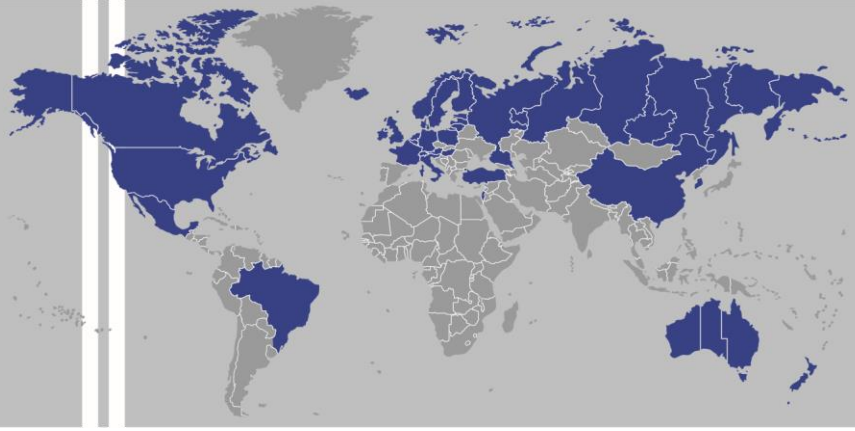

dunkermotoren
advanced motion solutions

DME 230 / 400

EtherCAT Interface
for servo drives series
- BN6773
- BN6783

Typ:	Part No:
DME 230x4-EC	81703.00102
DME 400x8-EC	81703.00111

Publication Ref: 160616

Dunkermotoren GmbH | Allmendstraße 11 | D-79848 Bonndorf/ Schwarzwald
Phone +49 (0) 7703 930-0 | Fax +49 (0) 7703 930-210/ 212 | info@dunkermotoren.com

AMETEK®
PRECISION MOTION CONTROL

TrioDrive D/ES / MidiDrive D/ES

Digital Servo Drives
for Direct Mains Connection

EtherCAT Interface

Operating Instructions 6745.232, V 1.0

These operating instructions apply to

- TrioDrive D/ES servo drives, compact design BN 6756 to BN 6758 with built-in power supply unit for single-phase AC voltage connection and integrated safety system
- MidiDrive D/ES servo drives, compact design BN 6745 to BN 6749 with built-in power supply unit for three-phase AC voltage connection and integrated safety system

with access to device functions via the EtherCAT interface (option F7)

These operating instructions are applicable together with

- Operating Instructions 6710.201 (Functions and Parameters)
- Operating Instructions 6755.202 or 6745.202 (Connection and Commissioning)
- Operating Instructions 6710.207 (SPP Windows Command and Commissioning Software)

ESR Pollmeier GmbH
Lindenstr. 20

64372 Ober-Ramstadt

Federal Republic of Germany

Tel. +49 6167 9306-0
Fax +49 6167 9306-77

E-mail info@esr-pollmeier.de
www.esr-pollmeier.de

Versions of the Document

2006-03-22	V 0.1, KS/Ri	new on the basis of the German version
2009-04-07	V 1.0, KS/Ri	section Interpolated Position Mode and corresponding variables added; list of TPDOs and RPDOs updated; minor corrections

O:\!DB\6\7\4\6745_232_10.wpd

These operating instructions have been prepared with care. However, ESR Pollmeier GmbH can accept no liability for any errors in these operating instructions or possible consequences. Neither can any liability be accepted for direct or indirect damage resulting from abuse of the device.

The relevant regulations concerning safety technology and electromagnetic compatibility must be complied with when using the device.

Subject to alteration.

Contents

Also see the *keyword index* starting on page 39.

Appendix

Figures

1 Preliminary Remarks

1.1 About this Description

These Operating Instructions 6745.232 explain the EtherCAT interface (option F7) of the digital servo drives TrioDrive D/ES (BN 6756 to BN 6758) and MidiDrive D/ES (BN 6745 to BN 6749). It particularly deals with access to the servo drive functions via EtherCAT communication interface.

The EtherCAT interface is usually used for connecting the servo drives to a higher-level controller. Function blocks described in detail in section 1.2 (page 7) are available for an integration of the servo drives in automation systems.

In principle, access to the servo drive functions is also possible via the serial interface.

These Operating Instructions 6745.232 are applicable together with

- Operating Instructions “Connection and Commissioning” of the servo drive (included in the scope of supply of the servo drive)
  - Operating Instructions 6755.202 (TrioDrive D/xS) or
  - Operating Instructions 6745.202 (MidiDrive D/xS)
- Operating Instructions “Functions and Parameters” of the servo drive (included in the scope of delivery of the servo drive)
  - Operating Instructions 6710.201
- Operating Instructions “SPP Windows Command and Commissioning Software” (delivered with the optional SPP Windows command and commissioning software)
  - Operating Instructions 6710.207

Parts of the CANopen Device Profile Drives and Motion Control CiA 402 (short: CANopen Drives Profile CiA 402 or CiA 402, standardized as IEC 61800-7-201 and IEC 61800-7-301) are implemented with the EtherCAT interface. The implemented parts of CiA 402 are described completely in these operating instructions so that the CANopen drive profile is not required for working with servo drives with EtherCAT interface.

The functions of the servo drives are accessed via the EtherCAT interface using so-called variables. For the purposes of these operating instructions, it is useful to differ between two types of variables, the drive variables and the EtherCAT variables:

- Drive variables

The drive functions of the servo drive are accessed independent of the communication interface (serial interface, EtherCAT, or another field bus interface) via drive variables (e.g. *Axis Operating Mode*, *Target Position*, *Actual Velocity*). Drive functions and drive variables of the servo drives are

described in Operating Instructions 6710.201 “Functions and Parameters”. Section “Descriptions of Variables” and appendix “List of Variables” include all information on the variables required for access via one of the communication interfaces.

You can try out these drive functions and drive variables of the servo drives without editing a program for your computer or controller or putting EtherCAT into operation. For doing so, use a PC with the SPP Windows command and commissioning software. You should not edit programs accessing these variables via the EtherCAT interface until you have learned about the functions of the servo drive using the PC and know which functions are carried out or queried by which variable access.

