

BAUMÜLLER

**OPTION CARD
CAN-INTERFACE**

Technical description and
operation manual

Edition October 1996

E

5.95076.03

BAUMÜLLER

**OPTION CARD
CAN-INTERFACE**

Technical description and operation manual

Edition October 1996

5.95076.03

This operation manual is intended as a complement to the technical description and the operation manual of the apparatus.

**READ AND COMPLY WITH THE OPERATION MANUAL AND
THE SAFETY NOTES BEFORE COMMISSIONING**

This manual contains the necessary information for normal operation of the products described therein. The drives may only be used, maintained and repaired by personnel familiar with the operation manual and the applicable regulations on working safety and accident prevention. The devices are manufactured to a high technical specification and are operationally safe. Provided that all safety instructions have been adhered to, there will be no personal danger during the installation and commissioning stages.

The commissioning is prohibited until it has been positively determined that the machine, into which these components are to be incorporated, complies with EC machine regulations.

This technical description replaces and nullifies all previous description. In order to provide the best possible service, we reserve the right to alter information without notice.

Manufacturer and Baumüller Nürnberg GmbH

Supplier's address: Ostendstr. 80
90482 Nürnberg
Germany

Tel. (++49-911) 5432 - 0

Fax (++49-911) 5432 - 130

Copyright: The technical description and the operation manual may not be copied or duplicated without our permission.

Country of origin: Made in Germany

Date of manufacture: Determined from the serial number on the machine/motor.

TABLE OF CONTENTS

1 Technical Data	1
1.1 General.....	1
1.2 Technical Data of the Option Board	3
2 Installation	5
2.1 Pin Assignments	5
2.2 Accessories	8
3 Communications	9
3.1 Configuration	9
3.1.1 Data Transfer Rrate	9
3.1.2 BUS Address	9
3.2 Process Data Transfer	10
3.2.1 Control Word	11
3.2.2 Status Word.....	11
3.2.3 Specified Value	12
3.2.4 Actual Value	12
3.2.5 Broadcast Messages.....	13
3.3 Demand Data Transfer.....	15
3.3.1 PO Messages	15
3.3.2 PI Message.....	16
3.3.3 Parameters.....	18
3.3.4 Objects	20
3.4 Object List	22
3.5 Communications Monitoring.....	25
3.6 Enabling Communications	26
3.7 Information on Cyclical Transfer	27

1 TECHNICAL DATA

1.1 General

In addition to closed-loop control, one of the most important features of digital drives is the capability to carry out a wide range of functions, some of which can be quite complex, and to communicate with other devices, like controllers and drives.

Apart from software, these functions need appropriate adaptation of the hardware: optical inputs and outputs, analog acquisition, encoder analysis, computer systems for communications and higher-level functions.

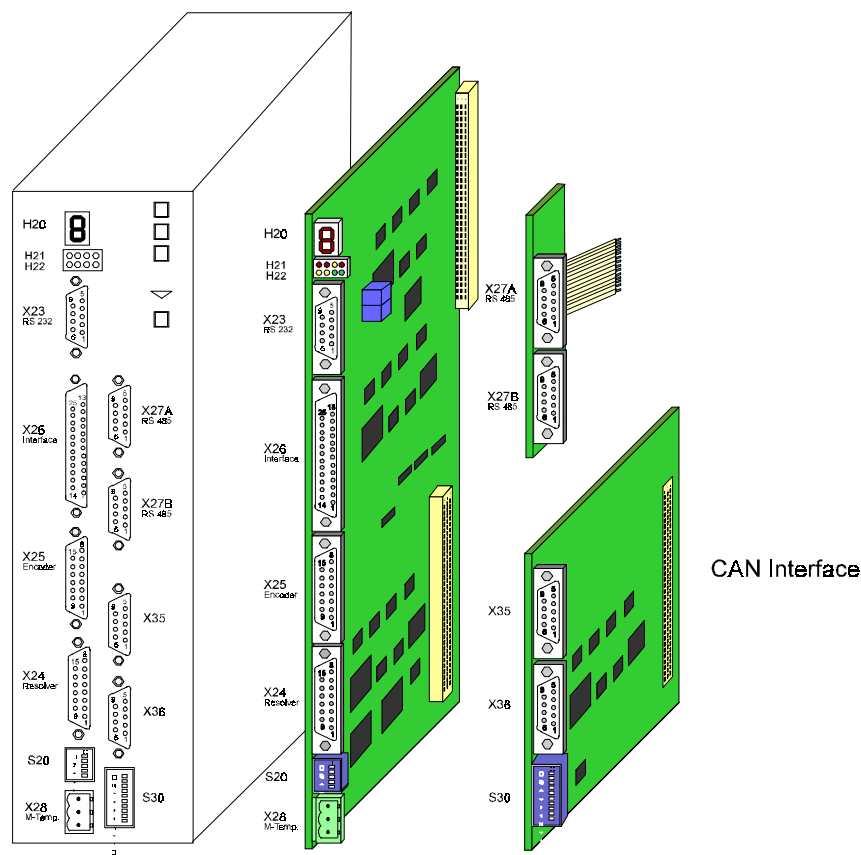
This means that digital drives are supposed, for example, to replace an analog device (± 10 V analog speed specified value) as well as to be designed to use future communication protocols.

