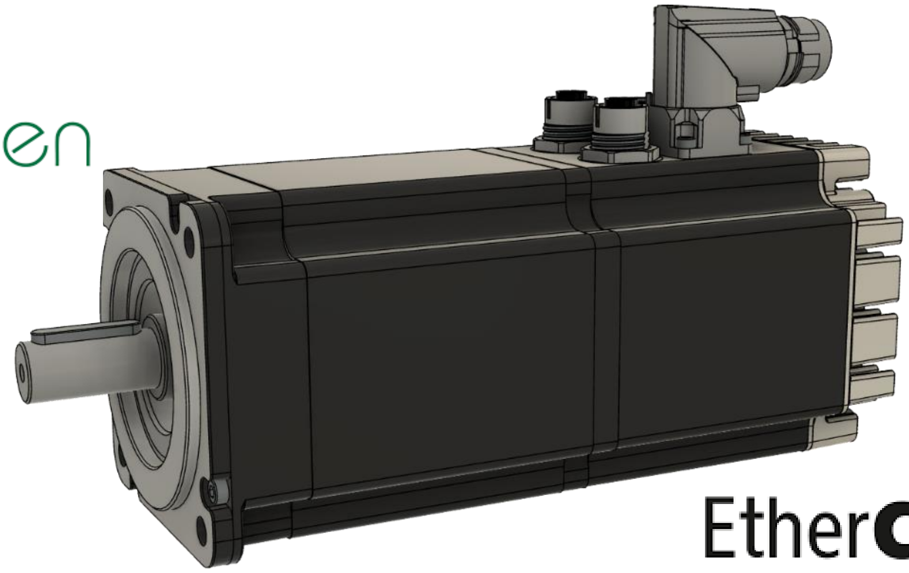
EtherCAT – CANopen Reference Manual

For DUET AD



Motor Power Company s.r.l.
Via Leonardo Da Vinci, 4
42024 Castelnovo Sotto Reggio Emilia - Italia
Tel. +39 0522 682710 - Fax +39 0522 683552
info@motorpowerco.it - motorpowerco.com
Cap. Soc. 250.000,00€ i.v. - R.E.A. di RE 175521
Iscr.Reg.Impr. di RE n.01308390358 - N. Mecc. RE 010210
C.F. e P.IVA IT 01308390358

1. Index

Notice:

- This guide is delivered subject to the following conditions and restrictions: This guide contains proprietary information belonging to Motor power Company Srl. Such information is supplied solely for the purpose of assisting users of the MPC Devices and Servo drive in its installation and configuration.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Motor power Company and the Motor power Company logo are trademarks of Motor power Company Srl.
- Information in this document is subject to change without notice.
-

CANopen Compatibility

This document is compatible with the following FW and EDS version and superior

EDS Version	***FW version***	***Note***
1.3	*1.2.X*	

EtherCAT Compatibility

XML Version	***FW version***	***Note***
1.5	*1.2.X*	

Document Versioning

Version	***Date***	***Author***	***Note***
1.0	*04/10/2022*	*Michele Piacentini*	*First Release*
1.1	*14/10/2022*	*Michele Piacentini* *Anouar Irhlam*	*FoE Addedd*
1.2	*1/06/2023*	*Michele Piacentini*	*Revision*

2. Introduction

Scope

This manual describes, in detail, the CANopen and EtherCAT implementation of Motor Power Company Duet AD

CANopen

CANopen is a standard established by the association "CAN in Automation". A great number of device manufacturers are organized in this association. This standard has replaced most of all manufacturer-specific CAN protocols, now a manufacturer independent communication interface is available for the user:

CiA Draft Standard 201...207: In these standards the general network administration and the transfer of objects are determined. This book is rather comprehensive. The relevant aspects are treated in the CANopen manual in hand so that it is not necessary in general to acquire the DS201..207.

CiA Draft Standard 301: In this standard the basic structure of the object dictionary of a CANopen device and the access to this directory are described. Besides this the statements made in the DS201..207 are described in detail. The elements needed for the the object directory and the access methods which belong to them are described in the present manual. It is advisable to acquire the DS301 but not necessary.

CiA Draft Standard 402: This standard describes the concrete implementation of CANopen in servo controllers. Though all implemented objects are also briefly documented and described in this CANopen manual the user should own this book.

Any information on CANopen protocol can be found at http://www.can-cia.de.

EtherCAT

EtherCAT is a high-performance, low-cost, easy to use Industrial Ethernet technology with a flexible topology.

With the CoE protocol, EtherCAT provides the same communication mechanisms as in CANopen®-Standard EN 50325-4: Object Dictionary, PDO Mapping (Process Data Objects) and SDO (Service Data Objects). The device profiles that is CiA 402, it has also been reused for EtherCAT.

DuetAD EtherCAT version supports:

- **CoE (CANOpen over EtherCAT)**
- **FoE (File access over EtherCAT**

Cabling and pin assignment CANopen

Follow the installation Guide of each device to find out the correct pinout configuration.

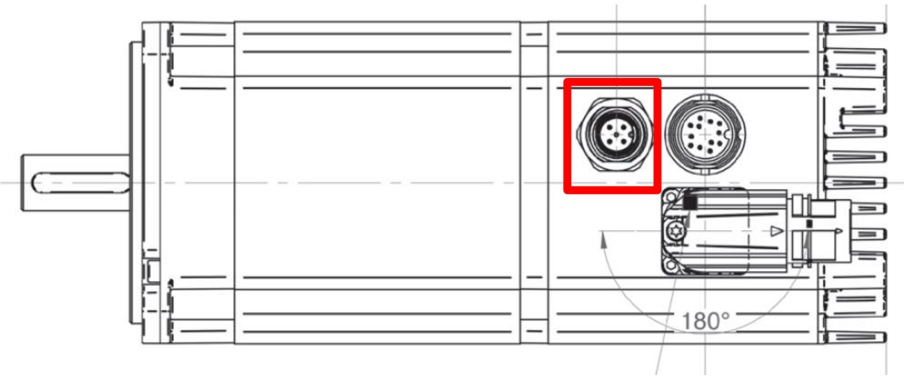
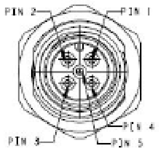


Figure 1 Duet AD CANopen version – CAN M12 position

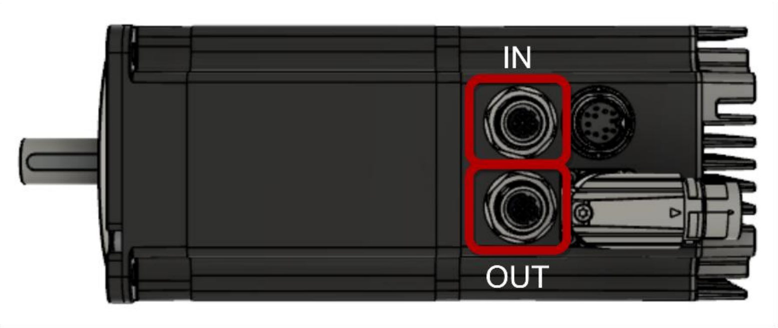


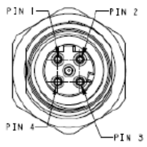
N° PIN	CN M12 5p Male Type A *CAN-OPEN Option*
1	n.c.
2	n.c.
3	CAN-GND
4	CAN-HIGH
5	CAN-LOW

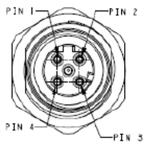
Please carefully respect the following information and notes for the cabling of the controller to get a stable and undisturbed communication system. A non-professional cabling can cause malfunctions of the CAN bus which hence the controller to shut down with an error.

Cabling and pin assignment EtherCAT

Follow the installation Guide of each device to find out the correct pinout configuration.







N° PIN	CN M12 4p Female D Type IN & OUT Connectors *ETHERCAT Option*
1	TD+
2	RD+
3	TD-
4	RD-

⚠ Please carefully respect the following information and notes for the cabling of the controller to get a stable and undisturbed communication system. A non-professional cabling can cause malfunctions of the EtherCAT bus which hence the controller to shut down with an error.

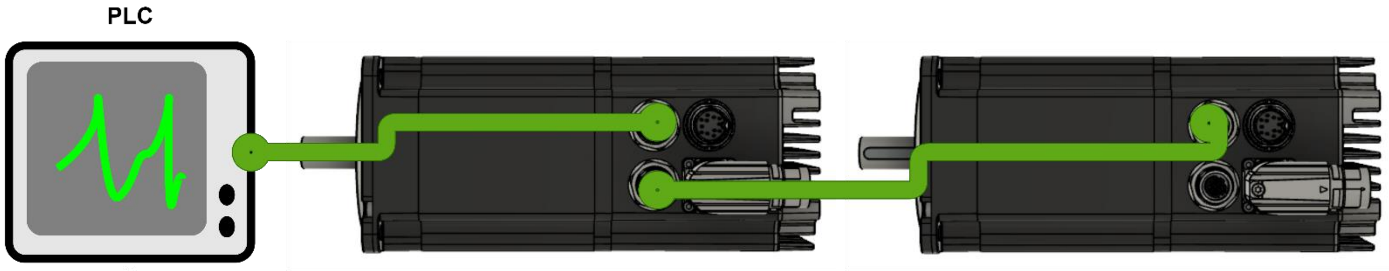


Figure 2 Typical connection for EtherCAT

No terminator resistor is required for EtherCAT drives.

Cabling Hints CANopen

The CAN bus offers an easy and safe way to connect all parts of a plant. As condition all following instructions must be respected carefully.

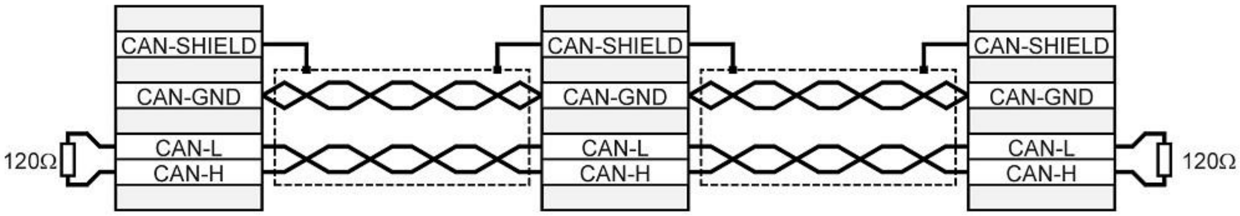


Figure 3 Can Bus Cabling

All nodes of a network are principally connected in series, so that the CAN cable is looped through all controllers.

The two ends of the CAN cable must be terminated by a resistor of 120Ω +/- 5%. Please note that such a resistor is often already installed in CAN cards or the PLC.

For cabling shielded cable with exactly two twisted pairs must be used. ➔ One twisted pair is used for CAN-H and CAN-L.

- One twisted pair is used commonly for CAN-GND.
- The shield of the cable is connected to CAN-SHIELD at all nodes.

A table with technical data of suitable cables can be found at the end of this chapter.

Recommended cables can be found in the product manual.

We dissuade from using connectors in between the CAN bus line. If it is still necessary to use connectors, assure that the connection of the shield is done by using metallic cases.

For less noise injection principally

- never place motor cables parallel to signal cables.
- use only motor cables specified by Motor Power Company.
- shield and earth motor cables correctly.

For further information refer to the Controller Area Network protocol specification, Ver.

2.0, Robert Bosch GmbH, 1991.

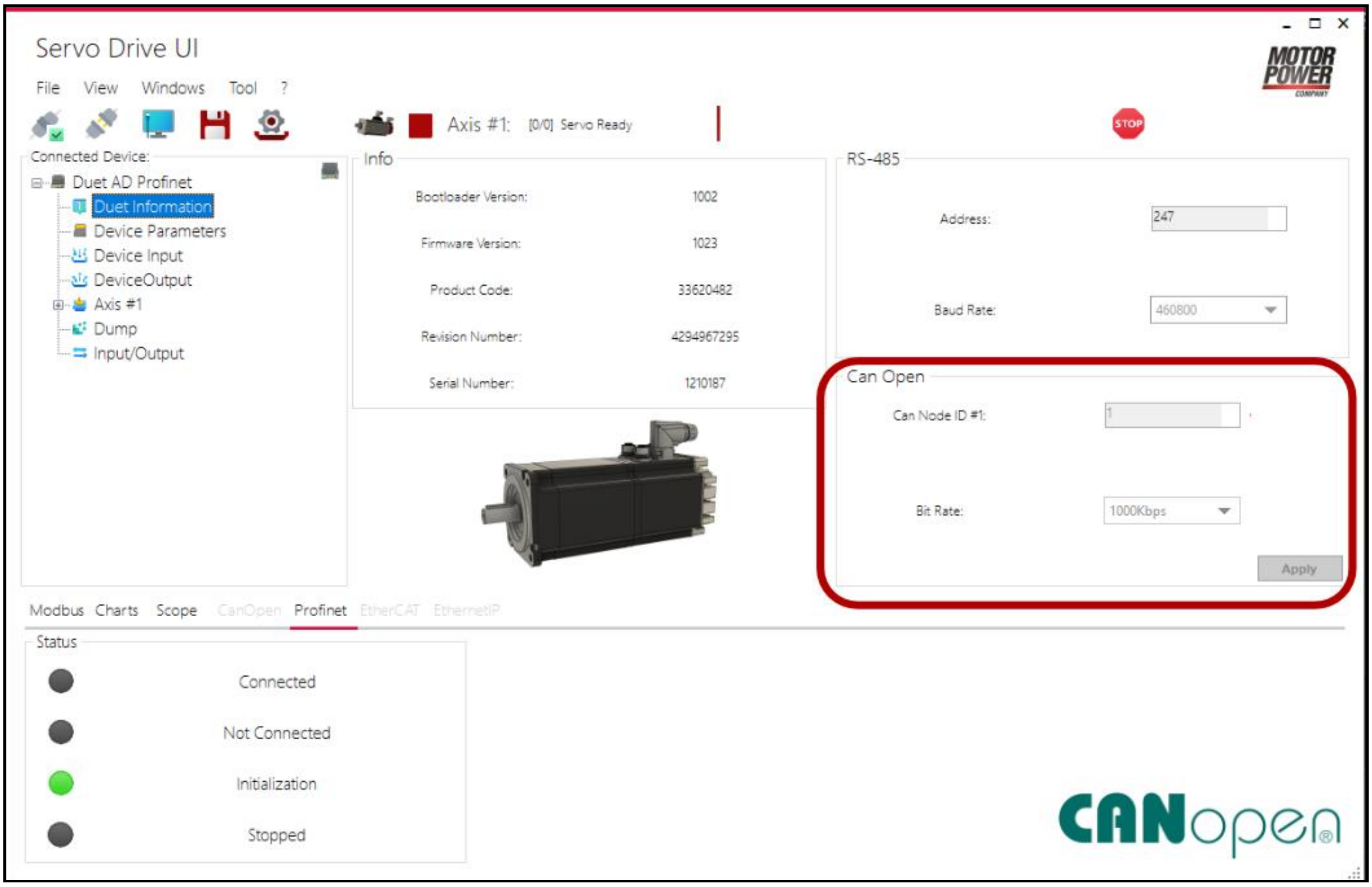
Technical data CAN bus cable: 2 twisted pairs, d ≥ 0,22 mm2 shielded loop resistance < 0,2 Ω/m char. impedance 100-120 Ω

⚠	120Ω Termination resistor No termination resistor is integrated in Motor Power Company Devices.

Node address in CANopen network

An unique node address (1 to 127) is settable with UI interface with following procedure:

- Connect to the UI interface via RS485
- Open generic parameters or DUET AD tab window and write the CANopen node ID



Axis #1: [0/0] Servo Ready

Bootloader Version: 1002
Firmware Version: 1023
Product Code: 33620482
Revision Number: 4294967295
Serial Number: 1210187

Address: 247
Baud Rate: 460800

Can Node ID #1: 1
Bit Rate: 1000Kbps

● Connected
● Not Connected
● Initialization
● Stopped

Figure 4-UI Interface

- Then save to EEPROM and reset/switch off and on. At restart new Node ID is set

NOTE: Default node ID is 1

Node address in CANopen network

Within an EtherCAT network, a physical node address (identification number) does not have to be specifically allocated to a device; the EtherCAT controller assign the address.

Bit Rate in CAN Network

Changing the baud rate follow the same procedure as for changing Node ID, except for the parameter that is CANopen Bit Rate. Allowed bit rate are:

Bit Rate	Parameter Value
1MB	0
500kbps	1
250kbps	2
125kbps	3

NOTE: Default Bit rate is 1MB

It is possible to change the bit rate for the node with following procedure:

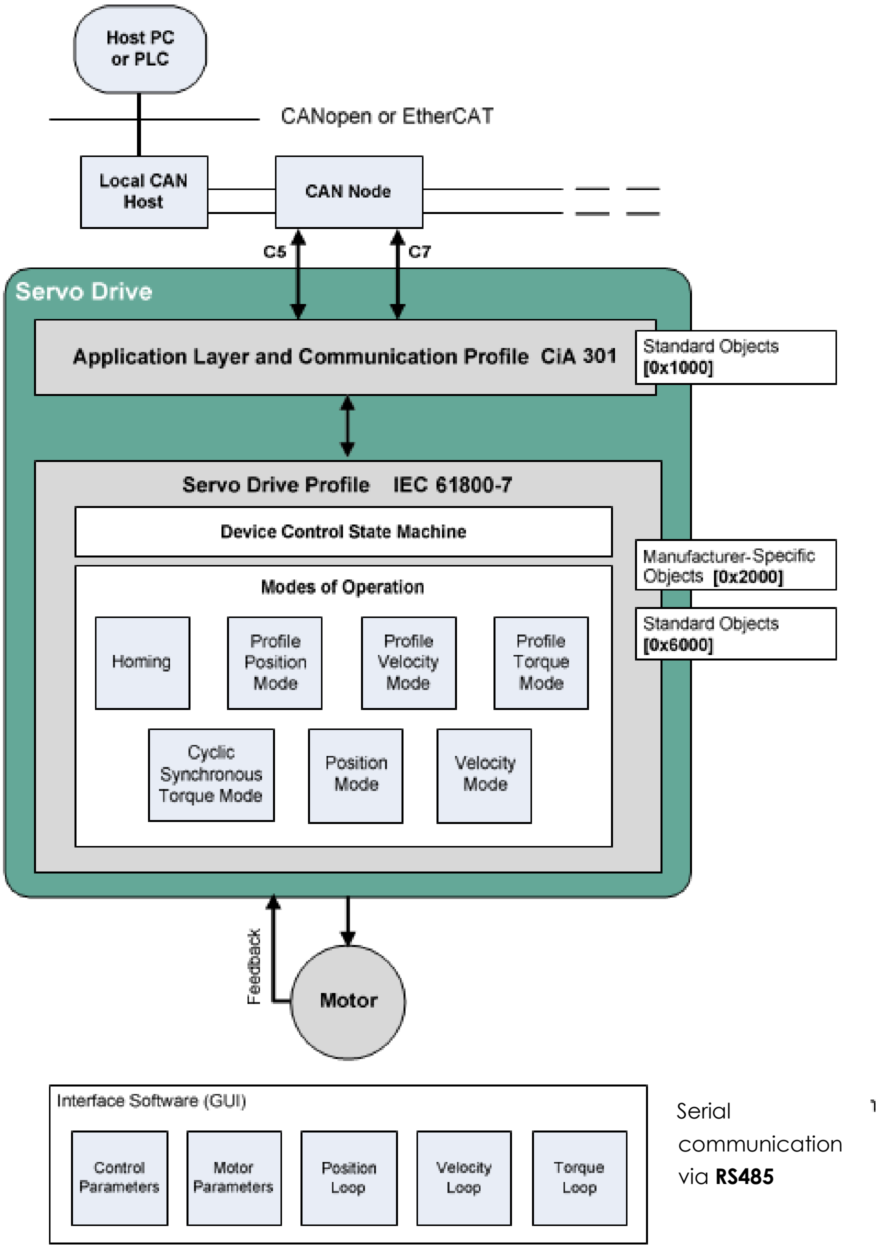
- Connect to the UI interface via RS485
- Open generic parameters window and write the CANopen bit rate as according to the previous table
- Then save to EEPROM and reset/switch off and on. At restart bit rate is set.

3. CANopen Operation

Device communication

The DuetAd conforms the following standards:

- CiA 301: CANopen Application Layer And Communication Profile
- CiA 402-1: General Definitions
- CiA 402-2: Operation Modes and Application Data
- CiA 402-3: PDO Mapping



Device Control: Starting and stopping of the drive and several mode-specific commands are executed by the state machine.

Modes of Operation: The operating mode defines the behavior of the drive.

Figura 1- Communication Model

Communication Objects

Communication objects are used for exchanging process and service data, for process or system time synchronization, for error state supervision, and for control and monitoring of node states. These objects are defined by their structure, transmission types and their CAN identifier.

Service Data Communication

Service data objects (SDOs) provide direct access to object entries in the CANopen device object dictionary. As these object entries contain data of arbitrary size and data type, the SDOs are used to transfer multiple data sets (each containing an arbitrary large block of data) from a client to a server and vice versa. The client controls, via a multiplexer (index and sub-index of the object dictionary), which data set is transferred. The content of the data set is defined within the object dictionary.

In general, an SDO is transferred as a sequence of segments. Prior to transferring the segments there is an initialization phase in which client and server prepare for transferring the segments. For SDOs, it is also possible to transfer a data set of up to four bytes during the initialization phase. This mechanism is called SDO expedited transfer.

The client always initiates an SDO transfer for any type of transfer. The owner of the accessed object dictionary is the server of the SDO. Either the client or the server can take the initiative to abort the transfer of an SDO.

By means of an SDO, a peer-to-peer communication channel between two CANopen devices is established. A CANopen device supports more than one SDO. One supported Server-SDO is the default case (Default SDO).

Process Data Communication

Process data objects (PDOs) perform real-time data transfer. The transfer of PDOs is performed without any protocol overhead.

The PDOs correspond to objects in the object dictionary and provide the interface to the application objects. Data type and mapping of application objects into a PDO is determined by a corresponding default PDO mapping structure within the object dictionary. DuetAD supports variable PDO mapping; therefore, the number of PDOs and the mapping of application objects into a PDO may be transmitted to a CANopen device during the configuration process, by applying the SDO services to the corresponding objects of the object dictionary.

PDOs are used for both data transmission and data reception – termed Transmit-PDO (TPDO) and Receive-PDO (RPDO), respectively. CANopen devices

supporting TPDO are PDO producers, and CANopen devices supporting RPDO are called PDO consumers. Duet AD supports both. The PDO communication parameter describes the communication capabilities of the PDO. The PDO mapping parameter contains information about the contents of the PDO.

For each PDO, a pair of communication and mapping parameters is mandatory.

- CANopen by default has 4 TPDOs and 4 RPDOs (8 bytes each). Mapping and configuration are active.

TPDO 1: Async
Position Actual value (6064h) 32 bits
StatusWord (6041h), 16 bits

TPDO 2: Cyclic Sync
Position Actual value (6064h) 32 bits
StatusWord (6041h), 16 bits

TPDO 3: Async
Velocity Actual value (606Ch), 16 bits
StatusWord (6041h), 16 bits

TPDO 4: Async - Not enabled

RPDO 1 Async
Target Position (607Ah) 32 bits
ControlWord (6040h), 16 bits

RPDO 2 Cyclic Sync
Param of IP function (60C1h:01), 16 bits
ControlWord (6040h), 16 bits

RPDO 3 Async
Target Velocity (60FFh) 32 bits
ControlWord (6040h), 16 bits

RPDO 4: Async - Not enabled

- **EtherCAT CoE by default has 2 TPDOs and 2 RPDOs** **<u>(Each limited to maximum 8 bytes</u>**) Mapping and configuration are active.

TPDO 1
StatusWord (6041h), 16 bits
Position Actual value (6064h) 32 bits
Torque actual value (6077h) 32 bits

TPDO 2
Velocity Actual value (606Ch), 16 bits
Digital Inputs (60FDh) 32 bits

RPDO 1
ControlWord (6040h), 16 bits
Target Position (607Ah) 32 bits
Max Current (6073h) 16 bits

RPDO 2
Target Torque (6081h), 16 bits
Profile Velocity (6081h) 32 bits

Device Control and State Machine

The power drive system finite-state automaton (PDS FSA) is a mathematical model that defines the behavior of the power drive system. Because a power drive system is required to provide local control even when the communication network is not functioning properly, the communication FSA and the PDS FSA are only loosely coupled. Figure shows how the power drive system operates remotely via the network, or locally.

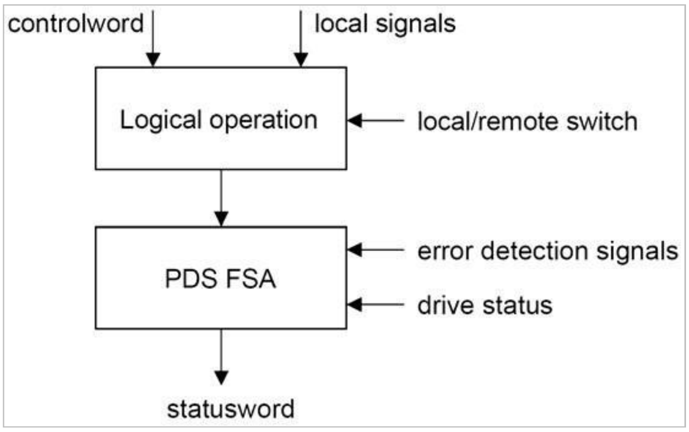


Figura 2 - Remote and local control

The power drive system is operated by the Controlword sent by the control device via the network. The state of the power drive system is reported by the Statusword produced by the drive device. The FSA is also controlled by error detection signals.

The PDS FSA defines the power drive system status and the possible control sequence of the power drive system. A single state represents a special internal or external behavior. The state of the power drive system also determines which commands are accepted. For example, it is only possible to start a point-to-point move when the drive is in the operation enabled state.

Indicating the Operating State

After switching on, and when an operating mode is started, the power drive system goes through a number of operating states. The operating states are internally monitored and influenced by monitoring functions

Figure 7-3 illustrates the PDS FSA behavior. It takes into consideration the control of the power electronics, in accordance with user commands and internal drive faults.

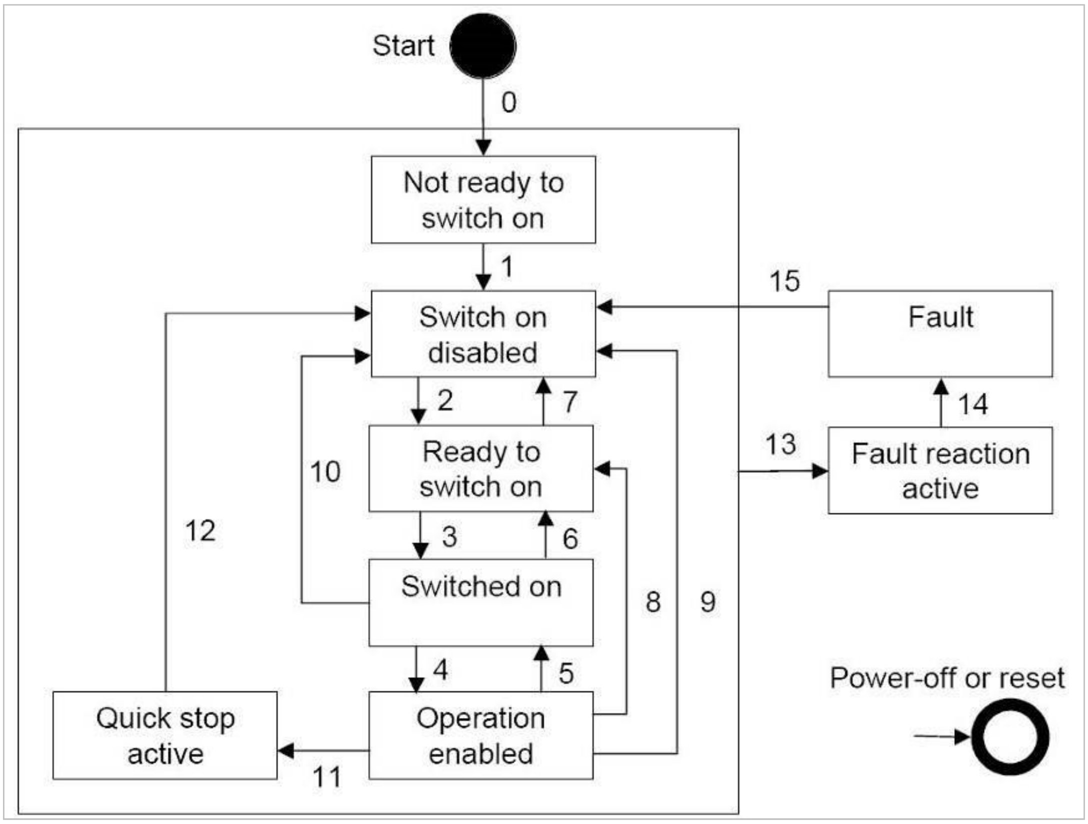


Figura 3 - Power Drive System State Diagram

Notes:

Not Ready to Switch On	"Not ready to operate" received from the controller.
Switch On Disabled	Ready to operate. Can read and write parameters. Motion functionality cannot be executed.
Ready to Switch On	Ready to operate. Can read and write parameters. Motion functionality cannot be executed. Bus voltage must be switched on.
Operation Enabled	Drive power stage is enabled. No fault is present. Motion functionality can be executed.
Quick Stop Active	Drive was stopped using controlled stop. Power stage is enabled. Motion functionality cannot be executed.
Fault Reaction Active	A fault has occurred. Drive is in the process of ramping down to 0 velocity (Active Disable process).
Fault	A fault has occurred. Power stage is disabled.

Bits 0, 1, 2, 3, 5 and 6 of the parameter Statusword provide information on the operating state.

	Operating State	Bit 6: Switch On Disabled	Bit 5: Quick Stop	Bit 3: Fault	Bit 2: Operation Enabled	Bit 1: Switch On	Bit 0: Ready to Switch On
2	Not Ready To Switch On	0	X	0	0	0	0
3	Switch On Disabled	1	X	0	0	0	0
4	Ready To Switch On	0	1	0	0	0	1
5	Switched On	0	1	0	0	1	1
6	Operation Enabled	0	1	0	1	1	1
7	Quick Stop Active	0	0	0	1	1	1
8	Fault Reaction Active	0	X	1	1	1	1
9	Fault	0	X	1	0	0	0

Parameter Name	Bit Assignments	Data Type R/W
Statusword	Bits 0–3 = Status bits	Unsigned16
	Bit 4 = Voltage enabled	Read Only
	Bits 5–6 = Status bits	
	Bit 7 = Warning	
	Bit 8 = Reserved	
	Bit 9 = Remote	
	Bit 10 = Target reached	
	Bit 11 = Internal limit is active	
	Bit 12 = Operating mode-specific	
	Bit 13 = Operating mode-specific	
	Bit 14 = Manufacturer-specific	
	Bit 15 = Manufacturer-specific	

Notes:

Bit 4	Bit 4=1 indicates whether the DC bus voltage is correct. If the voltage is missing or is too low, the device does not transition from operating state 3 to operating state 4.
Bit 7	If bit 7 (warning) of the status word is 1, it indicates the presence of a warning condition. Warning is not an error or fault (e.g., temperature limit exceeded, job refused). The status of the PDS FSA does not change. The cause of the warning may be given in the fault code parameter object (603Fh).

Bit 9	If bit 9 is set, the device carries out commands via the fieldbus. If Bit 9 is reset, the device is controlled via a different interface. In such a case, it is still possible to read or write parameters via the fieldbus.
Bit 10	Bit 10 is used for monitoring the current operating mode.
Bit 12	Bit 12 is used for monitoring the current operating mode.
Bit 13	Bit 13 only becomes 1 if an error needs to be resolved prior to further processing.

Changing the operating state

The parameter Controlword can be used to switch between operating states.

Parameter Name	**Bit Assignments**	**Data Type R/W**
Controlword	Bit 0 = Switch On	Unsigned16
	Bit 1 = Enable Voltage	Read Only
	Bit 2 = Quick Stop	
	Bit 3 = Enable Operation	
	Bits 4–6 = Operating Mode specific	
	Bit 7 = Fault Reset	
	Bit 8 = Halt	
	Bit 9 = Reserved	
	Bits 10–15 = Reserved (must be 0)	
	Changed settings become active immediately.	

Bits 0, 1, 2, 3 and 7 of the parameter Controlword allow you to switch between the operating states.

Fieldbus Command	**State Transitions**	**State Transition To**	**Bit 7: Fault Reset**	**Bit 3: Enable Operate**	**Bit 2: Quick Stop**	**Bit 1: Enable Voltage**	**Bit 0: Switch On**
Shutdown	T2, T6, T8	4 – Ready To Switch On	X	X	1	1	0
Switch On	T3	5 – Switched On	X	X	1	1	1
Disable Voltage	T7, T9, T10, T12	3 – Switch On Disabled	X	X	X	0	X

Quick Stop	T7, T10 T11	3 – Switch On Disabled 7 – Quick Stop Active	X	X	0	1	X
Disable Operation	T5	5 – Switched On	X	0	1	1	1
Enable Operation	T4, T16	6 – Operation Enabled	X	1	1	1	1
Fault Reset	T15	3 – Switch On Disabled	0 » 1	X	X	X	X

Notes:

Bit 4—6	Bits 4 to 6 are used for the operating mode-specific settings.
Bit 8	A Halt can be triggered with bit 8=1.
Bit 9–15	Reserved.

Starting and Changing an Operating Mode

The parameter Mode of Operation (6060h) is used to set the desired operating mode.

Parameter Name	**Description**	**Data Type R/W**
Mode of operation	Operating mode 1 Profile Position	Integer8 Read/Write
	3 Profile Velocity	
	4 Profile Torque	
	6 Homing	
	7 Interpolated Position	
	8 Cyclic Synchronous Position	
	Changed settings become active immediately.	

The parameter Mode of operation display (6061h) can be used to read the current operating mode.

Profile Position Mode

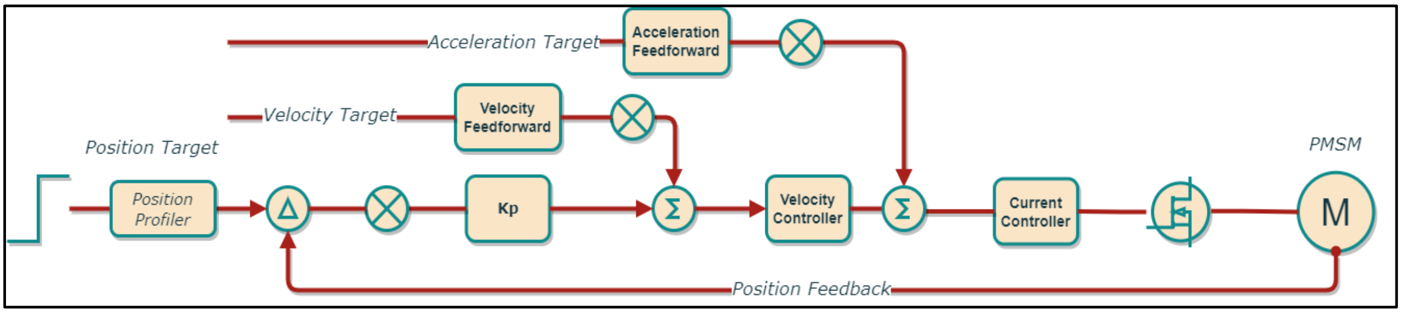


Figure 5 Position Controller

Description

In the operating mode Profile Position, a movement to a desired target position is performed.

