POSITIONING AND CONTOURING CONTROL SYSTEM

MCU-3000 AND MCU-3100

OPTIONS MANUAL

Rev. 11/052015

1 OPMF options print

1.1 Brief description

The OPMF options print is used to extend the MCU-3000 / MCU-3100 positioning and contouring control system. Essentially, it comprises the additional logic for controlling up to five further servo or stepper motor axes. The device electronics are fully compatible with those of the MCU-3000 / MCU-3100 motherboard. In addition, the OPMF can also process up to 8 analog input signals with a 12-bit (MCU-3000) or 16-bit (MCU-3100) resolution. This section describes the technical software and hardware characteristics of the OPMF.

1.2 Software

The OPMF can be planned and programmed using the standard TOOLSET software for the MCU-3000 / MCU-3100 (as of software revision V2.5.0). The corresponding additional functions are described in the following sections, in addition to the description in the programming and referencing manual [PM] for the MCU-3000 / MCU-3100.

1.2.1 Modified rwmos.elf operating system software

Specific *rwmos.elf* operating system software is required in some cases so that the additional functions of the OPMF can be used. The information in the following section must be noted when rigging to a four (five, six, seven or eight) axis system. In this case, the additional functions for axis channels 4 – 8 are used in the same manner as for the standard three-axis system.
However, if only individual function modules are available on the OPMF (partial component mounting), these function modules are accessed with the help of the SAP programming method. A series of new system parameters are defined for this in *rw_SymPas.* The example programmes included in the scope of delivery represent the simple access to this access method.

1.2.2 PCAP programming

The next sections describe the individual signals and their connection pins. The corresponding pin assignments are listed in section 1.3.1.

1.2.2.1 rdAin, read analog inputs

DESCRIPTION:	This function provides the current analog value of the *channel* channel of the *an* axis. A maximum of 4 analog input channels are assigned to each axis. As up to eight channels can be processed, the analog input channels 1 – 4 are assigned to axis channel 1 and analog input channels 5-8 are assigned to axis channel 2.
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
RETURN VALUE:	Analog value as integer.
NOTE:	Up to 8 analog values can be processed. There are slight differences in handling the analog inputs of the different control systems, which has to be considered by the user where required, especially in case of a system changeover. MCU-3000: One value (one channel) is refreshed per scan cycle, i.e. with eight channels, each channel is refreshed every 8 ms, with a pre-defined cycle of 1 ms. Each channel can be set individually to the following measurement ranges: 0 .. +5 V, 0 .. 10 V, -5 V .. +5 V und –10 V .. 10 V. Valid indices for *an* are 0 and 1, and for *channel* the values 0..3 are valid. If due to the equipment version, only 4 analog input channels are available, these are the first 4 channels. The analog channels have a 12-bit resolution. MCU-3100: In every scan cycle, all values are refreshed. All channels can be set only together to the following measurement ranges: 0 .. +5 V, 0 .. 10 V, -5 V .. +5 V und –10 V .. 10 V. Valid indices for *an* are 0 and 1, and for *channel* the values 0..3 are valid. If due to the equipment version, only 4 analog input channels are available, these are the first 2 channels with the first axis and the first 2 channels with the second axis. The analog channels have a 16-bit resolution.

1.2.2.2 rddigi, read digital inputs

DESCRIPTION:	This function is used to request the following signal statuses: • The current status of the 24 OPMF digital inputs • The current status of the zero track (index) signal from the incremental encoder • A temporarily stored error in the measurement value acquisition system • A temporarily stored edge of the zero track (index) signal from the incremental encoder • A temporarily stored edge of the hardware latch signal (strobe). If an input is active, this is indicated with the value 1 for the respective bit. As an option, all digital inputs can be planned with inversion in the *mcfg.exe* TOOLSET programme. It is also possible to plan the required polarity when using an incremental encoder with an index signal. You must ensure that the digital inputs are grouped axis-specifically on the OPMF.
BORLAND DELPHI:	procedure rddigi(var tsrp:TSRP);
C:	void rddigi(struct TSRP far *tsrp);

VISUAL BASIC:	Sub rddigi(DTSRP As TSRP)
TSRP COMPONENTS:	TSRP[n].digi n = 3 .. Number of axes-1
RETURN VALUE:	The bit-coded return value is in the *digi* structure or record component and is constructed as described in the table below.
NOTE:	See note PM rddigi() Digital inputs 17 - 32 (connector X1) are assigned to axis channels 4, 5 and 6. Digital inputs 33 - 40 (connector X2) are assigned to axis channels 7 and 8.

a) Digi axis qualifier for axis channels 4, 5 and 6

The *digi* register is used to check the status of the digital inputs in the MCU-3000. If the respective inputs are active, this is displayed with the value 1 at the respective bit position.

Table 1-1: Bit-coded structure of the digi word for axis channels 4, 5 and 6

Bit No.	Function	X1/Pin
0	Input 17	9
1	Input 18	10
2	Input 19	11
3	Input 20	12
4	Input 21	13
5	Input 22	14
6	Input 23	15
7	Input 24	16
8	Input 25	42
9	Input 26	43
10	Input 27	44
11	Input 28	45
12	Input 29	46
13	Input 30 and hardware strobe signal for latching the actual position of axis channel 4	47
14	Input 31 and hardware strobe signal for latching the actual position of axis channel 5	48
15	Input 32 and hardware strobe signal for latching the actual position of axis channel 6	49
16	Zero track of the incremental encoder, axis-specific	--
17	Error of the measurement value acquisition system, axis-specific	--
18	Temporarily stored value for the zero track signal of the incremental encoder, axis-specific	--
19	Temporarily stored value of the latch signal (hardware strobe), axis-specific	--
20..31	Not assigned, these flags always have the value 0	--

b) Digi axis qualifier for axis channels 7 and 8

The *digi* register is used to check the status of the digital inputs in the MCU-3000. If the respective inputs are active, this is displayed with the value 1 at the respective bit position.

Table 1-2: Bit-coded structure of the digi word for axis channels 7 and 8

Bit No.	Function	X2 / Pin (SUB-D)
0	Input 33	9
1	Input 34	10
2	Input 35	11
3	Input 36	12
4	Input 37	13
5	Input 38	14
6	Input 39 and hardware strobe signal for latching the actual position of axis channel 7	15
7	Input 40 and hardware strobe signal for latching the actual position of axis channel 8	16
8..15	Not connected	
16	Zero track of the incremental encoder, axis-specific	--
17	Error of the measurement value acquisition system, axis-specific	--
18	Temporarily stored value for the zero track signal of the incremental encoder, axis-specific	--
19	Temporarily stored value of the latch signal (hardware strobe), axis-specific	--
20..31	Not assigned, these flags always have the value 0	--

1.2.2.3 rddigib, read digital input bit

DESCRIPTION:	This function is used to request the current status of one OPMF digital input and various other logic signals. The axis number must be specified in the *an* parameter (0, 1, … *MAXAXIS-1)* Note: Counting for the *bit no.* starts at 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
RETURN VALUE:	The function returns the value 1 or TRUE, provided the corresponding *bit no.* input is active.
NOTE:	See rddigi() section [1.2.2.2] and Note PHB rddigib() section [1.2.2.3]

1.2.2.4 rddigo, read digital outputs

See wrdigo in section [1.22.6]

1.2.2.5 rddigob, read digital output bit

See wrdigob in section [1.2.2.7]

1.2.2.6 wrdigo, write digital outputs

DESCRIPTION:

This register can be used to set the digital outputs of the OPMF. You must ensure that the digital outputs are grouped axis-specifically on the OPMF. If you wish to set an output, set the respective bit. The bit-coded structure of the *digo* status word can be taken from the following table:

Table: Bit-coded structure of the digo word for axis channels 4, 5 and 6

Bit No.	Function	Connector X1 / PIN
0	Output 9	26
1	Output 10	27
2	Output 11	28
3	Output 12	29
4	Output 13	30
5	Output 14	31
6	Output 15	32
7	Output 16	33
8..31	Not assigned	--

Table: Bit-coded structure of the digo word for axis channels 7 and 8

Bit No.	Function	Connector X2 / PIN (SUB-D)
0	Output 17	26
1	Output 18	27
2	Output 19	28
3	Output 20	29
4..31	Not assigned	--

BORLAND DELPHI:	procedure wrdigo(var tsrp:TSRP);
C:	void wrdigo(struct TSRP far *tsrp);
VISUAL BASIC:	Sub wrdigo(DTSRP As TSRP)
TSRP COMPONENTS:	TSRP[n].digo

1.2.2.7 wrdigob, write digital output bit

DESCRIPTION:	This function is used to set or reset one OPMF digital output. The axis number must be specified in the *an* parameter (3, … *MAXAXIS-1)* The output is reset with the value 0 or FALSE. Note: Counting for the *bit no.* starts at 1.
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)
NOTE	PCAP command *wrdigo()*

1.2.3 SAP programming

1.2.3.1 digi and digo axis qualifiers

For the functionality of the *digi* and *digo* axis qualifiers for PCAP commands rddigi() and rddigo(), see sections [1.2.2.2] and [1.2.2.4] respectively.

1.2.3.2 Axis qualifiers ain0 .. ain3

Axis qualifiers *ain0 to ain3* are used to read in the analog input channels. The axis qualifiers are assigned to axis channels 1 and 2. This means that up to 8 analog inputs can be processed. The input information is returned as *integer* type.

OPMF of the MCU-3000: The value range is specified as values from -2048 to +2047, based on the AD converter 12-bit word width, including signs. The voltage level for this measurement value depends on the input voltage range of the respective A/D channel. This is usually -5 to +5 V. However, measurement ranges from -10 to +10 V, 0 to 5 V or 0 to 10 V can also be processed. Each input channel can be set individually to one of the above-mentioned measurement ranges. If only 4 input channels are fitted, these are accessed under the 4 channels of the first axis. For this, inputs AIN1 to AIN4 are wired.

OPMF of the MCU-3000: The value range is specified as values from -32768 to +32767, based on the AD converter 16-bit word width, including signs. The voltage level for this measurement value depends on the input voltage range of the respective A/D channel. This is usually -10 to +10 V. However, measurement ranges from -5 to +5 V can also be processed. All input channels can be set only together to one of the above-mentioned measurement ranges. If only 4 input channels are fitted, these are accessed under the 2 channels each of the first two axes. For this, inputs AIN1, AIN2, AIN5 and AIN6 are wired.

1.3 Hardware

The OPMF options print is designed as a daughter board for the MCU-3000 / MCU-3100 motherboard. Both modules are assembled in the factory with plug and screwed connections. The control system requires one or two PC slots for mechanical reasons, depending on the configuration, but electrically only one PCI slot is required.

Depending on the level of configuration, the peripheral electronics are connected to the OPMF via a 50-pin SUB-D connector (X1) and a 60-pin SUB-D connector (X2). There is a cable loom with a cardholder and integrated 50-pin SUB-D connector for the 60-pin FB connector X2 of the OPMF. This cardholder can be installed in a free PC slot adjacent to the control system. In the maximum configuration (8 axes), up to three 50-pin SUB-D connectors can be used.

All the following pin descriptions refer to these 50-pin SUB-D connectors.

1.3.1 Options print OPMF connection assignments

1.3.1.1 Connector X1: 50-pin SUB-D pin connector

Table 1-3: Pin assignment X1, (SUB-D connector)

Pin	Name	Group
1	SERVO4 / PULSE4+	Setpoint value 4/stepper 4
2	AGND4 / PULSE4-	Setpoint value 4/stepper 4
3	CHA4+ / CLKSSI4+	Actual value 4
4	CHA4- / CLKSSI4-	Actual value 4
5	CHB4+ / DATSSI4+	Actual value 4
6	CHB4- / DATSSI4-	Actual value 4
7	NDX4+ / SIGN4+	Actual value 4/stepper 4
8	NDX4- / SIGN4-	Actual value 4/stepper 4
9	I17	Digital inputs 17-24 (24V)
10	I18	Assignment to axis channel 4, 5 and 6
11	I19	
12	I20	
13	I21	
14	I22	
15	I23	
16	I24	
17	+ 24V	Power supply for the digital outputs 24V, if digital outputs are being used, this voltage must be supplied externally. **Should be connected with MCU-3000 motherboard connector X1, pin 17.**
18	SERVO5 / PULSE5+	Setpoint value 5/stepper 5
19	AGND5 / PULSE5-	Setpoint value 5/stepper 5
20	CHA5+ / CLKSSI5+	Actual value 5
21	CHA5- / CLKSSI5-	Actual value 5
22	CHB5+ / DATSSI5+	Actual value 5
23	CHB5- / DATSSI5-	Actual value 5
24	NDX5+ / SIGN5+	Actual value 5/stepper 5
25	NDX5- / SIGN5-	Actual value 5/stepper 5
26	O9	Digital outputs 9.0.16 (24V)
27	O10	Assignment to axis channel 4, 5 and 6
28	O11	
29	O12	
30	O13	
31	O14	
32	O15	
33	O16	
34	SERVO6 / PULSE6+	Setpoint value 6/stepper 6
35	AGND6 / PULSE6-	Setpoint value 6/stepper 6
36	CHA6+ / CLKSSI6+	Actual value 6
37	CHA6- / CLKSSI6-	Actual value 6
38	CHB6+ / DATSSI6+	Actual value 6
39	CHB6- / DATSSI6-	Actual value 6
40	NDX6+ / SIGN6+	Actual value 6/stepper 6
41	NDX6- / SIGN6-	Actual value 6/stepper 6
42	I25	Digital inputs 25-32 (24V)
43	I26	Assignment to axis channel 4, 5 and 6
44	I27	
45	I28	
46	I29	
47	I30	
48	I31	Faster latch input axis channel 4
49	I32	Faster latch input axis channel 5
		Faster latch input axis channel 6

Pin	Name	Group
50	GND-D	Reference potential for all signal sources. These include digital inputs and outputs and the transmitter actual value. GND-D must be connected with the mass potential of the external device electronics. **Should be connected with MCU-3000 motherboard connector X1, pin 50.**

1.3.1.2 Connector X2 FB connector X2 and transformation to SUB-D connector (pin / socket)

Table 1-4: Pin assignment X2, (SUB-D connector)

Pin SUB-D	Name	Description/comments	Pin FB-60 - X2
1		Relay 1, NC contact	1
2		Relay 2, NC contact	3
3		Relay 3, NC contact	5
4		Relay 4, NC contact	7
5		Relay 5, NC contact	9
6		Relay 1, NO contact, release axis channel 4	2
7		Relay 2, NO contact, release axis channel 5	4
8		Relay 3, NO contact, release axis channel 6	6
9		Relay 4, NO contact, release axis channel 7	8
		Relay 5, NO contact, release axis channel 8	10
1	Servo7 / Pulse7+	Axis channel 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 inputs 33-40 (24V) Assignment to axis channels 7 and 8	35
10	I34		38
11	I35		41
12	I36		44
13	I37		47
14	I38		50
15	I39	Faster latch input axis channel 7	53
16	I40	Faster latch input axis channel 8	56
17	+24V	Power supply for digital outputs is internally connected with X1-17.	59
18	Servo8/ Pulse8+	Axis channel 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
26	O17	Digital outputs 17.20 (24V) Assignment to axis channels 7 and 8	37
27	O18		40
28	O19		43
29	O20		46
30	RGND1	Reference output group 1	49
31	AREF1		52
32	RGND2	Reference output group 2	55
33	AREF2		58

Pin SUB-D	Name	Description/comments	Pin FB-60 - X2
34	AIN1+	Analog inputs 1 to 4 Assignment to axis channel 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 inputs 5 to 8 Assignment to axis channel 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	Power supply for digital outputs is internally connected with X1-50.	60

The colour fields highlighted in yellow indicate the pin assignment on the 50-pin SUB-D connector. An FB8001 connection cable is required for these.

