SIEB & MEYER

Drive System SD2

DNC Object Access

P-TD-0000345.2
2016-02-03

Copyright

Trademarks

SIEB & MEYER worldwide

For questions regarding our products and technical problems please contact us.

SIEB & MEYER AG
Auf dem Schmaarkamp 21
21339 Lüneburg
Germany

Phone: +49 4131 203 0
Fax: +49 4131 203 2000
support@sieb-meyer.de
http://www.sieb-meyer.de

SIEB & MEYER Shenzhen Trading Co. Ltd.
Room 306, 3rd Floor, Building A1,
Dongjiaotou Industrial Area, Houhai Dadao,
Shekou, Nanshan District,
Shenzhen City, 518067
China

Phone: +86 755 2681 1417 / +86 755 2681 2487
Fax: +86 755 2681 2967
sm.china.support@gmail.com
http://www.sieb-meyer.cn

SIEB & MEYER Asia Co. Ltd.
4 Fl, No. 532, Sec. 1
Min-Sheng N. Road
Kwei-Shan Hsiang
333 Tao-Yuan Hsien
Taiwan

Phone: +886 3 311 5560
Fax: +886 3 322 1224
smasia@ms42.hinet.net
http://www.sieb-meyer.com

SIEB & MEYER USA
3975 Port Union Road
Fairfield, OH 45014
USA

Phone: +1 513 563 0860
Fax: +1 513 563 7576
info@sieb-meyerusa.com
http://www.sieb-meyer.com

1 General Information

The DNC object access protocol is used for diagnosis and parameterization of SD2-drives. A PC can be used for the diagnosis and as parameter interface, interchanging data with the drive via the DNC object access protocol.

The following bus systems are supported for the connection of the drive:

- RS232
- RS485

The communication with the SD2 drive is established according to the master-slave method. The PC acts as master and the drive acts as slave. Thus, the drive is only active on request of the master.

The data are exchanged as data telegrams. The master sends a command telegram to the slave. When this is received the slave returns a response telegram to the master. The data are only exchanged via the bus system, when the master initiates the exchange.

☞ The DNC object access protocol does not have fixed response times. For this reason it should only be used for parameterization and diagnosis purposes of the drive.

2 Characteristics of the RS232 Communication

Consider the following descriptions for serial communication with the SD2.

2.1 Serial Connection

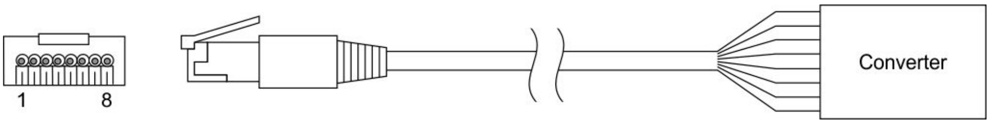
By means of customary PC data can be exchanged with the drive via the serial interface. For this purpose a free serial interface of the PC must be connected to the RS232/RS485 interface of the drive. The type of connection depends on the used drive.

2.1.1 RS232 Connection with SD2

Connect the female RJ45 connector X3 at the front panel of the SD2 drive via an RS232 to RS485 converter to a free serial interface of the PC.

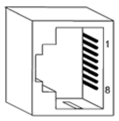
Connecting cable

- shielded round cable
- twisted-pair
- 8-pole male RJ45 connector ↔ open end



Pin assignment on SD2

- 8-pole female RJ45 connector

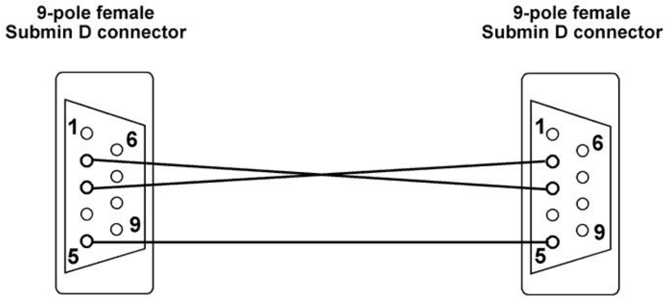
	Pin	I/O	Name	Meaning
	1		–	
	2		–	
	3	I/O	D+	RS485 interface
	4		–	
	5		–	
	6	I/O	D−	RS485 interface
	7	I/O	GND	Ground
	8	I/O	GND	Ground

2.1.2 RS232 Connection with SD2S

Connect X19 (9-pole male submin D connector) at the front panel of the SD2S-drive to a free serial interface of the PC (9-pole male submin D connector).

