CANopen & EtherCAT DS402 Specification

for 'Titanio-Platino-Vanadio' EVER Drives

Software Manual

Release : 1.05 Build 00

TITANIO VECTOR · SERVO · DRIVES

Platino BLDC · SERVO · DRIVES

VANADIO AC · SERVO · DRIVES

IMPORTANT NOTICE

EVER Elettronica srl
VIA DEL Commercio , 9/11 Loc. S. Grato Z.I.
26900 – LODI - ITALY
Tel. ++39(0)371412318 Fax ++39(0)371412367
E-mail: *infoever@everelettronica.it*
Web: *www.everelettronica.com*

Release History:

Release	*Date*	*Author*	*Description*
1.00 Build 00	23.02.2021	Rota L.	- First issue.
1.01 Build 00	31.05.2021	Rota L.	- Modified description of *60EF.0H*, *2012.1H*, *2012.2H* objects and related functions. - Added *Feedback_Sensor_Calibration_mode*. - Added description of bit12, bit13, bit14 of *Feedback_Settings* object. - Added '*Brake Control*' feature (*§7.6*) - Added *2C00.0H* , *2C01.0H*, *2C02.0H*, *2C03.0H*, *2C04.0H* objects.
1.02 Build 00	09.07.2021	Rota L.	- Modified description of the Halt function in the *Controlword* object. - Added *6007.0H* object. - Modified description of *Emergency Codes* Table.
1.03 Build 00	31.03.2022	Rota L.	- Added 'Braking Resistor' feature (§7.7) - Added *2B80.0H* , *2B81.0H* , *2B82.0H* , *2B83.0H* , *2B84.0H* objects
1.04 Build 00	07.06.2022	Rota L.	- Modified description of *Brake_Control_Settings* object.
1.05 Build 00	02.10.2023	Rota L.	- Modified description of *607F.0H* object. - Added description of bit7 in *DS402_Working_Settings* object. - Added Homing methods -1, -2, -3, -4, 33, 34 - Added *2220.07H* object - Added *2220.09H* object

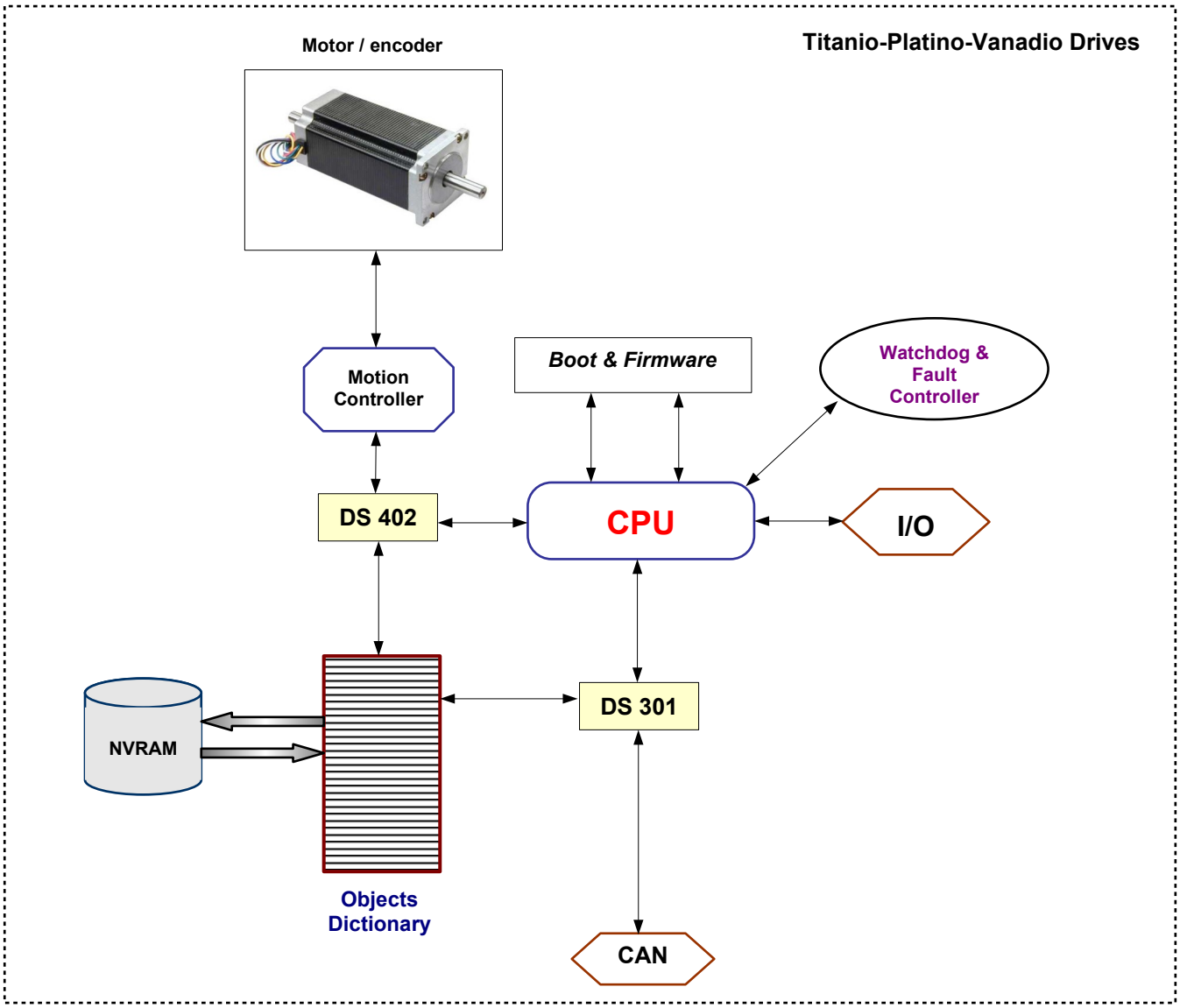
Related Publications

CiA DS 301 V4.02	**CANopen Application Layer and Communication Profile**
CiA DSP 402 V2.0	**CANopen Device Profile for Drives and Motion Control**
IEC-61800-7-201:2015	**Adjustable speed electrical power drive systems**

1.0 Introduction

The Titanio-Platino-Vanadio drives use a subset of the standard CANopen protocol to provide access to whole drive parameters. Several standard CANopen functions codes are supported as described in the CiA DS301 and CiA DS402. The Titanio-Platino-Vanadio family drives are CANopen slave devices and then they need a CANopen master system (PC, PLC, etc.) to be configured and managed by CAN bus. This manual describes the Ethercat and CANopen functionalities.

1.1 Drives Working Logic



2.0 CANopen Protocol

The CANopen protocol is one of the most common CAN protocols. Since 1995 the CANopen specification is handed over to CAN in Automation (CiA) international users and manufacturers group. The CANopen Device Specification version 4.01 has been accepted by the European standardization authorities as EN 50325-4.

The main concept of CANopen is based on use of an object dictionary. The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. To access to these objects two methods are used: SDO & PDO that are explained further in this manual.

2.1 CANopen Protocol Parameters

For drives with firmware V00r73 or lower :

CANopen Specifications	
CANopen Functionality	Slave
Device Id Number	1 to 127
Baud Rate Supported (Kbits)	1000,500,250,125
NMT	Slave
Server SDOs	1 (Standard)
Client SDOs	No
Receive PDOs	2
Transmit PDOs	2
PDO Mapping	Static
Emergency Telegram	Yes
Nodeguarding	No
Heartbeat	Yes
Sync supported	Yes

For drives with firmware V00r74 or superior :

CANopen Specifications	
CANopen Functionality	Slave
Device Id Number	1 to 127
Baud Rate Supported (Kbits)	1000,500,250,125
NMT	Slave
Server SDOs	1 (Standard)
Client SDOs	No
Receive PDOs	4
Transmit PDOs	4
PDO Mapping	Dynamic and Static
Emergency Telegram	Yes
Nodeguarding	Yes
Heartbeat	Yes
Sync supported	Yes

2.2 Baud Rate & Node Id Selection

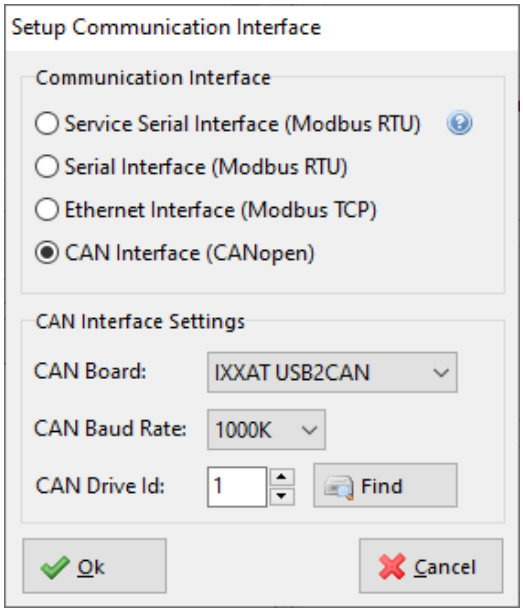
Drives with dip-switches and/or rotoswitches

For drives fitted with dips-switches and rotoswitches, look at hardware short manual for details on settings.

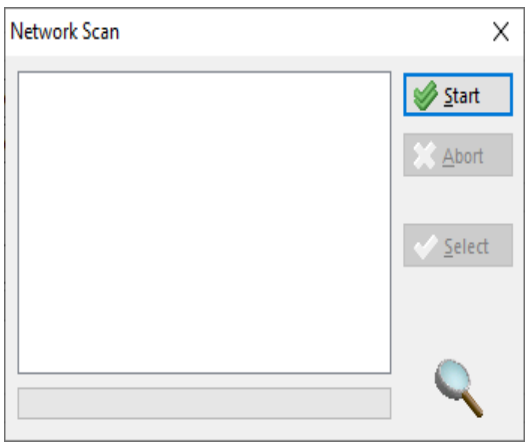
Drives without dip-switches and/or rotoswitches

The drives without dip-switches and rotoswitches can be configured by means of '***EVER Studio***' Software Tool (§8.0). The drives factory settings is baud rate = 500 Kbit and NodeId = 1. Follows the steps to perform to change the default settings:

- Open '***EVER Studio***' Software Tool (§8.0).
- From the main window, press on '***Change***' button in 'Drive Info & Communication' frame to open 'Setup Communication Interface' window:

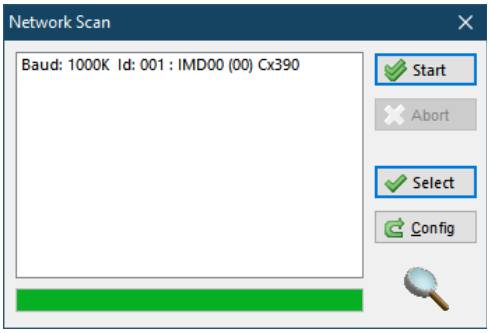


- Press on ***"Find"*** button :

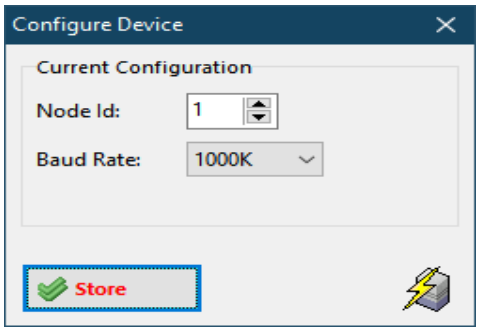


- Press on ***"Start"*** button.

- If the drive has been detected its system code will be added to the list box.



- Select the device and press on ***"Config"*** button. The following window will appear:



- Change the Node Id and Baud Rate settings as desired and press on ***"Store"*** button.
- The new settings will be used at the next drive's switch on.

Instead of using the '***EVER Studio***' Software Tool (§8.0), the Baud Rate & Node Id can be changed directly writing on the objects (*4000.7H* & *4000.8H*) according to the following instructions:

1. Write using SDO service 2 bytes in the object *4000.7H* (Node Id) keeping the high byte equal to 0xAA and setting in the low byte the new Id. (example to set the Node Id = 5 write 0xAA05)
2. Write using SDO service 2 bytes in the object *4000.8H* (Baud Rate) keeping the high byte equal to 0x55 and setting in the low byte the new baud rate according to the following table (example, to set the baud rate = 500K write 0x5501) :

Value	Baud Rate
0	1M
1	500K
2	250K
3	125K

3. After having written the new value it is necessary to wait about 1 second to permit to the system to store the new values in NVRAM.
4. The new BaudRate & NodeId will be effective at the next drive switch on.

2.3 CANopen SDO (Service Data Object)

Service Data Objects are used to establish a peer to peer connection between two CANopen devices. This kind of connection is based on a Client/Server based mechanism.
The SDO server is the device that is serving the object dictionary to which the access is required.
The SDO client is the device that wants to access the object dictionary of a specific device.
The SDO service is based on two CAN messages with different identifiers. One message is used by the SDO client and the second message is used by the SDO server.
There are two different methods for SDO download/upload:

➔ **Expedited SDO transfer:**
- For objects long no more than 4 bytes.

➔ **Segmented SDO transfer:**
- For objects longer than 4 bytes.

Request (Client → Server)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
600H+NodeId	ccs/cntrl	Object Index		SubIndex	Data (Optional)			

Response (Client ← Server)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
580H+NodeId	scs/cntrl	Object Index		SubIndex	Data (Optional)			

Examples:

SDO - Expedited protocol download (write an object 4 bytes long):

Request (Client → Server)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
600H+NodeId	22H	Index		SubIndex	Value			

Response (Client ← Server)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
580H+NodeId	60H	Reserved						

SDO - Expedited protocol upload (read an object 4 bytes long):

Request (Client → Server)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
600H+NodeId	40H	Index		SubIndex	Reserved			

Response (Client ← Server)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
580H+NodeId	43H	Index		SubIndex	Value			

2.4 CANopen PDO (Process Data Object)

Process Data Objects (PDO) are used to transmit any process data for the process control.
The PDOs are transmitted in broadcast and without any confirmation back to the transmitting device.

There are two types of PDOs :

- ➔ **Receive PDO** (RX_PDO, from master to slave)
- ➔ **Transmit PDO** (TX_PDO, from slave to master)

All PDOs can be asynchronous or synchronous :
The Receive PDOs are handled as soon as possible after their receipt if set as asynchronous.
For Transmit PDOs has been implemented *180x.5H* objects as described in the standard CiA DS 301 V4.01 that permits to specify also a transmission frequency for asynchronous PDOs.

Static Mapping is available only for drives with firmware V00r73 or lower.
Dynamic Mapping and Static Mapping are available for drives with firmware V00r74 or superior.

The Static Mapping defines the preset configuration also for Dynamic Mapping.

2.4.1 Static PDO Mapping

A Static PDO Mapping list is available for RX_PDO and TX_PDO. By mean of (*2200.5h* obj) and (*2200.1Ah* obj) is possible to choose Static PDO Mapping. Static Mapping list is available only for CANbus drives, for EtherCAT refers to *§5.2*.

RX_PDO1 Static Mapping list :

Static Mapping #0 (*2200.5h* obj , bit15÷bit12 = 0)

Cob_ID	B0	B1
200H+NodeId	*6040.0H*	
	Controlword	

Static Mapping #1 (*2200.5h* obj , bit15÷bit12 = 1)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
200H+NodeId	*6040.0H*		*6060.0H*	*6200.1H*	*607A.0H*			
	Controlword		*Modes of operation*	*B0_Digital _Outputs*	*Target_Position* (pp,csp)			

Static Mapping #2 (*2200.5h* obj , bit15÷bit12 = 2)

Cob_ID	B0	B1	B2
200H+NodeId	*6040.0H*		*6060.0H*
	Controlword		*Modes of operation*

RX_PDO2 Static Mapping list :

Static Mapping #0 (*2200.5h* obj , bit7÷bit4 = 0)

Cob_ID	B0	B1	B2
300H+NodeId	*6040.0H*		*6060.0H*
	Controlword		*Modes of operation*

Static Mapping #1 (*2200.5h* obj , bit7÷bit4 = 1)

Cob_ID	B0	B1	B2	B3	B4	B5
300H+NodeId	*6040.0H*		*607A.0H*			
	Controlword		*Target_Position* (pp,csp)			

Static Mapping #3 (*2200.5h* obj , bit7÷bit4 = 3)

Cob_ID	B0	B1	B2	B3
300H+NodeId	*6040.0H*		*6042.0H*	
	Controlword		*vl Target Velocity*	

Static Mapping #4 (*2200.5h* obj , bit7÷bit4 = 4)

Cob_ID	B0	B1	B2	B3	B4	B5
300H+NodeId	*6040.0H*		*60C1.1H*			
	Controlword		*Interpolation Data x1* (ip)			

Static Mapping #5 (*2200.5h* obj , bit7÷bit4 = 5)

Cob_ID	B0	B1	B2	B3	B4	B5
300H+NodeId	*6040.0H*		*60FF.0H*			
	Controlword		*Target_Velocity* (pv)			

Static Mapping #6 (*2200.5h* obj , bit7÷bit4 = 6)

Cob_ID	B0	B1	B2	B3	B4	B5	B6
300H+NodeId	*6040.0H*		*60FF.0H*				*6200.1H*
	Controlword		*Target_Velocity* (pv)				*B0_Digital _Outputs*

Static Mapping #9 (*2200.5h* obj , bit7÷bit4 = 9)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
300H+NodeId	*60C1.1H*				*60FF.0H*			
	Interpolation Data x1 (ip)				*Target_Velocity* (pv)			

Static Mapping #10 (*2200.5h* obj , bit7÷bit4 = 10)

Cob_ID	B0	B1	B2	B3	B4	B5
300H+NodeId	*607A.0H*				*6042.0H*	
	Target_Position (pp,csp)				*vl Target Velocity* (vl)	

RX_PDO3 Static Mapping list :

Static Mapping #0 (2200.1Ah obj , bit15÷bit12 = 0)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
400H+NodeId	*607A.0H*				*6081.0H*			
	Target_Position (pp,csp)				*Profile_Velocity* (pp)			

Static Mapping #1 (2200.1Ah obj , bit15÷bit12 = 1)

Cob_ID	B0	B1	B2	B3
400H+NodeId	*60C1.1H*			
	Interpolation Data x1 (ip)			

RX_PDO4 Static Mapping list :

Static Mapping #0 (2200.1Ah obj , bit7÷bit4 = 0)

Cob_ID	B0	B1	B2	B3
500H+NodeId	*60FF.0H*			
	Target_Velocity (pv)			

TX_PDO1 Static Mapping list :

Static Mapping #0 (*2200.5h* obj , bit11÷bit8 = 0)

Cob_ID	B0	B1
180H+NodeId	*6041.0H*	
	Statusword	

Static Mapping #1 (*2200.5h* obj , bit11÷bit8 = 1)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
180H+NodeId	*6041.0H*		*6061.0H*	*6000.1H*	*6064.0H*			
	Statusword		*Modes of operation display*	*B0_Digital_ Inputs*	*Position_Actual_Value*			

Static Mapping #1 (*2200.5h* obj , bit11÷bit8 = 2)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
180H+NodeId	*6041.0H*		*6061.0H*	*6000.1H*	*60FD.0H*			
	Statusword		*Modes of operation display*	*B0_Digital_ Inputs*	*Digital_Inputs*			

TX_PDO2 Static Mapping list :

Static Mapping #0 (*2200.5h* obj , bit3÷bit0 = 0)

Cob_ID	B0	B1	B2
280H+NodeId	*6041.0H*		*6061.0H*
	Statusword		*Modes of operation display*

Static Mapping #1 (*2200.5h* obj , bit3÷bit0 = 1)

Cob_ID	B0	B1	B2	B3	B4	B5
280H+NodeId	*6041.0H*		*6064.0H*			
	Statusword		*Position_Actual_Value*			

Static Mapping #2 (*2200.5h* obj , bit3÷bit0 = 2)

Cob_ID	B0	B1	B2	B3	B4	B5
280H+NodeId	*6041.0H*		*606C.0H*			
	Statusword		*Velocity_Actual_Value*			

Static Mapping #3 (*2200.5h* obj , bit3÷bit0 = 3)

Cob_ID	B0	B1	B2	B3
280H+NodeId	*6041.0H*		*6044.0H*	
	Statusword		*vl Control Effort*	

Static Mapping #4 (*2200.5h* obj , bit3÷bit0 = 4)

Cob_ID	B0	B1	B2	B3	B4	B5
280H+NodeId	*6041.0H*		*2007.0H*			
	Status Word		*Encoder_Actual_Value[0]*			

Static Mapping #5 (*2200.5h* obj , bit3÷bit0 = 5)

Cob_ID	B0	B1	B2	B3	B4	B5
280H+NodeId	*6041.0H*		*2008.0H*			
	Statusword		*Encoder_Actual_Value[1]*			

Static Mapping #6 (*2200.5h* obj , bit3÷bit0 = 6)

Cob_ID	B0	B1	B2	B3	B4	B5
280H+NodeId	*6041.0H*		*60F4.0H*			
	Statusword		*Following_error_actual_value*			

Static Mapping #7 (*2200.5h* obj , bit3÷bit0 = 7)

Cob_ID	B0	B1	B2	B3
280H+NodeId	*6064.0H*			
	Position_Actual_Value			

Static Mapping #8 (*2200.5h* obj , bit3÷bit0 = 8)

Cob_ID	B0	B1	B2	B3	B4	B5	B6
280H+NodeId	*2007.0H*				*6000.1H*	*6404.1H*	
	Encoder_Actual_Value[0]				*B0_Digital_Inputs*	*Analog_In[0]*	

Static Mapping #9 (*2200.5h* obj , bit3÷bit0 = 9)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
280H+NodeId	*606C.0H*				*60F4.0H*			
	Velocity_Actual_Value				*Following_error_actual_value*			

Static Mapping #10 (*2200.5h* obj , bit3÷bit0 = 10)

Cob_ID	B0	B1	B2	B3	B5	B6
280H+NodeId	*6064.0H*				*6404.1H*	
	Position_Actual_Value				*Analog_In[0]*	

TX_PDO3 Static Mapping list :

Static Mapping #0 (2200.1Ah obj, bit11÷bit8 = 0)

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
380H+NodeId	*606C.0H*				*6064.0H*			
	Velocity_Actual_Value				*Position_Actual_Value*			

TX_PDO4 Static Mapping list :

Static Mapping #0 (2200.1Ah obj, bit3÷bit0 = 0)

Cob_ID	B0	B1
480H+NodeId	*603F.0H*	
	Error Code	

2.5 CANopen SYNC (Synchronization Message)

The Cob-Id is fixed to 80h (CiA DS301 default value) and cannot be change by means of *1005.0H* obj (Cob-Id SYNC). The SYNC message is useful when it is necessary to retrieve PDOs from the drive only when requested by the master (SYNC producer) or to makes PDOs sent by the master processed at the same time by the drives.

Cob_ID
80H

2.6 CANopen Heartbeat

The Titanio-Platino-Vanadio family drives implement the heartbeat protocol as defined in CiA DS 301 V4.01. This permits to the Master to check the drive working condition. It is possible to change the frequency of heartbeat transmission with the *1017.0H* obj (Producer Heartbeat Time). At switch-on the drive send the Boot-up message that is a heartbeat message with Status = 0;

Cob_ID	B0
700H+NodeId	Status

The Titanio-Platino-Vanadio family drives support only Pre-Operational (127) and Operational (5) states.

2.6.1 CANopen Nodeguarding

The nodeguarding protocol is implemented as defined in CiA DS 301 V4.01. Nodeguarding cannot be enabled together with Heartbeat (*1017.0H*). To enable the nodeguarding, the *1017.0H* must be set to 0 and *100C.0H* and *100D.0H* must be set different than 0. The bit #7 of the status act a toggle and change at each remote request by the master.

Cob_ID	B0
700H+NodeId	Status

2.7 CANopen Emergency Telegram (EMCY)

The Titanio-Platino-Vanadio drives send an Emergency Telegram every time a fault (software or hardware) is detected and send also an Emergency Telegram at switch on without any data bytes (only Cob_Id).

Cob_ID	B0	B1	B2	B3	B4	B5	B6	B7
80H+NodeId	*Error Code*		*Error register*	*Manufacturer Specific Error Register*				

Error Code : standard CiA error code (object *603F.0H*)
Error Register : standard CiA error register (object *1001.0H*)
Manufacturer Specific Error Register : (object *1002.0H*)

Note: The EtherCAT version of drives supports either EMCY or Diagnostics.

2.7.1 Emergency Codes

Error Code	Error Register	Manufacturer Specific Error Register	Fault Type	Description	Note
FF01h	81h	01h	Non-Fatal	Division by zero	(3) (4)
FF08h	81h	08h	Non-Fatal	Value Out of range	(3) (4)
FF15h	81h	15h	Fatal	Thermal protection	(1) (3)
FF16h	81h	16h	Fatal	Voltage Protection	(1) (3)
FF17h	81h	17h	Fatal	Current Protection	(1) (2)
FF1Ah	81h	1Ah	Fatal	Watch dog	(1) (2)
FF28h	81h	28h	Fatal	Missing Calibration	(1) (2)
FF33h	81h	33h	Fatal	Open Phase : verify motor connections	(1) (2)
FF4Ah	81h	4Ah	Fatal	Internal Software	(1) (2)
FF59h	81h	59h	Fatal	EEprom Fail	(1) (2)
FF5Bh	81h	5Bh	Fatal	Motor current regulation out of range	(1) (3)
FF5Fh	81h	5Fh	Fatal	Feedback Error (closed loop)	(1) (3)
FF62h	81h	62h	Fatal	EEprom write overrun	(1)
FF63h	81h	63h	-	Reserved	
FF64h	81h	64h	Non-Fatal	Sync not received (ip / csp / csv / cst modes) – Abort communication	(3) (4) (5)
FF65h	81h	65h	Non-Fatal	Forward Limit switch reached	(3) (4)
FF66h	81h	66h	Non-Fatal	Backward Limit switch reached	(3) (4)
8130h	11h	67h	Non-Fatal	Guarding Fail	(3) (4) (5)
FF68	81h	68h	Fatal	Missing Torque Enable	(1) (3)
FF69	81h	69h	Fatal	I2T Protection	(1) (3)
FF6A	81h	6Ah	Non-Fatal	Heartbeat Monitoring Fail	(3) (4) (5)
FF7F	81h	7Fh	Non-Fatal	Bus Off	(3) (4) (5)

Note :

- [(1)] The drive device is not longer able to control the motor, so that an immediate switch-off of the motor is necessary.
- [(2)] To reset the fault condition is necessary to reset the drive by mean of 'Reset Node command' or switch off the drive. Contact 'Ever company' if is not possible to reset the fault condition. Before trying to reset the fault, make sure that the condition that caused the fault has been resolved. It is not possible to reset the fault condition by mean of Fault reset transtion #15 (See *§3.1.1*).
- [(3)] The fault is reset by mean of 'Fault' reset transition #15 (See *§3.1.1*).
- [(4)] See *605E.0H* object.
- [(5)] See *6007.0H* object.

2.8 CANopen Boot Up / NMT Protocols

At switch-on the Titanio-Platino-Vanadio drives are in Pre-Operational state, this means that PDOs are disabled. At switch-on the drive send the Boot-up message that is a heartbeat message with Status = 0.

The boot-up process could take 6-7 seconds.

Cob_ID	B0
700H+NodeId	0

The Master have to send the NMT frame with command Start Node.

Start Node Command

Cob_ID	B0	B1
00H	1	NodeId

Enter Pre-Operational State Command

Cob_ID	B0	B1
00H	128	NodeId

If NodeId = 0 all devices connected to CAN network will execute the command.

It is supported also the NMT - Reset Node Protocol to reset the drive:

Reset Node Command

Cob_ID	B0	B1
00H	129	NodeId

3.0 CANopen profile DS402

The purpose of the profile is to give drives an understandable and unique behavior on the CAN bus network.
The two principal advantages of the profile approach for device specification are in the areas of system integration and device standardization. A device profile defines a 'standard' device. This standard device represents really basic functionality, every device within this device class must support. This mandatory functionality is necessary to ensure that at least simple non-manufacturer-specific operation of a device is possible. The concept of device standardization is extended by the notion of *optional* functionality defined within the standardized device profile. Such optional functionality does not have to be implemented by all manufactures. However, if a manufacturer implements such optional functionality , he must do so in a fixed manner. Providing optional functionality is a very powerful mechanism to ensure all manufacturers implementing particular functionality in a defined fashion. The device profiles provide a mechanism by which manufacturers wishing to implement manufacturer specific functionality can do so as well. This is necessary since it would be impossible to anticipate all possible device functionality and define this in the optional category of each device class. By defining mandatory device features, basic network operation is guaranteed. By defining optional device features, a degree of defined flexibility can be built in. By leaving hooks for manufacturer specific functionality, manufacturers will not be constrained to an out-of-date standard.

3.1 Device Control

The device control function block controls all functions of the drive. It is divided into :

- **State machine**
- **Operation mode**

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. States may be changed by the *Controlword* (*6040.0H*) and/or according to internal events. The current state is shown in the *Statusword* (*6041.0H*).

The **Operation mode** defines the behavior of the drive. The drive functions depend from the select *Modes of operation* (*6060.0H*). The specific drive function is executed only when the drive status is **Operation Enabled**.
The following modes of operation are defined in the 'Titanio-Platino-Vanadio' drives :

- **Interpolated Position Mode (ip)**
- **Profile Position Mode (pp)**
- **Homing Mode (hm)**
- **Profile Velocity Mode (pv)**
- **Cyclic Synchronous Position Mode (csp)**
- **Cyclic Synchronous Velocity mode (csv)**
- **Velocity Mode (vl)**
- **Profile Torque Mode (tq)**
- **Cyclic Synchronous Torque mode (cst)**
- **Feedback Sensor Calibration mode (fsc)**

3.1.1 State machine

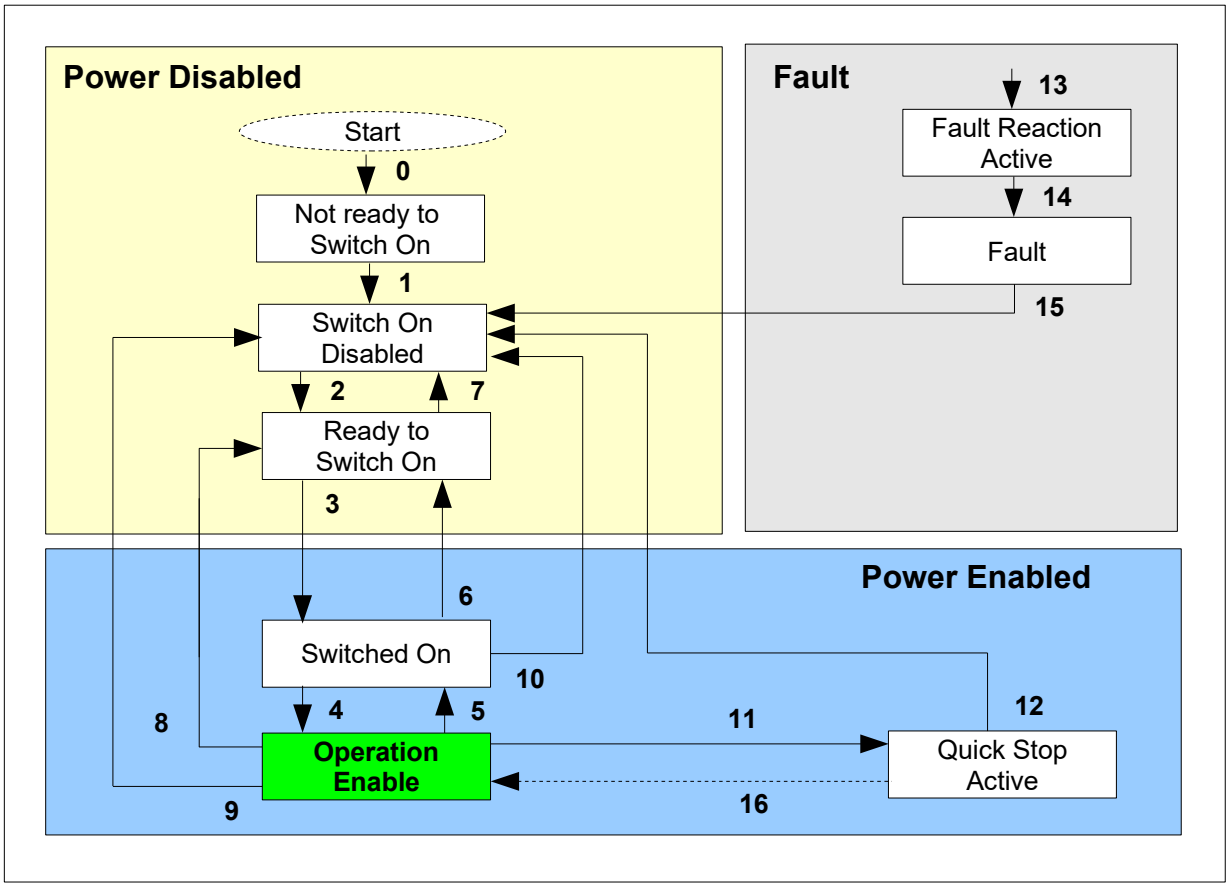


Figure 1 : Device Control State Machine

3.1.2 Drive states

State	StatusWord															
	15	14	13	12	11	0	9	8	7	6	5	4	3	2	1	0
Not Ready to Switch On	X	X	X	X	X	X	X	X	X	0	X	X	0	0	0	0
Switch On Disabled	X	X	X	X	X	X	X	X	X	1	X	X	0	0	0	0
Ready to Switch On	X	X	X	X	X	X	X	X	X	0	1	X	0	0	0	1
Switched On	X	X	X	X	X	X	X	X	X	0	1	X	0	0	1	1
Operation Enable	X	X	X	X	X	X	X	X	X	0	1	X	0	1	1	1
Quick Stop Active	X	X	X	X	X	X	X	X	X	0	0	X	0	1	1	1
Fault Reaction Active	X	X	X	X	X	X	X	X	X	0	X	X	1	1	1	1
Fault	X	X	X	X	X	X	X	X	X	0	X	X	1	0	0	0

• Not Ready to Switch On :
– The drive is being initialized
– The drive function is disabled (no energy is supplied to the motor)
– The drive is not ready to accept commands

• Switch On Disabled :
– Drive initialization is complete
– The drive parameters have been set up
– Drive parameters may be changed
– The drive function is disabled (no energy is supplied to the motor)

• Ready To Switch On :
– The drive parameters may be changed
– The drive function is disabled (no energy is supplied to the motor)

• Switched On :
– The power amplifier is ready
– The drive parameters may be changed
– The drive function is disabled (no energy is supplied to the motor)

• 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

• 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

• Fault Reaction Active(*) :
– The drive parameters may be changed
– A fault has occurred in the drive
– 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
– The drive function is disabled (no energy is supplied to the motor)

Note:
(*) Drive faults may be fatal or non-fatal. When a fatal fault occurs, the drive device is not longer able to control the motor, so that an immediate switch-off of the motor is necessary.

3.1.3 State Transitions

State transitions are caused by internal events in the drive or by commands from the host with the 'Controlword'.

Transition	EVENT		ACTION
	Command from the host	*Internal*	
0		Reset.	The drive self-initializes.
1		The drive has initialized successfully.	Activate communication.
2	'Shutdown'		None.
3	'Switch On'		The power section is switched on.
4	'Enable Operation'		The drive function is enabled (power is applied to the motor).
5	'Disable Operation'		The drive operation will be disabled (no energy will be supplied to the motor). See object 605Ch.
6	'Shutdown'		The drive function is disabled (power is not applied to the motor).
7	'Quick Stop' or 'Disable Voltage'		None.
8	'Shutdown'		The power section is switched off immediately. Power is not applied to the motor. The motor is free to rotate if unbraked. See object 605Bh.
9	'Disable Voltage'		The power section is switched off immediately. Power is not applied to the motor. The motor is free to rotate if unbraked.
10	'Disable Voltage' or 'Quick Stop'		The power section is switched off immediately. Power is not applied to the motor. The motor is free to rotate if unbraked.
11	'Quick Stop'		The quick stop function is executed.
12	'Disable Voltage'	'Quick Stop' action is completed .	The power section is switched off immediately. Power is not applied to the motor. The motor is free to rotate if unbraked.
13		A fault has occurred in the drive.	The drive executes appropriate fault reaction (see 605E.0 h object).
14		The fault reaction is completed.	The drive function is disabled (power is not applied to the motor).
15	'Fault Reset'		The fault condition is reset if no fault currently exists in the drive. After leaving Fault state, the Fault Reset bit of the Controlword must be cleared by the host.
16	'Enable Operation'		The drive function is enabled (power is applied to the motor).

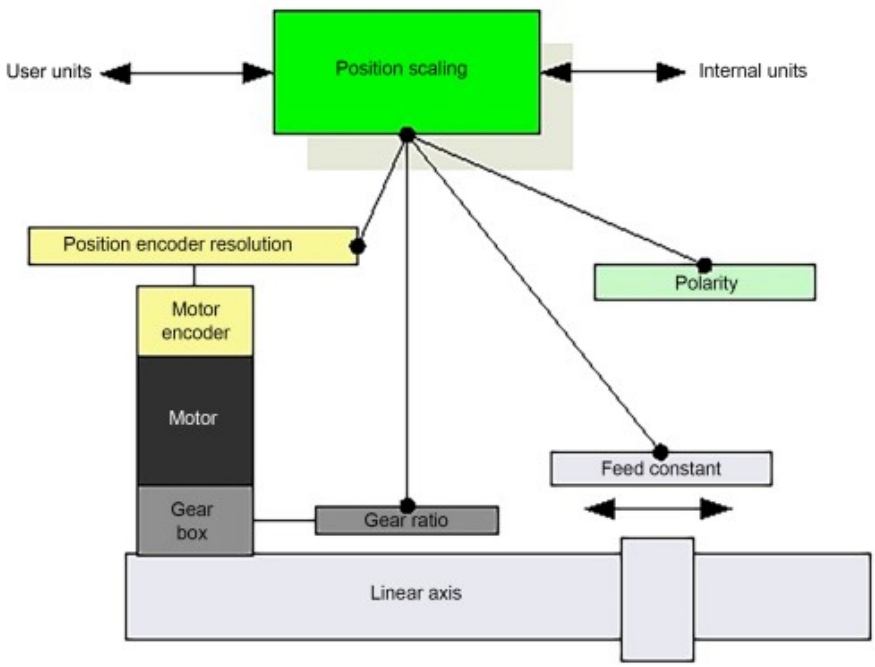
(Table 2 : State transitions)

Listed in the following table are the bit combination for the *Controlword* that result in the corresponding state transitions. An 'X' corresponds to a bit state that requires no futher consideration. The only exception is the resetting of the error (fault reset): the transition is only requested by the rising edge of the bit.

Command	Transition	ControlWord															
		15	14	13	12	11	0	9	8	7	6	5	4	3	2	1	0
Shutdown	[2][6][8]	X	X	X	X	X	X	X	X	X	X	X	X	X	**1**	**1**	**0**
Switch On	[3]	X	X	X	X	X	X	X	X	X	X	X	X	X	**1**	**1**	**1**
Disable Voltage	[7][9][10][12]	X	X	X	X	X	X	X	X	X	X	X	X	X	X	**0**	X
Quick Stop	[7][10][11]	X	X	X	X	X	X	X	X	X	X	X	X	X	**0**	**1**	X
Disable Operation	[5]	X	X	X	X	X	X	X	X	X	X	X	X	**0**	**1**	**1**	**1**
Enable Operation	[4][16]	X	X	X	X	X	X	X	X	X	X	X	X	**1**	**1**	**1**	**1**
Fault Reset	[15]	X	X	X	X	X	X	X	X	↑	X	X	X	X	X	X	X

3.2 Factor group

The factors defined in the factor group set up a relationship between 'device internal units' and 'user-defined units'.
The 'user-defined units' are used in the corresponding objects representing position, velocity and acceleration values.



The 'device internal units' are always related to *Motor_Resolution* and defined as :

'Position Internal unit' → Increments [Inc]
'Velocity Internal unit' → Increments/sec [Inc/s]
'Acceleration Internal unit' → Increments/sec² [Inc/s²]

The Factor group is available with firmware V00R80 or superior.
With firmware V00r79 or lower are considered only the 'device internal units'.

The unit conversion for the Position, Velocity and Acceleration values is done by the following formulas :

$$Position\ value\,(user\ defined\ unit) = \frac{Position\ value\,(device\ internal\ unit^{(1)}) * Feed\ constant}{65536\ inc/rev * Gear\ Ratio}$$

$$Velocity\ value\,(user\ defined\ unit) = \frac{Velocity\ value\,(device\ internal\ unit^{(2)})}{Velocity\ Factor}$$

$$Acceleration\ value\,(user\ defined\ unit) = \frac{Acceleration\ value\,(device\ internal\ unit^{(3)})}{Acceleration\ Factor}$$

$$Feed\ constant = \frac{6092.1\,h}{6092.2\,h}$$

$$Gear\ Ratio = \frac{6091.1\,h}{6091.2\,h}$$

$$Velocity\ Factor = \frac{2013.1\,h}{2013.2\,h}$$

$$Acceleration\ Factor = \frac{2013.3\,h}{2013.4\,h}$$

Notes :

(1) The 'device internal unit' for Position is defined in ***Increments***.
The Motor Resolution used for Position conversion is 65536 Inc/rev (*2012.1H* , *2012.2H* , *60EF.0H* objects are not considered for conversion).

(2) The 'device internal unit' for Velocity is defined in ***Increments/sec***.
The Motor Resolution used for Velocity conversion is defined by mean of *2012.1H* , *2012.2H* , *60EF.0H* objects.

(3) The 'device internal unit' for Acceleration is defined in ***Increments/sec²***.
The Motor Resolution used for Acceleration conversion is defined by mean of *2012.1H* , *2012.2H* , *60EF.0H* objects.

- If 'Sensor_selection_code' (*606A.0H*) object is equal to 0 or -2 then Encoder Resolution must be also defined (608F.1H and *608F.2H*).

- If 'Sensor_selection_code' (*606A.0H*) object is equal to -3 then BiSS Encoder configuration must be also defined (2A04.0H). The Encoder Resolution is always normalized to 65536 Inc/rev and then it is not considered parameters defined by mean of 608F.1H and *608F.2H* objects.

- If 'Sensor_selection_code' (*606A.0H*) object is equal to -1 and *Closed Loop* is enabled then Encoder Resolution must be also defined. If Sensor Type used is an Incremental Encoder the Encoder resolution is defined by mean of 608F.1H and 608F.2H objects. If sensor Type used is an Absolute BiSS Encoder the Encoder Resolution is always normalized to 65536 Inc/rev (608F.1H and 608F.2H objects are not rilevant) and BiSS encoder configuration must be defined (2A04.0H).
- The Factor Group are not used for conversion of 'Velocity Mode' objects (§3.10). For this mode the 'Velocity internal unit' is rpm (revolutions/min) and 'vl dimension factor ' objects are used for conversion.

