so interpretiert die IZ dies als Störung

der Kommunikation mit dem GA-Knoten und wechselt in den "Minimalbetrieb". Mit dem nächsten Schaltbefehl kehrt die IZ automatisch in den Normalbetrieb zurück.

**Hinweis: Der Planer muss den Minimalbetrieb der IZ projektspezifisch festlegen und sicherstellen, dass alle grundlegenden, insbesondere die sicherheitsrelevanten Funktionen von der IZ eigenständig erbracht werden.**

Erkennt der GA-Knoten den Ausfall einer IZ, so vermerkt er den Zustand im internen Schaltpunkt "%IZmmnnSTAT-oooo" ("mm" und "nn" sind die IZ-spezifischen Treiber-Identifikatoren) und meldet die Zustandsänderung entsprechend der Parametrierung des Meldungsverhaltens an die Leitzentrale.

Nach Wiederverfügbarkeit des ausgefallenen Gerätes wird die Kommunikation automatisch wieder aufgenommen.

Die Systemzeit des GA-Knoten basiert auf der vom Betriebssystem geführten Zeit. Diese weist erfahrungsgemäß Ungenauigkeiten auf, die über einer längere Laufzeit zu relevanten Abweichungen führen können. Aus diesem Grund kann parametrierung, anhand welcher Zeitbasis die Systemzeit abgeglichen werden soll (vgl. ZA-Zeile in der Konfigurationsdatei). Zur Verfügung stehen folgende Dienste:

- ISDN (INFO auf dem D-Kanal beim Verbindungs-Aufbau)
- NETDATE (Abgleich am Zeit-Server im Netzwerk; nur unter LINUX)
- DCF 77 o.ä. (funkgesteuerte Zeitbasis über integrierte Baugruppe)

Einzelne ISDN-Anschlüsse liefern keine Zeitinformation (INFO-Indication mit Info=0x0029). Wurde der Zeitabgleich im GA-Knoten aktiviert, so führte dieses zu kurzfristigen Wiederholungsversuchen. Nun wird die Wiederholung abgebrochen, wenn der Verbindungsaufbau im D-Kanal erfolgreich war (CONNECT\_ACTIVE-Indication). Ab der Version 1.3i verwendet der Zeitabgleich über ISDN nach einem fehlerhaften Versuch die Wartezeit aus der VBP-Datei des ersten Objektes, sonst 5 Minuten.

Verfügt ein GA-Knoten über keinen Zeitabgleich, so ist die Rechneruhr i.a. ungenauer als die IZ-interne Uhr. Wurde kein Zeitabgleich aktiviert, so wird nun der Zeitabgleich zu den IZ's unterdrückt. Bei Systemen, die einen regelmäßigen Zeitabgleich zur Erkennung des Minimalbetriebs verwenden, werden ungültige Werte gesandt.

Der Zeitabgleich zu den angeschlossenen IZ's (vgl. ZA-Zeile in der Konfigurationsdatei) nur dann, wenn der Zeitabgleich aktiviert ist und die Systemzeit des GA-Knotens neuer ist, als Datum/Zeit der aufgerufenen EXE-Datei (z.B. gak.exe,

iz\_test.exe). Dadurch bleibt die IZ-interne Zeit z.B. nach einem Neustart des GA-Knoten mit einer defekten BIOS-Uhr/leere Batterie unverändert erhalten.

## **5. Software-Entwurf**

### **5.1. Design**

#### **5.1.1. Vergabe von Programm- und Modulnamen**

Ergänzend zum Standard-GA-Knoten kann ein weiterer, herstellerspezifischer IZ-Treiber unter IZ04 in das Programm für den GA-Knoten eingebunden werden. Bei LINUX sind es sogar insgesamt 128 IZ-Treiber - IZ01 bis IZ80 (Hexadezimal numeriert !).

Die Namen der Dateien mit dem ablauffähigen Programm für den GA-Knoten sind wie folgt aufgebaut :

gak.exe	Standard-GA-Knoten
gak_xxl.exe	Standard-GA-Knoten mit allen frei verfügbaren herstellerspezifischen IZ-Treibern ab IZ04 in der bisher benötigten Anzahl.
gak_pppp.exe	Standard-GA-Knoten mit herstellerspezifischem IZ-Treiber IZ04 und "pppp" als max. 4-stelligem Kürzel für das herstellerspezifische Protokoll
gak_oooo.exe	Standard-GA-Knoten mit mehreren herstellerspezifischen IZ-Treibern und "oooo" als max. 4-stelliger Objekt-Kennung

Der GA-Knoten wird modular aufgebaut, d.h. alle nicht zum Hauptprogramm gehörenden Programmteile werden in gesonderten Verzeichnissen gespeichert, kompiliert und in einer Routinen-Bibliothek zusammengefasst. Folgende Module werden in den gleichnamigen Verzeichnissen als Routinen-Bibliotheken zusammengefasst :

BS	Bildschirm-Steuerung
CAPI_20	CAPI-Schnittstelle in der Version 2.0
ES	Elektronischer Schlüsseleratz
IP_4	Internet-Protokoll in der Version 4
IPC	Inter-Process-Communication
IZ	IZ-Treiber
LZ	LZ-Treiber
SP	Schaltprogramme
SY	Systemspezifische Funktionen
UT	allgemeine Funktionen (Utilities)
X25	X.25-Protokoll

Folgende Hauptprogramme werden in den gleichnamigen Verzeichnissen zusammengefasst :

GAK	GA-Knoten
I2X	ISDN-/X.25-Router
SSA	Standard-Schnittstellen-Adapter

Folgende Hilfsprogramme werden in den gleichnamigen Verzeichnissen zusammengefasst :

CFG2REF	(Rück-)Konvertierung der CFG-Datei in eine FND-Referenzdatei
COM_FIFO	Aktivierung des FIFO-Speichers im UART (nur DOS)
FND_LOG	Anzeige der in LOG-Dateien gespeicherten FND-Datagramme
FWD	Fernwartungsdienst
GEN_CFG	Konvertierung einer FND-Referenzdatei in die CFG-Datei
ESN2DOS	Konvertierung des ESn-Dateien unter LINUX in das DOS-Format
IZ_TEST	Interaktiver Test der IZ-Treiber
REBOOT	Neustart des Rechners (nur DOS)
TSYNC	Abgleich der Systemzeit (nur LINUX)

### 5.1.2. Vergabe von Funktions- und Variablennamen

Durch einen einheitlichen Aufbau der Namen für die Funktionsaufrufe und die zugehörigen Dateinamen wird erreicht, dass anhand des Funktionsnamens erkannt werden kann, wo die zugehörige Quelldatei zu finden ist und welchen Namen sie hat. Eine Einbeziehung der Extension ist nicht notwendig, da hier nur wenige Alternativen bestehen, die sich gegenseitig ausschließen (C- oder Assembler-Quelle). Als Extension werden die üblichen Voreinstellungen verwendet :

C	C-Quelle
ASM	Assembler-Quelle
OBJ	compilierter/assemblierter Objektcode unter DOS
O	compilierter/assemblierter Objektcode unter LINUX
LIB	Routinen-Bibliothek unter DOS
A	Routinen-Bibliothek unter LINUX
LNK	Binde-Makro
EXE	ablauffähiges Programm

Routinen, die dem Hauptprogramme spezifisch zugeordnet werden, erhalten den entsprechenden 3-stelligen Prefix "GAK" im Funktionsnamen, der durch das Trennzeichen "\_" vom Dateinamen abgesetzt wird.

**Beispiel:** Datei "ABC.c" im Verzeichnis "GAK" verwendet den Funktionsnamen "GAK\_ABC", mit dem die Funktion aufgerufen wird.

Andere Routinen sind unabhängig vom Hauptprogramm und werden ggf. in mehreren Programmen verwendet. Diese Routinen werden getrennt nach ihren Aufgabenbereichen in spezifischen Verzeichnissen verwaltet. Die Verwendung des spezifischen 2-stelligen Prefix erfolgt analog zu den Hauptprogrammen. Der Name des Verzeichnisses ist identisch zum Prefix. Alle Routinen werden in einer Objekt-Bibliothek zusammengefaßt, die auch mit dem Prefix im Dateinamen beginnt, gefolgt von "\_LIB" und der Extension "LIB". Als Prefix werden oben genannten Namen Kennungen verwendet.

**Beispiel:** Datei "ABC.c" im Verzeichnis "BS" verwendet den Funktionsnamen "BS\_ABC", mit dem die Funktion aufgerufen wird. Das Objekt "ABC.obj" wird im selben Verzeichnis in die Bibliothek "BS\_LIB.lib" eingebunden.

**Hinweis: Die Routinen-Bibliotheken BS und SY werden so aufgebaut, dass sie weitestgehend nicht spezifisch für den GA-Knoten sind, d.h. sie können später auch in anderen Software-Produkten eingesetzt werden.**

### **5.1.3. Definition der Systemzustände**

Der GA-Knoten arbeitet im zyklischen Polling-Verfahren unter Einbeziehung von Routinen für die Behandlung von Unterbrechungen (Interrupt-Service-Routinen), d.h. er ruft die einzelnen Module zyklisch auf (vgl. Routine GAK\_LOOP). Diese prüfen, ob eine Bearbeitung erforderlich ist, und führen diese ggf. aus.

Die einzelnen Ein- und Ausgaben zwischen dem GA-Knoten und den angeschlossenen Geräten (IZ's, LZ's und Drucker) als auch zwischen den programminternen Modulen werden vom GA-Knoten asynchron behandelt. Zu diesem Zweck werden auszugebende Daten in Warteschlangen-Dateien nach dem FIFO-Prinzip zwischengespeichert, bis der Empfänger bereit ist (vgl. Dateien \*.WSL in den Datenkatalogen).

Die von den IZ.-Treibern für Schaltprogramm-Befehle, Verkettungen und zyklische Abfragen angelegten Warteschlangen-Dateien werden nun in ihrer Größe beschränkt. Dadurch bleibt eine für den Programmablauf notwendige Festplatten-Kapazität weitestgehend gesichert.

Diese Begrenzung gilt ebenso für Protokoll-, Error-Log- und Log-Dateien (vgl. DR-/EL-Zeile in der Konfigurationsdatei, als auch Schalter /ln, /lr und /le).

Die Werte aller Datenpunkte werden bei jeder Erfassung zusammen mit dem Zeitstempel vermerkt. Dieses sekundäre Prozessabbild bildet die Grundlage für die bei der zyklischen Istwertabfrage (IWA; vgl. LZ-Modul) gelieferten Werte und Zustände.

Konnten Datenpunkte nicht gelesen werden, so der Zustand bzw. Wert UNDEFINIERT gemäß FND-Spezifikation mit dem zugehörigen Zeitstempel analog geliefert.

Melde- und Schaltpunkte werden i.a. aufgrund ihrer Spontanmeldungen gemäß FND-Spezifikation (USM) aktualisiert. Bleiben diese aufgrund von Ausblendungen, Flimmern, Abschaltung bzw. Neustart einer Unterstation oder Störung der Kommunikation innerhalb bzw. mit der MSR-Insel aus, so können Wertänderungen nicht angezeigt werden. Analog zu den Mess-, Stell- und Zählpunkten werden auch die Melde- und Schaltpunkte stündlich abgefragt und auf zu meldende Wertänderungen überwacht. Das Verfahren passt sich automatisch an die herstellerspezifischen Antwortzeiten der IZ an und stellt auch die Behandlung der anderen Programm-Module sicher.

### 5.1.4. Globale Variablen

Die Übergabe von Parametern zwischen Routinen unterschiedlicher Bibliotheken bzw. zu den Routinen des Hauptprogrammes erfolgen stets über die Aufrufparameter der Routine. Nur innerhalb der Routinen eines Verzeichnisses wird ein gemeinsamer Speicherbereich verwendet. Dieser wird in der Datei mit dem Namen "xx\_INCL.H" vereinbart, wobei "xx" dem 2- bis 4-stelligen Prefix entspricht, wobei beim 4-stelligen Prefix das Trennzeichen "\_" entfällt (z.B. CAPIINCL.H). Innerhalb dieser Datei werden auch die Funktionen mit ihrem Typ und den Typen der Aufrufparameter (Prototypen) definiert. Durch Verwendung des Preprozessor-Defines "DEF\_PROTOTYP" in Verbindung mit dem INCLUDE-Aufruf für die Datei lassen sich diese Funktions-Definitionen auch von anderen Modulen her laden. Dadurch ist die für die C-Programmierung wichtige Prüfung der Aufrufparameter bereits bei der Compilierung möglich.

Die einzige Ausnahme betrifft das dynamische Einschalten von Test-Funktionen (z.B. Trace-Meldungen). Diese Test-Funktionen werden bei der Software-Entwicklung nach Bedarf eingebaut und bleiben auch für spätere Tests erhalten, soweit dieses sinnvoll ist.

Das Hauptprogramm für den GA-Knoten wird i.a. ohne zusätzliche Parameter aufgerufen. Durch die Angabe der folgenden Schalter als Aufrufparameter können die Testfunktionen aktiviert werden :

- /Lx Erzeugen von LOG-Dateien (zu x siehe nachfolgende Erläuterungen)
- /Pnnn Ausblenden von Meldungen auf den Protokoll-Gerät beim Programmstart
- /Sd Ausgabe der aktuellen Werte und Status am Bildschirm, falls "d" auch mit Durchsatzanzeige
- /Tx Erzeugen von Trace-Meldungen (zu x siehe nachfolgende Erläuterungen)
- /Znnn Sprung der Systemzeit nach einer Wartezeit von nnn Sekunden

Empfängt der GA-Knoten von einer IZ ein REJECT-, ERROR- oder unzulässiges FND-Datagramm, so werden diese für angegebenen Schalter (/LR, /LE oder /LI) in getrennten LOG-Dateien binär protokolliert (Näheres vgl. im Datenkatalog).

Wird der Schalter "/P1" angegeben, so wird beim Programmstart bis auf die anstehenden Stör- und Gefahr-Meldungen (ab\_id:25 - 29 und 44) die Protokollierung unterdrückt. Nachfolgende Änderungen werden jedoch gemäß der Parametrierung protokolliert.

Die Ausgabe der aktuellen Werte und Status erfolgt für angegebenen Schalter /S in einer Bildschirmmaske, deren einzelne Felder mittels der ANSI-ESCAPE-Sequenzen positioniert werden. Der Aufbau der Maske ist

programmspezifisch. Bei Verwendung eines externen Bildschirms an der COM1-Schnittstelle muß sichergestellt sein, dass das Standard-Output-Device (stdout) darauf umgelenkt ist.

Die Trace-Meldungen benötigen zur Ausgabe ein scrollfähiges Gerät (Bildschirm, Drucker) oder eine Datei, d.h. die Meldungen werden sequentiell nacheinander geschrieben und jeweils mit einem Zeilenabschluß (Carriage Return und/oder LineFeed) abgeschlossen. Zum Erzeugen von Trace-Meldungen (Schalter "/T") wird jeweils einer der folgenden Schalter angefügt, der die Ereignisse genauer spezifiziert, für die eine Trace-Meldung ausgegeben werden soll :

- I Trace aller Eingabedaten (Input) von Schnittstellen, Dateien oder Tastatur
  - I1 wie "I", jedoch nur Eingabedaten von Schnittstellen
  - I2 wie "I", jedoch nur Eingabedaten, die aus Dateien gelesen werden
  - I3 wie "I", jedoch nur Eingabedaten von Tastatur
  
- O Trace aller Ausgabedaten (Output) auf Schnittstellen, Dateien
  - O1 wie "O", jedoch nur Ausgabedaten auf Schnittstellen
  - O2 wie "O", jedoch nur Ausgabedaten, die auf Dateien geschrieben werden
  - OS wie "O", jedoch nur Ausgabedaten, die auf Dateien für die Datenspeicherung geschrieben werden
  
- A Trace aller Routinen-Aufrufe (inkl. Parameter und Return-Werte) nach dem Aufruf (aus der aufgerufenen Routine), vor dem Aufruf (aus der aufrufenden Routine), nach der Rückkehr (aus der aufrufenden Routine) und vor der Rückkehr (aus der aufgerufenen Routine).
  - A1 wie "A", jedoch nur nach dem Aufruf (aus der aufgerufenen Routine)
  - A2 wie "A", jedoch nur vor dem Aufruf (aus der aufrufenden Routine)
  - A3 wie "A", jedoch nur nach der Rückkehr (aus der aufrufenden Routine)
  - A4 wie "A", jedoch nur vor der Rückkehr (aus der aufgerufenen Routine)
  
- R Rücksetzen des Trace-Ausgabe-Kanals auf den Anfang (Rewind / Reopen) nach jedem Schleifen-Durchlauf, in dem keine zu verarbeitenden Ereignisse auftraten (vgl. SY\_SIG\_xxxx).



- Dxxx Standardmäßig erfolgt die Ausgabe der Trace-Meldungen auf das Gerät, dass mittels "stderr" definiert ist. Durch Angabe des Schalters "D" mit einem gültigen Dateinamen für "xxx" erfolgt die Ausgabe der Tracemeldungen in die genannte Datei.
  
- R Rücksetzen der Trace-Ausgaben (Rewind auf der unter /TDxxx angegebenen Datei "xxx")
  
- X Werden zusätzliche Trace-Meldungen erzeugt, die nicht unter eine der bisher genannten Rubriken fallen, so können diese hiermit aktiviert werden.

Zum Testen von zeitabhängigen Abläufen (z.B. Zeitschaltprogrammen) kann der Schalter /Znnn angegeben werden. Nach einer Wartezeit von nnn Sekunden nach Empfang des letzten RSP-Datagrammes für die ausgegebenen Schaltbefehle oder nach Eingabe von "Z" an der Tastatur wird die Systemzeit des GA-Knotens auf den Zeitpunkt des nächsten auszuführenden Zeitschaltbefehls oder Kalender-Eintrags vorgestellt.

**Hinweis: Diese Funktion dient ausschließlich zu Testzwecken. Während des Programmlaufes sind andere Algorithmen für den Zeitabgleich (vgl. ZA-Zeile in der Konfigurationsdatei) zu deaktivieren. Nach Beendigung des Programmlaufes muß die Systemzeit des GA-Knotens restauriert werden. Dateien, die während des Programmlaufes angelegt oder aktualisiert werden (z.B. Protokoll-, Warteschlangen- oder LOG-Dateien), ist ggf. das Datum/Uhrzeit der jeweiligen vorgestellten Systemzeit zugeordnet.**

Zur Realisierung der Testfunktionen werden diverse Variable in der Datei "GAK\_TEST.H" zusammengefaßt, die als Schalter in den einzelnen Routinen abgefragt werden können und abhängig davon Testausgaben erzeugen. Aufgrund ihrer zentralen Funktion wird die Datei "GAK\_TEST.H" im übergeordneten Verzeichnis (...\\QUELLEN) verwaltet und von allen Routinen mittels INCLUDE-Befehl geladen. Die Namen der Variablen haben den Aufbau "TEST\_x\_y". Dabei ist "x" der Kennbuchstabe im Prefix und "y" die Kennung für den Schalter. Der Typ und die Belegung der Variablen ermöglichen i.a. eine logische Prüfung der folgenden Art :

if ( TEST\_T\_A || TEST\_T\_A2 ) dann Trace-Meldung ausgeben

Durch die Angabe eines Schalters (#define TEST\_T) für den Precompiler des C-Compilers kann angegeben werden, ob die Behandlung von Trace-Meldungen überhaupt berücksichtigt werden soll. Wird dieser Schalter in GAK\_TEST.H angegeben, so gilt er global für alle Programme. Andererseits kann er auch in der moduleigenen Datei "xx\_INCL.H" oder der jeweiligen Quell-Datei explizit gesetzt bzw. rückgesetzt werden, so dass er nur für dieses Modul bzw. diese Routine gilt, jedoch nicht für die Bibliotheken, die hinzugebunden werden.

**Hinweis:** Die Trace-Meldungen sind durch die maximale Größe des Segmentes, in dem die Text-Konstanten gespeichert werden, beschränkt, d.h. der Schalter TEST\_T wird standardmäßig nur in den Modulen GAK, IZ, IZnn und LZ gesetzt und muß bei Bedarf in den gewünschten Modulen ge- bzw. rückgesetzt werden.

**Hinweis:** Nach einer Änderung des Schalters müssen generell alle betroffenen Programmteile erneut kompiliert, ggf. zur Bibliothek zusammengefaßt und gebunden werden.

Die Trace-Meldungen haben stets den folgenden ggf. mehrzeiligen Aufbau :

(TT.MM.JJ HH:MM:SS) <Name der Funktion>: <Text und Parameter>

z.B.: (02.08.95 11:13:05) BS\_ABC: Nach Aufruf mit XYZ=123.456

Die Angabe des Datums und der Zeit ermöglichen auch die Rückverfolgung von Trace-Meldungen und deren Zuordnung zu anderen Ausgaben (z.B. Druckerausgaben) bei längeren Testläufen.

Eine spezifische Eigenart der Programmiersprache C ist die Definition der von mehreren Routinen gemeinsam verwendbaren Variablen. Diese müssen **einmalig vereinbart** und vorbelegt werden. Alle anderen Routinen müssen ihren Zugriff darauf als "extern" deklarieren. Die Vereinbarung erfolgt sinnvollerweise im Hauptprogramm bzw. in der Initialisierungsroutine eines Moduls (Routinen-Bibliothek). Hierzu müssen vor dem INCLUDE der jeweiligen Datei der Preprozessor-Schalter MAIN bzw. LIB\_INIT mittels "#define" gesetzt werden. Innerhalb der INCLUDE-Datei wird abhängig davon die jeweilige Variante ausgewählt.

### **5.1.5. Globale Konstanten**

Globale Konstanten (z.B. Fehlerkodierungen) werden analog zu den globalen Variablen in der modulspezifischen INCLUDE-Datei im system-sichtbaren Bereich vereinbart, so dass sie unabhängig vom Preprozessor-Defines "DEF\_PROTOTYP" verfügbar sind.

## **5.2. Basis-Module**

Wie schon bei der Vergabe der Programm- und Modulnamen bzw. der Funktions- und Variablennamen beschrieben wurde, werden wesentliche Funktionen in Routinen-Bibliotheken zusammengefaßt, die unabhängig vom Hauptprogramm sind. Diese Routinen sind aufgrund der Konventionen für den Funktionsnamen erkennbar und werden im Rahmen dieser Systemspezifikation, mit Ausnahme der Initialisierung-Routinen, nicht detailliert beschrieben.

**Hinweis: Nachfolgend werden häufig verwendete Funktionen als Beispiele genannt. Für detailliertere Informationen lesen Sie bitte die Beschreibungen und Kommentare in den Quelldateien!**

Folgende Routinen-Bibliotheken werden erstellt :

### **5.2.1. Übersicht der verwendeten Module**

**CAPILIB** In dieser Bibliothek im Unterverzeichnis CAPI\_20 werden alle Routinen zusammengefaßt, die mit der Kommunikation über die CAPI-Schnittstelle der ISDN-Karte zusammenhängen.

**IP LIB** In dieser Bibliothek im Unterverzeichnis IP\_4 werden alle Routinen zusammengefaßt, die mit der Kommunikation über die IP-Schnittstelle der Netzwerk-Karte zusammenhängen.

**IPC LIB** In dieser Bibliothek im Unterverzeichnis IPC werden alle Routinen zusammengefaßt, die mit der rechnerinternen Kommunikation mehrerer Prozesse zusammenhängen.

**X25 LIB** (nur DOS): In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit dem X.25-Protokoll zusammenhängen und nicht Bestandteil der nachfolgend beschriebenen Bibliothek X25MVL sind (z.B. Aufbau der Wählverbindung zur LZH, ...; nur DOS).

**Hinweis: Beim Binden des Hauptprogrammes werden die für die X.25-Schnittstelle erforderlichen Routinen-Bibliotheken X25\_LIB, X25MVL und OLDNAMES weiterhin angegeben, jedoch werden daraus Routinen nur dann eingebunden, wenn ein Aufruf vorhanden ist (vgl. GAK.LNK).**

**BS LIB** In dieser Bibliothek im Unterverzeichnis BS werden alle Routinen zusammengefaßt, die mit der Bildschirmsteuerung zusammenhängen (z.B. Löschen des Bildschirms, Positionieren des Cursor, ...).

**SY LIB** In dieser Bibliothek im Unterverzeichnis SY werden alle Routinen zusammengefaßt, die eine systemtechnische Funktion ausführen, d.h. einen direkten Zugriff auf eine Komponente des Rechner bzw. des Betriebssystems haben (z.B. Lesen/Schreiben der Real-Time-Clock, Test einer Speicherzelle, ...). Folgende Routinen aus dieser Bibliothek werden von nahezu allen anderen Routinen im Hauptprogramm und den anderen Modulen aufgerufen, so dass sie nicht als solche explizit dokumentiert werden:

SY_FI_...	Suchen von Dateien und zugehörigen Parametern
SY_PS_...	Verfahren zur Prozess-Steuerung wie Erzeugen und Überwachen von Kind-Prozessen
SY_SIG_...	Verfahren zur Signalisierung von Ereignissen

**UT LIB** In dieser Bibliothek im Unterverzeichnis UT werden alle die Funktionen zusammengefaßt, die nicht in eine der vorher genannten Bibliotheken gehören und trotzdem unabhängig von dem Hauptprogramm und den anderen Modulen sind. Sie lassen sich somit ggf. in mehreren Hauptprogrammen einsetzen. Folgende Routinen aus dieser Bibliothek werden von nahezu allen anderen Routinen im Hauptprogramm und den anderen Modulen aufgerufen, so dass sie nicht als solche explizit dokumentiert werden:

UT_DP_FIND...	Suchen von Einträgen in der Datenpunkt-Struktur
UT_ERR_LOG	Eintrag in der Error-Log-Datei anfügen
UT_T_TIME	Lesen der Systemzeit im Format "TT.MM.JJ HH:MM:SS"
UT_WSL...	Behandlung von Warteschlangen

**X25MVL** (nur DOS): Diese Bibliothek ist Bestandteil des "X.25 Network-Level Developer's Toolkit" der Firma EICON und wird uneingeschränkt verwendet. Sie dient zur Programmierung des X.25-Protokoll und wird in den Bibliotheken IZ01LIB und LZ\_LIB aufgerufen. Ergänzende Routinen zur Programmierung des X.25-Protokoll sind in der oben beschriebenen Bibliothek X25\_LIB zusammengefaßt.

**Hinweis: Diese Routinen entsprechen nicht den oben genannten Namenkonventionen, verwenden jedoch einheitlich den Prefix "x25" !**

**OLDNAMES** (nur DOS): Diese Bibliothek ist Bestandteil des Microsoft C/C++ - Compilers und wird uneingeschränkt verwendet. Sie ist erforderlich, da die Bibliothek X25MVL noch nicht ANSI-konforme Aufrufe der in Bibliothek LLIBCE enthaltenen Routinen durchführt (nur DOS).

**Hinweis: Diese Routinen entsprechen nicht den oben genannten Namenkonventionen !**

**LLIBCE** bzw. **LLIBC7** (nur DOS): Diese Bibliotheken sind Bestandteil des Microsoft C/C++ - Compilers und wird uneingeschränkt verwendet.

**Hinweis: Diese Routinen entsprechen nicht den oben genannten Namenkonventionen ! Die in dieser Bibliothek enthaltenen Routine werden als Bestandteil des Sprachumfanges der Programmiersprache C angesehen und werden in den Quelldateien zum GA-Knoten nicht explizit als aufgerufene Routinen dokumentiert.**

**LIBCAPI20** und **LIBM** (nur LINUX): Diese Bibliotheken sind Bestandteil der CAPI-Schnittstelle bzw. des GNU - Compilers und werden uneingeschränkt verwendet.

**Hinweis: Diese Routinen entsprechen nicht den oben genannten Namenkonventionen ! Die in dieser Bibliothek enthaltenen Routine werden als Bestandteil des Sprachumfanges der Programmiersprache C angesehen und werden in den Quelldateien zum GA-Knoten nicht explizit als aufgerufene Routinen dokumentiert.**

**5.2.2. Modulbeschreibungen**

Nachfolgend werden die Initialisierungs-Routinen detailliert beschrieben:

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.1       */
/*   Modulname :   UT allg. GAK-Routinen Stand   :   01.05.97   */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   UT_INIT ( argc, argv )   */
/*****
/*   Beschreibung :                               */
/*   Diese Routine führt die allg. Initialisierungen für GAK */
/*   nach dem Starten des Hauptprogrammes durch :          */
/*   - Auswerten und Vermerken der Aufrufparameter des Hauptprogr. */
/*   - Deaktivieren der Floating-Point-Error-Messages      */
/*   - Initialisieren der Ausgabe auf Protokoll-Drucker/-Datei */
/*   */
/*   Hinweis: Ab Version 1.1 erfolgt der Aufruf von UT_GA_CFG, */
/*           UT_OB_CFG und UT_DP_CFG direkt aus GAK_INIT, um eine */
/*           Verwendung der UT_LIB in anderen Systemen (z.B. I2X) */
/*           zu ermöglichen.                               */
/*   */
/*   aufgerufen von :           GAK_INIT           */
/*   ruft auf       :           UT_LOOPTIME, UT_WSL_INIT           */
/*   I/O-Kanäle    :                               */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name           Dim.   Beschreibung           */
/*   int      argc           */           Anzahl der Aufrufparameter */
/*   char*    argv[]         */           Zeiger auf Feld mit den */
/*                                           einzelnen Aufrufparameter */
/*   -----                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name           Dim.   Beschreibung           */
/*   int      UT_INIT        */           Returncode der Funktion */
/*                                           0=FALSE: KEIN Fehler */
/*                                           -ln: Fehler bei allg. INIT */
/*                                           >0: Zeilennummer aus der Konf.- */
/*                                           Datei, in der ein Fehler */
/*                                           erkannt wurde */
/*   -----                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name           Dim.   Beschreibung           */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                               */
/* Original :   W. Fries    01.05.97    1.1           */
/* Änderung:                               */
/*   -----                               */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.1       */
/*   Modulname :   UT allg. GAK-Routinen Stand   :   01.05.97   */
/*****
/*   Datei     :   GA_CFG.C              */
/*   Aufruf    :   UT_GA_CFG ( )        */
/*****
/*   Beschreibung :                               */
/*   Diese Routine lädt die allgemeinen GA-Parameter aus der */
/*   Konfigurationsdatei.                               */
/*   aufgerufen von :           GAK_INIT           */
/*   ruft auf       :           UT_NORM_STRING     */
/*   I/O-Kanäle    :                               */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*   -----                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----                               */
/*   int      UT_GA_CFG                               */
/*   Returnkode der Funktion                               */
/*   0=FALSE: KEIN Fehler                               */
/*   -1: OPEN-Fehler                                   */
/*   -2: READ-Fehler                                   */
/*   -3: WRITE-Fehler                                  */
/*   -4: ALLOC-Fehler                                  */
/*   -5: unzulässige EL-Zeile                           */
/*   -6: unzulässige DR-Zeile                           */
/*   -7: unzulässige ER-Zeile                           */
/*   -8: unzulässige ZA-Zeile                           */
/*   >0: Nummer der Zeile, in der                       */
/*   ein Fehler erkannt wurde                             */
/*   -----                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                               */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.1      */
/*      Modulname   :      UT allg. GAK-Routinen Stand   :   01.05.97  */
/*****
/*      Datei      :      OB_CFG.C              */
/*      Aufruf     :      UT_OB_CFG ( )        */
/*****
/*      Beschreibung :                          */
/*      Diese Routine lädt die Objekt-spezifischen Parameter aus der */
/*      Konfigurationsdatei.                          */
/*      aufgerufen von :      GAK_INIT          */
/*      ruft auf      :      SY_CALLOC, UT_NORM_STRING */
/*      I/O-Kanäle   :                          */
/*****
/*      Eingabe Parameter :                          */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung          */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                          */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung          */
/*      int      UT_OB_CFG          Returnkode der Funktion          */
/*                                          0=FALSE: KEIN Fehler          */
/*                                          -1: OPEN-Fehler          */
/*                                          -2: READ-Fehler          */
/*                                          -3: WRITE-Fehler          */
/*                                          -4: ALLOC-Fehler          */
/*                                          -5: unzulässige OB-Zeile          */
/*                                          >0: Nummer der Zeile, in der          */
/*                                          ein Fehler erkannt wurde          */
/*      ----- */
/*      Transiente Parameter :                          */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung          */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen          */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1          */
/*      Änderung:          */
/*      ----- */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.1       */
/*   Modulname :   UT allg. GAK-Routinen Stand   :   01.05.97    */
/*****
/*   Datei     :   DP_CFG.C              */
/*   Aufruf    :   UT_DP_CFG ( )        */
/*****
/*   Beschreibung :   */
/*   Diese Routine lädt die Datenpunkt-spezifischen Parameter aus der */
/*   Konfigurationsdatei.   */
/*   */
/*   aufgerufen von :   GAK_INIT        */
/*   */
/*   ruft auf       :   SY_CALLOC, UT_NORM_STRING, UT_DP_FINDFIRST */
/*   */
/*   I/O-Kanäle     :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   int      UT_DP_CFG          Returnkode der Funktion */
/*   */
/*   0=FALSE: KEIN Fehler */
/*   -1: OPEN-Fehler */
/*   -2: READ-Fehler */
/*   -3: WRITE-Fehler */
/*   -4: ALLOC-Fehler */
/*   -5: unzulässige DP-Zeile */
/*   -6: ES-Struktur fehlt */
/*   >0: Nummer der Zeile, in der */
/*   ein Fehler erkannt wurde */
/*   */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen  */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1         */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.1       */
/*   Modulname :   UT allg. GAK-Routinen Stand   :   01.05.97   */
/*****
/*   Datei    :   KL_CFG.C              */
/*   Aufruf   :   UT_KL_CFG ( )        */
/*****
/*   Beschreibung :                               */
/*   Diese Routine lädt die Kennlinien mit Stützstellen aus der */
/*   Konfigurationsdatei.                               */
/*   aufgerufen von :           GAK_INIT           */
/*   ruft auf      :           UT_NORM_STRING, SY_CALLOC       */
/*   I/O-Kanäle   :                               */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*   -----                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----                               */
/*   int      UT_KL_CFG                               */
/*   Returncode der Funktion                       */
/*   0=FALSE: KEIN Fehler                          */
/*   -1: FOPEN-Fehler                              */
/*   >0: Nummer der Zeile, in der                  */
/*   ein Fehler erkannt wurde                       */
/*   -----                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                               */
/*   Original :   W. Fries      01.05.97      1.1       */
/*   Änderung:                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.1       */
/*   Modulname :   UT allg. GAK-Routinen Stand   :   01.05.97   */
/*****
/*   Datei    :   VK_CFG.C               */
/*   Aufruf   :   UT_VK_CFG ( )         */
/*****
/*   Beschreibung :                       */
/*   Diese Routine lädt die Parameter für die Verkettung aus der */
/*   Konfigurationsdatei.                       */
/*   aufgerufen von :   GAK_INIT           */
/*   ruft auf      :   SY_CALLOC, UT_NORM_STRING */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung           */
/*   ----- */
/*   int      UT_VK_CFG           Returnkode der Funktion */
/*                                     0=FALSE: KEIN Fehler */
/*                                     -1: OPEN-Fehler */
/*                                     -2: READ-Fehler */
/*                                     -3: WRITE-Fehler */
/*                                     -4: ALLOC-Fehler */
/*                                     -5: unzulässige VK-Zeile */
/*                                     >0: Nummer der Zeile, in der */
/*                                     ein Fehler erkannt wurde */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:                                     */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### **5.3. Hauptprogramm des GA-Knoten (GAK)**

Das Programm GAK läuft normalerweise permanent ab, d.h. nachdem es erfolgreich gestartet wurde, werden die vorhandenen Bearbeitungsfunktionen im Polling-Verfahren zyklisch aufgerufen. Asynchrone Funktionen (z.B. Bedienung der seriellen Schnittstellen beim Empfangen und Senden von Daten) werden durch spezifische Routinen zur Unterbrechungs-Behandlung in Verbindung mit einer Zwischenpufferung realisiert. Auf diese Weise kann keine Funktion den Gesamt Ablauf dadurch behindern, dass erst auf eine Eingabe gewartet werden muß.

Das Programm unterstützt keine funktionellen interaktiven Bediener-Ein-/Ausgaben. Über die Tastatur läßt sich nur das Kommando STOP eingeben, das in Verbindung mit einer zusätzlichen expliziten Bestätigung (Eingabe von J auf die Frage "Wollen Sie das Programm wirklich beenden ? (J/N)") zu einem kontrollierten Programm-Stop führt. Ausgaben auf dem Bildschirm sind nur Fehlermeldungen, die zu einem Abbruch des Programmes führen (z.B. von der Eigenüberwachung erkannte Hardware-Fehler). Verwendet wird das "Standard-Output-Device" (stdout), der sich beim Programmaufruf auf eine Datei umlenken läßt (z.B. durch Angabe von ">GAK.LOG"). Auch die beim Programmstart aktivierten Trace-Meldungen lassen sich mit dem Schalter "/TD" auf eine Datei umlenken.

#### **5.3.1. Übersicht der verwendeten Module**

GAK	Hauptprogramm
ES	elektronischer Schlüsseleratz
IZ	Inselzentralen-Ebene (IZ-Treiber)
LZ	Leitzentralen-Ebene (LZ-Treiber; LZH, BWD-Gefahr, BWD-Einbruch)
SP	Schaltprogramme
CAPI	CAPI-Schnittstelle (Version 2.0)
IP	Internet-Protokoll
IPC	Inter-Process-Communication
X25	X.25-Schnittstelle (falls nicht deaktiviert)
UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

**5.3.2. Liste mit Namen und Aufgaben**

GAK	Hauptprogramm
GAK_INIT	Initialisierungen
GAK_LOOP	Zyklische Bearbeitung der Ein-/Ausgaben
GAK_IIZ	Bearbeitung der Eingaben von der IZ-Ebene
GAK_OIZ	Bearbeitung der Ausgaben an die IZ-Ebene
GAK_ILZ	Bearbeitung der Eingaben von der LZ-Ebene
GAK_ISP	Bearbeitung der Eingaben von den Schaltprogrammen
GAK_IES	Bearbeitung der Eingaben vom elektr. Schlüssellersatz
GAK_DPP_INIT	Initialisierung zur Bearbeitung des Karteipunktes für die Datenpunktparameter
GAK_DPP_CMD	Bearbeitung des Datagrammes zum Karteipunktes für die Datenpunktparameter
GAK_DPP_POLL	Polling zur Bearbeitung des Karteipunktes für die Datenpunktparameter
GAK_DPP_EXIT	Ende-Behandlung zur Bearbeitung des Karteipunktes für die Datenpunktparameter
GAK_FLIMMCHK	Flimmerkontrolle für Melde- und Schaltpunkte
GAK_ROLLMELD	Ausgabe von Meldungen am Statusbildschirm im Rollmodus
GAK_START_DP	Bearbeitung des Start/Stop-Datenpunktes
GAK_SYSCHK	Eigenüberwachung und Systemabgleich
GAK_ITAST	Behandlung von Tastatureingaben (STOP-Kommando)
GAK_EXIT	Ende-Behandlungen

**5.3.3. Aufrufstruktur**

GAK	
	GAK_INIT
	GAK_DPP_INIT
	IZ_INIT (vgl. Modul IZ)
	SP_INIT (vgl. Modul SP)
	LZ_INIT (vgl. Modul LZ)

GAK\_START\_DP

GAK\_LOOP

GAK\_IIZ

GAK\_FLIMMCHK

GAK\_ROLLMELD

IZ\_POLL (vgl. Modul IZ)

SP\_RSP\_USM (vgl. Modul SP)

LZ\_RSP\_USM (vgl. Modul LZ)

GAK\_ISP

SP\_POLL (vgl. Modul SP)

IZ\_CMD (vgl. Modul IZ)

GAK\_IES

ES\_POLL (vgl. Modul ES)

GAK\_ILZ

LZ\_POLL (vgl. Modul LZ)

IZ\_CMD (vgl. Modul IZ)

SP\_CMD (vgl. Modul SP)

LZ\_RSP\_USM (vgl. Modul LZ)

GAK\_DPP\_CMD

GAK\_START\_DP

GAK\_OIZ

IZ\_CMD (vgl. Modul IZ)

GAK\_DPP\_POLL

IZ\_CMD (vgl. Modul IZ)

GAK\_DPP\_INIT

GAK\_SYSCHK

IZ\_CMD (vgl. Modul IZ)

IZ\_SETTM (vgl. Modul IZ)

LZ\_ZA\_TIME (vgl. Modul LZ)

LZ\_STAT (vgl. Modul LZ)

GAK\_ITAST

GAK\_EXIT

GAK\_DPP\_EXIT

IZ\_EXIT (vgl. Modul IZ)

SP\_EXIT (vgl. Modul SP)

LZ\_EXIT (vgl. Modul LZ)

GAK\_START\_DP

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

**5.3.4. Modulbeschreibungen**

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand    :   01.10.98    */
/*****
/*   Datei    :   GAK.C                 */
/*   Aufruf   :   GAK [Schalter für Test-Funktionen] */
/*****
/*   Beschreibung :
/*   Dieses Hauptprogramm dient zur Realisierung des GA-Knotens
/*
/*   aufgerufen von :
/*
/*   ruft auf      :           GAK_INIT, GAK_LOOP, GAK_EXIT
/*                       BS_DF_INIT, BS_DF_OUTS
/*
/*   I/O-Kanäle   :           stdin, stdout, stderr
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.           Beschreibung
/*
/*   *** vgl. Schalter für Test-Funktionen in "GAK_TEST.H" ***
/*   *** und die Beschreibung zu deren Aufruf in der
/*   *** System-Spezifikation.
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.           Beschreibung
/*
/*   int      exit
/*                       EXIT-Kode des Hauptprogrammes
/*                       0: KEIN Fehler
/*                       >0: Zeilennummer in der Konf.-
/*                           in der ein Fehler auftrat
/*                       -1nn: ABBRUCH nach INIT
/*                       -2nn: ABBRUCH nach LOOP
/*                       -3nn: ABBRUCH nach EXIT
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.           Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*
/*   Autor      Datum      Version      Kurzzeichen
/*   -----
/*   Original :   W. Fries      01.05.97      1.1
/*   Änderung:   W. Fries      01.11.97      1.1a
/*   Änderung:   W. Fries      07.09.98      1.1d      WF2
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```



```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      GAK  GA-Knoten      Stand     :   01.10.98  */
/*****
/*      Datei      :      INIT.C                */
/*      Aufruf     :      GAK_INIT ( argc, argv ) */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die GA-Knoten-spezifischen Initialisierungen*
/*      nach dem Starten des Hauptprogrammes durch :
/*      - allg. UT-Initialisierungen mit Laden der zugehör. Strukturen */
/*      aus der Konfigurationsdatei und Ausgabe von Meldungen am
/*      Protokolldrucker
/*      - Laden der zu aktivierenden Datenpunkt-Parameter und des
/*      nächsten Aktivierungszeitpunkts
/*      - LZ-Initialisierungen mit Laden der zugehörigen Strukturen
/*      aus der Konfigurationsdatei bzw. den Verbindungsprofilen
/*      - IZ-Initialisierungen mit Laden der zugehörigen Strukturen
/*      aus der Konfigurationsdatei
/*      - Initialisierungen der Verkettungen
/*      - Initialisierungen der Schaltprogramm-Behandlung (SP-Routinen)
/*      - Initialisierungen des elektr. Schlüsselsystems (ES-Routinen)
/*      - Warteschlangen für die sofortige Ausgabe an LZH und BWD's
/*      - Aufbau des Status-Bildschirms
/*
/*      aufgerufen von :      GAK
/*
/*      ruft auf      :      GAK_DPP_INIT, GAK_EXIT, GAK_START_DP,
/*                          IZ_INIT, IZ_IZ_IZTEXT,
/*                          SP_INIT, SP_BS_INIT, ES_INIT, LZ_INIT,
/*                          UT_INIT, UT_GA_CFG, UT_OB_CFG, UT_DP_CFG,
/*                          UT_KL_CFG, UT_VK_CFG, UT_DR_OPROT,
/*                          UT_ER_GETER,
/*                          UT_DP_FINDFIRST, UT_DP_FINDNEXT,
/*                          BS_INIT, BS_DF_INIT, BS_DF_OUTS,
/*                          SY_CPU_HALT
/*
/*      I/O-Kanäle      :
/*
/*****
/*      Eingabe Parameter :
/*      -----
/*      Typ      Name      Dim.      Beschreibung
/*      *** KEINE ***
/*      -----
/*      Ausgabe Parameter:
/*      -----
/*      Typ      Name      Dim.      Beschreibung
/*      int      GAK_INIT      Returncode der Funktion
/*                          0=FALSE: KEIN Fehler
/*      -----
/*      Transiente Parameter :
/*      -----
/*      Typ      Name      Dim.      Beschreibung
/*      *** KEINE ***
/*****
/*      Autor      Datum      Version      Kurzzeichen
/*      -----
/*      Original :      W. Fries      01.05.97      1.1
/*      Änderung:      W. Fries      01.11.97      1.1a      WF1
/*
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand     :   01.10.98  */
/*****
/*   Datei    :   LOOP.C                */
/*   Aufruf   :   GAK_LOOP ( )         */
/*****
/*   Beschreibung :                   */
/*   Diese Routine führt die zyklische Bearbeitung der GAK- */
/*   Funktionen durch. Dabei werden nacheinander die folgenden */
/*   Bearbeitungsschritte zyklisch durchgeführt :           */
/*   - Behandlung von Eingabedaten von den DDC-IZ            */
/*   - Behandlung der Eingabedaten von den Schaltprogrammen */
/*   - Eingabe von Daten von der LZH, BWD-G oder BWD-E       */
/*   - Ausgabe von CMD-Datagrammen an die DDC-IZ's          */
/*   - Ausgabe von Meldungen auf den Protokolldrucker       */
/*   - Verwaltung zum Aktivieren der Datenpunkt-Parameter  */
/*   - Eigenüberwachung und Systemabgleich                  */
/*   - Behandlung von evtl. Tastatur-Eingaben               */
/*                                                         */
/*   aufgerufen von :   GAK                               */
/*                                                         */
/*   ruft auf          :   GAK_IIZ, GAK_ISP, GAK_IES, GAK_ILZ, GAK_OIZ, */
/*                       GAK_DPP_POLL, GAK_SYSCHEK, GAK_ITAST, */
/*                       UT_DR_OPROT, UT_LOOPTIME              */
/*                                                         */
/*   I/O-Kanäle       :                                   */
/*                                                         */
/*****
/*   Eingabe Parameter :                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*   -----                               */
/*   Ausgabe Parameter:                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----                               */
/*   int      GAK_LOOP      Dim.   Returnkode der Funktion */
/*                                     0=FALSE: KEIN Fehler  */
/*   -----                               */
/*   Transiente Parameter :               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*   -----                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen       */
/*   -----                               */
/*   Original :   W. Fries    01.05.97    1.1           */
/*   Änderung:                                     */
/*   -----                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      GAK GA-Knoten       Stand     :   01.10.98  */
/*****
/*      Datei       :      IIZ.C                */
/*      Aufruf      :      GAK_IIZ ( )         */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die Behandlungen der Eingaben von den */
/*      angeschlossenen DDC-IZ durch. Folgende Einzelfunktionen werden */
/*      durchgeführt :                        */
/*      - Polling der DDC-IZ's und Behandlung empfangener Datagramme */
/*      */
/*      aufgerufen von :      GAK_LOOP         */
/*      */
/*      ruft auf       :      GAK_FLIMMCHK, GAK_ROLLMELD, IZ_POLL, */
/*      SP_IZQ_CMD, SP_RSP_USM, ES_RSP_USM, */
/*      LZ_RSP_USM, */
/*      UT_DP_ATBTEXT, UT_DP_DIMTEXT, */
/*      UT_CFG_WRIT, UT_DRDPMELD, UT_UCGETBIT, */
/*      UT_DP_FINDFIRST, diverse Makros FND_... */
/*      SY_IEEE2F */
/*      */
/*      I/O-Kanäle    :                        */
/*      */
/*****
/*      Eingabe Parameter :                    */
/*      ----- */
/*      Typ      Name          Dim.    Beschreibung */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                    */
/*      ----- */
/*      Typ      Name          Dim.    Beschreibung */
/*      */
/*      int      GAK_IIZ          Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      -805: STOP durch FWD */
/*      ----- */
/*      Transiente Parameter :                */
/*      ----- */
/*      Typ      Name          Dim.    Beschreibung */
/*      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version    Kurzzeichen */
/*      ----- */
/*      Original :   W. Fries    01.05.97    1.1          */
/*      Änderung:   W. Fries    01.02.98    1.1b         WF1         */
/*      Änderung:   W. Fries    ???.??.98    1.1c         WF2         */
/*      Änderung:   W. Fries    07.09.98    1.1d         WF3         */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      GAK  GA-Knoten       Stand     :   01.10.98  */
/*****
/*      Datei      :      OIZ.C                */
/*      Aufruf     :      GAK_OIZ ( )         */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die Behandlungen der Ausgaben an die */
/*      angeschlossenen DDC-IZ durch. Folgende Einzelfunktionen werden */
/*      durchgeführt :                        */
/*      - Prüfung, ob zyklische Abfragen durchgeführt werden müssen */
/*      */
/*      aufgerufen von :      GAK_LOOP        */
/*      */
/*      ruft auf      :      IZ_CMD, UT_DT_ADD, UT_FNDNX2DT, */
/*      UT_FND_TIME, diverse Makros FND... und */
/*      SET_FND...      */
/*      */
/*      I/O-Kanäle   :                        */
/*      */
/*****
/*      Eingabe Parameter :                    */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      */
/*      ----- */
/*      Ausgabe Parameter:                    */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      int      GAK_OIZ      Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      */
/*      ----- */
/*      Transiente Parameter :                */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1 */
/*      Änderung: */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      GAK  GA-Knoten      Stand     :   01.10.98  */
/*****
/*      Datei       :      ILZ.C                */
/*      Aufruf      :      GAK_ILZ ( )         */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die Behandlungen der Eingaben von der LZH */
/*      und den BWD's durch. Folgende Einzelfunktionen werden */
/*      durchgeführt :                        */
/*      - LZ_SUBID = 1: Vermerken der Datenpunkt-Parameter bis zum */
/*      Aktivierungszeitpunkt (Weiterbearbeitung in GAK_DPP_POLL) und */
/*      Rückgabe des zugehörigen RSP-Datagrammes an das LZ-Modul. */
/*      - LZ_SUBID = 2: Weitergabe des CMD-Datagrammes an das SP-Modul */
/*      und Rückgabe des zugehörigen RSP-Datagrammes an das LZ-Modul. */
/*      - LZ_SUBID = 3: Weitergabe des CMD-Datagrammes an die DDC-IZ's */
/*      (direkter Zugriff durch die LZH) */
/*      - LZ_SUBID = 4: Weitergabe des CMD-Datagrammes an die DDC-IZ's */
/*      (direkter Zugriff durch den BWD-GEFAHR) */
/*      - LZ_SUBID = 5: Weitergabe des CMD-Datagrammes an die DDC-IZ's */
/*      (direkter Zugriff durch den BWD-EINBRUCH) */
/*      - LZ_SUBID = 6: Weitergabe des CMD-Datagrammes an das ES-Modul */
/*      und Rückgabe des zugehörigen RSP-Datagrammes an das LZ-Modul. */
/*      - LZ_SUBID >=10: Weitergabe des CMD-Datagrammes an die DDC-IZ's */
/*      (Istwert-Abfrage) */
/*      */
/*      aufgerufen von :      GAK_LOOP          */
/*      */
/*      ruft auf      :      GAK_DPP_CMD, GAK_START_DP, */
/*      IZ_CMD, SP_CMD, ES_CMD, LZ_RSP_USM */
/*      */
/*      I/O-Kanäle    :                        */
/*      */
/*****
/*      Eingabe Parameter :                    */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                    */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      int      GAK_ILZ          Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      -801: STOP durch LZH */
/*      -802: STOP durch BWD-GEFAHR */
/*      -803: STOP durch BWD-EINBRUCH */
/*      ----- */
/*      Transiente Parameter :                */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :   W. Fries   01.07.97     1.1      */
/*      Änderung:   W. Fries   01.11.97     1.1a     WF1      */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt   :      GA - Knoten           Version   :   1.21      */
/*      Modulname :      GAK  GA-Knoten       Stand     :   01.10.98  */
/*****
/*      Datei    :      IES.C                 */
/*      Aufruf   :      GAK_IES ( )          */
/*****
/*      Beschreibung :                       */
/*      Diese Routine führt die Behandlungen der Eingaben vom elektr. */
/*      Schlüsselssystem durch. Folgende Einzelfunktionen werden      */
/*      durchgeführt :                                               */
/*      - Weitergabe des CMD-Datagrammes an die DDC-IZ's             */
/*      aufgerufen von :      GAK_LOOP                                           */
/*      ruft auf      :      IZ_CMD, ES_POLL                                       */
/*      I/O-Kanäle   :                                                           */
/*****
/*      Eingabe Parameter :                                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung                    */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung                    */
/*      int      GAK_IES      Returncode der Funktion                    */
/*      0=FALSE: KEIN Fehler */
/*      ----- */
/*      Transiente Parameter :                                           */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung                    */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen                    */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1                    */
/*      Änderung:                                               */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand    :   01.10.98    */
/*****
/*   Datei     :   ISP.C                */
/*   Aufruf    :   GAK_ISP ( )         */
/*****
/*   Beschreibung :                   */
/*   Diese Routine führt die Behandlungen der Eingaben von den */
/*   Schaltprogrammen durch. Folgende Einzelfunktionen werden */
/*   durchgeführt :                   */
/*   - Weitergabe des CMD-Datagrammes an die DDC-IZ's          */
/*   aufgerufen von :           GAK_LOOP                          */
/*   ruft auf       :           IZ_CMD, SP_POLL                   */
/*   I/O-Kanäle     :                   */
/*****
/*   Eingabe Parameter :                   */
/*   -----                   */
/*   Typ      Name           Dim.   Beschreibung                   */
/*   -----                   */
/*   *** KEINE ***                   */
/*   -----                   */
/*   Ausgabe Parameter:                   */
/*   -----                   */
/*   Typ      Name           Dim.   Beschreibung                   */
/*   -----                   */
/*   int      GAK_ISP                Returnkode der Funktion    */
/*                                     0=FALSE: KEIN Fehler      */
/*   -----                   */
/*   Transiente Parameter :                   */
/*   -----                   */
/*   Typ      Name           Dim.   Beschreibung                   */
/*   -----                   */
/*   *** KEINE ***                   */
/*****
/*   Autor      Datum      Version   Kurzzeichen                   */
/*   -----                   */
/*   Original :   W. Fries      01.05.97      1.1                   */
/*   Änderung:                   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      GAK GA-Knoten       Stand     :   01.10.98  */
/*****
/*      Datei       :      DPP_CMD.C           */
/*      Aufruf      :      GAK_DPP_CMD ( cCMD ) */
/*****
/*      Beschreibung :      */
/*      Diese Routine vermerkt die in "cCMD" gelieferten Datenpunkt- */
/*      Parameter in einer zugehörigen Datei (Segmentierter Karteipunkt */
/*      gemäß LV unter 4.2 ff.). Ist der Aktivierungszeitpunkt früher */
/*      als der bisher als nächster vermerkte, so wird der neue */
/*      Zeitpunkt als nächster auszuführender vermerkt. */
/*      Als Ergebnis wird der Zeiger auf das für "cCMD" zu liefernde */
/*      RSP-Datagramm geliefert. */
/*      */
/*      Hinweis:      Das LV begrenzt den Zeilenindex auf 1.0, d.h. */
/*      pro Karteipunkt wird nur ein DPP-Satz übertragen. */
/*      Diese Routine wird bereits für laufenden Zeilen- */
/*      index vorbereitet, so dass zukünftig mehrere DPP- */
/*      Sätze für unterschiedliche Datenpunkte eines Typs */
/*      in einem Karteipunkt übertragen werden können, die */
/*      alle zum selben Zeitpunkt aktiviert werden. */
/*      (Notwendige Anpassungen in LZ_DPP_LOAD beachten !) */
/*      */
/*      aufgerufen von :      GAK_ILZ */
/*      ruft auf      :      UT_OBJ_GETI, UT_DP_FINDFIRST, UT_DT2CI, */
/*                       UT_FNDERROR, UT_FNDREJECT, UT_TAG_S2DT, */
/*                       diverse Makros FND_... und SET_FND_..., */
/*                       SY_IEEE2F */
/*      I/O-Kanäle    :      */
/*****
/*      Eingabe Parameter :      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      char*    cCMD      */      Zeiger auf das auszuwertende */
/*                       CMD-Datagramm mit den neuen */
/*                       Datenpunkt-Parametern */
/*      ----- */
/*      Ausgabe Parameter:      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      char*    GAK_DPP_CMD      */      Zeiger auf das RSP-Datagramm */
/*      ----- */
/*      Transiente Parameter :      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1      */
/*      Änderung:      */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      GAK  GA-Knoten       Stand     :   01.10.98  */
/*****
/*      Datei      :      DPP_EXIT.C           */
/*      Aufruf     :      GAK_DPP_EXIT ( )     */
/*****
/*      Beschreibung :                       */
/*      Diese Routine beendet die Verwaltung der Datenpunkt-Parameter */
/*      vor dem Stoppen des Hauptprogrammes durch :                       */
/*      aufgerufen von :      GAK_EXIT        */
/*      ruft auf      :                       */
/*      I/O-Kanäle   :                       */
/*****
/*      Eingabe Parameter :                   */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                   */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      int      GAK_DPP_EXIT      Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      ----- */
/*      Transiente Parameter :               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1 */
/*      Änderung: */
/*      ----- */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      GAK GA-Knoten       Stand     :   01.10.98  */
/*****
/*      Datei      :      DPP_INIT.C           */
/*      Aufruf     :      GAK_DPP_INIT ( )     */
/*****
/*      Beschreibung :                       */
/*      Diese Routine führt die Initialisierungen für die Verwaltung */
/*      der Datenpunkt-Parameter nach dem Starten des Hauptprogrammes */
/*      durch :                               */
/*      - Aktivieren der seit dem letzten Programmlauf zu aktivierenden */
/*      Datenpunkt-Parameter (inkl. Löschen der veralteten Dateien) */
/*      - Vermerken des nächsten Aktivierungszeitpunktes                */
/*      aufgerufen von : GAK_INIT                                           */
/*      ruft auf      : UT_CI2DT, SY_FI_FINDFIRST, SY_FI_FINDNEXT          */
/*      I/O-Kanäle   :                                                     */
/*****
/*      Eingabe Parameter :                                                 */
/*      -----                                                             */
/*      Typ      Name          Dim.    Beschreibung                       */
/*      *** KEINE ***                                                       */
/*      -----                                                             */
/*      Ausgabe Parameter:                                                 */
/*      -----                                                             */
/*      Typ      Name          Dim.    Beschreibung                       */
/*      int      GAK_DPP_INIT      Returnkode der Funktion              */
/*      0=FALSE: KEIN Fehler                                                */
/*      -1: OPEN-Fehler                                                      */
/*      -2: READ-Fehler                                                       */
/*      -3: WRITE-Fehler                                                       */
/*      -4: ALLOC-Fehler                                                       */
/*      -----                                                             */
/*      Transiente Parameter :                                             */
/*      -----                                                             */
/*      Typ      Name          Dim.    Beschreibung                       */
/*      *** KEINE ***                                                       */
/*****
/*      Autor      Datum      Version  Kurzzeichen                       */
/*      -----                                                             */
/*      Original :   W. Fries    01.05.97    1.1                          */
/*      Änderung:                                         */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand    :   01.10.98    */
/*****
/*   Datei     :   DPP_POLL.C           */
/*   Aufruf    :   GAK_DPP_POLL ( )    */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft, ob für einen Datenpunkt neue Parameter */
/*   aktiviert werden müssen. Falls JA werden diese geladen und in */
/*   der Konfigurations-Datei vermerkt. Anschließend werden die */
/*   neuen DPP zur zugehörigen DDC-IZ übertragen, falls sich in der */
/*   Datenpunkt-Tabelle DP_#n ein Parameter geändert hat. */
/*   Die veralteten Daten und Dateien werden gelöscht. Zusätzlich */
/*   wird der Zeitpunkt zum Aktivieren der nächsten Datenpunkt- */
/*   Parametern berechnet und vermerkt. */
/*   */
/*   aufgerufen von :   GAK_LOOP */
/*   */
/*   ruft auf      :   GAK_DPP_INIT, IZ_CMD, UT_CFG_WRIT, */
/*                   UT_DP_FINDFIRST, UT_DP_OBJID, */
/*                   diverse Makros FND_... und SET_FND_... */
/*                   SY_F2IEEE */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   */
/*   int      GAK_DPP_POLL          Returnkode der Funktion */
/*                   0=FALSE: KEIN Fehler */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1 */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand     :   01.10.98    */
/*****
/*   Datei     :   FLIMMCHK.C           */
/*   Aufruf    :   GAK_FLIMMCHK ( DP_ES12, uIndex )           */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft für die Ergänzungsstruktur zum Melde- bzw. */
/*   Schaltpunkt "DP_ES12", ob mit dem Zustandswechsel auf den neuen */
/*   "uIndex" ein Flimmern verbunden ist. Geprüft wird die min. */
/*   Dauer, die der vorherige Zustand angestanden sein muß und falls */
/*   dabei kein Flimmern erkannt wurde, ob die min. Periodendauer */
/*   eingehalten wurde. */
/*   */
/*   aufgerufen von :   GAK_IIZ           */
/*   */
/*   ruft auf       :   UT_UDATETAG, UT_UTIMESEK           */
/*   */
/*   I/O-Kanäle     :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----           */
/*   DP_ES12* DP_ES12      Zeiger auf die Ergänzungsstruktur. */
/*   uns.     uIndex      Neuer "actual", der einen */
/*   Zustandswechsel von GEHT nach */
/*   KOMMT bzw. umgekehrt darstellt */
/*   -----           */
/*   - - - - -           */
/*   Ausgabe Parameter:   */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----           */
/*   int      GAK_FLIMMCHK  Returnkode der Funktion */
/*   0=FALSE: KEIN Flimmerer */
/*   SONST: Flimmerer erkannt */
/*   -----           */
/*   - - - - -           */
/*   Transiente Parameter :   */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*   -----           */
/*****
/*   Autor      Datum      Version  Kurzzeichen   */
/*   -----           */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:   */
/*   -----           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand     :   01.10.98    */
/*****
/*   Datei     :   ROLLMELD.C           */
/*   Aufruf    :   GAK_ROLLMELD ( DP, cMeld ) */
/*****
/*   Beschreibung :
/*   Diese Routine gibt die Meldung "cMeld" für den Datenpunkt "DP"
/*   in den nächsten freien Zeile für die Rolling-Meldungen am
/*   Bildschirm aus, wenn der Statusbildschirm aktiviert ist.
/*   die Zeile zwischen der neuen und der ältesten Meldung wird
/*   gelöscht.
/*
/*   aufgerufen von :           GAK_IIZ
/*
/*   ruft auf      :           UT_BSDPMELD, BS_DF_INIT, BS_DF_OUTS
/*
/*   I/O-Kanäle   :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name           Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name           Dim.   Beschreibung
/*
/*   int      GAK_ROLLMELD
/*           Returnkode der Funktion
/*           0=FALSE: KEIN Fehler
/*
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name           Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*           Autor      Datum      Version   Kurzzeichen
/*   -----
/*   Original :   W. Fries      01.05.97      1.1
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand    :   01.10.98    */
/*****
/*   Datei    :   START_DP.C           */
/*   Aufruf   :   GAK_START_DP ( iModus, cAPDU )           */
/*****
/*   Beschreibung :                               */
/*   Diese Routine liefert abhängig von "iModus" für den Datenpunkt */
/*   "%GAKnnn_START" ein USM bzw. RSP-Datagramm in "cAPDU".       */
/*   */
/*   aufgerufen von :           GAK_INIT, GAK_EXIT, GAK_ILZ       */
/*   */
/*   ruft auf      :           UT_FND_TIME                       */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   -----
/*   Typ      Name           Dim.   Beschreibung           */
/*   int      iModus         */
/*           Modus für USM/RSP-Erzeugung           */
/*           0: CMD in cAPDU bearbeiten,           */
/*           d.h. RSP erzeugen                     */
/*           20: USM für START                     */
/*           21: USM für STOP                      */
/*   -----
/*   Ausgabe Parameter:                               */
/*   -----
/*   Typ      Name           Dim.   Beschreibung           */
/*   int      GAK_START_DP   */
/*           Returnkode der Funktion             */
/*           0=FALSE: KEIN Fehler                */
/*   -----
/*   Transiente Parameter :                               */
/*   -----
/*   Typ      Name           Dim.   Beschreibung           */
/*   *** KEINE ***
/*   -----
/*   Autor      Datum      Version   Kurzzeichen           */
/*   -----
/*   Original :   W. Fries     01.11.97   1.1a
/*   Änderung:
/*   -----
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt   :      GA - Knoten           Version   :   1.2      */
/*      Modulname :      GAK GA-Knoten        Stand     :   01.10.98  */
/*****
/*      Datei    :      SYSCHK.C              */
/*      Aufruf   :      GAK_SYSCHK ( )       */
/*****
/*      Beschreibung :                      */
/*      Diese Routine führt die Eigenüberwachung und den Abgleich aller */
/*      Systemkomponenten durch.          */
/*                                          */
/*      Mit jedem Aufruf dieser Routine wird die jeweils nächste      */
/*      Speicherzelle des Hauptspeichers getestet. Wird ein Fehler    */
/*      erkannt, wird das Programm beendet.          */
/*                                          */
/*      Ist das Ende des zu überwachenden Hauptspeicherbereiches (1 MB) */
/*      erreicht, so erfolgen die Funktionen zum Systemabgleich :    */
/*      - Ausgabe der aktuellen Datum/Uhrzeit am Bildschirm          */
/*      - Eigenüberwachung (Speichertest, verwendeten Festplatten-   */
/*      kapazitäten, (Res.: Temperatur-/Spannungsüberwachung, ...)  */
/*      - Ausgabe der aktuellen Status am Bildschirm und auf Prot.-Druck */
/*      - Abgleich der CMOS-Uhr (RTC) mit der LZ-Systemzeit o.ä.     */
/*      - DOS-Uhr mit der CMOS-Uhr abgleichen          */
/*                                          */
/*      aufgerufen von :          GAK_LOOP          */
/*                                          */
/*      ruft auf      :          IZ_CMD, IZ_SETTM, LZ_ZA_TIME, LZ_STAT, */
/*                      UT_ZA_GETZA, UT_SET_RTC, UT_RTC_DOS,          */
/*                      UT_OBJ_GETI, UT_DP_FINDFIRST,                 */
/*                      UT_DP_FINDNEXT, UT_UCGETBIT, UT_FND_TIME     */
/*                      diverse Makros FND... und SET_FND...,        */
/*                      BS_DF_INIT, BS_DF_OUTS,                      */
/*                      SY_FI_SIZE, SY_DISKFREE, SY_MEMCHK           */
/*                                          */
/*      I/O-Kanäle   :                      */
/*                                          */
/*****
/*      Eingabe Parameter :                      */
/*      -----                      */
/*      Typ      Name      Dim.      Beschreibung          */
/*      -----                      */
/*      *** KEINE ***                      */
/*      -----                      */
/*      Ausgabe Parameter:                      */
/*      -----                      */
/*      Typ      Name      Dim.      Beschreibung          */
/*      -----                      */
/*      int      GAK_SYSCHK          Returnkode der Funktion      */
/*                                          0=FALSE: KEIN Fehler      */
/*                                          -700: Temperatur-Fehler   */
/*                                          -701: Speicher-Fehler    */
/*      -----                      */
/*      Transiente Parameter :                      */
/*      -----                      */
/*      Typ      Name      Dim.      Beschreibung          */
/*      -----                      */
/*      *** KEINE ***                      */
/*      -----                      */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      -----                      */
/*      Original :      W. Fries      01.07.97      1.1          */
/*      Änderung:                      */
/*      -----                      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand    :   01.10.98    */
/*****
/*   Datei     :   ITAST.C              */
/*   Aufruf    :   GAK_ITAST ( )       */
/*****
/*   Beschreibung :                   */
/*   Diese Routine prüft, ob eine Tastatur-Eingabe erfolgte und */
/*   liest diese zeichenweise aus, um das eingelesene Zeichen   */
/*   weiterzuverarbeiten.                                       */
/*   In GAK ist nur die Eingabe der Abbruch-Sequenz STOP zulässig. */
/*   Es folgt die Abfrage, ob das Programm wirklich gestoppt werden */
/*   soll, diese ist ggf. mit J zu beantworten.                 */
/*   Die Eingabe der Zeichenfolge erfolgt ohne Abschlusszeichen (CR) */
/*   und lässt sich nicht editieren, d.h. kein DEL anwendbar.   */
/*   Jede Falscheingabe erzwingt die erneute Eingabe der gesamten */
/*   Sequenz.                                                    */
/*   aufgerufen von :   GAK_LOOP                                           */
/*   ruft auf         :   BS_DF_INIT, BS_DF_OUTS, UT_Z_JUMP              */
/*   I/O-Kanäle      :   stdin, stdout                                     */
/*****
/*   Eingabe Parameter :                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   *** KEINE ***                                                                */
/*   -----                               */
/*   Ausgabe Parameter:                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   int      GAK_ITAST      Returnkode der Funktion                         */
/*   0=FALSE: KEIN Fehler                                                         */
/*   -800: ABBRUCH durch den Bediener*/
/*   -----                               */
/*   Transiente Parameter :               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   *** KEINE ***                                                                */
/*****
/*   Autor      Datum      Version   Kurzzeichen                               */
/*   -----                               */
/*   Original :   W. Fries      01.05.97   1.1                               */
/*   Änderung:                                     */
/*   -----                               */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   GAK GA-Knoten        Stand     :   01.10.98    */
/*****
/*   Datei     :   EXIT.C               */
/*   Aufruf    :   GAK_EXIT ( )        */
/*****
/*   Beschreibung :                      */
/*   Diese Routine führt die GA-Knoten-spezifischen ENDE-Behandlungen*
/*   vor dem Beenden des Hauptprogrammes durch :
/*   - Eigenüberwachung (Watchdog, Speichertest, ...)
/*   - Beenden des elektr. Schlüsselsystems (ES-Routinen)
/*   - Beenden der Schaltprogramm-Behandlung (SP-Routinen)
/*   - spez. IZ-Ende-Behandlungen
/*   - spez. LZ-Ende-Behandlungen
/*   - Warteschlangen für die sofortige Ausgabe an LZH und BWD's
/*   - allg. UT-Ende-Behandlungen
/*   - Vermerken des GA-Knoten-Status GAK_STAT in Statusdatei
/*
/*   aufgerufen von :           GAK
/*
/*   ruft auf      :           GAK_DPP_EXIT, GAK_START_DP, GAK_ILZ,
/*                           IZ_EXIT, ES_EXIT, SP_EXIT, LZ_EXIT,
/*                           UT_EXIT, UT_DR_OPROT, UT_LOOPTIME
/*
/*   I/O-Kanäle   :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   int      GAK_EXIT      Returnkode der Funktion
/*                           0=FALSE: KEIN Fehler
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version   Kurzzeichen
/*   -----
/*   Original :   W. Fries   01.05.97   1.1
/*   Änderung:   W. Fries   01.11.97   1.1a   WF1
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

**5.3.5. Druckausgaben, Meldungstexte**

Fatale Fehler, die zum kontrollierten Beenden des Programmes führen, erzeugen eine entsprechende Fehlermeldung auf folgenden Ausgabekanälen, soweit diese verfügbar sind :

- als direkte Druckausgabe auf den Protokolldrucker, falls vorhanden
- Ausgabekanal für Trace-Meldungen, falls Schalter "/TD" beim Programmaufruf angegeben wurde.
- Standard-Error-Device (stderr), z.B. wie bei der Trace-Ausgabe ohne Angabe des Schalters "/TD"
- Standard-Output-Device (stdout), z.B. wie bei der Ausgabe aktueller Messwerte und Status. Dieser Ausgabekanal kann beim Programmaufruf mittels der Umlenkung durch ">" auf ein zusätzlich angeschlossenes Ausgabegerät oder eine Datei umgelenkt werden.

Der aktuelle Zustand des GA-Knoten wird im aktuellen Verzeichnis in der Datei **GAK.STA** vermerkt. Diese wird beim Programmstart geprüft und führt abhängig vom Zustand des letzten Programmlaufes im Falle eines für den weiteren Programmablauf kritischen Fehlers ggf. zu folgenden Meldungen:

- Protokolldrucker gestört
- ISDN-Zugriff gestört
- Watchdog- bzw. interner Fehler
- Speicher-Fehler

Der Watchdog- bzw. interne Fehler kennzeichnet einen unkontrollierten Programmabbruch (z.B. nach Reset oder Stromausfall). Eine hardwarebasierte Watchdog wird momentan nicht unterstützt.

Der Speicher-Fehler wird bei zyklischen Prüfung des konventionellen Hauptspeicherbereiches erkannt und ist der einzige für den weiteren Ablauf kritische Fehler und führt zu der Abfrage :

"Soll das Programm neu gestartet werden (J/N) >>"

Wird sie nicht innerhalb von 1 Minute mit "J" beantwortet, so versucht der GA-Knoten die CPU anzuhalten, da der weitere Programmablauf nicht gesichert ist.

**5.3.6. Datenkatalog**

Das Programm GAK bzw. ISSA oder LSSA liest und schreibt bei seinem Betrieb folgende Dateien , die für deren Konfiguration benötigt werden :

Folgende Dateien stehen im Verzeichnis \GAK\CFG bzw. SSA\CFG zur Verfügung :

- DP\_ATB.TXT
- DP\_DIM.TXT
- GAK.CFG bzw. ISSA.CFG oder LSSA.CFG
- DPP\_0000.nmn (nur GA-Knoten)
- ANRUFER.CHK

**5.3.6.1. DP ATB.TXT**

Die editierbare Text-Datei "DP\_ATB.TXT" enthält die Klartexte für die "atb\_id"-Kodierungen gemäß FND-Spezifikation im folgenden Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Code	dezimal kodierter Code gemäß FND-Spezifikation (vgl. Anhang E) mit führenden Nullen.
2	4	4	1	","	Trennzeichen (OHNE '"')
3	5	5	1	Index	Index für die Meldungsstufe (vgl. FND-Spezifikation zu "actual" des Meldepunktes).
4	6	7	3	"::"	Trennzeichen (OHNE '"')
5	9	128	<=119	Klartext	wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 128 Zeichen

**5.3.6.2. DP\_DIM.TXT**

Die editierbare Text-Datei "DP\_DIM.TXT" enthält die Klartexte für die "dimension"-Kodierungen gemäß FND-Spezifikation im folgenden Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Code	dezimal kodierter Code gemäß FND-Spezifikation (vgl. Anhang E) mit führenden Nullen.
2	4	5	3	"::"	Trennzeichen (OHNE '"')
3	7	128	<=121	Klartext	wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 128 Zeichen

**5.3.6.3. GAK.CFG**

Die editierbare Text-Datei "GAK.CFG" bzw. "ISSA.CFG" oder "LSSA.CFG" enthält die aktuellen Konfigurationsparameter des GA-Knotens bzw. IZ- oder LZ-SSA mit folgenden Zeilen-Kennungen :

Allgemeine Zeilen-Kennungen :

- ++ Fortsetzungszeile
- !Kommentarzeilen

Spezifische Parameter des GA-Knotens bzw. des IZ-/LZ-SSA :

- DR Zuordnung der Schnittstelle für den Protokolldrucker
- EL Zuordnung der Schnittstelle für das Error-Logging
- ER Eigene Rufnummer des GA-Knotens, wie sie seitens ISDN als "Called Party Number" geliefert wird.
- KK Kennung des GA-Knotens bzw. SSA
- IZ Vereinbarung eines IZ-Treibers und Zuordnung der zugehörigen Parameter (z.B. Schnittstelle)
- LZ Vereinbarung eines LZ-Treibers und Zuordnung der zugehörigen Parameter (z.B. Schnittstelle)
- KL Kennlinien zur Umrechnung binärer Werte in die phys. Werte

- SP Status-Profile für die Status-Zuordnung
- ZA Zeitabgleich über LZH-FND-Datagrammzeit oder interne DCF-77-Uhr
- ML Meldelinien zum BWD-GEFAHR bzw. BWD-EINBRUCH (nur GA-Knoten)

Objekt- und Datenpunkt-spezifische Parameter :

- DP Beginn eines Segmentes mit Vereinbarung der Datenpunkte für ein Objekt und einen IZ-Treiber
- ES Ergänzungsstruktur zum DP
- IS IZ-Struktur zum DP
- LS LZ-Struktur zum DP
- OB Vereinbarung eines Objektes
- VK Verkettung von Datenpunkten (nur GA-Knoten)
- SZ Zuordnung eines Status-Profiles

Soweit sinnvoll, lehnen sich die Formate für die einzelnen Parameter an die FND-Spezifikation an. Es gelten die folgenden Vereinbarungen :

- Alle Zeilen haben eine maximale Länge von 256 Zeichen (zzgl. <CR> (0x0D) und/oder <LF> (0x0A) ).
- Alle Zahlen stehen rechtsbündig in den entsprechenden Spalten und repräsentieren in Dezimaldarstellung die entsprechenden Bitkombinationen der FND-Variablen; Leerstellen sind mit Blanks (0x20) aufzufüllen.
- Die Datenpunkt-Adressen ("dp\_id") und die zugehörigen Bezeichnungen stehen linksbündig in der jeweiligen Spalte. Sie beginnen und enden mit einem alphanumerischen Zeichen. Nicht benutzte Stellen werden mit Blanks ausgefüllt.

**Hinweis:** Die abschließenden Blanks sind KEIN signifikanter Bestandteil der Datenpunkt-Adresse, d.h. diese werden gemäß FND-Spezifikation durch NULL-Bytes (0x00) ersetzt !!

Der Wertebereich für 3-stellige numerische Angabe ist, soweit nicht anders angegeben, auf  $0 \leq x \leq 255$  begrenzt. Für 5-stellige numerische Angaben gilt ein maximaler Wertebereich von  $0 \leq x \leq 32767$ .

Reelle Zahlen (14-stellig) können als ganze Zahlen bzw. in Gleitkomma- oder Exponential-Darstellung angegeben werden (inkl. Vorzeichen für Mantisse und Exponent). Dabei ist jedoch die Auflösung auf eine 7-stellige Mantisse und

der Wertebereich, soweit nicht anders angegeben, auf annähernd  $-3.4E+38 \leq x \leq +3.4E+38$  begrenzt. Ergänzend dazu werden "ungültige Werte", wenn sie gemäß FND-Spezifikation zulässig durch die Zeichenfolge "FFFFFFF" (ohne ") angegeben.

Werden Konfigurations-Parameter durch die LZH verändert (CMD-Datagramme), so werden diese Änderungen in dieser Datei vermerkt, d.h. nach einem Neustart des Programmes steht der geänderte Parameter wiederum zur Verfügung. Als Folge davon müssen Zeilen, die eine feste Länge haben (Länge der letzten Spalte explizit definiert; vgl. insbesondere ES-Zeilen), in der vollen Länge angegeben sein.

**Hinweis:** Einige Texteditoren entfernen Leerzeichen, die hinter dem letzten sichtbaren Zeichen angegeben sind. In diesen Fällen ist die geforderte rechtsbündige Ausrichtung unbedingt einzuhalten !!!!

Die ERSTE Zeile dieser Datei hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	5	5		Revisions-Nr.
2	7	16	10	Datum	Format: tt/mm/jjjj
3	18	22	5	Uhrzeit	Format: hh:mm
4	24	80	56	Projektname	
5	82	256	<=175	Autor	Name, Vorname, Firma, Anschrift, Telefon, Fax usw. in freiem Format  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Ab der ZWEITEN Zeile können die einzelnen Parameter in beliebiger Reihenfolge angegeben werden, da der GA-Knoten die einzelnen Parameter programmtechnisch in einer internen Reihenfolge einliest. Die Ausnahme bilden natürlich die Fortsetzungszeilen, da diese automatisch der zuvor definierten Parameterzeile zugeordnet werden.

Kommentarzeilen oder Leerzeilen (enthalten KEIN Zeichen > ' ') können an beliebiger Stelle eingefügt werden. Sie werden überlesen, d.h. der Benutzer trägt beim Editieren der Datei die Verantwortung dafür, dass die korrekte Syntax erhalten bleibt.

Die Verwendung von Tabulatoren ist NICHT zulässig.

Beschreibung der Parameter mit den allgemeinen Zeilen-Kennungen

KOMMENTAR-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	"!"	Zeilen-Kennung (OHNE '"')
2	2	256	<=255	Kommentartext	Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

FORTSETZUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"++"	Zeilen-Kennung (OHNE '"')
2	4	256	<=253	Fortsetzungstext	Der Aufbau ist abhängig vom Typ des zu definierenden Parameters, wie er zuvor definiert wurde  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die Fortsetzungszeile bezieht sich NICHT auf Kommentarzeilen, d.h. folgt nach einer Kommentarzeile eine Fortsetzungszeile, so wird diese der letzten zuvor definierten Parameterzeile zugeordnet.

Soweit Fortsetzungszeilen zulässig sind, wird es bei der Beschreibung der nachfolgenden Parameter explizit erwähnt.

Beschreibung der GA-Knoten-spezifischen Parameter :

DRUCKER-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"DR"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Schnittst.-Typ	0: KEIN Drucker verfügbar 1: parallel (z.B. LPTn) (lokale HW-Schnittstelle wird direkt angesprochen) 2: direkt seriell (z.B. COMn) (lokale HW-Schnittstelle wird direkt angesprochen) 10: SONSTIGES Gerät 20: Datei 255: NUR Warteschlange, OHNE Ausgabe. Nach Erreichen der max. Zeilenanzahl werden die ältesten 20% entfernt. Name der Datei ist DR_OPROT.WSL SONST: * reserviert *
3	8	10	3	Seiten-Parameter	Anzahl der Zeilen pro Seite 0: unbegrenzt, OHNE einen Seitenkopf und OHNE Seitenvorschub

					>0: Zeilenanzahl MIT Seitenkopf und Seitenvorschub
					255: unbegrenzt MIT Seitenkopf auf der ERSTEN Seite OHNE Seitenvorschub
4	12	16	5	WSL-Parameter Max.	Anzahl der in der für den Drucker-Warteschlange zu puffernden Zeilen 0: unbegrenzt
5	18	145	<=128	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von ihrem Typ. Für PC's unter DOS usw. wird hier nur die Ziffer für "n" angegeben, wenn die Schnittst. direkt angesprochen wird (vgl. Typ 1 und 2). Ansonsten wird der Name des Gerätes bzw. der Datei angegeben, unter dem es für den GA-Knoten verfügbar ist. (Für Typ 255 ist der Name fest DR_OPROT.WSL)
					Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Innerhalb der Konfigurations-Datei ist nur eine DRUCKER-Zeile zulässig.

Sind mehrere angegeben, so wird nur die ERSTE verwendet und die folgenden ignoriert.

**Hinweis:** Das Entfernen der ältesten 20% aller Zeilen aus der Warteschlange unterbricht den laufenden Betrieb, d.h. während dieser Phase findet KEINE zyklische Bearbeitung der anderen Programm-Module statt. Da der Aufwand proportional zur Anzahl der Zeilen ist, ist die Anzahl über den WSL-Parameter zu minimieren. Zusätzlich wird empfohlen, die Plattenzugriffe durch Verwendung eines Platten-Caches (z.B. SMARTDRV) zu beschleunigen.

Wird die Anzahl der Zeilen in der Warteschlange NICHT begrenzt, so nutzt der GA-Knoten die Plattenkapazität OHNE weitere Kontrolle voll aus. Um eine Störung des Betriebs durch das Entfernen einer sehr großen Datenmenge zu vermeiden, sollte dieser Modus nur zu Testzwecken unter regelmäßiger Kontrolle und manuellem Löschen verwendet werden.

ERROR\_LOG-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"EL"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Schnittst.-Typ	0: KEIN ERROR_LOG verfügbar 1: Ausgabe auf Protokoll-Drucker (vgl. DR-Zeile)



					255: NUR Warteschlange, OHNE Ausgabe. Nach Erreichen der max. Zeilenanzahl werden die ältesten 20% entfernt. Name der Datei ist die Schnittst.-Bez. SONST: * reserviert *
3	8	12	5	WSL-Parameter Max. Anzahl der in der für den ERROR_LOG-Warteschlange zu puffernden Zeilen 0: unbegrenzt	
4	14	27	<=12	Schnittst.-Bez.	Bezeichnung der Schnittstelle bzw. Datei.  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Innerhalb der Konfigurations-Datei ist nur eine ERROR\_LOG-Zeile zulässig. Sind mehrere angegeben, so wird nur die ERSTE verwendet und die folgenden ignoriert.

**Hinweis:** Der Hinweis zur DRUCKER-Zeile gilt analog.

RUFNUMMERN-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"ER"	Zeilen-Kennung (OHNE ''')
2	4	256	<=253	Rufnummer	Eigene Rufnummer, wie sie von ISDN als "Called Party Number" über die CAPI 2.0 geliefert wird, d.h. OHNE Vorwahl  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Innerhalb der Konfigurations-Datei ist nur eine RUFNUMMERN-Zeile zulässig. Sind mehrere angegeben, so wird nur die ERSTE verwendet und die folgenden ignoriert.

Ist KEINE RUFNUMMERN-Zeile angegeben, so wird die Kommunikation zur LZH und den BWD's deaktiviert, d.h. der GA-Knoten arbeitet nur lokal, d.h. folgende Funktionen werden deaktiviert :

- Zwischenspeicherung und spätere Übertragung von erfaßten Daten
- spontane Meldungen an die LZH und BWD's
- Annahme der Anrufe von der LZH und den BWD's

- Verwaltung und Bearbeitung der Verbindungs-Profile
- Verwaltung und Bearbeitung der Istwert-Abfragen

**Hinweis:** Wird z.B. für einen LZ-SSA ein spezifisches LZ-Modul eingebunden, so kann abhängig davon die RUFNUMMERN-Zeile entfallen und durch eine LZ-VEREINBARUNGS-Zeile ersetzt werden.

KENNUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"KK"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Kennung	Kennung des GA-Knotens bzw. SSA; num. Wert wie folgt : 0: KEINE Kennung definiert; zugehöriger Datenpunkt für die Fernwartung ist "%GAK_START" >0: Kennung; zugehöriger Datenpunkt für die Fernwartung ist "%GAKnnn_START" mit 001 <= nnn <= 127 (inkl. führenden Nullen)
3	8	47	<=40	Klartext	freier Klartext zur Beschreibung des GA-Knoten  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Der Klartext wird in der Kopfzeile des Statusbildschirms und dem Seitenkopf der Protokoll-Ausgabe dargestellt (vgl. DRUCKER-Zeile). Innerhalb der Konfigurations-Datei ist nur eine KENNUNGS-Zeile zulässig. Sind mehrere angegeben, so wird nur die ERSTE verwendet und die folgenden ignoriert. Ist keine Kennungs-Zeile angegeben, so wird der Klartext automatisch generiert aus der eigenen Rufnummer und dem Text "der Leitzentrale Haustechnik". Ist keine eigene Rufnummer angegeben, so wird der OBJEKT-Text (vgl. OBJEKT-Zeile) verwendet.

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen 1: FND über X.25/V.24 (ehemals) SMS/Fax/eMail über ISDN bzw. LAN 2: AEG-MODICON 3: FND über V.24/ISDN/LAN 4 ff.: Sondersysteme (vgl. unten)
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen DDC-IZ beginnend mit 0 für FND über ISDN, sonst stets 0.
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers

5	36	38	3	Verbindungs-Art	<p>natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20). Zusätzlich gilt : 40: UDP/IP über Netzwerk</p>
6	40	42	3	Protokoll	<p>nat. Zahl kodiert wie folgt : 0: FND gemäß Spez. 1.0 10: KS 120 (AEG MODICON)</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** FND: 0 *** AEG-MODICON: 10</p>
7	44	256	<=213	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und der Treiber-Id.</p> <p>IZ01 für FND über X.25/V.24: &gt;=0: EICON-X.25-Port (vgl. EC.CFG !!) Durch Leerzeichen getrennt ist noch die zu verwendende PVC-Nummer anzugeben (z.B. "2 1")</p> <p>-1: COM1 ... -4: COM4</p> <p>für SMS/Fax/eMail vgl. Sondersystem SMS</p> <p>IZ02: "COM1" - "COM4"</p> <p>IZ03: &gt;=0 Anzahl der ISDN-B-Kanäle, optional durch Leerzeichen und führendes '#' IP-Port (z.B. "0 #4405") V.24 (DOS): -1: COM1 ... -4: COM4</p> <p>V.24 (LINUX): /dev/ttyS0 ...</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

**Hinweis:** Für Sondersysteme können abweichende Festlegungen gelten. Beachten Sie bitte entsprechende Festlegungen unter "IZnn-Treiber für Sondersysteme".

Wird anstatt des LZ-Moduls für den ISDN-Zugriff ("Standard"-GA-Knoten) ein spezifischer LZ-Treiber eingebunden (z.B. beim LZ-SSA), so muß ggf. das verwendete spezifische Protokoll und die Schnittstelle definiert werden.

LZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"LZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den SSA und wird abhängig davon vorgegeben. 2: Sauter LZ für EY 2400  SONST: reserviert für zukünftige Erweiterungen
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen LZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !
4	12	34	23	LZ-/Treiber-Text	Klartext zur Beschreibung der LZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  Zusätzlich gilt : 40: UDP/IP über Netzwerk
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt : 40: 3964R / RK 512 (Sauter EY)  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !
7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem LZ-Typ.  Sauter: "COM1" - "COM4"  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

KENNLINIEN-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"KL"	Zeilen-Kennung (OHNE '"')
2	4	8	5	lfd. Nummer	Beginnend mit "0"
3	10	12	3	Typ der Stützst.	0: 16-Bit "unsigned" 1: 16-Bit "int", d.h. Bit 15 ist das Vorzeichen 2-255: * reserviert *
4	14	256	<=243	Stützstellen	Format "bW1:pW1" "bW2:pW2" ["bW3:pW3"["bW4:pW4" [...]]] (OHNE '"') mit

bWn: binärer Wert monoton  
steigend sortiert, d.h.  
bW(n) < bW(n+1)  
pWn: physikalischer Wert zum  
bWn

Wird durch das Zeilenabschluß-  
zeichen (<CR> und/oder <LF>,  
je nach verwendetem Editor)  
abgeschlossen.

Reicht die Zeilenlänge für die Definition aller Stützstellen einer Kennlinie nicht aus, so werden diese in einer oder mehreren Fortsetzungszeilen definiert.

STATUS-PROFIL-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"SP"	Zeilen-Kennung (OHNE '"')
2	4	8	5	lfd. Nummer	Beginnend mit "0"
3	10	10	1	Bit	1: niederwert.Bit (LSB,"fl") ... 8: höchstwert.Bit (MSB,"rl")
4	12	27	16	dp_id	Datenpunkt-Adresse (Quelle)
5	29	33	5	Flag f.Vergleich	Kode für Datenfeld (gemäß DV-technisches Feinkonzept des GA-Knoten, S.29)  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 11: actual
6	35	48	14	Vergleichswert	Bei erfolgreichem Vergleich auf akt.Wert >= Vergleichswert wird das Bit gesetzt.

**Hinweis: Bestehende Informationen in den zugeordneten Status-Bits des "info\_event" im Ziel-Datenpunkt werden überschrieben. Werden Status-Bits im "event"-Teil zugeordnet, so führt die Änderung des aktuellen Wertes im Quell-Datenpunkt nicht zur spontanen Meldung im Ziel-Datenpunkt (vgl. Zeilen-Kennung "SZ").**

ZEITABGLEICH-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"ZA"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Typ des Zeitabgl.	0: KEIN Zeitabgleich, da automatisch (z.B. DCF-77- Uhr mit TSR-Programm o.ä.) 1: Zeit aus LZ-Verbindungen 2: Zeit aus IZ-Verbindungen 255: Aufruf des Batches GA_GETTM.BAT (z.B. für

					spezifische Behandlung einer DCF-77-Uhr mit einem Hilfsprogramm) SONST: * reserviert *
3	8	10	3	Dimension	Dimension für Zeitabgleich gemäß FND-Ergänzungen (160ff.) (inkl. Zeitbasis für zykl. Wiederholungen)
4	12	25	14	Zeitpunkt	Zeitpunkt für Zeitabgleich bezogen auf die Dimension

Innerhalb der Konfigurations-Datei ist nur eine ZEITABGLEICHS-Zeile zulässig. Sind mehrere angegeben, so wird nur die ERSTE verwendet und die folgenden ignoriert.

MELDELINIEN-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"ML"	Zeilen-Kennung (OHNE '"')
2	4	4	1	BWD-Typ	Typ des zugehörigen BWD G: Gefahr E: Einbruch
3	6	10	5	ML-Nummer	Nummer der Meldelinie (beginnend mit "0" unabhängig von der Bezeichnung im Verbindungsprotokoll, d.h. "0" wird auf die ERSTE Meldelinie abgebildet )  *** Momentan begrenzt auf die *** Werte 0, ..., 15; *** bleibt reserviert für *** zukünftige Erweiterungen !
4	12	27	<=16	dp-id	Datenpunkt-Adresse  Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Der Zustand des Datenpunktes (Geber-Störung, BTA-Störung, Grenzwert-Verletzung (bereits ab der Warnung; NUR Mess- und Zählpunkte) bzw. aktuelle Meldungsstufe (NUR Melde- und Schaltpunkte) wird auf den Zustand der Meldelinie wie folgt abgebildet :

- 0 (Ruhezustand) : KEINES der Bits ist gesetzt
- 1 (Alarmzustand): Mindestens eines der Bits ist gesetzt

Es ist beachten, dass in der ES-Zeile für den angegebenen Datenpunkt das Flag für das Meldungsverhalten einen der BWD's zuweist. Umgekehrt darf nur für solche Datenpunkte das Flag für das Meldungsverhalten auf einen der BWD's verweisen, für den auch eine ML-Zeile angegeben ist.

Beschreibung der Objekt- und Datenpunkt-spezifischen Parameter :

DATENPUNKT-Zeilen haben den folgenden Aufbau (kompatibel zum FND-Referenz-File für die Datenpunkt-Liste gemäß FND-Spezifikation Teil 4 unter 7.4.2 bzw. 7.5.2) :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	5	5	lfd. Nr.	aufsteigende Reihenfolge, beginnend mit 1  *** Wird nicht ausgewertet *** und dient nur der *** Kompatibilität zur *** FND-Spezifikation
2	7	22	16	dp-id	Datenpunkt-Adresse
3	24	24	1	dp_type	Datenpunkt-Typ
4	26	26	1	dp_subtype	Datenpunkt-Subtyp
5	28	29	2	emq DP_#0	Modifikationskontrolle DP_#0
6	31	32	2	emq DP_#1	Modifikationskontrolle DP_#1
7	34	35	2	emq DP_#2	Modifikationskontrolle DP_#2
8	37	37	1	val	Valenz
9	39	39	1	msk	Ausblendung von Ereignismeld.
10	41	43	3	update_control	zusätzlich Mod.-Kontrolle
11	45	47	3	atb_id/dimension	Attribut-/Dimensions-Code
12	49	71	23	DP-Text	Bezeichnung des DP
13	73	74	2	Messnutzbereich	gültiger Wertebereich ( projektspez. codiert )  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** FND: unbegrenzt *** AEG-MODICON: unbegrenzt
14	76	80	5		* reserviert *

Ergänzend zur Syntax gemäß FND-Spezifikation Teil 4 wird eine DATENPUNKT-Zeile auch durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) nach dem "DP-Text" abgeschlossen, wodurch das Editieren der DATENPUNKT-Zeile erleichtert wird. Der Messnutzbereich wird dann unbegrenzt verwendet.

**Hinweis: Die im GA-Knoten integrierten IZ-Treiber unterstützen die Zugriffskontrolle für LZ-seitige Zugriffe gemäß FND-Spezifikation unter 2.8 bezogen auf "val" und die zusätzliche Modifikationkontrolle mittels "update\_control". Dabei ist jedoch zu beachten, dass die herstellerspezifischen Protokolle KEINE entsprechenden Verfahren zur Zugriffskontrolle bereitstellen. Die IZ-seitigen Zugriffe erfolgen also unabhängig und müssen ggf. projektspezifisch festgelegt werden. Die Angaben zu "emq DP\_#0", "emq DP\_1" und "emq\_DP#2" werden ignoriert.**

Für jeden Datenpunkt müssen eine DATENPUNKT-Zeile, sowie die zugehörigen ERGÄNZUNGSSTRUKTUR- und IZ-Struktur-Zeilen definiert werden. Die Position und Reihenfolge innerhalb der Konfigurations-Datei ist dabei jedoch ohne Bedeutung.

Ist für einen Grenzwert kein bzw. ein unzulässiger Wert angegeben, so wird dieser inaktiviert. Dieses gilt analog für Stell- und Zählpunkte, wobei "last\_save" mit 0.0 initialisiert wird.

ERGÄNZUNGSSTRUKTUR-Zeilen haben den folgenden Aufbau. Dabei sind sie abhängig DP-Typ ("dp\_type"), wobei diese der zugehörigen DATENPUNKT-Zeile entnommen wird :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"ES"	Zeilen-Kennung (OHNE ''')
2	4	19	16	dp_id	Adresse des zugehörigen DP
3	21	24	4	Objekt-Kennung	Kennung des zugehörigen Objektes
4	26	28	3	mask	Maskierungsbyte aus DP_#0
5	30	32	3	Meldungsverhalten	bitweise kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 10)  SONST: * reserviert *
6	34	36	3	Protokollierung	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 11).  SONST: * reserviert *
7 ff.	38	...		ESn	spezifischer Teil abhängig vom DP-Typ n ("dp_type")

Für n = 0 (Transferpunkt) gilt :

- mit "dp\_subtype" = 0 (Transferpunkt, transp. Datentransfer) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	42	5	TRF_Typ	Transfer-Typ, wie folgt kodiert: 0: IME-Anschluß an ES-Modul 1: P-IME-Anschluß an ES-Modul  SONST: * reserviert *

Für P-IME (TRF\_Typ = 1) gilt zusätzlich:



```
-----
```

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
8	44	59	16	IME_IM	Kennung (dp_id) der IME, an der nach der PIN-Eingabe das zugehörige IM eingelesen werden muss
9	61	65	5	IM-Wartezeit	Max. Wartezeit in Sekunden, innerhalb der das IM an der zugeordneten IME gelesen werden muss. Danach wird die eingegebene PIN ignoriert
10	67	71	5	PIN-Maske	Angabe der für die PIN signifikanten Zeichen in der von der P-IME gelieferten Zeichenfolge im Format "vv:bb" (z.B. "03:06": 1. und 2. Zeichen werden ignoriert, 3. bis 6. Zeichen sind PIN Sonst: Alle Zeichen signifikant

- sonst gilt :

```
-----
```

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	...		* LEER *	Die Zeile endet bei Spalte 36

Hinweis: Transferpunkte, mit Ausnahme des Fault-Managers, dienen ausschließlich der Kommunikation innerhalb des GA-Knotens und sind nicht an den IZ- oder LZ-Schnittstellen verfügbar !

Für n = 1 (Meldepunkt) gilt :