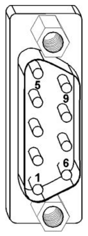
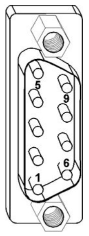
Connecting cable

- shielded round cable
- twisted-pair
- 9-pole female submin D connector ↔ 9-pole female submin D connector



Pin assignment on the device

- 9-pole male submin D connector ↔ 9-pole male submin D connector

SD2S			Name	Description	PC		
	Pin	I/O			I/O	Pin	
	1		–				
	2	I	RxD	Receive data from PC	O	3	
	3	O	TxD	Transmit data to PC	I	2	
	4		–				
	5	I/O	GND	Ground	I/O	5	
	6		–				
	7		–				
	8		–				
	9		–				

2.2 Parameters of the Serial Interface

The following settings are required for the serial interface:

- Baud rate: 57600 bit/s
- Data bits: 8
- Parity: None
- Stop bits: One

2.3 Time Response

When a DNC command block has been transmitted, the Drive will respond max. 250 µs after having received the last byte of the command block.

Please note that the transmission time depends on the quantity of data. Therefore the PC software *drivemaster2* uses a timeout of 1 s to make sure that the serial connection is definitely interrupted.

3 Data Types

The communication with the SD2 is carried out by means of DNC commands. These commands allow data exchange with the device via memory blocks.

Depending on the DNC command, memory blocks are organized bytewise, wordwise or 3-bytewise. The following applies for a memory block of the length n:

- <byte 0, word 0, three byte 0> = byte with the lowest address
- <byte n-1, word n-1, three byte n-1 > = byte with the highest address

That means: During sequential transmission of the blocks the <byte 0> must be transmitted first and the <byte n-1> must be transmitted last.

3.1 1 Byte Data Types

One byte is the smallest data format that can be transmitted with an DNC-command. A distinction is drawn between unsigned and signed 1 byte data types:

T36_S36_SHORTCARD	unsigned 8-bit number (0 … 255)
T36_S36_SHORTINT	signed 8-bit number (-128 … 127)

3.2 2 Byte Data Types

2 byte data types are coded as below:

Byte 0	Byte 1						
Bit 0 … 7	Bit 8 … 15						

A distinction is drawn between unsigned and signed data types:

T36_S36_CARDINAL	unsigned 16-bit number (0 … 65535)
T36_S36_INTEGER	signed 16-bit number (-32768 … 32767)

3.3 4 Byte Data Types

4 byte data types are coded as below:

Byte 0	Byte 1	Byte 2	Byte 3				
Bit 0 … 7	Bit 8 … 15	Bit 16 … 23	bit 24 … 31				

A distinction is drawn between unsigned and signed data types:

T36_S36_LONGCARD	unsigned 32-bit number (0 … 4,294,967,295)
T36_S36_LONGINT	signed 32-bit number (-2,147,483,647 … 2,147,483,647)

3.4 3 Byte Data Types

The devices of the series SD2 partly use 3-byte data items for internal processes.

Byte 0	Byte 1	Byte 2					
Bit 0 … 7	Bit 8 … 15	Bit 16 … 23					

The PC can not process this data type. Since some of the DNC commands use 3-byte data items, they must be represented by the existing PC data types. One 3-byte data item, for example, can be made up of 3 bytes of the type *T36_S36_SHORTCARD*. Two 3-byte data items can be replaced by three words of the type *T36_S36_CARDINAL*. And four 3-Byte data items can be represented by three double words of the type *T36_S36_LONGCARD*.