Factor Group related objects :

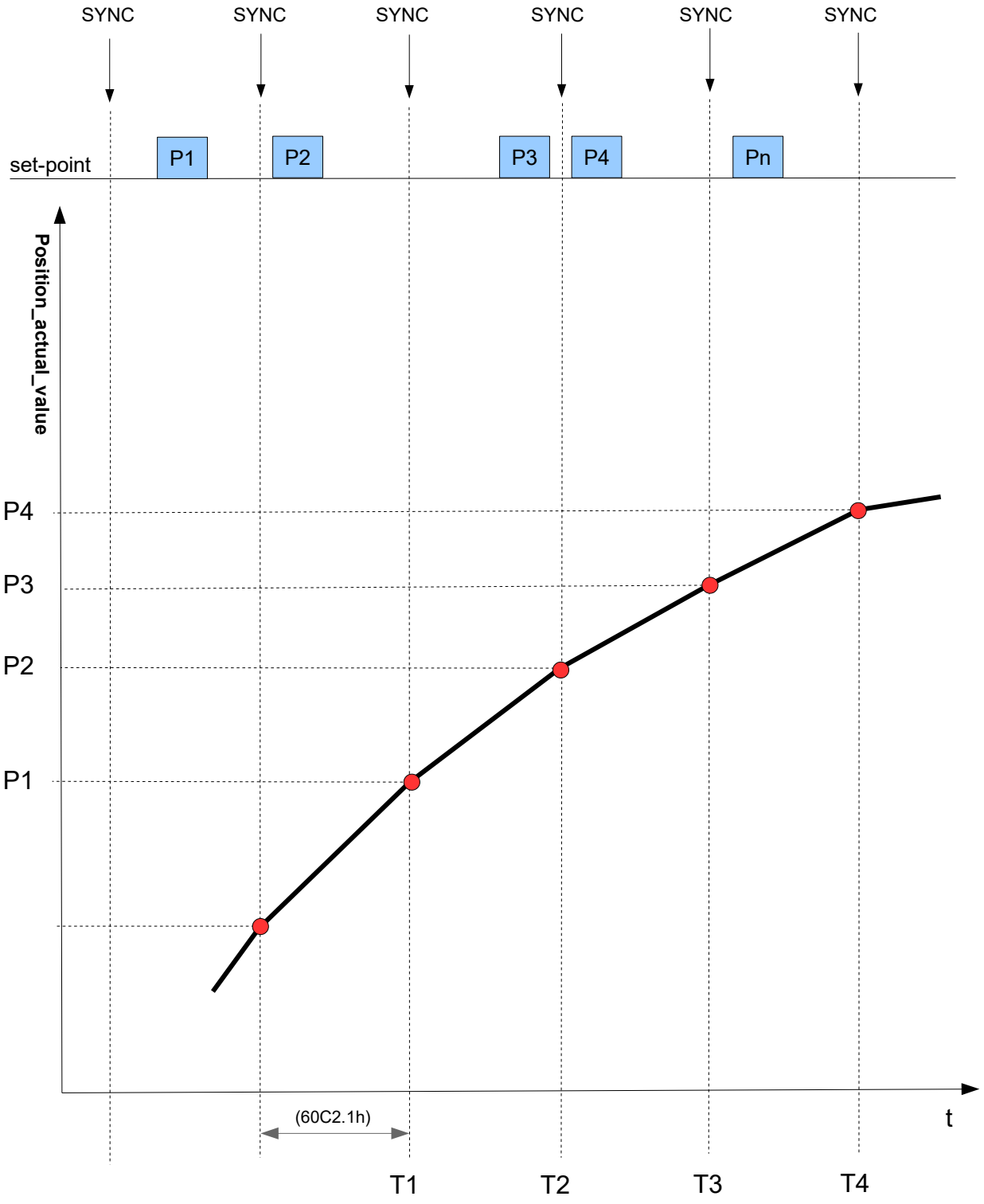
Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
2012H	RECORD	Motor Parameters		rw	M
2013H	RECORD	Motor Factor (1)	UNSIGNED32	rw	O
608FH	ARRAY	Position Encoder Resolution	UNSIGNED32	rw	O
6091H	ARRAY	Gear Ratio (1)	UNSIGNED32	rw	O
6092H	ARRAY	Feed Constant (1)	UNSIGNED32	rw	O
60EF.0H	VAR	Motor Resolution (1)	UNSIGNED32	ro	O

Notes :

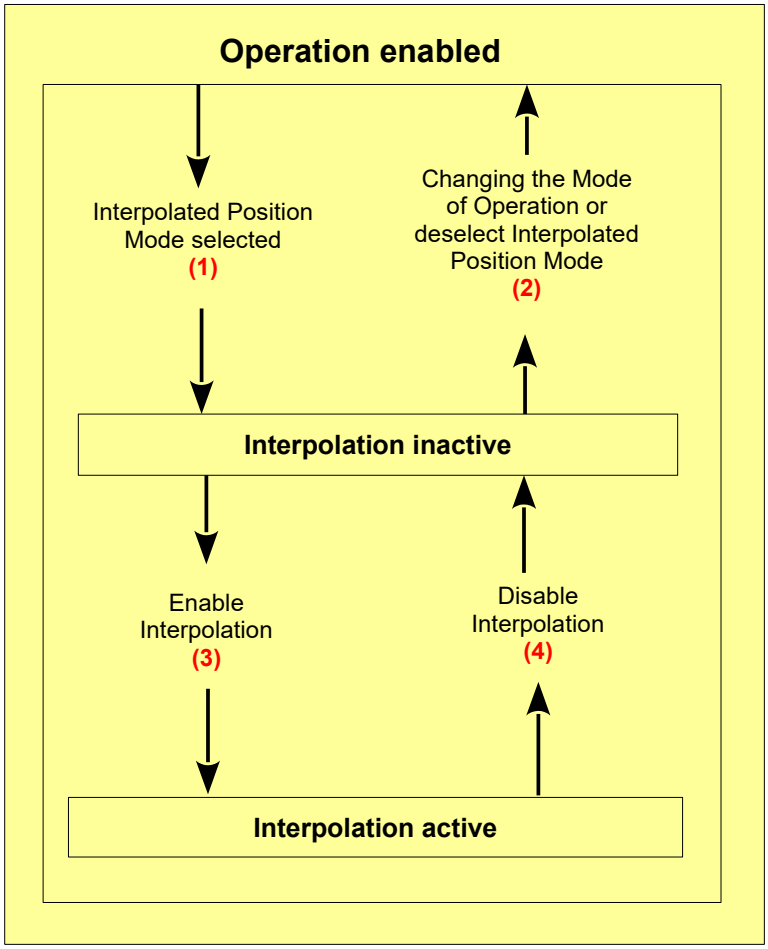
- (1) available with firmware V00R80 or superior.

3.3 Interpolated Position Mode (ip)

The interpolated Position mode is used to control multiple coordinated axles or a single axle with the need for time-interpolation of set set-point data. This modality uses the *SYNC* message to synchronize the interpolation points. The 'Titanio-Platino-Vanadio' Drives support **only the synchronous linear interpolation**, therefore the data structure used for interpolation has only one field and is referred to the position set-point (*60C1.1H*). To ensure proper operations the interpolation data should be supplied in real time via PDO service.



Internal States and State transitions



Interpolation inactive

This state is entered when the device is in state OPERATION ENABLED and the interpolated position mode is selected. The drive will accept input data but it does not move the axles.

Interpolation active

This state is entered when the device is in state OPERATION ENABLED and the interpolated position mode is selected and enabled. The drive will accept input data and it moves the axles.

State Transition (1)

NO IP-MODE SELECTED => IP-MODE INACTIVE
Event : Enter in the state OPERATIONAL ENABLE with *Controlword* and select ip mode with *modes of operation*

State Transition (2)

IP-MODE INACTIVE => NO IP-MODE SELECTED
Event : Leave the state OPERATION ENABLE with *Controlword* or select any other mode with *modes of operation* if it is allowed inside the state OPERATION ENABLE

State Transition (3)

IP-MODE INACTIVE => IP-MODE ACTIVE
Event : Set bit *enable ip mode* (bit4) of the *Controlword* while in ip mode and OPERATION ENABLED

State Transition (4)

IP-MODE ACTIVE => IP-MODE INACTIVE
Event : Reset bit *enable ip mode* (bit4) of the *Controlword* while in ip mode and OPERATION ENABLED

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
60C1.0H	ARRAY	Interpolation data record	INTEGER32	rw	O
60C2.0H	RECORD	Interpolation time period	Interpolation time period record	rw	O
60C3.0H	ARRAY	Interpolation sync definition	UNSIGNED8	rw	O

3.4 Profile Position Mode (pp)

A *Target_Position* (*607A.0H*) is applied to the trajectory generator; it generates a *Position_demand_value* (*6062.0H*) for the position control loop.

The setting of set-points is controlled by the *New_set_point* bit and the *Change_set_immediately* bit of the *Controlword* and the *Set_point_acknowledge* bit of the *Statusword*.

If the *Change_set_immediately* bit is set to 1, a ***Single set-point*** is expected by the device.
If the *Change_set_immediately* bit is set to 0, a ***Set of set-points*** is expected by the device.

After a set-point is sent to the slave device, the master device signals that the set-point is valid by a rising edge of the *New_set_point* bit of the *Controlword*. The slave device sets to 1 the *Set_point_acknowledge* bit of the *Statusword* to indicate that the new set-point has been received. Afterwards the slave device sets to 0 the *Set_point_acknowledge* bit when the device is ready to accept new set-points.

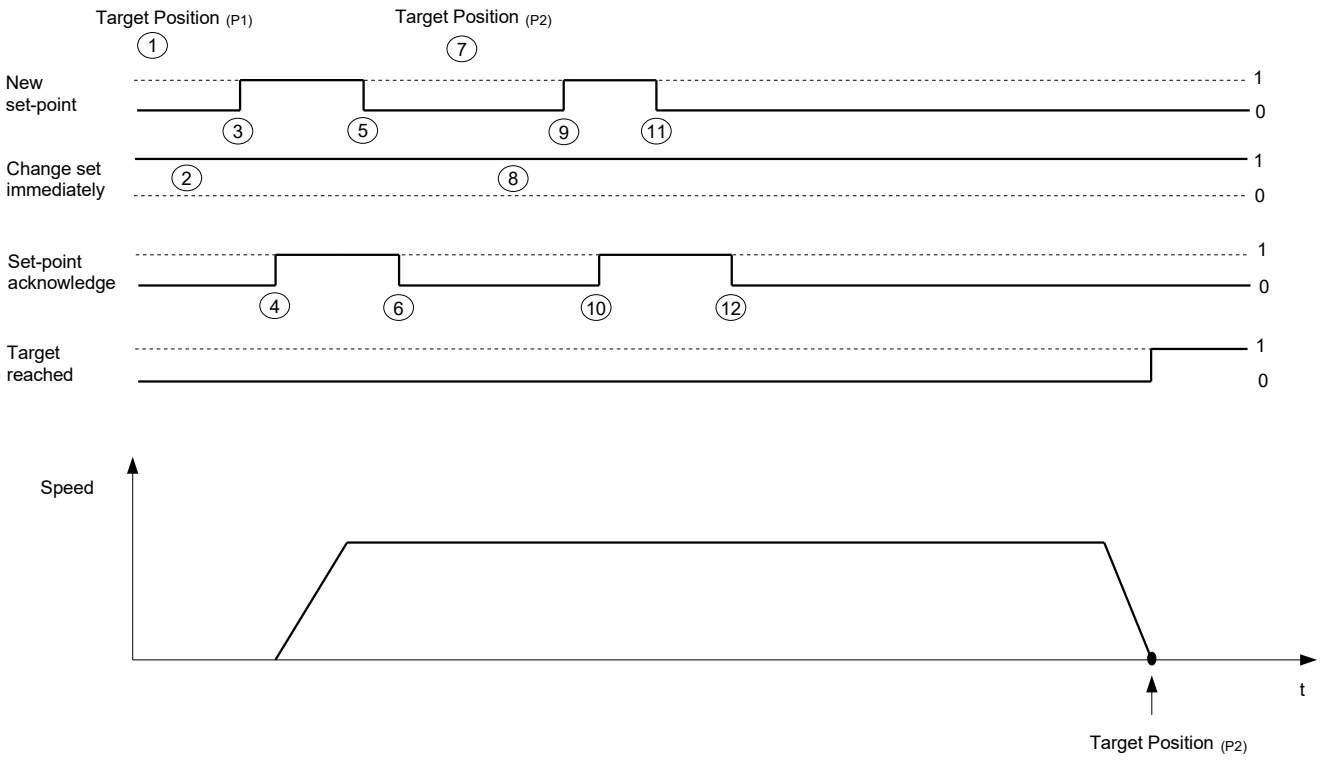
If one set-point is still in progress and new one is received from the slave device, two methods are supported :

Single set-point (*Change_set_immediately* = 1, bit #5 of *Controlword*)
The new set-point shall be processed immediately.

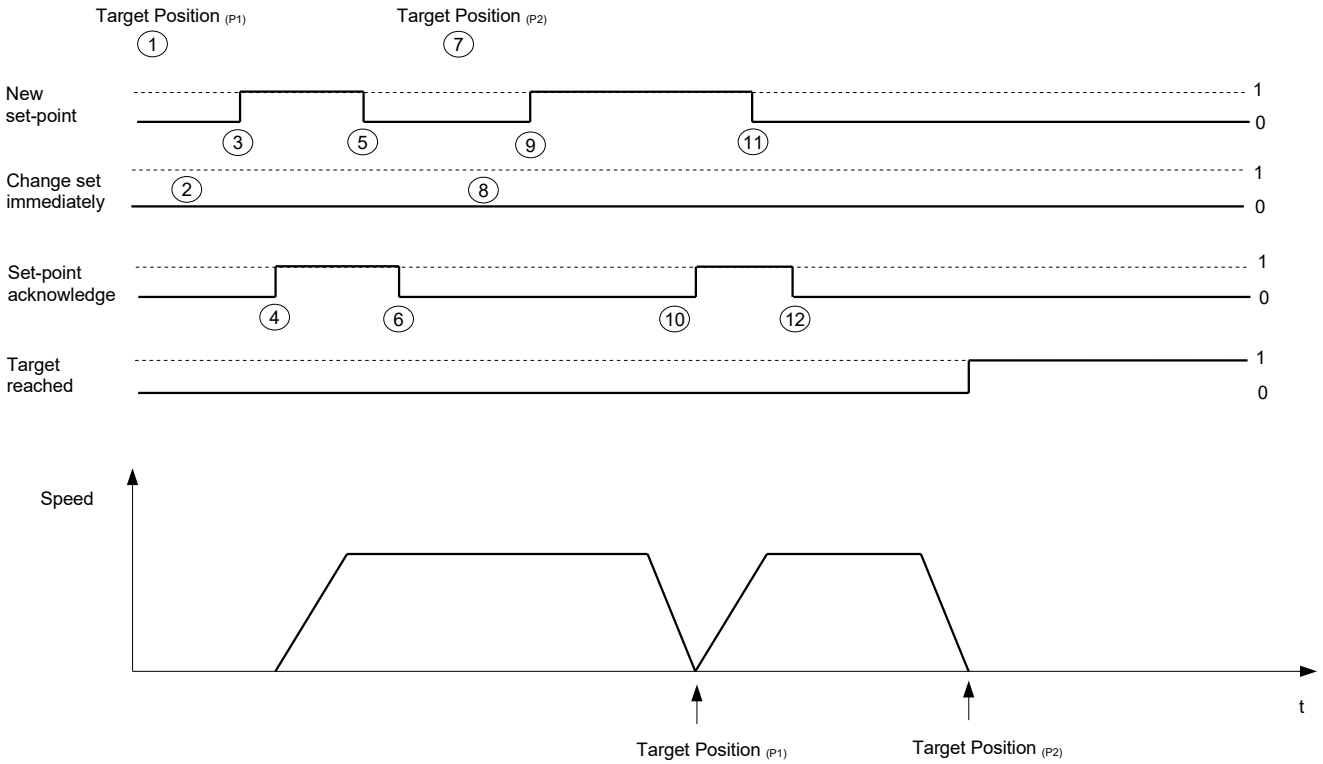
Set of set-points (*Change_set_immediately* = 0, bit #5 of *Controlword*)
The new set-point shall be processed only after the previous has been reached.
Up to two set-points can be set up. If all set-points available are busy (*Set_point_acknowledge* bit is 1) the reaction of the slave device depends on the *Change_set_immediately* bit. If it is set to 1, the new set-point shal be processed immediately as Single set-point.

The Target_reached bit of the *Statusword* shall remain to 0 until all set-points are processed.

The trajectory generator support only linear ramp (trapezoidal profile), with separate parameters for acceleration and deceleration.



(Single set-point)



(Set of set-points)

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6062.0H	VAR	Position demand value	INTEGER32	ro	O
6063.0H	VAR	Position actual value*	INTEGER32	ro	O
6064.0H	VAR	Position actual value	INTEGER32	ro	M
6065.0H	VAR	Following error window	UNSIGNED32	rw	O
6066.0H	VAR	Following error time out	UNSIGNED16	rw	O
6067.0H	VAR	Position window	UNSIGNED32	rw	O
6068.0H	VAR	Position window time	UNSIGNED16	rw	O
607A.0H	VAR	Target position	INTEGER32	rw	M
6081.0H	VAR	Profile velocity	UNSIGNED32	rw	M
6083.0H	VAR	Profile acceleration	UNSIGNED32	rw	M
6084.0H	VAR	Profile deceleration	UNSIGNED32	rw	O
6085.0H	VAR	Quick stop deceleration	UNSIGNED32	rw	O
6086.0H	VAR	Motion profile type	INTEGER16	rw	M
60F4.0H	VAR	Following error actual value	INTEGER32	ro	O
2010.0H	VAR	Min profile velocity	UNSIGNED16	rw	M

3.5 Homing mode (hm)

This method is used from a drive to seek the home position (also called, the datum reference point or zero point). There are various methods of achieving this, limit switches at the ends of travel or a home switch (zero point switch) in mid-travel, most of methods also use the index (zero) pulse train from an incremental encoder. The user can specify the speeds, acceleration and the method of homing.

It is possible to specify a *home offset* at the end of the seeking to displace zero in the coordinate system for the home position. The home offset is the difference between the zero position for the application and the machine home position. If the final home position must be a value different from zero, the user can define it by mean of *Preset_Homing_Position object (Homing_method_35) or Home_Offset object (Homing_method_37).*

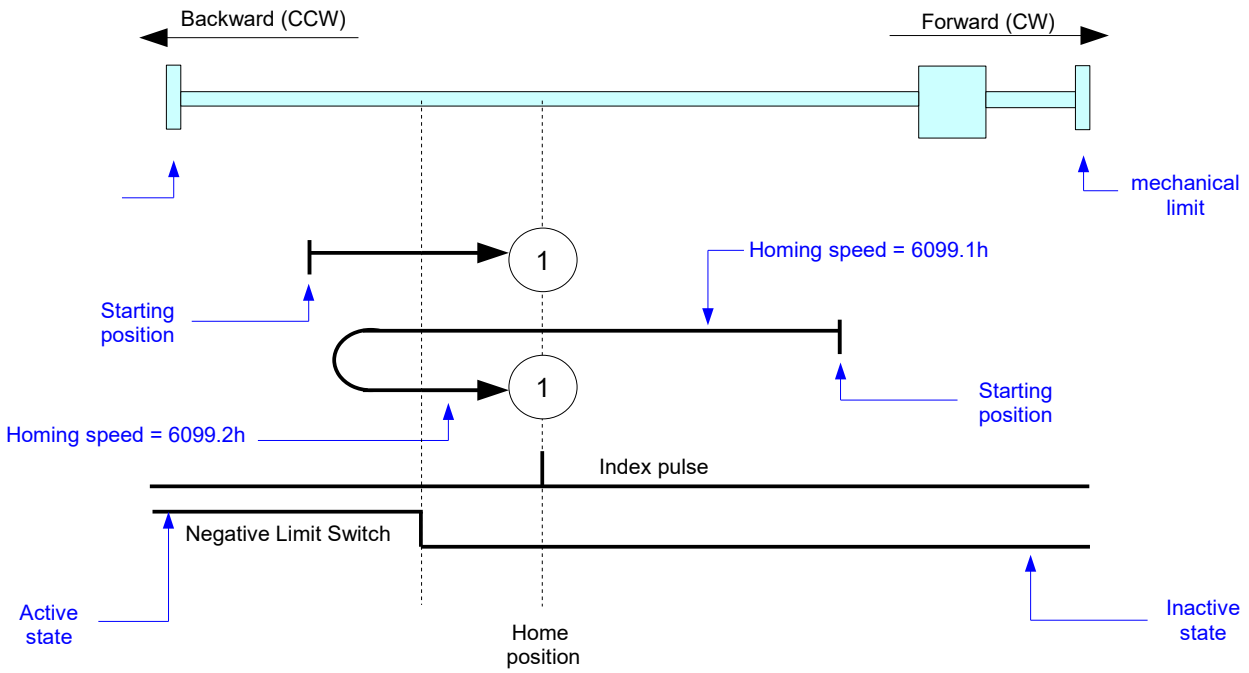
By choosing a method of homing, are defined the homing signal (positive limit switch, negative limit switch, home switch), the direction of actuation and where appropriate and the position of the index pulse. The sequence of the homing operation is described by the method.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
607C.0H	VAR	Home offset	INTEGER32	rw	O
6098.0H	VAR	Homing method	INTEGER8	rw	M
6099H	ARRAY	Homing speeds	UNSIGNED32	rw	M
609A.0H	VAR	Homing acceleration	UNSIGNED32	rw	O
6085.0H	VAR	Quick stop deceleration	UNSIGNED32	rw	O
2080.0H	VAR	Preset Homing Position	INTEGER32	rw	O
2081.0H	VAR	Drive Homing Inputs Setting	UNSIGNED32	rw	O

3.5.1 Homing method 1

The initial direction of movement is leftward if the *Negative Limit Switch* is inactive or rightward if the *Negative Limit Switch* is active. The home position is at the first *Index Pulse* to the right of the position where the *Negative Limit Switch* becomes inactive.



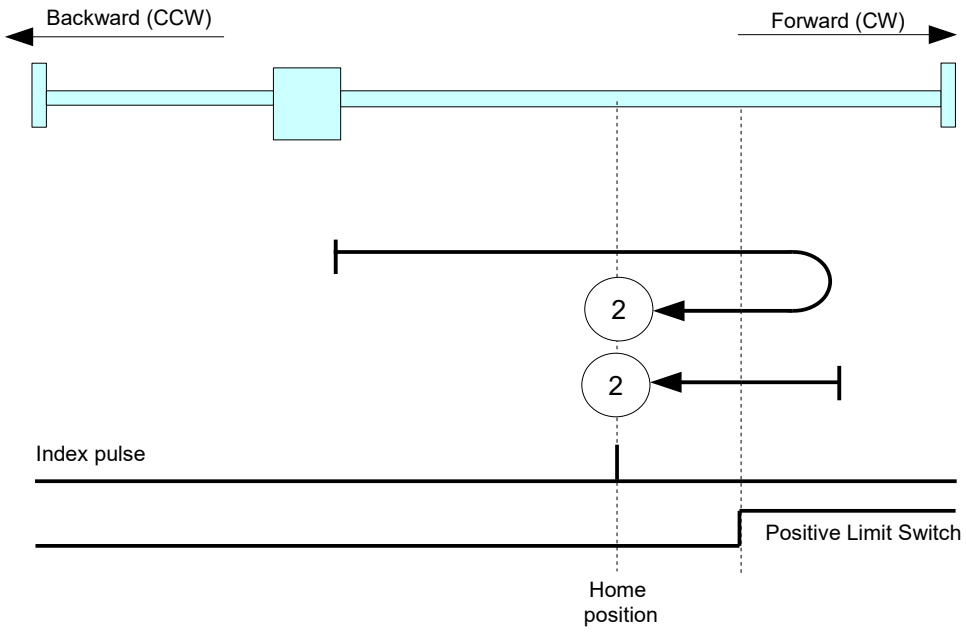
The Home Switch and Positive Limit Switch are not used.

Note:

- This Homing method is available with firmware version V00r69 or superior.

3.5.2 Homing method 2

The initial direction of movement is rightward if the *Positive Limit Switch* is inactive or leftward if the *Positive Limit Switch* is active. The home position is at the first *Index Pulse* to the left of the position where the *Positive Limit Switch* becomes inactive.



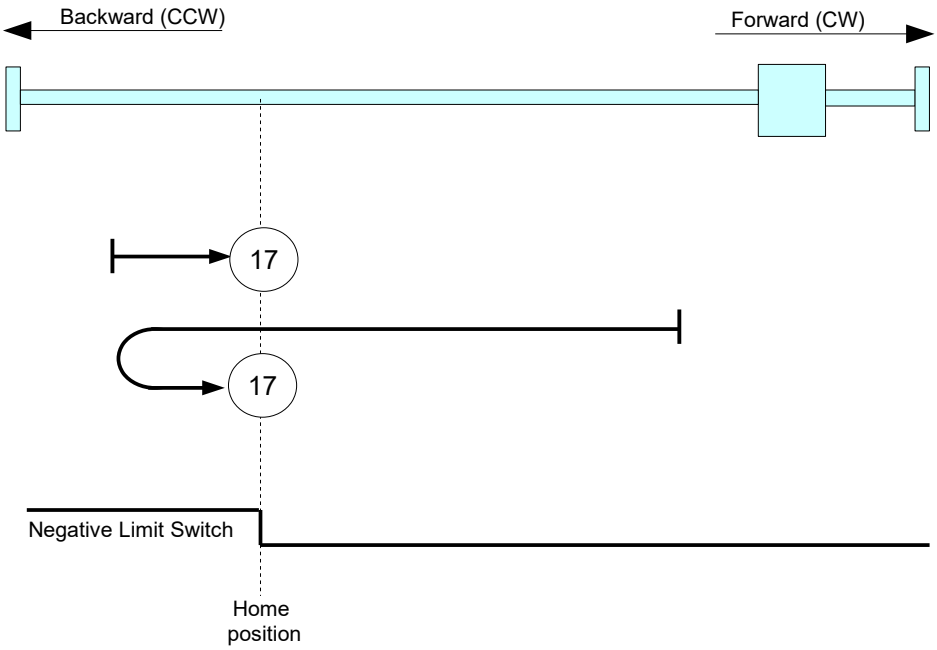
The Home switch and Negative Limit Switch are not used.

Note:

- This Homing method is available with firmware version V00r69 or superior.

3.5.3 Homing method 17

The initial direction of movement is leftward if the *Negative Limit Switch* is inactive or rightward if the *Negative Limit Switch*. The home position is on the transition from active to inactive state of *Negative Limit Switch*.



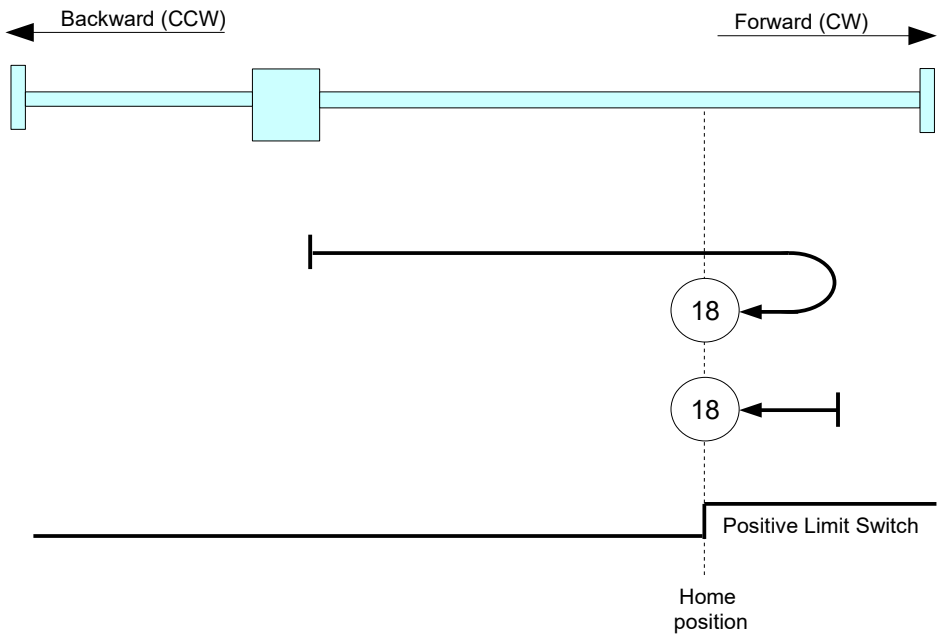
The Index pulse, Home Switch and Positive Limit switch are not used.

Note:

- This Homing method is available with firmware version V00r69 or superior.

3.5.4 Homing method 18

The initial direction of movement is rightward if the *Positive Limit Switch* is inactive or leftward if the *Positive Limit Switch* is active. The home position is on the transition from active to inactive state of *Positive Limit Switch*.



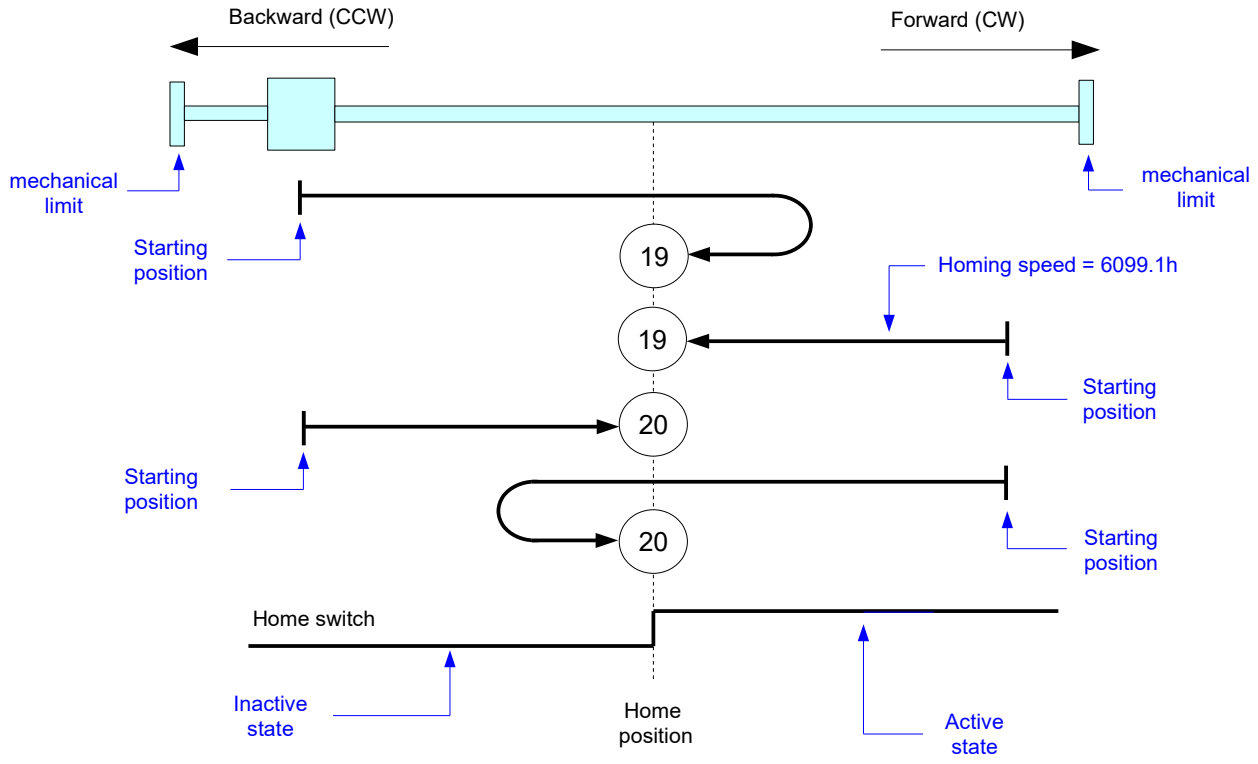
The Index pulse, Home switch and Negative Limit Switch are not used.

Note:

- This Homing method is available with firmware version V00r69 or superior.

3.5.5 Homing methods 19 and 20

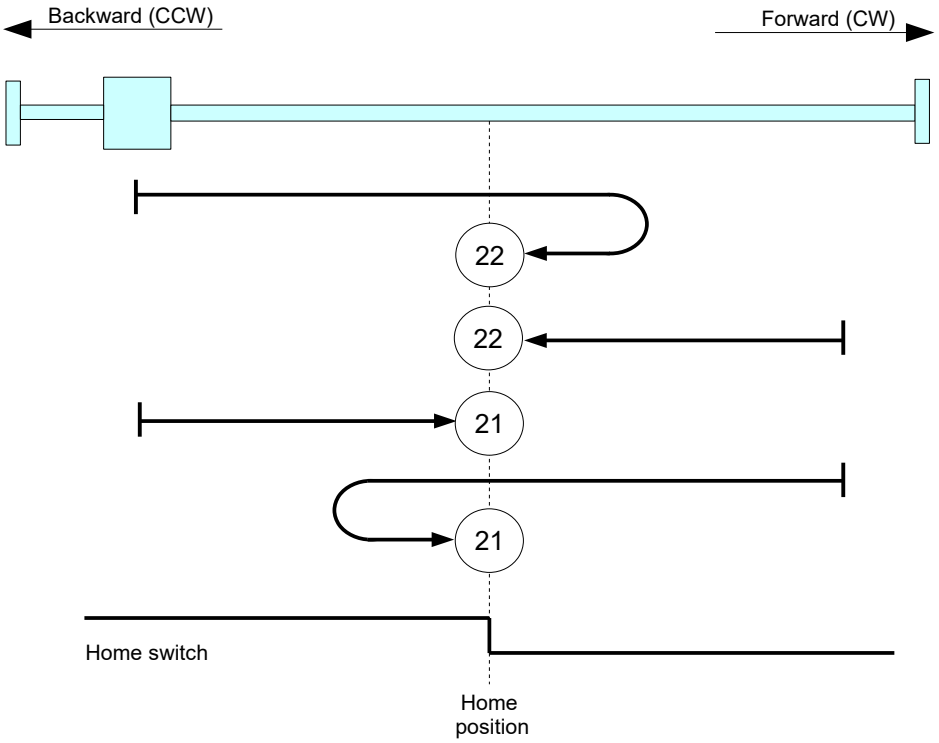
The initial direction of movement is dependent on the state of the *Home Switch*. The home position is on the point where the *Home Switch* changes its state. The point at which the reversal direction of movement takes place is anywhere after the change of state of the *Home Switch*.



The Index pulse, Negative Limit Switch and Positive Limit Switch are not used.

3.5.6 Homing methods 21 and 22

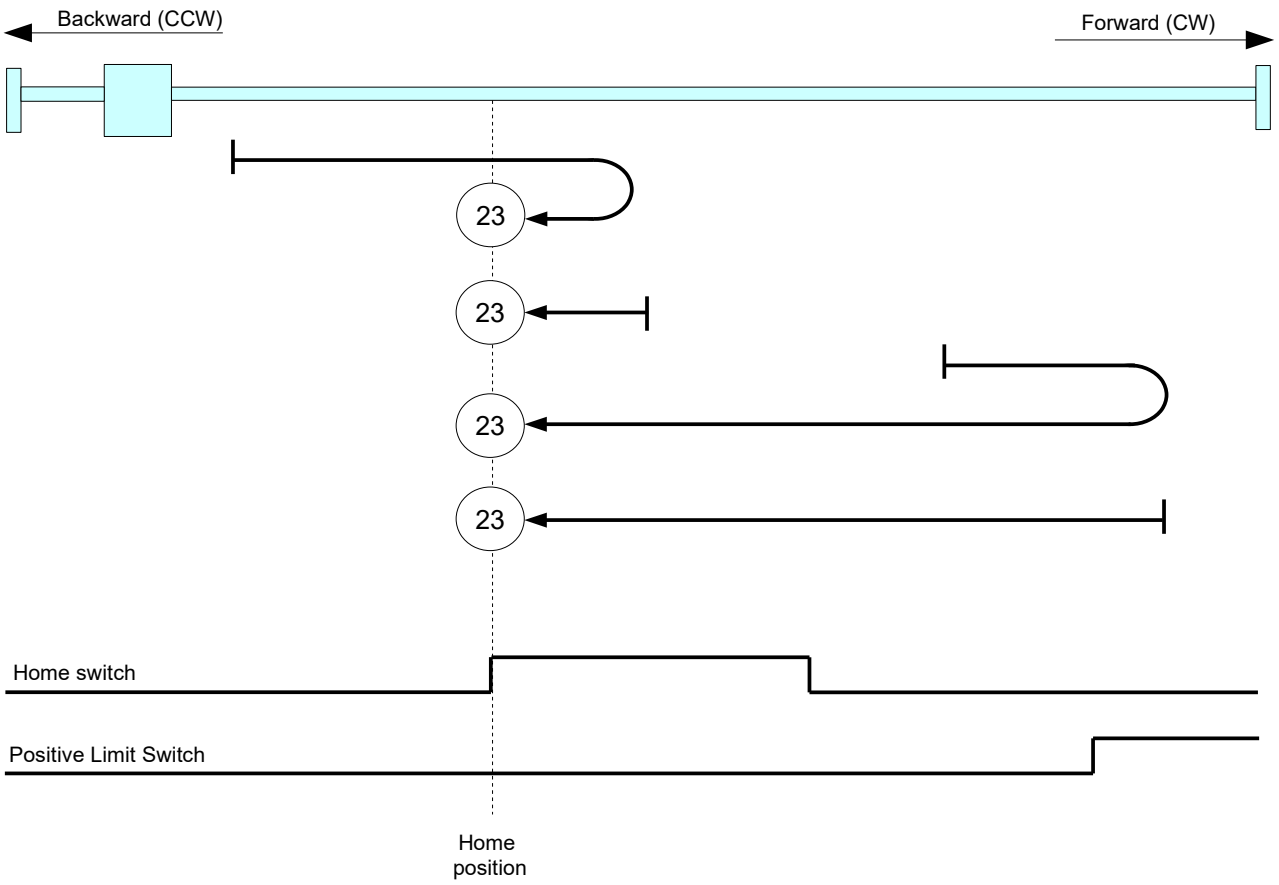
The initial direction of movement is dependent on the state of the *Home Switch*. The home position is on the point where the *Home Switch* changes its state. The point at which the reversal direction of movement takes place is anywhere after the change of state of the *Home Switch*.



The Index pulse, Negative Limit Switch and Positive Limit Switch are not used.

3.5.7 Homing method 23

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Positive Limit Switch*.



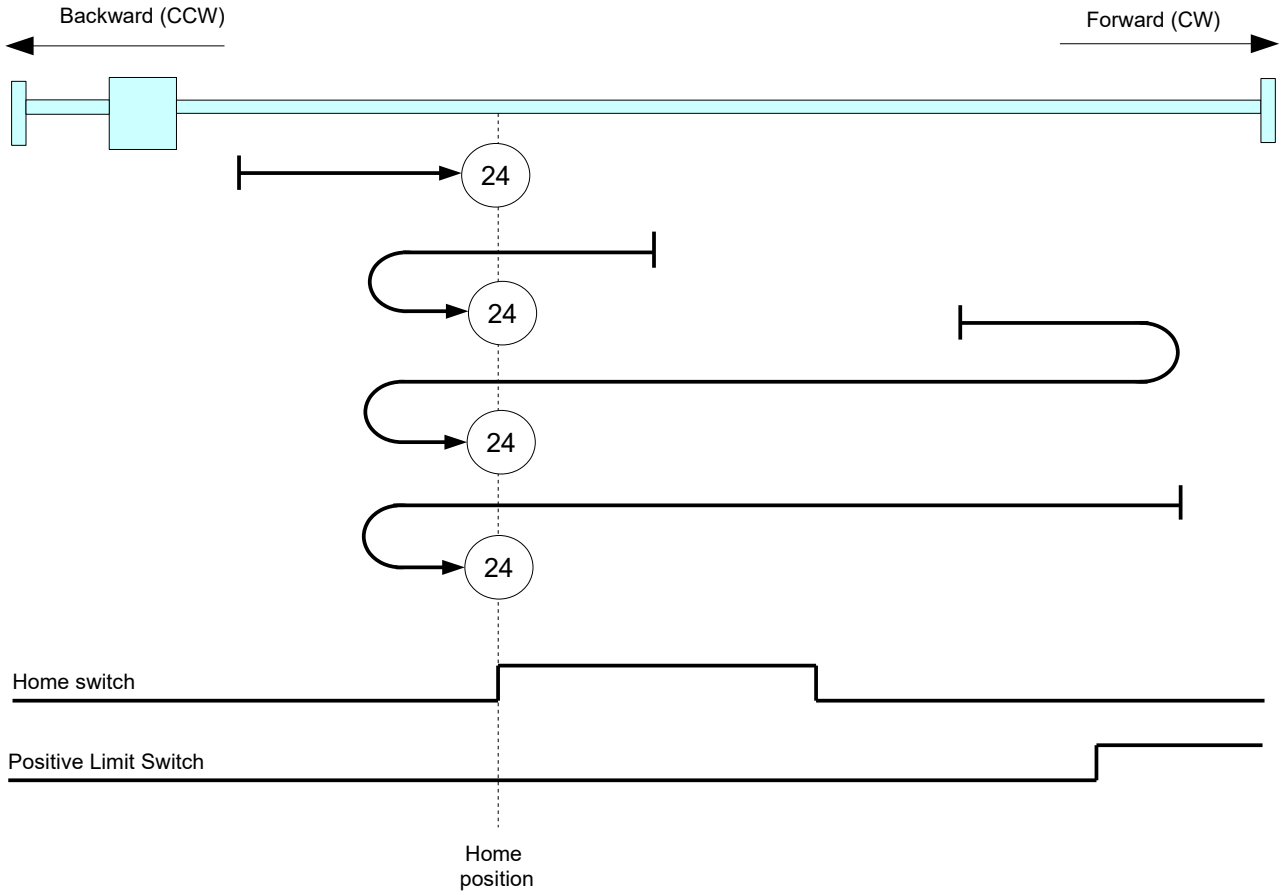
The Index pulse and Negative Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r86 or superior.

3.5.8 Homing method 24

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Positive Limit Switch*.



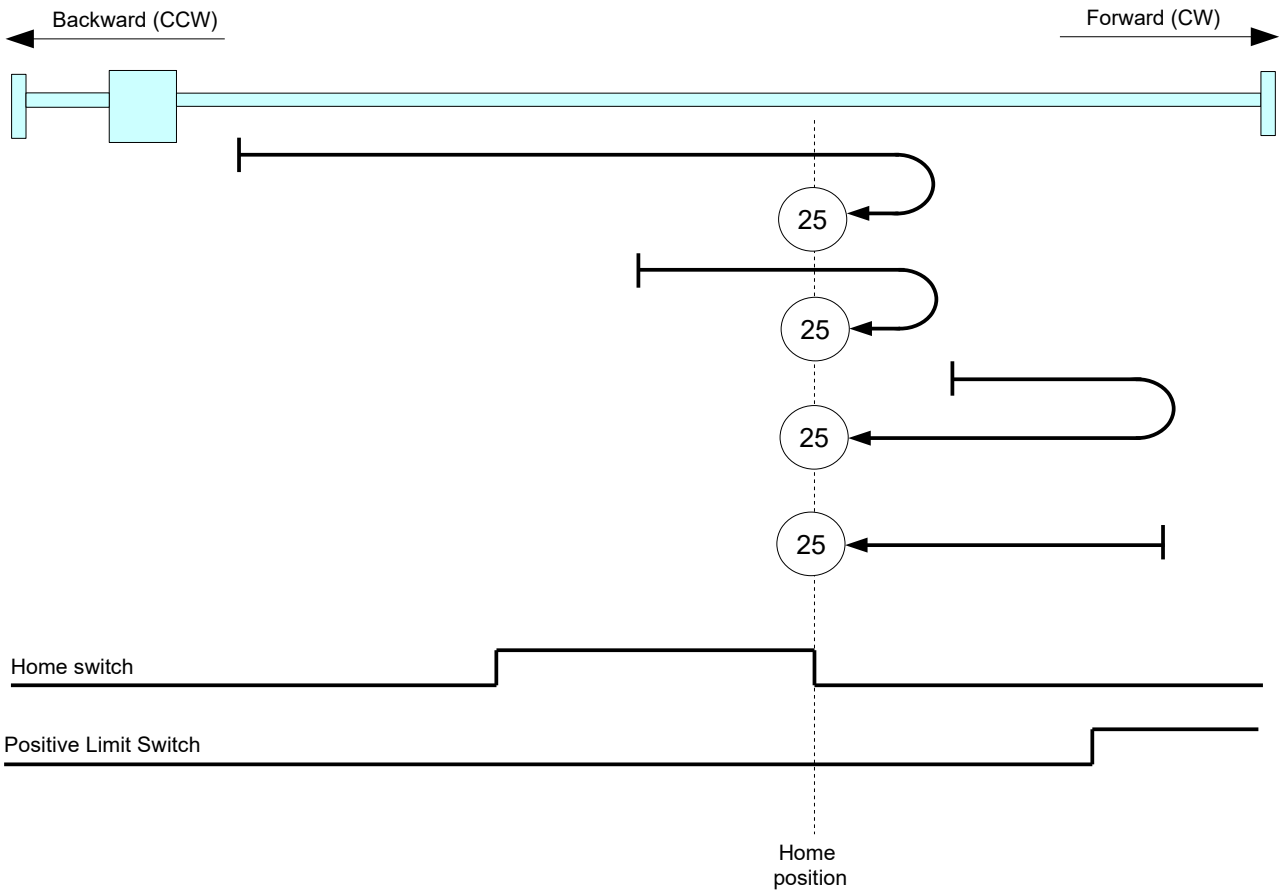
The Index pulse and Negative Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r86 or superior.

3.5.9 Homing method 25

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Positive Limit Switch*.



