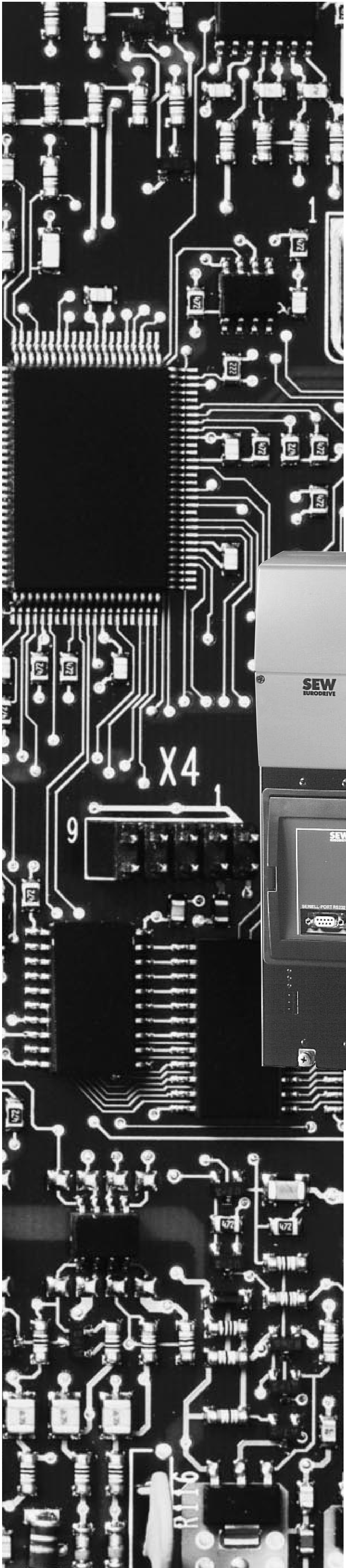


MOVIDYN®
Servo Controller

Manual

AFC 11A "CAN-Bus" Option card

Edition 08/99

16/042/95

CAN

0922 6567 / 0999

TÜV CERT DIN EN ISO 9001

SEW EURODRIVE

Preface

This *AFC11A (CAN Bus) Option User Manual* describes the procedure for installation of the CAN option card into the servo controller and for startup of the MOVIDYN® servo controller when connected to a CAN fieldbus system.

In addition to describing all the settings on the fieldbus option card, this manual further discusses the various options for connecting the servo controller to the CAN bus in the form of brief startup examples.

In addition to this *AFC11A (CAN) Option User Manual,* you should order the following more detailed documentation on fieldbuses for quick and easy connection of the MOVIDYN® to the CAN fieldbus system

– MOVIDYN® Fieldbus Unit Profile Manual (0922761X)
– MOVIDYN® Parameter List (09212868)

The *MOVIDYN® Fieldbus Unit Profile Manual* gives a detailed description of the fieldbus parameters and their codings and discusses various control concepts and application options in the form of brief startup examples (such as in S5 syntax).

The *MOVIDYN® Parameter List* contains a list of all servo controller parameters that can be read or written via the various communication interfaces, such as the RS-232 and RS-483, and the fieldbus interface.

Contents

1 Introduction . . . 6

2 Installation . . . 8
- 2.1 Supported Servo Controller Types . . . 8
- 2.2 Scope of Delivery . . . 8
- 2.3 Installation of Option card . . . 8
- 2.4 Pin Assignment . . . 10
- 2.5 CAN Bus Transmission Line . . . 11
- 2.6 Setting the DIP Switches . . . 12
- 2.7 Display Elements . . . 13

3 Project Planning and Startup . . . 14
- 3.1 Message Types . . . 14
- 3.1.1 Synchronization Message . . . 15
- 3.1.2 Process Data Messages . . . 16
- 3.1.3 Parameter Messages . . . 16
- 3.2 Configuration Examples . . . 16
- 3.2.1 Configuration Example 1 . . . 16
- 3.2.2 Configuration Example 2 . . . 18

4 Parameter Setting via CAN Bus . . . 21
- 4.1 Services . . . 21
- 4.1.1 Structure of the Parameter Message . . . 21
- 4.1.2 READ . . . 24
- 4.1.3 WRITE . . . 25
- 4.1.4 READ-SYNC . . . 26
- 4.1.5 WRITE-SYNC . . . 27
- 4.1.6 Parameter Adjustment in a CAN Environment . . . 28
- 4.1.7 Parameter Data Format . . . 28
- 4.2 Drive Parameter Description . . . 28

5 Parameter Adjustment Return Codes . . . 29
- 5.1 Special Cases . . . 29
- 5.1.1 Parameter Adjustment Errors . . . 29

6 Technical Data of the AFC11A Option . . . 30

Important Notes

- **Read this User Manual carefully before you start installation and startup work on MOVIDYN® servo controllers with CAN Bus options.**
  This user manual assumes that the user is familiar with and has at his disposal all relevant documentation on the MOVIDYN® system, particularly the installation and operating instructions.

- **Safety notes:**

  Always follow the safety notes contained in this user manual.
  Safety notes are marked as follows:

Electrical hazard, e.g. when working on live wires.

Mechanical hazard, e.g. when working on hoists.

Important instructions for safe and fault-free operation
e.g. pre-setting before startup.

- **General safety notes for bus systems:**

  The fieldbus option gives you a communications system, which allows you to match the MOVIDYN® drive system to the specifics of your application. As with all bus systems there is the risk of parameter changes, which are not visible from the outside but affect the behavior of the servo controller. This may result in unexpected but not uncontrolled system behavior.

- **In this manual, cross-references are marked with a →, for example,**
  (→ MD_SHELL) means: Please refer to the MD_SHELL User Manual for detailed information or information how to carry out this instruction.
  (→ Section x.x) means: Further information can be found in section x.x of this User Manual.

- Each unit is manufactured and tested to current SEW-EURODRIVE technical standards and specifications.
  The manufacturer reserves the right to make changes to the technical data and designs as well as the user interface herein described, which are in the interest of technical progress.
  A requirement for fault-free operation and fulfilment of any rights to claim under guarantee is that these instructions and notes are followed.
  These instructions contain important information for servicing, they should therefore be kept in the vicinity of the unit.

1 Introduction

Thanks to its high-performance, universal fieldbus interface, the MOVIDYN® servo controller with the *AFC11A (CAN Bus)* option enables connections to be made with higher-level automation systems via the open and standardized CAN bus system.

CAN Bus

The CAN bus (controller area network) is a serial, message-oriented communications protocol. The CAN bus allows stations which are connected to this bus to communicate with each other with a high degree of efficiency. The CAN protocol is defined in CAN specification 2.0, Parts A and B and fairly common in industrial applications due to the availability of corresponding ASICs from different manufacturers.

Typical CAN bus characteristics are:

- Transmission rate up to 1 Mbit/sec
- High degree of data security due to highly efficient fault detection mechanisms
- Wide availability on the market
- Flexible and effective message management (the extended CAN protocol is supported)
- Priority controlled exchange of messages

In addition, this bus system has multimaster capability, also allowing broadcast messages to be sent.

MOVIDYN® and CAN

The servo controller unit profile for CAN mode, i.e. the way the servo controller operates and responds when in CAN mode, is independent of the type of fieldbus, and thus consistent for all fieldbus types. This allows the user to develop his drive applications independent of a particular fieldbus or change to another bus system, e.g. the open standardized PROFIBUS-DP/FMS (*AFP11A* option) fieldbus system or the Interbus-S (*AFI11A* option).

MOVIDYN® offers digital access to all drive parameters and functions via the CAN bus interface. The servo controller is controlled by the high-speed process data. These process data messages provide the facility to specify setpoints such as setpoint speeds, ramp generator times for acceleration and deceleration etc., as well as various drive functions such as enable, controller inhibit, stop, rapid stop, etc. to be triggered. These messages can also be used to read back actual values from the servo controller, such as actual speed, current, unit status, error number or reference messages.