```
-----
```

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	40	3	Dim_GEHT	Zeitbezogene Dimension für "Min_GEHT" 12: Sekunde 13: Minute 14: Stunde 15: Tag
8	42	55	14	Min_GEHT	Minimale Anzahl von Zeiteinheiten aus "Dim_GEHT", die ein GEHT-Zustand aufrecht erhalten bleiben muß, um nicht als Flimmern interpretiert zu werden.
9	57	59	3	Dim_KOMMT	Zeitbezogene Dimension für "Min_KOMMT" 12: Sekunde 13: Minute 14: Stunde 15: Tag
10	61	74	14	Min_KOMMT	Minimale Anzahl von Zeiteinheiten aus "Dim_KOMMT", die ein KOMMT-Zustand aufrecht erhalten bleiben muß, um nicht als Flimmern interpretiert zu werden.
11	76	78	3	Dim_Periode	Zeitbezogene Dimension für "Min_Periode" 12: Sekunde 13: Minute 14: Stunde 15: Tag
12	80	93	14	Min_Periode	Minimale Anzahl von Zeiteinheiten aus "Dim_Periode", die eine Periode dauern muß, um nicht als Flimmern interpretiert zu werden.

zu werden.

13	95	99	5	Max_Flimmerer	Max. Anzahl in Folge erkannter Flimmerer, bevor der DP-Status "Geber gestört" intern und im aktuellen bzw. den folgenden USM gesetzt wird.
14	101	103	3	Flag_WB	Flag, ob weitere Bearbeitung dieses Datenpunktes erfolgt, wenn Flimmern erkannt wurde. 0: KEINE Bearbeitung, d.h. weitere Meldungen (USM) werden verhindert, indem alle Ereignismeldungen für den Datenpunkt ausgeblendet werden ("msk" = 3 ) SONST: Weitere Bearbeitung

Für n = 2 (Schaltpunkt) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	40	3	Dim_GEHT	Zeitbezogene Dimension für "Min_GEHT" 12: Sekunde 13: Minute 14: Stunde 15: Tag
8	42	55	14	Min_GEHT	Minimale Anzahl von Zeiteinheiten aus "Dim_GEHT", die ein GEHT-Zustand aufrecht erhalten bleiben muß, um nicht als Flimmern interpretiert zu werden.
9	57	59	3	Dim_KOMMT	Zeitbezogene Dimension für "Min_KOMMT" 12: Sekunde 13: Minute 14: Stunde 15: Tag
10	61	74	14	Min_KOMMT	Minimale Anzahl von Zeiteinheiten aus "Dim_KOMMT", die ein KOMMT-Zustand aufrecht erhalten bleiben muß, um nicht als Flimmern interpretiert zu werden.
11	76	78	3	Dim_Periode	Zeitbezogene Dimension für "Min_Periode" 12: Sekunde 13: Minute 14: Stunde 15: Tag
12	80	93	14	Min_Periode	Minimale Anzahl von Zeiteinheiten aus "Dim_Periode", die eine Periode dauern muß, um nicht als Flimmern interpretiert zu werden.
13	95	99	5	Max_Flimmerer	Max. Anzahl in Folge erkannter Flimmerer, bevor der DP-Status "Geber gestört" intern und im aktuellen bzw. den folgenden USM gesetzt wird.
14	101	103	3	Flag_WB	Flag, ob weitere Bearbeitung dieses Datenpunktes erfolgt, wenn Flimmern erkannt wurde. 0: KEINE Bearbeitung, d.h. weitere Meldungen (USM) werden verhindert, indem alle Ereignismeldungen für den Datenpunkt

ausgeblendet werden  
 ("msk" = 3 )  
 SONST: Weitere Bearbeitung

15      105      107      3      nominal      Vorgabe des Schaltwertes aus dem DP\_#1

Für n = 3 (Messpunkt) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	51	14	Zyklus	Zykluszeit in Minuten <=0: KEINE Zykluszeit def.
8	53	66	14	Hysterese	Hysterese, d.h. notwendige Änderung des "actual" bezogen auf den letzten gespeicherten actual-Wert <0: relative Änderung in [%] 0: KEINE Hysterese definiert >0: absolute Änderung in Einheiten von "dimension" in DP_#1
9	68	70	3	SyncCode	Zeitbezogene Dimension für "SyncZeit" (FND-Dimension 161 - 169 )
10	72	85	14	SyncZeit	Zeitpunkt, an dem generell eine Zwischenspeicherung erfolgen soll bezogen auf "SyncCode"
11	87	89	3	Vorverarb.-Art	Art der Vorverarbeitung 0: KEINE 1: MINIMUM-Bildung 2: MAXIMUM-Bildung 3: MITTELWERT-Bildung
12	91	104	14	Abfrage-Intervall	>0: Abfrageintervall in Minuten bezogen auf die letzte Abfrage bzw. den Programmstart 0: KEIN Abfrage-Int. def.
13	106	119	14	Skal-Koeff. a0	Skalierungs-Koeffizient a0
14	121	134	14	Skal-Koeff. a1	Skalierungs-Koeffizient a1
15	136	149	14	Skal-Koeff. a2	Skalierungs-Koeffizient a2
16	151	164	14	alarm_low	untere Alarm-Grenze
17	166	179	14	warning_low	untere Warn-Grenze
18	181	194	14	warning_high	obere Warn-Grenze
19	196	209	14	alarm_high	obere Alarm-Grenze dem DP_#1

Für n = 4 (Stellpunkt) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	51	14	Zyklus	Zykluszeit in Minuten <=0: KEINE Zykluszeit def.
8	53	66	14	Hysterese	Hysterese, d.h. notwendige Änderung des "actual" bezogen auf den letzten gespeicherten actual-Wert <0: relative Änderung in [%] 0: KEINE Hysterese definiert >0: absolute Änderung in Einheiten von "dimension" in DP_#1

9	68	70	3	SyncCode	Zeitbezogene Dimension für "SyncZeit" (FND-Dimension 161 - 169 )
10	72	87	14	SyncZeit	Zeitpunkt, an dem generell eine Zwischenspeicherung erfolgen soll bezogen auf "SyncCode"
11	87	89	3	Vorverarb.-Art	Art der Vorverarbeitung 0: KEINE 1: MINIMUM-Bildung 2: MAXIMUM-Bildung 3: MITTELWERT-Bildung
12	91	104	14	Abfrage-Intervall	>0: Abfrageintervall in Minuten bezogen auf die letzte Abfrage bzw. den Programmstart 0: KEIN Abfrage-Int. def.
13	106	119	14	Skal-Koeff. a0	Skalierungs-Koeffizient a0
14	121	134	14	Skal-Koeff. a1	Skalierungs-Koeffizient a1
15	136	149	14	Skal-Koeff. a2	Skalierungs-Koeffizient a2
16	151	164	14	nominal	Vorgabe für Stellwert

Für n = 5 (Zählpunkt) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	38	51	14	Zyklus	Zykluszeit in Minuten <=0: KEINE Zykluszeit def.
8	53	66	14	Hysterese	Hysterese, d.h. notwendige Änderung des "actual" bezogen auf den letzten gespeicherten actual-Wert <0: relative Änderung in [%] 0: KEINE Hysterese definiert >0: absolute Änderung in Einheiten von "dimension" in DP_#1
9	68	70	3	SyncCode	Zeitbezogene Dimension für "SyncZeit" (FND-Dimension 161 - 169 )
10	72	85	14	SyncZeit	Zeitpunkt, an dem generell eine Zwischenspeicherung erfolgen soll bezogen auf "SyncCode"
11	87	89	3	Vorverarb.-Art	Art der Vorverarbeitung 0: KEINE 1: MINIMUM-Bildung 2: MAXIMUM-Bildung 3: MITTELWERT-Bildung
12	91	104	14	Abfrage-Intervall	>0: Abfrageintervall in Minuten bezogen auf die letzte Abfrage bzw. den Programmstart 0: KEIN Abfrage-Int. def.
13	106	119	14	Skal-Koeff. a0	Skalierungs-Koeffizient a0
14	121	134	14	Skal-Koeff. a1	Skalierungs-Koeffizient a1
15	136	149	14	Skal-Koeff. a2	Skalierungs-Koeffizient a2
16	151	164	14	actual	akt. bzw. Rückssetzwert
17	166	179	14	last_save	Wert vor letzter Rücksetzung
18	181	194	14	warning_high	obere Warn-Grenze
19	196	209	14	alarm_high	obere Alarm-Grenze

**Hinweis:** Die Skalierungs-Koeffizienten a0 (Offset), a1 (lin. Faktor) und a2 (quadr. Faktor) sind so zu wählen, dass eine monotone steigende Funktion innerhalb des vom SSA gelieferten Wertebereichs gegeben ist.

Ist a2 angegeben, so MUSS folglich der Extremwert an der Stelle  $-a1/(2*a2)$  außerhalb oder auf einer der Grenzen dieses Wertebereiches liegen. Zusätzlich MUSS dann sichergestellt werden, dass die quadratische Gleichung bei der RE-Skalierung für das Stellen, (Rück)-Setzen, Parametrieren von Grenzwerten mind. 1 reelle Lösung liefert.

IZ-STRUKTUR-Zeilen haben den folgenden Aufbau. Dabei sind sie abhängig Typ/Protokoll der IZ, wobei dieser der zugehörigen IZ-VEREINBARUNGS-Zeilen entnommen wird :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IS"	Zeilen-Kennung (OHNE '"')
2	4	19	16	dp_id	Adresse des zugehörigen DP
3	21	23	3	IZ-Treiber-Id.	Id. des IZ-Treibers, d.h. der IZnn, auf der der Datenpunkt realisiert ist (vgl. IZ-VEREINBARUNGS-Zeile und IZ_MAX_TRB_ID in IZ_INCL.H)  240: interne Datenpunkte, zur Steuerung und Kommunikation des GA-Knoten, die programmtechnisch oder mittels Schaltprogr. modifiziert werden.
4	25	27	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.
5 ff.	29	...		IZn	spezifischer Teil abhängig vom Typ/Protokoll n der IZ

Für IZ-Treiber-Id. = 1 (SMS) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	40	12	SMS-Adresse	RUF-Datei mit gleichnamiger Extension für abgehende Rufe (z.B. H01P14.ruf mit "gaamm" gemäß GLT-Richtlinien, hier Pumpen-Störung)  RER-Datei mit gleichnamiger Extension für eingehende Rufe (Ruf-Ereignisse)  *** Wird für RER-Dateien *** noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets "*.rer"

Für IZ-Treiber-Id. = 2 (AEG MODICON mit KS 120) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

5	29	31	3	E/A_SYS	Operandentyp im D10 des KS 120-Telegrammes  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** 0x21: Merker-Byte (MB)
5	33	37	5	Adresse	Adresse des Operanden im D12 und D13 des KS 120-Telegrammes
6	39	43	5	Kennlinien-Nr.	Für Melde- und Schaltpunkt Nummer der Funktion, die zur Umrechnung der binären Werte verwendet werden soll wie folgt kodiert : 0: Keine Umrechnung 1: 1-stufige Invertierung 2: 2-stufige Invertierung  Für Mess-, Stell und Zählpunkt Nummer der Kennlinie, die zur Umrechnung der binären Werte in die phys. Werte angewendet werden soll (vgl. KL-Zeile).

Für IZ-Treiber-Id. = 3 (FND gemäß Spez. 1.0) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	...		* LEER *	Die Zeile endet bei Spalte 27 ( FND-IZ verwendet direkt die "dp_id", sodass eine IZ-spezifische Zuordnung entfällt )

Für IZ-Treiber-Id. = 4 (DOS) bzw. 4-16 (LINUX) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	...		herstellerspez.	Die Zeile ist herstellerspezifisch und wird beschrieben im Abschnitt 5.4.8 IZnn-Treiber für Sondersysteme

Für IZ-Treiber-Id. = 240 (interne Datenpunkte zur Steuerung des GA-Knoten) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	...		* LEER *	Die Zeile endet bei Spalte 27 ( GA-Knoten verwendet direkt die "dp_id", so dass eine IZ-spezifische Zuordnung entfällt )

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Wird anstatt des LZ-Moduls für den ISDN-Zugriff ("Standard"-GA-Knoten) ein spezifischer LZ-Treiber eingebunden (z.B. beim LZ-SSA), so muß der interne FND-Datenpunkt auf das spezifische Protokoll abgebildet werden.

LZ-STRUKTUR-Zeilen haben den folgenden Aufbau. Dabei sind sie abhängig Typ/Protokoll der LZ, wobei dieser der zugehörigen LZ-VEREINBARUNGS-Zeilen entnommen wird :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"LS"	Zeilen-Kennung (OHNE '"')
2	4	19	16	dp_id	Adresse des zugehörigen DP
3	21	23	3	LZ-Treiber-Id.	Id. des LZ-Treibers, d.h. der LZ, auf der der Datenpunkt realisiert ist (vgl. LZ-VEREINBARUNGS-Zeile)
4	25	27	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen LZ beginnend mit 0.
4 ff.	29	...		LZn	spezifischer Teil abhängig vom Typ/Protokoll der LZ

Für LZ-Treiber-Id. = 2 (Sauter LZ für EY 2400) gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
4	29	29	1	Befehlstyp	Befehlstyp gemäß 3964R  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** 'D': Datenbaustein
5	31	33	3	DB-Nr.	Nummer der Datenbausteins
6	35	37	3	DW-Nr.	Nummer der Datenwortes
7 ff.	39	...		DP_TYPEn	spezifischer Teil abhängig vom Datenpunkt-Typ

- Für Melde- und Schaltpunkte gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	39	41	3	Bit	Nummer des Bits in Datenwort im Bereich von 0, ..., 15

- Für Mess-, Stell- und Zählpunkte gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
7	39	43	5	Kennlinien-Nr.	Nummer der Kennlinie, die zur Umrechnung der binären Werte in die phys. Werte angewendet werden soll (Nur für Mess-,

Stell- und Zählpunkte, d.h.  
"dp\_type" 3, 4 und 5)

OBJEKT-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"OB"	Zeilen-Kennung (OHNE '"')
2	4	7	4	Objekt-Kennung	
3	9	48	40	Objekt-Text	Klartext zur Beschreibung des Objektes  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

VERKETTUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"VK"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Verkettungs-Typ	0: Duplizieren des Quell-DP (liefert internes RSP bzw. USM für den Ziel-DP auf Basis des RSP bzw. USM des Quell-DP). Dabei werden auch die Parameter und Modifikationen übernommen (z.B. Mittelwert <b>abhängig</b> vom Quell-DP).  1: Umlenkung des "actual" des Quell-DP in einen Schalt- bzw. Stellbefehl an den Ziel-DP  2: Ableitung des "actual" des Quell-DP in den Ziel-DP, d.h. wie das Duplizieren, jedoch eigene Protokollierung, Parametrierung usw. (z.B. Mittelwert <b>unabhängig</b> vom Quell-DP).  Sonst: reserviert
3	8	23	16	Quell-DP	Id. des Datenpunktes, der als Quelle der Verkettung dient (Auswertung der RSP und USM abhängig vom Verkettungs
4	25	40	16	Ziel-DP	Id. des Datenpunktes, der als Ziel der Verkettung dient. Für Umlenkung auch mit zeichen- bzw. abschließendem textsensitiven Platzhalter ('?' bzw. '*')  Ist ein If-DP angegeben, so erfolgt die Verkettung nur, wenn der Vergleich TRUE ergibt  Ist ein Else-DP angegeben, so ist diese Angabe optional



--- Nachfolgende Parameter sind optional und gelten derzeit nur für Umlenkungen (Bedingte Umlenkung) ---

5	42	46	5	Umrechnung	<p>Umrechnungsfunktion für den "actual" des Quell-DP vor Ausführung der Umlenkung</p> <p>Melde- und Schaltpunkte wie folgt kodiert:            0: KEINE            1: 1-stuf. Invertierung            2: 2-stuf. Invertierung</p> <p>Für Mess-,Stell-,Zählpkt.: Nummer der Kennlinie, die als Faktor, d.h Steigung bei 1, zur Umrechnung in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur)</p>
6	48	52	5	Wartezeit	<p>Anzahl der Sekunden, die vor Ausführung der Umlenkung gewartet werden soll</p> <p>Ist ein If-DP angegeben, so endet die Wartezeit, sobald der Vergleich TRUE ergibt - bei Ablauf der Wartezeit erfolgt die Umlenkung also i.a. auf den Else-DP, falls angegeben</p>
7	54	69	16	If-DP	<p>Id. des Datenpunktes, dessen "actual" mit dem If-Wert verglichen wird.</p> <p>Ist KEIN If-DP angegeben, so wird stets der Ziel-DP verwendet (implizites TRUE)</p>
8	71	84	14	If-Wert	<p>Vergleichswert für If-DP, falls dieser definiert. Vergleich liefert TRUE für Melde- und Schaltpkt.:            0: "actual" == 0 (GEHT)            2^n: "actual" &gt;= 2^n und "actual" != UNDEF mit (0 &lt;= n &lt;= 7; KOMMT)            für Mess-,Stell-,Zählpkt.: Wert:"actual" &gt;= Wert und "actual" != UNDEF</p>
9	86	101	16	Else-DP	<p>Id. des Datenpunktes, der als Ziel der Verkettung dient, wenn der Vergleich FALSE ergibt. Für Umlenkung auch mit zeichen- bzw. abschließendem textsensitiven Platzhalter ('?' bzw. '*')</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;), je nach verwendetem Editor) abgeschlossen.</p>

**Hinweis:** Je Quell-Datenpunkt darf nur eine Verkettung definiert werden. Ist eine Verkettung über mehrere Datenpunkte definiert, so erhält der Ziel-Datenpunkt die **FND-Datagramme aller vorgeschalteten Quell-Datenpunkte**. Werden für derart verkettete Ziel-Datenpunkte abweichende Arten der Vorverarbeitung definiert, so ist sicherzustellen, dass dabei keine Beeinträchtigung entsteht. So bezieht die Verkettungs-Folge

KEINE => MAX => MITTELWERT unzulässiger Weise den MAX-Wert in die Bildung des Mittelwertes mit ein. Korrekt ist die Verkettungs-Folge KEINE => MITTELWERT => MIN => MAX.

Die bedingte Umlenkung wird ausgeführt, wenn der If-Vergleich definiert ist und TRUE liefert oder die Wartezeit abgelaufen ist. Bei Ablauf der Wartezeit wird ggf. der Vergleich durchgeführt und die Umlenkung abhängig davon auf den Ziel- bzw. Else-DP ausgeführt. Ist der jeweilige DP nicht definiert, so wird die Umlenkung ignoriert.

STATUS-ZUORDNUNGS-zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"SZ"	Zeilen-Kennung (OHNE "'')
2	4	19	16	dp_id	Datenpunkt-Adresse (Ziel)
3	21	25	5	Nr. des SP	zugeordnetes Status-Profil (vgl. Zeilen-Kennung "SP")

**Hinweis: Bestehende Informationen in den zugeordneten Status-Bits des "info\_event" im Ziel-Datenpunkt werden überschrieben. Werden Status-Bits im "event"-Teil zugeordnet, so führt die Änderung des aktuellen Wertes im Quell-Datenpunkt (vgl. Zeilen-Kennung "SP") nicht zur spontanen Meldung im Ziel-Datenpunkt.**

Folgende Routinen-Aufrufe aus der Bibliothek UT\_LIB zum Basismodul UT stehen zum Lesen der Klartexte für Objekte, Datenpunkte, phys. Dimensionen und binäre Attribute aus den Dateien zur Verfügung :

- UT\_DP\_OBJTEXT
- UT\_DP\_DPTEXT
- UT\_DP\_DIMTEXT
- UT\_DP\_ATBTEXT

Die in den Dateien "DPP\_oooo.nnn" abgelegten Datenpunkt-Parameter sind die Vorgaben für die Steuerung der Datenpunkt-Bearbeitung durch den GA-Knoten. Es können mehrere Versionen mit unterschiedlichen Aktivierungszeitpunkten abgelegt sein.

Jede Datei enthält die Parameter für genau einen Datenpunkt zu einem Aktivierungszeitpunkt.

Die Text-Dateien und damit der Datenpunkt sind dem Objekt "oooo" zugeordnet mit der Versions-Kennung "nnn" im folgenden Format :

Die ERSTE Zeile hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	"001"	Kennung der ERSTEN Zeile
2	4	5	3	::"	Trennzeichen (OHNE '"')
3	7	20	14	Akt_Zeit	Aktivierungs-Zeitpunkt im Format JJJJMMTHHMMSS

Der Aufbau der ZWEITEN und DRITTEN Zeile entspricht dem der DATENPUNKT-und ERGÄNZUNGSSTRUKTUR-Zeilen (vgl. hierzu Datei GAK.CFG).

Die Extension "nnn" ist nicht zwangsläufig sortiert entsprechend der zeitlichen Reihenfolge der Aktivierung, sondern wird vom Programm eindeutig vergeben.

**Hinweis:** Im Verzeichnis dürfen nur ausschließlich die DPP-Dateien entsprechend der angegebenen Namenskonvention abgelegt werden, da das GAK-Programm alle vorhandenen Dateien prüft, deren Name mit "DPP" beginnt.

**Hinweis:** Entsprechend dem Format des Karteipunktes wird aus der Zeile 2 nur die Datenpunkt-Adresse ausgewertet. Die anderen Parameter in dieser Zeile können momentan leer sein, da keiner davon gemäß FND-Spezifikation gegenüber der Inbetriebnahme geändert werden darf bzw. eine Anpassung in der zugehörigen IZ impliziert.

Das verwendete Format ermöglicht spätere Erweiterungen und bleibt entsprechend dem Zeilenindex des zugehörigen Karteipunktes (vgl. Leistungsverzeichnis GA-Knoten) reserviert für zukünftige Erweiterungen (z.B. mehrere Datenpunkte zu einem Aktivierungszeitpunkt durch Wiederholung der Zeilen 2 und 3; zusätzliche Datenpunkt-Parameter im Karteipunkt wie z.B. die Bezeichnung des DP).

**5.3.6.4. ANRUFER.CHK**

Die Datei "ANRUFER.CHK" enthält die zulässigen Verbindungs-Adressen der LZH, den BWD's für GEFAHR und EINBRUCH und den über das ISDN angeschlossenen IZ's (vgl. IZ03), von denen der GA-Knoten angerufen werden darf. Zusätzlich werden auch die zulässigen Verbindungs-Adressen der Arbeitsplätze für die Fernwartungs-Dienste hinterlegt. Jeder Verbindungs-Adresse wird eine Kennung für das anrufende System (L, G, E, I oder W), die Verbindungs-Art und das zu verwendende Verbindungs-Protokoll zugeordnet. Jedem Eintrag für die LZH oder einem BWD wird zusätzlich die Priorität zugeordnet, über die gesteuert wird, ob eine ggf. aktive Verbindung abgebrochen werden soll.

Jede Zeile definiert eine zulässige Verbindungs-Adresse im folgenden Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	lfd. Nummer	beginnend mit 001 und führenden Nullen
2	4	5	3	"::"	Trennzeichen (OHNE '"')
3	7	7	1	Kennung	L: LZH, G: BWD-GEFAHR, E: BWD-EINBRUCH I: IZ W: Wartung
4	9	9	1	Zugriffrecht	als natürliche Zahl kodiert : 0: KEIN Zugriff 1: lesend (Abfragen) 2: schreibend (z.B. Schalten) 3: lesend und schreibend  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** LZH: lesend und schreibend *** BWD-GEFAHR: lesend und schreibend *** BWD-EINBRUCH: lesend und schreibend *** IZ: lesend und schreibend
5	11	13	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  Zusätzlich gilt : 40: UDP/IP über Netzwerk
6	15	17	3	Verbindungs-Prot.	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 21). Für die LZH ist "FND gemäß FND- Spezifikation mit FND- Erweiterungen" verbindlich festgelegt (d.h. 002).  *** Wird begrenzt ausgewertet *** und bleibt reserviert für

					<p>*** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt :</p> <p>*** LZH: 2=FND mit Erweiterungen</p> <p>*** BWD-GEFAHR: 10=TSS 13a                  oder : 1=FND 1.0</p> <p>*** BWD-EINBRUCH: 10=TSS 13a                  oder : 1=FND 1.0</p> <p>*** IZ: 1=FND</p> <p>*** Wartung: ohne Bedeutung                  (vgl. FWD-Programm)</p>
7	19	23	5	Verb.-Priorität	<p>Priorität des Anrufers für einen Verbindungsaufbau mit der LZH oder einem BWD. Eine aktive Verbindung wird nur dann unterbrochen, wenn deren Priorität kleiner als die hier angegebene Priorität ist. Ist der GA-Knoten MASTER einer bestehenden Verbindung, so gelten für diese abhängig vom Verbindungs-Partner folgende Prioritäten :</p> <ul style="list-style-type: none"> <li>- LZH: 10000</li> <li>- BWD-G: 20000</li> <li>- BWD-E: 30000</li> </ul>
8	25	49	25	Rufnummer	<p>"Calling Party Number" wie von ISDN geliefert, d.h. OHNE die führende Null in der Vorwahl und OHNE Trennzeichen wie ' ', '-', '/' usw.</p>
9	51	128	<=78	Sub-Adresse	<p>optional und abhängig von der Verbindungs-Art (z.B. X.25-DTE-Adresse). Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen, wenn nicht zusätzliche Parameter für einen IZ-Treiber folgen.</p> <p>Für Kennung "I" und den Treiber IZ03 (FND / CAPI / ISDN) besteht die Sub-Adresse aus folgenden Werten, die durch Leerzeichen getrennt angegeben werden :</p> <ul style="list-style-type: none"> <li>- IZ-Treiber-Id (Muß 3 sein !)</li> <li>- IZ-Treiber-SubId</li> <li>- Wartezeit in Sekunden nach einer Datenübertragung, nach der die ISDN-Verbindung abgebrochen wird.</li> </ul>

KOMMENTAR-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	"!"	Zeilen-Kennung (OHNE '"')
2	2	128	<=127	Kommentartext	Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 128 Zeichen.

### 5.3.6.5. GAK.STA

Der aktuelle Zustand des GA-Knoten wird im aktuellen Verzeichnis in der Datei **GAK.STA** vermerkt. Diese wird beim Programmstart geprüft und führt abhängig vom Zustand des letzten Programmlaufes zu den oben beschriebenen Meldungen.

### 5.3.6.6. DR\_OPROT.WSL

Die Protokoll-Ausgaben des GA-Knotens werden asynchron behandelt. Zu diesem Zweck werden auszugebende Zeilen in der folgenden Warteschlangen-Datei nach dem FIFO-Prinzip zwischengespeichert, bis der Drucker bereit ist:

**DR\_OPROT.WSL:** Ausgaben an das Protokoll-Gerät bzw. die Protokoll-Datei

### 5.3.6.7. Log-Dateien

Treten während des Programmlaufes Fehler (z.B. Kommunikationsfehler mit einer IZ oder Leitzentrale) auf, so werden diese in der über die Konfigurationsdatei GAK.CFG parametrisierten **Error-Log-Datei** (vgl. EL-Zeile) mit Zeitpunkt, Routinenname, 2 Fehlerkodierungen und einem optionalen Zusatztext protokolliert. Diese Fehlermeldungen dienen ausschließlich der Fehleranalyse in Verbindung mit den Programm-Quellen und werden somit nicht einzeln dokumentiert.

Empfängt der GA-Knoten von einer IZ ein NORMAL-, REJECT-, ERROR- oder unzulässiges FND-Datagramm, so werden diese bei gesetztem Schalter /LN, /LR, /LE bzw. /LI beim Programmstart in den folgenden Dateien binär protokolliert:

**NORMAL.LOG:** NORMAL-Datagramme

**REJECT.LOG:** REJECT-Datagramme

**ERROR.LOG:** ERROR-Datagramme

**INVALID.LOG:** unzulässige FND-Datagramme

**Hinweis:** Diese Dateien dienen ausschließlich Testzwecken und werden kontinuierlich fortgeschrieben, d.h. sie werden immer größer. Der Benutzer trägt die Verantwortung dafür, dass der Ablauf des GA-Knotens dadurch nicht beeinträchtigt wird, d.h. er muß sicherstellen, dass eine regelmäßige Überwachung der verfügbaren Festplatten-Kapazität und Löschung der LOG-Dateien durchgeführt wird.

#### 5.3.6.8. Kommando-Datei GAK.CMD

Findet der GA-Knoten während des Programmlaufs im Arbeitsverzeichnis die Datei **GAK.CMD**, so wird diese in **GAK\_ jjjmmthhmmss.CFG** umbenannt (mit "jjjmmthhmmss" Datum/Zeit der Umbenennung). Die darin enthaltenen Kommandos werden nacheinander abgearbeitet. Eine beim Programmstart vorhandene GAK.CMD wird ohne weitere Bearbeitung gelöscht.

Folgende Kommandos werden unterstützt :

- |                   |   |
|-------------------|---|
| - WARTE nn        | Warte "nn" Sekunden vor Bearbeitung des nächsten Kommandos  |
| - STOP            | wie Tastatur-Kommando STOP (vgl. unter 5.3)                 |
| - TRACE-DATEI xxx | wie /TDxxx (vgl. unter 5.1.4)                               |
| - TRACE-EIN       | Aktivierung Trace-Meldungen TI, TO, TA (vgl. unter 5.1.4)   |
| - TRACE-AUS       | Deaktivierung Trace-Meldungen TI, TO, TA (vgl. unter 5.1.4) |

#### 5.3.7. Definition der Datenübertragung

Der GA-Knoten stellt den angeschlossenen IZ's die Funktionen des FND-Protokolles in der Version 1.0 bereit. Zusätzlich unterstützt er eine Funktion zum Übergeben der Systemzeit an die IZ. Die im GA-Knoten realisierten Treiber (vgl. IZnLIB) für die IZ setzen diese Funktionen auf die system-spezifischen Schnittstellen und Protokolle um (Näheres vgl. im entsprechenden Abschnitt zum Modul IZ).

Der GA-Knoten kommuniziert alternativ über die ISDN- bzw. Netzwerk-Schnittstelle mit folgenden Leitzentralen:

LZH                      Leitzentrale Haustechnik

BWD_G	Bewachungsdienst (Gefahr)
BWD_E	Bewachungsdienst (Einbruch)

Zur LZH wird das FND-Protokoll inkl. der Ergänzungen (Karteipunkte) übertragen, während zu den BWD's das Protokoll TSS13a des VdS oder FND ohne Ergänzungen verwendet wird. (Näheres vgl. im entsprechenden Abschnitt zum Modul LZ). Innerhalb des GA-Knotens kommunizieren die einzelnen Module durch Übergabe von FND-Datagrammen als Aufrufparameter.

## **5.4. Inselzentralen (IZ und IZnn)**

**IZ LIB, IZnnLIB und IZ DTRB** In der Bibliothek **IZ\_LIB.LIB** werden alle Routinen zusammengefaßt, die mit der Kommunikation zwischen GA-Knoten und den Inselzentralen zusammenhängen. Die für die einzelnen IZ's spezifischen Programmteile (IZ-Treiber) werden in eigenen Unterverzeichnissen mit der Bibliothek **IZnnLIB.LIB** zusammengefaßt. (nn=01: SMS über ISDN; nn=02: KS120-Funktionen der AEG/Schneider Groupe Modicon direkt über V.24; nn=03: FND in der Version 1.0 über ISDN-/Netzwerk-Karte oder direkt über V.24). Vorbereitet werden die Anschlüsse für die Treiber 01 bis 04 (DOS) bzw. 16 (LINUX). Die nicht vorhandenen Treiber 04 ff. werden durch DUMMY-Treiber aus der Bibliothek **IZ\_DTRB.LIB** ersetzt.

### **5.4.1. Übersicht der verwendeten Module**

IZ	Inselzentralen-Ebene (FND 1.0, SPS)
CAPI	CAPI-Schnittstelle (Version 2.0)
IP	Internet-Protokoll
IPC	Inter-Process-Communication
X25	X.25-Schnittstelle (falls nicht deaktiviert)
UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek



### 5.4.2. Liste mit Namen und Aufgaben

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

IZ_INIT	Initialisierung beim Programmstart
IZ_CMD	Ausgabe eines CMD-Datagrammes an die IZ
IZ_SETTM	Ausgabe der Systemzeit an die IZ
IZ_POLL	Polling auf RSP- oder USM-Datagramm von der IZ an das Hauptprogramm
IZ_EXIT	Ende-Behandlung beim Programmstop

Folgende Routinen werden nur innerhalb dieses Modules verwendet:

IZ_IZ_CFG	Initialisierung der IZ-spezifischen Parameter beim Programmstart
IZ_IZ_IZTEXT	Klartext zur IZ aus der Konfigurationsdatei
IZ_IDP_INIT	Initialisierung der internen Datenpunkte beim Programmstart
IZ_IDP_CMD	Behandlung CMD-Datagrammes für internen Datenpunkt
IZ_IDP_POLL	Polling zur Behandlung der internen Datenpunkte
IZ_IDP_EXIT	Ende-Behandlung der internen Datenpunkte beim Programmstop

Als Aufruf-Schnittstelle für die IZ spezifischen Programmteile (Treiber) dienen folgende Routinen (mit "nn" = "01" bis "04" (DOS) bzw. "16" (LINUX)):

IZnn_INIT	Initialisierung beim Programmstart
IZnn_CMD	Ausgabe eines CMD-Datagrammes an die IZ
IZnn_SETTM	Ausgabe der Systemzeit an die IZ
IZnn_POLL	Polling auf RSP- oder USM-Datagramm von der IZ an das Hauptprogramm
IZnn_EXIT	Ende-Behandlung beim Programmstop

Spezifisch für die IZ "01" (SMS über ISDN) werden folgende Routinen aufgerufen, die die ISDN-Schnittstelle mittels CAPI 2.0 unterstützen:

IZ01_LIST_REQ	ISDN-Empfangsbereitschaft starten
IZ01_ALRT_REQ	ISDN-Verbindung reservieren
IZ01_CONN_REQ	ISDN-Verbindung aufbauen
IZ01_CONN_RES	ISDN-Anruf annehmen bzw. abweisen
IZ01_DATA_REQ	Datensatz als Paket im ISDN übertragen
IZ01_DISC_REQ	ISDN-Verbindung abbauen

Spezifisch für die IZ "02" (AEG/Schneider Groupe Modicon) werden folgende Routinen aufgerufen, die die V.24-Schnittstellen COMn (mit n = 1 - 4) unterstützen bzw. die Daten aus den Merkerbytes der SPS in ein FND-Datagramm konvertieren:

IZ02_COM_INIT	Initialisierung beim Programmstart
IZ02_COM_CIN	Ein Zeichen aus Eingabe-Puffer lesen, falls ex.
IZ02_COM_COUT	Ein Zeichen an Ausgabe-Puffer übergeben
IZ02_COM_PURG	Ausgabe-Puffer löschen
IZ02_COM_STAT	aktuellen Status abfragen
IZ02_COM_EVNT	aktuellen Status abfragen und Ereignisflags rücksetzen
IZ02_COM_KT80	Sonderbehandlung Polling für den Kommunikationsbetrieb (KT80)
IZ02_COM_EXIT	Ende-Behandlung beim Programmstop
IZ02_MB2FND	FND-Datagramm aus SPS-Merkerbytes erzeugen

Spezifisch für die IZ "03" (FND 1.0 über ISDN / V.24) werden folgende Routinen aufgerufen, die die ISDN- und V.24-Schnittstellen COMn (mit n = 1 - 4) unterstützen:

IZ03_LIST_REQ	ISDN-Empfangsbereitschaft starten
IZ03_CONN_REQ	ISDN-Verbindung aufbauen
IZ03_CONN_RES	ISDN-Anruf annehmen bzw. abweisen
IZ03_DATA_REQ	Datensatz als Paket im ISDN übertragen
IZ03_DISC_REQ	ISDN-Verbindung abbauen

IZ03_CAPIOLL	Polling der CAPI-Schnittstelle
IZ03_VB_SORT	Sortieren der ISDN-Verbindungen für die weitere Behandlung
IZ03_COM_INIT	Initialisierung beim Programmstart
IZ03_COM_CIN	Ein Zeichen aus Eingabe-Puffer lesen, falls es existiert
IZ03_COM_COUT	Ein Zeichen an Ausgabe-Puffer übergeben
IZ03_COM_PURG	Ausgabe-Puffer löschen
IZ03_COM_STAT	aktuellen Status abfragen
IZ03_COM_EVNT	aktuellen Status abfragen und Ereignisflags rücksetzen
IZ03_COM_SET_DTR	Setzen des DTR-Signal für die Modem-Control
IZ03_COM_EXIT	Ende-Behandlung beim Programmstop

### 5.4.3. Aufrufstruktur

IZ_INIT	
IZ_IZ_CFG	
IZ_IZ01_INIT	
IZ_IZ_IZTEXT	
CAPI_INIT	
IZ_IZ02_INIT	
IZ02_COM_INIT	
IZ02_COM_CIN	
IZ_IZ03_INIT	
IZ03_COM_INIT	
IZ03_COM_SET_DTR	
IZ03_COM_CIN	
IZ_IZ_IZTEXT	
CAPI_INIT	

IZ\_IZ04\_INIT

IZ\_IDP\_INIT

IZ\_CMD

IZ01\_CMD

IZ02\_CMD

IZ03\_CMD

IZ04\_CMD

IZ\_IDP\_CMD

IZ\_CMD

IZ01\_CMD

IZ02\_CMD

IZ03\_CMD

IZ04\_CMD

IZ\_IDP\_CMD

IZ\_SETTM

IZ01\_SETTM

IZ02\_SETTM

IZ03\_SETTM

IZ04\_SETTM

IZ\_POLL

IZ01\_POLL

IZ04\_SMS\_INIT

IZ04\_SMS\_CRC

IZ04\_SMS\_WRUF

IZ04\_CONN\_REQ

IZ04\_CONN\_RES

IZ04\_ALRT\_REQ

IZ04\_DATA\_REQ

IZ04\_DISC\_REQ

IZ02\_POLL

IZ02\_MB2FND

IZ02\_COM\_KT80

IZ02\_COM\_COUT

IZ02\_COM\_CIN

IZ02\_COM\_PURG

IZ02\_COM\_STAT

IZ02\_COM\_EVNT

IZ03\_POLL

IZ03\_COM\_SET\_DTR

IZ03\_COM\_STAT

IZ03\_COM\_EVNT

IZ03\_COM\_COUT

IZ03\_COM\_CIN

IZ03\_CONN\_REQ

IZ03\_DATA\_REQ

IZ03\_DISC\_REQ

IZ03\_CAPIOLL

CAPI\_ERROR

IZ04\_POLL

IZ\_IDP\_POLL

IZ\_EXIT

IZ01\_EXIT

CAPI\_EXIT

IZ02\_EXIT

IZ02\_COM\_STAT

IZ02\_COM\_EXIT

IZ03\_EXIT

IZ03\_COM\_SET\_DTR

IZ03\_COM\_STAT

IZ03\_COM\_EXIT

CAPI\_EXIT

IZ04\_EXIT

IZ\_IDP\_EXIT

**5.4.4. Modulbeschreibungen**

**5.4.4.1. Aufruf-Schnittstelle für das Hauptprogramm**

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   INIT.C                */
/*   Aufruf   :   IZ_INIT ( ullZ_STAT ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine führt die allg. Initialisierungen für die IZ- */
/*   Treiber nach dem Starten des Hauptprogrammes durch :       */
/*   - Laden der allgemeinen IZ-Parameter aus der Konf.-Datei und */
/*   Aufruf der spezifischen Initialisierungs-Routinen für die  */
/*   einzelnen IZ-Treiber                                       */
/*   */
/*   aufgerufen von :           GAK_INIT          */
/*   */
/*   ruft auf      :           IZ_IZ_CFG, IZnn_INIT,          */
/*                   IZ_IDP_INIT, IZ_CMD, UT_WSL_INIT,        */
/*                   UT_DP_FINDFIRST, UT_DP_FINDNEXT,         */
/*                   UT_FND_TIME, diverse Makros SET_FND_...  */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung          */
/*   uns. long ullZ_STAT          LZ_STAT nach der Initialisierung* */
/*                   (Kodierung vgl. in LZ_INCL.H)          */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung          */
/*   int      IZ_INIT          Returnkode der Funktion          */
/*                   0=FALSE: KEIN Fehler                    */
/*                   -1n: Fehler bei allg. INIT              */
/*                   -3n: Fehler bei IZ_CFG                   */
/*                   -4n: Fehler bei IZnn_INIT               */
/*                   -5n: Fehler bei IZ_IDP_INIT             */
/*                   >0: Zeilennummer aus der Konf.- */
/*                   Datei, in der ein Fehler          */
/*                   erkannt wurde                        */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung          */
/*   *** KEINE ***                                          */
/*****
/*   Autor      Datum      Version  Kurzzeichen  */
/*   ----- */
/* Original :   W. Fries    01.05.97    1.1          */
/* Änderung:                                         */
/*   ----- */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      IZ allg. IZ-Routinen  Stand    :   01.10.98 */
/*****
/*      Datei      :      CMD.C                  */
/*      Aufruf     :      IZ_CMD ( ucCMD, ucIZ_Id ) */
/*****
/*      Beschreibung :                          */
/*      Diese Routine gibt das CMD-Datagramm "ucCMD" an die DDC-IZ- */
/*      spezifische Routine IZnn_CMD weiter, Dabei ist "nn" = "ucIZ_ID". */
/*      Ist "ucIZ_Id" nicht definiert, so sucht die Routine den */
/*      zugehörigen Eintrag in der DP-Struktur und verwendet die dort */
/*      zugeordnete IZ_Id. */
/*      */
/*      aufgerufen von :      GAK_ILZ, GAK_ISP, GAK_OIZ, GAK_DPP_POLL */
/*      GAK_SYSCCHK, IZ_INIT */
/*      */
/*      ruft auf      :      UT_DP_FINDFIRST, UT_WSL_WRIT, */
/*      UT_FNDERROR, UT_FNDREJECT, */
/*      IZnn_CMD, IZ_IDP_CMD */
/*      */
/*      I/O-Kanäle    :                          */
/*      */
/*****
/*      Eingabe Parameter :                          */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      uns. char ucCMD      Puffer mit dem CMD-Datagramm */
/*      uns. char ucIZ_Id   IZ-Id der DDC-IZ, auf die das */
/*      CMD-Datagramm ausgegeben werden */
/*      soll. */
/*      "uc_UNDEF": IZ-ID nicht def., */
/*      d.h. in der DP-Struktur suchen */
/*      ----- */
/*      Ausgabe Parameter:                          */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      int      IZ_CMD      Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      -1: ucCMD unzulässig */
/*      -2: ucIZ_Id unzulässig */
/*      ----- */
/*      Transiente Parameter :                          */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1      */
/*      Änderung: */
/*      ----- */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   SETTM.C                */
/*   Aufruf   :   IZ_SETTM ( )          */
/*****
/*   Beschreibung :                       */
/*   Diese Routine startet den Zeitabgleich für alle DDC-IZ's */
/*   */
/*   aufgerufen von :           GAK_SYSCHK */
/*   */
/*   ruft auf      :           IZnn_SETTM  */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   */
/*   int      IZ_SETTM                Returnkode der Funktion */
/*   */
/*   0=FALSE: KEIN Fehler */
/*   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1 */
/*   Änderung: */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   POLL.C                */
/*   Aufruf   :   IZ_POLL ( )          */
/*****
/*   Beschreibung :                      */
/*   Diese Routine führt die zyklische Behandlungen der POLL-Routinen*
/*   für die einzelnen DDC-IZ's durch.  */
/*                                           */
/*   aufgerufen von :      GAK_IIZ      */
/*                                           */
/*   ruft auf      :      IZnn_POLL, IZ_IDP_POLL, UT_WSL_READ,
/*                   UT_DP_FINDFIRST, UT_FND_TIME, diverse
/*                   Makros FND_...
/*                                           */
/*   I/O-Kanäle   :                      */
/*                                           */
/*****
/*   Eingabe Parameter :                  */
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   *** KEINE ***
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   char *   IZ_POLL
/*                   Zeiger auf den RECV-Buffer,
/*                   der das von der DDC-IZ empf.
/*                   RSP- bzw. USM-Datagramm enthält
/*                   SONST: NULL-Zeiger
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   *** KEINE ***
/*   -----
/*   Autor      Datum      Version   Kurzzeichen
/*   -----
/*   Original :   W. Fries    01.05.97    1.1
/*   Änderung:   W. Fries    01.09.97    1.1    IME
/*   -----
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   EXIT.C                */
/*   Aufruf   :   IZ_EXIT ( )           */
/*****
/*   Beschreibung :                               */
/*   Diese Routine führt die allg. Ende-Behandlungen für die IZ- */
/*   Treiber vor dem Beenden des Hauptprogrammes durch :       */
/*   - Aufruf der spezifischen ENDE-Behandlungs-Routinen für die */
/*   einzelnen IZ-Treiber                                       */
/*   - Freigabe der ALLOCIerten Speicherbereiche               */
/*   */
/*   aufgerufen von :           GAK_EXIT                       */
/*   */
/*   ruft auf      :           IZnn_EXIT, UT_WSL_EXIT, SY_FREE  */
/*   */
/*   I/O-Kanäle   :                                           */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***                                           */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   int      IZ_EXIT                Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler                                           */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***                                           */
/*   ----- */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:                                     */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IZ_CFG.C               */
/*   Aufruf   :   IZ_IZ_CFG ( )         */
/*****
/*   Beschreibung :                       */
/*   Diese Routine lädt die allgemeinen IZ-Parameter aus der */
/*   Konfigurationsdatei.                               */
/*                                                     */
/*   aufgerufen von :           IZ_INIT           */
/*                                                     */
/*   ruft auf      :           SY_CALLOC          */
/*                                                     */
/*   I/O-Kanäle   :                       */
/*                                                     */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   int      IZ_IZ_CFG      Returnkode der Funktion       */
/*   0=FALSE: KEIN Fehler */
/*   -1: OPEN-Fehler      */
/*   -2: READ-Fehler      */
/*   -3: WRITE-Fehler     */
/*   -4: ALLOC-Fehler     */
/*   -5: unzulässige IZ-Zeile */
/*   >0: Nummer der Zeile, in der */
/*   ein Fehler erkannt wurde */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:                                     */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IZ_IZTEX.C           */
/*   Aufruf   :   IZ_IZ_IZTEXT ( iIZ_Index ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert als Ergebnis einen Zeiger auf den Klartext*
/*   für die IZ mit dem Index "iIZ_Index", falls ein solcher Text *
/*   existiert, sonst wird der Zeiger auf einen leeren Text geliefert*/
/*   */
/*   aufgerufen von :   GAK_INIT           */
/*   */
/*   ruft auf      :   UT_NORM_STRING      */
/*   */
/*   I/O-Kanäle   :   */
/*****
/*   Eingabe Parameter :   */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   int      iIZ_Index          Index des Eintrags in der */
/*   Index-Struktur, zu dem der Klartext* */
/*   geliefert werden soll.           */
/*   -----           */
/*   Ausgabe Parameter:   */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   char *   IZ_IZ_IZTEXT      Zeiger auf den Klartext */
/*   SONST: NULL-Zeiger           */
/*   -----           */
/*   Transiente Parameter :   */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   *** KEINE ***           */
/*****
/*   Autor      Datum      Version  Kurzzeichen   */
/*   -----           */
/*   Original :   W. Fries    01.05.97    1.1           */
/*   Änderung:           */
/*   -----           */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt   :      GA - Knoten           Version   :   1.2      */
/*      Modulname :      IZ allg. IZ-Routinen  Stand     :   01.10.98 */
/*****
/*      Datei     :      IDP_INIT.C           */
/*      Aufruf    :      IZ_IDP_INIT ( )     */
/*****
/*      Beschreibung :                       */
/*      Diese Routine führt die spezifischen Initialisierungen für die */
/*      Behandlung der internen Datenpunkte durch :                   */
/*      - Initialisieren der CMD-Warteschlange                         */
/*      - Laden der IZ-spezifischen IS-Parameter aus der Konf.-Datei und */
/*      spezifische Initialisierung                                   */
/*      */
/*      aufgerufen von :      IZ_INIT                                             */
/*      */
/*      ruft auf      :      UT_WSL_INIT, UT_NORM_STRING,                       */
/*      UT_DP_FINDFIRST                                         */
/*      */
/*      I/O-Kanäle    :      */
/*      */
/*****
/*      Eingabe Parameter :                       */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      */
/*      ----- */
/*      Ausgabe Parameter:                       */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      int      IZ_IDP_INIT      Returnkode der Funktion */
/*      */
/*      0=FALSE: KEIN Fehler */
/*      -1: OPEN-Fehler */
/*      -2: READ-Fehler */
/*      -3: WRITE-Fehler */
/*      -4: ALLOC-Fehler */
/*      -5: unzulässige IS-Zeile */
/*      -6: DDC-IZ-Init-Fehler */
/*      -7: WSL_INIT-Fehler */
/*      -8: unzulässige IZ_TRB_ID */
/*      >0: Nummer der Zeile, in der */
/*      ein Fehler erkannt wurde */
/*      */
/*      ----- */
/*      Transiente Parameter :                       */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1 */
/*      Änderung:      */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IDP_CMD.C             */
/*   Aufruf   :   IZ_IDP_CMD ( ucCMD )  */
/*****
/*   Beschreibung :                       */
/*   Diese Routine trägt das CMD-Datagramm "ucCMD" in die Warte- */
/*   schlange für CMD-Datagramme zur Behandlung der internen DP's ein*/
/*   */
/*   aufgerufen von :           IZ_CMD           */
/*   */
/*   ruft auf      :           UT_WSL_WRIT      */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***                               */
/*   */
/*   -----                               */
/*   Ausgabe Parameter:                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   int      IZ_IDP_CMD      Returnkode der Funktion      */
/*   */
/*   0=FALSE: KEIN Fehler                       */
/*   -1: OPEN-Fehler                             */
/*   -2: READ-Fehler                             */
/*   -3: WRITE-Fehler                            */
/*   */
/*   -----                               */
/*   Transiente Parameter :               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***                               */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                               */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:                                     */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IDP_POLL.C           */
/*   Aufruf   :   IZ_IDP_POLL ( )     */
/*****
/*   Beschreibung :                   */
/*   Diese Routine führt die zyklische Behandlung der internen */
/*   Datenpunkte (IDP) durch. Dabei wird die CMD-Warteschlange */
/*   ausgelesen und die Funktionen bearbeitet.                   */
/*   aufgerufen von :           IZ_POLL */
/*   ruft auf      :           UT_WSL_READ, UT_DP_FINDFIRST,    */
/*                   UT_FNDERROR, UT_FNDREJECT, diverse       */
/*                   Makros FND... und SET_FND...,            */
/*                   SY_F2IEEE                                 */
/*   I/O-Kanäle   :                   */
/*****
/*   Eingabe Parameter :                   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                   */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                   */
/*   ----- */
/*   char *  IZ_IDP_POLL          Zeiger auf das empfangenen FND- */
/*                                     Datagramm durch           */
/*                                     SONST: NULL-Zeiger          */
/*   ----- */
/*   Transiente Parameter :                   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                   */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1      */
/*   Änderung:   W. Fries    07.09.98    1.1d     WF1  */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IDP_EXIT.C             */
/*   Aufruf   :   IZ_IDP_EXIT ( )       */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*   internen Datenpunkte vor dem Stoppen des Hauptprogrammes durch :*/
/*   - Schließen der Warteschlange für CMD-Datagramme                */
/*                                                                    */
/*   aufgerufen von :           IZ_EXIT                                     */
/*                                                                    */
/*   ruft auf      :           UT_WSL_EXIT                               */
/*                                                                    */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   *** KEINE ***
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   int      IZ_IDP_EXIT          Returnkode der Funktion
/*                                     0=FALSE: KEIN Fehler
/*                                     -1: OPEN-Fehler
/*                                     -2: READ-Fehler
/*                                     -3: WRITE-Fehler
/*                                     -4: FREE-Fehler
/*                                     -5: unzulässige IS-Zeile
/*                                     -6: DDC-IZ-EXIT-Fehler
/*                                     -7: WSL_EXIT-Fehler
/*                                     -8: unzulässige IZ_TRB_ID
/*                                     >0: Nummer der Zeile, in der
/*                                     ein Fehler erkannt wurde
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   *** KEINE ***
/*****
/*   Autor      Datum      Version  Kurzzeichen
/*   -----
/*   Original :   W. Fries   01.05.97   1.1
/*   Änderung:
/*   -----
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IZ01DTRB.C           */
/*   Aufruf   :   IZ01_INIT ( ullZ_STAT ) */
/*   Aufruf   :   IZ01_EXIT ( )         */
/*   Aufruf   :   IZ01_CMD ( ucCMD )    */
/*   Aufruf   :   IZ01_POLL ( )        */
/*   Aufruf   :   IZ01_SETTM ( )       */
/*****
/*   Beschreibung :   */
/*   Diese Routine ist der DUMMY-Treiber für die IZ mit der ID 01 */
/*   und dient ausschließlich zum Erfüllen der Aufrufe aus anderen */
/*   Routinen. */
/*   Diese Datei enthält ALLE Routinen-Aufrufe für den Treiber und */
/*   wird in der Bibliothek IZ_DTRB.LIB mit den anderen DUMMY- */
/*   Treibern zusammengefaßt und darf erst nach den zu verwendenden */
/*   IZ-Treibern in das Programm eingebunden werden, so dass die */
/*   DUMMY-Routinen nur eingebunden werden, wenn kein IZ-Treiber */
/*   mit der entsprechenden IZ-Id. verwendet werden soll. */
/*   */
/*   aufgerufen von :   IZ_... */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   int      IZ01_INIT      Returnkode der Funktion */
/*   int      IZ01_EXIT      0=FALSE: KEIN Fehler */
/*   int      IZ01_CMD       -8: unzulässige IZ_TRB_ID, d.h. */
/*   int      IZ01_SETTM     DDC-IZ nicht verfügbar */
/*   */
/*   char*    IZ01_POLL      NULL-Zeiger */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1 */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IZ02DTRB.C           */
/*   Aufruf   :   IZ02_INIT ( ullZ_STAT ) */
/*   Aufruf   :   IZ02_EXIT ( )         */
/*   Aufruf   :   IZ02_CMD ( ucCMD )    */
/*   Aufruf   :   IZ02_POLL ( )        */
/*   Aufruf   :   IZ02_SETTM ( )       */
/*****
/*   Beschreibung :   */
/*   Diese Routine ist der DUMMY-Treiber für die IZ mit der ID 02 */
/*   und dient ausschließlich zum Erfüllen der Aufrufe aus anderen */
/*   Routinen. */
/*   Diese Datei enthält ALLE Routinen-Aufrufe für den Treiber und */
/*   wird in der Bibliothek IZ_DTRB.LIB mit den anderen DUMMY- */
/*   Treibern zusammengefaßt und darf erst nach den zu verwendenden */
/*   IZ-Treibern in das Programm eingebunden werden, so dass die */
/*   DUMMY-Routinen nur eingebunden werden, wenn kein IZ-Treiber */
/*   mit der entsprechenden IZ-Id. verwendet werden soll. */
/*   */
/*   aufgerufen von :   IZ_... */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   int      IZ02_INIT      Returnkode der Funktion */
/*   int      IZ02_EXIT      0=FALSE: KEIN Fehler */
/*   int      IZ02_CMD       -8: unzulässige IZ_TRB_ID, d.h. */
/*   int      IZ02_SETTM     DDC-IZ nicht verfügbar */
/*   */
/*   char*    IZ02_POLL      NULL-Zeiger */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1 */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IZ03DTRB.C           */
/*   Aufruf   :   IZ03_INIT ( ullZ_STAT ) */
/*   Aufruf   :   IZ03_EXIT ( )         */
/*   Aufruf   :   IZ03_CMD ( ucCMD )    */
/*   Aufruf   :   IZ03_POLL ( )        */
/*   Aufruf   :   IZ03_SETTM ( )       */
/*****
/*   Beschreibung :   */
/*   Diese Routine ist der DUMMY-Treiber für die IZ mit der ID 03 */
/*   und dient ausschließlich zum Erfüllen der Aufrufe aus anderen */
/*   Routinen. */
/*   Diese Datei enthält ALLE Routinen-Aufrufe für den Treiber und */
/*   wird in der Bibliothek IZ_DTRB.LIB mit den anderen DUMMY- */
/*   Treibern zusammengefaßt und darf erst nach den zu verwendenden */
/*   IZ-Treibern in das Programm eingebunden werden, so dass die */
/*   DUMMY-Routinen nur eingebunden werden, wenn kein IZ-Treiber */
/*   mit der entsprechenden IZ-Id. verwendet werden soll. */
/*   */
/*   aufgerufen von :   IZ_... */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   int      IZ03_INIT      Returnkode der Funktion */
/*   int      IZ03_EXIT      0=FALSE: KEIN Fehler */
/*   int      IZ03_CMD       -8: unzulässige IZ_TRB_ID, d.h. */
/*   int      IZ03_SETTM     DDC-IZ nicht verfügbar */
/*   */
/*   char*    IZ03_POLL      NULL-Zeiger */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1 */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ allg. IZ-Routinen  Stand    :   01.10.98  */
/*****
/*   Datei    :   IZ04DTRB.C           */
/*   Aufruf   :   IZ04_INIT ( ullZ_STAT ) */
/*   Aufruf   :   IZ04_EXIT ( )         */
/*   Aufruf   :   IZ04_CMD ( ucCMD )    */
/*   Aufruf   :   IZ04_POLL ( )        */
/*   Aufruf   :   IZ04_SETTM ( )       */
/*****
/*   Beschreibung :   */
/*   Diese Routine ist der DUMMY-Treiber für die IZ mit der ID 04 */
/*   und dient ausschließlich zum Erfüllen der Aufrufe aus anderen */
/*   Routinen. */
/*   Diese Datei enthält ALLE Routinen-Aufrufe für den Treiber und */
/*   wird in der Bibliothek IZ_DTRB.LIB mit den anderen DUMMY- */
/*   Treibern zusammengefaßt und darf erst nach den zu verwendenden */
/*   IZ-Treibern in das Programm eingebunden werden, so dass die */
/*   DUMMY-Routinen nur eingebunden werden, wenn kein IZ-Treiber */
/*   mit der entsprechenden IZ-Id. verwendet werden soll. */
/*   */
/*   aufgerufen von :   IZ_... */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   int      IZ04_INIT     Returnkode der Funktion */
/*   int      IZ04_EXIT     0=FALSE: KEIN Fehler */
/*   int      IZ04_CMD      -8: unzulässige IZ_TRB_ID, d.h. */
/*   int      IZ04_SETTM    DDC-IZ nicht verfügbar */
/*   char*    IZ04_POLL     NULL-Zeiger */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1 */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

5.4.4.2. Spezifisch für IZ01 (SMS über ISDN)

```

/*****
/*   Produkt      :   GA - Knoten           Version   :   1.3      */
/*   Modulname    :   IZ01 Treiber SMS / CAPI Stand : 04.12.2002 */
/*****
/*   Datei       :   INIT.C                */
/*   Aufruf      :   IZ01_INIT ( ulLZ_STAT ) */
/*****
/*   Beschreibung :                        */
/*   Diese Routine führt die spezifischen Initialisierungen für das */
/*   SMS-Protokoll nach dem Starten des Hauptprogrammes durch :   */
/*   - Laden der IZ-spezifischen IZ-Parameter aus der Konf.-Datei und*/
/*   spezifische Initialisierung */
/*   - Laden der IZ-spezifischen IS-Parameter aus der Konf.-Datei und*/
/*   spezifische Initialisierung */
/*   */
/*   aufgerufen von :      IZ_INIT          */
/*   */
/*   ruft auf      :      IZ_IZ_IZTEXT, CAPI_INIT, */
/*                   UT_WSL_INIT, UT_NORM_STRING, */
/*                   UT_DP_FINDFIRST, SY_CALLOC  */
/*   */
/*   I/O-Kanäle    :                        */
/*****
/*   Eingabe Parameter :                    */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   uns. long ulLZ_STAT      LZ_STAT nach der Initialisierung*/
/*                   (Kodierung vgl. in LZ_INCL.H) */
/*   ----- */
/*   Ausgabe Parameter:                    */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   int      IZ01_INIT      Returncode der Funktion */
/*                   0=FALSE: KEIN Fehler */
/*                   -1: OPEN-Fehler */
/*                   -2: READ-Fehler */
/*                   -3: WRITE-Fehler */
/*                   -4: ALLOC-Fehler */
/*                   -5: unzulässige IS-Zeile */
/*                   -6: IZ-Init-Fehler */
/*                   -7: WSL_INIT-Fehler */
/*                   -8: unzulässige IZ_TRB_ID */
/*                   >0: Nummer der Zeile, in der */
/*                   ein Fehler erkannt wurde */
/*   ----- */
/*   Transiente Parameter :                */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original  :   W. Fries   01.03.2000   1.3      */
/*   Änderung:   W. Fries   01.10.2002   1.3      WF1 (*.RER) */
/*   Änderung:   W. Fries   01.11.2002   1.3      WF2 (WSL_SP) */
/*   Änderung:   W. Fries   04.12.2002   1.3      WF3      */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3       */
/*   Modulname :   IZ01 Treiber SMS / CAPI Stand : 01.11.2002  */
/*****
/*   Datei    :   CMD.C                 */
/*   Aufruf   :   IZ01_CMD ( ucCMD )    */
/*****
/*   Beschreibung :                       */
/*   Diese Routine trägt das CMD-Datagramm "ucCMD" in die Warte- */
/*   schlange für CMD-Datagramme zur Ausgabe an die DDC-IZ ein.  */
/*   */
/*   aufgerufen von :           IZ_CMD   */
/*   */
/*   ruft auf      :           UT_WSL_WRIT */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   int      IZ01_CMD       */
/*   Returnkode der Funktion           */
/*   0=FALSE: KEIN Fehler             */
/*   -1: OPEN-Fehler                  */
/*   -2: READ-Fehler                  */
/*   -3: WRITE-Fehler                 */
/*   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.03.2000   1.3           */
/*   Änderung:   W. Fries   01.11.2002   1.3           WF1 (WSL_SP) */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3      */
/*      Modulname   :      IZ01 Treiber SMS / CAPI Stand : 01.03.2000 */
/*****
/*      Datei      :      SETTM.C                */
/*      Aufruf     :      IZ01_SETTM ( )        */
/*****
/*      Beschreibung :                          */
/*      Diese Routine initiiert das Setzen von Datum/Zeit auf der DDC-IZ*/
/*      anhand der LOOP-Zeit. Das eigentliche Setzen erfolgt in der  */
/*      Routine "POLL", damit die Kommunikation mit der DDC-IZ nicht */
/*      gestört wird.                                                */
/*      aufgerufen von :          IZ_SETTM          */
/*      ruft auf      :                          */
/*      I/O-Kanäle   :                          */
/*****
/*      Eingabe Parameter :                          */
/*      -----                          */
/*      Typ      Name          Dim.      Beschreibung          */
/*      *** KEINE ***                          */
/*      -----                          */
/*      Ausgabe Parameter:                          */
/*      -----                          */
/*      Typ      Name          Dim.      Beschreibung          */
/*      int      IZ01_SETTM          Returncode der Funktion  */
/*      0=FALSE: KEIN Fehler                          */
/*      -----                          */
/*      Transiente Parameter :                          */
/*      -----                          */
/*      Typ      Name          Dim.      Beschreibung          */
/*      *** KEINE ***                          */
/*****
/*      Autor      Datum          Version      Kurzzeichen      */
/*      -----                          */
/*      Original :      W. Fries          01.03.2000      1.3          */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3       */
/*   Modulname :   IZ01 Treiber SMS / CAPI Stand : 01.12.2002  */
/*****
/*   Datei    :   POLL.C                */
/*   Aufruf   :   IZ01_POLL ( )        */
/*****
/*   Beschreibung :                               */
/*   Diese Routine führt die zyklische Behandlung der SEND- und RECV- */
/*   Funktionen zur IZ durch. Dabei wird die CMD-Warteschlange        */
/*   ausgelesen, und falls der SEND-Puffer leer ist, an die IZ       */
/*   gesendet. Werden Datagramme von der IZ empfangen, so wird      */
/*   ein Zeiger auf das Datagramm als Ergebnis geliefert; SONST ist   */
/*   das Ergebnis ein NULL-Zeiger.                                     */
/*   aufgerufen von :           IZ_POLL                                     */
/*   ruft auf      :           IZ01_SMS_INIT, IZ01_SMS_CRC,           */
/*                   IZ01_SMS_WRUF, IZ01_CONN_REQ,                   */
/*                   IZ01_CONN_RES, IZ01_ALRT_REQ                     */
/*                   IZ01_DATA_REQ, IZ01_DISC_REQ                     */
/*                   UT_WSL_READ, UT_WSL_WRIT, UT_DT_ADD,           */
/*                   UT_KL_BW2PW, UT_UCGETBIT, UT_UCSETBIT,         */
/*                   UT_FNDERROR, UT_FNDREJECT, UT_FND_TIME,       */
/*                   UT_DP_FINDFIRST, diverse Makros FND_...       */
/*                   und SET_FND_...,                               */
/*                   SY_IEEE2F, SY_F2IEEE                             */
/*   I/O-Kanäle   :                                                     */
/*****
/*   Eingabe Parameter :                                               */
/*   -----                                                         */
/*   Typ      Name          Dim.    Beschreibung                       */
/*   -----                                                         */
/*   *** KEINE ***                                                    */
/*   -----                                                         */
/*   Ausgabe Parameter:                                               */
/*   -----                                                         */
/*   Typ      Name          Dim.    Beschreibung                       */
/*   -----                                                         */
/*   char *   IZ01_POLL      Zeiger auf das empfangenen FND-        */
/*                   Datagramm durch                               */
/*                   SONST: NULL-Zeiger                             */
/*   -----                                                         */
/*   Transiente Parameter :                                           */
/*   -----                                                         */
/*   Typ      Name          Dim.    Beschreibung                       */
/*   -----                                                         */
/*   *** KEINE ***                                                    */
/*****
/*   Autor      Datum      Version  Kurzzeichen                       */
/*   -----                                                         */
/*   Original :   W. Fries    01.03.2000   1.3                               */
/*   Änderung:   W. Fries    01.10.2001   1.3   WF1 (iVBP_Art,        */
/*                   iVBP_Prot) */
/*   Änderung:   W. Fries    01.10.2002   1.3   WF2 (D2, ePlus) */
/*   Änderung:   W. Fries    01.11.2002   1.3   WF3 (WSL_SP) */
/*   Änderung:   W. Fries    01.12.2002   1.3   WF4 (Mail,Fax) */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3      */
/*      Modulname   :      IZ01 Treiber SMS / CAPI Stand : 01.11.2002 */
/*****
/*      Datei      :      EXIT.C                */
/*      Aufruf     :      IZ01_EXIT ( )        */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*      FND-IZ vor dem Stoppen des Hauptprogrammes durch :          */
/*      - Schließen der Warteschlangen für CMD- und ACK-Datagramme  */
/*      - Freigabe allocierter Speicherbereiche                    */
/*      - Trennen der Verbindung zur DDC-IZ                        */
/*      */
/*      aufgerufen von :      IZ_EXIT          */
/*      */
/*      ruft auf      :      CAPI_EXIT, UT_WSL_EXIT, SY_FREE        */
/*      */
/*      I/O-Kanäle   :                        */
/*      */
/*****
/*      Eingabe Parameter :                        */
/*      ----- */
/*      Typ      Name          Dim.   Beschreibung          */
/*      */
/*      *** KEINE *** */
/*      */
/*      ----- */
/*      Ausgabe Parameter:                        */
/*      ----- */
/*      Typ      Name          Dim.   Beschreibung          */
/*      */
/*      int      IZ01_EXIT          Returnkode der Funktion */
/*      */
/*      0=FALSE: KEIN Fehler */
/*      -1: OPEN-Fehler */
/*      -2: READ-Fehler */
/*      -3: WRITE-Fehler */
/*      -4: ALLOC-Fehler */
/*      -5: unzulässige IS-Zeile */
/*      -6: DDC-IZ-Init-Fehler */
/*      >0: Nummer der Zeile, in der */
/*      ein Fehler erkannt wurde */
/*      */
/*      ----- */
/*      Transiente Parameter :                        */
/*      ----- */
/*      Typ      Name          Dim.   Beschreibung          */
/*      */
/*      *** KEINE *** */
/*      */
/*****
/*      Autor      Datum      Version   Kurzzeichen   */
/*      ----- */
/*      Original :   W. Fries   01.03.2000   1.3           */
/*      Änderung:   W. Fries   01.11.2002   1.3           WF1 (WSL_SP) */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3       */
/*   Modulname :   IZ01 Treiber SMS / CAPI Stand : 01.10.2002  */
/*****
/*   Datei    :   LIST_REQ.C           */
/*   Aufruf   :   IZ01_LIST_REQ ( iContr ) */
/*****
/*   Beschreibung :           */
/*   Diese Routine startet die Empfangsbereitschaft für den ISDN- */
/*   Controller "iContr".           */
/*           */
/*   aufgerufen von :           IZ01_INIT           */
/*           */
/*   ruft auf      :           CAPI_SEND           */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           -----           */
/*   int      iContr           Nummer des ISDN-Controller */
/*           gemäß CAPI 2.0           */
/*           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           -----           */
/*   int      IZ01_LIST_REQ           Returnkode von CAPI_SEND */
/*           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           -----           */
/*   *** KEINE ***           */
/*           */
/*****
/*           Autor      Datum      Version      Kurzzeichen   */
/*           -----           */
/* Original :   W. Fries      01.03.2000      1.3           */
/* Änderung:   W. Fries      01.10.2002      1.3           WF1 (*.RER) */
/*           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3      */
/*      Modulname    :      IZ01 Treiber SMS / CAPI Stand : 01.10.2002 */
/*****
/*      Datei       :      ALRT_REQ.C           */
/*      Aufruf      :      IZ01_ALRT_REQ ( VB )  */
/*****
/*      Beschreibung :                               */
/*      Diese Routine reserviert die Verbindung "VB". */
/*      */
/*      aufgerufen von :      IZ01_POLL         */
/*      */
/*      ruft auf      :      CAPI_SEND         */
/*      */
/*      I/O-Kanäle   :                               */
/*      */
/*****
/*      Eingabe Parameter :                               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      IZ01_VB*  VB              Zeiger auf den Eintrag in der */
/*      VB-Struktur für den der Verbin- */
/*      dungs-Abbau durchgeführt werden */
/*      soll. */
/*      ----- */
/*      Ausgabe Parameter:                               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      int      IZ01_ALRT_REQ      Returnkode von CAPI_SEND */
/*      ----- */
/*      Transiente Parameter :                               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/*      Original :      W. Fries      01.10.2002      1.3      */
/*      Änderung:      */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3      */
/*      Modulname   :      IZ01 Treiber SMS / CAPI Stand : 01.10.2002 */
/*****
/*      Datei      :      CONN_REQ.C           */
/*      Aufruf     :      IZ01_CONN_REQ ( VB, cRufnummer ) */
/*****
/*      Beschreibung :                               */
/*      Diese Routine startet für die Verbindung "VB" einen Verbindungs-*/
/*      Aufbau zur DDC-IZ mit der Rufnummer "cRufnummer" und identifi- */
/*      ziert sich mit der eigenen Rufnummer.                               */
/*                                                                           */
/*      aufgerufen von :      IZ01_POLL                                           */
/*                                                                           */
/*      ruft auf      :      CAPI_SEND                                           */
/*                                                                           */
/*      I/O-Kanäle   :                               */
/*****
/*      Eingabe Parameter :                               */
/*      -----                               */
/*      Typ      Name      Dim.      Beschreibung                               */
/*      -----                               */
/*      IZ01_VB* VB                               Zeiger auf den Eintrag in der */
/*      VB-Struktur                               */
/*      -----                               */
/*      char*   cRufnummer   Rufnummer der DDC-IZ                               */
/*      (Called Party Number)                               */
/*      -----                               */
/*      Ausgabe Parameter:                               */
/*      -----                               */
/*      Typ      Name      Dim.      Beschreibung                               */
/*      -----                               */
/*      int      IZ01_CONN_REQ   Returnkode von CAPI_SEND                               */
/*      -----                               */
/*      Transiente Parameter :                               */
/*      -----                               */
/*      Typ      Name      Dim.      Beschreibung                               */
/*      -----                               */
/*      *** KEINE ***                               */
/*****
/*      Autor      Datum      Version      Kurzzeichen   */
/*      -----                               */
/*      Original :   W. Fries      01.03.2000   1.3                               */
/*      Änderung:   W. Fries      01.10.2002   1.3      WF1(D2, ePlus) */
/*      -----                               */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3       */
/*   Modulname :   IZ01 Treiber SMS / CAPI Stand : 01.03.2000  */
/*****
/*   Datei    :   CONN_RES.C           */
/*   Aufruf   :   IZ01_CONN_RES ( VB, iREJECT )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine sendet für das CONNECT_IND zu "VB" das zugehörige */
/*   CONNECT_RESP mit dem REJECT-Kode "iREJECT".           */
/*           */
/*   aufgerufen von :           IZ01_POLL           */
/*           */
/*   ruft auf :           CAPI_SEND           */
/*           */
/*   I/O-Kanäle :           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----           */
/*   IZ01_VB* VB           Zeiger auf den Eintrag in der */
/*   VB-Struktur für den das */
/*   CONNECT_RESP gesendet werden */
/*   soll.           */
/*   int      iREJECT      REJECT für Anruf           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----           */
/*   int      IZ01_CONN_RES      Returnkode von CAPI_SEND */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----           */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:   W. Fries      01.12.99      1.3      WF1 (LINUX) */
/*   -----           */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3       */
/*   Modulname :   IZ01 Treiber SMS / CAPI Stand : 01.03.2000  */
/*****
/*   Datei    :   DATA_REQ.C           */
/*   Aufruf   :   IZ01_DATA_REQ ( VB, cDatagramm, iLen )      */
/*****
/*   Beschreibung :   */
/*   Diese Routine sendet die Daten in "cDatagramm" der Länge "iLen" */
/*   über die Verbindung "VB". */
/*   */
/*   aufgerufen von :   IZ01_POLL */
/*   */
/*   ruft auf      :   CAPI_SEND, CAPI_ERROR */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   IZ01_VB*  VB        */       Zeiger auf den Eintrag in der */
/*   */       VB-Struktur für den die Daten- */
/*   */       übertragung durchgeführt werden */
/*   */       soll. */
/*   char[]   cDatagramm */       Zu übertragendes Datagramm */
/*   */       */
/*   int      iLen        */       Länge des zu übertragenden Data- */
/*   */       grammes */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   int      IZ01_DATA_REQ */       Returnkode von CAPI_SEND */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries   01.03.2000   1.3 */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3      */
/*      Modulname   :      IZ01 Treiber SMS / CAPI Stand : 01.03.2000 */
/*****
/*      Datei      :      DISC_REQ.C           */
/*      Aufruf     :      IZ01_DISC_REQ ( VB ) */
/*****
/*      Beschreibung :                               */
/*      Diese Routine bricht die Verbindung "VB" ab. */
/*                                                     */
/*      aufgerufen von :      IZ01_POLL          */
/*                                                     */
/*      ruft auf      :      CAPI_SEND          */
/*                                                     */
/*      I/O-Kanäle   :                               */
/*                                                     */
/*****
/*      Eingabe Parameter :                               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*                                                     */
/*      IZ01_VB*  VB      Zeiger auf den Eintrag in der */
/*                                                     */
/*                                                     */
/*                                                     */
/*                                                     */
/*                                                     */
/*      ----- */
/*      Ausgabe Parameter:                               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*                                                     */
/*      int      IZ01_DISC_REQ      Returnkode von CAPI_SEND */
/*                                                     */
/*      ----- */
/*      Transiente Parameter :                               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*                                                     */
/*      *** KEINE *** */
/*                                                     */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/*      Original :      W. Fries      01.03.2000      1.3      */
/*      Änderung: */
/*                                                     */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



**5.4.4.3. Spezifisch für IZ02 (AEG/Schneider Groupe Modicon)**

```

/*****
/*   Produkt      :   GA - Knoten           Version   :   1.2      */
/*   Modulname    :   IZ02 Treiber AEG KS120 Stand   :   01.10.98 */
/*****
/*   Datei       :   INIT.C                */
/*   Aufruf      :   IZ02_INIT ( ullZ_STAT ) */
/*****
/*   Beschreibung :                        */
/*   Diese Routine führt die spezifischen Initialisierungen für die */
/*   AEG-KS120-IZ nach dem Starten des Hauptprogrammes durch :     */
/*   - Laden der IZ-spezifischen IS-Parameter aus der Konf.-Datei und */
/*     spezifische Initialisierung */
/*   aufgerufen von :           IZ_INIT */
/*   ruft auf      :           IZ02_COM_INIT, IZ02_COM_CIN, UT_WSL_INIT */
/*                               UT_NORM_STRING, UT_DP_FINDFIRST, */
/*                               SY_CALLOC */
/*   I/O-Kanäle   :                        */
/*****
/*   Eingabe Parameter :                    */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   uns. long  ullZ_STAT           LZ_STAT nach der Initialisierung */
/*                               (Kodierung vgl. in LZ_INCL.H) */
/*   ----- */
/*   Ausgabe Parameter:                    */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   int      IZ02_INIT           Returnkode der Funktion */
/*                               0=FALSE: KEIN Fehler */
/*                               -1: OPEN-Fehler */
/*                               -2: READ-Fehler */
/*                               -3: WRITE-Fehler */
/*                               -4: ALLOC-Fehler */
/*                               -5: unzulässige IS-Zeile */
/*                               -6: DDC-IZ-Init-Fehler */
/*                               -7: WSL_INIT-Fehler */
/*                               -8: unzulässige IZ_TRB_ID */
/*                               >0: Nummer der Zeile, in der */
/*                               ein Fehler erkannt wurde */
/*   ----- */
/*   Transiente Parameter :                */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1 */
/*   Änderung: */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ02 Treiber AEG KS120 Stand   :   01.10.98   */
/*****
/*   Datei     :   CMD.C                 */
/*   Aufruf    :   IZ02_CMD ( ucCMD )     */
/*****
/*   Beschreibung :                       */
/*   Diese Routine trägt das CMD-Datagramm "ucCMD" in die Warte- */
/*   schlange für CMD-Datagramme zur Ausgabe an die DDC-IZ ein.  */
/*   */
/*   aufgerufen von :           IZ_CMD     */
/*   */
/*   ruft auf      :           UT_WSL_WRIT */
/*   */
/*   I/O-Kanäle    :                       */
/*   */
/*****
/*   Eingabe Parameter :                 */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   *** KEINE *** */
/*   */
/* ----- */
/*   Ausgabe Parameter:                 */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   int      IZ02_CMD      */
/*           Returnkode der Funktion */
/*           0=FALSE: KEIN Fehler    */
/*           -1: OPEN-Fehler         */
/*           -2: READ-Fehler         */
/*           -3: WRITE-Fehler        */
/*   */
/* ----- */
/*   Transiente Parameter :             */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*           Autor      Datum      Version   Kurzzeichen */
/*           ----- */
/* Original :   W. Fries   01.05.97    1.1     */
/* Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ02 Treiber AEG KS120 Stand   :   01.10.98  */
/*****
/*   Datei     :   SETTM.C               */
/*   Aufruf    :   IZ02_SETTM ( )       */
/*****
/*   Beschreibung :                       */
/*   Diese Routine initiiert das Setzen von Datum/Zeit auf der DDC-IZ*/
/*   anhand der LOOP-Zeit. Das eigentliche Setzen erfolgt in der  */
/*   Routine "POLL", damit die Kommunikation mit der DDC-IZ nicht */
/*   gestört wird.                                               */
/*   aufgerufen von :           IZ_SETTM                               */
/*   ruft auf      :                                           */
/*   I/O-Kanäle   :                                           */
/*****
/*   Eingabe Parameter :                                           */
/*   -----                                                    */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   *** KEINE ***                                                    */
/*   -----                                                    */
/*   Ausgabe Parameter:                                           */
/*   -----                                                    */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   int      IZ02_SETTM          Returnkode der Funktion          */
/*                                          0=FALSE: KEIN Fehler          */
/*   -----                                                    */
/*   Transiente Parameter :                                           */
/*   -----                                                    */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   *** KEINE ***                                                    */
/*****
/*   Autor      Datum      Version   Kurzzeichen                */
/*   -----                                                    */
/*   Original :   W. Fries      01.05.97      1.1                */
/*   Änderung:                                           */
/*   -----                                                    */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      IZ02 Treiber AEG KS120 Stand   :   01.10.98 */
/*****
/*      Datei       :      POLL.C                */
/*      Aufruf      :      IZ02_POLL ( )        */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die zyklische Behandlung der SEND- und RECV- */
/*      Funktionen zur DDC-IZ durch. Dabei wird die CMD-Warteschlange */
/*      ausgelesen, und falls der SEND-Puffer leer ist, an die DDC-IZ */
/*      gesendet. Werden Datagramme von der DDC-IZ empfangen, so wird */
/*      ein Zeiger auf das Datagramm als Ergebnis geliefert; SONST ist */
/*      das Ergebnis ein NULL-Zeiger. */
/*      */
/*      WF1 - zyklisches Lesen des Merkerbyte 2 ( 1/Min.). Ist dieses auf */
/*      WF1 0x01 gesetzt, so wird dieses als IZ-Störung interpretiert und */
/*      WF1 im IZ-Status vermerkt. */
/*      WF1 - zyklisches Lesen des Merkerbyte-Bereiches für die IME-Daten- */
/*      WF1 Strukturen (1/Sek.) und deren Behandlung abhängig vom Flag. */
/*      */
/*      WF2 - Insbesondere nach einem OVERRUN traten bei der ALU 202 Puffer- */
/*      WF2 Probleme auf, die zu einem Versatz der gelesenen SPS-Merker- */
/*      WF2 Bereiche und damit zu einer fehlerhaften Zuordnung der Werte */
/*      WF2 zu den Datenpunkten führte. Gemäß den Vorgaben von Herrn */
/*      WF2 Spitzley (E-Mail vom 20.3.98) wird eine spezifische Initiali- */
/*      WF2 sierung-Behandlung durch geführt, die eine Normierung der */
/*      WF2 Puffer sicherstellen soll. */
/*      */
/*      */
/*      aufgerufen von :      IZ_POLL */
/*      */
/*      ruft auf      :      IZ02_MB2FND, IZ02_COM_KT80, */
/*      IZ02_COM_COUT, IZ02_COM_CIN, */
/*      IZ02_COM_PURG, IZ02_COM_STAT, */
/*      UT_FNDERROR, UT_FNDREJECT, UT_WSL_READ, */
/*      UT_DP_FINDFIRST, UT_FND_TIME, diverse */
/*      Makros FND_... und SET_FND_..., */
/*      SY_IEEE2F */
/*      */
/*      I/O-Kanäle      :      */
/*****
/*      Eingabe Parameter :                        */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                        */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      char *  IZ02_POLL      Zeiger auf das empfangenen FND- */
/*      Datum durch */
/*      SONST: NULL-Zeiger */
/*      ----- */
/*      Transiente Parameter :                        */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :  W. Fries      01.05.97      1.1      */
/*      Änderung:  W. Fries      01.02.98      1.2      WF1 */
/*      Änderung:  W. Fries      23.05.98      1.2      WF2 */
/*      */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ02 Treiber AEG KS120 Stand   :   01.10.98  */
/*****
/*   Datei     :   EXIT.C                */
/*   Aufruf    :   IZ02_EXIT ( )        */
/*****
/*   Beschreibung :                               */
/*   Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*   FND-IZ vor dem Stoppen des Hauptprogrammes durch :          */
/*   - Schließen der Warteschlange für CMD-Datagramme           */
/*   - Freigabe ALLOCierter Speicherbereiche                   */
/*   - Trennen der Verbindung zur DDC-IZ                       */
/*   */
/*   aufgerufen von :           IZ_EXIT                               */
/*   */
/*   ruft auf      :           IZ02_COM_STAT, IZ02_COM_EXIT,       */
/*                   UT_WSL_EXIT, SY_FREE                          */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***                                           */
/*   */
/*   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   int      IZ02_EXIT                               */
/*                   Returnkode der Funktion                */
/*                   0=FALSE: KEIN Fehler                   */
/*                   -1: OPEN-Fehler                        */
/*                   -2: READ-Fehler                        */
/*                   -3: WRITE-Fehler                       */
/*                   -4: ALLOC-Fehler                       */
/*                   -5: unzulässige IS-Zeile               */
/*                   -6: DDC-IZ-Init-Fehler                 */
/*                   >0: Nummer der Zeile, in der          */
/*                       ein Fehler erkannt wurde           */
/*   */
/*   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***                                           */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen           */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1             */
/*   Änderung:                               */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ02 Treiber AEG KS120 Stand   :   01.10.98   */
/*****
/*   Datei     :   MB2FND.C              */
/*   Aufruf    :   IZ02_MB2FND ( DP, ucMB, ucAPDU_Buf )         */
/*****
/*   Beschreibung :                               */
/*   Diese Routine setzt für den Datenpunkt DP die SPS-Merkerbytes */
/*   "ucMB" in das vorbereitete FND-Datagramm "ucAPDU_Buf" um.    */
/*   */
/*   aufgerufen von :           IZ02_POLL          */
/*   */
/*   ruft auf      :           UT_KL_BW2PW, UT_UCSETBIT, diverse */
/*   Makros FND_... und SET_FND_...,          */
/*   SY_F2IEEE                                         */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung          */
/*   */
/*   UT_DP*   DP            Zeiger auf Datenpunkt, zu dem      */
/*   die Merker-Bytes gehören          */
/*   uns. char ucMB[]      Merker-Bytes; Anzahl abhängig      */
/*   von "dp_type"                      */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung          */
/*   */
/*   int      IZ02_MB2FND   Returncode der Funktion          */
/*   0=FALSE: KEIN Fehler                */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung          */
/*   */
/*   uns. char ucAPDU_Buf FND_APDU_LEN FND-Datagramm; dieses wurde von */
/*   der aufrufenden Routine bereits */
/*   vorbereitet und enthält mind.      */
/*   "dp_type", "tab_id".                */
/*   Innerhalb dieser Routine werden    */
/*   die Datenpunkt-Tabellen DP_#0,     */
/*   DP_#1 und/oder DP_#2 ergänzt.      */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1          */
/*   Änderung:   W. Fries    01.02.98    1.2          WF1      */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

;*****/
;*   Produkt   :   GA - Knoten           Version   :   1.2       */
;*   Modulname :   IZ Treiber AEG SPS A120 Stand :   01.10.98  */
;*****/
;*   Datei    :   COMnxxxx.ASM          */
;*   Aufruf   :   IZ02_COMnxxxx ( Aufrufparameter siehe unten ) */
;*****/
;*
;*   Beschreibung :
;*   Diese Assembler-Quelle enthält alle Routinen-Aufrufe, die zur
;*   Steuerung der seriellen Schnittstellen COM1 - COM4 notwendig
;*   sind.
;*   Die Ein-/Ausgabe von Daten erfolgt zeichenweise über getrennte
;*   Datenpuffer in denen die empfangenen bzw. zu sendenden Daten
;*   zwischengepuffert werden. Die Zeichen im Datenpuffer werden
;*   über Interrupt-Service-Routinen aus dem I/O-Baustein (Intel
;*   8250-kompatibel) gelesen bzw. dorthin geschrieben, so dass eine
;*   Entkopplung des Programmes vom Datentransfer auf der seriellen
;*   Schnittstelle erfolgt. Die spezifischen Interrupts für die
;*   FIFO-Behandlung (z.B. UART 16550) werden unterdrückt, jedoch
;*   können die FIFO's mittels COM_FIFO aktiviert werden.
;*
;*   Gesteuert wird die Übertragung durch ein Hardware-Handshake
;*   über die RTS/CTS-Signale, d.h. diese Signale müssen stets ver-
;*   drahtet sein
;*
;*   Spezifisch für die Anforderungen der speicherprogrammierbaren
;*   Steuerung (SPS) vom Typ MODICON A120 der Firma AEG wird
;*   jedesmal, wenn von ihr ein Kurztelegramm 80 (KT80) empfangen
;*   wird, ein solches zurückgesendet. Dadurch wird der geforderte
;*   Zeitrahmen für die Kommunikation sichergestellt, da sonst
;*   die A120 die Anfrage des Master (Request) storniert, d.h. kein
;*   Antworttelegramm sendet.
;*
;*   Grundlage für die Realisierung ist das "Benutzerhandbuch A120
;*   KS-Funktionen" (Dok.Nr.: A91M.271974.20-0791). Hinzu kommen
;*   praktische Tests des Autors.
;*
;*   Hinweis:      Alle Aufruf-Parameter und das Ergebnis der
;*                 Funktion sind vom Typ "int" bzw. "INTEGER*2",
;*                 d.h. sie sind vom Datentyp "WORD".
;*
;*   Hinweis:      Die Schnittstellen-Parameter (Baudrate, Daten-,
;*                 Parity- und Stop-Bits müssen vor Aufruf von
;*                 COMnINIT erfolgen, damit in COMnINIT gemachte Initialisierungen
;*                 nicht zerstört werden !
;*
;*   Hinweis:      Aus programmiertechnischen Gründen sind ALLE in
;*                 dieser einen Routine zusammengefaßt, sodass jede
;*                 Routine nachfolgend detailliert beschrieben werden muß.
;*
;*   Folgende Aufrufe stehen zur Verfügung :
;*   =====
;*
;*   IZ02_COM_INIT   Initialisierung (ERSTE aufgerufene Routine)
;*   IZ02_COM_EXIT   ENDE-Behandlung (LETZTE aufgerufene Routine)
;*
;*   IZ02_COM_CIN   Ein Zeichen aus Eingabe-Puffer lesen, falls ex.
;*   IZ02_COM_COUT  Ein Zeichen an Ausgabe-Puffer übergeben
;*
;*   IZ02_COM_PURG  Ausgabe-Puffer löschen
;*   IZ02_COM_STAT  aktuellen Status abfragen
;*   IZ02_COM_EVNT  aktuellen Status abfragen und Events rücksetzen
;*   IZ02_COM_KT80  Sonderbehandlung Polling für den Kommunika-
;*                 tionsbetrieb (KT80)
;*****/
;*
;*   Autor      Datum      Version      Kurzzeichen
;*   -----
;*   Original :   W. Fries      01.05.97      1.1
;*   Änderung:
;*
;*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
;*****/

```

5.4.4.4. Spezifisch für IZ03 (FND 1.0 über ISDN / V.24)

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   IZ03_INIT ( ullZ_STAT ) */
/*****
/*   Beschreibung :
/*   Diese Routine führt die spezifischen Initialisierungen für die
/*   DDC-IZ nach dem Starten des Hauptprogrammes durch :
/*   - Laden der IZ-spezifischen IS-Parameter aus der Konf.-Datei und
/*   spezifische Initialisierung
/*   - Initialisieren der Schnittstelle zur DDC-IZ
/*
/*   aufgerufen von :           IZ_INIT
/*
/*   ruft auf      :           IZ03_COM_INIT, IZ03_COM_SET_DTR,
/*                           IZ03_COM_CIN, IZ_IZ_IZTEXT, CAPI_INIT,
/*                           UT_WSL_INIT, UT_NORM_STRING,
/*                           UT_DP_FINDFIRST, SY_CALLOC
/*
/*   I/O-Kanäle   :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   uns. long ullZ_STAT          LZ_STAT nach der Initialisierung*
/*                           (Kodierung vgl. in LZ_INCL.H)
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   int      IZ03_INIT          Returncode der Funktion
/*                           0=FALSE: KEIN Fehler
/*                           -1: OPEN-Fehler
/*                           -2: READ-Fehler
/*                           -3: WRITE-Fehler
/*                           -4: ALLOC-Fehler
/*                           -5: unzulässige IS-Zeile
/*                           -6: DDC-IZ-Init-Fehler
/*                           -7: WSL_INIT-Fehler
/*                           -8: unzulässige IZ_TRB_ID
/*                           >0: Nummer der Zeile, in der
/*                               ein Fehler erkannt wurde
/*
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version  Kurzzeichen
/*   -----
/*   Original :   W. Fries    01.05.97    1.1
/*   Änderung:
/*
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   CMD.C                 */
/*   Aufruf   :   IZ03_CMD ( ucCMD )    */
/*****
/*   Beschreibung :                       */
/*   Diese Routine trägt das CMD-Datagramm "ucCMD" in die Warte- */
/*   schlange für CMD-Datagramme zur Ausgabe an die DDC-IZ ein.  */
/*   */
/*   aufgerufen von :           IZ_CMD   */
/*   */
/*   ruft auf      :           UT_WSL_WRIT */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   int      IZ03_CMD      */
/*   */
/*   Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler   */
/*   -1: OPEN-Fehler       */
/*   -2: READ-Fehler       */
/*   -3: WRITE-Fehler      */
/*   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1         */
/*   Änderung:                                     */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   SETTM.C                */
/*   Aufruf   :   IZ03_SETTM ( )        */
/*****
/*   Beschreibung :                       */
/*   Diese Routine gleicht die DDC-IZ-interne Uhrzeit mit der */
/*   Uhrzeit des GA-Knotens ab.          */
/*   aufgerufen von :           IZ_SETTM */
/*   ruft auf      :           */
/*   I/O-Kanäle   :           */
/*****
/*   Eingabe Parameter :                 */
/*   -----                             */
/*   Typ      Name      Dim.   Beschreibung */
/*   *** KEINE ***                       */
/*   -----                             */
/*   Ausgabe Parameter:                 */
/*   -----                             */
/*   Typ      Name      Dim.   Beschreibung */
/*   int      IZ03_SETTM          Returnkode der Funktion */
/*                                          0=FALSE: KEIN Fehler */
/*   -----                             */
/*   Transiente Parameter :             */
/*   -----                             */
/*   Typ      Name      Dim.   Beschreibung */
/*   *** KEINE ***                       */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   -----                             */
/*   Original :   W. Fries      01.05.97      1.1      */
/*   Änderung:                                     */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   POLL.C                */
/*   Aufruf   :   IZ03_POLL ( )        */
/*****
/*   Beschreibung :                      */
/*   Diese Routine führt die zyklische Behandlung der SEND- und RECV- */
/*   Funktionen zur DDC-IZ durch. Dabei wird die CMD-Warteschlange  */
/*   ausgelesen, und falls der SEND-Puffer leer ist, an die DDC-IZ  */
/*   gesendet. Werden Datagramme von der DDC-IZ empfangen, so wird  */
/*   ein Zeiger auf das Datagramm als Ergebnis geliefert; SONST ist  */
/*   das Ergebnis ein NULL-Zeiger.                                         */
/*   */
/*   WF2 STOP des GA-Knotens bei Anruf eines FWD-Teilnehmers           */
/*   */
/*   aufgerufen von :           IZ_POLL                                     */
/*   */
/*   ruft auf      :           IZ03_COM_SET_DTR, IZ03_COM_STAT,         */
/*   IZ03_COM_COUT, IZ03_COM_CIN,                                       */
/*   IZ03_CONN_REQ, IZ03_DATA_REQ,                                       */
/*   IZ03_DISC_REQ, IZ03_CAPIPOLL,                                       */
/*   CAPI_ERROR,                                                         */
/*   UT_WSL_WRIT, UT_WSL_READ, UT_DT_ADD,                                 */
/*   UT_FNDREJECT, diverse Makros FND_...                               */
/*   und SET_FND...                                                       */
/*   */
/*   I/O-Kanäle   :                      */
/*   */
/*****
/*   Eingabe Parameter :                      */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                      */
/*   */
/*   *** KEINE ***                                                       */
/*   ----- */
/*   Ausgabe Parameter:                      */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                      */
/*   */
/*   char *   IZ03_POLL      Zeiger auf das empfangene FND-          */
/*   Datagramm                                                       */
/*   SONST: NULL-Zeiger                                              */
/*   ----- */
/*   Transiente Parameter :                      */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                      */
/*   */
/*   *** KEINE ***                                                       */
/*****
/*   Autor      Datum      Version   Kurzzeichen                      */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1                               */
/*   Änderung:   W. Fries   15.12.97   1.1b   WF1                       */
/*   Änderung:   W. Fries   07.09.98   1.1d   WF2                       */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770  */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   EXIT.C                */
/*   Aufruf   :   IZ03_EXIT ( )         */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*   FND-IZ vor dem Stoppen des Hauptprogrammes durch :           */
/*   - Schließen der Warteschlangen für CMD- und ACK-Datagramme  */
/*   - Freigabe allocierter Speicherbereiche                     */
/*   - Trennen der Verbindung zur DDC-IZ                         */
/*   */
/*   aufgerufen von :           IZ_EXIT                                     */
/*   */
/*   ruft auf      :           IZ03_COM_SET_DTR, IZ03_COM_STAT,      */
/*                       IZ03_COM_EXIT, CAPI_EXIT, UT_WSL_EXIT,    */
/*                       SY_FREE                                     */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                       */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                       */
/*   */
/*   int      IZ03_EXIT      Dim.   Returnkode der Funktion       */
/*   */
/*   0=FALSE: KEIN Fehler */
/*   -1: OPEN-Fehler      */
/*   -2: READ-Fehler     */
/*   -3: WRITE-Fehler    */
/*   -4: ALLOC-Fehler    */
/*   -5: unzulässige IS-Zeile */
/*   -6: DDC-IZ-Init-Fehler */
/*   >0: Nummer der Zeile, in der */
/*   ein Fehler erkannt wurde */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung                       */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*      Datei      :      LIST_REQ.C           */
/*      Aufruf     :      IZ03_LIST_REQ ( iContr ) */
/*****
/*      Beschreibung :                               */
/*      Diese Routine startet die Empfangsbereitschaft für den ISDN- */
/*      Controller "iContr".                               */
/*      aufgerufen von :          IZ03_INIT           */
/*      ruft auf      :          CAPI_SEND           */
/*      I/O-Kanäle   :                               */
/*****
/*      Eingabe Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung           */
/*      int      iContr          Nummer des ISDN-Controller */
/*      gemäß CAPI 2.0                               */
/*      -----                               */
/*      Ausgabe Parameter:                               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung           */
/*      int      IZ03_LIST_REQ      Returnkode von CAPI_SEND */
/*      -----                               */
/*      Transiente Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung           */
/*      *** KEINE ***                               */
/*****
/*      Autor      Datum      Version   Kurzzeichen   */
/*      -----                               */
/*      Original :      W. Fries      01.05.97      1.1           */
/*      Änderung:                               */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt   :      GA - Knoten           Version   :   1.2      */
/*      Modulname :      IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98 */
/*****
/*      Datei    :      CONN_REQ.C           */
/*      Aufruf   :      IZ03_CONN_REQ ( VB, cRufnummer )           */
/*****
/*      Beschreibung :           */
/*      Diese Routine startet für die Verbindung "VB" einen Verbindungs-*/
/*      Aufbau zur DDC-IZ mit der Rufnummer "cRufnummer" und identifi- */
/*      ziert sich mit der eigenen Rufnummer.           */
/*      */
/*      aufgerufen von :           IZ03_POLL           */
/*      */
/*      ruft auf      :           CAPI_SEND           */
/*      */
/*      I/O-Kanäle   :           */
/*      */
/*****
/*      Eingabe Parameter :           */
/*      -----           */
/*      Typ      Name           Dim.      Beschreibung           */
/*      */
/*      IZ03_VB*  VB           Zeiger auf den Eintrag in der VB-Struktur */
/*      */
/*      char*    cRufnummer     Rufnummer der DDC-IZ           */
/*      (Called Party Number)           */
/*      */
/*      -----           */
/*      Ausgabe Parameter:           */
/*      -----           */
/*      Typ      Name           Dim.      Beschreibung           */
/*      */
/*      int      IZ03_CONN_REQ   Returnkode von CAPI_SEND           */
/*      */
/*      -----           */
/*      Transiente Parameter :           */
/*      -----           */
/*      Typ      Name           Dim.      Beschreibung           */
/*      */
/*      *** KEINE ***           */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen           */
/*      -----           */
/*      Original  :      W. Fries      01.05.97      1.1           */
/*      Änderung:           */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   CONN_RES.C           */
/*   Aufruf   :   IZ03_CONN_RES ( VB, iREJECT )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine sendet für das CONNECT_IND zu "VB" das zugehörige */
/*   CONNECT_RESP mit dem REJECT-Kode "iREJECT".           */
/*           */
/*           */
/*   aufgerufen von :           IZ03_POLL           */
/*           */
/*   ruft auf      :           CAPI_SEND           */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*           -----           */
/*   IZ03_VB* VB           Zeiger auf den Eintrag in der */
/*           VB-Struktur für den das */
/*           CONNECT_RESP gesendet werden */
/*           soll.           */
/*           */
/*   int      iREJECT      REJECT für Anruf           */
/*           */
/* - - - - -           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*           -----           */
/*   int      IZ03_CONN_RES Returnkode von CAPI_SEND           */
/*           */
/* - - - - -           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*           -----           */
/*   *** KEINE ***           */
/*           */
/*****
/*           Autor      Datum      Version   Kurzzeichen   */
/*           -----           */
/* Original :   W. Fries      01.05.97      1.1           */
/* Änderung:           */
/*           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei    :   DATA_REQ.C           */
/*   Aufruf   :   IZ03_DATA_REQ ( VB, cDatagramm, iLen )      */
/*****
/*   Beschreibung :   */
/*   Diese Routine sendet die Daten in "cDatagramm" der Länge "iLen" */
/*   über die Verbindung "VB". */
/*   */
/*   aufgerufen von :   IZ03_POLL      */
/*   */
/*   ruft auf      :   CAPI_SEND, CAPI_ERROR */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung      */
/*   */
/*   IZ03_VB*  VB       */      Zeiger auf den Eintrag in der */
/*   */      VB-Struktur für den die Daten- */
/*   */      übertragung durchgeführt werden */
/*   */      soll. */
/*   */
/*   char[]   cDatagramm   Zu übertragendes Datagramm */
/*   */
/*   int      iLen          Länge des zu übertragenden Data- */
/*   */      grammes */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung      */
/*   */
/*   int      IZ03_DATA_REQ   Returnkode von CAPI_SEND */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung      */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98 */
/*****
/*      Datei      :      DISC_REQ.C             */
/*      Aufruf    :      IZ03_DISC_REQ ( VB )   */
/*****
/*      Beschreibung :                               */
/*      Diese Routine bricht die Verbindung "VB" ab. */
/*      */
/*      aufgerufen von :          IZ03_POLL      */
/*      */
/*      ruft auf      :          CAPI_SEND      */
/*      */
/*      I/O-Kanäle   :                               */
/*      */
/*****
/*      Eingabe Parameter :                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung                               */
/*      IZ03_VB*  VB          *          Zeiger auf den Eintrag in der          */
/*      VB-Struktur für den der Verbin- */
/*      dungs-Abbau durchgeführt werden */
/*      soll.                               */
/*      ----- */
/*      Ausgabe Parameter:                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung                               */
/*      int      IZ03_DISC_REQ          Returnkode von CAPI_SEND          */
/*      ----- */
/*      Transiente Parameter :                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung                               */
/*      *** KEINE ***                               */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen          */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1          */
/*      Änderung:                               */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98  */
/*****
/*   Datei     :   CAPIPOLL.C           */
/*   Aufruf    :   IZ03_CAPIPOLL ( ipSCMD, ipInfo, cpBuf, ipLen, VB) */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft, ob von der CAPI-Schnittstelle ein Command */
/*   empfangen wurde und bearbeitet es automatisch, d.h. ggf. wird */
/*   das zugehörige RESP-Datagramm erzeugt und der Verbindungs- bzw. */
/*   Abbau durchgeführt. */
/*   Jedes empfangene Command wird mit Zusatzinformationen an die */
/*   aufrufende Routine als Ergebnis geliefert. */
/*   Nach Empfang von CONNECT_IND wird auf den Aufruf von CONN_RES */
/*   gewartet, so dass die aufrufende Routine die Anrufberechtigung */
/*   prüfen muß. */
/*   */
/*   aufgerufen von :   IZ03_POLL */
/*   */
/*   ruft auf      :   CAPI_RECV, CAPI_SEND, */
/*   */
/*   I/O-Kanäle    :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   */
/*   int      IZ03_CAPIPOLL   */
/*   falls vorhanden, SONST = 0 */
/*   */
/*   int*     ipSCMD         */
/*   CAPI-Subcommand für das Ergebnis */
/*   falls vorhanden, SONST = 0 */
/*   */
/*   int*     ipInfo         */
/*   CAPI-Info für das Ergebnis, */
/*   falls vorhanden, SONST = 0 */
/*   - LISTEN_CONF: Info */
/*   - CONNECT_CONF: Info */
/*   - CONNECT_B3_CONF: Info */
/*   - DISCONNECT_CONF: Info */
/*   - DISCONNECT_IND: Reason */
/*   - DISCONNECT_B3_CONF: Info */
/*   - DISCONNECT_B3_IND: Reason_B3 */
/*   - DATA_CONF: Info */
/*   - DATA_IND: Flags (NUR Error) */
/*   - INFO_CONF: Info */
/*   - INFO_IND: Info-Number bzw. */
/*   Message-Type */
/*   */
/*   char**   cpBuf          */
/*   Zeiger auf den Datenpuffer */
/*   falls vorhanden, SONST = NULL */
/*   - CONN_IND: Calling Party Number */
/*   - DATA_IND: Data */
/*   */
/*   int*     ipLen          */
/*   Anzahl der Zeichen in "cpBuf" */
/*   falls vorhanden, SONST = 0 */
/*   */
/*   IZ03_VB** VB          */
/*   Zeiger auf den Eintrag in der */
/*   VB-Struktur, für den ein CAPI- */
/*   Command als Ergebnis vorliegt, */
/*   falls vorhanden, SONST = NULL */
/*   */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1 */
/*   Änderung:   */
/*   */
/*****

```

```
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */  
/*****
```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      IZ03 FND 1.0+ / CAPI   Stand    :   01.10.98 */
/*****
/*      Datei      :      VB_SORT.C              */
/*      Aufruf     :      IZ03_VB_SORT ( VB )    */
/*****
/*      Beschreibung :                          */
/*      Diese Routine sortiert den Eintrag für die Verbindung "VB" */
/*      chronologisch in die in die VB-Struktur ein.                */
/*      */
/*      aufgerufen von :      IZ03_POLL          */
/*      */
/*      ruft auf      :                          */
/*      */
/*      I/O-Kanäle   :                          */
/*      */
/*****
/*      Eingabe Parameter :                      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      IZ03_VB*  VB      Zeiger auf den Eintrag in der */
/*      VB-Struktur */
/*      ----- */
/*      Ausgabe Parameter:                      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      int      IZ03_VB_SORT      Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      ----- */
/*      Transiente Parameter :                  */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1 */
/*      Änderung: */
/*      ----- */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

;*****/
;*   Produkt   :   GA - Knoten           Version   :   1.2       */
;*   Modulname :   IZ03 Treiber FND 1.0   Stand    :   01.10.98  */
;*****/
;*   Datei    :   COM_xxxx.ASM           */
;*   Aufruf   :   IZ03_COM_xxxx ( Aufrufparameter siehe unten ) */
;*****/
;*
;*   Beschreibung :
;*   Diese Assembler-Quelle enthält alle Routinen-Aufrufe, die zur
;*   Steuerung der seriellen Schnittstellen COM1 - COM4 notwendig
;*   sind.
;*   Die Ein-/Ausgabe von Daten erfolgt zeichenweise über getrennte
;*   Datenpuffer in denen die empfangenen bzw. zu sendenden Daten
;*   zwischengepuffert werden. Die Zeichen im Datenpuffer werden
;*   über Interrupt-Service-Routinen aus dem I/O-Baustein (Intel
;*   8250-kompatibel) gelesen bzw. dorthin geschrieben, so dass eine
;*   Entkopplung des Programmes vom Datentransfer auf der seriellen
;*   Schnittstelle erfolgt. Die spezifischen Interrupts für die
;*   FIFO-Behandlung (z.B. UART 16550) werden unterdrückt, jedoch
;*   können die FIFO's mittels COM_FIFO aktiviert werden.
;*
;*   Gesteuert wird die Übertragung durch ein Hardware-Handshake
;*   über die RTS/CTS-Signale, d.h. diese Signale müssen stets ver-
;*   drahtet sein
;*
;*   Spezifisch für die Anforderungen des FND-SSA der Firma Sauter
;*   wird der Zugriff auf das DTR-Signal) mit der Routine SET_DTR
;*   eingeführt. Die Einhaltung des Protokolls ist in der aufrufenden
;*   Routine sicherzustellen (bzgl. der angepaßten Programmteile
;*   vgl. "mcr").
;*
;*   Grundlage für die Anpassungen ist die "FND-Schnittstellen-
;*   Beschreibung SSA / RS 232" in der Version 1.0 vom 10.5.95 der
;*   Firma SAUTER CUMULUS GmbH in 79108 Freiburg (Autor: H. Melchin;
;*   inkl. nachfolgender telefonischer Abklärungen vom 16.5.95 und
;*   22.6.95)
;*
;*   Hinweis:      Alle Aufruf-Parameter und das Ergebnis der
;*                 Funktion sind vom Typ "int" bzw. "INTEGER*2",
;*                 d.h. sie sind vom Datentyp "WORD".
;*
;*   Hinweis:      Die Schnittstellen-Parameter (Baudrate, Daten-,
;*                 Parity- und Stop-Bits müssen vor Aufruf von
;*                 COM_INIT erfolgen, damit in COM_INIT gemachte Initialisierungen
;*                 nicht zerstört werden !
;*
;*   Hinweis:      Aus programmiertechnischen Gründen sind ALLE in
;*                 dieser einen Routine zusammengefaßt, sodass jede
;*                 Routine nachfolgend detailliert beschrieben werden muß.
;*
;*   Folgende Aufrufe stehen zur Verfügung :
;*   =====
;*
;*   IZ03_COM_INIT   Initialisierung (ERSTE aufgerufene Routine) */
;*   IZ03_COM_EXIT   ENDE-Behandlung (LETZTE aufgerufene Routine) */
;*
;*   IZ03_COM_CIN    Ein Zeichen aus Eingabe-Puffer lesen, falls ex. */
;*   IZ03_COM_COUT   Ein Zeichen an Ausgabe-Puffer übergeben */
;*
;*   IZ03_COM_PURG   Ausgabe-Puffer löschen */
;*   IZ03_COM_STAT   aktuellen Status abfragen */
;*   IZ03_COM_EVNT   aktuellen Status abfragen und Events rücksetzen */
;*   IZ03_COM_SET_DTR Setzen des DTR-Signal für die Modem-Control */
;*****/
;*
;*   Autor      Datum      Version      Kurzzeichen */
;*   -----
;* Original :   W. Fries    01.05.97    1.1      */
;* Änderung:
;*
;*****/
;*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
;*****/

```

**5.4.4.5. Spezifisch für IZ04 (Sondersysteme)**

Im "Standard"-GA-Knoten ist der Treiber IZ04 nicht belegt und bleibt reserviert für die Integration herstellerspezifischer IZ-Treiber (z.B. TREND CNC, Viessmann DEKATEL, EIB, LON, ...) für Sondersysteme. Die zugehörigen Modulbeschreibungen entsprechen an der Aufruf-Schnittstelle denen der IZ-Treiber IZ01 - IZ03.

