Dieses Manual ist zweisprachig ausgeführt. Die deutschsprachige Dokumentation beginnt auf Seite 3

If you would prefer to read this manual in English please turn to page 129

B&R KOMPAKTSTEUERUNG
ANWENDERHANDBUCH

Version: 2.00 (Dezember 1995)
Herausgeber: Bernecker und Rainer Industrie-Elektronik GmbH.
Best. Nr.: MABRCOMP1-0E

Inhaltliche Änderungen dieses Handbuches behalten wir uns ohne Ankündigung vor. Die Bernecker und Rainer Industrie-Elektronik GmbH haftet nicht für technische oder drucktechnische Fehler und Mängel in diesem Handbuch. Außerdem übernimmt die Bernecker und Rainer Industrie-Elektronik GmbH keine Haftung für Schäden, die direkt oder indirekt auf Lieferung, Leistung und Nutzung dieses Materials zurückzuführen sind.

B&R KOMPAKTSTEUERUNG

Inhalt:

1. Allgemeines 9

2. Bestellnummern 10
   2.1 B&R Kompaktsteuerung 10
   2.2 Relais-Aufsteckkarte 11
   2.3 Standardsoftware 11
   2.4 Feldklemmen 11

3. Technische Daten 12

4. Abmessungen 13

5. Montage 14
   5.1 Einbaurichtlinien 14
   5.2 Es gibt zwei Montagemöglichkeiten 14
      5.2.1 Direkt an Schaltschrankrückwand 14
      5.2.2 Auf Hutschiene 14

6. Beschreibung der Komponenten 16

7. Spannungsversorgung 17

8. Sicherung 18

9. LEDs 19

10 Digitaleingänge 21
   10.1 Technische Daten 21
   10.2 Eingangsschaltung 22
   10.3 Verdrahtungsschema für Digitaleingänge 22

11. Digitalausgänge 23
   11.1 Technische Daten 23
   11.2 Ausgangsschaltung 24
   11.3 Verdrahtungsschema 24
   11.4 Schalten ohmsch-induktiver Lasten 25

12. Analogeingänge 27
12.1 Technische Daten 30
12.2 Pinbelegung 31
12.3 Eingangsschaltung 31
12.4 Schirm erden 32
12.5 Verdrahtungsschema 32
12.6 Softwaremäßige Bedienung 35
12.7 AINJ - Analogeingang Kompaktsteuerung 38
12.8 Analogumwandlung 41

13. Analogausgänge 43
13.1 Technische Daten 43
13.2 Pinbelegung 44
13.3 Ausgangsschaltung 44
13.4 Schirm erden 45
13.5 Verdrahtungsschema 45
13.6 Softwaremäßige Bedienung 46
13.7 AOTF - Analogausgang Kompaktsteuerung 47
13.8 Analogwandlung 48

14. Zählereingänge und Interrupteingang 49
14.1 Verdrahtungsschema 51
14.2 Bedienung der Zähler 51
14.3 CMDC - Zählerauswertung (BRCOMP) 52
14.4 CMDD - Zähler 0 Auswertung (BRCOMP) 54
14.5 Interruptmodus 56
14.6 Bedienung über AWL 57
14.6.1 Betriebssystemadressen der Zähler 57
14.6.2 Steuerregister $2120 58
14.6.3 Modusregister $2121 58
14.6.4 Initialisierungsregister $2122 59
14.6.5 Adressen der Betriebssystemroutinen 59
14.6.6 Funktion der Zähler 60
14.6.7 Zähler initialisieren 61
14.6.8 Aufruf von Betriebssystemroutinen 62

15. Relais-Aufsteckkarte für BRCOMP 67
15.1 Allgemeines 67
15.2 Relaisgruppen 67
15.3 Technische Daten 68
15.4 Pinbelegung 68
15.5 Abmessungen 69
15.6 Einbaurichtlinien 70

15.7 Montage 70
15.8 Kompaktsteuerung und Relais-Aufsteckkarte 72
15.9 Verbindung mit der Kompaktsteuerung 73
15.10 Relais-Versorgung 73
15.11 Schaltkreis 74

16. IF1 - RS232 Schnittstelle 75
16.1 Technische Daten 75
16.2 Pinbelegung 75
16.3 Verdrahtungsschema 76
16.4 Schirmung und Erdung 76
16.5 Softwaremäßige Bedienung 76
16.6 Compact MMI P120 und MMI P121 76

17. IF2 - RS232/RS485 Schnittstelle 77
17.1 Technische Daten 77
17.2 Pinbelegung 77
17.3 Verdrahtungsschema 78
17.4 Schirmung und Erdung 79
17.5 Softwaremäßige Bedienung 79

18. Bedienung der Schnittstellen IF1 und IF2 80
18.1 Verwendung des Schnittstelleninterrupts 85

19. IF3 - PATA/SSI Schnittstelle 89

20. IF4 - B&R Online Schnittstelle 90

21. IF5 - CAN Bus 91
21.1 Merkmale des CAN Busses 91
21.2 B&R und CAN 91
21.3 Buslänge und Kabeltyp 91
21.4 Anschlußbelegung 93
21.5 Verdrahtung 93
21.5.1 Verdrahtungsprinzip 94
21.5.2 Blockschaltbild eines Knotens 94
21.6 Bedienung des CAN Busses 95

22. Einschaltverhalten (Power-On) 96

23. Befehlssatz 96

24. Mathematikroutinen 97

25. Systemspeicherstellen 101

26. First Scan Flag 103

27. Lithium-Batterie 104
28. Zeittakte 106
29. Zeitimpulse 106
30. Echtzeituhr 107
31. Softwarezeiten 108
32. Inport/Outport Adresse $3400 110
33. Zusätzliches Anwender-EEPROM 111
33.1 Daten lesen 112
33.2 Daten schreiben 114
34. Inport Adresse $3480 117
35. Watchdog 118
35.1 Softwarewatchdog 118
35.2 Hardwarewatchdog 118
36. Timerinterruptroutinen 119
37. Fehlermeldungen 121
38. Stichwortverzeichnis 123

1. ALLGEMEINES

Mit der Steuerungsfamilie MINI-/MULTICONTROL bietet B&R ein Steuerungssystem an, das aufgrund seiner modularen Struktur und 100%igen Softwarekompatibilität individuell den kundenspezifischen Erfordernissen angepaßt werden kann.

Um diesen Anforderungen noch mehr gerecht zu werden, hat B&R seine Produktpalette mit einer neuen Kompaktsteuerung erweitert. Es handelt sich dabei um eine Steuerung die unterhalb der MINICONTROL angesiedelt ist.
Bei ihrer Entwicklung wurde auf die 100%ige Softwarekompatibilität zur MINICONTROL geachtet. Sollte daher aufgrund gestiegener Anforderungen ein Umstieg auf die MINICONTROL erforderlich sein, geht die geleistete Softwarearbeit nicht verloren.

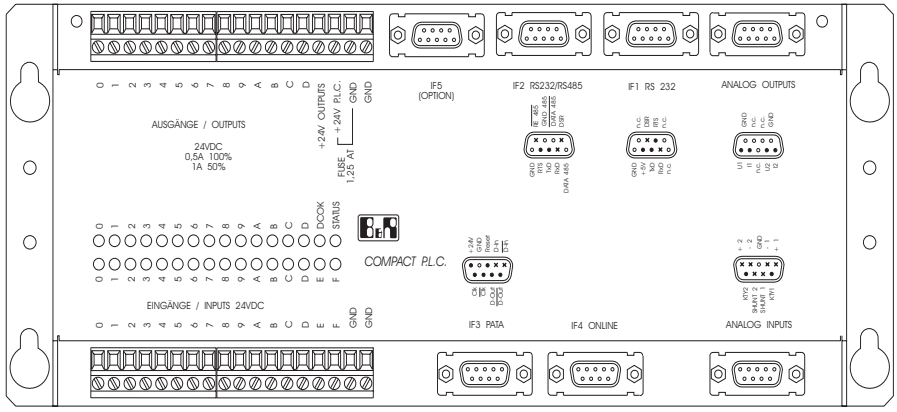
Obwohl aufgrund eines möglichst günstigen Preises auf eine modulare Bauweise verzichtet wurde, ist die Kompaktsteuerung durch ihre Konzeption und Ausstattung universell einsetzbar.

Ausstattung

- digitale Ein-/Ausgänge
- Relais-Aufsteckkarte
- analoge Eingänge (Spannung, Strom und Temperatur)
- analoge Ausgänge (Spannung und Strom)
- Ereigniszähler
- Interrupteingang
- Referenzeingang
- Kanäle A und B für Positionieraufgaben
- 2 serielle Schnittstellen
- CAN Bus (BRCOMP2-0)
- B&R Standard PATA Schnittstelle (für MINICONTROL-Bedientableau)
- SSI Schnittstelle für Anschluß von Absolutgebern
- B&R Online Schnittstelle
- 16 KBytes EEPROM Anwenderspeicher
- 16 KBytes Zusatz-EEPROM
- RS485 Netzwerk (B&R MININET)
- Bedientableaus: MINICONTROL-Bedientableau
Compact MMI P120 und MMI P121
andere Produkte aus der PANELWARE Familie

2. BESTELLNUMMERN

2.1 B&R KOMPAKTSTEUERUNG



Kompaktsteuerung, mit 6303 Prozessor, 16 KBytes Anwenderspeicher, ca. 4 ms / K Anweisungen, 16 Digitaleingängen, 14 Digitalausgängen, 2 Analogeingängen, 2 Analogausgängen, Kanäle A und B für Positionieraufgaben, Referenzeingang, Ereigniszähler, Interrupteingang, B&R Online Schnittstelle, RS232 Schnittstelle (nicht galvanisch getrennt), RS232/RS485 Schnittstelle (RS485 galvanisch getrennt), PATA/SSI Schnittstelle,

ohne CAN Bus	**BRCOMP1-0**
mit CAN Bus auf IF5	**BRCOMP2-0**

2.2 RELAIS-AUFSTECKKARTE

Für die Kompaktsteuerung ist eine Relais-Aufsteckkarte erhältlich. Dadurch erhält man zu den vorhandenen 14 Transistorausgängen zusätzlich 16 Relaisausgänge (A 060 - A 06F).

Die Relais-Aufsteckkarte ist im Abschnitt "15. Relais-Aufsteckkarte für BRCOMP" beschrieben.

2.3 STANDARDSOFTWARE

Speziell für die Kompaktsteuerung wurde eine Diskette mit der wichtigsten Standardsoftware zusammengestellt. Folgende Software ist auf der Diskette gespeichert:

- allgemeine Hilfsprogramme (Standardsoftwarepaket 1)
- Bedientableausoftware
- B&R MININET
- Positionieren

Sprache	Bestellnummer
deutsch	SWSPSBRC01-0
englisch	SWPLCBRC01-0

2.4 FELDKLEMMEN

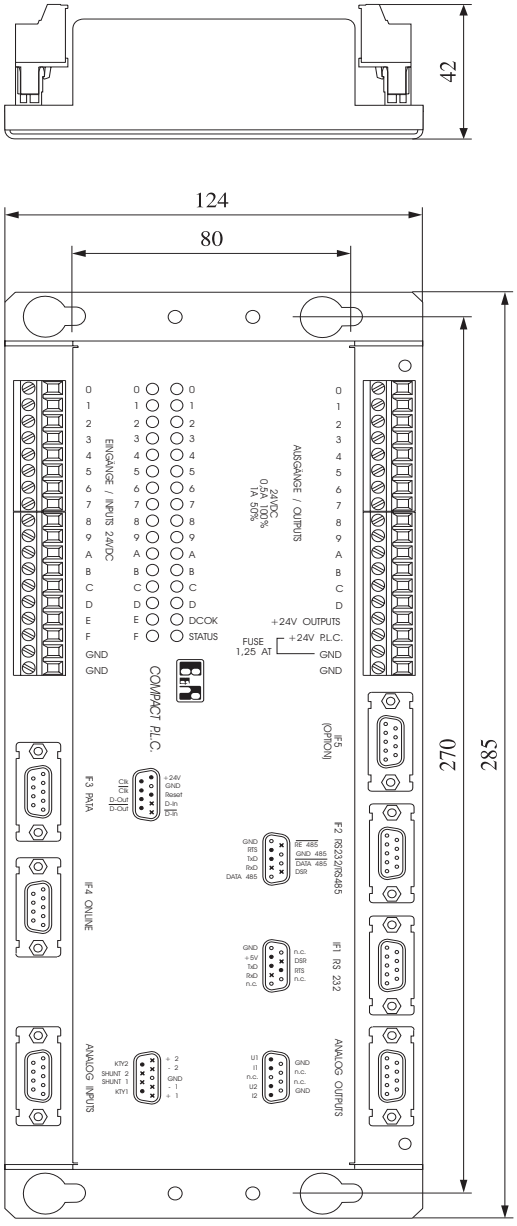
Als Zubehör kann ein Satz PHOENIX-Feldklemmen bestellt werden (BRTB0218-0). Im Lieferumfang sind enthalten:

Stück	Feldklemme
2	8polig
2	10polig

3. TECHNISCHE DATEN

Prozessor	6303
Bearbeitungszeit	ca. 4 ms / K Anweisungen
Anwenderprogrammspeicher	16 KBytes RAM/EEPROM
Erweiterungsspeicher	16 KBytes EEPROM (für Daten)
Status-LED	rot
Anzahl 8 Bit-Speicher remanent nicht remanent	7168 7148 20
Anzahl 1 Bit-Speicher remanent nicht remanent	800 300 500
Uhrzeit/Datum	Echtzeituhr
Software-Timer	64
Zeittakte/Zeitimpulse	10 ms, 100 ms, 1 s, 10 s
Digitaleingänge	16 vier davon können auch als Zähleingänge verwendet werden
Digitalausgänge	14
Analogeingänge	2
Analogausgänge	2
Serielle Schnittstellen IF1 IF2 IF3 IF4 IF5	 RS232 RS232/RS485 (RS485 galvanisch getrennt) PATA (Ansteuerung MINICONTROL-Bedientableau oder Relais-Aufsteckkarte) SSI (Anschluß von Absolutgebern) B&R Online Schnittstelle CAN Bus (BRCOMP2-0)
Spannungsversorgung	24 VDC ±25 %
max. Leistungsaufnahme ohne Bedientableaus bei 18 V bei 24 V bei 30 V	 6 W 6,5 W 7,5 W
Sicherung	T 1,25 A / 250 V
Soft- und Hardwarewatchdog	JA
Batterieüberwachung	JA
Betriebstemperatur	5 bis 55 °C
Luftfeuchtigkeit	30 - 95 % nicht kondensierend

4. ABMESSUNGEN



5. MONTAGE

5.1 EINBAURICHTLINIEN

Bei Geräten, die starke elektromagnetische Störungen verursachen (z. B. Frequenzumrichter, Transformatoren, Motorregler usw.) ist auf ausreichende räumliche Trennung zu achten. Der Abstand dieser Geräte zur SPS sollte so groß wie möglich sein. Gegebenenfalls ist eine Abschirmung durch Trennbleche (MU-Metall) vorzunehmen.

5.2 ES GIBT ZWEI MONTAGEMÖGLICHKEITEN

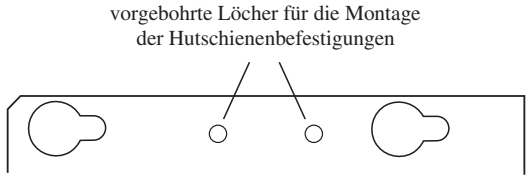
5.2.1 Direkt an Schaltschrankrückwand

Die Kompaktsteuerung kann horizontal oder vertikal montiert werden. Bei vertikaler Montage muß darauf geachtet werden, daß die digitale Ein-/Ausgangsgruppe nach oben gerichtet ist (siehe Zeichnung auf der nächsten Seite).
Das Chassis ist über die vier Befestigungslöcher gut leitend mit der Schaltschrankrückwand zu verschrauben.

Als Schrauben sind M5 Schrauben zu verwenden (Lochabstand: 80 * 270 mm).

5.2.2 Auf Hutschiene

Für die Montage auf einer Hutschiene (Trageschiene DIN EN 50022-35) müssen die zwei mitgelieferten Hutschienenbefestiger (Typ: KSA10) mit der Kompaktsteuerung verschraubt werden.

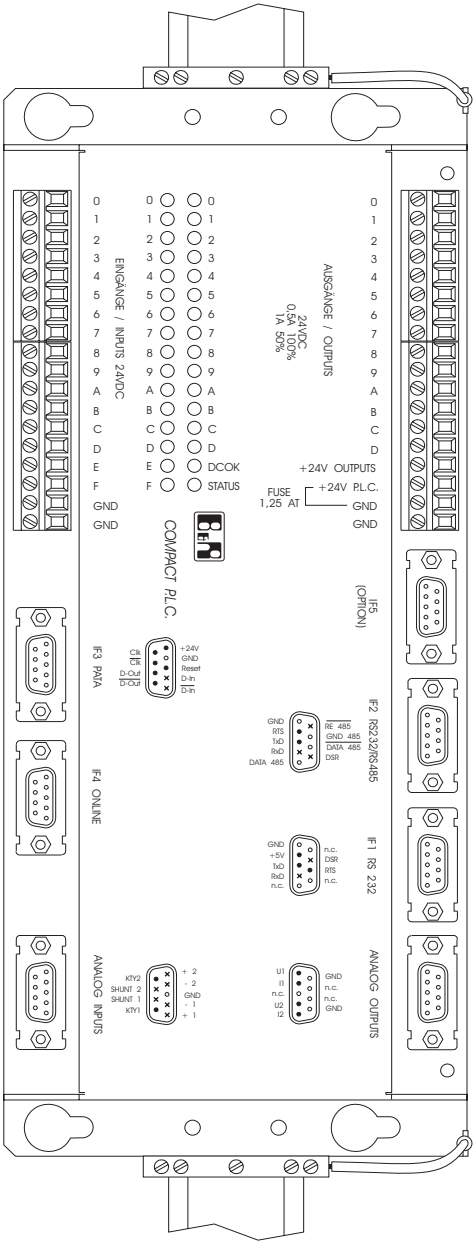


Kompaktsteuerung auf der Hutschiene befestigen.

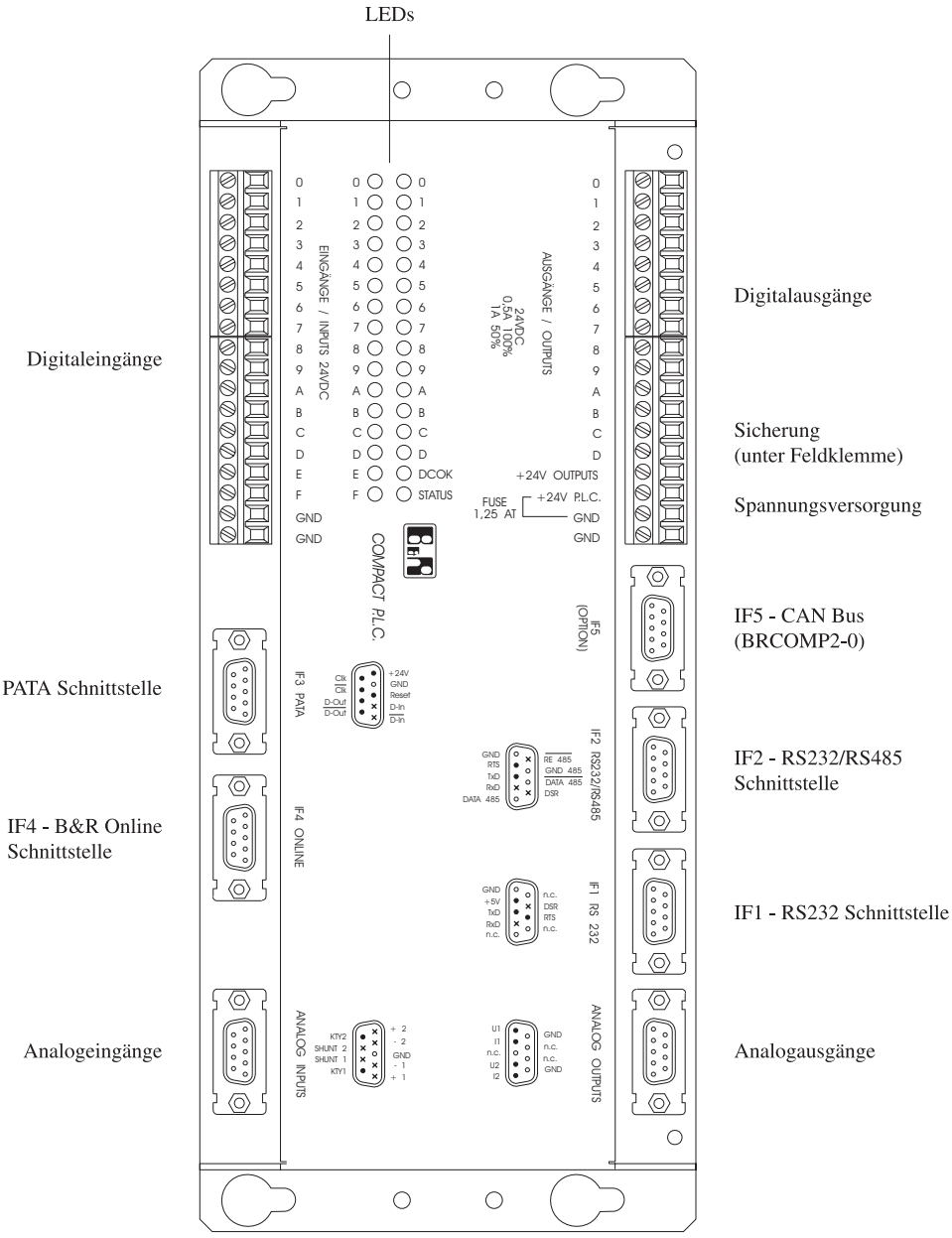
Links und rechts von der Kompaktsteuerung eine Schutzleiterklemme einsetzen. Kompaktsteuerung über die vorbereiteten Kontakte erden (siehe Zeichnung auf der nächsten Seite).

HINWEIS: Die Hutschiene muß geerdet sein.

Der durch die Befestigungslaschen bedingte Abstand zum Nachbarmodul reicht für eine ausreichende Luftzirkulation.



6. BESCHREIBUNG DER KOMPONENTEN



7. SPANNUNGSVERSORGUNG

Die LED DCOK zeigt an, daß die Steuerung versorgt wird. Es gibt zwei Möglichkeiten die Kompaktsteuerung mit Spannung zu versorgen:

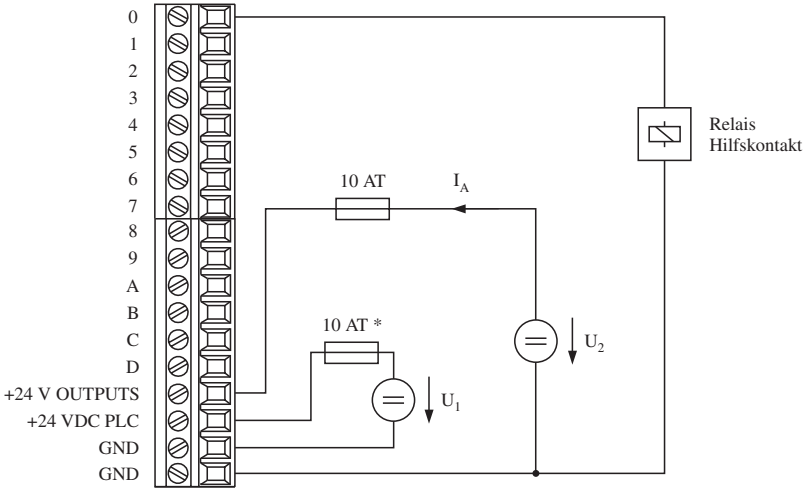
- ein Netzteil: - gemeinsame Versorgung für Ein- und Ausgänge
- zwei Netzteile: - eine für die Steuerung
  - eine für die digitalen Ausgänge

Ein Netzteil

Die Versorgung der Kompaktsteuerung mit einem Netzteil ist der Standardfall (Schaltung siehe Abschnitt "11. Digitalausgänge").

Zwei Netzteile

Steuerung und Ausgänge werden getrennt versorgt.



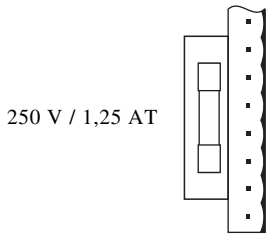
U_1 ... 24 VDC / 1 A
U_2 ... 24 VDC / I_A
I_A ... Summenstrom aller Ausgänge

* 10 A Vorsicherung (träge) zum Schutz des Anschlußsteckers im Kurzschluß- oder Verpolungsfall.

8. SICHERUNG

Die Kompaktsteuerung ist mit einer Sicherung geschützt (250 V / 1,25 AT). Der Sockel für die Sicherung befindet sich unter der Feldklemme für die Digitalausgänge.

HINWEIS: Die Digitalausgänge werden extra versorgt. Die Einspeisung muß über eine eigene Sicherung geführt werden (siehe Abschnitte "7. Spannungsversorgung" und "11. Digitalausgänge").



9. LEDs

32 LEDs informieren den Anwender über den Zustand der Digitaleingänge, der Digitalausgänge, der Spannungsversorgung und der Zentraleinheit.

0 ○ ○ 0
1 ○ ○ 1
2 ○ ○ 2
3 ○ ○ 3
4 ○ ○ 4
5 ○ ○ 5
6 ○ ○ 6
7 ○ ○ 7
8 ○ ○ 8
9 ○ ○ 9
A ○ ○ A
B ○ ○ B
C ○ ○ C
D ○ ○ D
E ○ ○ DCOK
F ○ ○ STATUS

LEDs für	Beschreibung
Digitaleingänge	Die grünen LEDs (0 - F) auf der linken Seite zeigen den Status der digitalen Eingänge an. aus ... Eingang = "0" ein ... Eingang = "1"
Digitalausgänge	Die orangen LEDs (0 - D) zeigen den Status der digitalen Ausgänge an. aus ... Ausgang = "0" ein ... Ausgang = "1"
Versorgung	Die orange LED DCOK zeigt an, daß die Steuerung versorgt wird. aus ... keine Versorgung ein ... Versorgungsspannung o.k.

Status-LED

Die Kompaktsteuerung ist mit einer roten Status-LED ausgestattet, die verschiedene Betriebszustände anzeigt.

LED-Status	Betriebszustand
HI / LO	Anwenderprogramm läuft im RAM
HI / LO	Zentraleinheit ist im HALT-Zustand
HI / LO	Onlinekabel während PROM-Programmieren abgesteckt
HI / LO	Fehler bei der Ausführung des Anwenderprogramms
HI / LO	Anwenderprogramm läuft im PROM

10 DIGITALEINGÄNGE

Die Digitaleingänge wandeln die binären Signale des Prozesses in die Binärwerte 0 und 1 um. Der Zustand der Eingänge wird über grüne Status-LEDs angezeigt.

Die Kompaktsteuerung ist mit 16 Digitaleingängen (E 040 - E 04F) ausgestattet. Die Eingänge E 04C - E 04F können auch für Sonderaufgaben verwendet werden:

- Ereigniszähler
- Interrupteingang
- Referenzeingang
- Inkrementalgeber (Kanäle A und B) für Positionieraufgaben

Die genaue Funktion ist im Abschnitt "14. Zählereingänge und Interrupteingang" beschrieben.

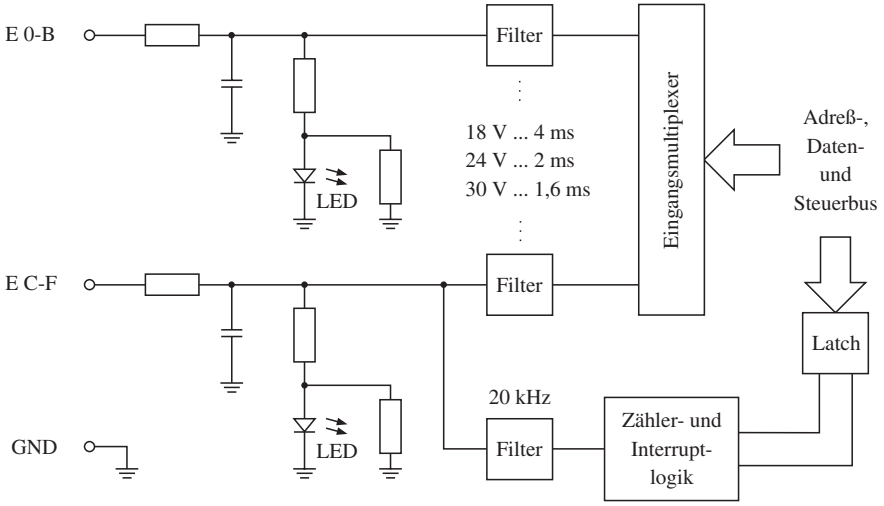
10.1 TECHNISCHE DATEN

Anzahl	16
Ausführung	Typ 1 nach IEC 1131-2
Statusanzeige	über 16 grüne LEDs
Potentialtrennung Eingang ↔ SPS Eingang ↔ Eingang	 NEIN NEIN
Eingangsspannung [1] minimal nominal maximal	 15 VDC 24 VDC 30 VDC
Eingangswiderstand	4 kΩ
Schaltschwellen log. 0 → 1 log. 1 → 0	 min. 15 VDC max. 5 VDC
Eingangsstrom bei 24 VDC	ca. 5 mA
Schaltverzögerung Eingänge 0 - F (log. 0 → 1, log. 1 → 0)	18 V: max. 4 ms 24 V: max. 2 ms 30 V: max. 1,6 ms
Eingangsfrequenz Eingänge C - F	max. 20 kHz
Übernahme der Eingänge durch die Zentraleinheit	automatisch bei Änderung
maximale Spitzenspannung	500 V für 50 µs, max. alle 100 ms [2]

[1] Bei Verwendung der Eingänge C - F als Zähleingänge sollte die Eingangsspannung im Bereich von 24 VDC ± 10 % liegen, damit die maximale Eingangsfrequenz von 20 kHz erreicht werden kann.

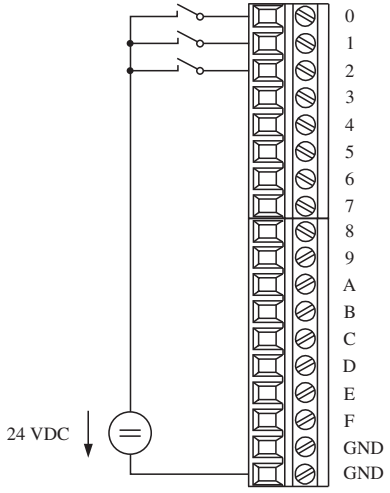
[2] Normimpuls 1,2/50 (IEC 60-2)

10.2 EINGANGSSCHALTUNG



Die Zähler- und Interruptlogik ist im Abschnitt "14. Zählereingänge und Interrupteingang" beschrieben.

10.3 VERDRAHTUNGSSCHEMA FÜR DIGITALEINGÄNGE



11. DIGITALAUSGÄNGE

Digitalausgänge dienen zur Ansteuerung von externen Lasten (Relais, Motoren, Magnetventile usw.). Der Zustand der Ausgänge wird über orange Status-LEDs angezeigt.

Die Kompaktsteuerung ist mit 14 Transistorausgängen (A 050 - A 05D) ausgestattet.

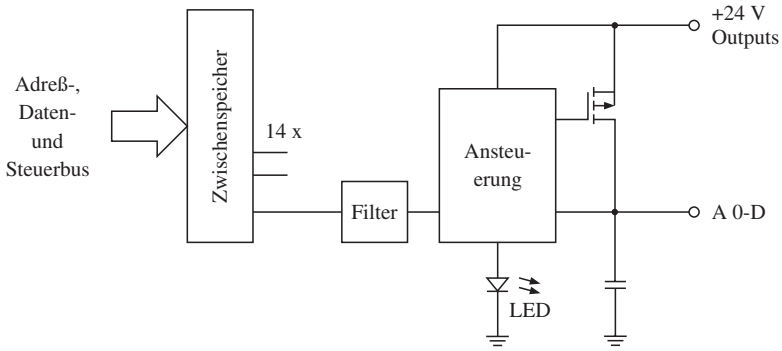
11.1 TECHNISCHE DATEN

Anzahl	14
Statusanzeige	über 14 orange LEDs
Potentialtrennung Ausgang ↔ SPS Ausgang ↔ Ausgang	 NEIN NEIN
Versorgungsspannung minimal nominal maximal	 18 VDC 24 VDC 30 VDC
Schaltstrom 50 % Gleichzeitigkeit 100 % Gleichzeitigkeit	 1,0 A [1)] 0,5 A
Schaltverzögerung log. 0 → 1 log. 1 → 0	 ca. 200 µs ca. 200 µs
Restspannung der Transistoren	< 1 V bei 1 A
Schutzbeschaltung [2)]	intern vorhanden
dauerkurzschlußfest	JA
Überlastschutz	automatische Abschaltung bei thermischer Überlastung

[1)] **HINWEIS:** Wenn bei 50 % Gleichzeitigkeit ein Strom von 1 A fließt, darf die Umgebungstemperatur nicht über 40 °C liegen.

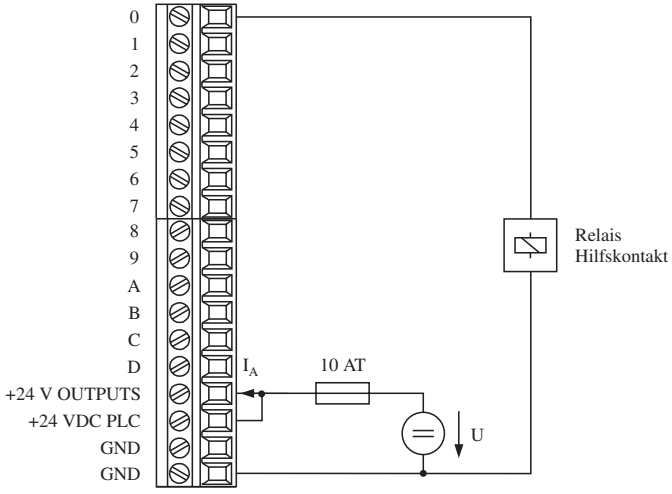
[2)] Eine externe Schutzbeschaltung kann unter Umständen erforderlich sein (siehe Abschnitt "11.4 Schalten ohmsch-induktiver Lasten").

11.2 AUSGANGSSCHALTUNG



11.3 VERDRAHTUNGSSCHEMA

Ein weiteres Verdrahtungsschema ist im Abschnitt "7. Spannungsversorgung" abgebildet.

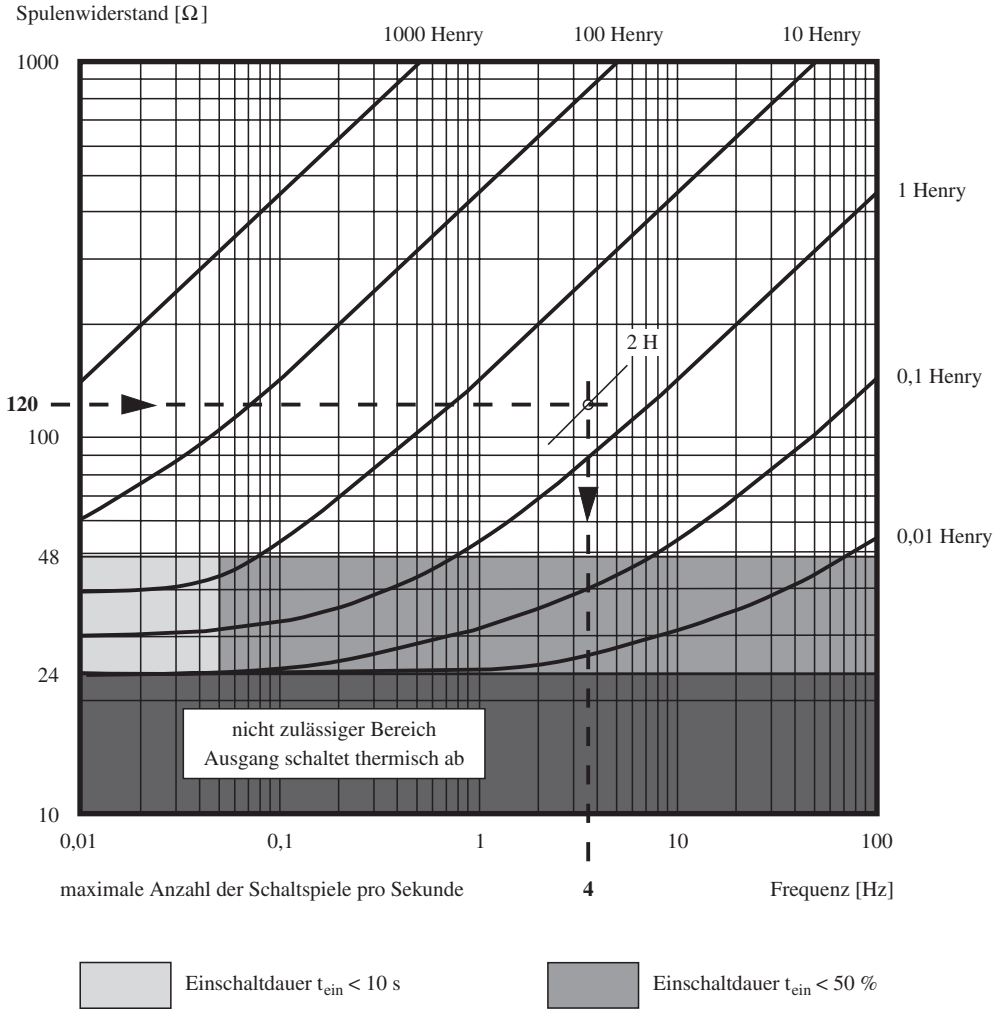


U ... 24 VDC / 1 A + I_A

I_A ... Summenstrom aller Ausgänge

11.4 SCHALTEN OHMSCH-INDUKTIVER LASTEN

Beim Schalten ohmsch-induktiver Lasten muß auf den Spulenwiderstand und auf die Spuleninduktivität geachtet werden.
Das unten abgebildete Diagramm zeigt die maximale Anzahl der Schaltspiele pro Sekunde im Zusammenhang von Spuleninduktivität und Spulenwiderstand. Es wird davon ausgegangen, daß alle Ausgänge gleichzeitig geschaltet werden (0,5 A bei 24 V).



Wenn der Spulenwiderstand kleiner 24 Ω ist, spricht der Thermoschutz der Digitalausgänge an und schaltet die Ausgänge ab.

Bei einem Spulenwiderstand zwischen 24 Ω und 48 Ω muß auf die Einschaltdauer geachtet werden. Je nach Schaltfrequenz darf sie nicht länger als 10 s bzw. nicht länger als die Ausschaltzeit sein. Ein Dauerbetrieb ist in diesem Bereich nicht möglich!
Wenn z. B. pro Sekunde ein Schaltspiel ausgeführt wird, darf die Einschaltdauer im Durchschnitt nicht länger als 0,5 s sein.

Beispiel Mit welcher maximalen Schaltfrequenz kann ein Magnetventil mit folgenden Spulendaten geschaltet werden?

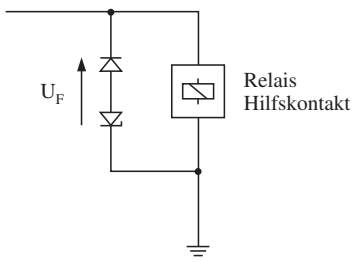
Induktivität:	2 H
Spulenwiderstand:	120 Ω

Man sucht den Spulenwiderstand auf der Y-Achse und folgt der horizontalen Linie nach rechts bis zum Schnittpunkt mit der gedachten Parameterkurve für 2 H (siehe Diagramm). Im Schnittpunkt folgt man der entsprechenden vertikalen Linie zur X-Achse und liest die maximale Schaltfrequenz ab (Achtung auf die logarithmischen Maßstäbe). Das Magnetventil kann mit einer maximalen Schaltfrequenz von 4 Hz betrieben werden.

Der Betrieb mit einer höheren Schaltfrequenz ist nicht zulässig und kann zur thermischen Abschaltung des Ausgangs führen.
Wenn eine höhere Schaltfrequenz erforderlich ist, muß eine externe Schutzbeschaltung angebracht werden. Die meisten Hersteller von Schützen und Magnetventilen bieten Schutzbeschaltungen für das jeweilige Element an.

Die Freilaufspannung U_F muß ≤ 8 V sein.

Beispiel für eine Schutzbeschaltung



12. ANALOGEINGÄNGE

Mit analogen Eingängen werden Meßwerte (Ströme, Spannungen, Widerstände oder Temperatur) in Zahlenwerte umgewandelt, die in der SPS verarbeitet werden können.

Die B&R Kompaktsteuerung ist mit zwei Analogeingängen ausgestattet. Für jeden Kanal kann softwaremäßig eines der folgenden Signale selektiert werden. Die Auswahl erfolgt durch Angabe einer Moduszahl (0 - 9).

Meßgröße	Modus	Eingangssignal	Auflösung bei Softwarefilter mit			Wert
			10 Hz / 50 Hz ±14 Bit	250 Hz ±12 Bit	1 kHz ±8 Bit	
Spannung	0	±10 V	±0,610 mV	±2,441 mV	±39,06 mV	±32767
	1	±2,5 V	±0,152 mV	±0,610 mV	±9,76 mV	±32767
Strom	2	0 - 20 mA	1,221 µA	4,883 µA	78,12 µA	0 - 32767

Meßgröße	Modus	Eingangssignal	Auflösung	Wert
Temperatur	3	KTY10 (-50 °C bis +150 °C)	0,01 °C	-5000 bis +15000
	4	KTY10 (-58 F bis +302 F)	0,01 F	-5800 bis +30200
reserviert	5	--	--	
	6	--	--	
	7	--	--	
Widerstand	8	Widerstandsmessung (0 - 65000 Ω) [1]	1 Ω	0 - 65000
	9	Widerstandsmessung (0 - 4500 Ω) [1]	0,1 Ω	0 - 45000

[1] Widerstandsmessung

Die Widerstandsmessung kann nur mit dem Standardfunktionsblock AINJ durchgeführt werden (siehe Abschnitt "12.7 AINJ - Analogeingang Kompaktsteuerung"). Die Auflösung Ω/Bit ist den Diagrammen auf den folgenden Seiten zu entnehmen.

Temperatur

Temperaturmessung mit Fühlern wie NTC, PTC, PT1000 usw. ist über Widerstandsmessung realisierbar (siehe Abschnitt "12.7 AINJ - Analogeingang Kompaktsteuerung").

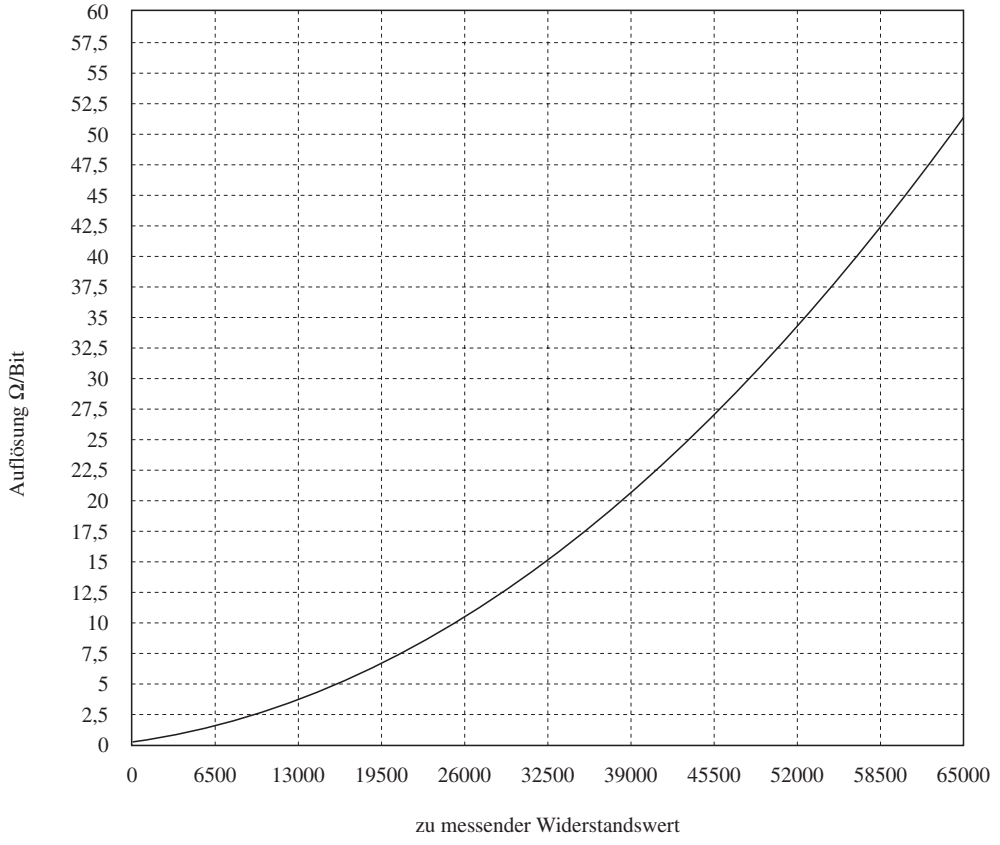
Utilities: ALIN, TLIN für nicht lineare Kennlinien
SCAL, TSCL für lineare Kennlinien

Diagramme für Widerstandsmessung

Aus den folgenden Diagrammen kann die Auflösung Ω/Bit bei einem bestimmten Widerstandswert abgelesen werden.

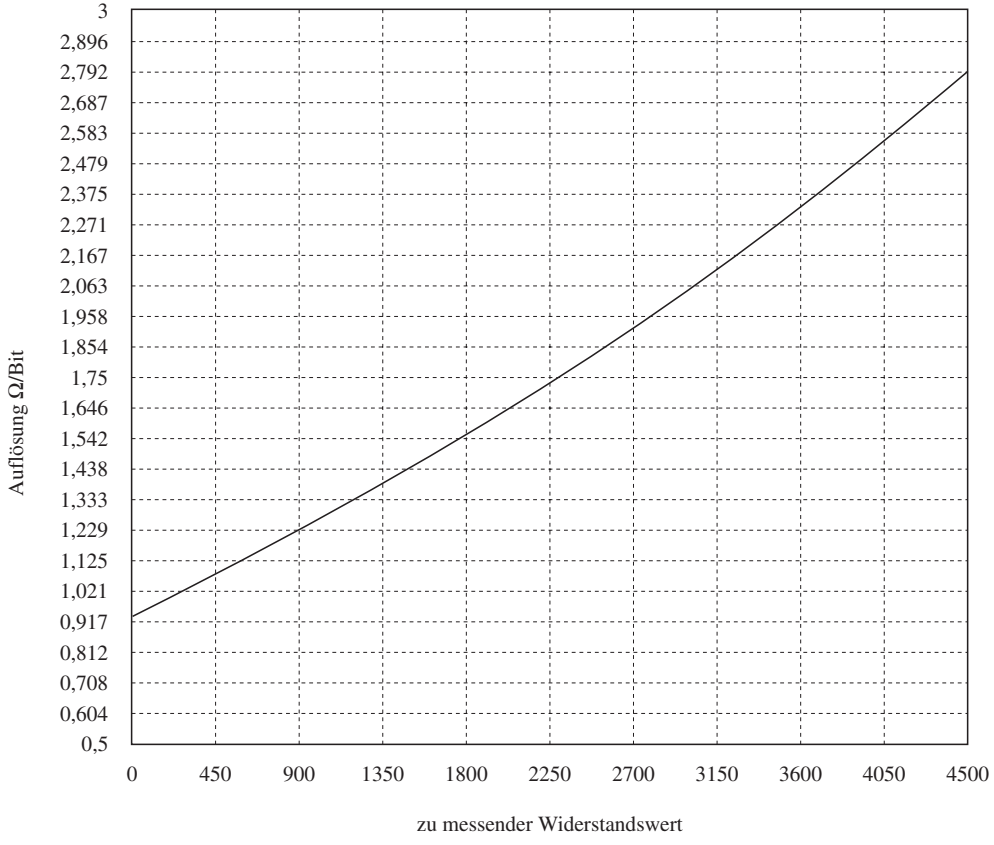
1) Widerstandsmessung 0 - 65000 Ω:

Modus: 8



2) Widerstandsmessung 0 - 4500 Ω:

Modus: 9



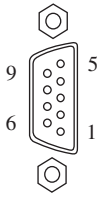
12.1 TECHNISCHE DATEN

Anzahl	2
galvanische Trennung	Differenzeingänge, nicht galvanisch getrennt
Gleichtaktbereich	±12 V
Eingangswiderstand	min. 10 MΩ
Spannungsfestigkeit	±30 V max.
Auflösung A/D Wandler	16 Bit
Adressierung	über Betriebssystem
Genauigkeit bei 25 °C	softwaremäßige Kompensation auf ±0,01 % (10 Hz Notch)
Temperaturdrift	±10 V Bereich: ±0,03 % / °C ±0,9 LSB / °C [1] ±2,5 V Bereich: ±0,02 % / °C ±3,5 LSB / °C
Hardwarefilter Eckfrequenz Steilheit Sprungantwort	 ca. 110 Hz ca. 20 dB/Dec. 63 % in 1,4 ms
Softwarefilter Eckfrequenz (1. Notch)	 10 Hz, 50 Hz, 250 Hz, 1 kHz, softwaremäßig umschaltbar
Wandlungszeiten	302 ms (10 Hz), 62 ms (50 Hz), 16,2 ms (250 Hz), 4,1 ms (1 kHz)
Kalibrierzeit	902 ms (10 Hz), 183 ms (50 Hz), 48 ms (250 Hz), 11,7 ms (1 kHz)
Dämpfung 1. Notch	>100 dB
Betriebsmodus	Trigger (kalibriert wird immer automatisch beim Wechsel der Notchfrequenz und nach einem Reset. Nach einem Reset ist die 50 Hz Notchfrequenz voreingestellt)
Eingangsspannung	±10 V / ±2,5 V softwaremäßig umschaltbar
Genauigkeit Spannungseingang	ca. ±14 Bit (10 Hz und 50 Hz), ca. ±12 Bit (250 Hz), ca. ±8 Bit (1 kHz)
Strommessung	es muß ein interner Shunt (124 Ω) beschaltet werden
Genauigkeit für 0 - 20 mA	ca. 14 Bit (10 Hz und 50 Hz), ca. 12 Bit (250 Hz), ca. 8 Bit (1 kHz)
KTY10 Temperatursensor Meßbereich Linearisation Auflösung	 -50 bis +150 °C hardwaremäßig auf ±0,3 °C im Bereich von -10 bis +110 °C 0,01 °C

[1] LSB ... Least Significant Bit

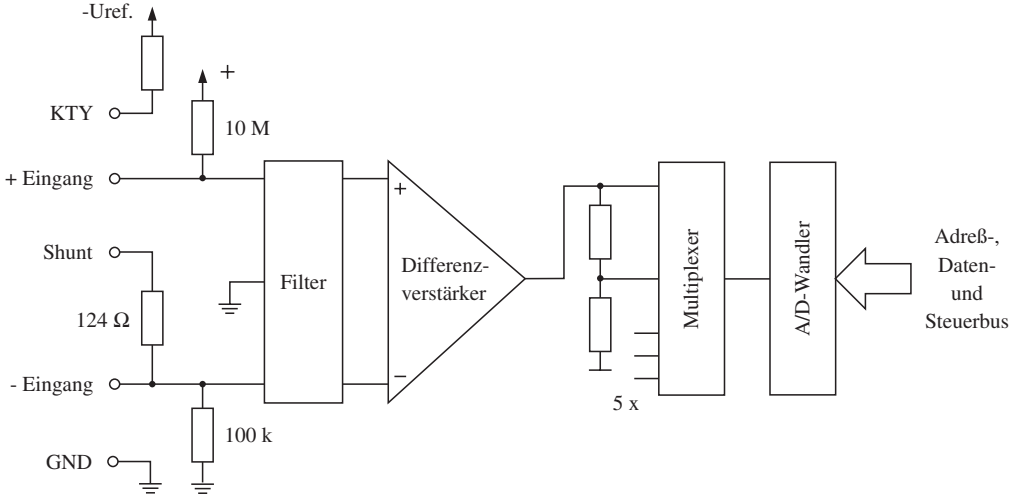
12.2 PINBELEGUNG

9polige DSUB-Buchse



Pin	Belegung
1	+ Eingang 1
2	- Eingang 1
3	GND
4	- Eingang 2
5	+ Eingang 2
6	KTY 1
7	Shunt 1
8	Shunt 2
9	KTY 2

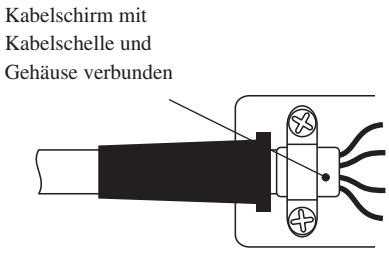
12.3 EINGANGSSCHALTUNG



12.4 SCHIRM ERDEN

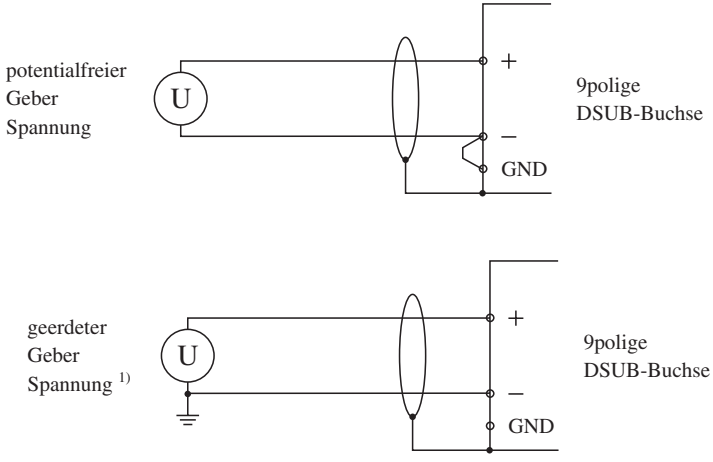
Vollmetallschrauben oder metallisierte Steckergehäuse stellen die einfachste und wirkungsvollste Lösung zur Ableitung der Störungen vom Kabelschirm zum Gehäuse der Kompaktsteuerung dar.

