
POSITIONIER- UND BAHNSTEUERUNG

MCU-3000

MCU-3100

OPTIONEN-HANDBUCH / OHB

Stand: 22.04.2015, ab Disk V2.53VD

Board-Revision: MCU-3000 Rev. D, OPMF-3001 Rev. E
MCU-3100 Rev. C, OPMF-3100 Rev. B

1	Optionsprint OPMF-3001 / OPMF-3100	7
1.1	Kurzbeschreibung	7
1.2	Software	8
1.2.1	Modifizierte Betriebssystemsoftware rwmos.elf	8
1.2.2	PCAP-Programmierung.....	8
1.2.2.1	rdAin, read analog inputs	9
1.2.2.2	rddigi, read digital inputs	10
1.2.2.2.1	Achsenqualifizierer digi für die Achskanäle 4, 5 und 6	11
1.2.2.2.2	Achsenqualifizierer digi für die Achskanäle 7 und 8	12
1.2.2.3	rddigib, read digital input bit	12
1.2.2.4	rddigo, read digital outputs	12
1.2.2.5	rddigob, read digital output bit	12
1.2.2.6	wrdigo, write digital outputs	13
1.2.2.7	wrdigob, write digital output bit	13
1.2.3	SAP-Programmierung	14
1.2.3.1	Achsenqualifizierer digi und digo	14
1.2.3.2	Achsenqualifizierer ain0 .. ain3	14
1.3	Hardware.....	15
1.3.1	Optionsprint OPMF-3001 / OPMF-3100, Anschlussbelegungen	16
1.3.1.1	Stecker X1: 50-poliger SUB-D-Steckverbinder (Stift)	16
1.3.1.2	Stecker X2: FB-Stecker X2 und Transformation auf SUB-D-Steckverbinder (Stift / Buchse).....	18
1.3.1.3	Sollwertkanäle	20
1.3.1.3.1	Sollwertkanal für Servomotorachsen	20
1.3.1.3.1.1	Pinbelegung Stecker X1, Achskanal 4	20
1.3.1.3.1.2	Pinbelegung Stecker X1, Achskanal 5	20
1.3.1.3.1.3	Pinbelegung Stecker X1, Achskanal 6	20
1.3.1.3.1.4	Pinbelegung Stecker X2, Achskanal 7	21
1.3.1.3.1.5	Pinbelegung Stecker X2, Achskanal 8	21
1.3.1.3.2	Sollwertkanal für Schrittmotorachsen	22
1.3.1.3.2.1	Pinbelegung Stecker X1, Achskanal 4	22
1.3.1.3.2.2	Pinbelegung Stecker X1, Achskanal 5	22
1.3.1.3.2.3	Pinbelegung Stecker X1, Achskanal 6	22
1.3.1.3.2.4	Pinbelegung Stecker X2, Achskanal 7	22
1.3.1.3.2.5	Pinbelegung Stecker X2, Achskanal 8	23
1.3.1.4	Pinbelegung Stecker X1, Digitale Eingänge	23
1.3.1.5	Pinbelegung Stecker X2, Digitale Eingänge	24
1.3.1.6	Pinbelegung Stecker X1, Digitale Ausgänge	24
1.3.1.7	Pinbelegung Stecker X2, Digitale Ausgänge	24
1.3.1.8	Pinbelegung Stecker X2, Freigaberelais.....	24
1.3.1.9	Impulserfassungskanäle.....	25
1.3.1.9.1	Inkrementalkoder mit invertierten Signalen (symmetrische Beschaltung).....	25
1.3.1.9.2	Inkrementalkoder ohne invertierte Signale (asymmetrische Beschaltung).....	25
1.3.1.9.3	Optische Entkopplung der Impulserfassungskanäle	26
1.3.1.9.4	Steckerbelegung für die Impulserfassungskanäle mit Inkrementalgebern	27
1.3.1.9.4.1	Steckerbelegung X1, Kanal 4	27
1.3.1.9.4.2	Steckerbelegung X1, Kanal 5	27
1.3.1.9.4.3	Steckerbelegung X1, Kanal 6	27

1.3.1.9.4.4	Steckerbelegung X2, Kanal 7	27
1.3.1.9.4.5	Steckerbelegung X2, Kanal 8	27
1.3.1.10	Steckerbelegung X2 für die Analogeingangskanäle	28
1.3.1.10.1	Steckerbelegung für die Referenzspannungsausgänge	28
1.3.2	Bestückungsdruck des OPMF-3001 Top-Layer	29
1.3.3	Bestückungsdruck des OPMF-3001 Bottom-Layer	30
1.3.4	Bestückungsdruck des OPMF-3100 Top-Layer	31
1.3.5	Bestückungsdruck des OPMF-3100 Bottom-Layer	32
1.3.6	Technische Daten des OPMF-3001/OPMF-3100	33

2 Option OPIBS34

2.1	Kurzbeschreibung des INTERBUS Master-Interface	34
2.2	Software	35
2.2.1	Modifizierte Betriebssystemsoftware rwmos.elf	35
2.2.2	Einführung	35
2.2.3	Hilfsprogramm fwsetup.exe	36
2.2.3.1	Interbus Seite: Configuration	36
2.2.3.2	Interbus Seite: SAP	38
2.2.3.2.1	Erläuterung der grafischen Bediensymbole	39
2.2.3.2.2	Erläuterung der Interbus SAP Bildschirmtextausgabe	40
2.2.3.3	Interbus Seite: Diagnostics	41
2.2.4	Interbus SAP-Programmierung	42
2.2.4.1	Interbus Status-Informationen und Kommandos	42
2.2.4.2	Vorgehensweise zur Verwendung des Interbus	43
2.2.4.3	Interbus Konfigurationsparameter	43
2.2.4.4	Interbus Fehlerbehandlung	43
2.2.5	Interbus PCAP-Programmierung	44
2.2.6	Interbus Fehlermeldungen (Fehlercodes)	44
2.2.6.1	Allgemeines	44
2.2.6.2	Logische Fehler (F0XX _{hex})	44
2.2.6.3	Port-Fehler (F1XX _{hex})	45
2.2.6.4	Logische Fehler in der seriellen Kommunikation (F2XX _{hex})	47
2.2.6.5	INTERBUS Fehler (F3XX _{hex} , F4XX _{hex})	48
2.2.6.6	Allgemeiner Fehler	50
2.3	Interbus Hardware	51
2.3.1	Stecker P1 (Interbus): 9-poliger SUB-D-Steckverbinder (Buchse)	51
2.3.2	Technische Daten der Option OPIBS	51

3 SUB-D-Stecker, für Freigabe-Relais, CNC-Bereit-Relais.....52

3.1	SUB-D-Adapter von 10-poligem FB-Stecker X5 auf 9-poligen SUB-D-Stecker	52
3.2	SUB-D-Adapter von 10-poligem FB-Stecker X5 bzw. X2 auf 25-poligen SUB-D-Stecker männlich	52

4 SUB-D-Stecker, für CAN-Bus.....54

4.1	SUB-D-Adapter von 10-poligem FB-Stecker P3 auf 9-poligen SUB-D-Stecker	54
-----	--	----

5 Anhang	55
5.1 Abbildungsverzeichnis	55
5.2 Tabellenverzeichnis	55

1 Optionsprint OPMF-3001 / OPMF-3100

1.1 Kurzbeschreibung

Der Optionsprint OPMF-3001 dient zur Systemerweiterung der Positionier- und Bahnsteuerung MCU-3000 / APCI-8001. Im wesentlichen beinhaltet er die Zusatzlogik zur Ansteuerung von bis zu fünf weiteren Servo- oder Schrittmotorachsen. Die Geräteelektronik ist dabei voll kompatibel zu denen der MCU-3000-Grundplatine. Darüber hinaus kann der OPMF-3001 zusätzlich bis zu 8 analoge Eingangssignale mit 12 bit Auflösung verarbeiten. Dieses Kapitel beschreibt die software- und hardwaretechnischen Merkmale des OPMF-3001.

Der Optionsprint OPMF-3100 dient zur Systemerweiterung der Positionier- und Bahnsteuerung MCU-3100 / APCI-8008. Im wesentlichen beinhaltet er die Zusatzlogik zur Ansteuerung von bis zu fünf weiteren Servo- oder Schrittmotorachsen. Die Geräteelektronik ist dabei voll kompatibel zu denen der MCU-3100-Grundplatine. Darüber hinaus kann der OPMF-3100 zusätzlich bis zu 8 analoge Eingangssignale mit 16 bit Auflösung verarbeiten. Unterschiede dieser Baugruppe zum OPMF-3001 sind in der nachfolgenden Beschreibung hervorgehoben.

1.2 Software

Der OPMF-3001 / OPMF-3100 kann mit der Standard-TOOLSET-Software für die MCU-3000 / MCU-3100 projektiert und programmiert werden (ab Software-Revision V2.5.0). Die entsprechenden Zusatzfunktionen werden aufbauend auf dem Programmierhandbuch [PHB] der MCU-3000 in den nachfolgenden Kapiteln beschrieben.

1.2.1 Modifizierte Betriebssystemsoftware *rwmos.elf*

Damit die Zusatzfunktionen des OPMF-3001 / OPMF-3100 genutzt werden können, ist in manchen Fällen eine spezielle Betriebssystemsoftware *rwmos.elf* notwendig. Bei der Aufrüstung zum Vier (Fünf, Sechs, Sieben oder Acht) -Achsen-System sind die Informationen der nachfolgenden Kapitel relevant. In diesem Fall werden die Zusatzfunktionen der Achskanäle 4 .. 8 genauso genutzt wie beim Standard-Drei-Achsen-System.

Sofern jedoch nur einzelne Funktionsbaugruppen auf dem OPMF-3001 / OPMF-3100 vorhanden sind (Teilbestückung), wird der Zugriff auf diese Funktionsbaugruppen mit Hilfe der SAP-Programmiermethode ausgeführt. Hierzu sind eine Reihe neuer Systemparameter in *rw_SymPas* definiert. Die im Lieferumfang enthaltenen Beispielprogramme veranschaulichen den einfachen Zugriff dieser Zugriffsmethode.

1.2.2 PCAP-Programmierung

In den nachfolgenden Kapiteln wird auf die einzelnen Signale und deren Anschlusspins verwiesen. Die entsprechenden Steckerbelegungen finden Sie im Kapitel 1.3.1

1.2.2.1 rdAin, read analog inputs

BESCHREIBUNG:	Diese Funktion liefert den aktuellen Analogwert des Kanals <i>channel</i> der Achse <i>an</i> zurück. Jeder Achse sind maximal 4 Analogeingangskanäle zugeordnet. Da bis zu acht Kanäle verarbeitet werden können, sind demnach die Analogeingangskanäle 1. . 4 Achskanal 1 und die Analogeingangskanäle 5 .. 8 dem Achskanal 2 zugeordnet.
BORLAND DELPHI:	function rdAin(an: integer; channel: integer): integer;
C:	int rdAin(int an, int channel);
VISUAL BASIC:	Function rdAin(ByVal an As Long, ByVal channel As Long) As Long
RÜCKGABEWERT:	Analogwert als Ganzzahl.
ANMERKUNG:	<p>Bis zu 8 acht Analogwerte können verarbeitet werden. Bei den unterschiedlichen Steuerungssystemen ergeben sich geringfügige Unterschiede bei der Handhabung der Analogeingänge, die vom Anwender ggf. berücksichtigt werden müssen, insbesondere bei einem Systemumstieg.</p> <p>MCU-3000 / APCI-8001: Pro Abtastzyklus wird ein Wert (ein Kanal) aktualisiert, d.h. bei acht Kanälen wird jeder Kanal alle 8ms bei einem voreingestellten Taktzyklus von 1ms aufgefrischt. Jeder Kanal kann individuell auf folgende Messbereiche eingestellt werden: 0 .. +5V, 0 .. 10V, -5V .. +5V und -10V .. 10V. Gültige Indizes für <i>an</i> sind 0 und 1, für <i>channel</i> dagegen sind die Werte 0..3 gültig. Wenn nur 4 Analoge Eingangskanäle aufgrund der Bestückungsversion zur Verfügung stehen, sind dies die ersten 4 Kanäle. Die Auflösung der Analogkanäle beträgt 12 bit.</p> <p>MCU-3100 / APCI-8008: In jedem Abtastzyklus werden alle Werte aktualisiert. Alle Kanäle können nur gemeinsam auf folgende Messbereiche eingestellt werden: -5V .. +5V und -10V .. 10V. Gültige Indizes für <i>an</i> sind 0 und 1, für <i>channel</i> dagegen sind die Werte 0..3 gültig. Wenn nur 4 Analoge Eingangskanäle aufgrund der Bestückungsversion zur Verfügung stehen, sind dies die ersten 2 Kanäle bei der ersten Achse und die ersten 2 Kanäle bei der zweiten Achse. Die Auflösung der Analogkanäle beträgt 16 bit.</p>

1.2.2.2 rddigi, read digital inputs

Beschreibung:	<p>Mit dieser Funktion können folgende Signalzustände abgefragt werden:</p> <ul style="list-style-type: none"> • Der aktuelle Zustand der 24 OPMF-xxxx Digital-Eingänge • Der aktuelle Zustand der Nullspur- (Index) Signals vom Inkrementalkoder • Ein zwischengespeicherter Fehler des Meßwerterfassungssystems • Eine zwischengespeicherte Flanke des Nullspur- (Index) Signals vom Inkrementalgeber • Eine zwischengespeicherte Flanke des Hardware-Latchsignals (Strobe) <p>Sofern ein Eingang aktiv ist, wird dies mit dem Wert 1 des jeweiligen Bit angezeigt. Optional können alle Digitaleingänge im TOOLSET Programm <i>mcf.exe</i> mit Invertierung projektiert werden. Ebenso ist es möglich, bei Verwendung eines Inkrementalkoders mit Index-Signal die gewünschte Polarität zu projektieren.</p> <p>Zu beachten ist, dass die Digitaleingänge auf dem OPMF achsspezifisch gruppiert sind.</p>
BORLAND DELPHI:	procedure rddigi(var tsrp:TSRP);
C:	void rddigi(struct TSRP far *tsrp);
VISUAL BASIC:	Sub rddigi(DTSRP As TSRP)
TSRP-KOMPONENTEN:	TSRP[n].digi n = 3 .. Anzahl der Achsen-1
RÜCKGABEWERT:	Der bitkodierte Rückgabewert befindet sich in der Struktur- bzw. Recordkomponente <i>digi</i> und hat den in nachfolgend abgedruckter Tabelle beschriebenen Aufbau.
ANMERKUNG:	Siehe Anmerkung PHB rddigi() Die Digitaleingänge 17 .. 32 (Stecker X1) sind den Achskanälen 4, 5 und 6 zugeordnet. Die Digitaleingänge 33 .. 40 (Stecker X2) sind den Achskanälen 7 und 8 zugeordnet.

1.2.2.2.1 Achsenqualifizierer digi für die Achskanäle 4, 5 und 6

Mit dem Register *digi* kann der Zustand der Digital-Eingänge der MCU-3000 abgeprüft werden. Sofern die jeweiligen Eingänge aktiv sind, wird dies mit dem Wert 1 an der jeweiligen Bitposition angezeigt.

Tabelle 1-1: Bitkodierter Aufbau des digi-Wortes für die [Achskanäle 4, 5 und 6](#)

Bit-Nr.	Funktion	X1/Pin
0	Eingang 17	9
1	Eingang 18	10
2	Eingang 19	11
3	Eingang 20	12
4	Eingang 21	13
5	Eingang 22	14
6	Eingang 23	15
7	Eingang 24	16
8	Eingang 25	42
9	Eingang 26	43
10	Eingang 27	44
11	Eingang 28	45
12	Eingang 29	46
13	Eingang 30 u. Hardware-Strobe-Signal zum Latchen der Istposition Achskanal 4	47
14	Eingang 31 u. Hardware-Strobe-Signal zum Latchen der Istposition Achskanal 5	48
15	Eingang 32 u. Hardware-Strobe-Signal zum Latchen der Istposition Achskanal 6	49
16	Null-Spur vom Inkrementalenkoder, achsspezifisch	--
17	Fehler des Messwerverfassungssystems, achsspezifisch	--
18	Zwischengespeicherter Wert des Nullspursignals vom Inkrementalgeber, achsspezifisch	--
19	Zwischengespeicherter Wert des Latchsignals (Hardware-Strobe), achsspezifisch	--
20..31	Nicht belegt, diese Flags haben einen undefinierten Wert	--

1.2.2.2.2 Achsenqualifizierer digi für die Achskanäle 7 und 8

Mit dem Register *digi* kann der Zustand der Digital-Eingänge der MCU-3000 abgeprüft werden. Sofern die jeweiligen Eingänge aktiv sind, wird dies mit dem Wert 1 an der jeweiligen Bitposition angezeigt.

