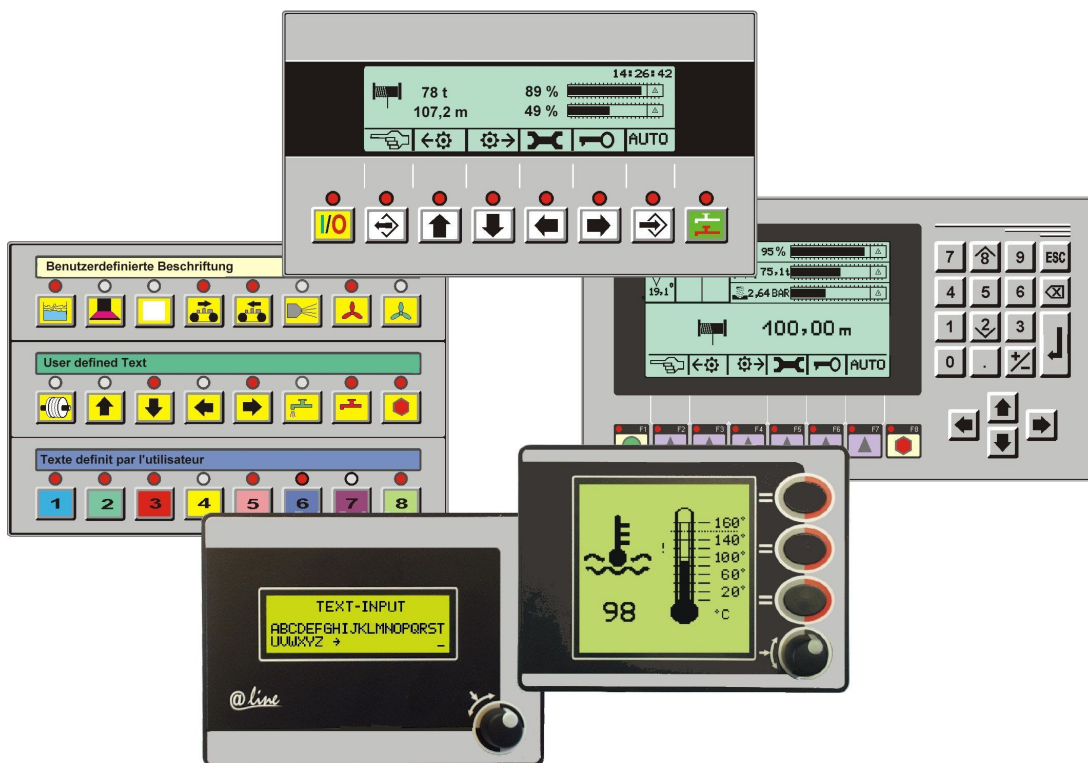




GRAF-SYTECO

Handbuch Kommunikation



Dokument: H094A2
Status: Freigegeben
Erstellt: März 2004

SYsteme **TE**chnischer **CO**mmunikation

GRAF-SYTECO GmbH & Co.KG * Kaiserstrasse 18 * D-78609 Tuningen
Tel: +49 7464 98660 * Fax: +49 7464 2550 * [http:// www.graf-syteco.de](http://www.graf-syteco.de) * eMail: info@graf-syteco.de

Inhalt Handbuch Bediengeräte

1	Kommunikation	1
1.1	Einführung	1
1.2	Telegrammformate	1
1.2.1	Aufbau der CAN-Telegramme	1
1.2.2	Aufbau der seriellen Telegramme	1
1.2.3	Aufbau der Nutzdaten	2
1.3	Telegrammarten nach Kategorie	2
1.3.1	Bild- und Meldeaufruf	2
1.3.2	Tasten und LEDs	2
1.3.3	Variablen	2
1.3.4	Status	3
1.3.5	Protokoll/Statistik	3
1.3.6	Speicher/Textübertragung	3
1.3.7	Cursorpositionierung	3
1.3.8	Sonstige	3
1.4	Beschreibung der Telegrammarten	4
1.4.1	REQUEST_VALUE (0x01)	4
1.4.2	SET_VALUE (0x02)	4
1.4.3	REPORT_VALUE (0x03)	5
1.4.4	MESSAGE_ON (0x04)	5
1.4.5	MESSAGE_OFF (0x05)	6
1.4.6	PAGE_ON (0x06)	6
1.4.7	PAGE_OFF (0x07)	6
1.4.8	PRIO_PAGE_ON (0x08)	7
1.4.9	REQUEST_STATUS (0x09)	7
1.4.10	REPORT_STATUS (0x0A)	8
1.4.11	ENABLE_REPORT_STATUS (0x0B)	9
1.4.12	DISABLE_REPORT_STATUS (0x0C)	9
1.4.13	REQUEST_MEMORY_WRITE (0x0D)	10
1.4.14	DISABLE_WRITE (0x0E)	10
1.4.15	WRITE_MEMORY (0x0F)	10
1.4.16	REQUEST_MEMORY_READ (0x10)	11
1.4.17	REPORT_READ_MEMORY (0x11)	11
1.4.18	RESET (0x12)	11
1.4.19	ACKNOWLEDGE (0x13)	12
1.4.20	REPORT_ERROR (0x14)	12
1.4.21	WRITE_PARAM (0x15)	13
1.4.22	SET_LED (0x16)	15
1.4.23	REPORT_KEY_DATA (0x17)	16
1.4.24	REQUEST_VERSION (0x18)	18
1.4.25	REPORT_VERSION (0x19)	18
1.4.26	REQUEST_CLOCK (0x1A)	19
1.4.27	REQUEST_RUNTIME (0x1B)	19
1.4.28	REQUEST_INTERN_VARIABLES (0x1C)	19
1.4.29	WRITE_CLOCK (0x1D)	20
1.4.30	REPORT_CLOCK (0x1E)	20
1.4.31	REPORT_RUNTIME (0x1F)	21
1.4.32	ASCII_TELEGRAM (0x20)	21
1.4.33	REQUEST_PROTOCOL (0x21)	22
1.4.34	REQUEST_STATISTIC (0x22)	23
1.4.35	REPORT_MENU_INDEX (0x25)	23
1.4.36	REPORT_OUTPUT_STATE (0x26)	24
1.4.37	REQUEST_CURSOR_POSITION (0x27)	24
1.4.38	WRITE_CURSOR_POSITION (0x28)	25
1.4.39	EXECUTE_MENU (0x29)	25
1.4.40	REPORT_CURSOR_POSITION (0x2A)	26
1.4.41	MENU_ON (0x2B)	27
1.4.42	CAN_INIT (0x2C)	27

Inhalt Handbuch Bediengeräte

1.4.43	WRITE_KEYCODE (0x2D)	28
1.4.44	DRAW (0x2E)	28
2	Der CAN-Bus	30
2.1	Verdrahtung	30
2.1.1	Busaufbau (Topologie)	30
2.1.2	Terminierung	30
2.1.3	Adressierung	30
2.2	Protokolle allgemein	31
2.2.1	Das SELECAN-Protokoll	31
2.2.2	Telegramminhalte	32
2.2.3	Bediengerät an SELECAN-SPS	32
2.2.4	Verwenden von GCM-Modulen	32
2.2.5	Das freie CAN-Protokoll	34
2.2.6	Das CANopen-Protokoll	34
2.2.7	Mischen von Protokollen	35
3	CAN-Open Treiber	36
3.1	Anforderungen	36
3.1.1	Betriebssystem (TOS)	36
3.1.2	Firmware (BIOS)	36
3.1.3	Projektiersoftware (Editor ITE)	36
3.1.4	Einstellungen im ITE	36
3.1.5	Feld ITS-CAN Konfiguration	36
3.1.6	Feld Knotennummer	36
3.1.7	Feld Überwachungszeit	36
3.1.8	Feld Zeitfenster	36
3.1.9	Anzeigefelder Download-ID's	37
3.1.10	Feld Intervallzeit für Abruf...	37
3.1.11	Feld Minimale Wartezeit	37
3.2	Die MASTER-Implementierung	37
3.2.1	Minimales Netzwerk-Management	37
3.2.2	Senden und Empfangen von SDO's	38
3.2.3	Beispiel: SDO senden	39
3.2.4	Beispiel: SDO lesen	39
3.2.5	Senden und Empfangen von PDO's	40
3.2.6	Projekt-Download	40
3.2.7	Objektverzeichnis	40
3.2.8	Zustandsübergänge	40
3.3	Die SLAVE-Implementierung	40
3.3.1	Zustandsdiagramm	40
3.3.2	Das Objektverzeichnis	42
4	SIMATIC S5-Treiber	50
4.1	Prinzipielle Funktion des Treibers	50
4.2	Grundsätzliche Überlegungen	50
4.3	Parametrierung des Treibers	51
4.3.1	Feld "Istwerte"	51
4.3.2	Feld "Sollwerte"	51
4.3.3	Felder "Untergrenzen", "Obergrenzen"	51
4.3.4	Feld "Schrittweite"	51
4.3.5	Feld "Status"	51
4.3.6	Feld "Bild/Meld"	51
4.3.7	Felder "Anzahl der Bilder, Meldungen, Prioritätsbilder, LEDs"	52
4.3.8	Feld "Tasten"	52
4.3.9	Feld "Anzahl der Tasten"	52
4.4	Der Status-Datenbaustein	53
4.4.1	Gerätestatus-Informationen	53
4.4.2	CAN-Gateway Sendepuffer	53
4.4.3	CAN-Gateway Empfangspuffer	55
4.4.4	CAN-Identifizier DW11 und DW17	55

Inhalt Handbuch Bediengeräte

4.4.5	Beispiele für das Gateway	55
5	MITSUBISHI FX	58
5.1	Prinzipielle Funktion des Treibers	58
5.2	Grundsätzliche Überlegungen	58
5.3	Parametrierung des Treibers	59
5.3.1	Feld Istwerte	59
5.3.2	Feld Sollwerte	59
5.3.3	Felder Untergrenzen, Obergrenzen	60
5.3.4	Feld Schrittwerte	60
5.3.5	Feld Status	60
5.3.6	Feld Bild/Meld	60
5.3.7	Felder Anzahl der Bilder, Meldungen, Prioritätsbilder, LEDs	60
5.3.8	Feld Tasten	61
5.3.9	Feld Anzahl der Tasten	61
5.4	Die Status-Datenregister	61
5.4.1	Gerätstatus-Informationen	62
5.4.2	CAN-Gateway Sendepuffer	62
5.4.3	CAN-Gateway Empfangspuffer	63
5.5	Beispiele für das Gateway	64
6	Request/Response Treiber	66
6.1	Schnittstellenbeschreibung:	66
6.1.1	Schnittstellenkommandos:	66
6.2	Prinzipielle Funktion des Treibers	67
6.2.1	Namensvereinbarungen	67
6.2.2	Grundsätzliche Überlegungen	67
6.3	Parametrierung des Treibers	68
6.3.1	Feld Istwerte	69
6.3.2	Feld Sollwerte	69
6.3.3	Felder Untergrenzen, Obergrenzen und Schrittwerte	69
6.3.4	Feld Status	69
6.3.5	Feld Bild/Meld	69
6.3.6	Felder ...x 8	69
6.3.7	Feld Tasten	70
6.3.8	Feld Anzahl der Tasten	70
6.3.9	Die Status-Datenregister	71
6.3.10	Gerätstatus-Informationen	71
6.3.11	CAN-Gateway Sendepuffer	71
6.3.12	CAN-Gateway Empfangspuffer	72
6.4	CAN-Identifizier D11 und D17	73
7	VT100-Treiber	74
7.1	Projektierung	74
7.2	Beschreibung der VT-100 Steuerfolgen	74
7.2.1	Control-Zeichen (Empfang)	74
7.2.2	ESC-Sequenzen (Empfang)	75
7.2.3	Tastencodes	77
7.3	Erweiterungen des VT100-Treibers	77
7.3.1	ESC-Sequenzen (Empfang)	78
7.3.2	Gesendete Tastencodes	78
8	Intercontrol DIGSYplus	79
8.1	Prinzipielle Funktion des Treibers	79
8.2	Grundsätzliche Überlegungen	79
8.3	Parametrierung des Treibers	80
8.3.1	P1: Basis-Adresse für Istwerte	80
8.3.2	P2: Basis-Adresse für Statusinfo	80
8.3.3	P3: Basis-Adresse für LED und Aufrufe	80
8.3.4	P4: Basis-Adresse für Tastenmerker	80
8.3.5	P5: Basisadresse für Sollwerte	80
8.3.6	P6: Basisadresse für Untergrenzen	80

Inhalt Handbuch Bediengeräte

8.3.7	P7: Basisadresse für Obergrenzen	80
8.3.8	P8: Basisadresse für Schrittweiten	81
8.3.9	Feld S1	81
8.3.10	Feld S2	81
8.3.11	Feld S3	81
8.3.12	Feld S4	81
8.3.13	Feld S5	81
8.4	Die Status-Datenregister	82
8.4.1	Gerätestatus-Informationen	82
8.4.2	CAN-Gateway Sendepuffer	83
8.4.3	CAN-Gateway Empfangspuffer	83
8.4.4	CAN-Identifizier MW+11 und MW+17	84
9	CAN-Module konfigurieren	85
9.1	Starten des Modul-Konfigurators	85
9.2	CAN-Modul anlegen	85
9.2.1	Feld "Name des Moduls"	85
9.2.2	Feld "Knoten-Nr.:"	85
9.2.3	Feld "Typ des Moduls"	85
9.3	Ein CAN-Modul konfigurieren	85
9.3.1	Modulserie GCM 200	86
9.3.2	Modulserie GCM 300	89
9.4	CAN-Modul entfernen	92
9.5	Modul kopieren	92

Handbuch Bediengeräte

1 Kommunikation

1.1 Einführung

Dieses Handbuch beschreibt die vielfältigen Möglichkeiten der Bediengeräte mit anderen Geräten zu kommunizieren, bzw. Daten auszutauschen. Die Anzahl und Art der verfügbaren Schnittstellen ist von Gerät zu Gerät je nach Bestückung unterschiedlich, und muß daher den Unterlagen zum jeweiligen Gerät entnommen werden.

Die Geräte sind von Haus aus in der Lage mit Steuerungen unterschiedlicher Hersteller direkt ohne Programmieraufwand zu kommunizieren. Für Steuerungen, die dieses Spektrum nicht abdeckt, besteht jederzeit die Möglichkeit eigene Kommunikationstreiber zu implementieren.

1.2 Telegrammformate

Dieses Kapitel beschreibt, wie das Bediengerät über die serielle Schnittstelle oder über den CAN-Bus angesteuert wird, und welche Informationen das Bediengerät aus dem Betriebssystem heraus direkt senden kann. Die übertragenen Nutzdaten sind prinzipiell identisch, lediglich der Telegrammrahmen und die Anzahl der übertragenen Nutzdatenbytes unterscheiden sich.

1.2.1 Aufbau der CAN-Telegramme

Den genauen Aufbau von CAN-Telegrammen und deren Telegrammrahmen entnehmen Sie bitte einschlägiger Literatur zum CAN-Bus. Hier soll nur ein schematischer Aufbau gezeigt werden, um die Funktionsweise der Datenkommunikation zu erläutern.

0	1	2	3	4	5	6	7	8	9	
Rahmen		Nutzdaten								Rahmen
Identifizier		Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	CRC

Der Identifizier im Telegrammrahmen dient nur der Identifikation des Gerätes am Bus. Die Anzahl der Datenbytes im Identifizier wird auf 8 Bytes festgelegt, unabhängig von der tatsächlich verwendeten Anzahl der Datenbytes. Die als nicht benutzt markierten Datenbytes sollten nach Möglichkeit mit 0x00 vorbelegt werden.

Bei CAN-Telegrammen werden immer alle 8 Nutzdatenbytes übertragen !

1.2.2 Aufbau der seriellen Telegramme

Grundsätzlich besteht jedes serielle Telegramm aus einem Startbyte (STX), einem Data-Length-Code (DLN, Größe des Datenbereichs), einer Geräteadresse (ID, Knotenadresse) einem Datenbereich (Data 0-N), sowie aus der Checksumme (CHK). Die Größe des Datenbereichs kann dabei zwischen 1 und 8 bytes variieren. Es ergibt sich also folgender Telegrammaufbau:

0	1	2	3	4		3+N	4+N	
Rahmen			Nutzdaten				Rahmen	
STX	DLN	ID	Data 0	Data 1	...	Data N	CHK	

- STX: Startbyte (Kennung 0Bhex)
- DLN: Datenlänge (von ID bis DataN, inclusive)
- ID: KnotenAdresse (eigene Adresse) = 0
- Data1-N: Nutzdaten
- CHK: Checksumme (ermittelt sich aus der Exklusiv-Oder-Verknüpfung der Bytes "DLN" bis "DataN", jeweils inclusive)

**Bei seriellen Telegrammen werden zum Teil nur die benötigten Datenbytes übertragen !
In den folgenden Tabellen zu den einzelnen Telegrammartentypen sind die über die serielle Schnittstelle übertragenen Datenbytes grau hinterlegt.**

Handbuch Bediengeräte

1.2.3 Aufbau der Nutzdaten

Das erste Byte in den Nutzdaten (D0) wird zur Codierung der Telegrammart (TA) benutzt. Auf diese Weise können in einem Telegramm maximal 7 Byte Nutzdaten übertragen werden

D0	D1	D2	D3	D4	D5	D6	D7
Telegramm-Art (TA)	Nutzdaten in Abhängigkeit von der Telegramm-Art						

Das Bediengerät arbeitet auf Frage-Antwort-Basis. Das heißt, die Steuerung stellt eine Anfrage an das Bediengerät und erhält die entsprechende Antwort.

Ergänzend gibt es einige Telegramme, in denen das Bediengerät der Steuerung seinen Status mitteilt. Diese Funktion ist über die Steuerung abschaltbar.

1.3 Telegrammarten nach Kategorie

In der folgenden Übersicht sind die Telegramme in verschiedene Kategorien eingeteilt und verweisen dann auf die jeweilige Beschreibung.

1.3.1 Bild- und Meldeaufruf

Telegramm	Funktion (grob)	TA=
MESSAGE_ON	Meldeaufruf: Meldung in den Meldungsstapel	0x04
MESSAGE_OFF	Meldeaufruf: Meldung aus dem Meldungsstapel	0x05
PAGE_ON	Bildaufruf: Bild in den Bilderstapel	0x06
PRIO_PAGE_ON	Bildaufruf: Bild mit Priorität	0x08
PAGE_OFF	Bildaufruf: Bild aus dem Bilderstapel	0x07
MENU_ON	Das Telegramm aktiviert ein Menübild	0x2B

1.3.2 Tasten und LEDs

Telegramm	Funktion (grob)	TA=
SET_LED	LED ein- und ausschalten	0x16
REPORT_KEY_DATA	Tasten-Status wird gemeldet	0x17
WRITE_KEY_CODE	Tastencode-Telegramm für Sollwerteingabe	0x2A
EXECUTE_MENU	Führt eine Menüfunktion aus, wirkt wie Tastendruck	0x29

1.3.3 Variablen

Telegramm	Funktion (grob)	TA=
REQUEST_VALUE	Wert einer externen Variable wird angefordert	0x01
SET_VALUE	Wert einer externen Variablen wird geliefert	0x02
REPORT_VALUE	Sollwert wird an die Steuerung gemeldet	0x03
REQUEST_CLOCK	integrierte Echtzeituhr lesen	0x1A
REQUEST_RUNTIME	Integrierten Betriebszeitähler lesen	0x1B
REQUEST_INTERN_VARIABLES	Wert einer internen Variablen anfordern	0x1C
WRITE_CLOCK	Uhrzeit der internen Echtzeituhr stellen	0x1D
REPORT_CLOCK	Bediengerät sendet Uhrzeit	0x1E
REPORT_RUNTIME	Bediengerät sendet Betriebszeitähler	0x1F

Handbuch Bediengeräte

1.3.4 Status

Telegramm	Funktion (grob)	TA=
REQUEST_STATUS	Status des Bediengeräts anfordern	0x09
REPORT_STATUS	Bediengerät meldet Status	0x0A
ENABLE_REPORT_STATUS	Bediengerät soll Status unaufgefordert melden	0x0B
DISABLE_REPORT_STATUS	Bediengerät soll Status nur auf Anforderung melden	0x0C
WRITE_PARAM	Geräte-Parameter ändern und überwachen	0x15
REQUEST_VERSION	Anfordern Firmware-Version	0x18
REPORT_VERSION	Firmware-Version wird geliefert	0x19
REPORT_OUTPUT_STATE	Der Status des Meldeausgangs wird gesendet	0x26
REPORT_MENU_INDEX	Der Index eines Menü-Eintrags wird gesendet	0x25

1.3.5 Protokoll/Statistik

Telegramm	Funktion (grob)	TA=
REQUEST_PROTOCOL	Anfordern des Protokollspeicherinhalts	0x21
REQUEST_STATISTIC	Anfordern des Statistikspeicherinhalts	0x22

1.3.6 Speicher/Textübertragung

Telegramm	Funktion (grob)	TA=
REQUEST_MEMORY_WRITE	Dem Bediengerät wird mitgeteilt, daß jetzt Daten folgen, die in den Textspeicher geschrieben werden sollen.	0x0D
DISABLE_WRITE	Beenden der Datenübertragung und RESET	0x0E
WRITE_MEMORY	Nutzdaten der Datenübertragung zum Bediengerät	0x0F
REQUEST_MEMORY_READ	Dem Bediengerät wird mitgeteilt, daß es die Daten aus dem Text-speicher senden soll.	0x10
REPORT_READ_MEMORY	Nutzdaten der Datenübertragung vom Bediengerät	0x11
REPORT_ERROR	Fehler bei Speicherfunktionen	0x14

1.3.7 Cursorpositionierung

Telegramm	Funktion (grob)	TA=
REQUEST_CURSOR_POSITION	Erfragt die aktuelle Cursorposition im Menu	0x27
WRITE_CURSOR_POSITION	Positioniert den Cursor auf einen Sollwert/Menüpunkt	0x28
REPORT_CURSOR_POSITION	Aktuelle Cursorposition wird gesendet	0x2A

1.3.8 Sonstige

Telegramm	Funktion (grob)	TA=
RESET	Rücksetzen des Geräts	0x12
CAN_INIT	Neuinitialisierung der CAN-Schnittstelle	0x2C
ACKNOWLEDGE	Bestätigung vom Bediengerät auf diverse Telegramme	0x13
ASCII_TELEGRAM	ASCII-Daten für Ausdruck Statistik, Protokoll	0x20
DRAW	Zeichnen von Linien und Rechtecken	0x2E

Handbuch Bediengeräte

1.4 Beschreibung der Telegrammarten

Ab diesem Abschnitt folgt die ausführliche Beschreibung der einzelnen Telegrammarten. In jeder Beschreibung finden Sie hier die Information über die Funktion, die Richtung des Telegramms (Bediengerät -> Master oder Master -> Bediengerät wobei der Master eine SPS, ein PC oder z.B. auch ein weiteres Bediengerät sein kann) und den Inhalt der Nutzdaten. Es wird das Telegrammformat beschrieben und Sie erhalten ein Beispiel zu jeder Telegrammart. Die Darstellung aller Zahlenwerte ist Hexa-Dezimal, was durch das jeder Zahl vorangestellte „0x“ ausgedrückt werden soll. (Beispiel: 0x10 = Dezimal 16)

1.4.1 REQUEST_VALUE (0x01)

Funktion:	Fordert den aktuellen Wert einer externen Variablen von der SPS an. Die SPS muß mit dem SET_VALUE Telegramm antworten.
Richtung:	Bediengerät -> Master
Nutzdaten:	Nummer (Handle) der angeforderten Variablen als Low-Byte und High-Byte Dualzahl ohne Vorzeichen (0..65500)
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x01	Handle Low-Byte	Handle High-Byte	nicht benutzt (0x00)				

Beispiel

0x01	0x03	0x01	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Variable mit Handle 259 angefordert ($1 \times 256 + 3$)

1.4.2 SET_VALUE (0x02)

Funktion:	Im Allgemeinen die Antwort auf das Telegramm REQUEST_VALUE. Der Wert der angeforderten Variablen wird übertragen. Das Telegramm wird auch dazu benutzt, den Wert von internen Variablen zu setzen (Variable laden). Die Unterscheidung interne/externe Variable projektieren Sie in der Projektieroberfläche. Anhand des Variablen-Handles erkennt das Bediengerät dann extern/intern
Richtung:	Master -> Bediengerät
Nutzdaten:	Handle: Nummer der Variablen, Dualzahl ohne Vorzeichen (0-65500) Byte 0: Niederwertigstes Byte des Wertes Byte 1: : Byte 2: : Byte 3: Höchstwertigstes Byte des Wertes
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x02	Handle Low-Byte	Handle High-Byte	nicht benutzt (0x00)	Byte 0	Byte 1	Byte 2	Byte 3

Beispiel

0x02	0x10	0x00	0x00	0x20	0x02	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Variable mit Handle 16 (=0x10) auf den Wert 544 gesetzt ($2 \times 256 + 32$).

Anmerkung: Die Bedeutung und die Anzahl der Felder „Byte 0“ bis „Byte 3“ hängen von der Datenart

Handbuch Bediengeräte

des über Handle angesprochenen Wertes ab. Die Zuordnung Handle / Datenart geschieht im Editor ITE.

1.4.3 REPORT_VALUE (0x03)

Funktion:	Übertragen eines Wertes an die SPS. Dieses Telegramm wird immer dann vom Bediengerät zur Steuerung gesendet, wenn ein Sollwert eingegeben wurde bzw. sich verändert hat oder die SPS den Wert über REQUEST_INTERN_VARIABLES angefordert hat
Richtung:	Bediengerät -> Master
Nutzdaten:	Handle: Nummer der Variablen, Dualzahl ohne Vorzeichen (0-65500) Byte 0: Niederwertigstes Byte des Sollwertes/Variablen Byte 1: : Byte 2: : Byte 3: Höchstwertigstes Byte des Sollwertes/Variablen
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x03	Handle Low-Byte	Handle High-Byte	nicht benutzt 0x00	Byte 0	Byte 1	Byte 2	Byte 3

Beispiel

0x03	0x25	0x00	0x00	0x80	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird der Wert 128 für die Variable mit Handle 37 (=0x25) angezeigt.

Anmerkung: Die Bedeutung und die Anzahl der Felder „Byte 0“ bis „Byte 3“ hängen von der Datenart des über Handle angesprochenen Wertes ab. Die Zuordnung Handle / Datenart geschieht im Editor ITE.

1.4.4 MESSAGE_ON (0x04)

Funktion:	Aktiviert eine Meldung
Richtung:	Master -> Bediengerät
Nutzdaten:	Meldungsnummer: 16 bit Nummer, 1-9999, ohne Vorzeichen
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x04	Meldungsnummer Low-Byte	Meldungsnummer High-Byte	nicht benutzt (0x00)				

Beispiel

0x04	0x12	0x01	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Meldung 274 (1 x 256 + 18) aktiviert.

Handbuch Bediengeräte

1.4.5 MESSAGE_OFF (0x05)

Funktion:	Deaktiviert eine Meldung
Richtung:	Master -> Bediengerät
Nutzdaten:	Meldungsnummer: 16 bit Nummer, 1-9999, ohne Vorzeichen
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x05	Meldungsnummer Low-Byte	Meldungsnummer High-Byte	nicht benutzt (0x00)				

Beispiel

0x05	0x03	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Meldung 3 deaktiviert.

1.4.6 PAGE_ON (0x06)

Funktion:	Aktiviert ein Bild
Richtung:	Master -> Bediengerät
Nutzdaten:	Bildnummer: 16 bit Nummer, 1-9999, ohne Vorzeichen
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x04	Bildnummer Low-Byte	Bildnummer High-Byte	nicht benutzt (0x00)				

Beispiel

0x06	0x04	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird Bild 4 aktiviert.

1.4.7 PAGE_OFF (0x07)

Funktion:	Deaktiviert ein Bild
Richtung:	Master -> Bediengerät
Nutzdaten:	Bildnummer: 16 bit Nummer, 1-9999, ohne Vorzeichen
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x07	Bildnummer Low-Byte	Bildnummer High-Byte	nicht benutzt (0x00)				

Beispiel

0x07	0x7	0x00	0x00	0x00	0x00	0x00	0x00
------	-----	------	------	------	------	------	------

Im Beispiel wird Bild 7 deaktiviert.

Handbuch Bediengeräte

1.4.8 PRIO_PAGE_ON (0x08)

Funktion:	Mit diesem Telegramm unterbricht die SPS alle laufenden Aktivitäten am Bediengerät und zeigt ein Bild als Dringlichkeits-Bild an
Richtung:	Master -> Bediengerät
Nutzdaten:	Bildnummer: 16 bit Nummer, 1-9999, ohne Vorzeichen
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x08	Bildnummer Low-Byte	Bildnummer High-Byte	nicht benutzt (0x00)				

Beispiel

0x08	0x64	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird das Bild Nr. 100 mit Priorität aktiviert.

1.4.9 REQUEST_STATUS (0x09)

Funktion:	Mit diesem Telegramm kann die SPS den aktuellen Zustand des Bediengerätes erfragen. Es ermöglicht die Abfrage von Tasten-, LED- und Geräte-Status. Das Bediengerät antwortet entsprechend mit den Telegrammen REPORT_KEY_DATA (Tasten- bzw. LED -Status), REPORT_STATUS (Geräte-Status) oder REPORT_OUTPUT_STATE (Meldeausgang)
Richtung:	Master -> Bediengerät
Nutzdaten:	Typ der abzufragenden Status-Daten 00 hex --> Geräte-Status (Antwort: REPORT_STATUS) 01 hex --> Tastatur-Status (Taste1 - 32) (Antwort: REPORT_KEY_DATA) 02 hex --> Tastatur-Status (Taste33 - 64) (Antwort: REPORT_KEY_DATA) 03 hex --> LED-Status (LED 1 - 32) (Antwort: REPORT_KEY_DATA) 04 hex --> LED-Status (LED 33 - 64) (Antwort: REPORT_KEY_DATA) 05 hex --> Status Meldeausgang (Antwort: REPORT_OUTPUT_STATE)
DLN	3

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x09	Mode	nicht benutzt (0x00)					

Beispiel

0x09	0x01	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird der Status der Tasten 1-32 abgefragt. Als Antwort würde das Telegramm REPORT_KEY_DATA vom Bediengerät gesendet.

Handbuch Bediengeräte

1.4.10 REPORT_STATUS (0x0A)

Funktion:	Dieses Telegramm wird vom Bediengerät an die SPS abgesandt und kann von der SPS zur Erkennung des Bediengerätestatus verwendet werden. Das Telegramm wird vom Bediengerät immer dann abgesandt, wenn sich eine Statusinformation im Bediengerät ändert oder das Telegramm REQUEST_STATUS empfangen wird. Wurde DISABLE_REPORT_STATUS empfangen, so wird das Telegramm nur noch auf Anforderung REQUEST_STATUS gesendet.
Richtung:	Bediengerät -> Master
Nutzdaten:	<p>Nummer des gerade angezeigten Bildes, Nummer der gerade angezeigten Meldung, Tasten- und Gerätestatus</p> <p>Byte 1,2: Bildnummer: 16 bit, 1-9999, ohne Vorzeichen</p> <p>Byte 3,4: Meldungsnummer: 16 bit, 1-9999, ohne Vorzeichen</p> <p>Byte 5: Tastenstatus</p> <ul style="list-style-type: none"> Bit0 = nicht benutzt Bit1 = ESC Bit2 = Pfeil links Bit3 = Pfeil rechts Bit4 = Pfeil runter Bit5 = Pfeil hoch Bit6 = ENTER Bit7 = nicht benutzt <p>Byte 6: Bediengerätestatus (Werte für Bits 0 bis 4, Bits 5 bis 7 sind reserviert)</p> <ul style="list-style-type: none"> 0: nicht definiert (Fehler !) 1: Ruhezustand 2: Bilder blättern 3: Stapel-Bilder blättern 4: Meldungen blättern 5: Menüeintrag auswählen 6: Sollwert auswählen 7: Sollwert mit Zifferneingabe editieren 8: Sollwert mit Schrittwertbearbeitung editieren 9: Meldung quittieren 10: Protokoll blättern 11: Statistik blättern 12-31: nicht definiert (Fehler !)
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0A	Bildnummer Low-Byte	Bildnummer High-Byte	Meldungs- nummer Low-Byte	Meldungs- nummer High-Byte	Tastenstatus	Gerätestatus	nicht benutzt

Beispiel

0x0A	0x05	0x01	0x04	0x00	0x04	0x05	0x00
------	------	------	------	------	------	------	------

Im Beispiel meldet das Bediengerät:

- *Bild 261 wird angezeigt,*
- *Meldung 4 wird angezeigt*
- *Taste 3 ist gedrückt*
- *Das Gerät ist im Status "Menüeintrag auswählen", das heißt der Cursor steht auf einem Menüpunkt.*

Handbuch Bediengeräte

1.4.11 ENABLE_REPORT_STATUS (0x0B)

Funktion:	Das automatische Senden bei Änderung des Status wird eingeschaltet. Das Bediengerät sendet jetzt bei jeder Statusänderung unaufgefordert ein REPORT_STATUS Telegramm
Richtung:	Master -> Bediengerät
Nutzdaten:	keine
DLN	2

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0B	nicht benutzt (0x00)						

Beispiel

0x0B	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Funktion REPORT_STATUS aktiviert. Bei Änderung des Geräte-Status sendet das Bediengerät ab sofort ein REPORT_STATUS - Telegramm.

Anmerkung: Änderungen des Tastenstatus (Taste gedrückt/losgelassen) werden nicht über das REPORT_STATUS Telegramm gemeldet, sondern über das Telegramm REPORT_KEY_DATA. Das Abschalten der Tastaturtelegramme erfolgt mit der Projektiersoftware ITE.

1.4.12 DISABLE_REPORT_STATUS (0x0C)

Funktion:	Das automatische Senden bei Änderung des Status wird ausgeschaltet. Wenn sich der Status des Bediengeräts ändert, wird dies der Steuerung nicht mehr mitgeteilt. Sie muß über REQUEST_STATUS den Status des Bediengeräts selbst abfragen.
Richtung:	Master -> Bediengerät
Nutzdaten:	keine
DLN	2

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0C	nicht benutzt (0x00)						

Beispiel

0x0C	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Funktion DISABLE_REPORT_STATUS ausgeführt.

Anmerkung: Änderungen des Tastenstatus (Taste gedrückt/losgelassen) werden nicht über das REPORT_STATUS Telegramm gemeldet, sondern über das Telegramm REPORT_KEY_DATA. Diese Funktion kann nur über Einstellungen im Projekt deaktiviert werden.

Handbuch Bediengeräte

1.4.13 REQUEST_MEMORY_WRITE (0x0D)

Funktion:	Mit diesem Telegramm wird die Schreibsperre des eingebauten Text/Programmspeichers aufgehoben sowie die Adresse und Anzahl der zu schreibenden Bytes bekanntgemacht
Richtung:	Master -> Bediengerät
Nutzdaten:	Anfangsadresse des zu schreibenden Bereichs und Anzahl der zu schreibenden Bytes Byte 1: Banknummer (Memory-Page) des zu schreibenden Bereichs Byte 2,3: Adressbits A0-A7 und A8-A15 für den Bereichsanfang Byte 4,5: Anzahl Datenbytes Low-Byte und High-Byte Byte 6,7: Heben den Schreibschutz des Speichers auf und müssen wie unten beschrieben übertragen werden

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0D	Bank-Nummer	Startadresse Low-Byte	Startadresse High-Byte	Anzahl Bytes Low-Byte	Anzahl Bytes High-Byte	0xE5	0xAA

WICHTIGER HINWEIS !

Die Anwendung dieses Telegramms erfordert genaueste Kenntnisse über die interne Speicheraufteilung des Bediengeräts. Wird dieses Telegramm falsch angewendet, so kann dies zu Funktionsstörungen des Geräts führen. Das Anwenderprojekt muß dann mit der Projektiersoftware neu in das Gerät geladen werden. Das Bediengerät wird durch dieses Telegramm in den "Terminal ready for Upload"-Zustand geschaltet. Anwender, die die SENDDATA.DAT Datenerzeugung nutzen, finden dieses Telegramm im Download-Datenstrom.

1.4.14 DISABLE_WRITE (0x0E)

Funktion:	Mit diesem Telegramm wird die Schreibsperre des eingebauten Text/Programmspeichers gesetzt und das Gerät rückgesetzt (RESET)
Richtung:	Master -> Bediengerät
Nutzdaten:	Die Datenbytes 1-7 müssen mit den unten angegebenen Werten gefüllt werden. Dies dient der Sicherung des Telegramms, denn nur wenn alle Datenbytes so empfangen wurden, wird das Telegramm akzeptiert.

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0E	0xD8	0x47	0x33	0xE5	0x4C	0xAA	0x29

ACHTUNG !

Beachten Sie auch den Hinweis zu REQUEST_MEMORY_WRITE. Das Telegramm beendet den durch REQUEST_WRITE_MEMORY bzw. REQUEST_MEMORY_READ ausgelösten Zustand "Terminal ready for Upload".

1.4.15 WRITE_MEMORY (0x0F)

Funktion:	Mit diesem Telegramm werden die zu schreibenden Daten übertragen. Pro Telegramm werden 5 Byte Nutzdaten übertragen
Richtung:	Master -> Bediengerät
Nutzdaten:	Byte 1,2: Laufende Telegrammnummer Low- und High-Byte Bytes 3-7: 5 Byte zu schreibende Daten

Handbuch Bediengeräte

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0F	lfd. Nummer Low-Byte	lfd. Nummer High-Byte	Datenbyte 0	Datenbyte 1	Datenbyte 2	Datenbyte 3	Datenbyte 4

ACHTUNG !

Beachten Sie auch die Hinweise zu REQUEST_WRITE_MEMORY und DISABLE_WRITE

1.4.16 REQUEST_MEMORY_READ (0x10)

Funktion:	Mit diesem Telegramm kann der Inhalt des Speichers im Bediengerät ausgelesen werden
Richtung:	Master -> Bediengerät
Nutzdaten:	Anfangsadresse des zu lesenden Bereichs und Anzahl der zu lesenden Bytes Byte 1: Banknummer (Memory-Page) des zu lesenden Bereichs Byte 2,3: Adressbits A0-A7 und A8-A15 für den Bereichsanfang Byte 4,5: Anzahl Datenbytes Low-Byte und High-Byte Byte 6,7: Heben den Schreibschutz des Speichers auf und müssen wie unten beschrieben übertragen werden

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x0E	0xD8	0x47	0x33	0xE5	0x4C	0xAA	0x29

HINWEIS!

Die Verwendung dieses Telegramms macht nur Sinn, wenn die interne Speicherstruktur des Bediengeräts gut bekannt ist. Das Bediengerät geht in "Terminal ready for Upload" und sendet die angeforderten Daten über REPORT_READ_MEMORY. Beendet wird die Datenübertragung über das Telegramm „DISABLE_WRITE“.

1.4.17 REPORT_READ_MEMORY (0x11)

Funktion:	Mit diesem Telegramm beantwortet das Bediengerät eine Leseanfrage REQUEST_MEMORY_READ
Richtung:	Bediengerät -> Master
Nutzdaten:	Byte 1,2: Laufende Telegrammnummer Low- und High-Byte erstes Telegramm hat die Nr 1 (!) Bytes 3-7: 5 Byte gelesene Daten

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x11	lfd. Nummer Low-Byte	lfd. Nummer High-Byte	Datenbyte 0	Datenbyte 1	Datenbyte 2	Datenbyte 3	Datenbyte 4

1.4.18 RESET (0x12)

Funktion:	Mit diesem Telegramm wird das Bediengerät neu "gestartet". Die Firmware verhält sich dabei so, als ob die Spannung aus- und wieder eingeschaltet wird
Richtung:	Master -> Bediengerät
Nutzdaten:	keine
DLN	2

Handbuch Bediengeräte

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x12	nicht benutzt (0x00)						

1.4.19 ACKNOWLEDGE (0x13)

Funktion:	Mit diesem Telegramm bestätigt das Bediengerät die Telegramme RESET, REQUEST_MEMORY_READ, REQUEST_MEMORY_WRITE, WRITE_MEMORY und DISABLE_WRITE. Das ACKNOWLEDGE - Telegramm wird zudem zur Signalisierung der Betriebsbereitschaft nach Spannungwiederkehr oder Watchdog-Fehler benutzt.
Richtung:	Bediengerät -> Master
Nutzdaten:	keine
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x13	nicht benutzt (0x00)						

1.4.20 REPORT_ERROR (0x14)

Funktion:	Mit diesem Telegramm teilt das Bediengerät der SPS das Auftreten eines Fehlers mit.
Richtung:	Bediengerät -> Master
Nutzdaten:	Fehlercode: 0, 1: Busoff (vermutlich Kabelproblem) 2: Communication Error (Kommunikationsfehler ohne Datenverlust) 3: Overrun (Datentelegramme gingen verloren) 10: REQUEST_MEMORY_WRITE war fehlerhaft 11: DISABLE_WRITE war fehlerhaft 12: Zwischenpuffer-Überlauf (WRITE_MEMORY) 13: Schreibschutz aktiv (WRITE_MEMORY) 14: WRITE_MEMORY ohne REQUEST_MEMORY_WRITE 15: falsche Telegrammnummer (WRITE_MEMORY) 16: REQUEST_MEMORY_READ war fehlerhaft 50: Fehler beim Schreiben der Projektdaten. Vermutlich Flash-Speicher defekt.
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x14	Fehlercode	nicht benutzt (0x00)					

Beispiel

0x14	0x0A	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird gemeldet, daß das Telegramm REQUEST_MEMORY_WRITE empfangen wurde, aber einen ungültigen Inhalt hat.