The colour fields highlighted in green indicate the pin assignment on the 9-pin SUB-D connector (see also section [1.3.1.8]). An FB-8001 connection cable is required for this.

The assignment of the OPMF of the MCU-3000 and the OPMF of the MCU-3100 is identical. With the analog inputs, the different resolutions should be noted.

1.3.1.3 Setpoint value channels

The OPMF system electronics supports operation of up to five more stepper or servo motor axes. The required motor system is planned and selected using the *mcfg.exe* TOOLST programme.

a) Setpoint value channel for servo motor axes

The analog output signal is used to control a power amplifier, which is activated as a speed or moment controller (current amplifier). In the factory, the offset of this setpoint value channel is stored in the non-volatile flash memory of the MCU-3000 and is taken into consideration by the software during output. The analog value output is only supported for *SERVO* planned axes.

- Pin assignment for connector X1, axis channel 4

Pin	Name	Group	Description
1	SERVO4	Setpoint value 4	Analog output signal 4 for controlling a power amplifier (+/-10V, 5mA). This signal is electrically isolated from the OPMF and has reference potential AGND4.
2	AGND4	Setpoint value 4	Reference potential for SERVO4. This potential is electrically isolated from the OPMF system electronics.

Note: Jumpers J1 and J2 must be bridged in position 2-3, so that the signals listed in the table are available at connector X1.

- Pin assignment for connector X1, axis channel 5

Pin	Name	Group	Description
18	SERVO5	Setpoint value 5	Analog output signal 5 for controlling a power amplifier (+/-10V, 5mA). This signal is electrically isolated from the OPMF and has reference potential AGND5.
19	AGND5	Setpoint value 5	Reference potential for SERVO5. This potential is electrically isolated from the OPMF system electronics.

Note: Jumpers J3 and J4 must be bridged in position 2-3, so that the signals listed in the table are available at connector X1.

- Pin assignment for connector X1, axis channel 6

Pin	Name	Group	Description
34	SERVO6	Setpoint value 6	Analog output signal 6 for controlling a power amplifier (+/-10V, 5mA). This signal is electrically isolated from the OPMF and has reference potential AGND6.
35	AGND6	Setpoint value 6	Reference potential for SERVO6. This potential is electrically isolated from the OPMF system electronics.

Note: Jumpers J5 and J6 must be bridged in position 2-3, so that the signals listed in the table are available at connector X1.

- Pin assignment for connector X2, axis channel 7

Pin (SUB-D)	Name	Group	Description
1	SERVO7	Setpoint value 7	Analog output signal 7 for controlling a power amplifier (+/-10V, 5mA). This signal is electrically isolated from the OPMF and has reference potential AGND7.
2	AGND7	Setpoint value 7	Reference potential for SERVO7. This potential is electrically isolated from the OPMF system electronics.

Note: Jumpers J7 and J8 must be bridged in position 2-3, so that the signals listed in the table are available at connector X2.

- Pin assignment for connector X2, axis channel 8

Pin (SUB-D)	Name	Group	Description
18	SERVO8	Setpoint value 8	Analog output signal 8 for controlling a power amplifier (+/-10V, 5mA). This signal is electrically isolated from the OPMF and has reference potential AGND8.
19	AGND8	Setpoint value 8	Reference potential for SERVO8. This potential is electrically isolated from the OPMF system electronics.

Note: Jumpers J9 and J10 must be bridged in position 2-3, so that the signals listed in the table are available at connector X2.

b) Setpoint value channel for stepper motor axes

There are four output signals to control a stepper motor power module. These comprise a pulse signal, a directional signal, and their inverted signals according to EIA standard RS422. All outputs deliver a typical output current of –60mA (max. –150mA). The maximum pulse frequency of the stepper signals is 10MHz.
Note: The positive edge of the PULSx+ stepper signal or the negative edge of the PULSx- stepper signal are crucial for the correct number of steps to be executed.

- Pin assignment for connector X1, axis channel 4

Pin	Name	Group	Description
1	PULSE4+	Stepper 4	Pulse signal
2	PULSE4-	Stepper 4	Inverted pulse signal
7	SIGN4+	Stepper 4	Directional signal
8	SIGN4-	Stepper 4	Inverted directional signal

Note: Jumpers J1 and J2 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X1.

- Pin assignment for connector X1, axis channel 5

Pin	Name	Group	Description
18	PULSE5+	Stepper 5	Pulse signal
19	PULSE5-	Stepper 5	Inverted pulse signal
24	SIGN5+	Stepper 5	Directional signal
25	SIGN5-	Stepper 5	Inverted directional signal

Note: Jumpers J3 and J4 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X1.

- Pin assignment for connector X1, axis channel 6

Pin	Name	Group	Description
34	PULSE6+	Stepper 6	Pulse signal
35	PULSE6-	Stepper 6	Inverted pulse signal
40	SIGN6+	Stepper 6	Directional signal
41	SIGN6-	Stepper 6	Inverted directional signal

Note: Jumpers J5 and J6 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X1.

- Pin assignment for connector X2, axis channel 7

Pin (SUB-D)	Name	Group	Description
1	PULSE7+	Stepper 7	Pulse signal
2	PULSE7-	Stepper 7	Inverted pulse signal
7	SIGN7+	Stepper 7	Directional signal
8	SIGN7-	Stepper 7	Inverted directional signal

Note: Jumpers J7 and J8 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X2.

- Pin assignment for connector X2, axis channel 8

Pin (SUB-D)	Name	Group	Description
18	PULSE8+	Stepper 8	Pulse signal
19	PULSE8-	Stepper 8	Inverted pulse signal
24	SIGN8+	Stepper 8	Directional signal
25	SIGN8-	Stepper 8	Inverted directional signal

Note: Jumpers J9 and J10 must be bridged in position 1-2, in order for the above-mentioned signals to be available at connector X2.

1.3.1.4 Pin assignment connector X1, digital inputs

The basic wiring diagrams for the digital inputs I17..I32 listed below are printed in [CM/section 5.2.7.1].

Pin	Name	Function
9	I17	Digital input 17
10	I18	Digital input 18
11	I19	Digital input 19
12	I20	Digital input 20
13	I21	Digital input 21
14	I22	Digital input 22
15	I23	Digital input 23
16	I24	Digital input 24
42	I25	Digital input 25
43	I26	Digital input 26
44	I27	Digital input 27
45	I28	Digital input 28
46	I29	Digital input 29
47	I30	Digital input 30 and faster hardware latch input to save the actual position of axis channel 4
48	I31	Digital input 31 and faster hardware latch input to save the actual position of axis channel 5
49	I32	Digital input 32 and faster hardware latch input to save the actual position of axis channel 6

1.3.1.5 Pin assignment connector X2, digital inputs

The basic wiring diagrams for the digital inputs I33..I40 listed below are printed in [CM/section 5.2.7.1].

Pin (SUB-D)	Name	Function
9	I33	Digital input 33
10	I34	Digital input 34
11	I35	Digital input 35
12	I36	Digital input 36
13	I37	Digital input 37
14	I38	Digital input 38
15	I39	Digital input 39 and faster hardware latch input to save the actual position of axis channel 7
16	I40	Digital input 40 and faster hardware latch input to save the actual position of axis channel 8

1.3.1.6 Pin assignment connector X1, digital outputs

The basic wiring diagrams for the digital outputs O9..O16 listed below are printed in [CM/section 5.2.8.1].

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

1.3.1.7 Pin assignment connector X2, digital outputs

The basic wiring diagrams for the digital outputs O17..O20 listed below are printed in [CM/section 5.2.8.1].

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

1.3.1.8 Pin assignment connector X2, release relay

At connector X2, relay points are provided for the amplifier releases. These are 'normally open' contacts. The relay is switched off after the PC is switched on, after a reset action or after an error occurs.

The release relay is activated for the respective selected axis channel by using the *cl()* command for PCAP and the *CL()* command for SAP.

Note: Depending on the configuration level of the OPMF, 1 to 5 relay outputs are available. In order to be able to use all relay outputs, these connections are made with the relay outputs of the MCU-3000 motherboard on a 25-pin SUB-D connector (see chapter 3.2).

The relay is a semi-conductor relay with a switching-on resistance of max. 25Ohm. The switching capability is 100mA, the switching voltage max. 60 V.

Pin (SUB-D)	Name	Function
1	R1-R	Relay S1, P contact, release of power amplifier axis channel 4
2	R2-R	Relay S2, P contact, release of power amplifier axis channel 5
3	R3-R	Relay S3, P contact, release of power amplifier axis channel 6
4	R4-R	Relay S4, P contact, release of power amplifier axis channel 7
5	R5-R	Relay S5, P contact, release of power amplifier axis channel 8
6	R1-S	Relay S1, NO contact, release of power amplifier axis channel 4
7	R2-S	Relay S2, NO contact, release of power amplifier axis channel 5
8	R3-S	Relay S3, NO contact, release of power amplifier axis channel 6
9	R4-S	Relay S4, NO contact, release of power amplifier axis channel 7
	R5-S	Relay S5, NO contact, release of power amplifier axis channel 8

1.3.1.9 Pulse acquisition channels

The OPMF is fitted with up to five pulse acquisition channels, to which various encoder types, such as linear scales or incremental or absolute encoders can be connected. Two 90° phase-shifted quadrature signals are processed as input signals, with a maximum pulse frequency of 2.0 MHz (optionally 5 MHz) and TTL sensors. A zero track (index signal) can also be evaluated. The signal levels acquired by the encoders are electronically quadrupled and processed internally as floating-point numbers with double accuracy. This means that the value range for the traverse path is virtually unrestricted.

a) Incremental encoders with inverted signals (symmetrical circuitry)

Incremental encoders with symmetrical outputs are particularly suitable for industrial use and are preferred, as the output signals are available with inverted and non-inverted signal sensors for all tracks. This enables pulses to be acquired reliably, even in environments that are subject to severe electromagnetic interference. The evaluation electronics on the OPMF are based on the RS422 standard and form a signal difference between the inverted and non-inverted input signal. Interference that is linked with transmission lines, can thus effectively be suppressed.
Important: In the factory, the OPMF is delivered for incremental encoders with symmetrical outputs, but can be configured by the user for asymmetrical encoders (see next table).

b) Incremental encoders without inverted signals (asymmetrical circuitry)

It is also possible to process incremental encoders without inverted pulse trains. However, these should only be used in environments that are not subject to severe electromagnetic interference, e.g. in laboratory applications. Please also ensure that the encoder cable is only a few metres long, especially for high pulse frequencies.

Table 1-5: Configuration of the incremental encoders for symmetrical and asymmetrical operating modes

Axis channel	Signal source	Solder jumper	Asymmetrical	Symmetrical
4	CHA4-	J15 (8001)	Bridged	Unbridged
		J15 (8008)	Do not connect pin 4 / X1	Connect pin 4 / X1
	CHB4-	J22 (8001)	Bridged	Unbridged
		J16 (8008)	Do not connect pin 6 / X1	Connect pin 6 / X1
	NDX4-	J23 (8001)	Bridged	Unbridged
		J17 (8008)	Do not connect pin 8 / X1	Connect pin 8 / X1
5	CHA5-	J14 (8001)	Bridged	Unbridged
		J19 (8008)	Do not connect pin 21 / X1	Connect pin 21 / X1
	CHB5-	J20 (8001)	Bridged	Unbridged
		J20 (8008)	Do not connect pin 23 / X1	Connect pin 23 / X1
	NDX5-	J21 (8001)	Bridged	Unbridged
		J21 (8008)	Do not connect pin 25 / X1	Connect pin 25 / X1
6	CHA6-	J13 (8001)	Bridged	Unbridged
		J23 (8008)	Do not connect pin 37 / X1	Connect pin 37 / X1
	CHB6-	J18 (8001)	Bridged	Unbridged
		J24 (8008)	Do not connect pin 39 / X1	Connect pin 39 / X1
	NDX6-	J19 (8001)	Bridged	Unbridged
		J25 (8008)	Do not connect pin 41 / X1	Connect pin 41 / X1
7	CHA7-	J16 (8001)	Bridged	Unbridged
		J27 (8008)	Do not connect pin 4 / X2	Connect pin 4 / X2
	CHB7-	J25 (8001)	Bridged	Unbridged
		J28 (8008)	Do not connect pin 6 / X2	Connect pin 6 / X2
	NDX7-	J24 (8001)	Bridged	Unbridged
		J29 (8008)	Do not connect pin 8 / X2	Connect pin 41 / X2

Axis channel	Signal source	Solder jumper	Asymmetrical	Symmetrical
8	CHA8-	J11 (8001)	Bridged	Unbridged
		J31 (8008)	Do not connect pin 21 / X2	Connect pin 21 / X2
	CHB8-	J12 (8001)	Bridged	Unbridged
		J32 (8008)	Do not connect pin 23 / X2	Connect pin 23 / X2
	NDX8-	J17 (8001)	Bridged	Unbridged
		J33 (8008)	Do not connect pin 25 / X2	Connect pin 25 / X2

Note: The solder jumpers listed in the table are located on the top left-hand corner of the solder side of the OPMF board.
The jumper identifiers in the table column „Solder jumper" are as follows: (8001) for the OPMF of the MCU-3000 and (8008) for the OPMF of the MCU-3100.

c) Optical decoupling of the pulse acquisition channels

All pulse acquisition channels of the OPMF are optically decoupled. This is advantageous particularly in an environment that is subject to severe electromagnetic interference.

d) Pin assignment for the pulse acquisition channels with incremental encoders

♦ Pin assignment X1, channel 4

Pin	Name	Function
3	CHA4+	Incremental signal (TTL square-wave pulse trains) track A
4	CHA4-	Inverted incremental signal track A
5	CHB4+	Incremental signal track B with 90° electrical phase shift to track A
6	CHB4-	Inverted incremental signal track B
7	NDX4+	Reference signal track 0
8	NDX4-	Inverted reference signal track 0

♦ Pin assignment X1, channel 5

Pin	Name	Function
20	CHA5+	Incremental signal (TTL square-wave pulse trains) track A
21	CHA5-	Inverted incremental signal track A
22	CHB5+	Incremental signal track B with 90° electrical phase shift to track A
23	CHB5-	Inverted incremental signal track B
24	NDX5+	Reference signal track 0
25	NDX5-	Inverted reference signal track 0

♦ Pin assignment X1, channel 6

Pin	Name	Function
36	CHA6+	Incremental signal (TTL square-wave pulse trains) track A
37	CHA6-	Inverted incremental signal track A
38	CHB6+	Incremental signal track B with 90° electrical phase shift to track A
39	CHB6-	Inverted incremental signal track B
40	NDX6+	Reference signal track 0
41	NDX6-	Inverted reference signal track 0

- Pin assignment X2, channel 7

Pin (SUB-D)	Name	Function
3	CHA7+	Incremental signal (TTL square-wave pulse trains) track A
4	CHA7-	Inverted incremental signal track A
5	CHB7+	Incremental signal track B with 90° electrical phase shift to track A
6	CHB7-	Inverted incremental signal track B
7	NDX7+	Reference signal track 0
8	NDX7-	Inverted reference signal track 0

- Pin assignment X2, channel 8

Pin (SUB-D)	Name	Function
20	CHA8+	Incremental signal (TTL square-wave pulse trains) track A
21	CHA8-	Inverted incremental signal track A
22	CHB8+	Incremental signal track B with 90° electrical phase shift to track A
23	CHB8-	Inverted incremental signal track B
24	NDX8+	Reference signal track 0
25	NDX8-	Inverted reference signal track 0

1.3.1.10 Pin assignment X2 for the analog input channels

The OPMF of the MCU-3000 is available with up to 8 analog inputs. The resolution of the measurement value channels is 12-bit including sign, and the measurement value range of 0…5 V, 0…10 V, -5 .. +5 V or -10 .. +10 V can be set individually for each channel. On the OPMF, the analog inputs are connected to the internal A/D module via differential amplifiers.

