ECAT-2094S

EtherCAT
4-Axis Stepper Motor
Controller/ Driver

User Manual

(Version 1.0)

WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

TRADEMARK

Names are used for identification only and may be registered trademarks of their respective companies.

CONTACT US

If you have any questions, please feel free to contact us via email at:
service@icpdas.com
service.icpdas@gmail.com

Revision

Revision	Date	Description	Author
1.00	05.09.2018	Initial version	M. K.

Contents

1 Product Overview

1.1 Introduction

The ECAT-2094S stepper motor controller is a cost-effective, two-phase bipolar stepper driver. The ECAT-2094S simultaneously controls up to four stepper motors. A motor voltage range between 5 and 46V DC and a maximum motor coil current of 1.5A is being supported. The running motor current, microstep resolution and other motion parameters are software selectable.

Two-phase bipolar stepper motors can be directly connected to the ECAT-2094S device. The device is designed to operate in a open loop. Configuration and motion control has to be done by the EtherCAT master and the application program. The torque and step control is done by an internal stepper motor driver IC. Each stepper motor is being independently controlled by a separated driver IC. The four driver ICs are not synchronized and work independently from each other. An integrated ramp generator automatically calculates the acceleration and deceleration distance. The motion controller drives the motor to the target position or accelerates the motor to the target velocity. All motion parameters can be changed on the fly. A minimum set of configuration data consists of acceleration, deceleration and maximum motion velocity. After receiving the target position the motor driver starts controlling the motion movement.

The ECAT-2094s has four integrated incremental encoder interfaces. Four 32 bit high frequency encoder counter counts the input signal of external incremental encoders. The encoder can for example be used for homing purposes and for consistency checks.

High resolution of up to 256 microsteps per full step is supported for a ensuring smooth and precise motor operation.

For each motor two digital input channels are provided. The digital inputs can be set to act as a simple DI, as a left and right hardware limit switch which automatically stops the motor when activated, or a latch trigger for latching the current motor and encoder position.

The module must be supplied by three power sources. Two motor supply and a 24Vdc control supply. Two motors share one power supply.

1.2 Technical Data

- Supports 4 stepper motor (2-phase bipolar)
- Stepper motor are controlled in an open loop operation
- Programmable coil current level: up to 1.5 A
- Programmable microstep size: maximum 256 microsteps per full step
- Voltage range of the motor output: 5 to 46V_{DC}
- 4 x Encoder interfaces (A, B, Z), differential
- 8 x Digital input. Two DI channels for each axis: reference switch input, latch input
- 2 x Digital output
- Automatic current reduction to reduce heat when motor is not moving
- Drive protection:
  - Over-temperature
  - Under voltage
  - Short circuit
- Optically isolated I/O
- LED indicators for I/O, EtherCAT and motion status
- Internal memory for storing configuration data
- EtherCAT:
  - 2 x RJ-45 bus interface
  - Distance between stations up to 100 m (100BASE-TX)
  - Support daisy chain connection
  - EtherCAT conformance test tool verified
  - Supports Free-Run, SM-Synchron and Distributed Clock (DC) operation modes
- Removable terminal block connector

1.3 Hardware Specification

Item	Specification
Motor Outputs	
Number of outputs	4x stepper motor, 2 phases
Output current	1.5A
Voltage range of the motor output	5 to 46V_{DC}
Current controller frequency	24.5 kHz
Maximum step frequency	8.388 MHz
Microsteps per step	256, 128, 64, 32, 16, 8, 4, 2
Encoder inputs	
Number of encoder inputs	4x encoder counter (A, B, Z), differential
Maximum encoder pulse frequency	4 MHz
Digital Inputs	
Number of digital inputs	8 (2x limit position for each motor)
Wet contact	• ON voltage level: +10 to 30V_{DC} • OFF voltage level: +5V_{DC} MAX
Photo-Isolation	3750V_{DC}
Digital Output	
Number of digital outputs	2
Output type	Open collector
Load voltage	+5 to 30V_{DC}
Max. load current	100mA
Isolation voltage	3750 V_{DC}
LED Indicators	
Diagnostic LED	Power, EtherCAT status, Digital IO, driving, temperature warning, over-temperature error, phase A and B under-voltage
Communication Interface	
Connector	2 x RJ-45
Protocol	EtherCAT
Distance between stations	Max. 100 m (100BASE-TX)
Data transfer medium	Ethernet/EtherCAT Cable (Min. CAT 5), Shielded
Power	
Input voltage range	20V ~ 30V_{DC}
EMS Protection	
ESD (IEC 61000-4-2)	4 KV Contact for Each Channel
EFT (IEC 61000-4-4)	Signal: 1 KV Class A; Power: 1 KV Class A
Surge (IEC 61000-4-5)	1 KV Class A
Mechanism	
Installation	DIN-Rail
Dimensions (LxWxH) [mm]	181 x 110 x 33 (without connectors)
Case material	Metal
Environment	
Operating temperature	-25℃ ~ 40℃
Storage temperature	-30℃ ~ 80℃
Relative humidity	10 ~ 90%, No Condensation

Table 1: Technical data

1.4 Dimensions

All dimension units are in millimeter.

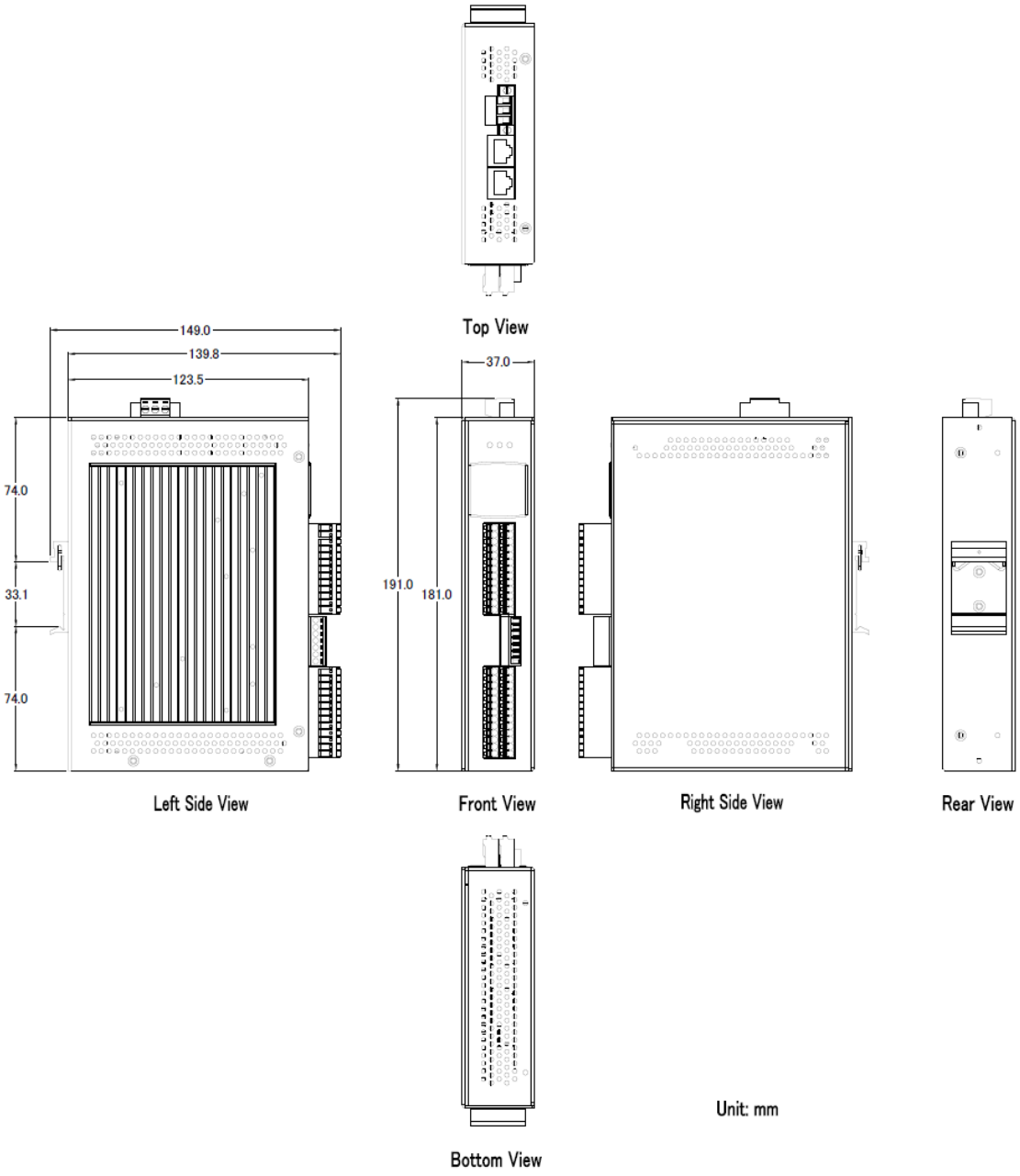


Figure 1: Dimension of the ECAT-2094S housing

2 Scope of Delivery

The shipping package includes the following items:

- 1 x ECAT-2094S
- 4 x 13-pin plug-in connector
- 1 x 3-pin plug in connector (for power supply)
- 1 x 7-pin plug in connector (for motor power supply)
- 1 x Quick Start manual

Figure 2: ECAT-2094S module and Quick Start manual

Note:

If any of these items are missing or damaged, please contact your local distributor. Please keep the original retail box with all retail packaging (Styrofoam, inner boxes, fasteners, etc.) in case you need to return the product.

More Information:

- Product website:
  http://www.icpdas.com/root/product/solutions/industrial_communication/fieldbus/ethercat/motion/ecat-2094s.html

- Manual:
  ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/motion/ecat-2094s/manual/

- XML EtherCAT Slave Information (ESI) file:
  ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/ethercat/slave/motion/ecat-2094s/esi/

- FAQ:
  http://www.icpdas.com/root/product/solutions/industrial_communication/fieldbus/ethercat/ethercat_faqs.html

- Technical support:
  service@icpdas.com

3 Wiring

3.1 LED Definition

The ECAT-2094s provides on the frontside of the connection cap several diagnostic LEDs. Furthermore there are three LEDs to indicate the network status for EtherCAT. The exact meaning of the LED indication is specified in the following tables:

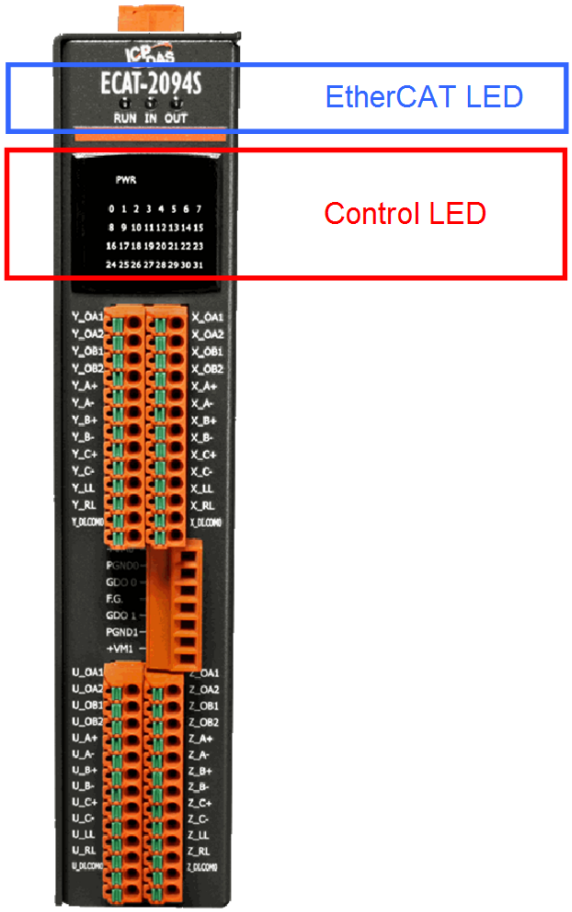


Figure 3: ECAT-2094S front and side view

EtherCAT LED	Color	State	Description
RUN	red		This LED indicates the operation state of the EtherCAT slave:
		Off	Device is in INIT state
		Flashing	Device is in PREOP state
		Single flash	Device is in SAFEOP state Outputs remain in safe state
		On	Device is in OP state

IN	green		Indicates the communication status of the EtherCAT port IN
		Off	No connection
		Flashing	Link and activity (e.g. data exchange with the master)
		On	Link without any activity
OUT	green		Indicates the communication status of the EtherCAT port OUT. Further EtherCAT slave can be connected to the port OUT
		Off	No EtherCAT slaves are connected to port OUT
		Flashing	Link and activity (e.g. data exchange connected slaves)
		On	Link without any activity

Table 2: EtherCAT status indicator

Control LED	Color	Description
*	red	- Power indicator
* * * * * * * * (first row) 0 1 2 3 4 5 6 7	green	- LED 0: Digital input channel 1 (X_LL) - Axis X - LED 1: Digital input channel 2 (X_RL) - Axis X - LED 2: Digital input channel 1 (Y_LL) - Axis Y - LED 3: Digital input channel 2 (Y_RL) - Axis Y - LED 4: Digital input channel 1 (Z_LL) - Axis Z - LED 5: Digital input channel 2 (Z_RL) - Axis Z - LED 6: Digital input channel 1 (U_LL) - Axis U - LED 7: Digital input channel 2 (U_RL) - Axis U
* * * * * * * * (second row) 8 9 10 11 12 13 14 15	green	- LED 8: Digital output (DO 0) - Axis X - LED 9: Digital output (DO 1) - Axis Y - LED 10: EEPROM access error - LED 11: reserved - LED 12: Driving output - Axis X - LED 13: Driving output - Axis Y - LED 14: Driving output - Axis Z - LED 15: Driving output - Axis U
* * * * * * * * (third row) 16 17 18 19 20 21 22 23	green	- LED 16: Motion error - Axis X - LED 17: Motion error - Axis Y - LED 18: Motion error - Axis Z - LED 19: Motion error - Axis U - LED 20: Over temperature error - Axis X - LED 21: Over temperature error - Axis Y - LED 22: Over temperature error - Axis Z - LED 23: Over temperature error - Axis U
* * * * * * * * (fourth row) 24 25 26 27 28 29 30 31	green	- LED 24: Short to ground error - Axis X - LED 25: Short to ground error - Axis Y - LED 26: Short to ground error - Axis Z - LED 27: Short to ground error - Axis U

