
# TSP10-PBE field bus appendix

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

AHS Antriebstechnik
Advanced Hybrid Stepper Systems

## Table of changes:

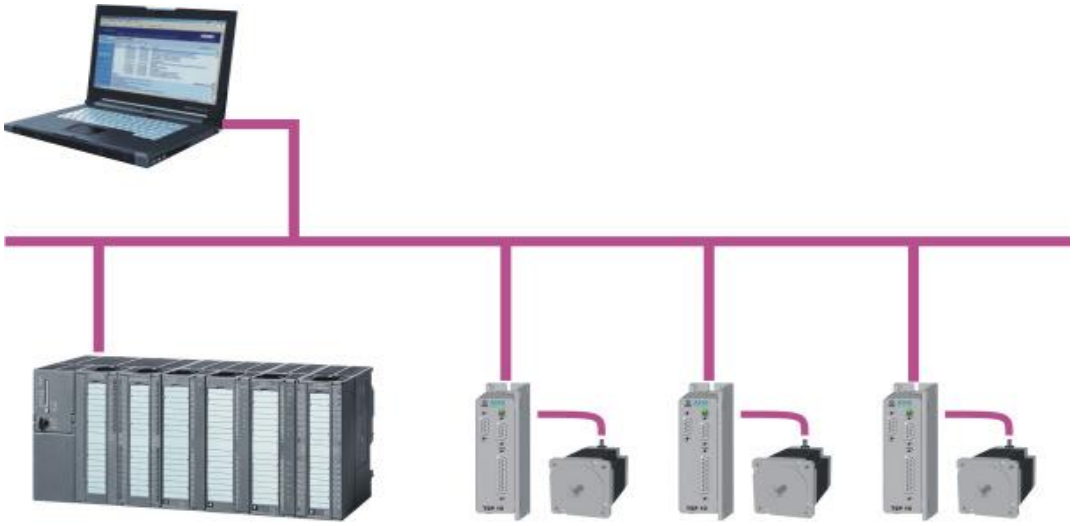
| Edition | Notes |
|---|---|
| 05.03.2015 | First edition |
| 15.01.2016 | Additions to the command and status word from revision 64 |
| 12.01.2024 | Revised version |
| | |

## Table of contents

# TSP10-PBE - Compact Stepper Motor Drives

## 1 Profibus-Motion tasks

The TSP10-PBE is a modular station with 2 modules. The Profibus ID is 0EE7h. There is an output module and an input module with 6 configurable words. The stepper motor drive can be operated in speed or positioning mode. The Profibus master can start actions of the drive by setting control bits in the command word.



The current status and the current position can be recorded by the control unit at any time by reading the input data.

The absolute target position that the stepper motor control should reach with the next movement profile is entered in the output data. This enables precise decentralized positioning without placing a load on the master. A sample project can be requested by e-mail (info@ahs-antriebstechnik.de).

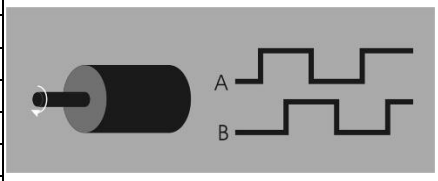
In this appendix you will find the additional functions of the TSP10-PBE and the differences to the basic device TSP10-BA. The general functions of the device are described in the TSP10 manual(https://www.ahs-antriebstechnik.de/en/products/stepmotor-drives).

## 1.1 Pin assignment

The Profibus is connected to the additional 9-pin Sub-D socket X5 to the left of the socket for the serial interface X1. All signals of this interface are optically isolated. Baud rates of up to 12 Mbaud are supported.