4 DNC Commands

In general the DNC communication with the Drive is established by the exchange of data blocks between the DNC master (PC or PLC) and the slave (Drive).

A DNC command has the following structure:

- Command block (sent from the master to the Drive)
- Response block (sent from the Drive to the master)

A DNC command is always started by the master. The Drive only can react to the DNC command.

The command and response blocks have a different structure and meaning for each DNC command. The command block describes the actual DNC command and may contain so-called sub commands.

4.1 General Structure of the Command Interface

Below the general structure of the command and reply telegram is described.

4.1.1 Command Block

Offset	Type	Name	Description
0x00	T36_S36_SHORTCARD	zero	Start signal for a command transmission: Always set to zero.
0x01	T36_S36_SHORTCARD	length	Protocol length in bytes: Calculated by the number of transmitted data, but only the leading zero, the checksum and the length specification. The smallest length is 3.
0x02	T36_S36_SHORTCARD	dest	Destination of the command block: Contains the desired module number plus 2.
0x03	T36_S36_SHORTCARD	source	Transmitter of the command: The value 1 is entered for the PC or the PLC.
0x04	T36_S36_SHORTCARD	cmd	Command number of the actual DNC command
0x05	T36_S36_SHORTCARD array with a maximum length of 48 bytes	data	User data of the command: Depending on the command up to 48 bytes, 24 words or 16 3-byte words can be transmitted.
0x05 + length-3	T36_S36_SHORTCARD	check	Check sum: Consists of the ones complement of the sum of the complete data block, except the checksum.

4.1.2 Reply Block

Offset	Type	Name	Description
0x00	T36_S36_SHORTCARD	zero	Start signal for a reply transmission: Always set to zero.
0x01	T36_S36_SHORTCARD	length	Protocol length in bytes: Calculated by the number of transmitted data, but only the leading zero, the checksum and the length specification. The smallest length is 3.
0x02	T36_S36_SHORTCARD	dest	Destination of the reply block: The value 1 is entered for the PC or the PLC.

Offset	Type	Name	Description
0x03	T36_S36_SHORTCARD	source	Transmitter the reply block: Contains the module number of the transmitter plus 2.
0x04	T36_S36_SHORTCARD	cmd	Number of the executed command: Note, that the most significant bit is additionally set to 1.
0x05	T36_S36_SHORTCARD array with a maximum length of 48 bytes	data	User data of the reply block: Depending on the command up to 48 bytes, 24 words or 16 3-byte words can be transmitted.
0x05 + length-3	T36_S36_SHORTCARD	check	Check sum: Consists of the ones complement of the sum of the complete data block, except the checksum.

5 Addressing the Devices

In order to control the drives via PLC or an appropriate PC program the DNC addresses used in the DNC protocol must be known. The DNC drive addresses are derived from the module addresses that are set by means of the address selection switch on each device. Via the module address the software *drivemaster2* communicates with the devices.

For more information on proper addressing of the modules refer to the documentation "drivemaster2 - User Manual", chapter "Communication".

In the DNC protocol each device (double-axis devices and single-axis devices) has always two DNC addresses, that means the master communicates with drive A via the lower DNC address and with drive B via the higher DNC address. Single-axis devices do not have a drive B. Therefore the higher DNC address is not used. The DNC addresses 0 and 1 are reserved for the master. For this reason the numbering starts with the DNC address 2.