In this chapter, all parameters are described which are necessary for a closed loop position control. The control loop is fed with the position demand value as one of the outputs of the trajectory generator and with the output of the position detection unit (position actual value) of encoder as input parameters. The behavior of the control may be influenced by control parameters which are externally applicable. To keep stable the loop, a relative limitation of the output using the previous control effort is possible. In order not to exceed physical limits of a drive, an absolute limit function is implemented for the control effort. The control effort may be a velocity demand value, a position demand value or any other output value, depending on the modes of operation implemented by a manufacturer. Especially in cascaded control structures, where a position control is followed by a torque control, e.g. the control effort of the position control loop is used as an input for a further calculation.

Functional Description

Two different ways to apply target positions to a drive, are supported by this device profile.

Set of set-points:

After reaching the target position the drive unit immediately processes the next target position which results in a move where the velocity of the drive normally is not reduced to zero after achieving a set-point.

Single set-point:

After reaching the target position the drive unit signals this status to a host computer and then receives a new set-point. After reaching a target position the velocity normally is reduced to zero before starting a move to the next set-point.

The two modes are controlled by the timing of the bits ‘new set-point’ and ‘change set immediately’ in the controlword and ‘set-point acknowledge’ in the statusword. These bits allow to set up a request-response mechanism in order to prepare a set of setpoints while another set still is processed in the drive unit. This minimizes reaction times within a control program on a host computer.

Procedure

- Set [Mode of operation (6060h)] to operating mode Profile position (1).
- Set [Target position (607Ah)] to the target position (unit = pulse).
- Set [Profile velocity (6081h)] to profile velocity (unit = pulses per second).
- Set [Controlword (6040h)] to start the movement.
- Query [Position actual value (6064h)] to get the actual position of the motor.
- Query [Statusword (6041h)] to get the current status of following error, set- point acknowledge and target reached.

Optional

Additional information on the operating mode Profile Position:

- Query [Position demand value (6062h)] to get the internal reference value (unit = pulse).
- Query [Position actual value (6063h)] to get the actual position value (unit = increments).

Following error:

- Set [Following error window (6065h)] to the permissible following error (unit = pulse).
- Query [Following error actual value (60F4h)] to get the current following error (unit = pulse).

Standstill window:

- Set [Position window (6067h)] to the value for the standstill window. If the difference between the target position and the current motor position remains in the standstill window for the time Position window time (6065h), the target position is considered to have been reached (unit = pulse).
- Set [Position window time (6068h)] to the value for the standstill window. If the difference between the target position and the current motor position remains in the standstill window

for the time Position window time (6065h), the target position is considered to have been reached (unit = pulse).

Associated Objects

Index	Sub-index	Object	PDO	Data Type	Takes Effect
6040h	0	Controlword	R_PDO	Unsigned16	Immediately
6041h	0	Statusword	T_PDO	Unsigned16	-
6060h	0	Modes of operation	No	Integer8	Immediately
6061h	0	Modes of operation display	No	Integer8	-
6062h	0	Position demand value	No	Integer32	-
6063h	0	Position actual value	No	Integer32	-
6064h	0	Position actual value	T_PDO	Integer32	-
6065h	0	Following error window	No	Unsigned32	-
6067h	0	Position window	No	Unsigned32	-
6068h	0	Position window time	No	Unsigned16	Immediately
6081h	0	Profile velocity	R_PDO	Unsigned32	Next movement
6091h 6092h	1	Numerator (Position factor)	R_PDO	Unsigned32	Immediately
6091h 6092h	2	Speed constant (Position factor)	R_PDO	Unsigned32	Immediately
60F2h	0	Position option code	No	Unsigned16	Next movement
60F4h	0	Following error actual value	No	Integer32	-
60FCh	0	Position demand value	No	Integer32	-

Example: Profile Position

Starting the Operating Mode

The operating mode must be set in the parameter Mode of operation (6060h). Writing the parameter value activates the operating mode. The movement is started via the Controlword.

Controlword

Bits 4–6 and bit 8 in the parameter Controlword (6040h) start a movement.

Bit 5: Change Set Point Immediately	Bit 4: New Target Value	Meaning
0	0 » 1	Starts a movement to a target position. New target values transmitted during a movement become effective only when the previous target is reached. The movement is stopped at the current target position.*
1	0 » 1	Starts a movement to a target position. New target values transmitted during a movement become immediately effective. Previous target will be ignored. The movement is stopped at the current target position.*

Note: Target values include target position, target velocity, acceleration and deceleration.

Parameter Value	Meaning
Bit 6 = Absolute / relative	0: Absolute movement 1: Relative movement
Bit 8 = Halt	Stop movement with Halt

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Target position reached
- Stop caused by Halt or Quick Stop
- Stop caused by an error

Statusword

Information on the current movement is available via bits 10 and 12–15 in the parameter Statusword (6041h).

Parameter Value	Meaning
Bit 10 = Target reached	0 = Target position not reached 1 = Target position reached
Bit 12 = Target value acknowledge	0 = New position possible 1 = New target position accepted
Bit 13 = Following error bit	0 = No following error 1 = Following error
Bit 14 = Manufacturer-specific	
Bit 15 = Manufacturer-specific	

Homing Mode

Description

In the operating mode Homing, a movement is performed to a defined position. This position is defined as the reference point.

This chapter describes the method by which a drive seeks the home position (also called, the datum, reference point or zero point). There are various methods of achieving this using limit switches at the ends of travel or a home switch (zero point switch) in midtravel, most of the methods also use the index (zero) pulse train from an incremental encoder.

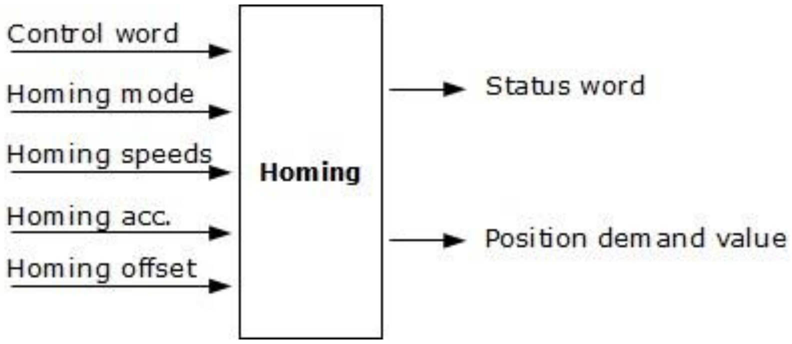


Figura 4: Homing function

Procedure

- Set [Mode of operation (6060h)] to operating mode Homing (6).
- Set [Home offset (607Ch)].
- Set [Home method (6098h)], the value range is 1 to 35 and specifies the different homing methods.
- Set [Home speeds (6099h sub-index 1)] to the value for velocity for the search for the limit switches (unit = min-1).
- Set [Home speeds (6099h sub-index 2)] to the value for velocity for the search for the index pulse (unit = min-1).
- Set [Home acceleration (6099h sub-index 2)] to the value for the acceleration ramp (unit = milliseconds form 0 to 3000 min-1).
- Set [Controlword (6040h)] to start the operating mode.
- Start Homing.
- Query [Statusword (6041h)] to get the device status.

Associated Objects

Index	Sub-index	Object	PDO	Data Type	Takes Effect
6040h	0	Controlword	R_PDO	Unsigned16	Immediately
6041h	0	Statusword	T_PDO	Unsigned16	-
6060h	0	Modes of operation	No	Integer8	Immediately
6061h	0	Modes of operation display	No	Integer8	-

607Ch	0	Home offset	No	Integer32	Next movement
6098h	0	Homing method	No	Integer8	Next movement
6099h	1	Speed during search for switch	No	Unsigned32	Next movement
6099h	2	Speed during search for zero	No	Unsigned32	Next movement
609Ah	0	Homing acceleratio n	No	Unsigned32	Next movement

Functional description

By choosing a method of homing by writing a value to homing method will clearly establish:

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

The home position and the zero position are offset by the home offset, see the definition of home offset for how this offset is used. Various homing positions are illustrated in the following diagrams. An encircled number indicates the code for selection of this homing position. The direction of movement is also indicated.

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

In the diagrams of homing sequences shown below, the encoder count increases as the axis position moves to the right, in other words the left is the minimum position and the right is the maximum position. 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. There are several, application-specific methods. The homing method is used for selection. The exact sequence of the homing operation is clearly described by the method. In some circumstances, a device has several methods to choose from, using the homing method.

Homing methods

The following sub-sections describe the details of how each of the homing modes shall function.

Method 1: Homing on the negative limit switch and index pulse

Using this method the initial direction of movement is leftward if the negative limit switch is inactive (here shown as low). The home position is at the first index pulse to the right of the position where the negative limit switch becomes inactive.

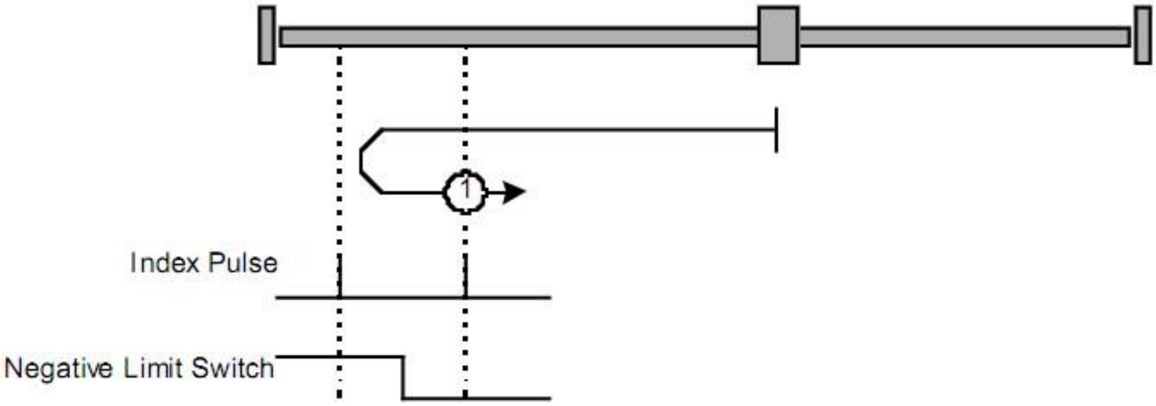


Figura 5- Homing on negative Limit switch and index pulse

Method 2: Homing on the positive limit switch and index pulse

Using this method the initial direction of movement is rightward if the positive limit switch is inactive (here shown as low). The position of home is at the first index pulse to the left of the position where the positive limit switch becomes inactive.

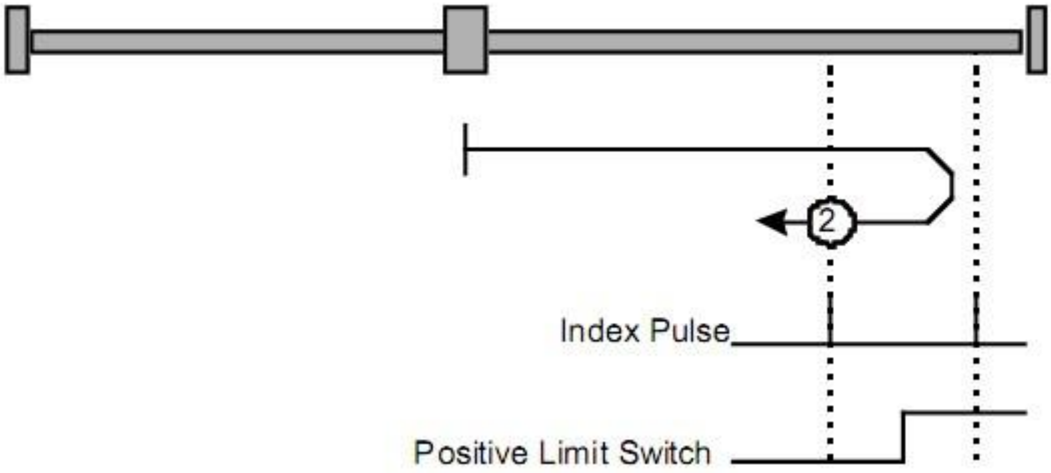


Figura 6- Homing on positive Limit switch and index pulse

Methods 3 and 4: Homing on the positive home switch and index pulse

Using methods 3 or 4 the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is sited so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

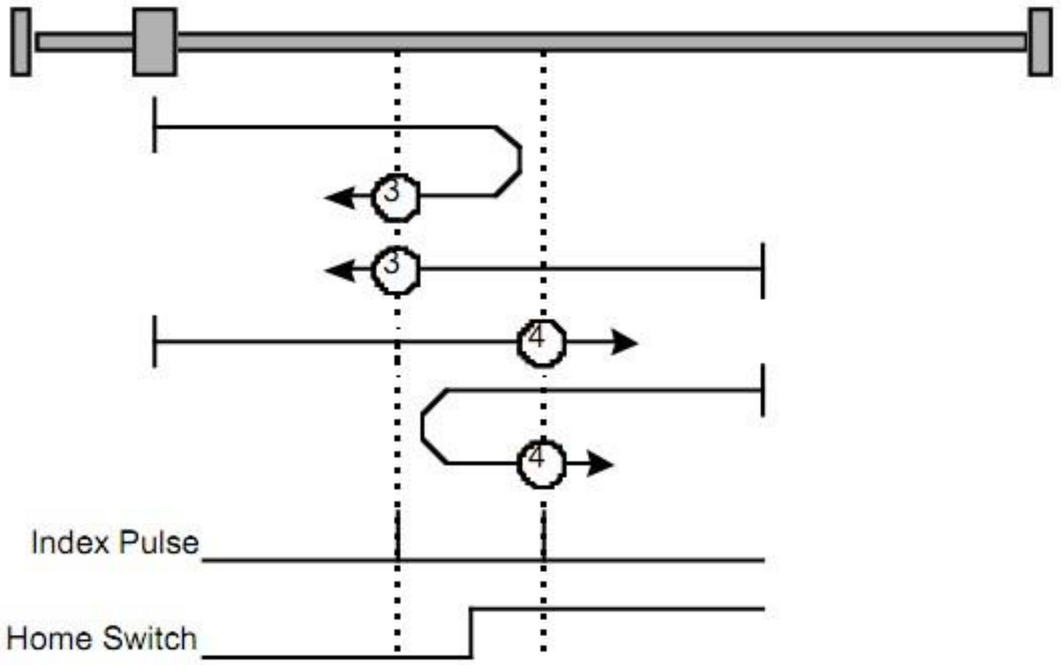


Figura 7- Homing on positive home switch and index pulse

Methods 5 and 6: Homing on the negative home switch and index pulse

Using methods 5 or 6 the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either to the left or the right of the point where the home switch changes state. If the initial position is sited so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.

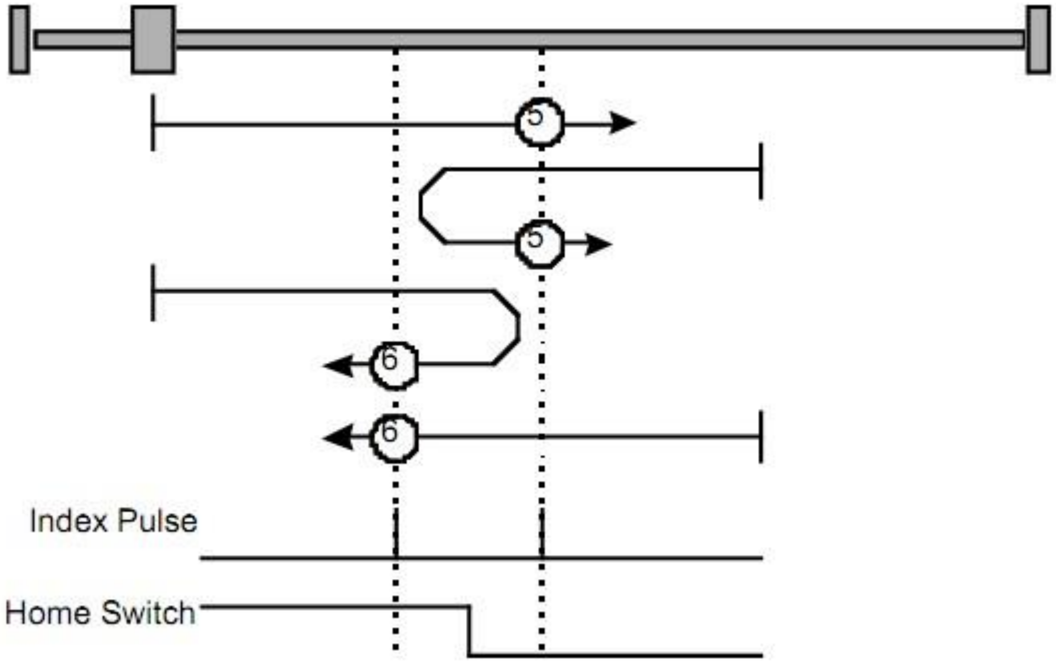


Figura 8- Homing on negative home switch and index pulse

Methods 7 to 14: Homing on the home switch and index pulse

These methods use a home switch which is active over only portion of the travel, in effect the switch has a ‘momentary’ action as the axle's position sweeps past the switch. Using methods 7 to 10 the initial direction of movement is to the right, and using methods 11 to 14 the initial direction of movement is to the left except if the home switch is active at the start of the motion.

In this case the initial direction of motion is Dependent on the edge being sought. The home position is at the index pulse on either side of the rising or falling edges of the home switch, as shown in the following two diagrams. If the initial direction of movement leads away from the home switch, the drive must reverse on encountering the relevant limit switch.

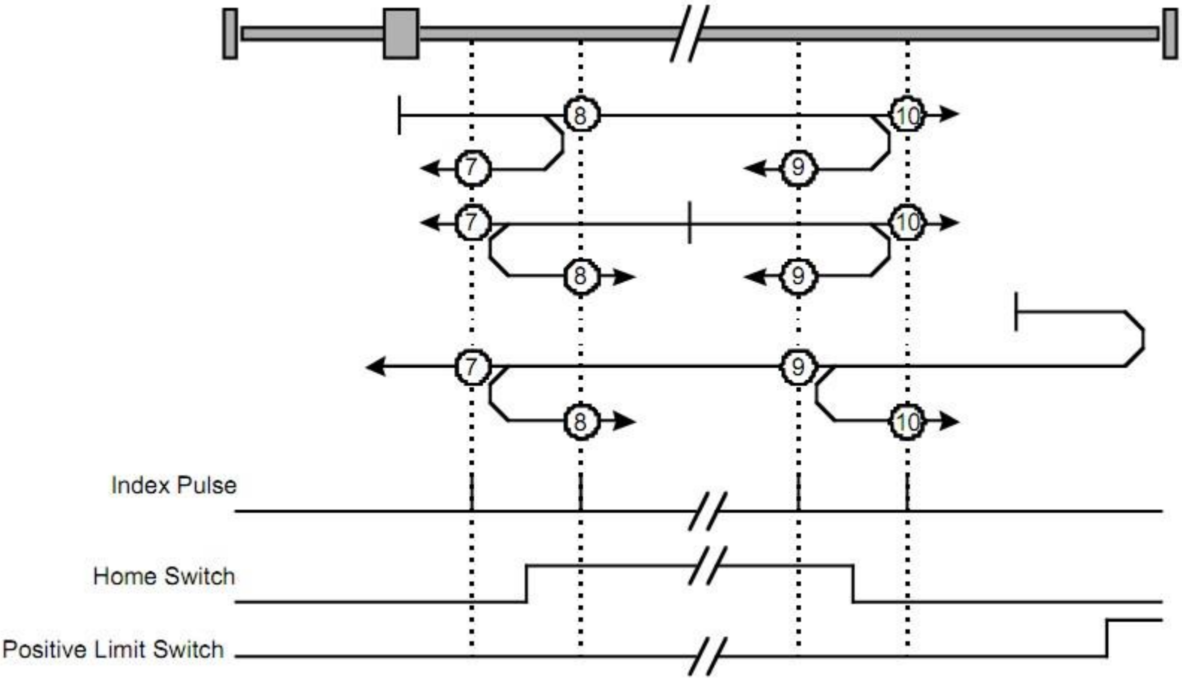


Figura 9- : Homing on the home switch and index pulse - positive initial move

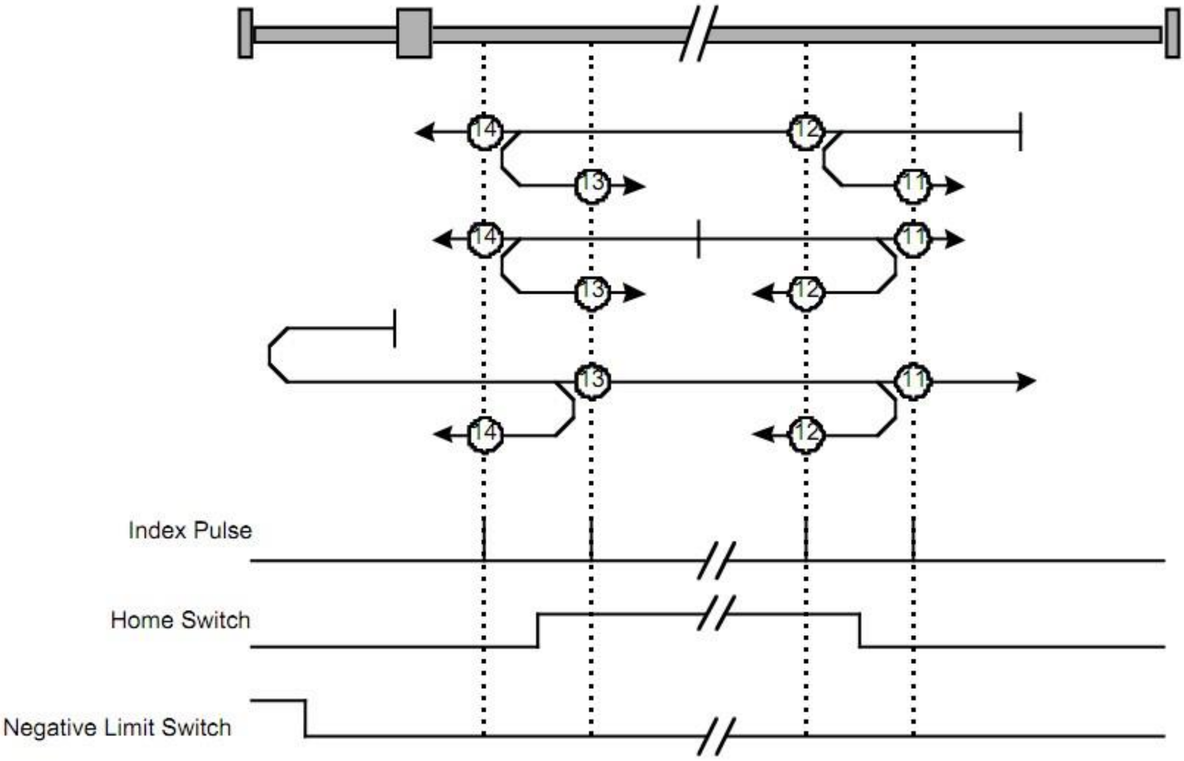


Figura 10 Homing on the home switch and index pulse - negative initial move

Methods 17 to 30: Homing without an index pulse

These methods are similar to methods 1 to 14 except that the home position is not dependent on the index pulse but only dependent on the relevant home or limit switch transitions. For example methods 19 and 20 are similar to methods 3 and 4 as shown in the following diagram.

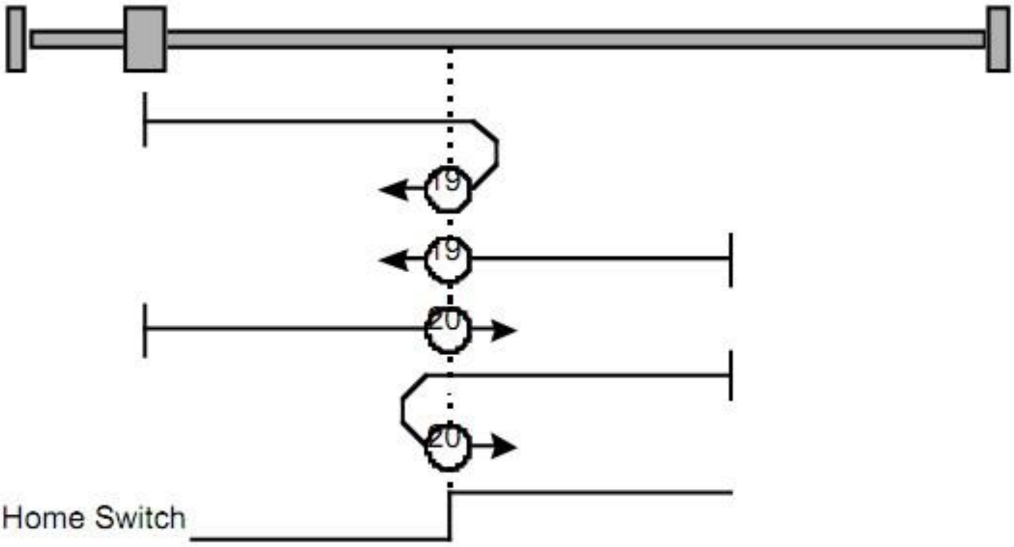


Figura 11- Homing without and index pulse

Methods 33 to 34: Homing on the index pulse

Using methods 33 or 34 the direction of homing is negative or positive respectively. The home position is at the index pulse found in the selected direction.

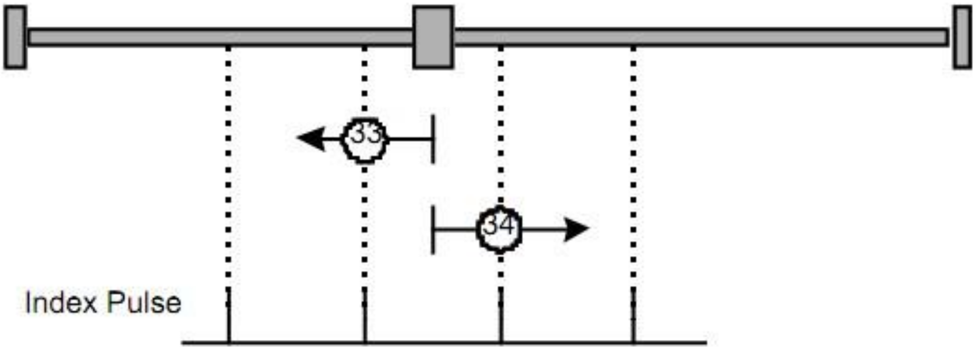


Figura 12 Homing on index pulse

Method 35: Homing on the current position

In method 35 the current position is taken to be the home position.

Method -17: Negative Stop Homing

In method -17 the axis will move in negative direction and an homing is executed if the current is greater then homing current threshold for a time greater than homing current threshold time

Method -18: Positive Stop Homing

In method -18 the axis will move in positive direction and an homing is executed if the current is greater then homing current threshold for a time greater than homing current threshold time

Example: Homing

Starting the Operating Mode:

The operating mode must be set in the parameter Mode of operation (6060h). Writing the parameter value activates the operating mode.

The movement is started via the Controlword.

Controlword

Bits 4 in the parameter Controlword (6040h) starts a movement, bit 8 terminates the movement.

Parameter Value	**Meaning**
Bit 4 = Homing operation start	Start Homing
Bit 5 = Reserved	Not relevant for this operating mode
Bit 6 = Reserved	Not relevant for this operating mode
Bit 8 = Halt	Stop movement with Halt

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Homing successful
- Stop caused by Halt or Quick Stop

- Stop caused by an error

Statusword

Information on the current movement is available via bits 10 and 12–15 in the parameter Statusword (6041h).

Parameter Value	Meaning
Bit 10 = Target reached	0 = Homing not completed 1 = Homing completed
Bit 12 = Homing attained	1 = Homing successfully completed
Bit 13 = Homing error	1 = Homing error
Bit 14 = Manufacturer-specific	
Bit 15 = Manufacturer-specific	

Profile Velocity Mode

Description

In the operating mode Profile Velocity, a movement is made with a desired target velocity.

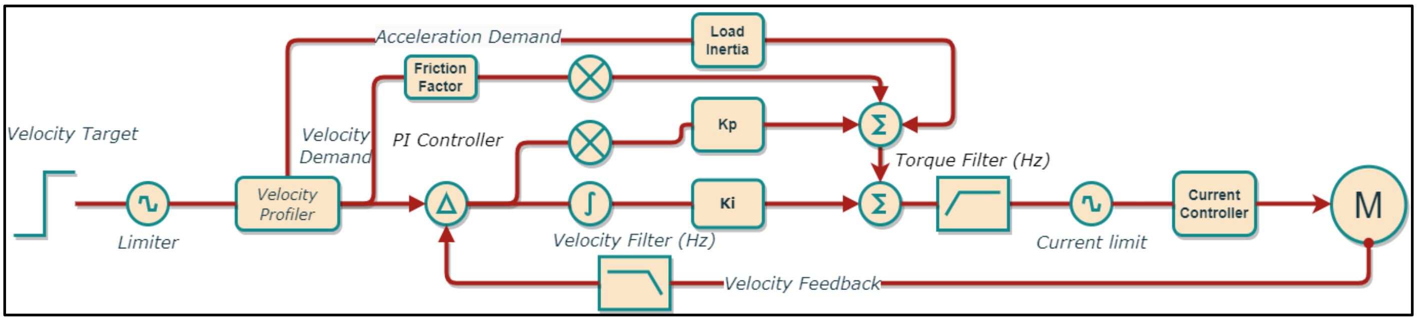


Figure 6 velocity loop controller

Procedure

- Set [Mode of operation (6060h)] to operating mode Profile Velocity (3).
- Set [Controlword (6040h)] to start the operating mode.
- Set [Target velocity (60FFh)] to the target velocity. If the power stage is enabled, the new target velocity will become active immediately and the movement will start. The value is reset to zero if the operating mode is changed, the power stage is disabled or a Quick Stop is triggered.
- Query [Statusword (6041h)] to get the device status.

Optional

- Query [Velocity demand value (606Bh)] to get the reference velocity.
- Query [Velocity actual value (60C3h)] to get the actual velocity.
- Set [Velocity window (606Dh)] to the value of the velocity window.
- Set [Velocity window time (606Eh)] to the duration in the velocity window required to consider the velocity to have been reached unit = milliseconds).
- Query [Velocity threshold (60F4h)] to set the standstill window.

Associated Objects

Index	Sub- index	Object	PDO	Data Type	Takes Effect

6040h	0	Controlword	R_PDO	Unsigned16	Immediately
6041h	0	Statusword	T_PDO	Unsigned16	-
6060h	0	Modes of Operation	No	Integer8	Immediately
6061h	0	Modes of Operation Display	No	Integer8	-
606Bh	0	Velocity Demand Value	No	Integer32	-
606Ch	0	Velocity Actual Value	No	Integer32	-
606Dh	0	Velocity Window	No	Unsigned16	Immediately
606Eh	0	Velocity Window Time	No	Unsigned16	Immediately
606Fh	0	Velocity Threshold	No	Unsigned16	Immediately
60FFh	0	Target Velocity	No	Integer32	Immediately

Example: Profile Velocity

Starting the Operating Mode

The operating mode must be set in the parameter Mode of operation (6060h). Writing the parameter value activates the operating mode.

The parameter Target velocity (60FFh) starts the movement.

Parameter Name	**Description**	**Data Type R/W**
Target Velocity	Target velocity for operating mode Profile Velocity Changed settings become active immediately.	Integer32 Read/Write

Controlword

Bit 8 in parameter Controlword (6040h) is used to stop a movement with Halt.

Parameter Value	**Meaning**
Bit 4 = Reserved	Not relevant for this operating mode

Bit 5 = Reserved	Not relevant for this operating mode
Bit 6 = Reserved	Not relevant for this operating mode
Bit 8 = Halt	Stop movement with Halt
Bit 9 = Change on set point	Not relevant for this operating mode

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by Halt or Quick Stop
- Stop caused by an error

Statusword

Information on the current movement is available via bits 10 and 12 in the parameter Statusword (6041h).

Parameter Value	Meaning
Bit 10 = Target reached	0 = Target velocity not reached 1 = Target velocity reached
Bit 12 = Velocity	0 = Velocity > 0 1 = Velocity = 0
Bit 14 = Manufacturer-specific	
Bit 15 = Manufacturer-specific	

Profile Torque Mode

Description

In the operating mode Profile Torque, a movement is made with a desired target torque.

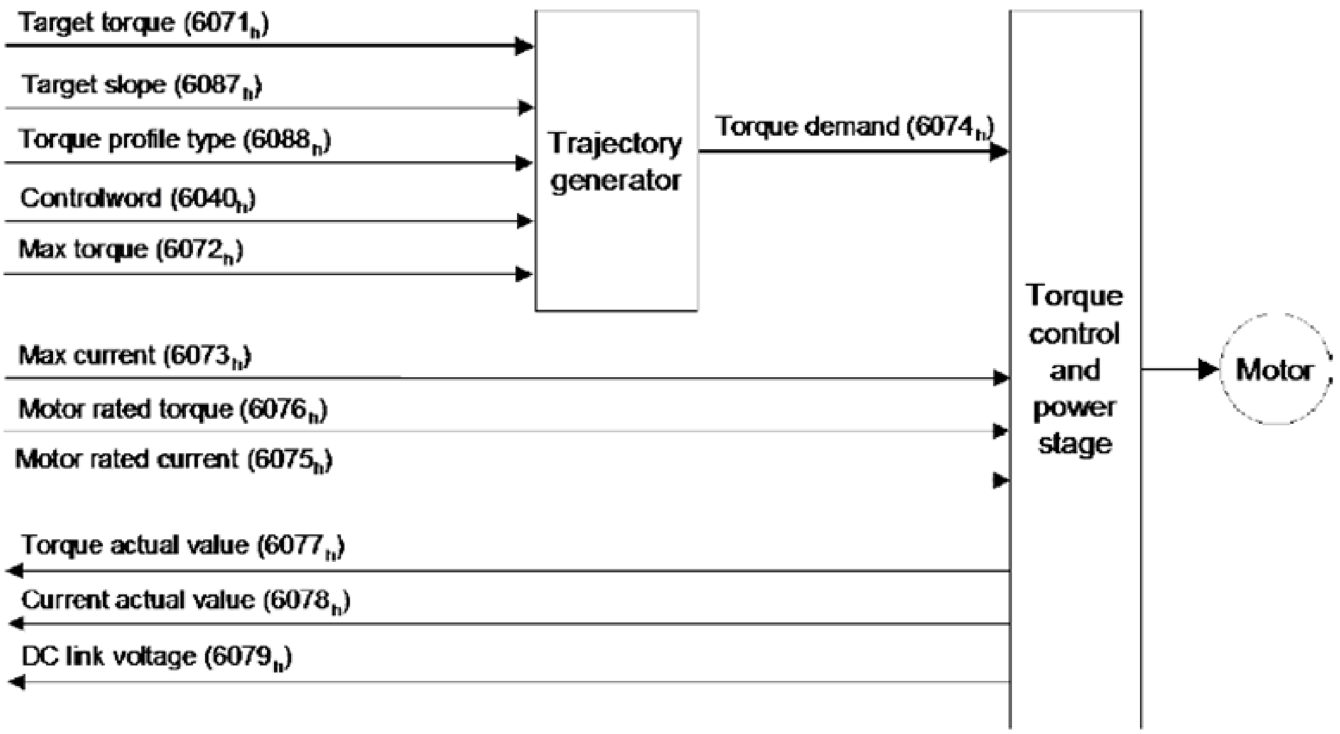

Target torque (6071_h)
Target slope (6087_h)
Torque profile type (6088_h)
Controlword (6040_h)
Max torque (6072_h)
Trajectory generator
Torque demand (6074_h)
Max current (6073_h)
Motor rated torque (6076_h)
Motor rated current (6075_h)
Torque control and power stage
Motor
Torque actual value (6077_h)
Current actual value (6078_h)
DC link voltage (6079_h)

Figura 13-Structure of the Profile Torque Mode

Procedure

- Set [Mode of operation (6060)] to operating mode Profile Torque (4).
- Set [Controlword (6040h)] to start the operating mode.
- Set [Motor rated current (6075)] to a value according to motor specifications (unit = mA).
- Set [Target torque (6071h)] to the value for the target torque.
- Set [Controlword (6040h)] to start the movement.