The exchange of parameter data via the READ, READ-SYNC and WRITE, WRITE-SYNC services enables applications where all major drive parameters are stored in the higher-level automation unit to be implemented, thus avoiding manual adjustment of parameters on the servo controller itself, which can often be very time-consuming.

Process data and the drive parameters can be sent synchronously or asynchronously to a synchronization message.

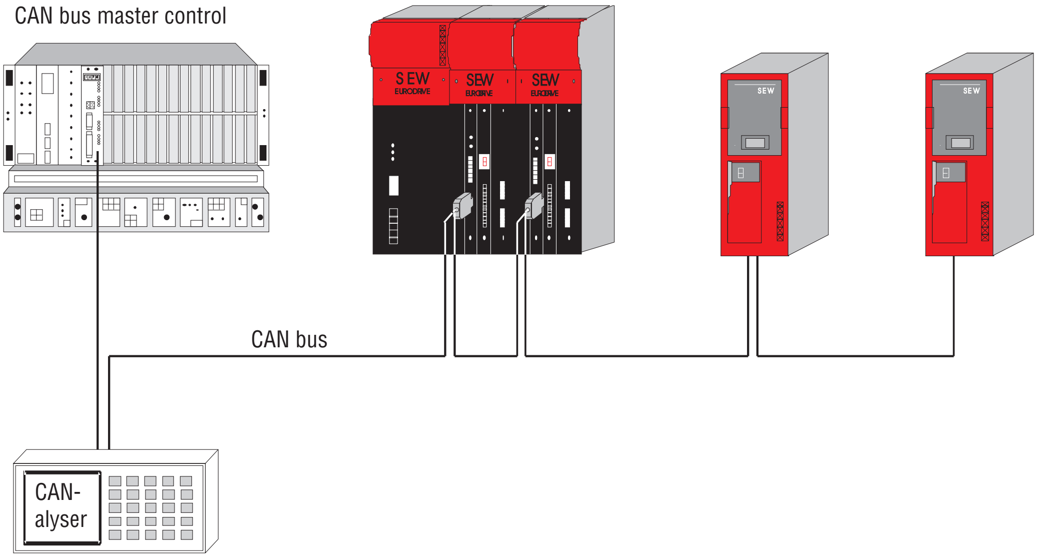


Fig. 1: MOVIDYN® in a CAN bus environment MD0205AE

The CAN option card has several DIP switches, i.e. for setting the baud rate, process data length, the basic ID of the module, and a DIP switch for connecting a terminating resistor.

The use of a fieldbus system in drive technology requires additional monitoring functions, such as fieldbus timeout or special emergency stop concepts. The monitoring functions of the MOVIDYN® can be matched to the specific application for which it is to be used. This feature enables you to specify which error response the servo controller should trigger if an error occurred in the bus. A rapid stop will be practical for many applications, but it is also possible to freeze the last setpoints, so that the drive can continue with the last valid setpoints (e.g. conveyor belt). As the functionality of the control terminals is also ensured when the servo controller is operated in the fieldbus mode, fieldbus-independent emergency stop concepts can still be implemented via the servo controller terminals.

The MOVIDYN® servo controller offers numerous diagnostic facilities for startup and servicing. The PC software MD_SHELL offers even more convenient diagnostic facilities in that it provides a detailed display of the fieldbus and unit status information as well as the facility to set all the drive parameters (including the fieldbus parameters).

2 Installation

Unless the *AFC11A* option is already installed in the MOVIDYN® servo controller, please check the components listed in the “Scope of Delivery” section below for completeness.

2.1 Supported Servo Controller Types

The *AFC11A* option for connection to a CAN bus can be used with all servo controllers of the MOVIDYN® family, firmware version 1.50 or higher.

To adjust the fieldbus parameters, you further need the *MD_SHELL* PC user interface, version V1.50 or higher.

2.2 Scope of Delivery

If you have ordered the *AFC11A* option package separately, please check the components delivered for completeness. The *AFC11A* option comprises the following components:

- 1 *AFC11A* option card
- 1 front cover for the *AFC11A* CAN option card
- 2 fastening screws for the front cover

2.3 Installing the Option Card

Please follow the instructions below when installing the option card:

Basic information:

The *AFC11A* option must only be installed by electronic equipment specialists, observing the following ESD protection measures:

- The person carrying out the work must be earthed before and during the work.
- Earth the unit and work bench.
- Store the option card in its original packaging and only take it out when it is required.
- Hold the option card by its edge and do not touch unnecessarily.

Procedure for installing the option card:

1. Ensure that the entire servo controller system is voltage-free. Switch off the mains power and, if necessary, the external 24V supply.

2. Take off the left front cover after removing the two recessed head screws.

3. Fit the *AFC11A* option card into the slot until it snaps into place in the backplane bus connector.

4. Fasten the front cover of the *AFC11A* option card with the two recessed head screws.

5. The *AFC11A* option card is now completely installed.

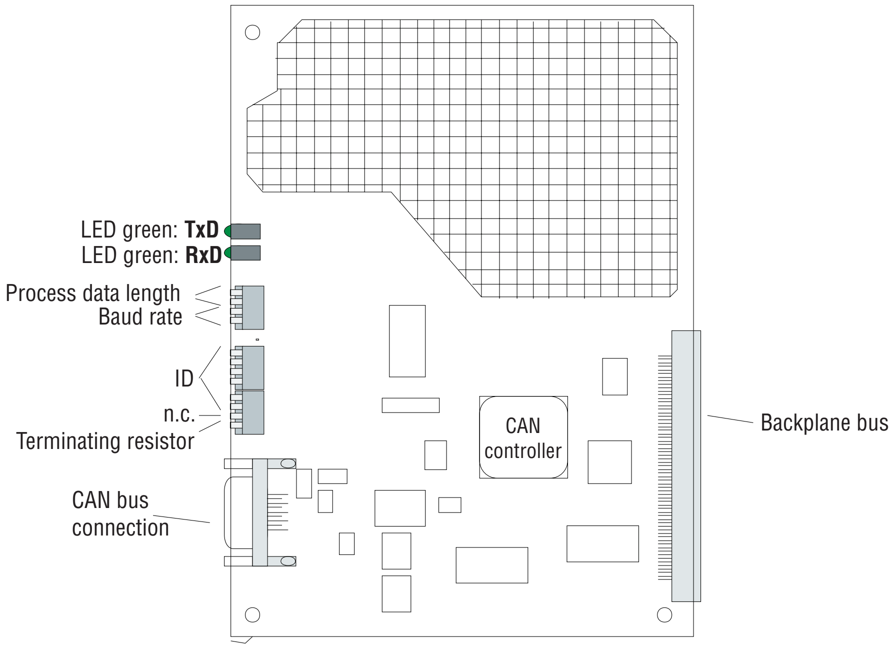


Fig. 2: The *AFC11A* option card MD0206AE

2.4 Pin Assignment

The *AFC11A* option card is fitted with a 9-pin type D connector. The CAN bus is connected to the 9-pin type D connector (pins, male). Since the spur line must be kept as short as possible, T-connectors may not be used. The bus cables must be looped through directly in the connector (see *Fig. 3*).