This can be achieved due to a modular drive concept:

You adapt the drive controller to various requirements by fitting plug-in expansion boards.

An option board is available as an expansion card for connecting to a CAN bus.

This expansion card links the close-loop controller to computers or programmable logic controllers (e.g. **Omegas**).



Sample Application: Expanding a Controller using a CAN Interface Option Board

By means of the CAN-BUS adapter, you can use process and demand data to exploit the functionality of the closed-loop controller to the full. From another CAN node (e.g. CAN-01/02/03 for **Omega**, Omega Drive Line), you can use the specified protocol to set the cyclical specified value (for process data transfer) and to set the parameters in the controller (for demand data transfer).

The CAN-BUS adapter has two connectors – that are connected to one another internally – with galvanic separation to the BUS 6 E-controller.

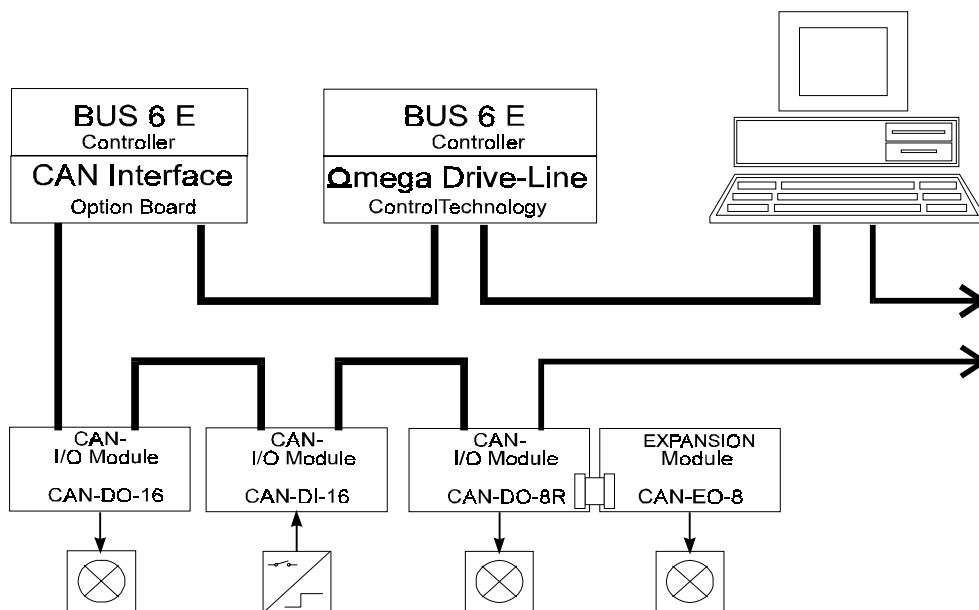
You can set a data transfer rate of up to 1 Mbps.

Characteristics

- Serial asynchronous bus system
- Real-time capability (maximum of Mbps on a 40-metre long bus)
- Broadcast/multicast and point-to-point communication
- Powerful error detection and treatment capability
- High level of reliability (Hamming distance of 6)
- Multimaster
- Priority-controlled bus allocation
- Guaranteed maximum latency for high-priority messages
- Conforms to international standards

Bus-Structure

In the contiguous CAN-BUS, you set the same data transfer rate for all the nodes. There must be 124-Ω terminal resistors at each end of the CAN-BUS.