Der Schirm ist direkt auf das Steckergehäuse zu legen.



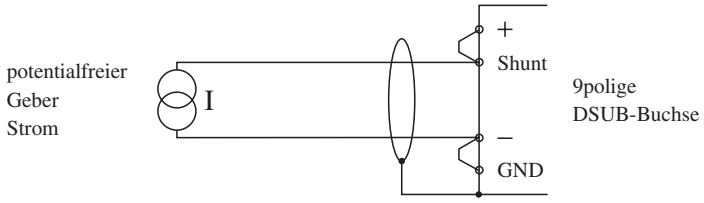
12.5 VERDRAHTUNGSSCHEMA

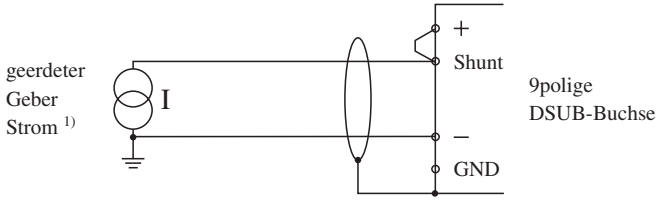
Spannungsgeber



1) Bei dieser Beschaltung ist keine sichere Leitungsbrucherkennung möglich.

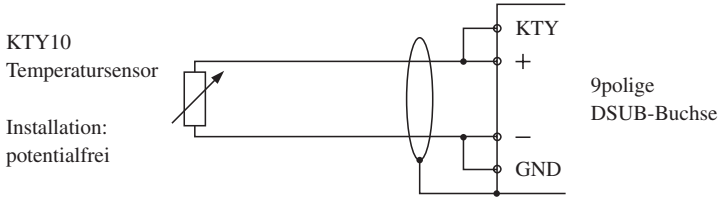
Stromgeber





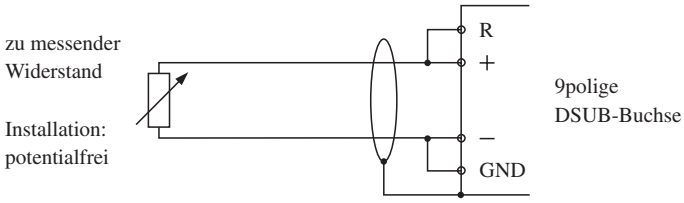
KTY10 Fühler

Die KTY Fühler sind Halbleiter Temperaturfühler. Die KTY10 Serie hat bei 25 °C einen Nennwiderstand von 2000 Ω (z. B. KTY10-6, KTY11-6, KTY13-6, KTY16-6 und KTY19-6).



1) Bei dieser Beschaltung ist keine sichere Leitungsbrucherkennung möglich.

Widerstandsmessung mit NTC, PTC, PT1000 usw.

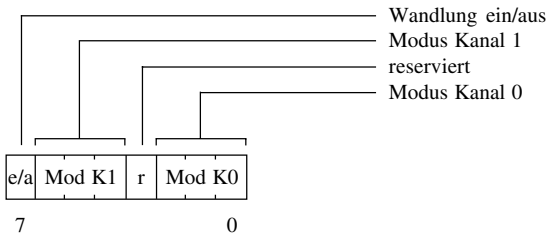


12.6 SOFTWAREMÄSSIGE BEDIENUNG

Die Initialisierung der Analogeingänge wird über die 8 Bit-Speicher C 1000 - C 1003 durchgeführt. Der Analogwert wird entweder in der END Routine oder durch einen AWL Aufruf gewandelt. Die gewandelten Werte werden in den Speichern C 1004 - C 1007 abgelegt.

Die Initialisierung kann entweder durch Befehle in der AWL oder mit Hilfe des Funktionsblocks AINJ (siehe Abschnitt 12.7) durchgeführt werden.

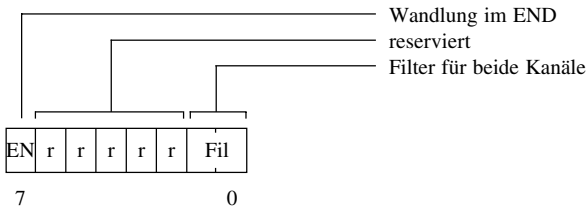
8 Bit-Speicher C 1000



Wandlung ein/aus	Schaltet die Analogwandlung ein/aus. Wenn sie ausgeschaltet ist, enthalten die Speicher C 1004 - C 1007 den Wert 0. 0 ... Wandlung aus 1 ... Wandlung ein
Modus Kanal 1	Mit den Bits 4 - 6 wird der Modus für Kanal 1 eingestellt.
reserviert	Für spätere Anwendung reserviert.
Modus Kanal 0	Mit den Bits 0 - 2 wird der Modus für Kanal 0 eingestellt. Es können die gleichen Modi wie für Kanal 1 eingestellt werden.

Modus	Signal
0	±10 V
1	±2,5 V
2	0 - 20 mA
3	KTY10 (Temperatur in °C)
4	KTY10 (Temperatur in F)

8 Bit-Speicher C 1001



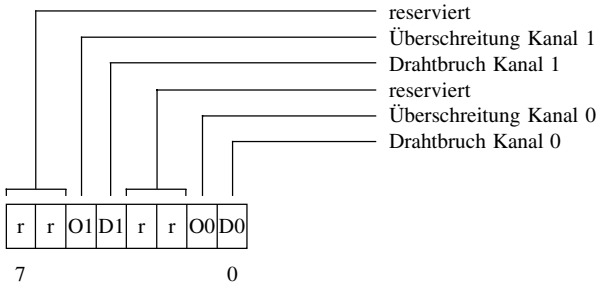
Wandlung im END	Definiert, ob die Wandlung während der Ausführung der END Routine durchgeführt werden soll. Wenn die Wandlung nicht in der END Routine durchgeführt wird, muß die Analogwandelroutine über einen AWL Aufruf gestartet werden (siehe Analogumwandlung). 0 ... Wandlung in der END Routine 1 ... Wandlung über AWL Aufruf
reserviert	Für spätere Anwendungen reserviert.
Filter für beide Kanäle	Definiert den Filter für beide Kanäle.

Filter	Frequenz	Wandelzeit
0	10 Hz	302 ms
1	50 Hz	62 ms
2	250 Hz	16,2 ms
3	1000 Hz	4,1 ms

8 Bit-Speicher C 1002

Wandelfrequenz Kanal 1	Definiert die Wandelfrequenz von Kanal 1 (0 - 255). 0 ... Kanal 1 wird nie gewandelt 1 ... Kanal 1 wird immer nach Kanal 0 gewandelt 2 ... Kanal 1 wird nach zweimal Kanal 0 gewandelt : ... : : ... : 255 ... Kanal 1 wird nach 255mal Kanal 0 gewandelt

8 Bit-Speicher C 1003



reserviert	Für spätere Anwendungen reserviert.
Überschreitung Kanal 1	1 ... Bereichsüberschreitung Kanal 1
Drahtbruch Kanal 1	1 ... Drahtbruch Kanal 1
reserviert	Für spätere Anwendungen reserviert.
Überschreitung Kanal 0	1 ... Bereichsüberschreitung Kanal 0
Drahtbruch Kanal 0	1 ... Drahtbruch Kanal 0

8 Bit-Speicher C 1004 - C 1007

Zielspeicher

In den 8 Bit-Speichern C 1004 - C 1007 werden die gewandelten Werte abgelegt.

Speicher	Kanal
C 1004&	0
C 1006&	1

Wertebereich:

Signal	Wert
±10 V	-32768 bis +32767
±2,5 V	-32768 bis +32767
0 - 20 mA	0 bis 32767
-50,00 bis +150,00 °C	-5000 bis +15000
-58,00 bis +302,00 F	-5800 bis +30200

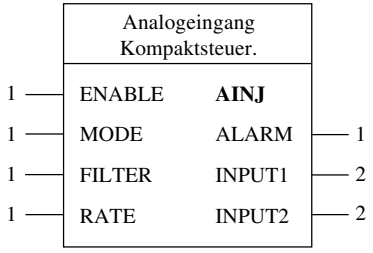
12.7 AINJ - ANALOGEINGANG KOMPAKTSTEUERUNG

Der Funktionsblock wird für die Initialisierung der Analogeingänge verwendet. Die gewandelten Werte und die Alarmbits für Bereichsüberschreitung und Drahtbruch werden auf Ausgänge gelegt.

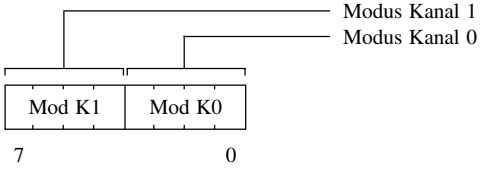
HINWEIS: Die Analogwandlung wird immer in der END Routine durchgeführt.

AINJ

Funktionsblocklänge: 84 Zeilen
First Scan: Nein



Ein-/Ausgänge	Beschreibung	Adreßtypen
ENABLE	Wenn der ENABLE Eingang 1 ist, wird der Funktionsblock ausgeführt.	C, M, #
MODE	Modus der Kanäle einstellen.	C, #



Modus	Signal
0	±10 V
1	±2,5 V
2	0 - 20 mA
3	KTY10 (Temperatur in °C)
4	KTY10 (Temperatur in F)
5	reserviert
6	reserviert
7	reserviert
8	Widerstandsmessung (0 - 65000 Ω) [1]
9	Widerstandsmessung (0 - 4500 Ω) [1]

[1] Ab Version 1.30

<table>
<tr><th>Ein-/Ausgänge</th><th>Beschreibung</th><th>Adreßtypen</th></tr>
<tr><td>FILTER</td><td>Definiert den Filter für beide Kanäle.
<table>
<tr><th>Filter</th><th>Frequenz</th><th>Wandelzeit</th><th>Auflösung</th></tr>
<tr><td>0</td><td>10 Hz</td><td>302 ms</td><td>±14 Bit</td></tr>
<tr><td>1</td><td>50 Hz</td><td>62 ms</td><td>±14 Bit</td></tr>
<tr><td>2</td><td>250 Hz</td><td>16,2 ms</td><td>±12 Bit</td></tr>
<tr><td>3</td><td>1000 Hz</td><td>4,1 ms</td><td>±8 Bit</td></tr>
</table></td><td>C, #</td></tr>
<tr><td>RATE</td><td>Definiert die Wandelfrequenz von Kanal 1 (0 - 255).

0 ... Kanal 1 wird nie gewandelt
1 ... Kanal 1 wird immer nach Kanal 0 gewandelt
2 ... Kanal 1 wird nach zweimal Kanal 0 gewandelt
: ... :
: ... :
255 ... Kanal 1 wird nach 255mal Kanal 0 gewandelt</td><td>C, #</td></tr>
<tr><td>ALARM</td><td>zeigt einen Drahtbruch oder eine Bereichsüberschreitung der Kanäle an.

Überschreitung Kanal 1
Drahtbruch Kanal 1
Überschreitung Kanal 0
Drahtbruch Kanal 0

| | | O1 | D1 | | | O0 | D0 |
7 0</td><td>C</td></tr>
<tr><td>INPUT0</td><td>enthält den gewandelten Wert von Kanal 0.

Wertebereich:
<table>
<tr><th>Signal</th><th>Wert</th></tr>
<tr><td>±10 V</td><td>-32768 bis +32767</td></tr>
<tr><td>±2,5 V</td><td>-32768 bis +32767</td></tr>
<tr><td>0 - 20 mA</td><td>0 bis 32767</td></tr>
<tr><td>-50,00 bis +150,00 °C</td><td>-5000 bis +15000</td></tr>
<tr><td>-58,00 bis +302,00 F</td><td>-5800 bis +30200</td></tr>
<tr><td>0 - 65000 Ω</td><td>0 bis 65000</td></tr>
<tr><td>0 - 4500 Ω</td><td>0 bis 45000</td></tr>
</table></td><td>C</td></tr>
</table>

Ein-/Ausgänge	Beschreibung	Adreßtypen
FILTER	enthält den gewandelten Wert von Kanal 1. Wertebereich:	C

Signal	Wert
±10 V	-32768 bis +32767
±2,5 V	-32768 bis +32767
0 - 20 mA	0 bis 32767
-50,00 bis +150,00 °C	-5000 bis +15000
-58,00 bis +302,00 F	-5800 bis +30200
0 bis 65000 Ω	0 bis 65000
0 bis 4500 Ω	0 bis 45000

12.8 ANALOGUMWANDLUNG

HINWEIS: Im ersten Programmdurchlauf sind die Werte in den Speichern C 1004 - C 1007 null.

Ist Bit 7 von C 1000 (Modusregister) nicht gesetzt, erfolgt keine Analogwandlung (C 1004 - C 1007 = 0).

1) Wandlung in der END Routine

Software	Wandlung
FUB	Die Wandlung wird immer in der END Routine durchgeführt.
AWL	Wenn Bit 7 der 8 Bit-Speicherstelle C 1001 gesetzt ist, wird die Wandlung in der END Routine durchgeführt.

2) Direkter Sprung auf Pointer im Betriebssystem

Software	Wandlung
FUB	Dieser Aufruf ist bei Verwendung des Funktionsblocks AINJ nicht möglich.

AWL	Anstelle der Wandlung in der END Routine kann man auch direkt vom Programm auf einen Pointer im Betriebssystem springen, welcher auf die Analogwandelroutine verweist (Pointer: $C10E).
	
	Auswertung des gewandelten Kanals: Carry = 0 ... Wandlung beendet
	Akku B ... gewandelter Kanal
	
	Beispiel
	`LD # $C10E Adresse der Betriebssystemroutine`
	`DXR`
	`SPU I 000 Aufruf der Betriebssystemroutine`
	`J< BUSY`
	`LRK C 1004 Auswertung des gewandelten Kanals`
	`SLB Akku B: gewandelte Kanalnummer (0/1)`
	`B+R`
	`LD I 000`
	`... ...`
	`... ...`
	`BUSY RET`

3) Kanalrefreshzeit (bei Wandlung in der END Routine)

a) Programmdurchlaufzeit ist kleiner als die Kanalwandelzeit (abhängig vom Filter)

Refreshzeit_{max} = Anzahl der Kanäle * (Kanalwandelzeit + Programmdurchlaufzeit)

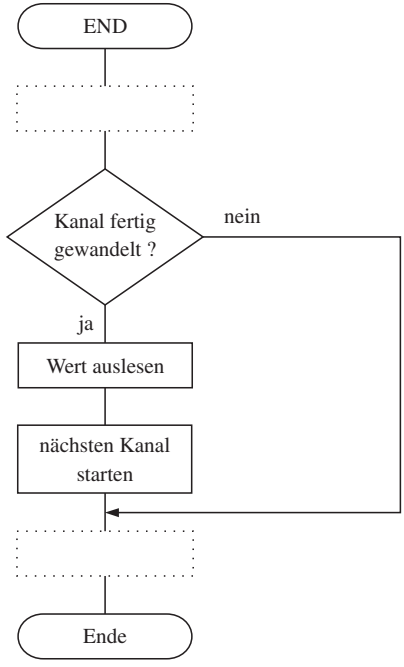
b) Programmdurchlaufzeit ist größer als die Kanalwandelzeit (abhängig vom Filter)

Refreshzeit = Anzahl der Kanäle * Programmdurchlaufzeit (pro Durchlauf ein Kanal)

4) END Routine

Wenn die Analogwandlung in der END Routine durchgeführt wird, kontrolliert die Routine ob die Wandlung eines Kanals abgeschlossen ist. Wenn "ja", wird der Wert ausgelesen und die Wandlung des nächsten Kanals gestartet.
Wenn "nein", wird der Teil für die Analogwandlung übersprungen.

Flußdiagramm:



13. ANALOGAUSGÄNGE

Analogausgänge werden verwendet, um interne Zahlenwerte der SPS in Ströme und Spannungen zu konvertieren.

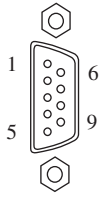
13.1 TECHNISCHE DATEN

Anzahl	2
Ausführung	kurzschlußfest
Adressierung	über Betriebssystem
Spannungsausgang	±10 V / 10 mA
Stromausgang	0 - 20 mA, Bürde 50 Ω (400 Ω max.)
Auflösung D/A Wandler	12 Bit
Offset bei 25 °C	softwaremäßige Kompensation auf ±1 LSB [1)]
Offsetdrift	±0,02 % of full scale / °C (±0,8 LSB/°C) [1)]
Gainfehler bei 25 °C	softwaremäßige Kompensation auf ±0,5 %
Gaindrift	±0,02 %/°C
Gainfehler Bürde	0,01 %/Ω
Linearität	±1 LSB [1)]
Einschwingzeitkonstante	< 1 ms
Maximaler Ausgangspegel während Resetzustand (bzw. vor Initialisierung)	±50 mV oder -0,3 mA

[1)] LSB ... Least Significant Bit

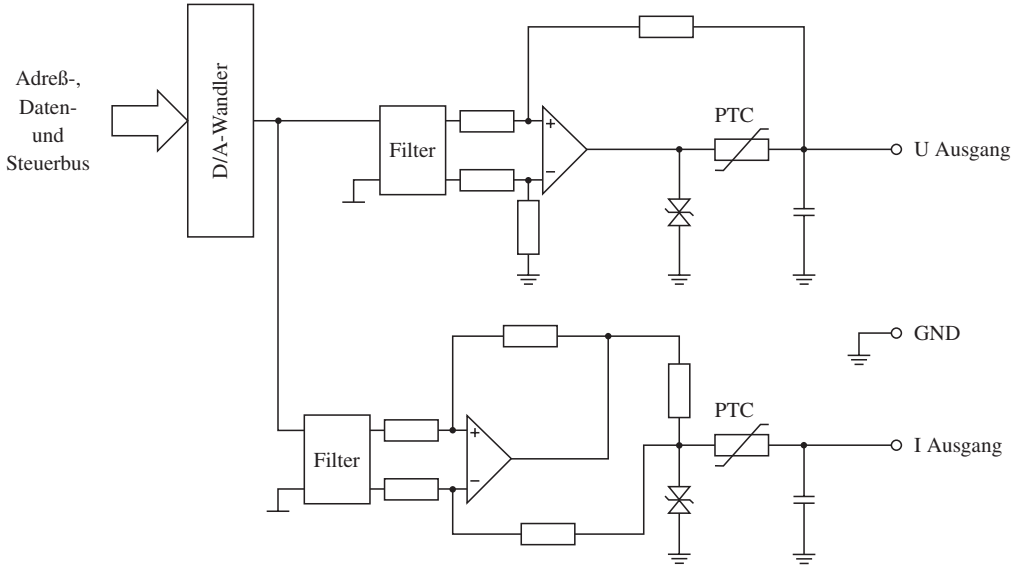
13.2 PINBELEGUNG

9polige DSUB-Buchse



Pin	Belegung
1	U Ausgang 1
2	I Ausgang 1
3	n.c.
4	U Ausgang 2
5	I Ausgang 2
6	GND
7	n.c.
8	n.c.
9	GND

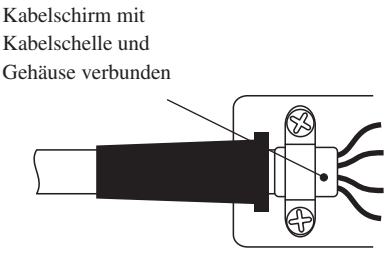
13.3 AUSGANGSSCHALTUNG



13.4 SCHIRM ERDEN

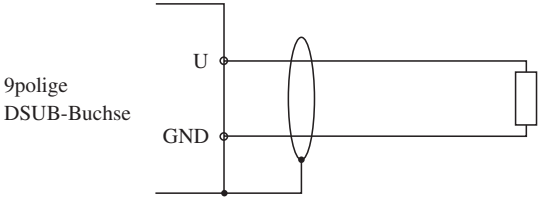
Vollmetallschrauben oder metallisierte Steckergehäuse stellen die einfachste und wirkungsvollste Lösung zur Ableitung der Störungen vom Kabelschirm zum Gehäuse der Kompaktsteuerung dar.

Der Schirm ist direkt auf das Steckergehäuse zu legen.

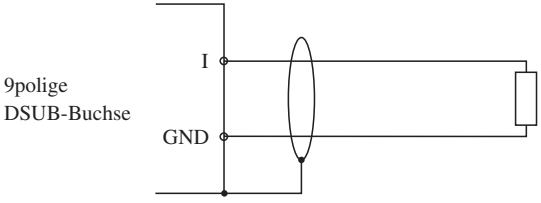


13.5 VERDRAHTUNGSSCHEMA

Spannungsausgang



Stromausgang

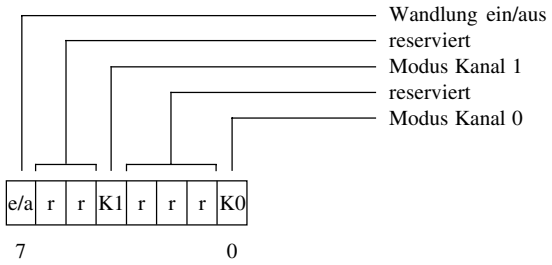


13.6 SOFTWAREMÄSSIGE BEDIENUNG

Die Initialisierung der Analogausgänge wird über den 8 Bit-Speicher C 1009 durchgeführt. Die zu wandelnden Werte sind in den Speichern C 1010 - C 1013 gespeichert.

Die Initialisierung kann entweder durch Befehle in der AWL oder mit Hilfe des Funktionsblocks AOTF (siehe Abschnitt 13.7) durchgeführt werden.

8 Bit-Speicher C 1009



Wandlung ein/aus	Schaltet die Analogwandlung ein/aus. Wenn sie ausgeschaltet ist, sind die Analogausgänge auf 0 (der Inhalt der Speicher C 1010 - C 1013 bleibt erhalten). 0 ... Wandlung aus 1 ... Wandlung ein
reserviert	Für spätere Anwendungen reserviert.
Modus Kanal 1	Mit Bit 4 wird der Modus für Kanal 1 eingestellt. **Modus** / **Signal** 0 / ±10 V 1 / 0 - 20 mA
reserviert	Für spätere Anwendungen reserviert.
Modus Kanal 0	Mit Bit 0 wird der Modus für Kanal 0 eingestellt. **Modus** / **Signal** 0 / ±10 V 1 / 0 - 20 mA

8 Bit-Speicher C 1010 - C 1013

Quellspeicher

In den 8 Bit-Speichern C 1010 - C 1013 sind die zu wandelnden Werte abgelegt.

Speicher	Kanal
C 1010&	0
C 1012&	1

Wertebereich:

Signal	Wert
±10 V	-32768 bis +32767
0 - 20 mA	0 bis 32767

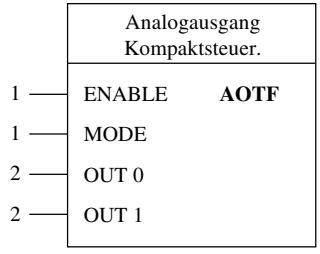
13.7 AOTF - ANALOGAUSGANG KOMPAKTSTEUERUNG

Der Funktionsblock wird für die Initialisierung der Analogausgänge verwendet. Die zu wandelnden Digitalwerte werden dem Funktionsblock über die Eingänge OUT 0 und OUT 1 übergeben.

HINWEIS: Die Analogwandlung erfolgt nur bei Wertänderung. Der Wertvergleich erfolgt im Systeminterrupt. Nach einem Power-on bzw. nach einem Reset (SW oder HW) sind die Ausgänge definiert 0.

AOTF

Funktionsblocklänge: 14 Zeilen
First Scan: Nein



Ein-/Ausgänge	Beschreibung	Adreßtypen
ENABLE	Wenn der ENABLE Eingang 1 ist, wird der Funktionsblock ausgeführt.	C, M, #
MODE	Modus der Kanäle einstellen.	C, #

Kanal	Bit
0	0
1	4

Modus	Signal
0	±10 V
1	0 - 20 mA

<table>
<tr><th>Ein-/Ausgänge</th><th>Beschreibung</th><th>Adreßtypen</th></tr>
<tr><td>OUT0</td><td>enthält den zu wandelnden Wert von Kanal 0.

Wertebereich:
<table>
<tr><th>Signal</th><th>Wert</th></tr>
<tr><td>±10 V</td><td>-32768 bis +32767</td></tr>
<tr><td>0 - 20 mA</td><td>0 bis 32767</td></tr>
</table></td><td>C</td></tr>
<tr><td>OUT1</td><td>enthält den zu wandelnden Wert von Kanal 1.

Wertebereich:
<table>
<tr><th>Signal</th><th>Wert</th></tr>
<tr><td>±10 V</td><td>-32768 bis +32767</td></tr>
<tr><td>0 - 20 mA</td><td>0 bis 32767</td></tr>
</table></td><td>C</td></tr>
</table>

13.8 ANALOGWANDLUNG

Die Analogwandlung erfolgt nur bei Wertänderung. Nach einem Power-on bzw. nach einem Reset (SW oder HW) sind die Speicher C 1010 - C 1013 und somit auch die Ausgänge definiert 0.

1) Wertvergleich im Systeminterrupt

Der Wertvergleich erfolgt im Systeminterrupt. Es wird je Systeminterrupt nur ein Kanal überprüft, das heißt, daß die schnellste Updatezeit 20 ms pro Kanal beträgt.

2) Direkter Sprung auf Pointer im Betriebssystem

Zusätzlich zum Wertevergleich im Systeminterrupt kann man direkt vom Programm auf einen Pointer im Betriebssystem springen, der auf die Analogwandelroutine verweist (Pointer: $C111). Dadurch wird eine kürzere Updatezeit erreicht. Pro Aufruf wird ein Kanal überprüft.

Beispiel:

```
LD    # $C111     Adresse der Betriebssystemroutine
DXR
SPU   I 000       Aufruf der Betriebssystemroutine
```

3) Einen bestimmten Kanal überprüfen

Diese Methode wird verwendet, wenn nur ein Analogausgang benötigt wird. Wie in Punkt 2 wird auf einen Pointer im Betriebssystem gesprungen, der auf die Analogwandelroutine verweist (Pointer: $C120). Die Kanalnummer wird im Akku B übergeben.

Beispiel:

```
LD    # $C120     Adresse der Betriebssystemroutine
DXR
LB    Kanalnummer (0/1)
SPU   I 000       Aufruf der Betriebssystemroutine
```

14. ZÄHLEREINGÄNGE UND INTERRUPTEINGANG

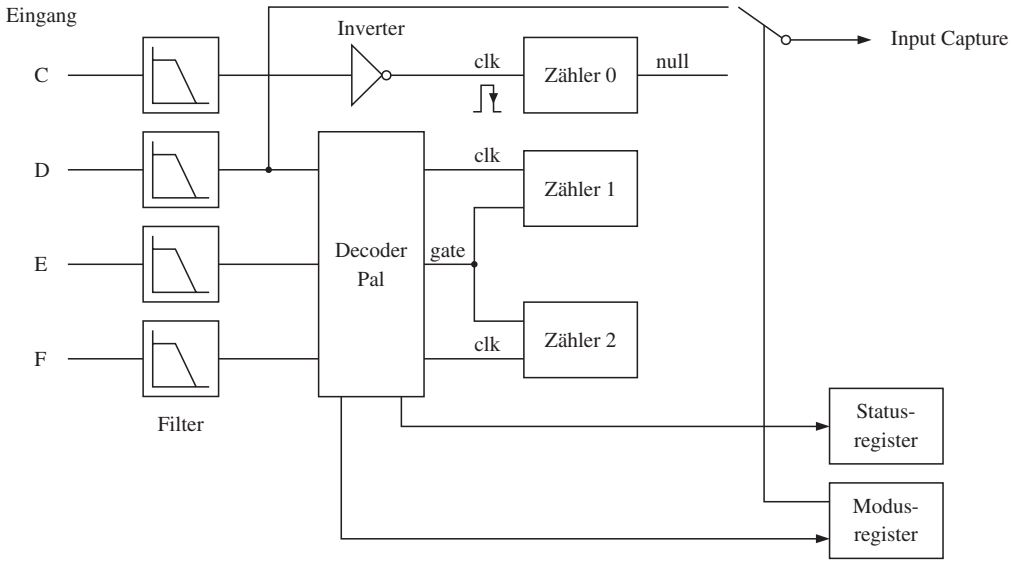
Die Kompaktsteuerung ist mit 16 Digitaleingängen ausgestattet. Die Eingänge C - F können auch für Sonderaufgaben verwendet werden:

- Ereigniszähler
- Interrupteingang
- Referenzeingang
- Inkrementalgeber (Kanäle A und B) für Positionieraufgaben

Die Software für die Bedienung der Eingänge ist auf der Standardsoftwarediskette für die Kompaktsteuerung gespeichert.

Im Anschluß werden die Funktionen der Eingänge einmal in graphischer Form (Übersicht) und einmal in tabellarischer Form beschrieben.

Übersicht der Funktionen

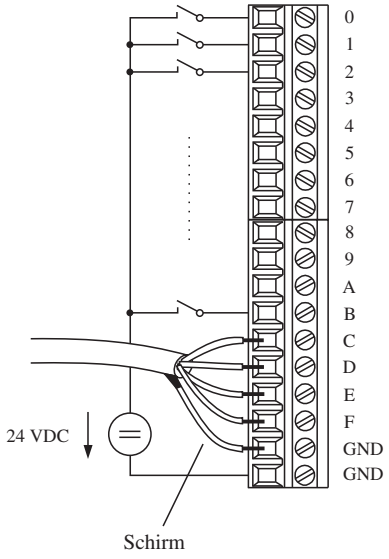


Funktionen der Eingänge

Eingang	Funktion
C	Dekrementiert mit jeder negativen Flanke den Zähler 0. Der Zähler 0 kann auf zwei verschiedene Arten verwendet werden. Die Definition erfolgt über den Funktionsblockeingang MODE oder im Modusregister (siehe Abschnitt "14.6.3 Modusregister"). 1) Ereigniszähler ... Zähler mit 4 Bytes (0 - 4 294 967 295) 2) HW-Zähler ... Zähler mit 2 Bytes (0 - 65535) Der Hardwarezähler kann für die Generierung eines Interrupts verwendet werden. Der Interrupt wird ausgelöst, wenn der Zähler den Wert 0 erreicht (siehe auch Abschnitt "14.5 Interruptmodus".
D	Wird als Referenzimpulseingang oder Interrupteingang verwendet. Die Definition erfolgt über den Funktionsblockeingang MODE oder im Modusregister (siehe Abschnitt "14.6.3 Modusregister"). 1) Referenzimpulseingang: Wird verwendet, um Zähler auf einen bestimmten Vorwahlwert zu setzen (minimale Impulsdauer: 50 µs). 2) Interrupteingang Die Verwendung von Eingang D als Interrupteingang ist im Abschnitt "14.5 Interruptmodus" beschrieben.
E	Dekrementiert den Zähler 1 oder wird als Eingang A bei Zweikanalzählung verwendet. Die Definition erfolgt über den Funktionsblockeingang MODE oder im Modusregister (siehe Abschnitt "14.6.3 Modusregister"). 1) Ereigniszähler ... Zähler mit 4 Bytes (0 - 4 294 967 295) 2) Eingang A Für Positionieraufgaben kann eine Zweikanalzählung mit Vierfachauswertung durchgeführt werden. An Eingang E wird der erste Kanal angeschlossen.
F	Dekrementiert den Zähler 2 oder wird als Eingang B bei Zweikanalzählung verwendet. Die Definition erfolgt über den Funktionsblockeingang MODE oder im Modusregister (siehe Abschnitt "14.6.3 Modusregister"). 1) Ereigniszähler ... Zähler mit 4 Bytes (0 - 4 294 967 295) 2) Eingang B Für Positionieraufgaben kann eine Zweikanalzählung mit Vierfachauswertung durchgeführt werden. An Eingang F wird der zweite Kanal angeschlossen.

14.1 VERDRAHTUNGSSCHEMA

Verdrahtungsschema mit bis zu zwölf Digitaleingängen, drei Zählereingängen und einem Interrupteingang:



14.2 BEDIENUNG DER ZÄHLER

Die Bedienung erfolgt entweder über den Funktionsblock CMDC oder CMDD oder über den Aufruf von Betriebssystemroutinen aus der AWL.
Es können aber auch beide Methoden miteinander gemischt werden.

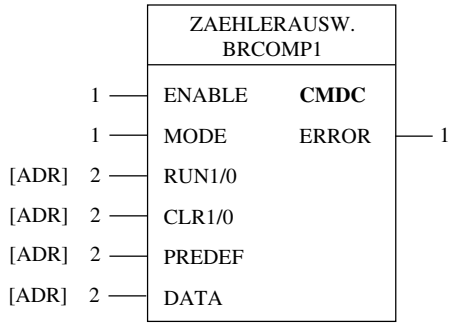
Die Funktionsblöcke sind auf der Standardsoftwarediskette für die Kompaktsteuerung gespeichert.

14.3 CMDC - ZÄHLERAUSWERTUNG (BRCOMP)

Bei Verwendung des Funktionsblockes werden Vorwahlwerte und neue Zählerstände im nächsten END (Programmende) übernommen bzw. aufgefrischt.

CMDC

Funktionsblocklänge: 120 Zeilen
First Scan: Ja



Ein-/Ausgänge	Beschreibung	Adreßtypen
ENABLE	Wenn der ENABLE Eingang 1 ist, wird der Funktionsblock ausgeführt.	C, M, #
MODE	Betriebsmodus der Zähler einstellen. Flanke für Input Capture Interrupt 0 ... negative Flanke 1 ... positive Flanke reserviert für spätere Anwendungen Interruptquelle für Input Capture Eingang 00 ... Interrupt gesperrt 01 ... Eingang D 10 ... Nullsignal Hardwarezähler 0 11 ... reserviert Zählermodus für Zähler 1 und 2 00 ... Ereigniszähler pos. Flanke 01 ... Ereigniszähler pos./ neg. Flanke 10 ... reserviert 11 ... Zweikanalzähler (Encoder) 7 6 5 4 3 2 1 0 Ereigniszähler 0 reagiert immer auf die positive Flanke. Diese Einstellung kann nicht geändert werden. Wenn das Nullsignal von Zähler 0 als Interruptquelle selektiert wird, muß Bit 7 auf die positive Flanke eingestellt werden. Anderenfalls wird kein Interrupt ausgelöst. Eine Änderung von Bit 7 wird nur übernommen, wenn sich der Status der Bits 0 - 3 ändert. Wenn der Zähler 0 als Interruptquelle verwendet wird, kann der Vorwahlwert für Zähler 0 nur zwischen 3 und 65535 liegen (Vorwahlwertübergabe siehe Abschnitt "14.6.8 Aufruf von Betriebssystemroutinen", Punkt 3).	C, #

Ein-/Ausgänge	Beschreibung	Adreßtypen
RUN1/0	An diesen Eingang wird die Startadresse eines 4 Bytes langen Speicherbereichs angeschlossen. Mit diesen Speichern kann jeder Zähler einzeln gestoppt und wieder freigegeben werden. Zuordnung der Speicher / Speicherzustände 1. Speicher ... Zähler 0 — 0 ... gestoppt 2. Speicher ... Zähler 1 — 1 ... weiterzählen 3. Speicher ... Zähler 2 4. Speicher ... Encoderposition Die Speicher können natürlich nur sinnvoll bedient werden, wenn auch der richtige Betriebsmodus eingestellt ist. Z. B. hat das Setzen der vierten Speicherstelle keinerlei Wirkung, wenn nicht der Zweikanalzählermodus selektiert ist.	[ADR] C, [ADR] M
CLR1/0	An diesen Eingang wird die Startadresse eines 4 Bytes langen Speicherbereichs angeschlossen. Mit diesen Speichern kann jeder Zähler einzeln auf den Vorwahlwert gesetzt werden. Zuordnung der Speicher / Speicherzustände 1. Speicher ... Zähler 0 — 0 ... weiterzählen 2. Speicher ... Zähler 1 — 1 ... auf Vorwahlwert setzen 3. Speicher ... Zähler 2 4. Speicher ... Encoderposition Die Speicher können natürlich nur sinnvoll bedient werden, wenn auch der richtige Betriebsmodus eingestellt ist. Z. B. hat das Setzen der vierten Speicherstelle keinerlei Wirkung, wenn nicht der Zweikanalzählermodus selektiert ist.	[ADR] C, [ADR] M
PREDEF	An diesen Eingang wird die Startadresse eines 16 Bytes langen 8 Bit-Speicherbereichs angeschlossen. Die Speicher enthalten die Vorwahlwerte der Zähler. Speicher 1 - 4 ... Vorwahlwert Zähler 0 Speicher 5 - 8 ... Vorwahlwert Zähler 1 Speicher 9 - 12 ... Vorwahlwert Zähler 2 Speicher 13 - 16 ... Vorwahlwert Encoderposition Die Speicher können natürlich nur sinnvoll bedient werden, wenn auch der richtige Betriebsmodus eingestellt ist. Z. B. hat das Beschreiben der Speicherstellen 13 - 16 keinerlei Wirkung, wenn nicht der Zweikanalzählermodus selektiert ist.	[ADR] C
DATA	An diesen Eingang wird die Startadresse eines 18 Bytes langen 8 Bit-Speicherbereichs angeschlossen. Die Speicher enthalten die Zählerstände der Zähler. Speicher 1 - 4 ... Zählerstand Zähler 0 Speicher 5 - 8 ... Zählerstand Zähler 1 Speicher 9 - 12 ... Zählerstand Zähler 2 Speicher 13 - 16 ... Zählerstand Encoderposition Speicher 17 - 18 ... Zählerstand Zähler 0 - Hardwarezähler Die Speicher können natürlich nur sinnvoll bedient werden, wenn auch der richtige Betriebsmodus eingestellt ist. Z. B. ist das Auswerten der Speicherstellen 13 - 16 nicht sinnvoll, wenn nicht der Zweikanalzählermodus selektiert ist.	[ADR] C
ERROR	Dieser Ausgang zeigt an, daß ein Eingang nicht oder fehlerhaft angeschlossen wurde.	C, M

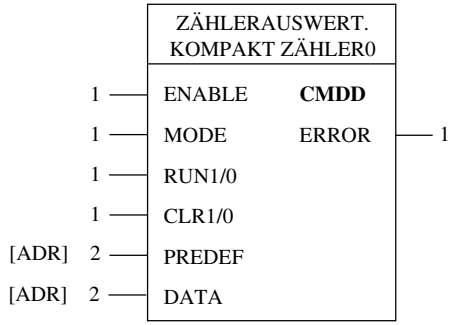
14.4 CMDD - ZÄHLER 0 AUSWERTUNG (BRCOMP)

Der Hardwarezähler kann nur vorgeladen werden, wenn der Zähler nicht als Interruptquelle ausgewählt ist. Nach dem Vorladen muß der Hardwarezähler als Interruptquelle definiert werden.
Der Istwert des Hardwarezählers ändert sich erst nach der nächsten Flanke am Zähleingang auf den Vorladewert.

Der Hardwarezähler kann nicht gestoppt werden.

CMDD

Funktionsblocklänge: 90 Zeilen
First Scan: Ja



Ein-/Ausgänge	Beschreibung	Adreßtypen
ENABLE	Wenn der ENABLE Eingang 1 ist, wird der Funktionsblock ausgeführt.	C, M, #
MODE	Betriebsmodus des Zählers 0 einstellen. Flanke für Input Capture Interrupt 0 ... negative Flanke 1 ... positive Flanke reserviert für spätere Anwendungen Interruptquelle für Input Capture Eingang 00 ... Interrupt gesperrt 01 ... Eingang D 10 ... Nullsignal Hardwarezähler 0 11 ... reserviert reserviert für spätere Anwendungen 7 6 5 4 3 2 1 0 Ereigniszähler 0 reagiert immer auf die positive Flanke. Diese Einstellung kann nicht geändert werden. Wenn das Nullsignal von Zähler 0 als Interruptquelle selektiert wird, muß Bit 7 auf die positive Flanke eingestellt werden. Anderenfalls wird kein Interrupt ausgelöst. Eine Änderung von Bit 7 wird nur übernommen, wenn sich der Status der Bits 0 - 3 ändert. Wenn der Zähler 0 als Interruptquelle verwendet wird, kann der Vorwahlwert für Zähler 0 nur zwischen 3 und 65535 liegen.	C, #

Ein-/Ausgänge	Beschreibung	Adreßtypen
RUN1/0	Zähler 0 stoppen bzw. freigeben. 0 ... stoppen 1 ... weiterzählen Der Speicher kann natürlich nur sinnvoll verwendet werden, wenn auch der richtige Betriebsmodus eingestellt ist. Z. B. hat das Setzen des Speichers keinerlei Wirkung, wenn als Interruptquelle Hardwarezähler 0 selektiert ist.	C, M
CLR1/0	Zähler auf den Vorwahlwert setzen. 0 ... weiterzählen 1 ... auf Vorwahlwert setzen	C, M
PREDEF	An diesen Eingang wird die Startadresse eines 4 Bytes langen 8 Bit-Speicherbereichs angeschlossen. Die Speicher enthalten den Vorwahlwert des Zählers 0. Wenn der Hardwarezähler 0 als Interruptquelle verwendet wird, kann der Vorwahlwert für den Hardwarezähler 0 nur zwischen 3 und 65535 liegen. Offset 0 ... HHB Vorwahlwert Zähler 0 1 ... HB Vorwahlwert Zähler 0 Offset 2 ... MB Vorwahlwert Zähler 0 bzw. Hardwarezähler 3 ... LB Vorwahlwert Zähler 0 bzw. Hardwarezähler	[ADR] C
DATA	An diesen Eingang wird die Startadresse eines 6 Bytes langen 8 Bit-Speicherbereichs angeschlossen. Die Speicher enthalten die Zählerstände der Zähler. Offset 0 ... HHB Zählerstand Zähler 0 1 ... HB Zählerstand Zähler 0 2 ... MB Zählerstand Zähler 0 3 ... LB Zählerstand Zähler 0 Offset 4 ... HB Zählerstand Hardwarezähler 5 ... LB Zählerstand Hardwarezähler	[ADR] C
ERROR	Dieser Ausgang zeigt an, daß ein Eingang nicht oder fehlerhaft angeschlossen wurde.	C, M

14.5 INTERRUPTMODUS

Im Modusregister bzw. über den Funktionsblockeingang MODE kann eine Interruptquelle für die Generierung eines Interrupts definiert werden.
Die Interrupts werden erst beim Ausführen des nächsten END (Programmende) freigegeben bzw. gesperrt und nicht nach der Definition des Modusregisters oder während der Exekution des Funktionsblockes.
Durch Aufruf der Betriebssystemroutine wird der Interrupt sofort freigegeben oder gesperrt.

1. Eingang D wird als Interruptquelle verwendet

Je nach Definition löst die positive oder negative Flanke an diesem Eingang einen Interrupt aus. Der Interrupt wiederholt sich bei jedem entsprechenden Ereignis am Eingang D.

Beim Funktionsblock CMDC erfolgt die Flankeneinstellung über den Eingang MODE. Beim Aufruf einer Betriebssystemroutine aus der AWL erfolgt die Definition im Initialisierungsregister $2122.

2. Hardwarezähler 0 wird als Interruptquelle verwendet

Der Interrupt wird ausgelöst, wenn der Hardwarezähler 0 vom eingestellten Vorwahlwert auf 0 gezählt hat. Da der Hardwarezähler nur über einen 16 Bit-Bereich verfügt, werden auch nur Vorwahlwerte im 2 Byte-Zahlenbereich übernommen. Gültiger Wertebereich ist 3 bis 65535 (Vorwahlwertübergabe siehe Abschnitt "14.6.8 Aufruf von Betriebssystemroutinen", Punkt 3).

Der Hardwarezähler 0 läuft nach dem Nulldurchgang auf 65535 über. Ausgehend von diesem Wert wird beim nächsten Nulldurchgang ein Interrupt ausgelöst.
Wenn von einem bestimmten Wert ausgegangen werden soll, muß der Hardwarezähler 0 nach jedem Nulldurchgang mit dem Vorwahlwert geladen werden.

Interruptroutinen

Durch einmaliges Beschreiben (z. B. First Scan) wird den 8 Bit-Speichern C 0968 und C 0969 die Adresse des Interruptprogramms zugewiesen.

```
        ...
        LRL   INTR
        =R    C 0968
        ...
```

Die Interruptroutine INTR wird als Unterprogramm in AWL geschrieben und mit RET abgeschlossen.

```
INTR    LAD   E 043
        ...
        ...
        RET
```

HINWEIS: Kontaktpläne können nicht als Interruptroutinen verwendet werden, da alle Kontaktpläne im Programm den gleichen temporären Speicher verwenden und es somit zu Fehlzuweisungen kommen kann.

14.6 BEDIENUNG ÜBER AWL

Die Bedienung der drei Zähler und des Referenz- bzw. Interrupteingangs kann auch über die AWL erfolgen. Die vorgenommenen Einstellungen werden in der END Routine übernommen.

Wenn eine sofortige Übernahme der Einstellungen nötig ist, können Betriebssystemroutinen aufgerufen werden. Dazu wird auf einen Pointer im Betriebssystem gesprungen. Der Pointer zeigt auf die Adresse der Routine.

Istwerte und Vorwahlwerte der Zähler liegen auf Betriebssystemadressen. Die Zählerinitialisierung erfolgt über drei Register (Steuerregister, Modusregister und Initialisierungsregister).

14.6.1 Betriebssystemadressen der Zähler

Istwerte

Adressen	Istwert	Bereich
$2100 - $2103	Zähler 0	0 - 4 294 967 295
$2104 - $2107	Zähler 1	0 - 4 294 967 295
$2108 - $210B	Zähler 2	0 - 4 294 967 295
$210C - $210F	Zweikanalzähler (Encoder)	0 - 4 294 967 295
$2123 - $2124	Hardwarezähler 0	0 - 65535

Vorwahlwerte

Adressen	Vorwahlwert	Bereich
$2110 - $2113	Zähler 0	0 - 4 294 967 295 [1)]
$2114 - $2117	Zähler 1	0 - 4 294 967 295
$2118 - $211B	Zähler 2	0 - 4 294 967 295
$211C - $211F	Zweikanalzähler (Encoder)	0 - 4 294 967 295

[1)] Wenn Zähler 0 als Hardwarezähler verwendet wird, liegt der Vorwahlwert zwischen 3 und 65535 (2 Byte-Zähler).