The encoder is connected via the 9-pin Sub-D socket X6 below the Profibus socket and has the following assignment:

| Signal | TSP10 X6 Pin | AE30 Wire color | M21 Wire color | DFS Wire color |
|---|---|---|---|---|
| +5V | 1 | Red | Red | Red |
| A | 2 | Violet | Green | White |
| B | 3 | Yellow | Orange | Pink |
| Index | 4 | Green | White | Violet |
| GND | 6 | Black | Black | Blue |
| /A | 7 | Brown | Red/Black | Brown |
| /B | 8 | Orange | White/Black | Black |
| /Index | 9 | Blue | Blue | Yellow |




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

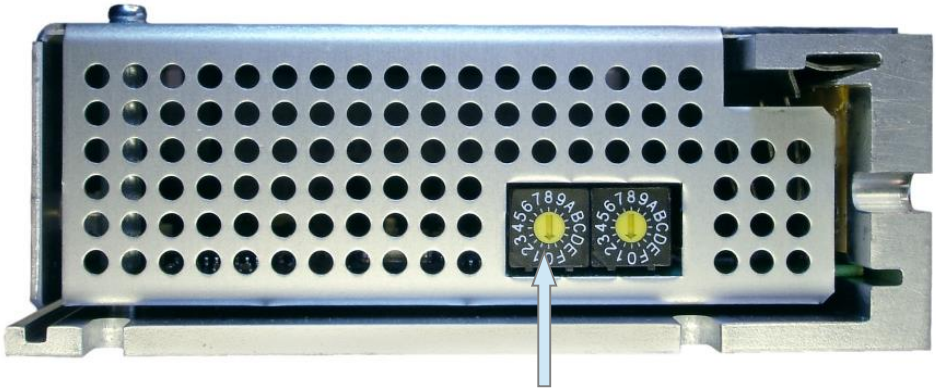
| | | |
|---|---|---|
| GND-DE | Reference potential | X2 Pin 1 |
| DE2 | Lower limit switch | X2 Pin 10 |
| DE3 | Upper limit switch | X2 Pin 11 |
| DE4 | Reference switch | X2 Pin 12 |
| DE5 | Stop switch | X2 Pin 13 |


If the motor direction of rotation is set to “counter direction of rotation”, the function of DE2 and DE3 is swapped.

The inputs are optically isolated and designed for 24V or 5V (see type code). The inputs can be deactivated via the parameter data (see 1.3). The level of all inputs can be read out via a Profibus word.

## 1.2 Profibus address

The two rotary switches on the TSP10-PBE are used to set the Profibus address. The motor current and the microstep factor are specified via the parameter data (see 1.3).



Higher value adresse (factor: 16)

The address is set in hexadecimal format. For an address up to 15, the left-hand rotary switch remains at zero. For higher addresses, the left-hand rotary switch position is multiplied by 16 and added to the right-hand value.

## 1.3 Parameter data

The settings for motor current, step factor (microstep factor), limit switch, homing, smoothing and standstill current reduction are specified via the Profibus user parameter data. Specifications are defined in the GSD file. Extracts from the GSD file are shown in *italics*.

Allgemein | Parameterzuordnung

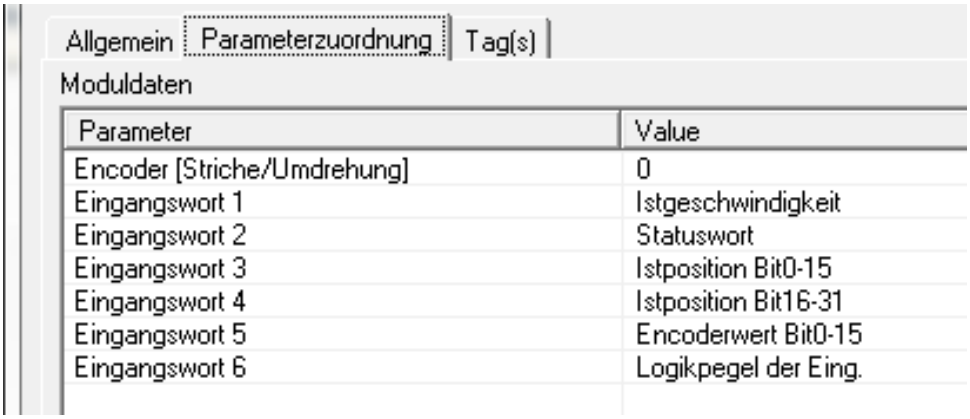
Modul Daten:

| Parameter | Value |
|---|---|
| Endschalter | Endschalter sind angeschlossen |
| Smoothing | ohne Smoothing |
| Stillstandsstromreduzierung | nach 100 ms |
| Motordrehrichtung | Standarddrehrichtung |
| Stopp-Schalter | High-Signal zum Anhalten |
| reduzierter Stillstandsstrom [%] | 50 |
| Motorstrom [mA eff] | 100 |
| Mikroschrittfaktor n*200 / Umdr. | 20 |
| Referenzfahrt | Istposition |
| DA1 | Bereit |
| DA2 | Aktiviert |
| DA3 | Ziel erreicht |
| DA4 | Fehler |
| Aktiviert | Aktiviert |

User Prm Daten:

| 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 | 013 | 014 | 015 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 8D | 32 | 00 | 64 | 14 | 23 | 01 | 02 | 03 | 04 | 02 |

The following illustration shows the default parameters of the input word module.




## 1.4 Micro-step factors (step size)

The microstep factor (byte 9) can be selected between 1 and 128 (200 and 25600 steps per revolution). A higher resolution offers smoother operation at low speeds.

```
ExtUserPrmData    = 1 "Micro step factor n*200 / revolution"
Unsigned8 20 1-128
EndExtUserPrmData
```

Example:
A drive rotates a spindle that generates a feed of 4 mm per revolution. The total travel is 700 mm - i.e. 175 revolutions. With a selected microstep factor of n = 20, a microstep resolution of 4000 steps per revolution is obtained. The position in µm (micrometres) can then be specified as the target position. The final position would therefore be 700,000.

## 1.5 Limit switch

If the limit switches (X2 pin 10 and 11) are not used, the parameter (byte 4 bit 0) must be set to zero so that the motor can move. The limit switches function as normally closed contacts and prevent the motor from moving when open.

```
PrmText      = 2
Text(0)      ="Limit switches not connected!"
Text(1)      ="Limit switches are connected."
EndPrmText
```

## 1.6 Smoothing

Smoother running can be achieved by switching on 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
```

## 1.7 Direction of motor rotation

In the standard setting, the motor rotates clockwise when looking at the shaft.

If ‘Counter-rotation direction’ is selected as the setting, the limit switch functions DE2 and DE3 are swapped.

```
PrmText                 = 8
Text(0)                 ="Default rotational direction"
Text(1)                 ="Reverse rotational direction"
EndPrmText
ExtUserPrmData          = 8 "Motor rotational direction"
Bit(6) 0 0-1
Prm_Text_Ref            = 8
EndExtUserPrmData
```

## 1.8 Stop switch polarity

The polarity of the stop input can be configured.

```
PrmText             = 9
Text(0)             ="Low level to stop"
Text(1)             ="High levelt o stop"
EndPrmText
ExtUserPrmData      = 9 "Stop switch"
Bit(7) 1 0-1
Prm_Text_Ref        = 9
EndExtUserPrmData
```

## 1.9 Motor current

The effective motor current can be set between 100 and 7000 mA (bytes 6 and 7). The parameterised value must never be higher than the permissible motor current, not even for a short time.

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

## 1.10 Standstill current reduction

The waiting time until the current is reduced can be parameterised in 8 stages (byte 4 bits 2-5). The motor current is specified as a percentage of the set value (byte 5) so that the heating of the motor is 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
```

## 1.11 Homing

The homing method can be specified via byte 10.

```
PrmText             = 5
Text(17)            ="Lower limit switch"
Text(18)            ="Upper limit switch"
Text(24)            ="Lower reference switch (Fahrtr. +)"
Text(25)            ="Upper reference switch (Fahrtr. +)"
Text(28)            ="Upper reference switch (Fahrtr. -)"
Text(29)            ="Lower reference switch (Fahrtr. -)"

Text(35)            ="Current position"
Text(250)           ="Lower mechanical limit"
Text(251)           ="Upper mechanical limit"
EndPrmText
ExtUserPrmData      = 7 "Homing"
Unsigned8 35 17-251
Prm Text Ref        = 5
EndExtUserPrmData
```

