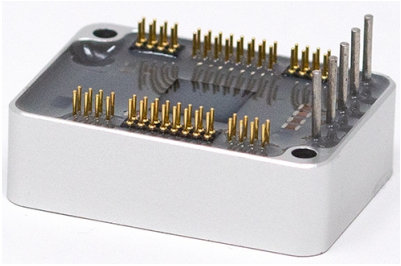
TMCM-1617 CoE Firmware Manual

Firmware Version V1.08 | Document Revision V1.03 • 2021-12-17

The TMCM-1617 is a low-weight miniaturized single axis servo drive for 3-phase BLDC motors with up to 18A RMS motor current and +24V supply. With CAN, RS485, and EtherCAT® it offers various communication options. TMCM-1617 supports incremental encoders and digital hall sensors as position feedback. Customization and different housing options are possible.



EtherCAT®

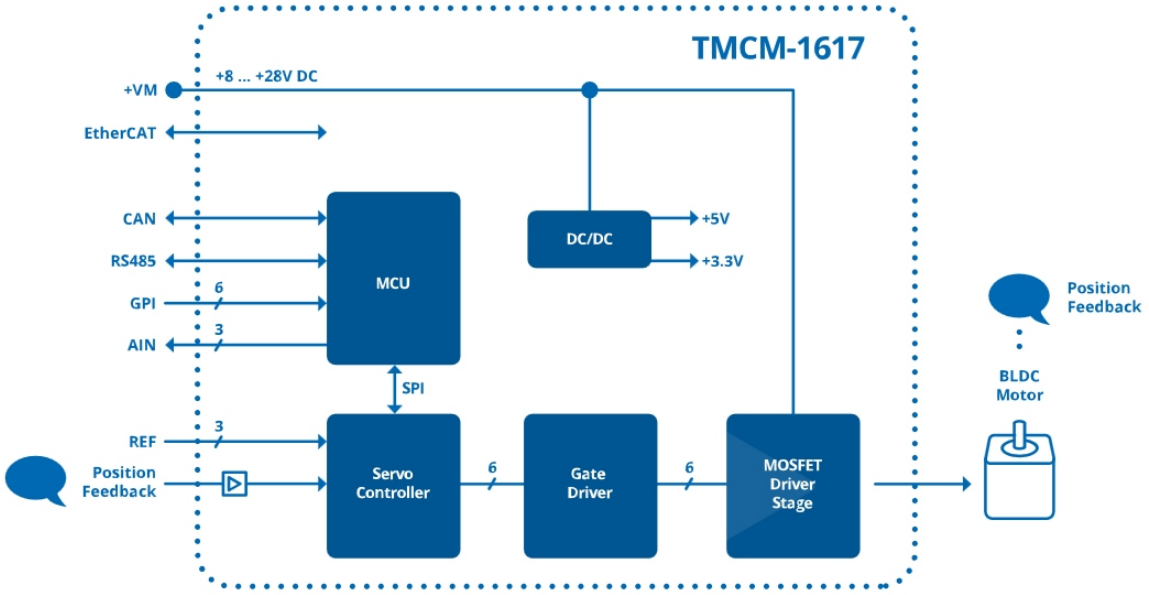
Features

- Servo drive for BLDC motors
- +8…28V DC supply voltage
- Up to 18A RMS max. motor current
- RS485, CAN & EtherCAT® interfaces
- Incremental encoder feedback
- Digital HALL sensor feedback
- Reference switch inputs
- Cooling via aluminum housing
- L/W/H: 36.8mm x 26.8mm x 11.1mm
- Weight: ca. 24g
- Customization options

Applications

- Robotics
- Laboratory Automation
- Manufacturing
- Factory Automation
- Servo Drives
- Low Inductance Motors

Simplified Block Diagram



ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

ADI Trinamic™

Read entire documentation.

Contents

1 Preface

This document specifies objects and modes of operation of the Triamic TMCM-1617 stepper motor control module with CANopen-over-EtherCAT (CoE) firmware. The CoE firmware is designed to fulfill the EtherCAT® version of the CANopen DS402 standards. The EtherCAT® conformance has also been tested. This manual assumes that the reader is already familiar with the basics of EtherCAT® and the CoE protocol (especially DS402).
If necessary it is always possible to turn the module into a TMCL™ or CANopen® module by loading the TMCM-1617 TMCL™ or CANopen® fimware again through the CAN or RS485 interface using the firmware update function of the TMCL-IDE 3.0.

1.1 General Features of this CoE Implementation

Main Characteristics

- Communication according to EtherCAT® standards
- Protocols: CoE, FoE

SDO Communication

- 1 server
- Expedited transfer
- Segmented transfer
- No block transfer

PDO Communication

- Producer
- Consumer
- RPDOs
  - Dynamic mapping with max. 6 mapping entries.
  - Default mappings: manufacturer specific.
- TPDOs
  - Dynamic mapping with max. 9 mapping entries.
  - Default mappings: manufacturer specific.

Sync managers

- Sync manager 0: receive mailbox used for SDO communication
- Sync manager 1: send mailbox used for SDO communication
- Sync manager 2: process data output (used for TPDO)
- Sync manager 3: process data input (used for RPDO)

Further Characteristics

- Emergency: producer

1.2 Abbreviations used in this Manual

Abbreviations	
CAN	Controller area network
CoE	CANopen over EtherCAT
CHGND	chassis ground / earth ground
COB	Communication object
FoE	File transfer over EtherCAT
FSA	Finite state automaton
FSM	Finite state machine
NMT	Network management
ID	Identifier
LSB	Least significant bit
MSB	Most significant bit
PDO	Process data object
PDS	Power drive system
RPDO	Receive process data object
SDO	Service data object
TPDO	Transmit process data object
EMCY	Emergency object
rw	Read and write
ro	Read only
hm	Homing mode
pp	Profile position mode
pv	Profile velocity mode
vm	Velocity mode

Table 1: Abbreviations used in this Manual

1.3 Firmware Update

The software running on the microprocessor consists of two parts, a boot loader and the CoE firmware itself. Whereas the boot loader is installed during production and testing at TRINAMIC and remains untouched throughout the whole lifetime, the CoE firmware can easily be updated by the user. The new firmware can either be loaded into the module via file transfer over EtherCAT (FoE) or via the firmware update function of the TMCL-IDE, using the CAN interface or the RS-485 interface of the module.

2 Communication

2.1 Reference Model

The application layer comprises a concept to configure and communicate real-time-data as well as the mechanisms for synchronization between devices. The functionality which the application layer offers to an application is logically divided over different service data objects (SDO) in the application layer. A service object offers a specific functionality and all the related services.

Applications interact by invoking services of a service object in the application layer. To realize these services this object exchanges data via the EtherCAT with peer service object(s) using a protocol.

The application and the application layer interact with service primitives.

Service Primitives	
Primitive	Definition
Request	Issued by the application to the application layer to request a service.
Indication	Issued by the application layer to the application to report an internal event detected by the application layer or indicate that a service is requested.
Response	Issued by the application to the application layer to respond to a previous received indication.
Confirmation	Issued by the application layer to the application to report the result of a previously issued request.

Table 2: Service Primitives

A service type defines the primitives that are exchanged between the application layer and the cooperating applications for a particular service of a service object. Unconfirmed and confirmed services are collectively called remote services.

Service Types	
Type	Definition
Local service	Involves only the local service object. The application issues a request to its local service object that executes the requested service without communicating with peer service object(s).
Unconfirmed service	Involves one or more peer service objects. The application issues a request to its local service object. This request is transferred to the peer service object(s) that each passes it to their application as an indication. The result is not confirmed back.
Confirmed service	Can involve only one peer service object. The application issues a request to its local service object. This request is transferred to the peer service object that passes it to the other application as an indication. The other application issues a response that is transferred to the originating service object that passes it as a confirmation to the requesting application.
Provider initiated service	Involves only the local service object. The service object (being the service provider) detects an event not solicited by a requested service. This event is then indicated to the application.

Table 3: Service Types

2.2 NMT State Machine

The finite state machine (FSM) or simply state machine is a model of behavior composed of a finite number of states, transitions between those states, and actions. It shows which way the logic runs when certain conditions are met.

Starting and resetting the device is controlled via the state machine. The NMT state machine consists of the states shown in figure 1.

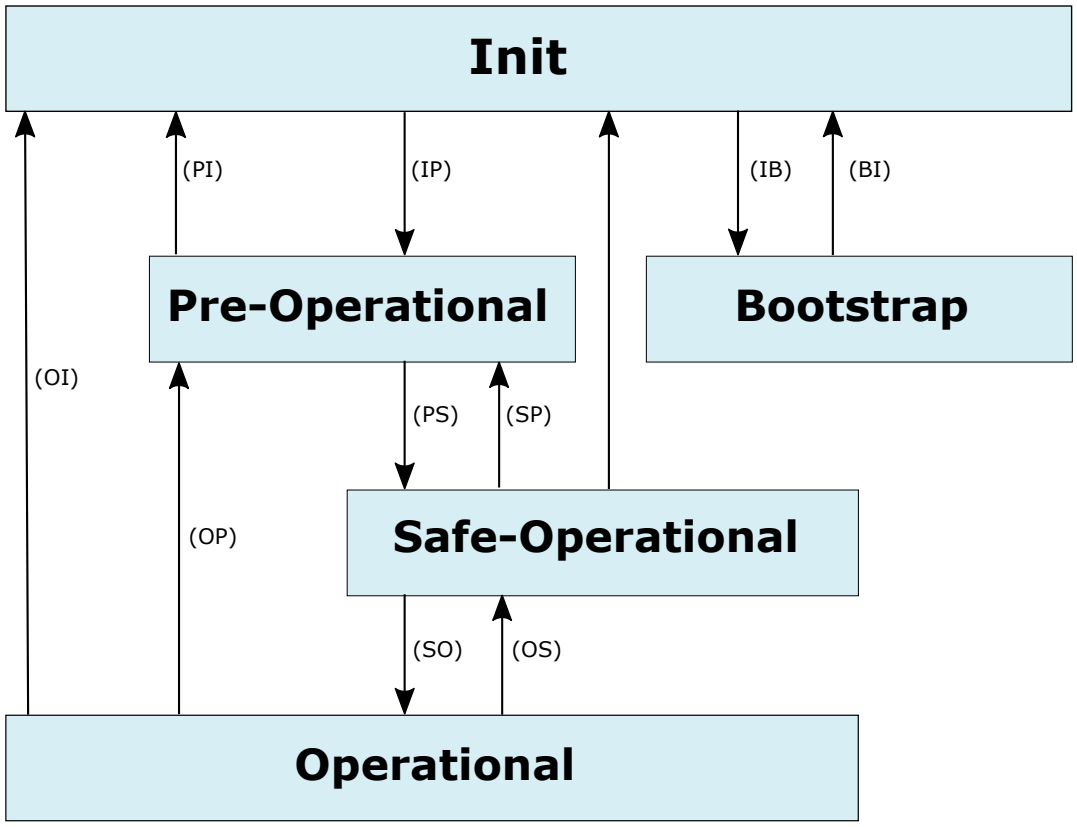


Figure 1: NMT State Machine

After power-on or reset the device enters the Initialization **(INIT)** state.

The master can then switch the device to Pre-Operational **(PRE-OP)** state. In this state, only SDO communication is possible. PDO communication is not possible.

In Safe-Operational **(SAFE-OP)** state, also PDO communication is possible. Inputs can be read, but outputs cannot be switched and the motor cannot be run.

In Operational **(OP)** state, all features of the module can be used. PDO communication is possible, outputs can be switched and the motor can be used. During Operational state the device can use all supported communication objects.

When switching from Operational to Safe-Operational state the motor will be stopped if it has been running. When the EtherCAT connection is lost during Operational state the device will also automatically

switch to Safe-Operational state.

The Bootstrap **(BOOT)** state is used for firmware updates via FoE. Before FoE can be used the device has to be switched to this state.

2.3 Device Model

A CoE device mainly consists of the following parts:

- *Communication:* This function unit provides the communication objects and the appropriate functionality to transport data items via the underlying network structure.
- *Object dictionary:* The object dictionary is a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on this device.
- *Application:* The application comprises the functionality of the device with respect to the interaction with the process environment.

