PD60-4H-1461-CoE CoE Firmware Manual

Firmware Version V1.00 | Document Revision V1.00 • 2022-SEP-15

PD60-4H-1461-CoE is a single axis stepper motor servo drive for up to +48V supply voltage and up to 3Nm torque. A built-in magnetic encoder is used for closed loop operation of the motor. The module communicates using the CANopen over EtherCAT® protocol. The PD60-4H-1461-CoE comes with a housing and can be interfaced with M8 type connectors.

EtherCAT®

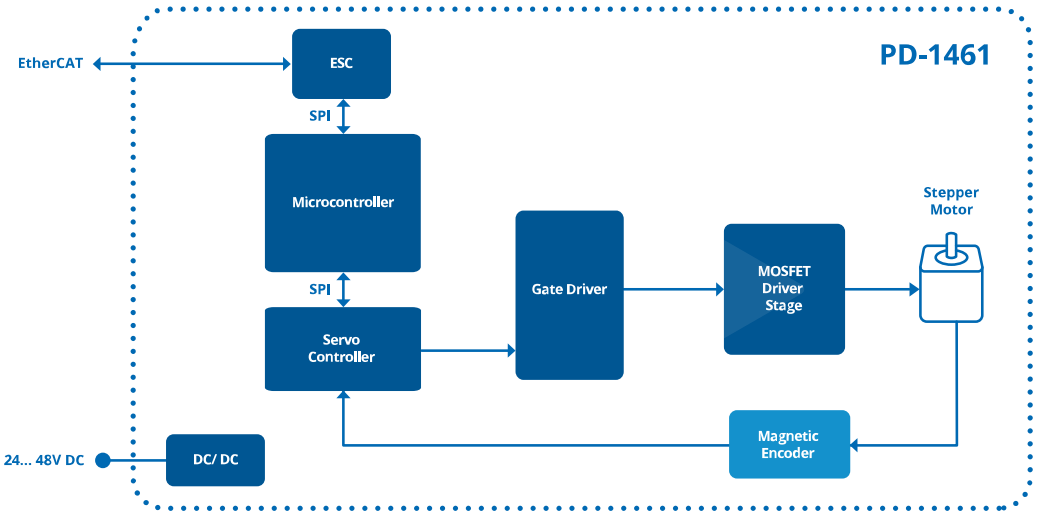
Features

- Supply Voltage +24 to +48V DC
- Up to 3Nm
- Field Oriented Control (FOC) of the stepper motor
- Integrated position feedback
- EtherCAT® interface
- CoE protocol
- NEMA24 size stepper motor
- Housing

Applications

- Lab-Automation
- Manufacturing
- Robotics
- Factory Automation
- CNC

Simplified Block Diagram



ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

ADI Trinamic™

Read entire documentation.

Contents

ADI Trinamic™

1 Preface

This document specifies objects and modes of operation of the Triamic PD60-4H-1461-CoE 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).

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. 9 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	
CoE	CANopen over EtherCAT
FoE	File transfer over EtherCAT
FSA	Finite state automaton
FSM	Finite state machine
NMT	Network management
LSB	Least significant bit/byte
MSB	Most significant bit/byte
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
csp	Cyclic synchronous position mode
csv	Cyclic synchronous velocity mode
cst	Cyclic synchronous torque 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. New firmware can be loaded into the module via file transfer over EtherCAT (FoE).

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.