Tabelle 1-2: Bitkodierter Aufbau des digi-Wortes für die [Achskanäle 7 und 8](#)

Bit-Nr.	Funktion	X2 / Pin (SUB-D)
0	Eingang 33	9
1	Eingang 34	10
2	Eingang 35	11
3	Eingang 36	12
4	Eingang 37	13
5	Eingang 38	14
6	Eingang 39 u. Hardware-Strobe-Signal zum Latchen der Istposition Achskanal 7	15
7	Eingang 40 u. Hardware-Strobe-Signal zum Latchen der Istposition Achskanal 8	16
8..15	nicht belegt	
16	Null-Spur vom Inkrementalkoder, achsspezifisch	--
17	Fehler des Messwerterfassungssystems, achsspezifisch	--
18	Zwischengespeicherter Wert des Nullspursignals vom Inkrementalgeber, achsspezifisch	--
19	Zwischengespeicherter Wert des Latchsignals (Hardware-Strobe), achsspezifisch	--
20..31	Nicht belegt, diese Flags haben einen undefinierten Wert	--

1.2.2.3 rddigib, read digital input bit

BESCHREIBUNG:	Mit dieser Funktion kann der aktuelle Zustand <u>eines</u> OPMF-xxxx Digital-Eingangs und diverser anderer Logiksignale abgefragt werden. Die Achsnummer muss im Parameter <i>an</i> (0, 1, ... <i>MAXAXIS-1</i>) spezifiziert werden. <u>Achtung:</u> die Zählweise von <i>bitnr</i> beginnt bei 1!
BORLAND DELPHI:	function rddigib(an:integer; bitnr:integer):boolean;
C:	int rddigib(int an, int bitnr);
VISUAL BASIC:	Function rddigib(ByVal an As Long, ByVal bitnr As Long) As Long
RÜCKGABEWERT:	Die Funktion liefert den Wert 1 bzw. TRUE zurück, sofern der entsprechende Eingang von <i>bitnr</i> aktiv ist.
ANMERKUNG:	Siehe rddigi() Kapitel [1.2.2.2] und Anmerkung PHB rddigib() Kapitel [1.2.2.3]

1.2.2.4 rddigo, read digital outputs

Siehe wrdigo im Kapitel [1.2.2.6]

1.2.2.5 rddigob, read digital output bit

Siehe wrdigob im Kapitel [1.2.2.7]

1.2.2.6 wrdigo, write digital outputs

BESCHREIBUNG:	<p>Mit diesem Register können die Digital-Ausgänge des OPMF-xxxx gesetzt werden. Zu beachten ist, dass die Digitalausgänge auf dem OPMF achsspezifisch gruppiert sind. Sofern ein Ausgang gesetzt werden soll, wird dies durch Setzen des jeweiligen Bits erreicht. Der bitkodierte Aufbau des <i>digo</i>-Statuswortes kann folgender Tabelle entnommen werden:</p> <p style="text-align: center;">Tabelle: Bitkodierter Aufbau des digo-Wortes für die Achskanäle 4, 5 und 6</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit-Nr.</th> <th>Funktion</th> <th>Stecker X1 / PIN</th> </tr> </thead> <tbody> <tr><td>0</td><td>Ausgang 9</td><td>26</td></tr> <tr><td>1</td><td>Ausgang 10</td><td>27</td></tr> <tr><td>2</td><td>Ausgang 11</td><td>28</td></tr> <tr><td>3</td><td>Ausgang 12</td><td>29</td></tr> <tr><td>4</td><td>Ausgang 13</td><td>30</td></tr> <tr><td>5</td><td>Ausgang 14</td><td>31</td></tr> <tr><td>6</td><td>Ausgang 15</td><td>32</td></tr> <tr><td>7</td><td>Ausgang 16</td><td>33</td></tr> <tr><td>8..31</td><td>Nicht belegt.</td><td>--</td></tr> </tbody> </table> <p style="text-align: center;">Tabelle: Bitkodierter Aufbau des digo-Wortes für die Achskanäle 7 und 8</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit-Nr.</th> <th>Funktion</th> <th>Stecker X2 / PIN (SUB-D)</th> </tr> </thead> <tbody> <tr><td>0</td><td>Ausgang 17</td><td>26</td></tr> <tr><td>1</td><td>Ausgang 18</td><td>27</td></tr> <tr><td>2</td><td>Ausgang 19</td><td>28</td></tr> <tr><td>3</td><td>Ausgang 20</td><td>29</td></tr> <tr><td>4..31</td><td>Nicht belegt.</td><td>--</td></tr> </tbody> </table>	Bit-Nr.	Funktion	Stecker X1 / PIN	0	Ausgang 9	26	1	Ausgang 10	27	2	Ausgang 11	28	3	Ausgang 12	29	4	Ausgang 13	30	5	Ausgang 14	31	6	Ausgang 15	32	7	Ausgang 16	33	8..31	Nicht belegt.	--	Bit-Nr.	Funktion	Stecker X2 / PIN (SUB-D)	0	Ausgang 17	26	1	Ausgang 18	27	2	Ausgang 19	28	3	Ausgang 20	29	4..31	Nicht belegt.	--
Bit-Nr.	Funktion	Stecker X1 / PIN																																															
0	Ausgang 9	26																																															
1	Ausgang 10	27																																															
2	Ausgang 11	28																																															
3	Ausgang 12	29																																															
4	Ausgang 13	30																																															
5	Ausgang 14	31																																															
6	Ausgang 15	32																																															
7	Ausgang 16	33																																															
8..31	Nicht belegt.	--																																															
Bit-Nr.	Funktion	Stecker X2 / PIN (SUB-D)																																															
0	Ausgang 17	26																																															
1	Ausgang 18	27																																															
2	Ausgang 19	28																																															
3	Ausgang 20	29																																															
4..31	Nicht belegt.	--																																															
BORLAND DELPHI:	procedure wrdigo(var tsrp:TSRP);																																																
C:	void wrdigo(struct TSRP far *tsrp);																																																
VISUAL BASIC:	Sub wrdigo(DTSRP As TSRP)																																																
TSRP-KOMPONENTEN:	TSRP[n].digo																																																

1.2.2.7 wrdigob, write digital output bit

BESCHREIBUNG:	<p>Mit dieser Funktion kann ein OPMF-xxxx Digital-Ausgang gesetzt bzw. rückgesetzt werden. Die Achsnummer muss im Parameter <i>an</i> (3 ... <i>MAXAXIS-1</i>) spezifiziert werden. Das Rücksetzen des Ausgangs erfolgt mit dem Wert 0 bzw. FALSE. <u>Achtung:</u> die Zählweise von <i>bitnr</i> beginnt bei 1!</p>
BORLAND DELPHI:	procedure wrdigob(an:integer; bitnr:integer; value: boolean);
C:	wrdigob(int an, int bitnr, int value);
VISUAL BASIC:	Sub wrdigob(ByVal an As Long, ByVal bitnr As Long, ByVal value As Long)
ANMERKUNG	PCAP-Befehl <i>wrdigo()</i>

1.2.3 SAP-Programmierung

1.2.3.1 Achsenqualifizierer digi und digo

Die Funktionsweise der Achsenqualifizierer *digi* und *digo* kann bei den PCAP-Befehlen `rddigi()` Kapitel [1.2.2.2] und `rddigo()` Kapitel [1.2.2.4] nachzulesen werden.

1.2.3.2 Achsenqualifizierer ain0 .. ain3

Die Achsenqualifizierer *ain0* .. *ain3* dienen zum Einlesen der Analogen Eingangskanäle. Die Achsenqualifizierer sind den Achskanälen 1 und 2 zugeordnet. Somit können bis zu 8 Analog-Eingänge verarbeitet werden. Die Eingangsinformation wird als *integer*-Typ zurückgeliefert.

OPMF-3001: Der Wertebereich ist aufgrund der AD-Wandler-Wortbreite 12 Bit inklusive Vorzeichen auf Werte von -2048.. 2047 festgelegt. Welchem Spannungswert dieser Messwert entspricht hängt vom Eingangsspannungsbereich des jeweiligen A/D-Kanals ab. Im Normalfall ist dies -5 .. 5V. Es können jedoch auch Messbereiche von -10 .. +10V, 0 .. 5V oder 0 .. 10V verarbeitet werden. Jeder Eingangskanal kann individuell auf einen der zuvor genannten Messbereiche eingestellt werden. Wenn nur 4 Eingangskanäle bestückt sind, werden diese unter den 4 Kanälen der ersten Achse angesprochen. Hierzu werden die Eingänge AIN1 bis AIN4 verdrahtet.

OPMF-3100: Der Wertebereich ist aufgrund der AD-Wandler-Wortbreite 16 Bit inklusive Vorzeichen auf Werte von -32768..+32767 festgelegt. Welchem Spannungswert dieser Messwert entspricht hängt vom Eingangsspannungsbereich des jeweiligen A/D-Kanals ab. Im Normalfall ist dies -10 .. +10V. Es können jedoch auch Messbereiche von -5 .. +5V verarbeitet werden. Es können nur alle Eingangskanäle gemeinsam auf einen der zuvor genannten Messbereiche eingestellt werden. Wenn nur 4 Eingangskanäle bestückt sind, werden diese jeweils unter den ersten zwei Kanälen der ersten beiden Achsen angesprochen. Hierzu werden die Eingänge AIN1, AIN2, AIN5 und AIN6 verdrahtet.

1.3 Hardware

Der Optionsprint OPMF-xxxx ist als Tochterboard für die MCU-3X00-Grundplatine realisiert. Die beiden Baugruppen werden mit Steck- und Schraubverbindungen werkseitig assembliert. Die Steuerung benötigt aus mechanischen Gründen je nach Ausbaustufe einen oder zwei PC-Steckplätze, elektrisch jedoch wird nur ein PCI-Steckplatz benötigt.

Die Peripherie-Elektronik wird je nach Ausbaustufe mit Hilfe eines 50poligen SUB-D-Steckverbinders (X1) und eines 60poligen SUB-D-Steckverbinder (X2) an dem OPMF-xxxx angeschlossen. Für den 60poligen FB-Steckverbinder X2 des OPMF-xxxx ist auch ein Kabelsatz mit Kartenhalter und integriertem 50pol. SUB-D-Steckverbinder erhältlich. Dieser Kartenhalter kann in einen freien PC-Slot neben der Steuerung eingebaut werden. In der maximalen Ausbaustufe (8 Achsen) kommen demnach bis zu drei 50polige SUB-D-Steckverbinder zum Einsatz.

Alle nachfolgenden Pinbeschreibungen beziehen sich auf diese 50poligen SUB-D Steckverbinder.

1.3.1 Optionsprint OPMF-3001 / OPMF-3100, Anschlussbelegungen

1.3.1.1 Stecker X1: 50-poliger SUB-D-Steckverbinder (Stift)

Tabelle 1-3: Steckerbelegung X1 (SUB-D-Steckverbinder)

Pin	Name	Gruppe
1	SERVO4 / PULSE4+	Sollwert 4 / Stepper 4
2	AGND4 / PULSE4-	Sollwert 4 / Stepper 4
3	CHA4+ / CLKSSI4+	Istwert 4
4	CHA4- / CLKSSI4-	Istwert 4
5	CHB4+ / DATSSI4+	Istwert 4
6	CHB4- / DATSSI4-	Istwert 4
7	NDX4+ / SIGN4+	Istwert 4 / Stepper 4
8	NDX4- / SIGN4-	Istwert 4 / Stepper 4
9	I17	Digital-Eingänge 17-24 (24V)
10	I18	Zuordnung zu Achskanal 4, 5 und 6
11	I19	
12	I20	
13	I21	
14	I22	
15	I23	
16	I24	
17	+24V	Spannungsversorgung für die digitalen <u>Ausgänge</u> 24V, sofern Digitalausgänge benutzt werden, muss diese Spannung von extern zugeführt werden. Sollte mit Grundplatine MCU-3000 Stecker X1, Pin 17 verbunden werden!
18	SERVO5 / PULSE5+	Sollwert 5 / Stepper 5
19	AGND5 / PULSE5-	Sollwert 5 / Stepper 5
20	CHA5+ / CLKSSI5+	Istwert 5
21	CHA5- / CLKSSI5-	Istwert 5
22	CHB5+ / DATSSI5+	Istwert 5
23	CHB5- / DATSSI5-	Istwert 5
24	NDX5+ / SIGN5+	Istwert 5 / Stepper 5
25	NDX5- / SIGN5-	Istwert 5 / Stepper 5
26	O9	Digitalausgänge 9..16 (24V)
27	O10	Zuordnung zu Achskanal 4, 5 und 6
28	O11	
29	O12	
30	O13	
31	O14	
32	O15	
33	O16	
34	SERVO6 / PULSE6+	Sollwert 6 / Stepper 6
35	AGND6 / PULSE6-	Sollwert 6 / Stepper 6
36	CHA6+ / CLKSSI6+	Istwert 6
37	CHA6- / CLKSSI6-	Istwert 6
38	CHB6+ / DATSSI6+	Istwert 6
39	CHB6- / DATSSI6-	Istwert 6
40	NDX6+ / SIGN6+	Istwert 6 / Stepper 6
41	NDX6- / SIGN6-	Istwert 6 / Stepper 6

Tabelle: Steckerbelegung X1 Fortsetzung

Pin	Name	Gruppe
42	I25	Digital-Eingänge 25-32 (24V)
43	I26	Zuordnung zu Achskanal 4, 5 und 6
44	I27	
45	I28	
46	I29	
47	I30	Schneller Latcheingang Achskanal 4
48	I31	Schneller Latcheingang Achskanal 5
49	I32	Schneller Latcheingang Achskanal 6
50	GND-D	Bezugspotential für alle Signalquellen. Dazu gehören u.a. digitale Ein- und Ausgänge und die Geberistwerte. GND-D muss mit dem Massepotential der externen Geräteelektronik verbunden werden. Sollte mit Grundplatine MCU-3000 Stecker X1, Pin 50 verbunden werden!

1.3.1.2 Stecker X2: FB-Stecker X2 und Transformation auf SUB-D-Steckverbinder (Stift / Buchse)

Tabelle 1-4: Steckerbelegung X2 (SUB-D-Steckverbinder)

Pin SUB-D	Name	Beschreibung / Anmerkungen	Pin FB-60 - X2
1		Relais 1, Ruhekontakt	1
2		Relais 2, Ruhekontakt	3
3		Relais 3, Ruhekontakt	5
4		Relais 4, Ruhekontakt	7
5		Relais 5, Ruhekontakt	9
6		Relais 1, Schließer, Freigabe Achskanal 4	2
7		Relais 2, Schließer, Freigabe Achskanal 5	4
8		Relais 3, Schließer, Freigabe Achskanal 6	6
9		Relais 4, Schließer, Freigabe Achskanal 7	8
		Relais 5, Schließer, Freigabe Achskanal 8	10
1	Servo7 / Pulse7+	Achskanal 7	11
2	AGND7 / Pulse7-		14
3	CHA7+ / CLKSSI7+		17
4	CHA7- / CLKSSI7-		20
5	CHB7+ /DATSSI7+		23
6	CHB7- / DATSSI7-		26
7	NDX7+ / SIGN7+		29
8	NDX7- / SIGN7-		32
9	I33	Digital-Eingänge 33-40 (24V) Zuordnung zu Achskanal 7 und 8	35
10	I34		38
11	I35		41
12	I36		44
13	I37		47
14	I38		50
15	I39		53
16	I40		56
17	+24V	Spannungsversorgung für Digitalausgänge mit X1-17 intern verbunden	59
18	Servo8/ Pulse8+	Achskanal 8	13
19	AGND8 / Pulse8-		16
20	CHA8+ / CLKSSI8+		19
21	CHA8- / CLKSSI8-		22
22	CHB8+ /DATSSI8+		25
23	CHB8- / DATSSI8-		28
24	NDX8+ / SIGN8+		31
25	NDX8- / SIGN8-		34

Tabelle: Steckerbelegung X2 Fortsetzung

Pin SUB-D	Name	Beschreibung / Anmerkungen	Pin FB-60 - X2
26	O17	Digitalausgänge 17. 20 (24V) Zuordnung zu Achskanal 7 und 8	37
27	O18		40
28	O19		43
29	O20		46
30	RGND1	Referenzausgang Gruppe 1	49
31	AREF1		52
32	RGND2	Referenzausgang Gruppe 2	55
33	AREF2		58
34	AIN1+	Analog-Eingänge 1 bis 4 Zuordnung zu Achskanal 1!	12
35	AIN1-		15
36	AIN2+		18
37	AIN2-		21
38	AIN3+		24
39	AIN3-		27
40	AIN4+		30
41	AIN4-		33
42	AIN5+	Analog-Eingänge 5 bis 8 Zuordnung zu Achskanal 2!	36
43	AIN5-		39
44	AIN6+		42
45	AIN6-		45
46	AIN7+		48
47	AIN7-		51
48	AIN8+		54
49	AIN8-		57
50	GND-D	Spannungsversorgung für Digitalausgänge mit X1-50 intern verbunden	60

Die mit der **Farbe gelb** gekennzeichneten Farbfelder kennzeichnen die PIN - Belegung am 50-poligen SUB-D-Steckverbinder. Hierzu wird ein FB-3001 bzw. FB-8001 Verbindungskabel benötigt.