The DNC address of a drive can be calculated from the position of the address selection switch on the module by means of the following formula:

DNC address drive A = (address selection switch × 2) + 2

DNC address drive B = (address selection switch × 2) + 3

The following figure illustrates the relation of the drive address used in *drivemaster2* and the drive address used in the DNC protocol:

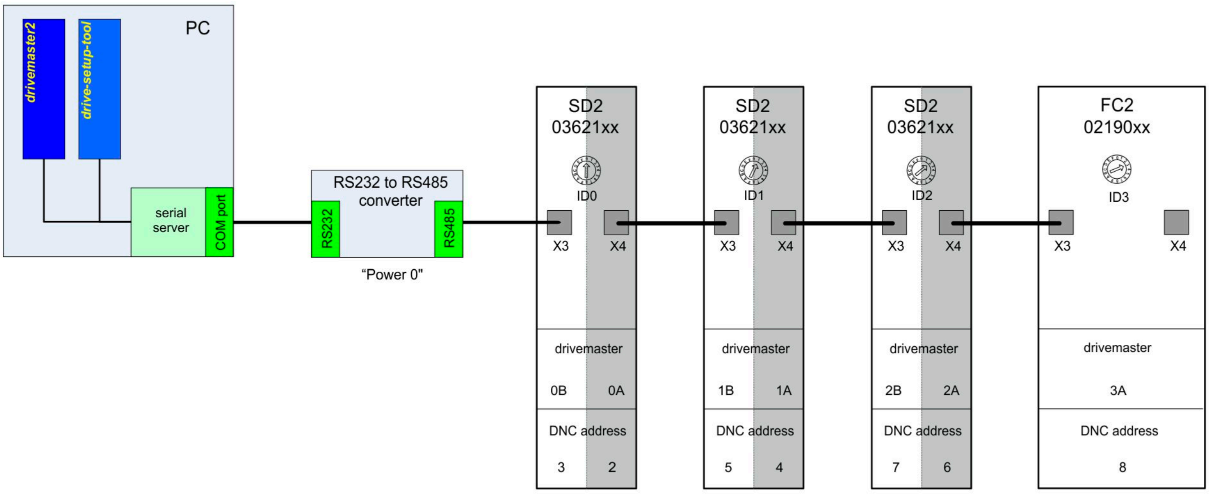


Fig. 1: Drive addresses in drivemaster2 and in the DNC protocol

6 Object Access

The telegrams "Read Wide Object" and "Write Wide Object" allow accessing objects. Each object has an index (object number) and a subindex (object sub-number) for identification.

You can display the object number of each drive object in the software *drivemaster2* (see Appendix, p. 23).

6.1 Read Wide Object

The telegram "Read Wide Object" allows reading an object from the drive.

The object index is a 16 bit value and saved in the bytes "index lo" and "index hi". Array and string objects have a subindex. The subindex is saved as 32 bit value.

The command block is structured as follows:

Name	Value
zero	0
length	9
dest	Module number plus 2
source	1 (for PC)
cmd	0x0d
index lo	Lower byte of the object number
index hi	Higher byte of the object number
subindex 0	Subindex bit [7…0]
subindex 1	Subindex bit [15…8]
subindex 2	Subindex bit [23…16]
subindex 3	Subindex bit [31…24]
check	Checksum

check = 0xFF − (length + dest + source + cmd + index lo + index hi + subindex 0 +subindex 1 + subindex 2 + subindex 3)

The following reply block is returned:

Name	Value
zero	0
length	5 + count
dest	1 (for PC)
source	Module number plus 2
cmd	0x8d
count	Number of read bytes
error code	See Error Codes, p. 21
data	Data 0
⋮	
data	Data (count − 1)
check	Checksum

check = 0xFF – (length + dest + source + cmd + count + errorcode + $\sum_{i=count} data_{(i)}$)

count = 0…48

6.2 Write Wide Object

The telegram "Write Wide Object" allows writing an object into the drive.

The object index is a 16 bit value and saved in the bytes "index lo" and "index hi". Array and string objects have a subindex. The subindex is saved as 32 bit value.

