CAN-CBM-OP5

CANopen Terminal mit Jog-Shuttle

Hardware-Handbuch

Dokument-Datei:	M:\texte\Doku\MANUALS\CAN\Cbm\OP5\can-cbm-op5_10h.ma9
Datum des Ausdrucks:	05.05.2003

Bestell-Nr.:	CAN-CBM-OP5 C.2120.05

Änderungen in den Kapiteln

Die hier aufgeführten Änderungen im Dokument betreffen sowohl Änderungen in der Hardware als auch reine Änderungen in der Beschreibung der Sachverhalte.

Kapitel	Änderungen gegenüber Vorversion
-	Erste Ausgabe

Weitere technische Änderungen vorbehalten.

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. **esd** übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei **esd**. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch **esd** gestattet.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover

Tel.: 0511/372 98-0
FAX : 0511/372 98-68
E-Mail: info@esd-electronics.com
Internet: www.esd-electronics.com

Inhalt

1. Übersicht 3
1.1 Zu diesem Handbuch 3
1.2 Aufbau 3

2. Das Gehäuse 5
2.1 Aufbau und Grundgehäuse 5
2.2 Frontplatte 5
2.3 Ansicht der Rückwand mit den Steckern 6
2.4 Einbaumaße 7
2.5 Technische Daten des Gehäuses 8

3. LCD-Display-Modul CAN-LCT 9
3.1 Übersicht 9
3.1.1 Funktionsbeschreibung 9
3.1.2 Allgemeine Technische Daten 10
3.1.3 Netzteil-Baugruppe 10
3.1.4 Microcontroller 11
3.1.5 I²C-Interface 11
3.1.6 Tastatur-Schnittstellen 12
3.1.6.1 MF2-Tastatur-Interface (Option) 12
3.1.6.2 Schnittstelle für Matrix-Tastatur und Jog-Shuttle 12
3.1.7 LC-Displays-Interface 13
3.1.7.1 Interface für grafikfähige LC-Displays 13
3.1.7.2 Interface für alphanumerische LC-Displays (Option) 13
3.1.7.3 Power-Modul für die LCD-Hintergrundbeleuchtung 13
3.1.7.4 Bewegungsmelder-Interface (Option) 14
3.1.8 CAN-Interfaces 14
3.1.8.1 CAN-Interface auf DSUB9 (CAN0) 14
3.1.8.2 CAN-Interface und Programmierung (CAN1) 15
3.1.8.3 Kodierschalter 15
3.2 Platinenansicht 16
3.3 Beschreibung der Baugruppen 17
3.3.1 CAN-Bus-Interfaces 17
3.3.1.1 CAN-Netz CAN0 17
3.3.1.2 CAN-Netz CAN1 und Programmierung 17
3.3.2 MF2-Tastatur (Option) 18
3.3.3 Interface für Grafikfähiges LC-Display 18
3.3.4 Hintergrundbeleuchtung für LC-Displays 18
3.4 Interne Steckverbinder auf der CAN-LCT-Platine 19
3.4.1 Lokaler CAN-Bus und Programmierung (X100) 19
3.4.2 MF2-Keyboard auf Pfostenstecker X200 (Option) 20
3.4.3 Matrix-Tastatur (X201) 21
3.4.4 Bewegungsmelder und LED auf separater Platine an X202 (Option) 22
3.4.5 Grafisches LC-Display (X300) 22
3.4.6 Optionales alphanumerisches LC-Display an X301 (Option) 23
3.4.7 I²C-Interface (X302) 23
3.4.8 Spannungsversorgung des LCD-Power-Moduls an X303 (Option) 24

3.4.9 LCD-Hintergrundbeleuchtung (X304) 24

4. Tastatur 25
4.1 Übersicht 25
4.1.1 Funktionsbeschreibung 25
4.1.2 Allgemeine Technische Daten 26
4.1.3 Tasten 26
4.1.4 Drehgeber (Jog-Shuttle) 26
4.1.5 Platinenansicht mit Zuordnung der Scan- und Return-Lines 27
4.1.6 I²C-Port-Baustein 28
4.1.7 Scan-Lines und Return-Lines 29
4.2 Interne Steckverbinder auf der Tastaturplatine 30
4.2.1 Tastatur und Drehgeber (X100) 30
4.2.2 I²C-Interface (X200) 31
4.2.3 Optionale LCD-Hintergrundbeleuchtung (X210) 32

5. Anhang 33
5.1 Externe Steckverbinder 33
5.1.1 Ansicht der Geräterückwand 33
5.1.2 Spannungsversorgung (X500) 34
5.1.3 CAN-Interface (X600) 35

1. Übersicht

1.1 Zu diesem Handbuch

Die Gliederung der Dokumentation des CAN-CBM-OP5 orientiert sich an den wesentlichen Baugruppen des Gerätes.

1.2 Aufbau

Das CAN-CBM-OP5 besteht aus

- einem Einbau-Gehäuse, in dessen Frontplatte ein

- LC-Display und die Tastaturplatine *CAN-PTAS* (siehe Kapitel 4) eingelassen ist,

- und der Controller-Platine *CAN-LCT* (siehe Kapitel 3).

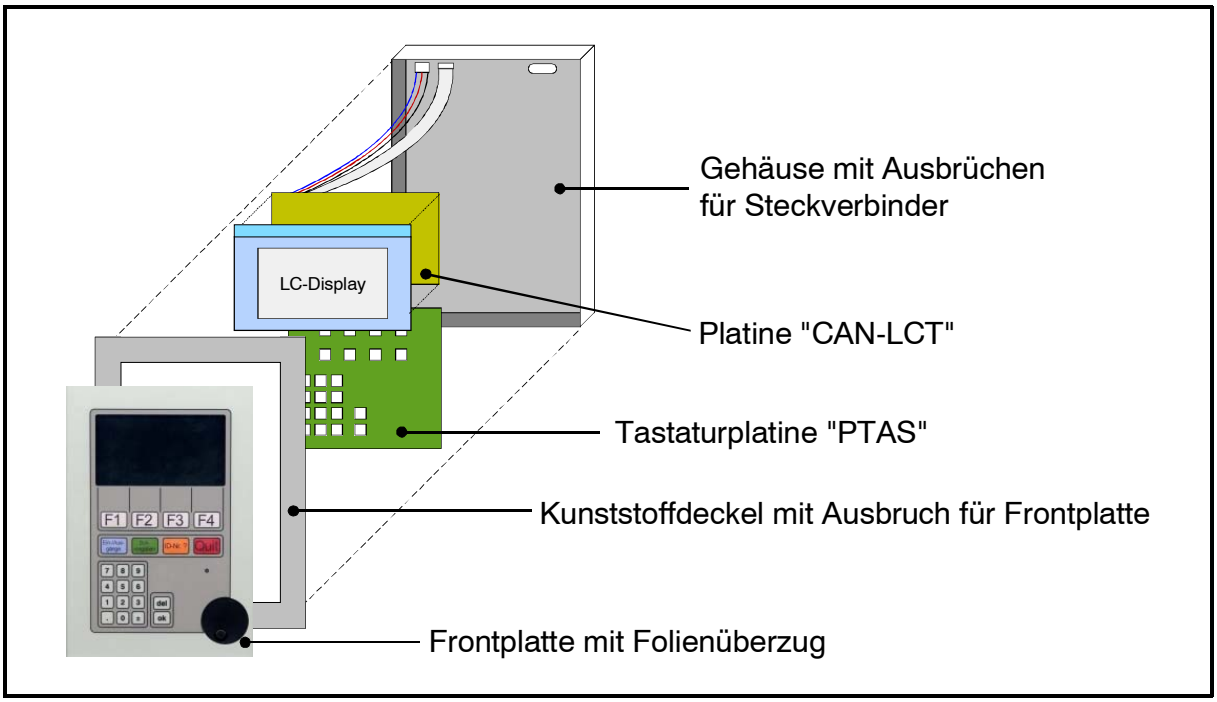


Abb. 1: Anordnung der Baugruppen im CAN-CBM-OP5-Gehäuse

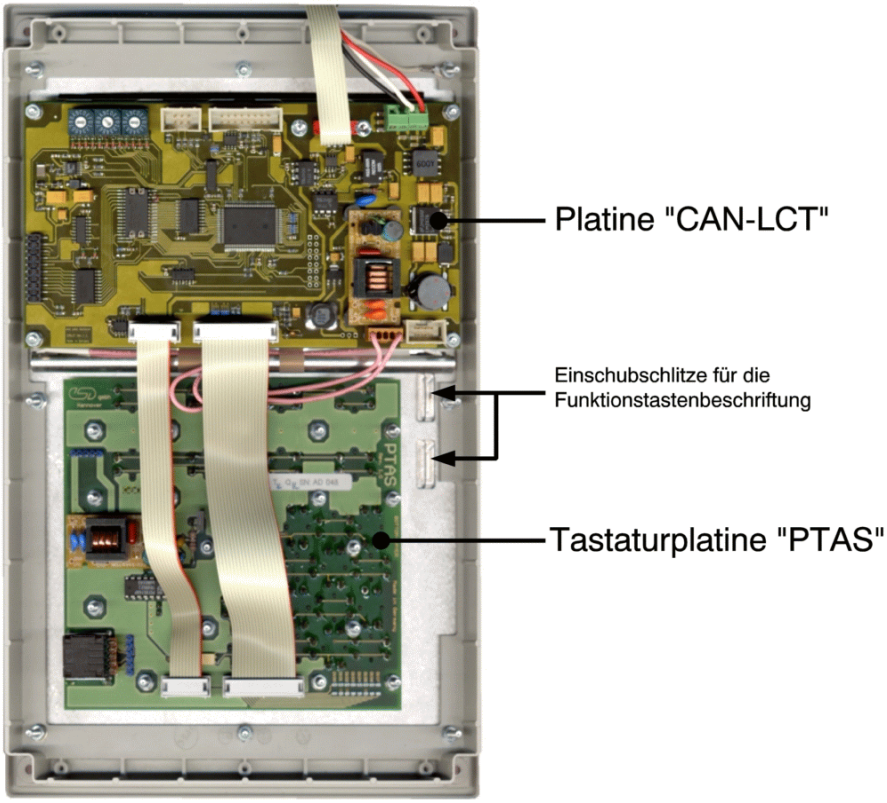


Abb. 2: Ansicht der Frontplatte von Hinten mit montierten Platinen

2. Das Gehäuse

2.1 Aufbau und Grundgehäuse

Das CAN-CBM-OP5 besteht aus einem Einbau-Gehäuse, das für die Installation in Schalttafeln konzipiert ist.

Die Seitenwände und die Rückwand sind aus punktverschweißtem lackiertem Stahlblech gefertigt. An der Rückwand sind Ausbrüche für die Steckverbinder und die Kodierschalter angebracht. Der Frontdeckel besteht aus Kunststoff (Polystyrol) und einer folienüberzogenen Aluminiumplatte. Hinter der Aluminiumplatte sind das LC-Display und die Tastaturplatine verschraubt.

2.2 Frontplatte

Die Folie ist im Bereich des LC-Displays und über den Tasten transparent gehalten. Von der Rückseite des Aluminiumdeckels lassen sich Papier- oder Folienstreifen zwischen die Tasten und die Frontfolie schieben (siehe Abb. 2). Dadurch ist für jedes Gerät eine individuelle Beschriftung der Tasten möglich, die sich auch später noch zerstörungsfrei ändern läßt.

Abb. 3: Ansicht der Frontplatte

2.3 Ansicht der Rückwand mit den Steckern

Von der Rückwand aus sind die Kodierschalter und die Steckverbinder zugänglich. Die Belegung der Steckverbinder kann dem Anhang ab Seite 33 entnommen werden.

Abb. 4: Ansicht der Geräterückwand

2.4 Einbaumaße

Die Frontplatte wird auf das Metallgehäuse aufgesetzt und das Gehäuse wird mit sechs Schrauben, die von Hinten in die Frontplatte geschraubt werden, befestigt. Das so verschlossene Gerät kann jetzt in einem Einbaurahmen oder einer Schalttafel befestigt werden.

Die Befestigung des kompletten Gehäuses erfolgt über vier Schrauben, die von Vorne durch die Kunststoff-Frontplatte in die Schalttafel geschraubt werden. Nach dem Festziehen der Schrauben können die vier Löcher in der Frontplatte mit den mitgelieferten quadratischen Kunststoffkappen abgedeckt werden.

Bei der Bemessung der Einbautiefe ist zu beachten, daß die Steckverbinder und die daran angeschlossenen Kabel nach Hinten herausragen.