- EtherCAT variables

  In addition to above-mentioned drive variables, further variables are used for managing the EtherCAT interface (EtherCAT variables). The EtherCAT variables of the servo drives are described in these operating instructions. Section 12 (Descriptions of Variables) and Appendix B (List of Variables) of these operating instructions contain all information on the variables required for access via the EtherCAT interface.

Before acccessing the servo drives via the EtherCAT interface according to these operating instructions, the servo drive system (servo drive and servo motor) should have been put into operation. A PC with SPP Windows command and commissioning software is required for putting the servo drives into operation. Please check whether or not these prerequisites are met.

1.2 Function Blocks

Function blocks are available for an easy integration of the servo drives into automation systems.

These are available for various controllers according to IEC 61131-3. For further information, please contact ESR.

Communication is made in the form of SDO/PDO communication via EtherCAT.

Supported functions:

- parameterization of the servo drives by the controller (e. g. after switch-on)
- triggering of movements (relative/absolute positioning, going to home position, speed setting, ...)
- influencing the drive-integrated positioning control (part program)
- input and output of binary signals (software inputs/outputs)
- example programs for using the function library as a basis for the development of own programs

The function blocks are based on the PLCopen specification “Function Blocks for Motion Control” which is based on IEC 61131-3.

For further information see data sheet 6710.260.

The function blocks simplify the use of the functions described in these operating instructions. The parameterization steps you have to carry out yourself are described in the operating instructions of the function blocks.

2 Safety Instructions

In any case, observe the safety instructions as well as the warnings and hints in the margins of the respective Operating Instructions “Connection and Commissioning” (6755.202 or 6745.202).

Access to the servo drives via the EtherCAT interface may trigger drive movements. If drive and/or machine have not been set up and secured properly, health and life of persons may be endangered.

Therefore, access via the EtherCAT interface is prohibited until the requirements of the machine directive have been met.

In bus systems, a bus participant can be influenced invisibly from outside. This can lead to an unexpected (uncontrollable) system behavior. Do not put the bus into operation unless you have made sure that all participants are properly connected and configured.

2.1 Type of Instructions

The warnings and hints in the margin must be observed under any circumstances:

- **Danger** to health and life due to electrical shock or motion of the drive.

- **Caution**: Noncompliance violates the safety regulations or statutory provisions and can lead to personal injury or material damage.

- **Check**: Prior to commissioning and in case of failures or problems, check these items first.

- **Tip**, useful hint.

3 EtherCAT Introduction

EtherCAT is an Ethernet based field bus system setting new standards which can be handled like a field bus due to flexible topology and simple configuration. It is an open technology to be standardized in the IEC supported by the EtherCAT Technology Group (ETG), an international association of users and manufacturers with more than 900 member companies.

The EtherCAT interface is installed in the servo drives as a module (option F7). For connection, two RJ 45 connectors are provided at the front panel of the servo drives. Connector pin assignment and signal levels correspond to Ethernet standard IEEE 802.3. The bus connection is separated galvanically from the CAN controller by optocouplers.

The ESR servo drives TrioDrive D/ES and MidiDrive D/ES support CoE (CAN over EtherCAT) and variable PDO mapping.

3.1 Terms and Abbreviations

To understand these operating instructions, the reader needs to be familiar with the CANopen terms, especially the terms of CANopen Communication Profile CiA 301. This section gives a short overview of the major terms and abbreviations. It cannot replace original documents and a corresponding training, if required.

DRIVECOM

Association of drive manufacturers who developed the standards for networking drives (profiles). DRIVECOM Profile 22 for positioning drives was the basis for the development of CANopen Drive Profile CiA 402 by CiA and is implemented in the servo drive.

EMCY (Emergency)

Emergency function for transmitting faults including fault codes to the master.

PDO (Process Data Object)

Is used for fast real-time access to selected data. For certain variables or groups of variables, mappings are pre-configured on certain PDOs. Further PDOs can be can be defined by the user (variable PDO mapping).

For access to all other variables, the SDO is provided.

Profile

In communication based on bus systems, profiles are documents serving for the standardization of devices. For that, either communication functions (in a communication profile) or device functions (in a device profile) are described from the point of view of the communication interface.

RPDO (Receive Process Data Object)

PDO received by the servo drive (contains e. g. *target position*).

SDO (Service Data Object = parameter object)

The SDO provides access to all variables in a CANopen or EtherCAT device. For servo drives, these are the drive and EtherCAT variables.

Usually, the SDO is used for configuration. For fast real-time access to selected variables, PDOs are used .

TPDO (Transmit Process Data Object)

PDO sent by the servo drive (contains e. g. *actual position*).

Variable

The user can access all drive and EtherCAT functions via variables. These variables may consist of one single word (e. g. *Position Controller Kp*), but may also be very extensive (e. g. Part Program with 500 sets). They can be accessed via SDOs or PDOs. In these operating instructions, the EtherCAT variables for the servo drive are printed in *italics* and described in a separate section.

4 Connection and Commissioning

For connection and status display, the servo drives are equipped with the following elements:

- bus connection
- LEDs

These elements are located on the front panel of the servo drives with EtherCAT interface.

Coding switches are not required for these devices as with EtherCAT, the IDs are automatically assigned by the master.

4.1 Bus Connection

X4.1/F7 EtherCAT in: RJ45 connector
X4.2/F7 EtherCAT out: RJ45 connector

Connector pin assignment and signal levels correspond to Ethernet standard IEEE 802.3.

The servo drive is connected to the bus using the IN connector. Other participants may be connected to the same line using the OUT connector. If no other devices are connected, the OUT connector remains unassigned.

Due to the standard Ethernet wiring, a termination of the first or last bus participant is not required.

4.2 LEDs

Run (green) displays the status of the EtherCAT state machine:

- off: the device is in INIT state
- blinking: pre-operational
- single flash: safe-operational
- on: operational

Error (red) displays errors such as watchdog timeout and unsolicited state changes:

- off: no error
- flickering: booting error, INIT State
- blinking: general configuration error
- single flash: unsolicited state change

- double flash: application watchdog timeout
- on: PDI watchdog timeout

Aux1 and Aux2 are currently not used.