The OPMF of the MCU-3100 is available with up to 8 analog inputs. The resolution of the measurement value channels is 16-bit including sign, and the measurement value range of -5 .. +5 V or -10 .. +10 V can be set only for all channels together. On the OPMF, the analog inputs are connected to the internal A/D module via differential amplifiers.
Caution: On the OPMF module of the MCU-3100 Rev. A, with all analog inputs, the + and – connectors are inverted. In the succeeding versions Rev. B and higher, this error is corrected.

Pin (SUB-D)	Name	Assignment	Function
34	AIN1+	Axis	Analog input 1
35	AIN1-	channel 1	Reference potential for AIN1+
36	AIN2+		Analog input 2
37	AIN2-		Reference potential for AIN2+
38	AIN3+		Analog input 3
39	AIN3-		Reference potential for AIN3+
40	AIN4+		Analog input 4
41	AIN4-		Reference potential for AIN4+
42	AIN5+	Axis	Analog input 5
43	AIN5-	channel 2	Reference potential for AIN5+
44	AIN6+		Analog input 6
45	AIN6-		Reference potential for AIN6+
46	AIN7+		Analog input 7
47	AIN7-		Reference potential for AIN7+
48	AIN8+		Analog input 8
49	AIN8-		Reference potential for AIN8+

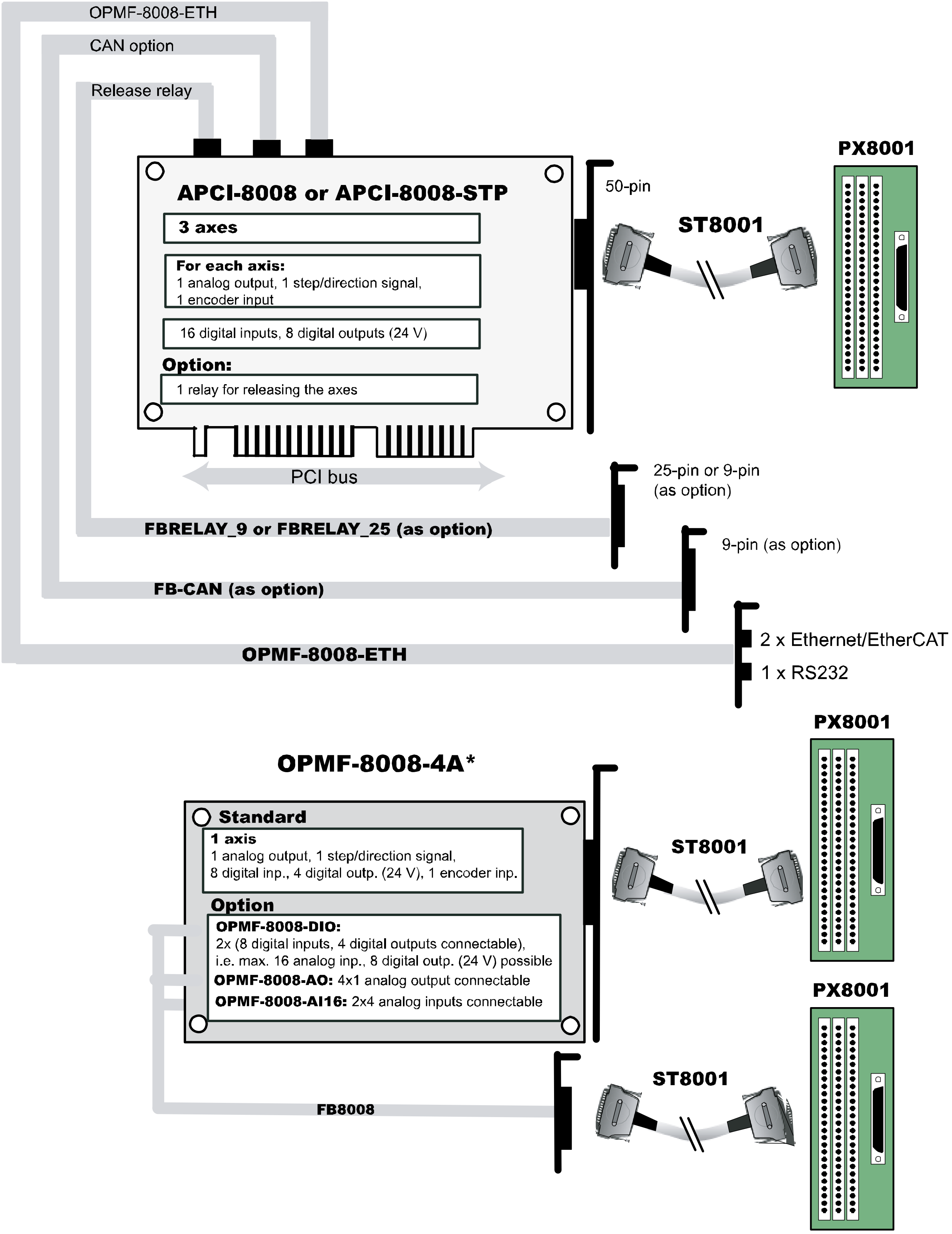
a) Pin assignment for reference voltage outputs

The OPMF of the MCU-3000 or MCU-3100 is available with up to 2 reference voltage outputs. These outputs are assigned to the above-mentioned AD groups. All outputs are set to 10 V. Each output can be loaded with 5 mA max. (if fewer than 7 axes: 10 mA).

Pin (SUB-D)	**Name**	**Assignment**	**Function**
30	RGND1	AIN1 ..	Reference potential for reference voltage output 1
31	AREF1	AIN4	Reference voltage output 1
32	RGND2	AIN5 ..	Reference potential for reference voltage output 2
33	AREF2	AIN8	Reference voltage output 2

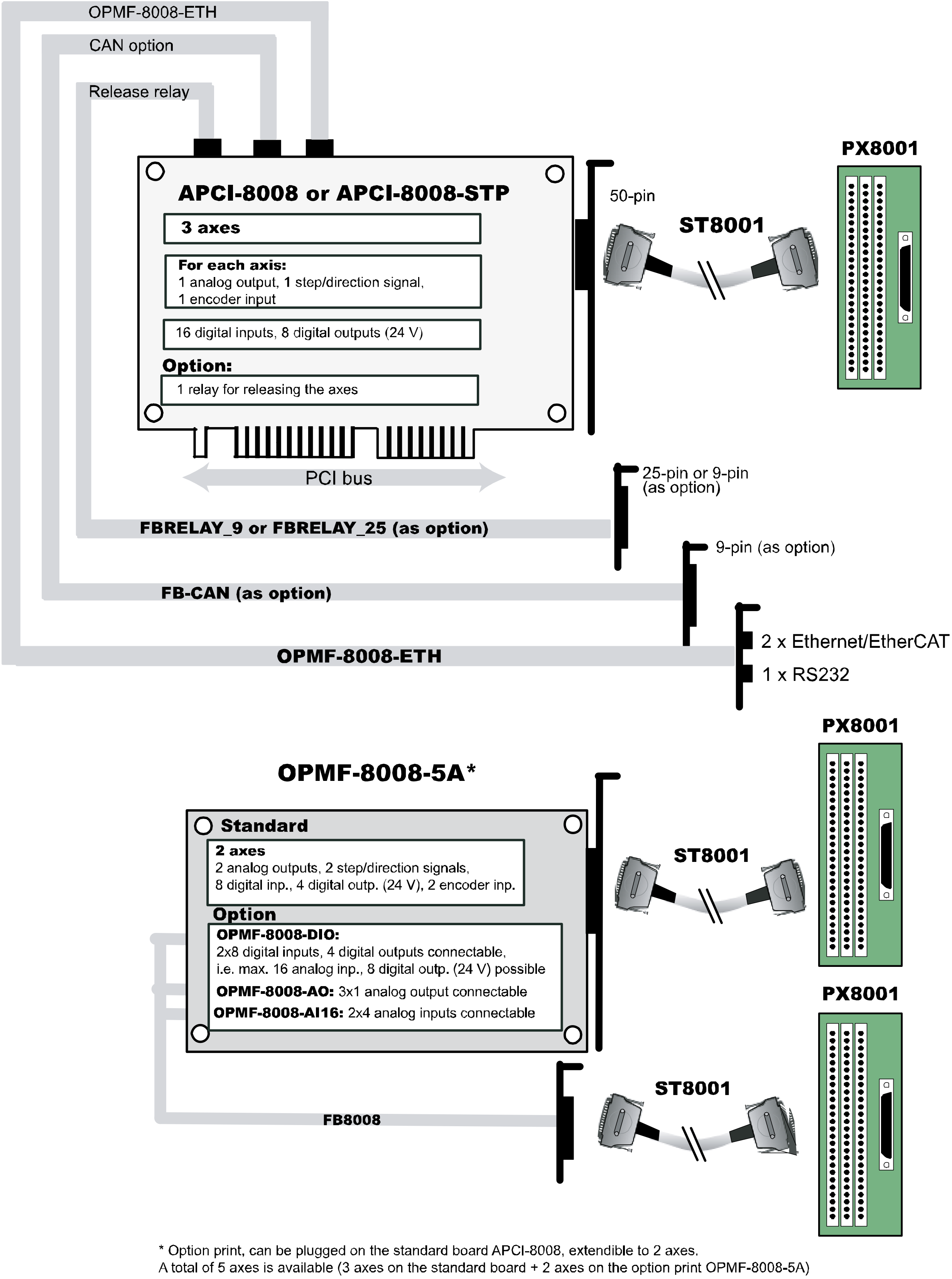
1.3.2 Connection of the OPMF options (MCU-3100)

Connection of the OPMF-8008-4A


OPMF-8008-ETH
CAN option
Release relay
APCI-8008 or APCI-8008-STP
3 axes
For each axis:
1 analog output, 1 step/direction signal,
1 encoder input
16 digital inputs, 8 digital outputs (24 V)
Option:
1 relay for releasing the axes
PCI bus
50-pin
ST8001
PX8001
25-pin or 9-pin
(as option)
FBRELAY_9 or FBRELAY_25 (as option)
9-pin (as option)
FB-CAN (as option)
2 x Ethernet/EtherCAT
1 x RS232
OPMF-8008-ETH
PX8001
OPMF-8008-4A*
Standard
1 axis
1 analog output, 1 step/direction signal,
8 digital inp., 4 digital outp. (24 V), 1 encoder inp.
Option
OPMF-8008-DIO:
2x (8 digital inputs, 4 digital outputs connectable),
i.e. max. 16 analog inp., 8 digital outp. (24 V) possible
OPMF-8008-AO: 4x1 analog output connectable
OPMF-8008-AI16: 2x4 analog inputs connectable
ST8001
PX8001
ST8001
FB8008

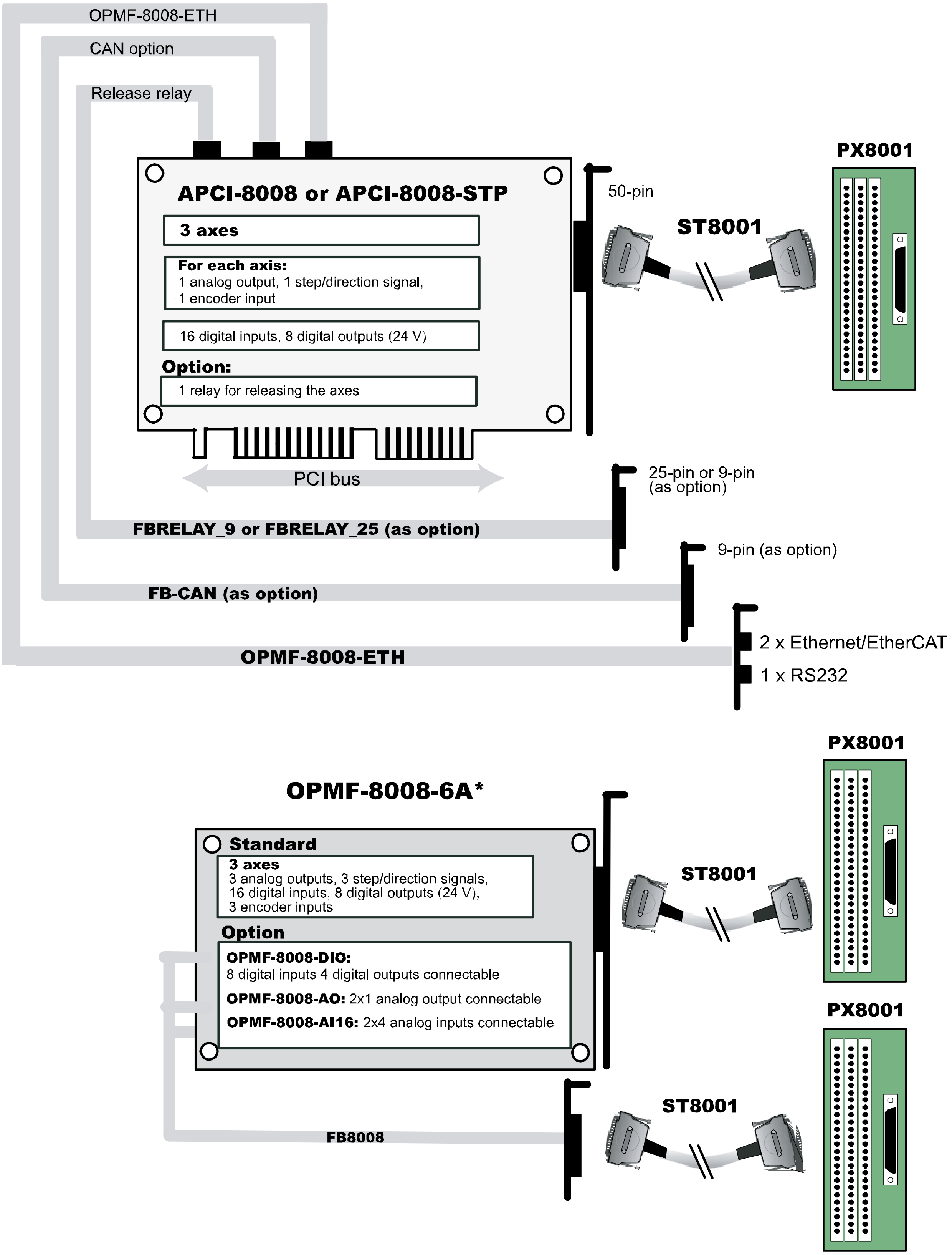
* Option print, can be plugged on the standard board APCI-8008, extendible to 1 axis.
A total of 4 axes is available (3 axes on the standard board + 1 axis on the option print OPMF-8008-4A)