## 1.12 Digital outputs

Each output can be assigned an internal function or controlled directly via the Profibus command word.

```
PrmText           = 12
Text(0)           ="Not used"
Text(1)           ="Ready"
Text(2)           ="Enabled"
Text(3)           ="Target reached"
Text(4)           ="Error"
Text(100)         ="Profibus command word"
EndPrmText
ExtUserPrmData    = 24 "DA1"
Unsigned8 1 0-100
Prm_Text_Ref      = 12
EndExtUserPrmData
ExtUserPrmData    = 25 "DA2"
Unsigned8 2 0-100
Prm_Text_Ref      = 12
EndExtUserPrmData
ExtUserPrmData    = 26 "DA3"
Unsigned8 3 0-100
Prm_Text_Ref      = 12
EndExtUserPrmData
ExtUserPrmData    = 27 "DA4"
Unsigned8 4 0-100
Prm_Text_Ref      = 12
EndExtUserPrmData
ExtUserPrmData    = 28 "Enabled output"
Unsigned8 2 0-100
Prm_Text_Ref      = 12
EndExtUserPrmData
```

## 1.13 Encoder resolution

```
ExtUserPrmData    = 10 "Encoder [lines/rev.]"
Signed16 0 -10000-10000
EndExtUserPrmData
```

The setting of this value has the following meaning
0: the pulse counter value is passed on unchanged (raw data)
1: the counter value can be reset with ‘Counter Reset’ (command word Bit1)
500-10000: the counter value is converted to the set microstep factor and can be reset with ‘Counter Reset’

## 1.14 Configuration of the profilbus input

The assignment of the input words can be changed via the parameters of the module.

```
PrmText           = 10
Text(0)           ="Not used"
Text(1)           ="Actual velocity"
Text(2)           ="Status word"
Text(3)           ="Actual position bit0-15"
Text(4)           =" Actual position bit16-31"
Text(5)           ="Encoder counter bit0-15"
Text(6)           ="Encoder counter bit16-31"
Text(7)           ="Encoder velocity"
Text(8)           ="Inputs logic level"
EndPrmText
ExtUserPrmData    = 10 "Encoder [lines/rev.]"
Signed16 0 -10000-10000
EndExtUserPrmData
ExtUserPrmData    = 12 "Input word 1"
Unsigned8 1 0-8
Prm_Text_Ref      = 10
EndExtUserPrmData
ExtUserPrmData    = 13 "Input word 2"
Unsigned8 2 0-8
Prm_Text_Ref      = 10
EndExtUserPrmData
ExtUserPrmData    = 14 "Input word 3"
Unsigned8 3 0-8
Prm_Text_Ref      = 10
EndExtUserPrmData
ExtUserPrmData    = 15 "Input word 4"
Unsigned8 4 0-8
Prm_Text_Ref      = 10
EndExtUserPrmData
ExtUserPrmData    = 16 "Input word 5"
Unsigned8 5 0-8
Prm_Text_Ref      = 10
EndExtUserPrmData
ExtUserPrmData    = 17 "Input word 6"
Unsigned8 8 0-8
Prm_Text_Ref      = 10
EndExtUserPrmData
```

## 1.15 Configuration of the profibus output

The default assignment of the output words can be changed via the module parameters.

```
; Module 1 with 6 parameterizable output words
PrmText             = 11
Text(0)             ="Not used"
Text(1)             ="Velocity"
Text(2)             ="Start velocity"
Text(3)             ="Acceleration time [ms]"
Text(4)             ="Command word"
Text(5)             ="Target position bit0-15"
Text(6)             ="Target position bit16-31"
EndPrmText
ExtUserPrmData      = 18 "Output word 1"
Unsigned8 1 0-8
Prm_Text_Ref        = 11
EndExtUserPrmData
ExtUserPrmData      = 19 "Output word 2"
Unsigned8 2 0-8
Prm_Text_Ref        = 11
EndExtUserPrmData
ExtUserPrmData      = 20 "Output word 3"
Unsigned8 3 0-8
Prm_Text_Ref        = 11
EndExtUserPrmData
ExtUserPrmData      = 21 "Output word 4"
Unsigned8 4 0-8
Prm_Text_Ref        = 11
EndExtUserPrmData
ExtUserPrmData      = 22 "Output word 5"
Unsigned8 5 0-8
Prm_Text_Ref        = 11
EndExtUserPrmData
ExtUserPrmData      = 23 "Output word 6"
Unsigned8 6 0-8
Prm_Text_Ref        = 11
EndExtUserPrmData
```