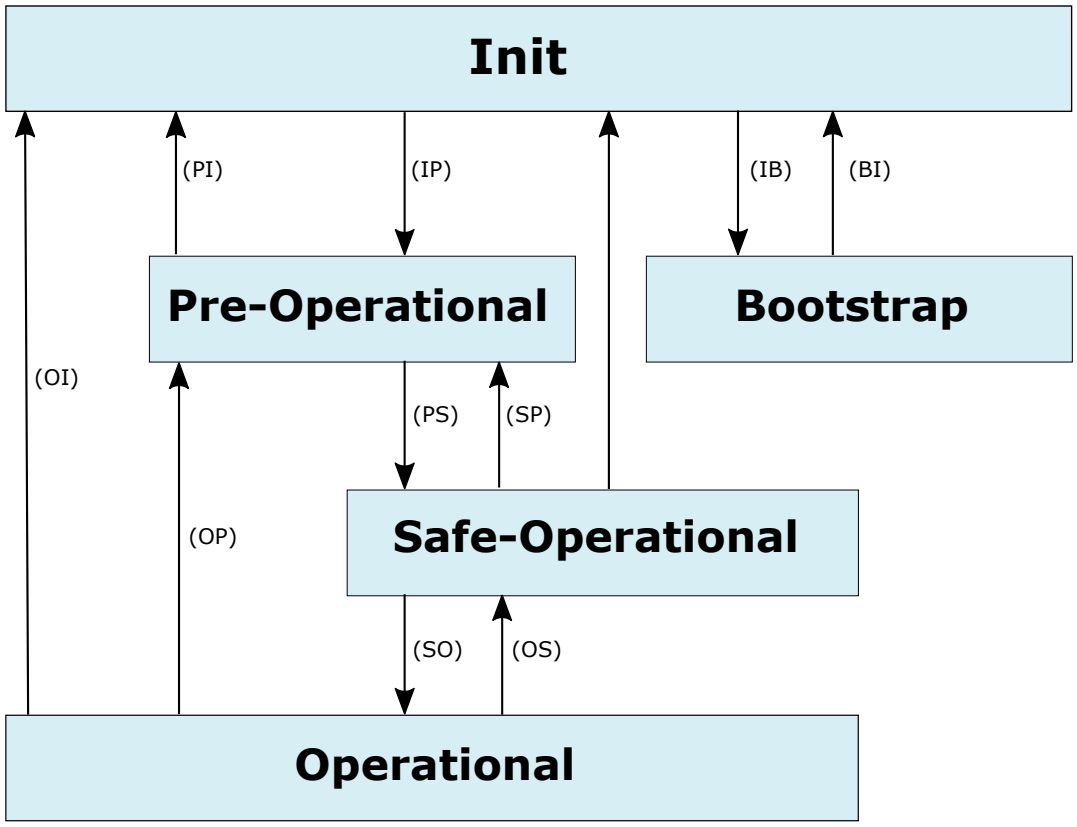


Figure 1: NMT State Machine

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.

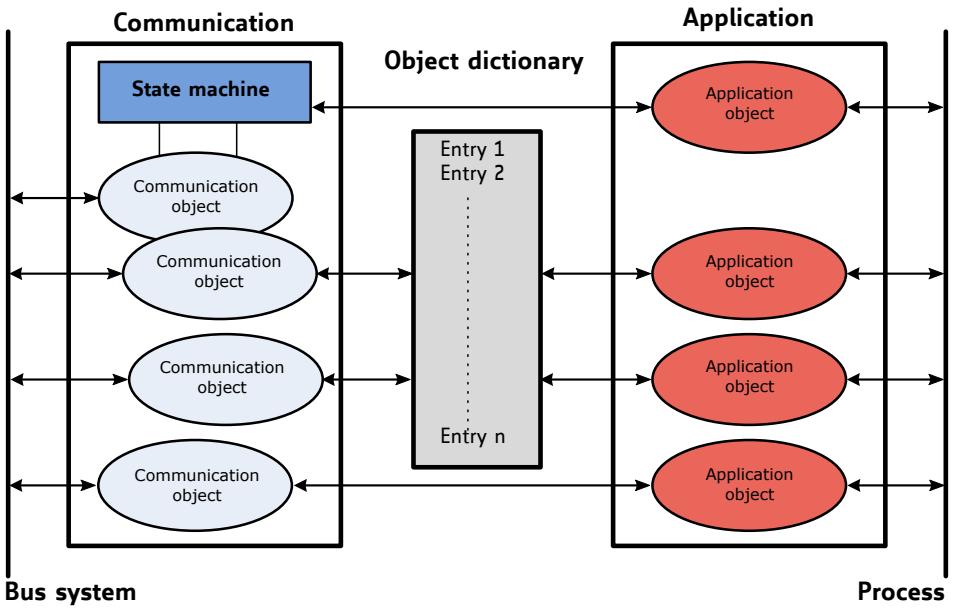


Figure 2: Device Model

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 PD60-4H-1461-CoE 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1000_h	Device Type	ro	no	UNSIGNED32	$0…2^{32}$-1	0	—

Table 5: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1001_h	Error Register	ro	no	UNSIGNED8	$0…2^{8}$-1	—	—

Table 6: Object 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 7: 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	Access	PDO Mapping	Data Type	Default Value
1008_h	Manufacturer Device Name	ro	no	Visible String	PD60-4H-1461-CoE

Table 8: Object Description (1008_h)

3.1.4 Object 1009_h: Manufacturer Hardware Version

This object contains the hardware version description.

Object Description					
Index	Name	Access	PDO Mapping	Data Type	Default Value
1009_h	Manufacturer Hardware Version	ro	no	Visible String	Depends on device, e.g. 1.00

Table 9: Object Description (1009_h)

3.1.5 Object $100A_h$: Manufacturer Software Version

This object contains the software version description.

Object Description					
Index	Name	Access	PDO Mapping	Data Type	Default Value
$100A_h$	Manufacturer Software Version	ro	no	Visible String	Depends on device, e.g. 1.00

Table 10: Object 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 PD60-4H-1461-CoE 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.

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 11).

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

Table 11: 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 12: Object Description (1010_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Save all parameters	rw	no	$0…2^{32}$-1	1	—
2	Save communication parameters	rw	no	$0…2^{32}$-1	1	—
3	Save device profile parameters	rw	no	$0…2^{32}$-1	0	—
4	Save motor 0 parameters	rw	no	$0…2^{32}$-1	1	—

Table 13: 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 PD60-4H-1461-CoE 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.

