

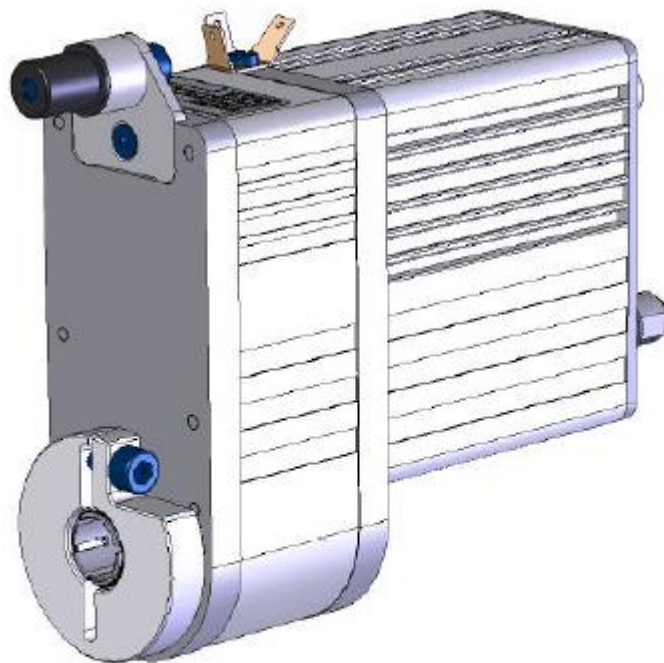
---

# User manual

---

magnetic absolute  
positioning drive

## AG02



<b>1</b>	<b>General remarks .....</b>	<b>5</b>
1.1	SYMBOLS AND THEIR MEANING .....	5
1.2	DOCUMENTATION .....	5
<b>2</b>	<b>Functional description .....</b>	<b>5</b>
2.1	SYSTEM STATUS WORD.....	5
2.2	OPERATING MODES.....	7
2.2.1	<i>Positioning mode</i> .....	7
2.2.1.1	Loop positioning.....	8
2.2.1.2	Inching operation .....	9
2.2.1.3	Flow chart 'Positioning mode' via standard protocol .....	10
2.2.2	<i>Velocity mode</i> .....	11
2.2.2.1	Flow chart 'Velocity mode' via standard protocol.....	12
<b>3</b>	<b>Calibration.....</b>	<b>12</b>
<b>4</b>	<b>External gear unit.....</b>	<b>13</b>
<b>5</b>	<b>Warnings / Faults .....</b>	<b>13</b>
5.1	WARNINGS .....	13
5.2	FAULTS.....	14
5.2.1	<i>Error codes</i> .....	14
<b>6</b>	<b>Standard protocol (RS232 / RS485) .....</b>	<b>14</b>
6.1	ERROR NUMBER CODING: .....	18
<b>7</b>	<b>Trouble shooting .....</b>	<b>19</b>
<b>8</b>	<b>Parameter description.....</b>	<b>19</b>
<b>9</b>	<b>Communication via Profibus – DP (optional).....</b>	<b>23</b>
9.1	GENERAL REMARKS .....	23
9.2	INTERFACE .....	23
9.3	DATA EXCHANGE.....	24
9.4	TELEGRAM SETUP (DATA EXCHANGE) .....	24
9.4.1	<i>Parameter Process Data Object</i> .....	24
9.4.2	<i>Parameter identification value (PKW)</i> .....	25
9.4.2.1	Parameter identification (PKE) .....	25
9.4.2.2	Sub-index (IND) .....	26
9.4.2.3	Parameter value (PWE).....	26
9.4.3	<i>Process data (PZD)</i> .....	27
9.5	FUNCTIONAL DESCRIPTION OF THE CONTROL UNITS.....	28
9.5.1	<i>Control word in positioning mode (master ⇒ slave)</i> .....	28
9.5.2	<i>Status word: Positioning mode (slave ⇒ master)</i> .....	29
9.5.3	<i>Flow chart: Operating mode: Positioning mode</i> .....	31
9.5.4	<i>Control word: Velocity mode (master ⇒ slave)</i> .....	32
9.5.5	<i>Status word: Velocity mode (slave ⇒ master)</i> .....	33
9.5.6	<i>Flow chart: Operating mode: Velocity mode</i> .....	34
9.6	PARAMETERIZATION VIA PROFIBUS .....	34
9.6.1	<i>Example: read parameter</i> .....	37
9.6.2	<i>Example: Write parameter</i> .....	38
9.7	DIAGNOSIS .....	39
9.8	PROFIBUS ADDRESS SETUP:.....	40
9.9	DEVICE DATA BASE AND CONFIGURATION .....	40
<b>10</b>	<b>Communication via CAN bus (optional).....</b>	<b>40</b>
10.1	GENERAL REMARKS .....	40
10.1.1	<i>Interface</i> .....	40
10.2	CANOPEN PROTOCOL .....	40
10.2.1	<i>Telegram setup</i> .....	40
10.2.2	<i>Network management (NMT)</i> .....	41
10.2.2.1	State Diagramm .....	42
10.2.2.2	NMT Status 'INITIALISATION' .....	42
10.2.2.3	NMT Status 'PRE-OPERATIONAL' .....	42
10.2.2.4	NMT Status 'OPERATIONAL' .....	42
10.2.2.5	NMT Status 'STOPPED' .....	43
10.2.2.6	Switching between communication states .....	43
10.2.3	<i>SYNC object</i> .....	43
10.2.4	<i>Process Data Objects (PDOs)</i> .....	43

10.2.5	Transmit PDOs .....	44
10.2.5.1	1 <sup>st</sup> Transmit PDO (TPDO1) .....	44
10.2.5.2	3 <sup>rd</sup> Transmit PDO (TPDO3) .....	44
10.2.5.3	4 <sup>th</sup> Transmit PDO (TPDO4) .....	45
10.2.5.4	Transfer types of the Transmit PDOs .....	45
10.2.6	Receive PDOs .....	45
10.2.6.1	1 <sup>st</sup> Receive PDO (RPDO1) .....	45
10.2.6.2	3 <sup>rd</sup> Receive PDO (RPDO3) .....	46
10.2.6.3	4 <sup>th</sup> Receive PDO (RPDO4) .....	46
10.2.6.4	Transfer types of the Receive PDOs .....	47
10.2.7	Service Data Objects (SDOs) .....	47
10.2.7.1	Error code .....	48
10.2.8	Example: Parameterization .....	49
10.2.8.1	Example: Read parameter .....	49
10.2.8.2	Example: Write parameter .....	49
10.2.9	Emergency Object (EMCY) .....	50
10.2.9.1	Error Code .....	51
10.2.10	Heartbeat protocol .....	52
10.2.11	Node Guarding .....	52
10.3	STATE MACHINE .....	53
10.4	STATUS WORD .....	55
10.5	CONTROL WORD .....	57
10.6	FLOWCHART OF THE PROFILE POSITION MODE (POSITIONING MODE) .....	59
10.7	FLOWCHART OF THE PROFILE VELOCITY MODE OPERATION MODE (VELOCITY MODE) .....	60
10.8	EXAMPLES .....	61
10.8.1	Example of the Profile Position Mode (positioning mode) .....	61
10.8.2	Example of the Profile Velocity Mode (velocity mode) .....	61
10.9	OVERVIEW OF CANOPEN IDENTIFIERS .....	62
10.10	SETTING THE CAN ADDRESS AND CAN BAUD RATE .....	62
10.11	EDS FILE .....	62
10.12	DIRECTORY OF OBJECTS .....	62
10.12.1	Overview of objects .....	63
10.12.2	Description of objects .....	64
10.12.2.1	1000 <sub>h</sub> : Device Type .....	64
10.12.2.2	1001 <sub>h</sub> : Error Register .....	65
10.12.2.3	1002 <sub>h</sub> : Manufacturer Status Register .....	65
10.12.2.4	1003 <sub>h</sub> : Pre-defined Error Field .....	65
10.12.2.5	1005 <sub>h</sub> : COB-ID Sync Message .....	66
10.12.2.6	100A <sub>h</sub> : Manufacturer Software Version .....	67
10.12.2.7	100C <sub>h</sub> : Guard Time .....	67
10.12.2.8	100D <sub>h</sub> : Life Time Factor .....	67
10.12.2.9	1011 <sub>h</sub> : Restore Default Parameters .....	68
10.12.2.10	1014 <sub>h</sub> : COB ID Emergency Message .....	69
10.12.2.11	1017 <sub>h</sub> : Producer Heartbeat Time .....	69
10.12.2.12	1018 <sub>h</sub> : Identity Object .....	70
10.12.2.13	1200 <sub>h</sub> : Server SDO Parameter .....	70
10.12.2.14	1400 <sub>h</sub> : 1 <sup>st</sup> Receive PDO Parameter .....	71
10.12.2.15	1402 <sub>h</sub> : 3 <sup>rd</sup> Receive PDO Parameter .....	72
10.12.2.16	1403 <sub>h</sub> : 4 <sup>th</sup> Receive PDO Parameter .....	74
10.12.2.17	1600 <sub>h</sub> : 1 <sup>st</sup> Receive PDO Mapping Parameter .....	75
10.12.2.18	1602 <sub>h</sub> : 3 <sup>rd</sup> Receive PDO Mapping Parameter .....	75
10.12.2.19	1603 <sub>h</sub> : 4 <sup>th</sup> Receive PDO Mapping Parameter .....	76
10.12.2.20	1800 <sub>h</sub> : 1 <sup>st</sup> Transmit PDO Parameter .....	77
10.12.2.21	1802 <sub>h</sub> : 3 <sup>rd</sup> Transmit PDO Parameter .....	79
10.12.2.22	1803 <sub>h</sub> : 4 <sup>th</sup> Transmit PDO Parameter .....	80
10.12.2.23	1A00 <sub>h</sub> : 1 <sup>st</sup> Transmit PDO Mapping Parameter .....	82
10.12.2.24	1A02 <sub>h</sub> : 3 <sup>rd</sup> Transmit PDO Mapping Parameter .....	82
10.12.2.25	1A03 <sub>h</sub> : 4 <sup>th</sup> Transmit PDO Mapping Parameter .....	83
10.12.2.26	2100 <sub>h</sub> : CAN baud rate .....	84
10.12.2.27	2101 <sub>h</sub> : Node-ID .....	85



10.12.2.28	2102 <sub>h</sub> : Gear reduction.....	85
10.12.2.29	2410 <sub>h</sub> : Motor Parameter Set.....	85
10.12.2.30	2412 <sub>h</sub> : Spindle Pitch Set.....	87
10.12.2.31	2413 <sub>h</sub> : Pos Type Set.....	88
10.12.2.32	2415 <sub>h</sub> : Delta Jog Set.....	88
10.12.2.33	6040 <sub>h</sub> : Control word.....	88
10.12.2.34	6041 <sub>h</sub> : Status word.....	89
10.12.2.35	6060 <sub>h</sub> : Modes of Operation.....	89
10.12.2.36	6064 <sub>h</sub> : Position Actual Value.....	89
10.12.2.37	6067 <sub>h</sub> : Position Window.....	90
10.12.2.38	606C <sub>h</sub> : Velocity Actual Value.....	90
10.12.2.39	607A <sub>h</sub> : Target Position.....	90
10.12.2.40	607C <sub>h</sub> : Calibration Value.....	91
10.12.2.41	607D <sub>h</sub> : Software Position Limit.....	91
10.12.2.42	607E <sub>h</sub> : Polarity.....	92
10.12.2.43	6091 <sub>h</sub> : Gear Ratio.....	92
10.12.2.44	60FF <sub>h</sub> : Target Velocity.....	93

## 1 General remarks

This user manual is valid with firmware version 2.00 or higher!

### 1.1 Symbols and their meaning



This symbol precedes passages in the text that should be read with particular care to ensure flawless use of the AG02.



This symbol precedes passages in the text that contain additional important information.

### 1.2 Documentation

This user manual is valid for the AG02 absolute positioning drive and is intended to provide you with the necessary information on programming and activation of the AG02 positioning drive.



**Please refer to the installation instruction for information on mechanical mounting, electrical connection, general safety instructions, correct usage as well as commissioning of the positioning drive.**

## 2 Functional description

This chapter describes the operating cycles, functions and states required for the operation of the positioning drive.

The standard protocol of the AG02 (RS232 or RS485 interfaces) is taken as a basis for this chapter.

### 2.1 System status word

The system status word of the AG02 consists of 2 bytes and reflects the state of the AG02. The system status word can be read out as a hexadecimal value via serial interface (RS232/RS485) (see chapter 6: Standard protocol: 'R' command).

High- Byte								Low- Byte							
Bit – number															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0
2				9				4				8			

Fig. 1: System status word format

Example (grey background):

binary: ⇒ 0010 1001 0100 1000

hex: ⇒ 2 9 4 8

The table below informs about the meaning of the individual bits of the status word:

Bit	State	Description
Bit 0	'1'	<b>Positioning mode: Limit switch 1:</b> Limit switch 1 active. Travelling is possible only in limit switch 2 direction in inching operation.
	'0'	Limit switch 1 inactive.
	'0'	<b>Velocity mode: no meaning</b>
Bit 1	'1'	<b>Positioning mode: Limit switch 2:</b> Limit switch 2 active: Travelling is possible only in limit switch 1 direction in inching operation.
	'0'	Limit switch 2 inactive
	'0'	<b>Velocity mode: no meaning</b>
Bit 2	'1'	<b>Calibration switch:</b> Calibration switch active
	'0'	Calibration switch inactive
Bit 3	'1'	<b>Positioning mode: In position</b> Actual position is within the positioning window of the programmed target value.
	'0'	Actual position is beyond the positioning window of the programmed target value.
	'1'	<b>Velocity mode: In position</b> Actual velocity is within the specified tolerance window of the target speed
	'0'	Actual speed is outside the specified tolerance window.
Bit 4	'1'	<b>Drive travels:</b> Drive travels
	'0'	Drive stands still (velocity <2 rev./min)
Bit 5	'1'	<b>Positioning mode: upper limit:</b> Actual position is above the programmed limit. Travelling is possible only in negative direction in inching operation.
	'0'	Actual position is below the programmed limit.
	'0'	<b>Velocity mode: no meaning</b>
Bit 6	'1'	<b>Positioning mode: lower limit:</b> Actual position is below the programmed limit. Travelling is possible only in positive direction in inching operation.
	'0'	Actual position is above the programmed limit.
	'0'	<b>Velocity mode: no meaning</b>
Bit 7	'1'	<b>Driver state:</b> Motor is enabled
	'0'	Motor in control
Bit 8	'1'	<b>Fault:</b> The AG02 has switched to fault. The cause of the fault must be cleared and acknowledged via 'S11103' command.
	'0'	For the fault cause see chapter 5.2 No fault present
Bit 9	'1'	<b>Positioning mode: loop travel:</b> If travel direction unequal start direction (with loop travel)
	'0'	If travel direction equal start direction
	'0'	<b>Velocity mode: no meaning</b>

Bit	State	Description
Bit 10	'1'	<b>Status of the enable input</b> Enable input deactivated Gear travel is disabled!
	'0'	Enable input activated
Bit 11	'1'	<b>Ready-to-travel:</b> Not ready-to-travel
	'0'	Ready-to-travel: <ul style="list-style-type: none"> <li>• AG02 not in fault state</li> <li>• No active positioning</li> <li>• no active limit switches (only positioning mode)</li> <li>• Enable input active</li> <li>• actual position within limits (only positioning mode)</li> </ul>
Bit 12	'1'	<b>Battery voltage:</b> Battery voltage too low for absolute encoder. Have the battery replaced at SIKO's within half a year after notice of error bit!
	'0'	Battery voltage OK.
Bit 13	'1'	<b>Motor current:</b> Motor current beyond permissible range. If this state lasts for more than 10 sec., the AG02 switches to fault.
	'0'	Motor current within permissible range.
Bit 14	'1'	<b>Positioning mode: status</b> Positioning active in positioning mode.
	'0'	Positioning inactive.
	'0'	<b>Velocity mode: no meaning</b>
Bit 15	'1'	<b>Lag error:</b> Lag error ⇒ AG02 cannot reach the preset speed due to too high load. This state should be avoided! (see also chapter 6: ⇒ 'N' command / 'O' command) Remedy: reduce programmed speed!
	'0'	No lag error ⇒ actual speed corresponds with required speed.

Table 1: System status word

## 2.2 Operating modes

The following operating modes are available: positioning mode and velocity mode. In the positioning mode there is the additional option of travelling with inching operation.

### 2.2.1 Positioning mode

Parameter no 20 'Operating mode' is programmed to positioning mode (see chapter 8: Parameter description).

In the positioning mode, positioning to the target value is effected by means of a ramp function (see fig. 2), which is calculated on the basis of the actual position as well as the programmed controller parameters P (proportional factor), I (integral factor), D (differential factor), acceleration and speed (see chapter 8: Parameter description).

After the 'Start' command, the AG02 accelerates to the preset speed with the programmed acceleration rate.

The measure of deceleration to reach the target value is also defined by the parameter 'a-pos'.

If the actual position is within the programmed window (see chapter 8: parameter no. 10), this will be signalled in the system status word (bit 3).

Independent of the size of the tolerance window, the AG02 always tries to drive exactly to the target position. Control to the target position remains active until the motor has been enabled (see chapter 6: *Standard protocol: command 'P'*) or until a new positioning job has been started.

Changing controller parameters during a positioning process does not influence the current positioning operation.

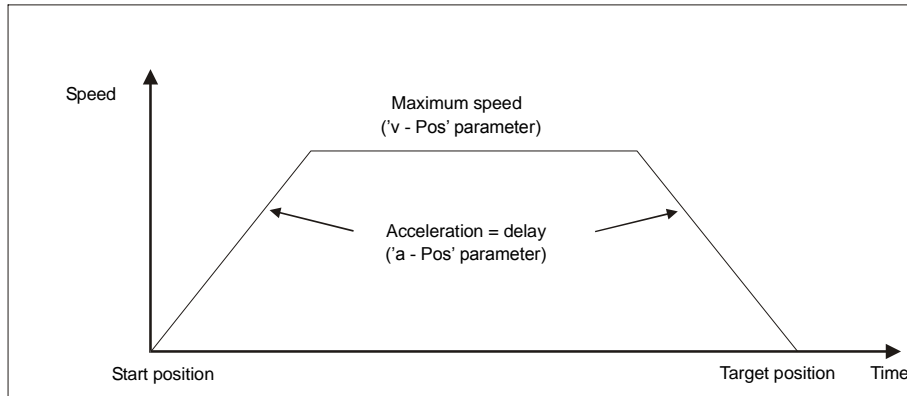


Fig. 2: Ramp travel in positioning mode (direct)

The following conditions must be met for enabling start of positioning via the 'M' command of the standard protocol:

- the AG02 must not be switched to fault (*system status word bit 8 = '0'*)
- no active limit switches (*system status word bits 1 + 2 = '0'*)
- actual position is within the programmed limiting values. (*system status word bits 5 + 6 = '0'*)
- no active travel job present (*system status word bit 14 = 0*)
- Enable input active (*system status word bit 10 = '0'*)

If these conditions are met, this will be signalled in the system status word by bit 11 = '0'.

### 2.2.1.1 Loop positioning

If the AG02 is operated on a spindle or an additional drive, the spindle or external drive play can be compensated by means of loop positioning.

In this case, travelling to the target value is always from the same direction.

This travel direction can be defined via the parameter no. 19 'Pos- Art' (see chapter 8: *Parameter description*).

#### Example:

Assumption: The direction of travelling for reaching any target position is positive (see chapter 8: *Parameter no. 19 'Pos Art' = loop*)

- Case 1  $\Rightarrow$  new position is greater than actual position:  
Direct travel to target position



- Case 2 ⇒ new position is smaller than actual position:

The AG02 travels back by half a revolution (bit 9 in the system status word signals travel direction unequal start direction); afterwards, travelling to the target value is in positive direction.

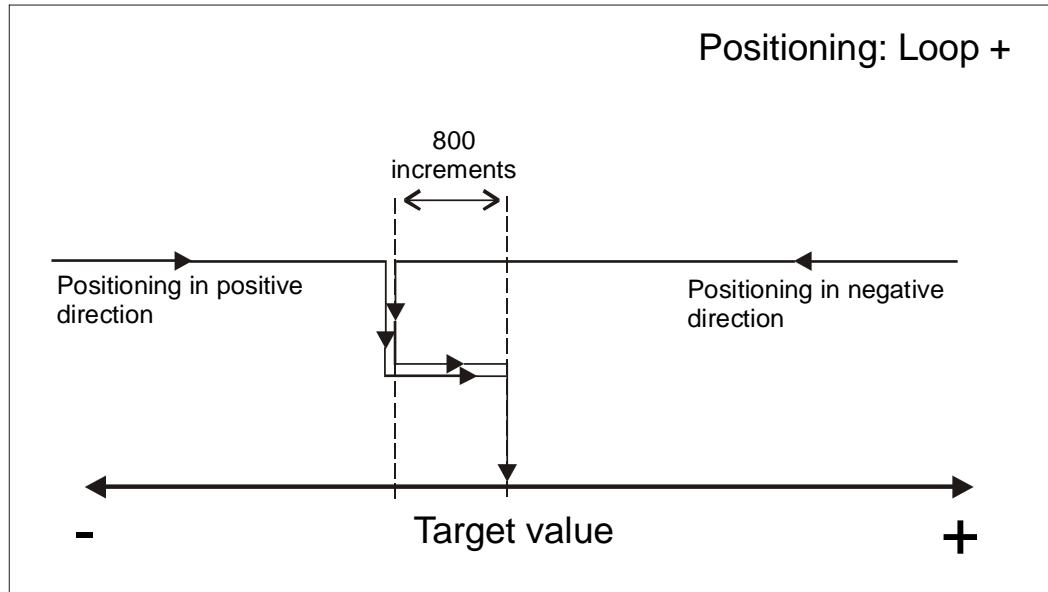


Fig. 3: Loop+ positioning

### 2.2.1.2 Inching operation

Inching operation is enabled in the 'Positioning mode' only. Acceleration and speed in the inching operation can be programmed via parameter no. 8 'a- Inch' and parameter no. 9 'v- Inch'.



**There is no compensation for spindle play (loop positioning) in this travel type!**

There are two options for travelling with inching operation:

- **Inching operation 1**

Inching operation 1 is started via the 'Y' command of the standard protocol. The drive travels once from the current position by the position 'Delta Inch'. The travel direction is positive or negative depending on the arithmetical sign (*parameter no. 17 'Delta- Inch'*).

- Parameter 'Delta Inch' < 0: negative travel direction
- Parameter 'Delta Inch' > 0: positive travel direction

If the parameter no. 13 'Spindle pitch' is programmed to zero, then there is an incremental travelling way. If the 'spindle pitch' is unequal zero, then the information of the 'Delta Inch' parameter refers to the travel distance in 1/100 mm.

Inching operation can be cancelled any time via the 'N' or 'O' commands.

Reaching the target position will be signalled in the system status word (bit 3).

• **Inching operation 2**

Inching operation 2 is started by sending ‘;’ (2C<sub>hex</sub>) or ‘.’ (2E<sub>hex</sub>), respectively. The drive travels with the programmed speed as long as the ‘;’ character or ‘.’ is sent permanently (*Interval between the characters sent must be <100 ms*). The drive stops as soon as the characters ‘;’ or ‘.’, respectively are no longer sent. Stop behaviour of inching operation 2 can be parameterized in different ways. In the default setting, the motor is stopped abruptly at the end with maximum delay. Optionally, stopping is also possible via parameterized inching acceleration. Selection may be done either via Profibus with the parameter 1021dez or via the serial interface with the commands “T3” or “T4”, respectively. The setting is stored non-volatily.

- ‘;’ character ⇒ positive travel direction
- ‘.’ character ⇒ negative travel direction

The following conditions must be met for enabling the start of inching operations 1 and 2 via standard protocol:

- The AG02 must not be switched to fault (*system status word bit 8 = ‘0’*)
- No active travel job present (*system status word bit 14 = ‘0’*)
- Enable input active (*system status word bit 10 = ‘0’*)



**If the actual position is outside the programmed limiting values or a limit switch is active, then travelling from this position to the respective direction is possible by means of inching operations 1 or 2!**

**2.2.1.3 Flow chart ‘Positioning mode’ via standard protocol**

The flow chart below shows the control of positioning in the positioning mode via standard protocol (*see chapter 6: Standard protocol*).