The command block is structured as follows:

Name	Value
zero	0
length	10 + count
dest	Module number plus 2
source	1 (for PC)
cmd	0x0e
index lo	Lower byte of the object number
index hi	Higher byte of the object number
subindex 0	Subindex bit [7…0]
subindex 1	Subindex bit [15…8]
subindex 2	Subindex bit [23…16]
subindex 3	Subindex bit [31…24]
count	Number of bytes to be written
data	Data 0
⋮	
data	Data (count − 1)
check	Checksum

$$\text{check} = \text{0xFF} – (\text{length} + \text{dest} + \text{source} + \text{cmd} + \text{index lo} + \text{index hi} + \text{subindex 0} \\ + \text{subindex 1} + \text{subindex 2} + \text{subindex 3} + \text{count} + \sum_{i=count} data_{(i)})$$

count = 1…48

The following reply block is returned:

Name	Value
zero	0
length	4
dest	1 (for PC)
source	Module number plus 2
cmd	0x8e
error code	See Error Codes, p. 21
check	Checksum

check = 0xFF - (length + dest + source + cmd + errorcode)

6.3 Error Codes of the Service Data Channel

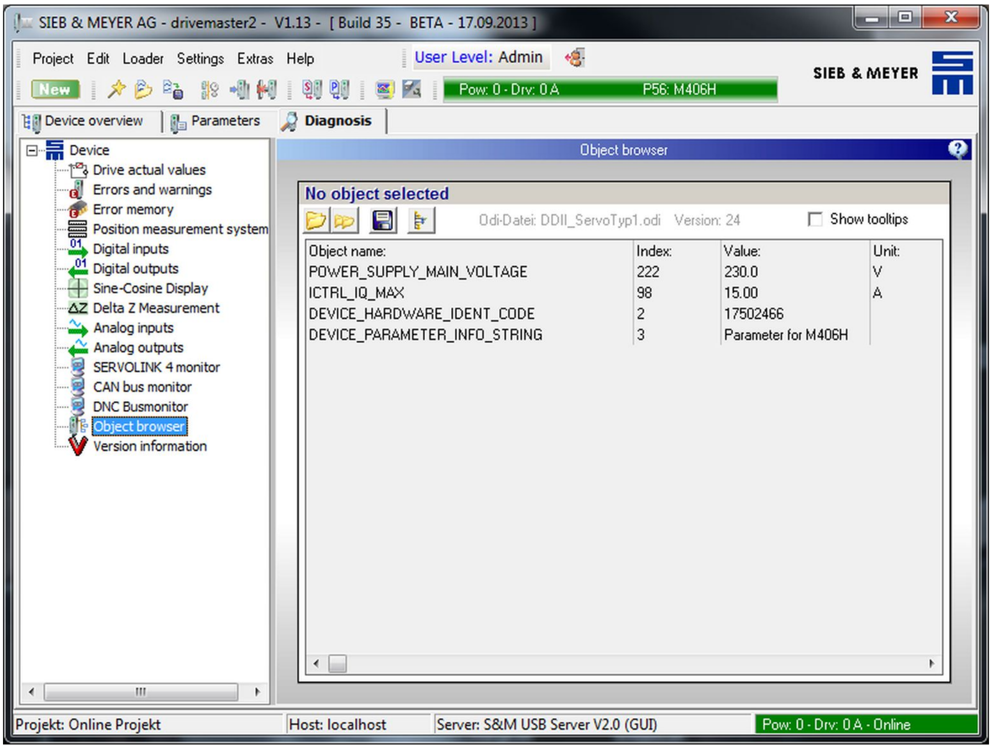
Error code	Description
0x00	No error
0x81 or 0x01	Toggle bit not alternated
0x86 or 0x06	CRC error
0x87 or 0x07	No free memory
0x88 or 0x08	Unsupported access to an object
0x89 or 0x09	Attempt to read a write-only object
0x8A or 0x0A	Attempt to write in a read-only object
0x8B or 0x0B	Object does not exist in the object dictionary!
0x8C or 0x0C	Reserved
0x8D or 0x0D	Reserved
0x8E or 0x0E	General parameter incompatibility reason
0x8F or 0x0F	General internal incompatibility in the device
0x90 or 0x10	Access failed due to hardware error
0x91 or 0x11	Data type not correct, length of the service parameter not correct
0x92 or 0x12	Data type not correct, length of the service parameter too long
0x93 or 0x13	Data type not correct, length of the service parameter too short
0x94 or 0x14	Subindex does not exist
0x95 or 0x15	Value range of the parameter exceeded (only for write access)
0x96 or 0x16	Value of the written parameter too high
0x97 or 0x17	Value of the written parameter too low
0x98 or 0x18	Maximum value is less than minimum value
0x99 or 0x19	General error
0x9A or 0x1A	Data can not be transmitted or saved in the application.
0x9B or 0x1B	Data can not be transmitted or saved in the application due to the status of the control.
0x9C or 0x1C	Data can not be transmitted or saved in the application due to the reset device.
0x9D or 0x1D	Dynamic generation of the object dictionary not possible or no object dictionary existing
0x9E or 0x1E	Read access denied
0x9F or 0x1F	Write access denied