		- LED 28: Over temperature/open load warning - Axis X - LED 29: Over temperature/open load warning - Axis Y - LED 30: Over temperature/open load warning - Axis Z - LED 31: Over temperature/open load warning - Axis U

Table 3: Diagnostic LEDs

3.2 Connection Interfaces

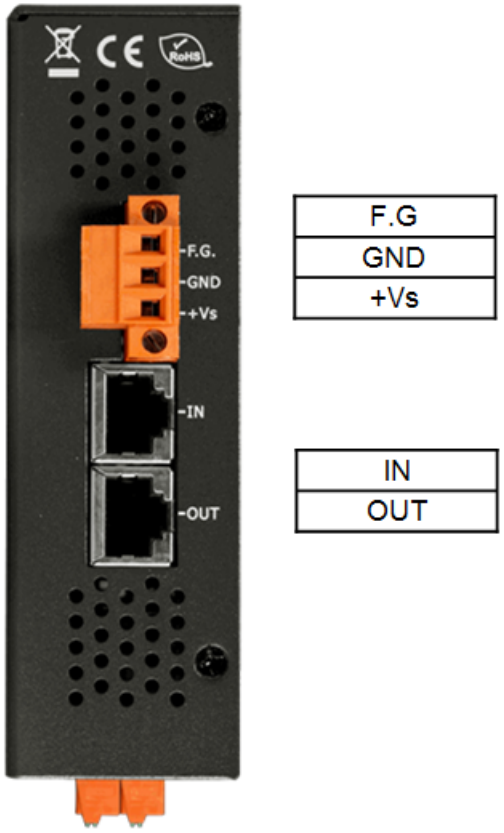
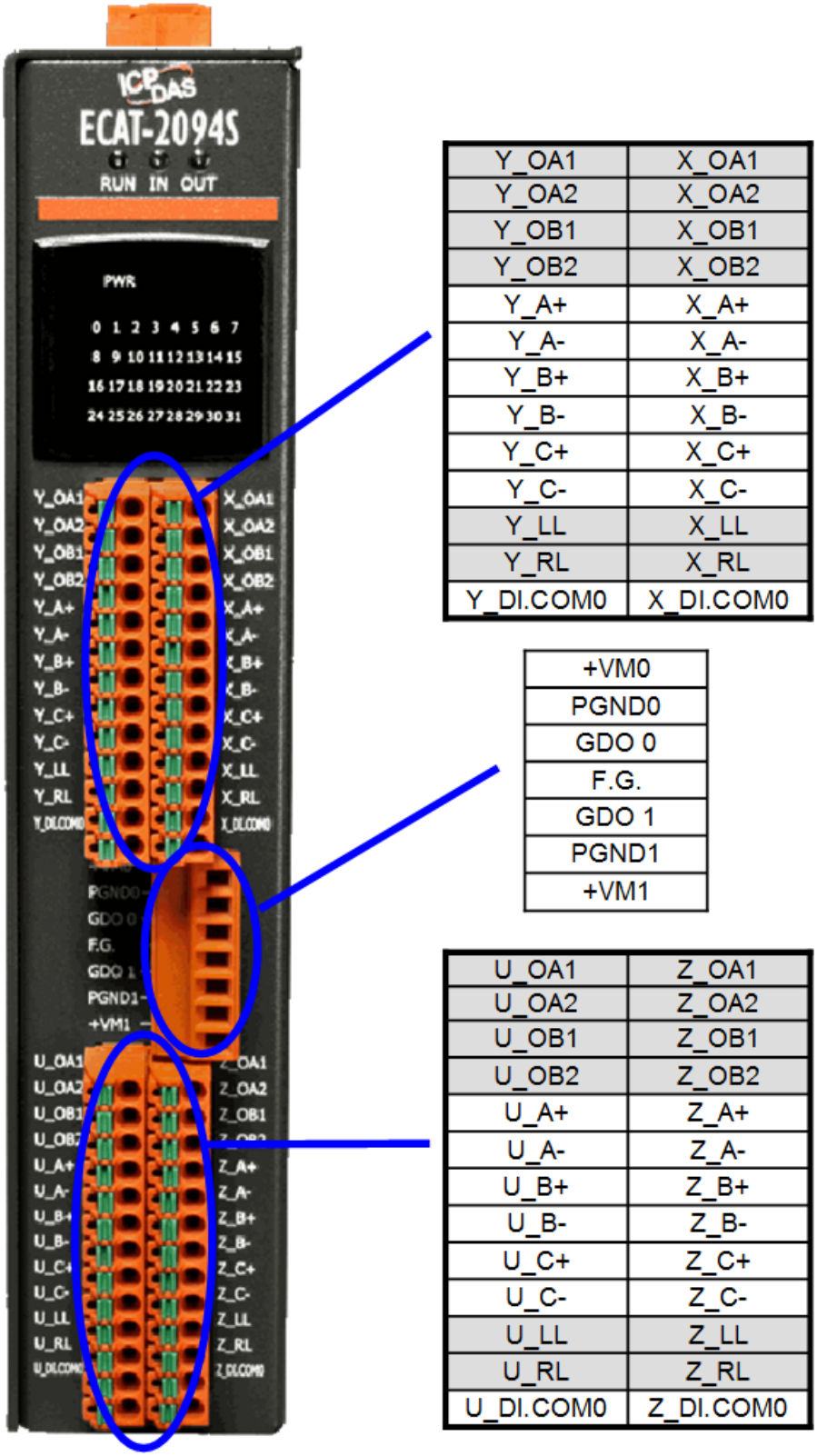


Figure 4: ECAT-2094S side view with power supply and EtherCAT connection

Name	Signal	Description
F.G	Frame ground	
GND	Power supply: Ground 0V (from negative power contact)	Feeding for ECAT-2094S
+Vs	Power supply: +24 V_{DC} (from positive power contact)	Feeding for ECAT-2094S
IN	EtherCAT signal input	Incoming EtherCAT cable
OUT	EtherCAT signal output	Outgoing EtherCAT cable

Table 4: ECAT-2094S power supply and EtherCAT interfaces



Y_OA1	X_OA1
Y_OA2	X_OA2
Y_OB1	X_OB1
Y_OB2	X_OB2
Y_A+	X_A+
Y_A-	X_A-
Y_B+	X_B+
Y_B-	X_B-
Y_C+	X_C+
Y_C-	X_C-
Y_LL	X_LL
Y_RL	X_RL
Y_DI.COM0	X_DI.COM0

+VM0
PGND0
GDO 0
F.G.
GDO 1
PGND1
+VM1

U_OA1	Z_OA1
U_OA2	Z_OA2
U_OB1	Z_OB1
U_OB2	Z_OB2
U_A+	Z_A+
U_A-	Z_A-
U_B+	Z_B+
U_B-	Z_B-
U_C+	Z_C+
U_C-	Z_C-
U_LL	Z_LL
U_RL	Z_RL
U_DI.COM0	Z_DI.COM0

Figure 5: ECAT-2094S front view with motor and encoder in- and outputs

Name	Signal	Signal Description
+VM0	+5 to 46V_{DC} (from positive power contact)	Power supply for motor X and Y
PGND0	Ground 0V (from negative power contact)	
GDO 0	Output	General purpose digital output channel 0
F.G.		Frame ground
GDO 1	Output	General purpose digital output channel 1
PGND1	+5 to 46V_{DC} (from positive power contact)	Power supply for motor Z and U
+VM1	Ground 0V (from negative power contact)	

Table 5: Motor power supply and general purposes DO interfaces

Name	Signal	Signal Description	
X_OA1	Output	Motor X winding A1	Motor X
X_OA2	Output	Motor X winding A2	
X_OB1	Output	Motor X winding B1	
X_OB2	Output	Motor X winding B2	
X_A+	Input	Encoder X input A+	Encoder X
X_A-	Input	Encoder X input A-	
X_B+	Input	Encoder X input B+	
X_B-	Input	Encoder X input B-	
X_C+	Input	Encoder X input C+	
X_C-	Input	Encoder X input C-	
X_LL	Input	Left limit switch for motor X	DI, limit switch or latch trigger for motor X
X_RL	Input	Right limit switch for motor X	
X_DI.COM0		Common DI X supply: +10 to +24V_{DC}	For X_LL and X_RL
Y_OA1	Output	Motor Y winding A1	Motor Y
Y_OA2	Output	Motor Y winding A2	
Y_OB1	Output	Motor Y winding B1	
Y_OB2	Output	Motor Y winding B2	
Y_A+	Input	Encoder Y input A+	Encoder Y
Y_A-	Input	Encoder Y input A-	
Y_B+	Input	Encoder Y input B+	
Y_B-	Input	Encoder Y input B-	
Y_C+	Input	Encoder Y input C+	
Y_C-	Input	Encoder Y input C-	
Y_LL	Input	Left limit switch for motor Y	DI, limit switch or latch trigger for motor Y
Y_RL	Input	Right limit switch for motor Y	
Y_DI.COM0		Common DI Y supply: +10 to +24V_{DC}	For Y_LL and Y_RL
Z_OA1	Output	Motor Z winding A1	Motor Z
Z_OA2	Output	Motor Z winding A2	
Z_OB1	Output	Motor Z winding B1	
Z_OB2	Output	Motor Z winding B2	
Z_A+	Input	Encoder Z input A+	Encoder Z
Z_A-	Input	Encoder Z input A-	
Z_B+	Input	Encoder Z input B+	

Z_B-	Input	Encoder Z input B-	
Z_C+	Input	Encoder Z input C+	
Z_C-	Input	Encoder Z input C-	
Z_LL	Input	Left limit switch for motor Z	DI, limit switch or latch trigger for motor Z
Z_RL	Input	Right limit switch for motor Z	
Z_DI.COM0		Common DI Z supply: +10 to +24V_{DC}	For Z_LL and Z_RL
U_OA1	Output	Motor U winding A1	Motor U
U_OA2	Output	Motor U winding A2	
U_OB1	Output	Motor U winding B1	
U_OB2	Output	Motor U winding B2	
U_A+	Input	Encoder U input A+	Encoder U
U_A-	Input	Encoder U input A-	
U_B+	Input	Encoder U input B+	
U_B-	Input	Encoder U input B-	
U_C+	Input	Encoder U input C+	
U_C-	Input	Encoder U input C-	
U_LL	Input	Left limit switch for motor U	DI, limit switch or latch trigger for motor U
U_RL	Input	Right limit switch for motor U	
U_DI.COM0		Common DI U supply: +10 to +24V_{DC}	For U_LL and U_RL

Table 6: Connection interfaces for the motor current outputs, encoder and digital inputs

3.3 Digital Input and Output Wiring

Digital Input		
Digital input channels		8 (2x limit position for each motor)
Input type		Wet
Wet contact	ON voltage level	+10 to 30 V_{DC}
	OFF voltage level	+5 V_{DC} MAX
Photo-isolation		3750 V_{DC}
Digital Output		
Digital output channels		2
Output type		Open collector
Load voltage		+5 to 30 V_{DC}
Max. load current		100mA
Isolation voltage		3750 V_{DC}

Table 7: Digital input and output specifications

The diagram for right (RL) and left (LL) limit switch wiring for axis X, Y, Z, U is shown below (Figure 6).

The digital input RL and LL can be used as a simple DI, a positive and negative limit switch and a position latch trigger. The DI channels can be set to simultaneously act as a limit switch and a position latch input. Each axis (X, Y, Z, U) is equipped with a pair of RL and LL digital inputs.
Example: The RL, LL, DI.COM symbols in the picture below represents the X_RL, X_LL and X_DI.COM for the X-motor.

