



ifm electronic



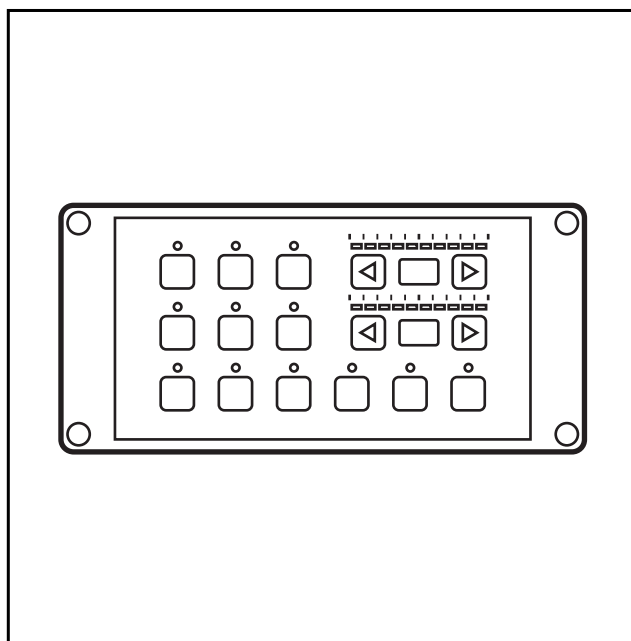
Geräte-Handbuch Device manual

ecomat 1000[®]

CAN Tastaturmodul
CPM 100

CAN key pad module
CPM 100

CR1500



Sachnr. 7390289/02 04/2009

DEUTSCH

ENGLISH

Sicherheitshinweise



Diese Beschreibung ist Bestandteil des Gerätes. Sie enthält Texte und Abbildungen zum korrekten Umgang mit dem Modul und muss vor einer Installation oder dem Einsatz gelesen werden.

Befolgen Sie die Angaben der Dokumentation. Nichtbeachten der Hinweise, Verwendung außerhalb der nachstehend genannten bestimmungsgemäßen Verwendung, falsche Installation oder Handhabung können Beeinträchtigungen der Sicherheit von Menschen und Anlagen zur Folge haben.

Das Gerät darf nur von einer Elektrofachkraft eingebaut, angeschlossen und in Betrieb gesetzt werden.

Schalten Sie das Gerät extern spannungsfrei bevor Sie irgendwelche Arbeiten an ihm vornehmen. Schalten Sie ggf. auch unabhängig versorgte Ausgangslastkreise ab.

Bei Fehlfunktion des Geräts oder bei Unklarheiten setzen Sie sich bitte mit dem Hersteller in Verbindung. Eingriffe in das Gerät können schwerwiegende Beeinträchtigungen der Sicherheit von Menschen und Anlagen zur Folge haben. Sie sind nicht zulässig und führen zu Haftungs- und Gewährleistungsausschluss.

Inhalt

Bestimmungsgemäße Verwendung/Funktion	Seite 3
Technische Daten	Seite 4
Maße	Seite 5
Tastenbeschriftung	Seite 5
Montage und elektrischer Anschluss	Seite 6
Tasten- und LED-Betriebsarten	Seite 8
Parameter- und EMCY-Objekt-Übersicht	Seite 10
Wartung, Instandsetzung und Entsorgung	Seite 11
Konformitätserklärung	Seite 11
Objektverzeichnisse	
Herstellerspezifische Profile; Index 2000 bis 5FFF	Seite 12
Kommunikationsprofile; Index 1000 bis 1FFF	Seite 14
Programmierung	Seite 18
Begriffe und Abkürzungen	Seite 24
Notizen	Seite 50

Bestimmungsgemäße Verwendung / Funktion

Das Tastatur- und Anzeigemodul CR1500 für CANopen Netzwerke kann als Auf- oder Einbaugerät in Führerstände, Bedienkonsolen oder Schaltschränken eingesetzt werden. Die Auswertung und Parametrierung der Tasten und LEDs erfolgt gemäß IEC 61131-3 innerhalb des Steuerungsprogramms.

Funktionsbibliotheken für das Programmiersystem CoDeSys unterstützen die Einbindung der Tastaturfunktionen in das Applikationsprogramm.

CANopen-Kommunikation im Überblick

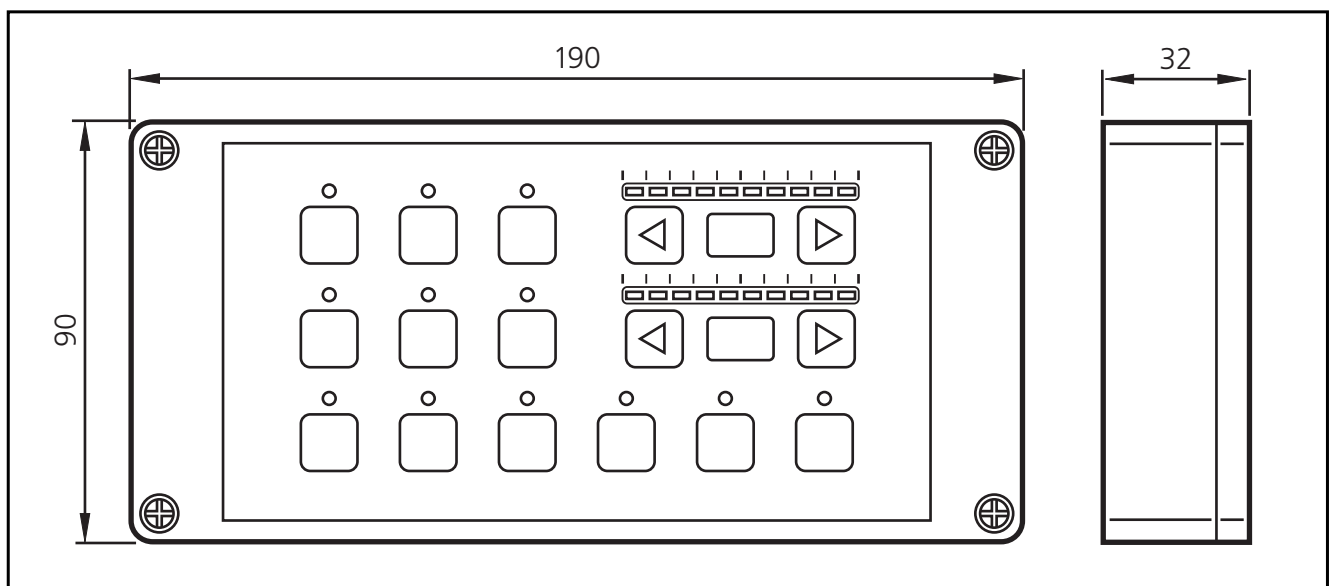
- Es sind 1 Server SDO und 2 Default PDOs gemäß CiA DS 401 eingerichtet. Die Default-Identifizierer sind entsprechend des „Predefined connection set“ vergeben.
- Die COB-IDs und die Übertragungsart (synch/asynch/cyclic) für das Trans PDO (Tastenstatus) und das Rec PDO (LED-Status) sind konfigurierbar. Die Übertragungsart wird spannungsausfallsicher gespeichert. Geänderte PDOs (PDO-linking) werden nicht spannungsausfallsicher gespeichert.
- Das Modul erwartet ein Synch-Objekt. Der CAN Identifizierer des Synch-Objektes ist konfigurierbar. Nach einer Änderung wird der ID automatisch spannungsausfallsicher gespeichert.
- Das Modul unterstützt „Node guarding“. Die „Guard time“, der „Life time factor“ und der CAN Identifizierer des Guard Objektes sind konfigurierbar und werden spannungsausfallsicher gespeichert.
- Das Modul unterstützt „Heartbeat“. Die „Producer heartbeat time“ und „Consumer heartbeat time“ sind konfigurierbar. Es können bis zu 4 Netzwerkteilnehmer überwacht werden.
- Das Modul generiert ein Emergency Objekt. Der COB-ID des EMCY-Objektes ist konfigurierbar.
- Das Modul speichert die 4 zuletzt aufgetretenen Fehler. Abgelegt wird der Fehlercode des jeweiligen Emergency Objektes.
- Das Modul unterstützt eine Reset-Funktion; d.h. die Belegung der Parameter mit den werkseitigen Default-Werten* nach Aufforderung.

*) Werkseitige Default-Einstellungen siehe „Parameterliste“, Seite 10

Technische Daten

Betriebsspannung	10...30 V DC; SELV
Stromaufnahme	200 mA (24 V DC) (Die Stromaufnahme im Fehlerfall muss durch geeignete Maßnahmen auf 1 A begrenzt werden)
Funktionstasten Bedientasten	12, über Anwenderprogramm auswertbar 4 Pfeiltasten, über Anwenderprogramm auswertbar
Tastenbeschriftung	3 wechselbare Einschubstreifen
Status-LEDs LED-Ketten	12, über Anwenderprogramm frei ansteuerbar 2 x 10-stellig, rot/grün, Betriebsart parametrierbar
Schnittstelle	CAN Interface (ISO 11898)
Baudrate	20 kBit/s...1 MBit/s (Defaulteinstellung 125 kBit/s)
Kommunikationsprofil Geräteprofil	CANopen, CiA DS301 V4.0 herstellerspezifisch
CAN	Full-CAN 2.0
Node-ID (Default)	hex 20 (= 32)
Betriebsstemperatur Lagertemperatur	-20...+85°C -30...+85°C
Schutzart / Schutzklasse	IP64 / III
Gehäusematerial	Noryl
Abmessungen (BxHxT) Bauhöhe bei Wandmontage Bauhöhe bei Schaltschrankmontage Einbautiefe bei Schaltschrankmontage Maximale Dicke der Fronttafel	190 x 90 x 32 mm 32 mm 7 mm 25 mm 2,5 mm
Anschluß	5 schraubenlose WAGO-Klemmen
Anschlußbelegung	1: CAN_H, 2: CAN_L, 3: GND, 4: GND, 5: +24 V

Maße

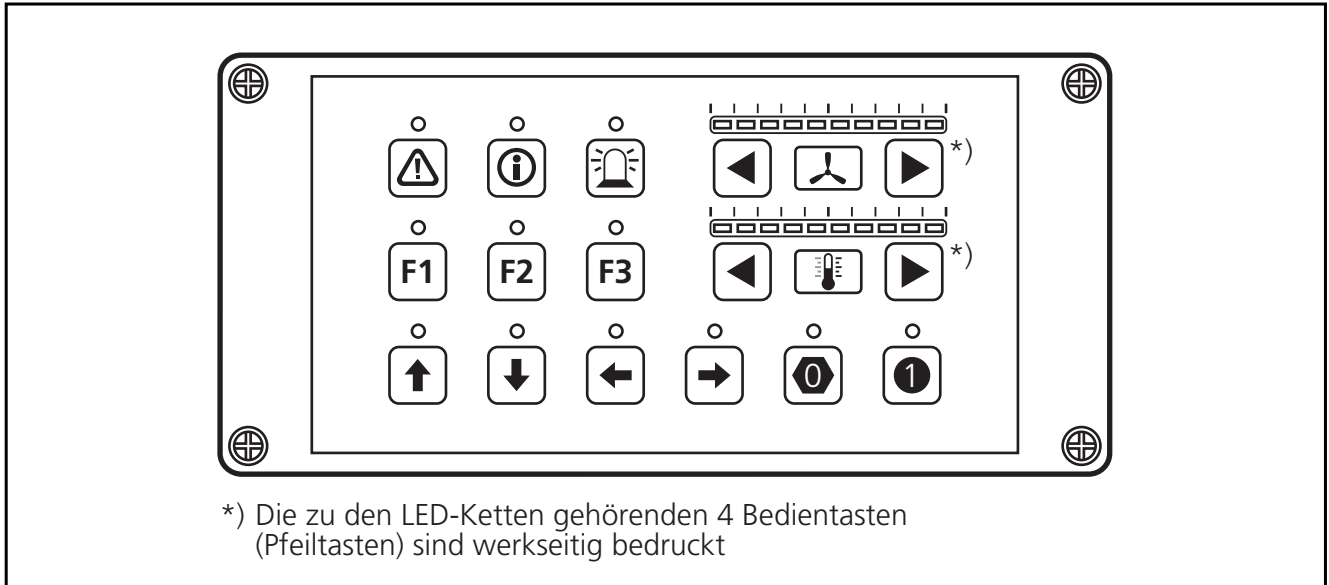


Tastenbeschriftung

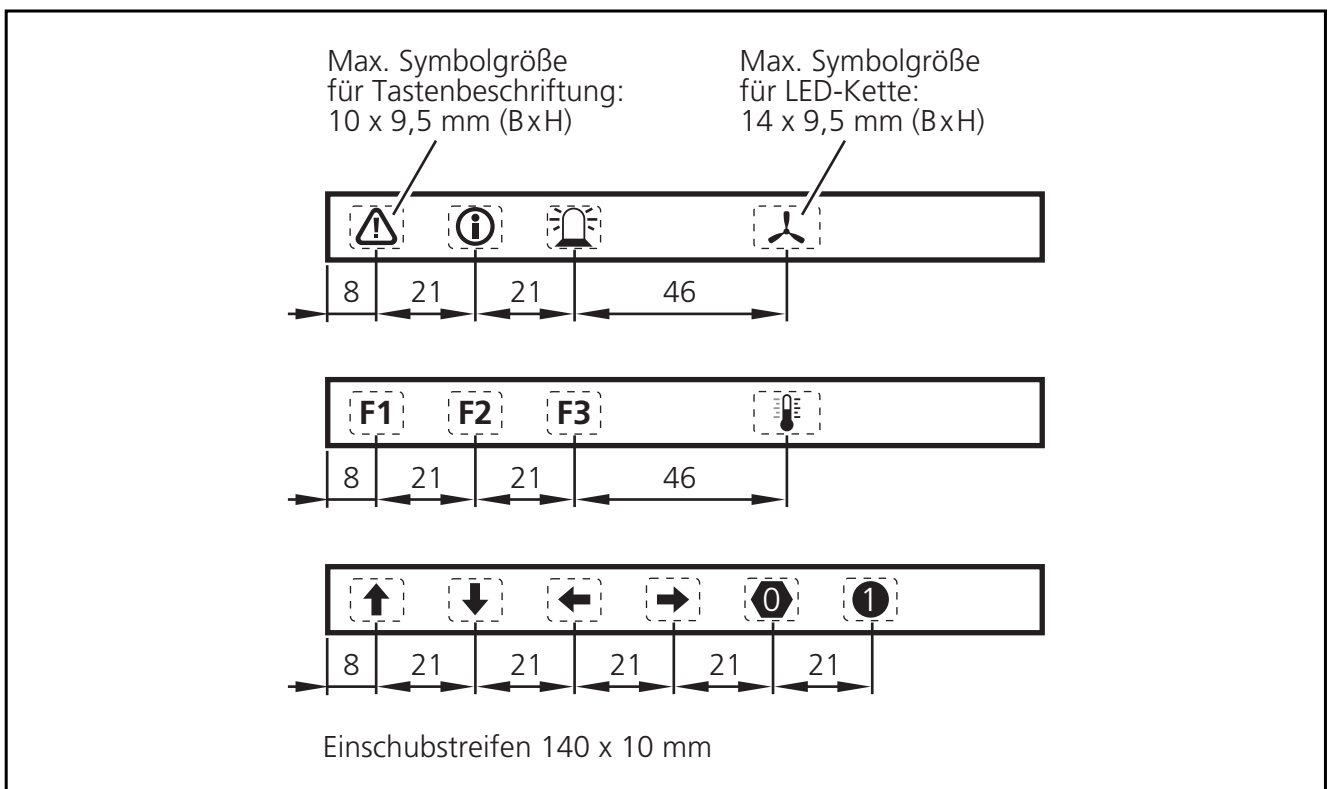
Die applikationsspezifische Beschriftung der Funktionstasten und LED-Ketten erfolgt über Einsschubstreifen. Diese Streifen können mit einem PC und einem beliebigen Text- oder Grafikprogramm erstellt und ausgedruckt werden.

Um das Einschieben der Streifen zu vereinfachen, sollte der Ausdruck vorzugsweise auf stärkerem Papier erfolgen. Besonders geeignet sind Fotopapiere auf Kunststoffbasis.