7 Appendix

7.A Display Object Data

You find information on the object of a drive via the object browser in the software *drivemaster2* (tab page "Diagnosis").

The following details are displayed for the objects loaded in the object browser: object name, index, current value and unit:

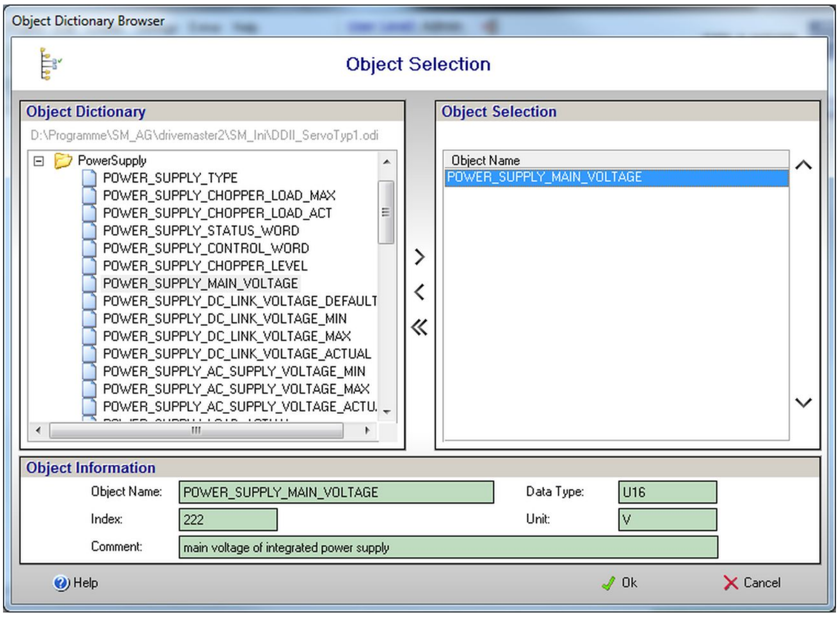


Object name:	Index:	Value:	Unit:
POWER_SUPPLY_MAIN_VOLTAGE	222	230.0	V
ICTRL_IQ_MAX	98	15.00	A
DEVICE_HARDWARE_IDENT_CODE	2	17502466	
DEVICE_PARAMETER_INFO_STRING	3	Parameter for M406H	

Projekt: Online Projekt | Host: localhost | Server: S&M USB Server V2.0 (GUI) | Pow: 0 - Drv: 0 A - Online

Fig. 2: Object browser in drivemaster2

The object selection dialog displays additional data in the bottom field "Object Information" for the object currently selected on the right side. These are the data type and a comment on the object