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 14).

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

Table 14: 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 PD60-4H-1461-CoE.

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

Table 15: Object Description (1011_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Restore all parameters	rw	no	$0…2^{32}$-1	—	—
2	Restore communication parameters	rw	no	$0…2^{32}$-1	—	—
3	Restore device profile parameters	rw	no	$0…2^{32}$-1	—	—
4	Restore motor 0 parameters	rw	no	$0…2^{32}$-1	—	—

Table 16: 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 Object Record

Table 17: Object Description (1018_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Vendor ID	ro	no	UNSIGNED32	$0...2^{32}$-1	0286_h	—
2	Product Code	ro	no	UNSIGNED32	$0...2^{32}$-1	1461	—
3	Revision Number	ro	no	UNSIGNED32	$0...2^{32}$-1	e.g. 20003_h for version 2.3	—

Table 18: 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 19: 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…5	Index 1600_h: 5
01_h	Mapping entry 1	rw	UNSIGNED32	60600008_h
02_h	Mapping entry 2	rw	UNSIGNED32	60400010_h
03_h	Mapping entry 3	rw	UNSIGNED32	$607A0020_h$
04_h	Mapping entry 4	rw	UNSIGNED32	$60FF0020_h$
05_h	Mapping entry 5	rw	UNSIGNED32	60710010_h

Table 20: 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 21: 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...8	8
01_h	Mapping entry 1	rw	UNSIGNED32	60610008_h
02_h	Mapping entry 2	rw	UNSIGNED32	60410010_h
03_h	Mapping entry 3	rw	UNSIGNED32	60620020_h
04_h	Mapping entry 4	rw	UNSIGNED32	60640020_h
05_h	Mapping entry 5	rw	UNSIGNED32	$606B0020_h$
06_h	Mapping entry 6	rw	UNSIGNED32	$606C0020_h$
07_h	Mapping entry 7	rw	UNSIGNED32	60740010_h
08_h	Mapping entry 8	rw	UNSIGNED32	60770010_h

Table 22: 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 23: Sync Manager Communication Types

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

Table 24: 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 25: 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 26: 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 27: 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 28: 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 29: 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 PD60-4H-1461-CoE.

4.1 Detailed Object Specifications

4.1.1 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2003_h	Maximum Current	rw	no	UNSIGNED32	0…60000	5000	mA

Table 30: Object Description (2003_h)

4.1.2 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2004_h	Open Loop Current	rw	no	UNSIGNED32	0…60000	3000	mA

Table 31: Object Description (2004_h)

4.1.3 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.

As the PD60-4H-1461-CoE drive does not have any limit switch inputs nor a home switch input, the default value of this object is 3. This internally disables any limit switch or home switch functionality. Do not change the setting of this object.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2005_h	Switch Parameters	rw	no	UNSIGNED32	0…63	0	—

Table 32: Object 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 33: Bit Definitions (2005_h)

4.1.4 Object 2041_h: Torque Mode Settings

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

Table 34: Object Description (2041_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Torque_P	rw	no	UNSIGNED16	$0…2^{16}$-1	200	—
2	Torque_I	rw	no	UNSIGNED16	$0…2^{16}$-1	15000	—
3	PI_Torque_Error	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	mA
4	PI_Torque_Error_Sum	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	—
5	PI_Flux_Error	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	mA
6	PI_Flux_Error_Sum	ro	no	SIGNED32	$-2^{31}…2^{31}$-1	—	—
7	PHI_E	ro	no	SIGNED16	$-2^{15}…2^{15}$-1	—	—

Table 35: Entry Description (2041_h)

Torque_P P parameter for the torque PI controller.

Torque_I I parameter for the torque PI controller.

4.1.5 Object 2042_h: Velocity Mode Settings

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

Table 36: Object Description (2042_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Velocity_P	rw	no	UNSIGNED16	0…2^{16}-1	100	—
2	Velocity_I	rw	no	UNSIGNED16	0…2^{16}-1	200	—
3	PI_Velocity_Error	ro	no	SIGNED32	-2^{31}…2^{31}-1	—	—
4	PI_Velocity_Error_Sum	ro	no	SIGNED32	-2^{31}…2^{31}-1	—	—
5	Sensor_Selection	ro	no	UNSIGNED8	0…2	0	—
6	Velocity_Filter	rw	no	UNSIGNED8	0…8	1	—
7	Velocity_Unit	rw	no	UNSIGNED8	0…1	0	—
8	Motor_Halted_Velocity	rw	no	UNSIGNED8	0…200000	0	—

Table 37: Entry Description (2042_h)

Velocity_P — P parameter for the velocity PI controller.

Velocity_I — I parameter for the velocity PI controller.

Sensor_Selection — Read only, as only option 0 (same as commutation) is usable with this drive.

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.