1.2 Technical Data of the Option Board

When fitted with the CAN Interface option board, the drive is considered to be a communication-enabled node on the CAN bus.

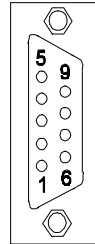
CPU	80C32, 16 MHz
Memory	38 kB of RAM, 64-kB EPROM
CAN Controller	PCA82C200
Physical Layer	ISO 11898
Baud Rate	Max. of 1 Mbps
Operating Voltage	+5 V (from controller supply)
Potential Separation	Optocoupler, DC/DC converter
Plug-in Connection to CAN Bus	9-pin SUB-D male and female connector

2 INSTALLATION

2.1 Pin Assignments

- X 35 (CAN board in conjunction with BUS 6 E-Controller)

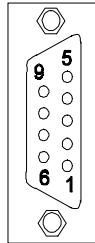
SUB-D plug connector



Pin No.	Assignment
1	Not assigned
2	CAN LOW
3	GND
4	Not assigned
5	Not assigned
6	Not assigned
7	CAN HIGH
8	Not assigned
9	Not assigned

- X 36 (CAN board in conjunction with BUS 6 E-Controller)

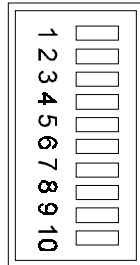
SUB-D socket



Pin No.	Assignment
1	Not assigned
2	CAN LOW
3	GND
4	Not assigned
5	Not assigned
6	Not assigned
7	CAN HIGH
8	Not assigned
9	Not assigned

- DIP switches

S20



Number	Assignment
1	Baud rate setting: To left = 1 Mbps, to right (ON) = 250 kbps
2	Reserved: Must be to right (ON)
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Bit 0 of the BUS address: To left = 0, to right (ON) = 1
8	Bit 1 of the BUS address: To left = 0, to right (ON) = 1
9	Bit 2 of the BUS address: To left = 0, to right (ON) = 1
10	Bit 3 of the BUS address: To left = 0, to right (ON) = 1

2.2 Accessories

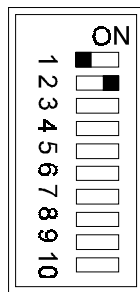
3 COMMUNICATIONS

3.1 Configuration

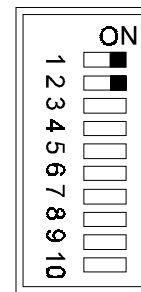
3.1.1 Data Transfer Rate

Using DIP switch S20, you can set a data transfer rate of 250 kbps or 1 Mbps. If the controller cassette is standing upright, DIP switch 1 set to the left means 1 Mbps, switch set to the right means a transfer rate of 250 kbps. The settings must be made when the controller is switched off: altering the settings during operation has no effect.

The lower data transfer rate is needed if there are other nodes on the CAN-BUS that are not capable of being run at 1 Mbps, or if the length of the CAN-BUS does not permit a rate of Mbps.



1 Mbps



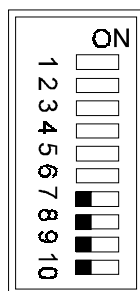
250 kbps

3.1.2 BUS Address

You use DIP switches 7 to 10 to carry out binary setting of the CAN-BUS adapter's BUS address. This means that you can choose a number between 0 and 15.

The settings must be made when the controller is switched off: altering the DIP switch settings during operation has no effect.

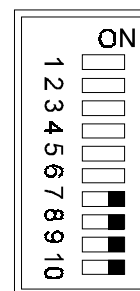
The setting of the number does not depend on the sequence of CAN nodes in the BUS cabling. On a contiguous CAN-BUS, you may only assign any one number to one CAN-BUS adapter. This means that it is impossible to assign numbers twice or more times. You do not, however, need to assign the BUS addresses in contiguous, ascending numerical order.



Address 0



Address 1



Address 15

3.2 Process Data Transfer

Process data transfer is cyclical data transfer of the control word, the status word, the specified and the actual values. The minimum transfer intervals are as follows: 1 ms for Specifiedvalue0 (Actualvalue0), 2 ms for Specifiedvalue1 (Actualvalue1), 4 ms for Specifiedvalue2 (Actualvalue2) and 8 ms for Specifiedvalue3 (Actualvalue3). The system accepts the control word after a maximum of 1 ms.