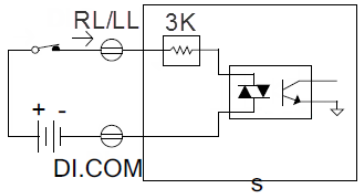
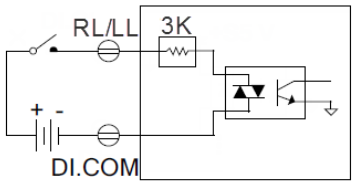
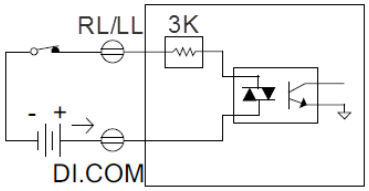
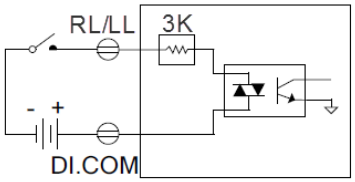
Digital Input	Readback as 1	Readback as 0
Sink	+10 ~ +24V DC	OPEN or <4 VDC
		
Source	+10 ~ +24V DC	OPEN or <4 VDC
		

Figure 6: Digital inputs RL and LL for motor X, Y, Z, U

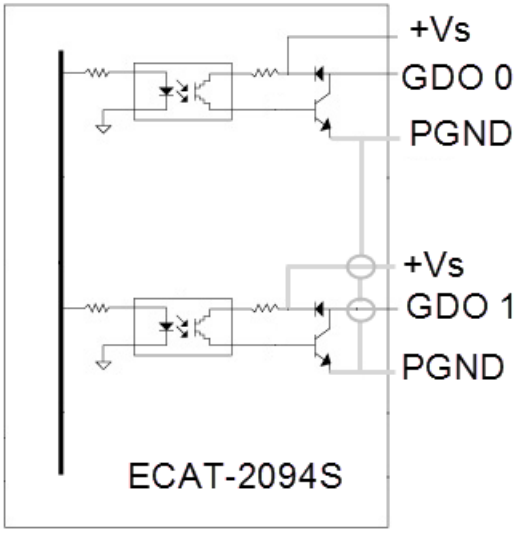


Figure 7: General purpose DO block diagram

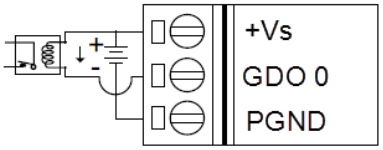
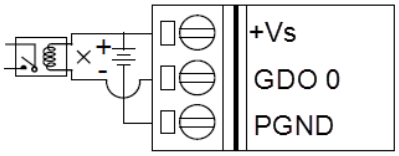
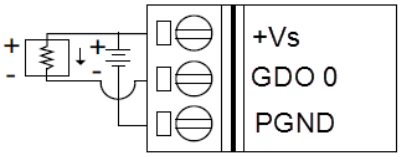
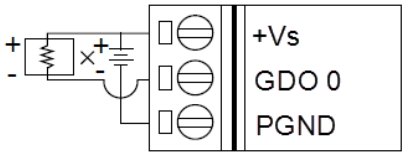
Output Type	ON State Readback as 1	OFF State Readback as 0
Driver Relay		
Resistance Load		

Figure 8: General purpose DO channel 0

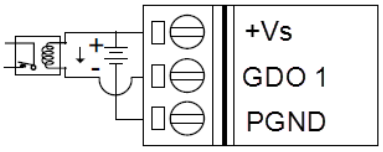
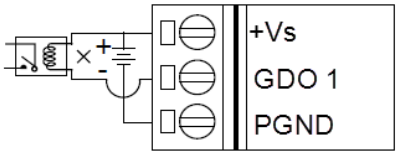
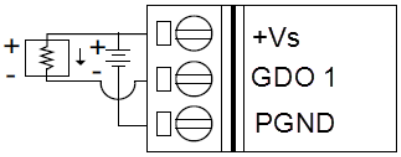
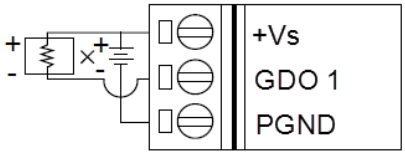
Output Type	ON State Readback as 1	OFF State Readback as 0
Driver Relay		
Resistance Load		

Figure 9: General purpose DO channel 1

3.4 Stepper Motor Wiring

3.4.1 Four Lead Motor

The Figure 10 below shows an example for a four lead two-phase motor connected to the X output of the ECAT-2094S.

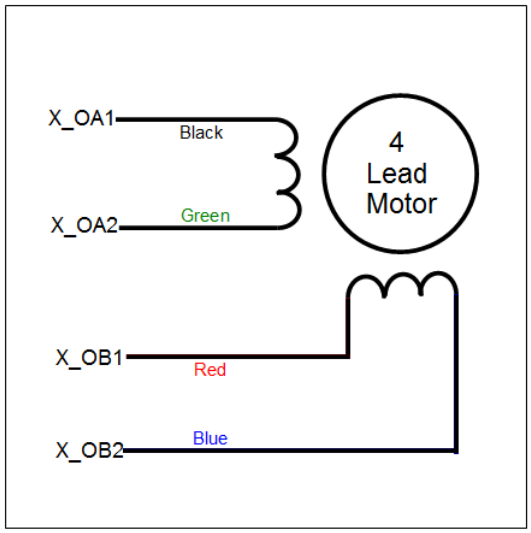


Figure 10: Four lead bipolar motor connected to the first axis output

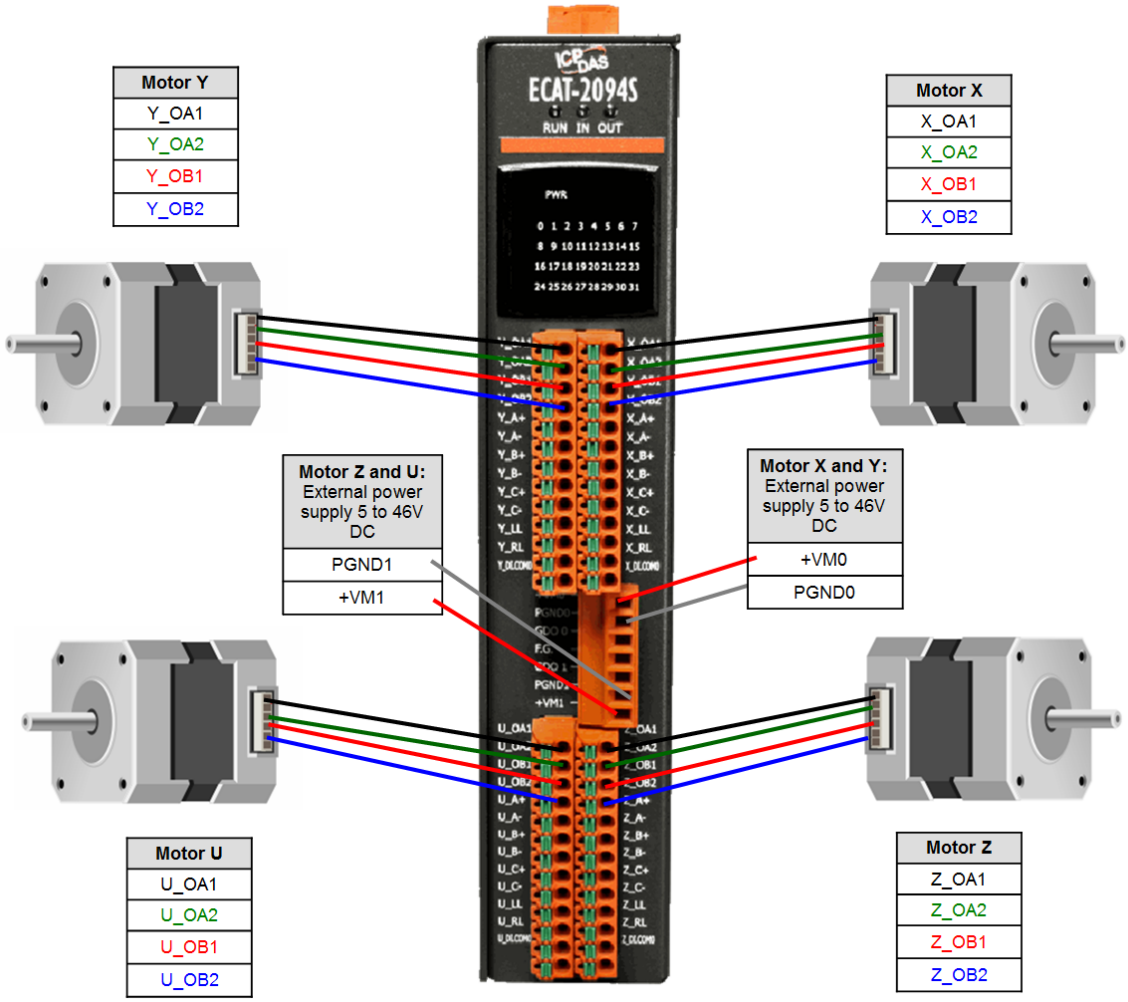


Figure 11: ECAT-2094S connected to four stepper motors

3.4.2 Eight Lead Motor

Eight lead motors can be connected in series or parallel. A series connected motor needs less current than one that is connected in parallel but it will not be able to run as fast.

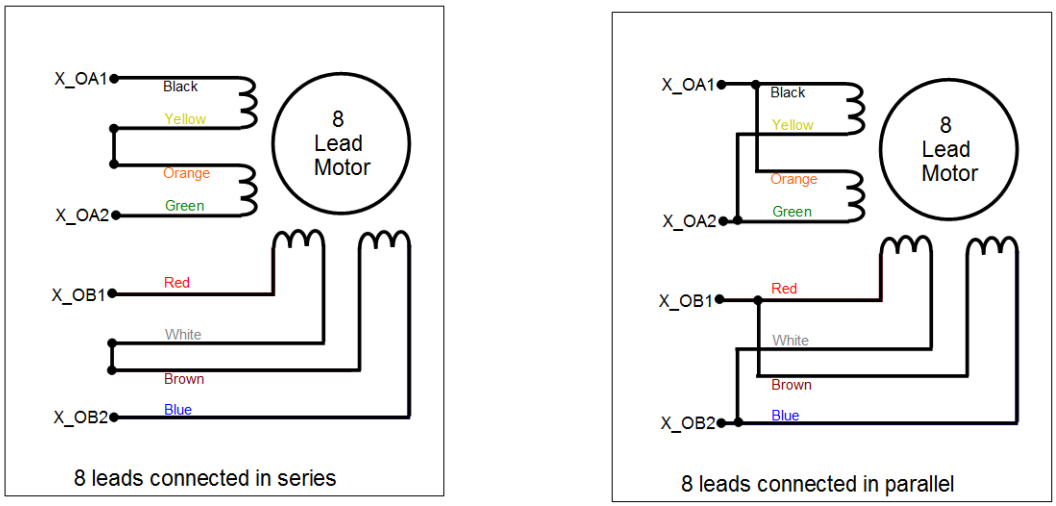


Figure 12: Eight lead bipolar motor connection (left: series, right: parallel)

3.4.3 Encoder Connection

Differential encoder:
The ECAT-2094S supports differential encoder by default.

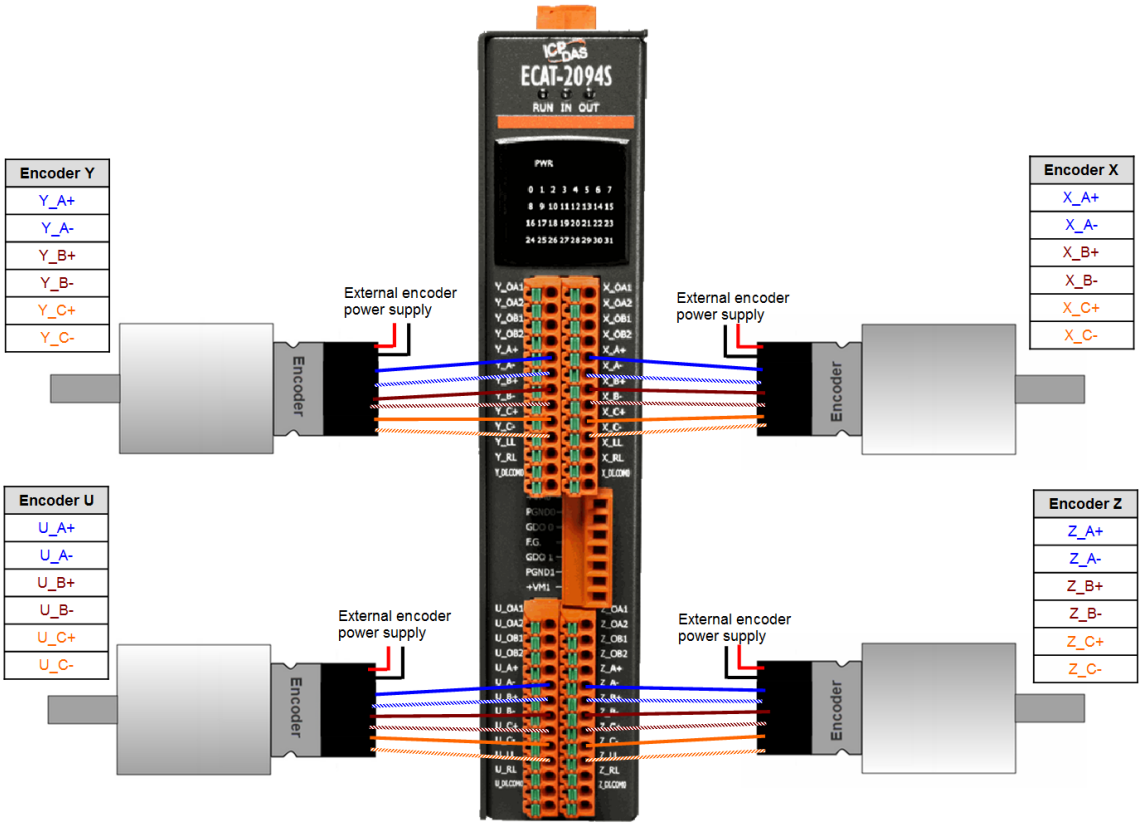


Figure 13: Connecting four differential encoder

Open collector type encoder:

For single-ended encoder connection refer to the Figure 14 which list the possible power supply values with the corresponding resistor sizes.