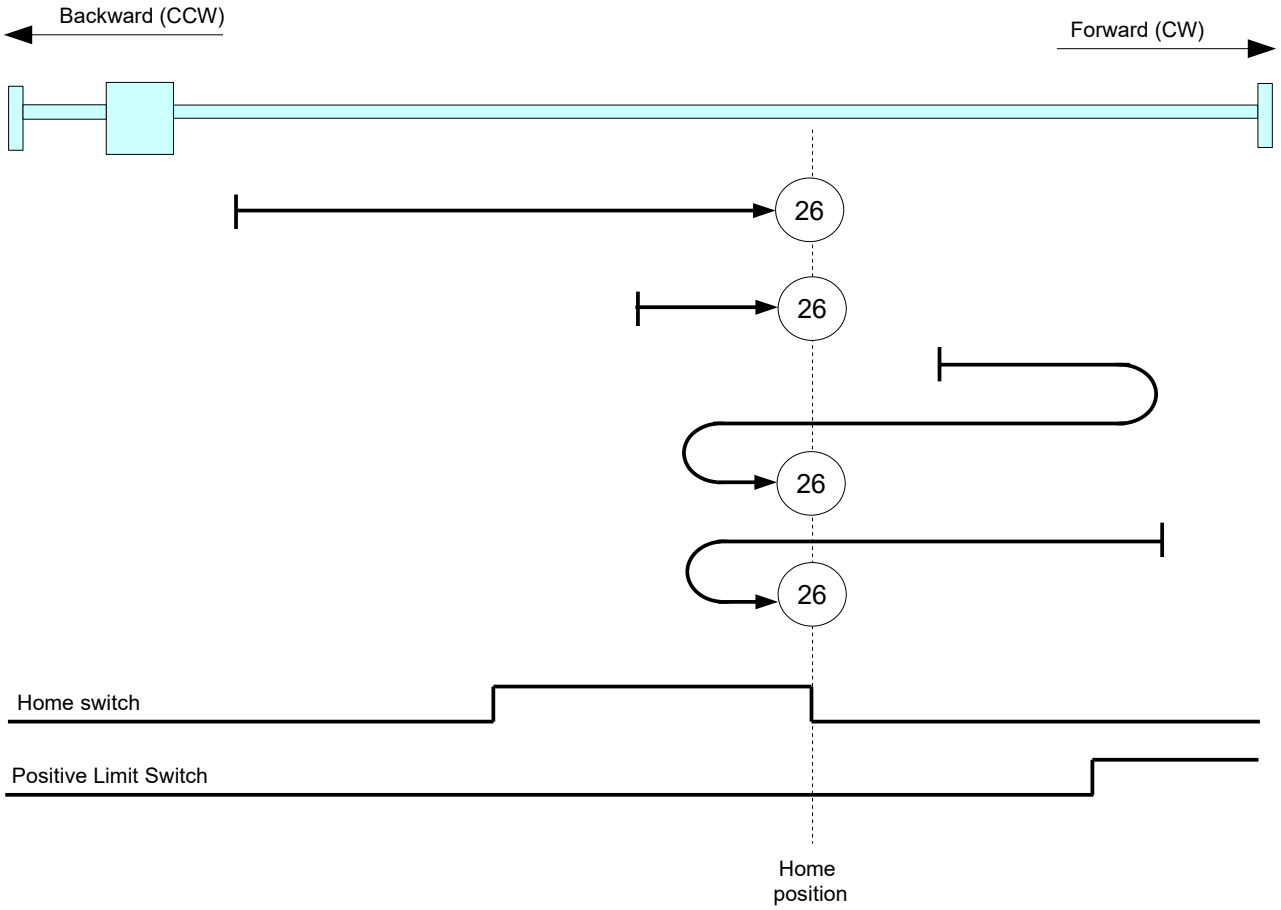
The Index pulse and Negative Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r86 or superior.

3.5.10 Homing method 26

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Positive Limit Switch*.



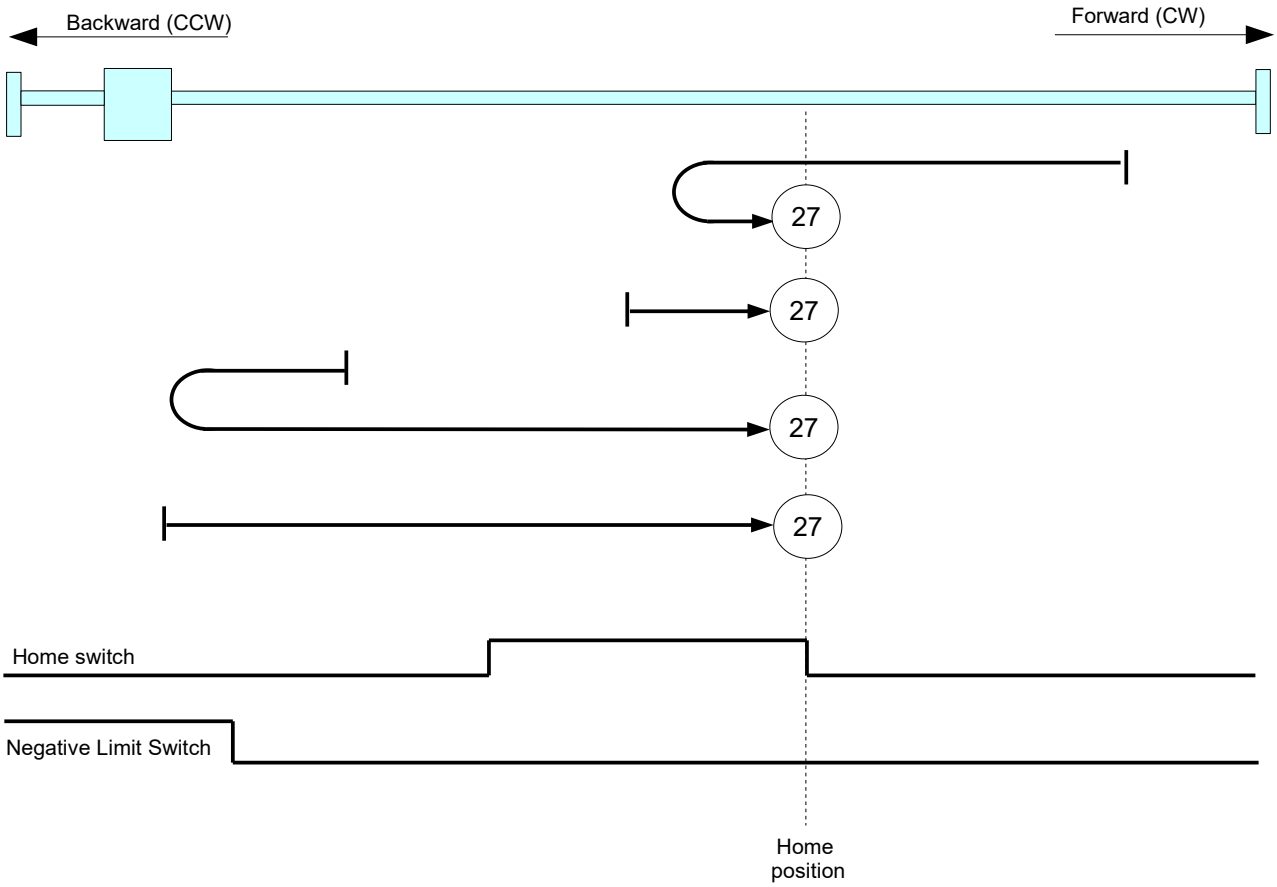
The Index pulse and Negative Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r86 or superior.

3.5.11 Homing method 27

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Negative Limit Switch*.



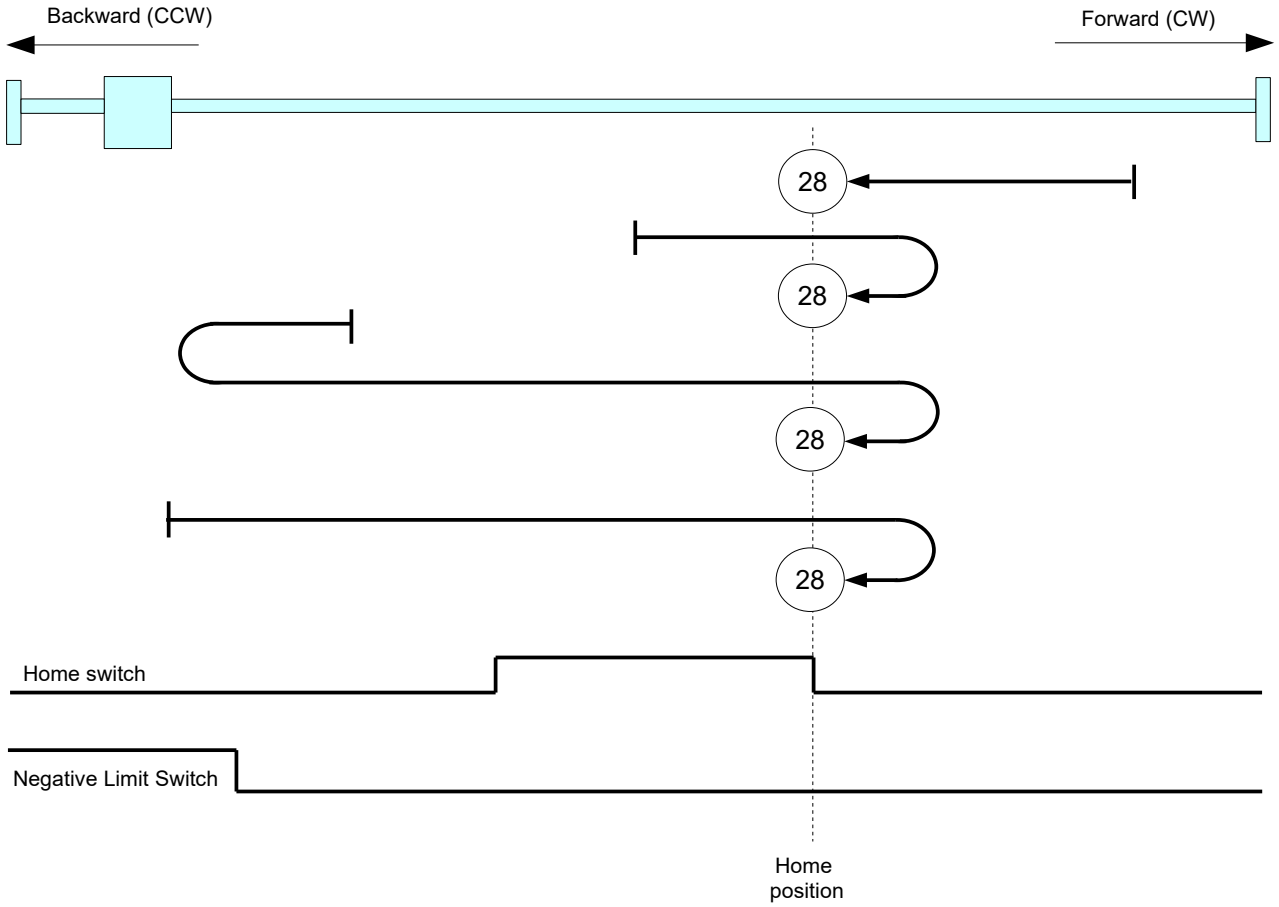
The Index pulse and Positive Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r86 or superior.

3.5.12 Homing method 28

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Negative Limit Switch*.



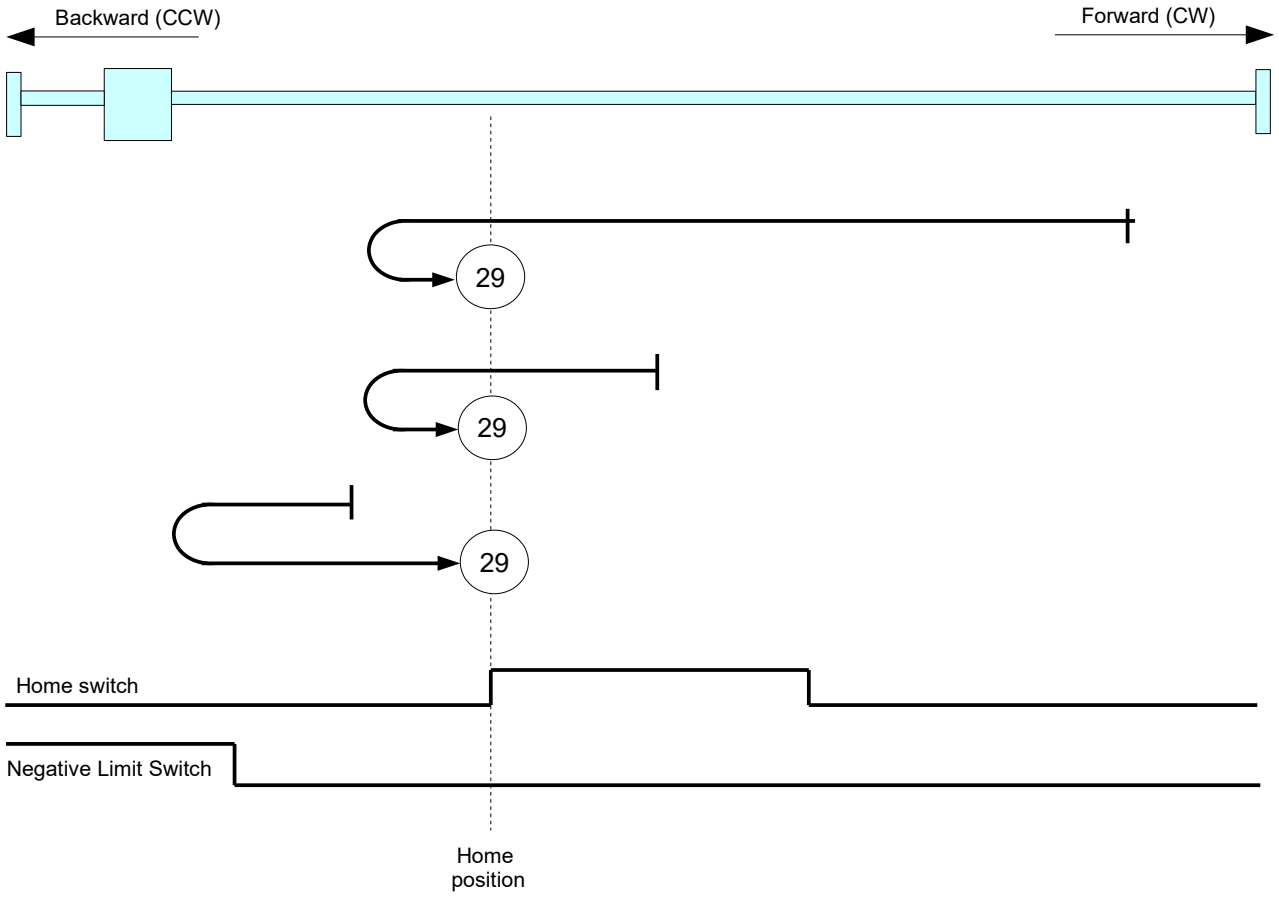
The Index pulse and Positive Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r86 or superior.

3.5.13 Homing method 29

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Negative Limit Switch*.



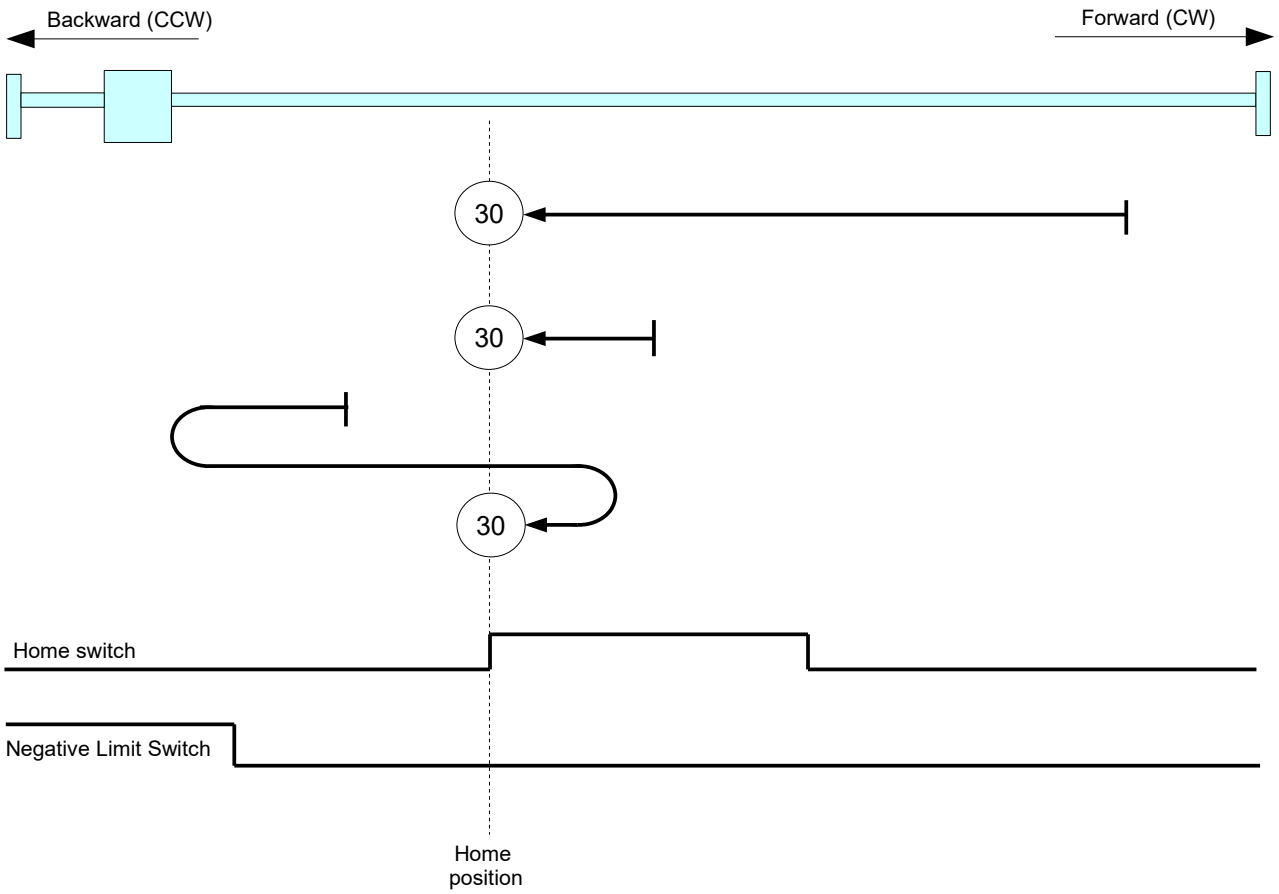
The Index pulse and Positive Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r97 or superior.

3.5.14 Homing method 30

This method uses a *Home Switch* which is present only on a portion of the travel. The initial and final directions of the movements are dependent on the state of the *Home Switch* and *Negative Limit Switch*.



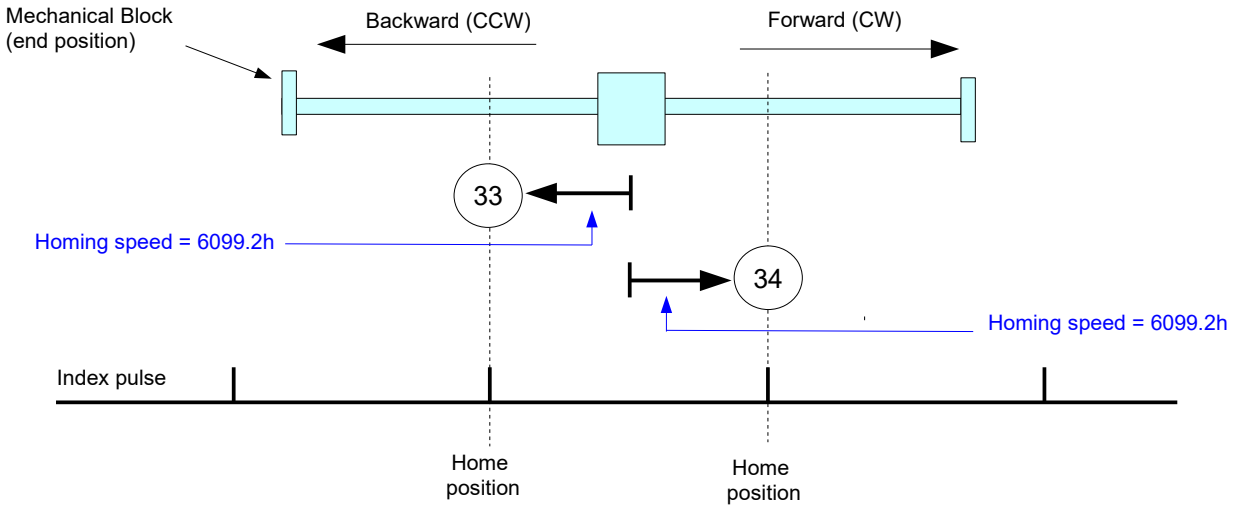
The Index pulse and Positive Limit Switch are not used.

Note:

- This Homing method is available with firmware version V02r97 or superior.

3.5.15 Homing method 33 and 34

For method 33 the initial direction of movement is leftward direction.
For method 34 the initial direction of movement is rightward direction.
The Home Position shall be at the first Index Pulse found in the selected direction.



Note: This Homing method is available with firmware version V03r60 or superior.

3.5.16 Homing method 35

In this method, no homing movement is executed. All position values (objects 6062.0H, 6063.0H, 6064.0H) are set to *Preset_Homing_Position* (default value = 0).

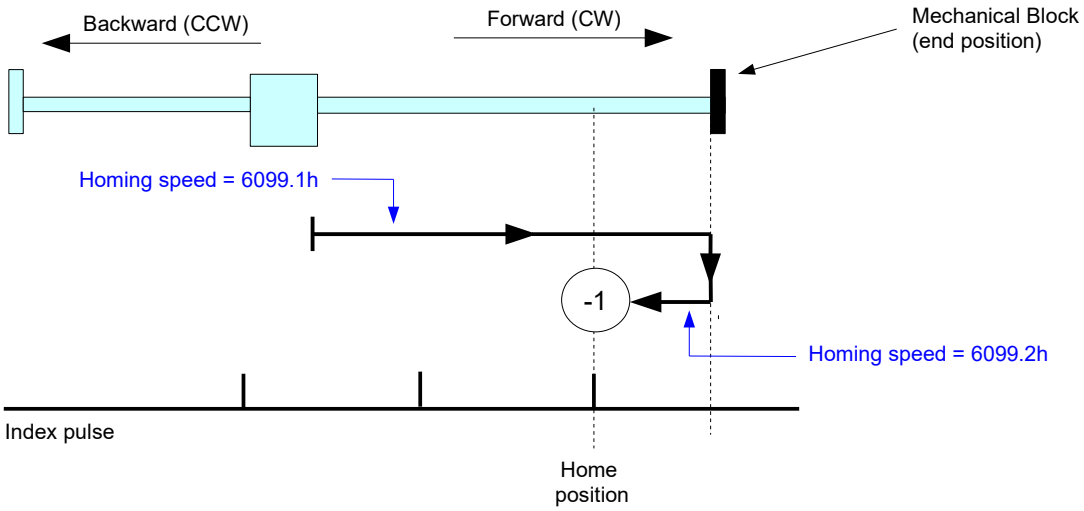
Note : This method is obsolete but it is handled for compatibility reasons.

3.5.17 Homing method 37

In this method, no homing movement is executed. All position values (objects 6062.0H, 6063.0H, 6064.0H) are set to *Home_Offset* (default value = 0).

3.5.18 Homing method -1

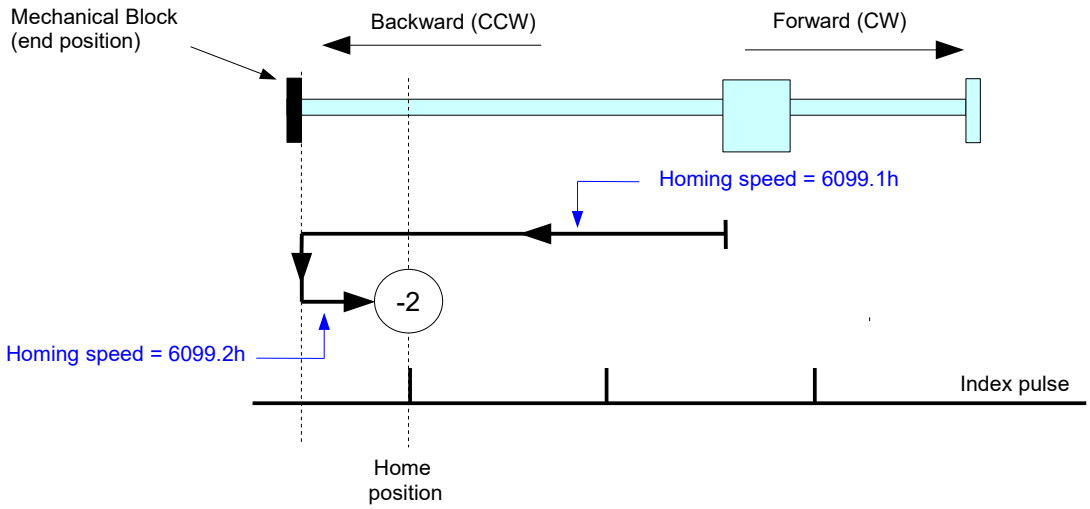
This method performs Homing on mechanical block and functions only in *Closed Loop* mode.
The initial direction of movement is rightward direction. The movement continues until the mechanical block is reached after that the direction is inverted and the movement ends on the first *Index Pulse*. *The mechanical block is detected when the Feedback_Actual_Position_Error is greater of Homing_On_Mechanical_Block_Position_Error_Limit*.
During the movement on mechanical block the motor torque is limited to *Homing_On_Mechanical_Block_Current_Limit*.
During the movement to search the *Index Pulse* the motor torque is limited to *Feedback_Boost_Current*.



Note: This Homing method is available with firmware version V03r74 or superior.

3.5.19 Homing method -2

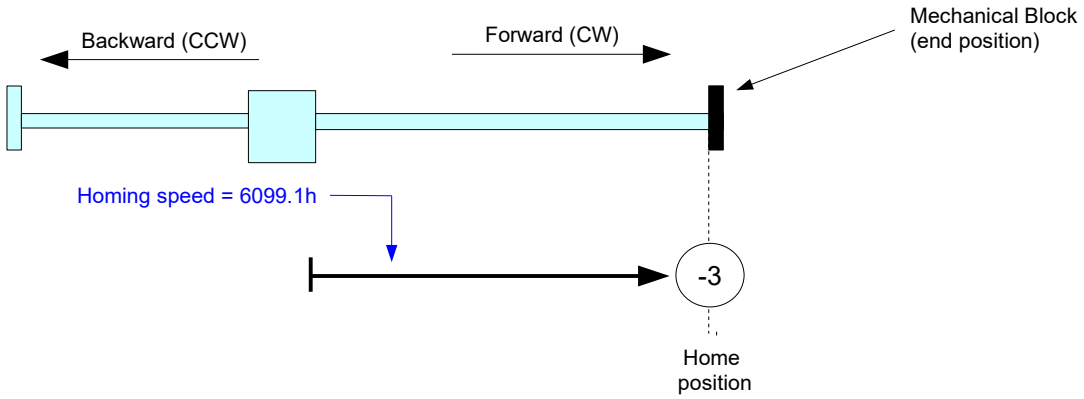
This method performs Homing on mechanical block and functions only in *Closed Loop* mode.
The initial direction of movement is leftward direction. The movement continues until the mechanical block is reached after that the direction is inverted and the movement ends on the first *Index Pulse*. *The mechanical block is detected when the Feedback_Actual_Position_Error is greater of Homing_On_Mechanical_Block_Position_Error_Limit*.
During the movement on mechanical block the motor torque is limited to *Homing_On_Mechanical_Block_Current_Limit*.
During the movement to search the *Index Pulse* the motor torque is limited to *Feedback_Boost_Current*.



Note: This Homing method is available with firmware version V03r74 or superior.

3.5.20 Homing method -3

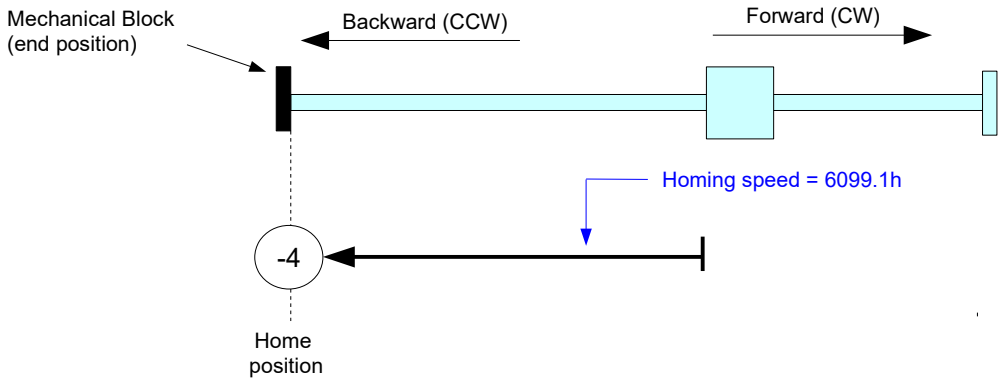
This method performs Homing on mechanical block and functions only in *Closed Loop* mode.
The direction of movement is rightward direction. The movement continues until the mechanical block is reached.
The mechanical block is detected when the *Feedback_Actual_Position_Error is greater of* *Homing_On_Mechanical_Block_Position_Error_Limit*.
During the movement on mechanical block the motor torque is limited to *Homing_On_Mechanical_Block_Current_Limit*.



Note: This Homing method is available with firmware version V03r74 or superior.

3.5.21 Homing method -4

This method performs Homing on mechanical block and functions only in *Closed Loop* mode.
The direction of movement is leftward direction. The movement continues until the mechanical block is reached.
The mechanical block is detected when the *Feedback_Actual_Position_Error is greater of* *Homing_On_Mechanical_Block_Position_Error_Limit*.
During the movement on mechanical block the motor torque is limited to *Homing_On_Mechanical_Block_Current_Limit*.



Note: This Homing method is available with firmware version V03r74 or superior.

3.6 Profile Velocity mode (pv)

This mode is used to move the motor only by mean of motor velocity. A *Target_Velocity* is applied to the trajectory generator and this generates a *Velocity_demand_value*.

The trajectory generator support only linear ramp (trapezoidal profile), with separate parameters for acceleration and deceleration.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6083.0H	VAR	Profile acceleration	UNSIGNED32	rw	M
6084.0H	VAR	Profile deceleration	UNSIGNED32	rw	O
6085.0H	VAR	Quick stop deceleration	UNSIGNED32	rw	O
606B.0H	VAR	Velocity demand value	INTEGER32	ro	M
606C.0H	VAR	Velocity actual value	INTEGER32	ro	M
606D.0H	VAR	Velocity window	UNSIGNED16	rw	O
606E.0H	VAR	Velocity window time	UNSIGNED16	rw	O
60FF.0H	VAR	Target velocity	INTEGER32	rw	M

3.7 Cyclic Synchronous Position mode (csp)

With this mode, the trajectory generator is located in the control device, not in the drive device. In cyclic synchronous manner, the control device provides a *Target_Position* to the drive device, which performs position control and velocity control. The *Target_Position* is transferred cyclically (via PDO service) every *Interpolation time period*.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6064.0H	VAR	Position actual value	INTEGER32	ro	M
6065.0H	VAR	Following error window	UNSIGNED32	rw	O
6066.0H	VAR	Following error time out	UNSIGNED16	rw	O
606C.0H	VAR	Velocity actual value	INTEGER32	ro	O
607A.0H	VAR	Target position	INTEGER32	rw	M
6085.0H	VAR	Quick stop deceleration	UNSIGNED32	rw	O
6086.0H	VAR	Motion profile type	INTEGER16	rw	O
60C2H	RECORD	Interpolation time period	Interpolation time period record	rw	O
60F4.0H	VAR	Following error actual value	INTEGER32	ro	O

3.8 Cyclic Synchronous Velocity mode (csv)

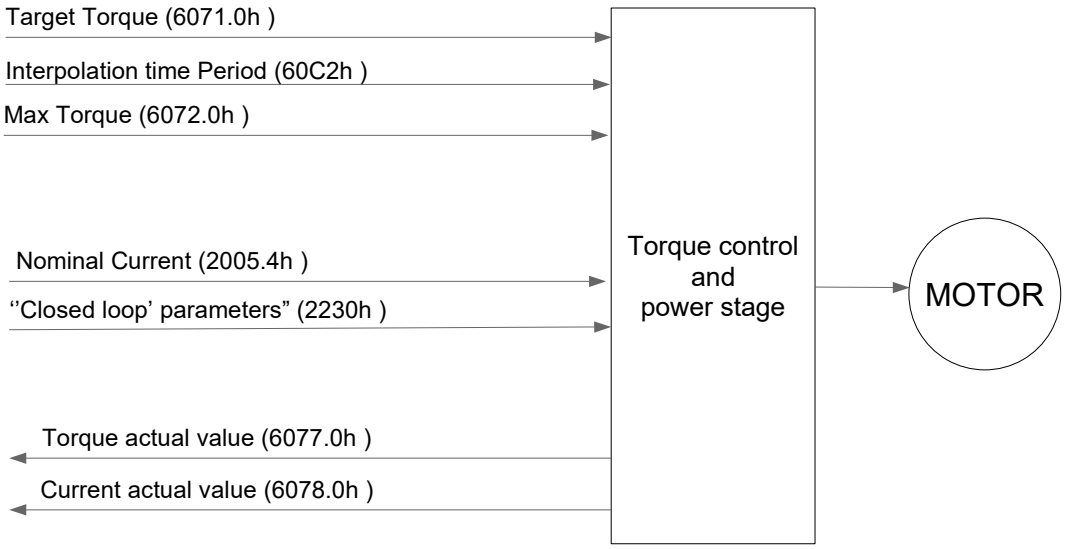
With this mode, the trajectory generator is located in the control device, not in the drive device. In cyclic synchronous manner, the control device provides a *Target_Velocity* to the drive device, which performs velocity control. The *Target_Velocity* is transferred cyclically (via PDO service) every *Interpolation time period*.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6064.0H	VAR	Position actual value	INTEGER32	ro	M
606C.0H	VAR	Velocity actual value	INTEGER32	ro	O
6085.0H	VAR	Quick stop deceleration	UNSIGNED32	rw	O
6086.0H	VAR	Motion profile type	INTEGER16	rw	O
60C2H	RECORD	Interpolation time period	Interpolation time period record	rw	O
60FF.0H	VAR	Target velocity	INTEGER32	rw	M

3.9 Cyclic Synchronous Torque (cst)

With this mode, the trajectory generator is located in the control device, not in the drive device. The control device provides a *Target_Torque* to the drive device cyclically (via PDO service) every *Interpolation time period* (*60C2H*). This mode can be used only if *Closed Loop* modality is actived.



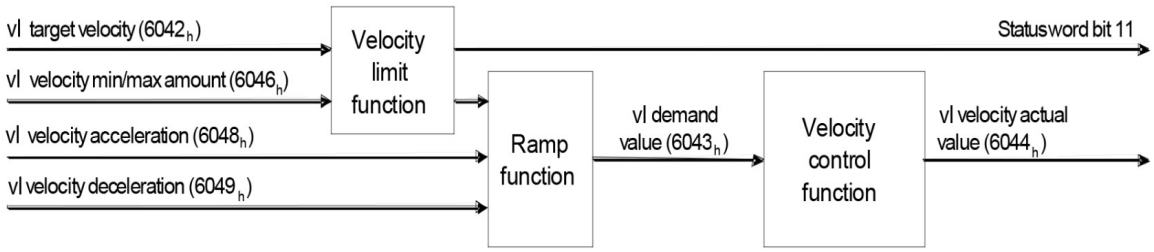
Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6071.0H	VAR	Target Torque	INTEGER16	rw	M
6072.0H	VAR	Max Torque	UNSIGNED16	rw	O
6074.0H	VAR	Torque demand	INTEGER16	ro	O
6077.0H	VAR	Torque actual value	INTEGER16	ro	M
6078.0H	VAR	Current actual value	INTEGER16	ro	O
2230H	ARRAY	'Closed loop' parameters	INTEGER32	rw	O
2005.4H	VAR	Nominal Current	UNSIGNED16	rw	M
60C2H	RECORD	Interpolation time period	Interpolation time period record	rw	O

3.10 Velocity mode (vl)

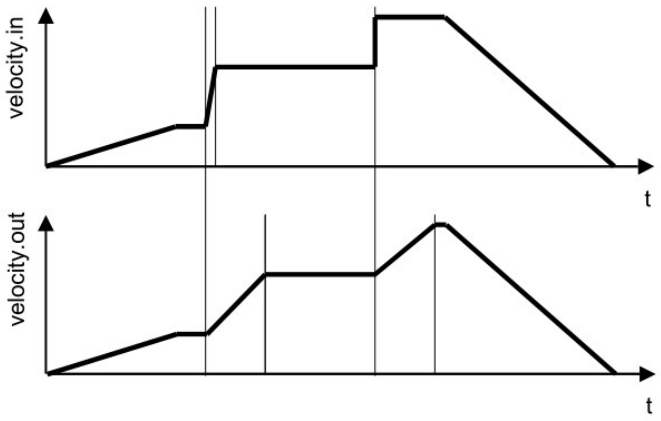
This mode is used for applications that use a velocity set-point and a *Controlword* for switching the drive device on and off.

All drive devices using this profile and supporting the velocity mode shall implement the mandatory objects and there functionality as shown in figure below :



Velocity limit function
The limits in the velocity limit function may be given in user-specific units by including the *vl Dimension Factor* in the velocity limit or in rotations per minute (rpm). The limit-value message is generated if the input value of the speed limit results in a value outside the speed limit's operating range. The limit-value message is mapped in the *Statusword* (bit11).

Ramp function
The velocity output is equal to the input as long as the changes are below as defined in *vl Velocity acceleration*, *vl Velocity deceleration* and velocity quickstop.



Velocity control function
On the basis of the *vl Velocity Demand*, the velocity control function provides the *vl Control Effort*.

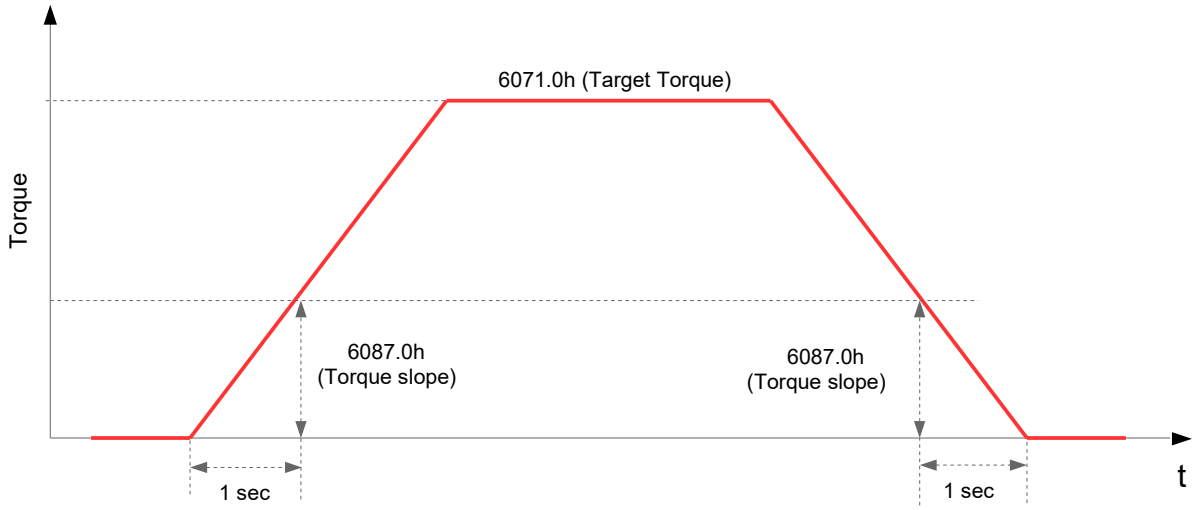
Factor function
The factor function multiplies the input variables by the assigned factors. The factor shall have a value of 1, if it is not implemented.

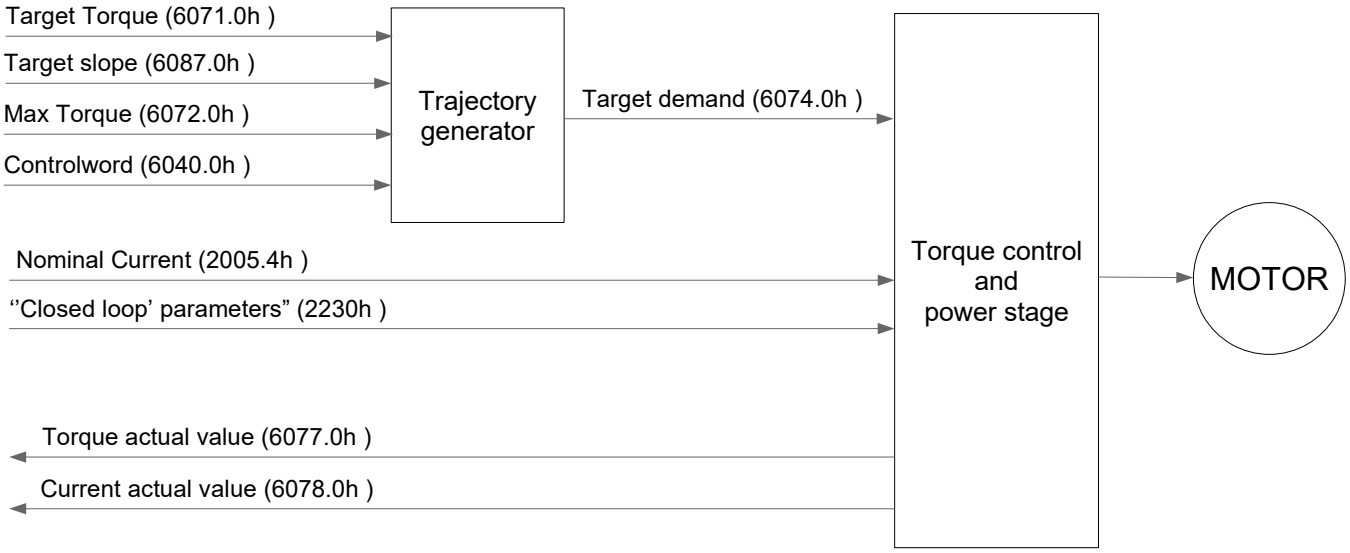
Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6042.0H	VAR	vl Target Velocity	INTEGER16	rw	M
6043.0H	VAR	vl Velocity Demand	INTEGER16	ro	M
6044.0H	VAR	vl Control Effort (vl Velocity Actual Value)	INTEGER16	ro	M
6046H	ARRAY	vl Velocity Min Max Amount	UNSIGNED 32	rw	M
6048H	RECORD	vl Velocity Acceleration	vl velocity acceleration deceleration	rw	M
6049H	RECORD	vl Velocity Deceleration	vl velocity acceleration deceleration	rw	M
604CH	ARRAY	vl Dimension Factor	INTEGER32	rw	O

3.11 Profile Torque mode (tq)

The 'Profile Torque mode' allows control device to transmit a *Target_Torque* value to drive device, which is processed via the trajectory generator by mean of *Torque_slope*. Only linear ramp (trapezoidal profile) is supported. This mode can be used only if *Closed Loop* modality is actived.





Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6071.0H	VAR	Target Torque	INTEGER16	rw	M
6072.0H	VAR	Max Torque	UNSIGNED16	rw	O
6074.0H	VAR	Torque demand	INTEGER16	ro	O
6077.0H	VAR	Torque actual value	INTEGER16	ro	M
6078.0H	VAR	Current actual value	INTEGER16	ro	O
6087.0H	VAR	Torque slope	UNSIGNED32	rw	M
2230H	ARRAY	'Closed loop' parameters	INTEGER32	rw	O
2005.4H	VAR	Nominal Current	UNSIGNED16	rw	M
2B08.0H	VAR	Torque window	UNSIGNED16	rw	O
2B09.0H	VAR	Torque window time	UNSIGNED16	rw	O

3.12 Touch Probe functionality

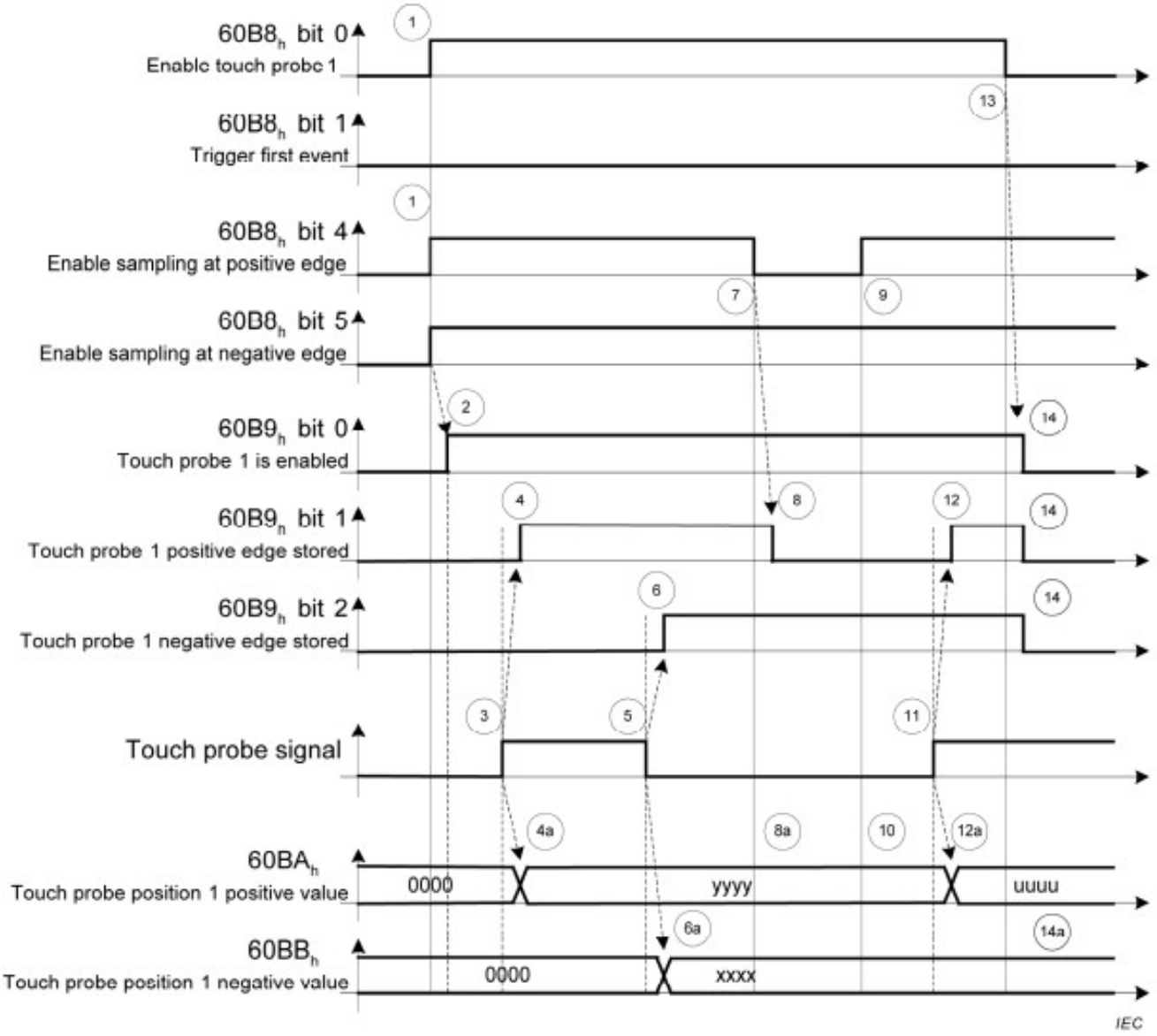
The Touch Probe functionality allows to capture the position value of the motor by sensing the edge-triggered digital input of the drive.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
60B8.0H	VAR	Touch Probe Function	UNSIGNED16	rw	M
60B9.0H	VAR	Touch Probe Status	UNSIGNED16	ro	M
60BA.0H	VAR	Touch Probe 1 Positive Edge Position	INTEGER32	ro	M
60BB.0H	VAR	Touch Probe 1 Negative Edge Position	INTEGER32	ro	M
60BC.0H	VAR	Touch Probe 2 Positive Edge Position	INTEGER32	ro	M
60BD.0H	VAR	Touch Probe 2 Negative Edge Position	INTEGER32	ro	M
60D0H	ARRAY	Touch Probe Source	INTEGER16	rw	R
60D5.0H	VAR	Touch Probe 1 Positive Edge Counter	UNSIGNED16	ro	O
60D6.0H	VAR	Touch Probe 1 Negative Edge Counter	UNSIGNED16	ro	O
60D7.0H	VAR	Touch Probe 2 Positive Edge Counter	UNSIGNED16	ro	O
60D8.0H	VAR	Touch Probe 2 Negative Edge Counter	UNSIGNED16	ro	O
2082.0H	VAR	Touch Probe 1 Filter	UNSIGNED32	rw	O
2083.0H	VAR	Touch Probe 2 Filter	UNSIGNED32	rw	O

Note:
Touch Probe functionality is available with firmware V01r11 or superior.

Touch Probe example



Number	Touch probe behavior	
(1)	$60B8_h$, bit 0 = 1_b	Enable touch probe 1
	$60B8_h$, bit 1, 4, 5	Configure and enable touch probe 1 positive and negative edge
(2)	→ $60B9_h$, bit 0 = 1_b	Status "Touch probe 1 enabled" is set
(3)	External touch probe signal has positive edge	
(4)	→ $60B9_h$, bit 1 = 1_b	Status "Touch probe 1 positve edge stored" is set
(4a)	→ $60BA_h$	Touch probe position 1 positive value is stored
(5)	External touch probe signal has negative edge	
(6)	→ $60B9_h$, bit 2 = 1_b	Status "Touch probe 1 negative edge stored" is set
(6a)	→ $60BB_h$	Touch probe position 1 negative value is stored
(7)	$60B8_h$, bit 4 = 0_b	Sample positive edge is disabled
(8)	→ $60B9_h$, bit 0 = 0_b	Status "Touch probe 1 positve edge stored" is reset
(8a)	→ $60BA_h$	Touch probe position 1 positive value is not changed
(9)	$60B8_h$, bit 4 = 1_b	Sample positive edge is enabled
(10)	→ $60BA_h$	Touch probe position 1 positive value is not changed
(11)	External touch probe signal has positive edge	
(12)	→ $60B9_h$, bit 1 = 1_b	Status "Touch probe 1 positve edge stored" is set
(12a)	→ $60BA_h$	Touch probe position 1 positive value is stored
(13)	$60B8_h$, bit 0 = 0_b	Touch probe 1 is disabled
(14)	→ $60B9_h$, bit 0, 1, 2 = 0_b	Status bits are reset
(14a)	→ $60BA_h$, $60BB_h$	Touch probe position 1 positive/negative value are not changed

3.13 Feedback Sensor Calibration mode (fsc)

By mean of this mode is possible the calibration of the Feedback Sensor used for *Closed Loop* feature.
The Feedback Sensor Calibration mode is available only with firmware version V03r21 or superior.

To activate the mode, the value "-1" must be set in the *6060.0H* object (Modes of Operation).

The type of Sensor to be calibrated is defined with the *Feedback_Settings* object (bit8÷bit11).

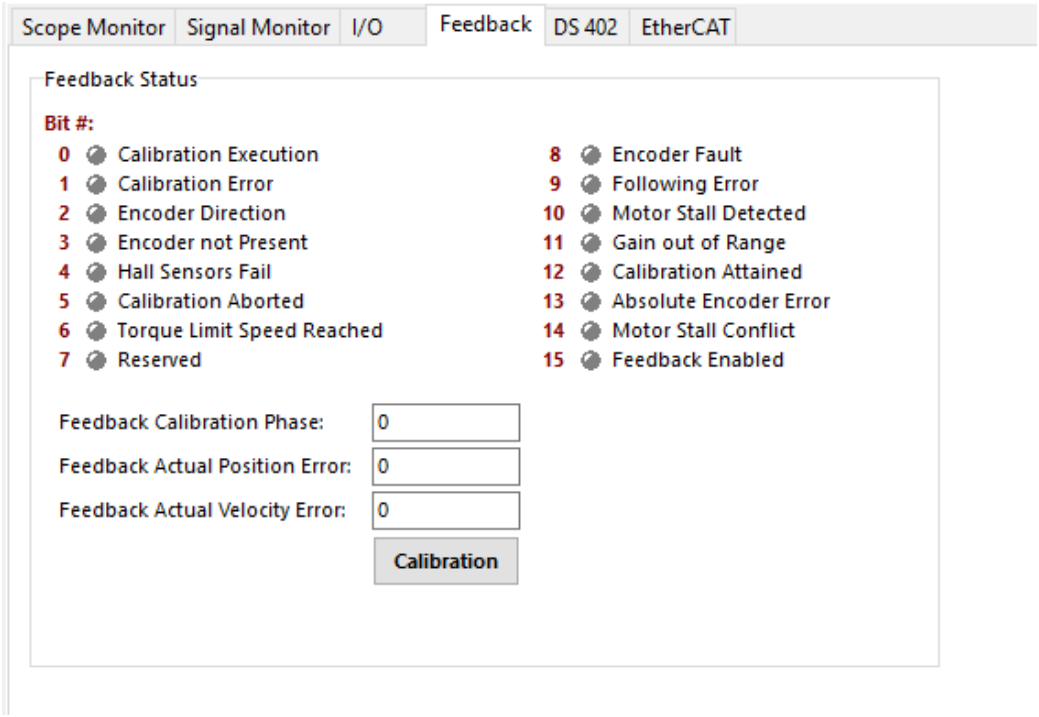
At the end of successful calibration, the calibrated values are stored in NVRAM.
Every time that *Closed Loop* feature will be actived, the calibrated values will be used and no additional calibration will be needed.

During calibration procedure the motor shaft must be load-free and free to turn in any direction.

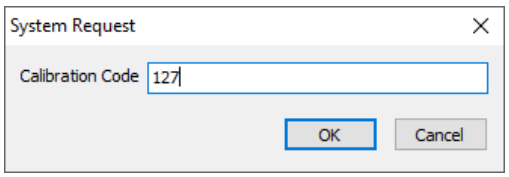
The 'EVER Studio' Tool can be also used for Feedback Sensor Calibration :

- go in the ONLINE condition by clicking on 'Connect' button of the Main window

- in the 'Feedback' window click on 'Calibration' button



And then insert the value 127 to active procedure



Before starting the calibration procedure must be defined the values of parameters concerning the type of sensor that will be calibrated .

3.13.1 Multi-Turn Absolute Encoder BiSS

Before starting the calibration procedure the following parameters must be defined :

- *Feedback_Calibration_Speed*
- *Feedback_Calibration_Current*
- *BiSS_Encoder_Config*
- *Nominal_Current*
- *Feedback_Source_PPR*
- *Motor_Pole_Pairs*
- *Motor_Step_Angle*
- *Motor_Resolution*
- *Drive_Working_Settings_Extended*
- *Motor_R* and *Motor_L* (*if bit9=0 of Drive_Working_Settings_Extended* object)

The type of Feedback Sensor has to be defined with value 6 in the (bit8÷bit11) of the *Feedback_Settings* object.
The bit12 of *Feedback_Settings* object must be set (1) to active the Full Feedback Calibration procedure.
The *Feedback_Source_PPR* object must be set to value 65536.

The calibration procedure is carried out on the transition from '0' to '1' of bit4 of the *Controlword* object.
During the procedure the bit12 of the *Statusword (fsc) object* is set (1). At the end of the procedure, if the procedure was successful the bit13 (calibration attained) of the *Statusword (fsc)* object is set (1) otherwise if an error occurred the bit14 (calibration error) of the *Statusword (fsc)* object is set (1) and the procedure is not considered successful.
The procedure can be interrupted with the transition from '1' to '0' of bit4 of the *Controlword* object.

The calibrated values that have been calibrated and stored in NVRAM can be used afterwards only if the bit14 of the *Feedback_Settings* is set (1). Otherwise if bit14 = 0 then the Sensor Calibration procedure is performed every time when the *Closed Loop* feature is actived.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6060.0H	VAR	Modes_of_Operation	INTEGER8	rw	M
2230.17H	VAR	Feedback_Settings	UNSIGNED16	rw	M
2230.18H	VAR	Feedback_Status	UNSIGNED16	ro	O
2230.1CH	VAR	Feedback_Calibration_Current	UNSIGNED16	rw	M
2230.1AH	VAR	Feedback_Calibration_Speed	UNSIGNED16	rw	M
2A04.0H	VAR	BiSS_Encoder_Config	UNSIGNED32	rw	M
2005.4H	VAR	Nominal_Current	UNSIGNED16	rw	M
2230.03H	VAR	Feedback_Source_PPR	UNSIGNED32	rw	M
2012.2H	VAR	Motor_Pole_Pairs	UNSIGNED16	rw	M
2012.1H	VAR	Motor_Step_Angle	UNSIGNED16	rw	M
60EF.0H	VAR	Motor_Resolution	UNSIGNED32	rw	M
2200.11H	VAR	Drive_Working_Settings_Extended	UNSIGNED16	rw	M
2005.6H	VAR	Motor_R	UNSIGNED32	rw	M
2005.7H	VAR	Motor_L	UNSIGNED32	rw	M

3.13.2 Single-Turn Magnetic Encoder

Before starting the calibration procedure the following parameters must be defined :

- *Feedback_Calibration_Speed*
- *Feedback_Calibration_Current*
- *Nominal_Current*
- *Feedback_Source_PPR*
- *Motor_Pole_Pairs*
- *Motor_Step_Angle*
- *Motor_Resolution*
- *Drive_Working_Settings_Extended*
- *Motor_R* and *Motor_L* *(if bit9=0 of Drive_Working_Settings_Extended* object)

The type of Feedback Sensor has to be defined with value 8 in the (bit8÷bit11) of the *Feedback_Settings* object.
The bit12 of *Feedback_Settings* object must be set (1) to active the Full Feedback Calibration procedure.

The calibration procedure is carried out on the transition from '0' to '1' of bit4 of the *Controlword* object.
During the procedure the bit12 of the *Statusword (fsc) object* is set (1). At the end of the procedure, if the procedure was successful the bit13 (calibration attained) of the *Statusword (fsc)* object is set (1) otherwise if an error occurred the bit14 (calibration error) of the *Statusword (fsc)* object is set (1) and the procedure is not considered successful.
The procedure can be interrupted with the transition from '1' to '0' of bit4 of the *Controlword* object.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6060.0H	VAR	Modes_of_Operation	INTEGER8	rw	M
2230.17H	VAR	Feedback_Settings	UNSIGNED16	rw	M
2230.18H	VAR	Feedback_Status	UNSIGNED16	ro	O
2230.1CH	VAR	Feedback_Calibration_Current	UNSIGNED16	rw	M
2230.1AH	VAR	Feedback_Calibration_Speed	UNSIGNED16	rw	M
2005.4H	VAR	Nominal_Current	UNSIGNED16	rw	M
2230.03H	VAR	Feedback_Source_PPR	UNSIGNED32	rw	M
2012.2H	VAR	Motor_Pole_Pairs	UNSIGNED16	rw	M
2012.1H	VAR	Motor_Step_Angle	UNSIGNED16	rw	M
60EF.0H	VAR	Motor_Resolution	UNSIGNED32	rw	M
2200.11H	VAR	Drive_Working_Settings_Extended	UNSIGNED16	rw	M
2005.6H	VAR	Motor_R	UNSIGNED32	rw	M
2005.7H	VAR	Motor_L	UNSIGNED32	rw	M

3.13.3 Hall Sensors

Before starting the calibration procedure the following parameters must be defined :

- *Feedback_Calibration_Speed*
- *Motor_Pole_Pairs*
- *Motor_Step_Angle*
- *Motor_Resolution*
- *Nominal_Current*
- *Min_Current*
- *Max_Current*
- *Boost_Current*
- *Drive_Working_Settings_Extended*
- *Motor_R* and *Motor_L* (*if bit9=0 of Drive_Working_Settings_Extended* object)

The type of Feedback Sensor has to be defined with value 1 (Hall Sensors) or 2 (Hall Sensors+Incremental Encoder) in the (bit8÷bit11) of the *Feedback_Settings* object.

The calibration procedure is carried out on the transition from '0' to '1' of bit4 of the *Controlword* object.
During the procedure the bit12 of the *Statusword (fsc) object* is set (1). At the end of the procedure, if the procedure was successful the bit13 (calibration attained) of the *Statusword (fsc)* object is set (1) otherwise if an error occurred the bit14 (calibration error) of the *Statusword (fsc)* object is set (1) and the procedure is not considered successful.
The procedure can be interrupted with the transition from '1' to '0' of bit4 of the *Controlword* object.

At the end of the succesfully procedure the Hall Sensors sequence detected (290A.0H) is stored in the *Hall_Sensors_Sequence_Settings* object.

Related objects :

Index	*Object*	*Name*	*Type*	*Attr.*	*M/O*
6040.0H	VAR	Controlword	UNSIGNED16	rw	M
6041.0H	VAR	Statusword	UNSIGNED16	ro	M
6060.0H	VAR	Modes_of_Operation	INTEGER8	rw	M
2230.17H	VAR	Feedback_Settings	UNSIGNED16	rw	M
2230.18H	VAR	Feedback_Status	UNSIGNED16	ro	O
2230.1AH	VAR	Feedback_Calibration_Speed	UNSIGNED16	rw	M
2005.4H	VAR	Nominal_Current	UNSIGNED16	rw	M
2012.2H	VAR	Motor_Pole_Pairs	UNSIGNED16	rw	M
2012.1H	VAR	Motor_Step_Angle	UNSIGNED16	rw	M
60EF.0H	VAR	Motor_Resolution	UNSIGNED32	rw	M
2005.1H	VAR	Min_Current	UNSIGNED16	rw	M
2005.2H	VAR	Max_Current	UNSIGNED16	rw	M
2005.3H	VAR	Boost_Current	UNSIGNED16	rw	M
2200.11H	VAR	Drive_Working_Settings_Extended	UNSIGNED16	rw	M
2005.6H	VAR	Motor_R	UNSIGNED32	rw	M
2005.7H	VAR	Motor_L	UNSIGNED32	rw	M
2901.0H	VAR	Hall_Sensors_Position	UNSIGNED16	ro	O
290A.0H	VAR	Hall_Sensors_Sequence_Settings	UNSIGNED32	rw	M
290D.0H	VAR	Hall_Sensors_Sequence_Detected	UNSIGNED32	ro	M
2900.0H	VAR	Hall_Sensors_Status	UNSIGNED16	ro	M

4.0 Object Dictionary

The object dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. The overall layout of the standard Object Dictionary is :

Index	Object
0000h - 0FFFh	Data definition / reserved
1000h - 1FFFh	Communication profile area (DS301)
2000h - 5FFFh	Manufacturer specific area
6000h - 9FFFh	Standardized device profile area (DS402)
A000h - FFFFh	Other profile / reserved

Each object within the dictionary is addressed using a 16-bit index. In case of a simple variable (VAR) the index directly references the value. In case of records (RECORD) and arrays (ARRAY), the index addresses the whole data structure. To allow individual elements of structures of data to be accessed, a sub-index has been defined. For single object dictionary entries such as an unsigned8, boolean, integer32,etc the value for the sub-index is always zero. For complex object dictionary entries such as arrays or records with multiple data fields , the sub-index refers to fields within a data-structure pointed to by the main index.

Name: **Device Type**
Index.Sub: **1000.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: 00040192H
Store Supported: --

Description: This object contains information about the device type.

Notes:

Name: **Error register**
Index.Sub: **1001.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: 0
Store Supported: No

Description: This object is an error register for the drive. More than one bit at time could be set to 1, meaning that more than one fault is active. Bit0 is set to 1 if one or more faults are active, is reset to 0 if all faults are cleared.

See §2.7 for more details.

Bit		Description	
0	**M**	**Generic error**	
1	**O**	Current	
2	**O**	Voltage	
3	**O**	Temperature	
4	**O**	Communication error	
5	**O**	Device profile specific	
6	**O**	Reserved (always 0)	(always 0)
7	**O**	**Manufacturer specific**	

Notes:

Name: **Manufacturer Specific Error Register**
Index.Sub: **1002.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: 0
Store Supported: No

Description: This object is the specific fault of the drive. See §2.7 for more details.

Notes:
This object is not available in EtherCAT fieldbus.

Name: **COB-ID Sync**
Index.Sub: **1005.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: 80h
Store Supported: --

Description: This object defines the COB-ID of Synchronization Object (*SYNC*).

Notes:

- This object is not available in EtherCAT fieldbus.

Name: **Manufacturer Device Name**
Index.Sub: **1008.0H**
Data Type: Visible String
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: **--**
Store Supported: --

Description: Contains the manufacturer device name.

Notes:

Name: **Manufacturer Software Version**
Index.Sub: **100A.0H**
Data Type: Visible String
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: **--**
Store Supported: --

Description: Contains the manufacturer software version description.

Notes:

Name: **Guard Time**
Index.Sub: **100C.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: Milliseconds
Range: 0 ÷ 65535
Default Value: 0
Store Supported: No

Description: The objects at index *100C.0H* and *100D.0H* include the guard time in milliseconds and the life time factor.The life time factor multiplied with the guard time gives the life time for the Life Guarding Protocol. It is 0 if not used.

Notes:

- This Object is not available in EtherCAT fieldbus.
- This Object is available with firmware version V00r74 or superior.

Name: **Life Time Factor**
Index.Sub: **100D.0H**
Data Type: Unsigned8
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ 255
Default Value: 0
Store Supported: No

Description: The life time factor multiplied with the guard time gives the life time for the node guarding protocol. It is 0 if not used.

Notes:

- This Object is not available in EtherCAT fieldbus.
- This Object is available with firmware version V00r74 or superior.

Name: **Store_Parameters (Save Parameters)**
Index.Sub: **1010.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range:

Value	Domains
65766173 Hex	Store Parameters
7070616D Hex	Store Ethercat Mapping (see *§5.0*)

Default Value: --
Store Supported: --

Description: This object supports the saving of drive parameters in non volatile memory. On reception of the correct signature the drive stores the current parameters values in non volatile memory. At the next drive switch on the parameters starting value will be equal to the value stored in non volatile memory.

Notes:

The storing process takes about 4-5 (worst case) seconds to be completed. If a further store parameters command is sent before the completion of the previous one, the drive will not answer to the communication interface until the previous storing process is completed. The non volatile ram is really written only if data have changed compared to the previous store process.

!!!WARNING!!! : The non volatile memory can be written for a limited number of times (typically 100,000 times), when reached that limit a message error can occur, and the drive should be sent to EVER for reparation.

Name: **Restore_Parameters**
Index.Sub: **1011.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: valid signature = 64616F6CH ('load' in ASCII hex)
Default Value: --
Store Supported: **--**

Description: Restore the drive's parameters default value at reception of the correct signature. At the next drive switch on the parameters starting value will be equal to the factory default value.

Notes: The restoring process takes about 4-5 (worst case) seconds to be completed. If a further restore parameters command is sent before the completion of the previous one, the drive will not answer to the communication interface until the previous storing process is completed. The non volatile ram is really written only if data have changed compared to the previous store process.

!!!WARNING!!! : The non volatile memory can be written for a limited number of times (typically 100,000 times), when reached that limit a message error can occur, and the drive should be sent to EVER for reparation.

Name: **COB-ID Emergency**
Index.Sub: **1014.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: 80h + NodeId
Store Supported: --

Description: This object defines the COB-ID of the Emergency Object (*EMCY*).

Notes:

- This object is not available in EtherCAT fieldbus.

Name: **Producer_Heartbeat_Time**
Index.Sub: **1017.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: 0 ÷ 65535
Default Value: 500
Store Supported: Yes

Description: This object defines the cycle time of the heartbeat message. If is 0 the heartbeat transmission is disabled.

Notes:

- This object is not available in EtherCAT fieldbus.

Name: **Vendor_ID**
Index.Sub: **1018.1H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: 4BH
Store Supported: No

Description: EVER Vendor ID assigned by CiA.

Notes:

Name: **Product_Code**
Index.Sub: **1018.2H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: --
Store Supported: No

Description: Drive Hardware Code.

Notes:

Name: **Revision_Number**
Index.Sub: **1018.3H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: --
Store Supported: No

Description: Drive Hardware Revision Number.

Notes:

Name: **Serial_Number**
Index.Sub: **1018.4H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: --
Store Supported: No

Description: This buffer contains the drive's serial number. If the serial number has not been stored in the drive this object return FFFFFFFFH.

Notes:

Name: **Server SDO Parameters (Cob-Id Client → Server)**
Index.Sub: **1200.1H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: 600h + NodeId
Store Supported: --

Description: This object defines the COB-ID of the SDO requests sent by the master to the drive.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Server SDO Parameters (Cob-Id Server → Client)**
Index.Sub: **1200.2H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: 580h + NodeId
Store Supported: --

Description: This object defines the COB-ID of the SDO answers sent by the drive to the master.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **RX_PDOx_Cob_Id**
Index.Sub: **1400.1H** (RX_PDO1)
1401.1H (RX_PDO2)
1402.1H (RX_PDO3)
1403.1H (RX_PDO4)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: 1h÷7FFh,
80000000h (PDO disabled)
Default Value: 200h+NodeId (RX_PDO1)
300h+NodeId (RX_PDO2)
400h+NodeId (RX_PDO3)
500h+NodeId (RX_PDO4)
Store Supported: No

Description: This is the Cob-Id of RX_PDOx. Can be changed anytime but cannot be stored in NVRAM.

Notes:
- This object is not available in EtherCAT fieldbus.
- The 1402.1H and 1403.1H objects are available with firmware version V00r67 or superior.

Name: **RX_PDOx_Transmission_Type**
Index.Sub: **1400.2H** (RX_PDO1)
1401.2H (RX_PDO2)
1402.2H (RX_PDO3)
1403.2H (RX_PDO4)
Data Type: Unsigned8
Access: rw
PDO Mapping: no
Unit: **--**
Range: 0÷255
Default Value: 1
Store Supported: No

Description: These are the Transmission Type of RX_PDOx. Can be changed anytime but cannot be stored in NVRAM. The allowed values are:

Value	*Description*
0÷240	PDO Synchronous. Handled after the reception of each SYNC message.
254/255	PDO Asynchronous. Handled as soon as after the reception.

Notes:
- This object is not available in EtherCAT fieldbus.
- The 1402.2H and 1403.2H objects are available with firmware version V00r67 or superior.

Name: **RX_PDOx_Mapping_HighestSubIndexSupported**
Index.Sub: **1600.0H** (RX_PDO1)
1601.0H (RX_PDO2)
1602.0H (RX_PDO3)
1603.0H (RX_PDO4) (available only for CANbus fieldbus)

Data Type: Unsigned8

Access: rw (for 1600.0H)
rw (for 1601.0H, 1602.0H, 1603.0H with CANbus fieldbus)
ro (for 1601.0H, 1602.0H with EtherCAT fieldbus)

PDO Mapping: no
Unit: –

Range: 0 ÷ 4 (for 1600.0H, 1601.0H, 1602.0H, 1603.0H with CANbus fieldbus)
0 ÷ 16 (for 1600.0H with EtherCAT fieldbus)
7 (for 1601.0H with EtherCAT fieldbus)
7 (for 1602.0H with EtherCAT fieldbus)

Default Value: 1600.0H = 1 (CANbus fieldbus) or 6 (EtherCAT fieldbus)
1601.0H = 2 (CANbus fieldbus) or 7 (EtherCAT fieldbus)
1602.0H = 2 (CANbus fieldbus) or 7 (EtherCAT fieldbus)
1603.0H = 1 (CANbus fieldbus)

Store Supported: No

Description: This object contains the number of valid entries within the mapping record. The number of entries shall be the number of objects received with the corresponding RX_PDOx.

Notes:
- The 1602.0H and 1603.0H objects are available with firmware version V00r67 or superior.
- For EtherCAT fieldbus only 1600.0H, 1601.0H, 1602.0H are available.
- For EtherCAT fieldbus see (§5.2).

Name: **RX_PDO1_Mapping**

Index.Sub:

1600.1H (1st **Obj)**
1600.2H (2nd **Obj)**
1600.3H (3rd **Obj)**
1600.4H (4th **Obj)**
1600.5H (5th **Obj)** (available only for Ethercat fieldbus)
1600.6H (6th **Obj)** (available only for Ethercat fieldbus)
1600.7H (7th **Obj)** (available only for Ethercat fieldbus)
1600.8H (8th **Obj)** (available only for Ethercat fieldbus)
1600.9H (9th **Obj)** (available only for Ethercat fieldbus)
1600.AH (10th **Obj)** (available only for Ethercat fieldbus)
1600.BH (11th **Obj)** (available only for Ethercat fieldbus)
1600.CH (12h **Obj)** (available only for Ethercat fieldbus)
1600.DH (13th **Obj)** (available only for Ethercat fieldbus)
1600.EH (14h **Obj)** (available only for Ethercat fieldbus)
1600.FH (15th **Obj)** (available only for Ethercat fieldbus)
1600.10H (16th **Obj)** (available only for Ethercat fieldbus)

Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh

Default Value:
1600.1H = 60400010h (CANbus fieldbus) or 60400010h EtherCAT fieldbus)
1600.2H = - - - - - - - - (CANbus fieldbus) or 60600008h (EtherCAT fieldbus)
1600.3H = - - - - - - - - (CANbus fieldbus) or 62000108h (EtherCAT fieldbus)
1600.4H = - - - - - - - - (CANbus fieldbus) or 607A0032h (EtherCAT fieldbus)
1600.5H = - - - - - - - - (CANbus fieldbus) or 60C10132h (EtherCAT fieldbus)
1600.6H = - - - - - - - - (CANbus fieldbus) or 60FF0032h (EtherCAT fieldbus)
1600.7H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.8H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.9H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.AH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.BH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.CH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.DH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.EH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.FH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1600.10H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)

Store Supported: No

Description:
Information concerning the Objects mapped into RX_PDO1.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:

- For EtherCAT fieldbus see (§5.2).

Name: **RX_PDO2_Mapping**
Index.Sub: **1601.1H (1st Obj)**
1601.2H (2nd Obj)
1601.3H (3rd Obj)
1601.4H (4th Obj)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh
Default Value: 60400010h (1st Obj)
60600008h (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)
Store Supported: No

Description: Information concerning the Objects mapped into RX_PDO2.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:
- For EtherCAT fieldbus see (§5.2).

Name: **RX_PDO3_Mapping**
Index.Sub: **1602.1H (1st Obj)**
1602.2H (2nd Obj)
1602.3H (3rd Obj)
1602.4H (4th Obj)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh
Default Value: 607A0020h (1st Obj)
60810020h (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)
Store Supported: No

Description: Information concerning the Objects mapped into RX_PDO3.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:
- For EtherCAT fieldbus see (§5.2).
- The objects are available with firmware version V00r67 or superior.

Name: **RX_PDO4_Mapping**
Index.Sub: **1603.1H (1st Obj)**
1603.2H (2nd Obj)
1603.3H (3rd Obj)
1603.4H (4th Obj)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh
Default Value: 60FF0020h (1st Obj)
- - - - - - - - (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)
Store Supported: No

Description: Information concerning the Objects mapped into RX_PDO4.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:
- These objects are not available in EtherCAT fieldbus.
- The objects are available with firmware version V00r67 or superior.

Name: **TX_PDOx_Cob_Id**
Index.Sub: **1800.1H** (TX_PDO1)
1801.1H (TX_PDO2)
1802.1H (TX_PDO3)
1803.1H (TX_PDO4)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: 1h÷7FFh,
80000000h (PDO disabled)
Default Value: 180h+NodeId (TX_PDO1)
280h+NodeId (TX_PDO2)
380h+NodeId (TX_PDO3)
480h+NodeId (TX_PDO4)
Store Supported: No

Description: This is the Cob-Id of transmit TX_PDOx. Can be changed anytime but cannot be stored in NVRAM.

Notes:
- This object is not available in EtherCAT fieldbus.
- The 1802.1H and 1803.1H objects are available with firmware version V00r67 or superior.

Name: **TX_PDOx_Transmission_Type**
Index.Sub: **1800.2H** (TX_PDO1)
1801.2H (TX_PDO2)
1802.2H (TX_PDO3)
1803.2H (TX_PDO4)
Data Type: Unsigned8
Access: rw
PDO Mapping: no
Unit: --
Range: 0÷255
Default Value: 1
Store Supported: No

Description: These are the Transmission Type of TX_PDOx. Can be changed anytime but cannot be stored in NVRAM. The allowed values are:

Value	*Description*
0	PDO Synchronous and Acyclic. Transmitted after the reception of each *SYNC* message
1÷240	PDO Synchronous. Transmitted after the reception of n *SYNC* messages (where n = Value)
254/255	PDO Asynchronous. Transmitted only when it changes (if *Event_Timer* = 0) or periodically every *Event_Timer*

Notes:

- This object is not available in EtherCAT fieldbus.
- The 1802.2H and 1803.2H are available with firmware version V00r67 or superior.

Name: **TX_PDOx_Event_Timer**
Index.Sub: **1800.5H** (TX_PDO1)
1801.5H (TX_PDO2)
1802.5H (TX_PDO3)
1803.5H (TX_PDO4)
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: 0÷65535
Default Value: 100
Store Supported: Yes

Description: This is the event timer of TX_PDOx. If value equal to 0 the TX_PDOx is transmitted only when it changes. If value > 0 the TX_PDOx is transmitted every event timer milliseconds.

Notes:

- This object is not available in EtherCAT fieldbus.
- The 1802.5H and 1803.5H objects are available with firmware version V00r67 or superior.

Name: **TX_PDOx_Mapping_HighestSubIndexSupported**
Index.Sub: **1A00.0H** (TX_PDO1)
1A01.0H (TX_PDO2)
1A02.0H (TX_PDO3)
1A03.0H (TX_PDO4) (available only for CANbus fieldbus)

Data Type: Unsigned8

Access: rw (for 1A00.0H)
rw (for 1A01.0H, 1A02.0H, 1A03.0H with CANbus fieldbus)
ro (for 1A01.0H, 1A02.0H with EtherCAT fieldbus)

PDO Mapping: no
Unit: –

Range: 0 ÷ 4 (for 1A00.0H, 1A01.0H, 1A02.0H, 1A03.0H with CANbus fieldbus)
0 ÷ 16 (for 1A00.0H with EtherCAT fieldbus)
10 (for 1A01.0H with EtherCAT fieldbus)
5 (for 1A02.0H with EtherCAT fieldbus)

Default Value: 1A00.0H = 1 (CANbus fieldbus) or 6 (EtherCAT fieldbus)
1A01.0H = 2 (CANbus fieldbus) or 10 (EtherCAT fieldbus)
1A02.0H = 2 (CANbus fieldbus) or 5 (EtherCAT fieldbus)
1A03.0H = 1 (CANbus fieldbus)

Store Supported: No

Description: This object contains the number of valid entries within the mapping record. The number of entries shall be the number of objects transmitted with the corresponding TX_PDOx.

Notes:
- The 1A02.0H and 1A03.0H objects are available with firmware version V00r67 or superior.
- For EtherCAT fieldbus only 1A00.0H, 1A01.0H, 1A02.0H are available.
- For EtherCAT fieldbus see (§5.2).

Name: **TX_PDO1_Mapping**

Index.Sub:

1A00.1H	**(1st**	**Obj)**
1A00.2H	**(2nd**	**Obj)**
1A00.3H	**(3rd**	**Obj)**
1A00.4H	**(4th**	**Obj)**
1A00.5H	**(5th**	**Obj)** (available only for Ethercat fieldbus)
1A00.6H	**(6th**	**Obj)** (available only for Ethercat fieldbus)
1A00.7H	**(7th**	**Obj)** (available only for Ethercat fieldbus)
1A00.8H	**(8th**	**Obj)** (available only for Ethercat fieldbus)
1A00.9H	**(9th**	**Obj)** (available only for Ethercat fieldbus)
1A00.AH	**(10th**	**Obj)** (available only for Ethercat fieldbus)
1A00.BH	**(11th**	**Obj)** (available only for Ethercat fieldbus)
1A00.CH	**(12h**	**Obj)** (available only for Ethercat fieldbus)
1A00.DH	**(13th**	**Obj)** (available only for Ethercat fieldbus)
1A00.EH	**(14h**	**Obj)** (available only for Ethercat fieldbus)
1A00.FH	**(15th**	**Obj)** (available only for Ethercat fieldbus)
1A00.10H	**(16th**	**Obj)** (available only for Ethercat fieldbus)

Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh

Default Value: 60410010h (1st Obj)
- - - - - - - - (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)

Default Value:
1A00.1H = 60410010h (CANbus fieldbus) or 60410010h EtherCAT fieldbus)
1A00.2H = - - - - - - - - (CANbus fieldbus) or 60610008h (EtherCAT fieldbus)
1A00.3H = - - - - - - - - (CANbus fieldbus) or 60000108h (EtherCAT fieldbus)
1A00.4H = - - - - - - - - (CANbus fieldbus) or 60640032h (EtherCAT fieldbus)
1A00.5H = - - - - - - - - (CANbus fieldbus) or 606C0032h (EtherCAT fieldbus)
1A00.6H = - - - - - - - - (CANbus fieldbus) or 60F40032h (EtherCAT fieldbus)
1A00.7H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.8H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.9H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.AH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.BH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.CH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.DH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.EH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.FH = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)
1A00.10H = - - - - - - - - (CANbus fieldbus) or - - - - - - - - (EtherCAT fieldbus)

Store Supported: No

Description: Information concerning the Objects mapped into TX_PDO1.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:

- For EtherCAT fieldbus see (§5.2).

Name: **TX_PDO2_Mapping**
Index.Sub: **1A01.1H (1st Obj)**
1A01.2H (2nd Obj)
1A01.3H (3rd Obj)
1A01.4H (4th Obj)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh
Default Value: 60410010h (1st Obj)
60610008h (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)
Store Supported: No

Description: Information concerning the Objects mapped into TX_PDO2.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:
- For EtherCAT fieldbus see (§5.2).

Name: **TX_PDO3_Mapping**
Index.Sub: **1A02.1H (1st Obj)**
1A02.2H (2nd Obj)
1A02.3H (3rd Obj)
1A02.4H (4th Obj)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh
Default Value: 606C0020h (1st Obj)
60640020h (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)
Store Supported: No

Description: Information concerning the Objects mapped into TX_PDO3.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:
- For EtherCAT fieldbus see (§5.2).
- The objects are available with firmware version V00r67 or superior.

Name: **TX_PDO4_Mapping**
Index.Sub: **1A03.1H (1st Obj)**
1A03.2H (2nd Obj)
1A03.3H (3rd Obj)
1A03.4H (4th Obj)
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0h÷FFFFFFFFh
Default Value: 603F0010h (1st Obj)
- - - - - - - - (2nd Obj)
- - - - - - - - (3rd Obj)
- - - - - - - - (4th Obj)
Store Supported: No

Description: Information concerning the Objects mapped into TX_PDO4.
For each mapped object the information are Index, sub-index and length (defined in number of bits) :

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

Notes:
- These objects are not available in EtherCAT fieldbus.
- The objects are available with firmware version V00r67 or superior.

Name: **EtherCAT_RxPDO_Assign_Outputs**
Index.Sub: **1C12.0H**
Data Type: UINT8
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: 1

Store Supported: Yes

Description: This object defines the number of PDO assigned Outputs for EtherCAT fieldbus.

Notes:
- This object is available only for EtherCAT fieldbus (not available for CANbus fieldbus).
- See (§5.2).

Name: **EtherCAT_RxPDO_Assign_1st_allocated**
Index.Sub: **1C12.1H**
Data Type: UINT8
Access: ro
PDO Mapping: no
Unit: --
Range: 1600H, 1601H, 1602H
Default Value: 1600H

Store Supported: Yes

Description: This object defines the Index of the associated RxPDO Mapping object for EtherCAT fieldbus.

Notes:
- This object is available only for EtherCAT fieldbus (not available for CANbus fieldbus). -
See (§5.2).

Name: **EtherCAT_TxPDO_Assign_Inputs**
Index.Sub: **1C13.0H**
Data Type: UINT8
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: 1

Store Supported: Yes

Description: This object defines the number of PDO assigned Inputs for EtherCAT fieldbus.

Notes:

- This object is available only for EtherCAT fieldbus (not available for CANbus fieldbus).
- See (§5.2).

Name: **EtherCAT_TxPDO_Assign_1st_allocated**
Index.Sub: **1C13.1H**
Data Type: UINT8
Access: ro
PDO Mapping: no
Unit: --
Range: 1A00H, 1A01H, 1A02H
Default Value: 1A00H

Store Supported: Yes

Description: This object defines the Index of the associated TxPDO Mapping object for EtherCAT fieldbus.

Notes:

- This object is available only for EtherCAT fieldbus (not available for CANbus fieldbus). - See (§5.2).

Name: **Dips**
Index.Sub: **2004.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: 0 ÷ 4095 (0FFFH)
Default Value: --
Store Supported: No

Description: This object contains the current status of drive's dip switches

Notes:

- This object is not available in EtherCAT fieldbus.

Name: **Min_Current**
Index.Sub: **2005.1H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 0
Store Supported: Yes

Description: It sets the motor's reduced current. The drive automatically reduces the current after a current reduction time at the end of the movement.

This object is used for *Open Loop*.

Notes:

- The current set is the Irms current.
- *(1) available with firmware V02r20 or superior.*

Name: **Max_Current**
Index.Sub: **2005.2H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 0
Store Supported: Yes

Description: It sets the motor's current when running at constant speed. During acceleration and deceleration, the *Boost_Current* is automatically set.

This object is used for *Open Loop* feature.

Notes:

- The current set is the Irms current.
- *(1) available with firmware V02r20 or superior.*

Name: **Boost_Current**
Index.Sub: **2005.3H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 0
Store Supported: Yes

Description: It sets the boost current in the motor. The boost current is enabled when the motor accelerates and decelerates.

This object is used for *Open Loop* feature.

Notes:

- The current set is the Irms current.
- *(1) available with firmware V02r20 or superior.*

Name: **Nominal_Current**
Index.Sub: **2005.4H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 65535
Store Supported: Yes

Description: This object defines the rated current (rms) of the motor. If value = 65535 no rated current is defined.

Notes:

- *(1) available with firmware V03r02 or superior.*

Name: **Motor_R**
Index.Sub: **2005.6H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: ohm $x10^{-3}$
Range: Unsigned32
Default Value: 380
Store Supported: Yes

Description: This object sets the motor R (resistance) used for Motor Current Regulation.
This value is used when Motor R,L detection function is disabled (bit#9 of *Drive_Working_Settings_Extended* object equal to 0).

The *Nominal_Current* of the Motor must be defined.

Notes: See *§7.1*

Name: **Motor_L**
Index.Sub: **2005.7H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: H $x10^{-6}$
Range: Unsigned32
Default Value: 2420
Store Supported: Yes

Description: This object sets the motor L (inductance) used for Motor Current Regulation.
This value is used when Motor R,L detection function is disabled (bit#9 of *Drive_Working_Settings_Extended* object equal to 0).

The *Nominal_Current* of the Motor must be defined.

Notes: See *§7.1*

Name: **Motor_R_detected**
Index.Sub: **2005.8H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: ohm $x10^{-3}$
Range: Unsigned32
Default Value: --
Store Supported: No

Description: This object returns the motor R (resistance) detected by mean of Motor R,L detection function (bit#9 of *Drive_Working_Settings_Extended* object equal to 1).

The *Nominal_Current* of the Motor must be defined.

Notes: - This object is not available in EtherCAT fieldbus.
- See *§7.1*

Name: **Motor_L_detected**
Index.Sub: **2005.9H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: H $x10^{-6}$
Range: Unsigned32
Default Value: --
Store Supported: No

Description: This object returns the motor L (inductance) detected by mean of Motor R,L detection function (bit#9 of *Drive_Working_Settings_Extended* object equal to 1).

The *Nominal_Current* of the Motor must be defined.

Notes: - This object is not available in EtherCAT fieldbus.
- See *§7.1*

Name: **Motor_Stall_Max_Err_Angle**
Index.Sub: **2005.AH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: (0.01 rad)
Range: 0÷65535
Default Value: 1256
Store Supported: Yes

Description: This object defines the maximum allowed displacement between theoretical angle and estimated angle of motor rotor position for 'Motor Stall detection' feature. When the displacement limit is reached , the motor is stopped (*Min_Current* object value is applied to the motor) and Fault State is issued.

Notes: - See *§7.3*
- This object is available only with firmware version V02r74 or superior.

Name: **Motor_Stall_Actual_Err_Angle**
Index.Sub: **2005.BH**
Data Type: Signed16
Access: ro
PDO Mapping: no
Unit: (0.01 rad)
Range: -32768÷32767
Default Value: --
Store Supported: No

Description: This object returns the actual displacement between theoretical angle and estimated angle of motor rotor position for 'Motor Stall detection' feature.

Notes: - See *§7.3*
- This object is available only with firmware version V02r74 or superior.

Name: **Motor_Stall_Filter_Time**
Index.Sub: **2005.CH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: microseconds
Range: 0÷65535
Default Value: 200
Store Supported: Yes

Description: This object defines the time of the software filter used to calculate the estimated angle of motor rotor position for 'Motor Stall detection' feature.

Notes: - See *§7.3*
- This object is available only with firmware version V02r74 or superior.

Name: **Encoder_Actual_Value[0÷1]**
Index.Sub: **2007.0H,2008.0H**
Data Type: Integer32
Access: rw
PDO Mapping: TX_PDO [1]
Unit: Increments
Range: -2147483648 ÷ 2147483647
Default Value: 0
Store Supported: No

Description: This object contains the encoder #0 and #1 current position.

Notes:
- [1] *available with firmware V00r74 or superior.*
- The number of encoders available depends on the version of the drive currently in use.

Name: **Drive_Voltage_Actual_Value**
Index.Sub: **2009.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: Volts
Range: Minimum Drive Voltage ÷ Maximum Drive Voltage
Default Value: --
Store Supported: No

Description: This object contains the current voltage that powers the drive.

Notes:

Name: **Drive_Temperature_Actual_Value**
Index.Sub: **200A.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: °C
Range: 0 ÷ Maximum Drive Temperature
Default Value: --
Store Supported: No

Description: This object contains the current drive's temperature.

Notes:

Name: **Min_Profile_Velocity**
Index.Sub: **2010.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit : User-defined velocity unit *(1)*
Range: 0 ÷ '*Max*' *(2)*
Default V alue: 250
Store Supported: No

Description: This object defines the minimum motor velocity for the acceleration and deceleration ramp.

It can only be set with the motor at a standstill and cannot be higher than the Max_Profile_Velocity object.
If the value to be set is lower than the minimum value of range, the minimum value is stored.
If the value to be set is higher than the maximum value of range, the maximum value is stored.

Notes:

- *(1)* See *§3.2*
- *(2)* '*Max*' value is calculated according to this formula :

*Max(User-defined velocity unit) = 150000 * (2013.2H / 2013.1H) * (60EF.0H / 65536)*

- This object is not available in EtherCAT fieldbus.