Handbuch Bediengeräte

1.4.21 WRITE_PARAM (0x15)

Funktion:	Mit diesem Telegramm kann die SPS verschiedene Geräte-Parameter des Bediengerätes überwachen und ändern. Achtung: Die Einstellungen werden nicht im Flash gespeichert und müssen somit in der Initialisierungsroutine immer neu gesetzt werden !	
Richtung:	Master -> Bediengerät	
DLN	PA	Parametrierdaten abhängig vom Feld PA
4	0	globale Softkey-Maske setzen für Menütasten Data 0: Jede Taste ist einem Bit zugeordnet. Bit 0 für Taste 0, Bit1 für Taste 1, usw. Ist das Bit=0 dann ist die Softkey-Funktion ausgeschaltet Ist das Bit=1 dann hat die betreffende Taste Softkey-Funktion Es gilt in diesem Fall folgende Standardzurodnung: Bit0 (Taste 0) : keine Zurodnung Bit1 (Taste 1): ESC Bit2 (Taste 2): Pfeil links Bit3 (Taste 3): Pfeil rechts Bit4 (Taste 4): Pfeil ab Bit5 (Taste 5): Pfeil auf Bit6 (Taste 6): ENTER
4	1	Kontrast setzen. Data 0 = 0-23, wobei 23 = maximaler Kontrast (dezimal)
	2	Hintergrund-Beleuchtung setzen. Data 0 = 0-7, wobei 7 = maximale Helligkeit
	3	ITS-Geräte bis TOS IO034S00: Status der Statuszeile setzen Data 0 = 0: Status-Zeile enabled (eingblendet) Data 0 = 1: Status-Zeile disabled (ausgeblendert) Data 0 = 2: Bildwerte (lokal im Bild definierte Parameter sind gültig) AT-Geräte: EL-Folienbeleuchtung ein/ausschalten D2 = 0: Beleuchtung aus D2 = 1: Beleuchtung ein
	4	nur bis TOS IO034S00)Zeile der Status-Zeile setzen. Data 0 = 0-7
	5	Rollierzeit der aktiven Meldungen einstellen Data 0=0: keine Rollierfunktion. Data 0=1-32: Rollierzeit in Sek.
	6	Rollierzeit der aktiven Bilder einstellen Data 0=0: keine Rollierfunktion. Data 0=1-32: Rollierzeit in Sek.
	7	Tastenbelegung der Menü-Tasten parametrieren Data 0=Tastennummer der ESC-Taste Data 1=Tastennummer der "Pfeil links"- Taste Data 2=Tastennummer der "Pfeil rechts"- Taste Data 3=Tastennummer der "Pfeil ab"- Taste Data 4=Tastennummer der "Pfeil auf"-Taste Data 5=Tastennummer der Enter Taste
	8	Meldeausgang setzen/rücksetzen Data 0=0: Meldeausgang rücksetzen Data 0=1: Meldeausgang aktivieren

Handbuch Bediengeräte

9	<p>Funktion "automatisches Sollwertsenden" im Zustand "Sollwertbearbeitung mit Stepfunktion" ein/ausschalten. Ist die Funktion aktiviert, wird bei jedem Tastendruck der "Up/Down"-Tasten der aktuelle Sollwert zur Steuerung gesendet. Data 0=0: Funktion ausschalten Data 0=1: Funktion einschalten, bei jedem Tastendruck wird mit dem Answer Data -Telegramm der aktuelle Sollwert gesendet.</p>
10	<p>Tastaturlayout für optionale PS/2 Tastatur setzen. Data 0=0 Amerikanische Belegung (US-Amerikanisch) Data 0=1 Deutsche Belegung Data 0=2 Französische Belegung Data 0=3 Englische Belegung Data 0=4 Italienische Belegung Data 0=5 Spanische Belegung Data 0=6 Schwedisch/finnische Belegung Data 0=7 Belgische Belegung Data 0=8 Dänische Belegung Data 0=9 Norwegische Belegung Data 0=10 Schweizer/Deutsche und Französische Belegung Data 0=11 Portugiesische Belegung Tastaturlayouts siehe Handbuch „Bedienen und Beobachten“</p>
11	<p>Umschalten Zeichensatz. Data 0=Banknummer in welcher sich der Zeichensatz befindet ! Diese Funktion nicht verwenden, nur aus Kompatibilitätsgründen vorhanden !</p>
12	<p>Blinktakt setzen in 10 ms-Schritten. Data 0=Low-Byte Data 1=High-Byte</p>
13	<p>Zeitzone umschalten Data 0=0:Winterzeit Data 0=1:Sommerzeit Data 0=2:keine Zeitzone verwenden</p>
14	<p>Update der Variablen in der Anzeige: Data 0=0: nur aufbereiten, wenn sich ihr Wert ändert Data 0=1: mit jedem "SetValue"-Telegramm neu aufbereiten</p>
15	<p>Buzzer Ein- / Ausschalten Data 0=Mode Data 1=Buzzer ON-Time in 10ms Schritten Data 2=Buzzer OFF-Timer in 10ms Schritten Data 3=Count</p> <p>Mode = 0 Buzzer ist ausgeschaltet, On- und Off-Time und Count werden ignoriert. Mode = 1 Buzzer ist eingeschaltet, leise Mode = 3 Buzzer ist eingeschaltet, laut Ist Mode <> 0 und On- und Off-Time beide = 0 bleibt der Buzzer eingeschaltet Ist Mode <> 0 und nur On-Time <> 0 dann wird der Buzzer für genau diese Zeit eingeschaltet (1 Puls - „Single shot“). Ist Mode <> 0 und On- und Off-Time beide <> 0 dann wird der Buzzer periodisch angesteuert. Nur diesem Fall wird auch Count ausgewertet und die in Count angegebene Anzahl Perioden ausgegeben.</p>

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x15	PA	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5

Handbuch Bediengeräte

Beispiele

0x15	0x0D	0x01	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird das Gerät auf Sommerzeit geschaltet.

1.4.22 SET_LED (0x16)

Funktion:	Mit diesem Telegramm kann die SPS die Tastatur-LEDs des Bediengerätes an-steuern.
Richtung:	Master -> Bediengerät
DLN	CTRL = Steuer-Parameter: 3 0 --> alle LEDs reset 5 1 --> LED-Maske mit dem übertragenen Wert UND-Verknüpfen 5 2 --> LED-Maske mit dem übertragenen Wert ODER-Verknüpfen 5 3 --> LED-Maske mit dem übertragenen Wert setzen 4 4 --> einzelne LED setzen 4 5 --> einzelne LED reset Nummer: LED-Masken-Nummer (0...7) für CTRL=1...3 (bitcodiert): 0=LEDs 1-8 1=LEDs 9..15 ... 7= LEDs 57-64 LED-Nummer (1-64), für CTRL= 4+5 1=LED 1 ... 64=LED 64
DLN	3..5 je nach Funktion CTRL

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x16	CTRL	Nummer	Wert	nicht benutzt (0x00)			

Beispiel

0x16	0x03	0x00	0x33	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel werden die LEDs 1, 2, 5 und 6 ein-, die LEDs 3, 4, 7 und 8 ausgeschaltet.

Handbuch Bediengeräte

1.4.23 REPORT_KEY_DATA (0x17)

Funktion:	Mit diesem Telegramm sendet das Bediengerät den Tasten- bzw. LED-Status an die SPS. Dies geschieht entweder auf Anfrage der SPS mit einem REQUEST_STATUS Telegramm, oder automatisch bei Änderung des Tasten-Status, sofern diese Funktion mit dem ITE eingeschaltet wurde.
Richtung:	Bediengerät -> Master
Nutzdaten:	<p>CTRL = Steuer-Parameter: 0 --> Senden von Status und Nummer einer einzelnen Taste 1 --> Senden des Tasten-Status der Tasten 1-32 (TA0-TA3) 2 --> Senden des Tasten-Status der Tasten 33-64 (TA4-TA7) 3 --> Senden des LED-Status der LED 1 - 32 (TA0-TA3) 4 --> Senden des LED-Status der LED 33 - 64 (TA4-TA7)</p> <p>Nummer:gültig für CTRL=0 enthält die Tasten-Nummer (bits 0-6), bit 7 = Status (gedrückt/losgelassen) Bsp: Nummer=0A hex -->Taste 10 wurde gedrückt Nummer=8A hex --> Taste 10 wurde losgelassen</p> <p>TA0 bis TA7:gültig für CTRL=1 bis 4 Statusbytes der Tasten bzw. Tasten-LEDs (bitorientiert) TA0 --> Tasten/LEDs 1-8 TA1 --> Tasten/LEDs 9-16 ... TA7 --> Tasten/LEDs 57-64</p>
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x17	CTRL	Nummer	nicht benutzt (0x00)	TA0 / TA4	TA1 / TA5	TA2 / TA6	TA3 / TA7

Beispiel

0x17	0x00	0x03	0x00	0x04	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird gemeldet, daß Taste 3 gedrückt wurde.

Nachfolgend eine Auflistung der Tastennummern:

ITS/AT 61/67		
1-48	TA0 bis TA5	entsprechen den Funktionstasten F1 - F48
65(0x41)	TA6 bit 0	nicht benutzt
66(0x42)	TA6 bit 1	ESC-Taste
67(0x43)	TA6 bit 2	"Pfeil links" / bzw. "Ziffer 4" (Doppelbelegung)
68(0x44)	TA6 bit 3	"Pfeil rechts" / bzw. "Ziffer 6" (Doppelbelegung)
69(0x45)	TA6 bit 4	"Pfeil ab" / bzw. "Ziffer 2" (Doppelbelegung)
70(0x46)	TA6 bit 5	"Pfeil auf" / bzw. "Ziffer 8" (Doppelbelegung)
71(0x47)	TA6 bit 6	ENTER-Taste
72(0x48)	TA6 bit 7	nicht benutzt
73(0x49)	TA7 bit 0	"Ziffer 0" -Taste

Handbuch Bediengeräte

74(0x4A)	TA7 bit 1	"Ziffer 1" -Taste
75(0x4B)	TA7 bit 2	"Ziffer 3" -Taste
76(0x4C)	TA7 bit 3	"Ziffer 5" -Taste
77(0x4D)	TA7 bit 4	"Ziffer 7" -Taste
78(0x4E)	TA7 bit 5	"Ziffer 9" -Taste
79(0x4F)	TA7 bit 6	"Dezimalpunkt" -Taste
80(0x50)	TA7 bit 7	"+/-" -Taste

ITS/AT 62/63/68/72/78

1-64	TA0 bis TA7	entsprechen den Funktions-Tasten F1 - F64
2-7	TA0 bit 1-6	besitzen eine Doppelfunktion entsprechend ihrer Definition als Menü- bzw. Funktionstasten

Taste F2-F7 als Menü-Tasten definiert:

2(0x02)	TA0 bit 1	ESC-Taste
3(0x03)	TA0 bit 2	"Pfeil links"
4(0x04)	TA0 bit 3	"Pfeil rechts"
5(0x05)	TA0 bit 4	"Pfeil Ab"
6(0x06)	TA0 bit 5	"Pfeil Auf"
7(0x07)	TA0 bit 6	ENTER-Taste

ITS/AT 71/77

1-40	TA0 bis TA4	entsprechen den Funktionstasten F1 - F40
41(0x29)	TA5 bit 0	nicht benutzt
42(0x2A)	TA5 bit 1	ESC-Taste
43(0x2B)	TA5 bit 2	"Pfeil links"
44(0x2C)	TA5 bit 3	"Pfeil rechts"
45(0x2D)	TA5 bit 4	"Pfeil Ab"
46(0x2E)	TA5 bit 5	"Pfeil Auf"
47(0x2F)	TA5 bit 6	ENTER-Taste
48(0x30)	TA5 bit 7	BACKSPACE-Taste
49(0x31)	TA6 bit 0	"Ziffer 0" -Taste
50(0x32)	TA6 bit 1	"Ziffer 7" -Taste
51(0x33)	TA6 bit 2	"Ziffer 8" -Taste
52(0x34)	TA6 bit 3	"Ziffer 9" -Taste
53(0x35)	TA6 bit 4	"Ziffer 4" -Taste
54(0x36)	TA6 bit 5	"Ziffer 5" -Taste
55(0x37)	TA6 bit 6	"Ziffer 6" -Taste
56(0x38)	TA6 bit 7	"Ziffer 1" -Taste
57(0x39)	TA7 bit 0	nicht benutzt
58(0x3A)	TA7 bit 1	"Ziffer 2" -Taste
59(0x3B)	TA7 bit 2	"Ziffer 3" -Taste
60(0x3C)	TA7 bit 3	"Dezimalpunkt" -Taste
61(0x3D)	TA7 bit 4	"+/-" -Taste
62(0x3E)	TA7 bit 5	nicht benutzt
63(0x3F)	TA7 bit 6	nicht benutzt
64(0x40)	TA7 bit 7	nicht benutzt

Handbuch Bediengeräte

1.4.24 REQUEST_VERSION (0x18)

Funktion:	Mit diesem Telegramm kann die SPS die Bediengeräte-Versionsnummern anfordern. Sowohl Firmware als auch Anwender-Versionsstand (Feld USERDATA) kann abgefragt werden. Die Antwort erfolgt mit dem Telegramm REPORT_VERSION.
Richtung:	Master -> Bediengerät
Nutzdaten:	Steuerparameter CTRL: CTRL=0: Bezeichnung der BIOS-Version anfordern CTRL=1: Bezeichnung der TOS-Version anfordern CTRL=2+n: Versions-Bezeichnung der Datenstruktur anfordern ab Stelle n (Feld USERDATA)
DLN	3

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x18	CTRL	nicht benutzt (0x00)					

Beispiel

0x18	0x02	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird das Feld "Userdata" ab Stelle 0 (2+0) abgerufen.

1.4.25 REPORT_VERSION (0x19)

Funktion:	Mit diesem Telegramm sendet das Bediengerät seine Versionsnummer an die SPS (ASCII-String, 7stellig). Ist die Antwort auf das Telegramm REQUEST_VERSION.
Richtung:	Bediengerät -> Master
Nutzdaten:	angeforderte Versionsnummer als ASCII-String ASCII 0 .. ASCII 6 ITB-xxxxyz: BIOS-Version ITO-xxxxyz: TOS-Version Dnnnnnn: Daten-Version (USERDATA) xxx = Programm-Stand y = Sonderbezeichnung zz = Sondernummer nnnnnn = beliebige ASCII-Zeichen
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x19	ASCII 0	ASCII 1	ASCII 2	ASCII 3	ASCII 4	ASCII 5	ASCII 6

Beispiel

0x19	'D'	'M'	'V'	'1'	'.'	'0'	'3'
------	-----	-----	-----	-----	-----	-----	-----

Im Beispiel meldet das Bediengerät, dass im Feld "Userdata" der String "MV1.00" steht.

Handbuch Bediengeräte

1.4.26 REQUEST_CLOCK (0x1A)

Funktion:	Anfordern von Uhrzeit und Datum der Echtzeituhr. Das Bediengerät beantwortet diese Anforderung mit dem REPORT_CLOCK - Telegramm. Anmerkung: ist das Gerät nicht mit Echtzeituhr ausgestattet, dann wird zwar ein REPORT_CLOCK Telegramm generiert, dies enthält aber zufällige Daten.
Richtung:	Master -> Bediengerät
Nutzdaten:	keine
DLN	2

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x1A	nicht benutzt (0x00)						

1.4.27 REQUEST_RUNTIME (0x1B)

Funktion:	Anfordern der internen Betriebsstundenzeit (Gesamteinschaltzeit des Geräts). Das Bediengerät antwortet mit dem REPORT_RUNTIME Telegramm.
Richtung:	Master -> Bediengerät
Nutzdaten:	keine
DLN	2

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x1B	nicht benutzt (0x00)						

1.4.28 REQUEST_INTERN_VARIABLES (0x1C)

Funktion:	Anfordern eines internen Variablenwertes ("interne Variablen" sind im Bediengerät gespeicherte Variablen). Die Anforderung wird mit dem REPORT_VALUE Telegramm beantwortet.
Richtung:	Master -> Bediengerät
Nutzdaten:	Handle: Variablen-Nummer (0-65500)
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x1C	Handle Low-Byte	Handle High-Byte	nicht benutzt (0x00)				

Beispiel

0x1C	0x04	0x01	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird der Wert der Variablen mit Handle 260 angefordert (4 + 256 * 1).

Handbuch Bediengeräte

1.4.29 WRITE_CLOCK (0x1D)

Funktion:	Setzen von Uhrzeit und Datum der Echtzeituhr.
Richtung:	Master -> Bediengerät
Nutzdaten:	Zu setzendes Datum/Uhrzeit im BCD-Format (!) TT: Tag MM: Monat JJ: Jahr HH: Stunde mm: Minute SS: Sekunde DW: Wochentag 0..6
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x1D	HH	mm	SS	TT	MM	JJ	DW

Beispiel

0x1D	0x14	0x24	0x32	0x12	0x05	0x01	0x02
------	------	------	------	------	------	------	------

Im Beispiel wird Datum: 12.05.2001, Uhrzeit 14:24:32, Wochentag 2 eingestellt.

1.4.30 REPORT_CLOCK (0x1E)

Funktion:	Antwort-Telegramm auf das "REQUEST_CLOCK" -Telegramm. Das Bediengerät sendet Datum und Uhrzeit der internen Echtzeituhr
Richtung:	Bediengerät -> Master
Nutzdaten:	Aktuelles Datum und aktuelle Uhrzeit im BCD-Format (!) TT: Tag MM: Monat JJ: Jahr HH: Stunde mm: Minute SS: Sekunde DW: Wochentag 0..6
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x1E	HH	mm	SS	TT	MM	JJ	DW

Beispiel

0x1E	0x15	0x02	0x16	0x17	0x11	0x0D	0x05
------	------	------	------	------	------	------	------

Im Beispiel meldet das Bediengerät Datum 17.11.2000, Uhrzeit 15:02:16, Wochentag 5.

Handbuch Bediengeräte

1.4.31 REPORT_RUNTIME (0x1F)

Funktion:	Antwort-Telegramm auf das REQUEST_RUNTIME -Telegramm. Das Bediengerät sendet seine Betriebsstundenzeit als Anzahl Sekunden.
Richtung:	Bediengerät -> Master
Nutzdaten:	Betriebszeit in Sekunden Time 0: Low-Byte der Betriebszeit Time 1: ... Time 4: High-Byte der Betriebszeit
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x1F	Time 0	Time 1	Time 2	Time 3	Time 4	nicht benutzt (0x00)	

Beispiel

0x1F	0x23	0xC0	0x12	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Laufzeit 1228835 Sekunden (entspricht 341 Stunden, 20 Minuten und 35 Sekunden) gemeldet.

1.4.32 ASCII_TELEGRAM (0x20)

Funktion:	Die Telegrammart ASCII_TELEGRAM ist ein Multiplex-Telegramm, bestehend aus mehreren Telegrammen zum Senden größerer ASCII-Blöcke. Es wird z.B. für den Protokoll- und Statistik-Ausdruck verwendet.
Richtung:	Bediengerät -> Master
Nutzdaten:	Daten im ASCII-Format (z.B. für Ausdrücke) LN: Anzahl im Telegramm gesendete ASCII-Zeichen (0..6) LN=0 kennzeichnet das Ende des Transfers von ASCII-Zeichen. In diesem Fall wird in ASCII 0 und ASCII 1 (16-Bit) die Anzahl der übertragenen Telegramme exclusive das Telegramm mit LN=0 gesendet
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x20	LN	ASCII 0 / Telegramme Low-Byte	ASCII 1 / Telegramme High-Byte	ASCII 2	ASCII 3	ASCII 4	ASCII 5

Beispiel

0x20	0x06	'M'	'e'	'l'	'd'	'u'	'n'
0x20	0x06	'g'	' '	'1'	' '	'g'	'e'
0x20	0x04	'h'	't'	0x0D	0x0A	0x00	0x00
0x20	0x00	0x03	0x00	0x00	0x00	0x00	0x00

Im Beispiel wird der String "Meldung 1 geht" mit Zeilenende und Wagenrücklauf gesendet.

Handbuch Bediengeräte

1.4.33 REQUEST_PROTOCOL (0x21)

Funktion:	Aufforderung zum Ausdrucken des Protokolls. Das Bediengerät antwortet ab dem Start des Druckvorganges mit ASCII-Telegrammen.
Richtung:	Master -> Bediengerät
Nutzdaten:	Angaben zum Format des Ausdrucks CTRL: Protokoll-Befehl 0=Protokollspeicher zurücksetzen 1=komplettes Protokoll senden 2=Nur Einträge eines Bildes (Bildnr. in NR) 3=Nur Einträge einer Meldung (Nr. in NR) 4=Nur Einträge mit bestimmtem Status 5=Nur Einträge einer Variablen (Nr. in NR) 6=Druck abbrechen 7=Variable protokollieren (Nummer steht in NR) OUTPUT: Ausgabemedium des Protokollausdrucks 0=Druck wird auf dem mit ITE festgelegten Drucker geleitet 1=Ausgabe auf die serielle Schnittstelle 2=Ausgabe auf die CAN-Schnittstelle FORMAT: Ausgabeformat des Protokollausdrucks 0=ASCII-Ausdruck (der komplette Text wird ausgedruckt) 1=Binär-Ausdruck (es wird nur Bild/Meld/Var-Nr. ausgedruckt) LFD: Nummer des Protokoll-Eintrags, ab welchem der Ausdruck gestartet wird. Wenn 0, wird ab dem zuletzt ausgedruckten Eintrag gestartet Wenn 0xFFFF werden alle Einträge ausgegeben NR: Bild-, Meldungs-Nummer bzw. Variablenhandle (nur gültig, wenn CTRL = 2,3,5,7)
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x21	CTRL	OUTPUT	FORMAT	LFD Low-Byte	LFD High-Byte	NR Low-Byte	NR High-Byte

Beispiel

0x21	0x02	0x02	0x00	0x40	0x00	0x07	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird der Protokollausdruck für das Bild 7 ab Eintrag Nr. 64 im ASCII-Format auf die CAN-Schnittstelle ausgelöst.

Handbuch Bediengeräte

1.4.34 REQUEST_STATISTIC (0x22)

Funktion:	Aufforderung zum Ausdrucken der Statistik. Das Bediengerät antwortet ab dem Start des Druckvorganges mit ASCII-Telegrammen.
Richtung:	Master -> Bediengerät
Nutzdaten:	Angaben zum Format des Ausdrucks CTRL: Statistik-Befehl 0=Statistikspeicher zurücksetzen 1=komplette Statistik senden 2=Nur Statistik einer Gruppe (Gruppennummer in NR) 3=Nur Statistik eines Bildes (Bildnummer in NR) 4=Nur Statistik einer Meldung (Meldungsnummer in NR) 5=Druck abbrechen OUTPUT: Ausgabemedium des Protokollausdrucks 0=Druck wird auf den mit ITE festgelegten Drucker geleitet 1=Ausgabe auf die serielle Schnittstelle 2=Ausgabe auf die CAN-Schnittstelle FORMAT: Ausgabeformat des Protokollausdrucks 0=ASCII-Ausdruck (der komplette Text wird ausgedruckt) NR: Bild-, Meldungs-Nummer bzw. Variablenhandle (nur gültig, wenn CTRL = 2,3,4)
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x22	CTRL	OUTPUT	FORMAT	NR Low-Byte	NR High-Byte	nicht benutzt (0x00)	

Beispiel

0x22	0x04	0x00	0x00	0x01	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird der Statistikausdruck für Meldung 1 im ASCII-Format auf den im Projekt eingestellten Drucker ausgelöst.

1.4.35 REPORT_MENU_INDEX (0x25)

Funktion:	Sendet den Menü-Index an die SPS, wenn ein entsprechender Menüpunkt mit "ENTER" ausgewählt wurde.						
Richtung:	Bediengerät -> Master						
Nutzdaten:	Index des Menüpunktes (16-Bit)						
DLN	9						

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x25	INDEX Low-Byte	INDEX High-Byte	nicht benutzt (0x00)				

Beispiel

0x25	0x05	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel sendet das Bediengerät den Menü-Index 5.

Handbuch Bediengeräte

1.4.36 REPORT_OUTPUT_STATE (0x26)

Funktion:	Sendet den Status des Meldeausgangs. Dieses Telegramm wird über REQUEST_STATUS mit Feld Mode=0x05 angefordert.
Richtung:	Bediengerät -> Master
Nutzdaten:	Status des Ausgangs OUTPUT=0: Ausgang nicht gesetzt OUTPUT=1: Ausgang gesetzt
DLN	3

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x26	OUTPUT	nicht benutzt (0x00)					

Beispiel

0x26	0x01	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel meldet das Bediengerät den Ausgang als gesetzt.

1.4.37 REQUEST_CURSOR_POSITION (0x27)

Funktion:	Erfragt die aktuelle Cursorposition im Menü
Richtung:	Master -> Bediengerät
Nutzdaten:	CTRL: Status der Sendeautomatik 0: Die aktuelle Cursor-Position wird mit dem REPORT_CURSOR_POSITION Telegramm einmal gesendet, dann nicht mehr. (Automatik aus) 1: Die aktuelle Cursor-Position wird mit dem REPORT_CURSOR_POSITION Telegramm gesendet. Zusätzlich wird eine Sendeautomatik eingeschaltet, die bei jeder Änderung der Cursorposition ein REPORT_CURSOR_POSITION - Telegramm ausgibt.
DLN	3

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x27	CTRL	nicht benutzt (0x00)					

Beispiel

0x27	0x01	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die Cursorposition erfragt und die Sendeautomatik eingeschaltet.

Handbuch Bediengeräte

1.4.38 WRITE_CURSOR_POSITION (0x28)

Funktion:	Positioniert den Cursor auf einen Menüpunkt /Sollwert, wenn ein dem Telegramm entsprechender Sollwert/Menüpunkt verfügbar ist. Anderenfalls erfolgt keine Änderung des Cursors.
Richtung:	Master -> Bediengerät
Nutzdaten:	<p>Position des Cursors nach Angabe des Modus</p> <p>MODE = 0,1: XY-Positionierung (Zeile/Spalte) DATA 0:X-Position (Spalte des Cursors - zeichenorientiert, nicht grafisch) DATA 1:Y-Position (Zeile des Cursors - zeichenorientiert, nicht grafisch)</p> <p>MODE = 2: Positionierung auf Variablen-Handle (Sollwert) DATA 0:Variablenhandle Low-Byte DATA 1:Variablenhandle High-Byte</p> <p>MODE = 3: Positionierung auf Menüpunkt DATA 0:Bildnummer/Menü-Index... Low-Byte (entsprechend DATA 2) DATA 1:Bildnummer/Menü-Index... High-Byte (entsprechend DATA 2) DATA 2:Funktion des Menüpunkts 0= ein Bild aufrufen 1= im Menü zurückgehen 2= globaler Abbruch 3= Sollwerteingabe beenden mit Speichern, anschl. Menübildaufruf 4= Sollwerteingabe beenden ohne Speichern, anschl. Menübildaufruf 5= Menü-Indexausgabe (Indexwert senden) 6= Menü-Indexausgabe mit globalem Abbruch 7= Menü-Index an KOP weitergeben</p> <p>Bemerkung: Der Cursor wird nur dann positioniert, wenn sich das Gerät in einem der Zustände "Ruhezustand", "Menüauswahl" oder "Sollwerteingabe" befindet und ein Menüpunkt und/oder ein Sollwert im Bild ist.</p>
DLN	6

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x28	MODE	DATA 0	DATA 1	DATA 2	nicht benutzt (0x00)		

Beispiel

0x28	0x03	0x0A	0x00	0x05	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird der Cursor auf einen Menüpunkt mit Funktion "Menü-Index 10 senden" positioniert.

1.4.39 EXECUTE_MENU (0x29)

Funktion:	Das Telegramm führt eine Menüfunktion aus, so als ob eine Menütaste gedrückt würde
Richtung:	Master -> Bediengerät
Nutzdaten:	CODE: Nummer der simulierten Taste
DLN	3

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x29	CODE	nicht benutzt (0x00)					

Handbuch Bediengeräte

Beispiel

0x29	0x47	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird am Bediengerät Serie ITS/AT 61/67 die Taste "ENTER" per CAN_Telegramm ausgeführt.

1.4.40 REPORT_CURSOR_POSITION (0x2A)

Funktion:	Die aktuelle Cursor-Position wird gesendet
Richtung:	Bediengerät -> Master
Nutzdaten:	<p>Position des Cursors</p> <p>MODE = 0: ungültig, kein Cursor einstellbar</p> <p>MODE = 1: Cursorposition wird als XY-Wert gemeldet (Zeile/Spalte) DATA 0: X-Position (Spalte des Cursors - zeichenorientiert, nicht grafisch) DATA 1: Y-Position (Zeile des Cursors - zeichenorientiert, nicht grafisch)</p> <p>MODE = 2: Cursorposition wird als Variablen-Handle gemeldet (steht auf Sollwert) DATA 0: Variablenhandle Low-Byte DATA 1: Variablenhandle High-Byte</p> <p>MODE = 3: Cursorposition wird als Menüpunkt gemeldet DATA 0: Bildnummer/Menü-Index... Low-Byte (entsprechend DATA 2) DATA 1: Bildnummer/Menü-Index... High-Byte (entsprechend DATA 2) DATA 2: Funktion des Menüpunkts 0= ein Bild aufrufen 1= im Menü zurückgehen 2= globaler Abbruch 3= Sollwerteingabe beenden mit Speichern, anschl. Menübildaufruf 4= Sollwerteingabe beenden ohne Speichern, anschl. Menübildaufruf 5= Menü-Indexausgabe (Indexwert senden) 6= Menü-Indexausgabe mit globalem Abbruch 7= Menü-Index an KOP weitergeben</p>
DLN	9

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x2A	MODE	DATA 0	DATA 1	DATA 2	nicht benutzt (0x00)		

Beispiel

0x2A	0x02	0x20	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel meldet das Bediengerät den Cursor auf einem Sollwert mit Handle 32.

Handbuch Bediengeräte

1.4.41 MENU_ON (0x2B)

Funktion:	Das Telegramm aktiviert ein Menübild, d.h. im aufgerufenen Bild wird ein Menüpunkt/Sollwert sofort angewählt. Somit besteht die Möglichkeit den Menübaum von extern über die CAN-Schnittstelle zu beeinflussen.
Richtung:	Master -> Bediengerät
Nutzdaten:	Nummer des Bildes 16bit-Wert, 1-9999, ohne Vorzeichen. Wird hier 0xFFFF (65535) angegeben, wird der Menübaum komplett gelöscht.
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7	
TA=0x2B	Bildnummer Low-Byte	Bildnummer High-Byte	nicht benutzt (0x00)					

Beispiel

0x2B	0x02	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird das Bild 2 aufgerufen und sofort der Menübaum aktiviert.

1.4.42 CAN_INIT (0x2C)

Funktion:	Über dieses Telegramm wird das Bediengerät veranlasst die CAN-Schnittstelle neu zu initialisieren. Dabei werden ggf. geänderte Parameter der Schnittstelle (Baudrate, Identifier) berücksichtigt.
Richtung:	Master -> Bediengerät
Nutzdaten:	EXTENDED = 0: Standardwert EXTENDED = 1...255: reserviert für zukünftige Funktionen
DLN	3

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x2C	EXTENDED	nicht benutzt (0x00)					

Beispiel

0x2C	0x00	0x00	0x00	0x00	0x00	0x00	0x00
------	------	------	------	------	------	------	------

Im Beispiel wird die CAN-Schnittstelle neu initialisiert.

Handbuch Bediengeräte

1.4.43 WRITE_KEYCODE (0x2D)

Funktion:	Tastencode-Telegramm (ITS7/AT-spezifisches Telegramm). Mit diesem Telegramm kann eine Sollwerteingabe von beliebigen ASCII-Zeichen vorgenommen werden (ASCII-Variablen !), d.h. ist ein Sollwert angewählt, wird das empfangene ASCII-Zeichen in den Sollwert eingefügt. Dient dazu, über Funktionstasten eine ASCII-Eingabe zu ermöglichen.
Richtung:	Master -> Bediengerät
Nutzdaten:	KEYCODE: ASCII-Code EXTENDED KEYCODE: reserviert für Erweiterungen (derzeit 0x00)
DLN	4

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x2D	KEYCODE	EXTENDED KEYCODE	nicht benutzt (0x00)				

Beispiel

0x2D	'A'	0x00	0x00	0x00	0x00	0x00	0x00
------	-----	------	------	------	------	------	------

Im Beispiel wird das Zeichen A als Eingabe für eine ASCII-Variable empfangen.

1.4.44 DRAW (0x2E)

1.4.44.1 Geräte der ITS-Serie

Funktion:	Funktion zum Zeichnen von grafischen Objekten
Richtung:	Master -> Bediengerät
Nutzdaten:	Attribut und Größe des grafischen Objekts: ATTRIBUT: Bit0=0: Löschen des angewählten Bereichs (Clear Pixel) Bit0=1: Anzeigen des angewählten Bereichs (Set Pixel) Bit1=1: Blinkende Darstellung des angewählten Bereichs Bit3=1: Inverse Darstellung des angewählten Bereichs STARTPOS: X,Y: Anfangsposition in Pixel (Wertebereich 0-255) ENDPOS: X, Y: Endposition in Pixel (Wertebereich 0-255) SHAPE: 0: Rechteck (beinhaltet: horizontale und vertikale Linien) 1: Gerade (wird auf Anfrage implementiert) 2: Kreis (wird auf Anfrage implementiert) 3: Raute (wird auf Anfrage implementiert)
DLN	8

Telegrammformat:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x2E	ATTRIBUT	STARTPOS-X	STARTPOS-Y	ENDPOS-X	ENDPOS-Y	SHAPE	nicht benutzt (0x00)

Beispiel (zeichnet ein Rechteck von (10,0)-(130,30))

0x2E	0x01	0x0A	0x00	0x82	0x1E	0x00	0x00
------	------	------	------	------	------	------	------

Handbuch Bediengeräte

1.4.44.2 Geräte der AT-Serie

Funktion:	Funktion zum Zeichnen von grafischen Objekten
Richtung:	Master -> Bediengerät
Nutzdaten:	<p>Attribut und Größe des grafischen Objekts: ATTRIBUT: Bit0=0: Löschen des angewählten Bereichs (Clear Pixel) Bit0=1: Anzeigen des angewählten Bereichs(Set Pixel) Bit1=0: Normale Darstellung Bit1=1: Blinkende Darstellung des angewählten Bereichs Bit3=1: Inverse Darstellung des angewählten Bereichs Bit7=0: statischer Bereich Bit7=1: dynamischer Bereich</p> <p>X1,Y1: Punkt 1 X2, Y2: Punkt 2 SHAPE: Zeichenfunktionen Bits 0-6 0: Rechteck nicht gefüllt. Punkt 1=linke obere Ecke; Punkt 2=rechte untere Ecke 1: Gerade von Punkt1 nach Punkt 2 2: Kreis, nicht gefüllt. Punkt 1=Mittelpunkt; D4=Radius 3: Raute (wird auf Anfrage implementiert) 4: Rechteck gefüllt. Punkt 1=linke obere Ecke; Punkt 2=rechte untere Ecke 5: Kreis, gefüllt. Punkt 1=Mittelpunkt; D4=Radius 6: Bildpunkt Farbe = Bit0 von D1 (schwarz/weiß) 7: Ellipse, gefüllt. Punkt 1=Mittelpunkt; D4=Radius X; D5=Radius Y Bit 7 0: indirektes Zeichnen im Speicher, danach Bildneuaufbau 1: indirektes Zeichnen im Speicher und auf dem Bildschirm, ohne Bildneuaufbau Unterschied: Methode 0 ist langsamer, überschreibt aber nur in der gewählten Zeichenebene. Methode 1 überschreibt im Bildschirm alle Bereiche, mit dem nächsten Bildaufbau (Änderung einer Variablen o.ä.) erscheint aber dann die korrekte Darstellung. Methode 1 bietet sich an, wenn man in Bereiche zeichnet, in denen keine sonstigen Elemente drin sind (leerer Bildbereich). Die Ausgabe erfolgt wesentlich schneller. Schiebeoperationen Quelle Punkt1 Ziel Punkt 2 Breite des Bereichs D1 Höhe des Bereichs D7 8: Bildschirmbereich schieben, statischer, nicht blinkender Bereich 9: Bildschirmbereich schieben, statischer, blinkender Bereich 10: Bildschirmbereich schieben, dynamischer, nicht blinkender Bereich 11: Bildschirmbereich schieben, dynamischer, blinkender Bereich</p>
DLN	8

Telegrammformat Zeichenfunktion / Schiebeoperation:

D0	D1	D2	D3	D4	D5	D6	D7
TA=0x2E	ATTRIBUT	X1	Y1	X2	Y2	SHAPE	nicht benutzt (0x00)
TA=0x2E	BREITE	X1	Y1	X2	Y2	SHAPE	HÖHE

Beispiel (zeichnet ein ein Rechteck von (10,0)-(130,30))

D0	D1	D2	D3	D4	D5	D6	D7
0x2E	0x01	0x0A	0x00	0x82	0x1E	0x00	0x00

2 Der CAN-Bus

Der CAN-Bus ist ein aus der Automobilindustrie kommendes Bussystem. Die Abkürzung bedeutet:

Controller
Area
Network

was soviel bedeutet wie etwa "Vernetzung zwischen Kleinststeuerungsgeräten".

Der CAN-Bus ist sehr einfach zu handhaben und ist sehr störungsempfindlich, wenn die Regeln zur Installation beachtet werden. Und er ist kostengünstig. Alle diese Gründe haben dazu beigetragen, den CAN-Bus in den Bediengeräten zu verwenden.

2.1 Verdrahtung

Der CAN-Bus ist - wie der Name schon sagt - ein Bussystem. In diesem System gelten Regeln, die eingehalten werden sollten. Geschieht dies nicht, kann nicht vorhergesagt werden, ob der Bus korrekt arbeitet oder nicht. Das geht soweit, dass einzelne Teilnehmer funktionieren, andere wiederum nicht. Im praktischen Betrieb hat sich der CAN-Bus aber als sehr unkompliziert erwiesen, wenn folgende Regeln eingehalten werden:

2.1.1 Busaufbau (Topologie)

Der CAN-Bus muß in einer Busstruktur aufgebaut sein, bei denen beide Enden jeweils mit einem Abschlußwiderstand von 120 Ohm versehen sind. Wenn eine Sterntopologie benötigt wird, müssen Repeater eingebaut werden, die lange Stichleitungen von anderen Bussegmenten elektrisch abkoppeln. Es gelten in diesem Zusammenhang die einschlägigen Aufbauvorschriften für CAN-Bussysteme.

2.1.1.1 Leitungsverbindungen

Als minimale Verdrahtung wird empfohlen, die Leitungen CAN-L, CAN-H und die Abschirmung CAN-SHLD zu verwenden. Legen Sie den Schirm beidseitig auf. Die Leitungen CAN-L und CAN-H dürfen an den Teilnehmern nicht vertauscht werden, sondern müssen 1:1 durchverdrahtet sein. Empfohlen wird, diese Leitungen auf ein verdrehtes Adernpaar zu führen.

Zu Erhöhung der Übertragungssicherheit können Sie noch die Leitung CAN-GND auf allen Teilnehmern anschließen. Wenn Sie in Ihrem Buskabel zusätzliche verdrehte Adernpaare haben, so schließen Sie bitte beide Adern eines verdrehten Paares an CAN-GND an.

2.1.1.2 Buskabel

Für Laboraufbauten mit kurzen Leitungen in störungsfreier Umgebung, können zu Testzwecken auch ungeschirmte verdrehte Leitungen verwendet werden. In Ihrer Anlage sollte ein Kabel nach DIN/ISO 11898 verwendet werden.