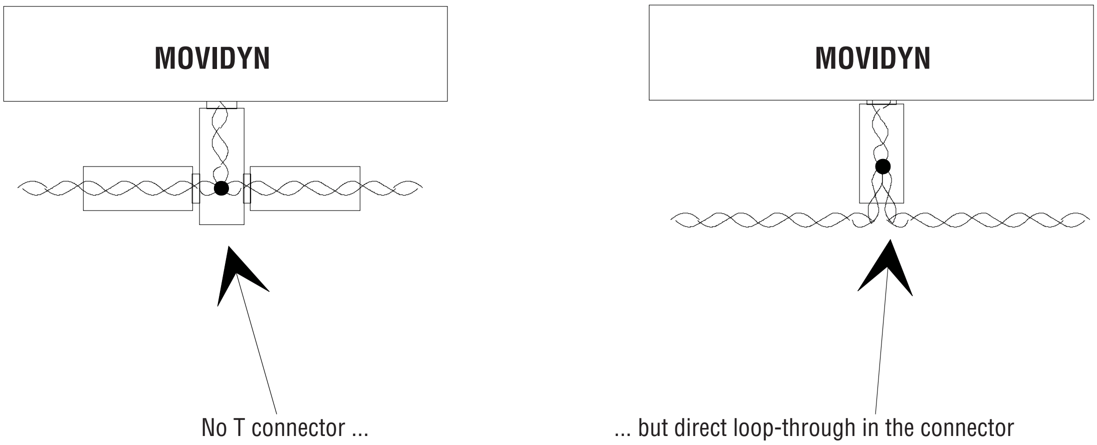


Fig. 3: Wiring of the CAN bus (top view) MD0207AE

Fig. 4 shows the pin assignment of the 9-pin type D connector which complies with the CiA (CAN in Automation) recommendations. To prevent EMC problems, use only plug connections with metal-plated housings for connecting the servo controller to the CAN bus.

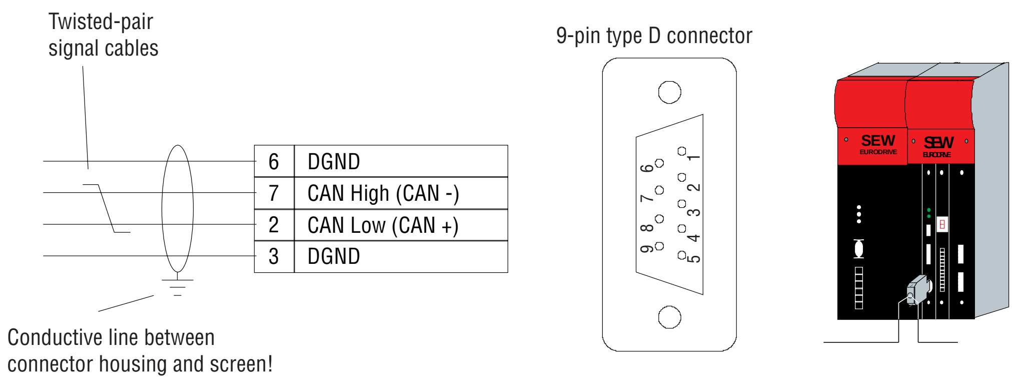


Fig. 4: Assignment of the 9-pin type D CAN bus connector MD0208AE

The MOVIDYN® servo controller is connected to the CAN bus system via the 2-wire remote bus by a 4-core screened cable with twisted-pair signal cables. The 2-wire bus basically consists of an RS-485 channel (signal lines CAN High [CAN -] and CAN Low [CAN +]).

2.5 CAN Bus Transmission Line

The CAN *AFC11A* option card supports the RS-485 transmission technology and requires the 2-core, screened, twisted-pair cable as a physical medium . The signal earths can be connected via two additional cables.

The maximum cable length depends on the set baud rate. Use the formula below to calculate the max. cable length:

$$\text{Cable length}_{max} < \frac{40 \times \frac{\text{MBit}}{\text{Second}} \cdot \text{Meter}}{\text{Baud rate}}$$

$$[\text{Baud rate}] = \frac{\text{MBit}}{\text{Second}}$$

Fig. 5 illustrates the relationship between maximum cable length and set baud rate.

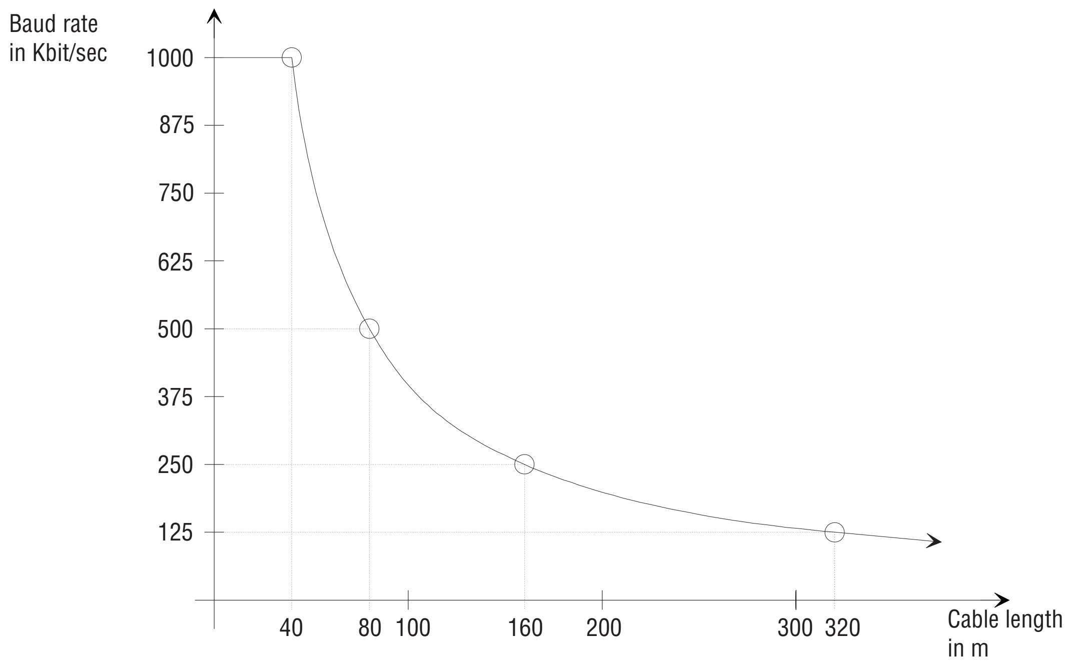


Fig. 5: Relationship between max. cable length and baud rate MD0209AE

2.6 Setting the DIP Switches

After the CAN option card has been installed, the DIP switches on the card must be set. You may change the DIP switch setting when the CAN option card is installed.

The function of the DIP switches on the CAN option card are shown in *Table 1*.

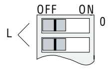
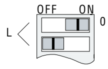
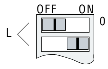
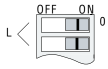
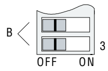
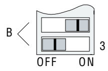
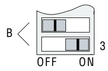
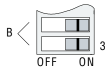
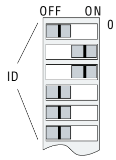
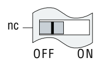
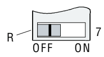
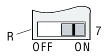
DIP Switch Block	Designation	Bit	Function	Assignment
S1	L	1-0	Process data length	 : not dedfined  : 1 process data word  : 2 process data words  : 3 process data words
S1	B	3-2	Baud rate	 : 125 kbaud  : 250 kbaud  : 500 kbaud  : 1 Mbaud
S2	ID	5-0	Identifier	 here e. g.: ID = 6
S2	nc	6	Reserved	 : Switch setting OFF
S2	R	7	Terminating resistor for the CAN bus cable	 : None  : 120 Ω

Table 1: DIP switch settings

The DIP switches on switch block *S1* (see *Fig. 6*) serve to set the process data length (identified by *'L')* and the baud rate (identified by *'B'*).

The DIP switches on switch block *S2* (see *Fig. 6*) serve to set the basic ID and the terminating resistor. The basic ID comprises 6 DIP switches. These are followed by a reserved dip switch, which is to be set to the *OFF* position. The bottom switch on the block serves to connect the terminating resistor of *120* Ω for the bus cable. This terminating resistor is needed for the first and the last unit of the bus line to avoid reflections.

The CAN option card must be initialized each time the DIP switch setting is changed. To do this, switch the supply off and then on again.