14.6.2 Steuerregister $2120

Bit	Funktion
0	Enable Zähler 0
1	Enable Zähler 1
2	Enable Zähler 2
3	Enable Zweikanalzähler (Encoder)
4	Clear Zähler 0
5	Clear Zähler 1
6	Clear Zähler 2
7	Clear Zweikanalzähler (Encoder)

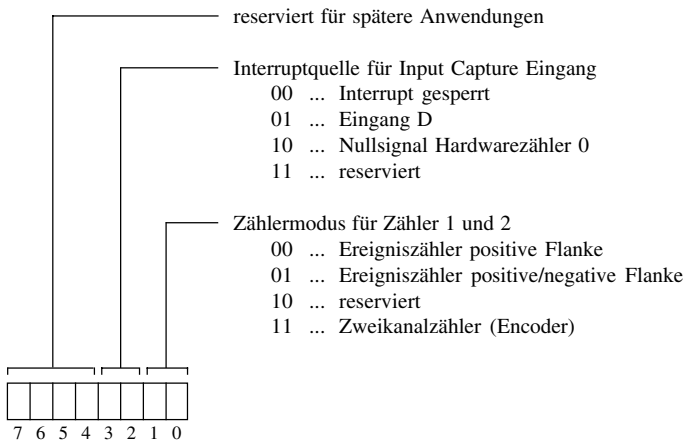
Enable Zähler

Mit diesen Bits kann jeder Zähler einzeln gestoppt und wieder freigegeben werden.

Clear Zähler

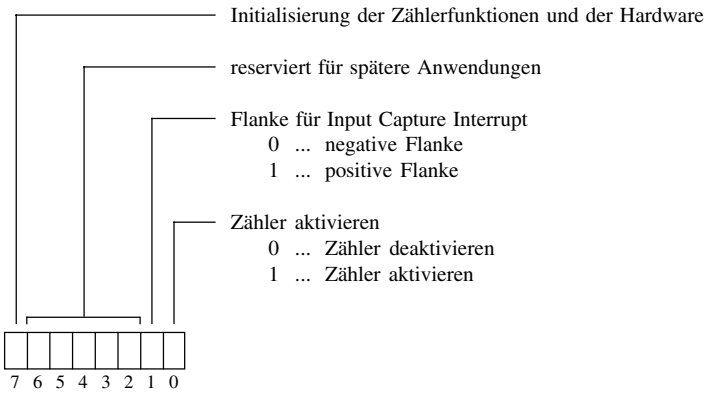
Diese Bits dürfen nur für einen Programmdurchlauf gesetzt werden. Wenn Bit 0 im Initialisierungsregister ($2122) gesetzt ist, wird der entsprechende Zähler während der Abarbeitung der END Routine auf den Vorwahlwert gesetzt.
Für die Betriebssystemroutinen "Lade Zähler x mit Vorwahlwert" müssen diese Bits nicht gesetzt werden.

14.6.3 Modusregister $2121



Der Ereigniszähler 0 reagiert immer auf die positive Flanke. Diese Einstellung kann nicht geändert werden.
Wenn der Hardwarezähler 0 als Interruptquelle verwendet wird, kann der Vorwahlwert für den Hardwarezähler 0 nur zwischen 3 und 65535 liegen.

14.6.4 Initialisierungsregister $2122



Wenn das Nullsignal von Hardwarezähler 0 als Interruptquelle selektiert wird, muß Bit 1 auf die positive Flanke eingestellt werden. Anderenfalls wird kein Interrupt ausgelöst.
Eine Änderung von Bit 1 wird nur übernommen, wenn sich die Bits 0 - 3 des Modusregisters ändern.

14.6.5 Adressen der Betriebssystemroutinen

Adresse	Istwert
$C114	Modusregister setzen
$C117	Referenzimpuls aktivieren
$C11A	Referenzimpuls lesen
$C11D	Interruptquelle Nullsignal Hardwarezähler 0 aktivieren
$C123	Zähler 0 mit Vorwahlwert laden
$C126	Zähler 1 mit Vorwahlwert laden
$C129	Zähler 2 mit Vorwahlwert laden
$C12C	Zweikanalzähler (Encoder) mit Vorwahlwert laden
$C12F	Zähler 0 lesen
$C132	Zähler 1 und 2 lesen
$C135	Zweikanalzähler (Encoder) lesen
$C138	Input Capture Interrupt freigeben
$C13B	Input Capture Interrupt sperren
$C13E	Referenzieren beenden
$C141	Zähler initialisieren

14.6.6 Funktion der Zähler

Im Initialisierungsregister ($2122) muß Bit 0 gesetzt werden, um die Zähler und die Generierung des Input Capture Interrupts zu aktivieren.

Die Zählerfunktionen können gesammelt oder für jeden Zähler einzeln gesetzt werden (Steuerregister $2120).

Im Steuerregister ($2120) kann jeder einzelne Kanal (abhängig vom Betriebsmodus) freigegeben bzw. gestoppt werden. Freigegebene Zähler dekrementieren den entsprechenden Istwertspeicherbereich ($2100 - $210F) der Zählerstände. Es handelt sich dabei um einen 4 Byte-Wert (0 - 4 294 967 295). Bei Überlauf wird beim maximalen Zahlenbereich weiter dekrementiert.

Wenn der Hardwarezähler 0 als Interruptquelle verwendet wird, liegt der Zahlenbereich zwischen 0 und 65535 (Vorwahlwert: 3 - 65535).

Im Steuerregister ($2120) kann jeder einzelne Kanal (abhängig vom Betriebsmodus) auf den Vorwahlwert gesetzt werden. Diese müssen vorher in den entsprechenden Speicher ($2110 - $211F) eingetragen werden.

Die Vorwahlwerte der Zähler können jederzeit geändert werden. Ebenso können die Zähler jederzeit auf den Vorwahlwert gesetzt werden.

14.6.7 Zähler initialisieren

Die Einstellung des Zählermodus, der Interruptquelle und die Aktivierung der Zähler erfolgt mit dem Modus- und dem Initialisierungsregister. Der Betriebsmodus kann allerdings auch während des Betriebs jederzeit geändert werden.

Damit die Einstellungen übernommen werden, muß Bit 7 im Initialisierungsregister ($2122) gesetzt werden. In der END Routine werden die Initialisierungen ausgeführt und Bit 7 wieder gelöscht.
Für eine spätere Änderung der Input Capture Flanke muß Bit 7 nicht gesetzt werden (siehe Abschnitt "14.6.8 Aufruf von Betriebssystemroutinen", Punkt 1).

HINWEIS: Nach einem Power-on oder nach einem Reset ist keine der Funktionen freigegeben.

Beispiel Initialisierungsregister $2122:
Zähler aktivieren
negative Flanke für Input Capture Interrupt
Bit 7 für Initialisierung setzen

Modusregister $2121:
Zähler 1 und 2 als Ereigniszähler mit positiver Flanke
Interruptquelle für Input Capture ist Eingang D

Der unten angeführte Programmteil wird nur einmal während dem ersten Programmdurchlauf ausgeführt. Die vorgenommenen Initialisierungen werden in der END Routine ausgeführt.

```
LD    # $2122          Initialisierungsregister
DXR
LAD   # %10000001      neg. Flanke Capture Interrupt, Zähler akt.
=     I 000
LD    # $2121          Modusregister
DXR
LAD   # %00000100      Interrupt Eingang D, Zähler pos. Flanke
=     I 000
```

14.6.8 Aufruf von Betriebssystemroutinen

1) Setze Modusregister

keine Aufrufparameter, keine Rückgabedaten

HINWEIS: Der Aufruf der Betriebssystemroutine darf frühestens einen Programmdurchlauf nach dem Initialisieren erfolgen.

Beispiel Initialisierungsregister $2122:

positive Flanke für Input Capture Interrupt

- Muß positiv sein, wenn Hardwarezähler 0 als Interruptquelle verwendet wird.
- Eine Änderung dieses Bits wird nur übernommen, wenn sich die Bits 0 - 3 des Modusregisters ändern.

Modusregister $2121:

Zähler 1 und 2 als Ereigniszähler mit positiver/negativer Flanke
Interruptquelle für Input Capture ist Hardwarezähler 0

Die Einstellungen werden sofort nach dem Aufruf der Betriebssystemroutine übernommen.

```
LD    # $2121          Modusregister
DXR
LAD   # %00001001      Zähler pos/neg Fl., Interrupt HW-Zähler 0
LB    # %00000011      positive Flanke Capture Interrupt
=D    I 000
LD    # $C114          Adr. der BS-Routine: "setze Modusregister"
DXR
SPU   I 000            Aufruf der Betriebssystemroutine
```

2) Referenzimpuls

Vor dem Aufruf "Lese Referenzimpuls" muß unbedingt die Funktion "Aktiviere Referenzimpuls" aufgerufen werden. Das Referenzbit 0 wird dabei auf 0 gesetzt.
Das Eintreffen eines Referenzimpulses setzt das Referenzbit auf 1. Dieses bleibt solange auf 1, bis wieder die Funktion "Aktiviere Referenzimpuls" aufgerufen wird.

Aktiviere Referenzimpuls:

keine Aufrufparameter, keine Rückgabedaten

```
LD    # $C117          Aufruf Aktiviere Referenzimpuls
DXR
SPU   I 000
```

Lese Referenzimpuls:

keine Aufrufparameter, Rückgabedaten in Akku A, Bit 0

```
LD    # $C11A          Aufruf Lese Referenzimpuls
DXR
SPU   I 000
B     # 001             Bit 0 in Akku A gesetzt ?
SN0   LABL
```

3) Aktiviere Interruptquelle Nullsignal Zähler 0

Übergabe des Vorwahlwertes in ERD, keine Rückgabedaten

```
LD    # $C11D          Aufruf Aktiviere Interruptquelle
DXR
LD    # 10500           z. B. Vorwahlwert = 10500 Impulse
SPU   I 000
```

Bei Aufruf dieser Funktion werden die nötigen Änderungen im Modusregister ($2121) und im Initialisierungsregister ($2122) ausgeführt. Der Zähler wird direkt mit dem Übergabewert in ERD beschrieben.

HINWEIS: Falls diese Funktion in Kombination mit dem Funktionsblock CMDC verwendet wird, muß der am MODE Eingang angeschlossene 8 Bit-Speicher angepaßt werden. Ansonsten wird der Betriebsmodus in der END Routine wieder verändert.

```
LAD   # % 100010xx
=     C xxxx
LD    # $C11D          Aufruf Aktiviere Interruptquelle
DXR
...
```

4) Vorwahlwert des Zählers x ändern

Die Vorwahlwerte der Zähler sind auf den Adressen $2110 - $211F gespeichert. Um die Werte zu ändern, muß wie im Beispiel beschrieben vorgegangen werden.

Beispiel Der Vorwahlwert von Zähler 0 wird geändert. Der Wert steht in den 8 Bit-Speichern C 2000 - C 2003.

```
LD    #$2110            Startadresse Vorwahlwert Zähler 0
DXR
LD    C 2000            Higher Word Vorwahlwert
=D    I 000
LD    C 2002            Lower Word Vorwahlwert
=D    I 002
```

5) Zähler x mit Vorwahlwert laden

Die Vorwahlwerte der Zähler sind auf den Adressen $2110 - $211F gespeichert. Die Pointer zu den Betriebssystemroutinen liegen auf den folgenden Adressen:

Adresse	Istwert
$C123	lade Zähler 0 mit Vorwahlwert
$C126	lade Zähler 1 mit Vorwahlwert
$C129	lade Zähler 2 mit Vorwahlwert
$C12C	lade Zweikanalzähler (Encoder) mit Vorwahlwert

HINWEIS: Die entsprechenden Bits (4 - 7) im Steuerregister **müssen nicht** gesetzt sein.

Um die Werte zu ändern, muß folgendermaßen vorgegangen werden:

```
LD    # $xxxx            Adresse des Pointers auf BS-Routine
DXR
SPU   I 000
```

6) Zähler x lesen

Die Istwerte der Zähler sind auf den Adressen $2100 - $210F gespeichert. Die Pointer zu den Betriebssystemroutinen liegen auf den folgenden Adressen:

Adresse	Istwert
$C12F	lese Zähler 0
$C132	lese Zähler 1 und 2
$C135	lese Zweikanalzähler (Encoder)

HINWEIS: Die entsprechenden Bits (0 - 3) im Steuerregister **müssen gesetzt** sein.

Um die Werte zu lesen, muß folgendermaßen vorgegangen werden:

```
LD    # $xxxx          Adresse des Pointers auf BS-Routine
DXR
SPU   I 000
```

7) Istwert in 8 Bit-Speicherbereich kopieren

Die Istwerte der Zähler sind auf den Adressen $2100 - $210F gespeichert. Um die Werte in einen 8 Bit-Speicherbereich zu kopieren, muß wie im Beispiel beschrieben vorgegangen werden.

Beispiel Der Istwert von Zähler 0 wird in den 8 Bit-Speicherbereich C 3000 - C 3003 kopiert.

```
LD    #$2100            Startadresse Istwert Zähler 0
DXR
LD    I 000             Higher Word Istwert
=D    C 3000
LD    I 002             Lower Word Istwert
=D    C 3002
```

8) Hardwarezähler 0

Der Istwert des Hardwarezählers 0 wird in den Adressen $2123 und $2124 gespeichert. Er wird mit der Funktion "Lade Zähler 0" ausgelesen (siehe Punkt 6).

Durch Hardwarebeschränkungen kann der Stand des 4 Byte-Zählers 0 im Speicherbereich $2100 - $2103 nicht mit dem Hardwarezähler synchronisiert werden.
Dieser Effekt tritt nur auf, wenn als Betriebsart **"Nullsignal Zähler 0 als Interruptquelle"** eingestellt ist. Beim Setzen des Zählers auf verschiedene Vorwahlwerte während des Betriebs, kann nicht mehr zwischen tatsächlicher Positionsänderung, Überlauf oder neuem Startwert unterschieden werden.
Bei allen anderen Betriebsarten treten keine Probleme auf.

Die Adressen $2123 und $2124 zeigen immer die verbleibenden Impulse bis zum Interrupt an.

HINWEIS: Wenn der Zähler mit dem Vorwahlwert geladen wird, wird der Vorwahlwert nicht sofort übernommen. Der nächste Impuls dekrementiert noch einmal den alten Zählerstand.

Beispiel

3000	Laden mit Vorwahlwert (z. B. mit 5000)
2999	der nächste Impuls dekrementiert den alten Zählerstand
5000	der Vorwahlwert wird übernommen
4999	der zweite Impuls dekrementiert vom Vorwahlwert ausgehend
:	

15. RELAIS-AUFSTECKKARTE FÜR BRCOMP

15.1 ALLGEMEINES

Für die Kompaktsteuerung ist eine Relais-Aufsteckkarte erhältlich. Dadurch erhält man zu den vorhandenen 14 Transistorausgängen zusätzlich 16 Relaisausgänge (A 060 - A 06F).

Die Kompaktsteuerung und die Relais-Aufsteckkarte können entweder nebeneinander oder übereinander montiert werden.
Mit dem Kabel BRKA08-0 wird über die PATA Schnittstelle die Verbindung zur Steuerung hergestellt. Das Kabel muß extra bestellt werden.

Der Funktionsbaustein DOUC überträgt die Ausgangszustände zur Relais-Aufsteckkarte.

15.2 RELAISGRUPPEN

Gruppe	Ausgänge	Ausführung
1	4	Schließer
2	4	Schließer
3	2	Schließer
4	2	Schließer
5	1	Schließer
6	1	Schließer
7	1	Umschalter
8	1	Umschalter

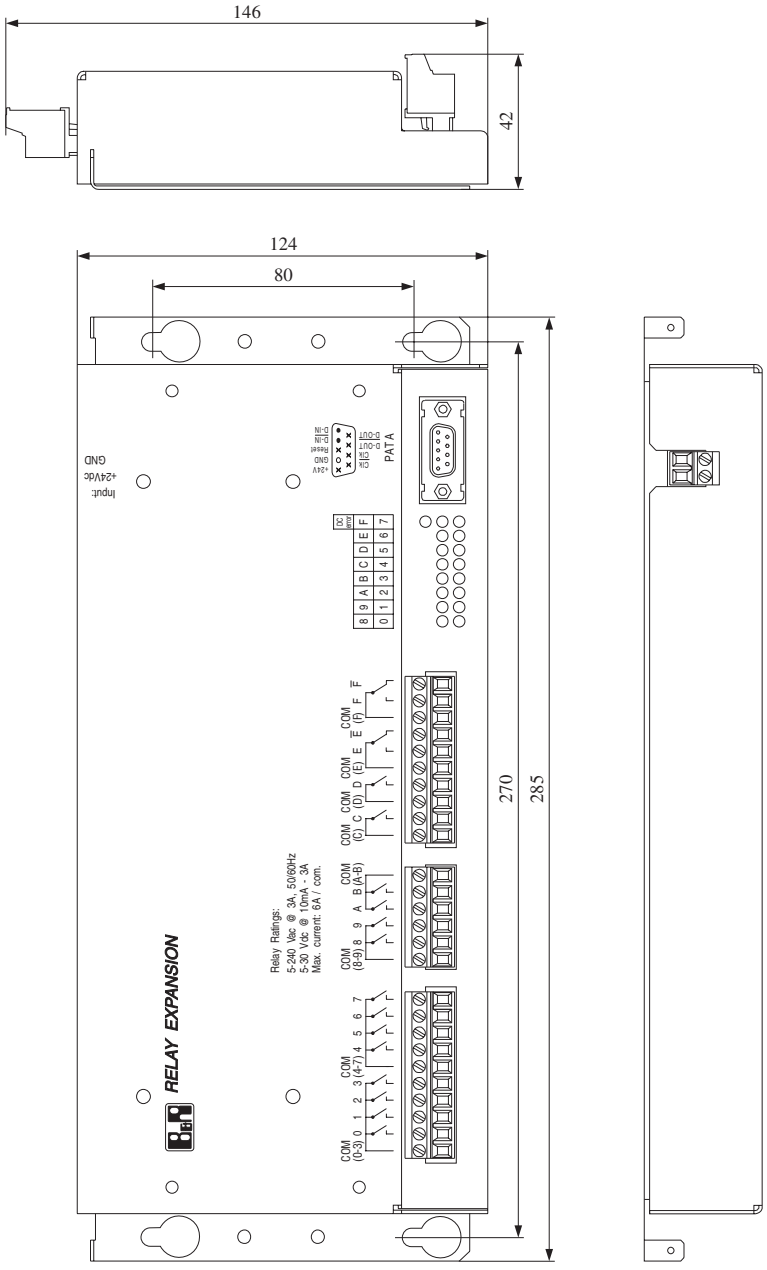
15.3 TECHNISCHE DATEN

Anzahl	16
Typ	Relais
Anzahl der Gruppen	8
Schaltspannung	5 - 240 VAC, 50/60 Hz 5 - 30 VDC
Schaltstrom pro Ausgang pro Gruppe	 max. 3 A max. 6 A
Schaltverzögerung log. 0 → 1 log. 1 → 0	 ca. 10 ms ca. 15 ms
Schutzbeschaltung	extern durch Anwender
Schaltspiele mechanisch elektrisch	 $> 2 \times 10^7$ $> 1 \times 10^5$
Relais-Versorgung	24 VDC ±15 %, max. 400 mA
Betriebstemperatur	0 bis 55 °C
Luftfeuchtigkeit	0 - 95 % nicht kondensierend

15.4 PINBELEGUNG

Feldklemme	Pin	Belegung
	1	Bezugspotential (A 060 - A 063)
1	2	Ausgang 0
2	3	Ausgang 1
3	4	Ausgang 2
4	5	Ausgang 3
5	6	Bezugspotential (A 064 - A 067)
6	7	Ausgang 4
7	8	Ausgang 5
8	9	Ausgang 6
9	10	Ausgang 7
10		
	11	Bezugspotential (A 068 - A 069)
11	12	Ausgang 8
12	13	Ausgang 9
13	14	Bezugspotential (A 06A - A 06B)
14	15	Ausgang A
15	16	Ausgang B
16		
	17	Bezugspotential (A 06C)
17	18	Ausgang C
18	19	Bezugspotential (A 06D)
19	20	Ausgang D
20		
21	21	Bezugspotential (A 06E)
22	22	Ausgang E - Schließer
23	23	Ausgang E - Öffner
24		
25	24	Bezugspotential (A 06F)
26	25	Ausgang F - Schließer
	26	Ausgang F - Öffner

15.5 ABMESSUNGEN



15.6 EINBAURICHTLINIEN

Die Relais-Aufsteckkarte kann horizontal oder vertikal montiert werden. Der durch die Befestigungslaschen bedingte Abstand zum Nachbarmodul reicht für eine ausreichende Luftzirkulation.

Die maximale Betriebstemperatur von 55 °C ist unterhalb der Relais-Aufsteckkarte einzuhalten. Es ist keine Fremdbelüftung des Gehäuses erforderlich.

15.7 MONTAGE

Es gibt zwei Montagemöglichkeiten

- direkt an Schaltschrankrückwand
- auf Hutschiene

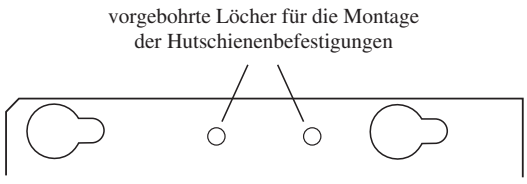
Direkt an Schaltschrankrückwand

Das Chassis ist über die vier Befestigungslöcher gut leitend mit der Schaltschrankrückwand zu verschrauben.

Als Schrauben sind M5 Schrauben zu verwenden (Lochabstand: 80 * 270 mm).

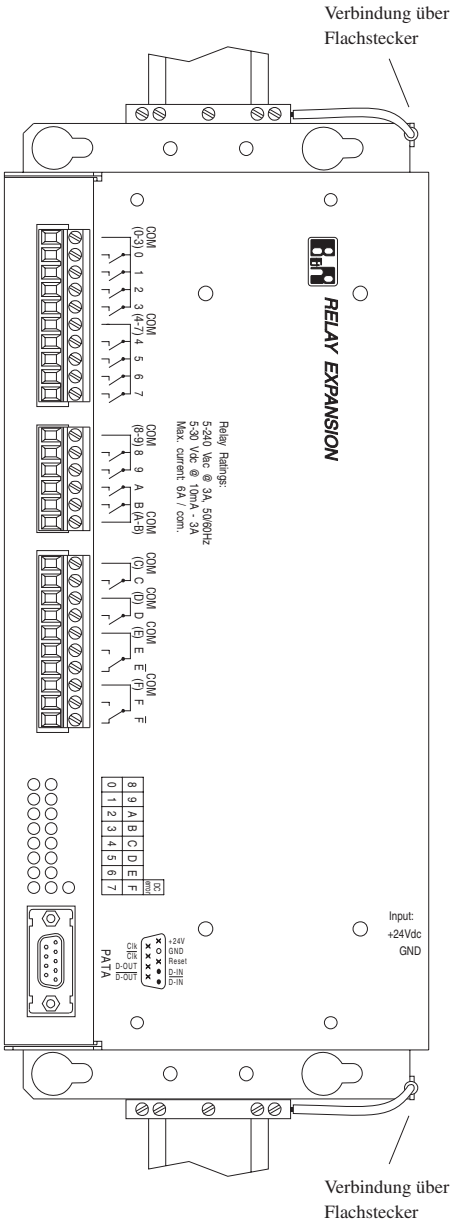
Auf Hutschiene

Für die Montage auf einer Hutschiene (Trageschiene DIN EN 50022-35) müssen die zwei mitgelieferten Hutschienenbefestiger (Typ: KSA10) mit der Relais-Aufsteckkarte verschraubt werden.



Relais-Aufsteckkarte auf der Hutschiene befestigen. Die Hutschiene muß geerdet sein!

Links und rechts von der Relais-Aufsteckkarte eine Schutzleiterklemme einsetzen. Relais-Aufsteckkarte über die vorbereiteten Kontakte erden.



15.8 KOMPAKTSTEUERUNG UND RELAIS-AUFSTECKKARTE

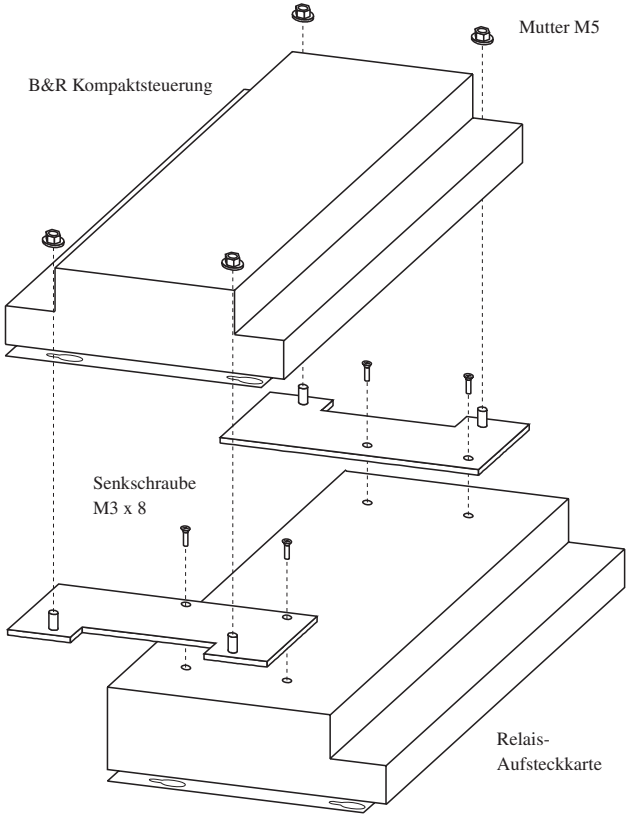
Die Kompaktsteuerung und die Relais-Aufsteckkarte können entweder nebeneinander oder übereinander montiert werden.

1. Nebeneinander

Wenn die beiden Gehäuse nebeneinander montiert werden, beachten Sie bitte die angegebenen Einbau- und Montagehinweise.

2. Übereinander

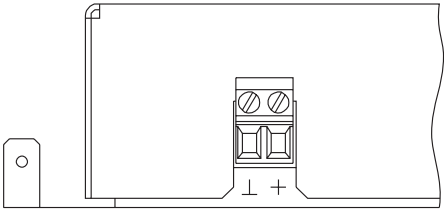
In diesem Fall wird zuerst die Relais-Aufsteckkarte direkt an die Schaltschrankrückwand oder auf die Hutschiene montiert (Hinweise beachten). Anschließend wird die Kompaktsteuerung über zwei Montageplatten mit der Relais-Aufsteckkarte verschraubt.
Die Montageplatten sind im Lieferumfang der Relais-Aufsteckkarte enthalten.



15.9 VERBINDUNG MIT DER KOMPAKTSTEUERUNG

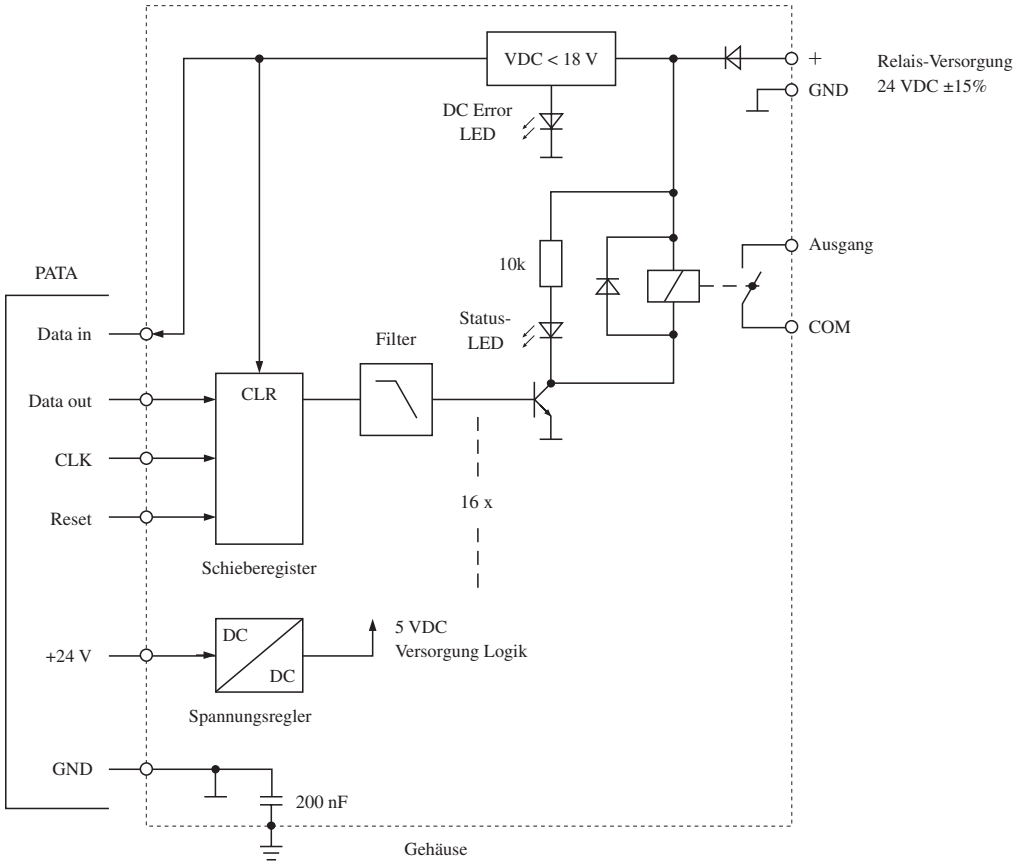
Mit dem Kabel BRKA08-0 wird über die PATA Schnittstelle die Verbindung zur Steuerung hergestellt. Das Kabel muß extra bestellt werden.

15.10 RELAIS-VERSORGUNG



+ +24 VDC ±15 %, max. 400 mA
⊥ GND

15.11 SCHALTKREIS



16. IF1 - RS232 SCHNITTSTELLE

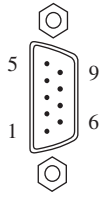
16.1 TECHNISCHE DATEN

Schnittstelle	RS232
galvanisch getrennt	NEIN
Anschluß	9poliger DSUB Stecker
Reichweite	max. 15 m, geschirmtes Kabel
Handshakeleitungen [1)]	DSR, RTS
Baudrate	300 - 19200 Baud
Datenformat Datenbits Parity Einstellung	 5 bis 8 ja / nein / gerade / ungerade softwaremäßig
Versorgung für Compact MMI P120 und MMI P121	+5 V / 500 mA

[1)] Das DTR Signal kann durch entsprechende Bedienung des Schnittstellenbausteins aus RTS gebildet werden (siehe Abschnitt "18. Bedienung der Schnittstellen IF1 und IF2").

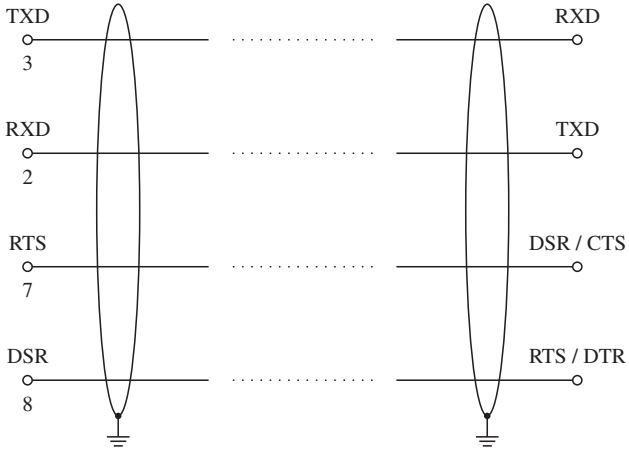
16.2 PINBELEGUNG

9poliger DSUB-Stecker



Pin	Belegung
1	
2	RXD
3	TXD
4	+5 V / 500 mA
5	GND
6	
7	RTS
8	DSR
9	

16.3 VERDRAHTUNGSSCHEMA



16.4 SCHIRMUNG UND ERDUNG

Für Schnittstellenverbindungen müssen geschirmte Kabel verwendet werden. Der Kabelschirm ist auf beiden Seiten zu erden.

16.5 SOFTWAREMÄSSIGE BEDIENUNG

Die softwaremäßige Bedienung der Schnittstelle ist im Abschnitt "18. Bedienung der Schnittstellen IF1 und IF2" beschrieben.

16.6 COMPACT MMI P120 UND MMI P121

Für die Kompaktsteuerung sind die Bedientableaus Compact MMI P120 und MMI P121 erhältlich. Sie werden an die Schnittstelle IF1 angeschlossen (Verbindungskabel: BRKACOMP1-0). Die Versorgung erfolgt über Pin 4.

17. IF2 - RS232/RS485 SCHNITTSTELLE

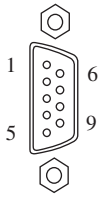
17.1 TECHNISCHE DATEN

Schnittstelle	RS232	RS485
galvanisch getrennt	NEIN	JA
Reichweite	max. 15 m, geschirmtes Kabel	max. 1200 m, geschirmtes Twisted Pair
Handshakeleitungen [1]	DSR, RTS	
Anschluß	9polige DSUB Buchse	
Baudrate	300 - 19200 Baud	
Datenformat Datenbits Parity Einstellung	 5 bis 8 ja / nein / gerade / ungerade softwaremäßig	

[1] Das DTR Signal kann durch entsprechende Bedienung des Schnittstellenbausteins aus RTS gebildet werden (siehe Abschnitt "18. Bedienung der Schnittstellen IF1 und IF2").

17.2 PINBELEGUNG

9polige DSUB-Buchse

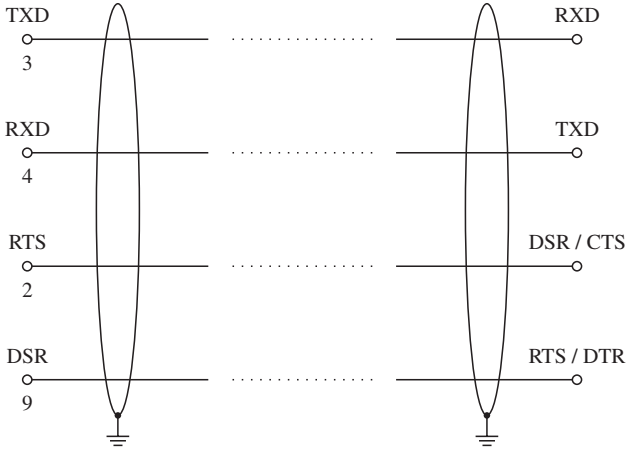


Pin	RS232	RS485
1	GND	
2	RTS	
3	TXD	
4	RXD	
5		$\overline{\text{DATA}}$
6		$\overline{\text{Enable}}$
7		GND [2]
8		$\overline{\text{DATA}}$
9	DSR	

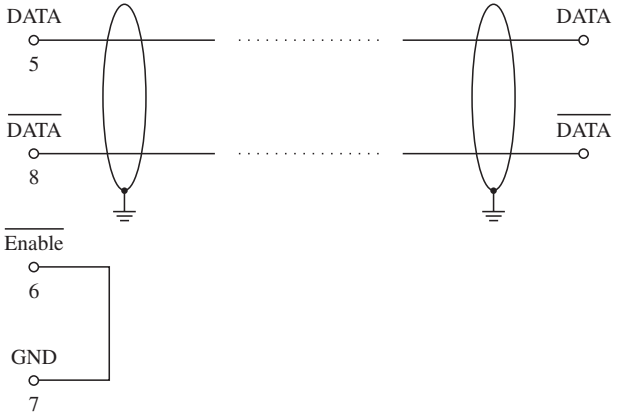
[2] Pin 6 ($\overline{\text{Enable}}$) muß bei RS485 Betrieb mit GND auf Pin 7 verbunden sein. Dies ist bei der Verdrahtung mit einem Standardkabel zu berücksichtigen (standardmäßig: Verbindung Pin 6 mit Pin 1).

17.3 VERDRAHTUNGSSCHEMA

RS232 Schnittstelle



RS485 Schnittstelle



17.4 SCHIRMUNG UND ERDUNG

Für Schnittstellenverbindungen müssen geschirmte Kabel verwendet werden. Der Kabelschirm ist auf beiden Seiten zu erden.

17.5 SOFTWAREMÄSSIGE BEDIENUNG

Die softwaremäßige Bedienung der Schnittstelle ist im Abschnitt "18. Bedienung der Schnittstellen IF1 und IF2" beschrieben.

18. BEDIENUNG DER SCHNITTSTELLEN IF1 UND IF2

Die softwaremäßige Bedienung der Anwenderschnittstelle erfolgt über die folgenden Register:

Register	IF1	IF2
Datenregister	P 100	P 300
Statusregister	P 101	P 301
Befehlsregister	P 102	P 302
Programmregister	P 103	P 303

Initialisierung

Bei der Initialisierung werden Programmregister und Befehlsregister mit bestimmten Vorwahlwerten beschrieben. Dadurch werden Baudrate, Datenformat, Parity usw. festgelegt. Die Initialisierung wird nur einmal unmittelbar nach dem Einschalten der SPS oder nach einem Reset durchgeführt.

Programmregister

7			0
SB	DB	1	BAUD

P 103 / P 303

SB	Anzahl Stoppbits	0 ... 1 Stoppbit	
		1 ... wenn DB=5 und kein Parity ... 1,5 Stoppbits	
		wenn DB=8 und Parity ... 1 Stoppbit	
		in allen anderen Fällen ... 2 Stoppbits	
DB	Anzahl Datenbits	00 ... 8 Datenbits	10 ... 6 Datenbits
		01 ... 7 Datenbits	11 ... 5 Datenbits
BAUD	Baudrate	0110 ... 300	1010 ... 2400
		0111 ... 600	1011 ... 3600
		1000 ... 1200	1100 ... 4800
		1001 ... 1800	
			1101 ... 7200
			1110 ... 9600
			1111 ... 19200

Befehlsregister

7						0
PAR	P_{on}	E	RT	0	1	1

P 102 / P 302

PAR	Parity	00 ... Parity ungerade (odd)
		01 ... Parity gerade (even)
		10 ... Paritybit beim Senden gesetzt
		11 ... Paritybit beim Senden gelöscht
P_{on}	Parity ein/aus	0 ... Kein Paritytest, Paritybit wird nicht generiert
		1 ... Paritytest aktiv
E	Echo Mode	0 ... Echo Mode aus
		1 ... Echo Mode ein, RT muß 0 sein
RT	RTS Leitung	0 ... RTS high, nicht sendebereit (-12 V)
		1 ... RTS low, sendebereit (+12 V)

Beispiel Initialisierung der Anwenderschnittstelle IF1:

Baudrate = 9600, 8 Datenbits, 1 Stoppbit, Parity aus, Echo Mode aus

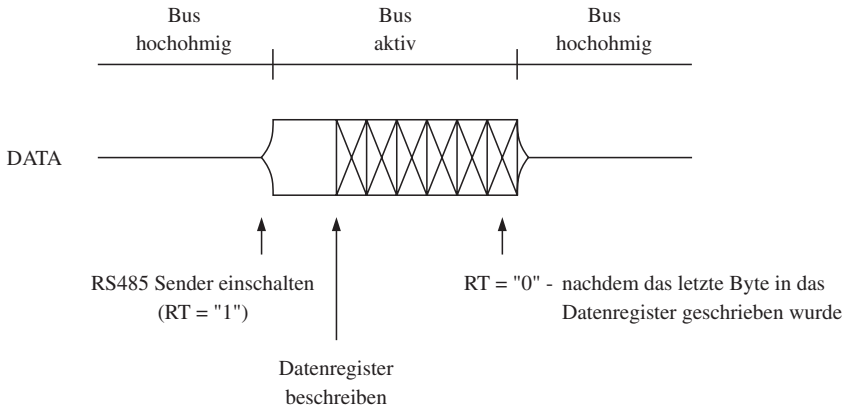
```
LB    # %00011110       9600 Baud, 8 Datenbits, 1 Stoppbit
LAD   # %00001011       Parity aus, Echo Mode aus
=D    P 102             Programmregister & Befehlsregister
```

Das RT Bit

RS485 Schnittstelle - Busankopplung:

Um den RS485 Sender einzuschalten, muß das RT Bit auf 1 gesetzt werden. Nach Beschreiben des Datenregisters mit dem letzten zu sendenden Byte, muß das RT Bit wieder rückgesetzt werden. Das Umschalten des Busses in den hochohmigen Zustand erfolgt nun automatisch nachdem das letzte Byte gesendet wurde.
Der Bus bleibt bis zur vollständigen Sendung des Zeichens aktiv (die maximale RTS Verzögerung beträgt 5 µs).

Wenn kein Sender aktiv und der Bus somit hochohmig ist, muß darauf geachtet werden, daß in diesem Zustand undefinierte Zeichen empfangen werden können.

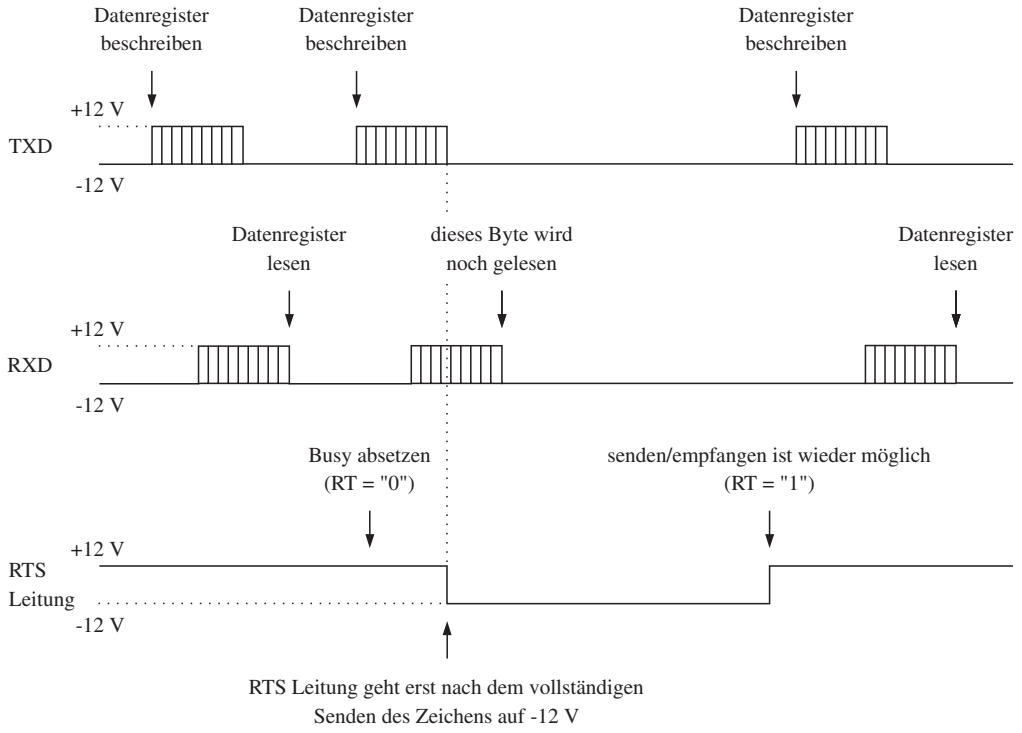


RS232 Schnittstelle - Punkt zu Punkt:

Bei einer RS232 Verbindung ist das RT Bit üblicherweise gesetzt. Wenn der Anwender ein Busy Signal absetzen möchte, kann er das RT Bit auf "0" setzen.

HINWEIS: Wenn das RT Bit "0" ist, ist der eigene Sender ebenfalls inaktiv!

Das folgende Diagramm zeigt die Signale einer RS232 Kommunikation.



HINWEIS: Der eigene Sender ist ebenfalls inaktiv.

Statusregister

Das Statusregister liefert Informationen über den Zustand der seriellen Schnittstelle und über eventuell aufgetretene Fehler. Der Zustand des Statusregisters muß bei jedem Sende- oder Empfangsvorgang berücksichtigt werden.

Statusregister

7						0
	DS		TR	RF	OV	FE	PE

P 101 / P 301

DS	Zustand DSR-Leitung	0 ...	DSR low - Gegenstation bereit
		1 ...	DSR high - Gegenstation nicht bereit
TR	Sender bereit	0 ...	Sender sendet Zeichen
		1 ...	Senderegister leer. Sender bereit, ein Zeichen zu senden
RF	Zeichen empfangen	0 ...	kein Zeichen empfangen
		1 ...	Zeichen wurde empfangen
OV	Overrunfehler	0 ...	kein Fehler
		1 ...	Fehler. Der Empfänger wurde nicht rechtzeitig gelesen, bevor ein neues Zeichen empfangen wurde
FE	Framingfehler	0 ...	kein Fehler
		1 ...	Fehler. Stoppbit nicht erkannt.
PE	Parityfehler	0 ...	kein Fehler
		1 ...	Fehler beim Paritytest

Datenregister

Das Datenregister hat zwei Funktionen:

- ankommende Zeichen werden aus dem Datenregister ausgelesen
- auszugebende Zeichen werden in das Datenregister geschrieben

Datenregister

7							0
			DATA				

P 100 / P 300

DATA auszugebende bzw. empfangene Daten

Zeichen ausgeben

Vor dem Beschreiben des Datenregisters mit dem auszugebenden Zeichen ist zu überprüfen, ob der Sender bereit ist, ein Zeichen zu senden (Bit 4 im Statusregister muß 1 sein).

```
LB    P 101            Statusregister
BB    # %00010000      Sender bereit ?
SP0   NO               Sprung, wenn Sender nicht bereit
LAD   x xxx            auszugebendes Zeichen
=     P 100            Datenregister
```

Zeichen einlesen

Durch Auswerten des Bits 3 im Statusregister wird festgestellt, ob ein Zeichen empfangen wurde. Ist dieses Bit = 1, wurde ein Zeichen empfangen. Die Bits 0 bis 2 des Statusregisters geben an, ob Übertragungsfehler aufgetreten sind (Parityfehler, Overrunfehler oder Framingfehler). Wenn eines dieser Fehlerbits gesetzt ist, ist das empfangene Zeichen ungültig.
Das Datenregister muß aber auch im Fehlerfall ausgelesen werden, da dadurch die Fehlermeldung quittiert wird.

```
        LB    P 101            Statusregister
        BB    # %00001000      Zeichen empfangen ?
        SP0   NO               Sprung, wenn kein Zeichen empfangen
        LAD   P 100            Datenregister auslesen
        BB    # %00000111      Übertragungsfehler aufgetreten ?
        SN0   FAIL             Sprung, wenn Übertragungsfehler
        :                      Auswerten des empfangenen Zeichens

FAIL    :
```

18.1 VERWENDUNG DES SCHNITTSTELLENINTERRUPTS

Der Schnittstellenbaustein liefert folgende Interrupts:

a) Sendeinterrupt: Wenn der Sender bereit ist, ein Zeichen zu senden.

b) Empfangsinterrupt: Wenn ein Zeichen empfangen wurde (auch wenn ein Übertragungsfehler auftrat).

Zur Quittierung des Interrupts muß in der Interruptroutine bei einem:

a) Sendeinterrupt: Das Datenregister beschrieben werden.

b) Empfangsinterrupt: Das Statusregister ausgelesen werden.

IRQ Vektor

Der Interruptvektor ist in den 8 Bit-Speichern C 0976 und C 0977 gespeichert. Bevor der Interrupt freigegeben wird, muß der Vektor auf die Adresse der User-Interruptroutine gestellt werden. Die Definition erfolgt in der Initialisierungsroutine.

Freigabe

Im Befehlsregister (P 102 bzw. P 302) werden die beiden Interrupts freigegeben bzw. gesperrt. Voraussetzung für die Freigabe der Interrupts ist, daß DTR aktiv ist (Bit 0 = 1).

Befehlsregister

7					0
PAR	P_{on}	E	TIC	I	1

P 102 / P 302

[1] RTS Pegel auf Leitung:

high ... -12 V
low ... +12 V

PAR Parity
- 00 ... Parity ungerade (odd)
- 01 ... Parity gerade (even)
- 10 ... Paritybit beim Senden gesetzt
- 11 ... Paritybit beim Senden gelöscht

P_{on} Parity ein/aus
- 0 ... Kein Paritytest, Paritybit wird nicht generiert
- 1 ... Paritytest aktiv

E Echo Mode
- 0 ... Echo Mode aus
- 1 ... Echo Mode ein, TIC muß 00 oder 01 sein

TIC Transmitter Interrupt Control
Steuerung des Sendeinterrupts und der Handshakeleitung RTS

- 00 ... RTS high [1], Sendeinterrupt gesperrt (Sender aus)
- 01 ... RTS low, Sendeinterrupt freigegeben (Sender ein)
- 10 ... RTS low, Sendeinterrupt gesperrt (Sender ein)
- 11 ... RTS low, Sendeinterrupt gesperrt (Break auf TXD = +12 V)

I IRD - Interrupt Request Disabled (Steuerung des Empfangsinterrupts)

- 0 ... Empfangsinterrupt freigegeben
- 1 ... Empfangsinterrupt gesperrt

Beispiele

1) Freigeben des Empfangsinterrupts:

```
LAD   # %00001001
=     P 102          Befehlsregister
```

2) Freigeben des Sendeinterrupts:

```
LAD   # %00000111
=     P 102          Befehlsregister
```

3) Freigeben des Sende- und Empfangsinterrupts:

```
LAD   # %00000101
=     P 102          Befehlsregister
```

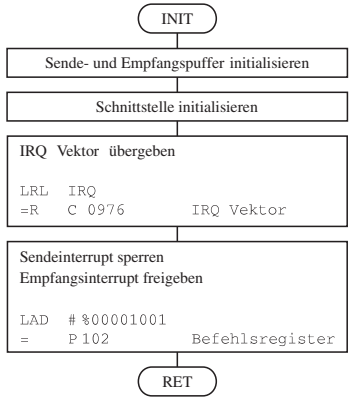
Interruptauswertung

Die beiden Interrupts können im Befehlsregister getrennt freigegeben und gesperrt werden. Sie führen jedoch auf denselben Vektor. Bei gleichzeitiger Verwendung des Empfangs- und Sendeinterrupts muß in der Interruptroutine festgestellt werden, ob der Interrupt vom Empfänger oder vom Sender verursacht wurde.
Das geschieht durch Auslesen des Bits RF "Zeichen empfangen" im Statusregister (Bit 3). Wenn das Bit = 0 ist, wurde der Interrupt vom Sender verursacht.

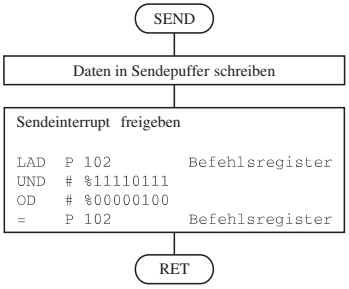
HINWEIS: Wenn beide Schnittstellen (IF1 und IF2) mit Schnittstelleninterrupt betrieben werden, müssen in der Interruptroutine die Statusregister **beider** Schnittstellen ausgewertet werden.