Die spezifischen Eigenschaften des Kabels finden Sie in der Norm. Im Zweifelsfall wenden Sie sich an einen Kabelhersteller, der Buskabel speziell für CAN herstellt.

2.1.1.3 Leitungslängen

Die folgende Tabelle gibt einen Anhaltswert für die maximale Kabellänge des CAN-Bus:

Baudrate	Widerstand	Leitungsquerschnitt	Max. Kabellänge
10 kBit/s	< 18 mOhm/m	1,00 mm ²	2000 m
20 kBit/s	< 25 mOhm/m	0,80 mm ²	1000 m
50 kBit/s	< 30 mOhm/m	0,65 mm ²	700 m
100 kBit/s	< 40 mOhm/m	0,50 mm ²	500 m
125 kBit/s	< 44 mOhm/m	0,45 mm ²	400 m
250 kBit/s	< 50 mOhm/m	0,40 mm ²	200 m
500 kBit/s	< 60 mOhm/m	0,34 mm ²	100 m
1 MBit/s	< 70 mOhm/m	0,25 mm ²	40 m

Bei größerer Leitungslänge sollten Sie ein Kabel verwenden mit geringerem spez. Widerstand - also dickeres Kabel.

Sollten Sie eine Anlage planen, die diese angegebenen Längen übersteigt, dann können Sie auch 2 CAN-Segmente planen und diese über einen Repeater verbinden. Der Repeater sorgt dafür, dass alle Nachrichten auf beiden Segmenten vorhanden sind, die Segmente jedoch elektrisch voneinander entkoppelt sind.

2.1.2 Terminierung

An den beiden Enden des Busses muss zwischen den Leitungen CAN-L und CAN-H ein Widerstand 120 Ohm verdrahtet werden. Am CAN-Modul dient dazu die 4-polige Schraubklemmenleiste für den CAN-Bus, einfach den Widerstand mit hineinstecken und Schrauben festziehen. Am Bediengerät finden Sie einen schaltbaren Widerstand. Neben dem DB-9 CAN-Steckverbinder ist ein Schiebeshalter. Wenn Sie diesen in die Stellung ON schieben, ist der intern eingebaute Busabschluss aktiviert.

2.1.3 Adressierung

Im Bus muss jedes Gerät eine eindeutige Nummer (bzw. einen eindeutigen Identifier) haben -

Handbuch Bediengeräte

vergeben Sie also keine Nummern doppelt. Die CAN-Module haben DIP-Schalter zum Einstellen der Adressen. Die Bediengeräte werden über den Editor auf eine Adresse eingestellt.

2.2 Protokolle allgemein

Auch für den CAN-Bus gibt es verschiedene Protokollspezifikationen. Obwohl der Datentransport bei allen Protokollen über den CAN-Controller identisch ist, wird die Vergabe der Identifier (=Adresse) und der Telegramminhalt von den verschiedenen Protokollen jeweils gesondert behandelt.

Die Bediengeräte beherrschen insgesamt 3 Protokolle.

2.2.1 Das SELECAN-Protokoll

Ursprünglich von der Firma SELECTRON entwickelt, hat dieses Protokoll eine nur geringe Verbreitung erhalten. Es basiert auf einem Master (Host), der die Konfiguration und Überwachung der angeschlossenen Slaves (Module) übernimmt. Es ist aufgrund der geringen Verbreitung als proprietäres Protokoll anzusehen.

2.2.1.1 Zusammensetzung des Identifiers

Der 11-bit CAN-Identifier enthält insgesamt 3 Bereiche zur Kennung; 3 bit für die Nachrichtenpriorität, 5 bit Knotenadresse (ergibt max. 32 Teilnehmer) und 3 bit Kennung für die Meldungsrichtung. Der Anwender kann nur die Knotenadresse festlegen, wobei der Master immer die Knotenadresse 0 bekommen sollte (wegen der Nachrichtenpriorität). Das SELECAN-Protokoll ist eigentlich nur erforderlich, wenn das ITS6 zusammen mit Modulen der Serie GCM betrieben werden soll. Die Datenlänge (DLC) ist immer 8, das RTR-Bit (R) immer 0. Das Identifier-Format ist also:

15-13			12-8					7-5			4	3-0						
Prio			Knotenadresse					Spez.				DLC						
x	x	x	x	x	x	x	x	x	x	x	0	1	0	0	0			
Master sendet an Modul											1	1	1	0	x	x	x	x
Digitales Modul an Master											0	1	0	0	x	x	x	x
Analoges Modul an Master											0	1	1	0	x	x	x	x

2.2.1.2 Identifiertabelle im SELECAN-Protokoll

Als Identifier (betrachtet werden die Bits 15 bis 5 = 11 bit) werden benötigt:

Ad	P0	P1	P2	P3	P4	P5	P6	P7
0	0-7	256-263	512-519	768-775	1024-1031	1280-1287	1536-1543	1792-1799
1	8-15	264-271	520-527	776-783	1032-1039	1288-1295	1544-1551	1800-1807
2	16-23	272-279	528-535	784-791	1040-1047	1296-1303	1552-1559	1808-1815
3	24-31	280-287	536-543	792-799	1048-1055	1304-1311	1560-1567	1816-1823
4	32-39	288-295	544-551	800-807	1056-1063	1312-1319	1568-1575	1824-1831
5	40-47	296-303	552-559	808-815	1064-1071	1320-1327	1576-1583	1832-1839
6	48-55	304-311	560-567	816-823	1072-1079	1328-1335	1584-1591	1840-1847
7	56-63	312-319	568-575	824-831	1080-1087	1336-1343	1592-1599	1848-1855
8	64-71	320-327	576-583	832-839	1088-1095	1344-1351	1600-1607	1856-1863
9	72-79	328-335	584-591	840-847	1096-1103	1352-1359	1608-1615	1864-1871
10	80-87	336-343	592-599	848-855	1104-1111	1360-1367	1616-1623	1872-1879
11	88-95	344-351	600-607	856-863	1112-1119	1368-1375	1624-1631	1880-1887
12	96-103	352-359	608-615	864-871	1120-1127	1376-1383	1632-1639	1888-1895
13	104-111	360-367	616-623	872-879	1128-1135	1384-1391	1640-1647	1896-1903
14	112-119	368-375	624-631	880-887	1136-1143	1392-1399	1648-1655	1904-1911
15	120-127	376-383	632-639	888-895	1144-1151	1400-1407	1656-1663	1912-1919
16	128-135	384-391	640-647	896-903	1152-1159	1408-1415	1664-1671	1920-1927
17	136-143	392-399	648-655	904-911	1160-1167	1416-1423	1672-1679	1928-1935
18	144-151	400-407	656-663	912-919	1168-1175	1424-1431	1680-1687	1936-1943
19	152-159	408-415	664-671	920-927	1176-1183	1432-1439	1688-1695	1944-1951
20	160-167	416-423	672-679	928-935	1184-1191	1440-1447	1696-1703	1952-1959
21	168-175	424-431	680-687	936-943	1192-1199	1448-1455	1704-1711	1960-1967
22	176-183	432-439	688-695	944-951	1200-1207	1456-1463	1712-1719	1968-1975
23	184-191	440-447	696-703	952-959	1208-1215	1464-1471	1720-1727	1976-1983
24	192-199	448-455	704-711	960-967	1216-1223	1472-1479	1728-1735	1984-1991
25	200-207	456-463	712-719	968-975	1224-1231	1480-1487	1736-1743	1992-1999
26	208-215	464-471	720-727	976-983	1232-1239	1488-1495	1744-1751	2000-2007
27	216-223	472-479	728-735	984-991	1240-1247	1496-1503	1752-1759	2008-2015
28	224-231	480-487	736-743	992-999	1248-1255	1504-1511	1760-1767	2016-2023
29	232-239	488-495	744-751	1000-1007	1256-1263	1512-1519	1768-1775	2024-2031
30	240-247	496-503	752-759	1008-1015	1264-1271	1520-1527	1776-1783	2032-2039
31	248-255	504-511	760-767	1016-1023	1272-1279	1528-1535	1784-1791	2040-2047

P0: Service, Broadcast, Host to Mod., Command
 P1: Service, Host to Module, Command
 P2: Service, Broadcast, Module to Host, Status
 P3: Service, Module to Host, Status
 P4: Data, Host to Module, Interrupt-Daten
 P5: Data, Host to Module
 P6: Data, Module to Host, Interrupt-Daten
 P7: Data, Module to Host
 (P0-P7: 3 Bit für die NachrichtenPriorität)

Handbuch Bediengeräte

2.2.2 Telegramminhalte

Die Telegramminhalte sind genau spezifiziert. Sie erhalten über das SELECAN-Protokoll auf Anfrage ein gesondertes Handbuch.

2.2.3 Bediengerät an SELECAN-SPS

Die Bediengeräte benutzen für die Kommunikation mit der Steuerung lediglich die Daten-Kanäle des SELECAN-Protokolls, das heißt, das Bit Nr. 10 ist immer auf 1 gesetzt. Über diese Daten-Kanäle können sämtliche Funktionen bedient werden. Dabei ist die Telegramm-Länge (Größe des Datenbereichs) immer auf 8 bytes (Maximalgröße) festgelegt. Die Adresse des Bediengerätes ("Gerätenummer") wird im ITE eingestellt. (Siehe Kapitel 10)

2.2.3.1 Steuerung --> ITS

Der Identifier hat folgendes Format (11-bit):
Bit-Nr.:

15	14	13	12	11	10	9	8	7	6	5
1	0	0	Adr4	Adr3	Adr2	Adr1	Adr0	1	1	1

Die Adresse errechnet sich zu:
Identifier=1031 + 8 x Gerätenr.

2.2.3.2 ITS-6000 --> Steuerung

Der Identifier hat folgendes Format (11-bit):

15	14	13	12	11	10	9	8	7	6	5
1	1	1	Adr4	Adr3	Adr2	Adr1	Adr0	0	1	0

Hier erhalten wir:
Identifier=1794 + 8 x Gerätenr.

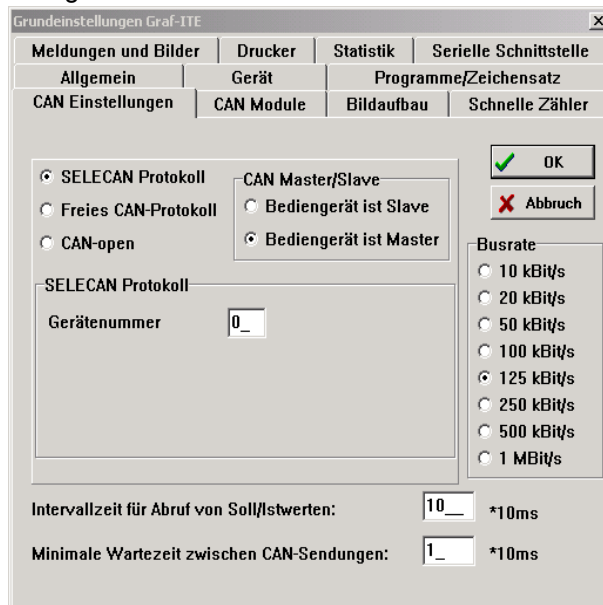
2.2.4 Verwenden von GCM-Modulen

Die Bediengeräte können CAN-Module der Serie GCM eigenständig überwachen und auswerten. Dies ermöglicht in vielen Fällen eine kostengünstige Lösung, um z.B. Fehlerüberwachungen in Anlagen oder Gebäuden zu installieren. Das Gerät kann dann als eigenständiges System ohne externe Steuerung arbeiten. Die Steuerlogik (SPS-Funktion) können Sie als Kontaktplan oder in C erstellen. Einfache Funktionen wie "Eingang auf Meldung zuordnen" oder "Zähler auf Eingang zuordnen" können über das Modul-Konfigurationstool erstellt werden, ohne eine einzige Zeile Programm schreiben zu müssen.

Erst wenn Logikfunktionen wie "Wenn Eingang 1 und Eingang 2, dann Bild 27" notwendig sind, führt am Steuerprogramm kein Weg vorbei.

2.2.4.1 Voraussetzungen

Derzeit ist für eine eigenständige Anlage mit Bediengerät und CAN-Modulen das SELECAN-Protokoll notwendig. Dies muss in der Maske "Geräte"/"Parametrieren", Registerkarte "CAN-Einstellungen" aktiviert werden. Dabei muss das Master-Bediengerät (es können auch mehrere Bediengeräte in den CAN-Bus geschaltet werden) auf die Geräteadresse 0 eingestellt und als Master konfiguriert sein:



In dem Feld "CAN Konfiguration" müssen Sie "ITS ist CAN-Master" einstellen.

Die Baudrate können Sie frei wählen.

2.2.4.2 Master-Slave-Konfigurationen

Wenn Sie ein zweites Bediengerät zur Beobachtung von Vorgängen einsetzen wollen, dann können Sie beim Slave dasselbe Projekt wie beim Master verwenden. Stellen Sie einfach "Bediengerät ist Slave", und eine unbenutzte Gerätenummer ungleich 0 ein.

Wenn Sie Ausgänge von beiden Bedienstationen aus ansteuern müssen, dann muß die Koordination beim Master gemacht werden. Das heißt, wenn Sie vom Slave aus einen Ausgang schalten wollen, so sollten Sie ein Telegramm an den Master senden (mit dem internen Steuerprogramm), in dem Sie z.B. eine Variable senden, die eine 0 für Ausgang aus, eine 1 für Ausgang ein enthält. Im Master-Steuerprogramm brauchen Sie dann nur den Wert der Variablen abfragen und den Ausgang entsprechend ansteuern. Das hört sich komplizierter an als es ist; im Steuerprogramm gibt es die Funktion „CAN-Telegramm senden“. Wenn Sie diese benutzen und dabei das Telegramm so aufbauen wie bei der Telegrammbeschreibung "SET VARIABLE VALUE" beschrieben, dann können Sie über eine einzige Variable bis zu 32 Aus-

Handbuch Bediengeräte

gänge schalten. Jede Variable hat bis zu 32 bit und jedes Bit kann dabei einzeln abgefragt werden (Hinweis: Variable in ein Merkerdoppelwort @MD kopieren, dann mit Merkern @Mx.y zugreifen.)

2.2.4.3 Einstellen der CAN-Module

Die CAN-Module verfügen über eine Reihe von DIP-Schaltern. Mit diesen können Sie die Baudrate und die Knotenadresse/Modulnummer einstellen.

Hierzu die folgenden Tabellen:

Baudrate	DIP 6	DIP 7	DIP 8
10 kBit/s	off	off	off
20 kBit/s	on	off	off
50 kBit/s	off	on	off
100 kBit/s	on	on	off
125 kBit/s	off	off	on
250 kBit/s	on	off	on
500 kBit/s	off	on	on
1 MBit/s*	on*	on*	on*

* Diese Baudraten-Einstellung ist z.Zt. nicht möglich. Diese wird von den Buskopplern der Module nicht unterstützt. Sollten Sie diese Busrate benötigen, so setzen Sie sich bitte mit uns in Verbindung.

Die Moduladresse vergeben Sie wie folgt:

Nr.	DIP 1	DIP 2	DIP 3	DIP 4	DIP 5
1	on	off	off	off	off
2	off	on	off	off	off
3	on	on	off	off	off
4	off	off	on	off	off
5	on	off	on	off	off
6	off	on	on	off	off
7	on	on	on	off	off
8	off	off	off	on	off
9	on	off	off	on	off
10	off	on	off	on	off
11	on	on	off	on	off
12	off	off	on	on	off
13	on	off	on	on	off
14	off	on	on	on	off
15	on	on	on	on	off
16	off	off	off	off	on
17	on	off	off	off	on
18	off	on	off	off	on
19	on	on	off	off	on
20	off	off	on	off	on
21	on	off	on	off	on
22	off	on	on	off	on
23	on	on	on	off	on
24	off	off	off	on	on
25	on	off	off	on	on
26	off	on	off	on	on
27	on	on	off	on	on
28	off	off	on	on	on
29	on	off	on	on	on
30	off	on	on	on	on
31	on	on	on	on	on
0	Reserviert für den Master (Bediengerät)				

2.2.4.4 Zuordnung Geräteadresse/Identifier

Wenn Sie mehrere Bediengeräte verwenden wollen, müssen Sie ggf. Daten zwischen diesen Geräten austauschen. Dies können Sie mit dem Kontaktplan KOP über die Funktion "CAN-Telegramm senden" erreichen. Aber Sie müssen wissen, welche Identifier Sie eintragen müssen. Die folgende Tabelle gibt eine Zuordnung von der Geräteadresse auf den Empfangs-Identifier des Datenkanals, den Sie im KOP einstellen müssen:

Geräteadresse	Empfangs-Identifier
0	1.031
1	1.047
2	1.055
3	1.063
4	1.071
5	1.079
6	1.087
7	1.095
8	1.103
9	1.111
10	1.119
11	1.127
12	1.135
13	1.143
14	1.151
15	1.159
16	1.167
17	1.175
18	1.183
19	1.191
20	1.199
21	1.207
22	1.215
23	1.223
24	1.231
25	1.239
26	1.247
27	1.255
28	1.263
29	1.271
30	1.279
31	1.287

Wie gesagt: Diese Identifier geben ausschließlich den Daten-Empfangskanal des jeweiligen Geräts an. Es werden auch weitere Identifier z.B. für Status- und Konfigurationsnachrichten verwendet. Diese finden Sie in einer Tabelle weiter oben.

2.2.4.5 Module aus dem internen Steuerungsprogramm ansprechen

Im Anhang finden Sie Hinweise zur Nummerierung der Eingänge, die von den hier aufgeführten Nummerierungen abweichen.

Handbuch Bediengeräte

KOP kann Module mit den Adressen 1-8 direkt über die KOP-Variablen "@Dlx.y" (Digital In), "@DOx.y" (Digital Out), "@Alx.y" (Analog In) und "@AOx.y" (Analog Out) ansprechen; x steht für die Moduladresse und y für die Ein/Ausgangsnummer.

Generell müssen Sie entscheiden, ob Sie Ausgangsfunktionen der Module über das Steuerprogramm steuern wollen oder ob Sie über das CAN-Konfigurationstool die Ausgänge über Variablen steuern wollen. Sie können nicht gleichzeitig über @DO und eine Variable auf einen Ausgang gehen. Bei Eingängen ist dies möglich.

Bei Modulen mit einer Adresse größer als 8 müssen Sie alles über interne Variablen machen, auf die das interne Steuerprogramm dann auch Zugriff hat.

2.2.5 Das freie CAN-Protokoll

Mit dem freien CAN-Protokoll kann das Bediengerät auf beliebige Identifier, sowohl für das Senden, als auch für das Empfangen, eingestellt werden. Für den Empfang sind bis zu 8 Empfangs-Identifier einstellbar. Dies wird interessant, wenn mehrere Steuerungen oder Bediengeräte miteinander kommunizieren sollen. Da dieses Protokoll das flexibelste ist, wird es auch am häufigsten eingesetzt.

Eine Identifiertabelle finden wir hier nicht, da alles frei eingestellt werden kann.

2.2.5.1 Zusammensetzung des Identifiers

Die Datenlänge der Telegramme (beim Senden und empfangen über das TOS, nicht KOP senden) ist immer 8, und das RTR-Bit (R) wird nicht beachtet.

Das Identifier-Format ist also:

15-5	4	3-0
Identifier	R	DLC
x x x x x x x x x x x x	x	1 0 0 0

2.2.5.2 Telegramminhalte

Der Telegramminhalt ist im Anhang beschrieben. Der Telegramminhalt wird bestimmt durch das "Multiplexer-Byte D0", anhand dessen Wert wird der nachfolgende Telegramminhalt D1-D7 ausgewertet.

Sowohl gesendete wie auch empfangene Telegramme müssen dieses Telegrammformat einhalten.

2.2.6 Das CANopen-Protokoll

CANopen ist ein Protokoll, das aus mehreren Ebenen besteht. Es legt fest, wie die Identifier der Teilnehmer aussehen sollten; und es teilt Datenübertragungen auf in die 3 Bereiche Netz-

werkmanagement (NMT), Prozeßdatentransfer (PDO) und Parametrierdatentransfer (SDO). Geräte, die sich an diese Spezifikationen halten, können grundsätzlich in einem CANopen-Netzwerk betrieben werden.

Darüber hinaus definiert CANopen sogenannte "Geräteprofile". In diesen Profilen wird für gleiche Gerätetypen (z.B. Drehgeber oder Frequenzrichter) jeweils ein Grundstandard für Parametrierdaten und Prozeßdaten definiert. Geräte von Herstellern, die sich an diese Geräteprofile halten, können auf dieselbe Weise angesteuert werden (meistens zwar nur die Grundfunktionen, aber immerhin!). Wie die Datenübertragung geschieht, legt wiederum das CANopen-Protokoll fest. CANopen ermöglicht den Betrieb von 128 Teilnehmern.

2.2.6.1 Grundverhalten als SLAVE

Das Bediengerät verhält sich als Slave wie ein digitales I/O-Modul nach dem Standard DS-401. Zur Zeit der Treiberentwicklung gab es noch keine Spezifikation (Profil) für Bediengeräte. Als Prozeßdaten (PDO) liefert bzw. empfängt das Bediengerät CANopen-Protokoll. Das Telegrammformat finden Sie im Anhang (man könnte die Übertragungsart als "multiplexed PDO" bezeichnen). Zusätzlich sind die SDO- bzw. NMT-Dienste implementiert.

2.2.6.2 Grundverhalten als MASTER

Als Master ist eine erweiterte Funktionalität verfügbar: das Bediengerät kopiert selbsttätig empfangene PDO's in Variablen mit entsprechendem Handle, falls vorhanden. Über das Steuerprogramm ist dann eine Auswertung von PDO-Daten ganz einfach möglich.

Ganz ins Detail gehen wir hier nicht. Im Anhang gibt es noch ein eigenes Kapitel über CAN-open.

2.2.6.3 Identifiertabelle von CANopen

Im CANopen können die Identifier über SDO-Dienste frei vergeben werden. Empfohlen wird aber, die Identifier nach dem sogenannten „Predefined connection set" zu vergeben, um einen Grundstandard zu erhalten. CANopen errechnet diese Identifier über einfache Formeln auf Basis der sogenannten Knotennummer „Node-ID" (alle Angaben in dezimaler Darstellung):

Art	ID =	ID-Bereich
Netzwerkmanagement NMT	Node-ID	0-127
Emergency EMCY	Node-ID+128	128-255

Sendeprozessdaten (TX-PDO)	Node- ID+384	384-511
Empfangsprozessdaten (RX-PDO)	Node- ID+512	512-639
Sendesystemdaten (TX-SDO)	Node- ID+1408	1408-1535
Empfangssystemdaten (RX-SDO)	Node- ID+1536	1536-1663
Knotenüberwachung (GUARD)	Node- ID+1792	1792-1919

2.2.7 Mischen von Protokollen

Wenn man auf demselben CAN-Bus mehrere Protokolle "fahren" will, muß man lediglich darauf achten, daß Identifier nicht doppelt vergeben werden. Außerdem müssen die Master in der Lage sein, entsprechende Identifier aus der Überwachung herauszuhalten. Sonst könnte eine Nachricht im falschen Protokoll ausgewertet werden und zu Fehlfunktionen führen.

3 CAN-Open Treiber

CANopen ist eine Protokolldefinition auf CAN-BUS-Hardware. Es handelt sich dabei um die Spezifikation 4.0 von CANopen, die auch der Programmierung der Bediengeräte zu Grunde liegt. Das vorliegende Handbuch soll nicht die CANopen-Spezifikation aufnehmen, sondern die Eigenschaften des Bediengerätes als CANopen-Teilnehmer darstellen.

In der vorliegenden Implementierung werden folgende Merkmale von CANopen unterstützt.

MASTER:

- Minimales Netzwerk-Management
- Senden und Empfangen von SDO's
- Senden und Empfangen von PDO's

SLAVE:

- Minimum-Bootup-Verhalten
- Pre-defined Connection Set
- Kein PDO mapping
- Boot-Up Node-Guard Frame

Diese Dokumentation verwendet die in der CANopen-Literatur verwendete Notation für Telegrammdarstellung usw.

3.1 Anforderungen

3.1.1 Betriebssystem (TOS)

Da die CANopen-Funktionalität nicht mehr in das Standard-Betriebssystem aufgenommen werden konnte, wird für CANopen ein eigenes Betriebssystem (TOS) mitgeliefert. Dieses TOS hat dieselbe Funktionalität wie das Standard-TOS, kann aber nur CANopen-Protokoll auf dem CAN-Bus fahren.

Konkret:

- Standard-TOS Bediengerät: IO0xxSxx.hex
- Standard-TOS ITS7000: IO1xxSxx.hex
- CANopen-TOS Bediengerät: IO0xxAxx.hex
- CANopen-TOS ITS7000: IO1xxAxx.hex (x variiert)

3.1.2 Firmware (BIOS)

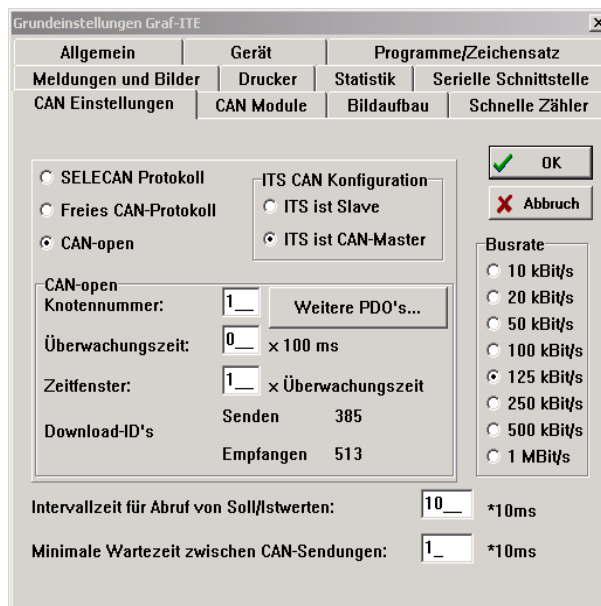
Ein besonderes BIOS ist nicht erforderlich. Das heißt, die CANopen-Software kann auch auf älteren Geräten eingesetzt werden.

3.1.3 Projektiersoftware (Editor ITE)

Um CANopen parametrieren zu können, ist die Projektiersoftware ITE6D16 oder eine neuere Version erforderlich. Diese Projektiersoftware berücksichtigt bereits die zusätzlichen TOS-Versionen für CANopen.

3.1.4 Einstellungen im ITE

Der Editor wurde um die CANopen Einstellmöglichkeiten ergänzt. Diese finden Sie, wenn Sie auf "Geräte"/"Parametrieren" in die Registerkarte "CAN-Einstellungen" klicken:



Diese Einstellfelder erscheinen, wenn Sie "CAN-Open" auswählen. Die Bedeutung der Felder ist:

3.1.5 Feld ITS-CAN Konfiguration

Hier stellen Sie ein, ob das Bediengerät als CAN-Open Master oder Slave verwendet werden soll. Details über den Unterschied "Master - Slave" finden Sie weiter hinten in dieser Dokumentation.

3.1.6 Feld Knotennummer

In diesem Feld stellen Sie ein, welche Knotennummer (Node-ID) das Bediengerät haben soll. Über die Knotennummer stellen Sie die SDO- und PDO Identifier (COB-IDs) ein, so wie sie von dem "Pre-defined connection set" von CANopen vorgeschlagen werden (aus der Sicht des Bediengeräts):

Emergency-Objekt	=	128 + Node-ID
TX-PDO-ID	=	384 + Node-ID
RX-PDO-ID	=	512 + Node-ID
TX-SDO-ID	=	1408 + Node-ID
RX-SDO-ID	=	1536 + Node-ID
Node Guard	=	1792 + Node-ID

3.1.7 Feld Überwachungszeit

Dieses Feld ist nur für den Slave von Bedeutung. Hier wird die Knotenüberwachungszeit (Node-Guard time) eingestellt, in vielfachen von 100 ms.

3.1.8 Feld Zeitfenster

Dieses Feld ist nur für den Slave von Bedeutung.

Handbuch Bediengeräte

Hier wird der "Lebenszyklus" (Life-Time) eingestellt. Die Bedeutung des Parameters ist, wie oft die unter "Überwachungszeit" eingestellte Zeit ablaufen darf, bevor das Bediengerät einen Fehler erkennt und meldet.

3.1.9 Anzeigefelder Download-ID's

In diesen Feldern erscheinen die Identifier, unter denen das Bediengerät einen Projekt-Download vornimmt.

Diese Einstellungen benötigen Sie, um einen Transfer des Projekts vom Editor (PC) zum Bediengerät vornehmen zu können. Tragen Sie in der Maske "Geräte"/"Schnittstelle" die hier angezeigte Download-ID "Empfangen" als Sendee-Identifizier ein.

Nebenbei: dies sind die Identifier (COB-ID's), unter denen die PDO-Transfers stattfinden.

3.1.10 Feld Intervallzeit für Abruf...

Dieses Feld ist nur für den Slave von Bedeutung. Hier bestimmen Sie, wie oft das Bediengerät die externen Variablen neu anfordert. Siehe Handbuch Bediengeräte: Bedienen und Beobachten.

3.1.11 Feld Minimale Wartezeit

Hier bestimmen Sie, wie viel Zeit mindestens zwischen zwei vom Bediengerät gesendeten CAN-Telegrammen verstreichen soll. Mithin können Sie hier also die Buslast für das Bediengerät einschränken.

Diese Einstellung wird von den NMT-Diensten ignoriert, da es hier zu Timeouts kommen könnte.

3.2 Die MASTER-Implementierung

Ziel der MASTER-Implementierung war, einfache CANopen-Teilnehmer über das Bediengerät anzusteuern und ggf. zu parametrieren. Dabei wurden einige Mechanismen implementiert, die nicht erst über das Steuerprogramm als KOP oder in C realisiert werden müssen.

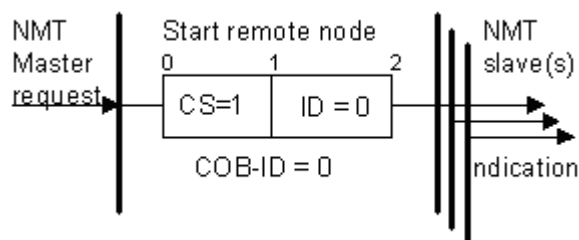
In der Registerkarte "CAN Einstellungen" (unter "Geräte"/"Parametrieren"), Feld "ITS CAN Konfiguration" wird der Master aktiviert.

3.2.1 Minimales Netzwerk-Management

Das Netzwerk-Management des Masters enthält die automatische Inbetriebnahme von Knoten. Eine Knotenüberwachung (Node-Guarding) ist nicht implementiert. Diese muß per KOP realisiert werden.

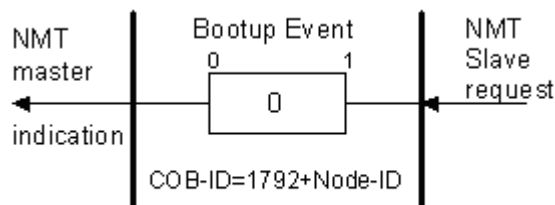
3.2.1.1 Master-Boot-Up

Der Master sendet beim Start ein "BROADCAST START REMOTE NODE" an alle Knoten:

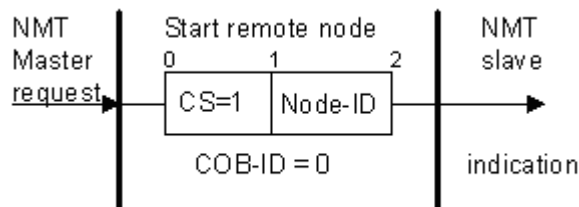


3.2.1.2 Slave-Boot-Up (Version 4.0)

Version 4.0 von CANopen fordert, daß ein Slave beim Übergang von "Initialisation" auf "Pre-Operational" ein Telegramm "Bootup Event" mit DL=1 und D0=0 sendet:



Auf dieses Telegramm antwortet der Master mit einem "Start remote node" speziell für diesen Knoten:

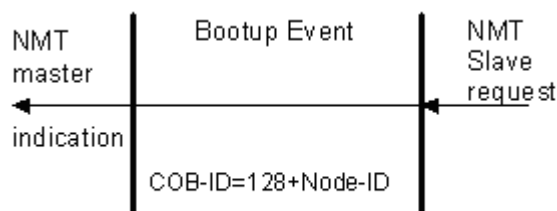


Somit werden Knoten, die einen Ausfall hatten oder sich neu anmelden, automatisch in Betrieb genommen.

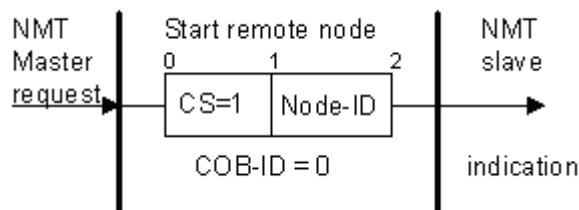
3.2.1.3 Slave-Boot-Up (RFC zu Version 3.0)

In diesem RFC wurde gefordert, dass ein Slave beim Übergang von "Initialisation" auf "Pre-Operational" ein "Emergency" ohne Inhalt sendet:

Handbuch Bediengeräte



Auch auf dieses Telegramm antwortet der Master mit einem "Start-Remote-Node":

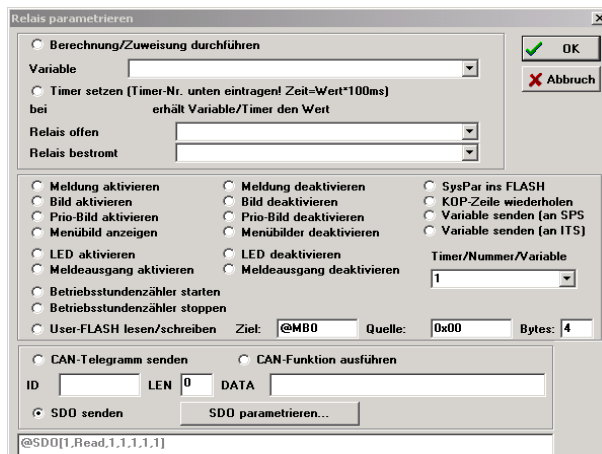


3.2.2 Senden und Empfangen von SDO's

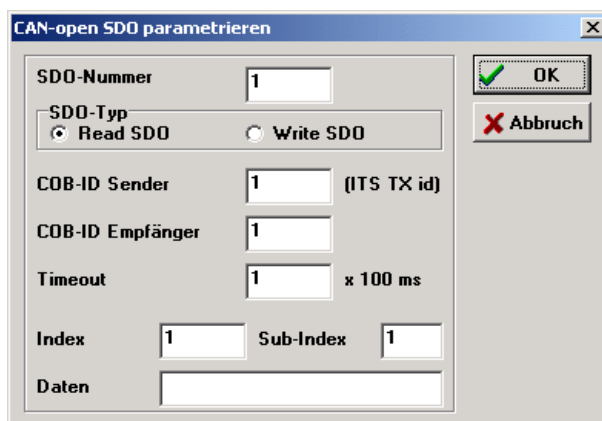
Das Senden und Empfangen von SDO's kann der Master nicht automatisch vornehmen. Jede Anwendung hat ihren eigenen Datenaustausch. Der Master überwacht aber den korrekten Austausch von SDO-Daten und zeigt Fehler an. Der Austausch von SDO-Daten wird über das Steuerprogramm, programmiert in KOP oder C realisiert.

3.2.2.1 Senden/Anfordern von SDO-Daten

Mit der Relais-Funktion "SDO senden" startet der KOP eine SDO-Sendung oder eine SDO-Abfrage. Sinnvollerweise wird so eine Funktion immer aufgrund einer Bedingung gestartet, siehe hierzu das Anwendungsbeispiel weiter unten. In der folgenden Maske wird die Funktion ausgewählt:



Drückt man den Knopf "SDO parametrieren...", dann erhält man die Einstellmöglichkeiten für die SDO-Parameter:



Feld SDO-Nummer

In diesem Feld müssen Sie eine Zahl eingeben, unter der Sie später die Statusabfrage für diesen SDO machen wollen. Das Bediengerät kann max. 15 SDO's zur selben Zeit bearbeiten. Vermeiden Sie, dass verschiedene SDO's mit derselben SDO-Nummer versehen werden. Dies könnte zu fehlerhaften Informationen führen. Die SDO-Nummer benötigt das Gerät nur intern. Diese Information erscheint nicht auf dem CAN-Bus.

Feld SDO-Typ

Hier können Sie auswählen, ob Sie SDO-Daten von einem anderen Knoten lesen wollen (Read SDO) oder ob Sie SDO-Daten an einen anderen Knoten senden wollen (Write SDO).

Feld COB-ID Sender

In dieses Feld geben Sie ein, mit welchem COB-ID das Bediengerät diesen SDO absenden soll. Die Eingabe ist dezimal, es kann aber auch eine Variable eingegeben werden (wie im Beispiel gezeigt).

Feld COB-ID Empfänger

Hier geben Sie ein, unter welchem COB-ID die

Handbuch Bediengeräte

Antwort/Bestätigung des Empfängers erwartet wird. Auch hier kann eine Variable eingegeben werden.

Feld Timeout

Geben Sie hier an, wie lange das Bediengerät maximal auf eine Antwort/Bestätigung warten soll.

Feld Index

Geben Sie hier den Index des SDO an. Die Eingabe ist standardmäßig dezimal, kann aber durch voranstellen von "0x" auch hexadezimal angegeben werden. Beispiel: 0x1000. Auch Variablen sind hier möglich.

Feld Sub-Index

Enthält den Sub-Index des SDO's. Eingaben wie beim Index.

Feld Daten

Dieses Feld müssen Sie nur ausfüllen, wenn Sie SDO-Daten an einen Knoten senden wollen. Die Eingabe ist wiederum dezimal, hexadezimal ("0x") oder eine Variable.

3.2.2.2 Abfrage SDO-Antwort

WICHTIG !!!

FÜR DIE ABFRAGE VON SDO-ANTWORTEN IM KOP IST IMMER DER STANDARD-SCHLIESER OHNE ABRUCH ZU VERWENDEN !!!

Die Statusabfrage einer SDO-Sendung oder -Anforderung erfolgt immer mit einem Standard-Schließer.

Der Zweig hinter dem Schließer wird bestromt durchlaufen, wenn eine erfolgreiche Antwort gekommen ist. Der Durchlauf erfolgt ein einziges Mal, danach wird der SDO aus der internen Überwachung herausgenommen.

Wenn ein Timeout oder ein SDO-Fehler auftritt, so wird der Zweig hinter dem Schließer unbestromt durchlaufen. Der Durchlauf erfolgt ebenfalls ein einziges Mal, danach wird der SDO aus der internen Überwachung herausgenommen.

Mit dieser Methodik kann man den SDO-Datenaustausch überwachen.

Die Parametrierung des Schließers wird in der Schließer-Maske vorgenommen:

Feld "Nummer/Variable/SDO"

In dem Feld "Nummer/Variable/SDO" geben Sie die SDO-Nummer ein, die Sie in der Maske "SDO Parametrieren" verwendet haben. So hat der KOP die Zuordnung, welchen SDO er überwachen soll.

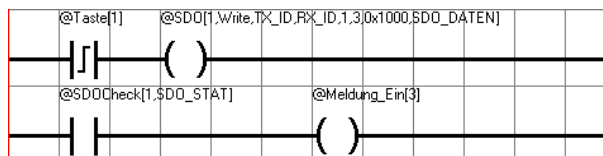
Feld "SDO Statusergebnis"

In das Feld "SDO Statusergebnis" tragen Sie ein, wo KOP das Ergebnis bzw. die Daten ablegen soll. Dies kann ein Merkerwort oder eine Variable sein.

Wie gesagt: Durchlauf des Zweigs mit Strom: SDO ist ok (Antwort ist dann in der Variablen) oder Durchlauf des Zweigs ohne Strom: SDO Fehler. In der Variablen findet sich dann ein erweiterter Fehlercode, der in der CANopen-Dokumentation näher erklärt wird.

3.2.3 Beispiel: SDO senden

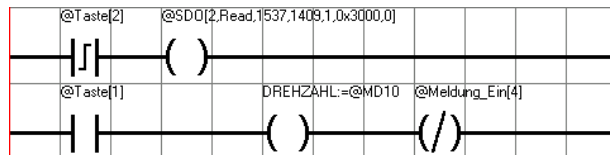
Im folgenden Beispiel werden Daten für einen SDO abgesendet, wenn die Taste 1 gedrückt wird. Die Überwachung aktiviert die Meldung 3, wenn ein Timeout erreicht wurde oder der SDO ungültig war.