Object Dictionary Browser

Object Selection

Object Dictionary

D:\Programme\SM_AG\drivemaster2\SM_Ini\DDII_ServoTyp1.odi

- PowerSupply
  - POWER_SUPPLY_TYPE
  - POWER_SUPPLY_CHOPPER_LOAD_MAX
  - POWER_SUPPLY_CHOPPER_LOAD_ACT
  - POWER_SUPPLY_STATUS_WORD
  - POWER_SUPPLY_CONTROL_WORD
  - POWER_SUPPLY_CHOPPER_LEVEL
  - POWER_SUPPLY_MAIN_VOLTAGE
  - POWER_SUPPLY_DC_LINK_VOLTAGE_DEFAULT
  - POWER_SUPPLY_DC_LINK_VOLTAGE_MIN
  - POWER_SUPPLY_DC_LINK_VOLTAGE_MAX
  - POWER_SUPPLY_DC_LINK_VOLTAGE_ACTUAL
  - POWER_SUPPLY_AC_SUPPLY_VOLTAGE_MIN
  - POWER_SUPPLY_AC_SUPPLY_VOLTAGE_MAX
  - POWER_SUPPLY_AC_SUPPLY_VOLTAGE_ACTU.

Object Selection

Object Name
POWER_SUPPLY_MAIN_VOLTAGE

Object Information

Object Name: POWER_SUPPLY_MAIN_VOLTAGE

Data Type: U16

Index: 222

Unit: V

Comment: main voltage of integrated power supply

Fig. 3: Object selection

7.B Examples

7.B.1 Speed Setting and Switch-on

The following example shows a simple control of the drive.

Requirements

The following settings/states in *drivemaster2* are required for this example:

- The control channel is "Serial interface/RS485/USB" or "DNC 8 Byte Telegram" (see "Parameter → Drive control → Control channel").
- The setpoint channel is "Serial interface/RS485/USB" or "DNC 8 Byte Telegram" (see "Parameter → Drive control → Setpoint channel").
- The drive signals the state "Switch On Disabled, Software Funktion 'Quick Stop'" (see "Diagnosis → Drive actual values").

7.B.1.1 Send Shutdown Command to Drive

After booting the drive, switch-on is disabled. The shutdown command enables switch-on. The object DEV_CTRL_CONTROL_WORD is written.

In order to trigger the shutdown command the **value 6** is written into the object. The available control commands are described in the document "Drive System SD2 - Device Control".

The object has the following properties:

DEV_CTRL_CONTROL_WORD	
Index	68
Type	u16 (unsigned 16 bit)
Value	6

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	count	data	data	check
0x00	0x0b	0x02	0x01	0x0e	0x44	0x00	0x00	0x00	0x00	0x00	0x02	0x06	0x00	0x97

The value "count" is 2 because 2 bytes must be transmitted, since the object is a 16 bit data type. Via "data" the 2 bytes are transmitted then.

The drive returns the following sequence:

Reply:

zero	length	dest	source	cmd	error code	check
0x00	0x04	0x01	0x02	0x8e	0x00	0x6a

Now the drive signals the state "Ready To Switch On" in the diagnosis of the software *drivemaster2*.

7.B.1.2 Set Reference Speed Value

The reference speed value is to be 1000 rpm. The object SPG_TARGET_VELOCITY_VL_UUNIT is written.

The object has the following properties:

SPG_TARGET_VELOCITY_VL_UUNIT	
Index	395
Type	i32 (integer 32 bit)
Value	1000000 (indicated in 0.001 rpm)

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	count	data	data	data	data	check
0x00	0x0b	0x02	0x01	0x0e	0x8b	0x01	0x00	0x00	0x00	0x00	0x04	0x40	0x42	0x0f	0x00	0xc2

The drive returns the following sequence:

Reply:

zero	length	dest	source	cmd	error code	check
0x00	0x04	0x01	0x02	0x8e	0x00	0x6a

7.B.1.3 Activate Controller

The **value 7** is written into the object DEV_CTRL_CONTROL_WORD.

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	count	data	data	check
0x00	0x0b	0x02	0x01	0x0e	0x44	0x00	0x00	0x00	0x00	0x00	0x02	0x07	0x00	0x96

The drive returns the following sequence:

Reply:

zero	length	dest	source	cmd	error code	check
0x00	0x04	0x01	0x02	0x8e	0x00	0x6a

Now the drive signals the state "Switched On" in the diagnosis of the software *drivemaster2*.