4.3 Commissioning

For commissioning, it is essential to observe the safety notes of section 2 (page 9).

Commissioning must be carried out the way described below.

4.3.1 Parameterization of the Drive Functions (Servo Drive, Servo Motor)

A PC with command and commissioning software SPP Windows is required for commissioning the ESR servo drives. For further information, see Operating Instructions 6710.201 “Functions and Parameters”.

4.3.2 Commissioning of the EtherCAT System (Master, Other Devices)

After having commissioned the servo drives, commission the EtherCAT system with the master and other devices.

5 Network Management

5.1 EtherCAT State Machine (ESM)

The EtherCAT interface communication behavior of a device is influenced by so-called network management functions. The following state diagram (ESM diagram) represents the communication behavior of the servo drive and possible actions to influence it.

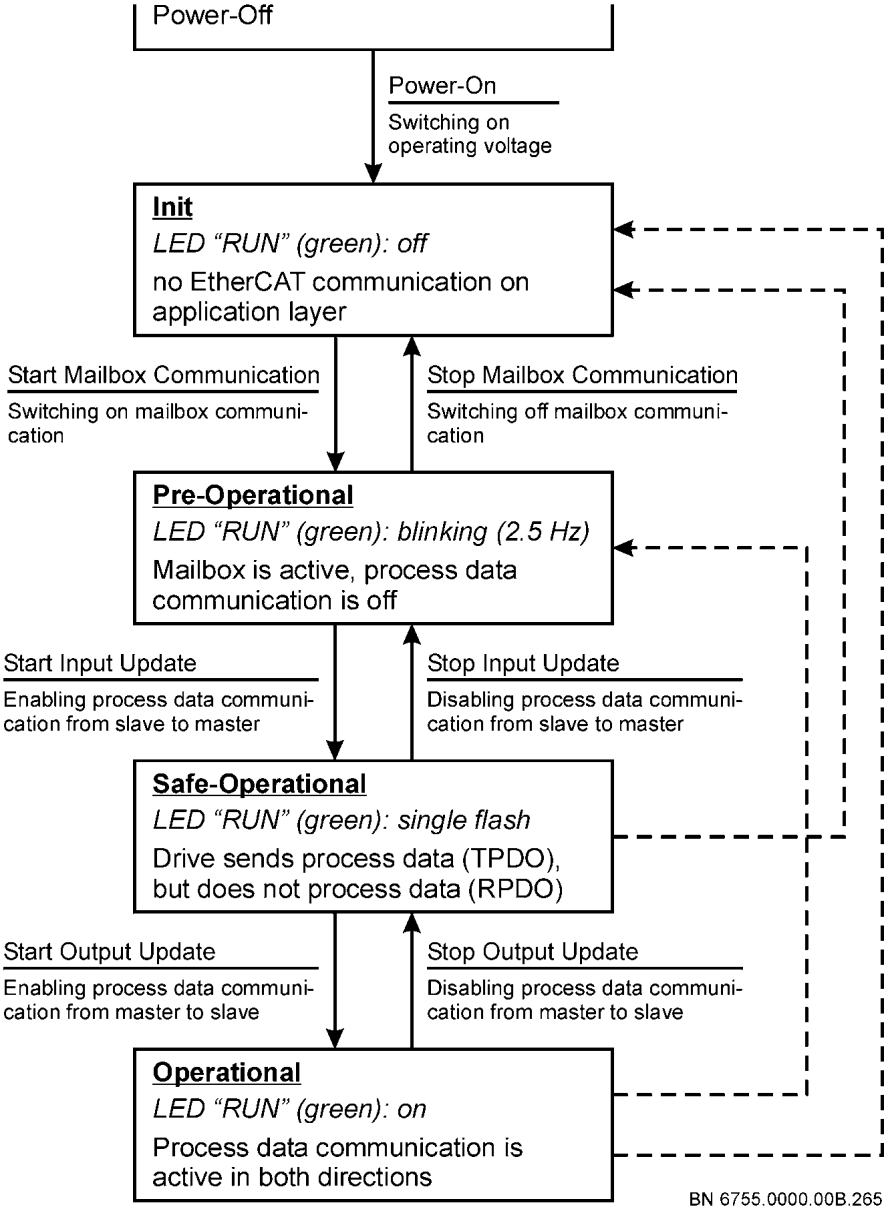

Figure 1: ESM Diagram Servo Drive

The figure shows the states and state transitions of the ESM diagram stated in the EtherCAT specification which are supported by the servo drive.

For basic information on state machines see Appendix A (page 36).

After switch-on of the control voltage and terminated internal initialization of the servo drive, the device automatically reaches state INIT.

All other state transitions are usually triggered by the master.

In the pre-operational state, only SDO communication is possible, PDO communication is blocked. In states safe-operational and operational, PDO communication takes place, however, in state safe-operational, only TPDO communication is possible.

By means of simultaneous sending of the corresponding commands, direct transitions from operational or safe-operational to pre-operation or INIT are possible.

TIP The current state is displayed by the green Run LED, see section 4.2 (page 12).

6 SDO Communication (Parameters)

All drive variables and EtherCAT variables of the servo drive can be accessed by SDO communication.

The EtherCAT variables are described in section 12 (page 25ff) of these operating instructions, the drive variables are described in Operating Instructions 6710.201 “Functions and Parameters”.

In SDO communication, the servo drive is a “server” in the sense of CANopen Communication Profile CiA 301. That means the servo drive permits other devices, called “clients”, reading or writing access to its variables.

SDO communication is possible in ESM diagram states pre-operational, safe-operational, and operational. These states are displayed by flashing or lighting of the green Run LED.

To reach a maximum flexibility in exchanging devices, it is useful to transmit the EtherCAT machine data to the servo drive via SDO communication after each system switch-on.

7 PDO Communication (Process Data)

The PDO communication is used for fast real-time access to selected variables.