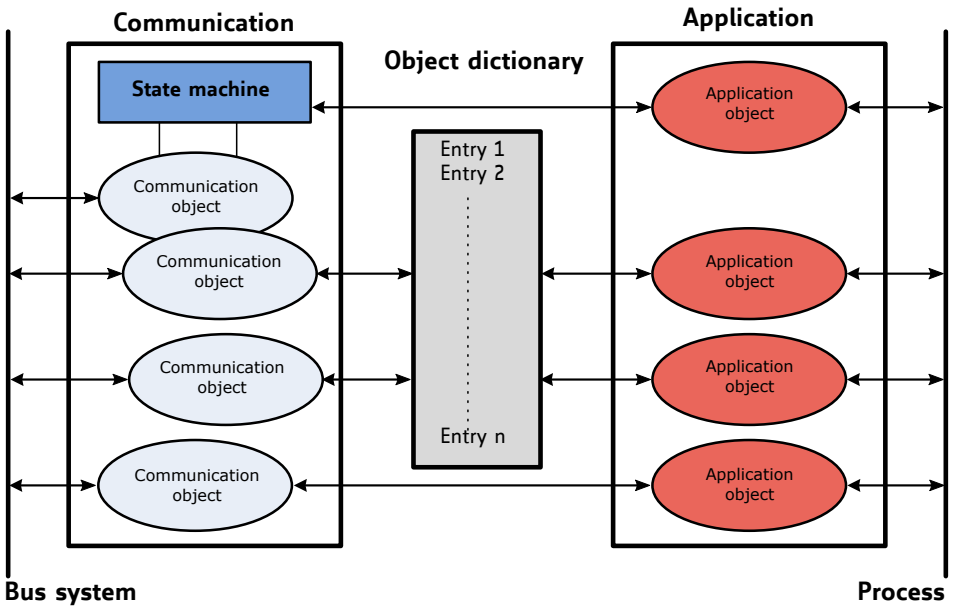


Figure 2: Device Model

2.4 Object Dictionary

The most important part of a device profile is the object dictionary description. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index. The overall layout of the standard object dictionary is shown in table 4:

Object Dictionary	
Index	Object
0000_h	Not used.
0001_h – $001F_h$	Static data types.
0020_h – $003F_h$	Complex data types.
0040_h – $005F_h$	Manufacturer specific complex data types.
0060_h – $007F_h$	Device profile specific static data types.
0080_h – $009F_h$	Device profile specific complex data types.
$00A0_h$ – $0FFF_h$	Reserved for further use.
1000_h – $1FFF_h$	Communication profile area.
2000_h – $5FFF_h$	Manufacturer specific profile area.
6000_h – $9FFF_h$	Standardized device profile area.
$A000_h$ – $BFFF_h$	Standardized interface profile area.
$C000_h$ – $FFFF_h$	Reserved for further use.

Table 4: Object Dictionary

The communication profile area at indices 1000_h through $1FFF_h$ contains the communication specific parameters for the CAN network. These entries are common to all devices.

The manufacturer segment at indices 2000_h through $5FFF_h$ contains manufacturer specific objects. These objects control the special features of the Trinamic TMCM-1617 motion control device.

The standardized device profile area at indices 6000_h through $9FFF_h$ contains all data objects common to a class of devices that can be read or written via the network. They describe the device parameters and the device functionality of the device profile.

3 Communication Area

The communication area contains all objects that define the communication parameters of the CoE device according to the EtherCAT standard.

3.1 Detailed Object Specifications

3.1.1 Object 1000_h: Device Type

This object contains information about the device type. The object 1000_h describes the type of device and its functionality. It is composed of a 16-bit field which describes the device profile that is used and a second 16-bit field which provides additional information about optional functionality of the device.

Object Description			
Index	Name	Object Type	Data Type
1000_h	Device type	Variable	UNSIGNED32

Table 5: Object Description (1000_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	$FFFC0192_h$

Table 6: Entry Description (1000_h)

3.1.2 Object 1001_h: Error Register

This object contains error information. The CANopen device maps internal errors into object 1001_h. It is part of an emergency object.

Object Description			
Index	Name	Object Type	Data Type
1001_h	Error register	Variable	UNSIGNED8

Table 7: Object Description (1001_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED8	0

Table 8: Entry Description (1001_h)

Error Register Bits	
Bit	Definition
0	Generic error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device profile specific
6	Reserved (always 0)
7	Manufacturer specific

Table 9: Error Register Bits

3.1.3 Object 1008_h: Manufacturer Device Name

This object contains the name of the device as given by the manufacturer.

Object Description			
Index	Name	Object Type	Data Type
1008_h	Manufacturer Device Name	Variable	Visible String

Table 10: Object Description (1008_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	TMCM-1617

Table 11: Entry Description (1008_h)

3.1.4 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

Object Description			
Index	Name	Object Type	Data Type
1009_h	Manufacturer Hardware Version	Variable	Visible String

Table 12: Object Description (1009_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on device, e.g. 1.00

Table 13: Entry Description (1009_h)

3.1.5 Object $100A_h$: Manufacturer Software Version

This object contains the software version description.

Object Description			
Index	Name	Object Type	Data Type
$100A_h$	Manufacturer Software Version	Variable	Visible String

Table 14: Object Description ($100A_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	—	Depends on device, e.g. 1.00.

Table 15: Entry Description ($100A_h$)

3.1.6 Object 1010_h: Store Parameters

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its saving capabilities.

The TMCM-1617 module supports saving of the following parameter groups:

- Sub-index 1: save all parameters.
- Sub-index 2: save communication parameters.
- Sub-index 3: save device profile parameters (not used).
- Sub-index 4: save motor 0 parameters.
- Sub-index 5: save device parameters (other non axis-related parameters).

Note In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-Index. This signature is "save" (65766173_h, see also table 16).

Save Signature			
e	v	a	s
65_h	76_h	61_h	73_h

Table 16: Save Signature

On reception of the correct signature in the appropriate sub-index the device stores the parameter and then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an abort SDO transfer (abort code: 06060000_h). If a wrong signature is written, the device refuses to store and responds with abort SDO transfer (abort code: $0800002x_h$).

On read access, each sub-index provides information if it is possible to store the parameter group. It reads 1 if yes and 0 if no.

Object Description			
Index	Name	Object Type	Data Type
1010_h	Store Parameters	Array	UNSIGNED32

Table 17: Object Description (1010_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest supported sub-index	ro	no	UNSIGNED8	5
1	Save all parameters	rw	no	UNSIGNED32	1
2	Save communication parameters	rw	no	UNSIGNED32	1
3	Save device profile parameters	rw	no	UNSIGNED32	0
4	Save motor 0 parameters	rw	no	UNSIGNED32	1
5	Save device parameters	rw	no	UNSIGNED32	1

Table 18: Entry Description (1010_h)

3.1.7 Object 1011_h: Restore Parameters

With this object the default values of parameters according to the communication or device profile are restored. By read access the device provides information about its capabilities to restore these values.

The TMCM-1617 module supports restoring of the following parameter groups:

- Sub-index 1: restore all parameters.
- Sub-index 2: restore communication parameters.
- Sub-index 3: restore device profile parameters (not used).
- Sub-index 4: restore motor 0 parameters.
- Sub-index 5: restore device parameters (other non axis-related parameters).

Note In order to avoid restoring the parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-Index. This signature is "load" ($64616F6C_h$, see also table 19).

Load Signature			
d	a	o	l
64_h	61_h	$6F_h$	$6C_h$

Table 19: Load Signature

On reception of the correct signature in the appropriate sub-index the device restores the parameter and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an abort SDO transfer (abort code: 06060000_h). If a wrong signature is written, the device refuses to restore and responds with abort SDO transfer (abort code: $0800002x_h$).

On read access, each sub-index provides information if it is possible to restore the parameter group. It reads 1 if yes and 0 if no.

After the default values have been restored they will become active after the next rest or power cycle of the TMCM-1617.

Object Description			
Index	Name	Object Type	Data Type
1011_h	Restore parameters	Array	UNSIGNED32

Table 20: Object Description (1011_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest supported sub-index	ro	no	UNSIGNED8	5
1	Restore all parameters	rw	no	UNSIGNED32	1
2	Restore communication parameters	rw	no	UNSIGNED32	1
3	Restore device profile parameters	rw	no	UNSIGNED32	0
4	Restore motor 0 parameters	rw	no	UNSIGNED32	1
5	Restore device parameters	rw	no	UNSIGNED32	1

Table 21: Entry Description (1011_h)

3.1.8 Object 1018_h: Identity Object

The object 1018_h contains general information about the device:

- The vendor ID (sub-index 01_h) contains a unique value allocated to each manufacturer. The vendor ID of Trinamic is 286_h.
- The manufacturer specific product code (sub-index 2_h) identifies a specific device version.
- The manufacturer specific revision number (sub-index 3_h) consists of a major revision number and a minor revision number.

Object Description			
Index	Name	Object Type	Data Type
1018_h	Identity object	Record	Identity

Table 22: Object Description (1018_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Number of entries	ro	no	0…3	3
1	Vendor ID	ro	no	UNSIGNED32	0286_h
2	Product code	ro	no	UNSIGNED32	1617
3	Revision number	ro	no	UNSIGNED32	e.g. 20003_h for version 2.3

Table 23: Entry Description (1018_h)

3.1.9 Object 1600_h: Receive PDO Mapping Parameter

This object contains the mapping parameters for the RPDO the device is able to receive. The sub-index 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be received with the corresponding RPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

Object Description			
Index	Name	Object Type	Data Type
1600_h	Receive PDO mapping parameter	RECORD	PDO Mapping

Table 24: Object Description (1600_h)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of mapped application objects in PDO	rw	0…9	Index 1600_h: 4
01_h	Mapping entry 1	rw	UNSIGNED32	60400010_h
02_h	Mapping entry 3	rw	UNSIGNED32	$607A0020_h$
03_h	Mapping entry 4	rw	UNSIGNED32	60710010_h
04_h	Mapping entry 5	rw	UNSIGNED32	$60FF0020_h$
05_h	Mapping entry 2	rw	UNSIGNED32	0_h
06_h	Mapping entry 6	rw	UNSIGNED32	0_h
07_h	Mapping entry 7	rw	UNSIGNED32	0_h
08_h	Mapping entry 8	rw	UNSIGNED32	0_h
09_h	Mapping entry 9	rw	UNSIGNED32	0_h

Table 25: Entry Description (1600_h)

3.1.10 Objects $1A00_h$: Transmit PDO Mapping Parameter

This object contains the mapping parameters for the TPDO the device is able to transmit. The sub-index 00_h contains the number of valid entries within the mapping record. This number of entries is also the number of the application variables which shall be transmitted with the corresponding TPDO. The sub-indices from 01_h to the number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length.

Object Description			
Index	Name	Object Type	Data Type
$1A00_h$	Transmit PDO mapping parameter	RECORD	PDO Mapping

Table 26: Object Description ($1A00_h$)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of mapped aapli-cation objects in PDO	rw	0...9	6
01_h	Mapping entry 1	rw	UNSIGNED32	60410010_h
02_h	Mapping entry 2	rw	UNSIGNED32	60610008_h
03_h	Mapping entry 3	rw	UNSIGNED32	60640020_h
04_h	Mapping entry 4	rw	UNSIGNED32	60770010_h
05_h	Mapping entry 5	rw	UNSIGNED32	$606C0020_h$
06_h	Mapping entry 6	rw	UNSIGNED32	$60FD0020_h$
07_h	Mapping entry 7	rw	UNSIGNED32	0_h
08_h	Mapping entry 8	rw	UNSIGNED32	0_h
09_h	Mapping entry 9	rw	UNSIGNED32	0_h

Table 27: Entry Description ($1A00_h$)

3.1.11 Objects $1C00_h$: Sync Manager Communication Type

This object describes the communication types of the EtherCAT sync managers. The types of the first four synch managers are normally fixed and should not be changed. Sync managers can have the following for communication types:

Sync Manager Communication Types	
Type	Description
1	Mailbox receive
2	Mailbox send
3	Process data input
4	Process data output

Table 28: Sync Manager Communication Types

Object Description			
Index	Name	Object Type	Data Type
$1C00_h$	Sync manager communication type	RECORD	UNSIGNED8

Table 29: Object Description ($1C00_h$)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of entries	rw	0...3	4
01_h	Communication type sync manager 1	rw	UNSIGNED8	1
02_h	Communication type sync manager 2	rw	UNSIGNED8	2
03_h	Communictaion type sync manager 3	rw	UNSIGNED8	3
04_h	Communictaion type sync manager 4	rw	UNSIGNED8	4

Table 30: Entry Description ($1C00_h$)

3.1.12 Objects $1C12_h$: Sync Manager 2 PDO Assignment

This object contains the index of the PDO definiton object that is assigned to sync manager 2. Normally, the RPDO objects are assigned to sync manager 2. Under most cicumstances there is no need to change this setting.

Object Description			
Index	Name	Object Type	Data Type
$1C12_h$	Sync manager 2 PDO assignment	RECORD	PDO assignment

Table 31: Object Description ($1C12_h$)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of assigned PDOs	rw	0...1	1
01_h	PDO mapping index of assigned RPDO	rw	UNSIGNED16	1600_h

Table 32: Entry Description ($1C12_h$)

3.1.13 Objects $1C13_h$: Sync Manager 3 PDO Assignment

This object contains the index of the PDO definiton object that is assigned to sync manager 3. Normally, the TPDO objects are assigned to sync manager 3. Under most cicumstances there is no need to change this setting.