4.1.6 Object 2043_h: Position Mode Settings

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

Table 38: Object Description (2043_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Position_P	rw	no	UNSIGNED16	$0 \ldots 2^{16}-1$	20	—
2	PI_Position_Error	ro	no	SIGNED32	$-2^{31} \ldots 2^{31}-1$	—	—
3	AlwaysUseEncoder	ro	no	UNSIGNED8	0…1	0	—

Table 39: Entry Description (2043_h)

Position_P	P parameter for the velocity PI controller.
PI_Position_Error	Error of the position PI controller.
AlwaysUseEncoder	Read only, as only option 0 can be used with this drive.

4.1.7 Object 2055_h: Commutation Mode

Using this object the commutation mode can be set. In most cases, encoder based commutation (default value) should be used, but in some cases also open loop is an option. Setting this object to 0 completely disables the motor coils.

Commutation Modes	
0	Disabled
1	Open loop
3	Encoder

Table 40: Commutation Modes

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2055_h	Commutation Mode	rw	no	UNSIGNED8	0…3	3	—

Table 41: Object Description (2055_h)

4.1.8 Object 2057_h: Motor Shaft Direction

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

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2057_h	Motor Shaft Direction	rw	no	UNSIGNED8	0…1	1	—

Table 42: Object Description (2057_h)

4.1.9 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2058_h	Position Scaler	rw	no	SIGNED32	6…2147483647	65536	—

Table 43: Object Description (2058_h)

4.1.10 Object 2060_h: ADC Configuration

Using this object the ADC offsets for the coil current measurement can be configured. This is done during manufacturing of the drive, so there should normally be no need to change these settings. Use the calibration tool from Trinamic if re-calibration of the ADC offsets should ever be necessary.

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

Table 44: Object Description (2060_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	ADC_I0_Raw	ro	no	UNSIGNED16	0…2^{16}-1	—	—
2	ADC_I1_Raw	ro	no	UNSIGNED16	0…2^{16}-1	—	—
3	ADC_I0_Offset	rw	no	UNSIGNED16	0…2^{16}-1	—	—
4	ADC_I1_Offset	rw	no	UNSIGNED16	0…2^{16}-1	—	—
5	ADC_I0	ro	no	SIGNED16	-2^{15}…2^{15}-1	—	—
6	ADC_I1	ro	no	SIGNED16	-2^{15}…2^{15}-1	—	—
7	ADC_I2	ro	no	SIGNED16	-2^{15}…2^{15}-1	—	—

Table 45: Entry Description (2060_h)

4.1.11 Object 2080_h: ABN Encoder Settings

Using this object all necessary encoder parameters can be set. With the PD60-4H-1461-CoE and its built-in encoder the encoder direction and encoder steps settings are fixed and cannot be changed (read only). Only the encoder initialization mode can be chosen as needed. For most cases mode 0 (default setting) is recommended.

Encoder Initialization Modes	
0	Estimate offset
1	Use offset

Table 46: Encoder Initialization Modes

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

Table 47: Object Description (2080_h)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Direction	ro	no	UNSIGNED8	0…1	0	—
2	Steps	ro	no	UNSIGNED32	0…2^{24}-1	16384	—
3	Init_Mode	rw	no	UNSIGNED8	0…1	0	—

Table 48: Entry Description (2080_h)

4.1.12 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 49.

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 49: Motor Status Flags (2101_h)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2101_h	Motor Status Flags	ro	no	UNSIGNED32	0…$3FFFF_h$	0	—

Table 50: Object Description (2101_h)

4.1.13 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2102_h	Open Loop Commutation Angle	ro	no	SIGNED16	$-2^{15}...2^{15}-1$	—	—

Table 51: Object Description (2102_h)

4.1.14 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2103_h	Encoder Commutation Angle	ro	no	SIGNED16	$-2^{15}...2^{15}-1$	—	—

Table 52: Object Description (2103_h)

4.1.15 Object 2110_h: Driver Temperature

This object shows the temperature of the motor driver in °C.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2110_h	Driver Temperature	ro	no	SIGNED16	$-2^{15}...2^{15}-1$	—	—

Table 53: Object Description (2110_h)

4.1.16 Object 2140_h: Home Offset Display

This object shows the home offset. The unit of this value is defined by the position scaler (see 4.1.9).

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2140_h	Home Offset Display	ro	no	SIGNED32	$-2^{31}...2^{31}-1$	—	—

Table 54: Object Description (2140_h)

4.1.17 Object $21FE_h$: ADC Store

This object is to be used by Trinamic only for permantly storing ADC offsets during claibration and test of the drive. Do not change anything here. Use the calibration tool from Trinamic if re-calibration of the

ADCs should ever be necessary.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$21FE_h$	ADC Store	rw	no	UNSIGNED32	$0 \ldots FFFFFFFF_h$	0	—

Table 55: Object Description ($21FE_h$)

4.1.18 Object $21FF_h$: Encoder Calibration

This object is to be used by Trinamic only for encoder calibration during factory tests of the drive. Do not change anything here. Use the calibration tool from Trinamic if re-calibration of the encoder should ever be necessary.