Die mit der **Farbe grün** gekennzeichneten Pinbelegung kennzeichnen die PIN – Belegung am 9-poligen SUB-D-Steckverbinder (siehe hierzu auch Kapitel [1.3.1.8]). Hierzu wird ein FB-3001 bzw. FB-8002 Verbindungskabel benötigt.

Belegung bei OPMF-3001 und OPMF-3100 ist gleich. Bei den Analog-Eingängen ist die unterschiedliche Auflösung zu beachten.

1.3.1.3 Sollwertkanäle

Die OPMF-xxxx-Systemelektronik unterstützt wahlweise den Betrieb von bis zu fünf weiteren Schritt- bzw. Servomotor-Achsen. Die Projektierung und Auswahl des gewünschten Motorsystems wird mit Hilfe des TOOLSET Programms *mcfg.exe* durchgeführt.

1.3.1.3.1 Sollwertkanal für Servomotorachsen

Das Analogausgangssignal dient zur Ansteuerung eines Leistungsverstärkers, welcher als Drehzahlregler bzw. Momentenregler (Stromverstärker) geschaltet ist. Der Offset dieses Sollwertkanals wird werkseitig im nichtflüchtigen Flash-Speicher der MCU-3000 abgelegt und bei der Ausgabe softwaremäßig berücksichtigt. Die Analogsollwertausgabe wird nur bei *SERVO*-projektierten Achsen unterstützt.

1.3.1.3.1.1 Pinbelegung Stecker X1, Achskanal 4

Pin	Name	Gruppe	Beschreibung
1	SERVO4	Sollwert 4	Analogausgangssignal 4 zur Ansteuerung eines Leistungsverstärkers (+/-10V, 5mA). Dieses Signal ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt und hat das Bezugspotential AGND4.
2	AGND4	Sollwert 4	Bezugspotential für SERVO4. Dieses Potential ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt.

Achtung: Die Stiftleisten J1 und J2 müssen in Stellung 2-3 gebrückt werden, damit die in der Tabelle aufgeführten Signale am Stecker X1 verfügbar sind!

1.3.1.3.1.2 Pinbelegung Stecker X1, Achskanal 5

Pin	Name	Gruppe	Beschreibung
18	SERVO5	Sollwert 5	Analogausgangssignal 5 zur Ansteuerung eines Leistungsverstärkers (+/-10V, 5mA). Dieses Signal ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt und hat das Bezugspotential AGND5.
19	AGND5	Sollwert 5	Bezugspotential für SERVO5. Dieses Potential ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt.

Achtung: Die Stiftleisten J3 und J4 müssen in Stellung 2-3 gebrückt werden, damit die in der Tabelle aufgeführten Signale am Stecker X1 verfügbar sind!

1.3.1.3.1.3 Pinbelegung Stecker X1, Achskanal 6

Pin	Name	Gruppe	Beschreibung
34	SERVO6	Sollwert 6	Analogausgangssignal 6 zur Ansteuerung eines Leistungsverstärkers (+/-10V, 5mA). Dieses Signal ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt und hat das Bezugspotential AGND6.
35	AGND6	Sollwert 6	Bezugspotential für SERVO6. Dieses Potential ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt.

Achtung: Die Stiftleisten J5 und J6 müssen in Stellung 2-3 gebrückt werden, damit die in der Tabelle aufgeführten Signale am Stecker X1 verfügbar sind!

1.3.1.3.1.4 Pinbelegung Stecker X2, Achskanal 7

Pin (SUB-D)	Name	Gruppe	Beschreibung
1	SERVO7	Sollwert 7	Analogausgangssignal 7 zur Ansteuerung eines Leistungsverstärkers (+/-10V, 5mA). Dieses Signal ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt und hat das Bezugspotential AGND7.
2	AGND7	Sollwert 7	Bezugspotential für SERVO7. Dieses Potential ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt.

Achtung: Die Stiftleisten J7 und J8 müssen in Stellung 2-3 gebrückt werden, damit die in der Tabelle aufgeführten Signale am Stecker X2 verfügbar sind!

1.3.1.3.1.5 Pinbelegung Stecker X2, Achskanal 8

Pin (SUB-D)	Name	Gruppe	Beschreibung
18	SERVO8	Sollwert 8	Analogausgangssignal 8 zur Ansteuerung eines Leistungsverstärkers (+/-10V, 5mA). Dieses Signal ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt und hat das Bezugspotential AGND8.
19	AGND8	Sollwert 8	Bezugspotential für SERVO8. Dieses Potential ist von der OPMF-xxxx-Systemelektronik galvanisch getrennt.

Achtung: Die Stiftleisten J9 und J10 müssen in Stellung 2-3 gebrückt werden, damit die in der Tabelle aufgeführten Signale am Stecker X2 verfügbar sind!

1.3.1.3.2 Sollwertkanal für Schrittmotorachsen

Zur Ansteuerung einer Schrittmotor-Leistungsbaugruppe stehen vier Ausgangssignale zur Verfügung. Dies sind ein Pulssignal, ein Richtungssignal und deren invertierte Signale nach EIA Standard RS422. Alle Ausgänge liefern einen typischen Ausgangsstrom von -60mA (max. -150mA). Die maximale Impulsfrequenz der Schrittsignale beträgt 10MHz.

Achtung: Maßgeblich für die korrekte Anzahl auszuführender Schritte ist die positive Flanke des Schrittsignals PULSx+ bzw. die negative Flanke des Schrittsignals PULSx-.

1.3.1.3.2.1 Pinbelegung Stecker X1, Achskanal 4

Pin	Name	Gruppe	Beschreibung
1	PULSE4+	Stepper 4	Pulssignal
2	PULSE4-	Stepper 4	Pulssignal invertiert
7	SIGN4+	Stepper 4	Richtungssignal
8	SIGN4-	Stepper 4	Richtungssignal invertiert

Achtung: Die Stiftleisten J1 und J2 müssen in Stellung 1-2 gebrückt werden, damit die oben aufgeführten Signale am Stecker X1 verfügbar sind!

1.3.1.3.2.2 Pinbelegung Stecker X1, Achskanal 5

Pin	Name	Gruppe	Beschreibung
18	PULSE5+	Stepper 5	Pulssignal
19	PULSE5-	Stepper 5	Pulssignal invertiert
24	SIGN5+	Stepper 5	Richtungssignal
25	SIGN5-	Stepper 5	Richtungssignal invertiert

Achtung: Die Stiftleisten J3 und J4 müssen in Stellung 1-2 gebrückt werden, damit die oben aufgeführten Signale am Stecker X1 verfügbar sind!

1.3.1.3.2.3 Pinbelegung Stecker X1, Achskanal 6

Pin	Name	Gruppe	Beschreibung
34	PULSE6+	Stepper 6	Pulssignal
35	PULSE6-	Stepper 6	Pulssignal invertiert
40	SIGN6+	Stepper 6	Richtungssignal
41	SIGN6-	Stepper 6	Richtungssignal invertiert

Achtung: Die Stiftleisten J5 und J6 müssen in Stellung 1-2 gebrückt werden, damit die oben aufgeführten Signale am Stecker X1 verfügbar sind!

1.3.1.3.2.4 Pinbelegung Stecker X2, Achskanal 7

Pin (SUB-D)	Name	Gruppe	Beschreibung
1	PULSE7+	Stepper 7	Pulssignal
2	PULSE7-	Stepper 7	Pulssignal invertiert
7	SIGN7+	Stepper 7	Richtungssignal
8	SIGN7-	Stepper 7	Richtungssignal invertiert

Achtung: Die Stiftleisten J7 und J8 müssen in Stellung 1-2 gebrückt werden, damit die oben aufgeführten Signale am Stecker X2 verfügbar sind!

1.3.1.3.2.5 Pinbelegung Stecker X2, Achskanal 8

Pin (SUB-D)	Name	Gruppe	Beschreibung
18	PULSE8+	Stepper 8	Pulssignal
19	PULSE8-	Stepper 8	Pulssignal invertiert
24	SIGN8+	Stepper 8	Richtungssignal
25	SIGN8-	Stepper 8	Richtungssignal invertiert

Achtung: Die Stiftleisten J9 und J19 müssen in Stellung 1-2 gebrückt werden, damit die oben aufgeführten Signale am Stecker X2 verfügbar sind!

1.3.1.4 Pinbelegung Stecker X1, Digitale Eingänge

Die Prinzipschaltbilder der nachfolgend aufgelisteten digitalen Eingänge I17..I32 sind im [IHB / Kapitel 5.2.8.1] abgedruckt.

Pin	Name	Funktion
9	I17	Digital-Eingang 17
10	I18	Digital-Eingang 18
11	I19	Digital-Eingang 19
12	I20	Digital-Eingang 20
13	I21	Digital-Eingang 21
14	I22	Digital-Eingang 22
15	I23	Digital-Eingang 23
16	I24	Digital-Eingang 24
42	I25	Digital-Eingang 25
43	I26	Digital-Eingang 26
44	I27	Digital-Eingang 27
45	I28	Digital-Eingang 28
46	I29	Digital-Eingang 29
47	I30	Digital-Eingang 30 und schneller Hardware-Latcheingang zum Speichern der Istposition Achskanal 4
48	I31	Digital-Eingang 31 und schneller Hardware-Latcheingang zum Speichern der Istposition Achskanal 5
49	I32	Digital-Eingang 32 und schneller Hardware-Latcheingang zum Speichern der Istposition Achskanal 6

1.3.1.5 Pinbelegung Stecker X2, Digitale Eingänge

Die Prinzipschaltbilder der nachfolgend aufgelisteten digitalen Eingänge I33..I40 sind im [IHB / Kapitel 5.2.8.1] abgedruckt.

Pin (SUB-D)	Name	Funktion
9	I33	Digital-Eingang 33
10	I34	Digital-Eingang 34
11	I35	Digital-Eingang 35
12	I36	Digital-Eingang 36
13	I37	Digital-Eingang 37
14	I38	Digital-Eingang 38
15	I39	Digital-Eingang 39 und schneller Hardware-Latcheingang zum Speichern der Istposition Achskanal 7
16	I40	Digital-Eingang 40 und schneller Hardware-Latcheingang zum Speichern der Istposition Achskanal 8

1.3.1.6 Pinbelegung Stecker X1, Digitale Ausgänge

Die Prinzipschaltbilder der nachfolgend aufgelisteten digitalen Ausgänge O9 .. O16 sind im [IHB / Kapitel 5.2.9.1] abgedruckt.

Pin	Name	Funktion
26	O9	Digital-Ausgang 9
27	O10	Digital-Ausgang 10
28	O11	Digital-Ausgang 11
29	O12	Digital-Ausgang 12
30	O13	Digital-Ausgang 13
31	O14	Digital-Ausgang 14
32	O15	Digital-Ausgang 15
33	O16	Digital-Ausgang 16

1.3.1.7 Pinbelegung Stecker X2, Digitale Ausgänge

Die Prinzipschaltbilder der nachfolgend aufgelisteten digitalen Ausgänge O17 .. O20 sind im [IHB / Kapitel 5.2.9.1] abgedruckt.

Pin (SUB-D)	Name	Funktion
26	O17	Digital-Ausgang 17
27	O18	Digital-Ausgang 18
28	O19	Digital-Ausgang 19
29	O20	Digital-Ausgang 20

1.3.1.8 Pinbelegung Stecker X2, Freigaberelais

Am Steckverbinder X2 werden Relaiskontakte für die Verstärkerfreigaben zur Verfügung gestellt. Es handelt sich hierbei um Schließer. Das Relais ist nach Einschalten des PC, nach einem Rücksetzvorgang oder nach einem Fehler abgeschaltet.

Das Freigaberelais wird beim PCAP-Befehl *cl()* und beim SAP-Befehl *CL()* für den entsprechend selektierten Achskanal aktiviert.

Anmerkung: je nach Ausbaustufe des OPMF-xxxx sind 1 bis 5 Relaisausgänge verfügbar. Um alle Relaisausgänge nutzen zu können, werden diese Anschlüsse ggf. mit den Relaisausgängen der MCU-3000 Grundplatine auf einen 25-poligen Sub-D Stecker geführt (siehe hierzu Kapitel 3.2).
Bei den Relais handelt es sich um Halbleiterrelais mit einem Einschaltwiderstand von max. 25 Ohm. Das Schaltvermögen liegt bei 100mA, Schaltspannung max. 60V.

Pin (SUB-D)	Name	Funktion
1	R1-R	Relais S1, P-Kontakt, Freigabe für Leistungsverstärker Achskanal 4
2	R2-R	Relais S2, P-Kontakt, Freigabe für Leistungsverstärker Achskanal 5
3	R3-R	Relais S3, P-Kontakt, Freigabe für Leistungsverstärker Achskanal 6
4	R4-R	Relais S4, P-Kontakt, Freigabe für Leistungsverstärker Achskanal 7
5	R5-R	Relais S5, P-Kontakt, Freigabe für Leistungsverstärker Achskanal 8
6	R1-S	Relais S1, Schließer, Freigabe für Leistungsverstärker Achskanal 4
7	R2-S	Relais S2, Schließer, Freigabe für Leistungsverstärker Achskanal 5
8	R3-S	Relais S3, Schließer, Freigabe für Leistungsverstärker Achskanal 6
9	R4-S	Relais S4, Schließer, Freigabe für Leistungsverstärker Achskanal 7
	R5-S	Relais S5, Schließer, Freigabe für Leistungsverstärker Achskanal 8

1.3.1.9 Impulserfassungskanäle

Der OPMF-xxxx ist mit bis zu fünf Impulserfassungskanälen ausgestattet, an welchen unterschiedliche Enkodertypen wie beispielsweise Längenmaßstäbe oder Inkremental- oder Absolut-Drehgeber angeschlossen werden können. Als Eingangssignale werden zwei um 90° phasenverschobene Quadratursignale mit einer maximalen Impulsfrequenz von 2.0MHz (optional 5MHz) und TTL-Pegel verarbeitet. Eine Nullspur (Indexsignal) kann ebenfalls ausgewertet werden. Die von den Enkodern erfassten Signalpegel werden elektronisch vervierfacht und intern als Gleitpunktzahl mit doppelter Genauigkeit geführt. Somit ergibt sich ein praktisch uneingeschränkter Wertebereich für den Verfahrensweg.

1.3.1.9.1 Inkrementalenkoder mit invertierten Signalen (symmetrische Beschaltung)

Die Inkrementalenkoder mit symmetrischen Ausgängen sind besonders für den industriellen Einsatz geeignet und zu bevorzugen, da die Ausgangssignale mit invertiertem und nichtinvertiertem Signalpegel aller Spuren zur Verfügung stehen. Dies ermöglicht eine zuverlässige Impulserfassung auch in elektromagnetisch stark gestörter Umgebung. Die Auswerteelektronik auf dem OPMF-xxxx beruht auf dem RS422-Standard und bildet eine Signaldifferenz zwischen den invertiertem und nichtinvertiertem Eingangssignal. Störungen die in die Übertragungsleitungen eingekoppelt werden, können somit wirksam unterdrückt werden.

