

Flächendeckende Anwendung

von

**firmenneutralen Datenübertragungssystemen
in öffentlichen Gebäuden und Liegenschaften**

gemäß

FND - Spezifikation

(DIN V 32735)

(EN V 1805/2)

bei der

Landeshauptstadt München

Stand: 18. November 2008

1	<u>ANWENDUNG DES FND BEI DER LANDESHAUPTSTADT MÜNCHEN.....</u>	7
1.1	Einführung	7
1.2	Wie arbeitet FND ?	9
1.3	Was ist das MEMS ?	10
1.4	Wie arbeitet ein GA-Knoten ?	11
1.5	Wie arbeitet die LZH ?	14
2	<u>FND-DETAILFESTLEGUNGEN</u>	15
2.1	Einführung	15
2.2	Definitionen und Bezeichnungen	15
2.3	Beschreibung der Datenpunkte	16
2.3.1	Meldepunkt	18
2.3.2	Schaltpunkt (mit Rückmeldung)	23
2.3.3	Messpunkt	27
2.3.4	Stellpunkt	34
2.3.5	Zählpunkt	38
2.4	FND – APDU-Format	44
2.5	Operationsbeschreibung.....	49
2.6	Beschreibung des DP-Status	49
2.7	Sperren von Ereignismeldungen.....	49
2.8	Zugriffskontrolle.....	49
2.9	Segmentierung.....	50
3	<u>ERGÄNZUNGEN ZUR FND-SPEZIFIKATION.....</u>	51
3.1	Zu (1): Spontanmeldungen (te-Bit)	51
3.2	Zu (2): DP_Typ 7 (Karteipunkt).....	53
3.2.1	Kartei-Koordinatensystem	53
3.2.2	Karteipunkt-Tabellen	56
3.2.3	Zellenstruktur.....	63
3.2.4	Kartei-Informationsebenen	68
3.2.5	Anmerkung zur Segmentierung:	70
3.3	Zu (3): Physikalische Größe „Zeitpunkt“	78
4	<u>VERWENDUNG DER KARTEIPUNKTE IM GA-KNOTEN.....</u>	81
4.1	Blockformat	81
4.2	Parametrierung von Datenpunkten.....	82
4.2.1	Aktivierungszeit der Datenpunkt-Parameter.....	83
4.2.2	Datenpunkt-Parameter für Melde- und Schaltpunkte	83

4.2.3 FND-Konvention für Parametriersatz von Melde- und Schaltpunkten	84
4.2.4 Datenpunkt-Parameter für Meß-, Stell- und Zählpunkte	85
4.2.5 FND-Konventionen für den Parametriersatz von Meß-, Stell- und Zählpunkten	85
4.3 Verbindungskontrolle	87
4.3.1 Aktivierungszeit des Verbindungs-Profiles	87
4.3.2 Verbindungs-Profil	87
4.3.3 FND-Konventionen für das Verbindungs-Profil.....	88
4.4 Schaltprogramm.....	91
4.4.1 Aktivierungszeit des Schaltprogrammes.....	92
4.4.2 Parametrierung des Schaltprogrammes (Schaltprogramm-Parameter)	92
4.4.3 FND-Konvention für die Schaltprogramm-Parameter	92
4.4.4 Parametrierung des Zeit-Schaltprogrammes	93
4.4.5 FND-Konvention für das Zeit-Schaltprogramm	95
4.4.6 Parametrierung, bzw. Syntax des Ereignis-Schaltprogrammes	96
4.4.7 FND-Konvention für das Ereignis-Schaltprogramm	105
4.4.8 Parametrierung für den Schaltprogramm-Kalender	105
4.4.9 FND-Konvention für den Schaltprogramm-Kalender	105
4.5 Komplexere Abfragen.....	106
4.5.1 Aktivierungszeit der Istwert-Abfrage	107
4.5.2 Parametrierung der Istwert-Abfrage	107
4.5.3 FND-Konvention für den Abfragesatz.....	107
<u>A AUTHENTIFIZIERUNG.....</u>	109
<u>B PROZEDUR-ELEMENTE</u>	109
<u>C NETZWERK-INTERFACE</u>	109
<u>D SPEZIELLE TRANSFERPUNKTE</u>	109
<u>E ATTRIBUTE UND DIMENSIONEN.....</u>	110
<u>F FEHLERBEHANDLUNG</u>	117
F.1 REJECT-APDU-Format (Ablauffehler).....	118
F.2 ERROR-APDU-Format (syntaktische/semantische Fehler)	119
<u>G ÜBERSICHT DER DATENPUNKT-TABELLEN.....</u>	120
G.1 Meldepunkt	120
G.2 Schaltpunkt (mit Rückmeldung)	121
G.3 Messpunkt	122
G.4 Stellpunkt	123
G.5 Zählpunkt.....	124
<u>H ÜBERSICHT DER CODIERUNGEN IM FND-APDU</u>	125

I	BEISPIELE FÜR FND-FUNKTIONEN	126
I.1	Tageslicht- und zeitabhängige Außenbeleuchtung.....	126
I.2	3-stufige Beleuchtung mit Orientierungslicht.....	127
I.3	Einzelraum-Regelung	128
J	BEISPIELE FÜR GA-FUNKTIONEN	130
J.1	Zentral-AUS	130
J.2	Ferientaster.....	130
J.3	Sonderfunktionen.....	130
J.4	EM-Profile	131
K	GLT-RICHTLINIEN, AUSSCHREIBUNGEN	132
K.1	Elektro	132
K.1.1	Aufbau der Datenpunkt-Adressen	132
K.1.2	Informationslisten	132
K.1.3	Ausschreibungstexte.....	132
K.2	Heizung, Klima, Lüftung	132
K.2.1	Aufbau der Datenpunkt-Adressen	132
K.2.1	Informationslisten	132
K.2.3	Ausschreibungstexte.....	132
	LITERATURVERZEICHNIS	133

Vorwort

Das *firmenneutrale* **D**atenübertragungssystem (**FND**) stellt eine umfassende Spezifikation der Datenkommunikation für Systeme insbesondere der Gebäudeleittechnik (**GLT**) dar und ist Voraussetzung für eine gewerkeübergreifende Gebäudeautomation (**GA**).

Ausgangspunkt für die Entwicklung des FND waren die bestehenden Probleme und Abhängigkeiten bei der Kopplung von Systemen für das Messen/Steuern/Regeln (MSR) unterschiedlicher Hersteller und deren Zusammenfassung auf einer gemeinsamen Leitzentrale (**LZ**).

Bereits Mitte der 80er Jahre nahm sich der „Arbeitskreis Maschinen- und Elektrotechnik staatlicher und kommunaler Verwaltungen“ (AMEV) dieser Problematik an. Basierend auf der VDI-Richtlinie 3814 wurde 1988 die FND-Spezifikation in der Version 1.0 vorgelegt. Diese wurde am 8.1.1990 durch Anmerkungen und Erweiterungen präzisiert und ist in dieser Form in die DIN V 32735 und die EN V 1805/2 eingegangen.

1993 wurde bei der Landeshauptstadt München (LHM) eine komplette Neukonzeption der Leitzentrale Haustechnik (**LZH**) erforderlich. Die Gründe dafür waren dieselben, die zur Spezifikation des FND geführt hatten. Folglich wurde der FND auf seine Eignung geprüft und als zentraler Baustein in ein umfassendes Gesamtkonzept für die neue LZH integriert. Dabei wurden sowohl technische als auch organisatorische und administrative Belange (z.B. bei den Ausschreibungen) berücksichtigt.

Der Aufbau der neuen LZH wurde in das „Münchner Energie-Management-System“ (**MEMS**) eingebunden und wird vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) gefördert. Die dabei erstellten Programme liegen der Landeshauptstadt München in Quellform vor und werden interessierten Verwaltungen zur kostenfreien Nutzung zur Verfügung gestellt. Durch Verwendung von Standards und die Entwicklung eines integrierenden Rechnersystems (**GA-Knoten**) konnte die bleibende Unabhängigkeit erreicht werden.

Ergänzend zu den Programmen stehen unter www.fnd-forum.de auch die von der Landeshauptstadt München formulierten Ausschreibungstexte und Richtlinien für die Gebäude-Automation für die einzelnen Gewerke zur Verfügung. Diese werden vom Planer - i.a. ohne projektspezifische Anpassungen - in die Ausschreibung übernommen, so dass ein vom Planer bis zum Betreiber und Nutzer handhabbares Gesamtsystem entsteht.

Dieses Verfahren wird inzwischen von mehreren Verwaltungen genutzt, führt bei der Beschaffung von MSR-Anlagen zu Einsparungen von 30-50% gegenüber produktgebundenen Vergaben und sichert langfristig Unabhängigkeit, Wettbewerb als auch die getätigten Investitionen.

Ergänzende Produkte zur LZH und den GA-Knoten werden auch von Firmen angeboten.

Die gemachten Erfahrungen werden im Arbeitskreis Gebäudeautomation des AMEV kontinuierlich diskutiert, veröffentlicht und bei der Fortschreibung berücksichtigt.

Im Rahmen der Schulungen für die LZH zeigte sich, dass das Verständnis für die FND-Spezifikation im praktischen Einsatz einer einweisenden Schulung bedarf. Anhand der bei der Schulung gemachten Erfahrungen wird nachfolgend die FND-Spezifikation für die praktische Anwendung erläutert.

Wolfgang Fries, Dezember 2004

1 Anwendung des FND bei der Landeshauptstadt München

1.1 Einführung

Die Landeshauptstadt München betreibt seit Anfang der 80er Jahre eine Leitzentrale für die Haustechnik. Gemäß dem damaligen technischen Stand war diese erste LZH über ein herstellereinspezifisches Datenübertragungs-Protokoll mit den einzelnen Anlagen verbunden.

Aufgrund der mit der ersten LZH gemachten Erfahrungen wurde für die neue LZH ein firmenneutrales Protokoll gefordert. FND erfüllt dabei neben den technischen auch wesentliche organisatorische und administrative Anforderungen. Dadurch wird die Abhängigkeit von einem Hersteller bzw. Produkt vermieden, der Wettbewerb bleibt stets erhalten und die getätigten Investitionen bleiben langfristig gesichert.

Bei der Landeshauptstadt München werden die Anlagen der einzelnen Gewerke unabhängig voneinander ausgeschrieben. Allen gemeinsam ist jedoch die Forderung zur Lieferung eines zugehörigen FND-Schnittstellenadapters (**SSA**). Alle SSA werden vorort an einen Gebäude-Automatisierungs-Knoten (**GA-Knoten**) angeschlossen, der wiederum über das ISDN-Netz der Telekom bzw. das stadt-eigene IP- oder ISDN-Netz an den zentralen Rechner der LZH (LZH-Server) angeschlossen ist (vgl. nachfolgende Abbildung).

Der GA-Knoten übernimmt somit eine Doppelfunktion. Aus Sicht der einzelnen angeschlossenen MSR-Systeme verhält er sich wie eine LZ. Aus Sicht der LZH verhält er sich wie eine gemeinsame Inselzentrale (**IZ**) für alle angeschlossenen MSR-Systeme, d.h. alle Datenpunkte sind aus Sicht Benutzers FND-Datenpunkte, unabhängig von ihrer herstellereinspezifischen, technischen Realisierung.

Somit ist die bleibende Unabhängigkeit von den Herstellern und deren sich über die Jahre ändernden Produkten gewährleistet. Die einzelnen MSR-Systeme als auch die LZH können jederzeit unabhängig voneinander ausgetauscht werden (z.B. Heizungs-Sanierung).

Die einzelnen Benutzer erhalten mit der Bedienstation an ihrem Arbeitsplatz über ein lokales Netzwerk (LAN) Zugriff auf die Daten der LZH. Die Bedienstation ermöglicht über die FND-Datenpunkte auch den Zugriff auf die einzelnen Sensoren bzw. Aktoren.

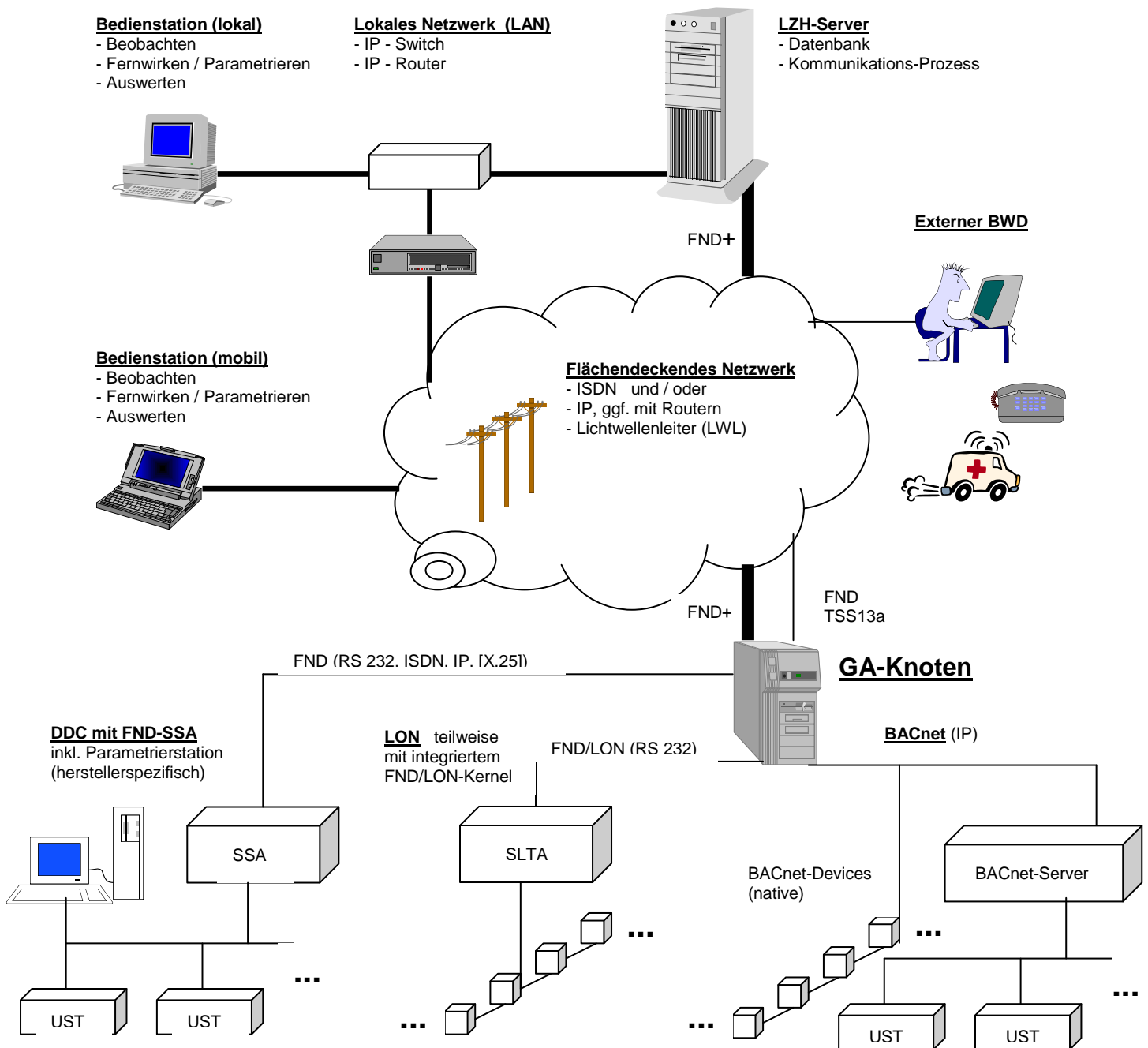
Für die eindeutige, gebäude- und gewerkeübergreifende Identifikation der einzelnen Datenpunkte hat die Landeshauptstadt München in ihren GLT-Richtlinien eine Syntax für die **Datenpunkt-Adressen** festgelegt (15-stellige, alphanumerische symbolische Adresse), aus der u.a. auch das zugeordnete Gewerk und die physikalische Größe erkenntlich ist.

Um übergeordnete Funktionen des GA-Knoten von der LZH aus nutzen und parametrieren zu können, wurden 1995 Ergänzungen zur FND-Spezifikation definiert (**FND+**). Diese ermöglichen die Übertragung:

- Zusätzlicher Ereignisse der einzelnen Datenpunkte (Triggering Event)
- im GA-Knoten zwischengespeicherten Werte (Blockformat)
- ergänzender Datenpunkt-Parameter z.B. Flimmerkontrolle, Zykluszeit, Hysterese, Synchronisations-Zeitpunkt
- zeit- und/oder ereignisabhängiger Schaltprogramme
- Schaltprogramm-Kalender
- Komplexerer Abfragen, z.B. zur Belegung von Anlagenbildern

HINWEIS: Diese Ergänzungen werden ausschließlich im GA-Knoten behandelt und betreffen somit nicht die MSR-Systeme.

Die nachfolgende Abbildung gibt einen Überblick über das Gesamtsystem. Für LON und BACnet ist der SSA als Software (FND-SSA/LIB) im GA-Knoten realisiert. Eine Aufstellung der am GA-Knoten abschließbaren MSR-Systeme finden Sie im Abschnitt 1.4.



Die verwendeten Abkürzungen haben folgende Bedeutung:

BWD	Bewachungsdienst
DDC	Direct Digital Control
FND	Firmenneutrales Datenübertragungssystem
GA	Gebäudeautomatisierung
LAN	Local Area Network
LON	Local Operating Network
LZH	Leitzentrale Haustechnik
SSA	Standardschnittstellenadapter
UST	Unterstation

Wesentliche Eigenschaften dieses Gesamtsystems sind:

- Einheitliche Bedienoberfläche für alle Benutzer, unabhängig von den installierten MSR-Systemen (Minimierung von Schulungskosten, Vertreterregelungen)
- Unabhängigkeit vom Standort des Benutzers und des bearbeiteten Objektes (flächendeckender Einsatz, mobile/externe Wartungstrupps)
- Herstellerunabhängigkeit für alle Komponenten des Gesamtsystems durch modularen Aufbau unter Verwendung von Standard-Schnittstellen für Hard- und Software (Investitionsschutz)
- Übergreifende, firmenneutrale zeit- und/oder ereignis-abhängige Verknüpfung von Datenpunkten (Zeit-/Ereignis-Schaltprogramme)
- Reduzierung der Datenmenge auf relevante Werte durch parametrierbare Flimmerkontrolle, Zykluszeit, Hysterese, Min-/Max-/Mittelwert-Bildung usw. (effektivere Auswertungen)
- Parametrierbares Übertragungsverhalten mit Zwischenspeicherung bzw. sofortige Übertragung zur LZH, einen BWD für Gefahr- oder Einbruchmeldungen für jeden einzelnen Datenpunkt (Einbindung externer LZ und BWD)
- Überwachung der angeschlossenen BTA und Eigenüberwachung des GA-Knotens (Sicherheitskonzept)

1.2 Wie arbeitet FND ?

FND basiert auf einer firmen-/produkt-neutralen, formalen, technischen „Sprache“ (Protokoll) zur Übertragung von Datenpunkt-Informationen und den zugehörigen Operationen.

Dieses Protokoll wird durch Übertragung spezifizierter Zeichenfolgen (FND-Telegramme) über eine Datenübertragungstrecke (DÜ-Strecke/Netz) realisiert. Dabei kommuniziert pro FND-Telegramm jeweils eine LZ mit jeweils einem Datenpunkt (DP; i.a. ein Sensor oder Aktor). Jeder DP wird über eine eindeutige, bis zu 16-stellige alphanumerische Datenpunkt-Adresse identifiziert (dp_id).

Die LZ sendet bei Bedarf Befehle (**Cmd**) zum Lesen oder Modifizieren von DP-Informationen und erhält i.a. eine Antwort (**Rsp**). Tritt am DP ein relevantes Ereignis auf (z.B. Zustands-/Wertänderung am Geber), so kann der DP auch spontan eine Meldung (**Usm**) an die LZ senden und erhält eine Bestätigung (**Ack**). Ereignismeldungen können bei Bedarf für jeden DP gesondert in 4 Prioritätsstufen gesperrt werden.

Je DP werden die Informationen in bis zu 3 Gruppen unterteilt. Neben den eigentlichen **Nutzdaten** (z.B. Zustand/Wert des Sensor bzw. Aktor) werden dem DP noch **Status-Informationen** (z.B. Geber-Störung, Bedienung vor Ort, temporäre Verriegelung,...) und **Parameter** (z.B. Grenzwerte) zugeordnet. Dadurch kann der Benut-

zer an der LZ erkennen, unter welchen Randbedingungen der Zustand/Wert interpretiert werden muß.

Meldet ein DP z.B. gleichzeitig einen erhöhten Wert mit Grenzwertverletzung und die Bedienung vor Ort, so kann davon ausgegangen werden, dass zu Testzwecken bei einer Wartung der Wert manuell eingestellt wurde.

Hinweis: FND bietet die Möglichkeit zur Zuordnung unter einer gemeinsamen Datenpunkt-Adresse und zur Übertragung der Informationen in einem gemeinsamen Telegramm. Die FND-Spezifikation legt aber nicht fest, ob und wie diese Informationen auf der Sensor-/Aktor-Ebene oder Einzelleiteebene realisiert werden müssen. Hier können durchaus datenpunkt- und herstellerspezifische, kostenrelevante Unterschiede auftreten (z.B. Geber-Störung für eine Meldung erfordert 4/20mA-Schleife).

In der FND-Spezifikation von 1988 wurde neben dem eigentlichen FND-Protokoll ein X.25-Netzwerk als DÜ-Strecke festgelegt. Diese Festlegung erwies sich später als zu aufwendig bei Installation und Betrieb, insbesondere, wenn nur eine feste Verbindung (z.B. Kabel) über kurze Entfernungen benötigt wurde. Neuere technische Entwicklungen bieten hier standardisierte, günstigere Alternativen (serielle Schnittstellen wie V.24/RS232, ISDN, IP-Netzwerke,...). Diese sind inzwischen auch in die FND-Spezifikation eingegangen, so dass nachfolgend ausschließlich das FND-Protokoll unabhängig von der verwendeten DÜ-Strecke betrachtet wird.

Beispiel: Auf diese Weise besteht die Möglichkeit, z.B. einen DP „Pumpe 1“ nach seinem aktuellen Zustand bzw. den DP „Kesseltemp.“ nach seinem aktuellen Wert zu „fragen“. Genauso kann der Benutzer einem DP „Flurlicht“ über die LZ „befehlen“: „Schalte EIN“.

1.3 Was ist das MEMS ?

Für ein effektives Energie-Management müssen die haustechnischen Daten der LZH mit anderen Datenbeständen der Landeshauptstadt München u.a. verknüpft werden.



Übersicht der Funktionsbausteine



Münchner-Energie-Management-System (MEMS)

16.04.1998

Der konsequente Einsatz des FND im GA-Knoten und der LZH schafft durch seinen firmenneutralen Ansatz erstmals die technischen Voraussetzungen für gewerke- und objektübergreifende Auswertungen. Erst dadurch können effektive Strategien für das flächendeckende, einheitliche Energie- und Gebäude-Management entwickelt, z.B. über Schaltprogramme realisiert und in ihrer Wirkung kontrolliert werden.

Erfolgreiche Strategien können in einfacher Weise auf die GA-Knoten anderer Liegenschaften übernommen werden.

1.4 Wie arbeitet ein GA-Knoten ?

Der GA-Knoten setzt sich aus folgenden Bestandteilen zusammen :

- ein handelsüblicher Personal Computer (PC; die Bauart Desktop, Mini-Tower, Notebook, Industrie-PC o.ä. richtet sich nach den räumlichen und technischen Anforderungen)
 - mit den notwendigen Schnittstellen zu den angeschlossenen Anlagen, dem IP- und/oder ISDN-Netzwerk
 - einem Disketten-Laufwerk
 - einer Festplatte
 - Betriebssystem LINUX
- und einem optionalen Ereignisdrucker

Nur für die Inbetriebnahme der Software „GA-Knoten“ und evtl. zur Fehlersuche werden Monitor und Tastatur benötigt. Die SSA werden über die seriellen Schnittstellen (V.24, RS232), ein IP- oder ISDN-Netz an den GA-Knoten angeschlossen.

Legen Hersteller ihre spezifischen Protokolle offen oder werden allgemein offengelegte Protokolle verwendet, so kann der SSA durch einen in den GA-Knoten integrierten Software-Treiber ersetzt werden. Dadurch entfallen die Kosten für Beschaffung, Inbetriebnahme und Wartung des SSA und die damit verbundenen Fehlerquellen.

Folgende standardisierte bzw. offengelegte Protokolle können direkt vom GA-Knoten firmenneutral umgesetzt werden :

- FND 1.0 (V.24,ISDN,IP-Netze; div. Hersteller)
- BACnet/IP (IP-Netze; diverse Hersteller)
- EIB (V.24; diverse Hersteller)
- LON (V.24; diverse Hersteller)

- ACCESS 3000 (V.24; z.B. Multitone; Personenruf)
- CC 600 (V.24; z.B. RAM)
- CNC+ (z.B. DDC111/250,IQ3) (V.24; z.B. TREND)
- DEKATEL (V.24; z.B. Viessmann)
- EY-2400 / EY-3600 (V.24; z.B. Sauter)
- INFINITY (V.24; z.B. ANDOVER)
- IPC (V.24; z.B. SE-Elektronic)
- KS 120 (z.B. A120) (V.24; z.B. Schneider Groupe)
- MILES (V.24; z.B. Messner)
- P90 (z.B. DDC3000, HRP, MRP) (V.24, z.B. Kieback & Peter)
- SAIA-Bus (z.B. PCD1, PCD2) (V.24; z.B. GfR, SAIA)
- Short-Message-Services (SMS) (ISDN; z.B. D1,D2,E+, Fax, eMail)
- Supramat DC97 (V.24; z.B. Fröling)
- SUCOM-A (z.B. PS306/316, PS4) (V.24; z.B. Moeller)
- PRV/EKL (V.24; z.B. Landis&Staefa)
- AS511 (z.B. S5 / S7) (V.24, z.B. SIEMENS)
- SBUS+ (V.24; z.B. GfR)
- PCOS (IP-Netze; z.B. PHOENIX Contact)
- MODBUS/IP (IP-Netze; div. Hersteller)
- MODBUS/RTU (V.24; div. Hersteller)

Über Systeme mit diesen Protokollen besteht häufig Zugriff auf weitere Protokolle wie z.B. INTERBUS-S, PROFIBUS, MODBUS usw.

Für folgende Protokolle kann die Vorlage für die Referenzdatei automatisiert erstellt werden :

- BACnet/IP (Online aus Controller)
- SBUS+ (Austausch-Datei; CEX-Format)
- PCOS (Online aus Controller)

Hinweis: Durch Verwendung des FND-Protokolls zum Anschluss der BTA als auch der LZ können GA-Knoten über das IP- oder ISDN-Netz über beliebig viele Ebenen kaskadiert werden. Auf diese Weise können auch die Datenpunkte in räumlich verteilten Objekten (z.B. Krankenhäuser) miteinander verknüpft werden. Ebenso kann ein GA-Knoten so zu einer einfachen Störmeldezentrale umfunktioniert werden

(Verknüpfung mit akustischem/optischem Signal, Quittierungstaster, Protokolldrucker)

Der GA-Knoten arbeitet nach dem Programmstart autark und führt die Erfassung, Vorverarbeitung, Zwischenspeicherung und Übertragung der FND-Daten an die LZH bzw. einen externen Bewachungsdienst (BWD) permanent aus.

Als Sonderlösung läßt sich der GA-Knoten auch an andere Leitzentralen anschließen. Folgende Protokolle stehen bereits zur Verfügung :

- FND 1.0 (V.24, ISDN, IP-Netze)
- FND 1.0 + Erweiterungen (ISDN, IP-Netze)
- BACnet/IP (IP-Netze; diverse Hersteller)
- LON (V.24; diverse Hersteller)
- TSS 13a (ISDN, z.B. Bewachungsdienste)
- 3964R / RK512 (V.24, z.B. Sauter EY 2400)

Zusätzlich erzeugt der GA-Knoten anhand von zeit- und/oder ereignisabhängigen Schaltprogrammen Befehle an die DP. Innerhalb eines GA-Knotens als auch eines gesamten, aus mehreren kaskadierten GA-Knoten bestehenden Systems können die Datenpunkte IZ-übergreifend miteinander verknüpft werden (virtuelle Verdrahtung).

Beispiel: Sind an einem GA-Knoten, wie oben abgebildet, eine MSR-Anlage und ein LON-System angeschlossen, so kann über einen Schalter an einem beliebigen LON-Knoten die MSR-Anlage auf Tag- bzw. Nachtbetrieb umgeschaltet werden. Aufgrund der Spontanmeldung des Schalters erzeugt der GA-Knoten den Schaltbefehl an die MSR.

Die firmenneutrale Programmierung ermöglicht dabei eine Übernahme bestehender Schaltprogramme auf die GA-Knoten in anderen Liegenschaften, wobei im wesentlichen nur die Datenpunkt-Adressen angepasst werden müssen.

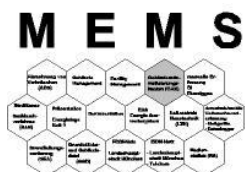
Neben den realen DP, denen ein Sensor/Aktor zugeordnet ist, können im GA-Knoten interne Datenpunkte für die Berechnung von vorverarbeiteten Werten (z.B. Min-/Max-/Mittel-Werte, Bildung einer Sammelstörung) oder die Steuerung des GA-Knotens durch die LZH vereinbart werden.

Der Zustand und die Werte der Sensoren bzw. Aktoren können anhand von ergänzenden Parametern (Flimmerkontrolle, Zykluszeit, Hysterese, ...), die für jeden Datenpunkt gesondert festgesetzt werden können, kontinuierlich erfasst und auf relevante Werte reduziert werden. Diese Parametrierung lässt sich zur Laufzeit mittels der Schaltprogramme oder von der LZH aus dynamisch verändern.

Für jeden DP kann gesondert parametrierbar werden, ob die relevanten Zustandänderungen bzw. Werte zwischengespeichert (z.B. Betriebswerte) oder sofort an die LZH (z.B. Störmeldungen) bzw. einen vorher festzulegenden BWD (z.B. Gefahr- und Einbruchmeldungen) weitergeleitet werden sollen. Dadurch ist keine ständige Verbindung zur LZH bzw. den BWD erforderlich, was den technischen Aufwand und die Kosten für die Datenübertragung erheblich reduziert, ohne wesentliche Einschränkungen mit sich zu bringen.

Der GA-Knoten überwacht die angeschlossenen Systeme und bildet erkannte Störungen auf interne FND-Datenpunkte ab, die spontan an die LZH gemeldet werden. In umgekehrter Richtung signalisiert der GA-Knoten seine Einsatzfähigkeit zyklisch,

so dass auch die angeschlossenen Systeme bei Ausfall des GA-Knoten eigenständig in den Minimalbetrieb umschalten können.



Landeshauptstadt
München

Funktionsbaustein: **Gebäude-Automatisierungs-Knoten (GAK)**



Münchner-Energie-Management-System (MEMS)

16.04.1998

1.5 Wie arbeitet die LZH ?

Alle Daten der haustechnischen Anlagen werden mit einer zentralen Datenbank auf dem LZH-Server verwaltet und gespeichert. Neben den datenpunktbezogenen Daten sind dieses auch objektbezogene Dokumente, Auswertungen und Stammdaten.

Die Bedienstationen der Benutzer werden über ein lokales Netzwerk (LAN) an den LZH-Server angeschlossen und greifen auf dessen Datenbestände zu.

Die Bedienoberfläche ist für alle gleich, unabhängig von Standort, Objekt und vorort installierten BTA. Zusätzlich können die aktuellen Ist-Werte abgefragt und in belebten Schemata dargestellt werden. Bei Bedarf können Schalt- und Stellbefehle an die einzelnen DP abgesetzt und die GA-Knoten parametrisiert werden.

Beispiel: Auf diese Weise besteht die Möglichkeit, z.B. einen DP „Pumpe 1“ nach seinem aktuellen Zustand bzw. den DP „Kesseltemp.“ nach seinem aktuellen Wert zu „fragen“. Genauso kann der Benutzer einem DP „Flurlicht“ über die LZ „befehlen“: „Schalte EIN“.

Die Verwendung von Standard-Schnittstellen ermöglicht eine Herstellerunabhängigkeit bei allen Systemkomponenten. Durch Einsatz von verfügbarer Standard-Hard- und Software werden die Beschaffungs-, Betriebs- und Schulungskosten gesenkt und bereits getätigte Investitionen langfristig gesichert.



Funktionsbaustein: **Leitzentrale Haustechnik (LZH)**



Münchner-Energie-Management-System (MEMS)

16.04.1998

2 FND-Detailfestlegungen

2.1 Einführung

Der „Verein deutscher Ingenieure“ (VDI) definiert im Blatt 1 seiner Richtlinie VDI 3814 die Strukturen, Begriffe und Funktionen der GLT. Die im FND definierten Datenpunkt-Typen basieren auf den in dieser Richtlinie definierten 5 Grundfunktionen :

- Melden
- Schalten
- Messen
- Stellen
- Zählen

Hinweis: Die Anmerkungen und Ergänzungen zur FND-Spezifikation vom 8.1.1990 wurden in dieses Dokument eingearbeitet.

2.2 Definitionen und Bezeichnungsweisen

Hinweis: Nachfolgend werden Kodierungen der FND-Spezifikationen überarbeitet, um ein besseres Verständnis zu erreichen.

2.3 Beschreibung der Datenpunkte

Bei der Spezifikation des FND wurde jeder Grundfunktion gemäß VDI3814 ein Datenpunkt-Typ zugeordnet:

- Meldepunkt
- Schalterpunkt (mit Rückmeldung)
- Messpunkt
- Stellpunkt
- Zählpunkt

zuzüglich wurden definiert

- Sammeladresspunkt.
- Transferpunkt,

die keinen Datenpunkt im Sinne der oben genannten Kategorien darstellt, sondern für projektspezifische Zwecke (wie etwa die Zusammenfassung bzw. die transparente Datenübertragung innerhalb eines homogenen Systems) verwendet wird.

In den FND-Telegrammen ist jedem Datenpunkt-Typ ein Wert im Feld „dp_type“ wie folgt zugeordnet:

dp_type	Datenpunkt-Typ
0	Transferpunkt
1	Meldepunkt
2	Schalterpunkt (mit Rückmeldung)
3	Meßpunkt
4	Stellpunkt
5	Zählpunkt
6	Sammeladresspunkt

Hinweis: Die LHM verwendet die Transferpunkte für den transparenten Datentransfer nur in Ausnahmefällen, da sonst ein erheblicher Aufwand für die projektspezifische Planung, Inbetriebnahme, Wartung usw. erforderlich ist und damit eine erhöhte Fehlerwahrscheinlichkeit gegeben ist. Projektunabhängig ist nur der Adresspunkt für Fehlermeldungen (Fault-Manager) und wird zur Kennzeichnung aufgetretener syntaktischer und semantischer Fehler innerhalb von FND-Telegrammen verwendet. Der Authentifizierungspunkt und der Sammeladresspunkt werden nicht verwendet. Nachfolgend wurde auf die spezifischen Spezifikationen für Transfer- und Sammeladresspunkte nicht eingegangen. Bei Bedarf sind diese der FND-Spezifikation zu entnehmen.

Bei der Beschreibung der Datenpunkte werden unter einer Datenpunkt-Adresse (dp_id) mehrere Einzelinformationen in Datenpunkt-Tabellen zusammengefaßt, die für jeden Datenpunkt-Typ spezifisch aufgebaut sind.

DP-Tabelle	Informations-Inhalt	Funktionen
# 0	Datenpunkt- Status	Anzeige von Ereignissen/Zuständen Sperrungen/Freigabe von Ereignismeldungen
# 1	DP-spezifische Nutzdaten (Schalterstellung, Meß-/Stellwert etc.)	Lesen Modifizieren Melden
# 2	(Konfigurations-) Parameter	Parametrieren

Hinweis: Dieser strikte formale Ansatz von 1988 schafft die Voraussetzung für eine strukturierte DV-technische Verarbeitung der FND-Telegramme, die bereits eine Form der **Objekt-Orientierung** (inkl. Vererbung) darstellt. Gegenüber text-orientierten Protokollen verschlechtert sich zwar die intuitive Lesbarkeit für den Menschen, jedoch wird der Programmier- und Testaufwand erheblich reduziert, was eine minimierte Fehlerrate zur Folge hat. Nur unter solchen Voraussetzungen lässt sich auch ein überschaubarer Konformitätstest definieren und durchführen, wie er in Teil 5 der FND-Spezifikation festgelegt wurde.

Als Bestandteil des „Operations-Codes“ ist die Nummer der DP-Tabelle (tab_id) im FND-Telegramm wie folgt vergeben:

tab_id	DP – Tabelle
0	DP_# 0
1	DP_# 1
2	DP_# 2
3	DP_# 1.DP_#2

Auf diesen DP-Tabellen können stets lesende und ggf. modifizierende Zugriffe durchgeführt werden. Innerhalb des FND-Telegrammes werden diese anhand der Funktionskennung (fct_id) wie folgt kodiert :

fct_id	Funktion
1	Lesen
2	Modifizieren (Schreiben)

Hinweis: Modifizierbare Parameter in einer Datenpunkt-Tabelle sind in den nachfolgenden Detailbeschreibungen mit „P_#1“, „P_#2“, „P_#3“ oder „P_#4“ gekennzeichnet.

(Für weitergehende Ausführungen bzgl. des Operations-Codes wird auf die Abschnitte 2.5.1 und 2.5.2, „Befehle“ und „Meldungen“ in der FND-Spezifikation verwiesen).

Eine detaillierte Beschreibung der oben genannten 5 Datenpunkt-Typen wird in den nachfolgenden Abschnitten gegeben, deren Nummer aus Konsistenzgründen mit dem entsprechenden dp_type übereinstimmt.

2.3.1 Meldepunkt Charakterisierung:

Einfach-Meldepunkte stellen binäre (boolesche) Größen dar, welche bei physikalischen Gebern i.a. den Zustand elektrischer Kontakte („offen“/ „geschlossen“) bei logischen (virtuellen) Gebern etwa den Aktivitätszustand eines Prozesses widerspiegeln.

Mehrstufigen Betriebsmeldungen sind bei physikalischen Gebern i.a. die Einzelkontakte eines Stufenschalters, bei logischen (virtuellen) Datenpunkten etwa Ablaufvarianten (Modi) von Zeit- oder Ereignisprogrammen zugeordnet.

Datenpunkt-Tabellen:

DP_# 0			DP_# 1	
mask	P_# 1		atb_id	
info_event			actual	

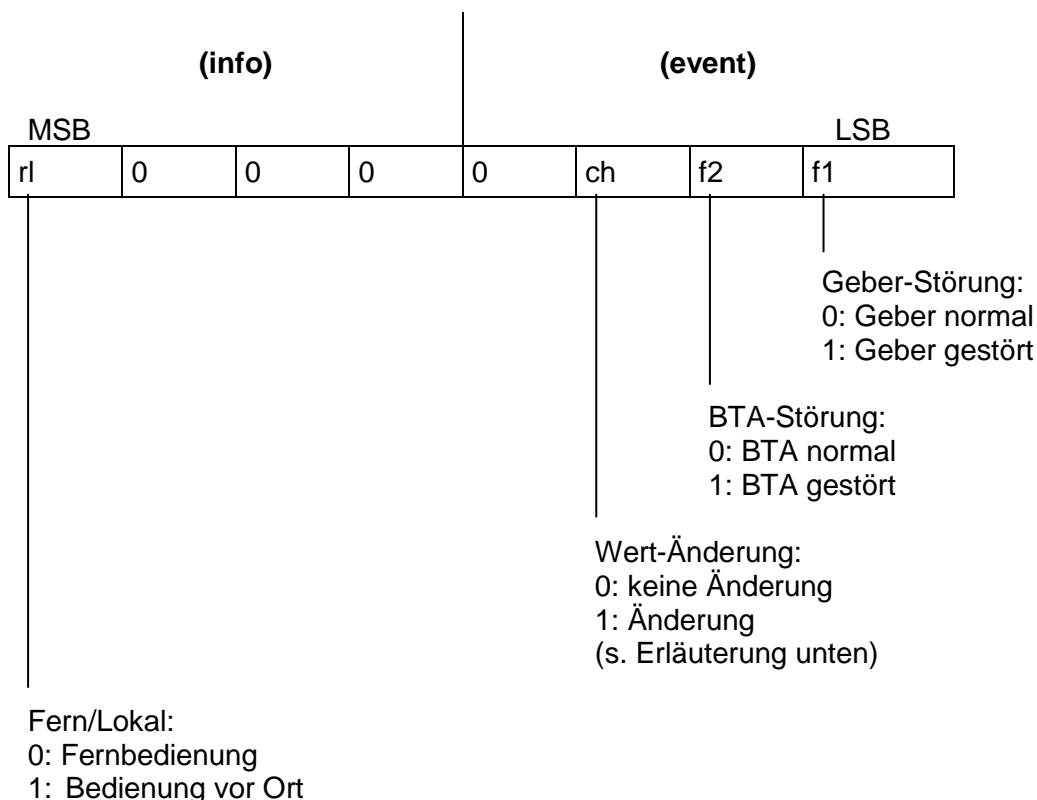
Variablenbeschreibung:

DP_#0

mask: (P_# 1)

(Zur Beschreibung dieses Oktetts:siehe Abschnitt 2.7, „Sperrungen von Meldungen“)

info_event:



Erläuterungen zu info_event:

- f1 (primary fault):
Ein gesetztes „f1“-Bit bedeutet eine Störung des Geberteils, die bei virtuellen Datenpunkten als ein erkannter interner Hard- oder Software-Fehler zu interpretieren ist.
Art und Ursache der Störung werden durch den Info-Code („exc“) näher spezifiziert.
- f2 (secondary fault):
Eine Störung der BTA (Warn-, Alarm-Zustand etc.) wird durch ein gesetztes „f2“-Bit angezeigt.
- ch (change):
Die Wertänderung eines Meldepunkts wird durch Setzen des „ch“-Bits signalisiert; Rücksetzen erfolgt aufgrund der LZ-seitigen Bestätigung der Meldung, bzw. bei Ausbleiben derselben nach einem intern gesetzten Timeout; der Rücksetzvorgang wird nicht als Ereignis gewertet.
Solange „ch“ auf 1 steht, bewirken weitere Wertänderungen keine Meldung. Sind Ereignisse dieser Art (Insel- oder LZ-seitig) ausgeblendet, so hat „ch“ während dieser Zeit den Wert 0.
- rl (remote/local):
Die Zustände Fern-/Vor-Ort-Bedienung werden durch das „rl“-Bit unterschieden. Diese Kennung ist insbesondere für solche Datenpunkte von Bedeutung, die Betriebsmeldungen zu Schaltpunkten darstellen.