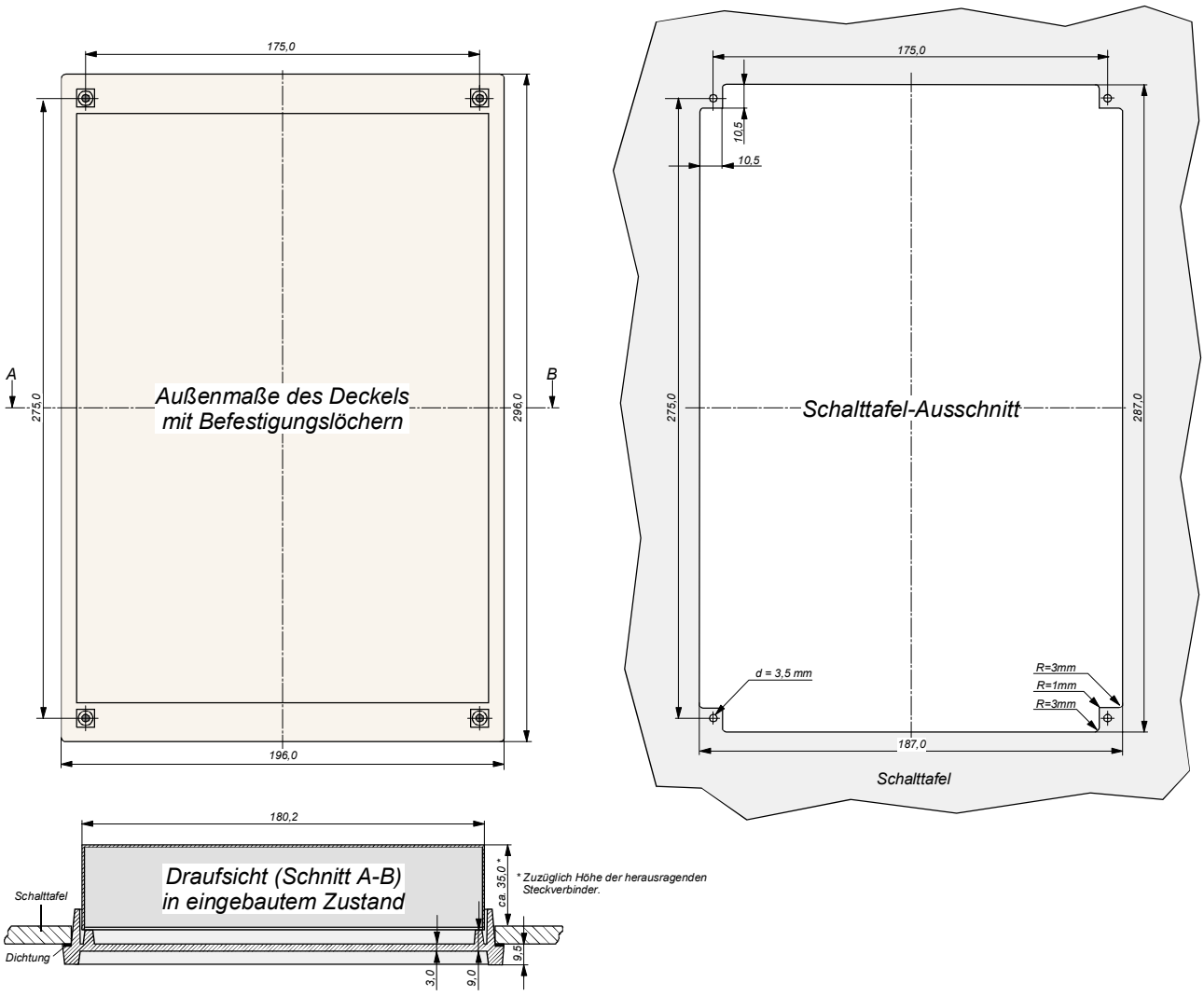


Abb. 5: Einbaumaße des CAN-CBM-OP5

2.5 Technische Daten des Gehäuses

Maße	196 x 296 x 50 mm
Material	Deckel (Befestigungsfrontrahmen): Polystyrol, grau Frontplatte: Aluminium, Folienüberzug Geräterückwandgehäuse: Stahlblech, grau lackiert
Gewicht	ca. 1550 g (komplett montiert mit allen Baugruppen)

Tabelle 1: Allgemeine Daten des Gehäuses

3. LCD-Display-Modul CAN-LCT

3.1 Übersicht

3.1.1 Funktionsbeschreibung

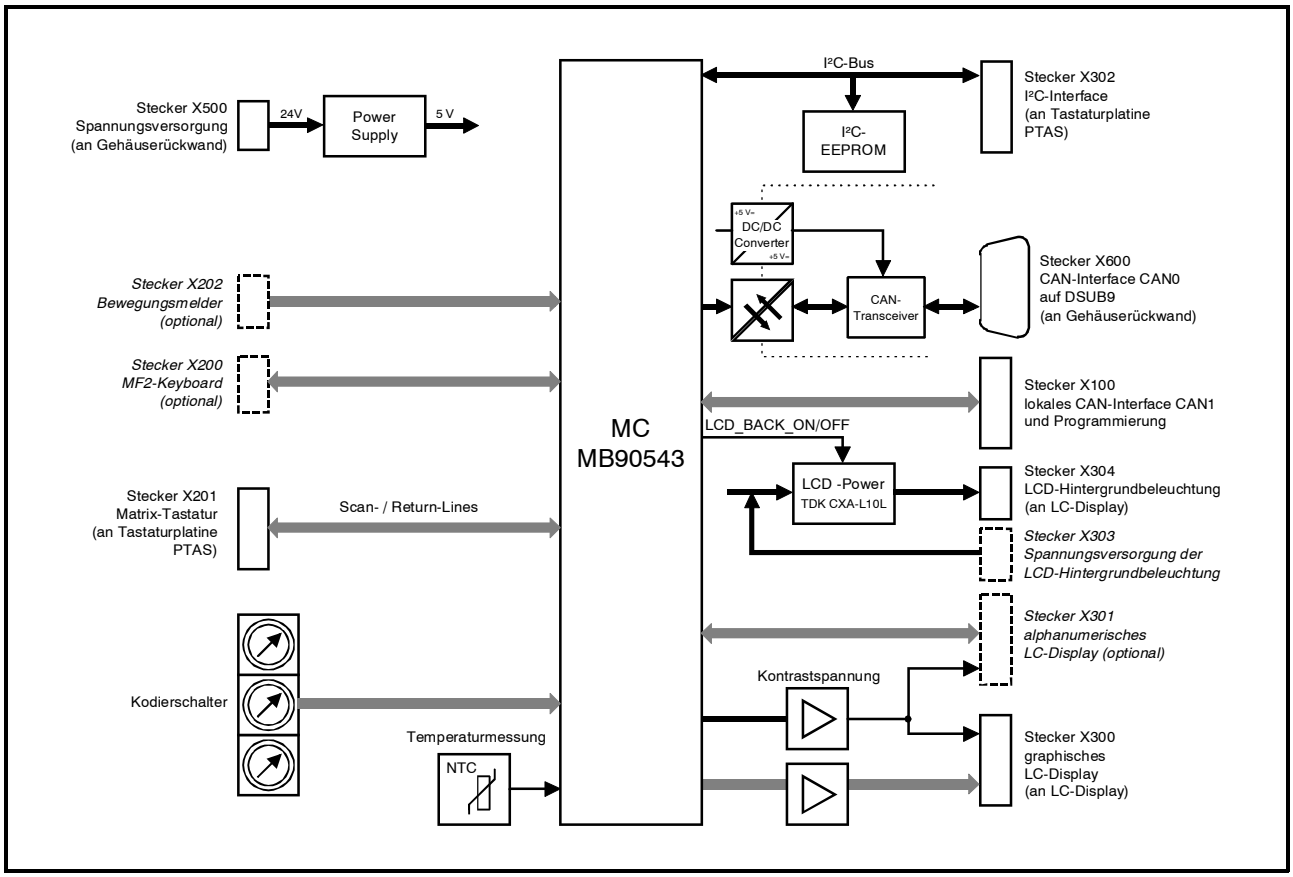


Abb. 6: Blockschaltbild der CAN-LCT-Platine

3.1.2 Allgemeine Technische Daten

Temperaturbereich	max. zulässige Umgebungstemperatur (d. h. innerhalb des Gehäuses): 0...50 ˚C
Steckverbinder	X100 - 16pol. Pfostenstecker (local CAN und Programmierung) *X200 - 6pol. Pfostenstecker (MF2-Tastatur/Option)* X201 - 20pol. Pfostenstecker (Matrix-Tastatur, an Tastaturplatine) *X202 - 6pol. Pfostenstecker (Bewegungsmelder/Option)* X300 - 20pol. Leiterplattenverbinder (grafisches LC-Display) *X301 - 16pol. Pfostenstecker (alphanumerisches LC-Display/Option)* X302 - 10pol. Pfostenstecker (I²C-Bus, an Tastaturplatine) *X303 - 3pol. Pfostenreihe (Spannungsversorgung des LCD-Power-Moduls/Option)* X304 - 5pol. Pfostenreihe (LCD-Hintergrundbeleuchtung) X500 - 3pol. Mini-Combicon (Spannungsversorgung) X600 - 9pol. DSUB (CAN-Interface)
Luftfeuchtigkeit	max. 90%, nicht kondensierend
Größe der Platine	170,2 mm x 84,1 mm

Tabelle 2: Allgemeine Daten des CAN-LCT-Moduls

3.1.3 Netzteil-Baugruppe

Eingangsspannung	24 VDC ± 10 %
Filter	Eingangsfilter, kein Verpolungsschutz
Anschluß	X500, 3pol. Mini-Combicon-Buchse

Tabelle 3: Netzteil-Baugruppe

3.1.4 Microcontroller

Typ	MB90543, 16 MHz
Speicher	intern: 6 k x 8 bit SRAM, 128 k x 8 bit Flash EPROM extern: serielles EEPROM (I²C)
Funktionen	u. a. 81 Ports, 2x 16 Bit Timer/Counter, 2-Channel UART, 8-Bit Serial I/O, CAN, Programm/Pulse Generation, 10-Bit A/D-Converter

Tabelle 4: Microcontroller

3.1.5 I²C-Interface

Lokale Bauelemente mit I²C-Interface	serielles EEPROM für Microcontroller (z.B. CAT24C04)
Anschluß externer I²C-Geräte	X302, 10pol. Pfostenstecker; Stecker ist neben I²C-Signalen zusätzlich mit einem IRQ-Eingang belegt

Tabelle 5: I²C-Interface

3.1.6 Tastatur-Schnittstellen

3.1.6.1 MF2-Tastatur-Interface (Option)

Kompatibilität	Signale AT-kompatibel
Signale	Daten, Takt, Reset und Power
Anschluß	X200, 6pol. Pfostenstecker

Tabelle 6: MF2-Tastatur (Keyboard)

3.1.6.2 Schnittstelle für Matrix-Tastatur und Jog-Shuttle

Anzahl Tasten	max. 8x8 Tasten plus Anschluß eines optischen Drehgebers (Jog-Shuttle)
Anschluß	X201, 20pol. Pfostenstecker

Tabelle 7: Matrix-Tastatur und optischen Drehgeber

3.1.7 LC-Displays-Interface

3.1.7.1 Interface für grafikfähige LC-Displays

Geeignete Displays	TLX-Displays, mit Controller T6963C (z.B. TLX-1741 mit 240 x 128 Pixel),
Kontrast	einstellbar per Software und automatische Kontrastkorrektur über Temperaturfühler (NTC)
Anschluß	X300, 20pol. Platinenverbinder oder 20pol. Pfostenstecker

Tabelle 8: Grafikfähige LC-Displays

3.1.7.2 Interface für alphanumerische LC-Displays (Option)

Kontrast	einstellbar per Software und automatische Kontrastkorrektur über Temperaturfühler (NTC)
Anschluß	X301, 16pol. Pfostenstecker

Tabelle 9: Alphanumerische LC-Displays

3.1.7.3 Power-Modul für die LCD-Hintergrundbeleuchtung

Modul	TDK CXA-L10L oder vergleichbare für CCFL-Backlight
Eingangsspannung	U_{VCCB} = 5V
Spannungszuführung des Power-Moduls	- über lokale +5V-Versorgung der CAN-LCT-Platine oder - optional von außen über Stecker X303 oder - über den Stecker des alphanumerischen Displays X301
Anschluß	Spannungsausgang (an Beleuchtung): X304, 5pol. Stiftleiste Zuführung: X303, 3pol. Stiftleiste (sofern nicht lokal oder von X301),

Tabelle 10: Power-Modul für Hintergrundbeleuchtung

3.1.7.4 Bewegungsmelder-Interface (Option)

Bewegungsmelder	externe Platine mit Bewegungsmelder und Anzeige-LED lieferbar
Daten der Bewegungsmelder-Platine	Sensor-Typ: Passiver Infrarot-Sensor NAIS AMN11111 Reichweite: 5 m im Bereich von ± 40˚ LED: zweifarbige LED rot/grün zur Anzeige der Aktivierung
Auswertung	automatisches Ein/Ausschalten der LC-Display-Hintergrundbeleuchtung bei Annäherung einer Person
Anschluß	X202, 6pol. Pfostenstecker

Tabelle 11: Bewegungsmelder

3.1.8 CAN-Interfaces

3.1.8.1 CAN-Interface auf DSUB9 (CAN0)

CAN-Controller	MB90543
CAN-Protokoll	CAN 2.0A/2.0B
Übertragungsrate	10 kBit/s ... 1 MBit/s
Physikalisches Interface	differentiell, Anschluß gemäß ISO 11898
Galvanische Trennung des CAN-Interfaces gegenüber anderen Baugruppen	Trennung über Optokoppler und DC/DC-Wandler, nach VDE 0110b §8, Isolationsgruppe C und Einbau in Schaltschrank): 300 V(DC), 250 V(AC)
Busabschluß	muß extern gesetzt werden
Anschluß	X600, DSUB9-Stecker (Stiftkontakte)

Tabelle 12: CAN-Interface CAN0 (optional)

3.1.8.2 CAN-Interface und Programmierung (CAN1)

Hinweis:	Das Bauteile des CAN-Netz CAN1 sind auf der CAN-CBM-OP5 bestückt, das CAN-Netz wird aber von der Firmware nicht unterstützt!

CAN-Controller	MB90543
CAN-Protokoll	CAN 2.0A/2.0B
CAN-Übertragungsrate	10 kBit/s ... 1 MBit/s
Physikalisches Interface	differentiell, Anschluß gemäß ISO 11898
Galvanische Trennung des CAN-Interfaces	keine
CAN-Busabschluß	120 Ω zwischen CAN_L und CAN_H, lokal auf CAN-LCT
Signale des Programmier-Interfaces	Tx, Rx, PROG*, PRESET*
Anschluß	X100, 16pol. Pfostenstecker

