TSP10-PB field bus appendix

AHS Antriebstechnik GmbH
Fichtenweg 17
64319 Pfungstadt
Phone: +49 6157 9866110
Fax: +49 6157 9866112

AHS
Antriebstechnik
www.AHS-Antriebstechnik.de

Previous issues:

Issue	Remarks
2013-08-22	First edition
2014-01-09	Small technical changes
2014-08-20	Change Firmware 2.0 (AHS+1648.GSD)

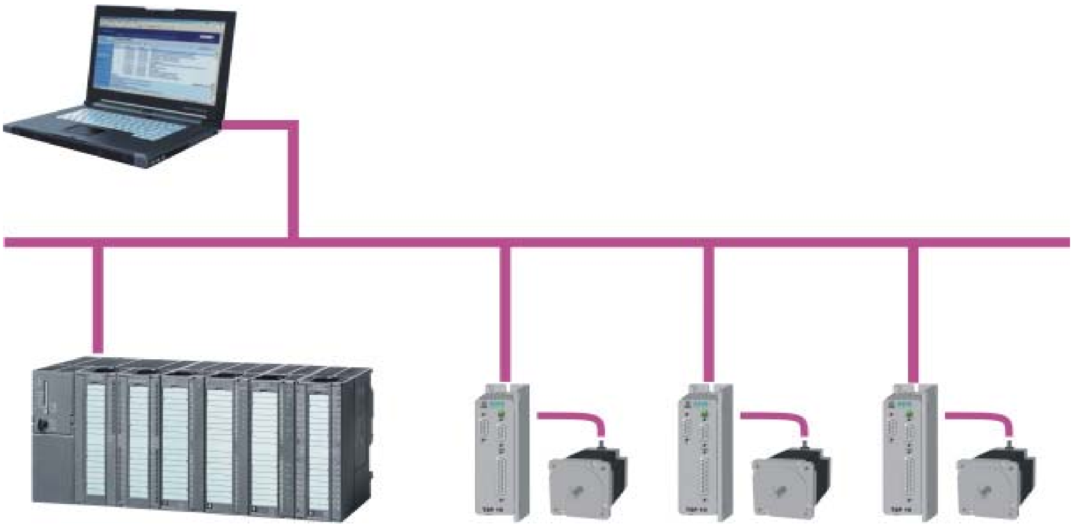
Errors excepted!
Changes of the documentation and the product's technology reserved!

The current issue of this manual is available online at http://www.ahs-antriebstechnik.de/home.html.

TSP10-PB - Compact stepper motor drive

1 Profibus settings

The TSP10-PB is a modular station with 2 modules. There are a 12-byte output module and an 8-byte input module available. The stepping motor control may be operated in velocity or positioning mode. The profibus master is able to initiate operations of the driving mechanism by setting control bits in the command word (bytes 7-8).



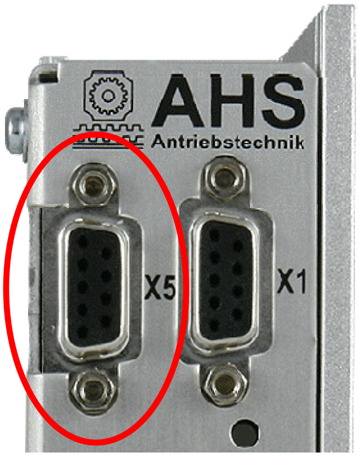
The control can any time request the current state and position by reading the input bytes.

The absolute target position that the stepping motor control should reach during the next motion profile is entered in the output bytes 5 to 8. This allows performing a precise decentralised relocation without burdening the master. Please, request a sample project of an S7 control by email (info@ahs-antriebstechnik.de).

In this appendix you may read about the add-ins of the TSP10-PB and the differences from the basic device TSP10 BA. The general functions of the device are described in the TSP10 manual (http://www.ahs-antriebstechnik.de/downloads2-gb.html).

1.1 Connector configuration

Connect the profibus to the additional 9-pin Sub-D connector X5 on the left, next to the serial interface X1 jack. All signals of this interface are opto-isolated. Baud rates up to 12 Mbaud are supported.