Parameter-Übersicht:

P_#	Name	Erläuterung
1	mask	Unterdrückung von Spontanmeldungen aufgrund spezieller Ereignisse

DP_# 1

atb_id:

Zur Kennzeichnung der (im physikalischen Sinn) dimensionslosen Meldungen wird eine (erforderlichenfalls erweiterbare) Liste codierter Attributkombinationen verwendet, welche auf die GLT-spezifischen Anforderungen zugeschnitten ist; Beispiele hierfür sind etwa die Kombinationen („EIN“/„AUS“), („AUF“/„ZU“), oder („Stufe 0“/„Stufe I“/„Stufe II“). Die Wertigkeit der Meldung (ein- oder mehrstufig) ist dabei implizit durch die Anzahl der in der Kombination enthaltenen Attribute gegeben. Der Wert von atb-id ist identisch mit der Code-Nummer der Attribut-Kombination.

Die derzeit definierten Attribut-Kombinationen einschließlich ihrer Codierungen sind in Anhang E aufgelistet.

actual:

Sofern <> 'FF'H (der Kennzeichnung für einen ungültigen Wert), enthält dieses Objekt die aktuelle Meldung(s-Stufe), wobei

- höchstens ein Bit gesetzt und
- die Nummer dieses Bits (1-8) höchstens gleich der um 1 verminderten Anzahl der in der durch „atb_id“ adressierten Attribut-Kombination vorhandenen Attribute sein darf.

„actual“ ist dabei als Index im Wertebereich von 0 – 8 in der durch atb_id adressierten Attribut-Kombination (welche als max. 9-dimensionale String-Table aufgefasst wird) wie folgt zu interpretieren:

actual (binär)	Index
0 0 0 0 0 0 0 0	0
0 0 0 0 0 0 0 1	1
0 0 0 0 0 0 1 0	2
0 0 0 0 0 1 0 0	3
0 0 0 0 1 0 0 0	4
0 0 0 1 0 0 0 0	5
0 0 1 0 0 0 0 0	6
0 1 0 0 0 0 0 0	7
1 0 0 0 0 0 0 0	8

Zwei Beispiele sollen die vorstehenden Ausführungen verdeutlichen:

Beispiel # 1:

Für einen Datenpunkt mit den Meldungen („EIN“/„AUS“) ergeben sich nach der durch Anhang E gegebenen Codierung für die DP-Tabelle # 1 die Werte

atb_id	0 0 0 0 1 0 1 0	(Code 10 für „EIN“/„AUS“)
actual	0 0 0 0 0 0 0 0	(Index: 0)

für „EIN“, und

atb_id	0 0 0 0 1 0 1 0	(Code 10 für „EIN“/„AUS“)
actual	0 0 0 0 0 0 0 1	(Index: 1)

für „AUS“.

Beispiel # 2:

Ein dreistufiger Meldepunkt für die Zustände „Normal“ / „Warnung“ / „Alarm“ ist folgendermaßen zu behandeln:

atb_id	0 0 1 0 1 1 0 0	(Code: 44)
actual	0 0 0 0 0 0 0 0	(Index: 0)

für „Normal“

atb_id	0 0 1 0 1 1 0 0	(Code: 44)
actual	0 0 0 0 0 0 0 1	(Index: 1)

für „Warnung“, und

atb_id	0 0 1 0 1 1 0 0	(Code: 44)
actual	0 0 0 0 0 0 1 0	(Index: 2)

für „Alarm“.

Operations-Übersicht

Meldepunkt		fct_id	
d_r	tab_id	1 (Lesen)	2 (Mod.)
0 (Cmd)	0 (DP_#0)	C1	C2
	1 (DP_#1)	C3	////////////////////
3 (Rsp)	0 (DP_#0)	R1	R2
	1 (DP_#1)	R3	////////////////////
2 (Usm)	1 (DP_#1)	U1	////////////////////
1 (Ack)	1 (DP_#1)	A1	////////////////////

(Schraffierter Bereich nicht anwendbar)

Symbol-Erklärung

Operation	Bedeutung	Daten-Teil
C1	DP-Status (DP_#0) anfordern	(leer)
C2	Meldungs-Kontroll-Feld (msk in DP_#0) mod.	(leer)
C3	DP-Nutzdaten (DP_#1) anfordern	(leer)
R1	DP-Status (DP_#0) übermitteln	(leer)
R2	Modifikationsbestätigung des Meldungs-Kontrollfeldes	(leer)
R3	DP-Nutzdaten (DP_#1) übermitteln	DP_# 1
U1	Spontanmeldung	DP_# 1
A1	Bestätigung Spontanmeldung	leer

Beispiel: Die LZ möchte die aktuelle Meldungs-Stufe lesen. Diese Information („actual“) ist Bestandteil der Nutzdaten in DP-Tabelle #1. Die LZ sendet somit ein CMD-Telegramm gemäß Operation C3 und erhält im RSP-Telegramm gemäß R3 als Datenteil die 2 Oktette „atd_id“ und „actual“ der DP_#1.

(Für weitergehende Ausführungen zu dieser tabellarischen Übersicht wird auf das Kapitel „Operationsbeschreibung“ verwiesen).

2.3.2 Schaltpunkt (mit Rückmeldung)

Charakterisierung:

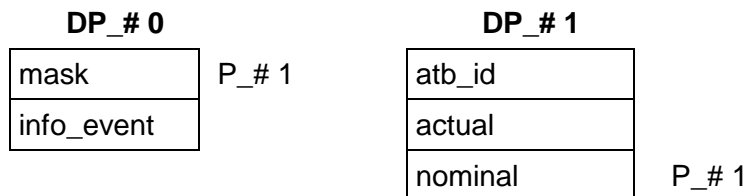
Ein Schaltpunkt mit Rückmeldung stellt einen Schaltpunkt mit „integriertem feedback“ dar, d.h. die zugehörige (Betriebs-)Meldung ist über die gleiche Datenpunkt-Adresse verfügbar.

Ein Schaltpunkt mit Rückmeldung entspricht einem Meldepunkt (inkl. Spontanmeldung bei Wertänderung), dem zusätzlich ein Sollwert für das Schalten vorgegeben werden kann. Die IZ entscheidet aufgrund der zugeordneten Funktion, ob und wann der Schaltvorgang ausgeführt wird.

Beispiel: Eine dämmerungs-abhängige Außenbeleuchtung wird in bestimmten Zeitbereichen genutzt, d.h. innerhalb dieser Zeitbereich wird der Zustand EIN für den Schaltpunkt vorgegeben. Die IZ schaltet die Außenbeleuchtung aber nur dann ein, wenn auch der integrierte Dämmerungsschalter einschaltet, und sendet für die Wertänderung spontan eine Schaltmeldung. Für den Vorgabewert AUS bleibt die Außenbeleuchtung unabhängig vom Dämmerungsschalter ausgeschaltet.

Hinweis: Die LHM setzt ausschließlich Schaltpunkte mit Rückmeldung ein. Dabei wird abhängig von der Funktion entschieden, ob die Rückmeldung über einen gesonderten Geber oder intern in der IZ erzeugt wird.

Datenpunkt-Tabellen:



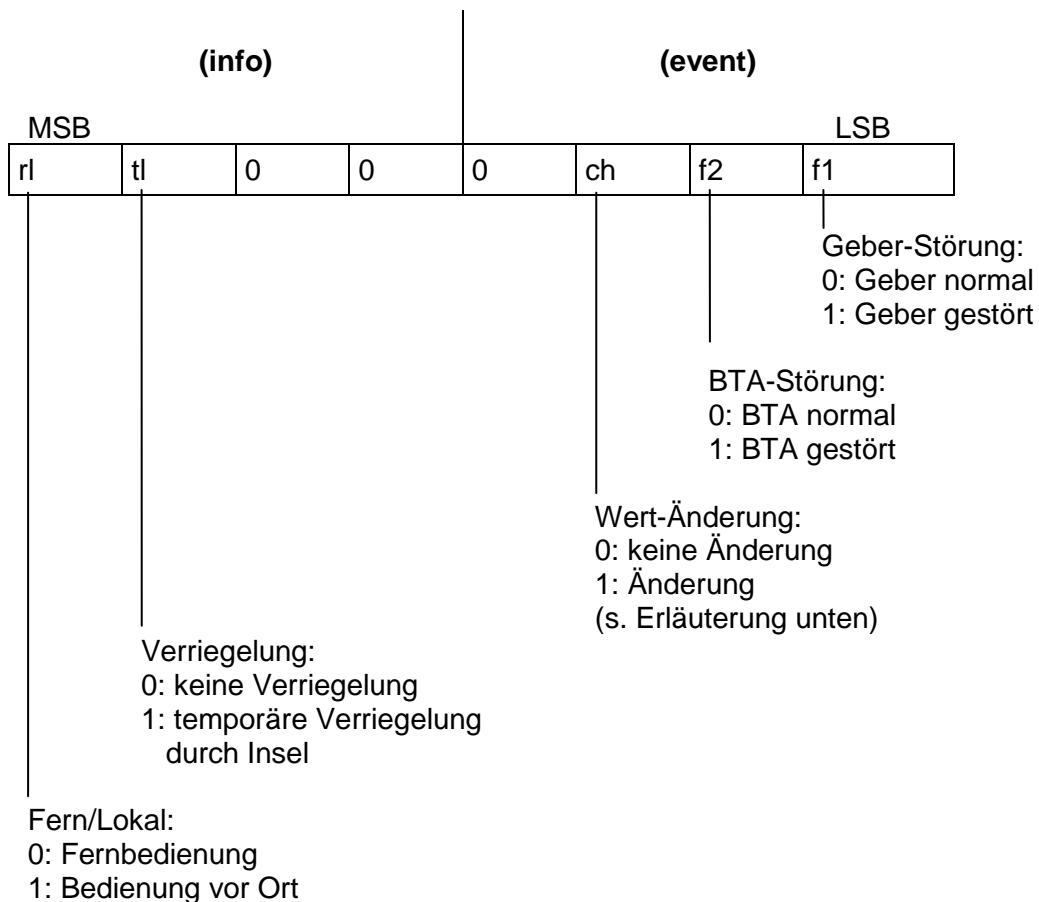
Variablenbeschreibung:

DP_# 0

mask: (P_# 1)

(Zur Beschreibung dieses Oktetts siehe Abschnitt 2.7, „Sperrern von Meldungen“)

info_event:



Erläuterungen zu info_event:

- f1 (primary fault):
Ein gesetztes „f1“-Bit bedeutet eine Störung des Geberteils, die bei virtuellen Datenpunkten als ein erkannter interner Hard- oder Software-Fehler zu interpretieren ist.
Art und Ursache der Störung werden durch den Info-Code („exc“) näher spezifiziert.
- f2 (secondary fault):
Eine Störung der BTA (Warn-, Alarm-Zustand etc.) wird durch ein gesetztes „f2“-Bit angezeigt.
- ch (change):
Jede Wertänderung eines Meldepunkts wird durch Setzen des „ch“-Bits signalisiert; Rücksetzen erfolgt aufgrund der LZ-seitigen Bestätigung der Meldung, bzw. bei Ausbleiben derselben nach einem intern gesetzten Timeout; der Rücksetzvorgang wird nicht als Ereignis gewertet.
Solange „ch“ auf 1 steht, bewirken weitere Wertänderungen keine Meldung. Sind Ereignisse dieser Art (Insel- oder LZ-seitig) ausgeblendet, so hat „ch“ während dieser Zeit den Wert 0.

In der Antwort auf einen Schaltbefehl ist das ch-Bit zu setzen, falls der Befehl sofort ausgeführt werden konnte.

Für Schaltpunkte ohne Rückmeldung ist das ch-Bit stets 0 zu setzen.

- tl (temporary lock):
Eine durch die Insel gesteuerte, zeitlich begrenzte Verriegelung des Schaltpunktes wird durch das "tl"-Bit angezeigt. Schaltbefehle seitens FND werden in diesem Zeitraum abgewiesen
- rl (remote/local):
Die Zustände Fern-/Vor-Ort-Bedienung werden durch das „rl“-Bit unterschieden.

Parameter-Übersicht

P_#	Name	Erläuterung
1	mask	Unterdrückung von Spontanmeldungen aufgrund spezieller Ereignisse

DP_# 1

atb_id:

(wie bei Meldepunkt, s. dort)

actual:

(wie bei Meldepunkt, s. dort). Für Schaltpunkte ohne Rückmeldung ist stets 'FF'H (der Kennzeichnung für einen ungültigen Wert) einzutragen.

nominal: (P_# 1)

Dieses Oktett enthält die FND-Vorgabe des Schaltwertes, wobei (analog zum Wert „actual“ eines Meldepunktes)

- höchstens ein Bit gesetzt sein darf (welches dem Schaltwert entspricht)
- die Nummer dieses Bits (1 – 8) höchstens gleich der um 1 verminderten Anzahl der in der durch „atb_id“ adressierten Attribut-Kombination vorhandenen Attribute sein darf.

(Bzgl. Der Codierung im Zusammenhang mit der Variablen „atb_id“ sei auf die entsprechenden Bemerkungen bei Meldepunkten verwiesen).

Parameter-Übersicht

P_#	Name	Erläuterung
1	nominal	Der zum FND – Schaltbefehl gehörende Wert *)

*) Anhand der auf einen LZ-seitigen Schaltbefehl eintreffenden Antwort kann abgelesen werden, ob der Vorgang bereits ausgeführt wurde, oder erst mit einer zeitlichen Verzögerung als Ereignis gemeldet wird. Kriterium hierfür ist, ob der in „actual“ stehende Wert mit dem über FND gegebenen „nominal“-Wert übereinstimmt (Befehl ausgeführt), oder noch nicht (Ausführungsbestätigung später in Form einer Spontanmeldung).

Operations-Übersicht

Schaltpunkt		fct_id	
d_r	tab_id	1 (Lesen)	2 (Mod.)
0 (Cmd)	0 (DP_#0)	C1	C2
	1 (DP_#1)	C3	C4
3 (Rsp)	2 (DP_#0)	R1	R2
	3 (DP_#1)	R3	R4
2 (Usm)	1 (DP_#1)	U1	////////////////////
1 (Ack)	2 (DP_#1)	A1	////////////////////

(Schraffierter Bereich nicht anwendbar)

Symbol-Erklärung

Operation	Bedeutung	Daten-Teil
C1	DP-Status (DP_#0) anfordern	(leer)
C2	Meldungs-Kontroll-Feld (msk in DP_#0) mod.	(leer)
C3	DP-Nutzdaten (DP_#1) anfordern	(leer)
C4	Schalten	DP_# 1
R1	DP-Status (DP_#0) übermitteln	(leer)
R2	Modifikationsbestätigung des Meldungs-Kontrollfeldes	(leer)
R3	DP-Nutzdaten (DP_#1) übermitteln	DP_# 1
R4	Schalt-Quittung	DP_# 1
U1	Schaltmeldung	DP_# 1
A1	Bestätigung Spontanmeldung	leer

Beispiel: Die LZ möchte den Zustand EIN vorgeben. Diese Information („nominal“) ist Bestandteil der Nutzdaten in DP-Tabelle #1. Die LZ sendet somit ein CMD-Telegramm gemäß Operation C4 und erhält im RSP-Telegramm gemäß R4. Wurde der Schaltbefehl sofort ausgeführt, so ist dabei bereits das ch-Bit gesetzt und der „actual“ entspricht dem „nominal“. Sonst bleibt der „actual“ unverändert und mit der späteren Ausführung wird eine spontane Schaltmeldung gemäß U1 erzeugt.

Für Schaltpunkte ohne Rückmeldung entfällt die Schaltmeldung (Operation U1 und A1)

(Für weitergehende Ausführungen zu dieser tabellarischen Übersicht wird auf das Kapitel „Operationsbeschreibung“ verwiesen).

2.3.3 Messpunkt

Charakterisierung:

Messpunkte sind kontinuierlich veränderbaren Größen zugeordnet, die auf Einhaltung vorgegebener Grenzwerte überwacht werden können. Gemäß der realen oder logischen (virtuellen) Natur des Messwertes kann es sich dabei um eine Temperatur, die Position eines Stellgliedes, aber auch um einen aus mehreren Größen rechnerisch ermittelten Funktionswert handeln.

Datenpunkt-Tabellen:

DP_# 0

mask	P_# 1
info_event	

DP_# 1

dimension
actual

DP_# 2

update_control	
fixed	
alarm_low	P_# 1
warning_low	P_# 2
warning_high	P_# 3
alarm_high	P_# 4

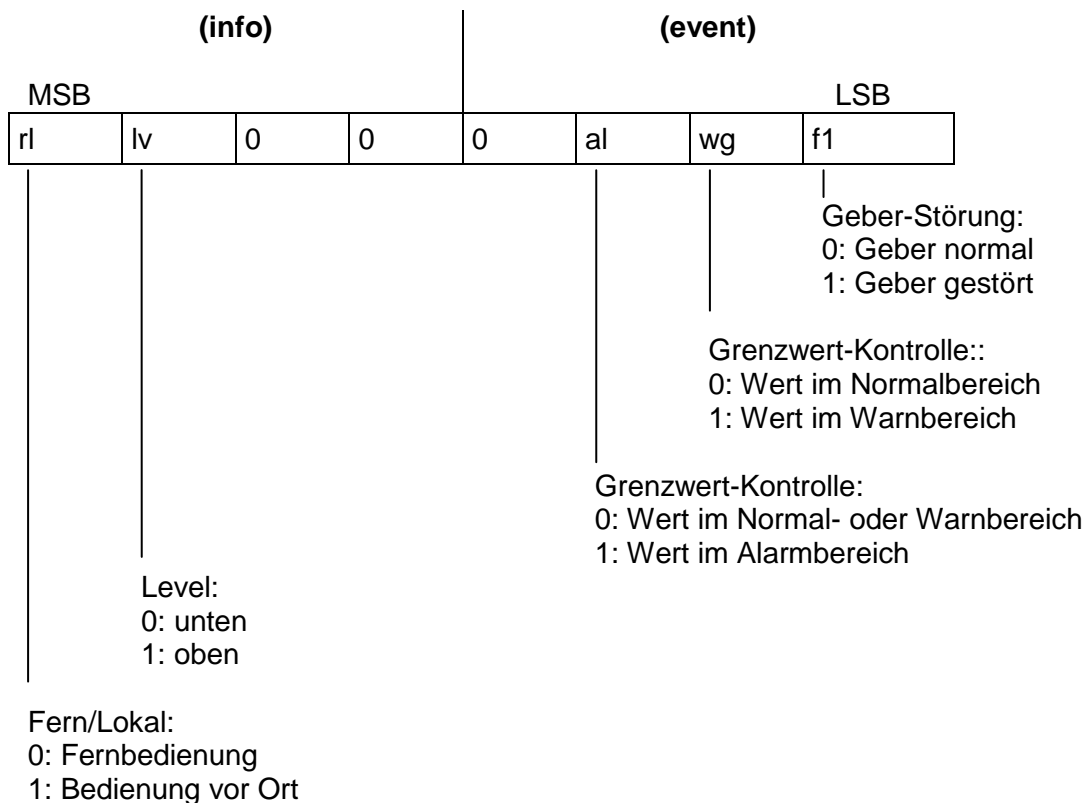
Variablenbeschreibung:

DP_# 0

mask: (P_# 1)

(Zur Beschreibung dieses Oktetts: siehe Abschnitt 2.7, „Sperren von Meldungen“)

info_event:



Erläuterungen zu info_event:

- f1 (primary fault):
ein gesetztes „f1“-Bit bedeutet eine Störung des Geberteils, die bei virtuellen Datenpunkten als ein erkannter interner Hard- oder Software-Fehler zu interpretieren ist.
Art und Ursache der Störung werden durch den Info-Code („exc“) näher spezifiziert.
- wg (warning):
Ein gesetztes „wg“-Bit zeigt an, dass der Messwert innerhalb des oberen (gekennzeichnet durch lv = 1) bzw. unteren (gekennzeichnet durch (lv = 0) Warnbereichs liegt. Der Übergang aus dem bzw. in den Warnbereich wird als Ergebnis gewertet.
- al (alarm):
Analog zu „wg“, wobei „Warnbereich“ durch „Alarmbereich“ zu ersetzen ist. Der Alarmbereich wird als Teilmenge des Warnbereiches betrachtet.
- lv (level):
Dieses Bit liefert bei Grenzwertmeldungen die Zusatzinformation „Überschreitung“ (= „oben“) bzw. „Unterschreitung“ (= „unten“).
- rl (remote/local):
Die Zustände Fern-/Vor-Ort-Bedienung werden durch das „rl“-Bit unterschieden

Bei den zulässigen Kombinationen der vier Variablen „wg“, „al“, „lv“ und „emq“ (siehe 2.5.1.3 und 2.5.2.3) ist zwischen einem statischen und transienten Kontext zu unterscheiden, wobei sich ersterer auf Abfragen und letzterer auf Spontanmeldungen bezieht.

Zulässige Kombination bei Abfragen				
emq	lv	al	wg	Bedeutung
0 0 0 0	0	1	1	Messwert im unteren Alarmbereich
0 0 0 0	0	0	1	Messwert im unteren Warnbereich
0 0 0 0	0	0	0	Messwert im Normalbereich
0 0 0 0	1	0	1	Messwert im oberen Warnbereich
0 0 0 0	1	1	1	Messwert im oberen Alarmbereich
andere Kombinationen: illegal				
Zulässige Kombination bei Spontanmeldungen (X = 0 oder 1)				
emq	lv	al	wg	Bedeutung
0 0 1 X	1	0	1	Überschreitung der oberen Warngrenze kommend Neuer Zustand: oberer Warnbereich
0 1 0 X	1	1	1	Überschreitung der oberen Alarmgrenze kommend Neuer Zustand: oberer Alarmbereich
0 1 0 X	1	0	1	Überschreitung der oberen Alarmgrenze gehend Neuer Zustand: oberer Warnbereich
0 0 1 X	1	0	0	Überschreitung der oberen Warngrenze gehend Neuer Zustand: Normalbereich
0 0 1 X	0	0	1	Unterschreitung der unteren Warngrenze kommend Neuer Zustand: unterer Warnbereich
0 1 0 X	0	1	1	Unterschreitung der unteren Alarmgrenze kommend Neuer Zustand unterer Alarmbereich
0 1 0 X	0	0	1	Unterschreitung der unteren Alarmgrenze gehend Neuer Zustand: unterer Warnbereich
0 0 1 X	0	0	0	Unterschreitung der unteren Warngrenze gehend Neuer Zustand: Normalbereich
andere Kombinationen: illegal				

Parameter-Übersicht

P_#	Name	Erläuterung
1	mask	Unterdrückung von Spontanmeldungen aufgrund spezieller Ereignisse

DP_# 1

dimension:

„dimension“ repräsentiert in codierter Form die physikalische Dimension des Messwerts. Die Liste der von FND unterstützten Größen und ihrer Codes ist in Anhang E enthalten.

actual:

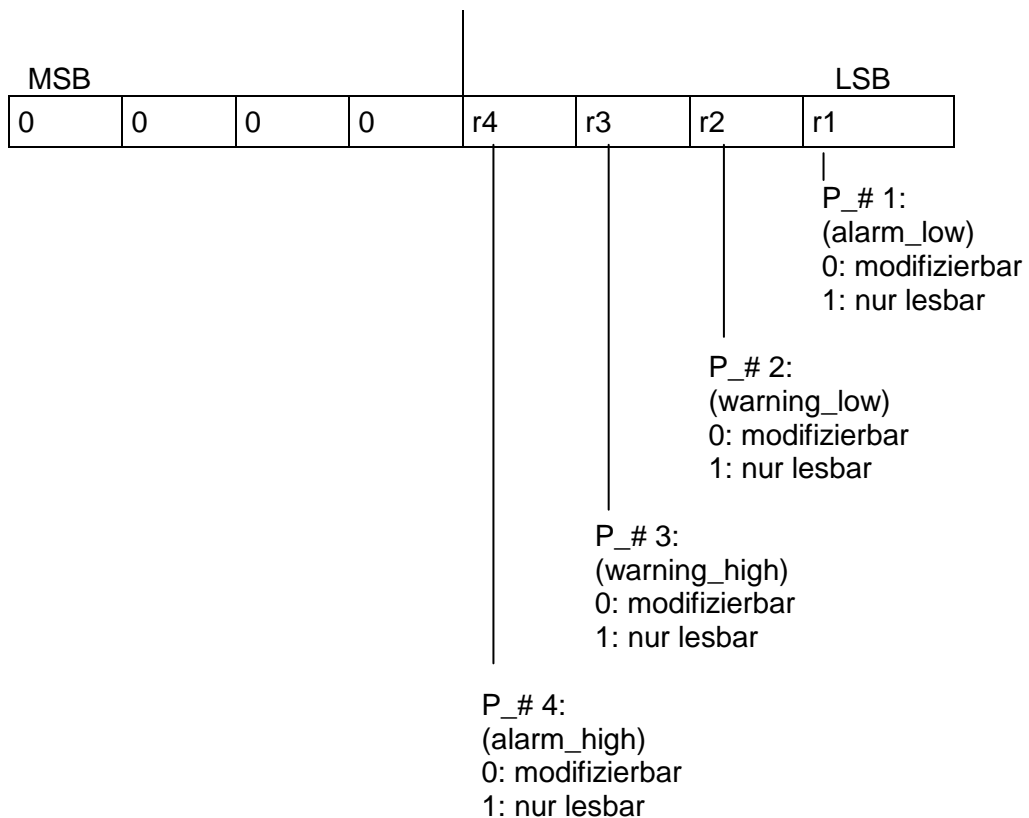
„actual“ enthält den aktuellen Messwert in IEEE 754-Darstellung. Ein Wert von 'FFFFFFFF'H (alle Bits gesetzt) markiert einen ungültigen Messwert.

DP_# 2

update_control:

Dieses Byte stellt eine zusätzliche Modifizierungskontrolle der nachstehenden Parameter P_# 1 bis P_# 4 in der Weise dar, dass ein an entsprechender Stelle gesetztes Bit für FND nur lesenden Zugriff auf den Parameter impliziert. Dies kann z.B. dann relevant sein, wenn ein Parameter von diesem Datenpunkt generell nicht unterstützt wird (angezeigt durch den Wert 'FFFFFFFF'H), oder etwa nur der obere Grenzwert von der LZ aus verändert werden darf.

update_control:



fixed:

Kennzeichnung, ob die nachstehenden Grenzen als absolut oder gleitend zu interpretieren sind.

Codierung:

- TRUE (=1): Grenzwerte absolut
- FALSE (=0): Grenzwerte gleitend

(Sofern keine Grenzwertvorgaben existieren, ist der Wert von „fixed“ ohne Belang).

Hinweis: Der Wert von „fixed“ ist stets auf TRUE zu setzen, da noch keine Spezifikation gleitender Grenzwerte vorliegt.

alarm_low: (P_# 1)

Untere Alarmgrenze des Messwerts. Ein Eintrag von `FFFFFFFF`H (alle Bits gesetzt) bedeutet die Inaktivität (oder Nichtexistenz) des Grenzwerts.

warning_low: (P_# 2)

Untere Warngrenze des Messwerts. Ein Eintrag von `FFFFFFFF`H (alle Bits gesetzt) bedeutet die Inaktivität (oder Nichtexistenz) des Grenzwerts.

warning_high: (P_# 3)

Obere Warngrenze des Messwerts. Ein Eintrag von `FFFFFFFF`H (alle Bits gesetzt) bedeutet die Inaktivität (oder Nichtexistenz) des Grenzwerts.

alarm_high (P_# 4)

Obere Alarmgrenze des Messwerts. Ein Eintrag von `FFFFFFFF`H (alle Bits gesetzt) bedeutet die Inaktivität (oder Nichtexistenz) des Grenzwerts.

Parameter-Übersicht:

P_#	Name	Erläuterung
1	alarm_low	Untere Alarmgrenze
2	warning_low	Untere Warngrenze
3	warning_high	Obere Warngrenze
4	alarm_high	Obere Alarmgrenze

Operations-Übersicht

Messpunkt		fct_id	
d_r	tab_id	1 (Lesen)	2 (Mod.)
0 (Cmd)	0 (DP_#0)	C1	C2
	1 (DP_#1)	C3	////////////////////////////////
	2 (DP_#2)	C4	C5
	3 (DP_#1.2)	C6	////////////////////////////////
3 (Rsp)	0 (DP_#0)	R1	R2
	1 (DP_#1)	R3	////////////////////////////////
	2 (DP_#2)	R4	R5
	3 (DP_#1.2)	R6	////////////////////////////////
2 (Usm)	1 (DP_#1)	U1	////////////////////////////////
1 (Ack)	1 (DP_#1)	A1	////////////////////////////////

(Schraffierter Bereich nicht anwendbar)

Symbol-Erklärung

Operation	Bedeutung	Daten-Teil
C1	DP-Status (DP_#0) anfordern	(leer)
C2	Meldungs-Kontroll-Feld (msk in DP_#0) mod.	(leer)
C3	DP-Nutzdaten (DP_#1) anfordern	(leer)
C4	DP-Parameter (DP_#2) anfordern	(leer)
C5	DP-Parameter (DP_#2) modifizieren	DP_#2
C6	DP-Nutzdaten+Par. (DP_#1+2) anfordern	(leer)
R1	DP-Status (DP_#0) übermitteln	(leer)
R2	Modifikationsbestätigung des Meldungs-Kontrollfeldes	(leer)
R3	DP-Nutzdaten (DP_#1) übermitteln	DP_# 1
R4	DP-Parameter (DP_#2) übermitteln	DP_# 2
R5	Modifikationsbestätigung Param. (DP_#2)	DP_# 2
R6	DP-Nutzdaten+Par. (DP_#1+2) übermitteln	DP_#1. 2
U1	Messwertmeldung	DP_# 1
A1	Bestätigung Messwertmeldung	(leer)

Beispiel: Die LZ möchte die neue Grenzwerte vorgeben. Diese Information („alarm_low“ usw.) ist Bestandteil der Parameter in DP-Tabelle #2. Die LZ sendet somit ein CMD-Telegramm gemäß Operation C5 und erhält im RSP-Telegramm gemäß R5.

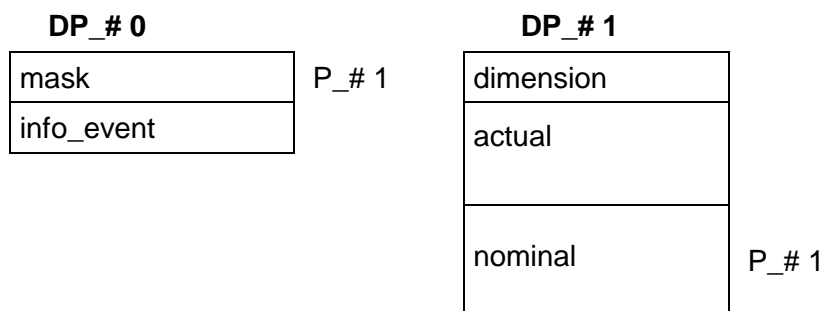
(Für weitergehende Ausführungen zu dieser tabellarischen Übersicht wird auf das Kapitel „Operationsbeschreibung“ verwiesen).

2.3.4 Stellpunkt

Charakterisierung:

Bei Stellpunkten führt die Vorgabe eines Stellwerts zu einer entsprechenden Stellung des Gebers. Gemäß dessen realer oder logischer (virtueller) Natur kann es sich dabei sowohl um die Position eines physikalischen Stellglieds, als auch etwa um einen Programm-Parameter handeln.

Datenpunkt-Tabellen:



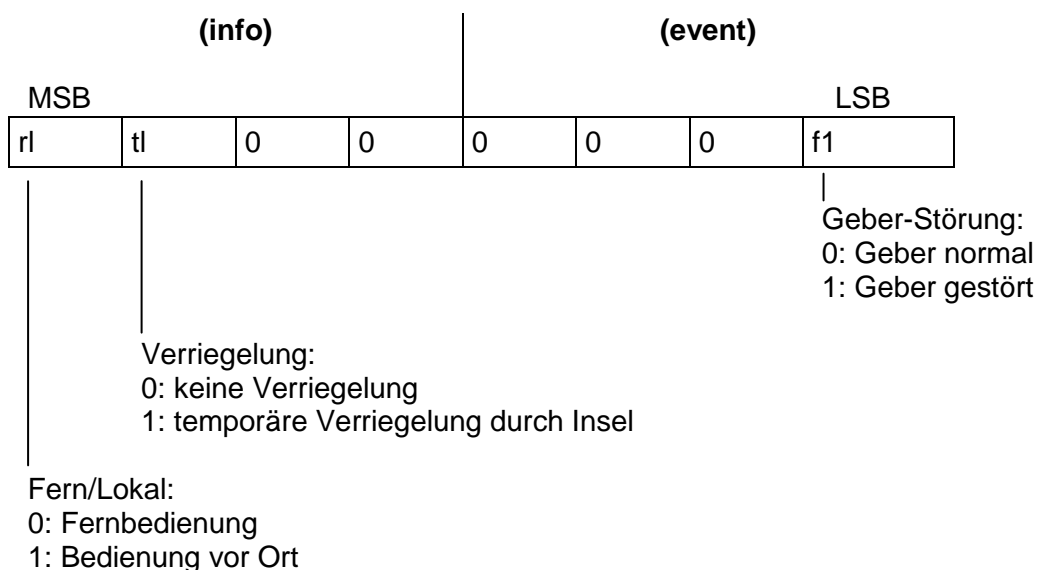
Variablenbeschreibung:

DP_# 0

mask: (P_# 1)

(Zur Beschreibung dieses Oktetts: siehe Abschnitt 2.7, „Sperren von Meldungen“)

info_event:



Erläuterungen zu info_event:

- f1 (primary fault):
ein gesetztes „f1“-Bit bedeutet eine Störung des Geberteils, die bei virtuellen Datenpunkten als ein erkannter interner Hard- oder Software-Fehler zu interpretieren ist.
Art und Ursache der Störung werden durch den Info-Code („exc“) näher spezifiziert.
- tl (temporary lock):
Eine durch die Insel gesteuerte, zeitlich begrenzte Verriegelung des Schaltpunktes wird durch das „tl“-Bit angezeigt. Stellbefehle seitens FND werden in diesem Zeitraum abgewiesen.
- rl (remote/local):

Die Zustände Fern-/Vor-Ort-Bedienung werden durch das „rl“-Bit unterschieden

Parameter-Übersicht:

P_#	Name	Erläuterung
1	mask	Unterdrückung von Spontanmeldungen aufgrund spezieller Ereignisse

DP_# 1

dimension:

(wie bei Messpunkt, s. dort)

actual:

„actual“ enthält die zu „nominal“ gehörende Stellungsmessung in Short-Real-Darstellung gemäß IEEE 754.

Ein Wert von 'FFFFFFFF'H (alle Bits gesetzt) markiert einen ungültigen Messwert.

nominal:

„nominal“ enthält den letzten (aktuellen) über FND vorgegebenen Stellwert, ebenfalls in Short-Real-Darstellung gemäß IEEE 754.

Parameter-Übersicht:

P_#	Name	Erläuterung
1	nominal	Vorgegebener Stellwert

Operations-Übersicht

Stellpunkt		fct_id	
d_r	tab_id	1 (Lesen)	2 (Mod.)
0 (Cmd)	0 (DP_#0)	C1	C2
	1 (DP_#1)	C3	C4
3 (Rsp)	0 (DP_#0)	R1	R2
	1 (DP_#1)	R3	R4
2 (Usm)	1 (DP_#1)	U1	////////////////////
1 (Ack)	1 (DP_#1)	A1	////////////////////

(Schraffierter Bereich nicht anwendbar)

Symbol-Erklärung

Operation	Bedeutung	Daten-Teil
C1	DP-Status (DP_#0) anfordern	(leer)
C2	Meldungs-Kontroll-Feld (msk in DP_#0) mod.	(leer)
C3	DP-Nutzdaten (DP_#1) anfordern	(leer)
C4	Stellen	DP_# 1
R1	DP-Status (DP_#0) übermitteln	(leer)
R2	Modifikationsbestätigung des Meldungs-Kontrollfeldes	(leer)
R3	DP-Nutzdaten (DP_#1) übermitteln	DP_# 1
R4	Stell-Quittung	DP_# 1
U1	Stellmeldung	DP_# 1
A1	Bestätigung Stellmeldung	leer

Beispiel: Die LZ möchte den Wert „50%“ vorgeben. Diese Information („nominal“) ist Bestandteil der Nutzdaten in DP-Tabelle #1. Die LZ sendet somit ein CMD-Telegramm gemäß Operation C4 und erhält im RSP-Telegramm gemäß R4.

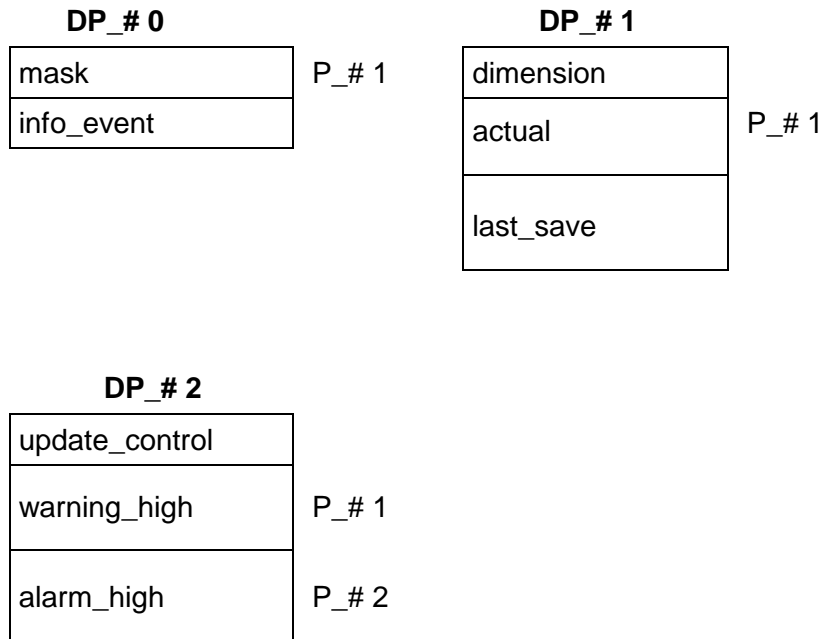
(Für weitergehende Ausführungen zu dieser tabellarischen Übersicht wird auf das Kapitel „Operationsbeschreibung“ verwiesen).

2.3.5 Zählpunkt

Charakterisierung:

Der Zählpunkt ist einer Größe zugeordnet, die im zeitlichen Verlauf monoton wachsenden Charakter hat. Unstetigkeiten entstehen lediglich durch (Rück-)Setzvorgänge oder Überlauf des Zählers.