Object Description			
Index	Name	Object Type	Data Type
$21FF_h$	Encoder Calibration	Variable	Array

Table 56: Object Description ($21FF_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Index	no	0	399	—		rw
2	Data	no	0	$FFFFFFFF_h$	—		rw
3	Magic	no	0	$FFFFFFFF_h$	—		rw
3	Values	no	0	$FFFFFFFF_h$	—		rw

Table 57: Entry Description ($21FF_h$)

4.1.19 Object $270E_h$: Analog Inputs

This object contains the supply voltage and the driver temperature as raw analog values. Its main use is for checking the module by Trinamic. Instead use the appropriate objects (2110_h and 2710_h) for reading these values.

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

Table 58: Object Description ($270E_h$)

Entry Description							
Sub-index	Name	PDO Mapping	Min	Max	Default	Unit	Access
1	Supply voltage	no	0	65535	0	raw	ro
2	Driver temperature	no	0	65535	0	raw	ro

Table 59: Entry Description ($270E_h$)

4.1.20 Object 2710_h: Supply Voltage

This object shows the supply voltage in units of 1/10 V. So for example a value of 237 means 23.7V.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
2710_h	Supply Voltage	ro	no	UNSIGNED16	$0…2^{16}$-1	—	—

Table 60: Object Description (2710_h)

5 Profile Specific Area

The profile segment contains CiA-402 standard motion control objects. These objects control the motion control functions of the PD60-4H-1461-CoE. 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, status-word, 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 PD60-4H-1461-CoE:

- 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 61: Value Description ($605A_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605A_h$	Quick Stop Option Code	rw	no	SIGNED16	1, 2, 5, 6	2	—

Table 62: Object 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 63: Value Description ($605B_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605B_h$	Shutdown Option Code	rw	no	SIGNED16	0	0	—

Table 64: Object 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 65: Value Description ($605C_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605C_h$	Disable Operation Option Code	rw	no	SIGNED16	1	1	—

Table 66: Object 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 67: Value Description ($605D_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605D_h$	Halt Option Code	rw	no	SIGNED16	1	1	—

Table 68: Object 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 69: Value Description ($605E_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$605E_h$	Fault Reaction Option Code	rw	no	SIGNED16	2	2	—

Table 70: Object 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 71: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6060_h	Modes of Operation	rw	yes	SIGNED8	see table 71	0	—

Table 72: Object 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 73: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6061_h	Modes of Operation Display	ro	yes	SIGNED8	see table 73	—	—

Table 74: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60FD_h$	Digital Inputs	ro	yes	UNSIGNED32	$0…2^{32}$-1	—	—

Table 75: Object 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 76: Value Definition (6502_h)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6502_h	Supported Drive Modes	ro	yes	UNSIGNED32	see table 76	-	—

Table 77: Object Description (6502_h)

5.1.10 Object $67FF_h$: Single Device Type

This object provides information on the device profile used for the individual axis. Its structure is similar to object 1000_h. The lower sixteen bits contain the device profile number which is always 402 (0192_h) with Trinamic motion control modules. The upper sixteen bits contain more information about the drive profile.

Value Definition	
Bit	Meaning
0…15	Device profile number
16	Frequency converter (0=no, 1=yes)
17	Servo drive (0=no, 1=yes)
18	Stepper motor (0=no, 1=yes)
19…21	Reserved
22	PDO set for generic drive device (0=yes, 1=no)
23	Multi device module (0=no, 1=yes)
24…31	Reserved

Table 78: Value Definition ($67FF_h$)

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$67ff_h$	Single Device Type	ro	yes	UNSIGNED32	see table 78	Depends on individual axis.	—

Table 79: Object Description ($67ff_h$)

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.