Wichtig: werksseitig wird der OPMF-xxxx für Inkrementalgeber mit symmetrischen Ausgängen ausgeliefert, kann jedoch durch den Anwender selbst für asymmetrische Enkoder (siehe nächste Tabelle) konfiguriert werden.

1.3.1.9.2 Inkrementalenkoder ohne invertierte Signale (asymmetrische Beschaltung)

Es ist auch möglich, Inkrementalenkoder ohne invertierte Impulsfolgen zu verarbeiten. Jedoch sollten diese nur in elektromagnetisch wenig gestörter Umgebung, z.B. in Laboranwendungen, eingesetzt werden. Ebenso ist zu beachten, dass die Leitungslänge des Enkoderkabels gerade bei hohen Impulsfrequenzen nur wenige Meter betragen darf.

Tabelle 1-5: Konfiguration der Inkrementalkoder für symmetrische und asymmetrische Betriebsart

Achskanal	Signalquelle	Lötjumper	asymmetrisch	symmetrisch
4	CHA4-	J15 (3001)	gebrückt	ungebrückt
		J15 (3100)	Pin 4 / X1 nicht beschalten!	Pin 4 / X1 beschalten!
	CHB4-	J22 (3001)	gebrückt	ungebrückt
		J16 (3100)	Pin 6 / X1 nicht beschalten!	Pin 6 / X1 beschalten!
	NDX4-	J23 (3001)	gebrückt	ungebrückt
		J17 (3100)	Pin 8 / X1 nicht beschalten!	Pin 8 / X1 beschalten!
5	CHA5-	J14 (3001)	gebrückt	ungebrückt
		J19 (3100)	Pin 21 / X1 nicht beschalten!	Pin 21 / X1 beschalten!
	CHB5-	J20 (3001)	gebrückt	ungebrückt
		J20 (3100)	Pin 23 / X1 nicht beschalten!	Pin 23 / X1 beschalten!
	NDX5-	J21 (3001)	gebrückt	ungebrückt
		J21 (3100)	Pin 25 / X1 nicht beschalten!	Pin 25 / X1 beschalten!
6	CHA6-	J13 (3001)	gebrückt	ungebrückt
		J23 (3100)	Pin 37 / X1 nicht beschalten!	Pin 37 / X1 beschalten!
	CHB6-	J18 (3001)	gebrückt	ungebrückt
		J24 (3100)	Pin 39 / X1 nicht beschalten!	Pin 39 / X1 beschalten!
	NDX6-	J19 (3001)	gebrückt	ungebrückt
		J25 (3100)	Pin 41 / X1 nicht beschalten!	Pin 41 / X1 beschalten!
7	CHA7-	J16 (3001)	gebrückt	ungebrückt
		J27 (3100)	Pin 4 / X2 nicht beschalten!	Pin 4 / X2 beschalten!
	CHB7-	J25 (3001)	gebrückt	ungebrückt
		J28 (3100)	Pin 6 / X2 nicht beschalten!	Pin 6 / X2 beschalten!
	NDX7-	J24 (3001)	gebrückt	ungebrückt
		J29 (3100)	Pin 8 / X2 nicht beschalten!	Pin 41 / X2 beschalten!
8	CHA8-	J11 (3001)	gebrückt	ungebrückt
		J31 (3100)	Pin 21 / X2 nicht beschalten!	Pin 21 / X2 beschalten!
	CHB8-	J12 (3001)	gebrückt	ungebrückt
		J32 (3100)	Pin 23 / X2 nicht beschalten!	Pin 23 / X2 beschalten!
	NDX8-	J17 (3001)	gebrückt	ungebrückt
		J33 (3100)	Pin 25 / X2 nicht beschalten!	Pin 25 / X2 beschalten!

Anmerkung: Die in der Tabelle aufgeführten Lötjumper befinden sich auf der Lötseite des OPMF-3001 / 3100 am oberen linken Kartenrand.

Die Bezeichner der Jumper sind in der Spalte Lötjumper gekennzeichnet für den OPMF-3001 mit (3001) und für den OPMF-3100 mit (3100).

1.3.1.9.3 Optische Entkopplung der Impulserfassungskanäle

Alle Impulserfassungskanäle des OPMF-xxxx sind optisch entkoppelt. Dies ist insbesondere in elektromagnetisch stark gestörter Umgebung von Vorteil.

1.3.1.9.4 Steckerbelegung für die Impulserfassungskanäle mit Inkrementalgebern

1.3.1.9.4.1 Steckerbelegung X1, Kanal 4

Pin	Name	Funktion
3	CHA4+	Inkrementalsignal (TTL-Rechteck-Impulsfolgen) Spur A
4	CHA4-	invertiertes Inkrementalsignal Spur A
5	CHB4+	Inkrementalsignal Spur B mit 90° elektrischem Phasenversatz zu Spur A
6	CHB4-	invertiertes Inkrementalsignal Spur B
7	NDX4+	Referenzsignal Spur 0
8	NDX4-	Invertiertes Referenzsignal Spur 0

1.3.1.9.4.2 Steckerbelegung X1, Kanal 5

Pin	Name	Funktion
20	CHA5+	Inkrementalsignal (TTL-Rechteck-Impulsfolgen) Spur A
21	CHA5-	invertiertes Inkrementalsignal Spur A
22	CHB5+	Inkrementalsignal Spur B mit 90° elektrischem Phasenversatz zu Spur A
23	CHB5-	invertiertes Inkrementalsignal Spur B
24	NDX5+	Referenzsignal Spur 0
25	NDX5-	Invertiertes Referenzsignal Spur 0

1.3.1.9.4.3 Steckerbelegung X1, Kanal 6

Pin	Name	Funktion
36	CHA6+	Inkrementalsignal (TTL-Rechteck-Impulsfolgen) Spur A
37	CHA6-	invertiertes Inkrementalsignal Spur A
38	CHB6+	Inkrementalsignal Spur B mit 90° elektrischem Phasenversatz zu Spur A
39	CHB6-	invertiertes Inkrementalsignal Spur B
40	NDX6+	Referenzsignal Spur 0
41	NDX6-	Invertiertes Referenzsignal Spur 0

1.3.1.9.4.4 Steckerbelegung X2, Kanal 7

Pin (SUB-D)	Name	Funktion
3	CHA7+	Inkrementalsignal (TTL-Rechteck-Impulsfolgen) Spur A
4	CHA7-	invertiertes Inkrementalsignal Spur A
5	CHB7+	Inkrementalsignal Spur B mit 90° elektrischem Phasenversatz zu Spur A
6	CHB7-	invertiertes Inkrementalsignal Spur B
7	NDX7+	Referenzsignal Spur 0
8	NDX7-	Invertiertes Referenzsignal Spur 0

1.3.1.9.4.5 Steckerbelegung X2, Kanal 8

Pin (SUB-D)	Name	Funktion
20	CHA8+	Inkrementalsignal (TTL-Rechteck-Impulsfolgen) Spur A
21	CHA8-	invertiertes Inkrementalsignal Spur A
22	CHB8+	Inkrementalsignal Spur B mit 90° elektrischem Phasenversatz zu Spur A
23	CHB8-	invertiertes Inkrementalsignal Spur B
24	NDX8+	Referenzsignal Spur 0
25	NDX8-	Invertiertes Referenzsignal Spur 0

1.3.1.10 Steckerbelegung X2 für die Analogeingangskanäle

Der **OPMF-3001** kann mit bis zu 8 Analogeingängen bezogen werden. Die Auflösung der Messwertkanäle beträgt 12bit inkl. Vorzeichen wobei der Messwertebereich 0 .. 5V, 0 .. 10V, -5 .. +5V oder -10 .. +10V für jeden Kanal individuell eingestellt werden kann. Die Analogeingänge werden auf dem OPMF-3001 über Differenzverstärker an die interne A/D-Baugruppe angeschlossen.

Der **OPMF-3100** kann mit bis zu 8 Analogeingängen bezogen werden. Die Auflösung der Messwertkanäle beträgt 16bit inkl. Vorzeichen wobei der Messwertebereich -5 .. +5V oder -10 .. +10V nur für alle Kanäle gemeinsam eingestellt werden kann. Die Analogeingänge werden auf dem OPMF-3100 direkt an die interne A/D-Baugruppe angeschlossen.

Vorsicht: Bei der Baugruppe OPMF-3100 Rev. A sind die Anschlüsse + und – bei allen Analogeingängen vertauscht. Bei den Folgeversionen Rev. B und neuer ist dieser Fehler behoben.

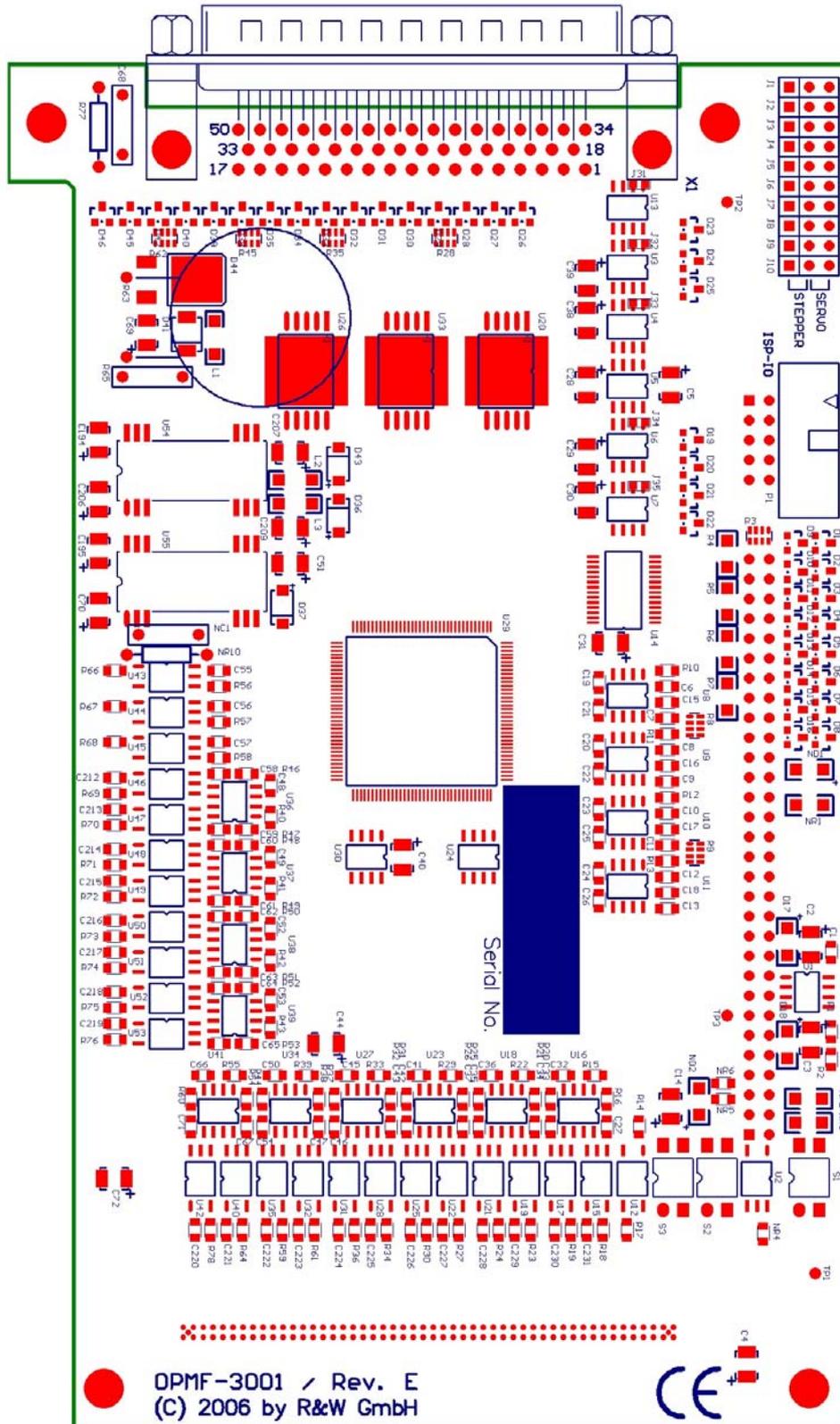
Pin (SUB-D)	Name	Zuordnung	Funktion
34	AIN1+	Achs-Kanal 1	Analog-Eingang 1
35	AIN1-		Bezugs-Potential für AIN1+
36	AIN2+		Analog-Eingang 2
37	AIN2-		Bezugs-Potential für AIN2+
38	AIN3+		Analog-Eingang 3
39	AIN3-		Bezugs-Potential für AIN3+
40	AIN4+		Analog-Eingang 4
41	AIN4-		Bezugs-Potential für AIN4+
42	AIN5+	Achs-Kanal 2	Analog-Eingang 5
43	AIN5-		Bezugs-Potential für AIN5+
44	AIN6+		Analog-Eingang 6
45	AIN6-		Bezugs-Potential für AIN6+
46	AIN7+		Analog-Eingang 7
47	AIN7-		Bezugs-Potential für AIN7+
48	AIN8+		Analog-Eingang 8
49	AIN8-		Bezugs-Potential für AIN8+

1.3.1.10.1 Steckerbelegung für die Referenzspannungsausgänge

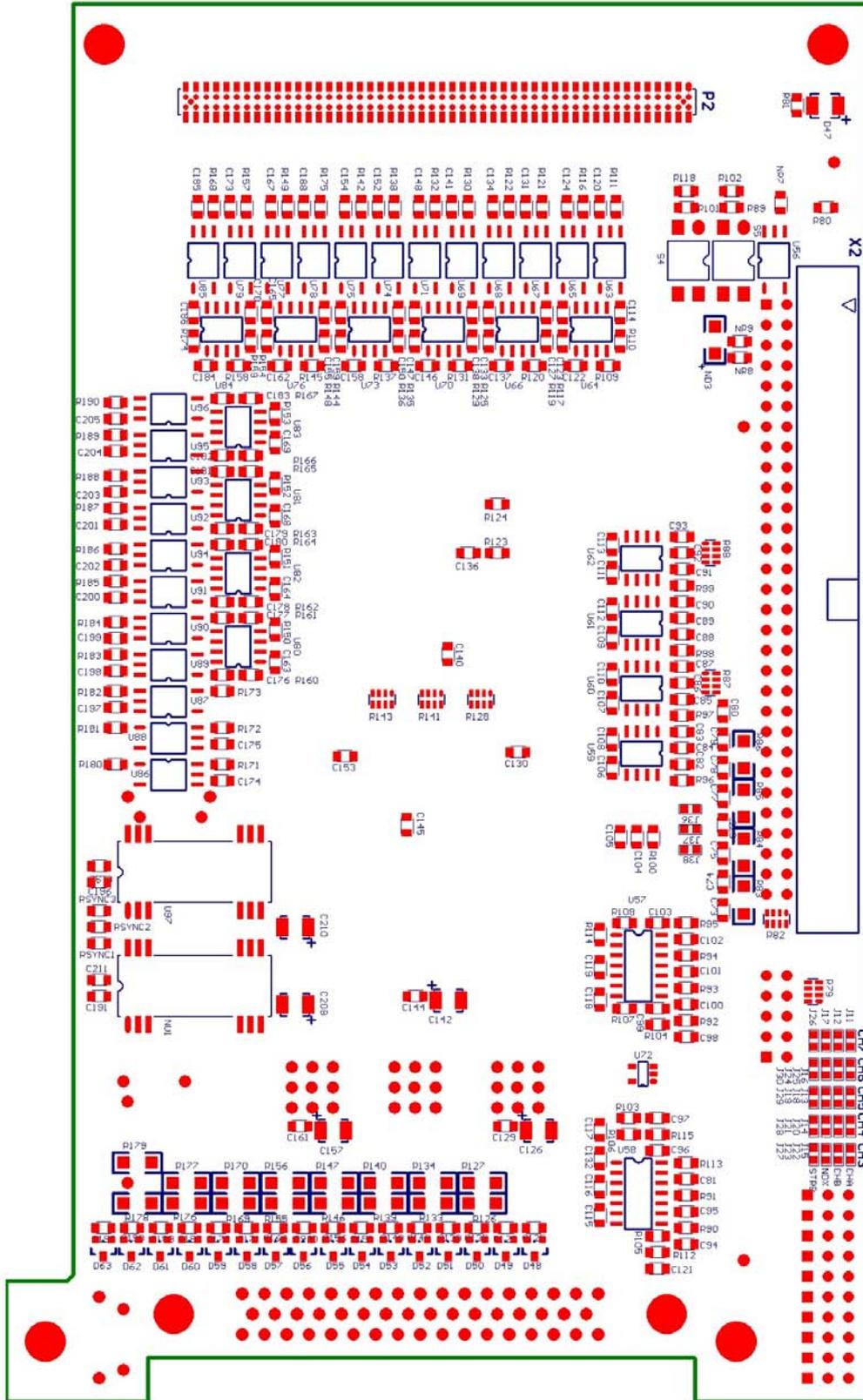
Der OPMF-3001 / OPMF-3100 kann mit bis zu 2 Referenzspannungsausgängen bezogen werden. Diese Ausgänge sind den oben genannten AD-Gruppen zugeordnet. Alle Ausgänge sind auf 10V eingestellt. Jeder Ausgang kann mit max. 5mA (bei weniger als 7 Achsen: 10mA) belastet werden.