Datenpunkt-Tabellen:



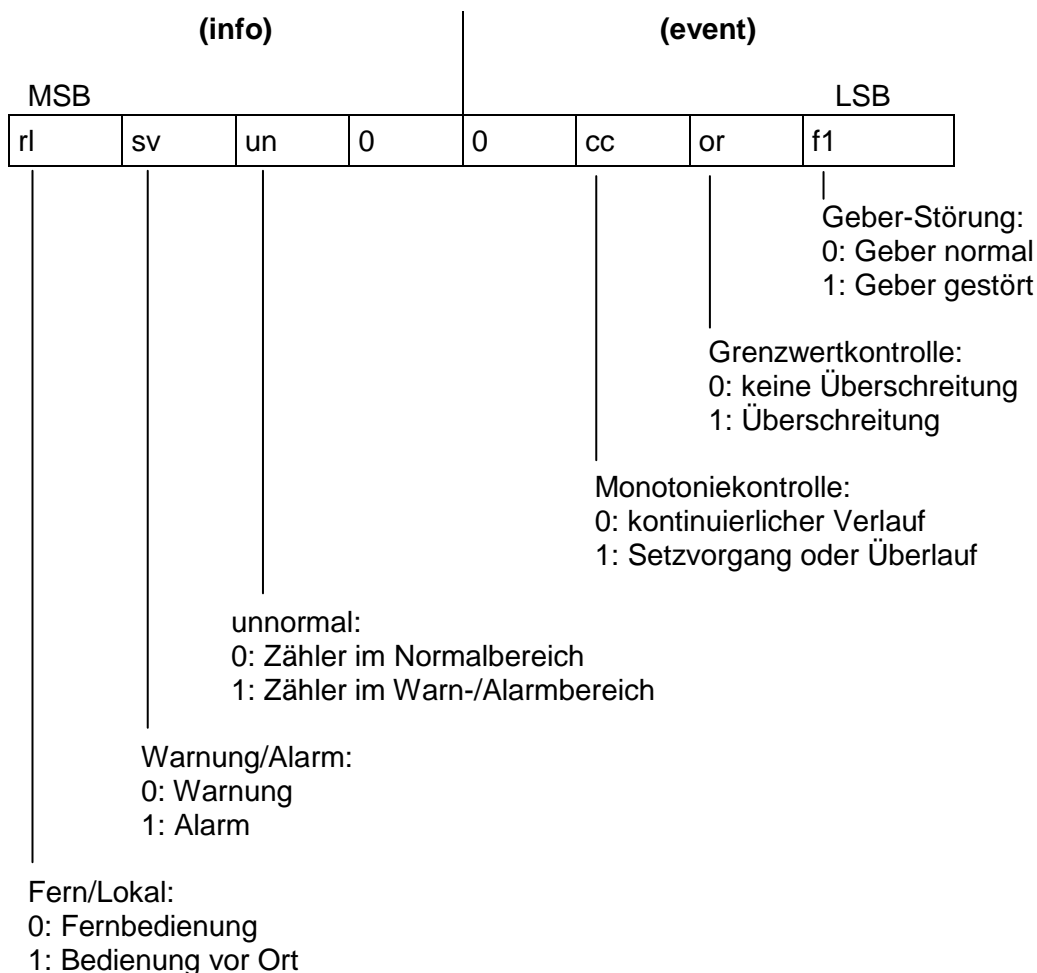
Variablenbeschreibung:

DP_# 0

mask: (P_# 1)

(Zur Beschreibung dieses Oktetts: siehe Abschnitt 2.7, „Sperrern von Meldungen“)

info_event:



Erläuterungen zu info_event:

- f1 (primary fault):
ein gesetztes „f1“-Bit bedeutet eine Störung des Geberteils, die bei virtuellen Datenpunkten als ein erkannter interner Hard- oder Software-Fehler zu interpretieren ist.
Art und Ursache der Störung werden durch den Info-Code („exc“) näher spezifiziert.
- or (overrange):
Ein gesetztes „or“-Bit zeigt die Überschreitung einer oberen Warn- oder Alarmgrenze an; der Wechsel vom Normalbereich in den Überschreitungsbe-
reich wird als Ereignis gewertet. Die Unterscheidung zwischen Warnung und Alarm wird durch as „sv“-Bit ermöglicht.
Das „or“-Bit ist nach der LZ-seitigen Bestätigung der Meldung bzw. bei Aus-
bleiben deselben nach einem intern gesetzten Timeout zurückzusetzen; der
Rücksetzvorgang wird nicht als Ereignis gewertet. Sind Ereignisse dieser Art
ausgeblendet, so hat or während dieser Zeit den Wert 0.
- cc (continuity control):
Ein gesetztes „cc“-Bit signalisiert einen (Rück-)Setzvorgang oder einen Über-
lauf des Zählers. Dabei steht der unmittelbar vor dem Ereignis gültige Zähl-
wert „last_save“, der neue Wert in „actual“.

Rücksetzen des „cc“-Bits erfolgt aufgrund der LZ-seitigen Bestätigung der Meldung, bzw. bei Ausbleiben derselben nach einem intern gesetzten Timeout; der Rücksetzvorgang wird nicht als Ereignis gewertet. Solange „cc“ auf 1 steht, bewirken weitere (Rück-)Setz- oder Überlaufvorgänge keine Meldung. Sind Ereignisse dieser Art (Insel- oder LZ-seitig) ausgeblendet, so hat „cc“ während dieser Zeit den Wert 0.

- un (unnormal):
Das „un“-Bit informiert über eine Grenzwert-Überschreitung, abgestuft in die Kategorien „Warnung“ und „Alarm“ gemäß „sv“-Bit. Es ist zu setzen, falls der Zählwert nicht im Normalbereich, also im Warn- oder Alarmbereich liegt.
- sv (severity):
Das „sv“-Bit informiert über den Grad der Grenzwert-Überschreitung, abgestuft in die Kategorien „Warnung“ und „Alarm“.
- rl (remote/local):
Die Zustände Fern-/Vor-Ort-Bedienung werden durch das „rl“-Bit unterschieden.

Die zulässigen Kombinationen der zwei Variablen „or“ und „sv“ sowie deren Bedeutung ist nachstehender Tabelle zu entnehmen:

sv	un	Bedeutung
0	0	Zählwert im Normalbereich
0	1	Zählwert im Warnbereich
1	1	Zählwert im Alarmbereich
1	0	(illegal)

Parameter-Übersicht:

P_#	Name	Erläuterung
1	mask	Unterdrückung von Spontanmeldungen aufgrund spezieller Ereignisse

DP_# 1

dimension:

(wie bei Messpunkt, s. dort)

actual:

„actual“ enthält den aktuellen Zählwert in IEEE 754-Darstellung; ein Eintrag von 'FFFFFFFF'H markiert die Ungültigkeit des Zählwerts.

last_save:

Der Wert dieser Variablen, die zu 0 initialisiert wird, enthält den gesicherten Wert unmittelbar vor dem letzten (Rück-)Setzvorgang oder Überlauf des Zählers.

DP_# 2

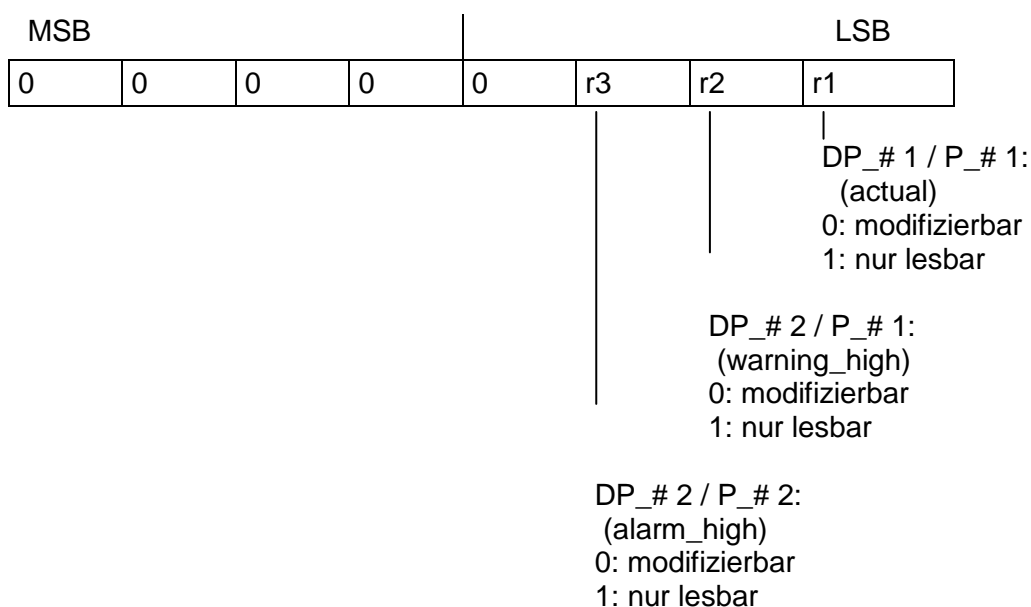
update_control:

Dieses Byte stellt eine zusätzliche Modifizierungskontrolle der Parameter

- DP_# 1/P_# 1
- DP_# 2/P_# 1
- DP_# 2/P_# 2

in der Weise dar, dass ein an entsprechender Stelle gesetztes Bit für FND nur lesenden Zugriff auf den Parameter erlaubt. Dies kann z.B. dann relevant sein, wenn ein Parameter von diesem Datenpunkt generell nicht unterstützt wird (angezeigt durch den Wert 'FFFFFFFF'J), oder etwa nur ein oberer Grenzwert von der LZ aus verändert werden darf.

update_control:



warning_high: (P_# 1)

Obere Warngrenze des Zählwerts in Real-Darstellung. Ein Eintrag von 'FFFFFFFF'H (alle Bits gesetzt) bedeutet die Inaktivität (oder Nichtexistenz) des Grenzwerts.

alarm_high (P_# 2)

Obere Alarmgrenze des Zählwerts in Real-Darstellung. Ein Eintrag von 'FFFFFFFF'H (alle Bits gesetzt) bedeutet die Inaktivität (oder Nichtexistenz) des Grenzwerts.

Parameter-Übersicht:

Parameter	Name	Erläuterung
DP-Tab. # 1 / P_# 1	actual / (preset)	Vorbesetzung des Zählwerts über FND
DP-Tab. # 2 / P_# 1	warning_high	Obere Warngrenze
DP-Tab. # 2 / P_# 2	alarm_high	Obere Alarmgrenze

Operations-Übersicht

Zählpunkt		fct_id	
d_r	tab_id	1 (Lesen)	2 (Mod.)
0 (Cmd)	0 (DP_#0)	C1	C2
	1 (DP_#1)	C3	C4
	2 (DP_#2)	C5	C6
	3 (DP_#1.2)	C7	////////////////////
3 (Rsp)	0 (DP_#0)	R1	R2
	1 (DP_#1)	R3	R4
	2 (DP_#2)	R4	R6
	3 (DP_#1.2)	R7	////////////////////
2 (Usm)	1 (DP_#1)	U1	////////////////////
1 (Ack)	1 (DP_#1)	A1	////////////////////

(Schraffierter Bereich nicht anwendbar)

Symbol-Erklärung

Operation	Bedeutung	Daten-Teil
C1	DP-Status (DP_#0) anfordern	(leer)
C2	Meldungs-Kontroll-Feld (msk in DP_#0) mod.	(leer)
C3	DP-Nutzdaten (DP_#1) anfordern	(leer)
C4	DP-Nutzdaten (DP_#1) modifizieren	DP_#1
C5	DP-Parameter (DP_#2) anfordern	(leer)
C6	DP-Parameter (DP_#2) modifizieren	DP_#2
C7	DP-Nutzdaten+Par. (DP_#1+2) anfordern	(leer)
R1	DP-Status (DP_#0) übermitteln	(leer)
R2	Modifikationsbestätigung des Meldungs-Kontrollfeldes	(leer)
R3	DP-Nutzdaten (DP_#1) übermitteln	DP_# 1
R4	Modifikationsbestätigung Nutzdaten (DP_#1)	DP_# 1 *)
R5	DP-Parameter (DP_#2) übermitteln	DP_# 2
R6	Modifikationsbestätigung Param. (DP_#2)	DP_# 2
R7	DP-Nutzdaten+Par. (DP_#1+2) übermitteln	DP_#1. 2
U1	Zählwertmeldung	DP_# 1
A1	Bestätigung Zählwertmeldung	(leer)

*) Anhand der auf einem LZ-seitigen (Rück-)Setzbefehl eintreffenden Antwort kann abgelesen werden, ob der Vorgang bereits ausgeführt wurde, oder erst mit einer zeitlichen Verzögerung als Ereignis gemeldet wird. Kriterium hierfür ist, ob der in „actual“ stehende Wert mit dem über FND vorgegebenen (Rück-)Setzwert im „actual“ des FND-Telegrammes gemäß C4 übereinstimmt (Befehl ausgeführt) oder noch nicht (Ausführungsbestätigung später in Form einer Spontanmeldung) „last_save“ enthält den unmittelbar vor dem Setzvorgang gültigen Zählwert.

Beispiel: Die LZ möchte nach einem Austausch des mechanischen Zählwerkes den Zählerstand „1.2345“ vorgeben. Diese Information („actual“) ist Bestandteil der Nutzdaten in DP-Tabelle #1. Die LZ sendet somit ein CMD-Telegramm gemäß Operation C4 und erhält im RSP-Telegramm gemäß R4. Da der setzbefehl sofort ausgeführt wird, enthält das RSP-Telegramm den bisherigen Zählwert nun im „last_save“, „actual“ enthält 1.2345 und das gesetzte „cc“-Bit zeigt den Setzvorgang an.

Werden über die gesamte Laufzeit der LZ alle Datensätze mit gesetztem „cc“-Bit gespeichert, so kann jederzeit der Gesamtzählwert (z.B. Verbrauch von Wasser, Gas, Wärme, Strom,...) durch Kumulieren der Differenz zwischen „actual“ und dem „last_save“ des nächsten Datensatzes berechnet werden.

(Für weitergehende Ausführungen zu dieser tabellarischen Übersicht wird auf das Kapitel „Operationsbeschreibung“ verwiesen).

2.4 FND – APDU-Format

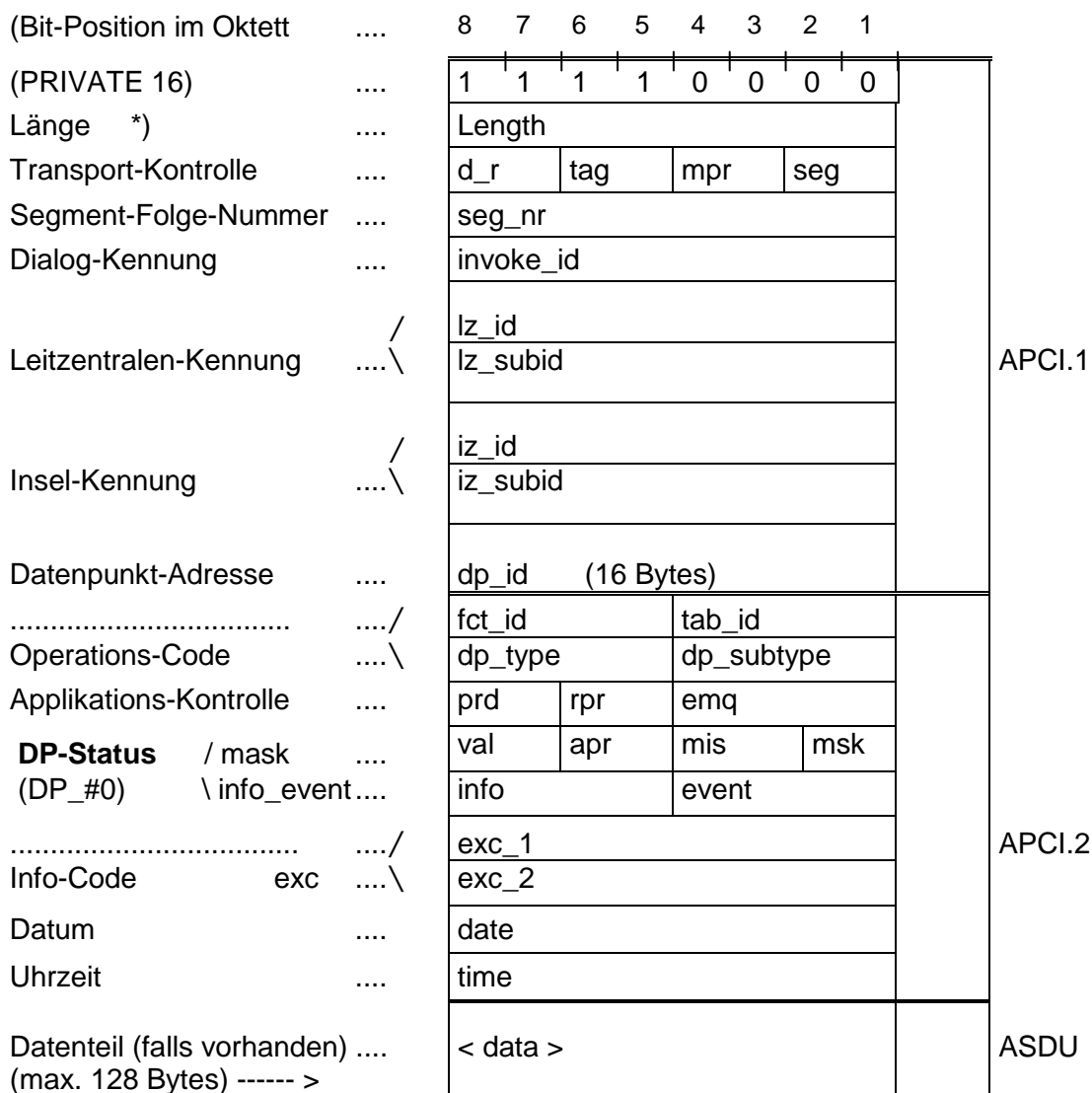
Eine **FND-APDU** (im folgenden auch einfach mit APDU bezeichnet) setzt sich prinzipiell aus zwei Teilstrukturen zusammen:

- einem für alle APDU's gleich strukturierten Vorspann fester Länge, der **APCI** (= Application-Protocol-Control-Information)
- einem (möglicherweise leeren) Datenteil, der ASDU (= Application-Service-Data-Unit).

Die APCI selbst gliedert sich wiederum in zwei Teilstrukturen (APCI 1 und APCI 2), wobei erstere Adress- und Kontrollinformationen bzgl. der inselinternen Weiterleitung an den bzw. von dem betreffenden Datenpunkt, und APCI 2 die eigentlichen DP-spezifischen Informationen enthält.

Länge und Inhalt des Datenteils sind abhängig vom Telegrammtyp, der gewünschten Funktion sowie den Prozessbedingungen im Umfeld des Datenpunkts und der LZ.

Die folgende Abbildung zeigt die Struktur der APDU in schematischer Form.



*) Länge = APDU-Länge-2

Erläuterung der einzelnen Variablen

(PRIVATE 16):

Dieses Oktett entspricht dem X.409-identifizier PRIVAT-USE (16), und steht aus Konformitätsgründen zur Norm RS 511 an erster Stelle.

Länge:

Die Längenangabe, ebenfalls in Übereinstimmung mit X.409, gibt die Anzahl der nachfolgenden, zur APDU gehörenden Oktette an; entsprechend der maximalen Paketlänge von 128 Bytes ist dieser Wert auf 126 beschränkt.

Transport-Kontrolle:

Die Transport-Steuerung setzt sich aus vier 2-Bit-Worten nachstehender Bedeutung zusammen:

„d_r“: (Richtungs- und Antwort-Bit)

„d_r“ (Bits # 8 und # 7) kennzeichnen vier Telegrammtypen, die nach Übertragungsrichtung und Mitteilungscharakter wie folgt unterschieden werden:

a) „d“- („direction“) Bit: Übertragungsrichtung

0: Richtung LZ → IZ

1: Richtung IZ → LZ

b) „r“- („response“) Bit: Antwortkennung

0: Befehls- oder Meldungs-Telegramm

1: Antwort- oder Bestätigungs-Telegramm

Die möglichen „d_r“-Werte, ihre Bedeutung sowie ihre Zuordnung zu den Telegrammtypen sind in nachstehender Tabelle zusammengefasst:

d_r	Tel. Typ	Erläuterung
0 (00'B)	Cmd	„Command“: Befehl der LZ an eine IZ
3 (11'B)	Rsp	„Response“: Antwort einer IZ auf einen Befehl der LZ
2 (10'B)	Usm	„Unsolicited Message“: Spontanmeldung einer IZ an die LZ
1 (01'B)	Ack	„Acknowledge“: Bestätigung eines Usm-Telegramms

Vereinbarung:

Entsprechend der in der dritten Spalte definierten Synonym-Bezeichnungen soll im folgenden auch von **Cmd-/Rsp-/Usm- und Ack-APDUs** (bzw. -Telegrammen) gesprochen werden.

* **„tag“: (Telegramm-Attribut)**

Zur Kennzeichnung von Telegrammbesonderheiten wird die durch Bit # 5 und # 8 codierte Variable „tag“ wie folgt verwendet:

tag		Erläuterung
0	0	NORMAL-Telegramm (kein Fehler)
0	1	REJECT-Telegramm (Ablauf-Fehler)
1	0	ERROR-Telegramm (syntaktisch/semantischer Fehler)
1	1	(reserviert)

Eine ausführliche Beschreibung des Exception-Handling ist **Anhang F** zu entnehmen.

* **„mpr“: (Nachrichten-Priorität)**

Die vier durch Bit # 3 und # 4 codierbaren, in nachstehender Reihenfolge wachsenden Vorrangstufen

- 0: (normal)
- 1:
- 2:
- 3: (dringlichst)

dienen der Prioritätskennzeichnung der Mitteilung auf Applikationsebene. Die Reaktion hierauf ist projektspezifisch zu vereinbaren.

Bei Rsp bzw. Ack-Telegrammen ist mpr aus dem jeweils zugehörigen Cmd- bzw. Usm-Telegramm unverändert zu übernehmen.

* **„seg“: (Segmentierungs-Kontrolle)**

Bits # 1 und # 2 stellen einen Mechanismus zur Segmentierung solcher Nachrichten zur Verfügung, die aufgrund ihrer Länge nicht mehr in ein Datenpaket passen. Codierung und Bedeutung sind wie folgt:

seg		Erläuterung
0	0	Keine Segmentierung)
1	1	Segmentierung: erstes Element der Folge
1	0	Segmentierung: weiteres, aber nicht letztes Element der Folge)
0	1	Segmentierung: letztes Element der Folge

Anmerkung zur Segmentierung:

- a) Bei segmentierten Nachrichten sind die Teiltelegramme in der korrekten Reihenfolge abzusenden.
- b) Mit Ausnahme des letzten sind alle Teiltelegramme vollständig (also unter Ausnutzung der maximalen Paketlänge von 128 Bytes) mit Daten zu belegen.
- c) Der Datenteil des End-Telegramms darf nicht leer sein.
(Einzelheiten hierzu sind in Abschnitt 2.9, „Segmentierung“, zu finden).

Segment-Folge-Nummer:

„seq_nr“ hat bei unsegmentierten Telegrammen stets den Wert 0. Im Segmentierungsfall dient „seq_nr“ (in mit 0 beginnender und lückenlos aufsteigender Reihenfolge) der Kennzeichnung der Anordnung, in welcher die einzelnen Teletelegramme wieder zu reassemblieren sind.

(Einzelheiten hierzu sind in Abschnitt 2.9, „Segmentierung“, zu finden).

Dialog-Kennung:

Die „invoke_id“ stellt einen Flußkontroll- und Dialogsicherungsmechanismus dar, dessen Beschreibung in Anhang B, „Prozedur-Elemente“, gegeben ist.

Leitzentralen-Kennung:

Die Leitzentralen-Kennung setzt sich aus den zwei Teilfeldern „lz_id“ und „lz_subid“ wie folgt zusammen:

„lz_id“:

Die „lz_id“ ist eine zur Kennzeichnung der LZ projektspezifisch zu vergebende Nummer im Bereich 0 – 255.

„lz_subid“:

Die „lz_subid“ ist eine von der LZ für Cmd-Telegramme vergebene optionelle Kennung (Defaultwert = 0), welche die Zuordnung dieses Telegramms zum entsprechenden internen auftraggebenden Prozess herstellt. Dieser Wert ist von der LZ ohne Interpretation in das korrespondierende Rsp-Telegramm zu übernehmen.

Bei Usm-Telegrammen kann die „lz_subid“ projektspezifisch bestimmten Nachrichtenkategorien zugeordnet sein, und ist von der LZ ohne Interpretation in das Ack-Telegramm zu übernehmen.

Insel-Kennung:

Die Insel-Kennung setzt sich aus den zwei Teilfeldern „iz_id“ und „iz_subid“ wie folgt zusammen:

„iz_id“:

Die Kennzeichnung der einzelnen DDC_Inseln geschieht durch die eindeutige Vergabe von Nummern im Bereich 0 – 255.

„iz_subid“:

Analog zur lz_subid ist die iz_subid eine von der LZ für Usm-Telegramme vergebene Nummer, welche die Zuordnung dieses Telegramms zum entsprechenden internen auftraggebenden Prozess herstellt und im korrespondierenden Ack-Telegramm von der LZ ohne Interpretation übernommen wird.

Bei Cmd-Telegrammen kann die iz_subid projektspezifisch bestimmten Nachrichtenkategorien zugeordnet sein, und ist von der Insel ohne Interpretation in das Rsp_Telegramm zu übernehmen.

Datenpunkt-Adresse:

Die Datenpunkt-Adresse wird durch einen bis zu maximal 16 Zeichen langen ASCII-String repräsentiert, welcher gemäß den Applikationsanforderungen die memotechnische Bezeichnung eines (realen wie virtuellen) Datenpunkts (symbolische Adresse) darstellt. Der zugelassene Zeichenvorrat entspricht hierbei dem Charakter-Set des Internationalen Alphabets Nr. 5, eingeschränkt auf die Spalten 2 - 7 (20'H < Char <= 7F'H), jedoch einschließlich des Zeichens NUL (0/0).

Datenpunkt-Adressen mit weniger als 16 Zeichen werden durch binäre Nullen (NUL) auf 16 Bytes ergänzt (Ende-Kennung).

Operations-Code:

Der in den vier Halbbytes „fct_id“, „tab_id“, „dp_type“, „dp_subtype“ verschlüsselte Operations-Code ist detailliert in den Abschnitten 2.5.1, „Befehle“ und 2.5.2, „Meldungen“, beschrieben.

Applikations-Kontrolle:

Das Applikations-Kontroll-Byte setzt sich aus zwei Bit-Worten und einem 4-Bit-Wort nachstehender Bedeutung zusammen:

*** „prd“: (Durchgriffs-Priorität):**

Bits # 7 und # 8 codieren die Durchgriffspriorität, die im Zusammenwirken mit der Sperrpriorität „rpr“ einen effektiven Arbitrierungsmechanismus zwischen inselinternen und LZ-spezifischem Zugriff auf Datenpunkte darstellt. Die durch die „prd“-Bitkombination darstellbaren Werte von 0 – 3 sind in aufsteigender Reihenfolge wachsenden Zugriffsprioritäten zugeordnet, wobei 0 dem „Normalanwender“, und der Wert 3 dem besonders privilegierten Systemmanager zugeordnet ist.

*** „rpr“-Bitgruppe: (Sperr-Priorität)**

Bits # 5 und # 6 codieren die bereits eben erwähnte Sperrpriorität, die im Zusammenwirken mit der Durchgriffspriorität als Arbitrierungsmechanismus zwischen LZ- und Insel-seitigem DP-Zugriff dient. Die durch die Bitkombination darstellbaren Werte 0 – 3 sind in aufsteigender Reihenfolge wachsenden Sperrprioritäten zugeordnet, wobei 0 allgemeinen, und der Wert 3 (sofern zugelassen) LZ-Befehlen ausschließlichen Zutritt sichert.

(Nähere Details zu diesem Thema sind in Abschnitt 2.8, „Zugriffskontrolle“, zu finden).

*** „emq“: (Ereignis-Anzeige/Modifikationskontrolle)**

Der in Bit # 1 – Bit # 4 verschlüsselte „event-modification-qualifier“ verkörpert eine Doppelrolle als Ereignis-Indikator bzw. Modifikations-Kontrolle, und ist in den Abschnitten 2.5.1, „Befehle“ und 2.5.2, „Meldungen“, detailliert beschrieben.

DP-Status:

Der Datenpunkt-Status, bestehend aus den beiden Bytes „mask“ und „info-event“, wird detailliert in den Abschnitten 2.5.2, „Meldungen“ und 2.8, „Zugriffskontrolle“, erläutert.

Info-Code:

Bedeutung und Verwendung des aus zwei Bytes „exc_1“, „exc_2“ zusammengesetzten Info-Codes „exc“ wird in Anhang F beschrieben.

Datum/Uhrzeit:

Datum und Uhrzeit werden in der 6-Byte-Form

Jahr	(0 – 99)
Monat	(1 – 12)
Tag	(1 – 31)
Stunden	(0 – 23)
Minuten	(0 – 59)
Sekunden	(0 – 59)

in dieser Reihenfolge eingetragen.

Bei Rsp- bzw. Ack-Telegrammen wird Datum/Uhrzeit aus dem jeweils zugehörigen Cmd- bzw. Usm-Telegramm unverändert übernommen.

Datenteil:

Die Beschreibung des Datenteils, welcher maßgeblich durch den Operations-Code bestimmt wird, ist Gegenstand der Abschnitte, 2.5.1 „Befehle, und 2.5.2, „Meldungen“

2.5 Operationsbeschreibung

Die detaillierte Beschreibung der Abläufe beim Senden von Befehlen (Cmd/Rsp) und Meldungen (Usm/Ack), inkl. der Modifikationskontrolle, entnehmen Sie bitte der FND-Spezifikation.

2.6 Beschreibung des DP-Status

Die detaillierte Beschreibung des DP-Status entnehmen Sie bitte der FND-Spezifikation.

2.7 Sperren von Ereignismeldungen

Das in der Datenpunkt-Tabelle DP_#0 enthaltene „info_event“-Byte setzt sich aus einem „info“- und einem „event“-Halbbyte zusammen. Letzteres spezifiziert die DP-spezifischen Ereignis-Quellen (z.B. Wertänderung, Grenzwertverletzung, Geber-/BTA-Störung). Die Änderung eines dieser Bits kennzeichnet das Ereignis und führt zu einer spontanen Meldung.

In gewissen Situationen kann es sinnvoll sein, die Meldungen einer bestimmten Kategorie zu unterdrücken.

Beispiel: Die Betriebsmeldungen eines Brenners sind nur während der Einfahrphase oder nach der Beseitigung einer Störung relevant. Während des größten Teils der Betriebszeit belasten diese Meldungen nur die Datenübertragung, die Speicherkapazität der Datenbank und verlängern die Auswertungszeiten.

Die Ausblendung von Ereignismeldungen erfolgt durch Modifikation des Meldungskontroll-Feldes „msk“ in der Datenpunkt-Tabelle DP_#0 in abgestufter Weise wie folgt:

mis/msk	Effekt
0	Freigabe aller Ereignismeldungen
1	Ausblendung der Betriebs-Meldungen; Gemeldet werden: Geber- und BTA-Störungen
2	Ausblendung der Betriebs- und BTA-Störungs-Meldungen; Gemeldet werden: Geber- Störungen
3	Ausblendung aller Ereignismeldungen

Die detaillierte Beschreibung der Ausblendung von Ereignismeldungen entnehmen Sie bitte der FND-Spezifikation.

2.8 Zugriffskontrolle

Zugriffe auf Datenpunkte können von zwei Seiten erfolgen:

- BTA-intern

- Von der LZ bzw. GA-Knoten

Während bei lesenden Zugriffen keine Konflikte entstehen, ist bei modifizierenden Zugriffen (z.B. Schaltprogramm) evtl. ein Entscheidungsmechanismus erforderlich, der die Bearbeitung konkurrierender Zugriffe, ggf. abhängig von wechselnden Zuständen, steuert.

Beispiel: In einer Schule soll bei ausreichender Helligkeit zu bestimmten Zeiten verhindert werden, dass Schüler das Flurlicht aus Spaß einschalten (Tastersperre).

Die detaillierte Beschreibung der Zugriffskontrolle entnehmen Sie bitte der FND-Spezifikation.

2.9 Segmentierung

Die Segmentierung betrifft nur die Transferpunkte. Diese werden bei der LHM nur in Ausnahmefällen verwendet.

Die detaillierte Beschreibung der Abläufe beim Senden von Befehlen (Cmd/Rsp) und Meldungen (Usm/Ack), inkl. der Modifikationskontrolle, entnehmen Sie bitte der FND-Spezifikation.

3 Ergänzungen zur FND-Spezifikation

Der nachfolgende Text wurde aus dem in Papierform vorliegenden Original von Wolfgang Schenk übernommen. Dabei wurde die Nummerierung der Abschnitte bzw. Abbildungen und Verweise so angepasst bzw. ergänzt, dass diese auch innerhalb dieses Dokumentes genutzt werden können.

Der Ausschreibung wird als Datenübertragungsprotokoll die DIN V 32 735,

„Firmenneutrales Datenübertragungsprotokoll (FND) für die Gebäudeleittechnik“

in einer den speziellen Projektanforderungen angepassten erweiterten Version zugrundegelegt.

Die Spezifikationserweiterung beinhaltet i.w. die folgenden **drei Zusatzfunktionen**:

1. **Ermöglichung von Spontanmeldungen aufgrund projektspezifisch vereinbarter (inselinterner) Triggermechanismen**
2. **Einführung eines neuen Datenpunkttyps „Karteipunkt“ (DP-Kategorie 7) zur tabellarischen Darstellung und Manipulation „spreadsheet-geeigneter“ Informationen**
3. **Aufnahme der physikalischen Dimension „Zeitpunkt“ in diversen praxisorientierten Varianten.**

3.1 Zu (1): Spontanmeldungen (te-Bit)

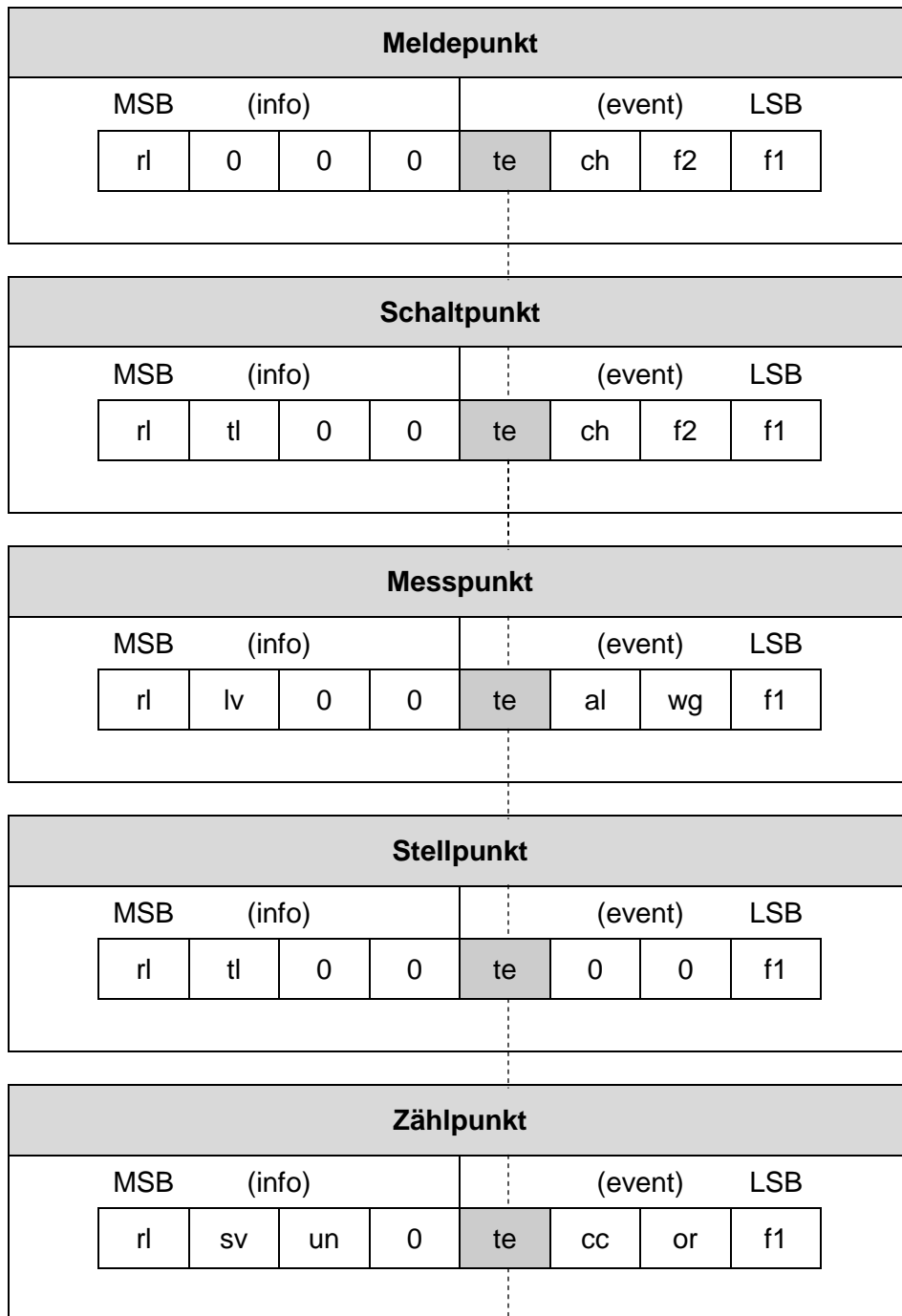
Abb.3-1 zeigt in Form einer schematischen Übersicht die Ergänzung des info_event-Bytes von Melde-, Schalt-, Meß-, Stell- und Zählpunkt um das sog. te-(„triggering-event“)Bits, für welches folgende Regelung gelten soll:

- **te** (triggering-event):
ein gesetztes te-Bit signalisiert das Eintreten eines projektspezifisch vereinbarten Ereignisses (z.B. den Ablauf eines inselinternen implementierten (zyklischen) Timers), welches die Übertragung der Datenpunkt-Tabelle DP_1 impliziert.

Das Rücksetzen des Bits im info_event_Byte der DP-Tabelle DP_0 erfolgt aufgrund der LZ-seitigen Bestätigung der Meldung bzw. bei Ausbleiben derselben nach einem intern gesetzten Timeout; der Rücksetzvorgang selbst wird nicht als Ereignis gewertet.

Solange te auf 1 steht, bewirken weitere te-bezogene Ereignisse keine Meldung.

Sind Ereignisse dieser Art (IZ-oder LZ-seitig) ausgeblendet, (d.h. $msk \geq 1$ oder $mis \geq 1$), so hat te während dieser Zeit den Wert 0.



„triggering event“:
 0: kein auslösendes Ereignis
 1: auslösendes Ereignis

Abb. 3-1: Ergänzung des „info_event“-Bytes (DP_0) um das „te“-Bit

3.2 Zu (2): DP_Typ 7 (Karteipunkt)

3.2.1 Kartei-Koordinatensystem

Die Motivation zur Einführung eines sogenannten Karteipunkts leitet sich u.a. aus dem Bedarf nach mittelfristiger (inselinterner) Speicherung zyklisch erfasster Daten, aber etwa auch nach Parametrierbarkeit von Prozessen durch Vorgabe ganzer Sollwertreihen ab.

Wie **Abb. 3-2** zeigt, basiert die Grundkonzeption des Karteipunkts auf der tabellarischen Datenorganisation bekannter Spreadsheet-Programme wie „Lotus -1-2-3“ oder „MS-Excel“. Eine **Tabelle** besteht hierbei aus **Zellen**, deren jede der Informationsspeicherung eines bestimmten, projektspezifisch vereinbarten Datentyps dient.

Die Definition von „**Zeilen**“ und „**Spalten**“ einer Tabelle ergibt sich in naheliegender Weise aus der horizontalen bzw. vertikalen Betrachtungsweise des Zellen-Rechteckrasters, die einer „**Kartei**“ aus der sequentiellen Aneinanderreihung von Einzeltabellen.

Der FND-Karteipunkt geht somit von einem „Kartei-Koordinatensystem“ aus, innerhalb dessen jede Zelle (oder synonym: jeder **Punkt**) durch Angabe

- der Tabelle (Index „**i**“)
- der Zeile (Index „**j**“)
- der Spalte (Index „**k**“)

eindeutig festgelegt (**adressiert**) ist, die Bezeichnung für eine Zelle ist i.a. einfach **[i,j,k]**.

Der virtuelle (d.h. von der Implementierung abstrahierende) Charakter des Karteipunkts wird dadurch unterstrichen, dass Zeilen-, Spalten- und Tabellenindizes zunächst nur durch die Genauigkeit der IEEE 754-Floating-Point-Codierung ¹⁾ beschränkt sind; eine eventuell projektspezifische Limitierung von Zeilen-, Spalten- bzw. Tabellen-Indizes (etwa als Ausdruck realer Kapazitätsbeschränkungen) ist hierbei selbstverständlich unbenommen.

Die Übertragung und Manipulation von Zellen erfolgt in sogenannten **Blöcken** – quaderartigen Karteiausschnitten, wie **Abb.3-3** veranschaulichen soll. Die Auswahl eines Blocks erfolgt in einfacher (und eindeutiger) Weise durch Angabe der Koordinaten $[i_0, j_0, k_0]$ des „ersten Eckpunktes“ (d.h. i_0 = erster Zeilenindex, j_0 = erster Zeilenindex und k_0 = erster Spaltenindex des Blocks), und der Koordinaten $[i_1, j_1, k_1]$ des „letzten Eckpunktes“ (d.h. i_1 = letzter Tabellenindex, j_1 = letzter Zeilenindex und k_1 = letzter Spaltenindex des Blocks). $[i_0, j_0, k_0]$ und $i_1, j_1, k_1]$ sind die **Blockdefinitions**punkte.