Connection of the OPMF-8008-5A


OPMF-8008-ETH
CAN option
Release relay
APCI-8008 or APCI-8008-STP
3 axes
For each axis:
1 analog output, 1 step/direction signal,
1 encoder input
16 digital inputs, 8 digital outputs (24 V)
Option:
1 relay for releasing the axes
PCI bus
50-pin
ST8001
PX8001
25-pin or 9-pin
(as option)
FBRELAY_9 or FBRELAY_25 (as option)
9-pin (as option)
FB-CAN (as option)
2 x Ethernet/EtherCAT
1 x RS232
OPMF-8008-ETH
PX8001
OPMF-8008-5A*
Standard
2 axes
2 analog outputs, 2 step/direction signals,
8 digital inp., 4 digital outp. (24 V), 2 encoder inp.
Option
OPMF-8008-DIO:
2x8 digital inputs, 4 digital outputs connectable,
i.e. max. 16 analog inp., 8 digital outp. (24 V) possible
OPMF-8008-AO: 3x1 analog output connectable
OPMF-8008-AI16: 2x4 analog inputs connectable
ST8001
PX8001
ST8001
FB8008

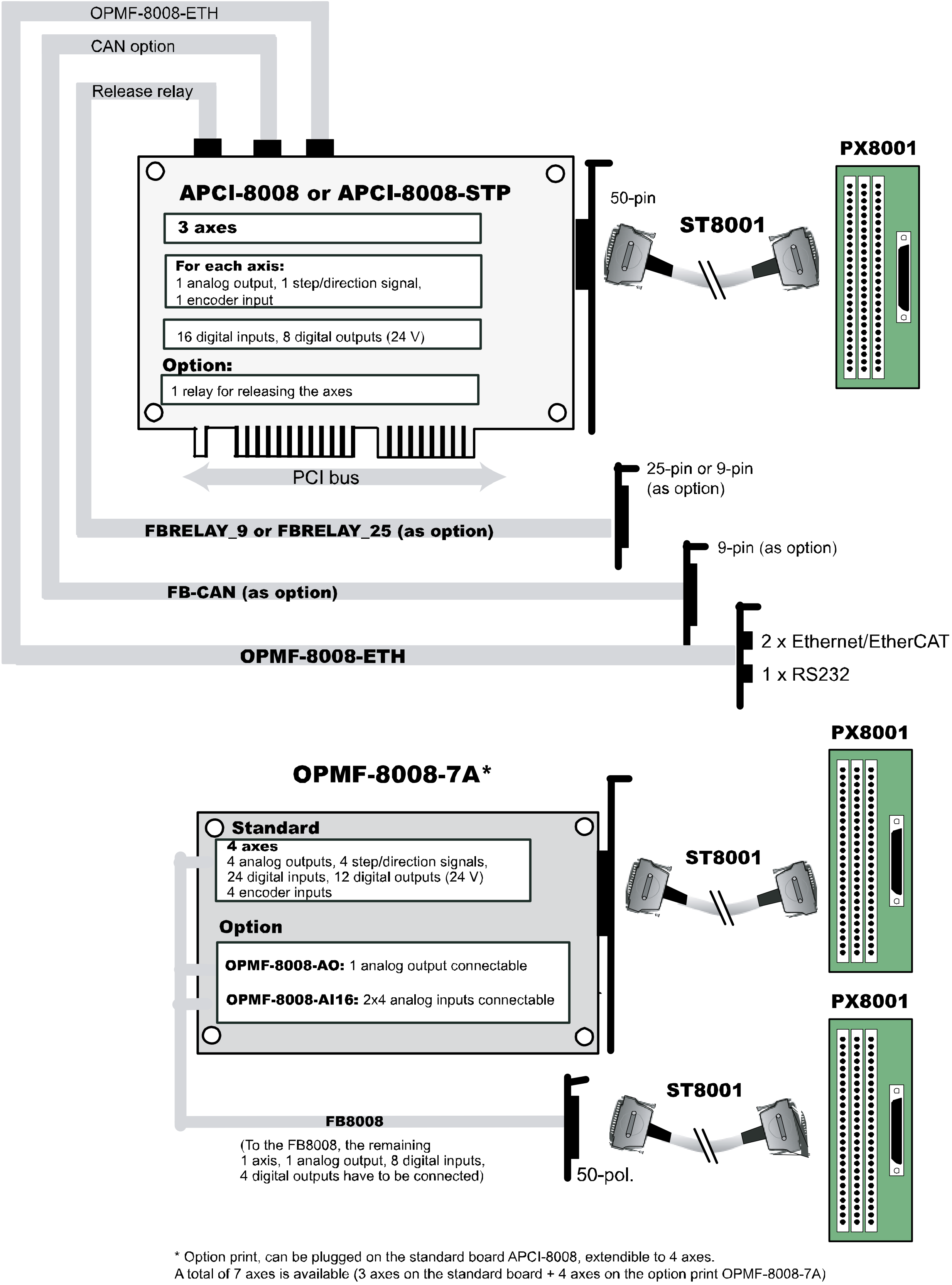
* Option print, can be plugged on the standard board APCI-8008, extendible to 2 axes.
A total of 5 axes is available (3 axes on the standard board + 2 axes on the option print OPMF-8008-5A)

Connection of the OPMF-8008-6A


OPMF-8008-ETH
CAN option
Release relay
APCI-8008 or APCI-8008-STP
3 axes
For each axis:
1 analog output, 1 step/direction signal,
1 encoder input
16 digital inputs, 8 digital outputs (24 V)
Option:
1 relay for releasing the axes
PCI bus
50-pin
ST8001
PX8001
25-pin or 9-pin
(as option)
FBRELAY_9 or FBRELAY_25 (as option)
9-pin (as option)
FB-CAN (as option)
2 x Ethernet/EtherCAT
1 x RS232
OPMF-8008-ETH
OPMF-8008-6A*
Standard
3 axes
3 analog outputs, 3 step/direction signals,
16 digital inputs, 8 digital outputs (24 V),
3 encoder inputs
Option
OPMF-8008-DIO:
8 digital inputs 4 digital outputs connectable
OPMF-8008-AO: 2x1 analog output connectable
OPMF-8008-AI16: 2x4 analog inputs connectable
ST8001
PX8001
ST8001
PX8001
FB8008

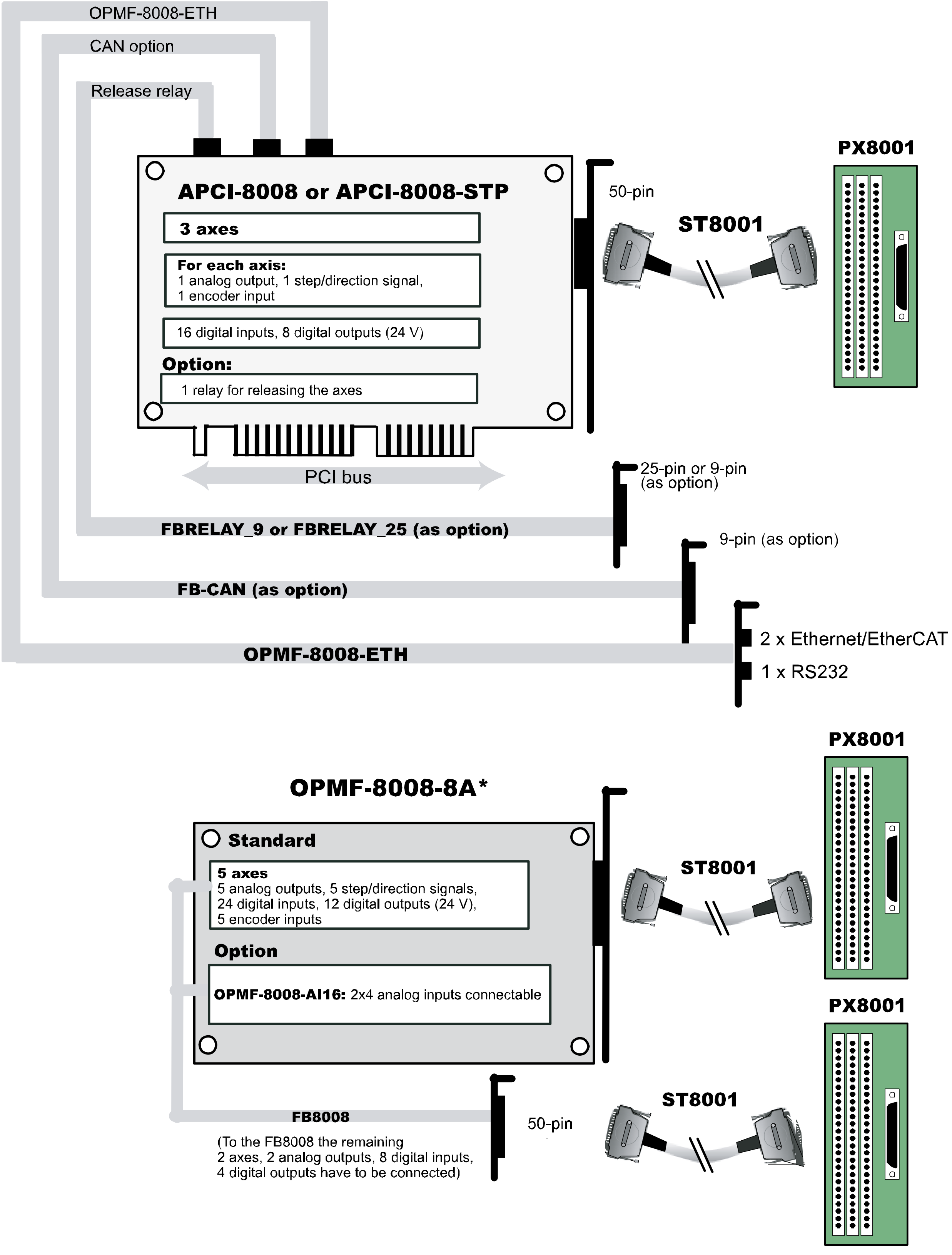
* Option print, can be plugged on the standard board APCI-8008, extendible to 3 axes.
A total of 6 axes is available (3 axes on the standard board + 3 axes on the option print OPMF-8008-6A)

Connection of the OPMF-8008-7A


OPMF-8008-ETH
CAN option
Release relay
APCI-8008 or APCI-8008-STP
3 axes
For each axis:
1 analog output, 1 step/direction signal,
1 encoder input
16 digital inputs, 8 digital outputs (24 V)
Option:
1 relay for releasing the axes
PCI bus
50-pin
ST8001
PX8001
25-pin or 9-pin
(as option)
FBRELAY_9 or FBRELAY_25 (as option)
9-pin (as option)
FB-CAN (as option)
2 x Ethernet/EtherCAT
1 x RS232
OPMF-8008-ETH
PX8001
OPMF-8008-7A*
Standard
4 axes
4 analog outputs, 4 step/direction signals,
24 digital inputs, 12 digital outputs (24 V)
4 encoder inputs
Option
OPMF-8008-AO: 1 analog output connectable
OPMF-8008-AI16: 2x4 analog inputs connectable
ST8001
PX8001
ST8001
FB8008
(To the FB8008, the remaining
1 axis, 1 analog output, 8 digital inputs,
4 digital outputs have to be connected)
50-pol.