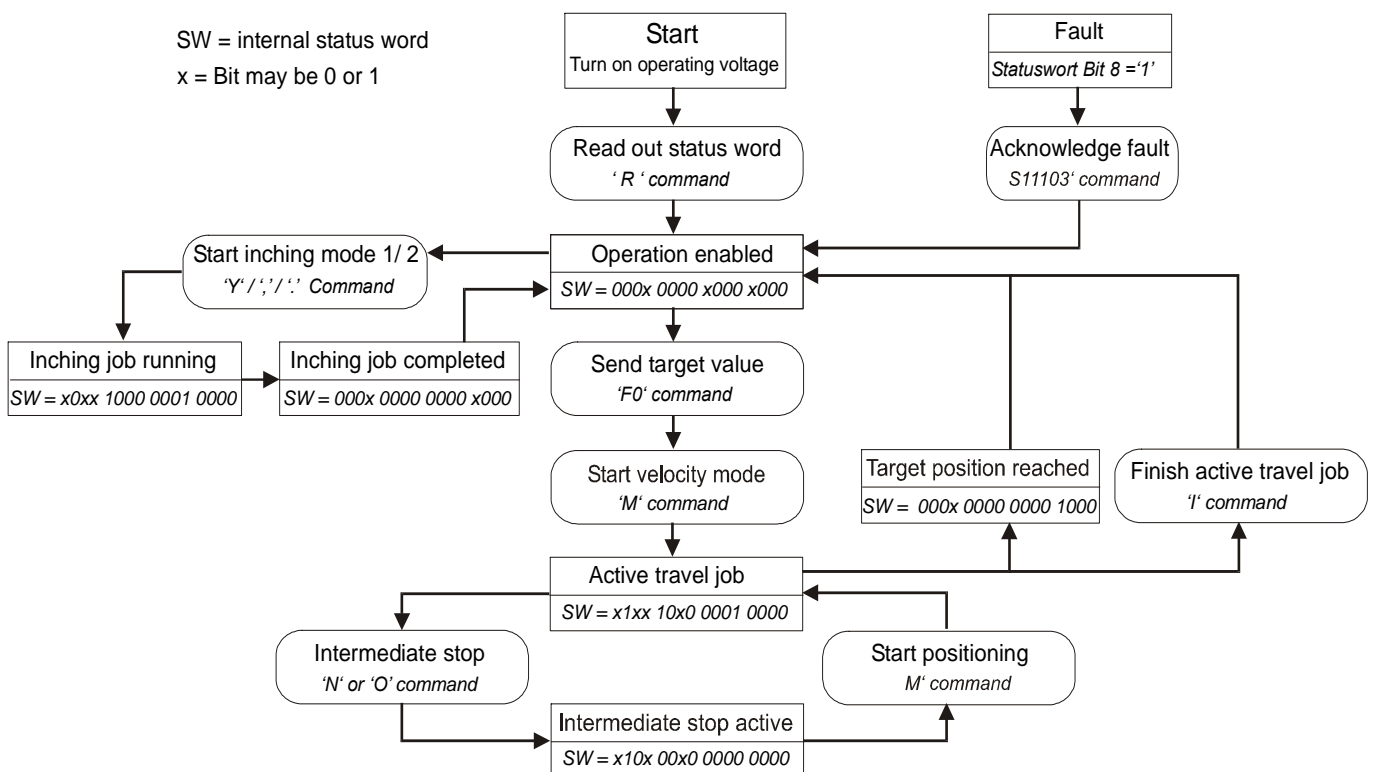


Fig. 4: Flow chart of positioning mode

## 2.2.2 Velocity mode

Parameter no 20 'operating mode' is programmed to velocity mode (see *chapter 8: Parameter description*).

In the velocity mode the AG02 accelerates after the 'Start' command to the preset speed with pre-programmed acceleration rate and maintains this velocity until it is stopped via 'Stop' command or a new target velocity is preset.

The speed is adjusted immediately to the new value when the rotational target speed is changed.

The arithmetical sign of the target value determines the travel direction in the velocity mode (see *chapter 8: Parameter description*).

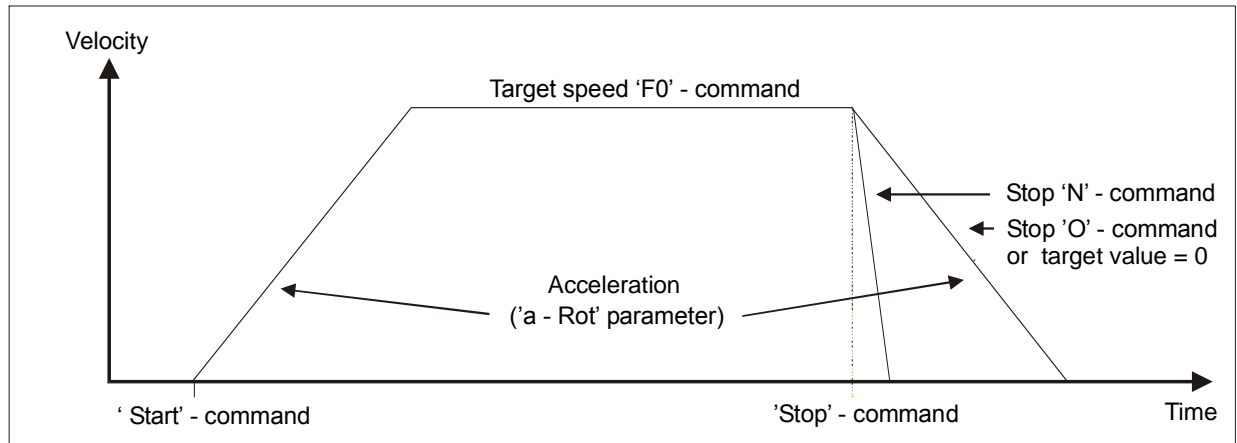


Fig. 5: Ramp velocity mode

The following conditions must be met for enabling start of velocity mode via the 'M' command of the standard protocol:

- The AG02 must not be switched to fault (*system status word bit 8 = '0'*)
- no active travel job present (*system status word bit 14 = '0'*)
- Enable input active (*system status word bit 10 = '0'*)

If these conditions are met, this will be signalled in the system status word by bit 11 = '0'.



**Limit switches as well as the upper and lower limits are deactivated in this Operating mode!**

### 2.2.2.1 Flow chart 'Velocity mode' via standard protocol

The flow chart below shows the control in the velocity mode via standard protocol (see chapter 6: Standard protocol).

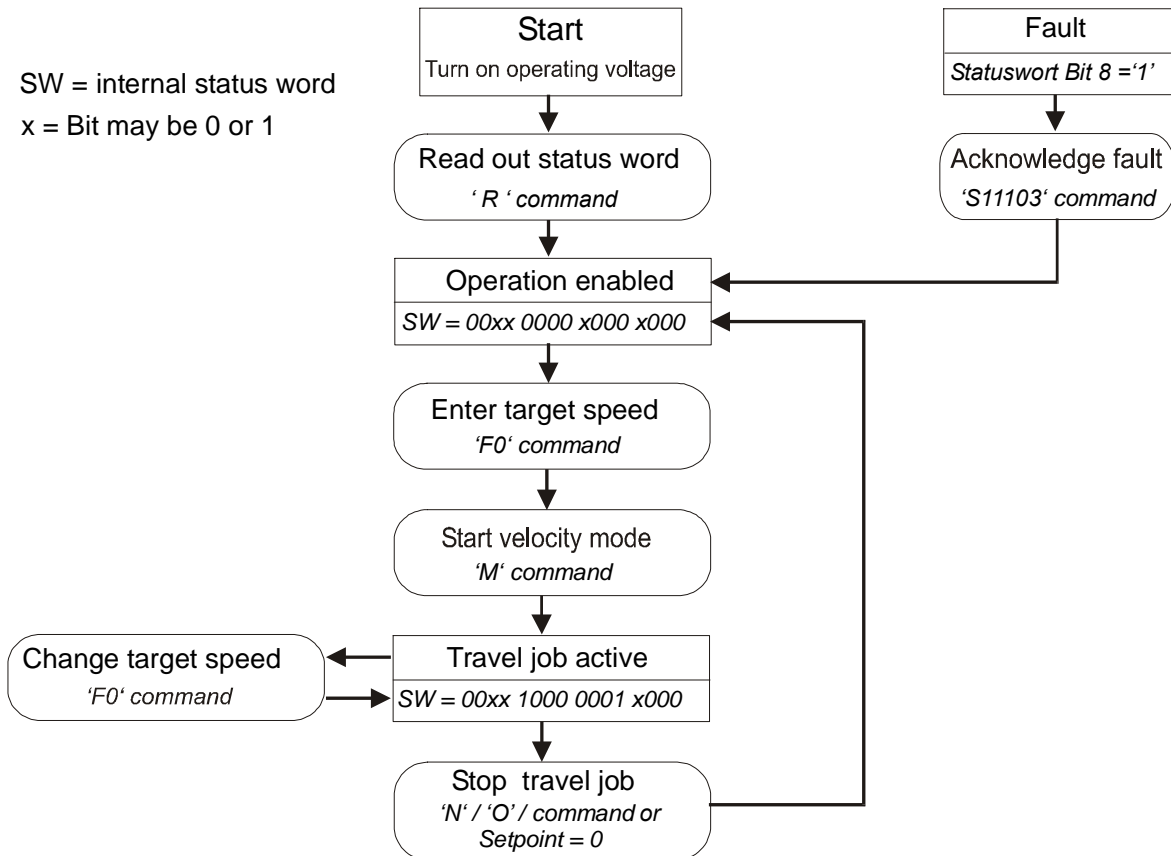


Fig. 6: Flow chart: Velocity mode

## 3 Calibration

Since the measuring system is an absolute system, calibration is necessary only once at start-up. During calibration, the position value of the AG02 is set to the programmed calibration value (see chapter 8: Parameter description (status word bit 8 = '0' parameter no. 14)).

There are two ways of calibrating the measuring system:

- Interface:  
Writing a value on the calibration value parameter (parameter no. 14) results in taking over of this value as the absolute position for the AG02.

Furthermore, the following commands can be used for calibration:

- standard protocol (see chapter 6: Standard protocol ⇒ 'S'- command)
- Profibus (see chapter 9.6: Parameterization via Profibus ⇒ param. no. 970)
- CAN-Bus (see chapter 10.12.2: Description of objects ⇒ Object 1011<sub>n</sub>)
- Actuating the external calibration switch on the 12-pin connector (see installation instruction ⇒ Electrical connection).



**Calibration is only possible when no travel job is active!**

## 4 External gear unit

If an external gear unit is used, a factor can be programmed via the parameter no. 11 'ü - Numerator' (numerator) and parameter no. 12 'ü - Denominator' (denominator), in order to include the gear ratio in position sensing (see also chapter 2.2.1.1).

Example (see fig. 7):

The AG02 is operated on a gear with transfer reduction of 5:1. For this purpose, the parameters 'ü-Numerator' and 'ü-Denominator' must be programmed as follows:

- Parameter 'ü - Numerator' :        5
- Parameter 'ü - Denominator' :     1

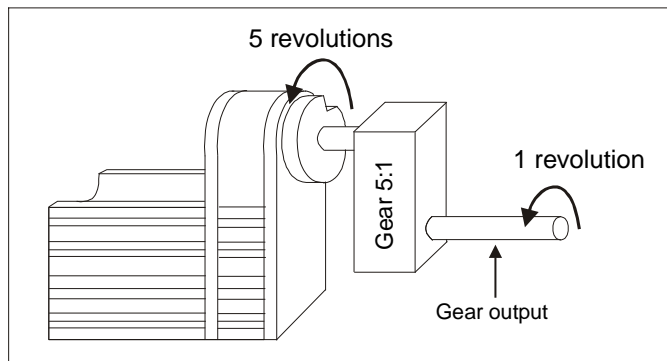


Fig. 7: External gear unit

Input of an odd gear reduction value is possible according to the following example:

Transfer reduction = 3.78

- Parameter 'ü - Numerator':        378
- Parameter 'ü - Denominator':     100

## 5 Warnings / Faults

The AG02 distinguishes between warnings and faults:

### 5.1 Warnings

Warnings do not influence the operation of the positioning drive. Warnings disappear after removing their causes.

Possible warnings:

- Battery voltage for absolute encoder is below limit  $\Rightarrow$  exchange battery within the next 6 months (see chapter 2.1: system status word  $\Rightarrow$  bit 12).
- Motor current beyond the permissible range (see chapter 2.1: System status word  $\Rightarrow$  bit 13)  
 $\Rightarrow$  if this condition lasts longer than 10 seconds, the AG02 will switch to fault.
- Lag error present (see chapter 2.1: System status word  $\Rightarrow$  bit 15)  $\Rightarrow$  AG02 cannot reach programmed speed because load is too high.

## 5.2 Faults

Faults induce an immediate stop of the positioning drive. An LED blinking red signalizes the fault status (see *installation instruction: fig. 7*). Based on the blink code, the cause of the fault can be tracked down (see *tab. 2: Error codes*). Furthermore, an existing fault can be recognized via bit 8 of the system status word. Faults must be acknowledged after clearing the faults ('S11103' command).

The error messages are entered in the error buffer in the order of their detection.

The last 10 fault messages are displayed when the fault buffer is full.

The fault buffer can be read out via the 'J' command (see *chapter 6: Standard protocol* ⇒ 'J'-command).

The cause of an error can be tracked down with the help of the error codes (see *table 2: Error codes*).

The error buffer is stored in the EEPROM and can be deleted via the 'S' command (see *chapter 6: Standard protocol*).

### 5.2.1 Error codes

The following table lists possible error causes:

Value error buffer		Blink code red LED	Error description
ASCII	HEX		
A	41h	blinking 1 time	supply voltage too low
B	42h	blinking 2 times	blocked drive shaft
C	43h	blinking 3 times	motor current too high
D	44h	blinking 4 times	output stage temperature too high
E	45h	blinking 5 times	error when reading the absolute position
F	46h	blinking 6 times	communication error with CAN. AG02 has switched to 'bus off' in the node state. Only with operation using the CAN bus!!!

Table 2: Error codes

## 6 Standard protocol (RS232 / RS485)

This chapter describes control or parameterization, respectively, of the AG02 via standard protocol of the serial interface (RS232 or RS485, respectively). If the AG02 is operated via RS485, a level transducer is required (RS485 ⇒ RS232), e. g. from Spectra company, type: I-7520 or appropriate PC hardware.

No additional devices must be connected to the RS485 interface since the standard protocol is not bus-compatible.

After switching on the power supply of the AG02 you may enter your commands using the command list of a suitable 'terminal program', e. g., sikoterm.exe or Microsoft 'hyperterminal' (component of Win 98, XP, etc.).

You can obtain the program 'sikoterm.exe' from SIKO GmbH or load it from the Internet under the address '<http://siko.de/download>'.

With the standard protocol, transfer functions so that the PC (terminal) sends a letter (ASCII), with additional parameters, if necessary. Subsequently, the AG02 sends a reply with a concluding 'CR'.

Parameter: 9600 baud, no parity, 8 data bits, 1 stop bit, no handshake

Note on input: lowercase and uppercase letters are accepted (ASCII).

Note on output: All reply telegrams are completed with the 'CR' ASCII character, except for the 'K', 'W' commands and in inching operation 2.

If an input is faulty or illegal, the AG02 replies with a “?” and a two-digit error number (example: ?02).

Chapter 6.1 explains the coding of the error numbers.



**Only ‘Read’ commands are enabled if the AG02 is operated with Profibus in the Data Exchange mode or with CAN bus in the NMT status ‘OPERATIONAL’ or ‘STOPPED’, with the exception of the ‘N’ and ‘O’ commands!**

The following table contains the commands of the standard protocol and their description!

Command	Length	Reply	Description
Ay	2/10	“xxxxxxx>“	<b>Device type / software version</b> y=0: hardware version y=1: software version y=2: bus interface y=3: Gear reduction
Ey	2/10	“±xxxxxxx>“	<b>Output 3-byte value</b> y = address ±xxxxxxx = decimal value in increments (spindle pitch = 0) ±xxxxxxx = decimal value 1/100 mm (spindle pitch = ≠ 0) y=0: actual target value (parameter no. 24) y=1: upper limit (parameter no. 15) y=2: lower limit (parameter no. 16) y=3: calibration value (parameter no. 14) y=4: delta travelling distance in inching operation (param. no. 18)
Fy±xxxxxxx	10/2	“>“	<b>Input 3-byte value</b> y = address ±xxxxxxx = decimal value in increments (spindle pitch = 0) ±xxxxxxx = decimal value 1/100 mm (spindle pitch = ≠ 0) y=0: Positioning mode: Target position (volatile) Spindle pitch = 0: Information refers to increments Spindle pitch ≠ 0: Information refers to travel distance in 1/100 mm Speed mode: Rotational target speed (volatile) (parameter no. 24) y=1: upper limit (parameter no. 15) y=2: lower limit (parameter no. 16) y=3: calibration value (parameter no. 14) y=4: delta travelling distance in inching operation 1 (parameter no. 18)

Command	Length	Reply	Description
Gyy	3/7	"xxxxx>"	<b>Output 2-byte value</b> yy = address xxxxx = decimal value yy=00: controller parameter P ( <i>parameter no. 1</i> ) yy=01: controller parameter I ( <i>parameter no. 2</i> ) yy=02: controller parameter D ( <i>parameter no. 3</i> ) yy=03: a-Pos ( <i>parameter no. 4</i> ) yy=04: v-Pos ( <i>parameter no. 5</i> ) yy=05: a-Rot ( <i>parameter no. 6</i> ) yy=06: reserved yy=07: a-Inch ( <i>parameter no. 8</i> ) yy=08: v- Inch ( <i>parameter no. 9</i> ) yy=09: Positioning window spindle pitch = 0 $\Rightarrow$ increments spindle pitch $\neq$ 0 $\Rightarrow$ 1/100 mm ( <i>parameter no. 10</i> ) yy=10: $\ddot{u}$ - Numerator ( <i>parameter no. 11</i> ) yy=11: $\ddot{u}$ - Denominator ( <i>parameter no. 12</i> ) yy=12: CAN baud rate (only with CAN option) (value: see 'H12' command) ( <i>parameter no. 23</i> ) yy=13: spindle pitch in 1/100 mm ( <i>parameter no. 13</i> ) yy=14: bus address (only with Profibus/CAN-Bus option)
Hyyxxxxx	8/2	">"	<b>Input 2-byte value</b> yy = address xxxxx = decimal value yy=00: controller parameter P ( <i>parameter no. 1</i> ) yy=01: controller parameter I ( <i>parameter no. 2</i> ) yy=02: controller parameter D ( <i>parameter no. 3</i> ) yy=03: a-Pos ( <i>parameter no. 4</i> ) yy=04: v-Pos ( <i>parameter no. 5</i> ) yy=05: a-Rot ( <i>parameter no. 6</i> ) yy=06: reserved yy=07: a-Inch ( <i>parameter no. 8</i> ) yy=08: v- Inch ( <i>parameter no. 9</i> ) yy=09: enter positioning window ( <i>parameter no. 10</i> ) spindle pitch = 0 $\Rightarrow$ increments spindle pitch $\neq$ 0 $\Rightarrow$ 1/100 mm yy=10: $\ddot{u}$ -Numerator ( <i>parameter no. 11</i> ) yy=11: $\ddot{u}$ -Denominator ( <i>parameter no. 12</i> ) yy=12: CAN-baud rate: ( <i>parameter no. 23</i> ) Value = 0: 15,625 kBaud Value = 1: 20 kBaud Value = 2: 25 kBaud Value = 3: 40 kBaud Value = 4: 50 kBaud Value = 5: 62,5 kBaud Value = 6: 100 kBaud Value = 7: 125 kBaud Value = 8: 200 kBaud Value = 9: 250 kBaud Value = 10: 500 kBaud Value = 11: 1000 kBaud yy=13: enter spindle pitch in 1/100 mm ( <i>parameter no. 13</i> ) yy=14: enter bus address ( <i>parameter no. 22</i> ) Profibus $\Rightarrow$ 0 ... 126 CAN bus $\Rightarrow$ 1 ... 127
I	1/2	">"	<b>Cancel current travel job in positioning mode</b> Motor remains in control state!



Command	Length	Reply	Description
Jy	2/4 2/12	"zz>" "xxxxxxxxx>"	<b>Read out fault buffer</b> y=0: zz = number of faults( decimal value ) y=1: read out fault memory x x x x x x x x x x 1 2 3 4 5 6 7 8 9 10 error no. <u>Example:</u> ABB - - - - - > 3 errors occurred 1 time fault A 2 times fault B letters: see chapter 5.2 ⇒ table 2: Fault codes
K	1/0		<b>Software reset</b>
Ly	2/2	">"	<b>Enter positioning type (parameter no. 20)</b> y=0: direct positioning y=1: positioning with loop positive y=2: positioning with loop negative
M	1 /2	">"	<b>Start of travel job</b> Positioning mode: - start positioning to programmed target value Velocity mode: - start of velocity mode
N	1/ 2	">"	<b>Motor emergency stop</b> motor decelerates with maximum deceleration. Motor remains in control state! <b>Caution!</b> If a lag error is pending at the time of the 'N' command (see chapter 2.1 ⇒ bit 15 = '1'), the motor will be enabled
O	1/2	">"	<b>Motor stop</b> motor decelerates with programmed deceleration. Motor remains in control state! <b>Caution!</b> If a lag error is pending at the time of the 'O' command (see chapter 2.1 ⇒ bit 15 = '1'), the motor will be enabled and the motor <b>does not</b> decelerate with programmed speed!
P	1 /2	">"	<b>Enable motor</b>
Q	1/ 4	"yy>"	<b>Output flag register</b> yy = value (hex) x x x x x x x x = binary representation of yy 7 6 5 4 3 2 1 0 bit Bit 0 ⇒sense of rotation: '0' = i , '1' = e; bit 1+2 ⇒ positioning type: '00' = direct: '01' = loop + '10' = loop -  bit 3 ⇒ not assigned bit 4 ⇒ operating mode: 0' = positioning mode '1' = rot. speed mode  bit 5 ⇒ not assigned bit 6 ⇒ not assigned bit 7 ⇒ not assigned
R	1/ 6	"xxyy>"	<b>Output system status word (HEX)</b> for the meaning of the individual bits see system status word table (see chapter 2.1) xx = high byte yy = low byte

Command	Length	Reply	Description
Sxxxxx	6/2	">"	<b>Reset device to basic state</b> (Bus address and CAN baud rate are maintained with the Profibus/Can bus option) x=11100: all parameters to basic state (see chapter 8: parameter description 'Default' column) x=11101: only standard parameters to basic state (controller or standard params see chapter 8) x=11102: only controller parameters in basic state (controller or standard params see chapter 8) x=11103: reset fault x=11104: calibrate AG02 x=11105: delete fault memory
Tx	2/2	">"	<b>Enter sense of rotation or stop mode in inching operation 2, resp. (parameter no. 18)</b> x=0: i sense of rotation x=1: e sense of rotation x=3: Stop mode inching operation 2 motor decelerates with maximum delay x=4: Stop mode inching operation 2 motor decelerates with programmed delay
V	1/6	"±xxx>"	<b>Output actual velocity</b>
W	1/2	"xxxx>"	<b>Binary position value</b> xxxx = 4 byte in 2-complement MSB ... LSB
Xy	2/2	">"	<b>Enter operating mode (parameter no. 20)</b> y=0: Positioning mode y=1: Velocity mode
Y	1/2	">"	<b>Start of inching operation 1 (only in positioning mode)</b>
Z	1/10	"±xxxxxxxx>"	<b>Output position value</b>
, (2C <sub>hex</sub> )	1/0		<b>Positive travelling in inching operation 2</b> drive travels in positive direction as long as the ',' character is sent permanently. (only in positioning mode)
. (2E <sub>hex</sub> )	1/0		<b>Negative travelling in inching operation 2</b> drive travels in negative direction as long as the '.' character is sent permanently. (only in positioning mode)

Table 3: Standard protocol commands

## 6.1 Error number coding:

Error number	Error description
01	input of illegal parameter number
02	illegal value range
03	no authorization (active control via Profibus/CAN bus)
04	input cannot be executed due to operating state (e. g. positioning active).
05	limit switch 1 active (travelling is possible only in limit switch 2 direction in inching operation)
06	limit switch 2 active (travelling is possible only in limit switch 1 direction in inching operation)
07	upper limit exceeded (travelling is possible only in negative direction in inching operation)
08	lower limit exceeded (travelling is possible only in positive direction in inching operation)
09	entered target value exceeds limiting value
10	The AG02 is in the fault state. The cause of the fault must be eliminated and acknowledged via 'S11103' command. The cause of the fault can be read out in the fault buffer (see chapter 6 ⇒ 'J' command)
11	Enable input inactive (drive travel disabled!)