Name: **Motor_Parameters (Highest sub-index supported)**
Index.Sub: **2012.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: --
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

Notes:

Name: **Motor_Parameters (Motor_Step_Angle)**
Index.Sub: **2012.1H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: --
Range: 1; 2; 4; 5; 8; 10; 16; 25; 32; 50; 64; 125; 128; 250; 256; 65535
Default Value: 1
Store Supported: Yes

Description: The *Motor_Step_Angle* and Motor_Pole_Pairs are used to define the motor resolution .

- If '*Motor_Step_Angle*' value is different from 65535, the number of motor Increments per revolution are computed as follow.

  TITANIO drives (Stepper motors) :

  *Motor_Resolution (Inc/rev) = (Motor_Pole_Pairs * 4) * Motor_Step_Angle;*

  PLATINO *drives (BLDC motors) &* ***VANADIO*** *drives (BLAC motors) :*

  *Motor_Resolution (Inc/rev) = (Motor_Pole_Pairs * 6) * Motor_Step_Angle;*

- If '*Motor_Step_Angle*' value is 65535(2) , the number of motor Increments per revolution is :

  Motor_Resolution (Inc/rev) = 65536

Notes:

- (1) *available with firmware V03r02 or superior.*

- This object cannot be modified when 'Operation Enable' State is active.
- If the value set is not valid the default value is stored.

- See *§3.2*
- See *2012.2H* , *60EF.0H* objects.

- With firmware V03r16 or lower the motor resolution is defined by mean of *Motor_Step_Angle* and *Motor_Pole_Pairs* objects (in this case *Motor_Resolution* object is read only and returns the motor resolution value).

- With firmware V03r17 or superior the motor resolution can be defined by mean of *Motor_Resolution* object or *Motor_Step_Angle*, *Motor_Pole_Pairs* objects. If *Motor_Resolution* object value is 0 then the motor resolution is defined by mean of *Motor_Step_Angle* and *Motor_Pole_Pairs* objects (for compatibility reasons). A value of *Motor_Resolution* object different from zero defines directly the motor resolution (*Motor_Step_Angle* object is not considered and only *Motor_Pole_Pairs* object have to be defined).

Name: **Motor_Parameters (Motor_Pole_Pairs)**
Index.Sub: **2012.2H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: # of motor Pole Pairs
Range: 2; 3; 4; 5; 6; 8; 10; 11; 12; 15; 25; 45; 50; 100
Default Value: 50 (typical value for stepper motors)
Store Supported: Yes

Description: It sets the number of motor pole pairs.

Notes:

- This object cannot be modified when 'Operation Enable' State is active.
- If the value set is not valid the default value is stored.

- See *§3.2*
- See *2012.1H* , *60EF.0H* objects.

- With firmware V03r16 or lower the motor resolution is defined by mean of *Motor_Step_Angle* and *Motor_Pole_Pairs* objects (in this case *Motor_Resolution* object is read only and returns the motor resolution value).

- With firmware V03r17 or superior the motor resolution can be defined by mean of *Motor_Resolution* object or *Motor_Step_Angle*, *Motor_Pole_Pairs* objects. If *Motor_Resolution* object value is 0 then the motor resolution is defined by mean of *Motor_Step_Angle* and *Motor_Pole_Pairs* objects (for compatibility reasons). A value of *Motor_Resolution* object different from zero defines directly the motor resolution (*Motor_Step_Angle* object is not considered and only *Motor_Pole_Pairs* object have to be defined).

Name: **Motor_Factor (Highest sub-index supported)**
Index.Sub: **2013.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: --
Range: 4
Default Value: 4
Store Supported: No

Description: Highest sub-index supported

Notes:
- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Motor_Factor (Velocity_Factor_Numerator)**
Index.Sub: **2013.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object defines the 'Numerator' of the ratio for Velocity factor used to convert User-defined velocity unit to Internal velocity unit and vice versa.

$$Velocity\ Factor = \frac{2013.1h}{2013.2h}$$

The Internal velocity unit is ***Hz (Inc/s)***.

Notes:
- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Motor_Factor (Velocity_Factor_Denominator)**
Index.Sub: **2013.2H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object defines the 'Denominator' of the ratio for Velocity factor used to convert User-defined velocity unit to Internal velocity unit and vice versa.

$$Velocity\ Factor = \frac{2013.1h}{2013.2h}$$

The Internal velocity unit is ***Hz (Inc/s)***.

Notes:
- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Motor_Factor (Acceleration_Factor_Numerator)**
Index.Sub: **2013.3H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object defines the 'Numerator' of the ratio for Acceleration factor used to convert user-defined Acceleration (and Deceleration) unit to Internal Acceleration (and Deceleration) unit and vice versa.

$$Acceleration\ Factor = \frac{2013.3\,h}{2013.4\,h}$$

The Internal Acceleration unit is **(Inc/s²)**.

Notes:
- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Motor_Factor (Acceleration_Factor_Denominator)**
Index.Sub: **2013.4H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object defines the 'Denominator' of the ratio for Acceleration factor used to convert user-defined Acceleration (and Deceleration) unit to Internal Acceleration (and Deceleration) unit and vice versa.

$$Acceleration\ Factor = \frac{2013.3\,h}{2013.4\,h}$$

The Internal Acceleration unit is **(Inc/s²)**

Notes:
- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Preset_Homing_Position**
Index.Sub: **2080.0H**
Data Type: Integer32
Access: rw
PDO Mapping: no
Unit: User-defined position unit [(1)]
Range: Integer32
Default Value: 0
Store Supported: Yes [(2)]

Description: This object is used during the Homing mode procedure and contains the desired position value for the Homing Zero Position. All position values (objects *6062.0H*, *6063.0H*, *6064.0H*) are set to *Preset_Homing_Position* value.

Notes:
- *[(1)] See §3.2.*
- [(2)] 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Drive_Homing_Inputs_Setting**
Index.Sub: **2081.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit: --
Range: 0 ÷ FFFFFFFFH
Default Value: 00000000H
Store Supported: Yes (2)

Description: This object is used to define drive digital inputs Bank #0 allocation for *Homing_mode*. For any function can be assigned one digital input that have to be specified in the four bits of the object concerning that function.

	Bit 31÷28	Bit 27÷24	Bit 23÷20	Bit 19÷16	Bit 15÷12	Bit 11÷8	Bit 7÷4	Bit 0÷3
					Positive Limit Switch	Negative Limit Switch	Index Pulse	Home Switch
Range	**0÷F**	**0÷F**	**0÷F**	**0÷F**	**0,1,2,3,5,6,7**	**0,1,2,3,5,6,7**	**0,1,2,3,5,6,7**	**0,1,2,3,5,6,7**
Default Value					0	0	0	0

0 = B0_IN0 digital input
1 = B0_IN1 digital input
2 = B0_IN2 digital input
3 = B0_IN3 digital input
4 = B0_IN4 digital input
5 = B0_IN5 digital input
6 = B0_IN6 digital input
7 = B0_IN7 digital input

Notes:

- (1) *available with firmware V03r05 or superior.*
- (2) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Touch_Probe1_Filter**
Index.Sub: **2082.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: microseconds
Range: 0 ÷ FFFFFFFFH
Default Value: 00080008H
Store Supported: Yes (1)

Description: This object is used to set the filters value of Touch Probe1 digital input.

	Touch Probe 1 Filter	
	(Digital Input High Level)	(Digital Input Low Level)
	Bit 31÷16	Bit 15÷0
Range	**0÷65535**	**0÷65535**
Default Value	8	8

Notes:

- This object is available with firmware V01r11 or superior.
- See *§3.12*
- (1) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Touch_Probe2_Filter**
Index.Sub: **2083.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: microseconds
Range: 0 ÷ FFFFFFFFH
Default Value: 00080008H
Store Supported: Yes (1)

Description: This object is used to set the filters value of Touch Probe2 digital input.

	Touch Probe 2 Filter	
	(Digital Input High Level)	**(Digital Input Low Level)**
	Bit 31÷16	**Bit 15÷0**
Range	**0÷65535**	**0÷65535**
Default Value	8	8

Notes:

- This object is available with firmware V01r11 or superior.
- See *§3.12*
- (1) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **DS402_Working_Settings**
Index.Sub: **2084.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit: --
Range: 0 ÷ FFFFFFFFH
Default Value: 0
Store Supported: Yes (2)

Description: This object is used to parameterize DS402 working general configuration.

Bit	Name	Description	Default Value
0	PP_mode_Position_Align	0 = Not align position 1 = Align position 2 = 3 =	0
1			
2	Torque_Enable_Mode	0 = STO/Torque Enable handled as Fault 1 = STO/Torque Enable not handled as Fault	0
3	Limit_Switch_Mode	0 = Fault on Limit Switch 1 = No Fault on Limit Switch	0
4	Digital_Input_Actual_Value	0 = Filtered and polarized Digital Inputs on bit 16÷31 of (*60FD.0H*) Object 1 = Direct value of Digital Inputs on bit 16÷31 of (*60FD.0H*) Object	0
5	Execute_Homing_Offset	0 = No Homing Offset executed 1 = *Preset_Homing_Position* offset executed	0
6			0
7	Disable_Under_Voltage_Fault_in_Power_Disabled_State	0 = The under voltage is considered as a Fault in Power Disabled State. 1 = The under voltage is not considered as a Fault in Power Disabled State.	0
8			0
9			0
10			0
11			0
12			0
13			0
14			0
15			0
16			0
17			0
18			0
19			0
20			0
21			0
22			0
23			0
24			0
25			0
26			0
27			0
28			0
29			0
30			0
31			0

Bit Explanation:

PP_mode_Position_Align (bit1,bit0):

This setting is used only for *Profile_Position_Mode* and it is relevant when *Sensor_selection_code* object is different from -1 value.

When the value is (1), with motor at a standstill and before to start a movement, the *Position_demand_value* is aligned to *Position_Actual_Value*.

When the value is (0), the Position align function is not active.

Torque_Enable_Mode:

This setting is used only for drives equipped with STO or Torque Enable input.

When the value is (1), the intervention of STO/Torque Enable input is not considered a Fault and the state machine will be kept to 'Switch On Disabled' until the input is off.

When the value is (0), the intervention of STO/Torque Enable input is considered a Fault.

Limit_Switch_Mode [3] :

When the value is (1), the intervention of a Limit Switch input is not considered a Fault.

When the value is (0), the intervention of a Limit Switch input is considered a Fault.

Digital_Input_Actual_Value:

When the value is (1), the status of B0_Digital_Inputs on bits 16÷31 of *60FD.0H* object is direct, not filtered and not polarized.

When the value is (0), the status of B0_Digital_Inputs on bits 16÷31 of *60FD.0H* object is filtered and polarized.

Execute_Homing_Offset:

When the value is (1), an additional movement to a relative position is performed at the end of Homing procedure (except for *Homing_method_35* and *Homing_method_37*). The relative position is defined by mean of *Preset_Homing_Position* object (*2080.0H*). When additional movement is concluded *Position_Actual_Value* is set to 0.

When the value is (0), no additional movement is performed at the end of Homing procedure (Homing standard procedure).

Disable_Under_Voltage_Fault_in_Power_Disabled_State [4] :

When the value is (1), the under voltage is not considered as a Fault in Power Disabled State.

When the value is (0), the under voltage is considered as a Fault in Power Disabled State.

Notes:

- *[1] available with firmware V03r05 or superior.*
- *[2]* 'Store supported' is available *with firmware V02r29 or superior.*
- *[3]* See *§7.4*
- *[4] available with firmware V03r42 or superior.*
- This object is available with firmware V01r28 or superior.

Name: **Drive_Inputs_Level**
Index.Sub: **2200.1H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ FFFFH
Default Value: --
Store Supported: Yes

Description: This object is used to set drive digital inputs working level for functionality defined by mean of bit#0, bit#1 and bit#2 of *Drive_Working_Settings* object.

Bit #	Name	Description	Default Value
0	Forward_Limit_Switch_Level	1 = Active high level 0 = Active low level	0
1	Backward_Limit_Switch_Level	1 = Active high level 0 = Active low level	0
2	Fast_Stop_Level	1 = Active high level 0 = Active low level	0
3	Reserved	Reserved	--
4	Reserved	Reserved	--
5	Reserved	Reserved	--
6	Reserved	Reserved	--
7	Reserved	Reserved	--
8	Reserved	Reserved	--
9	Reserved	Reserved	--
10	Reserved	Reserved	--
11	Reserved	Reserved	--
12	Reserved	Reserved	--
13	Reserved	Reserved	--
14	Reserved	Reserved	--
15	Reserved	Reserved	--

Bit Explanation:

Forward_Limit_Switch_Level [(1)] :

When this bit is set (1) the drive will consider the *Forward Limit Switch* intervention when the corresponding digital input (see *Drive_Inputs_Setting* object) will close.
When this bit is reset (0) the drive will consider the *Forward Limit Switch* intervention when the corresponding digital input (see *Drive_Inputs_Setting* object) will open.

Backward_Limit_Switch_Level [(1)] :

When this bit is set (1) the drive will consider the *Backward Limit Switch* intervention when the corresponding digital input (see *Drive_Inputs_Setting* object) will close.
When this bit is reset (0) the drive will consider the *Backward Limit Switch* intervention when the corresponding digital input (see *Drive_Inputs_Setting* object) will open.

Fast_Stop_Level [(2)] :

When this bit is set (1) the drive will consider the *Fast Stop* input intervention when the corresponding digital input (see *Drive_Inputs_Setting* object) will close.
When this bit is reset (0) the drive will consider the Fast Stop input intervention when the corresponding digital input (see *Drive_Inputs_Setting* object) will open.

Notes:

- *[(1)] See* §7.4
- *[(2)] See* §7.5

Name: **Drive_Working_Settings**
Index.Sub: **2200.2H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: --
Range: 0 ÷ FFFFH
Default Value: --
Store Supported: Yes

Description: This object is used to parametrize drive working modalities.

Bit #	Name	Description	Default Value
0	**Forward_Limit_Switch_Check**	1 = Forward limit switch check enabled 0 = Forward limit switch check disabled	0
1	**Backward_Limit_Switch_Check**	1 = Backward limit switch check enabled 0 = Backward limit switch check disabled	0
2	**Fast_Stop_From_Input** (1)	1 = Fast Stop Input enabled 0 = Fast Stop Input disabled	0
3	Reserved	Reserved	0
4	**Feedback_Motor_Check**	1 = Motor feedback enabled (*Closed Loop*) 0 = Motor feedback disabled (*Open Loop*)	0
5	Reserved	Reserved	--
6	Reserved	Reserved	--
7	**Disable_Digital_Outputs_FW_Handling**	1 = Disable Digital_Outputs handling by firmware (All DO user free) 0 = Enable Digital_Outputs handling by firmware (Not all DO user free)	0
8	Reserved	Reserved	--
9	Reserved	Reserved	--
10	**Motor_Rotation_Direction**	1 = counter clockwise rotation when motor move forward 0 = clockwise rotation when motor move forward	0
11	Reserved	Reserved	--
12	Reserved	Reserved	--
13	Reserved	Reserved	0
14	Reserved	Reserved	--
15	Reserved	Reserved	--

Bit Explanation:

Forward_Limit_Switch_Check (3) :

When this bit is set (1), outside of the Homing modality, the drive will check continuously for the intervention of *Forward Limit Switch* (see *Drive_Inputs_Setting* and *Drive_Inputs_Level* objects). If the limit switch intervenes and the motor is running forward then the motor will stop.
If bit3 of *DS402_Working_Settings* is *0* then the *Internal_limit_active* bit and *Fault* bit of *Statusword* object will be set immediately and an emergency message will be send.
If bit3 of *DS402_Working_Settings* is 1 the intervention of forward limit switch is not consider a Fault.

When this bit is reset (0) the drive will not check for the forward limit switch at all.

Backward_Limit_Switch_Check (3) :

When this bit is set (1), outside of the homing modality, the drive will check continuously for the intervention of *Backward Limit Switch* (see *Drive_Inputs_Setting* and *Drive_Inputs_Level* objects). If the limit switch intervenes and the motor is running backward then the motor will stop.
If bit3 of *DS402_Working_Settings* is *0* then the *Internal_limit_active* bit and *Fault* bit of *Statusword* object will be set immediately and an emergency message will be send.
If bit3 of *DS402_Working_Settings* is 1 the intervention of forward limit switch is not consider a Fault.

When this bit is reset (0) the drive will not check for the backward limit switch at all.

Fast_Stop_From_Input [2]:

When this bit is set (1) the drive will check continuously for the intervention of *Fast Stop* input (see *Drive_Inputs_Setting* and *Drive_Inputs_Level* objects). If the fast stop is on the state machine is kept to 'Switch On Disabled' until the input goes off.

When this bit is reset (0) the drive will not check for the fast stop input at all.

Disable_Digital_Outputs_FW_Handling:

When this bit is set (1) the firmware will not set/reset automatically the digital outputs assigned to Fault and Ready/Busy functions. Whole digital outputs are available to the user.

When this bit is reset (0) the firmware will set/reset automatically the digital outputs assigned to Fault and Ready/Busy functions. The remaining digital outputs are available to the user.

Motor_Rotation_Direction:

When this bit is set (1) the motor will rotate counter clockwise when the motion direction is forward (*Position_Actual_Value* object increase) and clockwise when the motion direction is backward (*Position_Actual_Value* object decrease).

When this bit is reset (0) the drive motor will rotate clockwise when the motion direction is forward (*Position_Actual_Value* object increase) and counter clockwise when the motion direction is backward (*Position_Actual_Value* object decrease).

Feedback_Motor_Check:

When this bit is set (1) the drive will enable *Closed Loop* feature to close the motion control loop. This will optimize the motor efficiency.

When this bit is reset (0) the *Closed Loop* feature is disabled and *Open Loop* feature is enabled.

Notes:

- *[1] available with firmware V03r05 or superior.*
- [2] available *with firmware* release V02r63 or superior. See *§7.5*
- [3] See *§7.4*

Name: **Drive_Inputs_Setting**
Index.Sub: **2200.3H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ FFFFFFFFH
Default Value: **--**
Store Supported: Yes

Description: This object is used to define drive digital inputs Bank #0 allocation used for functionality defined by mean of bit#0, bit#1 and bit#2 of *Drive_Working_Settings* object. For any function can be assigned one digital input that have to be specified in the four bits of the object concerning that function.

	Bit 31÷28	Bit 27÷24	Bit 23÷20	Bit 19÷16	Bit 15÷12	Bit 11÷8	Bit 7÷4	Bit 3÷0
						Fast Stop	Backward Limit Switch	Forward Limit Switch
Range						**0÷7**	**0÷7**	**0÷7**
Default Value	0	0	0	0	0	2	1	0

0 = B0_IN0 digital input
1 = B0_IN1 digital input
2 = B0_IN2 digital input
3 = B0_IN3 digital input
4 = B0_IN4 digital input
5 = B0_IN5 digital input
6 = B0_IN6 digital input
7 = B0_IN7 digital input

Notes: See *§7.4* and *§7.5*

Name: **Drive_CANopen_Config**
Index.Sub: **2200.5H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0000H ÷ FFFFH
Default Value: 0000H
Store Supported: Yes

Description: This object selects and actives the RX_PDO and TX_PDO Mapping selecting in a Static Mapping list.

	Bit 15÷12	Bit 11÷8	Bit 7÷4	Bit 0÷3
	RX_PDO1 Config	TX_PDO1 Config	RX_PDO2 Config	TX_PDO2 Config
Default Value	0	0	4	1

TX_PDOx Config:
00 → Mapping #0
01 → Mapping #1
02 → Mapping #2
……

RX_PDOx Config:
00 → Mapping #0
01 → Mapping #1
02 → Mapping #2
…….

Notes:
- This object is not available in EtherCAT fieldbus.
- '**RX_PDO1 Config**' and '**TX_PDO1 Config**' are available with firmware V00r67 or superior.
- See *§2.4.1*

Name: **Analog_Input[0]_K_Filter**
Index.Sub: **2200.6H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: **--**
Range: 0 ÷ 10000
Default Value: --
Store Supported: Yes

Description: It contains the value of the K constant used for filtering the *Analog_In[0]*. The formula for filtering the analog inputs is the following:

$$Analog\_Input\_Filtered_{(n)} = Analog\_Input\_Filtered_{(n-1)} + \frac{(Analog\_Input_{(n)} - Analog\_Input\_Filtered_{(n-1)})}{Analog\_Inputs\_K\_Filter}$$

High value of K lead to a more filtered analog input value. With K = 1 the filter is disabled and the *Analog_In[0]* object returns the instant value of analog input #0 without any filtering. With K = 0 the analog input #0 is disabled.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **B0_Digital_Inputs_Polarity**
Index.Sub: **2200.7H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: **--**
Range: 0 (all inputs have normal polarity) ÷ 2^n-1 (all inputs have inverted polarity, where n is the number of digital inputs available)
Default Value: 0
Store Supported: Yes

Description: It contains the polarity of all the digital inputs on Bank #0 of the drive. When the polarity is normal, the input status (B0_Digital_Inputs) is 1 if the voltage is supplied to the input and 0 if no voltage is supplied to the input. When the polarity is inverted, the input status (B0_Digital_Inputs) is 0 if the voltage is supplied to the input and 1 if no voltage is supplied to the input.

Bit #	Input	Description	Default Value
0	B0_IN0	1 = inverted polarity 0 = normal polarity	0
1	B0_IN1	1 = inverted polarity 0 = normal polarity	0
2	B0_IN2	1 = inverted polarity 0 = normal polarity	0
3	B0_IN3	1 = inverted polarity 0 = normal polarity	0
n	B0_INn	1 = inverted polarity 0 = normal polarity	0

Notes:

- *(1) available with firmware V03r05 or superior.*

Name: **Analog_Input[0]_Type**
Index.Sub: **2200.EH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ; 1

Value	Description
0	Differential ±10V
1	Potentiometer

Default Value: --
Store Supported: No

Description: This object contains Analog Input[0] Type.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Analog_Input[1]_Type**
Index.Sub: **2200.FH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ; 1

Value	Description
0	Differential ±10V
1	Potentiometer

Default Value: --
Store Supported: No

Description: This object contains Analog Input[1] Type.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Analog_Input1_K_Filter**
Index.Sub: **2200.10H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: **--**
Range: 0 ÷ 10000
Default Value: --
Store Supported: Yes

Description: It contains the value of the K constant used for filtering the *Analog_In[1]*. The formula for filtering the analog inputs is the following:

$$Analog\_Input\_Filtered_{(n)} = Analog\_Input\_Filtered_{(n-1)} + \frac{(Analog\_Input_{(n)} - Analog\_Input\_Filtered_{(n-1)})}{Analog\_Inputs\_K\_Filter}$$

High value of K lead to a more filtered analog input value. With K = 1 the filter is disabled and the *Analog_In[1]* object returns the instant value of analog input #1 without any filtering. With K = 0 the analog input #1 is disabled.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Drive_Working_Settings_Extended**
Index.Sub: **2200.11H**
Data Type: WORD
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ FFFFH
Default Value: 0
Store Supported: Yes

Description: This register is used to define drive working modalities.

Bit #	Name	Description	Default Value
0	Reserved	Reserved	--
1	Reserved	Reserved	--
2	Reserved	Reserved	--
3	Reserved	Reserved	--
4	Reserved	Reserved	--
5	Reserved	Reserved	--
6	Reserved	Reserved	**0**
7	**Encoder1_Rotation_Direction**	**1 = The encoder #1 counting is reversed** **0 = The encoder #1 counting is not reversed**	**0**
8	Reserved	Reserved	--
9	**Motor R,L detection**	**1 = The motor R,L parameters are automatically detected** **0 = The motor R,L parameters are defined by objects**	**1**
10	**Enable I2T protection**	**1 = I2T protection enabled** **0 = I2T protection disabled**	**0**
11	**Continuous motor Open Phase Check**	**1 = Continuous motor Open Phase Check enabled** **0 = Continuous motor Open Phase Check disabled**	**0**
12	Reserved	Reserved	--
13	**Motor Stall detection**	**1 = Motor Stall detection enabled** **0 = Motor Stall detection disabled**	**0**
14	Reserved	Reserved	--
15	Reserved	Reserved	--

Bit Explanation:

Encoder1_Rotation_Direction :

When this bit is set (1) the Encoder #1 counting is reversed.
When this bit is reset (0) the Encoder #1 counting is not reversed.

Motor R,L detection :

When this bit is set (1) the motor R,L parameters are detected by automatic calbration. For this procedure the object *2005.4H* must be set with *Nominal_Current* of the motor.
When this bit is reset (0) the motor R,L parameters are defined by mean of objects *2005.6H* and *2005.7H*.

Enable I2T protection :

When this bit is set (1) I2T protection is enabled (see *§7.2*).
When this bit is reset (0) I2T protection is disabled.

Continuous motor Open Phase Check :

When this bit is set (1) the motor Open Phase Check is always active.
When this bit is reset (0) the motor Open Phase Check is active only first time at power on of the drive.

Motor Stall detection [(1)] :

When this bit is set (1) the 'Motor Stall detection' feature is enabled.
When this bit is reset (0) the 'Motor Stall detection' feature is disabled.

Notes:
- [(1)] The 'Motor Stall detection' feature is available with firmware version V02r74 or superior.

Name: **B1_Digital_Inputs_Polarity**
Index.Sub: **2200.16H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: **--**
Range: 0 (all inputs have normal polarity) ÷ 2^n-1 (all inputs have inverted polarity, where n is the number of digital inputs available)
Default Value: 0

Bit #	Input	Description	Default Value
0	B1_In0	1 = inverted polarity 0 = normal polarity	0
1	B1_In1	1 = inverted polarity 0 = normal polarity	0
2	B1_In2	1 = inverted polarity 0 = normal polarity	0
3	B1_In3	1 = inverted polarity 0 = normal polarity	0
n	B1_Inn	1 = inverted polarity 0 = normal polarity	0

Store Supported: Yes

Description: It contains the polarity of all the digital inputs on bank 1 of the drive. When the polarity is normal, the input status (B1_Digital_Inputs) is 1 if the voltage is supplied to the input and 0 if no voltage is supplied to the input. When the polarity is inverted, the input status (B1_Digital_Inputs) is 0 if the voltage is supplied to the input and 1 if no voltage is supplied to the input.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Drive_CANopen_Config_Extended**
Index.Sub: **2200.1AH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0000H ÷ FFFFH
Default Value: 0000H
Store Supported: Yes

Description: This object selects and actives the RX_PDO and TX_PDO Mapping selecting in a Static Mapping list.

	Bit 15÷12	Bit 11÷8	Bit 7÷4	Bit 0÷3
	RX_PDO3 Config	**TX_PDO3 Config**	**RX_PDO4 Config**	**TX_PDO4 Config**
Default Value	0	0	0	0

TX_PDOx Config:
00 → Mapping #0
01 → Mapping #1
02 → Mapping #2
……

RX_PDOx Config:
00 → Mapping #0
01 → Mapping #1
02 → Mapping #2
…….

Notes:
- This object is not available in EtherCAT fieldbus.
- This object is available with firmware V00r70 or superior.
- See *§2.4.1*

Name: **Encoder_Frequency[0÷1]**
Index.Sub: **2211.1H,2211.2H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: Inc/s
Range: **--**
Default Value: 0
Store Supported: No

Description: This object contains the Encoder # 0 and #1 frequency.

Notes:

- The number of encoders available depends on the version of the drive currently in use.

Name: **Homing_On_Mechanical_Block_Current_Limit**
Index.Sub: **2220.07H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 1500
Store Supported: Yes

Description: This value defines the maximum motor current can be supplied to the motor during the Homing movement on Mechanical Block to limit the motor torque. It is used only if the *Closed Loop* feature is enabled and with *Homing_method* -1, -2, -3, -4.

Notes:

- the object is available with firmware V03r74 or superior.
- See §3.5

Name: **Homing_On_Mechanical_Block_Position_Error_Limit**
Index.Sub: **2220.09H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: IU (one motor turn = 65536 IU)
Range: Unsigned32
Default Value: 5000
Store Supported: Yes

Description: This value defines the maximum allowed displacement between motor reference position and motor actual position detected by Feedback Sensor during the Homing movement on Mechanical Block.
It is used only if the *Closed Loop* feature is enabled and with *Homing_method* -1, -2, -3, -4.

Notes:

- the object is available with firmware V03r74 or superior.
- See §3.5

Name: **Master_Encoder_Filter_Time**
Index.Sub: **2221.04H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: us
Range: 0 ÷ 20000
Default Value: 2000
Store Supported: Yes

Description: This value defines the filter value applied to Master Position Reference.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Feedback_Position_Error_Limit**
Index.Sub: **2230.01H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: IU (one motor turn = 65536 IU)
Range: Unsigned32
Default Value: 10000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the maximum allowed displacement between motor reference position and motor actual position detected by Feedback Sensor .

Notes:

Name: **Feedback_Actual_Position_Error**
Index.Sub: **2230.02H**
Data Type: Integer32
Access: ro
PDO Mapping: no
Unit: IU (one motor turn = 65536 IU)
Range: Integer32
Default Value: --
Store Supported: No

Description: This value is used for *Closed Loop* feature and returns the actual position displacement between motor reference position and motor actual position detected by Feedback Sensor.

Notes:

Name: **Feedback_Source_PPR**
Index.Sub: **2230.03H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: Inc
Range: Unsigned32
Default Value: 1600
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines pulses per evolution of Encoder installed on motor rear shaft.

Notes:

Name: **Feedback_Kp**
Index.Sub: **2230.04H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 20000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the position gain of PID's regulator. The value has to be adjust to keep the position error (*Feedback_Actual_Position_Error*) inside the tolerance of application. The value can be set to 0 value when is necessary to have only a velocity controller and has not to be considered the position error.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Kv**
Index.Sub: **2230.05H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 8000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the velocity gain of PID's regulator. The value has to be adjust to keep the velocity error (*Feedback_Actual_Velocity_Error*) inside the tolerance of application.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Kffw_Acc**
Index.Sub: **2230.06H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 14000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the gain used to calculate the Feedforward current while the motor is accelerating forward. The value has to be adjust to reduce position error (*Feedback_Actual_Position_Error*) and velocity error (*Feedback_Actual_Velocity_Error*) during acceleration in forward direction.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Kffw_Dec**
Index.Sub: **2230.07H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 14000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the gain used to calculate the Feedforward current while the motor is decelerating forward. The value has to be adjust to reduce position error (*Feedback_Actual_Position_Error*) and velocity error (*Feedback_Actual_Velocity_Error*) during deceleration in forward direction.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Ki**
Index.Sub: **2230.0BH**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 1000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the integral gain of PID's regulator. The value has to be adjust to reduce the position error (*Feedback_Actual_Position_Error*) or velocity error (*Feedback_Actual_Velocity_Error*) at the end of acceleration ramp or deceleration ramp.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Ki_Limit**
Index.Sub: **2230.0CH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 1500
Store Supported: Yes

Description: This value is used for *Closed Loop* feature to limit PID's integral (anti-windup).

Notes:

Name: **Feedback_Kalfas**
Index.Sub: **2230.0DH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: Yes

Description: This value is used for *Closed Loop* feature. It is used to change the advance angle depending by speed.

Notes:
- For speed value higher than 1500 rpm set value equal to 60000.

Name: **Feedback_Current_Filter_Time**
Index.Sub: **2230.0EH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: us
Range: 0 ÷ 10000
Default Value: 100
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the filter applied to output motor Current (Iq_out).

A values range from 100 to 500 are a good choice. If it's necessary a quicker velocity reaction decrease the value, if it is necessary to reduce the motor noise at very low speed increase the value.

Notes:

Name: **Feedback_Iq_min**
Index.Sub: **2230.0FH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: mA
Range: Unsigned16
Default Value: 500
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines minimum amplitude of the phase's current. It can be considered also as minimum current applied at the motor.

Notes:

Name: **Feedback_Boost_Current**
Index.Sub: **2230.12H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 1500
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines maximum current supplied to the motor. The parameter is also used to limit the motor torque.

Notes:

- *(1) available with firmware V00r75 or superior.*

Name: **Feedback_Kfbw_Acc**
Index.Sub: **2230.15H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 14000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the gain used to calculate the Feedforward current while the motor is accelerating backward. The value has to be adjust to reduce position error (*Feedback_Actual_Position_Error*) and velocity error (*Feedback_Actual_Velocity_Error*) during acceleration in backward direction.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Kfbw_Dec**
Index.Sub: **2230.16H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned32
Default Value: 14000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the gain used to calculate the Feedforward current while the motor is decelerating backward. The value has to be adjust to reduce position error (*Feedback_Actual_Position_Error*) and velocity error (*Feedback_Actual_Velocity_Error*) during deceleration in backward direction.

Notes:

- A value too high could introduce motor's vibration.

Name: **Feedback_Settings**
Index.Sub: **2230.17H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: Yes

Description: This value defines settings regarding the *Closed Loop* functionality.

Bit	Value	Description
0÷3	**Feedback Mode**	0 = Reserved 1,2 = Position Feedback Mode 3,4 = Velocity Feedback Mode 5 = Torque Feedback Mode **Note :** This parameter is forced directly from *Modes of operation* used and then it is not necessary to be set.
4	**Reserved**	
5	**Reserved**	
6	**Kff_Switch**	Define the managing of the feedforward current.
7	**Calibration Mode**	0 = Feedback Calibration Procedure is executed each time the Closed Loop has to be activated (for example after the power is supplied or after an Reset Alarm that disabled the closed loop).
		1 = Feedback Calibration procedure is executed only a first time the power is supplied or when happen an Alarm that require the Feedback Calibration (for instance encoder misalignment).
8÷11	**Sensor Type**	0 = Incremental encoder (Enc#0)
		1 = Hall sensor
		2 = Incremental encoder (Enc#0) + Hall sensors
		3 = Incremental encoder (Enc#0) + Zero enc
		4 = Multi-Turn Absolute encoder multi turn SSI
		5 = Multi-Turn encoder SSI + incremental encoder
		6 = Multi-Turn Absolute encoder BISS
		7 = Multi-Turn Absolute encoder BISS + incremental encoder
		8 = Single-Turn Magnetic encoder
		9 to 15 Reserved
12	**Calibration Type**	0 = Full Feedback Calibration procedure.
		1 = Light Feedback Calibration procedure.
13	**Torque Limit Speed Enable** [1]	0= Torque Limit Speed disabled
		1= Torque Limit Speed enabled
14	**Absolute Encoder Calibrated** [2]	0 = The Sensor Calibration procedure for Absoluter encoder BiSS is performed every time.
		1 = The calibrated values in NVRAM for Absoluter encoder BiSS can be used. No movement of the motor are performed. **Note :** This bit was introduced for compatibility reasons with previous firmware versions (the default value of this bit is equal to 0) and is used only for 'Sensor Type = 6'.
15	**Not used**	

Notes:

- [1] This function is available only with firmware version V02r82 or superior.
- [2] This function is available only with firmware version V03r21 or superior.

Name: **Feedback_Status**
Index.Sub: **2230.18H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: Yes

Description: This is used for *Closed Loop* functionality and returns the Feedback Status.

Bit	Value	Description
0	**Calibration in Progress**	0= Feedback Calibration NOT in progress.
		1= Feedback Calibration in progress.
1	**Calibration_Error**	0=Ok
		1= Feedback Calibration Procedure failed
2	**Encoder Direction Error**	0=Ok
		1= The bit is set during the Calibration Procedure, if the system detected that the encoder is counting in the wrong sense. The Encoder Phases has to be exchanged.
3	**Encoder Not Present**	0=Ok
		1= The bit is set during the Calibration Procedure, if the system detected that the encoder is not present. Check if the encoder wiring is correct.
4	**Hall Sensor Fail**	0=Ok
		1= The bit is set during the Calibration Procedure, if the system detected that the Hall sensors are not working well. Check if the Hall sensor wiring is correct
5	**Calibration Aborted**	0=Ok
		1= Calibration Procedure aborted.
6	**Torque Limit Speed Reached** [2]	0= Torque limit speed reached.
		1= Torque limit speed not reached.
7	**Reserved**	
8	**Encoder Fault**	0=Ok
		1= Detected problem on feedback encoder. The reasons can be:: - Encoder disconnected or broken. - The encoder is counting in reverse direction respect at the expected direction for a time more of TIMEOUT(encoder misalignment, check the encoder wiring or the mechanical mounting of it).
9	**Following Error**	0 = OK
		1= Feedback Following Error (Closed Loop).
10	**Motor Stall detected** [1]	0 =OK
		1= Motor Stall detected. This Error can be issued when 'Motor Stall' feature is enabled (bit#13 of *Drive_Working_Settings_Extended* object) and motor stall is detected. See *§7.3*
11	**GAIN out of range**	0=OK
		1= GAIN out of range. The bit is set if the system detected that the set Feedback GAIN sent the PID regulator out of control
12	**Calibration attained**	0 = Calibration not attained
		1 = Calibration attained
13	**Absolute Encoder Error**	0 = Ok
		1 = Absolute Encoder Error
14	**Motor Stall Conflict** [1]	0=OK

Bit	Value	Description
		1= Error : tried to enable 'Feedback' feature with 'Motor Stall detection' feature already enabled or vice versa. The 'Feedback' feature (bit#4 of *Drive_Working_Settings* object) and the 'Motor Stall detection' feature (bit#13 of *Drive_Working_Settings_Extended object)* cannot be both active at the same time. See *§7.3*
15	**Active**	0= Feedback not active
		1= Feedback Active (the system is working in *Closed loop*).

Notes :

- (1) This bit is available only with firmware version V02r74 or superior.
- (2) This bit is available only with firmware version V02r82 or superior.

Name: **Feedback_Limit_Speed**
Index.Sub: **2230.1BH**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO [1]
Unit: rpm
Range: 0 ÷ 4000
Default Value: 3000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the maximum motor speed.

Notes:

- [1] *available with firmware V03r07 or superior.*

Name: **Feedback_Calibration_Current**
Index.Sub: **2230.1CH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: mA
Range: 0 ÷ Motor Max Current
Default Value: 1000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines motor Current value used during the Feedback Calibration procedure.

Notes:

Name: **Feedback_Encoder_Filter_Time**
Index.Sub: **2230.1DH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: us
Range: 0 or 50 ÷ 20000
Default Value: 900
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the filter value applied to Feedback sensor speed.

Value	Description
0	Automatic filter activation
50÷20000	Valid filter value from 50 us to 20000 us

Generally a value from 500 to 2000 are a good choice. If it's necessary a quicker velocity reaction decrease the value, if it is necessary to reduce the motor noise at very low speed increase the value.

Notes:

Name: **Feedback_Calibration_Phase**
Index.Sub: **2230.19H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: 0 ÷ 50
Default Value: 0
Store Supported: No

Description: This object returns the feedback calibration phase during the Feeback Calibration Procedure (*Closed Loop*).

Value	Description
0	Feedback disactive
1÷49	Feedback calibration in execution
65535	Feedback actived

Notes:

Name: **Feedback_Calibration_Speed**
Index.Sub: **2230.1AH**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: 0.01 rpm
Range: 0 ÷ 5000
Default Value: 500
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines motor Speed value used during the Feedback Calibration procedure.

Notes:

Name: **Feedback_Velocity_Error_Limit**
Index.Sub: **2230.1EH**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: 0.01 rpm
Range: Unsigned32
Default Value: 1000
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines the maximum allowed displacement between motor reference velocity and motor actual velocity detected by Feedback Sensor .

Notes:

Name: **Feedback_Actual_Velocity_Error**
Index.Sub: **2230.1FH**
Data Type: Integer32
Access: ro
PDO Mapping: no
Unit: IU/sec (one motor turn = 65536 IU)
Range: Integer32
Default Value: --
Store Supported: No

Description: This value is used for *Closed Loop* feature and returns the actual velocity displacement between motor reference velocity and motor actual velocity detected by Feedback Sensor.

Notes:

Name: **Feedback_Iq_Min_Threshold**
Index.Sub: **2230.21H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: 0.01 rpm
Range: Unsigned32
Default Value: 18
Store Supported: Yes

Description: This value is used for *Closed Loop* feature and defines limit speed value under which the *Feedback_Iq_min* object is enabled to work.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Hall_Sensors_Status**
Index.Sub: **2900.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: 0 ÷ 7
Default Value: 0
Store Supported: No

Description: This object returns state of Hall Sensors:

A	B	C	Status	Note
0	0	0	0	Not valid
0	0	1	1	
0	1	0	2	
0	1	1	3	
1	0	0	4	
1	0	1	5	
1	1	0	6	
1	1	1	7	Not valid

The Status 0 and Status 7 are not valid combinations. One of these two status means issues on Hall Sensors connection and / or issues concerning Hall Sensors.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **Hall_Sensors_Position**
Index.Sub: **2901.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: 0.1 degree
Range: 0,600,1200,1800,2400,3000
Default Value: 0
Store Supported: No

Description: This object returns position of Hall Sensors.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **I2T_TMax_Peak_Current**
Index.Sub: **2902.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: milliseconds
Range: 0 ÷ 4294967295
Default Value: 1000
Store Supported: Yes

Description: This value is used for ***I^2T Protection*** functionality and specifies the maximum duration of the peak current (*I2T_Peak_Current* object).

Notes: - This object is not available in EtherCAT fieldbus.
- See *§7.2*

Name: **I2T_Current_Limit**
Index.Sub: **2908.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: $0.1A^2 \cdot sec$
Range: 0 ÷ 65535
Default Value: --
Store Supported: Yes

Description: This value is used for ***I^2T Protection*** functionality and returns the I2T current limit value :

$$I2T\_Current\_Limit = [\ (I2T\_Peak\_Current)^2 - (Nominal\_Current)^2\ ] \cdot I2T\_Tmax\_Peak\_Current$$

Notes: - This object is not available in EtherCAT fieldbus.
- See *§7.2*

Name: **I2T_Current_Actual**
Index.Sub: **2909.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: $0.1A^2 \cdot sec$
Range: 0 ÷ 65535
Default Value: --
Store Supported: No

Description: This value is used for ***I^2T Protection*** functionality and returns the I2T current actual :

$I2T\_Current\_Actual \mathrel{+}= [\ (Current\_Actual\_Value)^2 - (Nominal\_Current)^2\ ] \cdot Ts$

Ts = 50 µsec (Sampling time)

Notes: - This object is not available in EtherCAT fieldbus.
- See *§7.2*

Name: **Hall_Sensors_Sequence_Settings**
Index.Sub: **290A.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: --
Default Value: 326451h
Store Supported: Yes

Description: This value defines Hall Sensors sequence.

Notes:

Name: **I2T_Peak_Current**
Index.Sub: **290C.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: mA
Range: 0 ÷ Motor Max Current
Default Value: 0
Store Supported: Yes

Description: This value is used for ***I^2T Protection*** functionality and specifies the motor peak current (rms).

Notes: - This object is not available in EtherCAT fieldbus.
- See *§7.2*

Name: **Hall_Sensors_Sequence_Detected**
Index.Sub: **290D.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: 0
Store Supported: No

Description: This value returns Hall Sensors sequence detected after that Hall Sensors Procedure is done.

Notes: - This object is not available in EtherCAT fieldbus.

Name: **BiSS_Encoder_Actual_Value**
Index.Sub: **2A00.0H**
Data Type: Signed32
Access: ro
PDO Mapping: TX_PDO (1)
Unit: Increments (one motor turn = 65536 Inc)
Range: -2147483648 ÷ 2147483647
Default Value: 0
Store Supported: No

Description: This object contains the BiSS_Encoder_Internal_Value normalized to 32 bits and subtracted by the *BiSS_Encoder_Offset_Value*.

BiSS_Encoder_Actual_Value = *BiSS_Encoder_Internal_Value* - *BiSS_Encoder_Offset_Value*

Since the BiSS_Encoder_Internal_Value cannot be changed, it is possible to get the desired value acting on the *BiSS_Encoder_Offset_Value*. For Instance if a particular position motor position have to become the 0 position, just set the *BiSS_Encoder_Offset_Value* = BiSS_Encoder_Internal_Value .

Notes: - This object is not available in EtherCAT fieldbus.
- This object is available *with firmware V02r76 or superior.*
- *(1) available with firmware V02r76 or superior.*

Name: **BiSS_Encoder_Status**
Index.Sub: **2A02.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: 0 ÷ 65535
Default Value: 0
Store Supported: No

Description: This object contains the value of diagnostic bits of BiSS Encoder received communication frame. If the Encoder is correctly working the value of this object should be always equal to 3 otherwise there are some communication errors with the Encoder or the Encoder is faulty.

Bit #	Description
0	1 = Ok, Good 0 = Warning Condition
1	1 = Ok, Data Valid 0 = Error (HW Failure detected)
2 ÷ 15	Reserved

Notes: - This object is not available in EtherCAT fieldbus.
- This object is available *with firmware V02r76 or superior.*

Name: **BiSS_Encoder_Config**
Index.Sub: **2A04.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ FFFFFFFFH
Default Value: 0 (BiSS Encoder disabled)
Store Supported: Yes

Description: This object contains the configuration for the BiSS Encoder. For the correct reading of the BiSS Encoder connected to the drive it is necessary to specify the right resolution (# of bits of either single turn or multiturn). For Instance if the connected BiSS Encoder has a resolution of 17 bits single turn and 16 bits multiturn, The *BiSS_Encoder_Config* object must be set equal to 4113 (1011H).