Optional

- Query [Torque rated current (6075h)] to get the nominal current depending on the motor and the drive (unit = multiples of mA).
- Query [Current actual value (6078h)] to get the actual current.

Associated Objects

.

Index	**Sub-index**	**Object**	**PDO**	**Data Type**	**Takes Effect**
6040h	0	Controlword	R_PDO	Unsigned16	Immediately
6041h	0	Statusword	T_PDO	Unsigned16	-
6060h	0	Modes of Operation	No	Integer8	Immediately
6061h	0	Modes of Operation Display	No	Integer8	-
6071h	0	Target Torque	R_PDO	Integer16	Immediately
6074h	0	Torque demand value	No	Integer16	-
6075h	0	Motor rated current	No	Unsigned32	-
6087h	0	Torque slope	R_PDO	Unsigned32	Immediately

Interpolated Position Mode

Description

In the operating mode Interpolated Position, a movement to a target position is performed according to the value of the synchronous cyclic time.

This mode uses a buffer of position commands. The buffer size is always 1, thus it is not possible to give a list of target position commands in advance.

Note: only usable with CANopen.

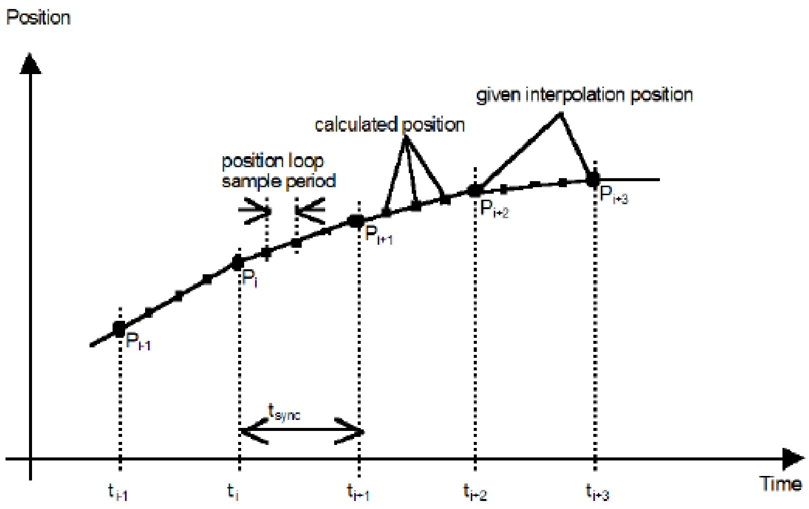


Figura 14-Linear Interpolation For 1 Axis

Procedure

- Set [Mode of operation (6060h)] to operating mode Interpolated Position (7).
- Set [interpolated data record (60C1h SubIdx 1)] to the target position (unit = pulse).
- Set [Controlword (6040h)] to start the movement.
- Query [Position actual value (6064h)] to get the actual position of the motor.
- Query [Statusword (6041h)] to get the current status of following error, set- point acknowledge and target reached.

Optional

Additional information on the operating mode Profile Position:

- Query [Position demand value (6062h)] to get the internal reference value (unit = pulse).

- Query [Position actual value (6063h)] to get the actual position value (unit = increments).
- Following error:
- Set [Following error window (6065h)] to the permissible following error (unit = pulse).
- Query [Following error actual value (60F4h)] to get the current following error (unit = pulse).

Index	Sub-index	Object	PDO	Data Type	Takes Effect
6040h	0	Controlword	R_PDO	Unsigned16	Immediately
6041h	0	Statusword	T_PDO	Unsigned16	-
6060h	0	Modes of operation	R_PDO	Integer8	Immediately
6061h	0	Modes of operation display	T_PDO	Integer8	-
6062h	0	Position demand value	No	Integer32	-
6063h	0	Position actual value	No	Integer32	-
6064h	0	Position actual value	T_PDO	Integer32	-
6065h	0	Following error window	No	Unsigned32	-
6067h	0	Position window	No	Unsigned32	-
6068h	0	Position window time	No	Unsigned16	Immediately
60F2h	0	Position option code	No	Unsigned16	Next movement
60F4h	0	Following error actual value	T_PDO	Integer32	-
60FCh	0	Position demand internal value	No	Integer32	-
60C1h	1	Interpolated Data Record	R_PDO	Integer32	Next movement
60C2h	1	Interpolation time period	no	Unsigned16	-
60C2h	2	Interpolation time index	no	Integer8	-

Example: Interpolated Position Profile

Starting the Operating Mode:

The operating mode must be set in the parameter Mode of Operation (6060h). Writing the parameter value activates the operating mode. The movement is started via the Controlword.

Controlword

Bit 4 in the parameter Controlword (6040h) start a movement

Bit 4: **New Target Value**	**Meaning**
0 » 1	Starts a movement to a target position. Target values transmitted during a movement every tick times according to the synchronous time value. The movement is stopped at the desired target position.

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Target position reached
- Stop caused by Quick Stop
- Stop caused by an error

Statusword

Information on the current movement is available via bits 10 and 12–15 in the parameter Statusword (6041h).

Parameter Value	**Meaning**
Bit 10 = Target reached	0 = Target position not reached 1 = Target position reached
Bit 12 = Target value acknowledge	0 = New position possible 1 = New target position accepted
Bit 13 = Following error bit	0 = No following error 1 = Following error
Bit 14 = Manufacturer-specific	
Bit 15 = Manufacturer-specific	

Cyclic Synchronous Position Mode

Description

In the operating mode Cyclic Synchronous Position, a movement to a target position is performed according to the value of the synchronous cyclic time.

Note: minimum supported cycle time (SYNC0 event in DC mode) is 1ms.

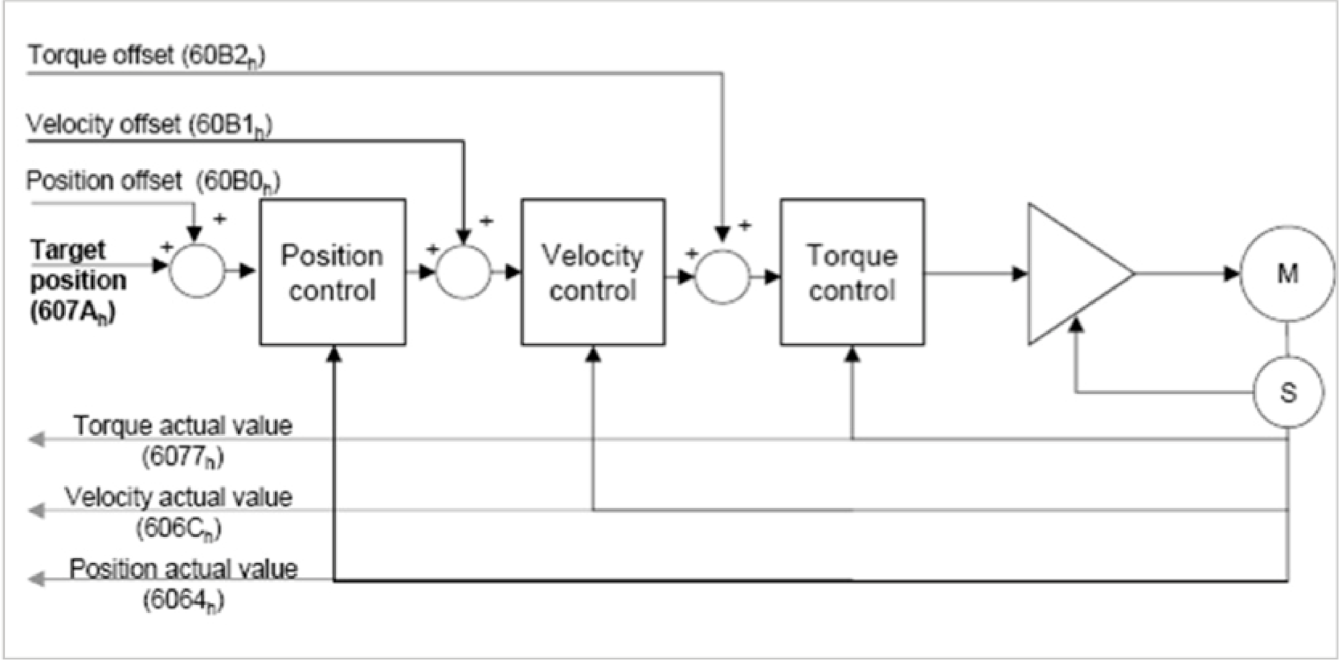


Figura 15 CSPM Overview

Procedure

- Set [Mode of operation (6060h)] to operating mode Cyclic Synchronous Position Profile (8).
- Set [Target position (607Ah)] to the target position (unit = pulse).
- Set [Controlword (6040h)] to start the movement.
- Query [Position actual value (6064h)] to get the actual position of the motor.
- Query [Statusword (6041h)] to get the current status of following error, set- point acknowledge and target reached.

Optional

Additional information on the operating mode Profile Position:

- Query [Position demand value (6062h)] to get the internal reference value (unit = pulse).
- Query [Position actual value (6063h)] to get the actual position value (unit = increments).

Following error:

- Set [Following error window (6065h)] to the permissible following error (unit = pulse).
- Query [Following error actual value (60F4h)] to get the current following error (unit = pulse).

Associated Objects

Index	Sub-index	Object	PDO	Data Type	Takes Effect
6040h	0	Controlword	R_PDO	Unsigned16	Immediately
6041h	0	Statusword	T_PDO	Unsigned16	-
6060h	0	Modes of operation	R_PDO	Integer8	Immediately
6061h	0	Modes of operation display	T_PDO	Integer8	-
6062h	0	Position demand value	No	Integer32	-
6063h	0	Position actual value	No	Integer32	-
6064h	0	Position actual value	T_PDO	Integer32	-
6065h	0	Following error window	No	Unsigned32	-
6067h	0	Position window	No	Unsigned32	-
6068h	0	Position window time	No	Unsigned16	Immediately
607Ah	0	Target Position	R_PDO	Integer32	Next movement
60F2h	0	Position option code	No	Unsigned16	Next movement
60F4h	0	Following error actual value	T_PDO	Integer32	-
60FCh	0	Position demand internal value	No	Integer32	-
60C2h	1	Interpolation time period	no	Unsigned16	-
60C2h	2	Interpolation time index	no	Integer8	-

Example: Cyclic Synchronous Position Profile

Starting the Operating Mode

The operating mode must be set in the parameter Mode of Operation (6060h). Writing the parameter value activates the operating mode. The movement is started via the Controlword.

Controlword

Bit 4 in the parameter Controlword (6040h) start a movement.

Bit 4: New Target Value	Meaning
0 » 1	Starts a movement to a target position. Target values transmitted during a movement every tick times according to the synchronous time value. The movement is stopped at the desired target position.

Terminating the Operating Mode

The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Target position reached
- Stop caused by Quick Stop
- Stop caused by an error

Statusword

Information on the current movement is available via bits 10 and 12–15 in the parameter Statusword (6041h).

Parameter Value	**Meaning**
Bit 10 = Target reached	0 = Target position not reached 1 = Target position reached
Bit 12 = Target value acknowledge	0 = New position possible 1 = New target position accepted
Bit 13 = Following error bit	0 = No following error 1 = Following error

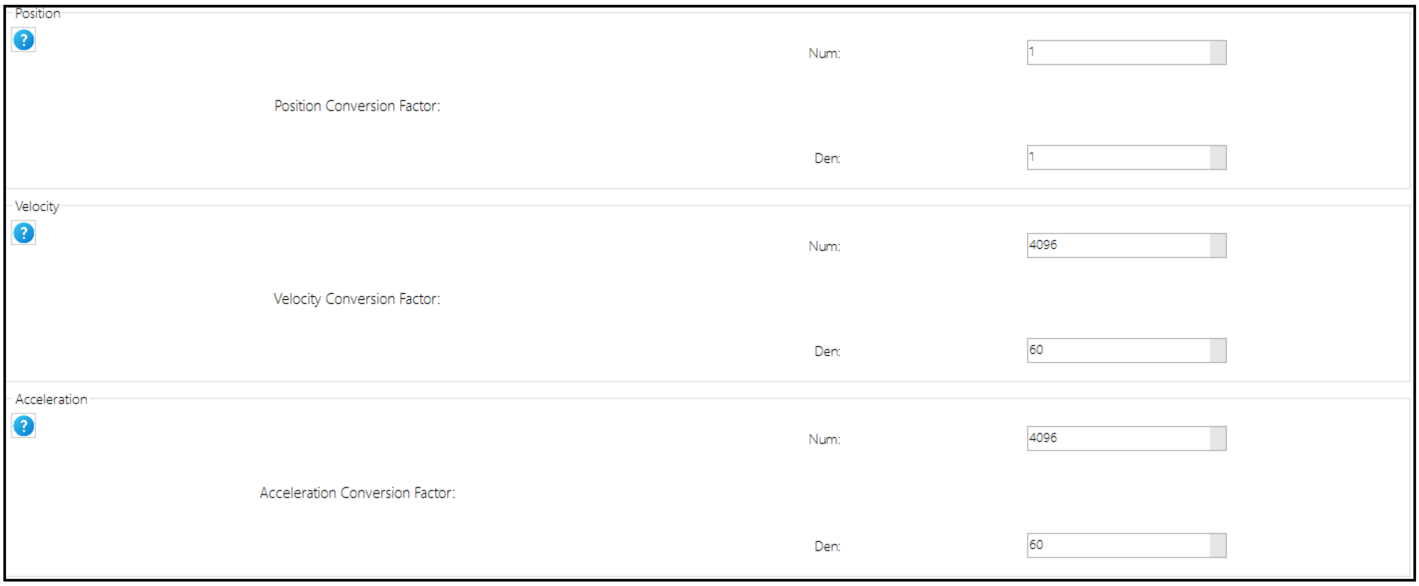
Bit 14 = Manufacturer-specific	
Bit 15 = Manufacturer-specific	

4. Units

This chapter define the available units and how to change for Gear Units, Position Units, Encoder Resolution, Acceleration Units and Velocity units.

They are changeable directly by the UI with ModBus or via CANopen/EtherCAT SDOs.

Opening the conversion factor window let one change the factors.



Position

Position Conversion Factor: Num: 1 Den: 1

Velocity

Velocity Conversion Factor: Num: 4096 Den: 60

Acceleration

Acceleration Conversion Factor: Num: 4096 Den: 60

Figure 7-Conversion Factors Window

Encoder resolution

Changing the (608Fh) parameter or its equivalent in Axis parameters by UI you can change encoder Line Per Revolution. Default Value is 1024 line per revolution.

Note: only expert users can do this, max encoder increments per revolution is 14 bits.

Conversion factor for position units

The factor n is calculated from the factor for numerator (6093h:01h) divided by the factor for denominator (6093h:02h).

So the final units for position is:

$$Position\ Units = Encoder\ Increment * \frac{6093\mathrm{h}{:}\,01\mathrm{h}}{6093\mathrm{h}{:}\,02\mathrm{h}}$$

CAN Object	Parameter	Default Value
6093h, sub-index 1	Position Conversion Factor Num	1
6093h, sub-index 2	Position Conversion Factor Den	1

With default values encoder increments are equal to position increments.

Conversion factor for the speed units

The factor n is calculated from the factor for numerator (6096h:01h) divided by the factor for denominator (6096h:01h).

This factor converts velocity in internal formats units $(\mathrm{increments/seconds})$

$$Velocity\ Units = \frac{Encoder\ Increment}{s} * \frac{6096\mathrm{h}{:}\,01\mathrm{h}}{6096\mathrm{h}{:}\,02\mathrm{h}}$$

CAN Object	Parameter	Default Value
6096h, sub-index 1	Velocity Conversion Factor Num	4096
6096h, sub-index 2	Velocity Conversion Factor Den	60

Default values for Velocity Units is RPM (Rounds per minute).

Example:

- Velocity data from the driver are in increments/s.

- Desired outputs/inputs are in RPM.
- Which is the proper conversion factor?
- 1 rounds = 4096 increments

Finding the right factors:

$$\frac{round}{minuts} = \frac{4096\ inc}{60\ seconds} = \frac{4096\ (6096\text{h}:01\text{h})}{60\ (6096\text{h}:02\text{h})}$$

Conversion factor for the acceleration units

The factor n is calculated from the factor for numerator (6097h:01h) divided by the factor for denominator (6097h:01h).

This factor converts acceleration in internal drive units $(\text{increments}/\text{seconds}^2)$

$$Acceleratuin\ Units = \frac{Encoder\ Increment}{s^2} * \frac{6097\text{h}:01\text{h}}{6097\text{h}:02\text{h}}$$

CAN Object	Parameter	Default Value
6097h, sub-index 1	Acceleration Conversion Factor Num	4096
6097h, sub-index 2	Acceleration Conversion Factor Den	60

5. Communication Profile

This chapter define all the Device object available in the communication profile area of the Object Dictionary.

Object 1000h – Device Type

This object contains information about the device type and functionality. It is comprised of a 16-bit field that describes the device profile used, and a second 16-bit field that gives additional information about optional functionality of the device.

MSB			LSB
BYTE 4	BYTE 3	BYTE 2	BYTE 1
Additional Information		Device profile Number	

Object Description:

Index	**1000$_h$**
Name	Device type
Object code	VAR
Data Type	UNSIGNED32
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x191

Object 1001h – Error Register

This object is an error register for the device.

Object Description:

Index	**1001$_h$**
Name	Error Register
Object code	VAR
Data Type	UNSIGNED8
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x0

Data description:

bit	**Meaning**
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error
5	Device Profile Specific
6	Reserved
7	Manufacturer Specific

If a bit is set to 1, the specified error has occurred.

Object 1003h – Pre-defined error field

The object at index 1003h holds the errors that have occurred on the device and have been signaled via the Emergency Object. In doing so it provides an error history.

1. The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1.
2. Every new error is stored at sub-index 1, the older ones move down the list.
3. Writing a „0“ to sub-index 0 deletes the entire error history (empties the array).
4. The error numbers are of type UNSIGNED32 and are composed of a 16 bit error code and a 16 bit additional error information field which is manufacturer specific. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB). The Error list contain 8 elements.

MSB			LSB
BYTE 4	BYTE 3	BYTE 2	BYTE 1
Additional Information		Error Code	

Object Description:

Index	**1003$_h$**
Name	Pre-defined error field
Object code	ARRAY
Data Type	UNSIGNED32
Category	Optional
Note	Valid only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of actual error
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x0

Sub-Index	***01 - 08***
Description	Standard Error Field
Entry Category	Mandatory
Access	Read
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x0

Object 1005h - COB-ID SYNC message

This object defines the COB-ID of the synchronization object (SYNC).

MSB				LSB
31	30	29	28-11	10-0
x	0	0	0	11-bit identifier

Object Description:

Index	**1005_h**
Name	COB-ID SYNC message
Object code	VAR
Data Type	UNSIGNED32
Category	**Mandatory**
Note	Valid only for CANopen

Entry Description:

Access	**Read/Write**
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x80

Object 1006h - Communication Cycle Period

This object defines the communication cycle period in µs. This period defines the SYNC interval. It is 0 if not used. If the communication cycle period on sync producer is changed to a new value unequal 0 the transmission of sync object resumes within 1 sync cycle of the new value.

Object Description:

Index	**1006_h**
Name	Communication Cycle Period
Object code	VAR

Data Type	UNSIGNED32
Category	Conditional; Mandatory for sync producers
Note	Valid only for CANopen

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x0
Note	Valid only for CANopen

Object 1007h - Synchronous Window Length

Contains the length of the time window for synchronous PDOs in ms. It is 0 if not used.

Object Description:

Index	**1007$_h$**
Name	Communication Cycle Period
Object code	VAR
Data Type	UNSIGNED32
Category	Optional
Note	Valid only for CANopen

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x0

Object 1008h: Manufacturer Device Name

Contains the manufacturer device name.

Object Description:

Index	**1008$_h$**
Name	Manufacturer Device Name
Object code	VAR
Data Type	Visible String
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	-
Default Value	-

Object 1009h: Manufacturer Hardware Version

Contains the manufacturer hardware version description.

Object Description:

Index	**1009$_h$**
Name	Manufacturer Hardware Version
Object code	VAR
Data Type	Visible String
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	-

Default Value	-

Object 100Ah: Manufacturer Software Version

Contains the manufacturer software version description.

Object Description:

Index	**1008$_h$**
Name	Manufacturer Software Version
Object code	VAR
Data Type	Visible String
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	-
Default Value	-

Object 1010h - Save parameters

This object is used to save parameters in non-volatile memory. Through read access, the drive provides information about its save capabilities, using:

(a) ƒSub-index 0: Largest supported sub-index

(b) Sub-index 1: Save all parameters

In order to avoid accidental storage, storage is only executed when a specific signature "save" is written to the appropriate sub-index.

MSB			LSB
BYTE 4	BYTE 3	BYTE 2	BYTE 1
"e"	"v"	"a"	"s"
65h	76h	61h	73h

Object Description:

Index	**1010$_h$**
Name	Save parameters
Object code	ARRAY
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Sub-Index	***0***
Description	largest sub-index supported
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	1

Sub-Index	***1***
Description	Save all parameters
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	-

Object 1011h - Restore default parameters

This object is used to restore parameters from non-volatile memory. Through read access, the drive provides information about its restore capabilities, using:

(a) ƒSub-index 0: Largest supported sub-index

(b) ƒSub-index 1: Restore all parameters

In order to avoid accidental storage, restore is only executed when a specific signature "load" is written to the appropriate sub-index.

MSB			LSB
BYTE 4	BYTE 3	BYTE 2	BYTE 1
"d"	"a"	"o"	"l"
64h	61h	6Fh	6Ch

Object 1014h - COB-ID Emergency Object

Index 1014h defines the COB-ID of the Emergency Object (EMCY).

UNSIGNED32

MSB			LSB	
31	30	29	28-11	10-0
Reserved (=0)		0	0	11-bit identifier

Object Description:

Index	**1014**$_h$
Name	COB-ID Emergency Object
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x08+Node-ID

Object 1015h - Inhibit Time EMCY

The inhibit time for the EMCY message can be adjusted via this entry. The time has to be a multiple of 100µs.

Object Description:

Index	**1014**$_{h}$
Name	Inhibit Time EMCY
Object code	VAR
Data Type	UNSIGNED16
Category	Optional
Note	CANopen Only

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0x00

Object 1016h - Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time and thus has to be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat.

Monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 the corresponding entry is not used. The time has to be a multiple of 1ms.

MSB		LSB
31-24	23-16	15-0
reserved (value: 00h)	Node-ID	heartbeat time
-	UNSIGNED8	UNSIGNED16

Object Description:

Index	**1016**$_h$
Name	Consumer Heartbeat Time
Object code	ARRAY
Data Type	UNSIGNED32
Category	Optional
Note	CANopen Only

Entry Description:

Sub-Index	***0***
Description	number entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x1

Sub-Index	***1***
Description	Consumer Heartbeat Time
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

This object is not active and reserved for future implementation.

Object 1017h: Producer Heartbeat Time

The producer hartbeat time defines the cycle time of the heartbeat. The producer heartbeat time is 0 if it not used. The time has to be a multiple of 1ms.

Object Description:

Index	**1017_h**
Name	Producer Heartbeat Time
Object code	VAR
Data Type	UNSIGNED16
Category	Optional
Note	CANopen Only

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0x00

This object is not active and reserved for future implementation.

Object 1018h: Identity Object

The object at index 1018h contains general information about the device.

The Vendor ID (sub-index 1h) contains a unique value allocated to each manufacturer. The manufacturer-specific Product code (sub-index 2h) identifies a specific device version.

The manufacturer-specific Revision number (sub-index 3h) consists of a major revision number and a minor revision number. The major revision number identifies a specific CANopen behavior. If the

CANopen functionality is expanded, the major revision has to be incremented. The minor revision number identifies different versions with the same CANopen behavior.

Object Description:

Index	**1018$_h$**
Name	Identity Object
Object code	ARRAY
Data Type	Identity
Category	Optional
Note	EtherCAT/CANopen

Entry Description:

Sub-Index	***0***
Description	number entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x4

Sub-Index	***1***
Description	Vendor ID
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	-

Sub-Index	***2***
Description	Product code

Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	-

Sub-Index	***3***
Description	Revision Number
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	-

Sub-Index	***4***
Description	Serial Number
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	-

Object 1029h - Error behavior

This object reports the CAN communication state after a heartbeat failure. The value of the object asserts that after such a failure, the CAN communication state is:

0: Pre-operational (only if current state is operational)

1: No state change

2: Stopped

The default value is 1 (no state change).

Object Description:

Index	**1029$_h$**
Name	Error behavior
Object code	ARRAY
Data Type	UNSIGNED8
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of error classes
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x1

Sub-Index	***1***
Description	Communication error
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	1

Object 1200h - Server SDO Parameter

This object is used to describe the SDO used on a device. The data type has the index 22h in the object dictionary. The number of supported entries in the SDO object record is specified by sub-index 0h. The values at 1h and 2h specify the COB-ID for this SDO.   Sub-index 3 gives the server of the SDO if the record describes an SDO for which the device is a client, and it gives the client of the SDO if the record describes an SDO for which the device is the server. The structure of the SDO COB-ID entry is as follows:

MSB				LSB
31	30	29	28-11	10-0
0	0	0	0	11-bit identifier

Bit number	**Value**	**Meaning**
31	0/1	1 = SDO Enabled, 0 = SDO not Enabled
30	0	Reserved (always 0)
29	0	11-bit ID (CAN 2.0A)
28-11	0	(Always 0)
10-0	x	SDO COB-ID

Object Description:

Index	**1200$_h$**
Name	Server SDO Parameter
Object code	RECORD
Data Type	SDO Parameter
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory

Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID Client->Server (rx)
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	601h…67Fh
Default Value	600h+Node-ID

Sub-Index	**2**
Description	COB-ID Server -> Client (tx)
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	581h … 5FFh
Default Value	580h+Node-ID

Object 1400h: - Receive PDO communication parameter 1

Contains the communication parameters for the PDOs the device is able to receive.

11-bit ID

MSB				LSB
31	30	29	28-11	10-0
0/1	0	0	0	11-bit identifier

Bit number	**Value**	**Meaning**
31	0/1	1 = PDO Enabled, 0 = PDO not Enabled
30	0	No RTR is allowed
29	0	11-bit ID (CAN 2.0A)
28-11	0	(Always 0)

10-0	x	PDO COB-ID

The transmission type (sub-index 2) defines the transmission/reception character of the PDO. Table following Table describes the usage of this entry. On an attempt to change the value of the transmission type to a value that is not supported by the device an abort message (abort code: 0609 0030h) is generated.

Transmission type	**PDO transmission**				
	cyclic	**acyclic**		**synchronous**	**Asynchronous**
0		x		x	
1-240	x			x	
241-251			Reserved		
254				x	
255				x	

Synchronous (transmission types 0-240 and 252) means that the transmission of the PDO shall be related to the SYNC object as described in 9.3. Preferably the devices use the SYNC as a trigger to output or actuate based on the previous synchronous Receive PDO respectively to update the data transmitted at the following synchronous Transmit PDO. Details of this mechanism depend on the device type and are defined in the device profile if applicable.

Asynchronous means that the transmission of the PDO is not related to the SYNC object. A transmission type of zero means that the message shall be transmitted synchronously with the SYNC object but not periodically.

A value between 1 and 240 means that the PDO is transferred synchronously and cyclically, the transmission type indicating the number of SYNC which are necessary to trigger PDO transmissions/receptions.

Object Description:

Index	**1401$_h$**
Name	Receive PDO parameter 2

Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1400h: 200h+NODE-ID

Sub-Index	**2**
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Object 1401h: - Receive PDO communication parameter 2

Contains the communication parameters for the PDOs the device is able to receive.

11-bit ID

MSB				LSB
31	30	29	28-11	10-0
0/1	0	0	0	11-bit identifier

Bit number	**Value**	**Meaning**
31	0/1	1 = PDO Enabled, 0 = PDO not Enabled
30	0	No RTR is allowed
29	0	11-bit ID (CAN 2.0A)
28-11	0	(Always 0)
10-0	x	PDO COB-ID

Transmission type	**PDO transmission**				
	cyclic	**acyclic**		**synchronous**	**Asynchronous**
0		x		x	
1-240	x			x	
241-251			Reserved		
254				x	
255				x	

Object Description:

Index	**1400$_h$**
Name	Receive PDO parameter 1
Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1401h: 300h+NODE-ID

Sub-Index	**2**
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO

Value Range	UNSIGNED8
Default Value	0

Object 1402h: - Receive PDO communication parameter 3

Contains the communication parameters for the PDOs the device is able to receive.

11-bit ID

MSB				LSB
31	30	29	28-11	10-0
0/1	0	0	0	11-bit identifier

Bit number	**Value**	**Meaning**
31	0/1	1 = PDO Enabled, 0 = PDO not Enabled
30	0	No RTR is allowed
29	0	11-bit ID (CAN 2.0A)
28-11	0	(Always 0)
10-0	x	PDO COB-ID

Transmission type	**PDO transmission**				
	cyclic	**acyclic**		**synchronous**	**Asynchronous**
0		x		x	
1-240	x			x	
241-251			Reserved		
254				x	
255				x	

Object Description:

Index	**1402h**
Name	Receive PDO parameter 3
Object code	RECORD
Data Type	PDO CommPar (object 0x20)

Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1402h: 400h+NODE-ID

Sub-Index	***2***
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Object 1403h: - Receive PDO communication parameter4

Contains the communication parameters for the PDOs the device is able to receive.

11-bit ID

MSB			LSB	
31	30	29	28-11	10-0
0/1	0	0	0	11-bit identifier

Bit number	**Value**	**Meaning**
31	0/1	1 = PDO Enabled, 0 = PDO not Enabled
30	0	No RTR is allowed
29	0	11-bit ID (CAN 2.0A)
28-11	0	(Always 0)
10-0	x	PDO COB-ID

Transmission type	**PDO transmission**				
	cyclic	**acyclic**		**synchronous**	**Asynchronous**
0		x		x	
1-240	x			x	
241-251			Reserved		
254				x	
255				x	

Object Description:

Index	**1403$_h$**
Name	Receive PDO parameter 4
Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***

Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1402h: 500h+NODE-ID

Sub-Index	***2***
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Object 1600h -: Receive PDO Mapping Parameter 1

Contains the mapping for the PDOs the device is able to receive. The sub-index 0h 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/received with the corresponding PDO. The sub-indices from 1h to number of entries contain the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length. All three values are hexadecimal coded. The length entry contains the length of the object in bit (1..8h). This parameter can be used to verify the overall mapping length. It is mandatory.

The structure of the entries from sub-index 1h – 8h is as follows:

MSB		LSB
Index (16 bit)	Sub-index (8 bit)	Object length (8 bit)

If the change of the PDO mapping cannot be executed (e.g. the PDO length is exceeded or the SDO client attempts to map an object that cannot be mapped) the device responds with an Abort SDO Transfer Service.

Sub-index 0 determines the valid number of objects that have been mapped. For changing the PDO mapping first sub-index 0 must be set to 0 (mapping is deactivated). Then the objects can be remapped. When a new object is mapped by writing a subindex between 1 and 8, the device may check whether the object specified by index /sub-index exists. If the object does not exist or the object cannot be mapped, the SDO transfer must be aborted with the Abort SDO Transfer Service with one of the abort codes 0602 0000h or 0604 0041h.

After all objects are mapped sub-index 0 is set to the valid number of mapped objects. When sub-index 0 is set to a value >0 the device may validate the new PDO mapping before transmitting the response of the SDO service. If an error is detected the device has to transmit the Abort SDO Transfer Service with one of the abort codes 0602 0000h, 0604 0041h or 0604 0042h.

A device that supports dynamic mapping of PDOs must support this during the state PRE-OPERATIONAL state. If dynamic mapping during the state OPERATIONAL is supported, the SDO client is responsible for data consistency.

Object Description:

Index	**1600$_h$**
Name	Receive PDO mapping parameters 1
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO

Entry Description:

Sub-Index	***0***

Description	Number of entries
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1601h -: Receive PDO Mapping Parameter 2

MSB		LSB
Index (16 bit)	Sub-index (8 bit)	Object length (8 bit)

Object Description:

Index	**1601**$_{h}$
Name	Receive PDO mapping parameters 1
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory

Access	Read/Write
PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1602h -: Receive PDO Mapping Parameter 3

MSB		LSB
Index (16 bit)	Sub-index (8 bit)	Object length (8 bit)

Object Description:

Index	**1602h**
Name	Receive PDO mapping parameters 1
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO
Note	Valid Only for CANopen (EtherCAT only support 1600-1601h)

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read/Write

PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1603h -: Receive PDO Mapping Parameter 4

MSB		LSB
Index (16 bit)	Sub-index (8 bit)	Object length (8 bit)

Object Description:

Index	**1603$_h$**
Name	Receive PDO mapping parameters 1
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO
Note	Valid Only for CANopen (EtherCAT only support 1600-1601h)

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO

Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1800h: Transmit PDO Communication Parameter 1

Contains the communication parameters for the PDOs the device is able to transmit. The type of the PDO communication parameter (20h). A detailed description of the entries is done in the section for the Receive PDO Communication Parameter (1400h – 1403h).

Index	**1800**$_{h}$
Name	Transmit PDO parameter
Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	**1**
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1800h: 180h+NODE-ID

Sub-Index	**2**
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	**3**
Description	inhibit time
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	**4**
Description	reserved
Entry Category	optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	***5***
Description	Event timer
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	0 not used - UNSIGNED16
Default Value	0

Event time

When a TPDO transmission type is 254 an event time can be used. The event occurs when the time is elapsed. The event time elapse is a multiple of 1 millisecond of subindex 5. It causes the transmission of this PDO in addition to other asynchronous events. The occurrence of an event sets the timer again. A value of 0 disables this function.

Inhibit time

Inhibit time specifications do not relate to the generating event but to the transmission of the TPDO. The inhibit time resolution is 100 microseconds. The exact inhibit times are not very accurate and can actually be up to 2 milliseconds (20 units of inhibit time) longer than defined by sub-index 3 of this object. For example, if an inhibit time is specified as 10 milliseconds, its actual inhibit time length may vary in the range of [10…12] milliseconds.

Object 1801h: Transmit PDO Communication Parameter 2

Index	**1801$_h$**
Name	Transmit PDO parameter 2
Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***

Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1800h: 280h+NODE-ID

Sub-Index	**2**
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	**3**
Description	inhibit time
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	***4***
Description	reserved
Entry Category	optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	***5***
Decriptino	Event timer
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	0 not used - UNSIGNED16
Default Value	0

Object 1802h: Transmit PDO Communication Parameter 3

Index	**1802**$_h$
Name	Transmit PDO parameter 3
Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	Index 1800h: 380h+NODE-ID

Sub-Index	**2**
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	***3***
Description	inhibit time
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	***4***
Description	reserved
Entry Category	optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	***5***
Description	Event timer
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	0 not used - UNSIGNED16
Default Value	0

Object 1803h: Transmit PDO Communication Parameter 4

Index	**1803_h**
Name	Transmit PDO parameter 4
Object code	RECORD
Data Type	PDO CommPar (object 0x20)
Category	Mandatory for supported PDO
Note	Valid Only for CANopen

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0x2

Sub-Index	***1***
Description	COB-ID used by PDO
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32

Default Value	Index 1800h: 480h+NODE-ID

Sub-Index	***2***
Description	transmission type
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	***3***
Description	inhibit time
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	***4***
Description	reserved
Entry Category	optional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	***5***
Description	Event timer
Entry Category	Optional
Access	Read/Write
PDO Mapping	NO
Value Range	0 not used - UNSIGNED16
Default Value	0

Object 1A00h : Transmit PDO Mapping Parameter 1

Contains the mapping for the PDOs the device is able to transmit. The type of the PDO mapping parameter (21h). A detailed description of the entries is done in the section for the Receive PDO Mapping Parameter (1600h – 1603h).