¹⁾ Die maximale exakt darstellbare Ganzzahl liegt in der Größenordnung von $2^{23} \approx 8$ Millionen

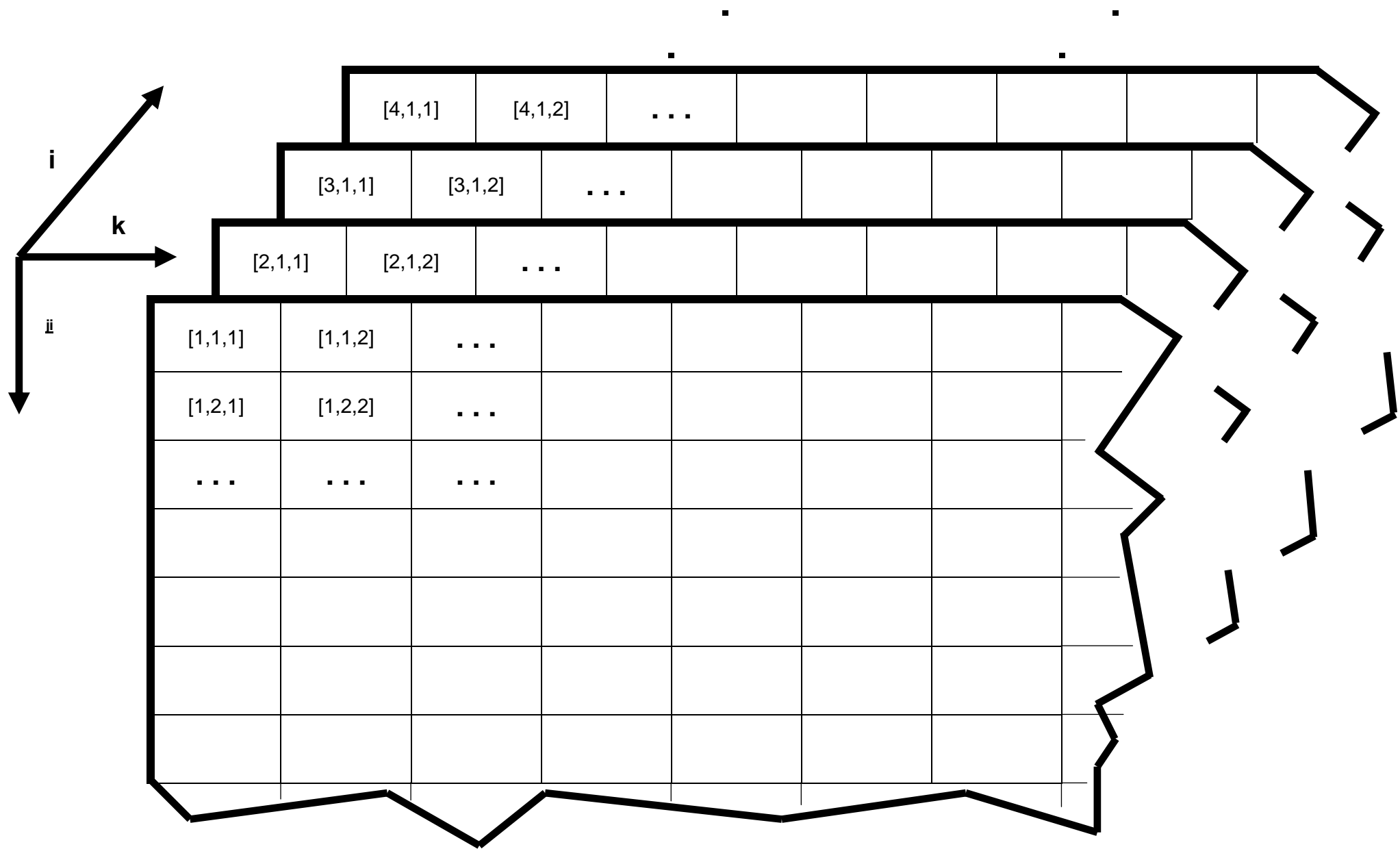


Abb. 3-2: Kartei-Koordinatensystem $[i,j,k]$

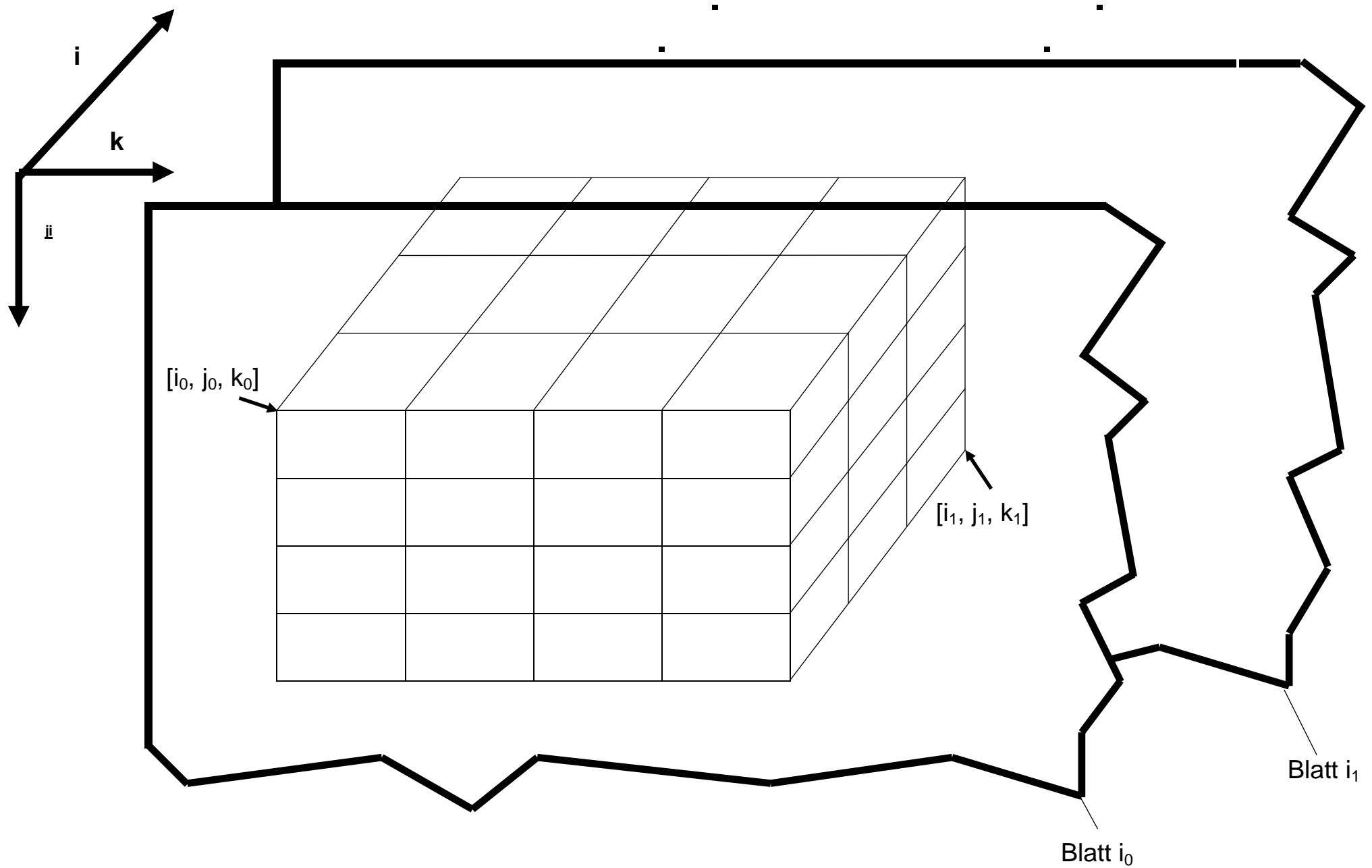


Abb. 3-3: Bereichsauswahl innerhalb einer Kartei („Block“)

3.2.2 Karteipunkt-Tabellen

Die prinzipielle Struktur der vier Datenpunkttabellen DP_0, DP_1, DP_2 und DP_3 des Karteipunkts ist **Abb. 3-4a** und **Abb. 3-4b** zu entnehmen.

Variablenbeschreibung:

DP_0

- **mask:** (P_1):

(Zur Beschreibung des Oktetts siehe DIN V 32 735, Teil 1, Abschnitt 5.3 „Sperrungen von Meldungen“ bzw. intern unter Abschnitt 2.7); anzumerken ist, dass zur Unterdrückung aller Karteimeldungen bereits die Bedingung $mis > 0$ oder $msk > 0$ ausreicht (Unterdrückungsstufe = „Betriebsmeldung“).

Erläuterungen zu „**info_event**“:

- **req** (request):

Da die Ausführung eines Lesebefehls auf einen Karteiblock aufgrund eventuell erforderlicher inselinterner Rückfragen eine gewisse Zeit in Anspruch nehmen kann, wird das Ergebnis gegebenenfalls als spontanes Ereignis gemeldet. Ein weiterer Grund für die Ergebnisübermittlung als spätere Meldung kann darin bestehen, dass die Antwort – selbst bei Segmentierung – nicht in einem einzigen Rsp-Telegramm unterzubringen ist, und deshalb vom Applikationsprogramm vorab in Teilabschnitte geeigneter Länge zerlegt werden muss (s. hierzu auch die späteren Anmerkungen zu Segmentierung). Bei der Beschreibung dieses Bits wird daher im folgenden zwischen einer Meldung und der Antwort auf einen Lesebefehl unterschieden.

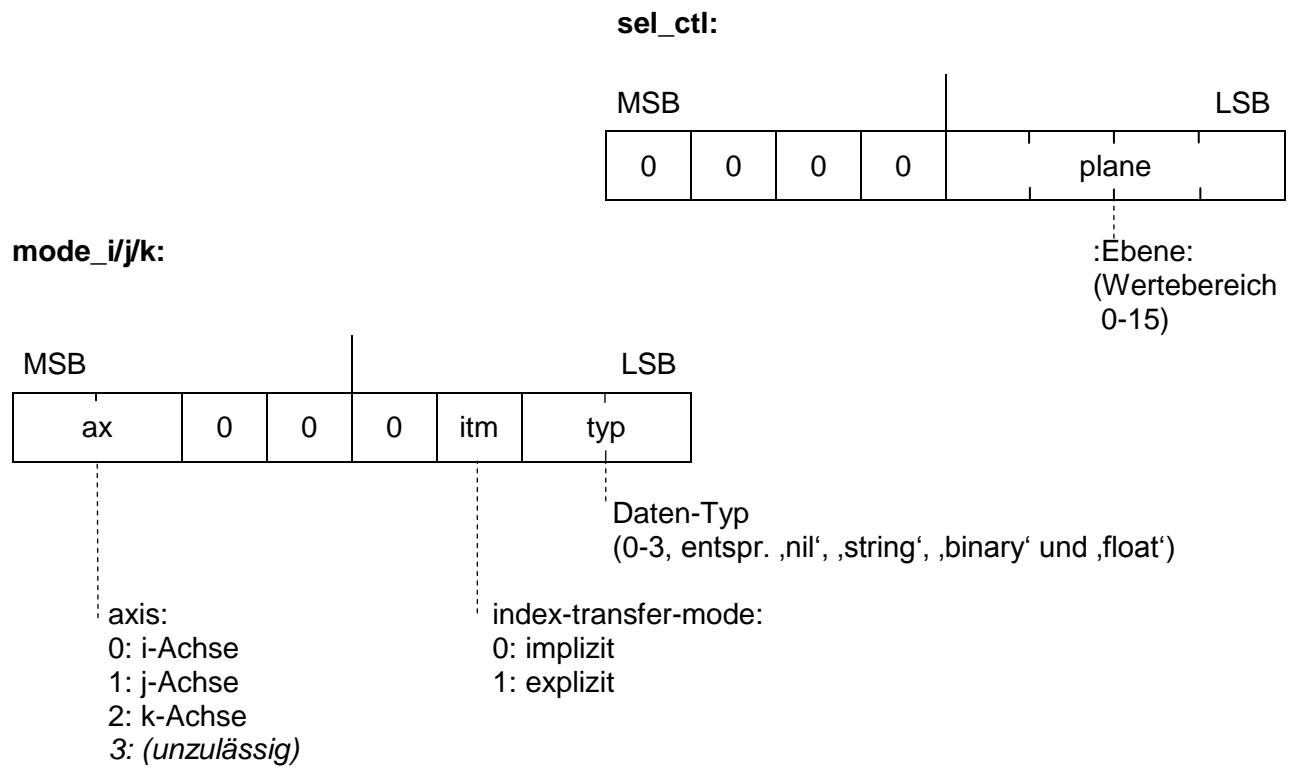
1. Antwort auf einen Lesebefehl:

Bei der Antwort auf einen Lesebefehl ist der Wert des req-Bits davon abhängig, ob der Befehl sofort oder erst mit einer gewissen Verzögerung ausgeführt werden kann. Explizit heißt dies:

- a) Ein Wert von 1 im req-Bit des Rsp-Telegramms bedeutet, dass die aktuellen Werte des selektierten Kartenblocks zur Verfügung stehen und in DP_2 des Response-Telegramms enthalten sind. Das Rücksetzen des Bits erfolgt implizit durch das Absenden des Rsp-Telegramms.
- b) Ein Wert von 0 im req-Bit des Rsp-Telegramms bedeutet dagegen, dass der Lesebefehl empfangen wurde, sich zum Zeitpunkt des Absendens des Rsp-Telegramms aber noch in der Aktualisierungsphase des Karteiblocks befindet. Die Beendigung der Aktualisierung wird durch Spontanmeldung (s.2.) signalisiert.

2. Meldung:

Der Abschluss eines durch einen Lesebefehl initiierten Aktualisierungsvorgang des Karteiblocks wird durch das req-Bit signalisiert, welches somit als 1 übertragen wird; das Rücksetzen des Bits im info_event-Byte der DP-Tabelle 0 erfolgt aufgrund der LZ-seitigen Bestätigung der Meldung bzw. bei Ausbleiben derselben nach einem intern gesetzten Timeout; der Rücksetzvorgang selbst wird nicht als Ergebnis gewertet.



DP_3

mode_i	mode_j	mode_k
atb_dim_i	atb_dim_j	atb_dim_k
org_i	org_j	org_k
scal_i	scal_j	scal_k
min_i	min_j	min_k
max_i	max_j	max_k

Abb. 3-4b: Datenpunkttabellen des Karteipunkts

blk (block-transmission):

Eine weitere Ereignisquelle für die Übermittlung eines (projektspezifisch festgelegten) Karteiblocks kann ein inselintern implementierter Mechanismus (z.B ein Timer) sein, der den entsprechenden Block zur LZ übermittelt. Rücksetzen des blk-Bits erfolgt aufgrund der LZ-seitigen Bestätigung der Meldung bzw. Ausbleiben derselben nach einem intern gesetzten Timeout.

Solange blk auf 1 steht, bewirken weitere blk-bezogene Ereignisse keine Meldung. Sind Ergebnisse dieser Art (IZ- oder LZ-seitig) ausgeblendet (d.h. msk > 0 oder mis >0), so hat blk während dieser Zeit den Wert 0.

rl (remote/local):

Die Zustände Fern-/Vor-Ort-Bedienung werden durch das rl-Bit unterschieden.

DP_1 („Auswahl-Steuerung“):

sel_ctl (selection-control):

Dieses „Auswahl-Kontroll-Byte“ enthält, wie **Abb 3-4b** zeigt, als Parameter die durch 4 Bit dargestellte Variable „plane“ (Wertbereich 0 - 15), deren Bedeutung weiter unten im Abschnitt 3.2.4, „**Kartei-Informationsebenen**“, erläutert wird.

first_i:

Tabellenindex des ersten Blockdefinitionspunktes in IEEE-754-Darstellung. (Näheres zu dieser Codierung s. DIN V 32735, Teil 1, Seite 18 oben).

first_j:

Zeilenindex des ersten Blockdefinitionspunktes in IEEE-754-Darstellung.

frist k:

Spaltenindex des ersten Blockdefinitionspunktes in IEEE-754-Darstellung.

last_i:

Tabellenindex des zweiten Blockdefinitionspunktes in IEEE-754-Darstellung.

last_j:

Zeilenindex des zweiten Blockdefinitionspunktes im IEEE-754-Darstellung.

last_k:

Spaltenindex des zweiten Blockdefinitionspunktes in IEEE-754-Darstellung.

Ein Wert von ‚FFFFFFFF‘H für first_i,..., last_k bedeutet „nicht angegeben“ (implizit).

DP_2:

block:

DP_2 ist als „virtuelle dynamische“ Tabelle aufzufassen, d.h. sie wird bei größerem Ausmaß nicht statisch im Speicher gehalten, sondern vielmehr gemäß des jeweils zugrundeliegenden Karteausschnitt (Blocks) dynamisch zusammengestellt werden. Die serielle Übertragung eines Blocks erfolgt durch Verkettung der einzelnen selektierten Zellen gemäß (der für Matrizen üblichen) Anordnungsregel $k \Rightarrow j \Rightarrow i$, also

„Spaltenindex vor Zeilenindex vor Tabellenindex“

Die eindeutige Dekodierung eines übertragenen Zellblock-Strings wird

- a) durch Angabe der Blockdefinitionspunkte
[first_i, first_j, first_k] und [last_i, last_j, last_k]
- b) durch die im Zellen-Status (bzw. Zellen-Modifikations-Kontrollbyte) enthaltene Information bezüglich des Datentyps (und damit der Länge des nachfolgenden Zelleninhalts)

sichergestellt.

DP_3:

Die Konfigurationstabelle DP_3 (charakterisiert durch **tab_id = ,0100'B**, vgl. DIN V 32 735, Teil 1, S.35 oben bzw. intern unter Abschnitt 2.3) enthält die (projektspezifisch festgelegten) charakteristischen Parameter des [i,j,k] - Koordinatensystems wie folgt:

- **mode_i,**
mode_j,
mode_k:

Kontrollbyte, welches zusammen mit atb_dim_i/j/k die Dimension des Achsenindex (z.B. Schaltzustände (0,1,2,3), oder Wochentage (Montag - Sonntag) bestimmt.
Zur Bedeutung der Variablen „itm“ siehe Erläuterungen zu „Used-Block“, Tabelle 3-3.

- **atb_dim_i,**
atb_dim_j,
atb_dim_k:

binäres Attribut („atb_id“) bzw. physikalische Dimension (dimension“) gemäß sort-descriptor in mode i/j/k des Achsenindex i/j/k; die Kombination sd=0 und atb_dim_i/j/k = 0 kennzeichnet den dimensionslosen Fall.

An dieser Stelle sei nochmals darauf hingewiesen, dass als Indexmenge zur Unterscheidung der einzelnen Tabellen, Zeilen und Spalten prinzipiell auch die Gesamtheit der reellen Zahlen herangezogen werden kann, dies ist z.B. dann von Vorteil, wenn als Tabellenindex etwa eine Zeitangabe oder ein Meßwert verwendet wird („Hash-Index“).

- **org_i,**
org_j,
org_k:

Durch das Zahlentripel [org_i, org_j, org_k] wird die „Ursprungszelle“ (= Nullpunkt) des [i,j,k]-Koordinatensystems fixiert.

- **scal_i,**
scal_j,
scal_k:

Die Skalierung („Auflösung“) der Werte der Tabellenachse wird durch den nichtnegativen Wert scal_i als inkrementelle Einheit des Tabellenindex festgelegt; sinngemäß ist scal_j für die Zeilen-, und scal_k für die Spaltenachse zu interpretieren. So bedeutet etwa ein Wert von scal_i = 1 eine **ganzzahlige** Tabellenindizierung, und scal_k= 10 eine Spaltenindexschrittweite von 10 Einheiten. Sind als Tabellen-/Zeilen-/Spaltenindizes **alle** reellen Zahlen zugelassen, so wird hierfür der Skalierungswert $scal_{i/j/k} = 0$ verwendet.

- **min_i,**
min_j,
min_k,
max_i,
max_j,
max_k:

- Die projektspezifisch festgelegten Größen min_i/j/k und max_i/j/k markieren den prinzipiell zur Verfügung stehenden [i,j,k]-Adressraum als Block mit den Definitionspunkten [min_i, min_j, min_k) und [max_i, max_j, max_k]; die Werte „-∞“ (für min_i/j/k) und „+∞“ (für max_i/j/k) sind zugelassen, und werden in der IEEE 754-Darstellung durch das Bitmuster ‚7F800000‘H bzw. ‚FF800000‘H repräsentiert.

Das in **Abb.3-5** gegebene Beispiel soll die Konzeption der Blockauswahl verdeutlichen. So soll der ausgewählte Karteiausschnitt an den Zellen

[6,2,3], [6,2,4], [6,2,5]	(Tabelle 6)
[6,3,3], [6,3,4], [6,3,5]	(Tabelle 6)
[7,2,3], [7,2,4], [7,2,5]	(Tabelle 7)
[7,3,3], [7,3,4], [7,3,5]	(Tabelle 7)

bestehen.

Diese Auswahl spiegelt sich in DP_1 durch die Angabe der „Blockeckpunkte“ [6,2,3] und [7,3,5].

DP_2 enthält die in den einzelnen Zellen enthaltenen Informationen in verketteter Form gemäß der „Spaltenindex-vor-Zeilenindex-vor-Tabellenindex“-Regel.
(Ende Beispiel)

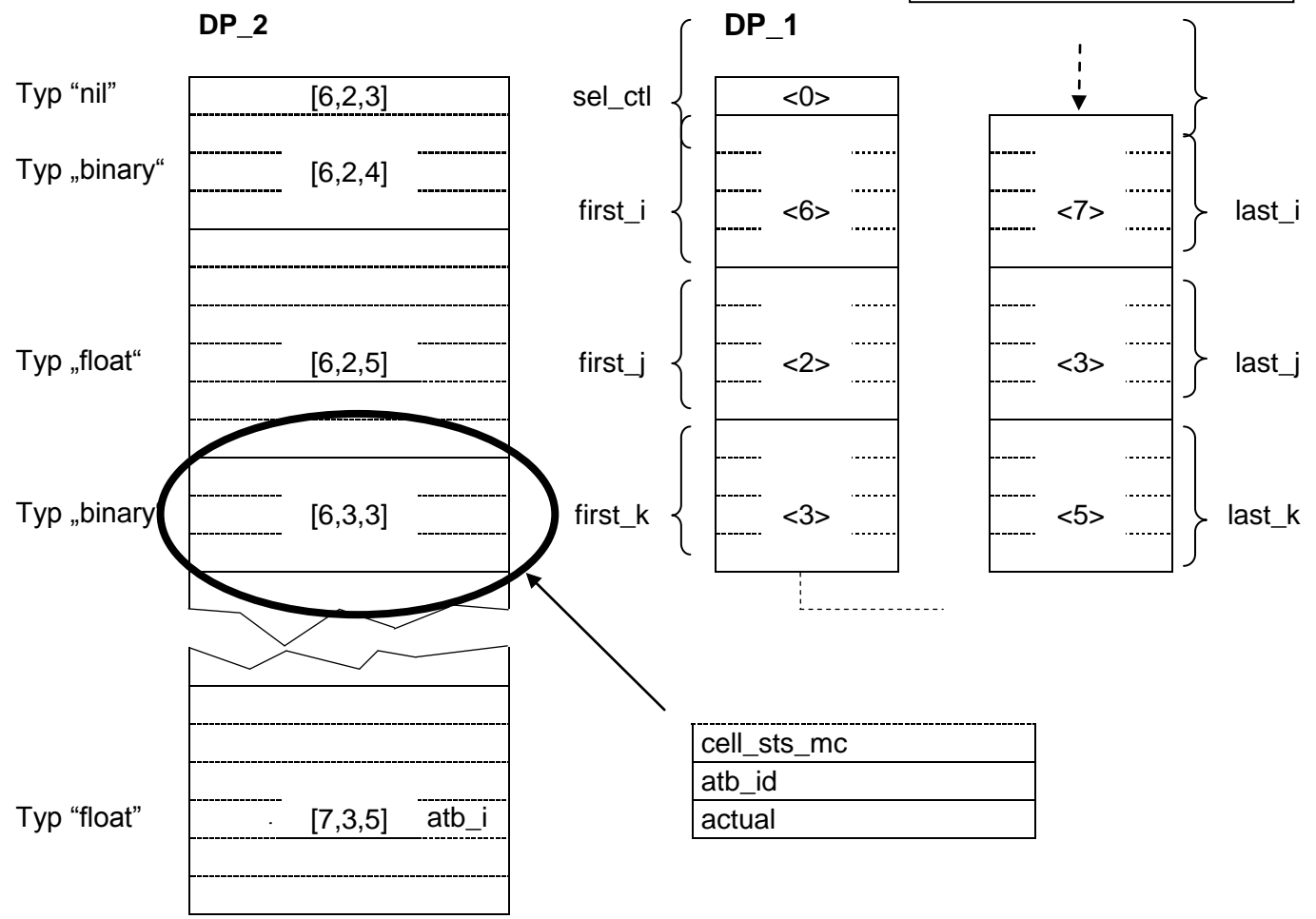
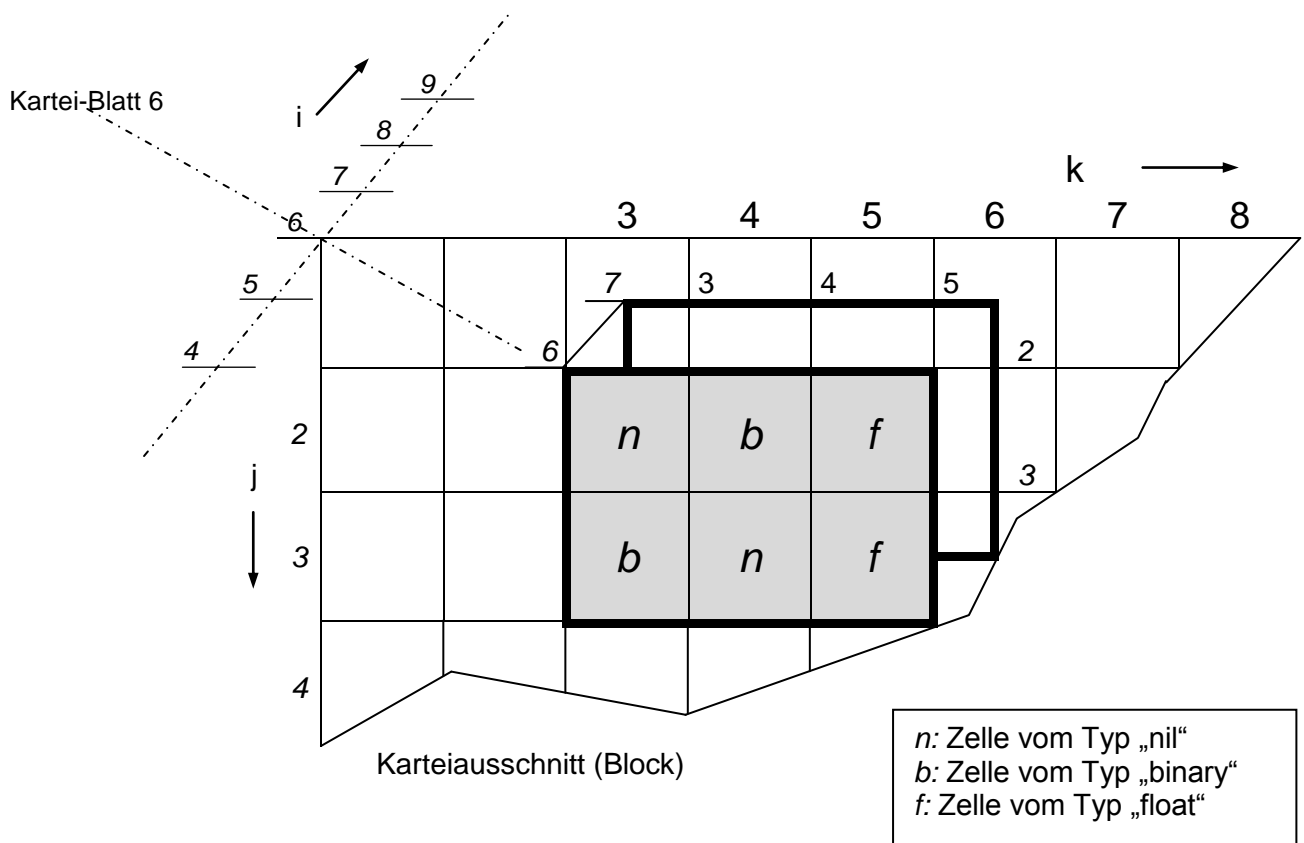


Abb. 3-5: Beispiel der Übertragung eines Zellblocks zur LZ

3.2.3 Zellenstruktur

In **Abb. 3-5** ist als Ausschnitt von DP_2 die Zelle [6,3,3] explizit dargestellt, welche ein mit „cell_sts_mc“ bezeichnetes Oktett ausweist. Dieses „Zellen-Status/Modifikations-Kontrollbyte“ ist – wie der Name bereits andeutet – mit einer Doppelfunktion belegt, die durch den Telegrammtyp (Cmd bzw. Rsp/Usm/Ack) bestimmt ist.

In der Rolle als **Zellen-Status** ist cell_sts_mc beim Rsp/Usm/Ack-Telegramm Bestandteil folgender Konzeption:

Jeder Zelle eines Karteipunkts sind zwei Größen zugeordnet:

- A) der Zellen-Status
- B) der Zellen-Wert

Der Zellen-Status enthält folgende Informationen:

- ein Konstantenkennzeichnungsbit „udc“
- die Kennzeichnung des der Zelle zugeordneten Datentyps „typ“.

3.2.3.1 Beschreibung des Zellen-Status

Die Bit-Codierung des Zellen-Status eines Karteipunkts zeigt die nachstehende **Abb.3-6**

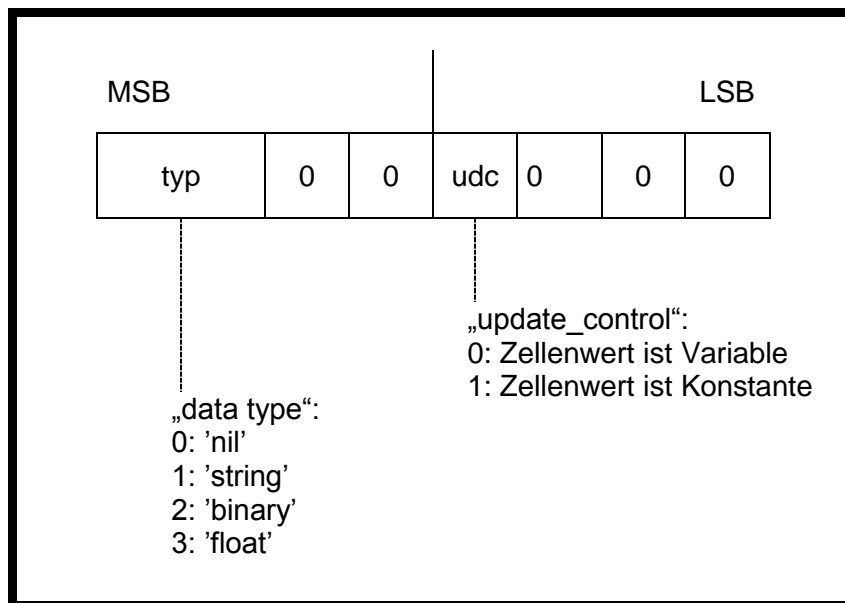


Abb. 3-6: Struktur des Zellen-Status-Bytes „cell_sts_mc“ beim Rsp-/Usm- und Ack-Telegramm

Die Bedeutung der beiden Variablen ist wie folgt:

- **udc** (update-control):
ein gesetztes Bit weist den Zelleninhalt als Konstante (d.h. als projektspezifisch festgelegte, nur durch Systemrekonfiguration veränderbare Größe) aus, in welchem Falle cap (da ohne Bedeutung) auf 0 zu setzen ist. Andernfalls ist der Zellenwert seitens LZ oder IZ (bei gegebener Zugriffsberechtigung) modifizierbar.
- **typ** (data-type):
diese Bitkombination weist den mit der Zelle verbundenen (projektspezifisch festgelegten) Datentyp gemäß nachstehender Tabelle 3-1 aus.

Tabelle 3-1: Codierung der Datentypen

typ		Datentyp
0	0	,nil'
0	1	,string'
1	0	,binary'
1	1	,float'

Hierbei bedeuten:

1. Datentyp **,nil'**: Zelle nicht belegt („leer“)
2. Datentyp **,string'**: Zelle enthält Oktett-Folge
3. Datentyp **,binary'**: Zelle enthält binären Zustand
4. Datentyp **,float'**: Zelle enthält physikalischen Wert

Mit einer Zelle des Typs **,nil'** ist – wie die Bezeichnung bereits nahelegt – prinzipiell kein Wert verbunden; leere Zellen entstehen i.d.R. als „Lücken“ in der Anordnung der „Nutzinformationszellen“.

Eine Zelle des Typs **,string'** enthält eine Sequenz von maximal 255 Oktetten, der ein Oktett mit der entsprechenden Längenangabe vorangestellt ist. Die leere Zeichenkette (**,Null-String'**) besteht folglich nur aus einem Oktett des Inhalts 0.

Die Codierung der Datentypen ‚**binary**‘ und ‚**float**‘ erfolgt in der ‚FND-üblichen‘ Weise durch kombinierte Angabe von

- ‚atb_id‘ + ‚actual‘ (binäres Attribut wie bei DP_1, Meldepunkt)
- ‚dimension‘ + ‚actual‘ (physikalische Dimension wie bei DP_3, Meßpunkt)

Die Werte ‚FF‘H bzw. ‚FFFFFFFF‘H stehen für unbesetzte (ungültige) binäre Attribute bzw. physikalische Größen.

Eine Zelle des Typs

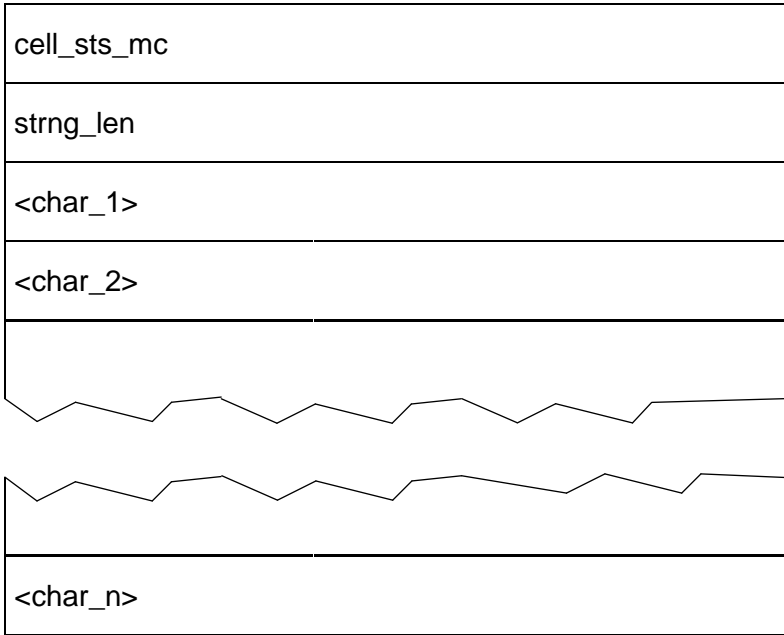
- ‚nil‘
- ‚string‘ des Inhalts „Nullstring“
- ‚binary‘ des Inhalts ‚FF‘H
- ‚float‘ des Inhalts ‚FFFFFFFF‘H

heißt **unbenutzt** („unused“).

Das für die einzelnen Datentypen im DP_2 verwendete Format zur Darstellung der Zelleninhaltsinhalte bei Usm-/Ack- und Rsp-Telegrammen ist in der umstehenden **Abb.3-7** gezeigt.



Typ des Zellenwerts = ,nil'

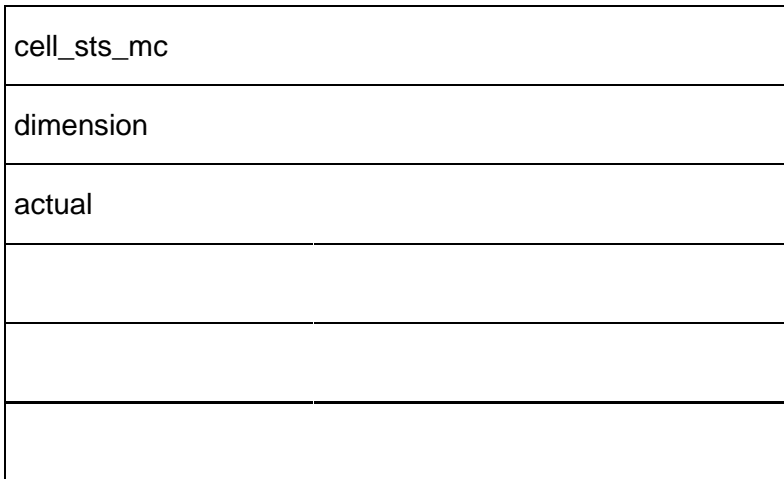


Typ des Zellenwerts = ,string'

n (=strng_len) Bytes



Typ des Zellenwerts = ,binary'



Typ des Zellenwerts = ,float'

Abb. 3-7: Zellenwert-Codierung in DP_2 beim Rsp/Usm/Ack-Telegramm

3.2.3.2. Zellen-Modifikations-Kontrolle

Wie bereits eingangs zu Abschnitt 3.2.3 erwähnt, kommt den mit „**cell_sts_mc**“ bezeichneten Oktetten in den Zellinformationen der Datenpunktstabelle DP_2 eine Doppelrolle zu, die durch den Telegrammtyp (Cmd bzw. Rsp/Usm/Ack) bestimmt ist.

Die Bedeutung der einzelnen Variablen von „**cell_sts_mc**“ in der Funktion als **Zellen-Modifikations-Kontrolle** ist beim Cmd-Telegramm wie folgt (s. **Abb.3-8**):

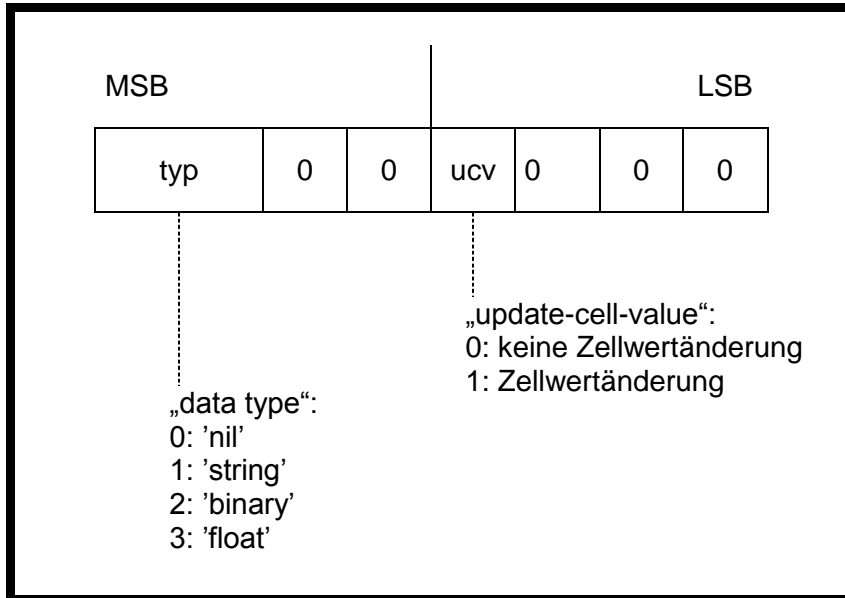


Abb. 3-8: Struktur des Zellen-Modifikations-Kontrollbytes „**cell_sts_mc**“ beim Cmd-Telegramm

- **ucv** (update cell value):
ucv steuert einen Zellenmodifikationsbefehl nach Maßgabe des – gemäß dem Datentyp in den im Cmd-Telegramm nachfolgenden Oktetten codierten – Zellenwertes.
- **typ** (data-type
(s. Beschreibung beim Zellen-Status)

Die Zellenwert-Codierung in DP_2 ist beim Cmd-Telegramm mit Ausnahme

- a) der Interpretation von „**cell_sts_mc**“ als Modifikations-Kontrollbyte (anstelle des Zellen-Status)
- b) des Umstands, dass die Oktette zur Codierung von ‚string‘-, ‚binary‘- und ‚float‘-Werten dem „**cell_sts_mc**“-Byte nur bei $ucv \neq 0$ angehängt werden

identisch mit der des Rsp/Usm/Ack-Telegramms (vgl. **Abb.3-7**).

3.2.4 Kartei-Informationsebenen

Der Begriff „Kartei-Koordinatensystem“ (zusammen mit den Achsenindizes „i“, „j“, „k“, und dem „[i,j,k]-Zellenadressierungsschema“) wurde bereits in Abschnitt 3.2.1 eingeführt. So ist der (durch die Datenpunkttabelle DP_3 in Größe und Rasterung festgelegte „Karteiraum“ als Teilmenge des \mathbf{R}^3) als Adressengrundmenge zu verstehen, dessen Elementen (=„Zellen“) bestimmte Informationen zugeordnet sind (hierfür wird i.a. die Sprechweise „Zelle [i,j,k]“ enthält die Information „l“ verwendet). Zur Darstellung der Information einer Zelle stehen die vier Basis-Datenpunkttypen

- ‚nil‘
- ‚string‘
- ‚binary‘
- ‚float‘

zur Verfügung.

In der Praxis können einer Zelle jedoch durchaus mehrere Informationen unterschiedlichen Charakters zugeordnet sein, die den Rahmen eines bloßen „Zahlenwertes“ deutlich übersteigen. So kann etwa eine solche (nach Art eines „Pascal-Records“ aus den vier oben genannten Basis-Datentypen zusammengesetzte Information) aus folgenden Teilinformationen bestehen:

0. der („eigentliche“) Zellenwert (z.B. eine physikalische Meßgröße)
1. ein Zellenattribut (z.B. ein zusätzlicher Name zur „technischen Zellen-Adresse“ [i,j,k])
2. die „Berechnungsformel“ des Zellenwertes
3. eine „Triggerformel“ zur Codierung eines Zelleneignisses
4. Kommentare (Notizen) bezüglich der Zelle

u.a. mehr.

Zur Unterstützung solcher „vektorwertiger“ Informationen dient als Konzept der Kartei-Informationsebenen, dessen Prinzip in **Abb. 3-9** dargestellt ist. So könnte – anhand des eben gegebenen Beispiels – jeder Zelle auf Ebene 0 der Zellenwert, auf Ebene 1 das Zellenattribut, auf Ebene 2 die Berechnungsformel usw. zugeordnet sein. Die Auswahl der (für Lese- und/oder Modifikationszwecke) gewünschten Ebene erfolgt durch die bereits bei der Beschreibung der Datenpunkttabelle DP_1 erwähnten Variablen „plane“ im „sel_ctl“-Byte.