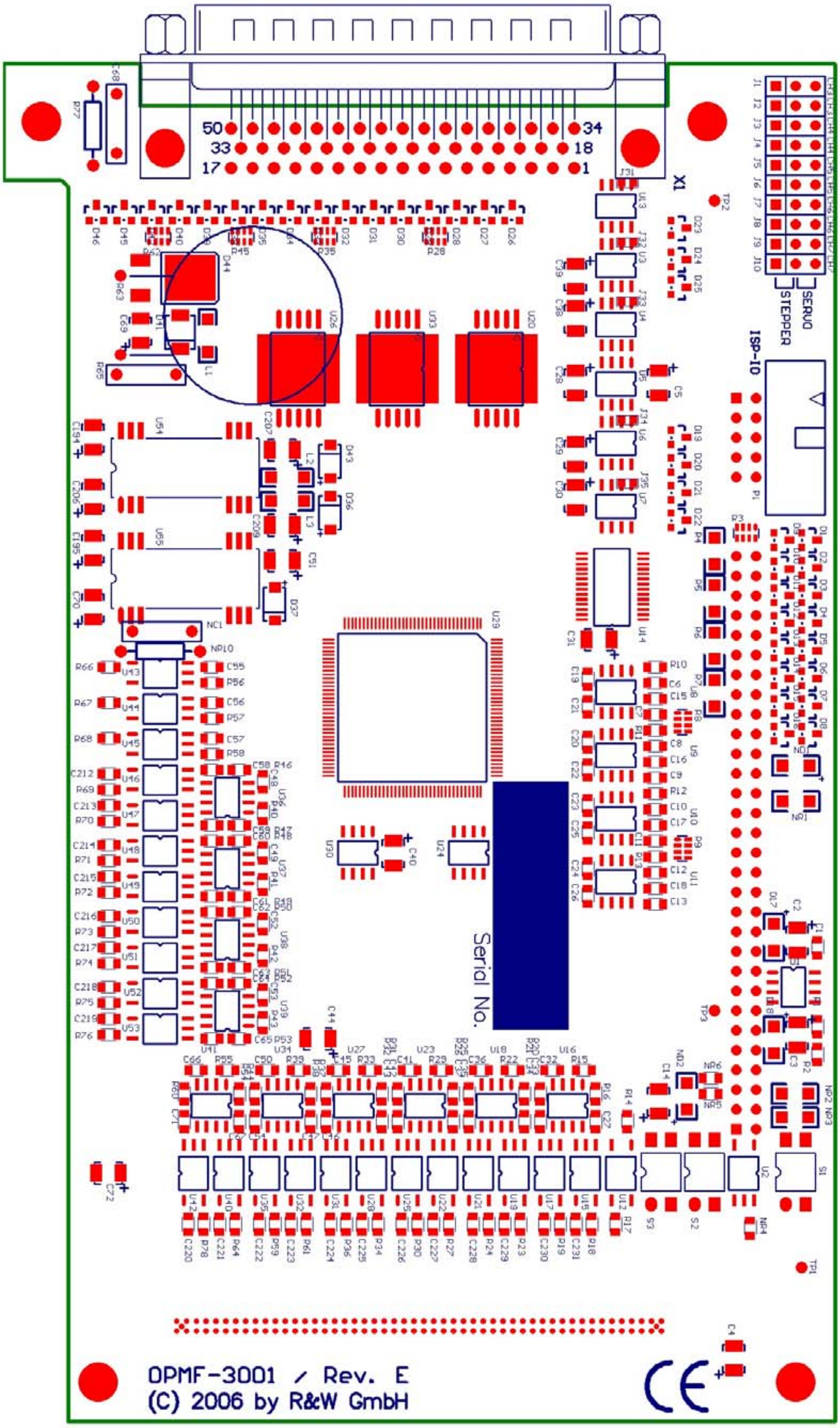
* Option print, can be plugged on the standard board APCI-8008, extendible to 4 axes.
A total of 7 axes is available (3 axes on the standard board + 4 axes on the option print OPMF-8008-7A)

Connection of the OPMF-8008-8A

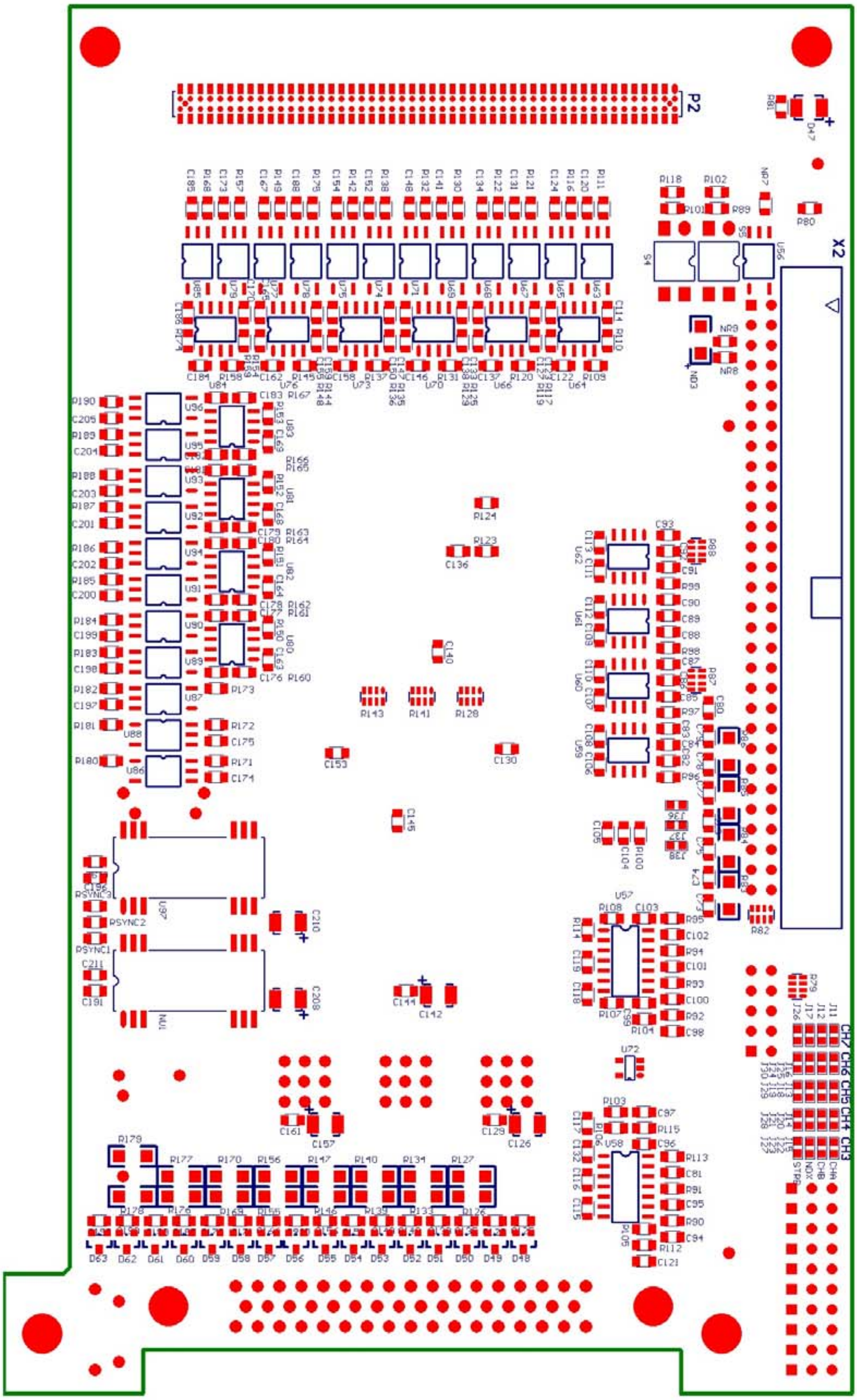

OPMF-8008-ETH
CAN option
Release relay
APCI-8008 or APCI-8008-STP
3 axes
For each axis:
1 analog output, 1 step/direction signal,
1 encoder input
16 digital inputs, 8 digital outputs (24 V)
Option:
1 relay for releasing the axes
PCI bus
50-pin
ST8001
PX8001
25-pin or 9-pin
(as option)
FBRELAY_9 or FBRELAY_25 (as option)
9-pin (as option)
FB-CAN (as option)
2 x Ethernet/EtherCAT
1 x RS232
OPMF-8008-ETH
PX8001
OPMF-8008-8A*
Standard
5 axes
5 analog outputs, 5 step/direction signals,
24 digital inputs, 12 digital outputs (24 V),
5 encoder inputs
Option
OPMF-8008-AI16: 2x4 analog inputs connectable
ST8001
PX8001
FB8008
(To the FB8008 the remaining
2 axes, 2 analog outputs, 8 digital inputs,
4 digital outputs have to be connected)
50-pin
ST8001

* Option print, can be plugged on the standard board APCI-8008, extendible to 5 axes.
A total of 8 axes is available (3 axes on the standard board + 5 axes on the option print OPMF-8008-8A)

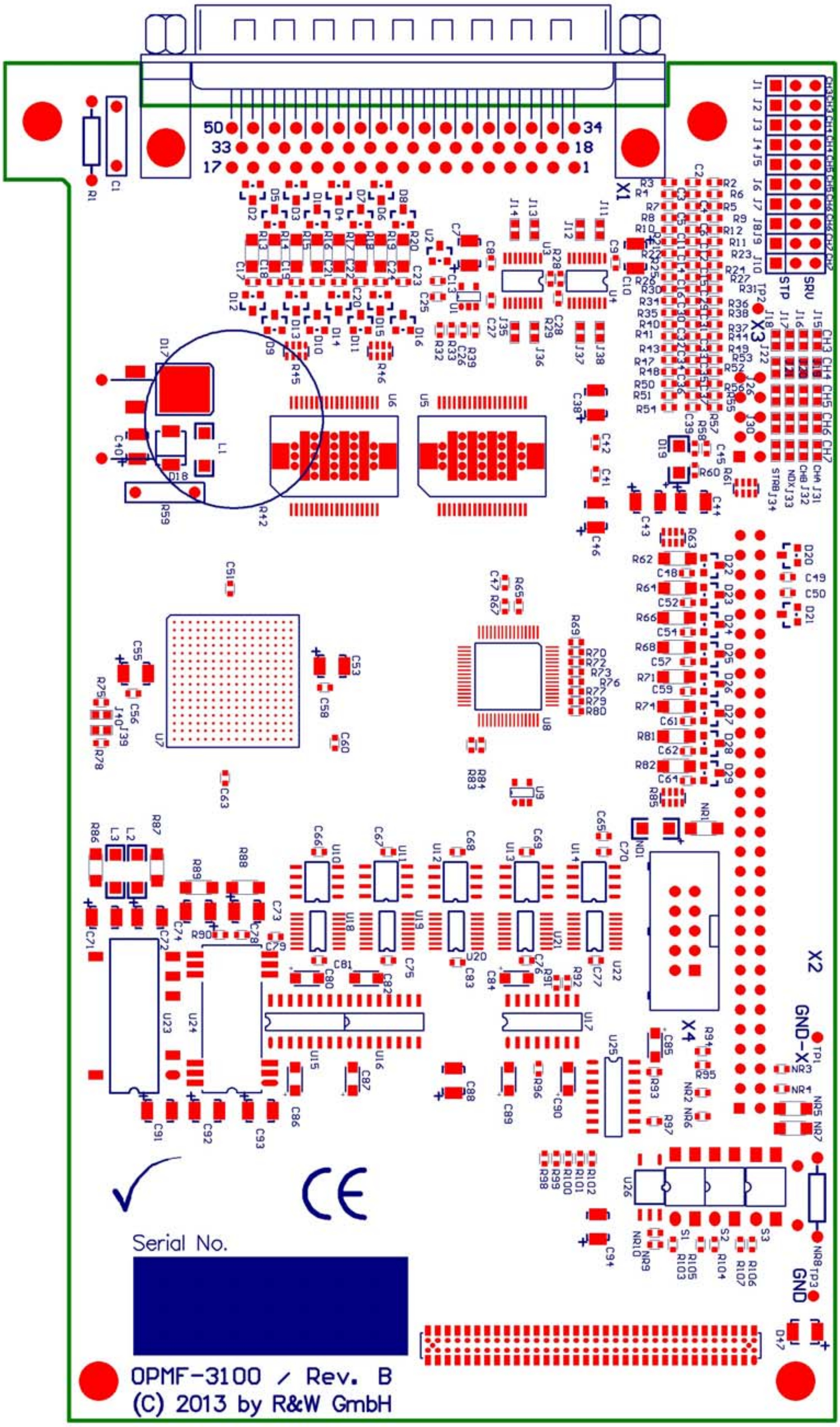
1.3.3 Component mounting diagram of the OPMF of the MCU-3000, top layer



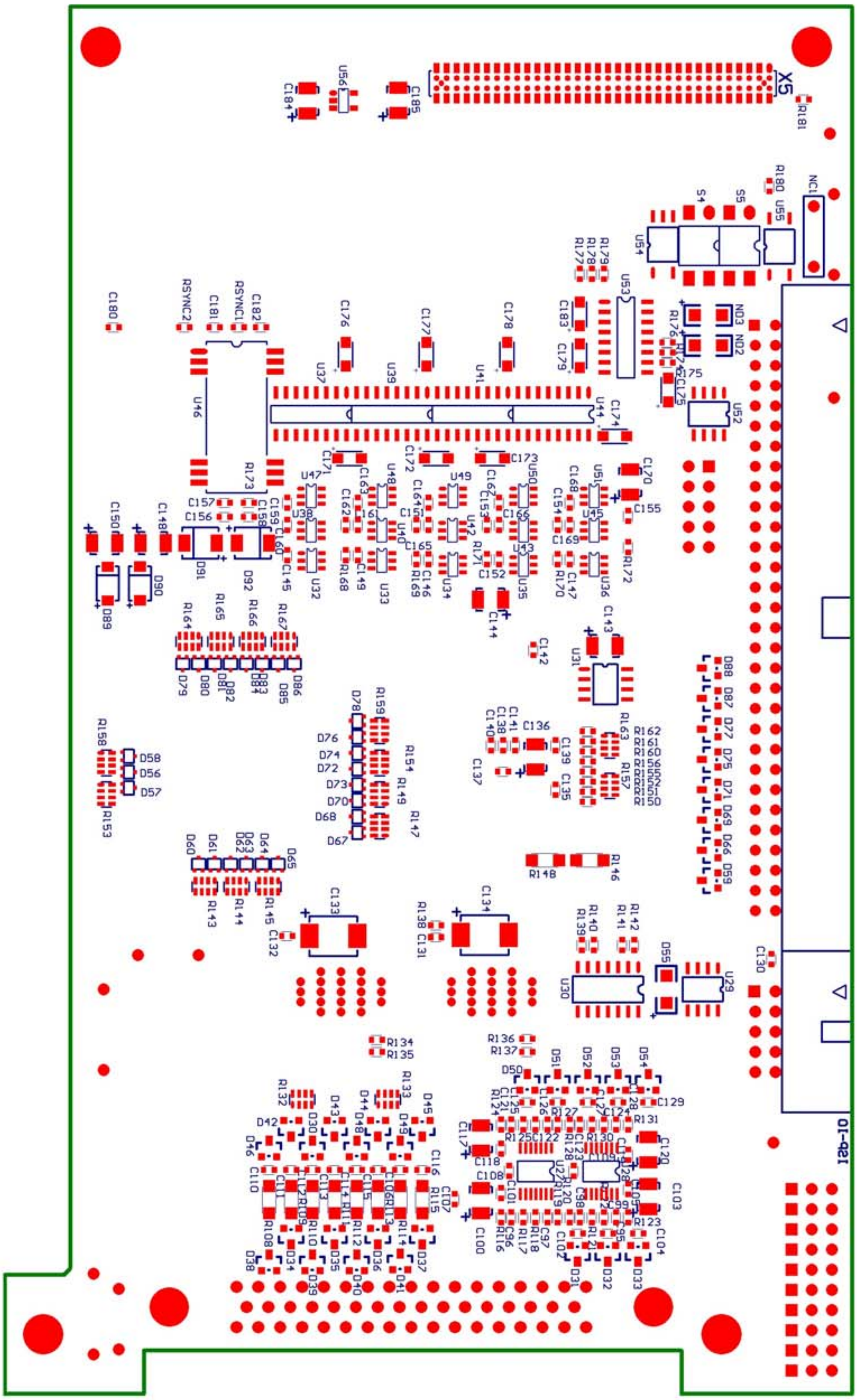
1.3.4 Component mounting diagram of the OPMF of the MCU-3000, bottom layer