Beschriftungsbeispiel



Maße der Beschriftungsstreifen



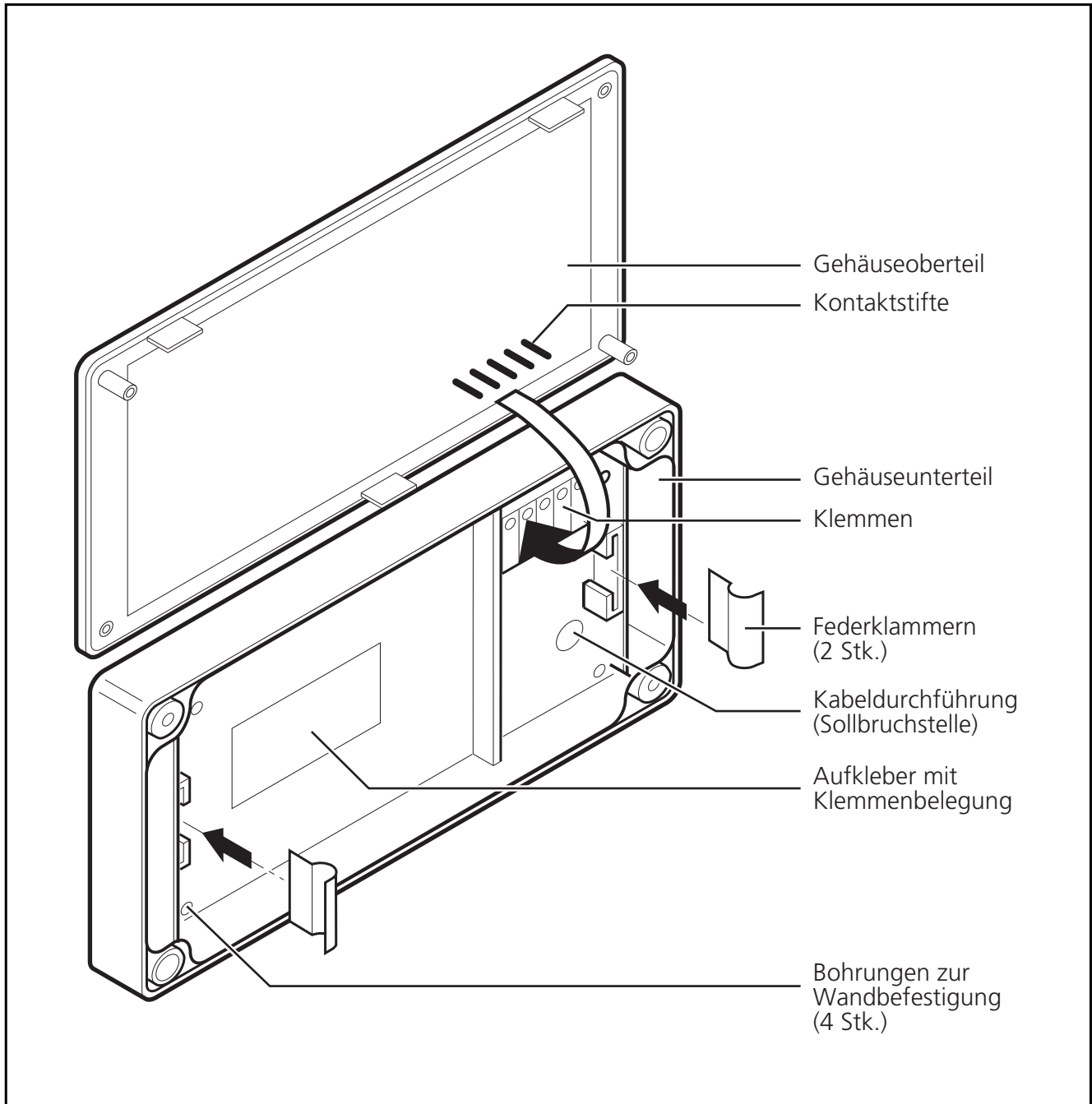
Montage und elektrischer Anschluss



Das Modul darf nur von einer Elektrofachkraft eingebaut, angeschlossen und in Betrieb gesetzt werden.



Das Tastaturmodul enthält C-MOS-Bauelemente die durch elektrostatische Entladung beschädigt werden können. Berühren Sie deshalb vor dem Öffnen des Moduls ein geerdetes Teil.



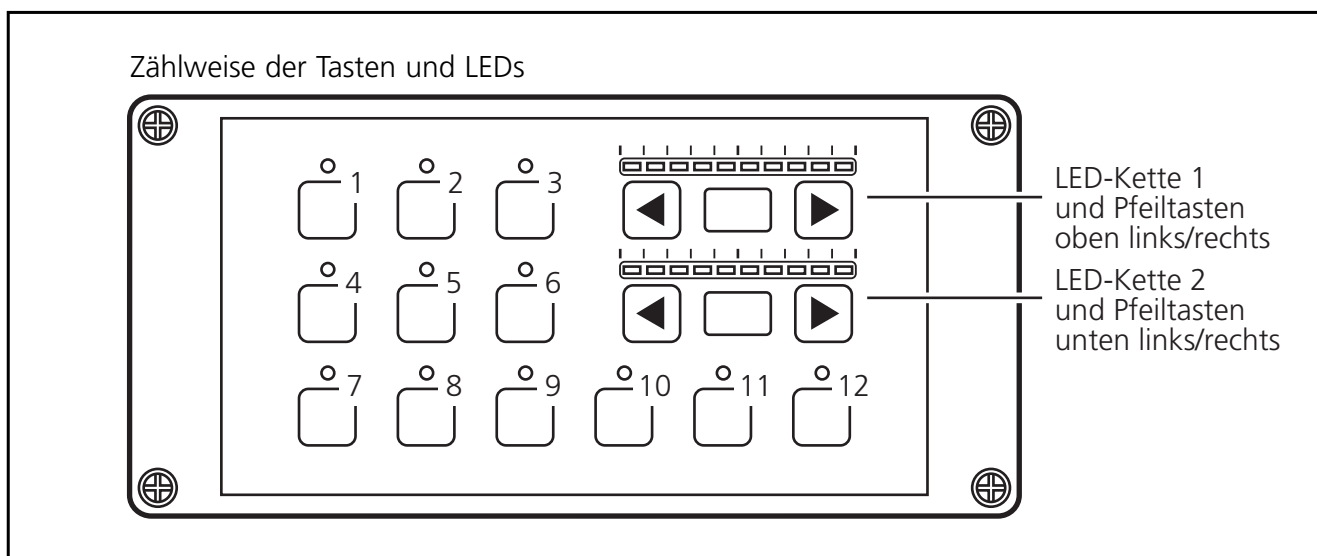
■ Schaltschrank- bzw. Bedienpultmontage (Einbau)

1. Bereiten Sie mit Hilfe der beiliegenden Montageschablone (#5310674) einen Ausschnitt mit 4 Bohrungen in der Fronttafel des Schaltschranks oder des Bedienpultes vor.
2. Lösen Sie die 4 Schrauben des Tastaturmoduls und heben Sie das Gehäuseoberteil vom Gehäuseunterteil ab.
3. Brechen Sie die Sollbruchstelle für die Kabeldurchführung im Boden des Gehäuseunterteils aus.
4. Setzen Sie die beiliegende PG-Kabeldurchführung ein.
5. Setzen Sie das Gehäuseunterteil von der Schaltschrankinnenseite in den Ausschnitt ein.
6. Schieben Sie die beiliegenden Federklammern von der Schaltschrankaußenseite in die Halterungen rechts und links im Gehäuseunterteil und befestigen Sie das Gehäuseunterteil im Ausschnitt.
7. Führen Sie das Kabel durch die Kabeldurchführung.
8. Stellen Sie die Anschlüsse gemäß Aufkleber im Gehäuseunterteil her.
9. Setzen Sie das Gehäuseoberteil so auf das Gehäuseunterteil, dass die Kontaktstifte des Gehäuseoberteils in die Klemmen im Gehäuseunterteil eingreifen.
10. Schrauben Sie das Gehäuseoberteil auf dem Gehäuseunterteil fest.

■ Wandmontage (Aufbau)

1. Lösen Sie die 4 Schrauben des Tastaturmoduls und heben Sie das Gehäuseoberteil vom Gehäuseunterteil ab.
2. Brechen Sie die Sollbruchstelle für die Kabeldurchführung im Boden des Gehäuseunterteils aus oder bohren Sie eine Kabeldurchführung in die unterer Seitenwand innerhalb des Anschlußraumes und schneiden Sie ein entsprechendes PG-Gewinde (siehe beiliegende Montageschablone).
3. Setzen Sie gegebenenfalls die beiliegende PG-Kabeldurchführung ein.
4. Führen Sie das Kabel durch die Kabeldurchführung.
5. Befestigen Sie das Gehäuseunterteil mit vier Schrauben o.ä. an der Wand.
6. Stellen Sie die Anschlüsse gemäß Aufkleber im Gehäuseunterteil her.
7. Setzen Sie das Gehäuseoberteil so auf das Gehäuseunterteil, dass die Kontaktstifte des Gehäuseoberteils in die Klemmen im Gehäuseunterteil eingreifen.
8. Schrauben Sie das Gehäuseoberteil auf dem Gehäuseunterteil fest.

Tasten- und LED-Betriebsarten



■ Tasten

Jede der 12 Funktions- und 4 Pfeiltasten ermöglicht folgende Betriebsarten:

- Taster-Betrieb (Tasten-Bit ist TRUE solange Taste gedrückt wird)
- Taster-Betrieb mit Verzögerung (Tasten-Bit wird erst TRUE, wenn die Taste x ms gedrückt wurde).

Die 12 Funktionstasten unterstützen zusätzlich die Betriebsart „Toggle“:

- Toggle-Betrieb (Tasten-Bit ist TRUE bis Taste ein weiteres Mal gedrückt wird).
- Toggle-Betrieb mit Verzögerung (Tasten-Bit wechselt den Zustand, wenn die Taste x ms gedrückt wurde).

Um die Auswertung der Tastenverzögerung (Filter) im Taster-Betrieb zu unterstützen, wird die Auswertung des Tastendrucks über zwei Bit pro Taste realisiert. Das erste Bit wird sofort nach dem Tastendruck übertragen (entsprechend der CANopen Betriebsart). Das zweite Bit wird erst nach Ablauf der eingestellten Verzögerungszeit TRUE. Anschließend wird der Tastenstatus erneut übertragen (diesmal beide Bits TRUE).

Im Toggle-Betrieb, mit und ohne aktivem Filter, werden stets beide Bits übertragen. Der Tastenstatus wird in den ersten 4 Byte des Transmit PDOs abgebildet.

z.B. Byte 0:	0b xxxx xx 01	Funktionstaste 1 gedrückt
	0b xxxx xx 11	Funktionstaste 1 gedrückt und Verzögerungszeit abgelaufen
	0b xxxx 01 xx	Funktionstaste 2 gedrückt
	0b xxxx 11 xx	Funktionstaste 2 gedrückt und Verzögerungszeit abgelaufen
	usw.	

Die Betriebsart der Tasten und die Verzögerungszeit wird über das Objektverzeichnis parametrisiert (ab Idx 2000). Die IEC-Funktion „CR1500“ der Programmiersoftware CoDeSys gibt den Tastenstatus an die Applikation weiter.

Bitte beachten: Die eingestellte Verzögerungszeit ist für alle Tasten gültig. Es kann maximal 1 Taste gleichzeitig gedrückt werden.

■ Status-LEDs

Jede Status-LED kann individuell angesteuert werden. Der LED-Anzeigemodus wird nicht über SDO parametrierbar, sondern kann direkt per Receive PDO aus der Applikation angesteuert werden. Die ersten 3 Byte des Rec PDOs beinhalten den LED-Status. 4 Betriebszustände stehen zur Verfügung:

- AUS
- Blinken (Zeit 1, einstellbar 0...65535 ms; Default 500 ms = 2 Hz)
- Blinken (Zeit 2, einstellbar 0...65535 ms; Default 200 ms = 5 Hz)
- EIN

Darüber hinaus signalisieren die Status-LEDs folgende Betriebszustände:

Betriebszustände (Zustandswechsel)	Status-LEDs
Aus → PREOPERATIONAL	leuchten 1 sek.
PREOPERATIONAL → OPERATIONAL	5 x blinkend, anschließend Zustand wie programmiert
Kommunikationsfehler	permanent schnell blinkend (5 Hz)

■ LED-Ketten

Die Betriebsart der LED-Ketten wird über das Objektverzeichnis parametrierbar (ab Idx 2002). Folgende Betriebsarten sind möglich:

Betriebsart	Beschreibung
1	rotes Einzel-Segment auf grünem Leuchtband
2	grünes Einzel-Segment auf rotem Leuchtband
3	rotes Einzel-Segment
4	grünes Einzel-Segment
5	rote Segmentkette auf grünem Leuchtband
6	grüne Segmentkette auf rotem Leuchtband
7	rote Segmentkette
8	grüne Segmentkette
9	rote Segmentkette mit Farbumschlagpunkt (hier 5)
10	grüne Segmentkette mit Farbumschlagpunkt (hier 7)
= rot = grün	

Die Ansteuerung der LED-Segmente erfolgt über einen Analogwert (0...10) in Byte 3 und 4 des Rec PDOs. Über die IEC-Funktion „NORM“ besteht die Möglichkeit Soll- und Istwerte für dies LED-Ketten zu normieren.

Parameter- und EMCY-Objekt-Übersicht

Über die Funktion „Restore“ (s. Objektverzeichnis, Idx 1011) können die Parameter (Ausnahme Baudrate und Node-ID) mit den werkseitig hinterlegten Default-Werten belegt werden. Diese sind dann nach dem nächsten Einschalten der Versorgungsspannung gültig.

Parameterliste

Parameter	Index im Objektverzeichnis	Defaultwert (werkseitig eingestellt)	Änderung automatisch gesichert	Änderung wirksam
Herstellerspezifische Profile; Index 2000 bis 5FFF				
Tastenkongfiguration	2000	0x00 (Taster-Betriebsart)	Ja	sofort
Verzögerungszeit Tasten (Filter)	2001	0x00 (0 ms = Off) (gültig für alle Tasten)	Ja	sofort
Betriebsart LED-Ketten	2002	0x01 (rotes Einzel-Segment auf grünem Leuchtband)	Ja	sofort
Umschlag. Farbwechsel	2003	0x01	Ja	sofort
Blinkzeit 1 Status-LED	2004 01	0x01F4 (500 ms = 2 Hz)	Ja	sofort
Blinkzeit 2 Status-LED	2004 02	0x00C8 (200 ms = 5 Hz)	Ja	sofort
Node-ID	20F0, 20F1	0x20 (= 32)	ja	nach Reset
Baudrate	20F2, 20F3	0x04 (= 125 kBit/s)	ja	nach Reset
Kommunikationsprofile; Index 1000 bis 1FFF				
COB-ID SYNC	1005	0x80	ja	sofort
Communication Cycle	1006	0x00 (Off)	ja	nach Pre-Op
Guard Time	100C	0x00 (Off)	Ja	sofort
Life Time Factor	100D	0x00	Ja	sofort
COB-ID EMCY	1014	0x80 + Node ID	Ja	sofort
Heartbeat time 1	1016 01	0x00 (Off)	Ja	sofort
Heartbeat time 2	1016 02	0x00 (Off)	Ja	sofort
Heartbeat time 3	1016 03	0x00 (Off)	Ja	sofort
Heartbeat time 4	1016 04	0x00 (Off)	Ja	sofort
Producer heartbeat time	1017	0x00 (Off)	Ja	sofort
COB-ID Rec PDO	1400 01	0x200 + Node ID	nein	sofort
Trans Type Rec PDO	1400 02	0x01 (synchron)	Ja	sofort
COB-ID Trans PDO	1800 01	0x180 + Node ID	nein	sofort
Trans Type Trans PDO	1800 02	0xFF (asynchron)	Ja	sofort

Life Time Factor 0 wird als 1 interpretiert.

Das erste Guardprotokoll wird als „Start Guarding“ gewertet, auch wenn zu diesem Zeitpunkt das Guarding noch nicht aktiviert ist (Guardtime = 0).

EMCY Objekte

Folgende Fehlercodes gemäß DSP-401 bzw. DSP-301 werden unterstützt:

EMCY Code	Error Reg	Zusatz Code	Beschreibung
0x6100	0x11	0x00	„Internal Software“ Überlauf einer Rx-Queue; Frequenz der Rx PDOs zu groß. Reset nur extern, über Eintrag in Idx 1003 00
0x6101	0x11	0x00	„Internal Software“ Überlauf einer Tx-Queue; Gerät kommt nicht auf den Bus. Reset nur extern, über Eintrag in Idx 1003 00
0x6200	0x81	0x00	„User Software“ Es wurde mehr als 1 Taste gleichzeitig gedrückt. Auswertung des Tastenstatus ist nicht möglich. Reset bei Lösen der Taste.
0x8100	0x11	0x00	„Monitoring“ (Guarding Error) Für „guard time“ * „life time factor“ wird kein Guard-Objekt empfangen. life time factor 0 entspricht 1 Reset bei node again
0x8200	0x11	0x00	„Monitoring“ (Synch Error) Für „communication cycle“ wird kein Synch-Objekt empfangen; (nur in OPERATIONAL) Reset bei Synch-OBJ bzw. PREOP

Wartung, Instandsetzung und Entsorgung

Da innerhalb des Moduls keine vom Anwender zu wartenden Bauteile enthalten sind, darf das Gehäuse nicht geöffnet werden. Die Instandsetzung des Moduls darf nur durch den Hersteller durchgeführt werden.