Pin (SUB-D)	Name	Zuordnung	Funktion
30	RGND1	AIN1 ..	Bezugs-Potential für Referenz-Spannungsausgang 1
31	AREF1	AIN4	Referenz-Spannungsausgang 1
32	RGND2	AIN5 ..	Bezugs-Potential für Referenz-Spannungsausgang 2
33	AREF2	AIN8	Referenz-Spannungsausgang 2

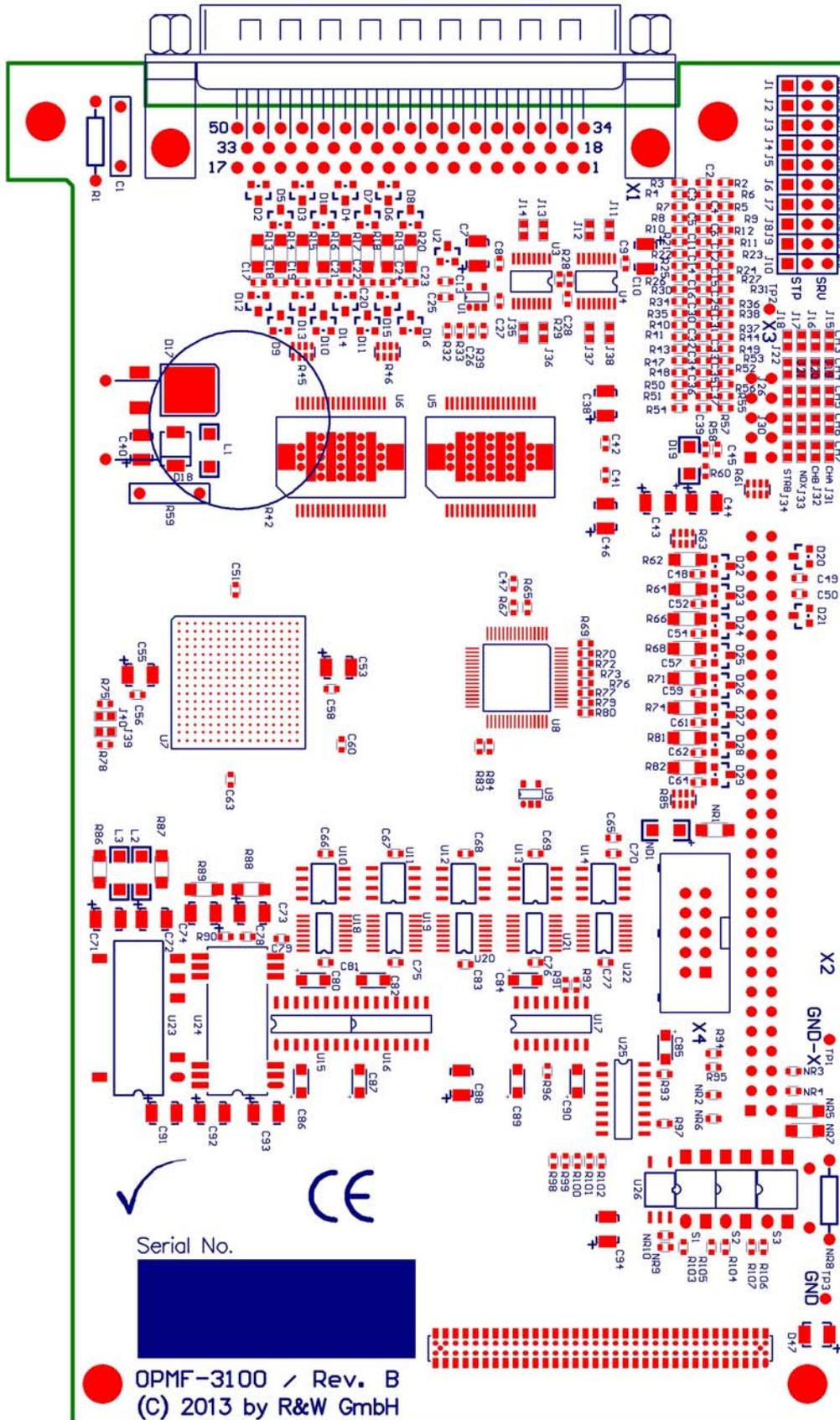
1.3.2 Bestückungsdruck des OPMF-3001 Top-Layer



1.3.3 Bestückungsdruck des OPMF-3001 Bottom-Layer



1.3.4 Bestückungsdruck des OPMF-3100 Top-Layer



1.3.6 Technische Daten des OPMF-3001/OPMF-3100

Merkmal	Beschreibung / Eigenschaften
Achskanäle	1, 2, 3, 4 oder 5. Gemischter Betrieb von Servo- oder Schrittmotoren möglich. Mit der MCU-3000-Grundplatine können somit bis zu maximal 8 Achsen angesteuert werden.
Gebereingänge	Richtungsdiskriminator für Inkrementalgeber mit 2 um 90° phasenverschobenen Impulsspuren und Nullimpuls, wahlweise deren invertierte Impulsspuren (6 Kanäle) oder SSI - Absolutwertgeber Impulspegel: 5V, TTL
Inkremental-Geberauswertung	4 fach, 32Bit mit Vorzeichen, 2.0 MHz, optional 5MHz (5 / 20MHz nach Vervielfachung)
SSI-Geberauswertung	1..32Bit, Gray-/Binär-Codes, variable Frequenz 30kHz .. 10MHz
Geberversorgung	externe Hilfsspannung je nach Gebertyp (5..30V)
Sollwertausgänge für Servo-Leistungsendstufen	16-Bit-DA-Wandler potentialfrei (optisch entkoppelt) <ul style="list-style-type: none"> • Ausgangsspannung +/-10V • Ausgangsstrom 5mA
Sollwertausgänge für Schrittmotor-Endstufen	RS422-Puls- und Richtungssignale und deren invertierte Impulsfolgen, Ausgangsstrom typisch: -60mA (max. -150mA) Impulsfrequenz: max. 10MHz
Digitale Eingänge	24 Eingänge optisch entkoppelt (Eingangsstrom typ. 8mA bei 24V) <ul style="list-style-type: none"> • UHmax: 30V / Eingangsstrom 12mA typ. • UHmin: 16V / Eingangsstrom 2mA typ. • ULmax: 10V / Eingangsstrom 0.3mA typ. • ULmin: 0V / Eingangsstrom 0mA typ. Funktionsweise aller Digital-Eingänge ist frei programmierbar.
Digitale Ausgänge	12 Ausgänge optisch entkoppelt, Ausgangstyp: PNP 24V, 500mA (interne Strombegrenzung bei 1A) Funktionsweise aller Ausgänge frei programmierbar, Sollzustand nach Reset programmierbar; Relaisausgänge max. 100mA/60V
Analoge Eingänge	1..8 differentiell, potentialfrei (optisch entkoppelt) <ul style="list-style-type: none"> • Auflösung 12bit • Eingangsspannungen: <ul style="list-style-type: none"> OPMF-3001: -5..5V, -10..+10V, 0..+5V und 0..+10V OPMF-3100: -5..5V, -10..+10V OPMF-3001: Eingangsspannungsbereich für jeden Kanal getrennt wählbar OPMF-3100: Eingangsspannungsbereich nur für alle Kanäle gemeinsam wählbar
Externe Stromversorgung	24V Stromaufnahme je nach Belastung der Digital-Ausgänge
Aufbau	Aufsteckplatine 106mm * 175mm, 6fach-Multilayer
PC-Stromversorgung	5V / ca. 0.5A in Maximalausbaustufe
Anschlussstecker	Achsperipherie <ul style="list-style-type: none"> • 50-poliger SUB-D-Steckverbinder für die Zusatzachsen 1..3 • 60-poliger FB-Steckverbinder für die Zusatzachsen 4..5 Freigaberelais für die Zusatzachsen 1..5 Analogeingänge 1..8

2 Option OPIBS

2.1 Kurzbeschreibung des INTERBUS Master-Interface

Die Option OPIBS dient zur Systemerweiterung der MCU-3000 unter Zuhilfenahme der international genormten **INTERBUS**-Feldbustechnik. Die MCU-3000-Grundplatine kann mit einer Sonderbestückung von wenigen zusätzlichen Bauteilen zum Interbus-Feldbus-Master aufgerüstet werden. Somit ist eine preisgünstige und flexible Anschaltung an zusätzliche Prozessperipherie möglich. Gerade bei räumlich verteilten Systemen oder sehr großer Anzahl von I/O-Punkten bietet die serielle Bustechnik gegenüber der konventionellen Parallelverdrahtung erhebliche Vorteile.

Beim OPIBS handelt es sich um ein INTERBUS-Master-Interface auf Basis des IBS UART Master Protokoll-Chips der Fa. Phoenix Contact. Der OPIBS wird über einen im *rwmos.ef*-Betriebsprogramm eingebundenen „Socket-Treiber“ angesprochen und stellt dem Anwender die wesentlichen Funktionen zur Verfügung um den INTERBUS zu kontrollieren und Prozess-Datenaustausch durchzuführen. Neben den E/A-Daten stellt der Treiber eine grundlegende Systemverwaltung und eine Basisdiagnose zur Verfügung.

Im Lieferumfang enthalten ist das Setup- und Diagnose-Programm *fwsetup.exe* mit dessen Hilfe die Einbindung in ein vom Anwender erstelltes PCAP oder SAP-Applikationsprogramm sehr vereinfacht wird.

Im Moment werden bis zu 512 Interbus-Teilnehmer und maximal 4096 I/O-Punkte durch die *rwmos.ef*-Betriebssystemsoftware unterstützt. Somit kann die MCU-3000 auf einfachste Weise mit neuen E/A-Funktionen ausgestattet werden. Am Markt ist hierzu eine große Anzahl unterschiedlichster Feldbusgeräte, z.T. sogar mit intelligenten Vorverarbeitungsfunktionen, verfügbar.

2.2 Software

Die OPIBS kann mit der Standard-TOOLSET-Software für die MCU-3000 projektiert und programmiert werden (ab Revision V2-50j). Die entsprechenden Zusatzfunktionen werden aufbauend auf dem Programmierhandbuch [PHB] der MCU-G3 in den nachfolgenden Kapiteln beschrieben.

2.2.1 Modifizierte Betriebssystemsoftware *rwmos.elf*

Damit die Zusatzfunktionen des OPIBS genutzt werden können, ist eine spezielle Betriebssystemsoftware *rwmos.elf* erforderlich! Diese Software ist Bestandteil des Lieferumfangs der Option OPIBS und wird nicht extra berechnet.

Die Interbus-Option kann mit Hilfe der PCAP-Programmierung als auch durch die SAP-Programmierung genutzt werden. Die im Lieferumfang enthaltenen Beispielprogramme veranschaulichen die einfache Handhabung dieser Zugriffsmethoden.

2.2.2 Einführung

Nachfolgend wird ein kurzer Überblick gegeben mit welchen Methoden die Option OPIBS softwaremäßig bedient wird. Um eine möglichst flexible und schnelle Konfiguration des vorgegebenen INTERBUS-Aufbaus zu realisieren, wurde folgender Weg beschritten:

Der Anwender definiert zunächst einen Busaufbau mit allen notwendigen Feldbusgeräten. Mit dem im Lieferumfang enthaltenen Konfigurationsprogramm *fwsetup.exe* wird der aktuelle Busaufbau des am MCU-3000 / OPIBS angeschlossenen Interbusses automatisch ermittelt. Die eingelesene Konfiguration kann zu Kontroll- und Dokumentationszwecken sowohl als ASCII-File wie auch als Binärfile abgespeichert werden. Zusätzlich sind dort weitere wichtige Informationen für die SAP-Programmierung enthalten.

Das Binärfile, in welchem der Sollzustand gespeichert ist, kann u.a. auch zum Vergleich mit der tatsächlichen Buskonfiguration verwendet werden. Für diesen Fall ist sichergestellt, dass ein veränderter Busaufbau in der endgültigen Anwendung nicht aktiviert werden kann.

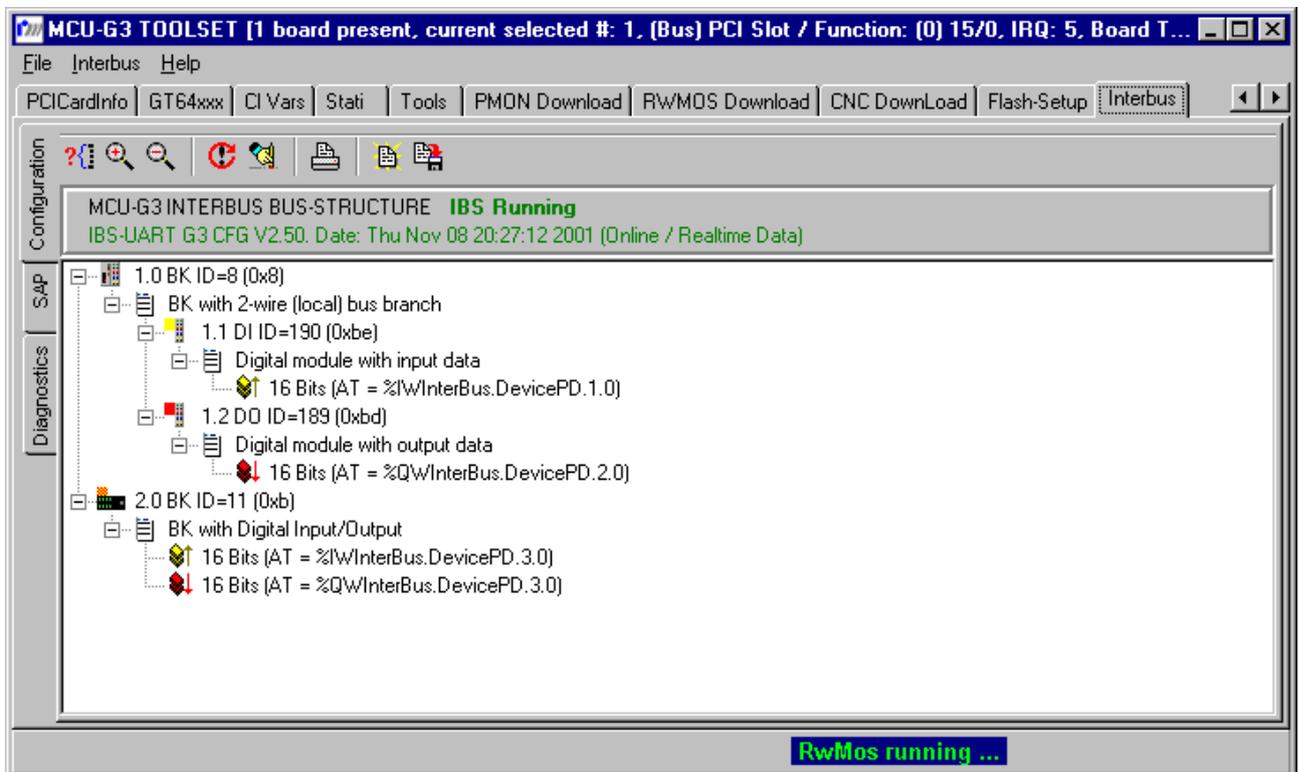
2.2.3 Hilfsprogramm fwsetup.exe

Mit Hilfe des TOOLSET-Programms *fwsetup.exe* kann der an der MCU-3000 angeschlossene Busaufbau automatisch ermittelt werden, sofern sich der angeschaltete Bus in einem ordnungsgemäßen Zustand befindet. Im Menü [Interbus] können die zusätzlichen Projektierungsmöglichkeiten für den OPIBS vorgenommen werden. Zur Zeit gibt es drei verschiedene Seiten auf denen wichtige Informationen über den aktuellen Busaufbau und Zustand ermittelt werden können. Verschiedene hierin enthaltene Daten werden für die SAP oder PCAP- Programmierung benötigt.