Tabelle 13: CAN-Interface CAN1 und Programmierung

3.1.8.3 Kodierschalter

Anzahl	3
Belegung	SW700: CAN-ID-LOW SW701: CAN-ID-HIGH SW702: Baud

Tabelle 14: Kodierschalter

3.2 Platinenansicht

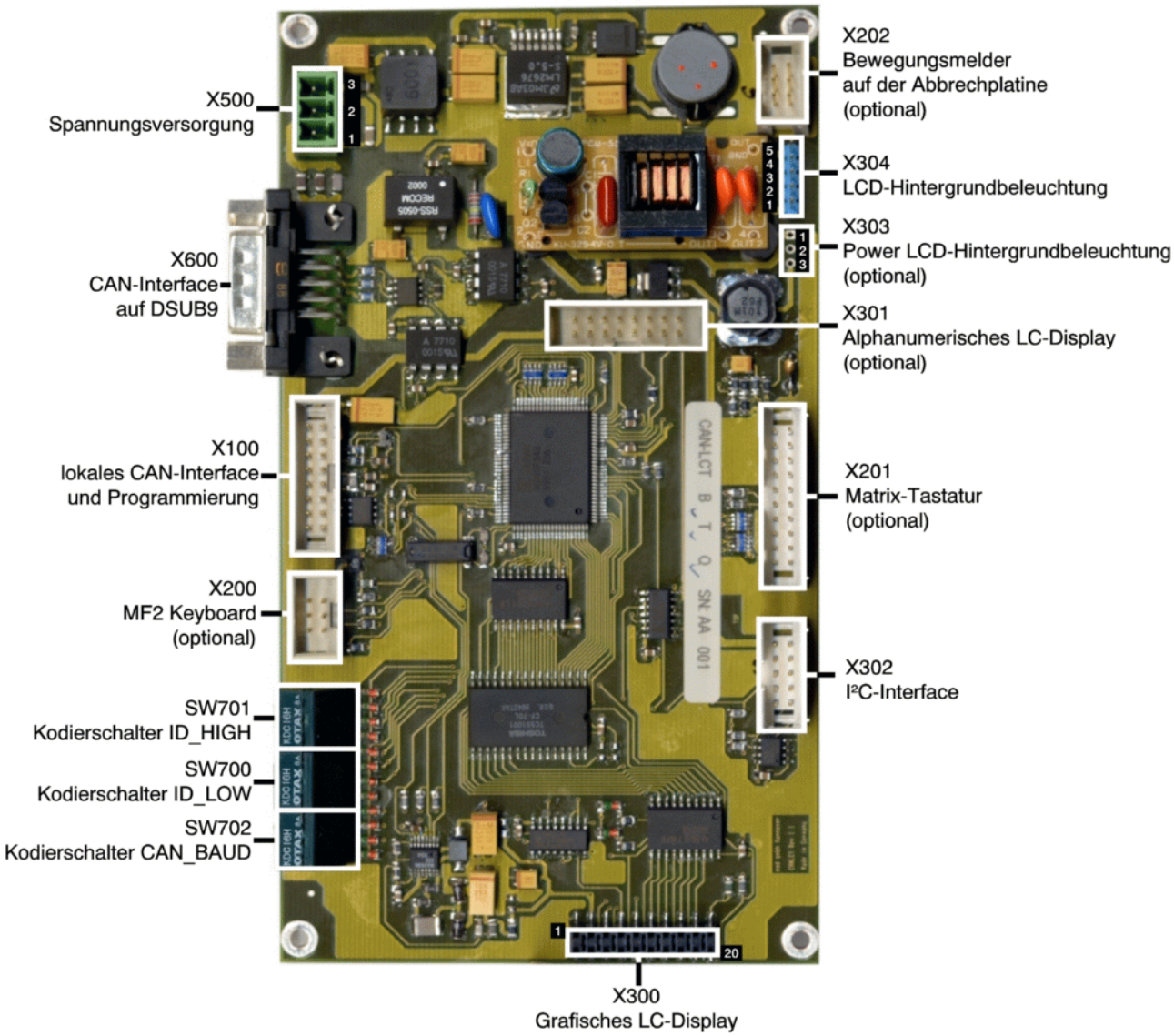


Abb. 7: CAN-LCT Platinenansicht mit Steckerbeschreibung

3.3 Beschreibung der Baugruppen

3.3.1 CAN-Bus-Interfaces

3.3.1.1 CAN-Netz CAN0

Das physikalische Interface der CAN-Schnittstelle CAN0 auf dem DSUB9-Stecker X600 entspricht der Norm ISO11898. Die Übertragungsgeschwindigkeit ist von 10 kBit/s bis 1,0 MBit/s programmierbar. Als CAN-Bus-Transceiver werden der Si9200 oder der 82C250 eingesetzt.

Das CAN-Interface ist über Optokoppler und DC/DC-Wandler von den anderen Baugruppen galvanisch getrennt.

Die Bitrate und die Node-ID können über Kodierschalter eingestellt werden.

3.3.1.2 CAN-Netz CAN1 und Programmierung

CAN-Netz CAN1 ist über den 16poligen Pfostenstecker X100 zugänglich. Als Transceiver wird ein Si9200 oder der 82C250 eingesetzt. Das CAN-Interface ist nicht galvanisch getrennt.

CAN1 ist lokal mit einem 120 Ω-Widerstand abgeschlossen, daher darf hier kein externer Abschlußwiderstand angeschlossen werden!

Hinweis:	Das Bauteile des CAN-Netz CAN1 sind auf der CAN-CBM-OP5 bestückt, das CAN-Netz wird aber von der Firmware nicht unterstützt!

Der Stecker X100 ist neben dem CAN-Interface noch mit den Signalen für die Programmierung des Microcontrollers belegt. Die Belegung des Steckers ist auf Seite 19 beschrieben.

3.3.2 MF2-Tastatur (Option)

Das CAN-LCT-Modul unterstützt IBM-MF-kompatible Tastaturen für AT- und PS/2-kompatible Systeme.

Das Interface ist über den 6poligen Pfostenstecker X200 zugänglich. Zum Anschluß einer handelsüblichen Tastatur ist ein Adapter notwendig, der den 6poligen Pfostenstecker auf eine 5polige Mini-DIN-Buchse (PS2) umsetzt. Die Belegung der beiden Stecker ist ab Seite 20 beschrieben.

Die Daten- und Taktleitungen des MF2-Interfaces sind mit den Ports des MB90543 verbunden. Der Controller schaltet außerdem über ein weiteres Port die Versorgungsspannung der Tastatur. Dadurch wird ein definiertes Rücksetzen der Tastatur während eines lokalen RESETs möglich.

3.3.3 Interface für Grafikfähiges LC-Display

Das LC-Display-Interface ist für Grafik-Displays vom Typ TLX (Toshiba) ausgelegt.

Die Kontrastspannung (VEE) wird auf der Prozessorplatine aus der +5 V-Versorgung erzeugt. Über einen lokalen NTC-Temperaturfühler wird die Kontrastspannung automatisch entsprechend der Umgebungstemperatur eingestellt.

3.3.4 Hintergrundbeleuchtung für LC-Displays

Auf der CAN-LCT-Platine ist ein Power-Modul, daß die Versorgungsspannung für die Hintergrundbeleuchtung des LC-Displays erzeugt, bestückt. Das Modul kann sowohl für die Beleuchtung des grafikfähigen LC-Displays als auch für das optionale alphanumerische LC-Display verwendet werden.

Die Spannungszuführung des Power Moduls kann über die lokale +5V-Spannungsversorgung der CAN-LCT oder einen eigenen optional bestückbaren Stecker (X303) oder über den optional bestückbaren Stecker des alphanumerischen Displays (X301) erfolgen.

Um die Lebensdauer der CCFL-Hintergrundbeleuchtung zu verlängern, kann optional eine automatische Abschaltung der Beleuchtung realisiert werden. An das optionale Bewegungsmelder-Interface auf Stecker X202 kann eine kleine Zusatzplatine, die mit einem Bewegungsmelder und einer Anzeige-LED bestückt ist, angeschlossen werden. Der Infrarot-Bewegungsmelder registriert einen Betrachter vor dem Display und schaltet die Beleuchtung ein, solange sich der Betrachter im Erfassungsradius des Bewegungsmelders befindet. Kann der Bewegungsmelder keinen Betrachter mehr feststellen, so wird nach einer programmierbaren Verzögerungszeit die Hintergrundbeleuchtung abgestellt.

3.4 Interne Steckverbinder auf der CAN-LCT-Platine

Dieses Kapitel enthält nur die intern zugänglichen Steckverbinder. Die an der Gehäuserückwand zugänglichen Steckverbinder sind ab Seite 33 beschrieben.

3.4.1 Lokaler CAN-Bus und Programmierung (X100)

Signal	Pin		Signal
GND	1	2	GND
GND	3	4	GND
CAN_L 1*)	5	6	CAN_H 1*)
GND	7	8	Preset*
GND	9	10	Prog*
RxD	11	12	TxD
VCC	13	14	VCC
VCCB	15	16	VCCB

16poliger Pfostenstecker

Tabelle 15: Belegung des lokalen CAN-Bus-Steckers

1*) Lokaler CAN-Abschlußwiderstand 120 Ω zwischen CAN_L und CAN_H.

3.4.2 MF2-Keyboard auf Pfostenstecker X200 (Option)

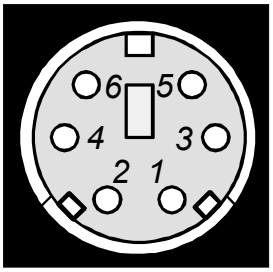
Signal	Pin		Signal
KBD D*	1	2	GND
VCC	3	4	KBD RST*
KBD C*	5	6	KBD VCC*

6poliger Pfostenstecker

Tabelle 16: Belegung des MF2-Tastatursteckers

Um eine handelsübliche MF2-Tastatur mit PS2-Interface anschließen zu können, ist ein Adapterkabel notwendig, daß eine Umsetzung vom 6poligen Pfostenstecker X200 auf eine 6polige Mini-DIN-Buchse vornimmt. Die folgende Tabelle zeigt die Belegung der Mini-DIN-Buchse:

Pin-Zuordnung:



Pin-Belegung:

Pin	Signal
1	KBD D*
2	KBD RST*
3	GND
4	KBD VCC*
5	KBD C*
6	-

6polige Mini-Din-Buchse

Tabelle 17: Belegung einer Mini-DIN-Buchse für ein PS2-Tastatur-Interface

3.4.3 Matrix-Tastatur (X201)

Signal	Pin		Signal
VCC	1	2	ROW0
ROW1	3	4	ROW2
ROW3	5	6	ROW4
ROW5	7	8	ROW6
ROW7	9	10	OPOT1
OPOT2	11	12	COL0
COL1	13	14	COL2
COL3	15	16	COL4
COL5	17	18	COL6
COL7	19	20	GND

20poliger Pfostenstecker

Tabelle 18: Belegung des Matrixtastatursteckers

VCC, GND ... Versorgungsspannung und Bezugspotential

ROWx ... Scan-Lines (Row) der Matrix-Tastatur (x = 0, 1, ... 7)

COLx ... Return-Lines (Column) der Matrix-Tastatur (x = 0, 1, ... 7)

OPT1, OPT2 ... Signale des optischen Drehgebers

3.4.4 Bewegungsmelder und LED auf separater Platine an X202 (Option)

Signal	Pin		Signal
GND	1	2	MDETECT
VCC	3	4	GND
LED1	5	6	LED0

6poliger Pfostenstecker

Tabelle 19: Belegung des Bewegungsmeldersteckers

3.4.5 Grafisches LC-Display (X300)

Signal	Pin		Signal
GND	1	2	GND
VCC	3	4	VEE
LCDWR*	5	6	LCDRD*
CS LCD*	7	8	C/D*
GND	9	10	LRESET*
DL0	11	12	DL1
DL2	13	14	DL3
DL4	15	16	DL5
DL6	17	18	DL7
LCD FONT	19	20	VCC

20poliger Leiterplattenverbinder

Tabelle 20: Belegung des Grafik-Display-Steckers

3.4.6 Optionales alphanumerisches LC-Display an X301 (Option)

Signal	Pin		Signal
VCC	1	2	GND
LCD2 RS	3	4	VEE
LCD2 E	5	6	LCD2 RW
n.c.	7	8	n.c.
n.c.	9	10	n.c.
LCD2 D5	11	12	LCD2 D4
LCD2 D7	13	14	LCD2 D6
geschaltetes GND	15	16	VCC

16poliger Pfostenstecker

Tabelle 21: Belegung des Steckers des alphanumerischen Displays

3.4.7 I²C-Interface (X302)

Signal	Pin		Signal
VCCB	1	2	VCCB
SDA	3	4	SCL
GND	5	6	GND
TIQ*	7	8	GND
GND	9	10	GND

10poliger Pfostenstecker

Tabelle 22: Belegung des I²C-Steckers

VCC, GND ... Versorgungsspannung und Bezugspotential

SDA, SCL ... I²C-Signale

3.4.8 Spannungsversorgung des LCD-Power-Moduls an X303 (Option)

Pin	Signal
1	VCC
2	GND
3	geschaltetes GND

3polige Stiftleiste

Tabelle 23: Belegung des Power-Modul-Steckers

3.4.9 LCD-Hintergrundbeleuchtung (X304)

Pin	Signal
1	LCDP1
2	n.c.
3	n.c.
4	n.c.
5	LCDP2

5polige Stiftleiste

Tabelle 24: Belegung des Steckers der Hintergrundbeleuchtung

4. Tastatur

4.1 Übersicht

4.1.1 Funktionsbeschreibung

Auf der Tastaturplatine sind 22 Taster, ein optischer Drehgeber (Jog-Shuttle) und eine zweifarbige LED (optional 3 LEDs) bestückt. Die Platine wird direkt unter dem Folienüberzug hinter der Frontplatte montiert und über zwei Flachbandkabel mit der LC-Display-Platine CAN-LCT verbunden.

Die Scan- und Return-Lines der Matrixtastatur werden auf einen 20poligen Pfostenstecker geführt. Hier sind auch die Signale des optischen Drehgebers angeschlossen.

Über einen I²C-Port-Baustein werden die drei möglichen LEDs angesteuert. Die Adressen der Port-Bausteine sind fest verdrahtet.

Optional kann auf der Tastaturplatine ein Power-Modul für die Hintergrundbeleuchtung eines LC-Displays bestückt werden. Dieses Modul stellt die Versorgungsspannung für die Leuchtröhre des Displays zur Verfügung. Der Anschluß an das Display erfolgt über eine 5polige Pfostenleiste.

Als weitere Option sind zwei weitere zweifarbige LEDs bestückbar.