Die Entsorgung muss gemäß den nationalen Umweltvorschriften erfolgen.

Konformitätserklärung

Das CE-Zeichen wird angebracht auf Basis der EMV-Richtlinie EMV 89/336/EWG sowie der Niederspannungsrichtlinie NS73/23/EWG.

Objektverzeichnis

Herstellerspezifische Profile; Index 2000 bis 5FFF

Index	S-Idx	Name	Typ	Default	Beschreibung
2000	0x00	Konfiguration der Tastenbetriebsart	u8, ro	0x10	Anzahl der Einträge (= Anzahl der Tasten = 16)
	0x01	Taste 1	u8, rw	0x00	0 = Taster-Betriebsart 1 = Toggle-Betriebsart
	⋮	⋮	⋮	⋮	⋮
	0x0A	Taste 10	u8, rw	0x00	0 = Taster-Betriebsart 1 = Toggle-Betriebsart
	0x0B	Taste 11	u8, rw	0x00	0 = Taster-Betriebsart 1 = Toggle-Betriebsart
	0x0C	Taste 12	u8, rw	0x00	0 = Taster-Betriebsart 1 = Toggle-Betriebsart
	0x0D	Pfeiltaste oben links	u8, ro	0x00	0 = Taster-Betriebsart Toggle-Betriebsart wird nicht unterstützt
	0x0E	Pfeiltaste oben rechts	u8, ro	0x00	0 = Taster-Betriebsart Toggle-Betriebsart wird nicht unterstützt
	0x0F	Pfeiltaste unten links	u8, ro	0x00	0 = Taster-Betriebsart Toggle-Betriebsart wird nicht unterstützt
	0x10	Pfeiltaste oben rechts	u8, ro	0x00	0 = Taster-Betriebsart Toggle-Betriebsart wird nicht unterstützt
2001	0x00	Verzögerungszeit für die Tastenauswertung	u16, rw	0x00	Einstellung in ms Bereich = 0...65535 ms (gültig für alle Tasten)
2002	0x00	Betriebsart der LED-Ketten	u8, ro	0x02	Anzahl der eingebundenen LED-Ketten
	0x01	Betriebsart LED-Kette 1	u8, rw	0x01	<p>Wertebereich 0...10 (0x00...0A)</p> <p>0 = LED-Kette nicht aktiv</p> <p>1 = rotes Einzel-Segment auf grünem Leuchtband (Default)</p> <p>2 = grünes Einzel-Segment auf rotem Leuchtband</p> <p>3 = rotes Einzels-Segment</p> <p>4 = grünes Einzels-Segment</p> <p>5 = rote Segmentkette auf grünem Leuchtband</p> <p>6 = grüne Segmentkette auf rotem Leuchtband</p> <p>7 = rote Segmentkette auf grünem Leuchtband</p> <p>8 = grüne Segmentkette auf rotem Leuchtband</p> <p>9 = rote Segmentkette mit Farbumschlag bei def. Umschlagpunkt</p> <p>10 = grüne Segmentkette mit Farbumschlag bei def. Umschlagpunkt (siehe auch Seite 9, LED-Ketten)</p> <p>1 </p> <p>2 </p> <p>3 </p> <p>4 </p> <p>5 </p> <p>6 </p> <p>7 </p> <p>8 </p> <p>9 </p> <p>10 </p> <p>■ = rot ▨ = grün</p>
	0x02	Betriebsart LED-Kette 2	u8, rw	0x01	Wertebereich 0...10 (0x00...0A) wie LED-Kette 1 (siehe auch Seite 9, LED-Ketten)

Objektverzeichnis

Herstellerspezifische Profile; Index 2000 bis 5FFF

Index	S-Idx	Name	Typ	Default	Beschreibung
2003	0x00	Umschlagpunkt für Farbwechsel	u8, ro	0x02	Anzahl der eingebundenen LED-Ketten
	0x01	Umschlagpunkt Farbwechsel LED-Kette 1	u8, rw	0x01	Farbumschlagpunkt LED-Kette 1 Wertebereich 1...10 (0x00...0A) (Werte < 1 und > 10 werden intern als 1 bzw. 10 ausgewertet) Nur möglich bei Betriebsart 9 und 10! (siehe auch Seite 9, LED-Ketten)
	0x02	Umschlagpunkt Farbwechsel LED-Kette 2	u8, rw	0x01	Farbumschlagpunkt LED-Kette 2 Wertebereich 1...10 (0x00...0A) (Werte < 1 und > 10 werden intern als 1 bzw. 10 ausgewertet) Nur möglich bei Betriebsart 9 und 10! (siehe auch Seite 9, LED-Ketten)
2004	0x00	Blinkzeit der Status-LEDs	u8, ro	0x02	Anzahl der eingebundenen Zeit-Objekte
	0x01	Blinkzeit 1	u16, rw	0x01F4	Blinkzeit 1 (Periodendauer) in ms Wertebereich 0...65535 ms (Default 500 ms / 2 Hz)
	0x02	Blinkzeit 2	u16, rw	0x00C8	Blinkzeit 2 (Periodendauer) in ms Wertebereich 0...65535 ms (Default 200 ms / 5 Hz)
20F0	0x00	Einstellung Node-ID	u8, rw	0x20 (= 32)	Node-ID unter dem das Modul im CANopen Netz angesprochen wird
20F1	0x00	Node-ID			
<p>Eine Node-ID Änderung wird nur dann übernommen, wenn in den Einträgen 20F0 und 20F1 der gleiche geänderte Wert eingetragen ist. Werte kleiner 1 / größer 127 werden nicht übernommen; die bestehende Einstellung bleibt erhalten. Damit die neuen Einträge gültig werden, muss nach dem Einstellen der Node ID ein Reset ausgelöst werden (Aus-/Einschalten des Moduls).</p>					
20F2	0x00	Einstellung Baudrate	u8, rw	0x04	Baudrate des CAN-Netzes 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud 4 = 125 kBaud (Default) 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
20F3	0x00	Baudrate			
<p>Eine Baudrate Änderung wird nur dann übernommen, wenn in den Einträgen 20F2 und 20F3 der gleiche geänderte Wert eingetragen ist. Werte größer 7 werden nicht übernommen; die bestehende Einstellung bleibt erhalten. Damit die neuen Einträge gültig werden, muss nach dem Einstellen der Baudrate ein Reset ausgelöst werden (Aus-/Einschalten des Moduls).</p>					

Erläuterung der Abkürzungen:

Ob... = binärer Zahlenwert

0x... = hexadezimaler Zahlenwert

rw = read-write

ro = read only

u8 = unsigned 8 bit

(siehe auch „Begriffe und Abkürzungen“, Seite 24)

Objektverzeichnis

Kommunikationsprofile; Index 1000 bis 1FFF

Index	S-Idx	Name	Typ	Default	Beschreibung
1000	0x00	device type	u32, ro	0x0000	herstellerspezifisch
1001	0x00	error register	u8, ro	0x00	Bitcodiert gemäß Prof 301; unterstützt wird: 0b 0000 0000 kein Fehler 0b x00x 0001 generic error 0b x001 000x communication error 0b 100x 000x manufacturer specific
1003	0x00	pre-defined errorfield	u8, ro	0x02	Es wird eine Fehlerliste mit 4 Einträgen unterstützt
	0x01-4	error history	u32, ro	0x00	Aufgetretener Fehler; codiert entsprechend EMCY Liste; der zuletzt aufgetretene Fehler steht jeweils in Sub-Index 1
1005	0x00	COB-ID SYNC	u32, rw	0x80000080	- Modul erwartet Synch Meldung (Bit 31 = 1) - Modul generiert keine Synch Meldung (Bit 30 = 0) - 11 Bit Identifier System (Bit 29 = 0) - Identifier der Synch Meldung
1006	0x00	communication cycle period	u32, rw	0x00000000	max. Zeit zwischen 2 Synch. Objekten in µs; Nutzauflösung = 1 ms
1008	0x00	device name	str, ro	CR1500	Gerätebezeichnung
1009	0x00	HW Version	str, ro	x.x	Hardware Version
100A	0x00	SW Version	str, ro	x.x	Software Version
100B	0x00	Node-ID	u32, ro		nur zur Abfrage
100C	0x00	guard time	u16, rw	0x0000	Zeit in ms Das Modul erwartet innerhalb dieser Zeit ein "node guarding" des Netz-Masters Wird hier der Wert 0 eingetragen, wird diese Funktion nicht unterstützt.
100D	0x00	life time factor	u8, rw	0x00	Wenn für "guard time" * "life time" kein "node guarding" empfangen wird, generiert das Modul ein EMCY. Das Produkt aus "guard time" * "life time" muss in dem Bereich zwischen 0...65535 liegen.
1010	0x00	number of save-options	u8, ro	0x01	Anzahl der Optionen "Sichern"
	0x01	store parameters	u32, rw	0x02	Alle Parameter werden bei einer Änderung automatisch gesichert.
1011	0x00	number of restore-options	u8, ro	0x01	Anzahl der Optionen "Reset"
	0x01	restore default parameters	u32, rw	0x01	Wird hier der String "load" eingetragen, werden die Parameter mit den werkseitigen Voreinstellungen belegt und sind nach dem nächsten Reset gültig.

Objektverzeichnis

Kommunikationsprofile ; Index 1000 bis 1FFF

Index	S-Idx	Name	Typ	Default	Beschreibung
1014	0x00	COB-ID EMCY	u32, rw	0x40000080 +Node ID	<ul style="list-style-type: none"> - Modul reagiert nicht auf fremde EMCY Mess. (Bit 31 = 0) - Modul generiert EMCY Mess. (Bit 30 = 1) - 11 Bit ID (Bit 29 = 0) - ID = 0x80 + Node ID CAN-Identifizierer kann vom Benutzer geändert werden.
1016	0x00	Consumer heartbeat time	u8, ro	0x04	Anzahl der überwachten Geräte
	0x01	Consumer heartbeat time device 1	u32, rw	0x00	00nn tttt tttt: Überwachungszeit in ms nn: Modulnummer des zu überwachenden Gerätes (wenn nn oder tttt gleich 0, findet keine Überwachung statt)
	0x02	heartbeat time device 2	u32, rw	0x00	wie Consumer heartbeat time device 1
	0x03	heartbeat time device 3	u32, rw	0x00	wie Consumer heartbeat time device 1
	0x04	heartbeat time device 4	u32, rw	0x00	wie Consumer heartbeat time device 1
1017	0x00	Producer heartbeat time	u16, rw	0x00	Heartbeat-Zeit des Gerätes in ms
1200	0x00	Server SDO	u8, ro	0x02	Anzahl der Einträge
	0x01	COB-ID Rec SDO	u32, ro	0x600 + Node ID	<ul style="list-style-type: none"> - SDO ist gültig (Bit 31 = 0) - CAN-ID des Receive SDOs
	0x02	COB-ID Trans SDO	u32, ro	0x580 + Node ID	<ul style="list-style-type: none"> - SDO ist gültig (Bit 31 = 0) - CAN-ID des Transmit SDOs
1400	0x00	Receive PDO <i>Sollwert Status-LEDs und LED-Ketten</i>	u8, ro	0x02	Anzahl der Einträge Rec PDO; (LED-Status)
	0x01	COB-ID Rec PDO	u32, rw	0x200 + Node ID	<ul style="list-style-type: none"> - PDO ist gültig (Bit 31 = 0) - CAN-ID des Rec PDOs
	0x02	Trans Type Rec PDO <i>Sollwert Status-LEDs und LED-Ketten</i>	u8, rw	0x01	<ul style="list-style-type: none"> 0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Ausgänge werden erst nach „n“ Synch Objekten aktualisiert n = 0x01 (1)...0xF0 (240) 0xFC nicht implementiert 0xFD nicht implementiert 0xFE = asynch man. spec. event; Ausgänge werden sofort aktualisiert 0xFF = asynch device profile event; Ausgänge werden sofort aktualisiert

Objektverzeichnis

Kommunikationsprofile; Index 1000 bis 1FFF

Index	S-Idx	Name	Typ	Default	Beschreibung
1600	0x00	Mapping Receive PDO	u32, rw	0x03	Anzahl der eingebundenen Applikations-Objekte zur Ansteuerung der Status-LEDs und der LED-Ketten
	0x01	PDO mapping for the 1st application object <i>Sollwert Status-LEDs</i>	u32, rw	0x6200 01	im Idx 6200 01 stehen 3 Byte LED-Status, Ansteuerung erfolgt über jeweils 2 Bit, z.B. Byte 0: 0b xxxx xx 00 LED 1 AUS 0b xxxx xx 01 LED 1 Blinkzeit 1 0b xxxx xx 10 LED 1 Blinkzeit 2 0b xxxx xx 11 LED 1 EIN 0b xxxx 00 xx LED 2 AUS 0b xxxx 01 xx LED 2 Blinkzeit 1 0b xxxx 10 xx LED 2 Blinkzeit 2 0b xxxx 11 xx LED 2 EIN 0b xx 00 xxxx LED 3 AUS 0b xx 01 xxxx LED 3 Blinkzeit 1 0b xx 10 xxxx LED 3 Blinkzeit 2 0b xx 11 xxxx LED 3 EIN usw. (s. auch Seite 22, CR1500 InOutStruct)
	0x02	PDO mapping for the 2nd application object <i>Sollwert LED-Kette 1</i>	u32, rw	0x6410 01	im Idx 6410 01 steht Sollwert der LED-Kette 1 (oben) Wertebereich von 0...10 0 = alle LED-Segmente AUS 1 = 1. LED-Segment EIN 2 = 2. LED-Segment EIN bzw. 1. und 2. LED-Segment EIN entspr. der gewählten Betriebsart 3 = 3. LED-Segment EIN bzw. 1. bis 3. LED-Segment EIN entspr. der gewählten Betriebsart usw. 10 = 10. LED-Segment EIN bzw. alle LED-Segmente EIN entspr. der gewählten Betriebsart Werte > 10 werden intern auf 10 „abgerundet“.
	0x03	PDO mapping for the 3rd application object <i>Sollwert LED-Kette 2</i>	u32, rw	0x6410 02	im Idx 6410 02 steht Sollwert der LED-Kette 2 (unten) Wertebereich von 0...10 0 = alle LED-Segmente AUS 1 = 1. LED-Segment EIN 2 = 2. LED-Segment EIN bzw. 1. und 2. LED-Segment EIN entspr. der gewählten Betriebsart 3 = 3. LED-Segment EIN bzw. 1. bis 3. LED-Segment EIN entspr. der gewählten Betriebsart usw. 10 = 10. LED-Segment EIN bzw. alle LED-Segmente EIN entspr. der gewählten Betriebsart Werte > 10 werden intern auf 10 „abgerundet“.

Objektverzeichnis

Kommunikationsprofile; Index 1000 bis 1FFF

Index	S-Idx	Name	Typ	Default	Beschreibung
1800	0x00	Transmit PDO <i>Funktions- und Pfeiltasten- Status</i>	u8, ro	0x05	Anzahl der Einträge Trans PDO; binär Eingänge (Tasten-Status)
	0x01	COB-ID Trans PDO	u32, rw	0x180 + Node ID	- PDO ist gültig (Bit 31 = 0) - CAN-ID des Trans PDOs
	0x02	Trans Type Trans PDO <i>Funktions- und Pfeiltasten- Status</i>	u8, rw	0xFF	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Ausgänge werden erst nach „n“ Synch Objekten aktualisiert n = 0x01 (1)...0xF0 (240) 0xFC nicht implementiert 0xFD nicht implementiert 0xFE = asynch man. spec. event; PDO wird bei Änderung der Eingänge sofort übertragen 0xFF = asynch device profile event; PDO wird bei Änderung der Eingänge sofort übertragen
	0x03	inhibit time	u16, rw	–	nicht implementiert
	0x04	(reserved)	u8, rw	–	nicht implementiert
	0x05	event timer	u16, rw	0x00	max. Übertragungspause des PDOs, wenn Trans Type = acyclic Einstellung in ms Bereich = 0...65535 ms (Default = 0 ms = Off)
1A00	0x00	Mapping Transmit PDO	u32, rw	0x01	Anzahl der eingebundenen Applikations-Objekte zur Abfrage des Tasten-Status
	0x01	PDO mapping for the 1st application object <i>Funktions- und Pfeiltasten- Status</i>	u32, rw	0x6000 01	in Idx 6000 01 stehen 4 Byte Tasten- Status z.B. Tasterbetrieb (Idx 2000 xx = 0x00) Byte 0: 0b xxxx xx 01 Taste 1 gedrückt 0b xxxx xx 11 Taste 1 gedrückt und Filterzeit abgelaufen 0b xxxx 01 xx Taste 2 gedrückt 0b xxxx 11 xx Taste 2 gedrückt und Filterzeit abgelaufen 0b xx 01 xxxx Taste 3 gedrückt 0b xx 11 xxxx Taste 3 gedrückt und Filterzeit abgelaufen 0b 01 xx xxxx Taste 4 gedrückt 0b 11 xx xxxx Taste 4 gedrückt und Filterzeit abgelaufen usw. (s. auch Seite 22, CR1500 InOutStruct)

Programmierung

Allgemeines

Das Tastaturmodul muss als CANopen-Slave mit den CANopen-Startfunktionen „COP_MSTR_BOOTUP“ und „COP_MSTR_MAIN“ vom R 360-Master initialisiert und in den Zustand „OPERATIONAL“ versetzt werden (Status-LEDs blinken 5 x).