2.2.3.1 Interbus Seite: Configuration

Auf der Seite „Configuration“ kann wie bei nachfolgendem Bildschirmabdruck dargestellt, der am OPIBS angeschlossener Busaufbau in Baumform grafisch dargestellt werden. In der obersten Hierarchieebene werden die Buskoppler (Busknoten) mit Gerätekenung und Klartextbeschreibung aufgelistet. Darunter werden die einzelnen Module mit ihren spezifischen Eigenschaften wie Gerätekenung, Beschreibung, Zugriffstyp und Wortbreite angezeigt.

Abbildung 2-1: Interbus Configuration



Um diese Bildschirmseite aktivieren zu können, muss das Betriebssystem *rwmos.elf* bereits auf der Karte gestartet sein (Online-Mode, *RwMos running*).

Anhand nachfolgender Tabelle werden die grafischen Bedien-Symbole (Speed-Buttons) von obenstehendem Bildschirmabdruck erläutert.

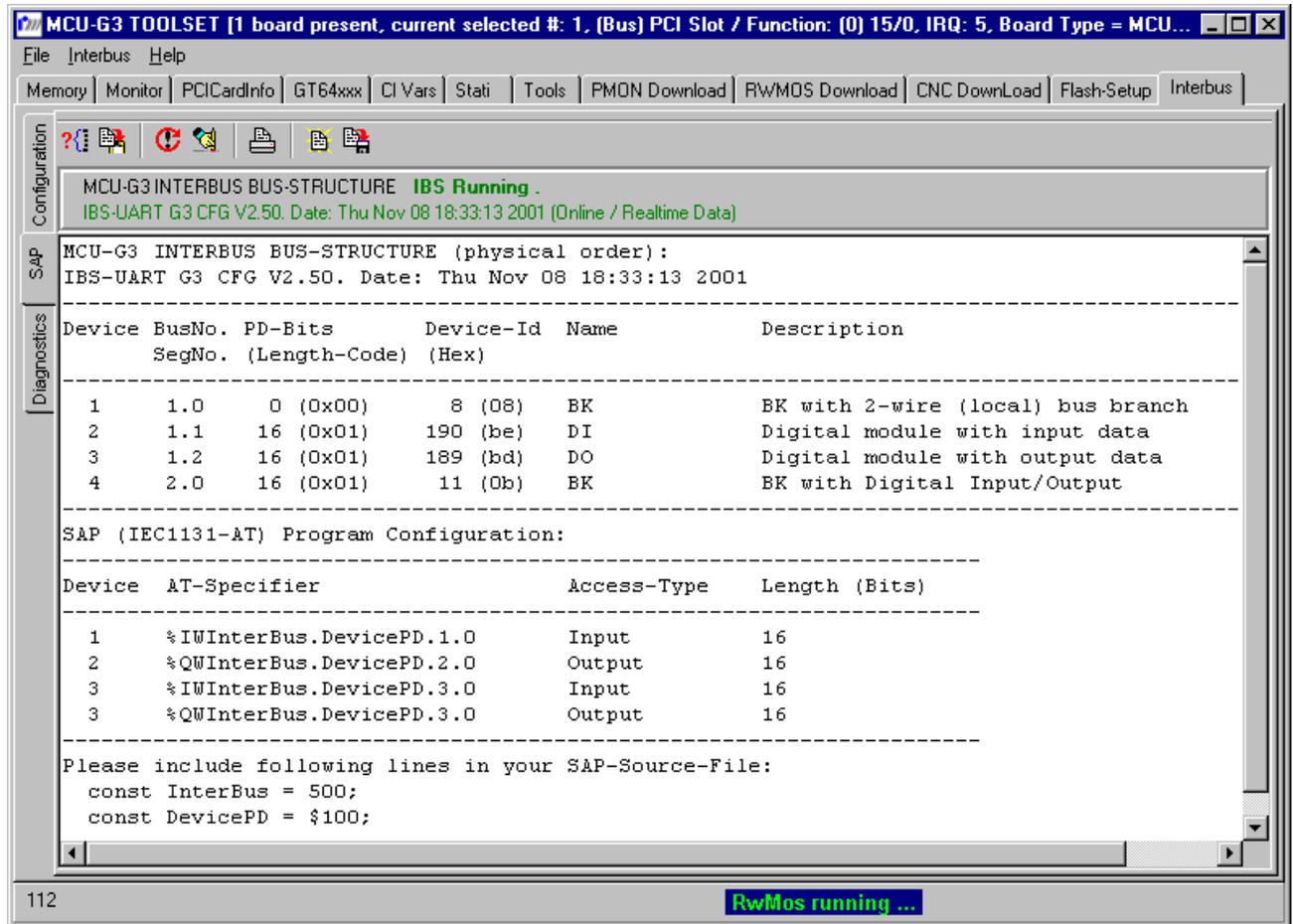
Tabelle 2-1: Bedeutung der Symbole auf der Seite [Interbus][Configuration]

Symbol	Bedeutung
	Interbus Initialisierung. Alle am Interbus vorhandenen Geräte werden automatisch ermittelt. Die gefundenen Teilnehmer werden am Bildschirm in einer grafischen Übersicht (wie oben) angezeigt. Nach erfolgreicher Initialisierung befindet sich der Interbus im Runmodus (IBS running ...). Vor dem Ausführen der Initialisierung werden ggf. gespeicherte Fehlermeldungen gelöscht.
	Zuletzt ermittelte Interbus-Konfiguration von der MCU-3000 einlesen. Es werden die Daten ausgewertet, welche bei der letzten Interbus – Initialisierung ermittelt wurden.
	Der aktuelle Geräte-Verzeichnisbaum wird erweitert. Es werden alle verfügbaren Informationen angezeigt.
	Der aktuelle Geräte-Verzeichnisbaum wird zusammengeklappt. Es werden nur noch Busknoten angezeigt.
	Sofern Fehler aufgetaucht sind, können diese mit Hilfe dieses Buttons gelöscht werden.
	Die aktuelle Konfiguration kann für Dokumentationszwecke auf einem Drucker ausgegeben werden.
	Die aktuelle Konfiguration kann in einer Datei binär gespeichert werden. Die Standard-Datei-Erweiterung ist „.cfg“ und steht für „Configuration Data“. Diese Daten können zur Weiterverarbeitung an einem anderen Arbeitsplatz oder für Archivierungs- oder Dokumentationszwecke genutzt werden.
	Eine gespeicherte Konfiguration kann eröffnet werden und am Bildschirm grafisch angezeigt werden.

2.2.3.2 Interbus Seite: SAP

Auf der Seite „SAP“ kann wie bei nachfolgendem Bildschirmabdruck dargestellt, der am OPIBS angeschlossener Busaufbau in Textform dargestellt werden.

Abbildung 2-2: Interbus SAP



Um diese Bildschirmseite aktivieren zu können, muss das Betriebssystem `rwmos.elf` bereits auf der Karte gestartet sein (Online-Mode, `RwMos running`).

2.2.3.2.1 Erläuterung der grafischen Bediensymbole

Anhand nachfolgender Tabelle werden die grafischen Symbole (Speed-Buttons) von obenstehendem Bildschirmabdruck erläutert.

Tabelle 2-2: Bedeutung der Symbole auf der Seite [Interbus][SAP]

Symbol	Bedeutung
	Siehe Tabelle Interbus Configuration
	Siehe Tabelle Interbus Configuration
	Siehe Tabelle Interbus Configuration
	Siehe Tabelle Interbus Configuration
	Siehe Tabelle Interbus Configuration
	Siehe Tabelle Interbus Configuration
	Siehe Tabelle Interbus Configuration
	Mit Hilfe dieses Buttons können die nach IEC1131 eingeführten AT-Spezifizierer zur Verwendung in die Windows-Zwischenablage zur weiteren Verwendung in einem SAP-Programm kopiert werden. Die Windows-Zwischenablage enthält nach Betätigen dieses Speed-Buttons für obenstehenden Busaufbau folgende Informationen:
	%IInterBus.DevicePD.1.0 %QInterBus.DevicePD.2.0 %IInterBus.DevicePD.3.0 %QInterBus.DevicePD.3.0
	Diese Definitionen können zur Variablendeklaration in einem SAP-Programm verwendet werden.

2.2.3.2.2 Erläuterung der Interbus SAP Bildschirmtextausgabe

Tabelle 2-3: Bildschirmausgabe Interbus SAP

```

MCU-G3 INTERBUS BUS-STRUCTURE (physical order):
IBSCFG REV 2.51. Date: Tue Oct 30 08:30:42 2001
-----
Device BusNo. PD-Bits      Device-Id Name      Description
      SegNo. (Length-Code) (Hex)
-----
  1     1.0     0 (0x00)    8 (08)  BK 2L-LB   BK with 2-wire (local) bus branch
  2     1.1    16 (0x01)   190 (be) DI        Digital module with input data
  3     1.2    16 (0x01)   189 (bd) DO        Digital module with output data
  4     2.0    16 (0x01)    11 (0b) BK-I/O-T   BK with Digital Input/Output
-----
SAP (IEC1131-AT) Program Configuration:
-----
Device  AT-Specifier              Access-Type  Length (Bits)
-----
  1     %IWinterBus.DevicePD.1.0  Input       16
  2     %QWinterBus.DevicePD.2.0  Output      16
  3     %IWinterBus.DevicePD.3.0  Input       16
  3     %QWinterBus.DevicePD.3.0  Output      16
-----
Please include following lines in your SAP-Source-File:
const InterBus = 500;
const DevicePD = $200;

```

Alle vorhandenen Interbus-Geräte erhalten eine fortlaufende Nummer (Spalte *Device*). Diese Nummer entspricht der tatsächlichen Position im Busaufbau. Die Spalte *BusNo.SegNo.* spezifiziert zu welchem Bussegment das jeweilige Gerät gehört und welche fortlaufende Nummer es in diesem Segment besitzt. Die Spalte *PD-Bits (LengthCode)* zeigt an welcher Adressraum durch das jeweilige Gerät im Host benötigt wird. Die Spalte *Device-Id* dient zur Beschreibung der Gerätefunktion. Normalerweise sind die ID-Codes auf den Interbus-Modulen in dezimaler und bei RT-Modulen in hexadezimaler Schreibweise aufgedruckt. Die Spalten *Name* und *Description* beschreiben das jeweilige Modul im Klartext.

Für die SAP- und PCAP-Programmierung sind ebenfalls wertvolle Informationen in dieser Bildschirmausgabe enthalten. Dazu gehören zunächst alle Typdefinitionen, mit deren Hilfe die jeweiligen Feldbusgeräte angesprochen werden können. Die Spalte *AT-Specifier* enthält die Typdefinitionen zur Deklaration von Systemvariablen für die SAP-Programmierung und enthält auch die notwendigen Informationen für den Objektzugriff mit Hilfe der PCAP-Programmierung. In der SAP-Programmiersprache wird in Anlehnung an die IEC1131-Programmiersprache der Zugriff auf die E/A-Ebene mit Hilfe des Schlüsselwortes AT geregelt [G3-Universelles-Objekt-Interface / Kapitel 2.2].

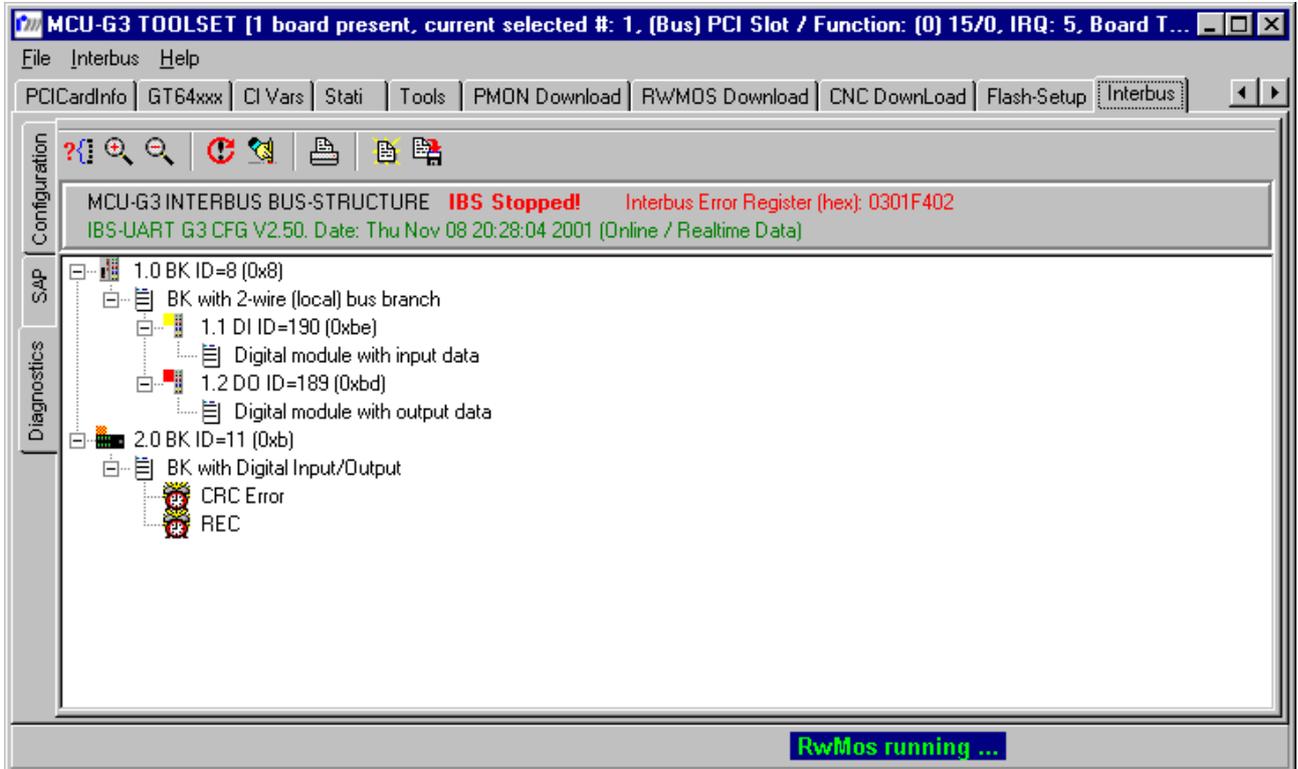
Mit Hilfe der Spalte *Access-Type* kann ermittelt werden, wie auf die jeweilige Systemvariable zugegriffen werden darf also lesend, schreibend oder beides. Die Spalte *Length* zeigt an, wie viele Bits in der jeweiligen SAP-Variable nutzbar sind.

Anmerkung: für den obenstehenden Busaufbau gibt es ein SAP-Demoprogramm, in welchem die flexible und einfache Interbuskommunikation demonstriert wird.

2.2.3.3 Interbus Seite: Diagnostics

Auf der Seite „Diagnostics“ können quasistatische Fehler des Interbusses, wie bei nachfolgendem Bildschirmabdruck dargestellt, in Baumform grafisch dargestellt werden.

Abbildung 2-3: Interbus Diagnostics



Um diese Bildschirmseite aktivieren zu können, muss das Betriebssystem rwmos.elf bereits auf der Karte gestartet sein (Online-Mode, RwMos running).

Anhand nachfolgender Tabelle werden die grafischen Symbole (Speed-Buttons) von obenstehendem Bildschirmabdruck erläutert.