The control word complies with the DRIVECOM standard. It has parameter number, ID-No. 120 in the controller. The closed-loop controller supplies the status word; it has parameter number, ID-No. 121.

All the parameters of the closed-loop controller are available as parameter numbers for the four specified values and the four actual values. The list of parameter numbers or the specified and actual values must be set using the PI or the PO message. This list can be stored in the controller by means of data set management (see also the BUS 6 E-controller documentation).

You can set cyclical transmission of the actual values via object 5FF2_h (refer to section 3.4, Object List). With software version 3.01 and above, it is possible to store the set cycle times via data set programming in the BUS 6 E-controller.

In the list of message types belonging to process data transfer that follows, the abbreviations used have the following meanings:

_h	=	Value as a hexadecimal number	
_b	=	Value as a binary number	
IDB 0	=	Identifier of byte 0	(refer to the structure of a message)
IDB 1	=	Identifier of bits 2 - 0	(")
RB	=	Remote bit	(")
DLC	=	Number of data bytes	(")
DB 0 - DB 7	=	Data bytes 0 to 7	(")

The bus address means the address (0-15) of the CAN adapter that you set using DIP switches 7 - 10 (refer to section 3.1.2, Bus Address).

3.2.1 Control Word

Control Word Message (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1
00_h + bus address	001_b	0	2	Control word parameter number 120	

Using the control word, which has a fixed length of two bytes, you can set a controller enable or trigger an emergency stop, amongst other things. For a complete description of control word functions, refer to the controller's Technical Description and Operating Instructions.

If the control word message results in a change of status in the controller, the controller transmits the new status in status word message as the response.

3.2.2 Status Word

Status Word Message (from the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1
10_h + bus address	001_b	0	2	Control word parameter number 121	

The status word, which has a fixed length of two bytes, shows controller enable and that the system is ready for use amongst other things. For a complete description of status word functions, refer to the BUS 6 E-Controller Description.

Status Word Request (to the controller)

IDB 0	IDB 1	RB	DLC
10_h + bus address	001_b	1	0

Using this message, it is possible to request the current state of the controller's status word. The controller responds with the status word message (see above).

3.2.3 Specified Value

Specified Value Message (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3
20_h + bus address	Specified value number	0	Length of specified value in bytes	Specified value data Minimum of 1 byte, maximum of 4 bytes			

The specified value number is in the range 0 - 3.

The length of the specified value in the message must match the format of this parameter in the controller (Byte length = 1, Word length = 2, Double word length = 4).

Actual Value Message (from the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3
30_h + bus address	Actual value number	0	Length of actual value in bytes	Actual value data Minimum of 1 byte, maximum of 4 bytes			

The actual value number is in the range 0 - 3.

The length of the actual value in the message matches the format of this parameter in the controller (Byte length = 1, Word length = 2, Double word length = 4).

3.2.4 Actual Value

Actual Value Request (to the controller)

IDB 0	IDB 1	RB	DLC
30_h + bus address	Actual value number	1	0

With the actual value request, the system requests the current actual value that has the number stated in IDB 1.

The actual value number is in the range 0 - 3.

The controller responds with the actual value message (see above).

3.2.5 Broadcast Messages

Apart from messages that must have an appropriate bus address, all CAN-BUS adapters receive broadcast messages.

Broadcast Control Word Message (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 0	DB 1
00_h	000_b	0	3	01_h	Control word Parameter number 120	

Using the control word, which has a fixed length of two bytes, you can set a controller enable or trigger an emergency stop, amongst other things. For a complete description of control word functions, refer to the controller's Technical Description and Operating Instructions.

If the control word message results in a change of status in the controller, the controller transmits the new status in status word message as the response.

Broadcast Status Word Request (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0
00_h	000_b	0	1	02_h

Using this message, it is possible to request the current state of the controller's status word. The controller responds with the status word message.

Broadcast Specified Value Message (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 1	DB 2	DB 3	DB 4
00_h	000_b	0	Length of specified value in bytes + 2	03_h	Specified value number	Specified value data Minimum of 1 byte, maximum of 4 bytes			

The specified value number is in the range 0 - 3.

The length of the specified value in the message must match the format of this parameter in the controller.

Specified value format	DLC
Byte	3
Word	4
Double word	6

Broadcast Actual Value Request (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4
00_h	000_b	0	1 + number of actual values	04_h	Actual value number 1	Actual value number 2	Actual value number 3	Actual value number 4

With the actual value request, between 1 and a maximum of 4 actual values can be requested. The numbers of the actual values are in data bytes 1 to 4. At a transfer rate of 1 Mbps, a maximum of two actual values may be requested in one message.