2.7 Display Elements

The *AFC11A* option card has two LEDs for diagnosing the CAN bus system. These LEDs provide information about the status of the CAN bus system. *Fig. 6* shows these diagnostic LEDs; the messages are shown in *Table 2*.

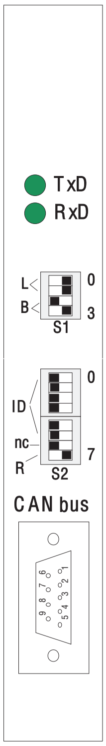


Fig. 6: Diagnostic LEDs for the CAN bus MD0222AE

Name of LED	Colour	State	Meaning
TxD	green	blinking	option card is sending messages
RxD	green	blinking	option card is receiving messages

Table 2: Meaning of the CAN bus diagnostic LEDs

3 Project Planning and Startup

This section explains project planning and startup for the MOVIDYN® servo controller with the *AFC11A* option from the CAN bus master control.

3.1 Message Types

Different message types have been defined for communication between the AFC11A option card and the master control. These message types can be divided into three categories:

- Synchronization message
- Process data messages
- Parameter messages.

These different message types must be distinguished by their ID on the CAN bus. The ID of a CAN bus message is therefore made up of the message type and the basic ID, which is set via the DIP switches. *Table 3* shows this relationship.

Identifier	Message Type
8·Basic ID + 0	Reserved
8·Basic ID + 1	Reserved
8·Basic ID + 2	Reserved
8·Basic ID + 3	Process output data message (PO), asynchronous
8·Basic ID + 4	Process input data message (PI), asynchronous
8·Basic ID + 5	Process data output message synchronizable (PO sync), synchronous
8·Basic ID + 6	Reserved
8·Basic ID + 7	Reserved
8·Basic ID + 512 + 0	Reserved
8·Basic ID + 512 + 1	Reserved
8·Basic ID + 512 + 2	Reserved
8·Basic ID + 512 + 3	Request message, synchronous/asynchronous
8·Basic ID + 512 + 4	Response message, synchronous/asynchronous
8·Basic ID + 512 + 5	Reserved
8·Basic ID + 512 + 6	Reserved
8·Basic ID + 512 + 7	Reserved

Table 3: Determining the identifiers

The examples given in Sections 3.2.1 and 3.2.2 illustrate this relationship.

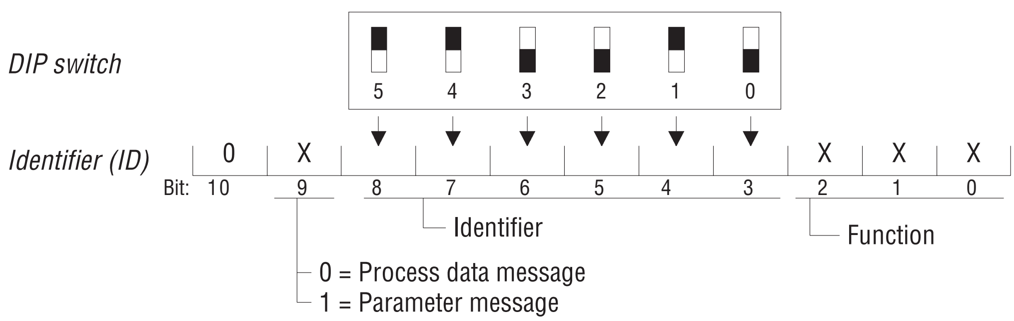


Fig. 7: Determining the identifiers

00003AEN

3.1.1 Synchronization Message

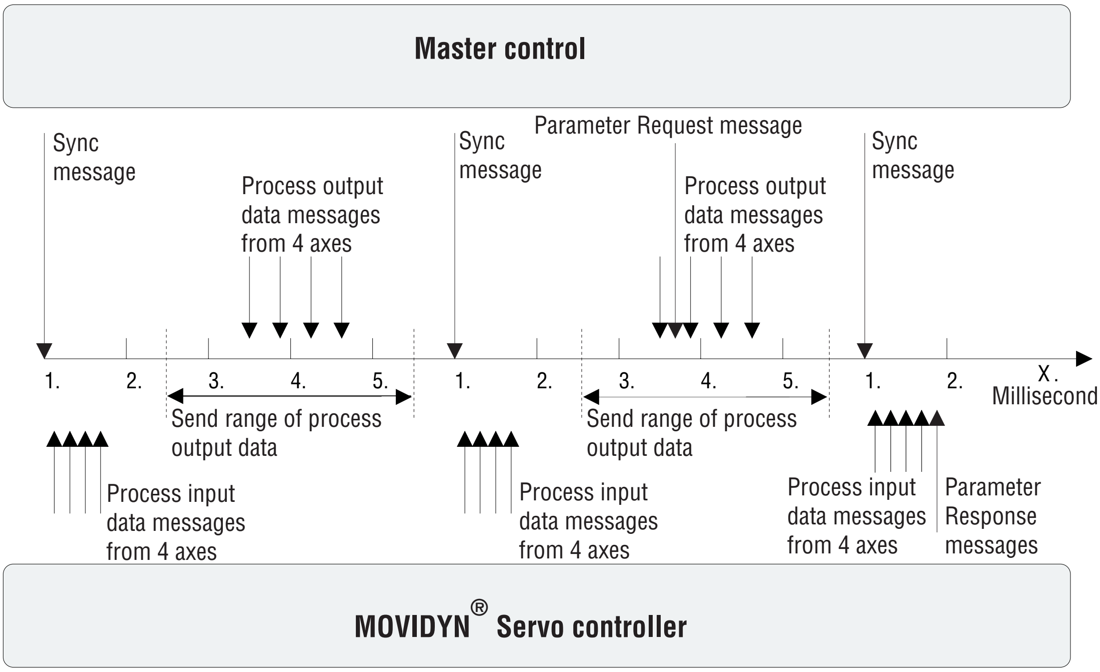


Fig. 8: The bus time is divided into bus cycles MD0223AE

A time basis of 5 milliseconds must be set for the transmission of process and parameter data. A synchronization message must then be sent every 5 milliseconds from the master control to the connected servo controllers. The servo controllers will then synchronize with the master control.

The synchronization message is a broadcast message. This means that all servo controllers will receive this message. The factory setting for the identifier of this message is 1. Any value between 0 and 2047 can be set, you must however ensure that there is no overlap of identifiers for process data and parameter data messages.

Parameter name:	**CAN SYNC ID**
Menu No.:	793
Index:	620
Save to EEPROM:	Yes
Parameter lock:	Active
Access attribute:	READ/WRITE
Coding:	BCD
Access:	READ and WRITE
Unit:	–
Minimum:	0.00
Maximum:	2047.00
Step range:	0.00 - 2047.00
Step	1.00
Factory setting:	**1.00**

Table 4: Value range of the CAN SYNC ID parameter

3.1.2 Process Data Messages

The process data are sent at certain points in time within the given time basis. A distinction is made between synchronous and asynchronous process data.

The synchronous process data are sent at certain times within the given time interval. The master control can send the process output data at the earliest 500µs after the 2nd millisecond and at the latest 500µs before the 1st millisecond (see Fig. 8). The process input data sent in response by the MOVIDYN®servo controller are transmitted in the 1st millisecond.

Asynchronous process data are not sent within the time interval. The master control can send the process output data at any point in time, the servo controller responds by sending a process input data message no later than 1 millisecond thereafter.

3.1.3 Parameter Messages

The parameter messages, too, are divided into synchronous and asynchronous messages.

The controller responds to synchronous parameter messages within the given time interval of 5 milliseconds. The response message is sent in the 1st millisecond.

The response to asynchronous parameter messages is independent of the time interval.