Tabelle 2-4: Bedeutung der Symbole auf der Seite [Interbus][Diagnostics]

Symbol	Bedeutung
	Siehe Tabelle Interbus Configuration
	Zuletzt gespeicherten quasistatische Interbus-Fehler von der MCU-G3 einlesen. Es werden die Daten ausgewertet, welche beim Ausführen des letzten Interbus – Diagnosezyklus ermittelt wurden.
	Siehe Tabelle Interbus Configuration

2.2.4 Interbus SAP-Programmierung

Die Software-Zusatzfunktionen der Option OPIBS kann neben der PCAP-Programmiermethode auch sehr effektiv durch die SAP-Programmiermethode genutzt werden.

Achtung: Zugriffe auf die Option OPIBS sollten nur aus einer Task erfolgen. Ansonsten muss gewährleistet werden, dass nur sequentiell (nicht gleichzeitig) aus mehreren Tasks auf die Option OPIBS zugegriffen wird. Ist dies nicht gewährleistet, kann dies zu einem unerklärlichen Laufzeitverhalten führen!

2.2.4.1 Interbus Status-Informationen und Kommandos

Die Interbus-Funktionalität kann durch Lesen und Schreiben vordefinierter Systemvariablen genutzt werden. Mit Hilfe dieser Variablen kann direkt mit dem im *rwmos.elf*-Betriebsprogramm integrierten „Socket-Treiber“ kommuniziert werden. Die Definitionen der nachfolgend beschriebenen Variablen befinden sich im Include-File *INTERBUS_AT_SPECS.INC*. Wie weiter oben bereits erwähnt wird der Variablenzugriff in Anlehnung an die IEC1131-Programmiersprache mit Hilfe des AT-Mechanismus geregelt. Dieser wird ausführlich im [G3-Universelles-Objekt-Interface / Kapitel 2.2] beschrieben.

Tabelle: 2-5 Interbus Status-Informationen und Kommandos

System-Parameter	Wert / Opcode	Beschreibung
IBSReleaseHandles_w	1	Zuweisung wenn alle AT-Descriptoren verworfen und neu aufgebaut werden sollen.
IBSOpen_r	0 1	Überprüfen ob Interbusstreiber bereits geöffnet wurde. Interbus-Treiber geschlossen Interbus-Treiber geöffnet
IBSOpen_w	1	Initialisiert den gesamten Treiber (Speicher, Timer), führt einen Hardware-Reset des Protokoll-Chips IBS UART aus und stellt die gewünschte Baudrate ein. Diese Funktion muss am Anfang jeder Applikation zuerst aufgerufen werden. Andernfalls ist die Herstellung einer Kommunikationsverbindung zwischen Anwenderprogramm und OPIBS nicht möglich. Wichtig: Das Kommando sollte nur ausgeführt werden, wenn der Treiber noch nicht geöffnet wurde. Dies kann mit <i>IBSOpen_r</i> festgestellt werden. Siehe dazu auch Fehler-Codes [Kapitel 2.2.6].
IBSClose_r	0 1	Überprüfen ob Interbusstreiber bereits geschlossen wurde. Interbus-Treiber geöffnet Interbus-Treiber geschlossen
IBSClose_w	1	Gibt zuvor mit <i>IBSOpen_w</i> initialisierte Ressourcen wie Speicher und Timer wieder frei. Außerdem wird ein Hardware-Reset des Protokoll-Chips IBS UART durchgeführt. Diese Funktion sollte unbedingt vor Verlassen des Programms ausgeführt werden. Siehe dazu auch Fehler-Codes [Kapitel 2.2.6].
IBSInit_r	0 1	Überprüfen ob Interbus bereits initialisiert wurde. Interbus wurde noch nicht initialisiert Interbus wurde bereits initialisiert
IBSInit_w	1	Nimmt das INTERBUS-Netzwerk in Betrieb und gibt, wenn möglich, die aktuell lauffähige Konfiguration an das <i>rwmos.elf</i> -Betriebsprogramm zurück. Deshalb wird diese Funktion auch zur Fehlerlokalisierung genutzt. Siehe dazu auch Fehler-Codes [Kapitel 2.2.6]. Hinweis: Dieser Befehl nimmt das System für ca. 1 Sekunde in Anspruch. In dieser Zeit werden PC und SAP-Zugriffe ausgesetzt. Um Fehlfunktionen z.B. beim Start einer Task in <i>mcfg</i> zu verhindern, kann vor diesem Kommando eine Wartezeit (z.B. 100ms) eingefügt werden.

Tabelle Fortsetzung

System-Parameter	Wert / Opcode	Beschreibung
IBSRun_w	1	In dieser Funktion wird ein sicherer Datenaustausch durchgeführt. Dazu gehört, dass pro IBSRun_w-Schreibzugriff ein Datenzyklus angestoßen wird. Wird während dieses Zyklus ein Fehler entdeckt, wird die Funktion mit der entsprechenden Fehlermeldung verlassen. <u>Wichtig:</u> Diese Funktion ist nur für Testzwecke vorhanden und sollte nicht direkt aufgerufen werden. Der zyklische Aufruf erfolgt automatisch durch die <i>rwmos.elf</i> -Betriebssystemsoftware. Siehe dazu auch Fehler-Codes [Kapitel 2.2.6].
IBSReset_w	1	Versetzt alle INTERBUS-Teilnehmer in den Reset-Zustand bis die Funktion IBUInit_w aufgerufen wird. Siehe dazu auch Fehler-Codes [Kapitel 2.2.6].

Anmerkung: alle Variablen vom Typ `.._w` sind Schreib- und `.._r` sind Leseregister.

2.2.4.2 Vorgehensweise zur Verwendung des Interbus

2.2.4.3 Interbus Konfigurationsparameter

Die Interbus-Funktionalität kann durch Lesen und Schreiben vordefinierter Systemvariablen genutzt werden. Mit Hilfe dieser Variablen kann direkt mit dem im *rwmos.elf*-Betriebsprogramm integrierten „Socket-Treiber“ kommuniziert werden. Die Definitionen der nachfolgend beschriebenen Variablen befinden sich im Include-File *INTERBUS_AT_SPECS.INC*. Wie weiter oben bereits erwähnt wird der Variablenzugriff in Anlehnung an die IEC1131-Programmiersprache mit Hilfe des AT-Mechanismus geregelt. Dieser wird ausführlich im [G3-Universelles-Objekt-Interface / Kapitel 2.2] beschrieben.

Tabelle: 2-6 Interbus Features

System-Parameter	Wert / Opcode	Beschreibung
IBSMaxRep_r		
IBSMaxRep_w		
IBSMaxDevice_r	402 hex	Maximale Anzahl der IBS Geräte lesen
IBSMaxDevice_w	402 hex	Maximale Anzahl der IBS Geräte schreiben
IBSMaxIoPoints_r		
IBSMaxIoPoints_w		
IBSBusyMode_r		
IBSBusyMode_w		
IBSExtraCycle_r		
IBSExtraCycle_w		
IBSStatusText_w		
IBSErrorReg_r		Interbus Fehlermeldungen lesen, Erläuterungen in Kapitel 2.2.6
IBSErrorReg_w		

Anmerkung: Im SAP-Anwenderprogramm sollte der INTERBUS überwacht werden z.B. durch Statusabfragen in einer Interbus-Event-Handler-Routine.

2.2.4.4 Interbus Fehlerbehandlung

Zur Fehlerbehandlung von Interbus-Fehlermeldungen kann der Eventhandler EVIBS verwendet werden. Vor der Aktivierung des Events sollte die Variable IBSErrorReg_w genullt werden. In der Fehleroutine kann dann der Inhalt der Variablen IBSErrorReg_r ausgewertet bzw. angezeigt werden.

Hinweis: Die Variable IBSErrorReg_r ist nur eine 16bit Ganzzahlvariable. Bei Auswertung als integer-Variable mit 32 bit, sind die oberen 16 bit ohne Bedeutung und sollten ausgeblendet werden.

2.2.5 Interbus PCAP-Programmierung

Die Zugriffe per PCAP-Programmierung werden über das universelle Objekt-Interface abgewickelt.

2.2.6 Interbus Fehlermeldungen (Fehlercodes)

2.2.6.1 Allgemeines

Die Fehler sind so definiert, dass jeder Fehlercode eindeutig ist, aber u.U. an verschiedenen Stellen im Socket-Treiber der Betriebssystem-Software `rwmos.elf` erzeugt werden kann. Alle Funktionen geben einen 32-Bit-Wert, den `Result`, zurück. Dieser enthält bei Fehlerfällen den treiberinternen Fehlerort und einen definierten Fehlercode. Damit ist eine einfache Auswertung und schnelle Bearbeitung durch die aufrufende Funktion gegeben. Für eine Auswertung ist nur der Fehlercode in den letzten beiden Bytes des `result`-Wortes relevant.

2.2.6.2 Logische Fehler (F0XX_{hex})

Tabelle 2-7: Logische Fehler

Code	Bedeutung	Erläuterung
F001 _{hex}	IBU_COMNo_is_not_open	Die angegebene COM-Schnittstelle ist nicht geöffnet. Bitte öffnen Sie diese mit der Funktion <code>IBU_Open()</code>
F002 _{hex}	IBU_comno_already_open	Diese Schnittstelle ist bereits geöffnet und wird benutzt
F003 _{hex}	IBU_comno_not_supported	Diese Schnittstelle wird nicht unterstützt
F004 _{hex}	IBU_Function_not_active	Interner Fehler. Der Status einer nicht aktiven Funktion wurde abgefragt
F005 _{hex}	IBU_comresource_already_in_use	Die entsprechende COM-Schnittstelle ist nicht frei
F506 _{hex}	IBU_no_timer	Timer nicht verfügbar
F507 _{hex}	IBU_memory_not released	Bei der Ausführung der Funktion <code>IBU_Close()</code> konnten die Ressourcen (Speicher bzw. IRQ) nicht wieder freigegeben werden
F508 _{hex}	IBU_timer_not released	Bei der Ausführung der Funktion <code>IBU_Close()</code> konnten die Ressourcen (Timer) nicht wieder freigegeben werden
F009 _{hex}	IBU_no_of_recchar_not_supported	Zu viele Zeichen empfangen (mehr als 512) (Maximal Wert überschritten)
F00A _{hex}	IBU_no_of_sendchar_not_supported	Zu viele Zeichen gesendet (mehr als 512) (Maximal Wert überschritten)
F00B _{hex}	IBU_no_setup_started	Das angeschlossene INTERBUS-System wurde mit <code>IBS_Init()</code> nicht eingelesen
F00C _{hex}	IBU_Parameter_invalid_range	Wert außerhalb des Bereiches
F00D _{hex}	IBU_Parameter_not_supported	Parameter wird von der Hardware nicht unterstützt
F00E _{hex}	IBU_api_required_crc_error	Die API-Schicht des Treibers hat einen ungültigen Zyklus angefordert

2.2.6.3 Port-Fehler (F1XX_{hex})

Tabelle 2-8: Port-Fehler

Code	Bedeutung	Erläuterung
F101 _{hex}	IBU_baudrate_not_supported	Die angegebene Baudrate wird vom Hostsystem nicht unterstützt.
F102 _{hex}	IBU_parity_not_supported	Even Parity wird vom Hostsystem nicht unterstützt.
F103 _{hex}	IBU_stopbits_not_supported	Anzahl der Stopbits (2) wird vom Hostsystem nicht unterstützt.
F104 _{hex}	IBU_databits_not_supported	Anzahl der Datenbits (8) wird vom Hostsystem nicht unterstützt.
F105 _{hex}	IBU_frame_error (Host receive)	Übertragungsfehler auf der seriellen Schnittstelle (Frame Error) (IBS UART → Hostsystem)
F106 _{hex}	IBU_host_parity_error	Übertragungsfehler auf der seriellen Schnittstelle (Parity Error) (IBS UART → Hostsystem)
F107 _{hex}	IBU_ibs_uart_parity_error	Übertragungsfehler auf der seriellen Schnittstelle (Hostsystem → IBS UART)
F108 _{hex}	IBU_send_error	Fehler beim Senden zum IBS UART
F109 _{hex}	IBU_receive_error	Fehler beim Empfangen vom IBS UART
F110 _{hex}	IBU_timer_is_already_running	Interner Fehler. Ein gestarteter Timer kann nicht noch einmal gestartet werden.
F111 _{hex}	IBU_timervalue_not_supported	Interner Fehler. Timer-Wert wird vom Host-System nicht unterstützt
F112 _{hex}	IBU_timer_is_stopped	Interner Fehler. Ein gestoppter Timer kann nicht noch einmal gestoppt werden.
F113 _{hex}	IBU_timer_timeout	Interner Fehler. Timer ist abgelaufen.

Tabelle: Fortsetzung Port-Fehler

Code	Bedeutung	Erläuterung
F120 _{hex}	IBU_serial_timeout	Keine serielle Verbindung zum INTERBUS UART. Mögliche Ursache: <ul style="list-style-type: none"> • Keine Spannung am IBS UART • DTR-Signal fehlt • Die eingestellten Übertragungsgeschwindigkeiten auf dem IBS UART und dem Hostsystem stimmen nicht überein
F121 _{hex}	IBU_overnun_error	Überlauf auf der seriellen Schnittstelle - festgestellt durch das Hostsystem
F122 _{hex}	IBU_break_error	Unterbrechung der Daten-übertragung auf der seriellen Schnittstelle - festgestellt durch das Hostsystem

2.2.6.4 Logische Fehler in der seriellen Kommunikation (F2XX_{hex})

Tabelle 2-9: Logische Fehler in der seriellen Kommunikation

Code	Bedeutung	Erläuterung
F201 _{hex}	IBU_too_many_char_received	Mehr Zeichen empfangen als erwartet
F202 _{hex}	IBU_delay_error	Das zulässige INTERBUS-Delay (Bus-delay) wurde überschritten; siehe (IBU_ML_Set_Bus_Delay ())
F203 _{hex}	IBU_send_finished_and_rec_busy	Weniger Zeichen empfangen als erwartet
F204 _{hex}	IBU_send_busy_and_rec_busy	Nicht alle Zeichen konnten gesendet bzw. empfangen werden
F205 _{hex}	IBU_send_busy_and_rec_finished	Nicht alle Zeichen konnten gesendet werden
F206 _{hex}	IBU_Wrong_State_Key	Treiber und Hardware sind nicht mehr synchron; Beispiel: Die Confirmation auf eine Dienstprimitive wurde von dem Treiber falsch empfangen.
F207 _{hex}	IBU_Wrong_State_Type	Treiber und Hardware sind nicht mehr synchron; Beispiel: Die Confirmation auf eine Dienstprimitive wurde von dem Treiber falsch empfangen.
F208 _{hex}	IBU_primitive_Error	Auf einen Request wurde vom Treiber eine falsche Confirmation empfangen
F209 _{hex}	IBU_service_Error	Treiber und Hardware sind nicht mehr synchron; Beispiel: Ein von der Applikation (Treiber) gesendeter Reset-Befehl erreicht nicht den UART-Chip, weil z.B. die Verbindung unterbrochen war

2.2.6.5 INTERBUS Fehler (F3XX_{hex} - F4XX_{hex})

Tabelle 2-10: INTERBUS Fehler

Code	Bedeutung	Erläuterung
F3X _{hex}	IBS_ibs_result_error	Für die Bedeutung des Result-Bytes („XX _{hex} “) siehe Tabelle 2-11
F400 _{hex}	IBU_ibs_timeout	Unterbrechung eines INTERBUS- Kabels od. defektes INTERBUS-Gerät
F401 _{hex}	IBU_lbw_error	Loop-back-Wort (LBW) Fehler; Das ausgesendete LBW konnte nicht wieder korrekt empfangen werden.
F402 _{hex}	IBU_crc_error	Kurzzeitige Störung auf dem INTERBUS; Mögliche Ursache: EMV-Wirkung
F403 _{hex}	IBU_no_Devices_are_connected	An den INTERBUS-Master sind keine Feldgeräte angeschlossen
F404 _{hex}	IBU_no_valid_input_data	Keine gültigen Eingangsdaten
F410 _{hex}	IBU_ibs_timeout_init_ri	Unterbrechung eines Kabels od. defektes INTERBUS-Gerät an der weiterführenden Schnittstelle des angegebenen Teilnehmers siehe Funktion: IBU_I-API_Set_Max_Devices()
F411 _{hex}	IBU_ibs_timeout_init_bi	Unterbrechung eines Kabels od. defektes INTERBUS-Gerät an der Stich-Schnittstelle des angegebenen Teilnehmers siehe Funktion: IBU_I-API_Set_Max_Devices()
F412 _{hex}	IBU_data_register_error	Datenregister-Fehler; Die physikalische Datenregisterlänge stimmt nicht mit der logischen Datenregisterlänge des angegebenen Teilnehmers überein.
F413 _{hex}	IBU_too_many_devices_conf_not_valid	Die Zahl der angeschlossenen INTERBUS-Geräte übersteigen die Maximale eingestellte Grenze bzw. 512 siehe Funktion: IBU_I-API_Set_Max_Devices()