According to the following table, seven mappings of variables on PDOs are predefined in sending and receiving direction each. PDO mapping can also be configured as desired (variable PDO mapping).

The following table shows the default mapping set by the manufacturer:

PDO-Designation	Description				
	Name	Index	Subindex	Length (Byte)	State
RPDO1	Axis control word	6040	0	2	active
RPDO2	Axis control word	6040	0	2	inactive
	Axis operating mode	6060	0	2	inactive
RPDO3	Axis control word	6040	0	2	inactive
	Target position	607a	0	4	inactive
RPDO4	Axis control word	6040	0	2	inactive
	Target velocity	6081	0	4	inactive
TPDO1	Axis status word	6041	0	2	active
TPDO2	Axis status word	6041	0	2	inactive
	Operating mode sel. code	6061	0	2	inactive
TPDO3	Axis status word	6041	0	2	inactive
	Actual position (in position units)	6064	0	4	inactive
TPDO4	Axis status word	6041	0	2	inactive
	Actual velocity	606c	0	4	inactive

The naming of the PDOs corresponds to the conventions of CANopen Drive Profile CiA 402.

PDO communication is possible in ESM diagram states safe-operational and operational. In the safe-operational state, only TPDOs are active (data from device to master), RPDOs are not transmitted. In the operational state, TPDOs and RPDOs are active. The states are displayed via LEDs, for a description of the LEDs, see section 4.2 (page 12).

The variables mapped on a PDO as described in the table above can be read and written via *Receive PDO 1 Mapping* to *Transmit PDO 4 Mapping*.

By default, one PDO is active (valid) in sending and receiving direction each: RPDO1 and TPDO1. All other PDOs are not active.

For a description of variables *Receive PDO 1* to *4* and *Transmit PDO 1* to *4*, see section 12.1 (page 28ff).

The standard assignment of the PDOs can be changed using the *PDO Mapping Parameters* (variable PDO mapping). 4 RPDOs and 4 TPDOs with a maximum of 8 entries per PDO are available. The following parameters can be mapped in PDOs:

Index	Object	Type	PDO Direction
1001	Error Register	u8	TPDO
5e93	I2t Load	u16	TPDO
5e95	Position sensor measured value2	i32	TPDO
5e96	Position sensor measured value2 PU	i32	TPDO
5e9f	Position sensor measured value1 PU	i32	TPDO
5f03	Resistor motor temperature sensor	u32	TPDO
5f0b	Heat sink temperature	i16	TPDO
5f0c	Motor temperature	i16	TPDO
5f1c	Position sensor measured value	i32	TPDO
603f	Fault code	u16	TPDO
6041	Status word	u16	TPDO
6061	Modes of operation display	i16	TPDO
6063	Position sensor actual position	i32	TPDO
6064	Actual position	i32	TPDO
606c	Actual velocity	i32	TPDO
6078	Actual current	i16	TPDO
6079	Bus voltage	u16	TPDO
5f54	Digital outputs	u8 (array)	TPDO, RPDO
5f56	Digital inputs	u8 (array)	TPDO, RPDO
5f5e	Program variables	i32 (array)	TPDO, RPDO
5e9c	Max current amount 2	u16	RPDO
5ef4	Acceleration time	u32	RPDO
5ef5	Deceleration time	u32	RPDO
5ef6	Quick stop time	u32	RPDO
5ef7	Ramps reference velocity	u32	RPDO
6040	Control word	u16	RPDO
6060	Axis operating mode	i16	RPDO

Index	Object	Type	PDO Direction
6071	External torque setpoint	i16	RPDO
6073	Max current amount	u16	RPDO
607a	Target position	i32	RPDO
6081	Target velocity	i32	RPDO
6086	Motion profile type	i16	RPDO
60b1	Velocity offset	i32	RPDO
60c1	Interpolation data record	i32	RPDO

The PDOs desired by the user must be activated. For that, variables *Sync Manager 2 PDO Assign* and *Sync Manager 3 PDO Assign* are available. They select the objects to be activated. For a description of these variables, see section 12.1.3 (page 31).

The list of TPDOs and RPDOs can be read out by the master if it supports this function.

8 Monitoring Mechanisms

8.1 Fault Handling (Error, Emergency)

Each fault is transmitted to the master by an emergency telegram. This emergency telegram has the following structure:

- Byte 0, 1: pre-defined error field, subindex 1 (fault code)
- Byte 2: error register
- Byte 3 .. 7: not used (0)

Bytes 0 und 1 contain the fault code of the device as reported in EtherCAT variable *Pre-Defined Error Field*, subindex 1.

Byte 2 contains the contents of EtherCAT variable *Error Register*.

Bytes 3 to 7 of the emergency telegram are not used.

8.2 Sync Manager Watchdog Timeout

In ESM state operational, the cyclic transmission of RPDOs is monitored by the SM2 watchdog. If SM2 telegrams are not sent for a certain period of time (default 100 ms, can be parameterized by the EtherCAT master), an internal fault is set. The reaction on this fault can be defined via variable *Abort Connection Option Code*.

9 Interpolated Position Mode

Axis operating mode “Interpolated Position Mode” serves for operating one or several axes for which a time interpolation of the setpoints is required (e. g. for contouring control). For that, the higher-level controller sends target positions to the drive at defined intervals via an RPDO. In addition to these positions, the fine interpolator of the drive calculates further target positions at the time interval of the position control loop (1 ms). In interpolated position mode, the target positions are transmitted in position sensor steps (PSS), only, not in position units (PU).

In case of a failure or a breakdown of the data transmission, the drive keeps the last target position until it receives new data.

Monitoring of hardware limit switches and function “measure position” are not active in operating mode interpolated position mode.

In the axis status word, operating mode-related bits do not exist for the interpolated position mode.

In the axis control word, bit 4 has the following special meaning:

- enable interpolation

  The interpolation is started by a positive edge in bit “enable interpolation”.

The axis operating mode can only be changed when the state of the axis state machine is “switch on disabled”, “ready to switch on”, or “switched on”.