**5.4.5. Druckausgaben, Meldungstexte**

\*\*\* KEINE \*\*\*

**5.4.6. Datenkatalog**

**IZ01\_LZ.WSL:** Warteschlange der an den FND-SSA auszugebenden CMD-Datagramme aus LZ-Modul

**IZ01\_SP.WSL:** Warteschlange der an den FND-SSA auszugebenden CMD-Datagramme aus SP-Modul

**IZ01POLL.WSL:** Warteschlange der an den FND-SSA auszugebenden CMD-Datagramme für Polling

**IZ01\_ACK.WSL:** Warteschlange der an den FND-SSA auszugebenden ACK-Datagramme

**IZ02\_LZ.WSL:** Warteschlange der an den FND-SSA auszugebenden CMD-Datagramme aus LZ-Modul

**IZ02\_SP.WSL:** Warteschlange der an den FND-SSA auszugebenden CMD-Datagramme aus SP-Modul

**IZ02POLL.WSL:** Warteschlange der an den FND-SSA auszugebenden CMD-Datagramme für Polling

**IZ03Lnnn.WSL:** analog IZ01\_LZ.WSL für IZ-Treiber-Subid. nnn

- IZ03Snnn.WSL:** analog IZ01\_SP.WSL für IZ-Treiber-Subid. nnn
- IZ03Pnnn.WSL:** analog IZ01POLL.WSL für IZ-Treiber-Subid. nnn
- IZ03Annn.WSL:** analog IZ01\_ACK.WSL für IZ-Treiber-Subid. nnn
- IZ03CONN.VBP:** Verbindungs-Profil für Verbindungs-Aufbau von IZ03 zum nächsten GA-Knoten
- IZIDPCMD.WSL:** Warteschlange der an die internen Datenpunkte auszugebenden CMD-Datagramme
- IZCMDERR.WSL:** Warteschlange der bei der Bearbeitung der CMD-Datagramme entstehenden ERROR-Datagramme (Fault-Manager), die dem Absender des CMD-Datagrammes als Ergebnis gesendet werden sollen.

Abhängig vom jeweiligen IZ-Treiber sind zusätzliche dateien einzurichten. Diese werden nachfolgend beschrieben.

#### **5.4.6.1. IZ01 (SMS über ISDN)**

Im Verzeichnis \GAK\IZ\SMS werden die Dateien zur Parametrierung des Versendens von Short-Message-Services (SMS) gespeichert. Unter LINUX können die SMS auch an eMail- oder Fax-Empfänger u.a. versandt werden, indem die zugehörigen Programme über Scripte aufgerufen werden.

Zusätzlich werden parametrisierte eingehende Rufe auf FND-Datenpunkte abgebildet, z.B. um Quittierungen zu melden.

**Wichtiger Hinweis: Dieser Treiber sendet die SMS, eMail usw. evtl. über kostenpflichtige Verbindungen. Durch flimmernde Datenpunkte, falsche Parametrierungen usw. kann es zu häufigem Verbindungsaufbau mit entsprechenden Kosten kommen.**

**Eine weitere Fehlerquelle sind die zum Versenden in den Scripten verwendeten Programme, die ggf. interne Wiederholungsstrategien realisieren und auch Ablauffehler aufweisen können.**

Stellen Sie sicher, dass ALLE aufgerufenen Scripte ggf. nach einer sinnvollen, minimalen Anzahl von Wiederholungsversuchen, spätestens nach 10 Minuten beendet werden. Danach wird der Script als fehlerhaft beendet betrachtet und die zugehörigen Prozesse verbleiben evtl. als "Zombie".

Im Fehlerfall hat das Script einen EXIT-Kode ungleich 0 zu liefern !

Der Nutzer dieser Software hat dafür Sorge zu tragen, dass jedes Fehlverhalten schnellstmöglich erkannt und behoben wird.

Daten-Technik Fries (DTF) übernimmt KEINE Haftung für entstehende Schäden jedweder Art, soweit dieses gesetzlich zulässig ist.

Grundlage für die Realisierung sind :

- Protokollbeschreibung D1: "TAP 97", Version 1.2.0, Stand 9.12.1997
- Protokollbeschreibung D1: "UCP SMSC Spec.", Version 3.1.2, Stand 17.9.1999
- Protokollbeschreibung D2: "SMSC EMI Spec.", Version 2.2, Stand 01.2001
- Protokollbeschreibung E+: "SMPP Prot. Spec.", Version 1.2, Stand 12.10.1999

Die SMS wird über das ISDN des GA-Knotens angeschlossen (CAPI 2.0).

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** SMS: 1
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen SMS-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** SMS: stets 0



4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der SMS-IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	<p>natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** SMS: 0</p>
6	40	42	3	Protokoll	<p>nat. Zahl kodiert wie folgt :</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** stets: 0</p>
7	44	256	<=213	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem SMS/IZ-Typ.</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** stets: leer</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Abgehende Teilnehmer-Rufe werden auf FND-Schaltpunkte abgebildet. Dabei wird jedem FND-Schaltpunkt eine Datei mit spezifischer Extension (.RUF) im Unterverzeichnis IZ/SMS zugeordnet.

Eingehende Ruf-Ereignisse werden auf FND-Meldepunkte abgebildet. Dabei wird jedem FND-Datenpunkt eine oder eine Gruppe von Dateien mit spezifischer Extension (.RER) im Unterverzeichnis IZ/SMS/CIP\_n zugeordnet, wobei die CallingPartyNumber beim CONN\_IND dem Namen der Unterverzeichnisse und der Datei mit der Extension "RER" entspricht, wenn diese zusammengesetzt werden müssen, weil das Betriebssystem die Anzahl der Zeichen im Dateinamen begrenzt (z.B. 12345 => 12345.RER, 01234567 => 01234567.RER und 01234567890 => 01234567890.RER (LINUX), jedoch => 01234567\890.RER (DOS)).

**Hinweis: Die Unterteilung in Unterverzeichnisse MUSS für DOS nach jeweils 8 Zeichen der CallingPartyNumber gemacht werden.**

Dabei entspricht "n" für "CIP\_n" dem CIP-Value gemäß der CAPI-Spezifikation, wie er beim CONNECT\_IND geliefert wird. So ist es z.B. möglich, abhängig vom verwendeten Datendienst, vom gleichen Anschluss aus unterschiedliche Ruf-Ereignisse zu melden. Hierzu kann das FWD-Programm mit dem Schalter "/sn" verwendet werden, mit "n" als CIP-Value, z.B. :

"FWD /s1 /num #<Rufnummer des GA-Knoten>" (CIP=1: Sprachdienst)

"FWD /s2 /num #<Rufnummer des GA-Knoten>" (CIP=2: Datendienst)

**Hinweis: Anrufe von Telefon, Handy usw. liefern den in der TK-Anlage bzw. Vermittlung zugeordneten CIP-Value - i.a. Sprachdienst mit CIP-Value 1 oder 4.**

Wird die RER-Datei im Unterverzeichnis CIP\_n NICHT gefunden, so wird der Versuch direkt auf dem Unterverzeichnis IZ/SMS wiederholt, d.h. die so angelegten RER-Dateien gelten für alle CIP-Value.

Eingehende Rufe werden vom Treiber BEDINGT angenommen, d.h. es entstehen evtl. Verbindungskosten. Wurde der Anrufer ggf. anhand des CIP-Value und der CallingPartyNumber erkannt, so wird Ruf mit CONN\_RES nach ca. 4 Sekunden akzeptiert und sofort abgebrochen. Die in der RER-Datei angegebenen Aktionen werden ausgeführt. Für Sprachrufe (CIP-Value 1, 4, ...) ergeben sich dabei folgende Signaltöne :

- nach ca. 3-4 Sekunden "besetzt": Abweisung nach erfolgreicher Bearbeitung
- sofort "besetzt": Der Anschluss ist wirklich besetzt bzw. aus anderen Gründen nicht erreichbar.
- andauerndes "Klingeln": Anrufer ist nicht für das Melden von Ruf-Ereignissen parametrier, d.h. die RER-Datei fehlt
- Anruf wird angenommen: Ein anderes Programm-Modul (LZ, IZ03, FWD, ...) oder ein anderer Teilnehmer am S0-Bus akzeptiert den Anruf (z.B. CallingPartyNumber aus ANRUFER.CHK entfernen)

Die Dateiformate sind allgemeingültig ausgelegt, so dass eine spätere Anpassung bzw. Erweiterung einfach möglich ist.

SMS-Funktion	Extension	FND-Schaltpunkt
Teilnehmer-Ruf	.RUF	max. 8-stufig
Ruf-Ereignis	.RER	max. 8-stufig

Abhängig vom zugeordneten FND-Attribut wird die Anzahl der Stufen festgelegt. Den zugehörigen FND-Indizes wird in der RUF-Datei zeilenweise der zugehörige Parametersatz zugeordnet, falls dem FND-Index keine programmspezifische Funktion zugeordnet ist. Die Reihenfolge der Einträge in den Dateien ist frei. Kommentarzeilen sind durch ein "!" am

Zeilenanfang gekennzeichnet. Ist dem FND-Index 0 in der Datei keine Funktion zugeordnet, so wird ggf. eine programmspezifische Funktion wie folgt zugeordnet:

SMS-Funktion	Index 0
Teilnehmer-Ruf	autom. über alle Indizes, bis Erfolg

Die RUF-Datei hat folgenden Zeilenaufbau:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Index	FND-Index (000 - 008)
2	5	29	25	T-Adresse	Teilnehmer-Adresse bestehend aus Prefix und Anschlussnummer des B-Teilnehmers (Tel.Nr. des Anzurufenden, z.B. 491603821144) oder symbolische Adresse als Verweis auf Eintrag im Adressbuch
3	31	190	<=160	Nachricht	Auszugebende Nachricht mit Steuerzeichen gemäß ANSI-C SMS: max. 160 darstellbare Zeichen

Jede Teilnehmer-Adresse für SMS-Dienste beginnt mit einem Prefix, bestehend aus der Länderkennung - i.a. 49 - und der Vorwahl des Providers OHNE führende Nullen - z.B. 171. Dieser dient als Schlüssel für die Zuordnung des zugehörigen Short-Message-Service-Centers (SMSC) des Providers. Die Rufnummer des SMSC wird dem Prefix in der Datei mit den Verbindungs-Profilen (VBP) zugeordnet.

Für eMail- und Fax-Dienste kann der jeweilige Prefix freigewählt werden, MUSS jedoch eindeutig sein (z.B. "Mail:" und "Fax:").

Ist die Zeichenfolge bestehend aus Prefix und Anschlussnummer als symbolische Adresse in der Datei für das Adressbuch (ADR\_BUCH.TXT) definiert, so wird die Zeichenfolge durch die dort angegebene Teilnehmer-Adresse ersetzt. Diese kann dann bis zu 228 Zeichen lang sein - z.B. für eMails.

Die ADRESSBUCH-Datei hat folgenden Zeilenaufbau:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	25	25	symp. Adresse	Symbolische Adresse
2	27	255	<=228	T-Adresse	Teilnehmer-Adresse wie in der Ruf-Datei. Für eMail's können ggf. mehrere Adressen in dem Format angegeben werden, wie sie beim Versenden im Script behandelt werden

Stellen die ersten 16 Zeichen der Nachricht eine Datenpunkt-Adresse dar, so wird die Nachricht bestehend aus der Objekt-Kennung, -Beschreibung, Datenpunkt-Adresse, -Beschreibung und der Meldung (Attributtext bzw. Wert mit Dimension) automatisch erzeugt und die nachfolgenden Zeichen angefügt.

Ist die Nachricht leer, so wird der Ruf an das SMSC unterdrückt und im RSP-Datagramm das tl-Bit (temporäre Verriegelung) gesetzt. Dadurch ist möglich, Nachrichten für einzelne FND-Indizes (z.B. Rücksetzen auf NORMAL) zu unterdrücken.

Abhängig vom zugeordneten FND-Attribut wird die Anzahl der Stufen festgelegt.

Den zugehörigen FND-Indizes wird in der RER-Datei der zugehörige Befehl - i.a. Meldungsstufe aus den Werten 0, 1, 2, 4 usw. - aus der ERSTEN gelesenen RER-Zeile zugeordnet. Kommentarzeilen sind durch ein "!" am Zeilenanfang gekennzeichnet.

Die RER-Datei hat folgenden Zeilenaufbau (RER-Zeile):

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	16	16	dp_id	Adresse des FND-Meldepunktes der diesem Treiber zugeordnet sein MUSS.
2	18	20	3	Datenfeld	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 29).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 011 für "actual"
3	22	35	14	Befehl	Meldungsstufe, Befehl bzw. (Vorgabe)-Wert abhängig vom Datenfeld (z.B. 1)
4	37	41	5	RESET-Zeit	Wartezeit in Sekunden bis zur Ausführung des RESET-Befehls >0: Reset nach ca. 4 Sek. (z.B. 4) Sonst: KEIN Reset
5	43	56	14	RESET-Befehl	Meldungsstufe abhängig vom Datenfeld für RESET nach Ablauf der RESET-Zeit (z.B. 0) Falls LEER, RESET auf Wert vor dem Ruf-Ereignis.

In den RUF-Dateien werden somit die Rufnummern der Teilnehmer angegeben. Der Prefix, d.h. die ersten signifikanten Ziffern entscheiden über den zu verwendenden Provider. Jeder Provider betreibt ein eigenes SMS-Center mit i.a. eigener Rufnummer und Protokoll. Über die Datei IZnnSMSC.VBP - unter LINUX in Kleinbuchstaben - wird nun jedem Prefix das zugehörige Verbindungs-Profil (VBP) zugeordnet.

**Hinweis: Eine Provider-übergreifende Weiterleitung vom SMS - z.B. über das D2-SMSC an einen ePlus-Teilnehmer wird derzeit von den Providern NICHT unterstützt. Der richtigen Zuordnung von Prefix und Verbindungs-Profil kommt somit entscheidende Bedeutung zu.**

Die VBP-Datei hat folgenden Zeilenaufbau:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	10	10	Prefix	Prefix für SMS bestehend aus Länderkennung und Vorwahl der Teilnehmer-Adresse, sonst frei wählbar, aber eindeutig
2	12	14	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  ISDN/CAPI: 020 SCRIPT : 200 (nur LINUX)
3	16	18	3	Protokoll	nat. Zahl kodiert wie folgt : 0: TAP 97 (D1) 1: UCP (D1) 2: UCP (D2) 3: TAP (ePlus)  Für Scripte wird dieser Wert zwar NICHT programmtechnisch ausgewertet, jedoch als Parameter beim Aufruf des Scriptes übergeben. Er kann somit frei vergeben und im jeweiligen Script ausgewertet werden.
4	20	44	25	Rufnummer/Script	Für SMS: Rufnummer des SMSC bestehend aus Amtsholung, Vorwahl und Anschlussnummer des SMSC. Für eMail, Fax u.a.: Name der Script-Datei zum Versenden der Message

Wurde das Versenden über ein Script angegeben, so wird die Message in der Datei "IZnn\_SMS.MSG" abgelegt ("nn" = Id. des IZ-Treibers, hier "01"). Zusätzlich werden die Dateien "IZnn\_SMS.TNA" und "IZnn\_SMS.ABS" erzeugt, die die Teilnehmer-Adresse(n) des/der Empfänger und Angaben zum sendenden GA-Knoten enthalten. Für eMail's kann die ABS-Datei z.B. als Identifikation in der Betreff-Zeile verwendet werden, wenn mehrere GA-Knoten den gleichen Provider-Zugang verwenden.

Ist die Teilnehmer-Adresse der Name einer Datei im Unterverzeichnis IZ/SMS, so wird diese Datei auf "IZnn\_SMS.TNA" kopiert. Damit ist es möglich, ganze Verteilerlisten im Script abzuarbeiten. Je Teilnehmer-Adresse stehen durchschnittlich 60 Sekunden für das Versenden über das Script zur Verfügung.

Der Aufruf der Script-Datei zum Versenden der Message erfolgt mit folgenden Parametern:

- Verbindungs-Protokoll gemäß verwendetem Eintrag aus IZnnSMSC.VBP
- Name der TNA-, MSG- und ABS-Datei OHNE Extension, d.h. "IZnn\_SMS"

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	40	12	SMS-Adresse	<p>RUF-Datei mit oben genannter Extension für abgehende Rufe (z.B. H01P14.ruf mit "gaamm" gemäß GLT-Richtlinien, hier Pumpen-Störung)</p> <p>RER-Datei mit oben genannter Extension für eingehende Rufe (Ruf-Ereignisse)</p> <p>*** Wird für RER-Dateien *** noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** stets "*.rer"</p>

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Es wird empfohlen, jedem Schaltungspunkt eine eigene Datei zuzuordnen. Dadurch besteht die Möglichkeit bei GA-Knoten mit Mehrbenutzer-Betriebssystem (z.B. LINUX) unter dem Programmlauf die Dateien auszutauschen. Hierzu wird empfohlen, den Dateinamen in Anlehnung an die Datenpunkt-Adresse zu vergeben (z.B. "gaamm" mit "g", "aa" und "mmm" gemäß der GLT-Richtlinien).

LINUX unterscheidet Dateinamen nach Groß-/Kleinschreibung. Aus diesem Grund MÜSSEN alle hier angegebenen Dateinamen so angegeben werden, dass sie auch in der Schreibweise mit den Namen der Dateien übereinstimmen. Die Verzeichnisse sind in Kleinbuchstaben einzurichten ( z.B "iz/sms")

**Wichtiger Hinweis:** Das SMSC von ePlus setzt derzeit spezifische Modems ein, die einen kombinierten Analog-/ISDN-Betrieb erlauben. Dabei kommt es zu Unverträglichkeiten mit einigen bei der LHM eingesetzten ISDN-Karten wie z.B. :

- Eicon Diehl SXN/SCOM

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_3_RT 2 0 15 15 15 0 0 015 045 Ruf-Ereignis          dddddd.rer
00002 Schaltpunkt_3_RT 2 0 15 15 15 0 0 015 008 Teilnehmerruf      dddddd.ruf
...
```

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

**5.4.6.2. IZ02 (AEG/Schneider Groupe Modicon)**

\*\*\* KEINE \*\*\*

**5.4.6.3. IZ03 (FND über ISDN bzw. Netzwerk)**

Folgende Dateien stehen im Verzeichnis \GAK\IZ zur Verfügung :

IZ03CONN.VBP

Die in den Dateien "**IZ03CONN.VBP**" abgelegten Verbindungs-Profile steuern den Verbindungsaufbau des IZ03-Treibers zum nächsten GA-Knoten.

Die editierbaren Text-Dateien werden im Unterverzeichnis \GAK\IZ abgelegt und enthalten das Verbindung-Profil im folgenden Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	lfd. Nummer	beginnend mit 001
2	4	5	3	"::"	Trennzeichen (OHNE "'')
3	7	9	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  Zusätzlich gilt : 40: UDP/IP über Netzwerk
4	11	13	3	Verbindungs-Prot.	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95

(vgl. Seite 21). Für die LZH ist "FND gemäß FND-Spezifikation mit FND-Erweiterungen" verbindlich festgelegt (d.h. 002).

\*\*\* Wird noch nicht ausgewertet  
\*\*\* und bleibt reserviert für  
\*\*\* zukünftige Erweiterungen !

\*\*\* Bis dahin gilt :  
\*\*\* LZH: 2=FND mit Erweiterungen  
\*\*\* BWD-GEFAHR: 10=TSS 13a  
\*\*\* BWD-EINBRUCH: 10=TSS 13a

5	15	19	5	Verbindungs-Dauer	Dauer in Sekunden, die die Verbindung nach der letzten Datenübertragung aufrecht erhalten werden soll (z.B. zum Warten auf RSP)
6	21	80	<=60	Verbindungs-Adr.	abhängig von der Verbindungs-Art wird die für den Verbindungsaufbau notwendige Adresse angegeben (z.B. Rufnummer)  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>), je nach verwendetem Editor) abgeschlossen.

#### 5.4.6.4. Spezifisch für IZ04 (Sondersysteme)

Im "Standard"-GA-Knoten ist der Treiber IZ04 nicht belegt und bleibt reserviert für die Integration herstellerspezifischer IZ-Treiber (z.B. TREND CNC, Viessmann DEKATEL, EIB, ...) für Sondersysteme. Die zugehörigen Dateien entsprechen denen der IZ-Treiber IZ01 - IZ03.

### 5.4.7. Definition der Datenübertragung

Der GA-Knoten ist über die EICON-Card, die ISDN-Karte bzw. über serielle Schnittstellen (V.24 / RS-232) mit den nachfolgenden IZ verbunden

**Hinweis:** Für Sondersysteme können abweichende Festlegungen gelten. Beachten Sie bitte entsprechende Festlegungen unter "IZnn-Treiber für Sondersysteme".

#### 5.4.7.1. FND 1.0 über X.25

Diese IZ ist über einen X.21/V.24-Umsetzer an die im GA-Knoten unter DOS integrierte EICON-Card/PC angeschlossen. Auf den OSI/ISO-Schichten 1 - 3 wird das X.25-Protokoll realisiert. Dazu wird als



Programmierschnittstelle das "X.25 Network-Level Developer's Toolkit" der Firma EICON verwendet. Bei Start des GA-Knotens muß zusätzlich der passend konfigurierte "OSI PC Gateway" der Firma EICON als Treiber geladen werden. Auf OSI/ISO-Schicht 7 wird das FND-Protokoll in der Version 1.0 realisiert. Die Funktion zur Übergabe der Systemzeit entfällt, da sie von dem FND-SSA nicht unterstützt wird.

**Hinweis:** Das X.25-Protokoll ist momentan deaktiviert

#### **5.4.7.2. FND 1.0 über V.24**

Diese IZ ist an eine der im GA-Knoten integrierten COM-Schnittstellen angeschlossen. Auf den OSI/ISO-Schichten 1 - 3 wird das V.24-Protokoll realisiert, wobei die Datensicherung über die im UART-Baustein existierenden Status-Flags für Framing, Parity- und Overrun-Fehler erfolgt. Vor dem Start des GA-Knotens müssen die Schnittstellen-Parameter mittels des MODE-Befehl gesetzt werden und ggf. der FIFO mittels COM\_FIFO aktiviert werden. Auf OSI/ISO-Schicht 7 wird das FND-Protokoll in der Version 1.0 realisiert. Die Funktion zur Übergabe der Systemzeit entfällt, da sie von dem FND-SSA nicht unterstützt wird.

#### **5.4.7.3. FND 1.0 über ISDN bzw. Netzwerk**

Bis zu 256 IZ's sind über das ISDN an die im GA-Knoten integrierte ISDN-Karte angeschlossen. Auf den OSI/ISO-Schichten 1 - 3 werden für ISDN als B-Kanal-Protokolle B1: 64kBit/s mit HDLC, B2: ISO 7776 (X.75 SLP) und B3: ISO 8208 (X.25 DTE-DTE) realisiert. Dazu wird als Programmierschnittstelle das COMMON-ISDN-API (CAPI) in der Version 2.0 verwendet. Bei Start des GA-Knotens muß zusätzlich der passend konfigurierte Treiber und die CAPI 2.0 des Herstellers der ISDN-Karte geladen werden. Auf OSI/ISO-Schicht 7 wird das FND-Protokoll in der Version 1.0 realisiert. Die Funktion zur Übergabe der Systemzeit entfällt, da sie von dem FND-SSA nicht unterstützt wird. Die Rufnummern der IZ's müssen in den Dateien ANRUFER.CHK und IZ03CONN.VBP eingetragen sein.

Unter LINUX steht zusätzlich das UDP/IP-Protokoll zur Verfügung. Das FND-Datagramm wird als ein Paket innerhalb des UDP/IP-Rahmens versandt.

**5.4.7.4. AEG/Schneider Groupe Modicon**

Diese SPS ist an eine der im GA-Knoten integrierten COM-Schnittstellen angeschlossen. Auf OSI/ISO-Schicht 7 wird das zugehörige Protokoll gemäß den KS-Funktionen realisiert (vgl. AEG/Schneider Groupe A120 KS-Funktionen, Benutzerhandbuch, Dok.-Nr.: A91M.271974.20-0791). Dieses Protokoll enthält auch Bestandteile für die Datensicherheit, wobei zusätzlich die im UART-Baustein existierenden Status-Flags für Framing, Parity- und Overrun-Fehler ausgewertet werden.

Bei Systemen mit der ALU200/202 traten Probleme mit deren Pufferverwaltung auf, bei denen beim Lesen eines Pufferbereiches die Antwort für den jeweils vorherigen Befehl geliefert wurde. Gemäß den ergänzenden Angaben der Fa. Schneider Groupe wurde eine Synchronisations- und Initialisierungsfunktion ergänzt, die nach einer Störung das korrekte Aufsetzen sicherstellen soll.

**5.4.8. IZnn-Treiber für Sondersysteme**

Ergänzend zum GA-Knoten wurden weitere IZ-Treiber realisiert, die in den reservierten IZ04-Treiber eingebunden werden können. In einzelnen Fällen stehen diese auch als Treiber für IZ01 - IZ03 zur Verfügung, wie nachfolgende Auflistung zeigt:

Nachfolgend werden die IZnn-Treiber für Sondersysteme mit ihren spezifischen Eigenschaften und Parametrierungen beschrieben. Dabei kann es insbesondere für den Aufbau der IZ-Zeile, der IS-Zeile und die Datenübertragung zu spezifischen Festlegungen kommen.

**Hinweis:** Soweit herstellerspezifische Systeme hier nicht beschrieben wurden, sind die Informationen der spezifischen Systemspezifikation des jeweiligen Herstellers zu entnehmen.

**5.4.8.1. Europäischer Installationsbus (EIB)**

Grundlage für die Realisierung sind folgenden Dokumente :

- "EIB-Handbook" Issue EIB 2.21 (insbes. Kap. 8.7)

**Hinweis:** Momentan werden nur Melde-, Schalt- und Messpunkte unterstützt, auch wenn nachfolgend bereits Definitionen für Stell- und Zählpunkte vorgenommen werden.

Lesebefehle auf EIB-Datenpunkte können zu unkontrollierten Schaltvorgängen führen, da EIB-Aktoren die Antwort auf einen EIB-Lesebefehl wie einen EIB-Schreibbefehl behandeln ! Bei der Projektierung/Ausschreibung ist explizit die Unterbindung dieses Effektes ("GA-Eignung") zu fordern.

Das EIB-System wird über einen EIB-Schnittstellen-Adapter an eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	EIS-Typ	vgl. EIB-Handbook 4.7.1 ff. Folgende Werte sind zulässig: 001: Melde- und Schaltpunkte 006: Mess-, Stell- und Zählpunkte
6	33	41	9	BLT-Adr	Bereichs-, Linie und Teilnehmer-Adresse im Format BB.LL.TTT
7	43	49	7	GA_actual	Gruppenadresse, die bei Änderung des Sensorzustands ("actual") angesprochen wird bestehend aus Haupt- und Untergruppen-Adresse im Format HH.UUUU

**Hinweis:** Für Schaltpunkte OHNE Rückmeldung und für Stellpunkte wird der Vorgabewert ("nominal") an GA\_actual ausgegeben.

Optional lassen sich folgende Parameter abhängig vom DP- und EIS-Typ angeben:

- Schaltpunkte mit Rückmeldung

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
8	51	57	7	GA_nominal	Gruppenadresse, die zur Änderung des Aktorzustands ("nominal") angesprochen wird bestehend aus Haupt- und Untergruppen-Adresse im Format HH.UUUU

Ist diese Gruppenadresse angegeben, so handelt es sich um einen Schaltpunkt, MIT Rückmeldung, d.h. das USM-Datagramme für Zustandsänderungen werden nur abhängig von dessen spontaner Meldung mit GA\_actual erzeugt. Ist

diese Gruppenadresse NICHT angegeben, so handelt es sich um einen Schaltpunkt OHNE Rückmeldung, d.h. der Schaltbefehl wird an GA\_actual gesendet, die sofortige Ausführung des Schaltbefehls angenommen und das zugehörige RSP-Datagramm mit der Zustandsänderung erzeugt. Das USM-Datagramm wird erzeugt, falls für einen EIB-internen Schreibbefehl kein zugehöriger Datenpunkt (Teilnehmer- und Gruppenadresse des Absenders) gefunden wird.

**Hinweis:** Schaltpunkte, die nicht sofort schalten (z.B. Zeitverzögerung, Priorität, Verknüpfung oder die auf mehrere Gruppenadressen reagieren, MÜSSEN explizit eine Rückmeldung erzeugen.

- Mess-, Stell- und Zählpunkte mit Zuordnung einer Kennlinie

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
8	51	55	5	KL_Id	Kennlinie

- Stellpunkte mit "Rückmeldung"

Diese Funktion ist in Anlehnung an die Schaltpunkte mit Rückmeldung hier vorbereitet für analoge Aktoren, die bei einem Lesebefehl bzw. spontan ihren aktuellen Wert liefern, unabhängig davon, welcher Vorgabewert definiert ist.

**Hinweis:** Durch Angabe von "GA\_nominal == GA\_actual" wird verhindert, dass bei einem Stellbefehl im RSP-Datagramm der "nominal" in den "actual" übernommen wird.

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
9	57	63	7	GA_nominal	Gruppenadresse, die zur Änderung des Aktorzustands ("nominal") angesprochen wird bestehend aus Haupt- und Untergruppen-Adresse im Format HH.UUUU

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt          001 01.01.001 01.0001
00002 Schaltpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schaltpkt.          001 01.01.002 02.0001
00003 Schaltpunkt_2mR___ 2 0 15 15 15 0 0 015 045 Klartext für SP + Rückm.          001 01.01.003 02.0002 02.0003
00004 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt          006 01.01.004 03.0001 00000
00005 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt          006 01.01.005 04.0001 00000
00006 Stellpunkt_2mR___ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt          006 01.01.005 04.0001 00000
04.0002
00007 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt          006 01.01.006 05.0001 00000
...
```

Die Grenzwert-Überwachung und die Ausblendung der Betriebsmeldungen wird vom IZ-Treiber emuliert .

**5.4.8.2. Local Operating Network (LON)**

Grundlage für die Realisierung des IZ-Treiber sind :

- bereitgestellte Dokumentation der Fa. ECHELON
- Spezifikation "Anschluß des LON an den GA-Knoten"
- Angebot LHMMFUNK/8011 vom ??.01.98
- Angebot LHMMFUNK/2122 vom 11.12.2002

Der LON wird über einen SLTA/2 o.ä. an eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

Unter LINUX und generell ab Version 1.3a muss, abweichend von der Spezifikation "Anschluß des LON an den GA-Knoten", als Link-Protokoll statt "ALERT/ACK" nun "Buffered" am SLTA eingestellt werden.

**Hinweis:** Der SLTA ist nur halbduplex-fähig, unterstützt aber KEIN stabiles Handshake-Verfahren. Stattdessen erwartet er, dass der Host-Rechner seinen Sendevorgang unterbricht, sobald der SLTA sendet. Aufgrund der bei Mehrbenutzersystemen üblichen Latenzzeiten (LINUX: typ. 20 ms) kann es zu "Kollisionen" der gesendeten Datagramme kommen, in deren Verlauf der SLTA das vom Host gesendete Datagramm "vergisst".

Da in diesem Fall seitens des SLTA keine Fehlermeldung erfolgt, kann der Host das Problem nur durch das Ausbleiben der Bestätigung nach Ablauf einer Wartezeit erkennen.

Auch der von ECHELON in einem Beispiel-Treiber vorgeschlagene Workaround reduziert nur die Häufigkeit, beseitigt aber das Problem nicht grundlegend. Als Folge davon können sporadisch Verzögerungen von einigen Sekunden und die Wiederholung von Datagrammen auftreten.

Eine Verbesserung schafft die SLTA-Emulation der Fa. MSR electronic GmbH, Pocking. Diese unterstützt alle für diesen Treiber erforderlichen Datagramme und puffert bis zu 128 Zeichen zwischen, so dass keine Zeichenverluste mehr beobachtet wurden.

In Verbindung mit der Dezentralen Erfassungs- und Melde-Anlage (DEMA) wird das SLTA-Protokoll - soweit für diesen Treiber erforderlich - direkt im LON-Knoten mit serieller Schnittstelle umgesetzt. Für diese Konfiguration wird

auch die Anbindung über Modem mit Hayes AT-Sequenzen unterstützt. Dabei ist das Modem so einzustellen - vgl. IZ-Zeile -, dass

- das sicherste Übertragungsverfahren (MNP5, V42.bis) bei  $\geq 9600$  Baud automatisch vereinbart wird, so dass Übertragungsfehler direkt erkannt und behoben werden
- auch bei Nebenstellen der Verbindungs-Aufbau sichergestellt ist - z.B. kein Warten auf das Freizeichen
- Rückkehr aus dem Daten- in den Befehls-Modus durch "+++" erfolgt

Folgende Einstellungen werden vom IZ-Treiber erzwungen:

- Rückmeldungen als Text (OK, ERROR, NO ..., RING, CONNECT) ausgegeben werden (ATQ0)
- eingehende Rufe (RING) werden nur explizit angenommen (ATS0=0, so dass der IZ-Treiber ATA senden muss)

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben.  GAK_XXL: 6, ..., 12
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0. Für die direkte Verbindung zum GA_Knoten über RS 232 ist dieser Wert stets 0  Für jede Modem-Verbindungen ist ein gesonderter Eintrag mit fortlaufender Nummer zu erstellen.
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung des LON bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "Leistungsverzeichnis eines GA-Knotens" vom 29.8.94 (vgl. Seite 20).  001: direkt V.24 mit Buff.-IO 010: Modem mit AT-Satz sonst: direkte V.24 mit ALERT/ACK unter DOS bzw. Buff.-IO unter LINUX  Hinweis: Bei Modem-Betrieb sind ===== dieser und alle folgenden Werte ohne Bedeutung, sollten aber

					analog zur Treiber-SubId 0 angegeben werden.
6	40	42	3	Protokoll	<p>nat. Zahl kodiert wie folgt :</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** LON: 0</p>
7	44	46	3	Subnet_SK	Subnet-Adr. des System-Knotens
8	48	50	3	Node_SK	Node-Adr. des System-Knotens Falls "Subnet_SK" und "Node_SK" auf 0 gesetzt sind, wird die Behandlung des System-Knotens unterdrückt
9	52	256	<=205	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.</p> <p>"COM1" - "COM4" (ohne " bzw. "/dev/ttyS0" ff.</p> <p>Für den Modem-Betrieb (Verb.- Art = 010) folgt - durch ein Leerzeichen getrennt - die Initialisierungs-Sequenz für das Modem - OHNE führendes "AT"</p> <p>Wird durch das Zeilenabschluß- zeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Gemäß der Spezifikation "Anschluß des LON an den GA-Knoten" werden die FND-Datenpunkte innerhalb der LON-Knoten umgesetzt, d.h. es werden FND-Datagramme mit einem reduzierten APCI im Datenbereich des LON-Datagrammes übertragen.

Empfangene "atb\_id"/"dimension" werden durch die in der Konfigurationsdatei angegebenen Werte ersetzt. Optional kann für Melde- und Schaltpunkte eine Umrechnungs-Funktion zur Invertierung von "actual" und "nominal" angegeben werden.

Die LON-Adressen werden in den IS-Zeilen wie folgt kodiert:

"[D.]SSS.NNN:III" mit :

D ...: Index in Domain-Table (optional, Voreinstellung 0)

SSS .: Subnet

NNN .: Node

III .: Index des Datenpunktes im LON-Knoten

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Für Modem-Verbindungen ist die Treiber-SubId explizit so anzugeben, dass der Datenpunkt der zugehörigen IZ-Zeile zugeordnet wird.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Melde-, Schalt-, Mess-, Stell- und Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	LON-Adresse	LON-Adresse gemäß obigem Format
6	43	47	5	UR_Fkt/KL_Id	Für Melde- und Schaltpunkte: Nummer der Umrechnungsfunktion des LON-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig)  Für Mess-, Stell- und Zählpunkte: ** Momentan nicht realisiert ** Nummer der Kennlinie, die zur Umrechnung der physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei :

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt D.SSS.NNN:III URFkt
00002 Schalterpunkt_1____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt. D.SSS.NNN:III URFkt
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt D.SSS.NNN:III KL_Id
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt D.SSS.NNN:III KL_Id
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt D.SSS.NNN:III KL_Id
...
    
```

Der Zugriff auf den LON wird beim Programmstart und nach jeder erkannten Störung neu initialisiert. Dabei erfolgt ein RESET, FLUSH\_CANCEL, SET NODE MODE und Abfragen der Neuron-Id.



Im Modem-Betrieb (Verb.-Art = 010) wird die Initialisierungs-Sequenz aus der IZ-Zeile für Treiber-SubId zyklisch an das zugeordnete Modem gesendet. Dadurch wird dessen Bereitschaft geprüft und ggf. neu initialisiert.

Nach jedem Verbindungs-Aufbau wird die LON-Initialisierung (RESET ff.) durchgeführt.

**Hinweis:** Die bisher eingesetzten Modems wiesen im Langzeittest Probleme auf, die durch geeignete Wahl der Initialisierungs- und Wähl-Sequenz teilweise behoben werden konnten. Insbesondere nach hersteller-spezifischen Fehlermeldungen wie "SYSTEM SUSPEND" oder "BLACKLISTED" könnte ein Neustart des Modems erforderlich sein.

Im Rahmen der vorbereitenden Maßnahmen ist ein geeignetes, langzeitstabiles Modem auszuwählen und mit dem TK-Anschluss als auch der Gegenstelle abzustimmen - z.B. Behandlung des Freizeichens bei Nebenstellen.

Jeder Datenpunkt ist durch die Treiber-SubId in der IS-Zeile einer der Modem-Verbindung zugeordnet. Die FND-Befehle werden für jede Modem-Verbindung in gesonderten Warteschlangen verwaltet.

Für nicht leere Warteschlangen wird nacheinander versucht die Modem-Verbindung aufzubauen, indem aus der Datei "IZnnCONN.VBP" im Unterverzeichnis "IZ" der zur Treiber-SubId gehörige AT-Befehl gelesen wird (nn=Treiber-Id, ggf. mit führender 0).

IZnnCONN.VBP hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	5	5	"nnn::"	Treiber-SubId als lfd. Nummer, beginnend mit nnn=000
2	7	9	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "Leistungsverzeichnis eines GA-Knotens" vom 29.8.94 (vgl. Seite 20).  010: Modem mit AT-Satz
3	11	13	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** LON: 000
4	15	19	5	Wartezeit	Wartezeit vor Abbruch der Verbindung in Sekunden  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** LON: 00010

5	21	112	<=92	AT-Sequenzen	<p>Zeichenfolge für den Aufbau der Verbindung zum entfernten Partner (z.B. DEMA) - OHNE führendes "AT" (z.B. "DT" mit zu wählender Rufnummer)</p> <p>Optional folgen - durch Leerzeichen voneinander getrennt - die AT-Sequenzen für Modem-Initialisierung und Verbindungs-Aufbau durch die DEMA - analog zu den Angaben in der IZ-Zeile und oben, d.h. OHNE führendes "AT". Diese werden ggf. nach jedem Verbindungs-Aufbau an die DEMA gesendet.</p> <p>Hinweis: Jede der 3 AT-==== Sequenzen darf jeweils 30 Zeichen NICHT überschreiten !!!</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>
---	----	-----	------	--------------	---

Bei jeder abgehenden Modem-Verbindung ist die zugehörige Treiber-SubId bekannt. Die bei der Initialisierung abgefragte Neuron-Id. des SLTA vor Ort wird in der Datei "IZnnNRID.LON" der Treiber-SubId zugeordnet.

Für eingehende Rufe wird der Anrufer anhand der abgefragten Neuron-Id. und dem zugehörigen Eintrag in dieser Datei identifiziert und somit der Treiber-SubId zugeordnet.

IZnnNRID.LON hat folgenden Aufbau :

"nnn: iiiiiiiiii" mit "nnn" lfd. Nummer und "ii...i" der Neuron-Id. (hex), sonst "FFFFFFFFFFFF"

Eingehende als auch abgehende Rufe werden in der Error-Log-Datei (vgl. EL-Zeile in GAK.CFG) mit Angabe der Treiber-SubId und der Verbindungs-Stadien protokolliert.

**Wichtiger Hinweis:** Da Spontan-Meldungen zum sofortigen Verbindungs-Aufbau durch die DEMA führen, kann z.B. ein Flimmern erhebliche Kosten verursachen. Empfohlen wird eine regelmäßige Kontrolle der Error-Log-Datei.

Ergänzend zur Sammelstörmeldung für die IZ (interner FND-Schaltpunkt "%IZmmnnSTAT-oooo" mit mm: Treiber-Id., nn: Treiber-Subid., oooo: Objekt-Kennung) wird der Status der einzelnen LON-Knoten in der Datei "izmmstat.lon" (mm: wie oben) und als interner FND-Schaltpunkt "%IZmm-d:sss.kkk" (mm: wie oben, d: Domain-Table-Index, sss: Subnet mit führenden Nullen, kkk:Knoten mit führenden Nullen) abgebildet.

Die internen FND-Schaltpunkte müssen in der Konfigurations-Datei mit atb\_id = 27 (NORMAL/STÖRUNG) definiert sein. Der Klartext zur Beschreibung sollte Aufschluss z.B. über Funktion und Einbauort geben.

Zyklische Abfragen von Datenpunkten (Polling), die keiner Protokollierung zugeordnet sind (lz\_subid 1) werden im Modem-Betrieb nur bei bestehender Verbindung behandelt. Sonst werden sie mit dem letzten gelesenen Wert beantwortet, um ggf. kostenpflichtige Verbindungen zu vermeiden, wie sie z.B. bei der Verwendung von Standard-Parametern mittels GEN\_CFG auftreten wurden.

Zyklische Abfragen für die Belegung der Anlagenbilder (IWA) führen jedoch zum Verbindungs-Aufbau.

#### **5.4.8.3. CC600 (Fa. RAM)**

Grundlage für die Realisierung sind :

- "CC 600 Schnittstellen", Version vom 15.11.93
- "EXLIS, Plausibilitätstabellen, Texte"
- Für Melde- und Schaltpunkte entfällt die Umsetzung der "atb\_id" abhängig von der "byTextNumber". Diese wird grundsätzlich der FND-Referenzdatei entnommen.
- Ein Zeitabgleich der CC 600 anhand der ISDN-Zeit erfolgt nun doch.

Die CC 600 wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

Die CC 600 wird beim Programmstart und nach jeder erkannten Störung neu initialisiert. Dabei erfolgt die Anmeldung ("byOpCode = ASERVICE) mittels dem in der IZ-Zeile definierten Passwort und der Kundennummer.

Die CC 600 kann NICHT spontan melden, so dass auch Melde- und Schaltpunkte zyklisch abgefragt werden müssen (Polling), um Änderungen im "actual" zu erkennen. Der Projektant hat für jeden Melde- und Schaltpunkt das

Abfrageintervall explizit so anzugeben, dass eine Änderung ausreichend schnell erkannt werden kann, ohne das CC600-System zu überlasten.

Der IZ-Treiber prüft vor jedem WRITE auf die CC 600 die Modifikations-Berechtigung. Die WRITE-Zugriffe auf die einzelnen Datenpunkte durch die CC 600 unterliegen keiner Kontrolle, d.h. die durch FND gesetzte Sperrpriorität wird nicht unterstützt.

Für Mess- und Zählpunkte können optional CC600-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die CC600-Adressen werden ca. 10-mal / Abfrageintervall gepollt und geprüft. Eine Behandlung der vom CC600 erzeugten Grenzwert-Alarmer erfolgt nicht. Bei Bedarf können diese jedoch gesondert als Meldepunkte definiert werden.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 108, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen DDC-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der DDC-IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !

					<p>*** Bis dahin gilt : *** CC 600: 0</p>
6	40	42	3	Protokoll	<p>nat. Zahl kodiert wie folgt :</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** CC 600: 0</p>
7	44	47	4	Passwort	Passwort für die Anmeldung am CC 600
8	49	53	5	Kundennummer	Kundennummer für die Anmeldung am CC 600
9	55	256	<=202	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.</p> <p>"COM1" - "COM4" (ohne ")</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die CC600-Adressen werden in den IS-Zeilen wie folgt kodiert:

"abcdefg½" mit :

ab ...: Anlage

cdefg : Code

½ .....: Wert

Dabei sind "a" bis "f" aus dem Zeichenvorrat "0-9", "A-E", während für den Wert "1" oder "2" anzugeben ist.

Für Mess-, Stell- und Zählpunkte kann optional eine Kennlinien zugeordnet werden. Dieses ist notwendig, da das CC600-System teilweise von der FND-Spezifikation abweichende Dimensionen verwendet.

Für Mess-, Stell- und Zählpunkte kann optional eine CC600-Adresse für die Geber-Störung und die Grenzwerte angegeben werden. Ist eine davon definiert so wird der Datenpunkt ca. 10-mal innerhalb des in der ES-Zeile definierten Abfrageintervalles gepollt.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_actual	CC600-Adresse für "actual"
6	38	42	5	UR_Fkt	Nummer der Umrechnungsfunktion des CC600-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung
7	44	48	5	AbfrInt	Abfrageintervall beim Polling in Sekunden (>= 2 Sek.)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schalterpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_nominal	CC600-Adresse für "nominal"
6	38	42	5	UR_Fkt	Nummer der Umrechnungsfunktion des CC600-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung
7	44	48	5	AbfrInt	Abfrageintervall beim Polling in Sekunden (>= 2 Sek.)
8	50	57	8	Adr_actual	CC600-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_actual	CC600-Adresse für "actual"
6	38	42	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der CC600- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	44	51	8	Adr_fl	CC600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	53	60	8	Adr_alarm_low	CC600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

9	62	69	8	Adr_alarm_high	CC600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	71	78	8	Adr_warning_low	CC600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	80	87	8	Adr_warning_high	CC600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_nominal	CC600-Adresse für "nominal"
6	38	42	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der CC600- in die physikalischen Werte angewen- det werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
6	44	51	8	Adr_f1	CC600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	53	60	8	Adr_actual	CC600-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_actual	CC600-Adresse für "actual"
6	38	42	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der CC600- in die physikalischen Werte angewen- det werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	44	51	8	Adr_f1	CC600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	53	60	8	Adr_alarm_high	CC600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	62	69	8	Adr_warning_high	CC600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpnt.
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt
...
ADR_actu URFkt AbfrI
ADR_nomi URFkt AbfrI ADR_actu
ADR_actu KL_Id ADR_f1__ ADR_a1__ ADR_ah__ ADR_w1__ ADR_wh__
ADR_nomi KL_Id ADR_f1__ ADR_actu
ADR_actu KL_Id ADR_f1__ ADR_a1__ ADR_wh
```

#### 5.4.8.4. CNC+ (Fa. TREND)

Grundlage für die Realisierung sind folgenden Dokumente :

- "Technical Manual CNC+ Interface Development" (Dok.Nr.: TD100659A Ausgabe 4/B 21/6/93)
- "IQ Konfiguration Referenzhandbuch Version 1.0"
- "IQ3 Strategy Engine Functional Specification", Stand: 3.03.2004
- diverse Anpassungen und Erweiterungen nach Absprache mit Fa, TREND und GfG
- Alarm-Messages werden auf die "Text Alarm Communication" beschränkt
- Das Quittierungsbit muß nach einem Alarm nicht explizit gesetzt werden, da dieses von der DDC aufgrund des ACK automatisch durchgeführt wird
- von einer IQ3 o.ä. intern invertierte Melde- und Schalterpunkte (ATTRIBUT-Communication) werden direkt unterstützt, für ältere DDC's wie IQ2 (RANDOM-Communication) muss diese in der IS-Zeile explizit parametrisiert werden

Die DDC wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4 bzw. /dev/ttyS0,...). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

9600 Baud, 1 Startbit, 7 Datenbit, ODD-Parity-Bit und 1 Stopbit



IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '''')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 114, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt : 0: CNC+ ohne PIN 1: CNC+ mit PIN 1905 (Ebene 99)
7	44	46	3	CNC-Adresse	Eigene CNC-Adresse Voreinstellung für LHM: 80
8	48	256	<=209	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (DOS: ohne ") bzw. "/dev/ttySn" (LINUX: mit "n" = 0,1,...; ohne ")  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>), je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Ab GA-Knoten-Version 1.2r wird ein Zeitabgleich zur DDC durchgeführt, falls für den GA-Knoten ein solcher Zeitabgleich in der ZA-Zeile parametrier ist. Der GA-Knoten übernimmt dabei die Funktion eines "Zeitverwalters", wobei Anfragen aus dem lokalen Netzwerk nicht unterstützt werden (nur IQ90 und feste Adresse 11).

Ab der Version 1.3e können externe CNC-Netzwerke über ANC+ oder TMN angeschlossen werden. Die Zuordnung erfolgt unter Verwendung der Treiber-SubId. in der IS-Zeile. Dabei ist innerhalb der Remote-Anlage sicherzustellen, dass alle dortigen lokalen Netze (vgl. dortige INC) über diesen einen ANC erreichbar sind. Es gilt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IS"	Zeilen-Kennung (OHNE '"')
...					
4	25	27	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ wie folgt: 0: Datenpunkt im lokalen Netz >0: Datenpunkt in externem Netz wobei die zugehörige Rufnummer in der VBP-Datei im Unterverzeichnis "iz" zugeordnet ist und in der Datei ANRUFER.CHK im Unterverzeichnis "cfg" (analog dem Verfahren zum IZ-Treiber FND_CAPI mit angepasster Treiber-Id.)
5 ff.	29	...		IZn	spezifischer Teil abhängig vom Typ/Protokoll n der DDC-IZ

**Hinweis:** Anders als beim IZ-Treiber FND\_CAPI sind für Treiber-SubId. > 0 KEINE eigenen IZ-Zeilen erforderlich, da der Zugriff auf die externen CNC-Netzwerke stets über ANC o.ä. erfolgt, die in das lokale CNC-Netzwerk integriert sind.

Wird ein Datenpunkt über ANC o.ä. aufgeschaltet, so führen vom GA-Knoten erzeugte zyklische Polling-Abfragen nur dann zum Verbindungs-Aufbau, wenn diese explizit für eine Protokollierung benötigt werden (d.h. IZ\_subid != 1). Ansonsten werden die Abfragen mit einem REJECT-Datagramm (IZ momentan nicht verfügbar) abgewiesen.

Auch der Zeitabgleich führt nicht zum Verbindungsaufbau über ANC o.ä.

Ab Version 1.3e wird auch die IQ3 unterstützt. Für diese Systeme wird die bisherige Adressierung mittels Index und ggf. Bit-Adresse NICHT mehr bzw. nur noch eingeschränkt unterstützt. Auch bestehen Abweichungen bei der Attribute-Communication. So werden von einer IQ3 o.ä. invertierte Melde- und Schaltpunkte (ATTRIBUT-Communication) werden direkt unterstützt.

**Wichtiger Hinweis:** Für CNC-Systeme sind die bisherigen Adressen zu verwenden. Für IQ3-Systeme sind folgende ITEM-Adressen zu verwenden :

Meldepunkt : Innnn(S)	(Input)
Schaltpunkt: Wnnnn(S) Innnn(S)	(Schalter+Input für Rückmeld.)
Messpunkt : Snnnn(V)	(Sensor)
Stellpunkt : Knnnn(V) Snnnn(V)	(Knopf+Sensor für Istwert)
Zählpunkt : Gnnnn(M) 00000	(Logic; KEIN expl. LastSave)
oder Snnnn(V) 00000	(Sensor; KEIN expl. LastSave; ab Version 1.3k)

Ist die Führungsgröße im ITEM-Format angegeben, so muss auch die optionale Angabe für die Rückmeldung usw. im ITEM-Format erfolgen oder es wird "0" angegeben.

Für 1- und 2-stufige Melde- und Schaltpunkte wird zur Umrechnung des Zustands für "actual" und "nominal" in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:

CNC	FND
0 <->	1
1 <->	0

Die Grenzwertüberwachung für Mess- und Zählpunkte ist nur für Sensoren/Fühler realisiert und ist beschränkt auf die Alarm-Grenzen. Folglich muß in der FND-Referenzdatei der zugehörige Wert für "update\_control" (Spalte 45-47) wie folgt eingetragen werden :

- 006 Messpunkt mit Grenzwertüberwachung (NUR Sensor/Fühler)
- 015 Messpunkt sonst
  
- 002 Zählpunkt mit Grenzwertüberwachung (NUR Sensor/Fühler)
- 006 Zählpunkt sonst

Soll für einen Zählpunkt der Setzvorgang durch die LZ verhindert werden, so sind die Werte 003 bzw. 007 zu verwenden. Für Melde-, Schalt- und Stellpunkte ist der Wert ohne Bedeutung und ist mit 015 zu besetzen.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) müssen durch eine Alarm-Message ausgelöst werden. Folglich muß das zugehörige ENABLE-Bit gesetzt sein, unabhängig davon, ob es sich um eine Betriebs- oder Stör-/Gefahr-Meldung handelt. Dadurch kommt es ggf. zu einer großen Anzahl von Einträgen für Betriebsmeldungen im Alarm-Puffer der DDC. Projektspezifisch ist somit festzulegen, welche Betriebsmeldungen an den GA-Knoten übergeben werden sollen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 folgenden Aufbau :

**- Meldepunkte**

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	LAN-Adresse	Adresse des TREND LAN-Station auf der der DP realisiert ist Ist dieser Wert != 0, so wird Extended Adressing verwendet. SONST: Standard Adressing, d.h. der DP befindet sich auf einer DDC innerhalb des LAN wie der GA-Knoten
6	33	35	3	DDC-Adresse	Adresse der TREND DDC-Station auf der der DP realisiert ist
7	37	41	5	Index/Bit_actual	Index und Bit-Adresse in der Bitmatrix für "actual" durch Komma getrennt (z.B. 1,0) Für ITEM-Adr.: Innnn
8	43	47	5	UR_Fkt	Nummer der Umrechnungsfunktion des CNC-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) (Dieser Parameter kann optional angegeben werden)

**- Schaltpunkte**

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	LAN-Adresse	Adresse des TREND LAN-Station auf der der DP realisiert ist Ist dieser Wert != 0, so wird Extended Adressing verwendet. SONST: Standard Adressing, d.h. der DP befindet sich auf einer DDC innerhalb des LAN wie der GA-Knoten
6	33	35	3	DDC-Adresse	Adresse der TREND DDC-Station auf der der DP realisiert ist
7	37	41	5	Index/Bit_nominal	Index und Bit-Adresse in der Bitmatrix für "nominal" durch Komma getrennt (z.B. 1,0) Für ITEM-Adr.: Wnnnn
8	43	47	5	Index/Bit_actual	Index und Bit-Adresse in der Bitmatrix für "actual" durch Komma getrennt (z.B. 1,0) Ist der Index == 0, so wird der "nominal" direkt übernommen (OHNE Rückmeldung) Für ITEM-Adr.: Innnn
9	49	53	5	UR_Fkt	Nummer der Umrechnungsfunktion des CNC-Wertes auf den FND-Index : 0: Keine Umrechnung

1: Invertierung (1-stufig)  
 (Dieser Parameter kann  
 optional angegeben werden)

**- Messpunkte**

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	LAN-Adresse	Adresse des TREND LAN-Station auf der der DP realisiert ist Ist dieser Wert != 0, so wird Extended Adressing verwendet. SONST: Standard Adressing, d.h. der DP befindet sich auf einer DDC innerhalb des LAN wie der GA-Knoten
6	33	35	3	DDC-Adresse	Adresse der TREND DDC-Station auf der der DP realisiert ist
7	37	41	5	Index_actual	Index in der analogen Datenmatrix für "actual". Ist ein Sensor/Fühler angegeben, so wird der untere und obere Alarm-Grenzwert unterstützt. Für ITEM-Adr.: Snnnn

**- Stellpunkte**

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	LAN-Adresse	Adresse des TREND LAN-Station auf der der DP realisiert ist Ist dieser Wert != 0, so wird Extended Adressing verwendet. SONST: Standard Adressing, d.h. der DP befindet sich auf einer DDC innerhalb des LAN wie der GA-Knoten
6	33	35	3	DDC-Adresse	Adresse der TREND DDC-Station auf der der DP realisiert ist
7	37	41	5	Index_nominal	Index in der analogen Datenmatrix für "nominal". Für ITEM-Adr.: Knnnn
8	43	47	5	Index_actual	Index in der analogen Datenmatrix für "actual". Ist der Index == 0, so wird der "nominal" direkt übernommen (analog zum Schaltpunkt OHNE Rückmeldung) Für ITEM-Adr.: Snnnn

**- Zählpunkte**

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	LAN-Adresse	Adresse des TREND LAN-Station auf der der DP realisiert ist Ist dieser Wert != 0, so wird Extended Adressing verwendet. SONST: Standard Adressing, d.h. der DP befindet sich auf einer DDC innerhalb des LAN wie der GA-Knoten
6	33	35	3	DDC-Adresse	Adresse der TREND DDC-Station auf der der DP realisiert ist
7	37	41	5	Index_actual	Index in der analogen Datenmatrix für "actual". Ist ein Sensor/Fühler angegeben, so wird der obere Alarm-Grenzwert unterstützt. Für ITEM-Adr.: Gnnnn
8	43	47	5	Index_last_save	Index in der analogen Daten-

matrix für "last\_save".  
 Ist der Index == 0, so wird  
 der "last\_save" beim (Rück-)  
 Setzvorgang aus dem letzten  
 vermerkten "actual" direkt  
 übernommen.  
 Für ITEM-Adr.: 00000

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei (CNC-Systeme) :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt LAN DDC aaa,a UrFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt. LAN DDC nnn,n aaa,a UrFkt
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt LAN DDC aaaaa
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt LAN DDC nnnnn aaaaa
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt LAN DDC aaaaa lllll
...
```

Beispiel für eine FND-Referenzdatei (IQ3-Systeme) :

```
00000 06/05/2004 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 000 045 Klartext für Meldepunkt LAN DDC Iaaaa URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 000 045 Klartext für Schalterpkt. LAN DDC WWnnn Iaaaa URFkt
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 006 108 Klartext für Messpunkt LAN DDC Saaaa
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 000 108 Klartext für Stellpunkt LAN DDC Knnnn Saaaa
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 007 014 Klartext für Zaehlpunkt LAN DDC Gaaaa 00000
...
```

**5.4.8.5. DEKATEL (Fa. Viessmann)**

Grundlage für die Realisierung sind :

- "Pflichtenheft Kommunikationsschnittstelle K-Bus", Version 2.2 vom 15.11.93
- "Technisches Datenblatt Dekatel-G"

Die DEKATEL-G wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

1200 Baud, 1 Startbit, 8 Datenbit, EVEN-Parity-Bit und 1 Stopbit

Die DEKATEL wird beim Programmstart und nach jeder erkannten Störung neu initialisiert. Dabei werden alle Datenpunkte, auf die nur lesend zugegriffen wird, als Byte vereinbart, damit insbesondere beim Polling für alle Statusbits innerhalb eines Bytes (z.B. DPA 0x44) nur ein Lesezugriff erforderlich ist.

**Hinweis:** Beim Synchronisieren der Schnittstelle nach dem Programmstart und beim Lesezugriff auf Datenpunkte mittels der "transparenten Datenübertragung" wurden abweichend vom Pflichtenheft Antwortzeiten von bis zu 120 Sekunden beobachtet. Nach einem Schreibzugriff mittels der "transparenten Datenübertragung" ist die DEKATEL für bis zu 60 Sekunden nicht mehr kommunikationsbereit (liefert NAK).

Also gilt : \*\*\* Bitte warten !! \*\*\*

Der IZ-Treiber prüft vor jedem WRITE auf die DEKATEL-G die Modifikations-Berechtigung. WRITE-Zugriffe mittels der "transparenten Datenübertragung" wird nur dann zugelassen, wenn der Datenpunkt vom Typ Byte ist.

Die WRITE-Zugriffe auf die einzelnen Datenpunkte durch die DEKATEL-G bzw. DEKAMATIC unterliegen keiner Kontrolle, d.h. die durch FND gesetzte Sperrpriorität wird nicht unterstützt.

**Hinweis:** Die Zuordnung der FND-Operationen auf die DEKATEL-Datenpunkte und die Zugriffskontrolle erfolgen ausschließlich anhand der in der Konfigurations-Datei angegebenen Parameter. Der Projektant bzw. Installateur hat sicherzustellen, dass ausschließlich die von VIESSMANN für die jeweilige Steuerung angegebenen Teilnehmer- und Datenpunkt-Adressen, sowie der zugehörige Datenpunkt-Typ parametrisiert wird. Abweichungen können zu fehlerhaften Werten und Beeinträchtigungen im Regel- und Steuerungsverhalten führen !

Es wird empfohlen eine allgemeine Vorlage für die FND-Referenzdatei bzw. die Konfigurationsdatei des GA-Knotens zu verwenden, in der nur die benötigten Datenpunkte aktiviert werden und die FND-Datenpunkt-Adresse, der FND-Datenpunkt-Text sowie die spezifischen Parameter in der ES-Zeile angepaßt werden. Nicht vorhandene Teilnehmer können durch Kopieren der Vorlage eines Teilnehmers desselben Steuerungstyps und die anschließende Anpassung der Teilnehmer-Adresse erzeugt werden.

Die Grenzwertüberwachung für Mess- und Zählpunkte wird vom IZ-Treiber emuliert. Zusätzlich wird bei Schalt-, Stell- und (Rück-)Setzbefehlen der binäre Vorgabewert vor der Übertragung an die DEKATEL-G gegenüber dem parametrisierten MIN-/MAX-Wert verglichen und bei Verletzung der Grenzen auf den MIN-/MAX-Wert gesetzt.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

Für die Schnittstellen-Bezeichnung in der IZ-Zeile der Konfigurationsdatei kann angegeben werden : "COM1", "COM2", "COM3" oder "COM4" (OHNE "). Für die Initialisierung der DEKATEL-G werden zusätzlich Fortsetzungszeilen zur IZ-Zeile mit dem folgenden Format definiert:

1	1	2	2	"++"	Zeilen-Kennung (OHNE ''')
2	4	6	3	TA	Teilnehmer-Adresse auf der der DP realisiert ist
3	8	10	3	DPA	Datenpunkt-Adresse auf der DEKAMATIC. Diese wird in Anlehnung an die VIESSMANN-Spezifikation hexadezimal angegeben !
4	12	14	3	BA	Basis-Adresse auf der DEKATEL-G (1, ..., 8)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für alle FND-Datenpunkt-Typen (1-5) folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	DP-Typ	Datenpunkt-Typ wie folgt kodiert : 1: 1 Bit 2: 2 Bit 4: Nibble / Halbbyte 8: Byte (unsigned)  13: Betriebszeit (Bzn...) 18: Byte (signed)
6	33	35	3	TA	Teilnehmer-Adresse auf der der DP realisiert ist
7	37	44	8	DPA	Datenpunkt-Adresse auf der DEKATEL/DEKAMATIC. Diese wird in Anlehnung an die VIESSMANN-Spezifikation hexadezimal angegeben !! Für Betriebszeiten (DP-Typ 13) werden die DPA für HST, STD und MIN durch Komma getrennt angegeben.
8	46	46	1	Bit	Bit ab dem der Wert im Byte kodiert ist (0, ..., 7) Für DP-Typen >= 8 wird dieser Wert ignoriert.
9	48	50	3	MIN_Wert	Minimal zulässiger Wert
10	52	54	3	MAX_Wert	Maximal zulässiger Wert
11	56	60	5	Kennlinie	Nummer der Kennlinie die zur Umrechnung der binären in die physikalischen Werte (und umgekehrt) verwendet werden soll. Dieser Wert ist nur für Mess-, Stell- und Zähl-Punkte anzugeben (vgl. KL-Zeile in der System-Spezifikation zum GA-Knoten)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellereigenen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt          DPT TA_ DPA_____ B MIN MAX
```



00002	Schaltpunkt_1_____	2	0	15	15	15	0	0	015	045	Klartext	für	Schaltpkt.	DPT	TA_	DPA_____	B	MIN	MAX	
00003	Messpunkt_1_____	3	0	15	15	15	0	0	015	108	Klartext	für	Messpunkt	DPT	TA_	DPA_____	B	MIN	MAX	KL_ID
00004	Stellpunkt_1_____	4	0	15	15	15	0	0	015	108	Klartext	für	Stellpunkt	DPT	TA_	DPA_____	B	MIN	MAX	KL_ID
00005	Zaehlpunkt_1_____	5	0	15	15	15	0	0	015	108	Klartext	für	Zaehlpunkt	DPT	TA_	DPA_____	B	MIN	MAX	KL_ID
...																				

**5.4.8.6. P90 (Fa. Kieback & Peter)**

Grundlage für die Realisierung sind :

- P90-Definitionen

**Hinweis:** Mit der DDC-Programm-Version 2.7 wurde eine Protokoll-Änderung durchgeführt, die im GA-Knoten ab Version 1.2 berücksichtigt ist.

Ab Version 1.3e wird auch die Protokoll-Variante für HRP- und MRP-Regler unterstützt.

Ab Version 1.3k werden auch die folgenden Datums-/Zeit-Formate (**DZF**) der P90-Parameter Datum, Uhrzeit, Nutzzeit und Sonder-Nutzzeit unterstützt:

Die DDC wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

Für MRP (vgl. Parameter: 1794, 1805, 1806 usw.):

4800 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit, 1 Stopbit und XON/XOFF

Die DDC wird beim Programmstart und nach jeder erkannten Störung neu initialisiert. Dabei erfolgt die Anmeldung (G0-Befehl).

Die DDC kann spontan melden, so dass Änderungen der Zustände von Melde- und Schaltpunkte daran erkannt werden. Relevante Änderungen der Zustände von Mess-, Stell- und Zählpunkten werden ebenfalls spontan gemeldet. Eine zyklische Abfrage (Polling) erfolgt nur gemäß dem in der Konfigurationsdatei definierten Abfrageintervall.

Alle P90-Datenpunkte - mit Ausnahme der DZF-Parameter - werden mit dem W0- und P0-Befehl gelesen, wobei gleichzeitig deren spontane Meldungen aktiviert werden. Beim Programmstopp werden die spontanen Meldungen wieder deaktiviert (W8-Befehl).

Unterstützt werden die Parametertypen, die die Werte 0 und 1 liefern (vgl. Ja/Nein-Pärchen). Verwendet wird grundsätzlich das in der Konfigurations- bzw. Referenz-Datei angegebene FND-Attribut unabhängig vom P90-Code.

Für 2-stufige Melde- und Schaltpunkte wird der gelöschte Wert ("-") auf den FND-Index 2 (0x02) abgebildet. Die 2-stufige Invertierung (vgl. UR\_Fkt) kann zur Anpassung an das FND-Attribut 43 (AUTOMATIK, HAND, AUS) verwendet werden.

Für Mess-, Stell- und Zählpunkte werden nur P90-Analogwerte unterstützt. Verwendet wird grundsätzlich das in der Konfigurations- bzw. Referenz-Datei angegebene FND-Dimension, unabhängig vom P90-Code.

Für Stellpunkte kann ab Version 1.3k alternativ zur Kennlinie auch ein Datums/Zeit-Format (DZF) angegeben werden. Damit lassen sich auch folgende P90-Parameter lesen und modifizieren:

<b><u>P90-Parameter</u></b>	<b><u>Datums/Zeit-Formate (KL Id) für FND-Stellpunkte</u></b>
Datum	DATUM
Uhrzeit	ZEIT
Nutzzeit (1-fach; z.B. HRP)	NZVON NZBIS
Nutzzeit (4in1; z.B.DDC3200)	NZV40 NZB40 NZV41 ... NZB43
Sondernutzzeit	SNDAT SNVON SNBIS

**Wichtiger Hinweis:** In der DDC3xxx können Sondernutzzeiten aus 2 getrennten P90-Parametern vom Typ Datum und Nutzzeit (4in1) realisiert sein (z.B. 5010/1 und 5011/1). Darüber hinaus unterscheiden sich DDC3xxx und HRP z.B. auch bei der Behandlung von Tagesanfang und -ende.

Über die gemeinsame P90-Adresse für "nominal" werden diese FND-Stellpunkte dem jeweiligen P90-Parameter zugeordnet.

Dabei erfolgt die Darstellung der einzelnen Zahlenwerte im folgenden Format, ggf. OHNE führende Nullen :

- "tmmjj" Datum für DATUM, SNDAT
- "ssmm" Zeit für sonstige
- "NAN" o.ä. P90-Parameter bzw. VON/BIS-Wert ist gelöscht (Not A Number)

Sind die "nominal" für alle einem P90-Parameter zugeordneten Datenpunkte definiert und gültig, so wird der Stellbefehl auf den P90-Parameter entsprechend ausgeführt. Ansonsten gilt :

- alle "nominal" UNDEF ("NAN") P90-Parameter wird gelöscht
- "tmmjj" ungültig/UNDEF (<0101 od. >3112, "NAN") -"
- "ssmm" für Zeit UNDEF (<0000 od. >2359, "NAN") -"
- "ssmm" für VON und BIS ungültig /UNDEF (<0000 od. >2359, "NAN",  
VON-Zeit = BIS-Zeit = 00:00) -"
- "ssmm" für sonstige Zeiten UNDEF Tagesanfang (VON) bzw. Tagesende (BIS)

**Beispiel:** Soll eine bisher gelöschte Nutzzeit auf den Zeitbereich 5:30-20:00 gestellt werden, so sind gesonderte Stellbefehle auf die Stellpunkte für NZVON und NZBIS mit den Werten 530 bzw. 2000 auszuführen.

Für Sonder-Nutzzeiten sind SNVON und SNBIS analog zu behandeln, wobei zusätzlich ein Stellbefehl auf den Stellpunkt für SNDAT auszuführen ist.

Zum Löschen eines Eintrags erst die VON-Zeit und dann die BIS-Zeit z.B. auf -1 setzen.

**Hinweis:** Der Ersteller der Konfigurations- bzw. Referenz-Datei hat sicherzustellen, dass für alle FND-Datenpunkte nur solche P90-Parameter definiert werden, deren Parametertyp vom IZ-Treiber unterstützt werden.

Der IZ-Treiber prüft vor jedem Setzbefehl (P1) die Modifikations-Berechtigung. Die internen Setz-Befehle auf die einzelnen Datenpunkte durch die DDC unterliegen keiner Kontrolle, d.h. die durch FND gesetzte Sperrpriorität wird nicht unterstützt.

Für Mess- und Zählpunkte können optional P90-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die P90-Adressen werden zyklisch gemäß dem in der ES-Zeile definierten Abfrageintervall gepollt und geprüft. Eine Behandlung der von der DDC spontan gemeldeten Grenzwert-Alarmer erfolgt nicht. Bei Bedarf können diese jedoch gesondert als Meldepunkte definiert werden.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** P90: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** P90: stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** P90: 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt : P90-DDC3000: 0 (Voreinstellung) P90-HRP ...: 1 P90-MRP ...: 2
7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (ohne ")  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die P90-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"AAA.BBB.GGG:PPPP[NN]" mit :

AAA ...: Anlage (Strang)

BBB ...: Bus-Nummer (Zentrale)

GGG ...: Geräte-Adresse (Modul)

PPPP ..: Parameter-Nummer

NN ....: Menü-Nummer (optional)

Für die folgenden optionalen P90-Adressen eines Datenpunktes (z.B. Geber-Störung, Grenzwerte) muß nur noch "PPPP[NN]" angegeben werden. Die anderen Bestandteile "AAA.BBB.GGG:" werden von der Führungsgröße übernommen. Sie können jedoch auch explizit angegeben werden, wenn sie von dieser abweichen.

Die Parameter- und Menü-Nummer müssen genauso angegeben werden, wie sie von der DDC in der Spontanmeldung oder Antwort geliefert werden (z.B. OHNE führende Nullen). Wird die Menü-Nummer nicht angegeben, so wird die in der Spontanmeldung bzw. Antwort enthaltene Menü-Nummer ignoriert. Bei der Erstellung der Referenzdatei ist sicherzustellen, dass die P90-Adresse eindeutig einem Datenpunkt zuzuordnen ist.

Für Mess-, Stell- und Zählpunkte kann optional eine Kennlinien zugeordnet werden. Dieses ist notwendig, da das P90-System teilweise von der FND-Spezifikation abweichende Dimensionen verwendet (z.B. "mb" bzw."b").

Für Mess-, Stell- und Zählpunkte kann optional eine P90-Adresse für die Geber-Störung und die Grenzwerte angegeben werden. Die Geber-Störung wird nach jedem Lesen oder einer spontanen Meldung erkannt. Die Grenzwert-Überwachung erfolgt nach jedem Lesen durch den IZ-Treiber.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	47	19	Adr_actual	P90-Adresse für "actual"

6	49	53	5	UR_Fkt	Nummer der Umrechnungsfunktion des P90-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig)
---	----	----	---	--------	--

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	47	19	Adr_nominal	P90-Adresse für "nominal"
6	49	53	5	UR_Fkt	Nummer der Umrechnungsfunktion des P90-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig)
7	55	73	19	Adr_actual	P90-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	47	19	Adr_actual	P90-Adresse für "actual"
6	49	53	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der P90- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	55	73	19	Adr_fl	P90-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	75	93	19	Adr_alarm_low	P90-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	95	113	19	Adr_alarm_high	P90-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	115	133	19	Adr_warning_low	P90-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	135	153	19	Adr_warning_high	P90-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

5	29	47	19	Adr_nominal	P90-Adresse für "nominal"
6	49	53	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der P90- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)  Alternativ kann eines der folgenden Datums/Zeit-Formate angegeben werden : - DATUM Datum - ZEIT Uhrzeit - NZVON Nutzzeit (VON) - NZBIS - " - (BIS) - SNDAT Sondernutzzeit (DATUM) - SNVON - " - (VON) - SNBIS - " - (BIS)
6	55	73	19	Adr_f1	P90-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
7	75	93	19	Adr_actual	P90-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	47	19	Adr_actual	P90-Adresse für "actual"
6	49	53	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der P90- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	55	73	19	Adr_f1	P90-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	75	93	19	Adr_alarm_high	P90-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	95	113	19	Adr_warning_high	P90-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt ADR_actual_____ URfkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt. ADR_nominal_____ URfkt ADR_actual_____
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt ADR_actual_____ KL_Id ADR_f1_____
ADR_al_____ ADR_ah_____ ADR_wl_____ ADR_wh_____
```

```

00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt      ADR_nominal_____ KL_Id ADR_f1_____
ADR_actual_____
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt      ADR_actual_____ KL_Id ADR_f1_____
ADR_al_____ ADR_wh_____
...

```

Beispiele für Datums/Zeit-Formate bei Stellpunkten :

```

...
00011 Stellpunkt_DATUM 4 0 15 15 15 0 0 015 019 Klartext für Stellpunkt      ADR_nominal__DATUM_ DATUM
...
00021 Stellpunkt_ZEIT 4 0 15 15 15 0 0 015 018 Klartext für Stellpunkt      ADR_nominal__ZEIT__ ZEIT
...
00031 Stellpunkt_NZVON 4 0 15 15 15 0 0 015 018 Klartext für Stellpunkt      ADR_nominal__NZ____ NZVON
00032 Stellpunkt_NZBIS 4 0 15 15 15 0 0 015 018 Klartext für Stellpunkt      ADR_nominal__NZ____ NZBIS
...
00041 Stellpunkt_SNDAT 4 0 15 15 15 0 0 015 019 Klartext für Stellpunkt      ADR_nominal__SN____ SNDAT
00042 Stellpunkt_SNVON 4 0 15 15 15 0 0 015 018 Klartext für Stellpunkt      ADR_nominal__SN____ SNVON
00043 Stellpunkt_SNBIS 4 0 15 15 15 0 0 015 018 Klartext für Stellpunkt      ADR_nominal__SN____ SNBIS

```

#### 5.4.8.7. SAIA-Bus / GfR

Grundlage für die Realisierung sind :

- "SAIA Bus Protocol description ...", Version vom 16.07.92

**Hinweis:** Unter LINUX ist ein direkter Zugriff auf den UART-Baustein erforderlich, um das FORCED/MARK-Parity-Bit setzen zu können. Aufgrund des erforderlichen System-Aufruf IOPERM besteht eine Begrenzung auf serielle Schnittstellen mit Port-Adressen <= 0x3FF (vgl. "man ioperm") und der Treiber kann nur mit SuperUser-Rechten (vgl. LINUX-Befehl "su" bzw. Login "root") verwendet werden.

Ab Version 1.3k unterstützt der IZ-Treiber alternativ ein Verfahren unter Verwendung des CMSPAR-Flags (vgl. Verbindungs-Protokoll in der IZ-Zeile). Dieses MUSS vom LINUX-Treiber "serial.o" unterstützt werden, ermöglicht dann aber höhere Port-Adressen (z.B. PCI-Steckkarten) und erfordert keine Superuser-Rechte.

Spezifisch für die Fa. GfR besteht ab Version 1.3k die spezifische Erweiterung des gemischten Betriebs von Stationen mit unterschiedlichen Protokollen (PCD2, PCD1/NT, ...) gemäß "SBUS+ Protocol" (vgl. unten).

Die SAIA-Bus-Systeme werden über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, FORCED/MARK-Parity-Bit und 1 Stopbit



Der Treiber fungiert dabei als Standard-SBUS-Master. Beachten Sie hierzu die Hinweise des Herstellers hinsichtlich der jeweils verwendeten Kabel und Einstellungen zur Schnittstelle.

Die SAIA-Bus-Systeme können NICHT spontan melden, so dass auch Melde- und Schaltpunkte zyklisch abgefragt werden müssen (Polling), um Änderungen im "actual" zu erkennen. Analog müssen auch die Mess-, Stell- und Zählpunkte zyklisch abgefragt werden, um Geberstörungen und das Über-/Unterschreiten von Grenzwerten zu erkennen. Nachdem alle Melde- und Schaltpunkte abgefragt wurden wird der nächste Mess-, Stell- bzw. Zählpunkt abgefragt, bevor wieder mit dem ersten Melde- bzw. Schaltpunkt begonnen wird.

Der IZ-Treiber prüft vor jedem WRITE auf die SAIA-Bus-Adresse die Modifikations-Berechtigung. Die WRITE-Zugriffe auf die einzelnen Datenpunkte durch das SAIA-Bus-System unterliegen keiner Kontrolle, d.h. die durch FND gesetzte Sperrpriorität wird nicht unterstützt.

Für Mess- und Zählpunkte können optional SAIA-Bus-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die SAIA-Bus-Adressen werden zyklisch gepollt und geprüft (vgl. oben). Eine Behandlung der vom SAIA-Bus-System erzeugten Grenzwert-Alarme erfolgt nicht. Bei Bedarf können diese jedoch gesondert als Meldepunkte definiert werden.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 110, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für

					<p>*** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** stets 0</p>
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der DDC-IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	<p>natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt : *** stets 0</p>
6	40	42	3	Protokoll	<p>nat. Zahl kodiert wie folgt :</p> <p>*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt unter DOS : *** stets 0</p> <p>*** Bis dahin gilt unter LINUX : 1: Zugriff mit CMSPAR-Flag (MUSS vom LINUX-Treiber "serial.o" unterstützt werden) sonst: stets 0 (Zugriff mit IOPERM)</p>
7	44	256	<=213	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.</p> <p>"COM1" - "COM4" (ohne ")</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die SAIA-Bus-Adressen werden in den IS-Zeilen wie folgt kodiert:

"sss.taaaaa" mit :

sss ..: Station

t ..: Typ

aaaaa : Adresse

Dabei sind "sss" und "aaaaa" dezimale Zahlen  $\geq 0$  und "t" der erste Buchstabe der nachfolgend aufgelisteten Typen (z.B. "R" für Register) :

- Register

- Counter
- Timer
- Flag
- Input
- Output

Für Melde- und Schaltpunkte kann optional eine Umrechnungsfunktion für die Invertierung der binären Werte angegeben werden. Mehrstufige Melde- und Schaltpunkte müssen auf Register mit den Werten 0,1,...,8 abgebildet und in der IZ dekodiert werden.

Für Mess-, Stell- und Zählpunkte muß eine Kennlinien zugeordnet werden. Optional kann eine SAIA-Bus-Adresse für die Geber-Störung und die Grenzwerte angegeben werden. Ist eine davon definiert so wird der Datenpunkt ca. 10-mal innerhalb des in der ES-Zeile definierten Abfrageintervalles gepollt.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_actual	SAIA-Bus-Adresse für "actual"
6	40	44	5	UR_Fkt	Nummer der Umrechnungsfunktion des SAIA-Bus-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_nominal	SAIA-Bus-Adresse für "nominal"
6	40	44	5	UR_Fkt	Nummer der Umrechnungsfunktion des SAIA-Bus-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (Dieser Parameter kann optional angegeben werden)

7	46	55	10	Adr_actual	SAIA-Bus-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)
---	----	----	----	------------	---

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_actual	SAIA-Bus-Adresse für "actual"
6	40	44	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der SAIA-Bus- in die physikalischen Werte angewen- det werden soll (Entspricht momentan dem Index in der KL-Struktur)  Hinweis: Momentan wird nur ----- das INTEGER-Format unterstützt, so dass eine Kennlinie zuge- ordnet werden muß !
7	46	55	10	Adr_f1	SAIA-Bus-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	57	66	10	Adr_alarm_low	SAIA-Bus-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	68	77	10	Adr_alarm_high	SAIA-Bus-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	79	88	10	Adr_warning_low	SAIA-Bus-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	90	99	10	Adr_warning_high	SAIA-Bus-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_nominal	SAIA-Bus-Adresse für "nominal"
6	40	44	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der SAIA-Bus- in die physikalischen Werte angewen- det werden soll (Entspricht momentan dem Index in der KL-Struktur)  Hinweis: Momentan wird nur ----- das INTEGER-Format unterstützt, so dass eine Kennlinie zuge- ordnet werden muß !
7	46	55	10	Adr_f1	SAIA-Bus-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional

					angegeben werden)
8	57	66	10	Adr_actual	SAIA-Bus-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_actual	SAIA-Bus-Adresse für "actual"
6	40	44	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der SAIA-Bus- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur)  Hinweis: Momentan wird nur ----- das INTEGER-Format unterstützt, so dass eine Kennlinie zugeordnet werden muß !
7	46	55	10	Adr_f1	SAIA-Bus-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	57	66	10	Adr_alarm_high	SAIA-Bus-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	68	77	10	Adr_warning_high	SAIA-Bus-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei für SBUS :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt           ADR_actual URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.       ADR_nomin. URFkt ADR_actual
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt           ADR_actual KL_Id ADR_f1_____
ADR_al_____ ADR_ah_____ ADR_wl_____ ADR_wh_____
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt           ADR_nomin. KL_Id ADR_f1_____
ADR_actual
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt           ADR_actual KL_Id ADR_f1_____
ADR_ah_____ ADR_wh_____
...
```

**5.4.8.7.1. GfR mit SBUS+**

Die Fa. GfR verwendet alternativ zum SBUS-Protokoll die spezifische Variante SBUS+.

Ab Version 1.3k unterstützt der IZ-Treiber den erweiterten Adressbereich für Register sowie die Formate Integer, Float, Date (TTMM) und Time (SSMM).

Ab Version 1.43c unterstützt der IZ-Treiber den lesenden und modifizierenden Zugriff auf die Real Time Clock einer jeden angeschlossenen Unterstation.

Ergänzt wird der IZ-Treiber durch das Programm CEX2REF. Mit dessen Hilfe kann für die in Form einer WebCad Datenexportdatei (\*.CEX mit CSV-Format) vorliegende SBUS+-Konfiguration die Vorlage einer FND-Referenzdatei automatisiert generiert werden.

Grundlage für die Realisierung sind :

- "SAIA Bus Protocol description ...", Version vom 16.07.92
- "Communication SBUS+ Protocol", Version vom 04.11.2004
- WebCad Datenexportdatei, Version vom 16.01.2006
- Bedienhandbuch für Gebäudeautomation WebCad, Version vom 21.11.2006
- Angebot GFRVUK/6101 vom 02.10.2006
- Angebot GFRVUG/0021 vom 15.02.2010

Die GfR-Adressen für SBUS+ werden in den IS-Zeilen in Anlehnung an die WebCad Datenexportdatei (CEX) wie folgt kodiert:

"sss/aaaaaaaa/ppp/ddd" mit :

sss .....	: Unterstation	(0,...,255)
aaaaaaaa	: Register-Adresse	(0,...,16777200; "RTC")
ppp .....	: Protokoll	(2, sonst: *reserviert*)
ddd .....	: DataType	(65: Integer, 82: Floating Point, 85: Time (SSMM) 86: Date (TTMM), Sonst: *reserviert* )

**Hinweis:** Wird als Register-Adresse die Zeichenfolge "RTC" angegeben, so sind als DataType nur Date und Time zulässig..

Für Mess-, Stell- und Zählpunkte muß eine Kennlinien zugeordnet werden, wenn nur SBUS-Protokoll oder DataType INTEGER.

Der Zugriff auf die RTC erfolgt je Unterstation über 2 Stellpunkte mit folgenden GfR-Adressen für SBUS+. Dabei ist die Angabe einer Kennlinie nicht erforderlich :

- "sss/RTC/ppp/85"                      RTC-Zeit
- "sss/RTC/ppp/86"                      RTC-Datum

Die übereinstimmende Angabe für "sss/RTC/ppp" sorgt für die gegenseitige Zuordnung der Stellpunkte. Jeder Stellpunkt kann für sich gelesen werden und liefert den jeweiligen Anteil (Datum/Zeit) der RTC.

Zum Setzen der RTC sind die Stellbefehle unabhängig voneinander ausführbar. Dabei gelten folgende Formate :

- TTMMJJ            (RTC-Datum)
- HHMM            (RTC-Zeit)

Für SBUS+ erfolgt der Aufbau der IS-Zeilen analog, jedoch gilt für die o.g. Spalten in Anlehnung an das Format der WebCad Datenexportdatei wie folgt:

- Die einzelnen Spalten werden durch ein Semikolon (;) voneinander getrennt, wobei die festen Längen, Anfang- und Ende-Positionen aufgehoben sind (CSV)
- Für leere Spalten werden KEINE Zeichen vor dem Semikolon angegeben, ggf. folgt das Semikolon direkt auf das vorherige
- Nachhängende Semikolon können entfallen

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei für SBUS+ :

```
00000 10/11/2006 14:00 LHM Stadtsteueramt
00002 Meldepunkt_2_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt          ADR_actual;URFkt
00004 Schaltpunkt_2_____ 2 0 15 15 15 0 0 015 045 Klartext für Schaltpkt.        ADR_nomin.;URFkt;ADR_actual
00006 Messpunkt_2_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt
ADR_actual;KL_Id;ADR_f1____;ADR_al____;ADR_ah____;ADR_wl____;ADR_wh____
00008 Stellpunkt_2_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt
ADR_nomin.;KL_Id;ADR_f1____;ADR_actual
00010 Zaehlpunkt_2_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt
ADR_actual;KL_Id;ADR_f1____;ADR_al____;ADR_wh____
...
```

**5.4.8.7.2. Zusatzprogramm CEX2REF**

Das Zusatzprogramm CEX2REF dient der automatisierten Generierung der Vorlage einer FND-Referenzdatei. Grundlage ist die SBUS+-Konfiguration in Form einer WebCad Datenexportdatei (\*.CEX im CSV-Format).

Der Aufruf erfolgt durch :

```
CEX2REF <[Pfad]\Name der CEX-Datei> [-eEndePrefix] [-bBeginnDP_Id] [-aAnfuegeDP_Id]
```

mit :

- EndePrefix      Zeichenfolge innerhalb der "UserAddress" vor dem Beginn der auszuwertenden DP\_Id
- BeginnDP\_Id    Zeichenfolge innerhalb der "UserAddress", ab der die DP\_Id beginnt
- AnfuegeDP\_Id   Zeichenfolge, die an die gebildete DP\_Id angefügt werden soll

Nach dem Öffnen der CEX- und REF-Datei liest CEX2REF zeilenweise die CEX-Datei und bearbeitet diese wie folgt :

- Scannen der einzelnen Parameter (CSV)
- Prüfen, ob gültiger "ModulTyp" und ggf. abhängig davon sowie von "DP\_Ist\_Soll" Zuordnung von "dp\_type" und "val"
- Übernahme "UserAddress" (max. 16 Zeichen) als "dp\_id", wobei optionale Aufrufparameter für EndePrefix, BeginnDP\_Id und AnfuegeDP\_Id ausgewertet werden.
- Übernahme "TextAddress" (max. 23 Zeichen) als "Datenpunkt-Text" (Klartext zur Beschreibung). Ist nach dem 23. Zeichen eine Zahl in rechteckigen Klammern ([nnn]) angegeben, so wird diese abhängig vom "dp\_type" als "atb\_id" bzw. "dimension" verwendet. Ansonsten ...
- Für Melde- und Schaltpunkte: Suche der "atb\_id" passend zu "DP\_Darst"
- Für Mess-, Stell- und Zählpunkte: Suche der "dimension" passend zu "DP\_Einheit". Dabei Sonderbehandlung für das Datum/Zeit-Format ("DAT" bzw. "UHR"). Hier wird anstelle von "DP\_Einheit" zur Suche der Text "TTMM" bzw. "SSMM" verwendet.
- Für ERSTE gültige Zeile: Übernahme der "Liegenschaft" in die Kopfzeile der Referenzdatei
- Aktuelle Datenpunkt-Zeile in REF-Datei anfügen. Dabei Sonderbehandlung für die ModulTypen FXX/EXX und WHR/EHR, indem für die 14 Datum/Zeit-Angaben einzelne Datenpunkt-Zeilen erzeugt werden. Die "dp\_id" erhält als 16. Zeichen die aufsteigende Buchstabenfolge beginnend mit "a".



- Für jede angegebene Unterstation werden jeweils 2 Stellpunkte für das Lesen und Setzen der RTC angefügt (ab Version 1.43c ; vgl. oben)

#### **5.4.8.7.3. Richtlinien für die Erzeugung der CEX-Datei**

Um das Zusatzprogramm CEX2REF optimiert nutzen zu können sollte die CEX-Datei folgenden Anforderungen entsprechen :

- Begrenzung der als Datenpunkte zu übernehmenden Einträge in der CEX-Datei auf den "ModulTyp" "DIE", "DIA", "ANE", "ANA", "XDE", "XAE", "DIF", "EIF", "FXX", "EXX", "WHR" oder "EHR"
- "UserAddress" muss den Vorgaben des Auftraggebers für "dp\_id" als Datenpunkt-Adresse entsprechen (z.B. Umsetzung mittels UAD-Datei)
- "TextAddress" muss den Vorgaben des Auftraggebers für "Datenpunkt- Text" als Klartext zur Beschreibung des Datenpunktes entsprechen
- Für Melde- und Schaltpunkte sollten "DP\_Darst" die Attributtexten gemäß den Vorgaben des Auftraggebers entsprechen (z.B. Umsetzung mittels UAD-Datei gemäß der vom GA-Knoten verwendeten Datei "dp\_atb.txt")
- Für Mess-, Stell- und Zählpunkte sollte "DP\_Einheit" den Dimensionstexten gemäß den Vorgaben des Auftraggebers entsprechen (z.B. Umsetzung mittels UAD-Datei gemäß der vom GA-Knoten verwendeten Datei "dp\_dim.txt")

#### **5.4.8.8. SUCOM-A (Fa. Moeller)**

Grundlage für die Realisierung sind :

- Protokollbeschreibung SUCOM-A, Version 1.04 vom 17.3.1998

Die SPS wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 2 Stopbit

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 64, ..., 67, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der DDC-IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
7	44	46	3	Typ	Typ der SPS  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
8	48	50	3	Teilnehmernr.	Teilnehmernummer / -Id. 3: PS306/316 7: PS4-201-MM1
9	52	256	<=205	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.

"COM1" - "COM4" (ohne ")

Wird durch das Zeilenabschluß-  
zeichen (<CR> und/oder <LF>,  
je nach verwendetem Editor)  
abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die SPS-Datenpunkte werden gemäß den Angaben der Landeshauptstadt München auf Merkerbyte-Bereiche abgebildet. In den IS-Zeilen werden diese Merkerbyte-Bereiche durch hexadezimale Angabe der Startadresse zugeordnet. Dabe sind auch die "Festlegungen zur FND-Kopplung mit SUCOM-A-Protokoll" zu beachten.

Für Mess-, Stell- und Zählpunkte muß eine Kennlinien zugeordnet werden, die die Umrechnung der binären in die zugeordneten reellen Werte bzw. umgekehrt ermöglich. Über den Kennlinie-Typ wird unterschieden, ob es sich um einen "unsigned"- oder "signed"-Wert handelt.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Melde- und Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	31	3	Op.-Typ	Operandentyp  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 021: Merkerbyte
5	33	37	5	Adresse	Adresse des Operanden
6	39	43	5	Kennlinien-Nr.	Für Melde- und Schaltpunkt Nummer der Funktion, die zur Umrechnung der binären Werte verwendet werden soll wie folgt kodiert : 0: Keine Umrechnung 1: 1-stufige Invertierung 2: 2-stufige Invertierung  Für Mess-, Stell- und Zählpunkt Nummer der Kennlinie, die zur Umrechnung der binären Werte in die phys. Werte angewendet werden soll (vgl. KL-Zeile).

**Hinweis: Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.**

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt      Typ Adr._ URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.  Typ Adr._ URFkt
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt      Typ Adr._ KL_Id
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt     Typ Adr._ KL_Id
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt     Typ Adr._ KL_Id
...
```

Die SPS wird nicht initialisiert.

Das SUCOM-A-Protokoll unterstützt keine spontanen Meldungen, so daß Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen.

Für 1- und 2-stufige Melde- und Schalterpunkte wird zur Umrechnung des SPS-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

```
1-stufige Invertierung:      SPS   FND
                             0 <-> 1
                             1 <-> 0
```

```
2-stufige Invertierung:      SPS   FND
                             0 <-> 2
                             1 <-> 1
                             2 <-> 0
```

**5.4.8.9. Supramat DC97 (Fa. Fröling)**

Grundlage für die Realisierung sind :

- bereitgestellte Dokumentation der Fa. Micro Key, (Version 1.1 vom 3.5.97)

Die DC97 wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

2400 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

Die Supramat wird nicht initialisiert. Das DC97-Protokoll unterstützt mit Ausnahme von Störmeldungen keine spontanen Meldungen, so dass Änderungen der Zustände von Melde- und Schaltpunkte durch ein zyklisches Polling erkannt werden müssen.

Unterstützt werden Werte von 0 bis 8. Verwendet wird grundsätzlich das in der Konfigurations- bzw. Referenz-Datei angegebene FND-Attribut. Für 1- und 2-stufige Melde- und Schaltpunkte wird zur Umrechnung des DC97-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:	DC97	FND	
	0	<->	1
	1	<->	0
2-stufige Invertierung:	DC97	FND	
	0	<->	2
	1	<->	1
	2	<->	0
2er-Grenze:	DC97	FND	DC97
	0	->	0 -> 0
	1	->	0 -> 0
	2	->	1 -> 2
	4	->	1 -> 2
			...

**Hinweis:** Für einzelne Supramat-Datenpunkte ergeben sich hieraus Einschränkungen, wenn der Klartext (z.B. für die "Betriebsart") bzw. die zulässigen DC97-Werte (z.B. >8 bei "Betriebsart") nicht von FND unterstützt werden.

Analog zu den Melde- und Schaltpunkten muß zur Erkennung relevante Änderungen der Zustände von Mess-, Stell- und Zählpunkten zyklisch gepollt werden. Für Mess-, Stell- und Zählpunkte werden nur DC97-Analogwerte unterstützt. Verwendet wird grundsätzlich das in der Konfigurations- bzw. Referenz-Datei angegebene FND-Dimension, unabhängig vom DC97-Code.

**Hinweis:** Der Ersteller der Konfigurations- bzw. Referenz-Datei hat sicherzustellen, dass für alle FND-Datenpunkte nur solche DC97-Adressen definiert werden, deren Werte und FND-Typen vom IZ-Treiber unterstützt werden.

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Für Mess- und Zählpunkte können optional DC97-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die DC97-Adressen werden zyklisch gemäß dem in der ES-Zeile definierten Abfrageintervall gepollt und geprüft.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 106, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :

				<pre> *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0 </pre>	
7	44	256	<=213	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.</p> <p>"COM1" - "COM4" (ohne ")</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die DC97-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"[SS:]AAAA[.LL]" mit :

SS ...: Semaphor (optional)

AAAA : Speicher-Adresse (hexadezimal)

LL ...: Länge in Byte (optional)

Für die folgenden optionalen DC97-Adressen eines Datenpunktes (z.B. Geber-Störung, Grenzwerte) muß nur noch "AAAA" angegeben werden. Die anderen Bestandteile "SS:" werden von der Führungsgröße übernommen. Sie können jedoch auch explizit angegeben werden, wenn sie von dieser abweichen. Bei der Erstellung der Referenzdatei ist sicherzustellen, dass die DC97-Adresse eindeutig zuzuordnen ist.

Für "SS" können die Werte 0 oder 1-8 angegeben werden. Ist keine Semaphor bzw. der Wert 0 angegeben, so erfolgt der Zugriff OHNE vorherige Synchronisation über die Semaphore. Für einen Wert  $1 \leq x \leq 8$  wird auf die Speicheradresse nur nach Freigabe über die Semaphore zugegriffen.

Die Speicher-Adresse "AAAA" muß hexadezimal so angegeben werden, dass dadurch direkt auf einen Wert entsprechend der Datentypen "unsigned/signed char", "unsigned/signed int" und "unsigned/signed long" der Länge "LL" (1, 2 oder 4 Byte) zugegriffen werden kann. Für einen Wert, der Bestandteil einer Datenstruktur (vgl. "typedef" in TYPES.H) ist, ist auf die Adresse der Datenstruktur der Offset für die Position des Wertes in der Datenstruktur zu

addieren. Ist die Länge "LL" nicht angegeben, so ist diese für binäre Werte wie z.B. Melde- und Schalterpunkte, sowie die Geber-Störung 1 Byte. Für analoge Werte werden 2 Byte angenommen.

Für Mess-, Stell- und Zählpunkte kann optional eine Kennlinien zugeordnet werden, die die Umrechnung der binären in die zugeordneten reellen Werte bzw. umgekehrt ermöglicht. Über den Kennlinie-Typ wird unterschieden, ob es sich um einen "unsigned"- oder "signed"-Wert handelt. Ist keine Kennlinie angegeben, so wird ein "unsigned"-Wert angenommen.

Für Mess-, Stell- und Zählpunkte kann optional eine DC97-Adresse für die Geber-Störung und die Grenzwerte angegeben werden. Die Geber-Störung wird nach jedem Lesen oder einer spontanen Meldung erkannt. Die Grenzwert-Überwachung erfolgt nach jedem Lesen durch den IZ-Treiber.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_actual	DC97-Adresse für "actual"
6	40	44	5	UR_Fkt	Nummer der Umrechnungsfunktion des DC97-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) 3: 2er-Grenze

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schalterpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_nominal	DC97-Adresse für "nominal"
6	40	44	5	UR_Fkt	Nummer der Umrechnungsfunktion des DC97-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) 3: 2er-Grenze
7	46	55	10	Adr_actual	DC97-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)



Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_actual	DC97-Adresse für "actual"
6	40	44	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der DC97- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	46	55	10	Adr_fl	DC97-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	57	66	10	Adr_alarm_low	DC97-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	68	77	10	Adr_alarm_high	DC97-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	79	88	10	Adr_warning_low	DC97-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	90	99	10	Adr_warning_high	DC97-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	38	10	Adr_nominal	DC97-Adresse für "nominal"
6	40	44	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der DC97- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
6	46	55	10	Adr_fl	DC97-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
7	57	66	10	Adr_actual	DC97-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

5	29	38	10	Adr_actual	DC97-Adresse für "actual"
6	40	44	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der DC97- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	46	55	10	Adr_f1	DC97-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	57	66	10	Adr_alarm_high	DC97-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	68	77	10	Adr_warning_high	DC97-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Beispiel für eine FND-Referenzdatei :

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt           ADR_actual URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.       ADR_nomin. URFkt ADR_actual
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt           ADR_actual KL_Id ADR_f1_____
ADR_al_____ ADR_ah_____ ADR_wl_____ ADR_wh_____
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt           ADR_nomin. KL_Id ADR_f1_____
ADR_actual
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt           ADR_actual KL_Id ADR_f1_____
ADR_al_____ ADR_wh_____
...
    
```

**5.4.8.10. Infinity (Fa. Andover)**

Grundlage für die Realisierung sind :

- Protokollbeschreibung Fa. AMANN mit Fax vom 21.1.99

Die IZ wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 96, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
7	44	53	10	Benutzername	Benutzername zum Entriegeln der Schnittstelle (Anmelden)
8	55	64	10	Paßwort	Paßwort zum Entriegeln der Schnittstelle (Anmelden)
9	65	256	<=192	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (ohne ")  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für alle Datenpunkt-Typen folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	44	16	MASTER-Adr.	Adresse des MASTER
6	46	61	16	SLAVE-Adr.	Adresse des SLAVE, falls definiert, sonst leer.
7	63	78	16	FG_Name	Name des Datenpunktes für die Führungsgröße ("actual" bei Melde-, Mess- und Zählpunkten, "nominal" bei Schalt- und Stellpunkten)

Ab Spalte 80 werden optionale Parameter angegeben, die für die einzelnen Datenpunkt-Typen spezifisch sind. Werden zusätzliche INFINITY-Datenpunkte zugeordnet, so MÜSSEN diese über denselben MASTER/SLAVE-Namen wie die Führungsgröße erreichbar sein.