3.2.4 Beispiel: SDO lesen

Im folgenden Beispiel werden die angeforderten SDO-Daten in die Variable DREHZAHN gespeichert, oder die Meldung 4 wird im Fehlerfall aufgerufen:

Handbuch Bediengeräte



Auf diese Weise ist das Lesen und Schreiben von SDO's mit wenig Programmieraufwand möglich.

3.2.5 Senden und Empfangen von PDO's

Das Senden von PDO's ist wesentlich einfacher als das SDO-Handling, da hier keine Antworten beachtet werden müssen.

3.2.5.1 Senden von PDO's

Diese Funktion wird von der Relais-Funktion "CAN-Telegramm senden" bereits seit langem erfüllt. Folgende Felder sind maßgeblich:

Feld "ID"

Tragen Sie hier den COB-ID ein, unter dem der PDO gesendet werden soll. Die Eingabe kann dezimal, hexadezimal ("0x") oder eine Variable sein.

Feld "LEN"

Hier tragen Sie die Länge des PDO's ein (wie viele bytes werden gesendet).

Feld "DATA"

Wenn das Feld "LEN" ungleich 0 ist, können Sie hier die zu sendenden Daten fest eintragen (jeweils 2 Ziffern ergeben 1 Byte: 05010206 ergibt D0=05, D1=01, D2=02, D3=06) oder über Merker variable Daten senden: @MB10 ergibt D0=MB10, D1=MB11 usw.

3.2.5.2 Empfangen von PDO's

Für den Empfang von PDO's brauchen Sie nur die entsprechenden internen Variablen anzulegen:

Variablenhandle für D0-D3 = COB-ID PDO
 Variablenhandle für D4-D7 = COB-ID PDO + 2000

Der Master legt die empfangenen PDO-Daten automatisch in diese Variablen ab, wenn er unter dem entsprechenden COB-ID Daten empfängt.

VORSICHT: Die Variablenhandles 384 bis 511 und 2384 bis 2511 sollten nicht anderweitig verwendet werden als für PDO-Empfang.

Steuern des Masters

Empfängt der Master auf der eigenen ID einen PDO (385 + Node-ID), so wertet er das PDO-Telegramm entsprechend der Beschreibung des freien CAN-Treibers aus.

3.2.6 Projekt-Download

Um mit dem ITE ein Projekt in das Bediengerät zu laden, muss der Sende-Identifizier unter "Geräte"/"Schnittstelle" auf 512 + Node-ID (zu finden unter "Geräte"/"Parametrieren", Registerkarte "CAN Einstellungen") eingestellt werden ("freies CAN" wählen). Dann "hört" das Bediengerät den Editor und beginnt mit dem Download.

3.2.7 Objektverzeichnis

Der Master hat zur Zeit kein Objektverzeichnis.

3.2.8 Zustandsübergänge

Der MASTER geht automatisch immer in den Zustand "Operational". Es gibt keine anderen Zustände.

3.3 Die SLAVE-Implementierung

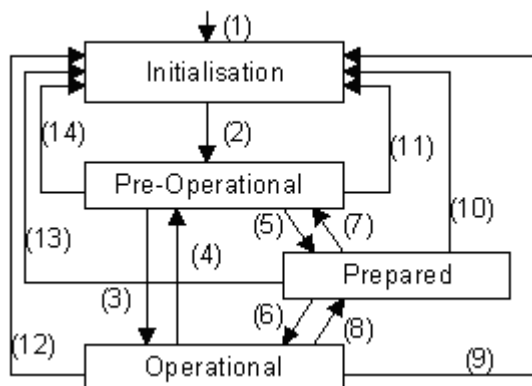
Die derzeitige SLAVE-Implementierung beschränkt sich auf die Emulation eines E/A-Teilnehmers. Dabei werden als Ausgangsdaten die Telegramme so ausgewertet, wie sie auch im freien CAN-Treiber definiert sind. Ebenso werden die Status-Telegramme einfach als Eingangs-PDO geliefert.

Hinzu kommt lediglich die Steuerung der Zustandsübergänge und das Objektverzeichnis.

In der Registerkarte "CAN Einstellungen" (unter "Geräte"/"Parametrieren"), Feld "ITS CAN Konfiguration" wird der Slave aktiviert.

3.3.1 Zustandsdiagramm

Der Slave verwendet das "Minimum-Bootup-Verhalten" nach der CANopen Draft Standard 301 Version 4.0:



Handbuch Bediengeräte

3.3.1.1 Zustandsübergangstabelle

(1)	Beim Einschalten wird der Zustand "Initialisierung" automatisch erreicht
(2)	Nach erfolgter Initialisierung wird automatisch der Zustand "Pre-Operational" erreicht und das Telegramm "Boot-Up Event" gesendet
(3), (6)	Anforderung "Start_Remote_Node"
(4), (7)	Anforderung "Enter_PRE-OPERATIONAL_State"
(5), (8)	Anforderung "Stop_Remote_Node"
(9),(10), (11)	Anforderung "Reset_Node"
(12),(13), (14)	Anforderung "Reset_Communication"

3.3.1.2 Beschreibung der Zustände

Folgende Bedingungen gelten in den einzelnen Zuständen:

Zustand Initialisation

Das Bediengerät initialisiert alle internen Puffer und Schnittstellen. Während dieses Zustands findet keine Kommunikation statt.

Zustand Pre-Operational

In diesem Zustand sind die NMT-Dienste sowie die SDO-Übertragung aktiv. PDO-Dienste werden nicht ausgeführt.

Ausnahme: da ein Download in jedem Zustand des Bediengeräts möglich sein muss, wird der PDO mit D0=0x10 (Request Memory read) trotzdem ausgeführt. Dieser PDO wird vom Editor dazu verwendet, den Download zu initiieren.

Spätere Versionen des Editors werden eine "Start_Remote_Node" Anforderung voranstellen, um den Zustand "Operational" zu erreichen, bevor das "Request Memory Read"-PDO gesendet wird.

Zustand Operational

In diesem Zustand sind alle Dienste aktiv.

Zustand Prepared

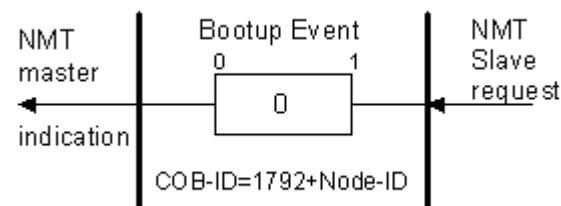
Wird der Zustand "Prepared" angefordert, so ist nur noch das Netzwerk-Management (NMT) aktiv. Weder PDO noch SDO Transfers sind möglich. Auch hier gilt die PDO-Ausnahme wie im Zustand "Pre-Operational"

3.3.1.3 Beschreibung des Telegrammverkehrs

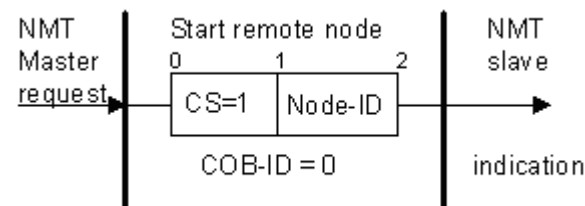
Folgende Telegramme sind für die Zustandswechsel verantwortlich. Das Bediengerät akzeptiert als Node-ID seine eigene oder die 0 (Broadcast = an

alle)

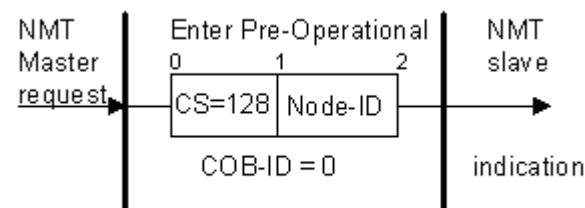
Boot-Up_Event



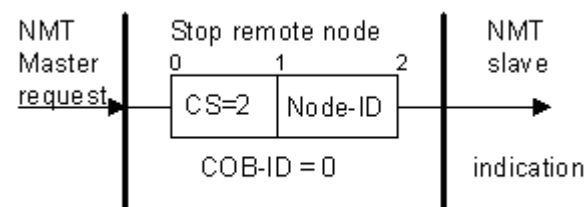
Start_Remote_Node



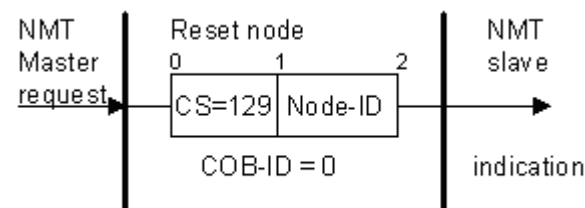
Enter_PRE-OPERATIONAL_State



Stop_Remote_Node

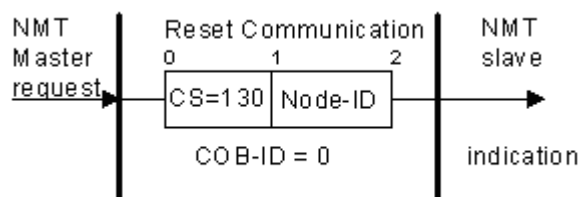


Reset_Node



Handbuch Bediengeräte

Reset_Communication



Wert	Bitcodiert gemäß DSP 301 Bit 0: Generic error (zwingend) Bit 1: Strom (nicht verwendet) Bit 2: Spannung (n.v.) Bit 3: Temperatur (n.v.) Bit 4: Kommunikationsfehler Bit 5: nicht verwendet Bit 6: nicht verwendet Bit 7: nicht verwendet
------	--

3.3.2 Das Objektverzeichnis

Das Objekt-Verzeichnis des Slave orientiert sich am DSP-401 für I/O-Module. Zur Zeit ist das Objektverzeichnis nur lesbar, nicht beschreibbar. Folgende Objekte sind definiert:

3.3.2.1 Objekt 1000h: Gerätetyp

Enthält Informationen über den Gerätetyp.

Objektbeschreibung:

INDEX	1000h
Name	Gerätetyp
Objektart	Einzelner Wert (VAR)
Datentyp	32 Bit ohne Vorzeichen (UNSIGNED32)
Kategorie	Zwingend

Objekteintrag

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Profilnummer und zusätzliche Informationen: D4/D5 = 401d = 191h D6/D7 = 3 (Ein-und Ausgänge)

3.3.2.2 Objekt 1001h: Fehlerregister

Enthält Informationen über aufgetretene Fehler

Objektbeschreibung:

INDEX	1001h
Name	Fehlerregister
Objektart	Einzelner Wert (VAR)
Datentyp	8-bit ohne Vorzeichen (UNSIGNED8)
Kategorie	Zwingend

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	Nein

3.3.2.3 Objekt 1004h: Anzahl der PDO's

Dieses Objekt enthält Informationen darüber, wie viele PDO's für das Bediengerät vorgesehen sind. Dieses Objekt entfällt ab DS 301 Version 4, ist aber aus Kompatibilitätsgründen noch enthalten.

Objektbeschreibung:

INDEX	1004h
Name	Anzahl der PDO's
Objektart	Feld (ARRAY)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32) D4/D5: Anzahl Transmit PDO's D6/D7: Anzahl Receive PDO's
Kategorie	Optional (V3.0)

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der unterstützten PDO's
Kategorie	Optional (V3.0)
Zugriff	Nur lesen
PDO Mapping	nein
Wert	D4/D5: 1 D6/D7: 1

Sub-Index	1
Beschreibung	Anzahl der synchronen PDO's
Kategorie	Optional (V3.0)
Zugriff	Nur lesen
PDO Mapping	nein
Wert	D4/D5: 0D6/D7: 0

Sub-Index	2
Beschreibung	Anzahl der asynchronen PDO's
Kategorie	Optional (V3.0)
Zugriff	Nur lesen
PDO Mapping	nein
Wert	D4/D5: 1 D6/D7: 1

3.3.2.4 Objekt 1008h: Hersteller-Geräte-

Handbuch Bediengeräte

name

In diesem Objekt ist ein 4-byte Kürzel für das Bediengerät zu finden.

Objektbeschreibung:

INDEX	1008h
Name	Hersteller-Gerätename
Objektart	Einzelner Wert (VAR)
Datentyp	String (4 byte)(Visible string)
Kategorie	Optional

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	'IT61' = ITS 6100-Serie 'IT62' = ITS 6200-Serie 'IT71' = ITS 7100-Serie 'IT72' = ITS 7200-Serie

3.3.2.5 Objekt 1009h: Hardware-Version

Dieses Objekt enthält die Nummer der im Gerät befindlichen Firmware. Beispiel: das BIOS IB055SE0 liefert ' 55'.

Objektbeschreibung:

INDEX	1009h
Name	Hardware-Version
Objektart	Einzelner Wert (VAR)
Datentyp	String (4 byte)(Visible string)
Kategorie	Optional

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Abhängig vom Gerätebios

3.3.2.6 Objekt 100Ah: Software-Version

Dieses Objekt enthält die Nummer des Betriebssystems (TOS), das im Gerät geladen ist. Beispiel: das TOS IO164A00 liefert ' 164'.

Objektbeschreibung:

INDEX	100Ah
Name	Software-Version
Objektart	Einzelner Wert (VAR)
Datentyp	String (4 byte)(visible string)

Kategorie	Optional
-----------	----------

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Abhängig vom TOS

3.3.2.7 Objekt 100Bh: Knotenadresse

Dieses Objekt enthält die parametrisierte Knotenadresse für das Bediengerät. Dieses Objekt entfällt ab DS 301 Version 4, ist aber aus Kompatibilitätsgründen noch enthalten.

Objektbeschreibung:

INDEX	100Bh
Name	Knotenadresse
Objektart	Einzelner Wert (VAR)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Kategorie	Optional

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Knotenadresse (Node-ID)

3.3.2.8 Objekt 100Ch: Überwachungszeit

Enthält die eingestellte Node-Guarding-Time (Überwachungszeit) des Moduls in Millisekunden.

Objektbeschreibung:

INDEX	100Ch
Name	Überwachungszeit (Guard Time)
Objektart	Einzelner Wert (VAR)
Datentyp	16-bit ohne Vorzeichen (UNSIGNED16)
Kategorie	Bedingt; zwingend, wenn Heartbeat-Protokoll nicht unterstützt wird.

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Eingestellte Überwachungszeit

3.3.2.9 Objekt 100Dh: Zeitfenster

Enthält den eingestellten Faktor für die Knotenüberwachung (Lebenszeit = Überwachungszeit *

Handbuch Bediengeräte

Faktor)

Objektbeschreibung

INDEX	100Dh
Name	Zeitfenster
Objektart	Einzelner Wert (VAR)
Datentyp	8-bit ohne Vorzeichen (UNSIGNED8)
Kategorie	Bedingt; zwingend, wenn Heartbeat-Protokoll nicht unterstützt wird.

Objekteintrag

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Eingestellter Faktor

3.3.2.10 Objekt 100Eh: Überwachungsidentifizier

Dieses Objekt enthält den Identifizier, über den die Knotenüberwachung durchgeführt wird. Dieses Objekt entfällt ab DS 301 Version 4, ist aber aus Kompatibilitätsgründen noch enthalten.

Objektbeschreibung

INDEX	100Eh
Name	Überwachungs-Identifizier
Objektart	Einzelner Wert (VAR)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Kategorie	Optional (V3.0)

Objekteintrag

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Knotenadresse + 1792

3.3.2.11 Objekt 100Fh: Anzahl der SDO's

Dieses Objekt enthält die Anzahl der SDO's, die das Bediengerät unterstützt. Dieses Objekt entfällt ab DS 301 Version 4, ist aber aus Kompatibilitätsgründen noch enthalten.

Objektbeschreibung:

INDEX	100Fh
Name	Anzahl der unterstützten SDO's
Objektart	Einzelner Wert (VAR)

Datentyp	32-bit ohne Vorzeichen (UNSIGNED32) D4/D5: Anzahl Server SDO's D6/D7: Anzahl Client SDO's
Kategorie	Optional (V3.0)

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	D4/D5: 1 Server SDO D6/D7: 0 Client SDO

3.3.2.12 Objekt 1010h: Parameter speichern

Das Bediengerät unterstützt keine Parameterspeicherung. Hier wird also als höchster Sub-Index die 0 angegeben, was keine Speicherung bedeutet.

Objektbeschreibung:

INDEX	1010h
Name	Parametersatz speichern
Objektart	Feld (ARRAY)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Kategorie	Optional

Objekteintrag:

Sub-Index	0
Beschreibung	Anzahl der unterstützten Parametersätze
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	0

3.3.2.13 Objekt 1011h: Parametersatz laden

Das Bediengerät unterstützt keine Parameterspeicherung. Hier wird also als höchster Sub-Index die 0 angegeben, was keine Speicherung bedeutet.

Objektbeschreibung:

INDEX	1011h
Name	Parametersatz laden
Objektart	Feld (ARRAY)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Kategorie	Optional

Objekteintrag

Sub-Index	0
-----------	---

Handbuch Bediengeräte

Beschreibung	Anzahl der unterstützten Parametersätze
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	0

3.3.2.14 Objekt 1014h: Identifier Emergency

Dieses Objekt enthält den Identifier, der für Emergency-Objekte verwendet wird.

Objektbeschreibung:

INDEX	1014h
Name	Emergency-Identifier
Objektart	Einzelner Wert (VAR)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Kategorie	Zwingend, wenn Emergency unterstützt wird.

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	Node-ID + 128

3.3.2.15 Objekt 1015h: Emergency Wartezeit

In diesem Objekt wird die Zeit gespeichert, die das Bediengerät zwischen zwei Emergency-Telegrammen mindestens warten muss. Dieser Eintrag muss beschreibbar sein. Da das Bediengerät derzeit kein Beschreiben des Objektverzeichnis zulässt, ist dieses Objekt noch nicht CANopen-konform. Dies wird in naher Zukunft nachgeholt. Das Objekt selbst existiert bereits.

Objektbeschreibung:

INDEX	1015h
Name	Emergency Wartezeit
Objektart	Einzelner Wert (VAR)
Datentyp	16-bit ohne Vorzeichen (UNSIGNED16)
Kategorie	Optional

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	nein
Wert	0

3.3.2.16 Obj. 1016h: Erwartete Herzschlagzeit

Mit dem Herzschlag (Heartbeat) können sich Geräte gegenseitig überwachen. Dieses Objekt enthält Informationen, welche Geräte vom Bediengerät überwacht werden. Z. Zt. Wird das Heartbeat nicht unterstützt, deshalb finden wir hier eine 0 im ersten Eintrag.

Objektbeschreibung:

INDEX	1016h
Name	Heartbeat-Time
Objektart	Feld (ARRAY)
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32) D4/D5: heartbeat time (UNSIGNED16) D6: Node-ID (UNSIGNED8)
Kategorie	Optional

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der Einträge
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	1

Sub-Index	1
Beschreibung	Heartbeat-Time Eintrag 1
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	0

3.3.2.17 Objekt 1017: Eigener Herzschlag

In diesem Objekt wird der "Herzschlag"-Rythmus (Heartbeat) des Geräts in ms eingetragen. Da das Heartbeat-Protokoll nicht unterstützt wird, finden wir eine 0 in diesem Eintrag.

Objektbeschreibung:

INDEX	1017h
Name	Eigener Herzschlag (Manufacturer Heartbeat)
Objektart	Einzelner Wert (VAR)
Datentyp	16-bit ohne Vorzeichen (UNSIGNED16)
Kategorie	Bedingt; zwingend, wenn Überwachung nicht unterstützt wird

Handbuch Bediengeräte

Objekteintrag:

Zugriff	Nur lesen
PDO Mapping	Nein
Wert	0

3.3.2.18 Objekt 1018h: Identitätsobjekt

Dieses Objekt enthält Daten, die einen CANopen-Teilnehmer eindeutig identifizieren. Hier werden Hersteller-Identifikation, Seriennummer usw. erzeugt. Im Eintrag 1 wird eine Herstellernummer eingetragen, die exklusiv vom CiA vergeben wird.

Objektbeschreibung:

INDEX	1018h
Name	Identifikationsobjekt
Objektart	Datensatz (RECORD)
Datentyp	Identität
Kategorie	Zwingend

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der Einträge
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	4

Sub-Index	1
Beschreibung	Hersteller-Kennzeichnung (Vendor ID)
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen
Wert	2Dh (Vendor ID GRAF-SYTECO)

Sub-Index	2
Beschreibung	Produktidentifikation
Kategorie	Optional
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Wert	Geräteabhängig: 6100d für ITS/AT 6100-Serie 6200d für ITS/AT 6200-Serie usw.

Sub-Index	3
Beschreibung	Revisionsnummer
Kategorie	optional
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Wert	1

Sub-Index	4
Beschreibung	Seriennummer
Kategorie	Optional
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen
Wert	z. Zt. Noch nicht unterstützt. Daten sind ungültig.

3.3.2.19 Objekt 1200h: Server SDO Parameter

In diesem Objekt finden sich die Identifier, die das Gerät benutzt, wenn SDO Daten abgefragt werden.

Objektbeschreibung:

INDEX	1200h
Name	Server SDO Parameter
Objektart	Datensatz (RECORD)
Datentyp	SDO Parameter
Kategorie	Bedingt

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der Einträge
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	2

Sub-Index	1
Beschreibung	COB-ID für SDO-Anforderung SDO Rx ID aus Gerätesicht
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	Nein
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Wert	1536 + NodeID

Sub-Index	2
-----------	---

Handbuch Bediengeräte

Beschreibung	COB-ID für SDO-Antwort SDO Tx ID aus Gerätesicht
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen
Wert	1792 + NodeID

3.3.2.20 Obj. 1400h: Empfangs-PDO-Param.

In diesem Objekt werden Parameter gespeichert, die den PDO-Empfang betreffen.

Objektbeschreibung:

INDEX	1400h
Name	Empfangs-PDO Parameter
Objektart	Datensatz (RECORD)
Datentyp	PDO CommPar
Kategorie	Zwingend für jeden unterstützten Empfangs-PDO

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl Einträge (Sub-Indizes)
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	2

Sub-Index	1
Beschreibung	Empfangs-ID (COB-ID)
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Wert	512 + Node-ID

Sub-Index	2
Beschreibung	Übertragungsart
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	8-bit ohne Vorzeichen (UNSIGNED8)
Wert	FEh (async PDO)

3.3.2.21 Obj. 1600h: Empfangs-PDO Mapping

Hier wird die Zuordnung der Empfangs-PDO-Da-

ten auf das Objektverzeichnis vorgenommen. Logischerweise sind hier die beiden Objekteinträge des Objekts 2000h zu finden.

Objektbeschreibung:

INDEX	1600h
Name	Empfangs-PDO Mapping
Objektart	Datensatz (RECORD)
Datentyp	PDO mapping
Kategorie	Zwingend für jeden unterstützten Empfangs-PDO

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der gemappten Objekte
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	Nein
Datentyp	8-bit ohne Vorzeichen
Wert	2

Sub-Index	1
Beschreibung	1. Gemapptes Objekt (Index 2000h Subidx. 1, 32 bit)
Kategorie	Bedingt; hier notwendig, da festes PDO-Mapping
Zugriff	Nur lesen
PDO Mapping	Nein
Datentyp	32 bit ohne Vorzeichen (UNSIGNED32)
Wert	20012000h

Sub-Index	2
Beschreibung	2. Gemapptes Objekt (Index 2000h Subidx. 2, 32 bit)
Kategorie	Bedingt; hier notwendig, da festes PDO-Mapping
Zugriff	Nur lesen
PDO Mapping	Nein
Datentyp	32 bit ohne Vorzeichen (UNSIGNED32)
Wert	20022000h

3.3.2.22 Objekt 1800h: Sende-PDO Parameter

In diesem Objekt werden Parameter gespeichert, die das PDO-Senden betreffen.

Objektbeschreibung:

INDEX	1800h
Name	Sende-PDO Parameter

Handbuch Bediengeräte

Objektart	Datensatz (RECORD)
Datentyp	PDO CommPar
Kategorie	Zwingend für jeden unterstützten Sende-PDO

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl Einträge (Sub-Indizes)
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Wert	2

Sub-Index	1
Beschreibung	Sende-ID (COB-ID)
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)
Wert	384 + Node-ID

Sub-Index	2
Beschreibung	Übertragungsart
Kategorie	Zwingend
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	8-bit ohne Vorzeichen (UNSIGNED8)
Wert	FEh (async PDO)

3.3.2.23 Objekt 1A00h: Sende-PDO Mapping

Hier wird die Zuordnung der Empfangs-PDO-Daten auf das Objektverzeichnis vorgenommen. Logischerweise sind hier die beiden Objekteinträge des Objekts 2001h zu finden.

Objektbeschreibung:

INDEX	1A00h
Name	Sende-PDO Mapping
Objektart	Datensatz (RECORD)
Datentyp	PDO mapping
Kategorie	Zwingend für jeden unterstützten Sende-PDO

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der gemappten Objekte
Kategorie	Zwingend
Zugriff	Nur lesen

PDO Mapping	Nein
Datentyp	8-bit ohne Vorzeichen
Wert	2

Sub-Index	1
Beschreibung	1. Gemapptes Objekt (Index 2001h Subidx. 1, 32 bit)
Kategorie	Bedingt; hier notwendig, da festes PDO-Mapping
Zugriff	Nur lesen
PDO Mapping	Nein
Datentyp	32 bit ohne Vorzeichen (UNSIGNED32)
Wert	20012001h

Sub-Index	2
Beschreibung	2. Gemapptes Objekt (Index 2001h Subidx. 2, 32 bit)
Kategorie	Bedingt; hier notwendig, da festes PDO-Mapping
Zugriff	Nur lesen
PDO Mapping	Nein
Datentyp	32 bit ohne Vorzeichen (UNSIGNED32)
Wert	20022001h

3.3.2.24 Objekt 2000h: Empfangsdaten

Dieses Objekt zählt zu den herstellerspezifischen Objekten. Das Bediengerät speichert in diesem Objekt die Daten des zuletzt empfangenen PDO's entsprechend dem im Objekt 1600h eingetragenen PDO-Mapping.

Objektbeschreibung

INDEX	2000h
Name	PDO-Empfangsdaten
Objektart	Datensatz (RECORD)

Objekteinträge:

Sub-Index	0
Beschreibung	Anzahl der Einträge
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	8 bit ohne Vorzeichen (UNSIGNED8)
Wert	2

Sub-Index	1
Beschreibung	Datenbytes D0 bis D3 des letzten empfangenen PDO's
Zugriff	Nur lesen

Handbuch Bediengeräte

PDO Mapping	ja
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)

Sub-Index	2
Beschreibung	Datenbytes D4 bis D7 des letzten empfangenen PDO's
Zugriff	Nur lesen
PDO Mapping	ja
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)

3.3.2.25 Objekt 2001h: Sendedaten

Dieses Objekt zählt zu den herstellerspezifischen Objekten. Das Bediengerät speichert in diesem Objekt die Daten des zuletzt gesendeten PDO's entsprechend dem im Objekt 1A00h eingetragenen PDO-Mapping.

Objektbeschreibung

INDEX	2001h
Name	PDO-Sendedaten
Objektart	Datensatz (RECORD)

Sub-Index	0
Beschreibung	Anzahl der Einträge
Zugriff	Nur lesen
PDO Mapping	nein
Datentyp	8 bit ohne Vorzeichen (UNSIGNED8)
Wert	2

Sub-Index	1
Beschreibung	Datenbytes D0 bis D3 des letzten gesendeten PDO's
Zugriff	Nur lesen
PDO Mapping	ja
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)

Sub-Index	2
Beschreibung	Datenbytes D4 bis D7 des letzten gesendeten PDO's
Zugriff	Nur lesen
PDO Mapping	ja
Datentyp	32-bit ohne Vorzeichen (UNSIGNED32)

4 SIMATIC S5-Treiber¹

Mit dem SIMATIC-S5 Treiber ausgestattet, kann ein Bediengerät direkt an die PG-Schnittstelle einer S5 angekoppelt werden. Diese Schnittstelle ist in jede S5-Steuerung eingebaut.

Die Datenübertragung von der S5 zum Bediengerät ist Sache des Treibers.

Es muß nur angegeben werden, wo die Melde-merker stehen, und welche Datenbausteine für Sollwerte usw. verwendet werden sollen.

4.1 Prinzipielle Funktion des Treibers

Das Gerät wird über die (eingebaute) 20mA/TTY-Schnittstelle direkt an die PG-Schnittstelle angeschlossen. Im ITE parametrieren Sie die Datenbausteine und Merkerbytes, über die Bild- und Meldeaufrufe sowie Variablenanzeigen realisiert werden sollen.

Der S5-Treiber simuliert jetzt ein Programmiergerät mit laufender Funktion "Status Variable" bzw. "Steuern Variable". Er fordert die Daten aus der SPS an und schreibt Daten (Sollwerte und Tasten) in die SPS zurück. Bediengeräteseitig ist außer der Parametrierung der Datenbereiche keine Arbeit erforderlich. Im S5-Programm müssen nur die im Treiber parametrierten Datenbausteine angelegt werden.

Zusätzlich funktioniert der Treiber als "Gateway" zwischen der CAN-Schnittstelle des Bediengeräts und der SPS. Das heißt, die SPS erhält indirekt Zugriff auf den CAN-Bus über das Bediengerät. Der S5-Treiber enthält folgende Funktionen:

- *Bildaufruf (auch PRIO-Bilder!) über Merker*
- *Meldeaufruf über Merker*
- *Istwertanzeige über Datenbausteine*
- *Sollwerteingabe in Datenbausteine*
- *Tastenabfrage über Merker*
- *LED ein- und ausschalten über Merker*
- *Gerätestatus beeinflussen über Datenbaustein*
- *Zugriff auf den CAN-Bus, der am Gerät angeschlossen sein kann (über Datenbaustein)*

4.2 Grundsätzliche Überlegungen

Zunächst muß geplant werden, wo die Daten für das Bediengerät in der SPS angelegt werden:

- *Bild- und Meldeaufrufe sowie Tasten- und LED-Funktionen werden über Merker abgewickelt. Der Treiber benötigt einen zusammenhängenden Bereich für alle diese Funktionen. Reservieren Sie also einen Block zusammengehöriger Merkerbytes*
- *Die Reihenfolge, wie die Funktionen auf Mer-*

kerbytes umgesetzt werden, ist immer gleich. Sie können angeben, wie viele Merkerbytes pro Funktion verwendet werden sollen.

- *Es wird immer in Merkerbyte-Schritten pro Funktion vorgegangen*
- *Die Reihenfolge ist immer wie folgt:*
 - *Merkerbytes für LED-Ansteuerung*
 - *Merkerbytes für Bildaufruf*
 - *Merkerbytes für Prioritätsbilder*
 - *Merkerbytes für Meldungen*
- *Es können maximal 32 Merkerbytes für Bild/Meldeaufrufe verwendet werden (Summe!). Genügt dies nicht, können weitere Aufrufe über die "Gateway"-Funktion vorgenommen werden.*
- *Maximal werden für diese Funktionen 40 Merkerbytes benötigt (32 für Aufruf, 8 für LEDs)*
- *Soll- und Istwerte (Variablen) werden über Datenbausteine ausgetauscht. Es ist möglich, für Sollwerte, Istwerte und Grenzen eigene Datenbausteine oder/und gemeinsam genutzte Datenbausteine zu parametrieren.*
- *Das sogenannte "Handle" (siehe Kapitel "Variablen") wird hier als Nummer des Datenworts verwendet. Wenn Sie also hier etwas von "Handle" lesen, so ist dies synonym zu "Datenwort-Nummer".*
- *Alle Variablen mit einer Länge von 1-16 Bit belegen automatisch ein ganzes Datenwort (siehe Kapitel "Variablen, Typen")*
- *Variablen mit einer Länge von 32 Bit belegen zwei aufeinanderfolgende Datenworte. Dies ist bei der Vergabe von Handles zu berücksichtigen. Beispiel: Hat eine Longword-Variablen das Handle 6, so belegt Sie automatisch die Datenworte DW6 und DW7 im Datenbaustein. Es sollte also keine Variable aus diesem Datenbaustein mit dem Handle 7 verwendet werden.*
- *Aufgrund von Treiber-Beschränkungen auf der PG-Schnittstelle dürfen innerhalb eines Bildes keine Variablen derselben Art (Sollwerte, Istwerte, Ober- Untergrenze) verwendet werden, deren Handles mehr als 64 Unterschied haben. Beispiel: Ist das kleinste Handle von Sollwerten im Bild 10, so darf das größte Handle von Sollwerten im Bild höchstens 74 sein. Ein Istwert darf aber jetzt z.B. wieder das Handle 90 haben, da er zu einem anderen Typ gehört. Innerhalb der Istwerte darf der "Handle-Unterschied" aber wieder nicht größer als 64 sein.*
- *Als Datenbausteine können beliebig DB2 bis DB255 gewählt werden. Die DB's müssen aber in der SPS angelegt werden. Fehlt ein parametrierter DB, so kommt keine Kommunikation zustande.*
- *Bilder und Meldungen müssen von 1 beginnend fortlaufend numeriert sein, wenn Bild/*

1. Die Begriffe SIMATIC und S5 sind eingetragene Warenzeichen der Siemens AG

Handbuch Bediengeräte

Meldeaufwurf über Merker erfolgt. Der Editor läßt zwar Lücken zu, bei Verwendung des S5-Treibers müssen Sie aber auf "lückenlose" Erstellung achten.

- **WICHTIG!!!** Alle Felder der Parametriermaske (siehe unten) müssen ausgefüllt werden. Dabei muß für jede Funktion ein Datenbaustein angegeben werden, auch wenn die Funktion nicht benutzt wird. Sonst kommt keine Kommunikation zustande.

In der Praxis zeigt sich, daß diese Regeln sehr einfach zu handhaben sind, da die Parametrierung des Treibers und der Variablen sehr komfortabel möglich ist.

4.3 Parametrierung des Treibers

Über das Menü "Gerät"/"Parametrieren", Registerkarte "Serielle Schnittstelle" erreichen Sie die Parametriermaske für den S5-Treiber. Klicken Sie den Knopf "Siemens S5 (AS511)" an. Dann erscheint die folgende Maske:

Datenbaustein(DB)/Merker(MB) für		Bilder	Meldungen
Istwerte:	0	8_ x8	16 x8
Status:	0	Prioritätsb.: 8_ x8	
Bild/Meld:	0	LEDs	Tasten
Tasten:	0	1 x8	1 x8

4.3.1 Feld "Istwerte"

Geben Sie hier an, in welchem Datenbaustein Sie die Istwerte für den Treiber hinterlegen wollen. Geben Sie hier z.B. DB5 ein, wenn Sie den DB5 für Istwerte verwenden wollen. Ein Istwert mit dem Handle 7 wird dann aus dem DB5, DW7 gelesen.

4.3.2 Feld "Sollwerte"

In dieses Feld geben Sie die Bezeichnung des Datenbausteins ein, in den der Treiber Sollwerte ablegen soll. Sollwerte werden immer dann in diesen Baustein abgelegt, wenn der Bediener am Gerät eine Sollwerteingabe mit Speichern vorgenommen hat.

Geben Sie die vollständige Bezeichnung des Datenbausteins ein, also z.B. DB8.

Sie können hier denselben Datenbaustein wie

auch bei den Istwerten angeben.

Das Handle der Variablen ist wieder die Nummer des Datenworts. Beispiel: Sie haben für Sollwerte DB8 gewählt. Dann wird der Sollwert mit dem Handle 20 im DB8 DW20 abgelegt.

4.3.3 Felder "Untergrenzen", "Obergrenzen"

Bei Sollwerteingaben können Unter- und Obergrenzen für die Eingabe festgelegt werden. Hier geben Sie an, aus welchen Datenbausteinen diese Grenzen geholt werden sollen, wenn die Grenzen nicht absolut als Wert, sondern als Variablen angelegt sind.

Geben Sie die vollständige Bezeichnung des Datenbausteins ein, also z.B. DB17.

Sie können hier denselben Datenbaustein wie auch bei den Soll- oder Istwerten angeben.

Das Handle der Variablen ist wieder die Nummer des Datenworts. Beispiel: Sie haben für Untergrenzen DB17 gewählt. Dann wird die Untergrenze-Variable mit dem Handle 12 aus dem DB17 DW12 gelesen.

4.3.4 Feld "Schrittweite"

Wenn Sie Sollwerte mit Schrittbearbeitung verwenden, können Sie die Schrittweite auch über eine Variable steuern. In diesem Feld können Sie jetzt einstellen, aus welchem DB Sie die Schrittweite lesen wollen.

Geben Sie die vollständige Bezeichnung des Datenbausteins ein, also z.B. DB2.

Sie können hier auch denselben Datenbaustein wie in einem anderen Feld angeben.

Das Handle der Schrittweite-Variablen ist wieder die Nummer des Datenworts. Beispiel: Sie haben für Schrittweite DB2 gewählt. Dann wird der Schrittweite mit dem Handle 0 aus dem DB2 DW0 gelesen.

4.3.5 Feld "Status"

Hier geben Sie an, in welchem Datenbaustein das Gerät seinen aktuellen Status ablegen soll. Der Treiber benötigt diesen Datenbaustein auf jeden Fall, er sollte auf jeden Fall nicht von Soll-, Ist- oder Grenzwerten belegt werden. Geben Sie hier also einen anderen DB an.

Dieser DB wird auch für die "Gateway"-Funktion verwendet; ebenso werden über diesen DB Bild/Meldeaufrufe abgewickelt, die sich nicht über den Merkerbereich erledigen lassen.

Der Aufbau des "Status-DB" wird später noch beschrieben.

4.3.6 Feld "Bild/Meld"

In diesem Feld parametrieren Sie das erste Merkerbyte, das für den Funktionsblock "LEDs/Bild-

Handbuch Bediengeräte

Meldeaufruf" verwendet werden soll.

Geben Sie die volle Bezeichnung ein, also z.B. MB30.

Dann beginnt der Merkerbereich, der für LED-Ansteuerung und Bild/Meldeaufruf verwendet wird, bei MB30.

Wieviele Merkerbytes jetzt benötigt werden, hängt von der Parametrierung der Felder "Anzahl der Bilder", "Anzahl der Meldungen" usw. ab.

4.3.7 Felder "Anzahl der Bilder, Meldungen, Prioritätsbilder, LEDs"

In diesen Feldern geben Sie jeweils an, wieviele Merkerbytes Sie für die einzelne Funktion reservieren wollen. Beachten Sie, daß Sie immer in 8-er Schritten arbeiten; Sie können also nicht 12 Merker für Bilder, dann 17 Merker für Meldungen und 11 Merker für Prioritätsbilder verwenden.

Wenn Sie eine Funktion nicht verwenden, geben Sie dort für die Anzahl 0 ein. Dann wird auch kein Merkerbyte für diese Funktion "verschwendet".

Ein Beispiel:

Sie verwenden das ITS6101. Sie wollen 25 Bilder und 25 Meldungen aufrufen können, ebenso sollen die 8 LEDs angesteuert werden. Es sollen 5 Bilder als Prioritätsbilder erscheinen können. Sie halten einen Bereich ab MB50 für das Gerät frei. Für 25 Bilder brauchen Sie 4 Merkerbytes ($4 \cdot 8 = 32$ Merker), ebenso für die Meldungen. Die Prioritätsbilder können sie über ein Merkerbyte steuern. Für die LEDs genügt ein Merkerbyte ($= 8$ Merker). Insgesamt werden 10 Merkerbytes ab MB50, also MB50-MB59, benötigt.

Geben Sie dazu folgende Werte in die Felder ein:

Feld	Eingabewert
Bild/Meld	MB50
Anzahl der Bilder	4
Anzahl der Meldungen	4
Anzahl der Prioritätsbilder	1
Anzahl der LEDs	1

Der Treiber liest jetzt MB50-MB59 (10 Merkerbytes) zyklisch aus und wertet die Merker einzeln als Aufrufe. Dabei ist die Zuordnung wie folgt

Merker	Led/Bild/Meldung
M50.0	LED 1
M50.1	LED 2
...	...
M50.7	LED 8
M51.0	Bild 0
M51.1	Bild 1
....
M51.7	Bild 7

M52.0	Bild 8
...	...
M54.7	Bild 31
M55.0	Bild 0, priorisiert
M55.1	Bild 1, priorisiert
...	...
M55.7	Bild 7, priorisiert
M56.0	Meldung 1
M56.1	Meldung 2
...	...
M56.7	Meldung 8
M57.0	Meldung 9
...	...
M59.7	Meldung 32

Um nun eine LED am Bediengerät einzuschalten wird einfach der zugehörige Merker im SPS-Programm gesetzt.

4.3.8 Feld "Tasten"

Hier muß angegeben werden, wo das Gerät seine Tasten in Merker spiegeln soll.

Es muß die volle Bezeichnung eingegeben werden z.B. MB100.

Dann beginnt der Merkerbereich, in den das Gerät seine Tasten spiegelt, bei MB100.

Wieviele Merkerbytes benötigt werden, hängt von der Parametrierung des Feldes "Anzahl der Tasten" ab.