Only linear axes are supported in the interpolated position mode, an operation as circular axis (endless axis) is not possible.

In operating mode interpolated position mode, the EtherCAT communication is influenced by the following parameters:

- Mapping PDOs
  - serve for the transmission of axis status word and axis control word as well as target positions (in *Interpolation Data Record*) and actual positions.

  The mapping PDOs are described in section 12.1 (page 28ff).
- Machine data interpolated position mode with *Interpolation Time Period*
  - determine the time interval for setting the target positions in milliseconds.

  For a description of the machine data interpolated position mode, see section 12.2.2 (page 32ff).

Due to the variable PDO mapping, PDOs for the interpolated position mode are no longer defined. For mapping, please note that the objects *GeraetSteuer* (index 6040, subindex 0) and *IpolDataRecord* (index 60c1, subindex 1) must be

mapped in an RPDO, objects *GeraetStatus* (index 6041, subindex 0) and *PosIstLg* (index 6063, subindex 1) must be mapped in a TPDO.

In axis operating mode interpolated position mode, the fine interpolator acts as setpoint generator together with the position controller, speed and current controller are subordinate to the position controller.

In axis operating mode interpolated position mode, the servo drives can be operated with controllers of different manufacturers. If required, please contact ESR.

9.1 Activating the Interpolated Position Mode

To control the axis state machine, the corresponding PDOs must be set to “valid”.

1. If not set via machine data *Axis operating mode at startup*, select axis operating mode interpolated position mode.
2. With enabling the PDO communication by the master, the ESM state machine is switched to “operational”.
3. For the initialization of the interpolated position mode, read out the current actual position and write it into the *Interpolation Data Record*.
4. Switch axis state machine to “operation enabled”.
5. Activate the interpolation via “enable interpolation” (bit 4 in the axis control word).

9.2 Deactivating the Interpolated Position Mode

For deactivating the interpolated position mode, the axis state machine has to be switched to state “switch on disabled”, “ready to switch on”, or “switched on”.

After that, the axis operating mode can be changed.

10 Memory Functions Command and Commissioning Software SPP Windows

The EtherCAT machine data can be stored from the RAM of the servo drive into the servo EPROM via menu item “Communication/Save in device” of command and commissioning software SPP Windows.

11 Device Information

Information on connected devices such as device type and manufacturer are also included in variables.

For a description of the variables *Device Type*, *Manufacturer Device Name*, *Manufacturer Hardware Version*, *Manufacturer Software Version*, and *Identity* which are used for device information, see section 12.5 (page 34ff).

12 Description of Variables

In the following sections, you will find information on the EtherCAT variables described in these operating instructions. In this part, the variable descriptions are sorted according to functions. Access according to the name is possible via the keyword index (page 39). In Appendix B, all EtherCAT variables are listed according to the index.

The description of variables is structured according to a standardized scheme. With corresponding adaptations, this scheme is also used for the structure of the list of variables in Appendix B (page 37) as well as for the description of variables in Operating Instructions 6710.201 “Functions and Parameters”.

Basic information on a variable is summarized in a table of the following form. The values stated in this table are always default values.

Name		Index: 1234, Short name: VarName
SPP Windows	...	
Type	...	
Access	...	
Variable type	...	
Unit	...	
Default value	...	

Name

The variable name is printed in bold type at the left top of each variable description. In the describing text, the variable name is printed in italics.

Index

The index is a hexadecimal number by which the variable can be accessed in SDO communication via EtherCAT.

Short name

This is a short name intended for a future use with special high-level language driver programs.

SPP Windows

In the SPP Windows command and commissioning software, the variable is displayed as stated in these operating instructions. As an example, this line contains the default factory setting (underlined). If a unit is assigned to this value, it is stated together with the value.

Type

This line is composed of various information:

- Object code:
  - Simple variable

    The simple variable contains one element of the stated data type each. In case of access to the simple variable, subindex 0 must always be stated.
  - Array

    The array contains several elements of the stated data type. In case of access to the array, the subindex of the desired element (1 or higher) has to be stated. Especially with EtherCAT variables, the number of elements of the array can be read out via subindex 0.
  - Record

    A record contains several elements of different data types. In case of access to the record, the subindex of the desired element (1 or higher) has to be stated. Especially with EtherCAT variables, the number of elements of the record can be read out via subindex 0.
- Data type:

  Possible values:
  - Boolean (Bool)
  - Integer8 (i8) = byte in double complement representation
  - Integer16 (i16) = word in double complement representation
  - Integer32 (i32) = double word in double complement representation
  - Unsigned8 (u8) = byte, without sign
  - Unsigned16 (u16) = word, without sign
  - Unsigned32 (u32) = double word, without sign
  - Float (float) = floating point, simple accuracy (32 bit)
  - Visible string (VisStr) with length = a series of bytes containing text
  - Octet string (OctStr) with length = a series of bytes containing binary coded information
  - PDO Mapping (PDOMap)
  - SDO Parameter (SDOPar)

Access, R/W

Possible values:

- read and write, short R W
- read only, short R

Variable type

Possible values (with characters as abbreviations):

– constant (variable value does not change), short “F”

– machine data EtherCAT (all machine data described in these operating instructions), short “ME“

– machine data, short “M”

– status information, short “S”

– control information, short “C”

– variable value, short “V”

Array element n

In case the individual elements of the array have certain names, units, and/or default values, these are stated in this line.

Record element n

In case the individual elements of the record have certain names, units, and/or default values, these are stated in this line.

Description

Depending on the variable, a short describing text appears before and/or after the table.

Example of a variable description:

Receive PDO 1 Mapping	Index: 1600, Short name: RPDO1Mapping
SPP Windows	Receive PDO 1 Mapping 1 60400010
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 1
Record element 1	Axis Control Word (index 6040, subindex 0, length 16 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

This variable shows the variable mapped on *Receive PDO 1*.

12.1 Description of Variables PDO Mapping

12.1.1 Receive PDOs Mapping

The PDOs described in the following can be found in SPP Windows in window “Parameterization” at “EtherCAT/Receive PDO Mapping”.