Tabelle: Fortsetzung INTERBUS Fehler

Code	Bedeutung	Erläuterung
F414 _{hex}	IBU_init_too_many_io_points	Die Zahl der angeschlossenen INTERBUS-I/O-Punkte übersteigen die Maximale Grenze von 4096 Eingängen und 4096 Ausgängen Teilnehmern
F415 _{hex}	IBU_too_many_errors	Während der Ausführung der IBU_IBS_Init() - Funktion sind zu viele Fehler aufgetreten Mögliche Ursachen: <ul style="list-style-type: none"> • max_repeat zu klein →IBU_I_API_Set_MaxRep() • BusDelay zu klein →IBU_ML_Set_Bus_Delay ()
F416 _{hex}	IBU_init_timeout	IBU_IBS_Init() konnte nicht innerhalb der vorgegebenen Zeit beendet werden Mögliche Ursachen: <ul style="list-style-type: none"> • max_repeat zu klein →IBU_I_API_Set_MaxRep() • BusDelay zu klein →IBU_ML_Set_Bus_Delay ()
F417 _{hex}	IBU_too_many_devices_config_valid	Die zuvor eingestellte Maximale Teilnehmerzahl ist gültig
F420 _{hex}	IBU_ibs_loop_diag_no_loop	Es ist kein System mit der Funktionalität „Loop Diagnose“ angeschlossen
F421 _{hex}	IBU_ibs_loop_diag_not_stable	Bei der teilnehmerorientierten Lokaltbusdiagnose wird kein stabiler Zustand erreicht Mögliche Ursache: Wackelkontakt

Tabelle 2-11: Ergebnis-Byte (Result byte)

Codierung (1 Byte)	Bedeutung	Erklärung
00 _{hex}	Kein Fehler	System läuft fehlerfrei.
01 _{hex}	CRC-Fehler letzter Weg	Empfangene und berechnete CR-Prüfsumme stimmen nicht überein. Auf dem vom IBS UART überwachten Abschnitt ist eine Störung aufgetreten.
02 _{hex}	CR-Fehler	CR-Flag im Zeichen-Kopf des gesendeten und empfangenen Zeichens stimmen nicht überein.
04 _{hex}	SL-Fehler	SL-Flag im Zeichen-Kopf des gesendeten und empfangenen Zeichens stimmen nicht überein.
08 _{hex}	NOISE-Fehler	Nach dem erwarteten Ende einer FCS wurde ein "nachlaufendes Telegramm " von der IBS UART-Hardware erkannt.
10 _{hex}	STOP-Fehler	STOP-Bit-Fehler. Anstelle des erwarteten "STOP-Bit" eines INTERBUS-Telegramms wurde von der IBS UART-Hardware ein High-Signal erkannt.
20 _{hex}	MAU-Fehler	Es wurde ein statisches "High"-Signal mit mehr als 64 Bit-Zeiten Länge am INTERBUS-Eingang (Anschluß DI) erkannt . Dieser Fehler tritt auch auf, wenn kein Gerät am IBS UART angeschlossen ist.
40 _{hex}	Parity-Fehler	Über das Parity-Bit der seriellen asynchronen Kommunikation wurde von der IBS UART-Hardware ein Übertragungsfehler erkannt
80 _{hex}	Service-Fehler	<ul style="list-style-type: none"> - Primitive nicht definiert - Beim Senden des Ergebnis-Bytes wurde ein weiterer Fehler erkannt - Primitive im derzeitigen Zustand nicht erlaubt wenn z.B. der IBS UART noch im Zustand "Reset" ist, sind nur die Primitives "ResetStop" oder "set" erlaubt.

2.2.6.6 Allgemeiner Fehler

Dieser ist ein Fehler, der nicht näher bestimmt werden kann (FFFF_{hex}).

Code	Bedeutung	Erläuterung
FFFF _{hex}	Internal_Error	Interner Fehler

2.3 Interbus Hardware

Die Option OPIBS kann als Bestückungsvariante auf dem Grundgerät MCU-3000 bezogen werden. d.h. es wird kein weiterer Steckplatz für die Interbus-Ankopplung benötigt. Die Peripheriegeräte werden mit Hilfe von 10poligen FB-Steckverbindern auf der MCU-3000 angeschlossen. Für die MCU-3000 ist u.a. auch ein Kabelsatz mit Kartenhalter und einem integrierten 9pol. SUB-D-Steckverbinder erhältlich. Dieser Kartenhalter kann in einen freien PC-Slot neben der Steuerung eingebaut werden. Ggf. kann aber auch ein vorhandener Ausbruch der PC-Mechanik für diesen Zweck verwendet werden.

Anmerkung: Der 10-polige FB-Steckverbinder P1 ist mit Interbus gekennzeichnet.

2.3.1 Stecker P1 (Interbus): 9-poliger SUB-D-Steckverbinder (Buchse)

Am Stecker P1 wird der Fernbus (2-Leiter) des OPIBS Interbus-Systems angeschlossen. Sie ist am Kartenhalter als 9-polige D-SUB-Buchse ausgeführt und galvanisch vom Potential des Host-PCs getrennt. Das Gehäuse der Buchse ist leitend mit dem PC-Kartenhalter verbunden. Die Steckerbelegung wurde so gewählt, dass die Standard Interbus-Fernbuskabel (z.B. IBS RBC METER-T) von PHOENIX CONTACT oder anderen Herstellern verwendet werden können.

Tabelle 2-12: Anschlussbelegung der Fernbusschnittstelle (SUB-D)

Pin SUB-D	Name	Beschreibung / Anmerkungen	Quelle (FB)
1	DO	Fernbus Sendeleitung	MCU3000 FB-10-P1.1
2	DI	Fernbus Empfangsleitung	MCU3000 FB-10-P1.3
3	COM	Masse	MCU3000 FB-10-P1.5
4		Nicht belegt / reserviert	
5	+5V	Interne Versorgungsspannung	MCU3000 FB-10-P1.9
6	/DO	Fernbus Sendeleitung invertiert	MCU3000 FB-10-P1.2
7	/DI	Fernbus Empfangsleitung invertiert	MCU3000 FB-10-P1.4
8		Nicht belegt / reserviert	
9		Nicht belegt / reserviert	

2.3.2 Technische Daten der Option OPIBS

Merkmal	Beschreibung / Eigenschaften
Max. Anzahl Interbus-Teilnehmer	512
Aufbau	Bestückungsvariante auf der Grundplatine MCU-3000
Max. Anzahl der Prozessdaten	256 Worte (4096 binäre Ein- und Ausgänge)
Max. Anzahl der PCP-Teilnehmer	62 (in Vorbereitung)
Anschlussstecker	10-poliger FB-Steckverbinder (P1) für Interbus-Peripherie-Anschaltung

3 SUB-D-Stecker, für Freigabe-Relais, CNC-Bereit-Relais

3.1 SUB-D-Adapter von 10-poligem FB-Stecker X5 auf 9-poligen SUB-D-Stecker

Am Steckverbinder X5 werden Relais-Kontakte für die CNC-Bereit-Abfrage und Verstärkerfreigaben zur Verfügung gestellt. Es handelt sich hierbei um Schließer. Alle Relais sind nach Einschalten des PC, nach einem Rücksetzvorgang oder nach einem Fehler abgeschaltet.

Das CNC-Bereit-Relais wird nach dem Bootvorgang (*mcbt.exe*) aktiviert.

Die Freigabe-Relais werden beim PCAP-Befehl *cl()* und beim SAP-Befehl *CL()* für die entsprechend selektierten Achskanäle aktiviert.

Mit diesem Adapterkabel wird der Stecker X-5 auf einen 9-poligen Sub-D-Steckverbinder konvertiert. Dieser Steckverbinder ist an einem PC-Slotblech montiert. Durch die Konvertierung ergibt sich eine andere Steckerbelegung als an X5.

Tabelle 3-1: Anschlussbelegung Relais-Adapter (SUB-D-09)

Pin SUB-D	Name	Beschreibung / Anmerkungen	Quelle
1	P-Kontakt	CNC-Betriebsbereit	MCU3000 FB-10-P5.1
2	P-Kontakt	Freigabe Achskanal 1	MCU3000 FB-10-P5.3
3	P-Kontakt	Freigabe Achskanal 2	MCU3000 FB-10-P5.5
4	P-Kontakt	Freigabe Achskanal 3	MCU3000 FB-10-P5.7
5		nicht belegt	
6	Schliesser	CNC-Betriebsbereit	MCU3000 FB-10-P5.2
7	Schliesser	Freigabe Achskanal 1	MCU3000 FB-10-P5.4
8	Schliesser	Freigabe Achskanal 2	MCU3000 FB-10-P5.6
9	Schliesser	Freigabe Achskanal 3	MCU3000 FB-10-P5.8

3.2 SUB-D-Adapter von 10-poligem FB-Stecker X5 bzw. X2 auf 25-poligen SUB-D-Stecker männlich

Am Steckverbinder X5 der MCU-3000 und an den Pins 1-10 von X2 des Optionsprints OPMF-3001 werden Relais-Kontakte für die CNC-Bereit-Abfrage und Verstärkerfreigaben zur Verfügung gestellt. Es handelt sich hierbei um Schließer. Somit stehen die Freigaberelais für maximal 8 Achsen auf einem Stecker zur Verfügung. Alle Relais sind nach Einschalten des PC, nach einem Rücksetzvorgang oder nach einem Fehler abgeschaltet.

Das CNC-Bereit-Relais wird nach dem Bootvorgang (*mcbt.exe*) aktiviert.

Die Freigabe-Relais werden beim PCAP-Befehl *cl()* und beim SAP-Befehl *CL()* für die entsprechend selektierten Achskanäle aktiviert.

Mit diesem Adapterkabel wird der Stecker X-5 der MCU-3000 und die Pins 1-10 von X2 des OPMF-3001 auf einen 25-poligen Sub-D-Steckverbinder konvertiert. Dieser Steckverbinder ist an einem PC-Slotblech montiert. Durch die Konvertierung ergibt sich eine andere Steckerbelegung als an X5 / X2.

Tabelle 3-2: Anschlussbelegung Relais-Adapter (SUB-D-25)

Pin SUB-D	Name	Beschreibung / Anmerkungen	Quelle
1	P-Kontakt	CNC-Betriebsbereit	MCU3000 FB-10-P5.1
2	P-Kontakt	Freigabe Achskanal 1	MCU3000 FB-10-P5.3
3	P-Kontakt	Freigabe Achskanal 2	MCU3000 FB-10-P5.5
4	P-Kontakt	Freigabe Achskanal 3	MCU3000 FB-10-P5.7
5			
6	P-Kontakt	Freigabe Achskanal 4	OPMF3001 FB-60-X2.1
7	P-Kontakt	Freigabe Achskanal 5	OPMF3001 FB-60-X2.3
8	P-Kontakt	Freigabe Achskanal 6	OPMF3001 FB-60-X2.5
9	P-Kontakt	Freigabe Achskanal 7	OPMF3001 FB-60-X2.7
10	P-Kontakt	Freigabe Achskanal 8	OPMF3001 FB-60-X2.9
11		nicht belegt	
12		nicht belegt	
13		nicht belegt	
14	Schliesser	CNC-Betriebsbereit	MCU3000 FB-10-P5.2
15	Schliesser	Freigabe Achskanal 1	MCU3000 FB-10-P5.4
16	Schliesser	Freigabe Achskanal 2	MCU3000 FB-10-P5.6
17	Schliesser	Freigabe Achskanal 3	MCU3000 FB-10-P5.8
18			
19	Schliesser	Freigabe Achskanal 4	OPMF3001 FB-60-X2.2
20	Schliesser	Freigabe Achskanal 5	OPMF3001 FB-60-X2.4
21	Schliesser	Freigabe Achskanal 6	OPMF3001 FB-60-X2.6
22	Schliesser	Freigabe Achskanal 7	OPMF3001 FB-60-X2.8
23	Schliesser	Freigabe Achskanal 8	OPMF3001 FB-60-X2.10
24		nicht belegt	
25		nicht belegt	

4 SUB-D-Stecker, für CAN-Bus

4.1 SUB-D-Adapter von 10-poligem FB-Stecker P3 auf 9-poligen SUB-D-Stecker

Am Steckverbinder P3 werden optional die Anschlüsse für den CAN-Bus zur Verfügung gestellt. Mit diesem Adapterkabel wird der Stecker P3 auf einen 9-poligen Sub-D-Steckverbinder konvertiert. Dieser Steckverbinder ist an einem PC-Slotblech montiert. Durch die Konvertierung ergibt sich eine andere Steckerbelegung als an P3.

CAN ist ein 2-Draht-Bussystem, an dem alle Teilnehmer parallel angeschlossen werden. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 Ohm abgeschlossen werden, um Reflexionen zu vermeiden. Dies ist auch bei kurzen Leitungslängen erforderlich.

Auf der Steuerungsbaugruppe ist kein Leitungsabschluss vorhanden, dieser muss in jedem Fall vom Anwender extern realisiert werden.

Tabelle 4-1: Anschlussbelegung CAN-Bus (SUB-D-09)

Pin SUB-D	Name	Beschreibung / Anmerkungen	Quelle
1			MCU3000 FB-10-P3.1
2	CAN_LOW	CAN Datenleitung -	MCU3000 FB-10-P3.3
3	GND	Ground	MCU3000 FB-10-P3.5
4			MCU3000 FB-10-P3.7
5			MCU3000 FB-10-P3.9
6	GND	Ground	MCU3000 FB-10-P3.2
7	CAN_HIGH	CAN Datenleitung +	MCU3000 FB-10-P3.4
8			MCU3000 FB-10-P3.6
9			MCU3000 FB-10-P3.8

5 Anhang

5.1 Abbildungsverzeichnis

Abbildung 2-1: Interbus Configuration.....	36
Abbildung 2-2: Interbus SAP	38
Abbildung 2-3: Interbus Diagnostics.....	41

5.2 Tabellenverzeichnis

Tabelle 1-1: Bitkodierter Aufbau des digi-Wortes für die Achskanäle 4, 5 und 6.....	11
Tabelle 1-2: Bitkodierter Aufbau des digi-Wortes für die Achskanäle 7 und 8.....	12
Tabelle 1-3: Steckerbelegung X1 (SUB-D-Steckverbinder).....	16
Tabelle 1-4: Steckerbelegung X2 (SUB-D-Steckverbinder).....	18
Tabelle 1-5: Konfiguration der Inkrementalkoder für symmetrische und asymmetrische Betriebsart.....	26
Tabelle 2-1: Bedeutung der Symbole auf der Seite [Interbus][Configuration].....	37
Tabelle 2-2: Bedeutung der Symbole auf der Seite [Interbus][SAP].....	39
Tabelle 2-3: Bildschirmausgabe Interbus SAP.....	40
Tabelle 2-4: Bedeutung der Symbole auf der Seite [Interbus][Diagnostics]	41
Tabelle: 2-5 Interbus Status-Informationen und Kommandos.....	42
Tabelle: 2-6 Interbus Features	43
Tabelle 2-7: Logische Fehler	44
Tabelle 2-8: Port-Fehler.....	45
Tabelle 2-9: Logische Fehler in der seriellen Kommunikation	47
Tabelle 2-10: INTERBUS Fehler.....	48
Tabelle 2-11: Ergebnis-Byte (Result byte)	50
Tabelle 2-12: Anschlussbelegung der Fernbusschnittstelle (SUB-D).....	51
Tabelle 3-1: Anschlussbelegung Relais-Adapter (SUB-D-09)	52
Tabelle 3-2: Anschlussbelegung Relais-Adapter (SUB-D-25)	53
Tabelle 4-1: Anschlussbelegung CAN-Bus (SUB-D-09)	54