Meldepunkt :

8	80	84	5	UR-Fkt.	Nummer der Funktion, die zur Umrechnung der binären Werte verwendet werden soll wie folgt kodiert : 1: 1-stufige Invertierung SONST: Keine Umrechnung
---	----	----	---	---------	---

Schaltpunkt :

8	80	84	5	UR-Fkt.	Nummer der Funktion, die zur Umrechnung der binären Werte verwendet werden soll wie folgt kodiert : 1: 1-stufige Invertierung SONST: Keine Umrechnung
9	86	101	16	ACTUAL_Name	Name des Datenpunktes für die Rückmeldung (actual).

Messpunkt :

8	80	84	5	Kennlinie	Nummer der Kennlinie, die zur Umrechnung der INFINITY-Werte in die FND-Werte angewendet werden soll (vgl. KL-Zeile).
9	86	101	16	AL_Name	Name des Datenpunktes für die untere Alarm-Grenze
10	103	118	16	AH_Name	Name des Datenpunktes für die obere Alarm-Grenze
11	120	135	16	WL_Name	Name des Datenpunktes für die untere Warn-Grenze
12	137	152	16	WH_Name	Name des Datenpunktes für die obere Warn-Grenze

Stellpunkt :

8	80	84	5	Kennlinie	Nummer der Kennlinie, die zur
---	----	----	---	-----------	-------------------------------

					Umrechnung der Infinity-Werte in die FND-Werte angewendet werden soll (vgl. KL-Zeile).
9	86	101	16	ACTUAL_Name	Name des Datenpunktes für die "Rückmeldung" (actual).
Zählpunkt:					
-----					
8	80	84	5	Kennlinie	Nummer der Kennlinie, die zur Umrechnung der Infinity-Werte in die FND-Werte angewendet werden soll (vgl. KL-Zeile).
9	86	101	16	AH_Name	Name des Datenpunktes für die obere Alarm-Grenze
10	103	118	16	WH_Name	Name des Datenpunktes für die obere Warn-Grenze

**Hinweis: Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein (IS-Spaltenangaben + 53 ==> REF-Spalte).**

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt          MASTER-Adr._____ SLAVE-Adr._____
DATENPUNKT-Name_ URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.          MASTER-Adr._____ SLAVE-Adr._____
DATENPUNKT-Name_ URFkt ACTUAL-Name_____
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt          MASTER-Adr._____ SLAVE-Adr._____
DATENPUNKT-Name_ KL_Id ALARM_LOW-Name__ ALARM_HIGH-Name_ WARN._LOW-Name__ WARN._HIGH-Name_
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt          MASTER-Adr._____ SLAVE-Adr._____
DATENPUNKT-Name_ KL_Id ACTUAL-Name_____
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt          MASTER-Adr._____ SLAVE-Adr._____
DATENPUNKT-Name_ KL_Id ALARM_HIGH-Name_ WARN._HIGH-Name_
...
```

Die IZ wird entriegelt, d.h. der Treiber meldet sich mit dem in der IZ-Zeile angegebenen Benutzernamen und Paßwort an. Dabei wird beim ersten Versuch die Kennung "&" verwendet. Erfolgt daraufhin keine Abfrage des Benutzernamens, so wird versucht diese durch Senden von "Befehl" zu aktivieren.

Das INFINITY-Protokoll unterstützt keine spontanen Meldungen, so dass Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen. Aufgrund des begrenzten Empfangspuffers der ANDOVER-Infinity müssen die Abfragen für jeden Datenpunkt in mehrere Teilabfragen aufgeteilt werden.

**Hinweis: Aufgrund der langen Antwortzeiten pro Teilabfrage (bis 15 Sekunden; typisch ca. 1 Sekunde) steigt durch Zeit für einen Zyklus zur Abfrage aller Datenpunkte mit deren Anzahl. Dadurch kann es zu erheblichen Verzögerungen zwischen der Zustandänderung am Geber und der spontanen FND-Meldung kommen.**

Für Melde- und Schaltpunkte wird der Wert "EIN" als 1, alle anderen als 0 interpretiert. Zur Umrechnung des INFINITY-Wertes (IZ) in den zugehörigen FND-Index wird die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:	IZ	FND
	EIN <->	0
	Sonst <->	1

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Die Grenzwertüberwachung wird vom IZ-Treiber emuliert. Werden für Mess- bzw. Zählpunkte INFINITY-Datenpunkte für die Grenzwerte angegeben, so werden zusätzlich die lesenden und modifizierenden Zugriffe auf diesen ausgeführt, wenn der Zugriff gewährt wird.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

#### **5.4.8.11. EY 2400 (Fa. Sauter)**

Grundlage für die Realisierung sind :

- EY 2400 Betrieb der RSZ (Dok.-Nr.: 304 852)
- EY 2400 RAM Substations (Dok.-Nr.: 7000261 003)

Die EY-2400 wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

1200 Baud, 1 Startbit, 7 Datenbit, EVEN-Parity-Bit und 1 Stopbit

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 100, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (ohne ")  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die EY-2400-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"US[:]FA[.AF][,BIT bzw. FMT]" mit :

US : Unterstation	(000,...,999)
FA : Maschinen-Fein-Adresse	(00,...,31, 40,...,71)
AF : Adress-File (Op.-Code)	(00,...,39 reserviert)
BIT: Bit-Nummer	(0,...,7 (8,...,15 reserviert))
FMT: Format	(20,...,255 reserviert)

Wird US und FA ohne trennenden ":" angegeben, so werden die letzten 2 Stellen als FA und der führende Teil als US interpretiert (5-stellige Darstellung gemäß Sauter-interner Kodierung als Gruppen-Adresse).

**Hinweis: Die Angaben für AF und FMT werden momentan nicht unterstützt und bleiben reserviert für zukünftige Erweiterungen !!**

Ist kein AF bzw. Format angegeben so gelten folgende Voreinstellungen :

- actual (Melde- und Schaltpunkte)	AF 8	1 Bit gemäß "BIT"
- actual (Mess- und Stellpunkte)	AF 1	4 Digit
- actual (Zählpunkte)	AF 4+5	8 Digit
- nominal (Schaltpunkte)	AF 10	1 Bit gemäß "BIT"
- nominal (Stellpunkte)	AF 13	4 Digit

Für Melde- und Schaltpunkte können n-stufige ( $1 \leq n \leq 8$ ) Datenpunkte angegeben werden, die auf n 1-stufige Datenpunkte abgebildet werden, wobei max. ein Datenpunkt gesetzt sein kann. Dazu sind die EY-2400-Bit-Nummern derselben FA in aufsteigender Reihenfolge implizit zugeordnet.

Beispiel für 8-stufigen Meldepunkt:

US:FA,Bit :	Stufe 0/I	(nur diese EY-2400-Adresse wird angegeben;
US:FA,Bit+1:	Stufe II	ist keines der einzelnen Bits gesetzt, so
...		entspricht dieses Stufe 0 )
US:FA,Bit+7:	Stufe VIII	



Für alle Datenpunkt-Typen kann optional ein EY-2400-Bit-Adresse für die Geber-Störung angegeben werden. Diese MUSS derselben US zugeordnet sein. Die Geber-Störung wird nach jedem Lesen/Polling geprüft.

Für Mess-, Stell- und Zählpunkte muss eine Kennlinien zugeordnet werden. Diese legt den Offset und die Steilheit bezogen auf "AF", "FMT" und die projektspezifischen, anlageninternen Vereinbarungen fest (vgl. Sauter-Dokumentationen, z.B. Dok.-Nr. 304 852 unter 6.0 ff.). Für Formate über mehrere EY-2400-Datenworte, z.B. Zähler aus AF 4+5, wird die Steilheit an der Stelle 1 als Faktor auf den EY-2400-Wert angewendet.

Für Mess-, Stell- und Zählpunkte kann optional eine EY-2400-Adresse für jeden Grenzwert angegeben werden. Diese MUSS derselben US zugeordnet sein. Die Grenzwert-Überwachung erfolgt nach jedem Lesen/Polling durch den IZ-Treiber.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	Adr_actual	EY-2400-Adresse für "actual"
6	43	47	5	UR_Fkt	Nummer der Umrechnungsfunktion des EY-2400-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)
7	49	61	13	Adr_f1	EY-2400-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	Adr_nominal	EY-2400-Adresse für "nominal"
6	43	47	5	UR_Fkt	Nummer der Umrechnungsfunktion des EY-2400-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)
7	49	61	13	Adr_f1	EY-2400-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	63	75	13	Adr_actual	EY-2400-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	Adr_actual	EY-2400-Adresse für "actual"
6	43	47	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der EY-2400- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	49	61	13	Adr_f1	EY-2400-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	63	75	13	Adr_alarm_low	EY-2400-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	77	89	13	Adr_alarm_high	EY-2400-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	91	103	13	Adr_warning_low	EY-2400-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	105	117	13	Adr_warning_high	EY-2400-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	Adr_nominal	EY-2400-Adresse für "nominal"
6	43	47	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der EY-2400- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
6	49	61	13	Adr_f1	EY-2400-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
7	63	75	13	Adr_actual	EY-2400-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	Adr_actual	EY-2400-Adresse für "actual"
6	43	47	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der EY-2400- in die physikalischen Werte angewendet werden soll (Entspricht

					momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	49	61	13	Adr_f1	EY-2400-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	63	75	13	Adr_alarm_high	EY-2400-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	77	89	13	Adr_warning_high	EY-2400-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis: Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.**

Beispiel für eine FND-Referenzdatei :

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt           ADR_actual... URFkt ADR_f1
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpnt.       ADR_nominal.. URFkt ADR_f1
ADR_actual...
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt           ADR_actual... KL_Id ADR_f1
ADR_al..... ADR_ah..... ADR_wl..... ADR_wh.....
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt           ADR_nominal.. KL_Id ADR_f1
ADR_actual...
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt           ADR_actual... KL_Id ADR_f1
ADR_al..... ADR_wh.....
...
    
```

Die EY-2400 wird nicht initialisiert. Zur Synchronisation wird ca. 5 Sekunden mit dem Senden der nächsten Zeichenfolge gewartet.

Das EY-2400-Protokoll unterstützt keine spontanen Meldungen, so dass Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen. Daraus entsteht eine permanente Belastung des EY-2400-Systems als auch des GA-Knotens.

Für 1- und 2-stufige Melde- und Schaltpunkte wird zur Umrechnung des EY-2400-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:	EY-2400	FND
	0	<-> 1
	1	<-> 0
2-stufige Invertierung:	EY-2400	FND
	0	<-> 2

1 <-> 1

2 <-> 0

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Die Grenzwertüberwachung wird vom IZ-Treiber emuliert.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

**5.4.8.12. EY 3600 (Fa. Sauter)**

Grundlage für die Realisierung sind :

- EY 3600 Router EYZ 291 (Dok.-Nr.: AN-LT-4220)
- EY 3600 ISA-Bu-Controller EYS 290 (Dok.-Nr.: AN-LT-4033c)
- EY 3600 Kommunikation und Datenaufbau (Dok.-Nr.: AN-LT-4007d)
- Software-Auftrag-Nr.00/104
- Software-Auftrag mit Bestellung 555888 vom 21.10.2005

Die EY-3600 wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

19200 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

**Hinweis: Die Baudrate MUSS am EYZ 291 auf der Rückseite entsprechend eingestellt werden (DIP-Schalter!). Die EY-3600-Kommunikations-Prozedur erfordert ein ausreichend schnelles Auslesen des EYZ 291. Dieses setzt neben der hohen Baudrate auch einen ausreichend schnellen Rechner voraus**

(>=400 MHz), bei dem auch entsprechende Beschleunigungs-Maßnahmen (z.B. Cache für Speicher- und Datei-Zugriffe) aktiviert sind.

Desweiteren sollten die Schnittstellen mit den höherpriorisierten Interrupts 10 und 11 (COM3/4) und UART-Bausteinen mit FIFO (16550A) verwendet werden.

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 102, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
7	44	46	3	PC/CU-Gruppenadr.	PC/CU-Gruppenadresse (0-255)
8	48	256	<=209	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (ohne ")

Wird durch das Zeilenabschluß-  
zeichen (<CR> und/oder <LF>,  
je nach verwendetem Editor)  
abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die EY-3600-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"US[:]FA[.DW][.BIT bzw. FMT]" mit :

US : Unterstation	(00000,...,32767)
FA : Maschinen-Fein-Adresse	(0,...,255)
DW : Datenwort	(000,...,127)
BIT: Bit-Nummer	(0,...,31)
FMT: Format	(reserviert)

Wird US und FA ohne trennenden ":" angegeben, so werden die letzten 2 Stellen als FA und der führende Teil als US interpretiert (8-stellige Darstellung gemäß Sauter-interner Kodierung als Gruppen-Adresse).

**Hinweis: Die Angaben für FMT werden momentan nicht unterstützt und bleiben reserviert für zukünftige Erweiterungen !!**

Ist kein DW bzw. Bit/Format angegeben so gelten folgende Voreinstellungen :

- actual (Melde- und Schaltpunkte) :	DW 2	1 Bit gemäß "BIT", sonst Bit 31
- actual (Mess- und Stellpunkte) ..:	DW 3	4 Byte (FP-Format)
- actual (Zählpunkte) .....	DW 6	4 Byte (FP-Format)
- nominal (Schaltpunkte) .....	DW 8	1 Bit gemäß "BIT", sonst Bit 31
- nominal (Stellpunkte) .....	DW 8	4 Byte (FP-Format)

Für Melde- und Schaltpunkte können n-stufige (1<=n<=6) Datenpunkte angegeben werden, wenn BIT gleich 31 ist. Ansonsten wird das angegebene Bit als 1-stufiger Datenpunkt zugeordnet. Schaltbefehle auf der gleichen FA beeinflussen sich gegenseitig, d.h. führen zum Rücksetzen der anderen Bits.

Für alle Datenpunkt-Typen kann optional ein EY-3600-Bit-Adresse für die Geber-Störung angegeben werden. Diese MUSS derselben US zugeordnet sein. Die Geber-Störung wird nach jedem Lesen/Polling geprüft.

Für Mess-, Stell- und Zählpunkte kann eine Kennlinien zugeordnet werden. Diese legt den Offset und die Steilheit bezogen auf "DW", "FMT" und die projektspezifischen, anlageninternen Vereinbarungen fest.

Für Mess- und Zählpunkte kann optional eine EY-3600-Adresse für jeden Grenzwert angegeben werden. Diese MUSS derselben US zugeordnet sein. Die Grenzwert-Überwachung erfolgt nach jedem Lesen/Polling durch den IZ-Treiber.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Zeichen mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Zeichen 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	45	17	Adr_actual	EY-3600-Adresse für "actual"
6	47	51	5	UR_Fkt	Nummer der Umrechnungsfunktion des EY-3600-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) (Dieser Parameter kann optional angegeben werden)
7	53	69	17	Adr_f1	EY-3600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Zeichen 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	45	17	Adr_nominal	EY-3600-Adresse für "nominal"
6	47	51	5	UR_Fkt	Nummer der Umrechnungsfunktion des EY-3600-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) (Dieser Parameter kann optional angegeben werden)
7	53	69	17	Adr_f1	EY-3600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	71	87	17	Adr_actual	EY-3600-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Zeichen 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	45	17	Adr_actual	EY-3600-Adresse für "actual"
6	47	51	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der EY-3600- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	53	69	17	Adr_fl	EY-3600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	71	87	17	Adr_alarm_low	EY-3600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	89	105	17	Adr_alarm_high	EY-3600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	107	123	17	Adr_warning_low	EY-3600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	125	141	17	Adr_warning_high	EY-3600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Zeichen 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	45	17	Adr_nominal	EY-3600-Adresse für "nominal"
6	47	51	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der EY-3600- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
6	53	69	17	Adr_fl	EY-3600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
7	71	87	17	Adr_actual	EY-3600-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Zeichen 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	45	17	Adr_actual	EY-3600-Adresse für "actual"
6	47	51	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der EY-3600- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur)



					(Dieser Parameter kann optional angegeben werden)
7	53	69	17	Adr_f1	EY-3600-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	71	87	17	Adr_alarm_high	EY-3600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	89	105	17	Adr_warning_high	EY-3600-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis: Der Aufbau der IS-Zeile ab Zeichen 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Zeichen 82 überein.**

Beispiel für eine FND-Referenzdatei :

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt          ADR_actual..... URFkt ADR_f1
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.      ADR_nominal..... URFkt ADR_f1
ADR_actual.....
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt          ADR_actual..... KL_Id ADR_f1
ADR_al..... ADR_ah..... ADR_wl..... ADR_wh.....
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt      ADR_nominal..... KL_Id ADR_f1
ADR_actual.....
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt        ADR_actual..... KL_Id ADR_f1
ADR_al..... ADR_wh.....
...
    
```

Der EYZ 291 wird nach Neustart und Kommunikations-Störungen gemäß AN-LT-4220 unter Abschnitt 9.4 initialisiert.

Das EY-3600-Protokoll unterstützt keine nutzbaren spontanen Meldungen, so dass Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen. Daraus entsteht eine permanente Belastung des EY-3600-Systems als auch des GA-Knotens.

Mit Version 1.3i (Software-Auftrag mit Bestellung 555888 vom 21.10.2005) wird das Polling wie folgt optimiert:

- Abbruch der Timeout- und Wiederholungs-Behandlung, falls ein Fehler im "Attention-Byte Status to Host-PC" (Comx) angezeigt wird (z.B. Arbeitsstation bzw. Einzelraum-Regler nicht vorhanden, wie im Projekt "Agilolfinger Platz").
- Polling durch Zusammenfassung von bis zu 28 Datenpunkten mittels "Random Collect"
- Auswertung von "Multiple Random Drop"-Telegrammen, die durch Anmeldung eines anderen Teilnehmer - z.B. NovaPro - initiiert wurden. Eine eigene Anmeldung mit Behandlung der Garbage-Collection wurde nicht vorgesehen.

Für 1- und 2-stufige Melde- und Schalterpunkte wird zur Umrechnung des EY-3600-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:            EY-3600        FND

0 <-> 1

1 <-> 0

2-stufige Invertierung:            EY-3600        FND

0 <-> 2

1 <-> 1

2 <-> 0

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Die Grenzwertüberwachung wird vom IZ-Treiber emuliert.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

#### **5.4.8.13.        IPC (Fa. SE-Electronic)**

Grundlage für die Realisierung sind :

- Protokollbeschreibung 21.1.2000

Die IPC wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 112, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (ohne "  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>), je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die IPC-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"US.DP#" mit :

US : Unterstation (01,...,63)

DP#: Datenpunkt-Nummer (1,...,255)

Für Melde- und Schaltpunkte können n-stufige ( $1 \leq n \leq 8$ ) Datenpunkte angegeben werden, die auf n 1-stufige Datenpunkte abgebildet werden, wobei max. ein Datenpunkt gesetzt sein kann. Dazu sind die IPC-Adressen in aufsteigender Reihenfolge implizit zugeordnet.

Beispiel für 8-stufigen Meldepunkt:

US.DP# : Stufe I	(nur diese IPC-Adresse wird angegeben;
US.DP#+1: Stufe II	ist keiner der einzelnen Datenpunkte
...	gesetzt => Stufe 0 )
US.DP#+7: Stufe VIII	

Für alle Datenpunkt-Typen kann optional eine IPC-Adresse für die Geber-Störung angegeben werden. Die Geber-Störung wird nach jedem Lesen/Polling geprüft.

Für Mess-, Stell- und Zählpunkte muss eine Kennlinien zugeordnet werden.

Für Mess-, Stell- und Zählpunkte kann optional eine IPC-Adresse für jeden Grenzwert angegeben werden. Die Grenzwert-Überwachung erfolgt nach jedem Lesen/Polling durch den IZ-Treiber.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	34	6	Adr_actual	IPC-Adresse für "actual"
6	36	40	5	UR_Fkt	Nummer der Umrechnungsfunktion des IPC-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)

7	42	47	6	Adr_f1	IPC-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
---	----	----	---	--------	--

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	34	6	Adr_nominal	IPC-Adresse für "nominal"
6	36	40	5	UR_Fkt	Nummer der Umrechnungsfunktion des IPC-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)
7	42	47	6	Adr_f1	IPC-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	49	54	6	Adr_actual	IPC-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	34	6	Adr_actual	IPC-Adresse für "actual"
6	36	40	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der IPC- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	42	47	6	Adr_f1	IPC-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	49	54	6	Adr_alarm_low	IPC-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	56	61	6	Adr_alarm_high	IPC-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	63	68	6	Adr_warning_low	IPC-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	70	75	6	Adr_warning_high	IPC-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	34	4	Adr_nominal	IPC-Adresse für "nominal"
6	36	40	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der IPC- in die physikalischen Werte angewendet werden soll

					det werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
6	42	47	6	Adr_f1	IPC-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
7	49	54	6	Adr_actual	IPC-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	34	6	Adr_actual	IPC-Adresse für "actual"
6	36	40	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der IPC- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	42	47	6	Adr_f1	IPC-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	49	54	6	Adr_alarm_high	IPC-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	56	61	6	Adr_warning_high	IPC-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis: Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.**

Beispiel für eine FND-Referenzdatei :

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt          ADR_a. URFkt ADR_f1
00002 Schaltpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schaltpkt.          ADR_n. URFkt ADR_f1 ADR_a.
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt          ADR_a. KL_Id ADR_f1 ADR_al ADR_ah
ADR_wl ADR_wh
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt          ADR_n. KL_Id ADR_f1 ADR_a.
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt          ADR_a. KL_Id ADR_f1 ADR_al ADR_wh
...
    
```

Die IPC wird nicht initialisiert. Zur Synchronisation wird ca. 5 Sekunden mit dem Senden der nächsten Zeichenfolge gewartet.

Das IPC-Protokoll unterstützt keine spontanen Meldungen, so dass Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen. Daraus entsteht eine permanente Belastung des IPC-Systems als auch des GA-Knotens.

Für 1- und 2-stufige Melde- und Schaltpunkte wird zur Umrechnung des IPC-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:           IPC    FND

0 <-> 1

1 <-> 0

2-stufige Invertierung:        IPC    FND

0 <-> 2

1 <-> 1

2 <-> 0

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Die Grenzwertüberwachung wird vom IZ-Treiber emuliert.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

#### **5.4.8.14.        ACCESS 3000 (Fa. Multitone)**

Grundlage für die Realisierung sind :

- Protokollbeschreibung "Entwickler-Paket D3NA, Ausgabe 1" vom August 1991

Die A3K wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert:

1200 Baud, 1 Startbit, 7 Datenbit, EVEN-Parity-Bit und 2 Stopbit

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 5, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen PSE-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der PSE-IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem PSE/IZ-Typ.  "COM1" - "COM4" (ohne "  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>), je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).



Alle PSE-Funktionen werden auf FND-Schaltpunkte abgebildet. Dabei wird jedem FND-Schaltpunkt eine Datei mit spezifischer Extension im Unterverzeichnis IZ/A3K zugeordnet. Die Dateiformate sind allgemeingültig ausgelegt, so dass eine spätere Anpassung bzw. Erweiterung einfach möglich ist.

PSE-Funktion	Extension	FND-Schaltpunkt
Teilnehmer-/Team-Ruf	.RUF	max. 8-stufig
Teilnehmer-Status abfragen/ändern	.STA	max. 8-stufig
Teilnehmer-Rufumleitung	.UML	max. 8-stufig

Abhängig vom zugeordneten FND-Attribut wird die Anzahl der Stufen festgelegt. Den zugehörigen FND-Indizes wird in der Datei zeilenweise der zugehörige Parametersatz zugeordnet, falls dem FND-Index keine programmspezifische Funktion zugeordnet ist. Die Reihenfolge der Einträge in den Dateien ist frei. Kommentarzeilen sind durch ein "!" am Zeilenanfang gekennzeichnet. Ist dem FND-Index 0 in der Datei keine Funktion zugeordnet, so wird ggf. eine programmspezifische Funktion wie folgt zugeordnet:

PSE-Funktion	Index 0
Teilnehmer-/Team-Ruf	autom. über alle Indizes, bis Erfolg KEINE; d.h. Index 0 MUSS explizit angegeben werden
Teilnehmer-Status abfragen/ändern	
Teilnehmer-Rufumleitung	autom. über alle Indizes, bis Erfolg

Die RUF-Datei hat folgenden Zeilenaufbau:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Index	FND-Index (000 - 008)
2	5	9	5	T-Adresse	Teilnehmer- bzw. Team-Adresse. Beginnt diese mit einem "*", so handelt es sich um eine Team-Adresse, sonst Teilnehmer
3	11	13	3	Rufton	Nummer für Rufton wie folgt kodiert: A3K: 1 wie in PSE hinterlegt ... 8 wie in PSE hinterlegt
4	15	17	3	Priorität	Priorität wie folgt kodiert: A3K : 1 niedrig (L) 2 normal (N) 3 hoch (H) 4 Notfall (E)
5	19	256	<=238	Nachricht	Auszugebende Nachricht mit Steuerzeichen gemäß ANSI-C (z.B. \r = Carriage Return) A3K: max. 60 darstellbare Zeichen

Stellen die ersten 16 Zeichen der Nachricht eine Datenpunkt-Adresse dar, so wird die Nachricht bestehend aus der Datenpunkt-Adresse, -Beschreibung und der Meldung (Attributtext bzw. Wert mit Dimension) automatisch erzeugt und die nachfolgenden Zeichen angefügt.

Ist die Nachricht leer, so wird der Ruf an die PSE unterdrückt und im RSP-Datagramm das tl-Bit (temporäre Verriegelung) gesetzt. Dadurch ist es möglich, Nachrichten für einzelne FND-Indizes (z.B. Rücksetzen auf NORMAL) zu unterdrücken.

Die STA-Datei hat folgenden Zeilenaufbau:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Index	FND-Index (000 - 008)
2	5	9	5	T-Adresse	Teilnehmer- bzw. Team-Adresse. Beginnt diese mit einem "*", so handelt es sich um eine Team-Adresse, sonst Teilnehmer A3K: Nur Teilnehmer zulässig
3	11	13	3	Status	Nummer für eingeschränkten Status beginnend mit 0 wie folgt kodiert: A3K : 0 EIN (anwesend,aktiv) 1 AUS (abwesend,passiv) 2 REGAL abwesend

Die UML-Datei hat folgenden Zeilenaufbau:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Index	FND-Index (000 - 008)
2	5	9	5	T-Adr.Quelle	Teilnehmer- bzw. Team-Adresse der Quelle für die Umleitung. Beginnt diese mit einem "*", so handelt es sich um eine Team-Adresse, sonst Teilnehmer A3K: Nur Teilnehmer zulässig
3	11	15	5	T-Adr. Ziel	Teilnehmer- bzw. Team-Adresse. des Ziels für die Umleitung. Beginnt diese mit einem "*", so handelt es sich um eine Team-Adresse, sonst Teilnehmer. Ist kein Ziel angegeben, so wird die Rufumleitung gelöscht A3K: Nur Teilnehmer zulässig

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	40	12	A3K-Adresse	Datei mit oben genannter Extension, Format wie nachfolgend beschrieben

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

Es wird empfohlen, jedem Schaltpunkt eine eigene Datei zuzuordnen. Dadurch besteht die Möglichkeit bei GA-Knoten mit Mehrbenutzer-Betriebssystem (z.B. LINUX) unter dem Programmlauf die Dateien auszutauschen. Hierzu wird empfohlen, den Dateinamen in Anlehnung an die Datenpunkt-Adresse zu vergeben (z.B. "aamm" mit "aa" und "mm" gemäß der GLT-Richtlinien).

Beispiel für eine FND-Referenzdatei :

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Schaltpunkt_1_RF 2 0 15 15 15 0 0 015 008 Funktionsruf          dddddd.RUF
00002 Schaltpunkt_2_RG 2 0 15 15 15 0 0 015 008 Gruppenruf           dddddd.RUF
00003 Schaltpunkt_3_RP 2 0 15 15 15 0 0 015 008 Personenruf          dddddd.RUF
00004 Schaltpunkt_4_RT 2 0 15 15 15 0 0 015 008 Teilnehmerruf        dddddd.RUF
00005 Schaltpunkt_5_RS 2 0 15 15 15 0 0 015 024 Teilnehmer-Status     dddddd.STA
00006 Schaltpunkt_6_RU 2 0 15 15 15 0 0 015 008 Teilnehmer-Rufumleitung dddddd.UML
...
```

Die A3K wird nicht initialisiert.

Das A3K-Protokoll unterstützt keine spontanen Meldungen, so dass Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen. Daraus entsteht eine permanente Belastung des A3K-Systems als auch des GA-Knotens.

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Die Grenzwertüberwachung entfällt.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

#### 5.4.8.15. MILES Arbeitsstation (Fa. Messner)

Grundlage für die Realisierung sind :

- Protokollbeschreibung MESSNER-Automationsstation vom 14.4.1999
- Mehrstufige Melde- und Schaltpunkte für MILES-Systeme der Fa. Messner am GA-Knoten vom 8.4.2004

Die Messner Arbeitsstation (MAS) wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 2 Stopbit

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 72,...,75, sonst: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** stets 0

7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  "COM1" - "COM4" (ohne ")  Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.
---	----	-----	-------	-----------------	--

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die MAS-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"M/A:G.NN" mit :

M : Multiplexer	(MX: 1,...,8)
A : Automationsstation	(AS: 1,...,8)
G : Gruppen-Adresse	(DI: 1,...,4; DO: 5,6; AI/AO: 7,...,9; Index für Zähler: A,...,F für 0,...,5 )
NN: lfd. Nummer	(1,...,48 bzw. 96)

**Hinweis: Als Kurzform ist auch das MESSNER-interne Format "MAGNN" zulässig. Dabei wird die Gruppen-Adresse "Z" auf "B" (Impulszähler) abgebildet.**

**Auf Datenpunkte mit Gruppen-Adresse "I" kann über das bisher offengelegte Protokoll nicht zugegriffen werden. Durch Parametrierung des MAS-Systems müssen diese Datenpunkte auf verfügbare Adressbereiche abgebildet werden.**

Technisch bedingt unterstützen MILES-Systeme für digitale Ein-/Ausgänge oder Pseudo-Datenpunkte nur 1-stufige Zustände mit je einer zugeordneten Maschinen-Adresse. Sollen mehrstufige Zustände abgebildet werden, so wird jeder Stufe eine eigene Maschinen-Adresse zugeordnet.

Die Anzahl der Meldungsstufen der FND-Melde- bzw. Schaltpunkte ergibt sich aus dem in der FND-Referenzdatei zugeordneten Attribut (vgl. "atb\_id"). Die in der FND-Referenzdatei angegebene Maschinen-Adresse definiert die Adresse für die ERSTE Meldungsstufe. Die FND-SSA/LIB ordnet die nachfolgenden Meldungsstufen in aufsteigender Reihenfolge implizit den nachfolgenden Maschinen-Adressen in aufsteigender Reihenfolge zu.

Alle zum n-stufigen FND-Datenpunkt gehörenden Maschinen-Adressen müssen der gleichen Gruppen-Adresse zugeordnet sein, d.h. die Angaben für Multiplexer, Arbeitsstation und Gruppe müssen in den Maschinen-Adressen übereinstimmen (vgl. Adressteil "M/A:G" in der Systembeschreibung des GA-Knotens zum FND-SSA/LIB).

Die MSR-Anlage stellt die Plausibilität und korrekte Funktion sicher. Unzulässige Vorgabewerte werden ignoriert.

Der FND-SSA/LIB liefert bei unzulässigen Kodierungen den "ungültigen" Wert (alle Bits gesetzt) als aktuelle FND-Meldestufe an den GA-Knoten.

Für Mess-, Stell- und Zählpunkte kann optional eine Kennlinien zugeordnet werden. Dieses kann erforderlich sein, wenn die MAS-System von der FND-Spezifikation abweichende Dimensionen verwendet.

Für Mess-, Stell- und Zählpunkte kann optional eine MAS-Adresse für die Geber-Störung und die Grenzwerte angegeben werden. Die Geber-Störung wird nach jedem Lesen oder einer spontanen Meldung erkannt. Die Grenzwert-Überwachung erfolgt nach jedem Lesen durch den IZ-Treiber.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_actual	MAS-Adresse für "actual"
6	38	42	5	UR_Fkt	Nummer der Umrechnungsfunktion des MAS-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)
7	44	51	8	Adr_f1	MAS-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_nominal	MAS-Adresse für "nominal"
6	38	42	5	UR_Fkt	Nummer der Umrechnungsfunktion des MAS-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)

7	44	51	8	Adr_f1	MAS-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	53	60	8	Adr_actual	MAS-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_actual	MAS-Adresse für "actual"
6	38	42	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der MAS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	44	51	8	Adr_f1	MAS-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	53	60	8	Adr_alarm_low	MAS-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	62	69	8	Adr_alarm_high	MAS-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	71	78	8	Adr_warning_low	MAS-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	80	87	8	Adr_warning_high	MAS-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	36	8	Adr_nominal	MAS-Adresse für "nominal"
6	38	42	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der MAS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
6	44	51	8	Adr_f1	MAS-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
7	53	60	8	Adr_actual	MAS-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

5	29	36	8	Adr_actual	MAS-Adresse für "actual"
6	38	42	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der MAS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	44	51	8	Adr_f1	MAS-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	53	60	8	Adr_alarm_high	MAS-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	62	69	89	Adr_warning_high	MAS-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis: Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.**

Beispiel für eine FND-Referenzdatei :

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt           ADR_act.  URFkt  ADR_f1__
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.       ADR_nom.  URFkt  ADR_f1__ ADR_act.
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt           ADR_act.  KL_Id  ADR_f1__ ADR_al__
ADR_ah__ ADR_wl__ ADR_wh__
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt           ADR_nom.  KL_Id  ADR_f1__ ADR_act.
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt           ADR_act.  KL_Id  ADR_f1__ ADR_al__
ADR_wh__
...
    
```

Die MAS wird nicht initialisiert. Zur Synchronisation wird ca. 5 Sekunden mit dem Senden der nächsten Zeichenfolge gewartet.

Das MAS-Protokoll unterstützt keine spontanen Meldungen, so dass Änderungen der Zustände von Datenpunkten durch ein zyklisches Polling erkannt werden müssen. Daraus entsteht eine permanente Belastung des MESSNER-Systems als auch des GA-Knotens.

Für 1-stufige Melde- und Schalterpunkte wird zur Umrechnung des MAS-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:

MAS	FND
0	<-> 1
1	<-> 0



Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe.

Die Grenzwertüberwachung wird vom IZ-Treiber emuliert.

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

#### **5.4.8.16. BACnet/IP**

Grundlage für die Realisierung sind :

- ASHRAE 135-2004
- Angebot JCISHM/5011 vom 27.01.2005
- Die von JCI bereitgestellte Testkonfiguration bestehend aus einem MS-NAE5520-0E dient als Referenzsystem.  
Testweise wurden am Treiber auch Produkte folgender Hersteller erfolgreich aufgeschaltet :
  - Andover
  - MBS
  - Neuberger
  - SAIA
  - Sauter

**Hinweis:** O.g. Aufschaltungen zeigten, dass herstellerspezifische Besonderheiten - technisch und organisatorisch - sowie abweichende Stände und Auslegungen der BACnet-Spezifikation system- und projekt-spezifische Festlegungen erforderlich machen.

Es wird empfohlen, VOR jedem Einsatz des Treibers die Eignung des vorgesehenen BACnet-Produkts im Rahmen einer Testaufschaltung sicherzustellen.

Folgende Erweiterungen bzw. Abweichungen wurden festgelegt :

- Vendor-Id. für Treiber-Id. 32 (IZ20 in GAK\_XXL.EXE) : 195 (AMEV Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen). Sonst (IZnn, mit nn != "20"): 201 (DTF Daten - Technik FRIES)
- Polling von "Event\_State", "Out\_Of\_Service" entfällt, da in "Status\_Flags" enthalten

- Object-Type "Multistate Value" wird zusätzlich unterstützt

Folgende Standard Object Types lassen sich auf Datenpunkte abbilden :

- Binary Input (BI <=> i.a. 1-stuf. Meldepunkt)
- Binary Output (BO <=> i.a. 1-stuf. Schaltpunkt)
- Binary Value (BV <=> i.a. 1-stuf. Schaltpunkt)
- Analog Input (AI <=> i.a. Messpunkt)
- Analog Output (AO <=> i.a. Stellpunkt)
- Analog Value (AV <=> i.a. Stellpunkt)
- MultiState Input (MI <=> i.a. n-stuf. Meldepunkt))
- MultiState Output (MO <=> i.a. n-stuf. Schaltpunkt)
- MultiState Value (MV <=> i.a. n-stuf. Schaltpunkt)

Unterstützt wird die Status- und Ereignis-Behandlung, explizite Rückmeldungen/Istwerte, Grenzwerte sowie das (Rück-) Setzen von Zählpunkten für die Verbrauchsmessung. Neben den aktuellen Werten ("Present\_Value") können auch andere numerische BACnet-Eigenschaften wie "Change\_Of\_State\_Count" auf Datenpunkte abgebildet werden.

Unterstützt ein BACnet-Gerät (Gerät B, Server) das "Dynamic Device Binding", so kann über das Zusatzprogramm BN2REF die Vorlage für die Referenzdatei automatisch erstellt werden. Optional können zusätzliche BACnet-Geräte explizit angegeben werden.

Für jedes bekannte BACnet-Gerät werden die verfügbaren binären und analogen Objekte inkl. deren Eigenschaften abgefragt. Die Inhalte der Eigenschaften "Object\_Name" und "Description" werden - soweit verfügbar - als Datenpunkt-Adresse und zugehörige Beschreibung übernommen. Der Datenpunkt-Typ wird direkt aus dem "Object\_Type" abgeleitet. Analog wird beim Meldungstext bzw. der physikalischen Einheit und anderen Eigenschaften verfahren.

Schalt- und Stellpunkten kann eine zusätzliche BACnet-Eigenschaft als Rückmeldung bzw. expliziter Istwert zugeordnet werden. Dadurch ist es möglich, funktionell zusammengehörige BACnet-Objekte auf einen Datenpunkt abzubilden.

Dem Wert des Datenpunktes wird als Voreinstellung der "Present\_Value" des BACnet-Objektes zugeordnet. Dieser kann bei Bedarf in jede verfügbare numerische BACnet-Eigenschaft geändert werden.

Während des Programmlaufs bildet der FND-SSA/LIB automatisch die verfügbaren Status- und Ereignis-Informationen des BACnet-Objektes ("Status\_Flags") auf die entsprechenden Informationen des Datenpunktes ab und löst ggf. die zugeordnete Spontanmeldung aus. Der FND-SSA/LIB erkennt automatisch, wenn ein BACnet-Objekt wesentliche Wertänderungen (Change\_Of\_Value; COV) melden kann und nutzt diesen Dienst.

Für Mess- und Zählpunkte erkennt der FND-SSA/LIB automatisch, ob ein BACnet-Objekt Grenzwerte ("High\_Limit", "Low\_Limit") unterstützt. Sind diese verfügbar, werden deren Werte übernommen und können - soweit schreibende Zugriffsrechte zugeordnet wurden - über den GALLUX verändert werden. Werden Grenzwerte nicht unterstützt, so verwaltet und überwacht sie der FND-SSA/LIB und erzeugt ggf. die zugehörigen Spontanmeldungen.

Der GALLUX gleicht seine Systemzeit bei der Kommunikation mit der Leitzentrale oder einem Zeit-Server ab. Der FND-SSA/LIB kann diese an die dafür vorbereiteten BACnet-Geräte weiterleiten (BIBB DM-TS-A). Damit wird die Systemzeit im Gesamtsystem synchronisiert.

**LAN-Schnittstelle zum Anschluß der IZ**

Die BACnet-Unterstation werden über ein lokales TCP/IP-Netzwerk (BACnet/IP) unter LINUX angeschlossen. Das Betriebssystem DOS wird NICHT unterstützt.

Dieser IZ-Treiber ist gleichzeitig BACnet-Router für ein eigenes BACnet-Netzwerk, an dem nur der im Treiber integrierte Device (BACnet-Client) "angeschlossen" ist.

**Hinweis :** Bei der Einrichtung des Rechners ist darauf zu achten, dass dieser dem bzw. den Netzwerken zugeordnet ist, über die die BACnet-Geräte erreichbar sind (vgl. "hostname -i"). Ggf. sind in der Firewall die verwendeten Ports (z.B. TCP-Port 47808) freizuschalten.

**Spezifische Anpassungen in der Konfigurationsdatei**

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige

					Erweiterungen
					*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 32, 33 *** sonst ..: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen DDC-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNC: stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNC: 40 (UDP/IP)
6	40	42	3	Protokoll	Verbindungs-Protokoll der US *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNC: 0
7	44	50	7	SDEV	Eigene Device-Instance gemäß so wird der BNC als lokales BACnet-Device betrieben.
9	58	69	12	SADR	Eigene MAC gemäß BACnet-Vorgaben (hexadezimal). Fehlt diese, so wird SADR aus IP-Adresse und -Port gebildet
10	71	256	<=190	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  Für UDP/IP Port im Format "[IP-Netz#]Port" mit IP-Netz aaa.bbb.ccc.ddd (IP4-Adr.) und "Port" gemäß BACnet- Vorgaben, i.a. 47808 (0xBAC0)  Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

**Hinweis:** Die Werte von SDEV, SNET und SADR müssen - wie auch IP-Adresse und -Port - mit dem System-Verantwortlichen abgestimmt sein, um Konflikte mit bestehenden und zukünftigen BACnet-Geräten zu

vermeiden !!! Es wird empfohlen, jedes BACnet-basierte MSR-System über eine gesonderte Netzwerkkarte am GA-Knoten aufzuschalten, um gegenseitige Beeinflussungen und das daraus resultierende "unbegrenzte" Störpotential weitestgehend auszuschließen.

Die BACnet-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

**"ddddddd.tttt#ooooooo[’ppppp]"** mit :

- ddddddd:** DEVICE-Id. (Instance des DEVICE-Objektes)
- tttt:** OBJEKT-Typ (Object-Type des Object-Identifier) wie folgt :
  - AI Analog Input (Synonym für 0000)
  - AO Analog Output (Synonym für 0001)
  - AV Analog Value (Synonym für 0002)
  - BI Binary Input (Synonym für 0003)
  - BO Binary Output (Synonym für 0004)
  - BV Binary Value (Synonym für 0005)
  - MI Multi-State Input (Synonym für 0013)
  - MO Multi-State Output (Synonym für 0014)
  - MV Multi-State Value (Synonym für 0019)
  - nnnn max. 4-stellige dezimale Nummer des Object-Type gemäß "BACnetObjectType"

- ooooooo:** OBJEKT-Nr. (Instance des Object-Identifier)
- ppppp:** PROPERTY-Id. Fehlt diese Angabe, so wird der "Present\_Value" (00085) der Führungsgröße bzw. dem explizit angegebenen "actual" zugeordnet

Für Melde- und Schaltpunkte kann eine Umrechnungs-Funktion zugeordnet werden, die z.B. eine Invertierung ermöglichen.

Für Mess-, Stell- und Zählpunkte kann eine Kennlinien zugeordnet werden. Deren Steilheit an der Stelle 1 wird als Korrekturfaktor auf den BACnet-Wert angewendet.

Die IS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	54	26	Adr_actual	BACnet-Adresse für "actual"
6	56	60	5	UR_Fkt	Nummer der Umrechnungs-

funktion des BACnet-Wertes auf den FND-Index :  
 0: Keine Umrechnung  
 1: Invertierung (1-stufig)

7	62	64	3	Priority	BACnet-Priorität für WRITE (Dieser Parameter kann optional angegeben werden) derzeit NUR * reserviert *
---	----	----	---	----------	---

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	54	26	Adr_nominal	BACnet-Adresse für "nominal"
6	56	60	5	UR_Fkt	Nummer der Umrechnungsfunktion des BACnet-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig)
7	62	64	3	Priority	BACnet-Priorität für WRITE (Dieser Parameter kann optional angegeben werden)
8	66	91	26	Adr_actual	BACnet-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	54	26	Adr_actual	BACnet-Adresse für "actual"
6	56	60	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der BACnet- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	62	64	3	Priority	BACnet-Priorität für WRITE (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	54	26	Adr_nominal	BACnet-Adresse für "nominal"
6	56	60	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der BACnet- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	62	64	3	Priority	BACnet-Priorität für WRITE (Dieser Parameter kann optional angegeben werden)

8	66	91	26	Adr_actual	BACnet-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)
---	----	----	----	------------	--

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	54	26	Adr_actual	BACnet-Adresse für "actual"
6	56	60	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der BACnet- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7	62	64	3	Priority	BACnet-Priorität für WRITE (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

**Beispiel für eine FND-Referenzdatei :**

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 000 045 Klartext für Meldepunkt      ADR_actual..... URFkt Pri
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 000 045 Klartext für Schalterpnt.  ADR_nominal..... URFkt Pri ADR_actual.....
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 000 108 Klartext für Messpunkt  ADR_actual..... KL_Id Pri
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 000 108 Klartext für Stellpunkt  ADR_nominal..... KL_Id Pri ADR_actual.....
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 000 108 Klartext für Zaehlpunkt  ADR_actual..... KL_Id Pri
...
```

**Initialisierung der BACnet**

Erreichbar sind alle die BACnet-Devices, die zum gleichen IP-Netzwerk wie der GA-Knoten gehören bzw. über entsprechende IP-Router erreichbar sind.

BACnet-Broadcasts sind an die BACnet-Device gerichtet, die mit dem GA-Knoten auf dem in der IZ-Zeile angegebenen Port kommunizieren. Diese BACnet-Devices können per "Dynamic Device Binding" erkannt und in der Routing-Tabelle geführt werden. Alle anderen BACnet-Devices müssen mittels der Device-Datei "iz04ddev.bn" im Unterverzeichnis "iz" manuell für die Routing-Tabelle parametrisiert werden. Diese hat folgendes Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	7	7	DDEV	BACnet Device-Id.

2	9	13	5	DNET	BACnet-Netzwerk
3	15	26	12	DADR	BACnet-Adresse (MAC)
4	28	48	21	IP-Adr./Port	IP-Adresse und -Port im Format "aaa.bbb.ccc.ddd#ppppp"

Das BACnet wird wie folgt initialisiert :

- Senden von "I-Am-Router-To-Network" für SNET aus der IZ-Zeile als Broadcast
- Senden von "Who-Is-Router-To-Network" als Broadcast
- Senden von "I-Am" als Broadcast mit :
  - max. APDU-Length 1476 Bytes
  - Segmentation: NO
- Senden von "Who-Is" als Broadcast
- Erwarten und Empfangen der "I-Am"-Datagramme mit Aufbau der Routing-Tabelle
- Senden von "SubscribeCOV" an alle parametrisierten BACnet-Objekte, sobald die Route für das BACnet-Device aufgrund eines empfangenen "I-Am" oder der Device-Datei in der Routing-Tabelle eingetragen wurde

Wartezeiten bei der Initialisierung und zyklisch unter der Laufzeit wird der Abgleich der Routing-Tabelle durch Senden von "Who-Is" als Broadcast erzwungen. Dabei empfangene Datagramme mit abweichender BACnet- oder IP-Adresse führen zur Aktualisierung der Routing-Tabelle und erneutem Senden von "SubscribeCOV" an die zugehörigen Objekte.

**Hinweis:** Momentan noch NICHT realisiert, da ein Ausfall des COV-Servers nicht erkannt wird (vgl. Probleme mit SAIA-Testsystem !!!)

Ist der Zeitabgleich aktiviert (ZA-Zeile), gibt der IZ-Treiber die neue Systemzeit (Datum/Uhrzeit) mittels des "TimeSynchronisation"-Service als Broadcast an die BACnet-Devices weiter.

Der IZ-Treiber ist - ungeachtet der BACnet-Interoperability-Building-Blocks (BIBB) - vorläufig begrenzt auf die BACnet-Services, die für die Umsetzung auf FND benötigt werden wie folgt :

- Initialisierung, wie oben beschrieben
- "ReadProperty" der für die Führungsgröße angegebenen Property
- "ReadProperty" der "Out-Of-Service"-Property (\*\*\* deaktiviert \*\*\*)
- "ReadProperty" der "Status-Flags"-Property
- "ReadProperty" der "Event-State"-Property (\*\*\* deaktiviert \*\*\*)



- "ReadProperty" der für "actual" angegebenen Property (nur Schalt- und Stellpunkte)
- "ReadProperty" der "Low-Limit"- und "High-Limit"-Property (nur Mess- und Zählpunkte)
- "WriteProperty" der für die Führungsgröße angegebenen Property (nur Schalt-, Stell- und Zählpunkte)
- "WriteProperty" der "Low-Limit"- und "High-Limit"-Property (nur Mess- und Zählpunkte)

"WriteProperty" werden mit der optional angegebenen Priority, sonst OHNE explizite Angabe einer Priority ausgeführt (Default-Wert gemäß BACnet-Spezifikation: Lowest-Priority = 16).

Der Ersteller der Anlage MUSS die damit geltenden Wert für die Voreinstellung anlagenintern umsetzen.

**Hinweis:** Der IZ-Treiber unterstützt KEINE Freigabe (Relinquish) mit der zugeordneten Priority. Der Ersteller der Anlage MUSS sicherstellen, dass die verwendete Priority bei Bedarf von anderer Stelle freigegeben wird.

Der IZ-Treiber stellt selbst nur das DEVICE-Object bereit, dessen Pflicht-Properties einzeln mittels "ReadProperty" gelesen werden können.

**Spontane Meldungen und Polling der Datenpunkte**

Das BACnet-Protokoll unterstützt zwar spontane Meldungen (Change-Of-Value (COV), AlarmEvents (AE)), diese sind jedoch nicht für alle BACnet-Geräte zwingend. Bei der Initialisierung wird geprüft und vermerkt, ob ein Objekt mittels COV spontan melden kann. Falls NICHT, wird das Objekt zyklisch gepollt. Falls JA ist darauf zu achten, dass das "COV\_Increment" geeignet parametrisiert wurde.

Für Alarm- und Event-Notifications MUSS der Ersteller der Anlage den IZ-Treiber (IP-Adr./Port bzw. BACnet-Device-Id.) in den betroffenen Objekten vom Typ "Notification Class" eines jeden betroffenen BACnet-Device eingetragen.

Für 1-stufige Melde- und Schaltpunkte wird zur Umrechnung des BACnet-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:	BACnet	FND
	0	<-> 1
	1	<-> 0

**Zugriffskontrolle**

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe gemäß FND-Spezifikation.

### **Grenzwert-Überwachung**

Das BACnet-Protokoll unterstützt zwar die Grenzwert-Überwachung, diese ist jedoch nicht für alle BACnet-Geräte zwingend. Bei der Initialisierung wird geprüft und vermerkt, ob für ein Objekt "LOW\_Limit" und "HIGH\_Limit" für die Alarm-Grenzen definiert sind. Falls NICHT, wird die Grenzwert-Überwachung für die Alarm- als auch alle Warn-Grenzen im IZ-Treiber emuliert.

**Hinweis:** Momentan noch NICHT realisiert, derzeit stets Zugriff auf die Limit-Werte, wobei BACnet-Error ignoriert werden

### **Ausblendung der Betriebsmeldungen**

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

### **Status-Ausgabe**

Soweit keine vorangigen Aufgaben anstehen, erzeugt FND-SSA/LIB die Datei IZnnSTAT.BN im aktuellen Arbeitsverzeichnis ("nn" Treiber-Id. hexadezimal kodiert). Darin werden folgende Status-Informationen ausgegeben:

- Status und ggf. Zeitpunkt der letzten Änderung für alle in der Routing-Tabelle geführten BACnet-Devices kodiert wie folgt:
  - 0: KEIN Fehler
  - 1: Ausfall
  - 2: Wiederkehr

### **BACnet-to-Referenz-Datei (BN2REF)**

Das Hilfs-Programm BN2REF (bn2ref.exe) dient zum automatischen Auslesen der BACnet-Konfiguration aller Devices in allen bzw. einem IP-Netz mit der angegebenen IP-Adr. und/oder Portnummer. Erzeugt wird die zugehörigen FND-Referenz-Datei "bn2ref.ref" wie folgt :

- Senden von "Who-Is" mit "Portnummer" an IP-Netz bzw. alle am Rechner eingerichteten IP-Netze
- Vermerken aller mit "I-Am" antwortenden BACnet-Devices

- Lesen der binären, analogen und MultiState-BACnet-Objekte für jedes Device sowie des zugehörigen "Object\_Identifier", "Object\_Name",

"Description" usw. sowie Generierung der zugehörigen Zeile in der FND-Referenzdatei.

Aus "Description" werden die ersten 23 Zeichen als Datenpunkt-Text übernommen. Folgt darauf eine Zahl, eingebettet in rechteckige Klammern ("[nnn]"), so wird die Zahl als FND-Kodierung für das binäre Attribut ("atb\_id" für Melde- und Schalterpunkte) bzw. die physikalische Einheit ("dimension" für Mess-, Stell- und Zählpunkte) übernommen.

Ist der Schalter "-e" mit der die Zeichenfolge "EndePrefix" angegeben, so wird hinter das letzte Vorkommen in "Object\_Name" positioniert.

Ist der Schalter -b mit der Zeichenfolge "BeginnDpId" angegeben, so wird nachfolgend auf den Anfang ihres erstes Vorkommens positioniert. Erst dann wird die verbleibende Zeichenfolge als Datenpunkt-Adresse übernommen.

Ist der Schalter -a mit der Zeichenfolge "AnfügeDpId" angegeben, so wird diese an die Datenpunkt-Adresse angefügt (z.B. Firmenbezeichnung/Hersteller-Kode bei der Universität Stuttgart).

- Ist die "atb\_id" NICHT als Bestandteil der "Description" angegeben, so wird diese durch Lesen von "Inactive/Active\_Text" bzw. "State\_Text" und Vergleich mit den Einträgen in der Datei cfg/dp\_atb.txt ermittelt.

Aufruf : BN2REF [ [IP-Netz#]Portnummer ] [ -aAnfügeDpId ] [ -bBeginnDpId ] [ -eEndePrefix ] [ Schalter für Test-Funktionen ]

#### 5.4.8.17. AS511

Grundlage für die Realisierung sind :

- "AS511 Protocol Notes", Stand: 17.03.2004, Autor: Luca Gallina
- Software FENDELOP Version 1.4, Stand: 14.07.2005, Autor: NFT Umweltsysteme GmbH
- Angebot AWMMIHDE/6021 vom 07.02.2006

**Hinweis:** Der Funktions-Umfang dieses Treibers bleibt vorerst beschränkt auf die Anforderungen als Ersatz für die o.g. Software im Projekt "Sickerwasser-Aufbereitung" (SiWa) im Entsorgungspark Freimann des Abfallwirtschaftsbetrieb München (AWM-ESP) gemäß dem o.g. Angebot, d.h. insbesondere :

- nur lesende Zugriffe
- nur Melde- und Messpunkte

- KEINE Invertierung
- KEINE optionalen AS511-Adressen

Folgende Abweichungen wurden festgelegt :

- Nachfolgende Angaben zu Schalt-, Stell- und Zählpunkten als auch modifizierende Zugriffe sind Reservierungen für zukünftige Erweiterung und stellen kein realisiertes Leistungsmerkmal dar.
- Dieser Treiber kommt vorläufig nur als Spezial-Software in Verbindung mit dem Treiber für die Leitzentrale (LZ) auf Basis des LON (Fa. Echelon) mittels des FND über LonTalk-Protokoll zum Einsatz (vgl. Unterverzeichnis LZ\LON und GAK\SSA\_S5\_L.EXE)

**V.24-Schnittstelle zum Anschluss eines AS511-Systems**

Das AS511-System wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, EVEN-Parity-Bit und 1 Stopbit

**Spezifische Anpassungen in der Konfigurationsdatei**

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 118 *** sonst : 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber

					angeschlossenen DDC-IZ beginnend mit 0.
					*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !
					*** Bis dahin gilt : *** AS511: stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der DDC-IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "Leistungsverzeichnis eines GA-Knotens" vom 29.8.94 (vgl. Seite 20).
					*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !
					*** Bis dahin gilt : *** AS511: 0
6	40	42	3	Protokoll	nat. Zahl kodiert wie folgt :
					*** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !
					*** Bis dahin gilt : *** AS511: 0
7	44	256	<=213	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem DDC/IZ-Typ.
					"COM1" - "COM4" (ohne " bzw. "/dev/ttyS0" ff.
					Wird durch das Zeilenabschlusszeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Die AS511-Adressen werden in den IS-Zeilen wie folgt kodiert:

"**tddd.www[,bb]**" mit :

**t** ..: Typ

**ddd** : Datenbaustein (DB, 0 <= ddd <= 255 )

**www** : Datenwort (DW, 0 <= ddd <= 255 )

**bb** .: Bit (optional, 0 (LSB) <= bb <= 15 (MSB) )

Dabei sind "ddd", "www" und "bb" dezimale Zahlen >= 0 und "t" der erste Buchstabe der nachfolgend aufgelisteten Typen (z.B. "D" für Datenbaustein) :

- Datenbaustein

Für Melde- und Schaltpunkte kann optional eine Umrechnungsfunktion für die Invertierung der binären Werte angegeben werden.

Für Mess-, Stell- und Zählpunkte muss eine Kennlinien zugeordnet werden.

Für Mess-, Stell- und Zählpunkte kann optional eine AS511-Adresse für die Geber-Störung und die Grenzwerte angegeben werden.

**Hinweis:** Alle AS511-Adressen eines Datenpunktes MÜSSEN im selben Datenbaustein realisiert sein !!!

Die IS-Zeilen werden durch das Zeilenabschlusszeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	39	11	Adr_actual	AS511-Adresse für "actual"
6	41	45	5	UR_Fkt	Nummer der Umrechnungsfunktion des AS511-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	39	11	Adr_nominal	AS511-Adresse für "nominal"
6	41	45	5	UR_Fkt	Nummer der Umrechnungsfunktion des AS511-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (Dieser Parameter kann optional angegeben werden)
7	47	57	11	Adr_actual	AS511-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	39	11	Adr_actual	AS511-Adresse für "actual"
6	41	45	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der AS511- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur)  Hinweis: Momentan wird nur ----- das INTEGER-Format unterstützt, so dass eine Kennlinie zugeordnet werden muss !
7	47	57	11	Adr_fl	AS511-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	59	69	11	Adr_alarm_low	AS511-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	71	81	11	Adr_alarm_high	AS511-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
10	83	93	11	Adr_warning_low	AS511-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
11	95	105	11	Adr_warning_high	AS511-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	39	11	Adr_actual	AS511-Adresse für "actual"
6	41	45	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung der AS511- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur)  Hinweis: Momentan wird nur ----- das INTEGER-Format unterstützt, so dass eine Kennlinie zugeordnet werden muss !
7	47	57	11	Adr_fl	AS511-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	59	69	11	Adr_actual	AS511-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

5	29	39	11	Adr_actual	AS511-Adresse für "actual"
6	41	45	5	KL_Id	<p>Nummer der Kennlinie, die zur Umrechnung der AS511- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur)</p> <p>Hinweis: Momentan wird nur ----- das INTEGER-Format unterstützt, so dass eine Kennlinie zugeordnet werden muss !</p>
7	47	57	11	Adr_fl	AS511-Adresse des Alarms für die Geber-Störung. (Dieser Parameter kann optional angegeben werden)
8	59	69	11	Adr_alarm_high	AS511-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)
9	71	81	11	Adr_warning_high	AS511-Adresse des Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein (d.h. Addieren von 53 !).

**Beispiel für eine FND-Referenzdatei :**

```

00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 015 045 Klartext für Meldepunkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 015 045 Klartext für Schalterpkt.
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 015 108 Klartext für Messpunkt
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 015 108 Klartext für Stellpunkt
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 015 108 Klartext für Zaehlpunkt
...
ADR_actual_URFkt
ADR_nominal_URFkt ADR_actual_
ADR_actual_KL_id ADR_fl_____ ADR_al_____ ADR_ah_____ ADR_wl_____ ADR_wh_____
ADR_nominal_KL_id ADR_fl_____ ADR_actual_
ADR_actual_KL_id ADR_fl_____ ADR_al_____ ADR_wh_____
    
```

**Initialisierung der AS511**

Nach dem Programmstart und jedem Kommunikationsfehler wird die "Initial Adress" und "Block Length" eines jeden Datenbausteins ausgelesen (Function B\_INFO) und vermerkt.

**Polling der Datenpunkte**

Die AS511 kann NICHT spontan melden, so dass auch Melde- und Schalterpunkte zyklisch abgefragt werden müssen (Polling), um Änderungen im "actual" zu erkennen.

Analog müssen auch die Mess-, Stell- und Zählerpunkte zyklisch abgefragt werden, um Geberstörungen und das Über-/Unterschreiten von Grenzwerten zu erkennen.



Nachdem alle Melde- und Schaltpunkte abgefragt wurden wird der nächste Mess-, Stell- bzw. Zählpunkt abgefragt, bevor wieder mit dem ersten Melde-bzw. Schaltpunkt begonnen wird.

### **Zugriffskontrolle**

Der IZ-Treiber prüft vor jedem WRITE auf die AS511 die Modifikations-Berechtigung.

Die WRITE-Zugriffe auf die einzelnen Datenpunkte durch die AS511 unterliegen keiner Kontrolle, d.h. die durch FND gesetzte Sperrpriorität wird nicht unterstützt.

### **Grenzwert-Überwachung**

Für Mess- und Zählpunkte können optional AS511-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die AS511-Adressen werden zyklisch gepollt und geprüft (vgl. oben).

Eine Behandlung der vom AS511 erzeugten Grenzwert-Alarmer erfolgt nicht. Bei Bedarf können diese jedoch gesondert als Meldepunkte definiert werden.

### **Ausblendung der Betriebsmeldungen**

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

#### **5.4.8.18. ProConOS (PCOS; Fa. PHOENIX Contact)**

Der Treiber für die Inselzentrale (IZ) auf Basis von Phoenix Contact-Steuerungen INLINE CONTROL (ILC) bzw. FC-Steckkarte wird als Routinen-Bibliothek (FND-SSA/LIB) zum Einbinden in den GA-Knoten realisiert.

Grundlage für die Realisierung sind, soweit bereitgestellt:

- Zugriff auf Variablen, V 1.0 vom 08.01.2008 der KW-Software GmbH (Dok.: AW-PxC-080103-01)
- ProConOS online communication services, 4.1. Developer's Manual
- Format for CSV File of SCG, V 1.0.2 vom 26.05.1997 der KW-Software GmbH (Dok.: AW-MWT-9611-002)

- eMail Hr. Weichelt vom 31.01.2008
- Angebot PCEBPFK/8011 vom 26.01.2008
- Bereitgestellte Testkonfiguration bestehend aus einem ILC 350 ETH sowie eine Steckkarte FC 350 PCI ETH dienen als Referenzsystem (IZ gemäß FND-Spezifikation).

Folgende Erweiterungen bzw. Abweichungen wurden festgelegt :

\* KEINE \*

Unterstützt wird die Status- und Ereignis-Behandlung, explizite Rückmeldungen/Istwerte, Grenzwerte sowie das (Rück-) Setzen von Zählpunkten für die Verbrauchsmessung, soweit vom PCOS-Gerätes unterstützt.

Schalt- und Stellpunkten kann eine zusätzliche Steuerungs-Variable als Rückmeldung bzw. expliziter Istwert zugeordnet werden. Dadurch ist es möglich, funktionell zusammengehörige Steuerungs-Variable auf einen Datenpunkt abzubilden.

Während des Programmlaufs bildet der FND-SSA/LIB automatisch die verfügbaren Status- und Ereignis-Informationen der Steuerungs-Variable auf die entsprechenden Informationen des Datenpunktes ab und löst ggf. die zugeordnete Spontanmeldung aus.

Der GA-Knoten gleicht seine Systemzeit bei der Kommunikation mit der Leitzentrale oder einem Zeit-Server ab.

### **LAN-Schnittstelle zum Anschluss der IZ**

Die ILC-/FC-Steuerungen werden über ein lokales TCP/IP-Netzwerk unter LINUX angeschlossen. Der Datenaustausch mit der FND-SSA/LIB erfolgt auf Grundlage der "ProConOS online communication services" (PCOS). Das Betriebssystem DOS wird NICHT unterstützt.

**Hinweis :** Bei der Einrichtung des Rechners ist darauf zu achten, dass dieser dem bzw. den Netzwerken zugeordnet ist, über die die PCOS-Geräte erreichbar sind (vgl. "hostname -i"). Ggf. sind in der Firewall die verwendeten Ports (z.B. TCP-Port 20547) freizuschalten.

### **Spezifische Anpassungen in der Konfigurationsdatei**

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 120 *** sonst .: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen DDC-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** PCOS: stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** PCOS: 41 (TCP/IP)
6	40	42	3	Protokoll	Verbindungs-Protokoll als Zahl kodiert wie folgt : 0: PCOS  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** PCOS: 0
7	44	256	<=190	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  Für TCP/IP Port im Format "[IP-Netz#]Port" mit IP-Netz aaa.bbb.ccc.ddd (IP4-Adr.) und "Port" gemäß Projekt-Vorgaben, i.a. 20547  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>), je nach verwendetem Editor abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

**Hinweis:** IP-Adresse und -Port müssen mit dem System-Verantwortlichen abgestimmt sein, um Konflikte mit bestehenden und zukünftigen Netzwerk-Geräten zu vermeiden !!!

Die PCOS-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

**"ddd:vvv...vvv"** mit :

- ddd:** lfd.Nr. des PCOS-Gerätes (vgl. "izNNconn.vbp" für Zuordnung von IP-Adr./-Port)
- vvv...vvv:** eindeutiger symbolischer Name der Variablen innerhalb des PCOS-Gerätes (vgl. rechtsbündiger Teil in Spalte 1 der SR-Datei, max. 30 Zeichen ohne Angabe der Resource, d.h. hinter dem letzten Punkt). Optional kann die Angabe inkl. der vorangestellten und durch '.' getrennten Resource erfolgen, max. 55 Zeichen.

Soweit einem FND-Datenpunkt weitere PCOS-Adressen zugeordnet werden, so müssen diese auf dem gleichen PCOS-Gerät realisiert sein.

Der PCOS-Datentyp der Variable sowie die technische Adresse innerhalb des zugeordneten PCOS-Gerätes wird zur Laufzeit aus der SR-Datei ermittelt (vgl. Spalte 3 der mittels der Services SYS\_INIT\_SEND, SYS\_SEND\_DATA und SYS\_END\_SEND gelesenen Datei "sr.zsv").

Für Melde- und Schaltpunkte kann eine Umrechnungs-Funktion zugeordnet werden, die z.B. eine Invertierung ermöglichen.

Für Mess-, Stell- und Zählpunkte kann eine Kennlinien zugeordnet werden. Für den PCOS-Datentyp REAL wird deren Steilheit an der Stelle 1 als Korrekturfaktor auf den PCOS-Wert angewendet.

Die IS-Zeilen werden ab Zeichen 29 werden im CSV-Format angegeben und durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen. Dabei werden die einzelnen Parameter durch ein ';' voneinander getrennt. Optionale Parameter müssen nicht mit Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels ';' zu kennzeichnen. Das ';' hinter dem letzten Parameter kann entfallen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	','		Adr_actual	PCOS-Adresse für "actual"
6		','		UR_Fkt	Nummer der Umrechnungs-

funktion des PCOS-Wertes auf den FND-Index :  
 0: Keine Umrechnung  
 1: Invertierung (1-stufig)  
 2: Invertierung (2-stufig)  
 (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	';		Adr_nominal	PCOS-Adresse für "nominal"
6		';		UR_Fkt	Nummer der Umrechnungsfunktion des PCOS-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) (Dieser Parameter kann optional angegeben werden)
7		';		Adr_actual	PCOS-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	';		Adr_actual	PCOS-Adresse für "actual"
6		';		KL_Id	Nummer der Kennlinie, die zur Umrechnung der PCOS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7		';		Adr_alarm_low	PCOS-Adresse des unteren Alarm-Grenzwertes (Dieser Parameter kann optional angegeben werden)
8		';		Adr_alarm_high	PCOS-Adresse des oberen Alarm-Grenzwertes (Dieser Parameter kann optional angegeben werden)
9		';		Adr_warning_low	PCOS-Adresse des unteren Warnungs-Grenzwertes (Dieser Parameter kann optional angegeben werden)
10		';		Adr_warning_high	PCOS-Adresse des oberen Warnungs-Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	';		Adr_nominal	PCOS-Adresse für "nominal"

6	';'	KL_Id	Nummer der Kennlinie, die zur Umrechnung der PCOS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
8	';'	Adr_actual	PCOS-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	';		Adr_actual	PCOS-Adresse für "actual"
6		';		KL_Id	Nummer der Kennlinie, die zur Umrechnung der PCOS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7		';		Adr_alarm_high	PCOS-Adresse des oberen Alarm-Grenzwertes (Dieser Parameter kann optional angegeben werden)
8		';		Adr_warning_high	PCOS-Adresse des oberen Warnungs-Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

**Beispiel für eine FND-Referenzdatei :**

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 000 045 Klartext für Meldepunkt           ADR_actual;URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 000 045 Klartext für Schalterpkt.       ADR_nominal;URFkt;ADR_actual
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 000 108 Klartext für Messpunkt           ADR_actual;KL_Id
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 000 108 Klartext für Stellpunkt       ADR_nominal;KL_Id;ADR_actual
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 000 108 Klartext für Zaehlpunkt           ADR_actual;KL_Id
...
```

**Initialisierung der PCOS-Geräte**

Erreichbar sind alle die PCOS-Steuerungen, die zum gleichen IP-Netzwerk wie der GA-Knoten gehören bzw. über entsprechende IP-Router erreichbar sind.

Voraussetzung für die Initialisierung ist das Programm UNZIP. Dieses muss lauffähig installiert und über den Pfad (vgl. PATH) aufrufbar sein.

Alle PCOS-Geräte müssen mittels der Datei "izNNconn.vbp" im Unterverzeichnis "iz" für die IP-Adressierung parametrieren werden (NN: Treiber-Id. (hexadezimal). Diese Datei hat das gleiche Format wie IZ03CONN.VBP", vgl. System-Spezifikation des GA-Knoten unter Abschnitt 5.4.6.3. Als Verbindungs-Art wird nur zugelassen :

41: TCP/IP über Netzwerk

Die Angabe der Verbindungs-Adresse erfolgt im Format :

"aaa.bbb.ccc.ddd#ppppp": IP-Adresse und -Port (ohne ")

Der Datenaustausch mit den PCOS-Geräten wird gemäß "Zugriff auf Variablen" wie folgt initialisiert :

- Aufbau der TCP/IP-Verbindung und Einloggen (Phase 1)
- OPC-Konfiguration hochladen (Phase 2, liefert die o.g. SR-Datei)

Die Initialisierung für ein PCOS-Gerät wird wiederholt, nachdem die Verbindung beendet wurde (vgl. Logout, Phase 5) oder eine Störung der Kommunikation behoben wurde.

Das Hochladen der OPC-Konfiguration wird wiederholt, nachdem der inkrementierte PDD-Counter die Änderung des Programmes auf dem PCOS-Gerät anzeigt.

**Spontane Meldungen und Polling der PCOS-Variablen**

Das PCOS-Protokoll unterstützt keine spontanen Meldungen. Daher werden die Werte aller parametrieren Variablen zyklisch gelesen (Polling im Master-Slave-Verfahren).

Für 1- und 2-stufige Melde- und Schaltpunkte wird zur Umrechnung des PCOS-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:	PCOS	FND
	0 <->	1
	1 <->	0
2-stufige Invertierung:	PCOS	FND
	0 <->	2
	1 <->	1

2 <-> 0

**Zugriffskontrolle**

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe gemäß FND-Spezifikation.

**Grenzwert-Überwachung**

Für Mess- und Zählpunkte können optional PCOS-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die PCOS-Adressen werden zyklisch gemäß dem in der ES-Zeile definierten Abfrageintervall gepollt und geprüft.

Soweit in der PCOS-Adresse eine lfd. Nummer für das PCOS-Gerät angegeben wird, so muss diese auf dem gleichen PCOS-Gerät realisiert sein wie die Führungsgröße.

**Ausblendung der Betriebsmeldungen**

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

**SR-Datei-to-Referenz-Datei (SR2REF)**

Das Hilfs-Programm SR2REF (sr2ref.exe) dient zum automatischen Auslesen der SR-Dateien (sr.zsv) aller in "iz/izNNconn.vbp" parametrisierten PCOS-Geräte mit der dort angegebenen IP-Adr. und/oder Portnummer.

Aufruf : SR2REF Treiber-Id. (dezimal od. hexadezimal mit Prefix "0x")

[ -aAnfügeDpId ] (Voreinstellung: KEINE)

[ -bBeginnDpId ] (Voreinstellung: "DPT\_aaa\_dp-id")

[ -eEndePrefix ] (Voreinstellung: ".")

[ Schalter für Test-Funktionen ]

Ist der Schalter "-e" mit der die Zeichenfolge "EndePrefix" angegeben, so wird hinter das letzte Vorkommen in Spalte 1 positioniert. Bei Verwendung der Voreinstellung wird somit hinter die Resource positioniert.



Ist "EndePrefix" leer, d.h. nur der Schalter -e angegeben, so werden alle Ressourcen unverändert als führender Bestandteil der Datenpunkt-Adresse übernommen.

Ist der Schalter -b mit der Zeichenfolge "BeginnDpId" angegeben, so wird nachfolgend auf den Anfang ihres erstes Vorkommens in Spalte 1 positioniert. Erst dann wird die verbleibende Zeichenfolge als Datenpunkt-Adresse übernommen. Bei Verwendung der Voreinstellung wird ein Aufbau des Variablen-Namens im folgenden Format "**DPt\_aaa\_dp-id**" erwartet wobei gilt ('\_' Trennzeichen) :

**DP:** feste Zeichenfolge als Prefix für Datenpunkt

**t:** Datenpunkt-Typ (dp\_type) gemäß FND-Spezifikation  
1: Meldepunkt  
2: Schalterpunkt  
3: Messpunkt  
4: Stellpunkt  
5: Zählpunkt

**aaa:** max. 3-stellige Nummer gemäß Anhang E der FND-Spezifikation zur Kodierung der Attribute (atb\_id) bei Melde- und Schalterpunkten bzw. der physikalischen Einheit (dimension) bei Mess-, Stell- und Zählpunkten

**dp-id:** Datenpunkt-Adresse (dp\_id) gemäß FND-Spezifikation als max. 16-stellige alphanumerische Zeichenfolge, i.a. unter Berücksichtigung der projektspezifischen Vorgaben (z.B. Adress-Schlüssel gemäß allg. GA-Richtlinien des Auftraggebers)

**Beispiel 1:** "DP1\_027\_ZGAAMMM010000II" erzeugt Eintrag für den Meldepunkt mit der Datenpunkt-Adresse ZGAAMMM010000II und dem Attribut NORMAL/STÖRUNG (vgl. atb\_id 27)

**Beispiel 2:** "DP3\_108\_ZGAAMMM030000II" erzeugt Eintrag für den Messpunkt mit der Datenpunkt-Adresse ZGAAMMM030000II und der phys. Einheit "°C" (Grad Celsius, vgl. dimension 108)

Ist "BeginnDpId" leer, d.h. nur der Schalter -b angegeben, so werden alle Variablen-Namen unverändert als Datenpunkt-Adresse übernommen. In Kombination mit leerem "EndePrefix" wird also die gesamte Zeichenfolge in Spalte 1 (inkl. Resource) als Datenpunkt-Adresse übernommen.

Ist der Schalter -a mit der Zeichenfolge "AnfügeDpId" angegeben, so wird diese an die Datenpunkt-Adresse angefügt (z.B. Firmenbezeichnung/Hersteller-Kode bei der Universität Stuttgart).

Erzeugt wird die zugehörige Vorlage für die Referenz-Datei "sr2ref.ref" durch Lesen der SR-Datei jedes einzelnen PCOS-Gerätes wie folgt :

- Aufbau der TCP/IP-Verbindung und Einloggen
- OPC-Konfiguration hochladen, speichern und ggf. dekomprimieren als SR-Datei

- Lesen der einzelnen Zeilen aus der SR-Datei und Prüfung des Variablen-Namens - OHNE Resource (Spalte 1, hinter dem letzten Punkt) auf Übereinstimmung mit der Zeichenfolge "BeginnDpId"
- Bei Übereinstimmung prüfen, ob der Variablen-Name mit "\_act", "\_al", "\_wl", "\_wh" oder "\_ah" endet. Falls JA, prüfen, ob Variablen-Name OHNE diesen Postfix für eine Führungsgröße besteht. Falls JA, deren Eintrag in der Referenz-Datei um die optionale PCOS-Adresse ergänzen für :
  - "actual" ("\_act", nur Schalt- und Stellpunkte)
  - "alarm\_low" bzw. "warning\_low" ("\_al" bzw. "\_wl", nur Messpunkte)
  - "warning\_high" bzw. "alarm\_high" ("\_wh" bzw. "\_ah", nur Mess- und Zählpunkte)

**Beispiel 3:** "DP3\_108\_ZGAAMMM03OOOOII\_ah" ergänzt Eintrag für den Messpunkt mit der Datenpunkt-Adresse ZGAAMMM03OOOOII und der phys. Einheit "°C" (Grad Celsius, vgl. oben) um die optionale PCOS-Adresse für den oberen Alarm-Grenzwert ("alarm\_high")

- Bei Übereinstimmung OHNE o.g. Postfix und Führungsgröße wird ein neuer Eintrag in der Referenzdatei angefügt. Dabei werden Datenpunkt-Adresse, ggf. Datenpunkt-Typ und die Kodierung für Attribut bzw. phys. Einheit aus dem Variablen-Namen übernommen. Der Klartext zur Beschreibung des Datenpunktes wird zusammengesetzt aus Speicherort und Größe (max. 23 Zeichen, vgl. Spalte 2 und 3)
- Anschließend werden alle Dateien "iz/izNN\*.csv" dahingehend geprüft, ob diese mittels EXCEL-Anbindung aus PC WorX exportiert wurden. Hierzu muss die erste Zeile mit der Zeichenfolge "Type;Name;Datatype;Comment" beginnen. Ggf. werden alle nachfolgenden Zeilen im CSV-Format mit Trennzeichen ';' dahingehend geprüft, ob in der 1. Spalte ("Type") der Buchstabe 'V' angegeben und in der 2. Spalte ("Name") der aktuell bearbeitete Variablen-Name. In diesem Fall werden max. die führenden 23 Zeichen der 4. Spalte ("Comment") als Klartext zur Beschreibung des Datenpunktes übernommen.

#### **5.4.8.19. MODBUS/IP und MODBUS/RTU**

Der Treiber für die Inselzentrale (IZ) auf Basis MODBUS-Protokoll über TCP/IP oder serielle Schnittstelle (RTU) wird als Routinen-Bibliothek (FND-SSA/LIB) zum Einbinden in den GA-Knoten realisiert.

Grundlage für die Realisierung sind, soweit bereitgestellt (vgl. unter [www.Modbus.org](http://www.Modbus.org) und [www.Modbus-IDA.org](http://www.Modbus-IDA.org)) :

- MODBUS Application Protocol Specification V1.1.a (Stand: 04.06.2004)
- MODBUS Messaging on TCP/IP Implementation Guide V1.0a (Stand: 04.06.2004)
- MODBUS over Serial Line Specification & Implementation Guide V1.0 (Stand: 12.02.2002)

- Angebot LHMMDR/8031 vom 01.03.2008
- Bereitgestellte Testkonfiguration bestehend aus einem WAGO 750-841 (TCP/IP, UDP/IP) sowie 750-814 (RTU) dienen als Referenzsystem (IZ gemäß FND-Spezifikation).
- WAGO 750-841 Handbuch (m084100d.pdf unter www.wago.com)
- Programmier-System CoDeSys (Version: 2.3.8.5)

Folgende Erweiterungen bzw. Abweichungen wurden festgelegt :

\* KEINE \*

Unterstützt wird die Status- und Ereignis-Behandlung, explizite Rückmeldungen/Istwerte, Grenzwerte sowie das (Rück-) Setzen von Zählpunkten für die Verbrauchsmessung, soweit vom MODBUS-Gerät unterstützt.

Schalt- und Stellpunkten kann eine zusätzliche Steuerungs-Variable als Rückmeldung bzw. expliziter Istwert zugeordnet werden. Dadurch ist es möglich, funktionell zusammengehörige Steuerungs-Variable auf einen Datenpunkt abzubilden.

Während des Programmablaufs bildet der FND-SSA/LIB automatisch die verfügbaren Status- und Ereignis-Informationen der Steuerungs-Variable auf die entsprechenden Informationen des Datenpunktes ab und löst ggf. die zugeordnete Spontanmeldung aus.

Der GA-Knoten gleicht seine Systemzeit bei der Kommunikation mit der Leitzentrale oder einem Zeit-Server ab. Ein Zeitabgleich zum MODBUS erfolgt jedoch nicht.

### **LAN-Schnittstelle zum Anschluss der IZ**

Die MODBUS-Geräte werden über ein lokales TCP/IP-Netzwerk oder serielle Schnittstelle mit Standard-Treiber unter LINUX angeschlossen.

Der Datenaustausch mit der FND-SSA/LIB erfolgt auf Grundlage der o.g. MODBUS-Spezifikationen. Das Betriebssystem DOS wird NICHT unterstützt.

**Hinweis** : Bei der Einrichtung des Rechners ist darauf zu achten, dass dieser dem bzw. den Netzwerken zugeordnet ist, über die die MODBUS-Geräte erreichbar sind (vgl. "hostname -i"). Ggf. sind in der Firewall die verwendeten Ports (z.B. TCP-Port 502) freizuschalten.

**V.24-Schnittstelle zum Anschluss der IZ (MODBUS/RTU)**

Die IZ wird über eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen. Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle (i.a. /dev/ttyS..) werden diese Funktionen über Puffer und den Standard-Treiber realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

19200 Baud, 1 Startbit, 8 Datenbit, EVEN-Parity-Bit und 1 Stopbit

**Spezifische Anpassungen in der Konfigurationsdatei**

IZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"IZ"	Zeilen-Kennung (OHNE '"')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben. 0: reserviert für zukünftige Erweiterungen  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** GAK_XXL: 40 *** sonst .: 4
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossen DDC-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** MODBUS: stets 0
4	12	34	23	IZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** MODBUS: 0 (RTU) 41 (TCP/IP)

6	40	42	3	Protokoll	Verbindungs-Protokoll als Zahl kodiert wie folgt : 0: MODBUS  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** MODBUS: 0
7	44	256	<=190	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.  Für RTU Name der seriellen Schnittstelle mit Standard- Treiber (i.a. /dev/ttyS...)  Für TCP/IP Port im Format "[IP-Netz#]Port" mit IP-Netz aaa.bbb.ccc.ddd (IP4-Adr.) und "Port" gemäß Projekt-Vorgaben, i.a. 502  Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

**Hinweis:** IP-Adresse und -Port müssen mit dem System-Verantwortlichen abgestimmt sein, um Konflikte mit bestehenden und zukünftigen Netzwerk-Geräten zu vermeiden !!!

Die MODBUS-Adressen für die Führungsgrößen ("nominal" für Schalt- und Stellpunkte, sonst "actual") werden in den IS-Zeilen wie folgt kodiert:

"ddd[.rrr]:t[k]aaaaa[.ff]" mit :

- ddd** : Unit Identifier (Slave-Id.) des MODBUS-Gerätes (für Verbindungs-Art TCP/IP laufende Nr. innerhalb "izNNconn.vbp" für Zuordnung von IP-Adr./-Port, sonst stets 0)
- rrr** : Unit Identifier (Slave-Id.) des remote MODBUS-Gerätes, z.B. falls Server-Gateway (TCP/IP ==> RTU; für RTU muss die Unit-Id. angegeben werden)
- t** : Typ der MODBUS-Variablen kodiert wie folgt :
  - C Coil
  - I Input (discrete)
  - R Register (Holding)
  - S Sensor (Input Register)
- k** : Kodierung der Start-Adresse kodiert wie folgt - keine Angabe = dezimal, sonst :
  - x hexadezimal
- aaaaa** : Start-Adresse
- ff** : Format der MODBUS-Variablen kodiert wie folgt :

Für Coil und Input (n=1,...,8) :

- Bn 1 aus n Bit analog "actual" für Meldepunkte gemäß FND-Spezifikation,  
sonst: UNDEF (Voreinstellung: B1)
- Xn Index für Meldepunkte gemäß FND-Spezifikation (Zahl 0,...,8 kodiert in n Bit)

Für Register und Sensor :

- Bn und Xn (2 Byte; sonst wie Coil/Input)
- I2 integer (2 Byte; Voreinstellung)
- I4 long (4 Byte)
- U2 unsigned (2 Byte)
- U4 unsigned long (4 Byte)
- R4 float / REAL (4 Byte)
  
- i4 swapped long (4 Byte)
- u4 swapped unsigned long (4 Byte)
- r4 swapped float / REAL (4 Byte)

**Hinweis:** Gemäß MODBUS-Spezifikation entspricht die Reihenfolge der Bytes "Big-Endian". Für I2 und I4 wird das MSB als Vorzeichen interpretiert. R4 entspricht IEEE-754. Bei 4 Bytes befinden sich die MSB im zweiten Register.

Soll die Reihenfolge der Worte abweichend davon MSB im ersten Register erfolgen, so sind die Swapped-Formate explizit anzugeben

Es wird empfohlen, Mess-, Stell- und Zählpunkte bereits innerhalb des MODBUS-Gerätes auf die jeweilige phys. Einheit gemäß FND-Dimension umzurechnen und ggf. als REAL-Wert bereitzustellen (s. Beispiel unten).

Abhängig vom angeschlossenen MODBUS-Gerät muss die spezifische Abbildung (Mapping) der internen Adressen (z.B. gemäß IEC-61131) auf die MODBUS-Adressen beachtet werden.

Für den WAGO 750-841 sind diese z.B. beschrieben im zugehörigen Handbuch (vgl. m084100d.pdf unter [www.wago.com](http://www.wago.com)) im Abschnitt 4.2.4. Hierzu einige Beispiele für das MODBUS-Gerät 001 :

- Der erste digitale Eingang (DI) liegt auf Coil 0 (=> 001:Cx0000.B1)
- Sollen einzelne Bits innerhalb der Merkerworte %MW0 ff. adressiert werden so sind folgende Angaben zu machen :

- Bit 0 in %MW0 001:Cx3000.B1
- Bit 1 in %MW0 001:Cx3001.B1
- ...
- Bit 15 in %MW0 001:Cx300F.B1
- Bit 0 in %MW1 001:Cx3010.B1
- ...
- Bit 15 in %MW1 001:Cx301F.B1
- ...
- Bit 0 in %MW1279 001:Cx4FF0.B1
- ...
- Bit 15 in %MW1279 001:Cx4FFF.B1

- Der Vorgabewert für eine Tag/Nacht-Umschaltung (1-stufig) soll auf die IEC-Adresse %MW1 geschrieben werden (=> 001:Rx3001.B1)
- Der erste analoge Eingang liefert den Rohwert auf Register 0 (=> 001:Rx0000.I2)
- Im weiteren Verlauf wird dieser Rohwert in den jeweiligen phys. Wert umgerechnet und mittels der Typkonvertierung als REAL (4-Byte) der IEC-Adresse %QW256 zugeordnet (=> Rx0100.R4; Rx101 für %QW257 implizit zugeordnet)
- Der Vorgabewert für ein Ventil (0...100%) soll auf die IEC-Adresse %MW4 geschrieben werden (=> 001:Rx3004.I2)
- Ein Stromzähler liefert die gezählten KWh als 4-Byte-Wert auf IEC-Adressen %QW260 (=> 001:Rx0104.U4; Rx0105 für %QW261 implizit zugeordnet)

Für die RTU-Klemme WAGO 750-650/003-000 muss ein gesonderter Baustein (z.B. TERMINAL\_MODBUS\_SLAVE\_RTU\_EN) in die Applikations-Software eingebunden werden. Dieser stellt ein Feld von Registern in Form von indizierten Variablen (typSLAVE\_DATEN) bereit, die programmtechnisch zugeordnet werden müssen.

Soweit einem FND-Datenpunkt weitere MODBUS-Adressen zugeordnet werden, so müssen diese auf dem gleichen MODBUS-Gerät realisiert sein.

Für Melde- und Schaltpunkte kann eine Umrechnungs-Funktion zugeordnet werden, die z.B. eine Invertierung ermöglichen.

Für Mess-, Stell- und Zählpunkte kann eine Kennlinien zugeordnet werden.

Für den MODBUS-Datentyp REAL wird deren Steilheit an der Stelle 1 als Korrekturfaktor auf den MODBUS-Wert angewendet.

Die IS-Zeilen werden ab Zeichen 29 werden im CSV-Format angegeben und durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen. Dabei werden die einzelnen Parameter durch ein ';' voneinander getrennt. Optionale Parameter müssen nicht mit Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels ';' zu kennzeichnen. Das ';' hinter dem letzten Parameter kann entfallen.

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Meldepunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	','		Adr_actual	MODBUS-Adresse für "actual"
6		','		UR_Fkt	Nummer der Umrechnungsfunktion des MODBUS-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Schaltpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	','		Adr_nominal	MODBUS-Adresse für "nominal"
6		','		UR_Fkt	Nummer der Umrechnungsfunktion des MODBUS-Wertes auf den FND-Index : 0: Keine Umrechnung 1: Invertierung (1-stufig) 2: Invertierung (2-stufig) (Dieser Parameter kann optional angegeben werden)
7		','		Adr_actual	MODBUS-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Messpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	','		Adr_actual	MODBUS-Adresse für "actual"
6		','		KL_Id	Nummer der Kennlinie, die zur Umrechnung der MODBUS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7		','		Adr_alarm_low	MODBUS-Adresse des unteren Alarm-Grenzwertes (Dieser Parameter kann optional angegeben werden)
8		','		Adr_alarm_high	MODBUS-Adresse des oberen Alarm-Grenzwertes (Dieser Parameter kann optional angegeben werden)
9		','		Adr_warning_low	MODBUS-Adresse des unteren Warnungs-Grenzwertes (Dieser Parameter kann optional angegeben werden)
10		','		Adr_warning_high	MODBUS-Adresse des oberen Warnungs-Grenzwertes (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Stellpunkte hat folgenden Aufbau :



Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	','		Adr_nominal	MODBUS-Adresse für "nominal"
6		','		KL_Id	Nummer der Kennlinie, die zur Umrechnung der MODBUS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
8		','		Adr_actual	MODBUS-Adresse für "actual" (Dieser Parameter kann optional angegeben werden)

Die IS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	','		Adr_actual	MODBUS-Adresse für "actual"
6		','		KL_Id	Nummer der Kennlinie, die zur Umrechnung der MODBUS- in die physikalischen Werte angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) (Dieser Parameter kann optional angegeben werden)
7		','		Adr_alarm_high	MODBUS-Adresse des oberen Alarm-Grenzwertes (Dieser Parameter kann optional angegeben werden)
8		','		Adr_warning_high	MODBUS-Adresse des oberen Warnungs-Grenzwertes (Dieser Parameter kann optional angegeben werden)

**Hinweis:** Der Aufbau der IS-Zeile ab Spalte 29 stimmt mit dem Aufbau der herstellerspezifischen Ergänzungen in der FND-Referenzdatei ab Spalte 82 überein.

**Beispiel für eine FND-Referenzdatei :**

```
00000 10/09/1996 14:00 LHM Stadtsteueramt
00001 Meldepunkt_1_____ 1 0 15 15 15 0 0 000 045 Klartext für Meldepunkt           ADR_actual;URFkt
00002 Schalterpunkt_1_____ 2 0 15 15 15 0 0 000 045 Klartext für Schalterpkt.       ADR_nominal;URFkt;ADR_actual
00003 Messpunkt_1_____ 3 0 15 15 15 0 0 000 108 Klartext für Messpunkt           ADR_actual;KL_Id
00004 Stellpunkt_1_____ 4 0 15 15 15 0 0 000 108 Klartext für Stellpunkt       ADR_nominal;KL_Id;ADR_actual
00005 Zaehlpunkt_1_____ 5 0 15 15 15 0 0 000 108 Klartext für Zaehlpunkt       ADR_actual;KL_Id
...
```

**Initialisierung im Netzwerk (MODBUS/IP)**

Erreichbar sind alle die MODBUS-Geräte, die zum gleichen IP-Netzwerk wie der GA-Knoten gehören bzw. über entsprechende IP-Router erreichbar sind.

Alle MODBUS-Geräte müssen mittels der Datei "izNNconn.vbp" im Unterverzeichnis "iz" für die IP-Adressierung parametrieren werden (NN: Treiber-Id. (hexadezimal). Diese Datei hat das gleiche Format wie IZ03CONN.VBP", vgl. System-Spezifikation des GA-Knoten unter Abschnitt 5.4.6.3. Als Verbindungs-Art wird nur zugelassen :

41: TCP/IP über Netzwerk

Die Angabe der Verbindungs-Adresse erfolgt im Format :

"aaa.bbb.ccc.ddd#ppppp": IP-Adresse und -Port (ohne ")

Der Datenaustausch mit den MODBUS-Geräten wird gemäß "MODBUS Messaging on TCP/IP Implementation Guide" wie folgt initialisiert :

- Aufbau der TCP/IP-Verbindung

Die Initialisierung für ein MODBUS-Gerät wird wiederholt, nachdem die Verbindung beendet wurde oder eine Störung der Kommunikation behoben wurde.

**Initialisierung im MODBUS (MODBUS/RTU)**

Erreichbar sind alle die MODBUS-Geräte, die zum gleichen MODBUS wie das über die serielle Schnittstelle am GA-Knoten angeschlossene MODBUS-Gerät gehören.

**Spontane Meldungen und Polling der MODBUS-Variablen**

Das MODBUS-Protokoll unterstützt keine spontanen Meldungen. Daher werden die Werte aller parametrieren Variablen zyklisch gelesen (Polling im Master-Slave-Verfahren).

Für 1- und 2-stufige Melde- und Schaltpunkte wird zur Umrechnung des MODBUS-Wertes in den zugehörigen FND-Index die Invertierung (vgl. UR\_Fkt) wie folgt bereitgestellt :

1-stufige Invertierung:        MODBUS    FND  
  0 <-> 1  
  1 <-> 0

2-stufige Invertierung:        MODBUS    FND  
  0 <-> 2  
  1 <-> 1  
  2 <-> 0

### **Zugriffskontrolle**

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe gemäß FND-Spezifikation.

### **Grenzwert-Überwachung**

Für Mess- und Zählpunkte können optional MODBUS-Adressen für die Grenzwerte definiert werden. Die Grenzwertüberwachung wird vom IZ-Treiber emuliert, d.h. die MODBUS-Adressen werden zyklisch gemäß dem in der ES-Zeile definierten Abfrageintervall gepollt und geprüft.

Soweit in der MODBUS-Adresse eine lfd. Nummer für das MODBUS-Gerät angegeben wird, so muss diese auf dem gleichen MODBUS-Gerät realisiert sein wie die Führungsgröße.

### **Ausblendung der Betriebsmeldungen**

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet.

## **5.5. Elektronischer Schlüssellersatz (ES)**

LON-basierte Anlagen werden auch für die Realisierung eines **elektronischen Schlüssellersatzes** eingesetzt werden (**ES-System** in Anlehnung an die DIN V VDE 0830-8-1).

Jeder zutrittsberechtigter Nutzer erhält anstatt eines Schlüssels einen lesbaren Datenträger (z.B. Magnetstreifen-, Chip- oder ähnliche Karten im Kreditkartenformat TD-1 nach ISO 7810/7816, ...), auf dem eine eindeutige Schlüsselnummer, bestehend aus einem dem Objekt fest zugeordneten **Firmenkode** und dem Identifikation-Merkmal (**IM**), gespeichert ist. Die im Objekt installierten Erfassungseinheiten (**IME**) werden an dafür programmierte SPS bzw. LON-Knoten angeschlossen.

**Hinweis:** Die Zuordnung der IM's zum Nutzer erfolgt außerhalb des LON, des GA-Knotens und der LZH, d.h. es werden keine personenbezogenen Daten gespeichert.

Bis zur Version 1.2 wurde der elektronische Schlüssel (ES-Modul) im GA-Knoten unabhängig von der LZH installiert und durch direkte Bearbeitung der Arbeitssätze in den IM-Dateien ("abc...xyz.im") konfiguriert. Im Rahmen der Erweiterung zum Zutrittskontrollsystem (ZKS, in Anlehnung an die DINV 0830-8-1 und EN 50133) wurde das ES-Modul und die zugehörigen Dateien um Funktionen wie Doppelbenutzerkontrolle, Raumzonenwechselkontrolle ergänzt. Zusätzlich wurden die RZ- und ZZ-Kennung (vgl. ZZ\_0000.nnn) gemäß LZH-Standard auf 10 Zeichen erweitert. Hinzu kommen weitere Freigabe- und Schalt-/Stellpunkte. Als Folge davon haben sich die 3 Datei-Formate geändert.

**Hinweis:** Da die 3 Dateien RZ\_0000.nnn, ZZ\_0000.nnn und IM\_0000.nnn bisher nicht verwendet wurden, besteht kein Anpassungsbedarf in einer übergeordneten Applikation (z.B. Bedienprozess der LZH), d.h. diese entfallen. Das Format der vom GA-Knoten bereits genutzten Dateien "xxx.IM" bleibt auf den bisher verwendeten Positionen bestehen und wird am Ende erweitert, so dass die Kompatibilität zu bestehenden Installationen gewährleistet bleibt.

Ab der Version 1.3i unterstützt das ES-Modul die Zutrittskontrolle mit Eingabe einer Persönlichen Identifikationsnummer (PIN). Unterstützt werden Kombigeräte (IME und PIN-Eingabe in einem Gerät) als auch gesonderte Geräte für die PIN-Eingabe (P-IME), die einer IME zugeordnet werden.

Ab der Version 1.3i werden die IM-Dateien und die darin enthaltenen Arbeitssätze vom ES-Modul zyklisch überwacht. Änderungen nach Anzahl, Name, Zeitstempel oder Länge der IM-Dateien als auch nach Inhalt der enthaltenen Arbeitssätze werden als Meldung von Datenpunkten an die Leitzentrale weitergeleitet.

Ab der Version 1.3i können die IM-Dateien in einer TGZ-Datei gepackt und derart an den GA-Knoten übertragen werden. Der GA-Knoten prüft das Vorhandensein, archiviert ggf. die bestehenden IM-Dateien und entpackt die neuen IM-Dateien.

Ab Version 1.3k wurde die Zutrittskontrolle um das Hilfsprogramm CLR\_IMS ergänzt. Dieses kann ohne Aufrufparameter z.B. mittels eines CRON-Jobs zyklisch gestartet im Arbeitsverzeichnis werden und dient zum Löschen der IMS-Dateien unter folgenden Bedingungen :

- 1.) die in der IMS-Datei vermerkte Raumzone "RZ\_Kenn" kann wegen abgelaufener Gültigkeit nicht mehr betreten bzw. verlassen werden
- 2.) die in der IMS-Datei vermerkte Raumzone "RZ\_Kenn" kann wegen nicht mehr bestehender Zutrittsberechtigung gemäß IM-Datei nicht mehr betreten bzw. verlassen werden

Der Aufruf des Programmes erfolgt im Arbeits-Verzeichnis des GA-Knotens.

Jeder Programmlauf wird in der Datei CLR\_IMS.LOG protokolliert. Dabei werden Zeitpunkt und Grund der Löschung für jede IMS-Dateien angegeben.

### **5.5.1. Funktionsbeschreibung**

Beim Start initialisiert der LON-Knoten die IME und erfasst die eingelesenen Schlüsselnummern (Firmenkode und IM) und andere Meldungen (z.B. Störung, Sabotage, usw.) permanent. Wird eine Schlüsselnummer empfangen, so prüft die LON-Knoten die Zulässigkeit des gelesenen Firmenskode mit der intern hinterlegten Liste. Im Fehlerfall weist bereits der LON-Knoten den Zutrittsversuch ab.

Andernfalls meldet der LON-Knoten das IM (ohne Firmenkode) im Datenteil eines spezifischen FND-Transferpunkt spontan (USM; vgl. unten) an den GA-Knoten. Jeder IME ist ein solcher Transferpunkt eindeutig zugeordnet, d.h. die IME wird über die zugehörige Datenpunkt-Adresse identifiziert.. Der GA-Knoten prüft das IM auf Zutrittsberechtigung anhand der zugeordneten Raum- und Zeitzone, sowie einem Datenpunkt für die Freigabe. Abhängig vom Ergebnis der Prüfung sendet der GA-Knoten für denselben Transferpunkt ein CMD-Datagramm, das das Akzeptieren (ACK) oder Abweisen (NAK) des Zutrittsversuches anzeigt.