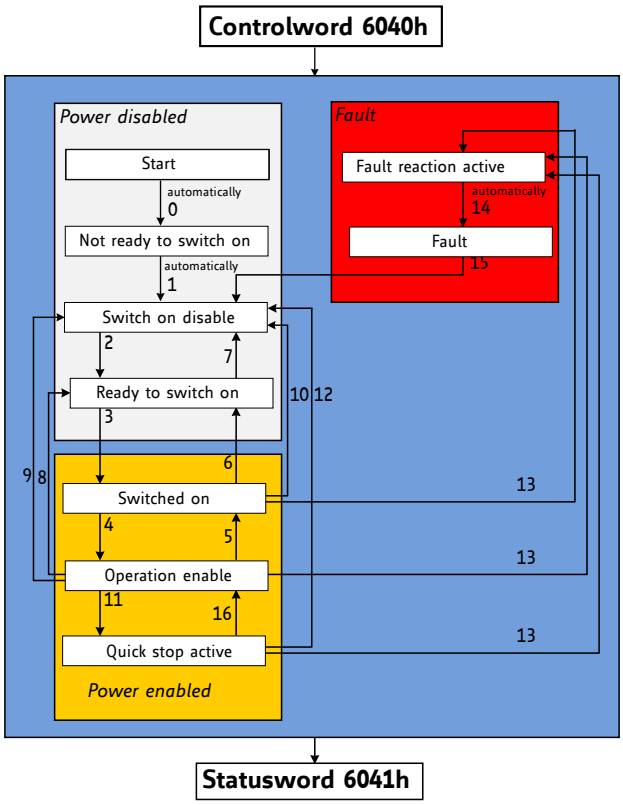


Figure 3: DS402 Finite State Machine

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: Controlword

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 Controlword										
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 80: Structure of the Controlword 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 81: Operation Mode specific Bits in pp Mode

Command Coding						
Command	Bits of Controlword					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 82: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 83: Object Description (6040_h)

6.1.2 Object 6041_h: Statusword

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 Statusword															
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 84: 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 85: 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 86: Operation Mode specific Bits in pp Mode

State Coding	
Statusword	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 87: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041$_h$	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 88: Object Description (6041$_h$)

6.1.3 Object 6062$_h$: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062$_h$	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 89: Object 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 microsteps. It is the same as object 6062$_h$.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 90: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 91: Object Description (6064_h)

6.1.6 Object 6065_h: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

When the difference between motor position (object 6062_h) and encoder position (object 6063_h or 6064_h) is greater than the value set here, the motor will be stopped and an emergency message will be sent. Setting this object to zero will turn off this feature completely.

Note Setting this object to a too low value will lead to false alarms.

Object Description			
Index	Name	Object Type	Data Type
6065_h	Following Error Window	Variable	UNSIGNED32

Table 92: Object Description (6065_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…2147483647	0

Table 93: Entry Description (6065_h)

6.1.7 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6067_h	Position Window	rw	no	UNSIGNED32	$0...2^{32}$-1	$FFFFFFFF_h$	—

Table 94: Object Description (6067_h)

6.1.8 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6068_h	Position Window Time	rw	no	UNSIGNED16	$0...2^{16}$-1	0	ms

Table 95: Object Description (6068_h)

6.1.9 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of pps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}...2^{31}$-1	—	—

Table 96: Object Description ($606C_h$)

6.1.10 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 microsteps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$607A_h$	Target Position	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 97: Object Description ($607A_h$)

6.1.11 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 98: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 99: Entry Description ($607D_h$)

6.1.12 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 pps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6081_h	Profile Velocity	rw	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 100: Object Description (6081_h)

6.1.13 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 in profile velocity mode. This value is given using pps^2 units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6083_h	Profile Acceleration	rw	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 101: Object Description (6083_h)

6.1.14 Object 6084_h: Profile Deceleration

This object indicates the configured deceleration. On the PD60-4H-1461-CoE module the deceleration ramp is always the same as the acceleration ramp. For this reason this object is a read-only object and always reads the same value as object 6083_h. This value is given in units of pps^2.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6084_h	Profile Deceleration	ro	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 102: Object Description (6084_h)

6.1.15 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6085_h	Quick Stop Deceleration	ro	yes	UNSIGNED32	$0…2^{32}$-1	0	—

Table 103: Object 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
- Profile deceleration
- Emergency stop
- Motion profile type

7.1 Detailed Object Specifications

7.1.1 Object 6040_h: Controlword

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 Controlword does not contain any operation mode specific bits.

Structure of the Controlword											
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 104: Structure of the Controlword in pv Mode

Command Coding						
Command	Bits of Controlword					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 105: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 106: Object Description (6040_h)

7.1.2 Object 6041_h: Statusword

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 Statusword															
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 107: Structure of the Statusword 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 108: 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 109: Operation Mode specific Bits in pv Mode

State Coding	
Statusword	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 110: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 111: Object Description (6041_h)

7.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062_h	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 112: Object Description (6062_h)

7.1.4 Object 6063_h: Position Actual Internal Value

This object provides the demanded position value. The value is given in microsteps. It is the same as object 6062_h.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 113: Object Description (6063_h)

7.1.5 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 114: Object Description (6064_h)

7.1.6 Object 6065_h: Following Error Window

This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients. The value shall be given in microsteps.

When the difference between motor position (object 6062_h) and encoder position (object 6063_h or 6064_h) is greater than the value set here, the motor will be stopped and an emergency message will be sent. Setting this object to zero will turn off this feature completely.

Note Setting this object to a too low value will lead to false alarms.