Byte 3	Byte 2	Byte 1	Byte 0
Reserved	Reserved	Multiturn Bits #	SingleTurn Bits #

Notes: - This object is not available in EtherCAT fieldbus.
- This object is available with BiSS Encoder input and only *with firmware V02r76 or superior.*

Name: **BiSS_Encoder_RxErr**
Index.Sub: **2A06.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ 65535
Default Value: 0
Store Supported: No

Description: This object contains the number of BiSS Encoder receive errors. This object should be always 0 otherwise there are some communication errors with the Encoder. Check the cabling and the cable length to solve communication errors. The error counter can be cleared setting it to 0.

Notes: - This object is not available in EtherCAT fieldbus.
- This object is available *with firmware V02r76 or superior.*

Name: **BiSS_Encoder_Offset_Value**
Index.Sub: **2A08.0H**
Data Type: Signed32
Access: rw
PDO Mapping: no
Unit: Increments (one motor turn = 65536 Inc)
Range: -2147483648 ÷ 2147483647
Default Value: 0
Store Supported: Yes

Description: This object contains the offset used to calculate the *BiSS_Encoder_Actual_Value* object.

Notes: - This object is not available in EtherCAT fieldbus.
- This object is available *with firmware V02r76 or superior.*

Name: **BiSS_Encoder_Internal_Value**
Index.Sub: **2A0A.0H**
Data Type: Signed32
Access: ro
PDO Mapping: no
Unit: Increments (one motor turn = 65536 Inc)
Range: -2147483648 ÷ 2147483647
Default Value: --
Store Supported: No

Description: This object contains the actual BiSS Encoder value normalized to 32 bits. Regardless of the Encoder multiturn and singleturn resolution, this object contains in the low word the single turn position and in the high word the number of turns.

Notes:
- This object is not available in EtherCAT fieldbus.
- This object is available *with firmware V02r76 or superior.*

Name: **Torque_window**
Index.Sub: **2B08.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: ‰ Nominal_Current
Range: 0 ÷ (‰ Max drive current)
Default Value: 0
Store Supported: Yes

Description: This object is used for *Profile_Torque_mode* (tq) and specifies a symmetrical range relative to the *Target_Torque* within the Target is considered having been met. If the value is set to "FFFFh" monitoring is switched off, the "Target reached" bit of *Statusword* is never set.

The *Target_Torque* (*6071.0H*) value is reached if *Torque_actual_value* (*6077.0H*) is inside *Torque_window* (*2B08.0H*) for *Torque_window_time* (*2B09.0H*).

Notes:
- *This object is available with firmware version V02r82 or superior.*
- *(1) available with firmware V03r03 or superior.*

Name: **Torque_window_time**
Index.Sub: **2B09.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: milliseconds
Range: 0 ÷ 65535
Default Value: 0
Store Supported: Yes

Description: This object is used for *Profile_Torque_mode* (tq) and defines the time used to consider the *Target_Torque* value reached.

The *Target_Torque* (*6071.0H*) value is reached if *Torque_actual_value* (*6077.0H*) is inside *Torque_window* (*2B08.0H*) for *Torque_window_time* (*2B09.0H*).

Notes:
- *This object is available with firmware version V02r82 or superior.*
- *(1) available with firmware V03r03 or superior.*

Name: **Braking_Resistor_Value**
Index.Sub: **2B80.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: Ohm
Range: 0 ÷ 65535
Default Value: 50
Store Supported: Yes

Description: This object defines the ohmic value of the braking resistor.

Notes: *- This object is available with firmware version V03r18 or superior.*
- See §7.7

Name: **Braking_Resistor_Power**
Index.Sub: **2B81.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: Watt
Range: 0 ÷ 65535
Default Value: 50
Store Supported: Yes

Description: This object defines the rated power of the braking resistor.

Notes: *- This object is available with firmware version V03r18 or superior.*
- See §7.7

Name: **Braking_Threshold_ON**
Index.Sub: **2B82.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: Volt
Range: 0 ÷ 65535
Default Value: 380
Store Supported: Yes

Description: This object defines the DC bus voltage threshold above which the braking resistor is activated.

Notes: *- This object is available with firmware version V03r18 or superior.*
- See §7.7

Name: **Braking_Threshold_OFF**
Index.Sub: **2B83.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: Volt
Range: 0 ÷ 65535
Default Value: 370
Store Supported: Yes

Description: This object defines the DC bus voltage threshold below which the braking resistor is deactivated.

Notes: *- This object is available with firmware version V03r18 or superior.*
- See §7.7

Name: **Braking_Resistor_Overload_Time**
Index.Sub: **2B84.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: 0.1ms
Range: 0 ÷ 65535
Default Value: 0
Store Supported: Yes

Description: This object defines the maximum time that the braking resistor can withstand the peak of the power.

Notes: *- This object is available with firmware version V03r18 or superior.*
- See §7.7

Name: **Brake_Control_Settings**
Index.Sub: **2C00.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ 65535
Default Value: 0
Store Supported: Yes

Description: This object defines settings regarding the *Brake Control* (*§7.6*) functionality.

Bit #	Name	Description	Default Value
0	**Automatic Brake Handling**	0 = Disabled	0
		1 = Enabled	0
1	**Brake Control Type**	0 = Type 0	0
		1 = Type 1	0
2	**Brake Digital Output use Mode**	0 = Mode 0	0
		1 = Mode1	0
3	**Brake Digital Output Bank**	0 = The Brake Digital Output is one of digital outputs of Bank #0	0
		1 = The Brake Digital Output is one of digital outputs of Bank #1	0
4÷7	**Brake Digital Output**	Digital output used for the Brake	0
8÷15	**Reserved**	Reserved	0

Bit Explanation:

Automatic Brake Handling :

When this bit is set (1) the Automatic Brake Handling is enabled.
The digital output, defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object, is used for Brake Handling according to the rules defined by bit1 and bit2 of of *Brake_Control_Settings* object.

When this bit is reset (0) the Automatic Handling is disabled and settings of bit1÷bit7 are not considered.

Brake Control Type :

Brake Control Type = 0

The Brake is close (active) :

- if the drive is in the '*Switched On*' state or *'Ready to switch on'* state or *'Switch on disabled'* state or *'Not ready to switch on'* state or *'Fault'* state.
- if the drive is in the *'Operation enabled'* state and bit2 of *Brake_Control_Settings* object is equal to 1 and the digital output (used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object) is reset by *B0_Digital_Outputs* object (if bit3 = 0 of *Brake_Control_Settings* object) *or* *B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object*)*.

The Brake is open (released) :

- if the drive is in the *'Operation enabled'* state and bit2 of *Brake_Control_Settings* object is equal to 0.
- if the drive is in the *'Operation enabled'* state and bit2 *of* *Brake_Control_Settings* object is equal to 1 and the digital output (used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object) is set by *B0_Digital_Outputs* object (if bit3 = 0 of *Brake_Control_Settings* object*)* or *B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object*)*.

Brake Control Type = 1

The Brake is close (active) :

- if the drive is in the *'Switched On'* state *or 'Ready to switch on'* state *or 'Switch on disabled'* state *or 'Not ready to switch on'* state *or 'Fault' state.*
- if the drive is in the *'Operation enabled'* state and bit2 of *Brake_Control_Settings* object is equal to 1 and the digital output (used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object) is reset by *B0_Digital_Outputs* object (if bit3 = 0 of *Brake_Control_Settings* object) *or B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object*)*.
- if the drive is in the *'Operation enabled'* state and the motor is at standstill and bit2 of *Brake_Control_Settings* object is equal to 0.

The Brake is open (released) :

- if the drive is in the *'Operation enabled'* state and bit2 *of Brake_Control_Settings* object is equal to 0 and the motor is running.
- if the drive is in the *'Operation enabled'* state and bit2 *of Brake_Control_Settings* object is equal to 1 and the digital output (used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object) is set by *B0_Digital_Outputs* object (if bit3 = 0 of *Brake_Control_Settings* object*)* or *B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object*)*.

Brake Digital Output use Mode :

When this bit is set (1), according to the Brake Control Type (bit1 of *Brake_Control_Settings* object), the state of Digital Output (used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object), can be changed by *B0_Digital_Outputs* object (if bit3 = 0 of *Brake_Control_Settings* object) *or B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object).

When this bit is reset (0) the state of Digital Output (used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object) can not be changed by *B0_Digital_Outputs object* (if bit3 = 0 of *Brake_Control_Settings* object) *or B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object).

Brake Digital Output Bank :

When this bit is set (1) the Digital Output, used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object, is related to digital outputs of Bank#1.

When this bit is set (0) the Digital Output, used for the Brake and defined by bit4÷bit7 and bit3 of *Brake_Control_Settings* object, is related to digital outputs of Bank#0.

Brake Digital Output :

The bit4÷bit7 define Digital output used for the Brake. It can be related to digital outputs of Bank#0 (if bit3 = 0 of *Brake_Control_Settings* object) or Bank#1 (if bit3 = 1 of *Brake_Control_Settings* object).

If the Digital Output used for the Brake is B0_OUT0 (digital output #0 of Bank#0) or B0_OUT1 (digital output #1 of Bank#0) is necessary to set (1) the bit7 of *Drive_Working_Settings* object.

Notes:

- This object is available with firmware V03r20 or superior.
- See §7.6
- See *2C01.0H*, *2C02.0H*, *2C03.0H*, *2C04.0H*

Name: **Brake_Control_Time1_Close_Brake**
Index.Sub: **2C01.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: 0 ÷ 1000
Default Value: 150
Store Supported: Yes

Description: This object is used for the *Brake Control* (*§7.6*) functionality and defines the time between motor standstill and closing of Brake.

Notes:

- This object is available with firmware V03r20 or superior.
- See §7.6
- See *2C00.0H*, *2C02.0H*, *2C03.0H*, *2C04.0H*

Name: **Brake_Control_Time2_Close_Brake**
Index.Sub: **2C02.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: 0 ÷ 1000
Default Value: 150
Store Supported: Yes

Description: This object is used for the *Brake Control* (*§7.6*) functionality and defines the time between closing of Brake and switching off of motor current.

Notes:

- This object is available with firmware V03r20 or superior.
- See *§7.6*
- See *2C00.0H*, *2C01.0H*, *2C03.0H*, *2C04.0H*

Name: **Brake_Control_Time1_Open_Brake**
Index.Sub: **2C03.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: 0 ÷ 1000
Default Value: 150
Store Supported: Yes

Description: This object is used for the *Brake Control* (*§7.6*) functionality and defines the time between the switching on of motor current and the release of the Brake. During this time motor movement are not allowed.

Notes:

- This object is available with firmware V03r20 or superior.
- See *§7.6*
- See *2C00.0H*, *2C01.0H*, *2C02.0H*, *2C04.0H*

Name: **Brake_Control_Time2_Open_Brake**
Index.Sub: **2C04.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: 0 ÷ 1000
Default Value: 150
Store Supported: Yes

Description: This object is used for the *Brake Control* (*§7.6*) functionality and defines the time for the release of the Brake. During this time motor movement are not allowed.

Notes:

- This object is available with firmware V03r20 or superior.
- See *§7.6*
- See *2C00.0H*, *2C01.0H*, *2C02.0H*, *2C03.0H*

Name: **Node_Id**
Index.Sub: **4000.7H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 1 ÷ 127
Default Value: 1
Store Supported: Yes

Description: This object contains the drive's NodeId.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Baud_Rate**
Index.Sub: **4000.8H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 ÷ 3 (0 = 1M, 1 = 500k, 2 = 250k, 3 = 125k)
Default Value: 1
Store Supported: Yes

Description: This object contains the drive's Baud Rate for the CAN interface.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Firmware_Version**
Index.Sub: **4004.0H**
Data Type: Unsigned16 (high byte = version ; low byte = release)
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: --
Store Supported: No

Description:

Notes:

Name: **Firmware_Checksum**
Index.Sub: **4004.1H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: 0 ÷ FFFFH
Default Value: --
Store Supported: No

Description: This buffer contains the checksum of the firmware stored in drive.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Boot_Version**
Index.Sub: **4004.2H**
Data Type: Unsigned16 (high byte = version ; low byte = release)
Access: ro
PDO Mapping: no
Unit: **--**
Range: **--**
Default Value: --
Store Supported: No

Description: This object returns Boot version of drive.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Drive_Type**
Index.Sub: **4004.6H**
Data Type: Unsigned16 (high byte = Drive Type ; low byte = hardware revision)
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: --
Store Supported: No

Description: It contains the information about the drive type and hardware version. The MSB contains the drive board family, while the LSB contains the hardware version.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Configuration_Code**
Index.Sub: **4004.7H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: --
Store Supported: No

Description: This object contains the firmware configuration code.

Notes:
- This object is not available in EtherCAT fieldbus.

Name: **Warning_Inserted**
Index.Sub: **59A2.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: --
Default Value: --
Store Supported: No

Description: This object contains the state of the internal warning of the firmware.

Bit #	Name	Description	Default value
0	Warning_Voltage	1 = Drive voltage is near limit 0 = Ok	0
1	Warning_Temperature	1 = Drive temperature is near limit 0 = Ok	0
2	Warning_EEprom_Near_Write_Overrun	1 = EEprom near Write Overrun. 0 = Ok	0
3	Warning_EEprom_Near_EOL	1 = EEprom near End of Life 0 = Ok	0
4	Reserved	Reserved	0
5	Reserved	Reserved	0
6	Reserved	Reserved	0
7	Reserved	Reserved	0
8	Warning_Missing_Inominal	1 = Missing Nominal Current parameter 0 = Ok	0
9	Reserved	Reserved	0
10	Reserved	Reserved	0
11	Reserved	Reserved	0
12	Reserved	Reserved	0
13	Reserved	Reserved	0
14	Reserved	Reserved	0
15	Reserved	Reserved	0

Notes: - This object is not available in EtherCAT fieldbus.
- This object is available with firmware V01r17 or superior.

Name: **B0_Digital_Inputs**
Index.Sub: **6000.1H**
Data Type: Unsigned8
Access: ro
PDO Mapping: TX_PDO (1)
Unit: **--**
Range: 0 (all inputs are open) ÷ 2^n-1 (all inputs are closed, where n is the number of digital inputs available)
Default Value: --
Store Supported: No

Description: It contains the status of all the inputs on Bank #0 of the drive.
A numeric value is associated to the input, following the procedure laid out hereafter :

DIGITAL_INPUT	VALUE
B0_IN0	1
B0_IN1	2
B0_IN2	4
B0_IN3	8
B0_IN(n)	2^n

Notes:
- *(1) available with firmware V00r74 or superior.*

Name: **Abort_connection_option_code**
Index.Sub: **6007.0H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 = No action
1 = Fault signal
2 = Disable voltage command
3 = Quick stop command

Default Value: 0
Store Supported: Yes

Description: This object defines the reaction to be executed when a communication abort event is triggered.

Notes:

Name: **Error Code**
Index.Sub: **603F.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: TX_PDO (1)
Unit: --
Range: --
Default Value: 0
Store Supported: No

Description: This object captures the code of the last error that occurred in the drive.

Value (hex)	Description
0000 hex	No error
FF01-FFFF hex	Manufacturer specific

Notes:

- (1) *available with firmware V00r74 or superior.*

Name: **Controlword**
Index.Sub: **6040.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: --
Range: --
Default Value: --
Store Supported: No

Description: The Controlword controls the state machine, operating modes and manufacturer specific options.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M	O			M	M	M	M

***Details of Bit0,Bit1,Bit2,Bit3,Bit7* :**

Command	Bit7	Bit3	Bit2	Bit1	Bit0
Shutdown	X	X	1	1	0
Switch On	X	X	1	1	1
Disable Voltage	X	X	X	0	X
Quick Stop	X	X	0	1	X
Disable Operation	X	0	1	1	1
Enable Operation	X	1	1	1	1
Fault Reset	↑	X	X	X	X

Notes :

- (1) *available with firmware V00r74 or superior.*

- ***Controlword for Interpolated Position Mode (ip)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	QuickStop	Enable Voltage	Switch on
O					O		O	M	O			M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	reserved		**Enable ip mode**	X	X	X	X

Bit	Name	Value	Description
4	**Enable ip mode**	0	Interpolated position mode inactive
		1	Interpolated position mode active
8	**Halt**	0	Execute the instruction of bit 4
		1	Axis shall be stopped accordingly to *Halt_option_code* object

- ***Controlword for Profile Position Mode (pp)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	QuickStop	Enable Voltage	Switch on
O					O		O	M				M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	**Abs / Rel**	**Change set immediately**	**New set point**	X	X	X	X

Bit	Name	Value	Description
4	**New set-point**	0	Does not assume *Target Position*
		1	Assume *Target Position*
5	**Change set immediately**	0	Finish the actual positioning and then start the next positioning
		1	Interrupt the actual positioning and start the next positioning
6	**Abs / Rel**	0	*Target Position* is an absolute value
		1	*Target Position* is a relative value
8	**Halt**	0	Execute positioning
		1	Axis shall be stopped accordingly to *Halt_option_code* object

- ***Controlword for Homing Mode (hm)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M				M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	reserved		**Homing operation start**	X	X	X	X

Bit	Name	Value	Description
4	**Homing operation Start**	0	Homing mode inactive
		0 → 1	Start homing mode
		1	Homing mode active
		1 → 0	Interrupt homing mode
8	**Halt**	0	Execute the instruction of bit 4
		1	Axis shall be stopped accordingly to *Halt_option_code* object

- ***Controlword for Profile Velocity Mode (pv)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M				M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	**X**	**X**	**X**	**X**	X	X	X

Bit	Name	Value	Description
8	**Halt**	0	Execute the motion
		1	Axis shall be stopped accordingly to *Halt_option_code* object

- ***Controlword for Cyclic Synchronous Position Mode (csp)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M				M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	**X**	**X**	**X**	**X**	X	X	X

Bit	Name	Value	Description
8	**Halt**	0	Execute the motion
		1	Axis shall be stopped accordingly to *Halt_option_code* object

Note: The Halt bit is not available in EtherCAT fieldbus

- ***Controlword for Cyclic Synchronous Velocity Mode (csv)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M				M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	**X**	**X**	**X**	**X**	X	X	X

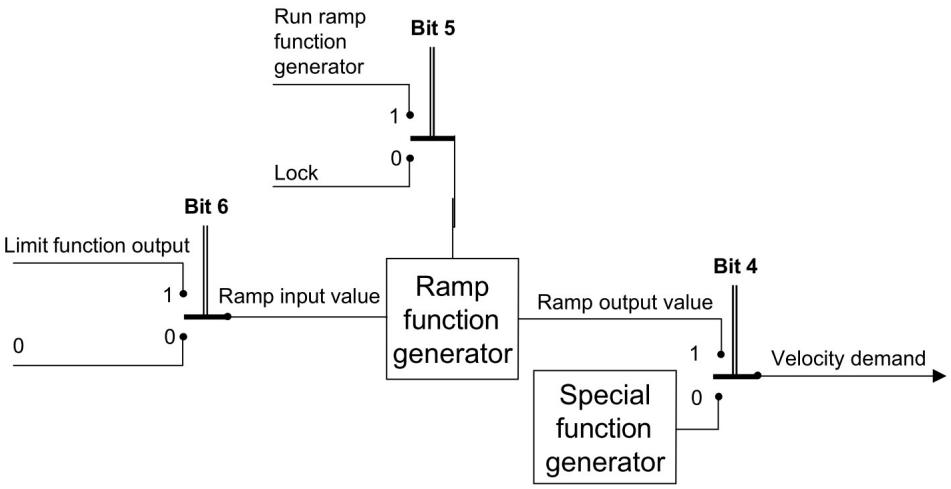
Bit	Name	Value	Description
8	**Halt**	0	Execute the motion
		1	Axis shall be stopped accordingly to *Halt_option_code* object

Note: The Halt bit is not available in EtherCAT fieldbus

- ***Controlword for Velocity Mode (vl)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Operation mode specific			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M	O	O	O	M	M	M	M
X	X	X	X	X	X	X	**Halt**	X	**Reference ramp**	**Unlock ramp**	**Enable ramp**	X	X	X	X

Bit	Name	Value	Description
4	**Enable ramp**	0	Velocity demand value shall be controlled in any other (manufacturer-specific) way
		1	Velocity demand value accords to ramp output value
5	**Unlock ramp**	0	Ramp output value is locked to current output value
		1	Ramp output value follows ramp input value
6	**Reference ramp**	0	Ramp input value is set to zero
		1	Ramp input value accords to ramp reference
8	**Halt**	0	Execute motion
		1	Stop axle



- ***Controlword for Profile Torque Mode (tq)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Reserved			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M	O	O	O	M	M	M	M
X	X	X	X	X	X	X	**X**	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
8	**Halt**	0	The motion shall be esecuted or continued
		1	Axis shall be stopped according to the Halt option code (605Dh)

- ***Controlword for Cyclic Synchronous Torque Mode (cst)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Reserved			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M	O	O	O	M	M	M	M
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

The cyclic synchronous torque mode uses no mode-specific bits of the Controlword.

- ***Controlword for Feedback Sensor Calibration Mode (fsc)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					Reserved		Halt	Fault reset	Reserved			Enable Operation	Quick Stop	Enable Voltage	Switch on
O					O		O	M	O	O	O	M	M	M	M
X	X	X	X	X	X	X	X	X	X	X	**Start Calibration**	X	X	X	X

Bit	Name	Value	Description
4	**Start Calibration**	0	Calibration procedure inactive
		0 → 1	Start Calibration procedure
		1	Calibration procedure active
		1 → 0	Interrupt Calibration procedure

Name: **Statusword**
Index.Sub: **6041.0H**
Data Type: Unsigned16
PDO Mapping: TX_PDO (1)
Access: ro
Unit: **--**
Range: --
Default Value: --
Store Supported: No

Description: The Statusword indicates the current state of the drive, the operating state of the mode and manufacturer specific options.

Notes:

- *(1) available with firmware V00r74 or superior.*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Op. mode specific		Int. limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M

Bits 0,1,2,3,5,6 : Device state (see table 1)

Bit4 : High voltage is applied to the drive (if bit is set to 1)

Bit5 : If this bit is equal to 0 , the drive is reacting on a quick stop request.

Bit7 : If this bit is set to 1, a drive warning is present. The cause means no error but a state that has to be mentioned. The status of the drive does not change. The cause of the warning may be found by reading the *59A2.0H* object. This bit is set and reset by the device.

Bit8 : This bit may be used by a drive manufacturer to implement any manufacturer specific functionality.

Bit9 : If this bit is set to 1, then parameters may be modified via CAN-network and the drive executes the content of a command message. If this bit is equal to 0, then the drive is in local mode and will not execute the command message. In local mode the drive will accept accesses via SDO.

Bit10 : If this bit is set to 1 then a set-point has been reached. The set-point is dependent on the operating mode. The description is in the chapter of the specific Operation mode.

Bit11 : This bit indicates that an internal limitation is active.

Bits 12,13 : This bits are operation mode specific. The description is in the chapter of the special mode.

Bits 14,15 : These bits may be used by a drive manufacturer to implement any manufacturer specific functionality.

The Bit6,Bit5,Bit3,Bit2,Bit1,Bit0 indicate the status of the device **:**

State	Bit6	Bit5	Bit3	Bit2	Bit1	Bit0
Not Ready to Switch On	0	X	0	0	0	0
Switch On Disabled	1	X	0	0	0	0
Ready to Switch On	0	1	0	0	0	1
Switched On	0	1	0	0	1	1
Operation Enabled	0	1	0	1	1	1
Fault	0	X	1	1	1	1
Fault Reaction Active	0	X	1	1	1	1
Quick Stop Active	0	0	0	1	1	1

- ***Statusword for Interpolated Position Mode (ip)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	reserved	Ip mode active	X	Target reached	X	X	X	X	X	X	X	X	X	X

Bit	Name		Description
10	**Target reached**	0	Halt = 0 : Position not (yet) reached Halt = 1 : Motor in deceleration
		1	Halt = 0 : Position reached Halt = 1 : Motor speed zero
12	**Ip mode active**	0	Interpolated position mode inactive
		1	Interpolated position mode active

- ***Statusword for Profile Position Mode (pp)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	Following error	Set-point acknowledge	X	Target reached	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
10	**Target reached**	0	Halt = 0 : *Target Position* not reached Halt = 1 : Axle decelerates
		1	Halt = 0 : *Target Position* reached Halt = 1 : Velocity of axle is 0
12	**Set-point acknowledge**	0	Trajectory generator has not assumed the positioning values (yet)
		1	Trajectory generator has assumed the positioning values
13	**Following error**	0	No following error
		1	Following error

- ***Statusword for Homing Mode (hm)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	Homing error	Homing attained	X	Target reached	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
10	**Target reached**	0	Halt = 0 : *Homing position* not reached Halt = 1 : Axle decelerates
		1	Halt = 0 : Homing position reached Halt = 1 : Velocity of axle is 0
12	**Homing attained**	0	Homing mode not yet completed
		1	Homing mode carried out successfully
13	**Homing error**	0	No Homing error
		1	Homing error occurred. Homing mode carried out not successfully. The error cause is found by reading the error code.

- ***Statusword for Profile Velocity Mode (pv)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	Max slippage error	Speed	X	Target reached	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
10	**Target reached**	0	Halt = 0 : *Target Velocity* not (yet) reached Halt = 1 : Axle decelerates
		1	Halt = 0 : *Target Velocity* reached Halt = 1 : Velocity of axle is 0
12	**Speed**	0	Not supported
		1	
13	**Max slippage error**	0	Not supported
		1	

- ***Statusword for Cyclic Synchronous Position Mode (csp)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	Following Error	Target Position Ignored	X	Reserv.	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
10	**Reserved**	0	Reserved
		1	Reserved
12	**Target Position Ignored**	0	Target position ignored
		1	Target position shall be used as input to position control loop
13	**Following Error**	0	No following error
		1	Following error

- ***Statusword for Cyclic Synchronous Velocity Mode (csv)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	Reserv.	Target Velocity Ignored	X	Reserv.	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
10	**Reserved**	0	Reserved
		1	Reserved
12	**Target Velocity Ignored**	0	Target velocity ignored
		1	Target velocity shall be used as input to velocity control loop
13	**Reserved**	0	Reserved
		1	Reserved

- ***Statusword for Velocity Mode (vl)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	reserved	reserved	X	reserved	X	X	X	X	X	X	X	X	X	X

- ***Statusword for Profile Torque Mode (tq)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	reserved	reserved	X	Target reached	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
10	**Target reached**	0	Halt (bit8 Controlword) = 0; Target torque not reached Halt (bit8 Controlword) = 1; Axis decelerates
		1	Halt (bit8 Controlword) = 0; Target torque reached Halt (bit8 Controlword) = 1; Velocity of axis is 0

- ***Statusword for Cyclic Synchronous Torque Mode (cst)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	X	reserved	**Drive follows the command value**	X	reserved	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
12	**Drive follows the command value**	0	Drive does not follow the command value – Target torque ignored
		1	Drive follows the command value – Target torque used as input to control loop

- ***Statusword for Feedback Sensor Calibration Mode (fsc)***

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manuf. specific		Operation mode specific		Internal limit active	Target reached	Remote	Manuf. specific	Warning	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on
O		O		M	M	M	O	O	M	M	M	M	M	M	M
X	Calibration Error	Calibration attained	Calibration in progress	X	reserved	X	X	X	X	X	X	X	X	X	X

Bit	Name	Value	Description
12	**Calibration in progress**	0	Feedback Sensor Calibration procedure not active
		1	Feedback Sensor Calibration procedure is in progress
13	**Calibration attained**	0	Feedback Sensor Calibration procedure not attained
		1	Feedback Sensor Calibration attained
14	**Calibration error**	0	Feedback Sensor Calibration procedure not active or in progress or not yet attained
		1	Feedback Sensor Calibration Error

Name: **vl Target Velocity**
Index.Sub: **6042.0H**
Data Type: Integer16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: User-defined velocity unit
Range: Integer16
Default Value: 0
Store Supported: No

Description: The 'vl Target Velocity' is the required velocity for the system.
Positive values shall indicate forward direction and negative values shall indicate backward direction.

The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of *vl Dimension Factor* :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

The 'device internal velocity unit' is 'rpm' [revs/min].

Notes:
- This object is used only for *Velocity_mode* (vl)
- *(1) available with firmware V00r74 or superior.*

Name: **vl Velocity Demand**
Index.Sub: **6043.0H**
Data Type: Integer16
Access: ro
PDO Mapping: no
Unit: User-defined velocity unit
Range: Integer16
Default Value: No
Store Supported: No

Description: This object is used for 'Velocity mode (vl)' and is the instantaneous velocity generated by the ramp function. Positive values shall indicate forward direction and negative values shall indicate backward direction.

The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of *vl Dimension Factor* :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

The 'device internal velocity unit' is 'rpm' [revs/min].

Notes: - This object is used only for *Velocity_mode* (vl).

Name: **vl Control Effort**
Index.Sub: **6044.0H**
Data Type: Integer16
Access: ro
PDO Mapping: TX_PDO (1)
Unit: User-defined velocity unit
Range: Integer16
Default Value: No
Store Supported: No

Description: This object is used for 'Velocity mode (vl)' and is the velocity at the motor spindle or load (without closed loop control this value reads the *vl Velocity Demand*). Positive values shall indicate forward direction and negative values shall indicate backward direction.

The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of *vl Dimension Factor* :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

The 'device internal velocity unit' is 'rpm' [revs/min].

Notes:

- *(1) available with firmware V00r74 or superior.*

Name: **vl Velocity min max amount (number of entries)**
Index.Sub: **6046.0H**
Data Type: Unsigned8
Access: ro
Unit: ---
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

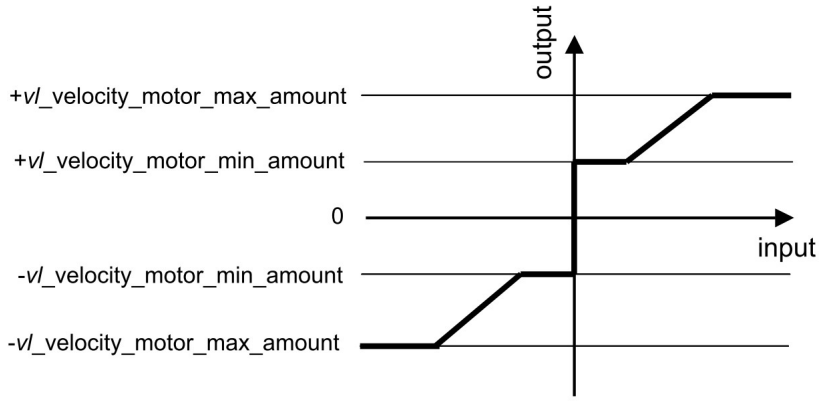
Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity min amount**
Index.Sub: **6046.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: User-defined velocity unit
Range: Unsigned32
Default Value: 1
Store Supported: No

Description: This object indicates the configured minimum amount of velocity.

The 'vl Velocity min amount' is mapped internally to the 'vl velocity min pos' and 'vl velocity min neg' values. Only the positive values is returned if the 'vl Velocity min amount' is read out.



The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of *vl Dimension Factor* :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

The 'device internal velocity unit' is 'rpm' [revs/min].

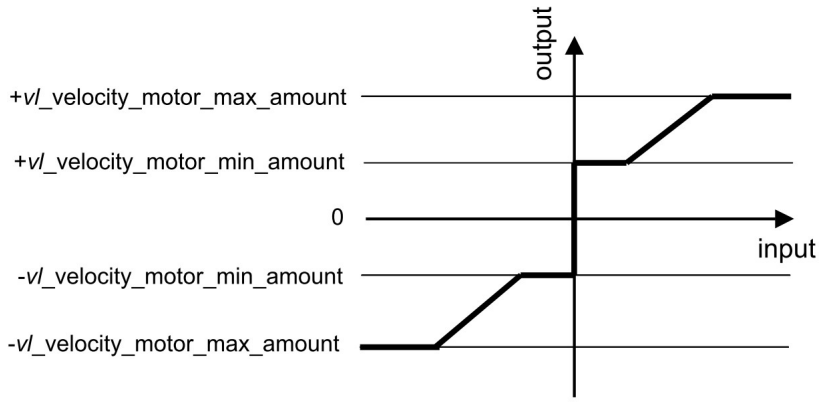
Notes :

- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity max amount**
Index.Sub: **6046.2H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: User-defined velocity unit
Range: Unsigned32
Default Value: 2000
Store Supported: No

Description: This object indicates the configured maximum amount of velocity.

The 'vl Velocity max amount' is mapped internally to the 'vl velocity max pos' and 'vl velocity max neg' values. Only the positive values is returned if the 'vl Velocity max amount' is read out.



The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of *vl Dimension Factor* :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

The 'device internal velocity unit' is 'rpm' [revs/min].

Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity acceleration (number of entries)**
Index.Sub: **6048.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: ---
Range: 2
Default Value: 2
Store Supported: No

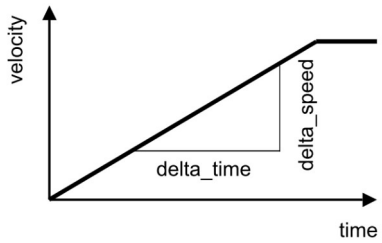
Description: Highest sub-index supported

Notes:
- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity acceleration (delta speed)**
Index.Sub: **6048.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: User-defined velocity unit
Range: Unsigned32
Default Value: 1500
Store Supported: No

Description: This object indicates the configured 'delta speed' of the slope for acceleration ramp :

$$vl\ velocity\ acceleration = \frac{delta\ speed}{delta\ time}$$



The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of *vl Dimension Factor* :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

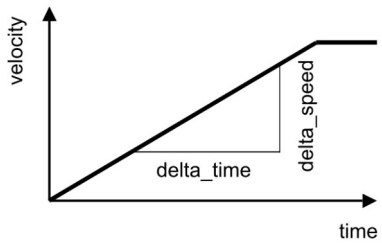
The 'device internal velocity unit' is 'rpm' [revs/min].

Notes:
- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity acceleration (delta time)**
Index.Sub: **6048.2H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: seconds
Range: Unsigned16
Default Value: 1
Store Supported: No

Description: This object indicates the configured 'delta time' of the slope for acceleration ramp :

$$vl\ velocity\ acceleration = \frac{delta\ speed}{delta\ time}$$



The value of 'delta time' shall be given in seconds.
This function directly follows the set-point if the parameter 0 is defined for the delta time value.

Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity deceleration (number of entries)**
Index.Sub: **6049.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: ---
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

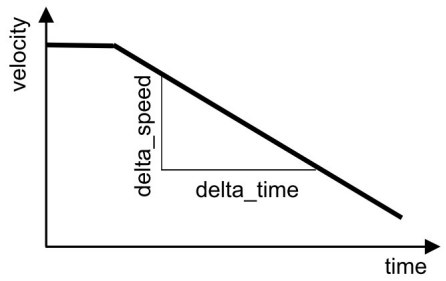
Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity deceleration (delta speed)**
Index.Sub: **6049.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: User-defined velocity unit
Range: Unsigned32
Default Value: 1500
Store Supported: No

Description: This object indicates the configured 'delta speed' of the slope for deceleration ramp :

$$vl\ velocity\ deceleration = \frac{delta\ speed}{delta\ time}$$



The User-defined velocity unit is converted to the revolution per minute (rpm) by mean of 'vl dimension factor' :

Velocity [user-defined units] x vl Dimension Factor = Velocity [rpm]

The 'device internal velocity unit' is 'rpm' [revs/min].

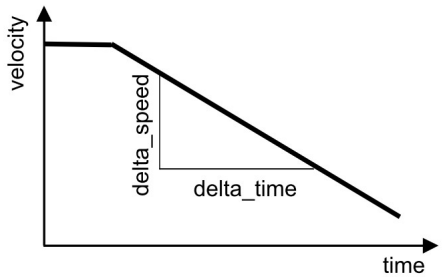
Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Velocity deceleration (delta time)**
Index.Sub: **6049.2H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: seconds
Range: Unsigned16
Default Value: 1
Store Supported: No

Description: This object indicates the configured 'delta time' of the slope for deceleration ramp :

$$vl\ velocity\ deceleration = \frac{delta\ speed}{delta\ time}$$



Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Dimension Factor (number of entries)**
Index.Sub: **604C.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: ---
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

Notes:

- This object is used only for *Velocity_mode* (vl).

Name: **vl Dimension Factor (numerator)**
Index.Sub: **604C.1H**
Data Type: Integer32
Access: rw
PDO Mapping: no
Unit: --
Range: Integer32 (except the value 0)
Default Value: 1
Store Supported: No

Description: This object indicates the numerator of the 'vl dimension factor'.

The *vl Dimension Factor* is used to include gearing in calculation or serves to scale the frequencies or specific units of the user. It influences the *vl Target Velocity*, *vl Velocity Demand*, *vl Control Effort* as well as the velocity limit function and ramp function.

The purpose of the 'vl dimension factor' is to convert the 'user-defined velocity unit' to the revolution/minute unit (rpm) :

Velocity [user-defined unit] x *vl Dimension Factor* = Velocity [rpm]

The *vl Dimension Factor* shall be in the range of -2147483648 to +2147483647, but the value 0 shall be not used.

Notes:

- The *vl Dimension Factor* is used only for objects retaled to the *Velocity_mode* (vl).

Name: **vl Dimension Factor (denominator)**
Index.Sub: **604C.2H**
Data Type: Integer32
Access: rw
PDO Mapping: no
Unit: --
Range: Integer32 (except the value 0)
Default Value: 1
Store Supported: No

Description: This object indicates the denominator of the 'vl dimension factor'.

The *vl Dimension Factor* is used to include gearing in calculation or serves to scale the frequencies or specific units of the user. It influences the *vl Target Velocity*, *vl Velocity Demand*, *vl Control Effort* as well as the velocity limit function and ramp function.

The purpose of the 'vl dimension factor' is to convert the 'user-defined velocity unit' to the revolution/minute unit (rpm) :

Velocity [user-defined unit] x *vl Dimension Factor* = Velocity [rpm]

The *vl Dimension Factor* shall be in the range of -2147483648 to +2147483647, but the value 0 shall be not used.

Notes:

- The *vl Dimension Factor* is used only for objects retaled to the *Velocity_mode* (vl).

Name: **Quick_stop_option_code**
Index.Sub: **605A.0H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 = Disable drive function, motor is free to rotate
2 = Slow down with quick stop ramp and transit into Switch On Disabled
6 = Slow down with quick stop ramp and stay in Quick Stop Active

Default Value: 0
Store Supported: Yes *(1)*

Description: This parameter determines what action should be taken if the *Quick Stop Function* is executed.

Notes:

- *(1)* 'Store supported' is available *with firmware V01r38 or superior.*

Name: **Halt_option_code**
Index.Sub: **605D.0H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: 2 = Slow down with quick stop ramp

Default Value: 2
Store Supported: No

Description: This parameter determines what action should be taken if the bit8 (Halt) in the *Controlword* is active

Notes:

Name: **Fault_reaction_option_code**
Index.Sub: **605E.0H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: 0 = Disable drive, motor is free to rotate
1 = Slow down with slow down ramp
2 = Slow down with quick stop ramp

Default Value: 0
Store Supported: Yes(1)

Description: This parameter determines what action should be taken if a non-fault occurs in the drive.

Notes:
- (1) 'Store supported' is available *with firmware V02r59 or superior.*

Name: **Modes of operation**
Index.Sub: **6060.0H**
Data Type: Integer8
Access: rw
PDO Mapping: RX_PDO (1)
Unit: **---**
Range: 9
Default Value: No
Store Supported: No

Description: This parameter switches the currently chosen operation mode. A read of **modes of operation** shows only the value of modes of operation. The current mode of the drive is reflected in the object **modes of operation display**.

Value	*Description*
-1	*Feedback_Sensor_Calibration_mode* (3)
1	*Profile_Position_Mode*
2	*Velocity_mode*
3	*Profile_Velocity_mode*
4	*Profile_Torque_mode* (2)
6	*Homing_mode*
7	*Interpolated_Position_Mode*
8	*Cyclic_Synchronous_Position_mode*
9	*Cyclic_Synchronous_Velocity_mode*
10	*Cyclic_Synchronous_Torque* (2)

Notes:
- (1) *available with firmware V00r74 or superior.*
- (2) *available with firmware V03r03 or superior.*
- (3) *available with firmware V03r21 or superior.*

Name: **Modes of operation display**
Index.Sub: **6061.0H**
Data Type: Integer8
Access: ro
PDO Mapping: TX_PDO [1]
Unit: **---**
Range: 9
Default Value: No
Store Supported: No

Description: This parameter shows the current mode of operation. The meaning of the returned value corresponds to that of the **modes of operation** (6060.0H).

Value	*Description*
-1	*Feedback_Sensor_Calibration_mode* [3]
1	*Profile_Position_Mode*
2	*Velocity_mode*
3	*Profile_Velocity_mode*
4	*Profile_Torque_mode* [2]
6	*Homing_mode*
7	*Interpolated_Position_Mode*
8	*Cyclic_Synchronous_Position_mode*
9	*Cyclic_Synchronous_Velocity_mode*
10	*Cyclic_Synchronous_Torque* [2]

Notes:

- [1] *available with firmware V00r74 or superior.*
- [2] *available with firmware V03r03 or superior.*
- [3] *available with firmware V03r21 or superior.*

Name: **Position_demand_value**
Index.Sub: **6062.0H**
Data Type: Integer32
Access: ro
PDO Mapping: no
Unit : User-defined position unit [1]
Range: Integer32
Default Value: --
Store Supported: No

Description: It indicates the present position demand value output from the trajectory generator.

Notes:

- [1] *See §3.2*

Name: **Position_Actual_Internal_Value**
Index.Sub: **6063.0H**
Data Type: Integer32
Access: ro
PDO Mapping: no
Unit: Increments
Range: -2147483648 ÷ 2147483647
Default Value: 0
Store Supported: No

Description: This object provides the position actual value of the Feedback Sensor or Motor (by mean of *Sensor_selection_code* object is possible to select the motor or encoder reference).

Notes:

Name: **Position_Actual_Value**
Index.Sub: **6064.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO [1]
Unit : User-defined position unit [2]
Range: -2147483648 ÷ 2147483647
Default Value: 0
Store Supported: No

Description: This object provides the position actual value.

Notes:
- [1] *available with firmware V00r74 or superior.*
- [2] *See §3.2*

Name: **Following_error_window**
Index.Sub: **6065.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit : User-defined position unit [1]
Range: Unsigned32
Default Value: FFFFFFFFH
Store Supported: No

Description: It defines a range of tolerated position values symmetrically to the *Position_demand_value*. If the *Position_Actual_Value* is out of the Following_error_window longer than the *Following_error_time_out* a following error occurs and the corresponding bit13 (following error) in the *Statusword*.

To reset following error state, is possible to use the *Homing_method_35* or *Homing_method_37* after that the emergency condition is reset by mean of the Fault_reset command in the *Controlword*.

If the value of *Following_error_window* is FFFFFFFFH, the following control will be switched off.

Notes:
- [1] *See §3.2*

Name: **Following_error_time_out**
Index.Sub: **6066.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: When the following error occurs longer than the defined value of the time-out, the corresponding Bit13 (following error) in the *Statusword* will be set to one.

Notes:

Name: **Position_window**
Index.Sub: **6067.0H**
Data Type: Integer32
Access: rw
PDO Mapping: no
Unit : User-defined position unit (1)
Range: Integer32
Default Value: FFFF FFFFh
Store Supported: Yes (2)

Description: This object defines a symmetrical range of accepted positions relatively to the target position: (*Target_Position* – position_window ; *Target_Position* + position_window).

If the position is within the position window, this *Target_Position* is regarded as reached.
If the value of the position window is 2^{32} -1, the position window control is switched off.

Notes:

- (1) *See* §3.2
- (2) 'Store supported' is available *with firmware V03r51 or superior.*

Name: **Position_window_time**
Index.Sub: **6068.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: Unsigned16
Default Value: 0
Store Supported: Yes (1)

Description: When the actual position is within the *Position_window* during the defined *position window time*, the corresponding bit10 *target reached* in the *Statusword* will be set to one.

Notes:

- (1) 'Store supported' is available *with firmware V03r51 or superior.*

Name: **Sensor_selection_code**
Index.Sub: **606A.0H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: --
Default Value: -1
Store Supported: Yes (1)

Description: This object defines if Position and Velocity values are related or not to an Feedback Sensor :

0 → Encoder#0 [Incremental Encoder]
-1 → *Open Loop* = No Encoder (theoretical Position and Velocity)
Closed Loop = Feeedback Sensor Type (Feedback_Settings , bit8÷bit11)
-2 → Encoder#1 [Incremental Encoder]
-3 → Encoder (BiSS) [Absolute Encoder multi turn BiSS] (1)

Notes:

- (1) *available with firmware V01r36 or superior.*
- *See* §3.2
- Refer to the hardware manual for connections

Name: **Velocity_demand_value**
Index.Sub: **606B.0H**
Data Type: Integer32
Access: ro
PDO Mapping: no
Unit : User-defined velocity unit *(1)*
Range: Integer32
Default Value: --
Store Supported: No

Description: It indicates the output of the trajectory generator.