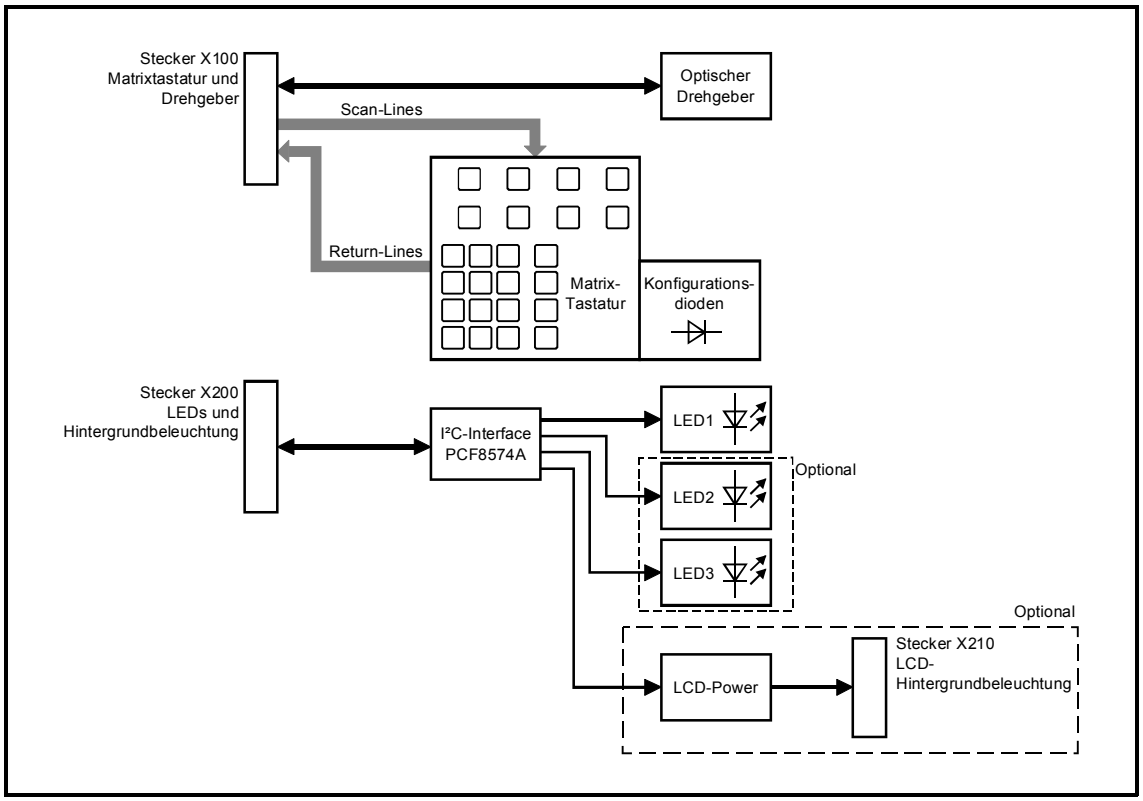


Abb. 9: Blockschaltbild der Tastaturplatine

4.1.2 Allgemeine Technische Daten

Temperaturbereich	max. zulässige Umgebungstemperatur (d. h. innerhalb des Gehäuses): 0...50 °C
Steckverbinder	X100 - 20pol. Pfostenstecker (Matrix-Tastatur und Drehgeber) X200 - 10pol. Pfostenstecker (I^2C-Bus) X210 (optional) - 5pol. Pfostenreihe (LCD-Hintergrundbeleuchtung)
Luftfeuchtigkeit	max. 90%, nicht kondensierend
Größe der Tastaturplatine	136,53 mm x 123,44 mm

Tabelle 25: Allgemeine Daten der Tastaturplatine

4.1.3 Tasten

Anzahl	22 (optional bis zu 24)
Ansteuerung	Matrixtastatur mit Scan- und Return-Leitungen, Auswertung auf LC-Display-Modul
Lebensdauer mechanisch	> 5.000.000 Betätigungen

Tabelle 26: Daten der Tasten

4.1.4 Drehgeber (Jog-Shuttle)

Typ	optischer Drehgeber HRPG-A (*HEWLETT PACKARD*)
Auflösung	effektive Auflösung: 32 Pulse pro Umdrehung
Anschluß	Drehschalter wird direkt auf der Platine bestückt

Tabelle 27: Drehschalter

4.1.5 Platinenansicht mit Zuordnung der Scan- und Return-Lines

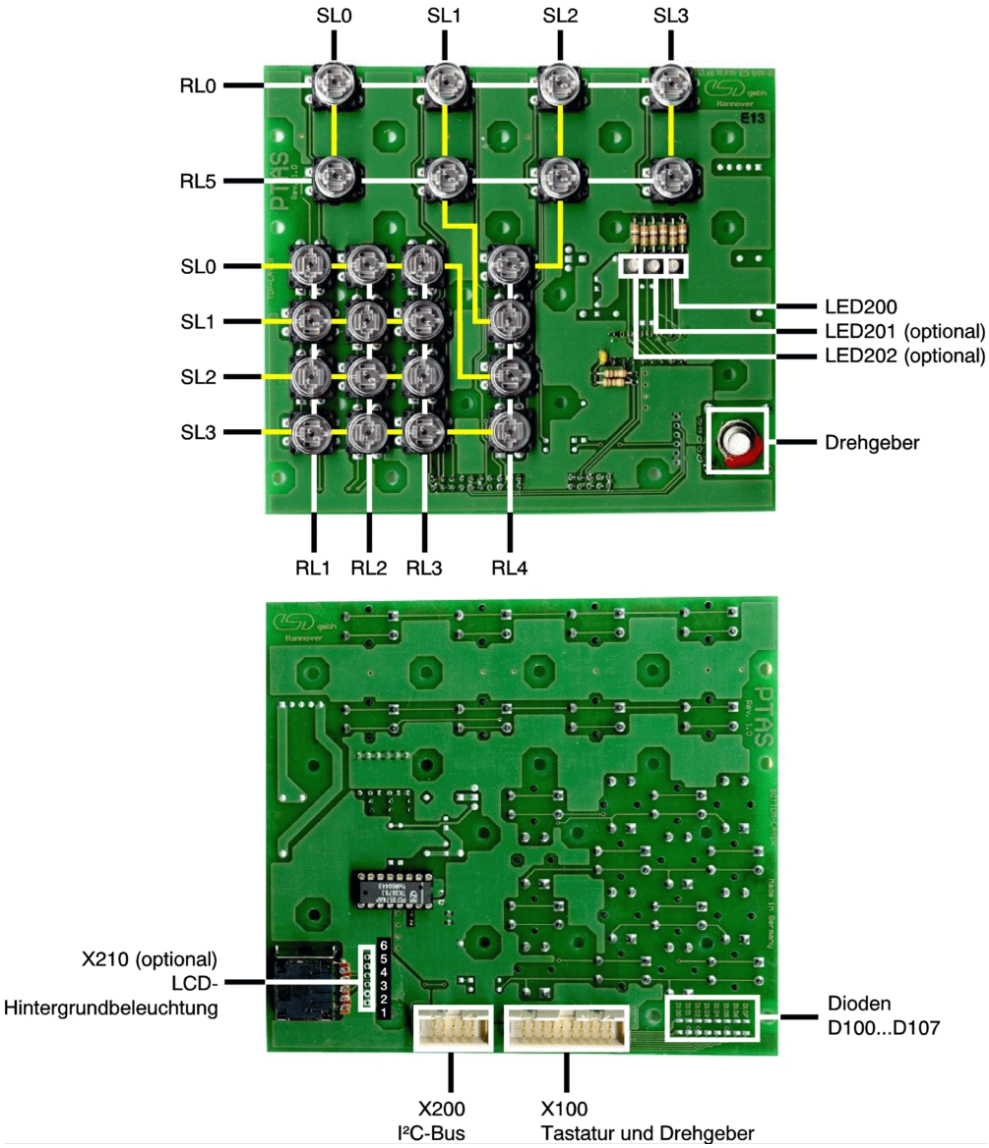


Abb. 10: Ansicht der Tastaturplatine

4.1.6 I²C-Port-Baustein

Der I²C-Baustein steuert die drei zweifarbigen LEDs (davon zwei optional in der Standardbestückung) und die optionale Spannungsversorgung des LC-Displays an.

I²C-Baustein-Port	Funktion
P0	optionale LCD-Hintergrundbeleuchtung
P1	
P2	**LED200**
P3	
P4	LED201 (optional)
P5	
P6	LED202 (optional)
P7	

Tabelle 28: Zuordnung der Ports zu den LEDs und der Hintergrundbeleuchtung

4.1.7 Scan-Lines und Return-Lines

Die Belegung der Scan- und Return-Lines mit den Tasten kann der Abbildung auf Seite 27 entnommen werden.

Hinweis: In der Standardbestückung sind nur 22 Tasten vorgesehen, d.h. die Tasten an SL1/RL4 und SL2/RL4 sind nicht bestückt!

Ist ein Drehgeber (Jog-Shuttle) mit integriertem Taster bestückt, so wird die Taste über die Scan-Line SL3 und die Return-Line RL4 ausgewertet.
Die Auswertung der Scan- und Return-Lines erfolgt auf dem CAN-LCT-Modul des CAN-CBM-OP5.

Optional kann die Tastaturplatine mit 8 Konfigurationsdioden bestückt werden. Diese Dioden brücken die Return-Lines RL0 bis RL7 auf die Scan-Line SL7:

Scan-Line SL7 verbunden mit Return-Line ...	Diode bestücken
RL0	D100
RL1	D101
RL2	D102
RL3	D103
RL4	D104
RL5	D105
RL6	D106
RL7	D107

Tabelle 29: Optionale Konfigurationsdioden an Scan- und Return-Lines

4.2 Interne Steckverbinder auf der Tastaturplatine

4.2.1 Tastatur und Drehgeber (X100)

Signal	Pin		Signal
VCC	1	2	SL0 (ROW0)
SL1 (ROW1)	3	4	SL2 (ROW2)
SL3 (ROW3)	5	6	n.c. (ROW4)
n.c. (ROW5)	7	8	n.c. (ROW6)
SL7 (ROW7)	9	10	OPOT1
OPOT2	11	12	RL0 (COL0)
RL1 (COL1)	13	14	RL2 (COL2)
RL3 (COL3)	15	16	RL4 (COL4)
RL5 (COL5)	17	18	RL6 (COL6)
RL7 (COL7)	19	20	GND

20poliger Pfostenstecker

Tabelle 30: Belegung des Matrixtastatursteckers

n.c. ... nicht angeschlossen

VCC, GND ... Versorgungsspannung und Bezugspotential

SLx (ROWx) ... Scan-Lines der Matrix-Tastatur (x = 0, 1, 2, 3)

RLx (COLx) ... Return-Lines der Matrix-Tastatur (x = 0, 1, ... 7)

OPT1, OPT2 ... Signale des optischen Drehgebers

4.2.2 I²C-Interface (X200)

Pin	Signal		Pin
1	VCC	VCC	2
3	SDA	SCL	4
5	GND	GND	6
7	-	GND	8
9	GND	GND	10

10poliger Pfostenstecker

Tabelle 31: Belegung des I²C-Steckers

VCC, GND ... Versorgungsspannung und Bezugspotential

SDA, SCL ... I²C-Signale

4.2.3 Optionale LCD-Hintergrundbeleuchtung (X210)

Pin	Signal
1	LCDP1
2	-
3	-
4	-
5	LCDP2

5polige Pfostenreihe

LCDP1, LCDP2 ... Anschlüsse der CF-Lampe

5. Anhang

5.1 Externe Steckverbinder

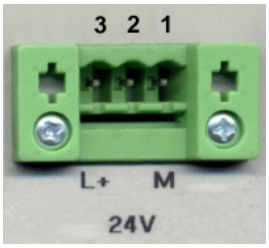
5.1.1 Ansicht der Geräterückwand

Abb. 11: Ansicht der Geräterückwand

5.1.2 Spannungsversorgung (X500)

Gerätestecker: PHOENIX DFK-MSTB2,5/3-G-5,08
Leitungsstecker: z.B. PHOENIX MSTB2,5/3-ST-5,08 (nicht im Lieferumfang enthalten)

Pin-Zuordnung:



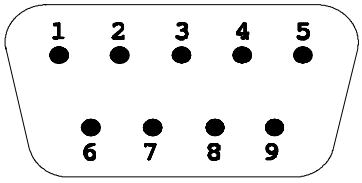
Pin-Belegung:

Pin	Beschriftung	Signal
3	L+	+24V
2	..	Shield
1	M	Masse

Tabelle 32: Belegung des Spannungsversorgungssteckers

5.1.3 CAN-Interface (X600)

Pin-Zuordnung:



Pin-Belegung:

Signal	Pin		Signal
CAN_GND	6	1	n.c
CAN_H	7	2	CAN_L
n.c	8	3	CAN_GND
n.c	9	4	n.c.
		5	Shield

9poliger DSUB-Stecker mit Stiftkontakten

Tabelle 33: Belegung des CAN-Steckers

Signalbeschreibung:

CAN_L, CAN_H ... CAN-Signalleitungen
CAN_GND ... Bezugspotential des lokalen CAN-Physical Layers
Shield ... Abschirmung
(an Schirmblech angeschlossen)
n.c. ... not connected

CAN-CBM-OP5

CANopen Terminal mit Jog-Shuttle

Software-Handbuch

Dokument-Datei:	I:\texte\Doku\MANUALS\CAN\CBM\OP5\can-cbm-op5_11s.ma9
Datum des Ausdrucks:	12.05.2004

Bestell-Nr.:	CAN-CBM-OP5 C.2120.05

Änderungen in den Kapiteln

Die hier aufgeführten Änderungen im Dokument betreffen sowohl Änderungen in der Hardware als auch reine Änderungen in der Beschreibung der Sachverhalte.

Kapitel	Änderungen gegenüber Vorversion
-	Sub-Index-Eintragungen der Objekte korrigiert.

Weitere technische Änderungen vorbehalten.

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. **esd** übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei **esd**. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch **esd** gestattet.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover

Tel.: 0511/372 98-0
FAX : 0511/372 98-68
E-Mail: info@esd-electronics.com
Internet: www.esd-electronics.com

Inhalt

1. Übersicht 3
1.1 Tabelle der wichtigsten Identifier und Nachrichten für CANopen 3

2. Belegung der Tx-PDOs 4
2.1 Übersicht 4
2.2 Tx-PDO1 4
2.3 Tx-PDO2 5

3. Belegung der Rx-PDOs 6
3.1 Übersicht 6
3.2 Rx-PDO1 6
3.3 Rx-PDO2 7
3.4 Rx-PDO3 7
3.5 Rx-PDO4 7

4. Service-Daten-Objekte 8

5. Übersicht der Kommunikationsparameter 9

6. Manufacturer-Specific Objects 10
6.1 Übersicht der implementierten Objekte 10
6.2 Beschreibung der Objekte 11
6.2.1 Status/Command des Bewegungsmelders und der LCD-Hintergrundbeleuchtung (2000_h) 11
6.2.2 LCD_Backlight_Time (2001_h) 12
6.2.3 Read_Matrix_Keyboard (2100_h) 13
6.2.4 Set_Matrix_Keyboard (2110_h) 14
6.2.5 MF2-Keyboard (2200_h) 15
6.2.6 Optischer Drehgeber (Jog-Shuttle, Object 2300_h) 17
6.2.7 LCD_Data (2400_h) 18
6.2.8 Graphic_Command (2410_h) 19
6.2.9 User_Character (2420_h) 23
6.2.10 Graphic_Font_16x12 (2430_h), Graphic_Font_8x6 (2401_h) 24
6.2.11 Contrast (2440_h) 25
6.2.12 Display_Temperature (2450_h) 26
6.2.13 HW_Serial_No ($2F00_h$) 27

7. Grafikfähige Displays 28
7.1 Organisation des Displays 28
7.2 Aufteilung der Ebenen 29
7.3 Konfiguration und Attribut-Ebene 31
7.4 Zeichensatz der Text-Ebenen 35

Diese Seite ist bewußt unbedruckt.

1. Übersicht

Den Kern des CAN-CBM-OP5 bildet das sogenannte LCT-Modul. Dieses Handbuch beschreibt die Firmware des CAN-LCT-Moduls. Zunächst werden die CANopen-Funktionalitäten beschrieben. Anschließend folgt das Kapitel "Grafikfähige Displays" in dem auf grundlegende Funktionen des eingesetzten Displays eingegangen wird und die Escape-Sequenzen zur Display-Ansteuerung beschrieben werden.