4.3.9 Feld "Anzahl der Tasten"

Geben Sie an, wieviele Merkerbytes Sie für den Tastenstatus vom Bediengerät reservieren wollen. Es werden immer 8 Merker gleichzeitig reserviert. Denn für das Gerät werden auch nur Tastenerweiterungen in 8-er Schritten angeboten.

Beispiel:

Sie verwenden ein ITS6204 mit insgesamt 32 Tasten. Alle Tasten sollen in Merker gespiegelt werden. Es sind 4 Merkerbytes nötig. Als Merkerbereich wollen Sie MB60-MB63 verwenden. Geben Sie in der Parametrieremaske die folgenden Einträge ein:

Feld	Eingabe
Tasten	MB60
Anzahl der Tasten	4

Jetzt spiegelt der Treiber die Tasten in die Merker. Die Merker bedeuten jetzt:

Merker	= Taste Nr.
M60.0	1 (1. Reihe, links)
M60.1	2

Handbuch Bediengeräte

...	...
M60.7	8 (1. Reihe, rechts)
M61.0	9 (2. Reihe, links)
....
M63.7	32 (4. Reihe, rechts)

Numerikblock beim ITS/AT 61/67/71/77:

Wenn die numerischen Tasten ebenfalls abgefragt werden sollen, so müssen auf jeden Fall 8 Merkerbytes für die Tastenabfrage reserviert werden. Die Numeriktasten sind dann unter den Tastennummern nach folgender Tabelle zu finden:

Taste	Tastenr.
Escape	50
"4"	51
"6"	52
"2"	53
"8"	54
Enter	55
"0"	57
"1"	58
"3"	59
"5"	60
"7"	61
"9"	62
"."	63
"+/-"	64

Die Merker für die Tasten 49 und 56 werden immer mit 0 gesetzt, bitte verwenden Sie diese Merker nicht anderweitig.

Die Funktionstasten am ITS6101 haben die Tastennummern 1-8, Wird das ITS6106 verwendet (Maximalausbau), sind die Funktionstasten von 1-48 nummeriert. Deshalb werden die Numeriktasten ab Tastennummer 50 in die SPS gespiegelt.

4.4 Der Status-Datenbaustein

Über den Status-DB hält das Gerät die SPS über Bediener-Aktionen auf dem Laufenden. Es legt ab, welches Bild und welche Meldung gerade zur Anzeige kommen und in welchem Bedienzustand es sich befindet. Aber über diesen Status-DB werden auch weitere Funktionen des Bediengeräts angesprochen. Sie haben Zugriff auf den CAN-Bus. Außerdem können Sie Geräteparameter wie Kontrast und Helligkeit über den Status-DB beeinflussen.

Die in der folgenden Beschreibung gemachten Angaben von Datenwörtern beziehen sich immer auf den Datenbaustein, den Sie im Feld "Status" der Parametrieremaske eingetragen haben. Die folgende Tabelle gibt Aufschluß über die Verwen-

dung der Datenworte im Status-DB:

Datenworte	Funktion
DW0-DW9	Gerätestatus ITS
DW10-DW15	Sendepuffer für CAN-Gateway
DW16-DW21	Empfangspuffer für CAN-Gateway

4.4.1 Gerätestatus-Informationen

Das Gerät informiert die SPS standardmäßig über die folgenden Datenworte mit dem in der Tabelle aufgeführten Inhalt:

Datenwort	Inhalt des Datenworts
DW0	Bildnummer des aktuell angezeigten Bildes
DW1	Meldenummer der aktuell angezeigten Meldung (0=keine Meldung wird angezeigt)
DW2	Gerätezustand siehe "REPORT_STATUS (0x0A)" auf Seite 8
DW3	Anzahl der aktiven Bilder
DW4	Anzahl der aktiven Meldungen
DW5-DW9	nicht belegt, reserviert

4.4.2 CAN-Gateway Sendepuffer

Die Datenworte DW10-DW15 des Status-DB's sind wie folgt belegt:

Datenwort	Funktion
DW10	Handshake. KH=0000: Sendepuffer frei KH=FFFF: Sendepuffer belegt
DW11	CAN-Identifizier Hier muß das Anwenderprogramm den Adressaten eintragen. Adresse 0 ist das Bediengerät.
DW12	Telegramm-Art/CAN-Nutzdaten
DW13	Funktionswort 0/CAN-Nutzdaten
DW14	Funktionswort 1/CAN-Nutzdaten
DW15	Funktionswort 2/CAN-Nutzdaten

Handshake über DW10:

Über das DW10 erfolgt eine Abstimmung zwischen SPS-Anwenderprogramm und Treiber/CAN-Bus. Mit dem folgenden "FB-Gerüst" wird die Kommunikation abgewickelt:

```
:A  DB  ...  Status-DB wählen
:L  KH  0    Freikennung
:L  DW  10   prüfen
```

Handbuch Bediengeräte

```

:><F      Sender frei?
:BEB      Wenn nicht, Ende
:
:.....   Hier Sendedaten in
:.....   DW11-DW15
:.....   eintragen
:
:L  KH  FFFF  Sendekennung
:T  DW  10   eintragen
:BE
    
```

Hiermit läßt sich verhindern, daß die SPS zu schnell Daten an den CAN-Bus oder das Bediengerät selbst ausgibt.

Telegrammart und Funktionsworte:

Die Telegrammart und die Funktionsworte sind abhängig vom Adressaten in DW11:

DW11: KH=0000 Daten für das Bediengerät an der PG-Schnittstelle	DW11:KH=xxxx (Daten sind für den CAN-Bus bestimmt)
DW12 Telegramm-Art - Bildaufruf - Bild löschen - Melde-Aufruf - Meldung löschen - Parametrier-Kommandos	DW12 KH=aabb CAN-Nutzdaten: aa = byte0 bb = byte1
DW13 Funktionswort 1 (entsprechend der Telegramm-Art DW12 belegt, siehe nachfolgender Abschnitt)	DW13 KH=ccdd CAN-Nutzdaten: cc = byte2 dd = byte3
DW14 Funktionswort 2 (entsprechend der Telegramm-Art DW12 belegt, siehe nachfolgender Abschnitt)	DW14 KH=eeff CAN-Nutzdaten: ee = byte4 ff = byte5
DW15 Funktionswort 3 (entsprechend der Telegramm-Art DW12 belegt, siehe nachfolgender Abschnitt)	DW15 KH=gghh CAN-Nutzdaten: gg = byte6 hh = byte7

Sendungen an das Gerät, DW11 KH=0:

Über den Sendepuffer (Status-DB) lassen sich folgende Kommandos an das ITS/AT absetzen:

Telegramm-Art in DW12	Funktionsworte 1-3 DW13 - DW15
KF=+2 Variable senden (Set Value)	DW13: Handle der Variablen DW14: Variablenwert (lowword) DW15: Variablenwert (highword)
KF=+4 Meldeaufruf	DW13: Nummer der aufzurufenden Meldung, von KF=+1 bis KF=+9999 DW14/15: nicht benutzt

KF=+5 Meldung austragen	DW13: Nummer der auszutragenden Meldung, von KF=+1 bis KF=+9999 DW14/15: nicht benutzt
KF=+6 Bild aufrufen	DW13: Nummer des aufzurufenden Bildes, von KF=+1 bis KF=+9999 DW14/15: nicht benutzt
KF=+7 Bild austragen	DW13: Nummer des auszutragenden Bildes, von KF=+1 bis KF=+9999 DW14/15: nicht benutzt
KF=+8 Prioritäts-Bild aufrufen	DW13: Nummer des aufzurufenden Prioritätsbildes, von KF=+1 bis +9999 DW14/15: nicht benutzt
KH=15xx	Geräteparameter setzen.
KH=1500 Globale Softkey-Maske setzen	DW13: Bitmaske für Softkeys, KH=00xx bit 0 = nicht belegt bit 1 = Menütaste 1 bit 2 = Menütaste 2 ... bit 6 = Menütaste 6 bit 7 = nicht belegt Ist das Bit einer Taste gesetzt, so wird die Softkeyfunktion für diese Taste gesetzt und die Menüfunktion wird abgeschaltet. Ist das Bit 0, so wird die Menüfunktion der Taste aktiviert. DW14/15: nicht benutzt
KH=1501 Kontrast setzen	DW13: Kontrastwert 0-23 (KF=+0 bis KF=+23). 23 ist maximaler Kontrast DW14/15: nicht benutzt
KH=1502 Helligkeit der Hintergrundbeleuchtung	DW13: Helligkeitswert 0-7 (KF=+0 bis KF=+7). 7 ist maximale Helligkeit. DW14/15: nicht benutzt
KH=1503 Statuszeile ein/aus	DW13: Statuszeilenfunktion KF=+0 bis KF=+2 0: Statuszeile eingeblendet 1: Statuszeile ausgeblendet 2: wie im Bild definiert DW14/15: nicht benutzt
KH=1504 Position der Statuszeile	DW13: Zeilennummer der Statuszeile 0-7 (KF=+0 bis +7). 0 ist oberste Zeile. DW14/15: nicht benutzt
KH=1505 Rollierzeit der Meldungen	DW13: Rollierzeit in Sekunden von KF=+0 bis KF=+32 0 = "Rollieren aus" DW14/15: nicht benutzt
KH=1506 Rollierzeit der Bilder	DW13: Rollierzeit in Sekunden von KF=+0 bis KF=+32. 0 = "Rollieren aus" DW14/15: nicht benutzt
KH=1507 Tastenbelegung der Menütasten	DW13: KH=uuvv mit uu=Nummer der ESC-Taste vv=Nummer der „Pfeil links“-Taste DW14: KH=wwxx ww=Nummer der „Pfeil rechts“-Taste xx=Nummer der „Pfeil ab“-Taste DW15: KH=yyzz yy=Nummer der „Pfeil auf“-Taste zz=Nummer der Enter-Taste
KH=1508 Meldeausgang	DW13: KH=0000 ausschalten KH=0100 einschalten DW14/15: nicht belegt

Diese Daten sind in Anlehnung an das CAN-Datenformat formatiert siehe "Beschreibung der Te-

Handbuch Bediengeräte

legrammarten" auf Seite 4.

4.4.3 CAN-Gateway Empfangspuffer

Bevor Sie allzu euphorisch auf den CAN-Bus zugreifen wollen: bedenken Sie, daß auf dem CAN-Bus eine Übertragungsrate von bis zu 1 MBit/s eingestellt werden kann. Auf der PG-Schnittstelle sind 9600 Baud fest eingestellt, von denen man ca. 75% für das Protokoll rechnen muß. Netto bleiben also ca. 2400 baud übrig.

Wenn jetzt ein CAN-Modul nur 10-mal pro Sekunde ein Telegramm absetzt, so müßte 10 Mal pro Sekunde der Empfangspuffer in die SPS geschrieben werden, und nebenher auch noch die Merkerbytes und Variablen ausgelesen werden - unmöglich!

Also wozu das Ganze?

Denken Sie daran, daß z.B. eine Bedientastatur über den CAN-Bus an das Bediengerät angeschlossen werden kann. Irgendwie müssen Sie dann gemeldet bekommen, wenn eine Taste auf dieser Bedientastatur gedrückt wird - und das geht nur über das Bediengerät. Und ehrlich gesagt: so einfach können Sie keine anderen Tastaturen an die SPS anschließen wie über den CAN-Bus.

Die Bedientastatur sendet nun jedesmal eine CAN-Nachricht an das Bediengerät, wenn eine Taste gedrückt wird. Der Treiber legt dann dieses Telegramm im Empfangspuffer ab.

Realistisch gesehen wird ein Bediener höchstens 2-3 mal pro Sekunde eine Taste drücken - und das kann das Bediengerät gegebenenfalls auch noch zwischenspeichern. Es hat nämlich einen CAN-FIFO-Puffer mit 20 Telegrammen Tiefe.

Und wenn Sie dem Bediener über eine LED signalisieren, daß sein Tastendruck registriert wurde, so wird er nicht anfangen, wie wild auf der Tastatur zu hämmern.

Aber nun zur Beschreibung der Datenworte des Empfangspuffers. Auch der Empfangspuffer hält ein Handshake-Wort zur Verfügung, anhand dessen der Datentransfer kontrolliert wird, sowie die Informationsbytes:

Datenwort	Funktion
DW16	Handshake. KH=0000: Empfangspuffer leer KH=FFFF: Daten im Puffer
DW17	CAN-Identifizier Hier erhält das Anwenderprogramm in der SPS die Adresse des Senders. Adresse 0 ist das Bediengerät selbst.
DW18	Telegramm-Art/CAN-Nutzdaten
DW19	Funktionswort 0/CAN-Nutzdaten
DW20	Funktionswort 1/CAN-Nutzdaten
DW21	Funktionswort 2/CAN-Nutzdaten

Sendungen vom Gerät: DW17 KH=0000

Derzeit sind keine Telegramme vom Bediengerät an die SPS definiert. Alle Funktionen werden über Merker und Datenbausteine abgewickelt.

Sendung von CAN-Modul: DW17 KH=xxxx:

In diesem Fall wird das CAN-Telegramm des Senders 1:1 in den Empfangspuffer geschrieben. Dabei werden die Bytes wie folgt abgelegt:

Datenwort	Inhalt
DW17	CAN-Identifizier (Senderadresse) im Format nach 11.2.5
DW18	CAN-Nutzdaten KH=aabb aa = byte 0 bb = byte 1
DW19	CAN-Nutzdaten KH=ccdd cc = byte 2 dd = byte 3
DW20	CAN-Nutzdaten KH=eeff ee = byte 4 ff = byte 5
DW21	CAN-Nutzdaten KH=gghh gg = byte 6 hh = byte 7

Der Inhalt des CAN-Telegramms ist abhängig vom Gerät, das das Telegramm gesendet hat. Schlagen Sie deshalb im Handbuch zu diesem Gerät nach, wenn Sie den Inhalt der Telegramme bestimmen müssen.

4.4.4 CAN-Identifizier DW11 und DW17

Der CAN-Identifizier setzt sich aus insgesamt 16 bit zusammen. Die einzelnen Bits haben folgende Bedeutung:

15-5	4	3-0
Identifizier	R	DLC
x x x x x x x x x x x x	x	1 0 0 0

In den Bits 0-3, DLC (Data Length Code), wird angegeben, wie viele bytes Nutzdaten das CAN-Telegramm enthält. Dieser Wert kann 0 bis 8 sein. Maximal kann ein CAN-Telegramm 8 byte Nutzdaten enthalten.

das RTR-Bit (R) wird derzeit nicht verwendet. Setzen Sie es deshalb auf 0.

Die ID-Bits 0-10 müssen die Nummer des Geräts enthalten. Diese wird meistens über DIP-Schalter oder Jumper gesetzt. Details entnehmen Sie bitte dem Handbuch des jeweiligen Geräts.

4.4.5 Beispiele für das Gateway

Anhand der Zusammenschaltung einer Siemens

Handbuch Bediengeräte

S5, eines ITS 6101 und eines ITS 6303 wollen wir den Datentransfer über den Status-DB veranschaulichen. Wir gehen davon aus, daß das ITS 6101 und die S5 über die PG-Schnittstelle kommunizieren und das ITS 6303 über den CAN-Bus an das ITS 6101 angeschlossen ist.

Die Geräteadresse des ITS 6101 am CAN-Bus ist nicht von Belang; das ITS 6303 sei auf die Adresse 5 eingestellt.

Aufgabe ist es jetzt, festzustellen, wenn am ITS 6303 eine Taste gedrückt wurde, ebenso sollen LED's an der Tastatur gesetzt werden.

Wir wollen es so machen, daß die LED einer Taste so lange aufleuchten soll, wie die Taste gedrückt gehalten wird.

Kurz ein Ausflug in die Funktion des ITS 6303: es verhält sich so, daß Tastenbetätigungen automatisch auf den CAN-Bus ausgegeben werden. Dies erfolgt mittels des REPORT_KEY_DATA-Telegramms. Siehe "REPORT_KEY_DATA (0x17)" auf Seite 16. Mit diesem Telegramm wird jeweils die Nummer der Taste übermittelt und auch, ob die Taste losgelassen oder gedrückt wurde. Als zusätzliche Information erhält man bitweise den Status der ersten 4 Tastenreihen.

Mit dem Telegramm SET_LED setzen/rücksetzen wir dann die jeweilige LED. Siehe "SET_LED (0x16)" auf Seite 15

Wir müssen für diese Funktionen 4 Merkerbytes reservieren (hier merken wir uns den Status der Tasten und des letzten Tastendrucks). Außerdem brauchen wir den Status-DB.

Wir verwenden:

MB10 für Tasten 1-8
 MB11 für Tasten 9-16
 MB12 für Tasten 17-24
 MB13 Nummer der letzten Taste
 DB10 Status-Baustein

Das folgende SPS-Programm erledigt unsere Aufgabe. Es besteht aus 2 Funktionsbausteinen:

FB10 Auswertung der Tastendaten
 FB11 Senden der LED-Information

Zunächst wollen wir den CAN-Identifizier für das ITS/AT 6303 ermitteln. Das Gerät hat die Adresse 5 (ID = 5) und immer 8 byte Nutzdaten im Telegramm. Also DLC=8. Bitweise kodiert erhalten wir:

15-5	4	3-0
Identifizier	R	DLC
0 0 0 0 0 0 0 1 0 0 1 0	1 0 0 0	0
0	1	2
		8

Als CAN-Identifizier müssen wir also KH=0128 für das ITS/AT 6303 verwenden.

Auswertung der Tastendaten, FB10:

```

:A DB 10 Status-DB
:L KH FFFF Statusabfrage
:L DW 16 mit Handshake
:><F was da ?
:BEB nein.
:L DW 17 ja, von wem ?
:L KH 0128 vom ITS6303 ?
:><F testen
:SPB =M001 nicht von ihm
:L DW 19 Tastennummer
:SRW 8 auslesen
:T MB 13 und merken
:L DW 20 Tasten 1-16
:T MW 10 merken
:L DW 21 Tasten 17-24
:SRW 8 normieren
:T MB 12 und merken
M001 :L KH 0 mache den
:T DW 16 Empfangspuffer
:BE frei
Das wars.
    
```

Den zweiten Teil unserer Aufgabe, das Setzen der LED, machen wir mit folgendem FB11:

```

:L MB 13 Tastencode fragen
:L KF +0 ist einer da ?
:!=F mal sehen
:BEB nein, nichts.
:A DB 10 Status-DB
:L KH FFFF schauen, ob der
:L DW 10 Sendepuffer frei
:!=F wenn belegt
:BEB keine Aktion
:L KF +4 Annahme: setzen
:UN M 13.7 Tastenstatus
:SPB =M001 abfragen: gesetzt
:L KF +5 losgelassen: AUS
M001 :L KH 1600 Senden: LED
DATA
:OW und Funktion
:T DW 12 als Nutzdaten 0, 1
:R M 13.7 Tastencode
:L MB 13 laden, = LED Nr.
:SLW 8 Nutzdaten 3
:T DW 13 die Nummer
:L KH 0128 Identifizier für
    
```

Handbuch Bediengeräte

```
:T    DW    11    ITS6303 eintra-
      gen
:L    KH    FFFF  signalisiere: Sen-
      de-
:T    DW    10    puffer gefüllt
:L    KF    +0    jetzt noch
:T    MB    13    MB13 löschen:
      fertig
:BE
```

Jetzt müssen Sie diese beiden FB's noch zyklisch aus dem OB1 heraus aufrufen.

Sie sehen: es ist wirklich einfach, den Gateway zum CAN-Bus zu benutzen. Einfach Datenworte auffüllen und abfragen, das ist schon alles.

Nebenbei: Vergessen Sie nicht, bei Neustarts der SPS die Handshake-Felder DW10 und DW16 auf 0 zu setzen. Sonst laufen diese FB's nicht an. Ebenso müssen Sie das MB13 auf 0 setzen, um keine falsche LED zu zünden. Diese 5 Programmzeilen sollten Sie also in den OB21/OB22 stellen:

```
:A    DB    10    Status-DB
:L    KF    +0    Handshakes
:T    DW    10    initialisieren
:T    DW    16    für Anlauf
:T    MB    13    und Tastencode
      aus
```

In den Merkerbytes MB10 bis MB12 haben Sie mit diesen paar Programmzeilen den Status von 24 Tasten vom CAN-Bus geholt und die entsprechenden LEDs angesteuert. Toll, oder nicht?

Damit schließen wir die Beschreibung des Treibers für die S5.

5 MITSUBISHI FX

Um eine einfache Anschlußmöglichkeit für die MITSUBISHI FX - Serie zu schaffen, kann das Bediengerät mit einer RS422-Schnittstelle geliefert werden und dann direkt an die PG-Schnittstelle der FX gesteckt werden. Bitte bestellen Sie das Bediengerät mit der entsprechenden Schnittstelle (RS422), wenn Sie an die FX koppeln wollen. Das Bediengerät wird dann über einen Schnittstellenadapter (Best. Nr. M232A oder M232B) über die 20mA/TTY-Schnittstelle an den PC angeschlossen.

Die Datenübertragung über die Schnittstelle erledigt das Bediengerät. Sie brauchen nur die erforderlichen Merker und Variablendaten in der SPS abzulegen - der Zugriff erfolgt parametrisiert vom Bediengerät. Sie brauchen also keinen "Transferprogrammcode" in Ihrem SPS-Programm vorzusehen.

5.1 Prinzipielle Funktion des Treibers

Im Editor ITE werden die Datenbereiche parametrisiert, auf die das Bediengerät zugreifen soll. Sie müssen nicht jede einzelne Meldung bzw. Variable zuweisen. Es wird jeweils für die einzelnen Daten eine Art Anfangsadresse angegeben, und über die Handle-Nummer wird dann der Abstand ("Offset") zu dieser Anfangsadresse angegeben. Das Bediengerät sorgt jetzt dafür, daß ständig ein Datenaustausch mit den Meldemerkern und den aktuell im Bild angezeigten Soll-/Istwerten stattfindet. Dies geschieht ohne das Zutun der SPS. Der FX-Treiber erlaubt folgende Funktionen:

- *Bildaufruf (auch PRIO-Bilder!) über Merker*
- *Meldeaufruf über Merker*
- *Istwertanzeige aus Datenregistern*
- *Sollwerteingabe in Datenregister*
- *Tastenabfrage über Merker*
- *LED ein- und ausschalten über Merker*
- *Gerätestatus beeinflussen*
- *Zugriff auf den CAN-Bus, der am Bediengerät angeschlossen sein kann ("Gateway")*

5.2 Grundsätzliche Überlegungen

Sie müssen zunächst planen, wo Sie die Daten für das Bediengerät in der SPS anlegen. Beachten Sie dazu folgende Vorgaben:

- *Bild- und Meldeaufrufe sowie Tasten- und LED-Funktionen werden über Merker abgewickelt. Das Bediengerät benötigt einen zusammenhängenden Bereich für alle diese Funktionen. Reservieren Sie also einen Block zusammengehöriger Merkerbytes (8 Merker-Schritte)*
- *Bilder und Meldungen müssen von 1 begin-*

nend fortlaufend nummeriert sein, wenn Bild/Meldeaufruf über Merker erfolgt. Der Editor läßt zwar Lücken zu, bei Verwendung des FX-Treibers müssen Sie aber auf "lückenlose" Erstellung achten.

- *Die Reihenfolge, wie die Funktionen auf Merkerbytes umgesetzt werden, ist immer gleich. Sie können angeben, wie viele Merkerbytes pro Funktion verwendet werden sollen.*
- *Es wird immer in Merkerbyte-Schritten (8 Merker) pro Funktion vorgegangen*
- *Die Reihenfolge ist immer wie folgt:*
 - *Merker für LED-Ansteuerung*
 - *Merker für Bildaufruf*
 - *Merker für Prioritätsbilder*
 - *Merker für Meldungen*
- *Es können maximal 256 Merker für Bild/Meldeaufrufe verwendet werden (Summe!). Genügt dies nicht, können weitere Aufrufe über die "Gateway"-Funktion vorgenommen werden.*
- *Maximal werden für diese Funktionen 320 Merker benötigt (256 für Aufruf, 64 für LEDs)*
- *Soll- und Istwerte (Variablen) werden über Datenregister ausgetauscht. Es ist möglich, für Sollwerte, Istwerte und Grenzen einen gemeinsam genützten Datenregister-Bereich zu parametrieren.*
- *Das sogenannte "Handle" (siehe Kapitel "Variablen") wird zu dem parametrisierten Wert des Datenregisters addiert ("Offset"). Wenn Sie also hier etwas von "Handle" lesen, so ist dies synonym zu "Datenregister-Offset". Selbstverständlich können Sie die Anfangsadresse für alle Datentypen gleich einstellen. Am einfachsten ist es sowieso, wenn Sie für die Datenregister jeweils eine 0 einstellen (außer Status - dazu später mehr). Denn dann ist die Handle-Nummer, die Sie für Variablen einstellen, identisch mit der Datenregister-Nummer. So sehen Sie anhand der Handle-Nummer sofort, auf welches Datenregister Sie (bzw. das Bediengerät) zugreifen.*
- *Alle Variablen mit einer Länge von 1-16 Bit belegen automatisch ein ganzes Datenregister. Variablen mit einer Länge von 32 Bit belegen zwei aufeinanderfolgende Datenregister. Dies ist bei der Vergabe von Handles zu berücksichtigen. Beispiel: Hat eine Longword-Variablen das Handle 6, so belegt Sie automatisch zwei Datenregister (z.B. D6 und D7). Es sollte also keine Variable mit dem Handle 7 verwendet werden.*
- *Aufgrund von Treiber-Beschränkungen auf der PG-Schnittstelle dürfen innerhalb eines Bildes keine Variablen derselben Art (Sollwerte, Istwerte, Ober- Untergrenze) verwendet werden, deren Handles mehr als 32 Unterschied*

haben. Beispiel: Ist das kleinste Handle von Sollwerten im Bild 10, so darf das größte Handle von Sollwerten im Bild höchstens 42 sein. Ein Istwert darf aber jetzt z.B. wieder das Handle 90 haben, da er zu einem anderen Typ gehört. Innerhalb der Istwerte darf der "Handle-Unterschied" aber wieder nicht größer als 32 sein.

- Als Datenregister können beliebig D0 bis D7999 gewählt werden.
- **WICHTIG!!!** Alle Felder der Parametriermaske (siehe unten) müssen ausgefüllt werden. Dabei muß für jede Funktion ein Datenregister/Merker angegeben werden, auch wenn die Funktion nicht benutzt wird. Sonst kommt keine Kommunikation zustande.

In der Praxis zeigt sich, daß diese Regeln sehr einfach zu handhaben sind, da die Parametrierung des Treibers und der Variablen sehr komfortabel möglich ist.

Folgende Datenarten können/müssen Sie einzeln parametrieren:

- Istwerte (Datenregister D0 - D7999)
- Sollwerte (Datenregister D0 - D7999)
- Untergrenzen (Datenregister D0 - D7999)
- Obergrenzen (Datenregister D0 - D7999)
- Schrittwerte (Datenregister D0 - D7999)
- Statusinfo's (Datenregister D0 - D7999)
- Bild/Meldeaufruf, LED's (M0 - M1536)
- Tasten (M0 - M1536)

Bei den Merkern müssen Sie beachten, daß Sie die Anfangsadresse nur auf 8-er Schritte legen dürfen, sonst erhalten Sie beim Übertragen zum Bediengerät eine Fehlermeldung.

5.3 Parametrierung des Treibers

Über das Menü "Gerät"/"Parametrieren", Registerkarte "Serielle Schnittstelle" erreichen Sie die Parametriermaske für den FX-Treiber. Klicken Sie den Knopf "Mitsubishi FX" an. Dann erscheint die folgende Maske:

Grundeinstellungen Graf-ITE

CAN Einstellungen	CAN Module	Bilddaufbau	Schnelle Zähler
Allgemein	Gerät	Programme/Zeichensatz	
Meldungen und Bilder	Drucker	Statistik	Serielle Schnittstelle

SPS-Treiber (RS232)

Kein Treiber Frei (Telegrammverkehr) OK

Siemens S5 (AS511) VT100 Abbruch

Mitsubishi FX Anderer Treiber

Frei (Request/Response)

Bitte auch den Treiber unter "Programme" angeben !

Datenregister/Merker für

Istwerte:	0	Sollwerte:	0	Bilder	8_ x8	Meldungen	16_ x8
Status:	0	Untergrenzen:	0	Prioritätsb.:	8_ x8		
Bild/Meld:	0	Obergrenzen:	0	LEDs	1_ x8	Tasten	1_ x8
Tasten:	0	Schrittwerte:	0				

Geben Sie in die Felder "Bild/Meld" und "Tasten" jeweils den Anfang eines Merkerbereichs ein.

Achten Sie darauf, daß die Merkeradresse durch 8 teilbar ist; also M0, M8, M16, M24

In die Felder Istwerte, Sollwerte, Untergrenze, Obergrenze, Schrittwerte und Status müssen Sie ein Datenregister (D0 - D999) eingeben. Geben Sie auch den Buchstaben "D" mit ein, also z.B. "D100".

In die Felder "...x 8" müssen Sie die Anzahl der "Merkerbytes" eingeben. Das Bediengerät greift immer in Schritten von 8 Merkern zu.

5.3.1 Feld Istwerte

Geben Sie hier an, ab welchem Datenregister Sie die Istwerte für das Bediengerät hinterlegen wollen. Geben Sie hier z.B. D5 ein, wenn Sie das Datenregister D5 für Istwerte verwenden wollen. Ein Istwert mit dem Handle 7 wird dann aus dem Datenregister D12 gelesen (Basis D5 + Handle 7 --> D12).

5.3.2 Feld Sollwerte

In dieses Feld geben Sie die Nummer des Datenregisters ein, ab dem das Bediengerät Sollwerte ablegen soll. Sollwerte werden immer dann in diesen Registern abgelegt, wenn der Bediener am Bediengerät eine Sollwerteingabe mit Speichern vorgenommen hat.

Geben Sie die vollständige Bezeichnung des Datenregisters ein, also z.B. D8.

Sie können hier dasselbe Datenregister wie auch bei den Istwerten angeben.

Das Handle der Variablen ist wieder der Offset des Datenworts. Beispiel: Sie haben für Sollwerte D8 gewählt. Dann wird der Sollwert mit dem Handle 20 im Register D28 abgelegt (Basis D8 + Handle 20 --> D28).

Handbuch Bediengeräte

5.3.3 Felder Untergrenzen, Obergrenzen

Bei Sollwerteingaben können Unter- und Obergrenzen für die Eingabe festgelegt werden. Hier geben Sie an, aus welchen Datenregistern diese Grenzen geholt werden sollen, wenn die Grenzen nicht absolut als Wert, sondern als Variablen angelegt sind.

Geben Sie die vollständige Bezeichnung des Datenregisters ein, also z.B. D17.

Sie können hier dasselbe Datenregister wie auch bei den Soll- oder Istwerten angeben.

Das Handle der Variablen ist wieder der Offset des Datenregisters. Beispiel: Sie haben für Untergrenzen D17 gewählt. Dann wird die Untergrenze-Variablen mit dem Handle 12 aus dem Register D29 gelesen (Basis D17 + Handle 12 --> D29).

5.3.4 Feld Schrittwerte

Wenn Sie Sollwerte mit Schrittbearbeitung verwenden, können Sie die Schrittweite auch über eine Variable steuern. In diesem Feld können Sie jetzt einstellen, ab welchem Datenregister Sie die Schrittwerte lesen wollen.

Geben Sie die vollständige Bezeichnung des Datenregisters ein, also z.B. D2.

Sie können hier auch dasselbe Datenregister wie in einem anderen Feld angeben.

Das Handle der Schrittwert-Variablen ist wieder der Offset des Datenregisters. Beispiel: Sie haben für Schrittwerte D2 gewählt. Dann wird der Schrittwert mit dem Handle 0 aus dem Register D2 gelesen (Basis D2 + Handle 0 --> D2).

5.3.5 Feld Status

Hier geben Sie an, ab welchem Datenregister das Bediengerät seinen aktuellen Status ablegen soll. Das Bediengerät benötigt dieses Datenregister auf jeden Fall, es sollte auf jeden Fall nicht von Soll-, Ist- oder Grenzwerten belegt werden. Geben Sie hier also ein anderes Register an.

Dieser Registerbereich wird auch für die "Gateway"-Funktion verwendet; ebenso werden über diese Register Bild/Meldeaufrufe abgewickelt, die sich nicht über den Merkerbereich erledigen lassen. Aufbau des "Status-Registerbereichs" siehe unten.

5.3.6 Feld Bild/Meld

In diesem Feld parametrieren Sie den ersten Merker, der für den Funktionsblock "LEDs/Bild-Meldeaufruf" verwendet werden soll.

Geben Sie die volle Bezeichnung ein, also z.B. M32. Dann beginnt der Merkerbereich, der für LED-Ansteuerung und Bild/Meldeaufruf verwendet wird, bei M32.

Wieviele Merker jetzt benötigt werden, hängt von

der Parametrierung der Felder "...x 8" ab.

5.3.7 Felder Anzahl der Bilder, Meldungen, Prioritätsbilder, LEDs

In diesen Feldern geben Sie jeweils an, wieviele Merkerbytes Sie für die einzelne Funktion reservieren wollen. Beachten Sie, daß Sie immer in 8-er Schritten arbeiten; Sie können also nicht 12 Merker für Bilder, dann 17 Merker für Meldungen und 11 Merker für Prioritätsbilder verwenden.

Wenn Sie eine Funktion nicht verwenden, geben Sie dort für die Anzahl 0 ein. Dann wird auch kein Merkerbyte für diese Funktion "verschwendet".

Ein Beispiel:

Sie verwenden das ITS6101. Sie wollen 25 Bilder und 25 Meldungen aufrufen können, ebenso sollen die 8 LEDs angesteuert werden. Es sollen 5 Bilder als Prioritätsbilder erscheinen können. Sie halten einen Bereich ab M64 für das Bediengerät frei.

Für 25 Bilder brauchen Sie 4 Merkerbytes ($4 \cdot 8 = 32$ Merker), ebenso für die Meldungen.

Die Prioritätsbilder können sie über ein Merkerbyte steuern.

Für die LEDs genügt ein Merkerbyte (=8 Merker). Insgesamt werden 10 Merkerbytes ab M64, also M64-M143, benötigt.

Geben Sie dazu folgende Werte in die Felder ein:

Feld	Eingabewert
Bild/Meld	M64
Anzahl der Bilder	4
Anzahl der Meldungen	4
Anzahl der Prioritätsbilder	1
Anzahl der LEDs	1

Das Bediengerät liest jetzt M64-M143 (80 Merker) zyklisch aus und wertet die Merker einzeln als Aufrufe. Dabei ist die Zuordnung wie folgt:

Merker	Led/Bild/Meldung
M64	LED 1
M65	LED 2
...	...
M71	LED 8
M72	Bild 0
M73	Bild 1
....
M79	Bild 7
M80	Bild 8
...	...
M103	Bild 31
M104	Bild 0, priorisiert
M105	Bild 1, priorisiert

Handbuch Bediengeräte

...	...
M111	Bild 7, priorisiert
M112	Meldung 1
M113	Meldung 2
...	...
M119	Meldung 8
M120	Meldung 9
...	...
M143	Meldung 32

Um nun z.B. eine LED zu setzen, setzen Sie einfach den Merker in Ihrem SPS-Programm (so wie einen Ausgang) - und am Bediengerät leuchtet die LED auf. Einfacher gehts nicht.

5.3.8 Feld Tasten

Hier müssen Sie angeben, wo das Bediengerät seine Tasten in Merker spiegeln soll. Geben Sie die volle Bezeichnung ein, also z.B. M160 (durch 8 teilbar !!!). Dann beginnt der Merkerbereich, wo das Bediengerät seine Tasten spiegelt, bei M160. Wieviele Merker jetzt benötigt werden, hängt von der Parametrierung des Feldes "Anzahl der Tasten" ab.

5.3.9 Feld Anzahl der Tasten

Geben Sie an, wieviele Merkerbytes Sie für den Tastenstatus vom Bediengerät reservieren wollen. Es werden immer 8 Merker gleichzeitig reserviert. Denn im Bediengerät werden auch nur Tastenerweiterungen in 8-er Schritten angeboten.

Beispiel:

Sie verwenden ein ITS6204 mit insgesamt 32 Tasten. Sie wollen alle Tasten in der SPS als Merker gespiegelt haben. Sie müssen also 4 Merkerbytes reservieren. Als Merkerbereich wollen Sie M160-M191 verwenden. Geben Sie in der Parametrieremaske die folgenden Einträge ein:

Feld	Eingabe
Tasten	M160
Anzahl der Tasten	4

Jetzt spiegelt das Bediengerät die Tasten in die Merker. Die Merker bedeuten jetzt:

Merker	= Taste Nr.
M160	1 (1. Reihe, links)
M161	2
...	...
M167	8 (1. Reihe, rechts)

M168	9 (2. Reihe, links)
....
M191	32 (4. Reihe, rechts)

Numerikblock beim ITS/AT 6100:

Wenn die numerischen Tasten des ITS6100 ebenfalls abgefragt werden sollen, so müssen auf jeden Fall 8 Merkerbytes (=64 Merker) für die Tastenabfrage reserviert werden. Die ersten 48 Merker sind immer den Funktionstasten zugeordnet. Der Numerikblock wird ab dem 49. Merker übertragen.

Die Numeriktasten sind dann unter den Tastennummern nach folgender Tabelle zu finden (Beispiel: M0 als Basis):.

Taste	Tastennr.	für Basis M0
Escape	50	M49
"4	"51	M50
"6	"52	M51
"2	"53	M52
"8	"54	M53
Enter	55	M54
"0	"57	M56
"1	"58	M57
"3	"59	M58
"5	"60	M59
"7	"61	M60
"9	"62	M61
."	"63	M62
"+/-	"64	M63

Die Merker für die Tasten 49 (M48) und 56 (M55) werden immer mit 0 gesetzt, bitte verwenden Sie diese Merker nicht anderweitig.

Die Funktionstasten am ITS6101 haben die Tastennummern 1-8, Wird das ITS6106 verwendet (Maximalausbau), sind die Funktionstasten von 1-48 numeriert. Deshalb werden die Numeriktasten ab Tastennummer 50 in die SPS gespiegelt.

5.4 Die Status-Datenregister

Über die Status-Datenregister hält das Bediengerät die SPS über Bediener-Aktionen auf dem Laufenden. Es legt ab, welches Bild und welche Meldung gerade zur Anzeige kommen und in welchem Bedienzustand es sich befindet.

Aber über diesen Status-Bereich werden auch weitere Funktionen des Bediengerät angesprochen. Sie haben Zugriff auf den CAN-Bus, der am Bediengerät angeschlossen sein kann. Außerdem können Sie Geräteparameter wie Kontrast und Helligkeit über den Status-Bereich beeinflussen. Die in der folgenden Beschreibung gemachten Angaben von Datenregistern beziehen sich immer

Handbuch Bediengeräte

als Offset auf das Basisregister, das Sie im Feld "Status" der Parametriermaske eingetragen haben (Beispiel: Steht im Text D2 und Sie haben im Feld "Status" D10 angegeben, so ist das "echte" Register D12). Die folgende Tabelle gibt Aufschluß über die Verwendung der Datenworte im Status-Bereich:

Datenregister	Funktion
D0-D9 D10-D15	Gerätstatus des Bediengerät Sendepuffer für CAN-Gateway
D16-D21	Empfangspuffer für CAN-Gateway

5.4.1 Gerätstatus-Informationen

Das Bediengerät informiert die SPS standardmäßig über die folgenden Datenworte mit dem in der Tabelle aufgeführten Inhalt:

Datenregister	Inhalt des Datenregisters
D0	Bildnummer des aktuell angezeigten Bildes
D1	Meldenummer der aktuell angezeigten Meldung (0=keine Meldung wird angezeigt)
D2	Gerätzustand. Siehe "REPORT_STATUS (0x0A)" auf Seite 8
D3	Anzahl der aktiven Bilder
D4	Anzahl der aktiven Meldungen
D5-D9	nicht belegt, reserviert

5.4.2 CAN-Gateway Sendepuffer

Die Datenworte DW10-DW15 des Status-DB's sind wie folgt belegt:

Datenwort	Funktion
D10	Handshake. K0: Sendepuffer frei sonst Sendepuffer belegt
D11	CAN-Identifizier Hier muß das Anwenderprogramm den Adressaten eintragen. Adresse 0 ist das Bediengerät selbst.
D12	Telegramm-Art/CAN-Nutzdaten
D13	Funktionswort 0/CAN-Nutzdaten
D14	Funktionswort 1/CAN-Nutzdaten
D15	Funktionswort 2/CAN-Nutzdaten

Handshake über D10:

Über das D10 erfolgt eine Abstimmung zwischen SPS-Anwenderprogramm und Bediengerät/CAN-Bus. Mit dem folgenden "AWL-Gerüst" wird die Kommunikation abgewickelt:

LD M8000
CMP

K0
D10
M0

Ist jetzt Merker M1 gesetzt, dann ist der Sendepuffer frei und die Daten dürfen in den Sendepuffer geschrieben werden.