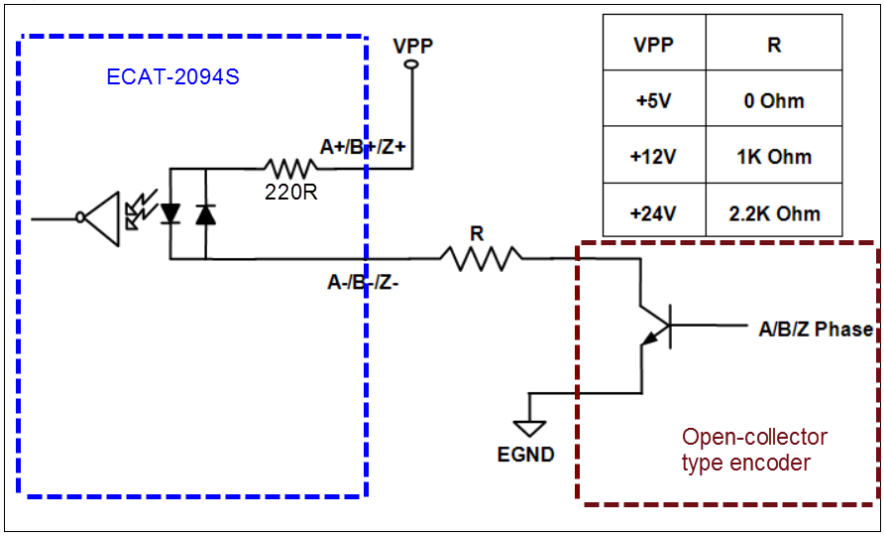


Figure 14: Open collector wiring diagram

4 Basics Communication

4.1 EtherCAT Cabling

The cable length between two EtherCAT devices must not exceed 100 m.

Cables and connectors
For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (CAT5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.
The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

4.2 EtherCAT State Machine

The state of the EtherCAT master and slave is controlled via the EtherCAT State Machine (ESM) . The state determines which functions are accessible or executable in the EtherCAT slave. State changes are typically initiated by requests of the master and acknowledged by the slave after the successful initialization. In case of an internal error, the slave automatically changes to a lower state.

The ECAT-2094S supports four states:
- Init (state after Reset)
- Pre-Operational
- Safe-Operational
- Operational

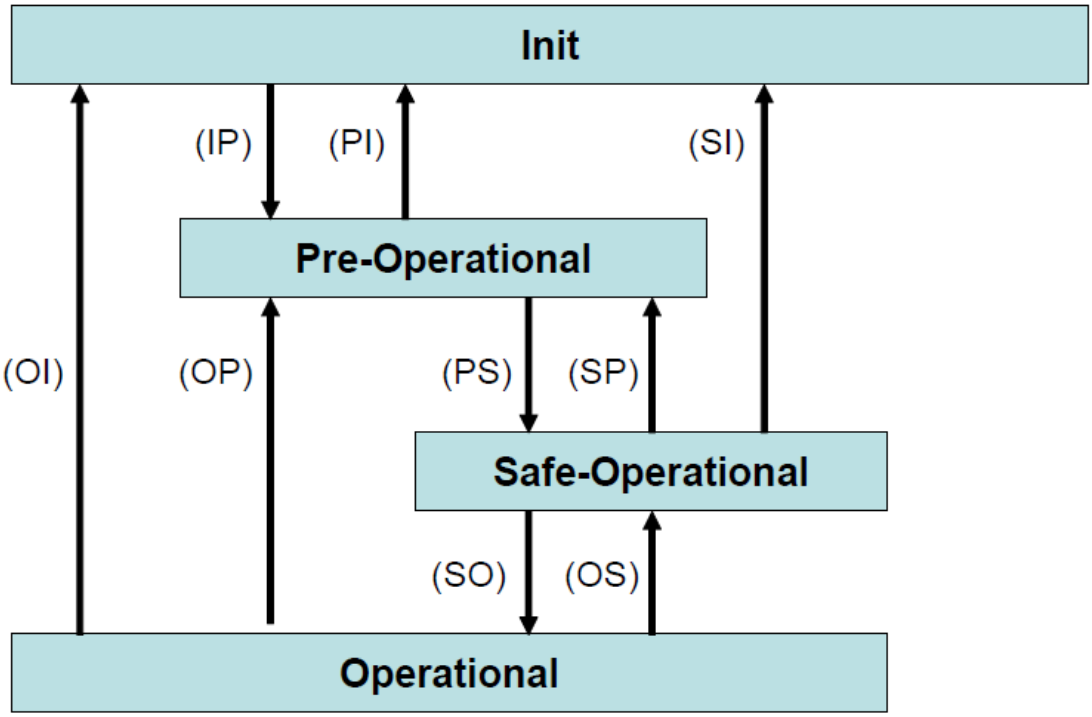


Figure 15: EtherCAT state machine

Init

After switch-on the EtherCAT slave is in the initial state. Only ESC register communication is possible, but no mailbox or process data communication. The slave initializes the service object data with default value or with values previously stored to the local memory. The EtherCAT master assigns the station address and configures the sync manager channels 0 and 1 for acyclic mailbox communication.

Pre-Operational (Pre-Op)

In Pre-Op state acyclic mailbox communication is possible, but not process data communication. In this state the EtherCAT master does the following configurations:

- Set the sync manager 2 and 3 of the ECAT-2094s for process data communication (from sync manager channel 2)
- The FMMU channels
- PDO mapping or the sync manager PDO assignment
- The user has the option to save motion control related configuration data (0x8000 -0x8321) to a non-volatile memory.

Safe-Operational (Safe-Op)

In Safe-Op state both mailbox and process data communication are enabled, but the slave keeps its outputs in a safe state, while the input data are updated cyclically. The slave will ignore the output data sent by the master and just return the current input

data (e.g. digital input, encoder value, etc.)

Outputs in Safe-Op state
The sync manager watchdog expires when the master application does not provide new output process data within the configured watchdog time. In this case the slave will automatically go from operational state to ERROR-SAFEOP state and set all the outputs in a safe state. The ECAT-2094S will stop the stepper motor, regulate the motor current to the configured safe level and switch the digital output to safe output values. All safe output value can be configured.

Operational (Op)
Here both the process data object (PDO) and service data object (SDP) are fully enabled. Master sends cyclic output data and read input data. The ECAT-2094S supports two type of Op modes: Free Run mode and Distributed Clock (DC) mode.

4.3 Synchronization Modes

ECAT-2094s devices support two different modes

- Free Run: The master cycle time and slave cycle time are independent and not synchronized.
- Distributed Clock (DC): The master cycle time and slave cycle time are synchronized.

4.3.1 Free Run Mode

The slave operates autonomously based on its own cycle and is not synchronized with the EtherCAT cycle. The master cycle time and the slave cycle time are fully independent which means each slave device reads/writes its own process data according to its local time, independent of the master's cycle time.

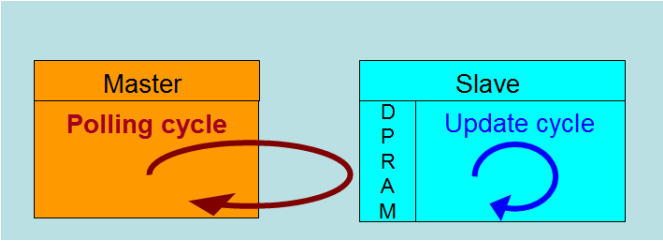


Figure 16: Master-slave cycle in Free Run mode

The following diagram shows the process timing of the slave in Free Run mode in detail:

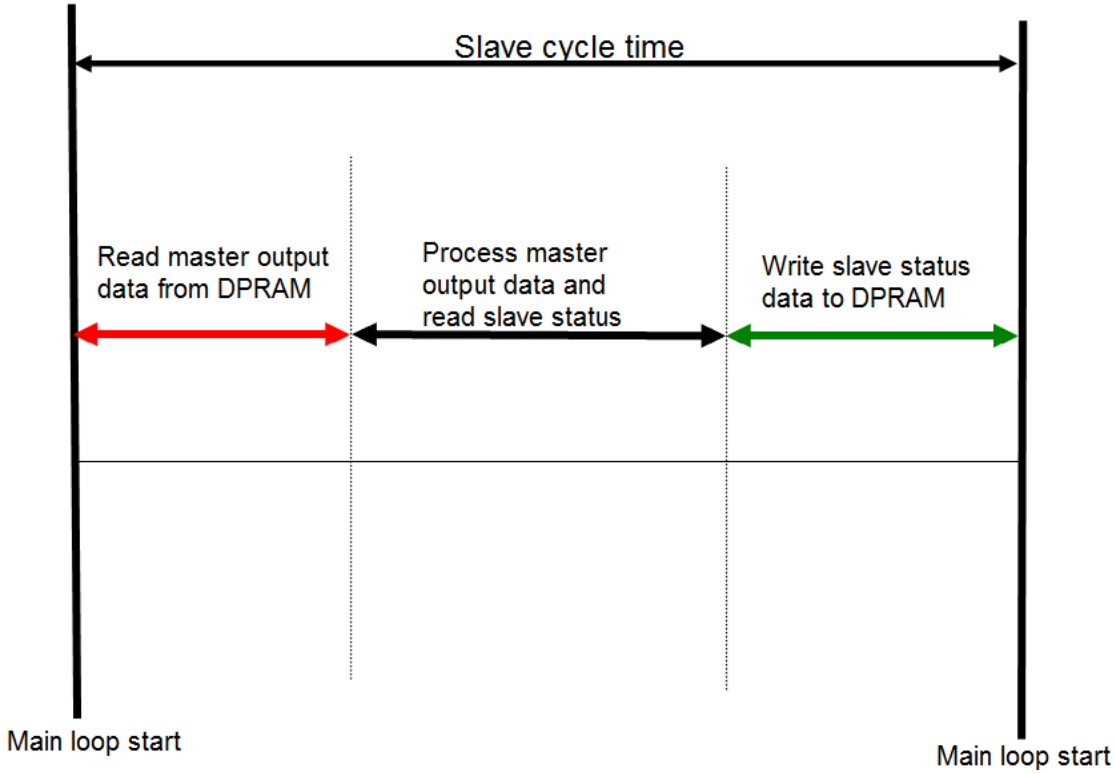


Figure 17: Slave processing sequence in Free-run mode

The slave firmware checks in each cycle time the memory of the EtherCAT slave chip (ESC) whether new output data has been received from the master. Newly received data will be processed, the motion path will be calculated and sent to the motion chip and digital output will be set. In the next step motion and digital input status are being read from motion chip. In the final step the read status are being written to the DPRAM, so that the master can retrieve the data ESC DPRAM in the next cycle time.

4.3.2 Distributed Clocks (DC Mode)

DC clock synchronization enables all EtherCAT devices (master and slaves) to share the same EtherCAT system time. The EtherCAT slaves in the network can be synchronized to each other. This enables the master to simultaneously set the output (e.g. digital output, pulse output) or to synchronously read inputs (e.g. digital input, encoder counter) of different slaves in the EtherCAT network.

For system synchronization all slaves are synchronized to one reference clock. Normally the first EtherCAT slave closest to the master with Distributed Clocks capability becomes the clock base for the master as well as for other DC slaves.

The EtherCAT slave is synchronized with the SYNC0 or SYNC1 event of the distributed

clock system. After the EtherCAT network has been set into DC communication mode by the master, the ESC (EtherCAT slave chip) of each slave generates fixed time hardware interrupt which triggers the slave firmware to process the PDO data received by the master. The master cycle time and the ESC hardware interrupt time interval are fully synchronized to the first slave in the network that is used as a reference clock with the SYNC0 signal.

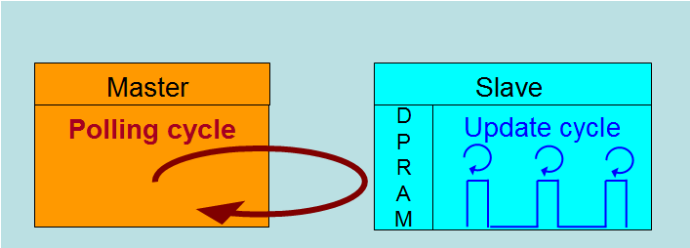


Figure 18: Master-slave cycle in DC mode

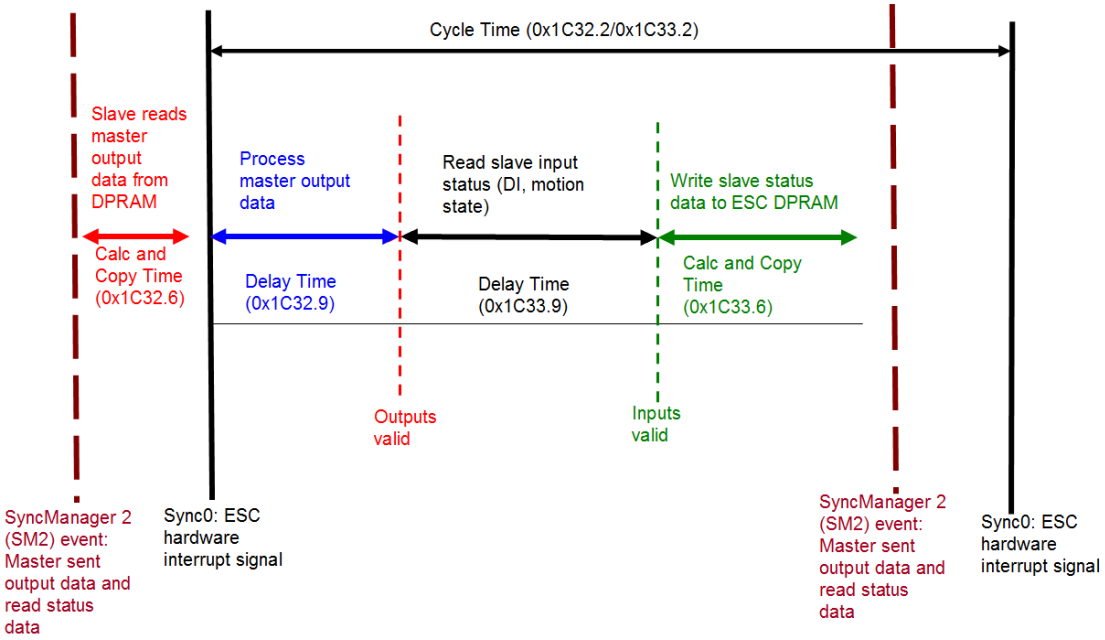


Figure 19: Internal slave processing sequence in DC mode

Once the slave receives process data (RxPDOs) from the master the SM2 event is triggered which causes the firmware to read the data from the ESC memory. The ESC interrupts the firmware at fixed time interval to process the data received from the master and write the status data to the ESC memory. Every time when the master fails to sent process data within the DC cycle time the internal sync error counter is being increase by three counts. This error counter is being decreased by one count for every successful DC cycle. Once the error counter reached the maximum count (default 4) a sync error will be generated and the slave goes into Safe OP mode (Sync Error 0x1C32:20 TRUE). The maximum count value can be set by changing the default value of the "Sync

Error Counter Limit" (0x10F1:02).