Die einzelnen Zutrittsberechtigungen sind in der IM-spezifischen Datei im Unterverzeichnis "es" mit dem Namen "<IM>.im" als Arbeitssätze zeilenweise vermerkt (vgl. Formatbeschreibung der IM-Datei im nachfolgenden Datenkatalog). Der Zutritt wird nur dann gewährt, wenn die IM-Datei vorhanden ist, zum Lesen geöffnet werden kann und ein Arbeitssatz allen Bedingungen der Zutrittsberechtigung erfüllt wie folgt:

- die angegebene IME-Kennung entspricht der Datenpunkt-Adresse des gelesenen FND-Transferpunktes
- für alle angegebenen Freigabe-Datenpunkte stimmt der "actual" mit dem zugeordneten Vergleichswert überein
- die aktuelle Systemzeit liegt im angegebenen Gültigkeits-Zeitraum
- die aktuelle Systemzeit liegt im angegebenen Zeitbereich (Zeitzone)

Wurde die Betriebsart ZKZ angegeben, so werden die zusätzliche Bedingungen geprüft. Dazu werden IM-spezifische Status- und PIN-Informationen in gesonderten Dateien vermerkt (vgl. Formatbeschreibung der IMS- und PIN-Datei im nachfolgenden Datenkatalog). Diese enthalten Informationen über den letzten gewährten Zutritt bzw. eine zuvor eingegebene PIN. Der Zutritt wird nur dann gewährt, wenn im Arbeitssatz die folgenden Bedingungen erfüllt sind:

- Ist die Doppelbenutzerkontrolle angegeben, so darf die Raumzone, die mit dem Zutritt erreicht wird, nicht als aktuelle Raumzone in der IMS-Datei vermerkt sein bzw. die IMS-Datei fehlt

- Ist die Raumzonenwechselkontrolle angegeben, so muss die Raumzone, die mit dem Zutritt verlassen wird, als aktuelle Raumzone in der IMS-Datei vermerkt sein, der Eintrag in der IMS-datei ist leer oder die IMS-Datei fehlt
- Ist eine PIN angegeben, so muss diese mit der von der IME gelesenen oder der in der PIN-Datei vermerkten, noch gültigen PIN (Freigabedauer, Anzahl ungültiger Wiederholungen) übereinstimmen oder die angegeben PIN ist "\_\_\_\_\_".

**Hinweis:** Im AfID wurde eine abweichende Raumzonen-Zuordnung gefordert. Die "harte" Raumzonenwechselkontrolle wird bei ZKZ-Betrieb alternativ aktiviert, wenn im Arbeitssatz eine Reset-IME angegeben ist.

### **5.5.2. PIN-Eingabe**

Das Übertragungsverfahren zwischen ZKZ und IME wird um die IM-bezogene Übertragung einer optionalen PIN erweitert. Bei einem Kombigerät folgt die PIN dem IM, getrennt durch ein Semikolon (";"). Erfolgt die PIN-Eingabe an einer gesonderten P-IME, so wird die P-IME einer IME über die Konfigurationsdatei (vgl. GAK.CFG) fest zugeordnet. Nach Übertragung des an dieser IME gelesenen IM wird die zuvor eingegebene PIN dem IM in der ZKZ zugeordnet. Die Eingabe des IM muss innerhalb einer parametrierbaren Wartezeit nach der PIN-Eingabe erfolgen. Bestehende Übertragungs-Funktionen bleiben unverändert erhalten (abwärtskompatibel).

Wurde eine PIN angegeben, so wird diese in die Prüfung der Zutrittsberechtigung einbezogen. Fehlt die PIN, so wird der Zutritt nur dann gewährt, wenn keine PIN parametrierbar wurde oder eine parametrierbare Freigabedauer (Parameter "A\_Zeit" in IM-Datei, vgl. auch Datenbankfeld "AUFENTHALTSZEIT") seit der letzten PIN-Eingabe noch nicht abgelaufen ist. Ist einer Zutrittsberechtigung im Arbeitssatz die PIN "\_\_\_\_\_" zugeordnet, so wird die Freigabedauer rückgesetzt. Auf diese Weise kann z.B. mit dem Verlassen eines Gebäudes die erneute PIN-Eingabe erzwungen werden (Abmeldung des Nutzers).

Wird der Zutritt 3-malig nacheinander abgelehnt, weil eine ungültige PIN angegeben wurde, so wird das IM für alle IME mit Betriebsart ZKS gesperrt, bis diese Sperrung mit der Reset-IME rückgesetzt wird.

### 5.5.3. Status-/Modus-Anzeige

Ab der Version 1.3i werden die IM-Dateien und die darin enthaltenen Arbeitssätze vom ES-Modul zyklisch überwacht. Änderungen nach Anzahl, Name, Zeitstempel oder Länge der IM-Dateien als auch nach Inhalt der enthaltenen Arbeitssätze werden als Meldung des nachfolgend beschriebenen Datenpunktes an die Leitzentrale weitergeleitet. Ausstehende Übertragungen lassen sich an der übergeordneten Zentrale daran erkennen, dass die erwartete Meldung ausbleibt :

`%ES_IMA_MODUS` interner Meldepunkt (Treiber-Id. 240) mit Attribut EIN/AUS (atb\_id: 10), START/STOP (14), AUTOMATIK/HAND (23), BETRIEB/INITIALISIERUNG (30) o.a. mit sinngemäßer Zuordnung der Meldungsstufen (Voreinstellung, falls IM-Datei vorhanden: Meldungsstufe 0, z.B. BETRIEB)

Störungen beim Zugriff auf die IM-Dateien oder beim Auswerten der Arbeitssätze werden auf den nachfolgend beschriebenen Datenpunkt abgebildet und ggf. an die Leitzentrale gemeldet.

`%ES_IMA_STATUS` interner Meldepunkt mit Attribut NORMAL/STÖRUNG (Voreinstellung: NORMAL)

**Hinweis:** Die zyklische Überwachung wird nur unter LINUX unterstützt. Die o.g. Datenpunkte müssen hierfür in der Konfigurationsdatei parametrierbar sein. Ist `%ES_IMA_MODUS` nicht parametrierbar, so wird die zyklische Überwachung unterdrückt.

Das Verfahren wird in einem Abstand von min. 1 Minute gestartet, so dass ein "Flimmern" der Meldungen auch während der Übertragung größerer IM-Dateien verhindert wird.

Im ersten Schritt eines Zyklus wird eine Liste mit den Namen aller IM-Dateien (ESIMACHK.LIM) erzeugt. Nachfolgend wird Zeitstempel und Länge für jede aufgelistete IM-Datei ermittelt und zusammen mit dem Dateinamen in die Berechnung eines Hash-Kodes (ähnlich Summen- bzw. CRC-Prüfung) einbezogen.

Auf diese Weise werden Änderungen im Normal-Betrieb schnell erkannt und gemeldet. Unerwartete Änderungen, die in Kenntnis des Verfahrens und unter Beibehaltung der Datei-Parameter durchgeführt werden, werden durch ein Zusatzverfahren ermittelt. Dabei wird der Inhalt der einzelnen Arbeitssätze in die Bildung des Hash-Kodes einbezogen. Dieses Zusatzverfahren läuft abhängig von der System-Belastung nur mit niedriger Priorität ab.

Wird eine Änderung der Anzahl oder des Hash-Kodes gegenüber dem letzten vermerkten Wert erkannt, so wird `%ES_IMA_MODUS` auf die Meldungsstufe 1 (z.B. INITIALISIERUNG) gesetzt und die Spontan-Meldung ausgelöst.

Liegt keine Änderung vor, so wird %ES\_IMA\_MODUS ggf. auf die Meldungsstufe 0 (z.B. BETRIEB) rückgesetzt und die Spontan-Meldung ausgelöst.

Hash-Kode und Anzahl der an der Berechnung beteiligten IM-Dateien werden in der Datei ESIMACHK.KOD vermerkt und beim nächsten Programmstart geladen.

Wird beim Zugriff auf eine IM- bzw. ESIMACHK-Datei oder beim Auswerten eines Arbeitssatzes einer der nachfolgend aufgeführten Fehler erkannt, so wird neben %ES\_IMA\_MODUS auch der Datenpunkt %ES\_IMA\_STATUS auf die Meldungsstufe 1 (STÖRUNG) gesetzt, der Name der Datei vermerkt und die Spontan-Meldung ausgelöst. Beim Programmstart bzw. wenn Zugriff und Auswertung auf die betroffene Datei wieder korrekt verlaufen, so werden %ES\_IMA\_MODUS und %ES\_IMA\_STATUS ggf. auf die Meldungsstufe 0 (z.B. NORMAL) rückgesetzt und die Spontan-Meldung ausgelöst. Auslöser für die STÖRUNG sind Fehler beim :

- Ausführen des Scripts "estgz2im"
- Zugriff auf die Datei "ESIMACHK.LIM"
- Zugriff auf eine IM-Datei, ggf. KEINE IM-Datei vorhanden
- Vergleich der in der IM-Datei angegebenen IME-Kenn\_KOMMT mit zugehörigem Datenpunkt in GAK.CFG
- Auswerten der Formatierung IM-Datei (Zeilenlänge <= 92 Zeichen, fehlendes Leerzeichen zwischen 2 Spalten)

#### **5.5.4. Entpacken vom IM-Dateien**

Desweiteren wird geprüft, ob im Unterverzeichnis "es" eine oder mehrere Dateien mit dem Namen "IM-  
jjjjmmthhmmss.INIT.tgz" oder "IM-  
jjjjmmthhmmss.CHANGE.tgz" ("jjjj..." Datum/Zeit) vorliegen. Falls JA, so werden die darin enthaltenen IM-Dateien durch Aufruf des Scripts "estgz2im" (Unterverzeichnis "exe") wie folgt übernommen:

- Sichern der aktuellen IM-Dateien als TGZ-Datei "IM-  
jjjjmmthhmmss.ARCHIV.tgz" im Unterverzeichnis es/ARCHIV
- Für "mmmm"=INIT Löschen der aktuellen IM-Dateien
- Entpacken der IM-Dateien aus der TGZ-Datei ins ES-Verzeichnis
- Verschieben der TGZ-Datei ins Unterverzeichnis es/ARCHIV

**ES LIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit der Bearbeitung der elektronischen Schlüssellersatzes zusammenhängen (Identifikations-Merkmale, Raum-/Zeitzone und zugehörige Karteipunkte).

**ES DLIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die als DUMMY in den GA-Knoten eingebunden werden müssen, wenn keine Unterstützung des elektronischen Schlüssellersatzes gewünscht ist.



**estgz2im**      Dieser Script steuert die Übernahme neuer IM-Dateien aus einer TGZ-Datei.

**ESIMACHK.LIM**      Liste mit den Namen der aktuellen IM-Dateien

**ESIMACHK.KOD**      Hash-Kode und Anzahl der aktuellen IM-Dateien

### **5.5.5. Übersicht der verwendeten Module**

ES	Elektronischer Schlüsseleratz.
UT	allgemeine Routinen-Bibliothek
SY	systemspezifische Routinen-Bibliothek

### **5.5.6. Liste mit Namen und Aufgaben**

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

ES_INIT	Initialisierung beim Programmstart
ES_CMD	Behandlung CMD-Datagramm von LZH (Karteipunkt)
ES_RSP_USM	Behandlung RSP-/USM-Datagramm von IZ's
ES_POLL	zyklische Bearbeitung der aktiven Schaltprogramme
ES_EXIT	Ende-Behandlung beim Programmstop

Folgende Routinen werden nur innerhalb dieses Modules verwendet:

ES_IM_CHK	Prüfen eines Identifikations-Merkmals auf Zutrittsberechtigung
ES_IMACHK	Zykl. Überwachung der IM-Dateien auf Änderung und Störung

**5.5.7. Aufrufstruktur**

Wegen der Komplexität der Aufrufstruktur werden Teilbäume nur erstmalig dargestellt und bei wiederholter Verwendung auf diesen verwiesen. Die Wurzel des Teilbaumes ist durch Fettschrift und Unterstreichung hervorgehoben.

**ES\_INIT**

**ES\_CMD**

**ES\_RSP\_USM**

**ES\_IM\_CHK**

**ES\_POLL**

**ES\_IMACHK**

**ES\_EXIT**

### 5.5.8. Modulbeschreibungen

Die nachfolgend beschriebenen Routinen verwenden die einheitlichen Fehlerkodierungen ES\_ERR\_... wie folgt:

```
/*      -- Fehler - Codes des SP-Moduls                                */
# define ES_NORMAL          0      /* KEIN Fehler                                */

/*      --- Allgemeine Fehler                                        */
# define ES_ERR_OPEN_FILE   -1     /* OPEN-Fehler auf Datei                    */
# define ES_ERR_READ_FILE  -2     /* READ-Fehler auf Datei                   */
# define ES_ERR_WRITE_FILE -3     /* WRITE-Fehler auf Datei                  */
# define ES_ERR_END_OF_FILE -4     /* Dateiende erreicht                      */
# define ES_ERR_ALLOC_MEM  -5     /* ALLOC-Fehler auf Speicher               */

# define ES_ERR_FORMAT      -6     /* Allgemeiner Fehler in der              */
/*                          /* Syntax bzw. dem Format                  */
/*                          /* einer Eingabe, Datei oder             */
/*                          /* Aufrufparameter                       */

# define ES_ERR_DP          -7     /* Unzulässiger Datenpunkt                */
# define ES_ERR_WERT        -8     /* Unzulässiger Wert für den              */
/*                          /* DP-Befehl                              */

/*      --- FND-Fehler für gesendeten Befehl empfangen                */
# define ES_ERR_FND_ERROR   -90    /* FND-ERROR empfangen                   */
# define ES_ERR_FND_REJECT -91    /* FND-REJECT empfangen                  */
```

```

/*****
/* Produkt : GA - Knoten Version : 1.3i */
/* Modulname : ES Elektr. Schlüssel Stand : 01.08.2005*/
/*****
/* Datei : INIT.C */
/* Aufruf : ES_INIT ( ) */
/*****
/* Beschreibung : */
/* Die Routine ES_INIT muß als erste Routine des ES-Modul */
/* aufgerufen werden. Sie aktiviert das Modul und führt alle */
/* nötigen Initialisierungen durch. */
/* */
/* aufgerufen von : GAK_INIT */
/* */
/* ruft auf : UT_WSL_INIT */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int ES_INIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* -7: WSL_INIT-Fehler */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.10.98 1.2 */
/* Änderung: W. Fries 01.07.2001 1.3 WF1 */
/* Änderung: W. Fries 01.08.2005 1.3i WF2 */
/* ----- */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : ES Elektr. Schlüssel Stand : 01.10.98 */
/*****
/* Datei : CMD.C */
/* Aufruf : ES_CMD ( cCMD ) */
/*****
/* Beschreibung : */
/* Diese Routine vermerkt den in "cCMD" gelieferten Raumzonen-, */
/* Zeitzonen- oder Zutrittssatz in einer zugehörigen Datei */
/* (Segmentierter Karteipunkt gemäß LV unter 4.4 ff.). */
/* Ist der Aktivierungszeitpunkt früher als der bisher */
/* als nächster vermerkte, so wird der neue Zeitpunkt als nächster */
/* auszuführender vermerkt. */
/* Als Ergebnis wird der Zeiger auf das für "cCMD" zu liefernde */
/* RSP-Datagramm geliefert. */
/* */
/* aufgerufen von : GAK_ILZ */
/* */
/* ruft auf : ES_RZ_CMD, ES_ZZ_CMD, ES_IM_CMD, */
/* UT_OBJ_GETI, UT_FNDERROR, UT_FNDREJECT, */
/* diverse Makros FND... und SET_FND... */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* char* cCMD Zeiger auf das auszuwertende */
/* CMD-Datagramm mit dem neuen */
/* Schaltprogramm bzw. Kalender- */
/* Eintrag */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* char* ES_CMD Zeiger auf das RSP-Datagramm */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.10.98 1.2 */
/* Änderung: */
/* ----- */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.3i   */
/*  Modulname :   ES Elektr. Schlüssel   Stand    :   01.08.2005*/
/*****
/*  Datei     :   RSP_USM.C             */
/*  Aufruf    :   ES_RSP_USM ( cDatagramm ) */
/*****
/*  Beschreibung :                       */
/*  Die Routine führt die Bearbeitung der von den DDC-IZ's empfang. */
/*  RSP- und USM-Datagramme durch.     */
/*                                     */
/*  aufgerufen von :   GAK_IIZ          */
/*                                     */
/*  ruft auf      :   ES_IM_CHK         */
/*                                     */
/*  I/O-Kanäle    :                       */
/*                                     */
/*****
/*  Eingabe Parameter :                       */
/*  ----- */
/*  Typ      Name                Dim.   Beschreibung          */
/*  char     cDatagramm          RSP-   bzw. USM-Datagramm    */
/*  ----- */
/*  Ausgabe Parameter:                       */
/*  ----- */
/*  Typ      Name                Dim.   Beschreibung          */
/*  int      ES_RSP_USM          Returnkode der Funktion */
/*  0=FALSE: KEIN Fehler          */
/*  ----- */
/*  Transiente Parameter :                       */
/*  ----- */
/*  Typ      Name                Dim.   Beschreibung          */
/*  *** KEINE ***                */
/*                                     */
/*****
/*                                     Autor      Datum      Version   Kurzzeichen */
/*  ----- */
/*  Original :   W. Fries          01.10.98      1.2        */
/*  Änderung:   W. Fries          01.12.2000    1.3        WF1 */
/*  Änderung:   W. Fries          01.08.2005    1.3i       WF2 */
/*                                     */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.3i */
/* Modulname : ES Elektr. Schlüssel Stand : 01.08.2005*/
/*****
/* Datei : POLL.C */
/* Aufruf : ES_POLL ( ) */
/*****
/* Beschreibung : */
/* Die Routine führt die zyklische Bearbeitung der elektronischen */
/* Schlüssel durch und liefert ein an die IZ weiterzuleitendes */
/* CMD-Datagramm als Ergebnis, wenn ein solches existiert. */
/* */
/* aufgerufen von : GAK_IES */
/* */
/* ruft auf : ES_IMACHK, UT_WSL_READ, UT_DP_FINDFIRST */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* char* ES_POLL Zeiger auf das an IZ auszugeben-*/
/* de CMD-Datagramm */
/* SONST: NULL-Zeiger */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.10.98 1.2 */
/* Änderung: W. Fries 01.08.2005 1.3i WF1 (IMACHK) */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : ES Elektr. Schlüssel Stand : 01.10.98 */
/*****
/* Datei : EXIT.C */
/* Aufruf : ES_EXIT ( ) */
/*****
/* Beschreibung : */
/* Diese Routine führt die spezifischen ENDE-Behandlungen für das */
/* ES-Modul vor dem Stoppen des Hauptprogrammes durch : */
/* - Schließen der Warteschlange für CMD-Datagramme */
/* */
/* aufgerufen von : GAK_EXIT */
/* */
/* ruft auf : UT_WSL_EXIT */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int ES_EXIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* -1: OPEN-Fehler */
/* -2: READ-Fehler */
/* -3: WRITE-Fehler */
/* -4: ALLOC-Fehler */
/* -5: unzulässige IS-Zeile */
/* -6: DDC-IZ-Init-Fehler */
/* >0: Nummer der Zeile, in der */
/* ein Fehler erkannt wurde */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.10.98 1.2 */
/* Änderung: */
/* */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/* Produkt : GA - Knoten Version : 1.3i */
/* Modulname : ES Elektr. Schlüssel Stand : 01.11.2005*/
/*****
/* Datei : IM_CHK.C */
/* Aufruf : ES_IM_CHK ( cIME, cIM, IM_ACK, uProtFlag ) */
/*****
/* Beschreibung : */
/* Die Routine führt die Prüfung des von "cIME" gelesenen "cIM" */
/* auf Zutrittsberechtigung durch und liefert den Status */
/* ES_IME_CMD_ACK, falls diese gewährt wird, sonst ES_IME_CMD_NAK. */
/* Wird der Zutritt gewährt, so enthält "IM_ACK" die auszuführenden*/
/* Schalt- und Stellbefehle gemäß der Struktur ES_IM_ACK. */
/* */
/* aufgerufen von : ES_RSP_USM */
/* */
/* ruft auf : */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* char cDatagramm RSP- bzw. USM-Datagramm */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* int ES_IM_CHK Returnkode der Funktion */
/* ES_IME_CMD_ACK: Zutritt gewährt */
/* ES_IME_CMD_NAK: Zutritt verwei- */
/* gert */
/* uns.* uProtFlag Protokollierungs-Flag bitweise */
/* kodiert */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.10.98 1.2 */
/* Änderung: W. Fries 01.12.98 1.2a WF1 */
/* Änderung: W. Fries 01.12.2000 1.3 WF2+ProtFlag */
/* Änderung: W. Fries 01.07.2001 1.3 WF3 (AfID) */
/* Änderung: W. Fries 01.11.2005 1.3i WF4 (P-IME) */
/* ----- */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.3i */
/* Modulname : ES Elektr. Schlüssel Stand : 14.07.2005*/
/*****
/* Datei : IMACHK.C */
/* Aufruf : ES_IMACHK ( ) */
/*****
/* Beschreibung : */
/* Diese Routine führt die zyklische Überprüfung aller IM-Dateien */
/* durch nach folgendem Verfahren: */
/* - Erstellen einer Liste mit den Namen aller IM-Dateien */
/* - Über alle Einträge berechnen von Hash- und CRC-Kode für Datei-*/
/* Name, -Zeitstempel und -Länge */
/* - Über alle Einträge berechnen von Hash- und CRC-Kode für IM-*/
/* Arbeitssätze */
/* */
/* Das Verfahren wird nach jedem Einzelschritt unterbrochen und */
/* abhängig von der Systembelastung fortgesetzt, so dass die Beein-*/
/* trächtigung anderer Module minimiert wird. */
/* */
/* aufgerufen von : ES_POLL */
/* */
/* ruft auf : ES_KOD_WRIT */
/* SY_PS_CPOLL, SY_PS_CFOURK, SY_PS_CSYSY */
/* SY_FI_FINDFIRST, SY_FI_FINDNEXT, */
/* SY_FI_DT, SY_FI_SIZE */
/* UT_DP_FINDFIRST, UT_NORM_STRING, */
/* UT_DT_ADD */
/* */
/* I/O-Kanäle : */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* uns. long ES_IMACHK Flags für Änderung usw. kodiert */
/* wie ulES_IMACHK_Flags */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 14.07.2005 1.3i */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.3i */
/* Modulname : ES Elektr. Schlüssel Stand : 14.07.2005*/
/*****
/* Datei : KOD_WRIT.C */
/* Aufruf : ES_KOD_WRIT ( ulIMACHK_Flags ) */
/*****
/* Beschreibung : */
/* Diese Routine vermerkt die akt. KOD-Struktur in der KOD-Datei. */
/* Die derzeitigen "ulIMACHK_Flags" werden abhängig vom Zugriffs- */
/* verhalten aktualisiert und als Ergebnis geliefert. */
/* */
/* aufgerufen von : ES_IMACHK */
/* */
/* ruft auf : */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter :
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* uns. long ulIMACHK_Flags Flags von KOD_WRIT kodiert */
/* wie ulES_IMACHK_Flags */
/* */
/* ----- */
/* Ausgabe Parameter:
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* uns. long ES_KOD_WRIT Flags nach KOD_WRIT kodiert */
/* wie ulES_IMACHK_Flags */
/* */
/* ----- */
/* Transiente Parameter :
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 14.07.2005 1.3i */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### **5.5.9. Druckausgaben, Meldungstexte**

In der Betriebsart ZKZ und bei gesetztem Protokoll-Flag wird der Zutrittsversuch protokolliert (vgl. DR-Zeile in der Konfigurationsdatei GAK.CFG).

### **5.5.10. Datenkatalog**

Für die zyklische Überwachung der IM-Dateien und der darin enthaltenen Arbeitssätze werden folgende Dateien im Arbeitsverzeichnis angelegt:

ESIMACHK.KOD

ESIMACHK.LIM

Die Datei "ESIMACHK.KOD" enthält, falls vorhanden, den letzten ermittelten Hash-Kode sowie die Anzahl der daran beteiligten IM-Dateien. Diese Angaben werden beim Programmstart vom ES-Modul geladen, so dass auch Änderungen erkannt werden, die bei gestopptem Programm durchgeführt werden.

"ESIMACHK.LIM" wird temporär erzeugt und enthält ggf. die Liste der zu prüfenden IM-Dateien..

Für die Zutrittskontrolle mit PIN-Eingabe wird folgende Datei im Arbeitsverzeichnis angelegt:

<IME>.PIN

"<IME>" entspricht der Kennung (dp\_id) der IME, an der nach der PIN-Eingabe das zugehörige IM eingelesen werden muss. Wird an einer P-IME eine PIN eingegeben, so werden in "<IME>.PIN" die eingegebene PIN und der Zeitpunkt vermerkt, bis zu dem das IM eingelesen sein muss.

Für die Konfiguration des elektronischen Schlüsselersatzes werden mehrere Dateien benötigt. Diese werden im Verzeichnis ES und den darunterliegenden Unterverzeichnissen des aktuellen Arbeitsverzeichnisses gehalten.

Folgende Dateien stehen im Verzeichnis \GAK\ES zur Verfügung :

ABCDEFGH\IJ...\XYZ.IM

Soweit sinnvoll, lehnen sich die Formate für die einzelnen Parameter an die FND-Spezifikation an. Es gelten die folgenden Vereinbarungen :

- Alle Zeilen haben eine maximale Länge von 128 Zeichen (zzgl. <CR> (0x0D) und/oder <LF> (0x0A) ).
- Alle Zahlen stehen rechtsbündig in den entsprechenden Spalten und repräsentieren in Dezimaldarstellung die entsprechenden Bitkombinationen der FND-Variablen.
- Die Datenpunkt-Adressen ("dp\_id") und die zugehörigen Bezeichnungen stehen linksbündig in der jeweiligen Spalte. Sie beginnen und enden mit einem alphanumerischen Zeichen. Nicht benutzte Stellen werden mit Leerzeichen ausgefüllt.

**Hinweis: Die abschließenden Leerzeichen sind KEIN signifikanter Bestandteil der Datenpunkt-Adresse, d.h. diese werden gemäß FND-Spezifikation durch NULL-Bytes (0x00) ersetzt !!**

Der Wertebereich für 3-stellige numerische Angabe ist, soweit nicht anders angegeben, auf  $0 \leq x \leq 255$  begrenzt. Für 5-stellige numerische Angaben gilt ein maximaler Wertebereich von  $0 \leq x \leq 32767$ .

Reelle Zahlen (14-stellig) können als ganze Zahlen bzw. in Gleitkomma- oder Exponential-Darstellung angegeben werden (inkl. Vorzeichen für Mantisse und Exponent). Dabei ist jedoch die Auflösung auf eine 7-stellige Mantisse und der Wertebereich, soweit nicht anders angegeben, auf annähernd  $-3.4E+38 \leq x \leq +3.4E+38$  begrenzt.

Jede Datei enthält einen Aktivierungszeitpunkt und sämtliche ab dann geltenden zugehörigen Parameter ihrer Art. Mit der Aktivierung werden alle zuvor geltenden Parameter (ältere Version) ungültig und die zugehörigen Dateien gelöscht. Zusätzlich werden aus den Parametern für die aktiven Raumzonen, Zeitzonen und Identifikationsmerkmale IM-Dateien erzeugt, wobei das Identifikationsmerkmal den Namen der Unterverzeichnisse und der Datei mit der Extension "IM" entspricht, wenn diese zusammengesetzt werden (z.B. ABC => ABC.IM, ABCDEFGH => ABCDEFGH.IM und ABCDEFGHIJK => ABCDEFGHIJK.IM).

Die Dateien <IM>.im - mit "<IM>" als dem IM im Namen der Unterverzeichnisse und der Datei - enthalten die IM-Datensätze ab Zeile 1 im folgenden Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	16	16	IME-Kenn_KOMMT	IME-Kennung wie in der LZH. Für IME an IZnn entspricht diese der DP-Id. des zugehörigen

					Transferpunktes vom Typ 0.
2	18	33	16	Freigabe-DP_1	Adresse eines Melde- oder Schaltpunktes für die Freigabe des IM (optional). Ist keine Adresse angegeben, so wird der Zutritt nur abhängig von der zugeordneten Raum- und Zeitzone gewährt
3	35	37	3	Freigabe-Wert_1	Nur falls der "actual" des Freigabe-DP mit diesem Wert übereinstimmt wird der Zutritt gewährt
4	39	41	3	Code_1	Zeitbezogene Dimension gemäß FND-Ergänzungen (160 ff.) für Zeitbereich 1
5	43	56	14	VON_1	
6	58	71	14	BIS_1	
7	73	75	3	Code_2	Zeitbezogene Dimension gemäß FND-Ergänzungen (160 ff.) für Zeitbereich 2
8	77	90	14	VON_2	
9	92	105	14	BIS_2	
10	107	110	4	PIN	PIN-Code
11	112	125	14	O_Zeit	Offenzeit in Minuten
12	127	140	14	A_Zeit	>0: Aufenthaltszeit in Minuten <0: Freigabedauer für eine zuvor angegeben PIN in Minuten
13	142	144	3	Prot.-Flag	Flag für Protokoll-Ausgaben (entspricht dem bitweisen ODER der Prot.-Flag aus dem Raumzonen-, Zeitzonen- und Zutrittssatz, aus dem dieser Satz gebildet wurde)  *** Bit-Zuordnung noch nicht unterstützt, deshalb momentan begrenzt auf die Werte 0 und 255 (KEINE bzw. VOLLE Protokollierung); *** bleibt reserviert für zukünftige Erweiterungen !  Die Betriebsart ist im Bit 8 kodiert, so dass gilt : 0 bis 255: ZKS 256 bis 511: ESE
14	146	161	16	DP_Adr._1	DP-Adr. für Schalt- bzw. Stellbefehl 1 oder LEER
15	163	176	14	NOM_1	"nominal" für Schalt- bzw. Stellbefehl 1, falls def.
16	178	193	16	DP_Adr._2	DP-Adr. für Schalt- bzw. Stellbefehl 2 oder LEER
17	195	208	14	NOM_2	"nominal" für Schalt- bzw. Stellbefehl 2, falls def.
18	210	225	16	DP_Adr._3	DP-Adr. für Schalt- bzw. Stellbefehl 3 oder LEER
19	227	240	14	NOM_3	"nominal" für Schalt- bzw. Stellbefehl 3, falls def.
20	242	257	16	DP_Adr._4	DP-Adr. für Schalt- bzw. Stellbefehl 4 oder LEER
21	259	272	14	NOM_4	"nominal" für Schalt- bzw. Stellbefehl 4, falls def.

```

--- Erweiterter Bereich ab V1.3 (Projekt AfID) -----
(wird aus Kompatibilittsgrnden zu den bestehenden Systemen in den
Objekten FRKA, BERS, ... angefgt. Aus Grnden der Funktionssicherheit
sollte dieser Bereich nur programmtechnisch erzeugt, jedoch NICHT
editiert werden !!!)
    
```

22	274	287	14	Gltig_AB	Datum/Zeit ab der der Eintrag gltig ist (JJJJMMTTHHMSS)
23	289	302	14	Gltig_BIS	Datum/Zeit bis zu der der Eintrag gltig ist (JJJJMMTTHHMSS)
24	304	319	16	Freigabe-DP_2	Adresse eines Melde- oder Schaltpunktes fr die Freigabe des IM oder LEER. Ist keine Adresse angegeben, so wird der Zutritt nur abhngig von der zugeordneten Raum- und Zeitzone gewhrt
25	321	323	3	Freigabe-Wert_2	Nur falls der "actual" des Freigabe-DP mit diesem Wert bereinstimmt wird der Zutritt gewhrt
26	325	340	16	Freigabe-DP_3	Adresse eines Melde- oder Schaltpunktes fr die Freigabe des IM oder LEER. Ist keine Adresse angegeben, so wird der Zutritt nur abhngig von der zugeordneten Raum- und Zeitzone gewhrt
27	342	344	3	Freigabe-Wert_3	Nur falls der "actual" des Freigabe-DP mit diesem Wert bereinstimmt wird der Zutritt gewhrt
28	346	348	3	Doppel_Benutz	>0: Doppelbenutzerkontrolle
29	350	352	3	RZ_Wechsel	>0: Raumzonenwechselkontrolle
30	354	367	14	Zutritts_Wdh_Zeit	Zeitraum in Minuten, in dem fr das IM ein erneuter Zutritt in die Raumzone unterbunden wird *** momentan nicht genutzt ***
31	369	371	3	Zwei_Pers_Zeit	Zeitraum in Sekunden innerhalb der 2 verschiedene IM ihre Zutrittsberechtigung nachweisen mssen (Zwei-Personen-Zutrittskontrolle) *** momentan nicht genutzt ***
32	373	375	3	Anw_Pers_min	min. Anzahl anwesender Personen (Mehr-Personen-Anwesenheitskontrolle) *** momentan nicht genutzt ***
33	377	392	16	IME-Kenn_RESET	IME-Kennung fr Rcksetzen der Kontroll-Funktionen (Raumwechsel, Doppelbenutzer usw.) wie in der LZH. Fr IME an IZnn entspricht diese der DP-Id. des zugehrigen Transferpunktes vom Typ 0.  Hinweis: Ist Reset-IME def., ===== dann erfolgt die AfID-spezifische "harte Doppelben.-/Raumzonenwechsel-Kontrolle" (vgl. IMS-Datei)
34	394	403	10	RZ-Kenn_KOMMT	Kennung der Raumzone wie in der LZH in die man beim Raumzonenwechsel KOMMT. Ist in "RZ_oooo.nnn" der "Teil_von_RZ" definiert, so wird diese Kennung verwendet, sonst die Kennung der RZ selbst.
35	405	414	10	RZ-Kenn_GEHT	Kennung der Raumzone wie in der LZH aus der man beim Raumwechsel

					GEHT (EINBAU_KENNUNG_RZ der IME)
36	416	425	10	ZZ-Kenn	Zeitzone-Kennung wie in der "ZZ_oooo.nnn", aus der dieser Datensatz erzeugt wurde
37	427	442	16	DP_Adr._5	DP-Adr. für Schalt- bzw. Stellbefehl 5 oder LEER
38	444	457	14	NOM_5	"nominal" für Schalt- bzw. Stellbefehl 5, falls def.
39	459	474	16	DP_Adr._6	DP-Adr. für Schalt- bzw. Stellbefehl 6 oder LEER
40	476	489	14	NOM_6	"nominal" für Schalt- bzw. Stellbefehl 6, falls def.
41	491	506	16	DP_Adr._7	DP-Adr. für Schalt- bzw. Stellbefehl 7 oder LEER
42	508	521	14	NOM_7	"nominal" für Schalt- bzw. Stellbefehl 7, falls def.
43	523	538	16	DP_Adr._8	DP-Adr. für Schalt- bzw. Stellbefehl 8 oder LEER
44	540	553	14	NOM_8	"nominal" für Schalt- bzw. Stellbefehl 8, falls def.
45	555	570	16	DP_Adr._9	DP-Adr. für Schalt- bzw. Stellbefehl 9 oder LEER
46	572	585	14	NOM_9	"nominal" für Schalt- bzw. Stellbefehl 9, falls def.
47	587	602	16	DP_Adr._10	DP-Adr. für Schalt- bzw. Stellbefehl 10 oder LEER
48	604	617	14	NOM_10	"nominal" für Schalt- bzw. Stellbefehl 10, falls def.
49	619	634	16	DP_Adr._11	DP-Adr. für Schalt- bzw. Stellbefehl 11 oder LEER
50	636	649	14	NOM_11	"nominal" für Schalt- bzw. Stellbefehl 11, falls def.
51	651	666	16	DP_Adr._12	DP-Adr. für Schalt- bzw. Stellbefehl 12 oder LEER
52	668	681	14	NOM_12	"nominal" für Schalt- bzw. Stellbefehl 12, falls def.
53	683	698	16	OFFEN-DP	Adresse eines Melde- oder Schaltungspunktes für die OFFEN-Meldung des ZK-Stellglieds oder LEER. Ist keine Adresse angegeben, so wird keine OFFEN-Überwachung durchgeführt *** momentan nicht genutzt ***
54	700	702	3	OFFEN-Wert	Nur falls der "actual" des OFFEN-DP mit diesem Wert übereinstimmt wird der Zutritt gewährt *** momentan nicht genutzt ***
55	704	719	16	DP_Adr._OFFEN	DP-Adr. für Schalt- bzw. Stellbefehl bei OFFEN-Alarm oder LEER *** momentan nicht genutzt ***
56	721	734	14	NOM_OFFEN	"nominal" für Schalt- bzw. Stellbefehl bei OFFEN-Alarm, falls def. *** momentan nicht genutzt ***
57	736	751	16	DP_Adr._AUFENTH	DP-Adr. für Schalt- bzw. Stellbefehl bei AUFENTHALTS-Alarm oder LEER *** momentan nicht genutzt ***



58        753        766        14        NOM\_AUFENTH        "nominal" für Schalt- bzw. Stellbefehl bei AUFENTHALTS-Alarm, falls def.  
 \*\*\* momentan nicht genutzt \*\*\*

**Bemerkung:** Die Freigabe-Datenpunkte (inkl. -Wert) werden bei der Generierung der Dateien "xxx.IM" nacheinander gemäß ihrem Auftreten in den Verknüpfungs-Strukturen für RZ, ZZ und IM besetzt und sind in aufsteigender Reihenfolge zu belegen. Analog wird mit den Schalt- bzw. Stellbefehlen 1-12 verfahren.

Wurde "IME-Kenn\_RESET" angegeben, so erfolgt das Rücksetzen der Kontrollfunktionen, nachdem der Zutritt für das IM an dieser IME gewährt wurde. Die IM-Datei muss also einen entsprechenden Eintrag für "IME-Kenn\_RESET" und die Betriebsart ZKS enthalten.

**Hinweis: Der Benutzer trägt beim Editieren der Dateien die Verantwortung dafür, dass das Format, die Syntax und die Parameter korrekt sind. Die Verwendung von Tabulatoren ist zur Spaltenpositionierung NICHT zulässig !!**

Die gleichnamige Text-Datei mit der Extension .IMS vermerkt den IM-Status, falls Betriebsart ZKS. In der ERSTEN Zeile steht der Eintrag für die Raumzone, in der der IM-Träger anwesend ist (inkl. Aufenthaltsdauerüberwachung). Alle folgenden Zeilen enthalten die Einträge, die für die Zutrittswiederholkontrolle noch vermerkt werden müssen.

Die IMS-Datei hat folgendes Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	10	10	RZ_Kenn	Kennung der Raumzone, in der sich der IM-Träger aufhält bzw. aufgehalten hat
2	12	25	14	ZutrittsWdh_AB	Datum/Zeit ab der der erneute Zutritt zur Raumzone erlaubt ist (JJJMMTTHHMSS) * momentan nicht genutzt *
3	27	40	14	Aufenthalt_BIS	Datum/Zeit bis zu der der IM-Träger in der Raumzone aufhalten darf (JJJMMTTHHMSS) * momentan nicht genutzt *
4	42	57	16	DP_Adr_AUFENTH	DP-Adr. für Schalt- bzw. Stellbefehl bei AUFENTHALTS-Alarm oder LEER * momentan nicht genutzt *
5	59	72	14	NOM_AUFENTH	"nominal" für Schalt- bzw. Stellbefehl bei AUFENTHALTS-Alarm, falls def. * momentan nicht genutzt *
6	74	89	16	IME_Reset	DP-Adr. für IME, die ein Rücksetzen der Kontrollfunktionen durchführt. Ist IME_Reset definiert, so erfolgt "harte" Kontrolle, d.h. der Zutritt wird nur gewährt, wenn Raumzonenwechsel (RZ_Kenn == RZ_Kenn_GEHT    IME_Reset == IME_Kenn_KOMMT) Falls nicht def. ("weiche" Kontrolle), führt jeder ge-

währte Zutritt zum Rücksetzen

--- Erweiterter Bereich ab V1.3i (Projekt Feuerwachen) -----  
 (wird aus Kompatibilitätsgründen zu den bestehenden Systemen in den  
 Objekten FRKA, BERS, ... angefügt.

7	91	94	4	PIN_Last	Letzte eingegebene PIN
8	96	109	14	PIN_Gueltig	Datum/Zeit bis zu dem PIN_Last gültig ist (JJJJMMTHHMMSS)
9	111	113	3	PIN_Wdh	Anzahl der noch verbleibenden Wiederholungsversuche für die PIN-Eingabe

**Hinweis:** Die IMS-Dateien werden beim Programmstart gelöscht und beim ersten ZKS-Zutritt initialisiert. Ist die zugehörige IMS-Datei nicht vorhanden, so erfolgt die Prüfung nur anhand der IMS-Datei.

Bei Aufruf des Hilfsprogrammes CLR\_IMS dient im Arbeitsverzeichnis die Datei CLR\_IMS.LOG der Protokollierung der gelöschten IMS-Dateien

**5.5.10.1. Fehlerbehandlung**

Beim Ablauf des elektronischen Schlüssellersatzes können Fehler auftreten, die vom ES-Modul durch die in der Datei "ES\_INCL.H" beschriebenen Kodierungen (ES\_ERR\_...) gekennzeichnet werden.

Ein beim Start des GA-Knotens erkannter Fehler führt zum Abbruch des Programmstarts und einer zugehörigen Fehlermeldung in der Error-Log-Datei (vgl. EL-Kommando in der Konfigurationsdatei GAK.CFG).

Wird der Fehler bei der Übernahme mittels eines Karteipunktes von der LZH erkannt, so sendet der GA-Knoten ein ERROR-Datagramm an die LZH und bricht die Übernahme ab.

Wird ein Fehler zur Laufzeit erkannt, so versucht das ES-Modul den Ablauf aufrecht zu erhalten, und schreibt eine Fehlermeldung in die Error-Log-Datei.

### **5.5.11. Definition der Datenübertragung**

Für die Kommunikation zwischen ES-Modul und den IZ wird jeder IME ein eigener FND-Transferpunkt für den transparenten Datentransfer (dp\_type = 0, dp\_subtype = 0) mit dem Transfer-Typ 0 zugeordnet (vgl. ES-Zeile für Transferpunkt in GAK.CFG). Der FND-Datenteil (ASDU) für DP\_#1 hat folgenden Aufbau :

Byte 0:	Typ des IME-Datagrammes
Byte 1 ff.:	Datenteil des IME-Datagrammes abhängig vom Typ

Für vom LON-Knoten spontan gesendete FND-Datagramme (USM) ist der Typ wie folgt hexadezimal kodiert :

0x10	Gelesenes IM zur Prüfung (IM im Datenteil)
0x20	allg. IME-Fehler (Datenteil leer)
0x21	IME-Störung (Datenteil leer)
0x22	IME-Alarm, inkl. Sabotage (Datenteil leer)

Für vom GA-Knoten an den LON-Knoten gesendete FND-Datagramme (CMD; modifizierender Zugriff auf DP\_#1) ist der Typ wie folgt hexadezimal kodiert :

0x00	Zutritt für gelesenes IM abweisen (NAK; Datenteil leer)
0x01	Zutritt für gelesenes IM akzeptieren (ACK; Datenteil leer)
0x10	IME außer Tätigkeit setzen (Datenteil leer)
0x11	IME in Tätigkeit setzen (Datenteil leer)
0x1F	IME rücksetzen (Datenteil leer)

Auf ein modifizierendes CMD antwortet der LON-Knoten mit einem FND-Datagramm (RSP), wobei der Typ und Datenteil aus den CMD übernommen werden.

Lesende Zugriffe auf den Transferpunkt werden momentan nicht genutzt, bleiben aber reserviert für zukünftigen Erweiterungen. Ggf. ist eine Fehlerbehandlung mittels ERROR-Datagramm durchzuführen.

## **5.6. Schaltprogramme (SP)**

**SP\_LIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit der Bearbeitung der Schaltprogramme zusammenhängen (Schaltprogramm-Parameter, Schaltprogramm-Kalender, Zeit-Schaltprogramme, Ereignis-Schaltprogramme und zugehörige Karteipunkte).

### **5.6.1. Übersicht der verwendeten Module**

SP	Schaltprogramme
UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### **5.6.2. Liste mit Namen und Aufgaben**

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

SP_INIT	Initialisierung beim Programmstart
SP_BS_INIT	Initialisierung des Status-Bildschirm für die Schaltprogramme beim Programmstart
SP_CMD	Behandlung CMD-Datagramm von LZH (Karteipunkt)
SP_RSP_USM	Behandlung RSP-/USM-Datagramm von IZ's
SP_POLL	zyklische Bearbeitung der aktiven Schaltprogramme
SP_EXIT	Ende-Behandlung beim Programmstop
SP_IZQ_CMD	Ausgabe eines oder mehrerer CMD-Datagramme an die IZ's

Folgende Routinen werden nur innerhalb dieses Modules verwendet:

SP_SPP_INIT	Initialisierung der Schaltprogramm-Parameter beim Programmstart
SP_SPP_AKTIV	Aktivieren eines Schaltprogrammes

SP_SPP_CMD	Behandlung CMD-Datagramm von LZH für Schaltprogramm-Parameter
SP_SPP_WRSTATUS	Vermerken der aktuellen Status für ein Schaltprogramm
SP_KAL_INIT	Initialisierung des Schaltprogramm-Kalenders beim Programmstart
SP_KAL_AKTIV	Aktivieren eines Eintrags aus dem Schaltprogramm-Kalender
SP_KAL_CMD	Behandlung CMD-Datagramm von LZH für Schaltprogramm-Kalender
SP_ZSP_INIT	Initialisierung der Zeit-Schaltprogramme beim Programmstart
SP_ZSP_START	Starten eines Zeit-Schaltprogrammes
SP_ZSP_STOP	Stoppen eines Zeit-Schaltprogrammes
SP_ZSP_AKTIV	Aktivieren des nächsten auszuführenden Zeit-Schaltbefehl
SP_ZSP_CMD	Behandlung CMD-Datagramm von LZH für Zeit-Schaltprogramm
SP_ESP_INIT	Initialisierung der Ereignis-Schaltprogramme beim Programmstart
SP_ESP_START	Starten eines Zeit-Schaltprogrammes
SP_ESP_STOP	Stoppen eines Zeit-Schaltprogrammes
SP_ESP_RETURN	Rücksprung in das aufrufende Ereignis-Schaltprogramm
SP_ESP_AKTIV	Aktivieren des nächsten auszuführenden Befehls im Ereignis-Schaltprogramm
SP_ESP_CMD	Behandlung CMD-Datagramm von LZH für Zeit-Schaltprogramm
SP_ESP_RSP_USM	Behandlung RSP-/USM-Datagramm von IZ's für Ereignis-Schaltprogramme
SP_ESP_PUSH	Zwischenergebnis auf Keller schreiben
SP_ESP_POP	Zwischenergebnis vom Keller lesen
SP_STC_WRITE	Keller auf STC-Datei schreiben
SP_ESP2ADE	Konvertierung Ereignis-Schaltprogramm in interne Arbeitsdatei
SP_ADE_SCAN	Abarbeitung der internen Arbeitsdatei
SP_DP_SCAN	aktueller Wert des Datenfeldes eines Datenpunktes
SP_SV_SCAN	aktueller Wert einer System-Variablen
SP_IZQ_RSP	Bearbeitung der von den IZ's empfangenen RSP-Datagramme

SP_IZQ_POLL	zyklische Bearbeitung der ausgegebenen CMD-Datagramme
SP_IZQ_SORT	Sortierung der ausgegebenen CMD-Datagramme

### 5.6.3. Aufrufstruktur

Wegen der Komplexität der Aufrufstruktur werden Teilbäume nur erstmalig dargestellt und bei wiederholter Verwendung auf diesen verwiesen. Die Wurzel des Teilbaumes ist durch Fettschrift und Unterstreichung hervorgehoben.

SP\_INIT

SP\_SPP\_INIT

#### SP SPP AKTIV

##### SP ZSP START

SP\_SPP\_WRSTATUS

##### SP ZSP AKTIV

##### SP ZSP STOP

SP\_SPP\_WRSTATUS

SP\_ESP\_RETURN

SP\_IZQ\_CMD

##### SP ESP START

SP\_SPP\_WRSTATUS

SP\_ZSP\_STOP (vgl. oben)

##### SP ESP STOP

SP\_SPP\_WRSTATUS

SP\_ESP\_RETURN

SP\_SPP\_WRSTATUS

SP\_KAL\_INIT

#### SP KAL AKTIV

SP\_ZSP\_START (vgl. oben)

SP\_ESP\_START (vgl. oben)

SP\_ZSP\_STOP (vgl. oben)

SP\_ESP\_STOP (vgl. oben)

SP\_SPP\_WRSTATUS

SP\_ZSP\_INIT

SP\_ZSP\_START (vgl. oben)

SP\_ESP\_INIT

### **SP ESP2ADE**

#### **SP ADE\_SCAN**

SP\_DP\_SCAN

SP\_IZQ\_CMD

#### **SP ESP PUSH**

SP\_STC\_WRITE

#### **SP ESP POP**

SP\_STC\_WRITE

SP\_SV\_SCAN

SP\_ESP\_PUSH (vgl. oben)

SP\_ZSP\_START (vgl. oben)

SP\_ESP\_START (vgl. oben)

SP\_ZSP\_STOP (vgl. oben)

SP\_ESP\_STOP (vgl. oben)

SP\_ESP\_PUSH (vgl. oben)

SP\_ESP\_POP (vgl. oben)

SP\_ESP\_PUSH (vgl. oben)

SP\_ESP\_POP (vgl. oben)

SP\_ESP\_START (vgl. oben)

SP\_BS\_INIT

SP\_ZSP\_AKTIV (vgl. oben)

SP\_CMD

SP\_ESP2ADE (vgl. oben)

SP\_SPP\_CMD

SP\_KAL\_CMD

SP\_ZSP\_CMD

SP\_ESP\_CMD

SP\_ESP2ADE (vgl. oben)

SP\_RSP\_USM

SP\_IZQ\_RSP

SP\_IZQ\_SORT

SP\_ESP\_RSP\_USM

SP\_ADE\_SCAN (vgl. oben)

SP\_POLL

SP\_SPP\_AKTIV (vgl. oben)

SP\_KAL\_AKTIV (vgl. oben)

SP\_ZSP\_AKTIV (vgl. oben)

SP\_ESP\_AKTIV

SP\_ESP2ADE (vgl. oben)

SP\_IZQ\_POLL

SP\_IZQ\_SORT

SP\_EXIT



5.6.4. Modulbeschreibungen

Die nachfolgend beschriebenen Routinen verwenden die einheitlichen Fehlerkodierungen SP\_ERR\_... wie folgt:

```

/*      -- Fehler - Codes des SP-Moduls      */
# define SP_NORMAL          0      /* KEIN Fehler      */

/*      --- Allgemeine Fehler      */
# define SP_ERR_OPEN_FILE   -1     /* OPEN-Fehler auf Datei      */
# define SP_ERR_READ_FILE   -2     /* READ-Fehler auf Datei      */
# define SP_ERR_WRITE_FILE  -3     /* WRITE-Fehler auf Datei     */
# define SP_ERR_END_OF_FILE  -4     /* Dateiende erreicht        */
# define SP_ERR_ALLOC_MEM   -5     /* ALLOC-Fehler auf Speicher  */

# define SP_ERR_FORMAT      -6     /* Allgemeiner Fehler in der  */
/* Syntax bzw. dem Format      */
/* einer Eingabe, Datei oder  */
/* Aufrufparameter           */

# define SP_ERR_DP         -7     /* Unzulässiger Datenpunkt bzw. */
/* zugehöriges Datenfeld (inkl. */
/* DP ist nicht dem Objekt des */
/* Schalprogrammes zugeordnet) */

# define SP_ERR_WERT       -8     /* Unzulässiger Wert für den   */
/* DP-Befehl bzw. die ESP-Opera- */
/* tion / -Funktion           */

/*      --- Fehler beim Aktivieren, Starten und Stoppen      */
# define SP_ERR_SP         -10     /* Allgemeiner SP-Fehler, z.B. */
/* unzulässige Schaltprogramm- */
/* Kennung                    */
# define SP_ERR_AKTIV      -11     /* Fehler beim Aktivieren eines */
/* Eintrags in der SPP-, KAL-, */
/* ZSP- oder ESP-Struktur     */
# define SP_ERR_START      -12     /* Fehler beim Starten eines    */
/* Schaltprogrammes          */
# define SP_ERR_STOP       -13     /* Fehler beim Stoppen eines   */
/* Schaltprogrammes          */

/*      --- Fehler bei der SYNTAX-Prüfung eines ESP      */
# define SP_ERR_ESP_SYNTAX -20     /* Allgemeiner Fehler in der  */
/* Syntax eines ESP            */
# define SP_ERR_ESP_DTYP   -21     /* Unzulässiger Datentyp für die */
/* aufgerufene Operation bzw. */
/* Funktion                   */

# define SP_ERR_ESP_KL_LOW  -22     /* schließende Klammer zuviel  */
# define SP_ERR_ESP_KL_HIGH -23     /* zuviele öffnende Klammern.  */
/* Max. 1 Ebene innerhalb einer */
/* Befehlsfolge                */

# define SP_ERR_ESP_BF_LOW  -24     /* Befehlsfolgen-Ebene NIEDRIG, */
/* d.h. Semikolon zuviel        */
# define SP_ERR_ESP_BF_HIGH -25     /* Befehlsfolgen-Ebene HOCH,   */
/* d.h. Semikolon fehlt        */

# define SP_ERR_ESP_STACK_LOW -30    /* Stack NIEDRIG (SP_ESP_POP)  */
# define SP_ERR_ESP_STACK_HIGH -31   /* Stack-Ebene zu HOCH, d.h. am */
/* Ende einer Befehlsfolge ist  */
/* der Stack nicht ausgeglichen */
/* d.h. bis auf das Ergebnis auf */
/* Ebene 1 entleert.           */
# define SP_ERR_ESP_STACK_TYP -32    /* Unzulässiger Typ des von POP */
/* gelieferten Wertes bzw. bei  */
/* PUSH angegeben.            */

/*      --- Fehler beim Ablauf eines ESP      */

```

```
# define SP_ERR_ESP_RUN          -50    /* Allgemeiner Fehler bei der */
/* Ausführung eines ESP        */
# define SP_ERR_ESP_CALLED      -51    /* Fehler bei CALL eines Schalt-*/
/* programmes, d.h. das zu    */
/* startende SP wurde bereits */
/* durch ein anders SP per CALL*/
/* gestartet.                 */

/*      --- FND-Fehler für gesendeten Befehl empfangen      */
# define SP_ERR_FND_ERROR       -90    /* FND-ERROR empfangen      */
# define SP_ERR_FND_REJECT      -91    /* FND-REJECT empfangen     */
```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : INIT.C */
/* Aufruf : SP_INIT ( ) */
/*****
/* Beschreibung : */
/* Die Routine SP_INIT muß als erste Routine für die */
/* Schaltprogramme aufgerufen werden. Sie aktiviert */
/* das Modul und führt alle nötigen Initialisierungen */
/* durch. */
/* Hinweis: Die Reihenfolge der Aufrufe für die INIT_xxx- */
/* Routinen darf nicht veraendert werden. !! */
/* */
/* aufgerufen von : GAK_INIT */
/* ruft auf : SPP_INIT, KAL_INIT, ZSP_INIT, ESP_INIT */
/* I/O-Kanäle : */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_INIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : BS_INIT.C */
/* Aufruf : SP_BS_INIT ( ) */
/*****
/* Beschreibung : */
/* Diese Routine initialisiert die Zeilen im Statusbildschirm mit */
/* dem nächsten auszuführenden Befehl, wenn ein solcher existiert. */
/* */
/* aufgerufen von : GAK_INIT */
/* */
/* ruft auf : SP_ZSP_AKTIV */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_BS_INIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*   Datei    :   CMD.C                 */
/*   Aufruf   :   SP_CMD ( cCMD )       */
/*****
/*   Beschreibung :                   */
/*   Diese Routine vermerkt das in "cCMD" gelieferte Schaltprogramm */
/*   bzw. den Eintrag für den Schaltprogramm-Kalender in einer     */
/*   zugehörigen Datei (Segmentierter Karteipunkt gemäß LV unter  */
/*   4.4 ff.). Ist der Aktivierungszeitpunkt früher als der bisher */
/*   als nächster vermerkte, so wird der neue Zeitpunkt als nächster */
/*   auszuführender vermerkt.                                     */
/*   Als Ergebnis wird der Zeiger auf das für "cCMD" zu liefernde  */
/*   RSP-Datagramm geliefert.                                     */
/*   */
/*   aufgerufen von :   GAK_ILZ        */
/*   */
/*   ruft auf          :   SP_ESP2ADE, SP_SPP_CMD, SP_KAL_CMD,    */
/*                       SP_ZSP_CMD, SP_ESP_CMD,                */
/*                       UT_OBJ_GETI, UT_FNDERROR, UT_FNDREJECT, */
/*                       diverse Makros FND_... und SET_FND_...  */
/*   */
/*   I/O-Kanäle       :                   */
/*   */
/*****
/*   Eingabe Parameter :                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   char*    cCMD          *      Zeiger auf das auszuwertende          */
/*                                               CMD-Datagramm mit dem neuen          */
/*                                               Schaltprogramm bzw. Kalender-          */
/*                                               Eintrag                               */
/*   -----                               */
/*   Ausgabe Parameter:                   */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   char*    SP_CMD          *      Zeiger auf das RSP-Datagramm        */
/*   -----                               */
/*   Transiente Parameter :               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                               */
/* Original :   W. Fries    01.05.97    1.1          */
/* Änderung:                                     */
/*   -----                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*  Datei    :   RSP_USM.C              */
/*  Aufruf   :   SP_RSP_USM ( cDatagramm ) */
/*****
/*  Beschreibung :                               */
/*  Die Routine führt die Bearbeitung der von den DDC-IZ's empfang. */
/*  RSP- und USM-Datagramme durch.           */
/*                                           */
/*  aufgerufen von :   GAK_IIZ             */
/*                                           */
/*  ruft auf      :   SP_IZQ_RSP, SP_ESP_RSP_USM, */
/*                  diverse Makros FND_...   */
/*                                           */
/*  I/O-Kanäle   :                               */
/*                                           */
/*****
/*  Eingabe Parameter :                               */
/*  -----                                           */
/*  Typ   Name                Dim.   Beschreibung           */
/*  char  cDatagramm          */
/*                  RSP- bzw. USM-Datagramm           */
/*  -----                                           */
/*  Ausgabe Parameter:                               */
/*  -----                                           */
/*  Typ   Name                Dim.   Beschreibung           */
/*  int   SP_RSP_USM          */
/*                  Returnkode der Funktion           */
/*                  0=FALSE: KEIN Fehler              */
/*  -----                                           */
/*  Transiente Parameter :                               */
/*  -----                                           */
/*  Typ   Name                Dim.   Beschreibung           */
/*  *** KEINE ***                                           */
/*****
/*  Autor      Datum      Version   Kurzzeichen   */
/*  -----                                           */
/*  Original :   W. Fries   01.05.97   1.1           */
/*  Änderung:                                           */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : POLL.C */
/* Aufruf : SP_POLL ( ) */
/*****
/* Beschreibung : */
/* Die Routine führt die zyklische Bearbeitung der Schaltprogramme */
/* durch und liefert ein an die IZ weiterzuleitendes CMD-Data- */
/* gramm als Ergebnis, wenn ein solches existiert. */
/* */
/* aufgerufen von : GAK_ISP */
/* */
/* ruft auf : SP_SPP_AKTIV, SP_KAL_AKTIV, SP_ZSP_AKTIV, */
/* SP_ESP_AKTIV, SP_IZQ_POLL */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* char* SP_POLL Zeiger auf das an IZ auszugeben- */
/* de CMD-Datagramm */
/* SONST: NULL-Zeiger */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : EXIT.C */
/* Aufruf : SP_EXIT ( ) */
/*****
/* Beschreibung : */
/* Die Routine muß als letzte Routine für die */
/* Schaltprogramme aufgerufen werden. Sie aktiviert */
/* das Modul und führt alle nötigen ENDE-Behandlungen durch. */
/* */
/* Hinweis: Die Reihenfolge der Aufrufe für die EXIT_xxx- */
/* Routinen darf nicht veraendert werden. !! */
/* */
/* aufgerufen von : GAK_EXIT */
/* ruft auf : SY_FREE */
/* I/O-Kanäle : */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_EXIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand     :   01.10.98  */
/*****
/*  Datei     :   SPP_INIT.C           */
/*  Aufruf    :   SP_SPP_INIT ( )      */
/*****
/*  Beschreibung :                       */
/*  Diese Routine lädt die Schaltprogramm-Parameter aller hinter- */
/*  legter Programme in die dynamische PRG-Tabelle und aktiviert */
/*  sie, wenn der Zeitpunkt erreicht ist.                          */
/*                                                                    */
/*  aufgerufen von :   SP_INIT                                                */
/*                                                                    */
/*  ruft auf      :   SP_SPP_AKTIV, UT_CI2DT, UT_NORM_STRING,              */
/*                  UT_OBJ_GETI, SY_CALLOC                                  */
/*                                                                    */
/*  I/O-Kanäle    :                                                           */
/*                                                                    */
/*****
/*  Eingabe Parameter :                                                       */
/*  -----                                                       */
/*  Typ      Name                Dim.   Beschreibung                       */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*  *** KEINE ***                                                       */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*  Ausgabe Parameter:                                                       */
/*  -----                                                       */
/*  Typ      Name                Dim.   Beschreibung                       */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*  int      SP_SPP_INIT          0      Returnkode der Funktion          */
/*                  0=FALSE: KEIN Fehler                                  */
/*                  SONST: vgl. SP_ERR_...                               */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*  Transiente Parameter :                                                       */
/*  -----                                                       */
/*  Typ      Name                Dim.   Beschreibung                       */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*  *** KEINE ***                                                       */
/*****
/*                  Autor      Datum      Version      Kurzzeichen      */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*  Original :   W. Fries      01.05.97      1.1              */
/*  Änderung:                                          */
/*  -----  -----  -----  -----  -----  -----  -----  -----  */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770  */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : SPP_AKTI.C */
/* Aufruf : SP_SPP_AKTIV ( iModus ) */
/*****
/* Beschreibung : */
/* Die Routine aktiviert das nächste Schaltprogramm. */
/* Ist eine ältere Version des Schaltprogrammes bereits gestartet, */
/* so wird diese zuerst gestoppt und die neue Version gestartet. */
/* */
/* aufgerufen von : SP_SPP_INIT, SP_POLL */
/* */
/* ruft auf : SP_ZSP_START, SP_ESP_START, SP_ZSP_STOP, */
/* SP_ESP_STOP, SP_SPP_WRSTATUS, */
/* UT_DR_OPROT, UT_NORM_STRING */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int iModus Modus für die Aktivierung */
/* 0: INIT */
/* 1: POLL */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_SPP_AKTIV Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* ----- */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   SP  Schalt-Programme  Stand    :   01.10.98  */
/*****
/*   Datei    :   SPP_CMD.C             */
/*   Aufruf   :   SP_SPP_CMD ( OBJ, cSatz, SPP ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine vermerkt den in "cSatz" gelieferten Satz für ein */
/*   Schaltprogramm inkl. der zugehörigen Parameter für das Objekt */
/*   "OBJ" in der zugehörigen Konfigurationsdatei und der SPP- */
/*   Struktur (Segmentierter Karteipunkt gemäß LV unter 4.4.3). */
/*   Ist der Aktivierungszeitpunkt früher als der bisher als nächster */
/*   vermerkte, so wird der neue Zeitpunkt als nächster auszu- */
/*   führender vermerkt. */
/*   */
/*   aufgerufen von :   SP_CMD */
/*   */
/*   ruft auf      :   UT_TAG_S2DT, UT_NORM_STRING, UT_DT2CI, */
/*                   SY_CALLOC, SY_IEEE2F */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   UT_OBJ *   OBJ      */      Zeiger auf den Eintrag in der */
/*   */      OBJ-Struktur, d.h. das Objekt, */
/*   */      dem das Schaltprogramm zugeord- */
/*   */      net ist. */
/*   */
/*   char*     cSatz     */      Zeiger auf den auszuwertenden */
/*   */      Karteipunktsatz bestehend aus */
/*   */      Tabellen-, Zeilen- und Spalten- */
/*   */      Index und der zugehörigen Zelle.*/
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   int      SP_SPP_CMD */      Returnkode der Funktion */
/*   */      0=FALSE: KEIN Fehler */
/*   */
/*   SP_SPP**  SPP      */      Zeiger auf den Eintrag in der */
/*   */      SPP-Struktur unter dem das neue */
/*   */      Schaltprogramm verwaltet wird. */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1      */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : SPP_WRST.C */
/* Aufruf : SP_SPP_WRSTATUS ( SPP, fp_PRG ) */
/*****
/* Beschreibung : */
/* Die Routine vermerkt den Status des ZSP und ESP für das durch */
/* SPP gekennzeichnete Schaltprogramm. Ist die PRG-Datei bereits */
/* geöffnet, so wird "fp_PRG" verwendet. */
/* */
/* aufgerufen von : SP_SPP_AKTIV, SP_KAL_AKTIV, SP_ZSP_START, */
/* SP_ESP_START, SP_ZSP_STOP, SP_ESP_STOP */
/* */
/* ruft auf : */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_SPP* SPP Zeiger auf die zugehörigen SPP */
/* FILE* fp_PRG FILE-Zeiger auf geöffnete PRG- */
/* Datei; SONST : NULL */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_SPP_WRSTATUS Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : KAL_INIT.C */
/* Aufruf : SP_KAL_INIT ( ) */
/*****
/* Beschreibung : */
/* Diese Routine lädt den nächsten auszuführenden Eintrag im */
/* Schaltprogramm-Kalender. Einträge, für die kein Schaltprogramm */
/* existiert, werden gelöscht. */
/* */
/* Hinweis: Die Überprüfung der Objekt- und Programm-Kennung */
/* erfolgt erst bei der Ausführung, da diese abhängig */
/* von der Aktivierung der zugehörigen Schaltprogramm- */
/* Version ist. */
/* */
/* Hinweis: Für den Schaltprogramm-Kalender wird KEINE dynamische */
/* Struktur im Hauptspeicher aufgebaut. */
/* */
/* */
/* aufgerufen von : SP_INIT */
/* */
/* ruft auf : SP_KAL_AKTIV, UT_CI2DT */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_KAL_INIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : KAL_AKTI.C */
/* Aufruf : SP_KAL_AKTIV ( iModus ) */
/*****
/* Beschreibung : */
/* Die Routine startet bzw. stoppt das Schaltprogramm, das im */
/* nächsten Eintrag im Schaltprogramm-Kalender angegeben ist. */
/* Anschließend wird der nächste auszuführende Eintrag vermerkt. */
/* */
/* aufgerufen von : SP_KAL_INIT, SP_POLL */
/* */
/* ruft auf : SP_ZSP_START, SP_ESP_START, SP_ZSP_STOP, */
/* SP_ESP_STOP, SP_SPP_WRSTATUS, */
/* UT_NORM_STRING, UT_DR_OPROT, UT_CI2DT */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int iModus Modus für die Aktivierung */
/* 0: INIT */
/* 1: POLL */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_KAL_AKTIV Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*   Datei    :   KAL_CMD.C              */
/*   Aufruf   :   SP_KAL_CMD ( OBJ, cSatz ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine vermerkt den in "cSatz" gelieferten Satz für den */
/*   Schaltprogramm-Kalender für das Objekt "OBJ" und das im Satz */
/*   spezifizierte Schaltprogramm in der zugehörigen Konfigurations- */
/*   datei (Segmentierter Karteipunkt gemäß LV unter 4.4.9).      */
/*   Ist der Aktivierungszeitpunkt früher als der bisher als nächster */
/*   vermerkte, so wird der neue Zeitpunkt als nächster auszu-   */
/*   führender vermerkt.                                         */
/*                                                                 */
/*   aufgerufen von :   SP_CMD                                         */
/*                                                                 */
/*   ruft auf      :   UT_NORM_STRING, UT_TAG_S2DT, UT_DT2CI,      */
/*                   SY_IEEE2F                                         */
/*                                                                 */
/*   I/O-Kanäle   :                               */
/*                                                                 */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   -----                               */
/*   char*    cSatz         *       Zeiger auf den auszuwertenden */
/*                                     Karteipunktsatz bestehend aus */
/*                                     Tabellen-, Zeilen- und Spalten- */
/*                                     Index und der zugehörigen Zelle.*/
/*   -----                               */
/*   - - - - -                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   -----                               */
/*   int      SP_KAL_CMD     *       Returnkode der Funktion         */
/*                                     0=FALSE: KEIN Fehler          */
/*   -----                               */
/*   - - - - -                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung                               */
/*   -----                               */
/*   *** KEINE ***                                         */
/*****
/*                                     Autor      Datum      Version      Kurzzeichen */
/*   -----                               */
/* Original :   W. Fries      01.05.97      1.1                               */
/* Änderung:                                         */
/*                                                                 */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*  Datei    :   ZSP_INIT.C             */
/*  Aufruf   :   SP_ZSP_INIT ( )       */
/*****
/*  Beschreibung :                       */
/*  Diese Routine lädt die gestarteten Zeitschaltprogramme. */
/*  aufgerufen von :   SP_INIT         */
/*  ruft auf      :   SP_ZSP_START, UT_LOOPTIME, SY_CALLOC */
/*  I/O-Kanäle   :                       */
/*****
/*  Eingabe Parameter :                       */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung */
/*  *** KEINE *** */
/*  ----- */
/*  Ausgabe Parameter:                       */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung */
/*  int     SP_ZSP_INIT          Returnkode der Funktion */
/*                                     0=FALSE: KEIN Fehler */
/*                                     SONST: vgl. SP_ERR_... */
/*  ----- */
/*  Transiente Parameter :                       */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung */
/*  *** KEINE *** */
/*****
/*          Autor      Datum      Version      Kurzzeichen */
/*  ----- */
/* Original :   W. Fries      01.05.97      1.1 */
/* Änderung: */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ZSP_STAR.C */
/* Aufruf : SP_ZSP_START ( SPP, iModus ) */
/*****
/* Beschreibung : */
/* Die Routine startet das Zeitschaltprogramm für "SPP". Abhängig */
/* von "iModus" wird für die Initialisierung und das Polling */
/* unterschieden. */
/* */
/* aufgerufen von : SP_SPP_AKTIV, SP_KAL_AKTIV, SP_ZSP_INIT */
/* SP_ADE_SCAN */
/* */
/* ruft auf : SP_SPP_WRSTATUS, SP_ZSP_AKTIV, */
/* SY_CALLOC */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_SPP* SPP Zeiger auf den Eintrag in der */
/* SPP-Struktur für den das ZSP */
/* gestartet werden soll. */
/* */
/* int iModus Modus für die Aktivierung */
/* 0: INIT */
/* 1: POLL */
/* Hinweis: Momentan NICHT verwen-*/
/* det. Bleibt reserviert*/
/* für zukünftige Er- */
/* weiterungen */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ZSP_START Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ZSP_STOP.C */
/* Aufruf : SP_ZSP_STOP ( SPP, iModus ) */
/*****
/* Beschreibung : */
/* Die Routine stoppt das Zeitschaltprogramm für "SPP". Abhängig */
/* von "iModus" wird für die Initialisierung und das Polling */
/* unterschieden. */
/* */
/* aufgerufen von : SP_SPP_AKTIV, SP_KAL_AKTIV, SP_ZSP_AKTIV, */
/* SP_ADE_SCAN */
/* */
/* ruft auf : SP_SPP_WRSTATUS, SP_ESP_RETURN, */
/* UT_DR_OPROT, UT_NORM_STRING */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_SPP* SPP Zeiger auf den Eintrag in der */
/* SPP-Struktur für den das ZSP */
/* gestoppt werden soll. */
/* int iModus Modus für die Aktivierung */
/* 0: INIT / KAL */
/* 1: POLL */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ZSP_STOP Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ZSP_AKTI.C */
/* Aufruf : SP_ZSP_AKTIV ( iModus ) */
/*****
/* Beschreibung : */
/* Diese Routine aktiviert den nächsten auszuführenden Eintrag in */
/* der ZSP-Struktur (nächster Zeit-Schaltbefehl). */
/* Hinweis: Ein SYNTAKTISCHER Fehler im ZSP wird zwar protokol- */
/* liert, führt jedoch nicht zum Abbruch des Zeit- */
/* Schaltprogrammes. */
/* aufgerufen von : SP_POLL, SP_ZSP_START, SP_BS_INIT */
/* ruft auf : SP_ZSP_STOP, SP_IZQ_CMD, */
/* UT_DP_FINDFIRST, UT_DP_FINDNEXT, UT_NORM_STRING, */
/* UT_DRDPMELD, UT_FNDZB2DT, UT_FNDNX2DT, */
/* UT_BSDPMELD, BS_DF_INIT, BS_DF_OUTS */
/* I/O-Kanäle : */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* int iModus Modus für die Aktivierung */
/* 0: INIT / START */
/* 1: POLL */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* int SP_ZSP_AKTIV Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.95 1.1 */
/* Änderung: */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   SP  Schalt-Programme  Stand    :   01.10.98  */
/*****
/*   Datei     :   ZSP_CMD.C             */
/*   Aufruf    :   SP_ZSP_CMD ( SPP, cSatz ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine vermerkt den in "cSatz" gelieferten Satz für das */
/*   Zeit-Schaltprogramm des Schaltprogrammes "SPP" in der */
/*   zugehörigen ZSP-Datei (Segmentierter Karteipunkt gemäß LV */
/*   unter 4.4.5). */
/*   */
/*   aufgerufen von :   SP_CMD */
/*   */
/*   ruft auf      :   UT_NORM_STRING, SY_IEEE2F */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   */
/*   SP_SPP*  SPP      */      Zeiger auf den Eintrag in der */
/*   */      SPP-Struktur dem der Satz */
/*   */      zum ESP zugeordnet ist. */
/*   */
/*   char*    cSatz     */      Zeiger auf den auszuwertenden */
/*   */      Karteipunktsatz bestehend aus */
/*   */      Tabellen-, Zeilen- und Spalten- */
/*   */      Index und der zugehörigen Zelle.*/
/*   */
/*   - - - - - */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   */
/*   int      SP_ZSP_CMD      */      Returnkode der Funktion */
/*   */      0=FALSE: KEIN Fehler */
/*   */
/*   - - - - - */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1 */
/*   Änderung: */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ESP_INIT.C */
/* Aufruf : SP_ESP_INIT ( ) */
/*****
/* Beschreibung : */
/* Diese Routine initialisiert die Behandlung der Ereignis- */
/* schaltprogramme und startet die vom letzten Programmlauf */
/* gestarteten ESP, um den Zustand zu restaurieren. */
/* */
/* aufgerufen von : SP_INIT */
/* */
/* ruft auf : SP_ESP2ADE, SP_ESP_START, SY_CALLOC, SY_FI_DT */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ESP_INIT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR_... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* ----- */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*  Datei    :   ESP_STAR.C             */
/*  Aufruf   :   SP_ESP_START ( SPP, iModus ) */
/*****
/*  Beschreibung :                               */
/*  Die Routine startet das Ereignisschaltprogramm für "SPP". */
/*  Abhängig von "iModus" wird für die Initialisierung und das */
/*  Polling unterschieden.                               */
/*  aufgerufen von :   SP_SPP_AKTIV, SP_KAL_AKTIV, SP_ESP_INIT */
/*                   SP_ADE_SCAN */
/*  ruft auf      :   SP_SPP_WRSTATUS, SY_CALLOC */
/*  I/O-Kanäle   :                               */
/*****
/*  Eingabe Parameter :                               */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung                               */
/*  SP_SPP*  SPP           Zeiger auf den Eintrag in der SPP-Struktur für den das ESP */
/*                   gestartet werden soll.                               */
/*  int      iModus        Modus für die Aktivierung                               */
/*                   0: INIT                                           */
/*                   1: POLL                                           */
/*                   Hinweis: Momentan NICHT verwendet. Bleibt reserviert* */
/*                   für zukünftige Erweiterungen                               */
/*  ----- */
/*  Ausgabe Parameter:                               */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung                               */
/*  int      SP_ESP_START  Returnkode der Funktion                               */
/*                   0=FALSE: KEIN Fehler                               */
/*                   SONST: vgl. SP_ERR...                               */
/*  ----- */
/*  Transiente Parameter :                               */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung                               */
/*  *** KEINE *** */
/*****
/*  Autor      Datum      Version  Kurzzeichen */
/*  ----- */
/*  Original :   W. Fries   01.05.97   1.1 */
/*  Änderung:                               */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ESP_STOP.C */
/* Aufruf : SP_ESP_STOP ( SPP, iModus ) */
/*****
/* Beschreibung : */
/* Die Routine stoppt das Ereignisschaltprogramm für "SPP". */
/* Abhängig von "iModus" wird für die Initialisierung und das */
/* Polling unterschieden. */
/* */
/* aufgerufen von : SP_SPP_AKTIV, SP_KAL_AKTIV, SP_ADE_SCAN */
/* */
/* ruft auf : SP_SPP_WRSTATUS, SP_ESP_RETURN, */
/* UT_DR_OPROT, UT_NORM_STRING */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_SPP* SPP Zeiger auf den Eintrag in der */
/* SPP-Struktur für den das ESP */
/* gestoppt werden soll. */
/* int iModus Modus für die Aktivierung */
/* 0: INIT / KAL */
/* 1: POLL */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ESP_STOP Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand     :   01.10.98  */
/*****
/*  Datei     :   ESP_RETURN.C          */
/*  Aufruf    :   SP_ESP_RETURN ( SPP, iModus ) */
/*****
/*  Beschreibung :                               */
/*  Die Routine prüft nachdem das ZSP oder ESP für "SPP" gestoppt */
/*  wurde, ob ein anderes Schaltprogramm dieses Schaltprogramm */
/*  mittels SP_CALL gestartet hat und nun auf dessen Beendigung */
/*  wartet. Abhängig von "iModus" wird für die Initialisierung und */
/*  das Polling unterschieden. */
/*  */
/*  aufgerufen von :   SP_ZSP_STOP, SP_ESP_STOP */
/*  */
/*  ruft auf      :   */
/*  */
/*  I/O-Kanäle    :   */
/*  */
/*****
/*  Eingabe Parameter :                               */
/*  ----- */
/*  Typ      Name           Dim.   Beschreibung                               */
/*  SP_SPP*  SPP           Zeiger auf den Eintrag in der */
/*  SPP-Struktur für den ggf. das */
/*  aufrufende Schaltprogramm */
/*  den Wartezustand beenden soll. */
/*  int      iModus        Modus für die Aktivierung */
/*  0: INIT / KAL */
/*  1: POLL */
/*  ----- */
/*  Ausgabe Parameter:                               */
/*  ----- */
/*  Typ      Name           Dim.   Beschreibung                               */
/*  int      SP_ESP_RETURN  Returnkode der Funktion */
/*  0=FALSE: KEIN Fehler */
/*  SONST: vgl. SP_ERR_... */
/*  ----- */
/*  Transiente Parameter :                               */
/*  ----- */
/*  Typ      Name           Dim.   Beschreibung                               */
/*  *** KEINE *** */
/*****
/*  Autor      Datum      Version  Kurzzeichen */
/*  ----- */
/*  Original :   W. Fries   01.05.97   1.1 */
/*  Änderung:   */
/*  ----- */
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2      */
/*  Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*  Datei    :   ESP_AKTI.C             */
/*  Aufruf   :   SP_ESP_AKTIV ( iModus ) */
/*****
/*  Beschreibung :                       */
/*  Diese Routine aktiviert den nächsten auszuführenden Eintrag in */
/*  der ESP-Struktur (nächster ESP-Ausdruck nach Wartezeit).      */
/*  aufgerufen von :   SP_POLL          */
/*  ruft auf       :   SP_ADE_SCAN     */
/*  I/O-Kanäle    :                       */
/*****
/*  Eingabe Parameter :                       */
/*  ----- */
/*  Typ   Name           Dim.   Beschreibung          */
/*  int   iModus         */
/*                               Modus für die Aktivierung */
/*                               0: INIT / START          */
/*                               1: POLL                 */
/*                               Hinweis: Momentan NICHT verwen- */
/*                               det. Bleibt reserviert */
/*                               für zukünftige Er- */
/*                               weiterungen              */
/* ----- */
/*  Ausgabe Parameter:                       */
/*  ----- */
/*  Typ   Name           Dim.   Beschreibung          */
/*  int   SP_ESP_AKTIV   */
/*                               Returnkode der Funktion */
/*                               0=FALSE: KEIN Fehler    */
/*                               SONST: vgl. SP_ERR_...  */
/* ----- */
/*  Transiente Parameter :                   */
/*  ----- */
/*  Typ   Name           Dim.   Beschreibung          */
/*  *** KEINE *** */
/*****
/*  Autor      Datum      Version   Kurzzeichen   */
/*  ----- */
/*  Original :   W. Fries   01.05.97   1.1          */
/*  Änderung:                                     */
/* ----- */
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   SP  Schalt-Programme  Stand    :   01.10.98  */
/*****
/*   Datei     :   ESP_CMD.C             */
/*   Aufruf    :   SP_ESP_CMD ( SPP, cSatz ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine vermerkt den in "cSatz" gelieferten Satz für das */
/*   Ereignis-Schaltprogramm des Schaltprogrammes "SPP" in der */
/*   zugehörigen ESP-Datei und startet die Konvertierung in die */
/*   ADE-Datei (Segmentierter Karteipunkt gemäß LV unter 4.4.7). */
/*   */
/*   aufgerufen von :   SP_CMD           */
/*   */
/*   ruft auf       :   SP_ESP2ADE      */
/*   */
/*   I/O-Kanäle     :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung      */
/*   */
/*   SP_SPP*  SPP              Zeiger auf den Eintrag in der */
/*   */
/*   */
/*   */
/*   char*    cSatz          Zeiger auf den auszuwertenden */
/*   */
/*   */
/*   */
/*   */
/*   Index und der zugehörigen Zelle.*/
/*   */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung      */
/*   */
/*   int      SP_ESP_CMD          Returnkode der Funktion */
/*   */
/*   */
/*   0=FALSE: KEIN Fehler */
/*   */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung      */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version      Kurzzeichen      */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:                               */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP  Schalt-Programme  Stand    :   01.10.98  */
/*****
/*  Datei    :   ESP_RSP_.C           */
/*  Aufruf   :   SP_ESP_RSP_USM ( cAPDU ) */
/*****
/*  Beschreibung :   */
/*  Diese Routine prüft für alle ESP, die auf ein Datenpunkt- */
/*  abhängiges Ereignis warten, ob dieses eingetreten ist.   */
/*  */
/*  aufgerufen von :   SP_RSP_USM   */
/*  */
/*  ruft auf      :   SP_ADE_SCAN, UT_DP_FINDFIRST, UT_DP_FINDINDEX */
/*  */
/*  I/O-Kanäle   :   */
/*  */
/*****
/*  Eingabe Parameter :   */
/*  ----- */
/*  Typ      Name           Dim.   Beschreibung           */
/*  */
/*  char *   cAPDU           Zeiger auf das RSP-/USM-Datagr., */
/*  das geprüft werden soll. */
/*  */
/*  ----- */
/*  Ausgabe Parameter:   */
/*  ----- */
/*  Typ      Name           Dim.   Beschreibung           */
/*  */
/*  int      SP_ESP_RSP_USM   Returnkode der Funktion */
/*  0=FALSE: KEIN Fehler     */
/*  SONST: vgl. SP_ERR_...   */
/*  */
/*  ----- */
/*  Transiente Parameter :   */
/*  ----- */
/*  Typ      Name           Dim.   Beschreibung           */
/*  */
/*  *** KEINE ***           */
/*  */
/*****
/*  Autor      Datum      Version   Kurzzeichen   */
/*  ----- */
/*  Original :   W. Fries   01.05.97   1.1           */
/*  Änderung:   */
/*  */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ESP_PUSH.C */
/* Aufruf : SP_ESP_PUSH ( ESP, iTyp, cWert ) */
/*****
/* Beschreibung : */
/* Diese Routine fügt für das "ESP" den Eintrag vom Typ "iTyp" */
/* mit dem Inhalt "cWert" an den ESP-spezifischen Stack an. */
/* Dabei wird der Inhalt erst dann aus dem Puffer in die STC-Datei */
/* geschrieben, wenn der Puffer anderweitig benötigt wird. Da sehr */
/* häufig der letzte Eintrag sofort wieder gelesen (POP) wird, */
/* werden dadurch die Datei-Zugriffe minimiert. */
/* */
/* aufgerufen von : SP_ESP2ADE, SP_ADE_SCAN, SP_DP_SCAN, SP_SV_SCAN */
/* */
/* ruft auf : SP_STC_WRITE */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_ESP* ESP Zeiger auf den Eintrag in der */
/* ESP-Struktur, für den der Stack */
/* geführt wird. */
/* */
/* int iTyp Typ des Wertes in "cWert" */
/* ( vgl. unter "cWert" ) */
/* */
/* char* cWert Auf dem Stack zu speichernder */
/* Wert, generell als Zeichenfolge */
/* kodiert, d.h. muß nach POP mit */
/* "sscanf" abhängig von "iTyp" in */
/* den binären Wert gewandelt */
/* werden. */
/* Abhängig von "iTyp" gilt : */
/* 1: logischer Wert (FALSE/TRUE) */
/* 2: binärer Wert (32 Bit) */
/* 3: numerischer Wert (float) */
/* */
/* 10: alphanum. Zeichenfolge */
/* */
/* 99: UNDEF */
/* */
/* Beim Erzeugen der ADE-Datei in */
/* SP_ESP_ADE werden folgende Werte */
/* für "iTyp" verwendet : */
/* -1: Pos. "?if(" in ADE-Datei */
/* -2: Pos. ")" dazu in ADE-Datei */
/* -3: Pos. "?else" in ADE-Datei */
/* -11: Pos. "?while(" in ADE-Datei */
/* in SP_ESP2ADE bzw. */
/* Schleifenzähler vorherige */
/* Wiederholung in ADE_SCAN */
/* -12: Pos. ")" dazu in ADE-Datei */
/* -101: Pos. "sp_ewarte(" in ADE- */
/* Datei */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ESP_PUSH Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */

```

```
/*  
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */  
*/
```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ESP_POP.C */
/* Aufruf : SP_ESP_POP ( ESP, iTyp, cWert ) */
/*****
/* Beschreibung : */
/* Diese Routine fügt für das "ESP" den letzten Datensatz aus dem */
/* Puffer bzw. der STC-Datei und liefert den Typ in "iTyp" und den */
/* Inhalt "cWert" als Ergebnis. */
/* Enthält der Puffer noch einen Datensatz für "ESP", so wird er */
/* direkt als Ergebnis geliefert. Sonst wird der letzte Datensatz */
/* aus der STC-Datei gelesen. */
/* */
/* aufgerufen von : SP_ESP2ADE, SP_ADE_SCAN, SP_DP_SCAN */
/* */
/* ruft auf : SP_STC_WRITE */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_ESP* ESP Zeiger auf den Eintrag in der */
/* ESP-Struktur, für den der Stack */
/* geführt wird. */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ESP_POP Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST vgl. SP_ERR_... */
/* */
/* int *iTyp Typ des Wertes in "cWert" */
/* ( vgl. unter "cWert" in PUSH ) */
/* */
/* char** cWert Wert abhängig vom Typ */
/* ( vgl. unter "cWert" in PUSH ) */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* ----- */
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*  Datei    :   STC_WRIT.C             */
/*  Aufruf   :   SP_STC_WRIT ()        */
/*****
/*  Beschreibung :                       */
/*  Diese Routine fügt für das zu "ESP_SP_STC_Buf" gehörende ESP */
/*  den Datensatz aus STC-Puffer an die zugehörige STC-Datei an.  */
/*  */
/*  aufgerufen von :   SP_ESP_PUSH, SP_ESP_POP                    */
/*  */
/*  ruft auf       :                                           */
/*  */
/*  I/O-Kanäle    :                                           */
/*  */
/*****
/*  Eingabe Parameter :                       */
/*  ----- */
/*  Typ   Name           Dim.   Beschreibung                    */
/*  */
/*  *** KEINE ***                                           */
/*  */
/*  ----- */
/*  Ausgabe Parameter:                       */
/*  ----- */
/*  Typ   Name           Dim.   Beschreibung                    */
/*  */
/*  int   SP_STC_WRIT                                Returnkode der Funktion */
/*  */
/*  */
/*  0=FALSE: KEIN Fehler                                  */
/*  SONST: vgl. SP_ERR...                                */
/*  */
/*  ----- */
/*  Transiente Parameter :                       */
/*  ----- */
/*  Typ   Name           Dim.   Beschreibung                    */
/*  */
/*  *** KEINE ***                                           */
/*  */
/*****
/*  Autor      Datum      Version   Kurzzeichen   */
/*  ----- */
/*  Original :   W. Fries   01.05.97   1.1          */
/*  Änderung:                                     */
/*  */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : ESP2ADE.C */
/* Aufruf : SP_ESP2ADE ( iModus, cBuf ) */
/*****
/* Beschreibung : */
/* Diese Routine erzeugt die ADE-Datei aus der ESP-Datei. Dabei */
/* wird abhängig von "iModus" die ADE-Datei initialisiert, die */
/* nächste ESP-Zeilen gescannt oder die ADE-Datei abgeschlossen. */
/* Beim Scannen werden folgende Zeichen als Trennzeichen verwendet:*/
/* - alle Zeichen < Leerzeichen */
/* - Leerzeichen, soweit nicht innerhalb eines Kommentares oder */
/* einer alphanumerischen Zeichenfolge */
/* - ')' analog */
/* - ';' analog */
/* - '"' analog, jedoch als Abschlußzeichen einer alphanumerischen */
/* Zeichenfolge */
/* */
/* aufgerufen von : SP_CMD, SP_ESP_INIT, SP_ESP_CMD */
/* */
/* ruft auf : SP_ADE_SCAN, SP_ESP_PUSH, SP_ESP_POP, */
/* UT_NORM_STRING */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int iModus ADE-Modus-Steuerung */
/* (vgl. SP_ESP2ADE...) */
/* char *cBuf Für CREATE: Zeiger auf SPP */
/* Für SCAN: Zeiger auf ESP-Zeile */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_ESP2ADE Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand     :   01.10.98  */
/*****
/*  Datei     :   ADE_SCAN.C           */
/*  Aufruf    :   SP_ADE_SCAN ( ESP, iModus, cAPDU )           */
/*****
/*  Beschreibung :   */
/*  Diese Routine scannt die ADE-Datei zu "ESP". Abhängig von */
/*  "iModus" werden die einzelnen Ausdrücke ausgeführt und der */
/*  weitere Programmablauf gesteuert ("?if", "?else", "?while", */
/*  "sp_ewarte" oder Zuweisungen an Datenpunkte).           */
/*  */
/*  aufgerufen von :   SP_ESP2ADE, SP_ESP_AKTIV, SP_ESP_RSP_USM */
/*  */
/*  ruft auf       :   SP_DP_SCAN, SP_SV_SCAN, SP_ZSP_START,   */
/*                   SP_ESP_START, SP_ZSP_STOP, SP_ESP_STOP,   */
/*                   SP_ESP_PUSH, SP_ESP_POP,                 */
/*                   UT_UDATETAG, UT_UTIMESEK, UT_DT_ADD,     */
/*                   UT_ESC_SPRINTF, UT_DR_OPROT, UT_NORM_STRING, */
/*                   UT_DP_FINDFIRST, UT_DP_FINDNEXT, UT_CFG_WRIT */
/*  */
/*  I/O-Kanäle     :   */
/*  */
/*****
/*  Eingabe Parameter :   */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung           */
/*  */
/*  SP_ESP*  ESP           */      Zeiger auf den Eintrag in der */
/*  */      ESP-Struktur für den der */
/*  */      nächste Teil einer Befehlsfolge */
/*  */      gescannt werden soll.     */
/*  */
/*  int      iModus        */      Modus für die Ablaufsteuerung */
/*  */      0: Ablauf eingeschränkt für */
/*  */      Prüfung aus ESP_ADE        */
/*  */      1: Ablaufsteuerung aktiv    */
/*  */
/*  char*    cAPDU         */      Zeiger auf das RSP-/USM-Datagr., */
/*  */      das auf für das ESP relevante */
/*  */      Ereignisse geprüft werden soll. */
/*  */      SONST: NULL-Zeiger         */
/*  */
/*  ----- */
/*  Ausgabe Parameter:   */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung           */
/*  */
/*  int      SP_ADE_SCAN    */      Returnkode der Funktion      */
/*  */      0=FALSE: KEIN Fehler       */
/*  */      SONST: vgl. SP_ERR...     */
/*  */
/*  ----- */
/*  Transiente Parameter :   */
/*  ----- */
/*  Typ      Name          Dim.   Beschreibung           */
/*  */
/*  *** KEINE ***         */
/*  */
/*****
/*  Autor      Datum      Version  Kurzzeichen   */
/*  ----- */
/*  Original :   W. Fries   01.05.97   1.1         */
/*  Änderung:   */
/*  */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : DP_SCAN.C */
/* Aufruf : SP_DP_SCAN ( ESP, iModus, cDP_Var, cAPDU ) */
/*****
/* Beschreibung : */
/* Diese Routine bearbeitet die DP-Variable zu "ESP", vermerkt den */
/* zugehörigen Wert aus der DP-Struktur oder, falls vorhanden, aus */
/* den APDU auf dem Stack bzw. setzt das zugehörige CMD an den/die */
/* Datenpunkt(e) abhängig von "iModus" ab, wenn eine Zuweisung */
/* vorliegt. */
/* */
/* aufgerufen von : SP_ADE_SCAN */
/* */
/* ruft auf : SP_IZQ_CMD, SP_ESP_PUSH, SP_ESP_POP, */
/* UT_DP_ATBTEXT, UT_DP_DIMTEXT, UT_DP_OBJTEXT, */
/* UT_DP_DPTEXT, UT_NORM_STRING, UT_DP_FINDFIRST, */
/* UT_DP_FINDNEXT, UT_DP_FINDINDEX, UT_BSDPMELD, */
/* UT_UCGETBIT, diverse Makros FND_..., */
/* BS_DF_INIT, SY_IEEE2F */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_ESP* ESP Zeiger auf den Eintrag in der */
/* ESP-Struktur für den der */
/* nächste Teil einer Befehlsfolge */
/* gescannt werden soll. */
/* */
/* int iModus Modus für die Ablaufsteuerung */
/* 0: Ablauf eingeschränkt für */
/* Prüfung aus ESP_ADE */
/* 1: Ablaufsteuerung aktiv */
/* 2: Ablaufsteuerung aktiv und */
/* DP-Index im ESP-Eintrag */
/* vermerken, wenn DP gelesen */
/* wird. */
/* */
/* char* cDP_Var Datenpunkt-Variable aus der ADE- */
/* Datei */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_DP_SCAN Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR_... */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : SV_SCAN.C */
/* Aufruf : SP_SV_SCAN ( ESP, cS_Var ) */
/*****
/* Beschreibung : */
/* Diese Routine berechnet die System-Variable zu "ESP", vermerkt */
/* den zugehörigen Wert auf dem Stack. */
/* */
/* aufgerufen von : SP_ADE_SCAN */
/* */
/* ruft auf : SP_ESP_PUSH, UT_UDATETAG, UT_UTIMESEK */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* SP_ESP* ESP Zeiger auf den Eintrag in der */
/* ESP-Struktur für den die */
/* System-Variable auf dem Stack */
/* vermerkt werden soll. */
/* */
/* char* cS_Var System-Variable aus ADE-Datei */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_SV_SCAN Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*   Datei    :   IZQ_CMD.C             */
/*   Aufruf   :   SP_IZQ_CMD ( cOBJ, cDP, iBefehl, fWert )    */
/*****
/*   Beschreibung :   */
/*   Diese Routine erzeugt für alle in "CDP" angegebenen Datenpunkte */
/*   (Angabe von zeichen- und/oder textsensitiven Platzhaltern ist */
/*   möglich !!), beschränkt auf das Objekt "cOBJ", das CMD-Datagramm*/
/*   zum Ausführen des Befehls "iBefehl" mit dem neuen Wert "fWert". */
/*   Jedes FND-Datagramm wird in der Datei zur Verwaltung der zu bzw.*/
/*   gesendeten CMD-Datagramme eingetragen und von der Routine */
/*   SP_IZQ_POLL als Ergebnis zur Ausgabe an die DDC-IZ geliefert.  */
/*   */
/* WF1 iBefehl >=120 und iBefehl <= 148 ergänzt !! */
/*   */
/*   aufgerufen von :   GAK_IIZ, SP_ZSP_AKTIV, SP_DP_SCAN */
/*   ruft auf       :   UT_DP_FINDFIRST, UT_DP_FINDNEXT, UT_FND_TIME, */
/*                   UT_CFG_WRIT, diverse Makros FND... und */
/*                   SET_FND..., SY_F2IEEE */
/*   I/O-Kanäle     :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ   Name           Dim.   Beschreibung           */
/*   char* cOBJ           */      Kennung des Objektes auf das der*/
/*   char* cDP           */      Schaltbefehl beschränkt wird */
/*   int   iBefehl        */      Datenpunkt-Adresse (ggf. mit */
/*   float fWert          */      Platzhaltern) an die der Schalt-*/
/*   */      befehl ausgegeben wird */
/*   */      Befehl bzw. Datenfeld auf das */
/*   */      der Schaltbefehl ausgegeben wird*/
/*   */      Auszugebender Wert */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ   Name           Dim.   Beschreibung           */
/*   int   SP_IZQ_CMD     */      Returnkode der Funktion */
/*   */      0=FALSE: KEIN Fehler */
/*   */      SONST: vgl. SP_ERR... */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ   Name           Dim.   Beschreibung           */
/*   */      */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:   W. Fries   07.09.98   1.1d         WF1       */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : IZQ_RSP.C */
/* Aufruf : SP_IZQ_RSP ( cRSP ) */
/*****
/* Beschreibung : */
/* Diese Routine aktualisiert die IZQ-Datei anhand des RSP-Data- */
/* grammes "cRSP", wenn ein zugehöriger IZQ-Satz gefunden wird. */
/* Der IZQ-Satz wird freigegeben oder für einen weiteren Sende- */
/* versuch aktiviert. */
/* */
/* aufgerufen von : SP_RSP_USM */
/* */
/* ruft auf : SP_IZQ_SORT, UT_Z_SET, diverse Makros FND_... */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* char * cRSP RSP-Datagramm */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_IZQ_RSP Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*  Produkt   :   GA - Knoten           Version   :   1.2       */
/*  Modulname :   SP Schalt-Programme   Stand    :   01.10.98  */
/*****
/*  Datei     :   IZQ_POLL.C           */
/*  Aufruf    :   SP_IZQ_POLL ( )     */
/*****
/*  Beschreibung :   */
/*  Diese Routine prüft, ob ein in der IZQ-Datei verwaltetes CMD- */
/*  Datagramm an eine DDC-IZ gesendet werden soll.             */
/*  Wenn JA, wird der IZQ-Satz mit dem Zeitpunkt für den nächsten */
/*  Wiederholungsversuch in die Verweiskette einsortiert bzw.   */
/*  freigegeben, wenn die maximale Anzahl der Sendeversuche     */
/*  erreicht ist.                                               */
/*  */
/*  aufgerufen von :   SP_POLL        */
/*  ruft auf       :   SP_IZQ_SORT, UT_DT_ADD, diverse Makros FND_... */
/*  I/O-Kanäle     :   */
/*****
/*  Eingabe Parameter :   */
/*  -----             */
/*  Typ   Name           Dim.   Beschreibung           */
/*  */
/*  *** KEINE ***      */
/*  -----             */
/*  Ausgabe Parameter:   */
/*  -----             */
/*  Typ   Name           Dim.   Beschreibung           */
/*  */
/*  char* SP_IZQ_POLL    Zeiger auf das zu sendende CMD- */
/*                      Datagramm                       */
/*                      SONST: NULL-Zeiger               */
/*  -----             */
/*  Transiente Parameter :   */
/*  -----             */
/*  Typ   Name           Dim.   Beschreibung           */
/*  */
/*  *** KEINE ***      */
/*  -----             */
/*  Autor      Datum     Version  Kurzzeichen      */
/*  -----             */
/*  Original :   W. Fries   01.05.97   1.1           */
/*  Änderung:   */
/*  -----             */
/*****
/*  Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/* Produkt : GA - Knoten Version : 1.2 */
/* Modulname : SP Schalt-Programme Stand : 01.10.98 */
/*****
/* Datei : IZQ_SORT.C */
/* Aufruf : SP_IZQ_SORT ( fp_IZQ, lPosAKT, IZQ_REC_AKT ) */
/*****
/* Beschreibung : */
/* Diese Routine sortiert den aktualisierten IZQ-Satz an der Posi- */
/* tion "lPosAKT" in die Verweiskette chronologisch ein. */
/* Dabei wird die Besonderheit ausgenutzt, dass der IZQ-Satz */
/* ausgehend von seiner aktuellen Position nur weiter in Richtung */
/* der Verweiskette verschoben muß, da "SendZeit" nur größer */
/* (inkl. UNDEF) geworden sein kann (bzgl. Neueintrag vgl. IZQ_CMD) */
/* */
/* aufgerufen von : SP_IZQ_POLL, SP_IZQ_RSP */
/* */
/* ruft auf : */
/* */
/* I/O-Kanäle : */
/* */
/*****
/* Eingabe Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* FILE * fp_IZQ File-Pointer auf die bereits */
/* geöffnete IZQ-Datei */
/* long lPosAKT Position des einzusortierenden */
/* IZQ-Satzes in der IZQ-Datei */
/* ( Hat noch den alten Inhalt !! ) */
/* SP_IZQ_REC* IZQ_REC_AKT Zeiger auf den einzusortierenden */
/* IZQ-Satz aus der aufrufenden. */
/* */
/* ----- */
/* Ausgabe Parameter: */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* int SP_IZQ_SORT Returnkode der Funktion */
/* 0=FALSE: KEIN Fehler */
/* SONST: vgl. SP_ERR... */
/* */
/* ----- */
/* Transiente Parameter : */
/* ----- */
/* Typ Name Dim. Beschreibung */
/* */
/* *** KEINE *** */
/* */
/*****
/* Autor Datum Version Kurzzeichen */
/* ----- */
/* Original : W. Fries 01.05.97 1.1 */
/* Änderung: */
/* */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### 5.6.5. Druckausgaben, Meldungstexte

Abhängig vom Flag für die Protokollierung wird beim Start und Stop eines Schaltprogrammes oder beim Senden eines Schaltbefehls an eine IZ eine Meldung auf dem Protokollierungsgerät ausgegeben, wenn dieses parametrierbar und verfügbar ist.

Standardmäßig wird bei jeder Protokollausgabe mittels "sp\_printf" der Zeitstempel, die Objekt- und die Schaltprogramm-Kennung ergänzt. Dieses kann nun unterdrückt werden, indem die Format-Definition mit "\r" beginnt.

### 5.6.6. Datenkatalog

Für die Konfiguration und den aktuellen Status der Schaltprogramme werden mehrere Dateien benötigt. Diese werden im Verzeichnis SP und den darunterliegenden Unterverzeichnissen des aktuellen Arbeitsverzeichnisses gehalten.

Folgende "Konfigurations"-Dateien stehen im Unterverzeichnis SP\CFG :

SPP.CFG

KAL.CFG

Die Dateien mit den zeit- bzw. ereignisabhängigen Befehlsfolgen (Zeit- und Ereignis-Schaltprogramme) stehen im Unterverzeichnis SP\PRG :

<Programm>.ZSP für Zeit-Schaltprogramme

<Programm>.ESP für Ereignis-Schaltprogramme

<Programm>.ADE für Arbeitsdatei für Ereignis-Schaltprogramme

<Programm>.AD\_ wie .ADE; jedoch unvollständig nach fehlerhafter Umsetzung aus .ESP

<Programm>.STC für Stackdatei für Ereignis-Schaltprogramme

"Programm" ist ein beliebiger Dateiname (OHNE Pfad und Extension). Die Zuordnung zu den Schalt-Programmen wird in der Konfigurationsdatei SPP.CFG getroffen. Bei der Übertragung von Schaltprogrammen mittels der Karteipunkte



erzeugt der GA-Knoten den Dateinamen "SP\_nnnn", wobei "nnnn" der Zeilennummer in der Datei SPP.CFG entspricht, in der der zugehörige Eintrag vorgenommen wird. Diese Dateinamen bleiben für den GA-Knoten reserviert und dürfen sonst nicht verwendet werden !