Object Description:

Index	**1A00$_h$**
Name	Transmit PDO mapping parameters 1
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1A01h : Transmit PDO Mapping Parameter 2

Object Description:

Index	**1A01h**
Name	Transmit PDO mapping parameters 2
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1A02h : Transmit PDO Mapping Parameter 3

Object Description:

Index	**1A02h**

Name	Transmit PDO mapping parameters 3
Object code	RECORD
Data Type	PDO Mapping
Category	Mandatory for each supported PDO
Note	Valid Only for CANopen (EtherCAT only support 1A00-1A01h)

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1A03h : Transmit PDO Mapping Parameter 4

Object Description:

Index	**1A03**$_{h}$
Name	Transmit PDO mapping parameters 4
Object code	RECORD

Data Type	PDO Mapping
Category	Mandatory for each supported PDO
Note	Valid Only for CANopen (EtherCAT only support 1A00-1A01h)

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	0 deactivated ; 1-8 activated
Default Value	0

Sub-Index	***N (1-8)***
Description	PDO mapping for the nth
Entry Category	Conditional
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 1C00h – Sync Manager Communication Type

Object Description

Index	1C00

Description	Up to 32 sync manager types can be described. The first four sync manager types are fixed, and the following can be configured to one of the first four types. The default configuration is the following: 1 mailbox receive 2 mailbox send 3 process data output 4 process data input
Note	Valid only for EtherCAT.
Object Code	Array
Data Type	Unsigned8

Entry Description

Sub-Index	000
Description	Number of entries
Access	Read Only
PDO Mapping	No
Default Value	4
Range	0 to 20

Sub-Index	001
Description	Sub-index 1
Data Type	Unsigned8
Access	Read Only
PDO Mapping	No
Default Value	1
Range	0 to 4

Sub-Index	002
Description	Sub-index 2
Data Type	Unsigned8
Access	Read Only
PDO Mapping	No
Default Value	2
Range	0 to 4

Sub-Index	003
Description	Sub-index 3
Data Type	Unsigned8
Access	Read Only
PDO Mapping	No
Default Value	3
Range	0 to 4

Sub-Index	004
Description	Sub-index 4
Data Type	Unsigned8
Access	Read Only
PDO Mapping	No
Default Value	4
Range	0 to 4

Object 1C10h – Sync Manager 0 PDO Assignment

Object Description

Index	1C10
Description	Using this object, PDOs can be assigned to the Sync Managers starting at Sync Manager 2.
Note	Valid only for EtherCAT.
Object Code	Array
Data Type	Unsigned16

Entry Description

Access	Read Only
PDO Mapping	No
Default Value	0
Range	0

Object 1C11h – Sync Manager 1 PDO Assignment

Object Description

Index	1C11
Description	Sync Manager 1 PDO Assignment
Note	Valid only for EtherCAT.
Object Code	Array
Data Type	Unsigned16

Entry Description

Access	Read Only
PDO Mapping	No
Default Value	0
Range	0

Object 1C12h – Sync Manager 2 PDO Assignment

Object Description

Index	1C12
Description	Sync Manager 2 PDO Assignment
Note	Valid only for EtherCAT.
Object Code	Array
Data Type	Unsigned16

Entry Description

Sub-Index	000
Description	Number of assigned Rx PDOs
Access	Read/Write
PDO Mapping	No
Default Value	4
Range	0 to 20

Sub-Index	001
Description	Sub-index 1
Data Type	Unsigned16
Access	Read/Write
PDO Mapping	No
Default Value	5632
Range	5632 to 6143

Sub-Index	002
Description	Sub-index 2
Data Type	Unsigned16
Access	Read/Write
PDO Mapping	No
Default Value	5633
Range	5632 to 6143

Sub-Index	003
Description	Sub-index 3
Data Type	Unsigned816
Access	Read/Write
PDO Mapping	No
Default Value	5634
Range	5632 to 6143

Sub-Index	004
Description	Sub-index 4
Data Type	Unsigned16
Access	Read/Write
PDO Mapping	No
Default Value	5635
Range	5632 to 6143

Object 1C13h – Sync Manager 3 PDO Assignment

Object Description

Index	1C12
Description	Sync Manager 3 PDO Assignment
Note	Valid only for EtherCAT.
Object Code	Array
Data Type	Unsigned16

Entry Description

Sub-Index	000
Description	Number of assigned Tx PDOs
Access	Read/Write
PDO Mapping	No
Default Value	4
Range	0 to 255

Sub-Index	001
Description	Sub-index 1
Data Type	Unsigned16
Access	Read/Write
PDO Mapping	No
Default Value	6656
Range	6656 to 7167

Sub-Index	002
Description	Sub-index 2
Data Type	Unsigned16
Access	Read/Write
PDO Mapping	No
Default Value	6657
Range	6656 to 7167

Sub-Index	003
Description	Sub-index 3
Data Type	Unsigned816
Access	Read/Write
PDO Mapping	No
Default Value	6658
Range	6656 to 7167

Sub-Index	004
Description	Sub-index 4
Data Type	Unsigned16
Access	Read/Write
PDO Mapping	No
Default Value	6659
Range	6656 to 7167

6. Manufacturer-specific Objects

Object 2000h – Manufacturer specific error register

This object show the status of internal Drive error register.

Object Description:

Index	**2000$_h$**
Name	Manufacturer specific error register CANopen error status bit
Object code	VAR
Data Type	UNSIGNED16
Category	Optional
Note	Valid Only for CANopen

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0x0

Object 2001h – EtherCAT FW Application Version

This object return the EtherCAT FW Application Version.

Object Description:

Index	**2001$_h$**
Name	EtherCAT FW Application Version
Object code	VAR
Data Type	UNSIGNED32
Category	Optional
Note	Valid Only for EtherCAT

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x0

Object 2002h – Drive FW Version

This object return the Drive FW Version.

Object Description:

Index	**2001$_h$**
Name	Drive FW Version
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0x0

Object 2100h – Velocity Error Radius

This object defines the error radius tolerance for velocity. It is related to velocity error radius time.

Object Description:

Index	**2100$_h$**
Name	Velocity Error Radius
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read /write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0x0

Object 2101h – Velocity Error Radius Time

Maximum time in which the velocity error could stay out of tolerance

Object Description:

Index	**2101$_h$**
Name	Velocity Error Radius Time
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read /Write
PDO Mapping	NO

Value Range	UNSIGNED16
Default Value	0x0

Object 2220h – Board Temperature

Temperature Value of the electronic board, in °C/1000.

Index	**2220$_h$**
Name	Board Temperature
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	INTEGER16
Default Value	0

Object 2201h – Bus Nominal Voltage

Nominal value of the DC Bus voltage (V)

Index	**2201$_h$**
Name	Bus Nominal Voltage
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read /Write

PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	48

Object 2201h – Motor Max Current

Maximum motor current to be available in the drive, in **Ampere/100**.

Index	**2202$_h$**
Name	Maximum Motor Current
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read /Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	3500
Units	A/100

Object 2205h: Analog IN

Value of analogs inputs of the drive.

Object Description:

Index	**2205$_h$**
Name	Analog IN
Object code	ARRAY
Data Type	INTEGER16
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	1
Default Value	1

Sub-Index	***1***
Description	Analog IN
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	INTEGER16
Default Value	-0

Object 2300h: User Program Data

This record handle the user program parameters for specific function of servodrive.

Object Description

Index	**2300$_h$**
Name	User Program Data
Object code	RECORD
Data Type	RECORD
Category	Optional

Entry Description

Sub-Index	**0**
Description	Number of Entries

Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	6

Sub-Index	**1**
Description	Enable User Program
Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	1

Sub-Index	**2**
Description	User Program Type
Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	3
Description	User Program Axis Sel
Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED8
Default Value	0

Sub-Index	4
Description	User Program Param 0
Entry Category	mandatory

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	5
Description	User Program Param 1
Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	6
Description	User Program Param 2
Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Object 2F80h – Boot NMT Status

With this object is possible to select the Boot NMT status.

Object Description:

Index	**2F80$_h$**
Name	Boot NMT Status
Object code	VAR
Data Type	UNSIGNED16
Category	Optional
Valid only for EtherCAT.	Valid only for CANopen

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	-
Default Value	0

List of possible values:

Value	**Description**
0x00	Pre-operational
0x04	Operational

Object 2F81h – Status word masking

For compatibility reasons it is possible to use this mask to hide the Manufacturer specific bit of the Status word. The mask affects all status word, please change only the 2 most significative bits if needed.

Object Description:

Index	**2F81$_h$**
Name	Status word masking
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	65535

Object 2F82h – Switch to Index

Switch to Index parameter,

Object Description:

Index	**2F82$_h$**
Name	Switch to Index
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	0

Object 2F83h – I2T Protection Type

Type of protection for I2T of the motor.

Possible Values

Value	**Meaning**
0	Fault on I2T protection
1	Motor self limits its current to the nominal value

Object Description:

Index	**2F83$_h$**
Name	I2T Type
Object code	VAR

Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Object 2FF0h – EtherCAT FW Update Start

This object starts the FW udate for the EthrCAT stack once is activated. Please follow the FW upgrade procedure and do not start it without follow it.

Possible Values

Object Description:

Index	**2FF0**$_h$
Name	EtherCAT FW Upgrade Start
Object code	VAR
Data Type	BOOL
Category	Optional
Note	Valid Only For EtherCAT

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	BOOL
Default Value	FALSE

Object 2FF1h – Drive FW Update Start

This object starts the FW udate for the Drive once is activated. Please follow the FW upgrade procedure and do not start it without follow it.

Possible Values

Object Description:

Index	**2FF1h**
Name	Drive FW Upgrade Start
Object code	VAR
Data Type	BOOL
Category	Optional
Note	Valid Only For EtherCAT

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	BOOL
Default Value	FALSE

7. Standard Servodrive Objects

General Information

The device control function block controls all functions of the drive (drive function and power section).

It is divided into:

- device control of the state machine
- operation mode function

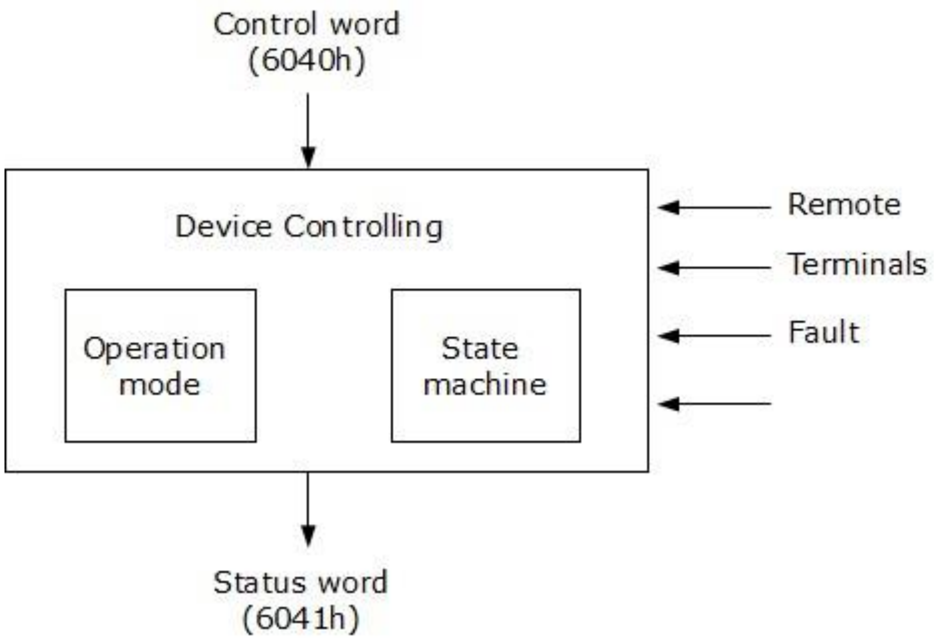


Figura 16 Device Control

The state of the device can be controlled by the *controlword*.

The state of the device is shown in the *statusword*.

In remote mode the device is controlled directly from the CANopen network by PDO and SDO.

The state machine is controlled externally by the controlword and external signals. The state machine is also controlled by internal signals like faults and modes of operation.

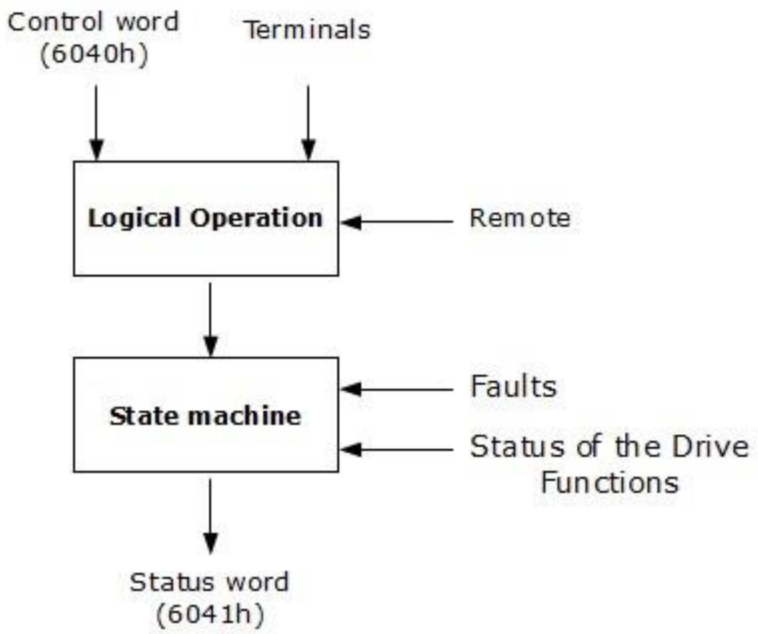


Figura 17- Device Control Flow

The Device is always in remote mode, thus the relative bit in the status word is always set to 1.

State machine

The state machine describes the device status and the possible control sequence of the drive. A single state represents a special internal or external behavior. The state of the drive also determines which commands are accepted. E.g. it is only possible to start a point-to-point move when the drive is in state OPERATION ENABLE.

States may be changed using the controlword and/or according to internal events. The current state can be read using the statusword.

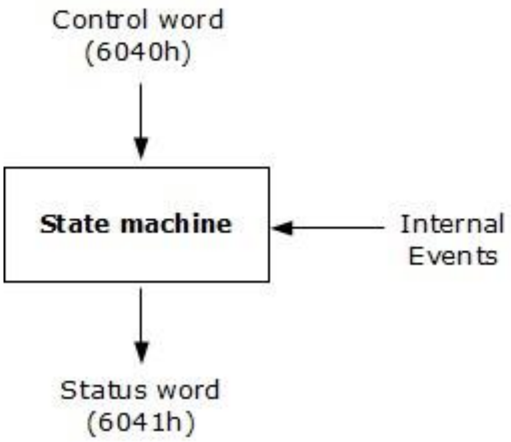


Figura 18-Flow Diagram

The state machine describes the state machine of the device with respect to control of the power electronics because of user commands and internal drive faults.

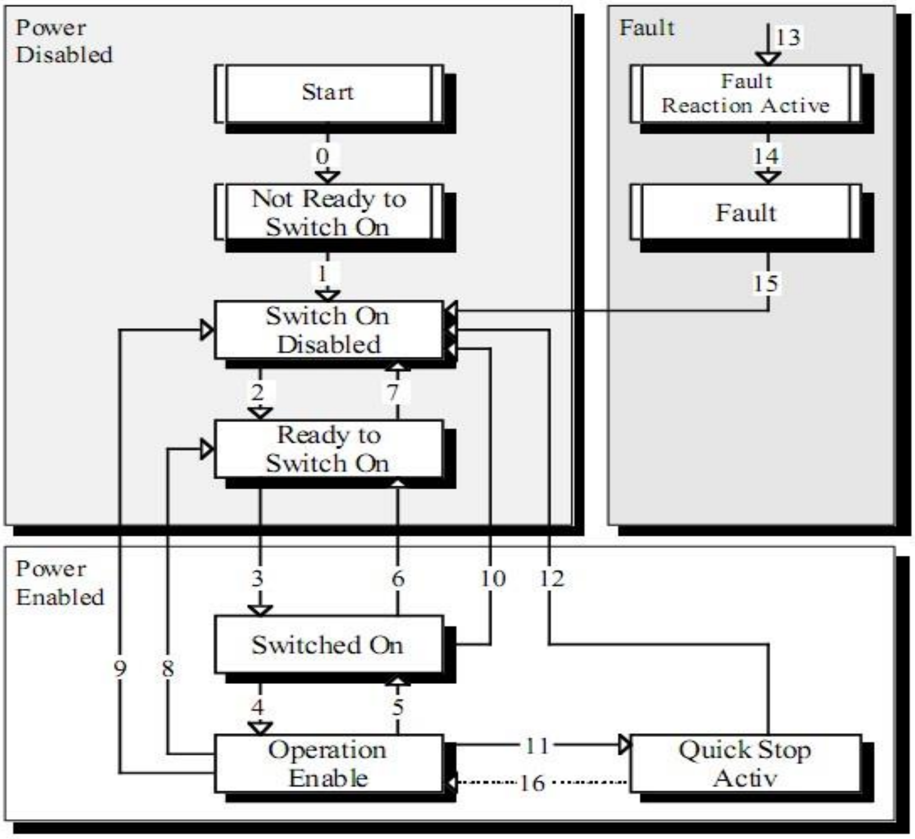


Figura 19- State Machine

The following states of the device are possible:

- **NOT READY TO SWITCH ON:**

  Low level power (e.g. ± 15V, 5V) has been applied to the drive.

  The drive is being initialized or is running self test.

  A brake, if present, has to be applied in this state.

  The drive function is disabled.

- **SWITCH ON DISABLED:**

  Drive initialization is complete.

  The drive parameters have been set up.

  Drive parameters may be changed.

  High voltage may not be applied to the drive, (e.g. for safety reasons). The drive function is disabled.

- **READY TO SWITCH ON:**

  High voltage may be applied to the drive.

  The drive parameters may be changed.

  The drive function is disabled.

- **SWITCHED ON:**

  High voltage has been applied to the drive.

  The power amplifier is ready.

  The drive parameters may be changed.

  The drive function is disabled.

- **OPERATION ENABLE:**

  No faults have been detected.

  The drive function is enabled and power is applied to the motor.

  The drive parameters may be changed.

  (This corresponds to normal operation of the drive.)

- **QUICK STOP ACTIVE:**

  The drive parameters may be changed.

  The quick stop function is being executed.

The drive function is enabled and power is applied to the motor. o If the quick stop option code is switched to 5 (stay in the state QUICK STOP ACTIVE), you can't leave the state QUICK STOP ACTIVE, but you can transmit to the state OPERATION ENABLE with the command 'Enable Operation'.

➢ **FAULT REACTION ACTIVE:**

The drive parameters may be changed.

A fault has occurred in the drive.

The quick stop function is being executed.

The drive function is enabled and power is applied to the motor.

➢ **FAULT:**

The drive parameters may be changed.

A fault has occurred in the drive.

High voltage switch-on/-off depends on the application. The drive function is disabled.

State transitions are caused by internal events in the drive or by commands from the host via the controlword.

State Transition 0: START ⇒ NOT READY TO SWITCH ON Event: Reset.

Action: The drive self-tests and/or self-initializes.

State Transition 1: NOT READY TO SWITCH ON ⇒ SWITCH ON DISABLED Event: The drive has self-tested and/or initialized successfully.

Action: Activate communication.

State Transition 2: SWITCH ON DISABLED ⇒ READY TO SWITCH ON Event: 'Shutdown' command received from host.

Action: None

State Transition 3: READY TO SWITCH ON ⇒ SWITCHED ON Event: 'Switch On' command received from host.

Action: The power section is switched on if it is not already switched on. **State Transition 4:** SWITCHED ON ⇒ OPERATION ENABLE Event: 'Enable Operation' command received from host. Action: The drive function is enabled.

State Transition 5: OPERATION ENABLE ⇒ SWITCHED ON Event: 'Disable Operation' command received from host. Action: The drive operation will be disabled.

State Transition 6: SWITCHED ON ⇒ READY TO SWITCH ON Event: 'Shutdown' command received from host. Action: The power section is switched off.

State Transition 7: READY TO SWITCH ON ⇒ SWITCH ON DISABLED

Event: 'Quick Stop' and 'Disable Voltage' command received from host. Action: None

State Transition 8: OPERATION ENABLE ⇒ READY TO SWITCH ON Event: 'Shutdown' command received from host.

Action: The power section is switched off immediately, and the motor is free to rotate if unbraked.

State Transition 9: OPERATION ENABLE ⇒ SWITCH ON DISABLED Event: 'Disable Voltage' command received from host.

Action: The power section is switched off immediately, and the motor is free to rotate if unbraked.

State Transition 10: SWITCHED ON ⇒ SWITCH ON DISABLED

Event: 'Disable Voltage' or 'Quick Stop' command received from host.

Action: The power section is switched off immediatly, and the motor is free to rotate if unbraked.

State Transition 11: OPERATION ENABLE ⇒ QUICK STOP ACTIVE Event: 'Quick Stop' command received from host.

Action: The quick stop function is executed.

State Transition 12: QUICK STOP ACTIVE ⇒ SWITCH ON DISABLED

Event: 'Quick Stop' is completed or 'Disable Voltage' command received from host.

This transition is possible, if the Quick-Stop-Option-Code is different 5 (stay in the state ‘Quick Stop Active’).

Action: The power section is switched off.

State Transition 13: All states ⇒ FAULT REACTION ACTIVE A fault has occurred in the drive.

Action: Execute appropriate fault reaction.

State Transition 14: FAULT REACTION ACTIVE ⇒ FAULT Event: The fault reaction is completed.

Action: The drive function is disabled. The power section may be switched off.

State Transition 15: FAULT ⇒ SWITCH ON DISABLED Event: 'Fault Reset' command received from host.

Action: A reset of the fault condition is carried out if no fault exists currently on the drive. After leaving the state Fault the Bit 'Fault Reset' of the controlword has to be cleared by the host.

State Transition 16: QUICK STOP ACTIVE ⇒ OPERATION ENABLE

Event: 'Enable Operation' command received from host. This transition is possible if the Quick-Stop-Option-Code is 5, 6, 7 or 8 (→ Chapter 10.3.5). Action: The drive function is enabled.

Object Description

Object 603Fh: Fault Register

This object returns the error code of the last error that occurred.

Object Description:

Index	**603F$_h$**
Name	Fault Register
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER32
Default Value	-

Object 6040h: Controlword

The controlword consist of bits for:

- the controlling of the state,
- the controlling of operating modes and
- manufacturer specific options.

Object Description:

Index	**6040$_h$**
Name	Control word
Object code	VAR
Data Type	UNSIGNED16

Category	Mandatory

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	-

The bits of the controlword are defined as follows:

15-11	10-9	8	7	6-4	3	2	1	0
Manufacturer specific	reserved	halt	Faul reset	Operation mode specific	Enable operation	Quick Stop	Enable voltage	Switch on
O	O	O	M	O	M	M	M	M
MSB						LSB		

O – Optional

M – Mandatory

Bits 0-3 and 7:

Device control commands are triggered by the following bit patterns in the controlword:

Command	Bits of the Controlword					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	x	1	1	0	2,6,8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
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		x	x	x	x	15

Bits marked with X are not relevant.

ƒ * The drive executes the functionality of SWITCH_ON.

ƒ ** The drive does nothing in this state, which is treated the same as in *

Bits 4,5,6 and 8:

Bit		Operation mode				
	Velocity mode	**Profile position mode**	**Profile velocity mode**	**Profile torque mode**	**Homing mode**	**Interpolation position mode**
4	rfg enable	New set-point	Reserved	Reserved	Homing operation start	Enable ip mode
5	rfg unlock	Change set immediately	Reserved	Reserved	Reserved	Reserved
6	rfg use ref	abs/rel	Reserved	Reserved	Reserved	Reserved
8	Halt	Halt	Halt	Halt	Halt	Halt

Bits: 9,10:

These bits are reserved for further use. They are inactive by setting to zero. If they have no special function, they must be set to zero.

Bits 11, 12, 13, 14 and 15:

These bits are manufacturer specific.

Object 6041h: Statusword

The statusword indicates the current state of the drive. No bits are latched. The statusword consist of bits for: • the current state of the drive,

- the operating state of the mode and
- manufacturer specific options.

Object Description:

Index	**6041h**
Name	Status word
Object code	VAR
Data Type	UNSIGNED16
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	-

The bits in the status word has the following meaning:

bit	**Description**	**M/O**
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	Fault	M
4	Voltage enabled	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacturer specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12	Operation mode specific	M
13	Operation mode specific	M
14	MS – not used	O

15	MS – Homing reached	O

Bits 0-3, 5 and 6:

The following bits indicate the status of the device:

Value (binary)	**State**
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Bit 4: VOLTAGE ENABLED
High voltage is applied to the drive when this bit is set to 1.

Bit 5: QUICK STOP
When reset, this bit indicates that the drive is reacting on a quick stop request. Bits 0, 1 and 2 of the statusword must be set to 1 to indicate that the drive is capable to regenerate. The setting of the other bits indicates the status of the drive (e.g. the drive is performing a quick stop as result of a reaction to a non-fatal fault. The fault bit is set as well as bits 0, 1 and 2).

Bit 7: WARNING
A drive warning is present if bit 7 is set. The cause means no error but a state that has to be mentioned, e.g. temperature limit, job refused. The status of the drive does not change. The cause of this warning may be found by reading the fault code parameter. The bit is set and reset by the device.

Bit 8: MANUFACTURER SPECIFIC Not used.

Bit 9: REMOTE
Always set to 1.

Bit 10: TARGET REACHED
If bit 10 is set by the drive, then a set-point has been reached. The set-point is dependent on the operating mode. The description is situated in the chapter of the special mode. The change of a target value by software alters this bit.

If quick stop option code is 5, 6, 7 or 8, this bit must be set, when the quick stop operation is finished and the drive is halted.

If halt occurred and the drive has halted then this bit is set too.

Bit 11: INTERNAL LIMIT ACTIVE
This bit set by the drive indicates, that an internal limitation is active (e.g. position range limit).

Bit 12 and 13:
These bits are operation mode specific. The description is situated in the chapter of the special mode.

The following table gives an overview:

Bit			Operation mode			
	vl	**pp**	**pv**	**tq**	**hm**	**ip**
12	reserved	Set point acknowledge	Speed	reserved	Homing attained	Ip mode active
13	reserved	Following error	Following error	reserved	Homing error	reserved

Bit 14 and 15:
14 - Not used

15 - Homing is Valid.

Object 605Ah: Quick Stop Option Code

The object contains the action that is to be executed on a transition of the CiA 402 Power State Machine to the Quick Stop active state.

Object Description:

Index	**6060$_h$**
Name	Quick Stop Option Cod
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	INTEGER16
Default Value	2

List of possible operation modes:

Value	**Description**
0	Immediate stop with subsequent state change to Switch on disabled
2	Braking with quick stop ramp (6085h) and subsequent state change to Switch on disabled
6	Braking with slow down ramp (deceleration ramp depending on operating mode) and subsequent state change to Quick stop active; control does not switch off and the motor remains energized. You can switch back to the Operation enabled state.

Object 6060h: Modes of operation

The parameter modes of operation switches the actually chosen operation mode.

Object Description:

Index	**6060$_h$**
Name	Modes of operation
Object code	VAR
Data Type	INTEGER8
Category	Mandatory

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	INTEGER8
Default Value	-

List of possible operation modes:

Value	**Description**
-4	Tuning Current Mode
0	Reserved
1	Profile Position Mode
3	Profile Velocity Mode
4	Torque Profile Mode
5	Reserved
6	Homing Mode
7	Interpolated Position Mode
8	Cyclic Synchronous Position Mode

Note:

A read of modes of operation shows only the value of modes of operation. The actual mode of the drive is reflected in the object modes of operation display. It may be changed by writing to modes of operation.

Object 6061h: Modes of operation display

The modes of operation display show the current mode of operation. The meaning of the returned value corresponds to that of the modes of operation option code (index 6060h).

Object Description:

Index	**6061$_h$**
Name	Modes of operation display
Object code	VAR
Data Type	INTEGER8
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER8
Default Value	-

Object 6062h: Position demand value

The position demand value is given in position units.

Object Description:

Index	**6062$_h$**
Name	Position demand value
Object code	VAR
Data Type	UNSIGNED32

Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 6063h: Position actual value

The actual value of the position measurement device is one of the two input values of the closed loop position control. The data unit is defined as increments.

Object Description:

Index	**6063_h**
Name	Position actual value
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 6064h: Position actual value

The actual value of the position measurement in user units.

Object Description:

Index	**6064$_h$**
Name	Position actual value
Object code	VAR
Data Type	UNSIGNED32
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 6065h: Following error window

The following error window defines a range of tolerated position values symmetrically to the position demand value. As it is in most cases used with user defined units, a transformation into increments with the position factor is necessary. If the position actual value is out of the following error window, an error occur.

Object Description:

Index	**6065$_h$**
Name	Following error window
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 6066h: Following error time out

When a following error occurs longer than the defined value of the time-out given in multiples of milliseconds, the corresponding bit 13 following error in the status word will be set to one. The reaction of the drive when a following error occurs, is manufacturer specific.

Object Description:

Index	**6066h**
Name	Following error time out
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0

Object 6067h: Position window

The position window defines a symmetrical range of accepted positions relatively to the target position. If the actual value of the position encoder is within the position window, this target position is regarded as reached. As the user mostly prefers to specify the position window in his application in user defined units, the position factor must be used to transform this value into increments. The target position has to be handled in the same manner as in the Trajectory Generator concerning limiting functions and transformation into internal machine units before it can be used with this function.

Object Description:

Index	**6067$_h$**
Name	Position window
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 6068h : Position window time

When the actual position is within the position window during the defined position window time which is given in multiples of milliseconds, the corresponding bit 10 target reached in the statusword will be set to one.

Object Description:

Index	**6068$_h$**
Name	Position window time
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES

Value Range	UNSIGNED16
Default Value	0

Object 6069h: Velocity sensor actual value

The velocity sensor actual value describes the value read from a velocity encoder.

Object Description:

Index	**6069**$_h$
Name	Velocity sensor actual value
Object code	VAR
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

Object 606Bh: Velocity demand value

The output value of the trajectory generator may be corrected by the output value of the position control function. It is then provided as a demand value for the velocity controller and given in the velocity units.

Object Description:

Index	**606B**$_h$
Name	Velocity demand value
Object code	VAR
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

Object 606Ch : Velocity actual value

The velocity actual value is also represented in velocity units and is coupled to the velocity used as input to the velocity controller.

Object Description:

Index	**606C$_h$**
Name	Velocity actual value
Object code	VAR
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

Object 606Dh: Velocity window

The velocity window monitors whether the required process velocity has been achieved after an eventual acceleration or deceleration (braking) phase. It is given in velocity units.

Object Description:

Index	**606D$_h$**
Name	Velocity window
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0

Object 606Eh: Velocity window time

The corresponding bit 10 target reached is set in the statusword when the difference between the target velocity and the velocity actual value is within the velocity window longer than the velocity window time. The value of the velocity window time is given in multiples of milliseconds.

Object Description:

Index	**606E$_h$**
Name	Velocity window time
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0

Object 606Fh: Velocity threshold

As soon as the velocity actual value exceeds the velocity threshold longer than the velocity threshold time bit 12 velocity = 0 is reset in the statusword. Below this threshold the bit is set and indicates that the axle is stationary. The value is given in velocity units.

Object Description:

Index	**606F$_h$**
Name	Velocity threshold
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0

Object 6070h: Velocity threshold time

The velocity threshold time is given in multiples of milliseconds.

Object Description:

Index	**606F$_h$**
Name	Velocity threshold time
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0

Object 6071h: Target Torque

Input value for torque controller in profile torque mode. It is expressed in **Ampere/100**.

Object Description:

Index	**6071$_h$**
Name	Target Torque
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	INTEGER16
Default Value	0
Units	Ampere/100

Object 6073h: Torque Limit

Maximum current that the driver could deliver to the motor in **(6073h*In)/1000**, thousandth of the nominal current. It is related to the I2t protection.

Object Description:

Index	**6073$_h$**

Name	Target Torque
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	3000
Units	thousandth of In

Object 6074h: Iq Demand

The output value for the current controller.

Object Description:

Index	**6074$_h$**
Name	Iq Demand
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER16
Default Value	0
Units	Ampere/100

Object 6075h: Nominal Current

Contains the rated current of the motor in **Ampere/100**.
Object Description:

Index	**6075$_h$**
Name	Target Torque
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	800
Units	A/100

Object 6077h: Actual Torque

Actual torque active in the motor, in thousandth of the nominal current (6077h*In)/1000.
For calculating the torque use the Kt of the motor with active current.
Object Description:

Index	**6077$_h$**
Name	Actual Torque
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description:

Access	Re Only
PDO Mapping	YES

Value Range	UNSIGNED32
Default Value	0
Units	thousandth of In

Object 6078h: Actual Current

Actual current active in the motor, in thousandth of the nominal current (6077h*In)/1000.
Object Description:

Index	**6078**$_h$
Name	Actual Torque
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0
Units	thousandth of In

Object 6079h: DC Bus Voltage

The Bus voltage measured by the sensor. It is the instantaneous power voltage DC Link active at the driver power stage.
Object Description:

Index	**6079**$_h$
Name	DC Bus Voltage
Object code	VAR
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0
Units	millivolts

Object 607Ah: Target Position

The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The target position is given in user defined position units. It is converted to position increments using the position factor. The target position will be interpreted as absolute or relative depending on the ‘abs / rel' flag in the control word.

Object Description:

Index	**607A$_h$**
Name	Target position
Object code	VAR
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	INTEGER32
Default Value	-

Object 607Bh: Position range limit

Position range limit contains two sub-parameters, min position range limit and max position range limit. These limit the numerical range of the input value. On reaching or exceeding these limits, the input value automatically wraps to the other end of the range. Wrap-around of the input value can be prevented by setting software position limits.

Object Description:

Index	**607B$_h$**
Name	Position range limit
Object code	ARRAY
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	Min position range limit
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	INTEGER32
Default Value	-2^{31}

Sub-Index	***2***

Description	Max position range limit
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	INTEGER32
Default Value	2^{31}-1

Object 607Ch: Home offset

The home offset object is the difference between the zero position for the application and the machine home position (found during homing), it is measured in position units. During 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. All subsequent absolute moves shall be taken relative to this new zero position. This is illustrated in the following diagram.

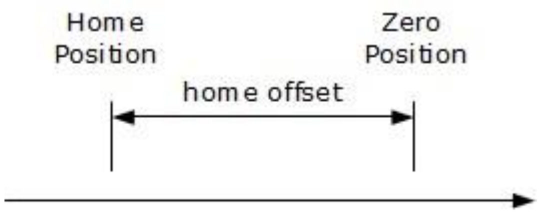


Figura 20 Home Offset description

If the home offset is not implemented, then it shall be zero.

Object Description:

Index	**607Ch**
Name	Home offset
Object code	VAR
Data Type	INTEGER32
Category	Optional

Entry Description:

Access	Read/Write

PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

Object 607Dh: Software position limit

Software position limit contains the sub-parameters min position limit and max position limit. These parameters define the absolute position limits for the position demand value and the position actual value. Every new target position must be checked against these limits. The limit positions are specified in position units (same as target position) and are always relative to the machine home position.

Object Description:

Index	**607D$_h$**
Name	Software position limit
Object code	ARRAY
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	Min position limit
Entry Category	Mandatory
Access	Read/Write

PDO Mapping	NO
Value Range	INTEGER32
Default Value	-2^{31}

Sub-Index	**2**
Description	Max position limit
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	INTEGER32
Default Value	$2^{31}-1$

Object 607Eh: Polarity

This Object shall indicate if the position and velocity demand value shall be multiplied by 1 or -1.

- A value of 0 means multiply by 1
- A value of 1 means multiply by -1

Object Description:

Index	**607e$_h$**
Name	Polarity
Object code	VAR
Data Type	UNSIGNED8
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	no
Value Range	0-1
Default Value	0

Object 6080h: Max Motor Velocity

The maximum speed allowed for the motor in either direction. It is used to protect the motor and is taken from the motor data sheet.