1.1 Tabelle der wichtigsten Identifier und Nachrichten für CANopen

CAN-Identifier [Hex]	Bezeichnung	Länge	Daten [Hex]	Bemerkungen
0	NMT	2	02 xx	Modul geht in Zustand *stopped*
0	NMT	2	01 xx	Start (Modul geht in Zustand *operational*)
0	NMT	2	80 xx	Modul geht in Zustand *preoperational*
0	NMT	2	81 xx	Reset Module
0	NMT	2	82 xx	Reset Communication
180_h + Node-ID	Tx_PDO_1	0...4 Bytes	Nutzdaten	vom CAN-LCT-Modul (Tx/Sende-PDO)
280_h + Node-ID	Tx_PDO_2			
200_h + Node-ID	Rx_PDO_1	0...8 Bytes	Nutzdaten	zum CAN-LCT-Modul
300_h + Node-ID	Rx_PDO_2			(Rx/Empfangs-PDO)
400_h + Node-ID	Rx_PDO_3			(Rx/Empfangs-PDO)
500_h + Node-ID	Rx_PDO_4			(Rx/Empfangs-PDO)
700_h + Node-ID	NMT	8 Bytes	Identifier	Node Guarding

xx = Node-ID, bzw. xx=00 wenn alle CANopen-Nodes angesprochen werden sollen

Hinweis: Die Node-ID des CAN-LCT-Moduls ist bei Auslieferung auf den Wert '1' eingestellt.

2. Belegung der Tx-PDOs

2.1 Übersicht

Tx-PDO	**Byte** (big endian)				
	0	**1**	**2**	**3**	**4...7**
Tx-PDO1	*LCT_Status*	-	-	-	-
Tx-PDO2	*Matrix_ Keyboard*	*MF2-Keyboard*	*Jog-Shuttle*		-

2.2 Tx-PDO1

Status-Message des Moduls mit Bewegungsmelder (Option) und LCD-Backlight-Status (entspricht Object 2000_h Lesezugriff)

Es wird nur Byte 0 gesendet. Das Byte ist wie folgt belegt:

Belegung der Variablen *LCT_Status*:

Bit	Name	Inhalt
0	*Movement_Detection*	0: keine Bewegung feststellbar 1: Bewegung erkannt
1	*LCD_Backlight*	0: Hintergrundbeleuchtung ist aus 1: Hintergrundbeleuchtung ist an
2	0	diese Bits werden immer als ‘0’ zurückgegeben
3	0	
4	0	
5	0	
6	0	
7	0	

2.3 Tx-PDO2

Matrix-Tastatur, MF2-Tastatur (Option) und optischer Drehgeber (Jog-Shuttle)

Byte (little endian)	**3**	**2**	**1**	**0**
	Jog_Shuttle		*MF2-Keyboard*	*Matrix_ Keyboard*

Jog-Shuttle ... aktuelle Stellung des optischen Drehschalters

MF2-Keyboard ... MF2-Tastatur-Code

Matrix_ Keyboard ... Matrix-Tastatur-Code nach der auf Seite 13 dargestellten Tabelle

3. Belegung der Rx-PDOs

3.1 Übersicht

Rx-PDO	**Byte** (big endian)							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Rx-PDO1	*LCT_ Command*	-	-	-	-	-	-	
Rx-PDO2	*Jog_Shuttle_Ref*		-	-	-	-	-	-
Rx-PDO3	*LCD_ Data_0*	*LCD_ Data_1*	*LCD_ Data_2*	*LCD_ Data_3*	*LCD_ Data_4*	*LCD_ Data_5*	*LCD_ Data_6*	*LCD_ Data_7*
Rx-PDO4	*Graphic_ Data_0*	*Graphic_ Data_1*	*Graphic_ Data_2*	*Graphic_ Data_3*	*Graphic_ Data_4*	*Graphic_ Data_5*	*Graphic_ Data_6*	*Graphic_ Data_7*

3.2 Rx-PDO1

LCD-Backlight und Anzeige LEDs (entspricht Object 2000_h Schreibzugriff)

Belegung der Variablen *LCT_Command*:

Bit	Name	Inhalt
0	*LCD_Backlight_On*	0: Hintergrundbeleuchtung aus 1: Hintergrundbeleuchtung an
1	*LED_200_L*	LED 200 rot/grün/aus
2	*LED_200_H*	
3	*LED_201_L*	LED 201(optional) rot/grün/aus
4	*LED_201_H*	
5	*LED_202_L*	LED 202 (optional) rot/grün/aus
6	*LED_202_H*	
7	*0*	dieses Bit wird nicht ausgewertet

3.3 Rx-PDO2

Referenzpunkt des optischen Drehschalters (Jog-Shuttle) setzen

Byte (little endian)	**1**	**0**
	Jog_Shuttle_Ref	

Jog_Shuttle_Ref ... Referenzpunkt des optischen Drehschalters
Der aktuellen Kodierschalterposition wird der Wert *Jog_Shuttle_Ref* zugeordnet.

3.4 Rx-PDO3

LC-Display-Daten setzen (entspricht Object 2400_h)

3.5 Rx-PDO4

Grafik-Display-Daten setzen (entspricht Object 2410_h)

4. Service-Daten-Objekte

Aufbau des SDO:

Byte (big endian)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	cmd	*idx*		*subidx*	*cfgD0*	*cfgD1*	*cfgD2*	*cfgD3*

cmd - Kommando Byte (Service → CAN-LCT)	
Wert [Hex]	Beschreibung
0	keine Auswertung
2F 2B 27 23	1 Byte Daten schreiben 2 Byte Daten schreiben 3 Byte Daten schreiben 4 Byte Daten schreiben
40	Daten lesen

cmd - Status Byte (Antwort ← CAN-LCT)	
Wert [Hex]	Beschreibung
0	keine Antwort
4F 4B 47 43	Antwort auf 1 Byte Daten lesen Antwort auf 2 Byte Daten lesen Antwort auf 3 Byte Daten lesen Antwort auf 4 Byte Daten lesen
60	Antwort auf ‘Daten schreiben’
80	Fehlermeldung

5. Übersicht der Kommunikationsparameter

Das Format der Kommunikationsparameter kann der CiA DS-301, Kap. 9.1 entnommen werden. Das Modul unterstützt nur die in der folgenden Tabelle aufgeführten Kommunikationsparameter.

Index [Hex]	Name	Subindex	Typ	Zugriff	Default
1000	*Device Type*	-	Unsigned32	ro	00010000_h
1001	*Error Register*	-	Unsigned8	ro	Error-Code
1008	*Manufacturer's Device Name*	-	Visible String	ro	LCT
1009	*Manufacturer's Hardware Version (see: DS-301)*	-	Visible String	ro	1.1 1*)
100A	*Manufacturer's Software Version*	-	Visible String	ro	1.5 1*)
100B	*Node-ID*	-	Unsigned32	ro	1 oder Kodier-schalterstellung
100C	*Guard Time*	-	Unsigned16	rw	0 Sek.
100D	*Life Time Factor*	-	Unsigned8	rw	0
100E	*Node Guarding ID*	-	Unsigned32	ro	700_h +Node-ID = 701_h
1400 - 1403	*Receive PDO Communication Parameter*	0,1,2	PDOCommPar	ro	--
1600 - 1603	*Receive PDO Mapping Parameter*	0,1	PDOMapping	ro	--
1800 - 1801	*Transmit PDO Communication Parameter*	0,1,2,3,4,5	PDOCommPar	rw	--
1A00-1A01	*Transmit PDO Mapping Parameter*	0,1,2, (3, 4, 5, 6)	PDOMapping	ro	--

ro - Read Only, rw - Read/Write
1*) abhängig vom Revisionsstand der Hardware/Software

6. Manufacturer-Specific Objects

6.1 Übersicht der implementierten Objekte

Index [Hex]	Name	Data Type
2000	*LCT_Command_Status*	Unsigned8
2001	*LCD_Backlight_Time*	Unsigned16
2100	*Read_Matrix_Keyboard*	Unsigned8
2110	*Set_Matrix_Keyboard*	Unsigned8
2200	*MF2_Keyboard*	3x Unsigned8
2300	*Rotary_Encoder*	Unsigned16
2400	*LCD_Data*	Datenstrom
2410	*Graphic_Command*	Datenstrom
2420	*User_Character*	Datenstrom
2430	*Graphic_Font_16x12*	Datenstrom
2431	*Graphic_Font_8x6*	Datenstrom
2440	*Contrast*	Signed8
2450	*Display_Temperature*	Signed8
2F00	*HW_Serial_No*	Unsigned16

6.2 Beschreibung der Objekte

6.2.1 Status/Command des Bewegungsmelders und der LCD-Hintergrundbeleuchtung (2000_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2000	1	*LCT_Command_Status*	00...03	-	Byte	ro

Bedeutung der Variablen *LCT_Command_Status*:

Diese Variable ist beim Lesen und beim Schreiben unterschiedlich besetzt !

Beim Lesen erhält man den *LCT_Status:*

Die Variable *LCT_Status* gibt den aktuellen Status des optionalen Bewegungsmelders (sofern bestückt) und der LCD-Hintergrundbeleuchtung wieder.

Bit	Name	Inhalt
0	*Movement_Detection*	0: keine Bewegung feststellbar 1: Bewegung erkannt
1	*LCD_Backlight*	0: Hintergrundbeleuchtung ist aus 1: Hintergrundbeleuchtung ist an
2	0	diese Bits werden immer als ‘0’ zurückgegeben
3	0	
4	0	
5	0	
6	0	
7	0	

Beim Schreiben wird *LCT_Command* gesetzt:

Bit	Name	Inhalt
0	*LCD_Backlight_On*	0: Hintergrundbeleuchtung aus 1: Hintergrundbeleuchtung an
1	*LED_200_L*	LED 200 rot/grün/aus
2	*LED_200_H*	
3	*LED_201_L*	LED 201 rot/grün/aus (optional)
4	*LED_201_H*	
5	*LED_202_L*	LED 202 rot/grün/aus (optional)
6	*LED_202_H*	
7	*0*	dieses Bit wird nicht ausgewertet

6.2.2 LCD_Backlight_Time (2001_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2001	0	*LCD_Backlight_Time*	0000...FFFF	30000	Byte	r/w

Bedeutung der Variablen *LCD_Backlight_Time*:

LCD_Backlight_Time ... Verzögerungszeit für das Ausschalten der Hintergrundbeleuchtung in [ms]
Um die Lebensdauer der Hintergrundbeleuchtung zu erhöhen, wird die Leuchtdauer eingeschränkt. Die Hintergrundbeleuchtung wird eingeschaltet, wenn der Bewegungsmelder z.B. durch einen Betrachter aktiviert wurde. Nachdem der Bewegungsmelder keinen Betrachter mehr wahrnimmt, wird die Beleuchtung nicht sofort ausgeschaltet, sondern bleibt für die Dauer von *LCD_Backlight_Time* an.

0000: 0 ms
FFFF: 65535 ms

6.2.3 Read_Matrix_Keyboard (2100_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2100	0	*Matrix_Keyboard*	00...FF	-	Byte	ro

Bedeutung der Variablen *Matrix_Keyboard*:

Diese Variable liefert den Tasten-Code der zuletzt gedrückten Taste der Matrixtastatur zurück. Die Default-Einstellung des Tasten-Codes ist in der folgenden Abbildung dargestellt. Der Tasten-Code kann über das Objekt 2110_h verändert werden.

Der Treiber ist standardmäßig mit der Tastenbelegung für die Tastatur der CAN-CBM-OP5 (PTAS) vorbesetzt. Der Tastaturtreiber liefert die dem Aufdruck der Tastatur entsprechenden ASCII-Zeichen, sofern sie eindeutig sind. Die folgende Abbildung zeigt die Tastatur mit den (in Klammern angegebenen) gesendeten Zeichen:

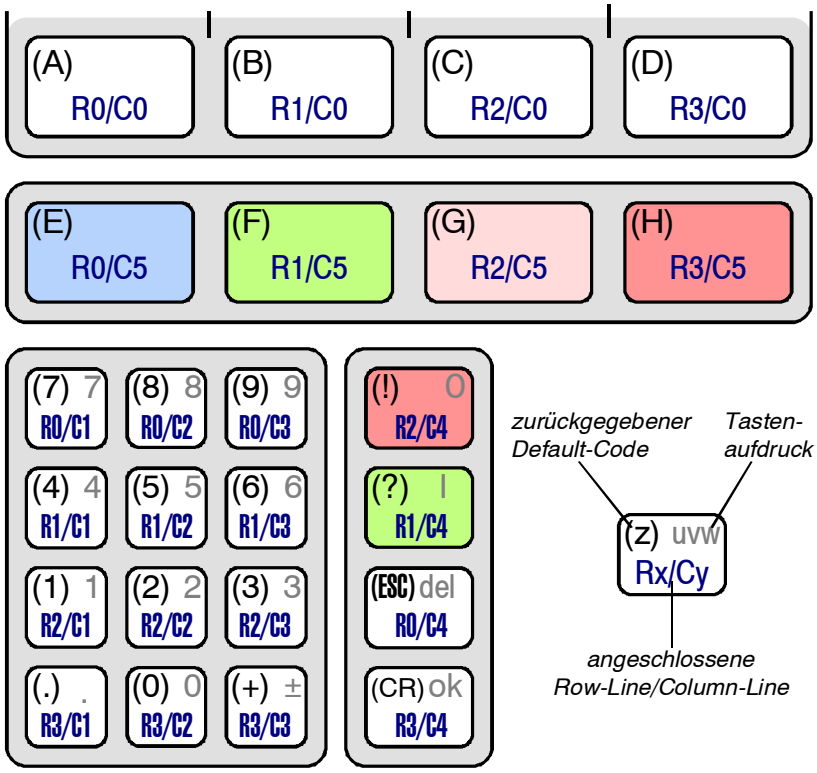


Abb. 1: Zurückgegebene Codes der Tastatur PTAS

6.2.4 Set_Matrix_Keyboard (2110_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2110	0	*Set_Matrix_Keyboard*	00...08, bzw. 00...FF	-	3x Byte	wo

Bedeutung der Variablen *Set_Matrix_Keyboard*:

Mit dieser Variablen können die Tasten-Code programmiert werden, die die Tasten beim Drücken zurückgeben. Es müssen immer drei aufeinanderfolgende Bytes übergeben werden.

Byte (little endian)	**2**	**1**	**0**
	Code	*Row*	*Column*

Row ... Row-Line, auf der die zu ändernde Taste liegt (siehe Seite 13)
Row = 0...8

Column ... Column-Line, auf der die zu ändernde Taste liegt (siehe Seite 13)
Column = 0...8

Code ... neuer Tasten-Code
Code = 00 ... FF_h

6.2.5 MF2-Keyboard (2200_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2200	0	*MF2_Keyboard*	00...FF	-	Byte	ro

Bedeutung der Variablen *MF2_Keyboard*:

Diese Variable liefert den Tastencode einer deutschen Tastatur gemäß der auf den folgenden Seiten angegebenen Tabelle.