**Hinweis: Der Benutzer trägt beim Editieren der Dateien die Verantwortung dafür, dass das Format, die Syntax und die Parameter korrekt sind (insbesondere eindeutige Dateinamen für "Programm" in SPP.CFG). Die Verwendung von Tabulatoren ist zur Spaltenpositionierung NICHT zulässig !!**

Soweit sinnvoll, lehnen sich die Formate für die einzelnen Parameter an die FND-Spezifikation an. Es gelten die folgenden Vereinbarungen :

- Alle Zeilen haben eine maximale Länge von 128 Zeichen (zzgl. <CR> (0x0D) und/oder <LF> (0x0A) ).
- Alle Zahlen stehen rechtsbündig in den entsprechenden Spalten und repräsentieren in Dezimaldarstellung die entsprechenden Bitkombinationen der FND-Variablen.
- Die Datenpunkt-Adressen ("dp\_id") und die zugehörigen Bezeichnungen stehen linksbündig in der jeweiligen Spalte. Sie beginnen und enden mit einem alphanumerischen Zeichen. Nicht benutzte Stellen werden mit Leerzeichen ausgefüllt.

**Hinweis: Die abschließenden Leerzeichen sind KEIN signifikanter Bestandteil der Datenpunkt-Adresse, d.h. diese werden gemäß FND-Spezifikation durch NULL-Bytes (0x00) ersetzt !!**

Der Wertebereich für 3-stellige numerische Angabe ist, soweit nicht anders angegeben, auf  $0 \leq x \leq 255$  begrenzt. Für 5-stellige numerische Angaben gilt ein maximaler Wertebereich von  $0 \leq x \leq 32767$ .

Reelle Zahlen (14-stellig) können als ganze Zahlen bzw. in Gleitkomma- oder Exponential-Darstellung angegeben werden (inkl. Vorzeichen für Mantisse und Exponent). Dabei ist jedoch die Auflösung auf eine 7-stellige Mantisse und der Wertebereich, soweit nicht anders angegeben, auf annähernd  $-3.4E+38 \leq x \leq +3.4E+38$  begrenzt. Ergänzend dazu werden "ungültige Werte", wenn sie gemäß FND-Spezifikation zulässig durch die Zeichenfolge "FFFFFFFF" (ohne ") angegeben.

Werden Konfigurations-Parameter durch die LZH verändert (CMD-Datagramme), so werden diese Änderungen in dieser Datei vermerkt, d.h. nach einem Neustart des Programmes steht der geänderte Parameter wiederum zur Verfügung. Als Folge davon müssen Zeilen, die eine feste Länge haben (Länge der letzten Spalte explizit definiert; vgl. insbesondere ES-Zeilen), in der vollen Länge angegeben sein.

**Hinweis: Einige Texteditoren entfernen Leerzeichen, die hinter dem letzten sichtbaren Zeichen angegeben sind. In diesen Fällen ist die geforderte rechtsbündige Ausrichtung unbedingt einzuhalten!!!!**

Die von den Schaltprogrammen versendeten FND-Befehle (CMD) werden in der Datei "SP\_IZQ.TMP" im aktuellen Verzeichnis verwaltet und an das IZ-Modul weiterleitet. Der Zustand bzw. Wert des Datenpunktes im GA-Knoten ändert sich aber erst nach dessen Bearbeitung durch die IZ und den Empfang der zugehörigen FND-Antwort (RSP) bzw. Rückmeldung (USM).

Werden während dieser Bearbeitungszeit weitere gleichartige FND-Befehle für den Datenpunkt erzeugt, so werden diese zwischengespeichert. Nach Empfang der ausstehenden FND-Antwort wird NUR der letzte FND-Befehl an das IZ-Modul weitergeleitet. Durch dieses Optimierungsverfahren wird sichergestellt, dass die aktuell gültige Vorgabe weitergeleitet wird, ohne die IZ mit überholten Vorgaben zu belasten.

Enthält die FND-Antwort den Hinweis auf einen temporären Ablauffehler (REJECT), so wird der letzte FND-Befehl nach einer Wartezeit wiederholt.

Die Datei "SPP.CFG" enthält die Schalt-Programm-Parameter (SPP) sowie die Zuordnung der Schaltprogramm-Kennungen zu den Programm-Dateinamen im Unterverzeichnis SP\PRG.

Werden Schaltprogramme durch die LZH verändert (CMD-Datagramme mit Karteipunkt "\$SCHALTPROG-oooo", vgl. LV unter 4.4 ff.), so werden diese Änderungen in dieser Datei vermerkt, d.h. nach einem Neustart des GAK-Programmes steht der geänderte Parameter wiederum zur Verfügung. Veränderung von Schaltprogrammen durch die LZH kann z.B. sein:

- Übertragung eines neuen Schaltprogramms bzw. einer neuen Version eines bereits hinterlegten Schaltprogrammes.
- Aktivierung eines hinterlegten Schgaltprogrammes

Alle Zeilen, die das Zeichen '!' als erstes Zeichen der Zeile enthalten (auch keine Leerzeichen davor), sind Kommentarzeilen und werden komplett ignoriert.

Die Zeile einer Zuordnungs-Definition hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
-----					

1	1	14	14	Akt.-Datum/Zeit	Format: JJJJMMTHHMMSS "99999999999999", falls die Zeile freigegeben ist, d.h. für weitere Verwendung reserviert wird.
2	16	19	4	Objekt-Kennung	Kennung des zugehörigen Objektes
3	21	30	10	Programm-Kennung	Kennung des Schalt-Programms
4	32	34	3	Versionsnummer	Versionsnummer des Schalt-Progr.
5	36	38	3	Zeitbasis	Zeitbasis des Schaltprogramms mit folgenden Kodierungen: 1 Stunde (zyklisch) 2 Tag (zyklisch) 3 Woche (zyklisch) 4 Monat (zyklisch) 5 Jahr (zyklisch) 11 Stunde (sporadisch) 12 Tag (sporadisch) 13 Woche (sporadisch) 14 Monat (sporadisch) 15 Jahr (sporadisch)
6	40	42	3	Protokollierung	Flag für Protokollierung 0 : Keine Protokollierung SONST: Mit Protokollierung
7	44	51	8	DateiName	Dateiname des zum Schaltprogramm gehörenden Zeit- und Ereignis-SP (nach DOS-Konvention)
8	53	55	3	Status ZSP	Status des Zeit-SP 0: NICHT aktiv / vorhanden 1: gestoppt 2: gestartet
9	57	59	3	Status ESP	Status des Ereignis-SP 0: NICHT aktiv / vorhanden 1: gestoppt 2: gestartet
10	61	120	60	Beschreibung	Beschreibungstext Beliebiger Text für die Protokollmeldungen  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

KOMMENTAR-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	"!"	Zeilen-Kennung (OHNE '"')
2	2	128	<=127	Kommentartext	Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 128 Zeichen

Die Datei "KAL.CFG" enthält den aktuellen Zustand des Schaltprogramm-Kalenders. Alle Zeilen, die das Zeichen "!" als erstes Zeichen der Zeile enthalten (auch keine Leerzeichen davor), sind Kommentarzeilen und werden komplett ignoriert.

Die Zeile einer Zuordnungs-Definition hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	14	14	Datum/Zeit	Format: JJJJMMTHHMMSS "99999999999999", falls die Zeile freigegeben ist, d.h. für weitere Verwendung reserviert wird.  Uhrzeit für das Starten bzw. Stoppen des Schaltprogrammes
2	16	19	4	Objekt-Kennung	Kennung des zugehörigen Objektes
3	21	30	10	Programm-Kennung	Kennung des Schalt-Programms
4	32	34	3	Funktion	auszuführenden Funktion mit folgenden Kodierungen: 0: STOP eines Schaltprogramms SONST: START eines Schaltprogramms
5	36	38	3	Protokollierung	Flag für Protokollierung 0 : Keine Protokollierung SONST: Mit Protokollierung
6	40	99	60	Beschreibung	Beschreibungstext Beliebiger Text für die Protokollmeldungen  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

KOMMENTAR-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	"!"	Zeilen-Kennung (OHNE '!')
2	2	128	<=127	Kommentartext	Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 128 Zeichen

Die Dateien "<Programm>.ZSP" enthalten die zeitabhängigen Befehlsfolgen (Zeit-Schaltprogramm) für die Ausgabe an die Datenpunkte. In jeder Zeile wird für einen Zeitpunkt der auszugebende Befehl und der zugehörige Datenpunkt angegeben (Zeitschaltbefehl).

Alle Zeilen, die das Zeichen '!' als erstes Zeichen der Zeile enthalten (auch keine Leerzeichen davor), sind Kommentarzeilen und werden komplett ignoriert.

Die Kommandos werden sequentiell in der Reihenfolge, in der sie in der Datei stehen, ausgeführt, d.h. sie müssen chronologisch sortiert sein. Jedes Kommando wird zu der spezifizierten relativen Zeit, bezogen auf den Zeitbezug

entsprechend der für das Schaltprogramm definierten Zeitbasis (vgl. SPP-Datei) ausgegeben. Falls für ein Zeile eine kleinere Zeit als für den Vorgänger angegeben ist, so wird der zugehörige Zeitschaltbefehl sofort ausgegeben.

Beim Starten eines ZSP werden die einzelnen Zeilen vom Beginn der Datei sequentiell abgearbeitet, bis der Zeitpunkt zum Ausführen des nächsten Befehls in der Zukunft liegt, d.h. es werden die Datenpunkt- und Datenfeld-spezifischen Befehle für die IZ's so erzeugt, als wäre das ZSP bereits zum Beginn des aktuellen Zeitbasis-Intervalles gestartet gewesen (Aufsetzen). Um ein korrektes Aufsetzen sicherzustellen, muß der Benutzer alle im ZSP angesprochenen Datenpunkte und Datenfelder für den Zeitpunkt "0.0" initialisieren.

**Hinweis: Der Benutzer trägt beim Editieren der Dateien die Verantwortung dafür, dass das Format, die Syntax und die Parameter korrekt sind.**

**Desweiteren muß der Benutzer sicherstellen, dass sich mehrere Schaltprogramme nicht ungewollt gegenseitig beeinflussen, indem sie dasselbe Datenfeld eines Datenpunktes modifizieren. Dieses gilt insbesondere auch für den Programmstart, da nicht sichergestellt werden kann, in welcher Reihenfolge die Schaltprogramme gestartet und aufgesetzt werden. Es wird empfohlen, die Schaltprogramme über die Funktionen SP\_START, SP\_STOP, SP\_CALL und SP\_EXIT für Ereignis-Schaltprogramme zu synchronisieren.**

Gegenüber dem LV unter 4.4.4 wurden datenpunktbezogenen Datenfelder ergänzt. Folgende Datenfelder dienen nur der Ausgabe zusätzlicher Datenpunkt-Informationen und können nur im ESP gelesen werden :

<b>Kode</b>	<b>Datenfeld</b>	<b>Format und Filtermaske</b>	<b>Bemerkung</b>
91	dp_type	(B*1 für "dp_type" = 1 bis 5)	Wert aus "dp_type" (1 - 5)
92	dp_subtype	(B*1 für "dp_type" = 1 bis 5)	Wert aus "dp_subtype"
100	obj_kennung	(Text für "dp_type" = 1 bis 5)	Kennung aus OBJEKT-Zeile
101	obj_text	(Text für "dp_type" = 1 bis 5)	Text aus OBJEKT-Zeile
102	dp_text	(Text für "dp_type" = 1 bis 5)	Text aus DATENPUNKT-Zeile
112	atb_text	(Text für "dp_type" = 1 und 2)	Text für "atb_id" und "actual"
114	dim_text	(Text für "dp_type" = 3 bis 5)	Text für "dimension"

Nachfolgende Datenfelder können im ESP gelesen als auch durch das ESP und ZSP modifiziert werden :

<b>Kode</b>	<b>Datenfeld</b>	<b>Format und Filtermaske</b>	<b>Bemerkung</b>
120	flag_meld	(B*1 für "dp_type" = 1 bis 5)	Flag für Meldungsverhalten
121	flag_prot	(B*1 für "dp_type" = 1 bis 5)	Flag für Protokollierung
130	geht_dim	(B*1 für "dp_type" = 1 und 2)	Dimension zu 131
131	geht_dauer	(R*4 für "dp_type" = 1 und 2)	min. Dauer GEHT
132	kommt_dim	(B*1 für "dp_type" = 1 und 2)	Dimension zu 133
133	kommt_dauer	(R*4 für "dp_type" = 1 und 2)	min. Dauer KOMMT
134	periode_dim	(B*1 für "dp_type" = 1 und 2)	Dimension zu 135
135	periode_dauer	(R*4 für "dp_type" = 1 und 2)	min. Dauer Periode
136	max_flimm	(B*1 für "dp_type" = 1 und 2)	max. Anzahl Flimmerer
137	flag_bearb	(B*1 für "dp_type" = 1 und 2)	Flag, ob weitere Bearbeitung
140	zykluszeit	(R*4 für "dp_type" = 3 bis 5)	Zykluszeit
141	hysterese	(R*4 für "dp_type" = 3 bis 5)	Hysterese
142	sync_code	(B*1 für "dp_type" = 3 bis 5)	Sync.-Kode / Dimension zu 143
143	sync_zeit	(R*4 für "dp_type" = 3 bis 5)	Sync.-Zeitpunkt
144	vorverarb	(B*1 für "dp_type" = 3 bis 5)	Vorverarbeitung
145	abfrage	(R*4 für "dp_type" = 3 bis 5)	Abfrageintervall
146	a0	(R*4 für "dp_type" = 3 bis 5)	Skalierungskoeffizient a0
147	a1	(R*4 für "dp_type" = 3 bis 5)	Skalierungskoeffizient a1
148	a2	(R*4 für "dp_type" = 3 bis 5)	Skalierungskoeffizient a2

Ab Version 1.42 können die aktuellen Werte interner Datenpunkte ("actual", vgl. Treiber-Id. 240) aus dem Zeit- und/oder Ereignis-Programm heraus modifiziert werden. damit ist es z.B. möglich, berechnete Werte analog zu dem messtechnisch erfassten Werte zu behandeln. Für Zählpunkte müssen die Werte stetig steigend sein, da sonst ein Setzvorgang durch die Monotoniekontrolle gemeldet wird.

Die Zeile eines Kommando-Definition hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung																		
1	1	14	14	Zeitpunkt	<p>Relativzeit für die Ausführung des zugehörigen Befehls bezogen auf die für das Schaltprogramm spezifizierte Zeitbasis (analog zu den FND-Dimensions-Kodierungen 164, 165, 167 und 169; vgl. FND-Ergänzungen)</p> <p>Zuordnung FND-Kode zur SPP-Zeitbasis wie folgt :</p> <table border="1"> <thead> <tr> <th>Zeitbasis</th> <th>FND-Kode</th> <th>Zeitpunkt in</th> </tr> </thead> <tbody> <tr> <td>1, 11:</td> <td>169</td> <td>Minuten / Stunde</td> </tr> <tr> <td>2, 12:</td> <td>167</td> <td>Stunden / Tag</td> </tr> <tr> <td>3, 13:</td> <td>165</td> <td>Tagen / Woche</td> </tr> <tr> <td>4, 14:</td> <td>164</td> <td>Tagen / Monat</td> </tr> <tr> <td>5, 15:</td> <td>161</td> <td>Tagen / Jahr</td> </tr> </tbody> </table>	Zeitbasis	FND-Kode	Zeitpunkt in	1, 11:	169	Minuten / Stunde	2, 12:	167	Stunden / Tag	3, 13:	165	Tagen / Woche	4, 14:	164	Tagen / Monat	5, 15:	161	Tagen / Jahr
Zeitbasis	FND-Kode	Zeitpunkt in																					
1, 11:	169	Minuten / Stunde																					
2, 12:	167	Stunden / Tag																					
3, 13:	165	Tagen / Woche																					
4, 14:	164	Tagen / Monat																					
5, 15:	161	Tagen / Jahr																					
2	16	31	16	dp-id	Zugehörige Datenpunkt-Adresse, ggf. mit zeichen- bzw. abschließendem textsensitiven Platzhalter ('?' bzw. '*')																		
3	33	35	3	Datenfeld	Datenfeld für die Befehlsausgabe Datenpunktadresse, z.B. "nominal" ( Kodierung gemäß LV unter 4.4.4 auf Seite 29 und obige Ergänzungen )																		
4	37	50	14	Befehl	Befehl bzw. Vorgabewert abhängig vom Datenfeld																		
5	52	54	3	Protokollierung	Flag für Protokollierung 0 : Keine Protokollierung SONST: Mit Protokollierung																		
6	56	115	<=60	Beschreibung	<p>Beschreibungstext Beliebiger Text für die Protokollmeldungen</p> <p>Wird durch das Zeilenabschlußzeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor) abgeschlossen.</p>																		

KOMMENTAR-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	"!"	Zeilen-Kennung (OHNE '"')
2	2	128	<=127	Kommentartext	Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 128 Zeichen

Die Dateien "<Programm>.ESP" enthalten die ereignisabhängigen Befehlsfolgen (Ereignis-Schaltprogramm) für die Ausgabe an die Datenpunkte in dem Format, in dem sie zeilenweise editiert bzw. übertragen wurden.

Die syntaktisch zulässigen Ausdrücke (Token) für die Ereignis-Schaltprogramme entsprechen den im DV-technischen Feinkonzept definierten Variablen, Konstanten, Operatoren, Zuweisungen, Funktionen und Kommentare.

Standardmäßig wird bei jeder Protokollausgabe mittels "sp\_printf" der Zeitstempel, die Objekt- und die Schaltprogramm-Kennung ergänzt. Dieses kann unterdrückt werden, indem die Format-Definition mit "\r" beginnt. Dadurch können beliebige Protokoll-Ausgaben aus dem Ereignis-Schaltprogramm heraus erzeugt werden.

Die Schreibweise ist Zeichensensitiv, d.h. sie entspricht der Schreibweise in der Leistungsverzeichnis. Leer- und Zeilenabschlußzeichen dienen als Abschlußzeichen für einen syntaktischen Ausdruck (Token). Die bei Variablen, Funktionen, Fallunterscheidungen und Wiederholungen innerhalb der Klammern genannten Ausdrücke sind eigenständige Token, so dass der Klammerinhalt über mehrere Zeilen verteilt werden kann.

Jede Zeile wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die maximale Zeilenlänge beträgt insgesamt 255 Zeichen.

**Hinweis: Bei Verwendung von "sp\_ewarte" wird empfohlen, die angegebene Bedingung NUR bei einer Wertänderung WAHR werden zu lassen. Ansonsten besteht die Gefahr, dass die nachfolgenden Befehle permanent ausgeführt werden und zu einer unnötigen Systembelastung führen. Zu diesem Zweck kann der aktuelle Vergleichswert in einer Hilfsvariablen vermerkt und in der Bedingung auf ungleich (" != ") verglichen werden.**

**Ist die Anzahl der in der Bedingung angegebenen Datenpunkte <= 4 (DOS) bzw. 64 (LINUX), so wird ein optimiertes Verfahren zur Prüfung der Bedingung angewandt, das die Systembelastung minimiert.**

Die Dateien "<Programm>.ADE" enthalten die ereignisabhängigen Befehlsfolgen (Ereignis-Schaltprogramm) für die Ausgabe an die Datenpunkte in einem für die Abarbeitung aufbereiteten Format. Zusätzlich wird der Inhalt der ESP-Datei um organisatorische Informationen ergänzt.

Zusätzlich wird der aktuelle Zustand des ablaufenden ESP vermerkt, so dass der Zustand nicht im Hauptspeicher vermerkt werden muß.

Bei jedem Programmstart und jeder Übertragung mittels des Karteipunktes von der LZH liest eine spezifische Routine im SP-Modul (Parser) die ESP-Datei und erzeugt dabei die ADE-Datei, wenn die ESP-Datei neueren Datums ist als die ADE-Datei. Tritt bei der Erzeugung der ADE-Datei ein Fehler auf, so wird die bis dahin erzeugte ADE-Datei in "<Programm>.AD\_" umbenannt, so dass anhand dieser nachvollzogen werden kann, an welcher Programmstelle der



Fehler auftrat. Gleichzeitig ist dadurch sichergestellt, dass keine fehlerhafte ADE-Datei existiert, die bei einem Ablauf zu Fehlern führen könnte.

Der Dateikopf mit den Zustandsinformationen hat folgenden Aufbau :

Zeile 1:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	14	14	Start-Position	FSEEK-Position der ERSTEN Programmzeile (Programmanfang) SONST: UNDEF

Zeile 2 bis 9:                   \*\*\* reserviert für zukünftige Erweiterungen \*\*\*

Zeile 10 bis 109:           Hilfsvariable

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	6	6	Datentyp	wie folgt kodiert : 0 leer SONST vgl. "SP_STC_DTYP_..." in SP_INCL.H
2	8	262	255	aktueller Wert	Abhängig vom Datentyp kodiert im Format "%s" bzw. "%G". Das '\0' dient als Ende- Kennung des String.

Zeile 110 ff.:               Programmzeilen

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	255	<=255	Token	Syntaktischer Ausdruck, der entsprechend der Postfix- Notation sequentiell abgear- beitet werden kann. (Beschreibung der zulässigen Token vgl. unten)  Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die syntaktisch zulässigen Ausdrücke (Token) für die Ereignis-Schaltprogramme entsprechen weitgehend den im Leistungsverzeichnis definierten Variablen, Konstanten, Operatoren, Zuweisungen, Funktionen und Kommentare. Die Schreibweise ist zeichensensitiv, d.h. sie entspricht der Schreibweise in der Leistungsverzeichnis. Ein Token darf keine Leerzeichen enthalten.

Für Funktionen mit Aufrufparametern, Fallunterscheidungen und Wiederholungen endet die Zeile mit der öffnenden Klammer. Die zugehörigen Parameter werden in den folgenden Zeilen als eigene Token behandelt.

Zur Optimierung der Ablaufgeschwindigkeit werden vom Parser beim Programmstart bzw. beim Empfang der CMD-Datagramme für den Karteipunkt von der LZH folgende Hilfsinformationen ergänzt :

1) FSEEK-Position des ERSTEN Token der Befehlsfolge 2 (ELSE-Fall) im Anschluß an die schließende Klammer für den logischen Ausdruck. Ist kein ELSE-Fall definiert oder bei Wiederholungen (WHILE-Schleife) wird die Position des ERSTEN Token HINTER der Befehlsfolge 1 angegeben, d.h. in diesen Fällen wird die Befehlsfolge 1 (THEN-Fall) übersprungen, wenn der logische Ausdruck FALSE liefert.

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	")"	Schließende Klammer (OHNE "")
2	3	16	14	Sprung-Adresse	FSEEK-Position zum Überspringen der Befehlsfolge 1

2) FSEEK-Position des ERSTEN Token HINTER der Befehlsfolge 2, d.h. falls die Befehlsfolge 1 (THEN-Fall) ausgeführt wurde, wird die Befehlsfolge 2 (ELSE-Fall) übersprungen.

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	5	5	"?else"	Kennung für ELSE-Fall (OHNE "")
2	7	20	14	Sprung-Adresse	FSEEK-Position zum Überspringen der Befehlsfolge 2

3) FSEEK-Position zum Rücksprung auf den Anfang einer Wiederholung (WHILE-Schleife), um den logischen Ausdruck erneut zu prüfen.

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	1	1	;"	Ende der Wiederholung (OHNE "")
2	3	16	14	Sprung-Adresse	FSEEK-Position zum Rücksprung auf den Anfang der Wiederholung ( "?while(" )

Die Dateien "<Programm>.STC" enthalten die beim Ablauf des zugehörigen Ereignis-Schaltprogrammes anfallenden Zwischenergebnisse, die auf dem Stack abgelegt werden.

Die Datei wird mit jedem Start des Ereignis-Schaltprogrammes neu erzeugt. Gespeichert werden sowohl Zwischenergebnisse des Programm-Zustandes, die sich abhängig vom bearbeiteten Ausdruck ergeben, als auch Daten.

Der Datei hat folgenden zeilenweisen Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

1	1	14	14	Pos. Vorgänger	FSEEK-Position des vorherigen Stackeintrags SONST: UNDEF
2	16	21	6	Typ	Typ-Kennung für den im Stack gespeicherten Wert (vgl. SP_STC_ATYP... und SP_STC_DTYP... in SP_INCL.H)
3	23	150	<=128	Wert	logischer, binärer, numerischer oder alphanumerischer Wert abhängig vom Typ  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach Dateiformat abgeschlossen.

Die Datei "SP\_IZQ.TMP" ist eine temporäre Hilfsdatei, die vom GA-Knoten erzeugt wird. Abhängig von dem zugehörigen RSP-Datagramm wird der Eintrag nach dessen Empfang gelöscht bzw. die Wiederholung eingeleitet, wenn Aussicht auf Behebung der das REJECT auslösenden Störung besteht (z.B. Datenpunkt momentan nicht verfügbar).

Die Datei ist binär entsprechend der in SP\_INCL.H definierten Struktur SP\_IZQ\_REC aufgebaut und wird ausschließlich programmtechnisch vom GA-Knoten bearbeitet, d.h. ein Editieren o.ä. ist nicht zulässig.

#### **5.6.6.1. Fehlerbehandlung**

Beim Ablauf der Schaltprogramme können Fehler auftreten, die vom SP-Modul durch die in der Datei "SP\_INCL.H" beschriebenen Kodierungen (SP\_ERR..) gekennzeichnet werden.

Ein beim Start des GA-Knotens erkannter Fehler führt zum Abbruch des Programmstarts und einer zugehörigen Fehlermeldung in der Error-Log-Datei (vgl. EL-Kommando in der Konfigurationsdatei GAK.CFG).

**Hinweis: Manuelle Änderungen an den Ereignis-Schaltprogrammen (z.B. mit einem Editor) dürfen ausschließlich in der ESP-Datei vorgenommen werden, da nur so sichergestellt ist, dass bei der automatischen Konvertierung in die ADE-Datei Fehler erkannt werden.**

Wird der Fehler bei der Übernahme mittels eines Karteipunktes von der LZH erkannt, so sendet der GA-Knoten ein ERROR-Datagramm an die LZH und bricht die Übernahme ab.

Wird ein Fehler zur Laufzeit erkannt, so versucht das SP-Modul den Ablauf aufrecht zu erhalten. Im Falle eines fatalen Fehler, der für das betroffene Schaltprogramm einen weiteren Ablauf nicht erlaubt, stoppt das SP-Modul das Schaltprogramm und schreibt eine Fehlermeldung in die Error-Log-Datei.

**Hinweis:** Eine besondere Eigenschaft der Postfix-Notation ist ihre Abarbeitung mittels eines pulsierenden Zwischenergebniskellers (Stack). Am Ende einer Formel steht das erarbeitete Ergebnis an der ERSTEN Position. Durch eine anschließende Zuweisung, den Vergleich des logischen Ausdrucks nach Warten auf ein Ereignis, einer Fallunterscheidung bzw. einer Wiederholung oder eine Textausgabe (vgl. Funktion "sp\_print") wird der Stack vollständig geleert.

Diese Eigenschaft gilt auch für die geschachtelten logischen Ausdrücke und Befehlsfolgen, d.h. der Stack muß nach Auswertung des logischen Ausdrucks bzw. am Ende einer Befehlsfolge auf derselben Ebene sein, wie vor Bearbeitung des logischen Ausdrucks bzw. der Befehlsfolge. Wird also innerhalb der Befehlsfolge für eine Fallunterscheidung oder Wiederholung ein Ergebnis berechnet, das im weiteren Verlauf des Programmes benötigt wird, so muß es einer Hilfs-Variablen zugewiesen werden.

### 5.6.7. Definition der Datenübertragung

\*\*\* KEINE \*\*\*

## 5.7. Leitzentralen (LZ)

**LZ LIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit der Kommunikation zu den Leitzentralen zusammenhängen (LZH, BWD-Gefahr, BWD-Einbruch und zugehörige Karteipunkte). Hierzu zählt auch die Bereitstellung der Daten für komplexere Abfragen gemäß DV-technischen Feinkonzept unter 4.5 als zyklische Istwertabfrage (IWA) - z.B. für Belebung der Anlagenbilder (Schema).

Die vom Hauptprogramm übergebenen FND-Datagramme werden zwischengespeichert. Abhängig vom zugeordneten Flag für das Meldeverhalten des jeweiligen Datenpunktes (vgl. ES-Zeile) wird versucht die Verbindung sofort aufzubauen oder auf ein im objekt-spezifischen Verbindungs-Profil (VBP) parametrisiertes Ereignis gewartet.

Die Werte aller Datenpunkte werden vom Hauptprogramm bei jeder Erfassung zusammen mit dem Zeitstempel vermerkt. Dieses sekundäre Prozessabbild bildet die Grundlage für die bei der zyklischen Istwertabfrage (IWA) gelieferten Werte und Zustände.

Zusätzlich zu dem in der ES-Zeile konfigurierten Abfrageintervall fragt das LZ-Modul alle die Mess-, Stell- und Zählpunkte nacheinander in der IZ ab, für die eine zyklische Istwertabfrage (IWA) parametrier ist. Dadurch werden die in Anlagenbildern dargestellten Werte in kürzeren Intervallen als bisher aktualisiert. Das Verfahren passt sich automatisch an die herstellerepezifischen Antwortzeiten der IZ an und stellt auch die Behandlung der anderen Programm-Module sicher.

### **5.7.1. Übersicht der verwendeten Module**

LZ	Leitzentralen-Ebene (LZH, BWD-Gefahr, BWD-Einbruch)
X25	X.25-Schnittstelle (falls nicht deaktiviert)
CAPI	CAPI-Schnittstelle (Version 2.0)
IP	Internet-Protokoll (Version 4)
IPC	Inter-Process-Communication
UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### **5.7.2. Liste mit Namen und Aufgaben**

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

LZ_INIT	Initialisierung beim Programmstart
LZ_RSP_USM	Übernahme und Bearbeitung der von den IZ bzw. den Modulen GAK und SP gesendeten RSP- und USM-Datagramme
LZ_POLL	Polling für die zyklische Bearbeitung der Kommunikation zu den Leitzentralen
LZ_STAT	Übergabe des aktuellen LZ-Status
LZ_ZA_TIME	Übergabe der von der ISDN-Schnittstelle gelesenen Datum/Zeit
LZ_EXIT	Ende-Behandlung beim Programmstop

Folgende Routinen werden nur innerhalb dieses Modules verwendet:

LZ_LZH_INIT	Initialisierung der Kommunikation zur LZH beim Programmstart
LZ_BWDGINIT	Initialisierung der Kommunikation zum BWD-GEFAHR beim Programmstart
LZ_BWDEINIT	Initialisierung der Kommunikation zum BWD-EINBRUCH beim Programmstart
LZ_ML_CFG	Initialisierung der Meldelinien gemäß TSS13a für die Kommunikation zum BWD-GEFAHR und -EINBRUCH beim Programmstart
LZ_VBP_INIT	Initialisierung der Behandlung der Verbindungs-Profile
LZ_IWA_INIT	Initialisierung der Behandlung der Istwert-Abfragen
LZ_VB_CONN	Verbindungsaufbau zur LZH und BWD's (CALL und LISTEN)
LZ_CAPIDATA	Datenübertragung mittels CAPI-Schnittstelle
LZ_X25_DATA	Datenübertragung mittels X.25-Schnittstelle (falls nicht deaktiviert)
LZ_VB_DISC	Verbindungsabbau zur LZH und BWD's
LZ_ANRUFCHK	Prüfung, ob eingehender Anruf angenommen werden soll
LZ_FND_SEND	Behandlung eines zu sendenden FND-Datagrammes
LZ_FND_RECV	Behandlung eines empfangenen FND-Datagrammes
LZ_FND_DATA	Bearbeitung eines empfangenen FND-Datagrammes
LZ_TSS_SEND	Behandlung eines zu sendenden TSS 13a-Datagrammes
LZ_TSS_RECV	Behandlung eines empfangenen TSS 13a-Datagrammes
LZ_TSS_DATA	Bearbeitung eines empfangenen TSS 13a-Datagrammes
LZ_SEND_RSP	Bevorzugtes Senden eines RSP-Datagrammes
LZ_VBP_CMD	Behandlung eines CMD-Datagrammes für ein Verbindungs-Profil
LZ_IWA_CMD	Behandlung eines CMD-Datagrammes für eine Istwert-Abfrage
LZ_IWA_USM	Erzeugen eines USM-Datagrammes für eine Istwert-Abfrage
LZ_VBP_POLL	zyklische Prüfung, ob ein Verbindungs-Profil aktiviert werden soll
LZ_VBP_LOAD	Verbindungs-Profil laden und aktivieren
LZ_VBP_NLZH	Berechnung des Zeitpunkts für den nächsten Verbindungs-Aufbau zur LZH
LZ_IWA_POLL	zyklische Prüfung, ob eine Istwert-Abfrage aktiviert oder bearbeitet werden soll

LZ\_IWA\_LOAD      Istwert-Abfrage laden und aktivieren

### 5.7.3. Aufrufstruktur

LZ\_INIT

    LZ\_LZH\_INIT

    LZ\_BWDGINIT

        LZ\_ML\_CFG

    LZ\_BWDEINIT

        LZ\_ML\_CFG

    LZ\_VBP\_INIT

        LZ\_VBP\_LOAD

            LZ\_VBP\_NLZH

    LZ\_IWA\_INIT

        LZ\_IWA\_LOAD

LZ\_RSP\_USM

    LZ\_SEND\_RSP

        LZ\_FND\_SEND

LZ\_POLL

    LZ\_VB\_CONN

        LZ\_VBP\_NLZH

        LZ\_ANRUFCHK

    LZ\_CAPIDATA

        LZ\_ANRUFCHK

        LZ\_FND\_RECV

        LZ\_FND\_DATA

		LZ_SEND_RSP	
			LZ_FND_SEND
		LZ_FND_SEND	
		LZ_VBP_CMD	
		LZ_VBP_NLZH	
		LZ_IWA_CMD	
			LZ_IWA_LOAD
		LZ_IWA_USM	
	LZ_TSS_RECV		
	LZ_TSS_DATA		
		LZ_TSS_SEND	
LZ_X25_DATA			
	LZ_FND_RECV		
	LZ_FND_DATA		
		LZ_SEND_RSP	
			LZ_FND_SEND
		LZ_FND_SEND	
		LZ_VBP_CMD	
		LZ_VBP_NLZH	
		LZ_IWA_CMD	
			LZ_IWA_LOAD
		LZ_IWA_USM	
LZ_VB_DISC			
	LZ_ANRUFCHK		
LZ_VBP_POLL			
	LZ_VBP_LOAD		
		LZ_VBP_NLZH	
LZ_IWA_POLL			
	LZ_RSP_USM		



LZ\_SEND\_RSP

LZ\_FND\_SEND

LZ\_STAT

LZ\_ZA\_TIME

LZ\_EXIT

LZ\_VB\_DISC

LZ\_ANRUFCHK

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

5.7.4. Modulbeschreibungen

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98  */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   LZ_INIT ( )           */
/*****
/*   Beschreibung :
/*   Diese Routine führt die LZ-spezifischen Initialisierungen
/*   nach dem Starten des Hauptprogrammes durch :
/*   - Initialisierungen für die Zwischenspeicherung und sofortige
/*   Ausgabe an LZH und BWD's.
/*   Aufgrund der Abfrage aller DP durch IZ_INIT mit lz_subid = 0
/*   werden nach Neustart ggf. alle DP's an die LZH bzw. BWD's
/*   gemeldet.
/*   - Initialisierungen der Verbindungsprofile
/*   - Initialisierungen der Istwertabfragen
/*   - Initialisierungen der CAPI-Schnittstelle
/*   - Initialisierungen der X.25-Schnittstelle
/*
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt-
/*   stelle werden abhängig vom in LZ_INCL.H definierten
/*   Schalter VB_ART_ISDN_X25 aus- oder eingeblendet.
/*
/*   aufgerufen von :           GAK_INIT
/*
/*   ruft auf      :           LZ_LZH_INIT, LZ_BWDGINIT, LZ_BWDEINIT,
/*                           LZ_VBP_INIT, LZ_IWA_INIT,
/*                           CAPI_INIT, CAPI_GET_VERSION,
/*                           CAPI_REGISTER, CAPI_ERROR, X25INIT,
/*                           X25ALLOC, X25ERROR, UT_ZA_GETZA,
/*                           UT_ER_GETER, UT_DP_FINDFIRST,
/*                           UT_DP_FINDNEXT, UT_UCGETBIT, SY_CALLOC
/*
/*   I/O-Kanäle   :
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   *** KEINE ***
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   int      LZ_INIT
/*           Returnkode der Funktion
/*           0=FALSE: KEIN Fehler
/*           -1n: Fehler bei allg. INIT
/*           -2n: Fehler bei LZH_INIT
/*           -3n: Fehler bei BWDGINIT
/*           -4n: Fehler bei BWDEINIT
/*           -5n: Fehler bei VBP_INIT
/*           -6n: Fehler bei IWA_INIT
/*           -7n: Fehler bei ISDN-Init.
/*                 (X.25 oder CAPI)
/*           >0: Zeilennummer aus der Konf.-
/*                 Datei, in der ein Fehler
/*                 erkannt wurde
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   *** KEINE ***
/*****
/*           Autor      Datum      Version   Kurzzeichen
/*           -----
/*   Original :   W. Fries   01.05.97   1.1
/*   Änderung:
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   RSP_USM.C             */
/*   Aufruf   :   LZ_RSP_USM ( cDatagramm ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine bearbeitet das FND-Datagramm "cDatagramm", indem */
/*   es abhängig vom DP und dem zugehörigen MFlag zwischengepuffert */
/*   oder zur direkten Ausgabe in die Warteschlange für die LZH oder */
/*   einen BWD eingereicht wird. Die Routine "LZ_POLL" übernimmt dann */
/*   die weitere Bearbeitung. Da sich der GA-Knoten gegenüber der LZH */
/*   wie eine IZ verhält, sind nur RSP- oder USM-Datagramme zulässig. */
/*   FND-Datagramme, deren Datenpunkt nicht in der DP-Struktur */
/*   vermerkt ist (z.B. Fault-Manager, Karteipunkte, ...) u.a., */
/*   werden direkt an die LZH ausgegeben. */
/*   RSP-Datagramme, die einer internen Abfrage des GA-Knotens oder */
/*   einem Schaltbefehl aus dem SP-Modul entstammen, werden in ein */
/*   USM-Datagramm umgewandelt, bevor sie zur direkten Ausgabe an die */
/*   LZH in die zugehörige Warteschlange eingereicht werden. */
/*   RSP-Datagramme, die einem CMD-Datagramm zugeordnet sind, das */
/*   von LZ-Modul erzeugt wurde, und auf das das LZ-Modul wartet, */
/*   wird unter Umgehung der Warteschlange direkt ausgegeben. */
/*   */
/*   aufgerufen von :           GAK_IIZ, GAK_ILZ, LZ_IWA_POLL */
/*   */
/*   ruft auf      :           LZ_SEND_RSP, UT_DP_FINDFIRST, */
/*                       UT_ER_GETER, UT_DP_OBJID, UT_RSP2USM, */
/*                       UT_WSL_WRIT, UT_UCSETBIT, diverse Makros */
/*                       FND... und SET_FND..., SY_F2IEEE */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   char     cDatagramm     */           RSP- bzw. USM-Datagramm */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   int      LZ_RSP_USM     */           Returnkode der Funktion */
/*                                           0=FALSE: KEIN Fehler */
/*                                           -1: OPEN-Fehler */
/*                                           -2: READ-Fehler */
/*                                           -3: WRITE-Fehler */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:                               */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   POLL.C                */
/*   Aufruf   :   LZ_POLL ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine prüft, ob ein Verbindungs-Aufbau (CALL / LISTEN) */
/*   zur LZH bzw. einem BWD durchgeführt werden muß. Für eine aktive */
/*   Verbindung werden die Datagramme entsprechend dem Protokoll */
/*   übertragen und ggf. die Verbindung abgebaut.                   */
/*   Als Ergebnis wird ggf. der Zeiger auf ein FND-Datagramm gemäß */
/*   Version 1.0 und der Ergänzungen (z.B. Karteipunkte) geliefert. */
/*   Dabei sind nur CMD-Datagramme zulässig; ACK-Datagramme werden */
/*   intern ignoriert, da sie für die DDC-IZ's im IZ-Modul erzeugt */
/*   werden und in den anderen Modulen nicht verwendet werden.     */
/*   */
/*   Zusätzlich wird geprüft, ob eine Istwert-Abfrage zyklisch     */
/*   gestartet werden soll.                                         */
/*   */
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt- */
/*   stelle werden abhängig vom in LZ_INCL.H definierten           */
/*   Schalter VB_ART_ISDN_X25 aus- oder eingeblendet.             */
/*   */
/*   aufgerufen von :           GAK_ILZ                               */
/*   */
/*   ruft auf      :           LZ_VB_CONN, LZ_CAPIDATA, LZ_X25_DATA, */
/*                       LZ_VB_DISC, LZ_VBP_POLL, LZ_IWA_POLL,     */
/*                       UT_ER_GETER, UT_DT_ADD                     */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                                           */
/*   -----                                                       */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   */
/*   *** KEINE ***                                               */
/*   -----                                                       */
/*   Ausgabe Parameter:                                           */
/*   -----                                                       */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   */
/*   char * LZ_POLL                Zeiger auf das CMD-Datagramm   */
/*                                   SONST: NULL-Zeiger             */
/*   -----                                                       */
/*   Transiente Parameter :                                       */
/*   -----                                                       */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   */
/*   *** KEINE ***                                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen                */
/*   -----                                                       */
/* Original :   W. Fries    01.05.97   1.1                       */
/* Änderung:   W. Fries    01.11.97   1.1a   WF1                 */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen     Stand     :   01.10.98    */
/*****
/*   Datei     :   STAT.C               */
/*   Aufruf    :   LZ_STAT ( )         */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert den LZ-internen Status als Ergebnis. */
/*                                           */
/*   aufgerufen von :       GAK_SYSCHK   */
/*                                           */
/*   ruft auf      :       *** KEINE *** */
/*                                           */
/*   I/O-Kanäle   :                       */
/*                                           */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   uns.long LZ_STAT          LZ-Status bitweise kodiert */
/*                               ( vgl. LZ_INCL.H) */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*****
/*           Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/* Original :   W. Fries      01.05.97      1.1 */
/* Änderung: */
/*           ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   ZA_TIME.C              */
/*   Aufruf   :   LZ_ZA_TIME ( )        */
/*****
/*   Beschreibung :                      */
/*   Diese Routine liefert die LZ-Systemzeit für den Zeitabgleich, */
/*   falls diese von VB_CONN bereitgestellt wurde.                */
/*                                                                */
/*   aufgerufen von :           GAK_SYSCHK                               */
/*                                                                */
/*   ruft auf      :           UT_ZA_GETZA, UT_FNDNX2DT                */
/*                                                                */
/*   I/O-Kanäle   :                      */
/*                                                                */
/*****
/*   Eingabe Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*                                                                */
/*   *** KEINE ***                      */
/*                                                                */
/* -----                      */
/*   Ausgabe Parameter:                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*                                                                */
/*   uns.long LZ_ZA_TIME           LZ-Systemzeit im DOS-Format      */
/*                                                                */
/*                                                                */
/* -----                      */
/*   Transiente Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*                                                                */
/*   *** KEINE ***                      */
/*                                                                */
/*****
/*           Autor      Datum      Version      Kurzzeichen      */
/*           -----                      */
/* Original :   W. Fries      01.05.97      1.1                      */
/* Änderung:                      */
/*                                                                */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   EXIT.C                 */
/*   Aufruf   :   LZ_EXIT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die LZ-spezifischen ENDE-Behandlungen */
/*   vor dem Stoppen des Hauptprogrammes durch :                */
/*   - Schließen der Warteschlange für die direkte Ausgabe an die */
/*     LZH, BWD_G und BWD_E.                                     */
/*   - Löschen bereits übertragener Datensätze aus den Dateien zum */
/*     Zwischenpuffern für die Übertragung zur LZH.              */
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt- */
/*             stelle werden abhängig vom in LZ_INCL.H definierten */
/*             Schalter VB_ART_ISDN_X25 aus- oder eingeblendet.  */
/*   aufgerufen von :           GAK_EXIT                               */
/*   ruft auf :                 LZ_VB_DISC, CAPI_RELEASE, CAPI_EXIT, */
/*                               X25EXIT, UT_LOOPTIME, UT_WSL_EXIT,  */
/*                               UT_OBJ_GETI, UT_ER_GETER, SY_FREE    */
/*   I/O-Kanäle :                                                         */
/*****
/*   Eingabe Parameter :                                               */
/*   -----                                                         */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   -----                                                         */
/*   *** KEINE ***                                                    */
/*   -----                                                         */
/*   Ausgabe Parameter:                                              */
/*   -----                                                         */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   -----                                                         */
/*   int      LZ_EXIT        -       Returnkode der Funktion        */
/*                                               0=FALSE: KEIN Fehler      */
/*                                               -1: OPEN-Fehler        */
/*                                               -2: READ-Fehler        */
/*                                               -3: WRITE-Fehler       */
/*                                               -4: FREE-Fehler        */
/*                                               -5: unzulässige ML-Zeile */
/*                                               -6: WSL_EXIT-Fehler     */
/*                                               >0: Nummer der Zeile, in der */
/*                                               ein Fehler erkannt wurde */
/*   -----                                                         */
/*   Transiente Parameter :                                           */
/*   -----                                                         */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   -----                                                         */
/*   *** KEINE ***                                                    */
/*****
/*   Autor      Datum      Version  Kurzzeichen                       */
/*   -----                                                         */
/*   Original :   W. Fries    01.05.97    1.1                          */
/*   Änderung:                                         */
/*   -----                                                         */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      LZ Leitzentralen      Stand     :   01.10.98  */
/*****
/*      Datei      :      LZH_INIT.C            */
/*      Aufruf    :      LZ_LZH_INIT ( )       */
/*****
/*      Beschreibung :                               */
/*      Diese Routine führt die LZ-spezifischen Initialisierungen */
/*      für die LZH nach dem Starten des Hauptprogrammes durch :  */
/*      - Warteschlange für die Eingaben von der LZH                */
/*      - Warteschlange für die sofortige Ausgabe an LZH            */
/*      */
/*      aufgerufen von :          LZ_INIT        */
/*      */
/*      ruft auf      :          UT_WSL_INIT    */
/*      */
/*      I/O-Kanäle   :                               */
/*      */
/*****
/*      Eingabe Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung      */
/*      */
/*      *** KEINE ***                               */
/*      */
/*      -----                               */
/*      Ausgabe Parameter:                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung      */
/*      */
/*      int      LZ_LZH_INIT      Returncode der Funktion */
/*      */
/*      0=FALSE: KEIN Fehler                        */
/*      -1: OPEN-Fehler                             */
/*      -2: READ-Fehler                             */
/*      -3: WRITE-Fehler                            */
/*      -4: ALLOC-Fehler                            */
/*      -5: unzulässige IS-Zeile                    */
/*      -7: WSL_INIT-Fehler                         */
/*      >0: Nummer der Zeile, in der                */
/*      Datei, in der ein Fehler                    */
/*      erkannt wurde                               */
/*      */
/*      -----                               */
/*      Transiente Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung      */
/*      */
/*      *** KEINE ***                               */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen   */
/*      -----                               */
/*      Original :      W. Fries      01.05.97      1.1      */
/*      Änderung:                               */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      LZ Leitzentralen     Stand     :   01.10.98  */
/*****
/*      Datei       :      BWDEINIT.C           */
/*      Aufruf      :      LZ_BWDEINIT ( )     */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die LZ-spezifischen Initialisierungen */
/*      für den BWD-EINBRUCH nach dem Starten des Hauptprogrammes durch:*/
/*      - Warteschlange für die Eingaben von dem BWD                */
/*      - Warteschlange für die sofortige Ausgabe an den BWD        */
/*      */
/*      aufgerufen von :      LZ_INIT          */
/*      */
/*      ruft auf      :      LZ_ML_CFG, UT_WSL_INIT, UT_WSL_WRIT    */
/*      */
/*      I/O-Kanäle   :                        */
/*      */
/*****
/*      Eingabe Parameter :                    */
/*      -----                    */
/*      Typ      Name      Dim.      Beschreibung                    */
/*      */
/*      *** KEINE ***                    */
/*      */
/*      -----                    */
/*      Ausgabe Parameter:                    */
/*      -----                    */
/*      Typ      Name      Dim.      Beschreibung                    */
/*      int      LZ_BWDEINIT          Returnkode der Funktion      */
/*      0=FALSE: KEIN Fehler              */
/*      -1: OPEN-Fehler                   */
/*      -2: READ-Fehler                   */
/*      -3: WRITE-Fehler                   */
/*      -4: ALLOC-Fehler                   */
/*      -5: ML_CFG-Fehler                  */
/*      -7: WSL_INIT-Fehler                */
/*      >0: Zeilennummer aus der Konf.-   */
/*      Datei, in der ein Fehler          */
/*      erkannt wurde                      */
/*      */
/*      -----                    */
/*      Transiente Parameter :                */
/*      -----                    */
/*      Typ      Name      Dim.      Beschreibung                    */
/*      */
/*      *** KEINE ***                    */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen            */
/*      -----                    */
/*      Original :   W. Fries      01.05.97      1.1                */
/*      Änderung:                                     */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      LZ Leitzentralen      Stand     :   01.10.98  */
/*****
/*      Datei      :      BWDGINIT.C           */
/*      Aufruf     :      LZ_BWDGINIT ( )      */
/*****
/*      Beschreibung :                       */
/*      Diese Routine führt die LZ-spezifischen Initialisierungen */
/*      für den BWD-GEFAHR nach dem Starten des Hauptprogrammes durch : */
/*      - Warteschlange für die Eingaben von dem BWD                */
/*      - Warteschlange für die sofortige Ausgabe an den BWD        */
/*      */
/*      aufgerufen von :          LZ_INIT      */
/*      */
/*      ruft auf       :          LZ_ML_CFG, UT_WSL_INIT, UT_WSL_WRIT */
/*      */
/*      I/O-Kanäle    :                       */
/*      */
/*****
/*      Eingabe Parameter :                   */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung          */
/*      */
/*      *** KEINE ***                          */
/*      */
/*      -----                               */
/*      Ausgabe Parameter:                   */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung          */
/*      */
/*      int      LZ_BWDGINIT          Returnkode der Funktion */
/*      */
/*      0=FALSE: KEIN Fehler                    */
/*      -1: OPEN-Fehler                          */
/*      -2: READ-Fehler                          */
/*      -3: WRITE-Fehler                         */
/*      -4: ALLOC-Fehler                         */
/*      -5: ML_CFG-Fehler                        */
/*      -7: WSL_INIT-Fehler                      */
/*      >0: Zeilennummer aus der Konf.- */
/*      Datei, in der ein Fehler          */
/*      erkannt wurde                          */
/*      */
/*      -----                               */
/*      Transiente Parameter :               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung          */
/*      */
/*      *** KEINE ***                          */
/*      */
/*****
/*      Autor      Datum          Version   Kurzzeichen   */
/*      -----                               */
/*      Original :   W. Fries      01.05.97    1.1          */
/*      Änderung:   */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei     :   ML_CFG.C              */
/*   Aufruf    :   LZ_ML_CFG ( cBWD, ML, uAnz_ML ) */
/*****
/*   Beschreibung :                       */
/*   Diese Routine lädt für den BWD "cBWD" die spezifischen Parameter*
/*   für die Meldelinien "ML" mit maximal "uAnz_ML" Einträgen aus
/*   der Konfigurationsdatei.             */
/*                                           */
/*   aufgerufen von :   LZ_BWDGINIT, LZ_BWDEINIT */
/*                                           */
/*   ruft auf      :   UT_NORM_STRING, UT_DP_FINDFIRST,
/*                   UT_DP_FINDNEXT, UT_UCGETBIT */
/*                                           */
/*   I/O-Kanäle   :                       */
/*                                           */
/*****
/*   Eingabe Parameter :                   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   char     cBWD      *          Typ des BWD */
/*                                           */
/*           G: GEFAHR */
/*           E: EINBRUCH */
/*                                           */
/*   LZ_ML*   ML        *          Zeiger auf die ML-Struktur für
/*                                           */
/*           den BWD vom Typ "cBWD" */
/*                                           */
/*   unsigned uAnz_ML   *          max. Anzahl der Meldelinien */
/*                                           */
/*   ----- */
/*   Ausgabe Parameter:                   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   int      LZ_ML_CFG *          Returnkode der Funktion
/*                                           */
/*           0=FALSE: KEIN Fehler */
/*           -1: OPEN-Fehler */
/*           -2: READ-Fehler */
/*           -3: WRITE-Fehler */
/*           -4: ALLOC-Fehler */
/*           -5: unzulässige ML-Zeile */
/*           -6: fehlende ML-Zeile für die
/*           Zuord. eines DP zum BWD_G */
/*           -7: fehlende ML-Zeile für die
/*           Zuord. eines DP zum BWD_E */
/*           >0: Nummer der Zeile, in der
/*           ein Fehler erkannt wurde */
/*                                           */
/*   ----- */
/*   Transiente Parameter :               */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*                                           */
/*****
/*           Autor      Datum      Version      Kurzzeichen */
/*           ----- */
/* Original :   W. Fries      01.05.97      1.1 */
/* Änderung: */
/*                                           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      LZ Leitzentralen     Stand     :   01.10.98  */
/*****
/*      Datei      :      VBP_INIT.C           */
/*      Aufruf     :      LZ_VBP_INIT ( )      */
/*****
/*      Beschreibung :                               */
/*      Diese Routine führt die LZ-spezifischen Initialisierungen */
/*      für die Verbindungs-Kontrolle zur LZH und den BWD's nach dem */
/*      Starten des Hauptprogrammes durch :                               */
/*      - Positionieren auf das aktuelle Verbindungs-Profil und des */
/*      Zeitpunkts zur Aktivierung des nächsten für jedes Objekt. */
/*      */
/*      aufgerufen von :          LZ_INIT      */
/*      */
/*      ruft auf      :          LZ_VBP_LOAD  */
/*      */
/*      I/O-Kanäle   :                               */
/*      */
/*****
/*      Eingabe Parameter :                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      */
/*      ----- */
/*      Ausgabe Parameter:                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung */
/*      */
/*      int      LZ_VBP_INIT      Returnkode der Funktion */
/*      */
/*      0=FALSE: KEIN Fehler */
/*      -1: OPEN-Fehler */
/*      -2: READ-Fehler */
/*      -3: WRITE-Fehler */
/*      -4: ALLOC-Fehler */
/*      >0: Nummer der Zeile, in der */
/*      Datei, in der ein Fehler */
/*      erkannt wurde */
/*      */
/*      ----- */
/*      Transiente Parameter :                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1      */
/*      Änderung:      */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname   :      LZ Leitzentralen     Stand     :   01.10.98  */
/*****
/*      Datei      :      IWA_INIT.C           */
/*      Aufruf     :      LZ_IWA_INIT ( )      */
/*****
/*      Beschreibung :                       */
/*      Diese Routine führt die LZ-spezifischen Initialisierungen */
/*      für die Istwert-Abfragen durch die LZH nach dem Starten des */
/*      Hauptprogrammes durch :             */
/*      - Laden der hinterlegten Istwert-Abfragen                    */
/*      aufgerufen von :           LZ_INIT          */
/*      ruft auf      :           LZ_IWA_LOAD, SY_FI_FINDFIRST, */
/*                      SY_FI_FINDNEXT             */
/*      I/O-Kanäle   :                       */
/*      */
/*****
/*      Eingabe Parameter :                   */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung                    */
/*      *** KEINE ***                               */
/*      -----                               */
/*      Ausgabe Parameter:                   */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung                    */
/*      int      LZ_IWA_INIT          Returnkode der Funktion      */
/*                                          0=FALSE: KEIN Fehler      */
/*                                          -1: OPEN-Fehler          */
/*                                          -2: READ-Fehler          */
/*                                          -3: WRITE-Fehler         */
/*                                          -4: ALLOC-Fehler         */
/*                                          -5: unzulässige IWA-Datei oder */
/*                                          Zeile in der Datei      */
/*                                          -6: Max. Anzahl der IWA erreicht*/
/*      -----                               */
/*      Transiente Parameter :               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung                    */
/*      *** KEINE ***                               */
/*      -----                               */
/*****
/*      Autor      Datum      Version   Kurzzeichen   */
/*      -----                               */
/*      Original :   W. Fries      01.05.97      1.1          */
/*      Änderung:                                     */
/*      -----                               */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98  */
/*****
/*   Datei    :   VB_CONN.C             */
/*   Aufruf   :   LZ_VB_CONN ( )       */
/*****
/*   Beschreibung :
/*   Diese Routine startet den Verbindungs-Aufbau (CONNECT) zur LZH
/*   und den BWD's wie folgt :
/*   - Auf der CAPI-Schnittstelle wird LISTEN_REQ gestartet
/*   - zyklisch wird geprüft, ob ein Anruf eingeht. Falls JA wird in
/*   die "Called Party Number" mit der eigenen Rufnummer (ER-Zeile
/*   aus der Konfigurationsdatei) und ggf. die "Calling Party
/*   Number" gegen die in ANRUFER.CHK hinterlegten Rufnummern
/*   geprüft. Bei Übereinstimmung wird die Verbindung entsprechend
/*   dem hinterlegten Protokoll aufgebaut.
/*   Falls NEIN wird geprüft, ob eine Verbindung durch den GA-
/*   Knoten aufgebaut werden soll und dieses entsprechend dem
/*   aktuellen Verbindungs-Profil für den anzurufenden LZH bzw.
/*   BWD.
/*
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt-
/*   stelle werden abhängig vom in LZ_INCL.H definierten
/*   Schalter VB_ART_ISDN_X25 aus- oder eingeblendet.
/*
/*   aufgerufen von :           LZ_POLL
/*
/*   ruft auf           :           LZ_VBP_NLZH, LZ_ANRUFCHK,
/*                               CAPI_SEND, CAPI_RECV, CAPI_ERROR,
/*                               CAPI_STD_RESP,
/*                               X25_DIAL, X25XCALL, X25XLISTEN,
/*                               X25XHANGUP, X25DONE, X25CANCEL, X25ERROR*/
/*                               UT_DT_ADD, UT_DP_FINDFIRST, UT_WSL_READ,*/
/*                               UT_ER_GETER, UT_ZA_GETZA, UT_NORM_STRING*/
/*                               UT_FNDNX2DT, UT_OBJ_GETI, UT_ESC_SPRINTF*/
/*
/*   I/O-Kanäle       :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   *** KEINE ***
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   int      LZ_VB_CONN    Returnkode der Funktion
/*                               0=FALSE: KEIN Fehler
/*                               -1: OPEN-Fehler
/*                               -2: READ/SCANF-Fehler
/*                               -3: WRITE-Fehler
/*                               -4: ALLOC-Fehler
/*                               -5: FND-Fehler
/*                               -6: CAPI-Fehler
/*                               -7: X25-Fehler (EICON-Bibl.)
/*                               -8: unerwartetes DATA_IND
/*                               -9: unerwartetes DISC_IND
/*                               -10: nicht berechtigter Anrufer
/*
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   -----
/*   *** KEINE ***
/*****
/*
/*   Autor      Datum      Version   Kurzzeichen
/*   -----
/*   Original :   W. Fries   01.05.97   1.1
/*   Änderung :   W. Fries   01.01.98   1.1b   WF1
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   CAPIDATA.C           */
/*   Aufruf   :   LZ_CAPIDATA ( )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine behandelt den Datenaustausch entsprechend dem */
/*   Verbindungs-Protokoll über die CAPI-Schnittstelle.         */
/*   */
/*   aufgerufen von :   LZ_POLL        */
/*   */
/*   ruft auf      :   LZ_ANRUFCHK, LZ_FND_RECV, LZ_FND_DATA, */
/*                   LZ_TSS_RECV, LZ_TSS_DATA, */
/*                   CAPI_SEND, CAPI_RECV, CAPI_ERROR, */
/*                   CAPI_STD_RESP, UT_DT_ADD */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   char *   LZ_CAPIDATA   */
/*                   Zeiger auf ein FND-Datagramm, */
/*                   das an ein anderes Modul (z.B. */
/*                   IZ, SP) weitergegeben werden */
/*                   soll. Das Ziel ist durch die */
/*                   LZ_SUBID gekennzeichnet. */
/*                   SONST: NULL-Zeiger */
/*   */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:   W. Fries   01.11.97   1.1a         WF1 */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   X25_DATA.C             */
/*   Aufruf   :   LZ_X25_DATA ( )       */
/*****
/*   Beschreibung :                       */
/*   Diese Routine behandelt den Datenaustausch entsprechend dem */
/*   Verbindungs-Protokoll über die X25-Schnittstelle.           */
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt- */
/*   stelle werden abhängig vom in LZ_INCL.H definierten */
/*   Schalter VB_ART_ISDN_X25 aus- oder eingeblendet, d.h. */
/*   diese Routine wird von LZ_POLL nur bedingt aufgerufen.*/
/*   */
/*   aufgerufen von :           LZ_POLL */
/*   ruft auf      :           LZ_FND_RECV, LZ_FND_DATA, */
/*                   X25SEND, X25RECV, X25DONE, X25CANCEL, */
/*                   X25ERROR, CAPI_RECV, CAPI_STD_RESP, */
/*                   UT_DT_ADD */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   char *  LZ_X25_DATA    *       Zeiger auf ein FND-Datagramm, */
/*                                     das an ein anderes Modul (z.B. */
/*                                     IZ, SP) weitergegeben werden */
/*                                     soll. Das Ziel ist durch die */
/*                                     LZ_SUBID gekennzeichnet. */
/*                                     SONST: NULL-Zeiger */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1      */
/*   Änderung:                                     */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98   */
/*****
/*   Datei    :   VB_DISC.C              */
/*   Aufruf   :   LZ_VB_DISC ( )        */
/*****
/*   Beschreibung :                      */
/*   Diese Routine startet den Verbindungs-Abbau (DISCONNECT) zur */
/*   LZH und den BWD's wie folgt :      */
/*   - Wurde ein unerwarteter Verbindungs-Abbau vom LZ-Partner    */
/*     veranlaßt (RECV_Flag == USDISC), so wird dieser behandelt.  */
/*   - Sonst wird der Verbindungs-Abbau durch den GA-Knoten verlaßt, */
/*     falls dieses noch nicht geschehen ist.                      */
/*   - Der laufende Verbindungs-Abbau wird durchgeführt und auf    */
/*     einen Timeout hin überwacht, nachdem der Abbau erzwungen wird.*/
/*   */
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt- */
/*            stelle werden abhängig vom in LZ_INCL.H definierten  */
/*            Schalter VB_ART_ISDN_X25 aus- oder eingeblendet.     */
/*   */
/*   aufgerufen von :           LZ_POLL; LZ_EXIT                    */
/*   */
/*   ruft auf      :           LZ_ANRUFCHK, CAPI_STD_RESP,          */
/*                       CAPI_SEND, CAPI_RECV, CAPI_ERROR,         */
/*                       X25XHANGUP, X25DONE, X25CANCEL, X25ERROR* */
/*                       X25XHANGUPCONFIRM, UT_DT_ADD              */
/*   */
/*   I/O-Kanäle   :                      */
/*****
/*   Eingabe Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*   */
/*   *** KEINE ***                      */
/*   -----                      */
/*   Ausgabe Parameter:                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*   */
/*   int      LZ_VB_DISC      Returnkode der Funktion              */
/*                       0=FALSE: KEIN Fehler                      */
/*                       -6: CAPI-Fehler                            */
/*                       -7: X25-Fehler (EICON-Bibl.)              */
/*   -----                      */
/*   Transiente Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*   */
/*   *** KEINE ***                      */
/*****
/*   Autor      Datum      Version   Kurzzeichen                      */
/*   -----                      */
/*   Original :   W. Fries    01.05.97    1.1                        */
/*   Änderung:                                     */
/*   -----                      */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei     :   ANRUFCHK.C           */
/*   Aufruf    :   LZ_ANRUFCHK ( cCONN_IND, cANRUFCHK_Buf )    */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft für das CONNECT_IND in "cCONN_IND", ob der */
/*   Anruf angenommen werden soll wie folgt :   */
/*   - Die "Called Party Number" mit der eigenen Rufnummer (ER-Zeile */
/*     aus der Konfigurationsdatei) und ggf. die "Calling Party */
/*     Number" gegen die in ANRUFER.CHK hinterlegten Rufnummern */
/*     geprüft. */
/*   - Bei Übereinstimmung wird noch geprüft, ob die Kennung in der */
/*     Zeile aus ANRUFER.CHK dem LZ-Modul zugeordnet ist (L, G und E) */
/*     und nicht bereits eine Verbindung mit einem höheren oder */
/*     gleichen Priorität aktiv ist ("E" hat höchste Priorität, dann */
/*     "G", gefolgt von "L"). */
/*   Soll der Anruf angenommen werden, so wird die aus ANRUFER.CHK */
/*   gelesene Zeile im Puffer "cANRUFCHK" und der REJECT-Kode für */
/*   ACCEPT als Ergebnis geliefert. Sonst wird der REJECT-Kode für */
/*   IGNORE oder REJECT als Ergebnis geliefert. */
/*   */
/*   Hinweis: Die Programmteile zur Unterstützung der X.25-Schnitt- */
/*             stelle werden abhängig vom in LZ_INCL.H definierten */
/*             Schalter VB_ART_ISDN_X25 aus- oder eingeblendet. */
/*   */
/*   aufgerufen von :   LZ_VB_CONN, LZ_CAPIDATA, LZ_VB_DISC    */
/*   ruft auf       :   UT_NORM_STRING                          */
/*   I/O-Kanäle     :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   char *   cCONN_IND      */     CONNECT_IND-Parameter */
/*   char *   cANRUFCHK_Buf  */     Puffer, in den die Zeile aus */
/*                                     ANRUFER.CHK gelesen werden soll */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   unsigned LZ_ANRUFCHK    */     REJECT-Kode gemäß CAPI 2.0 */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen          */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1                */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt   :      GA - Knoten           Version   :   1.2      */
/*      Modulname :      LZ Leitzentralen      Stand     :   01.10.98  */
/*****
/*      Datei     :      FND_DATA.C           */
/*      Aufruf    :      LZ_FND_DATA ( )      */
/*****
/*      Beschreibung :                       */
/*      Diese Routine behandelt die empfangenen FND-Datagramme */
/*      */
/*      aufgerufen von :      LZ_X25_DATA, LZ_CAPIDATA           */
/*      */
/*      ruft auf      :      LZ_SEND_RSP, LZ_FND_SEND, LZ_VBP_CMD, */
/*      LZ_VBP_NLZH, LZ_IWA_CMD, LZ_IWA_USM, */
/*      UT_OBJ_GETI, UT_WSL_READ, UT_FND_ERROR, */
/*      UT_UDATETAG, UT_TIMESEK, UT_FND_TIME, */
/*      diverse Makros FND... und SET_FND..., */
/*      SY_F2IEEE */
/*      */
/*      I/O-Kanäle   :                       */
/*      */
/*****
/*      Eingabe Parameter :                   */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                   */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      char * LZ_FND_DATA      Zeiger auf ein FND-Datagramm, */
/*      das an ein anderes Modul (z.B. */
/*      IZ, SP) weitergegeben werden */
/*      soll. Das Ziel ist durch die */
/*      LZ_SUBID gekennzeichnet. */
/*      SONST: NULL-Zeiger */
/*      ----- */
/*      Transiente Parameter :               */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1      */
/*      Änderung:      W. Fries      01.11.97      1.1a      WF1      */
/*      ----- */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   FND_RECV.C           */
/*   Aufruf   :   LZ_FND_RECV ( cDatagramm, iLen )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine behandelt das empf. FND-Datagramm "cDatagramm" */
/*   der Länge "iLen".           */
/*           */
/*   aufgerufen von :           LZ_X25_DATA, LZ_CAPIDATA           */
/*           */
/*   ruft auf      :           UT_DT_ADD, UT_WSL_WRIT, diverse Makros */
/*   FND... und SET_FND...           */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   char     cDatagramm           empfangenes FND-Datagramm           */
/*   -----           */
/*   int      iLen           Länge des FND-Datagrammes           */
/*   (Nur aus Kompatibilität zu and.           */
/*   Protokollen, verwendet wird           */
/*   "APCI->len" )           */
/*   -----           */
/*   - - - - -           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   int      LZ_FND_RECV           Returnkode der Funktion           */
/*   0=FALSE: KEIN Fehler           */
/*   -1: OPEN-Fehler           */
/*   -2: READ/SCANF-Fehler           */
/*   -3: WRITE-Fehler           */
/*   -4: ALLOC-Fehler           */
/*   -5: FND-Fehler           */
/*   -6: CAPI-Fehler           */
/*   -7: X25-Fehler (EICON-Bibl.)           */
/*   -8: unerwartetes DATA_IND           */
/*   -9: unerwartetes DISC_IND           */
/*   -10: nicht berechtigter Anrufer           */
/*   -----           */
/*   - - - - -           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*****
/*           Autor      Datum      Version      Kurzzeichen           */
/*   -----           */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:           */
/*           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei     :   FND_SEND.C           */
/*   Aufruf    :   LZ_FND_SEND ( cDatagramm, iLen )           */
/*****
/*   Beschreibung :   */
/*   Diese Routine behandelt das zu sendende FND-Datagramme der */
/*   Länge "iLen".                                           */
/*   */
/*   aufgerufen von :   LZ_SEND_RSP, LZ_FND_DATA           */
/*   */
/*   ruft auf      :   FND_... und SET_FND_...           */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   char     cDatagramm           zu sendendes FND-Datagramm */
/*   int      iLen              Länge des FND-Datagrammes */
/*                               (Nur aus Kompatibilität zu and. */
/*                               Protokollen, verwendet wird */
/*                               "APCI->len" )           */
/*   -----           */
/*   Ausgabe Parameter:   */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   int      LZ_FND_SEND           Returnkode der Funktion */
/*                               0=FALSE: KEIN Fehler */
/*   -----           */
/*   Transiente Parameter :   */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   *** KEINE ***           */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----           */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei     :   IWA_CMD.C             */
/*   Aufruf    :   LZ_IWA_CMD ( cCMD )   */
/*****
/*   Beschreibung :                       */
/*   Diese Routine vermerkt die in "cCMD" gelieferte Istwert- */
/*   Abfrage in einer zugehörigen Datei (Segmentierter Karteipunkt */
/*   gemäß LV unter 4.5 ff.). Ist der Aktivierungszeitpunkt früher */
/*   als der bisher als nächster vermerkte, so wird der neue */
/*   Zeitpunkt als nächster auszuführender vermerkt.           */
/*   Als Ergebnis wird der Zeiger auf das für "cCMD" zu liefernde */
/*   RSP-Datagramm geliefert.                                   */
/*   aufgerufen von :           LZ_FND_DATA                       */
/*   ruft auf      :           LZ_IWA_LOAD, UT_OBJ_GETI, UT_FNDERROR, */
/*                       UT_FNDREJECT, UT_DT2CI, UT_DP_FINDFIRST, */
/*                       UT_TAG_S2DT, diverse Makros FND... und */
/*                       SET_FND..., SY_IEEE2F                     */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   char*    cCMD          *       Zeiger auf das auszuwertende */
/*                                       CMD-Datagramm mit der neuen */
/*                                       Istwert-Abfrage           */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   char*    LZ_IWA_CMD    *       Zeiger auf das RSP-Datagramm   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version  Kurzzeichen         */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1             */
/*   Änderung:                                     */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   IWA_LOAD.C           */
/*   Aufruf   :   LZ_IWA_LOAD ( cDatei ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine lädt die in der IWA-Datei "cDatei" hinterlegten */
/*   Istwert-Abfragen in die IWA-Struktur                               */
/*   aufgerufen von :           LZ_IWA_INIT, LZ_IWA_CMD           */
/*   ruft auf      :           UT_CI2DT                           */
/*   I/O-Kanäle   :                               */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   char *   cDatei                Name der IWA-Datei, die in die */
/*   IWA-Struktur geladen werden soll */
/*   -----                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   int      LZ_IWA_LOAD          Returnkode der Funktion       */
/*   0=FALSE: KEIN Fehler           */
/*   -1: OPEN-Fehler                */
/*   -2: READ-Fehler                */
/*   -3: WRITE-Fehler               */
/*   -4: ALLOC-Fehler               */
/*   -5: unzulässige IWA-Datei oder */
/*   Zeile in der Datei             */
/*   -6: Max. Anzahl der IWA erreicht */
/*   -----                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung           */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version  Kurzzeichen          */
/*   -----                               */
/*   Original :   W. Fries    01.05.97   1.1              */
/*   Änderung:                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand     :   01.10.98  */
/*****
/*   Datei    :   IWA_POLL.C            */
/*   Aufruf   :   LZ_IWA_POLL ( )      */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft, ob eine Istwertabfrage gestartet werden */
/*   soll (Aktivierung einer IWA ist als ERSTER Start beim Laden */
/*   realisiert). Falls JA, wird dieses als OFFENE Schleifen über */
/*   alle Zeilen in der IWA-Datei und alle darin angegebenen Daten- */
/*   punkte (Vgl. Platzhalter !!) realisiert, d.h. jeder Aufruf von */
/*   LZ_IWA_POLL wird unterbrochen, sobald ein FND-Datagramm als */
/*   Ergebnis geliefert wird und setzt beim nächsten Aufruf mit der */
/*   Fortsetzung der unterbrochenen Schleife auf.                */
/*   */
/*   Als Ergebnis wird ggf. der Zeiger auf ein FND-Datagramm gemäß */
/*   Version 1.0 und der Ergänzungen (z.B. Karteipunkte) geliefert. */
/*   Dabei sind nur CMD-Datagramme zulässig, die an die DDC-IZ's */
/*   weitergeleitet werden.                                     */
/*   */
/*   Hinweis: Momentan wird der aktuelle Istwert aus der DP-Struktur* */
/*   gelesen und das zugehörige RSP-Datagramm direkt an */
/*   LZ_RSP_USM für die direkte Ausgabe an die LZH über- */
/*   geben. Diese Lösung entlastet das System (insbesondere* */
/*   die DDC-IZ's mit vielen Datenpunkten und langen */
/*   Antwortzeiten (z.B. Sauter FND) und liefert zusätz- */
/*   lich auch interne Zusatzinformationen (z.B. "Geber */
/*   gestört" nach Flimmerer. Die gesamte Programmstruktur */
/*   bleibt jedoch auch darauf ausgerichtet die IWA über */
/*   Lesezugriffe auf die DDC-IZ's zu realisieren.              */
/*   */
/*   aufgerufen von :   LZ_POLL                */
/*   ruft auf   :   LZ_RSP_USM, UT_CI2DT, UT_FNDNX2DT, */
/*                 UT_NORM_STRING, UT_DP_FINDFIRST, */
/*                 UT_DP_FINDNEXT, UT_DT_ADD, UT_FND_TIME, */
/*                 diverse Makros FND_... und SET_FND_..., */
/*                 SY_F2IEEE                    */
/*   I/O-Kanäle   :   */
/*****
/*   Eingabe Parameter :   */
/*   -----                */
/*   Typ      Name      Dim.   Beschreibung                */
/*   *** KEINE ***                */
/*   -----                */
/*   Ausgabe Parameter:   */
/*   -----                */
/*   Typ      Name      Dim.   Beschreibung                */
/*   char *   LZ_IWA_POLL           Zeiger auf das CMD-Datagramm */
/*                                     SONST: NULL-Zeiger                */
/*   -----                */
/*   Transiente Parameter :   */
/*   -----                */
/*   Typ      Name      Dim.   Beschreibung                */
/*   *** KEINE ***                */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                */
/*   Original :   W. Fries   01.05.97   1.1                */
/*   Änderung:                */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   IWA_USM.C              */
/*   Aufruf   :   LZ_IWA_USM ( cUSM )    */
/*****
/*   Beschreibung :                      */
/*   Diese Routine behandelt das Senden eines USM-Datagrammes, das */
/*   aufgrund einer Istwertabfrage erzeugt wurde. Die LZ_SUBID mit */
/*   der die Abfrage erfolgte, wird restauriert. Zusätzlich wird */
/*   der zugehörige Eintrag in der IWA-Struktur aktualisiert.      */
/*   Soll das modifizierte USM-Datagramm an die LZH gesendet werden, */
/*   wird es als Ergebnis geliefert.                                  */
/*                                                                    */
/*   aufgerufen von :           LZ_FND_DATA                          */
/*                                                                    */
/*   ruft auf      :           *** KEINE ***                          */
/*                                                                    */
/*   I/O-Kanäle   :                      */
/*                                                                    */
/*****
/*   Eingabe Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*   -----                      */
/*   *** KEINE ***                      */
/*   -----                      */
/*   Ausgabe Parameter:                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*   -----                      */
/*   char *   LZ_IWA_USM           Zeiger auf das USM-Datagramm, */
/*                                     das an die LZH gesendet werden */
/*                                     soll.                          */
/*                                     SONST: NULL-Zeiger              */
/*   -----                      */
/*   Transiente Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*   -----                      */
/*   char *   cUSM                Zeiger auf das USM-Datagramm, */
/*                                     das an die LZH gesendet werden */
/*                                     soll. (LZ_SUBID wird rest. !!) */
/*   -----                      */
/*****
/*   Autor      Datum      Version   Kurzzeichen                      */
/*   -----                      */
/*   Original :   W. Fries    01.05.97    1.1                          */
/*   Änderung:                      */
/*   -----                      */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   SEND_RSP.C           */
/*   Aufruf   :   LZ_SEND_RSP ( cDatagramm ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine sendet das FND-RSP-Datagramm "cDatagramm" unter */
/*   Umgehung der Warteschlange direkt an den LZ-Partner (BWD's bzw. */
/*   LZH), da dieser darauf wartet (vgl. Flag im "ullZ_STAT").      */
/*   Die vom LZ-Partner verwendete LZ_SUBID wird zuvor restauriert. */
/*                                                                    */
/*   aufgerufen von :           LZ_RSP_USM, LZ_FND_DATA           */
/*                                                                    */
/*   ruft auf      :           LZ_FND_SEND, UT_WSL_WRIT          */
/*                                                                    */
/*   I/O-Kanäle   :                               */
/*                                                                    */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung          */
/*   -----                               */
/*   char     cDatagramm          RSP-Datagramm          */
/*   -----                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung          */
/*   -----                               */
/*   int      LZ_SEND_RSP          Returncode der Funktion */
/*   -----                               */
/*   0=FALSE: KEIN Fehler          */
/*   -1: OPEN-Fehler              */
/*   -2: READ-Fehler              */
/*   -3: WRITE-Fehler             */
/*   -----                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name          Dim.   Beschreibung          */
/*   -----                               */
/*   *** KEINE ***                               */
/*   -----                               */
/*****
/*   Autor      Datum      Version  Kurzzeichen      */
/*   -----                               */
/*   Original :   W. Fries    01.05.97    1.1          */
/*   Änderung:                               */
/*   -----                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      LZ Leitzentralen     Stand     :   01.10.98  */
/*****
/*      Datei       :      TSS_DATA.C           */
/*      Aufruf      :      LZ_TSS_DATA ( )      */
/*****
/*      Beschreibung :                               */
/*      Diese Routine behandelt die empfangenen und zu sendenden */
/*      TSS13a-Datagramme                               */
/*      Hinweis: Seitens der BWD's sind mittels TSS 13a momentan nur */
/*      lesende Zugriffe erlaubt. Die lesenden Zugriffe werden */
/*      durch Lesen aus der DP-Struktur realisiert, d.h. es */
/*      wird KEIN FND-Datagramm zum Lesen aus der DDC-IZ */
/*      erzeugt.                                         */
/*      Die FND-Nomenklatur wird sinngemäß auf die TSS 13a- */
/*      Bezeichnungen angewandt (z.B. USM für Meldung, ACK für */
/*      Meldungs-Quittierung)                               */
/*      aufgerufen von :           LZ_CAPIDATA           */
/*      ruft auf      :           LZ_TSS_SEND, UT_DP_FINDFIRST, */
/*                        UT_WSL_WRIT, UT_WSL_READ, UT_UL_ONX  */
/*      I/O-Kanäle   :                               */
/*****
/*      Eingabe Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung           */
/*      *** KEINE ***                               */
/*      -----                               */
/*      Ausgabe Parameter:                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung           */
/*      char *  LZ_TSS_DATA           Zeiger auf ein FND-Datagramm, */
/*                                     das an ein anderes Modul (z.B. */
/*                                     IZ, SP) weitergegeben werden */
/*                                     soll. Das Ziel ist durch die */
/*                                     LZ_SUBID gekennzeichnet. */
/*                                     SONST: NULL-Zeiger           */
/*      -----                               */
/*      Transiente Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung           */
/*      *** KEINE ***                               */
/*****
/*      Autor      Datum      Version      Kurzzeichen           */
/*      -----                               */
/*      Original :   W. Fries      01.05.97      1.1              */
/*      Änderung:   W. Fries      01.11.97      1.1a      WF1      */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand     :   01.10.98    */
/*****
/*   Datei     :   TSS_RECV.C           */
/*   Aufruf    :   LZ_TSS_RECV ( cDatagramm, iLen )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine behandelt das empf. TSS13a-Datagramm "cDatagramm" */
/*   der Länge "iLen".           */
/*           */
/*   aufgerufen von :           LZ_CAPIDATA           */
/*           */
/*   ruft auf      :           UT_DT_ADD, UT_WSL_WRIT           */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----           */
/*   char     cDatagramm           empfangenes FND-Datagramm           */
/*   -----           */
/*   int      iLen           Länge des FND-Datagrammes           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----           */
/*   int      LZ_TSS_RECV           Returncode der Funktion           */
/*   -----           */
/*   0=FALSE: KEIN Fehler           */
/*   -1: OPEN-Fehler           */
/*   -2: READ/SCANF-Fehler           */
/*   -3: WRITE-Fehler           */
/*   -4: ALLOC-Fehler           */
/*   -5: FND-Fehler           */
/*   -6: CAPI-Fehler           */
/*   -7: X25-Fehler (EICON-Bibl.)           */
/*   -8: unerwartetes DATA_IND           */
/*   -9: unerwartetes DISC_IND           */
/*   -10: nicht berechtigter Anrufer           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*****
/*   Autor      Datum      Version   Kurzzeichen           */
/*   -----           */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:           */
/*   -----           */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   TSS_SEND.C           */
/*   Aufruf   :   LZ_TSS_SEND ( cDatagramm, iLen )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine behandelt das zu sendende FND-Datagramme der */
/*   Länge "iLen".           */
/*           */
/*   aufgerufen von :           LZ_TSS_DATA           */
/*           */
/*   ruft auf      :           *** KEINE ***           */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----           */
/*   char     cDatagramm           zu sendendes FND-Datagramm */
/*   -----           */
/*   int      iLen                Länge des FND-Datagrammes */
/*   -----           */
/*   (Nur aus Kompatibilität zu and. */
/*   Protokollen, verwendet wird */
/*   "APCI->len" )           */
/*   -----           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----           */
/*   int      LZ_TSS_SEND           Returnkode der Funktion */
/*   -----           */
/*   0=FALSE: KEIN Fehler           */
/*   -----           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.   Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*   -----           */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----           */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:           */
/*   -----           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   VBP_CMD.C             */
/*   Aufruf   :   LZ_VBP_CMD ( cCMD )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine vermerkt das in "cCMD" gelieferte Verbindungs- */
/*   Profil in einer zugehörigen Datei (Segmentierter Karteipunkt */
/*   gemäß LV unter 4.3 ff.). Ist der Aktivierungszeitpunkt früher */
/*   als der bisher als nächster vermerkte, so wird der neue      */
/*   Zeitpunkt als nächster auszuführender vermerkt.             */
/*   Als Ergebnis wird der Zeiger auf das für "cCMD" zu liefernde */
/*   RSP-Datagramm geliefert.                                     */
/*   */
/*   aufgerufen von :   LZ_FND_DATA                                         */
/*   */
/*   ruft auf      :   UT_DT2CI, UT_TAG_S2DT, UT_OBJ_GETI,                */
/*                   UT_FNDERROR, UT_FNDREJECT,                          */
/*                   UT_NORM_STRING, diverse Makros FND_...              */
/*                   und SET_FND_..., SY_IEEE2F                          */
/*   */
/*   I/O-Kanäle   :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung                    */
/*   char*    cCMD          */          Zeiger auf das auszuwertende */
/*                   */          CMD-Datagramm mit dem neuen */
/*                   */          Verbindungs-Profil                */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung                    */
/*   char*    LZ_VBP_CMD   */          Zeiger auf das RSP-Datagramm */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung                    */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen                    */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1                        */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   VBP_LOAD.C            */
/*   Aufruf   :   LZ_VBP_LOAD ( uIndOBJ ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine lädt das aktuelle Verbindungs-Profil und den */
/*   Zeitpunkt zur Aktivierung des nächsten VBP für das Objekt */
/*   mit dem Index "uIndOBJ" in der Objekt-Struktur. Für veraltete */
/*   Verbindungs-Profile wird die zugehörige Datei gelöscht.      */
/*   */
/*   aufgerufen von :           LZ_VBP_INIT, LZ_VBP_POLL          */
/*   */
/*   ruft auf      :           LZ_VBP_NLZH, UT_OBJ_GETI, UT_CI2DT, */
/*                       SY_FI_FINDFIRST, SY_FI_FINDNEXT,        */
/*                       SY_DISKFREE                             */
/*   */
/*   I/O-Kanäle   :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   unsigned uIndOBJ           Index der Eintrags in der OBJ- */
/*   Struktur für den das aktuelle */
/*   Verbindungs-Profil geladen */
/*   werden soll.           */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   int      LZ_VBP_LOAD      Returnkode der Funktion      */
/*   0=FALSE: KEIN Fehler      */
/*   -1: OPEN-Fehler          */
/*   -2: READ-Fehler          */
/*   -3: WRITE-Fehler         */
/*   -4: ALLOC-Fehler         */
/*   -9: KEIN Eintrag in OBJ-Struktur */
/*   mit angegebenem Index      */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***           */
/*****
/*   Autor      Datum      Version  Kurzzeichen  */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1         */
/*   Änderung:                               */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei    :   VBP_NLZH.C           */
/*   Aufruf   :   LZ_VBP_NLZH ( uOBJ_Index, uModus )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine berechnet bezogen auf den aktuellen Zeitpunkt den */
/*   nächsten Zeitpunkt für die Übertragung der zwischengespeicherten*/
/*   Daten an die LZH im DOS-Format.           */
/*           */
/*   aufgerufen von :           LZ_VB_CONN, LZ_FND_DATA, LZ_VBP_LOAD */
/*           */
/*   ruft auf      :           UT_OBJ_GETI, UT_DT_ADD, UT_FNDNX2DT   */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*           -----           */
/*   unsigned uOBJ_Index      Index des Objektes in der OBJ- */
/*                           Struktur, für das der nächste */
/*                           Übertragungszeitpunkt berechnet */
/*                           werden soll.           */
/*           */
/*   unsigned uModus          0: Initialisierung nach Start */
/*                           1: Nächster Zeitpunkt nach */
/*                           erfolgreicher Übertragung */
/*                           (Zyklus oder Sync.)           */
/*                           2: Nächster Zeitpunkt nach */
/*                           fehlerhafter Übertragung */
/*                           (Wartezeit)           */
/*           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*           -----           */
/*   uns.long LZ_VBP_NLZH      Datum/Zeit der nächsten Über- */
/*                           tragung im DOS-Format           */
/*                           SONST: 0xFFFFFFFF (UNDEF)           */
/*           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name          Dim.   Beschreibung           */
/*           -----           */
/*   *** KEINE ***           */
/*           */
/*****
/*           Autor      Datum      Version      Kurzzeichen   */
/*           -----           */
/* Original :   W. Fries      01.05.97      1.1           */
/* Änderung:           */
/*           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ Leitzentralen      Stand    :   01.10.98    */
/*****
/*   Datei     :   VBP_POLL.C           */
/*   Aufruf    :   LZ_VBP_POLL ( )      */
/*****
/*   Beschreibung :                       */
/*   Diese Routine prüft, ob ein Verbindungs-Profil geladen werden */
/*   soll. Falls JA, wird dieses objektspezifisch geladen und die  */
/*   Datei des veralteten VBP gelöscht. Zusätzlich wird der Index  */
/*   des Eintrags in der Objekt-Struktur vermerkt, für das als    */
/*   nächstes ein VBP geladen werden soll.                          */
/*   */
/*   aufgerufen von :           LZ_POLL */
/*   */
/*   ruft auf      :           LZ_VBP_LOAD, UT_OBJ_GETI */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   */
/*   int      LZ_VBP_POLL           Returnkode der Funktion */
/*   */
/*   0=FALSE: KEIN Fehler */
/*   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1      */
/*   Änderung:                                     */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### 5.7.5. Druckausgaben, Meldungstexte

\*\*\* KEINE \*\*\*

### 5.7.6. Datenkatalog

Folgende Dateien stehen im Verzeichnis \GAK\LZ bzw. den folgenden Unterverzeichnissen zur Verfügung :

VBP\VBP\_oooo.nnn

IWAIWA\_oooo.nnn

Die in den Dateien "**VBP\_oooo.nnn**" abgelegten Verbindungs-Profile steuern den Verbindungsaufbau des GA-Knotens zur LZH, dem BWD-G und BWD-E. Es können mehrere Versionen mit unterschiedlichen Aktivierungszeitpunkten abgelegt sein.

Die editierbaren Text-Dateien werden im Unterverzeichnis \GAK\LZ\VBP abgelegt und enthalten das Verbindung-Profil für das Objekt "oooo" mit der Versions-Kennung "nnn" im folgenden Format :

Die ERSTE Zeile hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	"001"	Kennung der ERSTEN Zeile
2	4	5	3	"::"	Trennzeichen (OHNE '"')
3	7	20	14	Akt_Zeit	Aktivierungs-Zeitpunkt im Format JJJJMMTTHHMMSS
4	22	24	3	Dim_Zyklus	Zeitbezogene Dimension für "ZyklusZeit" (FND-Dimension 12 - 15 )
5	26	39	14	ZyklusZeit	Übertragungszyklus, an dem die zwischengespeicherten Daten an die LZH übertragen werden sollen bezogen auf "Dim_Zyklus" <=0: KEINE zyklische Übertragung
6	41	54	14	Max_Puffer	Max. Kapazität in MB, die für die Zwischenspeicherung verwendet werden darf. <0: KEINE Puffergrenze

					berücksichtigen ( VORSICHT, da damit die Funktionstüchtigkeit des GA-Knotens gefährdet wird, wenn das Speicher- medium voll ist !!)
					0: KEINE Zwischenpufferung für ALLE Datenpunkte (sinnvoll NUR bei einer festen bzw.stehenden Verbindung !!)
7	56	58	3	Dim_B1	Zeitbezogene Dimension für "B1_AnfZeit" (FND-Dimension 161 - 169 )
8	60	73	14	B1_AnfZeit	Anfangszeitpunkt des ERSTEN Übertragungsbereiches, ab dem die zwischengespeicherten Daten an die LZH übertragen werden sollen bezogen auf "Dim_B1" <=0: Übertragungsbereich 1 NICHT definiert
9	75	88	14	B1_EndZeit	Endezeitpunkt des ERSTEN Übertragungsbereiches, bis zu dem die zwischengespeicherten Daten an die LZH übertragen werden sollen bezogen auf "Dim_B1" <=0: Übertragungsbereich 1 NICHT definiert
10	90	92	3	Dim_B2	Zeitbezogene Dimension für "B2_AnfZeit" (FND-Dimension 161 - 169 )
11	94	107	14	B2_AnfZeit	Anfangszeitpunkt des ZWEITEN Übertragungsbereiches, ab dem die zwischengespeicherten Daten an die LZH übertragen werden sollen bezogen auf "Dim_B2" <=0: Übertragungsbereich 2 NICHT definiert
12	109	122	14	B2_EndZeit	Endezeitpunkt des ZWEITEN Übertragungsbereiches, bis zu dem die zwischengespeicherten Daten an die LZH übertragen werden sollen bezogen auf "Dim_B2" <=0: Übertragungsbereich 2 NICHT definiert
13	124	126	3	SyncCode	Zeitbezogene Dimension für "SyncZeit" (FND-Dimension 161 - 169 )
14	128	141	14	SyncZeit	Zeitpunkt, an dem generell eine Übertragung der zwischengespeicherten Daten an die LZH erfolgen soll bezogen auf "SyncCode"
15	143	156	14	Wartezeit	Wartezeit nach einer einer fehlerhaften Übertragung nach der ein erneuter Verbindungsaufbau zur Fortsetzung der Übertragung versucht werden soll in Minuten bezogen auf den letzten Versuch (FND- Dimension 168)

Ab der ZWEITEN Zeile werden die einzelnen Verbindungen (jeweils 8) für zur LZH, dem BWD-GEFAHR und dem BWD-EINBRUCH zeilenweise definiert. Jede Zeile hat den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Kennung	beginnend mit LZ1 ... LZ8, BG1 ... BG8 und BE1 ... BE8
2	4	5	3	"::"	Trennzeichen (OHNE '"')
3	7	9	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** LZH: 20=ISDN / CAPI *** LZH: 40=UDP / IP *** BWD-GEFAHR: 20=ISDN / CAPI *** BWD-EINBRUCH: 20=ISDN / CAPI
4	11	13	3	Verbindungs-Prot.	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 21). Für die LZH ist "FND gemäß FND- Spezifikation mit FND- Erweiterungen" verbindlich festgelegt (d.h. 002).  *** Wird begrenzt ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** LZH: 2=FND mit Erweiterungen *** BWD-GEFAHR: 10=TSS 13a *** oder : 1=FND 1.0 *** BWD-EINBRUCH: 10=TSS 13a *** oder : 1=FND 1.0
5	15	74	<=60	Verbindungs-Adr.	abhängig von der Verbindungs- Art wird die für den Verbin- dungsaufbau notwendige Adresse angegeben (z.B. Rufnummer und/oder X.25-DTE-Adresse oder IP-Adr.#Port)  Ist keine Verbindungsadresse angegeben, so wird die zugehörige Warteschlange gelöscht, d.h. die Meldungen werden unterdrückt.  Wird durch das Zeilenabschluß- zeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Die Extension "nnn" ist nicht zwangsläufig sortiert entsprechend der zeitlichen Reihenfolge der Aktivierung, sondern wird vom Programm eindeutig vergeben.

**Hinweis: Im Verzeichnis dürfen nur ausschließlich die VBP-Dateien entsprechend der angegebenen Namenskonvention abgelegt werden, da das GAK-Programm alle vorhandenen Dateien prüft deren Name mit "VBP" beginnt.**

Bei der Installation des GA-Knotens muß mindestens eine VBP-Datei für jedes angeschlossene Objekt eingerichtet werden.

In den Dateien "IWA\_oooo.nnn" werden die Istwert-Abfragen gespeichert und verwaltet. Die Text-Dateien werden im Unterverzeichnis \GAK\LZIWA abgelegt und enthalten die Istwert-Abfrage für das Objekt "oooo" mit der von der LZH vergebenen LZ\_SUBID. Die Extension "nnn" entspricht der LZ\_SUBID mit führenden Nullen und muß eindeutig sein, d.h. es darf niemals mehreren Objekten die gleiche von der LZH vergabene LZ\_SUBID zugeordnet werden.

Die ERSTE Zeile hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	"001"	Kennung der ERSTEN Zeile
2	4	5	2	::"	Trennzeichen (OHNE '"')
3	7	20	14	Akt_Zeit	Aktivierungs-Zeitpunkt im Format JJJJMMTHHMMSS

Ab der ZWEITEN Zeile werden die einzelnen abzufragenden Datenpunkte mit den zugehörigen Abfrage-Parametern zeilenweise definiert. Jede Zeile hat den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	3	3	Zeilen-Nummer	beginnend mit 002
2	4	5	3	::"	Trennzeichen (OHNE '"')
3	7	22	16	dp-id	Datenpunkt-Adresse, ggf. mit zeichen- und/oder textsensitiven Platzhaltern ("?" bzw. "**")
4	24	37	14	ZyklusZeit	Zyklus in Minuten, in dem die oben definierten Datenpunkte abgefragt werden sollen <=0: KEINE zyklische Abfrage, d.h. nur einmalig
5	39	41	3	SyncCode	Zeitbezogene Dimension für "SyncZeit" (FND-Dimension 160 - 169 )
6	43	56	14	SyncZeit	Zeitpunkt, an dem generell eine Istwert-Abfrage gestartet werden soll bezogen auf "SyncCode" <0: KEINE Synchronisation
7	58	71	14	Verbindungsdauer	Dauer in Minuten, die die Verbindung aufrecht erhalten werden soll. <=0: Verbindung wird sofort, nachdem alle RSP-Datgramme an die LZH übertragen wurden, abgebaut.
8	73	86	14	Start-Zeit	Zeitpunkt für Start der nächsten zyklischen Abfrage im Format JJJJMMTHHMMSS

**Hinweis:** Im Verzeichnis dürfen nur ausschließlich die IWA-Dateien entsprechend der angegebenen Namenskonvention abgelegt werden, da das GAK-Programm alle vorhandenen Dateien prüft, deren Name mit "IWA" beginnt.

Im aktuellen Arbeitsverzeichnis werden temporär die folgenden Dateien angelegt:

**LZ\_ILZH.WSL:** Warteschlange der von der LZH empfangenen Datagramme

**LZ\_OLZH.WSL:** Warteschlange der an die LZH zu sendenden Datagramme

**LZ\_OBWDG.WSL:** Warteschlange der an den BWD-Gefahr zu sendenden Datagramme

**LZ\_OBWDE.WSL:** Warteschlange der an den BWD-Einbruch zu sendenden Datagramme

### 5.7.7. Definition der Datenübertragung

Der GA-Knoten ist über die ISDN- bzw. Netzwerk-Karte alternativ mit einer der folgenden Leitzentralen verbunden :

**LZH:** Auf den OSI/ISO-Schichten 1 - 3 werden die B-Kanal-Protokolle B1: 64 kBit/s mit HDLC, B2: ISO 7776 (X.75 SLP) und B3: ISO 8208 (X.25 DTE-DTE) gemäß CAPI 2.0 verwendet. Auf OSI/ISO-Schicht 7 wird das FND-Protokoll in der Version 1.0 zuzüglich der spezifizierten Ergänzungen realisiert. Die Übertragung der FND-Datagramme erfolgt binär.

**BWD-Gefahr:** Auf den OSI/ISO-Schichten 1 - 3 werden die B-Kanal-Protokolle B1: 64 kBit/s mit HDLC, B2: ISO 7776 (X.75 SLP) und B3: transparent gemäß CAPI 2.0 verwendet. Auf OSI/ISO-Schicht 7 wird das Protokoll TSS 13a oder FND 1.0 (ohne Ergänzungen) verwendet.

Beim TSS 13a werden die einzelnen Bytes hexadezimal kodiert und mittels zweier ASCII-Zeichen aus dem Zeichenvorrat '0' - '9' (0x30 - 0x39) und 'A' - 'F' (0x41 - 0x46) übertragen. Das Ende des Datagrammes wird durch ein Carriage Return (0x0D) gekennzeichnet.

Die Übertragung der FND-Datagramme erfolgt binär.

**BWD-Einbruch:**       \*\*\* Analog zum BWD-Gefahr \*\*\*

**Hinweis:** Die Übertragung über das ISDN erfolgt stets paketweise, d.h. je Datagramm wird ein Paket übertragen. Die maximale Paketlänge ist auf 128 Bytes begrenzt, so dass für keines der verwendeten Protokolle die Behandlung des M-Bit erforderlich ist.

Insbesondere bei Verwendung von Terminal-Adaptern auf seiten eines BWD ist die paketweise Übertragung sicherzustellen. Tests haben gezeigt, dass hierbei erhebliche Probleme auftreten können, da sich die einzelnen Produkte bei der Umsetzung der über die serielle Schnittstelle empfangenen Zeichen nicht deterministisch verhalten (z.B. Timeouts, interne Optimierungs-Algorithmen, fehlende Steuerinformationen für Paket-Anfang und -Ende). Auch für den BWD wird der Einsatz von ISDN-Steckkarten mit einer CAPI 2.0-Schnittstelle (z.B. DIVA, Fa. Diehl) empfohlen.

Der vom ISDN auf dem D-Kanal gelieferte Zeitstempel wird zum Abgleich der Systemzeit verwendet (vgl. ZA-Zeile in der Konfigurations-Datei CFG/GAK.CFG).

Die eigene Rufnummer als auch die Rufnummer des Verbindungs-Partner ("Called Party Number" bzw. "Calling Party Number") werden explizit gesetzt und müssen zur gegenseitigen Authentifizierung übertragen werden (vgl. korrespondierende Einträge in den Dateien CFG/ANRUFER.CHK, LZ/VBP/VBP\_0000.nnn und der ER-Zeile der Konfigurations-Datei CFG/GAK.CFG).

Unter LINUX steht zusätzlich das UDP/IP-Protokoll zur Verfügung. Das FND-Datagramm wird als ein Paket innerhalb des UDP/IP-Rahmens versandt.

## **5.8. Hauptprogramm des Standard-Schnittstellen-Adapters (SSA)**

Das Programm ISSA bzw. LSSA ist durch Reduzierung auf die grundlegenden Funktionen aus dem GA-Knoten abgeleitet. Insbesondere sind keine Funktionen verfügbar, die mittels der Karteipunkte parametrisiert werden (DPP, VBP, IWA, SP-Modul). Die Überwachung der IZ's, die Flimmerkontrolle usw. bleiben erhalten. Das Starten, der Ablauf und das Beenden erfolgt analog zum GA-Knoten.

**Hinweis:** Die Standard-Schnittstellen-Adapter stehen derzeit nur unter DOS zur Verfügung.

Die spezifischen EXE-Dateien werden durch das Binden erzeugt. IZ- und LZ-SSA nur durch Einbinden der Dateien ISSAINIT bzw. LSSAINIT unterschieden.

Der IZ-SSA wird gemäß FND-Spezifikation zwischen IZ und dem Netz, d.h. in räumlicher Nähe zur IZ, installiert. Es stehen die gleichen IZ-Treiber wie für den GA-Knoten zur Verfügung. Anstelle des LZ-Moduls für das ISDN wird ein spezifisches Modul mit FND 1.0 über V.24 (vgl. LZRS) eingebunden, d.h. der IZ-SSA kann an die Module IZ01 und IZ03 des GA-Knotens über V.24 angeschlossen werden.