Table 4: Coding of error number

## 7 Trouble shooting

Typical errors during operation of the AG02 via standard protocol:

Error description	Possible causes
Communication with PC failed	<ul style="list-style-type: none"> <li>- connections of serial interface interchanged (see installation instruction 'Electrical connection')</li> <li>- different baud rate of PC and AG02</li> <li>- Enable input inactive</li> <li>- supply voltage of AG02 not turned on</li> </ul>
Positioning failed	<ul style="list-style-type: none"> <li>- see interpretation of standard protocol error number (see chapter 6.1: Coding of error number)</li> </ul>
Inching operation cannot be started	<ul style="list-style-type: none"> <li>- positioning still active terminate current positioning job using the 'I' command (see chapter 6: Standard protocol)</li> <li>- velocity mode active</li> </ul>

Table 5: Trouble shooting

## 8 Parameter description

This chapter describes the parameters of the AG02.

All parameters are stored in the **non-volatile** EEPROM, except for parameter 24 (target value).

\* Controller parameters

No.	Name	Selection / value	Default	Description
1	Controller parameter P *	1 – 500	250	<b>P gain of controller:</b> valid for all operating modes (positioning mode, velocity mode, inching operation) <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1000<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 01</i>
2	Controller parameter I *	0 – 500	5	<b>I gain of controller:</b> valid for all operating modes (positioning mode, velocity mode, inching operation) <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1001<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 02</i>
3	Controller parameter D *	0 – 500	0	<b>D gain of controller:</b> valid for all operating modes (positioning mode, velocity mode, inching operation) <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1002<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 03</i>
4	a – Pos *	1 – 100	50	<b>Acceleration in positioning mode:</b> values in % <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1003<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 04</i>

No.	Name	Selection / value	Default	Description
5	v – Pos *	1 – 100 1 – 80 1 – 35	30	<b>Maximum speed in positioning mode:</b> values in revolutions/min gear ratio 55:1 ⇒ max. 100 rev/min gear ratio 62:1 ⇒ max. 80 rev/min gear ratio 135:1 ⇒ max. 35 rev/min <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1004<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 05</i>
6	a - Rot *	1 – 100	50	<b>Acceleration in velocity mode:</b> values in % <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1005<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 06</i>
7				reserved
8	a - Inch *	1 – 100	50	<b>Acceleration in inching operation 1 I/2:</b> values in % <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1007<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 08</i>
9	v - Inch *	1 – 100 1 – 80 1 – 35	30	<b>Maximum speed in inching operation 1 I/2:</b> values in revolutions/min gear ratio 55:1 ⇒ max. 100 rev/min gear ratio 62:1 ⇒ max. 80 rev/min gear ratio 135:1 ⇒ max. 35 rev/min <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1008<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2410<sub>h</sub> ⇒ sub-index 09</i>
10	Pos- Window	0 - 1000	10	<b>Positioning mode: positioning window</b> If the actual position of the AG02 is within the programmed target value ± this window, this is signalled by setting bit 3 in the system status word of the AG02 (see chapter 2.1) Spindle pitch = 0: Information refers to increments Spindle pitch ≠ 0: Information refers to travelling distance in 1/100 mm <b>Velocity mode: Velocity window</b> If the actual velocity is within the target velocity ± this window, this is signalled by setting bit 3 in the system status word of the AG02 (see chapter 2.1). <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1009<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 6067<sub>h</sub></i>
11	ü – Numerator	1 – 10000	1	<b>Numerator gear ratio:</b> a gear ratio can be programmed here when a gear unit is used (see chapter 10: external gear unit). <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1010<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 6091<sub>h</sub> ⇒ sub-index 01</i>
12	ü – Denominator	1 – 10000	1	<b>Denominator gear ratio:</b> a gear ratio can be programmed here when a gear unit is used (see chapter 10: external gear unit). <i>Standard protocol chapter 6: 'G'/'H' commands</i> <i>Profibus chapter 9.6: parameter no. 1011<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 6091<sub>h</sub> ⇒ sub-index 02</i>

No.	Name	Selection / value	Default	Description
13	Spindle pitch	0 – 1000	0	<p><b>Spindle pitch:</b>            Spindle pitch = 0:            Position value is output in increments (1,600 increments per revolution of the driving shaft of the AG02).            Spindle pitch parameter <math>\neq 0</math>:            (when operating the AG02 on a spindle)            Position value is no longer output as increments, but as travelling distance in 1/100 mm. Target position is entered now as 1/100 mm, too.            Spindle pitch value in 1/100 mm.            e. g. spindle with a pitch of 2 mm <math>\Rightarrow</math> spindle pitch parameter = 200.  <i>Standard protocol chapter 6: 'G'/'H' commands</i>  <i>Profibus chapter 9.6: parameter no. 1012<sub>dec</sub></i>  <i>CAN bus chapter 10.12.2: object 2412<sub>h</sub></i></p>
14	Calibration value	-999999 to 999999	0	<p><b>Calibration value:</b>            Writing a value in this parameter results in the take-over of this value as the absolute position for the AG02.            In the case of calibration via the external calibration input or the interface (RS232/Profibus/CAN bus), respectively, the absolute position of the AG02 is set to the value programmed here (see also chapter 3: Calibration).  <b>Caution!</b> Value must be within the range of the preprogrammed limiting values.  <i>Standard protocol chapter 6: 'E'/'F' commands</i>  <i>Profibus chapter 9.6: parameter no. 1018<sub>dec</sub></i>  <i>CAN bus chapter 10.12.2: object 607C<sub>h</sub></i></p>
15	Upper Limit	-9999999 to 9999999	1000000	<p><b>Positioning mode: upper limit</b>            specifies the maximum position in positive direction.            Spindle pitch = 0: Information refers to increments            Spindle pitch = <math>\neq 0</math>: Information refers to travelling distance in 1/100 mm            If the AG02 is beyond this limit, travelling is only possible in negative direction via inching operation.  <b>Caution!</b> Value must be greater than 'lower limit'.            Limit monitoring is deactivated if 'upper limit' is equal 'lower limit'. Please note that there is a jump of the actual position if the resolution of the absolute encoder is exceeded!  <b>Velocity mode:</b> no meaning  <i>Standard protocol chapter 6: 'E'/'F' commands</i>  <i>Profibus chapter 9.6: parameter no. 1016<sub>dec</sub></i>  <i>CAN bus chapter 10.12.2: object 607D<sub>h</sub> <math>\Rightarrow</math> sub-index 02</i></p>

No.	Name	Selection / value	Default	Description
16	Lower Limit	-9999999 to 9999999	- 1000000	<p><b>Positioning mode: lower limit</b> specifies the maximum position in negative direction. Spindle pitch = 0: Information refers to increments Spindle pitch = ≠0: Information refers to travelling distance in 1/100 mm If the AG02 is beyond this limit, travelling is only possible in positive direction via inching operation. <b>Caution!</b> Value must be smaller than 'upper limit'. Limit monitoring is deactivated if 'upper limit' is equal 'lower limit'. Please note that there is a jump of the actual position if the resolution of the absolute encoder is exceeded! <b>Velocity mode:</b> no meaning <i>Standard protocol chapter 6: 'E'/'F' commands</i> <i>Profibus chapter 9.6: parameter no. 1017<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 607D<sub>h</sub> ⇒ sub-index 01</i></p>
17	Delta Inch	-1000000 to 1000000	1600	<p><b>Delta travelling distance with inching operation 1:</b> indicates the relative travelling distance. positive value ⇒ positive travelling direction negative value ⇒ negative travelling direction Spindle pitch = 0: Information refers to increments Spindle pitch = ≠0: Information refers to travelling distance in 1/100 mm <i>Standard protocol chapter 6: 'E'/'F' commands</i> <i>Profibus chapter 9.6: parameter no. 1019<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2415<sub>h</sub></i></p>
18	Sense of rotation	i, e	i	<p><b>Counting direction of the measuring system:</b> When the shaft rotates counter-clockwise (view on the clamping ring of the AG02). i sense of rotation: ⇒ positive counting direction e sense of rotation: ⇒ negative counting direction <i>Standard protocol chapter 6: 'T'/'Q' commands</i> <i>Profibus chapter 9.6: parameter no. 1013<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 607E<sub>h</sub></i></p>
19	Pos Type	direct loop + sloop -	direct	<p><b>Positioning mode: Type of positioning</b> <i>direct:</i> direct travelling from actual position to target value <i>loop +:</i> travelling to the target value occurs always in positive direction to compensate for spindle play <i>loop -:</i> travelling to the target value occurs always in negative direction to compensate for spindle play. <b>Caution!</b> Loop positioning in positioning mode only. <b>Velocity mode:</b> no meaning <i>Standard protocol chapter 6: 'L'/'Q' commands</i> <i>Profibus chapter 9.6: parameter no. 1014<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 2413<sub>h</sub></i></p>
20	Operating mode	positioning mode / speed mode	positioning mode	<p><b>Positioning mode:</b> (see chapter 2.2.1: Positioning mode) <b>Velocity mode:</b> (see chapter 2.2.2: Velocity mode) <i>Standard protocol chapter 6: 'X'/'Q' commands</i> <i>Profibus chapter 9.6: parameter no. 930<sub>dec</sub></i> <i>CAN bus chapter 10.12.2: object 6060<sub>h</sub></i></p>
21				reserved
22	Address	Profibus 0 – 126 CAN bus 1 - 127	126  127	<p><b>Device address in bus operation:</b> (only for Profibus / CAN bus option) remains unchanged after 'S' command! <i>Standard protocol chapter 6: 'G14' / 'H14' commands</i> <i>CAN bus chapter 10.12.2: object 2101<sub>h</sub></i></p>



No.	Name	Selection / value	Default	Description
23	CAN Baud-rate	15.625, 20, 25, 40, 50, 62.5, 100, 125, 200, 250, 500, 1000	500 kBaud	<b>Baud rate of the CAN interface</b> (only with CAN bus option) remains unchanged after 'S' command! <i>Standard protocol chapter 6: 'G12' / 'H12' commands</i> <i>CAN-bus chapter 10.12.2: object 2100<sub>h</sub></i>
24	Setpoint	see column 'Description'	0	<b>Positioning mode:</b> indicates absolute target position. Spindle pitch = 0: Information refers to increments Spindle pitch = ≠0: Information refers to travelling distance in 1/100 mm Value range: depends on the pre-programmed target values ( <i>parameters 15/16</i> ) <b>Velocity mode:</b> indicates the target velocity in rev./min. Value range: gear ratio 55:1 ⇒ max. 100 rev/min gear ratio 62:1 ⇒ max. 80 rev/min gear ratio 135:1 ⇒ max. 35 rev/min <i>Standard protocol chapter 6: 'E0' / 'F0' commands</i> <i>Profibus see chapter 9.4: Telegram setup (Data Exchange)</i> <i>CAN bus, see chapter 10.2.5: Transmit PDOs</i> <b>Attention!</b> Target value will not be stored in the EEPROM
25	Stopmode Tipp 2	0 / 1		Stop behaviour of inching operation 2 can be parameterized in different ways. In the default setting, the motor is stopped abruptly at the end with maximum delay. Optionally, stopping is also possible via parameterized inching acceleration. Selection may be done either via Profibus with the parameter 1021dez or via the serial interface with the commands "T3" or "T4", respectively. The setting is stored non-volatily.

Table 6: Parameter description

## 9 Communication via Profibus – DP (optional)

### 9.1 General remarks

This chapter describes the activation and parameterization of the AG02 via Profibus interface. For the pin assignment of the Profibus interface please refer to the installation instruction.

### 9.2 Interface

Profibus – DP is an international standardized, open fieldbus standard and is defined in the standards:

- European Fieldbus Standard EN50170
- DIN 19245 parts 1 and 3

The fieldbus is used for cyclic data exchange between a master and the subordinate slaves.

- Masters determine data traffic on the bus and are called 'active stations'.
- Slaves can only acknowledge received messages or transfer messages to the master upon inquiry. Slaves are called 'passive stations'.

The AG02 is operated on the Profibus-DP as a slave (passive bus station) and can, therefore, acknowledge messages or send data only upon inquiry from the master.

The AG02 automatically recognizes the baud rate of the Profibus-DP. The master uniformly sets the baud rate for all devices connected to the Profibus.

The AG02 supports the following baud rates:

- 9.6 kBd, 19.2 kBd, 93.75 kBd, 187.5 kBd, 500 kBd, 1.5 MBd, 3 MBd, 6 MBd, 12 MBd

#### Line length:

Baud rate in kbit/s	9.6	19.2	93.75	187.5	500	1500	3000	6000	12000
Line length in m	1200	1200	1200	1000	400	200	100	100	100

Table 7: Line length

#### Termination of the Profibus-DP line:

If the ASG02 is at the end of the bus, the Profibus connection must be terminated using a defined bus terminator. With the AG02 the “Profibus” bus terminator is used, which is available from SIKO as an accessory. The terminator is screwed onto the last free bus connection.

### 9.3 Data exchange

The flow charts, control functions and status messages as well as the type of cyclic data traffic between master and AG02 (slave) are based on the processes and data structures defined in the document

#### Profibus profile for adjustable speed drives, PROFIDRIVE version 2

(September 1997 issue, PNO order no. 3.071).

### 9.4 Telegram setup (Data Exchange)

During operation ( Data exchange  $\Rightarrow$  yellow LED is lighting, see *installation instruction, fig. 7*), 7 data words are cyclically exchanged between a master and the AG02.

The structure of these data words for cyclic data traffic is called “Parameter Process data Object” (PPO) in the Profibus profile “Adjustable-speed Drives” PROFIDRIVE version 2.

#### 9.4.1 Parameter Process Data Object

Such a Parameter Process data Object (PPO) consists of two parts:

- Parameter data range (see *chapter 9.4.2: Parameter identification value*)

Any parameter in the AG02 can be monitored or changed by means of the PKW telegram portion (parameter identification value)

- Process data range (see *chapter 9.4.3: Process data*)

Control words and target values (master  $\Rightarrow$  slave) or status words and actual values (slave  $\Rightarrow$  master), respectively, can be transferred with the process data.



## Definition of the Parameter Process data Object

A special PPO type with fixed data length (7 data words) has been defined for the AG02:

PKW (4 words)				PZD (3 words)		
				PZD1	PZD2	PZD3
PKE	IND	PWE		STW ZSW	HSW HIW	HSW (master ⇒ slave) HIW (slave ⇒ master)
MSW						LSW
word 1	word 2	word 3	word 4	word 5	word 6	word 7

Fig. 8: PPO setup

PKW:	Parameter identification value
PKE:	Parameter identification
IND:	Index
PWE:	Parameter value
PZD1...3:	Process data 1 ...3
STW:	Control word
ZSW:	Status word:
HSW:	Main target value
HIW:	Main actual value
LSW:	lowest-value data word
MSW:	highest-value data word

### 9.4.2 Parameter identification value (PKW)

Parameter processing in cyclic data traffic is executed via the PKW portion of the PPO. Here the master formulates a job and sends it to the slave. The master repeats the job until the slave has processed the job and given its reply. The slave makes available the reply until the master formulates a new job. Only one job can be processed at the same time.

#### 9.4.2.1 Parameter identification (PKE)

The parameter identification PKE consists of a data word, in which the type of the job/reply and the associated parameter number has been coded.

The parameter identification is made up as follows:

PKE (Parameter identification)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AK				SPM	Parameter number (PNU)										

Fig. 9: PKE setup

AK:	Job or reply identification, respectively
SPM:	Toggle bit for spontaneous message ⇒ function not implemented (status does not matter)
PNU:	Parameter number

Job/reply processing is defined in such a way that the content of the job identification field determines which fields of the PKW interface (index and/or parameter value (PWE)) must be interpreted as well.

### Job identification (master ⇒ slave)

Job identification	Function	Reply identification (possible answers from slave)	
		positive	negative
0	No job	0	7 or 8
1	Request parameter value	1 or 2	
2	Change parameter value (word)	1	
3	Change parameter value (double word)	2	
4	Request description element	3	
5	Change description element	3	
6	Request parameter value (array)	4 or 5	
7	Change parameter value (array word)	4	
8	Change parameter value (double word)	5	
9	Request number of array elements	6	

Table 8: Job identification

The right column, Reply identification, refers to the replies from the slave (see table 9: Reply identification). In the normal case there is a positive reply, in case of a fault the reply is negative.

### Reply identification (slave ⇒ master)

Reply identification	Function
0	No reply
1	Parameter value transferred (word)
2	Parameter value transferred (double word)
3	Description element transferred
4	Parameter value transferred (array word)
5	Parameter value transferred (array double word)
6	Number of array elements transferred
7	Job cannot be executed
8	No authorization for PKW interface

Table 9: Reply identification

### Parameter number (master ⇒ slave, slave ⇒ master)

This field contains the number of the parameter whose data is transferred in the Parameter Value (PWE) field.

#### 9.4.2.2 Sub-index (IND)

This field contains the array sub-index in the case of jobs and replies that refer to array elements.

#### 9.4.2.3 Parameter value (PWE)

This field contains the numerical value of the parameter present in the Parameter Number (PNU) (see chapter 9.6: Parameterization ⇒ table 15).

The PWE transfer of word sizes is realized with word 4 of the PPO, transfer of double word sizes is realized with word 3 and word 4 of the PPO (see fig. 8: PPO setup).

In the case of non-executable jobs the slave replies with an error number as described in the table below:

No.	Meaning
0	illegal parameter number
1	parameter value cannot be changed
2	upper or lower limit exceeded
3	faulty sub-index
4	no array
5	wrong data type
6	no setting permitted (only resettable)
7	description element cannot be changed
8	PPO Write required in IR not available
9	description data not available
10	wrong access group
11	no authorization
12	wrong password
13	text in cyclic traffic not readable
14	name in cyclic traffic not readable
15	no text array available
16	PPO is missing
17	job cannot be executed due to operating state
18	other error
19	date in cyclic traffic not readable

Table 10: Error numbers with reply

### 9.4.3 Process data (PZD)

In the process data portion all information is transferred, which is exchanged in normal cyclic travelling operation, i. e., control commands, target values from master to drive or status word, respectively, actual values from drive to master. Due to the two operating modes of the AG02 (positioning mode and velocity mode) there are different meanings for individual parameters. The process data portion is set up as follows:

Data transfer (master  $\Rightarrow$  slave)

PZD		
STW	target value	
	high word	low word

Fig. 10: Process data master  $\Rightarrow$  slave

#### Control word (STW):

For a description of the control word in positioning mode refer to chapter 9.5.1.

For a description of the control word in velocity mode refer to chapter 9.5.4.

#### Target value:

The target value consists of 4 bytes and has a different meaning depending on the operating mode:

- Positioning mode  
Target value contains the target position applicable to the next travelling job. The value must be in the range of the pre-programmed limiting values. (see chapter 9.6: Parameterization, parameter nos 1016/1017!)
- Velocity mode  
Target value contains the target velocity value in rev./min.

Data transfer slave  $\Rightarrow$  master

PZD		
ZSW	actual value	
	high word	low word

Fig. 11: Process data slave  $\Rightarrow$  master**Status word ZSW:**

Description: Positioning mode, see chapter 9.5.2  
Velocity mode, see chapter 9.5.5

**Actual value:**

The actual value consists of 4 bytes and has a different meaning depending on the operating mode:

- Positioning mode  
The actual value contains the actual position value
- Velocity mode  
The actual value contains the actual velocity

**9.5 Functional description of the control units**

The control and status words are represented as follows:

high byte								low byte							
bit number															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0
2				9				4				8			

Fig. 12: Representation of control and status words

Example (grey background):

binary:  $\Rightarrow$  0010 1001 0100 1000

hex:  $\Rightarrow$  2 9 4 8

**9.5.1 Control word in positioning mode (master  $\Rightarrow$  slave)**

Bit	Value	Designation acc. to PROFIDRIVE	AG02 description
0	1	ON	OFF 1: not active (operational condition)
	0	OFF 1	OFF 1: cancellation of positioning job (motor enabled, ready-to-run state) / release from turn-on interlock
1	1	Operating condition	OFF 2: not active
	0	OFF 2	OFF 2: cancellation of positioning job (motor enabled, ready-to-run state)
2	1	Operating condition	Not implemented
	0	OFF 3	must be statically set to '0'

3	1	Operation enabled	Enable operation
	0	Disable operation	Operation disabled Motor decelerates with maximum deceleration and assumes ready-to-run state. Motor remains in control state.
4	1	Operating condition for positioning	Must continuously be pending for travel job. Activation of travelling job via edge on bit 6
	0	Stop	Drive decelerates with maximum deceleration. Current positioning job is discarded. Motor remains in control state.
5	1	Operating condition for positioning	Must continuously be pending for executing a travel job.
	0	Intermediate stop	Drive decelerates from an active travel job with pre-programmed deceleration to n=0 and comes to a standstill with a stop torque. The travel job is not discarded. The travel job is continued after changing to bit 5=1.
6	Edge 0/1 1/0	Activate travel job	Each edge enables a new travel job with the current target value.
7	1	Acknowledge	Acknowledge fault. Turning on of the AG02 will be disabled.
	0	no meaning	-
8	1	Inching 1 On	Condition: Operation is enabled and no active positioning process Drive travels once by delta-Inch target value.
	0	Inching 1 Off	Inching 1 off
9	1	Inching 2 On	Condition: Operation is enabled and no active positioning process. Drive travels until Inching 2 is OFF. Direction depends on bit 15.
	0	Inching 2 Off	Inching 2 off
10 - 14		is not supported	-
15	1	AG02-specific	negative travel direction with inching operation 2
	0	AG02-specific	positive travel direction with inching operation 2

Table 11: Control word: Positioning mode

### 9.5.2 Status word: Positioning mode (slave ⇒ master)

Bit	Value	Designation acc. to PROFIDRIVE	AG02 description
0	1	Ready to turn on	Supply voltage for motor and electronics unit applied.
	0	Not ready to turn on	
1	1	Ready	identical with bit 0.
	0	Not ready	
2	1	Operation enabled	Positioning operation enabled.
	0	Operation disabled	Positioning operation disabled.
3	1	Error	Drive faulty and, therefore, out of operation; turn-on disabled after acknowledgement and successful fault correction. Error code in fault buffer.
	0	Faultless	
4	1	No OFF 2	No pending OFF 2 command.
	0	OFF 2	OFF 2 command pending.
5	1	No OFF 3	Not implemented
	0	OFF 3	Statically on '1'