Object Description:

Index	**6080$_h$**
Name	Max Motor Velocity
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	3000 (User Units)

Object 6081h: Profile velocity

The profile velocity is the velocity normally attained at the end of the acceleration ramp during a profiled move and is valid for both directions of motion. The profile velocity is given in user defined speed units. It is converted to position increments per second using the velocity encoder factor.

Object Description:

Index	**6081$_h$**
Name	Profile velocity
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 6083h: Profile acceleration

The profile acceleration is given in user defined acceleration units. It is converted to position increments per second2 using the normalizing factors (see chapter 11).

Object Description:

Index	**6083**$_h$
Name	Profile acceleration
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	2500

Object 6084h: Profile deceleration

The profile deceleration is given in the same units as profile acceleration. If this parameter is not supported, then the profile acceleration value is also used for deceleration.

Object Description:

Index	**6084**$_h$
Name	Profile deceleration

Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	2500

Object 6085h: Quick stop deceleration

The quick stop deceleration is the deceleration used to stop the motor if the 'Quick Stop' command is given. The quick stop deceleration is given in the same units as the profile acceleration.

Object Description:

Index	**6085$_h$**
Name	Quick stop deceleration
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	2500

Object 6086h: Motion profile type

The motion profile type is used to select the type of motion profile used to perform a profiled move.

Object Description:

Index	**6086h**
Name	Motion profile type
Object code	VAR
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

List of possible values:

Value	Description
-32768 -1	Manufacturer specific
0	Linear ramp (trapezoidal profile)
1	Not used
2	Not used
3	Not used
4 .. 32767	reserved

Object 608Fh: Position Encoder Resolution

The resolution of the motor encoder in number of lines per revolution of the motor.
The drive parameters must be saved and the driver has to be reset whenever this object is modified.
Object Description:

Index	**608F$_h$**
Name	Position Encoder Resolution
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	4096

Object 6093h: Position Conversion Factor

The object position_factor converts all values of length of the application from position_units into the internal unit increments
It consists of numerator and divisor:
Object Description:

Index	**6093$_h$**
Name	Position Conversion Factors
Object code	ARRAY
Data Type	UNSIGNED32
Category	Optionaò

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2

Default Value	2

Sub-Index	***1***
Description	Numerator
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	1

Sub-Index	**2**
Description	Divisor
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	1

Object 6096h: Velocity Conversion Factor

The object velocity_factor converts all values of length of the application from velocity_units into the internal unit increments
It consists of numerator and divisor:
Object Description:

Index	**6096h**
Name	Velocity Conversion Factors
Object code	ARRAY
Data Type	UNSIGNED32
Category	Optionaò

Entry Description:

Sub-Index	***0***

Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	Numerator
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	4096

Sub-Index	***2***
Description	Divisor
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	60
Value Range	UNSIGNED32
Default Value	1

Object 6097h: Acceleration Conversion Factor

The object acceleration factor converts all values of length of the application from acceneleration_units into the internal unit increments
It consists of numerator and divisor:
Object Description:

Index	**6097**$_h$
Name	Acceleration Conversion Factors
Object code	ARRAY
Data Type	UNSIGNED32

Category	Optionaò

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	Num
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	4096

Sub-Index	***2***
Description	Divisor
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	60

Object 6098h: Homing method

The homing method object determines the method that will be used during homing.

Object Description:

Index	**6098_h**

Name	Homing method
Object code	VAR
Data Type	INTEGER8
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	INTEGER8
Default Value	0

List of possible values:

Value	Description
-17	See the functional description
-18	See the functional description
0	No homing operation required
1 - 35	See the functional description
36 - 127	reserved

Object 6099h: Homing speeds

This entry in the object dictionary defines the speeds used during homing and is given velocity units.

Object Description:

Index	**6099h**
Name	Homing speeds
Object code	ARRAY
Data Type	UNSIGNED32
Category	Mandatory

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	Speed during search for switch
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Sub-Index	***2***
Description	Speed during search for zero
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 609Ah: Homing acceleration

The homing acceleration establishes the acceleration to be used for all accelerations and decelerations with the standard homing modes and is given in acceleration units.

Object Description:

Index	**$609A_h$**

Name	Homing acceleration
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 60C0h: Interpolation Sub Mode Select

This object reflects or changes the actual chosen interpolation mode, selected by the user. The interpolation sub-modes can be changed only when the interpolated mode is inactive.

Object Description

Index	**60C0$_h$**
Name	Interpolated Sub Mode Select
Object code	VAR
Data Type	INTEGER16
Category	Optional

Entry Description

Access	Read/Write
PDO Mapping	NO
Value Range	0
Default Value	0 (Linear Interpolation)

Data Description

Value	**Description**
-3,...,-32768	Reserved
-2	Linear Interpolation no Buffering
-1	Reserved
0	Linear Interpolation

Object 60C1h: Interpolation data record

This object is the data words, which are necessary for performing the interpolation algorithm. The interpretation of the data words may vary with the different possible interpolation modes as set by 60C0h.

Object Description

Index	**60C0$_h$**
Name	Interpolation data record
Object code	60C0h = 0 : INTEGER32 60C0h > 0 : not defined
Data Type	INTEGER16
Category	Optional

Entry Description

Sub-Index	**0**
Description	Number of Entries
Entry Category	mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	1
Default Value	1

Sub-Index	**1**
Description	Parameter of IP Function
Entry Category	mandatory

Access	Read/Write
PDO Mapping	YES
Value Range	60C0h = 0 : INTEGER32 60C0h > 0 : not defined
Default Value	No

Object 60C2h: Interpolation time period

This object is used to define the relative time taken between two set points for the interpolation position modes. The interpolation time unit is given in $10^{\text{interpolation time index}}$ seconds.

The interpolation time period can be changed only when the interpolated mode is inactive.

Object Description

Index	**60C2$_h$**
Name	Interpolation time period
Object code	RECORD
Data Type	Interpolation time period record (0080h)
Category	Optional

Entry Description

Sub-Index	**0**
Description	Number of Entries
Entry Category	mandatory
Access	Read Only
PDO Mapping	YES
Value Range	2
Default Value	2

Sub-Index	**1**
Description	Interpolation time units
Entry Category	mandatory

Access	Read/Write
PDO Mapping	YES
Value Range	0,26
Default Value	1

Sub-Index	**2**
Description	Interpolation time index
Entry Category	mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	-128,63 (-3 ONLY)
Default Value	-3

Object 60C3h: Interpolation sync definition

Devices in the interpolation position mode often interact with other devices. Therefore it is necessary to define a communication object which is used to synchronize these interactions.

This can be done by the general Sync, or a specific group-sync-signal. Each reception of this trigger-signal or a specified number of occurrences of the trigger-signal can synchronize the devices; a second opportunity is to use fixed time slices for synchronization.

Object Description

Index	**60C3**$_h$
Name	Interpolation sync definition
Object code	ARRAY
Data Type	UNSIGNED8
Category	Optional

Entry Description

Sub-Index	**0**
Description	Number of Entries
Entry Category	mandatory
Access	Read Only
PDO Mapping	YES
Value Range	2
Default Value	2

Sub-Index	**1**
Description	Synchronize on Group
Entry Category	mandatory
Access	Read Only
PDO Mapping	NO
Value Range	0 .. 255
Default Value	0

Sub-Index	**2**
Description	IP Sync every n event
Entry Category	mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	0 .. 255
Default Value	1

Description of synchronize on group:

Value	**Description**
0	General Sync is Used
1 .. 255	Reserved

Object 60C4h: Interpolation data configuration

It is possible to offer different algorithms of interpolation. Most of them need a larger number of position to calculate the actual position the axles should reach. To enable the device to receive the needed data in advance a data space is used to store the positions and further data send by the host.

Object Description

Index	**60C4$_h$**
Name	Interpolation data configuration
Object code	RECORD
Data Type	Interpolation data configuration record (0081h)
Category	Optional

Entry Description

Sub-Index	**0**
Description	Number of Entries
Entry Category	mandatory
Access	Read Only
PDO Mapping	NO
Value Range	6
Default Value	6

Sub-Index	**1**
Description	Maximum buffer size
Entry Category	mandatory
Access	Read Only
PDO Mapping	NO
Value Range	UNSIGNED32
Default Value	1

Sub-Index	**2**
Description	Actual buffer size
Entry Category	mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Sub-Index	**3**
Description	Buffer organization
Entry Category	mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	0 .. 255
Default Value	0

Possible values for Buffer organization:

Value	**Description**
0	FIFO buffer
1	RING Buffer
2 .. 255	Reserved

Sub-Index	**4**
Description	Buffer position
Entry Category	mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED16
Default Value	0

Sub-Index	**5**

Description	Size of data record
Entry Category	mandatory
Access	Write Only
PDO Mapping	YES
Value Range	1 .. 254
Default Value	1

Sub-Index	**6**
Description	Buffer clear
Entry Category	mandatory
Access	Write Only
PDO Mapping	YES
Value Range	UNSIGNED8
Default Value	0

Possible values for Clear buffer:

Value	**Description**
0	Clear Input Buffer Access disabled Clear all ip data records
1	Enable access to the input buffer for the drive functions
2 .. 255	Reserved

Object 60F4h: Following error actual value

This object represents the actual value of the following error, it is given in user defined position units.

Object Description:

Index	**60F4$_h$**
Name	Following error actual value

Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 60F6h: Current control parameter set

In order to control the behavior of the current control loop, one or more parameters are necessary.

This object defines a rudimentary set of three parameters for a PI-control which may be enlarged by the manufacturer up to 255 parameters.

Object Description:

Index	**60F6$_h$**
Name	Current control parameter set
Object code	ARRAY
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2

Default Value	2

Sub-Index	***1***
Description	KP Current loop
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	**2**
Description	KI Current loop
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Object 60F9h: Velocity control parameter set

In order to control the behavior of the velocity control loop, one or more parameters are necessary.

This object defines a rudimentary set of three parameters for a PI-control which may be enlarged by the manufacturer up to 255 parameters.

Object Description:

Index	**60F9$_h$**
Name	Velocity control parameter set
Object code	ARRAY
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	KP Velocity loop
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Sub-Index	**2**
Description	KI Velocity loop
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Object 60FAh: Control effort

The output of the position control loop is the control effort. It is particular to the Position control function that the notation of the control effort is mode dependent and therefore not specified in the object description.

Object Description:

Index	**60FA$_h$**

Name	Control effort
Object code	VAR
Data Type	INTEGER32
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

Object 60FBh: Position control parameter set

The object define the manufacturer-specific object for the position control loop

Object Description:

Index	**60FB$_h$**
Name	Position control parameter set
Object code	ARRAY
Data Type	UNSIGNED16
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	KP Position loop
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	NO
Value Range	UNSIGNED16
Default Value	0

Object 60FDh: Digital Inputs

Indicates the state of the digital inputs. The digital inputs object has 32 bits.
The first 16 bits (bits 0-15) indicate the status of various types of switches. Those switches are functions assigned to some of the digital inputs (**not predefined, functions have to be assigned via UI**).

bit 0 = negative limit switch

- If bit is 0 on, the digital input assigned to the negative limit switch is on.
- If bit 0 is off, the digital input assigned to the negative limit switch is off.

bit 1 = positive limit switch

- If bit 1 is on, the digital input assigned to the positive limit switch is on.
- If bit 1 is off, the digital input assigned to the positive limit switch is off.

bit 2 = home switch

- If bit 2 is on, the digital input assigned to the home switch is on.
- If bit 2 is off, the digital input assigned to the home switch is off.

The last 16 bits indicate the status of each digital input, regardless of the input's functionality.

bit 16 = digital input 1

bit 17 = digital input 2

bit 18 = digital input 2

bit 19 = digital input 2

The bit values have the following meaning: 0 = switch is off

1 = switch is on Thus, for example:

If digital input 1 is on, bit 16 is set.

If digital input 2 is on, bit 17 is set.

If digital input 3 is on, bit 18 is set.

Object Description:

Index	**60FD$_h$**
Name	DIGITAL INPUTS
Object code	VAR
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	YES
Value Range	INTEGER32
Default Value	UNSIGNED32

Object 60FEh: Digital Outputs

Indicates the state of the digital outputs.

Sub-index 1 of this object indicates the state of the digital outputs. This sub-index has 32 bits.

The bits in the first word (bits 0-15) indicates the status of the brake.

- **bit 0** = brake*****

If bit 0 is on, the digital output assigned to the brake is on.

If bit 0 is off, the digital output assigned to the brake is off.

The bits in the second word (bits 16-31) indicate the state of each digital output, regardless of the output's functionality.

- **bit 16** = digital output 1
- **bit 17** = digital output 2

The bit values have the following meaning:

- 0 = off
- 1 = on

For example, to read the status of digital output 1 (regardless of its functionality; it can be idle), read bit 16.

The SubIndex 2 Output mask is not implemented by now.

Object Description:

Index	**60FE$_h$**
Name	Digital Outputs
Object code	ARRAY
Data Type	UNSIGNED32
Category	Optional

Entry Description:

Sub-Index	***0***
Description	Number of entries
Entry Category	Mandatory
Access	Read Only
PDO Mapping	NO
Value Range	2
Default Value	2

Sub-Index	***1***
Description	Physical outputs
Entry Category	Mandatory
Access	Read/Write

PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Sub-Index	**2**
Description	Output Mask
Entry Category	Mandatory
Access	Read/Write
PDO Mapping	YES
Value Range	UNSIGNED32
Default Value	0

Object 60FFh: Target velocity

The target velocity is the input for the trajectory generator and the value is given in velocity units.

Object Description:

Index	**60FF$_h$**
Name	Target velocity
Object code	VAR
Data Type	INTEGER32
Category	Mandatory

Entry Description:

Access	Read/Write
PDO Mapping	YES
Value Range	INTEGER32
Default Value	0

Object 6502h: Supported Drive Modes

This object is organized bit-wise. The bits have the following meaning:

Bit Description

- 1 = profile position mode
- 3 = profile velocity mode
- 4 = profile torque mode
- 6= homing mode
- 7= interpolated position mode
- 8= cyclic synchronous position mode
- 10-15 = reserved
- 16-31 = manufacturer-specific

- Object Description:

Index	**6502**$_h$
Name	Supported Drive Modes
Object code	VAR
Data Type	Unsigned32
Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	Unsigned32
Default Value	0

Object 6504h: Drive Manufacturer

- Object Description:

Index	**6502**$_h$
Name	Drive Manufacturer
Object code	VAR
Data Type	VISIBILE STRING

Category	Optional

Entry Description:

Access	Read Only
PDO Mapping	NO
Value Range	VISIBILE STRING
Default Value	MOTOR POWER COMPANI

8. Configuer EtherCAT Duet AD with TwinCAT controller

This chapter explains how to configure the Beckhoff controller for communication and operation with the DUET AD model.

The application system consists of the following elements:

- DUET AD motor.
- Beckhoff controller with EtherCAT communication module, and TwinCAT software.

Notes: Beckhoff controller refers to TwinCAT NC PTP (point-to-point axis positioning software).

TwinCAT NC PTP includes axis positioning software (set value generation, position control), an integrated software PLC with NC interface, operating program for commissioning and an I/O connection to the axes through various fieldbuses. TwinCAT NC PTP replaces conventional positioning modules and NC controllers. The controllers that are simulated by the PC cyclically exchange data with drives and measuring systems via the fieldbus.

Beckhoff controllers are programmed in accordance with the IEC 61131-3 programming standard.

Before activating the **TwinCAT System Manager**, copy the correct ***.xml**

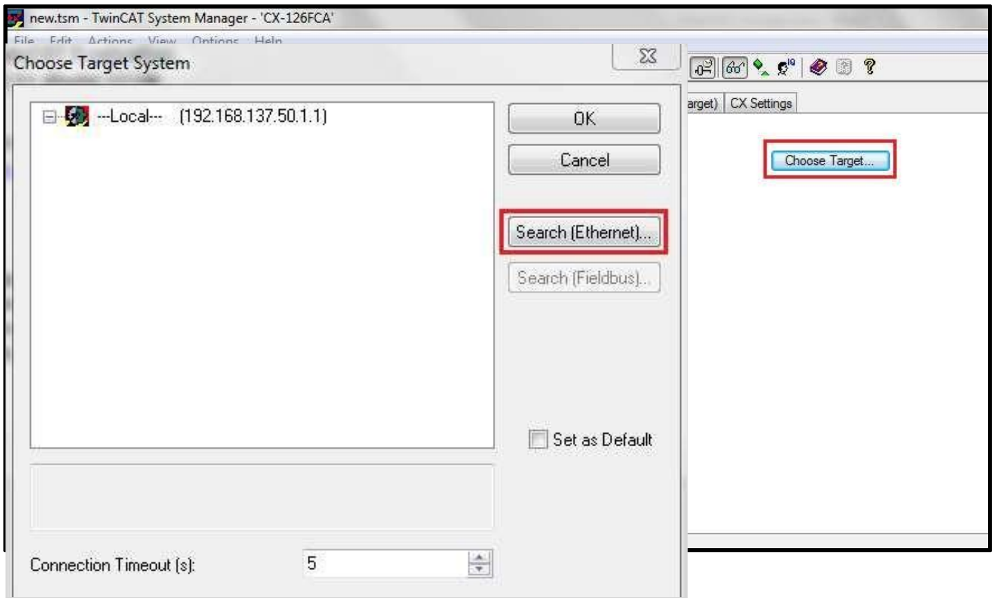
file taken from the site : https://www.motorpowerco.com/it/download-area/nav#201

Duet AD >ESI, into the PC folder: C:\TwinCAT\3.1\Config\Io\EtherCAT

Communication between Controller and PC

Using **TwinCAT** software, establish communication between the controller and the PC by performing the following steps.

- Activate **TwinCAT** software.
- In the navigation pane, select **SYSTEM – Configuration**. Then, in the **Version (Local) tab**, click **Choose Target.**



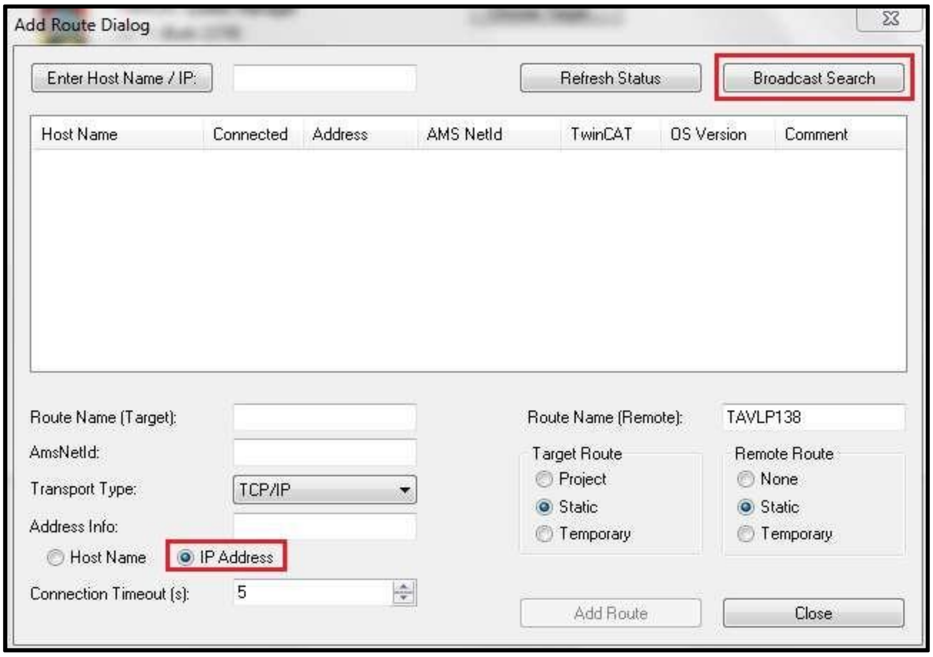
---Local--- (192.168.137.50.1.1)

Connection Timeout (s): 5

Figura 21

- Click **Search (Ethernet)** to search for the controller in the network.
- Enable the option **IP Address**, and click **Broadcast Search.**

Wait for the controller name (in the format CX-xxx) to appear.



Add Route Dialog

Enter Host Name / IP:

Refresh Status

Broadcast Search

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment

Route Name (Target):

AmsNetId:

Transport Type: TCP/IP

Address Info:

○ Host Name ● IP Address

Connection Timeout (s): 5

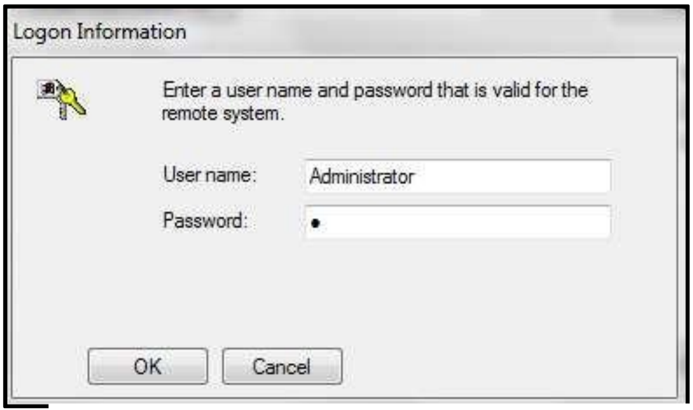
Route Name (Remote): TAVLP138

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

Figura 22

- After the controller appears, the option **Add Route** is displayed. Click Add Route
- In the Logon dialog box, enter the following: User Name: **Administrator** Password: 1 Click **OK.**



Logon Information

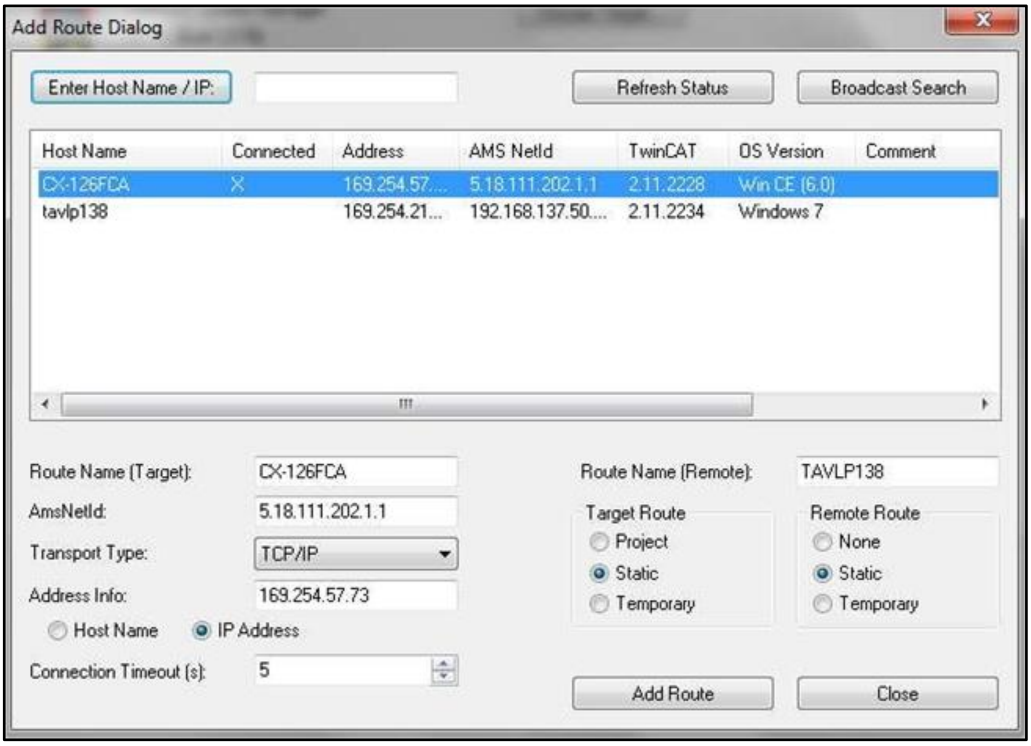
Enter a user name and password that is valid for the remote system.

User name: Administrator

Password: •

Figura 23

- In the Add Route dialog box, be sure an **X** appears next to the controller name. This means the controller is properly connected to the PC.
  Close this dialog box.



Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX-126FCA	X	169.254.57....	5.18.111.202.1.1	2.11.2228	Win CE (6.0)	
tavlp138		169.254.21...	192.168.137.50....	2.11.2234	Windows 7	

Route Name (Target): CX-126FCA

AmsNetId: 5.18.111.202.1.1

Transport Type: TCP/IP

Address Info: 169.254.57.73

○ Host Name ● IP Address

Connection Timeout (s): 5

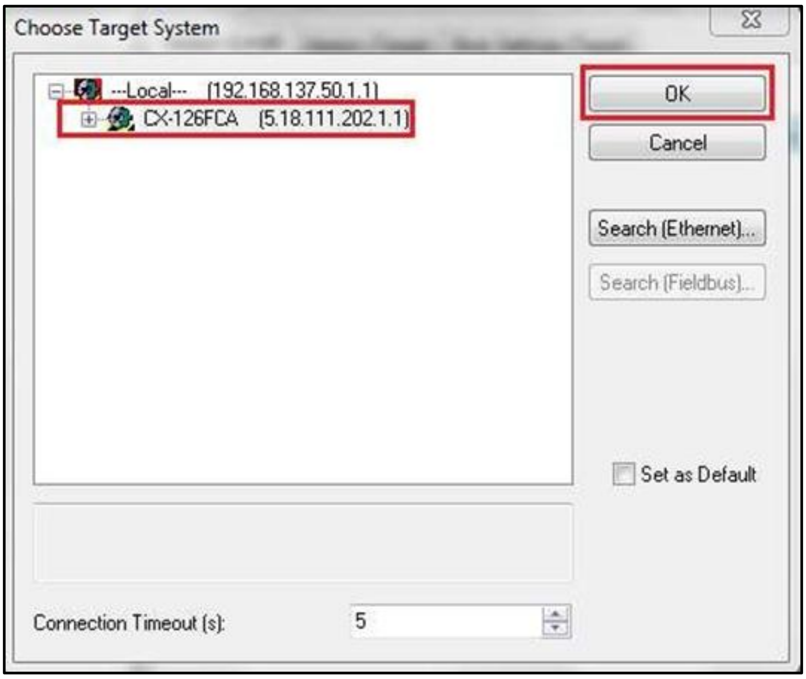
Route Name (Remote): TAVLP138

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

Figura 24

- In the Choose Target System dialog box, click on the controller, and click **OK**.



---Local--- (192.168.137.50.1.1)

CX-126FCA (5.18.111.202.1.1)

□ Set as Default

Connection Timeout (s): 5

Figura 25

- Open the TwinCAT System Manager, and make sure it is in **Config Mode**.

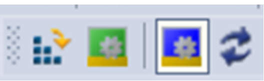


Figura 26

Communication between Controller and Drive

Using **TwinCAT** software, establish communication between the controller and the drive by performing the following steps.

- In the navigation pane, expand **I/O–Configuration**, and then right-click on **I/O Devices**.
- Select **Scan Devices.**

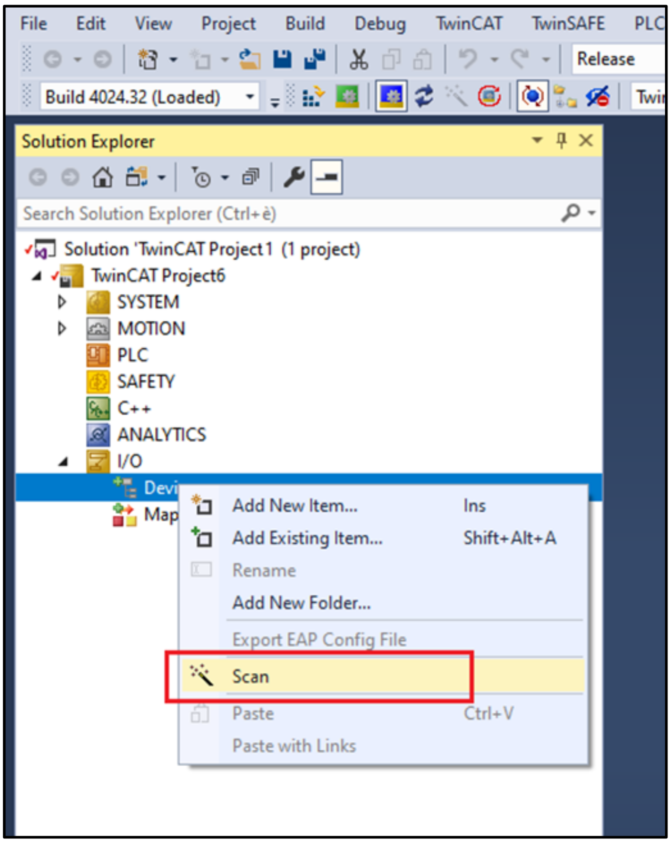


Figura 27

At the prompt, click **OK**.

- After the scanning, detected devices are displayed. DUET AD is identified as **Device 2 (EtherCAT).**

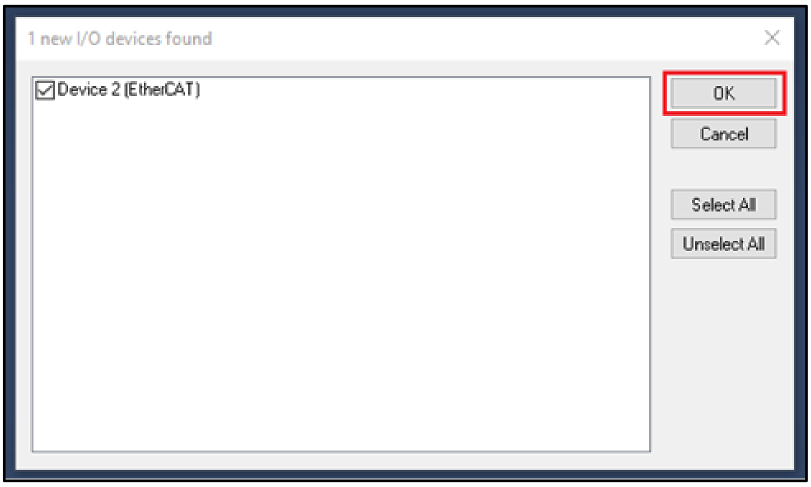


Figura 28

- Enable the option **Device 2 (EtherCAT**), and click **OK**
- At the prompt to scan for boxes (slaves), click **Yes**.
-

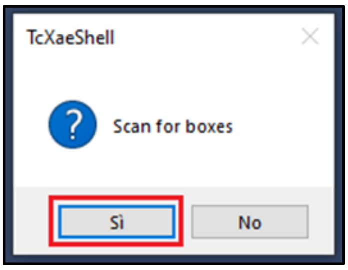


Figura 29

- At the prompt to append the linked axis to the NC configuration, click **Yes**.

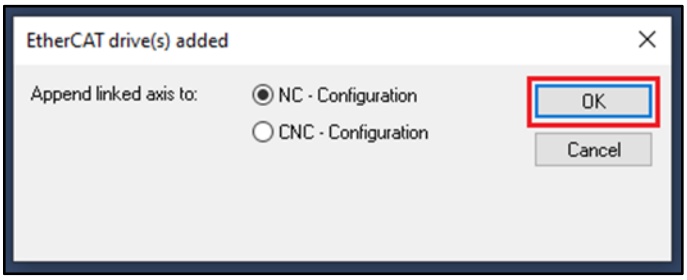
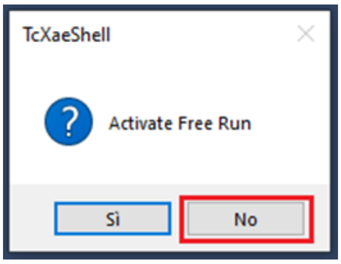


Figura 30

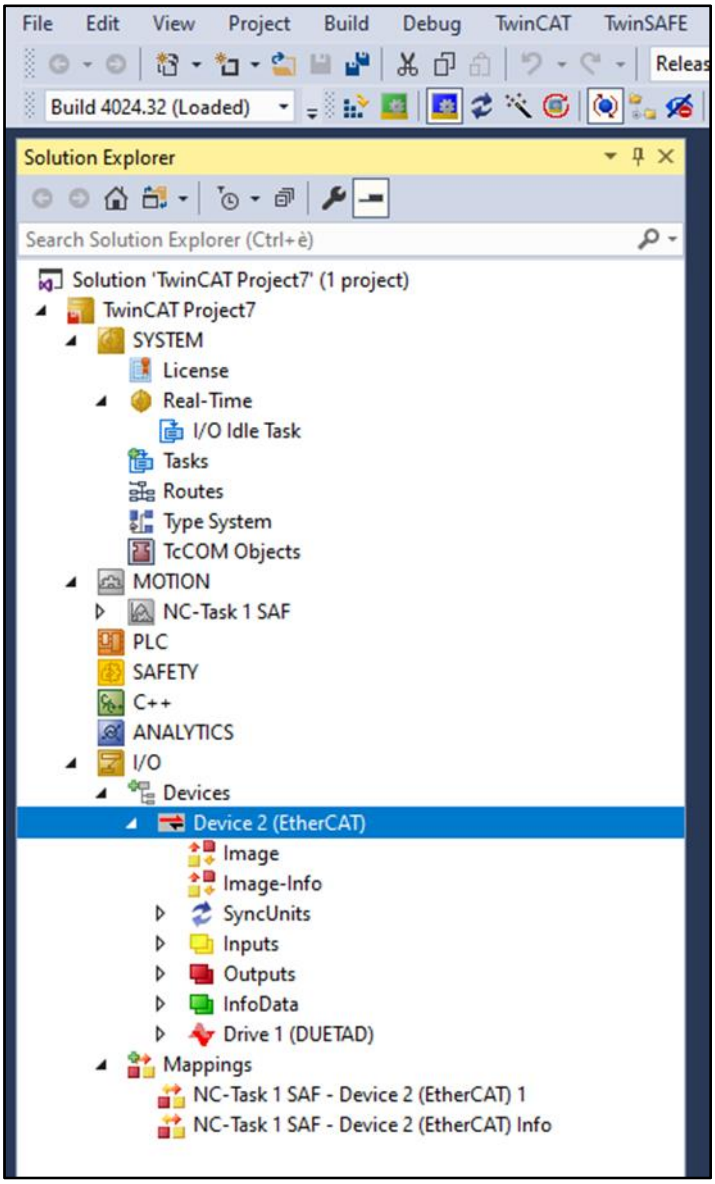
- At the prompt to activate FreeRun, click **No**.



TcXaeShell

Activate Free Run

Figura 31



```
Solution 'TwinCAT Project7' (1 project)
  TwinCAT Project7
    SYSTEM
      License
      Real-Time
        I/O Idle Task
      Tasks
      Routes
      Type System
      TcCOM Objects
    MOTION
      NC-Task 1 SAF
    PLC
    SAFETY
    C++
    ANALYTICS
    I/O
      Devices
        Device 2 (EtherCAT)
          Image
          Image-Info
          SyncUnits
          Inputs
          Outputs
          InfoData
          Drive 1 (DUETAD)
      Mappings
        NC-Task 1 SAF - Device 2 (EtherCAT) 1
        NC-Task 1 SAF - Device 2 (EtherCAT) Info
```

Figura 32

- At the end of this procedure, Device 2 (EtherCAT) is displayed in the navigation pane, with all components (TPDO and RPDO) listed and automatically linked to NC–Configuration>Axis 1.

Generating Motion

Setup for Motion

- Open the TwinCAT System Manager, and make sure it is in **Config Mode**.