## 1.16 Diagnostic data

In the user-specific diagnostics (EXT_DIAG), the revision number (last number) of the firmware is transmitted as a 16-bit value as status information.

Example: Firmware 2.1.60 User diagnostics:

Byte 1: 3 (Length)

Byte 2: 0 (High-Byte)

Byte 3: 60 (Low-Byte)

## 2 Modul 1 (Output words)

The individual output words for controlling the TSP10-PBE are listed in the table. The parts that are only used for positioning are coloured green. Values that are used in speed mode are coloured orange.

| Modul 1<br>6 Output words | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Velocity (rpm)** | Maximum speed in rpm | | | | | | | | | | | | | | | |
| 2 | **Initial velocity (rpm)** | Initial velocity of the acceleration ramp in rpm | | | | | | | | | | | | | | | |
| 3 | **Acceleration time** | Time of acceleration in ms | | | | | | | | | | | | | | | |
| 4 | **Command word** | | | | A | 4 | 3 | 2 | 1 | H | E | D | S | V | a | R | I |
| 5 | **Target position (32 bits integer)** | Low word | | | | | | | | | | | | | | | |
| 6 | | High word | | | | | | | | | | | | | | | |

The byte sequence is expected in big-endian. This means that the most significant byte must be sent first.

### 2.1 Command word

| Bit | | Meaning | Description |
|---|---|---|---|
| 0 | I | Sync Encoder | ↑ = Set actual position to encoder position |
| 1 | R | Counter Reset | 1 = Set current position at 0 |
| 2 | a | **Acceleration** | 0 = no ramp (jump)<br>1 = linear velocity change |
| 3 | V | Velocity Mode | 0 = Positioning<br>1 = Velocity mode |
| 4 | S | Motor-Start | 0 = Stop motor<br>1 = Start positioning |
| 5 | D | Direction | 0 = forward<br>1 = backward |
| 6 | E | Enable | 0 = Motor is without current<br>1 = Holding torque or torque active |
| 7 | H | Homing | 0 = Standard mode<br>1 = Homing |
| 8 | 1 | DA1 | Logic level of the output, if parameterised |
| 9 | 2 | DA2 | Logic level of the output, if parameterised |
| 10 | 3 | DA3 | Logic level of the output, if parameterised |
| 11 | 4 | DA4 | Logic level of the output, if parameterised |
| 12 | A | Activate | Logic level of the output, if parameterised |

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

The speed in revolutions per minute is entered in the first output word. If the motor is to jump to an initial speed at the start, the value can be entered in word 2.

## 2.3 Acceleration

The acceleration time is specified in word 3. The acceleration is calculated by dividing the set speed by the acceleration time. In velocity mode, 100 rpm is used as the speed for calculating the acceleration. The product of the time base and acceleration time gives the time for the speed ramp in milliseconds. If the set acceleration time is zero, the acceleration is set to 1 second.