6	1	Turn-on disabled	Restart only via "OFF 1" with subsequent "On".
	0	Turn-on not disabled	
7	1	Warning	Drive still in operation, no acknowledgement required ( <i>Warnings see chapter 5.1</i> ).
	0	No warning	No pending warning or warning has disappeared.
8	1	No lag error	No lag error present <i>see chapter 2.1: system status word bit 15</i> )
	0	Lag error	
9	1	Guide required	Not supported (statically on '1').
	0	On-site operation	
10	1	Target position reached	The absolute position value is at the end of a travel job in the positioning window.
	0	Outside target position	
11	1	Reference point set	Function not implemented because it is an absolute system (statically on '1').
	0	No reference point set	
12	Edge 0/1 1/0	Acknowledgement of target value	'Edge' acknowledges that a new positioning job was taken over.
13	1	Drive stands still	Signals standstill in case of intermediate stop and stop.
	0	Drive travels	Travel job is executed
14	1	Limiting value for position exceeded	Position limiting value exceeded in positive or negative direction. Travelling possible only via inching operation.
	0	Within position limit	Position value is within pre-programmed limiting values.
15	1	ready-to-travel	AG02 is ready-to-travel if operation has been enabled (ZSW.2 = '1') <ul style="list-style-type: none"> <li>• no limit switches are active</li> <li>• Enable input active</li> <li>• no limits have been exceeded</li> <li>• no fault is active</li> <li>• no current positioning is active</li> </ul>
	0	not ready-to-travel	AG02 is not ready-to-travel

Table 12: Status word: Positioning mode



9.5.3 Flow chart: Operating mode: Positioning mode

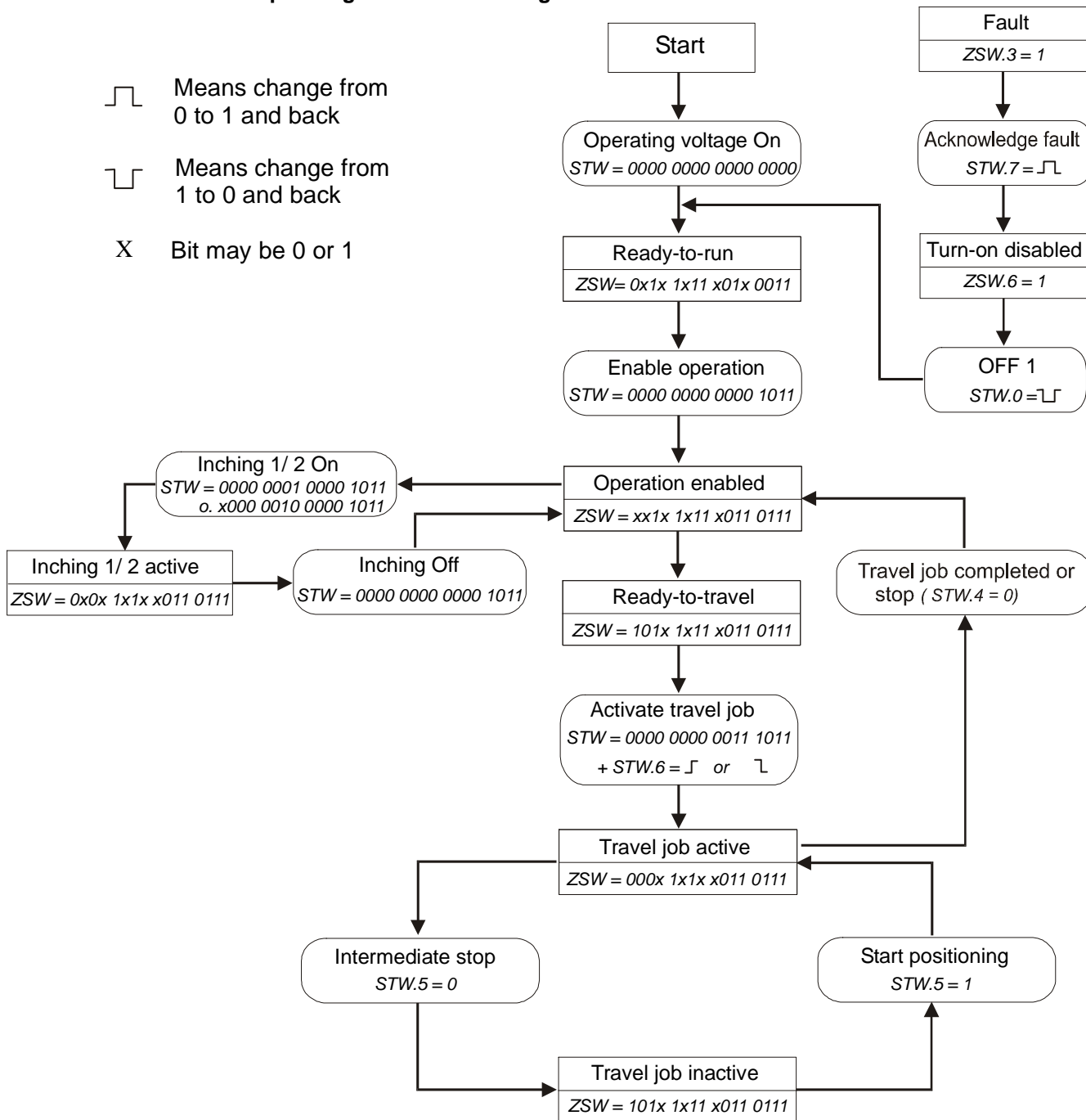


Fig. 13: Flow chart: Positioning mode

#### 9.5.4 Control word: Velocity mode (master ⇒ slave)

Bit	Value	Designation acc. to PROFIDRIVE	AG02 description
0	1	ON	OFF 1: not active (operational condition)
	0	OFF 1	OFF 1: Cancellation of travel job (motor enabled, ready-to-run state) / Release from turn-on interlock
1	1	Operating condition	OFF 2: not active
	0	OFF 2	OFF 2: Cancellation of positioning job (motor enabled, ready-to-run state)
2	1	Operating condition	Not implemented
	0	OFF 3	
3	1	Enable operation	Enable operation
	0	Disable operation	Disable operation Motor decelerates with maximum deceleration and assumes ready-to-run state. Motor remains in control state.
4	1	Operating condition	Not implemented
	0	Disable starting encoder	
5	1	Enable starting encoder	Not implemented
	0	Stop starting encoder	
6	1	Enable target value	Target value is enabled, drive rotates with preset velocity.
	0	Disable target value	Drive runs out with pre-programmed deceleration Motor remains in control state.
7	1	Acknowledge	Acknowledge fault. Turning on of the AG02 will be disabled.
	0	no meaning	
8 - 15		is not supported	

Table 13: Control word: Velocity mode



### 9.5.5 Status word: Velocity mode (slave ⇒ master)

Bit	Value	Designation acc. to PROFIDRIVE	AG02 description
0	1	Ready to turn on	Supply voltage for motor and electronics unit applied.
	0	Not ready to turn on	
1	1	Ready	identical with bit 0.
	0	Not ready	
2	1	Operation enabled	Velocity mode enabled.
	0	Operation disabled	Velocity mode disabled.
3	1	Error	Drive faulty and, therefore, out of operation; turn-on disabled after acknowledgement and successful error correction. Error code in fault buffer.
	0	Faultless	
4	1	no OFF 2	No pending OFF 2 command.
	0	OFF 2	OFF 2 command pending.
5	1	no OFF 3	Not implemented statically on '1'
	0	OFF 3	
6	1	Turn-on disabled	Restart only via "OFF 1" with subsequent "On".
	0	Turn-on not disabled	
7	1	Warning	Drive still in operation, no acknowledgement required ( <i>Warnings see chapter 5.1</i> ).
	0	No warning	Drive still in operation, no acknowledgement required.
8	1	Monitoring of required/actual values within tolerance range	Actual velocity is within the velocity window of the target velocity.
	0	Monitoring of required/actual values outside tolerance range	
9	1	Guide required	Not supported (statically on '1')
	0	On-site operation	
10	1	f or n reached	Not supported (statically on '1')
	0	Below f or n	
11 and 12		Device-specific	Not supported (statically on '1')
13	1	Drive stands still	Signals standstill at stop.
	0	drive travels	Drive shaft rotates.
14		Device-specific	Not supported (statically on '0')
15	1	ready-to-travel	AG02 is ready-to-travel if <ul style="list-style-type: none"> <li>• operation has been enabled (ZWS.2 = '1')</li> <li>• no fault is active</li> <li>• Enable input active</li> <li>• velocity mode is not active (STW.6 = '0')</li> </ul>
	0	not ready-to-travel	AG02 is not ready-to-travel

Table 14: Status word: Velocity mode

### 9.5.6 Flow chart: Operating mode: Velocity mode

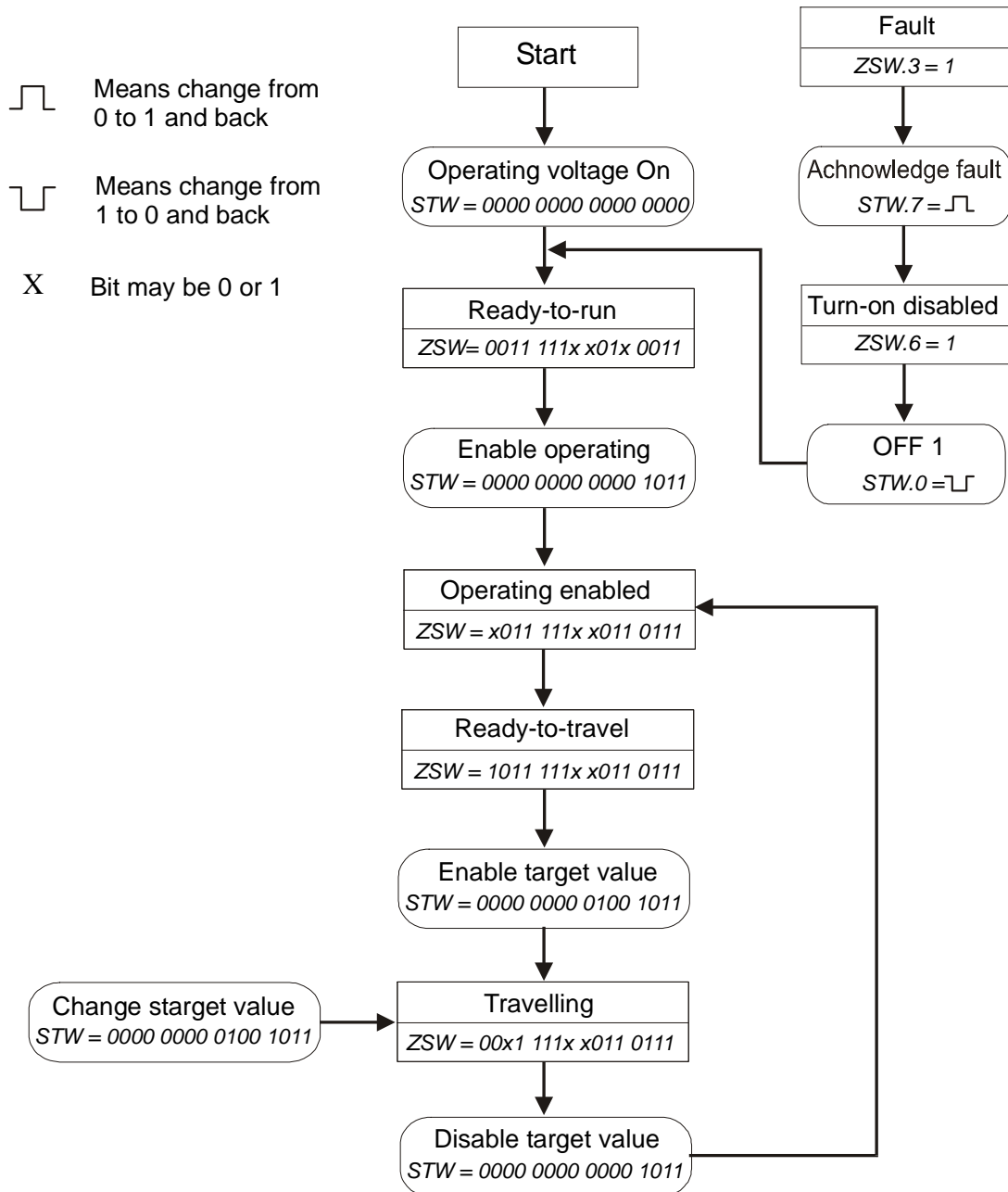


Fig. 14: Flow chart: Velocity mode

## 9.6 Parameterization via Profibus

A parameter number is assigned to every parameter of the AG02 that serves to read out or change the parameters by means of the PKW mechanism (see chapter 9.4.2: PKW). Changes of the parameters are stored in the non-volatile EEPROM.

If the Profibus master sends illegal values to the AG02, this results in an error message (see Table 10: Error numbers with reply).

For the meaning of the individual parameters please refer to chapter 8: 'Parameter description'.

Parameter No.	Name	Access	Format	Description
918 <sub>dez</sub> 396 <sub>hex</sub>	Profibus Address	read	word	Under this parameter number the preset Profibus address can be read out.
930 <sub>dez</sub> 3A2 <sub>hex</sub>	Operating mode	read / write	word	PWE = 1: Velocity mode PWE = 2: Positioning mode (see chapter 8: ⇒ parameter no. 21)
945 <sub>dez</sub> 3B1 <sub>hex</sub>	Fault buffer	read	Array [10] word	This parameter is defined as an array with 10 entries. In this array, occurring faults are entered with their fault codes. This fault code can be read out via parameter number and indication of the sub-index (0 to 9). The fault codes are stored in the EEPROM The fault codes are output as ASCII characters from 'A' to 'E'. The fault buffer can be deleted via parameter no. 970 <sub>dez</sub> . <u>Example:</u> PWE = 41: corresponds to ASCII 'A' For the meaning of the individual characters please refer to chapter 5.2: ⇒ table 2: error codes.
952 <sub>dez</sub> 3B8 <sub>hex</sub>	Number of faults	read	word	Here the number of faults occurring can be read out. The number of faults is stored in the EEPROM.
961 <sub>dez</sub> 3C1 <sub>hex</sub>	Hardware version	read	word	The gear reduction of the AG02 can be read out. PWE = 1: gear reduction 55:1 PWE = 2: gear reduction 62:1 PWE = 3: gear reduction 135:1
965 <sub>dez</sub> 3C5 <sub>hex</sub>	Software-version	read	word	The software version of the AG02 can be read out via this parameter, e. g., 0101 <sub>hex</sub> means version 1.01.
970 <sub>dez</sub> 3CA <sub>hex</sub>	Load parameter data record	write	word*	PWE = 1: Set all parameters to default values (bus address is not changed). PWE = 2: Set standard parameters to default value (see chapter 8: Parameter description). Controller parameters are maintained. PWE = 3: Set controller parameters to factory settings. Standard parameters are maintained. PWE = 4: Delete fault counter and fault buffer. PWE = 5: calibrate the AG02 PWE is set to zero after successful execution (for default values refer to chapter 8: Parameter description) * In deviation from the definition in chapter 9.4.2.1 Parameter identification (PKE), the command Change parameter value (word, job identifier = 2) is acknowledged by the response identifier = 2, Transfer parameter value (double word).
1000 <sub>dez</sub> 3E8 <sub>hex</sub>	Controller parameter P	read / write	word	Value range: 1 – 500 (meaning: see chapter 8: ⇒ parameter no. 1)
1001 <sub>dez</sub> 3E9 <sub>hex</sub>	Controller parameter I	read / write	word	Value range: 1 – 500 (meaning: see chapter 8: ⇒ parameter no. 1)

Parameter No.	Name	Access	Format	Description
1002 <sub>dez</sub> 3EA <sub>hex</sub>	Controller parameter D	read / write	word	Value range: 0 – 500 (meaning: see chapter 8: ⇒ parameter no. 3)
1003 <sub>dez</sub> 3EB <sub>hex</sub>	a – Pos	read / write	word	Value range: 0 – 100 (meaning: see chapter 8: ⇒ parameter no. 4)
1004 <sub>dez</sub> 3EC <sub>hex</sub>	v – Pos	read / write	word	Value range: gear ratio 55:1 ⇒ 1 – 100 gear ratio 62:1 ⇒ 1 – 80 gear ratio 135:1 ⇒ 1 – 35 (meaning: see chapter 8: ⇒ parameter no. 5)
1005 <sub>dez</sub> 3ED <sub>hex</sub>	a – Rot	read / write	word	Value range: 0 – 100 (meaning: see chapter 8: ⇒ parameter no. 6)
1006 <sub>dez</sub> 3EE <sub>hex</sub>				reserved
1007 <sub>dez</sub> 3EF <sub>hex</sub>	a – Inch	read / write	word	Value range: 0 – 100 (meaning: see chapter 8: ⇒ parameter no. 8)
1008 <sub>dez</sub> 3F0 <sub>hex</sub>	v – Inch	read / write	word	Value range: gear ratio 55:1 ⇒ 1 – 100 gear ratio 62:1 ⇒ 1 – 80 gear ratio 135:1 ⇒ 1 – 35 (meaning: see chapter 8: ⇒ parameter no. 9)
1009 <sub>dez</sub> 3F1 <sub>hex</sub>	Pos- Window	read / write	word	Value range: 0 – 1000 (meaning: see chapter 8: ⇒ parameter no. 10)
1010 <sub>dez</sub> 3F2 <sub>hex</sub>	ü- Numerator	read / write	word	Value range: 1 – 10000 (meaning: see chapter 8: ⇒ parameter no. 11)
1011 <sub>dez</sub> 3F3 <sub>hex</sub>	ü – Denominator	read / write	word	Value range: 1- 10000 (meaning: see chapter 8: ⇒ parameter no. 12)
1012 <sub>dez</sub> 3F4 <sub>hex</sub>	Spindle pitch	read / write	word	Value range: 0 – 1000 (meaning: see chapter 8: ⇒ parameter no. 13)
1013 <sub>dez</sub> 3F5 <sub>hex</sub>	Sense of rotation	read / write	word	PWE = 0: i sense of rotation PWE = 1: e sense of rotation (meaning: see chapter 8: ⇒ parameter no. 18)
1014 <sub>dez</sub> 3F6 <sub>hex</sub>	Pos- type	read / write	word	PWE = 0: direct PWE = 1: loop + PWE = 2: loop – (meaning: see chapter 8: ⇒ parameter no. 19)
1015 <sub>dez</sub> 3F7 <sub>hex</sub>				reserved
1016 <sub>dez</sub> 3F8 <sub>hex</sub>	upper limit	read / write	Double word	Value range: -9999999 ... 9999999 (meaning: see chapter 8: ⇒ parameter no. 15)
1017 <sub>dez</sub> 3F9 <sub>hex</sub>	lower limit	read / write	Double word	Value range: -9999999 ... 9999999 (meaning: see chapter 8: ⇒ parameter no. 16)
1018 <sub>dez</sub> 3FA <sub>hex</sub>	Calibration value	read / write	Double word	Value range: -999999 ... 999999 (for the meaning refer to see chapter 8: ⇒ parameter no. 14 and chapter 3: Calibration)
1019 <sub>dez</sub> 3FB <sub>hex</sub>	Delta Inch	read / write	Double word	Value range: - 1000000 ... 1000000 (meaning: see chapter 8: ⇒ parameter no. 17)
1020 <sub>dez</sub> 3FC <sub>hex</sub>	System status word	read	word	System status word of the AG02 for the meaning of the individual bits refer to chapter 2.1 'System status word'
1021 <sub>dez</sub> 3FD <sub>hex</sub>	Stopp-Mode Tipp2	read / write	word	0 = Tipp2 ⇒ motor decelerates with maximum delay 1 = motor decelerates with programmed delay see parameter no. 25

Table 15: Parameter description

### 9.6.1 Example: read parameter

The parameter Calibration value is to be read out:

Determination of parameter identification (PKE), consisting of AK, SPM, PNU:

1. Determination of job identification (AK)

Job identification = request parameter value = 1

(see table 8): Job identification

2. Determination of parameter number:

Parameter number 'calibration value' = 1018 = 3FA<sub>h</sub> (see table 15).

3. SPM = 0;

⇒ **PKE = 13FA<sub>h</sub>**

Parameter identification for the example 'Read parameter'

PKE (Parameter identification)																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designation	AK				SPM	Parameter number (PNU)										
Binary value	0	0	0	1	0	0	1	1	1	1	1	1	1	0	1	0
Hex value	1				3				F				A			

Determination of the parameter identification value (PKW)

1. Determination of parameter identification (see above).

PKE = 0x13FA

2. Determination of the index:

Index (IND) = 0;

3. Parameter value (PWE) = 0

Telegram from master ⇒ slave:

PKW ( 4 words )			
PKE	IND	PWE	
word 1	word 2	word 3	word 4
0x13FA	0x0000	0x0000	0x0000

Telegram from slave ⇒ master with correct execution:

PKW ( 4 words )			
PKE	IND	PWE	
word 1	word 2	word 3	word 4
0x23FA	0x0000	0x0000	0x2710

PKE = 23FA<sub>h</sub>

Reply identification = 2 = transfer parameter value (double word) (see table 9: Reply identification)

PWE = 2710<sub>h</sub> = 10000

The current calibration value is 10000.

### 9.6.2 Example: Write parameter

The parameter 'upper limit' is to be set to 250000:

Determination of parameter identification (PKE), consisting of AK, SPM, PNU:

1. Determination of job identification (AK)

Job identification = change parameter value double word = 3 = 3<sub>h</sub>  
(see table 8): Job identification

2. Determination of parameter number:

parameter number 'upper limit' = 1016 = 3F8<sub>h</sub> (see table 15).

3. SPM = 0;

⇒ PKE = 33F8<sub>h</sub>

Parameter identification for the example 'Write parameter'

PKE (Parameter identification)																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designation	AK				SPM	Parameter number (PNU)										
Binary value	0	0	1	1	0	0	1	1	1	1	1	1	1	0	0	0
ex value	3				3			F				8				

Determination of the parameter identification value (PKW)

1. Determination of parameter *identification* (see above):

PKE = 0x33F8

2. Determination of the index:

Index (IND) = 0

3. Determination of the parameter value (PWE)

Parameter value (PWE) = 250000 = 3D090<sub>h</sub>

Telegram from master ⇒ slave:

PKW ( 4words )			
PKE	IND	PWE	
word 1	word 2	word 3	word 4
0x33F8	0x0000	0x0003	0xD090

Telegram from slave ⇒ master with correct execution:

PKW ( 4words )			
PKE	IND	PWE	
word 1	word 2	word 3	word 4
0x23F8	0x0000	0x0003	0xD090

PKE = 23F8<sub>h</sub>

Reply identification = 2 = transfer parameter value (double word) (see table 9: Reply identification)

## 9.7 Diagnosis

Standard diagnosis for Profibus DP is supported.

Profibus diagnosis consists of 6 bytes with the following contents:

Byte	Bit	Description
Byte 1	0	Diag. station does not exist (set by master)
	1	Diag.station not ready Slave not ready for data exchange
	2	Diag.cfg_Fault Inconsistent configuration data
	3	Diag.ext_diag Slave has external diagnosis data
	4	Diag.not supported Requested function not supported by slave
	5	Diag.invalid slave response (fixes slave to 0)
	6	Diag.prm_fault wrong parameterization (ident number, etc.)
	7	Diag.master_lock (set by master) Slave has been parameterized by other master
Byte 2	0	Diag.prm_req Slave must be re-parameterized
	1	Diag.Stat_diag static diagnosis (byte diag- bits)
	2	fixed to '1
	3	Diag.WD_ON Response monitoring active
	4	Diag.freeze_mode freeze command received
	5	Sync_mode sync command received
	6	reserved
	7	Diag.deactivated (set by master)
Byte 3	0	reserved
	1	reserved
	2	reserved
	3	reserved
	4	reserved
	5	reserved
	6	reserved
	7	Diag.ext_overflow
Byte 4		Diag.master_add Master address after parameterization (FF without parameterization)
Byte 5		Ident number high byte
Byte 6		Ident number low byte

Table 16: Diagnosis bytes

### Stat\_diag:

The slave cannot provide valid data due to a condition in the application. As a result, the master requests only diagnosis data until this bit will be reset. The firmware does not support this bit (permanently on 0)

### Ext\_diag:

If this bit is set, then a diagnosis entry must be present in the user-specific diagnosis area. This bit is always 0 in the AG02, since no user-specific diagnosis data is supported.

### Ext\_diag\_overflow:

This bit is set if more diagnosis data have been provided than would fit the diagnosis data available. This bit is permanently on 0.