The variable descriptions of *Receive PDO...Mapping* show which function is accessed via the respective PDO (e. g. *Axis Control Word*). These variables are records with a type unsigned32 element for each variable mapped on the PDO. The following is entered for each mapped variable:

Record element x (u32)	bit 31 .. 16:	index (16 bit)
	bit 15 .. 8:	subindex (8 bit)
	bit 7 .. 0:	length in bit (8 bit)

The number of mapped variables (corresponds with the number of elements of the record) can be read and written via subindex 0.

The standard mapping described above can be overwritten (variable mapping). To do so change the number of mapped variables (subindex 0) to 0. After having changed the mapping, set the number of variables to the corresponding number of mapped elements (0–8).

The RPDO mapping cannot be changed in state operational, the TPDO mapping cannot be changed in states safe operational and operational.

Receive PDO 1 Mapping	Index: 1600, Short name: RPDO1Mapping
SPP Windows	Receive PDO 1 Mapping 1 60400010
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 1
Record element 1	Axis Control Word (index 6040, subindex 0, length 16 bit)
Access	read/write (via EtherCAT and serial)
Type	machine data EtherCAT

Receive PDO 2 Mapping	Index: 1601, Short name: RPDO2Mapping
SPP Windows	Receive PDO 2 Mapping 1 60400010 Receive PDO 2 Mapping 2 60600010
Type	record, max. 8 elements (subindex 0 .. 8), type PDO-Mapping
Record element 0	number of elements (0-8), default = 2
Record element 1	Axis Control Word (index 6040, subindex 0, length 16 bit)
Record element 2	Axis Operating Mode (index 6060, subindex 0, length 16 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

Receive PDO 3 Mapping	Index: 1614, Short name: RPDO3Mapping
SPP Windows	Receive PDO 3 Mapping 1 60400010 Receive PDO 3 Mapping 2 607A0020
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 2
Record element 1	Axis Control Word (index 6040, subindex 0, length 16 bit)
Record element 2	Target Position (index 607a, subindex 0, length 32 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

Receive PDO 4 Mapping	Index: 1615, Short name: RPDO4Mapping
SPP Windows	Receive PDO 4 Mapping 1 60400010 Receive PDO 4 Mapping 1 60810020
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 2
Record element 1	Axis Control Word (index 6040, subindex 0, length 16 bit)
Record element 2	Target Velocity (index 6081, subindex 0, length 32 Bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

12.1.2 Transmit PDOs Mapping

The PDOs described in the following can be found in SPP Windows in window “Parameterization” on “EtherCAT/Transmit PDO Mapping”.

The variable description of *Transmit PDO...Mapping* shows which function is accessed via the respective PDO. These variables are records with a type unsigned32 element for each variable mapped on the PDO. The following is entered for each mapped variable:

Record element x (u32)	bit 31 .. 16:	index (16 bit)
	bit 15 .. 8:	subindex (8 bit)
	bit 7 .. 0:	length in bit (8 bit)

The number of mapped variables (corresponds with the number of elements of the record) can be read and written via subindex 0.

Transmit PDO 1 Mapping	Index: 1a00, Short name: TPDO1Mapping
SPP Windows	Transmit PDO 1 Mapping 1 60410010
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 1
Record element 1	Axis Status Word (index 6041, subindex 0, length 16 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

Transmit PDO 2 Mapping	Index: 1a01, Short name: TPDO2Mapping
SPP Windows	Transmit PDO 2 Mapping 1 60410010 Transmit PDO 2 Mapping 2 60600010
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 2
Record element 1	Axis Status Word (index 6041, subindex 0, length 16 bit)
Record element 2	Axis Operating Mode Selection Code (index 6061, subindex 0, length 16 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

Transmit PDO 3 Mapping	Index: 1a14, Short name: TPDO3Mapping
SPP Windows	Transmit PDO 3 Mapping 1 60410010 Transmit PDO 3 Mapping 2 60640020
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 2
Record element 1	Axis Status Word (index 6041, subindex 0, length 16 bit)
Record element 2	Actual Position (index 6064, subindex 0, length 32 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

Transmit PDO 4 Mapping	Index: 1a15, Short name: TPDO4Mapping
SPP Windows	Transmit PDO 4 Mapping 1 60410010 Transmit PDO 4 Mapping 2 606C0020
Type	record, max. 8 elements (subindex 0 .. 8), type PDO Mapping
Record element 0	number of elements (0-8), default = 2
Record element 1	Axis Status Word (index 6041, subindex 0, length 16 bit)
Record element 2	Actual Speed (index 606c, subindex 0, length 32 bit)
Access	read/write (via EtherCAT and serial)
Variable type	machine data EtherCAT

12.1.3 PDO Assignment

The variables described below can be used for selecting the objects for variable PDO mapping.

Sync Manager 2 Communic. Type	Index: 1c00, Short name: SyncManagerCommType
SPP Windows	Communication type sync manager 0 Mailbox receive (master/slave) Communication type sync manager 1 Mailbox send (slave/master) Communication type sync manager 2 Proc. data outp. (master/slave) Communication type sync manager 3 Process data inp. (slave/master)
Type	array, 4 elements (subindex 0 .. 4), type unsigned8
Access	read
Variable type	constant (variable value does not change)

Sync Manager 2 PDO Assign	Index: 1c12, Short name: SyncMan2PDOAssign
SPP Windows	Sync Manager 2, Anzahl RPDOs 1 RPDO Mapping object index 1 1600 RPDO Mapping object index 2 1601 RPDO Mapping object index 3 1602 RPDO Mapping object index 4 1603
Type	array, 4 elements (subindex 0 .. 4), type unsigned16
Access	read and write
Variable type	machine data EtherCAT

Valid values for subindex 1 to 4 are the object indices of the RPDO mapping, i. e. 1600_{hex} to 1603_{hex}.