The actual value number is in the range 0 - 3. The controller responds with the actual value message (see above).

3.3 Demand Data Transfer

Demand data transfer is for directly accessing one of the controller's parameters or an object that is in the CAN-BUS adapter. On accessing, the system states the parameter (object) number, i.e. you do not have to set it in advance. The controller responds to read and write accesses with a message.

3.3.1 PO Messages

PO messages are for setting or reading out the parameter numbers of the specified values that are to be transferred via the specified value message (refer to 3.2.2). You must set the number of specified values that are to be cyclically transferred as well as the parameter numbers that each of these specified values have in the controller. A list of four words (8 bytes) is used for this:

Word Number	PO Parameter Number List
0	Parameter number of specified value 0
1	Parameter number of specified value 1
2	Parameter number of specified value 2
3	Parameter number of specified value 3

When assigning the specified value numbers, you should note that specified value 0 may have a maximum length of two bytes. If less than four specified values are to be transferred, you must enter the value 0000_h as the parameter number for the specified values that are not needed.

Write PO Data (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
80_h + Bus address	100_b	0	8	Spec. val. no. 0		Spec. val. no. 1		Spec. val no. 2		Spec. val no. 3	
PO Parameter Number List											

Command Write PO Data sets a new PO parameter number list in the controller. After this, the system must enable specified value transfer by writing the value FFFF_h to object 6002_h (see below). You can permanently store the settings, so that they are retained after switch-off, using data management by means of parameter ID-No. 190 in the BUS 6 E-Controller. The controller responds with the PO Write Response (see below).

PO Write Response (from the controller)

If the PO write access was successful, the system returns the following response.

IDB 0	IDB 1	RB	DLC	DB 0
80_h + bus address	101_b	0	1	00_h

In the event of an error, the system returns the following response:

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2
80_h + bus address	101_b	0	3	80_h	00_h	41_h

With software version 3.01 and above, the system returns the following response:

IDB 0	IDB 1	RB	DLC	DB 0
80_h + bus address	101_b	0	1	80_h

Read PO Data (to the controller)

IDB 0	IDB 1	RB	DLC
80_h + bus address	100_b	1	0

This command requests the current PO parameter number list in the controller. The controller responds with the PO Read Response.

PO Read Response (from the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
80_h + bus address	101_b	0	8	Spec. val. no. 0		Spec. val. no. 1		Spec. val. no. 2		Spec. val. no. 3	
PO Parameter Number List											

The PO Read Response contains the current PO Parameter Number List. The system transmits it as the response to the Read PO Data.

3.3.2 PI Message

PI messages are for setting or reading out the parameter numbers of the actual values that are to be transferred via the actual value message. You must set the number of actual values that are to be cyclically transferred as well as the parameter numbers that each of these actual values have in the controller. A list of four words (8 bytes) is used for this:

Word Number	Parameter Number List
0	Parameter number of actual value 0
1	Parameter number of actual value 1
2	Parameter number of actual value 2
3	Parameter number of actual value 3

When assigning the actual value numbers, you should note that actual value 0 may have a maximum length of two bytes. If less than four actual values are to be transferred, you must enter the value 0000_h as the parameter number for the actual values that are not needed.

Write PI Data (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
80_h + Bus address	110_b	0	8	Act. val. 0		Act. val. 1		Act. val. 2		Act. val. 3	
PI Parameter Number List											

Command Write PI Data sets a new PI parameter number list in the controller. You can permanently store the settings, so that they are retained after switch-off, using data management by means of parameter ID-No. 190 in the BUS 6 E-Controller.

The controller responds with the PI Write Response (see below).

PI Write Response (from the controller)

If the PI write access was successful, the system returns the following response:

IDB 0	IDB 1	RB	DLC	DB 0
80_h + bus address	111_b	0	1	00

In the event of an error, the system returns the following response:

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2
80_h + bus address	111_b	0	3	80_h	00_h	41_h

With software version 3.01 and above, the system returns the following response:

IDB 0	IDB 1	RB	DLC	DB 0
80_h + bus address	111_b	0	1	80_h

Read PI Data (to the controller)

IDB 0	IDB 1	RB	DLC
80_h + bus ad- dress	110_b	1	0

This command requests the current PI parameter number list in the controller.