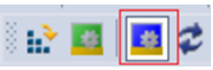
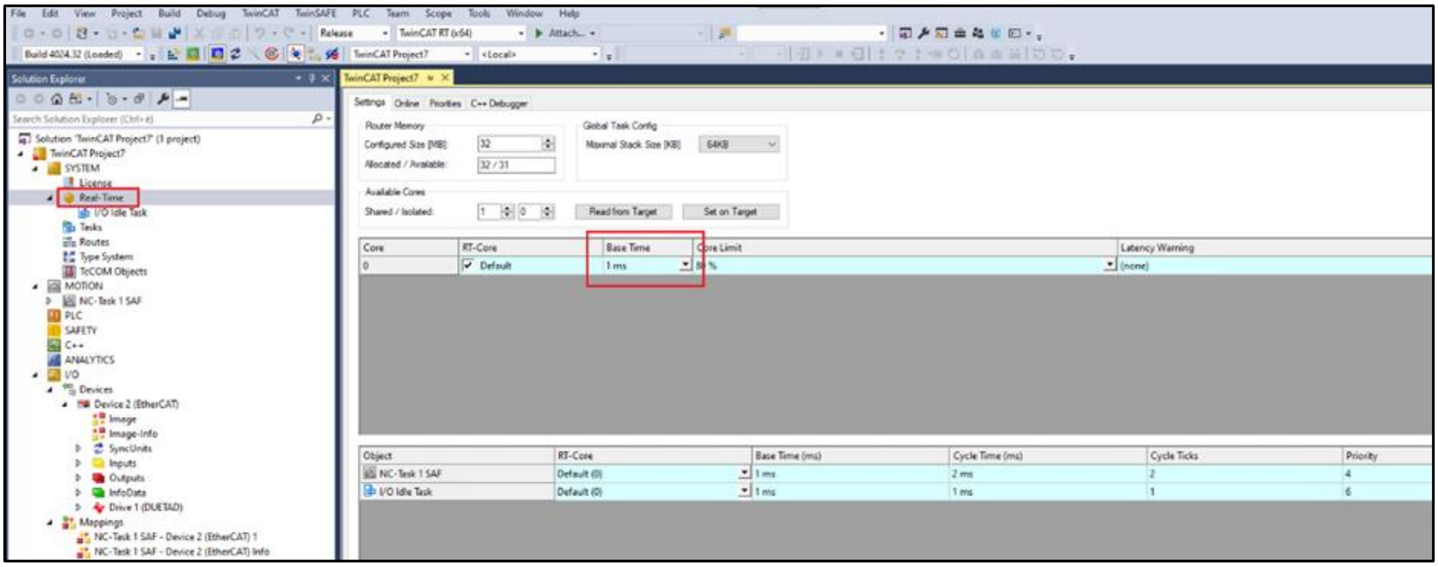


Figura 33

- In the navigation panel, expand **SYSTEM–Configuration**, and select **Real Time Settings**.
- In the **Settings** tab, select Base Time = 1 ms.



Router Memory

Configured Size [MB]: 32

Allocated / Available: 32 / 31

Global Task Config

Maximal Stack Size [KB]: 64KB

Available Cores

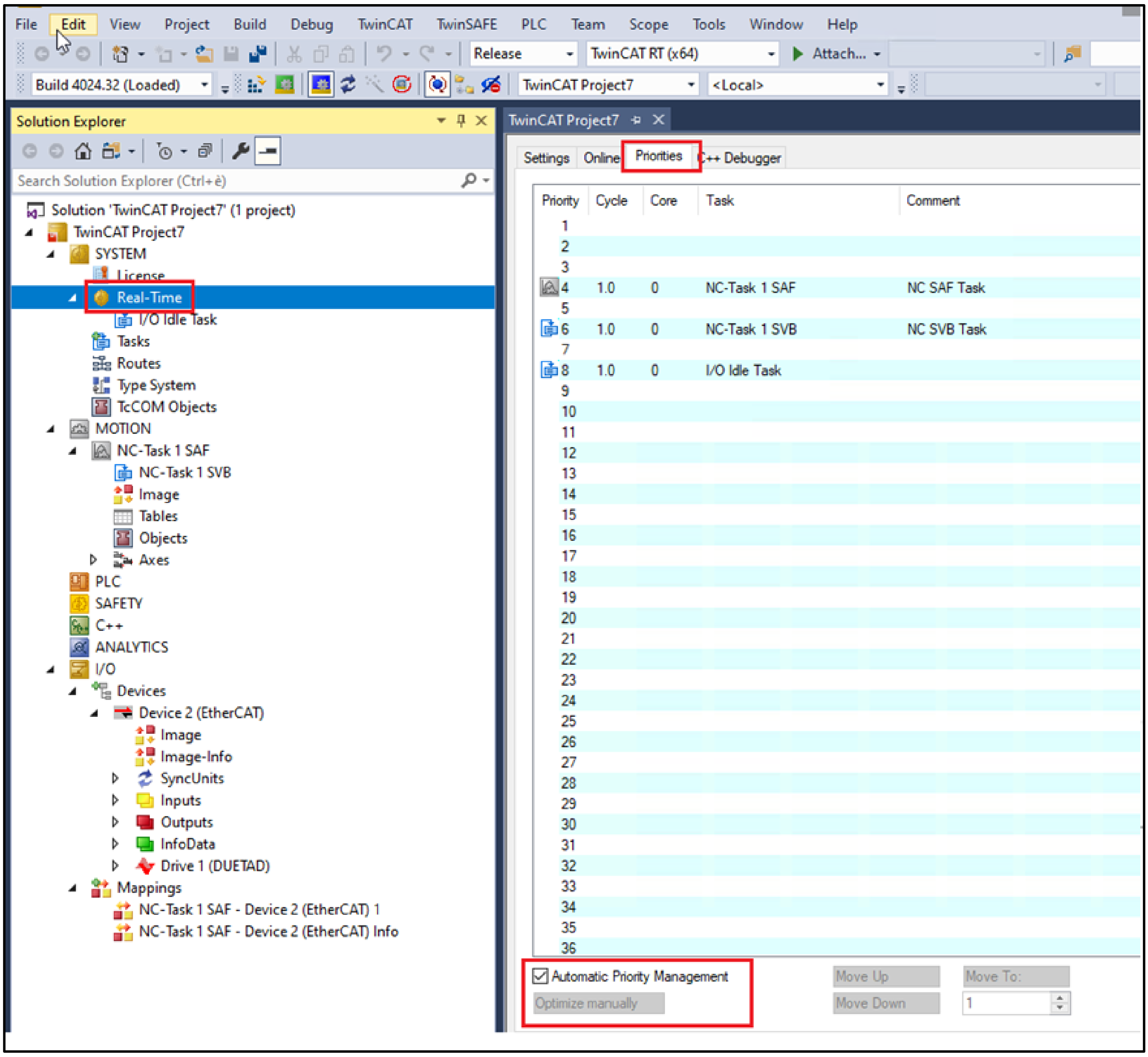
Shared / Isolated: 1 0

Core	RT-Core	Base Time	Core Limit	Latency Warning
0	☑ Default	1 ms	80 %	(none)

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks	Priority
NC-Task 1 SAF	Default (0)	1 ms	2 ms	2	4
I/O Idle Task	Default (0)	1 ms	1 ms	1	6

Figura 34

- In the Priorities tab, enable Automatic Priority Management.



Priority	Cycle	Core	Task	Comment
1				
2				
3				
4	1.0	0	NC-Task 1 SAF	NC SAF Task
5				
6	1.0	0	NC-Task 1 SVB	NC SVB Task
7				
8	1.0	0	I/O Idle Task	
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				

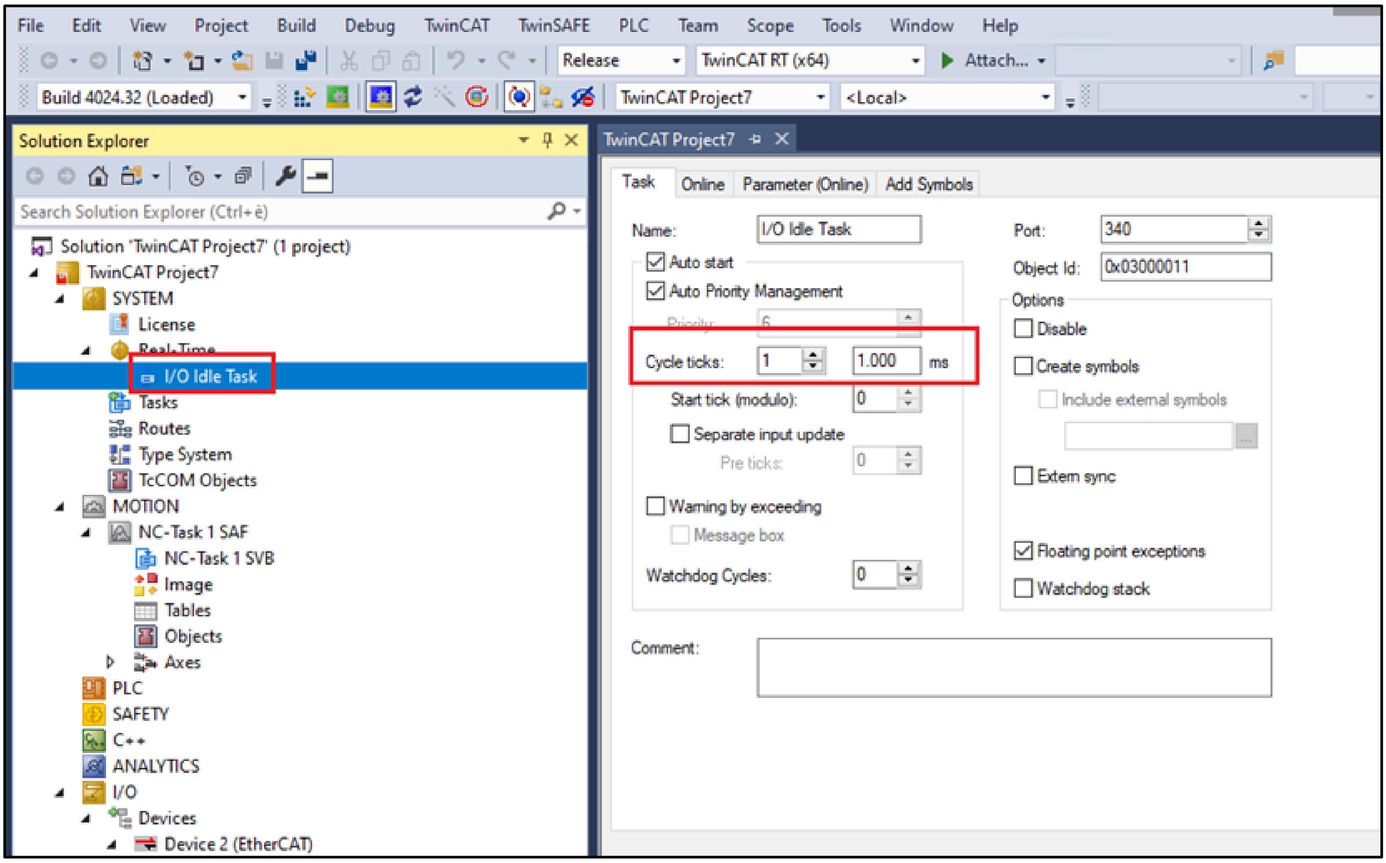
☑ Automatic Priority Management

Optimize manually

Move To: 1

Figura 35

- Expand **SYSTEM–Configuration**, and select Real Time Settings > **I/O Idle Task.**



Name: I/O Idle Task

☑ Auto start

☑ Auto Priority Management

Priority: 6

Cycle ticks: 1 1.000 ms

Start tick (modulo): 0

□ Separate input update

Pre ticks: 0

□ Warning by exceeding

□ Message box

Watchdog Cycles: 0

Port: 340

Object Id: 0x03000011

Options

□ Disable

□ Create symbols

□ Include external symbols

□ Extern sync

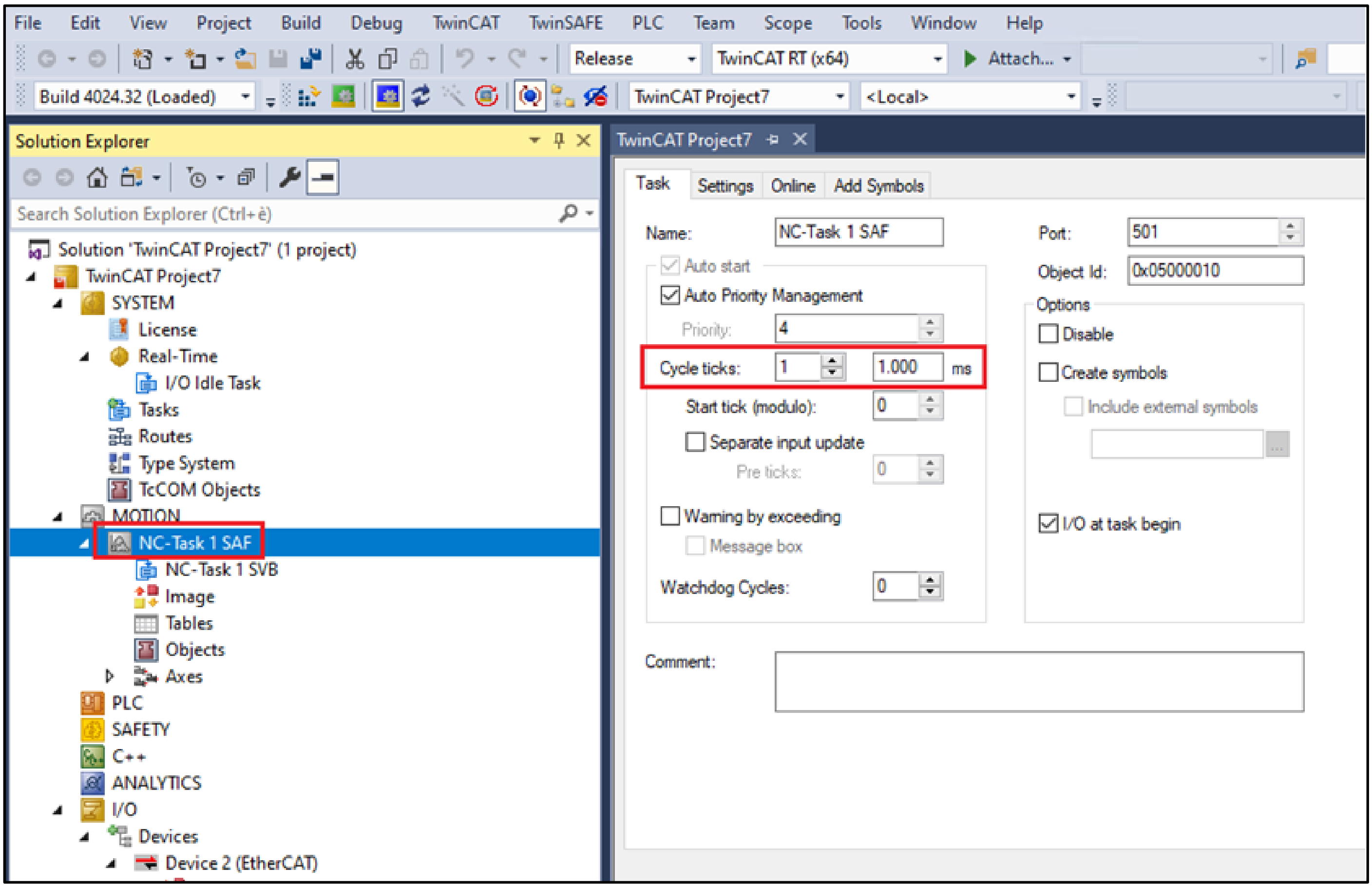
☑ Floating point exceptions

□ Watchdog stack

Comment:

Figura 36

- In the navigation pane, expand **NC–Configuration**, and select **NC-Task1SAF.**
- In the **Task** tab, select the desired cycle time: for example Cycle ticks = 1 ms. **NOTE**: minimum cycle time for DUET AD is 1ms.



Name: NC-Task 1 SAF

☑ Auto start

☑ Auto Priority Management

Priority: 4

Cycle ticks: 1 1.000 ms

Start tick (modulo): 0

□ Separate input update

Pre ticks: 0

□ Warning by exceeding

□ Message box

Watchdog Cycles: 0

Port: 501

Object Id: 0x05000010

Options

□ Disable

□ Create symbols

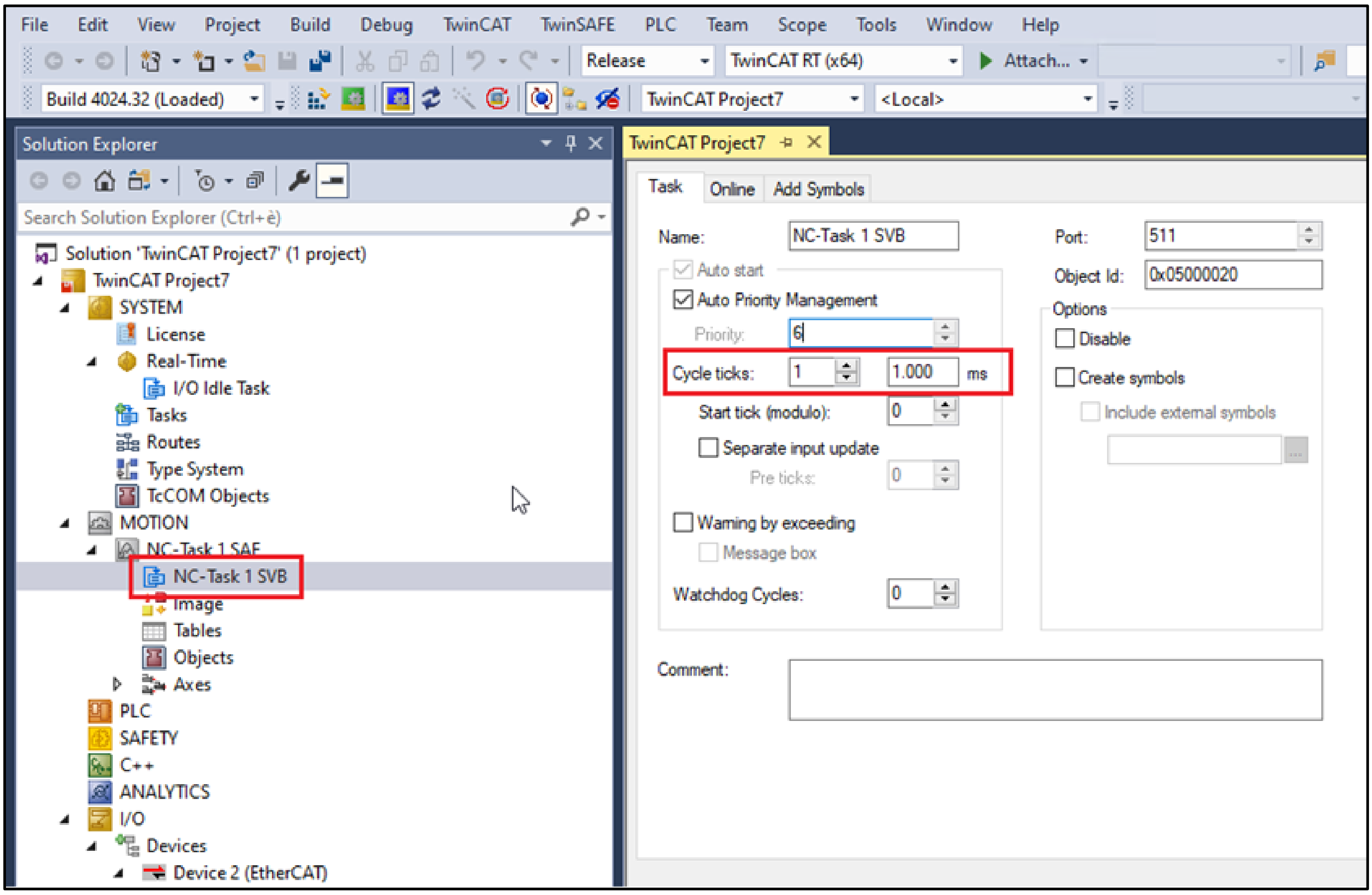
□ Include external symbols

☑ I/O at task begin

Comment:

Figura 37

- Expand NCT-Task1SAF, and select NC-Task1SVB.
- In the **Task** tab, select Cycle ticks = 1 ms.
- Be sure the priority of NC-Task1 SVB has a higher value than the priority of NC-Task1 SAF.



Solution 'TwinCAT Project7' (1 project)
- TwinCAT Project7
  - SYSTEM
    - License
    - Real-Time
      - I/O Idle Task
    - Tasks
    - Routes
    - Type System
    - TcCOM Objects
  - MOTION
    - NC-Task 1 SAF
      - NC-Task 1 SVB
      - Image
      - Tables
      - Objects
      - Axes
  - PLC
  - SAFETY
  - C++
  - ANALYTICS
  - I/O
    - Devices
      - Device 2 (EtherCAT)

Name: NC-Task 1 SVB

☑ Auto start

☑ Auto Priority Management

Priority: 6

Cycle ticks: 1 1.000 ms

Start tick (modulo): 0

☐ Separate input update

Pre ticks: 0

☐ Warning by exceeding

☐ Message box

Watchdog Cycles: 0

Port: 511

Object Id: 0x05000020

Options

☐ Disable

☐ Create symbols

☐ Include external symbols

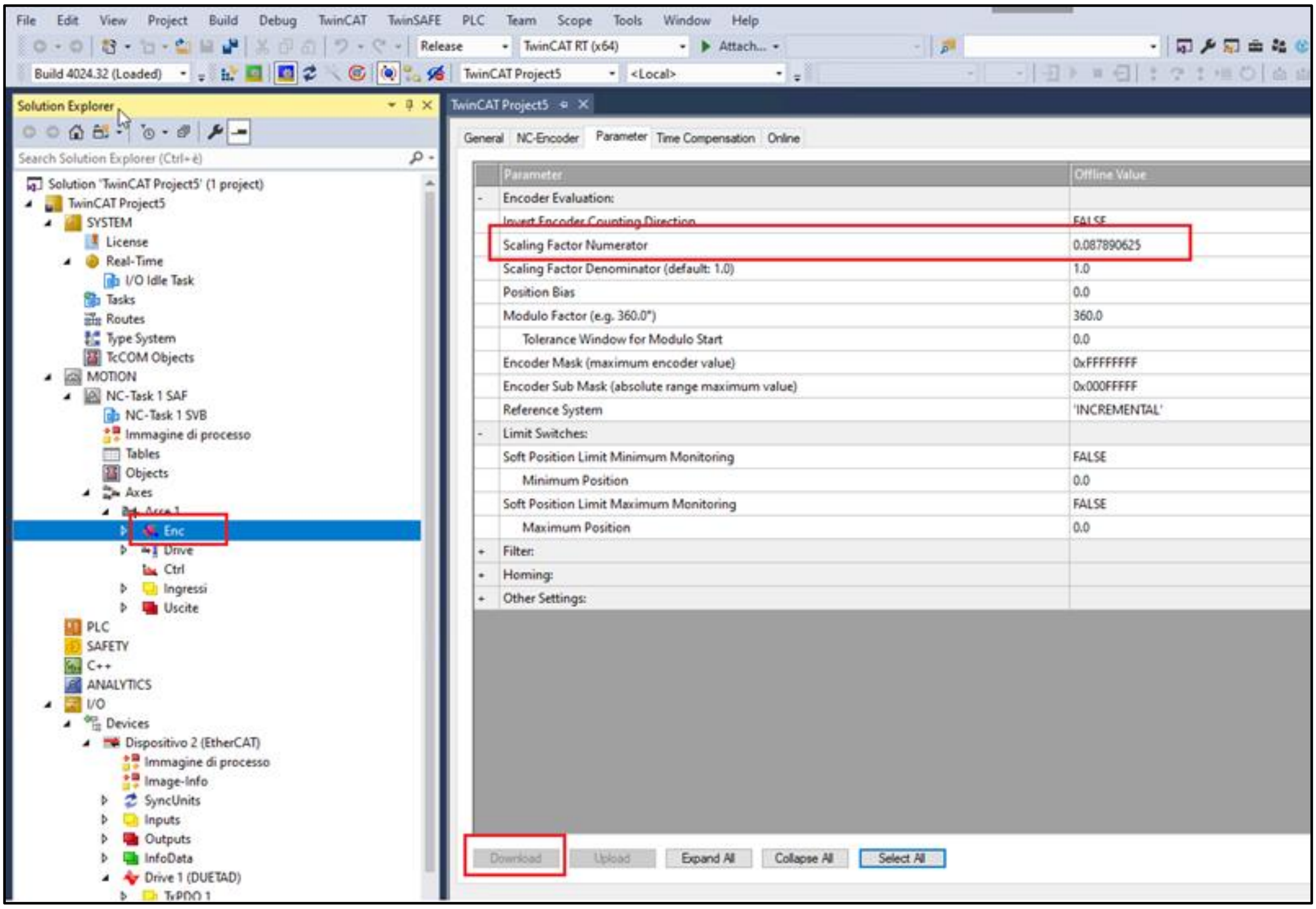
Comment:

Figura 38

- Expand **NC-Configuration** > Axes > Axis 1 > **Axis 1_Enc**. In the **Parameter** tab, do the following:
- Encoder Evaluation > Scaling Factor = 0.087890625

This value is given by : $\frac{360°}{4096\ count}$

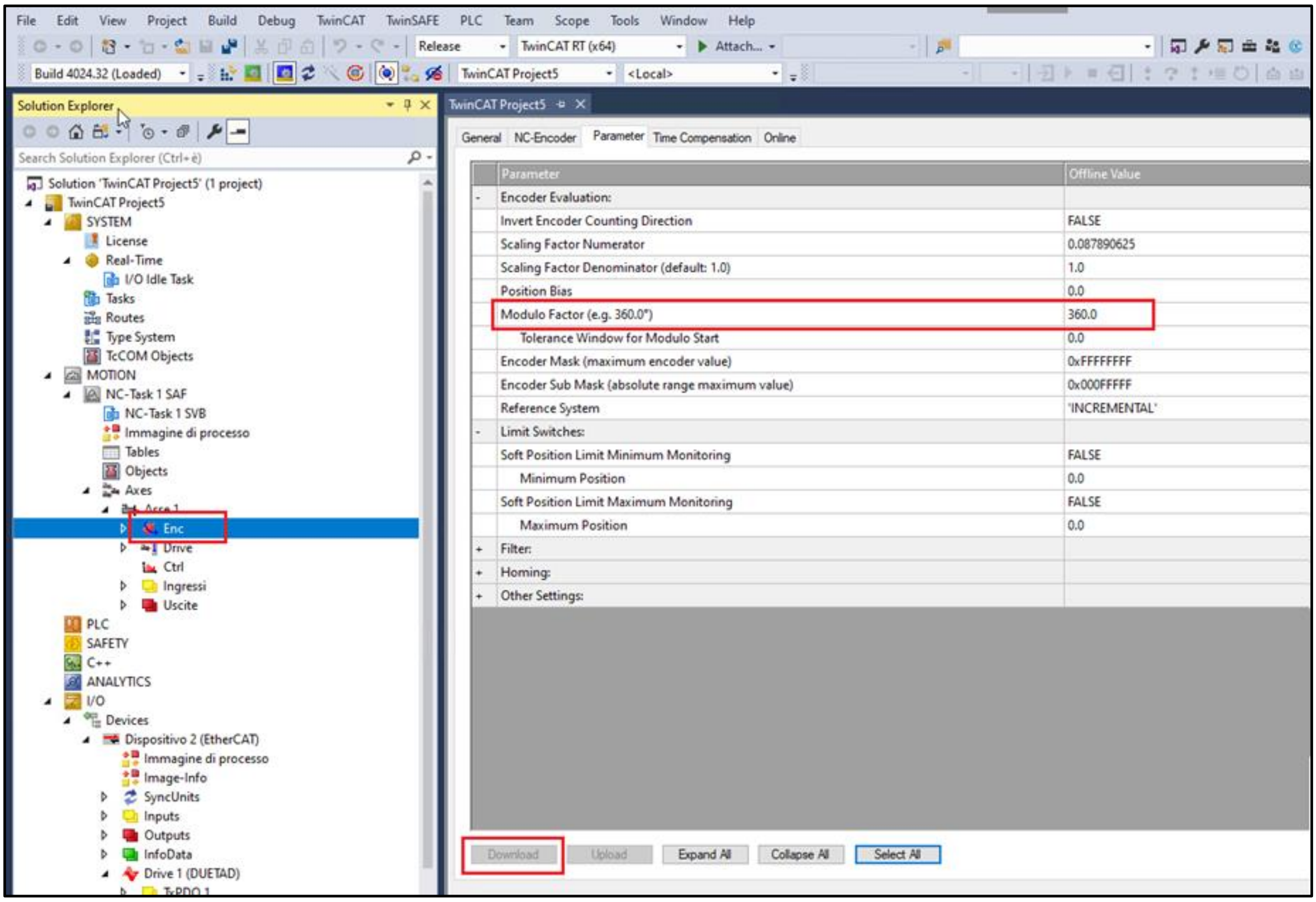
Click Download.



	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	0.087890625
	Scaling Factor Denominator (default: 1.0)	1.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0
	Encoder Mask (maximum encoder value)	0xFFFFFFFF
	Encoder Sub Mask (absolute range maximum value)	0x000FFFFF
	Reference System	'INCREMENTAL'
-	Limit Switches:	
	Soft Position Limit Minimum Monitoring	FALSE
	Minimum Position	0.0
	Soft Position Limit Maximum Monitoring	FALSE
	Maximum Position	0.0
+	Filter:	
+	Homing:	
+	Other Settings:	

Figura 39

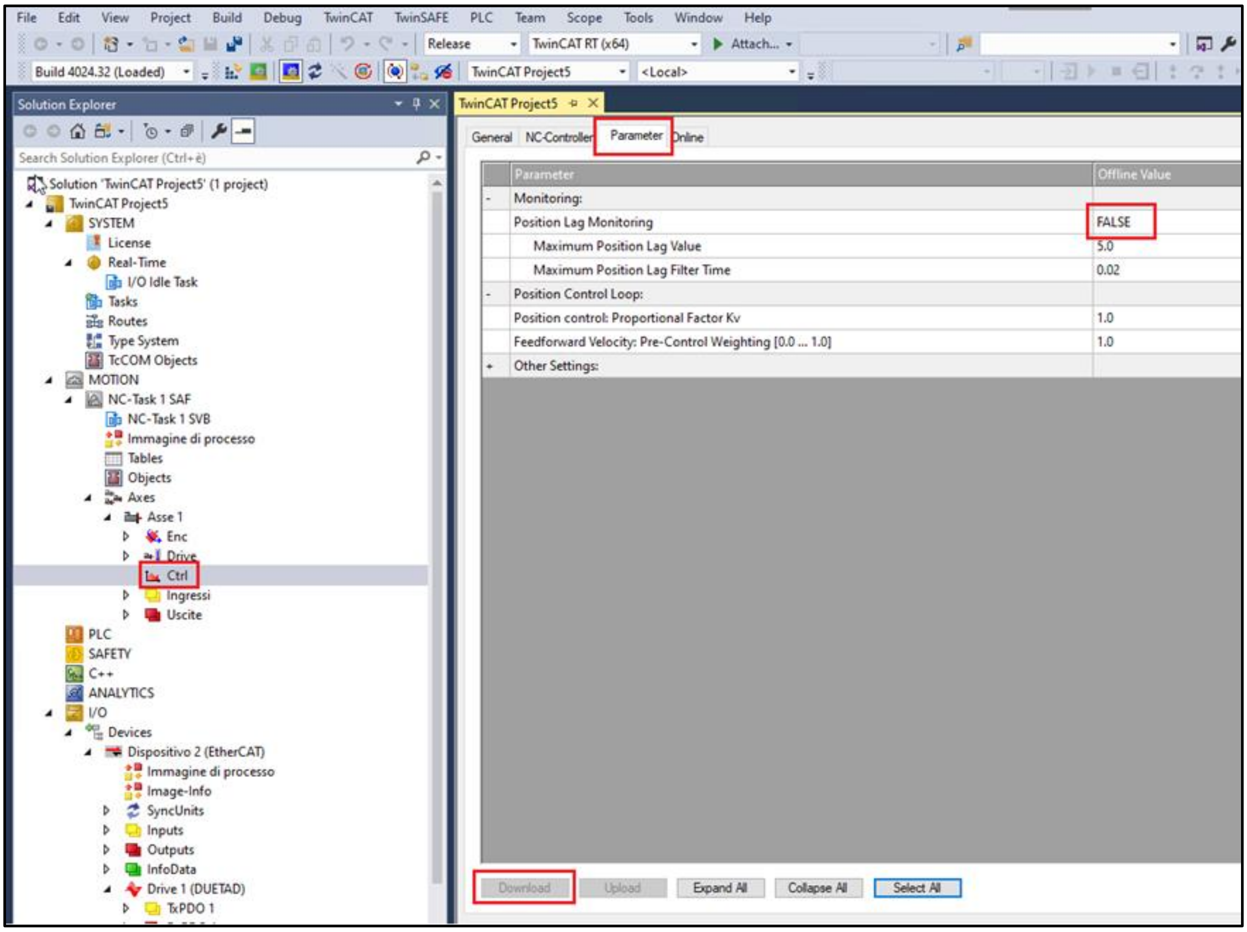
- Encoder Evaluation > Modulo Factor = 360.
  Click Download.