Notes:
- *(1) See §3.2*

Name: **Velocity_Actual_Value**
Index.Sub: **606C.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO (1)
Unit : User-defined velocity unit *(2)*
Range: Integer32
Default Value: 0
Store Supported: No

Description: This object contains the current velocity of the motor.

Notes:
- *(1) available with firmware V00r74 or superior.*
- *(2) See §3.2*

Name: **Velocity_window**
Index.Sub: **606D.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit : User-defined velocity unit *(1)*
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: This object is used to handle the bit10 (target reached) 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 bit10 is set.

Notes:
- *(1) See §3.2*

Name: **Velocity_window_time**
Index.Sub: **606E.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: no
Unit: ms
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: This object is used to handle the bit10 (target reached) 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 bit10 is set.

Notes:

Name: **Target_Torque**
Index.Sub: **6071.0H**
Data Type: Integer16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: ‰ Nominal_Current
Range: 0 ÷ (‰ Max drive current)
Default Value: 0
Store Supported: No

Description: This object defines the Target torque for *Profile_Torque* (tq) and *Cyclic_Synchronous_Torque* (cst) modes. The value is considered proportional to *Nominal_Current* (Motor rated current).

The value is defined as thousandths of the torque, e.g. '500' means '50%' of the Motor rated torque.

Notes:

- This object is available only with firmware version V03r03 or superior.
- *(1) available with firmware V03r03 or superior.*

Name: **Max_Torque**
Index.Sub: **6072.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: ‰ Nominal_Current
Range: 0 ÷ (‰ Max drive current)
Default Value: 1000
Store Supported: Yes

Description: This object defines the maximum torque for *Profile_Torque* (tq) and *Cyclic_Synchronous_Torque* (cst) modes. The value is considered proportional to *Nominal_Current* (Motor rated current).

The value is defined as thousandths of the torque, e.g. '500' means '50%' of the Motor rated torque.

Notes:

- This object is available only with firmware version V03r03 or superior.
- *(1) available with firmware V03r03 or superior.*

Name: **Torque_demand**
Index.Sub: **6074.0H**
Data Type: Integer16
Access: ro
PDO Mapping: TX_PDO (1)
Unit: ‰ Nominal_Current
Range: 0 ÷ (‰ Max drive current)
Default Value: 0
Store Supported: No

Description: This object is used for *Profile_Torque* (tq) and *Cyclic_Synchronous_Torque* (cst) modes and provides the actual torque value of the trajectory generator. The value is considered proportional to *Nominal_Current* (Motor rated current).

The value is defined as thousandths of the torque, e.g. '500' means '50%' of the Motor rated torque.

Notes:

- This object is available only with firmware version V03r03 or superior.
- *(1) available with firmware V03r03 or superior.*

Name: **Torque_actual_value**
Index.Sub: **6077.0H**
Data Type: Integer16
Access: ro
PDO Mapping: TX_PDO (1)
Unit: ‰ Nominal_Current
Range: 0 ÷ (‰ Max drive current)
Default Value: 0
Store Supported: No

Description: This object is used for *Profile_Torque* (tq) and *Cyclic_Synchronous_Torque* (cst) modes and provides the actual value of the torque. The value is considered proportional to *Nominal_Current* (Motor rated current).

The value is defined as thousandths of the torque, e.g. '500' means '50%' of the Motor rated torque.

Notes:

- This object is available only with firmware version V03r03 or superior.
- *(1) available with firmware V03r03 or superior.*

Name: **Torque_slope**
Index.Sub: **6087.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit: ‰ Nominal_Current /sec
Range: 0 ÷ 2147483647
Default Value: 10000
Store Supported: Yes

Description: This object defines the slope for *Profile_Torque* (tq) and *Cyclic_Synchronous_Torque* (cst) modes. The value is considered proportional to *Nominal_Current* (Motor rated current).

The value is defined as units of thousandths of the torque per second.

Notes:

- This object is available only with firmware version V03r03 or superior.
- *(1) available with firmware V03r03 or superior.*

Name: **Current_Actual_Value**
Index.Sub: **6078.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: TX_PDO (1)
Unit: mA
Range: 0 ÷ (max drive current)
Default Value: 0
Store Supported: No

Description: This object contains the actual value of the motor current (rms) supplied to the motor.

Notes:

- *(1) available with firmware V03r02 or superior.*

Name: **Target_Position**
Index.Sub: **607A.0H**
Data Type: Integer32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined position unit (2)
Range: Integer32
Default Value: --
Store Supported: No

Description: This object specifies the target position in *Profile_Position_Mode* (pp) and *Cyclic_Synchronous_Position_mode* (csp). It can be an absolute position or relative position depending on Bit6 setting of the *Controlword*(pp) and it is always an absolute position in *Cyclic_Synchronous_Position_mode* (csp).

Notes:
- *(1) available with firmware V00r74 or superior.*
- *(2) See §3.2*

Name: **Home_Offset**
Index.Sub: **607C.0H**
Data Type: Integer32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined position unit (2)
Range: Integer32
Default Value: 0
Store Supported: Yes (3)

Description: The *home offset* is the difference between the zero position for the application and the machine home position (found during homing). 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.

Notes:
- *(1) available with firmware V02r20 or superior.*
- *(2) See §3.2*
- *(3)* 'Store supported' is available *with firmware V01r36 or superior.*
- *See §3.5*

Name: **Max_Profile_Velocity**
Index.Sub: **607F.0H**
Data Type: Unsigned32
Access: rw (4)
ro (5)
PDO Mapping: RX_PDO (1)
Unit : User-defined velocity unit (2)
Range : Min = 250 ÷ 'Max' (3)
Default Value : 'Max' (3)
Store Supported: No

Description: The Max_Profile_Velocity is the maximum allowed speed during a profiled move.

This object can only be set with the motor at a standstill.
This object must be higher than the *Min_Profile_Velocity object.*
If the value to be set is lower than the minimum value range, the minimum value is stored.

Notes:
- *(1) available with firmware V02r20 or superior.*
- *(2) See §3.2*
- *(3) 'Max'* value is calculated according to this formula :
  *'Max'(User-defined velocity unit) = 3276800 * (2013.2H / 2013.1H) * (60EF.0H / 65536)*
- *(4) with firmware V02r23 or lower.*
- *(5) with firmware V02r24 or superior.*

Name: **Profile_Velocity**
Index.Sub: **6081.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined velocity unit (2)
Range: 1 ÷ *Max_Profile_Velocity* (*607F.0H*)
Default Value: --
Store Supported: Yes

Description: It sets the velocity normally attained at the end of the acceleration ramp during a profile move and is valid both directions of motion.

Notes:
- (1) *available with firmware V00r74 or superior.*
- (2) *See §3.2*

Name: **Profile_Acceleration**
Index.Sub: **6083.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined acceleration unit (2)
Range: unsigned32
Default Value: 20000
Store Supported: No

Description: It indicates the configured acceleration ramp.

Notes:
- (1) *available with firmware V02r20 or superior.*
- (2) *See §3.2*

Name: **Profile_Deceleration**
Index.Sub: **6084.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined acceleration unit (2)
Range: unsigned32
Default Value: 20000
Store Supported: No

Description: It indicates the configured deceleration ramp.

Notes:
- (1) *available with firmware V02r20 or superior.*
- (2) *See §3.2*

Name: **Quick_Stop_deceleration**
Index.Sub: **6085.0H**
Data Type: Unsigned32
Access: r/w
PDO Mapping: no
Unit : User-defined acceleration unit (1)
Range: Unsigned32
Default Value: 20000
Store Supported: Yes (2)

Description: This object sets the deceleration ramp used to stop the motor if the 'Quick Stop' command is given and the *Quick_stop_option_code* is set to 2.

Notes:
- (1) *See §3.2*
- (2) 'Store supported' is available *with firmware V03r60 or superior.*

Name: **Motion_profile_type**
Index.Sub: **6086.0H**
Data Type: Integer16
Access: ro
PDO Mapping: no
Unit: --
Range: 0 (Linear ramp, trapezoidal profile)
Default Value: 0
Store Supported: No

Description: This object indicates the configured type of motion profile used to perform a profiled motion. The 'Titanio-Platino-Vanadio Drives' supports only the linear ramp (trapezoidal profile).

Notes:

Name: **Position Encoder Resolution (number of entries)**
Index.Sub: **608F.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: **--**
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported.

Notes:
- See *§3.2*

Name: **Position Encoder Resolution (Encoder Increments)**
Index.Sub: **608F.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: Increments
Range: **--**
Default Value: 1600
Store Supported: Yes

Description: The *Position Encoder Resolution* defines the ratio of encoder increments per motor revolutions :

$$Position\ Encoder\ Resolution = \frac{Encoder\ Increments\,(608\,F\,.1\,h)}{Motor\ Revolution\,(608\,F\,.2\,h)}$$

Notes:
- *See* §3.2

Name: **Position Encoder Resolution (Motor Revolution)**
Index.Sub: **608F.2H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **---**
Range: 1 - Unsigned32
Default Value: 1
Store Supported: Yes

Description: The *Position Encoder Resolution* defines the ratio of encoder increments per motor revolutions :

$$Position\ Encoder\ Resolution = \frac{Encoder\ Increments\,(608F.1h)}{Motor\ Revolution\,(608F.2h)}$$

Notes:

- See *§3.2*

Name: **Gear_Ratio (Highest sub-index supported)**
Index.Sub: **6091.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: --
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

Notes:

Name: **Gear_Ratio (Motor_Shaft_Revolutions)**
Index.Sub: **6091.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: --
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object defines the 'Motor shaft revolutions' used to calculate the Gear ratio between Motor Shaft Revolutions and Driving Shaft revolutions :

$$Gear\ Ratio = \frac{6091.1h}{6091.2h}$$

Notes:

- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Gear_Ratio (Driving_Shaft_Revolutions)**
Index.Sub: **6091.2H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object defines the 'Driving shaft revolutions' used to calculate the Gear ratio between Motor Shaft Revolutions and Driving Shaft revolutions :

$$Gear\ Ratio = \frac{6091.1\,h}{6091.2\,h}$$

Notes:

- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Feed_Constant (Highest sub-index supported)**
Index.Sub: **6092.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: --
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

Notes:

Name: **Feed_Constant (Feed)**
Index.Sub: **6092.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_DO (1)
Unit: User-defined position unit
Range: 1÷4294967295
Default Value: 200
Store Supported: Yes

Description: This object is used to calculate the ratio of Feed (given in user-defined position units) per Driving Shaft revolutions :

$$Feed\ Constant = \frac{Feed\,(6092.1\,h)}{Driving\ Shaft\ Revolutions\,(6092.2\,h)}$$

Notes:

- This object is available with firmware V00r80 or superior.
- See *§3.2*
- *(1) available with firmware V03r02 or superior.*

Name: **Feed_Constant (Driving_Shaft_Revolutions)**
Index.Sub: **6092.2H**
Data Type: Unsigned32
Access: rw
PDO Mapping: no
Unit: **--**
Range: 1÷4294967295
Default Value: 1
Store Supported: Yes

Description: This object is used to calculate the ratio of Feed (given in user-defined position units) per Driving Shaft revolutions :

$$Feed\ Constant = \frac{Feed\,(6092.1\,h)}{Driving\,Shaft\,Revolutions\,(6092.2\,h)}$$

Notes:
- This object is available with firmware V00r80 or superior.
- See *§3.2*

Name: **Homing_method**
Index.Sub: **6098.0H**
Data Type: Integer8
Access: rw
PDO Mapping: RX_PDO [(1)]
Unit: --
Range: -1,-2,-3,-4,1,2,17,18,19,20,21,22,23,24,25,26,27,28,29,30,35,37
Default Value: 19
Store Supported: Yes [(2)]

Description: This object indicates the Homing method type used during the homing procedure.

Notes:
- *[(1)] available with firmware V02r20 or superior.*
- [(2)] 'Store supported' is available *with firmware V01r36 or superior.*
- *Homing methods -1, -2, -3, -4 are* available *with firmware V03r74 or superior.*
- *See §3.5*

Name: **Homing speeds (number of entries)**
Index.Sub: **6099.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: **--**
Range: 2
Default Value: 2
Store Supported: No

Description: Highest sub-index supported

Notes: - *See §3.5*

Name: **Homing speeds (Speed during search for switch)**
Index.Sub: **6099.1H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined velocity unit (2)
Range: 1 ÷ 'Max_Profile_Velocity' (*607F.0H*)
Default Value: 10000
Store Supported: Yes (3)

Description: This object is the Speed used during search for home switch for the Homing method (hm).

Notes:
- (1) *available with firmware V02r20 or superior.*
- (2) *See §3.2*
- (3) 'Store supported' is available *with firmware V01r36 or superior.*
- *See §3.5*

Name: **Homing speeds (Speed during search for zero)**
Index.Sub: **6099.2H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined velocity unit (2)
Range: 1 ÷ 'Max_Profile_Velocity' (*607F.0H*)
Default Value: 250
Store Supported: Yes (3)

Description: This object is the Speed used during search for zero position for *Homing_mode* (hm).

Notes:
- (1) *available with firmware V02r20 or superior.*
- (2) *See §3.2*
- (3) 'Store supported' is available with firmware V01r36 or superior.
- *See §3.5*

Name: **Homing acceleration**
Index.Sub: **609A.0H**
Data Type: Unsigned32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined acceleration unit (2)
Range: unsigned32
Default Value: 20000
Store Supported: Yes (3)

Description: It sets the acceleration and deceleration used for *Homing_mode* (hm).

Notes:
- (1) *available with firmware V02r20 or superior.*
- (2) *See §3.2*
- (3) 'Store supported' is available with firmware V01r36 or superior.
- *See §3.5*

Name: **Touch_Probe_Function**
Index.Sub: **60B8.0H**
Data Type: Unsigned16
Access: rw
PDO Mapping: RX_PDO (1)
Unit: --
Range: –
Default Value: 0000H
Store Supported: No

Description: This object indicates the configured function of the Touch Probe.

Bit	Value	Description
0	0	Switch off Touch Probe1
	1	Enable Touch Probe1
1	0	Trigger first event
	1	Trigger continuously
3,2	00b	Trigger with Touch Probe 1 Input (B0_IN0)
	01b	Trigger with zero impulse signal of position encoder
	10b	Touch Probe Source defined by object 60D0.1h
	11b	Reserved
4	0	Switch off sampling at positive edge of Touch Probe 1
	1	Enable sampling at positive edge of Touch Probe 1
5	0	Switch off sampling at negative edge of Touch Probe 1
	1	Enable sampling at negative edge of Touch Probe 1
6,7	--	Reserved
8	0	Switch off Touch Probe2
	1	Enable Touch Probe2
9	0	Trigger first event
	1	Trigger continuously
11,10	00b	Trigger with Touch Probe 2 Input (B0_IN1)
	01b	Trigger with zero impulse signal of position encoder
	10b	Touch Probe Source defined by object 60D0.2h
	11b	Reserved
12	0	Switch off sampling at positive edge of Touch Probe 2
	1	Enable sampling at positive edge of Touch Probe 2
13	0	Switch off sampling at negative edge of Touch Probe 2
	1	Enable sampling at negative edge of Touch Probe 2
14,15	--	Reserved

Notes:

- This object is available with firmware V01r11 or superior.
- See *§3.12*
- *(1) available with firmware V02r20 or superior.*

Name: **Touch_Probe_Status**
Index.Sub: **60B9.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: TX_PDO (1)
Unit: --
Range: –
Default Value: 0000H
Store Supported: No

Description: This object provides the status of the Touch Probe.

Bit	Value	Description
0	0	Touch Probe1 is switched off
	1	Touch Probe1 is enabled
1	0	Touch Probe1 no positive edge value stored
	1	Touch Probe1 positive edge position stored
2	0	Touch Probe1 no negative edge value stored
	1	Touch Probe1 negative edge position stored
3-5	0	Reserved
6	0	Reserved
7	0	Touch Probe1 Input = low level
	1	Touch Probe1 Input = high level
8	0	Touch Probe2 is switched off
	1	Touch Probe2 is enabled
9	0	Touch Probe2 no positive edge value stored
	1	Touch Probe2 positive edge position stored
10	0	Touch Probe2 no negative edge value stored
	1	Touch Probe2 negative edge position stored
11-13	0	Reserved
14	0	Reserved
15	0	Touch Probe2 Input = low level
	1	Touch Probe2 Input = high level

- Bit1 and bit2 are set to 0 when touch probe1 is switched off (object *60B8.0H* bit0 is 0).
- Bit9 and bit10 are set to 0 when touch probe2 is switched off (object *60B8.0H* bit8 is 0).

Notes:

- This object is available with firmware V01r11 or superior.
- See *§3.12*
- *(1) available with firmware V02r20 or superior.*

Name: **Touch_Probe1_Positive_Edge**
Index.Sub: **60BA.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO (1)
Unit: User-defined position unit (2)
Range: Integer32
Default Value: 0000H
Store Supported: No

Description: This object provides the position value of the Touch Probe1 at positive edge.

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*
- *(1) available with firmware V02r20 or superior.*
- *(2) See §3.2*

Name: **Touch_Probe1_Negative_Edge**
Index.Sub: **60BB.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO (1)
Unit: User-defined position unit (2)
Range: Integer32
Default Value: 0000H
Store Supported: No

Description: This object provides the position value of the Touch Probe1 at negative edge.

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*
- *(1) available with firmware V02r20 or superior.*
- *(2) See §3.2*

Name: **Touch_Probe2_Positive_Edge**
Index.Sub: **60BC.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO (1)
Unit: User-defined position unit (2)
Range: Integer32
Default Value: 0000H
Store Supported: No

Description: This object provides the position value of the Touch Probe2 at positive edge.

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*
- *(1) available with firmware V02r20 or superior.*
- *(2) See §3.2*

Name: **Touch_Probe2_Negative_Edge**
Index.Sub: **60BD.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO (1)
Unit: User-defined position unit (2)
Range: Integer32
Default Value: 0000H
Store Supported: No

Description: This object provides the position value of the Touch Probe2 at negative edge.

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*
- (1) *available with firmware V02r20 or superior.*
- (2) *See §3.2*

Name: **Touch_Probe_Source (Highest sub-index supported)**
Index.Sub: **60D0.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: --
Range: 1÷254
Default Value: 2
Store Supported: No

Description: This object provides the number of Touch Probe Sources supported.

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*

Name: **Touch_Probe1_Source**
Index.Sub: **60D0.1H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: see table below
Default Value: -1
Store Supported: Yes (1)

Description: This object provides the source of the Touch Probe 1 functions.

Value	Description
-8	B0_IN7 (Digital Input 7 Bank0)
-7	B0_IN6 (Digital Input 6 Bank0)
-6	B0_IN5 (Digital Input 5 Bank0)
-4	B0_IN3 (Digital Input 3 Bank0)
-3	B0_IN2 (Digital Input 2 Bank0)
-2	B0_IN1 (Digital Input 1 Bank0)
-1	B0_IN0 (Digital Input 0 Bank0)
+5	Hardware Zero impulse signal of position encoder

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*
- (1) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Touch_Probe2_Source**
Index.Sub: **60D0.2H**
Data Type: Integer16
Access: rw
PDO Mapping: no
Unit: --
Range: see table below
Default Value: -2
Store Supported: Yes (1)

Description: This object provides the source of the Touch Probe 2 functions.

Value	Description
-8	B0_IN7 (Digital Input 7 Bank0)
-7	B0_IN6 (Digital Input 6 Bank0)
-6	B0_IN5 (Digital Input 5 Bank0)
-4	B0_IN3 (Digital Input 3 Bank0)
-3	B0_IN2 (Digital Input 2 Bank0)
-2	B0_IN1 (Digital Input 1 Bank0)
-1	B0_IN0 (Digital Input 0 Bank0)
+5	Hardware Zero impulse signal of position encoder

Notes:

- This object is available with firmware V01r11 or superior.
- *See §3.12*
- (1) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Touch_Probe1_Positive_Edge_Counter**
Index.Sub: **60D5.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: This object provides a continuous counter that is incremented with each positive edge at touch probe 1. The counter is only valid if Touch probe input is enabled (*60B8.0H*, Bit0=1) and continuous touch probe mode is enabled (*60B8.0H*, Bit1 =1).

Notes:

- This object is available with firmware V01r11 or superior.
- *See §3.12*

Name: **Touch_Probe1_Negative_Edge_Counter**
Index.Sub: **60D6.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: This object provides a continuous counter that is incremented with each negative edge at touch probe 1. The counter is only valid if Touch probe input is enabled (*60B8.0H*, Bit0=1) and continuous touch probe mode is enabled (*60B8.0H*, Bit1 =1).

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*

Name: **Touch_Probe2_Positive_Edge_Counter**
Index.Sub: **60D7.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: This object provides a continuous counter that is incremented with each positive edge at touch probe 2. The counter is only valid if Touch probe input is enabled (*60B8.0H*, Bit8=1) and continuous touch probe mode is enabled (*60B8.0H*, Bit9 =1).

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*

Name: **Touch_Probe2_Negative_Edge_Counter**
Index.Sub: **60D8.0H**
Data Type: Unsigned16
Access: ro
PDO Mapping: no
Unit: --
Range: Unsigned16
Default Value: 0
Store Supported: No

Description: This object provides a continuous counter that is incremented with each negative edge at touch probe 2. The counter is only valid if Touch probe input is enabled (*60B8.0H*, Bit8=1) and continuous touch probe mode is enabled (*60B8.0H*, Bit9 =1).

Notes:
- This object is available with firmware V01r11 or superior.
- *See §3.12*

Name: **Interpolation data record (Highest sub-Index supported)**
Index.Sub: **60C1.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: **--**
Range: 1
Default Value: 1
Store Supported: No

Description: Number of sub-Index supported.

Notes:
- See *60C1.1H* object.

Name: **Interpolation data record (position set-point)**
Index.Sub: **60C1.1H**
Data Type: Integer32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined position unit (2)
Range: -2147483648 ÷ 2147483647
Default Value: --
Store Supported: No

Description: This object defines the position set-point used for linear interpolation used for *Interpolated_Position_Mode*.

Notes:
- *(1) available with firmware V00r74 or superior.*
- *(2) See §3.2*

Name: **Interpolation time period (Highest sub-Index supported)**
Index.Sub: **60C2.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: ---
Range: 2
Default Value: 2
Store Supported: No

Description: Number of sub-Index supported.

Notes:
- See *60C2.1H* and *60C2.2H* objects.

Name: **Interpolation time period (interpolation time units)**
Index.Sub: **60C2.1H**
Data Type: Unsigned8
Access: rw
PDO Mapping: RX_PDO (1)
Unit: ---
Range: ---
Default Value: 1
Store Supported: Yes (2)

Description: This object is used to define for interpolation period computation :

Interpolation time period = Interpolation_time_units * 10 $^{Interpolation\_time\_index}$ seconds

Notes:

- See *60C2.2H* object.
- (1) *available with firmware V03r03 or superior.*
- (2) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Interpolation time period (interpolation time index)**
Index.Sub: **60C2.2H**
Data Type: Signed8
Access: rw
PDO Mapping: no
Unit: ---
Range: only -3 (corresponds to the time basis in milliseconds)
Default Value: -3
Store Supported: No

Description: This object is used to define for interpolation period computation and defines the power of ten of interpolation time :

Interpolation time period = Interpolation_time_units * 10 $^{Interpolation\_time\_index}$ seconds

Notes:

- See *60C2.1H* object.

Name: **Interpolation sync definition (Highest sub-Index supported)**
Index.Sub: **60C3.0H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: ---
Range: 2
Default Value: 2
Store Supported: No

Description: Number of sub-Index supported.

Notes:

- This object is not available in EtherCAT fieldbus.

Name: **Interpolation sync definition (synchronize on group)**
Index.Sub: **60C3.1H**
Data Type: Unsigned8
Access: rw
PDO Mapping: no
Unit: ---
Range: 0 ÷ 255
Default Value: 0
Store Supported: Yes (1)

Description: This object defines the interpolation method.

0 = SYNC
1...255 = reserved (not implemented)

Notes: - This object is not available in EtherCAT fieldbus.
- (1) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Interpolation sync definition (ip sync every n event)**
Index.Sub: **60C3.2H**
Data Type: Unsigned8
Access: rw
PDO Mapping: no
Unit: ---
Range: 0 ÷ 255
Default Value: 1
Store Supported: Yes (1)

Description: This object defines the number of SYNC after that position set-point used for interpolation is carried out.

Notes: - This object is not available in EtherCAT fieldbus.
- (1) 'Store supported' is available *with firmware V01r36 or superior.*

Name: **Motor_Resolution**
Index.Sub: **60EF.0H**
Data Type: Unsigned32
Access: ro [1]
rw [2]
PDO Mapping: no
Unit: Increments
Range: 200[1] ÷ 65536
0 [2] ÷ 65536
Default Value: 200 [1]
0 [2]
Store Supported: No [1]
Yes [2]

Description: This objects is the Motor Resolution.

Notes:
- [1] *with firmware V03r16 or lower.*
- [2] *with firmware V03r17 or superior.*

- This object is not available in EtherCAT fieldbus.
- This object is available with firmware V00r80 or superior.
- This object cannot be modified when 'Operation Enable' State is active.

- See *§3.2*
- *See 2012.1H, 2012.2H*

- With firmware V03r16 or lower the motor resolution is defined by mean of *Motor_Step_Angle* and *Motor_Pole_Pairs* objects (in this case *Motor_Resolution* object is read only and returns the motor resolution value).

- With firmware V03r17 or superior the motor resolution can be defined by mean of *Motor_Resolution* object or *Motor_Step_Angle*, *Motor_Pole_Pairs* objects. If *Motor_Resolution* object value is 0 then the motor resolution is defined by mean of *Motor_Step_Angle* and *Motor_Pole_Pairs* objects (for compatibility reasons). A value of *Motor_Resolution* object different from zero defines directly the motor resolution (*Motor_Step_Angle* object is not considered and only *Motor_Pole_Pairs* object have to be defined).

Name: **Following_error_actual_value**
Index.Sub: **60F4.0H**
Data Type: Integer32
Access: ro
PDO Mapping: TX_PDO [1]
Unit : User-defined position unit [2]
Range: Integer32
Default Value: --
Store Supported: No

Description: It represents the actual value of the following error.

Notes:
- [1] *available with firmware V00r74 or superior.*
- [2] *See §3.2*

Name: **Digital_Inputs**
Index.Sub: **60FD.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: TX_PDO (1)
Unit: --
Range: Unsigned32
Default Value: --
Store Supported: No

Description: This object contains the status of Digital inputs of the drive :

Bit#	*Description*	
0	***Negative limit switch***	*Negative Limit Switch*
1	***Positive limit switch***	*Positive Limit Switch*
2	***Home switch***	*Home Switch*
3	Not implemented	Interlock
4		reserved
5		reserved
6		reserved
7		reserved
8		reserved
9		reserved
10		reserved
11		reserved
12		reserved
13		reserved
14		reserved
15		reserved
16	***B0_IN0***	Manufacturer specific
17	***B0_IN1***	Manufacturer specific
18	***B0_IN2***	Manufacturer specific
19	***B0_IN3***	Manufacturer specific
20	***B0_IN4***	Manufacturer specific
21	***B0_IN5***	Manufacturer specific
22	***B0_IN6***	Manufacturer specific
23	***B0_IN7***	Manufacturer specific
24		Manufacturer specific
25		Manufacturer specific
26		Manufacturer specific
27		Manufacturer specific
28		Manufacturer specific
29		Manufacturer specific
30		Manufacturer specific
31		Manufacturer specific

Notes:

- See 2081.0h object.
- This object is available *with firmware V00r70 or superior.*
- (1) *available with firmware V00r74 or superior.*

Name: **Target_Velocity**
Index.Sub: **60FF.0H**
Data Type: Integer32
Access: rw
PDO Mapping: RX_PDO (1)
Unit : User-defined velocity unit (2)
Range: 0 ÷ *Max_Profile_Velocity* (*607F.0H*)
Default Value: 0
Store Supported: No

Description: This object indicates the configured target velocity. It is used in *Profile_Velocity_mode* (pv) and *Cyclic_Synchronous_Velocity_mode*(csv).

Notes:

- (1) *available with firmware V00r74 or superior.*
- (2) *See §3.2*

Name: **B1_Digital_Inputs**
Index.Sub: **6100.2H**
Data Type: Unsigned8
Access: ro
PDO Mapping: no
Unit: **--**
Range: 0 (all inputs are open) ÷ 2^n-1 (all inputs are closed, where n is the number of digital inputs available)
Default Value: --
Store Supported: No

Description: It contains the status of all the inputs on bank 1 of the drive.
A numeric value is associated to the input :

INPUTS	VALUE
B1_IN0	1
B1_IN1	2
B1_IN2	4
B1_IN3	8
B1_IN4	16
B1_IN5	32
B1_INn	2^n

Notes:

Name: **B0_Digital_Outputs**
Index.Sub: **6200.1H**
Data Type: Unsigned8
Access: rw
PDO Mapping: RX_PDO [(1)]
Unit: --
Range: 0 (all outputs are open) ÷ 2^n-1 (all outputs are closed, where n is the number of digital outputs available)
Default Value: --
Store Supported: No

Description: It contains the status of all the outputs on bank 0 of the drive.
A numeric value is associated to the outputs :

OUTPUTS	VALUE
B0_OUT0	1
B0_OUT1	2
B0_OUT2	4
B0_OUT3	8
B0_OUT4	16
B0_OUT5	32
B0_OUTn	2^n

Notes:
- *[(1)] available with firmware V00r74 or superior.*

Name: **B1_Digital_Outputs**
Index.Sub: **6300.2H**
Data Type: Unsigned8
Access: rw
PDO Mapping: no
Unit: **--**
Range: 0 (all outputs are open) ÷ 2^n-1 (all outputs are closed, where n is the number of digital outputs available)
Default Value: --
Store Supported: No

Description: It contains the status of all the outputs on bank 1 of the drive.
A numeric value is associated to the outputs :

OUTPUTS	VALUE
B1_OUT0	1
B1_OUT1	2
B1_OUT2	4
B1_OUT3	8
B1_OUT4	16
B1_OUT5	32
B1_OUTn	2^n

Notes :

Name: **Analog_In[0÷1]**
Index.Sub: **6404.1H,6404.2H**
Data Type: Integer16
Access: ro
PDO Mapping: TX_DO [1] (only for 6404.1h)
Unit: mV
Range: -10000 ÷ 10000
Default Value: --
Store Supported: No

Description: This object contains the value of the drive's analog input 0 and 1.

Notes:

- The number of available analog inputs depends on the version of the drive currently in use.
- [1] *available with firmware V00r74 or superior.*

Name: **Supported_drive_modes**
Index.Sub: **6502.0H**
Data Type: Unsigned32
Access: ro
PDO Mapping: no
Unit: ---
Range: Unsigned32
Default Value: 03EF hex
Store Supported: No

Description: This object provides information on the supported drive modes :

Bit value = 0 : Not supported
Bit value = 1 : Supported

31		16	15		14	9	8	7	6	5	4	3	2	1	0
Manufacturer specific						**cst**	**csv**	**csp**	**ip**	**hm**	reserved	**tq**	**pv**	**vl**	**pp**

Notes:

5.0 EtherCAT

The drives equipped with EtherCAT fieldbus that support CiA 402 Device Profile have a configuration code of C680. They do not need to be configured as NodeId and BaudRate (*§2.2*). The protocols supported are: CoE (CANopen over EtherCAT) and FoE (File Access over EtherCAT). The EVER drives supports different types of synchronization: Free Run, Synchronous with SM Event, Distributed Clocks. The services EMCY (*§2.7*) and Diagnostics are supported too.

Interpolation time period (interpolation time units - *60C2.1H*) must be uqual to DC (Distribuited clock) cycle time except for DC cycle time lower than 1ms. For DC cycle time equal to 250us or 500us, you have to set *60C2.1H* =1 (automatic internal adaptation to run at 250us or 500us).

5.1 STATE LED

Near the EtherCAT Connector A there is a STATE LED that can be in one of the following situation:

State	Slave Condition	Communication
Off	Init	After switch-on the EtherCAT slave is in the *Init* state. No SDO or PDO communication is possible.
Blinking	Pre-Operational	In *Pre-Operational* state SDO communication is possible, but not PDO communication.
Single Flash	Safe-Operational	In *Safe-Operational* state SDO and PDO communication is possible, although the slave keeps its outputs (RPDO) in a safe state, while the input (TPDO) data are updated cyclically.
On	Operational	In the *Operational* state the slave copies the output data of the masters to its outputs (RPDO). PDO and SDO communication is possible.

5.2 PDO Mapping

The PDO mapping for EtherCAT devices is both Static than dynamic (dynamic only for Mapping1) and contains the whole objects necessary to all supported modes. The following table shows Ethercat PDO Mapping list and default objects for each PDO Mapping :

PDO Mapping List	Description
RxPdoMapping1 (0x1600)	Controlword (*6040.0H*), Modes_of_Operation (*6060.0H*), B0_Digital_Outputs (*6200.1H*), Target_Position (*607A.0H*), Interpolation_Data_x1 (*60C1.1H*), Target_Velocity (*60FF.0H*)
RxPdoMapping2 (0x1601)	Controlword (*6040.0H*), Modes_of_Operation (*6060.0H*), B0_Digital_Outputs (*6200.1H*), Target_Position (*607A.0H*), Interpolation_Data_x1 (*60C1.1H*), Target_Velocity (*60FF.0H*), Touch_Probe_Function (*60B8.0H*)
RxPdoMapping3 (0x1602)	Controlword (*6040.0H*), Modes_of_Operation (*6060.0H*), B0_Digital_Outputs (*6200.1H*), Target_Position (*607A.0H*), Target_Velocity (*60FF.0H*), Max_Current (*2005.2H*), Boost_Current (*2005.3H*)
TxPdoMapping1 (0x1A00)	Statusword (*6041.0H*), Modes_of_Operation_Display (*6061.0H*), B0_Digital_Inputs (*6000.1H*), Position_Actual_Value (*6064.0H*),Velocity_Actual_Value (*606C.0H*), Following_Error_Actual_Value (*60F4.0H*)
TxPdoMapping2 (0x1A01)	Statusword (*6041.0H*), Modes_of_Operation_Display (*6061.0H*), B0_Digital_Inputs (*6000.1H*), Position_Actual_Value (*6064.0H*),Velocity_Actual_Value (*606C.0H*), Following_Error_Actual_Value (*60F4.0H*), Touch_Probe1_Positive_Edge (*60BA.0H*), Touch_Probe2_Positive_Edge (*60BC.0H*), Touch_Prove_Status (*60B9.0H*), Error_Code (*603F.0H*)
TxPdoMapping3 (0x1A02)	Statusword (*6041.0H*), Modes_of_Operation_Display (*6061.0H*), B0_Digital_Inputs (*6000.1H*), Position_Actual_Value (*6064.0H*), Error_Code (*603F.0H*)

Only one RxPdoMapping can be active. To change the active RxPdoMapping is necessary to set 1C12.1h obj (default value = 1600H) in Pre-Operational condition..

Only one TxPdoMapping can be active. To change the active TxPdoMapping is necessary to set 1C13.1h obj (default value = 1A00H) in Pre-Operational condition.

5.2.1 TX PDO dynamic mappable Objects

TX PDO Name	*Data Type*	*Index.sub*	*Xml (ESI) Version*	*Version*
Statusword	Unsigned16	*6041.0H*	V2.0 or superior	V02r20 or superior
Mode of operation Display	Integer8	*6061.0H*	V2.0 or superior	V02r20 or superior
B0_Digital_Inputs	Unsigned8	*6000.1H*	V2.0 or superior	V02r20 or superior
Position_Actual_Value	Integer32	*6064.0H*	V2.0 or superior	V02r20 or superior
Digital_Inputs	Unsigned32	*60FD.0H*	V2.0 or superior	V02r20 or superior
Velocity_Actual_Value	Integer32	*606C.0H*	V2.0 or superior	V02r20 or superior
vl_Control_Effort	Integer16	*6044.0H*	V2.1 or superior	V02r20 or superior
Encoder_Actual_Value0	Integer32	*2007.0H*	V2.0 or superior	V02r20 or superior
Encoder_Actual_Value1	Integer32	*2008.0H*	V2.0 or superior	V02r20 or superior
Following_error_Actual_value	Integer32	*60F4.0H*	V2.0 or superior	V02r20 or superior
Analog_In0	Integer16	*6404.1H*	V2.1 or superior	V02r20 or superior
Error_code	Unsigned16	*603F.0H*	V2.0 or superior	V02r20 or superior
Touch_Probe_Status	Unsigned16	*60B9.0H*	V2.0 or superior	V02r20 or superior
Touch_Probe1_Pos_Edge	Integer32	*60BA.0H*	V2.0 or superior	V02r20 or superior
Touch_Probe2_Pos_Edge	Integer32	*60BC.0H*	V2.0 or superior	V02r20 or superior
Touch_Probe1_Neg_Edge	Integer32	*60BB.0H*	V2.0 or superior	V02r20 or superior
Touch_Probe2_Neg_Edge	Integer32	*60BD.0H*	V2.0 or superior	V02r20 or superior
Current_Actual_Value	Unsigned16	*6078.0H*	V2.4 or superior	V03r02_002 or superior
Torque_demand	Integer16	*6074.0H*	V2.5 or superior	V03r03_001 or superior
Torque_Actual_Value	Integer16	*6077.0H*	V2.5 or superior	V03r03_001 or superior

The maximum number of mappable TX PDO is **16.**

The maximum total sum of the TX PDO mappable bytes is **32.**

5.2.2 RX PDO dynamic mappable Objects

RX PDO Name	*Data Type*	*Index.sub*	*Xml (ESI) Version*	*Version*
Controlword	Unsigned16	*6040.0H*	V2.0 or superior	V02r20 or superior
Mode of operation	Integer8	*6060.0H*	V2.0 or superior	V02r20 or superior
B0_Digital_Outputs	Unsigned8	*6200.1H*	V2.0 or superior	V02r20 or superior
Target_Position	Integer32	*607A.0H*	V2.0 or superior	V02r20 or superior
vl_Target_Velocity	Integer16	*6042.0H*	V2.1 or superior	V02r20 or superior
Interpolation_data_record_x1	Integer32	*60C1.1H*	V2.0 or superior	V02r20 or superior
Target_Velocity	Integer32	*60FF.0H*	V2.0 or superior	V02r20 or superior
Profile_Velocity	Unsigned32	*6081.0H*	V2.0 or superior	V02r20 or superior
Feedback_Boost_Current	Unsigned16	*2230.12H*	V2.0 or superior	V02r20 or superior
Touch_Probe_Function	Unsigned16	*60B8.0H*	V2.0 or superior	V02r20 or superior
Max_Current	Unsigned16	*2005.2H*	V2.0 or superior	V02r20 or superior
Boost_Current	Unsigned16	*2005.3H*	V2.0 or superior	V02r20 or superior
Min_Current	Unsigned16	*2005.1H*	V2.0 or superior	V02r20 or superior
Max_Profile_Velocity	Unsigned32	*607F.0H*	V2.0 or superior	V02r20 or superior
Profile_Acceleration	Unsigned32	*6083.0H*	V2.0 or superior	V02r20 or superior
Profile_Deceleration	Unsigned32	*6084.0H*	V2.0 or superior	V02r20 or superior
Home_Offset	Integer32	*607C.0H*	V2.0 or superior	V02r20 or superior
Homing_Method	Integer8	*6098.0H*	V2.0 or superior	V02r20 or superior
Speed_Search_for_Switch	Unsigned32	*6099.1H*	V2.0 or superior	V02r20 or superior
Speed_Search_for_Zero	Unsigned32	*6099.2H*	V2.0 or superior	V02r20 or superior
Homing_Acceleration	Unsigned32	*609A.0H*	V2.0 or superior	V02r20 or superior
Nominal_Current	Unsigned16	*2005.4H*	V2.4 or superior	V03r02_002 or superior
Motor_Step_Angle	Unsigned16	*2012.1H*	V2.4 or superior	V03r02_002 or superior
Feed_Constant_feed	Unsigned32	*6092.1H*	V2.4 or superior	V03r02_002 or superior
Target_Torque	Integer16	*6071.0H*	V2.5 or superior	V03r03_001 or superior
Max_Torque	Unsigned16	*6072.0H*	V2.5 or superior	V03r03_001 or superior
Torque_Slope	Unsigned32	*6087.0H*	V2.5 or superior	V03r03_001 or superior
Torque_Window	Unsigned16	*2B08.0H*	V2.5 or superior	V03r03_001 or superior
Torque_Window_time	Unsigned16	*2B09.0H*	V2.5 or superior	V03r03_001 or superior
Interpolation_time_period	Unsigned8	*60C2.1H*	V2.5 or superior	V03r03_001 or superior
Homing_input_Settings	Unsigned32	*2081.0H*	V2.6 or superior	V03r05_005 or superior
DS402_Working_settings	Unsigned32	*2084.0H*	V2.6 or superior	V03r05_005 or superior
Drive_Working_Settings	Unsigned16	*2200.2H*	V2.6 or superior	V03r05_005 or superior
B0_Digital_Inputs_Polarity	Unsigned16	*2200.7H*	V2.6 or superior	V03r05_005 or superior
Feedback_Limit_Speed	Unsigned16	*2230.1BH*	V2.7 or superior	V03r07_002 or superior

The maximum number of mappable RX PDO is **16.**

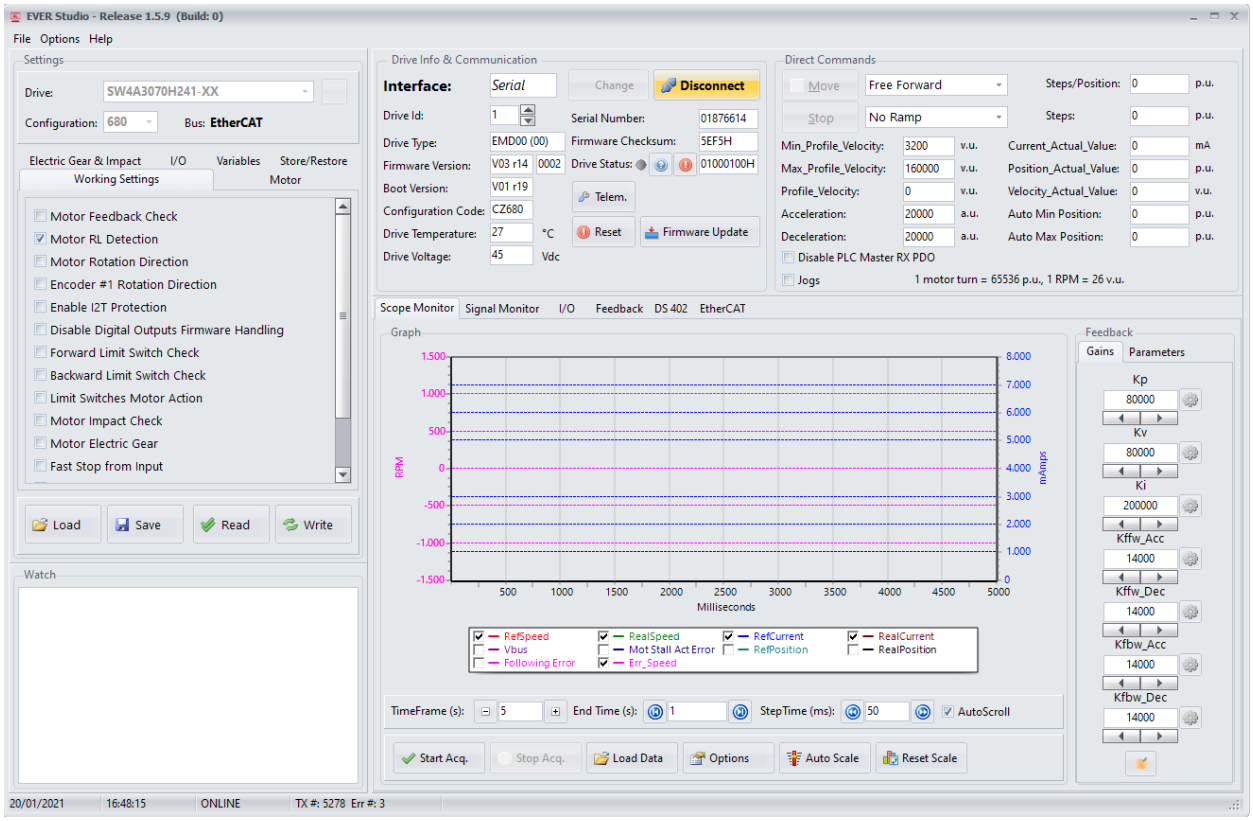
The maximum total sum of the RX PDO mappable bytes is **32.**

5.3 Store Dynamic Mapping in NVRAM

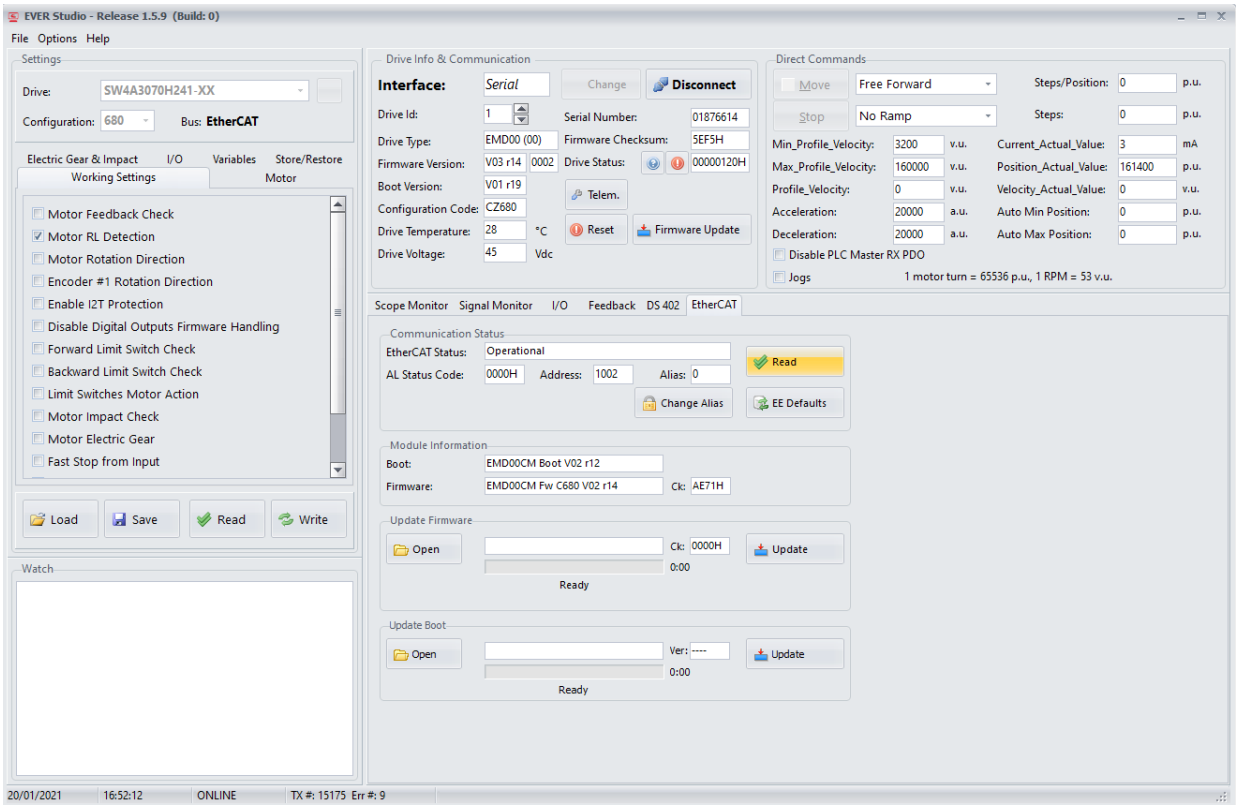
It is possible to store the dynamic PDO Mapping to NVRAM writing *1010.1H* object with **0x7070616D** value.
At next switch on you don't need to send the RX/TX PDO configuration mapping via SDO.