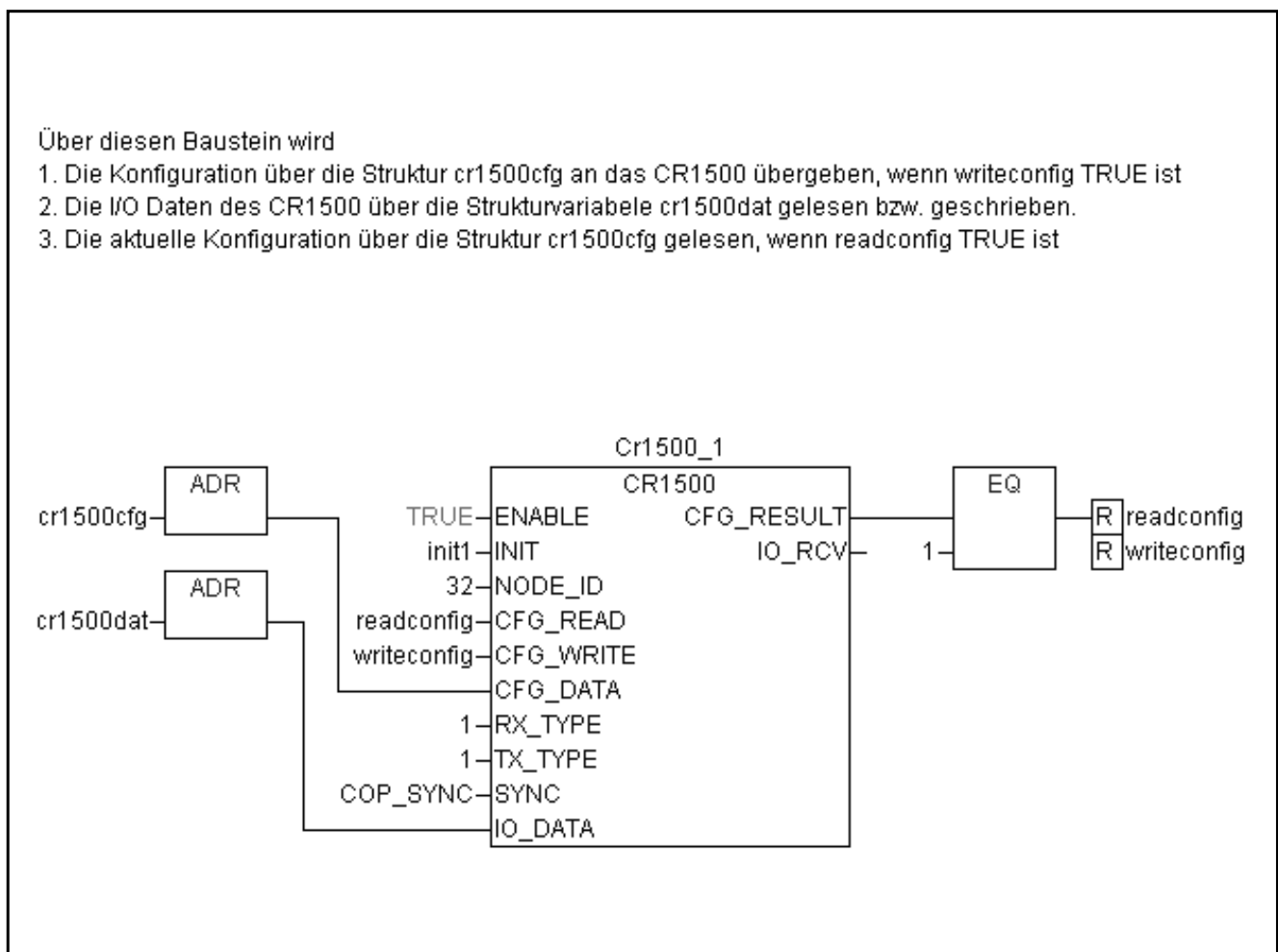
Programmier-Funktion

Wird die Funktion „CR1500“ in das Programm eingebunden, sorgt diese automatisch für eine ständige Aktualisierung der Tasten- und LED-Stati in der Steuerung. Die Funktion „CR1500“ befindet sich in der Bibliothek „CR1500_C.lib“ der Programmiersoftware CoDeSys.

Werden keine Konfigurationsdaten an das Tastaturmodul übertragen, arbeitet das Gerät mit den werksseitigen Default-Einstellungen.

Vor der Inbetriebnahme ist gegebenenfalls die werksseitige Node-ID des Tastaturmoduls zu ändern und die Baudrate von Master und Modul auf Gleichheit zu prüfen bzw. einzustellen.

Defaultwerte: Node-ID = 0x20 (= 32)
Baudrate = 0x04 (= 125 kBit/s)



Screenshot Funktionsbaustein

Datenstrukturen

Die Übergabe von CR1500-Konfigurations- und E/A-Daten erfolgt über Datenstrukturen. Im Deklarationsteil muss die Struktur – ebenso wie andere Variablentypen – deklariert werden. Für Konfigurationsdaten kann im Deklarationsteil bereits eine Wertzuweisung enthalten sein.

Im Programmablauf kann der Zugriff auf eine Strukturkomponente z.B. wie dargestellt erfolgen.

```

EcoPlus - CR1500.pro - [CANOPEN (FB-FUP)]
Datei Bearbeiten Projekt Einfügen Extras Online Fenster Hilfe
[Icons]
0069
0070 (*Die Konfiguration des CR1500 kann direct im Deklarationsteil zugewiesen werden*)
0071 cr1500cfg: CR1500ConfigStruct :=(GUARDTIME := T#500ms,
0072     LIFETIME :=3,
0073     Key1 := TRUE,
0074     Key2 := TRUE,
0075     Key3 := TRUE,
0076     Key4 := TRUE,
0077     Key5 := TRUE,
0078     Key6 := TRUE,
0079     Key7 := TRUE,
0080     Key8 := TRUE,
0081     Key9 := TRUE,
0082     Key10 := TRUE,
0083     Key11 := TRUE,
0084     Key12 := TRUE,
0085     KeyDelay := 500,
0086     LedBarMode1 := 10,
0087     LedBarMode2 := 10,
0088     LedBarColorChg1 :=3,
0089     LedBarColorChg2 := 3,
0090     StatusLedFlashTime1 := 500,
0091     StatusLedFlashTime2 := 200);
0092
0093
  
```

Screenshot Programmieroberfläche

Weitere CoDeSys Programmierbeispiele für das Tastaturmodul erhalten Sie auf Nachfrage von der ifm electronic gmbh.

- **Funktion:** **CR1500**
- **Library:** CR1500_C.lib
- **Zweck:** Parametriert und liest die Konfigurations- und E/A-Daten des Tastaturmoduls CR1500

CR1500	
ENABLE	CFG_RESULT
INIT	IO_RCV
NODE_ID	
CFG_READ	
CFG_WRITE	
CFG_DATA	
RX_TYPE	
TX_TYPE	
SYNC	
IO_DATA	

■ **Parameter**

Name	Datentyp	Beschreibung
Eingänge		
ENABLE	BOOL	TRUE: Funktion wird abgearbeitet
INIT	BOOL	TRUE: Funktionsinialisierung FALSE: zyklischer Funktionsaufruf
NODE_ID	BYTE	Knotenpunkt-Identifizier
CFG_READ	BOOL	TRUE: aktuelle Konfiguration des Moduls lesen
CFG_WRITE	BOOL	TRUE: aktuelle Konfiguration des Moduls schreiben
CFG_DATA	DWORD	Adresse der Konfigurationsdaten (Datenstruktur)
RX_TYPE	BYTE	Receive Transmission Type (Default = 1; synch cyclic) (von Steuerung aus betrachtet)
TX_TYPE	BYTE	Transmit Transmission Type (Default = 1; synch cyclic) (von Steuerung aus betrachtet)
SYNC	BOOL	CANopen-Synchronisationstakt (Systemvariable COB_SYNC)
IO_DATA	DWORD	Adresse der Ein-/Ausgangsdaten (Datenstruktur)
Ausgänge		
CFG_RESULT	BYTE	1 = Konfiguration wurde erfolgreich gelesen oder geschrieben 2 = Konfiguration wurde noch nicht gelesen oder geschrieben 3 = Konfiguration kann nicht gelesen oder geschrieben werden (fehlende bzw. falsche Node-ID oder Gerät defekt)
IO_RCV	BOOL	TRUE: für einen Zyklus, wenn neue Daten gesendet wurden.

Wenn nicht anders beschrieben, ist ein „FALSE“-Signal bei boolschen Datentypen stets die Negierung des beschriebenen „TRUE“-Signals.

■ Datenstruktur: CR1500 ConfigStruct

- Zweck:
Parameter- und Konfigurationsdaten können geschrieben oder gelesen werden. Die Datenstruktur wird dem Funktionseingang „CFG_DATA“ über den ADR-Operator zugewiesen.

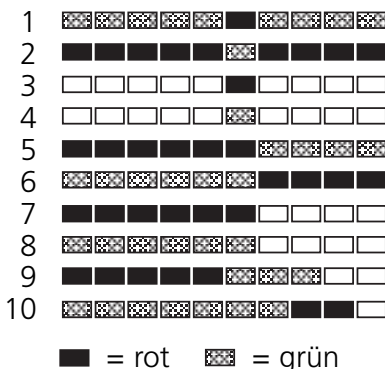
```

TYPE CR1500 ConfigStruct
STRUCT
  GUARDTIME: TIME;
  LIFETIME: BYTE;
  Key1: BOOL;
  Key2: BOOL;
  Key3: BOOL;
  Key4: BOOL;
  Key5: BOOL;
  ...
  Key12: BOOL;
  KeyDelay: WORD;
  LedBarMode1: BYTE;
  LedBarMode2: BYTE;
  LedBarColorChg1: BYTE;
  LedBarColorChg2: BYTE;
  StatusLedFlashTime1: WORD;
  StatusLedFlashTime2: WORD;
END_STRUCT
END_TYPE

```

■ Strukturkomponenten (ConfigStruct)

Name	Datentyp	Beschreibung
GUARDTIME	TIME	Guardingzeit des Tastaturmoduls [ms]
LIFETIME	BYTE	Lifetime des Tastaturmoduls
Key1...12	BOOL	Tastenbetriebsart FALSE = Tasterbetrieb (Default) TRUE = Toggle-Betrieb
KeyDelay	WORD	Verzögerungszeit (Filter) für die Tastenauswertung Wertebereich 0...65535 ms (gültig für alle Tasten) (Default 0 ms = Off)
LedBarMode1 LedBarMode2	BYTE BYTE	Betriebsart LED-Kette 1/2 (Wertebereich 0...10) 0 = LED-Kette nicht aktiv 1 = rotes Einzelsegment auf grünem Leuchtband (Default) 2 = grünes Einzelsegment auf rotem Leuchtband 3 = rotes Einzelsegment 4 = grünes Einzelsegment 5 = rote Segmentkette auf grünem Leuchtband 6 = grüne Segmentkette auf rotem Leuchtband 7 = rote Segmentkette 8 = grüne Segmentkette 9 = rote Segmentkette mit definiertem Umschlagpunkt für Farbwechsel (hier z.B. 5) 10 = grüne Segmentkette mit definiertem Umschlagpunkt für Farbwechsel (hier z.B. 7) (siehe auch Seite 9, LED-Ketten)
LedBarColorChg1 LedBarColorChg2	BYTE BYTE	Farbumschlagpunkt LED-Kette 1/2 Wertebereich 1...10 = Farbumschlag erfolgt nach Segment n Nur möglich bei Betriebsart 9 und 10
StatusLedFlashTime1 StatusLedFlashTime2	WORD WORD	Blinkzeit 1 für Status-LEDs (Default 500 ms = 2 Hz) Blinkzeit 2 für Status-LEDs (Default 200 ms = 5 Hz) Wertebereich 0...65535 ms



■ **Datenstruktur:
CR1500 InOutStruct**

■ Zweck:
Aktuelle E/A-Daten werden gelesen bzw. geschrieben. Die Datenstruktur wird dem Funktionseingang „IO_DATA“ über den ADR-Operator zugewiesen.

```

TYPE CR1500 InOutStruct
STRUCT
  Key1_Direct: BOOL;      (BinIn)
  Key1_Delay:  BOOL;      (BinIn)
  ...
  Key12_Delay: BOOL;      (BinIn)
  KeyLeft1_Direct: BOOL;  (BinIn)
  KeyLeft1_Delay: BOOL;   (BinIn)
  KeyRight1_Direct: BOOL; (BinIn)
  KeyRight1_Delay: BOOL;  (BinIn)
  ...
  KeyRight2_Delay: BOOL;  (BinIn)
  Led1_1:  BOOL;          (BinOut, LowBit)
  Led1_2:  BOOL;          (BinOut, HighBit)
  ...
  Led12_2: BOOL;          (BinOut, HighBit)
  LedBar1_Value: BYTE;    (AnalogOut)
  LedBar2_Value: BYTE;    (AnalogOut)
END_STRUCT
END_TYPE
    
```

■ Strukturkomponenten (Funktionstasten)

Name	Datentyp	Taste	Bit-Status im ...		Byte	Bit
			Taster-Betrieb	Toggle-Betrieb		
Key1_Direct Key1_Delay	BOOL BOOL	1	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet	0	0 1
Key2_Direct Key2_Delay	BOOL BOOL	2	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		2 3
Key3_Direct Key3_Delay	BOOL BOOL	3	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		4 5
Key4_Direct Key4_Delay	BOOL BOOL	4	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		6 7
Key5_Direct Key5_Delay	BOOL BOOL	5	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet	1	0 1
Key6_Direct Key6_Delay	BOOL BOOL	6	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		2 3
Key7_Direct Key7_Delay	BOOL BOOL	7	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		4 5
Key8_Direct Key8_Delay	BOOL BOOL	8	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		6 7
Key9_Direct Key9_Delay	BOOL BOOL	9	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet	2	0 1
Key10_Direct Key10_Delay	BOOL BOOL	10	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		2 3
Key11_Direct Key11_Delay	BOOL BOOL	11	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		4 5
Key12_Direct Key12_Delay	BOOL BOOL	12	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	TRUE, wenn geschaltet TRUE, wenn geschaltet		6 7

■ Strukturkomponenten (Pfeiltasten)

Name	Datentyp	Taste	Bit-Status im ...		Byte	Bit
			Taster-Betrieb	Toggle-Betrieb		
KeyLeft1_Direct KeyLeft1_Delay	BOOL BOOL	Taste o. links	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	(nicht unterstützt)	3	0 1
KeyRight1_Direct KeyRight1_Delay	BOOL BOOL	Taste o. rechts	TRUE wenn gedrückt TRUE, nach Ablauf d. Zeit	(nicht unterstützt)		2 3
KeyLeft2_Direct KeyLeft2_Delay	BOOL BOOL	Tasten u. links	TRUE, wenn gedrückt TRUE, nach Ablauf d. Zeit	(nicht unterstützt)		4 5
KeyRight2_Direct KeyRight2_Delay	BOOL BOOL	Taste u. rechts	TRUE wenn gedrückt TRUE, nach Ablauf d. Zeit	(nicht unterstützt)		6 7

■ Strukturkomponenten (Status-LEDs und LED-Ketten)