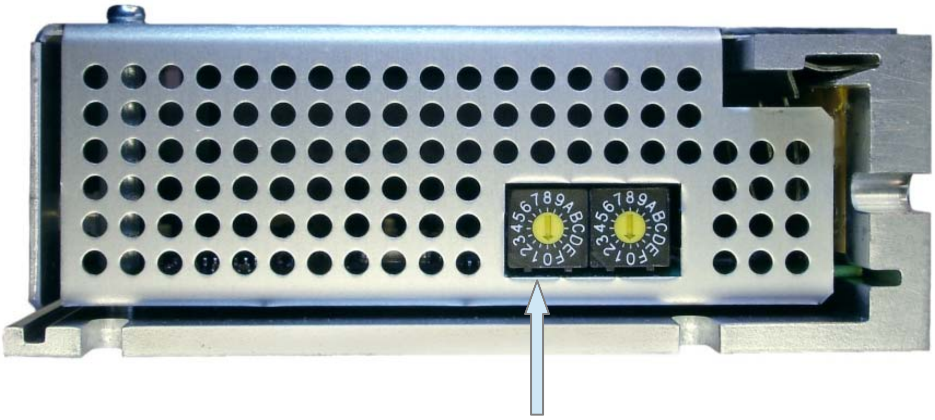
The inputs for limit, reference and stop switches of the TSP10-PB are defined as follows:

GND-DE	reference potential	X2 pin 1
DE2	lower limit switch	X2 pin 10
DE3	upper limit switch	X2 pin 11
DE4	reference switch	X2 pin 12
DE5	stop switch	X2 pin 13

The inputs are opto-isolated and designed for 24 V or 5 V (see ordering code). The inputs may be disabled via the parameter data (see chapter 1.3).

1.2 Profibus address

The two rotary switches of the TSP10-PB help to set the profibus address. Set motor current and step size via the parameter data (see 1.3).

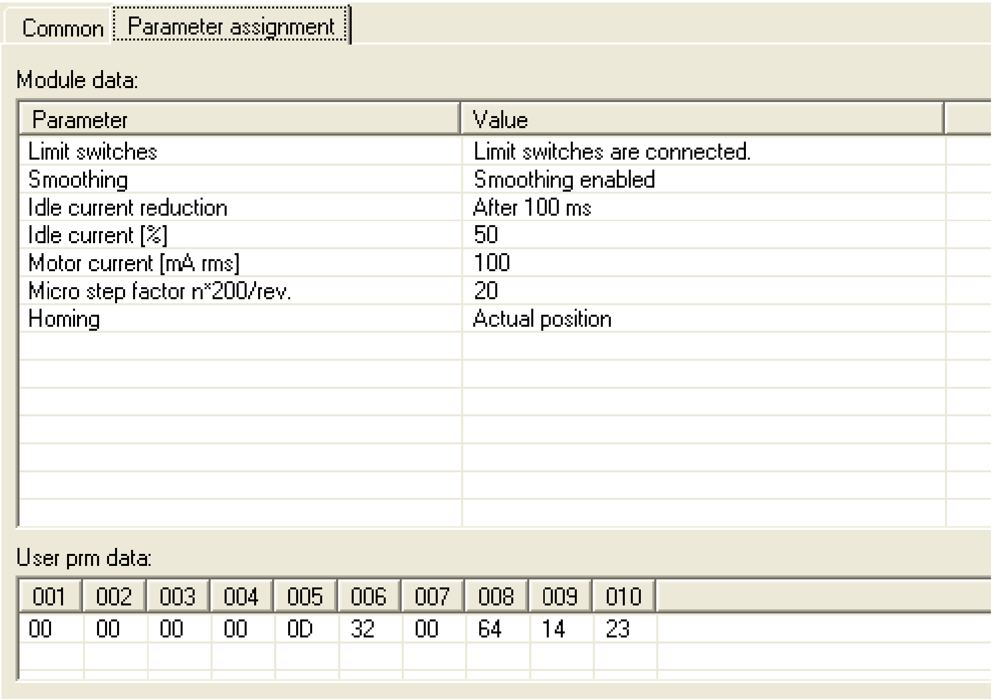


High-value address (factor: 16)

Set the address in hex values. For any address up to 15, keep the left rotary switch on zero. For higher addresses, multiply the left rotary switch setting by 16 and add to the value on the right.

1.3 Parameter data

Set the parameters for motor current, micro step factor, limit switch, homing, smoothing and idle current reduction via the Profibus user parameter data. The default is set in the GSD file. Excerpts from the GSD file are shown *`in italics`*.