Object Description			
Index	Name	Object Type	Data Type
$1C13_h$	Sync manager 3 PDO assignment	RECORD	PDO assignment

Table 33: Object Description ($1C13_h$)

Entry Description				
Sub-index	Description	Access	Value Range	Default Value
00_h	Number of assigned PDOs	rw	0…1	1
01_h	PDO mapping index of assigned TPDO	rw	UNSIGNED16	$1A00_h$

Table 34: Entry Description ($1C13_h$)

4 Manufacturer specific Area

The manufacturer segment contains manufacturer specific objects. These objects control the special features of the Trinamic Motion Control device TMCM-1617.

4.1 Detailed Object Specifications

4.1.1 Object 2000_h: Device Info

This object provides version information about the motor controller chip used on this module.

Object Description			
Index	Name	Object Type	Data Type
2000_h	Device Info	Variable	Record

Table 35: Object Description (2000_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	MC_Type	no	0	4294967295	0	—	ro
2	MC_Version	no	0	4294967295	0	—	ro
3	MC_Date	no	0	4294967295	0	—	ro
4	MC_Time	no	0	4294967295	0	—	ro
5	MC_Variant	no	0	4294967295	0	—	ro

Table 36: Entry Description (2000_h)

4.1.2 Object 2003_h: Maximum Current

This objects limits the maximum current that is used to drive the motor. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
2003_h	Maximum Current	Variable	UNSIGNED32

Table 37: Object Description (2003_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…18000	0

Table 38: Entry Description (2003_h)

4.1.3 Object 2004_h: Open Loop Current

This object controls the motor current used in open loop mode. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
2004_h	Open Loop Current	Variable	UNSIGNED32

Table 39: Object Description (2004_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…18000	0

Table 40: Entry Description (2004_h)

4.1.4 Object 2005_h: Switch Parameters

This object defines which limit switches are to be used. Bit 0 stands for the left and bit 1 stands for the right limit switch. If a bit is set, the corresponding limit switch will not be used. So this object has to be set to the value 3 if limit switches are not connected. The object can only be written when the drive is in the SWITCHED_ON_DISABLED state (but is always readable).

The limit switches can also be inverted using bit 2 and bit 3:

- Bit 2 inverts the left limit switch
- Bit 3 inverts the right limit switch

The polarity of the home switch can be set using bit 5.

Object Description			
Index	Name	Object Type	Data Type
2005_h	Limit switches	Variable	UNSIGNED32

Table 41: Object Description (2005_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…63	0

Table 42: Entry Description (2005_h)

Bit Definitions	
Bit	Definition
0	Left limit switch deactivated if set.
1	Right limit switch deactivated if set.
2	Left limit switch inverted if set.
3	Right limit switch inverted if set.
4	Home switch deactivated if set.
5	Home switch inverted if set.

Table 43: Bit Definitions (2005_h)

4.1.5 Object 2006_h: Brake Chopper

With this object the behaviour of the brake chopper output can be set up.

Object Description			
Index	Name	Object Type	Data Type
2006_h	Device Info	Variable	Record

Table 44: Object Description (2006_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Enable	no	0	1	0	—	RW
2	Voltage	no	60	300	300	0.1 V	RW
3	Hysteresis	no	0	50	5	0.1 V	RW

Table 45: Entry Description (2006_h)

4.1.6 Object 2041_h: Torque Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2041_h	Torque Mode Settings	Variable	Record

Table 46: Object Description (2041_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Torque_P	no	0	65535	300		RW
2	Torque_I	no	0	65535	300		RW
3	PI_Torque_Error	no	-2147483648	2147483647	0	[mA]	R
4	PI_Torque_Error_Sum	no	-2147483648	2147483647	0		R
5	PI_Flux_Error	no	-2147483648	2147483647	0	[mA]	R
6	PI_Flux_Error_Sum	no	-2147483648	2147483647	0		R
7	PHI_E	no	-32678	32767	0		R

Table 47: Entry Description (2041_h)

Torque_P P parameter for the torque PI controller.

Torque_I I parameter for the torque PI controller.

4.1.7 Object 2042_h: Velocity Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2042_h	Velocity Mode Settings	Variable	Record

Table 48: Object Description (2042_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Velocity_P	no	0	65535	300		RW
2	Velocity_I	no	0	65535	100		RW
3	PI_Velocity_Error	no	-2147483648	2147483647	0		R
4	PI_Velocity_Error_Sum	no	-2147483648	2147483647	0		R
5	Sensor_Selection	no	0	1	0		RW
6	Velocity_Filter	no	0	8	3		RW
7	Velocity_Unit	no	0	1	0		RW
8	Motor_Halted_Velocity	no	0	200000	10		RW

Table 49: Entry Description (2042_h)

Velocity_P — P parameter for the velocity PI controller.

Velocity_I — I parameter for the velocity PI controller.

Sensor_Selection — Select a commutation mode that fits best to your motor's sensors.

- 0: Same as commutation
- 1: ABN encoder

Velocity_Filter — Moving average filter.

Velocity_Unit — Select mechanical or electrical velocity unit.

- 0: Mechanical rpm
- 1: Electrical rpm

Motor_Halted_Velocity — If the actual velocity is below this value the motor halted flag will beset.

4.1.8 Object 2043_h: Position Mode Settings

Object Description			
Index	Name	Object Type	Data Type
2043_h	Position Mode Settings	Variable	Record

Table 50: Object Description (2043_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Position_P	no	0	65535	50		RW
2	PI_Position_Error	no	-2147483648	2147483647	0		R
3	Sensor_Selection	no	0	1	0		RW

Table 51: Entry Description (2043_h)

Position_P P parameter for the velocity PI controller.

Sensor_Selection Select a commutation mode that fits best to your motor's sensors.

- 0: Same as commutation
- 1: ABN encoder

4.1.9 Object 2050_h: Motor Type

With this object the motor type connected to the module can be set. The following settings are possible:

- Mode 0: no motor
- Mode 1: single phase DC motor
- Mode 3: three phase BLDC motor

Object Description			
Index	Name	Object Type	Data Type
2050_h	Limits	Variable	UNSIGNED8

Table 52: Object Description (2050_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0...3	3

Table 53: Entry Description (2050_h)

4.1.10 Object 2055_h: Commutation Mode

Select a commutation mode that fits best to your motor's sensors.

Commutation Modes	
0	FOC — disabled
1	FOC — open loop
2	FOC — digital hall
3	FOC — ABN encoder

Table 54: Commutation Modes

Object Description			
Index	Name	Object Type	Data Type
2055_h	Commutation Mode	Variable	UNSIGNED8

Table 55: Object Description (2055_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Commutation Mode	no	0	1	3		RW

Table 56: Entry Description (2055_h)

4.1.11 Object 2056_h: Motor Pole Pairs

Set this object to the number of pole pairs your motor is equipped with.

Object Description			
Index	Name	Object Type	Data Type
2056_h	Motor Pole Pairs	Variable	UNSIGNED8

Table 57: Object Description (2056_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Motor Pole Pairs	no	1	12	4		RW

Table 58: Entry Description (2056_h)

4.1.12 Object 2057_h: Motor Shaft Direction

Using this object the motor shaft direction can be reversed. Set it to 0 (default value) for normal shaft direction or 1 for reversed shaft direction.

Object Description			
Index	Name	Object Type	Data Type
2057_h	Motor Shaft Direction	Variable	UNSIGNED8

Table 59: Object Description (2057_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Motor Shaft Direction	no	0	1	0		RW

Table 60: Entry Description (2057_h)

4.1.13 Object 2058_h: Position Scaler

Using this object all position values can be scaled. It defines the number of steps per mechanical rotation. With its default value of 65536, a move of 65536 steps leads to one mechanical rotation.

Object Description			
Index	Name	Object Type	Data Type
2058_h	Position Scaler	Variable	SIGNED32

Table 61: Object Description (2058_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Position Scaler	no	-2147483648	2147483647	65536		RW

Table 62: Entry Description (2058_h)

4.1.14 Object 2060_h: ADC Configuration

Using this object the ADC offsets for the coil current measurement can be configured. This is necessary for each new motor type.

Object Description			
Index	Name	Object Type	Data Type
2060_h	ADC Configuration	Variable	Record

Table 63: Object Description (2060_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	ADC_I0_Raw	no	0	65535	0		RO
2	ADC_I1_Raw	no	0	65535	0		RO
3	ADC_I0_Offset	no	0	65535	33500		RW
4	ADC_I1_Offset	no	0	65535	33500		RW
5	ADC_I0	no	-32768	32767	0		RO
6	ADC_I1	no	-32768	32767	0		RO
7	ADC_I2	no	-32768	32767	0		RO

Table 64: Entry Description (2060_h)

4.1.15 Object 2070_h: Hall Sensor Settings

This object sets various parameters of the hall sensors. If the motor is equipped with hall sensors then set the necessary parameters here.

Object Description			
Index	Name	Object Type	Data Type
2070_h	Hall Sensor Settings	Variable	Record

Table 65: Object Description (2070_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Polarity	no	0	1	0		RW
2	Direction	no	0	1	0		RW
3	Interpolation	no	0	1	1		RW
4	PHI_E_Offset	no	-32768	32767	0		RW

Table 66: Entry Description (2070_h)

Polarity Hall sensor polarity.

- 0: Standard
- 1: Inverted

Direction Hall sensor direction.

- 0: Standard
- 1: Inverted

Interpolation Hall sensor interpolation.

- 0: On
- 1: Off

PHI_E_Offset Offset for electrical angle hall_phi_e of hall sensor.

4.1.16 Object 2080_h: ABN Encoder Settings

Using this object all necessary encoder parameters can be set. Check and set these parameters if your motor is equipped with an encoder. It is then also possible to choose between different encoder initialization modes.

Encoder Initialization Modes	
0	Estimate offset
1	Use offset
2	Use hall

Table 67: Encoder Initialization Modes

Object Description			
Index	Name	Object Type	Data Type
2080_h	ABN Encoder Settings	Variable	Record

Table 68: Object Description (2080_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Direction	no	0	1	0		RW
2	Steps	no	0	16777215	4096		RW
3	Init_Mode	no	0	2	0		RW

Table 69: Entry Description (2080_h)

4.1.17 Object 2101_h: Motor Status Flags

This object provides motor status and error flags. This can be a combination of the bits described in table 70.

Motor Status Flags		
Bit	Name	Meaning
0	Overcurrent	Too high current detected.
1	Undervoltage	Supply voltage too low.
2	Overvoltage	Supply voltage too high.
3	Overtemperature	Maximum driver temperature exceeded.
4	Motor halted	Motor stopped.
5	Hall error	Hall sensor error.
6	Driver error	Motor driver error.
7	Init error	Motor initialization error.
8	Stop mode	Motor in stop mode.
9	Velocity mode	Motor operating in velocity mode.
10	Position mode	Motor operating in position mode.
11	Torque mode	Motor operating in torque mode.
12	Emergency stop	Emergency stop active.
14	Position end	Target position reached.
15	Module initialized	Module initialization complete.
17	IIT exceeded	IIT limit exceeded.
18	Brake active	Brake output active.

Table 70: Motor Status Flags (2101_h)

Object Description			
Index	Name	Object Type	Data Type
2101_h	Device State	Variable	UNSIGNED32

Table 71: Object Description (2101_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	$0 \ldots 3FFFF_h$	0

Table 72: Entry Description (2101_h)

4.1.18 Object 2102_h: Open Loop Commutation Angle

This object shows the open loop commutation angle. It is mainly used by the Trinamic motor tuning tools.

Object Description			
Index	Name	Object Type	Data Type
2102_h	Open Loop Commutation Angle	Variable	SIGNED16

Table 73: Object Description (2102_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Open Loop Commutation Angle	no	-32768	32767	0		RO

Table 74: Entry Description (2102_h)

4.1.19 Object 2103_h: Encoder Commutation Angle

This object shows the encoder commutation angle. It is mainly used by the Trinamic motor tuning tools.

Object Description			
Index	Name	Object Type	Data Type
2103_h	Encoder Commutation Angle	Variable	SIGNED16

Table 75: Object Description (2103_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Encoder Commutation Angle	no	-32768	32767	0		RO

Table 76: Entry Description (2103_h)

4.1.20 Object 2104_h: Hall Commutation Angle

This object shows the hall sensor commutation angle. It is mainly used by the Trinamic motor tuning tools.

Object Description			
Index	Name	Object Type	Data Type
2104_h	Hall Commutation Angle	Variable	SIGNED16

Table 77: Object Description (2104_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Hall Commutation Angle	no	-32768	32767	0		RO

Table 78: Entry Description (2104_h)

4.1.21 Object 2140_h: Home Offset Display

This object shows the home offset. The value is given in encoder or hall increments.