## 9.8 Profibus address setup:

The station address is entered via the standard protocol of the serial RS232 or RS485 interfaces (see chapter 6: Standard protocol  $\Rightarrow$  'H'- command).

## 9.9 Device data base and configuration

For the AG02 a device database file (GSD) named SIKO08C9.GSD has been created. This file can be included in the device library by means of the configuration tool used, e. g., 'COM PROFIBUS' of the Siemens company (for the procedure to be applied please refer to the documentation of the configuration tool).

# 10 Communication via CAN bus (optional)

## 10.1 General remarks

This chapter describes activation and parameterization via CAN bus interface. For the connector pin assignment of the CAN bus interface please refer to the installation instruction.

### 10.1.1 Interface

The AG02 supports the following baud rates:

- 15.625 kBd, 20 kBd, 25 kBd, 40 kBd, 50 kBd, 62.5 kBd, 100 kBd, 125 kBd, 200 kBd, 250 kbd, 500 kBd, 1000 kBd

#### Termination of the CAN bus line:

If the ASG02 is at the end of the bus, the CAN bus connection must be terminated by a defined bus terminator. With the AG02 the bus terminator 'CAN-Bus' is used, which is available from SIKO as an accessory.

The terminator is screwed onto the last free bus connection.

## 10.2 CANopen protocol

The CANopen protocol is based on the CANopen communication profile *CiA DS-301 V4.0* as well as the device profile *Drives and Motion Control CiA DSP-402 V2.0*.

The details required for a better understanding and possible deviations are included in this documentation.

### 10.2.1 Telegram setup

The data telegram of a CAN message consists of the following fields:

#### SOF:

Start of Frame  $\Rightarrow$  start bit of the telegram



Identifier:

The Identifier field contains the identifier as well as bits for the recognition of the length of the identifiers (11 or 29 bits). The identifier determines the priority of the message.

Via identifier, CANopen determines additionally the device address, channel selection as well as data direction.

Control field:

contains bits concerning the number of user data and determines whether a data frame or RTR frame (Remote Transfer Request frame) is concerned.

Data field:

contains up to 8 bytes of user data. The user data has a different meaning depending on the channel selection.

CRC:

contains bits for error detection.

ACK/EOF:

The ACK/EOF field contains telegram acknowledgement bits as well as bits for determining the end of telegram.

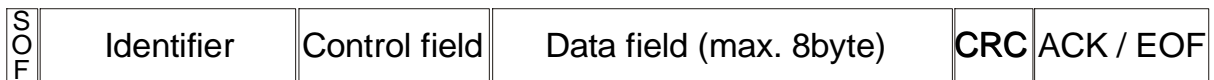


Fig. 15: Telegram setup

For a detailed description of the telegram please refer to the comprehensive CAN literature. For simplification, only identifier and data field will be dealt with in the subsequent telegram descriptions.

### 10.2.2 Network management (NMT)

The master configures, manages and monitors network nodes via the NMT service.

For switching between the four available communication states of a network node, 'INITIALISATION', 'PRE-OPERATIONAL', 'OPERATIONAL' and 'STOPPED', telegrams with the identifier '0' as well as 2 bytes of user data are used.

The identifier of the NMT protocol is limited to 11 bits.

### 10.2.2.1 State Diagramm

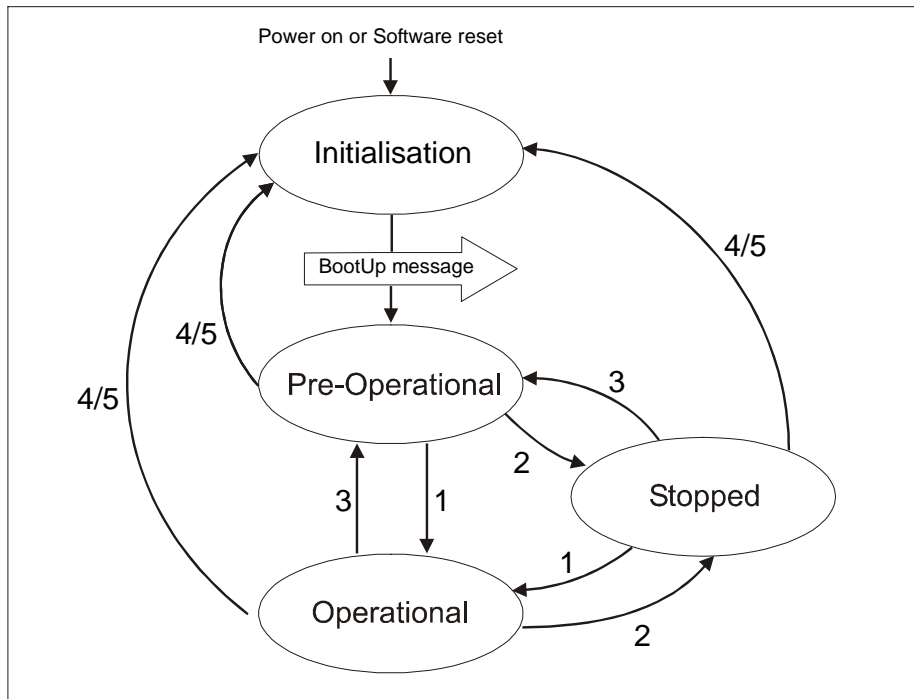


Fig. 16: State Diagramm

### 10.2.2.2 NMT Status 'INITIALISATION'

The AG02 is not involved in the bus actions in this state. All hardware and software components are initialised. This state is attained after switching on the device or after receipt of the command code  $82_{\text{h}}$  of the own or global addresses. After completion of initialisation, the AG02 goes automatically into the 'PRE-OPERATIONAL' status. This is signalled by a boot-up message consisting of the identifier '1791 + Node ID' and a data byte having the value '0'.

### 10.2.2.3 NMT Status 'PRE-OPERATIONAL'

The exchange of parameterization data (SDOs) between the AG02 and the bus master is enabled. However, no process data (PDOs) is transferred. Furthermore, the State Machine of the AG02 is set to the 'SWITCH ON DISABLED' state (see chapter 10.3) and the motor enabled.



**PDO parameters can be changed in this state only!**

### 10.2.2.4 NMT Status 'OPERATIONAL'

Exchange of process and parameterization data is enabled.

### 10.2.2.5 NMT Status 'STOPPED'

Exchange of all data is stopped with the exception of the heartbeat message (see chapter 10.2.10) and the node guarding protocol, if active (see chapter 10.2.11). Only NMT communication is enabled.

Furthermore, the State Machine of the AG02 is set to the 'SWITCH ON DISABLED' state (see chapter 10.3) and the motor enabled.

### 10.2.2.6 Switching between communication states

The network master can switch between the communication states by sending the following telegrams with the identifier '0'.

Status change		Data 1	Data 2
from	to		
PRE-OPERATIONAL / STOPPED	OPERATIONAL (1)	01h	xx
OPERATIONAL / PRE-OPERATIONAL	STOPPED (2)	02h	xx
OPERATIONAL / STOPPED	PRE-OPERATIONAL (3)	80h	xx
OPERATIONAL / PRE-OPERATIONAL / STOPPED	INITIALISATION (4/5)	81h	xx
OPERATIONAL / PRE-OPERATIONAL / STOPPED	INITIALISATION * (4/5)	82h	xx

Table 17: Switching between communication states

\* Cold start is triggered (power on)

xx = 0                                   ⇒ the telegram is intended for all devices on the bus  
 xx = device address               ⇒ the telegram is only intended for the device with the respective address.

### 10.2.3 SYNC object

CANopen enables the simultaneous query of all inputs and the simultaneous setting of all outputs. The synchronization telegram (SYNC), a CAN message with high priority, which contains no user data serves for this purpose.

The identifier of the Sync object can be set via object 1005<sub>h</sub> (see chapter 10.12.2: Description of objects).

### 10.2.4 Process Data Objects (PDOs)

Process data objects serve for fast exchange of short process data. Process data objects are transferred event-triggered, cyclically or on request.

A maximum of 8 bytes of user data can be transferred in a PDO.



**The exchange of PDOs is enabled in the NMT status 'OPERATIONAL' only!**

The AG02 provides 3 Transmit PDOs (process data from the AG02  $\Rightarrow$  NMT master) and 3 Receive PDOs (process data from the NMT master  $\Rightarrow$  to the AG02).  
The Receive PDOs, RPDO1, RPDO3, RPDO4 and the Transmit PDOs, TPDO1, TPDO3 and TPDO4 are supported according to the Device Profil 'Drives and Motion Control CIA DSP-402 Version 2.0'.

## 10.2.5 Transmit PDOs

### 10.2.5.1 1<sup>st</sup> Transmit PDO (TPDO1)

The first Transmit PDO contains 2 user data bytes on which the status word of the AG02 is mapped.  
The Transmit PDO is transferred by the AG02 asynchronously as a standard. Together with the Receive PDO1 on which the control word of the state machine is mapped it forms a handshake connection (acknowledgement operation) between the higher-order control and the drive controller.  
It is, therefore not recommended to change the transfer type of the TDO1.

The COB-ID of the first Transmit PDO is programmed to  $180_h + \text{Node-ID}$  by default.  
The communication parameters are set via the object  $1800_h$  (1<sup>st</sup> Transmit PDO parameter).

1 <sup>st</sup> Transmit PDO		
11 / 29 bit identifier	byte 1	byte 2
	LSB	MSB
	status word (object $6041_h$ )	

Fig. 17: 1<sup>st</sup> Transmit PDO

### 10.2.5.2 3<sup>rd</sup> Transmit PDO (TPDO3)

The third Transmit PDO contains 6 user data bytes on which the status word and the current position value of the AG02 are mapped.

The Transmit PDO3 is transferred only by a RTR frame (remote transfer request) as a standard, i. e., a higher-order control must request the TDO3 (polling). Synchronous transfer (value 0 to 240) can be set via the SYNC object or the time-triggered transfer (value 255) by means of a local cycle (event timer) as an alternative to polling.

The COB-ID of the third Transmit PDO is programmed to  $380_h + \text{Node-ID}$  by default.  
The communication parameters are set via the object  $1802_h$  (3<sup>rd</sup> Transmit PDO parameter).

3 <sup>rd</sup> Transmit PDO						
11/29 bit identifier	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
	LSB	MSB	LSB	NSB	NSB	MSB
	status word (object $6041_h$ )			position value (object $6064_h$ )		

Fig. 18: 3<sup>rd</sup> Transmit PDO

### 10.2.5.3 4<sup>th</sup> Transmit PDO (TPDO4)

The fourth Transmit PDO contains 6 user data bytes on which the status word and the actual velocity of the AG02 is mapped.

The Transmit PDO4 is transferred only by a RTR frame (remote transfer request) as a standard, i. e., a higher-order control must request the TDO4 (polling). Synchronous transfer (value 0 to 240) can be set via the SYNC object or the time-triggered transfer (value 255) by means of a local cycle (event timer) as an alternative to polling.

The COB-ID of the fourth Transmit PDO is programmed to  $480_h + \text{Node-ID}$  by default.

The communication parameters are set via the object  $1803_h$  (4<sup>th</sup> Transmit PDO parameter).

4 <sup>th</sup> Transmit PDO						
11/29 bit identifier	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
	LSB	MSB	LSB	NSB	NSB	MSB
	status word (object $6041_h$ )		actual velocity (object $606C_h$ )			

Fig. 19: 4<sup>th</sup> Transmit PDO

### 10.2.5.4 Transfer types of the Transmit PDOs

Different transfer types can be set for the individual PDOs via objects  $1800_h$  until  $1803_h$  'Transmit PDO Parameter' sub-index 2 (see chapter 10.2.2: Objects of the AG02).

#### Synchronous:

Sub-index 2 (transfer type) = 0:

The Transmit PDO is sent by the AG02 upon receipt of each SYNC telegram.

Sub-index 2 (transfer type) = 1...240:

The Transmit PDO is sent by the AG02 only after receipt of the number of SYNC telegrams indicated under 'transfer type'.

#### Asynchronous:

Event-Triggered: Sub-index 2 (transfer type) = 254:

A PDO is sent with every change of the position value (Profile Position Mode) or the actual velocity (Profile Velocity Mode), respectively (only possible with TPDO3 and TPDO4).

Time-Triggered: Sub-index 2 (transfer type) = 255:

The PDOs are transferred time-triggered.

The sub-index 5 'Event Timer' of the Transmit PDO parameters indicates the cycle time in milliseconds.

Sub-index 2 (transfer type) = 253:

The Transmit PDO is sent after receipt of a RTR frame with the identifier of the respective Transmit PDO.

## 10.2.6 Receive PDOs

### 10.2.6.1 1<sup>st</sup> Receive PDO (RPDO1)

The first Receive PDO contains 2 user data bytes on which the status word of the AG02 is mapped.

The control word in the Receive PDO1 serves for controlling the operation transitions of the state machine. The Receive PDO1 serves for getting the state machine into the OPERATION ENABLED state or for commanding travel interruption or cancellation of travel during running drive movement, respectively.

The drive controller processes the Receive PDO1 asynchronously as a standard. Together with the Transmit PDO1 on which the control word of the state machine is mapped it forms a handshake connection (acknowledgement operation) between the higher-order control and the drive controller.

It is, therefore not recommended to change the transfer type of the RPDO1.

The COB-ID of the first Receive PDO is programmed to  $200_h + \text{Node-ID}$  by default. The communication parameters are set via the object  $1400_h$  (1<sup>st</sup> Receive PDO parameter).

1 <sup>st</sup> Receive PDO		
11 / 29 bit identifier	byte 1	byte 2
	LSB	MSB
	control word (object $6040_h$ )	

Fig. 20: 1<sup>st</sup> Receive PDO

### 10.2.6.2 3<sup>rd</sup> Receive PDO (RPDO3)

The third Receive PDO is assigned to the operation mode 'Profile Position Mode' (positioning mode) and contains 6 user data bytes on which the control word as well as the current target value for the AG02 is mapped.

The position transferred is taken over as the absolute target position. Drive movement in the positioning mode can be executed from the 'OPERATION ENABLED' state of the state machine only.

The drive controller processes the Receive PDO3 asynchronously as a standard.

Together with the Transmit PDO1 on which the status word of the state machine is mapped, it forms a handshake connection (acknowledgement operation) between the higher-order control and the drive controller.

Synchronous transfer type (value 0) can be set to enable synchronous start of more than one drive. The data of the RPDO3 is processed only after receipt of the next SYNC telegram in this case (see chapter 10.2.3: Sync object).

The COB-ID of the third Transmit PDO is programmed to  $400_h + \text{Node-ID}$  by default. The communication parameters are set via the object  $1402_h$  (3<sup>rd</sup> Receive PDO parameter).

3 <sup>rd</sup> Receive PDO						
11/29 bit identifier	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
	LSB	MSB	LSB	NSB	NSB	MSB
	control word (object $6040_h$ )			target value (object $607A_h$ )		

Fig. 21: 3<sup>rd</sup> Receive PDO

### 10.2.6.3 4<sup>th</sup> Receive PDO (RPDO4)

The fourth Receive PDO is assigned to the operation mode 'Velocity Mode' and contains 6 user data bytes on which the control word as well as the current target velocity for the AG02 are mapped.

A driving movement in positive or negative sense of rotation is started via the control word in the Receive PDO4. Drive movement in the velocity mode can be executed from the 'OPERATION ENABLED' state of the state machine only.

The drive controller processes the Receive PDO4 asynchronously as a standard. Together with the Transmit PDO1 onto which the status word of the state machine is mapped, it forms a handshake connection (acknowledgement operation) between the higher-order control and the drive controller. Synchronous transfer type (value 0) can be set to enable synchronous start of more than one drive. The data of the RPDO4 is processed only after receipt of the next SYNC telegram in this case (see chapter 10.2.3: Sync object).

The COB-ID of the fourth Transmit PDO is programmed to  $500_h + \text{Node-ID}$  by default. The communication parameters are set via the object  $1403_h$  (4<sup>th</sup> Receive PDO parameter).

4 <sup>th</sup> Receive PDO						
11/29 bit identifier	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
	LSB	MSB	LSB	NSB	NSB	MSB
	control word (object $6040_h$ )			target velocity (object $60FF_h$ )		

Fig. 22: 4<sup>th</sup> Receive PDO

#### 10.2.6.4 Transfer types of the Receive PDOs

Different transfer types can be set for the individual PDOs via objects  $1400_h$  until  $1403_h$  'Receive PDO Parameter' sub-index 2 (see chapter 10.12.2: Description of objects).

##### Synchronous:

Sub-index 2 (transfer type) = 0 ... 240

In the synchronous transfer type, the Receive PDOs are processed only after receipt of a SYNC telegram.

##### Asynchronous:

Sub-index 2 (transfer type) = 254 ... 255

In the asynchronous transfer type, the Receive PDOs are processed by the AG02 immediately after receipt of the Receive PDO.

#### 10.2.7 Service Data Objects (SDOs)

Service Data Objects (SDOs) serve in the first place for transferring device configuration parameters.

8 bytes of user data are always transferred in a SDO. The identifier is set to 11 bits and cannot be changed.



**The exchange of SDOs is enabled in the NMT statuses 'PRE-OPERATIONAL' as well as 'OPERATIONAL'!**

There is each one COB ID for data transfer from the master to the AG02 (COB-ID  $600_h + \text{Node ID}$ ) and one COB ID for data transfer from the AG02 to the master (COB-ID  $580_h + \text{Node-ID}$ ).

Data transfer is always initiated and controlled by the master.

The COB IDs for the Service Data Objects cannot be changed.

SDO-Telegramm								
11 bit identifier	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
	command	parameter index		sub-index	data 1	data 2	data 3	data 4

Fig. 23: Setup of the SDO telegram

### Byte 1: Command code

The first byte contains the command code of the SDO telegram. The following table lists the available commands and describes their meaning.

Command	Command code hexadecimal	Command code decimal	Meaning
Write Request	23 <sub>h</sub>	35	send parameters to the AG02
Write Response	60 <sub>h</sub>	96	response of the AG02 to Write Request
Read Request	40 <sub>h</sub>	64	request of a parameter by the AG02
Read Response	42 <sub>h</sub>	66	response to the request with current value
Error Response	80 <sub>h</sub>	128	error message

Table 18: Command codes

### Bytes 2/3: Parameter index

The parameter index is entered in the user data byte 2 (low byte) and in the user data byte 3 (high byte) using the Intel data format.

Here, the index of the object to be parameterized is entered (see chapter 10.12.2: Description of objects).

### Byte 4: Sub-index

The sub-index indicates the number of the fields for objects realized as an array.

### Bytes 5 ... 8: Data area

In the data area, the value of the parameter is entered in left-aligned Intel notation.

Byte 5 = low-Byte ... Byte 8 = high Byte

## 10.2.7.1 Error code

The AG02 sends an error response (byte 1 = 80<sub>h</sub>) if a communication error has occurred. An error code is entered in the user data bytes (bytes 5 ... 8).

The table below shows the supported error codes of the AG02.

Command code	Data 1	Data 2	Data 3	Data 4	Meaning
80 <sub>h</sub>	11 <sub>h</sub>	00 <sub>h</sub>	09 <sub>h</sub>	06 <sub>h</sub>	Sub-index does not exist.
80 <sub>h</sub>	02 <sub>h</sub>	00 <sub>h</sub>	01 <sub>h</sub>	06 <sub>h</sub>	Attempt to write read-only object.
80 <sub>h</sub>	01 <sub>h</sub>	00 <sub>h</sub>	01 <sub>h</sub>	06 <sub>h</sub>	Attempt to read write-only object.
80 <sub>h</sub>	30 <sub>h</sub>	00 <sub>h</sub>	09 <sub>h</sub>	06 <sub>h</sub>	Value range of parameter exceeded.
80 <sub>h</sub>	36 <sub>h</sub>	00 <sub>h</sub>	09 <sub>h</sub>	06 <sub>h</sub>	Maximum value is less than minimum value.
80 <sub>h</sub>	00 <sub>h</sub>	00 <sub>h</sub>	02 <sub>h</sub>	06 <sub>h</sub>	Object does not exist.
80 <sub>h</sub>	00 <sub>h</sub>	00 <sub>h</sub>	01 <sub>h</sub>	06 <sub>h</sub>	Unsupported access to an object.
80 <sub>h</sub>	22 <sub>h</sub>	00 <sub>h</sub>	00 <sub>h</sub>	08 <sub>h</sub>	Data cannot be transferred to the application because of the present device state.

Table 19: Error codes



### 10.2.8 Example: Parameterization

The following 2 examples are intended to illustrate parameterization of the AG02 via Service Data Objects.

#### 10.2.8.1 Example: Read parameter

The AG02 has device address 5 and the calibration value is to be read out!

Calculation of the identifier:

Identifier of the parameter channel to the AG02 =  $600_h$  + device address

$$600_h = 1536_{dec}$$

$$\text{Identifier} = 1536 + 5 = 1541 = 605_h$$

Command code = Read request (= request of a parameter from the AG02) =  $40_h$

$$\text{Index} = 607C_h$$

The index of the parameter 'Calibration value' was taken from the directory of objects (*chapter 10.12.2: Description of objects*).

Sub-index = 0

The current calibration value is  $2500 = 9C4_h$ .

Telegram from the master to the AG02:

Identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$605_h$	$40_h$	$7C_h$	$60_h$	$00_h$	$00_h$	$00_h$	$00_h$	$00_h$

Response of the AG02:

Identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
$585_h$	$42_h$	$7C_h$	$60_h$	$00_h$	$C4_h$	$09_h$	$00_h$	$00_h$

Command code = Read Response =  $42_h$

Calculation of the identifier:

Identifier of the parameter channel from the AG02 to the master =  $580_h$  + device address

$$580_h = 1408_{dec}$$

$$\text{Identifier} = 1408 + 5 = 1413 = 585_h$$

#### 10.2.8.2 Example: Write parameter

For the AG02 with device address 5 the upper limit is to be set to 2000000!

Calculation of the identifier:

Identifier of the parameter channel to the AG02 =  $600_h$  + device address

$$600_h = 1536_{dec}$$

$$\text{Identifier} = 1536 + 5 = 1541 = 605_h$$

Command code = Write request (send parameter to the AG02) =  $23_h$

Index = 607D<sub>h</sub>

Sub-index = 2

The index and the sub-index of the parameter 'Upper limit' were taken from the directory of objects (*chapter 10.12.2: Description of objects*).

2000000 = 1E8480<sub>h</sub>

Telegram from the master to the AG02:

Identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
605 <sub>h</sub>	23 <sub>h</sub>	7D <sub>h</sub>	60 <sub>h</sub>	02 <sub>h</sub>	80 <sub>h</sub>	84 <sub>h</sub>	1E <sub>h</sub>	00 <sub>h</sub>

Response of the AG02 in the case of error-free execution:

Identifier	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
585 <sub>h</sub>	60 <sub>h</sub>	7D <sub>h</sub>	60 <sub>h</sub>	02 <sub>h</sub>	00 <sub>h</sub>	00 <sub>h</sub>	00 <sub>h</sub>	00 <sub>h</sub>

Command code = Write Response = 60<sub>h</sub>

Calculation of the identifier:

Identifier of the parameter channel from the AG02 to the master = 580<sub>h</sub> + device address

580<sub>h</sub> = 1408<sub>dec</sub>

Identifier = 1408 + 5 = 1413 = 585<sub>h</sub>

### 10.2.9 Emergency Object (EMCY)

In the case of an error, the node status is transferred via high-priority emergency messages (emergency telegrams). These telegrams have a data length of 8 bytes and contain error information.

The emergency telegram is transferred as soon as the AG02 goes into the fault state (*for the causes of faults refer to chapter 5.2: Faults*) or when a communication error has occurred (*see table 20 Error codes 8001 – 8120*).

Resolution of the cause of fault and resetting of the AG02 with undoing the fault state are signalled by sending of an emergency telegram with the error code 0000<sub>h</sub> (no error) (not applicable to communication errors ⇒ Error codes 8001 – 8021).

The cause of the fault is deposited in the fault buffer (*see object 1003<sub>h</sub>*).

Setup of the emergency telegram.