Name	Datentyp	LED	Betriebszustand ...				Byte	Bit
			AUS	Blink 1	Blink 2	EIN		
LED1_1 (Low) LED1_2 (High)	BOOL BOOL	1	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	0 1	
LED2_1 (Low) LED2_2 (High)	BOOL BOOL	2	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	2 3	
LED3_1 (Low) LED3_2 (High)	BOOL BOOL	3	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	4 5	
LED4_1 (Low) LED4_2 (High)	BOOL BOOL	4	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	6 7	
LED5_1 (Low) LED5_2 (High)	BOOL BOOL	5	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	1 1	
LED6_1 (Low) LED6_2 (High)	BOOL BOOL	6	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	2 3	
LED7_1 (Low) LED7_2 (High)	BOOL BOOL	7	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	4 5	
LED8_1 (Low) LED8_2 (High)	BOOL BOOL	8	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	6 7	
LED9_1 (Low) LED9_2 (High)	BOOL BOOL	9	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	2 1	
LED10_1 (Low) LED10_2 (High)	BOOL BOOL	10	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	2 3	
LED11_1 (Low) LED11_2 (High)	BOOL BOOL	11	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	4 5	
LED12_1 (Low) LED12_2 (High)	BOOL BOOL	12	FALSE FALSE	TRUE FALSE	FALSE TRUE	TRUE TRUE	6 7	
LEDBar1_Value	BYTE	LED- Kette 1	Analogwert für LED-Kette 1 (oben) (0...10)				3	
LEDBar2_Value	BYTE	LED- Kette 2	Analogwert für LED-Kette 2 (unten) (0...10)				4	

Begriffe und Abkürzungen

0b ...	binärer Zahlenwert (zur Bitcodierung), z.B. 0b0001 0000
0x ...	hexadezimaler Zahlenwert, z.B. 0x64 (= 100 dezimal)
Baudrate	Übertragungsgeschwindigkeit (1 Baud = 1 Bit/sec.)
CAL	CAN Application Layer
CAN	CAN basierendes Netzwerkprotokoll auf Applikationsebene
CAN_H	Controller Area Network (Bussystem für den Einsatz im Mobilbereich)
CAN_L	CAN-High; CAN-Anschluß/-Leitung mit dem hohen Spannungspegel
CANopen	CAN-Low; CAN-Anschluß/-Leitung mit dem niederen Spannungspegel
CANopen	CAN basierendes Netzwerkprotokoll auf Applikationsebene mit einer offenen Konfigurationsschnittstelle (Objektverzeichnis).
CiA	"CAN in Automation e.V." (Anwender- und Herstellerorganisation in Deutschland/Erlangen) Definitions- und Kontrollorgan für CAN und CAN-basierende Netzwerkprotokolle
CiA DS	Draft Standard (veröffentlichte CiA-Spezifikation, die in der Regel ein Jahr nicht geändert und erweitert wurde)
CiA DSP	Draft Standard Proposal (veröffentlichter CiA-Spezifikationsentwurf)
CiA WD	Work Draft (CiA-intern zur Diskussion akzeptiertes Arbeitspapier)
CiA DS 301	Spezifikation zum CANopen Kommunikationsprofil; beschreibt die grundlegenden Kommunikationsmechanismen zwischen den Netzwerkteilnehmern, wie z.B die Übertragung von Prozessdaten in Echtzeit, den Datenaustausch zwischen Geräten oder die Konfigurationsphase. Entspr. der Applikation ergänzt mit den nachfolgenden CiA-Spezifikationen:
CiA DS 401	Geräteprofil für digitale und analoge E/A-Baugruppen
CiA DS 402	Geräteprofil für Antriebe
CiA DS 403	Geräteprofil für Bediengeräte
CiA DS 404	Geräteprofil für Messtechnik und Regler
CiA DS 405	Spezifikation zur Schnittstelle zu programmierbaren Systemen (IEC 61131-3)
CiA DS 406	Geräteprofil für Drehgeber/Encoder
CiA DS 407	Applikationsprofil für den öffentlichen Nahverkehr
COB	CANopen Communication Object (PDO, SDO, EMCY, ...)
COB-ID	CANopen Identifier eines Communication Objects
Communication cycle	Die zu überwachende Synchronisationszeit; max. Zeit zwischen 2 Sync-Objekten
EMCY Object	Emergency Object (Alarmbotschaft; Gerät signalisiert einen Fehler)
Error Reg	Error Register (Eintrag mit einer Fehlerkennung)
Guarding Error	Knoten bzw. Netzwerkteilnehmer wurde bzw. wird nicht mehr gefunden Guard-MASTER: Einer oder mehrere SLAVES melden sich nicht mehr. Guard-SLAVE: Das Gerät (SLAVE) wird nicht mehr abgefragt.
Guard Time	Innerhalb dieser Zeit erwartet der Netzwerkteilnehmer ein "Node Guarding" des Netz-Masters
Heartbeat	Parametrierbare zyklische Überwachung von Netzwerkteilnehmern untereinander. Im Gegensatz zum „Node Guarding“ wird kein übergeordneter NMT-Master benötigt.
ID	Identifier; kennzeichnet eine CAN-Nachricht. Der numerische Wert des ID beinhaltet gleichzeitig eine Priorität bezüglich des Bus-Zugriffes. ID 0 = höchste Priorität.
Identifier	siehe ID
Idx	Index; bildet zusammen mit dem S-Index die Adresse eines Eintrages im Objektverzeichnis
Life Time Factor Monitoring	Anzahl der Versuche bei fehlender Guarding Antwort Wird verwendet um die Fehlerklasse (Guarding-Überwachung, Synch-, etc.) zu beschreiben.
NMT	Netzwerk-Management
NMT-Master/-Slaves	Der NMT-Master steuert die Betriebszustände der NMT-Slaves

Node Guarding	Parametrierbare zyklische Überwachung von Slave-Netzwerkteilnehmern durch einen übergeordneten Master-Knoten, sowie die Überwachung dieses Abfragemechanismus durch die Slave-Teilnehmer.
Node-ID	Knotenpunkt-Identifizier (Kennung eines Teilnehmers im CANopen Netz)
Objekt (auch OBJ)	Oberbegriff für austauschbare Daten/Botschaften innerhalb des CANopen-Netzwerks
Objektverzeichnis	enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten. Auf die einzelnen Einträge wird über den Index und S-Index zugegriffen.
Operational	Betriebszustand eines CANopen Teilnehmers. In diesem Modus können SDOs, NMT-Kommandos und PDOs übertragen werden.
PDO	Process Data Object; im CANopen Netz zur Übertragung von Prozessdaten in Echtzeit, wie z.B. Drehzahl eines Motors. PDOs besitzen eine höhere Priorität als SDOs; im Gegensatz zu SDOs werden sie unbestätigt übertragen. PDOs bestehen aus einer CAN-Nachricht mit Identifizier und bis zu 8 Byte Nutzdaten.
PDO Mapping	Beschreibt die Applikationsdaten, die mit einem PDO übertragen werden.
Pre-Op	Preoperational; Betriebszustand eines CANopen Teilnehmers. Nach dem Einschalten der Versorgungsspannung geht jeder Teilnehmer automatisch in diesen Zustand. Im CANopen-Netz können in diesem Modus nur SDOs und NMT-Kommandos übertragen werden, jedoch keine Prozessdaten
Prepared	(auch stopped) Betriebszustand eines CANopen Teilnehmers. In diesem Modus werden nur NMT- Kommandos übertragen.
Rec PDO (auch Rx PDO)	(Receive) Empfangs Process Data Object
ro	read only (unidirektional; nur Lesen)
rw	read-write (bidirektional; Lesen-Schreiben)
Rx-Queue	Empfangspuffer
s16	Datentyp signed 16 bit (mit Vorzeichen, 16 Bit-Format)
SDO	Service Data Object; Mit diesem Objekt wird gezielt auf das Objektverzeichnis eines Netzwerkteilnehmers zugegriffen (lesen/schreiben). Ein SDO kann aus mehreren CAN-Nachrichten bestehen. Die Übertragung der einzelnen Nachrichten wird von dem angesprochenen Teilnehmer bestätigt. Mit den SDOs lassen sich Geräte konfigurieren und parametrieren.
Server SDO	Mechanismus und Parametersatz um das "eigene" Objektverzeichnis eines Netzwerkteilnehmers anderen Teilnehmern (Clients) zugänglich zu machen.
S-Idx (auch SIdx)	Subindex innerhalb d. Objektverzeichnisses eines CANopen fähigen Gerätes
Start Guarding	Start der Knotenüberwachung
str	Datentyp String (Variable für Zeichenketten, wie z.B. Text "load")
Sync Error	Ausbleiben des Sync OBJ innerhalb der parametrierbaren Synchronisationszeit
Sync OBJ	Synchronisationsobjekt zur netzwerkweit gleichzeitigen Aktualisierung bzw. Übernahme der Prozessdaten der entsprechend parametrierten PDOs.
Sync Windows	Zeitfenster in dem die synchronen PDOs übertragenen werden müssen.
Time Stamp	Zeitstempel zum Abgleich evtl. vorhandener Uhren in Netzwerkteilnehmern
Trans Type	Art der Prozess-Datenübertragung; synchron, asynchron
Trans PDO (auch Tx PDO)	(Transmit) Sende Process Data Object
Trans SDO (auch Tx SDO)	(Transmit) Sende Service Data Object
Tx-Queue	(Transmit) Sendepuffer
u8 (16, 32)	Datentyp unsigned 8 (16, 32) bit (ohne Vorzeichen, 8 (16, 32) Bit-Format)
wo	write only (nur schreiben)

Safety instructions



These instructions are part of the unit. They contain text and figures about the correct handling of the module and must be read before installation or use of the unit.

Adhere to the information in the documentation. Non-observance of the instructions, operation which is not in accordance with use as prescribed below, wrong installation or handling can affect the safety of people and the plant.

The unit must be installed, connected and put into operation by a qualified electrician.

Disconnect the unit externally before handling it. Also disconnect any independently supplied output load circuits.

In case of malfunction of the unit or uncertainties please contact the manufacturer. Tampering with the unit can seriously affect the safety of people and equipment. This is not permitted and leads to an exclusion of liability and warranty.

Contents

Function and features	page 27
Technical data	page 28
Dimensions	page 29
Key labelling	page 29
Mounting and electrical connection	page 30
Key and LED operating modes	page 32
List of parameters and EMCY objects	page 34
Maintenance, repair and disposal	page 35
Declaration of conformity	page 35
Object directories	
Manufacturer-specific profiles, index 2000 to 5FFF	page 36
Communication profiles; index 1000 to 1FFF	page 38
Programming	page 42
Terms and abbreviations	page 48
Notes	page 50

Function and features

The keypad and display module CR1500 for CANopen systems can be mounted on or in driver's cabs, control panels or control cabinets. The evaluation and parameter settings of the keys and LEDs are made according to IEC 61131-3 within the controller software.

The function library for the CoDeSys programming system supports the integration of the keypad functions in the application program.

CANopen communication at once glance

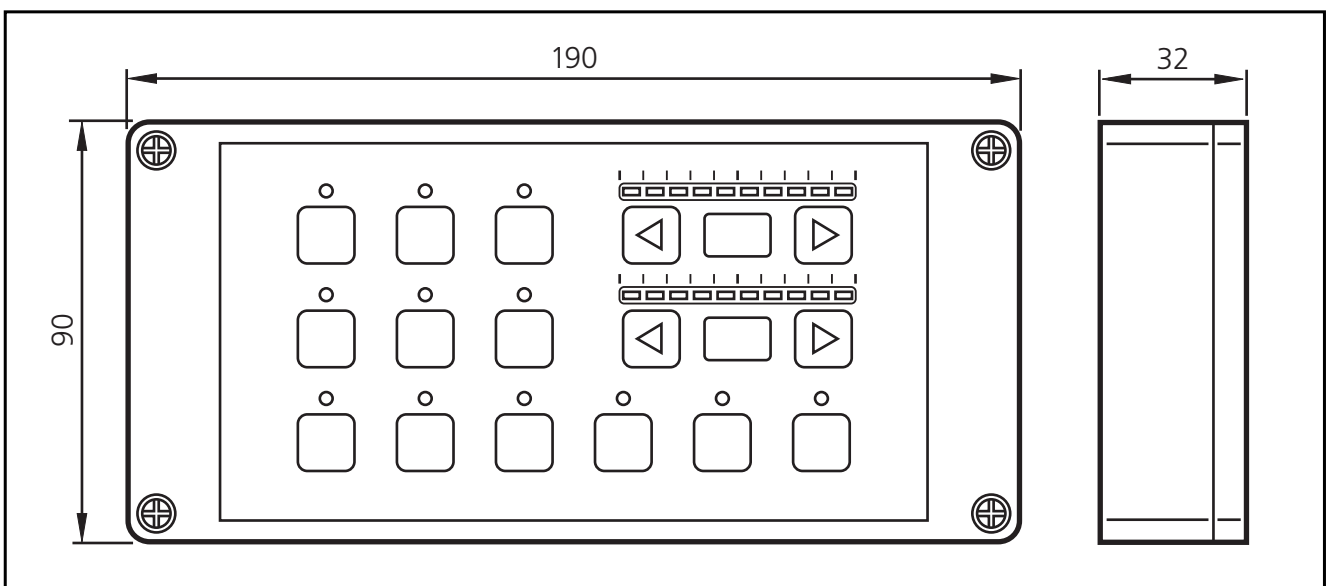
- There are 1 server SDO and 2 default PDOs according to CiA DS 401.
The default identifiers are assigned according to the "predefined connection set".
- The COB IDs and the transmission mode (synch/asynch/cyclic) for the Trans PDO (key status) and the Rec PDO (LED status) can be configured.
The transmission mode is stored non volatily. Modified PDOs (PDO linking) will be stored volatily.
- The module expects a synch object.
The CAN identifier of the synch object can be configured. After a modification the ID is automatically stored non volatily.
- The module supports "node guarding".
The "guard time", "life time factor" and the CAN identifier of the guard object can be configured and are stored non volatily.
- The module supports "Heartbeat".
"Producer heartbeat time" and "Consumer heartbeat time" can be configured. Max. 4 network participants can be monitored.
- The module generates an emergency object. The COB ID of the EMCY object can be configured.
- The module stores the last 4 errors.
The error code of the corresponding emergency object is stored.
- The module supports a reset function,
i.e. the assignment of the parameters to the factory default values * on request.

*) factory default values see "List of parameters", page 34

Technical data

Operating voltage	10...30 V DC; SELV
Current consumption	200 mA (24 V DC) (The current consumption in the case of an error must be limited to 1 A by appropriate measures)
Function keys Operating keys	12, to be evaluated via application program 4 arrow keys, to be evaluated via application program
Key labelling	3 exchangeable strips to be inserted
Status LEDs Rows of LEDs	12, freely configurable via application program 2 x 10-digit, red/green, operating mode selectable
Interface	CAN Interface (ISO 11898)
Baud rate	20 Kbits/s...1 Mbits/s (default setting 125 Kbits/s)
Communication profile Device profile	CANopen, CiA DS301 V4.0 manufacturer-specific
CAN	Full-CAN 2.0
Node ID (default)	hex 20 (= 32)
Operating temperature Storage temperature	-20...+85°C -30...+85°C
Protection	IP64 / III
Housing material	Noryl
Dimensions (WxHxD) Overall height for wall mounting height for control cabinet mounting depth for control cabinet mounting Maximum depth of the front panel	190 x 90 x 32 mm 32 mm 7 mm 25 mm 2.5 mm
Connection	5 WAGO cage-clamp terminals
Wiring	1: CAN_H, 2: CAN_L, 3: GND, 4: GND, 5: +24 V

Dimensions

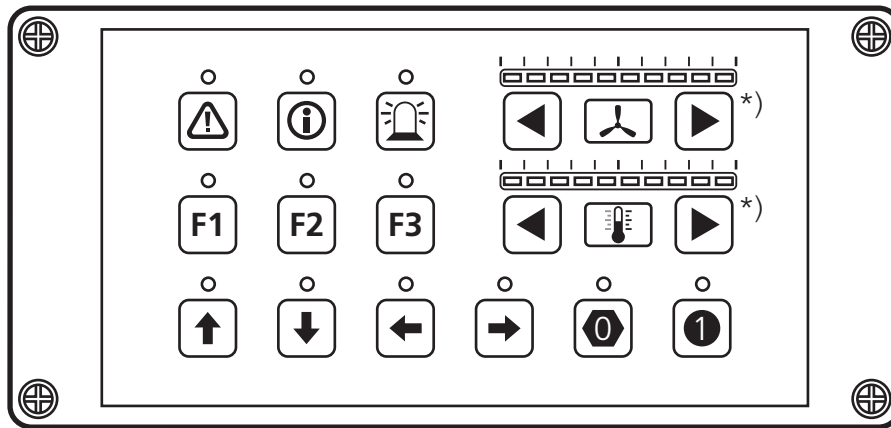


Key labelling

The application-specific labelling of the function keys and rows of LEDs is made by inserted strips. These strips can be labelled by means of any text or graphic software on a PC and can then be printed out.