Zurückgegebene Tastatur-Codes der MF2-Tastatur:

Code	Taste	Code	Taste	Code	Taste	Code	Taste
$00	'5'N_NoNum	$80	AltCtrl'<'	$40	Alt'Q'= @	$C0	CtrlF1
$01	Ctrl'a'	$81	'ü'	$41	'A'	$C1	CtrlF2 Alt'A'
$02	Ctrl'b'	$82	AltCtrl'b'	$42	'B'	$C2	CtrlF3 Alt'B'
$03	Ctrl'c'	$83	AltCtrl'c'	$43	'C'	$C3	CtrlF4 Alt'C'
$04	Ctrl'd'	$84	'ä'	$44	'D'	$C4	CtrlF5 Alt'D'
$05	Ctrl'e'	$85	AltCtrl'e'	$45	'E'	$C5	CtrlF6 Alt'E'
$06	Ctrl'f'	$86	AltCtrl'f'	$46	'F'	$C6	CtrlF7 Alt'F'
$07	Ctrl'g'	$87	AltCtrl'g'	$47	'G'	$C7	CtrlF8 Alt'G'
$08	BS/◂ Ctrl'j'	$88	AltBS	$48	'H'	$C8	CtrlF9 Alt'H'
$09	TAB Ctrl'i'	$89	AltTAB	$49	'I'	$C9	CtrlF10 Alt'I'
$0A	CUR▾ Ctrl'j'	$8A	AltCUR▾	$4A	'J'	$CA	CtrlF11 Alt'J'
$0B	CUR▴ Ctrl'k'	$8B	AltCUR▴	$4B	'K'	$CB	CtrlF12 Alt'K'
$0C	CUR▸ Ctrl'l'	$8C	AltCUR▸	$4C	'L'	$CC	Alt'L'
$0D	CR/Enter Ctrl'm'	$8D	AltCR	$4D	'M'	$CD	Alt'M'
$0E	Ctrl'n'	$8E	'Ä' AltCtrl'n'	$4E	'N'	$CE	Alt'N'
$0F	Ctrl'o'	$8F	AltCtrl'o'	$4F	'O'	$CF	Alt'O'
$10	Ctrl'p'	$90	AltCtrl'p'	$50	'P'	$D0	PgUp Alt'P'
$11	Ctrl'q'	$91	AltCtrl'q'	$51	'Q'	$D1	PgDwn Alt'Q'
$12	Ctrl'r'	$92	AltCtrl'r'	$52	'R'	$D2	INS Alt'R'
$13	Ctrl's'	$93	AltCtrl's'	$53	'S'	$D3	END Alt'S'
$14	Ctrl't'	$94	'ö' AltCtrl't'	$54	'T'	$D4	DRUCK Alt'T'
$15	'§' Ctrl'u'	$95	Alt'§'	$55	'U'	$D5	POS1 Alt'U'
$16	Ctrl'v'	$96	AltCtrl'v'	$56	'V'	$D6	Alt'V'
$17	Ctrl'w'	$97	AltCtrl'w'	$57	'W'	$D7	Alt'W'
$18	Ctrl'x'	$98	AltCtrl'x'	$58	'X'	$D8	Alt'X'
$19	Ctrl'y'	$99	'Ö' AltCtrl'y'	$59	'Y'	$D9	Alt'Y'
$1A	Ctrl'z'	$9A	'Ü' AltCtrl'z'	$5A	'Z'	$DA	Alt'Z'
$1B	ESC Ctrl'ä'	$9B	AltESC	$5B	Alt'('== [	$DB	AltCtrl'('
$1C	Ctrl'ö'	$9C	AltCtrl'ö'	$5C	Alt'?'== \	$DC	AltCtrl'?'
$1D	Ctrl'ü'	$9D	AltCtrl'ü'	$5D	Alt')'==]	$DD	AltCtrl')'
$1E	Ctrl'#'	$9E	AltCtrl'#'	$5E	'^'	$DE	AltCtrl'^'
$1F	Ctrl'-'	$9F	AltCtrl'-'	$5F	'_'	$DF	AltCtrl'_'
$20	SPACE	$A0	F1 AltSPACE	$60	'`'	$E0	Alt'`'
$21	'!'	$A1	F2 Alt'!'	$61	'a'	$E1	ß Alt'a'
$22	'"'	$A2	F3 Alt'"'	$62	'b'	$E2	Alt'b'
$23	'#'	$A3	F4 Alt'#'	$63	'c'	$E3	Alt'c'
$24	'$'	$A4	F5 Alt'$'	$64	'd'	$E4	Alt'd'
$25	'%'	$A5	F6 Alt'%'	$65	'e'	$E5	Alt'e'
$26	'&'	$A6	F7 Alt'&'	$66	'f'	$E6	Alt'f'
$27	'''	$A7	F8 Alt'''	$67	'g'	$E7	Alt'g'
$28	'('	$A8	F9 Alt'('	$68	'h'	$E8	Alt'h'
$29	')'	$A9	F10 Alt')'	$69	'i'	$E9	Alt'i'
$2A	'*'	$AA	F11 Alt'*'	$6A	'j'	$EA	Alt'j'
$2B	'+'	$AB	F12 Alt'+'	$6B	'k'	$EB	Alt'k'
$2C	','	$AC	Alt','	$6C	'l'	$EC	Alt'l'
$2D	'-'	$AD	Alt'-'	$6D	'm'	$ED	Alt'm'
$2E	'.'	$AE	Alt'.'	$6E	'n'	$EE	Alt'n'
$2F	'/'	$AF	Alt'/'	$6F	'o'	$EF	Alt'o'
$30	'0'	$B0	ShF1 Alt'0'	$70	'p'	$F0	Alt'p'
$31	'1'	$B1	ShF2 Alt'1'	$71	'q'	$F1	Alt'q'
$32	'2'	$B2	ShF3 Alt'2'	$72	'r'	$F2	Alt'r'
$33	'3'	$B3	ShF4 Alt'3'	$73	's'	$F3	Alt's'
$34	'4'	$B4	ShF5 Alt'4'	$74	't'	$F4	Alt't'
$35	'5'	$B5	ShF6 Alt'5'	$75	'u'	$F5	Alt'u'
$36	'6'	$B6	ShF7 Alt'6'	$76	'v'	$F6	Alt'v'
$37	'7'	$B7	ShF8 Alt'7'	$77	'w'	$F7	Alt'w'
$38	'8'	$B8	ShF9 Alt'8'	$78	'x'	$F8	'°' Alt'x'
$39	'9'	$B9	ShF10 Alt'9'	$79	'y'	$F9	Alt'y'
$3A	':'	$BA	ShF11 Alt':'	$7A	'z'	$FA	Alt'z'
$3B	';'	$BB	ShF12 Alt';'	$7B	Alt'/'== {	$FB	AltCtrl'/'
$3C	'<'	$BC	Alt'<'	$7C	Alt'>'== |	$FC	AltCtrl'>'
$3D	'='	$BD	Alt'='	$7D	Alt'='== }	$FD	AltCtrl'='
$3E	'>'	$BE	Alt'>'	$7E	Alt'*'== ~	$FE	AltCtrl'*'
$3F	'?'	$BF	Alt'?'	$7F	DEL	$FF	----------

6.2.6 Optischer Drehgeber (Jog-Shuttle, Object 2300_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2300	1	*Jog_Shuttle*	0000...FFFF	-	Daten-strom	r/w

Bedeutung der Variablen *Jog_Shuttle*:

Der optische Drehgeber gibt einen Wertebereich von -32768 bis +32767 zurück. Ein Lesezugriff auf das Objekt 2300_h gibt die aktuelle Stellung des Drehgebers zurück.

Mit einem Schreibzugriff kann dem Drehgeber ein neuer Referenzpunkt in dem oben angegebenen Wertebereich zugeordnet werden.

Wertebereich:

Jog_Shuttle = 0000 ... $FFFF_h$

6.2.7 LCD_Data (2400_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2400	0	*LCD_Data*	00...FF	-	Daten-strom	wo

Bedeutung der Variablen *LCD_Data*:

Die LCD-Daten und ESC-Sequenzenwerden als Zeichenkette (max. 2^{32} Bytes) über einen SDO-Upload Transfer übertragen.
Die ESC-Sequenzen werden ab Seite 28 ausführlich beschrieben.

Die folgende Tabelle zeigt beispielhaft den Transfer der Zeichenkette “Hallo Welt”

Der Datenaustausch beim Upload des Strings muß wie in der folgenden Tabelle dargestellt, ablaufen (alle Zahlenwerte hexadezimal). In den übergebenen Kommandos müssen immer 8 Datenbytes übertragen werden.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Bemerkung
Client:	**21**	**00**	**24**	**00**	**0A**	00	00	00	download request index 240_{0h} sub-index=0, len=10_D
CAN-LCT:	**30**	**00**	**24**	00	00	00	00	00	download ackn., len = 0
Client:	**00**	**48**	**61**	**6C**	**6C**	**6F**	**20**	**57**	segment download request
CAN-LCT:	**20**	00	00	00	00	00	00	00	‘Hallo_W’, len = 7
Client:	**15**	**65**	**6C**	**74**	00	00	00	00	segment download request
CAN-LCT:	**30**	00	00	00	00	00	00	00	‘elt’, len = 3, ended

Eine ausführliche Beschreibung des Domain Downloads ist der CiA DS-301 (SDO-Protocols) zu entnehmen.

6.2.8 Graphic_Command (2410_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2410	0	*Graphic_Command*	00... FF	-	Byte/ Daten-strom	wo

Bedeutung der Variablen *Graphic_Command*:

Die Grafik-Kommandos können Byte-weise oder als SDO-Upload/Download übertragen werden. Je nach Grafik-Kommando werden mindestens zwei Parameter-Bytes benötigt, die direkt an das Kommando-Byte anschließend übertragen werden müssen.

Kommando	Parameter					
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
SETPIX	*xpos*	*ypos*	-	-	-	-
LINE	*x1pos*	*y1pos*	*x2pos*	*y2pos*	-	-
GAREA	*x0*	*y0*	*dx*	*dy*	*mode*	*data*

Beschreibung der Grafikkommandos:

SETPIX

- setzt Pixel an der definierten Position

Byte	Inhalt	Wertebereich
0	SETPIX-Kommando-Code	10_h ... Pixel aus [Hex] 11_h ...17_h ... Pixel an [Hex]
1	*xpos*	0...239 [Dez]
2	*ypos*	0...127 [Dez]

LINE

- zieht Linie von Punkt *x1pos/y1pos* zu *x2pos/y2pos*

Byte	Inhalt	Wertebereich
0	LINE-Kommando-Code	20 ... Pixel aus [Hex] 21 ...27 ... Pixel an [Hex]
1	*x1pos*	0...239 [Dez]
2	*y1pos*	0...127 [Dez]
3	*x2pos*	0...239 [Dez]
4	*y2pos*	0...127 [Dez]

GAREA

- bearbeitet Bildausschnitt (Füllen mit verschiedenen Inhalten)

Byte	Inhalt	Wertebereich
0	GAREA-Kommando-Code	40 [Hex]
1	*x0* *x-Koordinate für den Nullpunkt des Bildausschnitts.*	0, 6, 12, 18, ... , 234 [Dez]
2	*y0* *y-Koordinate für den Nullpunkt des Bildausschnitts*	0, 1, 2, 3, ... , 127 [Dez]
3	*dx* *Breite des Bildausschnitts in x-Richtung*	6, 12, 18, ... , 234 [Dez]
4	*dy* *Höhe des Bildausschnitts in y-Richtung*	1, 2, 3, ... , 128 [Dez]
5	*mode* *siehe unten*	8000...E000 [Hex] *)1
6	*data* *siehe unten*	6 Bit Pixel-Information *)1

Die Parameter *mode* und *data* werden auf den folgenden Seiten ausführlich beschrieben.

mode... Über diesen Parameter wird festgelegt, welche Aktionen mit dem Bildausschnitt durchgeführt werden sollen:

mode		Bedeutung
[Bit]	[Hex]	
0000 0000 0000 0000	00 00	Ausschnitt lesen
1000 0000 0000 0000	80 00	Ausschnitt mit Inhalt von Feld (*data*) beschreiben
1100 0000 0000 0000	C0 00	Ausschnitt mit '0' füllen
1100 1000 0000 0000	C8 00	Ausschnitt mit '1' füllen
1110 0000 0000 0000	E0 00	alle Zellen des Ausschnitt mit dem Inhalt des ersten Elementes des Feldes (*data*) beschreiben (write pattern)

data... In diesem eindimensionalem Feld werden die Daten für die Pixel abgelegt. Jeweils sechs Pixel in x-Richtung werden in einem Byte beschrieben, wobei die obersten zwei Bits nicht ausgewertet werden (siehe auch folgende Abbildung). Durch die Aufteilung von sechs Pixeln/Byte ergeben sich die zulässigen Werte für x0 und dx mit (0, 6, 12, 18, ...,234).

Die Zellen des *data*-Feldes werden den Koordinaten des Bildausschnitts wie folgt zugeordnet: (x0/y0, x1/y0, x2/y0,...x0/y1, x1/y1,.....xn/ym)

Die Mindestanzahl der Elemente des Feldes *data* wird von der Größe des Bildausschnitts festgelegt:

$$\text{Länge (data)} = dy \cdot \frac{dx}{6}$$

Die folgende Abbildung zeigt die Parameter, die den Bildausschnitt definieren. Außerdem ist als Beispiel für die Pixel-Zuordnung zu den Elementen im Feld *data*, die Pixel-Gruppe mit den (absoluten) Koordinaten x = x0 und y = y0 dargestellt.