3.2 Project Planning

3.2.1 Project Planning Example 1

The following settings are to be entered:

- 4 axis modules
- Process data length 1
- Baud rate 125 Kbit/s

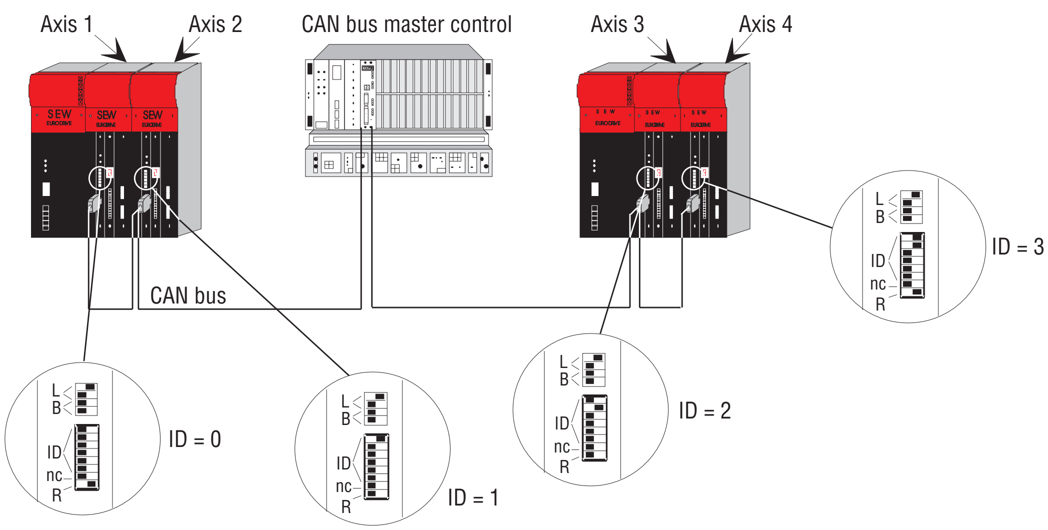


Fig. 9: Example for project planning 1

MD0225AE

Fig. 9 shows the system set up by project planning. The respective DIP switch settings are shown enlarged.

First of all you have to set the process data length and the baud rate on all the axes on the upper DIP switch block. See *Table 1* for the setting of one process data word and a baud rate of 125 Kbit/s.

Then the basic ID is set for the axes. In this example, the basic ID is allocated to the axes in accordance with *Table 5*:

Axis	Basic ID
1	000000 bin. = 0 dec.
2	000001 bin. = 1 dec.
3	000010 bin. = 2 dec.
4	000011 bin. = 3 dec.

Table 5: Assigning the basic IDs to the axes

In addition, the terminating resistor must be connected on the line ends. Set the DIP switch identified by *R* on axes 1 and 4 to *ON*.

On the CAN bus the IDs listed in *Table 6* below are assigned in this combination:

Axis	ID	Message type
1	3	Process output data message (PO)
	4	Process input data message (PI)
	5	Process output data message synchronized (PO SYNC)
	515	Parameter Request message
	516	Parameter Response message
2	11	Process output data message (PO)
	12	Process input data message (PI)
	13	Process output data message synchronized (PO SYNC)
	523	Parameter request message
	524	Parameter response message
3	19	Process output data message (PO)
	20	Process input data message (PI)
	21	Process output data message synchronized (PO SYNC)
	531	Parameter request message
	532	Parameter response message
4	27	Process output data message (PO)
	28	Process input data message (PI)
	29	Process output data message synchronize (PO SYNC)
	539	Parameter request message
	540	Parameter response message

Table 6: IDs used on the CAN bus

The IDs are calculated from the basic ID set via the DIP switches and the offset describing the corresponding message (see *Table 3*).

When the process data length = 1, the servo controller will receive exactly one process output data word and then send one process input data word to the master control. *Fig. 10* illustrates this concept.

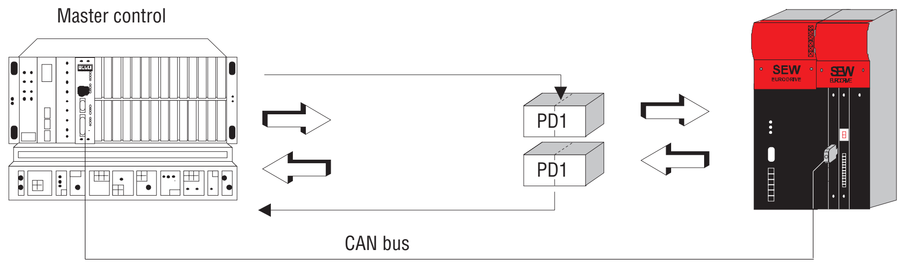


Fig. 10: Programming a process data word MD0224AE

The contents of a process data word are defined by the *Process Output Data Description 1* and *Process Input Data Description 1* parameters.

3.2.2 Project Planning Example 2

The following settings are to be entered:

- 4 axis modules
- Process data length 3
- Baud rate 500 Kbit/s

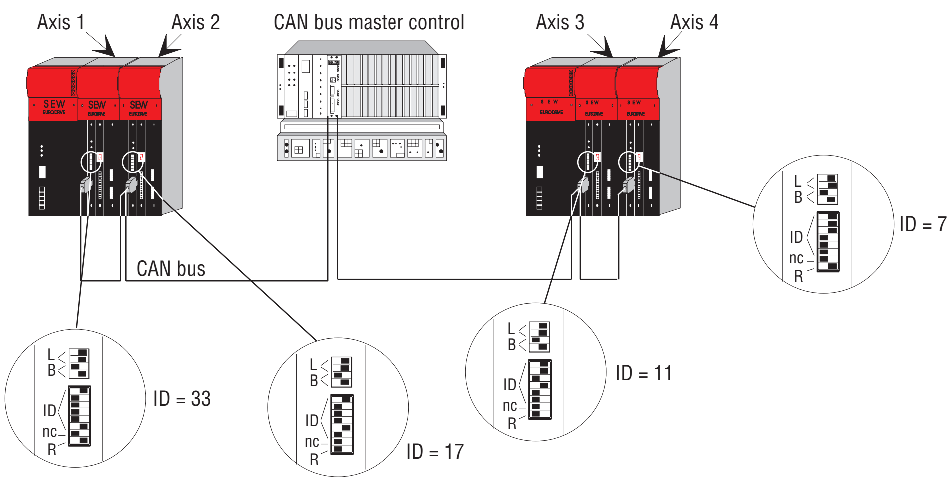


Fig. 11: Example for project planning 2 MD0226AE

Fig. 11 shows the system set up by project planning. The respective DIP switch settings are shown enlarged.

First of all you have to set the process data length and the baud rate on all the axes on the upper DIP switch block. See *Table 1* for the setting of three process data words and a baud rate of 500 Kbit/s.

Then the basic ID is set for the axes. In this example the basic ID is allocated to the axes in accordance with *Table 7*:

Axis	Basic ID
1	100001 bin. = 33 dec.
2	010001 bin. = 17 dec.
3	001011 bin. = 11 dec.
4	000111 bin. = 7 dec.

Table 7: Assigning the basic IDs to the axes

As you can see, you need not observe a specific order when setting the basic ID via the DIP switches. However, the basic IDs may not be assigned more than once, i.e. no two axes may have the same basic ID.

In addition, the terminating resistor must be connected on the line ends. Set the DIP switch identified by *R* on axes 1 and 4 to *ON*.

The IDs listed in *Table 8* below are assigned in this combination on the CAN bus:

Axis	ID	Message type
1	267	Process output data message (PO)
	268	Process input data message (PI)
	269	Process output data message synchronized (PO SYNC)
	779	Parameter Request message
	780	Parameter Response message
2	139	Process output data message (PO)
	140	Process input data message (PI)
	141	Process output data message synchronized (PO SYNC)
	651	Parameter request message
	652	Parameter response message
3	91	Process output data message (PO)
	92	Process input data message (PI)
	93	Process output data message synchronized (PO SYNC)
	603	Parameter request message
	604	Parameter response message
4	59	Process output data message (PO)
	60	Process input data message (PI)
	61	Process output data message synchronize (PO SYNC)
	571	Parameter request message
	572	Parameter response message

Table 8: IDs used on the CAN bus

The IDs are calculated from the basic ID set via the DIP switches and the offset describing the corresponding message (see *Table 3*).