Object Description			
Index	Name	Object Type	Data Type
2140_h	Home Offset Display	Variable	SIGNED32

Table 79: Object Description (2140_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Home Offset Display	no	-2147483648	2147483647	0		R

Table 80: Entry Description (2140_h)

4.1.22 Object 2702_h: Device Digital Inputs

Bit Definitions	
Bit	Description
16	REF_R
17	REF_L
18	REF_H
19	GPIO0
20	GPIO1

Table 81: Bit Definitions (2702_h)

Object Description			
Index	Name	Object Type	Data Type
2702_h	Device Digital Inputs	Variable	UNSIGNED32

Table 82: Object Description (2702_h)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
0	Digital Inputs	no	0	3	0		R

Table 83: Entry Description (2702_h)

4.1.23 Object 2703_h: Device Digital Outputs

With this object the digital outputs (general purpose outputs) can be set. Bits 0...3 of sub index 1 switch the outputs of the module. Bits 0...3 of sub index 2 determine which outputs can be switched. The number of available digital outputs depends on the module type.

Bit Definitions	
Bit	Description
16	GPO2
17	GPO3
18	GPO4
19	GPO5

Table 84: Bit Definitions (2703_h)

Object Description			
Index	Name	Object Type	Data Type
2703_h	Device Digital Outputs	Variable	ARRAY

Table 85: Object Description (2703_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Physical Outputs	rw	yes	UNSIGNED32	0
2	Output Mask	rw	yes	UNSIGNED32	0

Table 86: Entry Description (2703_h)

4.1.24 Object $270E_h$: Device Analog Inputs

Object Description			
Index	Name	Object Type	Data Type
$270E_h$	Analog Inputs	Variable	Record

Table 87: Object Description ($270E_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	AIN0	no	0	4095	0		R
2	AIN1	no	0	4095	0		R
3	ADC2	no	0	4095	0		R
4	ADC3	no	0	4095	0		R
5	ADC4	no	0	4095	0		R
6	ADC5	no	0	4095	0		R
7	ADC6	no	0	4095	0		R
8	ADC7	no	0	4095	0		R

Table 88: Entry Description ($270E_h$)

ADC value of the analog input pins AIN0 to AIN2, as well as ADC values of some on-board voltages.

AIN0	Analog input pin AIN0 ADC value.
AIN1	Analog input pin AIN1 ADC value.
AIN2	Analog input pin AIN2 ADC value.
AIN3	ADC value of the on-board logic supply voltage measurement circuit.
AIN4	ADC value of the on-board 5V supply voltage measurement circuit.
AIN5	Driver ADC value with offset.
AIN6	Driver ADC value without offset.
AIN7	No voltage source connected.

5 Profile Specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the TMCM-1617. Since it is not possible to operate the modes in parallel, the user is able to activate the required function by selecting a mode of operation. The control device writes to the modes of operation object in order to select the operation mode. The drive device provides the modes of operation display object to indicate the actual activated operation mode. Controlword, statusword, and set-points are used mode-specific. This implies the responsibility of the control device to avoid inconsistencies and erroneous behavior.
The following operating modes (selectable via object 6060_h, please see 5.1.6) are implemented on the TMCM-1617:

- Profile position mode (pp)
- Profile velocity mode (pv)
- Homing mode (hm)
- Cyclic position mode (csp)
- Cyclic velocity mode (csv)
- Cyclic torque mode (cst)

5.1 Detailed Object Specifications

5.1.1 Object $605A_h$: Quick Stop Option Code

This object indicates what action is performed when the quick stop function is executed. The slow down ramp is the deceleration value of the used mode of operation. The following quick stop option codes are supported in the current version of the CANopen firmware:

Value Definition	
Value	Definition
1	Slow down on *slow down ramp* and transit into *switch on disabled*
2	Slow down on *quick stop ramp* and transit into *switch on disabled*
5	Slow down on *slow down ramp* and stay in *quick stop active*)
6	Slow down on *quick stop ramp* and stay in *quick stop active*

Table 89: Value Description ($605A_h$)

Object Description			
Index	Name	Object Type	Data Type
$605A_h$	Quick stop option code	Variable	SIGNED16

Table 90: Object Description ($605A_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1/2/5/6	2

Table 91: Entry Description ($605A_h$)

5.1.2 Object $605B_h$: Shutdown Option Code

This object indicates what action is performed if there is a transition from *operation enabled* state to *ready to switch on state*. The shutdown option code always has the value 0 as only this is supported.

Value Definition	
Value	Definition
0	Disable drive function (switch off the power stage)

Table 92: Value Description ($605B_h$)

Object Description			
Index	Name	Object Type	Data Type
$605B_h$	Shutdown option code	Variable	UNSIGNED16

Table 93: Object Description ($605B_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0	0

Table 94: Entry Description ($605B_h$)

5.1.3 Object $605C_h$: Disable Operation Option Code

This object indicates what action is performed if there is a transition from *operation enabled* state to *switched on* state. The disable operation option code always has the value 1 as only this is supported. The slow down ramp is the deceleration value of the used mode of operation.

Value Definition	
Value	Definition
1	Slow down on slow down ramp

Table 95: Value Description ($605C_h$)

Object Description			
Index	Name	Object Type	Data Type
$605C_h$	Disable operation option code	Variable	UNSIGNED16

Table 96: Object Description ($605C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1	1

Table 97: Entry Description ($605C_h$)

5.1.4 Object $605D_h$: Halt Option Code

This object indicates what action is performed when the halt function is executed. The slow down ramp is the deceleration value of the used mode of operation.

Value Definition	
Value	Definition
1	Slow down on slow down ramp and stay in *operation enabled*

Table 98: Value Description ($605D_h$)

Object Description			
Index	Name	Object Type	Data Type
$605D_h$	Halt option code	Variable	UNSIGNED16

Table 99: Object Description ($605D_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	1	1

Table 100: Entry Description ($605D_h$)

5.1.5 Object $605E_h$: Fault Reaction Option Code

This object indicates what action is performed when fault is detected in the power drive system. The slow down ramp is the deceleration value of the used mode of operation. The fault reaction option code always has the value 2 as only this is supported.

Value Definition	
Value	Definition
2	Slow down on quick stop ramp

Table 101: Value Description ($605E_h$)

Object Description			
Index	Name	Object Type	Data Type
$605E_h$	Fault reaction option code	Variable	UNSIGNED16

Table 102: Object Description ($605E_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	2	2

Table 103: Entry Description ($605E_h$)

5.1.6 Object 6060_h: Modes of Operation

This object indicates the requested operation mode. Supported operating modes are:

Value Definition	
Value	Mode
0	No mode
1	Profile position mode (pp)
3	Profile velocity mode (pv)
6	Homing mode (hm)
8	Cyclic synchronous position mode (csp)
9	Cyclic synchronous velocity mode (csv)
10	Cyclic synchronous torque mode (cst)

Table 104: Value Description (6060_h)

The motor will not run when the operating mode is set to 0. It will be stopped when the motor is running in one of the supported operating modes and the operating mode is then switched to 0.

Object Description			
Index	Name	Object Type	Data Type
6060_h	Modes of operation	Variable	SIGNED8

Table 105: Object Description (6060_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 104	0

Table 106: Entry Description (6060_h)

5.1.7 Object 6061_h: Modes of Operation Display

This object shows the operating mode that is currently set.

Value Definition	
Value	Mode
0	No mode
1	Profile position mode (pp)
3	Profile velocity mode (pv)
6	Homing mode (hm)
8	Cyclic synchronous position mode (csp)
9	Cyclic synchronous velocity mode (csv)
10	Cyclic synchronous torque mode (cst)

Table 107: Value Description (6061_h)

The motor will not run when the operating mode is set to 0. It will be stopped when the motor is running in one of the supported operating modes and the operating mode is then switched to 0.

Object Description			
Index	Name	Object Type	Data Type
6061_h	Modes of operation display	Variable	SIGNED8

Table 108: Object Description (6061_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	see table 107	0

Table 109: Entry Description (6061_h)

5.1.8 Object $60FD_h$: Digital Inputs

This object contains the states of the digital inputs of the module. Starting from bit 0, every bit reflects the state of one digital input. The number of valid bits depends on the number of digital inputs on the module used.

Object Description			
Index	Name	Object Type	Data Type
$60FD_h$	Digital inputs	Variable	UNSIGNED32

Table 110: Object Description ($60FD_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	UNSIGNED32	0

Table 111: Entry Description ($60FD_h$)

5.1.9 Object 6502_h: Supported Drive Modes

This object provides information on the supported drive modes. A bit that is set means that the mode is supported, a bit that is not set means that the mode is not supported by the drive.

Value Definition	
Bit	Mode
0	Profile position mode (pp)
1	Velocity mode (vl)
2	Profile velocity mode (pv)
3	Torque mode (tq)
4	Reserved
5	Homing mode (hm)
6	Interpolated position mode (ip)
7	Cyclic synchronous position mode (csp)
8	Cyclic synchronous velocity mode (csv)
9	Cyclic synchronous torque mode (cst)

Table 112: Value Definition (6502_h)

Object Description			
Index	Name	Object Type	Data Type
6502_h	Supported drive modes	Variable	UNSIGNED32

Table 113: Object Description (6502_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	Depends on supported modes.

Table 114: Entry Description (6502_h)

6 Profile Position Mode

A target position is applied to the trajectory generator. It is generating a position demand value for the position control loop described in the position control function.

Please refer to object 6060_h (section 5.1.6) for information about how to choose an operation mode. Object 6061_h (section 5.1.7) shows the operation mode that is set.

6.1 Detailed Object Specifications

The following text offers detailed object specifications. For a better understanding, it is necessary to see how the state machine works.

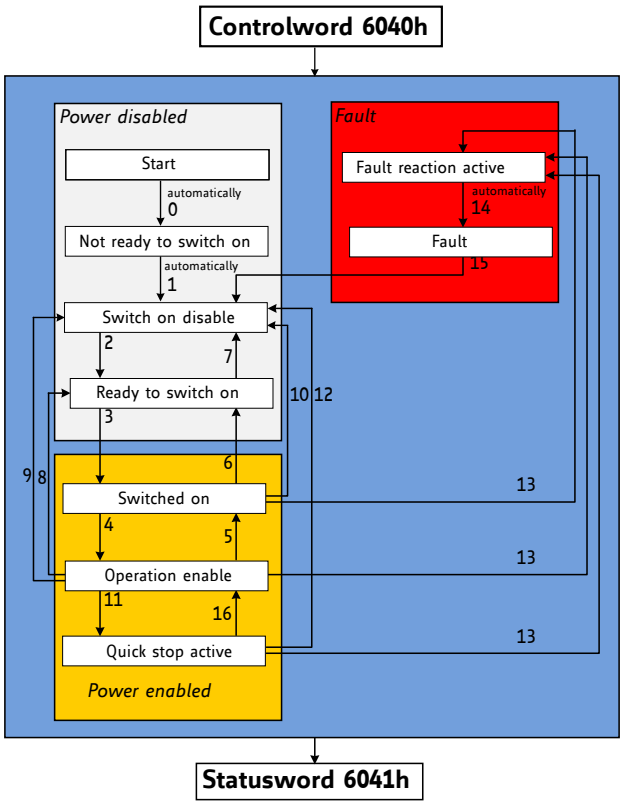


Figure 3: DS402 Finite State Machine

Notes on state transitions:

- Commands directing a change in state are processed completely and the new state achieved before additional state change commands are processed.
- Transitions 0 and 1 occur automatically at drive power-on or reset. Transition 14 occurs automatically, too. All other state changes must be directed by the host.
- Drive function disabled indicates that no current is being supplied to the motor.
- Drive function enabled indicates that current is available for the motor and profile position and profile velocity reference values may be processed.

6.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information.

Structure of the Control Word											
15	11	10	9	8	7	6	4	3	2	1	0
nu		r	oms	h	fr	oms		eo	qs	ev	so
MSB											LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 115: Structure of the Control Word in pp Mode

Operation Mode specific Bits in pp Mode		
Bit	Name	Definition
4	New set point	0-to-1: the next positioning will be started.
5	Change immediately	Not supported.
6	Absolute / relative	0: New position is absolute. 1: New position is relative.
9	Change set point	Not supported.

Table 116: Operation Mode specific Bits in pp Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 117: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 118: Object Description (6040_h in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 119: Entry Description (6040_h in pp Mode)

6.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 120: Structure of the Staus Word in pp Mode

Trinamic Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 121: Trinamic Specific Bits

Operation Mode specific Bits in pp Mode		
Bit	Name	Definition
10	Target reached	Set when the motor is within the position window.
12	Set point acknowledged	0: Set point processed. 1: Set point still in process.
13	Following error	Not supported.

Table 122: Operation Mode specific Bits in pp Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000$_h$	Not ready to switch on
xxxx xxxx x1xx 0000$_h$	Switch on disabled
xxxx xxxx x01x 0001$_h$	Ready to switch on
xxxx xxxx x01x 0011$_h$	Switched on
xxxx xxxx x01x 0111$_h$	Operation enabled
xxxx xxxx x00x 0111$_h$	Quick stop active
xxxx xxxx x0xx 1111$_h$	Fault reaction active
xxxx xxxx x0xx 1000$_h$	Fault

Table 123: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041$_h$	Controlword	Variable	UNSIGNED16

Table 124: Object Description (6041$_h$ in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above.	

