

PBX21-AXIS

Technical Information

Submodules for PBX21-AXIS (4-axis controller)

Outputs:

MAO 4 4 analog outputs
MDO 8 / MDIO 16 8 / 16 digital outputs

Inputs:

MAI 4 4 analog inputs
MDI 8 / MDIO 16 8 / 16 digital inputs
MSSI 2 2 SSI absolute encoder interfaces
MINC 2 2 incremental encoder interfaces
MHAS 2 2 high-velocity asynchronous interfaces

Edition date/Rev. date: 19.02.1999
Document no./Rev. no.: TRS - V - BA - GB - 0088 - 03
Software version:
File name: TRS-V-BA-GB-0088.DOC
Author: DOL / MÜJ

TRSystemtechnik GmbH

Eglishalde 6

D-78647 Trossingen

Germany

Tel. +49 - (0) 7425 / 228-0

Fax +49 - (0) 7425 / 228-34

Imprint

TRSystemtechnik GmbH

D-78647 Trossingen
Eglishalde 6
Tel.: (+49) 07425/228-0
Fax: (+49) 07425/228-34

© Copyright 1998 TRSystemtechnik

Right of Modification

In the interests of our ongoing efforts to improve our products, TRSystemtechnik reserve the right to alter the information contained in this document without prior notice.

Print

This manual was edited on a DOS personal computer using text formatting software. The text was printed in *Arial*.

Notation

Italics and **bold** type are used for the title of a document or to emphasize text passages.

Passages written in *Courier* indicate text which is visible on the display as well as software menu selections.

"< >" refers to keys on your computer keyboard (e.g. <RETURN>).

Note

Text following the "NOTE" symbol describes important features of the respective product.

Copyright Information ©

MS-DOS is a registered trademark of Microsoft Corporation.

Revision History

i

Note:

The cover of this document shows the current revision status and the corresponding date. Since each individual page has its own revision status and date in the footer, there may be different revision statuses within the document.

Document created:

03.04.1998

Revision	Date
New chapter "Example Axis Initialization"	21.04.1998
Text supplemented on page 41; text modified and supplemented on page 48	19.02.1999

Table of contents

1 PBXAXIS hardware	7
1.1 Profibus interface.....	7
1.1.1 Data transfer between positioning module and computer	7
2 Controller algorithm	9
2.1 Block diagram PBXAXIS controller.....	9
2.2 PID algorithm.....	9
2.2.1 The proportional controller (P-controller).....	10
2.2.2 The integral controller (I-controller)	10
2.2.3 The differential controller (D-controller).....	10
2.2.4 PBXAXIS special function	11
2.2.4.1 Set point value correction by integration via constants.....	11
2.2.4.2 Smoothing of manipulated variable with P-controller	11
3 PBXAXIS DPRAM definition	12
3.1 Type specification.....	12
3.2 PBXAXIS parameters.....	13
3.3 PBXAXIS write parameters (control byte: bit $2^7 = 1$).....	13
3.3.1 Controller system parameters	14
3.3.1.1 Sampling interval	14
3.3.1.2 Number of axes.....	14
3.3.2 Initialization parameters.....	15
3.3.2.1 Controller control block.....	15
3.3.3 Movement parameters.....	18
3.3.3.1 Variable for home return command	20
3.3.3.2 Variable for controlled home return.....	21
3.3.4 Digital outputs.....	22
3.3.5 Special parameters.....	23
3.3.5.1 Master / slave operation	23
3.3.5.2 Set axis to preset position value	24
3.4 Input data (read parameters, (control byte: bit $2^7 = 0$).....	25
3.4.1 System data	25
3.4.1.1 Error flag (error message signal)	25
3.4.2 Axis data	25
3.4.2.1 Error code (error message mask)	25
3.4.2.2 Axis status.....	26
3.4.2.3 Controller state.....	26
3.4.2.4 Moving state (internal state of controller).....	26
3.4.2.5 Actual value	26
3.4.2.6 Dynamic set point value	27
3.4.2.7 Delta actual value.....	27
3.4.2.8 Current drift compensation / deviation	27
3.4.2.9 Home return started.....	27
3.4.3 Digital inputs.....	28
4 PBXAXIS commands (COMCODE)	29
4.1 Controller features and “controller control block “	29
4.1.1 Accept controller control block (COMCODE = 100).....	29
4.1.2 Controller ON (COMCODE = 110)	29
4.1.3 Controller OFF (COMCODE = 111)	29