When the process data length = 3, the servo controller will receive exactly three process output data words and then send three process input data words to the master control. *Fig. 12* illustrates this concept.

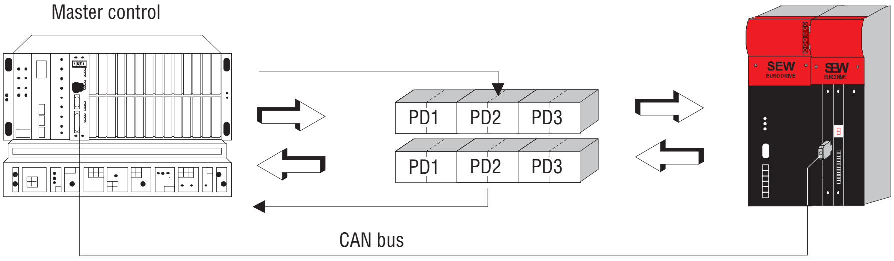


Fig. 12: Programming a process data word MD0227AE

The contents of the process data words are defined by the *Process Output Data Description 1...3* and *Process Input Data Description 1...3* parameters.

4 Parameter Setting via CAN Bus

The MOVIDYN® servo controller with the *AFC11A* option supports the services shown in *Fig. 13*.

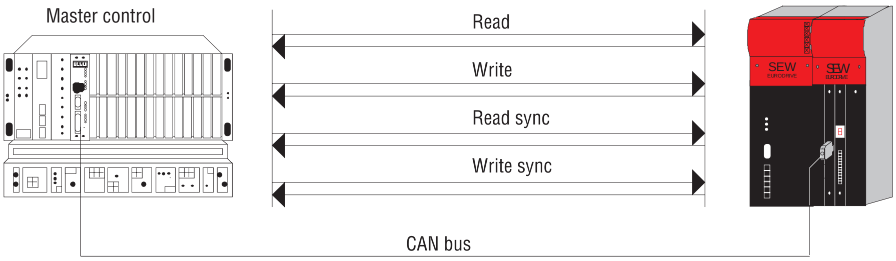


Fig. 13: CAN bus services MD0228AE

4.1 Services

The READ, READ-SYNC, WRITE and WRITE-SYNC services of the application layer (layer 7) are used to read and write controller parameters through the fieldbus system.

4.1.1 Structure of the Parameter Message

The most important functions and services, such as READ and WRITE, must be emulated to read and write parameters and set the parameters of field units through fieldbus systems which do not offer an application layer. For CAN, a parameter message is defined for this purpose. This parameter message is described by an identifier, which depends on the basic address set on the CAN option card. The parameter message enables the exchange of parameter data (Fig. 14).

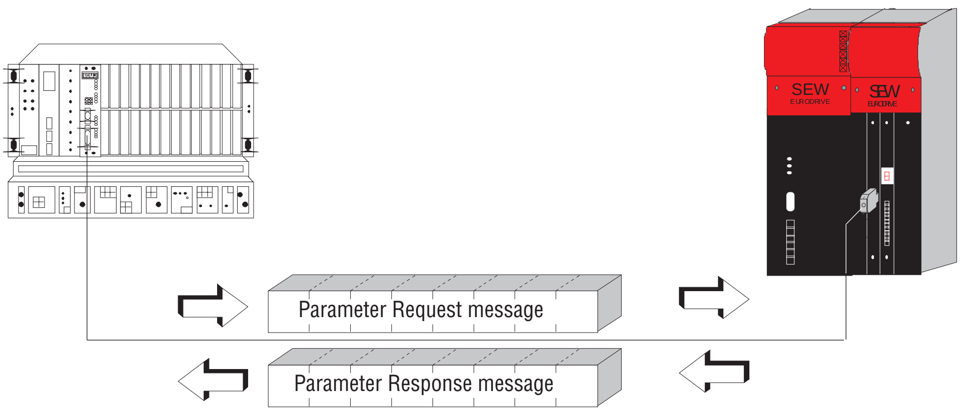


Fig. 14: Parameter message for CAN MD0229AE

Fig. 15 shows the structure of a parameter message. A parameter message on principle consists of a management byte, an index word, a reserved byte and four data bytes.

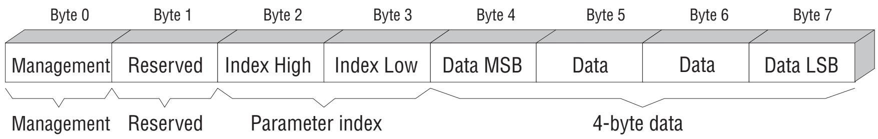


Fig. 15: Structure of the parameter message MD0230AE

4.1.1.1 Management of the Parameter Message

The complete process of parameter adjustment is coordinated through byte 0: *Management*. This byte provides important service parameters, such as service identification, data length, performance and status of the performed service. Fig. 16 shows that bits 0, 1 and 2 contain the service identification, i.e. define which service is to be performed. Bit 3 is reserved for the time being and must remain set at zero. Bits 4 and 5 specify the data length in byte for the Write service. For SEW controllers, the data length is generally set at 4 bytes.

The handshake mode bit (bit 6) determines when the servo controller is to respond to a Parameter Request message. When in asynchronous operating mode, the servo controller will send a response message immediately after the service has been performed. When in synchronous operating mode, however, the servo controller will only send a response message after it has received a SYNC message. Status bit 7 indicates whether the service was performed correctly or whether it was faulty.

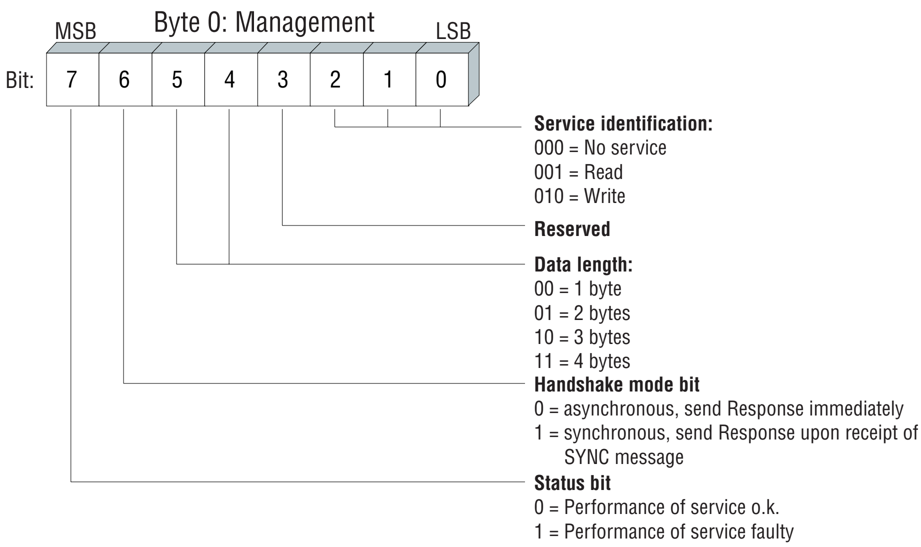


Fig. 16: Structure of the management byte MD0231AE

4.1.1.2 Index Addressing

Byte 2: *Index High* and Byte 3: *Index Low* determine the parameter which is to be read or written via the fieldbus system. Independent of the connected fieldbus system, the parameters of a servo controller are all addressed with a uniform index. Byte 1 is reserved and must generally be set at 0x00.

4.1.1.3 Data Field

As shown in Fig. 17, the data are contained in bytes 4 to 7 of the parameter message. This means that a maximum of 4 byte data can be transmitted with each service. The data are generally entered justified to the right, i.e. byte 7 contains the least significant data byte (data LSB), byte 4 and the most significant data byte (data MSB).