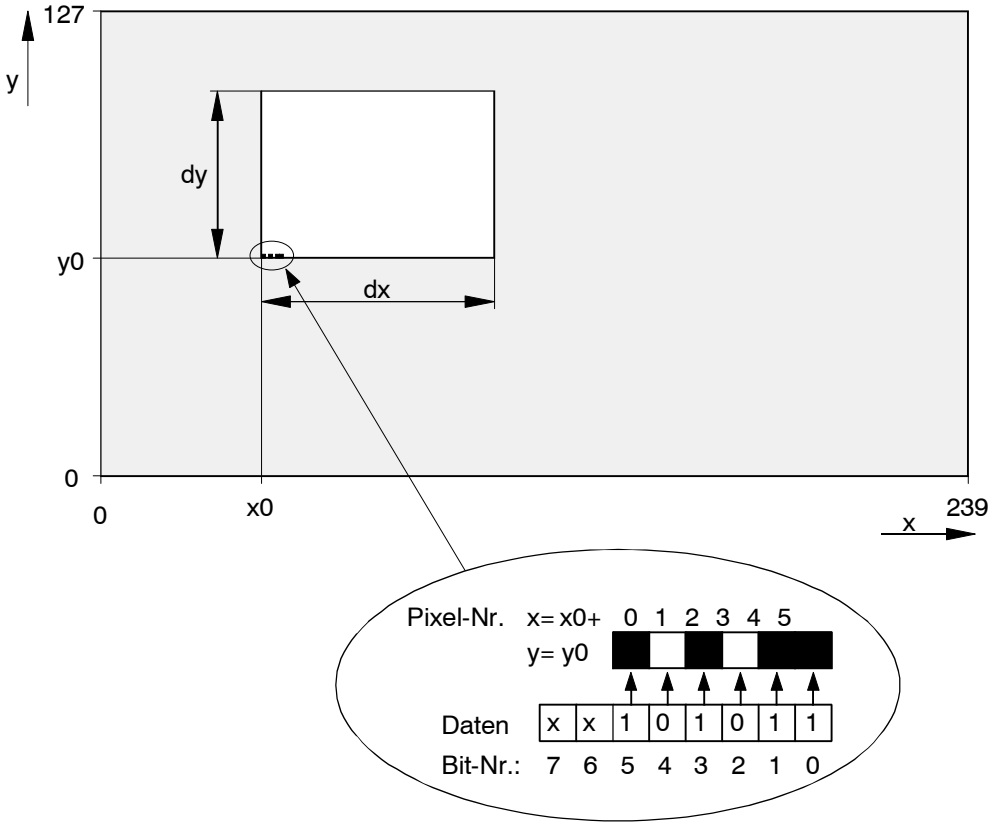


Abb. 2: Definition eines Bildausschnittes und Pixel-Zuordnung

6.2.9 User_Character (2420_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2420	0	*User_Character*	00... FF	-	Daten-strom	wo

Bedeutung der Variablen *User_Character*:

Über die Variable *User_Character* können die 96 vordefinierten Sonderzeichen des LCD-Controllers durch eigene, selbst definierte Zeichen ersetzt werden. Für die Definition eines Zeichens sind 9 Bytes erforderlich, die über SDO-Upload Transfers in das CAN-LCT-Modul geladen werden.

Die Zeichensätze und das Definieren eigener Sonderzeichen werden ab Seite 35 näher beschreiben.

6.2.10 Graphic_Font_16x12 (2430_h), Graphic_Font_8x6 (2401_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2430	0	*Graphic_Font_16x12*	00... FF	-	Daten-strom	wo

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2431	0	*Graphic_Font_8x6*	00... FF	-	Daten-strom	wo

Bedeutung der Variablen *Graphic_Font_16x12 und Graphic_Font_8x3*:
Mit Hilfe dieser Variablen lassen sich die Grafik-Fonts 16x12 und 8x6 darstellen. Hierzu muß die Grafikebene aktiviert sein.
Die Daten müssen über SDO-Upload Transfers in das CAN-LCT-Modul geladen.

Als Referenzpunkt eines Zeichens des Grafik-Fonts wird die obere linke Ecke herangezogen. Der Referenzpunkt ist z.B. bei der direkten Cursor-Adressierung über `ESC Y` von Bedeutung.

Im Gegensatz zum Text/Attribut-Mode erfolgt beim Zeichnen der Character des Grafik-Fonts kein automatischer Zeilenumbruch am Zeilenende (kein auto wrap). Hier sind die Steuerzeichen `CR` und `LF` vom Anwender zu setzen.
Es erfolgt ebenfalls kein 'scrollen' des Bildschirms, wenn die letzte Zeile mit `CR&LF` umgebrochen wird. Die letzte Zeile würde in diesem Fall immer wieder überschrieben werden.

Hinweis:
Zur Zeit ist die Anzeige für den Font 08x06 als zusätzliche *Textebene ohne Attribute* implementiert. Daher wird diese Ebene auch nicht beim Empfang des Kommandos "Grafikebene löschen" gelöscht. Die Positionierung der Zeichen erfolgt spalten- und zeilenweise. Es gibt in dieser Ebene keinen Auto-Wrap.
Ein Kommando `ESC H, ESC J` über den FONT08x06-Pfadnamen löscht nur die Textebene für die 8x6 Zeichen. Beide Ebenen werden über die Ausgabe von $1A_h$ gelöscht.

Ein Kommando `ESC H, ESC J` über den FONT16x12-Pfadnamen löscht nur die Grafikebene für die 16x12 Zeichen.

6.2.11 Contrast (2440_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2440	0	*Contrast*	C1 ... 3F	-	Byte	wo

Bedeutung der Variablen *Contrast*:
Über diese Variable kann der Kontrast des LC-Displays in 63 Stufen verändert werden. Werte größer ‘0’ erhöhen den Kontrast um die übergebene Anzahl Stufen und Werte unter ‘0’ verringern den Kontrast entsprechend.

Bei Auslieferung der Karte arbeitet die Kontrasteinstellung mit einem Default-Wert, der im I²C-EEPROM abgelegt ist. Der neue geänderte Wert wird nach der Änderung ebenfalls im EEPROM abgelegt und überschreibt den Default-Wert.

Beispiele:

Contrast = 03_h ... Der Kontrast wir um drei Stufen erhöht.
Contrast = $F6_h$ ... Der Kontrast wir um zehn Stufen verringert.

6.2.12 Display_Temperature (2450_h)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2450	0	*Display_Temperature*	00... FF	-	Byte	ro

Bedeutung der Variablen *Display_Temperature*:
Die Temperatur auf der CAN-LCT-Platine wird für die dynamische Anpassung des Kontrastes des LC-Displays verwendet. Sie kann auch vom Anwender über die Variable *Display_Temperature* gelesen werden.

Wertebereich:

Display_Temperature = -20 ˚C ... +70 ˚C (EC_h ...00_h...46_h)

6.2.13 HW_Serial_No ($2F00_h$)

Index [Hex]	Subindex [Dez]	Beschreibung	Wertebereich [Hex]	Default [Dez]	Datentyp	R/W
2F00	0	*HW_Serial_No*	0000 ... FFFF	-	Word	ro

Bedeutung der Variablen *HW_Serial_No*:
Diese Variable enthält die Seriennummer des CAN-LCT-Moduls. Es ist die selbe Nummer, die auf dem Seriennummer-Label auf der Platine angebracht ist. Die Buchstabenkennung des Produktions-Loses ist nicht in *HW_Serial_No* enthalten.

Wertebereich:

HW_Serial_No = 0 ... $FFFF_h$

7. Grafikfähige Displays

7.1 Organisation des Displays

Die Ansteuerung des Displays erfolgt über sogenannte Ebenen: Es sind zwei Text-Ebenen, eine Attribut-Ebene und eine Grafik-Ebene vorhanden.
Über die Text-Ebenen können ein fest installierter Text-Font und frei definierbare Zeichen angezeigt werden.
Mit Hilfe der Attribut-Ebene kann die Anzeige z.B. zum Blinken gebracht werden oder es können inverse Darstellungen angezeigt werden.
In der Grafik-Ebene sind Zeichenoperationen, wie z.B. das Setzen von Pixeln oder das Zeichnen von Linien möglich. Außerdem können in der Grafik-Ebene Texte über sogenannte 'Grafik-Fonts' angezeigt werden.

Die gleichzeitige Darstellung verschiedener Ebenen erfolgt im wesentlichen in zwei Kombinationen (siehe auch folgende Abbildung 'Kombinationsmöglichkeiten der Display-Ebenen'): Dem sogenannten 'Text/Attribut-Mode' und dem 'Grafik-Mode'.

7.2 Aufteilung der Ebenen

Der Speicher des Display-Controllers wird von der Software in verschiedene 'Ebenen' aufgeteilt. Die Ebenen bieten verschiedene Funktionen zur Anzeige und Konfiguration. Die Selektion der einzelnen Ebenen erfolgt über ESC-Sequenzen. Die ESC-Sequenzen werden in den folgenden Kapiteln beschrieben.

Die folgende Abbildung zeigt die zwei Kombinationsmöglichkeiten der Ebenen mit einigen Eigenschaften und Funktionen des ESC-Sequenzen.

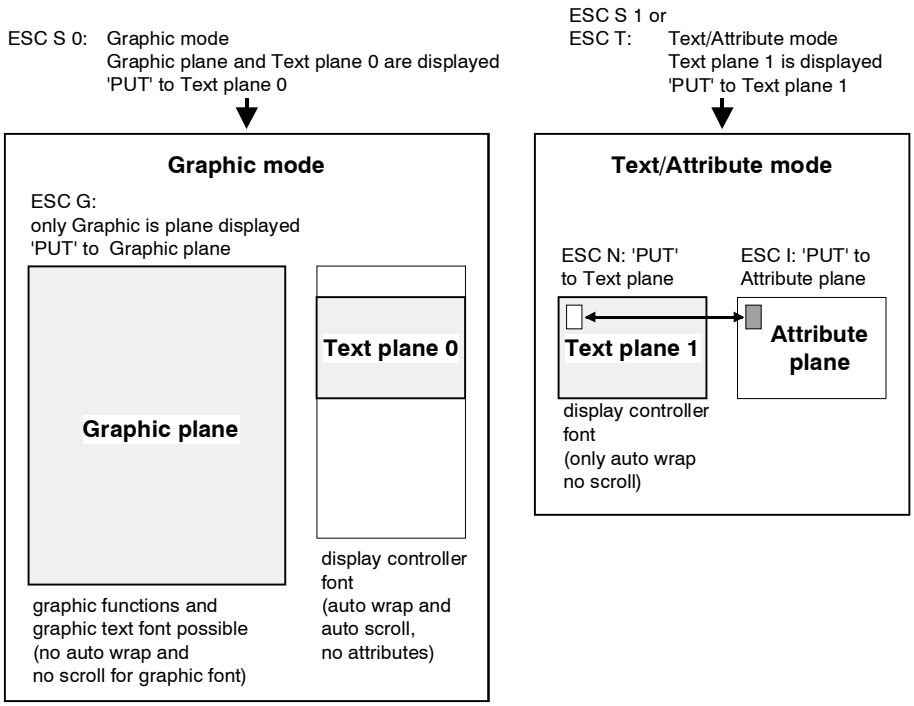


Abb. 3: Kombinationsmöglichkeiten der Display-Ebenen

Der rechte Block der Abbildung zeigt die Ebenen des sogenannten 'Text/Attribut-Modes'. In diesem Mode wird die Text-Plane 1 angezeigt. Jeder Zelle der Text-Plane ist eine Zelle der Attribut-Plane zugeordnet. Über diese Zellen können den einzelnen Zeichen Attribute wie 'blinkend', 'inverse Darstellung' usw. zugewiesen werden.

Werden in eine Zeile mehr Zeichen geschrieben als anzeigbar sind, so erfolgt ein automatischer Zeilenumbruch. Ist die letzte Zeile gefüllt, so erfolgt kein 'scrollen', d.h. nach CR und LF wird die letzte Zeile wieder überschrieben.

Die Zeichen, die in der Text-Plane dargestellt werden können, entsprechen dem internen Font des Display-Controllers. Außerdem können frei definierbare Zeichen (die nach RESET oder ABORT mit Sonderzeichen vorbesetzt sind) angezeigt werden.

Der linke Block zeigt den Grafik-Mode. Hier können eine Grafik-Ebene und eine Text-Ebene gemeinsam (überlagernd) oder alternativ angezeigt werden. Für die Fonts der Text-Ebene gilt das gleiche wie im Text/Attribut-Mode. Den Zeichen können jedoch keine Attribute zugewiesen werden, weil die Pointer

der Attribut-Ebene für die Grafik-Ebene benötigt werden.

Im Gegensatz zur Text-Plane 1 ist bei der Text-Plane 0 neben dem automatischen Zeilenumbruch außerdem das automatische 'Scrollen' des Bildschirms beim Überschreiten der untersten Zeile implementiert. Das Scrollen wird möglich, weil für die Text-Plane ein größerer Speicherbereich reserviert wurde und auf die Attribut-Zuweisung verzichtet wurde.

In der Grafik-Ebene können neben grafischen Operationen, wie Linien ziehen oder Pixel-Felder bearbeiten, ebenfalls Texte (im folgenden auch 'Grafik-Fonts' genannt) eingefügt werden. Die Zeichen des Grafik-Fonts werden über die Datenstation '`FONTxyz`' aufgerufen. Für die Grafik-Fonts ist kein automatischer Zeilenumbruch und kein automatisches Scrollen möglich.

7.3 Konfiguration und Attribut-Ebene

Die Steuerung der Anzeige erfolgt ähnlich wie bei Terminals über ESC-Sequenzen.

Um einen Text mit Attributen zu versehen, ist wie folgt vorzugehen:

1. Text-Ebene 0 auswählen
2. Text eingeben
3. Umschaltung auf Attribut-Ebene
4. Cursor auf die gewünschte Textposition einstellen
5. Attribute eingeben

Der Text wird immer mit dem jeweiligen Attribut verknüpft, wenn die Attribut-Ebene und die Text-Ebene 0 aktiviert sind.

Zur Eingabe der Attribute wird eine Zeichenkette an die CAN-LCT gesandt, deren untere vier Bits das Attribut enthalten.