4.1.4	Accept controller control block & controller "ON" (COMCODE=140)	29
4.1.5	Clear I-component (COMCODE = 150)	29
4.1.6	Activate drift compensation (COMCODE = 152)	30
4.1.7	Disable drift compensation (COMCODE = 151)	30
4.1.8	Delete drift compensation (COMCODE = 153)	30
4.1.9	Activate smoothing of manipulated variable (COMCODE = 112)	30
4.1.10	Disable smoothing of manipulated variable (COMCODE = 113)	31
4.1.11	Activate integral set point value correction (COMCODE = 115)	31
4.1.12	Disable integral set point value correction (COMCODE = 116)	31
4.2	Movement commands	31
4.2.1	Axis to target or controlled movement (COMCODE = 130)	31
4.2.2	Stop axis (COMCODE = 131)	32
4.2.3	Axis ramp stop (COMCODE = 134)	32
4.2.4	Home return (COMCODE = 122)	32
4.2.5	Controlled home return (COMCODE = 123)	33
4.2.6	Home return to the stop (COMCODE = 124)	33
4.2.7	Movement with variable acceleration (COMCODE = 135)	34
4.2.8	Controlled movement with home position_V1 (COMCODE = 132)	34
4.2.9	Controlled movement with home position_V2 (COMCODE = 133)	34
4.2.10	Interpolating motion of several axes (COMCODE = 190)	35
4.3	System commands	36
4.3.1	Clear error (COMCODE = 500)	36
4.3.2	Set sampling interval (COMCODE = 501)	36
4.3.3	Set number of axes (COMCODE = 502)	36
4.3.4	Clear error code (COMCODE = 199)	36
4.4	Special commands	37
4.4.1	Activate deviation output (COMCODE 117)	37
4.4.2	Activate drift compensation output (COMCODE 118)	37
4.4.3	Activate synchronous run (COMCODE = 128)	37
4.4.4	Set preset value (COMCODE = 126)	38
4.4.5	Limit switch monitoring (COMCODE = 510-512)	38
4.4.5.1	Mode 1: Start-up with command 510	38
4.4.5.2	Mode 2: Start-up with command 511	38
4.4.5.3	Disable limit switch monitoring with command 512	38
4.4.6	Axis release (COMCODE = 513-514)	39
4.4.7	Brake operation (COMCODE = 515-516)	39
5	Positioning with digital outputs	40
5.1	Positioning with MDIO16	40
5.1.1	Initialize/activate mode	40
5.1.1.1	Describe parameters	40
5.1.1.2	Accept initialization data (COMCODE = 300)	43
5.1.2	Positioning	43
5.1.2.1	Program target	43
5.1.2.2	Start positioning (COMCODE = 320)	43
5.1.3	Manual operation	44
5.1.4	Digital inputs/outputs	45
5.1.4.1	Inputs	45
5.1.4.2	Outputs	45
5.1.5	Actual values	46
5.1.6	Home return with reset pulse (COMCODE = 322)	47

5.1.7 Home return without reset pulse (COMCODE = 324)	48
5.2 Positioning with MDO8.....	49
5.2.1 Initialize/activate mode	49
5.2.1.1 Describe parameters	49
5.2.1.2 Accept initialization data (COMCODE = 400).....	53
5.2.2 Positioning.....	53
5.2.2.1 Program target	53
5.2.2.2 Start positioning (COMCODE = 420)	53
5.2.3 Manual operation	54
5.2.4 Home return (COMCODE = 424)	54
5.2.5 Start cut-off point correction (COMCODE = 430)	55
6 PBXAXIS parameter definition (address / type).....	56
6.1 Mode: Positioning with digital outputs.....	60
7 Summary of comcodes	63
7.1 Mode: Positioning with digital outputs.....	64
8 Example Axis Initialization	65

1 PBXAXIS hardware

The 4-axis controller has been developed from the FOX-20 basic module (B20) with the PBX-21 bus connection. The hardware components can be taken from the resources of the FOX system.