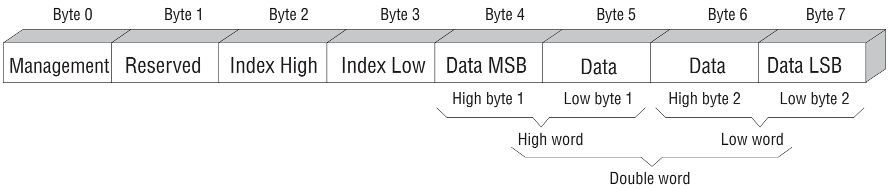


Fig. 17: Definition of the data field in the parameter message MD0232AE

4.1.1.4 Faulty Performance of Service

A faulty performance of a service is indicated by the status bit setting in the management byte. If the status bit indicates an error, the error code is entered in the data field of the parameter message (Fig. 18). Bytes 4-7 contain the return code in a structured format (see *MOVIDYN® Fieldbus Unit Profile Manual*).

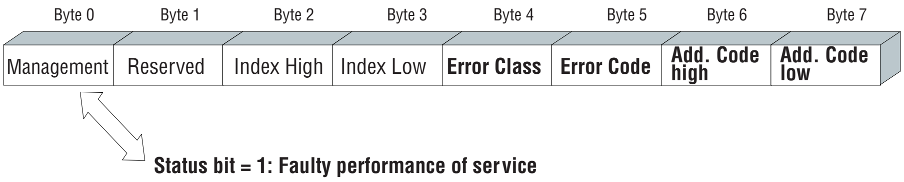


Fig. 18: Structure of the parameter message in the event of a faulty service performance MD0233AE

4.1.2 READ

The CAN bus master control uses the *READ* service to access all communications objects (drive parameters) of the MOVIDYN® servo controller for reading. For details of the drive parameters and their coding see the *MOVIDYN®* Parameter List documentation.

To read a parameter proceed as follows:

1. Enter the index of the parameter to be read in byte 2 (index high) and byte 3 (index low).

2. Enter the service identification for the READ service and the asynchronous operating mode in the management byte (byte 0).

3. Send the READ service to the controller.

Since this is a READ service, the sent data bytes (bytes 4 to 7) and the data length (in the management byte) are ignored and consequently need not be set.

The servo controller will now process the READ service and then return the service acknowledgement independent of the SYNC message. When the message is sent, the priority given by the identifier is taken into account.

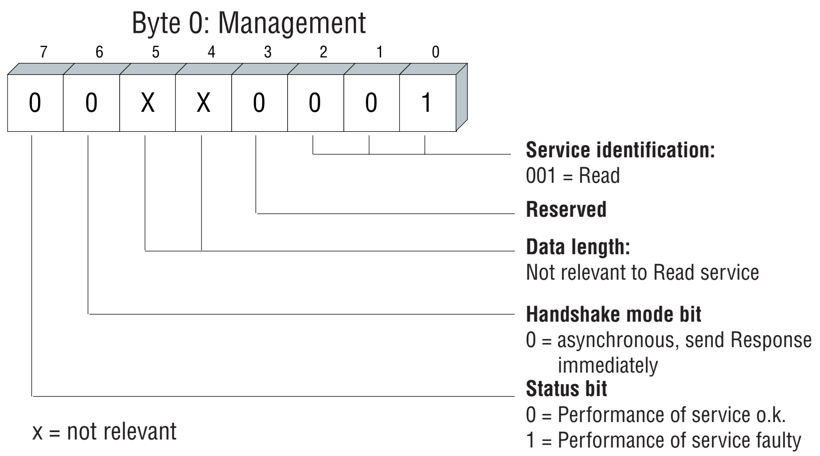


Fig. 19: Coding of the READ service in the management byte MD0234AE

Figure 19 shows the coding of the READ service in the management byte. The data length is not relevant, only the service identification for the READ service and the handshake mode bit for the asynchronous operating mode must be entered. This service is activated in the servo controller by sending the parameter message.

4.1.3 WRITE

The CAN bus master control uses the *WRITE* service to access all communications objects (drive parameters) of the MOVIDYN® servo controller for writing.

To write a parameter proceed as follows:

1. Enter the index of the parameter to be written in byte 2 (index high) and byte 3 (index low).

2. Enter the data to be written in bytes 4 to 7.

3. Enter the service identification, the data length and the handshake mode bit for the WRITE service in the management byte (byte 0).

4. Send the WRITE service to the controller.

The controller will now process the WRITE service and then return the service acknowledgement independent of the SYNC message. When the message is sent, the priority given by the identifier is taken into account.

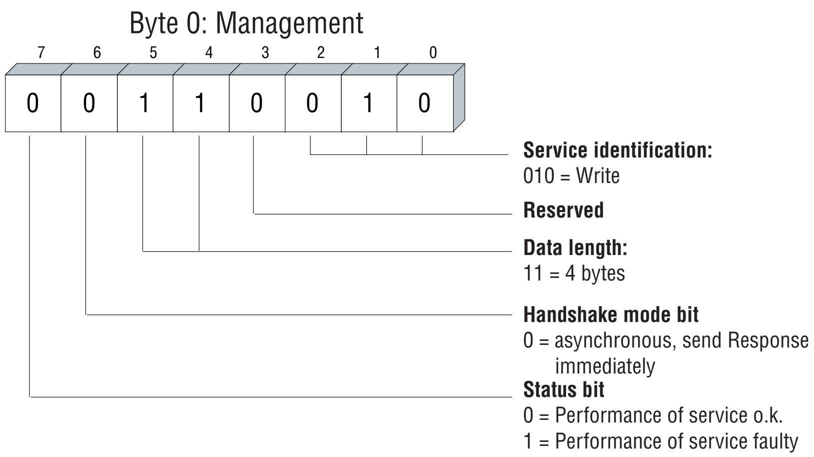


Fig. 20: Coding of the WRITE service in the management byte MD0235AE

Fig. 20 shows the coding of the WRITE service in the management byte. The data length for all SEW servo controller parameters is 4 byte. This service is activated in the servo controller by sending the parameter message.

4.1.4 READ-SYNC

The CAN bus master control uses the *READ-SYNC* service to access all communications objects (drive parameters) of the MOVIDYN® servo controller for reading. The *Read Response* is sent to the master control in the first millisecond of the defined time basis.

To read a parameter proceed as follows:

1. Enter the index of the parameter to be read in byte 2 (index high) and byte 3 (index low).

2. Enter the service identification for the READ-SYNC service and the synchronous operating mode in the management byte (byte 0).

3. Send the READ service to the controller.

Since this is a READ service, the sent data bytes (bytes 4 to 7) and the data length (in the management byte) are ignored and consequently need not be set.

The controller will now process the READ-SYNC service and then return the service acknowledgement immediately upon receipt of the SYNC message.

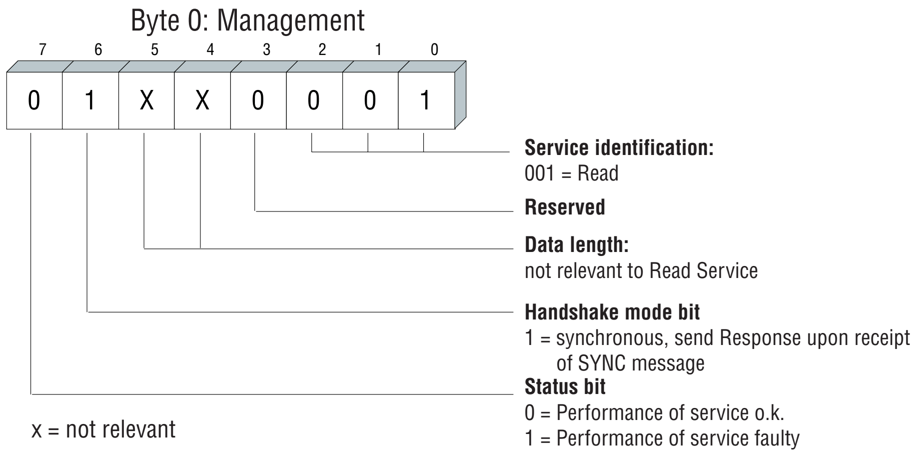


Fig. 21: Coding of the READ-SYNC service in the management byte MD0236AE

Fig. 21 shows the coding of the READ-SYNC service in the management byte. The data length is not relevant, only the service identification for the READ-SYNC service and the handshake mode bit for the synchronous operating mode must be entered. This service is activated in the servo controller by sending the parameter message.