Hiermit läßt sich verhindern, daß die SPS zu schnell Daten an den CAN-Bus (oder das Bediengerät selbst) ausgibt.

Telegrammart und Funktionsworte:

Die Telegrammart und die Funktionsworte sind abhängig vom Adressaten in D11:

D11=K0 Daten für das Bediengerät an der PG-Schnittstelle	D11<>K0 Daten sind für den CAN-Bus bestimmt
D12 Telegramm-Art - Bildaufruf - Bild löschen - Melde-Aufruf - Meldung löschen - Parametrier-Kommandos	D12 KH=aabb CAN-Nutzdaten: aa = byte0 bb = byte1
D13 Funktionswort 1 (entsprechend der Telegramm-Art D12 belegt, siehe nachfolgender Abschnitt)	D13 KH=ccdd CAN-Nutzdaten: cc = byte2 dd = byte3
D14 Funktionswort 2 (entsprechend der Telegramm-Art D12 belegt, siehe nachfolgender Abschnitt)	D14 KH=eeff CAN-Nutzdaten: ee = byte4 ff = byte5
D15 Funktionswort 3 (entsprechend der Telegramm-Art D12 belegt, siehe nachfolgender Abschnitt)	D15 KH=gghh CAN-Nutzdaten: gg = byte6 hh = byte7

Sendungen an das Bediengerät, D11=K0:

Über den Sendepuffer (Status-Bereich) lassen sich folgende Kommandos an das Bediengerät absetzen:

Telegramm-Art in D12	Funktionsworte 1-3 D13 - D15
K2 Variable senden(Set Value)	D13: Handle der Variablen D14: Variablenwert (lowword) D15: Variablenwert (highword)
K4 Meldeaufruf	D13: Nummer der aufzurufenden Meldung, von K1 bis K9999 D14/15: nicht benutzt
K5 Meldung austragen	D13: Nummer der auszutragenden Meldung, von K1 bis K9999 D14/15: nicht benutzt
K6 Bild aufrufen	D13: Nummer des aufzurufenden Bildes, von K1 bis K9999 D14/15: nicht benutzt
K7 Bild austragen	D13: Nummer des auszutragenden Bildes, von K1 bis K9999 D14/15: nicht benutzt

Handbuch Bediengeräte

K8Prioritäts-Bild aufrufen	D13: Nummer des aufzurufenden Prioritätsbildes, von K1 bis K9999 D14/15: nicht benutzt
K5376 bis K5384	Geräteparameter setzen. Vergleiche "WRITE_PARAM (0x15)" auf Seite 13
K5376 Globale Softkey-Maske setzen	D13: Bitmaske für Softkeys, KH=00xx bit 0 = nicht belegt bit 1 = Menütaste 1 bit 2 = Menütaste 2bit 6 = Menütaste 6 bit 7 = nicht belegt Ist das Bit einer Taste gesetzt, so wird die Softkeyfunktion für diese Taste gesetzt und die Menüfunktion wird abgeschaltet. Ist das Bit 0, so wird die Menüfunktion der Taste aktiviert. D14/15: nicht benutzt
K5377 Kontrast setzen	D13: Kontrastwert 0-23 (K0 bis K23). 23 ist maximaler Kontrast D14/15: nicht benutzt
K5378 Helligkeit der Hintergrundbeleuchtung	D13: Helligkeitswert 0-7 (K0 bis K7). 7 ist maximale Helligkeit. D14/15: nicht benutzt
K5379 Statuszeile ein/aus	D13: Statuszeilenfunktion K0 bis K2 0: Statuszeile eingeblendet 1: Statuszeile ausgeblendet 2: wie im Bild definiert D14/15: nicht benutzt
K5380 Position der Statuszeile	D13: Zeilennummer der Statuszeile 0-7 (K0 bis K7). 0 ist oberste Zeile. D14/15: nicht benutzt
K5381 Rollierzeit der Meldungen	D13: Rollierzeit in Sekunden, von K0 bis K32. 0 = "Rollieren aus" D14/15: nicht benutzt
K5382 Rollierzeit der Bilder	D13: Rollierzeit in Sekunden, von K0 bis K32. 0 = "Rollieren aus" D14/15: nicht benutzt
K5383 Tastenbelegung der Menütasten	D13: KH=uvvv mit uu=Nummer der ESC-Taste vv=Nummer der "Pfeil links" -Taste D14: KH=wwxx mit ww=Nummer der "Pfeil rechts"-Taste xx=Nummer der "Pfeil ab" -Taste D15: KH=yyzz mit yy=Nummer der "Pfeil auf"-Taste zz=Nummer der Enter-Taste
K5384 Meldeausgang	D13: K0 ausschalten K256 einschalten D14/15: nicht belegt

Diese Daten sind in Anlehnung an das CAN-Datenformat, formatiert. Siehe "Beschreibung der Telegrammarten" auf Seite 4

5.4.3 CAN-Gateway Empfangspuffer

Bevor Sie allzu euphorisch auf den CAN-Bus zugreifen wollen: bedenken Sie, daß auf dem CAN-Bus eine Übertragungsrate von bis zu 1 MBit/s eingestellt werden kann. Auf der PG-Schnittstelle sind fest eingestellt 9600 Baud, von denen man ca. 75% für das Protokoll rechnen muß. Netto bleiben also ca. 2400 baud übrig.

Wenn jetzt ein CAN-Modul nur 10-mal pro Sekun-

de ein Telegramm absetzt, so müßte 10 Mal pro Sekunde der Empfangspuffer in die SPS geschrieben werden, und nebenher auch noch die Merkerbytes und Variablen ausgelesen werden - unmöglich!

Also wozu das Ganze?

Denken Sie daran, daß z.B. eine Bedientastatur ITS6300 über den CAN-Bus an das Bediengerät angeschlossen werden kann. Irgendwie müssen Sie dann gemeldet bekommen, wenn eine Taste auf dem ITS6300 gedrückt wird - und das geht nur über das Bediengerät. Und ehrlich gesagt: so einfach können Sie keine anderen Tastaturen an die SPS anschließen wie über den CAN-Bus.

Das ITS6300 sendet nun jedesmal eine CAN-Nachricht an das Bediengerät, wenn eine Taste gedrückt wird. Das Bediengerät legt dann dieses Telegramm im Empfangspuffer ab.

Realistisch gesehen wird ein Bediener höchstens 2-3 mal pro Sekunde eine Taste drücken. Das Bediengerät kann die Tastendrücke gegebenenfalls in einem FIFO-Puffer mit 20 Telegrammen Tiefe zwischenspeichern.

Wenn Sie dem Bediener über eine LED signalisieren, daß sein Tastendruck registriert wurde, so wird er nicht anfangen, wie wild auf der Tastatur zu hämmern.

Aber nun zur Beschreibung der Datenregister des Empfangspuffers. Auch der Empfangspuffer hält ein Handshake-Register zur Verfügung, anhand dessen der Datentransfer kontrolliert wird, sowie die Informationsbytes:

Datenregister	Funktion
D16	Handshake. K0: Empfangspuffer leer sonst Daten im Puffer
D17	CAN-Identifizier Hier erhält das Anwenderprogramm in der SPS die Adresse des Senders. Adresse 0 ist das Bediengerät selbst.
D18	Telegramm-Art/CAN-Nutzdaten
D19	Funktionswort 0/CAN-Nutzdaten
D20	Funktionswort 1/CAN-Nutzdaten
D21	Funktionswort 2/CAN-Nutzdaten

Sendungen vom Bediengerät: D17=K0

Derzeit sind keine Telegramme vom Bediengerät an die SPS definiert. Alle Funktionen werden über Merker und Datenbausteine abgewickelt.

Sendung von CAN-Modul: D17<->K0:

In diesem Fall wird das CAN-Telegramm des Senders 1:1 in den Empfangspuffer geschrieben. Dabei werden die Bytes wie folgt abgelegt:

Datenreg.	Inhalt
D17	CAN-Identifizier (Senderadresse)

Handbuch Bediengeräte

D18	CAN-Nutzdaten KH=aabb aa = byte 0 bb = byte 1
D19	CAN-Nutzdaten KH=ccdd cc = byte 2 dd = byte 3
D20	CAN-Nutzdaten KH=eeff ee = byte 4 ff = byte 5
D21	CAN-Nutzdaten KH=gghh gg = byte 6 hh = byte 7

Der Inhalt des CAN-Telegramms ist abhängig vom Gerät, das das Telegramm gesendet hat. Schlagen Sie deshalb im Handbuch zu diesem Gerät nach, wenn Sie den Inhalt der Telegramme bestimmen müssen.

CAN-Identifizier D11 und D17

Der CAN-Identifizier setzt sich aus insgesamt 16 bit zusammen. Die einzelnen Bits haben folgende Bedeutung:

15-5	4	3-0
Identifizier	R	DLC
x x x x x x x x x x x x	0	1 0 0 0

In den Bits 0-3, DLC (Data Length Code), wird angegeben, wie viele Bytes Nutzdaten das CAN-Telegramm enthält. Dieser Wert kann 0 bis 8 sein. Maximal kann ein CAN-Telegramm 8 Byte Nutzdaten enthalten.

das RTR-Bit (R) wird derzeit nicht verwendet. Setzen Sie es deshalb auf 0.

Die ID-Bits 0-10 müssen die Nummer des Geräts enthalten. Diese wird meistens über DIP-Schalter oder Jumper gesetzt. Details entnehmen Sie bitte dem Handbuch des jeweiligen Geräts.

5.5 Beispiele für das Gateway

Anhand der Zusammenschaltung einer MITSUBISHI FX, eines ITS6101 und eines ITS6303 wollen wir den Datentransfer über die Status-Register veranschaulichen. Wir gehen davon aus, daß das ITS6101 und die FX über die PG-Schnittstelle kommunizieren und das ITS6303 über den CAN-Bus an das ITS6101 angeschlossen ist.

Die Geräteadresse des ITS6101 am CAN-Bus ist nicht von Belang; das ITS6303 sei auf die Adresse 5 eingestellt.

Aufgabe ist es jetzt, festzustellen, wenn am ITS6303 eine Taste gedrückt wurde, ebenso sollen LED's an der Tastatur gesetzt werden.

Wir wollen es so machen, daß die LED einer Taste so lange aufleuchten soll, wie die Taste gedrückt gehalten wird.

Kurz ein Ausflug in die Funktion des ITS6303: es verhält sich so, daß Tastenbetätigungen automa-

tisch auf den CAN-Bus ausgegeben werden. Dies erfolgt mittels des REPORT_KEY_DATA Telegramms (siehe "REPORT_KEY_DATA (0x17)" auf Seite 16). Hier wird jeweils die Nummer der Taste übermittelt und auch, ob die Taste losgelassen oder gedrückt wurde. Als zusätzliche Information erhält man bitweise den Status der ersten 4 Tastenreihen.

Mit dem Telegramm SET_LED setzen/rücksetzen wir dann die jeweilige LED (siehe "SET_LED (0x16)" auf Seite 15).

Wir müssen für diese Funktionen 4 Merkerbytes reservieren (hier merken wir uns den Status der Tasten und des letzten Tastendrucks). Außerdem brauchen wir die Status-Register.

Wir verwenden:

M0-M7	für Vergleichsoperationen
M80-M87	für Tasten 1-8
M88-M95	für Tasten 9-16
M96-M103	für Tasten 17-24
M104-M119	temporäre Merker
D30	Nummer der letzten Taste
D0	Status-Baustein

Das folgende SPS-Programm erledigt unsere Aufgabe.

Zunächst wollen wir den CAN-Identifizier für das ITS6303 ermitteln. Das Gerät hat die Adresse 5 (ID = 5) und immer 8 byte Nutzdaten im Telegramm. Also DLC=8. Bitweise kodiert erhalten wir: Bit-Nr.

15-5	4	3-0
Identifizier	R	DLC
0 0 0 0 0 0 0 1 0 0 1	0	1 0 0 0
0	1	2
		8

Als CAN-Identifizier müssen wir demnach KH=0128 für das ITS6303 verwenden. Dies entspricht der Dezimalzahl K296.

Auswertung der Tastendaten:

LD	M8000	
CMP		Status-DB
	K0	Statusabfrage
	D16	mit Handshake
	M0	
LD	M1	was da ?
CMP		ja, von wem ?
	K296	vom ITS6303 ?
	D11	testen
	M0	

Handbuch Bediengeräte

LD	M1	vom ITS6303	K4M104	
MOV		Tasten 1-16	D13	in Nutzdaten 3
	D20	in die Merker laden	MOV	
	K4M80		K296	Identifizier ITS6303
MOV		Tasten 17-32	D11	eintragen
	D21		MOV	
	K4M96		K-1	Sendepuffer
MOV		Tastennummer	D10	absenden
	D19	zunächst	MOV	
	D30	speichern	K0	Tastencode
MOV		Empfangspuffer	D30	löschen
	K0	freimachen		
	D10	das wars.		

Einfach, oder nicht? Fast schon wie das Abfragen einer Eingabebaugruppe. Bedenken Sie, daß Sie auf diese Weise quasi zum Nulltarif 24 Tasten zusätzlich an die SPS angeschlossen haben und in den Merkern M80-M103 abfragen können. Den zweiten Teil unserer Aufgabe, das Setzen der LED, machen wir mit folgendem Programm:

```

LD M8000
CMP D30      Tastencode fragen
              ist einer da ?
K0           mal sehen
M0
LDI M1       wenn nicht 0: Taste
CMP K0       Sendepuffer frei ?
D10
M4           Ja --> M5 gesetzt
LD M8000    Tastencode in Merker
MOV D30
K4M104      104-119
LD M5       Sendepuffer frei ?
ANI M119    und gedrückt ?
MOV K5636   KH 1604: SET LED
D12         als Kommando
LD M5       Sendepuffer frei ?
AND M119    und gedrückt ?
MOV K5637   KH 1605: RESET LED
D12         als Kommando
LDI M8000   jetzt noch Tastennr.
OUT M119    ohne Bit 7
LD M5       Sendepuffer frei ?
MOV         ja, Tastencode=LED
    
```

Sie sehen: es ist wirklich einfach, den Gateway zum CAN-Bus zu benutzen. Einfach Datenregister auffüllen und abfragen, das ist schon alles. Nebenbei: Vergessen Sie nicht, bei Neustarts der SPS die Handshake-Register D10 und D16 auf 0 zu setzen. Sonst laufen diese Funktionen nicht an. Ebenso müssen Sie das D30 auf 0 setzen, um keine falsche LED zu zünden. Die folgenden Programmzeilen sollten Sie also als Initialisierungsprogramm verwenden:

```

LD M8002    Anlaufmerker
MOV K0
D10         Sendepuffer frei
MOV K0
D16         Empfangspuffer frei
MOV K0
D30         Keine Taste da
    
```

In den Merkern M80 bis M103 haben Sie mit diesen paar Programmzeilen den Status von 24 Tasten vom CAN-Bus geholt und die entsprechenden LEDs angesteuert. Toll, oder nicht?

Damit schließen wir die Beschreibung des Treibers für die MITSUBISHI FX.

Handbuch Bediengeräte

6 Request/Response Treiber

Um eine einfache Anschlußmöglichkeit für beliebige Steuerungen zu schaffen, kann das Bediengerät über eine RS232-Schnittstelle (optional RS422) angesteuert werden.

Die Datenübertragung über die Schnittstelle erledigt das Bediengerät. Sie müssen lediglich die erforderlichen Basisadressen für Variablen, Bild- und Meldeaufrufe, LEDs und Tasten im Editor ITE konfigurieren - der Zugriff erfolgt parametrisiert vom Bediengerät auf die von Ihnen eingestellte Basisadresse. Sie müssen also nur auf Anfragen des Bediengerätes reagieren, was den Programmieraufwand wesentlich vereinfacht.

6.1 Schnittstellenbeschreibung:

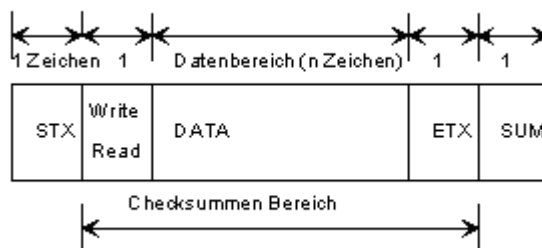
Schnittstelle: RS232C Standard
 Protokoll: nachfolgend beschrieben
 Baudrate: 300 - 9600 Baud
 Parity: no, even, odd
 Stopbits: 1,2
 Datenbits: 8

6.1.1 Schnittstellenkommandos:

Zeichen	Hex Code	Beschreibung
ENQ	05 hex	Enquiry: Anfrage des Bediengeräts
ACK	06 hex	Acknowledge: Antwort auf ein ENQ
NACK	15 hex	Negative Ack: Antwort bei einem Übertragungsfehler
STX	02 hex	Start of Text: Startbyte eines Telegramms
ETX	03 hex	End of Text: Endkennung eines Telegramms
WRITE	31 hex	Bediengerät sendet Daten an die Steuerung
READ	30 hex	Bediengerät fordert Daten aus der Steuerung an

Übertragungsformat:

ENQ, ACK und NACK werden als einzelne Kommandos gesendet. Alle anderen Kommandos (STX, ETX, Write, Read) werden innerhalb eines Übertragungsrahmens gesendet:



Die Checksumme wird als einzelnes Zeichen (byte) nach dem ETX-Code gesendet.

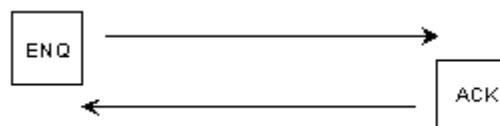
Beispiel:

STX	CMD	high	low	DLN	ETX	Checksum
	Read	TOP-ADDRESS				
02h	30h	10h	00h	04h	03h	47h
	30h + 10h + 00h + 04h + 03h = 47h					

Alle Hex-Bytes ab dem Byte CMD bis ETX werden aufaddiert und ergeben die Checksumme. DLN: DataLengthCode (Anzahl Bytes die angefordert werden)

6.1.1.1 Kommunikations-Telegramme:

Initialisierungsscheck
 Bediengerät <> Steuerung



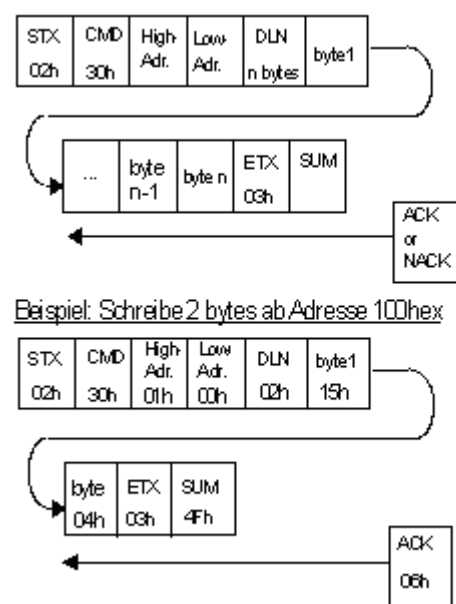
max. Delay = 500ms zwischen Request und Response

Handbuch Bediengeräte

6.1.1.2 Read-Telegramm



6.1.1.3 Write-Telegramm



6.2 Prinzipielle Funktion des Treibers

Das Kommunikationsmodell des "Request/Response"-Treibers ist dem Datenzugriff und der Speicherorganisation einer SPS nachgebildet. Das Bediengerät liest/schreibt aus/in Daten- und Merkerbereiche der Steuerung. Adresse und Größe der zu übertragenden Datenblöcke sind in den "Read/Write"-Telegrammen enthalten. Die Steuerung verhält sich passiv und beantwortet lediglich die "Anfragen" (Read/Write) des Bediengeräts. Dadurch wird der Programmieraufwand für den Kommunikationstreiber auf der Steuerungsseite sehr vereinfacht.

Im Editor ITE werden die Datenbereiche parametrisiert, auf die das Bediengerät zugreifen soll. Sie müssen nicht jede einzelne Meldung bzw. Variable zuweisen. Es wird jeweils für die einzelnen Da-

ten eine Art Anfangsadresse angegeben, und über die Handle-Nummer wird dann der Abstand ("Offset") zu dieser Anfangsadresse angegeben.

Das Bediengerät sorgt jetzt dafür, daß ständig ein Datenaustausch mit den Meldemerkern und den aktuell im Bild angezeigten Soll-/Istwerten stattfindet. Dies geschieht ohne das Zutun der Steuerung.

Der "freie Request/Response"-Treiber erlaubt folgende Funktionen:

- *Bildaufwurf (auch PRIO-Bilder!) über Merker*
- *Meldeaufwurf über Merker*
- *Istwertanzeige aus Datenregistern*
- *Sollwerteingabe in Datenregister*
- *Tastenabfrage über Merker*
- *LED ein- und ausschalten über Merker*
- *Gerätestatus beeinflussen*
- *Zugriff auf den CAN-Bus, der am Bediengerät angeschlossen sein kann ("Gateway")*

6.2.1 Namensvereinbarungen

In der nachfolgenden Beschreibung zur Konfiguration des Treibers werden Begriffe aus der SPS-Umgebung, wie Merker, Merkerbyte, Datenregister usw. verwendet. Dabei gilt folgende Vereinbarung:

Merker	M	Bit 0..7 innerhalb eines Merkerbytes	Bit-Position innerhalb eines Datenbytes, adressierbar über Adresse im Read/Write-Telegramm
Merkerbyte	MB	1 Byte	adressierbar über Adresse im Read/Write-Telegramm
Datenwort	D	2 Bytes	adressierbar über Adresse im Read/Write-Telegramm

Beispiele:			
Basisadresse = A102			
Merker M102.5 = bit 5, innerhalb des Datenbytes unter der Adresse 102hex			
Datenwort D102 = 16bit (2bytes) Daten unter der Adresse 102hex (belegt Adresse 102hex und 103hex) --> das nächste adressierbare Datenwort ist D104 (Adresse 104hex).			
Merkerbyte M102 = 1byte Datum unter der Adresse 102hex			

6.2.2 Grundsätzliche Überlegungen

Sie müssen zunächst planen, wo Sie die Daten für das Bediengerät in der Steuerung anlegen. Beachten Sie dazu folgende Vorgaben:

- *Bild- und Meldeaufrufe sowie Tasten- und LED-Funktionen werden über Merker (bitorien-*

- tiert) abgewickelt. Das Bediengerät benötigt einen zusammenhängenden Bereich für alle diese Funktionen. Reservieren Sie also einen Block zusammengehöriger Merkerbytes.
- Bilder und Meldungen sollten von 0 beginnend fortlaufend numeriert sein. Die Bild/Meldenummer ist dem entsprechenden "Bit", ausgehend von der im Editor ITE eingestellten Basisadresse, zugeordnet.
Beispiel: BildNr=9, eingestellte Basisadr = 100hex --> bit1 der Adr. 101hex ist dem Bild 9 zugeordnet.
 - Die Reihenfolge, wie die Funktionen auf Merkerbytes umgesetzt werden, ist immer gleich. Sie können angeben, wie viele Merkerbytes pro Funktion verwendet werden sollen.
 - Es wird immer in Merkerbyte-Schritten (8 Merker) pro Funktion vorgegangen
 - Die Reihenfolge ist immer wie folgt:
 - Merker für LED-Ansteuerung
 - Merker für Bildaufruf
 - Merker für Prioritätsbilder
 - Merker für Meldungen
 - Es können maximal 256 Merker für Bild/Meldeaufrufe verwendet werden (Summe!). Genügt dies nicht, können weitere Aufrufe über die "Gateway"-Funktion vorgenommen werden.
 - Maximal werden für diese Funktionen 320 Merker benötigt (256 für Aufruf, 64 für LEDs)
 - Soll- und Istwerte (Variablen) werden über Datenregister ausgetauscht. Es ist möglich, für Sollwerte, Istwerte und Grenzen einen gemeinsam genutzten Datenregister-Bereich zu parametrieren.
 - Das Handle wird zu dem parametrierten Wert des Datenregisters addiert ("Offset"). Wenn Sie also hier etwas von Handle lesen, so ist dies synonym zu "Datenregister-Offset". Selbstverständlich können Sie die Anfangsadresse für alle Datentypen gleich einstellen. Am einfachsten ist es sowieso, wenn Sie für die Datenregister jeweils eine 0 einstellen (außer Status - dazu später mehr). Denn dann ist die Handle-Nummer, die Sie für Variablen einstellen, identisch mit der Datenregister-Nummer. So sehen Sie anhand der Handle-Nummer sofort, auf welches Datenregister Sie (bzw. das Bediengerät) zugreifen.
 - Alle Variablen mit einer Länge von 1-16 Bit belegen automatisch ein ganzes Datenregister
 - Variablen mit einer Länge von 32 Bit belegen zwei aufeinanderfolgende Datenregister. Dies ist bei der Vergabe von Handles zu berücksichtigen. Beispiel: Hat eine Longword-Variable das Handle 6, so belegt Sie automatisch zwei Datenregister (z.B. D6 und D7). Es sollte also keine Variable mit dem Handle 7 verwendet werden.
 - Aufgrund von Treiber-Beschränkungen auf der seriellen Schnittstelle dürfen innerhalb eines Bildes keine Variablen derselben Art (Sollwerte, Istwerte, Ober- Untergrenze) verwendet werden, deren Handles mehr als 64 Unterschied haben. Beispiel: Ist das kleinste Handle von Sollwerten im Bild 10, so darf das größte Handle von Sollwerten im Bild höchstens 74 sein. Ein Istwert darf aber jetzt z.B. wieder das Handle 90 haben, da er zu einem anderen Typ gehört. Innerhalb der Istwerte darf der "Handle-Unterschied" aber wieder nicht größer als 64 sein.
 - Als Datenregister können beliebig D0 bis D999 gewählt werden.
 - **WICHTIG!!!** Alle Felder der Parametrieremaske (siehe unten) müssen ausgefüllt werden. Dabei muß für jede Funktion ein Datenregister/Merker angegeben werden, auch wenn die Funktion nicht benutzt wird. Sonst kommt keine Kommunikation zustande.
 - In der Praxis zeigt sich, daß diese Regeln sehr einfach zu handhaben sind, da die Parametrierung des Treibers und der Variablen sehr komfortabel möglich ist.
 - Folgende Bereiche (16 bit HEX-Adressen) können/müssen Sie einzeln parametrieren:
 - Istwerte (Adresse A0 - AFFFF)
 - Sollwerte (Datenregister A0 - AFFFF)
 - Untergrenzen (Datenregister A0 - AFFFF)
 - Obergrenzen (Datenregister A0 - AFFFF)
 - Schrittwerte (Datenregister A0 - AFFFF)
 - Statusinfo's (Datenregister A0 - AFFFF)
 - Bild/Meldeaufruf, LED's (A0 - AFFFF, Merkerorientiert)
 - Tasten (A0 - AFFFF, Merkerorientiert)

6.3 Parametrierung des Treibers

Über das Menü "Gerät"/"Parametrieren", Registerkarte "Serielle Schnittstelle" erreichen Sie die Parametrieremaske für den "Request/Response"-Treiber. Klicken Sie den Knopf "Request/Response" an. Dann erscheint die folgende Maske:

Geben Sie in die Felder "Bild/Meld" und "Tasten" jeweils den Anfang eines Merkerbereichs ein, z.B. A1F0 für die Adresse 1F0hex.

In die Felder Istwerte, Sollwerte, Untergrenze, Obergrenze, Schrittwerte und Status müssen Sie ebenfalls eine Basisadresse im HEX-Format (A0 - AFFFF) eingeben. Geben Sie auch den Buchstaben "A" mit ein, also z.B. "A5DC0" für die Adresse 5DC0hex.

In die Felder "...x 8" müssen Sie die Anzahl der "Merkerbytes" eingeben. Das Bediengerät greift immer in Schritten von 8 Merkern zu.

6.3.1 Feld Istwerte

Geben Sie hier an, ab welcher Adresse Sie die Istwerte für das Bediengerät hinterlegen wollen. Geben Sie hier z.B. A6 ein, wenn Sie die Adresse 0006hex für Istwerte verwenden wollen. Ein Istwert mit dem Handle 7 wird dann mit der Adresse 0014hex angefordert (Basisadr. = 0006hex + Handle 7 * 2bytes [pro Variablen-Handle sind 2bytes reserviert] --> 0014hex).

6.3.2 Feld Sollwerte

In dieses Feld geben Sie die HEX-Adresse ein, ab dem das Bediengerät Sollwerte ablegen soll. Sollwerte werden immer dann unter dieser Adresse abgelegt, wenn der Bediener am Bediengerät eine Sollwerteingabe mit Speichern vorgenommen hat. Geben Sie die vollständige Bezeichnung der Basisadresse ein, also z.B. A100.

Sie können hier dieselbe Basisadresse wie auch bei den Istwerten angeben.

Das Handle der Variablen ist wieder der Offset auf die eingestellte Adresse (2 Bytes Offset pro Variablen-Handlenummer).

Beispiel:

Sie haben für Sollwerte A100 gewählt. Dann wird der Sollwert mit dem Handle 20 unter der Adresse A128hex abgelegt (Basis A100 + Handle 20 [2bytes pro Handle] --> 128hex).

6.3.3 Felder Untergrenzen, Obergrenzen und Schrittwerte

Bei Sollwerteingaben können Unter- und Obergrenzen für die Eingabe festgelegt werden. Hier geben Sie an, ab welchen Adressen diese Grenzen geholt werden sollen, wenn die Grenzen nicht absolut als Wert, sondern als Variablen angelegt sind.

Geben Sie die vollständige Bezeichnung der HEX-Adresse ein, also z.B. A1017.

Sie können hier dieselbe Adresse wie auch bei den Soll- oder Istwerten angeben.

Das Handle der Variablen ist wieder der Offset der Basisadresse. Beispiel: Sie haben für Untergrenzen A1017 gewählt. Dann wird die Untergrenze-Variable mit dem Handle 12 mit der Adresse 1028h angefordert (Basis 1017hex + Handle 12 [2bytes pro Handle] --> 1028hex).

6.3.4 Feld Status

Hier geben Sie an, ab welcher Adresse das Bediengerät seinen aktuellen Status ablegen soll. Das Bediengerät benötigt diesen Adressbereich auf jeden Fall, er sollte daher nicht von Soll-, Ist- oder Grenzwerten belegt werden. Geben Sie hier also eine andere Basis-Adresse an.

Dieser Adressbereich wird auch für die "Gateway"-Funktion verwendet; ebenso werden über diese Adresse Bild/Meldeaufrufe abgewickelt, die sich nicht über den Merkerbereich erledigen lassen.

Der Aufbau des "Status-Datenbereichs" wird später noch beschrieben.

6.3.5 Feld Bild/Meld

In diesem Feld parametrieren Sie den Adressbereich (wird merker(bit)orientiert ausgewertet), der für den Funktionsblock "LEDs/Bild-Meldeaufwurf" verwendet werden soll.

Geben Sie die volle Bezeichnung ein, also z.B. A50.

Dann beginnt der Adressbereich, der für LED-Ansteuerung und Bild/Meldeaufwurf verwendet wird, bei Adresse 50hex.

Wieviele Merker (bits) jetzt benötigt werden, hängt von der Parametrierung der Felder "...x 8" ab.

6.3.6 Felder ...x 8

In diesen Feldern geben Sie jeweils an, wieviele Merkerbytes Sie für die einzelne Funktion reservieren wollen.

Wenn Sie eine Funktion nicht verwenden, geben

Handbuch Bediengeräte

Sie dort für die Anzahl 0 ein. Dann wird auch keine Adresse für diese Funktion "verschwendet".

Ein Beispiel:

Sie verwenden das ITS6101. Sie wollen 25 Bilder und 25 Meldungen aufrufen können, ebenso sollen die 8 LEDs angesteuert werden. Es sollen 5 Bilder als Prioritätsbilder erscheinen können. Sie halten einen Bereich ab Adresse A50 für das Bediengerät frei.

Für 25 Bilder benötigen Sie 4 Merkerbytes (4*8=32 Merker), ebenso für die Meldungen. Die Prioritätsbilder können sie über ein Merkerbyte (8 Merker) steuern.

Für die LEDs genügt ein Merkerbyte (=8 Merker). Insgesamt werden 10 Merkerbytes ab Adresse A50, also A50 - A5A (50 hex bis 5A hex) benötigt. Geben Sie dazu folgende Werte in die Felder ein:

Feld	Eingabewert
Bild/Meld	A50
Anzahl der Bilder	4
Anzahl der Meldungen	4
Anzahl der Prioritätsbilder	1
Anzahl der LEDs	1

Das Bediengerät fordert jetzt mit einem Read-Telegramm mit der Adresse 0050hex und der Datenlänge 10byte den Merkerbereich zyklisch an und wertet die Merker einzeln als Aufrufe aus (Adresse 50hex - 59hex =80 Merker).

Dabei ist die Zuordnung wie folgt:

Merker	Led/Bild/Meldung
A50.0	LED 1
A50.1	LED 2
...	...
A50.7	LED 8
A51.0	Bild 0
A51.1	Bild 1
....
A51.7	Bild 7
A51.8	Bild 8
...	...
A54.7	Bild 31
A55.0	Bild 0, priorisiert
A55.1	Bild 1, priorisiert
...	...
A55.7	Bild 7, priorisiert
A56.0	Meldung 1
A56.1	Meldung 2
...	...
A56.7	Meldung 8

A57.0	Meldung 9
...	...
A59.7	Meldung 32

Um nun z.B. eine LED zu setzen, setzen Sie einfach den Merker in Ihrem Steuerungsprogramm im Merkerbyte, welches der Adresse A50 zugeordnet ist - und am Bediengerät leuchtet die LED auf. Einfacher gehts nicht.

6.3.7 Feld Tasten

Hier müssen Sie angeben, wo das Bediengerät seine Tasten in "Merker" spiegeln soll.

Geben Sie die volle Bezeichnung ein, also z.B. A160.

Dann beginnt der Merkerbereich, wo das Bediengerät seine Tasten spiegelt, bei Adresse A160. Wieviele Merker jetzt benötigt werden, hängt von der Parametrierung des Feldes "Anzahl der Tasten" ab.

6.3.8 Feld Anzahl der Tasten

Geben Sie an, wieviele Merkerbytes Sie für den Tastenstatus vom Bediengerät reservieren wollen. Es werden immer 8 Merker gleichzeitig reserviert. Denn im Bediengerät werden auch nur Tastenerweiterungen in 8-er Schritten angeboten.

Beispiel:

Sie verwenden ein ITS6204 mit insgesamt 32 Tasten. Sie wollen alle Tasten in der Steuerung in Merker gespiegelt haben.

Sie müssen also 4 Merkerbytes reservieren. Als Adressbereich wollen Sie A160-A163 verwenden. Geben Sie in der Parametrieremaske die folgenden Einträge ein:

Feld	Eingabe
Tasten	A160
Anzahl der Tasten	4

Jetzt sendet Bediengerät den Tastenstatus als Write-Telegramm mit der Adresse 0160hex und der Datenlänge 4byte (Telegrammformat siehe Abschnitt 1.2.1). Die Merkerbytes (Merker) unter der Adresse A160 bedeuten jetzt:

Adresse (Merker)	= Taste Nr.
A160.0	1 (1. Reihe, links)
A160.1	2
...	...
A160.7	8 (1. Reihe, rechts)
A161.0	9 (2. Reihe, links)
....
A163.7	32 (4. Reihe, rechts)

Handbuch Bediengeräte

Numerikblock beim ITS/AT 6100:

Wenn die numerischen Tasten des ITS/AT 6100 ebenfalls abgefragt werden sollen, so müssen auf jeden Fall 8 Merkerbytes (=64 Merker) für die Tastenabfrage reserviert werden. Die ersten 48 Merker sind immer den Funktionstasten zugeordnet. Der Numerikblock wird ab dem 49. Merker übertragen.

Die Numeriktasten sind dann unter den Tastennummern nach folgender Tabelle zu finden (Beispiel: M0 als Basis):

Taste	Tastennr.	für Basis A0
Escape	50	A6.1
"4	"51	A6.2
"6	"52	A6.3
"2	"53	A6.4
"8	"54	A6.5
Enter	55	A6.6
"0	"57	A7.0
"1	"58	A7.1
"3	"59	A7.2
"5	"60	A7.3
"7	"61	A7.4
"9	"62	A7.5
."	"63	A7.6
"+/-	"64	A7.7

Die Merker für die Tasten 49 (A6.0) und 56 (A6.7) werden immer mit 0 gesetzt, bitte verwenden Sie diese Merker nicht anderweitig.

Die Funktionstasten am ITS6101 haben die Tastennummern 1-8, Wird das ITS6106 verwendet (Maximalausbau), sind die Funktionstasten von 1-48 nummeriert. Deshalb werden die Numeriktasten ab Tastennummer 50 in die Steuerung gespiegelt.

6.3.9 Die Status-Datenregister

Über die Status-Daten hält das Bediengerät die Steuerung über Bediener-Aktionen auf dem Laufenden. Es legt ab, welches Bild und welche Meldung gerade zur Anzeige kommen und in welchem Bedien-zustand es sich befindet.

Aber über diesen Status-Bereich werden auch weitere Funktionen des Bediengerät angesprochen. Sie haben Zugriff auf den CAN-Bus, der am Bediengerät angeschlossen sein kann. Außerdem können Sie Geräteparameter wie Kontrast und Helligkeit über den Status-Bereich beeinflussen. Die in der folgenden Beschreibung gemachten Angaben von Statusdaten-Registern (Datenworte) beziehen sich immer als Offset auf die eingestellte Basisadresse, die Sie im Feld "Status" der Parametriermaske eingetragen haben (Beispiel: Steht im Text D2 und Sie haben im Feld "Status" A200

angegeben, so ist die "echte" Adresse des Statusbytes A204 [2bytes pro Datenregister reserviert]). Die folgende Tabelle gibt Aufschluß über die Verwendung der Datenregister im Status-Bereich:

Datenregister	Funktion
D0-D9	Gerätstatus des Bediengerät
D10-D15	Sendepuffer für CAN-Gateway
D16-D21	Empfangspuffer für CAN-Gateway

6.3.10 Gerätstatus-Informationen

Das Bediengerät informiert die SPS standardmäßig über die folgenden Datenworte mit dem in der Tabelle aufgeführten Inhalt:

Datenregister	Inhalt des Datenregisters
D0	Bildnummer des aktuell angezeigten Bildes
D1	Meldenummer der aktuell angezeigten Meldung 0=keine Meldung wird angezeigt
D2	Gerätezustand.
D3	Anzahl der aktiven Bilder
D4	Anzahl der aktiven Meldungen
D5-D9	nicht belegt, reserviert

6.3.11 CAN-Gateway Sendepuffer

Die Datenworte D10-D15 des Statusbereichs sind wie folgt belegt:

Wort	Funktion
D10	Handshake. K0: Sendepuffer frei sonst Sendepuffer belegt
D11	CAN-Identifizier Hier muß das Anwenderprogramm den Adressaten eintragen. Adresse 0 ist das Bediengerät selbst.
D12	Telegramm-Art/CAN-Nutzdaten
D13	Funktionswort 0/CAN-Nutzdaten
D14	Funktionswort 1/CAN-Nutzdaten
D15	Funktionswort 2/CAN-Nutzdaten

Handshake über D10:

Über das D10 erfolgt eine Abstimmung zwischen Anwenderprogramm und Bediengerät/CAN-Bus. Enthält das Datenregister D0 den Wert 0, dann ist der Sendepuffer frei und die Daten dürfen in den Sendepuffer geschrieben werden.

Hiermit läßt sich verhindern, daß die SPS zu schnell Daten an den CAN-Bus (oder das Bedien-

Handbuch Bediengeräte

gerät selbst) ausgibt.