Table 125: Entry Description (6041$_h$ in pp Mode)

6.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058$_h$: Position Scaler). Object 6062$_h$ indicates the actual position that the motor should have. It is not to be confused with objects 6063$_h$ and 6064$_h$.

Object Description			
Index	Name	Object Type	Data Type
6062$_h$	Position Demand Value	Variable	SIGNED32

Table 126: Object Description (6062$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 127: Entry Description (6062_h)

6.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063_h	Position Actual Internal Value	Variable	SIGNED32

Table 128: Object Description (6063_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 129: Entry Description (6063_h)

6.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

Object Description			
Index	Name	Object Type	Data Type
6064_h	Position Actual Value	Variable	SIGNED32

Table 130: Object Description (6064_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 131: Entry Description (6064_h)

6.1.6 Object 6067_h: Position Window

This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as

having been reached. The value is given in increments. If the value of the position window is FFFFFFFF$_h$, the position window control is switched off. If this object is set to zero, the target reached event will be signaled when the demand position (6062$_h$) has reached the target position (6064$_h$). When the position window is set to a value greater than zero, the target reached event will be signaled when the actual encoder position value (6064$_h$) is within $(target\_position - position\_window)$ and $(target\_position + position\_window)$.

Object Description			
Index	Name	Object Type	Data Type
6067$_h$	Position Window	Variable	UNSIGNED32

Table 132: Object Description (6067$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	FFFFFFFF$_h$

Table 133: Entry Description (6067$_h$)

6.1.7 Object 6068$_h$: Position Window Time

This object indicates the configured time, during which the actual position within the position window is measured. The value is given in ms. If this object is set to a value greater than zero and also the position window (6067$_h$) is set to a value greater than zero the target reached event will not be signaled until the actual position (6064$_h$) is at least as many milliseconds within the position window as defined by this object.

Object Description			
Index	Name	Object Type	Data Type
6068$_h$	Position Window Time	Variable	UNSIGNED16

Table 134: Object Description (6068$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED16	0

Table 135: Entry Description (6068$_h$)

6.1.8 Object 606C$_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042$_h$ sub index 7).

Object Description			
Index	Name	Object Type	Data Type
$606C_h$	Velocity Actual Value	Variable	SIGNED32

Table 136: Object Description ($606C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 137: Entry Description ($606C_h$)

6.1.9 Object $607A_h$: Target Position

The target position is the position that the drive should move to in profile position mode using the current settings of motion control parameters (such as velocity, acceleration, deceleration, motion profile type etc.). The value of this object is interpreted as absolute or relative depending on the abs/rel flag in the controlword. It is given in PositionScaler steps (see Object 2058_h: Position Scaler).

Object Description			
Index	Name	Object Type	Data Type
$607A_h$	Target Position	Variable	SIGNED32

Table 138: Object Description ($607A_h$ in pp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 139: Entry Description ($607A_h$ in pp Mode)

6.1.10 Object $607D_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 140: Object Description ($607D_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 141: Entry Description ($607D_h$)

6.1.11 Object 6081_h: Profile Velocity

This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The profile velocity is the maximum velocity used when driving to a new position. It is given in units of RPM/s.

Object Description			
Index	Name	Object Type	Data Type
6081_h	Profile Velocity	Variable	UNSIGNED32

Table 142: Object Description (6081_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	0

Table 143: Entry Description (6081_h)

6.1.12 Object 6083_h: Profile Acceleration

This object indicates the configured acceleration. Object 6083_h sets the maximum acceleration to be used in profile position and profile velocity mode.
This value is given using RPM/s units.
In profile velocity mode, this object also sets the deceleration to be used (the deceleration ramp is always the same as the acceleration ramp in pv mode).

Object Description			
Index	Name	Object Type	Data Type
6083_h	Profile Acceleration	Variable	UNSIGNED32

Table 144: Object Description (6083_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	0

Table 145: Entry Description (6083_h)

6.1.13 Object 6084_h: Profile Deceleration

This object indicates the configured deceleration. Object 6084_h sets the maximum deceleration to be used in profile positioning mode.
This value is given in units of RPM/s.

Object Description			
Index	Name	Object Type	Data Type
6084_h	Profile Deceleration	Variable	UNSIGNED32

Table 146: Object Description (6084_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	no	UNSIGNED32	0

Table 147: Entry Description (6084_h)

6.1.14 Object 6085_h: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code object $605A_h$ is set to 2 (or 6). The value is given in the same unit as profile acceleration object 6083_h.

Object Description			
Index	Name	Object Type	Data Type
6085_h	Quick stop deceleration	Variable	UNSIGNED32

Table 148: Object Description (6085_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	51200

Table 149: Entry Description (6085_h)

6.2 How to move a Motor in pp Mode

Here is a little example that shows how to get a motor running in pp mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operational) by NMT commands before. Please note that the values are decimal.

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select pp mode by writing 1 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target position (e.g. 500000) to object $607A_h$.
- Mark the new target position as active by writing 31 to object 6040_h. The motor starts moving now.
- Reset the activation by writing 15 to object 6040_h (this can be done while the motor is still moving).

7 Profile Velocity Mode

The profile velocity mode is used to control the velocity of the drive without a special regard of the position. It contains limit functions and trajectory generation.

The profile velocity mode covers the following sub-functions:

- Demand value input via trajectory generator.
- Monitoring of the profile velocity using a window-function.
- Monitoring of velocity actual value using a threshold.

The operation of the reference value generator and its input parameters include:

- Profile velocity
- Profile acceleration
- Motion profile type

7.0.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information.

In pv mode the control word does not contain any operation mode specific bits.

Structure of the Control Word											
15	11	10	9	8	7	6	4	3	2	1	0
nu		r	r	h	fr	r		eo	qs	ev	so
MSB											LSB

Legend: nu=not used; r=reserved; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 150: Structure of the Control Word in pv Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 151: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 152: Object Description (6040_h in pv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 153: Entry Description (6040_h in pv Mode)

7.0.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 154: Structure of the Status Word in pv Mode

Trinamic Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 155: Trinamic Specific Bits

Operation Mode specific Bits in pv Mode		
Bit	Name	Definition
10	Target reached	Indicates that the target speed has been reached.
12	Speed	Not supported.
13	Max. slippage error	Not supported.

Table 156: Operation Mode specific Bits in pv Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000_h	Not ready to switch on
xxxx xxxx x1xx 0000_h	Switch on disabled
xxxx xxxx x01x 0001_h	Ready to switch on
xxxx xxxx x01x 0011_h	Switched on
xxxx xxxx x01x 0111_h	Operation enabled
xxxx xxxx x00x 0111_h	Quick stop active
xxxx xxxx x0xx 1111_h	Fault reaction active
xxxx xxxx x0xx 1000_h	Fault

Table 157: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041_h	Controlword	Variable	UNSIGNED16

Table 158: Object Description (6041_h in pv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 159: Entry Description (6041_h in pv Mode)

7.0.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). Object 6062_h indicates the actual position that the motor should have. It is not to be confused with objects 6063_h and 6064_h.

Object Description			
Index	Name	Object Type	Data Type
6062_h	Position Demand Value	Variable	SIGNED32

Table 160: Object Description (6062_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 161: Entry Description (6062_h)

7.0.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063_h	Position Actual Internal Value	Variable	SIGNED32

Table 162: Object Description (6063_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 163: Entry Description (6063_h)

7.0.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

Object Description			
Index	Name	Object Type	Data Type
6064_h	Position Actual Value	Variable	SIGNED32

Table 164: Object Description (6064_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 165: Entry Description (6064_h)

7.0.6 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7).

Object Description			
Index	Name	Object Type	Data Type
$606C_h$	Velocity Actual Value	Variable	SIGNED32

Table 166: Object Description ($606C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 167: Entry Description ($606C_h$)

7.0.7 Object $607D_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 168: Object Description ($607D_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 169: Entry Description ($607D_h$)

7.0.8 Object 6083_h: Profile Acceleration

This object indicates the configured acceleration. Object 6083_h sets the maximum acceleration to be used in profile position and profile velocity mode.
This value is given using RPM/s units.
In profile velocity mode, this object also sets the deceleration to be used (the deceleration ramp is always the same as the acceleration ramp in pv mode).

Object Description			
Index	Name	Object Type	Data Type
6083_h	Profile Acceleration	Variable	UNSIGNED32

Table 170: Object Description (6083_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	0

Table 171: Entry Description (6083_h)

7.0.9 Object 6085_h: Quick Stop Deceleration

This object indicates the configured deceleration used to stop the motor when the quick stop function is activated and the quick stop code object $605A_h$ is set to 2 (or 6). The value is given in the same unit as profile acceleration object 6083_h.

Object Description			
Index	Name	Object Type	Data Type
6085_h	Quick stop deceleration	Variable	UNSIGNED32

Table 172: Object Description (6085_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	51200

Table 173: Entry Description (6085_h)

7.0.10 Object $60FF_h$: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator. Object $60FF_h$ sets the target velocity when using profile velocity mode. The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083_h and 6084_h. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7) units.

Object Description			
Index	Name	Object Type	Data Type
$60FF_h$	Target Velocity	Variable	SIGNED32

Table 174: Object Description ($60FF_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 175: Entry Description ($60FF_h$)

7.1 How to move a Motor in pv Mode

Here is a little example that shows how to get a motor running in pv mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operational) by NMT commands before.

- If you do not have any limit switches connected, first disable the limit switch inputs by writing 3 to object 2005_h.
- Select pv mode by writing 3 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Write the desired target speed (e.g. 100000) to object $60FF_h$. The motor now accelerates to that speed.
- Stop the motor by writing 0 to object $60FF_h$.

8 Homing Mode

This chapter describes the method by which a drive seeks the home position (reference point). There are various methods of achieving this using limit switches at the ends of travel or a home switch in mid-travel. Some methods also use the index (zero) pulse train from an incremental encoder. The user may specify the speeds, acceleration and the method of homing.

There is no output data except for those bits in the statusword which return the status or result of the homing process and the demand to the position control loops.

There are four sources of the homing signal available: these are positive and negative limit switches, the home switch and the index pulse from an encoder.

Figure 4 shows the defined input objects as well as the output objects. The user can specify the speeds, acceleration and method of homing. The home offset object $607C_h$ allows displacing the zero in point the coordinate system for the home position.

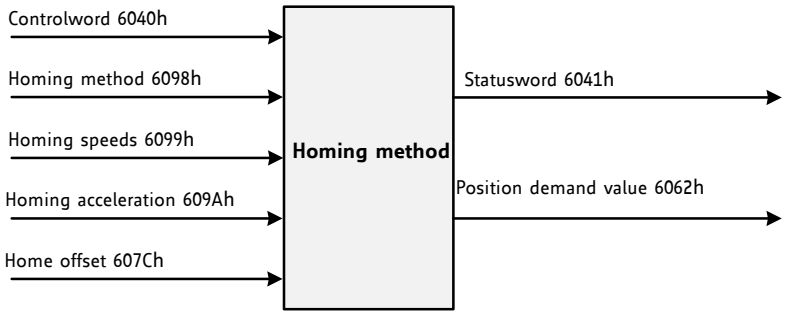


Figure 4: Homing Mode Function

Choosing a homing mode determines the following things:

- The homing signal (positive limit switch, negative limit switch, and home switch).
- The direction of actuation where appropriate.
- The position of the index pulse.

The home position and the zero position are offset by the home offset (see object $607C_h$, section 8.2.4).

Depending on the module there are different sources of homing methods available:

- Negative and positive limit switches.
- Home switch.
- Index pulse of an encoder.

For the operation of positioning drives, an exact knowledge of the absolute position is normally required. Since for cost reasons drives often do not have an absolute encoder, a homing operation is necessary.

8.1 Homing Methods

The TMCM-1617 supports a subset of different standard CANopen homing methods. The homing method that is to be used can be chosen via object 6098_h (section 8.2.5).

Supported Homing Methods	
Method	Description
0	No homing (default value for object 6098_h).
17	Search the left end switch.
18	Search the right end switch.
19	Search the positive edge of the home switch.
21	Search the negative edge of the home switch.
35	The actual position is used as home position. All position values (objects 6062h, 6063h, and 6064h) are set to zero, but the motor will not move.

Table 176: Supported CANopen Homing Methods

When using homing methods that need end switch inputs or home switch inputs please take care of their configuration (object 2005_h, section 4.1.4).

8.1.1 Homing Method 17: Homing on negative Limit Switch

Using this method, the initial direction of movement shall be leftward if the negative limit switch is inactive (here: low). The home position shall at the point where the negative limit switch becomes inactive.