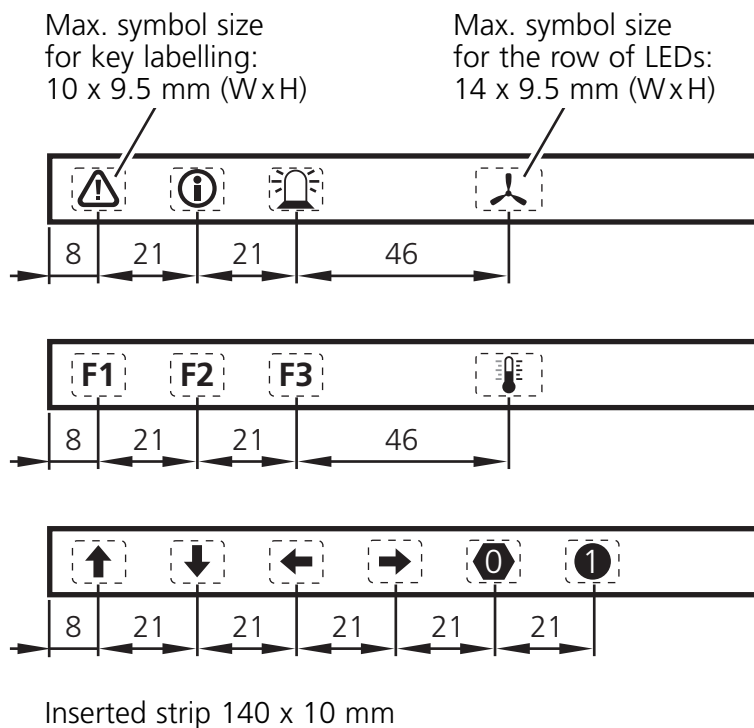
To simplify the insertion of the strips thicker paper should be used for the print-out. Photopaper on plastic basis is particularly suitable for this purpose.

Labelling example



*) The 4 operating keys (arrow keys) belonging to the rows of LEDs are printed at the factory.

Dimensions of the labelling strips



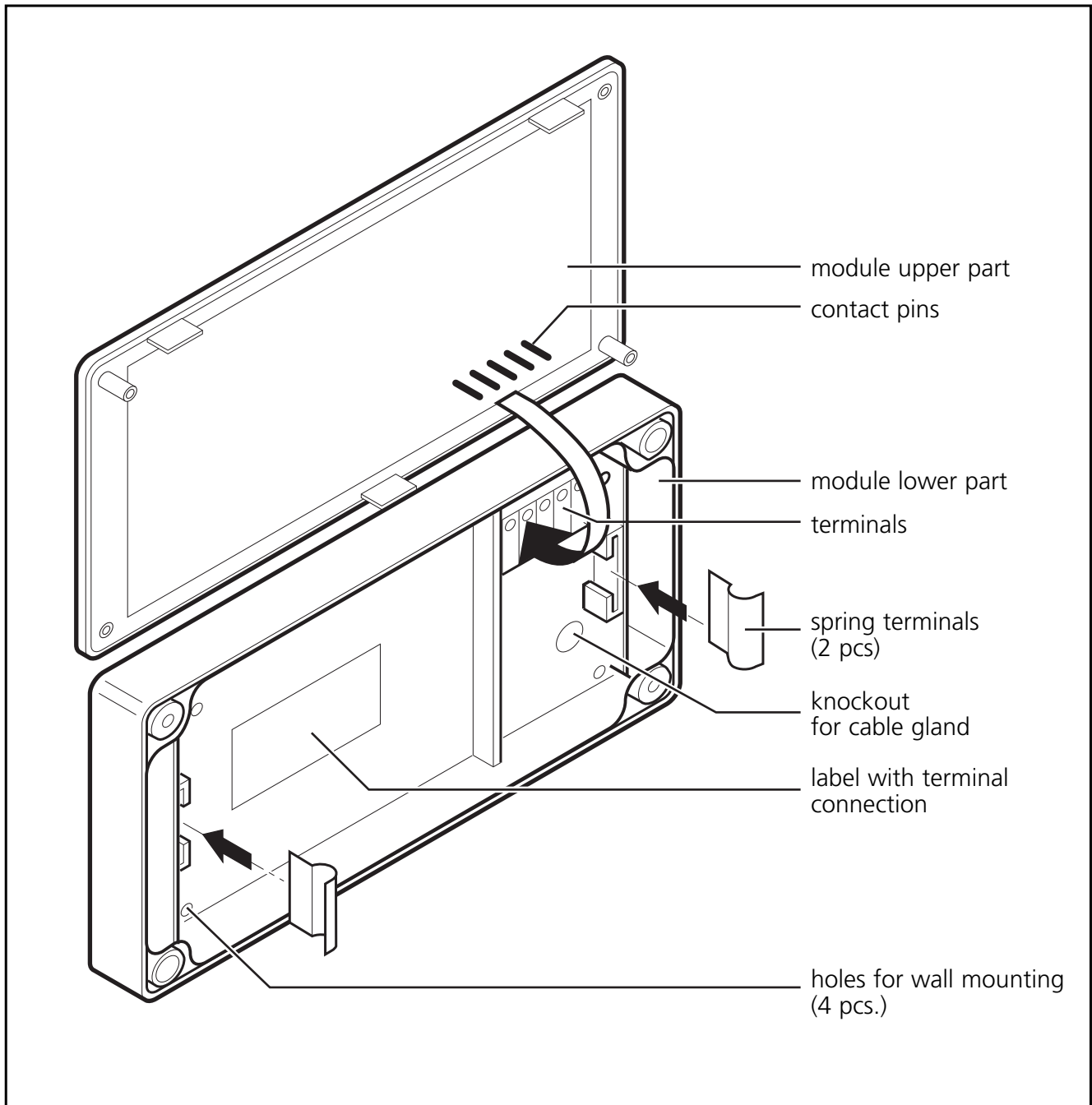
Mounting and electrical connection



The module must be installed, connected and put into operation by a qualified electrician.



The keypad module contains C-MOS elements which can be damaged by electrostatic discharge. Touch a grounded part before opening the module.



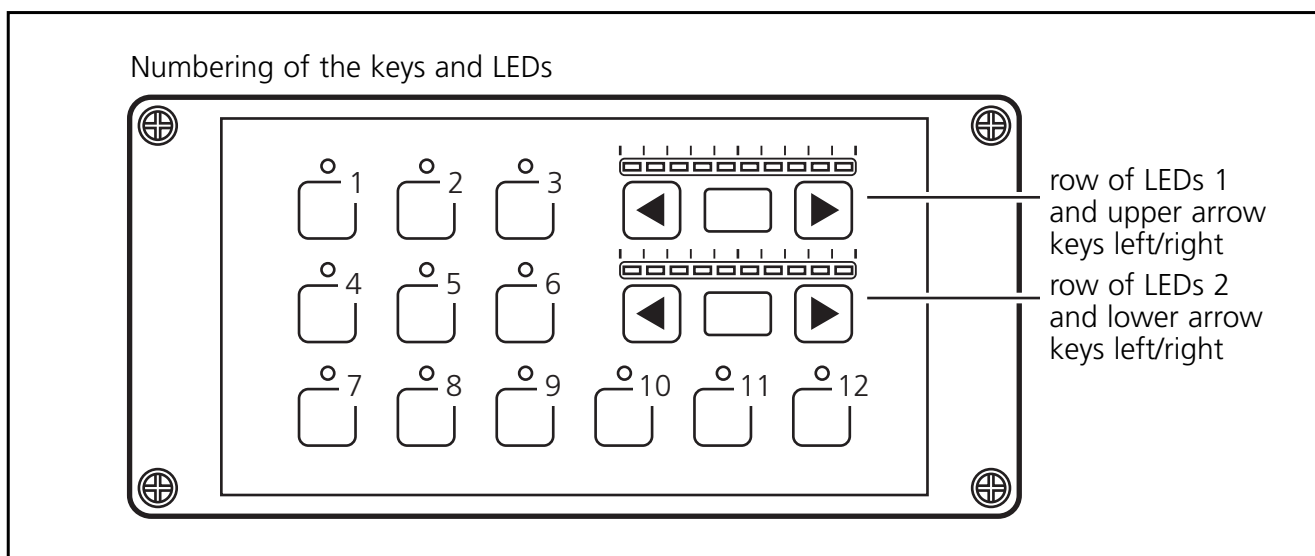
■ Mounting in a control cabinet or operating panel

1. Prepare a cutout with 4 holes in the front panel of the control cabinet or the operating panel by means of the enclosed mounting template (#5310674).
2. Loosen the 4 screws of the keypad module and remove the module upper part from the module lower part.
3. Remove the knockout for the cable in the bottom of the module lower part.
4. Insert the supplied cable gland.
5. Insert the module lower part in the cutout from the inside of the control cabinet.
6. Push the supplied spring terminals into the brackets to the right and to the left of the module lower part from the outside of the control cabinet and fasten the module lower part in the cutout.
7. Push the cable through the cable gland.
8. Connect the wires according to the label in the module lower part.
9. Put the module upper part onto the lower part in a way that the contact pins of the upper part fit into the terminals in the lower part.
10. Screw the module upper part onto the lower part.

■ Mounting on a wall

1. Loosen the 4 screws of the keypad module and remove the module upper part from the module lower part.
2. Remove the knockout for the cable in the bottom of the module lower part or
drill a cable entry into the lower side wall within the terminal chamber and cut a suitable cable gland thread (see supplied mounting template).
3. Insert the supplied cable gland if necessary.
4. Push the cable through the cable gland.
5. Fix the module lower part onto the wall with four screws or similar.
6. Connect the wires according to the label in the module lower part.
7. Put the module upper part onto the lower part in a way that the contact pins of the upper part fit into the terminals in the lower part.
8. Screw the module upper part onto the lower part.

Key and LED operating modes



■ Keys

Each of the function and arrow keys enables the following operating modes:

- push button operation (key bit is TRUE as long as the key is pressed)
- push button operation with delay (key bit becomes TRUE only if the key has been pressed for x ms).

The 12 function keys also support the operating mode "Toggle":

- Toggle operation (key bit is TRUE until the key is pressed once again).
- Toggle operation with delay (key bit changes state if the key has been pressed for x ms).

To support the evaluation of the key delay (filter) in the push button mode the pressing of a key is evaluated by 2 bits per key.

The first bit is transferred immediately as soon as the key has been pressed (corresponds to the CANopen operating mode). The second bit only becomes TRUE when the set delay time has elapsed. Then the key status is transferred once again (this time both bits are TRUE).

In the toggle operation, with or without active filter, both bits are always transferred. The key status is displayed in the first 4 bytes of the Transmit PDO.

e.g. byte 0:	0b xxxx xx 01	function key 1 pressed
	0b xxxx xx 11	function key 1 pressed and delay time elapsed
	0b xxxx 01 xx	function key 2 pressed
	0b xxxx 11 xx	function key 2 pressed and delay time elapsed
	etc.	

The parameters for the operating mode of the buttons and the delay time are set via the object directory (as of Idx 2000). The IEC function "CR1500" of the CoDeSys programming software passes the key status on to the application.

Please note: The set delay time applies to all keys.

A maximum of 1 key can be pressed at the same time.

■ Status LEDs

Each status LED can be triggered individually. The parameters of the LED display mode are not set via SDO but can be triggered directly by Receive PDO from the application. The first 3 bytes of the Rec PDO contain the LED status. 4 operating states are available:












- OFF
- flashing (timer 1, adjustable 0...65535 ms, default 500 ms = 2 Hz)
- flashing (timer 2, adjustable 0...65535 ms, default 200 ms = 5 Hz)
- ON

The status LEDs also indicate the following operating states:

operating states (change in state)	Status LEDs
OFF → PREOPERATIONAL	light for 1 second
PREOPERATIONAL → OPERATIONAL	5 x flashing, then state as programmed
Communication error	permanent fast flashing (5 Hz)

■ Rows of LEDs

The parameters of the operating mode of the rows of LEDs are set via the object directory (as of Idx 2002). The following operating modes are available:

Operating mode	Description
1 	red individual segment in a row of LEDs lighting green
2 	green individual segment in a row of LEDs lighting red
3 	red individual segment
4 	green individual segment
5 	red row of segments in a row of LEDs lighting green
6 	green row of segments in a row of LEDs lighting red
7 	red row of segments
8 	green row of segments
9 	red row of segments with colour change point (here 5)
10 	green row of segments with colour change p. (here 7)
	

The LED segments are triggered via an analogue value (0...10) in bytes 3 and 4 of the Rec PDO. The IEC function "NORM" allows to standardise actual and preset values for these rows of LEDs.

List of parameters and EMCY objects

With the function "Restore" (see object directory, Idx 1011) the factory default values can be restored (except for the parameters "Baud rate" and the "Node ID"). With the next power on they become valid.

List of parameters

Parameter	Index in object directory	Default value (factory preset)	Change saved automatically	Change effective
Manufacturer-specific profiles, index 2000 to 5FFF				
Key configuration	2000	0x00 (push button mode)	yes	at once
Delay time keys (filter)	2001	0x00 (0 ms = Off) (valid for all keys)	yes	at once
Operating mode rows of LEDs	2002	0x01 (red individual segment in a row of LEDs lighting green)	yes	at once
Colour change point	2003	0x01	yes	at once
Flashing time 1 status LED	2004 01	0x01F4 (500 ms = 2 Hz)	yes	at once
Flashing time 2	2004 02	0x00C8 (200 ms = 5 Hz)	yes	at once
Node-ID	20F0, 20F1	0x20 (= 32)	yes	after reset
Baud rate	20F2, 20F3	0x04 (= 125 Kbits/s)	yes	after reset
Communication profiles; index 1000 to 1FFF				
COB-ID SYNC	1005	0x80	yes	at once
Communication Cycle	1006	0x00 (Off)	yes	after Pre-Op
Guard Time	100C	0x00 (Off)	yes	at once
Life Time Factor	100D	0x00	yes	at once
COB-ID EMCY	1014	0x80 + Node ID	yes	at once
Heartbeat time 1	1016 01	0x00 (Off)	yes	at once
Heartbeat time 2	1016 02	0x00 (Off)	yes	at once
Heartbeat time 3	1016 03	0x00 (Off)	yes	at once
Heartbeat time 4	1016 04	0x00 (Off)	yes	at once
Producer heartbeat time	1017	0x00 (Off)	yes	at once
COB-ID Rec PDO	1400 01	0x200 + Node ID	no	at once
Trans Type Rec PDO	1400 02	0x01 (synchronous)	yes	at once
COB-ID Trans PDO	1800 01	0x180 + Node ID	no	at once
Trans Type Trans PDO	1800 02	0xFF (asynchronous)	yes	at once

The life time factor 0 is interpreted as 1.

The first guard protocol is interpreted as "Start Guarding" even if guarding is not yet active at that time (guard time = 0).

EMCY objects

The following error codes according to DSP-401 or DSP-301 are supported:

EMCY code	Error Reg	Additional code	Description
0x6100	0x11	0x00	"Internal Software" Overflow of an Rx queue; frequency of the Rx PDOs is too high. Only external reset, via an entry in Idx 1003 00.
0x6101	0x11	0x00	"Internal Software" Overflow of a Tx queue; e.g. device does not communicate with the bus. Only external reset via an entry in Idx 1003 00.
0x6200	0x81	0x00	"User Software" More than 1 key has been pressed simultaneously. Evaluation of the key status not possible. Reset when the key is released.
0x8100	0x11	0x00	"Monitoring" (Guarding Error) For "guard time" * "life time factor" no guard object is received. Life time factor 0 corresponds to 1. Reset for node again.
0x8200	0x11	0x00	"Monitoring" (Synch Error) For "communication cycle" no synch object is received; (only in OPERATIONAL) Reset with the next synch OBJ or PREOP.

Maintenance, repair and disposal

As the module does not contain any components which must be maintained, the housing must not be opened. The repair of the module may only be carried out by the manufacturer.

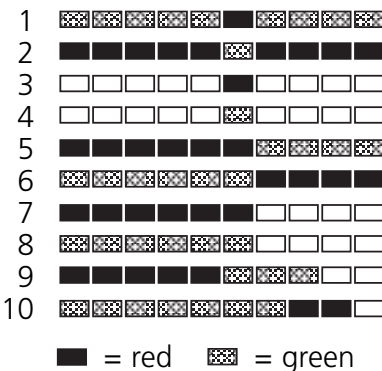
The disposal must be carried out according to the corresponding national environmental regulations.

Declaration of conformity

The CE marking is applied on the basis of the EMC Directive EMC 89/336/EEC as well as the Low Voltage Directive NS73/23/EEC.