Telegrammart und Funktionsworte:

Die Telegrammart und die Funktionsworte sind abhängig vom Adressaten in D11:

D11=0 Daten für das Bediengerät an der PG-Schnittstelle	D11<>0 Daten sind für den CAN-Bus bestimmt
D12 Telegramm-Art - Bildaufruf - Bild löschen - Melde-Aufruf - Meldung löschen - Parametrier-Kommandos	D12 KH=aabb CAN-Nutzdaten: aa = byte0 bb = byte1
D13 Funktionswort 1 entsprechend der Telegramm-Art D12 belegt, siehe nachfolgender Abschnitt	D13 KH=ccdd CAN-Nutzdaten: cc = byte2 dd = byte3
D14 Funktionswort 2 (entsprechend der Telegramm-Art D12 belegt, siehe nachfolgender Abschnitt)	D14 KH=eeff CAN-Nutzdaten: ee = byte4 ff = byte5
D15 Funktionswort 3(entsprechend der Telegramm-Art D12 belegt, siehe nachfolgender Abschnitt)	D15 KH=gghh CAN-Nutzdaten: gg = byte6 hh = byte7

Sendungen an das Bediengerät, D11=0:

Über den Sendepuffer (Status-Bereich) lassen sich folgende Kommandos an das Bediengerät absetzen:

Telegramm-Art in D12	Funktionsworte 1-3 D13 - D15
2 Variable senden (Set Value)	D13: Handle der Variablen D14: Variablenwert (lowword) D15: Variablenwert (highword)
4 Meldeaufruf	D13: Nummer der aufzurufenden Meldung, von 1 bis 9999 D14/15: nicht benutzt
5 Meldung austragen	D13: Nummer der auszutragenden Meldung, von 1 bis 9999 D14/15: nicht benutzt
6 Bild aufrufen	D13: Nummer des aufzurufenden Bildes, von 1 bis 9999 D14/15: nicht benutzt
7 Bild austragen	D13: Nummer des auszutragenden Bildes, von 1 bis 9999 D14/15: nicht benutzt
8 Prioritäts-Bild aufrufen	D13: Nummer des aufzurufenden Prioritätsbildes, von K1 bis K9999 D14/15: nicht benutzt
5376 bis 5384	Geräteparameter setzen. Siehe "WRITE_PARAM (0x15)" auf Seite 13

5376 Globale Softkey-Maske setzen	D13: Bitmaske für Softkeys, KH=00xx bit 0 = nicht belegt bit 1 = Menütaste 1 bit 2 = Menütaste 2 ... bit 6 = Menütaste 6 bit 7 = nicht belegt Ist das Bit einer Taste gesetzt, so wird die Softkeyfunktion für diese Taste gesetzt und die Menüfunktion wird abgeschaltet. Ist das Bit 0, so wird die Menüfunktion der Taste aktiviert. D14/15: nicht benutzt
5377 Kontrast setzen	D13: Kontrastwert 0-15 (0 bis 23). 23 ist maximaler Kontrast D14/15: nicht benutzt
5378 Helligkeit der Hintergrundbeleuchtung	D13: Helligkeitswert 0-7 (0 bis 7). 7 ist maximale Helligkeit. D14/15: nicht benutzt
5379 Statuszeile ein/aus	D13: Statuszeilenfunktion 0 bis 2 0: Statuszeile eingeblendet 1: Statuszeile ausgeblendet 2: wie im Bild definiert D14/15: nicht benutzt
5380 Position der Statuszeile	D13: Zeilennummer der Statuszeile (0 bis 7). 0 ist oberste Zeile. D14/15: nicht benutzt
5381 Rollierzeit der Meldungen	D13: Rollierzeit in Sekunden, von 0 bis 32. 0 = "Rollieren aus" D14/15: nicht benutzt
5382 Rollierzeit der Bilder	D13: Rollierzeit in Sekunden, von 0 bis 32. 0 = "Rollieren aus" D14/15: nicht benutzt
5383 Tastenbelegung der Menütasten	D13: KH=uuvv mit uu=Nummer der ESC-Taste vv=Nummer der "Pfeil links" -Taste D14: KH=wwxx ww=Nummer der "Pfeil rechts"-Taste xx=Nummer der "Pfeil ab" -Taste D15: KH=yyzz yy=Nummer der "Pfeil auf"-Taste zz=Nummer der Enter-Taste
K5384 Meldeausgang	D13: 0 ausschalten 256 einschalten D14/15: nicht belegt

Diese Daten sind in Anlehnung an das CAN-Datenformat formatiert (siehe "Beschreibung der Telegrammart" auf Seite 4).

6.3.12 CAN-Gateway Empfangspuffer

Bevor Sie allzu euphorisch auf den CAN-Bus zugreifen wollen: bedenken Sie, daß auf dem CAN-Bus eine Übertragungsrate von bis zu 1 MBit/s eingestellt werden kann. Auf der RS232 Schnittstelle sind max. 9600 Baud einstellbar.

Wenn jetzt ein CAN-Modul nur 10-mal pro Sekunde ein Telegramm absetzt, so müßte 10 Mal pro Sekunde der Empfangspuffer in die Steuerung geschrieben werden, und nebenher auch noch die Merkerbytes und Variablen ausgelesen werden - unmöglich!

Handbuch Bediengeräte

Also wozu das Ganze?

Denken Sie daran, daß z.B. eine Bedientastatur ITS6300 über den CAN-Bus an das Bediengerät angeschlossen werden kann. Irgendwie müssen Sie dann gemeldet bekommen, wenn eine Taste auf dem ITS6300 gedrückt wird - und das geht nur über das Bediengerät. Und ehrlich gesagt: so einfach können Sie keine anderen Tastaturen an die Steuerung anschließen wie über den CAN-Bus. Das ITS6300 sendet nun jedesmal eine CAN-Nachricht an das Bediengerät, wenn eine Taste gedrückt wird. Das Bediengerät legt dann dieses Telegramm im Empfangspuffer ab.

Realistisch gesehen wird ein Bediener höchstens 2-3 mal pro Sekunde eine Taste drücken. Das Bediengerät kann Tasteneingaben gegebenenfalls in einem FIFO-Puffer mit 20 Telegrammen Tiefe zwischenspeichern.

Wenn Sie dem Bediener über eine LED signalisieren, daß sein Tastendruck registriert wurde, so wird er nicht anfangen, wie wild auf der Tastatur zu hämmern.

Aber nun zur Beschreibung der Datenregister des Empfangspuffers. Auch der Empfangspuffer hält ein Handshake-Register zur Verfügung, anhand dessen der Datentransfer kontrolliert wird, sowie die Informationsbytes:

Datenregister	Funktion
D16	Handshake. 0: Empfangspuffer leer sonst Daten im Puffer
D17	CAN-Identifizier Hier erhält das Anwenderprogramm in der SPS die Adresse des Senders. Adresse 0 ist das Bediengerät selbst.
D18	Telegramm-Art/CAN-Nutzdaten
D19	Funktionswort 0/CAN-Nutzdaten
D20	Funktionswort 1/CAN-Nutzdaten
D21	Funktionswort 2/CAN-Nutzdaten

Sendungen vom Bediengerät: D17=0

Derzeit sind keine Telegramme vom Bediengerät an die SPS definiert. Alle Funktionen werden über Merker und Datenregister abgewickelt.

Sendung von CAN-Modul: D17<>0:

In diesem Fall wird das CAN-Telegramm des Senders 1:1 in den Empfangspuffer geschrieben. Dabei werden die Bytes wie folgt abgelegt:

Datenreg.	Inhalt
D17	CAN-Identifizier (Senderadresse)

D18	CAN-Nutzdaten KH=aabb aa = byte 0 bb = byte 1
D19	CAN-Nutzdaten KH=ccdd cc = byte 2 dd = byte 3
D20	CAN-Nutzdaten KH=eeff ee = byte 4 ff = byte 5
D21	CAN-Nutzdaten KH=gghh gg = byte 6 hh = byte 7

Der Inhalt des CAN-Telegramms ist abhängig vom Gerät, das das Telegramm gesendet hat. Schlagen Sie deshalb im Handbuch zu diesem Gerät nach, wenn Sie den Inhalt der Telegramme bestimmen müssen.

6.4 CAN-Identifizier D11 und D17

Der CAN-Identifizier setzt sich aus insgesamt 16 Bit zusammen. Die einzelnen Bits haben folgende Bedeutung:

15-5	4	3-0	
Identifizier	R	DLC	
0 0 0 0 0 0 0 1 0 0 1 0	1	0 0 0 0	
0	1	2	8

In den Bits 0-3, DLC (Data Length Code), wird angegeben, wie viele Bytes Nutzdaten das CAN-Telegramm enthält. Dieser Wert kann 0 bis 8 sein. Maximal kann ein CAN-Telegramm 8 Byte Nutzdaten enthalten.

Das RTR-Bit (R) wird derzeit nicht verwendet. Setzen Sie es deshalb auf 0.

Die ID-Bits 0-10 müssen die Nummer des Geräts enthalten. Diese wird meistens über DIP-Schalter oder Jumper gesetzt. Details entnehmen Sie bitte dem Handbuch des jeweiligen Geräts.

Handbuch Bediengeräte

7 VT100-Treiber

Mit dem VT100-Treiber kann das Bediengerät wie ein VT100-Terminal angesteuert werden. Dabei sind auch XOn-XOff-Steuersequenzen implementiert.

Bilder können frei gestaltet werden und über die VT100-Steuersequenzen können dann Bildinhalte verändert werden. Sie können auch mit einem gänzlich leeren Bild beginnen und alle Informationen übertragen.

7.1 Projektierung

Mit dem ITE wird die Projektdatei VT100.txt mitgeliefert, die einen leeren Bildschirm darstellt und den VT100-Treiber initialisiert. Dieses Projekt beinhaltet folgende Voreinstellungen:

- "Bild 0" ist als Leerbild definiert, d.h. es befinden sich weder Texte, Grafiken noch Variablen in dieser Bildschirmseite
- ("Bild 0" ist die angezeigte Bildschirmseite nach einem "Power-Up" des Bediengeräts)
- Im Ordner "Gerät/Parametrieren/Programme" sind die aktuellen TOS und Treiberversionen IO042S00.hex (TOS) und VT042S00.drv (VT100-Treiber) ausgewählt
- Im Ordner "Gerät/Parametrieren/Gerät" ist die Geräteversion ITS6100 ausgewählt
- Im Ordner "Gerät/Parametrieren/serielle-Schnittstelle" ist die Einstellung "VT100-Treiber" und die VT-Option "0" ("0"=Standard Tastencodes, "1"=zusätzliche Codes) ausgewählt. Die Schnittstellen Parameter sind auf 9600Baud, 2stopbits, 8 databits, no parity eingestellt.

Sollte es notwendig sein können Sie nun eine Änderung der eingestellten Werte vornehmen und das Projekt ins Bediengerät übertragen. Verbinden Sie hierzu das Bediengerät mittels des Adapters ITK100 mit Ihrem PC und übertragen Sie Ihr Projekt ins Gerät.

Nach dem Einschalten des Bediengerätes erscheint für ca. 2 sec. ein Werbebild (dieses kann im Editor ITE erstellt werden, siehe Handbuch Bediengerät). In Zeile 7 und 8 des Werbebilds werden die Versionen von BIOS (Firmware), TOS (Betriebssystem), Treiber, sowie der Projektnamen (UserData) angezeigt.

Wurde der VT100-Treiber geladen erscheint in Zeile 7 des Werbebilds der Text "DRV:VT". Ein Blinken der Treiberkennung "VT" zeigt an, daß Treiber und Betriebssystem (TOS) inkompatibel sind. Stellen Sie sicher, daß die Versionsnummern von Treiber und TOS identisch sind (Ordner Gerät/Parametrieren/Programme, z.B. IO042S00.hex

und VT042S00.drv für Version 042S00).

Nach Ablauf der Einschaltzeit (ca. 2 sec) erscheint das projektierte Leerbild (Bild0) in der Anzeige. Die Kommunikation via VT100-Schnittstelle kann beginnen (siehe nachfolgende Protokollbeschreibung).

7.2 Beschreibung der VT-100 Steuerfolgen

Nachfolgend sind alle Escape- und Control Sequenzen des VT100-Treibers der Bediengeräte-Reihe beschrieben. Die Sequenzen sind dem Terminal-Handbuch VT101 der Firma "Digital Equipment Corporation (DEC)" entnommen.

7.2.1 Control-Zeichen (Empfang)

Name	Mnemonic	HEX	Beschreibung
Null	NUL	0x00	Füllzeichen (wird ignoriert)
Enquire	ENQ	0x05	Sendet Rückantwort
Bell	BEL	0x07	Alarm-Ton (Meldeausgang wird 500ms aktiviert)
BackSpace	BS	0x08	Cursor left
LineFeed	LF	0x0A	Zeile nach unten / Spaltenpos. wird beibehalten
VerticalTab	VT	0x0B	wie LF
FormFeed	FF	0x0C	wie LF
Carriage-Return	CR	0x0D	Cursor auf Anfang der aktuellen Zeile positionieren
XON	DC1	0x11	XON: Terminal Sendefreigabe
XOFF	DC3	0x13	XOFF: Terminal darf keine weiteren Zeichen senden
Cancel	CAN	0x18	Empfangene Sequenz wird gelöscht, substitution Character wird angezeigt
Substitute	SUB	0x1A	wie Cancel (CAN)
Escape	ESC	0x1B	Startbyte einer Sequenz

Handbuch Bediengeräte

7.2.2 ESC-Sequenzen (Empfang)

Name	Mnemonic	SequenzASCII /HEX	Beschreibung
Cursor Positionierung			
Cursor Up	CUU	ESC [n A (1B 5B n 41)	Cursor um "n"-Zeilen nach oben verschieben wobei n = '0'-'99' (ASCII)
Cursor Down	CUD	ESC [n B (1B 5B n 42)	Cursor um "n"-Zeilen nach unten verschieben wobei n = '0'-'99' (ASCII)
Cursor right	CUF	ESC [n C (1B 5B k 43)	Cursor um "n"-Stellen nach rechts verschieben wobei n = '0'-'99' (ASCII)
Cursor left	CUB	ESC [n D (1B 5B n 44)	Cursor um "n"-Stellen nach links verschieben wobei n = '0'-'99' (ASCII-Eingabe)
Cursor Position	CUP	ESC [n ; m H (1B 5B n 3D m 48)	Cursor auf Zeile "n" und Spalte "m" wobei n,m = '0'-'99' (ASCII-Eingabe)
CursorPos-Home	CUP	ESC [H (1B 5B 48)	Cursor Home-Position: "links oben"
Index	IND	ESC D (1B 44)	Cursor 1. Zeile nach unten Spaltenposition bleibt (LF)
Reverse Index	RI	ESC M (1B 4D)	Cursor 1. Zeile nach oben, Spaltenposition bleibt
NextLine	NEL	ESC E (1B 45)	Cursor auf 1. Spalte in der nächsten Zeile
Zeichenausgabe			
Character		ASCII-Code 20hex - FFhex	Alle ASCII-Zeichen ab 20hex können angezeigt werden (Steuerzeichen 0 - 1F hex werden nicht angezeigt)
Löschen			
Erase in line	EL	ESC [K (1B 5B 4B)	Lösche von Cursor bis Zeilenende
Erase in line	EL	ESC [0 K (1B 5B 30 4B)	Lösche von Cursor bis Zeilenende (siehe oben)
Erase in line	EL	ESC [1 K (1B 5B 31 4B)	Lösche vom Anfang der Zeile bis zur Cursorpos.
Erase in line	EL	ESC [2 K (1B 5B 32 4B)	ganze Zeile löschen (Zeile des Cursors)
Erase in display	ED	ESC [J (1B 5B 4A)	Lösche von Cursor bis Bildschirmende
Erase in display	ED	ESC [0J (1B 5B 30 4A)	Lösche von Cursor bis Bildschirmende (wie oben)
Erase in display	ED	ESC [1J (1B 5B 31 4A)	Lösche vom Bildschirmanfang bis zum Cursor
Erase in display	ED	ESC [2J (1B 5B 32 4A)	Lösche ganzes Display
Reports			
Status Report	DSR	ESC [5 n (1B 5B 35 6E)	Status anfordern Antwort: ESC [0 n (Terminal OK) ESC [3 n (Terminal not OK)

Handbuch Bediengeräte

Status Report	DSR	ESC [6 n (1B 5B 36 6E)	Cursor Position anfordern Antwort: ESC [n ; m R (n=Zeile, m=Spalte) wobei n,m = '0' - '99' (ASCII-Format)
Terminal Reset			
Reset	RIS	ESC c (1B 63)	Terminal Reset auslösen
LED-Ansteuerung			
Load LEDs	DE- CLL	ESC [q (1B 5B 71)	alle LEDs off
Load LEDs	DE- CLL	ESC [0 q (1B 5B 30 71)	alle LEDs off (wie oben)
Load LEDs	DE- CLL	ESC [n q (1B 5B n 71)	LED "n" on (n = LED-Nummer)wo- bei n = '0'-'99' (ASCII)
Zeichenattribute			
Attribute reset	SGR	ESC [m (1B 5B 6D)	Attribute ausschalten: Schriftgröße 8 Pixel Normaldarstellung (kein blinken)
Attribute reset	SGR	ESC [0 m (1B 5B 30 6D)	Attribute ausschalten: (wie oben)
Underline	SGR	ESC [4 m (1B 5B 34 6D)	Zeichen unterstrichen
Schriftgröße 16	SGR	ESC [5 m (1B 5B 35 6D)	Schriftgröße 16Pixel
Schriftgröße 16	SGR	ESC [6 m (1B 5B 36 6D)	Schriftgröße 32Pixel
Invers	SGR	ESC [7 m (1B 5B 37 6D)	Zeichen invers
Blinken "EIN"	"SGR	ESC [8 m (1B 5B 38 6D)	Zeichen blinkend
Blinken "AUS"	"SGR	ESC [9 m (1B 5B 39 6D)	Funktion "blinken" wird deaktiviert
Sondertelegramme (Bediengerät-Erweiterungen)			
Meldeausgang OFF	EXT	ESC [0 x (1B 5B 30 78)	Meldeausgang deaktivieren
Meldeausgang ON	EXT	ESC [1 x (1B 5B 31 78)	Meldeausgang aktivieren
Cursor OFF	EXT	ESC [2 x (1B 5B 32 78)	Cursor wird nicht mehr angezeigt
Cursor ON (Block)	EXT	ESC [3 x (1B 5B 33 78)	Cursor wird als Block angezeigt
Cursor ON (Underline)	EXT	ESC [4 x (1B 5B 34 78)	Cursor wird als "Underline" ange- zeigt
Displayhelligkeit einstellen	EXT	ESC [n l (1B 5B n 6C)	Helligkeit (n) des Displays setzen- wobei n = '0'-'7' (ASCII)
Displaykontrast einstellen	EXT	ESC [n c (1B 5B n 63)	Kontrastwert (n) für das Display setzenwobei n = '0'-'23' (ASCII)

Handbuch Bediengeräte

7.2.3 Tastencodes

(Terminal Transmitted Characters)

Name	Sequenz /Code	Beschreibung
Cursor-Tasten (nur ITS7000-Reihe)		
Up	ESC [A	Cursor Up gedrückt
Down	ESC [B	Cursor Dwon gedrückt
Right	ESC [C	Cursor right gedrückt
Left	ESC [D	Cursor left gedrückt
Numerik-Tasten (Hex-Format)		
0	0x30	
1	0x31	
2	0x32	
3	0x33	
4	0x34	
5	0x35	
6	0x36	
7	0x37	
8	0x38	
9	0x39	
.	0x2C	Komma
+/-	0x2D	Minus
ENTER	0x0D	Enter
ESC	0x1B	Escape
BS	0x08	BackSpace
Funktionstasten		
F1	'A' (0x41)	F1 sendet ein ASCII 'A'
F2	'B' (0x42)	F2 sendet ein ASCII 'B'
F3	'C' (0x43)	F3 sendet ein ASCII 'C'
F4	'D' (0x44)	F4 sendet ein ASCII 'D'
F5	'E' (0x45)	F5 sendet ein ASCII 'E'
F6	'F' (0x46)	F6 sendet ein ASCII 'F'
F7	'G' (0x47)	F7 sendet ein ASCII 'G'
F8	'H' (0x48)	F8 sendet ein ASCII 'H'

- *Tasten senden nur mit positiver Flanke ihren Code ("Taste losgelassen" wird nicht gemeldet).*
- *Answerback-Telegramm (= Antwort auf ein ENQ der Steuerung) ist implementiert. Das Answerback Telegramm ist ein Identifikationsstring mit max. 20 Zeichen (in der Bediengerät-Reihe ist der Answerback-String als Konstante 'Bediengerät' festgelegt).*

7.3 Erweiterungen des VT100-Treibers

In der Einstellung "Option = 1" sind folgende zusätzliche Steuersequenzen verfügbar:

Handbuch Bediengeräte

7.3.1 ESC-Sequenzen (Empfang)

Name	Sequenz	Beschreibung
Cursor positioning	ESC=XY	X: Zeilenposition 1 bis 8 Y: Spaltenpos. 1 bis 40 X-Wert ist zwischen 0x20 bis 0x27 (Zeile 1 bis 8) Y-Wert ist zwischen 0x20 bis 0x66 (Spalte 1 bis 40) Beispiel: 1B 3D 21 26 (Zeile 2, Spalte 7)
Line erasing	ESC T (1B 54)	Komplette Zeile des Cursors wird gelöscht
Display erasing	ESC + (1B 2B)	Das komplette Display wird gelöscht
Blinking start	ESC j (1B 6A)	Die nachfolgend gesendeten Zeichen werden "blinkend" dargestellt
Blinking stop	ESC k (1B 6B)	Die Funktion "Blinken" wird ausgeschaltet

Taste Minus	' '	0x20
ESC kombiniert mit Taste Minus	','	0x2E
Taste ENTER	CR	0x0D

7.3.2 Gesendete Tastencodes

Taste gedrückt	ASCII	HEX-Code
Funktionstaste F1	'A'	0x41
Funktionstaste F2	'B'	0x42
Funktionstaste F3	'G'	0x47
Funktionstaste F4	'I'	0x49
Funktionstaste F5	'K'	0x4B
Funktionstaste F6	'C'	0x43
Funktionstaste F7	'F'	0x46
Funktionstaste F8	'J'	0x4A
ESC kombiniert mit Ziffer 1	'D'	0x44
ESC kombiniert mit Ziffer 2	'H'	0x48
ESC kombiniert mit Ziffer 3	'L'	0x4C
ESC kombiniert mit Ziffer 4	'E'	0x45
ESC kombiniert mit Ziffer 5	'M'	0x4D
ESC kombiniert mit Ziffer 6	'N'	0x4E
ESC kombiniert mit Ziffer 7	'O'	0x4F
ESC kombiniert mit Ziffer 8	'P'	0x50
Zifferntaste '0'	'0'	0x30
Zifferntaste '1'	'1'	0x31
Zifferntaste '2'	'2'	0x32
Zifferntaste '3'	'3'	0x33
Zifferntaste '4'	'4'	0x34
Zifferntaste '5'	'5'	0x35
Zifferntaste '6'	'6'	0x36
Zifferntaste '7'	'7'	0x37
Zifferntaste '8'	'8'	0x38
Zifferntaste '9'	'9'	0x39
Taste Dezimalpunkt	','	0x2E

8 Intercontrol DIGSYplus

Die Intercontrol DIGSYplus Steuerung kann über die COM-SP Schnittstelle an das Bediengerät angeschlossen werden. Die Datenübertragung über die Schnittstelle erledigt das Bediengerät. Sie brauchen nur die erforderlichen Merker und Variablendaten in der SPS abzulegen - der Zugriff erfolgt parametrisiert vom Bediengerät. Sie brauchen also keinen "Transferprogrammcode" in Ihrem SPS-Programm vorzusehen.

8.1 Prinzipielle Funktion des Treibers

Im Editor ITE werden die Datenbereiche parametrisiert, auf die das Bediengerät zugreifen soll. Sie müssen nicht jede einzelne Meldung bzw. Variable zuweisen. Es wird jeweils für die einzelnen Daten eine Art Anfangsadresse angegeben, und über die Handle-Nummer wird dann der Abstand ("Offset") zu dieser Anfangsadresse angegeben. Das Bediengerät sorgt jetzt dafür, daß ständig ein Datenaustausch mit den Meldemerkern ("Wortbits") und den aktuell im Bild angezeigten Soll-/Istwerten stattfindet. Dies geschieht ohne das Zutun der SPS.

Der Treiber erlaubt folgende Funktionen:

- Bildaufruf (auch PRIO-Bilder!) über Wortbits
- Meldeaufruf über Wortbits
- Istwertanzeige aus Merkerworten
- Sollwerteingabe in Merkerworte
- Tasten- und Touchscreenabfrage über Wortbits
- LED ein- und ausschalten über Wortbits
- Gerätestatus beeinflussen
- Zugriff auf den CAN-Bus, der am Bediengerät angeschlossen sein kann ("Gateway")

8.2 Grundsätzliche Überlegungen

Sie müssen zunächst planen, wo Sie die Daten für das Bediengerät in der SPS anlegen. Beachten Sie dazu folgende Vorgaben:

- Bild- und Meldeaufrufe sowie Tasten- und LED-Funktionen werden über Wortbits (in Merkerworten) abgewickelt. Das Bediengerät benötigt einen zusammenhängenden Bereich für alle diese Funktionen. Reservieren Sie also einen Block zusammengehöriger Merkerworte.
- Bilder und Meldungen müssen von 1 beginnend fortlaufend nummeriert sein, wenn Bild/Meldeaufruf über Merker erfolgt. Der Editor läßt zwar Lücken zu, bei Verwendung des Intercontrol-Treibers müssen Sie aber auf "lückenlose" Erstellung achten.
- Die Reihenfolge, wie die Funktionen auf Merkerworte umgesetzt werden, ist immer gleich.

Sie können angeben, wie viele Merkerbytes pro Funktion verwendet werden sollen.

- Es wird immer in Merkerbyte-Schritten (8 Merker) pro Funktion vorgegangen
- Die Reihenfolge ist immer wie folgt:
 - Merker für LED-Ansteuerung
 - Merker für Bildaufruf
 - Merker für Prioritätsbilder
 - Merker für Meldungen
- Es können maximal 256 Merker (=16 Merkerworte) für Bild/Meldeaufrufe verwendet werden (Summe!). Genügt dies nicht, können weitere Aufrufe über die "Gateway"-Funktion vorgenommen werden.
- Maximal werden für diese Funktionen 320 Merker benötigt (256 für Bild/Meldeaufrufe, 64 für LED-Ansteuerung)
- Soll- und Istwerte (Variablen) werden ebenfalls über Merkerworte ausgetauscht. Es ist möglich, für Sollwerte, Istwerte und Grenzen einen gemeinsam genützten Merkerwort-Bereich zu parametrieren.
- Das Handle wird zu dem parametrisierten Wert des Datenregisters addiert ("Offset"). Wenn Sie also hier etwas von Handle lesen, so ist dies synonym zu "Merkerwort-Offset". Selbstverständlich können Sie die Anfangsadresse für alle Datentypen gleich einstellen. Am einfachsten ist es sowieso, wenn Sie für die Merkerworte jeweils die Adresse 3FE einstellen (außer Status - dazu später mehr). Denn dann ist die Handle-Nummer, die Sie für Variablen einstellen, identisch mit der Merkerwort-Nummer. So sehen Sie anhand der Handle-Nummer sofort, auf welches Merkerwort Sie (bzw. das Bediengerät) zugreifen.
- Alle Variablen mit einer Länge von 1-16 Bit belegen automatisch ein ganzes Merkerwort
- Variablen mit einer Länge von 32 Bit belegen zwei aufeinanderfolgende Merkerworte. Dies ist bei der Vergabe von Handles zu berücksichtigen. Beispiel: Hat eine Longword-Variablen das Handle 6, so belegt Sie automatisch zwei Merkerworte (z.B. MW6 und MW7). Es sollte also keine Variable mit dem Handle 7 verwendet werden.
- Aufgrund von Treiber-Beschränkungen auf der SP-Schnittstelle dürfen innerhalb eines Bildes keine Variablen derselben Art (Sollwerte, Istwerte, Ober- Untergrenze) verwendet werden, deren Handles mehr als 32 Unterschied haben. Beispiel: Ist das kleinste Handle von Sollwerten im Bild 10, so darf das größte Handle von Sollwerten im Bild höchstens 42 sein. Ein Istwert darf aber jetzt z.B. wieder das Handle 90 haben, da er zu einem anderen Typ gehört. Innerhalb der Istwerte darf der "Handle-Unterschied" aber wieder nicht größer

Handbuch Bediengeräte

als 32 sein.

WICHTIG!!! Alle Felder der Parametriermaske müssen ausgefüllt werden. Dabei muß für jede Funktion eine Merkerwortadresse angegeben werden, auch wenn die Funktion nicht benutzt wird. Sonst kommt keine Kommunikation zustande.

In der Praxis zeigt sich, daß diese Regeln sehr einfach zu handhaben sind, da die Parametrierung des Treibers und der Variablen sehr komfortabel möglich ist.

Folgende Datenarten können/müssen Sie einzeln parametrieren:

- Istwerte (P1 in der Registerkarte)
- Sollwerte (P5 in der Registerkarte)
- Untergrenzen (P6 in der Registerkarte)
- Obergrenzen (P7 in der Registerkarte)
- Schrittwerte (P8 in der Registerkarte)
- Statusinfo's (P2 in der Registerkarte)
- Bild/Meldeaufruf, LED's (P3 in der Registerkarte)
- Tasten (P4 in der Registerkarte)

8.3 Parametrierung des Treibers

Über das Menü "Gerät"/"Parametrieren", Registerkarte "Serielle Schnittstelle" erreichen Sie die Parametriermaske für den Treiber. Klicken Sie den Knopf "Anderer Treiber" an. Dann erscheint die folgende Maske:

The screenshot shows a software configuration window titled "Grund-Einstellungen Graf-ITE" with a close button (X). It has several tabs: "CAN Einstellungen", "CAN Module", "Bildaufbau", "Schnelle Zähler", "Allgemein", "Gerät", "Programme/Zeichensatz", "Meldungen und Bilder", "Drucker", "Statistik", and "Serielle Schnittstelle". The "Serielle Schnittstelle" tab is active. It contains a section for "SPS-Treiber (RS232)" with radio buttons for "Kein Treiber", "Siemens S5 [AS511]", "Mitsubishi FX", "Frei (Request/Response)", "Frei (Telegrammverkehr)", and "VT100". There is an "Anderer Treiber" text field and "OK" and "Abbruch" buttons. Below this is a note: "Bitte auch den Treiber unter 'Programme' angeben!". The "Treibereinstellungen" section has a grid of input fields for P1 through P8, with S1-S5 also visible. P1-P8 are set to 0. S1 is 8, S2 is 16, S3 is 8, S4 is 1, and S5 is 1. The "Einstellungen serielle Schnittstelle" section has radio buttons for Baudrate (9600, 1200, 4800, 600, 2400, 300), Parity (none, odd, even), Datenbits (7, 8), and Stopbits (1, 2).

In das Feld hinter der Auswahl "Anderer Treiber" geben Sie "IC" ein (für InterControl). Die Felder P1 bis P8 werden mit absoluten Adressen (Angabe hexadezimal) für verschiedene Aufgabenbereiche gefüttert. Die Adressen errechnen sich zu:

$$\text{MW-Adresse} = 1024 + (\text{Merkerwortnummer}-1) \cdot 2$$

Beispiel: das Merkerwort 1 hat die Adresse $1024+(1-1) \cdot 2 = 1024 = 400\text{hex}$

Beispiel: das Merkerwort 50 hat die Adresse $1024+(50-1) \cdot 2 = 1122 = 462\text{hex}$

Die Felder bedeuten:

8.3.1 P1: Basis-Adresse für Istwerte

Diese Einstellung gibt an, ab welchem Merkerwort die Istwert-Variablen aus der SPS gelesen werden. Zu dieser Adresse wird jeweils der Handle der Variablen addiert. Auf diese Weise können bis zu 256 Istwerte "adressiert" werden. Beispiel: wenn Sie als Basisadresse 400(hex) eingegeben haben, so entspricht die Variable mit Handle 0 dem MW1, Variable mit Handle 1 dem MW2 usw. Tip: Wenn Sie als Anfangsadresse die 3FE (hex) verwenden, dann entspricht die Handle-Nummer genau der MW-Nummer. Sie dürfen dann aber das Handle 0 nicht verwenden. MW0 gibt's ja schließlich nicht....

8.3.2 P2: Basis-Adresse für Statusinfo

Hier geben Sie die Adresse des Merkerworts an, ab dem die Statusinformationen des Bediengeräts abgelegt werden. Es werden 22 Merkerworte hierfür benötigt. Auch der Gateway-Puffer fällt in diesen Bereich.

8.3.3 P3: Basis-Adresse für LED und Aufrufe

Dieses Feld muß die Anfangsadresse für die LED-Merker, Bild- und Meldeaufrufmerker ("Wortbits") enthalten. Die Anzahl der erforderlichen Merkerworte hängt von der Parametrierung der Felder S1 bis S4 ab.

8.3.4 P4: Basis-Adresse für Tastenmerker

Das Bediengerät meldet gedrückte Tasten bitweise in die SPS. Hier können Sie die Adresse des Merkerworts angeben, ab dem die Tastenmerker abgelegt werden. Die Anzahl der Merker parametrieren Sie über das Feld S5.

8.3.5 P5: Basisadresse für Sollwerte

Parametrierung wie P1

8.3.6 P6: Basisadresse für Untergrenzen

Parametrierung wie P1

8.3.7 P7: Basisadresse für Obergrenzen

Parametrierung wie P1

Handbuch Bediengeräte

8.3.8 P8: Basisadresse für Schrittweiten

Parametrierung wie P1

8.3.9 Feld S1

In diesem Feld parametrieren Sie, wie viele Merkerworte Sie für den Bildaufruf verwenden wollen. Zu beachten ist hierbei, dass die Eingabe dezimal erfolgt und in 2-er Schritten erwartet wird. Wenn Sie also 2 Merkerworte = 32 Bild-Merker (32 Wort-Bits) parametrieren wollen, geben Sie bitte eine 4 ein. Anmerkung: dies entspricht den Bildern 1-32.

8.3.10 Feld S2

In diesem Feld parametrieren Sie, wie viele Merkerworte Sie für den Meldungsaufwurf verwenden wollen. Zu beachten ist, dass die Eingabe dezimal erfolgt und in 2-er Schritten erwartet wird. Wenn Sie also 1 Merkerwort = 16 Melde-Merker (16 Wortbits) parametrieren wollen, geben Sie bitte eine 2 ein. Anmerkung: dies entspricht den Meldungen 1-16.

8.3.11 Feld S3

In diesem Feld parametrieren Sie, wie viele Merkerworte Sie für den priorisierten Bildaufruf verwenden wollen. Zu beachten ist hierbei, dass die Eingabe dezimal erfolgt und in 2-er Schritten erwartet wird. Wenn Sie also 1 Merkerwort = 16 Bild-Merker (16 Wortbits) parametrieren wollen, geben Sie bitte eine 2 ein. Anmerkung: dies entspricht den Bildern 1-16.

8.3.12 Feld S4

In diesem Feld parametrieren Sie, wie viele Merkerworte Sie für die Ansteuerung der LEDs des Bediengeräts verwenden wollen. Zu beachten ist hierbei, dass die Eingabe dezimal erfolgt und in 2-er Schritten erwartet wird. Wenn Sie also 1 Merkerwort = 16 LED-Merker (16 Wortbits) parametrieren wollen, geben Sie bitte eine 2 ein. Anmerkung: dies entspricht den LEDs 1-16.

Ein Beispiel hierzu:

Wenn Sie als Basis für Bild/Meldeaufrufe das Merkerwort MW1 (Adresse 400 hex) angegeben haben und die Felder S1 bis S4 wie in der Tabelle ausfüllen, dann sind die Wortbits wie folgt:

Feld/Wert	Merkerwort/Wortbits	Bits entsprechen
S1/2	MW1/ WB1.1 bis WB1.16	LEDs 1 - 16
S2/4	MW2, MW3/ WB2.1 bis WB3.16	Bilder 1 - 32

S4/2	MW4/ WB4.1 bis WB4.16	Prio-Bilder 1 - 16
S3/4	MW5, MW6/ WB5.1 bis WB6.16	Meldungen 1 - 32

Das Bediengerät liest diese Merkerworte zyklisch aus und wertet die Merker einzeln als Aufrufe.

Um nun z.B. ein Bild anzuzeigen, setzen Sie einfach den entsprechenden Merker in Ihrem SPS-Programm (so wie einen Ausgang) - und am Bediengerät erscheint das Bild. Einfacher gehts nicht.

8.3.13 Feld S5

Hier müssen Sie angeben, wieviele Tasten in Merker gespiegelt werden sollen.

Es wird eine Angabe in 2-er Schritten erwartet. Eine 2 heißt, dass 16 Tastenmerker / 1 Merkerwort benutzt werden.

Zusätzlich werden 4 Merkerworte angehängt, in die Sie frei beliebige Daten aus dem Bediengerät einspielen können. Und zwar sind die internen Variablen mit Handles 1000 und 1001 für diese Aufgabe erweitert worden.

Die Werte dieser Variablen erscheinen in den 4 Merkerworten, die auf die für Tasten angegebenen Merkerworte folgen.

Ein Beispiel: Wenn Sie MW32 als Merkerwort für Tasten angegeben haben (Adresse 440 hex) und im Feld S2 eine 2 angeben, erscheinen im MW32 die Tasten als Wortbits, im MW33/MW34 der Wert der internen Variablen mit Handle 1000 und in MW35/MW36 der Wert der internen Variablen mit Handle 1001.

Auf diese Weise haben Sie z.B. bei der Verwendung des Touch-Screens die Möglichkeit, einzelne Bits für einzelne Touch-Tasten zu verwenden.

Hinweis:

Diese Merkerworte werden mit zufälligen Werten beschrieben, wenn die internen Variablen 1000 und 1001 nicht projiziert sind. Also dann keinesfalls verwenden.

Ein anderes Beispiel:

Sie verwenden ein ITS6204 mit insgesamt 32 Tasten. Sie wollen alle Tasten in der SPS als Merker gespiegelt haben.

Sie müssen also 4 Merkerbytes reservieren. Als Merkerbereich wollen Sie WB10.1 bis WB11.16 verwenden.

Geben Sie in der Parametrieremaske die folgenden Einträge ein:

Feld	Eingabe
P4	414

Handbuch Bediengeräte

S5	4
----	---

Beachten Sie, dass dann die MW12/13 und MW14/15 für die Variablenwerte der Variablen 1000 und 1001 reserviert sind.

Jetzt spiegelt das Bediengerät die Tasten in die Merker. Die Merker bedeuten jetzt:

Merker	= Taste Nr.
WB10.1	1 (1. Reihe, links)
WB10.2	2
...	...
WB10.8	8 (1. Reihe, rechts)
WB10.9	9 (2. Reihe, links)
....
WB11.16	32 (4. Reihe, rechts)

Numerikblock beim ITS6100:

Wenn die numerischen Tasten des ITS6100 ebenfalls abgefragt werden sollen, so müssen auf jeden Fall 4 Merkerworte (=64 Merker) für die Tastenabfrage reserviert werden. Die ersten 48 Merker sind immer den Funktionstasten zugeordnet. Der Numerikblock wird ab dem 49. Merker übertragen.

Die Numeriktasten sind dann unter den Tastennummern nach folgender Tabelle zu finden (Beispiel: MW17/adr. 420 hex als Basis):

Taste	Tastennr.	für Basis M0
Escape	50	WB20.2
"4"	51	WB20.3
"6"	52	WB20.4
"2"	53	WB20.5
"8"	54	WB20.6
Enter	55	WB20.7
"0"	57	WB20.9
"1"	58	WB20.10
"3"	59	WB20.11
"5"	60	WB20.12
"7"	61	WB20.13
"9"	62	WB20.14
"."	63	WB20.15
"+/-"	64	WB20.16

Die Merker für die Tasten 49 (WB20.1) und 56 (WB20.8) werden immer mit 0 gesetzt, bitte verwenden Sie diese Merker nicht anderweitig. Die Funktionstasten am ITS6101 haben die Tastennummern 1-8, Wird das ITS6106 verwendet (Maximalausbau), sind die Funktionstasten von 1-48 nummeriert. Deshalb werden die Numeriktasten ab Tastennummer 50 in die SPS gespiegelt.

8.4 Die Status-Datenregister

Über die Status-Datenregister hält das Bediengerät die SPS über Bediener-Aktionen auf dem Laufenden. Es legt ab, welches Bild und welche Meldung gerade zur Anzeige kommen und in welchem Bedienzustand es sich befindet.

Aber über diesen Status-Bereich werden auch weitere Funktionen des Bediengerät angesprochen. Sie haben Zugriff auf den CAN-Bus, der am Bediengerät angeschlossen sein kann. Außerdem können Sie Geräteparameter wie Kontrast und Helligkeit über den Status-Bereich beeinflussen.