	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	0.087890625
	Scaling Factor Denominator (default: 1.0)	1.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0
	Encoder Mask (maximum encoder value)	0xFFFFFFFF
	Encoder Sub Mask (absolute range maximum value)	0x000FFFFF
	Reference System	'INCREMENTAL'
-	Limit Switches:	
	Soft Position Limit Minimum Monitoring	FALSE
	Minimum Position	0.0
	Soft Position Limit Maximum Monitoring	FALSE
	Maximum Position	0.0
+	Filter:	
+	Homing:	
+	Other Settings:	

Figura 40

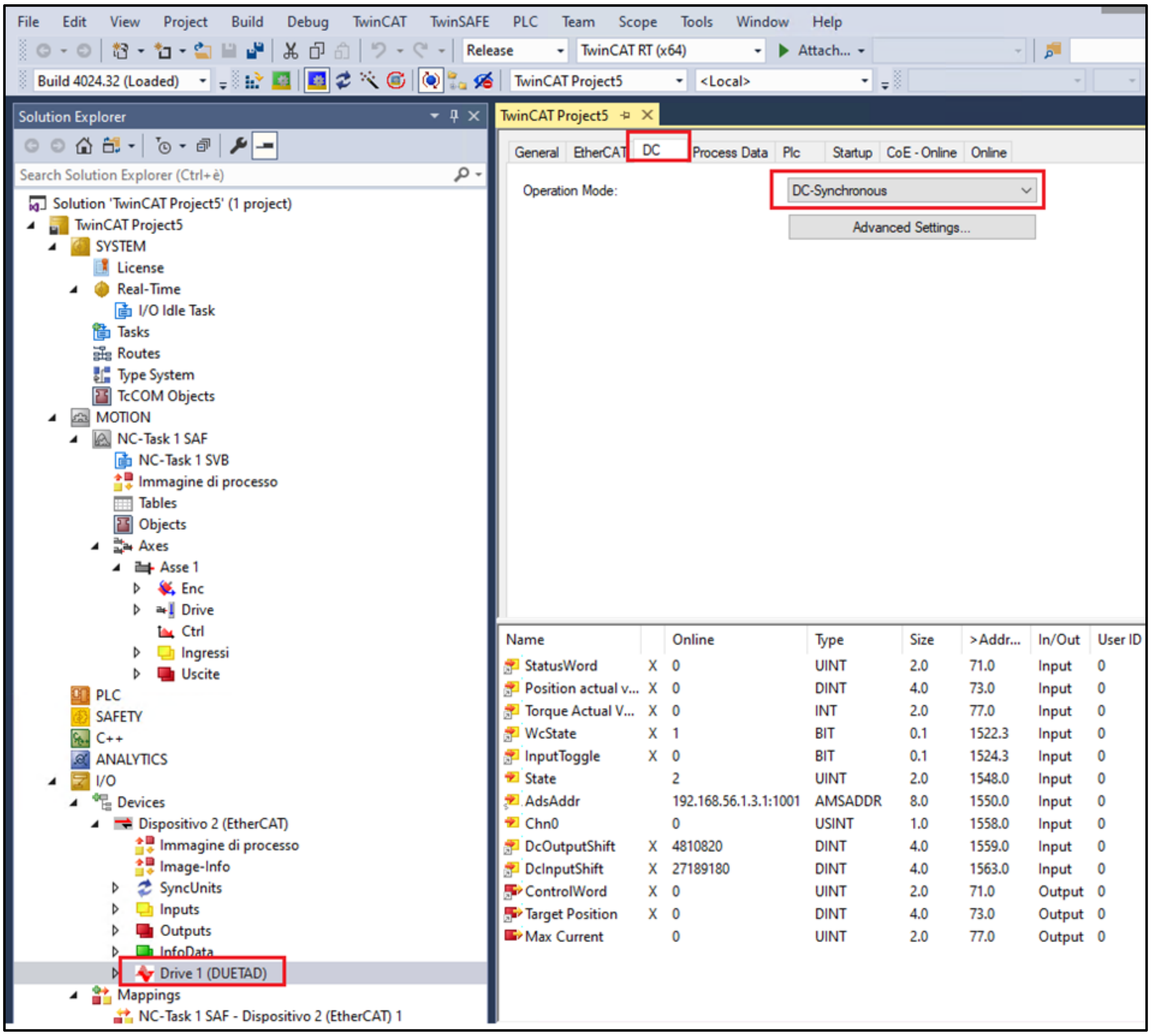
- Expand **NC-Configuration** > Axes > Axis 1 > **Axis 1_Ctrl**. In the **Parameter** tab, do the following:
- **Monitoring** > Position Lag Monitoring = FALSE



	Parameter	Offline Value
-	Monitoring:	
	Position Lag Monitoring	FALSE
	Maximum Position Lag Value	5.0
	Maximum Position Lag Filter Time	0.02
-	Position Control Loop:	
	Position control: Proportional Factor Kv	1.0
	Feedforward Velocity: Pre-Control Weighting [0.0 ... 1.0]	1.0
+	Other Settings:	

Figura 41

- Expand **IO-Configuration** > **I/O Devices** > Device 2 (EtherCAT), and select the drive indicated by the red icon (Drive 1 (DUETAD)).
- In the **DC** tab, select Operation Mode = **DC-Synchronous**



Operation Mode: DC-Synchronous

Name		Online	Type	Size	>Addr...	In/Out	User ID
StatusWord	X	0	UINT	2.0	71.0	Input	0
Position actual v...	X	0	DINT	4.0	73.0	Input	0
Torque Actual V...	X	0	INT	2.0	77.0	Input	0
WcState	X	1	BIT	0.1	1522.3	Input	0
InputToggle	X	0	BIT	0.1	1524.3	Input	0
State		2	UINT	2.0	1548.0	Input	0
AdsAddr		192.168.56.1.3.1:1001	AMSADDR	8.0	1550.0	Input	0
Chn0		0	USINT	1.0	1558.0	Input	0
DcOutputShift	X	4810820	DINT	4.0	1559.0	Input	0
DcInputShift	X	27189180	DINT	4.0	1563.0	Input	0
ControlWord	X	0	UINT	2.0	71.0	Output	0
Target Position	X	0	DINT	4.0	73.0	Output	0
Max Current		0	UINT	2.0	77.0	Output	0

Figura 42

- Press the **Run Mode** button in the toolbar.

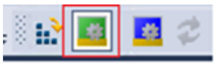


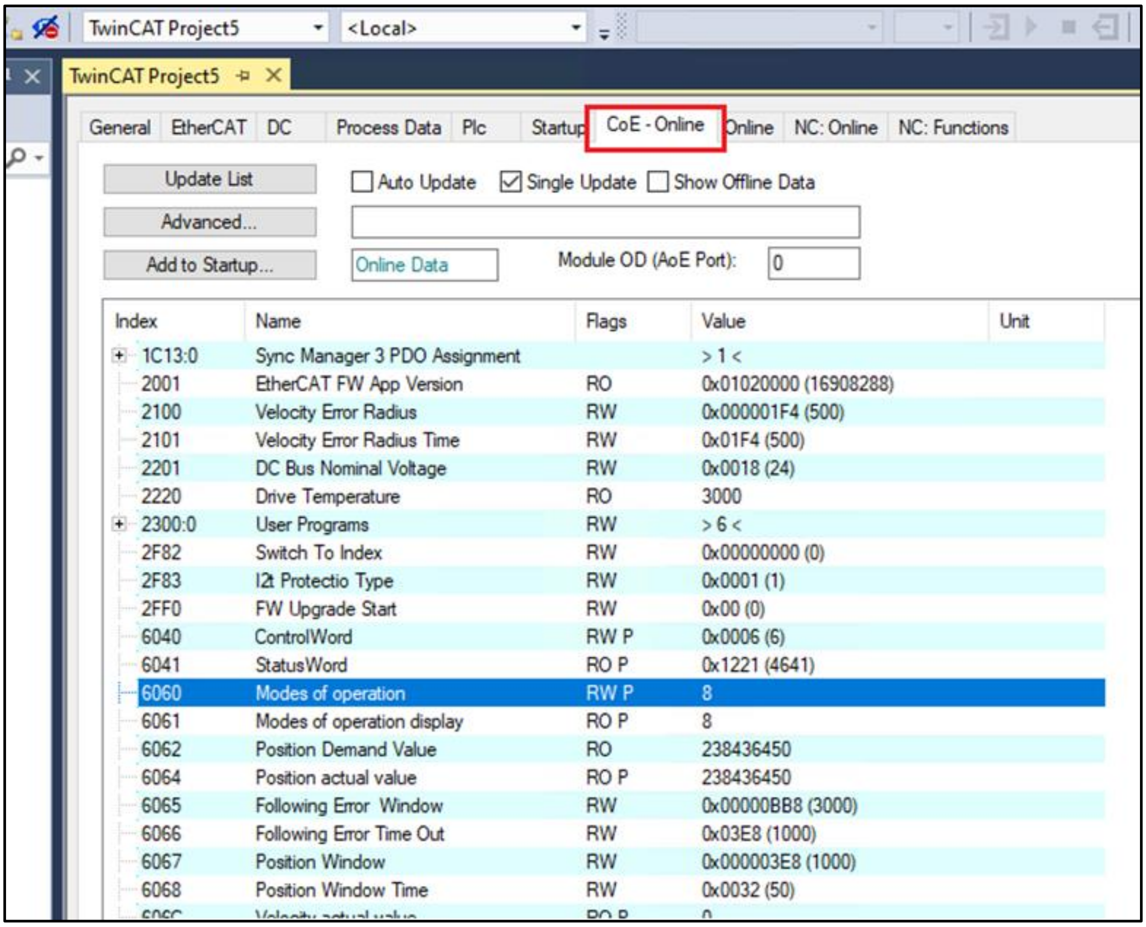
Figura 43

Additional tabs are now available.

- Go to the **CoE Online** tab.The CoE Online tab shows only the SDO objects (DUETAD EtherCAT parameters) that the drive manages.

Be sure the values of the objects 6060h and 60C2h are as follows:

- Object 6060h = 8
  The drive is set to Cyclic Synchronous Position mode, OPMODE 8, through protocol object 6060



□ Auto Update ☑ Single Update □ Show Offline Data

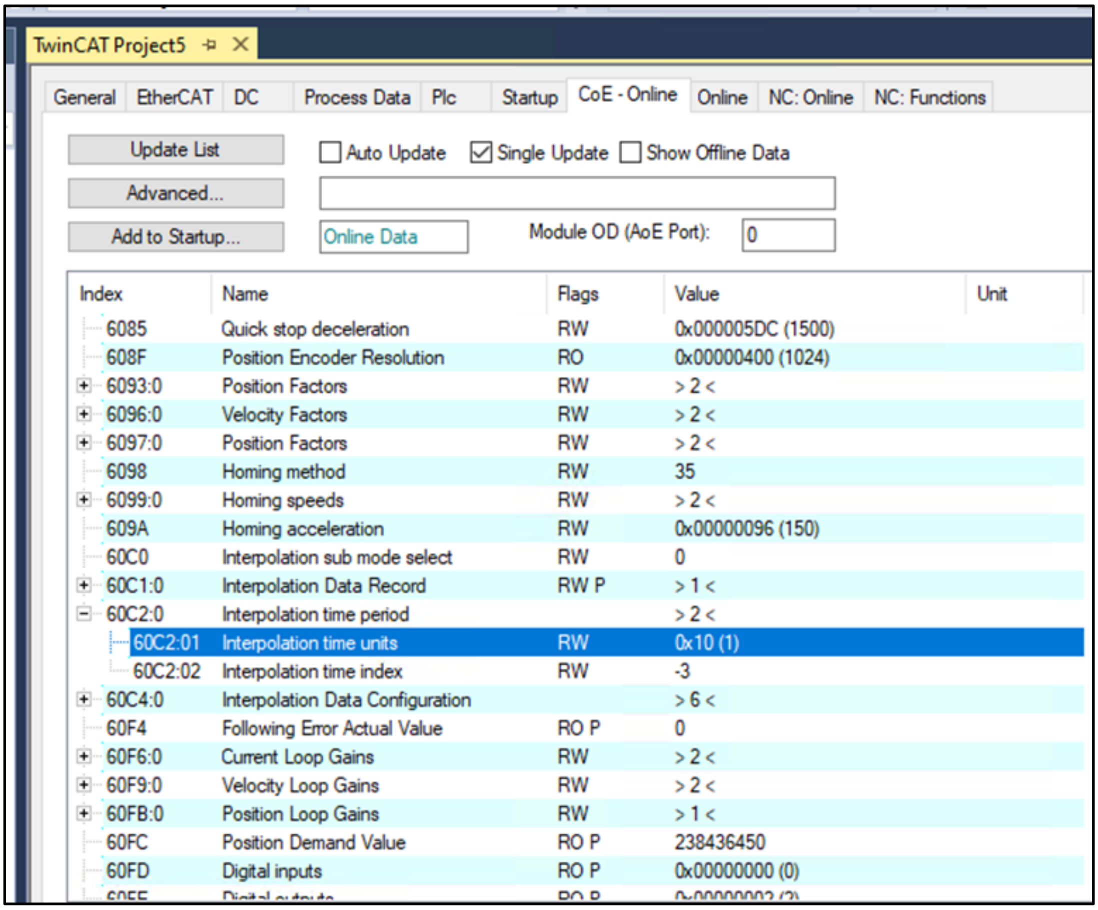
Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 1C13:0	Sync Manager 3 PDO Assignment		> 1 <	
2001	EtherCAT FW App Version	RO	0x01020000 (16908288)	
2100	Velocity Error Radius	RW	0x000001F4 (500)	
2101	Velocity Error Radius Time	RW	0x01F4 (500)	
2201	DC Bus Nominal Voltage	RW	0x0018 (24)	
2220	Drive Temperature	RO	3000	
+ 2300:0	User Programs	RW	> 6 <	
2F82	Switch To Index	RW	0x00000000 (0)	
2F83	I2t Protectio Type	RW	0x0001 (1)	
2FF0	FW Upgrade Start	RW	0x00 (0)	
6040	ControlWord	RW P	0x0006 (6)	
6041	StatusWord	RO P	0x1221 (4641)	
6060	Modes of operation	RW P	8	
6061	Modes of operation display	RO P	8	
6062	Position Demand Value	RO	238436450	
6064	Position actual value	RO P	238436450	
6065	Following Error Window	RW	0x00000BB8 (3000)	
6066	Following Error Time Out	RW	0x03E8 (1000)	
6067	Position Window	RW	0x000003E8 (1000)	
6068	Position Window Time	RW	0x0032 (50)	
606C	Velocity actual value	RO P	0	

Figura 44

- Object 60C2h:
  - Sub-index 01 (60C2:01) = 1
  - Sub-index 02 (60C2:02) = -3

The interpolation time for the Cyclic Synchronous operation modes is set through object 60C2h (sub-index 01 and sub-index 02).



TwinCAT Project5

General | EtherCAT | DC | Process Data | Plc | Startup | CoE - Online | Online | NC: Online | NC: Functions

Update List | □ Auto Update | ☑ Single Update | □ Show Offline Data

Advanced...

Add to Startup... | Online Data | Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
6085	Quick stop deceleration	RW	0x000005DC (1500)	
608F	Position Encoder Resolution	RO	0x00000400 (1024)	
+ 6093:0	Position Factors	RW	> 2 <	
+ 6096:0	Velocity Factors	RW	> 2 <	
+ 6097:0	Position Factors	RW	> 2 <	
6098	Homing method	RW	35	
+ 6099:0	Homing speeds	RW	> 2 <	
609A	Homing acceleration	RW	0x00000096 (150)	
60C0	Interpolation sub mode select	RW	0	
+ 60C1:0	Interpolation Data Record	RW P	> 1 <	
− 60C2:0	Interpolation time period		> 2 <	
60C2:01	Interpolation time units	RW	0x10 (1)	
60C2:02	Interpolation time index	RW	-3	
+ 60C4:0	Interpolation Data Configuration		> 6 <	
60F4	Following Error Actual Value	RO P	0	
+ 60F6:0	Current Loop Gains	RW	> 2 <	
+ 60F9:0	Velocity Loop Gains	RW	> 2 <	
+ 60FB:0	Position Loop Gains	RW	> 1 <	
60FC	Position Demand Value	RO P	238436450	
60FD	Digital inputs	RO P	0x00000000 (0)	
60FE	Digital outputs	RO P	0x00000002 (2)	

Figura 45

Note: The interpolation time must be configured with the same value of cycle ticks as configured in I/O Idle Task, in NC-Task 1 SAF, and in NC-Task 1 SVB.

- Now activate **Run Mode** by pressing the following two buttons in the toolbar:
  Generate Mappings
  Check Configuration

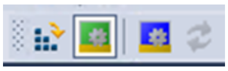
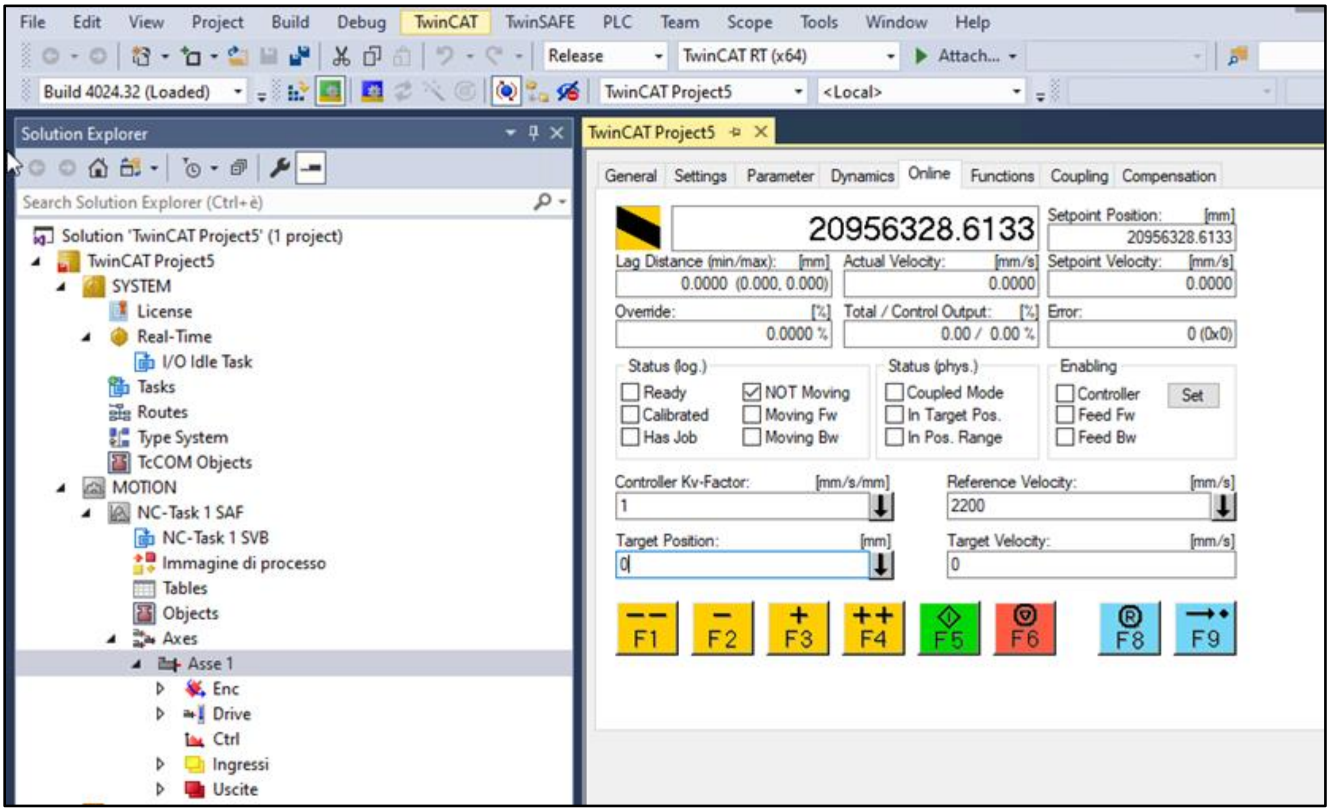


Figura 46

In Run mode, motion can be generated. The NC PTP communicates with the drive and receives all the values of the variables contained in each of the PDO objects (which were automatically mapped by the controller).



20956328.6133

Setpoint Position: [mm] 20956328.6133

Lag Distance (min/max): [mm] 0.0000 (0.000, 0.000)

Actual Velocity: [mm/s] 0.0000

Setpoint Velocity: [mm/s] 0.0000

Override: [%] 0.0000 %

Total / Control Output: [%] 0.00 / 0.00 %

Error: 0 (0x0)

Status (log.)

☐ Ready
☐ Calibrated
☐ Has Job
☑ NOT Moving
☐ Moving Fw
☐ Moving Bw

Status (phys.)

☐ Coupled Mode
☐ In Target Pos.
☐ In Pos. Range

Enabling

☐ Controller
☐ Feed Fw
☐ Feed Bw

Controller Kv-Factor: [mm/s/mm] 1

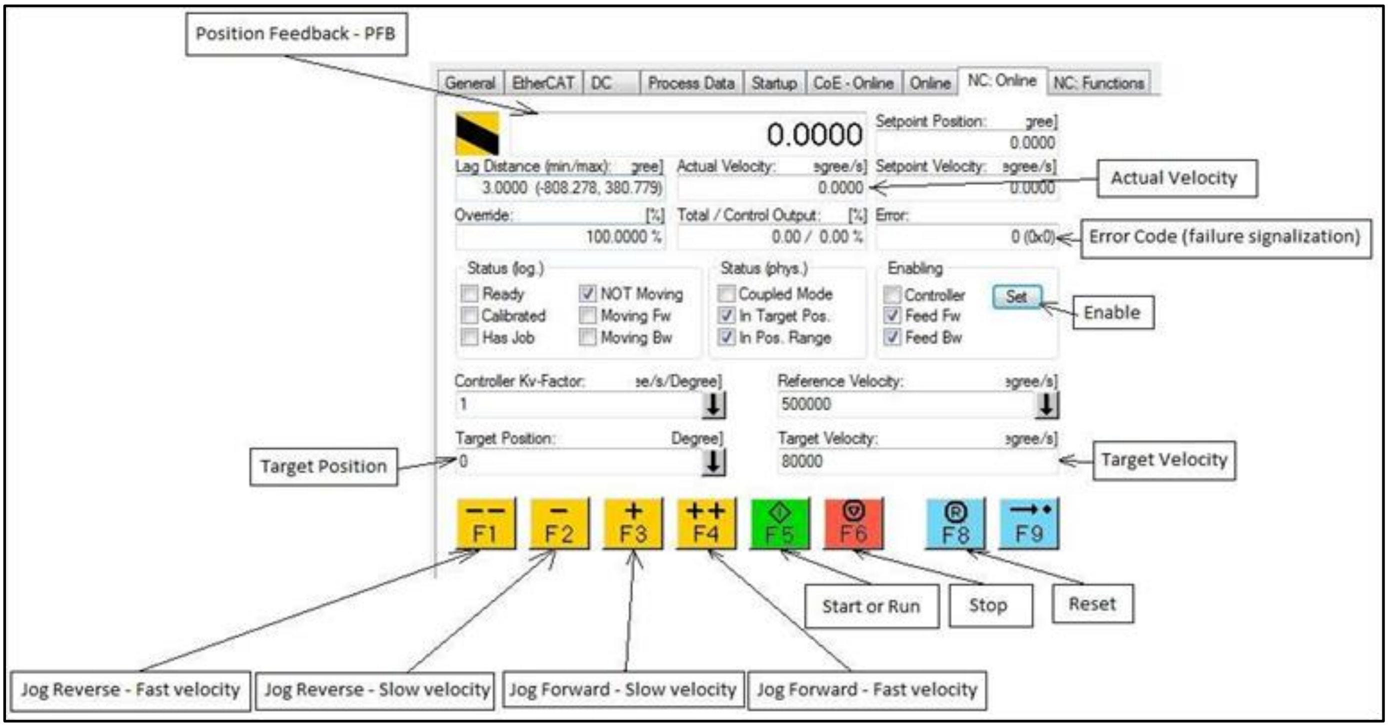
Reference Velocity: [mm/s] 2200

Target Position: [mm] 0

Target Velocity: [mm/s] 0

Figura 47

- In the **NC-Online** tab, test communication with the drive:Take hold of the motor shaft, and turn it manually; check whether the position feedback value changes.
  Refer to the following figure, which shows the various functions.



0.0000
Setpoint Position: gree] 0.0000
Lag Distance (min/max): gree] 3.0000 (-808.278, 380.779)
Actual Velocity: egree/s] 0.0000
Setpoint Velocity: egree/s] 0.0000
Override: [%] 100.0000 %
Total / Control Output: [%] 0.00 / 0.00 %
Error: 0 (0x0)

Status (log.)
□ Ready
□ Calibrated
□ Has Job
☑ NOT Moving
□ Moving Fw
□ Moving Bw

Status (phys.)
□ Coupled Mode
☑ In Target Pos.
☑ In Pos. Range

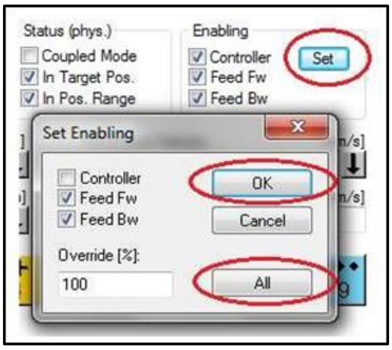
Enabling
□ Controller
☑ Feed Fw
☑ Feed Bw

Controller Kv-Factor: se/s/Degree] 1
Reference Velocity: egree/s] 500000
Target Position: Degree] 0
Target Velocity: egree/s] 80000

Figura 48

Generating Motion in Cyclic Synchronous Position Mode

- Enable the drive:
  - NC-Online Screen > Enabling > Set
  - Enable the options: **Controller**, **Feed Fw** and **Feed Bw**, or select **All**
  - OK



Status (phys.)
□ Coupled Mode
☑ In Target Pos.
☑ In Pos. Range

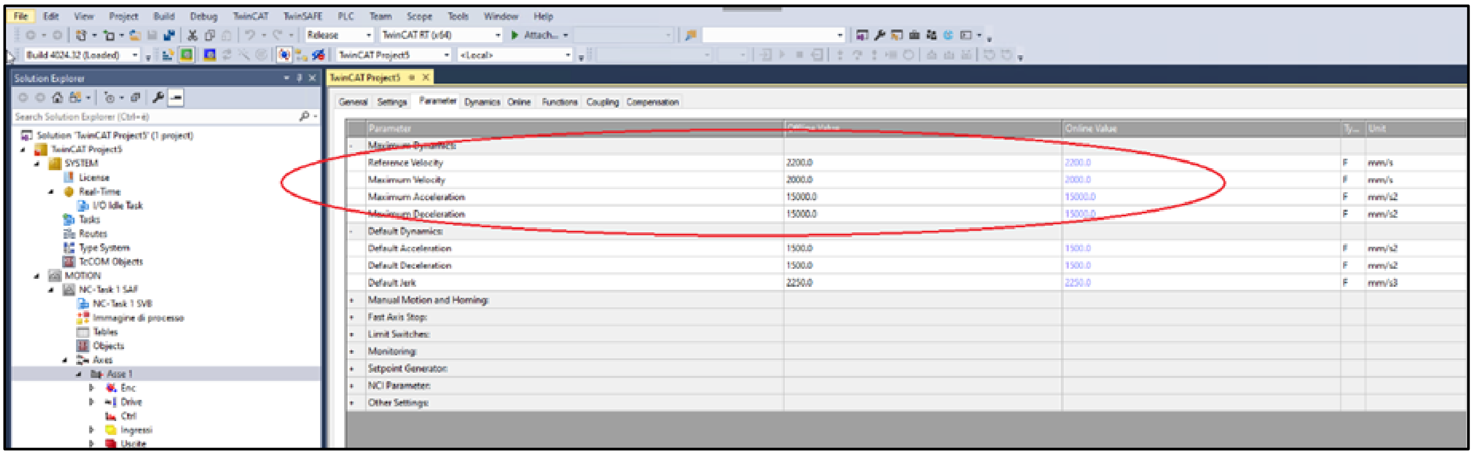
Enabling
☑ Controller
☑ Feed Fw
☑ Feed Bw

Set Enabling
□ Controller
☑ Feed Fw
☑ Feed Bw
Override [%]: 100

Figura 49

Note:To disable the drive:

- NC-Online Screen > **Enabling** > Clear the option **Controller** > **OK**
- Use the motion buttons F1, F2, F3 and F4 to generate the following motion profiles.
- Configure the velocity in the controller using **NC-Configuration** > **NC- Task1 SAF** > **Axes** > **Axis1** > **Parameters** > **Manual Velocity** (Slow and Fast), as shown in the following figure.

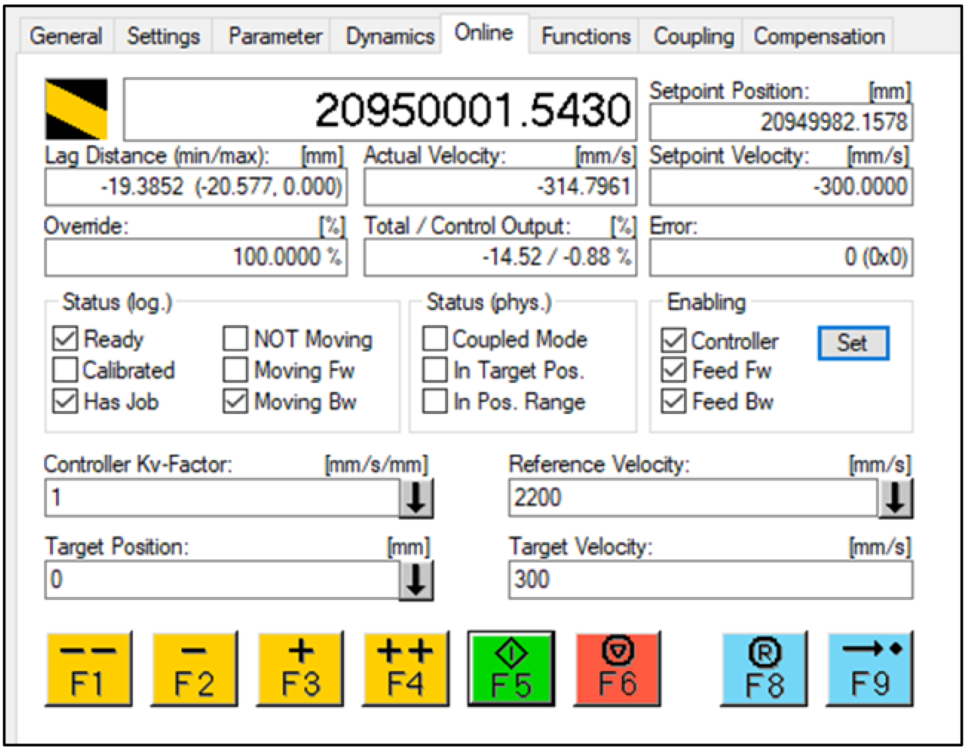


Parameter	Offline Value	Online Value	Ty...	Unit
Maximum Dynamics:				
Reference Velocity	2200.0	2200.0	F	mm/s
Maximum Velocity	2000.0	2000.0	F	mm/s
Maximum Acceleration	15000.0	15000.0	F	mm/s2
Maximum Deceleration	15000.0	15000.0	F	mm/s2
Default Dynamics:				
Default Acceleration	1500.0	1500.0	F	mm/s2
Default Deceleration	1500.0	1500.0	F	mm/s2
Default Jerk	2250.0	2250.0	F	mm/s3
Manual Motion and Homing:				
Fast Axis Stop:				
Limit Switches:				
Monitoring:				
Setpoint Generator:				
NCI Parameter:				
Other Settings:				

Figura 50

- **F1**: Sends a jog command in the negative direction (CCW) with a fast velocity.
- **F2**: Sends a jog command in the negative direction (CCW) with slow velocity.
- **F3**: Sends a jog command in the positive direction (CW) with slow velocity.
- **F4**: Sends a jog command in the positive direction (CW) with fast velocity.

a) Set values for **Target Position** and **Target Velocity** as shown in the following figure.



20950001.5430

Setpoint Position: [mm] 20949982.1578

Lag Distance (min/max): [mm] -19.3852 (-20.577, 0.000)

Actual Velocity: [mm/s] -314.7961

Setpoint Velocity: [mm/s] -300.0000

Override: [%] 100.0000 %

Total / Control Output: [%] -14.52 / -0.88 %

Error: 0 (0x0)

Status (log.)
☑ Ready
□ Calibrated
☑ Has Job
□ NOT Moving
□ Moving Fw
☑ Moving Bw

Status (phys.)
□ Coupled Mode
□ In Target Pos.
□ In Pos. Range

Enabling
☑ Controller
☑ Feed Fw
☑ Feed Bw

Controller Kv-Factor: [mm/s/mm] 1

Reference Velocity: [mm/s] 2200

Target Position: [mm] 0

Target Velocity: [mm/s] 300

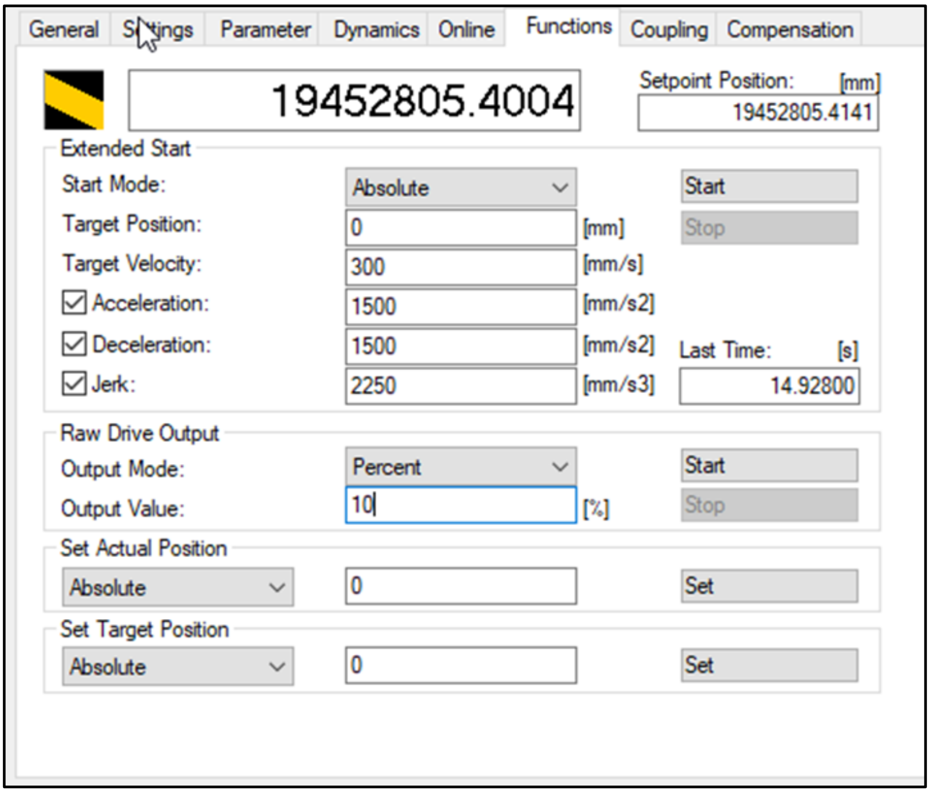
Figura 51

- Press **F5** (green button) to start the motion profile in Synchronous Position mode.
- Press **F6** (red button) to stop the motion.
- Press **F8** (blue button) to clear any faults.

Generating Absolute and Relative Motion

To generate absolute or relative motion in Position Profile mode, refer to the following two figures, and do the following:

- Go to the **Functions** tab.
- Configure the target position, the target velocity, the acceleration and deceleration, and the jerk of the motion.



19452805.4004

Setpoint Position: [mm] 19452805.4141

Extended Start

Start Mode: Absolute

Target Position: 0 [mm]

Target Velocity: 300 [mm/s]

☑ Acceleration: 1500 [mm/s2]

☑ Deceleration: 1500 [mm/s2]

☑ Jerk: 2250 [mm/s3]

Last Time: [s] 14.92800

Raw Drive Output

Output Mode: Percent

Output Value: 10 [%]

Set Actual Position

Absolute 0

Set Target Position

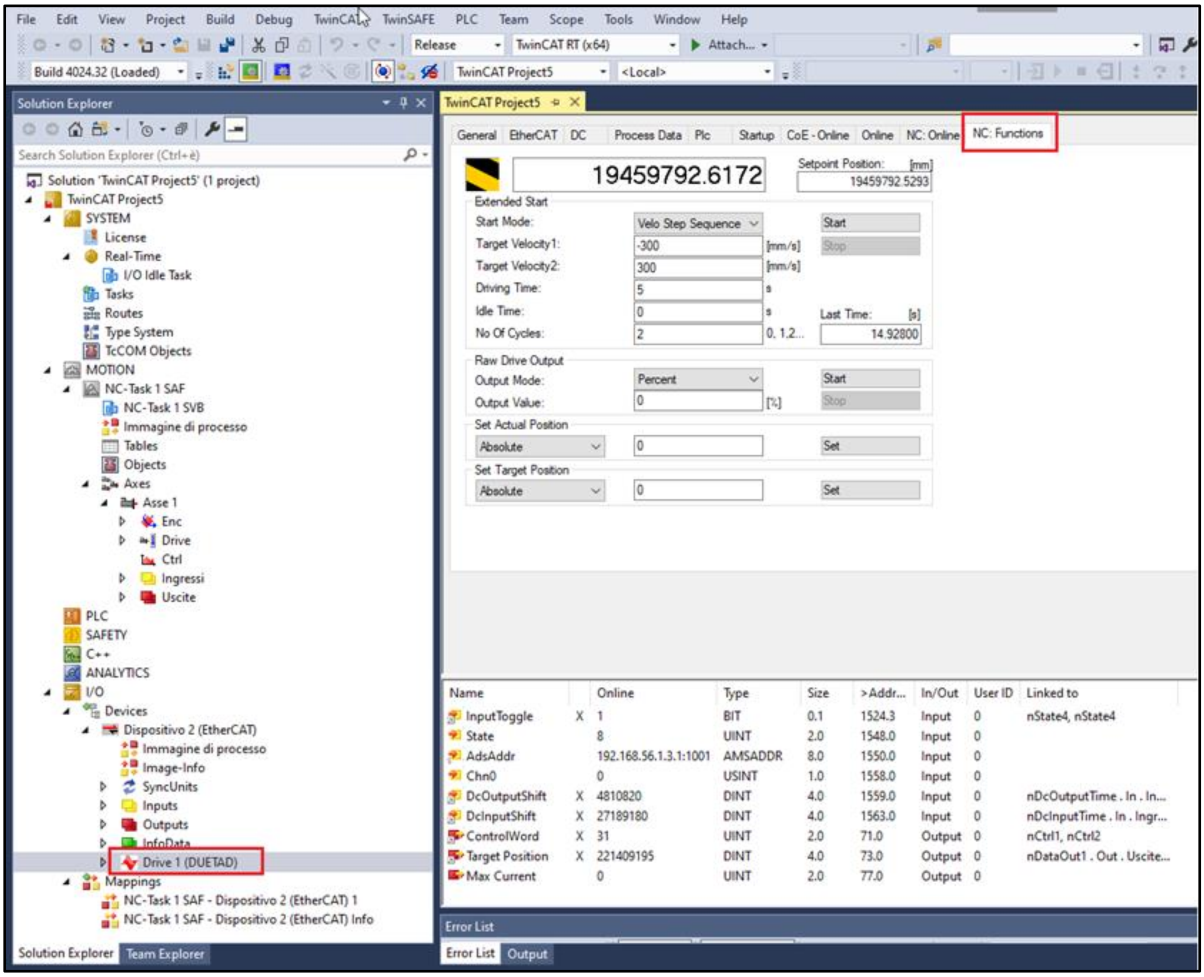
Absolute 0

Figura 52

Generating a Step Motion

To generate a step sequence in the Velocity profile, refer to the following two figures, and do the following:

- Go to the **Functions** tab.
- Configure the target velocity, and a time (duration) for the step.