Anwendungsbeispiel

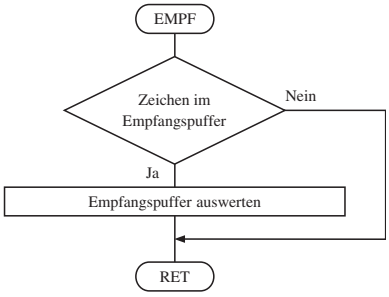
In diesem Beispiel werden Sende- und Empfangsinterrupt gleichzeitig verwendet. Im Unterprogramm "INIT" werden Initialisierungen vorgenommen.



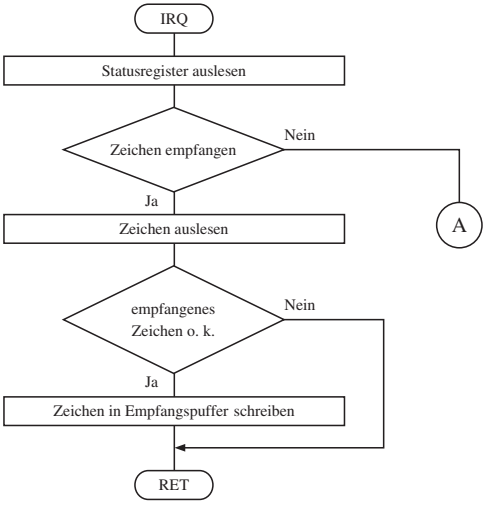
Im Unterprogramm "SEND" werden Daten in einen Sendepuffer geschrieben und der Sendeinterrupt freigegeben.



Im Unterprogramm "EMPF" wird kontrolliert, ob ein Zeichen im Empfangspuffer ist. Wenn ein Zeichen empfangen wurde, wird es ausgewertet.



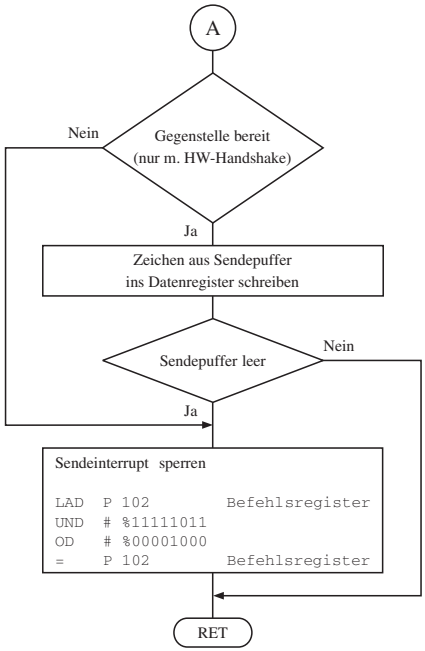
Die Interruptroutine "IRQ" ist in zwei Teile geteilt. In dem einen Teil wird ein empfangenes Zeichen kontrolliert und in den Empfangspuffer geschrieben. In dem anderen Teil der Routine wird ein Zeichen aus dem Sendepuffer in das Datenregister geschrieben.



HINWEIS:

Wenn die Gegenstelle nicht bereit ist, wird der Sendeinterrupt gesperrt. Die erneute Freigabe erfolgt im Unterprogramm "SEND".

Durch das Sperren des Interrupts wird verhindert, daß die Abarbeitung des Anwenderprogramms durch den dauernden Aufruf der Interruptroutine blockiert wird.



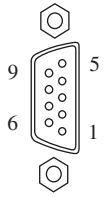
19. IF3 - PATA/SSI SCHNITTSTELLE

Die Schnittstelle IF3 ist eine modifizierte RS422 Schnittstelle. An die IF3 kann ein MINICONTROL Bedientableau oder eine Relais-Aufsteckkarte angeschlossen werden (PATA Schnittstelle).

Für den Anschluß absoluter Wegmeßsysteme ist sie auch als SSI Schnittstelle ausgeführt.

Pinbelegung

9polige DSUB-Buchse



Pin	Belegung
1	$\overline{\text{DATA IN}}$
2	DATA IN
3	Reset
4	GND
5	+24 V
6	$\overline{\text{DATA OUT}}$
7	DATA OUT
8	$\overline{\text{CLK}}$
9	CLK

Schirmung und Erdung

Für Schnittstellenverbindungen müssen geschirmte Kabel verwendet werden. Der Kabelschirm ist auf beiden Seiten zu erden.

PATA Schnittstelle

Die softwaremäßige Bedienung der MINICONTROL-Bedientableaus ist im "Bedientableaus Anwenderhandbuch" (MATERM2-0) beschrieben.

Für die Ansteuerung der MINICONTROL-Bedientableaus ist von B&R das Standardsoftwarepaket 2 SWSPSSTD02-0 (ab Rev. 00.31) erhältlich. Auf der Standardsoftwarediskette SWSPSBRC01-0 für die Kompaktsteuerung ist die Bedientableausoftware ebenfalls gespeichert.
Die Bedienung des Bedientableaus erfolgt über Funktionsblöcke und durch das Parametrieren von Tabellen.

Wichtige Adressen:

Adresse	Beschreibung
E 05F	Tastencode seriell
A 05E	Anzeigedaten seriell
A 05F	Resetsignal

SSI Schnittstelle

Für den Anschluß von Absolutgebern an die SSI Schnittstelle müssen folgende Voraussetzungen erfüllt werden:

- Es können nur Absolutgeber mit einer Monoflopzeit zwischen 20 µs und 260 µs verwendet werden.
- Es können Absolutgeber mit maximal 24 Bit (AG24) bzw. 32 Bit (AG32) eingelesen werden.

Für die Bedienung stehen die Funktionsblöcke AG24 und AG32 zur Verfügung. Sie sind im Standardsoftwarepaket 4 SWSPSPOS01-0 (ab Rev. 00.32) enthalten.

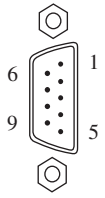
20. IF4 - B&R ONLINE SCHNITTSTELLE

Zur Kommunikation mit dem Programmiergerät verfügt die Kompaktsteuerung über eine Online Schnittstelle. Die Online Schnittstelle ist eine TTY Schnittstelle mit 62,5 kBaud, die nur für den Onlinebetrieb mit dem Programmiergerät verwendet werden kann.

Die Verbindung zum Programmiergerät wird mit einem Onlinekabel (BRKAOL-0) hergestellt.

Pinbelegung

9poliger DSUB-Stecker



Pin	Belegung
1	TXD
2	reserviert
3	RXD RET
4	Reset RET
5	reserviert
6	TXD RET
7	RXD
8	Reset
9	reserviert

21. IF5 - CAN BUS

21.1 MERKMALE DES CAN BUSSES

- Feldbus
- geringe Kosten
- hohe Störsicherheit durch Differenzsignale
- Busstruktur
- offenes System
- schnelle Datenübertragung für kleine Datenpakete (bis zu 8 Bytes)
- Fehlererkennung mittels CRC (Cyclic Redundancy Check) und Rahmenprüfung -> Hamming Distance 6
- vorhersagbare Übertragungszeiten für hochpriore Meldungen (Echtzeitverhalten)
- einfache Anwendung

21.2 B&R UND CAN

Der von B&R eingesetzte Controller entspricht der CAN Bus Spezifikation 2.0B. Auf einem Bus können somit die Protokolle Standard-CAN und Extended-CAN betrieben werden.

Die B&R Software unterstützt derzeit die Standard-CAN Identifier (11 Bit).

21.3 BUSLÄNGE UND KABELTYP

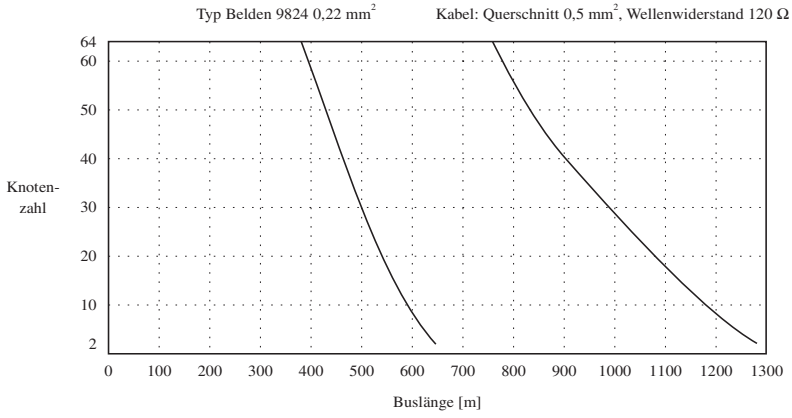
Der zu verwendende Kabeltyp hängt weitgehend von der geforderten Buslänge und der Knotenzahl ab. Die Buslänge wird hauptsächlich von der Bitrate bestimmt.

Die Tabelle auf der nächsten Seite enthält einige Werte für die maximale Buslänge in Abhängigkeit von der Übertragungsgeschwindigkeit und der Synchronisationssprungweite (SJW). In der vierten Spalte ist die zulässige Oszillatortoleranz angegeben.

Die Synchronisationssprungweite (SJW) ist der Faktor, der bestimmt, wie weit der Controller nachsynchronisieren kann. Je größer die Synchronisationssprungweite ist, desto kleiner wird die Buslänge.

Bitrate [kBit/s]	Synchronisationssprungweite (SJW)	Buslänge [m]	zul. Oszillatortoleranz [%]
500	0	67	0,121
	1	56	0,242
	2	33	0,363
	3	10	0,485
250	0	215	0,121
	1	192	0,242
	2	147	0,363
	3	101	0,485
125	0	510	0,121
	1	465	0,242
	2	374	0,363
	3	283	0,485
100	0	658	0,121
	1	601	0,242
	2	488	0,363
	3	374	0,485
50	0	1397 [1)]	0,121
	1	1284 [1)]	0,242
	2	1056 [1)]	0,363
	3	829	0,485
20	0	3613 [1)]	0,121
	1	3329 [1)]	0,242
	2	2761 [1)]	0,363
	3	2193 [1)]	0,485
10	0	7306 [1)]	0,121
	1	6738 [1)]	0,242
	2	5602 [1)]	0,363
	3	4456 [1)]	0,485

Zusammenhang zwischen Knotenzahl und Buslänge bei bestimmten Kabeltypen:



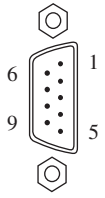
[1)] Nach CiA (CAN in Automation) ist die maximale Buslänge 1000 m.

21.4 ANSCHLUSSBELEGUNG

Die Kompaktsteuerung **BRCOMP2-0** (Betriebssystemversion 4.1) ist mit einer CAN Bus-Schnittstelle nach ISO-DIS 11898 ausgestattet. Die Anschlußbelegung ist nach CiA DS 102-1 ausgeführt.

Pinbelegung

9poliger DSUB-Stecker



Pin	Belegung
1	
2	CAN L
3	CAN GND
4	
5	
6	
7	CAN H
8	
9	

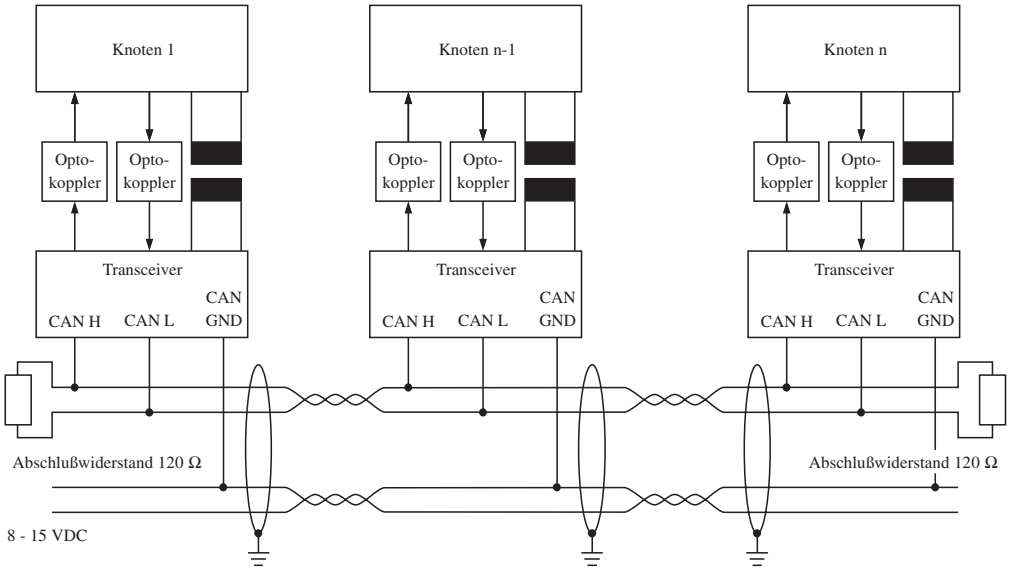
21.5 VERDRAHTUNG

Die Verdrahtung erfolgt in einer Busstruktur. Beide Busenden sind mit einem Abschlußwiderstand zu beschalten. Ein Knoten kann mit einer Stichleitung an den Bus angeschlossen werden. Die Länge der Stichleitung darf 30 cm nicht überschreiten.
Für das Buskabel ist grundsätzlich ein 4adriges Kabel, in Paaren verdrillt, zu verwenden.

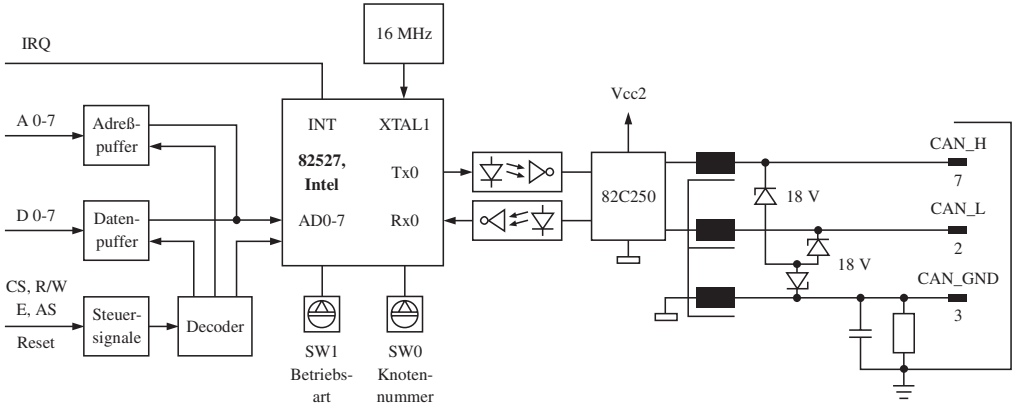
CAN Signale	CAN H	...	CAN High
	CAN L	...	CAN Low
	CAN GND	...	CAN Ground
	CAN+	...	CAN Versorgung 8 - 15 V

Da sämtliche CAN Schnittstellen von B&R intern versorgt werden, wird CAN+ nicht angeschlossen.

21.5.1 Verdrahtungsprinzip



21.5.2 Blockschaltbild eines Knotens



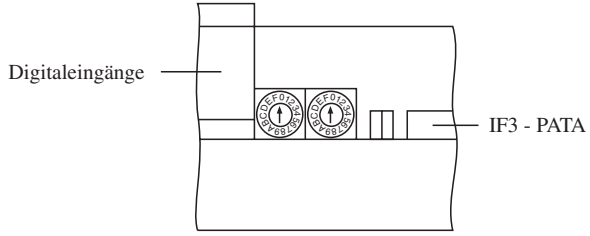
21.6 BEDIENUNG DES CAN BUSSES

Der CAN Bus wird über die Funktionsblöcke CNSW und CNCS bedient. Die Funktionsblöcke sind als Standardsoftware erhältlich (auf 3,5" Disketten).

Sprache	Bestellnummer
deutsch	SWSPSCAN01-0
englisch	SWPLCCAN01-0

21.6.1 CNSW - CAN Knotenschalter

Der Funktionsblock liest die Schalterstellungen der beiden Hex-Schalter (zwischen Digitaleingänge und IF3 - PATA) ein und stellt sie dem CAN Client/Server Funktionsblock CNCS zur Verfügung. Zusätzlich wird jede Schalterstellung direkt als Wert zwischen 0 - 15 ausgegeben. Der Funktionsblock CNSW erleichtert die Verwendung gleicher Programme für unterschiedliche Client-Stationen.



21.6.2 CNCS - CAN Client/Server

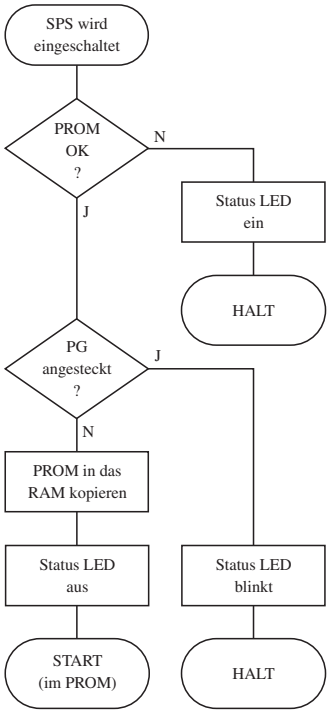
Der Funktionsblock ermöglicht die Kommunikation über ein Standard CAN Netz (11 Bit-ID) mittels CAL/CMS Diensten zur Übertragung von Objektdaten. Es werden keine Layer- oder Netzwerk-Managementdienste und keine Identifier-Verteilungsdienste unterstützt.
Das heißt, es handelt sich nach CAL um einen LMT/NMT/DBT-Slave der Klasse 0 mit statischer CAN ID Verteilung. Die Kommunikation erfolgt mittels des CMS Protokolls für Variablen und für "Uncontrolled Events".
Die Client/Server Dienste für "Read-Only Access, Basic Variable" sind nicht implementiert. Von den CMS-Datentypen werden nur Integer als Byte-Vielfaches unterstützt.

Transferdaten (max. 8 Bytes) werden über den Bus in aufsteigender Byte-Reihenfolge (LSB bis MSB) übertragen (Little Endian).

Abkürzungen

CAL	...	CAN Application Layer
CMS	...	CAN based Message Specification
LMT	...	Layer Management
NMT	...	Network Management
DBT	...	Kommunikationsobjekt Distributor

22. EINSCHALTVERHALTEN (POWER-ON)



23. BEFEHLSSATZ

In der Kompaktsteuerung kommt ein 6303 Prozessor (Hitachi) zum Einsatz. Das ist derselbe Prozessor, der auch in den Zentraleinheiten der MINICONTROL verwendet wird. Dadurch ist volle Software-kompatibilität zum nächstgrößeren SPS-System gegeben.

Eine vollständige Beschreibung des Befehlssatzes des 6303 Prozessors ist in der Kurzbeschreibung "AWL Befehlsbeschreibung" (MAAWLKB-0) zu finden.

Diese Kurzbeschreibung enthält u. A. folgende Informationen:

- B&R- und MOTOROLA-Mnemoniks
- Befehlsbeschreibung
- mögliche Adressierungsarten
- mögliche Adreßvorwahlen
- Länge und Dauer der Befehle
- veränderte Flags

24. MATHEMATIKROUTINEN

Die Zentraleinheit der Kompaktsteuerung ist standardmäßig mit schnellen Fließkomma Mathematikroutinen ausgestattet. Diese Routinen sind Bestandteil des Betriebssystems. Sie werden durch Befehlsmnemoniks aus der Anweisungsliste aufgerufen.
Neben den Grundrechenarten Addition, Subtraktion, Multiplikation, Division und Quadratwurzel stehen zahlreiche Umwandlungs- und Hilfsprogramme zur Verfügung (z. B. zum Vergleichen oder Kopieren). Zur Zahlendarstellung wird das genormte 4 Byte IEEE-Format verwendet. Eine detaillierte Beschreibung der Mathematikroutinen ist in der Kurzbeschreibung MAAWLKB-0 zu finden.

HINWEIS: Mathematikroutinen dürfen nicht in Interruptprogrammen verwendet werden.

Zahlenformate

Format	Zahlenbereich
IEEE-Fließkommaformat 31 24 23 16 15 8 7 0 S EXP S MANTISSE S ... Vorzeichen EXP ... 7 Bit Exponent MANTISSE ... 23 Bit Mantisse	$-9{,}22 * 10^{18}$ bis $-9{,}22 * 10^{-18}$ und $9{,}22 * 10^{-18}$ bis $9{,}22 * 10^{18}$
Absolut mit Vorzeichen lang 31 24 23 16 15 8 7 0 S ABSOLUTBETRAG	$\pm 2{,}15 * 10^{9}$
Absolut mit Vorzeichen kurz 15 14 8 7 0 S ABSOLUTBETRAG	±32767
Integer lang 31 24 23 16 15 8 7 0 S 2er-Komplement	$\pm 2{,}15 * 10^{9}$
Integer kurz 15 14 8 7 0 S 2er-Komplement	-32768 bis +32767

Bef.	Funktion	Quelle bzw. Operanden	Ziel bzw. Ergebnis	Ausführungszeit in µs	Mögliche Fehlermeldungen												
					1	2	3	4	5	6	7	8	9	10	11	12	13	14
MADD	OP1 := OP1 + OP2	OP1, OP2	OP1	209/690	●	●				●	●					●		
MSUB	OP1 := OP1 - OP2	OP1, OP2	OP1	219/700	●	●				●	●					●		
MMUL	OP1 := OP1 * OP2	OP1, OP2	OP1	209/803	●	●				●	●					●		
MDIV	OP1 := OP1 / OP2	OP1, OP2	OP1	190/1980	●	●	●			●	●					●		
MSQR	OP1 := SQR(OP1)	OP1	OP1	71/8065	●	●				●	●	●				●		
MSGN	OP1 := OP1 * (-1)	OP1	OP1	85/85														
MCOP	OP2 := OP1	OP1	OP2	46/46														
MEXG	OP1 ↔ OP2	OP1, OP2	OP2, OP1	76/76														
LAL1	Lade OP1, abs. mit Vz. 4 Byte	(R)	OP1	190/339					●									
LAL2	Lade OP2, abs. mit Vz. 4 Byte	(R)	OP2	190/339					●									
LAW1	Lade OP1, abs. mit Vz. 2 Byte	ERD	OP1	83/250														
LAW2	Lade OP2, abs. mit Vz. 2 Byte	ERD	OP2	83/250														
LIL1	Lade OP1, int. 4 Byte	(R)	OP1	197/381					●									
LIL2	Lade OP2, int. 4 Byte	(R)	OP2	194/378					●									
LIW1	Lade OP1, int. 2 Byte	ERD	OP1	87/260														
LIW2	Lade OP2, int. 2 Byte	ERD	OP2	84/257														
LF1	Lade OP1, IEEE	(R)	OP1	88/125	●					●	●					●		
LF2	Lade OP2, IEEE	(R)	OP2	88/125	●					●	●					●		
CAF	ASCII - IEEE	(R)	OP1	280/2140	●					●	●		●					
SAW	Speichere OP1, abs. mit Vz. 2 Byte	OP1	ERD	158/373				●								●		
SAL	Speichere OP1, abs. mit Vz. 4 Byte	OP1	(R)	169/408				●								●		
SIW	Speichere OP1, int. 2 Byte	OP1	ERD	158/380				●								●		
SIL	Speichere OP1, int. 4 Byte	OP1	(R)	172/424				●								●		
SFX	Speichere OP1, IEEE	OP1	(R)	43/43														
CFA	OP1 - ASCII	OP1	(R)	352/7310	●			●		●	●					●		
CFA0	OP1 - ASCII mit Vornullen	OP1	(R)	310/7190	●			●		●	●					●		
CFEA	OP1 - ASCII mit Exp.	OP1	(R)	570/7140	●					●	●					●		
SFM1	Speichere OP1 in Speicher 1	OP1	MEM1	60/60														
SFM2	Speichere OP1 in Speicher 2	OP1	MEM2	60/60														
SFM3	Speichere OP1 in Speicher 3	OP1	MEM3	60/60														
RFM1	Lade OP2 aus Speicher 1	MEM1	OP2	56/56														
RFM2	Lade OP2 aus Speicher 2	MEM2	OP2	56/56														
RFM3	Lade OP2 aus Speicher 3	MEM3	OP2	56/56														
FM2B	Multiplikation 2 x 2 Byte	(R) int. 2 Byte, ERD	(C1048, 1049) 4 Byte	115/191														
FM3B	Multiplikation 3 x 2 Byte	(R) int. 3 Byte, ERD	(C1048, 1049) 5 Byte	156/270														
FM4B	Multiplikation 4 x 2 Byte	(R) int. 4 Byte, ERD	(C1048, 1049) 6 Byte	192/344														
CBCD	Binär - BCD	(ERD) abs. 3 Byte	(R) BCD 3 Byte	192/1180				●										
CBIN	BCD - Binär	(ERD) BCD 3 Byte	(R) abs. 3 Byte	112/223														
CIA	Binär - ASCII	(C1048, 1049)	(R)	380/2020				●										
CIA0	Binär - ASCII mit Vornullen	(C1048, 1049)	(R)	310/1960				●										
CBPP	Binär - physikalisch (Parameterber.)	(R)	(C1048, 1049)	2500/6700	●					●	●					●		
CBPQ	Binär - physikalisch (schnell)	ERD, (R)	ERD, OP1	780/1700	●					●	●					●		
CPBQ	Physikalisch - binär (schnell)	ERD, (R)	ERD, OP1	780/1500	●					●	●					●		
CBP	Binär - physikalisch	(C1046, 1047), (R)	(C1048, 1049), ERD, OP1	3400/8300	●					●	●					●		
CPB	Physikalisch - binär	(C1046, 1047), (R)	(C1048, 1049), ERD, OP1	3400/8300	●					●	●					●		
CIM	Inch - metrisch	(C1046, 1047), ERD	(R), ERD	307/472													●	●
CMI	Metrisch - Inch	(C1046, 1047), ERD	(R), ERD	307/472													●	●
FCOP	Speicherbereich kopieren	(R), ERD	(C1048, 1049)															
FSMB	Speicher mit Byte-Werten laden	(R), ERD, C1052	(R)	48 + L * 12														
FSMW	Speicher mit Wort-Werten laden	(R), ERD, C1052	(R)	48 + L * 14														
FCLR	Speicherbereich löschen	(R), ERD	(R)	48 + L * 12														
MCMP	OP1 mit OP2 vergleichen	OP1, OP2		201/223												●		
MHIL	Wenn OP1 > OP2 dann OP1 := OP2	OP1, OP2	OP1	215/271												●		
MLOL	Wenn OP1 < OP2 dann OP1 := OP2	OP1, OP2	OP1	215/271												●		

Fehlermeldungen

Die in der Tabelle (siehe vorhergehende Seite) mit ● gekennzeichneten Fehlermeldungen, sind für die jeweilige Funktion möglich. Tritt bei der Ausführung einer Routine ein Fehler auf, wird das Carry Flag gesetzt und die 8 Bit-Speicherstelle C 1024 enthält die Fehlernummer.

Fehlernummer	Beschreibung
1	Bei einer Berechnung wurde der darstellbare Zahlenbereich überschritten
2	Bei einer Berechnung wurde der darstellbare Zahlenbereich unterschritten
3	Division durch 0
4	Bereichsüberschreitung beim Umwandeln von Zahlenformaten
5	Beschneidung des Lower Significant Byte (LSB) beim Laden von 4 Byte-Mantissen
6	Bereichsüberschreitung beim Laden von Zahlen
7	Bereichsunterschreitung beim Laden von Zahlen
8	Negativer Operand bei Quadratwurzelberechnung
9	Unzulässiges Zeichen bei Stringumwandlungsroutine
10	nicht verwendet
11	Unzulässiges Kommando (Trapfehler wird ausgelöst)
12	Zahl nicht im Rechenbereich
13	Exponentfehler bei inch-metrisch- bzw. metrisch-inch-Umwandlung
14	Datenüberlauf bei inch-metrisch- bzw. metrisch-inch-Umwandlung

Operanden und Speicher

Speicherstellen	Funktion
C 1024	Fehlernummer
C 1025	reserviert
C 1026 bis C 1029	Operand 1 (OP1)
C 1030 bis C 1033	Operand 2 (OP2)
C 1034 bis C 1037	Zwischenspeicher 1 (MEM1)
C 1038 bis C 1041	Zwischenspeicher 2 (MEM2)
C 1042 bis C 1045	Zwischenspeicher 3 (MEM3)
C 1046 bis C 1047	Quelladresse
C 1048 bis C 1049	Zieladresse
C 1050 bis C 1051	Länge
C 1052 bis C 1053	Daten

25. SYSTEMSPEICHERSTELLEN

Einige 8 Bit-Speicher und 1 Bit-Speicher sind für Betriebssystemfunktionen reserviert. Diese dürfen vom Anwenderprogramm nicht bzw. nur eingeschränkt verwendet werden:

8 Bit-Speicher: C 0800 bis C 1499
1 Bit-Speicher: M 800 bis M 999

1 Bit-Speicher mit Adressen ab M 800, die für Betriebssystem-Sonderfunktionen verwendet sind, werden mit Adressen F Dxx bzw. Z Dxx eingegeben:

Adresse	**Einzugeben als** [1)]
M 800	F D00
M 801	F D01
:	:
M 899	F D99
M 900	Z D00
M 901	Z D01
:	:
M 999	Z D99

[1)] Das Programmiergerät erlaubt auch die Eingabe der M-Adresse, nach Abschluß der Eingabe mit [↵] wird die Adresse automatisch in die Form F Dxx oder Z Dxx umgewandelt.

Eingabe: M 820
Wird nach [↵] geändert in: F D20

Eingabe: M 980
Wird nach [↵] geändert in: Z D80

Im folgenden Abschnitt sind die Systemspeicherstellen beschrieben, die vom Anwenderprogramm nur eingeschränkt verwendet werden dürfen:

Zulässiger Zugriff		Adresse(n)	Funktion
Lesen	Schreiben		
		C 0800 bis C 0863	Vorteiler für Softwarezeiten
●		C 0899	First Scan Speicher
		C 0900 bis C 0963	Zähler für Softwarezeiten
●	●	C 0968, C 0969	HW-Zählerinterrupt oder Interrupt-Eingangsevektor
		C 0972, C 0973	Timerinterruptvektor
		C 0974, C 0975	Timerinterruptzeit
●	●	C 0976, C 0977	IRQ Vektor
		C 0978, C 0979	Trapvektor
●	●	C 0980 bis C 0987	Echtzeituhr
●	●	C 0990	Breakpointsonderfunktion
		C 0991 bis C 0993	Zähler/Teiler
		C 0998, C 0999	Runtimeüberwachung
●	●	C 1000 bis C 1013	Analogeingänge / Analogausgänge
●	●	C 1024 bis C 1053	Operanden u. Speicher der Mathematikroutinen
		C 1054 bis C 1499	Reserviert für Standardfunktionsblöcke
●	●	F D00 bis F D63	Freigaben für Softwarezeiten
●	●	F D85, F D86	Steuerbits für Echtzeituhr
●		Z D00 bis Z D63	Softwarezeiten
●		Z D64	First Scan Flag
●		Z D80 bis Z D83	Zeittakte
●		Z D90 bis Z D93	Zeitimpulse
●		Z D99	Batteriekontrolle

26. FIRST SCAN FLAG

Das First Scan Flag ist eine 1 Bit-Speicherstelle (Z D64), die vom Betriebssystem automatisch während des ersten Programmzyklus auf 1 gesetzt wird, sonst ist dieses Flag 0. Das First Scan Flag wird für Programminitialisierungen verwendet. Auch die Speicherstelle C 0899 liefert die First Scan Funktion:

Z D64	First Scan Flag (1 = erster Programmzyklus)
C 0899	First Scan Speicher (1 = erster Programmzyklus)

Beispiel

```
INIT  LAD   Z D64       First Scan
      SP0   INIR        Sprung, wenn schon initialisiert
      :
      :                 Initialisierungen
      :
INIR  RET
```

Im Kontaktplan kann das First Scan Flag an den Enable Eingang von Funktionsblöcken angeschlossen werden, die nur einmal während des ersten Programmzyklus ausgeführt werden sollen.

HINWEIS: Mit dem Befehl "XFER" des B&R Programmiersystems können Programme ohne Unterbrechung des laufenden Anwenderprogramms in den RAM Speicher der Zentraleinheit übertragen werden.
Der Anwender muß nach erfolgter Übertragung manuell mit einem Befehl vom Programmiergerät auf das neue Programm umschalten. In diesem Fall sind die First Scan Speicherstellen während des ersten Programmzyklus des neuen Programms nicht gesetzt!

Da die First Scan Speicherstellen nicht gesetzt sind, ist bei Verwendung von Interruptroutinen oder B&R MININET Vorsicht geboten.
Die Adresse der Interruptroutine wird nach einer Programmänderung nicht aktualisiert, wenn die Definition in der Initialisierungsroutine erfolgt.

27. LITHIUM-BATTERIE

Die Kompaktsteuerung ist mit einer Lithium-Batterie ausgestattet. Sie wird für die Pufferung des SRAM und der Echtzeituhr benötigt.

HINWEIS: Lithium-Batterien sind nicht umweltverträglich und fallen in die Kategorie Sondermüll. Bitte beachten Sie die in Ihrem Land geltenden gesetzlichen Entsorgungsvorschriften.

Pufferdauer

3 Jahre bei 70 °C
6 Jahre bei 40 °C

HINWEIS: Es ist zu beachten, daß im Betrieb die Gehäuseinnentemperatur um ca. 20 °C über der Außentemperatur liegen kann.

Spannungsüberwachung

Der Zustand der Batterie kann mit dem Bit 1 der Inport Adresse $3480 (siehe auch Abschnitt "34. Inport Adresse $3480) kontrolliert werden (Hardwareüberwachung).

0 ... Batterie OK (Spannung > 2,55 V)
1 ... Batterie leer (Spannung < 2,45 V)

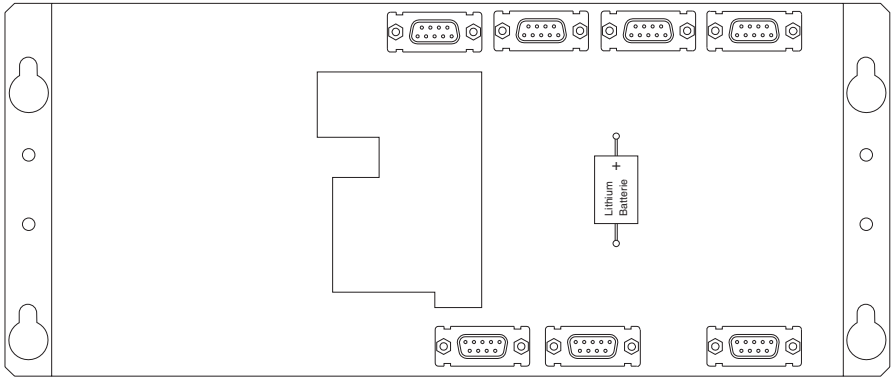
Der Bitstatus kann auf einen Ausgang gelegt werden. Bei Verwendung eines Bedientableaus kann auch eine entsprechende Meldung ausgegeben werden.

Aus kompatibilitätsgründen kann der Zustand der Batterie auch mit der 1 Bit-Speicherstelle Z D99 kontrolliert werden. Diese Methode sollte allerdings nicht verwendet werden, da es sich um eine Softwareüberwachung handelt!

Batteriewechsel

Der Wechsel muß im spannungslosen Zustand erfolgen.

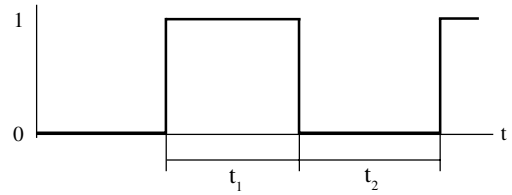
HINWEIS: Die im RAM gespeicherten Daten und die Uhrzeit gehen verloren. Mit dem Befehl "UPDL" des B&R PROgrammierSYStems können die Daten jedoch gesichert werden.



28. ZEITTAKTE

Zeittakte sind 1 Bit-Adressen, die vom Betriebssystem automatisch mit Blinktakten angesteuert werden:

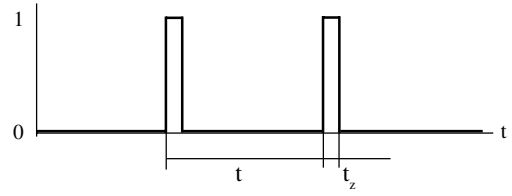
Adresse	t_1	t_2
Z D80	10 ms	10 ms
Z D81	40 ms	60 ms
Z D82	0,4 s	0,6 s
Z D83	4 s	6 s



29. ZEITIMPULSE

Zeitimpulse sind 1 Bit-Adressen, die vom Betriebssystem automatisch für die Dauer eines Programmzyklus auf 1 gesetzt werden.

Adresse	t
Z D90	10 ms
Z D91	100 ms
Z D92	1 s
Z D93	10 s



t_z ... Programmzyklus

30. ECHTZEITUHR

Wenn die SPS ausgeschaltet ist, läuft die Uhrzeit weiter (gepuffert von der Lithium-Batterie).

Uhrzeitspeicherstellen (alle Angaben in BCD):

C 0980	1/100 Sekunden ($00 bis $99)
C 0981	Sekunden ($00 bis $59)
C 0982	Minuten ($00 bis $59)
C 0983	Stunden ($00 bis $23)
C 0984	Tag ($01 bis $31)
C 0985	Monat ($01 bis $12)
C 0986	Jahr ($00 bis $99)
C 0987	Wochentag (0 bis 6)

Die Steuerung der Echtzeituhr erfolgt über zwei Speicherstellen:

F D85	Uhr ein/aus (1 = ein)
F D86	Uhr stellen ein/aus (0 = stellen ein)

Stellen der Echtzeituhr (Uhr muß eingeschaltet sein, d. h. F D85 muß 1 sein):

- Uhr stellen ein (F D86 löschen)
- Uhrzeitspeicherstellen C 0980 bis C 0987 mit Uhrzeit/Datum laden
- F D86 wird beim nächsten Programmdurchlauf automatisch wieder gesetzt

31. SOFTWAREZEITEN

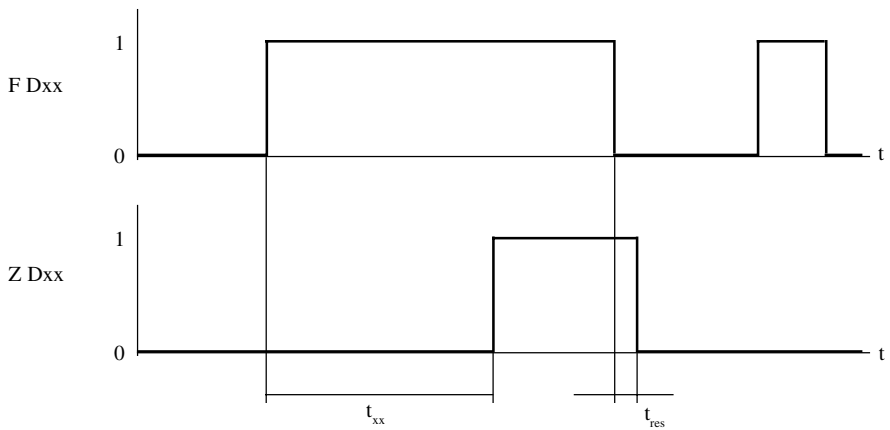
Die Zentraleinheit der Kompaktsteuerung verfügt über 64 Softwarezeiten, die als Anzugsverzögerung arbeiten. Jede Softwarezeit besteht aus folgenden Adressen:

F Dxx — Freigabe (starten) der Softwarezeit. Durch Beschreiben dieser Speicherstelle mit 1 wird die Softwarezeit xx (xx = 00 bis 63) gestartet. Diese Speicherstelle kann auch gelesen werden (z. B. um festzustellen, ob eine Softwarezeit gestartet ist).

Z Dxx — Ergebnis. Wenn diese Speicherstelle 1 ist, dann ist die dazugehörige Softwarezeit abgelaufen. Diese Speicherstelle kann nur gelesen werden. Das Zurücksetzen erfolgt durch Löschen der Freigabe F Dxx.

Zxx n"nn — Zeitdefinition. Mit der Anweisung Zxx wird die Dauer der Softwarezeit in Sekunden und 1/100 Sekunden festgelegt. Diese Anweisung muß immer durchlaufen werden. Sie steht deshalb meist am Anfang des Anwenderprogramms.

Zeitlicher Ablauf



Nach dem Start der Softwarezeit xx durch Beschreiben der Freigabeadresse F Dxx mit 1 und Ablauf der mit der Zeitdefinition Zxx eingestellten Zeit t_{xx} wird die Zeitadresse Z Dxx ebenfalls 1.

Nach dem Rücksetzen der Freigabeadresse F Dxx wird die Zeitadresse Z Dxx beim nächsten Durchlauf durch die Zeitdefinition Zxx zurückgesetzt. Die Rücksetzzeit t_{res} kann im ungünstigsten Fall einen Programmzyklus lang sein.

Beispiel 5,5 Sekunden nach Betätigen eines Tasters (E 042) wird ein Motor (A 058) gestartet. Mit einem weiteren Taster (E 043) wird der Motor wieder gestoppt:

```
0000        Z10         5"50        Zeitdefinition
0001        LAD    N    E 042       Taster START
0002        PRS         M 100       Pos. Flanke von E 042
0003        EXO         M 100       Pos. Flanke von E 042
0004        RST         M 100       Pos. Flanke von E 042
0005        PRS         F D10       Start Motorverzögerung
0006        LAD         E 043       Taster STOP
0007        RST         F D10       Start Motorverzögerung
0008        LAD         Z D10       Motorverzögerung
0009        =           A 058       Motor
0010        END
```

Dasselbe Programmbeispiel kann auch mit einem Kontaktplan gelöst werden:

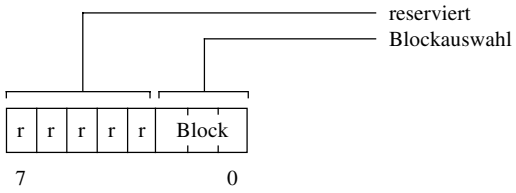
```
0000        Z10         5"50        Zeitdefinition
0001        SPU         KOP1        Kontaktplanaufruf
0002        END
```

```
!  E 042     M 100                                                    F D10
00 --I I---+---I+I---+---------+---------+---------+---------+---------+---(P)---
!  M START   FLANKE                                                   M VERZ.
!
!  E 043                                                              F D10
01 --I I---+---------+---------+---------+---------+---------+---------+---(R)---
!  M STOP                                                             M VERZ.
!
!   Z D10                                                             A 058
02 --I I---+---------+---------+---------+---------+---------+---------+---( )---
!  MOT.EIN                                                            MOTOR
```

Die Zeitdefinition Zxx muß bei jedem Programmdurchlauf genau einmal durchlaufen werden. Wird sie nicht durchlaufen, ist die Funktion der Softwarezeit nicht mehr gewährleistet. Wenn sie mehrmals je Programmzyklus durchlaufen wird, ist die angegebene Zeit nicht korrekt.

Jede Softwarezeit belegt eine 8 Bit-Speicherstelle im Bereich von C 0800 bis C 0863, der als Vorteiler verwendet wird und eine weitere 8 Bit-Speicherstelle im Bereich von C 0900 bis C 0963 als Zähler. Die Zeitdefinition Zxx ist ein Softwareinterrupt, der ca. 0,5 ms dauert (bei Verwendung vieler Softwarezeiten Auswirkung auf die Programmzykluszeit beachten!).

32. INPORT/OUTPORT ADRESSE $3400



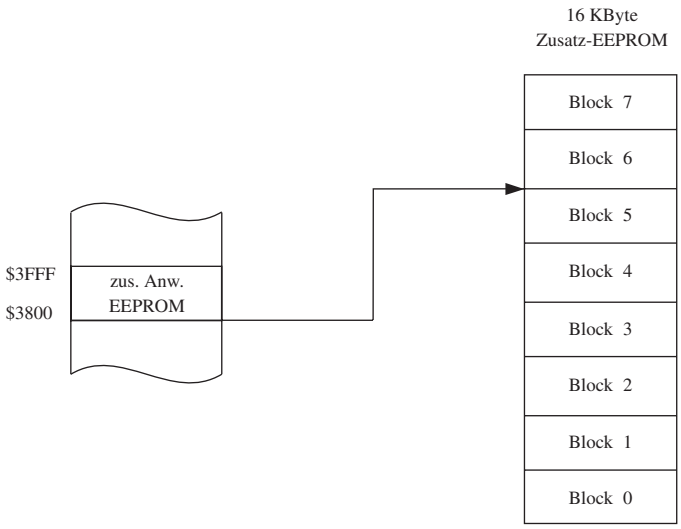
reserviert	Für spätere Anwendungen reserviert.
Blockauswahl	Mit den Bits 0 - 2 wird der anzusprechende 2 KByte Speicherbereich des Zusatz-EEPROM ausgewählt. 0 ... Block 0 1 ... Block 1 : : : : 7 ... Block 7 Nähere Beschreibung siehe Abschnitt "33. Zusätzliches Anwender-EEPROM".

33. ZUSÄTZLICHES ANWENDER-EEPROM

Von diesem EEPROM stehen dem Anwender 16 KBytes zur Verfügung. Es ist in 8 Blöcke zu je 2 KBytes unterteilt. Der gewünschte Block wird mit den ersten 3 Bits des Inport/Outport Bytes (Adresse $3400) definiert.

Beispiel Definition von Block 6 des Zusatz-EEPROM.

```
LD    # $3400          ERD mit $3400 laden
DXR                    Indexregister auf Adresse $3400 setzen
LAD   I 000            Inhalt in ERA laden
UND   # %11111000      Bit 0 bis 2 löschen
OD    # 006            Block 6 definieren
=     I 000            ERA in Adresse $3400 speichern
```



Der Anwender kann auf den selektierten Block des zusätzlichen Anwender-EEPROM über die Adressen $3800 bis $3FFF zugreifen.

33.1 DATEN LESEN

Zum Lesen von Daten aus einem Block wird das AWL Makro DFEE verwendet

Übergabeparameter:	Quelle	...	Blocknummer in C 0881 Offset zu Adresse $3800 in C 0882&
	Ziel	...	Indexregister
	Datenlänge	...	in Ergebnisregister D
Rückgabeparameter:	Kein Fehler aufgetreten:		
	Carry = 0	...	Datentransfer o.k.
	Fehler aufgetreten:		
	Carry = 1	...	Die Summe von Offset und Datenlänge liegt außerhalb des gültigen Bereichs (> $3FFF)
verwendete 8 Bit-Speicher:	C 0866&	...	Quelladresse
	C 0868&	...	Zieladresse
	C 0884&	...	aktuelle Datenlänge

```
DFEE  =D    C 0884 DATA 04          aktuelle Datenlänge
      =R    C 0868 & DEST 0
      LD    C 0882 DATA 02          Offset
      UND   # %00000111             begrenzen auf 2 KBytes / Block
      +D    # $3800                  Startadresse EEPROM-Block
      =D    C 0866 & SOURCE 0
      +D    C 0884 DATA 04          aktuelle Datenlänge
      -D    # $3FFF
      J<=   DFE0
*----------------------
      SEC                           Daten außerhalb gültigem Bereich
      RET
*----------------------
DFE0  LD    # $3400                  Inport/Outport Adresse
      DXR
      LAD   C 0881 DATA 01
      UND   # %00001111             Blocknummer auf 0 ... F begrenzen
      =     C 0881 DATA 01
      LAD   I 000
      UND   # %11110000
      OD    C 0881 DATA 01
      =     I 000                   Block anwählen
*---
```

```
      LD    C 0884 DATA 04          aktuelle Datenlänge
      SRD
      JC0   DFE1                    keine ungerade Datenlänge
      =D    C 0884 DATA 04          aktuelle Datenlänge
      LR    C 0866 & SOURCE 0
      LAD   I 000                   erstes Byte lesen
      IR
      =R    C 0866 & SOURCE
      LR    C 0868 & DEST 0
      =     I 000                   erstes Byte speichern
      IR
      =R    C 0868 & DEST 0
      LD    C 0884 DATA 04          aktuelle Datenlänge
*---
DFE1  =D    C 0884 DATA 04          aktuelle Datenlänge
      SP0   DFE2                    fertig
      LR    C 0866 & SOURCE 0
      LD    I 000                   Daten lesen
      IR
      IR
      =R    C 0866 & SOURCE 0
      LR    C 0868 & DEST 0
      =D    I 000                   Daten speichern
      IR
      IR
      =R    C 0868 & DEST 0
      LD    C 0884 DATA 04          aktuelle Datenlänge
      -D    # 00001
      SPI   DFE1                    noch nicht alle Daten kopiert
*----------------------
DFE2  CLC                           Datentransfer o.k.
      RET
```

Beispiel Aus Block 4 werden ab der Adresse $3A00 50 Bytes ausgelesen. Gespeichert werden die Daten ab C 2000.

```
      LAD   # 004                   Blocknummer 4
      =     C 0881 DATA 00
      LD    # $0200                  Offset zu $3800 (Quelladr. = $3A00)
      =D    C 0882 DATA 01
      LRK   C 2000 Zieladresse
      LD    # 00050                  Datenlänge
      SPU   DFEE                    Daten aus Zusatz-EEPROM lesen
      JC0   OK
      :                             ERROR-Auswertung
      :
```

33.2 DATEN SCHREIBEN

Beim Schreiben von Daten in ein EEPROM ist zu beachten, daß dies im Gegensatz zum Schreiben in einen 1 oder 8 Bit-Speicherbereich mit einer gewissen Verzögerung geschieht.

Zum Schreiben von Daten in einen Block wird das AWL Makro DTEE verwendet

Das AWL Makro DTEE eignet sich zum Programmieren von Parameterdaten, die sich während des Betriebs einer Anlage nicht ändern.

HINWEIS: Wenn das Anwenderprogramm im EEPROM läuft, kann das Zusatz-EEPROM nicht programmiert werden.

Wenn Daten in das EEPROM programmiert werden, wird das weitere Programm **nicht** bearbeitet!
Für das Beschreiben des EEPROM während eines Programmdurchlaufs kann der Funktionsblock MCEE verwendet werden. Er ist auf der Diskette "Utility.01" des Standardsoftwarepaketes 1 (SWSPSSTD01-0 ab Rev. 00.30) gespeichert.