Note:

The following rules have to be observed:

- Slot 1 is reserved for a MAO4 (4-channel analog output).
The following module/channel assignment applies:
 - Channel 1 = Axis 1
 - Channel 2 = Axis 2
 - Channel 3 = Axis 3
 - Channel 4 = Axis 4
- Slots 2, 3 and 4 are reserved for checkbacks of the actual values and are freely selectable.
- Digital input/output modules are also supplied, but may only be inserted in slots 2 and 3.
- On application of the "limit switch monitoring function", the MDI module must be inserted in slot 2.
- On application of the "axis release function", the MDO module must be inserted in slot 3.

1.1 Profibus interface

1.1.1 Data transfer between positioning module and computer

The positioning module is accessed via a 6-byte wide interface.

Identification code of the positioning module:

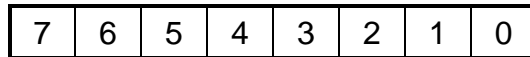
6 byte I/O, consistency over entire length: 0xB5 (181)

This identification can be entered up to four times. So it is possible to address all four axis at the same cycle!

Definition of the data channel (viewed from the master):

	Input	Output
Byte 0	Status byte	Control byte
Byte 1		Parameter number (address)
Byte 2	MSB byte	MSB byte
Byte 3		
Byte 4		
Byte 5	LSB byte	LSB byte

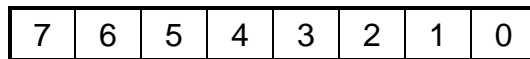
Control byte



Bit 7: Write/read bit: 0: Data are read, 1: Data are written
 Bit 6: Toggle bit
 Bit 5: PLC identification bit 1: PLC in operation, 0: PLC in STOP (all axes stationary)

Status byte:

The status byte is the copied control byte.



Bit 6: Toggle bit
 Bit 5: Communication status bit 1: Parameter/command understood
 0: Parameter/command not defined

Data:

The data are stored in bytes 2 to 5.

Value type:	-word (2 bytes):	MSB byte = byte 4
		LSB byte = byte 5
	- long type (4 bytes):	MSB byte = byte 2
		LSB byte = byte 5

Send parameters to positioning module

If data are to be sent to the module, the data and the parameter number must first be entered, then the write bit is set. On receipt, the data are copied to the sending buffer, and the higher-level computer can read out the sent data.

In order to detect whether the positioning module has received the data, the status of the toggle bit must be modified after each transfer. As the data are recopied to the sending buffer on receipt of a message, the toggle bit can be read out again, and the receipt of the message can be detected via the status change.

Note:

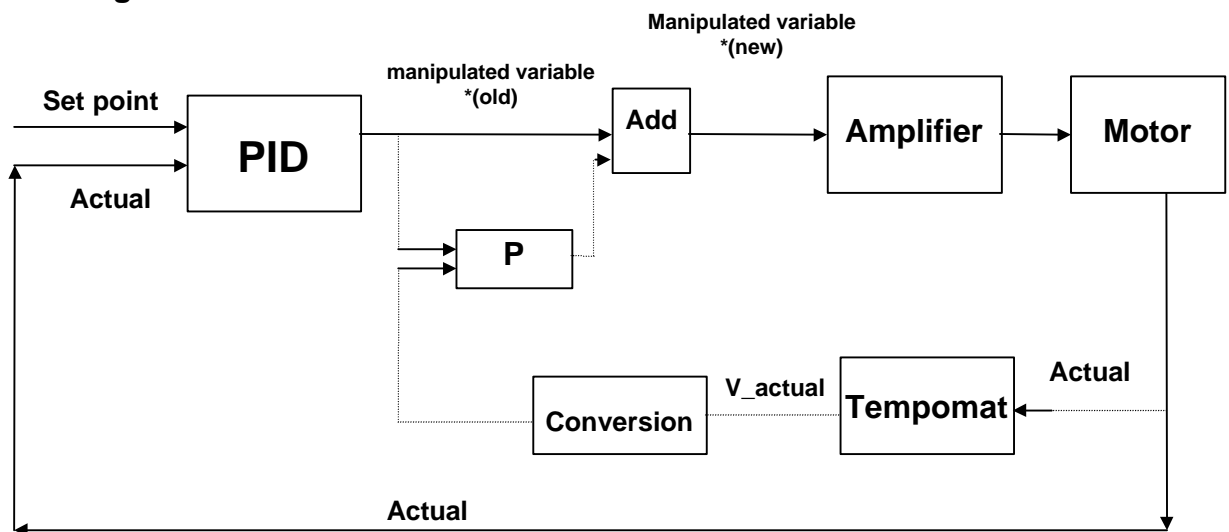
The above-mentioned method of sending data to the module only works properly if consistency over 6 bytes is ensured. If not, the write/read bit should be set to 0 first.

Read parameters from positioning module

If data are to be read from the module, the write/read bit must be set to 0. As soon as a message has been received, the desired data are copied to bytes 2 to 5 and the control byte and the parameter number are copied to the sending buffer and can be read out by the master.

2 Controller algorithm

2.1 Block diagram PBXAXIS controller



The PID controller in the FOX-20 is a PID controller with variable structure. The mathematical description is given below.

* If dotted control loop is switched on

2.2 PID algorithm

In continuous-action devices, a PID controller regulates the manipulated variable in order to reset the deviation (e). The deviation here means the difference between set point value (w) and actual value (x). The principle of the PID controller is based on the equation representing the manipulated variable M (t) as the product of a proportional component, an integral component and a differential component. To implement the control unit in a digital computer, the continuously operating function must be converted to regular scanning of the deviation with subsequent calculation of the manipulated variable.

Definition: The deviation is: $e = w - x$

$M(t) = K_{Prop} * e + K_{integral} * \int e dt + K_{diff} * de/dt$ <p>Manip. variable = proporti. compo. + integral component + differential component</p>

2.2.1 The proportional controller (P-controller)

The proportional controller is defined by a simple mathematical relation

$$Y = K_{\text{proportional}} * e + C .$$

This continuously operating function must be converted to regular scanning of the deviation. The following relation applies to the proportional component:

$$Y_P = K_{\text{proportional}} * e_n$$

2.2.2 The integral controller (I-controller)

The integral controller (I-controller) has the equation

$$Y_{\text{integral}} = K_{\text{integral}} * \int e dt.$$

The following applies to the conversion to regular scanning in digital technology:

$$Y_{\text{Integral}} = K_I * \sum_1^n e_n$$

Integral controller with variable structure

For starting up control loops, it has proved advantageous to start with a P-algorithm only, whereby the I-controller is not introduced until the system deviation (w-x) has diminished. For this reason, a parameter is defined which allows the user to determine the range in which the I-controller is to operate, in which the integral component is kept constant.

2.2.3 The differential controller (D-controller)

The differential controller (D-controller) has the equation

$$Y_{\text{differential}} = K_{\text{differential}} * de/dt.$$

Here the differentiation **de/dt** is replaced by

$$\frac{\Delta e}{\Delta T} = \frac{e_{\text{new}} - e_{\text{old}}}{\Delta T}$$

This yields $Y_{\text{diff}} = K_{\text{diff}} * (e_{\text{new}} - e_{\text{old}})$

The following equation therefore results for the manipulated variable in the PBXAXIS:

$$Y_{\text{PBXAXIS}} = Y_{\text{proportional}} + Y_{\text{integral}} + Y_{\text{differential}};$$