5.4 PC Support Tool

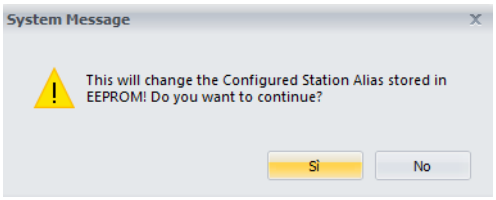
It is possible to check/update the firmware of the EtherCAT Board using the *'EVER Studio'* and connecting to Service Serial Interface (with a baud rate of 115200) provided by the drive. From the main window it is possible to access to settings of the power board of the drive and update the firmware if necessary.



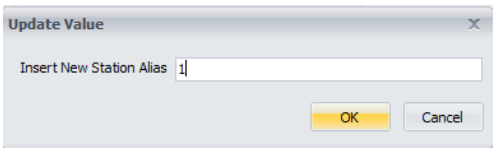
Under the TAB *'EtherCAT'* it is possible to see information of the EtherCAT communication board of the drive and update the firmware (or Boot) if necessary.



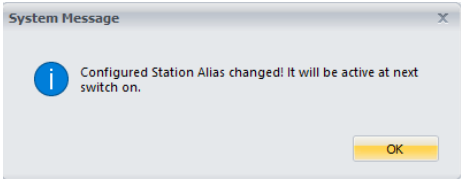
It is possible also change Alias value (press Change Alias button):



insert the desired value and press OK



the new value will be active at next switch on.

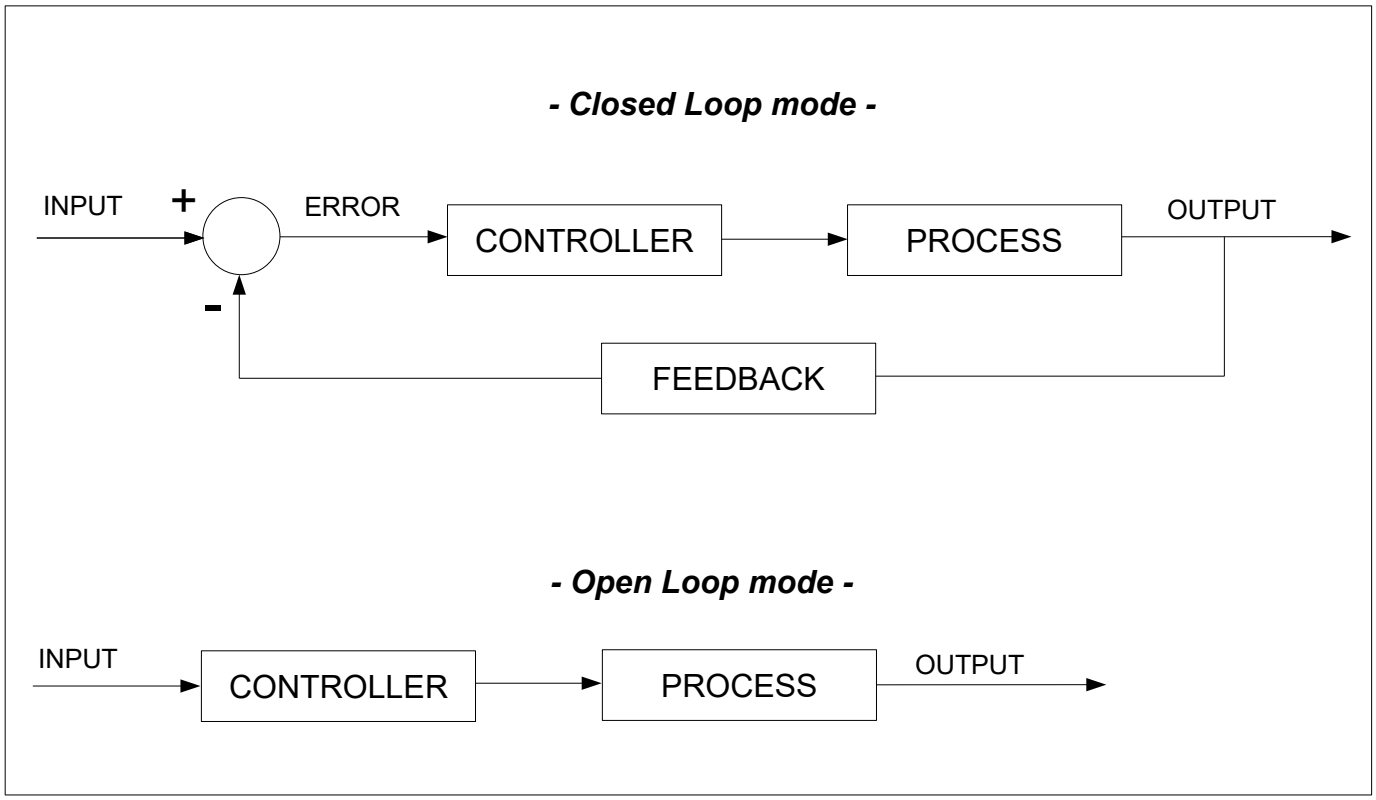


5.5 EtherCAT Slave Information (ESI)

The EtherCAT Slave Information file (XML file) is available for configuring the EtherCAT master. It can be found on the CD-ROM / USB KEY supplied with the drive software kit under the \XML directory.

6.0 Motor Control modes

There are two modes to control motor that can depend from motor type and if feedback Sensor is used or not. The control mode without feedback is called *Open Loop* and with feedback is called *Closed Loop*. For controllers with feedback the measured actual value is constantly compared with a target set point value and any deviation between these values is readjusts according to the control feedback parameters. In Closed Loop the motor can be controlled in position, velocity and torque.



Motor Control mode	*Stepper motor (Titanio drives)*	*BLDC motor (Platino drives)*	*BLAC motor (Vanadio drives)*
Open Loop	***YES***	***NO***	***NO***
Closed Loop	***YES***	***YES***	***YES***

Depending on the type of motor and controller drive the feedback sensor can be an Incremental Encoder, Absolute Encoder , Hall Sensors, etc.

Feedback Sensor Type	*Stepper motor (Titanio drives)*	*BLDC motor (Platino drives)*	*BLAC motor (Vanadio drives)*
Hall Sensors	***NO***	***YES***	***YES***
Hall Sensors + Encoder	***NO***	***YES***	***YES***
Encoder	***YES***	***YES***	***YES***

Depending on the control mode various operating modes can be used. The following table indicates all possible operation types in the various control modes :

DS402 Operating mode	*Control mode*	
	Open Loop	*Closed Loop*
(hm) *Homing_mode*	*YES*	*YES*
(pp) *Profile_Position_Mode*	*YES*	*YES*
(vl) *Velocity_mode*	*YES*	*YES*
(tq) *Profile_Torque_mode*	*NO*	*YES*
(pv) *Profile_Velocity_mode*	*YES*	*YES*
(ip) *Interpolated_Position_Mode*	*YES*	*YES*
(csp) *Cyclic_Synchronous_Position_mode*	*YES*	*YES*
(csv) *Cyclic_Synchronous_Velocity_mode*	*YES*	*YES*
(cst) *Cyclic_Synchronous_Torque*	*NO*	*YES*

The *Profile_Torque_mode* (tq) and *Cyclic_Synchronous_Torque* (cst) cannot work in the Open Loop control mode because necessarily require a Feedback sensor.

The bit#4 of 'Drive_Working_Settings' obj (*2200.2H*) is used to enable Closed Loop mode or Open Loop mode.

6.1 Open Loop

The Open Loop mode can be used only for stepper motors and drives. The motor moves without step losses until the maximum motor torque is not reached. Due to the lack of feedback, to compensate for the torque variations required from the motor, the controller will generally have to supply the maximum current value to the motor to satisfy the required torque condition in the worst case of the duty cycle.

Reset bit#4 of '*Drive_Working_Settings*' obj (2200.2H) to enable *Open Loop* mode.

Related objects :

Object name	Note
Motor_Pole_Pairs	
Motor_Step_Angle	
Min_Current	
Max_Current	
Boost_Current	
Nominal_Current	
Motor_R	The values can be omitted if it's enabled the automatic motor parameters detection. (bit9 *of Drive_Working_Settings_Extended obj)*
Motor_L	
Drive_Working_Settings	
Drive_Working_Settings_Extended	

6.2 Closed Loop

The ***Closed Loop*** feature permits to grant the perfect synchronization between motor rotor and stator (the motor doesn't lose steps) and optimize the motor efficiency since only the really needed current is supplied. For the motor feedback feature is necessary to have a Feedback sensor directly on the rear shaft of a double shaft motor.

With regard to an 'Open Loop' controller :

- Reliable positioning without synchronization loss;
- Keeps the original position stable and recovers it automatically in case of positioning errors caused by external factors such as mechanical vibrations;
- 100% use of the motor torque;
- Capacity to operate at high velocity related to the current control, which is adjusted depending on the load variations, where the normal systems in open loop use a constant current control at all velocities without considering the load variations.

To enable *Closed Loop* mode :

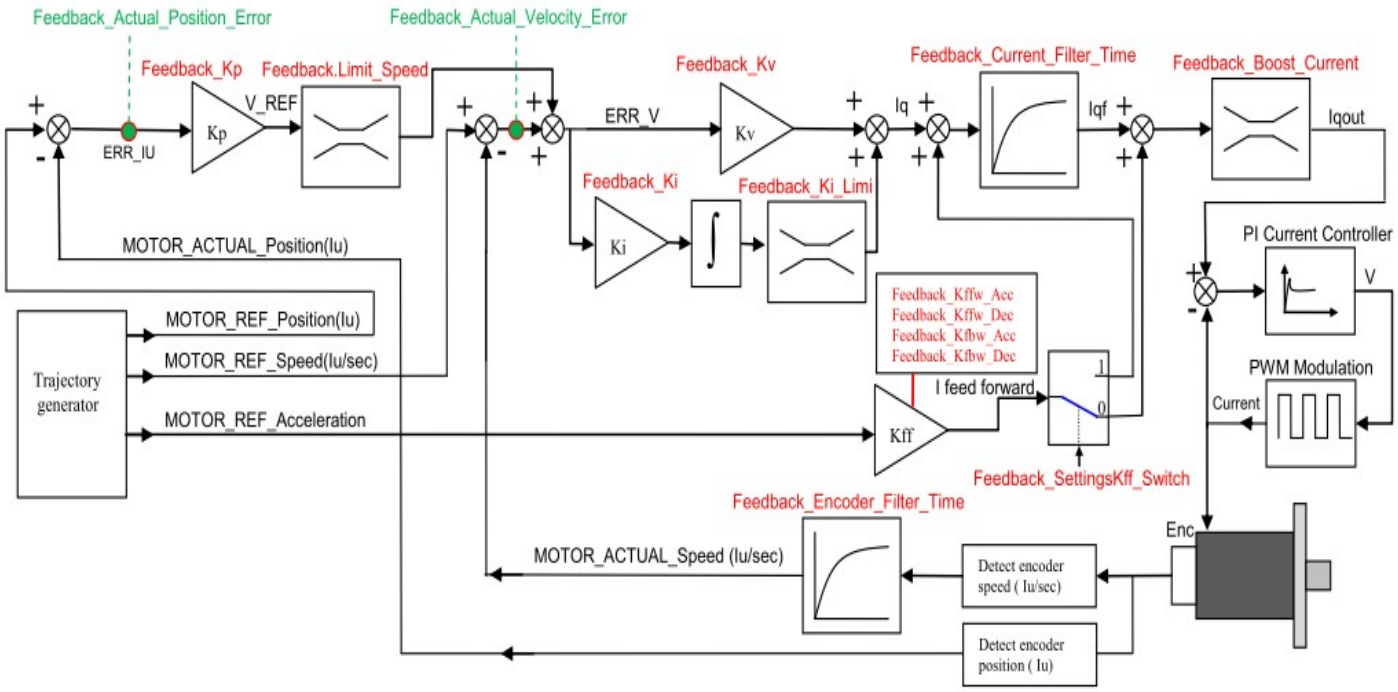
- Define Feedback parameters of PID regulator (see *§6.2.1* and *§6.2.2*)
- Define Feeedback Sensor Type (Feedback_Settings , bit8÷bit11)
- Define Feedback Sensor Resolution depending from Feedback Sensor Type
- Set 'Sensor_selection_code = -1'
- Set bit#4 of 'Drive_Working_Settings' obj (2200.2H).

Related objects :

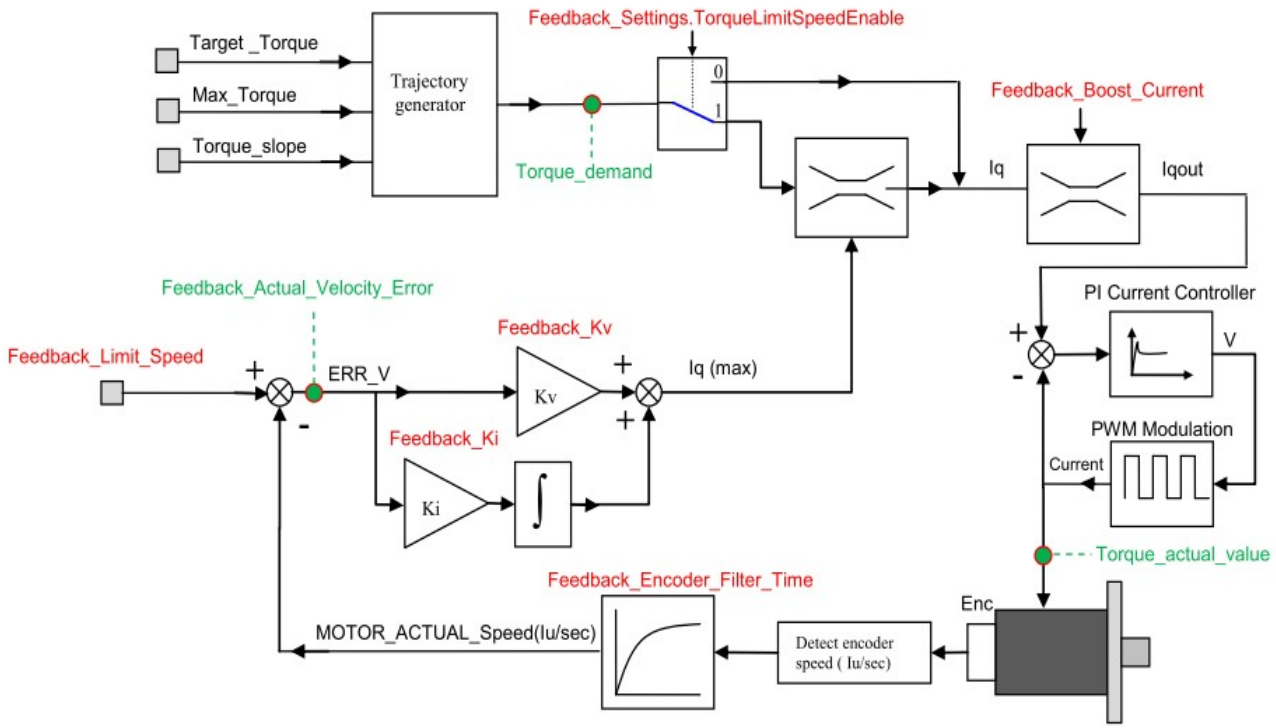
Object name	Note
Motor_Pole_Pairs	
Motor_Step_Angle	
Min_Current	Used to keep the motor in torque when the closed loop is disabled for example at switch-on or when happen the Feedback Error Alarm
Nominal_Current	
Motor_R	The values can be omitted if it's enabled the automatic motor parameters detection. (bit9 *of Drive_Working_Settings_Extended obj)*
Motor_L	
Drive_Working_Settings	
Drive_Working_Settings_Extended	
Feedback_Source_PPR	
Feedback_Kp	
Feedback_Kv	
Feedback_Ki	
Feedback_Ki_Limit	
Feedback_Kalfas	
Feedback_Encoder_Filter_Time	
Feedback_Kffw_Acc	
Feedback_Kffw_Dec	
Feedback_Kfbw_Acc	
Feedback_Kfbw_Dec	
Feedback_Iq_min	
Feedback_Iq_Min_Threshold	
Feedback_Boost_Current	
Feedback_Current_Filter_Time	
Feedback_Position_Error_Limit	
Feedback_Calibration_Speed	
Feedback_Limit_Speed	
Feedback_Calibration_Current	
Feedback_Velocity_Error_Limit	
Feedback_Settings	
Feedback_Status	
Feedback_Actual_Position_Error	
Feedback_Actual_Velocity_Error	
Target_Torque	
Max_Torque	
Torque_slope	
Torque_demand	
Torque_actual_value	
I2T_Peak_Current	
I2T_TMax_Peak_Current	
I2T_Current_Limit	
I2T_Current_Actual	
Hall_Sensors_Sequence_Settings	

Hall_Sensors_Sequence_Detected	
Hall_Sensors_Status	
Hall_Sensors_Position	
BiSS_Encoder_Config	

6.2.1 Schematic for Position & Velocity modes



6.2.2 Schematic for Torque mode



7.0 Special Functions

The Titanio-Platino-Vanadio Drives implement some special functions described below.

7.1 Motor R,L detection

The Titanio-Platino-Vanadio Drives implement the procedure to detect the Motor R and Motor L of the motor. For this procedure are required the *Nominal_Current* of the motor and bit9 of *Drive_Working_Settings_Extended* object must be set to 1. At the end of procedure, the Motor R,L detected are stored into *Motor_L_detected* and *Motor_R_detected* objects and used for motor current regulation.

The procedure is activated only the first time when the system enters operating mode.

If bit#9 of *Drive_Working_Settings_Extended* is equal to 0 then Motor R,L are not automatically detected and their values must be set by mean of *Motor_R* and *Motor_L* objects.

Related objects :

Object name	Note
Nominal_Current	
Motor_R	
Motor_L	
Motor_L_detected	
Motor_R_detected	
Drive_Working_Settings_Extended	

7.2 I²T Motor overload Protection

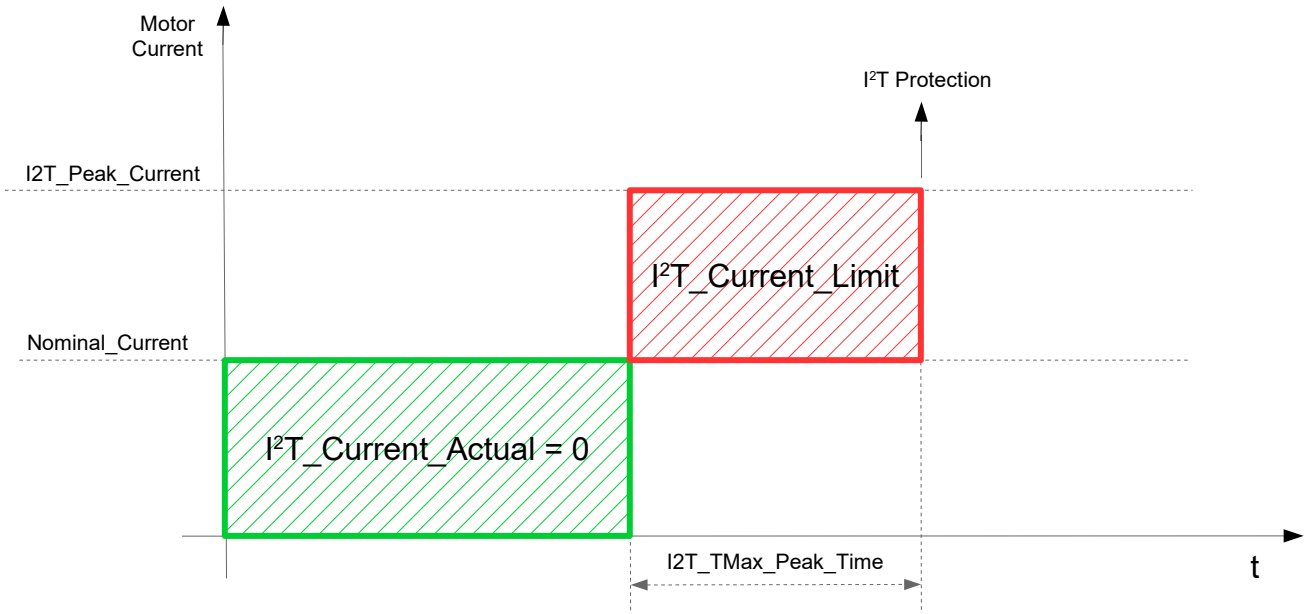
The energy dissipated by a motor is proportional to the square of the current circulating through it and to the time this current lasts circulating through it. The nominal current is the current that a motor can stand in a continuous manner without exceeding its thermal limits. Any current above the nominal current can create an accumulation of thermal energy in the motor. If this process of accumulating thermal energy exceeds the cooling motor's ability to dissipate it, the system is bound to reach its thermal limits, and permanent damage to the motor.

The **I²T** (defined in *Ampere² per second*) indicates the excess of thermal energy and is an indirect magnitude proportional to the energy dissipated.

The **I²T** protection is a control mechanism used to ensure that the integral of the power dissipated by the motor in the form of thermal energy does not exceed its thermal limits. Under nominal current value, the motor can work for infinite time but once actual motor current crosses the motor nominal current, the algorithm starts to integrate the excess of thermal energy.
The parameters used for I2T control algorithm are :

I2T_Current_Limit = [(*I2T_Peak_Current*)² - (*Nominal_Current*)²] · *I2T_Tmax_Peak_Current*
I2T_Current_Actual += [(*Current_Actual_Value)²* - *(Nominal_Current)²*] · Ts *(sampling time)*

If *I2T_Current_Actual* > *I2T_Current_Limit* the device enter in the Fault state.



To enable/disable I2T Protection control mechanism is used Bit#10 of *Drive_Working_Settings_Extended* object.
For ***Closed Loop*** Bit#4 of *Drive_Working_Settings* object must be set.

Related objects :

Object name	Note
Nominal_Current	
I2T_Tmax_Peak_Current	
I2T_Peak_Current	
I2T_Current_Limit	
I2T_Current_Actual	
Drive_Working_Settings_Extended	
Drive_Working_Settings	

7.3 Motor Stall detection

This feature allows to recognize the Motor Stall without using an external sensor (for example Incremental Encoder or Absolute Encoder). To enable 'Motor Stall' feature set bit#13 of *Drive_Working_Settings_Extended* object.

The *Closed Loop* feature (bit#4 of *Drive_Working_Settings* object) and 'Motor Stall detection' feature (bit#13 of *Drive_Working_Settings_Extended* object) cannot be both active at the same time. Try to enable *Closed Loop* feature with 'Motor Stall detection' feature already enabled or vice versa will issue a Fault condition (bit#14 of *Feedback_Status* object will set to 1).

The Motor Stall detection consists in comparing the theoretical angle and estimated angle of motor rotor position :

If abs(*Motor_Stall_Actual_Err_Angle*) > *Motor_Stall_Max_Err_Angle* then 'Motor Stall detected' !!

When the motor stall is detected :

- the motor is stopped immediately.
- *Min_Current* object value is applied to the motor.
- Bit#10 (Motor_Stall_detected) of Feedback_Status *object is set to 1.*
- A Emergency Telegram will be send and drive will enter into Fault State.

Related objects :

Object name	Note
Nominal_Current	
Min_Current	
Max_Current	
Boost_Current	
Motor_Stall_Filter_Time	
Motor_Stall_Max_Err_Angle	
Motor_Stall_Actual_Err_Angle	
Drive_Working_Settings_Extended	
Drive_Working_Settings	

Note :
The 'Motor Stall detection' feature is available only with firmware version V02r74 or superior.

7.4 Limit Switch Check

Outside of the *Homing_mode*, it is possible to enable a continuous check of intervention of *Forward Limit Switch* or/and *Backward Limit Switch*. If the limit switch intervenes and the motor is running in the direction of limit switch (forward direction for *Forward Limit Switch* and backward direction for *Backward Limit Switch)* then the motor will stop.

The digital input used for *Forward Limit Switch* or/and *Backward Limit Switch* is defined by mean of *Drive_Inputs_Setting object*. The digital input level consider for intervantion of limit switch is defined by mean of *Drive_Inputs_Level* object.

Depending on bit3 of *DS402_Working_Settings* (*2084.0H*) the intervention limit switch can be considered as a Fault or not.

To enable this feature set bit0 and/or bit1 of *Drive_Working_Settings* (*2200.2H*) object.

Related objects :

Object name	Note
Drive_Working_Settings	
DS402_Working_Settings	
Drive_Inputs_Level	
Drive_Inputs_Setting	
Statusword	

7.5 Fast Stop

This feature allows to check continuously the intervention of *Fast Stop* input. If the *Fast Stop* input is active the state machine is kept to 'Switch On Disabled' until *Fast Stop* input is not active.

The digital input used for *Fast Stop* *is defined* by mean of *Drive_Inputs_Setting* *object.* The digital input level consider for intervantion of fast stop is defined by mean of *Drive_Inputs_Level* object.

To enable this feature set bit2 of *Drive_Working_Settings* (*2200.2H*) object.

Related objects :

Object name	Note
Drive_Working_Settings	
Drive_Inputs_Level	
Drive_Inputs_Setting	
Statusword	

7.6 Brake Control

By mean of *Brake_Control_Settings* object is possible to enable/disable (bit0) the Automatic Brake Control, define the digital output used for the Brake (bit4÷bit7 and bit3) and the details of the Brake control handling (bit1 and bit2).

The following descriptions are related to Automatic Brake Control enabled (bit0 = 1 of *Brake_Control_Settings* object).

When the drive is in the '*Switched On*' state or *'Ready to switch on'* state or *'Switch on disabled'* state or *'Not ready to switch on'* state or *'Fault'* state then the Brake is close (active) and the settings of the bit1 and bit2 of *Brake_Control_Settings* object are not considered. When the drive enters in one of the indicated states the Brake is immediately closed (actived) and the motor current is switched off.

When the drive is in the '*Operation enable*' state then the state close or open of the Brake depends from settings of the bit1 and bit2 of *Brake_Control_Settings* object :

bit2	bit1	bit0	'Operation mode' state
0	0	1	The Brake is open (released).
0	1	1	The procedure described below can be used when is required an automatic handling of the closing and opening of the Brake during the stop and movement of the motor. The Brake is closed (actived) when the motor is at standstill and open (released) when motor is running. The following automatic sequence is performed from the drive when the Brake is closed : - after the motor stop the time defined by *2C01.0H* is allowed to elapse. - the Brake is closed (actived). - the time defined by *2C02.0H* is allowed to elapse. - the motor current is switched off. The following automatic sequence is performed from the drive when the Brake is open : - the motor current is switched on. - the time defined by *2C03.0H* is allowed to elapse. - the Brake is open (released). - the time defined by *2C04.0H* is allowed to elapse. - the drive can perform motor movements. Note : This type of Automatic Brake Handling can be used for *Profile_Position_Mode* and *Homing_mode.* For other modes is suggested to use the Brake handling as defined by settings of the bit2 = 1 and bit1 = 0 (the Master Ethercat / CANopen controls directly the Brake, included the timing and switch off and switch on of the motor current). For example, with *Cyclic_Synchronous_Position_mode* the slave drive does not know when it will receive the first step to move and therefore could apply the automatic sequence to open the brake (described above) simultaneously with the reception of the subsequent movement steps, risking a block of the motor movement.
1	X	1	The state of Brake can be changed by *B0_Digital_Outputs* object (if bit3 = 0 of *Brake_Control_Settings* object) or *B1_Digital_Outputs* object (if bit3 = 1 of *Brake_Control_Settings* object). The Master Ethercat / CANopen can control directly the Brake during 'Operation Enable' state and must to handle the sequence concerning the open/close of the Brake, included the timing of the Brake and switch off and switch on of the motor current (*2C01.0H*, *2C02.0H*, *2C03.0H*, *2C04.0H* objects are not considered in this case). To switch off the motor current : - for *Open Loop* mode the objects *Min_Current* , *Max_Current* , *Boost_Current* must be set to 0 value. - for *Closed Loop* mode the objects *Feedback_Boost_Current* must be set to 0 value. To switch on the motor current restore the values previously used for the objects *Min_Current* , *Max_Current* , *Boost_Current* , *Feedback_Boost_Current* before they were forced to 0 by mean of switch off motor current.

The Brake Control is available with firmware V03r20 or superior.

Related objects :

Object name	Note
Brake_Control_Settings	
Brake_Control_Time1_Close_Brake	
Brake_Control_Time2_Close_Brake	
Brake_Control_Time1_Open_Brake	
Brake_Control_Time2_Open_Brake	
B0_Digital_Outputs	
B1_Digital_Outputs	

7.7 Braking Resistor Function

When load is accelerated electrical energy is converted into mechanical energy. During deceleration the conversion is reversed. If during the braking too much energy is generated an external resistor (or internal if the driver is provided with), known as braking resistor, is used to dissipate the excess energy preventing that the driver's overvoltage protection wil shut down the driver.

To drive the external resistor the driver must be equipped with a dedicated internal circuit, refer to the hardware manual of a specific drive to check if an external braking resitor can be used.

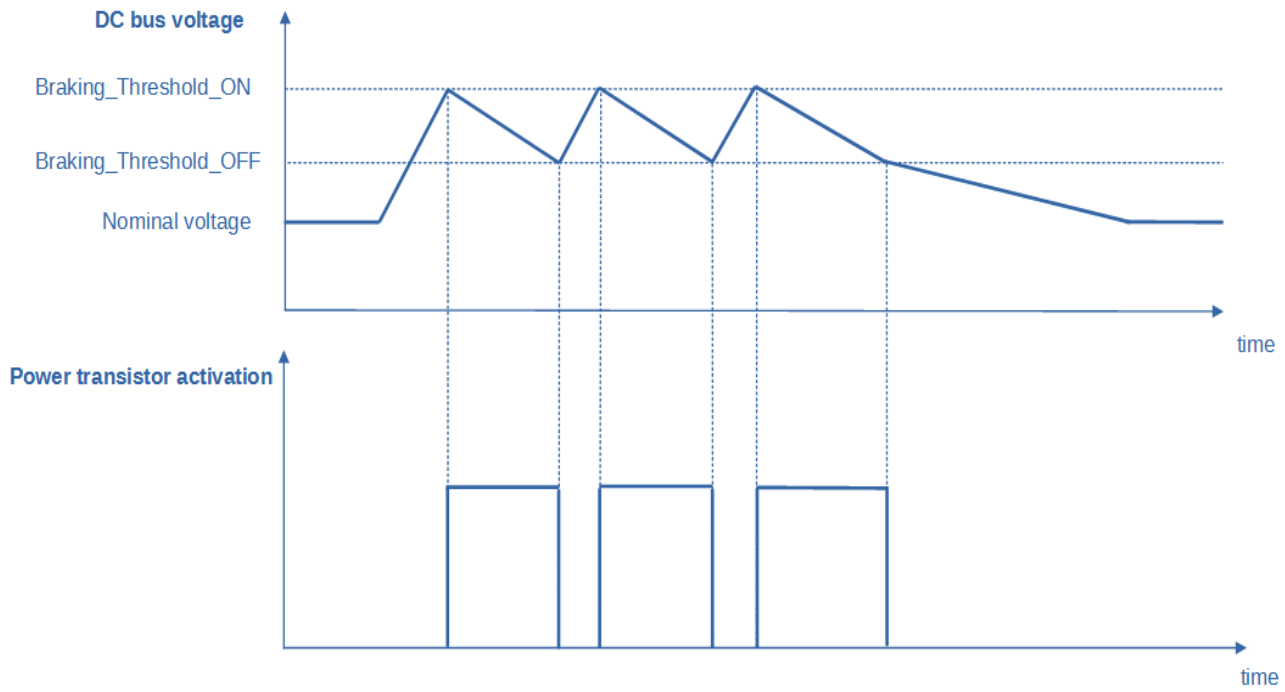
Operation of braking resitor function is quite simple: if during a braking the internal voltage goes beyond the "*Braking_Threshold_ON*", a power transistor is turned on and the external braking resistor starts to shunt energy until the internal voltage returns below the "*Braking_Threshold_OFF*".

The driver protects the braking circuit against short circuit, and uses I2T peak current/time algorithms to protect both the braking resistor and internal transistor.

To make the protection work properly one must set the following parameters:

- *Braking_Resistor_Value* : ohmic value of the resistor expressed in ohms.
- *Braking_Resistor_Power* : rated power of the resistor expressed in watts
- *Braking_Resistor_Overload_Time* : maximum time the braking resistor can withstand at the power peak expressed in tenths of milliseconds

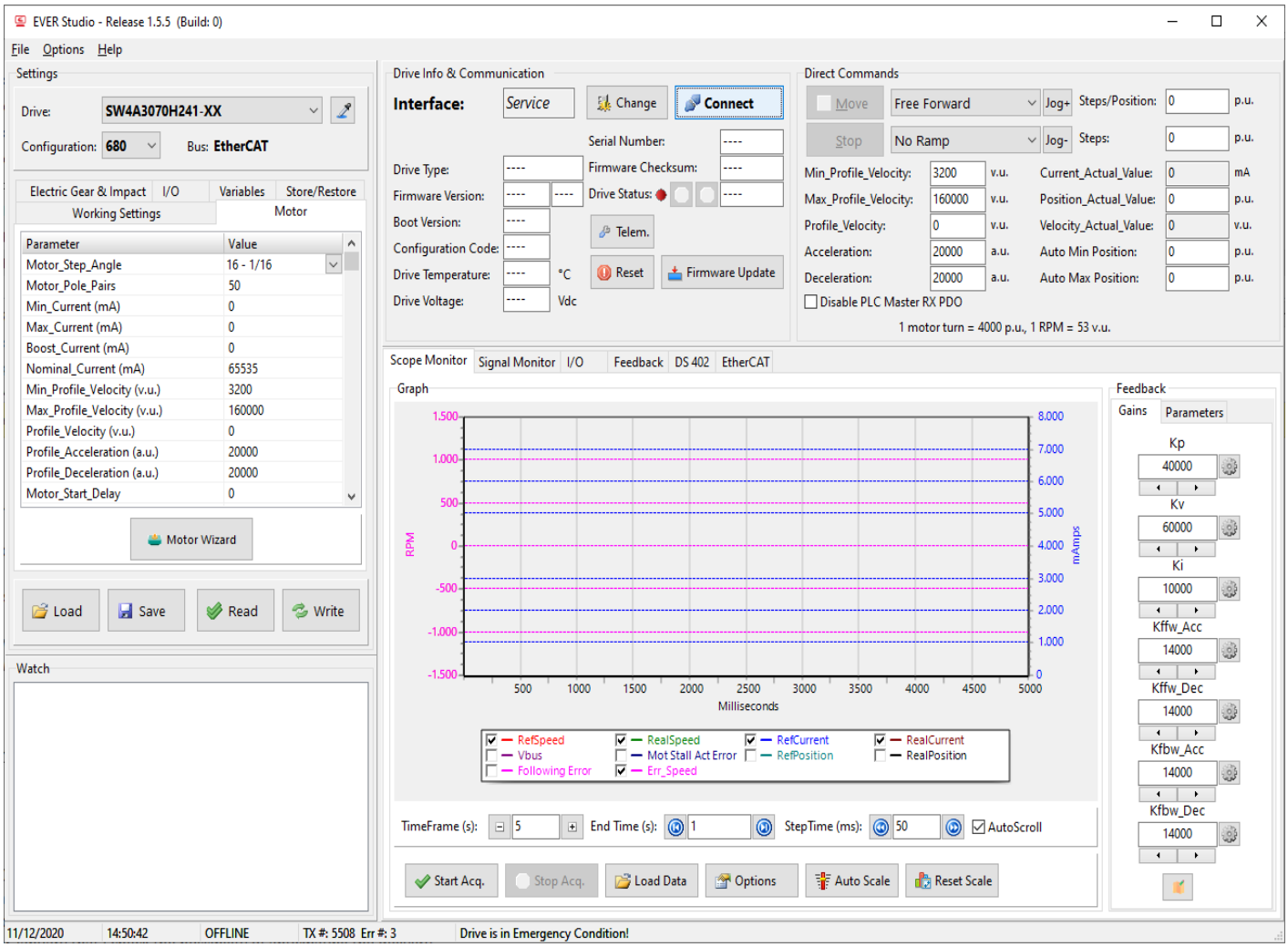
The peak power can be estimated as (Braking_Threshold_ON^2) / Braking_Resistor_Value.



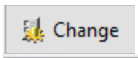
8.0 EVER STUDIO Tool

The EVER STUDIO (*https://www.everelettronica.com/en/products/software/configurators/ever-studio*). is a complete software Tool for PC Windows used for the configuration and diagnostic of 'Titanio-Platino-Vanadio' drives. Through simple windows and wizards it is possible to monitor and set all the operating parameters required for the drives and application. The Tool integrates a digital oscilloscope to watch and sample in real time the main parameters of the motion profile used for the motor control. It is also available a specific window (and sub-windows) dedicated to DS402 profile and related functions. The connection between the PC and the drive can be made via dedicated Serial Interface (Service Interface on drive) with a specific EVER cable.

Main Window

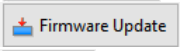


Drive Info & Communication Window

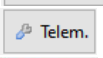


Change the PC – Drive communication settings

Connect / Disconnect to drive



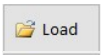
Update the drive's Firmware



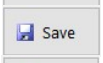
Open the Telemetry Window

Software Reset of the drive.

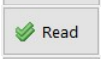
Settings Window



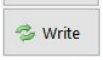
Load a previously saved configuration file



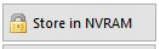
Save the current parameter's value in a configuration file



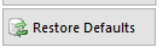
Read the parameter's value from drive



Write the current parameter's value to the drive



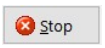
Store the current drive's parameters in NVRAM



Restore the default value for drive's parameters

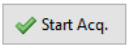
Direct Commands Window

Move the motor according to the selected mode. A warning message will popup that the drive will be set in operation enable modality.

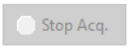


Stop the motor according to the selected mode. A warning message will popup that the drive will be set in operation enable modality.

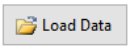
Scope Monitor Window



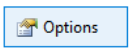
Start the acquisition of some drive's internal variables. Useful for Feedback tuning or to verify the motion behavior.



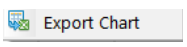
Stop the acquisition.



Load a previously saved acquisition.



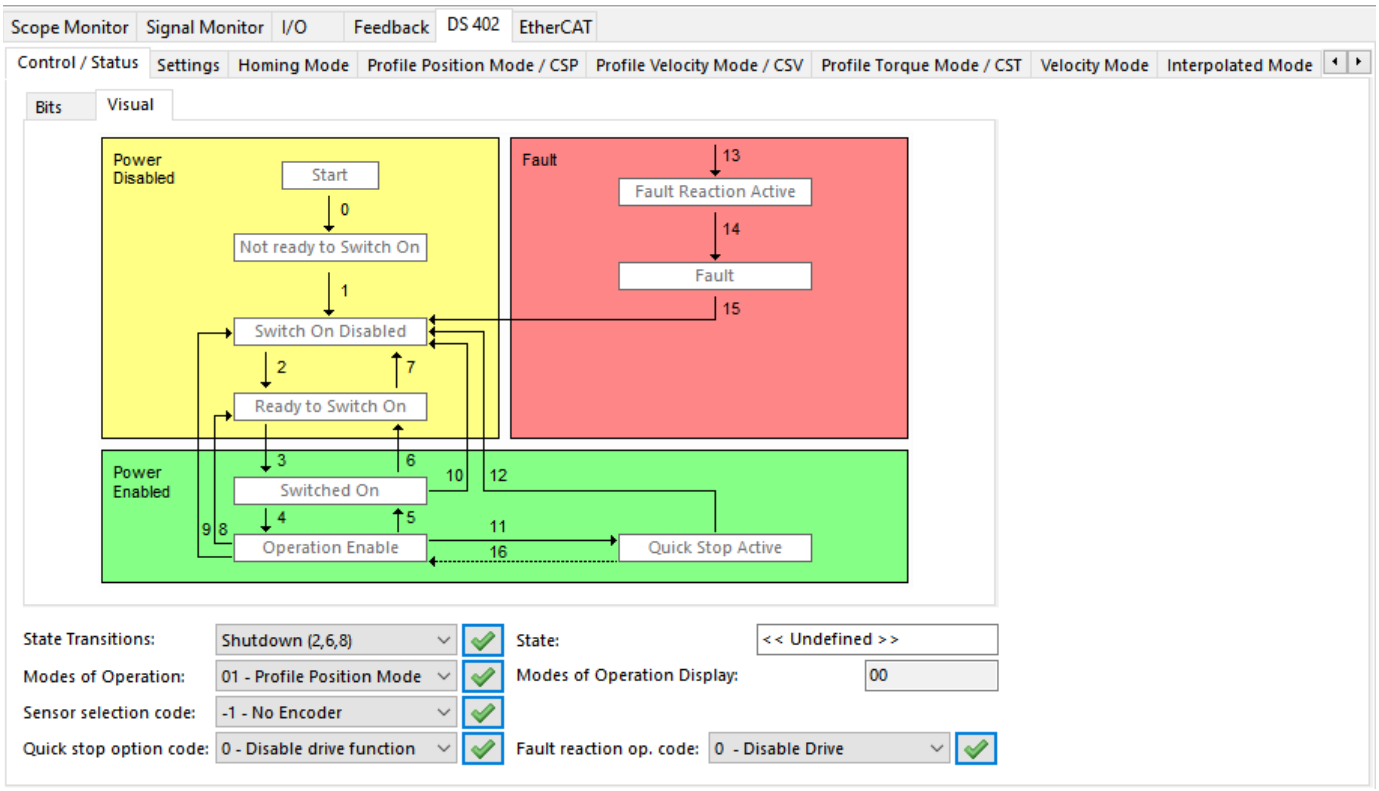
Change the graph / series option.



Pressing left mouse button inside the graph a pop-up menu is showed. The Export Chart function enable the export of the graph values in various formats (Excel,Text,HTML)

DS402 Window

The DS402 Window is useful to check the drive's current state machine condition and to change all the DS402 parameters. If the Master is not connected to the drive is it possible to force the state machine change pressing on the number corresponding to the desired transition.



Objects Index (by name)

Objects Index (by index)