4.1.5 WRITE-SYNC

The CAN bus master control uses the *WRITE-SYNC* service to access all communications objects (drive parameters) of the MOVIDYN® servo controller for writing.

To write a parameter proceed as follows:

1. Enter the index of the parameter to be written in byte 2 (index high) and byte 3 (index low).

2. Enter the data to be written in bytes 4 to 7.

3. Enter the service identification, the data length and the handshake mode bit for the WRITE service in the management byte (byte 0).

4. Send the WRITE-SYNC service to the controller.

The servo controller will now process the WRITE-SYNC service and return the service acknowledgement immediately upon receipt of the SYNC message.

Fig. 22 shows the coding of the WRITE-SYNC service in the management byte. The data length for all SEW servo controller parameters is four bytes. This service is activated in the servo controller by sending the parameter message.

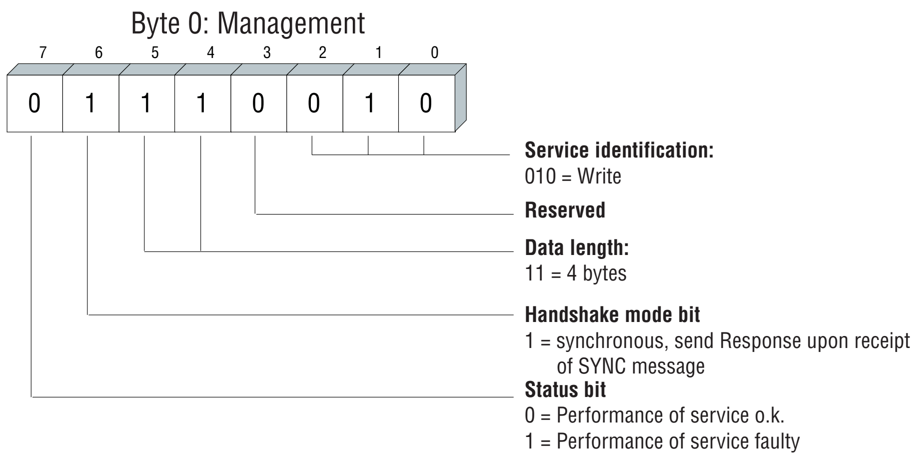


Fig. 22: Coding of the WRITE-SYNC service in the management byte MD0237AE

4.1.6 Parameter Adjustment in a CAN Environment

We will use the WRITE-SYNC service as an example to illustrate a parameter setting sequence between the higher-level control and the servo controller through CAN. For easier understanding of the sequence, Fig. 23 shows only the management byte of the parameter message.

While the higher-level control is preparing the parameter message for the WRITE-SYNC service, the servo controller receives SYNC messages and receives and returns process data messages. The service is activated after the parameter request message has been received. The servo controller will now interpret the parameter message and then process the WRITE-SYNC service. At the same time, it will respond to all process data messages. The servo controller will not send the parameter response message until it has received the SYNC message.

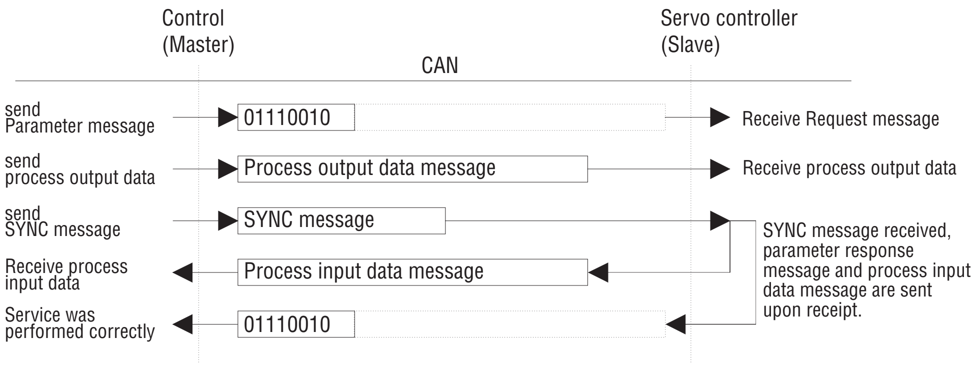


Fig. 23: Parameter adjustment via CAN MD0238AE

4.1.7 Parameter Data Format

The parameter coding used for parameter setting via the fieldbus interface is the same as the coding via RS-232 and RS-485 serial interfaces. The majority of the parameters is transmitted in 4-byte BCD format. 32-bit values are written directly into the parameter message as 4-byte hex values.

Please refer to the *MOVIDYN® Parameter List* documentation by SEW for details on the data formats and value ranges of the individual parameters.

4.2 Drive Parameter Description

The *MOVIDYN® Parameter List* documentation by SEW gives a detailed description of the drive parameters of the MOVIDYN® servo controller. In addition to the parameter index, i.e. the number which you must use to address the corresponding parameter via the communications interfaces of the servo controller, this documentation contains further information on the coding, value range and meaning of the parameter data.

You must add a value of 1000_{dec} to the index given in the parameter list in order to obtain the fieldbus index and to access all drive parameters via the CAN bus.

For READ or WRITE access to the drive parameters, the fieldbus index is

Fieldbus Index = Parameter Index + 1000_{dec}

5 Parameter Adjustment Return Codes

If the parameter adjustment has been faulty, the servo controller sends different return codes to the parameterizing master, which provide detailed information on the cause of the fault. The structure of these return codes is in accordance with DIN 19245 Part 2. They comprise the following elements

- Error class
- Error code
- Additional code.

These return codes are discussed in the *MOVIDYN® Fieldbus Unit Profile Manual*.

5.1 Special Cases

The fieldbus software describes parameter setting errors, which can neither be identified by layer 7 nor by the system software of the servo controller.
The special cases are:

5.1.1 Parameter Setting Errors

When performing a READ or WRITE service via the CAN bus, a wrong coding was entered in the management byte.

	Code (dec)	**Meaning**
Error class:	**5**	Service
Error code:	**5**	Invalid value
Add. code high:	**0**	–
Add. code low:	**0**	–

Table 9: Return code 'Incorrect coding of the management byte'

6 Technical Data of the *AFC11A* Option

Technical Data of the *AFC11A* Option

- Number of process data words:
  DIP switch selectable: 1, 2 or 3 process data words

- Baud rate:
  DIP switch selectable: 125, 250, 500 or 1000 Kbit/sec

- Basic ID: 3 ... 1020
  DIP switch selectable: 0..63

- Connection technology:
  2-wire bus with 9-pin type D connectors
  For use with the MOVIDYN® servo controller family from EPROM version 1.50
  Pin assignment to CiA standard

- Terminating resistor (120 Ω):
  Connected via DIP switch

We are available, wherever you need us. Worldwide.

SEW-EURODRIVE is your competent partner right around the globe in matters of power transmission with manufacturing and assembly plants in most industrial countries.

SEW-EURODRIVE GmbH & Co · P.O.Box 30 23 · D-76642 Bruchsal/Germany
Tel. +49-7251-75-0 · Fax +49-7251-75-19 70 · Telex 7 822 391
http://www.SEW-EURODRIVE.com · sew@sew-eurodrive.com