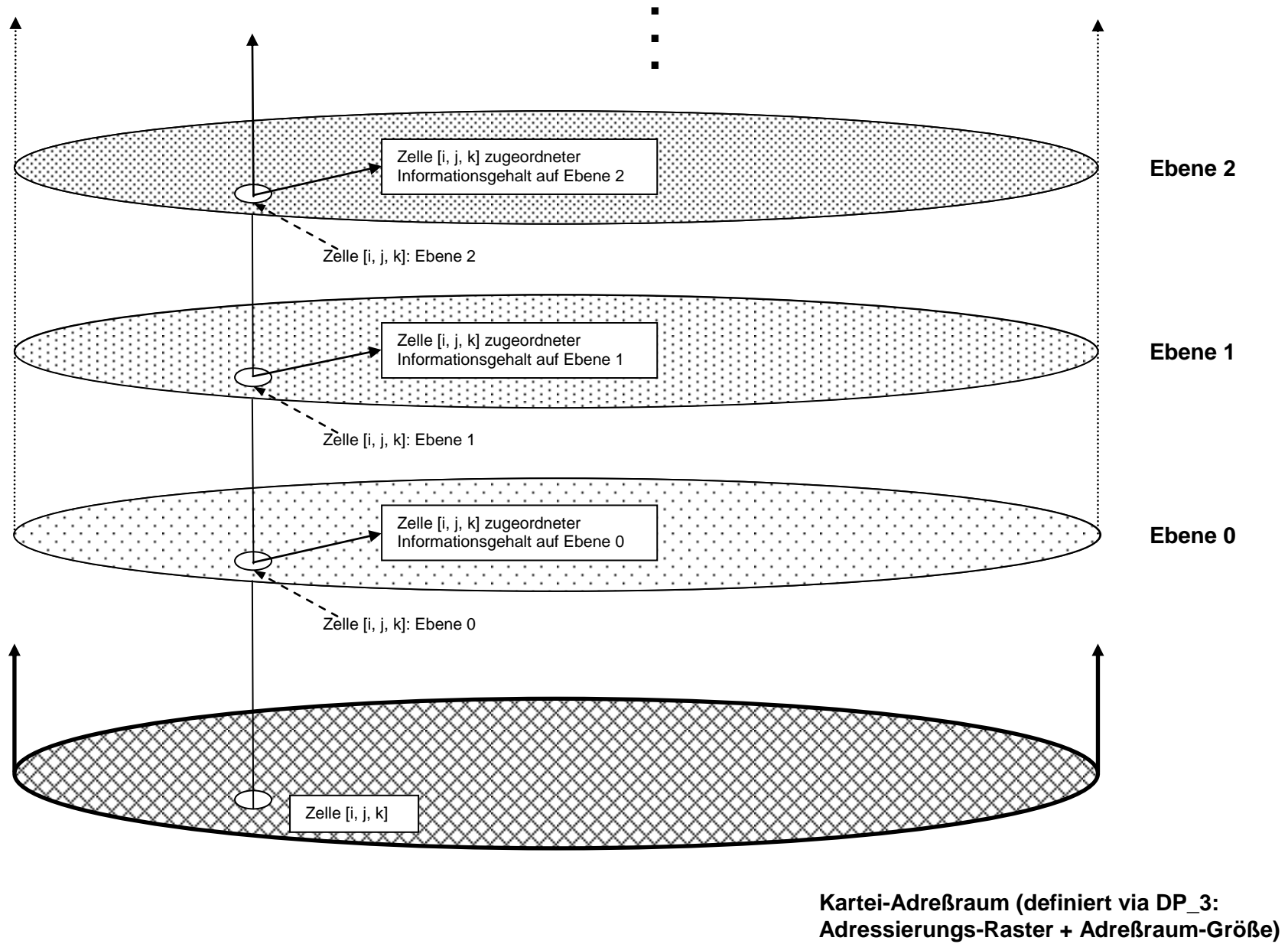


Abb. 3-9: Konzept der Karte-Informationsebenen

3.2.5 Anmerkung zur Segmentierung:

- a) Bei einem segmentierten, DP_2 modifizierenden Cmd-Telegramm enthält das zum Cmd-Endtelegramm gehörende Rsp-Endtelegramm
- bei info_event.req = 1 die (evtl. segmentierte) Antwort auf den Modifikationsbefehl
 - bei info_event.req = 0 nur die letzte Kurzquittung; die Modifikationsbestätigung erfolgt dann - wie schon mehrfach erwähnt – in Form einer Meldung mit info_event.req = 1.
- b) Da die in der DIN V 32 735 festgelegte Beschränkung eines FND-Telegramms auf max. 256 Teiltelegramme (Segmente) aufgrund des (via der „invoke_id“-Variablen) bereits implementierten Flusskontrollmechanismus als unnötige Restriktion betrachtet wird, ist die Segment-Folge-Nummer („seq_nr“) als Zähler modulo 256 zu behandeln.

Eine allgemeine Operationsübersicht des Karteipunkts geben **Abb. 3-9.a** und **Abb. 3-9.b**.

Karteipunkt		fct_id	
d_r	tab_id	,0001'B	,0010'B
„Cmd“	,0000'B	A	B
	,0001'B	C	
	,0011'B	D	E
	,0100'B	F	
„Rsp“	,0000'B	G	H
	,0001'B	I	
	,0011'B	J	K
	,0100'B	L	
„Usm“	,0011'B	M	
	,0100'B	N	
„Ack“	,0011'B	O	
	,0100'B	P	

Schattierter Bereich: nicht anwendbar

Zur Bedeutung der Variablen d_r, tab_id und fct_id siehe DIN V 32 735, Teil 1, Abschnitt 5: „FND-APDU-Format“

Abb. 3-9.a: Operations-Übersicht des Karteipunkts

Operation	Bedeutung	Datenteil
A	DP-Status DP_0 anfordern	(leer)
B	Meldungs-Kontroll-Feld modifizieren	(leer)
C	„Used-Block“-Definition (DP_1) anfordern (siehe „Erläuterung zu C“)	(leer)
D	Block DP_2 (gemäß Auswahl DP_1) anfordern	DP_1
E	Block DP_2 (gemäß Auswahl DP_1) modifizieren	DP_1 ^ DP_2
F	Konfigurationstabelle DP_3 anfordern	(leer)
G	DP-Status DP_0 übermitteln	(leer)
H	Modifikationsbestätigung des Meldungs-Kontroll-Felds	(leer)
I	„Used-Block“-Definition (DP_1) übermitteln (siehe Erläuterung zu C*)	DP_1
J	Block DP_2 zusammen mit Auswahl DP_1 übermitteln	[DP_1 ^ DP_2] ^{*)}
K	Modifikationsbestätigung der DP-Tabelle DP_2 zus. mit Auswahl DP_1	[DP_1 ^ DP_2] ^{*)}
L	DP-Tabelle DP_3 übermitteln	DP_3
M	Meldung des Blocks DP_2 zusammen mit Auswahl DP_1	DP_1 ^ DP_2
N	Meldung einer Kartei-Konfigurationsänderung (DP_3)	DP_3
O	Bestätigung der Blockmeldung	(leer)
P	Bestätigung der Kartei-Konfigurationsänderung	DP_3
^{*)} Falls Datenteil leer (info_event.req = 0) erfolgt die Übermittlung von DP_1 ^ DP_2 (mittels info_event.req = 1) später optional in Form einer Spontanmeldung		

Abb. 3-9.b: Symbol-Erklärung des Karteipunkts

Erläuterung zu C:

Zur Feststellung des gegenwärtig in der betreffenden Kartei-Ebene „kleinsten Blocks, außerhalb dessen alle Zellen unbenutzt sind“ kann seitens der LZ ein Lesebefehl auf eine virtuelle Tabelle namens „Used-Block“ des Formats DP_1 erteilt werden, auf welchen hin diese übermittelt wird (s. **Abb. 3-10**); mittels der so erhaltenen Information kann die LZ z.B. – nach Auslesen der betreffenden Nutzdaten – die Löschung (und damit die „Freigabe“) des betreffenden Zellenbereichs veranlassen.

Da jedoch insbesondere bei Achsenindizes mit reellwertiger Skalierung (d.h. $scal_{i/j/k} = 0$) Unsicherheiten über die zwischen $first_used_{i/j/k}$ und $last_used_{i/j/k}$ liegende Anzahl der belegten Zellen bestehen können, besteht die Möglichkeit der „expliziten Indexübertragung“, erkennbar am gesetzten „itm“-Bit der Variablen $mode_{i/j/k}$ in DP_3. Bei dieser Form der Blockübertragung werden nur die „relevanten“ Zellbereiche transferiert, d.h. die ineffiziente Übermittlung überwiegend leerer Zellen vermieden.

Zur Illustration zeigt **Abb.3-11** einen (fiktiven) Karteipunkt, mittels dessen etwa die im Verlauf einer Woche von 55 Sensoren registrierten Störmeldungen (einschl. exaktem Zeitbezug) erfasst werden sollen. Eine mögliche Achsenzuordnung ist somit:

- **i** - Achse: kontinuierlicher Zeitbezug innerhalb des aktuellen Tages (8⁰⁰ bis 17⁰⁰)
- **j** - Achse: Tag innerhalb der aktuellen Woche (Montag - Freitag)
- **k** - Achse: laufende Nummer der Sensoren

Der Inhalt jeder Zelle enthält jeweils eines der beiden binären Attribute „NORMAL“ bzw. „ALARM“. Interessiert nun am Ende einer Woche die Gesamtheit der in diesem Zeitraum aufgelaufenen Störmeldungen, so könnte seitens der Leitzentrale als Block der gesamte in Abb.3-11 gezeigte Bereich abgerufen werden. Selbst bei einer zeitlichen Auflösung von nur 10 Minuten ergäbe dies jedoch ein Übertragungsvolumen von $(9 \times 6) \times 5 \times 55 \approx 15.000$ Zellwerten, von denen nur fünf (A-E) tatsächlich relevant sind; ein Effektivitätsgrad von $5/15.000 \approx 0,3 \%$ kann sicherlich nicht als zufriedenstellend bezeichnet werden.

Wesentlich effizienter ist demgegenüber die ausschließliche Berücksichtigung der Zeitpunkte i_A, i_B, \dots, i_E , zu denen die einzelnen Störmeldungen jeweils registriert wurden; das Übertragungsvolumen reduziert sich hierdurch bereits auf $5 \times 5 \times 55 \approx 1.400$ Zellenwerte.

Da auch der in diesem Fall resultierende Effizienzgrad von $5/(5 \times 5 \times 55) = 0,35 \%$ noch verbesserungsbedürftig erscheint, wird sinnvollerweise auch eine selektive Übertragung der relevanten Sensoren k_A, k_B, \dots, k_E - mit einem Nutzeneffekt von $5/(5 \times 5) = 20 \%$ - in Erwägung gezogen werden.

DP_1

sel_ctl
first_used_i
first_used_j
first_used_k
last_used_i
last_used_j
last_used_k

Abb. 3-10: „Used Block“ des Karteipunkts

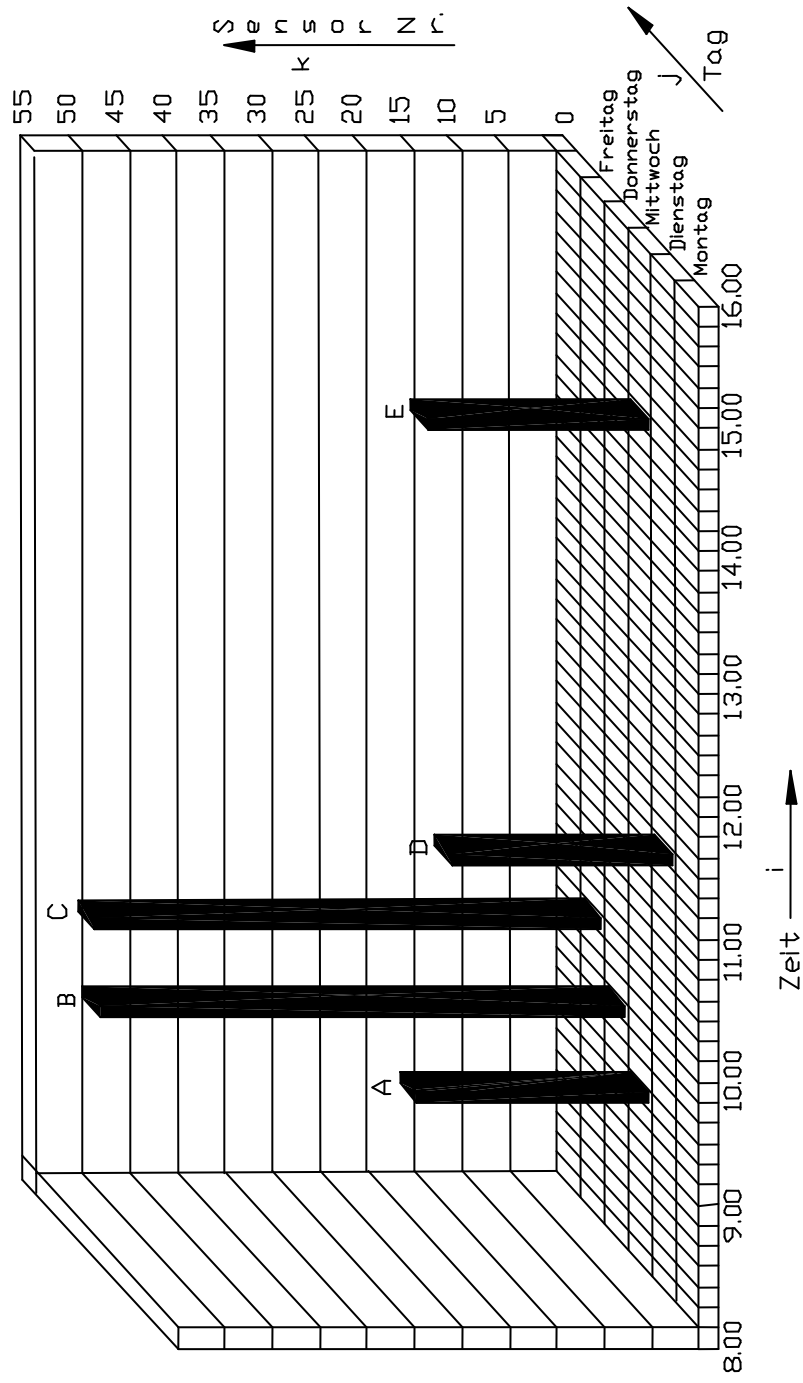


Abb. 11: reellwertige Achseneinstellung (Beispiel)

Die „ökonomischste“ Lösung schließlich besteht in der exklusiven Übertragung der Zellen $[i,j,k]_A, [i,j,k]_B, \dots, [i,j,k]_E$.

Zu beachten ist allerdings, dass bei diesem Übertragungsmodus der mit einer festen Tabellen-, Zeilen- und Spaltenanzahl verbundene Vorteil - einfache Zuordnung der im übertragenen Block enthaltenen Zellen zu deren jeweiliger Position innerhalb der Kartei – (partiell) entfällt: folgt im „impliziten Transfermodus“ die Zuordnung einer Block-Zelle zur zugehörigen $[i,j,k]$ -Position der „Spalten-vor-Zeilen-vor-Tabellenindex“-Regel, so sind bei selektiver Übertragung evtl. fehlende (d.h. nicht implizit rekonstruierbare) Index-Informationen im Block selbst mitzuliefern.

Dies geschieht dadurch, dass

- bei $mode_i.itm = 1$: dem Tabelleninhalt der Tabellenindex
- bei $mode_j.itm = 1$: dem Zeileninhalt der Zeilenindex
- bei $mode_k.itm = 1$: der Spaltenindex dem Zelleninhalt

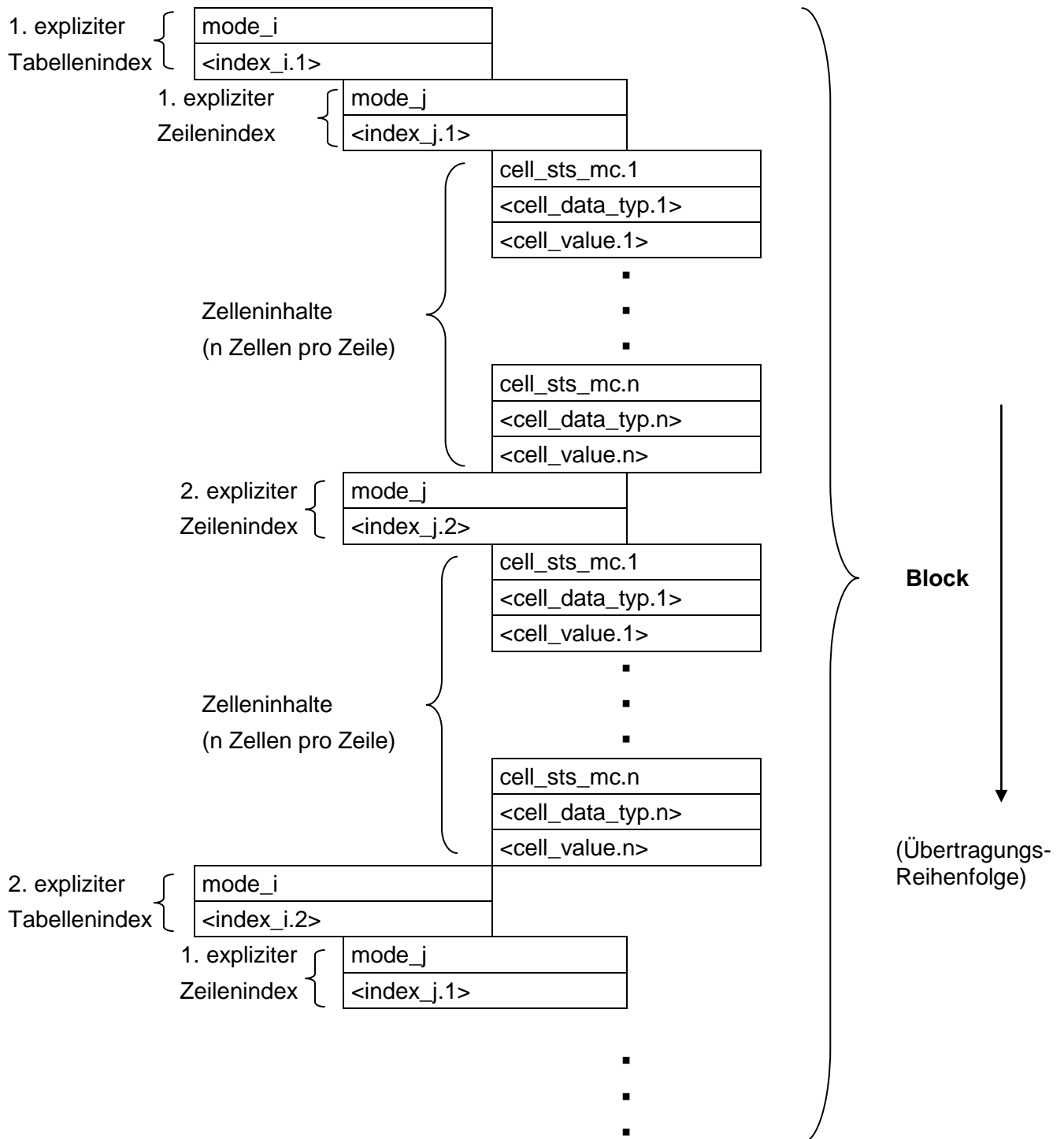
bei der sequentiellen Blockübertragung vorangestellt wird

Abb. 3-12 soll diesen Sachverhalt schematisch veranschaulichen.

Als weitere Konsequenz sind zur Sicherstellung der eindeutigen Adressenzuordnung bei selektiver Zellbereichsübertragung die $i/j/k$ -Übertragungsmodi nicht beliebig wählbar, sondern es bestehen zwischen $mode_i.itm$, $mode_j.itm$ und $mode_k.itm$ die in der nachstehenden Tabelle gezeichneten Zusammenhänge:

Tabelle 3-3: itm-Kombinationen ($mode_i/j/k$)

	mode_i	mode_j	mode_k
	0	0	0
	1	0	0
itm	1	1	0
	1	1	1
andere Kombinationen: illegal			



Anmerkung:

- a) Spaltenzahl (= n) für alle Zeilen und Tabellen konstant
- b) mode_i.itm = 1 (explizite Indexangabe, da Tabellen selektiert)
- mode_j.itm = 1 (explizite Indexangabe, da Zeilen selektiert)
- mode_k.itm = 0 (implizite Indexangabe, da Spaltenzahl (= n) konstant)

Abb. 3-12: Explizite Indexangabe (Beispiel, schematisch)

3.3 Zu (3): Physikalische Größe „Zeitpunkt“

Die für eine Vielzahl von Anwendungen nützliche Codierung diverser Formen des Begriffs „Zeitpunkt“ zeigt die **Abb. 3-13**.

Für Code = 160 (Tag, absolut seit 01.01.1900) ist anzumerken, dass aufgrund der eingeschränkten Genauigkeit der IEEE 754-Floating-Point-Darstellung nur eine „Rasterschärfe“ ≥ 10 Minuten realisierbar ist. Erweist sich dies für spezielle Anwendungen als ungenügend, so besteht eine Lösungsmöglichkeit z.B. in der Verwendung zweier assoziierter Zellen, deren erste mittels Code 160 den absoluten Tag, und deren zweite mittels Code 167 (oder 168) den exakten Zeitpunkt innerhalb dieses Tages fixiert.

Code	Bedeutung	Wertebereich ^{*)}	Beispiel	
			Zeitpunkt	Codierung ^{**)}
160	Tag, absolut seit 01.01.1900	[0, ∞)	01.01.1900, 06:00	0,25
			15.03.1993, 12:00	34.042,50
161	Tag, relativ zum aktuellen Jahr	[0, 366)	1. Januar, 12:00	0,50
162	Kal.-Woche, relativ zum aktuellen Jahr	[0, 53)	2. KW, Do., 12 Uhr	1,50
163	(res.)	(res.)	(res.)	(res.)
164	Tag, relativ zum aktuellen Monat	[0, 31)	02., 18 Uhr	1,75
165	Tag, relativ zur aktuellen Woche	[0, 7)	Mo, 12 Uhr	0,5
			Do., 18 Uhr	3,75
166	Stunde relativ zur aktuellen Woche	[0, 168)	Di, 18 Uhr	42,0
167	Stunde relativ zum aktuellen Tag	[0, 24)	18:30 Uhr	18,5
168	Minute relativ zum aktuellen Tag	[0, 1440)	12:50	770,0
169	Minute relativ zur aktuellen Stunde	[0, 60)	50 Minuten	50,0
<p>*) Intervallschreibweise [x, y): x gehört zum Wertebereich, y nicht mehr.</p> <p>***) Nachkommastellen sind bezogen auf die Einheiten Stunde, Tag bzw. Woche zu interpretieren, d.h. 0,5 bedeutet</p> <p>bei Stundenbezug: 30 Minuten der angefangenen Stunde</p> <p>bei Tagesbezug: 12 Uhr Mittag des Tages</p> <p>bei Wochenbezug: Donnerstag, 12 Uhr der Woche</p>				

Abb. 3-13: Codierung von Zeitpunkten als physikalische Größe

4 Verwendung der Karteipunkte im GA-Knoten

Der nachfolgende Text wurde dem „DV-technischen Feinkonzept des GA-Knoten“ mit Stand vom 22.9.1995 entnommen. Ergänzende Kodierungen für Verbindungs-Art (vgl. 4.3.2) und Datenfelder in Schaltprogrammen (vgl. unter 4.4.6.1) wurden eingearbeitet.

Zur Realisierung von anwendungs- bzw. benutzerspezifischen Datenstrukturen bietet die "FND-Spezifikation Version 1.0" den frei definierbaren Transferpunkt (FND-Datenpunkttyp 0) an. Dieser stellt einen universell einsetzbaren "Transportrahmen" für definierbare Datenstrukturen dar. Da wesentliche FND-Mechanismen (z.B. Zugriffskontrolle, Maskierung, ...) genutzt werden sollen, wurde in den "Ergänzungen zur FND-Spezifikation DIN V 32735" der FND-Datentyp 7 (Karteipunkt) spezifiziert (Vgl. Anlage).

In den folgenden Abschnitten werden die Datenstrukturen definiert, die mittels des "Karteipunktes" transportiert werden. FND-intern wird die **Datenstruktur in einer einzigen Spalte vom Typ "string"** zeichenweise abgebildet, d.h. nur die Gesamtstruktur läßt sich über den FND-Spaltenindex adressieren (d.h. stets gilt $k=1$), jedoch nicht die einzelnen Spalten innerhalb der Struktur. Für die Übertragung der Karteipunkte wird die Segmentierung gemäß FND-Spezifikation dahingehend erweitert, daß die Segmentnummer ("seq_nr") zyklisch als Zähler modulo 256 definiert wird. Das Ende des Karteipunktes wird durch die Segmentierungs-Kontrolle "seg" = 01 gekennzeichnet.

4.1 Blockformat

Die in der FND-Spezifikation definierten Datenpunkt-Typen 1 bis 5 (Melde-, Schalt-, Meß-, Stell- und Zählpunkt) haben keine separat zugeordnete Meßzeit, d.h. die im Header.2 angegebene Übertragungszeit (Datum/Zeit) wird gleichzeitig als Meßzeit verwendet. Sollen mehrere Werte eines solchen Datenpunkt-Typs unabhängig von der Meßzeit in einem Block übertragen werden, so muß die Meßzeit zusätzlich zugeordnet werden. Gleichzeitig muß eine Selektion des auszulesenden Datenbereiches über die Meßzeit definierbar sein. Dieses erfolgt über die Bereichsauswahl für den Karteipunkt.

Generell müssen für die Blockformate andere Datenpunkt-Adressen verwendet werden, als bei der Übertragung der Einzelwerte, d.h. die Kennzeichnung erfolgt durch **Verwendung der spezifischen Datenpunkt-Adresse "\$DP-TYP_n-oooo"** ("n" = dp_type (Datenpunkt-Typ) und "oooo" Objektkennung) in Verbindung mit dem **Datenpunkt-Typ "7"** unter dp_type im Header.2 und der **Kartei-Informationsebene 0**. Die Datenpunkt-Adresse ist über "n" einem Datenpunkt-Typ fest zugeordnet, d.h. innerhalb dieses Karteipunktes können nur Datenpunkte dieses Typs gemeinsam übertragen werden.

Der DP_2 des Karteipunktes für das Blockformat hat den folgenden Aufbau:

Tabellenindex: Datum der Messung (IEEE-754 mit Dimension 160) wird explizit angegeben (mode_i.itm=1)

Zeilenindex: Zeit der Messung (IEEE-754 mit Dimension 167, bezogen auf das im Tabellenindex angegebene Datum) wird explizit angegeben (mode_j.itm=1)

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = y

<u>Spalte</u>	<u>Byte</u>	<u>Bezeichnung</u>	<u>Format</u>
---------------	-------------	--------------------	---------------

```

1:      1 ... 16   Datenpunkt-Adresse 16 Byte
2:      17 ... 18   DP_0[dp_type]          gemäß FND-Spez. Version 1.0
3:      19         DP_1[dp_type]          gemäß FND-Spez. Version 1.0
4:      ... y      DP_2[dp_type]          gemäß FND-Spez. Version 1.0

```

Zu 1: Datenpunkt-Adresse

Zu 2 ff.: DP_0, DP_1 und DP_2 haben denselben Aufbau wie er für den jeweiligen Datenpunkt-Typ in der FND-Spezifikation Version 1.0 definiert ist, d.h. für

$$y = \text{len}(\text{DP}_0[\text{dp_type}]) + \text{len}(\text{DP}_1[\text{dp_type}]) + \text{len}(\text{DP}_2[\text{dp_type}]) + 16.$$

4.2 Parametrierung von Datenpunkten

Zur Parametrierung von Datenpunkten unterstützt die FND-Spezifikation in der Version 1.0 nur Informationen, die sich auf den Datenpunkt beziehen, nicht jedoch auf die Verarbeitung. Für alle Datenpunkt-Typen gemeinsam werden die folgenden Informationen benötigt. Wie sich der GA-Knoten verhalten soll, wenn eine Meldung erzeugt wird, bestimmt ein **Flag** für das Meldungsverhalten für jeden Datenpunkt. Je nachdem wie die Parameter im Verbindungs-Profil definiert sind, werden die Meldungen im GA-Knoten zwischengespeichert, sofort an die LZH, an einen BWD-GEFAHR oder an einen BWD-EINBRUCH übertragen.

Die Kodierung der Bits 0 und 1 als natürliche Zahl ist wie folgt:

- 0: KEINE sofortige Übertragung an die LZH, d.h. die Meldung wird bis zur nächsten Übertragung gemäß der im Verbindungs-Profil definierten Parameter zwischengespeichert.
- 1: Sofortige Übertragung an die LZH, d.h. gemäß der im Verbindungs-Profil definierten Parameter wird die Verbindung aufgebaut.
- 2: Sofortige Übertragung der Meldung an einen BWD-GEFAHR, d.h. gemäß der im Verbindungs-Profil definierten Parameter wird die Verbindung aufgebaut.
- 3: Sofortige Übertragung der Meldung an einen BWD-EINBRUCH, d.h. gemäß der im Verbindungs-Profil definierten Parameter wird die Verbindung aufgebaut.
- Ist Bit 2 gesetzt, so ist der Datenpunkt ausgeblendet, d.h. sämtliche Ein-/Ausgaben vom bzw. zum Datenpunkt werden ignoriert bzw. unterdrückt. Dieser Parameter wird auch durch die Funktion "sp_abl" und "sp_ebl" für Ereignisschaltprogramme modifiziert.
- Die Bits 3-7 bleiben reserviert für zukünftige Erweiterungen

Mit der Parametrierung eines Datenpunktes wird ein Flag gesetzt um die Protokollierung zu steuern.

Die Protokollierung wird bitweise wie folgt kodiert:

- Bit 0 und 1: Protokollierung von Ereignismeldungen mit folgender Kodierung (analog zum Sperren von Ereignismeldungen; vgl. FND-Spezifikation unter 2.7):
 - 0: KEINE Protokollierung von Ereignismeldungen (d.h. Ausblendung aller Ereignismeldungen)
 - 1: Geber-Störungsmeldungen (d.h. Ausblendung der Betriebs- und BTA-Störungs-Meldungen)
 - 2: Geber- und BTA-Störungs-Meldungen (d.h. Ausblendung der Betriebsmeldungen)
 - 3: Protokollierung aller Ereignismeldungen
- Bit 2: Protokollierung von "actual"-Werten
- Bit 3: reserviert
- Bit 4: Protokollierung der Änderung von "atb_id"- bzw. "dimension"-Werten

- Bit 5: Protokollierung der Änderung von "nominal"-Werten
- Bit 6: Protokollierung der Änderung von Grenzwerten ("alarm/warning_low" und "alarm/warning_high")
- Bit 7: reserviert

4.2.1 Aktivierungszeit der Datenpunkt-Parameter

Wenn ein Datenpunkt parametrisiert wird, kann der Zeitpunkt (Datum/Zeit) festgelegt werden, ab dem der GA-Knoten diesen Parametersatz aktivieren soll. Werden Meldungen zwischengespeichert, wird der zu diesem Zeitpunkt gültige Parametersatz zugeordnet. Der Parametersatz wird erst gelöscht, wenn ein neuer Parametersatz aktiviert ist und keine Meldungen für den dann ungültigen Parametersatz mehr zwischengespeichert sind.

4.2.2 Datenpunkt-Parameter für Melde- und Schaltpunkte

Datenpunkte dieser Typen beschreiben einen binären Zustand, d.h. eine Größe mit einer endlichen Anzahl von Werten (max. 9). Vertreter sind die Gefahr- und Störmeldungen, die i.a. 1-stufige Zustände darstellen (z.B. Normal - Störung). Gemäß FND-Spezifikation werden bis zu 8-stufige Zustände unterstützt (z.B. 8-stufige Schalter). Für Schaltpunkte kann zusätzlich ein Sollwert vorgegeben werden ("nominal" im DP_#1).

Der Zustand der Werte kann sich spontan ändern. Ein Flag für das Meldungsverhalten steuert, ob die aus dem Zustandwechsel abgeleitete Meldung von dem GA-Knoten zur LZH, oder zum BWD übertragen wird. Die Änderung erfolgt dabei in jeder Richtung (zu- und abnehmende Meldungsstufe), wobei man bei den 1-stufigen Datenpunkten den Wechsel umschreibt, indem man z.B. für eine Störung sagt "Störung KOMMT" bzw. "Störung GEHT". Dabei reicht die Unterscheidung der Zustände für KOMMT und GEHT auch für mehrstufige Datenpunkte, wobei der **GEHT-Zustand** stets mit dem Index 0 gleichgesetzt wird, d.h. keines der Bits in "actual" ist gesetzt (vgl. "FND-Spezifikation" unter 2.3.1 auf Seite 37). Der **KOMMT-Zustand** repräsentiert alle anderen Meldungsstufen, d.h. eines der Bits in "actual" ist gesetzt.

Treten Zustandswechsel in zu kurzen Zeitabständen auf, so kann dieses ein Anzeichen für einen Geberfehler oder eine falsche Parametrierung des Datenpunktes sein. Wird ein solches **Flimmern** nicht erkannt und behoben, so kann es zu einem unnötigen Aufblähen der dynamisch gespeicherten Daten kommen.

Dazu werden im Datensatz der Zeitpunkt für die letzten, d.h. vorherigen Zustandwechsel auf GEHT und KOMMT vermerkt, wobei diese beiden zusammen mit dem aktuellen Wert aus dem Parametersatz ein Tripel bilden, das den Verlauf der letzten GEHT- und KOMMT-Zustände, d.h. der letzten Periode, beschreibt. Geprüft wird die Dauer des GEHT-, des KOMMT- und der Summe aus beiden gegen eine hinterlegte minimale Zeitspanne. Wird diese für einen der Werte unterschritten, so wird ein Zähler inkrementiert. Erreicht der Zähler einen maximalen Wert, wird im Parametersatz der Status "Geber gestört" gesetzt (vgl. DP_#0 "info_event" Bit "f1" in der "FND_Spezifikation" unter 2.3.1 auf Seite 36), was abhängig vom definierten Meldungsverhalten z.B. zu einer Meldung an die LZH führt. Zusätzlich kann anhand eines Flags entschieden werden, ob weitere Meldungen zu diesem Datenpunkt bearbeitet und gespeichert werden sollen, obwohl der Status "Geber gestört" gesetzt ist.

4.2.3 FND-Konvention für Parametriersatz von Melde- und Schaltpunkten

Generell werden für die Parametrierung dieselben Datenpunkt-Adressen verwendet, wie für das Blockformat ("**\$DP-TYP_n-oooo**" mit "n" = dp_type (Datenpunkt-Typ) und "oooo" Objektkennung), d.h. sie werden allein durch Verwendung der **Kartei-Informationsebene 1** unterschieden.

Der DP_2 des Karteipunktes für den Parametriersatz der Melde- und Schaltpunkte hat den folgenden Aufbau:

Tabellenindex: Aktivierungszeit (IEEE-754 mit Dimension 160) wird explizit angegeben (mode_i.itm=1). Ist der Wert mit < 0, vorbesetzt, so erfolgt die Aktivierung sofort.

Zeilenindex: Reserviert für zukünftige Erweiterungen. Momentan stets 1.0 explizit angegeben (mode_j.itm=1).

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = y

Sp.	Byte	Bezeichnung	Format
1:	1...16	Datenpunkt-Adresse	16 Bytes
2:	17	Flag für Meldungsverhalten	1 Byte
3:	18	Flag für Protokollierung	1 Byte
4:	19	Dimension zu 5:	1 Byte
5:	20...23	min. Dauer GEHT	IEEE-754 mit Dimension aus 4:
6:	24	Dimension zu 7:	1 Byte
7:	25...28	min. Dauer KOMMT	IEEE-754 mit Dimension aus 6:
8:	29	Dimension zu 9:	1 Byte
9:	30...33	min. Dauer Periode	IEEE-754 mit Dimension aus 8:
10:	34	max. Anzahl Flimmern	1 Byte
11:	35	Flag, ob weitere Bearbeit.	1Byte
12:	36...	DP_0[dp_type]	gemäß FND-Spez. Version 1.0
13:	...	DP_1[dp_type]	gemäß FND-Spez. Version 1.0
14:	... y	DP_2[dp_type]	gemäß FND-Spez. Version 1.0

Zu 1: Datenpunkt-Adresse

Zu 2: Verhalten des GA-Knoten, sobald für den Datenpunkt eine Meldung erzeugt wird.

Zu 3: Protokollierungsverhalten des GA-Knoten, sobald für den Datenpunkt eine Meldung erzeugt bzw. einer der Datenpunkt-Parameter modifiziert wird.

Zu 4: Zeitbezogene Dimension zu Punkt 5: gemäß FND-Spezifikation Anhang E (12 = Sekunde, 13 = Minute, 14 = Stunde, 15 = Tag).

Zu 5: Minimale Anzahl von Zeiteinheiten aus Punkt 4:, die der GEHT-Zustand aufrecht erhalten bleiben muß, um nicht als Flimmern interpretiert zu werden. Wird der Wert 0 eingetragen, wird dieser Zustand nicht überwacht.

Zu 6: Zeitbezogene Dimension zu Punkt 7: gemäß FND-Spezifikation Anhang E (12 = Sekunde, 13 = Minute, 14 = Stunde, 15 = Tag).

Zu 7: Minimale Anzahl von Zeiteinheiten aus Punkt 6:, die der KOMMT-Zustand aufrecht erhalten bleiben muß, um nicht als Flimmern interpretiert zu werden. Wird der Wert 0 eingetragen, wird dieser Zustand nicht überwacht.

Zu 8: Zeitbezogene Dimension zu Punkt 9: gemäß FND-Spezifikation Anhang E (12 = Sekunde, 13 = Minute, 14 = Stunde, 15 = Tag).

Zu 9: Minimale Anzahl von Zeiteinheiten aus Punkt 8:, die eine Periode (GEHT -> KOMMT -> GEHT bzw. KOMMT -> GEHT -> KOMMT) dauern muß, um nicht als Flimmern interpretiert zu werden. Wird der Wert 0 eingetragen, wird dieser Zustand nicht überwacht.

Zu 10: Maximale Anzahl in Folge erkannter Flimmerer, bevor der Status "Geber gestört" intern gesetzt wird. Für 0 wird keine Überwachung auf Flimmern durchgeführt.

- Zu 11: Für den Wert 0 werden weiterer Meldungen für diesen Datenpunkt weder übertragen, noch zwischengespeichert, falls ein Flimmern erkannt wurde. Für Werte > 0 werden weitere Meldungen übertragen, bzw. zwischengespeichert entsprechend dem Flag für das Meldungsverhalten. Die Kodierung der Werte 2 - 255, oder der einzelnen Bits 1 - 7 bleibt für spätere Verwendung reserviert, dementsprechend wird jetzt der Wert mit 1 vorbesetzt.
- Zu 12 ff: DP_0, DP_1 und DP_2 haben denselben Aufbau wie er für den jeweiligen Datenpunkt-Typ in der FND-Spezifikation Version 1.0 definiert ist, d.h. "y" ist abhängig vom Datenpunkt-Typ mit

$$y = \text{len}(\text{DP}_0[\text{dp_type}]) + \text{len}(\text{DP}_1[\text{dp_type}]) + \text{len}(\text{DP}_2[\text{dp_type}]) + 35.$$

4.2.4 Datenpunkt-Parameter für Meß-, Stell- und Zählpunkte

Für Meßpunkte führt der GA-Knoten eine Vorverarbeitung der zyklisch zu erfassenden Meldungen durch. Diese Vorverarbeitung muß von der LZH für jeden Datenpunkt über Parameter durch einen Parametriersatz gesteuert werden. Abhängig vom Datenpunkt-Typ werden unterschiedliche Informationen zur Vorverarbeitung benötigt, d.h. der Parametriersatz ist für jeden Datenpunkt-Typ spezifisch aufgebaut.

Mittels der **Zykluszeit** und der **Hysterese** kann ein Datenpunkt so parametrierbar werden, daß der Istwert erst dann gespeichert wird, wenn mindestens eine definierbare Zeit seit der letzten Speicherung vergangen ist und/oder eine relevante Änderung stattgefunden hat. Dabei ist die Hysterese als absoluter und relativer Wert definierbar.

Die Angabe eines **festen Datums/Uhrzeit** stellt sicher, daß in regelmäßigen Abständen die Meßwerte gespeichert werden, unabhängig davon, ob die Bedingung entsprechend der Zykluszeit und Hysterese erfüllt ist. Zykluszeit und Hysterese beziehen sich dann auf diesen neuen Zeitpunkt und Wert.

Soll von dem GA-Knoten für einen Datenpunkt eine **Vorverarbeitung** (Min-, Max- oder Mittelwert-Bildung) über die Zykluszeit durchgeführt werden, so kann die Vorverarbeitung und das **Abfrageintervall** angegeben werden (z.B. Viertelstundenwert).

Es gibt Geber, die Werte nicht in der gewünschten physikalischen Einheit liefern, sodaß vor einer Weiterverarbeitung eine **Skalierung** vorgenommen werden muß (z.B. Umrechnung von Impulsen in [m3] oder [kWh], Linearisierung PT100, ...). Zu diesem Zweck lassen sich die Koeffizienten eines Polynoms zweiten Grades (a0, a1 und a2) angeben.

Abhängig vom Datenpunkt-Typ sind verschiedene **überwachungen** erforderlich (z.B. Warnungs- und Alarm-Grenzen bei Meß- und Zählwerten, Modifikations-Kontrolle, ...). Die zugehörigen Grenzwerte werden aus den typspezifischen DP_0, DP_1 und DP_2 entsprechend der FND-Spezifikation Version 1.0 übernommen.

4.2.5 FND-Konventionen für den Parametriersatz von Meß-, Stell- und Zählpunkten

Generell werden für die Parametrierung dieselben Datenpunkt-Adressen verwendet, wie für das Blockformat ("**\$DP-TYP_n-oooo**" mit "n" = dp_type (Datenpunkt-Typ) und "oooo" Objektkennung), d.h. die Unterscheidung erfolgt allein durch Verwendung der **Kartei-Informationsebene 1**.