Übergabeparameter:	Quelle	...	Indexregister
	Ziel	...	Blocknummer in C 0881 Offset zu Adresse $3800 in C 0882&
	Datenlänge	...	in Ergebnisregister D
Rückgabeparameter:	Kein Fehler aufgetreten:		
	Carry = 0	...	Datentransfer o.k.
	C 0881	...	Blocknummer des nächsten freien Bytes
	C 0882&	...	Offset zu $3800 des nächsten freien Bytes
	Fehler aufgetreten:		
	Carry = 1	...	Datentransfer fehlerhaft
	ERA	...	Fehlernummer: 1 - Datenlänge größer als freier Speicher 2 - EEPROM defekt
	C 0881	...	Blocknummer der defekten Speicherstelle
	C 0882&	...	Offset zu $3800 der defekten Speicherstelle
verwendete 8 Bit-Speicher:	C 0880	...	Runtimezähler
	C 0884&	...	aktuelle Datenlänge
	C 0886&	...	Quellpointer

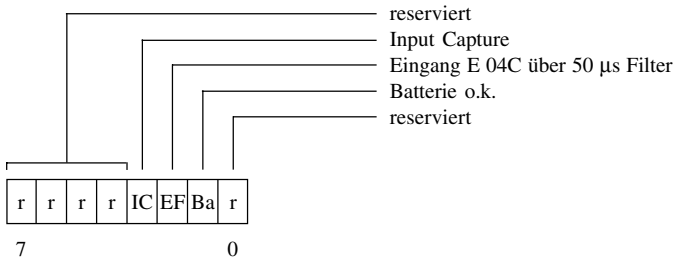
```
DTEE  =D    C 0884 DATA 04          aktuelle Datenlänge
      =R    C 0866 & SOURCE 0
      LD    C 0882 DATA 02          Offset
      UND   # %00000111             begrenzen auf 2 KBytes / Block
      +D    # $3800                 Startadresse EEPROM-Block
      =D    C 0882 DATA 02
      +D    C 0884 DATA 04
      -D    # $3FFF
      J<=   DTE5
*----------------------
      LAD   # 001                   ERROR ... Datenlänge größer als
      SEC                                     freier Speicher
      RET
*----------------------
DTE5  LAD   C 0998 CYCLE TIME COUNTER
      =     C 0886 DATA 06
DTE3  LAD   C 0886 DATA 06
      =     C 0998 CYCLE TIME COUNTER
*
      LD    # $3400                 Inport/Outport Adresse
      DXR
      LAD   C 0881 DATA 01
      UND   # %00001111             Blocknummer auf 0 ... F begrenzen
      =     C 0881 DATA 01
      LAD   I 000
      UND   # %11110000
      OD    C 0881 DATA 01
      =     I 000                   Block anwählen
*
      LR    C 0866 & SOURCE 0
      LAD   I 000                   Daten lesen
      LR    C 0882 DATA 02
      =     I 000                   akt. Kopierdaten abspeichern
*---
      ANS
      LD    # 01500
DTE0  -D    # 00001                 Warteschleife
      SN0   DTE0
      AVS
*---
      CLR   C 0880 DATA 00          Runtimezähler rücksetzen
DTE1  LB    I 000                   Daten von EEPROM mit aktuellen
      AVB                           Kopierdaten vergleichen
      SP0   DTE2
      INC   C 0880 DATA 00          Runtimezähler erhöhen
      LB    C 0880 DATA 00
      VB    # 200                   mit Runtime MAX vergleichen
      SP<   DTE1
*----------------------
      LAD   # 002
      SEC                           ERROR ... EEPROM defekt
      RET
*----------------------
```

```
DTE2  LD    C 0882 DATA 02          MEM-Adreßpointer erhöhen
      +D    # 00001
      =D    C 0882 DATA 02
      LR    C 0866 & SOURCE 0
      IR
      =R    C 0866 & SOURCE 0
      LD    C 0884 DATA 04          aktuelle Datenlänge
      -D    # 00001                 alle Daten kopiert ?
      =D    C 0884 DATA 04
      SP0   DTE4
      SPI   DTE3
DTE4  CLC                           Datentransfer o.k.
      RET
```

Beispiel In Block 7 werden ab der Adresse $3B00 40 Bytes geschrieben. Die zu schreibenden Daten sind ab der Speicherstelle C 2500 gespeichert.

```
LAD   # 007                   Blocknummer 7
=     C 0881 DATA 00
LD    # $0300                 Offset zu $3800 (Zieladr. = $3B00)
=D    C 0882 DATA 01
LRK   C 2500 Quelladresse
LD    # 00040                 Datenlänge
SPU   DTEE                    Daten in Zusatz-EEPROM schreiben
JC0   OK
:                             ERROR-Auswertung
:
```

34. INPORT ADRESSE $3480



reserviert	Für spätere Anwendungen reserviert.
Input Capture	Durch das Modusregister $2121 (siehe Abschnitt "14.6.3 Modusregister $2121") wird die Interruptquelle für den Input Capture Eingang bestimmt. Dieses Bit zeigt daher den Zustand des Eingangs E 04D bzw. das Nullsignal des Hardwarezählers 0 an.
Eingang E 04C	Der Eingang E 04C wird über ein 50 µs Filter gelesen (schneller Eingang).
Batterie o.k.	Dieses Bit dient zur Überwachung der Lithium-Batterie. Der Batteriezustand wird hardwaremäßig überwacht. 0 ... Batterie leer (Spannung < 2,45 V) 1 ... Batterie o.k. (Spannung > 2,55 V)
reserviert	Für spätere Anwendung reserviert.

35. WATCHDOG

35.1 SOFTWAREWATCHDOG

Der Softwarewatchdog überwacht die maximal zulässige Programmzykluszeit von 100 ms. Ist ein Programmzyklus nach dieser Zeit noch nicht beendet, wird das Anwenderprogramm gestoppt und ein Softwarereset ausgelöst (alle Ausgänge werden zurückgesetzt).
Ein Laufzeitfehler wird im Statustest des Programmiergerätes und durch Einschalten der Status-LED angezeigt.

35.2 HARDWAREWATCHDOG

Die Zentraleinheit der Kompaktsteuerung verfügt über einen Hardwarewatchdog, der selbst bei völligem Ausfall der Zentraleinheit noch in der Lage ist, das System in einen sicheren Betriebszustand zu bringen.

36. TIMERINTERRUPTROUTINEN

Unabhängig von der Länge des Anwenderprogramms wird alle 10 ms ein Interrupt ausgelöst und die sogenannte Timerinterruptroutine ausgeführt. Diese Betriebssystemfunktion wird für Sicherheits- und Diagnosefunktionen sowie für die Generierung von Softwarezeiten, Uhrzeitfunktionen, Zeittakten und Zeitimpulsen verwendet.

Der Timerinterruptvektor (die Adresse der Timerinterruptroutine) steht in C 0972&. Die Timerinterruptzeit ist in C 0974& gespeichert (Einheit µs). Timerinterruptvektor und Timerinterruptzeit dürfen vom Anwenderprogramm nicht geändert werden.

Zusätzlich zu den Betriebssystemfunktionen kann der Anwender selbst einen oder zwei Programmteile zeitgesteuert ausführen lassen (User-Timerinterruptroutinen). Dazu werden die Timerinterrupthandler $US1 und $US2 verwendet. Die Parameter:

ERA	gewünschter Zeitintervall in ms
R	Anfangsadresse der User-Timerinterruptroutine

Aufruf `SPU $US1` bzw. `SPU $US2`

Die User-Timerinterruptroutine wird mit RET abgeschlossen. Unabhängig vom gewählten Zeitintervall für die User-Timerinterruptroutine wird die Betriebssystem-Timerinterruptroutine alle 10 ms ausgeführt.

HINWEIS: Timerinterruptroutinen werden nicht ausgeführt, wenn die SPS im HALT-Zustand ist.

Zu häufiges Aufrufen von langen Timerinterruptroutinen kann die Programmzykluszeit wesentlich verlängern und zu Systemstörungen führen. Die Summe der Ausführungszeiten beider Timerinterruptroutinen darf maximal 300 µs betragen.

In Timerinterruptroutinen dürfen keine Betriebssystemmathematikroutinen verwendet werden.

Zum Ausschalten einer aktivierten User-Timerinterruptroutine wird ERA mit 0 geladen und der Interrupthandler ($US1 oder $US2) erneut aufgerufen.

Beispiel Alle 3 ms wird der Zählerstand eines Abwärtszählers ausgelesen und mit 10000 verglichen. Bei Unterschreitung dieses Wertes wird ein Ausgang gesetzt. Der Timerinterrupthandler $US1 wird nur einmal in einer Initialisierungsroutine aufgerufen:

```
INIT  LAD   Z D64       First Scan
      SP0   INIR
      LAD   # 003       3 ms
      LRL   TEST        Adresse der Interruptroutine
      SPU   $US1
INIR  RET

TEST  SPU   READ        Zählerstand auslesen
      -D    # 10000     Vergleich mit 10000
      JC0   TESR
      SET   A 040       Zähler low !
TESR  RET
```

37. FEHLERMELDUNGEN

Die Zentraleinheit ist mit umfangreichen Sicherheits- und Diagnosefunktionen ausgestattet (z. B. Programm-Checksumtest bei Power-on). Im Fehlerfall wird das Anwenderprogramm angehalten, die Status-LED eingeschaltet und ein Softwarereset ausgelöst, d. h. alle digitalen Ausgänge werden gelöscht, alle analogen Ausgänge werden auf 0 V bzw. 0 mA zurückgesetzt. Falls ein Programmiergerät angeschlossen ist, wird im Statustest eine Klartext-Fehlermeldung angezeigt (z. B. RUNTIME-FEHLER).

Die folgende Tabelle gibt eine Übersicht über alle möglichen Fehlermeldungen der Kompaktsteuerung:

Bezeichnung	**Beschreibung/Ursachen**	**Abhilfe**
Übertragungsfehler bei Download	Beim Übertragen eines Programms vom Programmiergerät in die SPS (Download) tritt ein Fehler auf. **Mögliche Ursachen:** Die Onlineverbindung zwischen PG und SPS wird durch starke, elektromagnetische Störungen beeinträchtigt	Programm erneut in die SPS übertragen.
Checksum-Fehler nach RUN	Ein mit RUN übertragenes Programm weist im RAM der SPS eine falsche Prüfsumme (Checksum) auf. **Ursache:** Programmspeicher defekt.	Programm erneut übertragen, im Wiederholungsfall EE32 tauschen
Checksumfehler	Die Prüfsumme (Checksum) des Anwenderprogramms ist nach Reset oder Power-on falsch. **Mögliche Ursachen:** Bei PROM-Programm PROM-Speicher defekt, bei RAM-Programm Batteriepufferung ausgefallen (leer oder defekt) oder Softwarefehler, der das Anwenderprogramm überschreibt.	Programm erneut übertragen. Im Wiederholungsfall Batteriepufferung überprüfen, Anwenderprogramm auf Softwarefehler untersuchen, Programmspeichermodul tauschen.
Runtimefehler	Die zulässige Programmzykluszeit von 100 ms wurde überschritten. **Mögliche Ursachen:** Softwarefehler, zu viele Programmschleifen, Endlosschleife.	Programmfehler beheben.

Bezeichnung	Beschreibung/Ursachen	Abhilfe
Pointerfehler	Beim Checksumtest während Power-on wurde festgestellt, daß Betriebs-systemvektoren nicht stimmen. **Mögliche Ursachen:** siehe "Check-sumfehler".	Siehe "Checksumfehler".
Kommunikations-fehler	Bei der Kommunikation zwischen dem Programmiergerät und der Zen-traleinheit (RUN, Statustest) tritt ein Fehler auf. **Mögliche Ursachen:** Die Online-verbindung zwischen PG und SPS wird durch starke, elektromagnetische Störungen beeinträchtigt.	Funktion wiederholen.
Stapelzeigerfehler	Am Programmende (END) steht der Stapelzeiger (Stackpointer) falsch. **Mögliche Ursachen:** Fehler im An-wenderprogramm (Unterprogramm nicht mit RET abgeschlossen, Fehler bei Verwendung des Systemstacks zur Datenspeicherung).	Programmfehler beheben.
Trapfehler	Unbekannter Prozessorbefehl **Mögliche Ursachen:** Fehler im An-wenderprogramm (z. B. indizierter Sprung auf Datenbereich).	Programmfehler beheben.

38. STICHWORTVERZEICHNIS

A

Abmessungen
- BRCOMP 13
- Relais-Aufsteckkarte 69

Adressen
- Inport Adresse $3480 117
- Inport/Outport Adresse $3400 110

AINJ - Analogeingang Kompaktsteuerung 38
Analogausgänge 43
- Analogwandlung 48
- AOTF - Analogausgang Kompaktsteuerung 47
- Ausgangsschaltung 44
- Betriebssystemroutine 48
- Pinbelegung 44
- Schirm erden 45
- Softwaremäßige Bedienung 46
- Technische Daten 43
- Verdrahtungsschema 45

Analogeingänge 27
- AINJ - Analogeingang Kompaktsteuerung 38
- Analogumwandlung 41
- Betriebssystemroutine 41
- Eingangsschaltung 31
- Kanalrefreshzeit 42
- Pinbelegung 31
- Schirm erden 32
- Softwaremäßige Bedienung 35
- Technische Daten 30
- Verdrahtungsschema 32

Anwender-EEPROM 111
AOTF - Analogausgang Kompaktsteuerung 47

B

Batterie 104
Befehlssatz 96
Bestellnummern 10

C

CAN Bus (BRCOMP2-0) 91
CMDC - Zählerauswertung (BRCOMP) 52
CMDD - Auswertung Zähler 0 (BRCOMP) 54
CNCS - CAN Client/Server 95
CNSW - CAN Knotenschalter 95
Compact MMI P120 und MMI P121 76

D

Digitalausgänge 23
- Ausgangsschaltung 24
- LEDs 19
- Schalten ohmsch-induktiver Lasten 25
- Schutzbeschaltung 26
- Technische Daten 23
- Verdrahtungsschema 24

Digitaleingänge 21
- Eingangsschaltung 22
- LEDs 19
- Technische Daten 21
- Verdrahtungsschema 22

E

Echtzeituhr 107
EEPROM 111
Einschaltverhalten 96

F

Fehlermeldungen 121
Feldklemmen (Zubehörsatz) 11
First Scan Flag 103
Funktionsblöcke
- AINJ - Analogeingang Kompaktsteuerung 38
- AOTF - Analogausgang Kompaktsteuerung 47
- CMDC - Zählerauswertung (BRCOMP) 52
- CMDD - Auswertung Zähler 0 (BRCOMP) 54
- CNCS - CAN Client/Server 95
- CNSW - CAN Knotenschalter 95

H

Hardwarewatchdog 118
Hutschiene 14

I

IF1 - RS232 Schnittstelle 75
IF2 - RS232/RS485 Schnittstelle 77
IF3 - PATA/SSI Schnittstelle 89
IF4 - B&R Online Schnittstelle 90
IF5 - CAN Bus (BRCOMP2-0) 91
Initialisierungsregister $2122 59
Inport Adresse $3480 117
Inport/Outport Adresse $3400 110
Input Capture 117
Interrupteingang 49, 50, 56

K

Komponenten 16

L

LEDs 19
- Betriebszustand 20
- Digitalausgänge 19
- Digitaleingänge 19
- Versorgung 19

Lithium-Batterie 104

M

Mathematikroutinen 97
MMI P120 76
MMI P121 76
Modusregister $2121 58
Montage
- BRCOMP 14
- Relais-Aufsteckkarte 70, 72

O

Online Schnittstelle 90

P

PATA/SSI Schnittstelle (IF3) 89
Power-On 96
Prozessor 12, 96

R

Referenzeingang 49, 50
Relais-Aufsteckkarte 67
- Abmessungen 69
- Einbaurichtlinien 70
- Kompaktsteuerung und Relais-Aufsteckkarte 72
- Montage 70
- Pinbelegung 68
- Relais-Versorgung 73
- Relaisgruppen 67
- Schaltkreis 74
- Technische Daten 68
- Verbindung mit der Kompaktsteuerung 73

RS232 Schnittstelle (IF1) 75
RS232/RS485 Schnittstelle (IF2) 77

S

Schnittstellen
- Bedienung 80
- IF1 - RS232 Schnittstelle 75
- IF2 - RS232/RS485 Schnittstelle 77
- IF3 - PATA/SSI Schnittstelle 89
- IF4 - B&R Online Schnittstelle 90
- IF5 - CAN Bus (BRCOMP2-0) 91
- Schnittstelleninterrupt 85

Sicherung 17, 18
Software für BRCOMP 11
Softwarewatchdog 118
Softwarezeiten 108
Spannungsüberwachung 117
Spannungsüberwachung Batterie 104
Spannungsversorgung 17
- LED DCOK 19

SSI/PATA Schnittstelle 89
Standardsoftware für BRCOMP 11
Status-LED 20
Steuerregister $2120 58
Systemspeicherstellen 101

T

Technische Daten
- BRCOMP 12
- Relais-Aufsteckkarte 68

Timerinterruptroutinen 119

U

Uhr (Echtzeit) 107

W

Watchdog
- Hardware 118
- Software 118

X

"XFER" - Programm übertragen 103

Z

Zähler 49
- Bedienung 51
- Betriebssystemadressen 57
- Betriebssystemroutinen
  - Adressen 59
  - Aufruf 62
- CMDC - Zählerauswertung (BRCOMP) 52
- CMDD - Auswertung Zähler 0 (BRCOMP) 54
- Funktion 50, 60
- Hardwarezähler 0 50, 56, 66
- Initialisierung 61
- Initialisierungsregister $2122 59, 61
- Interrupteingang 49, 50, 56
- Interruptmodus 56
- Istwert 57, 65
- Modusregister $2121 58, 62
- Referenzeingang 49, 50, 63
- Steuerregister $2120 58
- Verdrahtungsschema 51
- Vorwahlwert 57, 64
- Zweikanalzähler 50

Zeitimpulse 106
Zeittakte 106
Zubehör
- Feldklemmen 11
- Relais-Aufsteckkarte 67
- Standardsoftware 11

Zusätzliches Anwender-EEPROM 111
Zweikanalzähler 50

STAMMHAUS

B&R ÖSTERREICH, A-5142 Eggelsberg 120
Tel: ++43 / 7748 /6586-0, Fax: ++43 / 7748 / 6586-26

VERTRIEBS- UND APPLIKATIONSZENTRALEN

B&R DEUTSCHLAND, Norsk-Data-Str. 3, D-61352 Bad Homburg, Tel: ++49 / 6172 / 4019-0, Fax: ++49 / 6172 / 457790
B&R GROSSBRITANNIEN, Milnyard Square, Orton Southgate, Peterborough, PE2 6GX, Tel: ++44 / 1733 / 371320, Fax: ++44 / 1733 / 371306
B&R ITALIEN, Viale Certosa 191, I-20151 Milano, Tel: ++39 / 2 / 3085844, 3083822, 3084358, Fax: ++39 / 2 / 38001864
B&R SCHWEIZ, Langfeldstr. 88, CH-8500 Frauenfeld, Tel: ++41 / 54 / 217425, Fax: ++41 / 54 / 217424
B&R BENELUX, Hoge Schouw 1, 4817 BZ Breda, Tel: ++31 / 76 / 5715303, Fax: ++31 / 76 / 5715306
B&R USA, 1325 Northmeadow Parkway Suite 130, Roswell, Georgia 30076, Tel: ++01 / 770 / 772-0400, Fax: ++01 / 770 / 772-0243

TECHNISCHE BÜROS (TB) IN DEUTSCHLAND

TB Bayern, Freisinger Str. 34, D-85737 Ismaning, Tel: ++49 / 89 / 996554-0, Fax: ++49 / 89 / 996554-99
TB Südwest, Friedrich-Dürr-Str. 70, D-74074 Heilbronn, Tel: ++49 / 7131 / 5971-0, Fax: ++49 / 7131 / 5971-71
TB West, Kimplerstr. 294, D-47807 Krefeld, Tel: ++49 / 2151 / 3334-5, Fax: ++49 / 2151 / 3334-60
TB Mitte, Norsk-Data-Str. 3, D-61352 Bad Homburg, Tel: ++49 / 6172 / 4019-0, Fax: ++49 / 6172 / 457790
TB Südost, Calvisius Str. 38, D-04177 Leipzig, Tel: ++49 / 341 / 44646-0, Fax: ++49 / 341 / 4419100
TB Nord, Rotenburger Str. 3, D-30659 Hannover 51, Tel: ++49 / 511 / 614077, Fax: ++49 / 511 / 612337

DISTRIBUTIONS- UND SERVICEPARTNER

DÄNEMARK, HANO Elektroteknik A/S**,** Kallerupvej 60, DK-5230 Odense, Tel: ++45 / 66 / 190600, Fax: ++45 / 66 / 190700
NORWEGEN, Kverneland Elektriske A/S**,** Bedriftsv. 6, N-4062 Klepp St, Tel: ++47 / 51 / 423840, Fax: ++47 / 51 / 423577
POLEN, CONSTEL Sp.zo.o., 40548 Katowice, ul. Brynowska 66, Tel: ++48 / 3 / 1052951, Fax: ++48 / 3 / 1052951
SPANIEN, LARRAIOZ**,** Larraioz Etxea Ind. (Apartado 193), E-20800 Zarautz (Gipuzkoa), Tel: ++34 / 43 / 140139, Fax: ++34 / 43 / 140327
PORTUGAL, TECNILAB, Av. Columbano Bordalo, Pinheiro 97 -2° Dt°, P-1000 Lisbon, Tel: ++351 / 1 / 7269129, Fax: ++351 / 1 / 7264550
FRANKREICH, GROUPE JEAMBRUN BALLUFF Dpt. ABR, 2, rue du Vallon, F-94440 Marolles en Brie, Tel: ++33 / 1 / 45692332, Fax: ++33 / 1 / 45990864
ZYPERN, TONIS G. PAVLIDES, Electrical & Mechanical Contractor, 75 c, Makarios III Ave., Limassol, Cyprus, Tel: ++357 / 5-337906, Fax: ++357 / 5-339322
WEISSRUSSLAND, ENTAS, 220113 Minsk, P.O. Box 500, Tel: ++7 / 0172 / 372598, Fax: ++7 / 0172 / 686537
PAKISTAN, INTECH, 119-B Tipu Block, Tech. Society, Lahore - 54590, Tel: ++92 / 42 / 5411724, Fax: ++92 / 42 / 5411724
AUSTRALIEN, PROCON INSTRUMENT TECHNOLOGY PTY. LTD., 365 Montague Rd., West End, Brisbane, Queensland, Australia 4101, P.O. Box 3511 Coorparoo, Qld. 4151, Tel: ++61 / 7-846 3511, Fax: ++61 / 7-846 1588
UKRAINE, SKIF ENGINEERING Ltd., Cherepovezka 17, 274028 Chernivci, Tel: ++7 / 03722 / 45207, 44024, Fax: ++7 / 03722 / 47332, 61968
SÜDAFRIKA, PROLOCON, P.O. Box. 130658, Bryanston, Transvaal, 2021, Tel: ++27 / 11 / 7062896, Fax: ++27 / 11 / 4636455
SLOWENIEN, ISKRA ROBOTEC, Stegne 13, 61000 Ljubljana, Tel: ++386 / 61 / 575054, Fax: ++386 / 61 / 572247
SLOWAKEI, URAP AUTOMATIZACIA, Po Box B152, 02141 Zilina, Tel: ++42 / 89 / 622070, Fax: ++42 / 89 / 42340

B&R COMPACT PLC
USER'S MANUAL

Version: 2.00 (December 1995)
Published by: Bernecker und Rainer Industrie-Elektronik GmbH.
Model No.: MABRCOMP1-0E

The information contained herein is believed to be accurate as of the date of publication, however, Bernecker und Rainer Industrie-Elektronik Ges.m.b.H. makes no warranty, expressed or implied, with regards to the products or the documentation contained within this book. Bernecker und Rainer Industrie-Elektronik Ges.m.b.H. shall not be liable in any event for incidental or consequential damages in connection with or arising from the furnishing, performance or use of these products.

B&R COMPACT PLC

Contents:

1. General Information 135

2. Model Number 136
   2.1 B&R Compact PLC 136
   2.2 Relais-Aufsteckkarte 137
   2.3 Standardsoftware 137
   2.4 Feldklemmen 137

3. Technical Data 138

4. Measurements 139

5. Installation Mounting 140
   5.1 Installation Guidelines 140
   5.2 There are two Installation Possibilities 140
      5.2.1 Directly to Control Cabinet Wall 140
      5.2.2 Rail Mount 140

6. Description of Components 142

7. Power Supply 143

8. Fuse 144

9. LEDs 145

10. Digital Inputs 147
   10.1 Technical Data 147
   10.2 Input Circuit 148
   10.3 Digital Input Wiring 148

11. Digital Outputs 149
   11.1 Technical Data 149
   11.2 Output Circuit 150
   11.3 Wiring 150
   11.4 Switching Resistive-Inductive Load 151

12. Analog Inputs 153
12.1 Technical Data 156
12.2 Pin-outs 157
12.3 Input Circuit 157
12.4 Shield Grounding 158
12.5 Wiring 158
12.6 Software Operation 161
12.7 AINJ - Analog Input Compact PLC 164
12.8 Analog Conversion 167

13. Analog Outputs 169
13.1 Technical Data 169
13.2 Pin-outs 170
13.3 Output Circuit 170
13.4 Grounding Your Cable Shielding 171
13.5 Wiring 171
13.6 Software Operation 172
13.7 AOTF - Analog Output Compact PLC 173
13.8 Analog Conversion 174

14. Counter Inputs and Interrupt Input 175
14.1 Wiring 177
14.2 Using the Counters 177
14.3 CMDC - Counter Module Driver (BRCOMP) 178
14.4 CMDD - 0 Counter Module Driver (BRCOMP) 180
14.5 Interrupt Mode 182
14.6 Operation in STL 183
14.6.1 Counter Operating System Addresses 183
14.6.2 Control Register $2120 184
14.6.3 Mode Register $2121 184
14.6.4 Initialization Register $2122 185
14.6.5 Addresses of Operating System Routines 185
14.6.6 Function of the Counter 186
14.6.7 Counter Initialization 187
14.6.8 Calling Operating System Routines 188

15. Relay expansion card for BRCOMP 193
15.1 General Information 193
15.2 Relay Groups 193
15.3 Technical Data 194
15.4 Pin Assignments 194
15.5 Measurements 195
15.6 Installation Guidelines 196

15.7 Mounting 196
15.8 Compact PLC and Relay Expansion Card 198
15.9 Connection to the Compact PLC 199
15.10 Relay Supply 199
15.11 Switching Circuit 200

16. IF1 - RS232 Interface 201
16.1 Technical Data 201
16.2 Pin-outs 201
16.3 Wiring 202
16.4 Grounding and Shielding 202
16.5 Software Operation 202
16.6 Compact MMI P120 and MMI P121 202

17. IF2 - RS232/RS485 Interface 203
17.1 Technical Data 203
17.2 Pin-outs 203
17.3 Wiring 204
17.4 Grounding and Shielding 205
17.5 Software Operation 205

18. Using Interfaces IF1 and IF2 206
18.1 Using Interface interrupts 211

19. IF3 - PATA/SSI Interface 215

20. IF4 - B&R Online Interface 216

21. IF5 - CAN Bus 217
21.1 Merkmale des CAN Busses 217
21.2 B&R und CAN 217
21.3 Buslänge und Kabeltyp 217
21.4 Anschlussbelegung 219
21.5 Verdrahtung 219
21.5.1 Verdrahtungsprinzip 220
21.5.2 Blockschaltbild eines Knotens 220
21.6 Bedienung des CAN Busses 221

22. Power-On Sequence 222

23. Command Set 222

24. Mathematic routines 223

25. System Registers and Flags 227

26. First Scan Flag 229

27. Lithium Battery 230
28. Time Cycles 232
29. Time pulses 232
30. Real-time Clock 233
31. Software Timers 234
32. Inport/Outport Address $3400 236
33. Additional User EEPROM 237
33.1 Reading Data 238
33.2 Writing Data 240
34. Inport Address $3480 243
35. Watchdog 244
35.1 Software Watchdog 244
35.2 Hardware Watchdog 244
36. Timer Interrupt Routines 245
37. Error Messages 247
38. Index 249

1. GENERAL INFORMATION

The MINI/MULTICONTROL family is offered by B&R as control system which can be adapted to all customer specific wishes because of its modular structure and 100% software compatibility.

To enhance this list, B&R has now expanded the product palette with a new Compact PLC. The compact is basically a controller which settles into the family just under the MINICONTROL. Again, 100% software compatibility to the MINICONTROL was called for. Therefore, if in the future an upgrade is required, the MINICONTROL can be implemented without losing your software.

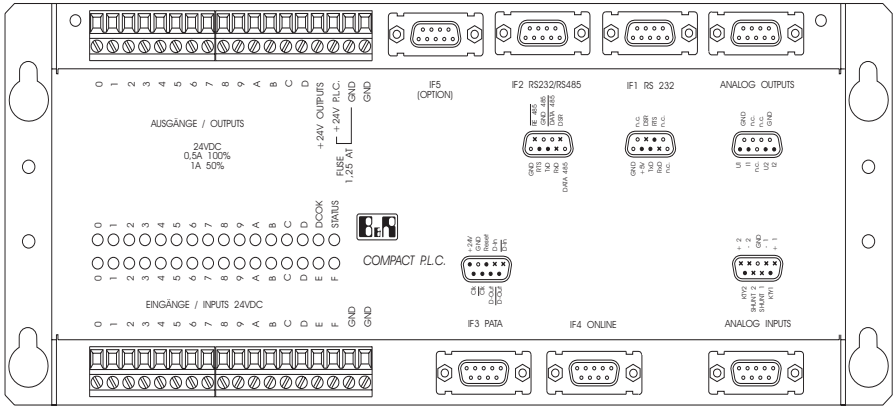
Even though the Compact PLC is priced so low, because we have kept the previous modularity all in one unit, the Compact PLC is in concept and construction able to be used universally.

Construction

- Digital Inputs/Outputs
- Relay expansion card
- Analog Inputs (voltage, current and temperature)
- Analog Outputs (voltage and current)
- Event counter
- Interrupt input
- Reference input
- Channels A and B for positioning applications
- 2 serial interfaces
- CAN Bus (BRCOMP2-0)
- B&R standard PATA interface (for MINICONTROL operator interface panel)
- SSI interface für connection of absolute encoders
- B&R on-line interface
- 16 Kbytes EEPROM application memory
- 16 Kbytes additional EEPROM
- RS485 Network (B&R MININET)
- Operator panels: MINICONTROL Operator Interface Panel
Compact MMI P120 and MMI P121
and other products from the PANELWARE family

2. MODEL NUMBER

2.1 B&R COMPACT PLC



Compact PLC, with 6303 processor, 16 Kbytes application memory, approx. 4 msec / K instructions, 16 digital inputs, 14 digital outputs, 2 analog inputs, 2 analog outputs, channels A and B for positioning tasks, reference input, event counter, interrupt input, B&R on-line interface, RS232 interface (not galvanically isolated), RS232/RS485 interface (RS485 galvanically isolated), PATA/SSI interface,

without CAN Bus	**BRCOMP1-0**
with CAN Bus on IF5	**BRCOMP2-0**

2.2 RELAY EXPANSION CARD

A relay expansion card is available for the Compact Control which adds an additional 16 relay outputs to the 14 transistor outputs (O 060 - O 06F).

The relay expansion card is described in section "15. Relay Expansion Card for BRCOMP".

2.3 STANDARD SOFTWARE

A diskette with the most important standard software has been put together especially for the Compact Control. The following software is saved on the diskette:

- Standard Utility Programs (standard software package 1)
- Operator Panel Software
- B&R MININET
- Positioning

Language	Model Number
German	SWSPSBRC01-0
English	SWPLCBRC01-0

2.4 TERMINAL BLOCKS

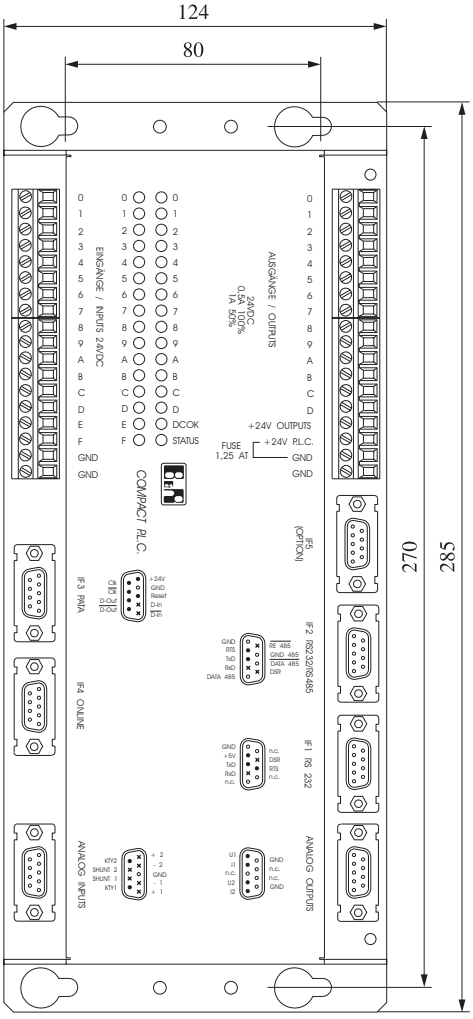
A set of PHOENIX terminal blocks can be ordered (BRTB0218-0) as an accessory. Contained in delivery:

Amount	Terminal Blocks
2	8 Pin
2	10 Pin

3. TECHNICAL DATA

Processor	6303
Processing time	approx. 4 msec / K instructions
User program memory	16 Kbytes RAM/EEPROM
Expansion memory	16 Kbytes EEPROM (for data)
Status-LED	Red
Number of registers Non-volotile Volotile	7168 7148 20
Number of flags Non-volotile Volotile	800 300 500
Time/date	Real-time clock
Software timers	64
Clock pulses	10 msec, 100 msec, 1 sec, 10 sec
Digital inputs	16 of which four can be used as counter inputs
Digital outputs	14
Analog inputs	2
Analog outputs	2
Serial interface IF1 IF2 IF3 IF4 IF5	 RS232 RS232/RS485 (RS485 galvanically isolated) PATA (MINICONTROL operator interface panel or relay expansion card) SSI (connecting absolute encoders) B&R on-line interface CAN Bus (BRCOMP2-0)
Supply voltage	24 VDC ±25 %
Max. power consumption without operator panels at 18 V at 24 V at 30 V	 6.0 W 6.5 W 7.5 W
Fuse	T 1.25 A / 250 V
Soft and Hardware watchdogs	YES
Battery monitor	YES
Operating temperature	5 to 55 °C
Relative humidity	30 - 95 % non-condensing

4. MEASUREMENTS



5. INSTALLATION MOUNTING

5.1 INSTALLATION GUIDELINES

Make sure that devices that create heavy electromagnetic disturbances (e.g. frequency converter, transformer, motor controller etc.) are at a sufficient distance. The distance from these devices to the PLC should be as large as possible. If necessary, they are to be separated with a magnetic shielding partition (VACOPERMâ 70).

5.2 THERE ARE TWO INSTALLATION POSSIBILITIES

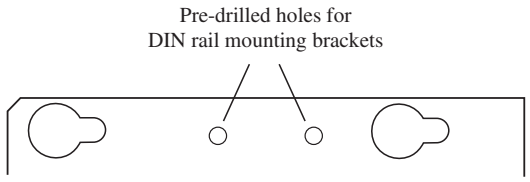
5.2.1 Directly to Control Cabinet Wall

The Compact PLC can be mounted horizontally or vertically. If it should be mounted vertically, be sure that the digital inputs/outputs are at the top (see drawing on the next page).
The chassis should be screwed firmly to the wall of the cabinet using the four screw holes provided for vertical mounting.

M5 screws are to used in mounting the Compact PLC (if M5 screws are not available, use appropriate replacement screws) (hole dimensions: 80 * 270 mm or 3.14 * 10.62 inches).

5.2.2 Rail Mount

For mounting the unit to a standard mounting rail (DIN EN 50022-35) two mounting brackets are provided in the delivery which must be screwed onto the Compact PLC.

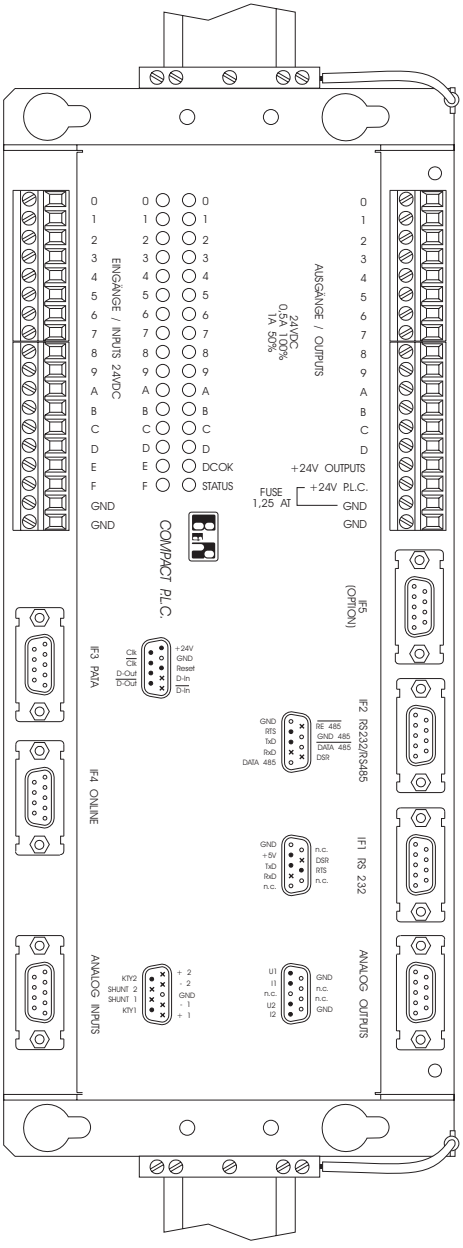


Fastening the Compact PLC to the DIN rail.

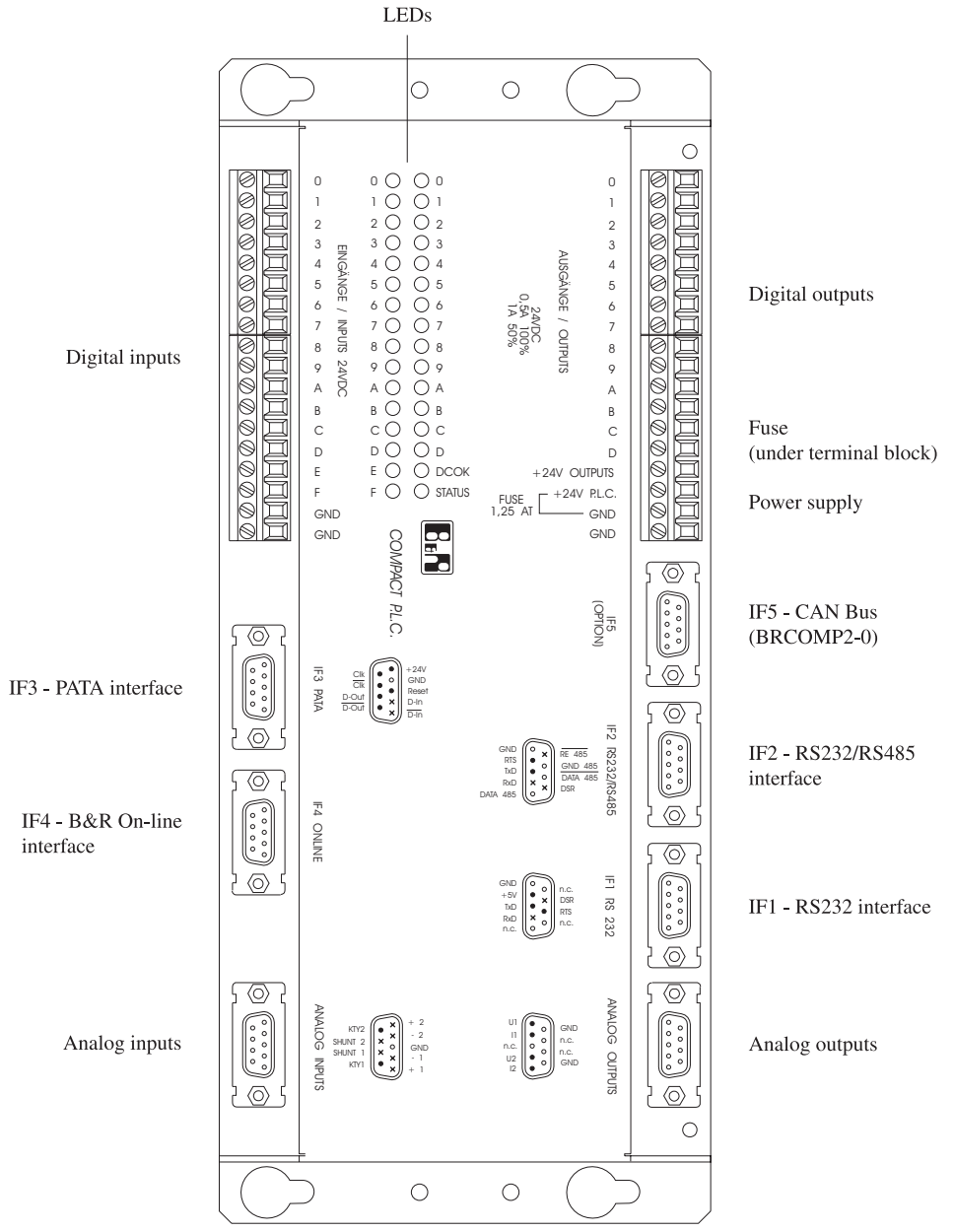
Cable clamps should be fastened to the left and right of the Compact PLC. The unit is to be grounded with the attachment provided (see drawing on the next page).

NOTE: The DIN rail must also be grounded.

The width of the mounting plates which are attached to the unit provide enough space between the Compact PLC and neighboring modules for ventilation.



6. DESCRIPTION OF COMPONENTS



7. POWER SUPPLY

The LED DCOK indicates that the control is being supplied with power. There are two different ways to supply the Compact PLC:

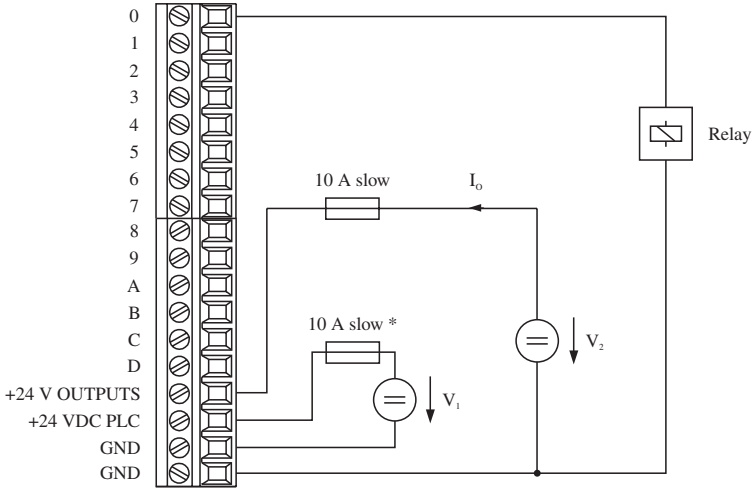
- One power supply: - Combinational supply for inputs and outputs
- Two power supplies: - One for the control
  - One for the digital outputs

One Power Supply

A single Compact PLC supply is standard (For wiring see section "11. Digital Outputs").

Two Power Supplies

Control and outputs are supplied separately.



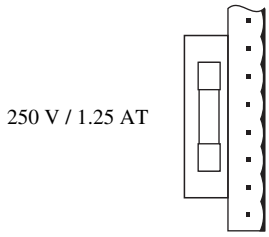
V_1 ... 24 VDC / 1A
V_2 ... 24 VDC / I_O
I_O ... Total current of all outputs

* 10 A fuse (slow) for protecting the connector in case of a short circuit or reverse poles.

8. FUSE

The Compact PLC is protected with a 250 V / 1.25 AT fuse. The fuse compartment can be found under the terminal block for the digital outputs.

NOTE: The digital outputs are supplied separately. The supply must be fed through a separate fuse (see section "7. Voltage Supply" and "11. Digital Outputs").



9. LEDs

32 LEDs inform the user of digital input status, digital output status, power supply and the CPU activity.

0 ○ ○ 0
1 ○ ○ 1
2 ○ ○ 2
3 ○ ○ 3
4 ○ ○ 4
5 ○ ○ 5
6 ○ ○ 6
7 ○ ○ 7
8 ○ ○ 8
9 ○ ○ 9
A ○ ○ A
B ○ ○ B
C ○ ○ C
D ○ ○ D
E ○ ○ DCOK
F ○ ○ STATUS

LEDs for	Description
Digital inputs	The green LEDs (0 - F) on the left-hand side of the unit indicate the status of the digital inputs. Off ... Input = "0" On ... Input = "1"
Digital outputs	The orange LEDs (0 - D) indicate the status of the digital outputs. Off ... Output = "0" On ... Output = "1"
Supply	The orange LED DCOK indicates, whether power is being supplied or not. Off ... No supply On ... Supply voltage OK.

Status-LED

The Compact PLC is equipped with a red status LED which is used to indicate the various operating states.

LED-Status	Operating status
HI LO	User program is running in RAM
HI LO	CPU in HALT status
HI LO	On-line unplugged during PROM programming
HI LO	Error during user program execution
HI LO	User program is running in PROM

10. DIGITAL INPUTS

The digital inputs convert the binary signals of the process in binary values 0 and 1. States of the inputs are indicated by means of the green LEDs.

The Compact PLC is equipped with 16 digital inputs (I 040 - I 04F). Inputs I 04C - I 04F can also be used for special tasks:

- Event counter
- Interrupt input
- Reference input
- Incremental encoder (channels A and B) for positioning tasks

The functions are explained in more detail in section "14. Counter Inputs and Interrupt Input".

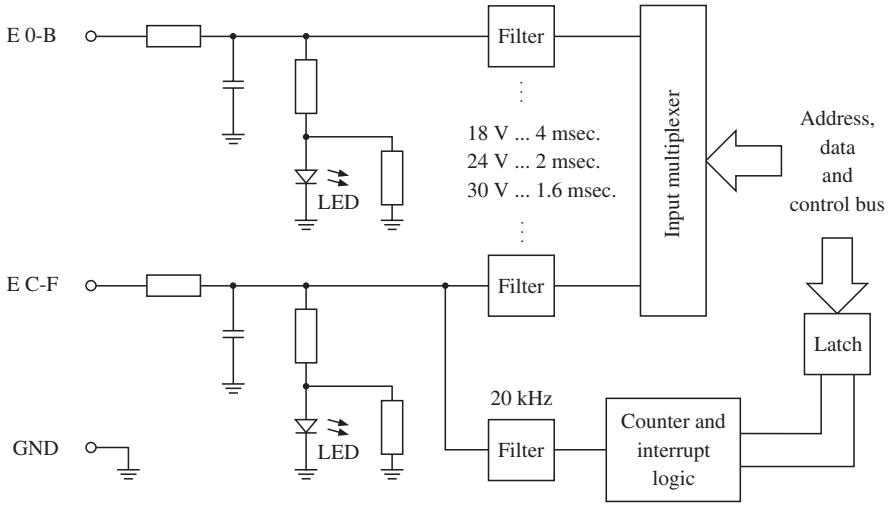
10.1 TECHNICAL DATA

Amount	16
Revision	Type 1 conforming to IEC 1131-2
Status display	16 green LEDs
Electrical isolation Input ↔ PLC Input ↔ Input	 NO NO
Input voltage [1] Minimal Nominal Maximal	 15 VDC 24 VDC 30 VDC
Input resistance	4 kΩ
Switching threshold log. 0 → 1 log. 1 → 0	 Min. 15 VDC Max. 5 VDC
Input current at 24 VDC	Approx. 5 mA
Switch delay inputs 0 - F (log. 0 → 1, log. 1 → 0)	18 V: Max. 4 msec. 24 V: Max. 2 msec. 30 V: Max. 1.6 msec.
Input frequency inputs C - F	Max. 20 kHz
Update of input data by CPU	Automatically with any change
Maximum peak voltage	500 V for 50 µsec, max. every 100 msec [2]

[1] If using inputs C - F as counter inputs, the input voltage should be 24 VDC ± 10 %, so that the maximum input frequency of 20 kHz can be reached.

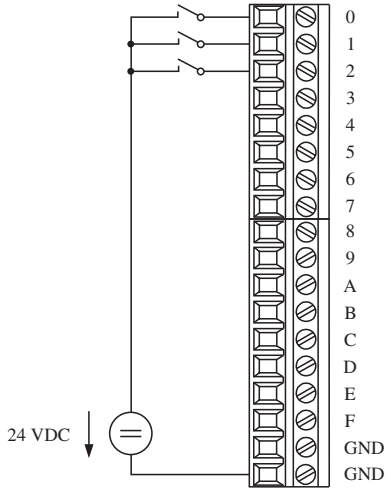
[2] Standard pulse 1.2/50 (IEC 60-2)

10.2 INPUT CIRCUIT



Counter and interrupt logic is described in section "14. Counter Inputs and Interrupt Input".

10.3 DIGITAL INPUT WIRING



11. DIGITAL OUTPUTS

Digital outputs are for controlling external loads (Relays, Motors, Magnetic valves, etc.). The status of the outputs is indicated with orange LEDs.

The Compact PLC is equipped with 14 transistor outputs (O 050 - O 05D).

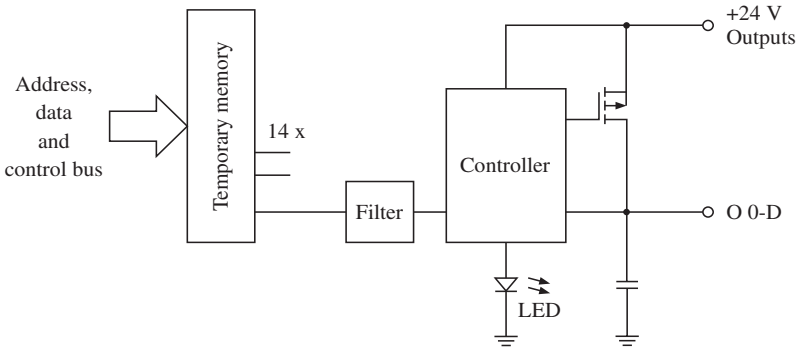
11.1 TECHNICAL DATA

Amount	14
Status display	14 orange LEDs
Electrical isolation Output ↔ PLC Output ↔ Output	 NO NO
Supply voltage Minimal Nominal Maximal	 18 VDC 24 VDC 30 VDC
Switching current 50 % Simultaneousness 100 % Simultaneousness	 1.0 A[1)] 0.5 A
Switching delay log. 0 → 1 log. 1 → 0	 Approx. 200 µsec. Approx. 200 µsec.
Residual voltage	< 1 V at 1 A
Protection circuit [2)]	Internal
Short circuit proof	YES
Overload protection	Automatic shut-off by thermal overload

[1)] **NOTE:** If current flows at 50 % simultaneousness the maximum allowed environmental temperature is of 40 °C.