Identifier	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
11/29 Bit	Emergency Error Code (see chapter 10.2.9.1)		Error Register (Object 1001 <sub>h</sub> )	Manufacturer-specific error field (not used)				

Fig. 24: Emergency protocol

The identifier of the emergency object is set to 128 + Node-ID by default; however, it can be changed via object 1014<sub>h</sub> (*see chapter 10.12.2: Description of objects*).

Transfer of an emergency telegram is enabled in the NMT statuses 'OPERATIONAL' or 'PRE-OPERATIONAL' only!



The emergency telegram must be enabled via object 1014<sub>n</sub>!

### 10.2.9.1 Error Code

The following table contains the possible error codes of the emergency telegram:

Error Code		Meaning
Byte 0 (Highbyte)	Byte 1 (Lowbyte)	
00	00 <sub>n</sub>	No error (no fault present) Is sent after clearing the fault state. (see chapter 5.2: Faults)
21	00 <sub>n</sub>	Current on device input side (overcurrent: see chapter 5.2.1 ⇒ value 'C') The state machine was set to the 'Fault' state. Fault is deposited in the fault buffer.
31	20 <sub>n</sub>	Mains under voltage (Undervoltage: see chapter 5.2.1 ⇒ value 'A') The state machine was set to the 'Fault' state. Fault is deposited in the fault buffer.
43	10 <sub>n</sub>	Excess temperature drive (Overtemperature in output stage: see chapter 5.2.1 ⇒ value 'D') The state machine was set to the 'Fault' state. Fault is deposited in the fault buffer.
71	21 <sub>n</sub>	Motor blocked (Drive shaft blocked: see chapter 5.2.1 ⇒ value 'B') The state machine was set to the 'Fault' state. Fault is deposited in the fault buffer.
80	01 <sub>n</sub>	Bus Error Ack: The message transferred has not been acknowledged by another node. Is <b>not</b> deposited in the fault buffer.
80	02 <sub>n</sub>	Bus Error Form: A portion of the telegram received has the wrong format. Is <b>not</b> deposited in the fault buffer.
80	03 <sub>n</sub>	Bus Error CRC: The checksum of the cyclical block check is incorrect. Is <b>not</b> deposited in the fault buffer.
80	04 <sub>n</sub>	Bus Error Stuff: More than 5 equal bits in a row are present in a portion of the telegram received where this is not permitted. Is <b>not</b> deposited in the fault buffer.
81	20 <sub>n</sub>	The bus status has changed to 'Error Passive' mode. (The AG02 switches to Fault when the bus status 'Bus off' occurs. Fault is deposited in the fault buffer, see chapter 5.2.1 ⇒ value 'E'). Communication via CANopen is no longer enabled!
F0	00 <sub>n</sub>	Manufacturer-specific (Error when reading the absolute position: see chapter 5.2.1 ⇒ value 'E') Error when reading the absolute position The state machine is set to the 'Fault' state. Fault is deposited in the fault buffer.

Table 20: Error Code.

### 10.2.10 Heartbeat protocol

The master monitors the state of the AG02 via heartbeat protocol. While doing this, the AG02 sends cyclically its NMT status.

The AG02 sends the heartbeat telegram independently, without request via RTR frame. The AG02 is a heartbeat producer, it does not receive nor process heartbeat protocols.

The cycle time of the heartbeat telegram is set via object 1017<sub>h</sub> (see chapter 10.12.2: *Description of objects*).

The heartbeat protocol is deactivated if the cycle time is 0.

The heartbeat protocol consists of one byte.

Identifier	Byte1
11 bit	status

Fig. 25: Heartbeat protocol

Status = 0: 'INITIALISATION'  
 Status = 4: 'STOPPED'  
 Status = 5: 'OPERATIONAL'  
 Status = 127: 'PRE-OPERATIONAL'

The identifier of the heartbeat protocol is permanently set to 1792 + Node-ID and cannot be changed.

Heartbeat telegram are sent in the NMT statuses 'OPERATIONAL', 'PRE-OPERATIONAL' or 'STOPPED'.



**The heartbeat protocol is only enabled when node guarding is deactivated!**

### 10.2.11 Node Guarding

Node-guarding or life-guarding mechanisms are available for failure monitoring of the CANopen network. Via node guarding the nodes are monitored that can detect a failure of the master. During guarding, the master transmits remote frames (remot transmit request, message request telegrams) on the guarding identifiers of the nodes to be monitored. The latter respond with the guarding message containing the current status of the slave as well as a toggle bit which must change after each message.

The master assumes that a node error as occurred if status or toggle bits do not correspond with those expected by the master.

Via the objects 100C<sub>h</sub> (Guard Time), 100D<sub>h</sub> (Life Time Factor) the time interval (life time) is set within which the AG02 expects a node query (RTR frame with the COB-ID 1792 + Node-ID) by the NMT master.

The time interval 'Life time' is calculated from the cycle time 'Guard time' multiplied with the factor 'Life Time Factor'.

The AG02 State Machine switches to the 'SWITCH ON DISABLED' state if the AG02 receives no RTR frame from the master during the 'Life Time'.

By sending the first RTR frame of the master to the AG02, Node Guarding of the AG02 is activated after switching on.

Node Guarding is deactivated if the value of either object (100C<sub>h</sub>/100D<sub>h</sub>) is zero.

The response of the AG02 to the RTR frame of the master consists of one byte of user data.

Identifier	Byte 1	
11 bit	bit 7: toggle Bit	bit 6 ... 0: status

Fig. 26: Node-Guarding Telegram

### Toggle Bit:

The toggle bit must alternate between two subsequent responses of the AG02.  
The value of the toggle bit with the first response of the AG02 is 0 after activation of the guarding protocol.

### Status:

Status = 0: 'INITIALISATION'  
Status = 4: 'STOPPED'  
Status = 5: 'OPERATIONAL'  
Status = 127: 'PRE-OPERATIONAL'

The identifier of the heartbeat protocol is permanently set to 1792 + Node-ID and cannot be changed.

A node guard telegram can be sent in the NMT statuses 'OPERATIONAL', 'PRE-OPERATIONAL' or 'STOPPED'.



**The node-guard protocol is only enabled when the heartbeat protocol is deactivated!**

## 10.3 State Machine

In the status word, the CANopen state machine indicates operational and error states of the drive resulting from operational transitions.

The states of the state machine can change via control word (see chapter 10.5) or due to internal events (e. g., occurrence of a fault).

The current state of the state machine can be read via the status word (see chapter 10.4: Table 22).



**The states of the state machine can be set via PDOs only!**

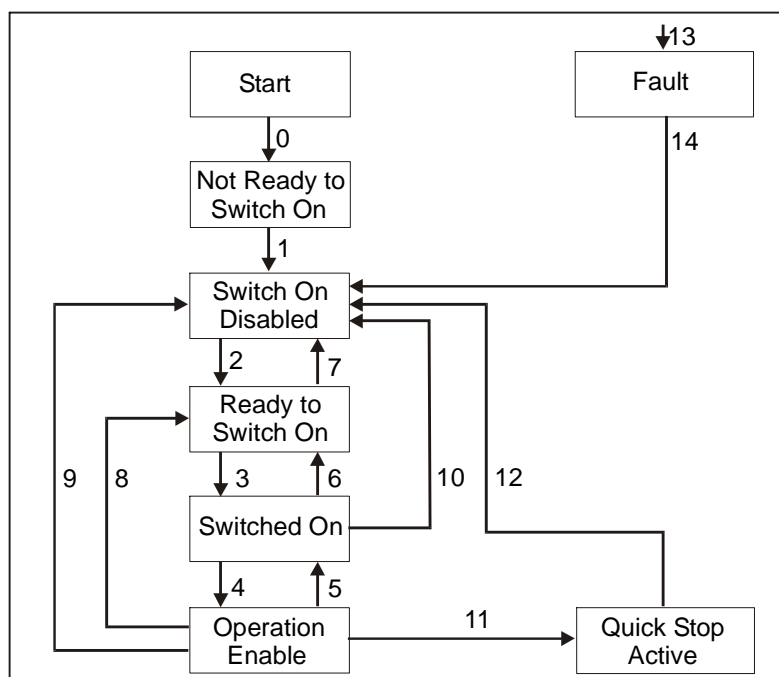


Fig. 27: State machine

The following states of the state machine of the AG02 are available:

- 'NOT READY TO SWITCH ON'  
The AG02 is being initialized after switching on.  
No travel commands can be accepted.  
Motor is enabled
- 'SWITCH ON DISABLED'  
Initialization completed.  
No travel commands can be accepted.  
Motor is enabled
- 'READY TO SWITCH ON'  
No travel commands can be accepted.  
Motor is enabled
- 'SWITCHED ON'  
No travel commands can be accepted.  
Motor is enabled
- 'OPERATION ENABLED'  
Travel commands can be accepted.  
Motor is in control state.
- 'QUICK STOP ACTIVE'  
The Quick Stop command was executed.  
Motor decelerates with maximum deceleration and stops with stop torque.  
Current positioning is cancelled.  
No travel commands can be accepted.
- 'FAULT'  
An error has occurred.  
Motor is enabled  
Current positioning is cancelled.  
No travel commands can be accepted.

The states of the State Machine can be changed via internal events or through commands of the master via control word (see *table 24*).

- State change 0: START ⇒ NOT READY TO SWITCH ON  
Power on or software reset of the AG02
- State change 1: NOT READY TO SWITCH ON ⇒ SWITCH ON DISABLED  
Initialization and self-test of the AG02 successfully completed.
- State change 2: SWITCH ON DISABLED ⇒ READY TO SWITCH ON  
'Shut-down' command by master
- State change 3: READY TO SWITCH ON ⇒ SWITCHED ON  
'Switch-On' command by master
- State change 4: SWITCHED ON ⇒ OPERATION ENABLE  
'Enable Operation' command by master
- State change 5: OPERATION ENABLE ⇒ SWITCHED ON  
'Disable Operation' command by master

- State change 6: SWITCHED ON ⇒ READY TO SWITCH ON  
'Shut-down' command by master
- State change 7: READY TO SWITCH ON ⇒ SWITCH ON DISABLED  
'Disable Voltage' command by master
- State change 8: OPERATION ENABLE ⇒ READY TO SWITCH ON  
'Shut-down' command by master
- State change 9: OPERATION ENABLE ⇒ SWITCH ON DISABLED  
'Disable Voltage' command by master
- State change 10: SWITCHED ON ⇒ SWITCH ON DISABLED  
'Disable Voltage' command by master
- State change 11: OPERATION ENABLE ⇒ QUICK STOP ACTIVE  
'Quick-Stop' command by master
- State change 12: QUICK STOP ACTIVE ⇒ SWITCH ON DISABLED  
'Disable Voltage' command by master
- State change 13: All states ⇒ FAULT  
A fault has occurred
- State change 14: FAULT ⇒ SWITCH ON DISABLED  
'Fault Reset' command by master

## 10.4 Status word

The status word reflects the current status of the AG02. It consists of 16 bits and is mapped on object 6041<sub>n</sub> and on the 3 Transmit PDO's.

Status word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB		High Byte						Low Byte						LSB	

Fig. 28: Status word

The following table lists the designations of the individual bits of the status word and their meanings.

Bit	Designation	Description
0	Ready to switch on	indicates the state of the State Machine (see table 22)
1	Switched on	indicates the state of the State Machine (see table 22)
2	Operation enabled	indicates the state of the State Machine (see table 22)
3	Fault	indicates the state of the State Machine (see table 22)
4	Voltage enabled	bit 4 is set when the supply voltage is within the tolerance limit
5	Quick stop	indicates the state of the State Machine (see table 22) bit 5 is set when the AG02 is not in the 'QUICK STOP ACTIVE' state.
6	Switch on disabled	indicates the state of the State Machine (see table 22)
7	Warning	bit 7 is set when a warning is active (see chapter 5.1: Warnings)

Bit	Designation	Description
8	Profile Position Mode: show readiness for travel	bit 8 is set if the State Machine is in the 'OPERATION ENABLED' state und the follwing conditions are met: <ul style="list-style-type: none"> <li>• no fault present</li> <li>• no limit switches are active</li> <li>• the enable input active</li> <li>• no limit have been exceeded</li> <li>• no active travel job</li> </ul>
	Profile Velocity Mode: show readiness for travel	Bit 8 is set if the State Machine is in the 'OPERATION ENABLED' state und the follwing conditions are met: <ul style="list-style-type: none"> <li>• no fault present</li> <li>• no active travel job</li> <li>• the enable input active</li> </ul>
9	Remote	Bit 9 is set when the AG02 is in the NMT status 'OPERATIONAL' or 'STOPPED'. The AG02 receives commands via CAN interface in this case. Only Read commands are enabled via RS232/485 interfaces.
10	Profile Position Mode: Target reached	Bit 10 is set when the drive has come to a stop at the pre-defined target position within the defined window after a successfully executed positioning command.
	Profile Velocity Mode: Target reached	Bit 10 is set when the actual speed is within the defined window of the target speed.
11	internal Limit	Bit 11 is set when the upper or lower limits have been exceeded.
12	Profile Position Mode: Set Point Acknowledged	Bit 12 is set when the drive controller has started a travel command in the positioning mode. A travel job is started via bit 'New Setpoint' in the control word ( <i>object 6040<sub>n</sub>, control word bit 4</i> ) (value 0 ⇒ 1). Subsequently, the controller firmware plausibilizes the target position, the operation and control parameters and the local state of the drive and sets bit 12 after the check has been passed successfully. Bit 12 is deleted when bit 4 in the control word has been reset to zero following a positioning job (Clear new setpoint).
	Profile Velocity Mode: Speed	Bit 12 is set when the drive stands still.
13	reserved	statically on 0
14	Profile Position Mode: Pos active	Bit 14 is set when there is an active positioning job in the operation mode 'Profile Position Mode' (positioning mode). <b>Attention!</b> No new setpoint is accepted and travelling in inching operation is disabled as long as bit 14 is set!
	Profile Velocity Mode:	no meaning, statically on 0
15	Profile Position Mode: drive travels	The drive shaft of the AG02 is moving if bit 15 is set.
	Profile Velocity Mode:	statically on 0

Table 21: Bit description of the status word



The following table represents the possible states of the state machine and the resulting bit values.

The field containing an x are irrelevant for the states of the state machine.

State	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Ready to Switch On	x	0	x	x	0	0	0	0
Switch On Disabled	x	1	x	x	0	0	0	0
Ready to Switch On	x	0	1	x	0	0	0	1
Switched On	x	0	1	x	0	0	1	1
Operation Enabled	x	0	1	x	0	1	1	1
Quick Stop Activ	x	0	0	x	0	1	1	1
Fault	x	0	x	x	1	0	0	0

Table 22: Low byte status word States of the state machine

## 10.5 Control word

The control word consists of 16 bits and is mapped on the object 6040<sub>h</sub>, and in the 3 Receive PDOs.

It contains bits for controlling the state machine as well as controlling the operational modes, Profile Position Mode (positioning mode) and Profile Velocity Mode (velocity mode).

Control word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB High Byte								Low Byte LSB							

Fig. 29: Control word

The following table lists the designations of the individual bits of the control word and their meanings.

Bit	Designation	Description
0	Switch on	controls the state of the State Machine (see table 24)
1	Disable voltage	controls the state of the State Machine (see table 24)
2	Quick stop	controls the state of the State Machine (see table 24)
3	Enable operation	controls the state of the State Machine (see table 24)
4	Profile Position Mode: New Setpoint	By bit 4, positioning is triggered in the drive controller in the OPERATION ENABLED state (value 0 $\Rightarrow$ 1). The drive controller acknowledges the travel command via bit 12 'Setpoint acknowledged' in the status word (see chapter 10.4).
	Profile Velocity Mode:	no meaning
5	reserved	
6	reserved	
7	Fault reset	If the state machine of the AG02 is in the FAULT state, the fault is reset by an edge on bit 7 (0 $\Rightarrow$ 1) and the state machine is set to the SWITCH ON DISABLED state on the condition that the cause of the fault has been resolved in advance (see chapter 5.2: Faults).
8	Profile Position Mode: Stop	By setting bit 8 on value 1, interruption of travel can be triggered during a running positioning event. Motor runs out with programmed deceleration and stops in the control state. Positioning is resumed and completed after resetting the bit (value 1 $\Rightarrow$ 0).
	Profile Velocity Mode: Stop	By bit 8, drive movement is triggered in the velocity mode in the OPERATION ENABLED state (value 1 $\Rightarrow$ 0).
9	reserved	
10	reserved	
11	reserved	
12	reserved	

Bit	Designation	Description
13	Profile Position Mode: Inching operation 1	Inching operation 1 is started by an edge change (value 0 $\Rightarrow$ 1) on bit 13 (see chapter 2.2.1.2: Inching operation).
	Profile Velocity Mode:	no meaning
14	Profile Position Mode: Inching operation 2 positive	Inching operation 2 is started in positive travel direction by an edge change (value 0 $\Rightarrow$ 1) on bit 14 (see chapter 2.2.1.2: Inching operation). The drive travels in positive direction until bit 14 has been deleted.
	Profile Velocity Mode:	no meaning
15	Profile Position Mode: Inching operation 2 negative	Inching operation 2 is started in negative travel direction by an edge change (value 0 $\Rightarrow$ 1) on bit 15 (see chapter 2.2.1.2: Inching operation). The drive travels in negative direction until bit 15 has been deleted.
	Profile Velocity Mode:	no meaning

Table 23: Bit description of the control word

The following table shows the control of the state machine with the bit combinations of the control word required.

The fields containing an x are irrelevant for the control of the state machine.

Command	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Shutdown	0	x	x	x	x	1	1	0
Switch On	0	x	x	x	0	1	1	1
Disable Voltage	0	x	x	x	x	x	0	x
Quick Stop	0	x	x	x	x	0	1	x
Disable Operation	0	x	x	x	0	1	1	1
Enable Operation	0	x	x	x	1	1	1	1
Fault Reset	0 $\Rightarrow$ 1	x	x	x	x	x	x	x

Table 24: Low Byte Control word<sub>h</sub> control of state machine

- The states of the state machine can be changed via PDOs only!
- PDOs are enabled in the NMT status 'OPERATIONAL' only.
- Travel commands are enabled in the 'OPERATION ENABLED' state of the state machine.



### 10.6 Flowchart of the Profile Position Mode (positioning mode)

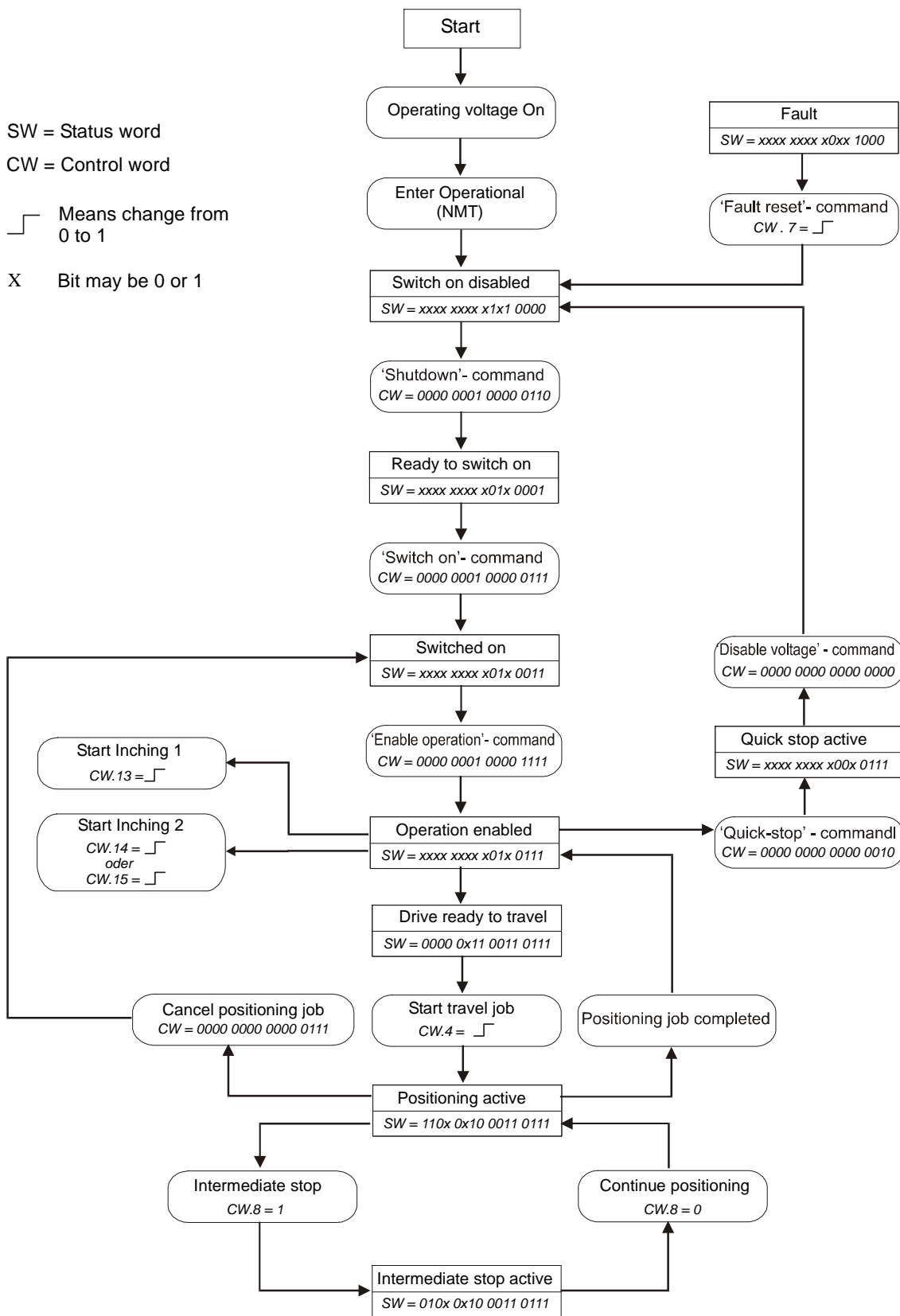


Fig. 30: Flowchart of the Profile Position Mode (positioning mode)



### 10.7 Flowchart of the Profile Velocity Mode operation mode (velocity mode)

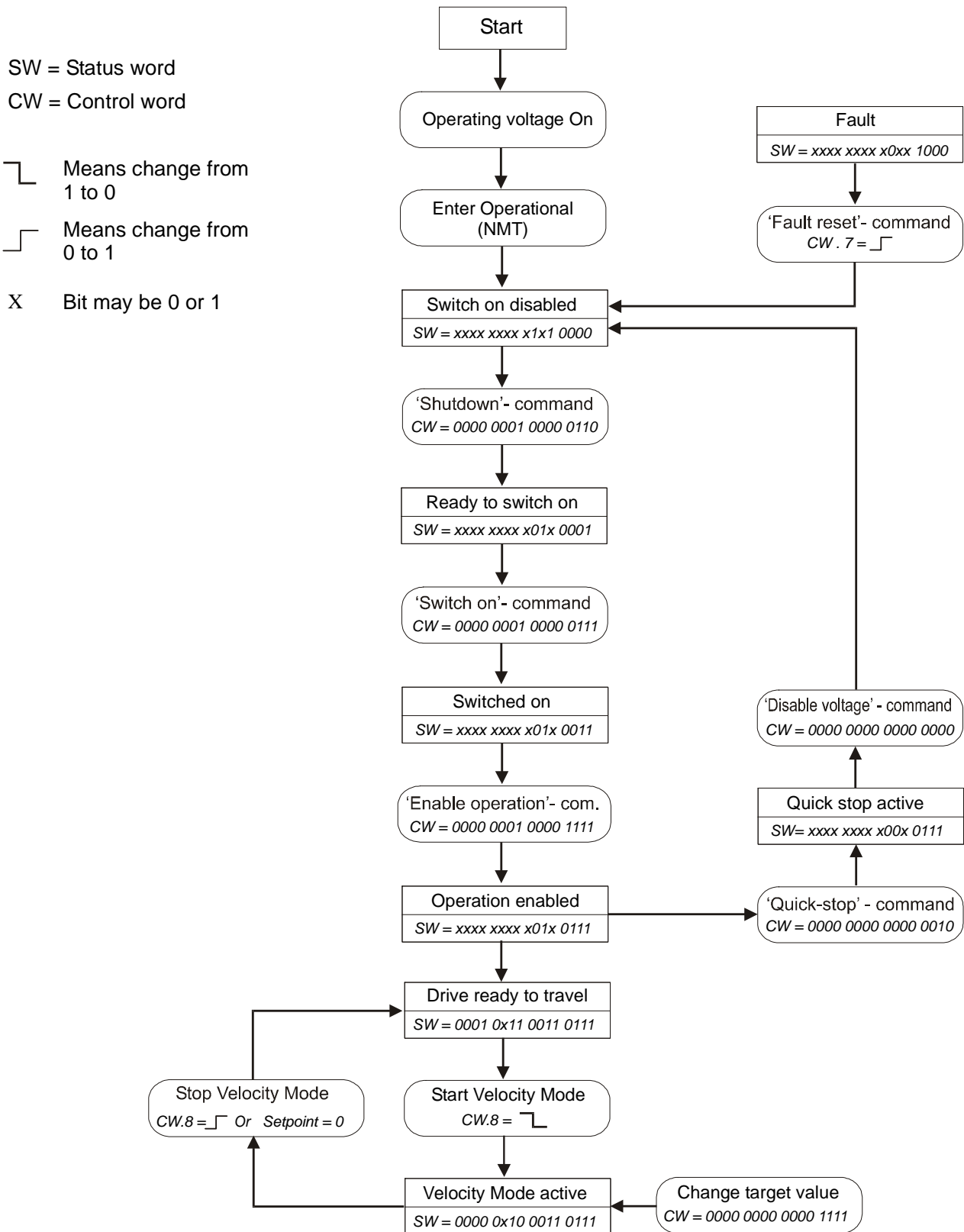


Fig. 31: Flowchart of the Profile Velocity Mode (velocity mode)

## 10.8 Examples

### 10.8.1 Example of the Profile Position Mode (positioning mode)

In the following table a positioning example in the Profile Position Mode (positioning mode) is represented.