Object directory

Manufacturer-specific profiles, index 2000 to 5FFF

Index	S-Idx	Name	Type	Default	Description
2000	0x00	Configuration of the push button operating mode	u8, ro	0x10	Number of entries (= number of keys = 16)
	0x01	key 1	u8, rw	0x00	0 = push button operating mode 1 = toggle operating mode
	⋮	⋮	⋮	⋮	⋮
	0x0A	key 10	u8, rw	0x00	0 = push button operating mode 1 = toggle operating mode
	0x0B	key 11	u8, rw	0x00	0 = push button operating mode 1 = toggle operating mode
	0x0C	key 12	u8, rw	0x00	0 = push button operating mode 1 = toggle operating mode
	0x0D	upper left arrow key	u8, ro	0x00	0 = push button operating mode toggle operating mode is not supported
	0x0E	upper right arrow key	u8, ro	0x00	0 = push button operating mode toggle operating mode is not supported
	0x0F	lower left arrow key	u8, ro	0x00	0 = push button operating mode toggle operating mode is not supported
	0x10	upper right arrow key	u8, ro	0x00	0 = push button operating mode toggle operating mode is not supported
2001	0x00	delay time for key evaluation	u16, rw	0x00	Setting in ms range = 0...65535 ms (valid for all keys)
2002	0x00	operating mode of the rows of LEDs	u8, ro	0x02	number of integrated rows of LEDs
	0x01	operating mode row of LEDs 1	u8, rw	0x01	<p>value range 0...10 (0x00...0A)</p> <p>0 = row of LEDs not active</p> <p>1 = red individual segment in a row of LEDs lighting green (default)</p> <p>2 = green individual segment in a row of LEDs lighting red</p> <p>3 = red individual segment</p> <p>4 = green individual segment</p> <p>5 = red row of segments in a row LEDs lighting green</p> <p>6 = green row of segments in a row of LEDs lighting red</p> <p>7 = red row of segments</p> <p>8 = green row of segments</p> <p>9 = red row of segments with defined colour change point</p> <p>10 = green row of segments with defined colour change point (also see page 33, rows of LEDs)</p>  <p>■ = red ▨ = green</p>
	0x02	operating mode row of LEDs 2	u8, rw	0x01	value range 0...10 (0x00...0A) see row of LEDs 1 (also see page 33, rows of LEDs)

Object directory

Manufacturer-specific profiles, index 2000 to 5FFF

Index	S-Idx	Name	Type	Default	Description
2003	0x00	colour change point	u8, ro	0x02	number of integrated rows of LEDs
	0x01	colour change point row of LEDs 1	u8, rw	0x01	colour change point row of LEDs 1 value range 1...10 (0x00...0A) (values < 1 and > 10 are evaluated internally as 1 or 10), only possible for operating modes 9 and 10! (also see page 33, row of LEDs)
	0x02	colour change point row of LEDs 2	u8, rw	0x01	colour change point row of LEDs 2 value range 1...10 (0x00...0A) (values < 1 and > 10 are evaluated internally as 1 or 10), only possible for operating modes 9 and 10! (also see page 33, row of LEDs)
2004	0x00	flashing time of the status LEDs	u8, ro	0x02	number of integrated timers
	0x01	flashing time 1	u16, rw	0x01F4	flashing time 1 (time period) in ms value range = 0...65535 ms (default 500 ms / 2 Hz)
	0x02	flashing time 2	u16, rw	0x00C8	flashing time 2 (time period) in ms value range = 0...65535 ms (default 200 ms / 5 Hz)
20F0	0x00	setting	u8, rw	0x20	Node-ID used to access the module in the CANopen system
20F1	0x00	Node-ID		(= 32)	
<p>A node ID change is only accepted if the entries 20F0 and 20F1 contain the same changed value. Values below 1 and above 127 are not accepted, the existing setting remains valid. After setting of the node ID a reset must be made (switch off the module for a short time) so that the new entries become valid.</p>					
20F2	0x00	setting	u8, rw	0x04	Baud rate of the CAN system 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud 4 = 125 kBaud (default) 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
20F3	0x00	Baud rate			
<p>A baud rate change is only accepted if the entries 20F2 and 20F3 contain the same changed value. Values above 7 are not accepted, the existing setting remains valid. After setting of the baud rate a reset must be made (switch off the module for a short time) so that the new entries become valid.</p>					

Explanation of the abbreviations:

0b... = binary number

0x... = hexadecimal number

rw = read-write

ro = read only

u8 = unsigned 8 bit

(also see "Terms and abbreviations", page 48)

Object directory

Communication profiles; index 1000 to 1FFF

Index	S-Idx	Name	Type	Default	Description
1000	0x00	device type	u32, ro	0x0000	manufacturer-specific
1001	0x00	error register	u8, ro	0x00	bit coded according to profile 301; the following is supported: 0b 0000 0000 no error 0b x00x 0001 generic error 0b x001 000x communication error 0b 100x 000x manufacturer-specific
1003	0x00	pre-defined errorfield	u8, ro	0x02	An error list with 4 entries is supported
	0x01-4	error history	u32, ro	0x00	error occurred, coded according to the EMCY list, the last error occurred is in the sub index 1
1005	0x00	COB-ID SYNC	u32, rw	0x80000080	- module expects synch message (bit 31 = 1) - module generates no synch message (bit 30 = 0) - 11-bit identifier system (bit 29 = 0) - identifier of the synch message
1006	0x00	communication cycle period	u32, rw	0x00000000	max. time between 2 synch objects in μ s; useful resolution = 1 ms
1008	0x00	device name	str, ro	CR1500	device designation
1009	0x00	HW Version	str, ro	x.x	hardware version
100A	0x00	SW Version	str, ro	x.x	software version
100B	0x00	Node-ID	u32, ro		read only
100C	0x00	guard time	u16, rw	0x0000	time in ms Within this time the module expects a "node guarding" of the system master. If the value 0 is entered here, this function is not supported.
100D	0x00	life time factor	u8, rw	0x00	If no "node guarding" is received for "guard time" * "life time", the module generates an EMCY. The product of "guard time" * "life time" must be between 0 and 65535.
1010	0x00	number of save-options	u8, ro	0x01	number of "save options"
	0x01	store parameters	u32, rw	0x02	All parameters are automatically stored after a change.
1011	0x00	number of restore-options	u8, ro	0x01	number of "restore options"
	0x01	restore default parameters	u32, rw	0x01	If the string "load" is entered here, the factory default parameters are restored and are valid after the next reset.

Object directory

Communication profiles; index 1000 to 1FFF

Index	S-Idx	Name	Type	Default	Description
1014	0x00	COB-ID EMCY	u32, rw	0x40000080 +Node ID	<ul style="list-style-type: none"> - The module does not react on foreign EMCY mess. (bit 31 = 0) - Module generates EMCY message (bit 30 = 1) - 11 Bit ID (bit 29 = 0) - ID = 0x80 + Node ID CAN identifier can be changed by the user.
1016	0x00	Consumer heartbeat time	u8, ro	0x04	number of units monitored
	0x01	Consumer heartbeat time device 1	u32, rw	0x00	00nn tttt tttt: monitoring time in ms nn: module number of the device to be monitored (if nn or tttt = 0, no monitoring)
	0x02	heartbeat time device 2	u32, rw	0x00	like Consumer heartbeat time device 1
	0x03	heartbeat time device 3	u32, rw	0x00	like Consumer heartbeat time device 1
	0x04	heartbeat time device 4	u32, rw	0x00	like Consumer heartbeat time device 1
1017	0x00	Producer heartbeat time	u16, rw	0x00	heartbeat time of the unit in ms
1200	0x00	Server SDO	u8, ro	0x02	Number of entries
	0x01	COB-ID Rec SDO	u32, ro	0x600 + Node ID	<ul style="list-style-type: none"> - SDO is valid (bit 31 = 0) - CAN-ID of the Receive SDOs
	0x02	COB-ID Trans SDO	u32, ro	0x580 + Node ID	<ul style="list-style-type: none"> - SDO is valid (bit 31 = 0) - CAN-ID of the Transmit SDOs
1400	0x00	Receive PDO <i>preset value status LEDs and rows of LEDs</i>	u8, ro	0x02	Number of entries Rec PDO; (LED status)
	0x01	COB-ID Rec PDO	u32, rw	0x200 + Node ID	<ul style="list-style-type: none"> - PDO is valid (bit 31 = 0) - CAN-ID of Rec PDOs
	0x02	Trans Type Rec PDO <i>preset value status LEDs and rows of LEDs</i>	u8, rw	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; The outputs are updated only after "n" synch objects n = 0x01 (1)...0xF0 (240) 0xFC not implemented 0xFD not implemented 0xFE = asynch man. spec. event; outputs are updated immediately 0xFF = asynch device profile event; outputs are updated immediately

Object directory

Communication profiles; index 1000 to 1FFF

Index	S-Idx	Name	Type	Default	Description
1600	0x00	Mapping Receive PDO	u32, rw	0x03	number of integrated application objects for triggering of the status LEDs and the rows of LEDs.
	0x01	PDO mapping for the 1st application object <i>preset value status LEDs</i>	u32, rw	0x6200 01	Idx 6200 01 contains 3-byte LED states, triggered by 2 bits each, e.g. byte 0: 0b xxxx xx 00 LED 1 OFF 0b xxxx xx 01 LED 1 flashing time 1 0b xxxx xx 10 LED 1 flashing time 2 0b xxxx xx 11 LED 1 ON 0b xxxx 00 xx LED 2 OFF 0b xxxx 01 xx LED 2 flashing time 1 0b xxxx 10 xx LED 2 flashing time 2 0b xxxx 11 xx LED 2 ON 0b xx 00 xxxx LED 3 OFF 0b xx 01 xxxx LED 3 flashing time 1 0b xx 10 xxxx LED 3 flashing time 2 0b xx 11 xxxx LED 3 ON etc. (also see page 46, CR1500 InOutStruct)
	0x02	PDO mapping for the 2nd application object <i>preset value row of LEDs 1</i>	u32, rw	0x6410 01	Idx 6410 01 contains preset value of the row of LEDs 1 (top) value range from 0...10 0 = all LED segments OFF 1 = 1st LED segment ON 2 = 2nd LED segment ON or 1st and 2nd LED segment ON according to the selected mode 3 = 3rd LED segment ON or 1st to 3rd LED segment ON according to the selected mode etc. 10 = 10th LED segment ON or all LED segments ON according to the selected mode values >10 are internally rounded off to 10
	0x03	PDO mapping for the 3rd application object <i>preset value row of LEDs 2</i>	u32, rw	0x6410 02	Idx 6410 02 contains preset value of the row of LEDs 2 (bottom) value range from 0...10 0 = all LED segments OFF 1 = 1st LED segment ON 2 = 2nd LED segment ON or 1st and 2nd LED segment ON according to the selected mode 3 = 3rd LED segment ON or 1st to 3rd LED segment ON according to the selected mode etc. 10 = 10th LED segment ON or all LED segments ON according to the selected mode values >10 are internally rounded off to 10

Object directory

Communication profiles; index 1000 to 1FFF

Index	S-Idx	Name	Type	Default	Description
1800	0x00	Transmit PDO <i>function and arrow key status</i>	u8, ro	0x05	number of entries Trans PDO; binary inputs (key status)
	0x01	COB-ID Trans PDO	u32, rw	0x180 + Node ID	- PDO is valid (bit 31 = 0) - CAN-ID of Trans PDOs
	0x02	Trans Type Trans PDO <i>Funktions- und Pfeiltasten- Status</i>	u8, rw	0xFF	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; outputs are only updated after "n" synch objects n = 0x01 (1)...0xF0 (240) 0xFC not implemented 0xFD not implemented 0xFE = asynch man. spec. event; PDO is immediately transferred if the input states change 0xFF = asynch device profile event; PDO is immediately transferred if the input states change
	0x03	inhibit time	u16, rw	–	not implemented
	0x04	(reserved)	u8, rw	–	not implemented
	0x05	event timer	u16, rw	0x00	max. transmission pause of the PDOs, if Trans Type = acyclic setting in ms range = 0...65535 ms (default = 0 ms = Off)
1A00	0x00	Mapping Transmit PDO	u32, rw	0x01	number of integrated application objects to read the key status
	0x01	PDO mapping for the 1st application object <i>function and arrow key status</i>	u32, rw	0x6000 01	Idx 6000 01 contains 4-byte key states, e.g. push button operation (Idx 2000 xx = 0x00) Byte 0: 0b xxxx xx01 key 1 pressed 0b xxxx xx11 key 1 pressed and filter time elapsed 0b xxxx 01xx key 2 pressed 0b xxxx 11xx key 2 pressed and filter time elapsed 0b xx01 xxxx key 3 pressed 0b xx11 xxxx key 3 pressed and filter time elapsed 0b 01xx xxxx key 4 pressed 0b 11xx xxxx key 4 pressed and filter time elapsed etc. (also see page 46, CR1500 InOutStruct)

Programming

General remarks

The keypad module must be initialised as CANopen slave with the CANopen start functions "COP_MSTR_BOOTUP" and "COP_MSTR_MAIN" by the R360 master and set to the state "OPERATIONAL" (status LEDs flashing 5 times).

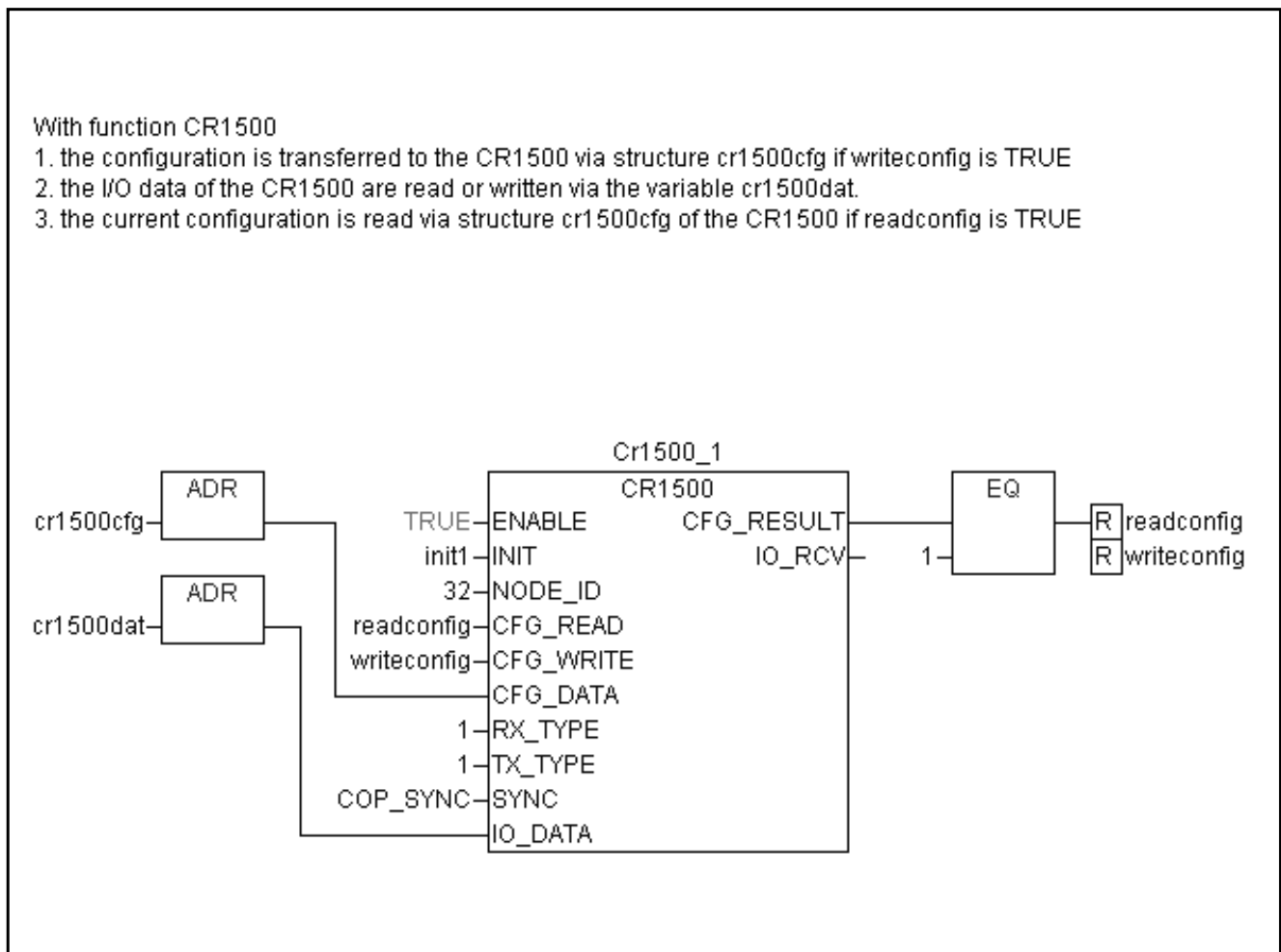
Programming function

If the function "CR1500" is integrated into the program, this automatically ensures a continuous updating of the key and LED states in the controller. The function "CR1500" can be found in the library "CR1500_C.lib" of the ifm programming software CoDeSys.