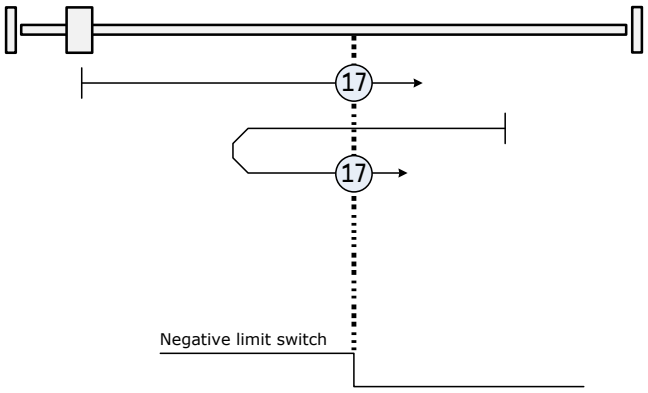


Figure 5: Homing Method 17

8.1.2 Homing Method 18: Homing on positive Limit Switch

Using this method, the initial direction of movement shall be rightward if the positive limit switch is inactive (here: low). The home position shall be at point the where the positive limit switch becomes inactive.

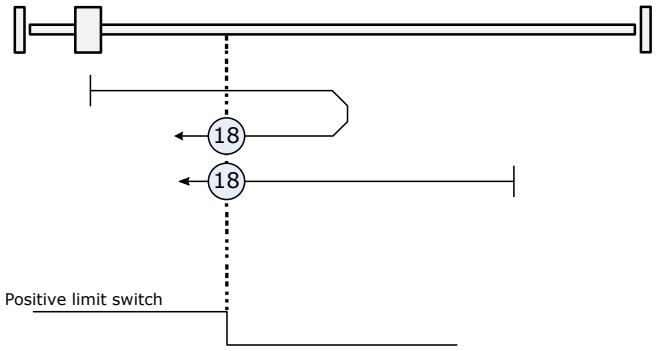


Figure 6: Homing Method 18

8.1.3 Homing Method 19: Homing on positive Home Switch

Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the point where the home switch changes state. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

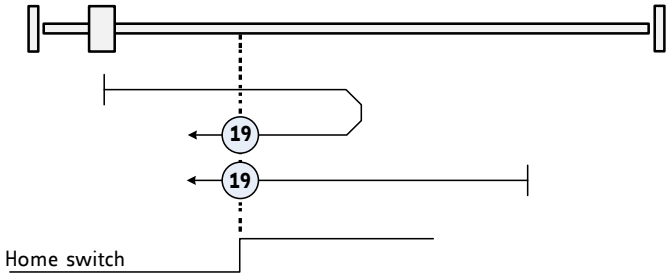


Figure 7: Homing Method 19

8.1.4 Homing Method 21: Homing on negative Home Switch

Using this method, the initial direction of movement shall be dependent on the state of the home switch. The home position shall be at the point where the home switch changes state. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

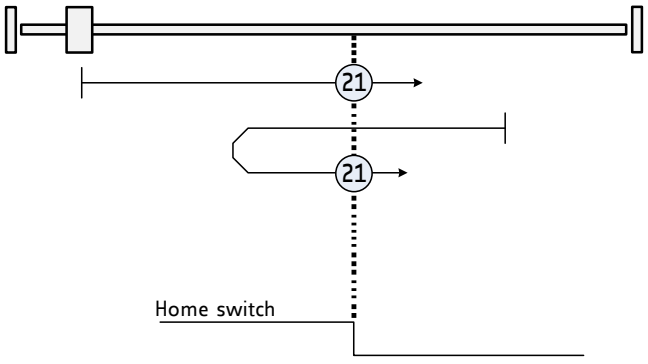


Figure 8: Homing Method 21

8.1.5 Homing Method 35: Current Position as Home Position

In this method, the current position shall be taken to be the home position. This method does not require the drive device to be in operation enabled state.

8.2 Detailed Object Specifications

8.2.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information.

Structure of the Control Word																
15				11	10	9	8	7	6		4	3	2	1	0	
nu					r	oms	h	fr	oms			eo	qs	ev	so	
MSB																LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 177: Structure of the Control Word in hm Mode

Operation Mode specific Bits in hm Mode		
Bit	Name	Definition
4	Homing operation start	1: start homing; 0: stop homing
8	Halt	Not supported.

Table 178: Operation Mode specific Bits in hm Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 179: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 180: Object Description (6040_h in hm Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 181: Entry Description (6040_h in hm Mode)

8.2.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 182: Structure of the Status Word in hm Mode

Trinamic Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 183: Trinamic Specific Bits

Operation Mode specific Bits in hm Mode		
Bit	Name	Definition
10	Target reached	Set when the zero position has been found or homing has been stopped by setting controlword bit 4 to zero.
12	Home attained	Set when zero position has been found.
13	Homing error	Not supported.

Table 184: Operation Mode specific Bits in hm Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000_h	Not ready to switch on
xxxx xxxx x1xx 0000_h	Switch on disabled
xxxx xxxx x01x 0001_h	Ready to switch on
xxxx xxxx x01x 0011_h	Switched on
xxxx xxxx x01x 0111_h	Operation enabled
xxxx xxxx x00x 0111_h	Quick stop active
xxxx xxxx x0xx 1111_h	Fault reaction active
xxxx xxxx x0xx 1000_h	Fault

Table 185: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041_h	Controlword	Variable	UNSIGNED16

Table 186: Object Description (6041_h in hm Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above.	

Table 187: Entry Description (6041_h in hm Mode)

8.2.3 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7).

Object Description			
Index	Name	Object Type	Data Type
$606C_h$	Velocity Actual Value	Variable	SIGNED32

Table 188: Object Description ($606C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 189: Entry Description ($606C_h$)

8.2.4 Object $607C_h$: Home Offset

This object indicates the configured difference between the zero position for the application and the machine home position/home switch (found during homing). While homing, the machine home position is found and once the homing is completed, the zero position is offset from the home position by adding the home offset to the home position. The effect of setting the home position to a non-zero value depends on the selected homing method. The value of this object is given in PositionScaler steps (see Object 2058_h: Position Scaler). Negative values indicate the opposite direction.

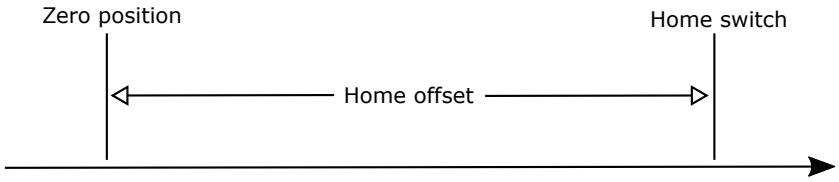


Figure 9: Home Offset

Object Description			
Index	Name	Object Type	Data Type
$607C_h$	Home offset	Variable	SIGNED32

Table 190: Object Description ($607C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	SIGNED32	0

Table 191: Entry Description ($607C_h$)

8.2.5 Object 6098_h: Homing Method

The homing method to be used can be selected by writing to this object. Please see table 176 for a list of homing methods supported by the current version of the TMCM-1617 CANopen firmware.

Object Description			
Index	Name	Object Type	Data Type
6098_h	Homing method	Variable	SIGNED8

Table 192: Object Description (6098_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	SIGNED8	0

Table 193: Entry Description (6098_h)

8.2.6 Object 6099_h: Homing Speeds

This object indicates the configured speeds used during homing procedure. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7) units. Using object 6099_h a fast and a slow homing speed can be set. In most homing modes, the home switch is searched with the fast speed first. When the home switch has been found, the motor will be decelerated to the slow speed (using the homing acceleration, object $609A_h$) to search for the exact switch point. When the switch point has been found the motor will be stopped at that point.

Object Description			
Index	Name	Object Type	Data Type
6099_h	Homing speeds	Array	UNSIGNED32

Table 194: Object Description (6099_h)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Fast homing speed	rw	no	UNSIGNED32	0
2	Slow homing speed	rw	no	UNSIGNED32	0

Table 195: Entry Description (6099_h)

8.2.7 Object $609A_h$: Homing Acceleration

This object indicates the configured acceleration and deceleration to be used during homing operation. This object used RPM/s units.

Object Description			
Index	Name	Object Type	Data Type
$609A_h$	Homing acceleration	Variable	UNSIGNED32

Table 196: Object Description ($609A_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	UNSIGNED32	0

Table 197: Entry Description ($609A_h$)

8.3 How to start a Homing in hm Mode

Here is a little example that shows how to home the motor in hm mode. In this little example we assume that the module has been reset (and then switched to pre-operational or operational) by NMT commands before. The home switch must be connected to the home switch input. It can be operated manually.

- Select hm mode by writing 6 to object 6060_h.
- Write 6 to object 6040_h to switch to READY_TO_SWITCH_ON state.
- Write 7 to object 6040_h to switch to SWITCHED_ON state.
- Write 15 to object 6040_h to switch to OPERATION_ENABLED state.
- Select homing method 19 by writing 19 to object 6098_h.
- Set the homing speeds by writing e.g. 50000 to object 6099_h sub index 1 and e.g. 10000 to object 6099_h sub index 2.
- Write 31 to object 6040_h to start the homing process.
- Press and release the home switch.
- When homing has finished, write 15 to object 6040_h again.

9 Cyclic synchronous Position Mode

The cyclic synchronous position mode is used to directly control the position of the motor. It contains limit functions, but not a trajectory generator. The trajectory generator is located in the control device (the master), not in the drive device. In cyclic synchronous manner, the control device provides a target position to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target position (object $607A_h$, see section 9.1.7) and the interpolation time period (object $60C2_h$, see section 9.1.10). The drive automatically sets the velocity in such a manner that the next target position is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

The cyclic synchronous position mode covers the following sub-functions:

- Position demand value input directly via an object.
- Monitoring of the position.
- Limiting the position using the software limits or the hardware limit switches.

9.1 Detailed Object Specifications

9.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information. The cyclic synchronous position mode does not use any mode specific bits of the control word.

Structure of the Control Word									
15	9	8	7	6	4	3	2	1	0
nu		h	fr	nu		eo	qs	ev	so
MSB									LSB

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 198: Structure of the Control Word in csp Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 199: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 200: Object Description (6040_h in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 201: Entry Description (6040_h in csp Mode)

9.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	r	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 202: Structure of the Status Word in csp Mode

Trinamic Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 203: Trinamic Specific Bits

Operation Mode specific Bits in csp Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target position ignored	0: Target position ignored. 1: Target position used as input to position controller.
13	Following error	0: No following error. 1: Following error.

Table 204: Operation Mode specific Bits in csp Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000_h	Not ready to switch on
xxxx xxxx x1xx 0000_h	Switch on disabled
xxxx xxxx x01x 0001_h	Ready to switch on
xxxx xxxx x01x 0011_h	Switched on
xxxx xxxx x01x 0111_h	Operation enabled
xxxx xxxx x00x 0111_h	Quick stop active
xxxx xxxx x0xx 1111_h	Fault reaction active
xxxx xxxx x0xx 1000_h	Fault

Table 205: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041_h	Controlword	Variable	UNSIGNED16

Table 206: Object Description (6041_h in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 207: Entry Description (6041_h in csp Mode)

9.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). Object 6062_h indicates the actual position that the motor should have. It is not to be confused with objects 6063_h and 6064_h.

Object Description			
Index	Name	Object Type	Data Type
6062_h	Position Demand Value	Variable	SIGNED32

Table 208: Object Description (6062_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 209: Entry Description (6062_h)

9.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063_h	Position Actual Internal Value	Variable	SIGNED32

Table 210: Object Description (6063_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 211: Entry Description (6063_h)

9.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

Object Description			
Index	Name	Object Type	Data Type
6064_h	Position Actual Value	Variable	SIGNED32

Table 212: Object Description (6064_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 213: Entry Description (6064_h)

9.1.6 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7).

Object Description			
Index	Name	Object Type	Data Type
$606C_h$	Velocity Actual Value	Variable	SIGNED32

Table 214: Object Description ($606C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 215: Entry Description ($606C_h$)

9.1.7 Object $607A_h$: Target Position

The target position is the position that the drive should move to in cyclic synchronous position mode using the current interpolation time period. In csp mode this value is always interpreted as an absolute value.

Object Description			
Index	Name	Object Type	Data Type
$607A_h$	Target Position	Variable	SIGNED32

Table 216: Object Description ($607A_h$ in csp Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 217: Entry Description ($607A_h$ in csp Mode)

9.1.8 Object $607D_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 218: Object Description ($607D_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 219: Entry Description ($607D_h$)

9.1.9 Object $60B0_h$: Position Offset

This object provides an offset to the target position (object $607A_h$, see section 9.1.7)). The value is given in PositionScaler steps (see Object 2058_h: Position Scaler) and will be added to the target position.