The node address of the AG02 is 5 in this example.

Identifier	Message	Meaning
0x000	0x01 0x05	NMT: Enter OPERATIONAL
0x205	0x06 0x01	RPDO1: Shutdown command
0x205	0x07 0x01	RPDO1: Switch On command
0x205	0x0F 0x01	RPDO1: Enable Operation command
0x405	0x1F 0x00 0x88 0x13 0x00 0x00	RPDO3: Travel to position +5000
-	-	wait until target position has been reached
0x205	0x0F 0x01	RPDO1: clear New Setpoint
0x405	0x1F 0x00 0x78 0xEC 0xFF 0xFF	RPDO3: Travel to position -5000
0x205	0x1F 0x01	RPDO1: Intermediate stop
0x205	0x1F 0x00	RPDO1: continue positioning
-	-	wait until target position has been reached
0x205	0x0F 0x01	RPDO1: clear New Setpoint
0x205	0x07 0x01	RPDO1: Disable Operation command
0x205	0x06 0x01	RPDO1: Shutdown command
0x205	0x00 0x01	RPDO1: Disable Voltage command
0x000	0x80 0x05	NMT: Enter PRE-OPERATIONAL

Table 25: Positioning example in Profile Position Mode:

### 10.8.2 Example of the Profile Velocity Mode (velocity mode)

In the following table an example in the Profile Velocity Mode (velocity mode) is represented.

The node address of the AG02 is 5 in this example.

Operating mode change via parameter 20, chapter 8 parameter description (Default: Profile Position Mode)

Identifier	Message	Meaning
0x000	0x01 0x05	NMT: Enter OPERATIONAL
0x205	0x06 0x01	RPDO1: Shutdown command
0x205	0x07 0x01	RPDO1: Switch On command
0x205	0x0F 0x01	RPDO1: Enable Operation command
0x505	0x0F 0x00 0x44 0x00 0x00 0x00	RPDO4: Start of velocity mode with the target speed of +68 rev/min positive sense of rotation
-	-	wait until target speed has been reached
0x505	0x0F 0x00 0x1E 0x00 0x00 0x00	RPDO4: Change of velocity to +30 rev/min
0x205	0x0F 0x01	RPDO1: Stop drive movement
0x505	0x0F 0x00 0xB8 0xFF 0xFF 0xFF	RPDO4: Start of velocity mode with the target speed of -72 rev/min (negative sense of rotation)
-	-	wait until target speed has been reached
0x205	0x0F 0x01	RPDO1: Stop drive movement
0x205	0x07 0x01	RPDO1: Disable Operation command
0x205	0x06 0x01	RPDO1: Shutdown command
0x205	0x00 0x01	RPDO1: Disable Voltage command
0x000	0x80 0x05	NMT: Enter PRE-OPERATIONAL

Table 26: Example of Profile Velocity Mode

## 10.9 Overview of CANopen identifiers

The following table offers an overview of the identifiers used in the AG02.

Default identifier (hexadecimal)	Default identifier (decimal)	Description	Own setting
0	0	Network management (NMT)	
80	128	SYNC – Message	
80 + Node-ID	128 + Node-ID	Emergency Message	
180 + Node-ID	384 + Node-ID	TPDO1	
200 + Node-ID	512 + Node-ID	RPDO1	
380 + Node-ID	896 + Node-ID	TPDO3	
400 + Node-ID	1024 + Node-ID	RPDO3	
480 + Node-ID	1152 + Node-ID	TPDO4	
500 + Node-ID	1280 + Node-ID	RPDO4	
580 + Node-ID	1408 + Node-ID	SDO (tx)	
600 + Node-ID	1536 + Node-ID	SDO (rx)	
700 + Node-ID	1792 + Node-ID	Heartbeat Message	
700 + Node-ID	1792 + Node-ID	Node-Guard Message	

Table 27: Overview of identifiers

## 10.10 Setting the CAN address and CAN baud rate

Station address and CAN baud rate are entered via standard protocol of the serial interface RS232 or RS485, respectively (see chapter 6: Standard protocol  $\Rightarrow$  'H'-command) or via objects 2100<sub>h</sub> (CAN baud rate) and 2101<sub>h</sub> Node-ID (see chapter 10.12.2: Description of objects).

The factory setting of the station address is programmed to 127 and the baud rate is programmed to 500 kbaud.

## 10.11 EDS file

The EDS file SIKOAG02.EDS (electronic data sheet) is available for the AG02.

This file enables easy integration and configuration of the AG02 in a CANopen network using commercial CANopen configurations.

## 10.12 Directory of objects

Every CANopen device keeps a directory of objects where all parameters of the device are deposited in the form of object entries. The object entries can be accessed via SDO communication services (see chapter 10.2.7). Thus, a parameter can be read (SDO upload) and written (SDO download) if permitted by the object entry access rights or the state of the device, respectively.

The following index ranges are used:

1000 <sub>h</sub> - 1FFF <sub>h</sub>	objects of the communication profile CIA DS-301 V4.0.
2000 <sub>h</sub> - 5FFF <sub>h</sub>	manufacturer-specific object entries.
6000 <sub>h</sub> - 9FFF <sub>h</sub>	objects of the device profile CIA DSP-402 V2.0.

### 10.12.1 Overview of objects

The following table offers an overview of the objects of the AG02.

Index	Name	Description
1000 <sub>h</sub>	Device type	The object indicates the device profile number of the AG02.
1001 <sub>h</sub>	Error register	The object shows error states of the AG02.
1002 <sub>h</sub>	Manufacturer Status Register	Contains the system status word of the AG02 ( <i>see chapter 2.1</i> ).
1003 <sub>h</sub>	Pre-Defined Error Field	The object stores up to 10 error messages.
1005 <sub>h</sub>	COB-ID Sync Message	Setting of the COB ID of the SYNC object.
100A <sub>h</sub>	Manufacturer Software Version	Indicates the software version of the controller firmware.
100C <sub>h</sub>	Guard Time	Setting of the 'Guard Time' for the Node Guarding protocol ( <i>see chapter 10.2.11</i> ).
100D <sub>h</sub>	Life Time Factor	Setting of the time interval 'Life Time' ( <i>see chapter 10.2.11</i> ).
1011 <sub>h</sub>	Restore Default Parameters	Restoration of factory-set states of the changeable parameters and calibration of the AG02 ( <i>see chapter 3: Calibration</i> ).
1014 <sub>h</sub>	COB-ID Emergency Message	Setting of the COB ID of the Emergency object.
1017 <sub>h</sub>	Producer Heartbeat Time	Setting of the cycle time for the heartbeat protocol ( <i>see chapter 10.2.10</i> ).
1018 <sub>h</sub>	Identity Objekt	Contains the Vendor ID of the device manufacturer.
1200 <sub>h</sub>	Server SDO Parameter	Contains the COB IDs of the default server SDO.
1400 <sub>h</sub>	1 <sup>st</sup> Receive PDO Parameter	Setting of the communication parameters of the RPDO1.
1402 <sub>h</sub>	3 <sup>rd</sup> Receive PDO Parameter	Setting of the communication parameters of the RPDO3.
1403 <sub>h</sub>	4 <sup>th</sup> Receive PDO Parameter	Setting of the communication parameters of the RPDO4.
1600 <sub>h</sub>	1 <sup>st</sup> Receive PDO Mapping Parameter	Contains the objects mapped on the RPDO1 ( <i>see chapter 10.2.6.1</i> ).
1602 <sub>h</sub>	3 <sup>rd</sup> Receive PDO Mapping Parameter	Contains the objects mapped on the RPDO3 ( <i>see chapter 10.2.6.2</i> ).
1603 <sub>h</sub>	4 <sup>th</sup> Receive PDO Mapping Parameter	Contains the objects mapped on the RPDO4 ( <i>see chapter 10.2.6.3</i> ).
1800 <sub>h</sub>	1 <sup>st</sup> Transmit PDO parameter	Setting of the communication parameters of the TPDO1.
1802 <sub>h</sub>	3 <sup>rd</sup> Transmit PDO Parameter	Setting of the communication parameters of the TPDO3.
1803 <sub>h</sub>	4 <sup>th</sup> Transmit PDO Parameter	Setting of the communication parameters of the TPDO4.
1A00 <sub>h</sub>	1 <sup>st</sup> Transmit PDO Mapping Parameter	Contains the objects mapped on the TPDO1 ( <i>see chapter 10.2.5.1</i> ).
1A02 <sub>h</sub>	3 <sup>rd</sup> Transmit PDO Mapping Parameter	Contains the objects mapped on the TPDO3 ( <i>see chapter 10.2.5.2</i> ).
1A03 <sub>h</sub>	4 <sup>th</sup> Transmit PDO Mapping Parameter	Contains the objects mapped on the TPDO4 ( <i>see chapter 10.2.5.3</i> ).

2100 <sub>h</sub>	Can-Baud rate	Setting of the CAN baud rate of the AG02.
2101 <sub>h</sub>	Node-ID	Setting of the node address of the AG02.
2102 <sub>h</sub>	Transfer reduction	Contains the gear reduction of the AG02.
2410 <sub>h</sub>	Motor Parameter Set	Setting of the control parameters of the drive controller (see chapter 8: parameter description).
2412 <sub>h</sub>	Spindle Pitch Set	Setting of the spindle pitch (see chapter 8: parameter 13).
2413 <sub>h</sub>	Pos Type Set	Setting of the positioning type (see chapter 8: parameter 19).
2415 <sub>h</sub>	Delta Jog Set	Setting of the travel distance in inching operation 1 (see chapter 8: parameter 17).
6040 <sub>h</sub>	Controlword	Contains the control word of the state machine for drives
6041 <sub>h</sub>	Statusword	Contains the status word of the state machine for drives.
6060 <sub>h</sub>	Modes of Operation	Setting of the operation mode: Profile Position Mode / Profile Velocity Mode (see chapter 8: parameter 20).
6064 <sub>h</sub>	Position Actual Value	Contains the absolute actual position in the operation mode Profile Position Mode (positioning mode).
6067 <sub>h</sub>	Position Window	Setting of the tolerance window (see chapter 8: parameter 10).
606C <sub>h</sub>	Velocity Actual Value	Contains the actual velocity in the operation mode Profile Velocity Mode (velocity mode).
607A <sub>h</sub>	Target Position	Contains the target position in the operation mode Profile Position Mode (positioning mode).
607C <sub>h</sub>	Calibration Value	Calibration of the AG02 (see chapter 8: parameter 14 and chapter 3: Calibration).
607D <sub>h</sub>	Software Position Limit	Setting of the limits (see chapter 8: parameter 15/16).
607E <sub>h</sub>	Polarity	Setting of the polarity (sense of rotation) of the AG02 (see chapter 8: parameter 18).
6091 <sub>h</sub>	Gear Ratio	Setting of gear ratio (see chapter 8: parameter 11/12).
60FF <sub>h</sub>	Target Velocity	Contains the target velocity in the operation mode Profile Velocity Mode (velocity mode).

Table 28: Overview of objects

## 10.12.2 Description of objects

Below, all objects of the positioning drive AG02 are described sorted by their indexes.

### 10.12.2.1 1000<sub>h</sub>: Device Type

Object 1000<sub>h</sub> indicates the device profile number.

Subindex	00 <sub>h</sub>
Description	Information on the device profile
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	00000192 <sub>h</sub>
EEPROM	no

#### Format description

Bit 31 - 24	Manufacturer-specific (not used)
Bit 23 - 16	drive type (not used)
Bit 15 - 0	device profile number



### 10.12.2.2 1001<sub>h</sub>: Error Register

Object 1001<sub>h</sub> indicates the error state of the device.

Subindex	00 <sub>h</sub>
Description	current error code
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	no
EEPROM	no

#### Format description

Bit 7:	Error when reading the absolute position
Bit 6:	reserved (statically 0)
Bit 5:	Drive shaft blocked
Bit 4:	Error during data transfer
Bit 3:	Overtemperature in output stage
Bit 2:	Undervoltage in supply voltage
Bit 1:	Overcurrent
Bit 0:	General error (is set when error is present)

A detailed error evaluation can be attained via object 1003<sub>h</sub> (predefined error field). Faults and errors are signalled at the time of their occurrence by an emergency message (see chapter 10.2.9: *Emergency Object*).

### 10.12.2.3 1002<sub>h</sub>: Manufacturer Status Register

Object 1002<sub>h</sub> indicates the system status word of the AG02 (see chapter 2.1: *System Status Word*).

Subindex	00 <sub>h</sub>
Description	Manufacturer-specific status register
Access	read-only
PDO mapping	no
Data type	Unsigned16
Default	no
EEPROM	no

#### Format description

Bit 31 - 16	not used
Bit 15 - 0	see chapter 2.1: System Status Word

### 10.12.2.4 1003<sub>h</sub>: Pre-defined Error Field

Object 1003<sub>h</sub> stores the last 10 causes of faults.

The entry under sub-index 00<sub>h</sub> contains the number of faults currently stored.

The latest fault message is stored under sub-index 01<sub>h</sub>.

Writing a '0' on sub-index 00<sub>h</sub> resets the fault buffer.

Subindex	00 <sub>h</sub>
Description	number of fault messages
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	0
EEPROM	no
Value range	0 ... 10

Subindex	01 <sub>h</sub> ... 0A <sub>h</sub>
Description	faults that occurred
Access	read-only
PDO mapping	no
Data type	Unsigned16
Default	no
EEPROM	yes

#### Format description

Bit 31 - 16	not used
Bit 15 - 8	error code high byte (see chapter 10.2.9.1)
Bit 7 - 0	error code low byte (see chapter 10.2.9.1)

### 10.12.2.5 1005<sub>h</sub>: COB-ID Sync Message

The COB-ID of the SYNC object is set via object 1005<sub>h</sub>.  
The SYNC message is sent to all network participant (broadcast object).

Subindex	00 <sub>h</sub>
Description	COB ID SYNC message
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	80 <sub>h</sub>
EEPROM	no

#### Format description

Bit 31 - 30	not used
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 -11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 0	bit 10 ...0 of the identifier



**The COB-ID of the Sync message can be changed in the NMT status 'PREOPERATIONAL' only!**

### 10.12.2.6 100A<sub>h</sub>: Manufacturer Software Version

Object 100A<sub>h</sub> indicates the software version of the controller firmware.

Subindex	00 <sub>h</sub>
Description	Software version as ASCII characters
Access	read-only
PDO mapping	no
Data type	Unsigned32 (max. 4 characters)
Default	no
EEPROM	no

Format description (example)

Bit 31 - 24	31 <sub>h</sub> = ' 1 ' (ASCII character, ISO 8859)
Bit 23 - 16	2E <sub>h</sub> = ' . ' (ASCII character, ISO 8859)
Bit 15 - 8	30 <sub>h</sub> = ' 0 ' (ASCII character, ISO 8859)
Bit 7 - 0	30 <sub>h</sub> = ' 0 ' (ASCII character, ISO 8859)

Software version = V 1.00

### 10.12.2.7 100C<sub>h</sub>: Guard Time

The cycle time 'Guard Time' for node guarding is set via object 100C<sub>h</sub>.  
The cycle time 'Guard Time' is given in milliseconds (see chapter 10.2.11: Node Guarding).

Subindex	00 <sub>h</sub>
Description	Guard Time
Access	read-write
PDO mapping	no
Unit	milliseconds
Data type	Unsigned16
Default	no
EEPROM	no

Data description

Value '0' means that node guarding is deactivated.

### 10.12.2.8 100D<sub>h</sub>: Life Time Factor

The time interval 'Life Time' for life guarding is set via object 100D<sub>h</sub> (see chapter 10.2.11: Node Guarding).

Subindex	00 <sub>h</sub>
Description	Life Time Factor
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	0
EEPROM	no

### Data description

Value '0' means that Life Guarding is deactivated.

#### 10.12.2.9 1011<sub>n</sub>: Restore Default Parameters

The default values of the changeable parameters can be retrieved via object 1011<sub>n</sub>. Parameter ranges are specified by selecting the respective subindex.

Sub-index 01<sub>n</sub>: set all parameters to default values.

Sub-index 02<sub>n</sub>: set only standard parameters to default values.

Sub-index 03<sub>n</sub>: set only control parameters to default values.

Sub-index 04<sub>n</sub>: calibrate the AG02.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	4
EEPROM	no

Subindex	01 <sub>h</sub>
Description	set all parameters to default values
Access	write-only
PDO mapping	no
Data type	Unsigned32
Default	no
EEPROM	no

Subindex	02 <sub>h</sub>
Description	set standard parameters to default values
Access	write-only
PDO mapping	no
Data type	Unsigned32
Default	no
EEPROM	no

Subindex	03 <sub>h</sub>
Description	set controller parameters to default values
Access	write-only
PDO mapping	no
Data type	Unsigned32
Default	no
EEPROM	no

Subindex	04 <sub>h</sub>
Description	calibrate the AG02
Access	write-only
PDO mapping	no
Data type	Unsigned32
Default	no
EEPROM	no

#### Format description sub-index 01<sub>h</sub> – 04<sub>h</sub>:

Bit 31 - 24	64 <sub>h</sub> = ' d ' (ASCII character, ISO 8859)
Bit 23 - 16	61 <sub>h</sub> = ' a ' (ASCII character, ISO 8859)
Bit 15 - 8	6F <sub>h</sub> = ' o ' (ASCII character, ISO 8859)
Bit 7 - 0	6C <sub>h</sub> = ' l ' (ASCII character, ISO 8859)

By writing the signature 'load' to a sub-index 0...3, the factory settings (see chapter 8: *Parameter description* ⇒ *Default column*) of the respective parameters are retrieved. By writing the signature 'load' to sub-index 04, the AG02 is calibrated (see chapter 3: *Calibration*).



**Parameter Bus address or CAN baud rate are not changed by retrieval of the factory settings!**

#### 10.12.2.10 1014<sub>h</sub>: COB ID Emergency Message

The COB-ID of the Emergency object is set via object 1014<sub>h</sub> (see chapter 10.2.9: *Emergency Object*).

Subindex	00 <sub>h</sub>
Description	COB ID of the Emergency Message
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	80 <sub>h</sub> + Node-ID
EEPROM	no

#### Format description

Bit 31	not used
Bit 30	0 = device sends no emergency message 1 = device sends emergency message
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 0	bit 10 ...0 of the identifier



**The COB ID of the Emergency message can be changed in the NMT status 'PREOPERATIONAL' only!**

#### 10.12.2.11 1017<sub>h</sub>: Producer Heartbeat Time

The cycle time 'Heartbeat Time' for the heartbeat protocol is set via object 1017<sub>h</sub>. The 'Heartbeat Time' is given in milliseconds (see chapter 10.2.10: *Heartbeat protocol*).

Subindex	00 <sub>h</sub>
Description	Producer Heartbeat Time
Access	read-write
PDO mapping	no
Unit	milliseconds
Data type	Unsigned16
Default	0
EEPROM	no

#### Data description

Value '0' means that the heartbeat protocol is deactivated.

#### 10.12.2.12 1018<sub>h</sub>: Identity Object

The Vendor ID of the manufacturer is indicated via object 1018<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	1
EEPROM	no

Subindex	01 <sub>h</sub>
Description	Vendor ID
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	195 <sub>h</sub> (SIKO GmbH)
EEPROM	no

#### Vendor ID:

The Vendor ID is assigned by the CAN user organization CiA e. V. (CAN in Automation). Vendor ID '195<sub>h</sub>' has been assigned to the company SIKO GmbH.

#### 10.12.2.13 1200<sub>h</sub>: Server SDO Parameter

The COB IDs for the Default Server SDO are indicated via object 1200<sub>h</sub> (*see also chapter 10.2.7: Service Data Objects*).

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	COB ID Master ⇒ AG02 (rx)
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	600 <sub>h</sub> + Node-ID
EEPROM	no

Subindex	02 <sub>h</sub>
Description	COB-ID AG02 ⇒ Master (tx)
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	580 <sub>h</sub> + Node-ID
EEPROM	no

#### Format description

Bit 31	0 = SDO valid 1 = SDO not valid
Bit 30	reserved (statically 0)
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node ID of the identifier

#### Data description

The default DSO cannot be changed (according to the CiA DS-301 Predefined Connection Set).

#### **10.12.2.14 1400<sub>h</sub>: 1<sup>st</sup> Receive PDO Parameter**

The communication parameters of the first Receive PDO (RPDO1) are set via object 1400<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	COB-ID
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	200 <sub>h</sub> + Node-ID
EEPROM	no

Subindex	02 <sub>h</sub>
Description	Transfer type (see chapter 10.2.6.4: Transfer types of Receive PDOs)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	255
EEPROM	no

#### Format description sub-index 01<sub>h</sub>:

Bit 31	not used
Bit 30	not used
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node ID of the identifier

#### Data description sub-index 02<sub>h</sub>:

0	synchronous: RPDO1 is processed only after a SYNC message has been received
1 ... 240	synchronous: identical with value 0
241 ... 251	reserved
252	reserved
253	reserved
254	identical with value 255
255	asynchronous: RPDO1 is immediately processed

#### PDO mapping:

see object 1600<sub>h</sub> (1<sup>st</sup> receive PDO mapping parameter).

#### Processing of PDOs:

Receive PDOs are processed in the NMT status 'OPERATIONAL' only. It is not recommended to change the transfer type of the RPDO1 since otherwise the functioning of the state machine would no longer be ensured.

#### Changing PDO parameters:

PDO parameters can be changed in the NMT status 'PRE-OPERATIONAL' only.

### 10.12.2.15 1402<sub>h</sub>: 3<sup>rd</sup> Receive PDO Parameter

The communication parameters of the third Receive PDO (RPDO3) are set via object 1402<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no



Subindex	01 <sub>h</sub>
Description	COB-ID
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	400 <sub>h</sub> + Node-ID
EEPROM	no

Subindex	02 <sub>h</sub>
Description	Transfer type (see chapter 10.2.6.4: Transfer types of Receive PDOs)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	255
EEPROM	no

#### Format description sub-index 01<sub>h</sub>:

Bit 31	not used
Bit 30	not used
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node ID of the identifier

#### Data description sub-index 02<sub>h</sub>:

0	synchronous: RPDO1 is processed only after a SYNC message has been received
1 ... 240	synchronous: identical with value 0
241 ... 251	reserved
252	reserved
253	reserved
254	identical with value 255
255	asynchronous: RPDO1 is immediately processed

#### PDO mapping:

see object 1602<sub>h</sub> (3<sup>rd</sup> receive PDO mapping parameter).