Object Description			
Index	Name	Object Type	Data Type
6065_h	Following Error Window	Variable	UNSIGNED32

Table 115: Object Description (6065_h)

Entry Description				
Sub-index	Access	PDO Mapping	Value Range	Default Value
0	rw	no	0…2147483647	0

Table 116: Entry Description (6065_h)

7.1.7 Object $606C_h$: Velocity Actual Value

This object shows the actual velocity value of the motor. The value is given in units of pps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}…2^{31}-1$	—	—

Table 117: Object Description ($606C_h$)

7.1.8 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 118: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 119: Entry Description ($607D_h$)

7.1.9 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 in profile velocity mode. This value is given using pps^2 units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6083_h	Profile Acceleration	rw	yes	UNSIGNED32	$0\ldots2^{32}-1$	0	—

Table 120: Object Description (6083_h)

7.1.10 Object 6084_h: Profile Deceleration

This object indicates the configured deceleration. On the PD60-4H-1461-CoE module the deceleration ramp is always the same as the acceleration ramp. For this reason this object is a read-only object and always reads the same value as object 6083_h. This value is given in units of pps^2.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6084_h	Profile Deceleration	ro	yes	UNSIGNED32	$0\ldots2^{32}-1$	0	—

Table 121: Object Description (6084_h)

7.1.11 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6085_h	Quick Stop Deceleration	ro	yes	UNSIGNED32	$0...2^{32}$-1	0	—

Table 122: Object Description (6085_h)

7.1.12 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 pps units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60FF_h$	Target Velocity	rw	yes	SIGNED32	$-2^{31}...2^{31}$-1	0	—

Table 123: Object Description ($60FF_h$)

7.2 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.

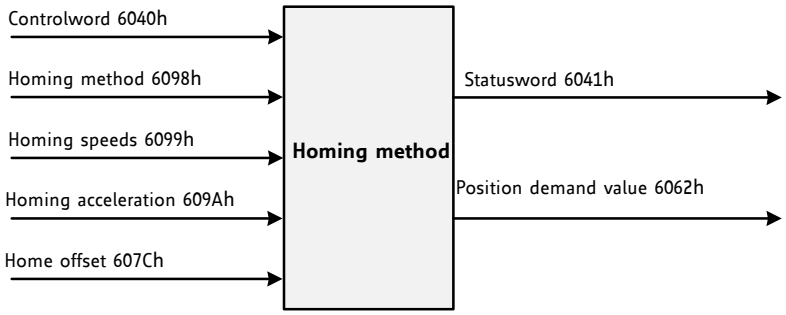


Figure 4: Homing Mode Function

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 PD60-4H-1461-CoE 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).
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 124: Supported CANopen Homing Methods

8.1.1 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: Controlword

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 Controlword										
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 125: Structure of the Controlword 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 126: Operation Mode specific Bits in hm Mode

Command Coding						
Command	Bits of Controlword					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 127: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 128: Object Description (6040_h)

8.2.2 Object 6041_h: Statusword

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 Statusword															
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 129: Structure of the Statusword 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 130: 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 131: Operation Mode specific Bits in hm Mode

State Coding	
Statusword	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 132: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041$_h$	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 133: Object Description (6041$_h$)

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 pps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
606C$_h$	Velocity Actual Value	ro	yes	SIGNED32	-2^{31}...2^{31}-1	—	—

Table 134: Object 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 microsteps. Negative values indicate the opposite direction.

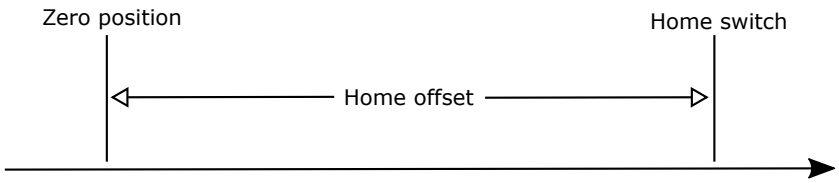


Figure 5: Home Offset

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$607C_h$	Home Offset	rw	no	SIGNED32	$-2^{31}...2^{31}-1$	0	—

Table 135: Object 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 124 for a list of homing methods supported by the current version of the PD60-4H-1461-CoE CANopen firmware.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6098_h	Homing Method	rw	yes	SIGNED8	See table 124	0	—

Table 136: Object 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 pps 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 137: Object Description (6099_h)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Fast Homing Speed 1	rw	no	0…2^{32}-1	0	—
2	Slow Homing Speed	rw	no	0…2^{32}-1	0	—

Table 138: 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 pps^2 units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$609A_h$	Homing Acceleration	ro	no	UNSIGNED32	0…2^{32}-1	0	—

Table 139: Object 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: Controlword

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 Controlword.