Index	Name	Flags	Value
10F1:0	Error Settings		> 2 <
10F1:01	Local Error Reaction	RW	0x00000001 (1)
10F1:02	Sync Error Counter Limit	RW	0x0004 (4)

Figure 20: Sync error counter limit object

The setting of the sync manager for the output and input data is available at the TwinCAT "CoE online" tab.

Index	Name	Flags	Value
1C32:0	SM output parameter		> 32 <
1C32:01	Synchronization Type	RW	0x0002 (2)
1C32:02	Cycle Time	RO	0x00000000 (0)
1C32:04	Synchronization Types supported	RO	0x401F (16415)
1C32:05	Minimum Cycle Time	RO	0x001E8480 (2000000)
1C32:06	Calc and Copy Time	RO	0x0007A120 (500000)
1C32:08	Get Cycle Time	RW	0x0001 (1)
1C32:09	Delay Time	RO	0x000927C0 (600000)
1C32:0A	Sync0 Cycle Time	RW	0x005B8D80 (6000000)
1C32:0B	SM-Event Missed	RO	0x0000 (0)
1C32:0C	Cycle Time Too Small	RO	0x0000 (0)
1C32:20	Sync Error	RO	FALSE

Figure 21: SyncManager 2 parameters

SyncManager parameter description (time unit: nanosecond):

- Calc and Copy Time (0x1C32.6 / 0x1C33.6): Required time to copy the process data from the ESC to the local memory and calculate the output value.
- Delay Time (0x1C32.9 / 0x1C33.9): Delay from receiving the trigger to set the output or latch the input.
- Cycle Time (0x1C32.2 / 0x1C33.2): The current cycle time for the application. When using DC synchronization the value is read from register 0x9A0:0x9A3.
- 0x1C32.5 / 0x1C33.5 (Min Cycle Time): Minimum cycle time for the application. It is the total execution time of all slave application related operations.

5 Project Integration

In this chapter the integration of the ECAT-2094S device into a TwinCAT controlled EtherCAT network is being described. In general the ECAT-2094S is a standard EtherCAT slave which can be controlled by any standard EtherCAT master (e.g. Acontis, CODESYS, etc.).

5.1 ESI File

A ESI file describes the properties and functions supported by the ECAT-2094S. By using the ESI file an easy and abstract integration of an EtherCAT device in a project tool is realized. With the help of the ESI file a detailed knowledge of EtherCAT is not required to configure the device. The TwinCAT EtherCAT master/System Manager needs the device description files in order to generate device configuration in online or offline mode.

5.1.1 Import of ESI File

Copy the XML description file "ECAT-2094S.xml" of the ECAT-2094S device into the TwinCAT system directory and restart the TwinCAT system.
For TwinCat 3.1 copy the ESI file "ECAT-2094S.xml" in the following directory:
C:\TwinCAT\3.1\Config\Io\EtherCAT

Software	Default directory path
Beckhoff EtherCAT Configuration	C:\EtherCAT Configurator\EtherCAT
Beckhoff TwinCAT 3.x	C:\TwinCAT\3.x\Config\Io\EtherCAT
Beckhof TwinCAT 2.x	C:\TwinCAT\Io\EtherCAT

Table 8: ESI file target directory

5.2 Device Setup and Configuration

In this manual only the online configuration of the slave module will be discussed. For offline configuration procedure please consult the TwinCAT user manual.

CAUTION:
Automatic start of stepper motor!

- Risk of death or serious injury for humans working in the machine.
- It can not ruled out that the stepper motor may perform unplanned movement during the ECAT-2094S setup and configuration
- Make sure that, even if the drive starts to move unintentionally, no danger can result for personnel or machinery. The measures you must take in this regard for your task are based on the risk assessment of the application.

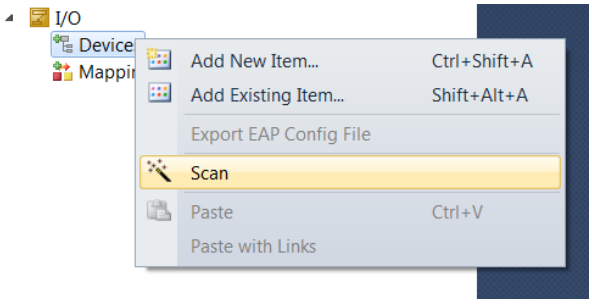
The following conditions must be met before a configuration can be set up:

- The ECAT-2094S slave devices must be connected via EtherCAT cables to the EtherCAT master. In this manual TwinCAT 3.1 version is being used as the EtherCAT master and configuration tool
- The ECAT-2094S devices has to be connected to the power supply and ready for communication
- Set the TwinCAT in CONFIG mode.

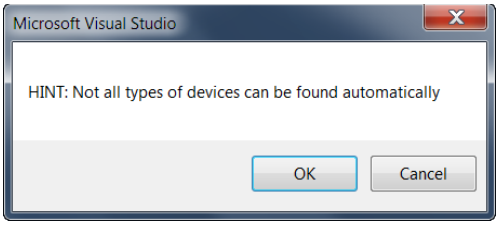
5.2.1 Scanning of the EtherCAT Device

After the TwinCAT has been set into CONFIG mode the online device search can be started.

Step 1: Right-click the “Devices” in the configuration tree to open the scan dialog. Click "Scan" to search the ECAT-2094S device.



Step 2: Select "OK"



Step 3: Select the Ethernet device (Ethernet chip) to which the ECAT-2094S is connected to. Confirm the selection with "OK".

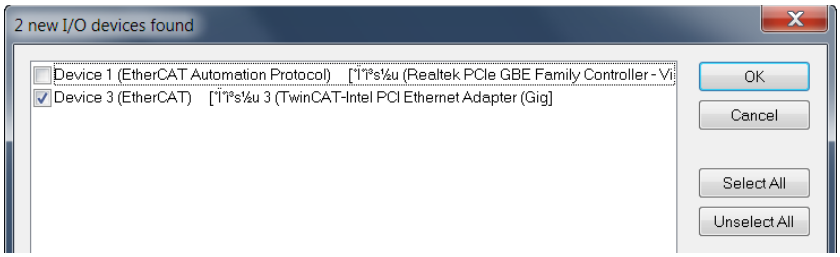
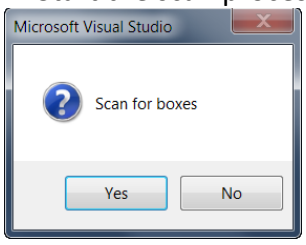
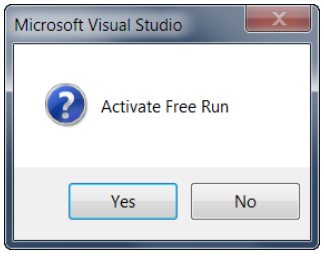


Figure 22:List of Ethernet chips detected on the EtherCAT master PC

Step 4: Start the scan process by clicking "Yes"



Step 5: Set the ECAT-2094S into Free-Run mode by clicking "Yes"



The ECAT-2094S is by default in the velocity mode. All the parameter used by the velocity control mode is being displayed in the tree view:

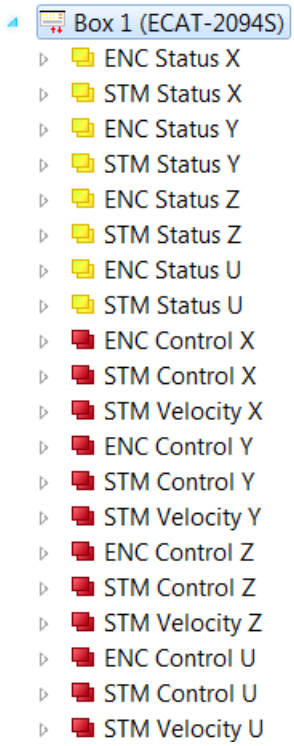


Figure 23: Default parameter selection for the velocity control mode

5.2.2 EtherCAT Slave Process Data Settings

The user has to select the process data which is being transferred between the EtherCAT master and slave during each cycle (Process Data Objects, PDOs). The process data exist of two parts:

- TxPDO: Data which is being read by the master (e.g. motion status)
- RxPDO: data or parameters which is being sent to the slave (e.g. target position of the stepper motor) .

The process data image is determined by the application program and will be updated cyclically.

The ECAT-2094S basically support four types of motion modes:

- Velocity control
- Position control
- Position interface compact
- Position interface

By selecting one of the motion mode from the list box (Figure 24) all the relevant parameters are automatically assigned and mapped to the process data objects (TxPDO, RxPDO). If required, additional objects can be assigned to the process data by selecting

the object listed under "PDO Assignment (0x1C12)" and "PDO Assignment (0x1C13)".

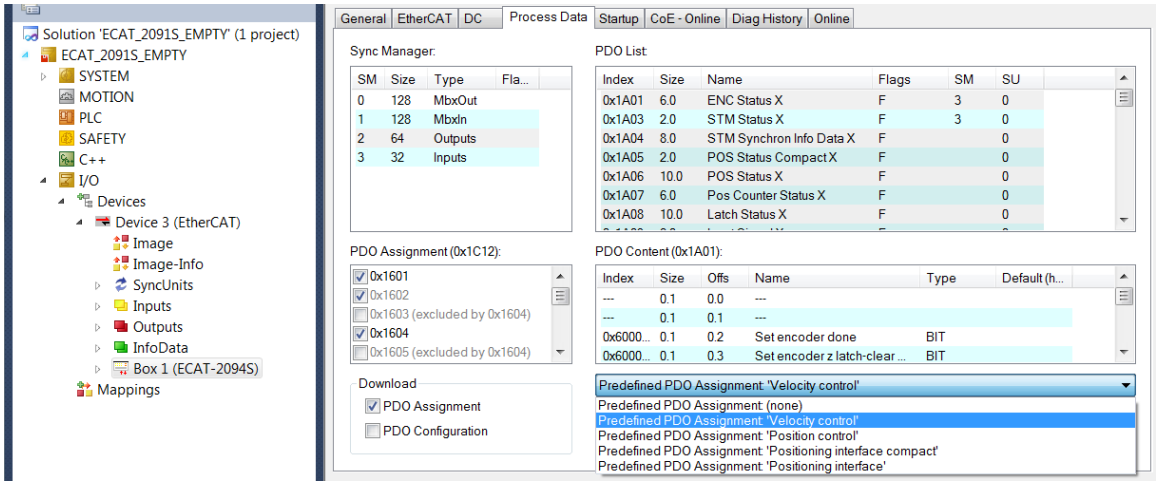


Figure 24: Predefined PDO assignment selection

Download the new PDO assignment to the Sync manager of the slave by clicking "Restart TwinCAT (Config Mode)" in the drop down menu.

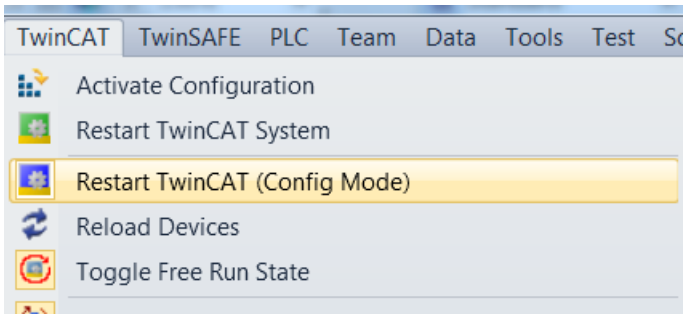


Figure 25: Download PDO assignment and restart TwinCAT

5.2.3 Basic Stepper Driver Configuration

Motion parameters which do only need to be configured once before the actual motion control starts are listed in the "CoE online" tab. These parameters have to be accessed via the CANopen over EtherCAT (CoE) protocol. The CoE protocol has a lower priority than the cyclic process data object (PDO) communication. Therefore CoE motion parameters will not be updated in every cycle but only when the master has spare time.