The controller responds with the PI Read Response.

PI Read Response (from the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
80_h + bus address	111_b	0	8	Act. val. no. 0		Act. val. no. 1		Act. val. no. 2		Act. val. no. 3	
PI Parameter Number List											

The PI Read Response contains the current PI Parameter Number List. The system transmits it as the response to the Read PI Data.

3.3.3 Parameters

Write Parameter (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
80_h + bus address	010_b	0	5,6,8	Com- mand	Element	Parameter number		Data 1 byte min. 4 bytes max.			

You use this command to write a parameter in the controller.

"Command" contains the write command combined with the parameter format.

Write a	Command	DLC	Number of Data Bytes
Byte parameter	08_h	5	5
Word parameter	09_h	6	6
Double word parameter	0A_h	8	8

"Element" contains the element of the parameter that is to be written. To write the datum (value) of the parameter, 07_h is in "Element". You enter the controller parameter number in "Parameter Number".

The controller responds with the Parameter Write Response

Parameter Write Response (from the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2
80_h + bus address	011_b	0	1 3 in the event of an error	Result	Error number (in the event of an error)	

The controller responds to a parameter write command with this message.

If the parameter write access was successful, Result contains 08_h. In the event of an error, Result contains the value 28_h and DB 1 and DB 2 show the error number from the controller (refer to the BUS 6 E-Controller documentation).

Read Parameter (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB3
80_h + bus address	010_b	0	4	Format	Element	Parameter number	

You use this command to read a parameter in the controller. In this command, it is possible to enter the parameter's format in "Format".

Read a	Format
Byte parameter	00_h
Word parameter	01_h
Double word parameter	02_h

"Element" contains the element of the parameter that is to be read. To read the datum (value) of the parameter, 07_h is in "Element". You enter the controller parameter number in "Parameter Number".

The controller responds with the Parameter Read Response

Parameter Read Response (from the controller)

A successful response to Parameter Read looks like this:

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4
80_h + bus address	011_b	0	2,3,5	Format	Data 1 byte min. 4 bytes max.			

The system returns the format of the read parameter in "Format".

Read a	Format	DLC
Byte parameter	00_h	2
Word parameter	01_h	3
Double word parameter	02_h	5

The number of data bytes sent corresponds to the format (see also DLC).

If the system cannot carry out the read access, the Parameter Read Response looks like this:

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2
80_h + bus address	011_b	0	3	20_h	Error number	

The controller returns an error number.

3.3.4 Objects

Write Object (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
80_h + bus address	000_b	0	5,6,8	Com- mand	Subindex	Object number			Data 1 byte min. 4 bytes max.		

You use this command to write an object of the CAN-BUS adapter.

"Command" contains the write command combined with the object format.

Write a	Command	DLC	Number of Data Bytes
Byte parameter	08_h	5	5
Word parameter	09_h	6	6
Double word parameter	0A_h	8	8

"Subindex" contains the subindex of the object to be written. To write the entire object, 00_h is in the subindex. You enter the CAN-BUS object number in "Object Number" (see Object List below).

The controller responds with the Object Write Response.

Object Write Response (from the controller)

IDB 0	IDB 1	RB	DLC	DB 0
80_h + bus address	001_b	0	1	Result

The controller responds to an object write command with this message (see above).

If the object write access was successful, Result contains 00_h. In the event of an error, Result contains the value 50_h.

Read Object (to the controller)

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB3
80_h + bus address	000_b	0	4	Format	Subindex	Object number	

You use this command to read an object in the controller. In this command, you must enter the format of the object in "Format".

Read a	Format
Byte parameter	00_h
Word parameter	01_h
Double word parameter	02_h

"Subindex" contains the subindex of the object to be read. To read the entire object, 00_h is in the subindex. You enter the CAN-BUS object number in "Object Number" (see Object List below).

The controller responds with the Object Read Response.

Object Read Response (from the controller)

A successful response to Object Read looks like this:

IDB 0	IDB 1	RB	DLC	DB 0	DB 1	DB 2	DB 3	DB 4
80_h + bus address	001_b	0	2,3,5	00_h	Data 1 byte min. 4 bytes max.			

The number of data bytes sent corresponds to the format of the object (see also DLC).