If no configuration data are transferred to the keypad module, the device uses the default values set at the factory.

Before start-up change the node ID of the keypad module set at the factory, if necessary. Check whether the baud rate of the master and that of the module are identical or set accordingly.

Default values: Node-ID = 0x20 (= 32)
Baud rate = 0x04 (= 125 Kbits/s)



Screenshot of the function

Data structures

The CR1500 configuration and I/O data are transmitted via data structures. The structure as well as other variable types must be declared in the declaration part. For configuration data the declaration part can already contain an assignment of values.

In the program access to a structure component can be represented as follows:

The screenshot shows the EcoPlus - CR1500.pro programming environment. The main window displays a configuration structure declaration for 'cr1500cfg'. The code is as follows:

```

0069
0070 (*The configuration of the CR1500 can be assigned directly in the deccaration part*)
0071 cr1500cfg: CR1500ConfigStruct :=(GUARDTIME := T#500ms,
0072     LIFETIME :=3,
0073     Key1 := TRUE,
0074     Key2 := TRUE,
0075     Key3 := TRUE,
0076     Key4 := TRUE,
0077     Key5 := TRUE,
0078     Key6 := TRUE,
0079     Key7 := TRUE,
0080     Key8 := TRUE,
0081     Key9 := TRUE,
0082     Key10 := TRUE,
0083     Key11 := TRUE,
0084     Key12 := TRUE,
0085     KeyDelay := 500,
0086     LedBarMode1 := 10,
0087     LedBarMode2 := 10,
0088     LedBarColorChg1 :=3,
0089     LedBarColorChg2 := 3,
0090     StatusLedFlashTime1 := 500,
0091     StatusLedFlashTime2 := 200);
0092
0093
  
```

Screenshot of the programming platform

More CoDeSys programming examples of the keypad module can be obtained from ifm electronic gmbh on request.

- **Function:** CR1500
- **Library:** CR1500_C.lib
- **Purpose:** Sets the parameters and reads the configuration and I/O data of the keypad module
CR1500

CR1500	
ENABLE	CFG_RESULT
INIT	IO_RCV
NODE_ID	
CFG_READ	
CFG_WRITE	
CFG_DATA	
RX_TYPE	
TX_TYPE	
SYNC	
IO_DATA	

■ Parameter

Name	Data type	Description
Inputs		
ENABLE	BOOL	TRUE: function processing
INIT	BOOL	TRUE: function initialisation FALSE: cyclic function call
NODE_ID	BYTE	node identifier
CFG_READ	BOOL	TRUE: read the current configuration of the module
CFG_WRITE	BOOL	TRUE: write the current configuration of the module
CFG_DATA	DWORD	address of the configuration data (data structure)
RX_TYPE	BYTE	receive transmission type (default = 1; synch cyclic) (seen from the controller)
TX_TYPE	BYTE	transmit transmission type (default = 1; synch cyclic) (seen from the controller)
SYNC	BOOL	CANopen synchronisation cycle (system variable COB_SYNC)
IO_DATA	DWORD	address of the input/output data (data structure)
Outputs		
CFG_RESULT	BYTE	1 = configuration read or written successfully 2 = configuration not yet read or written 3 = configuration cannot be read or written (missing or incorrect node ID or faulty device)
IO_RCV	BOOL	TRUE: for one cycle if new data were transmitted

If not described otherwise, a "FALSE" signal for boolean data types is always the negation of the described "TRUE" signal.

■ Data structure: CR1500 ConfigStruct

■ Purpose:

Parameter and configuration data can be written or read.

The data structure is assigned to the function input "CFG_DATA" via the ADR operator.

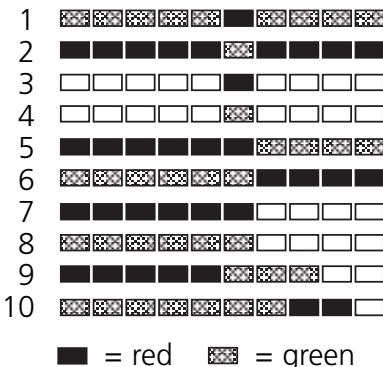
```

TYPE CR1500 ConfigStruct
STRUCT
  GUARDTIME: TIME;
  LIFETIME: BYTE;
  Key1: BOOL;
  Key2: BOOL;
  Key3: BOOL;
  Key4: BOOL;
  Key5: BOOL;
  ⋮
  Key12: BOOL;
  KeyDelay: WORD;
  LedBarMode1: BYTE;
  LedBarMode2: BYTE;
  LedBarColorChg1: BYTE;
  LedBarColorChg2: BYTE;
  StatusLedFlashTime1: WORD;
  StatusLedFlashTime2: WORD;
END_STRUCT
END_TYPE

```

■ Structure components (ConfigStruct)

Name	Data Type	Description
GUARDTIME	TIME	guarding time of the keypad module [ms]
LIFETIME	BYTE	lifetime of the keypad module
Key1...12	BOOL	push button operation FALSE = push button operation (default) TRUE = toggle operation
KeyDelay	WORD	delay time (filter) for key evaluation value range 0...65535 ms (valid for all keys) (default 0 ms = Off)
LedBarMode1 LedBarMode2	BYTE BYTE	operating mode row of LEDs 1/2 (value range 0...10) 0 = row of LEDs not active 1 = red individual segment in a row of LEDs lighting green (default) 2 = green individual segment in a row of LEDs lighting red 3 = red individual segment 4 = green individual segment 5 = red row of segments in a row of LEDs lighting green 6 = green row of segments in a row of LEDs lighting red 7 = red row of segments 8 = green row of segments 9 = red row of segments with defined colour change point (here e.g. 5) 10 = green row of segments with defined colour change point (here e.g. 7) (also see page 33, row of LEDs)
LedBarColorChg1 LedBarColorChg2	BYTE BYTE	colour change point row of LEDs 1/2 value range 1...10 = colour changes after segment "n" only possible for operating modes 9 and 10
StatusLedFlashTime1 StatusLedFlashTime2	WORD WORD	flashing time 1 for status LEDs (default 500 ms = 2 Hz) flashing time 2 for status LEDs (default 200 ms = 5 Hz) value range = 0...65535 ms



■ Data structure: CR1500 InOutStruct

- Purpose:
Current I/O data are read or written. The data structure is assigned to the function input "IO_DATA" via the ADR operator.

```

TYPE CR1500 InOutStruct
STRUCT
  Key1_Direct: BOOL;      (BinIn)
  Key1_Delay:  BOOL;      (BinIn)
  ...
  Key12_Delay: BOOL;      (BinIn)
  KeyLeft1_Direct: BOOL;  (BinIn)
  KeyLeft1_Delay: BOOL;   (BinIn)
  KeyRight1_Direct: BOOL; (BinIn)
  KeyRight1_Delay: BOOL;  (BinIn)
  ...
  KeyRight2_Delay: BOOL;  (BinIn)
  Led1_1:  BOOL;          (BinOut, LowBit)
  Led1_2:  BOOL;          (BinOut, HighBit)
  ...
  Led12_2: BOOL;          (BinOut, HighBit)
  LedBar1_Value: BYTE;    (AnalogOut)
  LedBar2_Value: BYTE;    (AnalogOut)
END_STRUCT
END_TYPE

```

■ Structure components (function keys)

Name	Data type	Key	Bit status in ... push button operation	toggle operation	Byte	Bit
Key1_Direct Key1_Delay	BOOL BOOL	1	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched	0	0 1
Key2_Direct Key2_Delay	BOOL BOOL	2	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		2 3
Key3_Direct Key3_Delay	BOOL BOOL	3	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		4 5
Key4_Direct Key4_Delay	BOOL BOOL	4	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		6 7
Key5_Direct Key5_Delay	BOOL BOOL	5	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched	1	0 1
Key6_Direct Key6_Delay	BOOL BOOL	6	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		2 3
Key7_Direct Key7_Delay	BOOL BOOL	7	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		4 5
Key8_Direct Key8_Delay	BOOL BOOL	8	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		6 7
Key9_Direct Key9_Delay	BOOL BOOL	9	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched	2	0 1
Key10_Direct Key10_Delay	BOOL BOOL	10	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		2 3
Key11_Direct Key11_Delay	BOOL BOOL	11	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		4 5
Key12_Direct Key12_Delay	BOOL BOOL	12	TRUE if pressed TRUE when delay time has elapsed	TRUE if switched TRUE if switched		6 7

■ Structure components (arrow keys)

Name	Data type	Key	Bit status in ...		Byte	Bit
			push button operation	toggle operation		
KeyLeft1_Direct	BOOL	arrow key upper left	TRUE if pressed	(not supported)	3	0
KeyLeft1_Delay	BOOL		TRUE when delay time has elapsed			1
KeyRight1_Direct	BOOL	arrow key upper right	TRUE if pressed	(not supported)		2
KeyRight1_Delay	BOOL		TRUE when delay time has elapsed			3
KeyLeft2_Direct	BOOL	arrow key lower left	TRUE if pressed	(not supported)		4
KeyLeft2_Delay	BOOL		TRUE when delay time has elapsed			5
KeyRight2_Direct	BOOL	arrow key lower right	TRUE if pressed	(not supported)		6
KeyRight2_Delay	BOOL		TRUE when delay time has elapsed			7

■ Structure components (status LEDs and rows of LEDs)

Name	Data type	LED	Operating status ...				Byte	Bit
			OFF	Flashing 1	Flashing 2	ON		
LED1_1 <i>(Low)</i>	BOOL	1	FALSE	TRUE	FALSE	TRUE	0	0
LED1_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		1
LED2_1 <i>(Low)</i>	BOOL	2	FALSE	TRUE	FALSE	TRUE		2
LED2_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		3
LED3_1 <i>(Low)</i>	BOOL	3	FALSE	TRUE	FALSE	TRUE		4
LED3_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		5
LED4_1 <i>(Low)</i>	BOOL	4	FALSE	TRUE	FALSE	TRUE		6
LED4_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		7
LED5_1 <i>(Low)</i>	BOOL	5	FALSE	TRUE	FALSE	TRUE	1	0
LED5_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		1
LED6_1 <i>(Low)</i>	BOOL	6	FALSE	TRUE	FALSE	TRUE		2
LED6_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		3
LED7_1 <i>(Low)</i>	BOOL	7	FALSE	TRUE	FALSE	TRUE		4
LED7_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		5
LED8_1 <i>(Low)</i>	BOOL	8	FALSE	TRUE	FALSE	TRUE		6
LED8_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		7
LED9_1 <i>(Low)</i>	BOOL	9	FALSE	TRUE	FALSE	TRUE	2	0
LED9_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		1
LED10_1 <i>(Low)</i>	BOOL	10	FALSE	TRUE	FALSE	TRUE		2
LED10_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		3
LED11_1 <i>(Low)</i>	BOOL	11	FALSE	TRUE	FALSE	TRUE		4
LED11_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		5
LED12_1 <i>(Low)</i>	BOOL	12	FALSE	TRUE	FALSE	TRUE		6
LED12_2 <i>(High)</i>	BOOL		FALSE	FALSE	TRUE	TRUE		7
LEDBar1_Value	BYTE	LED row 1	Analogue value for row of LEDs 1 (top) (0...10)				3	
LEDBar2_Value	BYTE	LED row 2	Analogue value for row of LEDs 2 (bottom) (0...10)				4	

Terms and abbreviations

0b ...	binary value (for bit coding), e.g. 0b0001 0000
0x ...	hexadecimal value, e.g. 0x64 (= 100 decimal)
Baudrate	transmission speed (1 baud = 1 bit/s)
CAL	CAN Application Layer CAN-based network protocol on application level
CAN	Controller Area Network (bus system for use in mobile applications)
CAN_H	CAN-High; CAN connection /cable with high voltage level
CAN_L	CAN-Low; CAN connection /cable with low voltage level
CANopen	CAN-based network protocol on application level with an open configuration interface (object directory)
CiA	"CAN in Automation e.V." (user and manufacturer organisation in Germany /Erlangen) Definition and control body for CAN and CAN-based network protocols
CiA DS	Draft Standard (published CiA specification which usually has not been modified or supplemented for one year)
CiA DSP	Draft Standard Proposal (published CiA specification draft)
CiA WD	Work Draft (work draft accepted for discussion within CiA)
CiA DS 301	Specification for CANopen communication profile; describes the basic communication between network participants, such as the transfer of process data in real time, the exchange of data between units or the configuration stage. Depending on the application this is completed by the following CiA specifications:
CiA DS 401	Device profile for digital and analog I/O modules
CiA DS 402	Device profile for drives
CiA DS 403	Device profile for HMI
CiA DS 404	Device profile for measurement and control technology
CiA DS 405	Specification for interfaces to programmable systems (IEC 1131)
CiA DS 406	Device profile for encoders
CiA DS 407	Application profile for local public transport
COB	CANopen Communication Object (PDO, SDO EMCY, ...)
COB ID	CANopen Identifier of a Communication Object
Communication cycle	the synchronisation time to be monitored, max. time between 2 Sync objects
EMCY Object	Emergency Object (alarm message, device indicates an error)
Error Reg	Error Register (entry with an error code)
Guarding Error	Node or network participant could or can no longer be found Guard Master: one or several slaves no longer reply Guard Slave: no polling of the slave
Guard Time	During this time the network participant expects a "Node Guarding" of the network master
Heartbeat	Cyclic monitoring with parameter setting among network participants. In contrast to "node guarding" no superior NMT master is required.
ID	Identifier; identifies a CAN message. The numerical value of the ID also contains a priority for the access to the bus system ID 0 = top priority
Identifier	see ID
Idx	index; together with the S index it forms the address of an entry in the object directory
Life Time Factor	number of attempts in case of a missing Guarding reply
Monitoring	is used to describe the error class (guarding monitoring, synch etc.)
NMT	network management
NMT master/slaves	The NMT master controls the operating states of the NMT slaves
Node Guarding	adjustable cyclic monitoring of slave network participants by a higher master node as well as the monitoring of this polling process by the slave participants

Node ID	node identifier (identification of a participant in the CANopen network)
Object (also OBJ)	term for data/messages which can be exchanged in the CANopen network
Object directory	contains all CANopen communication parameters of a device as well as device-specific parameters and data Access to the individual entries is possible via the index and S index.
Operational	Operating state of a CANopen participant In this mode SDOs, NMT commands and PDOs can be transferred.
PDO	Process Data Object; in the CANopen network for transfer of process data in real time; such as the speed of a motor PDOs have a higher priority than SDOs; in contrast to the SDOs they are transferred without confirmation. PDOs consist of a CAN message with identifier and up to 8 bytes of user data.
PDO Mapping	describes the application data transferred with a PDO.
Pre-Op	Preoperational; operating state of a CANopen participant. After application of the supply voltage each participant automatically goes into this state. In the CANopen network only SDOs and NMT commands can be transferred in this mode but no process data.
Prepared	(also stopped) operating state of a CANopen participant In this mode only NMT commands are transferred.
Rec PDO (also Rx PDO)	Receive Process Data Object
ro	read only (unidirectional)
rw	read-write (bidirectional)
RX-Queue	reception buffer
s16	data type signed 16 bit
SDO	Service Data Object; With this object direct access to the object directory of a network participant is possible (read/write). An SDO can consist of several CAN messages. The transfer of the individual messages is confirmed by the addressed participant. With the SDOs devices can be configured and parameters can be set.
Server SDO	process and parameter set to make the object directory of a network participant available to other participants (clients).
S-Idx (also SIdx)	Subindex within the object directory of a CANopen device
Start Guarding	start node guarding
str	data type string (variable for strings such as text "load")
Sync Error	missing Sync OBJ in the adjustable communication cycle
Sync object	synchronisation object for simultaneous update in the complete network or for accepting process data of the respective parameterised PDOs.
Sync Windows	time during which the synchronous PDOs have to be transferred
Time Stamp	time stamp to align existing clocks in network participants
Trans Type	type of process data transmission; synchronous, asynchronous
Trans PDO (also Tx PDO)	transmit process data object
Trans SDO (also Tx SDO)	transmit service data object
Tx Queue	(transmit) transmission buffer
u8 (16, 32)	data type unsigned 8 (16, 32) bits
wo	write only