Der DP_2 des Karteipunktes für den Parametriersatz der Meß-, Stell- und

Zählpunkte hat den folgenden Aufbau:

Tabellenindex: Aktivierungszeit (IEEE-754 mit Dimension 160) wird explizit angegeben (mode_i.itm=1). Ist der Wert mit < 0 vorbesetzt, so erfolgt die Aktivierung sofort.

Zeilenindex: Reserviert für zukünftige Erweiterungen. Momentan stets 1.0 explizit angegeben (mode_j.itm=1).

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = y

<u>Sp.</u>	<u>Byte</u>	<u>Bezeichnung</u>	<u>Format</u>
1:	1...16	Datenpunkt-Adresse	16 Bytes
2:	17	Flag für Meldungsverhalten	1 Byte
3:	18	Flag für Protokollierung	1 Byte
4:	19...22	Zykluszeit	IEEE-754 mit Dimension 13
5:	23...26	Hysterese	IEEE-754
6:	27	Sync.-Code	1 Byte
7:	28...31	Sync.-Zeit	IEEE-754 mit Dimension aus 6:
8:	32	Vorverarbeitung	1 Byte
9:	33...36	Abfrageintervall	IEEE-754 mit Dimension 13
10:	37...48	Skalierungskoeff.	3 x IEEE-754 (a0, a1 und a2)
11:	49...	DP_0[dp_type]	gemäß FND-Spez. Version 1.0
12:	...	DP_1[dp_type]	gemäß FND-Spez. Version 1.0
13:	...y	DP_2[dp_type]	gemäß FND-Spez. Version 1.0

Zu 1: Datenpunkt-Adresse

Zu 2: Verhalten des GA-Knoten, sobald für den Datenpunkt eine Meldung erzeugt wird.

Zu 3: Protokollierungsverhalten des GA-Knoten, sobald für den Datenpunkt eine Meldung erzeugt bzw. einer der Datenpunkt-Parameter modifiziert wird.

Zu 4: Zulässig sind nur Werte > 0. Ungültige Werte, d.h. < 0 werden so interpretiert, daß die Zykluszeit nicht berücksichtigt wird. Es erfolgt eine Speicherung nur abhängig von der angegebenen Hysterese bzw. der Synchronisationszeit.

Zu 5: Für dimensionsbehaftete Datenpunkte (Meß-, Stell- und Zählpunkte) definieren Werte > 0 die notwendige absolute Änderung bezogen auf den letzten gespeicherten Wert, bevor eine erneute Speicherung erfolgen darf. Für Werte < 0 gilt entsprechendes als relative Angabe (in % des letzten gespeicherten Wertes). Für einen Wert = 0 erfolgt keine Berücksichtigung der Hysterese.

Zu 6: Dimensions-Code für die Synchronisationszeit unter 7:.

Zu 7: Zeitpunkt mit Dimension aus 6:, an dem generell eine Speicherung vorgenommen werden soll, unabhängig davon, ob die Bedingung entsprechend der Zykluszeit und/oder der Hysterese erfüllt ist. Für Werte < 0 wird keine Synchronisation durchgeführt.

Zu 8: Die Art der Vorverarbeitung wird als Zahl wie folgt kodiert:

- 0: KEINE
- 1: MINIMUM-Bildung
- 2: MAXIMUM-Bildung
- 3: MITTELWERT-Bildung

Zu 9: Zeitabstand zwischen den Istwertabfragen an die DDC-Insel in Minuten bezogen auf die letzte Istwertabfrage.

Zu 10: Die identische Skalierung entspricht dem Koeffizienten-Tripel (0.0, 1.0, 0.0). Wird (0.0, 0.0, 0.0) angegeben, so wird die Skalierung ignoriert, d.h. der vom Geber gelesene Wert wird identisch weiterverarbeitet.

Zu 11 ff: DP_0, DP_1 und DP_2 haben denselben Aufbau wie er für den jeweiligen Datenpunkt-Typ in der FND-Spezifikation Version 1.0

definiert ist, d.h "y" ist abhängig vom Datenpunkt-Typ mit
 $y = \text{len}(\text{DP}_0[\text{dp_type}]) + \text{len}(\text{DP}_1[\text{dp_type}]) +$
 $\text{len}(\text{DP}_2[\text{dp_type}]) + 48.$

4.3 Verbindungskontrolle

Die Kosten für die Übertragung der Daten setzen sich aus einem Anteil für die Verbindungszeit und/oder die übertragende Datenmenge zusammen. Die Datenmenge läßt sich durch die oben beschriebene Parametrierung für die Zwischenspeicherung der Meldungen von zyklischen Datenpunkten sinnvoll minimieren (Zykluszeit, Hysterese, ...). Für die vielen kleinen Objekten ist eine **Festverbindung nicht erforderlich**. Es reicht aus, wenn die zwischengespeicherten Daten in längeren Abständen als ein Block übertragen werden.

Ausgenommen sind hiervon Meldungen von Datenpunkten, für die ein entsprechendes Meldungsverhalten definiert ist, z.B. Gefahrmeldungen, evtl. Störmeldungen oder Alarmer. Diese Meldungen müssen sofort an die LZH bzw. an einen BWD weitergeleitet werden.

4.3.1 Aktivierungszeit des Verbindungs-Profiles

Wenn ein Datenpunkt parametrierung wird, kann der Zeitpunkt (Datum/Zeit) festgelegt werden, ab dem der GA-Knoten diesen Parametersatz aktivieren soll. Mit Aktivieren des Parametersatzes gilt dann das darin angegebene Verbindungs-Profil (Vgl. Pkt. 4.2.1).

4.3.2 Verbindungs-Profil

Der von dem GA-Knoten vorzunehmende Verbindungsaufbau zur Übertragung der zwischengespeicherten Meldungen wird parametrierung, d.h. es wird festgelegt:

- Die **Zykluszeit** zwischen 2 Übertragungen.
- Die max. Puffergröße der **zwischenzuspeichernden Datenmenge**.
- 2 Zeitbereiche
- Größerer **Zeitintervalle**, wie Wochen- oder Monatszyklen, können durch die geeignete Wahl der Zeiteinheit gemäß der FND-Ergänzungen angegeben werden (z.B. Dimension 164, 165).
- Um die Daten zu einem bestimmten Zeitpunkt auf der LZH verfügbar zu haben (z.B. Monatskontrolle), wird ein **Synchronisationszeitpunkt** definiert. Zum Synchronisationszeitpunkt werden die Daten dann unabhängig der definierten Zykluszeit und/oder Puffergrenze übertragen.
- Ob eine Meldung für einen Datenpunkt zwischengespeichert oder sofort an die LZH, oder an einen vorgegebenen BWD übertragen werden soll.
- Es können max. 8 **Verbindungs-Adressen** im Verbindungs-Profil eingetragen werden.
- über Schalterpunkte kann durch die LZH oder durch ein Zeit- und/oder Ereignisschaltprogramm festgesetzt werden, welche Verbindungs-Adresse gewählt werden soll.
Die Datenpunktadresse des Schalterpunktes lautet "%VB-KONTR_x-oooo":
 - x=L für LZH
 - x=G für BWD-GEFAHR
 - x=E für BWD-EINBRUCH
 - oooo für Objektkennung.
- Jeder **Verbindungs-Adresse** wird eine **Verbindungs-Art** und zur Übertragung an einen BWD auch ein eigenes **Verbindungs-Protokoll** zugeordnet.

Hinweis: Die im Verbindungs-Profil angegebenen Verbindungs-Adresse werden

innerhalb des FND-Protokolls nicht syntaktisch oder semantisch ausgewertet, sondern nur als eine Zeichenfolge übertragen. Die korrekte Verwendung der verfügbaren Verbindungs-Adressen muß innerhalb der Software des GA-Knoten erfolgen.

Kommt eine Verbindung nicht zustande, so wird nach einer definierbaren Wartezeit der Versuch wiederholt.

Die **Verbindungs-Art** (Vermittlungsschicht gemäß OSI-Schicht 3) wird als natürliche Zahl wie folgt kodiert:

- 0: stehende Verbindung, z.B. verlegtes Kabel, Standleitung; keine Adresse notwendig. Das Verbindungsprotokoll wird direkt verwendet.
- 1: stehende Verbindung, X.25; Adresse ist X.25-Adresse bzw. Port-Nummer oder PVC-Nummer.
- 2: stehende Verbindung, X.25; Adresse ist Dateiname für Verbindungsparameter.
- 3-9: reserviert
- 10: wählbare Modem-Verbindung, Hayes-AT-Satz; Adresse ist Rufnummer. Das Verbindungsprotokoll wird direkt verwendet.
- 11: wählbare Modem-Verbindung, Hayes-AT-Satz; Adresse ist Dateiname für Verbindungsparameter. Das Verbindungsprotokoll wird direkt verwendet.
- 12-19: reserviert
- 20: wählbare ISDN-Verbindung, CAPI-Schnittstelle; Adresse ist Rufnummer. Das Verbindungsprotokoll wird direkt verwendet.
- 21: wählbare ISDN-Verbindung, X.25; Adresse ist nur X.25-Adresse, d.h. die Rufnummer ist fest zugeordnet.
- 22: wählbare ISDN-Verbindung, X.25; Adresse ist Rufnummer und X.25-Adresse.
- 23: wählbare ISDN-Verbindung, X.25; Adresse ist Rufnummer und Dateiname für X.25-Verbindungs-Parameter.
- 24-29: reserviert
- 31-33: Analog zu 21 ff. mit Short-Hold-Modus, d.h. die wählbare ISDN-Verbindung zwischen GA-Knoten und X.25-Knoten-Rechner wird nur bei Bedarf aufgebaut, die X.25-Verbindung bleibt jedoch bestehen.
- 34-39: reserviert
- 40: UDP/IP
- 41: reserviert
- 42: reserviert

Das **Verbindungs-Protokoll** wird als natürliche Zahl wie folgt kodiert:

- 1: FND gemäß FND-Spezifikation
- 2: FND gemäß FND-Spezifikation mit FND-Erweiterung
- 10: TSS 13a nach VDS 2305
- 11: VDS 2465
- 110: TSS 13a verpackt in einen FND-Transferpunkt
- 111: VDS 2465 verpackt in einen FND-Transferpunkt

4.3.3 FND-Konventionen für das Verbindungs-Profil

Als Datenpunkt-Adresse für das Verbindungs-Profil wird die **Datenpunkt-Adresse "\$VB-PROFIL-oooo"** verwendet ("oooo" entspricht der Objektkennung, wie sie auch innerhalb der LZH verwendet wird), die mit keiner der anderen Datenpunkt-Adressen einen Konflikt bildet. Durch Verwendung des Karteipunktes (Datenpunkt-Typs "7") unter `dp_type` im Header.2 und der **Kartei-Informationsebene 0** werden die Verbindungsparameter im DP_2 übertragen. Da die Länge für ein Verbindungs-Profil die maximal zulässige Länge für einen "string" im Karteipunkt überschreitet, muß das Verbindungs-Profil in mehreren Teilen

übertragen werden. Zur Unterscheidung wird der Zeilenindex inkrementiert.

Der DP_2 des Karteipunktes für das Verbindungs-Profil hat den folgenden Aufbau:

- Tabellenindex: Aktivierungszeit (IEEE-754 mit Dimension 160) wird explizit angegeben (mode_i.itm=1). Wird hier ein Wert < 0 eingetragen, so erfolgt die Aktivierung sofort.
- Zeilenindex: Die Länge des gesamten Datensatzes übersteigt die maximal zulässige Anzahl von 255 Zeichen. Die Übertragung erfolgt also in Teilen mit inkrementiertem Zeilenindex beginnend bei 1.0; explizit angegeben (mode_j.itm=1).
- Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = Länge des zu übertragenden Teiles

Sp. Byte	Bezeichnung	Format
<u>Teil 1: Übertragung mit Zeilenindex = 1.0 und strng len = 219</u>		
1: 1	Dimension für Zykluszeit	1 Byte
2: 2...5	Zykluszeit	IEEE-754 Dimension aus 1:
3: 6...9	Puffergrenze	IEEE-754
4: 10	Bereich_1-Code	1 Byte
5: 11..14	Bereich_1-Von_Zeit	IEEE-754 mit Dimension zu 3
6: 15..18	Bereich_1-Bis_Zeit	IEEE-754 mit Dimension zu 3
7: 19	Bereich_2-Code	1 Byte
8: 20..23	Bereich_2-Von_Zeit	IEEE-754 mit Dimension zu 6
9: 24..27	Bereich_2-Bis_Zeit	IEEE-754 mit Dimension zu 6
10: 28	Sync.-Code	1 Byte
11: 29..32	Sync.-Zeit	IEEE-754 mit Dimension zu 9
12: 33..36	Wartezeit	IEEE-754 mit Dimension 13
13: 37	Verbindungs-Art zu 14:	1 Byte
14: 38...97	LZ-Verbindungs-Adresse 1	60 Byte
15: 98	Verbindungs-Art zu 16:	1 Byte
16: 99...158	LZ-Verbindungs-Adresse 2	60 Byte
17: 159	Verbindungs-Art zu 18:	1 Byte
18: 160..219	LZ-Verbindungs-Adresse 3	60 Byte
<u>Teil 2: Übertragung mit Zeilenindex = 2.0 und strng len = 244</u>		
19: 220	Verbindungs-Art zu 20:	1 Byte
20: 221..280	LZ-Verbindungs-Adresse 4	60 Byte
21: 281	Verbindungs-Art zu 22:	1 Byte
22: 282..341	LZ-Verbindungs-Adresse 5	60 Byte
23: 342	Verbindungs-Art zu 24:	1 Byte
24: 343..402	LZ-Verbindungs-Adresse 6	60 Byte
25: 403	Verbindungs-Art zu 26:	1 Byte
26: 404..463	LZ-Verbindungs-Adresse 7	60 Byte
<u>Teil 3: Übertragung mit Zeilenindex = 3.0 und strng len = 247</u>		
27: 464	Verbindungs-Art zu 28:	1 Byte
28: 465..424	LZ-Verbindungs-Adresse 8	60 Byte
29: 425	Verbindungs-Art zu 31:	1 Byte
30: 426	Verbindungs-Protokoll zu 31:	1 Byte
31: 427..486	GEFAHR-Verbindungs-Adresse 1	60 Byte
32: 487	Verbindungs-Art zu 34:	1 Byte
33: 488	Verbindungs-Protokoll zu 34:	1 Byte
34: 489..548	GEFAHR-Verbindungs-Adresse 2	60 Byte
35: 549	Verbindungs-Art zu 37:	1 Byte
36: 550	Verbindungs-Protokoll zu 37:	1 Byte
37: 551..610	GEFAHR-Verbindungs-Adresse 3	60 Byte
<u>Teil 4: Übertragung mit Zeilenindex = 4.0 und strng len = 248</u>		
38: 611	Verbindungs-Art zu 40:	1 Byte
39: 612	Verbindungs-Protokoll zu 40:	1 Byte
40: 613..672	GEFAHR-Verbindungs-Adresse 4	60 Byte

41: 673	Verbindungs-Art zu 43:	1 Byte
42: 674	Verbindungs-Protokoll zu 43:	1 Byte
43: 675..734	GEFAHR-Verbindungs-Adresse 5	60 Byte
44: 735	Verbindungs-Art zu 46:	1 Byte
45: 736	Verbindungs-Protokoll zu 46:	1 Byte
46: 737..796	GEFAHR-Verbindungs-Adresse 6	60 Byte
47: 797	Verbindungs-Art zu 49:	1 Byte
48: 798	Verbindungs-Protokoll zu 49:	1 Byte
49: 799..858	GEFAHR-Verbindungs-Adresse 7	60 Byte

Teil 5: Übertragung mit Zeilenindex = 5.0 und strng len = 248

50: 859	Verbindungs-Art zu 52:	1 Byte
51: 860	Verbindungs-Protokoll zu 52:	1 Byte
52: 861..920	GEFAHR-Verbindungs-Adresse 8	60 Byte
53: 921	Verbindungs-Art zu 55:	1 Byte
54: 922	Verbindungs-Protokoll zu 55:	1 Byte
55: 923..982	EINBRUCH-Verbindungs-Adresse 1	60 Byte
56: 983	Verbindungs-Art zu 58:	1 Byte
57: 984	Verbindungs-Protokoll zu 58:	1 Byte
58: 985...1044	EINBRUCH-Verbindungs-Adresse 2	60 Byte
59: 1045	Verbindungs-Art zu 61:	1 Byte
60: 1046	Verbindungs-Protokoll zu 61:	1 Byte
61: 1047..1106	EINBRUCH-Verbindungs-Adresse 3	60 Byte

Teil 6: Übertragung mit Zeilenindex = 6.0 und strng len = 248

62: 1107	Verbindungs-Art zu 64:	1 Byte
63: 1108	Verbindungs-Protokoll zu 64:	1 Byte
64: 1109..1168	EINBRUCH-Verbindungs-Adresse 4	60 Byte
65: 1169	Verbindungs-Art zu 67:	1 Byte
66: 1170	Verbindungs-Protokoll zu 67:	1 Byte
67: 1171..1230	EINBRUCH-Verbindungs-Adresse 5	60 Byte
68: 1231	Verbindungs-Art zu 70:	1 Byte
69: 1232	Verbindungs-Protokoll zu 70:	1 Byte
70: 1233..1292	EINBRUCH-Verbindungs-Adresse 6	60 Byte
71: 1293	Verbindungs-Art zu 73:	1 Byte
72: 1294	Verbindungs-Protokoll zu 73:	1 Byte
73: 1295..1354	EINBRUCH-Verbindungs-Adresse 7	60 Byte

Teil 7: Übertragung mit Zeilenindex = 7.0 und strng len = 62

74: 1355	Verbindungs-Art zu 76:	1 Byte
75: 1356	Verbindungs-Protokoll zu 76:	1 Byte
76: 1357..1416	EINBRUCH-Verbindungs-Adresse 8	60 Byte

- Zu 1: Dimensions-Code für 2:. Durch die in den FND-Ergänzungen definierten Kodierungen 12-15 lassen sich entsprechende Übertragungszyklen (Stunden-, Tages-, Wochen, Monats- und Jahres-Zyklen) festlegen.
- Zu 2: Zulässig sind nur Werte > 0. Ungültige Werte werden so interpretiert, daß die Zykluszeit nicht berücksichtigt wird, d.h. die Daten werden nur abhängig von der angegebenen Puffergrenze bzw. der Synchronisationszeit übertragen.
- Zu 3: Maximale zulässige Speicherkapazität in MB, die für die Zwischenpufferung verwendet werden darf. Für den Werte = 0 werden die Daten sofort übertragen, für den Wert < 0 wird keine Puffergrenze berücksichtigt.
- Zu 4: Dimensions-Code für den Übertragungsbereich 1 unter 5: und 6:
- Zu 5: Anfangs-Zeitpunkt mit Dimension aus 4:, für den Übertragungsbereich 1, wenn die Bedingung entsprechend der Zykluszeit oder der Puffergrenze erfüllt ist. Für Werte < 0 wird dieser Bereich ignoriert.
- Zu 6: Ende-Zeitpunkt mit Dimension aus 4:, für den Übertragungsbereich 1. Für Werte < Anfangszeitpunkt wird ein überlauf behandelt, z.B. tägliche Übertragung (Code 167) von

- 18:00 Uhr (Anfangszeit 18.0) bis zum nächsten Morgen um 6:00 Uhr (Endezeit 6.0). Für Werte < 0 wird dieser Bereich ignoriert.
- Zu 7: analog zu 4;; jedoch für den Übertragungsbereich 2
 - Zu 8: analog zu 5;; jedoch für den Übertragungsbereich 2
 - Zu 9: analog zu 6;; jedoch für den Übertragungsbereich 2
 - Zu 10: Dimensions-Code für die Synchronisationszeit unter 11:.
 - Zu 11: Zeitpunkt mit Dimension aus 10:, an dem generell eine Übertragung vorgenommen werden soll, unabhängig davon, ob die Bedingung entsprechend der Zykluszeit und/oder der Puffergrenze erfüllt ist. Für Werte < 0 wird keine Synchronisation durchgeführt.
 - Zu 12: Zulässig sind nur Werte >0. Ungültige Werte < 0 werden so interpretiert, daß keine Wartezeit berücksichtigt wird, d.h. es erfolgt sofort ein erneuter Versuch zum Verbindungsaufbau in der definierten Reihenfolge.
 - Zu 13 bis 28: 8-mal Verbindungs-Art und -Adresse zur LZH
 - Zu 29 bis 52: 8-mal zusätzliche Verbindungs-Art, -Protokoll und -Adresse zur Weiterleitung von Gefahrmeldungen an einen BWD.
 - Zu 53 bis 76: 8-mal zusätzliche Verbindungs-Art, -Protokoll und -Adresse zur Weiterleitung von Einbruchmeldungen an einen BWD.

4.4 Schaltprogramm

Schaltprogramme sind Befehlsfolgen für die zeit- und/oder ereignisabhängige Befehlsausgabe an mehrere Datenpunkte (**Zeit- und Ereignis-Schaltprogramm**; z.B. Sollwertausgabe an Schalt- oder Stellpunkte). Diese Befehlsfolgen werden vom Anwenderprogramm des GA-Knoten hintereinander abgearbeitet.

Die Schaltprogramme werden von der LZH verwaltet und zu den GA-Knoten übertragen. Zuerst müssen die Schaltprogramm-Parameter übertragen werden. Die Referenzliste der Zeitschaltprogramme (Zeitschaltprogramm) und/oder das Ereignisschaltprogramm folgen anschließend.

Die Befehlsausgabe entspricht dabei der Änderung des Datenfeldes in den Datenpunkt-Parametern DP_#0 - DP_#2 (z.B. Sollwert, Grenzwerte, Maskierung, ...). Der GA-Knoten führt die Konvertierung der Datenpunkt-Parameter DP_#0 - DP_#2 in die entsprechenden gerätespezifischen Befehle an die DDC-Insel durch. Die Berechtigung zur Befehlsausgabe bzw. zur Ausführung des Befehls unterliegt der "Modifikationsberechtigung der Insel" (vgl. FND-Spezifikation unter 2.8), der Modifikationskontrolle durch die Applikations-Kontrolle (vgl. FND-Spezifikation unter 2.5.1.3) sowie der für Meß- und Zählpunkte ("dp_type" = 3 und 5) definierten "update_control" im DP_#2.

Zusammen mit der aktuellen Systemzeit (Datum/Zeit) und der ausgeführten Funktion können auf dem **Protokoll-Gerät** Meldungen ausgegeben werden (z.B. Starten/Stoppen eines Schaltprogrammes, Ausführen eines Zeitschaltbefehls oder Ausgabe einer im Ereignisschaltprogramm explizit angegebenen Meldung).

Ist die Befehlsfolge des Schaltprogrammes abgearbeitet oder wird das Schaltprogramm explizit gestoppt, so wird es beendet. Ein zyklisch ablaufendes Schaltprogramm wiederholt den Ablauf zu einem festgelegten Zeitpunkt, solange es nicht explizit über den Schaltprogramm-Kalender gestoppt wird. Dieser Zeitpunkt entspricht dem Anfang einer **Zeitzyklus** (z.B. Stunde, Tag, Woche oder Monat), z.B. können so Wochenprogramme erstellt werden, die jeweils am Montag um 0.00 Uhr neu starten. Ein sporadisch ablaufendes Schaltprogramm muß nach dessen Beendigung über den Schaltprogramm-Kalender explizit wieder gestartet werden.

Neben dem gegenseitigen Starten/Stoppen von Schaltprogrammen kann der Benutzer auf der LZH über einen **Schaltprogramm-Kalender** für jedes Schaltprogramm beliebig viele Termine verwalten und bearbeiten, an denen der Ablauf des Schaltprogramms gestartet oder gestoppt werden soll. In einer **Referenzliste der Zeitschaltpunkte** können, bezogen auf die Zeitbasis, Zeitpunkte angegeben werden, an denen eine Befehlsausgabe an einen Datenpunkt durchgeführt werden soll. Bei chronologischer Abarbeitung der Einträge für ein Schaltprogramm durch den GA-Knoten entspricht dieses dem **Zeitschaltprogramm**. Ist die Referenzliste "leer", so ist kein Zeitschaltprogramm definiert.

Zeitgleich zum Zeitschaltprogramm kann ein **Ereignisschaltprogramm** ablaufen. Dieses ist eine beliebige Befehlsfolge in einer erweiterten Postfix-Notation (Vgl. Pkt 4.4.6). Diese Befehlsfolge muß abhängig von datenpunkt- und systemspezifischen Ereignissen abgearbeitet werden.

Hierfür wird eine für die LZH einheitliche Syntax definiert. Diese enthält Konstanten, Variablen, Operatoren, Funktionen, Vergleichoperatoren,

4.4.1 Aktivierungszeit des Schaltprogrammes

Durch Angabe eines **Zeitpunktes (Datum/Zeit)**, an dem ein **Schaltprogramm aktiv werden soll**, können neue Schaltprogramm-Versionen im voraus auf dem GA-Knoten hinterlegt werden. Die Aktivierung bedeutet nicht gleichzeitig das Starten des Schaltprogrammes, sondern nur, daß ab dem Aktivierungszeitpunkt diese Version unter der zugehörigen Schaltprogramm-Kennung zu verwenden ist. Auf diese Weise kann eine neue Version eines Schaltprogrammes im voraus hinterlegt werden. Das Schaltprogramm wird gestartet und gestoppt abhängig vom Schaltprogramm-Kalender oder durch ein anderes Ereignis-Schaltprogramm. Der GA-Knoten speichert alle hinterlegten Schaltprogramme. Automatisch gelöscht wird ein Schaltprogramm dann, wenn es durch eine neue Version ersetzt wurde. Ist dieses neue Schaltprogramm "leer", d.h. es wurde weder ein zugehöriges Zeit- noch ein Ereignisschaltprogramm zu den Schaltprogramm-Parametern übertragen, so werden auch die Schaltprogramm-Parameter nach der Aktivierung gelöscht, d.h. es kann auch im voraus das Löschen eines Schaltprogrammes hinterlegt werden.

4.4.2 Parametrierung des Schaltprogrammes (Schaltprogramm-Parameter)

Im **Schaltprogramm-Kalender** werden die Zeitpunkte für das Starten und Stoppen aller Schaltprogramme gemeinsam verwaltet.

Die Schaltprogramme bestehen aus Schaltprogramm-Parametern und deren zeit- und/oder ereignisabhängige Befehlsfolge. Sie werden zusammen über eine für den GA-Knoten eindeutige **Schaltprogramm-Kennung** verwaltet. Der Klartext zur **Beschreibung** wird für die Protokoll-Meldungen verwendet, wenn das Schaltprogramm gestartet oder gestoppt wird. über eine Zeitbasis wird angegeben, ob und in welchen Abständen ein Schaltprogramm zyklisch wiederholt werden soll.

4.4.3 FND-Konvention für die Schaltprogramm-Parameter

Als Datenpunkt-Adresse für die Übertragung der Schaltprogramm-Parameter wird die **Datenpunkt-Adresse "\$SCHALTPROG-oooo"** verwendet ("oooo" entspricht der Objektkennung, wie sie auch innerhalb der LZH verwendet wird), die mit keiner der anderen Datenpunkt-Adressen einen Konflikt bildet. Durch Verwendung des Karteipunktes (Datenpunkt-Typs "7") unter

dp_type im Header.2 und der **Kartei-Informationsebene 0** werden die Schaltprogramm-Parameter im DP_2 übertragen.

Die Übertragung der Referenzliste der Zeitschaltpunkte (Zeitschaltprogramm) und des Ereignisschaltprogrammes erfolgen unter Verwendung derselben Datenpunkt-Adresse, jedoch werden die Kartei-Informationsebenen 1 und 2 mit einem eigenen Format im DP_2 verwendet.

Der GA-Knoten ordnet das von der LZH übertragene Zeit- und Ereignis-Schaltprogramm immer der Schaltprogramm-Kennung der zuletzt übertragenen Schaltprogramm-Parameter zu.

Der DP_2 des Karteipunktes für die Schaltprogramm-Parameter hat den folgenden Aufbau:

Tabellenindex: Aktivierungszeit (IEEE-754 mit Dimension 160) wird explizit angegeben (mode_i.itm=1). Ist hier der Wert < 0 vorbesetzt, so wird sie sofort aktiviert.

Zeilenindex: Reserviert für zukünftige Erweiterungen. Momentan stets 1.0 explizit angegeben (mode_j.itm=1).

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = 72.

<u>Sp.</u>	<u>Byte</u>	<u>Bezeichnung</u>	<u>Format</u>
1:	1...10	Schaltprogramm-Kennung	10 Byte
2:	11...70	Beschreibung	60 Byte
3:	71	Zeitbasis	1 Byte
4:	72	Flag für Protokollierung	1 Byte

Zu 1: Kennung des Schaltprogrammes. Diese ist für das Objekt, dem der GA-Knoten zugeordnet ist, eindeutig.

Zu 2: Klartext zur Beschreibung des Schaltprogrammes, ... Dieser Text wird für eine Protokoll-Meldung verwendet, wenn das Flag unter 4: gesetzt ist.

Zu 3: Zeitbasis des Schaltprogrammes. Folgende Kodierungen sind möglich:

- 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)

Zu 4: Ist das Flag gesetzt: Es erfolgt eine Protokoll-Meldung, wenn das Schaltprogramm gestartet oder gestoppt wurde. SONST: Keine Protokoll-Meldung.

4.4.4 Parametrierung des Zeit-Schaltprogrammes

In dieser Referenzliste werden alle **Zeitpunkte** (Tag/Zeit), der **Datenpunkt** und der zugehörige **auszugebende Befehl** (Zeitschaltprogramm) vermerkt. Jeder Datenpunkt besteht aus mehreren Datenfeldern in den DP_0 bis DP_2, an die spezifische Befehle ausgegeben werden sollen. Abhängig von dem Datentyp dieses Datenfeldes wird ein binärer Wert (B*1; 1 Byte zzgl. 1 Byte Filtermaske) oder eine reelle Zahl (R*4; 4 Byte) als auszugebender Befehl verwendet. Abhängig von der für das Schaltprogramm definierten Zeitbasis werden die Zeitpunkte für die Befehlsausgabe durch den **Tag**,

bezogen auf den Anfang des Zeitzyklus, und die **Zeit** definiert. Wird für die Zeitbasen "Woche" und "Monat" kein Tag angegeben (Tabellenindex < 0; vgl. unten), so wird dieser Befehl an jedem Tag ausgegeben.

Mit dem Datentyp des Datenfeldes, an das die Befehlsausgabe erfolgen soll, wird auch der Datentyp des auszugebenden Befehls festgelegt. Die Befehlsausgabe ist beschränkt auf die Datenfelder, die gemäß FND-Spezifikation verändert werden dürfen. Es sind jedoch alle Datenfelder hier aufgeführt, da Ereignisschaltprogramme auch lesende Zugriffe durchführen. Es werden zur Definition der Datenfelder im DP_0 bis DP_2 die folgenden Kodierungen verwendet:

Datenfelder aus DP_#0:

<u>Kode</u>	<u>Datenfeld</u>	<u>Format und Filtermaske =0xff</u>
1	mask	(B*1 für "dp_type" = 1 bis 5)
2	info_event	(B*1 für "dp_type" = 1 bis 5)

Datenfelder aus DP_#1:

<u>Kode</u>	<u>Datenfeld</u>	<u>Format und Filtermaske =0xff</u>
11	actual	(B*1 für "dp_type" = 1 und 2; R*4 für "dp_type" 3 bis 5)
12	atb_id	(B*1 für "dp_type" = 1 und 2)
13	nominal	(B*1 für "dp_type" = 2; R*4 für "dp_type" 4)
14	dimension	(B*1 für "dp_type" = 3 bis 5)
15	last_save	(B*1 für "dp_type" = 5)

Datenfelder aus DP_#2:

<u>Kode</u>	<u>Datenfeld</u>	<u>Format und Filtermaske =0xff</u>
21	update_control	(B*1 für "dp_type" = 3 und 5)
22	fixed	(B*1 für "dp_type" = 3)
23	alarm_low	(R*4 für "dp_type" = 3)
24	warning_low	(R*4 für "dp_type" = 3)
25	warning_high	(R*4 für "dp_type" = 3 und 5)
26	alarm_high	(R*4 für "dp_type" = 3 und 5)

Für die Datenfelder "mask", "info_event" und "update_control" müssen auch interne Bitkombinationen als Datenfeld verfügbar sein (vgl. FND-Spezifikation unter 2.3 ff., 2.7 und 2.8). Hierzu muß eine Bitmaskierung (**Filtermaske**) vorgenommen werden. Diese Datenfelder aus DP_#0 und DP_#2 gegen festen Vergleichswert:

<u>Kode</u>	<u>Datenfeld</u>	<u>Format und Filtermaske</u>
41	mask_msk	(B*1 mit Filtermaske 0x03)
42	mask_mis	(B*1 mit Filtermaske 0x0c)
43	mask_apr	(B*1 mit Filtermaske 0x30)
44	mask_val	(B*1 mit Filtermaske 0xc0)
51	ie_f1	(B*1 mit Filtermaske 0x01)
52	ie_te	(B*1 mit Filtermaske 0x08)
53	ie_rl	(B*1 mit Filtermaske 0x80)
54	ie_f2	(B*1 mit Filtermaske 0x02)
55	ie_ch	(B*1 mit Filtermaske 0x04)
56	ie_tl	(B*1 mit Filtermaske 0x40)
57	ie_wg	(B*1 mit Filtermaske 0x02)
58	ie_al	(B*1 mit Filtermaske 0x04)
59	ie_lv	(B*1 mit Filtermaske 0x40)
60	ie_or	(B*1 mit Filtermaske 0x02)
61	ie_cc	(B*1 mit Filtermaske 0x04)
62	ie_sv	(B*1 mit Filtermaske 0x40)

Für "ie_sv" zusammengefaßt mit "iek_or" gilt:

67	ie_so	(B*1 mit Filtermaske 0x42)
----	-------	----------------------------

Kode	Datenfeld	Format und Filtermaske
71	uc_r1	(B*1 mit Filtermaske 0x01)
72	uc_r2	(B*1 mit Filtermaske 0x02)
73	uc_r3	(B*1 mit Filtermaske 0x04)
74	uc_r4	(B*1 mit Filtermaske 0x08)

Für die Ausgabe eines binären Befehls ist es notwendig, auch einzelne Bits in dem Datenfeld zu modifizieren. Dazu wird im auszugebenden Befehl ein Byte als **Filtermaske** und ein zweites Byte als Befehl verwendet. In der Filtermaske sind alle die Bits gesetzt, die zu modifizieren sind. Durch die bitweise Verknüpfung der invertierten Filtermaske mit dem binären Wert aus dem Datenfeld werden alle zu modifizierenden Bits auf 0 gesetzt. Durch die bitweise Verknüpfung der Filtermaske mit dem auszugebenden Befehl wird sichergestellt, daß alle Bits im auszugebenden Befehl auf 0 gesetzt sind, die nicht für die Modifikation verwendet werden. Anschließend wird durch eine Verknüpfung beider Werte der neue binäre Wert für das Datenfeld gebildet.

Für den Datentyp R*4 wird der auszugebende reelle Wert direkt als Befehl, d.h. neuer Wert, für das Datenfeld verwendet.

Nachfolgender Datenfelder wurden ergänzt und steuern das Melde- und Protokollierungsverhalten des Datenpunktes (vgl. unter Abschnitt 4.2):

Kode	Datenfeld	Format und Filtermaske
120	flag_meld	(B*1 für "dp_type" = 1 bis 5)
121	flag_prot	(B*1 für "dp_type" = 1 bis 5)

Nachfolgender Datenfelder wurden ergänzt und steuern die ergänzenden Parameter von Melde- und Schaltpunkten (vgl. unter Abschnitt 4.2.2):

Kode	Datenfeld	Format und Filtermaske
130	geht_dim	(B*1 für "dp_type" = 1 und 2)
131	geht_dauer	(R*4 für "dp_type" = 1 und 2)
132	kommt_dim	(B*1 für "dp_type" = 1 und 2)
133	kommt_dauer	(R*4 für "dp_type" = 1 und 2)
134	periode_dim	(B*1 für "dp_type" = 1 und 2)
135	periode_dauer	(R*4 für "dp_type" = 1 und 2)
136	max_flimm	(B*1 für "dp_type" = 1 und 2)
137	flag_bearb	(B*1 für "dp_type" = 1 und 2)

Nachfolgender Datenfelder wurden ergänzt und steuern die ergänzenden Parameter von Mess-, Stell- und Zählpunkten (vgl. unter Abschnitt 4.2.4):

Kode	Datenfeld	Format und Filtermaske
140	zykluszeit	(R*4 für "dp_type" = 3 bis 5)
141	hysterese	(R*4 für "dp_type" = 3 bis 5)
142	sync_code	(B*1 für "dp_type" = 3 bis 5)
143	sync_zeit	(R*4 für "dp_type" = 3 bis 5)
144	vorverarb	(B*1 für "dp_type" = 3 bis 5)
145	abfrage	(R*4 für "dp_type" = 3 bis 5)
146	a0	(R*4 für "dp_type" = 3 bis 5)
147	a1	(R*4 für "dp_type" = 3 bis 5)
148	a2	(R*4 für "dp_type" = 3 bis 5)

4.4.5 FND-Konvention für das Zeit-Schaltprogramm

Als Datenpunkt-Adresse für die Übertragung des Zeitschaltprogramms wird die Datenpunkt-Adresse "\$SCHALTPROG-oooo" verwendet ("oooo" entspricht der Objektkennung, wie sie auch innerhalb der LZH verwendet wird), die mit keiner der anderen Datenpunkt-Adressen einen Konflikt bildet. Durch

Verwendung des Karteipunktes (Datenpunkt-Typs "7") unter dp_type im Header.2 und der **Kartei-Informationsebene 1** werden die einzelnen Zeitschaltpunkte im DP_2 chronologisch sortiert (Vgl. Tabellenindex) übertragen.

Der GA-Knoten ordnet das übertragene Zeitschaltprogramm immer der Schaltprogramm-Kennung der zuletzt übertragenen Schaltprogramm-Parameter zu.

Der DP_2 des Karteipunktes für das Zeitschaltprogramm hat den folgenden Aufbau:

Tabellenindex: Zeitpunkt der Befehlsausgabe (IEEE-754 mit Dimension entsprechend der im Schaltprogramm-Parameter angegebenen Zeitbasis; vgl. FND-Dimensionen 161 - 169 gemäß FND-Ergänzungen wird explizit angegeben (mode_i.itm=1).
 - Zeitbasis 1 und 11: Dimension 169, Angaben in Minuten
 - Zeitbasis 2 und 12: Dimension 167, Angaben in Stunden
 - Zeitbasis 3 und 13: Dimension 165, Angaben in Tagen
 - Zeitbasis 4 und 14: Dimension 164, Angaben in Tagen
 - Zeitbasis 5 und 15: Dimension 161, Angaben in Tagen
 Wird für die Zeitbasen "Woche" und "Monat" ein Wert < 0 angegeben, so wird dieser Befehl an jedem Tag ausgegeben, die Zeit entspricht dem Absolut-Betrag des angegebenen Wertes.
 Sind für die Zeitbasen "Jahr", "Tag" und "Stunde" Werte < 0 angegeben, so werden diese analog zu "Woche" und "Monat" behandelt, wobei der Befehl für "Tag" jede Stunde und für "Stunde" jede Minute ausgegeben wird.

Zeilenindex: Reserviert für zukünftige Erweiterungen. Momentan stets 1.0 explizit angegeben (mode_j.itm=1).

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = 82.

Die einzelnen Datensätze für das Zeitschaltprogramm werden von der LZH chronologisch sortiert an den GA-Knoten übertragen.

<u>Sp.</u>	<u>Byte</u>	<u>Bezeichnung</u>	<u>Format</u>
1:	1 ... 16	Datenpunkt-Adresse	16 Byte
2:	17	Datenfeld für Befehlsausgabe	1 Byte
3:	18 ... 21	auszugebender Befehl	4 Byte
4:	22 ... 81	Beschreibung	60 Byte
5:	82	Flag für Protokollierung	1 Byte

- Zu 1: Identifikator des Datenpunktes, an den der Befehl ausgegeben werden soll.
- Zu 2: Kennung für das Datenfeld im DP_#0 - DP_#2 in das der Befehl ausgegeben werden soll (Kodierung; vgl. oben).
- Zu 3: Filtermaske und binärer Befehl bzw. reeller Wert als auszugebender Befehl.
- Zu 4: Freier Text zur Beschreibung der Funktion wird für eine Protokoll-Meldung verwendet, wenn das Flag unter 5: gesetzt ist.
- Zu 5: Ist das Flag gesetzt wird die Protokoll-Meldung ausgegeben.

4.4.6 Parametrierung, bzw. Syntax des Ereignis-Schaltprogrammes

Während Zeitschaltprogramme nur eine chronologische Befehlsfolge von

Schaltbefehlen sind, können mit Ereignisschaltprogramme Befehlsfolgen komplexere Abhängigkeiten (Zeit, Wartezeit, Zustand bzw. Wert eines Datenpunktes, ...) realisiert werden. Ein Ereignisschaltprogramm ist ein auf der LZH editierter Text, der in einer formalen Sprache in Postfix-Notation (Umgekehrte polnische Notation) abgefaßt ist. Diese formale Sprache enthält die folgenden Grundelemente, die sich in ihrer Syntax an die Programmiersprache C (ANSI) anlehnen, wobei die Klammern "(" und ")" ausschließlich als syntaktische Trennzeichen verwendet werden:

- alphanumerische, numerische und binäre Variablen
- alphanumerische, numerische und binäre Konstanten
- alphanumerische, numerische und binäre Operatoren, Zuweisung und Funktionen
- alphanumerische, numerische und binäre Vergleiche
- Fallunterscheidungen (IF/THEN/ELSE)
- Wiederholungen (WHILE-Schleife)
- Kommentare

Die Verwendung der Postfix-Notation erlaubt eine klammerfreie Schreibweise, die bereits der Abarbeitungsreihenfolge beim Berechnen entspricht und somit direkt bei der Berechnung interpretiert werden kann. Ein Umwandeln des editierten Textes in eine maschinenlesbare Form ist nicht notwendig.