Der LZ-SSA wird gemäß FND-Spezifikation zwischen dem Netz und der LZ, d.h. in räumlicher Nähe zur LZ, installiert. Es stehen die IZ-Treiber IZ01 und IZ03 (FND 1.0) zur Verfügung. Ergänzend dazu wird auch der Treiber IZ02 eingebunden, so dass zusätzliche Datenpunkte über die SPS an der LZ aufgelegt werden können. Anstelle des LZ-Moduls für das ISDN wird ein spezifisches Modul für die jeweilige LZ eingebunden. Es stehen folgende LZ-Treiber zur Verfügung :

LZEY        3964R / RK512 über V.24 (z.B. für Sauter EY 2400-LS)

### **5.8.1. Übersicht der verwendeten Module**

SSA	Hauptprogramm
IZ	Inselzentralen-Ebene (FND, SPS, ...)
LZ	Leitzentralen-Ebene (GA-Knoten, Fremdhersteller)
CAPI	CAPI-Schnittstelle (Version 2.0)
UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### **5.8.2. Liste mit Namen und Aufgaben**

SSA	Hauptprogramm
SSA_INIT	Initialisierungen (Datei ISSAINIT bzw. LSSAINIT)
SSA_LOOP	Zyklische Bearbeitung der Ein-/Ausgaben



SSA_IIZ	Bearbeitung der Eingaben von der IZ-Ebene
SSA_OIZ	Bearbeitung der Ausgaben an die IZ-Ebene
SSA_ILZ	Bearbeitung der Eingaben von der LZ-Ebene
SSA_FLIMMCHK	Flimmerkontrolle für Melde- und Schaltpunkte
SSA_ROLLMELD	Ausgabe von Meldungen am Statusbildschirm im Rollmodus
SSA_SYSCHK	Eigenüberwachung und Systemabgleich
SSA_ITAST	Behandlung von Tastatureingaben (STOP-Kommando)
SSA_EXIT	Ende-Behandlungen

### 5.8.3. Aufrufstruktur

SSA

SSA\_INIT

GAK\_DPP\_INIT

IZ\_INIT (vgl. Modul IZ)

LZ\_INIT (vgl. Modul LZ)

SSA\_LOOP

SSA\_IIZ

SSA\_FLIMMCHK

SSA\_ROLLMELD

IZ\_POLL (vgl. Modul IZ)

LZ\_RSP\_USM (vgl. Modul LZ)

SSA\_ILZ

LZ\_POLL (vgl. Modul LZ)

IZ\_CMD (vgl. Modul IZ)

LZ\_RSP\_USM (vgl. Modul LZ)

SSA\_OIZ

IZ\_CMD (vgl. Modul IZ)

SSA\_SYSCHK

IZ\_CMD (vgl. Modul IZ)  
 IZ\_SETTM (vgl. Modul IZ)  
 LZ\_ZA\_TIME (vgl. Modul LZ)  
 LZ\_STAT (vgl. Modul LZ)

SSA\_ITAST

SSA\_EXIT

GAK\_DPP\_EXIT

IZ\_EXIT (vgl. Modul IZ)

LZ\_EXIT (vgl. Modul LZ)

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

**5.8.4. Modulbeschreibungen**

```

/*****
/*   Produkt   :   SSA                               Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA                     Stand     :   01.10.98  */
/*****
/*   Datei    :   SSA.C                               */
/*   Aufruf   :   SSA [ Schalter für Test-Funktionen] */
/*****
/*   Beschreibung :
/*   Dieses Hauptprogramm dient zur Realisierung des GA-Knotens
/*
/*   aufgerufen von :
/*
/*   ruft auf      :      SSA_INIT, SSA_LOOP, SSA_EXIT
/*                   BS_DF_INIT, BS_DF_OUTS
/*
/*   I/O-Kanäle   :      stdin, stdout, stderr
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   *** vgl. Schalter für Test-Funktionen in "GAK_TEST.H" ***
/*   *** und die Beschreibung zu deren Aufruf in der
/*   *** System-Spezifikation.
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   int      exit
/*                   EXIT-Kode des Hauptprogrammes
/*                   0: KEIN Fehler
/*                   >0: Zeilennummer in der Konf.-
/*                       in der ein Fehler auftrat
/*                   -1nn: ABRUCH nach INIT
/*                   -2nn: ABRUCH nach LOOP
/*                   -3nn: ABRUCH nach EXIT
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
    
```

```

/*                                                                 */
/*      *** KEINE ***                                           */
/*                                                                 */
/******                                                                 */
/*      Autor      Datum      Version      Kurzzeichen      */
/*-----|-----|-----|-----|-----|-----|-----|-----|
/* Original :    W. Fries      01.05.97      1.1              */
/* Änderung:    W. Fries      07.09.98      1.1d      WF2     */
/*                                                                 */
/******                                                                 */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770  */
/******                                                                 */

```

```

/*****
/*      Produkt      :      SSA              Version   :   1.2      */
/*      Modulname   :      SSA IZ/LZ-SSA      Stand    :   01.10.98 */
/*****
/*      Datei      :      INIT.C              */
/*      Aufruf     :      SSA_INIT ( argc, argv ) */
/*****
/*      Beschreibung :                      */
/*      Diese Routine führt die GA-Knoten-spezifischen Initialisierungen*
/*      nach dem Starten des Hauptprogrammes "ISSA.." durch :
/*      - allg. UT-Initialisierungen mit Laden der zugehör. Strukturen */
/*      aus der Konfigurationsdatei und Ausgabe von Meldungen am */
/*      Protokolldrucker */
/*      - LZ-Initialisierungen mit Laden der zugehörigen Strukturen */
/*      aus der Konfigurationsdatei bzw. den Verbindungsprofilen */
/*      - IZ-Initialisierungen mit Laden der zugehörigen Strukturen */
/*      aus der Konfigurationsdatei */
/*      - Aufbau des Status-Bildschirms */
/*      */
/*      aufgerufen von :      SSA */
/*      */
/*      ruft auf      :      SSA_EXIT, LZ_INIT,
/*      IZ_INIT, IZ_IZ_IZTEXT,
/*      UT_INIT, UT_GA_CFG, UT_OB_CFG, UT_DP_CFG*
/*      UT_KL_CFG, UT_DR_OPROT, UT_ER_GETER, */
/*      UT_DP_FINDFIRST, UT_DP_FINDNEXT, */
/*      BS_INIT, BS_DF_INIT, BS_DF_OUTS, */
/*      SY_CPU_HALT */
/*      */
/*      I/O-Kanäle      :                      */
/*****
/*      Eingabe Parameter :                      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      int      SSA_INIT      Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler */
/*      ----- */
/*      Transiente Parameter :                      */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung */
/*      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1 */
/*      Änderung:
/*      -----
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   INIT.C         */
/*   Aufruf    :   SSA_INIT ( argc, argv ) */
/*****
/*   Beschreibung :
/*   Diese Routine führt die GA-Knoten-spezifischen Initialisierungen*
/*   nach dem Starten des Hauptprogrammes "LSSA.." durch:
/*   - allg. UT-Initialisierungen mit Laden der zugehör. Strukturen */
/*   aus der Konfigurationsdatei und Ausgabe von Meldungen am */
/*   Protokolldrucker */
/*   - LZ-Initialisierungen mit Laden der zugehörigen Strukturen */
/*   aus der Konfigurationsdatei bzw. den Verbindungsprofilen */
/*   - IZ-Initialisierungen mit Laden der zugehörigen Strukturen */
/*   aus der Konfigurationsdatei */
/*   - Aufbau des Status-Bildschirms */
/*
/*   aufgerufen von :   SSA
/*
/*   ruft auf :
/*   SSA_EXIT, LZ_INIT,
/*   IZ_INIT, IZ_IZ_IZTEXT,
/*   UT_INIT, UT_GA_CFG, UT_OB_CFG, UT_DP_CFG*
/*   UT_KL_CFG, UT_DR_OPROT, UT_ER_GETER,
/*   UT_DP_FINDFIRST, UT_DP_FINDNEXT,
/*   BS_INIT, BS_DF_INIT, BS_DF_OUTS,
/*   SY_CPU_HALT
/*
/*   I/O-Kanäle :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   *** KEINE ***
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   int      SSA_INIT
/*           Returnkode der Funktion
/*           0=FALSE: KEIN Fehler
/*
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version  Kurzzeichen
/*   -----
/*   Original :   W. Fries   01.05.97   1.1
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      SSA              Version   :   1.2      */
/*      Modulname   :      SSA IZ/LZ-SSA      Stand    :   01.10.98 */
/*****
/*      Datei      :      LOOP.C              */
/*      Aufruf     :      SSA_LOOP ( )       */
/*****
/*      Beschreibung :                      */
/*      Diese Routine führt die zyklische Bearbeitung der SSA- */
/*      Funktionen durch. Dabei werden nacheinander die folgenden */
/*      Bearbeitungsschritte zyklisch durchgeführt :          */
/*      - Behandlung von Eingabedaten von den DDC-IZ          */
/*      - Eingabe von Daten von der LZ                        */
/*      - Ausgabe von CMD-Datagrammen an die DDC-IZ's        */
/*      - Ausgabe von Meldungen auf den Protokolldrucker     */
/*      - Eigenüberwachung und Systemabgleich               */
/*      - Behandlung von evtl. Tastatur-Eingaben            */
/*      aufgerufen von : SSA                                */
/*      ruft auf      : SSA_IIZ, SSA_ILZ, SSA_OIZ,          */
/*                    SSA_SYSCHK, SSA_ITAST,              */
/*                    UT_DR_OPROT, UT_LOOPTIME            */
/*      I/O-Kanäle   :                      */
/*****
/*      Eingabe Parameter :                      */
/*      -----                      */
/*      Typ      Name      Dim.      Beschreibung          */
/*      *** KEINE ***                      */
/*      -----                      */
/*      Ausgabe Parameter:                      */
/*      -----                      */
/*      Typ      Name      Dim.      Beschreibung          */
/*      int      SSA_LOOP                      Returnkode der Funktion */
/*                    0=FALSE: KEIN Fehler          */
/*      -----                      */
/*      Transiente Parameter :                      */
/*      -----                      */
/*      Typ      Name      Dim.      Beschreibung          */
/*      *** KEINE ***                      */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      -----                      */
/*      Original :      W. Fries      01.05.97      1.1          */
/*      Änderung:                      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   IIZ.C         */
/*   Aufruf    :   SSA_IIZ ( )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine führt die Behandlungen der Eingaben von den */
/*   angeschlossenen DDC-IZ durch. Folgende Einzelfunktionen werden */
/*   durchgeführt :   */
/*   - Polling der DDC-IZ's und Behandlung empfangener Datagramme */
/*   */
/*   aufgerufen von :   SSA_LOOP */
/*   */
/*   ruft auf      :   SSA_FLIMMCHK, SSA_ROLLMELD, */
/*                   IZ_POLL, LZ_RSP_USM, */
/*                   UT_DP_ATBTEXT, UT_DP_DIMTEXT, */
/*                   UT_CFG_WRIT, UT_DRDPMELD, UT_UCGETBIT, */
/*                   UT_DP_FINDFIRST, diverse Makros FND_..., */
/*                   SY_IEEE2F */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung      */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung      */
/*   */
/*   int      SSA_IIZ              Returnkode der Funktion */
/*                   0=FALSE: KEIN Fehler */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung      */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original  :   W. Fries   01.05.97   1.1           */
/*   Änderung  :   W. Fries   01.02.98   1.1b         WF1          */
/*   Änderung  :   W. Fries   07.09.98   1.1d         WF3          */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      SSA              Version   :   1.2      */
/*      Modulname   :      SSA IZ/LZ-SSA      Stand    :   01.10.98 */
/*****
/*      Datei      :      OIZ.C              */
/*      Aufruf     :      SSA_OIZ ( )       */
/*****
/*      Beschreibung :                      */
/*      Diese Routine führt die Behandlungen der Ausgaben an die */
/*      angeschlossenen DDC-IZ durch. Folgende Einzelfunktionen werden */
/*      durchgeführt :                      */
/*      - Prüfung, ob zyklische Abfragen durchgeführt werden müssen */
/*      */
/*      aufgerufen von :      SSA_LOOP      */
/*      */
/*      ruft auf      :      IZ_CMD, UT_DT_ADD, UT_FNDNX2DT, */
/*      UT_FND_TIME, diverse Makros FND... und */
/*      SET_FND...      */
/*      */
/*      I/O-Kanäle   :                      */
/*      */
/*****
/*      Eingabe Parameter :                  */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      */
/*      *** KEINE ***      */
/*      ----- */
/*      Ausgabe Parameter:                  */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      */
/*      int      SSA_OIZ      Returnkode der Funktion */
/*      0=FALSE: KEIN Fehler      */
/*      ----- */
/*      Transiente Parameter :              */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung      */
/*      */
/*      *** KEINE ***      */
/*      ----- */
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/*      Original :      W. Fries      01.05.97      1.1      */
/*      ----- */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   ILZ.C           */
/*   Aufruf    :   SSA_ILZ ( )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine führt die Behandlungen der Eingaben von der LZ. */
/*   */
/*   aufgerufen von :   SSA_LOOP   */
/*   */
/*   ruft auf      :   IZ_CMD      */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----            */
/*   Typ      Name      Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***      */
/*   */
/* - - - - -            */
/*   Ausgabe Parameter:   */
/*   -----            */
/*   Typ      Name      Dim.   Beschreibung           */
/*   */
/*   int      SSA_ILZ           Returnkode der Funktion */
/*                                     0=FALSE: KEIN Fehler */
/*   */
/* - - - - -            */
/*   Transiente Parameter :   */
/*   -----            */
/*   Typ      Name      Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***      */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen     */
/*   -----            */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   FLIMMCHK.C      */
/*   Aufruf    :   SSA_FLIMMCHK ( DP_ES12, uIndex )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft für die Ergänzungsstruktur zum Melde- bzw. */
/*   Schaltpunkt "DP_ES12", ob mit dem Zustandswechsel auf den neuen */
/*   "uIndex" ein Flimmern verbunden ist. Geprüft wird die min. */
/*   Dauer, die der vorherige Zustand angestanden sein muß und falls */
/*   dabei kein Flimmern erkannt wurde, ob die min. Periodendauer */
/*   eingehalten wurde. */
/*   */
/*   aufgerufen von :   SSA_IIZ */
/*   */
/*   ruft auf      :   UT_UDATETAG, UT_UTIMESEK */
/*   */
/*   I/O-Kanäle    :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   DP_ES12* DP_ES12      Zeiger auf die Ergänzungsstruktur. */
/*   uns.     uIndex      Neuer "actual", der einen */
/*   */
/*   Zustandswechsel von GEHT nach */
/*   KOMMT bzw. umgekehrt darstellt */
/*   */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   int      SSA_FLIMMCHK      Returnkode der Funktion */
/*   */
/*   0=FALSE: KEIN Flimmerer */
/*   SONST: Flimmerer erkannt */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1 */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   ROLLMELD.C      */
/*   Aufruf    :   SSA_ROLLMELD ( DP, cMeld )           */
/*****
/*   Beschreibung :   */
/*   Diese Routine gibt die Meldung "cMeld" für den Datenpunkt "DP" */
/*   in den nächsten freien Zeile für die Rolling-Meldungen am      */
/*   Bildschirm aus, wenn der Statusbildschirm aktiviert ist.      */
/*   die Zeile zwischen der neuen und der ältesten Meldung wird   */
/*   gelöscht.                                                     */
/*   */
/*   aufgerufen von :   SSA_IIZ                                     */
/*   */
/*   ruft auf      :   UT_BSDPMELD, BS_DF_INIT, BS_DF_OUTS        */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***                                             */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   ----- */
/*   int      SSA_ROLLMELD   Returncode der Funktion      */
/*   0=FALSE: KEIN Fehler                                         */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***                                             */
/*   ----- */
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   SYSCHK.C       */
/*   Aufruf    :   SSA_SYSCHK ( ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine führt die Eigenüberwachung und den Abgleich aller */
/*   Systemkomponenten durch.   */
/*   */
/*   Mit jedem Aufruf dieser Routine wird die jeweils nächste */
/*   Speicherzelle des Hauptspeichers getestet. Wird ein Fehler */
/*   erkannt, wird das Programm beendet.   */
/*   */
/*   Ist das Ende des zu überwachenden Hauptspeicherbereiches (1 MB) */
/*   erreicht, so erfolgen die Funktionen zum Systemabgleich :   */
/*   - Ausgabe der aktuellen Datum/Uhrzeit am Bildschirm   */
/*   - Eigenüberwachung (Speichertest, verwendeten Festplatten- */
/*     kapazitäten, (Res.: Temperatur-/Spannungsüberwachung, ...) */
/*   - Ausgabe der aktuellen Status am Bildschirm und auf Prot.-Druck */
/*   - Abgleich der CMOS-Uhr (RTC) mit der LZ-Systemzeit o.ä.   */
/*   - DOS-Uhr mit der CMOS-Uhr abgleichen   */
/*   */
/*   aufgerufen von :   SSA_LOOP   */
/*   */
/*   ruft auf :   IZ_CMD, IZ_SETTM, LZ_ZA_TIME, LZ_STAT,   */
/*               UT_ZA_GETZA, UT_SET_RTC, UT_RTC_DOS,   */
/*               UT_OBJ_GETI, UT_DP_FINDFIRST,   */
/*               UT_DP_FINDNEXT, UT_UCGETBIT, UT_FND_TIME */
/*               diverse Makros FND... und SET_FND...,   */
/*               BS_DF_INIT, BS_DF_OUTS,   */
/*               SY_FI_SIZE, SY_DISKFREE, SY_MEMCHK   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.   Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name      Dim.   Beschreibung   */
/*   */
/*   int      SSA_SYSCHK   Returnkode der Funktion   */
/*               0=FALSE: KEIN Fehler   */
/*               -700: Temperatur-Fehler   */
/*               -701: Speicher-Fehler   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.   Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*****
/*   Autor      Datum      Version      Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.05.97      1.1   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*   Produkt   :   SSA           Version   :   1.2       */
/*   Modulname :   SSA IZ/LZ-SSA   Stand    :   01.10.98  */
/*****
/*   Datei     :   ITAST.C        */
/*   Aufruf    :   SSA_ITAST ( )  */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft, ob eine Tastatur-Eingabe erfolgte und   */
/*   liest diese zeichenweise aus, um das eingelesene Zeichen    */
/*   weiterzuverarbeiten.                                         */
/*   In SSA ist nur die Eingabe der Abbruch-Sequenz STOP zulässig. */
/*   Es folgt die Abfrage, ob das Programm wirklich gestoppt werden */
/*   soll, diese ist ggf. mit J zu beantworten.                  */
/*   Die Eingabe der Zeichenfolge erfolgt ohne Abschlusszeichen (CR) */
/*   und lässt sich nicht editieren, d.h. kein DEL anwendbar.    */
/*   Jede Falscheingabe erzwingt die erneute Eingabe der gesamten */
/*   Sequenz.                                                     */
/*   */
/*   aufgerufen von :   SSA_LOOP   */
/*   */
/*   ruft auf       :   BS_DF_INIT, BS_DF_OUTS   */
/*   */
/*   I/O-Kanäle     :   stdin, stdout   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   int      SSA_ITAST     Dim.    Returnkode der Funktion   */
/*   */
/*   0=FALSE: KEIN Fehler   */
/*   -800: ABBRUCH durch den Bediener*/
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries    01.05.97    1.1       */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*      Produkt      :      SSA              Version   :   1.2      */
/*      Modulname   :      SSA IZ/LZ-SSA      Stand    :   01.10.98 */
/*****
/*      Datei      :      EXIT.C              */
/*      Aufruf     :      SSA_EXIT ( )        */
/*****
/*      Beschreibung :                      */
/*      Diese Routine führt die GA-Knoten-spezifischen ENDE-Behandlungen*
/*      vor dem Beenden des Hauptprogrammes durch :                      */
/*      - Eigenüberwachung (Watchdog, Speichertest, ...)                  */
/*      - spez. IZ-Ende-Behandlungen                                     */
/*      - spez. LZ-Ende-Behandlungen                                     */
/*      - Warteschlangen für die sofortige Ausgabe an die LZ             */
/*      - allg. UT-Ende-Behandlungen                                     */
/*      - Vermerken des SSA-Status SSA_STAT in Statusdatei                */
/*      */
/*      aufgerufen von :          SSA                                           */
/*      */
/*      ruft auf      :          IZ_EXIT, LZ_EXIT,                               */
/*      UT_EXIT, UT_DR_OPROT, UT_LOOPTIME                                     */
/*      */
/*      I/O-Kanäle   :                      */
/*      */
/*****
/*      Eingabe Parameter :                      */
/*      -----                      */
/*      Typ      Name          Dim.    Beschreibung                      */
/*      */
/*      *** KEINE ***                      */
/*      */
/*      -----                      */
/*      Ausgabe Parameter:                      */
/*      -----                      */
/*      Typ      Name          Dim.    Beschreibung                      */
/*      */
/*      int      SSA_EXIT                      Returnkode der Funktion   */
/*      0=FALSE: KEIN Fehler                      */
/*      */
/*      -----                      */
/*      Transiente Parameter :                      */
/*      -----                      */
/*      Typ      Name          Dim.    Beschreibung                      */
/*      */
/*      *** KEINE ***                      */
/*      */
/*****
/*      Autor      Datum      Version   Kurzzeichen  */
/*      -----                      */
/*      Original :   W. Fries    01.05.97    1.1          */
/*      Änderung:                                     */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### **5.8.5. Druckausgaben, Meldungstexte**

Fatale Fehler, die zum kontrollierten Beenden des Programmes führen, erzeugen eine entsprechende Fehlermeldung auf folgenden Ausgabekanälen, soweit diese verfügbar sind :

- als direkte Druckausgabe auf den Protokolldrucker, falls vorhanden
- Ausgabekanal für Trace-Meldungen, falls Schalter "/TD" beim Programmaufruf angegeben wurde.
- Standard-Error-Device (stderr), z.B. wie bei der Trace-Ausgabe ohne Angabe des Schalters "/TD"
- Standard-Output-Device (stdout), z.B. wie bei der Ausgabe aktueller Messwerte und Status. Dieser Ausgabekanal kann beim Programmaufruf mittels der Umlenkung durch ">" auf ein zusätzlich angeschlossenes Ausgabegerät oder eine Datei umgelenkt werden.

Der aktuelle Zustand des GA-Knoten wird im aktuellen Verzeichnis in der Datei **ISSA.STA** bzw. **LSSA.STA** vermerkt. Diese wird beim Programmstart geprüft und führt abhängig vom Zustand des letzten Programmlaufes im Falle eines für den weiteren Programmablauf kritischen Fehlers ggf. zu folgenden Meldungen:

- Protokolldrucker gestört
- Watchdog- bzw. interner Fehler
- Speicher-Fehler

Der Watchdog- bzw. interne Fehler kennzeichnet einen unkontrollierten Programmabbruch (z.B. nach Reset oder Stromausfall). Eine hardwarebasierte Watchdog wird momentan nicht unterstützt.

Der Speicher-Fehler wird bei zyklischen Prüfung des konventionellen Hauptspeicherbereiches erkannt und ist der einzige für den weiteren Ablauf kritische Fehler und führt zu der Abfrage :

"Soll das Programm neu gestartet werden (J/N) >>"

Wird sie nicht innerhalb von 1 Minute mit "J" beantwortet, so versucht der SSA die CPU anzuhalten, da der weitere Programmablauf nicht gesichert ist.

### **5.8.6. Datenkatalog**

Das Programm liest und schreibt bei seinem Betrieb folgende Dateien , die für die Konfiguration des SSA benötigt werden :

Folgende Dateien stehen im Verzeichnis \SSA\CFG zur Verfügung :

DP\_ATB.TXT

DP\_DIM.TXT

ISSA.CFG bzw. LSSA.CFG

Die editierbaren Text-Dateien "**DP\_ATB.TXT**" und "**DP\_DIM.TXT**" entsprechen denen des GA-Knotens.

Die editierbaren Text-Dateien "**ISSA.CFG**" und "**LSSA.CFG**" entsprechen der "**GAK.CFG**" des GA-Knotens.

Folgende Konfigurations-Parameter des GA-Knotens werden jedoch nicht ausgewertet :

- ML Meldelinien zum BWD-GEFAHR bzw. BWD-EINBRUCH

Der aktuelle Zustand des SSA wird im aktuellen Verzeichnis in der Datei **ISSA.STA** bzw. **LSSA.STA** vermerkt. Diese wird beim Programmstart geprüft und führt abhängig vom Zustand des letzten Programmlaufes zu den oben beschriebenen Meldungen.

Die einzelnen Ein- und Ausgaben zwischen dem SSA und den angeschlossenen Geräten als auch zwischen den programminternen Modulen werden vom SSA asynchron behandelt. Zu diesem Zweck werden empfangene bzw. auszugebende Daten in folgenden Warteschlangen-Dateien zwischengespeichert:

**DR\_OPROT.WSL:** Ausgaben an das Protokoll-Gerät bzw. die Protokoll-Datei

Treten während des Programmlaufes Fehler (z.B. Kommunikationsfehler mit einer IZ oder Leitzentrale) auf, so werden diese in der über die Konfigurationsdatei ISSA.CFG bzw. LSSA.CFG parametrisierten **Error-Log-Datei** (vgl. EL-Kommando) mit Zeitpunkt, Routinename, 2 Fehlerkodierungen und einem optionalen Zusatztext protokolliert. Diese Fehlermeldungen dienen ausschließlich der Fehleranalyse in Verbindung mit den Programm-Quellen und werden somit nicht einzeln dokumentiert.



Empfängt der SSA von einer IZ ein REJECT-, ERROR- oder unzulässiges FND-Datagramm, so werden diese bei gesetztem Schalter /LR, /LE bzw. /LI beim Programmstart in den folgenden Dateien binär protokolliert:

**REJECT.LOG:** REJECT-Datagramme

**ERROR.LOG:** ERROR-Datagramme

**INVALID.LOG:** unzulässige FND-Datagramme

**Hinweis:** Diese Dateien dienen ausschließlich Testzwecken und werden kontinuierlich fortgeschrieben, d.h. sie werden immer größer. Der Benutzer trägt die Verantwortung dafür, dass der Ablauf des GA-Knotens dadurch nicht beeinträchtigt wird, d.h. er muß sicherstellen, dass eine regelmäßige Überwachung der verfügbaren Festplatten-Kapazität und Löschung der LOG-Dateien durchgeführt wird.

### **5.8.7. Definition der Datenübertragung**

Die Datenübertragung zu den angeschlossenen IZ entspricht denen der verwendeten IZ-Treiber (vgl. unter IZ, IZnn).

Die Datenübertragung zu der angeschlossenen Leitzentrale ist abhängig vom verwendeten Protokoll und der Schnittstelle. Diese werden nachfolgend beschrieben.

### **5.9. FND 1.0 über V.24 (LZRS)**

**LZRSLIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit der Kommunikation mittels FND 1.0 zur Leitzentrale bzw. GA-Knoten über V.24 (RS 232) zusammenhängen .

**Hinweis:** Diese Bibliothek steht derzeit nur unter DOS zur Verfügung !

### 5.9.1. Übersicht der verwendeten Module

UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### 5.9.2. Liste mit Namen und Aufgaben

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

LZ_INIT	Initialisierung beim Programmstart
LZ_RSP_USM	Übernahme und Bearbeitung der von den IZ bzw. den Modulen GAK oder SSA gesendeten RSP- und USM-Datagramme
LZ_POLL	Polling für die zyklische Bearbeitung der Kommunikation zu den Leitzentralen
LZ_STAT	Übergabe des aktuellen LZ-Status
LZ_ZA_TIME	Übergabe der von der LZ gelesenen Datum/Zeit (hier OHNE Funktion)
LZ_EXIT	Ende-Behandlung beim Programmstop

### 5.9.3. Aufrufstruktur

LZ_INIT
LZ_COM_INIT
LZ_COM_CIN
LZ_RSP_USM
LZ_POLL
LZ_COM_CIN
LZ_COM_COUT

LZ\_COM\_SET\_DTR

LZ\_COM\_STAT

LZ\_COM\_EVNT

LZ\_STAT

LZ\_ZA\_TIME

LZ\_EXIT

LZ\_COM\_STAT

LZ\_COM\_EXIT

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

**5.9.4. Modulbeschreibungen**

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/RS LZ mit RS 232   Stand    :   01.10.98  */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   LZ_INIT ( )           */
/*****
/*   Beschreibung :
/*   Diese Routine führt die spezifischen Initialisierungen für die
/*   LZ nach dem Starten des Hauptprogrammes durch :
/*   - Laden der LZ-spezifischen LS-Parameter aus der Konf.-Datei und
/*   spezifische Initialisierung
/*
/*   aufgerufen von :           GAK_/SSA_INIT
/*
/*   ruft auf      :           LZ_COM_INIT, LZ_COM_CIN, UT_WSL_INIT
/*                           UT_NORM_STRING, UT_DP_FINDFIRST,
/*                           SY_CALLOC
/*
/*   I/O-Kanäle   :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   int      LZ_INIT
/*                           Returnkode der Funktion
/*                           0=FALSE: KEIN Fehler
/*                           -1: OPEN-Fehler
/*                           -2: READ-Fehler
/*                           -3: WRITE-Fehler
/*                           -4: ALLOC-Fehler
/*                           -5: unzulässige LS-Zeile
/*                           -7: WSL_INIT-Fehler
/*                           -9: INIT-Fehler der LZ-Verb.
/*                           >0: Nummer der Zeile, in der
/*                           Datei, in der ein Fehler
/*                           erkannt wurde
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*****
/*
/*   Autor      Datum      Version   Kurzzeichen
/*   -----
/*   Original :   W. Fries   01.05.97   1.1
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/RS LZ mit RS 232   Stand    :   01.10.98  */
/*****
/*   Datei    :   RSP_USM.C              */
/*   Aufruf   :   LZ_RSP_USM ( cDatagramm ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine bearbeitet das FND-Datagramm "cDatagramm", indem */
/*   es dieses in die zugehörige Warteschlange für die Ausgabe an   */
/*   die LZ eingetragen wird.                                         */
/*   aufgerufen von :           GAK_/SSA_IIZ, GAK_/SSA_ILZ         */
/*   ruft auf      :           UT_WSL_WRIT, UT_RSP2USM              */
/*   I/O-Kanäle   :                               */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   char     cDatagramm          RSP- bzw. USM-Datagramm */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   int      LZ_RSP_USM          Returnkode der Funktion */
/*           0=FALSE: KEIN Fehler                          */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   *** KEINE ***                                         */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/* Original :   W. Fries     01.05.97   1.1           */
/* Änderung:                                         */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.2      */
/*      Modulname    :      LZ/RS LZ mit RS-Bus   Stand    :   01.10.98 */
/*****
/*      Datei       :      POLL.C                */
/*      Aufruf      :      LZ_POLL ( )          */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die zyklische Behandlung der SEND- und RECV- */
/*      Funktionen zur LZ durch. Dabei wird die RSP- vor der USM-Warte- */
/*      schlange ausgelesen, falls der SEND-Puffer leer ist, und an die */
/*      LZ gesendet. Werden CMD-Datagramme von der LZ empfangen, so wird */
/*      dieses in die Warteschlange eingereiht und, falls nicht auf ein */
/*      RSP-Datagramm gewartet wird, ein Zeiger auf das Datagramm als */
/*      Ergebnis geliefert; SONST ist das Ergebnis ein NULL-Zeiger.     */
/*      Hinweis: Treten beim SEND Fehler auf, so wird das FND-Datagramm */
/*      nach einer Wartezeit solange wiederholt, bis die Übertragung */
/*      klappt.
/*
/*      aufgerufen von :      GAK_/SSA_ILZ
/*
/*      ruft auf      :      LZ_COM_COUT, LZ_COM_CIN, LZ_COM_SET_DTR, */
/*      LZ_COM_EVNT, LZ_COM_STAT,
/*      UT_FNDREJECT, UT_WSL_WRIT, UT_WSL_READ,
/*      UT_DT_ADD, diverse Makros FND... und
/*      SET_FND...
/*
/*      I/O-Kanäle      :
/*
/*****
/*      Eingabe Parameter :
/*      -----
/*      Typ      Name      Dim.      Beschreibung
/*      *** KEINE ***
/*
/*      -----
/*      Ausgabe Parameter:
/*      -----
/*      Typ      Name      Dim.      Beschreibung
/*      char *  LZ_POLL      Zeiger auf das empfangenen FND-
/*      Datum durch
/*      SONST: NULL-Zeiger
/*
/*      -----
/*      Transiente Parameter :
/*      -----
/*      Typ      Name      Dim.      Beschreibung
/*      *** KEINE ***
/*
/*****
/*      Autor      Datum      Version      Kurzzeichen
/*      -----
/*      Original :      W. Fries      01.05.97      1.1
/*      Änderung:
/*
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/RS LZ mit RS 232   Stand    :   01.10.98  */
/*****
/*   Datei    :   STAT.C                 */
/*   Aufruf   :   LZ_STAT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert den LZ-internen Status als Ergebnis. */
/*   */
/*   aufgerufen von :           GAK_/SSA_SYSCHK           */
/*   */
/*   ruft auf :                       */
/*   */
/*   I/O-Kanäle :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   uns.long LZ_STAT           LZ-Status bitweise kodiert */
/*   ( vgl. LZ_INCL.H) */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version   Kurzzeichen           */
/*   ----- */
/*   Original :   W. Fries      01.05.97   1.1 */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/RS LZ mit RS 232   Stand    :   01.10.98   */
/*****
/*   Datei    :   ZA_TIME.C             */
/*   Aufruf   :   LZ_ZA_TIME ( )       */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert die LZ-Systemzeit für den Zeitabgleich, */
/*   falls diese von VB_CONN bereitgestellt wurde.                */
/*   aufgerufen von :           GAK_/SSA_SYSCHK                    */
/*   ruft auf      :           *** KEINE ***                        */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                    */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                    */
/*   ----- */
/*   uns.long LZ_ZA_TIME           LZ-Systemzeit im DOS-Format    */
/*   STETS: UNDEF, da keine LZ-Zeit */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                    */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.05.97   1.1           */
/*   Änderung:                                     */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/RS LZ mit RS 232   Stand    :   01.10.98  */
/*****
/*   Datei    :   EXIT.C                 */
/*   Aufruf   :   LZ_EXIT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*   LZ vor dem Stoppen des Hauptprogrammes durch :                */
/*   - Schließen der Warteschlange für Datagramme von bzw. zur LZ  */
/*   - Freigabe ALLOCierter Speicherbereiche                       */
/*   - Trennen der Verbindung zur LZ                               */
/*   */
/*   aufgerufen von :           GAK_/SSA_EXIT                       */
/*   */
/*   ruft auf      :           LZ_COM_STAT, LZ_COM_EXIT,            */
/*                   UT_WSL_EXIT, SY_FREE                          */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   */
/*   int      LZ_EXIT                Returnkode der Funktion        */
/*                                           0=FALSE: KEIN Fehler        */
/*                                           -1: OPEN-Fehler           */
/*                                           -2: READ-Fehler          */
/*                                           -3: WRITE-Fehler         */
/*                                           -4: ALLOC-Fehler         */
/*                                           -5: unzulässige IS-Zeile  */
/*                                           -6: DDC-IZ-Init-Fehler   */
/*                                           >0: Nummer der Zeile, in der */
/*                                           ein Fehler erkannt wurde  */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### 5.9.5. Druckausgaben, Meldungstexte

\*\*\* KEINE \*\*\*

### 5.9.6. Datenkatalog

Im aktuellen Arbeitsverzeichnis werden temporär die folgenden Dateien angelegt:

**LZ\_ICMD.WSL:** Warteschlange der vom GA-Knoten empfangenen CMD-Datagramme

**LZ\_ORSP.WSL:** Warteschlange der an den GA-Knoten zu sendenden RSP-Datagramme

**LZ\_OUSM.WSL:** Warteschlange der an den GA-Knoten zu sendenden USM-Datagramme

### 5.9.7. Definition der Datenübertragung

Die Leitzentrale bzw. der GA-Knoten ist über eine serielle Schnittstelle (COM1 - COM4) angeschlossen, wobei nur Datagramme gemäß FND 1.0 übertragen werden.

Die Schnittstellen-Parameter und ggf. die FIFO-Behandlung müssen vor Aufruf des Programmes mittels des MODE-Befehls und COM\_FIFO gesetzt werden.

### 5.10. 3964R/RK512 über V.24 (LZEY)

**LZEYLIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit der Kommunikation mittels des Protokolles 3964R/RK512 zur Leitzentrale über V.24 (RS 232) zusammenhängen .

**Hinweis: Diese Bibliothek steht derzeit nur unter DOS zur Verfügung !**

### 5.10.1.Übersicht der verwendeten Module

UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### 5.10.2.Liste mit Namen und Aufgaben

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

LZ_INIT	Initialisierung beim Programmstart
LZ_RSP_USM	Übernahme und Bearbeitung der von den IZ bzw. den Modulen GAK oder SSA gesendeten RSP- und USM-Datagramme
LZ_POLL	Polling für die zyklische Bearbeitung der Kommunikation zu den Leitzentralen
LZ_STAT	Übergabe des aktuellen LZ-Status
LZ_ZA_TIME	Übergabe der von der LZ gelesenen Datum/Zeit (hier OHNE Funktion)
LZ_EXIT	Ende-Behandlung beim Programmstop

### 5.10.3.Aufrufstruktur

LZ_INIT
LZ_COM_INIT
LZ_COM_CIN
LZ_RSP_USM
LZ_POLL
LZ_EY_LRC
LZ_COM_CIN

LZ\_COM\_COUT

LZ\_COM\_PURG

LZ\_COM\_STAT

LZ\_STAT

LZ\_ZA\_TIME

LZ\_EXIT

LZ\_COM\_STAT

LZ\_COM\_EXIT

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

**5.10.4. Modulbeschreibungen**

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus    Stand    :   01.10.98  */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   LZ_INIT ( )           */
/*****
/*   Beschreibung :
/*   Diese Routine führt die spezifischen Initialisierungen für die
/*   Sauter LZ nach dem Starten des Hauptprogrammes durch : */
/*   - Laden der LZ-spezifischen LS-Parameter aus der Konf.-Datei und*/
/*     spezifische Initialisierung
/*
/*   aufgerufen von :           GAK_/SSA_INIT
/*
/*   ruft auf      :           LZ_COM_INIT, LZ_COM_CIN, UT_WSL_INIT,
/*                             UT_NORM_STRING, UT_DP_FINDFIRST,
/*                             SY_CALLOC
/*
/*   I/O-Kanäle   :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   int      LZ_INIT
/*
/*           Returnkode der Funktion
/*           0=FALSE: KEIN Fehler
/*           -1: OPEN-Fehler
/*           -2: READ-Fehler
/*           -3: WRITE-Fehler
/*           -4: ALLOC-Fehler
/*           -5: unzulässige LS-Zeile
/*           -7: WSL_INIT-Fehler
/*           -9: INIT-Fehler der LZ-Verb.
/*           >0: Nummer der Zeile, in der
/*               Datei, in der ein Fehler
/*               erkannt wurde
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   *** KEINE ***
/*****
/*
/*           Autor      Datum      Version      Kurzzeichen
/*           -----
/*   Original :   W. Fries      01.05.97      1.1
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus   Stand    :   01.10.98  */
/*****
/*   Datei    :   RSP_USM.C             */
/*   Aufruf   :   LZ_RSP_USM ( cDatagramm ) */
/*****
/*   Beschreibung :
/*   Diese Routine bearbeitet das FND-Datagramm "cDatagramm", indem
/*   es dieses schlicht und einfach ignoriert, da die Fa. Sauter
/*   keine SEND-Datagramme des 3964R-Protokolles unterstützt.
/*   Der neue Wert wird in GAK_IIZ in der DP-Struktur vermerkt
/*   und beim nächsten FETCH des zyklischen Pollings der LZ von dort
/*   ausgelesen.
/*
/*   aufgerufen von :           GAK_IIZ, GAK_ILZ
/*
/*   ruft auf       :           *** KEINE ***
/*
/*   I/O-Kanäle    :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   char     cDatagramm          RSP- bzw. USM-Datagramm
/*
/* - - - - -
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   int      LZ_RSP_USM          Returnkode der Funktion
/*                                     0=FALSE: KEIN Fehler
/*
/* - - - - -
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version   Kurzzeichen
/*   -----
/* Original :   W. Fries    01.05.97    1.1
/* Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus    Stand    :   01.10.98  */
/*****
/*   Datei    :   POLL.C                 */
/*   Aufruf   :   LZ_POLL ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die zyklische Behandlung der SEND- und RECV- */
/*   Funktionen zur LZ durch. Dabei wird die OLZ-Warteschlange      */
/*   ausgelesen, und falls der SEND-Puffer leer ist, an die LZ      */
/*   gesendet. Werden Datagramme von der LZ empfangen, so wird     */
/*   ein Zeiger auf das Datagramm als Ergebnis geliefert; SONST ist  */
/*   das Ergebnis ein NULL-Zeiger.                                     */
/*   */
/*   aufgerufen von :           GAK_/SSA_ILZ                          */
/*   */
/*   ruft auf      :           LZ_EY_LRC, LZ_COM_COUT, LZ_COM_CIN,    */
/*   */
/*   */
/*   LZ_COM_PURG, LZ_COM_STAT,                                     */
/*   */
/*   UT_KL_BW2PW, UT_KL_PW2BW, UT_DT_ADD,                         */
/*   */
/*   UT_FND_TIME, diverse Makros FND_... und                      */
/*   SET_FND_..., SY_F2IEEE                                       */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                                           */
/*   -----                                                       */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   */
/*   *** KEINE ***                                               */
/*   */
/*   -----                                                       */
/*   Ausgabe Parameter:                                           */
/*   -----                                                       */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   */
/*   char *  LZ_POLL                Zeiger auf das empfangenen FND- */
/*   */
/*   */
/*   */
/*   Datum durch                                                    */
/*   SONST: NULL-Zeiger                                           */
/*   */
/*   -----                                                       */
/*   Transiente Parameter :                                         */
/*   -----                                                       */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   */
/*   *** KEINE ***                                               */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen                    */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1                      */
/*   Änderung:                                         */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus   Stand    :   01.10.98  */
/*****
/*   Datei    :   STAT.C                 */
/*   Aufruf   :   LZ_STAT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert den LZ-internen Status als Ergebnis. */
/*   */
/*   aufgerufen von :   GAK_/SSA_SYSCHK  */
/*   */
/*   ruft auf      :   *** KEINE ***     */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                 */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                 */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   uns.long LZ_STAT          LZ-Status bitweise kodiert */
/*   ( vgl. LZ_INCL.H) */
/*   */
/*   ----- */
/*   Transiente Parameter :             */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries    01.05.97    1.1         */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus    Stand    :   01.10.98  */
/*****
/*   Datei    :   ZA_TIME.C              */
/*   Aufruf   :   LZ_ZA_TIME ( )        */
/*****
/*   Beschreibung :                      */
/*   Diese Routine liefert die LZ-Systemzeit für den Zeitabgleich, */
/*   falls diese von VB_CONN bereitgestellt wurde.                */
/*                                                                */
/*   aufgerufen von :           GAK_SYSCHK                            */
/*                                                                */
/*   ruft auf      :           UT_ZA_GETZA, UT_FNDNX2DT              */
/*                                                                */
/*   I/O-Kanäle   :                      */
/*                                                                */
/*****
/*   Eingabe Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*                                                                */
/*   *** KEINE ***                      */
/*                                                                */
/* -----                      */
/*   Ausgabe Parameter:                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*                                                                */
/*   uns.long LZ_ZA_TIME           LZ-Systemzeit im DOS-Format      */
/*                                                                */
/*                                                                */
/* -----                      */
/*   Transiente Parameter :                      */
/*   -----                      */
/*   Typ      Name           Dim.   Beschreibung                      */
/*                                                                */
/*   *** KEINE ***                      */
/*                                                                */
/*****
/*           Autor      Datum      Version      Kurzzeichen      */
/*   -----                      */
/* Original :   W. Fries      01.05.97      1.1                      */
/* Änderung:                      */
/*                                                                */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus   Stand    :   01.10.98  */
/*****
/*   Datei    :   EXIT.C                 */
/*   Aufruf   :   LZ_EXIT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*   LZ vor dem Stoppen des Hauptprogrammes durch :                */
/*   - Schließen der Warteschlange für Datagramme von bzw. zur LZ  */
/*   - Freigabe ALLOCierter Speicherbereiche                       */
/*   - Trennen der Verbindung zur LZ                               */
/*   */
/*   aufgerufen von :           GAK_/SSA_EXIT                       */
/*   */
/*   ruft auf      :           LZ_COM_STAT, LZ_COM_EXIT,            */
/*                   UT_WSL_EXIT, SY_FREE                          */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   */
/*   int      LZ_EXIT                Returnkode der Funktion         */
/*                                               0=FALSE: KEIN Fehler         */
/*                                               -1: OPEN-Fehler             */
/*                                               -2: READ-Fehler            */
/*                                               -3: WRITE-Fehler           */
/*                                               -4: ALLOC-Fehler           */
/*                                               -5: unzulässige IS-Zeile   */
/*                                               -6: DDC-IZ-Init-Fehler    */
/*                                               >0: Nummer der Zeile, in der */
/*                                               ein Fehler erkannt wurde   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                       */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1           */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.2       */
/*   Modulname :   LZ/EY LZ mit EY-Bus    Stand    :   01.10.98  */
/*****
/*   Datei    :   EY_LRC.C               */
/*   Aufruf   :   LZ_EY_LRC ( cBuf, iLen ) */
/*****
/*   Beschreibung :                       */
/*   Diese Routine berechnet für "cBuf" der Länge "iLen" den LRC. */
/*   Der Algorithmus ist in Anlehnung an die Routine "LrcCheck" der */
/*   Fa. Sauter mit Fax vom 11.11.96 programmiert.                */
/*   Die Sonderbehandlung für die DLE-Verdoppelung entfällt, da  */
/*   diese bereits beim CTN in der Routine POLL behandelt wird.    */
/*   */
/*   aufgerufen von :           LZ_POLL */
/*   */
/*   ruft auf      :           *** KEINE *** */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   ----- */
/*   char*    cBuf           */   Datenrahmen und */
/*   int      iLen           */   Anzahl der Zeichen darin, für */
/*   */   die das LRC berechnet werden */
/*   */   soll. */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   ----- */
/*   unsigned LZ_EY_LRC           LRC-Wert zu "cBuf" */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.05.97      1.1 */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

### 5.10.5. Druckausgaben, Meldungstexte

\*\*\* KEINE \*\*\*

### 5.10.6. Datenkatalog

Im aktuellen Arbeitsverzeichnis werden temporär die folgenden Dateien angelegt:

**LZ\_ILZ.WSL:** Warteschlange der von der LZ empfangenen 3964R/RK512-Datagramme

### 5.10.7. Definition der Datenübertragung

Die Leitzentrale ist über eine serielle Schnittstelle (COM1 - COM4) angeschlossen, wobei nur Datagramme gemäß 3964R/RK512 im Pollingverfahren übertragen werden, d.h. die Leitzentrale fragt die Datenpunkte zyklisch ab. Der SSA sendet keine spontanen Datagramme.

Die Schnittstellen-Parameter werden beim Programmstart gesetzt, jedoch muß ggf. der FIFO vor Aufruf des Programmes mittels COM\_FIFO aktiviert werden.

## 5.11. LonTalk mit FND 1.0 über V.24 (LZLN)

**LZLNLIB** In dieser Bibliothek werden alle Routinen zusammengefaßt, die mit der Kommunikation mittels des Protokolles LonTalk mit FND 1.0 zur Leitzentrale über V.24 (RS 232) zusammenhängen .

**Hinweis:** Dieser Treiber kommt vorläufig nur als Spezial-Software in Verbindung mit dem Treiber für das AS511-Protokoll (IZ) auf Basis des GA-Knotens unter LINUX zum Einsatz (vgl. Unterverzeichnis IZ\AS511 und GAK\SSA\_S5\_L.EXE)

### 5.11.1.Übersicht der verwendeten Module

UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### 5.11.2.Liste mit Namen und Aufgaben

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

LZ_INIT	Initialisierung beim Programmstart
LZ_RSP_USM	Übernahme und Bearbeitung der von den IZ bzw. den Modulen GAK oder SSA gesendeten RSP- und USM-Datagramme
LZ_POLL	Polling für die zyklische Bearbeitung der Kommunikation zu den Leitzentralen
LZ_STAT	Übergabe des aktuellen LZ-Status
LZ_ZA_TIME	Übergabe der von der LZ gelesenen Datum/Zeit (hier OHNE Funktion)
LZ_EXIT	Ende-Behandlung beim Programmstop
LZ_RECV_ANZ	Anzahl empfangener Datagramme
LZ_SEND_ANZ	Anzahl gesendeter Datagramme

### 5.11.3.Aufrufstruktur

LZ\_INIT  
    LZ\_LON\_ADR  
    LZ\_COM\_INIT  
    LZ\_COM\_CIN

LZ\_RSP\_USM

LZ\_POLL

LZ\_LON\_CRC

LZ\_COM\_CIN

LZ\_COM\_COUT

LZ\_COM\_PURG

LZ\_COM\_STAT

LZ\_STAT

LZ\_ZA\_TIME

LZ\_EXIT

LZ\_COM\_STAT

LZ\_COM\_EXIT

LZ\_RECV\_ANZ

LZ\_SEND\_ANZ

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

**5.11.4. Modulbeschreibungen**

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   LZ_INIT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die spezifischen Initialisierungen für die */
/*   LZ-Anbindung über FND/LON nach dem Starten des Hauptprogrammes */
/*   durch :                               */
/*   - Laden der LZ-spezifischen LS-Parameter aus der Konf.-Datei und */
/*     spezifische Initialisierung         */
/*                                           */
/*   aufgerufen von :   GAK_/SSA_INIT     */
/*                                           */
/*   ruft auf      :   LZ_LON_ADR,       */
/*                   LZ_COM_INIT, LZ_COM_CIN, UT_WSL_INIT,   */
/*                   UT_NORM_STRING, UT_DP_FINDFIRST,        */
/*                   SY_CALLOC           */
/*                                           */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :                   */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung   */
/*   -----                               */
/*   *** KEINE ***                         */
/*   -----                               */
/*   Ausgabe Parameter:                   */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung   */
/*   -----                               */
/*   int      LZ_INIT                                     */
/*                                           */
/*   Returnkode der Funktion                 */
/*   0=FALSE: KEIN Fehler                    */
/*   -1: OPEN-Fehler                         */
/*   -2: READ-Fehler                         */
/*   -3: WRITE-Fehler                        */
/*   -4: ALLOC-Fehler                        */
/*   -5: unzulässige LS-Zeile                */
/*   -7: WSL_INIT-Fehler                     */
/*   -9: INIT-Fehler der LZ-Verb.           */
/*   >0: Nummer der Zeile, in der           */
/*       Datei, in der ein Fehler           */
/*       erkannt wurde                       */
/*   -----                               */
/*   Transiente Parameter :                 */
/*   -----                               */
/*   Typ      Name      Dim.   Beschreibung   */
/*   -----                               */
/*   *** KEINE ***                         */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----                               */
/*   Original :   W. Fries      01.04.2006   1.3j     */
/*   Änderung:                                     */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   RSP_USM.C             */
/*   Aufruf    :   LZ_RSP_USM ( cDatagramm ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine bearbeitet das FND-Datagramm "cDatagramm", indem */
/*   es dieses in die zugehörige Warteschlange für die Ausgabe an */
/*   die LZ eingetragen wird. */
/*   */
/*   aufgerufen von :   GAK_/SSA_IIZ, GAK_/SSA_ILZ */
/*   */
/*   ruft auf      :   UT_WSL_WRIT, UT_RSP2USM */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung */
/*   char     cDatagramm          RSP- bzw. USM-Datagramm */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung */
/*   int      LZ_RSP_USM          Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/* Original :   W. Fries      01.04.2006      1.3j */
/* Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Verson   :   1.3j       */
/*   Modulname :   LZ/LON LZ mit FND/LON  Stand    :   01.04.2006  */
/*****
/*   Datei     :   POLL.C                 */
/*   Aufruf    :   LZ_POLL ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die zyklische Behandlung der SEND- und RECV-*/
/*   Funktionen zur LZ durch. Dabei wird die RSP- vor der USM-Warte-*/
/*   schlange ausgelesen, falls der SEND-Puffer leer ist, und an die */
/*   LZ gesendet. Werden CMD-Datagramme von der LZ empfangen, so wird*/
/*   dieses in die Warteschlange eingereiht und, falls nicht auf ein */
/*   RSP-Datagramm gewartet wird, ein Zeiger auf das Datagramm als */
/*   Ergebnis geliefert; SONST ist das Ergebnis ein NULL-Zeiger.     */
/*                                                                    */
/*   Bei "Kollision" zu sender und empfangener Datagramme unter-    */
/*   drückt der SLTA manchmal, jedoch nicht immer das "niRESPONSE"  */
/*   und das FND-RSP. Da die Kollision nicht sicher erkannt werden   */
/*   kann, werden vorläufig die Timeout-Zeiten gesplittet und die    */
/*   Wartezeit vor der Sende-Wiederholung deaktiviert.              */
/*   Desweiteren wird die sofortige LOOP-Bearbeitung für "niACK"    */
/*   erzwungen, d.h. die Wartezeiten von SY_SIG_WAIT unterdrückt.   */
/*   Dadurch werden die "Kollisionen" zwar nicht verhindert, jedoch */
/*   deren Wahrscheinlichkeit reduziert.                             */
/*   Das FND-CMD wird in 2 Teilen (Header+Daten) gesendet, damit    */
/*   weniger als 16 Zeichen von der Kollision betroffen sind und vom */
/*   SLTA zwischengespeichert werden können. Vor Senden der Daten   */
/*   wird erneut auf Kollision geprüft und das Senden ggf. auf das  */
/*   Ende des empfangenen Datagrammes verschoben.                   */
/*                                                                    */
/*   aufgerufen von :          GAK_/SSA_ILZ                          */
/*                                                                    */
/*   ruft auf      :          LZ_LON_CRC,                             */
/*                  LZ_COM_SYNC, LZ_COM_COUT,                       */
/*                  LZ_COM_CIN,  LZ_COM_PURG,                       */
/*                  LZ_COM_STAT, LZ_COM_EVNT,                       */
/*                  UT_DT_ADD,                                       */
/*                  UT_FNDREJECT, UT_WSL_READ, UT_FND_TIME,        */
/*                  diverse Makros FND_... und SET_FND_...,         */
/*                  SY_GETTICKS, SY_SIG_INKR                         */
/*                                                                    */
/*   I/O-Kanäle   :          */
/*****
/*   Eingabe Parameter :                                           */
/*   -----                                                    */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   *** KEINE ***                                                    */
/*   -----                                                    */
/*   Ausgabe Parameter:                                           */
/*   -----                                                    */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   char *  LZ_POLL                               Zeiger auf das empfangenen FND-*/
/*                                                  Datagramm durch          */
/*                                                  SONST: NULL-Zeiger      */
/*   -----                                                    */
/*   Transiente Parameter :                                         */
/*   -----                                                    */
/*   Typ      Name          Dim.   Beschreibung                    */
/*   *** KEINE ***                                                    */
/*****
/*   Autor      Datum      Version  Kurzzeichen                    */
/*   -----                                                    */
/*   Original :   W. Fries    01.04.2006   1.3j                    */
/*   Änderung:                                         */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   STAT.C                 */
/*   Aufruf    :   LZ_STAT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert den LZ-internen Status als Ergebnis. */
/*   */
/*   aufgerufen von :           GAK_/SSA_SYSCHK           */
/*   */
/*   ruft auf      :           */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***           */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   uns.long  LZ_STAT           LZ-Status bitweise kodiert */
/*   ( vgl. LZ_INCL.H)           */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE ***           */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen           */
/*   ----- */
/*   Original :   W. Fries      01.04.2006   1.3j           */
/*   Änderung:           */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   ZA_TIME.C             */
/*   Aufruf    :   LZ_ZA_TIME ( )       */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert die LZ-Systemzeit für den Zeitabgleich, */
/*   falls diese von VB_CONN bereitgestellt wurde.                */
/*   */
/*   aufgerufen von :           GAK_/SSA_SYSCHK                    */
/*   */
/*   ruft auf      :           *** KEINE ***                        */
/*   */
/*   I/O-Kanäle   :                       */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                    */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                    */
/*   */
/*   uns.long LZ_ZA_TIME           LZ-Systemzeit im DOS-Format */
/*   STETS: UNDEF, da keine LZ-Zeit */
/*   */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung                    */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*           Autor      Datum      Version      Kurzzeichen   */
/*   ----- */
/* Original :   W. Fries      01.04.2006      1.3j           */
/* Änderung:                                     */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   EXIT.C                 */
/*   Aufruf    :   LZ_EXIT ( )           */
/*****
/*   Beschreibung :                       */
/*   Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*   LZ vor dem Stoppen des Hauptprogrammes durch :                */
/*   - Schließen der Warteschlange für Datagramme von bzw. zur LZ  */
/*   - Freigabe ALLOCierter Speicherbereiche                       */
/*   - Trennen der Verbindung zur LZ                               */
/*   */
/*   aufgerufen von :           GAK_/SSA_EXIT                       */
/*   */
/*   ruft auf      :           LZ_COM_STAT, LZ_COM_EXIT,           */
/*                   UT_WSL_EXIT, SY_FREE                          */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   int      LZ_EXIT                Returnkode der Funktion */
/*                                           0=FALSE: KEIN Fehler */
/*                                           -1: OPEN-Fehler */
/*                                           -2: READ-Fehler */
/*                                           -3: WRITE-Fehler */
/*                                           -4: ALLOC-Fehler */
/*                                           -5: unzulässige IS-Zeile */
/*                                           -6: DDC-IZ-Init-Fehler */
/*                                           >0: Nummer der Zeile, in der */
/*                                           ein Fehler erkannt wurde */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version   Kurzzeichen           */
/*   ----- */
/*   Original :   W. Fries      01.04.2006   1.3j           */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   RECV_ANZ.C           */
/*   Aufruf    :   LZ_RECV_ANZ ( )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert den LZ-internen Zähler für gesendete */
/*   Datagramme als Ergebnis.   */
/*   */
/*   aufgerufen von :   GAK_SYSCHK   */
/*   */
/*   ruft auf      :   *** KEINE ***   */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   uns.long LZ_RECV_ANZ   Anzahl gesendeter Datagramme */
/*   ( vgl. LZ_INCL.H)   */
/*   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries   01.04.2006   1.3j   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3j      */
/*      Modulname   :      LZ/LON LZ mit FND/LON  Stand    :   01.04.2006 */
/*****
/*      Datei      :      SEND_ANZ.C           */
/*      Aufruf     :      LZ_SEND_ANZ ( )      */
/*****
/*      Beschreibung :                               */
/*      Diese Routine liefert den LZ-internen Zähler für gesendete */
/*      Datagramme als Ergebnis.                               */
/*      aufgerufen von :          GAK_SYSCHK      */
/*      ruft auf      :          *** KEINE ***     */
/*      I/O-Kanäle   :                               */
/*****
/*      Eingabe Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung      */
/*      -----                               */
/*      *** KEINE ***                               */
/*      -----                               */
/*      Ausgabe Parameter:                               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung      */
/*      -----                               */
/*      uns.long LZ_SEND_ANZ          Anzahl gesendeter Datagramme */
/*      ( vgl. LZ_INCL.H)                               */
/*      -----                               */
/*      Transiente Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.   Beschreibung      */
/*      -----                               */
/*      *** KEINE ***                               */
/*****
/*      Autor      Datum      Version   Kurzzeichen   */
/*      -----                               */
/*      Original :   W. Fries      01.04.2006   1.3j      */
/*      Änderung:                               */
/*      -----                               */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   LON_ADR.C             */
/*   Aufruf    :   LZ_LON_ADR ( cAdr )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine wandelt die LON-Adresse "cAdr" im ASCII-Format */
/*   "[D.]SSS.NNN:III" in die binäre Form (vgl. LZLNINCL.H). Im */
/*   Fehlerfall wird UNDEF als Ergebnis geliefert.             */
/*   */
/*   aufgerufen von :   LZ_INIT          */
/*   */
/*   ruft auf       :   UT_NORM_STRING   */
/*   */
/*   I/O-Kanäle     :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----             */
/*   Typ      Name          Dim.   Beschreibung             */
/*   char*    cAdr          */     LON-Adresse im ASCII-Format */
/*   -----             */
/*   Ausgabe Parameter:   */
/*   -----             */
/*   Typ      Name          Dim.   Beschreibung             */
/*   uns.long  LZ_LON_ADR    */     LON-Adresse im binären Format */
/*   SONST: 0                                     */
/*   -----             */
/*   Transiente Parameter :   */
/*   -----             */
/*   Typ      Name          Dim.   Beschreibung             */
/*   *** KEINE ***                                     */
/*   -----             */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----             */
/*   Original :   W. Fries    01.04.2006   1.3j         */
/*   Änderung:                                     */
/*   -----             */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3j   */
/*   Modulname :   LZ/LON LZ mit FND/LON   Stand    :   01.04.2006 */
/*****
/*   Datei     :   LON_CRC.C             */
/*   Aufruf    :   LZ_LON_CRC ( ucString ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine berechnet für das LON-Datagramm "ucString" den */
/*   8-Bit-CRC-Wert gemäß "Serial LonTalk Adapter and Serial Gateway */
/*   User's Guide (vgl. Seite 8-6). */
/*   */
/*   aufgerufen von :   LZ_POLL */
/*   */
/*   ruft auf      :   */
/*   */
/*   I/O-Kanäle    :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   uns.char* ucString          Auszuwertender String */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   uns. char LZ_LON_CRC          CRC-Wert zu "ucString" */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.04.2006   1.3j */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



### **5.11.5. Druckausgaben, Meldungstexte**

\*\*\* KEINE \*\*\*

### **5.11.6. Datenkatalog**

Im aktuellen Arbeitsverzeichnis werden temporär die folgenden Dateien angelegt:

**LZ\_ICMD.WSL:** Warteschlange der von der LZ empfangenen CMD-Datagramme

**LZ\_ORSP.WSL:** Warteschlange der an die LZ zu sendenden RSP-Datagramme (vorrangig)

**LZ\_OUSM.WSL:** Warteschlange der an die LZ zu sendenden USM-Datagramme

### **5.11.7. Definition der Datenübertragung**

Grundlage für die Realisierung sind :

- bereitgestellte Dokumentation
- Spezifikation "Anschluss des LON an den GA-Knoten"
- Angebot AWMMIHDE/6021 vom 07.02.2006

**Hinweis:** Der Funktions-Umfang dieses Treibers bleibt vorerst beschränkt auf die Anforderungen als Ersatz für die o.g. Software im Projekt "Sickerwasser-Aufbereitung" (SiWa) im Entsorgungspark Freimann des Abfallwirtschaftsbetrieb München (AWM-ESP) gemäß dem o.g. Angebot, d.h. insbesondere :

- nur lesende Zugriffe
- nur Melde- und Messpunkte

Folgende Abweichungen wurden festgelegt :

\*\*\* KEINE \*\*\*

**V.24-Schnittstelle zum Anschluss des LON**

Der LON wird über einen SLTA/2 o.ä. an eine serielle Schnittstelle (RS 232 C) des GA-Knotens angeschlossen (COM1 - COM4). Zur Entkopplung der Ein-/Ausgabe auf dieser Schnittstelle werden diese Funktionen über Puffer und Interrupt-Service-Routinen (ISR) realisiert. Es werden folgende Schnittstellenparameter verwendet und beim Programmstart initialisiert :

9600 Baud, 1 Startbit, 8 Datenbit, NO-Parity-Bit und 1 Stopbit

Unter LINUX muss, abweichend von der Spezifikation "Anschluss des LON an den GA-Knoten", als Link-Protokoll statt "ALERT/ACK" nun "Buffered" am SLTA eingestellt werden.

**Hinweis:** Der SLTA ist nur halbduplex-fähig, unterstützt aber KEIN stabiles Handshake-Verfahren. Stattdessen erwartet er, dass der Host-Rechner seinen Sendevorgang unterbricht, sobald der SLTA sendet. Aufgrund der bei Mehrbenutzersystemen üblichen Latenzzeiten (LINUX: typ. 20 ms) kann es zu "Kollisionen" der gesendeten Datagramme kommen, in deren Verlauf der SLTA das vom Host gesendete Datagramm "vergisst".

Da in diesem Fall seitens des SLTA keine Fehlermeldung erfolgt, kann der Host das Problem nur durch das Ausbleiben der Bestätigung nach Ablauf einer Wartezeit erkennen.

Auch der von ECHELON in einem Beispiel-Treiber vorgeschlagene Workaround reduziert nur die Häufigkeit, beseitigt aber das Problem nicht grundlegend. Als Folge davon können sporadisch Verzögerungen von einigen Sekunden und die Wiederholung von Datagrammen auftreten.

Eine Verbesserung schafft die SLTA-Emulation der Fa. MSR electronic GmbH, Pocking. Diese unterstützt alle für diesen Treiber erforderlichen Datagramme und puffert bis zu 128 Zeichen zwischen, so dass keine Zeichen mehr verloren gehen können.

**Spezifische Anpassungen in der Konfigurationsdatei**

LZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"LZ"	Zeilen-Kennung (OHNE ''')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen

					<p>wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten und wird abhängig davon vorgegeben.                  0: reserviert für zukünftige Erweiterungen</p> <p>*** Wird noch nicht ausgewertet                  *** und bleibt reserviert für                  *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt :                  *** LON: 0</p>
3	8	10	3	Treiber-Subid.	<p>lfd. Nummer der an den Treiber angeschlossenen IZ beginnend mit 0.</p> <p>Für die direkte Verbindung zum GA_Knoten über RS 232 ist dieser Wert stets 0</p>
4	12	34	23	LZ-/Treiber-Text	<p>Klartext zur Beschreibung des LON bzw. des Treibers</p>
5	36	38	3	Verbindungs-Art	<p>natürliche Zahl kodiert gemäß "Leistungsverzeichnis eines GA-Knotens" vom 29.8.94 (vgl. Seite 20).</p> <p>*** Wird noch nicht ausgewertet                  *** und bleibt reserviert für                  *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt :                  *** LON: 1</p>
6	40	42	3	Protokoll	<p>nat. Zahl kodiert wie folgt :</p> <p>*** Wird noch nicht ausgewertet                  *** und bleibt reserviert für                  *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt :                  *** LON: 0</p>
7	44	46	3	Subnet_SK	<p>Subnet-Adr. des System-Knotens</p> <p>*** Wird noch nicht ausgewertet                  *** und bleibt reserviert für                  *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt :                  *** LON: 0</p>
8	48	50	3	Node_SK	<p>Node-Adr. des System-Knotens Falls "Subnet_SK" und "Node_SK" auf 0 gesetzt sind, wird die Behandlung des System-Knotens unterdrückt</p> <p>*** Wird noch nicht ausgewertet                  *** und bleibt reserviert für                  *** zukünftige Erweiterungen !</p> <p>*** Bis dahin gilt :                  *** LON: 0</p>
9	52	256	<=205	Schnittst.-Bez.	<p>Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem IZ-Typ.</p> <p>"COM1" - "COM4" (ohne "                  bzw.                  "/dev/ttyS0" ff.</p> <p>Wird durch das Zeilenabschlusszeichen (&lt;CR&gt; und/oder &lt;LF&gt;, je nach verwendetem Editor)</p>

abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

Gemäß der Spezifikation "Anschluss des LON an den GA-Knoten" werden die FND-Datenpunkte innerhalb des Treibers umgesetzt, d.h. es werden FND-Datagramme mit einem reduzierten APCI im Datenbereich des LON-Datagrammes übertragen.

Die LON-Adressen werden in den IS-Zeilen wie folgt kodiert:

**"[D.]SSS.NNN:III"** mit :

- D** ...: Index in Domain-Table (optional, Voreinstellung 0)
- SSS** .: Subnet
- NNN** .: Node
- III** .: Index des Datenpunktes im LON-Knoten

Die LS-Zeilen werden durch das Zeilenabschlusszeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die LS-Zeile in der Konfigurationsdatei hat ab Spalte 29 für Melde-, Schalt-, Mess-, Stell- und Zählpunkte hat folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	41	13	LON-Adresse	LON-Adresse gemäß obigem Format

**Hinweis:** Der Aufbau der LS-Zeile entspricht soweit der IS-Zeile des korrespondierenden IZ-Treibers für FND über LON.

**Initialisierungen**

Der Zugriff auf den LON wird beim Programmstart und nach jeder erkannten Störung neu initialisiert. Dabei erfolgt ein RESET, FLUSH\_CANCEL, SET NODE MODE und Abfragen der Neuron-Id.

## 5.12. BACnet/IP (LZBN)

**LZBNLIB** In dieser Bibliothek werden alle Routinen zusammengefasst, die mit der Kommunikation mittels des Protokolles BACnet/IP zur Leitzentrale/Management-Station zusammenhängen .

**Hinweis:** Dieser Treiber macht den GA-Knoten zu einem BACnet-Server (vgl. Unterverzeichnis GAK\bns\_...exe). Damit können alle IZ-seitig unterstützten MSR-Systeme auf BACnet-basierte Leitzentralen/Management-Stationen aufgeschaltet werden.

Das SP-Modul bleibt dabei integriert, so dass von GA-Knoten genutzte Schaltprogramme bei Bedarf unverändert ablauffähig sind, jedoch nicht mehr von der LZH aus modifiziert werden können. Hierzu ist der Abstimmung mit den Schedule-Objekten erforderlich.

Bei der Einrichtung des Rechners ist darauf zu achten, dass dieser dem bzw. den Netzwerken zugeordnet ist, über die es für BACnet-Geräte erreichbar sein soll (vgl. "hostname -i"). Ggf. sind in der Firewall die verwendeten Ports (z.B. TCP-Port 47808) freizuschalten.

### 5.12.1.Übersicht der verwendeten Module

IP	Internet Protocol (UDP/IP)
UT	allgemeine Routinen-Bibliothek
BS	allgemeine Routinen-Bibliothek zur Bildschirmsteuerung
SY	systemspezifische Routinen-Bibliothek

### 5.12.2.Liste mit Namen und Aufgaben

Als Aufruf-Schnittstelle für das Hauptprogramm dienen folgende Routinen:

LZ_INIT	Initialisierung beim Programmstart
LZ_RSP_USM	Übernahme und Bearbeitung der von den IZ bzw. den Modulen GAK oder SSA gesendeten RSP- und USM-Datagramme
LZ_POLL	Polling für die zyklische Bearbeitung der Kommunikation zu den Leitzentralen
LZ_STAT	Übergabe des aktuellen LZ-Status

LZ_ZA_TIME	Übergabe der von der LZ gelesenen Datum/Zeit (hier OHNE Funktion)
LZ_EXIT	Ende-Behandlung beim Programmstop
LZ_RECV_ANZ	Anzahl empfangener Datagramme
LZ_SEND_ANZ	Anzahl gesendeter Datagramme

### 5.12.3. Aufrufstruktur

LZ\_INIT

    LZ\_BN\_ADR

    LZ\_CAL\_INIT

    LZ\_SCD\_INIT

        LZ\_SCD\_START

            LZ\_SCD\_SYNC

LZ\_RSP\_USM

LZ\_POLL

    LZ\_SCD\_POLL

        LZ\_SCD\_START

            LZ\_SCD\_SYNC

        LZ\_SCD\_SYNC

    LZ\_BN\_MSGCHK

        LZ\_BN\_MSGNW

        LZ\_BN\_MSGRPR

        LZ\_BN\_MSGWPR

        LZ\_BN\_MSGGEI

LZ\_STAT

LZ\_ZA\_TIME

LZ\_EXIT

LZ\_RECV\_ANZ

LZ\_SEND\_ANZ

sowie diverse Routinen aus den oben beschriebenen Basis-Modulen

5.12.4. Modulbeschreibungen

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   INIT.C                 */
/*   Aufruf   :   LZ_INIT ( )           */
/*****
/*   Beschreibung :
/*   Diese Routine führt die spezifischen Initialisierungen für die
/*   LZ-Anbindung über BACnet/IP nach dem Starten des Hauptprogrammes*/
/*   durch :
/*   - Laden der LZ-spezifischen LS-Parameter aus der Konf.-Datei und*/
/*   spezifische Initialisierung
/*   - Initialisieren der Schnittstelle zur LZ
/*
/*   aufgerufen von :   GAK_/SSA_INIT
/*
/*   ruft auf :
/*   LZ_CAL_INIT, LZ_SCD_INIT,
/*   LZ_BN_OBJTYPE, LZ_BN_ADR,
/*   IP_INIT, IP_OPEN, IP_GET_HOST,
/*   IP_RECV, IP_SEND, IP_A2PORT,
/*   UT_DP_ATBMIND, UT_DP_FINDFIRST,
/*   UT_WSL_INIT, UT_NORM_STRING,
/*   SY_FI_FINDFIRST, SY_FI_FINDNEXT,
/*   SY_CALLOC, SY_WAIT
/*
/*   I/O-Kanäle :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   *** KEINE ***
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   int      LZ_INIT
/*           Returnkode der Funktion
/*           0=FALSE: KEIN Fehler
/*           -1: OPEN-Fehler
/*           -2: READ-Fehler
/*           -3: WRITE-Fehler
/*           -4: ALLOC-Fehler
/*           -5: unzulässige LS-Zeile
/*           -7: WSL_INIT-Fehler
/*           -9: INIT-Fehler der LZ-Verb.
/*           >0: Nummer der Zeile, in der
/*           Datei, in der ein Fehler
/*           erkannt wurde
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.    Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*           Autor      Datum      Version   Kurzzeichen
/*           -----
/*   Original :   W. Fries   01.03.2007   1.3k
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   RSP_USM.C             */
/*   Aufruf   :   LZ_RSP_USM ( cDatagramm ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine bearbeitet das FND-Datagramm "cDatagramm", indem */
/*   es dieses in die zugehörige Warteschlange für die Ausgabe an */
/*   die LZ eingetragen wird. */
/*   */
/*   aufgerufen von :   GAK_/SSA_IIZ, GAK_/SSA_ILZ */
/*   */
/*   ruft auf      :   UT_WSL_WRIT, UT_RSP2USM */
/*   */
/*   I/O-Kanäle   :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung */
/*   char     cDatagramm          RSP- bzw. USM-Datagramm */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung */
/*   int      LZ_RSP_USM          Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.      Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   POLL.C                */
/*   Aufruf   :   LZ_POLL ( )          */
/*****
/*   Beschreibung :                */
/*   Diese Routine führt die zyklische Behandlung der SEND- und RECV- */
/*   Funktionen zur LZ durch. Dabei wird die RSP- vor der USM-Warte- */
/*   schlange ausgelesen, falls der SEND-Puffer leer ist, und an die */
/*   LZ gesendet. Werden CMD-Datagramme von der LZ empfangen, so wird */
/*   dieses in die Warteschlange eingereiht und, falls nicht auf ein */
/*   RSP-Datagramm gewartet wird, ein Zeiger auf das Datagramm als */
/*   Ergebnis geliefert; SONST ist das Ergebnis ein NULL-Zeiger.    */
/*   */
/*   aufgerufen von :           GAK_/SSA_ILZ                */
/*   */
/*   ruft auf      :           LZ_SCD_POLL, LZ_BN_MSGCHK, LZ_BN_RPDP, */
/*   LZ_BN_INIT, LZ_BN_GETSOBJ, */
/*   LZ_BN_RT_FIND, LZ_BN_S2BNS, LZ_BN_L2BNL, */
/*   IP_GET_HOST, IP_GET_NET, IP_RECV, */
/*   IP_SEND, */
/*   UT_DP_FINDFIRST, UT_DT_ADD, UT_WSL_READ, */
/*   diverse Makros FND_... und SET_FND_..., */
/*   SY_F2IEEE, SY_SIG_INKR */
/*   */
/*   I/O-Kanäle   :                */
/*   */
/*****
/*   Eingabe Parameter :                */
/*   -----                */
/*   Typ      Name          Dim.      Beschreibung                */
/*   */
/*   *** KEINE ***                */
/*   -----                */
/*   Ausgabe Parameter:                */
/*   -----                */
/*   Typ      Name          Dim.      Beschreibung                */
/*   */
/*   char *  LZ_POLL                Zeiger auf das empfangenen FND- */
/*   Datum          Datagramm durch */
/*   SONST: NULL-Zeiger                */
/*   -----                */
/*   Transiente Parameter :                */
/*   -----                */
/*   Typ      Name          Dim.      Beschreibung                */
/*   */
/*   *** KEINE ***                */
/*   -----                */
/*   Autor      Datum      Version   Kurzzeichen                */
/*   -----                */
/*   Original :   W. Fries    01.03.2007   1.3k                */
/*   Änderung:                */
/*   -----                */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   STAT.C                */
/*   Aufruf   :   LZ_STAT ( )          */
/*****
/*   Beschreibung :                      */
/*   Diese Routine liefert den LZ-internen Status als Ergebnis. */
/*   */
/*   aufgerufen von :          GAK_/SSA_SYSCHK          */
/*   */
/*   ruft auf :                      */
/*   */
/*   I/O-Kanäle :                      */
/*   */
/*****
/*   Eingabe Parameter :                      */
/*   -----                      */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   *** KEINE ***                      */
/*   */
/*   -----                      */
/*   Ausgabe Parameter:                      */
/*   -----                      */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   uns.long  LZ_STAT          LZ-Status bitweise kodiert */
/*   ( vgl. LZ_INCL.H)          */
/*   -----                      */
/*   Transiente Parameter :                      */
/*   -----                      */
/*   Typ      Name          Dim.   Beschreibung          */
/*   */
/*   *** KEINE ***                      */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen          */
/*   -----                      */
/*   Original :   W. Fries      01.03.2007   1.3k          */
/*   Änderung:                      */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   ZA_TIME.C             */
/*   Aufruf   :   LZ_ZA_TIME ( )       */
/*****
/*   Beschreibung :                       */
/*   Diese Routine liefert die LZ-Systemzeit für den Zeitabgleich, */
/*   falls diese von VB_CONN bereitgestellt wurde.                 */
/*   aufgerufen von :           GAK_/SSA_SYSCHK                     */
/*   ruft auf      :           *** KEINE ***                         */
/*   I/O-Kanäle   :                       */
/*****
/*   Eingabe Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***                                           */
/*   ----- */
/*   Ausgabe Parameter:                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   uns.long LZ_ZA_TIME           LZ-Systemzeit im DOS-Format */
/*   STETS: UNDEF, da keine LZ-Zeit */
/*   ----- */
/*   Transiente Parameter :                       */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung           */
/*   ----- */
/*   *** KEINE ***                                           */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.03.2007   1.3k       */
/*   Änderung:                                     */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3k      */
/*      Modulname   :      LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*      Datei      :      EXIT.C                */
/*      Aufruf     :      LZ_EXIT ( )          */
/*****
/*      Beschreibung :                        */
/*      Diese Routine führt die spezifischen ENDE-Behandlungen für die */
/*      LZ vor dem Stoppen des Hauptprogrammes durch :                */
/*      - Schließen der Warteschlange für Datagramme von bzw. zur LZ */
/*      - Freigabe ALLOCierter Speicherbereiche                       */
/*      - Trennen der Verbindung zur LZ                               */
/*      */
/*      aufgerufen von :      GAK_/SSA_EXIT          */
/*      */
/*      ruft auf      :      IP_CLOSE, UT_WSL_EXIT, SY_FREE          */
/*      */
/*      I/O-Kanäle   :                        */
/*      */
/*****
/*      Eingabe Parameter :                        */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung          */
/*      */
/*      *** KEINE *** */
/*      */
/*      ----- */
/*      Ausgabe Parameter:                        */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung          */
/*      */
/*      int      LZ_EXIT          Returnkode der Funktion          */
/*      */
/*      0=FALSE: KEIN Fehler          */
/*      -1: OPEN-Fehler                */
/*      -2: READ-Fehler                */
/*      -3: WRITE-Fehler               */
/*      -4: ALLOC-Fehler               */
/*      -5: unzulässige IS-Zeile       */
/*      -6: DDC-IZ-Init-Fehler         */
/*      >0: Nummer der Zeile, in der  */
/*      ein Fehler erkannt wurde       */
/*      */
/*      ----- */
/*      Transiente Parameter :                    */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung          */
/*      */
/*      *** KEINE *** */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/*      Original :      W. Fries      01.03.2007      1.3k          */
/*      Änderung:                                          */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   RECV_ANZ.C           */
/*   Aufruf   :   LZ_RECV_ANZ ( )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert den LZ-internen Zähler für gesendete */
/*   Datagramme als Ergebnis.   */
/*   */
/*   aufgerufen von :   GAK_SYSCHK     */
/*   */
/*   ruft auf      :   *** KEINE ***   */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   uns.long LZ_RECV_ANZ      Anzahl gesendeter Datagramme */
/*   ( vgl. LZ_INCL.H)   */
/*   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version      Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007      1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   SEND_ANZ.C           */
/*   Aufruf   :   LZ_SEND_ANZ ( )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert den LZ-internen Zähler für gesendete */
/*   Datagramme als Ergebnis.   */
/*   */
/*   aufgerufen von :   GAK_SYSCHK     */
/*   */
/*   ruft auf      :   *** KEINE ***   */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   uns.long LZ_SEND_ANZ      Anzahl gesendeter Datagramme */
/*   ( vgl. LZ_INCL.H)   */
/*   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version      Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007      1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   CAL_INIT.C           */
/*   Aufruf   :   LZ_CAL_INIT ( )     */
/*****
/*   Beschreibung :   */
/*   Diese Routine führt die spezifischen Initialisierungen für die */
/*   Behandlung von SCHEDULE-Objekten durch :   */
/*   - Laden der CAL-spezifischen Parameter aus den CAL-Dateien   */
/*   - Initialisieren der CAL-Struktur   */
/*   */
/*   aufgerufen von :   LZ_INIT   */
/*   */
/*   ruft auf :   LZ_BN_OBJINSTANCE,   */
/*               UT_NORM_STRING,   */
/*               SY_FI_FINDFIRST, SY_FI_FINDNEXT,   */
/*               SY_CALLOC, SY_FREE   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/* - - - - -   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   int      LZ_CAL_INIT   */
/*               Returnkode der Funktion   */
/*               0=FALSE: KEIN Fehler   */
/*               -1: OPEN-Fehler   */
/*               -2: READ-Fehler   */
/*               -3: WRITE-Fehler   */
/*               -4: ALLOC-Fehler   */
/*               -5: unzulässige CAL-Zeile   */
/*               -7: WSL_INIT-Fehler   */
/*               -9: INIT-Fehler der LZ-Verb.   */
/*               >0: Nummer der Zeile, in der   */
/*                   Datei, in der ein Fehler   */
/*                   erkannt wurde   */
/*   */
/* - - - - -   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007   1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3k      */
/*      Modulname   :      LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*      Datei      :      SCD_INIT.C           */
/*      Aufruf     :      LZ_SCD_INIT ( )     */
/*****
/*      Beschreibung :                       */
/*      Diese Routine führt die spezifischen Initialisierungen für die */
/*      Behandlung von SCHEDULE-Objekten durch :                       */
/*      - Laden der SCD-spezifischen Parameter aus den SCD-Dateien     */
/*      - Initialisieren der SCD-Struktur                               */
/*      */
/*      aufgerufen von :      LZ_INIT         */
/*      */
/*      ruft auf      :      LZ_SCD_START, LZ_BN_OBJINSTANCE,          */
/*      UT_NORM_STRING, SY_CALLOC,                                       */
/*      SY_FI_FINDFIRST, SY_FI_FINDNEXT                                   */
/*      */
/*      I/O-Kanäle   :                       */
/*      */
/*****
/*      Eingabe Parameter :                       */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung          */
/*      */
/*      *** KEINE *** */
/*      ----- */
/*      Ausgabe Parameter:                       */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung          */
/*      */
/*      int      LZ_SCD_INIT      Returnkode der Funktion          */
/*      */
/*      0=FALSE: KEIN Fehler          */
/*      -1: OPEN-Fehler              */
/*      -2: READ-Fehler              */
/*      -3: WRITE-Fehler             */
/*      -4: ALLOC-Fehler             */
/*      -5: unzulässige SCD-Zeile    */
/*      -7: WSL_INIT-Fehler          */
/*      -9: INIT-Fehler der LZ-Verb. */
/*      >0: Nummer der Zeile, in der  */
/*      Datei, in der ein Fehler     */
/*      erkannt wurde                */
/*      ----- */
/*      Transiente Parameter :                       */
/*      ----- */
/*      Typ      Name      Dim.      Beschreibung          */
/*      */
/*      *** KEINE *** */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/*      Original :      W. Fries      01.03.2007      1.3k          */
/*      Änderung:                                           */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3k      */
/*      Modulname   :      LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*      Datei      :      SCD_POLL.C           */
/*      Aufruf     :      LZ_SCD_POLL ( )      */
/*****
/*      Beschreibung :      */
/*      Diese Routine führt die zyklische Bearbeitung (Polling) für die */
/*      SCHEDULE-Objekte durch :      */
/*      */
/*      aufgerufen von :      LZ_POLL          */
/*      */
/*      ruft auf      :      LZ_SCD_START, LZ_SCD_SYNC, LZ_BN_ADR, */
/*      LZ_BN_OBJTYPE, LZ_BN_OBJINSTANCE, */
/*      LZ_BN_WPDP, */
/*      UT_DT_ADD, UT_DT2CI, UT_Z_SET      */
/*      */
/*      I/O-Kanäle   :      */
/*      */
/*****
/*      Eingabe Parameter :      */
/*      -----      */
/*      Typ      Name      Dim.      Beschreibung      */
/*      */
/*      *** KEINE ***      */
/*      */
/*      -----      */
/*      Ausgabe Parameter:      */
/*      -----      */
/*      Typ      Name      Dim.      Beschreibung      */
/*      */
/*      int      LZ_SCD_POLL      Returnkode der Funktion      */
/*      0=FALSE: KEIN Fehler      */
/*      -1: OPEN-Fehler      */
/*      -2: READ-Fehler      */
/*      -3: WRITE-Fehler      */
/*      -4: ALLOC-Fehler      */
/*      -5: unzulässige SCD-Zeile      */
/*      -7: WSL_INIT-Fehler      */
/*      -9: INIT-Fehler der LZ-Verb.      */
/*      >0: Nummer der Zeile, in der      */
/*      Datei, in der ein Fehler      */
/*      erkannt wurde      */
/*      */
/*      -----      */
/*      Transiente Parameter :      */
/*      -----      */
/*      Typ      Name      Dim.      Beschreibung      */
/*      */
/*      *** KEINE ***      */
/*      */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      -----      */
/*      Original :      W. Fries      01.03.2007      1.3k      */
/*      Änderung:      */
/*      */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   SCD_START.C           */
/*   Aufruf   :   LZ_SCD_START ( uInd_SCD ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine startet den SCD-Eintrag mit dem Index "uInd_SCD" */
/*   derart, dass Zeitpunkt, Vorgabe-Wert und Priorität */
/*   des letzten Schalt-/Stell-Befehls für dieses Schedule-Objekt */
/*   im SCD-Eintrag vermerkt sind. Ist hierfür eine Calendar-Ref. */
/*   zugeordnet, so wird auch diese vermerkt. */
/*   */
/*   Hierzu wird aus den Properties "Exception_Schedule" und "Weekly-*/
/*   Schedule" der letzte Zeitpunkt in der Vergangenheit gesucht, der*/
/*   innerhalb der "Effective_Period" liegt (vgl. ASHRAE 135-2004, */
/*   unter 12.14.4 (... "on or before current time"). */
/*   */
/*   */
/*   aufgerufen von :           LZ_SCD_SYNC, UT_DT_ADD */
/*   */
/*   ruft auf           :           */
/*   */
/*   I/O-Kanäle       :           */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   ----- */
/*   unsigned uInd_SCD           Index des zu bearbeitenden */
/*   Eintrags in der SCD-Struktur */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   ----- */
/*   int      LZ_SCD_START       Returnkode der Funktion */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name           Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007   1.3k */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   SCD_SYNC.C           */
/*   Aufruf   :   LZ_SCD_SYNC ( uInd_SCD ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine synchronisiert den SCD-Eintrag mit dem Index */
/*   "uInd_SCD" derart, dass Zeitpunkt, Vorgabe-Wert und Priorität */
/*   des nächsten Schalt-/Stell-Befehls für dieses Schedule-Objekt */
/*   im SCD-Eintrag vermerkt sind. Ist hierfür eine Calendar-Ref. */
/*   zugeordnet, so wird auch diese vermerkt. */
/*   */
/*   Hierzu wird aus den Properties "Exception_Schedule" und "Weekly- */
/*   Schedule" der früheste Zeitpunkt in der Zukunft gesucht, der */
/*   innerhalb der "Effective_Period" liegt. */
/*   */
/*   aufgerufen von :   LZ_SCD_START, LZ_SCD_POLL */
/*   ruft auf       :   LZ_BN_OBJINSTANCE, UT_DT_ADD, UT_CI2DT */
/*   I/O-Kanäle     :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   unsigned uInd_SCD      1       Index des zu bearbeitenden */
/*   Eintrags in der SCD-Struktur */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   int      LZ_SCD_SYNC    1       Returnkode der Funktion */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries    01.03.2007   1.3k */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   SCD_WRITE.C           */
/*   Aufruf   :   LZ_SCD_WRITE ( uInd_SCD ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine erzeugt für SCD-Eintrag mit dem Index "uInd_SCD" */
/*   die zugehörige SCD-Datei */
/*   */
/*   aufgerufen von :   LZ_BN_WPSCD */
/*   */
/*   ruft auf      :   LZ_BN_OBJINSTANCE, UT_DT2CI */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   unsigned uInd_SCD      Index des zu bearbeitenden */
/*   Eintrags in der SCD-Struktur */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   int      LZ_SCD_WRITE   Returnkode der Funktion */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries   01.03.2007   1.3k */
/*   Änderung: */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*      Produkt      :      GA - Knoten          Version   :   1.3k      */
/*      Modulname   :      LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*      Datei      :      BN_ADR.C              */
/*      Aufruf     :      IZ04_BN_ADR ( cBN_Adr, ulpObjId, uspPropId ) */
/*****
/*      Beschreibung :                               */
/*      Diese Routine liefert den DDEV, DNET, Obj.-Id. und die Prop-Id. */
/*      für "cBN_Adr".                               */
/*                                                    */
/*      aufgerufen von :          diversen          */
/*                                                    */
/*      ruft auf      :                               */
/*                                                    */
/*      I/O-Kanäle   :                               */
/*                                                    */
/*****
/*      Eingabe Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung          */
/*      -----                               */
/*      *** KEINE ***                               */
/*      -----                               */
/*      Ausgabe Parameter:                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung          */
/*      -----                               */
/*      uns. long IZ04_BN_ADR          BN-DDEV (<=MAX_OBJINSTANCE) */
/*                                                    Sonst: UNDEF          */
/*      -----                               */
/*      Transiente Parameter :                               */
/*      -----                               */
/*      Typ      Name          Dim.      Beschreibung          */
/*      -----                               */
/*      *** KEINE ***                               */
/*      -----                               */
/*****
/*      Autor      Datum      Version      Kurzzeichen      */
/*      -----                               */
/*      Original :      W. Fries      01.03.2007      1.3k      */
/*      Änderung:                               */
/*      -----                               */
/*****
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_BNL2L.C           */
/*   Aufruf   :   LZ_BN_BNL2L ( ucBN_Wert ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine wandelt "ucBN_Wert" in das zugeh. binäre Format */
/*   */
/*   aufgerufen von :   diversen */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   uns.char * ucBN_Wert          Zeiger auf zu wandelnden Wert */
/*   im BN-Format */
/*   */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   int      LZ_BN_BNL2L          Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler */
/*   */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_BNS2S.C           */
/*   Aufruf   :   LZ_BN_BNS2S ( ucBN_Wert ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine wandelt "ucBN_Wert" in das zugeh. binäre Format */
/*   "unsigned short". */
/*   */
/*   aufgerufen von :   diversen */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   ----- */
/*   uns.char * ucBN_Wert      Zeiger auf zu wandelnden Wert */
/*   im BN-Format */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   ----- */
/*   int      LZ_BN_BNS2S      Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k   */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei     :   BN_GETSOBJ.C         */
/*   Aufruf    :   LZ_BN_GETSOBJ ( ucRECV_Buf, ulpObjId )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert SDEV und die Obj.-Id. des Datenpunktes */
/*   aus "ucRECV_Buf" als Ergebnis.   */
/*   */
/*   aufgerufen von :   diversen   */
/*   */
/*   ruft auf      :   LZ_BN_BNS2S, LZ_BN_POSTAG, LZ_BN_TAG2L, */
/*                   LZ_BN_TAGLEN, LZ_BN_OBJINSTANCE   */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   uns.char * ucRECV_Buf          Zeiger auf empf. BN-Datagramm */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   uns.long  LZ_BN_GETSOBJ          Instance SDEV des BN-Datagrammes */
/*                   SONST: UNDEF   */
/*   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007   1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_INIT.C             */
/*   Aufruf   :   LZ_BN_INIT ( iFlag, ucSEND_Buf, RPort )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine führt die Initialisierung der BN-IZ durch.   */
/*   Abhängig von "iFlag" wird die erste bzw. nächste zu sendende   */
/*   Initialisierungs-Sequenz in "ucSEND_Buf" erzeugt.   */
/*   */
/*   aufgerufen von :   LZ_POLL   */
/*   */
/*   ruft auf :   LZ_BN_RT_INIT, LZ_BN_S2BNS, LZ_BN_L2BNL, */
/*               LZ_BN_NXINVOKE, IP_GET_HOST, IP_GET_NET */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   int      iFlag      */      Flag für Initialisierungs-Modus */
/*   */      */      */      INIT: Erste Init.-Sequenz   */
/*   */      */      */      BUSY: Nächste Init.-Sequenz   */
/*   */      */      */      */
/*   uns.char * ucSEND_Buf      */      Puffer zur Übergabe der Init.- */
/*   */      */      */      Sequenz   */
/*   */      */      */      */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   int      LZ_BN_INIT      */      Returnkode der Funktion   */
/*   */      */      */      >0: Länge des BN-Datagrammes   */
/*   */      */      */      in "ucSEND_Buf" (Conf.)   */
/*   */      */      */      0: Ende der Initialisierung   */
/*   */      */      */      <0: Länge, jedoch Unconf.   */
/*   */      */      */      */
/*   IP_PORT*  RPort      */      IP-Adresse+Port des Empfängers */
/*   */      */      */      */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name      Dim.      Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version      Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007      1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_L2BNL.C           */
/*   Aufruf   :   LZ_BN_L2BNL ( ulWert, ucBN_Wert ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine wandelt "ulWert" in das zugehörige BN-Format */
/*   */
/*   aufgerufen von :   diversen */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   uns.long  ulWert      zu wandelnder Wert */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   int      LZ_BN_L2BNL      Returnkode der Funktion */
/*   0=FALSE: KEIN Fehler */
/*   uns.char* ucBN_Wert      Zeiger auf Puffer für den Wert */
/*   im BN-Format */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_MSGCHK.C           */
/*   Aufruf   :   LZ_BN_MSGCHK ( PORT, ucRECV_Buf ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft für "PORT" das BN-Datagramm "ucRECV_Buf" */
/*   und liefert den zugehörigen FND-D_R als Ergebnis, falls ein */
/*   RSP oder USM vorliegt. */
/*   Im Fehlerfall wird der BN-Error-Code bzw. Reject-Reason - */
/*   kodiert im Zahlenbereich <=1000 - als Ergebnis geliefert. */
/*   */
/*   Folgende BN-Datagramme werden intern wie folgt behandelt : */
/*   - "Network Layer/Control Message" werden beantwortet, soweit */
/*   gemäß ENV-1805/1 erforderlich. */
/*   - "ReadProperty" auf das DEVICE-Object wird beantwortet, soweit */
/*   gemäß ENV-1805/1 erforderlich. */
/*   - "I-Am" aktualisiert den zugehörigen Routing-Eintrag */
/*   - "Who-Has" wird mit "I-Have" beantwortet */
/*   - "Who-Is" wird mit "I-Am" beantwortet */
/*   - Alle anderen APDU's werden mit Error "Service-Request-Denied" */
/*   bzw. "Unsupported-Object-Type" abgewiesen. */
/*   */
/*   aufgerufen von :   LZ_POLL */
/*   */
/*   ruft auf   :   LZ_BN_MSGNW, LZ_BN_MSGRPR, LZ_BN_MSGWPR */
/*   LZ_BN_RT_FIND, LZ_BN_RT_SET, */
/*   LZ_BN_S2BNS, LZ_BN_BNS2S, LZ_BN_L2BNL, */
/*   LZ_BN_POSNPDU, LZ_BN_POSTAG, */
/*   LZ_BN_TAGLEN, LZ_BN_TAG2L, */
/*   LZ_BN_OBJINSTANCE, */
/*   LZ_BN_SEGRD, LZ_BN_SEGRM, */
/*   IP_CHK_HOST, IP_GET_NET, IP_SEND, */
/*   UT_KK_GETKK, UT_CS2CS */
/*   */
/*   I/O-Kanäle   :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   IP_PORT *   PORT      IP-Adr./-Port von der das BN- */
/*   Datagramm empfangen wurde */
/*   uns.char * ucRECV_Buf      Puffer mit BN-Datagramm */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   int      LZ_BN_MSGCHK      Returncode der Funktion */
/*   >=0: FND-D_R */
/*   <=-1000: BN-Error-Code */
/*   <=-2000: BN-Reject-Reason */
/*   Sonst: UNDEF */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_MSGNW.C           */
/*   Aufruf   :   LZ_BN_MSGNW ( PORT, ucRECV_Buf )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine prüft für "PORT" das BN-Datagramm "ucRECV_Buf" */
/*   auf "Network Control/Layer Message" und beantwortet diese, */
/*   soweit gemäß ENV 1805/1 erforderlich.           */
/*           */
/*   aufgerufen von :           LZ_POLL           */
/*           */
/*   ruft auf :           LZ_BN_BNS2S, LZ_BN_S2BNS, LZ_BN_POSNPDU */
/*           IP_SEND           */
/*           */
/*   I/O-Kanäle :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   IP_PORT *  PORT           IP-Adr./-Port von der das BN- */
/*   Datagramm empfangen wurde           */
/*   -----           */
/*   uns.char * ucRECV_Buf           Puffer mit BN-Datagramm           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   int      LZ_BN_MSGNW           Returnkode der Funktion           */
/*   0=FALSE: KEIN Fehler           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*****
/*   Autor      Datum      Version      Kurzzeichen           */
/*   -----           */
/*   Original :   W. Fries      01.03.2007      1.3k           */
/*   Änderung:           */
/*   -----           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_MSGRPR.C           */
/*   Aufruf   :   LZ_BN_MSGRPR ( PORT, ucRECV_Buf ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine prüft für "PORT" das BN-Datagramm "ucRECV_Buf" */
/*   auf Service "ReadProperty" für das DEVICE-Object und beantwortet */
/*   dieses, soweit gemäß ENV 1805/1 erforderlich. Alle anderen */
/*   APDU's werden mit Error "Service-Request-Denied" bzw. */
/*   "Unsupported-Object-Type" abgewiesen. */
/*   */
/*   */
/*   aufgerufen von :   LZ_BN_MSGCHK */
/*   */
/*   ruft auf :   LZ_BN_RPDEV, LZ_BN_RPDP, LZ_BN_RPSCD, */
/*               LZ_BN_RPCAL, LZ_BN_OBJTYPE, */
/*               LZ_BN_POSNPDU, LZ_BN_POSTAG, */
/*               LZ_BN_POSAPDU, LZ_BN_SEGRD, LZ_BN_SEGRM, */
/*               LZ_BN_SEGWR, LZ_BN_TAGLEN, LZ_BN_TAG2L, */
/*               LZ_BN_BNS2S, LZ_BN_S2BNS, LZ_BN_L2BNL, */
/*               IP_SEND */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   */
/*   IP_PORT *   PORT           IP-Adr./-Port von der das BN- */
/*   */           Datagramm empfangen wurde */
/*   */
/*   uns.char * ucRECV_Buf           Puffer mit BN-Datagramm */
/*   */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   */
/*   int      LZ_BN_MSGRPR           Returnkode der Funktion */
/*   */           0=FALSE: KEIN Fehler */
/*   */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007   1.3k */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_MSGWPR.C           */
/*   Aufruf   :   LZ_BN_MSGWPR ( PORT, ucRECV_Buf )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine prüft für "PORT" das BN-Datagramm "ucRECV_Buf" auf */
/*   Service "WriteProperty" für das DEVICE-Object und beantwortet */
/*   dieses, soweit gemäß ENV 1805/1 erforderlich. Alle anderen */
/*   APDU's werden mit Error "Service-Request-Denied" bzw. */
/*   "Unsupported-Object-Type" abgewiesen.           */
/*           */
/*   aufgerufen von :           LZ_BN_MSGCHK           */
/*           */
/*   ruft auf      :           LZ_BN_WPDP, LZ_BN_WPSCD, LZ_BN_WPCAL, */
/*                               LZ_BN_OBJTYPE, LZ_BN_POSNPDU, */
/*                               LZ_BN_POSTAG, LZ_BN_POSAPDU, */
/*                               LZ_BN_SEGRD, LZ_BN_SEGRM, LZ_BN_SEGWR, */
/*                               LZ_BN_TAGLEN, LZ_BN_TAG2L, */
/*                               LZ_BN_BNS2S, LZ_BN_S2BNS, LZ_BN_L2BNL, */
/*                               IP_SEND, SY_IEEE2F           */
/*           */
/*   I/O-Kanäle   :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   IP_PORT *   PORT           IP-Adr./-Port von der das BN- */
/*   Datagramm empfangen wurde           */
/*   uns.char * ucRECV_Buf           Puffer mit BN-Datagramm           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   int      LZ_BN_MSGWPR           Returnkode der Funktion           */
/*   0=FALSE: KEIN Fehler           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*   -----           */
/*   *** KEINE ***           */
/*****
/*           Autor      Datum      Version      Kurzzeichen           */
/*   -----           */
/*   Original :   W. Fries      01.03.2007      1.3k           */
/*   Änderung:           */
/*           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_NXINVOKE.C         */
/*   Aufruf   :   LZ_BN_NXINVOKE ( )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert die nächste BN-InvokeId.   */
/*   */
/*   aufgerufen von :   diversen   */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*   - - - - -   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   uns.char  LZ_BN_NXINVOKE      Nächste BN-Invoke-Id.   */
/*   */
/*   - - - - -   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007    1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_OBJINSTANCE.C      */
/*   Aufruf   :   LZ_BN_OBJINSTANCE ( uObjId ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert die Object-Instance für "uObjId". */
/*   */
/*   aufgerufen von :   diversen        */
/*   */
/*   ruft auf      :   */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung      */
/*   */
/*   *** KEINE *** */
/*   */
/* - - - - - */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung      */
/*   */
/*   uns. long LZ_BN_OBJINSTANCE  Object-Instance */
/*   */
/*   Sonst: UNDEF */
/*   */
/* - - - - - */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung      */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries    01.03.2007    1.3k      */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_OBJTYPE.C           */
/*   Aufruf   :   LZ_BN_OBJTYPE ( ulObjId ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine liefert den Object-Type für "ulObjId". */
/*   */
/*   aufgerufen von :           diversen           */
/*   */
/*   ruft auf :                               */
/*   */
/*   I/O-Kanäle :                               */
/*   */
/*****
/*   Eingabe Parameter :                               */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   */
/*   ----- */
/*   Ausgabe Parameter:                               */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung           */
/*   */
/*   uns. long LZ_BN_OBJTYPE           Object-Type           */
/*   */
/*   Sonst: UNDEF */
/*   */
/*   ----- */
/*   Transiente Parameter :                               */
/*   ----- */
/*   Typ      Name          Dim.    Beschreibung           */
/*   */
/*   *** KEINE *** */
/*   */
/*****
/*   Autor      Datum      Version  Kurzzeichen           */
/*   ----- */
/*   Original :   W. Fries      01.03.2007    1.3k           */
/*   Änderung:                               */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_POSAPDU.C           */
/*   Aufruf   :   LZ_BN_POSAPDU ( ucRECV_Buf ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert den Index im BN-Datagramm "ucRECV_Buf" */
/*   ab dem der APDU beginnt. */
/*   */
/*   aufgerufen von :   diversen */
/*   */
/*   ruft auf      :   LZ_BN_POSNPDU, LZ_BN_BNS2S */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   uns.char * ucRECV_Buf      Puffer mit BN-Datagramm */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   int      LZ_BN_POSAPDU      Index, ab dem APDU beginnt */
/*   Sonst: UNDEF */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*   ----- */
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung:   */
/*   ----- */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_POSNPDU.C           */
/*   Aufruf   :   LZ_BN_POSNPDU ( ucRECV_Buf ) */
/*****
/*   Beschreibung :                               */
/*   Diese Routine liefert den Index im BN-Datagramm "ucRECV_Buf" */
/*   ab dem der NPDU beginnt.                               */
/*   aufgerufen von :           diversen           */
/*   ruft auf      :           LZ_BN_BNS2S         */
/*   I/O-Kanäle   :                               */
/*****
/*   Eingabe Parameter :                               */
/*   -----                               */
/*   Typ      Name           Dim.   Beschreibung           */
/*   -----                               */
/*   uns.char * ucRECV_Buf           Puffer mit BN-Datagramm */
/*   -----                               */
/*   Ausgabe Parameter:                               */
/*   -----                               */
/*   Typ      Name           Dim.   Beschreibung           */
/*   -----                               */
/*   int      LZ_BN_POSNPDU           Index, ab dem NPDU beginnt */
/*   Sonst: UNDEF                               */
/*   -----                               */
/*   Transiente Parameter :                               */
/*   -----                               */
/*   Typ      Name           Dim.   Beschreibung           */
/*   -----                               */
/*   *** KEINE ***                               */
/*****
/*   Autor      Datum      Version   Kurzzeichen           */
/*   -----                               */
/*   Original :   W. Fries      01.03.2007   1.3k           */
/*   Änderung:                               */
/*   -----                               */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_POSTAG.C           */
/*   Aufruf   :   LZ_BN_POSTAG ( ucRECV_Buf ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert den Index im BN-Datagramm "ucRECV_Buf" */
/*   ab dem der erste TAG im APDU beginnt.   */
/*   */
/*   aufgerufen von :   diversen   */
/*   */
/*   ruft auf      :   LZ_BN_POSAPDU, LZ_BN_BNS2S   */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   uns.char * ucRECV_Buf          Puffer mit BN-Datagramm   */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   int      LZ_BN_POSTAG          Index, ab dem TAG beginnt   */
/*   Sonst: UNDEF   */
/*   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007   1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_RPCAL.C           */
/*   Aufruf   :   LZ_BN_RPCAL ( ulObjId, ulProp, ulIndex, iInvokeId, */
/*                                     iBufLen, ucSEND_Buf)*/
/*****
/*   Beschreibung   :   */
/*   Diese Routine fügt die Antwort für das Lesen der Property */
/*   "ulProp", ggf. mit Array-Index "ulIndex" des Objektes mit der */
/*   Id. "ulObjId" am Puffer "ucSEND_Buf" an. */
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei */
/*   übergeben. */
/*   Unterstützt werden die folgenden Objekt-Typen: */
/*   - Schedule */
/*   aufgerufen von :   LZ_BN_MSGRPR */
/*   ruft auf       :   LZ_BN_OBJINSTANCE, LZ_BN_L2BNL, */
/*                       UT_CI2DT, UT_CS2CS */
/*   I/O-Kanäle     :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   uns.long ulObjId    Id. des zu lesenden Objektes */
/*   uns.long ulProp     zu lesende Property */
/*   uns.long ulIndex    zu lesender Array-Index */
/*                       Sonst: UNDEF */
/*   int      iInvokeId  >=0: InvokeId */
/*   int      iBufLen    >0: Länge des (Teil)-Datagramms */
/*                       in "ucSEND_Buf" für TRANSMIT */
/*   uns.char * ucSEND_Buf Puffer mit Teil des BN-Datagramm */
/*                       Für BN_SEG_OPEN muss dieser den */
/*                       BLVC, NPDU und APDU-Header */
/*                       enthalten, jedoch OHNE Flag für */
/*                       SEG, MORE, SA, Sequence-Number */
/*                       und ProposedWindowSize. */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   int      LZ_BN_RPCAL Returnkode der Funktion */
/*                       >0: Anzahl zu sendender Bytes */
/*                       für segm. Datagramm */
/*                       0: KEIN BN-Datagramm zu senden */
/*                       <0: Anzahl zu sendender Bytes */
/*                       für ERROR-Datagramm (abs) */
/*   interne Kodes für "ier", die */
/*   Returnkode 0 liefern : */
/*   -1: Unzuläss. Objekt-Id. */
/*   -2: Unzuläss. Property */
/*   -3: Unzuläss. Array-Index */
/*   -4: Unzuläss. InvokeId */
/*   -5: Unzuläss. Länge */
/*   -6: Unzuläss. SEND-Buf. */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung: */
/*****