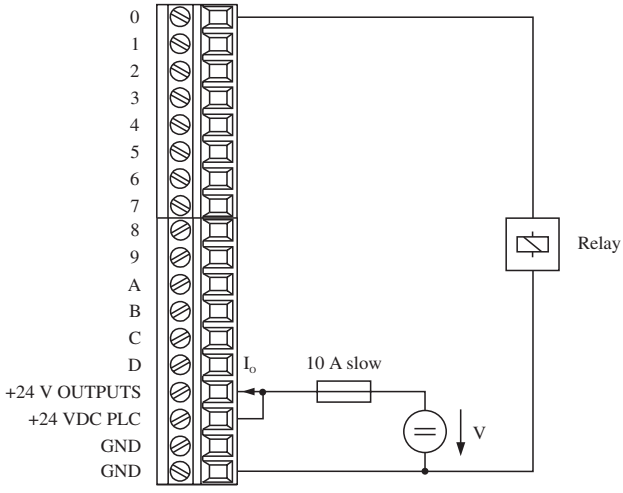
[2)] In some circumstances an external protection circuit may be required (see section "11.4 Switching Resistive-Inductive Loads").

11.2 OUTPUT CIRCUIT



11.3 WIRING

Another wiring diagram is shown in section "7. Power Supply".

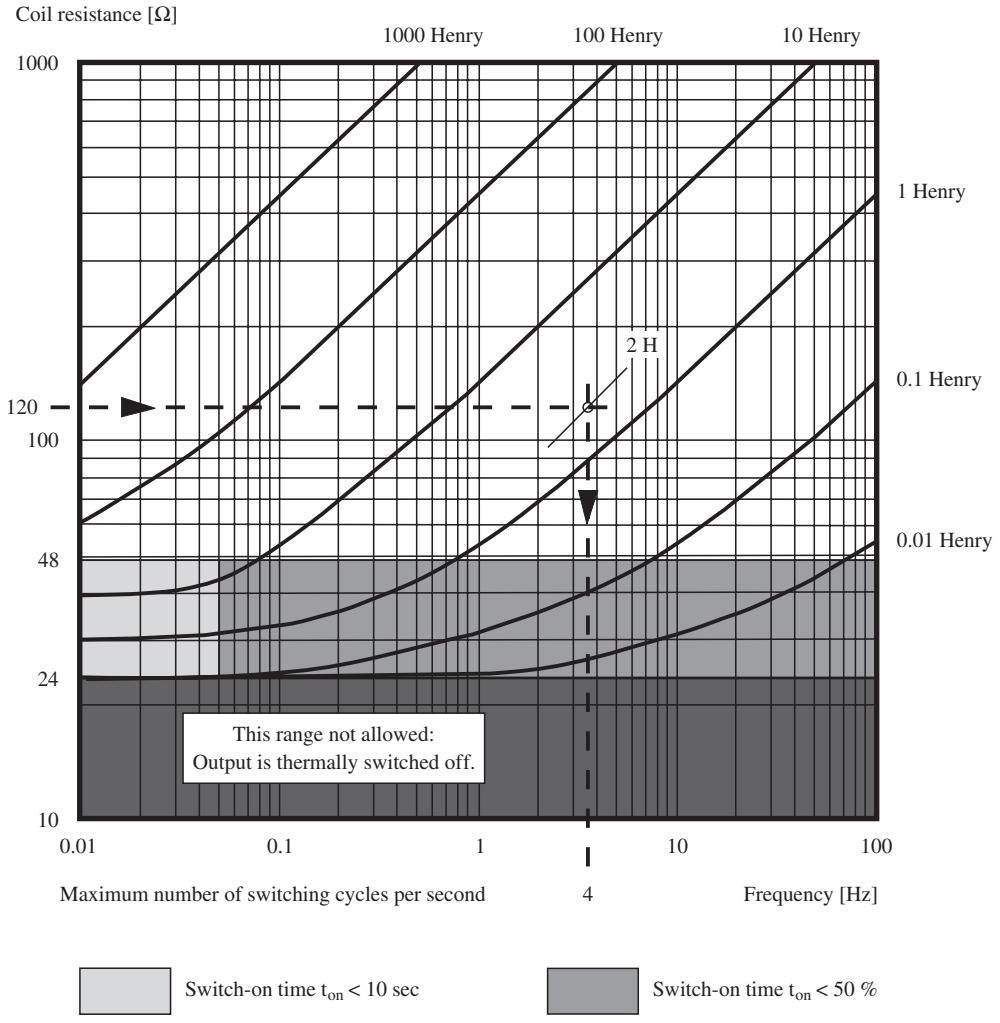


V ... 24 VDC / 1 A + I_O

I_O ... Total current of all outputs

11.4 SWITCHING RESISTIVE-INDUCTIVE LOAD

When switching resistive-inductive loads, attention must be paid to coil resistance and to coil inductivity.
The diagram shown below indicates the maximum number of switching cycles per second in connection with the coil resistance and coil inductivity. All outputs are switched simultaneously (0.5 A at 24 V).



If the coil resistance is less than 24 Ω, the thermal protection unit takes over and switches the output off.

At a coil resistance of between 24 Ω and 48 Ω the switch-on tiem should be paid attention to. Depending on the switching frequency, it may not be longer than 10 seconds or not longer than the switch-off time. Continued operation is not possible in this range!
If, for example, one switching cycle per second is executed, the switch-on time may not be longer than 0.5 sec.

Example With which maximum switching frequency can a magnetic valve be switched with the following coil data?

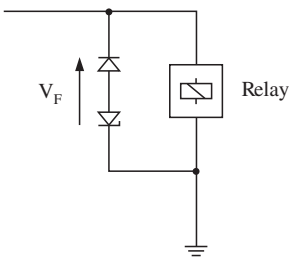
Inductivity:	2 H
Coil resistance:	120 Ω

Find your coil resistance on the Y axis and follow the horizontal line to the right until you reach the intersecting parameter curve for 2 H (see diagram). At this intersection the vertical line to the x axis can be found. Follow this line to the x axis and read the maximum switching frequency (pay attention to the logarithmic scale).
The magnetic valve can be operated at a maximum switching frequency of 4 Hz.

Operation with a higher switching frequency is not allowed and can lead to a thermal shutdown of the output.
If a higher switching frequency is required, an external protection circuit must be implemented. Most manufacturers of relays and magnetic valves provide protective circuits for the respective elements.

The inverse voltage V_F must be ≤ 8 V.

Example of a protection circuit



12. ANALOG INPUTS

With analog inputs, measurement values (current, voltage, resistance or temperature) are converted to numerical values which can be processed by the PLC.

The B&R Compact Control is equipped with two analog inputs. One of the following signals can be selected with software for each channel. The selection is made by entering a mode number (0 - 9).

Measurement	Mode	Input Signal	Software Filter Resolution at			Value
			10 Hz / 50 Hz ±14 Bit	250 Hz ±12 Bit	1 kHz ±8 Bit	
Voltage	0	±10 V	±0.610 mV	±2.441 mV	±39.06 mV	±32767
	1	±2.5 V	±0.152 mV	±0.610 mV	±9.76 mV	±32767
Current	2	0 - 20 mA	1.221 µA	4.883 µA	78.12 µA	0 - 32767

Measurement	Mode	Input Signal	Resolution	Value
Temperature	3	KTY10 (-50 °C to +150 °C)	0.01 °C	-5000 to +15000
	4	KTY10 (-58 F to +302 F)	0.01 F	-5800 to +30200
reserved	5	--	--	
	6	--	--	
	7	--	--	
Resistance	8	Resistance measurement (0 - 65000 Ω) [1]	1 Ω	0 - 65000
	9	Resistance measurement (0 - 4500 Ω) [1]	0.1 Ω	0 - 45000

[1] Resistance Measurement

The resistance measurement can only be made with the standard function block AINJ (see section "12.7 AINJ - Analog Input Compact Control"). The resolution Ω/bit can be seen on the diagrams on the following pages.

Temperature

Temperature measurement with sensors such as NTC, PTC, PT1000 etc. is possible with the resistance measurement (see section "12.7 AINJ - Analog Input Compact PLC").

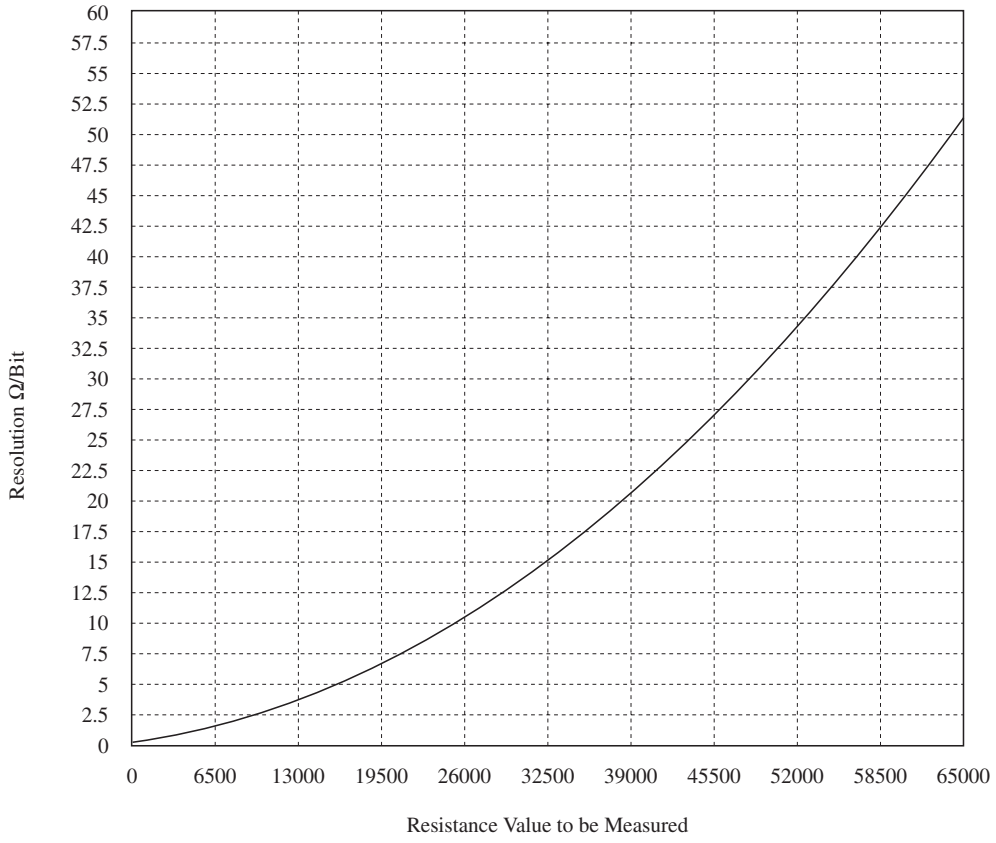
Utilities: ALIN, TLIN for nonlinear characteristic curves
SCAL, TSCL for linear characteristic curves

Diagram for Resistance Measurements

The resolution Ω/bit for a certain resistance value can be read from the following diagrams.

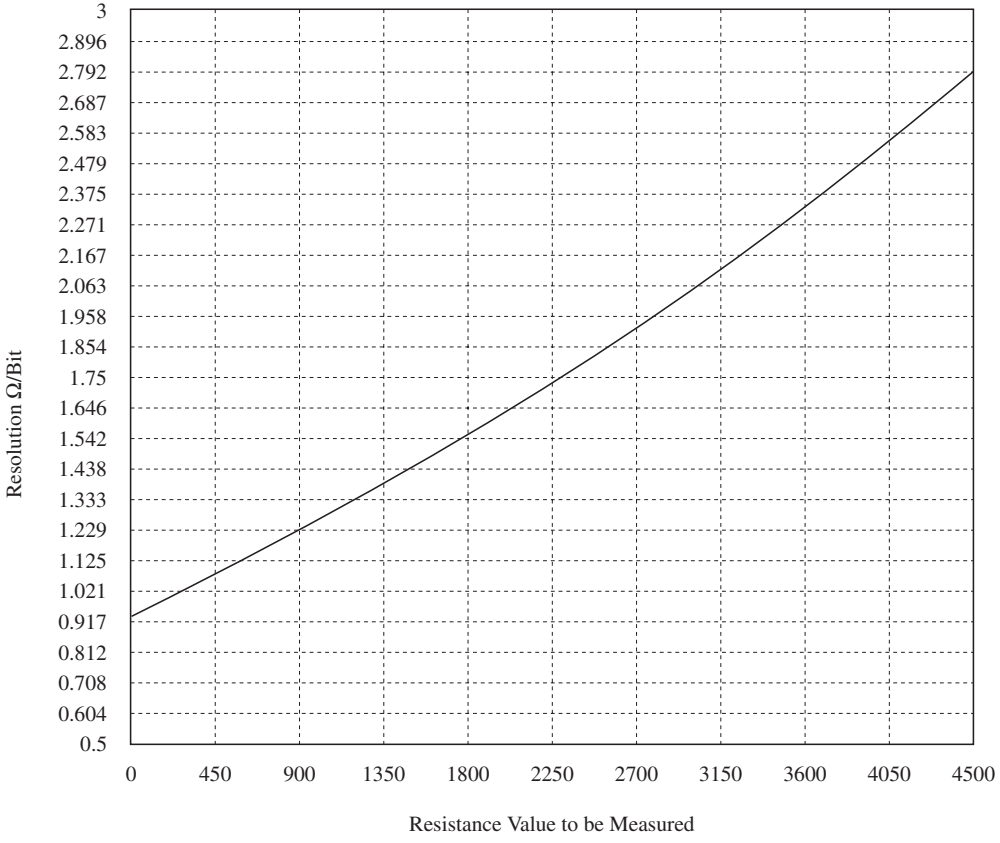
1) Resistance measurement 0 - 65000 Ω:

Mode: 8



2) Resistance Measurement 0 - 4500 Ω:

Mode: 9



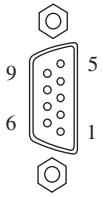
12.1 TECHNICAL DATA

Amount	2
Galvanic isolation	Differential inputs, not galvanically isolated
Common mode voltage	±12 V
Input resistance	min. 10 MΩ
Dielectric strength	±30 V max.
Resolution A/D converter	16 Bit
Addressing	Operating system
Precision at 25 °C	Software compensation to ±0.01 % (10 Hz Notch)
Temperature drift	±10 V range: ±0.03 % / °C ±0.9 LSB / °C [1)] ±2.5 V range: ±0.02 % / °C ±3.5 LSB / °C
Hardware filter Cutoff frequency Rolloff Step response	 Approx. 110 Hz Approx. 20 dB/Dec. 63 % in 1.4 msec
Software filter Cutoff frequency (1. Notch)	 10 Hz, 50 Hz, 250 Hz, 1 kHz, software selectable
Conversion times	302 msec (10 Hz), 62 msec (50 Hz), 16.2 msec (250 Hz), 4.1 msec (1 kHz)
Calibration times	902 msec (10 Hz), 183 msec (50 Hz), 48 msec (250 Hz), 11,7 msec (1 kHz)
Damping 1st order notch	>100 dB
Operating mode	Trigger (a calibration is performed after every notch frequency change and after a reset. After a reset a notch frequency of 50 Hz is set automatically again)
Input voltage	±10 V / ±2.5 V software selectable
Precision voltage input	appr. ±14 bit (10 Hz & 50 Hz), appr. ±12 bit (250 Hz), appr. ±8 bit (1 kHz)
Current measurement	an internal shunt must be used (124 Ω)
Precision for 0 - 20 mA	appr. 14 bit (10 Hz & 50 Hz), appr. 12 bit (250 Hz), appr. 8 bit (1 kHz)
KTY10 Temperature sensor Range of measurement Linearization Resolution	 -50 to +150 °C Hardware to ±0.3 °C from -10 to +110 °C 0.01 °C

[1)] LSB ... Least Significant Bit

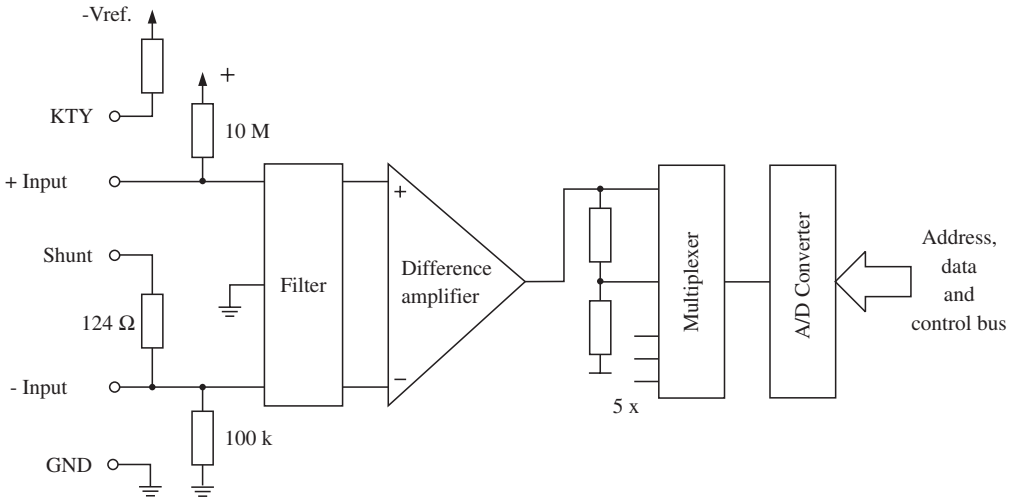
12.2 PIN-OUTS

9 pin D-type (female)



Pin	Assignment
1	+ Input 1
2	- Input 1
3	GND
4	- Input 2
5	+ Input 2
6	KTY 1
7	Shunt 1
8	Shunt 2
9	KTY 2

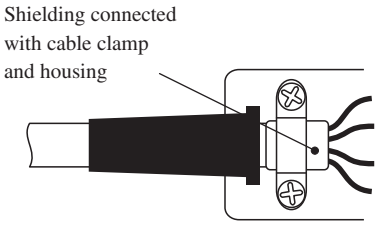
12.3 INPUT CIRCUIT



12.4 SHIELD GROUNDING

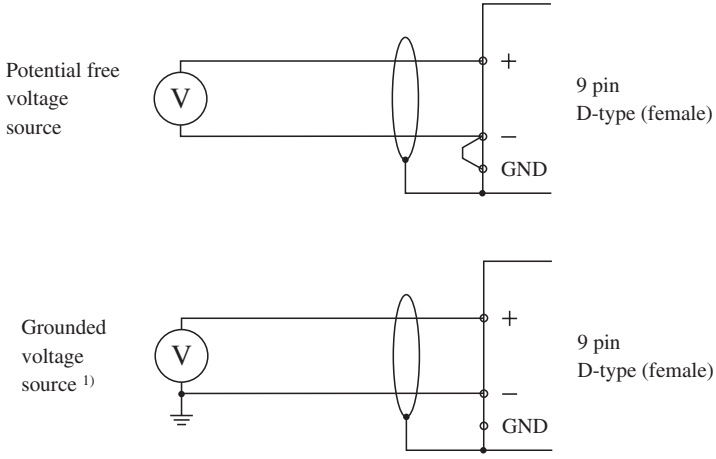
Metal screws or a metallic connector housing are the easiest and most efficient methods of diverting any possible disturbances to the housing of the Compact PLC.

The cable shielding should be in direct contact with the connector housing.



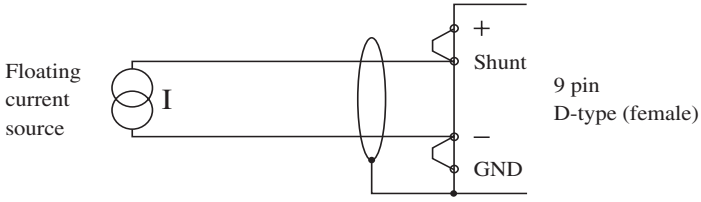
12.5 WIRING

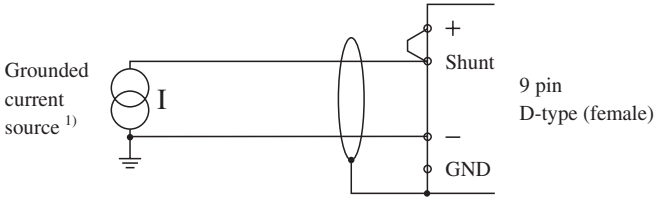
Voltage source



1) No voltage dropout monitoring is possible for this connection.

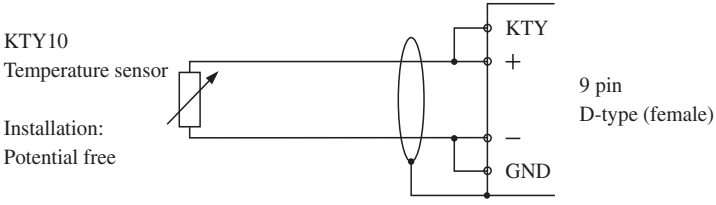
Current source





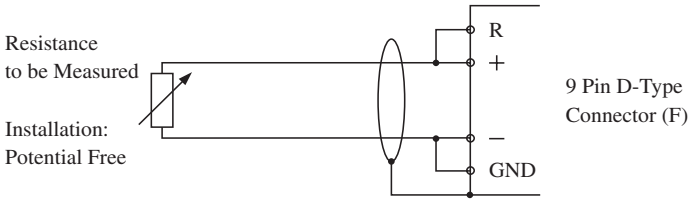
KTY10 sensor

The KTY sensors are semiconductor temperature sensors. The KTY10 series has a nominal resistance of 2000 Ω at 25 °C (e.g. KTY10-6, KTY11-6, KTY13-6, KTY16-6 and KTY19-6).



1) No current dropout monitoring is possible for this connection.

Resistance measurement with NTC, PTC, PT1000 etc.

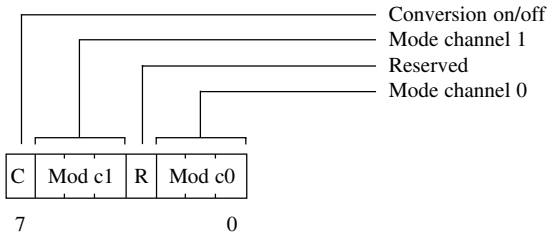


12.6 SOFTWARE OPERATION

The initialization of the analog input is handled through registers R 1000 - R 1003. The analog value is either converted in the END routine or by an STL call. The converted values are then stored in registers R 1004 - R 1007.

The initialization can either be done with commands in STL or with the help of the AINJ function block (see section 12.7).

Register R 1000

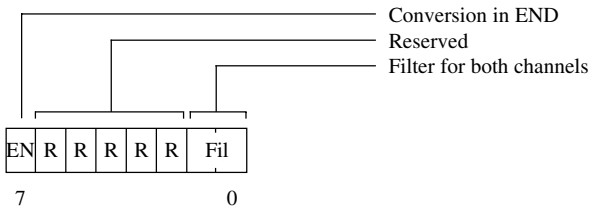


Conversion on/off	Switches the analog conversion on/off. If it is switched off, registers R 1004 - R 1007 contain a value of 0. 0 ... Conversion off 1 ... Conversion on
Mode channel 1	Bits 4 - 6 are used to set the mode for channel 1.

Mode	Signal
0	±10 V
1	±2.5 V
2	0 - 20 mA
3	KTY10 (Temperature in °C)
4	KTY10 (Temperature in F)

Reserved	Reserved for later use.
Mode channel 0	Bits 0 - 2 are used to set the mode for channel 0. The same modes can be set as for channel 1.

Register R 1001



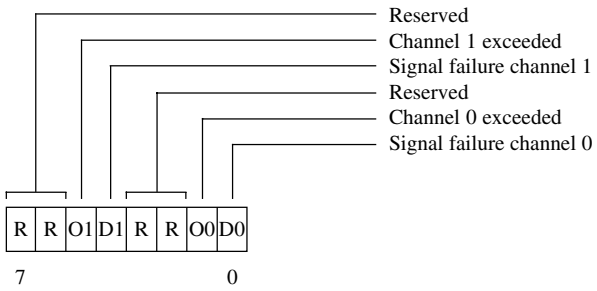
Conversion in END	Defines whether the conversion should be done during the END routine execution or not. If the conversion is not done during the END routine, the analog conversion routine must be started with an STL call (see Analog Conversion). 0 ... Conversion in the END routine 1 ... Conversion with STL call
Reserved	Reserved for later use.
Filter for both channels	Defines the filter for both channels.

Filter	Frequency	Conversion time
0	10 Hz	302 msec.
1	50 Hz	62 msec.
2	250 Hz	16.2 msec.
3	1000 Hz	4.1 msec.

Register R 1002

Conversion frequency channel 1	Defines the conversion frequency of channel 1 (0 - 255). 0 ... Channel 1 is never converted 1 ... Channel 1 is always converted to channel 0 2 ... Channel 1 is converted to double channel 0 : ... : : ... : 255 ... Channel 1 is converted to 255 times channel 0

Register R 1003



Reserved	Reserved for later use.
Channel 1 exceeded	1 ... Range exceeded for channel 1
Signal failure channel 1	1 ... Signal failure channel 1
Reserved	Reserved for later use.
Channel 0 exceeded	1 ... Range exceeded for channel 0
Signal failure channel 0	1 ... Signal failure channel 0

Registers R 1004 - R 1007

Destination memory

In registers R 1004 - R 1007, the converted values are stored.

Memory	Channel
R 1004&	0
R 1006&	1

Value range:

Signal	Value
±10 V	-32768 to +32767
±2.5 V	-32768 to +32767
0 - 20 mA	0 to 32767
-50.00 to +150.00 °C	-5000 to +15000
-58.00 to +302.00 F	-5800 to +30200

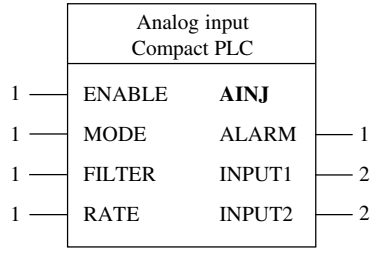
12.7 AINJ - ANALOG INPUT COMPACT PLC

The function block is used for initializing the analog inputs. The converted values and the alarm bits for range exceeding and Signal failures are sent to outputs.

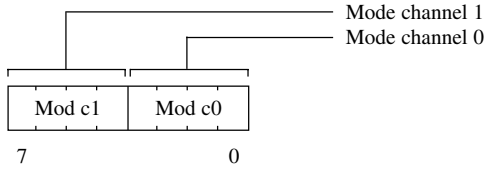
NOTE: The analog conversion is always done in the END routine.

AINJ

Function block length: 84 lines
First Scan: No



Inputs/Outputs	Description	Address types
ENABLE	If the ENABLE input is set to 1, the function block is executed.	R, F, #
MODE	Channel mode setting.	R, #



Mode	Signal
0	±10 V
1	±2.5 V
2	0 - 20 mA
3	KTY10 (Temperature in °C)
4	KTY10 (Temperature in F)
5	reserved
6	reserved
7	reserved
8	Resistance measurement (0 - 65000 Ω) [1]
9	Resistance measurement (0 - 4500 Ω) [1]

[1] Version 1.30 and higher

<table>
<tr><th>Inputs/Outputs</th><th>Description</th><th>Address types</th></tr>
<tr><td>FILTER</td><td>Defines the filter for both channels.
<table>
<tr><th>Filter</th><th>Frequency</th><th>Conversion time</th><th>Resolution</th></tr>
<tr><td>0</td><td>10 Hz</td><td>302 msec.</td><td>±14 bit</td></tr>
<tr><td>1</td><td>50 Hz</td><td>62 msec.</td><td>±14 bit</td></tr>
<tr><td>2</td><td>250 Hz</td><td>16.2 msec.</td><td>±12 bit</td></tr>
<tr><td>3</td><td>1000 Hz</td><td>4.1 msec.</td><td>±8 bit</td></tr>
</table></td><td>R, #</td></tr>
<tr><td>RATE</td><td>Defines the conversion time of channel 1 (0 - 255).

0 ... Channel 1 is never converted
1 ... Channel 1 is always converted to channel 0
2 ... Channel 1 is converted to double channel 0
: ... :
: ... :
255 ... Channel 1 is converted to 255 times channel 0</td><td>R, #</td></tr>
<tr><td>ALARM</td><td>indicates a signal failure or an exceed on a channel.

Channel 1 exceeded
Signal failure channel 1
Channel 0 exceeded
Signal failure channel 0

| | | O1 | D1 | | | O0 | D0 |
7 0</td><td>R</td></tr>
<tr><td>INPUT0</td><td>Contains the converted value of channel 0.

Value range:
<table>
<tr><th>Signal</th><th>Value</th></tr>
<tr><td>±10 V</td><td>-32768 to +32767</td></tr>
<tr><td>±2.5 V</td><td>-32768 to +32767</td></tr>
<tr><td>0 - 20 mA</td><td>0 to 32767</td></tr>
<tr><td>-50.00 to +150.00 °C</td><td>-5000 to +15000</td></tr>
<tr><td>-58.00 to +302.00 F</td><td>-5800 to +30200</td></tr>
<tr><td>0 - 65000 Ω</td><td>0 to 65000</td></tr>
<tr><td>0 - 4500 Ω</td><td>0 to 4500</td></tr>
</table></td><td>R</td></tr>
</table>

Inputs/Outputs	Description	Address types
INPUT1	Contains the converted value from channel 1. Value range:	R

Signal	Value
±10 V	-32768 to +32767
±2.5 V	-32768 to +32767
0 - 20 mA	0 to 32767
-50.00 to +150.00 °C	-5000 to +15000
-58.00 to +302.00 F	-5800 to +30200
0 - 65000 Ω	0 to 65000
0 - 4500 Ω	0 to 4500

12.8 ANALOG CONVERSION

NOTE: The values in registers R 1004 - R 1007 are zero for the first program cycle.

If bit 7 of R 1000 (Mode register) is not set, no analog conversion is performed (R 1004 - R 1007 = 0).

1) Conversion in the END routine

Software	Conversion
FBK	The conversion is always done during the END routine.
STL	If bit 7 of register R 1001 is set, the conversion is done in the END routine.

2) Direct Jump to Pointer in Operating System

Software	Conversion
FBK	This call is not possible when using the AINJ function block.

STL	Instead of the conversion being done in the END routine, you can jump directly from the program to a pointer in the operating system, which references the analog conversion routine (Pointer: $C10E).
	
	Analysis of converted channels: Carry = 0 ... Conversion ended
	Accu B ... Converted channel
	
	Example
	`LDD # $C10E`
	`XGDX`
	`JSR X 000`
	`BCS BUSY`
	`LDX# R 1004 Analysis of the converted channel`
	`ASLB Accu B: converted channel number (0/1)`
	`ABX`
	`LDD X 000`
	`... ...`
	`... ...`
	`BUSY RTS`

3) Channel Refresh Time (with conversion in END routine)

a) Program cycle time is less than channel conversion time (depends on the filter)

$$\text{Refresh time}_{max} = \text{Number of channels} * (\text{Channel conversion time} + \text{Program cycle time})$$

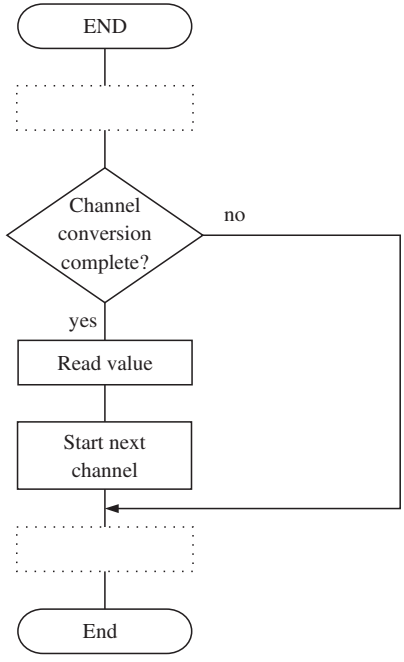
b) Program cycle time is longer than the channel conversion time (depends on the filter)

Refresh time = Number of channels * Program cycle time (one channel per cycle)

4) END Routine

If the analog conversion is executed in the END routine, the routine controls whether the conversion of a channel is finished. If "yes", the value is read and the conversion of the next channel is started. If "no", the analog conversion part of the program is skipped.

Flow chart:



13. ANALOG OUTPUTS

Analog outputs are utilized for converting internal numerical values from the PLC into current and voltage.

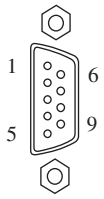
13.1 TECHNICAL DATA

Amount	2
Revision	Short circuit protected
Addressing	Through the operating system
Voltage output	±10 V / 10 mA
Current output	0 - 20 mA, load 50 Ω (400 Ω max.)
Resolution D/A conversion	12 bit
Offset at 25 °C	Software compensation to ±1 LSB[1)]
Offset drift	±0.02 % of full scale / °C (±0.8 LSB/°C)[1)]
Gain error at 25 °C	Software compensation to ±0.5 %
Gain drift	±0.02 %/°C
Gain error load	0.01 %/Ω
Linearity	±1 LSB[1)]
Response time constant	< 1 msec
Maximum output level at reset (or before initialization)	±50 mV or -0.3 mA

[1)] LSB ... Least Significant Bit

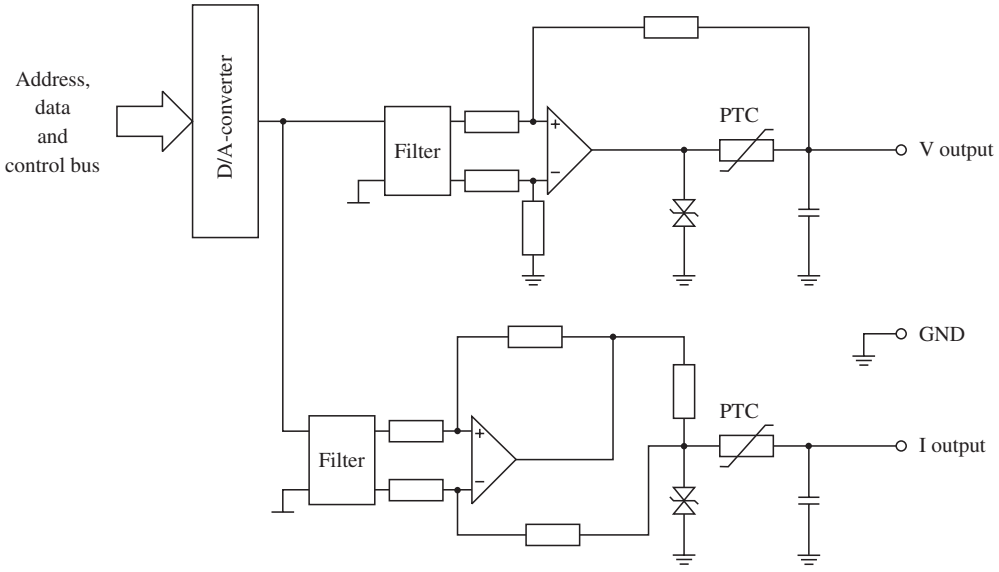
13.2 PIN-OUTS

9 pin D-type (female)



Pin	Assignment
1	V output 1
2	I output 1
3	n.c.
4	V output 2
5	I output 2
6	GND
7	n.c.
8	n.c.
9	GND

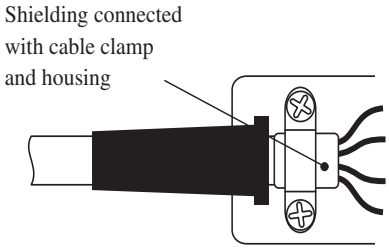
13.3 OUTPUT CIRCUIT



13.4 GROUNDING YOUR CABLE SHIELDING

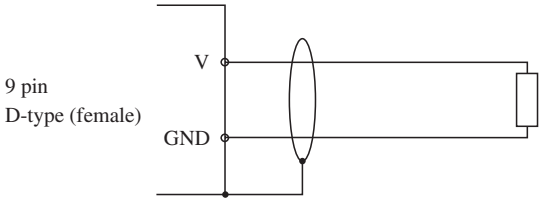
Metal screws or a metallic connector housing are the easiest and most efficient methods of diverting any possible disturbances to the housing of the Compact PLC.

The cable shielding should be in direct contact with the connector housing.

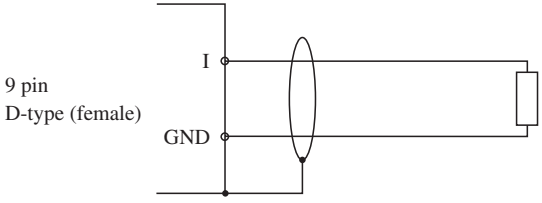


13.5 WIRING

Voltage output



Current output

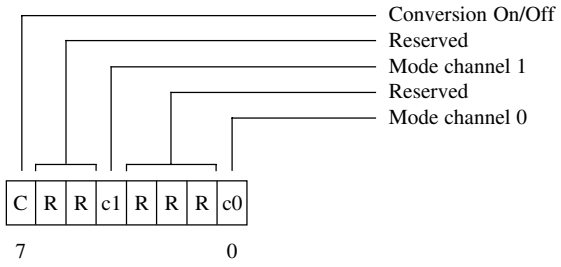


13.6 SOFTWARE OPERATION

The initialization of the analog outputs is done with the register R 1009. The values to be converted are stored in registers R 1010 - R 1013.

The initialization can either be done with commands in STL or with the help of the AOTF function block (see section 13.7).

Register R 1009



Conversion On/Off	Switches the analog conversion on/off. If it is switched off, the analog outputs are 0 (the contents of registers R 1010 - R 1013 are retained). 0 ... Conversion off 1 ... Conversion on
Reserved	Reserved for later use.
Mode channel 1	Bit 4 is used to set the mode for channel 1. **Mode** / **Signal** 0 / ±10 V 1 / 0 - 20 mA
Reserved	Reserved for later use.
Mode channel 0	Bit 4 is used to set the mode for channel 0. **Mode** / **Signal** 0 / ±10 V 1 / 0 - 20 mA

Registers R 1010 - R 1013

Source registers	In registers R 1010 - R 1013, the values to be converted are stored. Value range:

Registers	Channel
R 1010&	0
R 1012&	1

Signal	Value
±10 V	-32768 to +32767
0 - 20 mA	0 to 32767

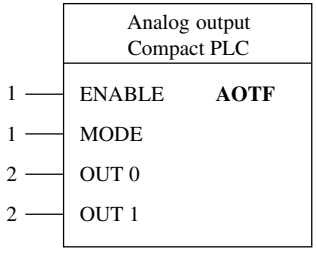
13.7 AOTF - ANALOG OUTPUT COMPACT PLC

This function block is used to initialize the analog outputs. The digital values to be converted are given to the function block through inputs OUT 0 and OUT 1.

NOTE: The analog conversion is only done with a value change. The value comparison is done in a system interrupt. After a power on or a reset (SW or HW), the outputs are set to 0.

AOTF

Function block length: 14 lines
First Scan: No



Inputs/Outputs	Description	Address types
ENABLE	If the ENABLE input is set to 1, the function block is executed.	R, F, #
MODE	Sets the modes for the channels.	R, #

Channel	Bit
0	0
1	4

Mode	Signal
0	±10 V
1	0 - 20 mA

<table>
<tr><th>Inputs/Outputs</th><th>Description</th><th>Address types</th></tr>
<tr><td>OUT0</td><td>Contains the values to be converted from channel 0.

Value range:
<table>
<tr><th>Signal</th><th>Value</th></tr>
<tr><td>±10 V</td><td>-32768 to +32767</td></tr>
<tr><td>0 - 20 mA</td><td>0 to 32767</td></tr>
</table></td><td>R</td></tr>
<tr><td>OUT1</td><td>Contains the values to be converted from channel 1.

Value range:
<table>
<tr><th>Signal</th><th>Value</th></tr>
<tr><td>±10 V</td><td>-32768 to +32767</td></tr>
<tr><td>0 - 20 mA</td><td>0 to 32767</td></tr>
</table></td><td>R</td></tr>
</table>

13.8 ANALOG CONVERSION

The analog conversion is only done with a value change. After a power on or a reset (SW or HW) registers R 1010 - R 1013 and therefore the outputs as well, are set to 0.

1) Value comparison in system interrupt

The value comparison is done in a system interrupt. Only one channel is checked per system interrupt, i.e., the fastest update time per channel is 20 msec.

2) Direct jump to pointer in operating system

In addition to the value comparison in the system interrupt, a pointer can be jumped to directly from the program. This pointer references the analog conversion routine (Pointer: $C111). Shorter update times are accomplished in this way. One channel is checked per call.

Example:

```
LDD   # $C111     Address of the operating system routine
XGDX
JSR   X 000       Operating system routine call
```

3) Checking a single channel

This method is used if only one analog output is required. As in point 2 you make a jump to a pointer in the operating system which references the analog conversion routine (Pointer: $C120). The channel number is sent to Accu B.

Example:

```
LDD   # $C120     Address of the operating system routine
XGDX
LDAB  Channel number (0/1)
JSR   X 000       Operating system routine call
```

14. COUNTER INPUTS AND INTERRUPT INPUT

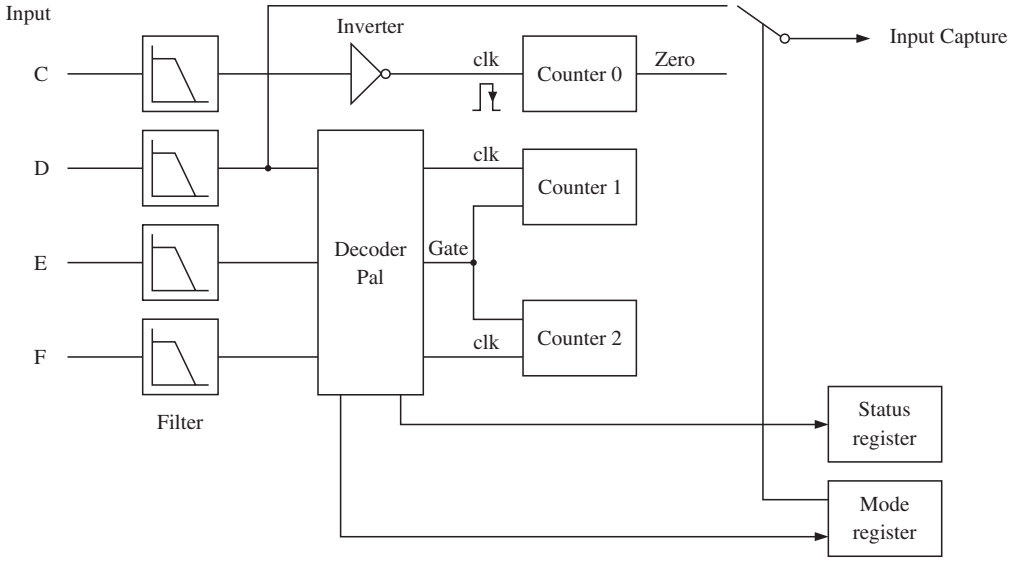
The Compact PLC is equipped with 16 digital inputs. Inputs C - F can also be used for special assignments:

- Event counter
- Interrupt input
- Reference input
- Incremental encoder (Channels A and B) for positioning tasks

The software for the operation of the inputs is saved on the standard software diskette for the Compact Control.

To follow is a description of the input functions, once in graphic format (an overview) and once in table form.

Overview of Functions

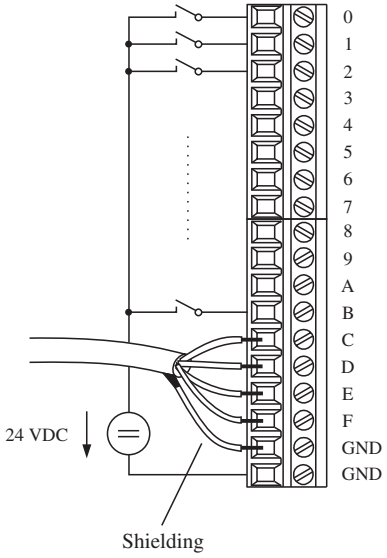


Input Functions

Input	Function
C	Decrements with every negative edge of counter 0. Counter 0 can be used in two ways. The definition is done through the MODE function block input or in the mode register (see section "14.6.3 Mode Register"). 1) Event counter ... Counter with 4 bytes (0 - 4 294 967 295) 2) HW counter ... Counter with 2 bytes (0 - 65535) The hardware counter can be used for generating interrupts. The interrupt is executed if the counter reaches a value of 0 (see also section "14.5 Interrupt Mode").
D	This is used as a reference pulse input or an interrupt input. The definition is done through the MODE function block input or in the mode register (see section "14.6.3 Mode Register"). 1) Reference pulse input: Used to set the counter to a predetermined value (minimum pulse duration: 50 µsec). 2) Interrupt input The use of input D as an interrupt input is described in section "14.5 Interrupt Mode".
E	Decrements counter 1 or is used as input A for two channel counting. The definition is done through the MODE function block input or in the mode register (see section "14.6.3 Mode Register"). 1) Event counter ... Counter with 4 bytes (0 - 4 294 967 295) 2) Input A For positioning tasks, a two channel counting with quadruple evaluation can be executed. The first channel is connected to input E.
F	Decrements counter 2 or is used as input B for two channel counting. The definition is done through the MODE function block input or in the mode register (see section "14.6.3 Mode Register"). 1) Event counter ... Counter with 4 bytes (0 - 4 294 967 295) 2) Input B For positioning tasks, a two channel counting with quadruple evaluation can be executed. The second channel is connected to input F.

14.1 WIRING

Wiring schematic with up to twelve digital inputs, three counter inputs and an interrupt input:



14.2 USING THE COUNTERS

The counters can be operated with the CMDC function block or the CMDD function block or by calling operating system routines from the STL.
These two methods can also be combined with each other.

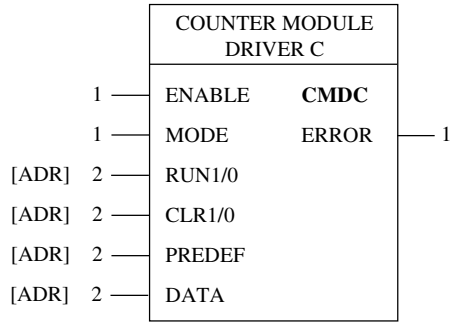
The software for the operation of the inputs is saved on the standard software diskette for the Compact Control.

14.3 CMDC - COUNTER MODULE DRIVER (BRCOMP)

When using this function block, preselected values and new counter states are taken or updated at the next END (program end).

CMDC

Function block length: 120 lines
First Scan: Yes



Inputs/Outputs	Description	Address types
ENABLE	If the ENABLE input is set to 1, the function block is executed.	R, F, #
MODE	Sets the counter operating mode. Edge for Input Capture Interrupt 0 ... Negative edge 1 ... Positive edge Reserved for later use Interrupt source for Input Capture Input 00 ... Interrupt locked 01 ... Input D 10 ... Zero signal hardware counter 0 11 ... Reserved Counter mode for counters 1 and 2 00 ... Event counter pos. edge 01 ... Event counter pos./ neg. edges 10 ... Reserved 11 ... Two channel counter (Encoder) 7 6 5 4 3 2 1 0 Event counter 0 always responds to the positive edge. This setting cannot be changed. If the zero signal from counter 0 is selected as the interrupt source, bit 7 must be set for the positive edge. Otherwise no interrupt is executed. Bit 7 will only change if the status of bits 0 - 3 changes. If counter 0 is used as the interrupt source, the preset value for counter 0 may only be a value in between 3 and 65535 (see section "14.6.8 Calling Operating System Routines", point 3 for the transfer of preset values).	R, #

Inputs/Outputs	Description	Address types
RUN1/0	The start address of a 4 byte long range of memory is attached to this input. Every counter can be stopped or restarted again with this range of memory. Memory assignment / Memory status 1st byte ... Counter 0 / 0 ... Stopped 2nd byte ... Counter 1 / 1 ... Continue 3rd byte ... Counter 2 4th byte ... Encoder position This memory can naturally only be used sensibly if the proper operating status mode is also set. E.g.: Setting the fourth byte is useless if two channel counter mode is not selected.	[ADR] R, [ADR] F
CLR1/0	The start address of a 4 byte long range of memory is attached to this input. Every counter can be set to the preset value with this memory. Assigning the Memory / Memory status 1st byte ... Counter 0 / 0 ... Continue 2nd byte ... Counter 1 / 1 ... Set to preset value 3rd byte ... Counter 2 4th byte ... Encoder position This memory can naturally only be used sensibly if the proper operating status mode is also set. E.g.: Setting the fourth byte is useless if two channel counter mode is not selected.	[ADR] R, [ADR] F
PREDEF	The start address of a 16 byte range of registers is attached to this input. This memory contains the preset values of the counter. Bytes 1 - 4 ... Preset value counter 0 Bytes 5 - 8 ... Preset value counter 1 Bytes 9 - 12 ... Preset value counter 2 Bytes 13 - 16 ... Preset value encoder position This memory can naturally only be used sensibly if the proper operating status mode is also set. E.g.: Setting registers 13 - 16 is useless if two channel counter mode is not selected.	[ADR] R
DATA	The start address of an 18 byte range of registers is attached to this input. This memory contains the counter states of the counters Bytes 1 - 4 ... Counter status counter 0 Bytes 5 - 8 ... Counter status counter 1 Bytes 9 - 12 ... Counter status counter 2 Bytes 13 - 16 ... Counter status encoder position Bytes 17 - 18 ... Counter status counter 0 - Hardware counter This memory can naturally only be used sensibly if the proper operating status mode is also set. E.g.: Reading registers 13 - 16 is useless if two channel counter mode is not selected.	[ADR] R
ERROR	This output indicates that an input is either not connected or is connected incorrectly.	R, F

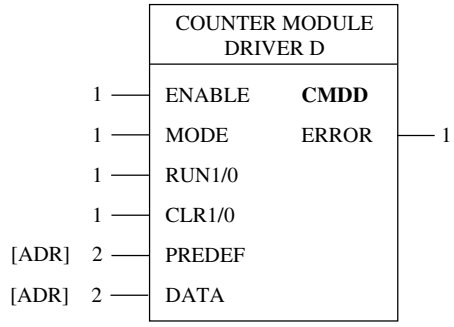
14.4 CMDD - COUNTER 0 MODULE DRIVER (BRCOMP)

The hardware counter can only be predefined if the counter is not selected as interrupt source. After the hardware counter is predefined, it must be defined as interrupt source.
The actual value of the hardware counter only changes to the predefined value after the next edge on the counter input.