Object Description			
Index	Name	Object Type	Data Type
$60B0_h$	Position Offset	Variable	SIGNED32

Table 220: Object Description ($60B0_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-2147483648...2147483647	0

Table 221: Entry Description ($60B0_h$)

9.1.10 Object $60C2_h$: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01_h) is given in $10^{interpolation\_time\_index}$ s. The interpolation time index (sub-index 02_h) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
$60C2_h$	Interpolation Time Period	Record	Interpolation time period record (0080_h)

Table 222: Object Description ($60C2_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	UNSIGNED8	2
1	Interpolation time period value	rw	no	UNSIGNED8	1
2	Interpolation time index	rw	no	-3...3	-3

Table 223: Entry Description ($60C2_h$)

10 Cyclic synchronous Velocity Mode

The cyclic synchronous velocity mode is used to directly control the velocity of the motor. It contains limit functions, but not a trajectory generator. The trajectory generator is located in the control device (the master), not in the drive device. In cyclic synchronous manner, the control device provides a target velocity to the drive device, which performs position control, velocity control and torque control.

The main control parameters are the target velocity (object $60FF_h$, see section 10.1.4) and the interpolation time period (object $60C2_h$, see section 10.1.7). The drive automatically sets the acceleration in such a manner that the next target velocity is reached within the interpolation time period. Acceleration and deceleration ramps are not used in this mode.

The cyclic synchronous velocity mode covers the following sub-functions:

- Velocity demand value input directly via an object.
- Monitoring of the position.
- Limiting the position using the software limits or the hardware limit switches.

10.1 Detailed Object Specifications

10.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information. The cyclic synchronous velocity mode does not use any mode specific bits of the control word.

Structure of the Control Word									
15	9	8	7	6	4	3	2	1	0
nu		h	fr	nu		eo	qs	ev	so
MSB									LSB

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 224: Structure of the Control Word in csv Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 225: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 226: Object Description (6040_h in csv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 227: Entry Description (6040_h in csv Mode)

10.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	r	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 228: Structure of the Status Word in csv Mode

Trinamic Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 229: Trinamic Specific Bits

Operation Mode specific Bits in csv Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target position ignored	0: Target velocity ignored. 1: Target velocity used as input to velocity controller.
13	Reserved	Not used.

Table 230: Operation Mode specific Bits in csv Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000_h	Not ready to switch on
xxxx xxxx x1xx 0000_h	Switch on disabled
xxxx xxxx x01x 0001_h	Ready to switch on
xxxx xxxx x01x 0011_h	Switched on
xxxx xxxx x01x 0111_h	Operation enabled
xxxx xxxx x00x 0111_h	Quick stop active
xxxx xxxx x0xx 1111_h	Fault reaction active
xxxx xxxx x0xx 1000_h	Fault

Table 231: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041_h	Controlword	Variable	UNSIGNED16

Table 232: Object Description (6041_h in csv Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 233: Entry Description (6041_h in csv Mode)

10.1.3 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7).

Object Description			
Index	Name	Object Type	Data Type
$606C_h$	Velocity Actual Value	Variable	SIGNED32

Table 234: Object Description ($606C_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 235: Entry Description ($606C_h$)

10.1.4 Object $60FF_h$: Target Velocity

In csv mode the target velocity specifies the velocity that is to be reached within the interpolation time period. The values are given in mechanical RPM or electrical RPM (depending on the setting made to 2042_h sub index 7) units.

Object Description			
Index	Name	Object Type	Data Type
$60FF_h$	Target Velocity	Variable	SIGNED32

Table 236: Object Description ($60FF_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	SIGNED32	0

Table 237: Entry Description ($60FF_h$)

10.1.5 Object 607D$_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
607D$_h$	Software Position Limit	Array	SIGNED32

Table 238: Object Description (607D$_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 239: Entry Description (607D$_h$)

10.1.6 Object 60B1$_h$: Velocity Offset

This object provides an offset to the target velocity (object 60FF$_h$, see section 10.1.4)). The value will be added to the target velocity.

Object Description			
Index	Name	Object Type	Data Type
60B1$_h$	Velocity Offset	Variable	SIGNED32

Table 240: Object Description (60B1$_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-2147483648...2147483647	0

Table 241: Entry Description (60B1$_h$)

10.1.7 Object $60C2_h$: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01_h) is given in $10^{interpolation\_time\_index}$ s. The interpolation time index (sub-index 02_h) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
$60C2_h$	Interpolation Time Period	Record	Interpolation time period record (0080_h)

Table 242: Object Description ($60C2_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	UNSIGNED8	2
1	Interpolation time period value	rw	no	UNSIGNED8	1
2	Interpolation time index	rw	no	-3...3	-3

Table 243: Entry Description ($60C2_h$)

11 Cyclic synchronous Torque Mode

The cyclic synchronous torque mode is used to directly control the torque of the motor, without the need for position or velocity control. It contains limit functions, but not a trajectory generator.
The cyclic synchronous torque mode covers the following sub-functions:

- Demand value input directly via an object.
- Monitoring of the torque.
- Limiting the position using the software limits or the hardware limit switches.

11.1 Detailed Object Specifications

11.1.1 Object 6040_h: Control Word

This object indicates the received command controlling the power drive system finite state automaton (PDS FSA). The CiA-402 state machine can be controlled using this object. Please refer to figure 3 for detailed information. The cyclic synchronous torque mode does not use any mode specific bits of the control word.

Structure of the Control Word									
15	9	8	7	6	4	3	2	1	0
nu		h	fr	nu		eo	qs	ev	so
MSB									LSB

Legend: nu=not used; h=halt; fr=fault reset; eo=enable operation; qs=quick stop; ev=enable voltage; so=switch on.

Table 244: Structure of the Control Word in cst Mode

Command Coding						
Command	Bits of Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on & enable operation	0	1	1	1	1	3, 4
Disable voltage	0	x	x	0	x	7,9,10,12
Quick stop	0	x	0	1	x	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0-to-1	x	x	x	x	15

Table 245: Command Coding

Object Description			
Index	Name	Object Type	Data Type
6040_h	Controlword	Variable	UNSIGNED16

Table 246: Object Description (6040_h in cst Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See command coding above.	

Table 247: Entry Description (6040_h in cst Mode)

11.1.2 Object 6041_h: Status Word

This object provides the status of the PDS FSA. It reflects the status of the CiA-402 state machine. Please refer to figure 3 for detailed information. The object is structured as defined below.
For more information about the coding please refer to the CANopen Drives and motion control device profile, part 2.

Structure of the Status Word															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dir	mot	oms		ila	r	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Legend: nu=not used; r=reserved; oms=operation mode specific; h=halt; fr=fault reset; oe=operation enable; qs=quick stop; ve=voltage enable; so=switch on.

Table 248: Structure of the Status Word in cst Mode

Trinamic Specific Bits		
Bit	Name	Definition
14	Motor activity	0: Motor stands still. 1: Motor rotates.
15	Direction of rotation	This bit shows the direction of rotation.

Table 249: Trinamic Specific Bits

Operation Mode specific Bits in cst Mode		
Bit	Name	Definition
10	Reserved	Not used.
12	Target torque ignored	0: Target torque ignored. 1: Target torque used as input to control loop.
13	Reserved	Not used.

Table 250: Operation Mode specific Bits in cst Mode

State Coding	
Status word	FSA state
xxxx xxxx x0xx 0000_h	Not ready to switch on
xxxx xxxx x1xx 0000_h	Switch on disabled
xxxx xxxx x01x 0001_h	Ready to switch on
xxxx xxxx x01x 0011_h	Switched on
xxxx xxxx x01x 0111_h	Operation enabled
xxxx xxxx x00x 0111_h	Quick stop active
xxxx xxxx x0xx 1111_h	Fault reaction active
xxxx xxxx x0xx 1000_h	Fault

Table 251: State Coding

Object Description			
Index	Name	Object Type	Data Type
6041_h	Controlword	Variable	UNSIGNED16

Table 252: Object Description (6041_h in cst Mode)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	See state coding above	

Table 253: Entry Description (6041_h in cst Mode)

11.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). Object 6062_h indicates the actual position that the motor should have. It is not to be confused with objects 6063_h and 6064_h.

Object Description			
Index	Name	Object Type	Data Type
6062_h	Position Demand Value	Variable	SIGNED32

Table 254: Object Description (6062_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 255: Entry Description (6062_h)

11.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in PositionScaler steps (see Object 2058_h: Position Scaler). It is the same as object 6062_h.

Object Description			
Index	Name	Object Type	Data Type
6063_h	Position Actual Internal Value	Variable	SIGNED32

Table 256: Object Description (6063_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 257: Entry Description (6063_h)

11.1.5 Object 6064_h: Position Actual Value

This object provides the actual value of the position measurement device. It always contains the same value as object 6063_h.

Object Description			
Index	Name	Object Type	Data Type
6064_h	Position Actual Value	Variable	SIGNED32

Table 258: Object Description (6064_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	SIGNED32	no

Table 259: Entry Description (6064_h)

11.1.6 Object 6071_h: Target Torque

This object sets the desired torque value. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
6071_h	Target torque	Variable	INTEGER16

Table 260: Object Description (6071_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-32768...32767	0

Table 261: Entry Description (6071_h)

11.1.7 Object 6077_h: Torque actual Value

This object provides the actual torque value. The value is given in mA.

Object Description			
Index	Name	Object Type	Data Type
6077_h	Torque actual Value	Variable	INTEGER16

Table 262: Object Description (6077_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	ro	yes	-32768...32767	0

Table 263: Entry Description (6077_h)

11.1.8 Object $607D_h$: Software Position Limit

This object indicates the configured maximal and minimal software position limits. These parameters define the absolute position limits for the position demand value and the position actual value. Every

new target position is checked against these limits. The limit positions are always relative to the machine home position. Before being compared with the target position, they are corrected internally by the home offset as follows:

$$Corrected\_min\_position\_limit = min\_position\_limit - home\_offset$$
$$Corrected\_max\_position\_limit = max\_position\_limit - home\_offset$$

Object Description			
Index	Name	Object Type	Data Type
$607D_h$	Software Position Limit	Array	SIGNED32

Table 264: Object Description ($607D_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
1	Minimum Position Limit	rw	no	SIGNED32	-2147483648
2	Maximum Position Limit	rw	no	SIGNED32	2147483647

Table 265: Entry Description ($607D_h$)

11.1.9 Object $60B2_h$: Torque Offset

This object provides an offset to the torque value. It will be added to the target torque (object 6071_h, see section 11.1.6).

Object Description			
Index	Name	Object Type	Data Type
$60B2_h$	Torque Offset	Variable	SIGNED16

Table 266: Object Description ($60B2_h$)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	yes	-32768...32767	0

Table 267: Entry Description ($60B2_h$)

11.1.10 Object $60C2_h$: Interpolation Time Period

This object indicates the interpolation cycle time. The interpolation time period (sub-index 01_h) is given in $10^{interpolation\_time\_index}$ s. The interpolation time index (sub-index 02_h) is dimensionless.

Object Description			
Index	Name	Object Type	Data Type
$60C2_h$	Interpolation Time Period	Record	Interpolation time period record (0080_h)

Table 268: Object Description ($60C2_h$)

Entry Description					
Sub-index	Description	Access	PDO Mapping	Value Range	Default Value
0	Highest sub-index supported	ro	no	UNSIGNED8	2
1	Interpolation time period value	rw	no	UNSIGNED8	1
2	Interpolation time index	rw	no	-3…3	-3

Table 269: Entry Description ($60C2_h$)

12 Emergency Messages (EMCY)

The module sends an emergency message if an error occurs. The message contains information about the error type. The module can map internal errors and object 1001_h (error register) is part of every emergency object.

Emergency Messages (EMCY) of the TMCM-1617						
Error code	Additional byte					Description
	1	2	3	4	5	
0000_h	0	0	0	0	0	**Fault reset** The fault reset command has been executed.
4310_h	2	0	0	0	0	**Overtemperature error** The motor driver has been switched off because the temperature limit has been exceeded.
5441_h	0	255	0	0	0	**Shutdown switch active** The enable signal is missing (due to the shutdown switch) and the motor driver has been switched off.
6320_h	0	255	0	0	0	**Parameter error** The data in the received PDO is either wrong or cannot be accepted due to the internal state of the drive.
8611_h	0	0	0	0	0	**Following error** The deviation between motor position counter and encoder position counter has exceeded the following error window.
$ff00_h$	0	0	0	0	0	**Undervoltage** The supply voltage is too low to drive a motor.
$ff01_h$	1	0	0	0	0	**Positive software limit** The actual position is outside the range defined by object $607d_h$.
$ff01_h$	2	0	0	0	0	**Negative software limit** The actual position is outside the range defined by object $607d_h$.
$ff01_h$	3	0	0	0	0	**Positive limit switch** The positive limit switch has been touched outside of the homing function.
$ff01_h$	4	0	0	0	0	**Negative limit switch** The negative limit switch has been touched outside of the homing function.