Sync Manager 3 PDO Assign	Index: 1c13, Short name: SyncMan3PDOAssign
SPP Windows	Sync Manager 2, Anzahl TPDOs 1 TPDO Mapping object index 1 1a00 TPDO Mapping object index 2 1a01 TPDO Mapping object index 3 1a02 TPDO Mapping object index 4 1a03
Type	array, 4 elements (subindex 0 .. 4), type unsigned16
Access	read and write
Variable type	machine data EtherCAT

Valid values for subindex 1 to 4 are the object indices of the RPDO mapping, i. e. $1a00_{hex}$ to $1a03_{hex}$.

12.2 Description of Variables Interpolated Position Mode

12.2.1 Process Data Mapping Interpolated Position Mode

Interpolation Data Record	Index: 60c1, Short name: IpolDataRecord
SPP Windows	–
Type	record, 1 element (subindex 0 .. 1), integer32
Record element 1	target position, interpolated position mode (index 60c1, subindex 1, length 32 bit)
Access	read and write (via EtherCAT)
Value range	$-2^{24}..2^{24}-1$ PSS

Via this variable, the target position is transmitted (in position sensor steps)

Velocity Offset	Index: 60b1, Short name: VelocityOffset
SPP Windows	–
Type	simple variable, integer32
Access	read and write (via EtherCAT)
Unit	0.25 r.p.m.
Default value	–

If this variable is mapped in interpolated position mode, the velocity offset will be taken from this value, the master has to supply valid values cyclically via the *IpolDataRecord*.

In other operating modes, this variable does not have an effect.

If this value is not mapped in interpolated position mode, the velocity offset is calculated internally.

12.2.2 Machine Data Interpolated Position Mode

The parameters described in the following can be found in SPP Windows in window “Parameterization” on “Axis Data/Interpolated Position Mode”.

Interpolation Sub Mode Select	Index: 60c0, Short name: IpolSubmodeAuswahl
SPP Windows	Interpolation Submode 0
Type	simple variable, type integer16
Access	read only (via EtherCAT), writing possible but without effects
Var. type	constant (value does not change)
Default value	0 (linear interpolation)

This variable determines the interpolation mode. Only linear interpolation is supported.

Interpolation Time Period	Index: 60c2, Short name: IpolTimePeriod
SPP Windows	Interpolation Time Units 4 10^{idx} s Interpolation Time Index –3
Type	record, 2 elements (subindex 0 .. 2)
Record element 1	interpolation time units (2 .. 10), unsigned8
Record element 2	interpolation time index (–3), integer8
Access	record element 1: read and write record element 2: read only, writing possible but without effects
Var. type	machine data EtherCAT, interpolated position mode

The time interval at which the target positions are transmitted in interpolation position mode are determined via the *Interpolation Time Unit*. The *Interpolation Time Index* determines the unit for record element 1 (–3 for 10^{-3} s = 1 ms).

Interpolation Sync Definition	Index: 60c3, Short name: IpolSyncDefinition
SPP Windows	Synchronize on group 0 ip_sync every n events 1
Type	array, 2 elements (subindex 0 .. 2), unsigned8
Array element 1	synchronize on group
Array element 2	ip sync every n events
Access	read only (via EtherCAT), writing possible but without effects
Var. type	constant (value does not change)

Interpolation Data Configuration	Index: 60c4, Short name: IpolDataConfig
SPP Windows	–
Type	record, 6 elements (subindex 0 .. 6)
Record element 1	max. buffer size, unsigned32; constant 1
Record element 2	actual size, unsigned32; constant 1
Record element 3	buffer organization, unsigned8; constant 0 (FIFO buffer)
Record element 4	buffer position, unsigned16; constant 0
Record element 5	size of data record, unsigned8; constant 1
Record element 6	buffer clear, unsigned8; constant 1
Access	read only (via EtherCAT), writing possible but without effects
Var. type	constant (value does not change)

12.3 Description of Variables Storage Functions

(Currently not supported)

12.4 Description of Variables Monitoring

Abort Connection Option Code	Index: 6007, Kurzname: AbortOptionCode
SPP Windows	–
Type	simple variable, integer16
Access	read and write (via EtherCAT)
Default value	0 = no reaction (default), 1 = creates fault code 8100 (option)

This variable defines the fault reaction on a communication breakdown (sync manager WD timeout).

12.5 Description of Variables Device Information

Device Type	Index: 1000, Short name: DeviceType
SPP Windows	–
Type	simple variable, unsigned32
Access	read only (via EtherCAT)
Variable type	constant (variable value does not change)
Standard value	device profile: CiA 402, drive type: servo drive (00020192_{hex})

The device type can be read out via variable *Device Type*.

According to CANopen communication profile CiA 301 and CANopen device profile CiA 402, the hexadecimal value 20192 of this variable describes the device type and its function:

Bit	Meaning
15	device profile: $192_{hex} = 402_{dec}$ = CiA 402
16 .. 23	drive type: 2 = servo drive
24 .. 31	always 0

Manufacturer Device Name	Index: 1008 Short name: DeviceName
SPP Windows	–
Type	simple variable, visible string, length 16
Access	read only (via EtherCAT)
Variable type	constant (variable value does not change)

The manufacturer name can be read out via variable *Manufacturer Device Name* (e. g. TrioDrive D/ES).

Manufacturer Hardware Version	Index: 1009, Short name: HWVersion
SPP Windows	–
Type	simple variable, visible string, length 16
Access	read only (via EtherCAT)
Variable type	constant (variable value does not change)

The hardware version of the servo drive can be read out via variable *Manufacturer Hardware Version*.

Manufacturer Software Version	Index: 100a, Short name: SWVersion
SPP Windows	–
Type	simple variable, visible string, length 16
Access	read only (via EtherCAT)
Variable type	constant (variable value does not change)

The software version of the servo drive can be read out via variable *Manufacturer Software Version*.

Identity	Index: 1018, Short name: Identity
SPP Windows	–
Type	array, 4 elements (subindex 0 .. 4), type unsigned32
Array element 1	vendor ID (15)
Array element 2	product code (6755 or 6745)
Array element 3	EtherCAT revision number
Array element 4	serial number
Access	read only (via EtherCAT)
Variable type	constant (variable value does not change)

The manufacturer of the servo drive can be read out via variable *Identity*.