Motion relevant CoE parameter are

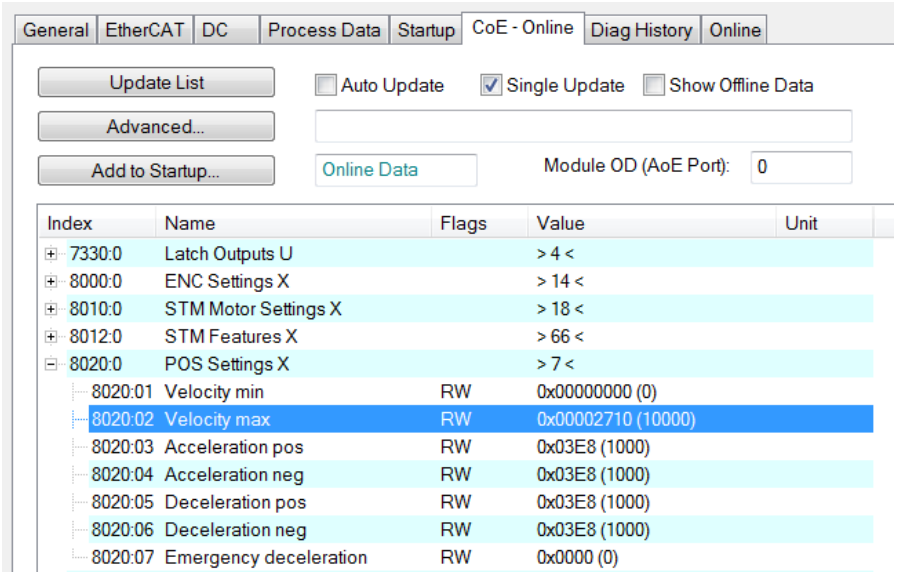
- Encoder setting (Index 8n00)
- Stepper motor setting (Index 8n10)
- Stepper motor features (Index 8n12)
- POS setting (Index 8n20)

- POS features (Index 8n21)

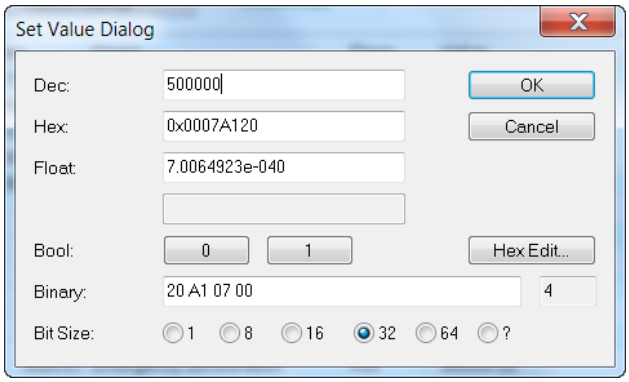
Whereby "n" represents the motor number (0 to 3)

Example of setting the maximum allowable speed for motor X:

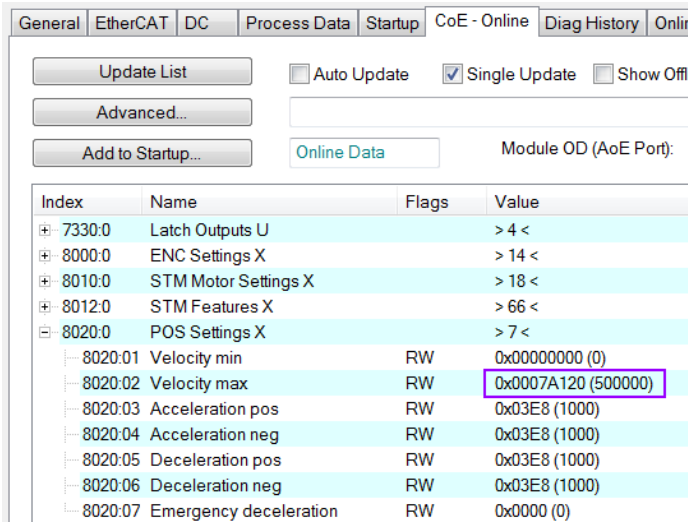
Step 1: Go to POS Setting X. Extend the index tree and double click "Velocity max" with the index 8020:02.



Step 2: Enter a new value for the maximum allowable velocity [steps/second] and click "OK"



Step 3: Once the value has been successfully sent to the slave it will be displayed in the CoE online parameter list:



This value needs only to be set once and therefore does not have to be sent in every cycle time.

All the relevant motion parameters have to be set first before the actual real time motion control starts. Once the parameters are set, the four motors are basically ready for operation.

6 Position Control Setting

The position interface allows the user to set a target position and the motion controller automatically drives the motor to the specified position. The basic motion configuration data such as the acceleration and deceleration values and the maximum motion velocity have to be set before motion control execution can be started.

6.1 Positioning Interface Types

Two predefined PDO assignment types for the position interface are provided:

- Positioning interface
- Positioning interface compact

The predefined PDO assignment enables a simplified selection of the process data.

The "Positioning interface" type activates all the position control PDOs required to execute point to point motion. The created process data image is quite large because it contains motion parameters needed to control the four motors. This large process data is being transmitted in every cycle and slows down the system. If communication speed and a small process data image are a criteria for the system setup then the "Positioning interface compact" type should be activated. Here most motion parameter values are not send at a fixed, deterministic cycle but set via CoE. In an application where the motion parameters (velocity, acceleration, deceleration etc.) only need to be set once in a while the "Positioning interface compact" type is the better option.

In the following the parameter settings for both positioning interface types will be discussed in details.

6.2 Positioning Interface

The sequence of executing and controlling a travel command in "Positioning interface" mode is shown in the following flow diagram (Figure 26). The diagram shows the sequence of parameter setting and status checking during the execution of a position command. The configuration parameter setting has to be done beforehand.

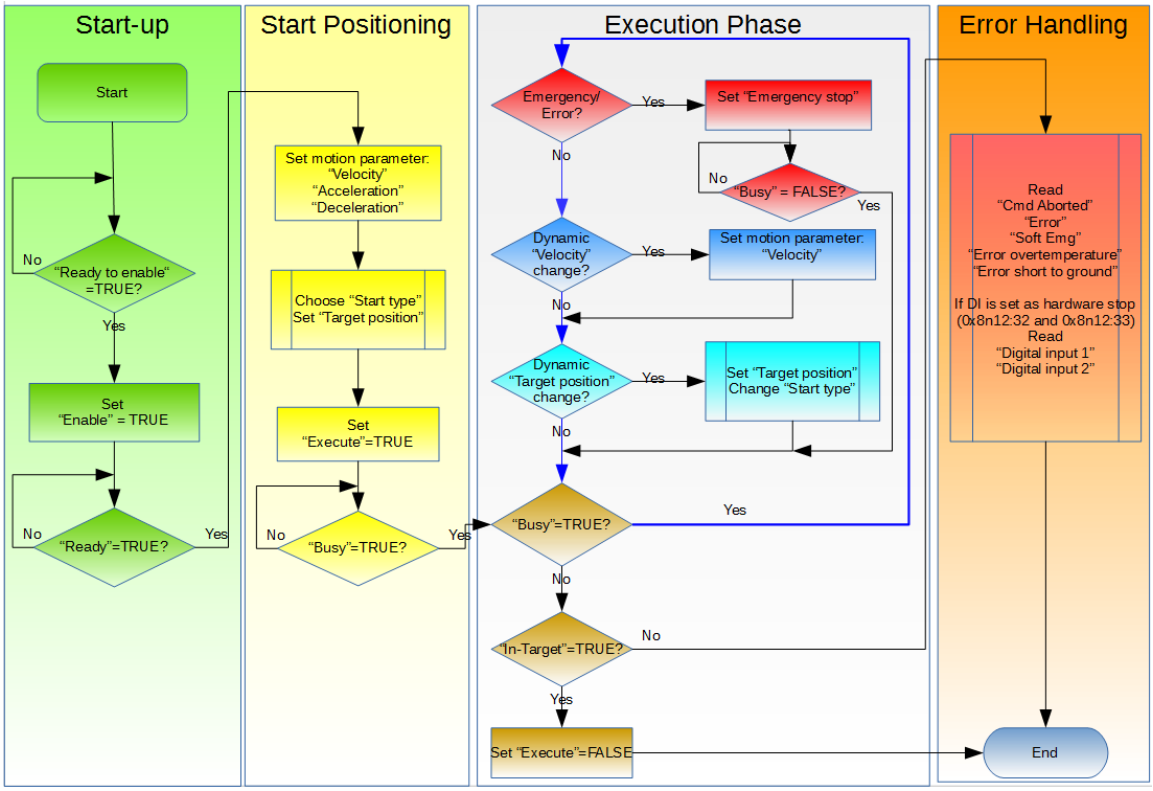


Figure 26: Flow diagram for position interface

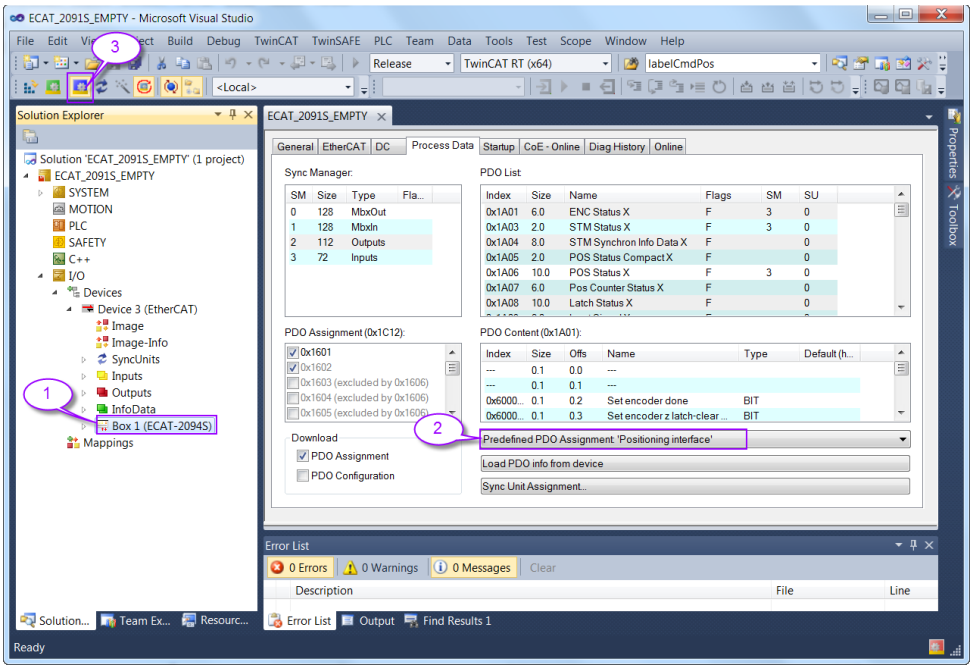
Execution procedure:

Step 1: PDO assignment

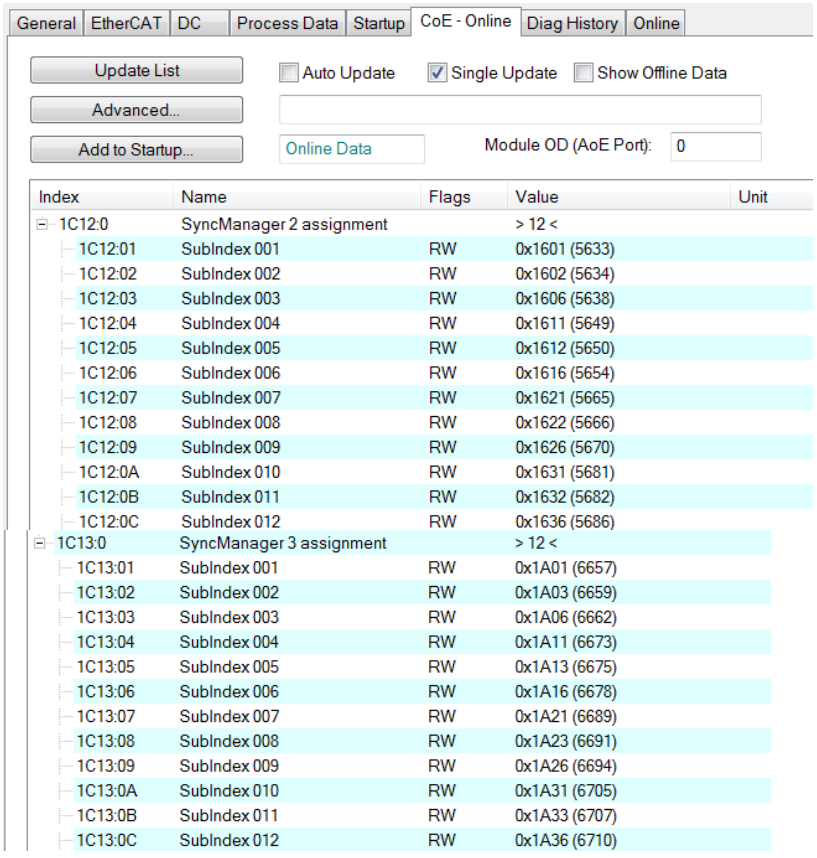
Select the function "Positioning interface" in the lower part of the "Process data" tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.

Procedure:

1. Select the "Process Data" tab of the ECAT-2094S
2. Select the "Predefined PDO Assignment: "Position interface" from the combo box
3. Send the PDO assignment to the slave by clicking the "Reload I/O device" button



The SyncManager 2 and 3 in the "CoE-Online" tab displays the new PDO assignment:



Index	Name	Flags	Value	Unit
1C12:0	SyncManager 2 assignment		> 12 <	
1C12:01	SubIndex 001	RW	0x1601 (5633)	
1C12:02	SubIndex 002	RW	0x1602 (5634)	
1C12:03	SubIndex 003	RW	0x1606 (5638)	
1C12:04	SubIndex 004	RW	0x1611 (5649)	
1C12:05	SubIndex 005	RW	0x1612 (5650)	
1C12:06	SubIndex 006	RW	0x1616 (5654)	
1C12:07	SubIndex 007	RW	0x1621 (5665)	
1C12:08	SubIndex 008	RW	0x1622 (5666)	
1C12:09	SubIndex 009	RW	0x1626 (5670)	
1C12:0A	SubIndex 010	RW	0x1631 (5681)	
1C12:0B	SubIndex 011	RW	0x1632 (5682)	
1C12:0C	SubIndex 012	RW	0x1636 (5686)	
1C13:0	SyncManager 3 assignment		> 12 <	
1C13:01	SubIndex 001	RW	0x1A01 (6657)	
1C13:02	SubIndex 002	RW	0x1A03 (6659)	
1C13:03	SubIndex 003	RW	0x1A06 (6662)	
1C13:04	SubIndex 004	RW	0x1A11 (6673)	
1C13:05	SubIndex 005	RW	0x1A13 (6675)	
1C13:06	SubIndex 006	RW	0x1A16 (6678)	
1C13:07	SubIndex 007	RW	0x1A21 (6689)	
1C13:08	SubIndex 008	RW	0x1A23 (6691)	
1C13:09	SubIndex 009	RW	0x1A26 (6694)	
1C13:0A	SubIndex 010	RW	0x1A31 (6705)	
1C13:0B	SubIndex 011	RW	0x1A33 (6707)	
1C13:0C	SubIndex 012	RW	0x1A36 (6710)	

Step 2: Set the motor torque:
Be careful when setting the torque. Motor current fine tuning is required to lower motor temperature and reduce the current to save power.