Table 270: Emergency Messages (EMCY)

13 SDO Abort Codes

Trying to access an object via SDO read or SDO write may result in an error. In such a case an SDO abort transfer message containing an abort code will be sent. The following table lists all SDO abort codes defined by the ETG.1000.6 standard. Not all of these are used by the TMCM-1617 module.

SDO Abort Codes	
Abort code	Description
05030000_h	Toggle bit not alternated.
05040000_h	SDO protocol timed out.
05040001_h	Client/server command specifier not valid or unknown.
05040005_h	Out of memory.
06010000_h	Unsupported access to an object.
06010001_h	Attempt to read a write only object.
06010002_h	Attempt to write a read only object.
06010003_h	Subindex cannot be written, SI0 must be 0 for write access.
06010004_h	SDO complete access not supported for objects of variable length such as ENUM object types.
06010005_h	Object length exceeds mailbox size.
06010006_h	Object mapped to RPDO, SDO download blocked.
06020000_h	Object does not exist in object dictionary.
06040041_h	Object cannot be mapped to the PDO.
06040042_h	The number and length of the objects to be mapped would exceed the PDO length.
06040043_h	General parameter incompatibility reason.
06040047_h	General internal incompatibility in the device.
06060000_h	Access failed due to a hardware error.
06070010_h	Data type does not match, length of service parameter does not match.
06070012_h	Data type does not match, length of service parameter too high.
06070013_h	Data type does not match, length of service parameter too low.
06090011_h	Sub-index does not exist.
06090030_h	Value range of parameter exceeded.
06090031_h	Value of parameter too high.
06090032_h	Value of parameter too low.
06090036_h	Maximum value is less than minimum value.
08000000_h	General error.
08000020_h	Data cannot be transferred or stored to the application.
08000021_h	Data cannot be transferred or stored to the application because of local control.

Abort code	Description
08000022_h	Data cannot be transferred or stored to the application because of the present device state.
08000023_h	Object dictionary dynamic generation failed or no object dictionary is present.

Table 271: SDO Abort Codes

14 Firmware Update

The firmware update files can be downloaded from the TMCM-1617's product page [1].

Note	Once the update is completed, it is mandatory to also update the "Revision Number" in the SII EEPROM of the TMCM-1617. Check out on how to accomplish this using TwinCAT 3 in section 14.3.

14.1 Over RS485

If the used base board for the TMCM-1617 is equipped with an RS485 transceiver the firmware can be updated with an USB-RS485 converter using the TMCL-IDE [2]. For the upload via RS485 the .hex-file is used.

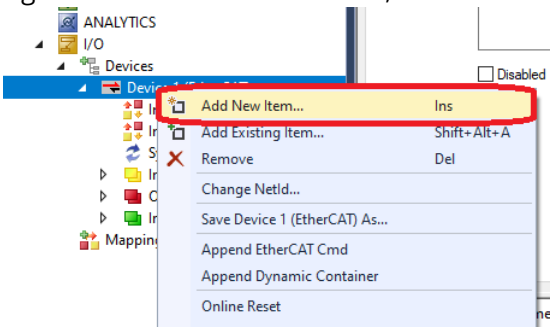
14.2 Over FoE

The firmware can also be updated using the File over EtherCAT (FoE) protocol. Before starting the download the TMCM-1617 must be set to bootstrap mode. In order to make the upload work it is necessary to set the mailbox timeout to at least 5000 ms. The FoE protocol requires to give a file name and password for a file to be uploaded, both are not evaluated by the TMCM-1617 so these values do not matter. Make sure to upload a valid firmware file, for FoE .bin-files are used. After the upload is complete, a power cycle is required in order to complete the update.

14.3 Updating the SII EEPROM using TwinCAT 3

This is a step by step description on how to update the SII EEPROM after a firmware update. It is expected that the TMCM-1617 to be updated is connected to the TwinCAT master system. Also make sure the new ESI-XML file that came with the new firmware was copied into the TwinCAT install directory.

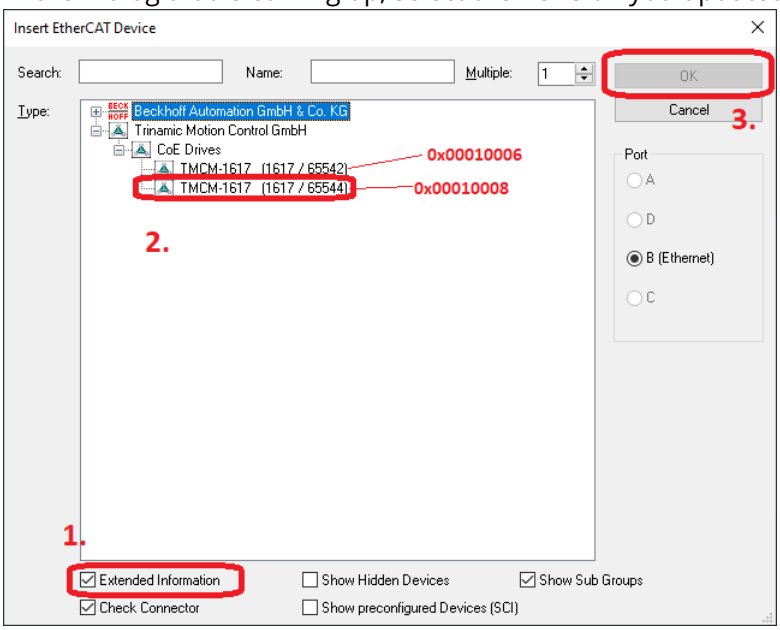
- right-click the EtherCAT master, the TMCM-1617 is connected to and click "Add New Item...".



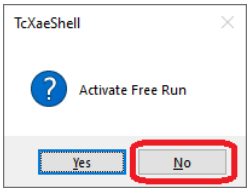
[1] TMCM-1617 product page: https://www.trinamic.com/products/modules/details/tmcm-1617/

[2] TMCL-IDE: https://www.trinamic.com/support/software/tmcl-ide

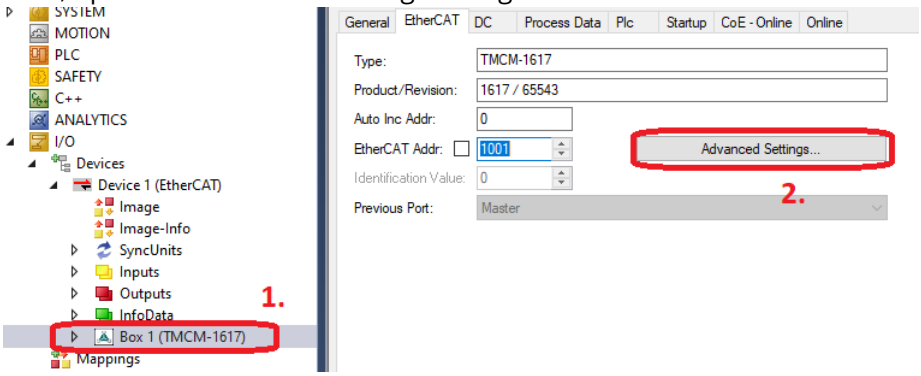
- In the Dialog that is coming up, select the Revision you updated the firmware to.



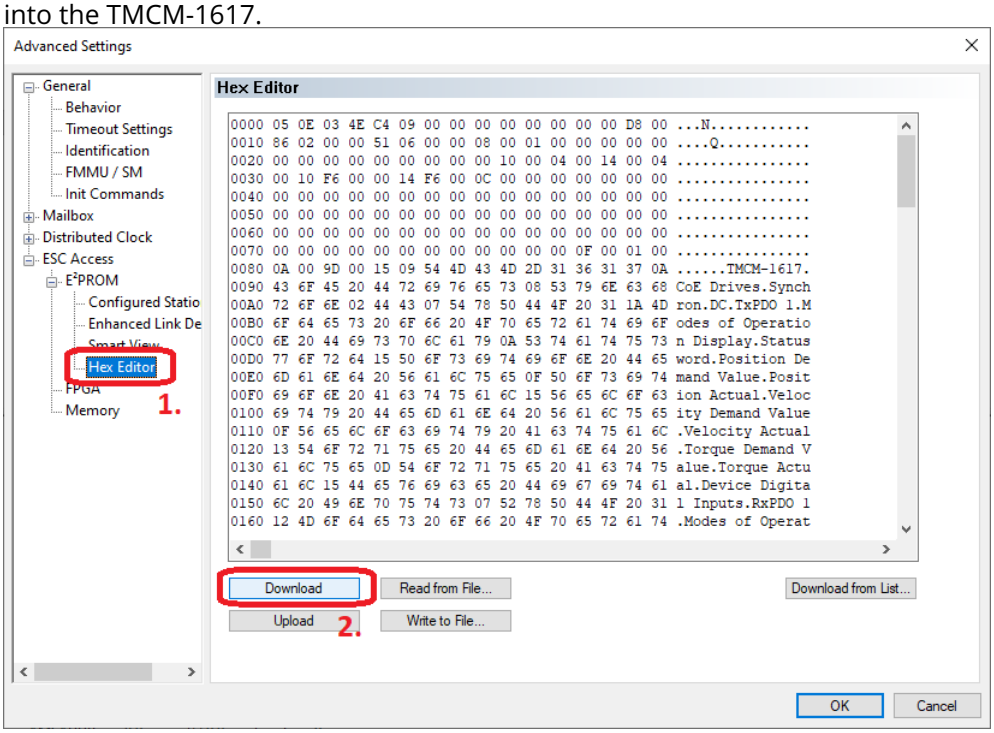
- Click “Reload Devices”  and if you are asked to “Activate Free Run” click “No”.



- Now, open the “Advanced Settings Dialog”.



- There you just download the EEPROM contend that was generated from the ESI-XML information into the TMCM-1617.



- The download will take several minutes.

To check if the Revision number in the SII EEPROM is in sync with the firmware version in the flash memory, compare Revision Number in the SII EEPROM with number stored in subindex 3 of object Object 1018_h: Identity Object, they must be equal.

15 Figures Index

16 Tables Index

17 Supplemental Directives

17.1 Producer Information

17.2 Copyright

17.3 Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by TRINAMIC or by other manufacturers, whose products are used or referred to in combination with TRINAMIC's products and TRINAMIC's product documentation.

This CoE Firmware Manual is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

17.4 Target User

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems or devices, in which the user incorporates the product.

17.5 Disclaimer: Life Support Systems

TRINAMIC Motion Control GmbH & Co. KG does not authorize or warrant any of its products for use in life support systems, without the specific written consent of TRINAMIC Motion Control GmbH & Co. KG.

Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

17.6 Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose

or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. TRINAMIC products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without TRINAMIC's specific written consent.

TRINAMIC products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by TRINAMIC. TRINAMIC conveys no patent, copyright, mask work right or other trade mark right to this product. TRINAMIC assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

17.7 Collateral Documents & Tools

This product documentation is related and/or associated with additional tool kits, firmware and other items, as provided on the product page at: www.trinamic.com.

18 Revision History

18.1 Firmware Revision

Version	Date	Author	Description
V1.06	2020-03-25	ED/OK	First release.
V1.08	2021-12-17	BP	Update release. • The following object entries were added: – Within Object 2042_h: Velocity Mode Settings: * Sensor_Selection (subindex 5) * Velocity_Filter (subindex 6) * Velocity_Unit (subindex 7) * Motor_Halted_Velocity (subindex 8) – Sensor_Selection (subindex 5) within Object 2043_h: Position Mode Settings • The data type of Steps (subindex 2) within Object 2080_h: ABN Encoder Settings changed from UINT to UDINT • Default parameters were changed: – Object 2050_h: Motor Type from 0 (no motor) to 3 (Three phase BLDC motor) – P_ Parameter (subindex 1) of the Object 2043_h: Position Mode Settings changed from 300 to 50 • Rework of the implementation of Object 1010_h: Store Parameters and Object 1011_h: Restore Parameters. PDO configuration can now be stored as part of the “Communication Parameters”. • The internal scaling of the PI regulators P parameter changed. • The Overcurrent flag and Driver Error flag of the Object 2101_h: Motor Status Flags got implemented. • The error LED output handling got implemented. • Fix: Bad ramp at first position move after power-on reset. • Fix: In PDO the “motor rotates” status flag does not go active if the motor turns.

Table 272: Firmware Revision

18.2 Document Revision

Version	Date	Author	Description
V1.00	2020-03-26	OK	First release.
V1.01	2021-02-28	SK	Removed analog encoder option.

Version	Date	Author	Description
V1.02	2021-03-22	OK	Object 2702_h corrected.
V1.03	2021-12-17	BP/OK	New firmware Release (V1.08). The changes to the firmware and ESI file are integrated into this document, see the Firmware Revision History. • Removed object 6065_h and $60F2_h$ as they are not available in the firmware. • Object 6084_h corrected - access is “ro” instead of “rw”. • Added section SDO Abort Codes. • Added section Firmware Update.

Table 273: Document Revision