4.4.6.1 Variable

Als **Variablen** sind die Inhalte aus allen Datenfeldern verfügbar. Für den Zugriff auf die Datenpunkt-Parameter DP_#0 bis DP_#2 im Parametersatz des Datenpunktes wird eine eigene Syntax (**Variablen-Identifikator**) wie folgt definiert (|, [und] sind nur syntaktische Zeichen, d.h. nicht Bestandteil des Variablen-Identifikators !):

A(DP_Adr[.DP_F]) z.B. "A(ZGAAMMMTTOOOOIR.actual)" bzw.
D(DP_Id[.DP_F]) z.B. "D(HHHH:ZGAAMMMTTOOOOIR.actual)"

A und **D** Kennzeichen für eine Datenpunkt-Variable.

(und) dienen als Trennzeichen

DP_Adr Datenpunkt-Adresse;

die Objektkennung entfällt, da sie implizit angegeben ist. Für virtuelle und durch Gruppenbildung definierte Datenpunkte wird dessen Objektkennung verwendet. In der Funktion "sp_ewarte" für Schaltprogramme ist das das Schaltprogramm einem Objekt fest zugeordnet.

DP_Id Datenpunkt-Identifikator

Aus Kompatibilitätsgründen zur LZH wird "HHHH" als Objektkennung verwendet und durch ":" von der Datenpunkt-Adresse syntaktisch getrennt.

DP_F Name des Datenfeldes aus der DP-Tabelle,

dessen Inhalt verwendet werden soll (falls nicht angegeben, so wird "actual" verwendet):

- mask ("dp_type" = 1 bis 5)
- info_event ("dp_type" = 1 bis 5)
- actual ("dp_type" = 1 bis 5)

- atb_id ("dp_type" = 1 und 2)
- nominal ("dp_type" = 2 und 4)
- dimension ("dp_type" = 3 bis 5)
- update_control ("dp_type" = 3 und 5)
- fixed ("dp_type" = 3)
- alarm_low ("dp_type" = 3)
- warning_low ("dp_type" = 3)

- warning_high ("dp_type" = 3 und 5)
- alarm_high ("dp_type" = 3 und 5)
- last_save ("dp_type" = 5)

Für die Datenfelder "mask", "info_event" und "update_control" müssen auch interne Bitkombinationen zugreifbar sein (vgl. FND-Spezifikation unter 2.3 ff., 2.7 und 2.8). Hierfür werden folgende Datenfelder festgelegt:

- mask_msk ("dp_type" = 1 bis 5)
- mask_mis ("dp_type" = 1 bis 5)
- mask_apr ("dp_type" = 1 bis 5)
- mask_val ("dp_type" = 1 bis 5)

- ie_f1 ("dp_type" = 1 bis 5)
- ie_te ("dp_type" = 1 bis 5)
- ie_rl ("dp_type" = 1 bis 5)
- ie_f2 ("dp_type" = 1 und 2)
- ie_ch ("dp_type" = 1 und 2)
- ie_tl ("dp_type" = 2 und 4)
- ie_wg ("dp_type" = 3)
- ie_al ("dp_type" = 3)
- ie_lv ("dp_type" = 3)
- ie_or ("dp_type" = 5)
- ie_cc ("dp_type" = 5)
- ie_sv ("dp_type" = 5)

Für "ie_sv" zusammengefaßt mit "iek_or" gilt:

- ie_so ("dp_type" = 5)

- uc_r1 ("dp_type" = 3 und 5)
- uc_r2 ("dp_type" = 3 und 5)
- uc_r3 ("dp_type" = 3 und 5)
- uc_r4 ("dp_type" = 3)

Nachfolgende Datenfelder wurden ergänzt und liefern allgemeine Informationen zum Datenpunkt, dem zugeordnete Objekt sowie den Attributs- bzw. Dimensions-Text:

- dp_type ("dp_type" = 1 bis 5)
- dp_subtype ("dp_type" = 1 bis 5)
- obj_kennung ("dp_type" = 1 bis 5)
- obj_text ("dp_type" = 1 bis 5)
- dp_text ("dp_type" = 1 bis 5)
- atb_text ("dp_type" = 1 und 2)
- dim_text ("dp_type" = 3 bis 5)

Nachfolgender Datenfelder wurden ergänzt und steuern das Melde- und Protokollierungsverhalten des Datenpunktes (vgl. unter Abschnitt 4.2):

- flag_meld ("dp_type" = 1 bis 5)
- flag_prot ("dp_type" = 1 bis 5)

Nachfolgender Datenfelder wurden ergänzt und steuern die ergänzenden Parameter von Melde- und Schaltpunkten (vgl. unter Abschnitt 4.2.2):

- geht_dim ("dp_type" = 1 und 2)
- geht_dauer ("dp_type" = 1 und 2)
- kommt_dim ("dp_type" = 1 und 2)
- kommt_dauer ("dp_type" = 1 und 2)
- periode_dim ("dp_type" = 1 und 2)
- periode_dauer ("dp_type" = 1 und 2)
- max_flimm ("dp_type" = 1 und 2)
- flag_bearb ("dp_type" = 1 und 2)

Nachfolgender Datenfelder wurden ergänzt und steuern die ergänzenden Parameter von Mess-, Stell- und Zählpunkten (vgl. unter Abschnitt 4.2.4):

- zykluszeit ("dp_type" = 3 bis 5)

- hysterese ("dp_type" = 3 bis 5)
- sync_code ("dp_type" = 3 bis 5)
- sync_zeit ("dp_type" = 3 bis 5)
- vorverarb ("dp_type" = 3 bis 5)
- abfrage ("dp_type" = 3 bis 5)
- a0 ("dp_type" = 3 bis 5)
- a1 ("dp_type" = 3 bis 5)
- a2 ("dp_type" = 3 bis 5)

Für das Zwischenspeichern von Teilergebnissen bei der Berechnung werden **Hilfs-Variable** eingeführt. Dieses Verfahren bedeutet zwar ein Aufbrechen der klassischen Postfix-Notation, ist jedoch hilfreich, um z.B. bei Zugriff auf Datenfelder Schlüssel angeben zu können, die zuvor aus mehreren Datenfeldern oder Text-Konstanten zusammengesetzt wurden. Zusätzlich ist hiermit die Möglichkeit geschaffen, Zwischenergebnisse über den gesamten Berechnungsvorgang zu halten, wodurch sich komplizierte Formeln und eine mehrfache Berechnung desselben Zwischenergebnisses vermeiden lassen.

Die Hilfs-Variable bei ihrer ersten Verwendung mit dem Operator "=" angelegt und erhält automatisch den Typ des zugewiesenen Wertes. Bereits vorhandene Hilfs-Variable werden überschrieben, wobei auch ein Wechsel des Typs zulässig ist. Die Kennzeichnung einer Hilfs-Variablen erfolgt wie folgt:

H(nn)

H Kennzeichen für eine Hilfs-Variable

(und) dienen als Trennzeichen

nn Ist eine maximal 2-stellige dezimale Nummer von 0 bis 99

Beispiel: 1 2 + =H(12) speichert den aktuellen Wert aus dem Akkumulator (1+2=3) in der Hilfs-Variablen 12.

1 H(12) + addiert den zuvor gespeicherten Wert aus der Hilfs-Variablen 12 auf die Konstante. Neuer Wert des Akkumulators ist also 1+3=4.

System-Variable

sind interne Informationen des GA-Knoten und werden mit einer ähnlichen Syntax wie die Hilfs-Variable bereitgestellt, wobei nur ein lesender Zugriff möglich ist:

S(ss)

S

Kennzeichen für eine System-Variable

(und) dienen als Trennzeichen

ss

Ist eine der nachfolgend beschriebenen Kennungen:

T_1900 Tag, absolut seit 1.1.1900

T_J Tag, relativ zum aktuellen Jahr

W_J Woche, relativ zum aktuellen Jahr

W_M Woche, relativ zum aktuellen Monat

T_M Tag, relativ zum aktuellen Monat

T_W Tag, relativ zur aktuellen Woche

S_W Stunde, relativ zur aktuellen Woche

S_T Stunde, relativ zum aktuellen Tag

M_T Minute, relativ zum aktuellen Tag

M_S Minute, relativ zur aktuellen Stunde

WZ_M Wartezeit in Minuten

WZ_S Wartezeit in Sekunden

AT_M Anzahl der Tage im aktuellen Monat (28.0, 29.0, 30.0 oder 31.0)

AT_J Anzahl der Tage im aktuellen Jahr (365.0 oder 366.0)

UTC Zeitverschiebung gegenüber der "Universal Time Coordinated" (Greenwich Meantime; 1.0 = MEZ oder 2.0 = MESZ)

WHILE Anzahl der Schleifendurchläufe für die WHILE-Schleife, in der sich die aktuelle Abfrage befindet (=1.0 nach Einsprung in

die Schleife)
FEHLER Fehler-Nummer, die von der letzten durchgeführten Operation bzw. Funktion erzeugt wurde. Die Kodierung wird später vorgegeben. Ist kein Fehler aufgetreten, so wird "0" (=FALSE) geliefert.

Die Werte werden im Datenformat Real*4 angegeben. Für die Angaben zur **aktuellen Systemzeit** wird diese gemäß den FND-Dimensionen mit den Kodierungen 160 - 169 kodiert (vgl. FND-Ergänzungen), d.h. Nachkommastellen sind als Dezimalwerte bezogen auf die angegebene Dimension zu interpretieren.

Beispiel: S(T_J)=12.25 entspricht dem 12. Januar des aktuellen Jahres um 6.00 Uhr
S(T_M)=12.25 entspricht dem 12. des aktuellen Monats um 6.00 Uhr
S(S_T)=12.25 entspricht 12.15 Uhr

4.4.6.2 Konstante

Konstante werden wie unter ANSI-C angegeben, d.h. alphanumerische Zeichenfolgen werden durch eingrenzende Doppel-Hochkomma (z.B. "abcd ...") gekennzeichnet. Es stehen auch die ESCAPE-Sequenzen zur Verfügung (Syntax für Sonderzeichen beginnend mit "\"). Eine "leere" Zeichenfolge entspricht ("").

Hinweis: Einzelne alphanumerische Zeichen werden wie eine Zeichenfolge behandelt. Auf diese Weise wird die verwirrende Behandlung des Datentyps "char" in C vermieden (z.B. if('C' == 67) liefert TRUE!).

Die numerischen und binären Konstanten werden als dezimale (ddd) oder hexadezimale Zahl (0xdd) angegeben. Dabei wird bei den binären Konstanten die Bitkodierung verwendet.

Beispiele für zulässige Konstanten:

alphanumerisch: "abcd", ""
numerisch: 1, -1, 1.234, -1.234, 1.234E12
binär: 1 ... 255, 0x00 ... 0xff

Für die Datenfelder "mask" und "info_event" im DP_#0 müssen auch interne Bitkombinationen zugreifbar sein (vgl. FND-Spezifikation unter 2.3 ff., 2.7 und 2.8). Die zugehörigen Namen für diese Datenfelder wurden bereits oben beschrieben (vgl. "Variablen"). Um die Verwendung dieser Datenfelder im Ereignisschaltprogramm zu vereinfachen, werden die für diese Felder möglichen Bitkombinationen als **benannte Konstanten** definiert. Werden diese verwendet, so werden automatisch die nicht verwendeten Bits maskiert, d.h. diese sind nach der Operation unverändert. Folgende benannte Konstanten werden definiert:

- mask_msk_0 Freigabe aller Ereignismeldungen
- mask_msk_1 Ausblendung Betriebsmeldungen
- mask_msk_2 Ausblendung Betriebs- und BTA-Störungsmeldungen
- mask_msk_3 Ausblendung aller Ereignismeldungen

- mask_mis_0 Freigabe aller Ereignismeldungen
- mask_mis_1 Ausblendung Betriebsmeldungen
- mask_mis_2 Ausblendung Betriebs- und BTA-Störungsmeldungen
- mask_mis_3 Ausblendung aller Ereignismeldungen

- mask_apr_0 "access-priority" 0
- mask_apr_1 "access-priority" 1
- mask_apr_2 "access-priority" 2
- mask_apr_3 "access-priority" 3

- mask_val_0 Leitzentralen-Exklusivität
- mask_val_1 Leitzentralen-Präferenz
- mask_val_2 Inselzentralen-Präferenz
- mask_val_3 Inselzentralen-Exklusivität

Die benannte Konstanten für "info_event" und "update_control" beziehen sich auf ein einzelnes Bit mit den Werten "0" und "1" :

- Datenfeld Funktion: "n = 0" // "n = 1"**
- ie_f1_n Geber-Störung: Geber normal // Geber gestört
 - ie_f2_n BTA-Störung: BTA normal // BTA gestört
 - ie_ch_n Wert-Änderung: KEINE // Änderung
 - ie_te_n auslösendes Ereignis (trigger-event):
KEINES // auslösendes Ereignis
 - ie_rl_n Fern/lokal: Fernbedienung // Bedienung vor Ort
 - ie_tl_n Verriegelung: KEINE // temporäre Verriegelung durch Insel
 - ie_wg_n Grenzwert-Kontrolle: Normalbereich // Warnbereich
 - ie_al_n Grenzwert-Kontrolle: Normal- oder
Warnbereich // Alarmbereich
 - ie_lv_n Ebene (Level): unten // oben
 - ie_or_n Obere Grenzwert-Kontrolle: KEINE
Überschreitung // Überschreitung
 - ie_cc_n Monotonie-Kontrolle: kontinuierlich // Setzvorgang oder
Überlauf
 - ie_sv_n Warnung/Alarm: Warnung // Alarm
 - uc_r1_n P_#1 bzw. DP_#1/P_#1: modifizierbar // nur lesbar
 - uc_r2_n P_#2 bzw. DP_#2/P_#1: modifizierbar // nur lesbar
 - uc_r3_n P_#3 bzw. DP_#2/P_#2: modifizierbar // nur lesbar
 - uc_r4_n P_#4 : modifizierbar // nur lesbar

Für "ie_sv" zusammengefaßt mit "ie_or" gilt:

- ie_so_0 Zählwert im Normalbereich
- ie_so_1 Zählwert im oberen Warnbereich
- ie_so_3 Zählwert im oberen Alarmbereich

Die benannte Konstante "UNDEF" entspricht dem ungültigen FND-Wert.

- binär 0xFF
- float/IEEE-754: 0xFFFFFFFF

4.4.6.3 Operatoren, Zuweisung und Funktionen

Für die Berechnung numerischer und binärer Werte dienen die aus ANSI-C bekannten **Operatoren**: für natürliche, ganze und reelle Zahlen

- + Addition
- Subtraktion bzw. arithmetische Negation
- * Multiplikation
- / Division

für binäre Zahlen (bitweise)

- & UND
- | inklusive ODER
- ^ exklusives ODER
- << Links-Shift
- >> Rechts-Shift

für boolesche (logische) Ausdrücke

- ! Negation
- && UND
- || ODER

Als zusätzlicher Operator wird "=" für die **Zuweisung** des aktuellen Wertes aus dem Akkumulator eingeführt. Das Zuweisungsziel (Hilfs-Variable oder Datenfeld) wird dem Operator als Parameter mitgegeben.

Beispiel: 1 2 + =H(12) speichert den aktuellen Wert aus dem Akkumulator (1+2=3) in der Hilfs-Variablen 12.
 1 2 + =D(HHHH:ZGAAMMTT0000IIR.nominal) speichert den Wert unter "nominal" im DP_#1 des Datenpunktes.

Als **Funktionen** werden die in der ANSI-C-Bibliothek definierten verwendet, soweit sie der Bearbeitung numerischer Werte oder Zeichenfolgen dienen. Die Aufrufparameter müssen entsprechend der Reihenfolge im Aufruf auf den Keller (Stack), bzw. der letzte im Akkumulator geladen worden sein. Das Ergebnis steht anschließend im Akkumulator. Dabei werden die Anpassungen an den Datentyp für die Aufruf- und Ergebnis-Parameter implizit vorgenommen, d.h. es wird nur der allgemeine Funktionsname verwendet:

Mathematische Funktionen

Funktionsname	Beschreibung
abs	Absolut-Betrag
exp	Exponential-Funktion (Basis "e")
log	natürlicher Logarithmus (Basis "e")
log10	10er-Logarithmus (Basis 10)
max	Maximum von 2 Werten
min	Minimum von 2 Werten
mod	Modulo-Funktion
pow	Exponential-Funktion (Basis ist Parameter 1)
sqrt	Quadrat-Wurzel

Trigonometrische Funktionen

Funktionsname	Beschreibung
sin	Sinus
cos	Cosinus
tan	Tangens
asin	Arcus-Sinus
acos	Arcus-Cosinus
atan	Arcus-Tangens
sinh	Sinus-Hyperbolicus
cosh	Cosinus-Hyperbolicus
tanh	Tangens-Hyperbolicus

Funktionen für Zeichenfolgen

Funktionsname	Beschreibung
strcat	Zusammensetzen von 2 Zeichenfolgen
strcmp	Vergleich von 2 Zeichenfolgen
strlen	Länge einer Zeichenfolge

Speziell für die Ereignis-Schaltprogramme werden die folgenden Funktionen eingeführt:

- sp_ewarte** (**logischer Ausdruck**) wartet auf ein Ereignis
- sp_etyf** liefert den Typ der Variablen, deren Ereignis "sp_ewarte" beendet
- sp_evar** liefert den Namen der Variablen, deren Ereignis "sp_ewarte" beendet
- sp_call** ("Schaltprogramm-Kennung")
 startet angegebenes Schaltprogramm und wartet auf dessen Beendigung
- sp_start** ("Schaltprogramm-Kennung") startet angegebenes Schaltprogramm
- sp_stop** (["Schaltprogramm-Kennung"]) stoppt angegebenes Schaltprogramm

sp_exit stoppt nur das aufrufende Ereignisschaltprogramm

sp_printf ("Format-Definition") Textausgabe auf Protokoll-Gerät

sp_abl ([...[["D-Variable"], "D-Variable"]...]) ausblenden v. Datenpkt.

sp_ebl ([...[["D-Variable"], "D-Variable"]...]) einblenden v. Datenpkt.

Die Funktion "**sp_ewarte**" enthält analog zur Fallunterscheidung (IF/THEN/ELSE) einen **logischen Ausdruck**, z.B. eine geschachtelte Vergleichsoperation. Die Vergleichsoperation liefert als Ergebnis den logischen Wert "TRUE" (beendet das Warten) oder "FALSE" (wartet weiter). Liefert die Vergleichsoperation nicht explizit einen logischen Wert, so wird nur ein "leerer" Wert (i.a. 0) des jeweiligen Datentyps als "FALSE" interpretiert, während alle anderen Werte als "TRUE" angesehen werden. Die Überprüfung des logischen Ausdrucks erfolgt jedesmal, wenn sich eine der darin angegebenen Variablen geändert hat. Als Variable kann nur ein Datenpunkt oder eine Systemvariable angegeben sein. Hilfs-Variable sind nur als Vergleichswert zulässig, da sich ihr Wert während des Wartens nicht verändert.

Werden Wartezeiten angegeben (Systemvariable mit Kennung "WZ_M" bzw. "WZ_S"), so beginnt die Wartezeit mit dem Aufruf der Funktion "**sp_ewarte**".

Nach dem Warten kann die Kennung der Variablen, die das Ereignis auslöste, mit der Funktion "**sp_ety**" abgefragt werden (liefert "A", "D" oder "S"). Für die Variablenkennungen "A" und "D" liefert die Funktion "**sp_evar**" die Datenpunkt-Adresse bzw. den Datenpunkt-Identifikator des Datenpunktes der das Ereignis auslöste. Für den Variablentyp "S" wird der Name der Systemvariablen geliefert.

Die Funktionen "**sp_call**", "**sp_start**", "**sp_stop**" und "**sp_exit**" dienen zum programmgesteuerten Starten und Stoppen von Schaltprogrammen. So ist es möglich, abhängig vom Zustand bzw. Wert eines oder mehrerer Datenpunkte Einfluß auf den Ablauf anderer Schaltprogramme zu nehmen. Als Parameter wird die Schaltprogramm-Kennung angegeben. Im Unterschied zu "**sp_start**" wartet das aufrufende Ereignisschaltprogramm nach dem Aufruf von "**sp_call**" auf die Beendigung des aufgerufenen Schaltprogrammes. Gehört zum aufrufenden Schaltprogramm auch ein parallel ablaufendes Zeitschaltprogramm, so wartet dieses nicht. Dadurch ist es möglich, Schaltprogramme modulweise zu erstellen, d.h. einmalig erstellte Schaltprogramme können mehrfach verwendet werden (Schaltprogramm-Bibliothek). Ist für "**sp_stop**" keine Schaltprogramm-Kennung angegeben, so wird das laufende Schaltprogramm durch sich selbst gestoppt (Zeit- und Ereignisschaltprogramm).

Für die Funktion "**sp_exit**" gibt es keine Aufrufparameter. Ihr Aufruf beendet das Ereignisschaltprogramm. Das Ereignisschaltprogramm wird gestartet, z.B. zyklisch, oder durch ein anderes Ereignisschaltprogramm mit "**sp_start**", oder durch den Schaltprogramm-Kalender. Das Zeitschaltprogramm ist im Gegensatz zum Aufruf "**sp_stop**" nicht betroffen. Die Funktion "**sp_printf**" dient zur Ausgabe einer Meldung auf dem Protokoll-Gerät. Die Syntax der "Format-Definition" entspricht der Syntax für die Funktion "printf" aus ANSI-C. Darin angegebene Ausgabe-Parameter für die mit "%" gekennzeichneten Formatfelder müssen zuvor in der Reihenfolge ihres Auftretens auf den Keller geladen worden sein.

Die Funktionen "**sp_abl**" und "**sp_ebl**" enthalten als Aufrufparameter eine beliebige Anzahl von Datenpunkt-Variablen, die durch Komma getrennt sind. Die Syntax entspricht der zur Funktion "**sp_ewarte**" beschriebenen Variablen, jedoch beschränkt auf die Datenpunkte. Zusätzlich können zeichen- und text-sensitive Platzhalter ("?" bzw. "*") verwendet werden.

Dadurch ist es z.B. möglich, alle Datenpunkte einer Anlage ein- bzw. auszublenden.

4.4.6.4 Vergleiche

Für **Vergleiche** stehen die unter C bekannten Operatoren zur Verfügung:

<	kleiner
>	größer
==	gleich
!=	NICHT gleich
<=	kleiner oder gleich
>=	größer oder gleich

Jeder Vergleich liefert einen binären Wert, wobei dem Wert 0 der boolesche/logische Wert FALSE zugeordnet wird. Alle anderen Werte entsprechen dem booleschen/logischen Wert TRUE.

Der Vergleich von Zeichenfolgen erfolgt mit der aus C bekannten Funktion "strcmp", die einen numerischen Wert (< 0, 0, > 0) als Ergebnis liefert, der dann mit den Vergleichsoperatoren geprüft werden kann.

4.4.6.5 Fallunterscheidungen (IF/THEN/ELSE)

Für die **Fallunterscheidungen** stehen schachtelbare IF/THEN/ELSE-Strukturen zur Verfügung:

```
?if ([log. Ausdruck]) Befehlsfolge 1 [ ?else Befehlsfolge 2 ] ;  
? Kennzeichen für eine Fallunterscheidung mit folgender Befehlsfolge
```

Der **logische Ausdruck** ist das Ergebnis einer geschachtelten Vergleichsoperation mit dem Wert "TRUE" (startet **Befehlsfolge 1**) oder "FALSE" (startet **Befehlsfolge 2**). Liefert die Vergleichsoperation nicht explizit einen logischen Wert, so wird nur ein "leerer" Wert des jeweiligen Datentyps (i.a. "0") als "FALSE" interpretiert, während alle anderen Werte als "TRUE" angesehen werden.

Ist innerhalb der Klammern kein Ausdruck angegeben, so wird der aktuelle Wert im Akkumulator auf "leer" geprüft.

; Kennzeichen, daß die Fallunterscheidung beendet ist.

4.4.6.6 Wiederholungen (WHILE-Schleifen)

Für die **Wiederholung** stehen schachtelbare WHILE-Schleifen zur Verfügung. Als Abbruchkriterium können eine oder mehrere Hilfs-Variablen verwendet werden, die innerhalb der Befehlsfolge modifiziert werden. Vor dem Aufruf muß eine Initialisierung der Hilfs-Variablen den korrekten Schleifenstart sicherstellen. Die Anzahl der Schleifendurchläufe wird nicht begrenzt.

Die Systemvariable wird auf den Wert $2^{32}-2$ begrenzt. Darüber hinaus wird nur noch der Wert $2^{32}-1$ (0xFFFFFFFF) geliefert, dem die benannte Konstante "UNDEF" ebenfalls gleichgesetzt ist. Die Anzahl der Schleifendurchläufe kann über die Systemvariable "S(WHILE)" abgefragt werden. Es wird folgende Syntax verwendet:

```
?while ([log. Ausdruck]) Befehlsfolge 1;
```

*** Bestandteile analog zu IF/THEN/ELSE-Struktur ***

Beispiel: ?WHILE(S(WHILE) 5 <) 1 S(WHILE) +=H(12); speichert immer die Anzahl der Schleifendurchläufe + 1 in der Hilfs-Variablen 12.

4.4.6.7 Kommentare

Zur Dokumentation können Kommentare eingefügt werden. Diese dürfen nur einzeilig sein und müssen der folgenden Syntax genügen:

```
!([Kommentar]);
```

! Kennzeichen für den Anfang eines Kommentares Fallunterscheidung mit folgender Befehlsfolge:

Kommentare dürfen nicht geschachtelt werden. Wird die Zeichenfolge "!(
innerhalb eines Kommentares oder einer alphanumerischen Konstanten
angegeben, so wird sie als Bestandteil des Kommentares bzw. der
Konstanten und nicht als Anfang eines Kommentares interpretiert.

4.4.7 FND-Konvention für das Ereignis-Schaltprogramm

Als Datenpunkt-Adresse für die Übertragung des Ereignisschaltprogrammes wird die Datenpunkt-Adresse "**\$SCHALTPROG-oooo**" verwendet ("oooo" entspricht der Objektkennung, wie sie auch innerhalb der LZH verwendet wird), die mit keiner der anderen Datenpunkt-Adressen einen Konflikt bildet. Durch Verwendung des Karteipunktes (Datenpunkt-Typs "7") unter dp_type im Header.2 und der **Kartei-Informationsebene 2** werden die einzelnen Zeilen des Ereignisschaltprogrammes im DP_2 Übertragen.

Ereignisschaltprogramme werden zeilenweise von der LZH übertragen. Wird die maximale Zeilenlänge von 255 Zeichen überschritten, so wird eine solche Zeile an dem letzten Trennzeichen (* Leerzeichen ' ') aufgebrochen, das noch übertragen werden kann. Die restlichen Zeichen der Zeile werden als Folgezeile bzw. Folgezeilen übertragen.

Das Zeitschaltprogramm hat den folgenden Aufbau:

Tabellenindex: Reserviert für zukünftige Erweiterungen. Momentan stets 1.0 explizit angegeben (mode_i.itm=1).

Zeilenindex: Laufende Zeilennummer, beginnend mit 1.0 in Inkrementen von 1.0 für jede Zeile (d.h. ganzzahliger Anteil) und 0.01 für jede Folgezeile (d.h. Nachkommastellen), wird explizit angegeben (mode_j.itm=1).

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = Länge der zu übertragene Zeile

<u>Spalte</u>	<u>Byte</u>	<u>Bezeichnung</u>	<u>Format</u>
1:	1 ... y	Zeile der Befehlsfolge	y Byte

4.4.8 Parametrierung für den Schaltprogramm-Kalender

In den Schaltprogramm-Kalender werden Zeitpunkte eingetragen, an denen Schaltprogramme gestartet oder gestoppt werden sollen.

4.4.9 FND-Konvention für den Schaltprogramm-Kalender

Als Datenpunkt-Adresse für den Schaltprogramm-Kalender wird die Datenpunkt-Adresse "**\$SCHALT-KAL-oooo**" verwendet ("oooo" entspricht der Objektkennung, wie sie auch innerhalb der LZH verwendet wird), die mit keiner der anderen Datenpunkt-Adressen einen Konflikt bildet. Durch Verwendung des Karteipunktes (Datenpunkt-Typs "7") unter dp_type im Header.2 und der **Kartei-Informationsebene 0** werden die Schaltprogramm-Parameter im DP_2 übertragen.

Der DP_2 des Karteipunktes für den Schaltprogramm-Kalender hat den folgenden Aufbau:

Tabellenindex: Datum für Start bzw. Stop des Schaltprogrammes (IEEE-754 mit Dimension 160) wird explizit angegeben

(mode_i.itm=1).
 Zeilenindex: Zeit für Start bzw. Stop des Schaltprogrammes (IEEE-754 mit Dimension 167, bezogen auf das im Tabellenindex angegebene Datum) wird explizit angegeben (mode_j.itm=1).
 Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird, d.h. strng_len = 72.

Sp.	Byte	Bezeichnung	Format
1:	1...10	Schaltprogramm-Kennung	10 Byte
2:	11	Flag für Start/Stop-Funktion	1 Byte
3:	12...71	Beschreibung	60 Byte
4:	72	Flag für Protokollierung	1 Byte

Zu 1: Schaltprogramm-Kennung des Schaltprogrammes, das gestartet bzw. gestoppt werden soll.
 Zu 2: Ist das Flag gesetzt: Das Schaltprogramm wird zum Zeitpunkt aus Tabellen- und Zeilenindex gestartet, SONST: gestoppt.
 Zu 3: Freier Text zur Beschreibung, warum das Schaltprogramm gestartet bzw. gestoppt werden soll. Dieser Text wird für eine Protokoll-Meldung verwendet, wenn das Flag unter 4: gesetzt ist.
 Zu 4: Ist das Flag gesetzt: Es erfolgt eine Protokoll-Meldung, wenn das Schaltprogramm aufgrund dieses Eintrags gestartet bzw. gestoppt wurde. SONST: Keine Protokoll-Meldung

4.5 Komplexere Abfragen

Insbesondere bei größeren Objekten kommt es vor, daß zur Feststellung des aktuellen Zustands die Funktion des FND-Sammeladepunktes nicht ausreicht und die explizite Istwertabfrage aller Datenpunkte zu aufwendig ist. Notwendig ist eine **allgemeine Generalabfrage**, d.h. die **Selektion aller oder einer Gruppe von Datenpunkten** durch Verwendung von Platzhaltern (Wildcards, Templates) bei der Angabe der Datenpunktadresse (zeichen- und textsensitiv, z.B analog zu "?" und "*" in MS-DOS bei der Definition von Dateinamen). Können die abzufragenden Datenpunkte nicht durch Verwendung einer mit Platzhaltern versehenen Datenpunkt-Adresse definiert werden (z.B. Auswahl mehrerer explizit angegebener Datenpunkte), so kann auch eine Auflistung mehrerer Datenpunkt-Adressen übertragen werden.

Die Syntax für die Datenpunkt-Adresse im Header.1 erlaubt nicht die Angabe von Platzhaltern. Aus diesem Grunde wird der Abfrage-Funktion ein Karteipunkt (Datenpunkt-Typ 7) mit der Datenpunkt-Adresse **"\$ISTWERTE-oooo"** zugeordnet ("oooo" entspricht der Objektkennung, wie sie auch innerhalb der LZH verwendet wird). Die Angabe der abzufragenden Datenpunkt-Adressen mit und ohne Platzhaltern) und den zugehörigen Parametern erfolgt dann im übertragenen Datensatz. Bei der Abfrage liefert die LZH im Feld "lz_subid" im Header.1 eine Kennung, die die Abfrage zusammen mit der "lz_id" eindeutig identifiziert.

Durch die Abfrage wird eine nicht berechenbare Anzahl von Istwert-Meldungen ausgelöst. Diese erfolgen im Format entsprechend dem jeweiligen Datenpunkt-Typ (1 bis 5), wie er in der FND-Spezifikation Version 1.0 festgelegt ist oder im Blockformat. Dabei wird zusätzlich das "te-Bit" im "info_event" gesetzt (vgl. "Ergänzungen zur FND-Spezifikation"). Zusätzlich wird ein Bezug zwischen Istwert-Meldung und der auslösenden Abfrage hergestellt, indem die Istwert-Meldungen

dieselbe "lz_id" und "lz_subid" verwenden, wie sie bei der Abfrage angegeben wurden. Dadurch kann die LZH diese Meldungen dem Benutzer zuordnen, der die Abfrage vornahm.

Eine laufende Istwert-Abfrage muß jederzeit gestoppt werden können. Zu diesem Zweck wird eine erneute Abfrage unter Verwendung derselben "lz_id" und "lz_subid" an den GA-Knoten gesendet. Anstatt der mit Platzhaltern definierten Datenpunkt-Adressen enthält der Datensatz jedoch die Zeichenfolge "**ISTWERTE_STOP**". Der GA-Knoten beendet daraufhin alle zu "lz_subid" laufenden Istwert-Abfragen.

4.5.1 Aktivierungszeit der Istwert-Abfrage

Durch Angabe eines **Zeitpunktes (Datum/Zeit)**, an dem eine Abfrage aktiv werden soll, kann der Start der Istwert-Abfrage im voraus hinterlegt werden. Der GA-Knoten speichert alle hinterlegten Abfragesätze. Gelöscht wird ein Abfragesatz automatisch dann, wenn er abgearbeitet ist bzw. durch den Benutzer explizit gestoppt wird.

4.5.2 Parametrierung der Istwert-Abfrage

Mittels der **Zykluszeit** kann eine Istwert-Abfrage so parametrierung werden, daß eine zyklische Übertragung der abgefragten Istwerte erfolgt. Die Angabe eines **festen Datums/Uhrzeit** stellt sicher, daß in regelmäßigen Abständen die Daten übertragen werden, unabhängig davon, ob die Bedingung entsprechend der Zykluszeit erfüllt ist. Damit wird eine Synchronisation erreicht, wodurch auch längerfristig Meßwerte aus vergleichbaren Zeiträumen entstehen.

Durch Angabe der **Verbindungsdauer** kann eine Optimierung der Verbindungskosten erreicht werden. Bei Zykluszeiten, die kürzer als der verfügbare Zeittakt sind, bietet sich an, die gewählte Verbindung bestehen zu lassen. Leerzeiten werden für die Übertragung der zwischengespeicherten Daten im Blockformat genutzt, die Istwert-Abfrage wird dabei jedoch vorrangig behandelt. Auch bei einmaligen Istwertabfragen ist es sinnvoll, eine Verbindungszeit entsprechend dem Zeittakt zu wählen, um diesen optimal zu nutzen. Eine bestehende gewählte Verbindung wird beendet, wenn ein neuer Abfragesatz aktiv wird, der eine andere Verbindungszeit definiert bzw. keine Istwert-Abfrage mehr aktiv ist.

4.5.3 FND-Konvention für den Abfragesatz

Als Datenpunkt-Adresse für den Abfragesatz wird eine noch festzulegende Kodierung verwendet, die mit keiner der anderen Datenpunkt-Adressen einen Konflikt bildet. Durch Verwendung des Karteipunktes (Datenpunkt-Typs "7") unter dp_type im Header.2 und der **Kartei-Informationsebene 0** werden die Abfrageparameter im DP_2 übertragen.

Der DP_2 des Karteipunktes für den Abfragesatz hat den folgenden Aufbau:

Tabellenindex: Datum der Aktivierungszeit (IEEE-754 mit Dimension 160) wird explizit angegeben (mode_i.itm=1). Ist ein Wert < 0 vorbesetzt, so erfolgt die Aktivierung sofort.

Zeilenindex: Werden mehrere Datenpunkt-Adressen innerhalb einer Abfrage definiert, so werden die Daten mit inkrementiertem Zeilenindex beginnend bei 1.0 nacheinander übertragen; explizit angegeben (mode_j.itm=1).

Spaltenindex: stets 1 (mode_k.itm=0), da der nachfolgend beschriebenen Datenstruktur ein "string" gleicher Länge überlagert wird,

d.h. strng_len = 29

Spalte	Byte	Bezeichnung	Format
1:	1 ... 16	Datenpunkt-Adresse	16 Byte
2:	17 ... 20	Zykluszeit	IEEE-754 mit Dimension 13
3:	21		Sync.-Code 1 Byte
4:	22 ... 25	Sync.-Zeit	IEEE-754 mit Dimension zu 3
5:	26 ... 29	Verbindungsdauer	IEEE-754 mit Dimension 13 (Minuten)

- Zu 1: Angabe der Datenpunkt-Adresse gemäß FND-Spezifikation Version 1.0. Zusätzlich werden als gültige Zeichen "?" als zeichen- und "*" als textsensitiver Platzhalter zugelassen. Die Datenpunktadresse wird nach "*" mit binären Nullen (NUL) auf 16 Bytes analog zu der FND-Spezifikation ergänzt.
- Zu 2: Zulässig sind nur Werte > 0. Ungültige Werte, d.h. *0 werden so interpretiert, daß die Zykluszeit nicht berücksichtigt wird, d.h. die Istwerte werden nur einmal übertragen.
- Zu 3: Dimensions-Code für die Synchronisationszeit unter 4:.
- Zu 4: Zeitpunkt mit Dimension aus 3:, an dem generell eine Übertragung vorgenommen werden soll, unabhängig davon, ob die Bedingung entsprechend der Zykluszeit erfüllt ist. Für Werte < 0 wird keine Synchronisation durchgeführt.
- Zu 5: Zulässige Zeitangaben sind Werte > 0. Werte * 0 werden so interpretiert, daß die Verbindung sofort nach der Übertragung der angeforderten Istwerte wieder abgebrochen wird. Wird +* (vgl. IEEE-754) und eine gültige Zykluszeit angegeben, so bleibt Verbindung bestehen.

Anhang

A Authentifizierung

Diese Funktion wird von der LHM nicht genutzt, Bei Bedarf vergleichen Sie bitte den entsprechenden Anhang in der FND-Spezifikation.

B Prozedur-Elemente

Bei Bedarf vergleichen Sie bitte den entsprechenden Anhang in der FND-Spezifikation.

C Netzwerk-Interface

Bei Bedarf vergleichen Sie bitte den entsprechenden Anhang in der FND-Spezifikation.

D Spezielle Transferpunkte

Diese Funktion wird von der LHM nicht genutzt, Bei Bedarf vergleichen Sie bitte den entsprechenden Anhang in der FND-Spezifikation.

E Attribute und Dimensionen

Die nachstehenden Tabellen geben die Belegung der in den Datenpunkt-Beschreibungen mit „atb_id“ (für Melde- und Schaltpunkte) bzw. „dimension“ (für Mess-, Stell- und Zählpunkte) bezeichneten Variablen wieder. Sie sind dabei als Grundversion zu verstehen, die sich durch Erweiterung dem Anforderungsprofil in flexibler Weise anpassen lassen.

Hinweis: Die LHM setzt ausschließlich die nachfolgend angegebenen Codierungen ein, um projektspezifische Vereinbarungen mit ihren aufwendigen Test- und Dokumentations-Problemen zu vermeiden.