Steuer-Codes der Attribut-Ebene:

Bit 7...4	Bit 3	Bit 2	Bit 1	Bit 0	Hex	ASCII	Funktion	
0011	0	0	0	0	30	'0'	normales Display	(auch bei 31_h ,32_h ,37_h)
0011	0	1	0	1	35	'5'	inverse Textdarstellung	(auch bei 34_h ,36_h)
0011	0	0	1	1	33	'3'	keine Anzeige	(auch bei $3B_h$)
0011	1	0	0	0	38	'8'	blinkendes normales Display	(auch bei 39_h ,$3A_h$,$3F_h$)
0011	1	1	0	1	3D	'='	blinkendes inverses Display	

ESC-Sequenzen zur Steuerung und Konfiguration:

`ESC S` *n1*	Auswahl der Text-Plane und somit des Anzeige-Modes: *n1* = 0 -▸ Text-Plane 0 -▸ Text/Attribut-Mode *n1* = 1 -▸ Text-Plane 1 -▸ Grafik-Mode
`ESC T`	Text-Mode aktivieren (Text- und Attribut-Ebene benutzt) Mode-Set = 84_h, Display-Mode = $9D_h$ oder $9F_h$ Die Grafik-Ebene wird nicht dargestellt.
`ESC I`	Attribut-Ebene aktiv (nur für Text/Attribut-Mode relevant) (für Normal, Invers, Blinkend, Display Aus)
`ESC N`	Text-Ebene aktiv (nur für Text/Attribut-Mode relevant)
`ESC G`	Grafik-Mode aktivieren (nur eine Grafik-Plane) Mode-Set = 80_h, Display-Mode = 98_h
`ESC O`	Cursor Ein
`ESC Q`	Cursor Aus
`ESC P` *n1*	Höhe des Cursors (Cursor Pattern); *n1*: 1-8
`ESC R`	Software-RESET (alle Default-Werte neu gesetzt) Default-Werte: Auto-Wrap (Zeilenumbruch), Cursor auf Position 1 (Home), Text-, Attribut- und Grafik-Ebene gelöscht

`ESC L` *n1*

Display-Mode-Setzen
Über den Display-Mode werden der Cursor, die Text- und die Grafik-Plane zur Anzeige ein- oder ausgeschaltet.

n1 = 1001 xxxx	Bedeutung
1001 xxx0 1001 xxx1	kein blinkender Cursor blinkender Cursor
1001 xx0x 1001 xx1x	Cursor ausgeschaltet Cursor eingeschaltet
1001 x0xx 1001 x1xx	Text aus Texte ein
1001 0xxx 1001 1xxx	Grafik aus Grafik ein

`ESC M` *n1*

Mode-Set
Hier kann festgelegt werden, wie die verschiedenen Ebenen dargestellt, bzw. verwendet werden.

n1 = 1000 xxxx	Bedeutung
1000 x000	Grafik und Text 'ORED', d.h. Grafik und Text überlagern sich
1000 x001	Grafik und Text 'EXORED', d.h. Pixel werden nur dort aktiviert, wo entweder nur Text- oder nur Grafik-Pixel aktiv sind
1000 x011	Grafik und Text 'ANDED', d.h. Pixel werden nur dort aktiviert, wo gleichzeitig Text- und Grafik-Pixel aktiv sind
1000 x100	Grafik-Ebene dient als Attribute-Ebene des Textes
1000 0xxx 1000 1xxx	interne Character Erzeugung externe Character Erzeugung (nicht implementiert)

ESC-Sequenzen zum Editieren der Anzeige:

ESC-Sequenz	Funktion		
		Position Text/ Attribut-Mode	Position Grafik-Mode
`ESC A`	Cursor nach oben	y = y + 1	y = 1 + h
`ESC B`	Cursor nach unten	y = y - 1	y = y - h
`ESC C`	Cursor nach rechts	x = x + 1	x = x + b
`ESC D`	Cursor nach links	x = x -1	x = x - b
`ESC Y` *n1 n2*	Direkte Cursor-Adressierung, *n1, n2* binär	n1 = 1...8 (16) n2 = 1...40	n1 = 1...63 (128) n2 = 1...240, (Referenzpunkt der Zeichen ist oben links)
`ESC '='` *m1 m2*	Direkte Cursor-Adressierung, immer Zeichen-orientiert *m1* = `TOCHAR (zeile-1+32)`, *m2* = `TOCHAR (spalte-1+32)`	m1 = 1...8(16) m2 = 1...40	m1 = 1...8 (16) m2 = 1...40
`ESC H`	Cursor an Position 1 (Home: Zeile 1, Spalte 1)		
`ESC E`	Grafik-Ebene löschen		
`ESC @`	Grafik- oder Attribut-Ebene löschen		
`ESC J`	Löschen bis zum Ende des Bildschirms		(nur im Text/Attribut-Mode)
`ESC K`	Löschen bis zum Zeilenende		(nur im Text/Attribut-Mode)
`ESC W`	automatischer Zeilenumbruch am Zeilenende (Auto Wrap)		(nur im Text/Attribut-Mode)
`ESC V`	kein Auto Wrap		(nur im Text/Attribut-Mode)

Verwendete Abkürzungen:

x...	x-Position auf Display, je nach Mode in Pixeln oder Zeichen angegeben
y...	y-Position auf Display, je nach Mode in Pixeln oder Zeichen angegeben
b...	Breite eines Zeichens eines 'Grafik-Fonts' in Pixeln
h...	Höhe eines Zeichens eines 'Grafik-Fonts' in Pixeln
n1, m1...	einzugebende Zeilenposition
n2, m2...	einzugebende Spaltenposition

Außerdem werden folgende Steuerzeichen ausgewertet:

`<cr>`	00_h	Carriage Return
`<lf>`	$0A_h$	Neue Zeile
`<home>`	$1A_h$	Home & Clear

7.4 Zeichensatz der Text-Ebenen

Der Display-Controller besitzt für die Text-Plane 128 fest programmierte Character. Diese Zeichen entsprechen den Zeichen 32 bis 159 (dezimal) der IBM Zeichenumsetzungstabelle 437. Ein Abdruck dieser Tabelle findet sich auf der folgenden Seite.

Ebenfalls aus der Zeichentabelle können die 96 vordefinierten Sonderzeichen entnommen werden. Sie entsprechen den Zeichen 160...255 (dezimal) aus der Zeichentabelle.

Diese Zeichen sind, wie die 128 unveränderlichen Zeichen, im EPROM abgelegt. Sie werden nach einem ABORT oder RESET in das RAM des Display-Controllers geladen. Dort können die veränderlichen Zeichen vom Anwender editiert werden. Die geänderten Zeichen gehen nach einem RESET wieder verloren.

Für jeden veränderlichen Character steht ein Feld von 6 Pixel Breite und 8 Pixel Höhe zur Verfügung. Ein Zeichen wird durch 8 Bytes beschrieben, die jeweils für eine der 8 Pixel-Zeilen stehen. Innerhalb eines Bytes sind die 6 waagerechten Pixel rechtsbündig kodiert.

IBM Zeichenumsetzungstabelle 437

Dem Anwender stehen die Zeichen 32...159 als unveränderliche Zeichen und die Zeichen 160... 255 als veränderliche Sonderzeichen zur Verfügung. Die Zeichen 0...31 können nicht aufgerufen werden!

Legende zu den Feldern der folgenden Tabellen:

Grafische Darstellung des Zeichens	Dezimale Zeichennummer (dreistellig)
Bezeichnung nach ISO 8879:1986	Entsprechender Unicode-Zeichen-Wert

IBM-Zeichensatz-Tabelle 437, Teil 1:

	000	▶	016		032	0	048	@	064	P	080	`	096	p	112
		rtrif	25B6	sp	0020		0030	commat	0040		0050	grave	0060		0070
☺	001	◀	017	!	033	1	049	A	065	Q	081	a	097	q	113
	263A	ltrif	25C0	excl	0021		0031		0041		0051		0061		0071
☻	002	↕	018	"	034	2	050	B	066	R	082	b	098	r	114
	263B	varr	2195	quot	0022		0032		0042		0052		0062		0072
♥	003	‼	019	#	035	3	051	C	067	S	083	c	099	s	115
hearts	2665		203C	num	0023		0033		0043		0053		0063		0073
♦	004	¶	020	$	036	4	052	D	068	T	084	d	100	t	116
diams	2666	para	00B6	dollar	0024		0034		0044		0054		0064		0074
♣	005	§	021	%	037	5	053	E	069	U	085	e	101	u	117
clubs	2663	sect	00A7	percnt	0025		0035		0045		0055		0065		0075
♠	006	▬	022	&	038	6	054	F	070	V	086	f	102	v	118
spades	2660		25AC	amp	0026		0036		0046		0056		0066		0076
•	007	↨	023	'	039	7	055	G	071	W	087	g	103	w	119
bull	2022		21A8	apos	0027		0037		0047		0057		0067		0077
◘	008	↑	024	(	040	8	056	H	072	X	088	h	104	x	120
	25D8	uarr	2191	lpar	0028		0038		0048		0058		0068		0078
○	009	↓	025	)	041	9	057	I	073	Y	089	i	105	y	121
	25CB	darr	2193	rpar	0029		0039		0049		0059		0069		0079
◙	010	→	026	*	042	:	058	J	074	Z	090	j	106	z	122
	25D9	rarr	2192	ast	002A	colon	003A		004A		005A		006A		007A
♂	011	←	027	+	043	;	059	K	075	[	091	k	107	{	123
male	2642	larr	2190	plus	002B	semi	003B		004B	lsqb	005B		006B	lcub	007B
♀	012	∟	028	,	044	<	060	L	076	\	092	l	108	|	124
female	2640		221F	comma	002C	lt	003C		004C	bsol	005C		006C	verbar	007C
	013		029	-	045	=	061	M	077	]	093	m	109	}	125
sung	266A	harr	2194	hyphen	002D	equals	003D		004D	rsqb	005D		006D	rcub	007D
♫	014	▲	030	.	046	>	062	N	078	^	094	n	110	~	126
	266B	utrif	25B2	period	002E	gt	003E		004E	circ	005E		006E	tilde	007E
☼	015	▼	031	/	047	?	063	O	079	_	095	o	111	⌂	127
	263C	dtrif	25BC	sol	002F	quest	003F		004F	lowbar	005F		006F		2302

IBM-Zeichensatz-Tabelle 437, Teil 2:

Ç	128	É	144	á	160	▒	176	└	192	ð	208	Ó	224	-	240
Ccedil	00C7	Eacute	00C9	aacute	00E1	blk14	2591		2514	eth	00F0	Oacute	00D3	shy	00AD
ü	129	æ	145	í	161	▓	177	┴	193	Ð	209	ß	225	±	241
uuml	00FC	aelig	00E6	iacute	00ED	blk12	2592		2534	ETH	00D0	szlig	00DF	plusmn	00B1
é	130	Æ	146	ó	162	▓	178	┬	194	Ê	210	Ô	226	=	242
eacute	00E9	AElig	00C6	oacute	00F3	blk34	2593		252C	Ecirc	00CA	Ocirc	00D4		2017
â	131	ô	147	ú	163	│	179	├	195	Ë	211	Ò	227	¾	243
acirc	00E2	ocirc	00F4	uacute	00FA		2502		251C	Euml	00CB	Ograve	00D2	frac34	00BE
ä	132	ö	148	ñ	164	┤	180	─	196	È	212	õ	228	¶	244
auml	00E4	ouml	00F6	ntilde	00F1		2524		2500	Egrave	00C8	otilde	00F5	para	00B6
à	133	ò	149	Ñ	165	Á	181	┼	197	ı	213	Õ	229	§	245
agrave	00E0	ograve	00F2	Ntilde	00D1	Aacute	00C1		253C	inodot	0131	Otilde	00D5	sect	00A7
å	134	û	150	ª	166	Â	182	ã	198	Í	214	µ	230	÷	246
aring	00E5	ucirc	00FB	ordf	00AA	Acirc	00C2	atilde	00E3	Iacute	00CD	micro	00B5	divide	00F7
ç	135	ù	151	º	167	À	183	Ã	199	Î	215	þ	231	¸	247
ccedil	00E7	ugrave	00F9	ordm	00BA	Agrave	00C0	Atilde	00C3	Icirc	00CE	thorn	00FE	cedil	00B8
ê	136	ÿ	152	¿	168	©	184	╚	200	Ï	216	Þ	232	°	248
ecirc	00EA	yuml	00FF	iquest	00BF	copy	00A9		255A	Iuml	00CF	THORN	00DE	deg	00B0
ë	137	Ö	153	®	169	╣	185	╔	201	┘	217	Ú	233	¨	249
euml	00EB	Ouml	00D6	reg	00AE		2563		2554		2518	Uacute	00DA	uml	00A8
è	138	Ü	154	¬	170	║	186	╩	202	┌	218	Û	234	·	250
egrave	00E8	Uuml	00DC	not	00AC		2551		2569		250C	Ucirc	00DB	middot	00B7
ï	139	Ø	155	½	171	╗	187	╦	203	█	219	Ù	235	¹	251
iuml	00EF	oslash	00F8	frac12	00BD		2557		2566	block	2588	Ugrave	00D9	sup1	00B9
î	140	£	156	¼	172	╝	188	╠	204	▄	220	ý	236	³	252
icirc	00EE	pound	00A3	frac14	00BC		255D		2560	lhblk	2584	yacute	00FD	sup3	00B3
ì	141	Ø	157	¡	173	¢	189	═	205	¦	221	Ý	237	²	253
igrave	00EC	Oslash	00D8	iexcl	00A1	cent	00A2		2550	brvbar	00A6	Yacute	00DD	sup2	00B2
Ä	142	×	158	«	174	¥	190	╬	206	Ì	222	¯	238	■	254
Auml	00C4	times	00D7	laquo	00AB	yen	00A5		256C	Igrave	00CC	macr	00AF	squf	25A0
Å	143	ƒ	159	»	175	╕	191	¤	207	▀	223	´	239		255
Aring	00C5	fnof	0192	raquo	00BB		2510	curren	00A4	uhblk	2580	acute	00B4	nbsp	00A0

Beispiel: Definition einer (fett gezeichneten) Linien-Ecke

	Bit 5	4	3	2	1	0	Eingabe [Hex]
Byte 0:	0	0	0	0	0	0	00
Byte 1:	0	0	0	0	0	0	00
Byte 2:	0	0	0	0	0	0	00
Byte 3:	0	0	**1**	**1**	**1**	**1**	0F
Byte 4:	0	0	**1**	**1**	**1**	**1**	0F
Byte 5:	0	0	**1**	**1**	0	0	0C
Byte 6:	0	0	**1**	**1**	0	0	0C
Byte 7:	0	0	**1**	**1**	0	0	0C

Die obersten zwei Bits eines Bytes werden nicht ausgewertet. Sie können beliebig gesetzt werden.

Die Character-Definition und das Senden des Characters zum Display erfolgt durch die Übertragung eines Strings. Der gesendete String besteht aus neun Bytes, die wie folgt ausgewertet werden:

Byte-Nummer	Auswertung
1	Nummer des Zeichens in der Zeichentabelle (160...255)
2	Byte 0 des Pixel-Feldes
3	Byte 1 :
4	Byte 2 :
5	Byte 3 :
6	Byte 4 :
7	Byte 5 :
8	Byte 6 :
9	Byte 7 des Pixel-Feldes

Es können auch mehr als neun Bytes übertragen werden. Jeweils neun aufeinanderfolgende Bytes werden als ein Zeichen interpretiert. Überschüssige Bytes werden ignoriert.

Da der Display-Controller bei der Darstellung der Zeichen auf dem Display den Pointer auf die Zelle in der Zeichentabelle als Referenz nimmt, kann auch zuerst der Character an das Display gesendet und nachträglich die Zeichendefinition verändert werden. Das dargestellte Zeichen würde sich dann ebenfalls verändern.