Module data:

Parameter	Value
Limit switches	Limit switches are connected.
Smoothing	Smoothing enabled
Idle current reduction	After 100 ms
Idle current [%]	50
Motor current [mA rms]	100
Micro step factor n*200/rev.	20
Homing	Actual position

User prm data:

001	002	003	004	005	006	007	008	009	010
00	00	00	00	0D	32	00	64	14	23

There are 10 parameter bytes: *`User_Prm_Data_Len = 10`*

1.3.1 Micro step factor (step size)

Select the micro step factor (byte 9) anywhere between 1 and 128 (200 and 25600 steps per turn) are available. Higher resolution provides steadier operation at low speed.

```
ExtUserPrmData     = 1 "micro step factor n*200 /revolution"

Unsigned8 20 1-128

EndExtUserPrmData
```

Example:

A drive rotates a spindle that generates per revolution 4 mm feed. The total distance is 700 mm (175 revolutions). With a selected micro step factor of n = 20 the micro step resolution is 4000 steps per revolution. The target position can then be specified in µm (micrometers). The end position would be 700,000.

1.3.2 Limit switch

If the limit switches (X2 pin 10 and 11) are not used, set the parameter (byte 4 bits 0) at zero, so that the motor is able to move. The limit switches serve as openers and, when open, prevent any motion of the motor.

```
PrmText              = 2

Text(0)=" Limit switches not connected!"

Text(1)=" Limit switches are connected."

EndPrmText
```

1.3.3 Smooting

You achieve smooth running of the motor by enabling the function Smoothing (byte 4 bit 1).

```
PrmText            = 3

Text(0)="without Smoothing"

Text(1)="with Smoothing"

EndPrmText

ExtUserPrmData     = 3 "Smoothing"
Bit (1) 0 0-1
Prm Text Ref       = 3
```

1.3.4 Motor current

The motor current may be modified between 100 and 7000mA (byte 6 and 7). The value set may never exceed the permissible motor current, not even briefly.

```
ExtUserPrmData      = 6 "Motor current [mA rms]"
Unsigned16 100 100-7000
EndExtUserPrmData
```

1.3.5 Idle current reduction

The holding time until the idle current is reduced may be set in 8 steps (byte 4 bits 2-5). The motor current is dictated as a percentage of the set value (byte 5) so that the heating of the motor will be reduced.

```
PrmText             = 4
Text(0)="No idle current reduction"
Text(1)="After 25 ms"
Text(2)="After 50 ms"
Text(3)="After 100 ms"
Text(4)="After 250 ms"
Text(5)="After 500 ms"
Text(6)="After 1 second"
Text(7)="After 2 second"
EndPrmText
ExtUserPrmData      = 4 "idle current reduction"
BitArea (2-5) 3 0-7
Prm Text Ref        = 4
EndExtUserPrmData
ExtUserPrmData      = 5 "reduced idle current [%]"
Unsigned8 50 0-100
EndExtUserPrmData
```

1.3.6 Homing

Using the homing method you can set over byte 9.

```
PrmText             = 5

Text(35)="Current position"

Text(17)="Lower limit switch"

Text(18)="Upper limit switch"

Text(24)="Lower reference switch (Fahrtr. +)"

Text(29)="Lower reference switch (Fahrtr. -)"

Text(25)="Upper reference switch (Fahrtr. +)"

Text(28)="Upper reference switch (Fahrtr. -)"

Text(250)="Lower mechanical limit"

Text(251)="Upper mechanical limit"

EndPrmText

ExtUserPrmData      = 7 "Homing"

Unsigned8 35 17-251

Prm Text Ref        = 5

EndExtUserPrmData
```

1.3.7 Diagnostic data

The revision number (last number) at the firmware as a 16-bit value is transmitted as status information in the user-specific diagnostics (EXT_DIAG).

For e.g.: Firmware 2.0.49

User diagnostics:

Byte 1: 3 (Length)

Byte 2: 0 (High-Byte)

Byte 3: 49 (Low-Byte)

2 Module 1 (output bytes)

The following table lists the individual output bytes of the TSP10-PB control. Elements used only for positioning have a green background. Values used in the velocity mode are highlighted in orange.