CODIERUNG BINÄRER ATTRIBUTE		
Attributtext		atb_id
STUFE (MODUS)	0	1
STUFE (MODUS)	I	
STUFE (MODUS)	0	2
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	0	3
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	III	
STUFE (MODUS)	0	4
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	III	
STUFE (MODUS)	IV	
STUFE (MODUS)	0	5
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	III	
STUFE (MODUS)	IV	
STUFE (MODUS)	V	
STUFE (MODUS)	0	6
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	III	
STUFE (MODUS)	IV	
STUFE (MODUS)	V	
STUFE (MODUS)	VI	

CODIERUNG BINÄRER ATTRIBUTE		
Attributtext		atb_id
STUFE (MODUS)	0	7
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	III	
STUFE (MODUS)	IV	
STUFE (MODUS)	V	
STUFE (MODUS)	VI	
STUFE (MODUS)	VII	
STUFE (MODUS)	0	8
STUFE (MODUS)	I	
STUFE (MODUS)	II	
STUFE (MODUS)	III	
STUFE (MODUS)	IV	
STUFE (MODUS)	V	
STUFE (MODUS)	VI	
STUFE (MODUS)	VII	
STUFE (MODUS)	VIII	
EIN AUS		10
OFFEN GESCHLOSSEN		11
AUF AB		12
AUF ZU		13
START STOP		14
SETZEN RÜCKSETZEN		15
VOR ZURÜCK		16
AUSGANGSSTELLUNG ENDSTELLUNG		17
TAGBETRIEB NACHTBETRIEB		18

CODIERUNG BINÄRER ATTRIBUTE	
Attributtext	atb_id
SCHNELL LANGSAM	19
HEIZEN KÜHLEN	20
SOMMER WINTER	21
RECHTS LINKS	22
AUTOMATIK HAND	23
AKTIV PASSIV	24
NORMAL ANORMAL	25
NORMAL WARTUNG	26
NORMAL STÖRUNG	27
NORMAL ALARM	28
NORMAL GEFAHR	29
(NORMAL-) BETRIEB INITIALISIERUNG	30
(NORMAL-) BETRIEB OPTIMIERUNG	31
UNTEN MITTE (LSTELLUNG) OBEN	32
AUF MITTE (LSTELLUNG) ZURÜCK	33

CODIERUNG BINÄRER ATTRIBUTE	
Attributtext	atb_id
VOR MITTE (LSTELLUNG) ZURÜCK	34
AUSGANGSSTELLUNG MITTELSTELLUNG ENDSTELLUNG	35
SCHNELL MITTEL LANGSAM	36
SCHNELL LANGSAM AUS	37
HEIZEN NEUTRAL KÜHLEN	38
RECHTS MITTE LINKS	39
RECHTS AUSGANGSSTELLUNG LINKS	40
RECHTS RUHESTELLUNG LINKS	41
RECHTS AUS LINKS	42
AUTOMATIK HAND AUS	43
NORMAL WARNUNG ALARM	44
AUS EIN	45

CODIERUNG PHYSIKALISCHER DIMENSIONEN		
Art	Einheit	dimension
	(dimensionslos)	0
Länge	Millimeter	1
	Meter	2
	Kilometer	3
Fläche	Quadratmeter	6
Volumen	Liter	9
	Kubikmeter	10
Zeit	Sekunde	12
	Minute	13
	Stunde	14
	Tag	15
Kraft	Newton	20
	Kilonewton	21
	Meganewton	22
Druck	Pascal	25
Masse	Milligramm	28
	Gramm	29
	Kilogramm	30
	Tonne	31
Energie, Arbeit	Joule	36
	Kilojoule	37
	Megajoule	38
	Wattstunde	40
	Kilowattstunde	41
	Megawattstunde	42
Wirkleistung	Watt	48
	Kilowatt	49
	Megawatt	50

CODIERUNG PHYSIKALISCHER DIMENSIONEN		
Art	Einheit	dimension
Scheinleistung	Voltampere	51
	Kilovoltampere	52
	Megavoltampere	53
Drehzahl	1 / Sekunde	56
	1 / Minute	57
	1 / Stunde	58
Winkel	Sekunde	60
	Minute	61
	(Alt) Grad	62
	Neugrad	63
	Radiant	64
Geschwindigkeit	Millimeter / Sekunde	68
	Meter / Sekunde	69
	Millimeter / Minute	70
	Meter / Minute	71
	Kilometer / Minute	72
	Millimeter / Stunde	73
	Meter / Stunde	74
	Kilometer / Stunde	75
Volumenstrom	Liter / Sekunde	80
	Liter / Minute	81
	Liter / Stunde	82
	Kubikmeter / Sekunde	83
	Kubikmeter / Minute	84
	Kubikmeter / Stunde	85
Massenstrom	Gramm/Sekunde	90
	Kilogramm / Sekunde	91
	Tonne / Sekunde	92
	Gramm / Minute	93
	Kilogramm / Minute	94
	Tonne / Minute	95
	Gramm / Stunde	96
	Kilogramm / Stunde	97
Tonne / Stunde	98	

CODIERUNG PHYSIKALISCHER DIMENSIONEN		
Art	Einheit	dimension
Drehmoment	Newtonmeter	100
	Kilonewtonmeter	101
	Meganewtonmeter	102
Temperatur	Grad Celsius	108
Temperatur-Differenz	Kelvin	109
Entropie	Joule / Kelvin	112
	Kilojoule / Kelvin	113
	Megajoule / Kelvin	114
Enthalpie	Joule / Kilogramm	115
	Kilojoule / Kilogramm	116
	Megajoule / Kilogramm	117
Elektrische Spannung	Volt	120
	Kilovolt	121
Elektrischer Strom	Milliampere	124
	Ampere	125
	Kiloampere	126
Elektrischer Widerstand	Milliohm	130
	Ohm	131
	Kiloohm	132
	Megaohm	133
Stellung	Prozent	150
relative Feuchte	Prozent	151
absolute Feuchte	Gramm / Kilogramm	152
(keine Dimensionsangabe)		255

Anmerkung:

Code # 255 ist für den Fall reserviert, dass die Dimension entweder unbekannt oder implizit vereinbart ist.

F Fehlerbehandlung

Die Variable „tag“ in den FND-Telegrammen weist jeder FND-APDU eine der drei Kategorien zu:

- NORMAL (tag=0: kein Fehler, Normalfall)
- REJECT (tag=1: Zurückweisung beim Ablauf)
- ERROR (tag=2: syntaktischer/semantischer Fehler)

Bei der Zurückweisung kann ein korrektes FND-Telegramm aufgrund des augenblicklichen Zustands des Datenpunktes (z.B. Verriegelung) momentan nicht ausgeführt werden. Wird dasselbe FND-Telegramm zu einem späteren Zeitpunkt erneut gesendet, so ist eine Ausführung evtl. dann möglich.

Für FND-Telegramme, die wegen eines syntaktischen oder semantischen Fehlers (z.B. unzulässige „fct_id“ oder „dp_id“) abgewiesen werden, macht ein wiederholtes Senden keinen Sinn.

F.1 REJECT-APDU-Format (Ablauffehler)

Ein REJECT-Telegramm ist ein Rsp- oder Ack-Telegramm mit „tag“=1. Der Grund für die Zurückweisung wird im Info-Code wie folgt angegeben:

exc_1	exc_2	Beschreibung
1	0	Unerwarteter Empfang eines Segment-Teil-Telegrammes (seg = ,10'B oder ,01'B)
2	0	Unerwarteter Empfang eines Segment-Start- oder Einzel-Telegrammes in der Reassemblierungsphase (seg = ,11'B oder ,00'B)
3	seq_exp	Unerwartete Segment-Folge-Nummer „seq_nr“; „seq_exp“ gibt dabei die erwartete Nummer an
4	0	„invoke_id“ noch belegt; Bearbeitung eines (anderen) Vorgangs noch nicht abgeschlossen
5	0 1 2	Datenpunkt gegenwärtig für FND nicht verfügbar Datenpunkt lokal bedient Datenpunkt verriegelt
6	0	Datenpunkt noch belegt; Bearbeitung eines (anderen) Vorgangs noch nicht abgeschlossen
7	0	LZ gegenwärtig nicht verfügbar
8	0	IZ gegenwärtig nicht verfügbar
9	apr	Ungenügende Durchgriffspriorität (prd < apr)
10	0	Authentifizierungsverfahren aktiv

Bei Bedarf vergleichen Sie bitte den entsprechenden Anhang in der FND-Spezifikation.

F.2 ERROR-APDU-Format (syntaktische/semantische Fehler)

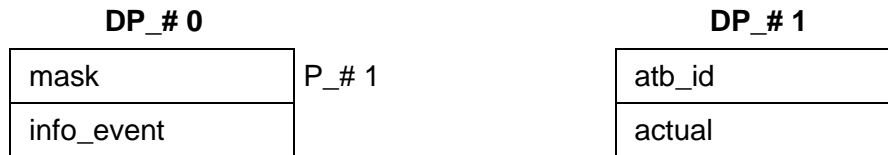
Ein ERROR-Telegramm ist ein Rsp-Telegramm mit „tag“=2. Der Datenteil enthält den APCI des als syntaktisch falsch eingestuften Cmd-Datagrammes. Der Info-Code „exc_1“ bezeichnet das Byte im Datenteil (beginnend mit 0), bei dem der Fehler erkannt wurde. „exc_2“ Zusatzinformationen enthalten.

Bei Bedarf vergleichen Sie bitte den entsprechenden Anhang in der FND-Spezifikation.

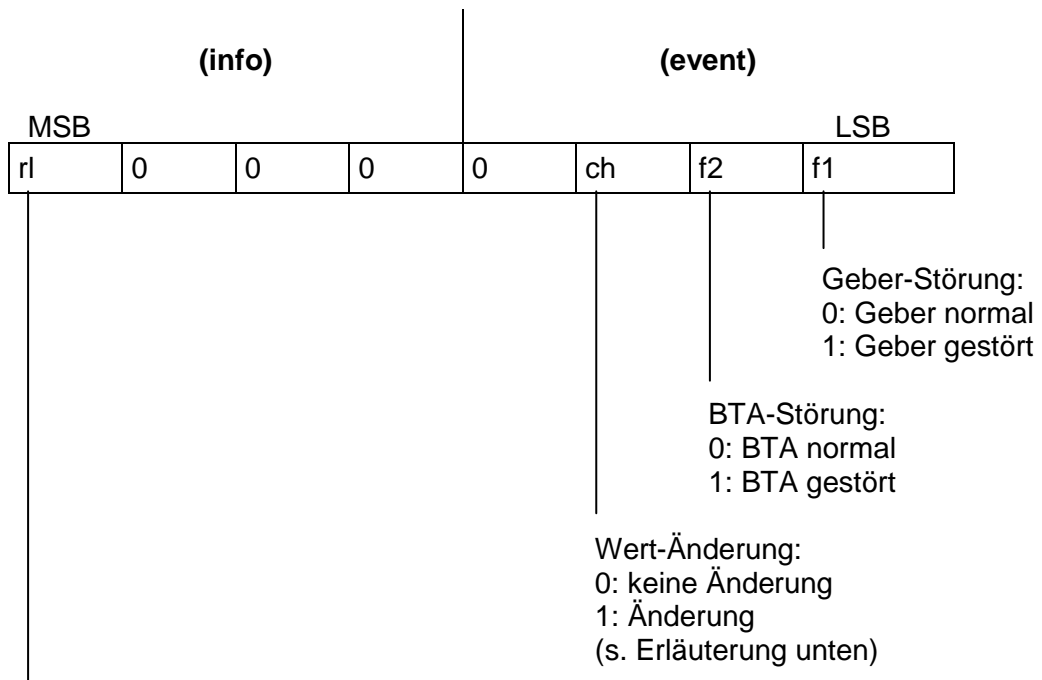
G Übersicht der Datenpunkt-Tabellen

Nachfolgend werden die Datenpunkt-Tabellen kurz zusammengefasst.

G.1 Meldepunkt



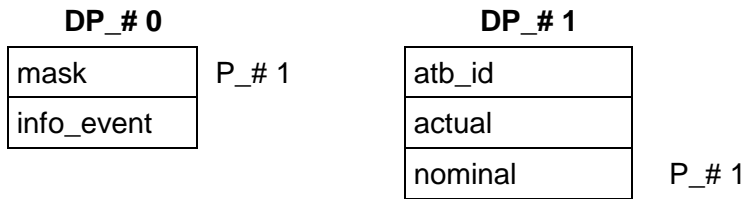
info_event:



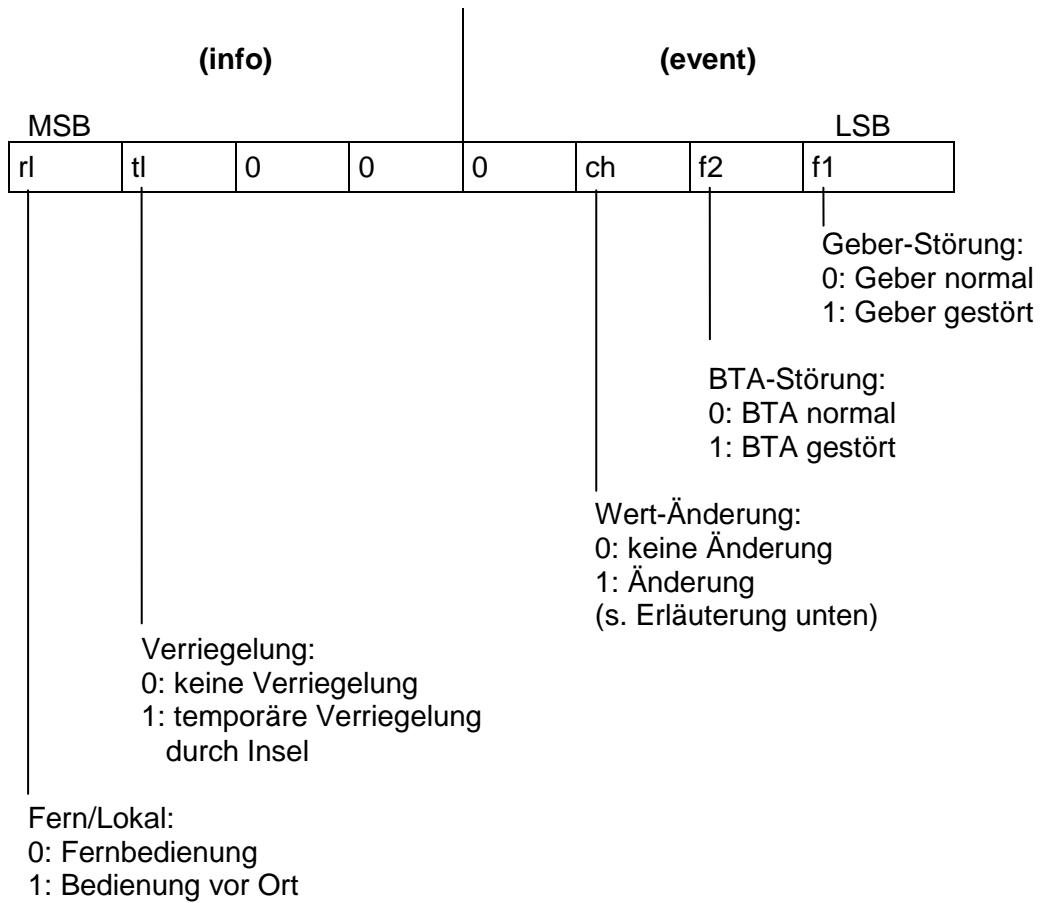
Fern/Lokal:
0: Fernbedienung
1: Bedienung vor Ort

G.2 Schaltpunkt (mit Rückmeldung)

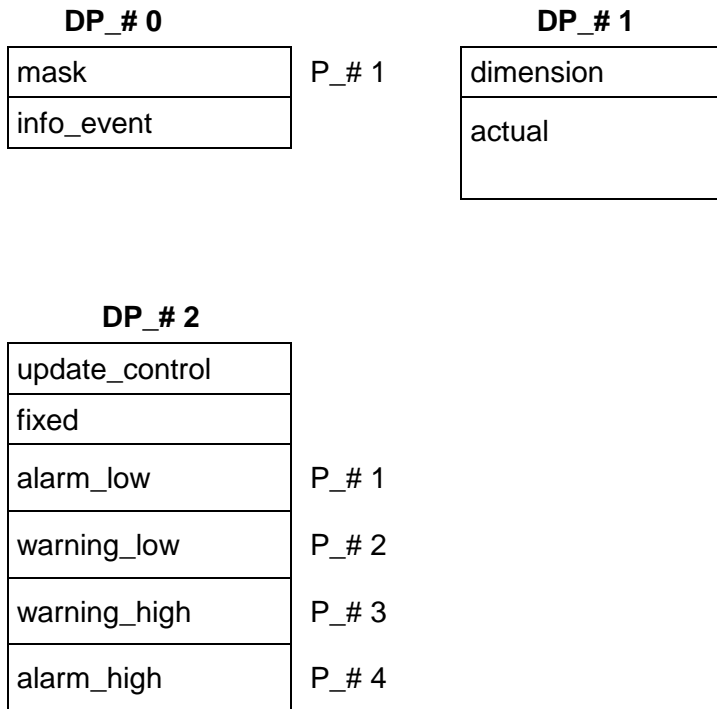
Datenpunkt-Tabellen:



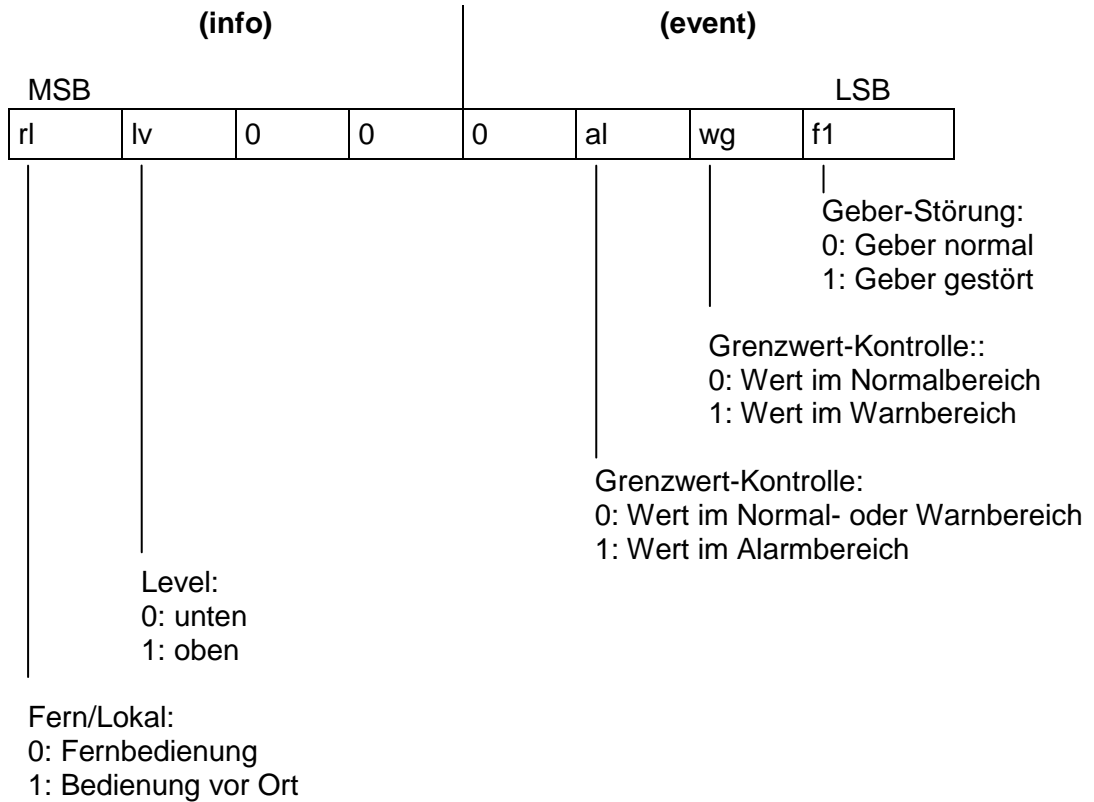
info_event:



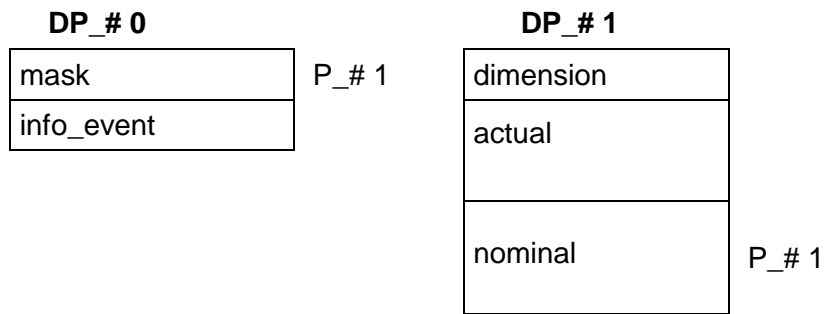
G.3 Messpunkt



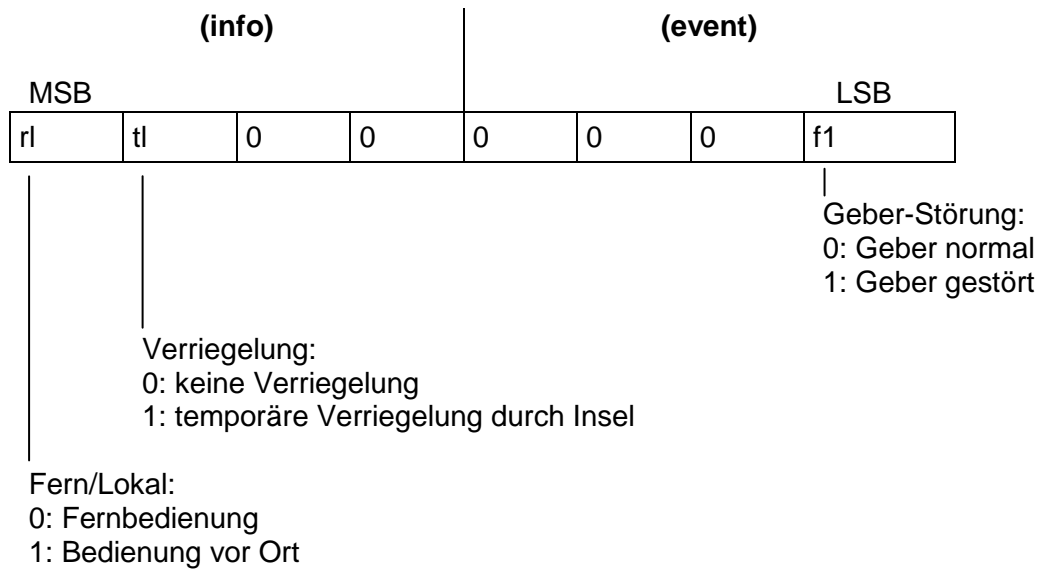
info_event:



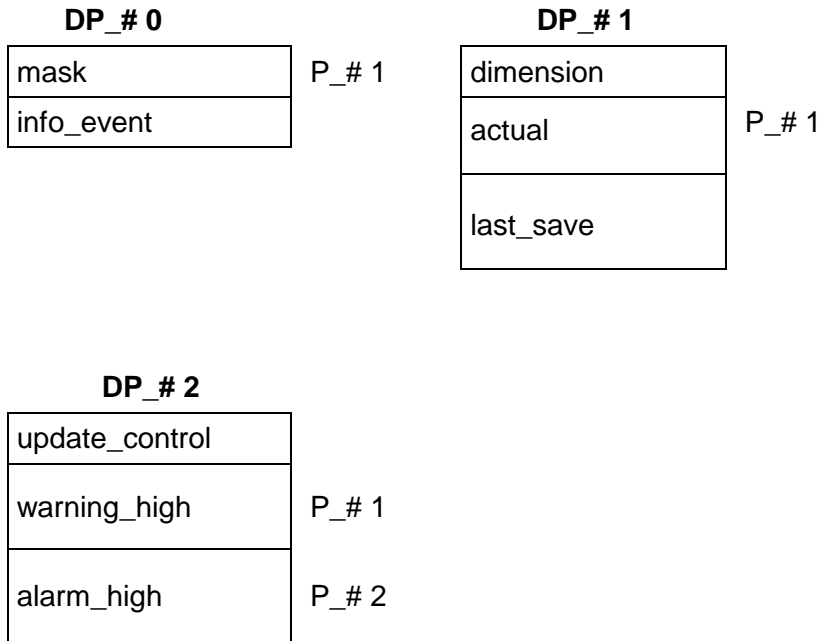
G.4 Stellpunkt



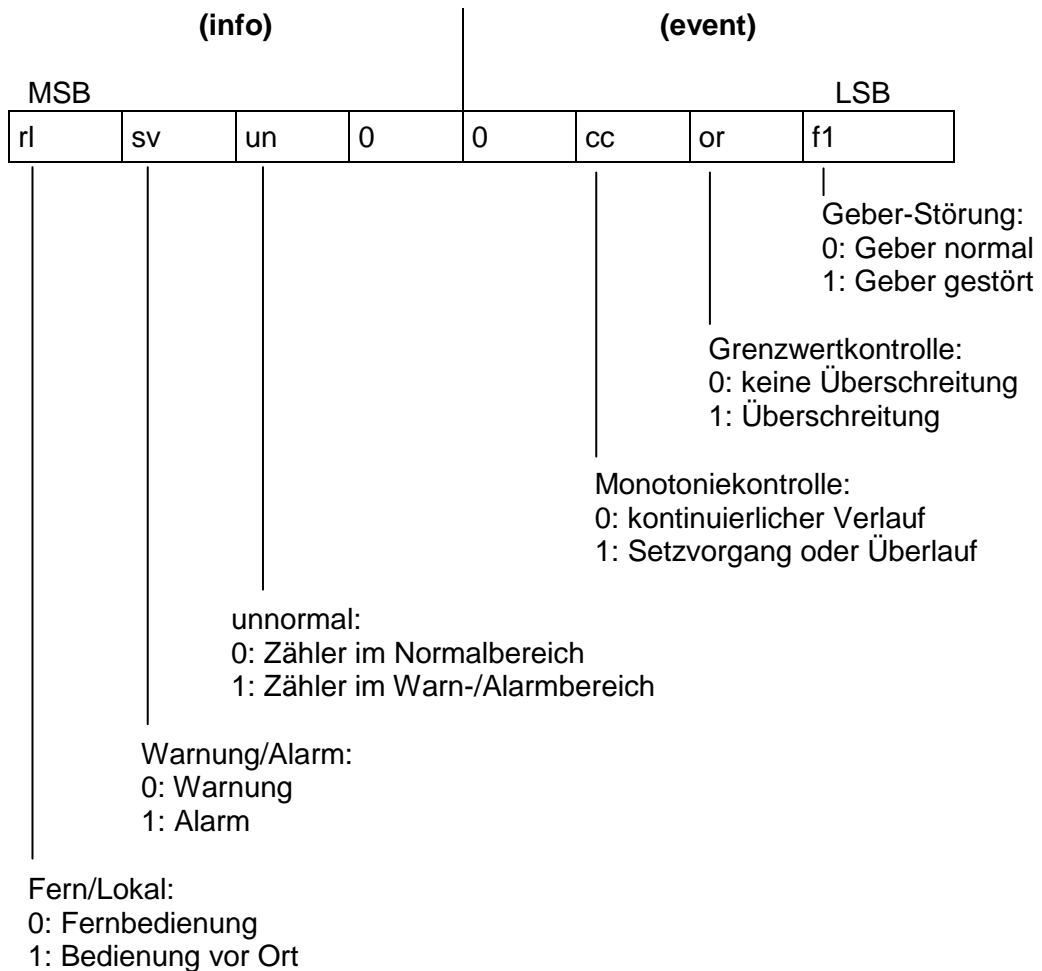
info_event:



G.5 Zählpunkt



info_event:



H Übersicht der Codierungen im FND-APDU

Nachfolgend werden nochmals alle wesentlichen Parameter und ihre Codierungen zum FND-Datagramm zusammengefasst. Die zentrale Rolle übernimmt die Datenpunkt-Adresse.

dp_type	Datenpunkt-Typ
1	Meldepunkt
2	Schaltpunkt (mit Rückmeldung)
3	Meßpunkt
4	Stellpunkt
5	Zählpunkt
6	Sammeladresspunkt
7	Karteipunkt

tab_id	Datenpunkt – Tabelle
0	DP_# 0
1	DP_# 1
2	DP_# 2
3	DP_# 1.DP_#2
4	DP_# 3 (nur Karteipunkt)

fct_id	Funktion
1	Lesen
2	Modifizieren (Schreiben)

d_r	Typ	Erläuterung
0 (00'B)	Cmd	„Command“: Befehl der LZ an eine IZ
3 (11'B)	Rsp	„Response“: Antwort einer IZ auf einen Befehl der LZ
2 (10'B)	Usm	„Unsolicited Message“: Spontanmeldung einer IZ an die LZ
1 (01'B)	Ack	„Acknowledge“: Bestätigung eines Usm-Telegramms

mis/msk	Effekt
0	Freigabe aller Ereignismeldungen
1	Ausblendung der Betriebs-Meldungen; Gemeldet werden: Geber- und BTA-Störungen
2	Ausblendung der Betriebs- und BTA-Störungs-Meldungen; Gemeldet werden: Geber- Störungen
3	Ausblendung aller Ereignismeldungen

I Beispiele für FND-Funktionen

Nachfolgend soll anhand von einigen Beispielen erläutert werden, wie Funktionen der unterschiedlichen Gewerke auf FND-Datenpunkte umgesetzt werden können. Modifizierende Zugriffe (z.B. Schalt- oder Stellbefehle) können dabei sowohl vom GA-Knoten als auch der LZH aus erfolgen.

I.1 Tageslicht- und zeitabhängige Außenbeleuchtung

Aufgabe: Die Außenbeleuchtung soll nur an Werktagen außerhalb der Nachtstunden momentan von 22:00 bis 5:30 abhängig vom Dämmerungsschalter eingeschaltet werden. Die Zeitpunkte sollen bei Bedarf von der LZH aus verändert werden können. Zur Prüfung der Einstellungen des Dämmerungsschalters soll der aktuelle Zustand der Außenbeleuchtung und des Dämmerungsschalters an die LZH gemeldet werden.

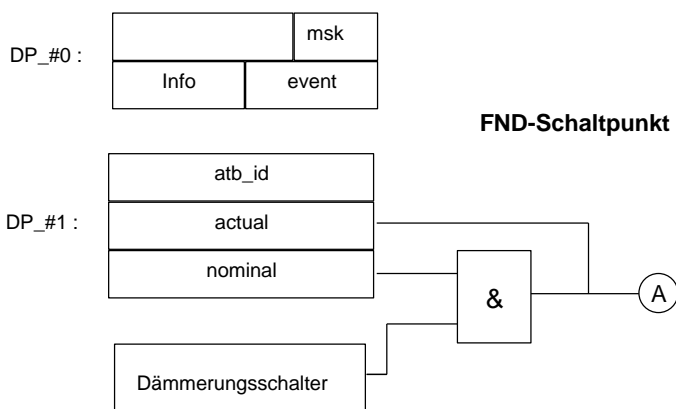
Lösung: Der Außenbeleuchtung wird ein FND-Schaltpunkt mit dem Attribut AUS/EIN (atb_id = 45) zugeordnet, dessen Vorgabewert innerhalb der Steuerung logisch mit dem Dämmerungsschalter verknüpft wird, so dass die Außenbeleuchtung nur dann eingeschaltet wird, wenn beide eingeschaltet sind.

Der aktuelle Zustand wird auf den „actual“ abgebildet.

Im GA-Knoten wird ein Zeitschaltprogramm aktiviert, das an Werktagen um 5:30 den Schaltbefehl EIN und um 22:00 den Schaltbefehl AUS sendet.

Dem nachfolgenden Funktionsplan ist die Zuordnung der einzelnen Ein-/ Ausgänge zum FND-Schaltpunkt zu entnehmen.

Tageslichtabhängige Steuerung der Außenbeleuchtung (ohne Rückmeldekontakt)



Minimalbetrieb: Tageslichtabhängiger Betrieb (nominal = 1)
(z.B. Ausfall GA-Knoten)

Initialisierung: nominal = 1
(Einschalten des LON)

I.2 3-stufige Beleuchtung mit Orientierungslicht

Aufgabe: In jedem Bereich einer großen Halle ist eine Lampe mit 3 Beleuchtungsmittel vorhanden, die über getrennt Stromkreise unabhängig voneinander geschaltet werden können. Ist die Hallenbeleuchtung ausgeschaltet, so muss in jedem Bereich ein gesondert installiertes Orientierungslicht eingeschaltet werden.

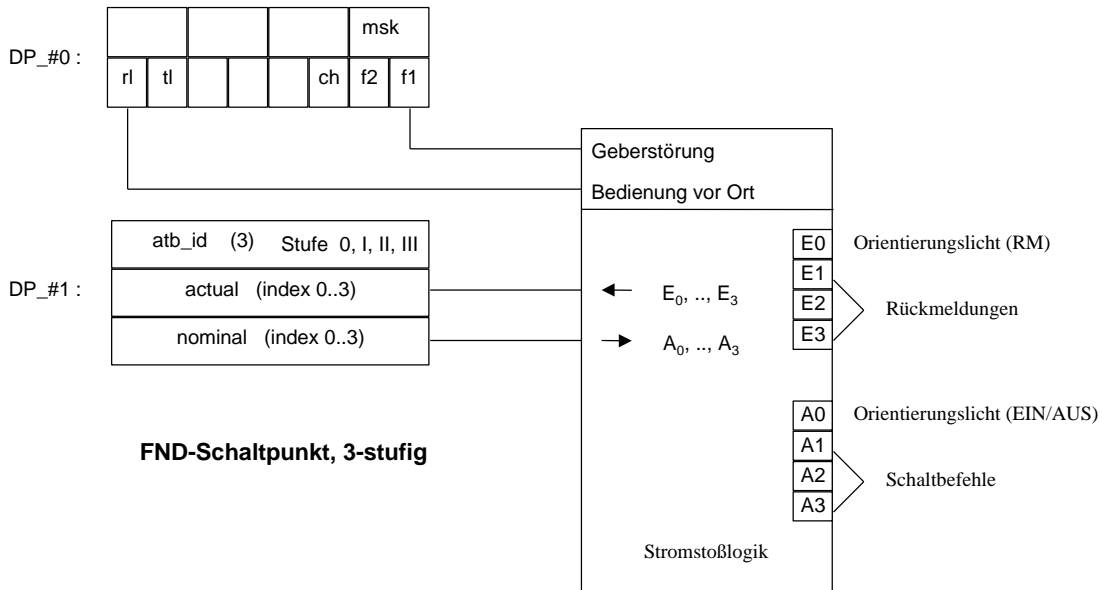
Für Sonderveranstaltungen besteht die Möglichkeit, die Beleuchtung am Schütz von Hand zu steuern, was spontan an die LZ zu melden ist.

Der jeweilige Zustand ist über einen gesonderten Rückmeldekontakt am Schütz im Datenpunkt darzustellen.

Führt ein Schaltbefehl innerhalb einer Wartezeit nicht zur Rückmeldung, so ist eine Geber-Störung zu melden.

Lösung: Je Bereich wird ein FND-Schaltpunkt mit dem Attribut STUFE 0...III (atb_id = 3) zugeordnet, dessen Vorgabewert innerhalb der Steuerung logisch mit den Ausgängen A₀...A₃ verknüpft wird. Der Ausgang A₀ steuert das Orientierungslicht, während die Ausgänge A₁...A₃ jeweils ein Beleuchtungsmittel ansteuern. Die Rückmeldekontakte am Schütz für das Orientierungslicht und die einzelnen Stromkreise werden an den Eingängen E₀...E₃ aufgeschaltet und auf den „actual“ abgebildet.

Die zeitabhängige Steuerung übernimmt der GA-Knoten.



- Geberstörung :** Innerhalb einer Wartezeit t nach Schaltbefehlausgabe bleibt **actual** ungleich **nominal**
- Bedienung vor Ort :** $E_{0..3}$ ändert Zustand unabhängig vom Vorgabewert (Spontanmeldung wegen Wertänderung, ch-Bit)
- Minimalbetrieb :** Bei Ausfall der übergeordneten Gebäudeleittechnik Umschalten auf Stufe 0, Bedienung vor Ort möglich

I.3 Einzelraum-Regelung

Aufgabe: Die Raumtemperatur soll abhängig von einem zeitlich vorgegeben Sollwert, einer manuellen Korrektoreinstellung und vom Fensterkontakt regelt werden. Die Zeitpunkte sollen bei Bedarf von der LZH aus verändert werden können. Die aktuelle Raumtemperatur, der effektive Sollwert und der Zustand des Fensterkontaktes sollen an der LZH ersichtlich sein. Der Ausfall der Regelung muss spontan an die LZH gemeldet werden.

Lösung: Der Regelung wird ein FND-Stellpunkt mit der Einheit Grad Celsius (dimension = 108) zugeordnet. Der nominelle Vorgabewert wird der Regelung per Stellbefehl übertragen. Beim Lesen und Melden des Datenpunktes wird jedoch der effektive, d.h. der um die Korrektoreinstellung veränderte Wert übertragen.

Die aktuelle Raumtemperatur wird auf den „actual“ abgebildet.

Ist das Fenster geöffnet, so ist die Regelung verriegelt, was durch ein gesetztes „tl“-Bit im „info_event“ angezeigt wird. In diesem Fall schließt die Regelung das Stellglied am Heizkörper, solange nicht eine festeingestellte, minimale Raumtemperatur unterschritten ist (Frostschutz).

Erkennt die Regelung eine Störung (z.B. defekter Fühler), so wird das „f1“-Bit im „info_event“ gesetzt und ggf. eine spontane Meßwertmeldung an die LZH gesendet.

Im GA-Knoten wird ein Zeit- und/oder Ereignis-Schaltprogramm aktiviert, dass zu den gewünschten Zeitpunkten die Solltemperatur für jeden einzelnen Raum mittels Stellbefehl vorgibt.

J Beispiele für GA-Funktionen

Nachfolgend soll anhand von einigen Beispielen erläutert werden, wie gewerkeübergreifende zeit- und/oder ereignis-abhängige Gebäude-Funktionen auf GA-Knoten und der LZH umgesetzt werden können.

J.1 Zentral-AUS

Aufgabe: Abhängig vom Blockschloss der Einbruchmeldeanlage werden sämtliche Gebäude-Funktionen (Licht, Heizung, Lüftung...) auf Nachtbetrieb geschaltet.

Lösung: Das Blockschloss wird als FND-Meldepunkt eingerichtet. Ein Ereignis-Schaltprogramm wartet auf die Scharfschaltung und setzt abhängig davon die entsprechenden Schaltbefehle ab.

J.2 Ferientaster

Aufgabe: Das Führen eines Schaltprogramm-Kalender erfordert insbesondere mit flexiblen Ferien-/Feiertagen einen personellen Aufwand und unterbindet eine flexible Nutzung, da eine starke Abhängigkeit vom Bedienpersonal besteht. In vielen Objekten der Landeshauptstadt München steht Bedienpersonal nur bedingt zur Verfügung, so dass der Nutzer selbst mittels eines Taster am Tableau steuernd eingreifen muss.

Lösung: Im GA-Knoten wird der Taster am Tableau als Rückmeldung auf einen FND-Schaltprogramm eingerichtet. Auf diese Weise kann sowohl vom Taster als auch von der Bedienstation oder einem Schaltprogramm aus geschaltet werden.

Aufgrund der Nutzung durch andere Gruppen, wirkt der Schaltbefehl erst mit Beginn des nächsten Tages, wenn er nach 8:00 Uhr erfolgt, sonst sofort.

Auf diese Weise kann z.B. ein Amtsmeister am letzten Schultag mittags die Ferienfunktion aktivieren, obwohl bis in den Abend hinein Sportvereine die Turnhalle nutzen. Am ersten Schultag reicht das rechtzeitige Drücken des Tasters am Morgen.

Ist im GA-Knoten ein Schlüsselerersatz-System aktiviert, so kann zusätzlich das Auslösen der Funktion am Taster abhängig von einem Berechtigungs-Schlüssel erfolgen.

J.3 Sonderfunktionen

Aufgabe: Wie beim Ferientaster muss dem Nutzer für diverse Sonderfunktionen (z.B. kurzfristig anberaumte Lehrerkonferenz, Sportveranstaltungen am Wochenende,...) eine Möglichkeit zur flexiblen Nutzung des Gebäudes unabhängig vom Bedienpersonal gegeben werden.

Lösung: Im GA-Knoten wird der Taster am Tableau als Rückmeldung auf einen FND-Schaltprogramm eingerichtet. Auf diese Weise kann sowohl vom Taster als auch von der Bedienstation oder einem Schaltprogramm aus geschaltet werden.

Die oben beschriebene Zentral-AUS-Funktion stellt das energiesparende Rücksetzen sicher.

Ist im GA-Knoten ein Schlüsslersatz-System aktiviert, so kann zusätzlich das Auslösen der Funktion am Taster abhängig von einem Berechtigungs-Schlüssel erfolgen.

J.4 EM-Profile

Aufgabe: Für das Energie-Management werden unterschiedliche Profile für die Erfassung von Messdaten benötigt, die von der Bedienstation aus aktiviert werden und spätestens nach Ablauf einer vorgegebenen Laufzeit automatisch zurückgesetzt werden.

Lösung: Im GA-Knoten wird ein interner FND-Schaltpunkt eingerichtet, der von der Bedienstation oder einem Schaltprogramm aus geschaltet werden kann. Jeder Schaltvorgang löst in einem darauf wartenden Ereignis-Schaltprogramm das Setzen der neuen Erfassungs-Parameter aus und startet einen Zähler für die Laufzeitüberwachung. Wird innerhalb der Laufzeit erneut geschaltet, so beginnt der Vorgang von vorne. Der Ablauf der Laufzeit führt als Ereignis zum Rücksetzen der Erfassungs-Parameter.

K GLT-Richtlinien, Ausschreibungen

Nachfolgend werden die GLT-Richtlinien der Abteilungen Hochbau 6 (Elektro) und 7 (Heizung, Klima, Lüftung, Sanitär und Aufzüge) beigelegt.

K.1 Elektro

K.1.1 Aufbau der Datenpunkt-Adressen

K.1.2 Informationslisten

K.1.3 Ausschreibungstexte

K.2 Heizung, Klima, Lüftung

K.2.1 Aufbau der Datenpunkt-Adressen

K.2.1 Informationslisten

K.2.3 Ausschreibungstexte

Literaturverzeichnis

- [1] AMEV 1988 ff.: Planung und Ausführung von firmenneutralen Datenübertragungssystemen in öffentlichen Gebäuden und Liegenschaften (FND) – **Version 1.0**
- Teil 1: FND-Spezifikation (inkl. Anmerkungen und Ergänzungen vom 8.1.90)
 - Teil 2 und 3: FND-Ausschreibung
 - Teil 4: FND-Konformitätssicherung
 - Teil 5: FND-Konformitätstest (Testhandbuch)
- [2] LHM 1995: Ergänzungen zur FND-Spezifikation DIN V 32735
- [3] LHM 1995: DV-technisches Feinkonzept des Gebäude-Automatisierungs-Knoten
- [4] LHM 1995: Fachliches Feinkonzept für die Leitzentrale Haustechnik (LZH)
- [5] VdS 1991: VdS 2305: Verband der Sachversicherer e.V. (09/1991): Richtlinien für Gefahrenmeldeanlagen, TEMEX-Übertragungsprotokolle für Gefahrenmeldungen (TSS13a, TSS13b)
- [6] VdS 1994: VdS 2465 Version 0.4b Entwurf (23.03.1994): Verband der Sachversicherer e.V. (03/1994): Einheitliches Übertragungsprotokoll für Gefahrmeldungen für Übertragungsanlagen