The hardware counter cannot be stopped.

CMDD

Function block length: 90 lines
First Scan: Yes



Inputs/Outputs	Description	Address types
ENABLE	If the ENABLE input is set to 1, the function block is executed.	R, F, #
MODE	Sets the operating mode of the 0 counter. Edge for Input Capture Interrupt 0 ... Negative edge 1 ... Positive edge Reserved for later use Interrupt source for Input Capture Input 00 ... Interrupt locked 01 ... Input D 10 ... Zero signal hardware counter 0 11 ... Reserved Reserved for later use 7 6 5 4 3 2 1 0 Event counter 0 always responds to the positive edge. This setting cannot be changed. If the zero signal from counter 0 is selected as the interrupt source, bit 7 must be set for the positive edge. Otherwise no interrupt is executed. Bit 7 will only change if the status of bits 0 - 3 changes. If counter 0 is used as the interrupt source, the preset value for counter 0 may only be a value in between 3 and 65535.	R, #

Inputs/Outputs	Description	Address types
RUN1/0	Stop or free counter 0. 0 ... stop 1 ... continue counting Naturally, the memory can only be used in manner that makes sense if the correct operating mode is set. e.g. Setting the memory only has no effect if hardware counter 0 is selected as interrupt source.	R, F
CLR1/0	Setting the counter to a predefined value. 0 ... continue counting 1 ... set to predefined value	R, F
PREDEF	The start address of a 4 byte long register area is connected to this input. The memory contains the predefined value for the counter 0. If the 0 hardware counter is used as the interrupt source, the predefined value for the 0 hardware counter can only lie between 3 and 65535. Offset 0 ... HHB predefined value for counter 0 1 ... HB predefined value for counter 0 Offset 2 ... MB predefined value for counter 0 or hardware counter 3 ... LB predefined value for counter 0 or hardware counter	[ADR] R
DATA	The start address of a 6 byte long register area is connected to this input. The memory contains the position of the counter. Offset 0 ... HHB position of the counter 0 1 ... HB position of the counter 0 2 ... MB position of the counter 0 3 ... LB position of the counter 0 Offset 4 ... HB position of the hardware counter 5 ... LB position of the hardware counter	[ADR] R
ERROR	This output indicates if an input is not connected or is incorrectly connected.	R, F

14.5 INTERRUPT MODE

An interrupt source for the generation of an interrupt can be defined in the mode register or with the MODE function block input.
Interrupts are only enabled or locked when the next END (program end) is executed and not after the definition of the mode register or during the execution of the function block.
By calling the operating system routine, the interrupt is immediately enabled or locked.

1. Input D is used as Interrupt Source

Depending on the definition, the positive or negative edge activates an interrupt on this input. The interrupt is repeated for every respective event on input D.

For function block CMDC, the edge definition is done with the MODE input. By calling an operating system routine from the STL the initialization is done in the initialization register $2122.

2. Hardware Counter 0 is used as Interrupt Source

The interrupt is executed if hardware counter 0 has counted from the preset value to 0. Since the hardware counter only has a 16 bit range, only preset values are sent into the 2 byte numerical range. The valid value range is from 3 to 65535 (see section "14.6.8 Calling Operating System Routines", point 3 for the transfer of preset values).

Hardware counter 0 counts down to 0 and carries on the count again from 65535. The next time zero is reached an interrupt is executed.
If the procedure should be started from a certain value, hardware counter 0 must be loaded with the preset value after every time zero is reached.

Interrupt Routines

With a One-time write operation (e.g. First Scan), registers R 0968 and R 0969 are assigned with the address of the interrupt program.

```
        ...
        LDXL  INTR
        STX   R 0968
        ...
```

Interrupt routine INTR is written as a subprogram in STL and closed with RTS.

```
INTR    LDAA  I 043
        ...
        ...
        RTS
```

NOTE: Ladder diagrams cannot be used as interrupt routines, since all ladder diagrams in the program use the same temporary memory which could lead to incorrect assignments.

14.6 OPERATION IN STL

Utilizing the three counters and the reference or interrupt input can also be done in STL. The settings are made in the END routine.

If the settings must be used immediately, operating system routines can be called. This is done with a pointer in the operating system. The pointer refers to the address of the routine.

Actual values and preset values of the counter sit at operating system addresses. The counter initialization is done with three registers (control register, mode register and initialization register).

14.6.1 Counter Operating System Addresses

Actual value

Addresses	Actual value	Range
$2100 - $2103	Counter 0	0 - 4 294 967 295
$2104 - $2107	Counter 1	0 - 4 294 967 295
$2108 - $210B	Counter 2	0 - 4 294 967 295
$210C - $210F	2 channel counter (Encoder)	0 - 4 294 967 295
$2123 - $2124	Hardware counter 0	0 - 65535

Preset values

Addresses	Preset values	Range
$2110 - $2113	Counter 0	0 - 4 294 967 295 [1)]
$2114 - $2117	Counter 1	0 - 4 294 967 295
$2118 - $211B	Counter 2	0 - 4 294 967 295
$211C - $211F	2 channel counter (Encoder)	0 - 4 294 967 295

[1)] If counter 0 is used as a hardware counter, the preset value is between 3 and 65535 (2 byte counter).

14.6.2 Control Register $2120

Bit	Function
0	Enable counter 0
1	Enable counter 1
2	Enable counter 2
3	Enable 2 channel counter (Encoder)
4	Clear counter 0
5	Clear counter 1
6	Clear counter 2
7	Clear 2 channel counter (Encoder)

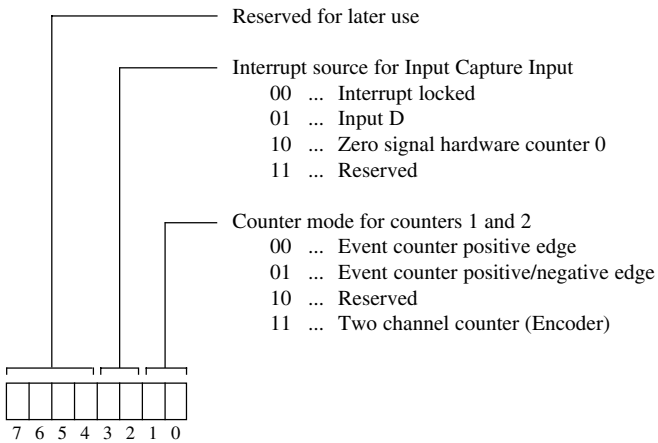
Enable Counter

These bits can be used to individually stop every counter and enable it again.

Clear Counter

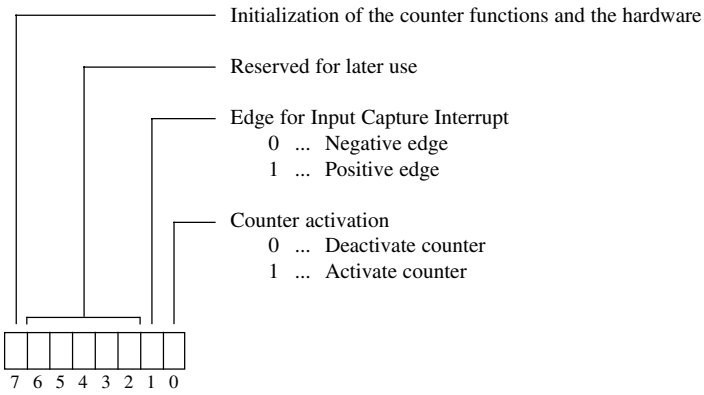
These bits may only be set for one program cycle. If bit 0 in the initialization register ($2122) is set, the respective counter is set to the preset value during the END routine execution.
For the operating system routines "Load counter x with preset value" these bits do not have to be set.

14.6.3 Mode Register $2121



Event counter 0 always responds to the positive edge. This setting cannot be changed.
If hardware counter 0 is used as the interrupt source, the preset value for the hardware counter 0 can only be between 3 and 65535.

14.6.4 Initialization Register $2122



If the zero signal from hardware counter 0 is selected as the interrupt source, bit 1 must be set on the positive edge. Otherwise no interrupt is executed.
Changing bit 1 is only accepted if bits 0 - 3 of the mode register are changed.

14.6.5 Addresses of Operating System Routines

Address	Actual value
$C114	Set mode register
$C117	Activate reference pulse
$C11A	Read reference pulse
$C11D	Activate interrupt source zero signal hardware counter 0
$C123	Load counter 0 with preset value
$C126	Load counter 1 with preset value
$C129	Load counter 2 with preset value
$C12C	Load 2 channel counter (encoder) with preset value
$C12F	Read counter 0
$C132	Read counters 1 and 2
$C135	Read two channel counter (encoder)
$C138	Enable Input Capture Interrupt
$C13B	Disable Input Capture Interrupt
$C13E	End referencing
$C141	Initialize counter

14.6.6 Function of the Counter

Bit 0 must be set in initialization register ($2122), in order to activate the counter and the generation of the Input Capture Interrupt.

The counter functions can be collected or be set for each counter individually (control register $2120).

In the control register ($2120), every individual channel (depending on the operating system mode) can be enabled or stopped. Enabled counters decrement the respective actual value range ($2100 - $210F) of the counter status. This is a 4 byte value (0 - 4 294 967 295). When 0 is reached, it starts decrementing again from the maximum count.

If hardware counter 0 is used as the interrupt source, the counting range is between 0 and 65535 (Preset value: 3 - 65535).

In the control register ($2120), every individual channel (depending on the operating system mode) can be set to the preset value. This must be entered previously in the respective memory location ($2110 - $211F).

The preset value of the counter can be changed at any time. The counter can be set to the preset value at any time as well.

14.6.7 Counter Initialization

Setting the counter mode, the interrupt source and activating the counter are done with the mode register and the initialization register. The operating mode can also be changed at any time during operation.

In order to put the settings to use, bit 7 in the initialization register ($2122) must be set. The initialization is executed and bit 7 is cleared again in the END routine.
For later changes to the Input Capture Edge, bit 7 must not be set (see section "14.6.8 Calling Operating System Routines", Point 1).

NOTE: After a power-on or a reset, no functions are enabled.

Example Initialization register $2122:

Activate counter
Negative edge for Input Capture Interrupt
Set bit 7 for initialization

Mode register $2121:

Counters 1 and 2 as event counters with positive edge
Interrupt source for Input Capture is input D

The program segment shown below is only executed one time during the first program cycle. The defined initialization is executed in the END routine.

```
LDD   # $2122          Initialization register
XGDX
LDAA  # %10000001      neg. edge Capture Interrupt, Counter act.
STAA  X 000
LDD   # $2121          Mode register
XGDX
LDAA  # %00000100      Interrupt Input D, Counter pos. edge
STAA  X 000
```

14.6.8 Calling Operating System Routines

1) Setting the Mode Register

No call parameters, no response data

NOTE: Calling the operating system routine may be done one program cycle after the initialization at the earliest.

Example Initialization register $2122:

Positive edge for Input Capture Interrupt

- Must be positive, if hardware counter 0 is used as the interrupt source.
- A change in this bit is only accepted if bits 0 - 3 in the mode register are also changed.

Mode register $2121:

Counters 1 and 2 as event counters with positive/negative edge
interrupt source for Input Capture is hardware counter 0

The setting is accepted immediately after the operating system routine is called.

```
LDD   # $2121          Mode register
XGDX
LDAA  # %00001001      Counter pos/neg edge, Int HW-counter 0
LDAB  # %00000011      Positive Edge Capture Interrupt
STD   X 000
LDD   # $C114          Addr of the OS routine: "Set Mode register"
XGDX
JSR   X 000            Operating system routine call
```

2) Reference Pulse

Before the call "Read reference pulse", the function "Activate reference pulse" must be called first. Reference bit 0 is set to 0.
When the reference pulse occurs the reference bit is set to 1. This remains 1 until the function "Activate reference pulse" is called again.

Activate Reference Pulse:

No call parameters, no response data

```
LDD   # $C117          Activate reference pulse call
XGDX
JSR   X 000
```

Read Reference Pulse:

No call parameters, response data in accu A, bit 0

```
LDD   # $C11A          Read reference pulse call
XGDX
JSR   X 000
BITA  # 001            Bit 0 in Accu A set ?
BEQ   LABL
```

3) Activate Interrupt Source Zero Signal Counter 0

Transfer preset value to accumulator D, no response data

```
LDD   # $C11D          Activate interrupt source call
XGDX
LDD   # 10500          E.g. preset value = 10500 pulses
JSR   X 000
```

For the call of this function, the required changes are executed in the mode register ($2121) and in the initialization register ($2122). The counter is written to directly with the transfer value in accumulator D.

NOTE: If this function is used in conjunction with the CMDC function block, it must be coordinated with the register connected to the MODE input. Otherwise the operating mode will be changed again during the END routine.

```
LDAA  # % 100010xx
STAA  R xxxx
LDD   # $C11D          Activate interrupt source call
XGDX
...
```

4) Changing the Preset Value of Counter x

The preset values of the counter are stored in addresses $2110 - $211F. In order to change the values, the procedure in the following example must be followed.

Example The preset value of counter 0 is changed. The value sits in registers R 2000 - R 2003.

```
LDD   #$2110            Start address preset value counter 0
XGDX
LDD   R 2000            Higher Word preset value
STD   X 000
LDD   R 2002            Lower Word preset value
STD   X 002
```

5) Load Counter x with Preset Value

The preset values of the counters are stored at addresses $2110 - $211F. The pointers to the operating system routines sit at the following addresses:

Address	Actual value
$C123	Load counter 0 with preset value
$C126	Load counter 1 with preset value
$C129	Load counter 2 with preset value
$C12C	Load 2 channel counter (Encoder) with preset value

NOTE: The corresponding bits (4 - 7) in the control register **must not** be set .

In order to change the values, the following procedure must be followed:

```
LDD   # $xxxx           Address of the pointer to OS routine
XGDX
JSR   X 000
```

6) Read Counter x

The actual values of the counters are stored at addresses $2100 - $210F. The pointers to the operating system routines sit at the following addresses:

Address	Actual value
$C12F	Read counter 0
$C132	Read counters 1 and 2
$C135	Read 2 channel counter (Encoder)

NOTE: The corresponding bits (0 - 3) in the control register **must be set**.

In order to read the values, follow the procedure shown below:

```
LDD   # $xxxx          Address of pointer to OS routine
XGDX
JSR   X 000
```

7) Copy Actual Value to Registers

The actual values of the counters are stored at addresses $2100 - $210F. In order to copy the values to registers, the procedure shown in the example below must be followed.

Example The actual value of counter 0 will be copied to registers R 3000 - R 3003.

```
LDD   #$2100           Start address Actual value Counter 0
XGDX
LDD   X 000            Higher Word Actual value
STD   R 3000
LDD   X 002            Lower Word Actual value
STD   R 3002
```

8) Hardware Counter 0

The actual value of hardware counter 0 is stored at addresses $2123 and $2124. It is read by using the "Load Counter 0" function (see point 6).

Because of hardware limitations, the status of the 4 byte counter 0 in the memory range $2100 - $2103 cannot be synchronized with the hardware counter.
The effect only occurs if **"Zero Signal Counter 0 as Interrupt Source"** is set. By setting the counter to different preset values during operation, differentiation between real positioning changes, overflows or new start values can no longer be made.
This is not the case for any of the other operating modes.

Addresses $2123 and $2124 always indicate the pulse remaining until the interrupt.

NOTE: If the counter is read with the preset value, the preset value is not accepted immediately. The next pulse decrements the old counter state one more time.

Example

3000	Load with preset value (e.g. with 5000)
2999	The next pulse decrements the old counter state
5000	The preset value is accepted
4999	The second pulse decrements starting from the preset value
:	

15. RELAY EXPANSION CARD FOR BRCOMP

15.1 GENERAL INFORMATION

A relay expansion card is available for the Compact PLC which adds 16 additional relay outputs to the 14 transistor outputs that already exist (O 060 - O 06F).

The Compact PLC and the relay expansion card can be mounted either beside each other or piggyback (one on top of the other).
The connection to the controller is made with the cable BRKA08-0 via the PATA interface. The cable must be ordered separately.

The DOUC function block transfers the output states to the relay expansion card.

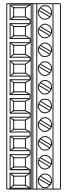
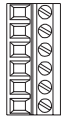
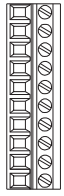
15.2 RELAY GROUPS

Group	Outputs	Type
1	4	N.O.
2	4	N.O.
3	2	N.O.
4	2	N.O.
5	1	N.O.
6	1	N.O.
7	1	Switching
8	1	Switching

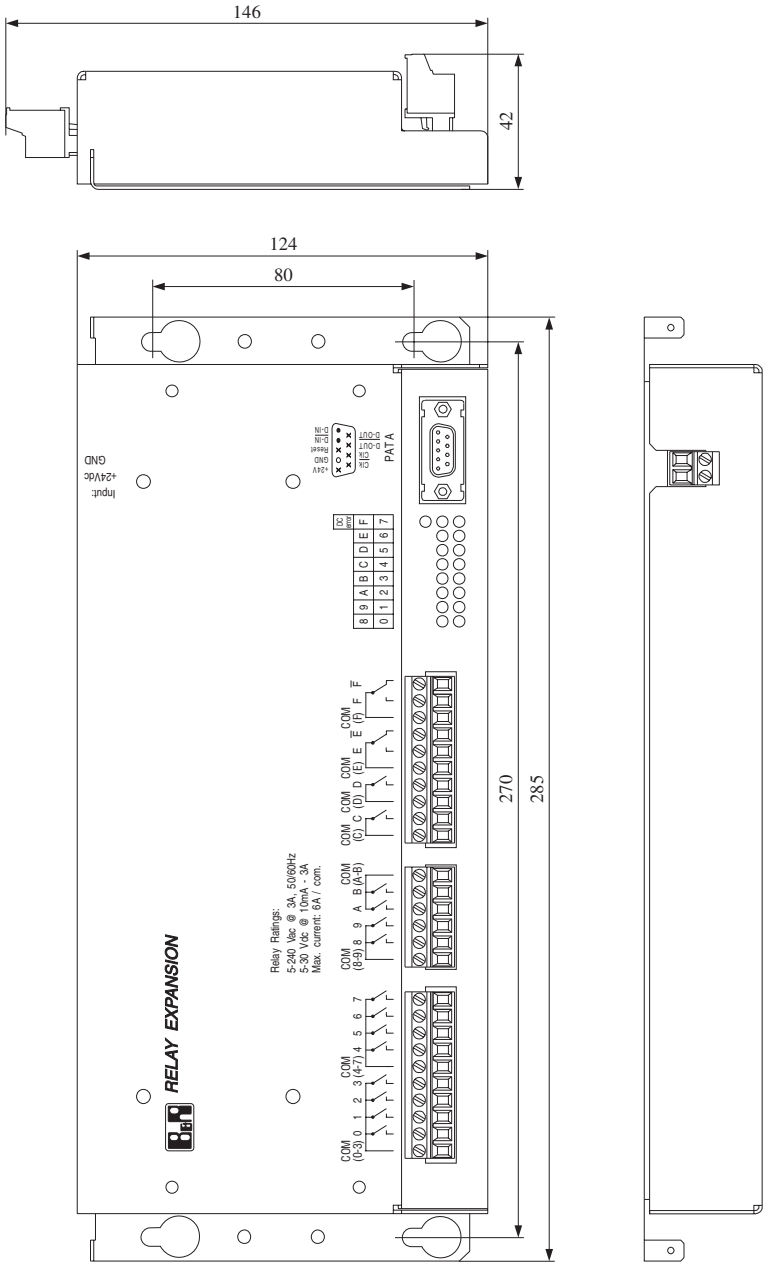
15.3 TECHNICAL DATA

Number	16
Type	Relay
Number of Groups	8
Switching Voltage	5 - 240 VAC, 50/60 Hz 5 - 30 VDC
Switching Current per Output per Group	 max. 3 A max. 6 A
Switching Delay log. 0 → 1 log. 1 → 0	 approx. 10 msec approx. 15 msec
Protection Circuit	external from user
Switching Play mechanical electrical	 $> 2 \times 10^7$ $> 1 \times 10^5$
Relay Supply	24 VDC ±15 %, max. 400 mA
Operating Temperature	0 to 55 °C
Relative Humidity	0 - 95 % non-condensing

15.4 PIN ASSIGNMENTS

Terminal Block	Pin	Assignment
	1	GND (O 060 - O 063)
	2	Output 0
	3	Output 1
	4	Output 2
	5	Output 3
	6	GND (O 064 - O 067)
	7	Output 4
	8	Output 5
	9	Output 6
	10	Output 7
	11	GND (O 068 - O 069)
	12	Output 8
	13	Output 9
	14	GND (O 06A - O 06B)
	15	Output A
	16	Output B
	17	GND (O 06C)
	18	Output C
	19	GND (O 06D)
	20	Output D
	21	GND (O 06E)
	22	Output E - N.O.
	23	Output E - N.C.
	24	GND (O 06F)
	25	Output F - N.O.
	26	Output F - N.C.

15.5 MEASUREMENTS



15.6 INSTALLATION GUIDELINES

The relay expansion card can be mounted either horizontally or vertically. The distance to the neighboring module caused by the fastening bracket is enough to ensure sufficient air circulation.

The area under the relay expansion card is to be kept below the maximum operating temperature of 55 °C. A fan is not required to cool the housing.

15.7 MOUNTING

Two Possibilities

- directly on the back panel of the enclosure
- on mounting rail

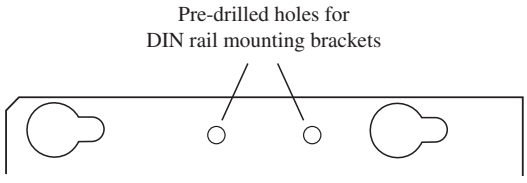
Directly on the Back Panel of the Enclosure

The chassis is to be screwed onto the back panel of the enclosure with the four mounting holes making sure there is good contact with the back panel.

M5 screws are to be used (distance between holes: 80 * 270 mm).

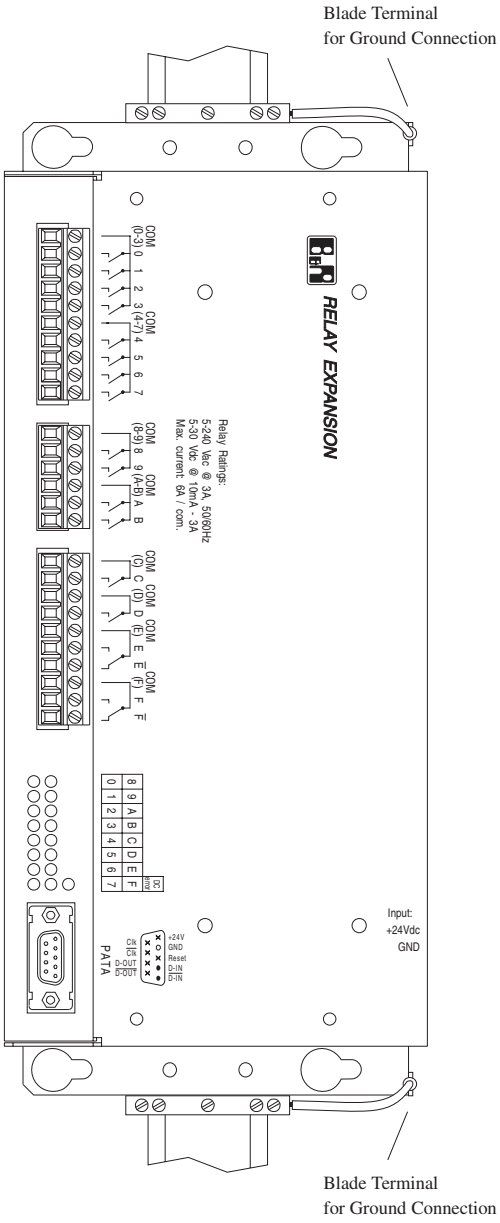
On Mounting Rail

In order to mount the chassis on mounting rail (DIN EN 50022-35), the two accompanying mounting rail fasteners (type: KSA10) must be screwed to the relay expansion card.



Install relay expansion card on the mounting rail. The mounting rail must be grounded!

Install a grounding clamp left and right of the relay expansion card. Ground the relay expansion card with the blade terminal provided.



15.8 COMPACT PLC AND RELAY EXPANSION CARD

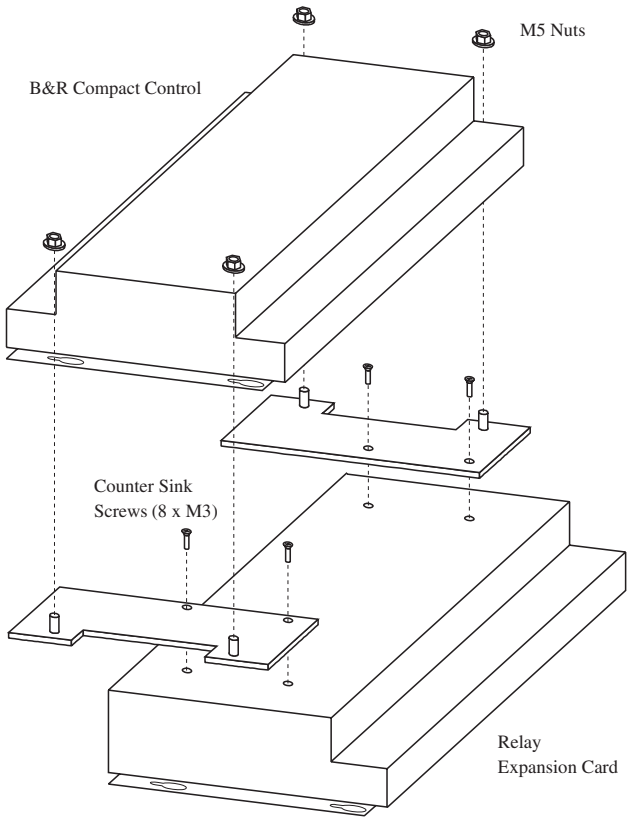
The Compact PLC and the relay expansion card can be mounted either beside each other or piggyback (one on top of the other).

1. Beside Each Other

If the two housings are mounted next to each other, please follow installation and mounting instructions given.

2. Piggyback

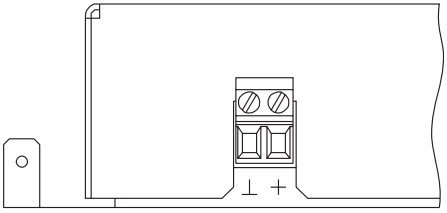
With this type of installation, the relay expansion card is first mounted either directly on the back panel of the enclosure or on the mounting rail (see instructions). Then the Compact PLC is fastened to the relay expansion card with the two mounting plates.
The mounting plates are included with the delivery of the relay expansion card.



15.9 CONNECTION TO THE COMPACT PLC

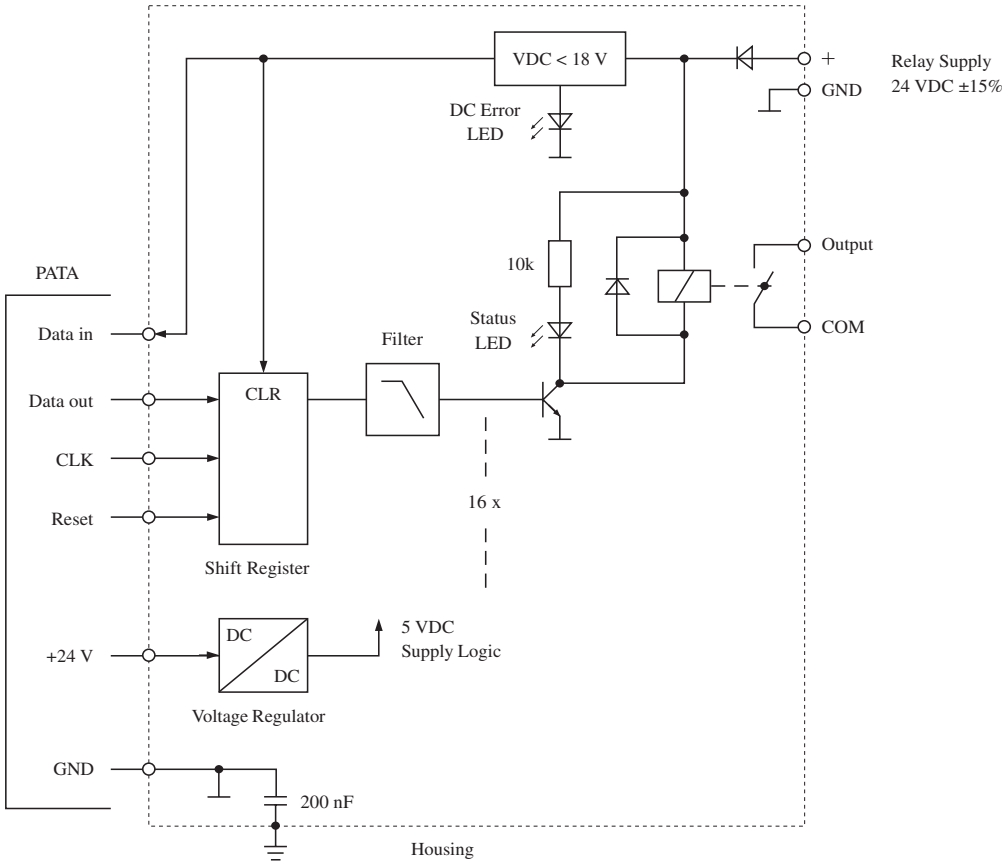
The connection to the Compact PLC is made with the cable BRKA08-0 via the PATA interface. The cable must be ordered separately.

15.10 RELAY SUPPLY



+ +24 VDC ±15 %, max. 400 mA
⊥ GND

15.11 SWITCHING CIRCUIT



16. IF1 - RS232 INTERFACE

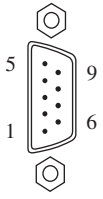
16.1 TECHNICAL DATA

Interface	RS232
Galvanic isolation	NO
Connection	9 pin D-type (male)
Max. distance	15 m, shielded cable
Handshake lines [1)]	DSR, RTS
Baudrate	300 - 19200 Baud
Data format Data bits Parity Configuration	 5 to 8 yes / no / even / odd software
Supply for Compact MMI P120 and MMI P121	+5 V / 500 mA

[1)] The DTR signal can be created from the RTS with the appropriate interface software (see section "18. Using Interfaces IF1 and IF2").

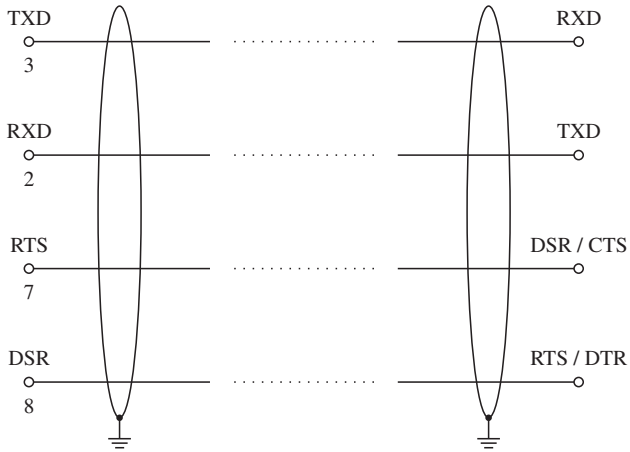
16.2 PIN-OUTS

9 pin D-type (male)



Pin	Assignment
1	
2	RXD
3	TXD
4	+5 V / 500 mA
5	GND
6	
7	RTS
8	DSR
9	

16.3 WIRING



16.4 GROUNDING AND SHIELDING

Shielded cable must be used for interface connections. The cable shielding must be grounded at both ends.

16.5 SOFTWARE OPERATION

The software functionality of the interface is described in section "18. Using Interfaces IF1 and IF2".

16.6 COMPACT MMI P120 AND MMI P121

Interface panels Compact MMI P120 and MMI P121 are available for the Compact PLC. These are connected to the IF1 interface (connecting cable: BRKACOMP1-0). Supply is through pin 4.

17. IF2 - RS232/RS485 INTERFACE

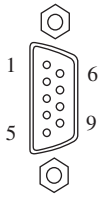
17.1 TECHNICAL DATA

Interface	RS232	RS485
Galvanic isolation	NO	YES
Max. distance	15 m, Shielded Cable	1200 m, Shielded Twisted Pair
Handshake lines [1]	DSR, RTS	
Connection	9 pin D-type (female)	
Baudrate	300 - 19200 Baud	
Data format Data bits Parity Configuration	 5 to 8 yes / no / even / odd software	

[1] The DTR signal can be created from the RTS with the appropriate interface software (see section "18. Using Interfaces IF1 and IF2").

17.2 PIN-OUTS

9 pin D-type (female)

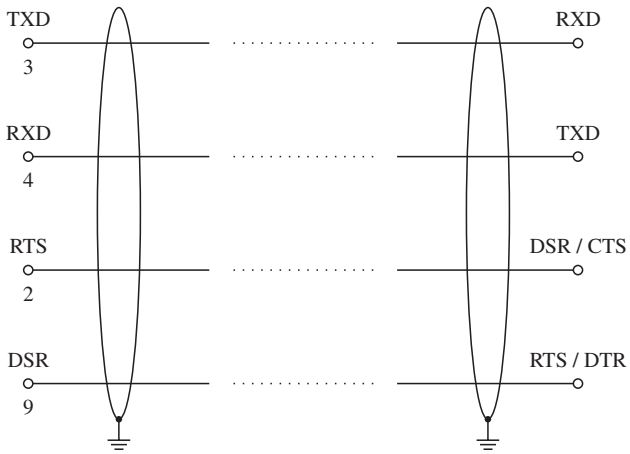


Pin	RS232	RS485
1	GND	
2	RTS	
3	TXD	
4	RXD	
5		DATA
6		$\overline{\text{Enable}}$
7		GND [2]
8		$\overline{\text{DATA}}$
9	DSR	

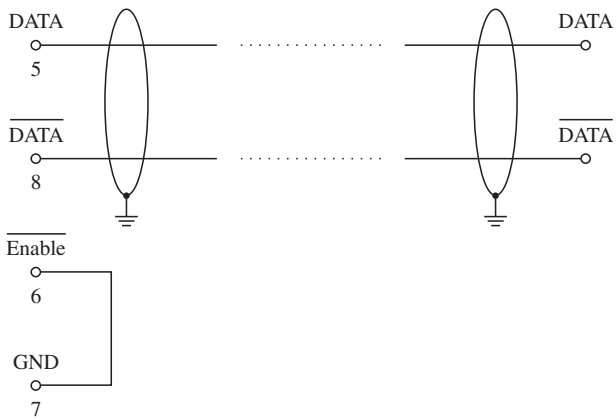
[2] Pin 6 ($\overline{\text{Enable}}$) must be connected with GND on pin 7 during RS485 operation. This should be taken into consideration when cabling with standard cable (standard: connection Pin 6 with Pin 1).

17.3 WIRING

RS232 interface



RS485 interface



17.4 GROUNDING AND SHIELDING

Shielded cable must be used for interface connections. The shielding must be grounded on both ends.

17.5 SOFTWARE OPERATION

The software functionality of the interface is described in section "18. Using Interfaces IF1 and IF2".

18. USING INTERFACES IF1 AND IF2

Operating the user interfaces is done with the following registers:

Register	IF1	IF2
Data register	P 100	P 300
Status register	P 101	P 301
Command register	P 102	P 302
Program register	P 103	P 303

Initialization

The program register and the command register are written to with certain preset values during the initialization. This defines the baudrate, data format, parity etc. The initialization is only executed one time immediately after switching on the PLC or after a reset.

Program register

7			0
SB	DB	1	BAUD

P 103 / P 303

SB Stopbits
- 0 ... 1 Stopbit
- 1 ... if DB=5 and no parity ... 1.5 Stopbits
  - if DB=8 and parity ... 1 Stopbit
  - in all other cases ... 2 Stopbits

DB Databits
- 00 ... 8 Databits
- 01 ... 7 Databits
- 10 ... 6 Databits
- 11 ... 5 Databits

BAUD Baudrate

0110	...	300	1010	...	2400	1101	...	7200
0111	...	600	1011	...	3600	1110	...	9600
1000	...	1200	1100	...	4800	1111	...	19200
1001	...	1800						

Command register

7						0
PAR	P_{on}	E	RT	0	1	1

P 102 / P 302

PAR Parity
- 00 ... Parity odd
- 01 ... Parity even
- 10 ... Parity bit set during send
- 11 ... Parity bit cleared during send

P_{on} Parity on/off
- 0 ... No parity checking, Parity bit not generated
- 1 ... Parity checking active

E Echo Mode
- 0 ... Echo Mode off
- 1 ... Echo Mode on, RT must be 0

RT RTS line
- 0 ... RTS high, not ready to send (-12 V)
- 1 ... RTS low, ready to send (+12 V)

Example Initializing user interface IF1:

Baudrate = 9600, 8 Databits, 1 Stopbit, Parity off, Echo Mode off

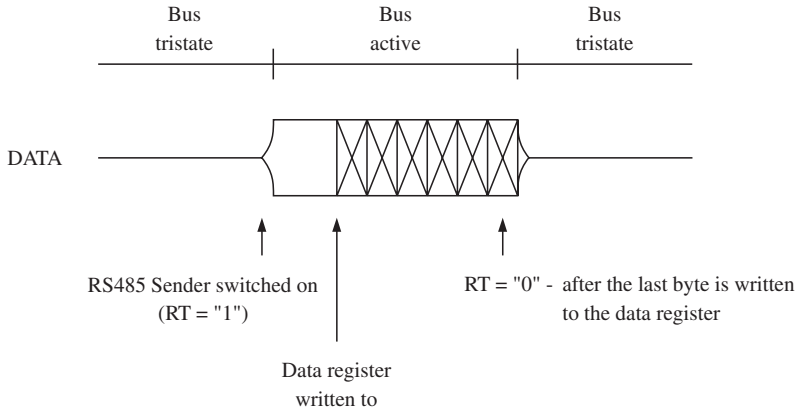
```
LDAB  # %00011110       9600 Baud, 8 Databits, 1 Stopbit
LDAA  # %00001011       Parity off, Echo Mode off
STD   P 102             Program register & command register
```

The RT Bit

RS485 interface - Bus connection:

In order to switch on the RS485 sender, the RT bit must first be set to 1. After writing the last byte to be transmitted to the data register, the RT bit must be reset again. Switching the bus to tristate (high resistance state) is now done automatically after the last byte is sent.
The bus remains active until the character is completely sent (maximum RTS delay is 5 µsec.).

If there is no active sender which also means that the bus is in a state of high resistance, there is a possibility of undefined characters being received.

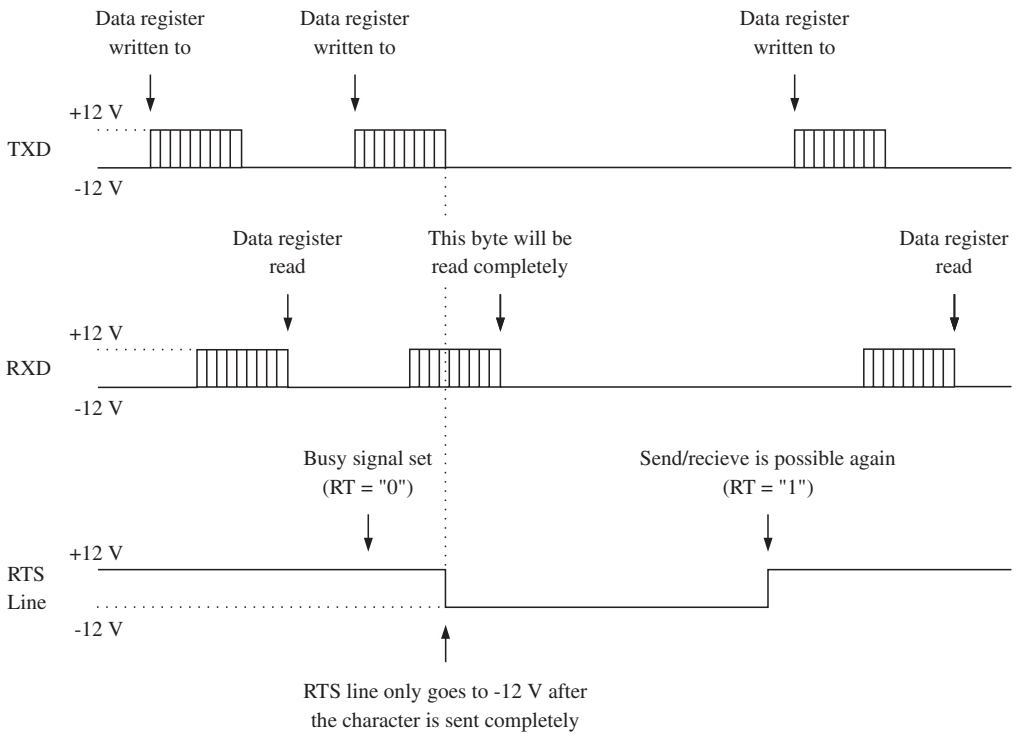


RS232 Interface - Point to Point:

The RT bit is normally set for RS232 communication. If the user wants to set a busy signal, the RT bit can be set to "0".

NOTE: If the RT bit is "0", this sender is also inactive!

The following diagram shows signals in an RS232 communication.



NOTE: The originating sender is also inactive.

Status Register

The status register delivers information about the status of the serial interface and about possible errors. The status of the status register must be checked for every send or receive procedure.

Status register

7							0
	DS		TR	RF	OV	FE	PE

P 101 / P 301

DS	DSR Line Status	0 ...	DSR low - Other station ready
		1 ...	DSR high - Other station not ready
TR	Transmit ready	0 ...	Transmitter sends frame (character)
		1 ...	Send register empty. Sender ready to send a frame
RF	Receive frame	0 ...	No frame (character) received
		1 ...	Frame received
OV	Overrun error	0 ...	No error
		1 ...	Error. The receiver was not properly read before a new frame (character) was received
FE	Framing error	0 ...	No error
		1 ...	Error. Stopbit not recognized.
PE	Parity error	0 ...	No error
		1 ...	Parity test error

Data Register

The data register has two different functions:

- Frames to be received are read from the data register
- Frames to be sent are written to the data register

Data register

7							0
DATA							

P 100 / P 300

DATA Data to be sent or received

Outputting a Character

Before writing the character to be output to the data register, the transmitter (sender) must first be checked for whether it is ready to send a character or not (Bit 4 must be 1 in the status register).

```
        LDAB  P 101            Status register
        BITB  # %00010000      Transmitter ready ?
        BLS   NO               Branch, if transmitter not ready
        LDAA  x xxx            Frame (character) to be output
        STAA  P 100            Data register
```

Reading a Character

Whether a character has been received or not is determined by checking bit 3 in the status register. If bit 3 = 1, a character has been received. Bits 0 through 2 in the status register indicate whether transmission errors have occurred (parity error, overrun error or framing error). If one of these error bits is set, the received character is invalid.
The data register must also be read if an error has occurred, since the error message must be acknowledged through it.

```
        LDAB  P 101            Status register
        BITB  # %00001000      Frame (character) received ?
        BLS   NO               Branch, if no character received
        LDAA  P 100            Read data register
        BITB  # %00000111      Transmission error ?
        BLS   FAIL             Branch, if transmission error
        :                      Check received character

FAIL    :
```

18.1 USING INTERFACE INTERRUPTS

The interface module triggers the following interrupts:

a) Send Interrupt: If the transmitter is ready to send a character.

b) Receive Interrupt: If a character is received (also if an error occurs).

In order to acknowledge the interrupt, the interrupt routine must:

a) Send Interrupt: write to the data register.

b) Receive Interrupt: read from the status register.

IRQ Vector

The interrupt vector is stored in registers R 0976 and R 0977. Before the interrupt is enabled, the vector must be set to the address of the user interrupt routine. This definition is done in the initialization routine.

Enabling

Both interrupts are enabled or disabled in command registers (P 102 or P 302). DTR must be active (bit 0 = 1) in order to trigger interrupts.

Command register

7					0
PAR	P_{on}	E	TIC	I	1

P 102 / P 302

[1] RTS level on line:

High ... -12 V
Low ... +12 V

PAR Parity
- 00 ... Parity odd
- 01 ... Parity even
- 10 ... Parity bit set during transmission
- 11 ... Parity bit cleared during transmission

P_{on} Parity on/off
- 0 ... No parity test, parity bit not generated
- 1 ... Parity test active

E Echo Mode
- 0 ... Echo Mode off
- 1 ... Echo Mode on, TIC must be 00 or 01

TIC Transmitter Interrupt Control
Send interrupt and handshake line RTS control

- 00 ... RTS high [1], Send interrupt disabled (Sender off)
- 01 ... RTS low, Send interrupt enabled (Sender on)
- 10 ... RTS low, Send interrupt disabled (Sender on)
- 11 ... RTS low, Send interrupt disabled (Break on TXD = +12 V)

I IRD - Interrupt Request Disabled (Receive interrupt control)

- 0 ... Receive interrupt enabled
- 1 ... Receive interrupt disabled

Example

1) Enabling the receive interrupt:

```
LDAA  # %00001001
STAA  P 102          Command register
```

2) Enabling the send interrupt:

```
LDAA  # %00000111
STAA  P 102          Command register
```

3) Enabling the send and receive interrupts:

```
LDAA  # %00000101
STAA  P 102          Command register
```

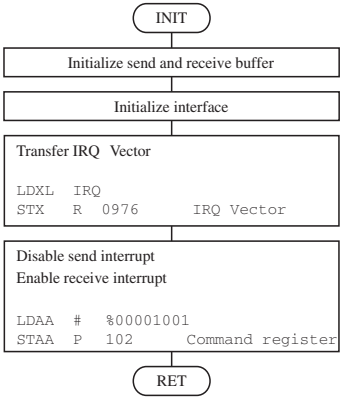
Interrupt Utilization

These two interrupts can be separately enabled or disabled in command registers. They both deal with the same vector however. In order to use the receive and the send interrupts simultaneously, it must be determined whether the interrupt is initiated by the receiver or the sender.
This is done by reading the RF "Receive Frame" bit in the status register (bit 3). If the bit is 0, the interrupt has been initiated by the sender.

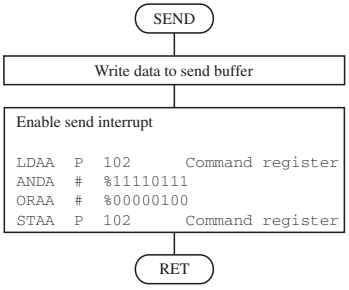
NOTE: If both interfaces (IF1 and IF2) operate with interface interrupts, the status registers of **both** interfaces must be evaluated in an interrupt routine.

Application Example

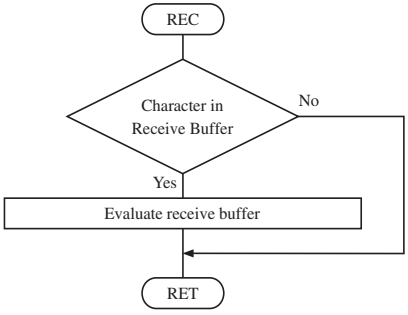
In this example, send and receive interrupts are used at the same time. Initialization takes place in the "INIT" subroutine.



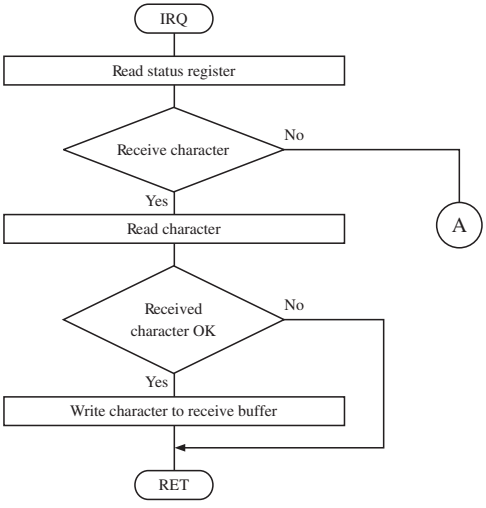
Data is written in a send buffer and the send interrupt is triggered in the "SEND" subroutine.



A check is made by the in the "REC" subroutine to see if a character is in the receive buffer. If a character is received, it is evaluated.



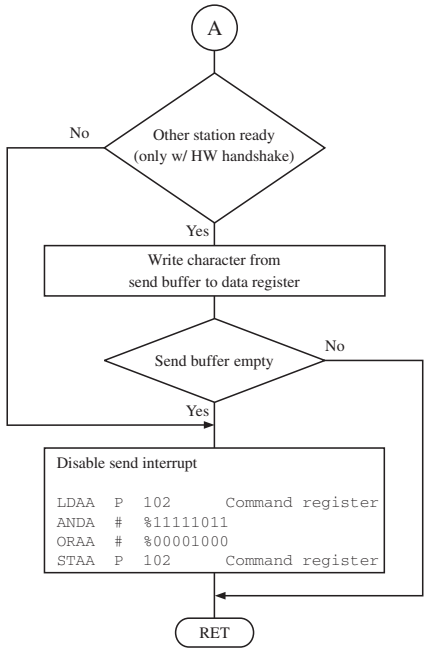
The interrupt routine "IRQ" is divided into two sections. In one section, the received characters are checked and written to the receive buffer. In the other section of the routine, a character from the send buffer is written to the data register.



NOTE:

If the other station isn't ready, the send interrupt will be disabled. A re-enable is done with the "SEND" subroutine.

By disabling the interrupt, the application program is no longer disrupted by continuous interrupt calls.



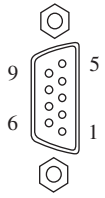
19. IF3 - PATA/SSI INTERFACE

The IF3 interface is a modified RS422 interface. A MINICONTROL operator panel or a relay expansion card can be connected to the IF3 interface (PATA interface).

It is also set up as an SSI interface for connecting absolute encoder systems.

Pin-outs

9 pin D-type (female)



Pin	Assignment
1	$\overline{\text{DATA IN}}$
2	DATA IN
3	Reset
4	GND
5	+24 V
6	$\overline{\text{DATA OUT}}$
7	DATA OUT
8	$\overline{\text{CLK}}$
9	CLK

Shielding and Grounding

For interface connections, shielded cables must always be used. This shielding must always be grounded at both ends.

PATA Interface

Software operation for the MINICONTROL operator panels is described in the "Operator Interface Panel User's Manual" (MATERMINAL-E).

For MINICONTROL operator panel control, the standard software package SWPLCSTD02-0 (Rev. 00.31 or higher) is available from B&R. However, the operator panel software for the Compact Control is saved on the standard software diskette SWPLCBRC01-0.
Panel operation is done with function blocks and table parameter definitions.