19459792.6172

Setpoint Position: [mm] 19459792.5293

Extended Start

Start Mode: Velo Step Sequence
Target Velocity1: -300 [mm/s]
Target Velocity2: 300 [mm/s]
Driving Time: 5 s
Idle Time: 0 s
No Of Cycles: 2 0, 1,2...
Last Time: [s] 14.92800

Raw Drive Output

Output Mode: Percent
Output Value: 0 [%]

Set Actual Position

Absolute 0

Set Target Position

Absolute 0

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
InputToggle	X	1	BIT	0.1	1524.3	Input	0	nState4, nState4
State		8	UINT	2.0	1548.0	Input	0	
AdsAddr		192.168.56.1.3.1:1001	AMSADDR	8.0	1550.0	Input	0	
Chn0		0	USINT	1.0	1558.0	Input	0	
DcOutputShift	X	4810820	DINT	4.0	1559.0	Input	0	nDcOutputTime . In . In...
DcInputShift	X	27189180	DINT	4.0	1563.0	Input	0	nDcInputTime . In . Ingr...
ControlWord	X	31	UINT	2.0	71.0	Output	0	nCtrl1, nCtrl2
Target Position	X	221409195	DINT	4.0	73.0	Output	0	nDataOut1 . Out . Uscite...
Max Current		0	UINT	2.0	77.0	Output	0	

Figura 53

9. Firmware upgrade over FoE with EtherCAT

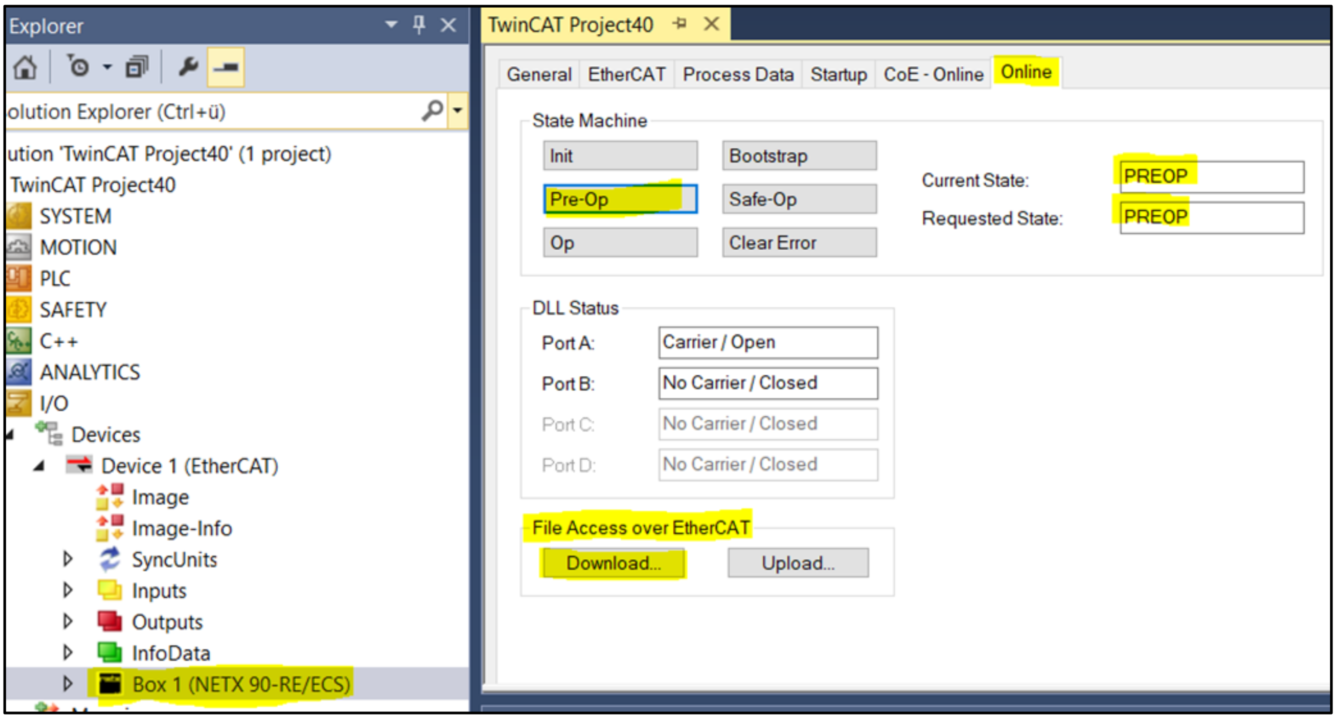
FoE (File Access Over EtherCAT) is a simple protocol similar to TFTP (Trivial File Transfer Protocol) enables file access in a device and a uniform firmware upload to devices across a network. The protocol has been deliberately specified in a lean manner, so that it can be supported by boot loader programs – a TCP/IP stack isn't required.

DuetAD supports FoE for upload firmwares in an easy and fast way. Two types of files are supported:

- *FWUPDATE.ZIP* is a specific compressed set of files given by Motor Power Company for update the EtherCAT stack firmware
- *FW_DuetDownload.bin* is a binary file that contains the drive section of the DuetAD

EtherCAT Stack FW Update

1. Start TwinCat. Change the device state to PREOP and select "Download...":



Current State: PREOP

Requested State: PREOP

Port A: Carrier / Open

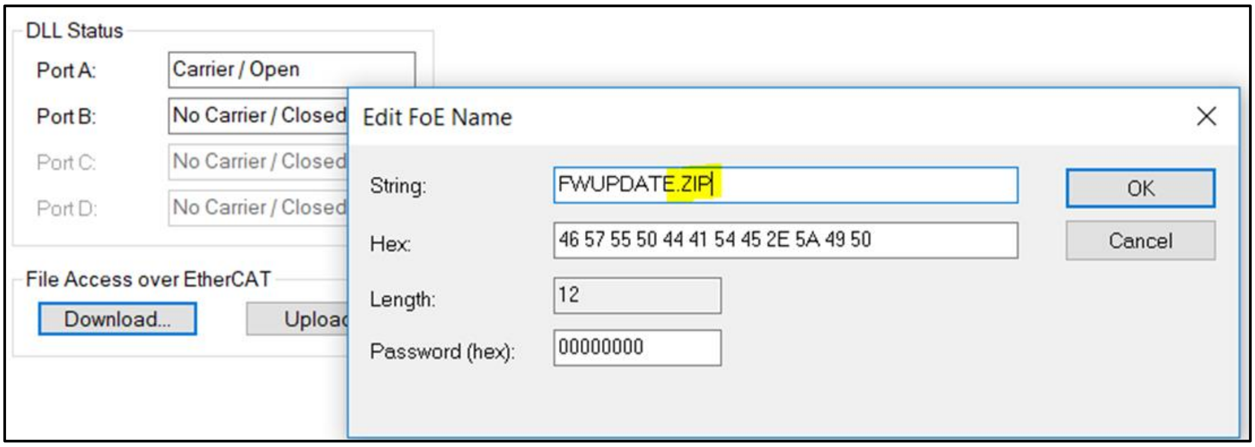
Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Figura 54

2. Select the zip folder given by Motor Power Company. The name must be FWUPDATE.ZIP. Do not forget to add .ZIP:



DLL Status

Port A: Carrier / Open

Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

File Access over EtherCAT

Edit FoE Name

String: FWUPDATE.ZIP

Hex: 46 57 55 50 44 41 54 45 2E 5A 49 50

Length: 12

Password (hex): 00000000

Figura 55

3. Click OK and wait for download process to be finished. You can see the progress status at the right bottom of Twincat. It can takes several seconds to finish.

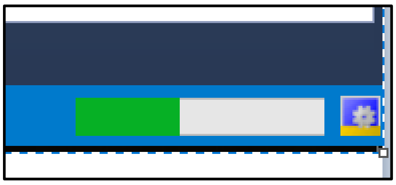
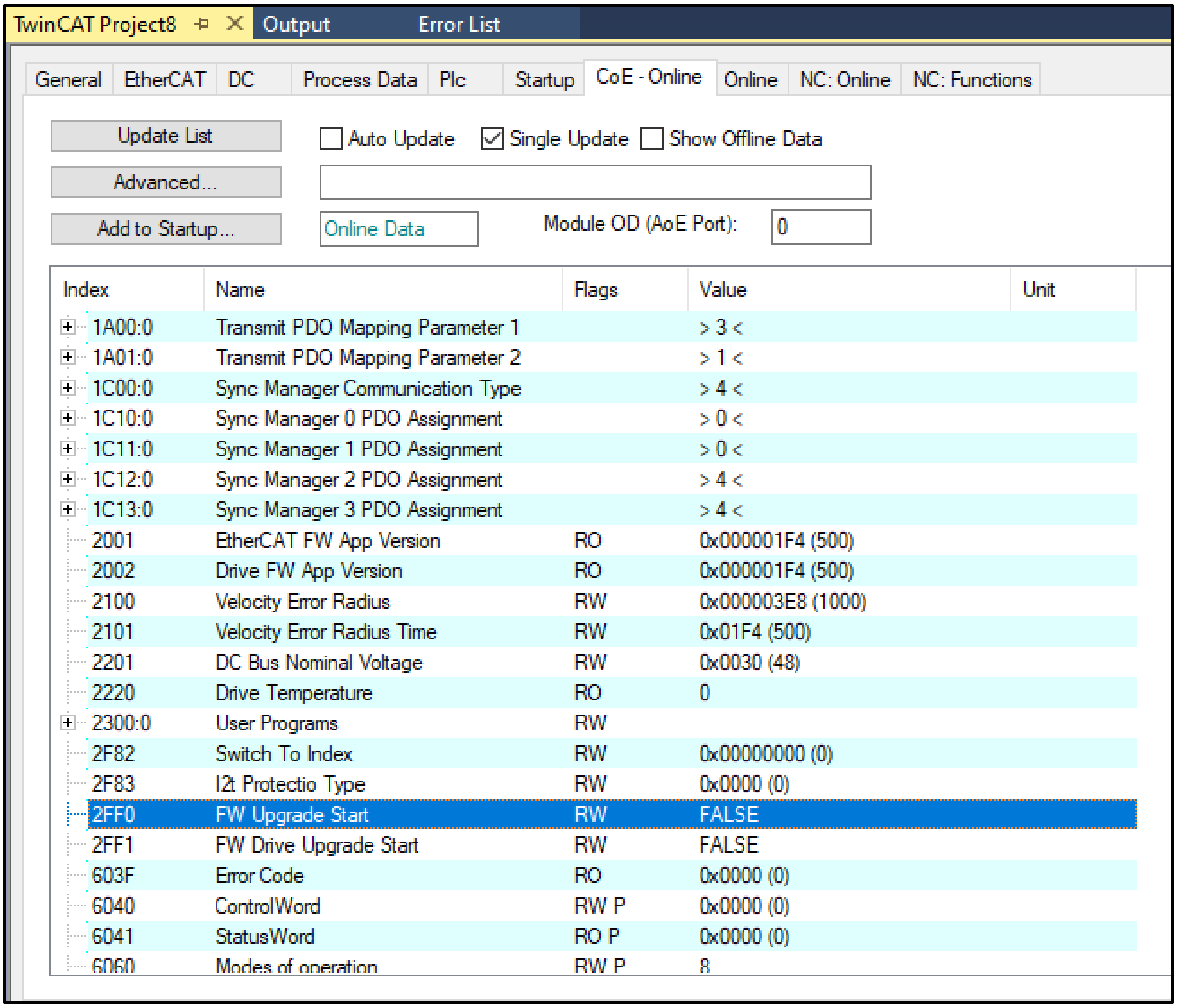


Figure 56

4 After the download finished successfully, go to CoE-Online. Try to write TRUE to object 2FF0, the update reset command will be trigged.



□ Auto Update ☑ Single Update □ Show Offline Data

Online Data

Module OD (AoE Port): 0

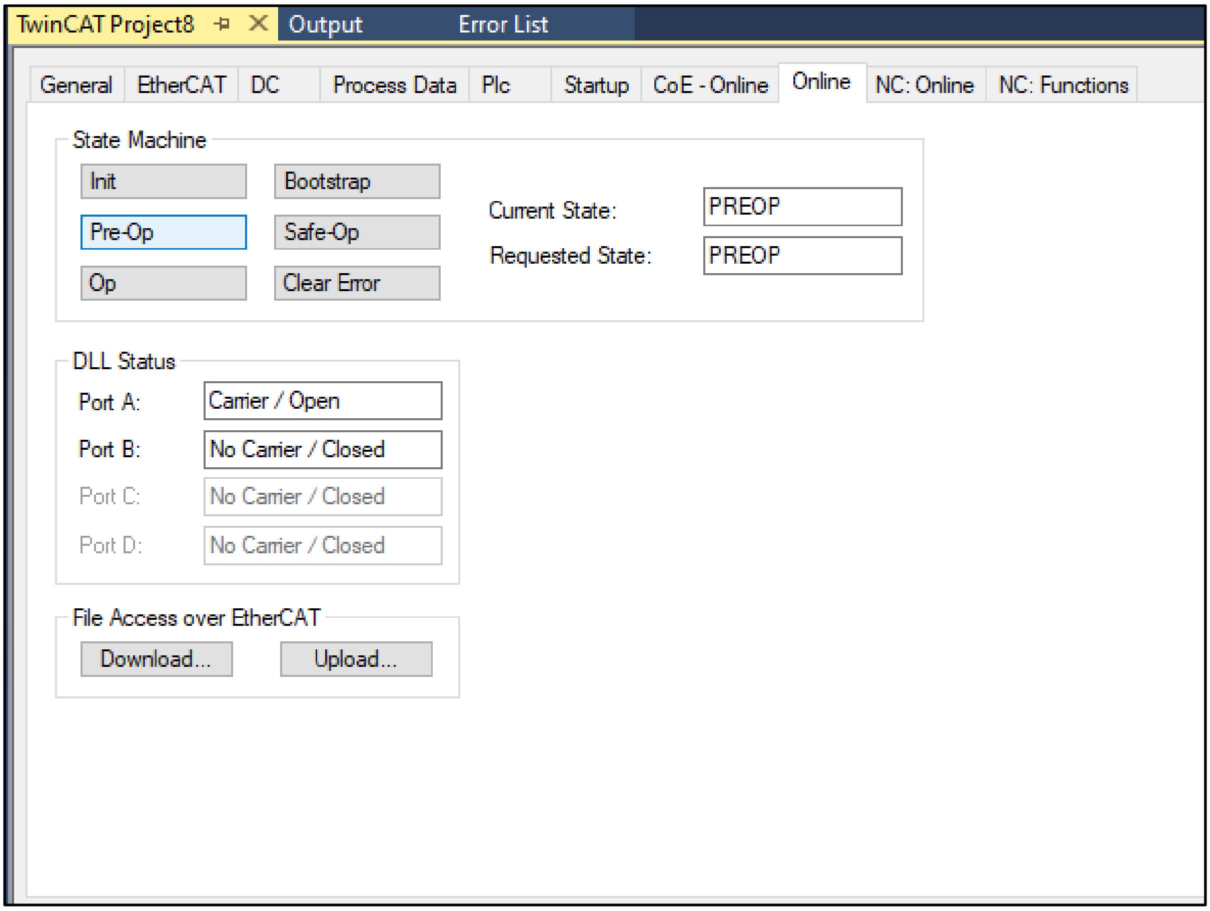
Index	Name	Flags	Value	Unit
⊞ 1A00:0	Transmit PDO Mapping Parameter 1		> 3 <	
⊞ 1A01:0	Transmit PDO Mapping Parameter 2		> 1 <	
⊞ 1C00:0	Sync Manager Communication Type		> 4 <	
⊞ 1C10:0	Sync Manager 0 PDO Assignment		> 0 <	
⊞ 1C11:0	Sync Manager 1 PDO Assignment		> 0 <	
⊞ 1C12:0	Sync Manager 2 PDO Assignment		> 4 <	
⊞ 1C13:0	Sync Manager 3 PDO Assignment		> 4 <	
2001	EtherCAT FW App Version	RO	0x000001F4 (500)	
2002	Drive FW App Version	RO	0x000001F4 (500)	
2100	Velocity Error Radius	RW	0x000003E8 (1000)	
2101	Velocity Error Radius Time	RW	0x01F4 (500)	
2201	DC Bus Nominal Voltage	RW	0x0030 (48)	
2220	Drive Temperature	RO	0	
⊞ 2300:0	User Programs	RW		
2F82	Switch To Index	RW	0x00000000 (0)	
2F83	I2t Protectio Type	RW	0x0000 (0)	
2FF0	FW Upgrade Start	RW	FALSE	
2FF1	FW Drive Upgrade Start	RW	FALSE	
603F	Error Code	RO	0x0000 (0)	
6040	ControlWord	RW P	0x0000 (0)	
6041	StatusWord	RO P	0x0000 (0)	
6060	Modes of operation	RW P	8	

Figura 57

5 Wait untill the procedure will end. If procedure was successful the node will restart automatically.

Drive FW Update via FoE

1 Start TwinCat. Change the device state to PREOP:



Current State: PREOP

Requested State: PREOP

Port A: Carrier / Open

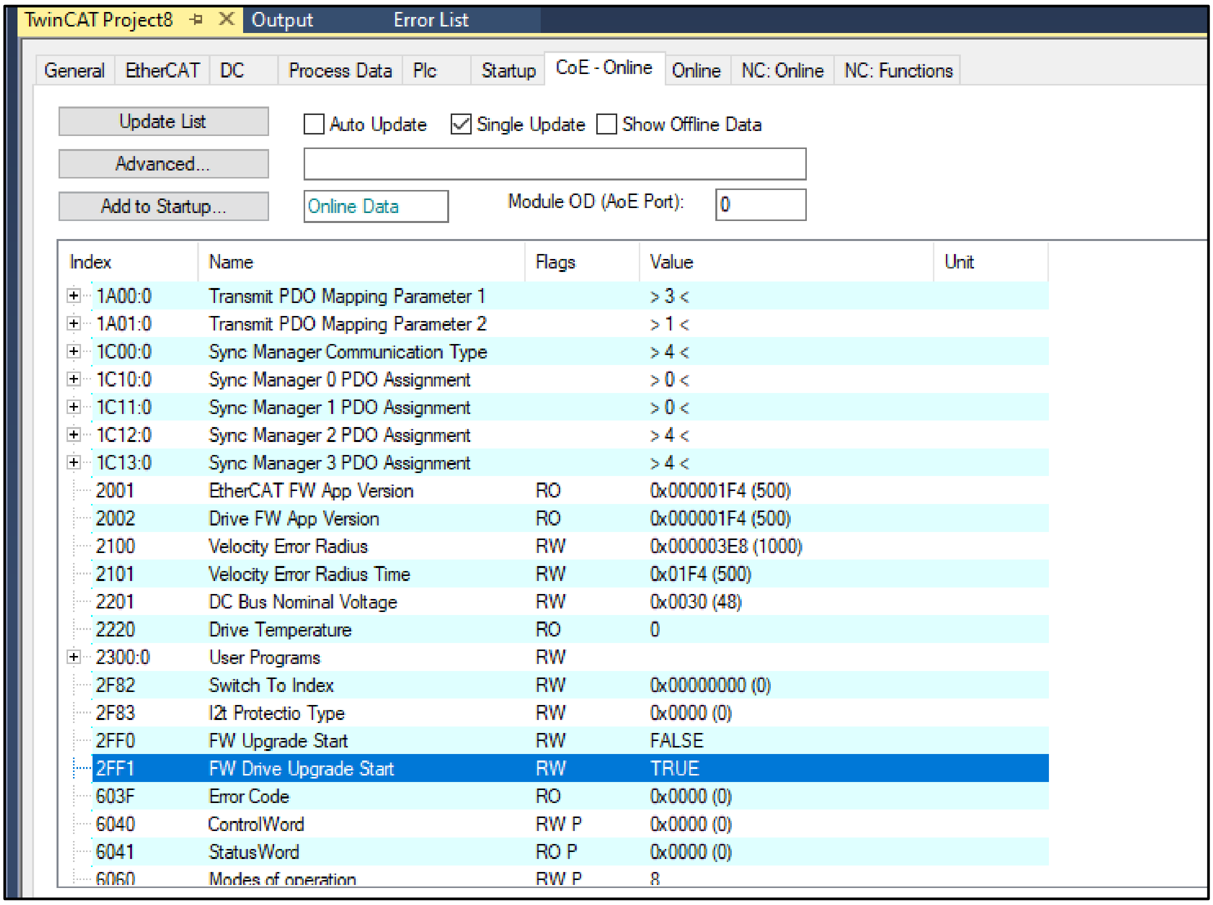
Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Figura 58

2 Now the Drive FW part has to be put in set in Bootloader Mode. This will be done by putting TRUE to object 0x2FF1. Wait at least 3 seconds for the following steps. Ignore node errors if they occurs in this part.



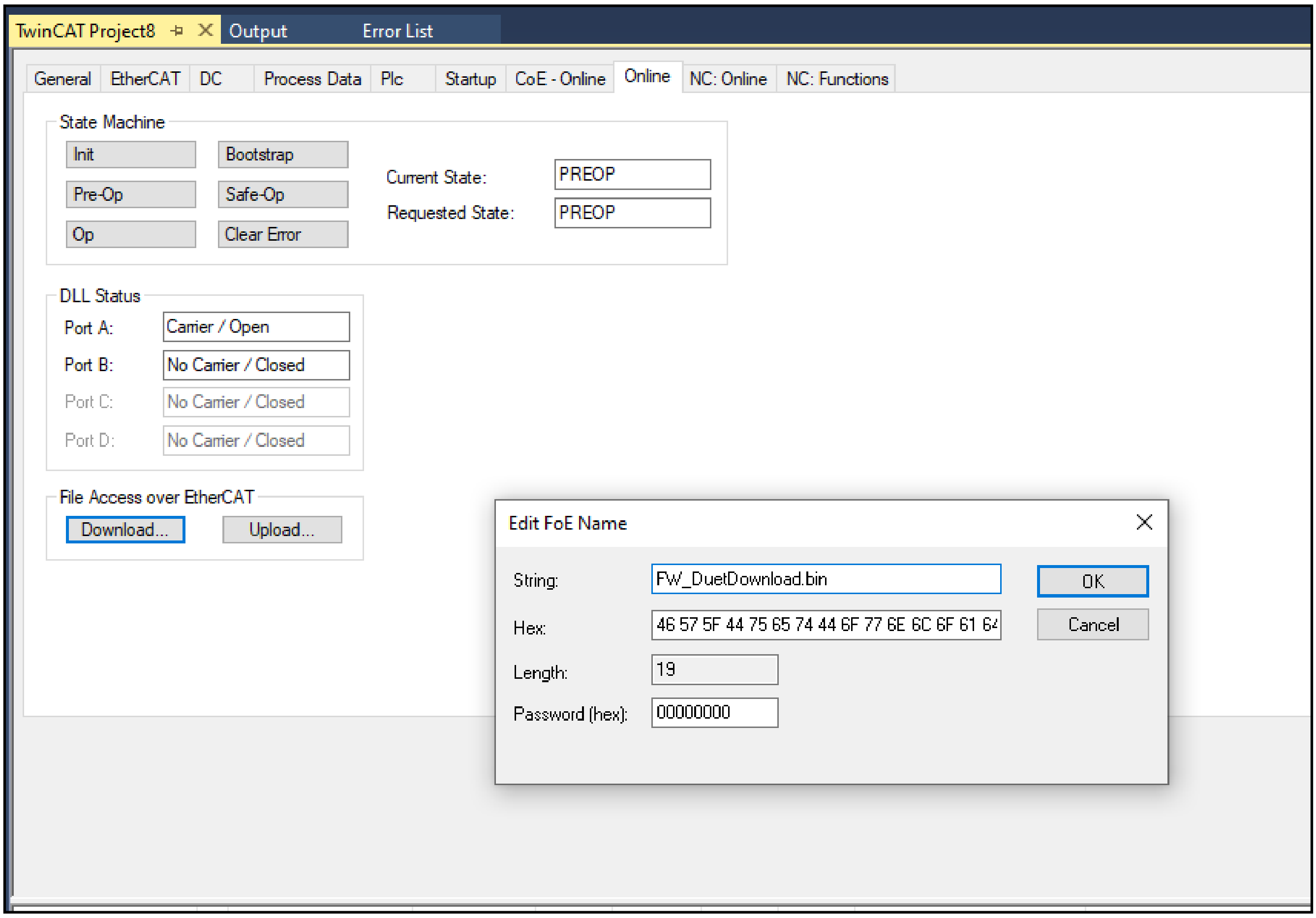
□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1A00:0	Transmit PDO Mapping Parameter 1		> 3 <	
1A01:0	Transmit PDO Mapping Parameter 2		> 1 <	
1C00:0	Sync Manager Communication Type		> 4 <	
1C10:0	Sync Manager 0 PDO Assignment		> 0 <	
1C11:0	Sync Manager 1 PDO Assignment		> 0 <	
1C12:0	Sync Manager 2 PDO Assignment		> 4 <	
1C13:0	Sync Manager 3 PDO Assignment		> 4 <	
2001	EtherCAT FW App Version	RO	0x000001F4 (500)	
2002	Drive FW App Version	RO	0x000001F4 (500)	
2100	Velocity Error Radius	RW	0x000003E8 (1000)	
2101	Velocity Error Radius Time	RW	0x01F4 (500)	
2201	DC Bus Nominal Voltage	RW	0x0030 (48)	
2220	Drive Temperature	RO	0	
2300:0	User Programs	RW		
2F82	Switch To Index	RW	0x00000000 (0)	
2F83	I2t Protectio Type	RW	0x0000 (0)	
2FF0	FW Upgrade Start	RW	FALSE	
2FF1	FW Drive Upgrade Start	RW	TRUE	
603F	Error Code	RO	0x0000 (0)	
6040	ControlWord	RW P	0x0000 (0)	
6041	StatusWord	RO P	0x0000 (0)	
6060	Modes of operation	RW P	8	

Figura 59

3 Now select the bin file given by Motor Power Company. The name must be FW_DuetDownload.bin. Then download it. Do not forget to add .bin



Current State: PREOP

Requested State: PREOP

Port A: Carrier / Open

Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Edit FoE Name

String: FW_DuetDownload.bin

Hex: 46 57 5F 44 75 65 74 44 6F 77 6E 6C 6F 61 64

Length: 19

Password (hex): 00000000

Figura 60

4 Click OK and wait for download process to be finished. You can see the progress status at the right bottom of Twincat. It can takes several seconds to finish.

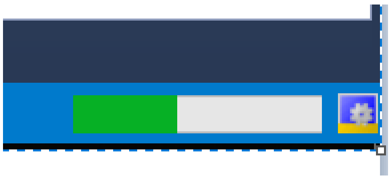


Figura 61

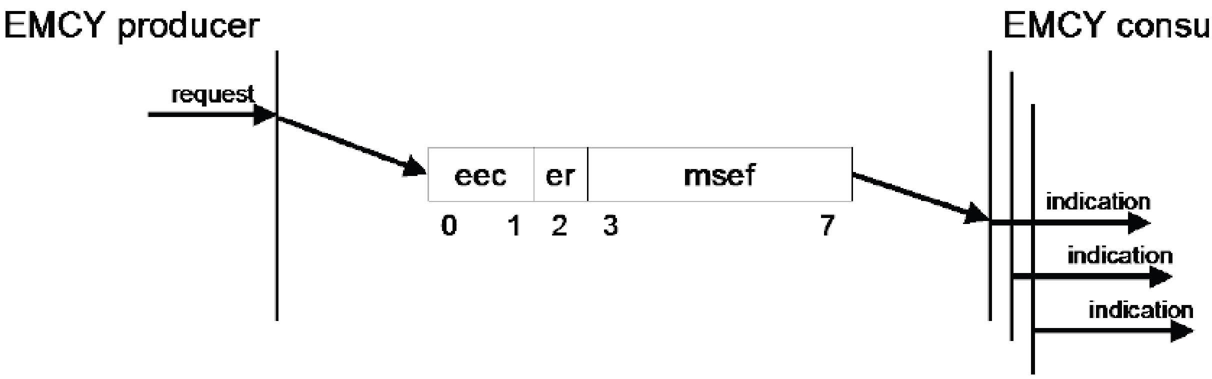
5 Wait untill the procedure will end. If procedure was successful the node will restart automatically.

10. CANopen And EtherCAT Errors

Error codes

Emergency messages are triggered by internal errors and severe warnings detected within the drive device. They are defined in detail in the /CiA402-3/. They shall contain the 16-bit error code. Error codes from xx00$_h$ to xx7F$_h$ are defined in /CiA301/ or in Table 24. Error codes between xx80$_h$ and xxFF$_h$ are manufacturer-specific.

NOTE Here, the "xx" stands substitutionally for any two-place hexadecimal number.



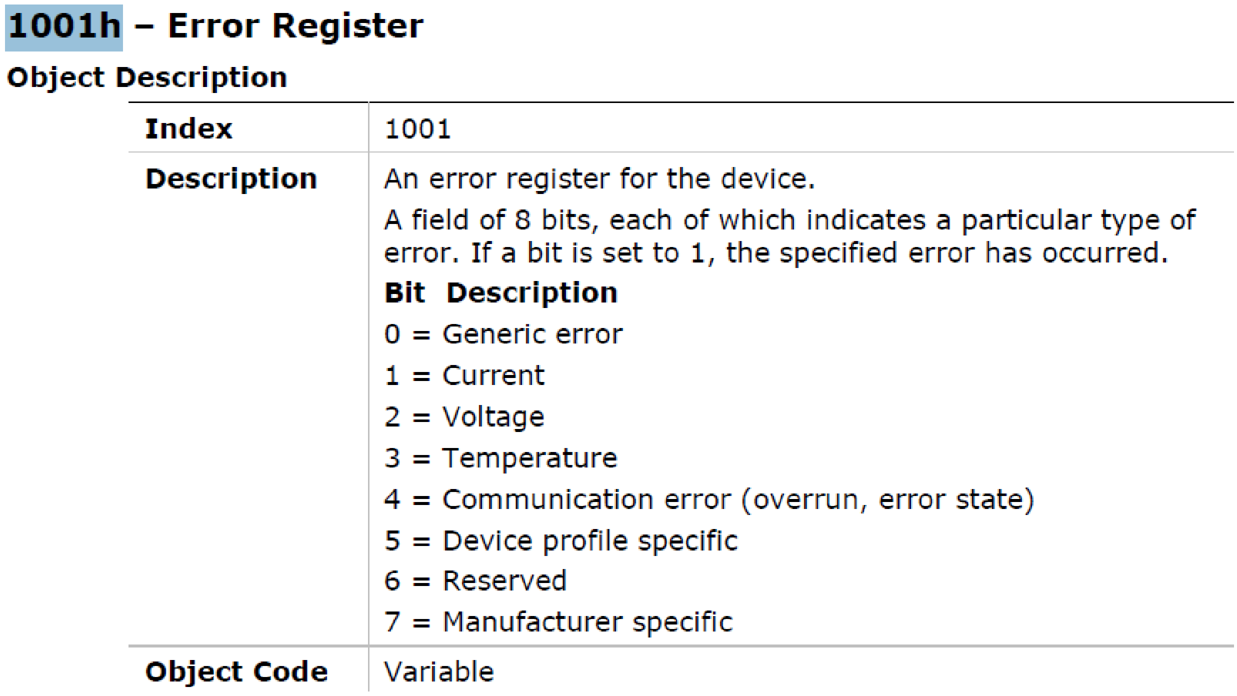
- **eec:** Emergency error code (see Table 26)
- **er:** Error register (see object 1001$_h$)
- **msef:** Manufacturer-specific error code

Figure 38: Protocol EMCY write

Figura 62

When an illegal state occurs in the drive, the drive sends the code to the master device as object 603Fh (Error Code).

Whenever the value of 603Fh is not zero, there is a fault in the drive. The CANopen state machine enters Fault mode, and the drive cannot be enabled.



1001h – Error Register

Object Description

Index	1001
Description	An error register for the device. A field of 8 bits, each of which indicates a particular type of error. If a bit is set to 1, the specified error has occurred. **Bit Description** 0 = Generic error 1 = Current 2 = Voltage 3 = Temperature 4 = Communication error (overrun, error state) 5 = Device profile specific 6 = Reserved 7 = Manufacturer specific
Object Code	Variable

Figura 63

ERROR TYPE	ERROR CODE
ERROR_NO_ERROR	0x0000
ERROR_ErrorReport_ParametersNotSupp	0x6100
ERROR_isr_low_WrongInterrupt	0x6110
ERROR_wrong_PDO_mapping	0x6120
ERROR_isr_timer_overflow	0x6000
ERROR_WrongNodeIDorBitRate	0x6300
ERROR_CAN_RXB_OVERFLOW	0x8110
ERROR_CAN_TX_BUS_OFF	0x8140
ERROR_CAN_TX_BUS_PASSIVE	0x8120
ERROR_CAN_RX_BUS_PASSIVE	0x8120
ERROR_CAN_TX_OVERFLOW	0x8110
ERROR_TPDO_OUTSIDE_WINDOW	0x8100
ERROR_HEARTBEAT_CONSUMER	0x8130
ERROR_SYNC_TIME_OUT	0x8100
ERROR_USER_0_IMPORTANT	0x1001
ERROR_USER_1_IMPORTANT	0x1002
ERROR_USER_2_IMPORTANT	0x1003
ERROR_USER_3_IMPORTANT	0x1004
ERROR_USER_4	0x1005
ERROR_USER_5	0x1006

ERROR_USER_6	0x1007
ERROR_USER_7	0x1008
ERROR_CO_RXMSG_PDOlength	0x8210
ERROR_CO_RXMSG_Length	0x8200
ERROR_CO_RXMSG_NMTcmd	0x8200
ERROR_CO_RXMSG_Inhibit	0x8200
ERROR_CAN_BUS_WARNING	
ERROR_SYNC_EARLY	0x8100
ERROR_CANRXTX_INT	0x8100
ERROR_RATED_MOTOR_CURRENT	0x2312
ERROR_PEAK_MOTOR_CURRENT	0x8321
ERROR_SHORT_CIRCUIT	0x2340
ERROR_VEL_SPEED_CNTL	0x8400
ERROR_POSITION_CNTL	0x8500
ERROR_DC_LINK_OVER_VOLTAGE 0x28	0x3210
ERROR_DC_LINK_UNDER_VOLTAGE	0x3220
ERROR_ENCODER_UNPLUGGED	0x7305
ERROR_ENCODER_ERROR	0x7300
ERROR_DRIVE_OVERTEMPERATURE	0x4310
ERROR_WWD_RESET	0x6010
ERROR_LOSS_OF_PARAMETERS	0x6310
ERROR_PARAMETERS_ERRORS	0x6320