1.3.5 Component mounting diagram of the OPMF of the MCU-3100, top layer



1.3.6 Component mounting diagram of the OPMF of the MCU-3100, bottom layer



1.3.7 OPMF technical data

Characteristic	Description/properties
Axis channels	1, 2, 3, 4 or 5. Mixed operation of servo or stepper motors possible. Up to 8 axes can be controlled using the MCU-3000 motherboard.
Encoder inputs	Directional discriminator for incremental encoders with 2 90° phase-shifted pulse tracks and zero pulse, or their inverted pulse tracks (6 channels) or SSI absolute encoder Pulse signal: 5V, TTL
Incremental encoder evaluation	x4, 32bit with sign, 2.0 MHz, optional 5 MHz (5 / 20 MHz after quadrupling)
SSI encoder evaluation	1..32bit, Gray/binary codes, variable frequency 30 kHz .. 10M Hz
Encoder supply	External auxiliary voltage depending on encoder type (5..30 V)
Setpoint value outputs for servo power output stages	16 bit DA converter potential-free (optically decoupled) • Output voltage +/- 10 V • Output current 5 mA
Setpoint value outputs for stepper motor output stages	RS422 pulse and directional signals and their inverted pulse trains, output current usually: -60mA (max. -150mA) pulse frequency: max. 10 MHz
Digital inputs	24 optically decoupled inputs (input current usually 8 mA at 24 V) • UHmax: 30 V / input current 12 mA usually • UHmin: 16 V / input current 2 mA usually • ULmax: 10 V / input current 0.3 mA usually • ULmin: 0 V / input current 0 mA usually Function mode of all digital inputs is freely programmable.
Digital outputs	12 optically decoupled outputs, output type: PNP 2 4V, 50 0mA (internal current limit at 1 A) Function mode of all outputs is freely programmable, setpoint state programmable after reset. Relay outputs: Max. 100 mA/60 V
Analog inputs	1..8 differential, potential-free (optically decoupled) • Resolution 12 bit • Input voltages: OPMF of the MCU-3000: -5..5V, -10..+10V, 0..+5V and 0..+10V, the input voltage level can be selected separately for each channel OPMF of the MCU-3100: -5..5V, -10..+10V, the input voltage level can be selected only for all channels together
External power supply	24V power consumption depending on load of the digital outputs
Design	Plug-in board 106mm * 175mm, x6 multi-layer
PC power supply	5V / approx. 0.5A in maximum configuration level
Connectors	Axis periphery • 50-pin SUB-D connector for additional axes 1-3 • 60-pin FB connector for additional axes 4-5 Release relay for additional axes 1-5 Analog inputs 1-8

2 OPIBS option

2.1 Brief description of the INTERBUS master interface

The OPIBS option is used to extend the MCU-3000 system with the aid of internationally standardised **INTERBUS** field bus technology. The MCU-3000 motherboard can be fitted with a special component mounting of few additional components for the Interbus field bus master. Thus, a cost-effective and flexible connection to additional process peripheries is possible. Serial bus technology presents significant advantages compared with the conventional parallel wiring, in particular for spatially distributed systems or a very large number of I/O points.

OPBIS is an INTERBUS master interface based on the IBS UART master protocol chips by Phoenix Contact. OPIBS is triggered via a “socket driver” integrated in the *rwmos.elf* operating programme, and provides the user with essential functions for controlling the INTERBUS and exchanging process data. In addition to the E/A data, the driver provides a basic system administration and basis diagnosis.

The scope of delivery includes the fwsetup.exe set-up and diagnosis programme, which is used to simplify the integration of a PCAP or SAP application programme created by the user.

At present, the *rwmos.elf* operating system software supports up to 512 Interbus subscribers and a maximum of 4096 I/O points. This means that the MCU-3000 can easily be fitted with new E/A functions. A large number of different field bus devices are available on the market, sometimes even with intelligent pre-processing functions.

2.2 Software

The OPIBS can be planned and programmed using the standard TOOLSET software for the MCU-3000 (as of revision V2.50j). The corresponding additional functions are described in the following sections, in addition to the description in the programming and referencing manual [PM].

2.2.1 Modified rwmos.elf operating system software

Specific *rwmos.elf* operating system software is required before the additional functions of the OPIBS can be used. This software is part of the scope of delivery of the OPIBS option and is not accounted for separately.
The Interbus option can be used with the aid of PCAP programming and SAP programming. The example programmes included in the scope of delivery represent the simple handling of these access methods.

2.2.2 Introduction

Below, a short overview is given of the methods with which the OPIBS option can be operated, in terms of software. To configure the specified INTERBUS structure as flexibly and quickly as possible, the following was undertaken:
The user initially defines a bus structure with all the necessary field bus devices. The current bus structure of the Interbus that is connected to the MCU-3000/OPIBS is automatically determined using the *fwsetup.*exe configuration programme included in the scope of delivery. The read configuration can be saved as an ASCII file or as a binary file for control and documentation purposes. It also includes further important information for SAP programming.

The binary file in which the setpoint state is saved can be used for a comparison with the actual bus configuration. In this case, it is ensured that a changed bus structure cannot be activated in the final application.

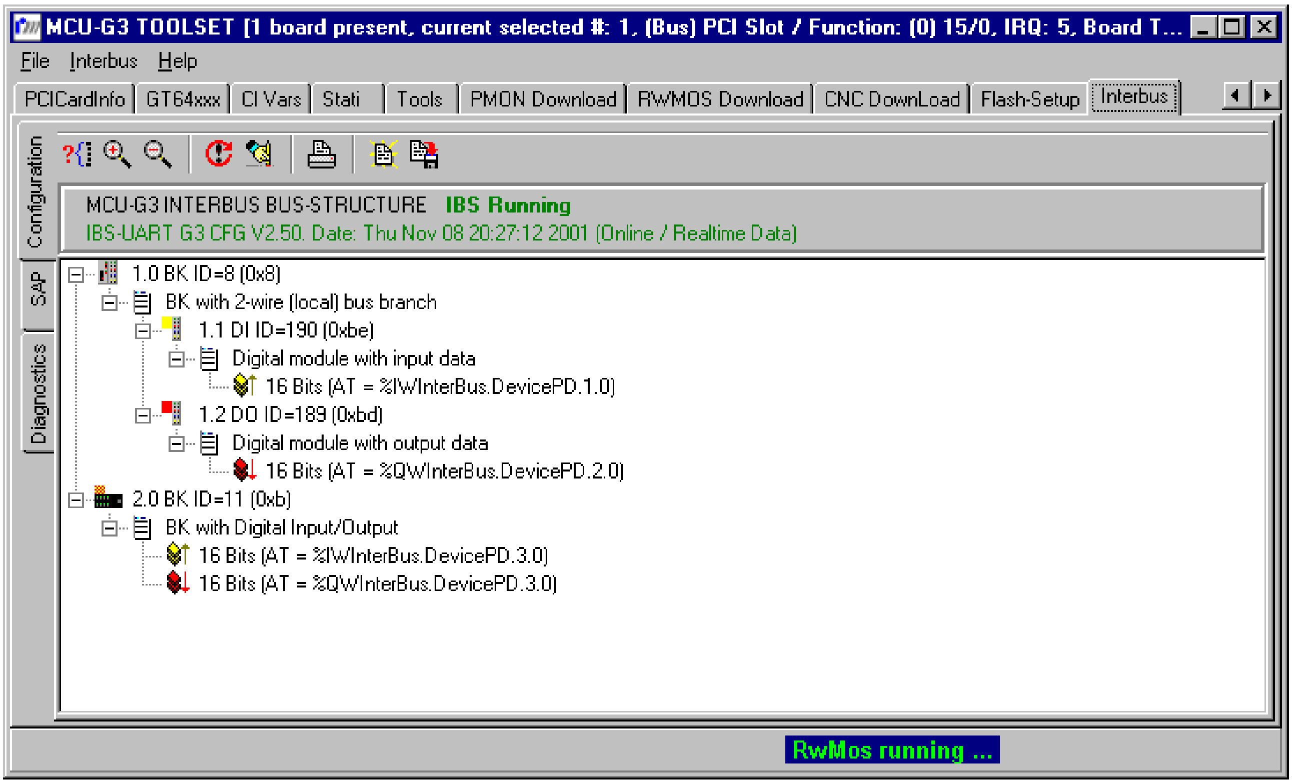
2.2.3 fwsetup.exe help programme

Using the *fwsetup.exe* TOOLSET programme, the bus structure that is connected to the MCU-3000 can be determined automatically, provided the connected bus is in a proper state. The additional planning options for the OPIBS can be set in the [Interbus] menu. At present, there are three different pages providing important information on the current bus structure and state. Various data contained in these are required for SAP or PCAP programming.

2.2.3.1 Interbus page: Configuration

On the “Configuration” page, the bus structure that is connected to the OPIBS can be displayed graphically in tree form, as shown in the screenshot below. In the top hierarchy level, the bus couplers (bus nodes) are listed with the device name and plain text description. The individual modules are displayed below this, with their specific properties, such as device name, description, access mode, and word width.

Figure 2-1: Interbus configuration



MCU-G3 INTERBUS BUS-STRUCTURE **IBS Running**
IBS-UART G3 CFG V2.50. Date: Thu Nov 08 20:27:12 2001 (Online / Realtime Data)

```
1.0 BK ID=8 (0x8)
    BK with 2-wire (local) bus branch
        1.1 DI ID=190 (0xbe)
            Digital module with input data
                16 Bits (AT = %IWInterBus.DevicePD.1.0)
        1.2 DO ID=189 (0xbd)
            Digital module with output data
                16 Bits (AT = %QWInterBus.DevicePD.2.0)
2.0 BK ID=11 (0xb)
    BK with Digital Input/Output
        16 Bits (AT = %IWInterBus.DevicePD.3.0)
        16 Bits (AT = %QWInterBus.DevicePD.3.0)
```

RwMos running ...

In order to activate this screen, the rwmos.elf operating system must already be launched on the board (online mode, RwMos running)

The following table explains the graphic operating symbols (speed buttons) from the screenshot above.

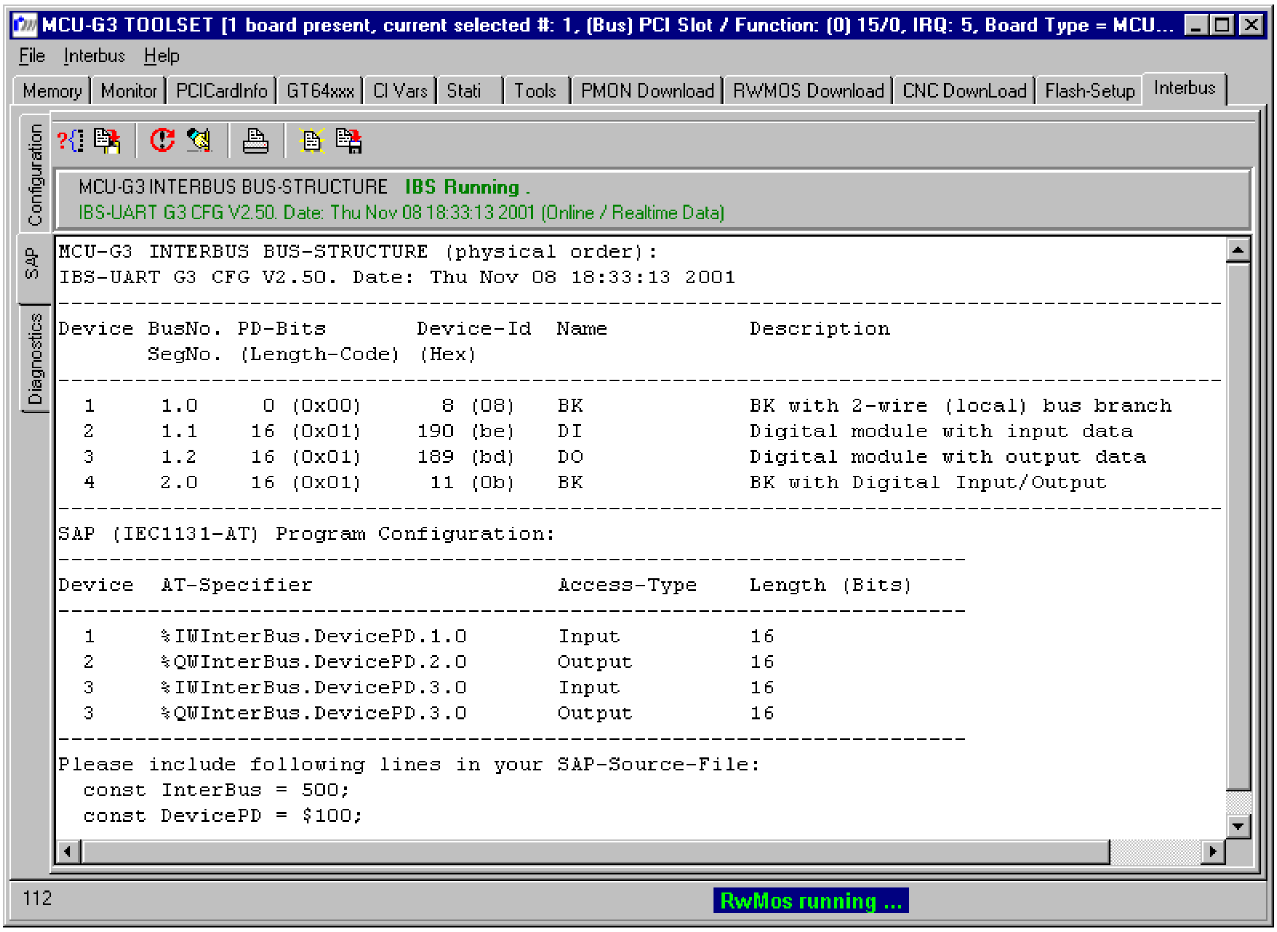
Table 2-1: Meaning of the symbols on the [Interbus][Configuration] page

Symbol	Meaning
❗	Interbus initialisation. All devices connected to the Interbus are automatically detected. The subscribers found are displayed on the screen in a graphical overview (as above). After successful initialisation, the Interbus is in run mode (IBS running…). Before executing the initialisation, saved error messages may be deleted.
?{	Read in the last detected Interbus configuration from the MCU-3000. The data that was detected during the last Interbus initialisation is analysed.
⊕	The current device directory tree is expanded. All available information is displayed.
⊖	The current device directory tree is collapsed. Only the bus nodes are displayed.
✗	If errors appeared, they can be deleted using this button.
🖶	The current configuration can be printed on a printer for documentation purposes.
💾	The current configuration can be saved in a binary file. The default file extension is “cfg”, which stands for “configuration data”. This data can be used for further processing at another workstation or for archiving or documentation purposes.
📄	A saved configuration can be opened and displayed graphically on the screen.