Torque produced by the stepper motors is directly proportional to the current, but the amount of heat generated is roughly proportional to the *square* of the current. If the motor is operated at 90% of rated current, 90% of the rated torque will be outputted. But the motor will produce approximately 81% as much heat compared to the maximum torque output. At 70% current, the torque is reduced to 70% and the heating to about 50%.

Attention:
If the motor current is set at or above 1.1A for increased periods of time the ECAT-2094S will heat up and emit increasing heat as the resistive power dissipation raises with the square of the motor current.

Four torque settings have to be done. The valid range for the motor current setting is 0 to 1500 mA. The unit for the motor current parameters is milliamperes [mA].

1. The "Maximal run current" sets the motor driving current. This torque setting will be applied once the motion execution flag (0x7n10:01 - Enable) has been activated
2. "Reduce run current" output is triggered once the "Reduce torque" Boolean has been set to true (0x7n10:03 - Reduced torque).
3. The "Maximal hold current" sets the motor standstill current. This torque setting will be applied once the motion execution flag (0x7n10:01 - Enable) has been activated
4. "Reduce hold current" output is triggered once the "Reduce torque" Boolean has been set to true (0x7n10:03 - Reduced torque).
5. The purpose of the "Power on motor current" (0x7n10:08) variable is to apply a torque to the driver directly after power on to prevent freewheeling.
6. In case the EtherCAT communication is interrupted or when the EtherCAT master sets the ECAT-2094s from OP mode into a non-OP mode while a motion command is being executed then the ECAT-2094s uses the "Safe motor current" (0x7n10:12) setting to prevent the motor from freewheeling.

Index	Name	Flags	Value	Unit
8010:0	STM Motor Settings X		> 18 <	
8010:01	Maximum run current	RW	0x02EE (750)	
8010:02	Reduced run current	RW	0x0177 (375)	
8010:03	Maximum hold current	RW	0x02EE (750)	
8010:04	Reduced hold current	RW	0x0177 (375)	
8010:06	Motor fullsteps	RW	0x00C8 (200)	
8010:07	Micro Steps	RW	256 (8)	
8010:08	Power on motor current	RW	0x0177 (375)	
8010:09	Max Start Velocity	RW	0x0064 (100)	
8010:12	Safe motor current	RW	0x0177 (375)	

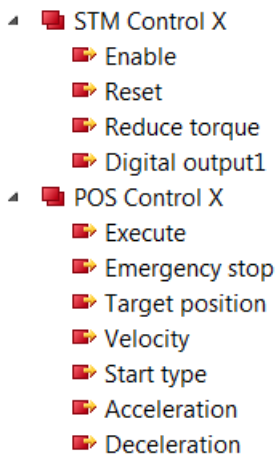
Step 3: Set the number of micro-steps per full step (8n10:07). The motor runs smoother and with less vibration with higher micro-steps value setting, but also requires a higher step pulse frequency to achieve maximum speed.

Step 4: Set the motion parameters for the system: max velocity, max acceleration, etc.

1. Set the start velocity (unit: steps/second) (0x8n10:09)
2. Set the velocity range of the system (unit: steps/second) (0x8n20:01 and 0x8n20:02). The maximal velocity "Velocity max" ensures that under no circumstances the motor velocity will exceed this maximal value. The minimal velocity "Velocity min" defines the lowest velocity of the system and is being applied when changing the velocity during driving.
3. "Emergency deceleration" describes the deceleration time in milliseconds after the emergency stop flag has been raised (0x7n20:02 - Emergency stop)

Index	Name	Flags	Value
8010:0	STM Motor Settings X		> 18 <
8010:01	Maximum run current	RW	0x02EE (750)
8010:02	Reduced run current	RW	0x0177 (375)
8010:03	Maximum hold current	RW	0x02EE (750)
8010:04	Reduced hold current	RW	0x0177 (375)
8010:06	Motor fullsteps	RW	0x00C8 (200)
8010:07	Micro Steps	RW	256 (8)
8010:08	Power on motor current	RW	0x0177 (375)
8010:09	Max Start Velocity	RW	0x0064 (100)
8010:12	Safe motor current	RW	0x0177 (375)
8012:0	STM Features X		> 66 <
8020:0	POS Settings X		> 7 <
8020:01	Velocity min	RW	0x00000000 (0)
8020:02	Velocity max	RW	0x00002710 (10000)
8020:03	Acceleration pos	RW	0x03E8 (1000)
8020:04	Acceleration neg	RW	0x03E8 (1000)
8020:05	Deceleration pos	RW	0x03E8 (1000)
8020:06	Deceleration neg	RW	0x03E8 (1000)
8020:07	Emergency deceleration	RW	0x0000 (0)

Step 5: Motion execution procedure:



1. Activate the Enable (0x7n10:01)
2. Set the motion parameters: acceleration and deceleration time (milliseconds), the target velocity (steps/seconds) and target position (steps).
   i. Set the target velocity (unit: steps/second)
   ii. The acceleration time (unit: milliseconds) is defined as the time to accelerate the motor from "Velocity min" (0x8n20:01) to "Velocity max" (0x8n20:02) and the deceleration time is defined as the time needed to decelerate from "Velocity max" (0x8n20:02) to "Velocity min" (0x8n20:01). Attention: the acceleration time is not defined as the time needed to accelerate the motor from the current velocity to the target velocity.
   iii. The "Start type" (0x7n20:22) describes whether the target position is a relative or absolute position. In addition it is possible to set with the start type parameter whether a running motion command can be overwritten.

Name	Command	Description
ABSOLUTE	0x0001	The motor travels from the current position to the target position. The distance to travel depends on the distance difference between the current and target position
RELATIVE	0x0002	A specified position difference is added to the current position
ADDITIVE	0x0006	A specified position difference is added to the last target position **Note:** The RELATIVE and ADDITIVE type are similar when the last command was completed successfully. In this case both types will travel the same position because both start positions are the same. If an error occurred during the execution of the previous command (e.g. motor stall, emergency stop) then the current position is arbitrary. Now

		the RELATIVE type will use the current arbitrary position as the start position but the ADDITIVE type will use the last target position as the start position. By selecting the ADDITIVE type the user has the advantage that he can use the last target position for determining the next target position. Therefore no home search needs to be done in case of an error.
ABSOLUTE_CHANGE	0x1001	Change of the target position on the fly: Dynamic change of the target position during a travel command to a new absolute position
RELATIVE_CHANGE	0x1002	Dynamic change of the target position during a travel command to a new relative position (the current changing position value is used here also) Attention: Due to propagation delays it is not possible to determine exactly the actual position of the running motor. Reading the current position takes time and during this time the motor has already move to a new position. Therefore, there will be a difference between the desired target position and the actual target position.
ADDITIVE_CHANGE	0x1006	Dynamic change of the target position during a travel command to a new additive position (the last target position is used here)

Table 9: Start type definition

iv. Set the target position (unit: steps). The target position can be a relative distance or a absolute position. The behavior of this parameter is being determined by the "Start type" setting.

3. Start motion execution by setting the "Execute"-variable to true (0x7n20:01).

4. If an emergency stop (0x7n20:02) has been activated during driving, then the "Emergency stop"-variable has to be set to false and "Execute" back to false before the next command can be executed

5. Error: If an error occurred during driving (overheating, EtherCAT communication failed, Master sets slave from OP to none OP mode, etc.) the error flag is activated (0x6n10:04 Error). In order to clear this flag the "Reset" variable has to be activated (0x7n10:02 - Reset) for one cycle time.

Example:

Dynamic change of the target position

Time	POS Control X Outputs	POS Status X Inputs	Descriptions
t1	Execute = 1	Busy = 1	• Set the motion parameters

	Target position = 250000 Velocity = 10000 Start type = 0x0001 Acceleration = 1000 Deceleration = 1000	Accelerate = 1 Deceleration = 0 In-Target = 0	• Start executing the travel command ▪ Acceleration phase
t2		Busy = 1 Accelerate = 0 Deceleration = 0 In-Target = 0	• Target velocity has been reached
t3	Target position = 220000 Velocity = 8000 Start type = 0x1001 Acceleration = 500 Deceleration = 500	Busy = 1 Accelerate = 0 Deceleration = 1 In-Target = 0	• Change target position, velocity and acc/dec on the fly
t4		Busy = 1 Accelerate = 0 Deceleration = 0 In-Target = 0	• New target velocity has been reached
t5		Busy = 1 Accelerate = 0 Deceleration = 1 In-Target = 0	• Start the deceleration phase to the target position
T6	Execute = 0	Busy = 0 Accelerate = 0 Deceleration = 0 In-Target = 1	• Target position has been reached • Set Execute to false

Table 10: Change the target position on the fly

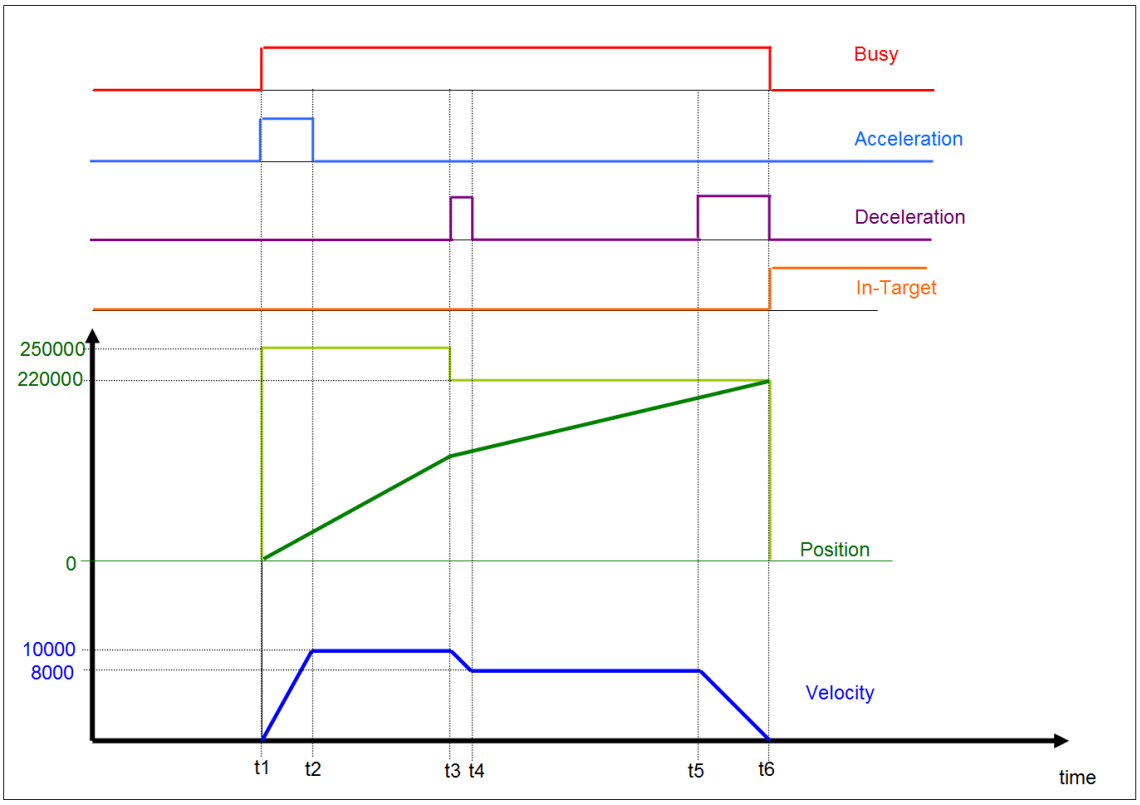


Figure 27: Output and input parameters on a time graph for changing the target position on the fly

6.3 Positioning Interface Compact

In the following the procedure for executing a travel command in "Positioning interface compact" mode is being described.