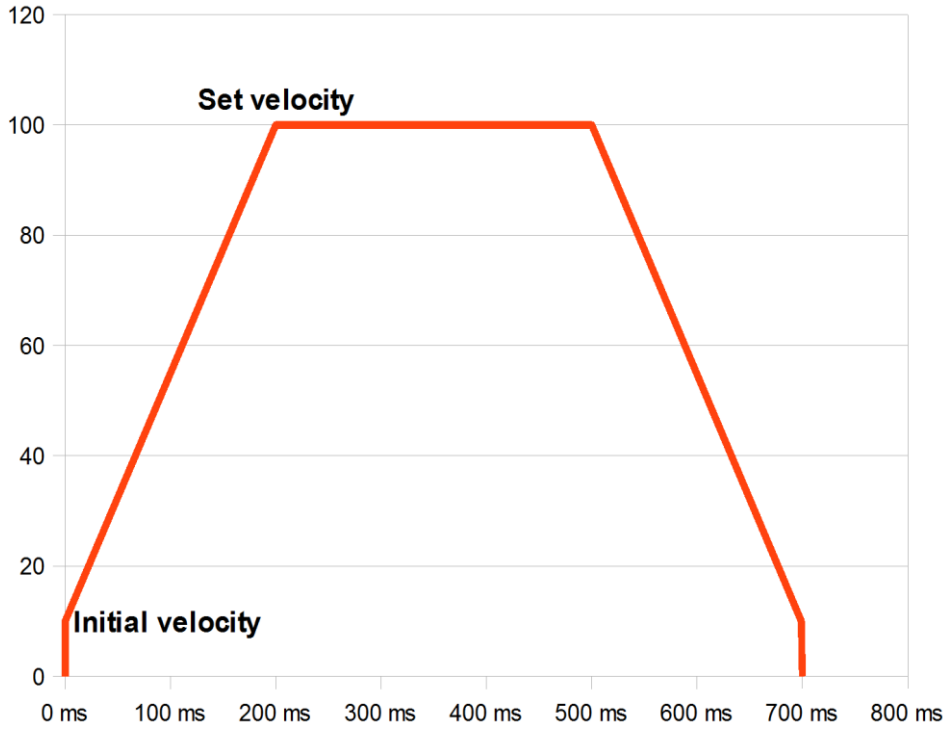



*Figure 5: Speed profile of a positioning, acceleration time = 200 ms*

## 2.4 Target position

The desired target position for the movement task is specified in output words 5 and 6. In order to be able to work with a high resolution and absolute positioning over a long distance, the target position is specified as a 32-bit integer value.

The reference point is always used for orientation during absolute positioning.

The actual position of the stepper is displayed in input words 3 and 4 with standard parameterisation. After reaching the reference point, the counter is set to 0 during the reference run and the counter status is set to referenced in the status word. As the position is undefined when the device is switched on, the counter is also set to 0 in this state to allow the user to move in both directions. By querying the current counter reading, the controller can determine the current position of the drive and utilise it within the user program.

The current counter reading does not need to be queried by the control system for positioning; the busy message can be used in the status word for this purpose. The module compares the actual position with the target position for each step and ends the movement job as soon as both match.

Once the stepper has reached the target position, the user programme preselects a new target position. Before the stepper starts running again, however, the motor start bit of the command word, which may still be active due to the previous run, must first be set to inactive and then active again.

It should also be noted that the stepper specifies the direction of travel itself in positioning mode. The corresponding command bit is therefore ineffective. The direction of travel selected by the stepper always results from the comparison of the actual and target positions.

The stop conditions (stop and limit switches) are constantly checked during the journey. They naturally have a higher priority, i.e. if the specified position value has not yet been reached and the corresponding switch is actuated, the motor is stopped immediately.

In the event of limit switch or alarm stops or stops due to reaching the target position, the next motor start only takes place when the motor start bit is first reset and then set again.

## 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:

Output words:

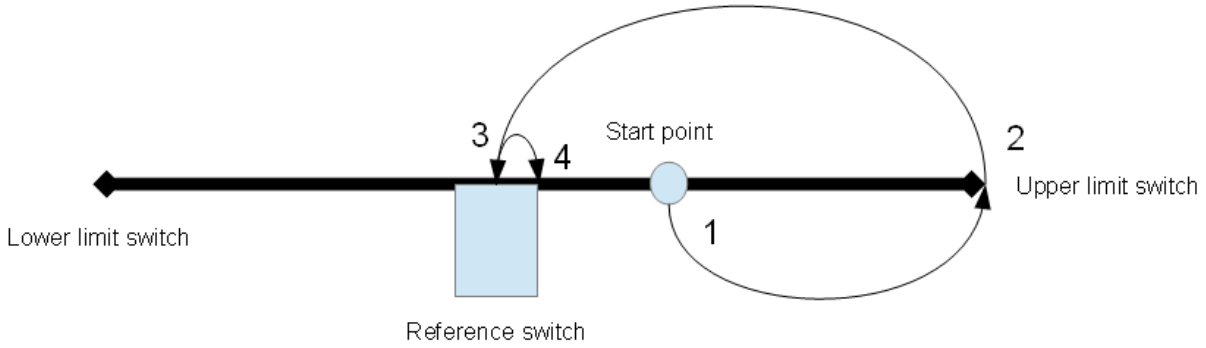
| | |
|---|---|
| **Velocity** | 0x64 |
| **Command word** | 0x00C0 |
| **Target position** | Any |

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

| | |
|---|---|
| **Current velocity** | 0x64 |
| **Status bytes** | 0x1000 |

Procedure of homing:

| | |
|---|---|
| **1** | Start: Position counter increments, velocity as set. |
| **2** | Upper limit switch responds. Direction inverts, position counter decrements. Velocity as set.<br>Reference point is below the start position. |
| **3** | Reference switch responds. Direction inverts. Position counter value increments.<br>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.<br>Set position counter at 0 (zero).<br>Motor has arrived at the reference point. |




## 3 Modul 2 (Input words)

| **Modul 2<br>6 Input bytes** | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Current velocity (rpm)** | Step speed in rpm | | | | | | | | | | | | | | |
| 2 | **Status word** | | E | U | C | A | | | | B | I | V | D | ↑ | R | ↓ | S |
| 3 | **Current position<br>(32 bits integer)** | Low word | | | | | | | | | | | | | | |
| 4 | | High word | | | | | | | | | | | | | | |
| 5 | **Encoder position** | Counter value Low word | | | | | | | | | | | | | | |
| | **Encoder position** | Counter value High word | | | | | | | | | | | | | | |
| | **Encoder velocity** | Encoder velocity in rpm | | | | | | | | | | | | | | |
| 6 | **Logikpegel der Eingänge<br>DE1 – DE10 und Freigabe** | | | | | | F | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |

The numbering corresponds to the standard parameterisation.
The byte sequence is expected in big-endian. This means that the most significant byte must be sent first.

## 3.1 Status word

In order to be able to monitor the status of the stepper motor drives in the control unit, the status word of the stepper motor drive is read. The table below shows which displays can be analysed within the status word.

| Bit | | Meaning | Description |
|---|---|---|---|
| 0 | S | Stop switch | 0 = Stop switch not enabled<br>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)<br>1 = backward |
| 5 | V | Final velocity | 0 = not reached<br>1 = reached |
| 6 | I | In position | 0 = Encoder value deviation<br>1 = Encoder value in the range of the current position |
| 7 | B | Busy | 0 = Travel command terminated<br>1 = Travel command enabled |
| 8 | | | |
| 9 | | | |
| 10 | | | |
| 11 | A | Emergency shutdown/alarm | 1 = Alarm or emergency stop enabled |
| 12 | C | Counter status | 1 = Counter reading is referenced |
| 13 | U | Bus power | DC link voltage ok |
| 14 | E | Error | 1 = Error |

## 3.2 Current position and velocity

The values for the current position and speed have the same format as the default values in the output words.

## 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. Overcurrent
3. Overtemperature

**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 has been created to end the alarm status without switching off. A **pseudo code** for the speed output word was introduced for this purpose. It is the code **0xAA55**. This code does not occur in normal operation as the maximum speed is set to 3000. If the stepper reads the pseudo code, it checks again whether the conditions for the alarm status are still present. If the conditions are no longer present, it switches off the alarm state and returns to the operating state once the pseudo code **has been completed**. It should be noted here that the pseudo code output 0xAA55 is **actually exited**, as only then is the normal operating state resumed.