7.B.1.4 Operation enable

The **value 15** is written into the object DEV_CTRL_CONTROL_WORD.

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	count	data	data	check
0x00	0x0b	0x02	0x01	0x0e	0x44	0x00	0x00	0x00	0x00	0x00	0x02	0x0f	0x00	0x8e

The drive returns the following sequence:

Reply:

zero	Length	dest	source	cmd	error code	check
0x00	0x04	0x01	0x02	0x8e	0x00	0x6a

Now the drive signals the state "Operation Enabled" in the diagnosis of the software *drivemaster2* and the motor rotates.

7.B.1.5 Disable Operation

The **value 7** is written into the control word to stop the motor (see section 7.B.1.3 "Activate Controller", p. 26).

7.B.1.6 Deactivate Controller

The **value 6** is written into the control word to switch the controller off (see section 7.B.1.1 "Send Shutdown Command to Drive", p. 25).

7.B.2 Read Actual Speed Value

The object VCTRL_VELOCITY_ACTUAL_VALUE_UUNIT is read.

The object has the following properties:

VCTRL_VELOCITY_ACTUAL_VALUE_UUNIT	
Index	398
Type	i32 (integer 32 bit)

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	check

0x00	0x09	0x02	0x01	0x0d	0x8e	0x01	0x00	0x00	0x00	0x00	0x57

The drive returns the following sequence:

Reply:

zero	length	dest	source	cmd	count	error code	data 0	data 1	data 2	data 3	check
0x00	0x09	0x01	0x02	0x8d	0x04	0x00	0x7f	0x50	0x0f	0x00	0x84

The actual speed is 0x000f507f = 1003647 ≈ 1004 rpm.

7.B.3 Read Status Word

The object DEV_CTRL_STATUS_WORD is read.

The object has the following properties:

DEV_CTRL_STATUS_WORD	
Index	67
Type	u16 (unsigned 16 bit)

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	check
0x00	0x09	0x02	0x01	0x0d	0x43	0x00	0x00	0x00	0x00	0x00	0xa3

The drive returns the following sequence:

Reply:

zero	length	dest	source	cmd	count	error code	data 0	data 1	check
0x00	0x09	0x01	0x02	0x8d	0x02	0x00	0x37	0x66	0x84

The current status word is 0x6637.

7.B.4 Read Parameter Set Identification

The object DEVICE_PARAMETER_IDENT_STRING is read.

The object has the following properties:

DEVICE_PARAMETER_IDENT_STRING	
Index	22
Type	string

Send:

zero	length	dest	source	cmd	index lo	index hi	sub ID 0	sub ID 1	sub ID 2	sub ID 3	check
0x00	0x09	0x02	0x01	0x0d	0x16	0x00	0x00	0x00	0x00	0x00	0xd0

The drive returns the following sequence:

Reply:

zero	length	dest	source	cmd	count	error code	data 0	data 1	data 2	data 3	data 4	•••
0x00	0x26	0x01	0x02	0x8d	0x21	0x00	0x20	0x54	0x65	0x73	0x74	
•••	data 5	data 6	data 7	data 8	data 9	data 10	data 11	data 12	data 13	data 14	data 15	•••
	0x20	0x4d	0x6f	0x74	0x6f	0x72	0x00	0x00	0x00	0x00	0x00	
•••	data 16	data 17	data 18	data 19	data 20	data 21	data 22	data 23	data 24	data 25	data 26	•••
	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
					•••	data 27	data 28	data 29	data 30	data 31	data 32	check
						0x00	0x00	0x00	0x00	0x00	0x00	0x37

The "data" blocks contain the string using the following format:

data 0	Length of the string 0x20 means that the string has a length of 32 characters. If less than the maximum number of 32 characters was entered during parameterization, it is a null-terminated string.
data 1 – 32	Output of each character in ASCII format

Thus, the read string for the parameter set identification is "Test Motor".