2.2.3.2 Interbus page: SAP

On the “SAP” page, the bus structure that is connected to the OPIBS can be displayed in text form, as shown in the screenshot below.

Figure 2-2: Interbus SAP



MCU-G3 TOOLSET [1 board present, current selected #: 1, (Bus) PCI Slot / Function: (0) 15/0, IRQ: 5, Board Type = MCU...

MCU-G3 INTERBUS BUS-STRUCTURE IBS Running .
IBS-UART G3 CFG V2.50. Date: Thu Nov 08 18:33:13 2001 (Online / Realtime Data)

```
MCU-G3 INTERBUS BUS-STRUCTURE (physical order):
IBS-UART G3 CFG V2.50. Date: Thu Nov 08 18:33:13 2001
-----------------------------------------------------------------------------------------------
Device BusNo. PD-Bits       Device-Id  Name           Description
       SegNo. (Length-Code) (Hex)
-----------------------------------------------------------------------------------------------
  1     1.0      0 (0x00)      8  (08)  BK             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             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 = $100;
```

112 RwMos running ...

In order to activate this screen, the rwmos.elf operating system must already be launched on the board (online mode, RwMos running).

a) Explanation of the graphical operating symbols

The following table explains the graphical symbols (speed buttons) from the screenshot above.

Table 2-2: Meaning of the symbols on the [Interbus][SAP] page

Symbol	Meaning
⟳	See Interbus configuration table
?{	See Interbus configuration table
🔍+	See Interbus configuration table
🔍−	See Interbus configuration table
✍	See Interbus configuration table
🖨	See Interbus configuration table
💾	See Interbus configuration table
📄	See Interbus configuration table
📋	Using this button, the AT specifiers introduced according to IEC1131, for use in the Windows clipboard, can be copied into an SAP programme for further use. The Windows clipboard contains the following information after you activate this speed button for the above-mentioned bus structure: %IWInterBus.DevicePD.1.0 %QWInterBus.DevicePD.2.0 %IWInterBus.DevicePD.3.0 %QWInterBus.DevicePD.3.0 These definitions can be used for variable declaration in an SAP programme.

b) Explanation of the SAP Interbus screen text output

Table 2-3: SAP Interbus screen output

```
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 the following lines in your SAP-Source-File:
  const InterBus = 500;
  const DevicePD = $200;
```

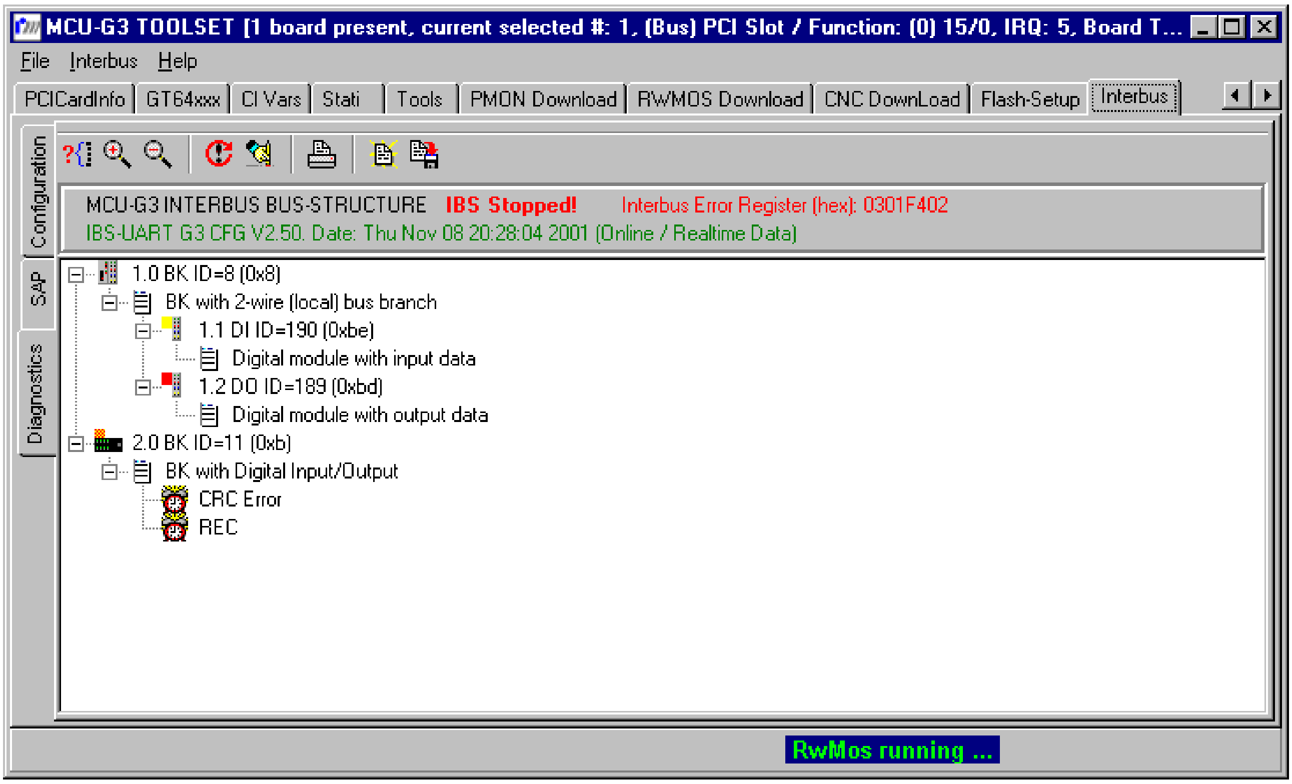
All available Interbus devices receive a consecutive number (*Device* column). This number corresponds to the actual position in the bus structure. The *BusNo.SegNo.* column specifies the bus segment to which the respective device belongs, and which consecutive number it has in this segment. The PD-Bits column (*LengthCode*) displays the address space required by the respective device in the host. The *Device-Id* column is used to describe the device function. Usually, the ID codes are expressed in decimal form on the Interbus module and in hexadecimal form for the RT modules. The *Name* and *Description* columns describe the respective module in plain text.
For SAP and PCAP programming, valuable information is also contained in this screen output. This includes all mode definitions, which are used to help activate the respective field bus device. The *AT specifier* column contains the mode definitions for declaring system variables for SAP programming and also includes the necessary information for object access using PCAP programming. In SAP programming language, access to the E/A level is regulated according to the IEC1131 programming language, using the keyword AT [G3 Universal Object Interface / section 2.2.]
The *Access type* column can be used to determine how the respective system variable can be accessed, i.e. read, write or both. The *Length* column shows how many bits can be used in the respective SAP variable.

Note: For the above-mentioned bus structure, there is an SAP demo programme in which the flexible and simple Interbus communication is demonstrated.

2.2.3.3 Interbus page: Diagnostics

On the “Diagnostics” page, quasi-statistic Interbus errors, as shown in the following screenshot, are displayed graphically in tree form.

Figure 2-3: Diagnostics Interbus



MCU-G3 TOOLSET [1 board present, current selected #: 1, (Bus) PCI Slot / Function: (0) 15/0, IRQ: 5, Board T...

MCU-G3 INTERBUS BUS-STRUCTURE IBS Stopped! Interbus Error Register (hex): 0301F402
IBS-UART G3 CFG V2.50. Date: Thu Nov 08 20:28:04 2001 (Online / Realtime Data)

```
1.0 BK ID=8 (0x8)
  BK with 2-wire (local) bus branch
    1.1 DI ID=190 (0xbe)
      Digital module with input data
    1.2 DO ID=189 (0xbd)
      Digital module with output data
2.0 BK ID=11 (0xb)
  BK with Digital Input/Output
    CRC Error
    REC
```

RwMos running ...

In order to activate this screen, the rwmos.elf operating system must already be launched on the board (online mode, RwMos running)

The following table explains the graphical symbols (speed buttons) from the screenshot above.

Table 2-4: Meaning of the symbols on the [Interbus][Diagnostics] page

Symbol	Meaning
!	See Interbus configuration table
?{	Read in the last saved quasi-statistical Interbus errors from the MCU-G3. The data that was detected when executing the last Interbus diagnosis cycle is analysed.
🔍+	See Interbus configuration table
🔍−	See Interbus configuration table
🔍	See Interbus configuration table
🖶	See Interbus configuration table
🖫	See Interbus configuration table
🗎	See Interbus configuration table

2.2.4 Interbus SAP programming

The additional software functions of the OPIBS Option can be used very effectively by the SAP programming method, in addition to the PCAP programming method.
Note: Accesses to the OPIBS Option should only be made from one task. Otherwise, it must be ensured that the OPIBS Option is only accessed sequentially (not simultaneously) from several tasks. If this is not ensured, unexplainable runtime behaviour may occur.

2.2.4.1 Interbus status information and commands

The Interbus functionality can be used by reading and writing pre-defined system variables. Using these variables, you can communicate directly with the “socket driver” that is integrated in the *rwmos.elf* operating programme. The definitions of the variables described below are in the *INTERBUS_AT_SPECS.INC* include file. As already mentioned above, the variable access is regulated in accordance with the IEC1131 programming language, with the aid of the AT mechanism. This is described in detail in [Universal Object Interface / section 2.2].

Table 2-5: Interbus status information and commands

System parameter	Value / Opcode	Description
IBSRelease Handles_w	1	Commands when all AT descriptors are discarded and must be strutured again.
IBSOpen_r	 0 1	Checks if the Interbus driver has already been opened. Interbus driver closed Interbus driver open
IBSOpen_w	1	Initialises the overall driver (memory, timer), executes a hardware reset for the IBS UART protocol chip and sets the required baud rate. This function must first be called at the start of each application. Otherwise it is not possible to produce a communication connection between the user programme and the OPIBS. Important: The command should only be executed if the driver has not yet been opened. This can be determined using *IBSOpen_r*. See also the error codes in [section 2.2.6].
IBSClose_r	 0 1	Checks if the Interbus driver has already been closed. Interbus driver open Interbus driver closed

System parameter	Value / Opcode	Description
IBSClose_w	1	Releases resources, such as memory and timer, that have already been initialised with IBSOpen_w. Furthermore, a hardware reset is performed on the IBS UART protocol chip. This function must be executed before exiting the programme. See also the error codes in [section 2.2.6].
IBSInit_r	 0 1	Checks if the Interbus driver has already been initialised. Interbus has not yet been initialised Interbus has already been initialised
IBSInit_w	1	Commissions the INTERBUS network and, if possible, provides the current executable configuration to the rwmos.elf operating programme. This function is therefore also used for error location. See also the error codes in [section 2.2.6]. **Note:** This command requires the systems for approx. 1 second. During this period PC and SAP accesses are switched OFF. To prevent error functions e.g. by starting a task in mcfg, a delay time can be set for this command (e.g. 100ms).
IBSRun_w	1	A secure data exchange is executed in this function. This includes a data cycle being triggered for each IBSRun_w write access. If an error is discovered during this cycle, the function is exited with the corresponding error message. Important: This function is only available for test purposes, and should not be called directly. Cyclical calling takes place automatically through the *rwmos.elf* operating system software. See also the error codes in [chapter 2.2.6].
IBSReset_w	1	Returns all INTERBUS subscribers to the reset state until the IBUInit_w function is called. See also the error codes in [chapter 2.2.6].

Note: All _w variables are write variables, and all _r variables are read variables.

2.2.4.2 Proceeding to use the Interbus

2.2.4.3 Interbus configuration parameters

The Interbus functionality can be used by reading and writing pre-defined system variables. Using these variables, you can communicate directly with the “socket driver” that is integrated in the *rwmos.elf* operating programme. The definitions of the variables described below are in the *INTERBUS_AT_SPECS.INC* include file. As already mentioned above, the variable access is regulated in accordance with the IEC1131 programming language, with the aid of the AT mechanism. This is described in detail in [Universal Object Interface / section 2.2].

Table 2-6: Interbus features

System parameter	Value / Opcode	Description
IBSMaxRep_r		
IBSMaxRep_w		
IBSMaxDevice_r	402 hex	Reads the maximum number of IBS devices
IBSMaxDevice_w	402 hex	Writes the maximum number of IBS devices
IBSMaxIoPoints_r		
IBSMaxIoPoints_w		
IBSBusyMode_r		
IBSBusyMode_w		
IBSExtraCycle_r		

System parameter	Value / Opcode	Description
IBSExtraCycle_w		
IBSStatusText_w		
IBSErrorReg_r		Reads Interbus error messages (please also see Chapter 2.2.6)
IBSErrorReg_w		

Note: The INTERBUS should be monitored in the SAP user programme, through status requests in an Interbus event handler routine, for example.

2.2.4.4 Interbus error handling

To handle Interbus error messages, the event handler EVIBS can be used. Before the event is activated, the variable IBSErrorReg_w should be reset. In the error routine, the content of the variable IBSErrorReg_r can then be analysed or displayed.

Note: The variable IBSErrorReg_r is only a 16-bit integer variable. When it is analysed as a 32-bit integer variable, the upper 16-bit are of no significance and therefore should be masked out.

2.2.5 Interbus PCAP programming

Accesses via PCAP programming are developed via the universal object interface.

2.2.6 Interbus error messages (error codes)

2.2.6.1 General

The errors are defined such that each error code is unique, but, if necessary, can be generated at different places in the socket driver of the rwmos.elf operating system software. All functions return a 32-bit value, the result. In the event of an error, this contains the driver-internal error location and a defined error code. This gives a simple evaluation and fast processing by the calling function. For analysis, only the error code in the last two bytes of the result word is relevant.