#### Processing of PDOs:

Receive PDOs are processed in the NMT status 'OPERATIONAL' only.

#### Changing PDO parameters:

PDO parameters can be changed in the NMT status 'PRE-OPERATIONAL' only.

### 10.12.2.16 1403<sub>h</sub>: 4<sup>th</sup> Receive PDO Parameter

The communication parameters of the fourth Receive PDO (RPDO4) are set via object 1403<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	COB ID
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	500 <sub>h</sub> + Node-ID
EEPROM	no

Subindex	02 <sub>h</sub>
Description	Transfer type (see chapter 10.2.6.4: Transfer types of Receive PDOs)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	255
EEPROM	no

Format description sub-index 01<sub>h</sub>:

Bit 31	not used
Bit 30	not used
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node ID of the identifier

Data description sub-index 02<sub>h</sub>:

0	synchronous: RPDO1 is processed only after a SYNC message has been received
1 ... 240	synchronous: identical with value 0
241 ... 251	reserved
252	reserved
253	reserved
254	identical with value 255
255	asynchronous: RPDO1 is immediately processed

PDO mapping:

see object 1603<sub>h</sub> (4<sup>th</sup> receive PDO mapping parameter).

### Processing of PDOs:

Receive PDOs are processed in the NMT status 'OPERATIONAL' only.

### Changing PDO parameters:

PDO parameters can be changed in the NMT status 'PRE-OPERATIONAL' only.

#### 10.12.2.17 1600<sub>h</sub>: 1<sup>st</sup> Receive PDO Mapping Parameter

Object 1600<sub>h</sub> determines the objects that are mapped on the first Receive PDO (RPDO1).

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	1
EEPROM	no

Subindex	01 <sub>h</sub>
Description	first object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60400010 <sub>h</sub>
EEPROM	no

#### Format description sub-index 01<sub>h</sub>:

Bit 31 - 16	Index 16 bits
Bit 15 - 8	Sub-index 8 bits
Bit 7 - 0	8 bits object length

#### Data description

The object cannot be changed (static mapping).

#### Mapped objects:

- object 6040<sub>h</sub> (control word) in bytes 0 and 1.

#### 10.12.2.18 1602<sub>h</sub>: 3<sup>rd</sup> Receive PDO Mapping Parameter

Object 1602<sub>h</sub> determines the objects that are mapped on the third Receive PDO (RPDO3).

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	first object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60400010 <sub>h</sub>
EEPROM	no

Subindex	02 <sub>h</sub>
Description	second object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	607A0020 <sub>h</sub>
EEPROM	no

Format description sub-index 01<sub>h</sub> – 02<sub>h</sub>:

Bit 31 -16	index 16 Bit
Bit 15 -8	sub-index 8 bits
Bit 7 0	8 bits object length

#### Data Description:

The object cannot be changed (static mapping).

#### Mapped objects:

- object 6040<sub>h</sub> (control word) in bytes 0 and 1.
- object 607A<sub>h</sub> (target position) in bytes 2 to 5.

### 10.12.2.19 1603<sub>h</sub>: 4<sup>th</sup> Receive PDO Mapping Parameter

Object 1603<sub>h</sub> determines the objects that are mapped on the fourth Receive PDO (RPDO4).

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	first object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60400010 <sub>h</sub>
EEPROM	no

Subindex	02 <sub>h</sub>
Description	second object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60FF0020 <sub>h</sub>
EEPROM	no

Format description sub-index 01<sub>h</sub> – 02<sub>h</sub>:

Bit 31 - 16	index 16 bits
Bit 15 - 8	sub-index 8 bits
Bit 7 - 0	8 bits object length

Data description:

The object cannot be changed (static mapping).

Mapped objects:

- object 6040<sub>h</sub> (control word) in bytes 0 and 1.
- object 60FF<sub>h</sub> (target velocity) in bytes 2 to 5.

#### 10.12.2.20 1800<sub>h</sub>: 1<sup>st</sup> Transmit PDO Parameter

The communication parameters of the first Transmit PDO (TPDO1) are set via object 1800<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	largest sub-index supported
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	5
EEPROM	no

Subindex	01 <sub>h</sub>
Description	COB-ID
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	180 <sub>h</sub> + Node-ID
EEPROM	no

Subindex	02 <sub>h</sub>
Description	Transfer type (see chapter 10.2.5.4: Transfer types of Transmit PDOs)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	255
EEPROM	no

Subindex	05 <sub>h</sub>
Description	Event Timer
Access	read-write
PDO mapping	no
Unit	milliseconds
Data type	Unsigned16
Default	100
EEPROM	no
Value range	10 ... 65535

#### Format description sub-index 01<sub>h</sub>:

Bit 31	not used
Bit 30	0 = request via RTR frame enabled 1 = request via RTR frame disabled
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node-ID of the identifier

#### Data description sub-index 02<sub>h</sub>:

0	synchronous: acyclic, PDO is sent after each SYNC message.
1 ... 240	synchronous: cyclic, PDO is sent after 1 ... 240 SYNC messages received.
241 ... 251	reserved
252	reserved
253	asynchronous: on request (RTR frame). PDO is sent immediately after receipt of the RTR frame. <b>Attention!</b> Must have been enabled via bit 30 of sub-index 1.
254	identical with value 255
255	asynchronous: time-triggered

#### PDO mapping:

see object 1A00<sub>h</sub> (*1<sup>st</sup> transmit PDO mapping parameter*).

#### Event Timer:

The 'Event Timer' parameter serves for setting a cycle time (in milliseconds) for time-triggered transfer of the Transmit PDO1.

#### Processing of PDOs:

Transmit PDOs are transferred in the NMT status 'OPERATIONAL' only.  
It is not recommended to change the transfer type of the TPDO1 since otherwise the functioning of the state machine would no longer be ensured.

#### Changing PDO parameters:

PDO parameters can be changed in the NMT status 'PRE-OPERATIONAL' only.

### 10.12.2.21 1802<sub>h</sub>: 3<sup>rd</sup> Transmit PDO Parameter

The communication parameters of the third Transmit PDO (TPDO3) are set via object 1802<sub>h</sub> (TPDO3) eingestellt.

Subindex	00 <sub>h</sub>
Description	largest sub-index supported
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	5
EEPROM	no

Subindex	01 <sub>h</sub>
Description	COB-ID
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	380 <sub>h</sub> + Node-ID
EEPROM	no

Subindex	02 <sub>h</sub>
Description	Transfer type (see chapter 10.2.5.4: Transfer types of Transmit PDOs)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	253
EEPROM	no

Subindex	05 <sub>h</sub>
Description	Event Timer
Access	read-write
PDO mapping	no
Unit	milliseconds
Data type	Unsigned16
Default	100
EEPROM	no
Value range	10 ... 65535

Format description sub-index 01<sub>h</sub>:

Bit 31	not used
Bit 30	0 = request via RTR frame enabled 1 = request via RTR frame disabled
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node-ID of the identifier

#### Data description sub-index 02<sub>h</sub>:

0	synchronous: acyclic, PDO is sent after each SYNC message.
1 ... 240	synchronous: cyclic, PDO is sent after 1 ... 240 SYNC messages received.
241 ... 251	reserved
252	reserved
253	asynchronous: on request (RTR frame). PDO is sent immediately after receipt of the RTR frame. <b>Attention!</b> Must have been enabled via bit 30 of sub-index 1.
254	asynchronous: event-triggered (with every change of the position value)
255	asynchronous: time-triggered

#### PDO mapping:

see object 1A02<sub>h</sub> (3<sup>rd</sup> transmit PDO mapping parameter).

#### Event Timer:

The 'Event Timer' parameter serves for setting a cycle time (in milliseconds) for time-triggered transfer of the Transmit PDO3.

#### Processing of PDOs:

Transmit PDOs are transferred in the NMT status 'OPERATIONAL' only.

#### Changing PDO parameters:

PDO parameters can be changed in the NMT status 'PRE-OPERATIONAL' only.

### 10.12.2.22 1803<sub>h</sub>: 4<sup>th</sup> Transmit PDO Parameter

The communication parameters of the fourth Transmit PDO (TPDO4) are set via object 1803<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	largest sub-index supported
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	5
EEPROM	no

Subindex	01 <sub>h</sub>
Description	COB-ID
Access	read-write
PDO mapping	no
Data type	Unsigned32
Default	480 <sub>h</sub> + Node-ID
EEPROM	no



Subindex	02 <sub>h</sub>
Description	Transfer type (see chapter 10.2.5.4: Transfer types of Transmit PDOs)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	253
EEPROM	no

Subindex	05 <sub>h</sub>
Description	Event Timer
Access	read-write
PDO mapping	no
Unit	milliseconds
Data type	Unsigned16
Default	100
EEPROM	no
Value range	10 ... 65535

#### Format description sub-index 01<sub>h</sub>:

Bit 31	not used
Bit 30	0 = request via RTR frame enabled 1 = request via RTR frame disabled
Bit 29	0 = 11-bit identifier (CAN 2.0A) 1 = 29-bit identifier (CAN 2.0B)
Bit 28 - 11	if bit 29=1, bit 28 ... 11 of the 29-bit identifier
Bit 10 - 7	4 bit function code of the identifier
Bit 6 - 0	7 bit Node-ID of the identifier

#### Data description sub-index 02<sub>h</sub>:

0	synchronous: acyclic, PDO is sent after each SYNC message.
1 ... 240	synchronous: cyclic, PDO is sent after 1 ... 240 SYNC messages received.
241 ... 251	reserved
252	reserved
253	asynchronous: on request (RTR frame). PDO is sent immediately after receipt of the RTR frame. <b>Attention!</b> Must have been enabled via bit 30 of sub-index 1.
254	asynchronous: Event-triggered (with every change of actual velocity).
255	asynchronous: time-triggered

#### PDO mapping:

see object 1A03<sub>h</sub> (4<sup>th</sup> transmit PDO mapping parameter).

#### Event Timer:

The 'Event Timer' parameter serves for setting a cycle time (in milliseconds) for time-triggered transfer of the Transmit PDO4.

### Processing of PDOs:

Transmit PDOs are transferred in the NMT status 'OPERATIONAL' only.

### Changing PDO parameters:

PDO parameters can be changed in the NMT status 'PRE-OPERATIONAL' only.

## 10.12.2.23 1A00<sub>h</sub>: 1<sup>st</sup> Transmit PDO Mapping Parameter

Object 1A00<sub>h</sub> determines the objects that are mapped on the first Transmit PDO (TPDO1).

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	1
EEPROM	no

Subindex	01 <sub>h</sub>
Description	first object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60410010 <sub>h</sub>
EEPROM	no

### Format description sub-index 01<sub>h</sub>:

Bit 31 - 16	index 16 bits
Bit 15 - 8	sub-index 8 bits
Bit 7 - 0	8 bits object length

### Data description:

The object cannot be changed (static mapping).

### Mapped objects:

- object 6041<sub>h</sub> (status word) in bytes 0 and 1.

## 10.12.2.24 1A02<sub>h</sub>: 3<sup>rd</sup> Transmit PDO Mapping Parameter

Object 1A02<sub>h</sub> determines the objects that are mapped on the third Transmit PDO (TPDO3).

Subindex	00 <sub>h</sub>
Description	Number of sub-indexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	first object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60410010 <sub>h</sub>
EEPROM	no

Subindex	02 <sub>h</sub>
Description	second object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60640020 <sub>h</sub>
EEPROM	no

Format description sub-index 01<sub>h</sub> – 02<sub>h</sub>:

Bit 31 - 16	index 16 bits
Bit 15 - 8	sub-index 8 bits
Bit 7 - 0	8 bits object length

Data description:

The object cannot be changed (static mapping).

Mapped objects:

- object 6041<sub>h</sub> (status word) in bytes 0 and 1.
- object 6064<sub>h</sub> (position actual value) in bytes 2 to 5.

#### 10.12.2.25 1A03<sub>h</sub>: 4<sup>th</sup> Transmit PDO Mapping Parameter

Object 1A03<sub>h</sub> determines the objects that are mapped on the fourth Transmit PDO (TPDO4).

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	first object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	60410010 <sub>h</sub>
EEPROM	no

Subindex	02 <sub>h</sub>
Description	second object mapped
Access	read-only
PDO mapping	no
Data type	Unsigned32
Default	606C0020 <sub>h</sub>
EEPROM	no

Format description sub-index 01<sub>h</sub> – 02<sub>h</sub>:

Bit 31 - 16	index 16 bits
Bit 15 - 8	sub-index 8 bits
Bit 7 - 0	8 bits object length

Data description:

The object cannot be changed (static mapping).

Mapped objects:

- object 6041<sub>h</sub> (status word) in bytes 0 and 1.
- object 606C<sub>h</sub> (velocity actual value) in bytes 2 to 5.

#### 10.12.2.26 2100<sub>h</sub>: CAN baud rate

The CAN baud rate is set via object 2100<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	CAN baud rate (see chapter 8: Parameter description ⇒ parameter no. 23)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	10
EEPROM	yes
Value range	0 ... 11

Data description

Value = 0:	15.625	kBaud
Value = 1:	20	kBaud
Value = 2:	25	kBaud
Value = 3:	40	kBaud
Value = 4:	50	kBaud
Value = 5:	62.5	kBaud
Value = 6:	100	kBaud
Value = 7:	125	kBaud
Value = 8:	200	kBaud
Value = 9:	250	kBaud
Value = 10:	500	kBaud
Value = 11:	1000	kBaud



The baud rate is taken over only after the next initialization of the positioning drive.  
The master must be adjusted to the new baud rate as well!

#### 10.12.2.27 2101<sub>h</sub>: Node-ID

The Node-ID of the AG02 is set via object 2101<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Node-ID (see chapter 8: Parameter description ⇒ parameter no. 22)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	127
EEPROM	yes
Value range	1 ... 127



The address is taken over only after the next initialization of the positioning drive!

#### 10.12.2.28 2102<sub>h</sub>: Gear reduction

The gear reduction of the AG02 can be read out via object 2102<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Gear reduction
Access	read
PDO mapping	no
Data type	Unsigned8
Default	no
EEPROM	no
Value range	1 ... 2

##### Data description:

Value = 1: Gear reduction 55:1  
Value = 2: Gear reduction 62:1  
Value = 3: Gear reduction 135:1

#### 10.12.2.29 2410<sub>h</sub>: Motor Parameter Set

Object 2410<sub>h</sub> contains all adjustable control parameters of the drive controller.

Subindex	00 <sub>h</sub>
Description	largest sub-index supported
Access	read-only
PDO mapping	no
Data type	Unsigned8
Default	9
EEPROM	no

Subindex	01 <sub>h</sub>
Description	Controller parameter P (see chapter 8: Parameter description ⇒ parameter no. 1)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	250
EEPROM	yes
Value range	1 ... 500

Subindex	02 <sub>h</sub>
Description	Controller parameter I (see chapter 8: Parameter description ⇒ parameter no. 2)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	5
EEPROM	yes
Value range	0 ... 500

Subindex	03 <sub>h</sub>
Description	Controller parameter D (see chapter 8: Parameter description ⇒ parameter no. 3)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	0
EEPROM	yes
Value range	0 ... 500

Subindex	04 <sub>h</sub>
Description	a- Pos (acceleration positioning mode) (see chapter 8: Parameter description ⇒ parameter no. 4)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	50
EEPROM	yes
Value range	1 ... 100

Subindex	05 <sub>h</sub>
Description	v- Pos (velocity positioning mode) (see chapter 8: Parameter description ⇒ parameter no. 5)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	50
EEPROM	yes
Value range	Gear 55:1 ⇒ 1...100 Gear 62:1 ⇒ 1...80 Gear 135:1 ⇒ 1...35

Subindex	06 <sub>h</sub>
Description	a – Rot (acceleration velocity mode) (see chapter 8: Parameter description ⇒ parameter no. 6)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	50
EEPROM	yes
Value range	1 ... 100

Subindex	08 <sub>h</sub>
Description	a – Inch (acceleration inching operation) (see chapter 8: Parameter description ⇒ parameter no. 8)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	50
EEPROM	yes
Value range	1 ... 100

Subindex	09 <sub>h</sub>
Description	v – Inch (velocity inching operation) (see chapter 8: Parameter description ⇒ parameter no. 9)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	50
EEPROM	yes
Value range	Gar 55:1 ⇒ 1...100 Gar 62:1 ⇒ 1...80 Gar 135:1 ⇒ 1...35

### 10.12.2.30 2412<sub>h</sub>: Spindle Pitch Set

Spindle pitch is set via object 2412<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Spindle pitch (see chapter 8: Parameter description ⇒ parameter no. 13)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	0
EEPROM	yes
Value range	0 ... 1000

### 10.12.2.31 2413<sub>h</sub>: Pos Type Set

The positioning type is set via object 2413<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Positioning type (see chapter 8: Parameter description ⇒ parameter no. 19)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	0
EEPROM	yes
Value range	0 ... 2

#### Data description:

Value = 0: direct  
 Value = 1: loop +  
 Value = 2: loop –

### 10.12.2.32 2415<sub>h</sub>: Delta Jog Set

The travel distance in inching operation 1 is set via object 2415<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Travel distance inching operation 1 (see chapter 8: Parameter description ⇒ parameter no. 17)
Access	read-write
PDO mapping	no
Data type	Integer32
Default	1600
EEPROM	yes
Value range	-1000000 ... +1000000

### 10.12.2.33 6040<sub>h</sub>: Control word

The object 6040<sub>h</sub> is the control word of the state machine for drives according to the device profile CiA DSP-402.

Subindex	00 <sub>h</sub>
Description	control word
Access	read-write
PDO mapping	yes
Data type	Unsigned16
Default	no
EEPROM	no

#### Format description:

see chapter 10.5: Control word

#### PDO mapping:

The control word is mapped on the three Receive PDOs (see objects 1600<sub>h</sub> – 1603<sub>h</sub>).



### 10.12.2.34 6041<sub>h</sub>: Status word

The object 6041<sub>h</sub> is the status word of the state machine for drives according to the device profile CiA DSP-402.

Subindex	00 <sub>h</sub>
Description	Status word
Access	read-only
PDO mapping	yes
Data type	Unsigned16
Default	no
EEPROM	no

#### Format description:

see chapter 10.4: Status word

#### PDO mapping:

The status word is mapped on the three Transmit PDOs (see objects 1A00<sub>h</sub> – 1A03<sub>h</sub>).

### 10.12.2.35 6060<sub>h</sub>: Modes of Operation

The operation mode of the AG02 is set via object 6060<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Operating mode (see chapter 8: Parameter description ⇒ parameter no. 20)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	1
EEPROM	yes
Value range	1 ... 2

#### Data description:

Value = 1: Profile Position Mode (positioning mode)

Value = 2: Profile Velocity Mode (velocity mode)

### 10.12.2.36 6064<sub>h</sub>: Position Actual Value

Object 6064<sub>h</sub> contains the actual position value in the Profile Position Mode (positioning mode).

Subindex	00 <sub>h</sub>
Description	absolute position value in the positioning mode
Access	read-only
PDO mapping	yes
Data type	Integer32
Default	no
EEPROM	no

PDO mapping:

The absolute position value and the status word of the state machine are mapped on the Transmit PDO3, see object 1A02<sub>h</sub> (3<sup>rd</sup> Transmit PDO mapping parameter).

#### 10.12.2.37 6067<sub>h</sub>: Position Window

A symmetric range of tolerable positions for standstill monitoring in the target point of positioning is set via object 6067<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Pos- window (see chapter 8: Parameter description ⇒ parameter no. 10)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	10
EEPROM	yes
Value range	0 ... 1000

#### 10.12.2.38 606C<sub>h</sub>: Velocity Actual Value

Object 606C<sub>h</sub> contains the actual velocity in the Profile Velocity Mode (velocity mode).

Subindex	00 <sub>h</sub>
Description	Actual velocity in velocity mode
Access	read-only
PDO mapping	yes
Data type	Integer32
Default	no
EEPROM	no

PDO mapping:

The actual velocity and the status word of the state machine are mapped on the Transmit PDO4, see object 1A03<sub>h</sub> (4<sup>th</sup> Transmit PDO mapping parameter).

#### 10.12.2.39 607A<sub>h</sub>: Target Position

The target position of a drive movement in the operation mode Profile Position Mode (position mode) is entered via object 607A<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	setpoint in the positioning mode
Access	read-write
PDO mapping	yes
Data type	Integer32
Default	no
EEPROM	no
Value range	±9999999

### PDO mapping:

The target position and the control word of the state machine are mapped on the Receive PDO3, see object 1602<sub>h</sub> (3<sup>rd</sup> Receive PDO mapping parameter).

#### 10.12.2.40 607C<sub>h</sub>: Calibration Value

The calibration value is programmed and the programmed calibration value taken over as the absolute position value via object 607C<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Calibration value (see chapter 8: Parameter description ⇒ parameter no. 14 and chapter 3: Calibration)
Access	read-write
PDO mapping	no
Data type	Integer32
Default	0
EEPROM	yes
Value range	±999999

#### 10.12.2.41 607D<sub>h</sub>: Software Position Limit

Via object 607D<sub>h</sub>, the software limit switches defining the working range of the drive are set.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Integer32
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	lower limit: (see chapter 8: Parameter description ⇒ parameter no. 16)
Access	read-write
PDO mapping	no
Data type	Integer32
Default	-1000000
EEPROM	yes
Value range	±9999999

Subindex	02 <sub>h</sub>
Description	upper limit (see chapter 8: Parameter description ⇒ parameter no. 15)
Access	read-write
PDO mapping	no
Data type	Integer32
Default	+1000000
EEPROM	yes
Value range	±9999999

#### 10.12.2.42 607E<sub>n</sub>: Polarity

The polarity of the drive's sense of rotation is set via object 607E<sub>n</sub>.

Subindex	00 <sub>h</sub>
Description	Sense of rotation (see chapter 8: Parameter description ⇒ parameter no. 18)
Access	read-write
PDO mapping	no
Data type	Unsigned8
Default	0
EEPROM	yes
Value range	0 ... 1

##### Data description:

Value '0' = sense of rotation 'i'  
Value '1' = sense of rotation 'e'

#### 10.12.2.43 6091<sub>n</sub>: Gear Ratio

A gear ratio can be programmed via object 6091<sub>n</sub>.

Subindex	00 <sub>h</sub>
Description	Number of subindexes
Access	read-only
PDO mapping	no
Data type	Unsigned16
Default	2
EEPROM	no

Subindex	01 <sub>h</sub>
Description	Numerator gear ratio (see chapter 8: Parameter description ⇒ parameter no. 11)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	1
EEPROM	yes
Value range	1 ... 10000

Subindex	02 <sub>h</sub>
Description	Denominator gear ratio (see chapter 8: Parameter description ⇒ parameter no. 12)
Access	read-write
PDO mapping	no
Data type	Unsigned16
Default	1
EEPROM	yes
Value range	1 ... 10000

see also chapter 4: External gear unit.



#### 10.12.2.44 60FF<sub>h</sub>: Target Velocity

The target velocity of a drive movement in the operation mode Profile Velocity Mode (velocity mode) is entered via object 60FF<sub>h</sub>.

Subindex	00 <sub>h</sub>
Description	Target velocity in the velocity mode of operation
Access	read-write
PDO mapping	yes
Data type	Integer32
Default	no
EEPROM	no
Value range	Gear 55:1 ⇒ ±100 Gear 62:1 ⇒ ±80 Gear 135:1 ⇒ ±35

#### PDO mapping:

The target velocity and the control word of the state machine are mapped on the Receive PDO4, see object 1603<sub>h</sub> (4<sup>th</sup> Receive PDO mapping parameter).