Important addresses:

Address	Description
E 05F	Serial key code
A 05E	Serial dislay data
A 05F	Reset signal

SSI Interface

The following conditions must be met in order to connect absolute encoders to the SSI interface:

- Only absolute encoders with a monoflop time between 20 µsec and 260 µsec can be used.
- Absolute encoders with a maximum 24 Bit (AG24) or 32 Bit (AG32) can be read.

The function blocks AG24 and AG32 are provided for the operation. They are contained in the standard software package 4 SWPLCPOS01-0 (rev. 00.32 or higher).

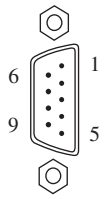
20. IF4 - B&R ONLINE INTERFACE

For communication with the programming device, the Compact PLC uses an on-line interface. The on-line interface is a TTY interface which runs at 62.5 kBaud and is only used for on-line operation with the programming device.

The connection to the programming device is made with an online cable (BRKAOL-0).

Pin-outs

9 Pin D-Type
Connector (M)



Pin	Assignment
1	TXD
2	Reserved
3	RXD RET
4	Reset RET
5	Reserved
6	TXD RET
7	RXD
8	Reset
9	Reserved

21. IF5 - CAN BUS

21.1 CHARACTERISTICS OF THE CAN BUS

- Field bus
- Low cost
- High resistance to disturbance because of differential signal
- Bus structure
- Open system
- Fast data transfer for small data packets (up to 8 bytes)
- Error recognition via CRC (Cyclic Redundancy Check) and frame check -> hamming distance 6
- Predictable transfer time for high priority messages (real time behavior)
- Simple use

21.2 B&R AND CAN

The controller used by B&R conforms to the CAN Bus Specification 2.0B. In this way, the protocols Standard CAN and Extended CAN can be operated on one bus.

The B&R software supports the Standard CAN Identifier (11 bit) at the time being.

21.3 BUS LENGTH AND CABLE TYPE

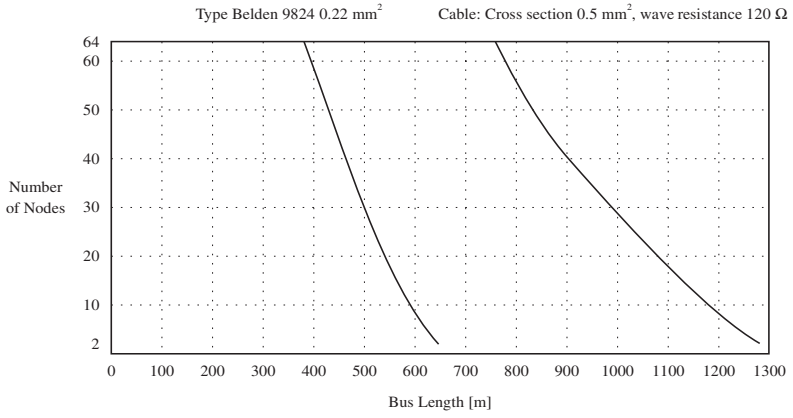
The cable type to be used is highly dependent on the bus length required and the number of nodes. The bus length is determined mainly from the bit rate.

The table on the next page contains some values for the maximum bus length in relation to the transfer rate and the synchronization jump width (SJW). The fourth column contains the permissible oscillator tolerance.

The synchronization jump width (SJW) is the factor that determines how close the controller can synchronized. The larger the synchronization jump width, the smaller the bus length.

Bit Rate [kBit/sec]	Synchronization Jump Width (SJW)	Bus Length [m]	Permissible Osc. Tolerance [%]
500	0 1 2 3	67 56 33 10	0.121 0.242 0.363 0.485
250	0 1 2 3	215 192 147 101	0.121 0.242 0.363 0.485
125	0 1 2 3	510 465 374 283	0.121 0.242 0.363 0.485
100	0 1 2 3	658 601 488 374	0.121 0.242 0.363 0.485
50	0 1 2 3	1397 [1)] 1284 [1)] 1056 [1)] 829	0.121 0.242 0.363 0.485
20	0 1 2 3	3613 [1)] 3329 [1)] 2761 [1)] 2193 [1)]	0.121 0.242 0.363 0.485
10	0 1 2 3	7306 [1)] 6738 [1)] 5602 [1)] 4456 [1)]	0.121 0.242 0.363 0.485

Relationship between number of nodes and bus length for certain cable types:



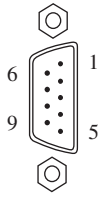
[1)] According to CiA (CAN in Automation), the maximum bus length is 1000 m.

21.4 CONNECTOR PIN ASSIGNMENTS

The Compact Control **BRCOMP2-0** (operating system version 4.1) is equipped with a CAN Bus interface that conforms to ISO-DIS 11898. The pin assignments of the connector conform to CiA DS 102-1.

Pin Assignments

9 Pin D-Type
Connector (M)



Pin	Assignment
1	
2	CAN L
3	CAN GND
4	
5	
6	
7	CAN H
8	
9	

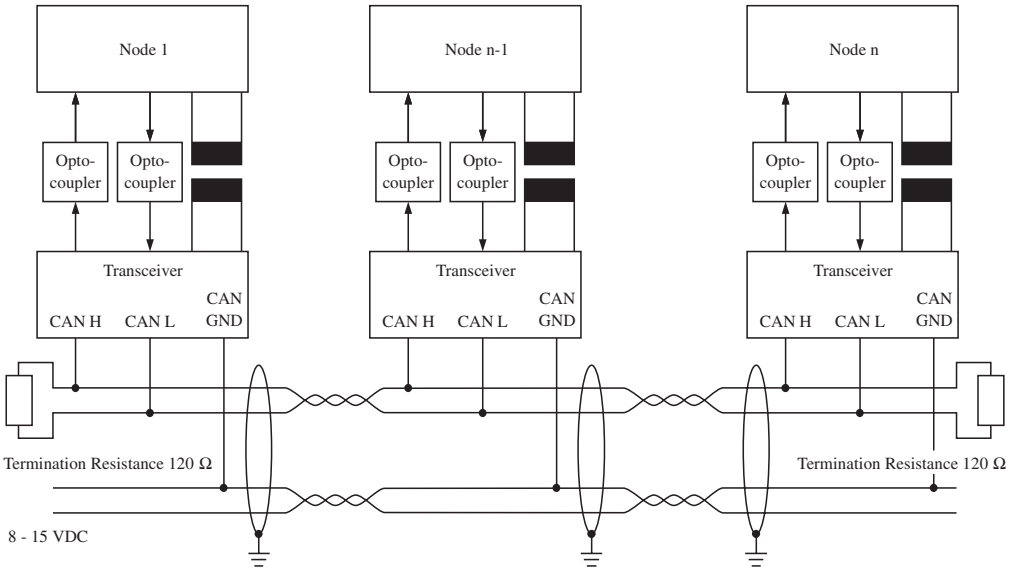
21.5 WIRING

The wiring is done in the form of a bus structure. Both bus ends are to be terminated with a termination resistance. A node can be connected to the bus with a branch line. The length of the branch line may not exceed 30 cm.
A four conductor twisted pair cable is to be used for the bus cable.

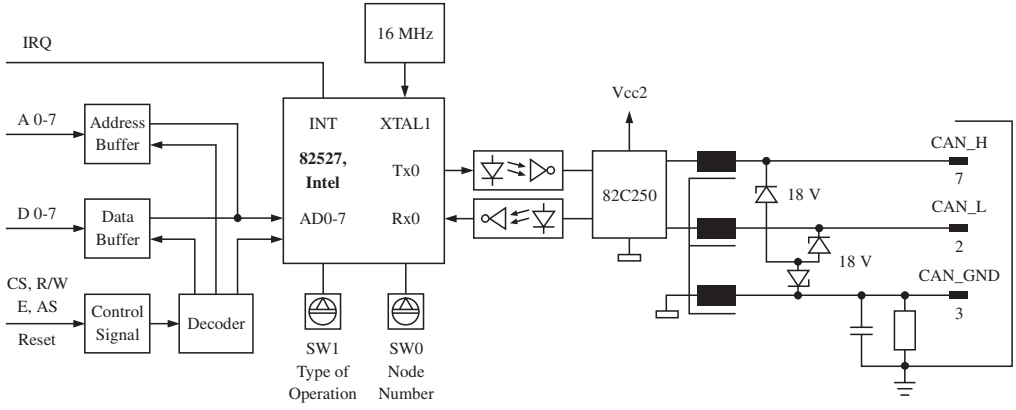
CAN Signal	CAN H	...	CAN High
	CAN L	...	CAN Low
	CAN GND	...	CAN Ground
	CAN+	...	CAN Supply 8 - 15 V

All of the CAN interfaces from B&R are supplied internally, CAN+ is not connected.

21.5.1 Wiring Principle



21.5.2 Block Diagram of a Node



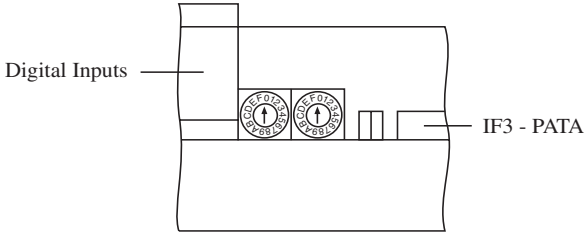
21.6 OPERATING THE CAN BUS

The CAN bus is operated with the function blocks CNSW and CNCS. The function blocks are available as standard software (on 3.5" diskettes).

Language	Model Number
German	SWSPSCAN01-0
English	SWPLCCAN01-0

21.6.1 CNSW - CAN Node Switch

The function block reads the switch settings of both hex switches (between the digital inputs and IF3 - PATA) and provides this information to the CAN Client/Server function block CNCS. Additionally, all switch settings are output directly as values between 0 - 15. The CNSW function block eases the use of the same program for different client stations.



21.6.2 CNCS - CAN Client/Server

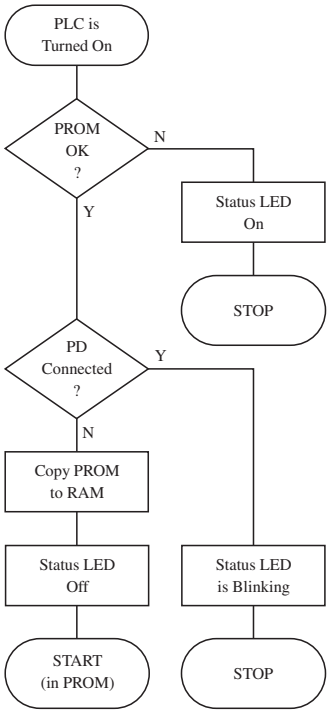
The function block allows the communication a standard CAN net (11 bit ID) via CAL/CMS services for transmitting object data. Layer or network management services and identifier distribution services are supported.
That means that CAL is a LMT/NMT/DBT slave of class 0 with static CAN ID distribution. The communication takes place via the CMS protocol for variables and for uncontrolled events.
The Client/Server services for "Read-Only Access, Basic Variable" are not implemented. Of the CMS data types, only integer as byte multiples are supported.

Transfer data (max. 8 bytes) is transferred on the bus in increasing byte order (LSB to MSB) (Little Endian).

Abbreviations

CAL	...	CAN Application Layer
CMS	...	CAN based Message Specification
LMT	...	Layer Management
NMT	...	Network Management
DBT	...	Communication Object Distributor

22. POWER-ON SEQUENCE



23. COMMAND SET

A 6303 processor (Hitachi) is implemented in the Compact PLC. This is the same processor which is used in the MINICONTROL CPU. Therefore giving full software compatibility to the next size of PLC system.

A complete description of the 6303 processor command set is provided in the "STL Instructions" booklet (MAAWLKB-E).

This booklet contains the following useful information:

- B&R and MOTOROLA Mnemonics
- Descriptions for all instructions
- The possible types of addressing and opcodes
- The possible address codes
- Instruction lengths and executions times
- Changed Flags

24. MATHEMATIC ROUTINES

The CPU of the Compact PLC is equipped with fast floating point mathematic routines. These routines are components of the operating system. They are called by means of statement list instruction mnemonics.
In addition to the basic mathematics addition, subtraction, multiplication, division and square root numerous conversion and utility programs available (e.g. for comparing and copying). Numbers are represented in standard 4 byte IEEE format. A detailed description of these mathematic routines can be found in the MAAWLKB-E booklet.

NOTE: Mathematic routines may not be used in the interrupt routines.

Number Formats

Format	Range
IEEE-Floating point format 31 24 23 16 15 8 7 0 S EXP S MANTISSA S ... Sign EXP ... 7 bit exponent MANTISSA ... 23 bit mantissa	$-9.22 * 10^{18}$ to $-9.22 * 10^{-18}$ and $9.22 * 10^{-18}$ to $9.22 * 10^{18}$
Absolute with sign, long 31 24 23 16 15 8 7 0 S ABSOLUTE VALUE	$\pm 2.15 * 10^{9}$
Absolute with sign, short 15 14 8 7 0 S ABSOLUTE VALUE	±32767
Integer, long 31 24 23 16 15 8 7 0 S 2's complement	$\pm 2.15 * 10^{9}$
Integer, short 15 14 8 7 0 S 2's complement	-32768 to +32767

Instr.	Function	Source or Operands	Destination or Result	Execution time in µsec.	Possible Error Messages												
					1	2	3	4	5	6	7	8	9	10	11	12	13	14
MADD	OP1 := OP1 + OP2	OP1, OP2	OP1	209/690	●	●				●	●					●		
MSUB	OP1 := OP1 - OP2	OP1, OP2	OP1	219/700	●	●				●	●					●		
MMUL	OP1 := OP1 * OP2	OP1, OP2	OP1	209/803	●	●				●	●					●		
MDIV	OP1 := OP1 / OP2	OP1, OP2	OP1	190/1980	●	●	●			●	●					●		
MSQR	OP1 := SQR(OP1)	OP1	OP1	71/8065	●	●				●	●	●				●		
MSGN	OP1 := OP1 * (-1)	OP1	OP1	85/85														
MCOP	OP2 := OP1	OP1	OP2	46/46														
MEXG	OP1 ↔ OP2	OP1, OP2	OP2, OP1	76/76														
LAL1	Load OP1, abs. with sign 4 byte	(X)	OP1	190/339					●									
LAL2	Load OP2, abs. with sign 4 byte	(X)	OP2	190/339					●									
LAW1	Load OP1, abs. with sign 2 byte	AD	OP1	83/250														
LAW2	Load OP2, abs. with sign 2 byte	AD	OP2	83/250														
LIL1	Load OP1, int. 4 byte	(X)	OP1	197/381					●									
LIL2	Load OP2, int. 4 byte	(X)	OP2	194/378					●									
LIW1	Load OP1, int. 2 byte	AD	OP1	87/260														
LIW2	Load OP2, int. 2 byte	AD	OP2	84/257														
LF1	Load OP1, IEEE	(X)	OP1	88/125	●					●	●					●		
LF2	Load OP2, IEEE	(X)	OP2	88/125	●					●	●					●		
CAF	ASCII - IEEE	(X)	OP1	280/2140	●					●	●		●					
SAW	Store OP1, abs. with sign 2 byte	OP1	AD	158/373				●								●		
SAL	Store OP1, abs. with sign 4 byte	OP1	(X)	169/408				●								●		
SIW	Store OP1, int. 2 byte	OP1	AD	158/380				●								●		
SIL	Store OP1, int. 4 byte	OP1	(X)	172/424				●								●		
SFX	Store OP1, IEEE	OP1	(X)	43/43														
CFA	OP1 - ASCII	OP1	(X)	352/7310	●			●		●	●					●		
CFA0	OP1 - ASCII with leading zeros	OP1	(X)	310/7190	●			●		●	●					●		
CFEA	OP1 - ASCII with exp.	OP1	(X)	570/7140	●					●	●					●		
SFM1	Store OP1 in memory 1	OP1	MEM1	60/60														
SFM2	Store OP1 in memory 2	OP1	MEM2	60/60														
SFM3	Store OP1 in memory 3	OP1	MEM3	60/60														
RFM1	Load OP2 from memory 1	MEM1	OP2	56/56														
RFM2	Load OP2 from memory 2	MEM2	OP2	56/56														
RFM3	Load OP2 from memory 3	MEM3	OP2	56/56														
FM2B	Multiplication 2 x 2 byte	(X) int. 2 byte, AD	(R1048, 1049) 4 byte	115/191														
FM3B	Multiplication 3 x 2 byte	(X) int. 3 byte, AD	(R1048, 1049) 5 byte	156/270														
FM4B	Multiplication 4 x 2 byte	(X) int. 4 byte, AD	(R1048, 1049) 6 byte	192/344														
CBCD	Binary - BCD	(AD) abs. 3 byte	(X) BCD 3 byte	192/1180				●										
CBIN	BCD - Binary	(AD) BCD 3 byte	(X) abs. 3 byte	112/223														
CIA	Binary - ASCII	(R1048, 1049)	(X)	380/2020				●										
CIA0	Binary - ASCII with leading zeros	(R1048, 1049)	(X)	310/1960				●										
CBPP	Binary - physical (parameters)	(X)	(R1048, 1049)	2500/6700	●					●	●					●		
CBPQ	Binary - physical (fast)	AD, (X)	AD, OP1	780/1700	●					●	●					●		
CPBQ	Physical - binary (fast)	AD, (X)	AD, OP1	780/1500	●					●	●					●		
CBP	Binary - physical	(R1046, 1047), (X)	(R1048, 1049), AD, OP1	3400/8300	●					●	●					●		
CPB	Physical - binary	(R1046, 1047), (X)	(R1048, 1049), AD, OP1	3400/8300	●					●	●					●		
CIM	Inch - metric	(R1046, 1047), AD	(X), AD	307/472													●	●
CMI	Metric - Inch	(R1046, 1047), AD	(X), AD	307/472													●	●
FCOP	Copy data block	(X), AD	(R1048, 1049)															
FSMB	Load memory area with byte values	(X), AD, R1052	(X)	48 + L * 12														
FSMW	Load memory area with word values	(X), AD, R1052	(X)	48 + L * 14														
FCLR	Clear memory area	(X), AD	(X)	48 + L * 12														
MCMP	Compare OP1 with OP2	OP1, OP2		201/223												●		
MHIL	If OP1 > OP2 then OP1 := OP2	OP1, OP2	OP1	215/271												●		
MLOL	If OP1 < OP2 then OP1 := OP2	OP1, OP2	OP1	215/271												●		

Error Messages

The error messages in the table (see previous page) which are marked with ●, are possible for the respective function on the same line. If an error occurs during the execution of a routine, the Carry Flag is set and the error number is put into register R 1024.

Error number	Description
1	Range overflow during computation
2	Range underflow during computation
3	Division by 0
4	Range overflow during number format conversion
5	Trimming of Lower Significant Byte (LSB) while loading 4 Byte Mantissa
6	Range overflow while loading numbers
7	Range underflow while loading numbers
8	Negative operand for square root computation
9	Invalid character for string conversion routine
10	Not used
11	Invalid command (trap error set off)
12	Number out of range
13	Exponent error in metric-inch or inch-metric conversion
14	Data overflow in metric-inch or inch-metric conversion

Operands and Registers

Registers	Function
R 1024	Error number
R 1025	Reserved
R 1026 to R 1029	Operand 1 (OP1)
R 1030 to R 1033	Operand 2 (OP2)
R 1034 to R 1037	Interim memory 1 (MEM1)
R 1038 to R 1041	Interim memory 2 (MEM2)
R 1042 to R 1045	Interim memory 3 (MEM3)
R 1046 to R 1047	Source address
R 1048 to R 1049	Destination address
R 1050 to R 1051	Length
R 1052 to R 1053	Data

25. SYSTEM REGISTERS AND FLAGS

Several registers and flags are reserved for the operating system. These may not be used or may only be partially utilized by application programs:

Registers: R 0800 to R 1499
Flags: F 800 to F 999

Flag addresses beginning at F 800, which are used for operating system functions are entered with the addresses S Dxx or T Dxx:

Addresses	Entered as [1]
F 800	S D00
F 801	S D01
:	:
F 899	S D99
F 900	T D00
F 901	T D01
:	:
F 999	T D99

[1] The programming device permits the entry of the F address, but after [↵] is pressed, the address is automatically converted to S Dxx or T Dxx format.

Enter: F 820
After [↵] is modified to: S D20

Enter: F 980
After [↵] is modified to: T D80

The registers and flags which can be used in limited form by applications programs are described below:

Permitted Access		Address(es)	Function
Read	Write		
		R 0800 to R 0863	Pre-multiplexer for software timers
●		R 0899	First scan register
		R 0900 to R 0963	Counter for software timers
●	●	R 0968, R 0969	HW counter interrupt or interrupt input vector
		R 0972, R 0973	Timer interrupt vector
		R 0974, R 0975	Timer interrupt time
●	●	R 0976, R 0977	IRQ vector
		R 0978, R 0979	Trap vector
●	●	R 0980 to R 0987	Real-time clock
●	●	R 0990	Breakpoint special function
		R 0991 to R 0993	Counter / Multiplexer
		R 0998, R 0999	Run-time watchdog
●	●	R 1000 to R 1013	Analog inputs / analog outputs
●	●	R 1024 to R 1053	Operands and memories for math routines
		R 1054 to R 1499	Reserved for standard function blocks
●	●	S D00 to S D63	Start flags for software timers
●	●	S D85, S D86	Control bits for real-time clock
●		T D00 to T D63	Software times
●		T D64	First Scan Flag
●		T D80 to T D83	Time cycles
●		T D90 to T D93	Time pulses
●		T D99	Battery test

26. FIRST SCAN FLAG

The first scan flag is a flag (T D64), which is set to 1 automatically during the first program cycle and otherwise is 0. The first scan flag is used for program initialization. Register R 0899 also provides the first scan functions:

T D64	First Scan Flag (1 = first program cycle)
R 0899	First Scan Register (1 = first program cycle)

Example

```
INIT  LDAA  Z D64       First Scan
      BEQ   INIR        Branch, if already initialized
      :
      :                 Initialization
      :
INIR  RTS
```

In the ladder diagram the first scan flag can be connected to the enable input of function blocks which are supposed to be executed only once during the program's first cycle.

NOTE: The B&R PROgramming SYStem command "XFER" allows programs to be downloaded to the CPU without interrupting the application program running in the RAM of the CPU.
The user must switch to the new program manually with a command from the programming device. In this case neither the first scan flag or the first scan register are set during the first program cycle of the new program!

Since the First Scan flag and register are not set, caution is bid when using interrupt routines or B&R MININET.
The address of the interrupt routine is not updated after a program change, if the definition is made in the initialization routine.

27. LITHIUM BATTERY

The Compact PLC is provided with a lithium battery. This is required in order to provide back-up for the SRAM and the real-time clock.

NOTE: Lithium batteries fall into the catagory of hazardous waiste. Please be sure to follow the guidelines for disposal that are valid for your area.

Buffer Life

3 years at 70 °C
6 years at 40 °C

NOTE: Take note that the temperature within the housing of the unit can sometimes be up to 20 °C above the surrounding temperature.

Power Monitor

The state of the battery can checked with bit 1 of the Inport Address $3480 (see section "34. Inport Address $3480") (hardware monitoring).

0 ... Battery OK (Voltage > 2.55 V)
1 ... Battery dead (Voltage < 2.45 V)

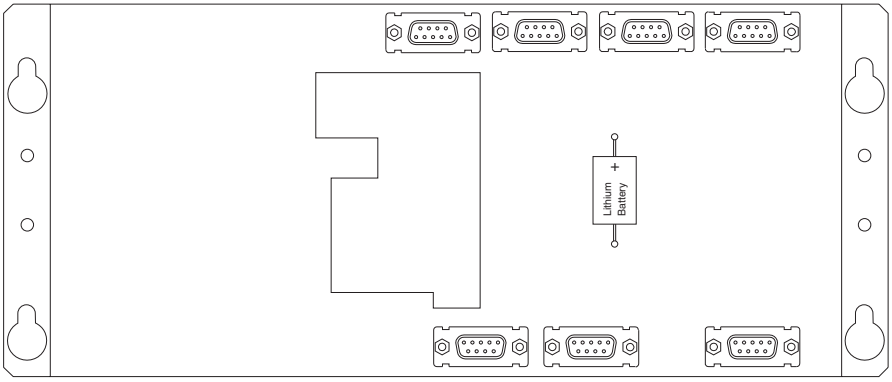
The bit status can be placed on an output. If you are using an operating panel, a respective message can also be output.

For reasons of compatibility, the state of the battery can also be checked with the register T D99. However, this method should not be used since it is software monitoring!

Changing the Battery

To change the battery, all power must be switched off.

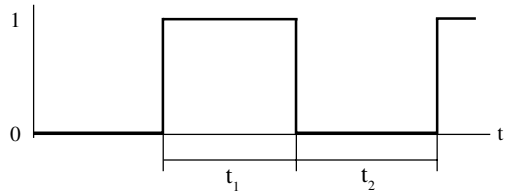
NOTE: Data stored in RAM and the clock time is lost. By using the B&R PROgramming SYStem "UPDL" command the data can be saved however.



28. TIME CYCLES

Time cycles are flags which are generated by the operating system:

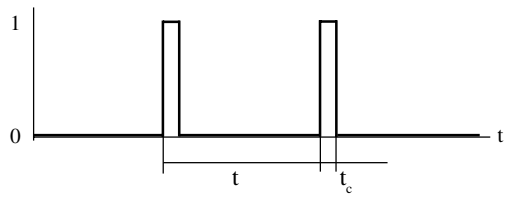
Address	t_1	t_2
T D80	10 msec	10 msec
T D81	40 msec	60 msec
T D82	0.4 sec	0.6 sec
T D83	4 sec	6 sec



29. TIME PULSES

Time pulses are flags which are automatically set to 1 for the length of one program cycle by the operating system.

Address	t
T D90	10 msec
T D91	100 msec
T D92	1 sec
T D93	10 sec



t_c ... Program cycle

30. REAL-TIME CLOCK

If the PLC is switched off, the real-time clock keeps running (buffered by the lithium battery).

Time registers (all specifications in BCD):

R 0980	1/100 Seconds ($00 to $99)
R 0981	Seconds ($00 to $59)
R 0982	Minutes ($00 to $59)
R 0983	Hours ($00 to $23)
R 0984	Day ($01 to $31)
R 0985	Month ($01 to $12)
R 0986	Year ($00 to $99)
R 0987	Day of the week (0 to 6)

Control of the real-time clock is done by means of two memory locations:

S D85	Clock ON/OFF (1 = ON)
S D86	Set clock ON/OFF (0 = Set clock ON)

To set the real-time clock (the clock must be switched on, i.e. S D85 must be 1):

- Set clock ON (clear S D86)
- Load clock time registers R 0980 to R 0987 with time/date
- S D86 is set automatically at the end of the next program cycle

31. SOFTWARE TIMERS

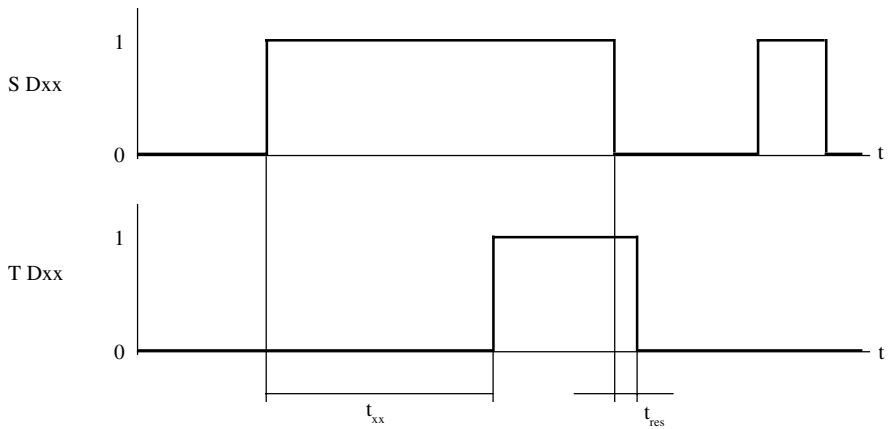
The CPU of the Compact PLC has 64 timers that work on an On-Delay. Every software timer consists of the following addresses:

S Dxx — Starts the software timer. By writing a 1 to this flag, the software timer xx (xx = 00 to 63) is started. This flag can also be read (e.g., to determine whether a certain software timer has been started or not).

T Dxx — Result. If this flag contains 1, then the associated software timer has elapsed. This is a read-only flag. It is reset when the S Dxx start flag is cleared.

Txx n"nn — Time definition. The Txx instruction defines the duration of the software timer in seconds and 100ths of seconds. This instruction must always be executed; thus it is usually placed at the beginning of an application program.

Time Sequence



After the software timer xx has been started by writing 1 to the start flag S Dxx and the elapse of the time t_{xx} set in the time definition Txx, the timer flag T Dxx is also set to 1.

After the start flag S Dxx has been reset, timer flag T Dxx is reset in the next program scan by the time definition Txx. In the worst case the reset time t_{res} can be one program cycle long.

Example 5.5 seconds after pressing a key (I 042) an motor (O 058) is to be started. Another key (I 043) is used to stop the motor:

```
0000        T10         5"50        Time definition
0001        LDAA   N    I 042       START key
0002        PRS         F 100       Pos. edge of I 042
0003        EORA        F 100       Pos. edge of I 042
0004        RST         F 100       Pos. edge of I 042
0005        PRS         S D10       Start Motor offset
0006        LDAA        I 043       STOP key
0007        RST         S D10       Start Motor offset
0008        LDAA        T D10       Motor offset
0009        STAA        O 058       Motor
0010        END
```

The same program example can also be done in ladder diagram:

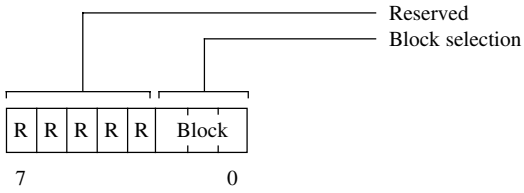
```
0000        T10         5"50        Time definition
0001        JSR         LAD1        Ladder diagram call
0002        END
```

```
!  I 042     F 100                                                    S D10
00 --I  I---+---I+I---+---------+---------+---------+---------+---------+---(P)---
!  M START   EDGE                                                   M OFFS.
!
!  I 043                                                              S D10
01 --I  I---+---------+---------+---------+---------+---------+---------+---(X)---
!  M STOP                                                             M OFFS.
!
!   T D10                                                             O 058
02 --I  I---+---------+---------+---------+---------+---------+---------+---( )---
!  MOT.ON                                                            MOTOR
```

Time definition Txx must be executed one time for every program cycle. If it is not executed, the function of the software timer cannot be guaranteed, if it is executed more than once in a program cycle, then the time is incorrect.

Each software timer occupies one register between R 0800 and R 863 which serves a pre-scalar and another register from R 0900 to R 0963 which serves as a counter. The time definition Txx is a software interrupt that takes approximately 0.5 msec (note that there will be an effect on the program scan when numerous software timers are used!).

32. INPORT/OUTPORT ADDRESS $3400



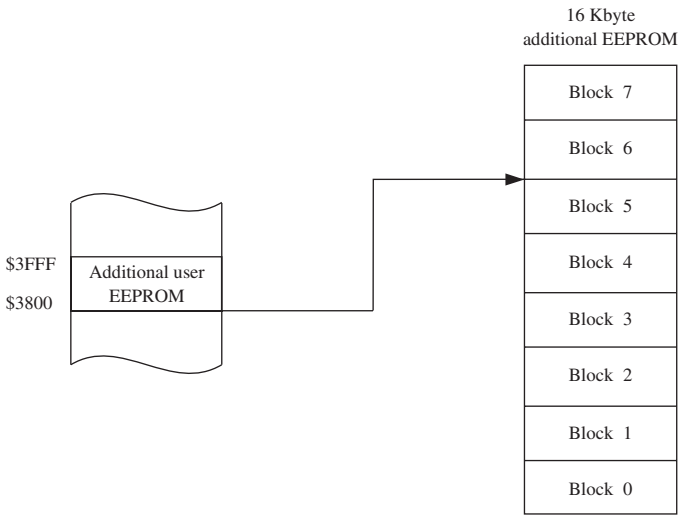
Reserved	Reserved for later use.
Block selection	With the bits 0 - 3, the desired 2 Kbyte memory area in the additional EEPROM can be selected. 0 ... Block 0 1 ... Block 1 : : : : 7 ... Block 7 For a more detailed description see section "33. Additional User EEPROM".

33. ADDITIONAL USER EEPROM

The user is provided with 16 Kbytes of additional EEPROM memory. It is split up into 8 blocks of 2 Kbytes each. The desired block is defined with the first 3 bits of the Inport/Outport byte (address $3400).

Example Definition of block 6 of the additional EEPROM.

```
LDD   # $3400          Load accu. D with $3400
XGDX                   Set index register to address $3400
LDAA  X 000            Load contents to accu. A
ANDA  # %11111000      Clear bits 0 to 2
ORAA  # 006            Define block 6
STAA  X 000            Store accu. A to address $3400
```



The user can access the selected block of the additional EEPROM via addresses $3800 to $3FFF.

33.1 READING DATA

To read data from a block, the STL macro DFEE is used

Send parameters:	Source	...	Block number in R 0881 Offset to address $3800 in R 0882&
	Destination	...	Index register
	Data length	...	In accumulator D
Return parameters:	No error:		
	Carry = 0	...	Data transfer OK
	Error:		
	Carry = 1	...	The sum of Offset and Data length is outside of the valid range (> $3FFF)
Used registers:	R 0866&	...	Source address
	R 0868&	...	Destination address
	R 0884&	...	Current data length

```
DFEE  STD   R 0884 DATA 04          Current data length
      STX   R 0868 & DEST 0
      LDD   R 0882 DATA 02          Offset
      ANDA  # %00000111             Limited to 2 Kbytes / Block
      ADDD  # $3800                 Start address EEPROM-Block
      STD   R 0866 & SOURCE 0
      ADDD  R 0884 DATA 04          Current data length
      SUBD  # $3FFF
      BMI   DFE0
*----------------------
      SEC                           Data outside of valid range
      RTS
*----------------------
DFE0  LDD   # $3400                 Inport/Outport Address
      XGDX
      LDAA  R 0881 DATA 01
      ANDA  # %00001111             Block number limited to 0 ... F
      STAA  R 0881 DATA 01
      LDAA  X 000
      ANDA  # %11110000
      ORAA  R 0881 DATA 01
      STAA  X 000                   Select block
*---
```

```
      LDD   R 0884 DATA 04          Current data length
      LSRD
      BCC   DFE1                    No odd data lengths
      STD   R 0884 DATA 04          Current data length
      LDX   R 0866 & SOURCE 0
      LDAA  X 000                   Read first byte
      INX
      STX   R 0866 & SOURCE
      LDX   R 0868 & DEST 0
      STAA  X 000                   Store first byte
      INX
      STX   R 0868 & DEST 0
      LDD   R 0884 DATA 04          Current data length
*---
DFE1  STD   R 0884 DATA 04          Current data length
      BEQ   DFE2                    Finished
      LDX   R 0866 & SOURCE 0
      LDD   X 000                   Read data
      INX
      INX
      STX   R 0866 & SOURCE 0
      LDX   R 0868 & DEST 0
      STX   X 000                   Store data
      INX
      INX
      STX   R 0868 & DEST 0
      LDD   R 0884 DATA 04          Current data length
      SUBD  # 00001
      JMP   DFE1                    Not all data copied
*----------------------
DFE2  CLC                           Data transfer OK
      RTS
```

Example Starting at address $3A00, 50 bytes are read from block 4. The data is stored from R 2000 onward.

```
      LDAA  # 004                   Block number 4
      STAA  R 0881 DATA 00
      LDD   # $0200                 Offset to $3800 (Source adr. = $3A00)
      STD   R 0882 DATA 01
      LDX#  R 2000 Dest.addr.
      LDD   # 00050                 Data length
      JSR   DFEE                    Read data from additional EEPROM
      BCC   OK
      :                             ERROR Check
      :
```

33.2 WRITING DATA

When writing data to an EEPROM, remember that there is a slight delay in comparison to writing to a flag or register.

To write data to a block, the STL macro DTEE is used

The STL macro DTEE is used for programming parameter data which does not change while a facility is in operation.

NOTE: If the application program is running in the EEPROM, the additional EEPROM cannot be programmed.

If data is programmed in the EEPROM, the rest of the program is not processed!
For writing to the EEPROM during a program cycle, the function block MCEE can be used. This is stored on the "Utility.01" diskette of standard software package 1 (SWPLCSTD01-0 Rev. 00.30 or higher).

Send parameters:	Source	...	Index register
	Destination	...	Block number in R 0881
			Offset to address $3800 in R 0882&
	Data length	...	In accumulator D
Return parameters:	No error:		
	Carry = 0	...	Data transfer OK
	R 0881	...	Block number of the next free byte
	R 0882&	...	Offset to $3800 of the next free byte
	Error:		
	Carry = 1	...	Data transfer faulty
	Accu. A	...	Error number:
			1 - Data length greater than free memory
			2 - EEPROM defect
	R 0881	...	Block number of the defect memory location
	R 0882&	...	Offset to $3800 of the defect memory location
Used registers:	R 0880	...	Run-time counter
	R 0884&	...	Current data length
	R 0886&	...	Source pointer

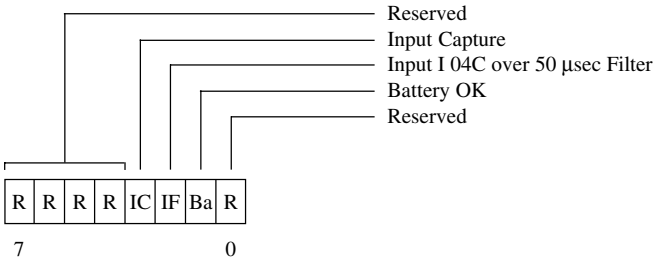
```
DTEE  STD   R 0884 DATA 04          Current data length
      STX   R 0866 & SOURCE 0
      LDD   R 0882 DATA 02          Offset
      ANDA  # %00000111             Limit to 2 Kbytes / Block
      ADDD  # $3800                 Start address of EEPROM block
      STD   R 0882 DATA 02
      ADDD  R 0884 DATA 04
      SUBD  # $3FFF
      BLS   DTE5
*----------------------
      LDAA  # 001                   ERROR ... Data length greater than
      SEC                                     free memory
      RTS
*----------------------
DTE5  LDAA  R 0998 CYCLE TIME COUNTER
      STAA  R 0886 DATA 06
DTE3  LDAA  R 0886 DATA 06
      STAA  R 0998 CYCLE TIME COUNTER
*
      LDD   # $3400                 Inport/Outport Address
      DXR
      LDAA  R 0881 DATA 01
      ANDA  # %00001111             Limit block number to 0 ... F
      STAA  R 0881 DATA 01
      LDAA  X 000
      ANDA  # %11110000
      ORAA  R 0881 DATA 01
      STAA  X 000                   Block selection
*
      LDX   R 0866 & SOURCE 0
      LDAA  X 000                   Read data
      LDX   R 0882 DATA 02
      STAA  X 000                   Store current copy data
*---
      PSHA
      LDD   # 01500
DTE0  SUBD  # 00001                 Wait state
      BNE   DTE0
      CBA
*---
      CLR   R 0880 DATA 00          Reset run-time counter
DTE1  LDAB  X 000                   Compare EEPROM data with current
      CBA                           copy data
      BEQ   DTE2
      INC   R 0880 DATA 00          Increase run-time counter
      LDAB  R 0880 DATA 00
      CMPB  # 200                   Compare with run-time MAX
      BCS   DTE1
*----------------------
      LDAA  # 002
      SEC                           ERROR ... EEPROM defect
      RTS
*----------------------
```

```
DTE2  LDD   R 0882 DATA 02          Increase MEM address pointer
      ADDD  # 00001
      STD   R 0882 DATA 02
      LDX   R 0866 & SOURCE 0
      INX
      STX   R 0866 & SOURCE 0
      LDD   R 0884 DATA 04          Current data length
      SUBD  # 00001                 All data copied ?
      STD   R 0884 DATA 04
      BEQ   DTE4
      JMP   DTE3
DTE4  CLC                           Data transfer OK
      RET
```

Example Starting at address $3B00, 40 bytes are written to block 7. The data to be written is stored in the memory starting from R 2500.

```
LDAA  # 007                   Block number 7
STAA  R 0881 DATA 00
LDD   # $0300                 Offset to $3800 (dest. = $3B00)
STD   R 0882 DATA 01
LDX#  R 2500 Source address
LDD   # 00040                 Data length
JSR   DTEE                    Write data to additional EEPROM
BCC   OK
:                             ERROR evaluation
:
```

34. INPORT ADDRESS $3480



Reserved	Reserved for later use.
Input Capture	The interrupt source for the Input Capture Input is determined with the mode register $2121 (see section "14.6.3 Mode Register $2121"). This bit indicates the status of input I 04D or the zero signal of hardware counter 0.
Input I 04C	Input I 04C is read through a 50 µsec filter (fast input).
Battery OK	This bit is used for monitoring the lithium battery. The state of the battery is checked using hardware monitoring. 0 ... Battery dead (Voltage < 2.45 V) 1 ... Battery OK (Voltage > 2.55 V)
Reserved	Reserved for later use.

35. WATCHDOG

35.1 SOFTWARE WATCHDOG

The software watchdog monitors the maximum permitted program cycle time of 100 msec. If a program cycle has not ended when this time elapses, the application program is stopped and a software reset is executed (all outputs are reset).
A run-time error is indicated in the debugger of the programming device and through the status LED.

35.2 HARDWARE WATCHDOG

The CPU of the Compact PLC has a hardware watchdog, which can bring the system to a secure state of operation even if the CPU fails completely.

36. TIMER INTERRUPT ROUTINES

Independent of the length of an application program, an interrupt is triggered every 10 msec and the timer interrupt routine is called. This operating system function is used for security and diagnostic functions as well as the generation of software timers, clock functions, time cycles and time pulses.

The timer interrupt vector (the address of the timer interrupt routine) sits at R 0972&. The timer interrupt time is stored in R 0974& (units µsec.). Neither the timer interrupt vector nor the timer interrupt time may be changed by the application program.

In addition to the operating system functions, the user can have one or two program segments executed on a time basis (user timer interrupt routines). The timer interrupt handlers $US1 and $US2 are used for this purpose with parameters:

AA	Desired time interval in msec.
R	Start address of user timer interrupt routine

Call `JSR $US1` or `JSR $US2`

The user timer interrupt routine is terminated with RTS. Independent of the time interval chosen for the user timer interrupt routine, the operating system timer interrupt routine is executed every 10 msec.

NOTE: Timer interrupt routines are not executed if the PLC is in HALT state.

Calling lengthy timer interrupt routines too often can extend the program scan time and cause disturbance in the system. The sum of the execution times of the two user timer interrupt routines must not exceed 300 µsec.

Operating system math routines must not be used in timer interrupt routines.

To switch off an active timer interrupt routine, load the AA with 0 and call the interrupt handler ($US1 or $US2) again.

Example Every 3 msec a descending counter is to be read and compared to 10 000. If the counter falls below this value, a flag is to be set. The timer interrupt handler $US1 is only called once in the initialization:

```
INIT  LDAA  T D64       First Scan
      BEQ   INIR
      LDAA  # 003       3 msec
      LDXL  TEST        Address of the interrupt routine
      JSR   $US1
INIR  RTS

TEST  JSR   READ        Read counter state
      SUBD  # 10000     Compare with 10,000
      BCC   TESR
      SET   O 040       Counter low !
TESR  RTS
```

37. ERROR MESSAGES

The CPU is equipped with extensive security and diagnostic functions (e.g. Program checksum tests at Power-on). In the event of an error the application program is stopped, the status LED is switched on and a software reset is executed. All digital outputs are reset, all analog outputs are set to 0 V or 0 mA. If a programming system is connected, an appropriate error message is displayed in the debugger (e.g. RUNTIME ERROR).

The following table provides a summary of all possible errors on the Compact PLC:

Message	Description/Cause	Solution
Transmission error during download	An error occurred while downloading a program from the programming device to the PLC. **Possible causes:** The Online link between the programming device and the PLC is affected by electromagnetic disturbance.	Perform the download again.
Checksum error after RUN	An application program downloaded with RUN is indicated as having the wrong checksum in the RAM of the PLC. **Cause:** Program memory defect.	Download the program again; if the error occurs repeatedly, replace the EE32 module.
Checksum error	The checksum of the application program is incorrect after a power-on or reset. **Possible causes:** For PROM programs the PROM memory is defective, for RAM programs either the battery is low or dead or there is a software problem which overwrites the application program.	Download the application program again. If the error occurs repeatedly, check the battery, examine the application program for errors, or replace the application program memory module.
Runtime error	The permissible program cycle time of 100 msec was exceeded. **Possible causes:** Software error, too many program loops, an endless loop.	Check program again and correct errors.
Pointer error	During the checksum test at power-on, it was determined that the operating system vectors are incorrect. **Possible causes:** see "Checksum error".	See "Checksum error".

Message	Description/Cause	Solution
Communication error	An error occurred during communication between the programming device and the PLC (RUN, debug). **Possible causes:** The Online link between the programming device and the PLC is affected by electromagnetic disturbance.	Perform the download again.
Stack pointer error	At the end of a program scan (END) the stack pointer is incorrect. **Possible causes:** Error in the application program (subroutine not terminated with RTS, error in use of system stack for data storage).	Correct program error.
Trap error	Unknown processor instruction. **Possible causes:** Error in the application program (e.g., indexed jump to data region)	Correct program error.

38. INDEX

A

Accessories
- Relay Expansion Card 193
- Standard Software 137
- Terminal Blocks 137

Additional Application EEPROM 237
Addresses
- Inport Address $3480 243
- Inport/Outport Address $3400 110

AINJ - Analog Input Compact Control 164
Analog Inputs 153
- AINJ - Analog Input Compact Control 38
- Analog Conversion 167
- Channel Refresh Time 168
- Input Circuit 157
- Operating System Routine 167
- Pin Assignments 157
- Shield Grounding 158
- Software Operation 161
- Technical Data 156
- Wiring Schematic 158

Analog Outputs 169
- Analog Conversion 174
- AOTF - Analog Output Compact Control 173
- Operating System Routine 174
- Output Circuit 170
- Pin Assignments 170
- Shield Grounding 171
- Software Operation 172
- Technical Data 169
- Wiring Schematic 171

AOTF - Analog Output Compact Control 173
Application EEPROM 237

B

Battery 230

C

CAN Bus (BRCOMP2-0) 217
Clock (real time) 233
CMDC - Evaluation of Counter (BRCOMP) 178
CMDD - Evaluation of Counter 0 (BRCOMP) 180
CNCS - CAN Client/Server 221
CNSW - CAN Node Switch 221
Compact MMI P120 and MMI P121 202
Components 142
Control Register $2120 184
Counter 175
- Actual Value 183, 191
- CMDC - Evaluating Counter (BRCOMP) 178
- CMDD - Evaluating Counter 0 (BRCOMP) 180
- Control Register $2120 184
- Function 176, 186
- Hardware Counter 0 176, 182, 192
- Initialization 187
- Initialization Register $2122 185, 187
- Interrupt Input 175, 176, 182
- Interrupt Mode 182
- Mode Register $2121 184, 188
- Operating System Addresses 183
- Operating System Routines
  - Addresses 185
  - Call 188
- Operation 177
- Predefined Value 183, 190
- Reference Input 175, 176, 189
- Two Channel Counter 176
- Wiring Schematic 177

D

Digital Inputs 147
- Input Circuit 148
- LEDs 145
- Technical Data 147
- Wiring Schematic 148

Digital Outputs 149
- LEDs 145
- Output Circuit 150
- Protective Circuits 152
- Switching Inductive Loads 151
- Technical Data 149
- Wiring Schematic 150

E

EEPROM 237
Error Messages 247

F

First Scan Flag 229
Function Blocks
- AINJ - Analog Input Compact Control 164
- AOTF - Analog Output Compact Control 173
- CMDC - Evaluation of Counter (BRCOMP) 178
- CMDD - Evaluation of Counter 0 (BRCOMP) 180
- CNCS - CAN Client/Server 221
- CNSW - CAN Node Switch 221

Fuse 143, 144

H

Hardware Watchdog 244

I

IF1 - RS232 Interface 201
IF2 - RS232/RS485 Interface 203
IF3 - PATA/SSI Interface 215
IF4 - B&R Online Interface 216
IF5 - CAN Bus (BRCOMP2-0) 217
Initialization Register $2122 185
Inport Address $3480 243
Inport/Outport Address $3400 236
Input Capture 243
Instruction Set 222
Interfaces
- IF1 - RS232 Interface 201
- IF2 - RS232/RS485 Interface 203
- IF3 - PATA/SSI Interface 215
- IF4 - B&R Online Interface 216
- IF5 - CAN Bus (BRCOMP2-0) 217
- Interface Interrupt 211
- Operation 206

Interrupt Input 175, 176, 182

L

LEDs 145
- Digital Inputs 145
- Digital Outputs 145
- Operating Status 146
- Supply 145

Lithium Battery 230

M

Mathematics Routines 223
Measurements
- BRCOMP 139
- Relay Expansion Card 195

MMI P120 202
MMI P121 202
Mode Register $2121 184
Model Numbers 136
Mounting
- BRCOMP 140
- Relay Expansion Card 196, 198

Mounting Rail 140

O

Online Interface 216

P

PATA/SSI Interface (IF3) 215
Power-On 222
Power-on Sequence 222
Processor 138, 222

R

Real Time Clock 233
Reference Input 49, 50
Relay Expansion Card 193
- Compact Control and Relay Expansion Card 198
- Connection with the Compact Control 199
- Installation Guidelines 196
- Measurements 195
- Mounting 196
- Pin Assignments 194
- Relay Groups 193
- Relay Supply 199
- Switching Circuit 200
- Technical Data 194

RS232 Interface (IF1) 201
RS232/RS485 Interface (IF2) 203

S

Software for BRCOMP 137
Software Times 234
Software Watchdog 244
SSI/PATA Interface 215
Standard Software for BRCOMP 137
Status LED 146
Supply Voltage 143
    LED DCOK 145
System Registers and Flags 227

T

Technical Data
    BRCOMP 138
    Relay Expansion Card 194
Terminal Blocks (accessories) 137
Time Clock 232
Time Pulse 232
Timer Interrupt Routines 245
Two Channel Counter 176

U

Voltage Monitoring 243
Voltage Monitoring Battery 230

W

Watchdog
    Hardware 244
    Software 244

X

“XFER” - Transfer Program 229