If the system cannot carry out the read access, the Object Read Response looks like this:

IDB 0	IDB 1	RB	DLC	DB 0
80_h + bus address	001_b	0	1	50_h

3.4 Object List

The table below lists permissible object numbers.

NOTE

Accessing other object numbers is not allowed.

Object Number	Data Type	Object Type	Meaning
5FF2_h	Word	Array	Actual value cycle time
5FF4_h	Word	Variable	CAN communications monitoring
6002_h	16 bit	Variable	Specified value enable

Object 5FF4_h only exists in Version 3.01 and above.

In general, the data set management cannot store the contents of the objects in the controller. This means that the settings must be written again each time that you switch on the controller.

Every time you switch on the controller, the system automatically enables all specified values defined by the PO data message. This is the initial value for object 6002_h.

With Version 3.01 and above, it is possible to store the actual value cycle times (object 5FF2) in the controller. The values are reloaded when you switch on the controller.

Actual Value Cycle Time (5FF2_h)

In the actual value cycle time object, you can set for each of the four actual values (see Process Data Transfer) the cycle time with which it is to be automatically transferred by the controller. It is not possible to access this entire object via subindex = 0. Access must be by means of subindex 1 to 4. IN this connection you must make the following settings: for actual value 0, subindex = 1, for actual value 1, subindex = 2, etc.

The time base for specifying the cycle time depends on the CAN transfer rate:

- for speeds of up to 250 kbps, the time base is 4 ms
- for speeds of 500 kbps and above, the time base is 0.5 ms.

With Version 3.01 and above, the time base is 0.5 ms in all cases.

If you set the cycle time to 0, the accompanying actual value is not transferred cyclically. In this case, the value can only be read via the actual value request.

Object attribute	Value
Object number	5FF2_h
Object type	Array
Data type	Word (16 Bit)
Number of elements	4

Subindex	Actual value cycle time of
1	Actual value 0
2	Actual value 1
3	Actual value 2
4	Actual value 3

The default setting is that no actual value is transmitted cyclically.

CAN Communication Monitoring Timeout (5FF4_h) (Version 3.01 and Above)

You can use this object to set a monitoring timeout after which the system suspends communications with the controller if no more messages have been receive at the CAN (see also communications monitoring).

Object attribute	Value
Object number	5FF4_h
Object type	Variable
Data type	Word (16-bit)
Number of elements	1
Meaning	CAN communications monitoring timeout in ms

Specified Value Enable (6002_h)

Using this object, you can carry out a specified value enable for the maximum of four process data specified values. This is necessary, above all, after the PO data has changed (refer to Process Data Transfer).

The object is a 16-bit variable in which each bit allows enabling of one byte of the process data transfer. The first two bits (bit 0 and bit 1) are for enabling of the control word. The next two (bit 2 and bit 3) make possible enabling of the first specified value (the word size). The higher-numbered bits enable the bytes of the other specified values (if they are defined in the PO data description). You can enable all the bytes by writing FFFF_h to the specified value object.

Object attribute	Value
Object number	6002_h
Object type	Variable
Data type	Word (16-bit)
Number of elements	1

Bit	Enabling of
0	Control word byte 0
1	Control word byte 1
2	Specified value 0 byte 0
3	Specified value 0 byte 1
4	Specified value 1 byte 0
5	Specified value 1 byte 1
6	Specified value 1 byte 2
7	Specified value 1 byte 3
8	Specified value 2 byte 0
9	Specified value 2 byte 1
10	Specified value 2 byte 2
11	Specified value 2 byte 3
12	Specified value 3 byte 0
13	Specified value 3 byte 1
14	Specified value 3 byte 2
15	Specified value 3 byte 3

The default setting is that all the specified values entered in the PO data are enabled.

3.5 Communications Monitoring

In the controller, it is possible to monitor communications with the CAN option board. You do this by means of parameter ID-Nos. 127 to 129 (refer to the E-Controller description). The monitoring facility triggers if the option board has not communicated with the controller for the time set using parameter ID-No. 128. This monitors whether the option board is operational or not. Equally, this is intended to monitor whether the CAN bus is still active.

Up to Version 2, a communications cycle with the controller was only carried out if a message was received at the CAN. This meant that the controller could also monitor the CAN bus. However, it was not possible to send actual values from the controller than messages were sent to the controller, since new actual values were only returned to the CAN board on each communication cycle.