Appendix

Appendix A State Machines

State machines describe the behavior of systems. The graphical representation of a state machine is called state diagram.

The elements of a state machine are

- states
- transitions
- events
- actions

A representation of the elements of a state machine in the state diagram is shown in the following figure:

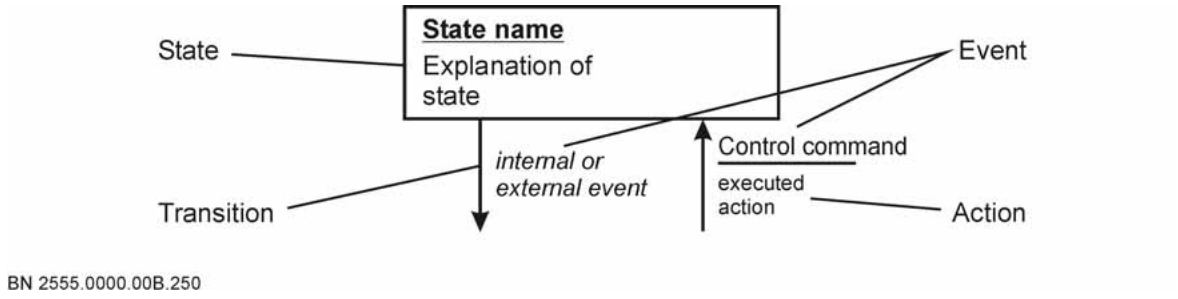

Figure 2: Elements of State Machines

States (shown as a rectangle with the name of the state and further explanations, if required) can be changed by **transitions** (displayed as arrows). A transition is performed when an **event** occurs. In the case of a transition, an **action** is carried out (represented below the event, separated from it by a line). Transmissions for which an action is not carried out are also permitted.

In the application of state machines, the following types of events are differentiated:

- control commands
- internal or external events

Control commands are events the user can trigger e. g. by inducing the master to carry out an ESM service.

Internal or external events (represented in italics) are triggered by the servo drive or a component connected to one of its interfaces. This can be e. g. the switch-on of the control voltage or the termination of the initialization.

Appendix B List of Variables

This index provides an overview of all EtherCAT variables of the servo drive, sorted according to the index.

For explanations regarding the individual columns, please see section 12 (page 25).

Index	Name	Short name	Obj. code elements	Data type length	R/W	Type	Unit
1000	Device Type	DeviceType	Var	u32	R	F	-
1008	Manufacturer Device Name	Device Name	Var	VisStr 16	R	F	-
1009	Manufacturer Hardware Version	HWVersion	Var	VisStr 16	R	F	-
100a	Manufacturer Software Version	SWVersion	Var	VisStr 16	R	F	-
1018	Identity	Identity	Array 4	u32	R	F	-
1600	Receive PDO 1 Mapping	RPDO1Mapping	Record 8	PDOMap	R/W	ME	-
1601	Receive PDO 2 Mapping	RPDO2Mapping	Record 8	PDOMap	R/W	ME	-
1614	Receive PDO 3 Mapping	RPDO3Mapping	Record 8	PDOMap	R/W	ME	-
1615	Receive PDO 4 Mapping	RPDO4Mapping	Record 8	PDOMap	R/W	ME	-
1a00	Transmit PDO 1 Mapping	TPDO1Mapping	Record 8	PDOMap	R/W	ME	-
1a01	Transmit PDO 2 Mapping	TPDO2Mapping	Record 8	PDOMap	R/W	ME	-
1a14	Transmit PDO 3 Mapping	TPDO3Mapping	Record 8	PDOMap	R/W	ME	-
1a15	Transmit PDO 4 Mapping	TPDO4Mapping	Record 8	PDOMap	R/W	ME	-
1c00	Sync Manager Communication Type	SyncManagerCommType	Array 4	u8	R	ME	-
1c12	Sync Manager 2 PDO Assign	SyncMan2PDOAssign	Array 4	u16	R/W	ME	-
1c13	Sync Manager 3 PDO Assign	SyncMan3PDOAssign	Array 4	u16	R/W	ME	-
6007	Abort Connection Option Code	AbortOptionCode	Var	i16	R/W	ME	-
60b1	Velocity Offset	VelocityOffset	Var	i32	R/W	ME	r.p.m.
60c0	Interpolation Sub Mode Select	IpolSubmodeAuswahl	Var	i16	R/W	ME	-
60c1	Interpolation Data Record	IpolDataRecord	Record 1	i32	R/W	ME	PSS
60c2	Interpolation Time Period	IpolTimePeriod	Record 2	IpolTime	R/W	ME	-
60c3	Interpolation Sync Definition	IpolSyncDefinition	Array 2	u8	R/W	ME	-
60c4	Interpolation Data Configuration	IpolDataConfig	Record 6	IpolConf	R/W	ME	-

Appendix C Fault Codes

In case of a fault, the device status (displayed in the axis status word) changes to “Fault”, the corresponding axis fault code is displayed. When the cause of the fault has been removed, the fault status can be reset by a transition of bit “reset fault” from 0 to 1 in the axis control word.

There are two types of faults:

- Faults of the drive which are documented in Operating Instructions 6710.201 “Functions and Parameters” and displayed in drive variable *Axis Fault Code*.

  See operating instructions 6710.201 “Functions and Parameters”, section “Axis Fault Code”.

- Faults of the EtherCAT interface or the EtherCAT controller.

In the following, the possible EtherCAT fault codes (hexadecimal) are listed with fault designation and a short explanation.

A000 EtherCAT State Machine Transition Error

Wrong configuration of the sync manager for input or output process data (e. g. invalid length).

8100 Generic communication error

Communication breakdown

Appendix D Keyword Index

In the keyword index, groups were set up to provide a better overview of certain keywords in addition to the direct entry in alphabetical order.

Variables — The variable description of the corresponding variable can be found via the designation of a variable.

Variables short names — Via the short name of a variable, you can find the variable description of the corresponding variable.

Further groups were set up for an illustration of the connections.