<table>
<tr><th colspan="20">Module 1
12 output bytes</th></tr>
<tr><td>1</td><td rowspan="2">Velocity (rpm)</td><td colspan="18">High byte</td></tr>
<tr><td>2</td><td colspan="18">Low byte</td></tr>
<tr><td>3</td><td rowspan="2">Initial velocity (rpm)</td><td colspan="18">High byte</td></tr>
<tr><td>4</td><td colspan="18">Low byte</td></tr>
<tr><td>5</td><td>Time base of acceleration</td><td colspan="18">[ms]</td></tr>
<tr><td>6</td><td>Acceleration time</td><td colspan="18">Time of acceleration;
Unit is the time base (Byte 5)</td></tr>
<tr><td>7+8</td><td>Command word</td><td></td><td></td><td></td><td></td><td></td><td></td><td>C</td><td>P</td><td>H</td><td>E</td><td>D</td><td>S</td><td>V</td><td>A</td><td>R</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td rowspan="4">Target position
(32 bits integer)</td><td colspan="8" rowspan="2">Low word</td><td colspan="10">High byte</td></tr>
<tr><td>10</td><td colspan="10">Low byte</td></tr>
<tr><td>11</td><td colspan="8" rowspan="2">High word</td><td colspan="10">High byte</td></tr>
<tr><td>12</td><td colspan="10">Low byte</td></tr>
</table>

2.1 Command word

Byte	Bit		Meaning	Description
8	0			
	1	R	Counter Reset	1 = Set current position at 0
	2	A	Acceleration	0 = no ramp (jump) 1 = linear velocity change
	3	V	Velocity mode	0 = Positioning 1 = Velocity mode
	4	S	Start motor	0 = Stop motor 1 = Start positioning
	5	D	Direction	0 = forward 1 = backward
	6	E	Enable	0 = Motor is without current 1 = Holding torque or torque active
	7	H	Homing	0 = Standard mode 1 = Homing
7	0	P	Polarity	Preferred direction of rotation (forward clockwise)
	1	C	Stop switch	0 = closer, 1 = opener

The positioning mode always applies a ramp, the sense of rotation (direction) is a function of set and current position.

2.2 Velocity and start velocity

In the first two output bytes, the speed in revolutions per minute is registered. Should the motor when starting first to jump on an initial velocity, you must preset the value in byte 3 and 4.

2.3 Acceleration

In byte 5 is given the time base for the acceleration and in byte 6 is given the acceleration time. The acceleration results from the set speed divided by the time for the acceleration. In velocity mode, 100 rpm as speed is used for the calculation of acceleration. The product for the time base and acceleration time results at the time for the speed ramp in milliseconds. Is the product 0 (zero) the acceleration is set to 1 second.

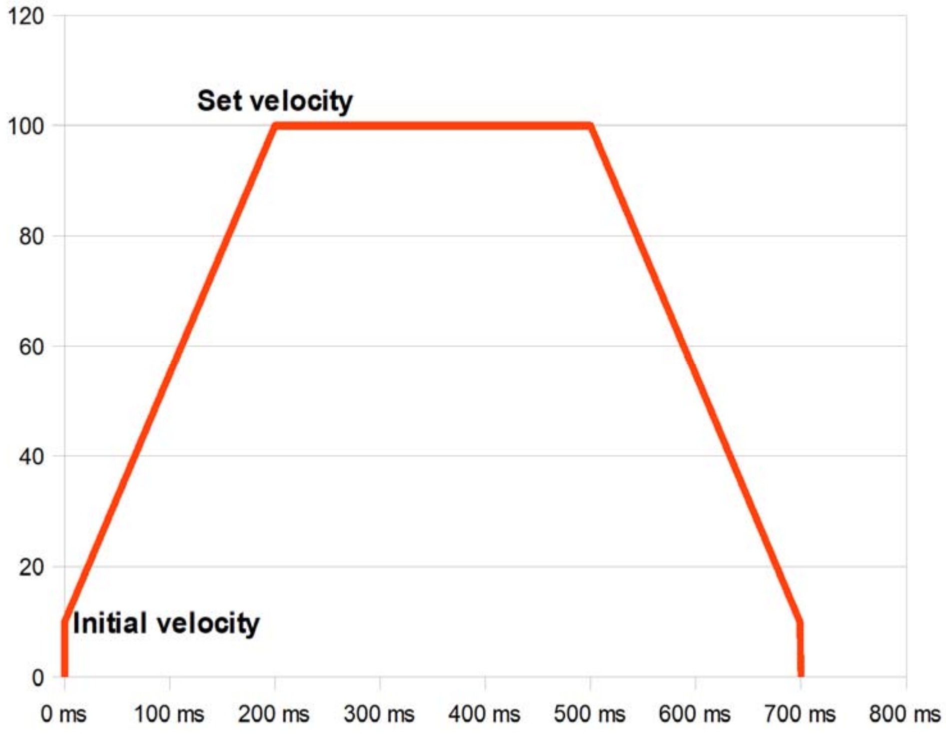


Fig. 1: Velocity profile of a positioning

Time base = 10 ms, acceleration time = 20 [10 ms]