2.2.6.2 Logical errors (F0XX$_{hex}$)

Table 2-7: Logical errors

Code	Meaning	Explanation
F001$_{hex}$	IBU_COMNo_is_not_open	The specified COM interface is not open. Please open it using the IBU_Open() function
F002$_{hex}$	IBU_comno_already_open	This interface is already open and is being used
F003$_{hex}$	IBU_comno_not_supported	This interface is not supported
F004$_{hex}$	IBU_Function_not_active	Internal error. The status of a non-active function has been requested.
F005$_{hex}$	IBU_comresource_already_in_use	The corresponding COM interface is not free.
F506$_{hex}$	IBU_no_timer	Timer not available

Code	Meaning	Explanation
F507$_{hex}$	IBU_memory_not released	When executing the IBU_Close() function, the resources (memory or IRQ) could not be released again.
F508$_{hex}$	IBU_timer_not released	When executing the IBU_Close() function, the resources (timer) could not be released again.
F009$_{hex}$	IBU_no_of_recchar_not_supported	Too many characters received (more than 512) (maximum value exceeded)
F00A$_{hex}$	IBU_no_of_sendchar_not_supported	Too many characters sent (more than 512) (maximum value exceeded)
F00B$_{hex}$	IBU_no_setup_started	The connected INTERBUS system was not read in with IBS_Init()
F00C$_{hex}$	IBU_Parameter_invalid_range	Value outside the range
F00D$_{hex}$	IBU_Parameter_not_supported	Parameter is not supported by the hardware
F00E$_{hex}$	IBU_api_required_crc_error	The API layer of the driver has requested an invalid cycle

2.2.6.3 Port errors (F1XX$_{hex}$)

Table 2-8: Port errors

Code	Meaning	Explanation
F101$_{hex}$	IBU_baudrate_not_supported	The specified baud rate is not supported by the host system.
F102$_{hex}$	IBU_parity_not_supported	Even parity is not supported by the host system.
F103$_{hex}$	IBU_stopbits_not_supported	The number of stopbits (2) is not supported by the host system.
F104$_{hex}$	IBU_databits_not_supported	The number of databits (8) is not supported by the host system.
F105$_{hex}$	IBU_frame_error (Host receive)	Transfer error on the serial interface (frame error) (IBS UART ➔ Host system)
F106$_{hex}$	IBU_host_parity_error	Transfer error on the serial interface (parity error) (IBS UART ➔ Host system)
F107$_{hex}$	IBU_ibs_uart_parity_error	Transfer error on the serial interface (Host system ➔ IBS UART)
F108$_{hex}$	IBU_send_error	Error when sending to IBS UART
F109$_{hex}$	IBU_receive_error	Error when receiving from IBS UART
F110$_{hex}$	IBU_timer_is_already_running	Internal error. A started timer cannot be started again.
F111$_{hex}$	IBU_timervalue_not_supported	Internal error. Timer value is not supported by the host system.
F112$_{hex}$	IBU_timer_is_stopped	Internal error. A stopped timer cannot be stopped again.

Code	Meaning	Explanation
F113$_{hex}$	IBU_timer_timeout	Internal error. Timer has expired.
F120$_{hex}$	IBU_serial_timeout	No serial connection to INTERBUS UART. Possible causes: • No voltage to IBS UART • No DTR signal • The transfer speeds set on the IBS UART and the host system do not coincide.
F121$_{hex}$	IBU_overrun_error	Overrun at the serial interface detected by the host system
F122$_{hex}$	IBU_break_error	Interruption to data transfer at the serial interface detected by the host system

2.2.6.4 Logical errors in serial communication (F2XX$_{hex}$)

Table 2-9: Logical errors in serial communication

Code	Meaning	Explanation
F201$_{hex}$	IBU_too_many_char_received	More characters received than expected
F202$_{hex}$	IBU_delay_error	The reliable INTERBUS delay (bus delay) has been exceeded; see (IBU_ML_Set_Bus_Delay ())
F203$_{hex}$	IBU_send_finished_and_rec_busy	Fewer characters received than expected
F204$_{hex}$	IBU_send_busy_and_rec_busy	Not all characters could be sent or received
F205$_{hex}$	IBU_send_busy_and_rec_finished	Not all characters could be sent
F206$_{hex}$	IBU_Wrong_State_Key	The driver and hardware are no longer synchronous; **Example:** Confirmation on a service primitive was incorrectly received by the driver.
F207$_{hex}$	IBU_Wrong_State_Type	The driver and hardware are no longer synchronous; **Example:** Confirmation on a service primitive was incorrectly received by the driver.
F208$_{hex}$	IBU_primitive_Error	An incorrect confirmation was received from the driver for a request.
F209$_{hex}$	IBU_service_Error	The driver and hardware are no longer synchronous; **Example:** A reset command sent by the application (driver) does not reach the UART chip, because the connection was interrupted, for example.

2.2.6.5 INTERBUS errors (F3XX $_{hex}$, F4XX $_{hex}$)

Table 2-10: INTERBUS errors

Code	Meaning	Explanation
F3Xx$_{hex}$	IBS_ibs_result_error	For the meaning of the result byte (“Xx$_{hex}$”) see Table 2-11.
F400$_{hex}$	IBU_ibs_timeout	Interruption of an INTERBUS cable or defective INTERBUS device.
F401$_{hex}$	IBU_lbw_error	Loop-back word (LBW) error, The transmitted LBW could not be correctly received again.
F402$_{hex}$	IBU_crc_error	Temporary error on the INTERBUS; Possible cause: EMV effect
F403$_{hex}$	IBU_no_Devices_are_connected	No field devices are connected to the INTERBUS master.
F404$_{hex}$	IBU_no_valid_input_data	No valid input data
F410$_{hex}$	IBU_ibs_timeout_init_ri	Interruption of a cable or defective INTERBUS device to the continuing interface of the specified subscriber See function: IBU_IAPI_Set_Max_Devices()
F411$_{hex}$	IBU_ibs_timeout_init_bi	Interruption of a cable or defective INTERBUS device to the branch interface of the specified subscriber See function: IBU_IAPI_Set_Max_Devices()
F412$_{hex}$	IBU_data_register_error	Data register error; The physical data register length does not coincide with the logical data register length of the specified subscriber.
F413$_{hex}$	IBU_too_many_devices _conf_not_valid	The number of connected INTERBUS devices exceeds the maximum limit set or 512 See function: IBU_IAPI_Set_Max_Devices()
F414$_{hex}$	IBU_init_too_many_io_points	The number of connected INTERBUS I/O points exceeds the maximum limit of 4096 inputs and 4096 outputs subscribers
F415$_{hex}$	IBU_too_many_errors	Too many errors occurred during the execution of the IBU_IBS_Init() function Possible causes: • **max_repeat** too small ➔IBU_IAPI_Set_MaxRep() • **BusDelay** too small ➔IBU_ML_Set_Bus_Delay ()

Code	Meaning	Explanation
F416hex	IBU_init_timeout	IBU_IBS_Init() could not be completed within the specified time Possible causes: • **max_repeat** too small ➔IBU_IAPI_Set_MaxRep() • **BusDelay** too small ➔IBU_ML_Set_Bus_Delay ()
F417hex	IBU_too_many_devices_config_valid	The previously set maximum number of subscribers is valid.
F420hex	IBU_ibs_loop_diag_no_loop	No system with the “loop diagnosis” function is connected
F421hex	IBU_ibs_loop_diag_not_stable	No stable state is achieved for subscriber-oriented local bus diagnosis. Possible cause: loose connection

Table 2-11: Result byte

Coding (1 Byte)	Meaning	Explanation
00*hex*	No error	System runs without errors.
01*hex*	CRC error last path	Received and calculated CR check sums do not coincide. A fault has occurred on the section monitored by IBS UART.
02*hex*	CR error	CR flag in the character header of the sent and received characters do not coincide.
04*hex*	SL error	SL flag in the character header of the sent and received characters do not coincide.
08*hex*	NOISE error	After the expected end of an FCS, a “follow-on telegram” was detected by the IBS UART hardware.
10*hex*	STOP error	STOP bit error. Instead of the expected “STOP bit” of an INTERBUS telegram, a high signal was identified by the IBS UART hardware.
20*hex*	MAU error	A static “high” signal with more than 64 bit times length was identified at the INTERBUS input (connection DI). This error occurs if no device is connected to the IBS UART.
40*hex*	Parity error	A transmission error was detected by the IBS UART hardware via the parity bit of the serial asynchronous communication.
80*hex*	Service error	- Primitive not defined - When sending the result byte, a further error was identified - Primitive not allowed in the current status if, for example the IBS UART is still in the “reset” status only the “ResetStop” or “Set” primitives are allowed.

2.2.6.6 General error

This is an error that cannot be more closely defined (FFFF$_{hex}$).

Code	Meaning	Explanation
FFFF$_{hex}$	Internal_Error	Internal error

2.3 Interbus hardware

The OPIBS option can be used on the MCU-3000 motherboard as a component mounting variant, that is, no other slots are required for the Interbus coupling. The periphery devices are connected to the MCU-3000 with the aid of 10-pin FB connectors. A loom of cables with a cardholder and an integrated 9-pin SUB D connector is also available for the MCU-3000. This cardholder can be installed in a free PC slot next to the control system. If necessary, an existing open section of the PC mechanics can be used for this purpose.

Note: The 10-pin FB connector P1 is marked with Interbus.

2.3.1 Connector P1 (Interbus): 9-pin SUB-D connector

The remote bus (2 conductor) of the OPIBS Interbus system is connected to connector P1. It is executed on the cardholder as a 9-pile D-SUB plug and is electrically isolated from the potential of the host PC. The casing of the plug is connected conductively to the PC cardholder. The pin assignment has been selected so that the standard Interbus remote bus cable (e.g. IBS RBC METER-T) by PHOENIX CONTACT or other manufacturers can be used.

Table 2-12: Pin assignment of the remote bus interface (SUB-D)

Pin SUB-D	Name	Description/comments	Source (FB)
1	DO	Remote bus outgoing line	MCU-3000 FB-10-P1.1
2	DI	Remote bus receiving line	MCU-3000 FB-10-P1.3
3	COM	Earth	MCU-3000 FB-10-P1.5
4		Not assigned/reserved	
5	+5V	Internal distribution voltage	MCU-3000 FB-10-P1.9
6	/DO	Remote bus inverted outgoing line	MCU-3000 FB-10-P1.2
7	/DI	Remote bus inverted receiving line	MCU-3000 FB-10-P1.4
8		Not assigned/reserved	
9		Not assigned/reserved	

2.3.2 OPIBS option technical data

Characteristic	Description/properties
Max. number of Interbus subscribers	512
Structure	Component mounting variant on the APIC-8001 motherboard
Max. number of process data	256 words (4096 binary inputs and outputs)
Max. number of PCP subscribers	62 (in preparation)
Connector	10-pin FB connector (P1) for Interbus periphery connection

3 SUB-D connector for release relay, CNC ready relay

3.1 SUB-D adapter from 10-pin FB connector X5 to 9-pin SUB-D connector

At connector X5, relay points are provided for the CNC ready-request and amplifier release. These are ‘normally open’ contacts. All relays are switched off after the PC is switched on, after a reset action or after an error occurs.
The CNC ready relay is activated after the boot process (*mcbt.exe*).
The release relay is activated for the respective selected axis channels by using the *cl()* command for PCAP and the *CL()* command for SAP.
This adapter cable is used to convert the X5 connector to a 9-pin SUB-D connector. This connector is fitted to a PC slot board. The conversion results in a different pin assignment than to X5.

Table 3-1: Pin assignment for relay adapter (SUB-D-09)

Pin SUB-D	Name	Description/comments	Source
1	P contact	CNC ready	MCU-3000 FB-10-P5.1
2	P contact	Release axis channel 1	MCU-3000 FB-10-P5.3
3	P contact	Release axis channel 2	MCU-3000 FB-10-P5.5
4	P contact	Release axis channel 3	MCU-3000 FB-10-P5.7
5		Not connected	
6	‘Normally open’ contact	CNC ready	MCU-3000 FB-10-P5.2
7	‘Normally open’ contact	Release axis channel 1	MCU-3000 FB-10-P5.4
8	‘Normally open’ contact	Release axis channel 2	MCU-3000 FB-10-P5.6
9	‘Normally open’ contact	Release axis channel 3	MCU-3000 FB-10-P5.8

3.2 SUB-D adapter from 10-pin FB connector X5 or X2 to 25-pin SUB-D connector (male)

Relay points for the CNC ready request and the amplifier releases are provided at connector X5 of the MCU-3000 and to pins 1-10 of X2 of the OPMF Optionprint. These are ‘normally open’ contacts. This means that the release relays are available for up to 8 axes on a connector. All relays are switched off after the PC is switched on, after a reset action or after an error occurs.
The CNC ready relay is activated after the boot process (*mcbt.exe*).
The release relay is activated for the respective selected axis channels by using the *cl()* command for PCAP and the *CL()* command for SAP.
With this adapter cable, connector X5 of the MCU-3000 and pins 1-10 of X2 of the OPMF are converted to a 25-pin SUB-D connector. This connector is fitted to a PC slot board. The conversion results in a different pin assignment than to X5/X2.

Table 3-2: Pin assignment for relay adapter (SUB-D-25)

Pin SUB-D	Name	Description/comments	Source
1	P contact	CNC ready	MCU-3000 FB-10-P5.1
2	P contact	Release axis channel 1	MCU-3000 FB-10-P5.3
3	P contact	Release axis channel 2	MCU-3000 FB-10-P5.5
4	P contact	Release axis channel 3	MCU-3000 FB-10-P5.7
5			
6	P contact	Release axis channel 4	OPMF FB-60-X2.1
7	P contact	Release axis channel 5	OPMF FB-60-X2.3
8	P contact	Release axis channel 6	OPMF FB-60-X2.5
9	P contact	Release axis channel 7	OPMF FB-60-X2.7
10	P contact	Release axis channel 8	OPMF FB-60-X2.9
11		Not connected	
12		Not connected	
13		Not connected	
14	‘Normally open’ contact	CNC ready	MCU-3000 FB-10-P5.2
15	‘Normally open’ contact	Release axis channel 1	MCU-3000 FB-10-P5.4
16	‘Normally open’ contact	Release axis channel 2	MCU-3000 FB-10-P5.6
17	‘Normally open’ contact	Release axis channel 3	MCU-3000 FB-10-P5.8
18			
19	‘Normally open’ contact	Release axis channel 4	OPMF FB-60-X2.2
20	‘Normally open’ contact	Release axis channel 5	OPMF FB-60-X2.4
21	‘Normally open’ contact	Release axis channel 6	OPMF FB-60-X2.6
22	‘Normally open’ contact	Release axis channel 7	OPMF FB-60-X2.8
23	‘Normally open’ contact	Release axis channel 8	OPMF FB-60-X2.10
24		Not connected	
25		Not connected	

4 SUB-D connector for CAN bus

4.1 SUB-D adapter from 10-pin FB connector P3 to 9-pin SUB-D connector

At connector P3, the connectors for the CAN bus are provided as an option. This adapter cable is used to convert the P3 connector to a 9-pin SUB-D connector. This connector is fitted to a PC slot board. The conversion results in a different pin assignment than to P3.
CAN is a 2-wire bus system to which all subscribers are connected in parallel. The bus has to be terminated at each end by a termination resistor of 120 ohms in order to avoid any reflection. This is also required for a short line length.
On the control unit, there is no line termination available. So it is absolutely necessary for the user to realise this externally.

Table 4-1: CAN bus pin assignment (SUB-D-09)

Pin SUB-D	Name	Description/comments	Source
1			MCU-3000 FB-10-P3.1
2	CAN_LOW	CAN data line -	MCU-3000 FB-10-P3.3
3	GND	Ground	MCU-3000 FB-10-P3.5
4			MCU-3000 FB-10-P3.7
5			MCU-3000 FB-10-P3.9
6	GND	Ground	MCU-3000 FB-10-P3.2
7	CAN_HIGH	CAN data line +	MCU-3000 FB-10-P3.4
8			MCU-3000 FB-10-P3.6
9			MCU-3000 FB-10-P3.8

5 Annex

5.1 Index of figures

5.2 Index of tables