```

```
/*  
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */  
*/
```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_RPDEV.C           */
/*   Aufruf   :   LZ_BN_RPDEV ( ulObjId, ulProp, ulIndex, iInvokeId, */
/*                               iBufLen, ucSEND_Buf)*/
/*****
/*   Beschreibung   :   */
/*   Diese Routine fügt die Antwort für das Lesen der Property */
/*   "ulProp", ggf. mit Array-Index "ulIndex" des Objektes mit der */
/*   Id. "ulObjId" am Puffer "ucSEND_Buf" an. */
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei */
/*   übergeben. */
/*   */
/*   Unterstützt werden die folgenden Objekt-Typen: */
/*   - Device */
/*   */
/*   aufgerufen von :   LZ_BN_MSGRPR */
/*   */
/*   ruft auf      :   LZ_BN_SEGWR, LZ_BN_S2BNS, LZ_BN_L2BNL, */
/*                   UT_KK_GETKK, UT_CS2CS */
/*   */
/*   I/O-Kanäle    :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   uns.long ulObjId      Id. des zu lesenden Objektes */
/*   */
/*   uns.long ulProp      zu lesende Property */
/*   */
/*   uns.long ulIndex     zu lesender Array-Index */
/*   */
/*   int      iInvokeId   >=0: InvokeId */
/*   */
/*   int      iBufLen     >0: Länge des (Teil)-Datagramms */
/*   */
/*   uns.char * ucSEND_Buf Puffer mit Teil des BN-Datagramm */
/*   */
/*   Für BN_SEG_OPEN muss dieser den */
/*   BLVC, NPDU und APDU-Header */
/*   enthalten, jedoch OHNE Flag für */
/*   SEG, MORE, SA, Sequence-Number */
/*   und ProposedWindowSize. */
/*   */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   int      LZ_BN_RPDEV Returnkode der Funktion */
/*   */
/*   >0: Anzahl zu sender Bytes */
/*   für segm. Datagramm */
/*   0: KEIN BN-Datagramm zu senden */
/*   <0: Anzahl zu sender Bytes */
/*   für ERROR-Datagramm (abs) */
/*   */
/*   interne Codes für "ier", die */
/*   Returnkode 0 liefern : */
/*   -1: Unzuläss. Objekt-Id. */
/*   -2: Unzuläss. Property */
/*   -3: Unzuläss. Array-Index */
/*   -4: Unzuläss. InvokeId */
/*   -5: Unzuläss. Länge */
/*   -6: Unzuläss. SEND-Buf. */
/*   */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version      Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k */
/*   Änderung: */
/*   */

```



```
/*  
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */  
/*
```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_RPDP.C             */
/*   Aufruf   :   LZ_BN_RPDP ( ulObjId, ulProp, ulIndex, iInvokeId,
/*                                     iBufLen, ucSEND_Buf)*/
/*****
/*   Beschreibung :
/*   Diese Routine fügt die Antwort für das Lesen der Property
/*   "ulProp", ggf. mit Array-Index "ulIndex" des Objektes mit der
/*   Id. "ulObjId" am Puffer "ucSEND_Buf" an.
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei*
/*   übergeben.
/*
/*   Unterstützt werden die folgenden Objekt-Typen:
/*   - Analog Input      (AI)
/*   - Analog Output     (AO)
/*   - Analog Value      (AV)
/*   - Binary Input      (BI)
/*   - Binary Output     (BO)
/*   - Binary Value      (BV)
/*   - MultiState Input  (MI)
/*   - MultiState Output (MO)
/*   - MultiState Value  (MV)
/*
/*   aufgerufen von :      LZ_BN_MSGRP
/*
/*   ruft auf :           LZ_BN_OBJTYPE, LZ_BN_L2BNL,
/*                       UT_DP_ATBMIND, UT_DP_ATBTEXT, UT_CS2CS,
/*                       UT_DP_DPTEXT, UT_FND_TIME, UT_WSL_WRIT,
/*                       SY_F2IEEEE, div. Macros SET_FND_...
/*
/*   I/O-Kanäle :
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*   uns.long ulObjId    Id. des zu lesenden Objektes
/*   uns.long ulProp     zu lesende Property
/*   uns.long ulIndex    zu lesender Array-Index
/*                       Sonst: UNDEF
/*   int      iInvokeId  >=0: InvokeId
/*   int      iBufLen    >0: Länge des (Teil)-Datagramms
/*                       in "ucSEND_Buf" für TRANSMIT*
/*   uns.char * ucSEND_Buf Puffer mit Teil des BN-Datagramm*
/*                       Für BN_SEG_OPEN muss dieser den
/*                       BLVC, NPDU und APDU-Header
/*                       enthalten, jedoch OHNE Flag für
/*                       SEG, MORE, SA, Sequence-Number
/*                       und ProposedWindowSize.
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*   int      LZ_BN_RPDP Returnkode der Funktion
/*                       >0: Anzahl zu sender Bytes
/*                           für segm. Datagramm
/*                       0: KEIN BN-Datagramm zu senden*
/*                       <0: Anzahl zu sender Bytes
/*                           für ERROR-Datagramm (abs)
/*
/*                       interne Kodes für "ier", die
/*                       Returnkode 0 liefern :
/*                       -1: Unzuläss. Objekt-Id.
/*                       -2: Unzuläss. Property
/*                       -3: Unzuläss. Array-Index
/*                       -4: Unzuläss. InvokeId
/*                       -5: Unzuläss. Länge
/*                       -6: Unzuläss. SEND-Buf.
/*   -----
/*   Transiente Parameter :
/*   -----

```

```
/*      Typ      Name          Dim.    Beschreibung          */
/*      *** KEINE ***          */
/*      *****/
/*      Autor      Datum      Version    Kurzzeichen    */
/*      -----
/* Original :    W. Fries      01.03.2007    1.3k           */
/* Änderung:
/*
/*      *****/
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*      *****/
```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_RPSCD.C           */
/*   Aufruf   :   LZ_BN_RPSCD ( ulObjId, ulProp, ulIndex, iInvokeId, */
/*                               iBufLen, ucSEND_Buf)*/
/*****
/*   Beschreibung   :   */
/*   Diese Routine fügt die Antwort für das Lesen der Property */
/*   "ulProp", ggf. mit Array-Index "ulIndex" des Objektes mit der */
/*   Id. "ulObjId" am Puffer "ucSEND_Buf" an. */
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei */
/*   übergeben. */
/*   Unterstützt werden die folgenden Objekt-Typen: */
/*   - Schedule */
/*   aufgerufen von :   LZ_BN_MSGRPR */
/*   ruft auf      :   LZ_BN_ADR, LZ_BN_OBJINSTANCE, */
/*                   LZ_BN_SEGWR, LZ_BN_L2BNL, */
/*                   UT_CI2DT, UT_CS2CS, SY_F2IEEE */
/*   I/O-Kanäle    :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   ----- */
/*   uns.long ulObjId      Id. des zu lesenden Objektes */
/*   uns.long ulProp      zu lesende Property */
/*   uns.long ulIndex     zu lesender Array-Index */
/*                               Sonst: UNDEF */
/*   int      iInvokeId   >=0: InvokeId */
/*   int      iBufLen     >0: Länge des (Teil)-Datagramms */
/*                               in "ucSEND_Buf" für TRANSMIT*/
/*   uns.char * ucSEND_Buf Puffer mit Teil des BN-Datagramm*/
/*                               Für BN_SEG_OPEN muss dieser den */
/*                               BLVC, NPDU und APDU-Header */
/*                               enthalten, jedoch OHNE Flag für */
/*                               SEG, MORE, SA, Sequence-Number */
/*                               und ProposedWindowSize. */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   ----- */
/*   int      LZ_BN_RPSCD   Returnkode der Funktion */
/*                               >0: Anzahl zu sendender Bytes */
/*                               für segm. Datagramm */
/*                               0: KEIN BN-Datagramm zu senden*/
/*                               <0: Anzahl zu sendender Bytes */
/*                               für ERROR-Datagramm (abs) */
/*   interne Kodes für "ier", die */
/*   Returnkode 0 liefern : */
/*   -1: Unzuläss. Objekt-Id. */
/*   -2: Unzuläss. Property */
/*   -3: Unzuläss. Array-Index */
/*   -4: Unzuläss. InvokeId */
/*   -5: Unzuläss. Länge */
/*   -6: Unzuläss. SEND-Buf. */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung          */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen */
/*   ----- */
/*   Original :   W. Fries   01.03.2007  1.3k */
/*   Änderung:

```

```
/* */  
/*****/  
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */  
/*****/
```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_RT_FIND.C           */
/*   Aufruf   :   LZ_BN_RT_FIND ( ulSDEV, usSNET, ucSLEN, ucSADR,
/*                                     PORT ) */
/*****
/*   Beschreibung :
/*   Diese Routine besetzt den Eintrag in der Routing-Tabelle für
/*   "ulSDEV" mit den Werten aus "usSNET" ff. und protokolliert ggf.
/*   unerwartete Veränderungen.
/*
/*   aufgerufen von :   diversen
/*
/*   ruft auf :
/*
/*   I/O-Kanäle :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   uns.long  ulSDEV      SDEV (Device-Instance)
/*                                     oder UNDEF
/*   uns.short usSNET      SNET (Network-Id.)
/*                                     oder UNDEF
/*   uns.char  ucSLEN      SLEN (Länge der SADR)
/*                                     oder UNDEF
/*   uns.char * ucSADR      SADR (MAC der Länge SLEN)
/*   IP_PORT *  PORT       IP-Adr. und -Port
/*                                     oder NULL
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   LZ_BN_ROUTE* LZ_BN_RT_FIND  Zeiger auf Eintrag in der
/*                                     Routing-Tabelle
/*                                     Sonst: NULL
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version      Kurzzeichen
/*   -----
/*   Original :   W. Fries      01.03.2007      1.3k
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei     :   BN_RT_INIT.C         */
/*   Aufruf    :   LZ_BN_RT_INIT ( )    */
/*****
/*   Beschreibung :
/*   Diese Routine liest die Device-Datei und initialisiert die
/*   Routing-Tabelle.
/*
/*   aufgerufen von :           LZ_INIT, LZ_POLL
/*
/*   ruft auf      :           LZ_BN_RT_FIND, LZ_BN_RT_SET,
/*                           IP_A2PORT, UT_NORM_STRING
/*
/*   I/O-Kanäle   :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/* -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   int      LZ_BN_RT_INIT          Returnkode der Funktion
/*                           0=FALSE: KEIN Fehler
/*
/* -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name          Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version   Kurzzeichen
/*   -----
/*   Original :   W. Fries      01.03.2007   1.3k
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_RT_SET.C           */
/*   Aufruf   :   LZ_BN_RT_SET ( ulSDEV, usSNET, ucSLEN, ucSADR,
/*                                     PORT ) */
/*****
/*   Beschreibung :
/*   Diese Routine besetzt den Eintrag in der Routing-Tabelle für
/*   "ulSDEV" mit den Werten aus "usSNET" ff. und protokolliert ggf.
/*   unerwartete Veränderungen.
/*
/*   aufgerufen von :   diversen
/*
/*   ruft auf :
/*
/*   I/O-Kanäle :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   uns.long  ulSDEV    SDEV (Device-Instance)
/*   uns.short usSNET    SNET (Network-Id.)
/*   uns.char  ucSLEN    SLEN (Länge der SADR)
/*   uns.char * ucSADR    SADR (MAC der Länge SLEN)
/*   IP_PORT * PORT      IP-Adr. und -Port
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   int      LZ_BN_RT_SET
/*                                     Index des Eintrags für SDEV in
/*                                     der Routing-Tabelle
/*                                     Sonst: UNDEF
/*
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version      Kurzzeichen
/*   -----
/*   Original :   W. Fries      01.03.2007      1.3k
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```



```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_S2BNS.C           */
/*   Aufruf   :   LZ_BN_S2BNS ( usWert, ucBN_Wert )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine wandelt "usWert" in das zugehörige BN-Format */
/*           */
/*   aufgerufen von :           diversen           */
/*           */
/*   ruft auf :           */
/*           */
/*   I/O-Kanäle :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           */
/*   uns.short usWert           zu wandelnder Wert           */
/*           */
/*   - - - - -           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           */
/*   int      LZ_BN_S2BNS           Returnkode der Funktion */
/*           0=FALSE: KEIN Fehler           */
/*           */
/*   uns.char* ucBN_Wert           Zeiger auf Puffer für den Wert */
/*           im BN-Format           */
/*           */
/*   - - - - -           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           */
/*   *** KEINE ***           */
/*           */
/*****
/*           Autor      Datum      Version      Kurzzeichen   */
/*           -----           */
/*   Original :   W. Fries      01.03.2007      1.3k           */
/*   Änderung:           */
/*           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_SEGRD.C           */
/*   Aufruf   :   LZ_BN_SEGRD ( iInvokeId, iLen, ucSEND_Buf ) */
/*****
/*   Beschreibung :           */
/*   Diese Routine liest max. "iLen" Zeichen aus der aktuellen SEG- */
/*   Datei in den "ucSEND_Buf" abhängig von "iInvokeId" aus.      */
/*   */
/*   aufgerufen von :           diversen */
/*   */
/*   ruft auf      :           LZ_BN_S2BNS */
/*   */
/*   I/O-Kanäle   :           */
/*   */
/*****
/*   Eingabe Parameter :           */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung           */
/*   int      iInvokeId           >=0: InvokeId           */
/*   int      iLen              >0: Länge des (Teil)-Datagramms */
/*   in "ucSEnDBuf" für TRANSMIT */
/*   <0: dito für RECEIVE */
/*   uns.char * ucSEND_Buf      Puffer mit ggf. segmentiertem */
/*   BN-Datagramm */
/*   ----- */
/*   Ausgabe Parameter:           */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung           */
/*   int      LZ_BN_SEGRD      Returnkode der Funktion */
/*   >0: Anzahl zu sendender Bytes */
/*   für segm. Datagramm */
/*   0: KEIN BN-Datagramm zu senden*/
/*   <0: Anzahl zu sendender Bytes */
/*   für NICHT segm. Datagramm */
/*   interne Codes für "ier", die */
/*   Returnkode 0 liefern :      */
/*   -1: Unzuläss. InvokeId      */
/*   -2: Unzuläss. Länge         */
/*   -3: Unzuläss. SEND-Buf.     */
/*   -1n: OPEN-Fehler            */
/*   -2n: READ-Fehler            */
/*   -3n: WRITE-Fehler           */
/*   -4n: APDU-Fehler            */
/*   ----- */
/*   Transiente Parameter :       */
/*   ----- */
/*   Typ      Name      Dim.      Beschreibung           */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version  Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries   01.03.2007  1.3k      */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_SEGRM.C           */
/*   Aufruf   :   LZ_BN_SEGRM ( iInvokeId, iLen )           */
/*****
/*   Beschreibung :           */
/*   Diese Routine löscht die SEG-Datei abhängig von "iInvokeId". */
/*           */
/*   aufgerufen von :           diversen           */
/*           */
/*   ruft auf :           */
/*           */
/*   I/O-Kanäle :           */
/*           */
/*****
/*   Eingabe Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           -----           */
/*   int      iInvokeId           >=0: InvokeId           */
/*           -----           */
/*   int      iLen                >0: Länge des (Teil)-Dategramms */
/*                                     in "ucSEnDBuf" für TRANSMIT */
/*                                     <0: dito für RECEIVE           */
/*           -----           */
/*   -----           */
/*   Ausgabe Parameter:           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           -----           */
/*   int      LZ_BN_SEGRM           Returnkode der Funktion           */
/*                                     0=FALSE: KEIN Fehler           */
/*                                     -1: Unzuläss. InvokeId           */
/*                                     -2: Unzuläss. Länge           */
/*                                     -3: Unzuläss. SEND-Buf.           */
/*                                     -10: OPEN-Fehler           */
/*                                     -20: READ-Fehler           */
/*                                     -30: WRITE-Fehler           */
/*                                     -40: APDU-Fehler           */
/*           -----           */
/*   -----           */
/*   Transiente Parameter :           */
/*   -----           */
/*   Typ      Name      Dim.      Beschreibung           */
/*           -----           */
/*   *** KEINE ***           */
/*           -----           */
/*****
/*           Autor      Datum      Version      Kurzzeichen           */
/*           -----           */
/* Original :   W. Fries      01.03.2007      1.3k           */
/* Änderung:           */
/*           -----           */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_SEGWR.C           */
/*   Aufruf   :   LZ_BN_SEGWR ( iInvokeId, ucMaxSegmAPDU, iLen,
/*                               ucSEND_Buf )
/*****
/*   Beschreibung :
/*   Diese Routine fügt "iLen" Zeichen aus "ucSEND_Buf" an das seg-
/*   mentiert auszubende BN-Datagramm abhängig von "iInvokeId" an.
/*
/*   Hinweis: Das (Teil)-Datagramm in "ucSEND_Buf" muss stets so
/*   ===== aufgebaut sein, dass in dessen Anschluss ein Segment
/*   abgeschlossen werden kann (vgl. "Octet Boundery" in
/*   ASHRAE 135-2004 unter 5.2.1.1)
/*
/*   aufgerufen von :           diversen
/*
/*   ruft auf       :
/*
/*   I/O-Kanäle    :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   int      iInvokeId          >=0: InvokeId
/*
/*   uns.char ucMaxSegmAPDU      Max. Anzahl Segmente und max.
/*                               Länge des APDU kodiert gemäß
/*                               ASHRAE 135-2004 Abschnitt 20.1.2*
/*                               Sonst: UNDEF
/*
/*   int      iLen              >0: Länge des (Teil)-Dategramms
/*                               in "ucSEND_Buf" für TRANSMIT
/*                               <0: dito für RECEIVE
/*
/*   uns.char * ucSEND_Buf      Puffer mit Teil des BN-Datagramm*
/*                               Für BN_SEG_OPEN muss dieser den
/*                               BLVC, NPDU und APDU-Header
/*                               enthalten, jedoch OHNE Flag für
/*                               SEG, MORE, SA, Sequence-Number
/*                               und ProposedWindowSize.
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   int      LZ_BN_SEGWR      Returnkode der Funktion
/*                               0=FALSE: KEIN Fehler
/*                               -1: Unzuläss. InvokeId
/*                               -2: Unzuläss. Länge
/*                               -3: Unzuläss. SEND-Buf.
/*                               -10: OPEN-Fehler
/*                               -20: READ-Fehler
/*                               -30: WRITE-Fehler
/*                               -40: APDU-Fehler
/*
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   *** KEINE ***
/*
/*****
/*   Autor      Datum      Version      Kurzzeichen
/*   -----
/*   Original :   W. Fries      01.03.2007      1.3k
/*   Änderung:
/*
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei     :   BN_TAG2L.C           */
/*   Aufruf    :   LZ_BN_TAG2L ( ucBN_Tag, iTagContext )   */
/*****
/*   Beschreibung :   */
/*   Diese Routine wandelt "ucBN_Tag" in das zugeh. binäre Format */
/*   von 4 Byte wie folgt :   */
/*   - NULL    :   stets 0   */
/*   - BOOL    :   0 oder 1   */
/*   - UNS.    :   rechtsbündig mit führenden 0 für Cast   */
/*   - SIG.    :   rechtsbündig mit führenden 0 bzw. 1 für Cast (2-Compl) */
/*   - FLOAT   :   internes FLOAT-Format für Cast   */
/*   - ENUM    :   rechtsbündig mit führenden 0 für Cast   */
/*   - DATE    :   DOS-Format für Datum (16-Bit !!)   */
/*   - TIME    :   DOS-Format für Zeit (16-Bit !!)   */
/*   - OBJID   :   rechtsbündig mit führenden 0 für Cast   */
/*   - Sonst   :   ** wird derzeit nicht unterstützt; liefert UNDEF ***   */
/*   Hinweis: Die Casts sind mittels Zeiger (z.B. float *) durchzu- */
/*   ===== damit die binäre Codierung erhalten bleibt !!   */
/*   */
/*   aufgerufen von :   diversen   */
/*   ruft auf      :   SY_IEEE2F   */
/*   I/O-Kanäle    :   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   uns.char * ucBN_Tag          Zeiger auf zu wandelnden Tag   */
/*   im BN-Format (vgl. 20.2.1.4 ff.) */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   uns.long  LZ_BN_TAG2L      binärer Wert des Tag   */
/*   SONST: UNDEF (kann auch gültiger */
/*   binärer Wert sein !!)   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.   Beschreibung   */
/*   *** KEINE ***   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----   -----   -----   -----   */
/*   Original :   W. Fries      01.03.2007   1.3k   */
/*   Änderung:   */
/*   -----   -----   -----   -----   */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_TAGLEN.C           */
/*   Aufruf   :   LZ_BN_TAGLEN ( ucBN_Tag ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert die Länge für "ucBN_Tag" in Bytes (vgl. */
/*   20.2.1.3) */
/*   */
/*   aufgerufen von :   diversen */
/*   */
/*   ruft auf      :   LZ_BN_BNS2S, LZ_BN_BNL2L */
/*   */
/*   I/O-Kanäle   :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung   */
/*   ----- */
/*   uns.char * ucBN_Tag      Zeiger auf zu wandelnden Tag */
/*   im BN-Format (vgl. 20.2.1.4 ff.) */
/*   ----- */
/*   Ausgabe Parameter:   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung   */
/*   ----- */
/*   int      LZ_BN_TAGLEN      Anzahl der Bytes für das Tag */
/*   SONST: 0 */
/*   ----- */
/*   Transiente Parameter :   */
/*   ----- */
/*   Typ      Name      Dim.   Beschreibung   */
/*   ----- */
/*   *** KEINE *** */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   ----- */
/*   Original :   W. Fries      01.03.2007      1.3k   */
/*   Änderung:   */
/*   ----- */
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand : 01.03.2007 */
/*****
/*   Datei    :   BN_TAGNUM.C           */
/*   Aufruf   :   LZ_BN_TAGNUM ( ucBN_Tag ) */
/*****
/*   Beschreibung :   */
/*   Diese Routine liefert die Tag-Number für "ucBN_Tag".   */
/*   */
/*   aufgerufen von :   diversen   */
/*   */
/*   ruft auf :   */
/*   */
/*   I/O-Kanäle :   */
/*   */
/*****
/*   Eingabe Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   uns.char * ucBN_Tag      Zeiger auf zu wandelnden Tag   */
/*   im BN-Format (vgl. 20.2.1.4 ff.) */
/*   */
/*   -----   */
/*   Ausgabe Parameter:   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   int      LZ_BN_TAGNUM    Tag-Number   */
/*   SONST: UNDEF   */
/*   */
/*   -----   */
/*   Transiente Parameter :   */
/*   -----   */
/*   Typ      Name          Dim.    Beschreibung   */
/*   */
/*   *** KEINE ***   */
/*   */
/*****
/*   Autor      Datum      Version   Kurzzeichen   */
/*   -----   */
/*   Original :   W. Fries      01.03.2007   1.3k   */
/*   Änderung:   */
/*   */
/*****
/*   Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770   */
/*****

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_WPCAL.C           */
/*   Aufruf   :   LZ_BN_WPCAL ( ulObjId, ulProp, ulIndex, ucRECV_Buf,
/*                   usBufInd, iInvokeId, iBufLen, ucSEND_Buf) */
/*****
/*   Beschreibung :
/*   Diese Routine fügt die Antwort für das Schreiben von "ucRECVBuf"
/*   der Property "ulProp", ggf. mit Array-Index/Priority "ulIndex"
/*   des Objektes mit der Id. "ulObjId" am Puffer "ucSEND_Buf" an.
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei
/*   übergeben.
/*
/*   Unterstützt werden die folgenden Objekt-Typen:
/*   - Calendar (CAL)
/*
/*   aufgerufen von : LZ_BN_MSGWP
/*
/*   ruft auf : LZ_SCD_START, LZ_BN_OBJINSTANCE,
/*             LZ_BN_TAG2L, LZ_BN_TAGLEN, UT_DT2CI
/*
/*   I/O-Kanäle :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*   uns.long ulObjId      Id. des zu lesenden Objektes
/*   uns.long ulProp      zu schreibende Property
/*   uns.long ulIndex      zu schreibender Array-Index bzw.
/*                           für "present_value" die Priority
/*                           Sonst: UNDEF
/*   uns.char * ucRECV_Buf Puffer mit Teil des BN-Datagramm
/*                           mit spezifischen Daten der CAL-
/*                           Property
/*   int      iInvokeId    >=0: InvokeId
/*   int      iBufLen      >0: Länge des (Teil)-Datagramms
/*                           in "ucSEND_Buf" für TRANSMIT
/*   uns.char * ucSEND_Buf Puffer mit Teil des BN-Datagramm
/*                           Für BN_SEG_OPEN muss dieser den
/*                           BLVC, NPDU und APDU-Header
/*                           enthalten, jedoch OHNE Flag für
/*                           SEG, MORE, SA, Sequence-Number
/*                           und ProposedWindowSize.
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*   int      LZ_BN_WPCAL Returnkode der Funktion
/*                           >0: Anzahl zu sender Bytes
/*                           für segm. Datagramm
/*                           0: KEIN BN-Datagramm zu senden
/*                           <0: Anzahl zu sender Bytes
/*                           für ERROR-Datagramm (abs)
/*
/*                           interne Codes für "ier", die
/*                           Returnkode 0 liefern :
/*                           -1: Unzuläss. Objekt-Id.
/*                           -2: Unzuläss. Property
/*                           -3: Unzuläss. Array-Index
/*                           -4: Unzuläss. InvokeId
/*                           -5: Unzuläss. Länge
/*                           -6: Unzuläss. SEND-Buf.
/*                           -7: WRITE-Fehler
/*   -----
/*   Transiente Parameter :
/*   -----
/*   Typ      Name      Dim.      Beschreibung
/*   uns.short usBufInd      Index auf RECV_Buf mit den spez.
/*                           Daten der CAL-Property

```



```
/*          - bei Aufruf: auf OPEN-Tag          */
/*          - bei Return: hinter CLOSE-Tag      */
/*          für weitere Objekte bzw.          */
/*          Properties bei WPM                  */
/*          */
/*****
/*          Autor          Datum          Version          Kurzzeichen          */
/*          -----          -----          -----          -----          */
/* Original :      W. Fries          01.03.2007          1.3k          */
/* Änderung:
/*          */
/*****
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****
```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_WPDP.C             */
/*   Aufruf   :   LZ_BN_WPDP ( ulObjId, ulProp, ulIndex, fWert,
/*                   iInvokeId, iBufLen, ucSEND_Buf) */
/*****
/*   Beschreibung :
/*   Diese Routine fügt die Antwort für das Schreiben von "fWert"
/*   der Property "ulProp", ggf. mit Array-Index/Priority "ulIndex"
/*   des Objektes mit der Id. "ulObjId" am Puffer "ucSEND_Buf" an.
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei
/*   übergeben.
/*
/*   Unterstützt werden die folgenden Objekt-Typen:
/*   - Analog Input      (AI)
/*   - Analog Output     (AO)
/*   - Analog Value      (AV)
/*   - Binary Input      (BI)
/*   - Binary Output     (BO)
/*   - Binary Value      (BV)
/*   - MultiState Input  (MI)
/*   - MultiState Output (MO)
/*   - MultiState Value  (MV)
/*
/*   aufgerufen von :      LZ_BN_MSGWP, LZ_SCD_POLL
/*
/*   ruft auf :           UT_CFG_WRIT, UT_WSL_WRIT, UT_FND_TIME,
/*   div. Macros SET_FND_..., SY_F2IEEE
/*
/*   I/O-Kanäle :
/*
/*****
/*   Eingabe Parameter :
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   uns.long ulObjId    Id. des zu schreibenden Objektes
/*
/*   uns.long ulProp     zu schreibende Property
/*
/*   uns.long ulIndex    zu schreibender Array-Index bzw.
/*   für "present_value" die Priority
/*   Sonst: UNDEF
/*
/*   float    fWert     zu schreibender Wert
/*
/*   int      iInvokeId  >=0: InvokeId
/*
/*   int      iBufLen    >0: Länge des (Teil)-Datagramms
/*   in "ucSEND_Buf" für TRANSMIT
/*   0: nur FND-Befehle ausführen
/*   für SCD_POLL mit lz_subid=4
/*
/*   uns.char * ucSEND_Buf Puffer mit Teil des BN-Datagramm
/*   Für BN_SEG_OPEN muss dieser den
/*   BLVC, NPDU und APDU-Header
/*   enthalten, jedoch OHNE Flag für
/*   SEG, MORE, SA, Sequence-Number
/*   und ProposedWindowSize.
/*
/*   -----
/*   Ausgabe Parameter:
/*   -----
/*   Typ      Name      Dim.   Beschreibung
/*
/*   int      LZ_BN_WPDP Returnkode der Funktion
/*   >0: Anzahl zu sender Bytes
/*   für segm. Datagramm
/*   0: KEIN BN-Datagramm zu senden
/*   <0: Anzahl zu sender Bytes
/*   für ERROR-Datagramm (abs)
/*
/*   interne Codes für "ier", die
/*   Returnkode 0 liefern :
/*   -1: Unzuläss. Objekt-Id.
/*   -2: Unzuläss. Property
/*   -3: Unzuläss. Array-Index
/*   -4: Unzuläss. InvokeId
/*   -5: Unzuläss. Länge
/*   -6: Unzuläss. SEND-Buf.
/*
/*****

```

```

/* ----- */
/*      Transiente Parameter :                               */
/*      ----- */
/*      Typ      Name          Dim.      Beschreibung        */
/*      ----- */
/*      *** KEINE ***                                       */
/*      ----- */
/*      ***** */
/*      Autor      Datum      Version      Kurzzeichen      */
/*      ----- */
/* Original :      W. Fries      01.03.2007      1.3k         */
/* Änderung:                                             */
/* ----- */
/*      ***** */
/*      Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*      ***** */

```

```

/*****
/*   Produkt   :   GA - Knoten           Version   :   1.3k   */
/*   Modulname :   LZ/BNS LZ mit BACnet/IP Stand :   01.03.2007 */
/*****
/*   Datei    :   BN_WPSCD.C           */
/*   Aufruf   :   LZ_BN_WPSCD ( ulObjId, ulProp, ulIndex, ucRECV_Buf, */
/*               usBufInd, iInvokeId, iBufLen, ucSEND_Buf) */
/*****
/*   Beschreibung :   */
/*   Diese Routine fügt die Antwort für das Schreiben von "ucRECVBuf" */
/*   der Property "ulProp", ggf. mit Array-Index/Priority "ulIndex" */
/*   des Objektes mit der Id. "ulObjId" am Puffer "ucSEND_Buf" an. */
/*   Für definiertes "iInvokeId" wird die Ausgabe an die SEG-Datei */
/*   übergeben. */
/*   Unterstützt werden die folgenden Objekt-Typen: */
/*   - Schedule (SCD) */
/*   aufgerufen von :   LZ_BN_MSGWP */
/*   ruft auf :   LZ_SCD_START, LZ_SCD_WRITE, */
/*               LZ_BN_OBJTYPE, LZ_BN_OBJINSTANCE, */
/*               LZ_BN_TAG2L, LZ_BN_TAGLEN, */
/*               UT_DT_ADD, UT_DT2CI, SY_IEEE2F */
/*   I/O-Kanäle :   */
/*****
/*   Eingabe Parameter :   */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   uns.long  ulObjId      Id. des zu lesenden Objektes */
/*   uns.long  ulProp      zu schreibende Property */
/*   uns.long  ulIndex     zu schreibender Array-Index bzw. */
/*                       für "present_value" die Priority */
/*                       Sonst: UNDEF */
/*   uns.char * ucRECV_Buf  Puffer mit Teil des BN-Datagramm */
/*                       mit spezifischen Daten der SCD- */
/*                       Property */
/*   int       iInvokeId   >=0: InvokeId */
/*   int       iBufLen     >0: Länge des (Teil)-Dategramms */
/*                       in "ucSEND_Buf" für TRANSMIT */
/*   uns.char * ucSEND_Buf  Puffer mit Teil des BN-Datagramm */
/*                       Für BN_SEG_OPEN muss dieser den */
/*                       BLVC, NPDU und APDU-Header */
/*                       enthalten, jedoch OHNE Flag für */
/*                       SEG, MORE, SA, Sequence-Number */
/*                       und ProposedWindowSize. */
/*   ----- */
/*   Ausgabe Parameter: */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */
/*   int       LZ_BN_WPSCD  Returnkode der Funktion */
/*                       >0: Anzahl zu sendender Bytes */
/*                       für segm. Datagramm */
/*                       0: KEIN BN-Datagramm zu senden */
/*                       <0: Anzahl zu sendender Bytes */
/*                       für ERROR-Datagramm (abs) */
/*   interne Kodes für "ier", die */
/*   Returnkode 0 liefern : */
/*   -1: Unzuläss. Objekt-Id. */
/*   -2: Unzuläss. Property */
/*   -3: Unzuläss. Array-Index */
/*   -4: Unzuläss. InvokeId */
/*   -5: Unzuläss. Länge */
/*   -6: Unzuläss. SEND-Buf. */
/*   -7: WRITE-Fehler */
/*   ----- */
/*   Transiente Parameter : */
/*   ----- */
/*   Typ      Name          Dim.   Beschreibung */

```

```
/* uns.short   usBufInd           Index auf RECV_Buf mit den spez.*/
/*                                           Daten der SCD-Property      */
/*                                           - bei Aufruf: auf OPEN-Tag  */
/*                                           - bei Return: hinter CLOSE-Tag */
/*                                           für weitere Objekte bzw.*/
/*                                           Properties bei WPM          */
/*                                           */
/*****/
/*           Autor      Datum      Version      Kurzzeichen      */
/* -----|-----|-----|-----|-----|-----|-----|-----|-----| */
/* Original :   W. Fries      01.03.2007      1.3k                    */
/* Änderung:                                         */
/*                                           */
/*****/
/* Copyright: Daten-Technik Fries, D-85221 Dachau, Tel.:08131/80770 */
/*****/
```

### 5.12.5. Druckausgaben, Meldungstexte

\*\*\* KEINE \*\*\*

### 5.12.6. Datenkatalog

Im aktuellen Arbeitsverzeichnis werden temporär die folgenden Dateien angelegt:

<b>LZ_ICMD.WSL:</b>	Warteschlange der von der LZ empfangenen CMD-Datagramme
<b>LZ_ORSP.WSL:</b>	Warteschlange der an die LZ zu sendenden RSP-Datagramme (vorrangig)
<b>LZ_OUSM.WSL:</b>	Warteschlange der an die LZ zu sendenden USM-Datagramme
<b>LZBNdiii.SEG:</b>	Segmentierung zu sender BACnet-Datagramme

Aus dem Unterverzeichnis "lz/bns" werden die folgenden Dateien für SCHEDULE-Objekte verwendet:

<b>dp_units.txt:</b>	Zuordnung von FND-Dimensionen und BACnet-Units
<b>scd_&lt;Instanz&gt;.wsc:</b>	Weekly_Schedule
<b>scd_&lt;Instanz&gt;.esc:</b>	Exection_Schedule
<b>scd_&lt;Instanz&gt;.opr:</b>	List_Of_Object_Property_Reference
<b>scd_&lt;Instanz&gt;.scd:</b>	restliche SCHEDULE-Properties

Aus dem Unterverzeichnis "Iz/bns" werden die folgenden Dateien für CALENDAR-Objekte verwendet:

**cal\_<Instanz>.dtl:**            Date\_List

**cal\_<Instanz>.cal:**            restliche CALENDAR-Properties

### **5.12.7. Definition der Datenübertragung**

Grundlage für die Realisierung sind :

- ASHRAE 135-2004
- Angebot JCISES/6101 vom 11.10.2006 (Auftrag vom 13.11.2007)
- Die IZ- und LZ-seitig vom Auftraggeber bereitgestellte Testkonfiguration bestehend aus einem MS-NAE5520-0E und einer BACnet-fähigen Managementstation M5/MPlus der Fa. Johnson Controls Systems & Services (JCI) dient als Referenzsystem in folgender Konfiguration:
  - MPlus Explorer                    Build 5.4.13
  - MpOpc Launcher
  - MAlarm
  
  - Sitebook                        Version 5.3
  - Mplus DBI-Tool                Release 5.4.13

Testweise wurden am Treiber auch Produkte folgender Hersteller erfolgreich aufgeschaltet :

- noch KEINE

**Hinweis:** IZ-Aufschaltungen zeigten, dass herstellerspezifische Besonderheiten - technisch und organisatorisch - sowie abweichende Stände und Auslegungen der BACnet-Spezifikation system- und projekt-spezifische Festlegungen erforderlich machen.

Es wird empfohlen, VOR jedem Einsatz des Treibers die Eignung des vorgesehenen BACnet-Produkts im Rahmen einer Testaufschaltung sicherzustellen.

Folgende Erweiterungen bzw. Abweichungen wurden festgelegt :

- Erweiterungen mit Ergänzungs-Angebot JCISES/6121 vom 28.12.2006 (Auftrag vom 19.02.2007, DS-RPM-B, SEGMENTED\_TRANSMIT, DS-WPM-B)
- Vendor-Id. (LZBN in BNS\_XXL.EXE) : 201 (DTF Daten - Technik FRIES)

Folgende Standard Object Types lassen sich auf Datenpunkte abbilden :

- Binary Input (BI <=> i.a. 1-stuf. Meldepunkt)
- Binary Output (BO <=> i.a. 1-stuf. Schalterpunkt)
- Binary Value (BV <=> i.a. 1-stuf. Schalterpunkt)
- Analog Input (AI <=> i.a. Messpunkt)
- Analog Output (AO <=> i.a. Stellpunkt)
- Analog Value (AV <=> i.a. Stellpunkt)
- MultiState Input (MI <=> i.a. n-stuf. Meldepunkt))
- MultiState Output (MO <=> i.a. n-stuf. Schalterpunkt)
- MultiState Value (MV <=> i.a. n-stuf. Schalterpunkt)

Folgende Standard Object Types dienen erweiterten Funktionen :

- Device obligatorisch
- Schedule Zeit-Schaltprogramme (ZSP)
- Calendar Schaltprogramm-Kalender (SPK)

Unterstützt werden folgende BIBB's:

- DS-RP-B
- DS-RPM-B
- DS-WP-B
- DS-WPM-B
- DS-COVU-B
- AE-N-B



- AE-INFO-B (ab Version 1.4b)
- SCHED-I-B
- DM-DDB-B
- DM-DOB-B
- DM-DCC-B
- DM-TS-B
- DM-RD-B

**Wichtiger Hinweis:** Wesentliche Teile der Kommunikation des GA-Knotens mit der bestehenden Leitzentrale Haustechnik (LZH) lassen sich mit BACnet/IP nicht oder nur eingeschränkt abbilden. Als Folge davon kommt es zu Einschränkungen bei Parametrierung und Funktion des GA-Knotens.

**5.12.7.1 LAN-Schnittstelle zum Anschluss der LZ**

Die BACnet-Managementstation wird über ein lokales TCP/IP-Netzwerk (BACnet/IP) unter LINUX angeschlossen. Das Betriebssystem DOS wird NICHT unterstützt.

Bei der Einrichtung des Rechner ist darauf zu achten, dass dieser dem bzw. den Netzwerken zugeordnet ist, über die die anderen BACnet-Devices erreichbar sind (vgl. "hostname -i").

**5.12.7.2 Spezifische Anpassungen in der Konfigurationsdatei**

LZ-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"LZ"	Zeilen-Kennung (OHNE ''')
2	4	6	3	Treiber-Id.	>0: Nummer unter der der zugehörige IZ-Treiber angesprochen wird. Dieser Parameter ist abhängig von der Programm-Generierung für den GA-Knoten

					und wird abhängig davon vorgegeben.
					0: reserviert für zukünftige Erweiterungen *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNS: stets 0
3	8	10	3	Treiber-Subid.	lfd. Nummer der an den Treiber angeschlossenen DDC-IZ beginnend mit 0.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNS: stets 0
4	12	34	23	LZ-/Treiber-Text	Klartext zur Beschreibung der IZ bzw. des Treibers
5	36	38	3	Verbindungs-Art	natürliche Zahl kodiert gemäß "DV-technisches Feinkonzept des GA-Knotens" vom 22.9.95 (vgl. Seite 20).  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNS: 40 (UDP/IP)
6	40	42	3	Protokoll	Verbindungs-Protokoll als Zahl kodiert wie folgt : 0: BACnet Version 1  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNS: 0
7	44	50	7	SDEV	Eigene Device-Instance gemäß BACnet-Vorgaben
8	52	56	5	SNET	Eigene Network-Number für den Routing-Betrieb gemäß BACnet-Vorgaben. Fehlt diese, so wird der BNS als lokales BACnet-Device betrieben.
9	58	69	12	SADR	Eigene MAC gemäß BACnet-Vorgabe (hexadezimal). Fehlt diese, so wird SADR aus IP-Adresse und -Port gebildet
10	71	256	<=190	Schnittst.-Bez.	Bezeichnung der Schnittstelle abhängig von der Verbindungsart und dem LZ-Typ.  Für UDP/IP Port im Format "[IP-Netz#]Port" mit IP-Netz aaa.bbb.ccc.ddd (IP4-Adr.) und "Port" gemäß BACnet-Vorgaben, i.a. 47808 (0xBAC0)  Wird durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen.

Abhängig vom Betriebssystem des GA-Knotens sind für die Schnittstellen-Bezeichnung spezifische Beschränkungen der Syntax und Zeichenanzahl zu beachten (z.B. max. 8 Zeichen für Dateiname unter MS-DOS, ...).

**Hinweis:** Die Werte von SDEV, SNET und SADR müssen - wie auch IP-Adresse und -Port - mit dem System-Verantwortlichen abgestimmt sein, um Konflikte mit bestehenden und zukünftigen BACnet-Geräten zu vermeiden !!!

Die Notification-Classes werden über gesonderte Vereinbarungszeilen definiert.

NC-VEREINBARUNGS-Zeilen haben den folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"NC"	Zeilen-Kennung (OHNE '"')
2	4	10	7	NotifClass	Nummer der Notication-Class
3	12	14	3	Prio_OFFNORMAL	Priorität für TO-OFFNORMAL
4	16	18	3	Prio_FAULT	Priorität für TO-FAULT
5	20	22	3	Prio_NORMAL	Priorität für TO-NORMAL
6	24	24	1	Ack_Required	Bitweise Flags dezimal kodiert (0,...,7) gemäß BACnet-Spez.  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNS: 0
7	26	256	<=230	CharStrings	CharacterString's (optional) durch ";" getrennt für : - Object_Name - Description - Profile_Name  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !  *** Bis dahin gilt : *** BNS: LEER

Auf eine NC-Zeile nachfolgende Fortsetzungszeilen definieren jeweils einen Eintrag in der Recipient-List wie folgt:

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	2	2	"++"	Fortsetzungszeile (OHNE '"')
2	4	10	7	Device	Instanz des Device-Objects an das gesendet werden soll
3	12	18	7	Process-Id.	Process-Identifizier
4	20	20	1	ConfirmedNotif	Flags für Confirmed Notif. 0: Unconfirmed Notification 1: Confirmed Notification  *** Wird noch nicht ausgewertet *** und bleibt reserviert für *** zukünftige Erweiterungen !

```

*** Bis dahin gilt :
*** BNS: 0

5      22      22      1      Transitions      Bitweise Flags dezimal
kodierte (0,...,7) gemäß
BACnet-Spez.

*** Wird noch nicht ausgewertet
*** und bleibt reserviert für
*** zukünftige Erweiterungen !

*** Bis dahin gilt :
*** BNS: 7
    
```

**Hinweis:** Derzeit wird max. 1 Eintrag in der Recipient-List unterstützt. Angaben zu "ValidDays", FromTime" und "ToTime" werden momentan noch nicht unterstützt. Nachfolgende Spalten bleiben reserviert für zukünftige Erweiterungen ! Bis dahin gilt : 7 Tage, 24 Stunden

Die BACnet-Adressen für die FND-Datenpunkte werden in den LS-Zeilen wie folgt kodiert:

"**ddddddd.tttt#oooooof'ppppp**" mit :

<b>ddddddd:</b>	DEVICE-Id.	(Instance des DEVICE-Objektes der LZ)
<b>tttt:</b>	OBJEKT-Typ	(Object-Type des Object-Identifizier) wie folgt
- AI	Analog Input	(Synonym für 0000)
- AO	Analog Output	(Synonym für 0001)
- AV	Analog Value	(Synonym für 0002)
- BI	Binary Input	(Synonym für 0003)
- BO	Binary Output	(Synonym für 0004)
- BV	Binary Value	(Synonym für 0005)
- MI	Multi-State Input	(Synonym für 0013)
- MO	Multi-State Output	(Synonym für 0014)
- MV	Multi-State Value	(Synonym für 0019)
- nnnn	max. 4-stellige dezimale Nummer des Object-Type gemäß "BACnetObjectType"	

<b>ooooooo:</b>	OBJEKT-Nr.	(Instance des Object-Identifizier)
<b>ppppp:</b>	PROPERTY-Id.	Fehlt diese Angabe, so wird der "Present_Value"

(00085) der  
 Führungsgröße bzw. dem explizit angegebenen "actual" zugeordnet  
 \*\*\* Wird noch nicht ausgewertet und bleibt reserviert für  
 \*\*\* zukünftige Erweiterungen ! Bis dahin gilt :  
 \*\*\* BNS: 00085

Die LS-Zeilen werden durch das Zeilenabschlußzeichen (<CR> und/oder <LF>, je nach verwendetem Editor) abgeschlossen, d.h. optionale Parameter müssen nicht mittels Leerzeichen aufgefüllt werden. Sollen führende optionale Parameter nicht angegeben werden, so sind die zugeordneten Spalten mittels Leerzeichen aufzufüllen.

Die LS-Zeile in der Konfigurationsdatei hat ab Spalte 29 folgenden Aufbau :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
5	29	54	26	Adr_DP_Id	BACnet-Adresse für den gesamten FND-Datenpunkt

6	56	62	7	NotifClass	Notification Class gemäß BACnet-Spez. 13.8.1.6 für Event-Notification Sonst (leer): COV-Notification (Dieser Parameter kann optional angegeben werden)
7	64	64	1	NotifType	Notify Type kodiert gemäß BACnet-Spez. unter Kap.21 zu "BACnetNotifyType" Sonst: ALARM, falls NotifClass definiert (Dieser Parameter kann optional angegeben werden)
8	66	70	5	KL_Id	Nummer der Kennlinie, die zur Umrechnung "Present_Value" in TO-OFFNORMAL-Event angewendet werden soll (Entspricht momentan dem Index in der KL-Struktur) bzw. Umrechnungsfunktion (vgl. Properties "Polarity", "Alarm_Values"). Vorläufig gilt: "INV01" PV==0: OFFNORMAL PV!=0: NORMAL Sonst PV==0: NORMAL PV!=0: OFFNORMAL  (Dieser Parameter kann optional angegeben werden)

**5.12.7.3 Initialisierung der BACnet**

Erreichbar sind alle die BACnet-Devices, die zum gleichen IP-Netzwerk wie der GA-Knoten gehören bzw. über entsprechende IP-Router erreichbar sind.

BACnet-Broadcasts sind an die BACnet-Device gerichtet, die mit dem GA-Knoten auf dem in der LZ-Zeile angegebenen Port kommunizieren. Diese BACnet-Devices können per "Dynamic Device Binding" erkannt und in der Routing-Tabelle geführt werden. Alle anderen BACnet-Devices müssen mittels der Device-Datei "lzbnddev.bn" im Unterverzeichnis "lz" manuell für die Routing-Tabelle parametrisiert werden. Diese hat folgendes Format :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	7	7	DDEV	BACnet Device-Id.
2	9	13	5	DNET	BACnet-Netzwerk
3	15	26	12	DADR	BACnet-Adresse (MAC)
4	28	48	21	IP-Adr./Port	IP-Adresse und -Port im Format "aaa.bbb.ccc.ddd#ppppp"

Die BACnet wird wie folgt initialisiert :

- Senden von "I-Am-Router-To-Network" für SNET aus der IZ-Zeile als Broadcast
- Senden von "Who-Is-Router-To-Network" als Broadcast

- Senden von "I-Am" als Broadcast mit : max. APDU-Length 1476 Bytes, Segmentation: SEGMENTED\_TRANSMIT
- Senden von "Who-Is" als Broadcast
- Erwarten und Empfangen der "I-Am"-Datagramme mit Aufbau der Routing-Tabelle

Wartezeiten bei der Initialisierung und zyklisch unter der Laufzeit wird der Abgleich der Routing-Tabelle durch Senden von "Who-Is" als Broadcast erzwungen. Dabei empfangene Datagramme mit abweichender BACnet- oder IP-Adresse führen zur Aktualisierung der Routing-Tabelle.

Wird ein BACnet-Zeitabgleich empfangen (DM-TS-B), so gibt der LZ-Treiber die neue Systemzeit (Datum/Uhrzeit) mittels der Routine LZ\_ZA\_TIME an das Hauptprogramm des GA-Knotens weiter (vgl. ZA-Zeile in der Konfigurations-Datei und Hilfsprogramm TSYNC).

Der LZ-Treiber ist - ungeachtet der BACnet-Interoperability-Building-Blocks (BIBB) - vorläufig begrenzt auf die BACnet-Services, die für die Umsetzung auf das o.g. Referenzsystem benötigt werden.

#### **5.12.7.4 Spontane Meldungen**

Das BACnet-Protokoll unterstützt zwar spontane Meldungen (Change-Of-Value (COV), AlarmEvents (AE)), diese sind jedoch nicht für alle BACnet-Geräte zwingend.

Der LZ-Treiber behandelt spontane FND-Meldungen (USM) oder gemäß ES-Zeile zu protokollierende Werte wie folgt:

- wird einem Datenpunkt in der LS eine Notification-Class (vgl. NC-Zeile) zugeordnet, so wird die zugehörige Event-Notification gesendet (BIBB: AE-N-B). Anstehende Alarme/Events werden auch auf Anfrage mittels des Service "GetEventInformation" geliefert (BIBB: AE-INFO-B; ab Version 1.4b)
- ansonsten erfolgt die Meldung als unsolicited COV (BIBB: DS-COVU-B)

#### **5.12.7.5 Polling der Datenpunkte**

Der LZ-Treiber führt für jeden Datenpunkt ein sekundäres Prozess-Abbild und beantwortet lesende Zugriffe (ReadProperty, BIBB: DS-RP-B, DS-RPM-B) der BACnet-Geräte aus dem sekundären Prozess-Abbild. Dieses wird mit jedem IZ-seitig empfangenen RSP bzw. USM aktualisiert, so dass Zykluszeit, Hysterese, Synchronisationszeit und Abfrageintervall in der ES-Zeile geeignet parametrisiert werden müssen.

Dieses Verfahren verhindert eine Überbelastung der IZ-seitigen Kommunikation durch das LZ-seitige Polling.

Soll der aktuelle Zustand/Wert des Datenpunktes aus der IZ explizit gelesen werden, so muss die Property "Event-Enable" gelesen werden.

**Hinweis:** Diese Property wird i.a. nur dann abgefragt, wenn am o.g. Referenzsystem eine Untersuchung/Aktualisierung unter dem Menüpunkt M-View aktiv angefordert wurde. Daraufhin wird der Wert des sekundären Prozess-Abbildes SOFORT als Ergebnis geliefert und zusätzlich das IZ-seitige Lesen des Datenpunkt initiiert. Das zugehörige RSP löst dann eine COV- bzw. Event-Notification aus.

#### **5.12.7.6 Zugriffskontrolle**

Der IZ-Treiber prüft die Modifikations-Berechtigung für LZ-seitige Zugriffe gemäß FND-Spezifikation.

Zusätzlich unterstützt der LZ-Treiber die Priorität-Steuerung gemäß Abschnitt 19 der ASHRAE 135-2004.

**Hinweis:** Bei diesem Verfahren werden Vorgabewerte abhängig von der zugeordneten Priorität vermerkt und nur dann IZ-seitig gesendet, wenn KEINE höher priorisierter Vorgabe besteht.

#### **5.12.7.7 Grenzwert-Überwachung**

Die Alarm-Grenzwerte der FND-Datenpunkte ("alarm\_low" bzw. alarm\_high") werden den Properties "LOW\_Limit" und "HIGH\_Limit" direkt zugeordnet.

**Hinweis:** Warn-Grenzwerte der FND-Datenpunkte ("warning\_low" bzw. "warning\_high") werden von BACnet nicht unterstützt.

#### **5.12.7.8 Ausblendung der Betriebsmeldungen**

Betriebsmeldungen (Änderung des Status/Meldungsstufe, Grenzwert-Verletzung,...) werden durch den IZ-Treiber ausgeblendet, wenn in der ES-Zeile eine entsprechende Parametrierung besteht.

#### **5.12.7.9 Zuordnung der Attribute und Dimensionen**

Die den FND-Datenpunkten zugeordneten Attribute und Dimensionen (vgl. Anhang E der FND-Spezifikation sowie die Dateien CFG/DP\_ATB.TXT und CFG/DP\_DIM.TXT) werden auf die BACnet-Properties wie folgt abgebildet:

Für Melde- bzw. Schalterpunkte werden die aus CFG/DP\_ATB.TXT gelesenen Attribut-Texte direkt auf die BACnet-Properties "Active\_Text" und "Inactive\_Text" der Binary-Objekte bzw. "State\_Text" der Multistate-Objekte abgebildet.

Für Mess-, Stell- und Zählpunkte wird die zugeordnete Dimension aus Anhang E der FND-Spezifikation sinngemäß auf die BACnet-Property "Units" abgebildet. Optional kann beim Programmstart eine Zuordnungsliste aus der Datei LZ/BNS/DP\_UNITS.TXT geladen werden. Diese hat den Aufbau wie CFG/DP\_DIM.TXT, jedoch anstelle der Dimensionstexte die Nummer gemäß BACnetEngineeringUnit .



**5.12.7.10 Segmentierung**

Das o.g. Referenzsystem kommuniziert unkontrolliert mittels DS-RPM und erzwingt somit Responses (ComplexACK), die über die max. APDU-Länge hinausgehen können.

In Hinblick auf zukünftige Erweiterung zur Unterstützung auch von segmentierten Requests wird folgendes Verfahren festgelegt:

- potentiell segmentierte Datagramme werden in einer SEG-Datei zwischengespeichert
- gesteuert durch die InvokeId unterstützt das Verfahren das unabhängige und ggf. verschachtelte Receive und Transmit
- der Header eines BN-Datagrammes - bestehend aus BLVC, NPDU und APDU-Header - als auch die APDU-Daten werden für unsegmentierte BN-Datagramme gespeichert, d.h. die Segmentierung findet nur bei der Kommunikation mit der LZ statt, im Programm werden stets unsegmentierte BN-Datagramme beliebiger Länge behandelt
- bei Bedarf werden vor dem Parsen empfangener BN-Requests dynamische Puffer ausreichender Länge allociert
- Die SEG-Datei wird gelöscht, sobald für BN\_SEGRD ein unsegmentiertes BN-Datagramm als Ergebnis geliefert wurde oder kein weiteres Segment geliefert werden kann (i.a. Aufruf von BN\_SEGRD nach SEG-ACK für letztes Segment mit MORE=0)

Die SEG-Datei hat folgenden Namen:

"LZBNdiii.SEG" mit :

- d: Direction kodiert als  
 "T": Transmit an die LZ  
 "R": Receive von der LZ

iii: InvokeId

Die SEG-Datei ist eine Binär-Datei und hat folgenden Aufbau :

lfd.Nr.	Typ	Name	Bemerkung
1	long	lPosSEG	Position zum Lesen des nächsten Segments aus dem APDU UNDEF: solange SEG-Datei beschrieben wird
2	long	lMaxSegm	Max. Anzahl der Segmente; sonst UNDEF
3	long	lMaxAPDU	Max. Länge des APDU je Segment; sonst UNDEF
4	long	lLenHDR	Länge des nachfolgenden Headers ("ucHDR")
5	long	lLenAPDU	Länge der nachfolgenden APDU-Daten ("ucAPDU")
6	long	lLenRdAPDU	Anzahl der bereits aus "ucAPDU" gelesenen

		Bytes
7	uchar ucHDR[lLenHDR]	Header für ein unsegmentiertes BN-Datagramm beliebiger Länge bestehend aus BLVC, NPDU und APDU-Header (ohne SEG, MORE, SA, SequenceNumber, ProposedWindowSize)
8	uchar ucAPDU[lLenAPDU]	APDU-Daten für ein unsegmentiertes BN-Datagramm

Folgende Routinen behandeln die Segmentierung :

- BN\_SEGWR Erzeugen bzw. Erweitern der SEG-Datei für ein InvokeId
- BN\_SEGRD Lesen eines Segmentes aus der SEG-Datei für ein InvokeId, ggf. mit automatischem Löschen der SEG-Datei
- BN\_SEGRM Löschen der SEG-Datei für eine InvokeId

**5.12.7.11 Zeit-Schaltprogramme (SCHEDULE)**

Grundlage für die Behandlung der SCHEDULE-Objekte ist Abschnitt 12.24 der BACnet-Spezifikation sowie deren Behandlung im o.g. Referenzsystem.

Die Properties werden im Unterverzeichnis LZ/BNS in Dateien gespeichert, die folgender Namens-Syntax genügen (<Instanz> = Nummer für Instanz des SCHEDULE-Objektes) :

- scd\_<Instanz>.wsc Weekly\_Schedule
- scd\_<Instanz>.esc Exection\_Schedule
- scd\_<Instanz>.opr List\_Of\_Object\_Property\_Reference
- scd\_<Instanz>.scd restliche SCHEDULE-Properties

Die WSC-Datei enthält zeilenweise die wöchentlich auszuführenden Schalt-/Stell-Befehle wie folgt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	14	14	Wochentag/Zeit	Chronoindex 000000tthhmmss mit tt = 01: Montag,...,07:Sonntag >7 (0x1F): jeden Tag hh: Stunde (00,...,23) mm: Minute (00,...,59) ss: Sekunde (00,...,59)
2	16	29	14	Vorgabe-Wert	Gleitkomma-Wert oder FFFFFFFF für UNDEF

Die ESC-Datei enthält zeilenweise die als Ausnahme auszuführenden Schalt-/Stell-Befehle wie folgt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	14	14	VON-Datum/Zeit	Datum/Zeit ab dem die Ausnahme gilt kodiert als: - jjjjmmtt (Datum explizit) - 00000000 (CalendarRef.) - 99mmwwdd (analog WeekNDay)  - hhhmss (Zeit derzeit NICHT ausgewertet, aber reserviert )  - 999...99 UNDEF
2	16	29	14	BIS-Datum/Zeit	Datum/Zeit bis zu dem die Ausnahme gilt kodiert als: - jjjjmmtt (Datum explizit) - nnnnnnnn (Instanz für CalendarRef.) - ???????? (nicht verwendet für WeekNDay)  - hhhmss (Zeit derzeit NICHT ausgewertet, aber reserviert )  - 999...99 UNDEF
3	31	36	6	Zeit	Chronoindex, wobei für die Zeit gilt : hhhmss Sonst: 999999 für UNDEF
4	38	51	14	Vorgabe-Wert	Gleitkomma-Wert oder LEER bzw. FFFFFFFF für UNDEF
5	53	54	2	EventPriority	wie für SpecialEvent (1,...,16) Sonst: UNDEF

**Hinweis:** Jeder Eintrag in der List\_Of\_BACnetTimeValue führt zu einem eigenen Eintrag in der ESC-Datei. Dabei bleiben VON-/ und BIS-Datum unverändert.

Die OPR-Datei enthält zeilenweise die auszuführenden Object-/Property-Zuordnungen wie folgt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
1	1	26	26	Adr_DP_Id	BACnet-Adresse, inkl. Property wie oben beschrieben, wobei die DEVICE-Id. ignoriert wird

Unterstützt werden folgende Properties :

- Present\_Value      Binary-, Multistate-, Analog-Output/-Value, die einem FND-Schalt-, Stell- oder Zählpunkt zugeordnet sind
- Low\_/High\_Limit    Analog-Output/-Value, die einem FND-Messpunkt zugeordnet sind

Die SCD-Datei enthält die restlichen parametrierbaren SCHEDULE-Properties zeilenweise wie folgt :

- Object_Name	CharacterString
- Description	CharacterString
- EffectivePeriod	Datum/Zeit_VON-..._BIS (jjjjmmthhmmss-jjjjmmth
- Schedule_Default	float
- Priority_For_Writing	1,...,16; Sonst: UNDEF

**Hinweis:** Die Object-Type und Object\_Id. werden implizit aus dem Prefix "scd\_" und der <Instanz> im Datei-Namen gebildet. Modifizierende Zugriffe auf Properties müssen ggf. die gesamte Liste umfassen.

Nach dem Programm-Start, jedem Write-Property auf eine der SCD-Dateien oder der Ausführung eines Schalt-/Stell-Befehls wird Datum/Zeit und Vorgabe-Wert des nächsten auszuführenden Schalt-/Stell-Befehls für jedes betroffene SCHEDULE-Objekt ermittelt und programmintern vermerkt.

Dabei wird ggf. auch Present\_Value, Status\_Flags, Reliability und Out\_Of\_Service für das SCHEDULE-Object ermittelt und programmintern vermerkt.

#### **5.12.7.12 Kalender (CALENDAR)**

Grundlage für die Behandlung der CALENDAR-Objekte ist Abschnitt 12.9 der BACnet-Spezifikation sowie deren Behandlung im o.g. Referenzsystem.

Die Properties werden im Unterverzeichnis LZ/BNS in Dateien gespeichert, die folgender Namens-Syntax genügen (<Instanz> = Nummer für Instanz des CALENDAR-Objektes) :

- cal_<Instanz>.dtl	Date_List
- cal_<Instanz>.cal	restliche CALENDAR-Properties

Die DTL-Datei ist analog zur WSC-Datei des SCHEDULE-Objekte aufgebaut und enthält zeilenweise Datum, Datumsbereich bzw. WeekNDay für die ausnahmsweise auszuführenden Schalt-/Stell-Befehle wie folgt :

Spalte	Anfang	Ende	Länge	Inhalt	Bemerkung
--------	--------	------	-------	--------	-----------

-----					
1	1	14	14	VON-Datum/Zeit	Datum/Zeit ab dem die Ausnahme gilt kodiert als: - jjjjmmtt (Datum explizit) - 99mmwdd (analog WeekNDay)  - hhhmss (Zeit derzeit NICHT ausgewertet, aber reserviert )  - 999...99 UNDEF
2	16	29	14	BIS-Datum/Zeit	Datum/Zeit bis zu dem die Ausnahme gilt kodiert als: - jjjjmmtt (Datum explizit) - ???????? (nicht verwendet für WeekNDay)  - hhhmss (Zeit derzeit NICHT ausgewertet, aber reserviert )  - 999...99 UNDEF

Die CAL-Datei enthält die restlichen parametrierbaren SCHEDULE-Properties zeilenweise wie folgt :

- Object\_Name                      CharacterString
- Description                      CharacterString

**Hinweis:** Die Object-Type und Object\_Id. werden implizit aus dem Prefix "scd\_" und der <Instanz> im Datei-Namen gebildet. Modifizierende Zugriffe auf Properties müssen ggf. die gesamte Liste umfassen.

### **5.12.7.13 Erweiterung CFG-Datei um BACnet-spezifische Parameter (CFG4BN)**

Das Hilfs-Programm CFG4BN (cfg4bn.exe) dient zum automatischen Erweitern der Konfigurations-Datei des GA-Knotens um BACnet-spezifische Parameter.

Soweit noch nicht vorhanden wird die LZ-Zeile eingefügt sowie für jeden einzelnen FND-Datenpunkt die zugehörige LS-Zeile mit Zuordnung der jeweiligen Object-Id. (Object-Type, Instanz, ggf. Notification-Class).

Soweit ein optionaler Schalter "/nxxx" für die Generierung der Notification-Behandlung angegeben wurde, wird jede LS-Zeile um die Parameter für NotificationClass und NotifyType ergänzt. Folgende Notification-Behandlungen sind realisiert für die Angabe von "xxx" :

- 1 Auswertung des 1.Zeichens der Datenpunkt-Bezeichnung (entspricht der Property "description") im Wertebereich 1 bis 6 als Instanz der zuzuordnenden Notification-Class. Die zugehörigen NC-Zeilen werden in der Konfigurationsdatei eingefügt mit dem 10-fachen Wert für Prio\_OFFNORMAL, Prio\_FAULT und Prio\_NORMAL (z.B. Universität Stuttgart).

**Beispiel:** Die Datenpunkt-Beschreibung "2\_Frost-Alarm" führt zur Zuordnung der Notification-Class 2 mit den Prioritäten 20

Der Aufruf erfolgt durch :

```
CFG4BN <Name der Konfigurations-Datei> <Device.-Id. der LZ (nur Instanz)> [ Schalter für Erweiterung] [ Schalter für Test]
```

Folgende Schalter für Erweiterung können optional angegeben werden:

```
/nxxx      Notification-Behandlung (z.B. xxx=1 für Universität Stuttgart)
```

## **6. Hilfsprogramme**

### **6.1. Generierung der CFG-Datei (GEN\_CFG und ESN2DOS)**

Anhand der FND-Referenzdatei gemäß der FND-Spezifikation (Teil 4, FND-Konformitätssicherung) und von Vorlagendateien erzeugt das Programm GEN\_CFG die Konfigurationsdatei für den GA-Knoten bzw. die Standard-Schnittstellen-Adapter.

Ab Version 1.42d ist GEN\_CFG auch im Arbeitsverzeichnis der GA-Knotens ablauffähig und nutzt die vorhandene Unterverzeichnis-Struktur. Zusätzlich wird eine Kopie von GEN\_CFG.CFG im Unterverzeichnis CFG mit dem Namen gak\_OOOO.cfg angelegt, wobei OOOO der beim Aufruf angegebenen Objekt-Kennung entspricht.

Der Aufruf erfolgt durch :

```
GEN_CFG <Name der FND-Ref.-Datei> <Obj.-Kennung> <IZ-Treiber-Id.[,IZ-Treiber-Subid.]> [ Schalter für Generierung] [ Schalter für Test]
```

Folgende Schalter für Generierung können optional angegeben werden:

- /a Anfüge-Modus
- /cl Name der ESn-Datei in Kleinbuchstaben (lowercase)
- /cu Name der ESn-Datei in Großbuchstaben (uppercase)
- /p\_ Zeichen für Platzhalter im Datenamen (hier "\_"; Voreinstellung)
- /f Fremd-Generierung

Wird anstelle der FND-Referenzdatei eine Konfigurationsdatei angegeben, so erkennt GEN\_CFG diese automatisch und bietet nun die Aktualisierung der ES-Zeilen anhand der Vorlagedateien an. Wird eine Objekt-Kennung angegeben, so wird diese verwendet, sonst wird die Objekt-Kennung aus der vorhandenen ES-Zeile übernommen.

Der Aufruf erfolgt durch :

```
GEN_CFG <Name der Konfigurations-Datei> [<Obj.-Kennung> ] [ Schalter für Test]
```

Folgende Schalter für Generierung können optional angegeben werden:

- /cl Name der ESn-Datei in Kleinbuchstaben (lowercase)
- /cu Name der ESn-Datei in Großbuchstaben (uppercase)
- /p\_ Zeichen für Platzhalter im Datenamen (hier "\_"; Voreinstellung)
- /f Fremd-Generierung

Im aktuellen Arbeitsverzeichnis bzw. Unterverzeichnis GEN\_CFG (ab Version 1.42d) müssen folgende Dateien vorhanden sein (unter LINUX in Kleinbuchstaben):

- GEN\_CFG.XX (enthält die allgemeinen Vereinbarungs-Zeilen wie z.B. DR, EL, ER, KL, ZA)
- GEN\_CFG.IZ (enthält je IZ-Treiber-Id. die Vorlage für die IZ-Zeile)
- GEN\_CFG.ES (enthält je Datenpunkt-Typ die Vorlage für die ES-Zeile und optional die Beschreibung zum Aufbau eines bis zu 16-stelligen Schlüssels "sss", mit dem ES-Zeilen aus spezifischen Dateien geladen werden können )

- sss.ESn (enthält für den Datenpunkt-Typ "n" die Vorlage für die spezifische ES-Zeile für Datenpunkte gemäß dem Schlüssel "sss", vgl. Datenkatalog. Der Dateiname kann Platzhalter - z.B. "\_" enthalten.)
- GEN\_CFG.IS (enthält je IZ-Treiber-Id. und je Datenpunkt-Typ die Vorlage für die IS-Zeile)

**Hinweis:** Für den Programmlauf sollte ein gesondertes Arbeitsverzeichnis (z.B. Unterverzeichnis GEN\_CFG zum Arbeitsverzeichnis des GA-Knotens bzw. SSA) eingerichtet werden, damit beim Löschen temporären Dateien des GA-Knotens bzw. SSA die Vorlagedateien für GEN\_CFG erhalten bleiben.

Unter DOS darf die Umgebungs-Variable DIRCMD nicht gesetzt sein !

Mit Erweiterung der Anzahl der einbindbaren IZ-Treiber (z.B. GAK\_XXL) müssen auch die Dateien GEN\_CFG.IZ und GEN\_CFG-IS entsprechend erweitert und in das zugehörige Arbeitsverzeichnis übernommen werden.

Ab Version 1.3j unterstützt GEN\_CFG auch die Fremd-Generierung (vgl. Schalter "/f"). Damit können z.B. unter DOS Konfigurationsdateien für ein Fremd-System wie LINUX erzeugt werden. Die o.g. Dateien müssen dann für das jeweilige Fremd-System angepasst sein (z.B. Einträge für max. Anzahl der IZ's, Schnittstellen-Bezeichnungen).

Die Datei GEN\_CFG.CFG wird aus der FND-Referenzdatei wie folgt erzeugt:

- Übernahme der 1. Zeile aus der FND-Referenz-Datei
- Anfügen von GEN\_CFG.XX
- Übernahme der Vorlage-Zeile aus GEN\_CFG.IZ abhängig von der angegebenen IZ-Treiber-Id.
- Erzeugen der OB-Zeile mit DUMMY-Klartext
- zeilenweise Übernahme der DP-Zeilen aus der FND-Referenz-Datei mit folgenden weiteren Bearbeitungsschritten :
  - Übernahme der Vorlage für die ES-Zeile aus "sss.ESn" bzw. GEN\_CFG.ES abhängig vom Datenpunkt-Typ mit Anpassung der Objekt-Kennung und der Datenpunkt-Adresse. Vorrangig wird die ESn-Datei mit dem höheren Festlegungsgrad verwendet, d.h. die linksbündig am weitesten explizit definiert ist. Erst wenn keine ESn-Datei gefunden wurde wird die Vorlage aus GEN\_CFG.ES verwendet. Ist das Flag für das Meldeverhalten oder die Protokollierung nicht angegeben, so wird dieses für Melde- und Schaltpunkte automatisch auf "sofort zur LZH" (001) bzw. "alles protokollieren" (255) gesetzt, wenn die ATB\_ID eine Stör-, Gefahr- oder Wartungs-Meldung zuordnet (ATB\_ID: 25,...,29 oder 44). Sonst wird "zwischen gespeichert" (000) und die Protokollierung unterdrückt (000). In der Datei GEN\_CFG.LOG wird für jeden Datenpunkt protokolliert, aus welcher Datei die Vorlage für die ES-Zeile verwendet wurde.



-- Übernahme der Vorlage für die IS-Zeile aus GEN\_CFG.IS abhängig von der angegebenen IZ-Treiber-Id. und dem Datenpunkt-Typ mit Anpassung der Datenpunkt-Adresse, der IZ-Treiber-Id. und der IZ-Treiber-Subid. Ist ab Spalte 29 für den IZ-spezifischen Teil die Zeichenfolge ">>>" (ohne ") angegeben, so wird dieser durch die Zeichenfolge ab Spalte 82 der aus der FND-Referenz-Datei gelesenen DP-Zeile ersetzt. Auf diese Weise können IZ-spezifische Angaben zum Datenpunkt (z.B. Zuordnung der Merkerbyte-Adresse) aus der FND-Referenz-Datei übernommen werden.

- Erzeugen der DP-, ES- und IS-Zeilen der internen Datenpunkte für die Verbindungs-Kontrolle und die IZ-Status

Ist als Aufruf-Parameter der Anfüge-Modus mittels "/a" angegeben, so wird an die vorhandene Datei GEN\_CFG.CFG angefügt, wobei mit der Übernahme der Vorlage-Zeile aus GEN\_CFG.IZ begonnen wird. Die vorherigen Bearbeitungsschritte entfallen.

Ab Version 1.42d wird geprüft, ob ein Unterverzeichnis GEN\_CFG vorlag. Ggf. wird GEN\_CFG.CFG nun als GAK\_oooo.CFG in das Unterverzeichnis CFG kopiert, wobei oooo der beim Aufruf angegebenen Objekt-Kennung entspricht.

Nachfolgend müssen fehlende Zuordnungen, Klartexte, Parameter usw. mittels eines geeigneten Editor-Programmes manuell bearbeitet bzw. ergänzt werden und die so erzeugte Konfigurationsdatei für den GA-Knoten als GAK.CFG installiert werden.

**Hinweis:** Das Programm führt keine Plausibilitäts- oder Syntax-Prüfungen durch, d.h. Fehler werden erst beim Start des GA-Knotens mit der erzeugten Konfigurationsdatei erkannt.

Bei der Aktualisierung wird die Datei GEN\_CFG.CFG wie folgt erzeugt:

- Übernahme aller Zeilen aus der Konfigurationsdatei, soweit diese nicht mit "ES " beginnen
- Für DP-Zeilen wird die zugehörige ES-Zeile wie bei der Erzeugung gebildet (s.oben)

Um eine einmalige Pflege der ESN-Dateien unter LINUX zu ermöglichen, können die dort gespeicherten ESN-Dateien mittels des Programmes ESN2DOS in das DOS-Format - 8-stellig Unterverzeichnis, Rest Dateiname - konvertiert werden.

Der Aufruf erfolgt unter LINUX im Arbeitsverzeichnis von GEN\_CFG durch :

ESN2DOS [Schalter für Test]

Im aktuellen Arbeitsverzeichnis müssen folgende Dateien vorhanden sein (unter LINUX in Kleinbuchstaben):

- GEN\_CFG.ES (enthält je Datenpunkt-Typ die Vorlage für die ES-Zeile und optional die Beschreibung zum Aufbau eines bis zu 16-stelligen Schlüssels "sss", mit dem ES-Zeilen aus spezifischen Dateien geladen werden können )
- sss.ESn (enthält für den Datenpunkt-Typ "n" die Vorlage für die spezifische ES-Zeile für Datenpunkte gemäß dem Schlüssel "sss", vgl. Datenkatalog. Der Dateiname kann Platzhalter - z.B. "\_" enthalten.)

Die bei der Konvertierung erzeugten ESn-Unterverzeichnisse und -Dateien werden im Unterverzeichnis ESN2DOS gespeichert.

### **6.1.1. Datenkatalog**

Im aktuellen Arbeitsverzeichnis werden temporär die oben genannten Dateien analog zu den zugehörigen Zeilen in der Konfigurationsdatei angelegt bzw. vom Programmlauf erzeugt. Die Datei GEN\_CFG.ES hat dabei folgenden erweiterten Aufbau:

**ES ...** Spalte 3 ist ein Leerzeichen; Zeile 1 bis 5 entsprechen den ES-Zeilen für die Datenpunkttypen 1 bis 5.

Ab Version 1.2 wird alternativ folgendes Format unterstützt:

**ESn...** Spalte 3 ist eine Ziffer aus dem Bereich 0 bis 5 und ordnet die jeweilige ES-Zeile den Datenpunkttypen 0 bis 5 zu.

**SSS s1 ... s16** Schlüsseldefinition über die Spaltennummern "s1" bis maximal "s16" der Datenpunktzeile aus der Referenzdatei, aus deren Inhalt der Dateiname "sss.ESn" gebildet wird. Besteht "sss" aus mehr als 8 Zeichen, so entsprechen die ersten 8 Zeichen unter DOS dem Namen eines Unterverzeichnisses, die restlichen Zeichen sind der Name der ESn-Datei in diesem Verzeichnis. Unter LINUX entfällt das Unterverzeichnis, d.h. bis zu 16 Zeichen bilden den Dateinamen.

**Hinweis:** Die Reihenfolge der SSS- und ESn-Zeilen ist beliebig.

## 6.2. Rückführung der CFG-Datei (CFG2REF)

Dieses Hilfsprogramm dient der Rückführung einer Konfigurationsdatei (z.B. GAK.CFG) in die FND-Referenz-Datei CFG2REF.REF.

Der Aufruf erfolgt durch :

```
CFG2REF <Name der Konfigurationsdatei> [ Schalter Übernahme der IZ-Parameter] [ Schalter für Test-Fkt.]
```

Die Datei CFG\_REF.REF wie folgt erzeugt:

- Übernahme der 1. Zeile aus der Konfigurationsdatei
- Übernahme aller DATENPUNKT-Zeilen
- Wurde beim Aufruf der Parameter "/i" angegeben, so werden für jede DATENPUNKT-Zeile die IZ-Parameter aus der zugehörigen IZ-STRUKTUR-Zeile ab Spalte 29 in der Spalte 82 angefügt.

**Hinweis:** Das Programm führt keine Plausibilitäts- oder Syntax-Prüfungen durch, d.h. Fehler werden erst beim Start des GA-Knotens mit der erzeugten Konfigurationsdatei erkannt.

## 6.3. Dialogorientiertes Testen der IZ 's (IZ TEST)

Analog zum GA-Knoten werden anhand der Konfigurationsdatei GAK.CFG die IZ-Verbindungen aufgebaut und die zugehörigen Datenpunkte eingerichtet. Die zu sendenden CMD-Datagramme lassen sich per Dialog erzeugen. Die empfangenen RSP- und USM-Datagramme werden am Bildschirm angezeigt.

Der Aufruf erfolgt analog zum GA-Knoten im Arbeitsverzeichnis durch :

IZ\_TEST [ Schalter für Test-Funktionen ]

**Hinweis:** Das Programm IZ\_TEST enthält dieselben IZ-Treiber wie der GA-Knoten (IZ01, IZ02 und IZ03).

Sollen herstellerspezifische IZ's getestet werden, so muß deren IZ-Treiber (z.B. IZ04) in einer spezifischen EXE.Datei eingebunden werden.

#### **6.4. LOG-Dateien in Klartext (FND LOG)**

Der Inhalt der beim Programmlauf erzeugten Log-Dateien NORMAL.LOG, REJECT.LOG und ERROR.LOG wird mittels des Programmes FND\_LOG pro FND-Datagramm am Bildschirm in Klartext angezeigt. Optional kann die Ausgabe mittels des Test-Schalters "/Td<Dateiname>" auf die Datei <Dateiname> umgelenkt werden.

Wird der optionale Schalter "/rnnn" (n = Satzlänge in Bytes) angegeben, so können auch die FND-Datagramme anderer binärer Dateien mit fester Satzlänge (z.B. WSL-Dateien) angezeigt werden.

Der Aufruf erfolgt durch :

FND\_LOG <Name der LOG-Datei> [Schalter für feste Satzlänge] [ Schalter für Test-Funktionen ]

#### **6.5. Aktivieren des UART-Fifo (COM FIFO)**

Nach dem Systemstart unter DOS befinden sich die Bausteine für die seriellen Schnittstellen (UART) im 8250-Modus, d.h. sie sind kompatibel zu den ehemaligen XT-Rechnern. In diesem Modus kann nur ein empfangenes Zeichen zwischengespeichert werden, so dass bei einer Übertragungsrates von 9600 Baud das Datenregister innerhalb von 1 ms ausgelesen werden muß. Insbesondere bei Ausnutzung des Protected-Mode durch DOS-Extender (z.B. DPMS) ist ein Echtzeitbetrieb, der dieser Anforderung genügt, nicht mehr sichergestellt. Neuere Rechner verfügen über UART's mit einem integriertem Datenpuffer (FIFO) von 16 oder mehr Zeichen (z.B. 16550 ff.). Mittels des Programmes COM\_FIFO kann der Datenpuffer aktiviert werden.

**Hinweis: Dieses Programm steht nur unter DOS zur Verfügung !**

Der Aufruf erfolgt durch :

COM\_FIFO <Nummer der COM-Schnittstelle (1,...,4)>

**Hinweis:** Die Datenpuffer werden bei Programmende NICHT deaktiviert, so dass es bei nachfolgendem Aufruf anderer Programme, die die serielle Schnittstelle nutzen zu Störungen bzw. einem Systemabsturz kommen kann. Es wird empfohlen, im Zweifelsfall den Rechner neu zu starten.

## **6.6. Fernwartungs-Dienste (FWD)**

GA-Knoten als auch ISDN/X.25-Router lassen sich durch die LZH stoppen. Im Fehlerfall oder nach Stop durch die LZH soll i.a. eine Fernwartung von der LZH aus erfolgen und im Anschluß daran der GA-Knoten bzw. ISDN/X.25-Router mit den ggf. modifizierten Parametern (inkl. CONFIG.SYS und AUTOEXEC.BAT) neu gestartet werden.

Das Programm für die Fernwartungs-Dienste FWD wurde aus dem Programm ACOPY2 der Fa. EICON Technology Diehl anhand der bereitgestellten Quellen abgeleitet. Dabei wurden folgende Eigenschaften ergänzt:

- Prüfung der Anrufer-Berechtigung am FWD-Server analog zum GA-Knoten (vgl. ANRUFER.CHK)
- Erweiterung um das Kommando STOP zum Beenden des FWD-Servers durch den FWD-Client

Im Fehlerfall oder nach Stop durch die LZH wird vom GA-Knoten als auch ISDN/X.25-Router ein ERRORLEVEL > 0 geliefert, so dass in dem aufrufenden Batch (z.B. GAKSTART.BAT) abhängig vom ERRORLEVEL verzweigt werden kann. Der aufrufende Batch startet nun auf dem GA-Knoten bzw. ISDN/X.25-Router den FWD-Server. Der Benutzer kann nun den FWD-Server mittels des FWD-Clients anrufen und, falls er als Anrufer berechtigt ist, die Fernwartung mittels der verfügbaren Kommandos durchführen. Anschließend wird der FWD-Server durch Eingabe des STOP-Kommandos am FWD-Client beendet. Der aufrufende Batch veranlaßt nun den Neustart (z.B. Programm REBOOT).

Der Aufruf erfolgt durch :

FWD <Kommandos und Optionen gemäß ACOPY2 zzgl. Kommando STOP>

Wird kein oder ein unzulässiger Aufrufparameter angegeben, so erfolgt eine Fehlermeldung und die Beschreibung der zulässigen Aufrufparameter.

## **6.7. Fernsicherung (FWD\_SICH)**

Um Änderungen an der Konfiguration (z.B. CONFIG.SYS, AUTOEXEC.BAT, GAK.CFG, Schaltprogramme, ...) z.B. nach einer Fernwartung sichern zu können, ist eine Fernsicherung auf einen FWD-Server erforderlich. Diese kann durch Aufruf der Batchdatei FWD\_SICH ausgeführt werden.

Der Aufruf erfolgt durch :

```
FWD_SICH <Kennung des GA-Knotens> <Rufnummer des FWD-Servers> [<Name der LOG-Datei>]
```

Der Aufruf startet das FWD-Programm und erzeugt unterhalb des aktuellen Arbeitsverzeichnisses auf dem FWD-Server mit der angegebenen Rufnummer ein Unterverzeichnis mit der Kennung des GA-Knotens. In dieses werden die CONFIG.SYS und alle BAT-Dateien aus den Root-Verzeichnis (C:\) des aufrufenden GA-Knotens kopiert und die unter dem aktuellen Arbeitsverzeichnis (i.a. GAK) bestehende Unterverzeichnis-Struktur mit den parametrierenden Dateien kopiert. Ist eine LOG-Datei (z.B. FWD\_SICH.LOG) angegeben, so wird diese abschließend auch noch kopiert, so dass am FWD-Server das Protokoll der Fernsicherung abgefragt werden kann.

## **6.8. Warm- oder Kaltstart des Systems (REBOOT)**

Um Änderungen an der Systemkonfiguration unter DOS (z.B. CONFIG.SYS, AUTOEXEC.BAT,...) aktivieren zu können, ist ein Neustart des Systems erforderlich. Dieser kann durch Aufruf des Programmes REBOOT wahlweise als Warm- oder Kaltstart (durch Angabe der Option "K") ausgeführt werden.

**Hinweis: Dieses Programm steht nur unter DOS zur Verfügung !**

Der Aufruf erfolgt durch :

```
REBOOT [K]
```

### **6.9. Zeitabgleich und Stoppen des Rechners (TSYNC)**

Unter LINUX kann der Zeitabgleich nur von einem Prozess mit privilegierten Rechten ausgeführt werden. Zu diesem Zweck wird das Programm TSYNC beim System-Start (z.B. in /etc/init.d/rc) als Dämon-Prozess aufgerufen. Die zu setzende Zeit wird mittels InterProcessCommunication (IPC) über die Pipe "/tmp/TSYNC\_FIFO" übertragen.

Im aktuellen Verzeichnis werden durchgeführte Aktionen in der Datei TSYNC.LOG protokolliert.

Ab Version 1.3i wird der Kommunikations-Prozess (KP) inkl. des Betriebssystems/Rechners nach ausgewiesenen Fehlern gestoppt. Grund ist das Verhalten der TK-Anlage (TENOVIS) im Technischen Rathaus (BREF) der Landeshauptstadt München. Diese stürzt seit einem Update ab, da abgewiesene Anrufe protokolliert werden und der Puffer überläuft.

Dazu wird analog zum Zeitabgleich ein System-Aufruf an den Prozess TSYNC gesendet, der diesen dann privilegiert ausführt.

**Hinweis: Dieses Programm steht nur unter LINUX zur Verfügung und beruht auf der Vorlage von Dr. Oeser !**

Der Aufruf erfolgt als privilegierter Benutzer (z.B. root) durch :

```
tsync.exe [ Schalter für Test-Funktionen ]
```

### **6.10. Löschen der IMS-Dateien (CLR IMS)**

Ab Version 1.3k wurde die Zutrittskontrolle um das Hilfsprogramm CLR\_IMS ergänzt. Dieses kann z.B. mittels eines CRON-Jobs zyklisch gestartet werden und dient zum Löschen der IMS-Dateien unter folgenden Bedingungen :

- 1.) die in der IMS-Datei vermerkte Raumzone "RZ\_Kenn" kann wegen abgelaufener Gültigkeit nicht mehr betreten bzw. verlassen werden

2.) die in der IMS-Datei vermerkte Raumzone "RZ\_Kenn" kann wegen nicht mehr bestehender Zutrittsberechtigung gemäß IM-Datei nicht mehr betreten bzw. verlassen werden

Der Aufruf erfolgt im Arbeits-Verzeichnis des GA-Knotens durch :

```
clr_ims.exe [ Schalter für Test-Funktionen ]
```

Jeder Programmlauf wird in der Datei CLR\_IMS.LOG protokolliert. Dabei werden Zeitpunkt und Grund der Löschung für jede IMS-Dateien angegeben.



<b>1. Einleitung</b>	<b>1</b>
<b>2. Änderungen und Erweiterungen</b>	<b>2</b>
2.1. Anschluss der DEMA.....	2
2.2. Sortierung der ESn-Dateien für GEN_CFG und Vorbelegungen.....	3
2.3. Konvertierung der ESn-Dateien für GEN_CFG unter DOS.....	3
2.4. Vorrangige Belegung von Anlagenbildern.....	3
2.5. Zyklische Überwachung von Melde- und Schaltpunkten.....	4
2.6. Vorrangige Übertragung zwischengespeicherter Werte.....	4
2.7. Erweiterung auf max. 128 IZ-Treiber (GAK_XXL.EXE).....	4
2.8. Unterstützung der HRP- und MRP-Regler der Fa. Kieback&Peter .....	5
2.9. Mehrstufige Melde-/Schaltpunkte für MILES-Arbeitsstationen der Fa. Messner.....	5
2.10. Unterstützung der Systeme IQ3 und ANC der Fa. Trend.....	5
2.11. Bedingte Umlenkungen.....	5
2.12. Aktualisierung von ES-Zeilen in der Konfigurationsdatei.....	6
2.13. Plausibilitätsprüfung beim IZ-Zeitabgleich .....	6
2.14. Zyklische Überwachung und Übernahme der IM-Dateien.....	6
2.15. Zutrittskontrolle mit PIN-Eingabe .....	7
2.16. Optimierungen Sauter EY 3600 .....	7
2.17. Wartezeit beim Zeitabgleich .....	7
2.18. Abkündigung DOS-Version.....	7
2.19. TSYNC: Stoppen des Rechners nach ausgewiesenen Fehlern.....	8

2.20. Fremd-Generierung mit GEN_CFG.....	8
2.21. IZ-Treiber für BACnet/IP (BACnet-Client) .....	8
2.22. IZ-Treiber für AS511 .....	8
2.23. LZ-Treiber mit FND/LON-Kernel .....	9
2.24. SAIA-Bus/GfR-Treiber mit CMSPAR-Flag .....	9
2.25. GfR-Treiber mit SBUS+ und automatische REF-Generierung .....	9
2.26. P90-Treiber mit Datums/Zeit-Formaten .....	9
2.27. Automatisches Löschen von IMS-Dateien .....	10
2.28. LZ-Treiber für BACnet/IP (BACnet-Server).....	10
2.29. IZ-Treiber für ILC 350, FC 350 o.ä. über PCOS (Fa. PHOENIX Contact).....	10
2.30. IZ-Treiber für MODBUS/IP und MODBUS/RTU .....	11
2.31. GA-Knoten auf Basis des VPN-Routers GenuBOX (GenuGAK, Fa. GeNUA).....	11
2.32. Swapped Formate für MODBUS/IP und MODBUS/RTU .....	11
2.33. Start von GEN_CFG im Arbeitsverzeichnis und gak_OOOO.cfg .....	11
2.34. Kommando-Datei GAK.CMD.....	11
2.35. GfR-Treiber mit SBUS+ und Zugriff auf RealTimeClock (RTC).....	12
<b>3. DV-Architektur</b> .....	<b>12</b>
3.1. Software-Einheiten.....	12
3.2. Hardware-Einheiten.....	14
3.3. Systeminterne Schnittstellen.....	16
3.4. Systemexterne Schnittstellen.....	17
3.5. Übersicht .....	17

<b>4.</b>	<b>Software-Architektur</b>	<b>18</b>
<b>4.1.</b>	<b>Entwicklungsumgebung</b>	<b>18</b>
4.1.1.	Generierung aus den Programm-Quellen	19
4.1.2.	Einbinden spezifischer IZ-Treiber unter DOS	20
4.1.3.	Einbinden spezifischer IZ-Treiber unter LINUX	21
4.1.4.	Einbinden spezifischer LZ-Treiber unter DOS	23
4.1.5.	Einbinden spezifischer LZ-Treiber unter LINUX	25
4.1.6.	GAK_XXL.EXE unter LINUX	25
<b>4.2.</b>	<b>Einsatzumgebung</b>	<b>28</b>
<b>4.3.</b>	<b>Prozesse</b>	<b>32</b>
<b>4.4.</b>	<b>Daten- und Funktionssicherung</b>	<b>33</b>
<b>5.</b>	<b>Software-Entwurf</b>	<b>35</b>
<b>5.1.</b>	<b>Design</b>	<b>35</b>
5.1.1.	Vergabe von Programm- und Modulnamen	35
5.1.2.	Vergabe von Funktions- und Variablennamen	37
5.1.3.	Definition der Systemzustände	38
5.1.4.	Globale Variablen	39
5.1.5.	Globale Konstanten	42
<b>5.2.</b>	<b>Basis-Module</b>	<b>43</b>
5.2.1.	Übersicht der verwendeten Module	43
5.2.2.	Modulbeschreibungen	46
<b>5.3.</b>	<b>Hauptprogramm des GA-Knoten (GAK)</b>	<b>52</b>
5.3.1.	Übersicht der verwendeten Module	52
5.3.2.	Liste mit Namen und Aufgaben	53
5.3.3.	Aufrufstruktur	53
5.3.4.	Modulbeschreibungen	56

5.3.5.	Druckausgaben, Meldungstexte .....	74
5.3.6.	Datenkatalog .....	75
5.3.7.	Definition der Datenübertragung .....	103
<b>5.4.</b>	<b>Inselzentralen (IZ und IZnn).....</b>	<b>104</b>
5.4.1.	Übersicht der verwendeten Module .....	104
5.4.2.	Liste mit Namen und Aufgaben.....	105
5.4.3.	Aufrufstruktur .....	107
5.4.4.	Modulbeschreibungen.....	111
5.4.5.	Druckausgaben, Meldungstexte .....	158
5.4.6.	Datenkatalog .....	158
5.4.7.	Definition der Datenübertragung .....	168
5.4.8.	IZnn-Treiber für Sondersysteme .....	170
<b>5.5.</b>	<b>Elektronischer Schlüssellersatz (ES).....</b>	<b>283</b>
5.5.1.	Funktionsbeschreibung .....	285
5.5.2.	PIN-Eingabe .....	286
5.5.3.	Status-/Modus-Anzeige .....	287
5.5.4.	Entpacken vom IM-Dateien.....	288
5.5.5.	Übersicht der verwendeten Module .....	289
5.5.6.	Liste mit Namen und Aufgaben.....	289
5.5.7.	Aufrufstruktur .....	290
5.5.8.	Modulbeschreibungen.....	291
5.5.9.	Druckausgaben, Meldungstexte .....	300
5.5.10.	Datenkatalog .....	300
5.5.11.	Definition der Datenübertragung .....	307
<b>5.6.</b>	<b>Schaltprogramme (SP).....</b>	<b>308</b>
5.6.1.	Übersicht der verwendeten Module .....	308
5.6.2.	Liste mit Namen und Aufgaben.....	308
5.6.3.	Aufrufstruktur .....	310

5.6.4.	Modulbeschreibungen.....	313
5.6.5.	Druckausgaben, Meldungstexte .....	352
5.6.6.	Datenkatalog.....	352
5.6.7.	Definition der Datenübertragung .....	364
<b>5.7.</b>	<b>Leitzentralen (LZ).....</b>	<b>364</b>
5.7.1.	Übersicht der verwendeten Module .....	365
5.7.2.	Liste mit Namen und Aufgaben.....	365
5.7.3.	Aufrufstruktur .....	367
5.7.4.	Modulbeschreibungen.....	370
5.7.5.	Druckausgaben, Meldungstexte .....	402
5.7.6.	Datenkatalog.....	402
5.7.7.	Definition der Datenübertragung .....	406
<b>5.8.</b>	<b>Hauptprogramm des Standard-Schnittstellen-Adapters (SSA).....</b>	<b>407</b>
5.8.1.	Übersicht der verwendeten Module .....	408
5.8.2.	Liste mit Namen und Aufgaben.....	408
5.8.3.	Aufrufstruktur .....	409
5.8.4.	Modulbeschreibungen.....	410
5.8.5.	Druckausgaben, Meldungstexte .....	423
5.8.6.	Datenkatalog.....	424
5.8.7.	Definition der Datenübertragung .....	425
<b>5.9.</b>	<b>FND 1.0 über V.24 (LZRS).....</b>	<b>425</b>
5.9.1.	Übersicht der verwendeten Module .....	426
5.9.2.	Liste mit Namen und Aufgaben.....	426
5.9.3.	Aufrufstruktur .....	426
5.9.4.	Modulbeschreibungen.....	428
5.9.5.	Druckausgaben, Meldungstexte .....	434
5.9.6.	Datenkatalog.....	434
5.9.7.	Definition der Datenübertragung .....	434

<b>5.10. 3964R/RK512 über V.24 (LZEY) .....</b>	<b>434</b>
5.10.1. Übersicht der verwendeten Module .....	435
5.10.2. Liste mit Namen und Aufgaben.....	435
5.10.3. Aufrufstruktur .....	435
5.10.4. Modulbeschreibungen.....	437
5.10.5. Druckausgaben, Meldungstexte .....	444
5.10.6. Datenkatalog.....	444
5.10.7. Definition der Datenübertragung .....	444
<b>5.11. LonTalk mit FND 1.0 über V.24 (LZLN) .....</b>	<b>444</b>
5.11.1. Übersicht der verwendeten Module .....	445
5.11.2. Liste mit Namen und Aufgaben.....	445
5.11.3. Aufrufstruktur .....	445
5.11.4. Modulbeschreibungen.....	447
5.11.5. Druckausgaben, Meldungstexte .....	457
5.11.6. Datenkatalog.....	457
5.11.7. Definition der Datenübertragung .....	457
<b>5.12. BACnet/IP (LZBN).....</b>	<b>461</b>
5.12.1. Übersicht der verwendeten Module .....	461
5.12.2. Liste mit Namen und Aufgaben.....	461
5.12.3. Aufrufstruktur .....	462
5.12.4. Modulbeschreibungen.....	464
5.12.5. Druckausgaben, Meldungstexte .....	518
5.12.6. Datenkatalog.....	518
5.12.7. Definition der Datenübertragung .....	519
<b>6. Hilfsprogramme .....</b>	<b>534</b>
<b>6.1. Generierung der CFG-Datei (GEN_CFG und ESN2DOS) .....</b>	<b>534</b>
6.1.1. Datenkatalog.....	538

<b>6.2.</b>	<b>Rückführung der CFG-Datei (CFG2REF) .....</b>	<b>539</b>
<b>6.3.</b>	<b>Dialogorientiertes Testen der IZ´s (IZ_TEST) .....</b>	<b>539</b>
<b>6.4.</b>	<b>LOG-Dateien in Klartext (FND_LOG) .....</b>	<b>540</b>
<b>6.5.</b>	<b>Aktivieren des UART-Fifo (COM_FIFO) .....</b>	<b>540</b>
<b>6.6.</b>	<b>Fernwartungs-Dienste (FWD) .....</b>	<b>541</b>
<b>6.7.</b>	<b>Fernsicherung (FWD_SICH) .....</b>	<b>542</b>
<b>6.8.</b>	<b>Warm- oder Kaltstart des Systems (REBOOT) .....</b>	<b>542</b>
<b>6.9.</b>	<b>Zeitabgleich und Stoppen des Rechners (TSYNC) .....</b>	<b>543</b>
<b>6.10.</b>	<b>Löschen der IMS-Dateien (CLR_IMS) .....</b>	<b>543</b>

**Systemspezifikation  
für  
GA-Knoten,  
BACnet-Server  
und  
Standard-Schnittstellen-Adapter  
in der Version 1.44**

erstellt im Auftrag der

**Landeshauptstadt München  
Baureferat Hochbau 6  
Friedenstraße 40  
D-81660 München  
u.a.**

durch

**Daten - Technik Fries  
Hochstraße 25  
D-85221 Dachau**

**Autor : Dipl.-Informatiker (Univ.) Wolfgang Fries**

**Stand : 16. Juni 2010**