2.4 Target position

The output bytes 9-12 contain the target position desired in the travel command. It has 32 bits integer to permit high resolution and absolute positioning even across a great distance.

The reference point basically serves to align in absolute positioning. This is the position 0 (zero). That value was defined to provide a sufficient value range for positioning left or right of the reference point. You can easily convert the position value into the user’s number format. The preferred 32-bit value is a reasonable starting point to convert into different customised number formats. Analogue input devices in the realm of PROFIBUS-DP and other field bus systems apply the same principle.

The current position of the stepper is registered in the input bytes 5-8. As mentioned before, the counter is set to 0 (zero) and the counter status in the status word is set to ‘Referenced’ as soon as the reference point has been reached by travelling. The position is undefined when the device is powered on, hence, the counter is then set to 0 (zero), allowing the user to travel either way. The control requests the actual counter reading to identify the current position of the driving mechanism and process it in the application software.

The control does not need to request the actual counter reading for positioning, the Busy message in the status word meets the same requirement. For each step, the module compares the current position and the target position to terminate the travel command as soon as both values agree.

Once the Stepper has reached the target position, the application software preselects a new target position. Before the Stepper starts again, though, the motor start bit of the command byte, which may have been left enabled by the preceding run, has to be disabled and re-enabled.

Note as well that the stepper automatically sets the direction in the positioning mode. Hence, the corresponding command bit is disabled. The direction chosen by the stepper always results from a match of current and target position.

The stop conditions (stop, limit, emergency shutdown switch) are constantly checked while travelling. They enjoy higher priority, of course, i. e. if the given position value is not reached yet and the corresponding switch is pressed, then the motor is shut down at once.

If there is any limit switch or alarm stop or if reaching the target position triggers a stop, the motor will restart only after the motor start bit was disabled and re-enabled.

2.5 Homing

The homing is a specific feature among the commands of the stepper motor controller. It is always started by setting the command bit 7. It gives different methods to making the homing run. The homing methods are listed in the parameter data. You can set the Homing velocity and acceleration as described in chapter 2.2 and 2.3.

After the successful homing, the drive is exactly on the reference point and the position counter get up 0 (zero). In the status word, it is indicated that the data of the position counter is valid. An additional homing is not running before with "counter reset" the status bit is deleted.

Now the controller is ready to accept absolute position commands. The control may initiate the homing any time. The homing is interrupted by the emergency shutdown, the stop switch or when the second limit switch is reached, because the reference switch was not found in that case. Emergency stop and reaching the lower limit switch trigger the alarm status while a response of the stop switch during the homing merely stops the driving mechanism, terminating the homing. The control can read all states from the status bytes and initiate corresponding procedures.

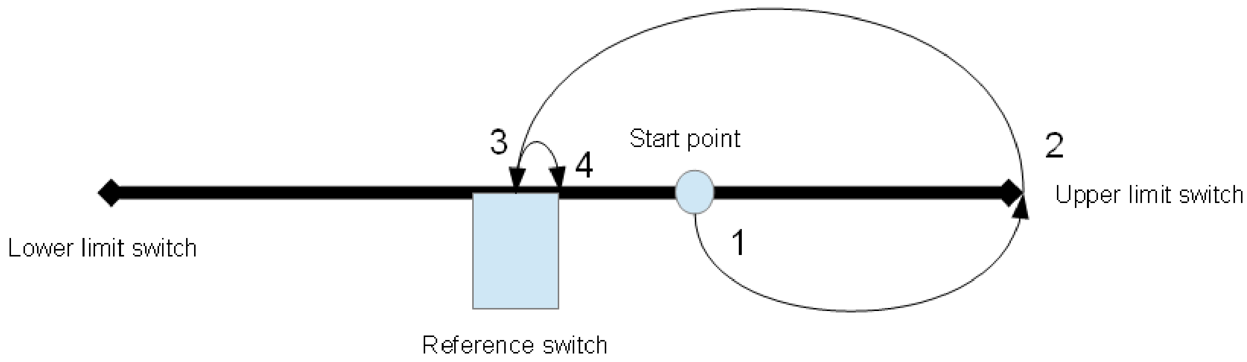
Example: **upper reference switch (positive direction)**, slide between reference point and upper limit switch: Configuration of the PBS-OUT bytes:

Velocity	PBS-OUT bytes 1+2	0x64
Command word	PBS-OUT bytes 7+8	0x00C0
Target position	PBS-OUT bytes 9-12	Any

After the command was given, the following configuration of the bytes PBS-IN results:

Current velocity	PBS-IN Bytes 1+2	0x64
Status bytes	PBS-IN Bytes 3+4	0x1000

1	Start: Position counter increments, velocity as set.
2	Upper limit switch responds. Direction inverts, position counter decrements. Velocity as set. Reference point is below the start position.
3	Reference switch responds. Direction inverts. Position counter value increments. Reference switch is met from above, back to edge of the reference switch.
4	Reference point is verified by leaving and triggering the reference switch again. Set position counter at 0 (zero). Motor has arrived at the reference point.



3 Module 2 (input bytes)

Module 2 8 input bytes																	
1	Current velocity	High byte															
2		Low byte															
3+4	**Status word**		E		C	A			L	B		V	D	↑	R	↓	S E E
5	**Current position (32 bits integer)**	Low word								High byte							
6										Low byte							
7		High word								High byte							
8										Low byte							

3.1 Status word

To monitor the state of the stepper motor controller in the control, read the status bytes (input bytes 3+4) of the controller. The following table tells which positions of the status bytes may be evaluated.

The input bytes 3 and 4 have the following significance:

Byte	Bit		Meaning	Description
4	0	S	Stop switch	0 = Stop switch not enabled 1 = Stop switch triggered
	1	↓	Lower limit switch	1 = Lower limit switch enabled
	2	R	Reference switch	1 = Reference switch enabled
	3	↑	Upper limit switch	1 = Upper limit switch enabled
	4	D	Direction of travel	0 = forward (incrementing current position) 1 = backward
	5	V	Final velocity	0 = not reached 1 = reached
	6			
	7	B	Busy	0 = Travel command terminated 1 = Travel command enabled
3	0	L	Limit switch	0 = no limit switch reached 1 = at least one limit switch reached
	1		Logic level Upper limit switch	
	2		Logic level Lower limit switch	
	3	A	Emergency shutdown/alarm	1 = Alarm or emergency stop enabled
	4	C	Counter status	1 = Counter reading is referenced
	5			
	6	E	Error	1 = Error

3.2 Current position and velocity

The values of the current position (5-8) and velocity (1-2) share the format of the specified values in the output bytes.

3.3 Emergency shutdown

The stepping motor control provides an option to trigger an emergency shutdown if any hazard or error has occurred. If an emergency shutdown occurred, the driving mechanism is stopped immediately and does not accept any further commands. The PROFIBUS DP is not interrupted, though, so that the other PROFIBUS DP members may continue to work.

The emergency shutdown function has no special input but is triggered by the simultaneous activation of both limit switches. Since the limit switches are basically designed as openers - to exclude malfunctions from broken wires -, the wiring of the emergency shutdown function should be designed in such a way that the inputs DE2 and DE3 are isolated from the 24 V level as soon as the emergency shutdown switch is pressed.

3.4 Alarm status

The alarm status is triggered by the following events:

1. At the homing to the reference switch was the second limit switch activated.
2. A processor error has occurred.

No commands are processed while the alarm status applies. The stepper blocks all command information. However, status information and position messages are still forwarded. The status information includes the "Alarm" message.

An option to cancel the alarm status without shutdown has been implemented by introducing a **pseudo-code** for the output bytes 1+2 (velocity). This is code **0xAA55**. That code does not occur in standard operation because the highest velocity is represented by 3000. If the stepper reads the pseudo-code it checks **once more** whether the conditions of the alarm status are still applying. If they aren’t, the stepper cancels the alarm status and returns to standard operation **after terminating** the pseudo-code. Make sure that the pseudo-code output 0xAA55 **is really terminated**, because the standard operation will resume only then.