Structure of the Controlword									
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 140: Structure of the Controlword in csp Mode

Command Coding						
Command	Bits of Controlword					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 141: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 142: Object Description (6040_h)

9.1.2 Object 6041_h: Statusword

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 Statusword															
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 143: Structure of the Statusword 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 144: 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 145: Operation Mode specific Bits in csp Mode

State Coding	
Statusword	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 146: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 147: Object Description (6041_h)

9.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062_h	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 148: Object 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 microsteps. It is the same as object 6062_h.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 149: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 150: Object 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 pps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$606C_h$	Velocity Actual Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 151: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
607A$_h$	Target Position	rw	yes	SIGNED32	-2^{31}…2^{31}-1	0	—

Table 152: Object Description (607A$_h$)

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 153: Object Description (607D$_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	-2^{31}…2^{31}-1	-2^{31}	—
2	Maximum Position Limit	rw	no	-2^{31}…2^{31}-1	2^{31}-1	—

Table 154: 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 microsteps and will be added to the target position.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60B0_h$	Position Offset	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 155: Object 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 156: Object Description ($60C2_h$)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Interpolation time period value	rw	no	UNSIGNED8	$0\ldots2^8-1$	1	—
2	Interpolation time index	rw	no	SIGNED8	-3…3	-3	—

Table 157: Entry Description (2043_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: Controlword

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 Controlword.

Structure of the Controlword									
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 158: Structure of the Controlword in csv Mode

Command Coding						
Command	Bits of Controlword					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 159: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 160: Object Description (6040_h)

10.1.2 Object 6041_h: Statusword

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 Statusword															
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 161: Structure of the Statusword 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 162: 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 163: Operation Mode specific Bits in csv Mode

State Coding	
Statusword	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 164: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 165: Object Description (6041_h)

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 pps.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
606C$_h$	Velocity Actual Value	ro	yes	SIGNED32	-2^{31}...2^{31}-1	—	—

Table 166: Object 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 pps units.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
60FF$_h$	Target Velocity	rw	yes	SIGNED32	-2^{31}...2^{31}-1	0	—

Table 167: Object 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 168: Object Description (607D$_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}\ldots2^{31}-1$	$2^{31}-1$	—

Table 169: 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
$60B1_h$	Target Offset	rw	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	0	—

Table 170: Object 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 171: Object Description ($60C2_h$)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Interpolation time period value	rw	no	UNSIGNED8	$0\ldots2^{8}-1$	1	—
2	Interpolation time index	rw	no	SIGNED8	-3…3	-3	—

Table 172: Entry Description (2043_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: Controlword

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 Controlword.

Structure of the Controlword									
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 173: Structure of the Controlword in cst Mode

Command Coding						
Command	Bits of Controlword					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 174: Command Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6040_h	Controlword	rw	yes	UNSIGNED16	See command coding above.	0	—

Table 175: Object Description (6040_h)

11.1.2 Object 6041_h: Statusword

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 Statusword															
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 176: Structure of the Statusword 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 177: 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 178: Operation Mode specific Bits in cst Mode

State Coding	
Statusword	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 179: State Coding

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6041_h	Statusword	ro	yes	UNSIGNED16	See state coding above.	—	—

Table 180: Object Description (6041_h)

11.1.3 Object 6062_h: Position Demand Value

This object provides the demanded position value. The value is given in microsteps. 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6062_h	Position Demand Value	ro	yes	SIGNED32	$-2^{31}\ldots2^{31}-1$	—	—

Table 181: Object 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 microsteps. It is the same as object 6062_h.

Object Description							
Index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6063_h	Position Actual Internal Value	ro	yes	SIGNED32	$-2^{31}...2^{31}-1$	—	—

Table 182: Object 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	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
6064_h	Position Actual Value	ro	yes	SIGNED32	$-2^{31}...2^{31}-1$	—	—

Table 183: Object 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 184: Object Description (6071_h)

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

Table 185: 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 186: Object Description (6077_h)

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

Table 187: 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 188: Object Description ($607D_h$)

Entry Description						
Sub-index	Name	Access	PDO Mapping	Value Range	Default Value	Unit
1	Minimum Position Limit	rw	no	$-2^{31}…2^{31}-1$	-2^{31}	—
2	Maximum Position Limit	rw	no	$-2^{31}…2^{31}-1$	$2^{31}-1$	—

Table 189: 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 190: Object Description ($60B2_h$)

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

Table 191: 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 192: Object Description ($60C2_h$)

Entry Description							
Sub-index	Name	Access	PDO Mapping	Data Type	Value Range	Default Value	Unit
1	Interpolation time period value	rw	no	UNSIGNED8	$0…2^8$-1	1	—
2	Interpolation time index	rw	no	SIGNED8	-3…3	-3	—

Table 193: Entry Description (2043_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 PD60-4H-1461-CoE						
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 194: 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 PD60-4H-1461-CoE 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 195: SDO Abort Codes

14 Figures Index

15 Tables Index

16 Supplemental Directives

16.1 Producer Information

16.2 Copyright

16.3 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.

16.4 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.

16.5 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.

16.6 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.

16.7 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.

17 Revision History

17.1 Firmware Revision

Version	Date	Author	Description
V1.00	2022-SEP-15	ED/OK	First release.

Table 196: Firmware Revision

17.2 Document Revision

Version	Date	Author	Description
V1.00	2022-SEP-15	OK	First release.

Table 197: Document Revision