From Version 3.01 onwards, a timeout runs in the CAN board too (object 5FF4_h).

After each received CAN message, the system continues communication with the controller for this period of time and, with this, transfers new actual values from the controller. If no message is received at the CAN for longer than the CAN communications timeout, the system ends communication to the controller. The timeout then begins running in the controller.

When you switch on the controller, object 5FF4 is preset with the value that is set in parameter ID-No. 128. Changing the parameter via Write CAN Parameter also changes the object. In addition, the object can be changed by Write Object. However, this value is lost when you switch off the controller.

You can achieve particularly strict communications monitoring by setting the timeout in controller parameter ID-No. 128 to 1 ms and object 5FF4 to the maximum time that passes between two messages being transferred to the controller.

3.6 Enabling Communications

With software version 3.01 and above, after switching on the controller you must enable communications across the CAN bus. Before enabling has been carried out, the CAN option board ignores all the other messages on the CAN bus and does not send any itself (even if actual values have to be sent cyclically). Enabling can be carried out in two different ways:

1. From the master, the system transmits the 1-byte-long broadcast message with the identifier 000_h that has the byte value FF_h.
(With software version 3.01 and above, an **Omega** master transmits this message automatically after system starting.)
2. From the master, the system transmits the message with the identifier OAA_h, that is two-bytes-long – AA_h; 04_h. Communications is enabled 200 ms after receipt of this message. (An **Omega** master cyclically transmits this message during CAN operation. This means that communications can be enabled for a controller that is switched on with a delay.)

3.7 Information on Cyclical Transfer

Process data transfer (see 3.2) is intended to be cyclical. On the CAN option board, you can set cycle times for the actual value (refer to 3.4, Object 5FF2_h).

The two tables below are intended to provide you with ideas for setting the cycle times. The tables list the shortest cycle times in dependence on the baud rate and the number of actual values that are to be transmitted cyclically. You should note that CAN bus loading together with the other CAN bus nodes does not reach 100%, since, in this case, lower-priority messages like demand data would be seriously delayed or it would not be possible to send them at all.

Minimum possible cycle times if only actual values are transmitted:

Baud rate	1 actual value				2 actual values				3 actual values				4 actual values	
	2 bytes		4 bytes		2 bytes		4 bytes		2 bytes		4 bytes		2 bytes	
In	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL
1000	0.5	14%	0.5	17%	0.5	29%	1.0	18%	1.0	23%	1.0	26%	1.0	36%
500	0.5	25%	0.5	33%	1.0	26%	1.0	35%	1.0	45%	1.0	51%	1.5	40%
250	0.5	54%	0.5	66%	1.0	53%	1.0	67%	1.5	52%	1.5	87%	2.0	52%
125	1.0	53%	1.0	65%	1.5	70%	2.0	68%	2.0	80%	2.5	82%	3.0	70%

Abbreviations:

ZZ: cycle time in ms

BL: approximate CAN bus loading in percent

Minimum possible cycle times if actual values and specified values are transmitted with the same cycle time

Baud rate	1 specified value/1 actual value				2 specified values/2 actual values				3 specified values/3 actual values				4 specified values/4 actual values	
	2 bytes		4 bytes		2 bytes		4 bytes		2 bytes		4 bytes		2 bytes	
In	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL	ZZ	BL
1000	1.0	13%	1.0	17%	1.0	28%	1.0	38%	1.0	40%	1.0	58%	1.0	53%
500	1.0	27%	1.0	34%	1.0	54%	1.0	69%	1.0	84%	2.0	52%	2.0	58%
250	1.0	53%	1.0	66%	2.0	53%	2.0	69%	2.0	81%	3.0	68%	3.0	72%
125	2.0	53%	2.0	65%	3.0	70%	3.0	86%	4.0	80%	5.0	80%	5.0	84%

Abbreviations:

ZZ: cycle time in ms

BL: approximate CAN bus loading in percent

If you use DRIVECOM actual value objects 6044_h (absolute speed in RPM) or 6054_h (speed in percent 100% = 4000_h), the minimum possible cycle time increases by 0.5 ms. Using the corresponding specified value objects 6042_h (absolute speed) or 6052_h (speed in percent) increases the minimum possible cycle time by 1 ms. For this reason, it is advisable not to use these objects; rather, you should directly use the speed specified and actual values in standardization of the controller (100% = 10000).