Die in der folgenden Beschreibung gemachten Angaben von Merkerworten beziehen sich immer als Offset auf das Basisregister, das Sie im Feld "P2" der Parametriermaske eingetragen haben (Beispiel: Steht im Text MW+2 und Sie haben im Feld "Status" MW10 angegeben, so ist das "echte" Register MW10+2). Die folgende Tabelle gibt Aufschluß über die Verwendung der Datenworte im Status-Bereich:

Datenregister	Funktion
MW+0 bis MW+9	Gerätestatus des Bediengerät
MW+10 bis MW+15	Sendepuffer für CAN-Gateway
MW+16 bis MW+21	Empfangspuffer für CAN-Gateway

8.4.1 Gerätestatus-Informationen

Das Bediengerät informiert die SPS standardmäßig über die folgenden Datenworte mit dem in der Tabelle aufgeführten Inhalt (Bezug Feld P2):

Datenregister	Inhalt des Datenregisters
MW+0	Bildnummer des aktuell angezeigten Bildes
MW+1	Meldenummer der aktuell angezeigten Meldung (0=keine Meldung wird angezeigt)
MW+2	Gerätezustand. Siehe Liste unter TA=0x0A
MW+3	Anzahl der aktiven Bilder
MW+4	Anzahl der aktiven Meldungen
MW+5	Sekunde und Minute, BCD-codiert
MW+6	Stunde und Wochentag, BCD-codiert
MW+7	Tag und Monat, BCD-codiert
MW+8	Jahr 4-stellig, BCD-codiert
MW+9	nicht belegt, reserviert

Handbuch Bediengeräte

8.4.2 CAN-Gateway Sendepuffer

Die Datenworte DW+10 bis DW+15 des Status-DB's sind wie folgt belegt (Bezug Feld P2):

Datenwort	Funktion
MW+10	Handshake.0: Sendepuffer frei-sonst Sendepuffer belegt
MW+11	CAN-Identifizier Hier muß das Anwenderprogramm den Adressaten eintragen. Adresse 0 ist das Bediengerät selbst.
MW+12	CAN-Nutzdaten D0/D1
MW+13	CAN-Nutzdaten D2/D3
MW+14	CAN-Nutzdaten D4/D5
MW+15	CAN-Nutzdaten D6/D7

Handshake über MW+10:

Über das MW+10 erfolgt eine Abstimmung zwischen SPS-Anwenderprogramm und Bediengerät/CAN-Bus. Zuerst muß geprüft werden, ob MW+10=0 ist. Dann werden die Daten in MW+11 bis MW+15 eingetragen und anschließend (!) MW+10 auf 1 gesetzt.

Hiermit läßt sich verhindern, daß die SPS zu schnell Daten an den CAN-Bus (oder das Bediengerät selbst) ausgibt.

Telegrammart und Funktionsworte:

Die Telegrammart und die Funktionsworte sind abhängig vom Adressaten in MW+11:

MW+11=0 Daten für das Bediengerät an der PG-Schnittstelle	MW+11<>0 (Daten sind für den CAN-Bus bestimmt)
MW+12 CAN-Nutzdaten entsprechend Beschreibung CAN-Telegramme	MW+12 KH=bbaa aa = D0 bb = D1
MW+13 CAN-Nutzdaten entsprechend Beschreibung CAN-Telegramme	MW+13 KH=ddcc cc = D2 dd = D3
MW+14 CAN-Nutzdaten entsprechend Beschreibung CAN-Telegramme	MW+14 KH=ffee ee = D4 ff = D5
MW+15 CAN-Nutzdaten entsprechend Beschreibung CAN-Telegramme	MW+15 KH=gghh gg = byte6 hh = byte7

8.4.3 CAN-Gateway Empfangspuffer

Bevor Sie allzu euphorisch auf den CAN-Bus zugreifen wollen: bedenken Sie, daß auf dem CAN-

Bus eine Übertragungsrate von bis zu 1 MBit/s eingestellt werden kann. Auf der PG-Schnittstelle sind fest 9600 Baud eingestellt, von denen man ca. 75% für das Protokoll rechnen muß. Netto bleiben also ca. 2400 baud übrig.

Wenn jetzt ein CAN-Modul nur 10-mal pro Sekunde ein Telegramm absetzt, so müßte 10 Mal pro Sekunde der Empfangspuffer in die SPS geschrieben werden, und nebenher auch noch die Merkerbytes und Variablen ausgelesen werden - unmöglich!

Also wozu das Ganze?

Denken Sie daran, daß z.B. eine Bedientastatur ITS6300 über den CAN-Bus an das Bediengerät angeschlossen werden kann. Irgendwie müssen Sie dann gemeldet bekommen, wenn eine Taste auf dem ITS6300 gedrückt wird - und das geht nur über das Bediengerät. Und ehrlich gesagt: so einfach können Sie keine anderen Tastaturen an die SPS anschließen wie über den CAN-Bus.

Das ITS6300 sendet nun jedesmal eine CAN-Nachricht an das Bediengerät, wenn eine Taste gedrückt wird. Das Bediengerät legt dann dieses Telegramm im Empfangspuffer ab.

Realistisch gesehen wird ein Bediener höchstens 2-3 mal pro Sekunde eine Taste drücken. Das Bediengerät kann Tasteneingaben gegebenenfalls in einem FIFO-Puffer mit 20 Telegrammen Tiefe zwischenspeichern.

Wenn Sie dem Bediener über eine LED signalisieren, daß sein Tastendruck registriert wurde, so wird er nicht anfangen, wie wild auf der Tastatur zu hämmern.

Aber nun zur Beschreibung der Datenregister des Empfangspuffers. Auch der Empfangspuffer hält ein Handshake-Register zur Verfügung, anhand dessen der Datentransfer kontrolliert wird, sowie die Informationsbytes:

Datenregister	Funktion
MW+16	Handshake. 0: Empfangspuffer leer sonst Daten im Puffer
MW+17	CAN-Identifizier Hier erhält das Anwenderprogramm in der SPS die Adresse des Senders. Adresse 0 ist das Bediengerät selbst.
MW+18	CAN-Nutzdaten D0/D1
MW+19	CAN-Nutzdaten D2/D3
MW+20	CAN-Nutzdaten D4/D5
MW21	CAN-Nutzdaten D6/D7

Sendungen vom Bediengerät: MW+17=0

Derzeit sind keine Telegramme vom Bediengerät

Handbuch Bediengeräte

an die SPS definiert. Alle Funktionen werden über Merker und Datenbausteine abgewickelt.

Sendung vom CAN-Bus: MW+17<>0:

In diesem Fall wird das CAN-Telegramm des Senders 1:1 in den Empfangspuffer geschrieben. Dabei werden die Bytes wie folgt abgelegt:

Datenregister	Inhalt
MW+17	CAN-Identifizier (Senderadresse)
MW+18	CAN-Nutzdaten KH=bbaa aa = D0 bb = D1
MW+19	CAN-Nutzdaten KH=ddcc cc = D2 dd = D3
MW+20	CAN-Nutzdaten KH=ffee ee = D4 ff = D5
MW+21	CAN-Nutzdaten KH=hhgg gg = D6 hh = D7

Der Inhalt des CAN-Telegramms ist abhängig vom Gerät, das das Telegramm gesendet hat. Schlagen Sie deshalb im Handbuch zu diesem Gerät nach, wenn Sie den Inhalt der Telegramme bestimmen müssen.

8.4.4 CAN-Identifizier MW+11 und MW+17

Der CAN-Identifizier setzt sich aus insgesamt 16 bit zusammen. Die einzelnen Bits haben folgende Bedeutung:

15-5	4	3-0
Identifizier	R	DLC
x x x x x x x x x x x x	x	1 0 0 0

In den Bits 0-3, DLC (Data Length Code), wird angegeben, wie viele Bytes Nutzdaten das CAN-Telegramm enthält. Dieser Wert kann 0 bis 8 sein. Maximal kann ein CAN-Telegramm 8 Byte Nutzdaten enthalten.


Das RTR-Bit (R) wird derzeit nicht verwendet. Setzen Sie es deshalb auf 0.

Die ID-Bits 0-10 müssen die Nummer des Geräts enthalten. Diese wird meistens über DIP-Schalter oder Jumper gesetzt. Details entnehmen Sie bitte dem Handbuch des jeweiligen Geräts.

9 CAN-Module konfigurieren

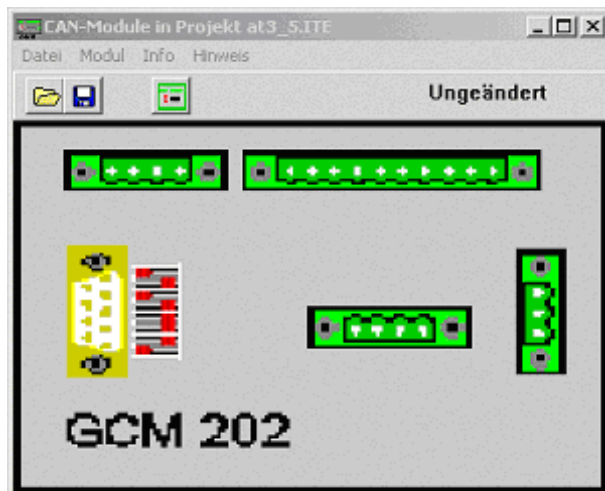
9.1 Starten des Modul-Konfiguratoren

Das Einbinden von CAN-Modulen wird mit einem Konfigurationsprogramm vorgenommen, das Sie aus dem Editor heraus aufrufen können.

Dazu verwenden Sie den Knopf  oder den Menüpunkt "Programme"/"CAN-Module konfigurieren".


Vorher müssen Sie das angefangene Projekt bereits abgespeichert haben.

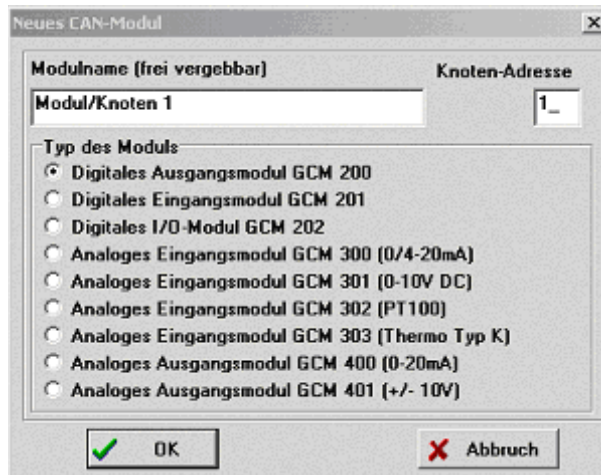
Das Programm meldet sich mit dem folgenden Fenster:



In diesem Programm können Sie nun CAN-Module anlegen, konfigurieren und löschen. Ebenso können Sie Eingangs- und Ausgangsfunktionen definieren. Logische Verknüpfungen können Sie hier nicht vornehmen - verwenden Sie hierzu das Steuerprogramm.

9.2 CAN-Modul anlegen

Nachdem Sie das CAN-Konfigurationsprogramm gestartet haben, werden Sie zunächst einmal noch keine Module haben. Wenn Sie den Menüpunkt "Modul"/"neu" aufrufen oder den Knopf  drücken, erhalten Sie eine Liste der Module, die derzeit bekannt sind:



Es handelt sich hier um die Grundmodul-Maske. Wenn Sie Erweiterungsmodule des Typs GCM 205 oder GCM 206 einsetzen: diese werden später eingestellt. Wählen Sie hier das Grundmodul aus.

9.2.1 Feld "Name des Moduls"

In das Feld "Name des Moduls" dürfen Sie eintragen, was Sie wollen. Lassen Sie aber das Feld nicht leer, sondern verteilen Sie einen sprechenden Namen wie z.B.: "Temperaturmessung Kühlhaus" oder "Meldungserfassung Schaltschrank". Sie finden die Module dann einfacher wieder.

9.2.2 Feld "Knoten-Nr.:"

In dieses Feld tragen Sie die Knotennummer des Moduls ein. Die Knotennummer wird am Modul über die DIP-Schalter 1-5 eingestellt. (Siehe oben)


9.2.3 Feld "Typ des Moduls"

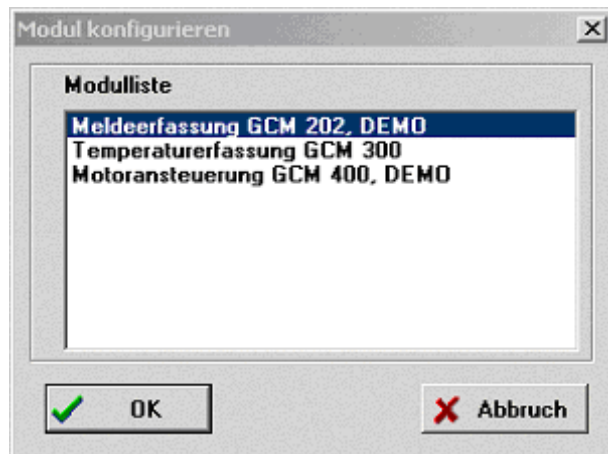
Wählen Sie hier den Typ des Grundmoduls, das Sie an dieser Adresse als Modul einsetzen. Zum Schluss drücken Sie "OK", und das Modul wird in das Projekt übernommen.

9.3 Ein CAN-Modul konfigurieren

Wenn Sie ein Modul konfigurieren wollen, muss es erst über "Modul neu" angelegt werden - siehe vorigen Abschnitt.

Wenn im Projekt ein Modul angelegt ist, dann erscheinen weitere Knöpfe im Editor und der Menüpunkt "Modul konfigurieren" ist wählbar.

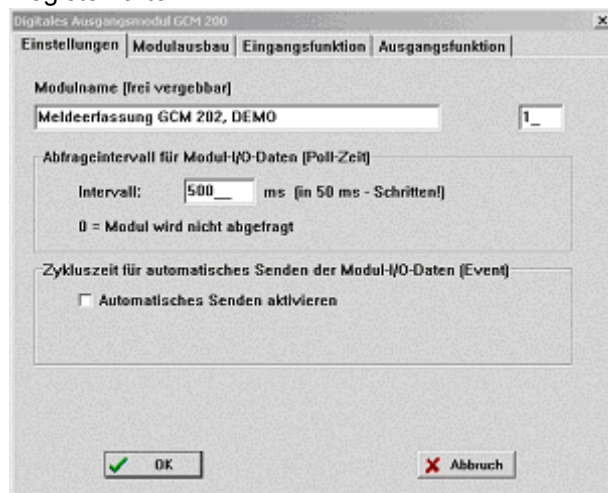
Der Knopf  ruft ebenso wie "Modul konfigurieren" die folgende Auswahlmaske auf:



In dieser Auswahlmaske werden die Namen angezeigt, die Sie beim Anlegen der Module vergeben haben (darum sollten Sie auch einen Namen angeben!). Markieren Sie das Modul, das Sie bearbeiten wollen und drücken Sie "OK". Abhängig vom Modultyp erhalten Sie dann eine Konfigurationsmaske mit unterschiedlichen Möglichkeiten.

9.3.1 Modulserie GCM 200

Hier erhalten Sie eine Konfigurationsmaske mit 4 Registerkarten:



Die Registerkarten haben folgenden Inhalt.

9.3.1.1 Registerkarte "Einstellungen"

Hier werden die Grundeinstellungen des Moduls vorgenommen. Die Felder haben folgende Bedeutung:

Modulname

Den haben Sie beim Anlegen des Moduls schon gesehen. Hier können Sie diesen ggf. korrigieren.

Knoten-Adresse

Hier tragen Sie die Knotennummer (Adresse) des

Moduls ein.

Abfrageintervall...(Poll-Zeit)

Es gibt verschiedene Einstellungen, wann und wie ein Modul seine I/O-Daten auf dem Bus meldet. Mit dieser Poll-Zeit können Sie den Master (also das Bediengerät) anweisen, zyklisch in ms die I/O-Daten vom Modul anzufordern. Hier ist der Master derjenige, der den Datentransfer auslöst. Wenn Sie 0 eingeben, werden die Daten nicht vom Master in zeitlichen Abständen angefordert, sondern das Modul muss seine Daten selber senden. In den meisten Fällen können Sie die Einstellung von 500 ms beibehalten.

Die Poll-Zeit wird auch dazu verwendet, einen Refresh der Ausgänge des Moduls durchzuführen. Die Ausgänge werden zwar immer sofort geschrieben, wenn sie sich ändern. Aber bei einem Stromausfall oder einer Störung durch EMV könnte ein Modul falsche Ausgangsdaten haben. Durch die Poll-Zeit wird aber dann erreicht, dass spätestens nach dieser Zeit wieder alle Ausgänge so sitzen, wie sich das gehört.

Zykluszeit (Event)

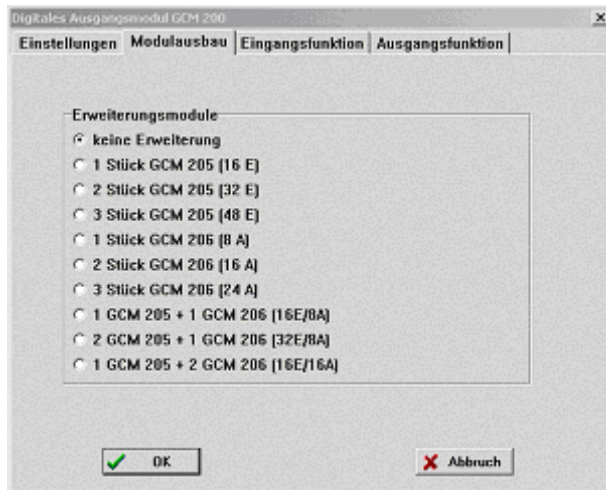
Mit dieser Einstellung bestimmen Sie, ob das Modul selbsttätig seine gesamten Daten in bestimmten zeitlichen Abständen senden soll. Wenn Sie das Feld "Automatisches Senden aktivieren" ankreuzen, können Sie in dem daraufhin erscheinenden Feld "Zykluszeit" den gewünschten zeitlichen Abstand einstellen.

Sie können beide Einstellungen, Poll-Zeit und Zyklus-Zeit gleichzeitig verwenden, dies macht aber nur Sinn, wenn das Modul Ausgänge hat (Ausgangs-Refresh!). Wir empfehlen, nur eine der Möglichkeiten anzuwenden, wobei bei Ausgangsmodulen die Poll-Zeit verwendet werden sollte. Sie können aber jedes Modul unterschiedlich behandeln.

9.3.1.2 Registerkarte "Modulausbau"

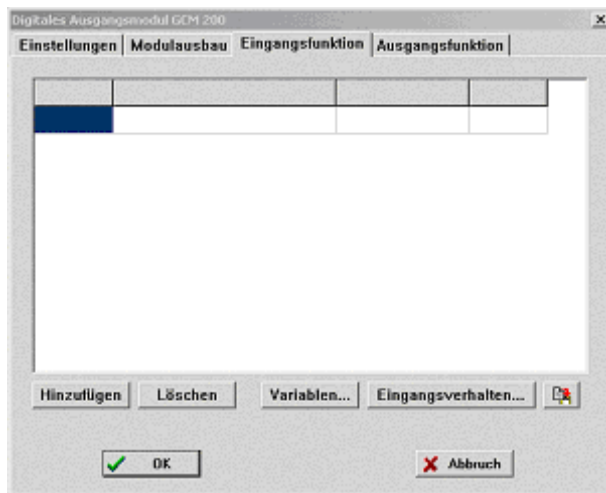
Hier müssen Sie angeben, welche Art von Erweiterungsmodulen und wie viele davon am Grundmodul vorhanden sind. Sie haben hier die komplette Auswahl:

Handbuch Bediengeräte



9.3.1.3 Registerkarte "Eingangsfunktion"

In dieser Registerkarte sehen Sie eine Tabelle, die die Eingangsfunktionen des I/O-Moduls enthält. Sie können jedem Eingang mehrere Funktionen zuweisen. So könnten Sie z.B. eine Meldung aufrufen und eine Variable hochzählen, wenn am Eingang eine positive Flanke eintrifft und die Meldung wieder austragen, wenn eine negative Flanke eintrifft. Diese Funktions-Zuordnungen sind, nach Eingang sortiert, in der Tabelle dieser Registerkarte enthalten. Und so präsentiert sich die Registerkarte:



Die erste Spalte der Tabelle enthält die Eingangsnummer. Die Zählung beginnt bei 0. In der zweiten Spalte wird die Funktion in einer Nummer codiert angezeigt, und Spalte 3 enthält Informationen abhängig von Spalte 2.

Die Nummern bedeuten:

Nr./Funktion	Spalte 2	Bedeutung Spalte 3
0	Meldung aufrufen	Meldenummer
1	Meldung austragen	Meldenummer

2	Bild aufrufen	Bildnummer
3	Prioritätsbild aufrufen	Bildnummer
4	Bild austragen	Bildnummer
5	Variable anzeigen (Eingang wird in Variable übertragen)	Handle
6	Variable inkrementieren	Handle
7	Variable dekrementieren	Handle
8	Variable löschen	Handle

In Spalte 4 wird angegeben, wann die Funktion ausgeführt werden soll:

Spalte 4	Funktion wird ...
P	... bei positiver Flanke ausgeführt
N	... bei negativer Flanke ausgeführt
B	... bei beiden Flanken ausgeführt

Die Angaben in der Tabelle können Sie direkt in der Tabelle oder durch Doppelklick in die jeweilige Zeile ändern. Sie erhalten dann eine Einstellmaske für die jeweilige Funktion (siehe "Mögliche Eingangsfunktionen" weiter unten).

Knopf "Hinzufügen"

Mit diesem Knopf können Sie eine Eingabemaske aufrufen, mit der Sie die Eingangsfunktion interaktiv festlegen können. Siehe "Mögliche Eingangsfunktionen"

Knopf "Löschen"

Mit diesem Knopf können Sie die aktuell in der Tabelle markierte Funktionszeile löschen.

Knopf "Variablen"

Dieser ruft die Tabelle zur Bearbeitung der Variablen auf. Diese wird im Kapitel "Variablen" näher beschrieben.

Handbuch Bediengeräte

Knopf "Eingangsverhalten"

Mit diesem Knopf erhalten Sie eine neue Einstellmaske, die das Modul mit Ein- und Ausgängen zeigt:



Für jeden Eingang gibt es 3 Optionen:

Filter (erscheint manchmal nur als F)

Es werden nur Signale gemeldet, die länger als 6 ms anstehen (Entprellung)

P (positive Flanke)

Wenn eine positive Flanke erkannt wird, sendet das Modul sofort eine Datennachricht auf den Bus, unabhängig von den Einstellungen "Poll-Zeit" und "Zyklus-Zeit" im Register "Einstellungen". Es werden immer sämtliche Eingänge übertragen.

N (negative Flanke)

Wenn eine negative Flanke erkannt wird, sendet das Modul sofort eine Datennachricht auf den Bus, unabhängig von den Einstellungen "Poll-Zeit" und "Zyklus-Zeit" im Register "Einstellungen". Es werden immer sämtliche Eingänge übertragen.

Wir empfehlen, dass Sie bei den verwendeten Eingängen alle 3 Optionen aktivieren. So erhalten Sie die schnellstmögliche Signalübertragung.

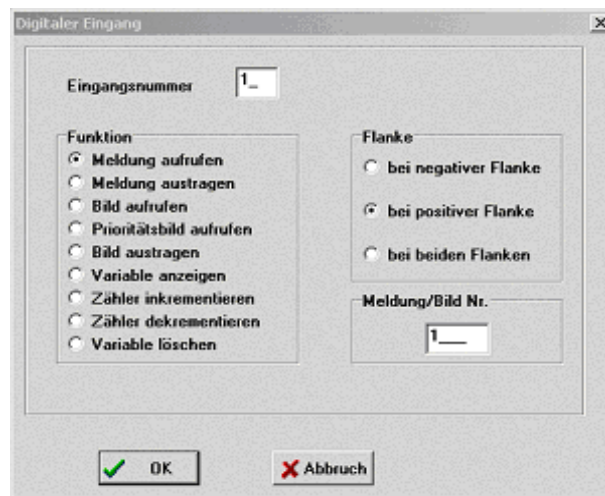
Knopf "Eingangsfunktion kopieren"

Stellen Sie sich vor, Sie haben ein Eingangsmodul mit 40 Eingängen. Auf jeden Eingang wollen Sie nun die Funktionen "Meldung aufrufen bei positiver Flanke" und "Meldung austragen bei negativer Flanke" legen. Das gibt 80 Funktionen. Dieser Knopf ermöglicht es, alle auf einem Eingang definierten Funktionen auf andere Eingänge zu übertragen; und zwar wird dabei auch automatisch die Nummer der Meldung (des Bildes oder des Variablenhandles) für die Kopie erhöht. Das erleichtert die Arbeit bei umfangreichen Projekten.

Mögliche Eingangsfunktionen

Wenn Sie in die Tabelle doppelklicken oder den Knopf "Hinzufügen" drücken, erhalten Sie die Eingabemaske für die Eingangsfunktion. In dieser

Maske können Sie komfortabel eine Funktion für einen Eingang festlegen. Sie können auch mehrere Funktionen pro Eingang festlegen, dies muss aber nacheinander erfolgen. Die Maske hat folgende Einstellmöglichkeiten:



Wählen Sie zunächst die Eingangsnummer im Feld "Eingangsnummer". Im Feld "Funktion" stellen Sie die Funktion ein. Abhängig von der Funktion können sich die Felder "Flanke" und "Meldung/Bild-Nr." ändern. Das sehen Sie bei der Benutzung; diese Funktion unterstützt Sie bei der Eingabe.

Nummerierung der Eingänge

Gezählt wird beginnend bei "0" (!), wobei "0" der erste Eingang am ersten Modulteil beginnend von links ist.

Weiter geht es dann bei den Erweiterungsmodulen, wo bei jedem Modul zunächst die obere, dann die untere Klemme fortlaufend gezählt wird.

Beispiel:

Bei einem GCM 201 mit einem GCM 205 haben Sie die Eingänge 0-7 am Grundmodul GCM 201, die Eingänge 8-15 an der oberen Klemme des GCM 205 und die Eingänge 16-23 an der unteren Klemme des GCM 205. Die Nummerierung ist jeweils von links nach rechts.

Zur Beachtung

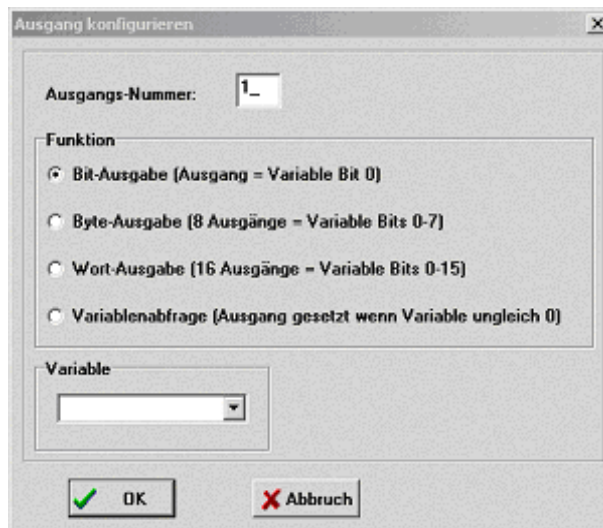
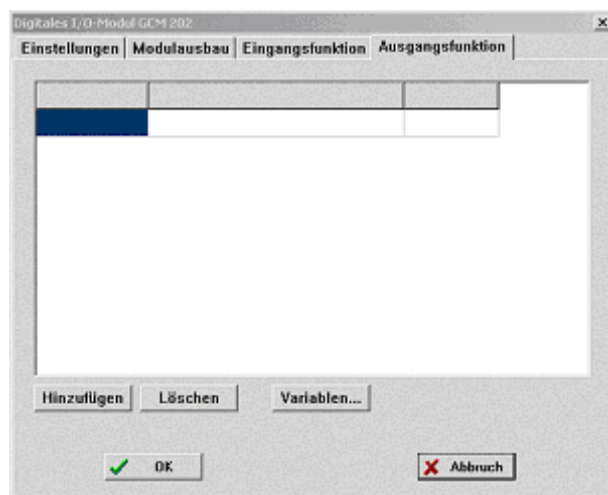
Wenn Sie erreichen wollen, dass ein Bild oder eine Meldung so lange aktiv sein soll, wie der Eingang ansteht, dann müssen Sie zwei Funktionen parametrieren: Eine, die bei positiver Flanke die Meldung oder das Bild aufruft und bei negativer Flanke wieder austrägt.

9.3.1.4 Karte "Ausgangsfunktionen"

In dieser Registerkarte werden die Funktionen tabellarisch angezeigt, die den Ausgängen des Mo-

Handbuch Bediengeräte

duls zugeordnet sind:



In der ersten Spalte wird die Ausgangsnummer angezeigt. Die Spalte 2 enthält die Funktionsnummer und Spalte 3 den Variablenhandle.

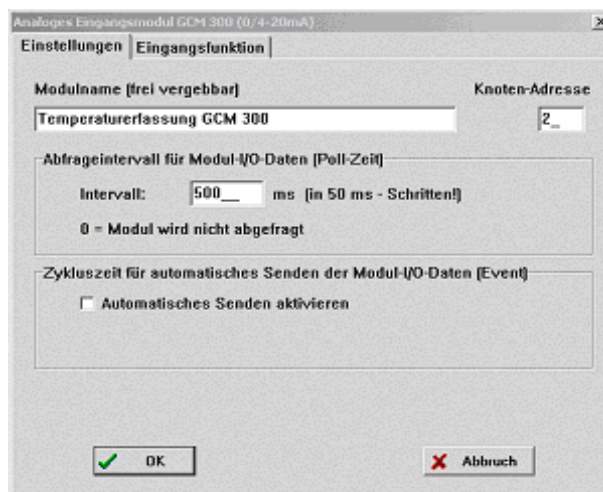
Als Funktionen stehen zur Verfügung:

Nr.	Funktion
0	Das Bit Nr. 0 der angegebenen Variablen wird in den Ausgang kopiert. Das heißt, wenn die Variable einen ungeraden Wert hat, ist der Ausgang eingeschaltet.
1	Es werden die niedrigstwertigen 8 Bit einer Variablen in 8 Ausgänge übertragen. Verwenden Sie für diese Funktion bitte nur die Ausgangsnummern 0, 8, 16 und 24
2	Es werden die niedrigstwertigen 16 Bit einer Variablen in die 16 Ausgänge übertragen. Verwenden Sie für diese Funktion bitte nur die Ausgangsnummern 0, 8 und 16
3	Der Ausgang ist eingeschaltet, solange die Variable ungleich 0 ist

Mit dem Knopf "Hinzufügen" oder durch Doppelklick auf eine Tabellenzeile erreichen Sie die Eingabemaske für Ausgangsfunktionen:

9.3.2 Modulerie GCM 300

Diese Modulerie kann 4 gleichartige Eingänge pro Modul bearbeiten. Wenn Sie ein solches Modul konfigurieren, geschieht dies über die folgende Maske:



9.3.2.1 Registerkarte "Einstellungen"

Diese Einstellungen sind gleich wie bei den digitalen E/A-Modulen der Serie GCM 200. Schlagen Sie weiter oben in diesem Kapitel nach.

9.3.2.2 Karte "Eingangsfunktionen"

Hier werden tabellarisch die von Ihnen erstellten Eingangsfunktionen aufgelistet. Mit einem Doppelklick in eine Funktionszeile in der Tabelle erhalten Sie die Eingangs-Parametrieremaske "Analoger Eingang". Viel Auswahl haben Sie allerdings nicht; es gibt nur wenige Optionen. Als Eingangsfunktion steht nur "Eingang in Variable übertragen" zur Verfügung; sie können nun nur noch angeben, auf welche Variable der Eingangswert

Handbuch Bediengeräte

übertragen werden soll. Eine Skalierung müssen Sie im KOP erledigen, das können Sie hier nicht machen. Auch Grenzwertüberwachung kann ausschließlich im Steuerprogramm erfolgen. Die Eingänge sind an der Eingangsklemme von links nach rechts nummeriert von 0-3.

Knopf "Hinzufügen"

Mit diesem Knopf fügen Sie eine weitere Eingangsfunktion hinzu. Sie erhalten die weiter unten gezeigte Eingangs-Parametrieremaske "Analoger Eingang".

Knopf "Löschen"

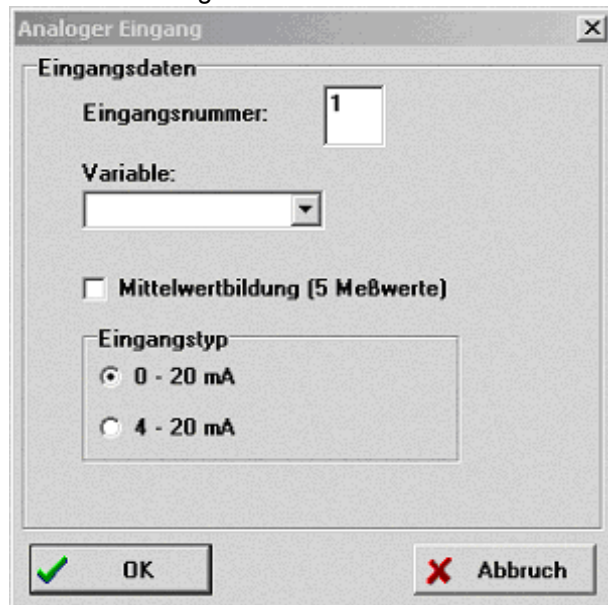
Die markierte Funktionszeile in der Tabelle wird gelöscht.

Knopf "Variablen"

Dieser Knopf ruft die allseits bekannte und im Kapitel 7 beschriebene Variablen-tabelle auf.

Parametrieremaske "Analoger Eingang"

Die Konfigurationsmaske hat - abhängig vom Typ des Moduls - folgendes Bild:



Je nach Art des Eingangsmoduls können Sie wählen, ob Sie am jeweiligen Eingang eine automatische Mittelwertbildung über die letzten 5 Messwerte aktivieren wollen und ob Sie einen 0-20 mA oder 4 - 20 mA Geber anschließen wollen. Eine Skalierung der Eingangswerte muß über den KOP erfolgen. Hierzu können Sie die folgenden Angaben heranziehen:

Wertebereich der Eingänge

Module der Serie **GCM300** liefern als Variablenwert den folgenden Bereich:

0 = 0 μ A; 1 = 5 μ A; ...; 4000 = 20 mA
Also: $I_{in} = \text{Variablenwert} \times 5 \mu\text{A}$

Module der Serie **GCM301** liefern als Variablenwert den folgenden Bereich:

0 = 0 mV; 1 = 2,5 mV; ...; 4000 = 10 V
Also: $V_{in} = \text{Variablenwert} \times 2,5 \text{ mV}$

Module der Serie **GCM302** liefern als Variablenwert den folgenden Bereich:

0 = -50°C; 1=-49,75°C; ...; 4000 = 950°C
Also: $T_{in} = \text{Variablenwert} \times 0,25^\circ\text{C} - 50^\circ\text{C}$

Module der Serie **GCM303** liefern als Variablenwert

1600 = 0°C; 1601 = 0,5°C; ...; 3600 = 1000°C
Also: $T_{in} = (\text{Variablenwert} - 1600) \times 0,5^\circ\text{C}$

9.3.2.3 Modulserie GCM 400

Bei der Modulserie GCM 400 haben Sie jeweils 4 gleichartige analoge Ausgänge 0-20 mA oder 0-10V. Diese Module werden über die folgende Maske konfiguriert:



Die Ausgänge des Moduls werden immer dann geschrieben, wenn sie sich ändern. Über die Poll-Zeit können Sie aber zusätzlich bestimmen, wie oft die Ausgänge unabhängig von Änderungen geschrieben werden sollen.

9.3.2.4 Registerkarte "Einstellungen"

Die Einstellungen in dieser Registerkarte sind identisch wie bei den Modulen GCM 200 und GCM 300. Aber es gibt noch Besonderheiten:

Poll-Zeit:

Die Poll-Zeit wird hier genutzt als Ausgangs-Refresh-Intervall. Das heißt, diese Zeit gibt an, in welchen Abständen die Ausgangswerte des Moduls

Handbuch Bediengeräte

vom Master an das Ausgangsmodul gesendet werden. Das Ausgangsmodul sendet als Bestätigung die Ausgangswerte als Datentelegramm, sodass ein Bediengeräte-Slave die Ausgangswerte ebenfalls anzeigen kann.

Zykluszeit

Einzig wenn ein Bediengerät als Slave angeschlossen ist, können die Ausgangsdaten interessant sein und dann über die Zykluszeit laufend aktualisiert werden.

9.3.2.5 Karte "Ausgangsfunktionen"

In dieser Registerkarte werden die Ausgangsfunktionen tabellarisch aufgelistet. Normalerweise haben Sie pro Ausgang eine Funktion; nämlich die Ausgabe eines Variablenwerts an den Ausgang. In der ersten Spalte steht die Ausgangsnummer, in der zweiten die Funktionsnummer (hier immer 2, das bedeutet Ausgabe der niederwertigen 16 Bit einer Variablen) und in der dritten das Handle der Variablen. Ein Doppelklick auf eine Tabellenzeile öffnet das Konfigurationsfenster für die entsprechende Zeile (siehe unten)

Knopf "Hinzufügen"

Wenn Sie diesen Knopf drücken, fügen Sie eine neue Ausgangsfunktion hinzu. Sie erhalten die Maske "analoger Ausgang".

Knopf "Löschen"

Die in der Tabelle markierte Funktionszeile wird gelöscht (und damit die Zuweisung einer Variablen auf den Ausgang).

Knopf "Variablen"

Hier landen Sie wieder in der bekannten Variablen-tabelle

Maske "Analoger Ausgang"

Hier können Sie einem Ausgang eine Variable zu-

ordnen:

Eine Skalierung der Daten muß mit KOP erfolgen. Die hierzu notwendigen Informationen sind:

Wertebereich der Ausgänge

Module der Serie **GCM400** benötigen als Variablenwert den folgenden Bereich:

0 = 0 μ A; 1 = 5 μ A; ...; 4000 = 20 mA


Also: **Iout = Variablenwert x 5 μ A**

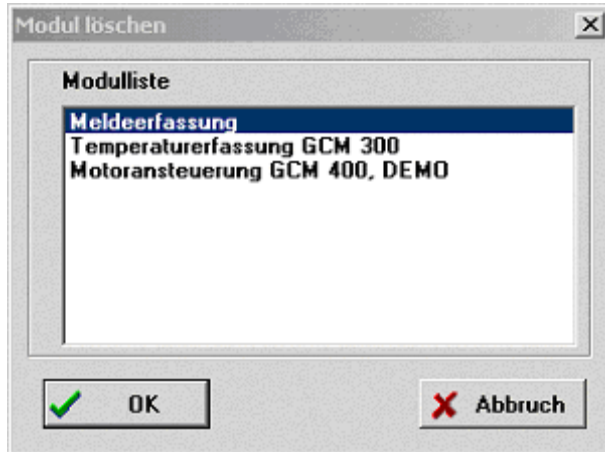
Module der Serie **GCM401** benötigen als Variablenwert den folgenden Bereich:

48 = -10 V; ...; 2048 = 0V; ...; 4048 = 10 V

Also: **Vout = (Variablenwert-2048)x2,5 mV-10,24V**

9.4 CAN-Modul entfernen


Wenn Sie ein Modul entfernen wollen, so benutzen Sie den Knopf  oder den Menüeintrag "Modul löschen". Markieren Sie in der nun erscheinenden Auswahlliste das Modul

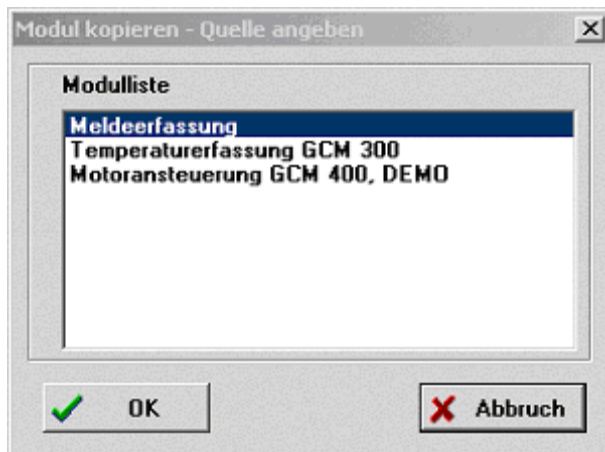


und drücken Sie "OK".

9.5 Modul kopieren

Wenn Sie mehrere Module haben, bei denen Sie die gleiche Funktion erstellen wollen (z.B. Meldeaufrufe oder Bildaufrufe), dann können Sie ein Modul komplett kopieren.

Wählen Sie hierzu den Knopf  oder den Menüpunkt "Modul kopieren". Daraufhin erscheint eine Liste, aus der Sie das zu kopierende Modul auswählen können:



Markieren Sie das gewünschte Modul, und klicken Sie auf "OK". Daraufhin wird eine weitere Maske angezeigt. Diese entspricht der Konfigurationsmaske für das angegebene Modul. Hier sind in den Tabellen bereits die I/O-Funktionen des Quellmoduls angegeben und das Eingangsverhalten wurde auch übernommen. Sie müssen jetzt nur noch Bild bzw. Meldenummern und Variablen anpassen.