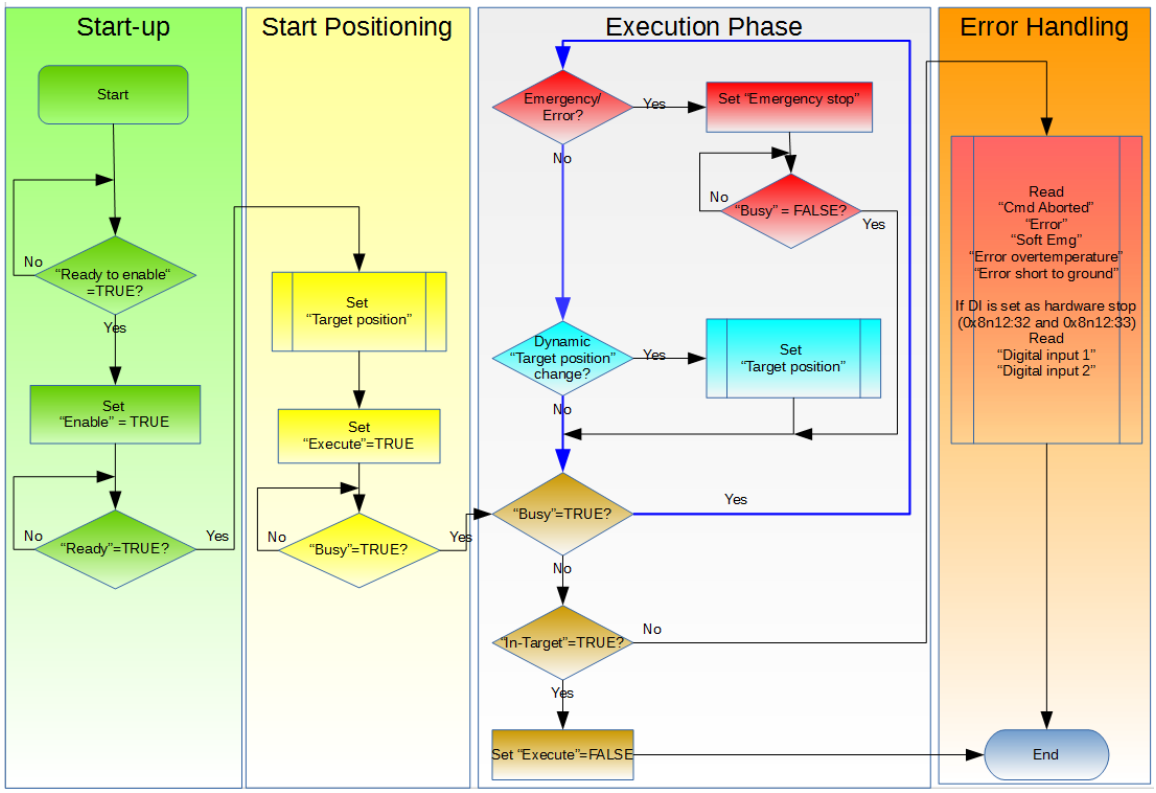


Figure 28: "Positioning interface compact" setting sequence

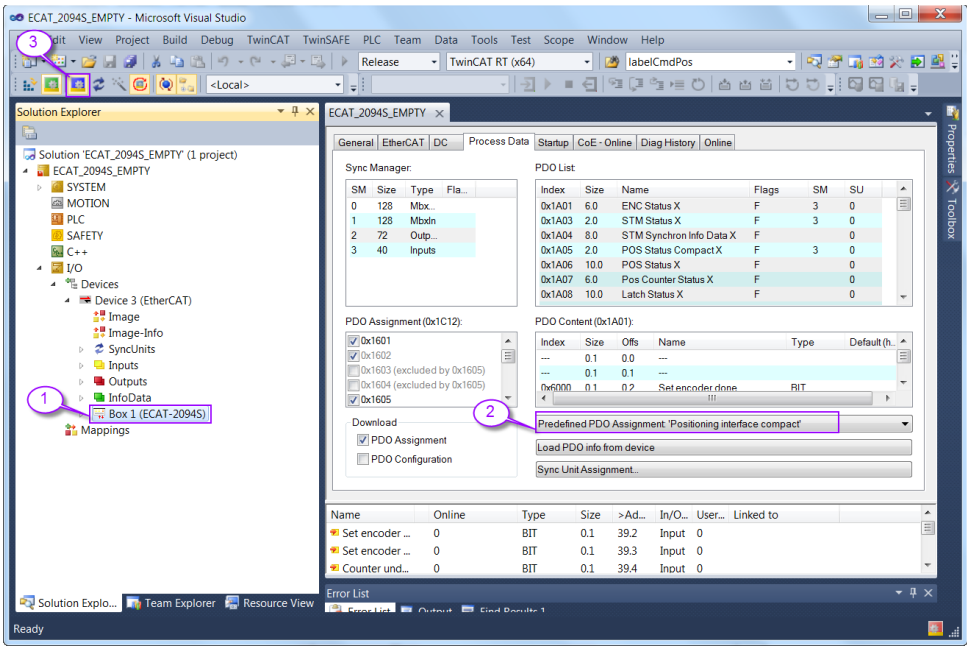
Operating procedure:

Step 1: PDO assignment

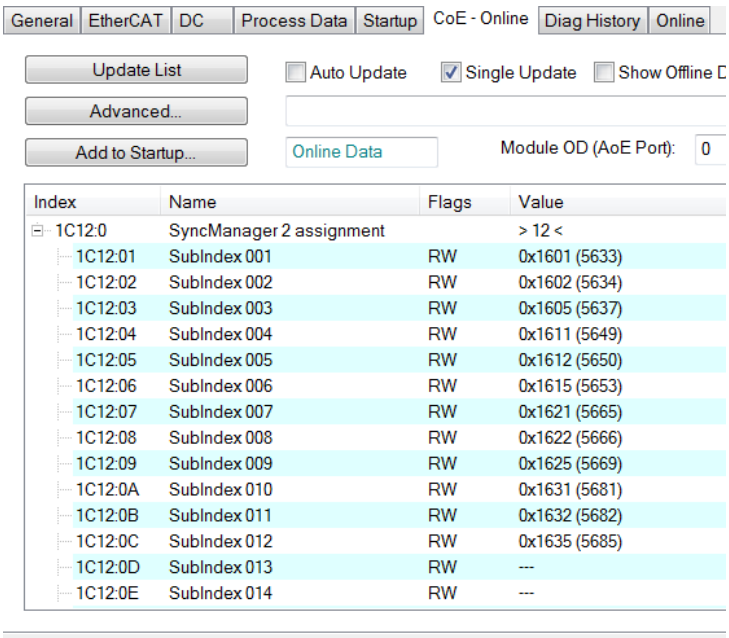
Select the function "Positioning interface compact" from the predefined PDO assignment selection box in the lower part of the "Process data" tab. This causes TwinCAT to automatically activated all necessary PDOs and deactivate the unnecessary ones.

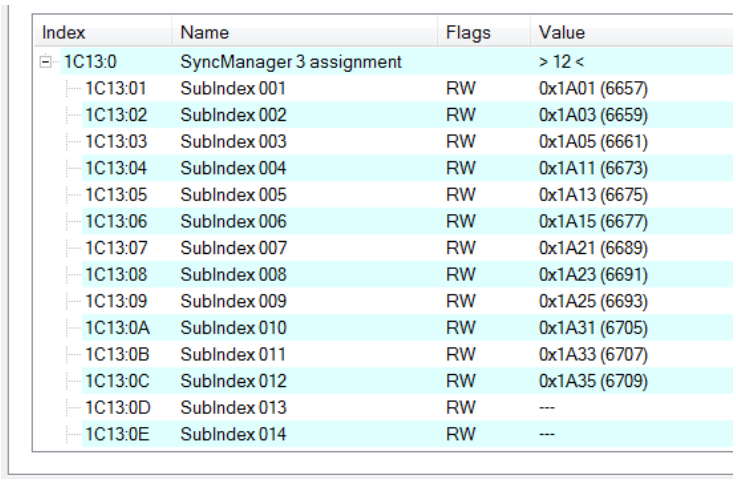
Procedure:

1. Select the "Process Data" tab of the ECAT-2094S
2. Select the "Predefined PDO Assignment: "Position interface compact" from the combo box
3. Send the PDO assignment to the slave by clicking the "Reload I/O device" button



The SyncManager 2 and 3 in the "CoE-Online" tab displays the new PDO assignment:





Index	Name	Flags	Value
1C13:0	SyncManager 3 assignment		> 12 <
1C13:01	SubIndex 001	RW	0x1A01 (6657)
1C13:02	SubIndex 002	RW	0x1A03 (6659)
1C13:03	SubIndex 003	RW	0x1A05 (6661)
1C13:04	SubIndex 004	RW	0x1A11 (6673)
1C13:05	SubIndex 005	RW	0x1A13 (6675)
1C13:06	SubIndex 006	RW	0x1A15 (6677)
1C13:07	SubIndex 007	RW	0x1A21 (6689)
1C13:08	SubIndex 008	RW	0x1A23 (6691)
1C13:09	SubIndex 009	RW	0x1A25 (6693)
1C13:0A	SubIndex 010	RW	0x1A31 (6705)
1C13:0B	SubIndex 011	RW	0x1A33 (6707)
1C13:0C	SubIndex 012	RW	0x1A35 (6709)
1C13:0D	SubIndex 013	RW	---
1C13:0E	SubIndex 014	RW	---

Step 2: Set the motor torque (see "Positioning interface", chapter 6.2 Step 2:)

Step 3: Set the number of micro-steps per full step (8n10:07). The motor runs smoother and with less vibration with higher micro-steps value setting.

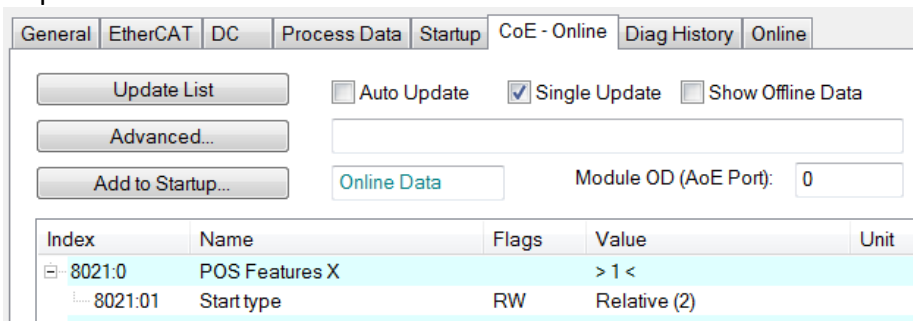
Step 4: Set the motion parameters for the system: max velocity, max acceleration, etc.

1. The velocity has to be set according the description of "Positioning interface" (chapter 6.2 Step 4:)
2. In addition the acceleration and deceleration time (unit: milliseconds) have to set. The time for both the positive and negative direction are required.
   i. Acceleration pos (0x8n20:03): Acceleration time in the positive direction of rotation.
   ii. Acceleration neg (0x8n20:04): Acceleration time in the negative direction of rotation
   iii. Deceleration pos (0x8n20:05): Deceleration time in the positive direction of rotation
   iv. Deceleration neg (0x8n20:06): Deceleration time in the negative direction of rotation

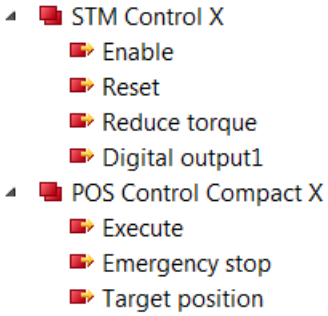
   The acceleration time is defined as the time needed to accelerate the motor from "Velocity min" (0x8n20:01) to "Velocity max" (0x8n20:02) and the deceleration time is defined as the time required to decelerate the motor from "Velocity max" (0x8n20:02) to "Velocity min" (0x8n20:01).
3. "Emergency deceleration" describes the deceleration time in milliseconds needed to stop the motor after the emergency stop flag has been set to TRUE (0x7n20:02 - Emergency stop)

Index	Name	Flags	Value
8010:0	STM Motor Settings X		> 18 <
8010:01	Maximum run current	RW	0x02EE (750)
8010:02	Reduced run current	RW	0x0177 (375)
8010:03	Maximum hold current	RW	0x02EE (750)
8010:04	Reduced hold current	RW	0x0177 (375)
8010:06	Motor fullsteps	RW	0x00C8 (200)
8010:07	Micro Steps	RW	256 (8)
8010:08	Power on motor current	RW	0x0177 (375)
8010:09	Max Start Velocity	RW	0x0064 (100)
8010:12	Safe motor current	RW	0x0177 (375)
8012:0	STM Features X		> 66 <
8020:0	POS Settings X		> 7 <
8020:01	Velocity min	RW	0x00000000 (0)
8020:02	Velocity max	RW	0x00002710 (10000)
8020:03	Acceleration pos	RW	0x03E8 (1000)
8020:04	Acceleration neg	RW	0x03E8 (1000)
8020:05	Deceleration pos	RW	0x03E8 (1000)
8020:06	Deceleration neg	RW	0x03E8 (1000)
8020:07	Emergency deceleration	RW	0x0000 (0)

Step 5: Set the start type. The "Start type" (0x8n21:01) describes whether the target position is a relative or absolute position. In addition the user can determine whether the target position can be changed on the fly. Consult Table 9 for the correct parameter value.



Step 6: Motion execution procedure:



1. Activate the "Enable" (0x7n10:01) flag
2. Set the target position (unit: steps) (0x7n20:11). The target position distance is being defined by the "Start type" (0x8n21:01) configuration.
3. Start motion execution by setting the "Execute"-variable to true

(0x7n20:01).

4. If an emergency stop (0x7n20:02) has been activated during driving, then the "Emergency stop"-variable has to be set to false and "Execute" back to false before the next command can be executed
5. Error: If an error occurred during driving (overheating, EtherCAT communication failed, Master sets slave from OP to none OP mode, etc.) the error flag is activated (0x6n10:04 Error). In order to clear this flag the "Reset" variable has to be activated (0x7n10:02 - Reset) for one cycle time.

6.4 Position Control

Position control mode has to be selected if the application program needs to sent a new absolute target position in every communication cycle. The maximum velocity and the acceleration time have to be set at a high value in order for the driver to reach the new target position at the end of the cycle time. In this mode the application program basically calculates and control the velocity profile of the motor.

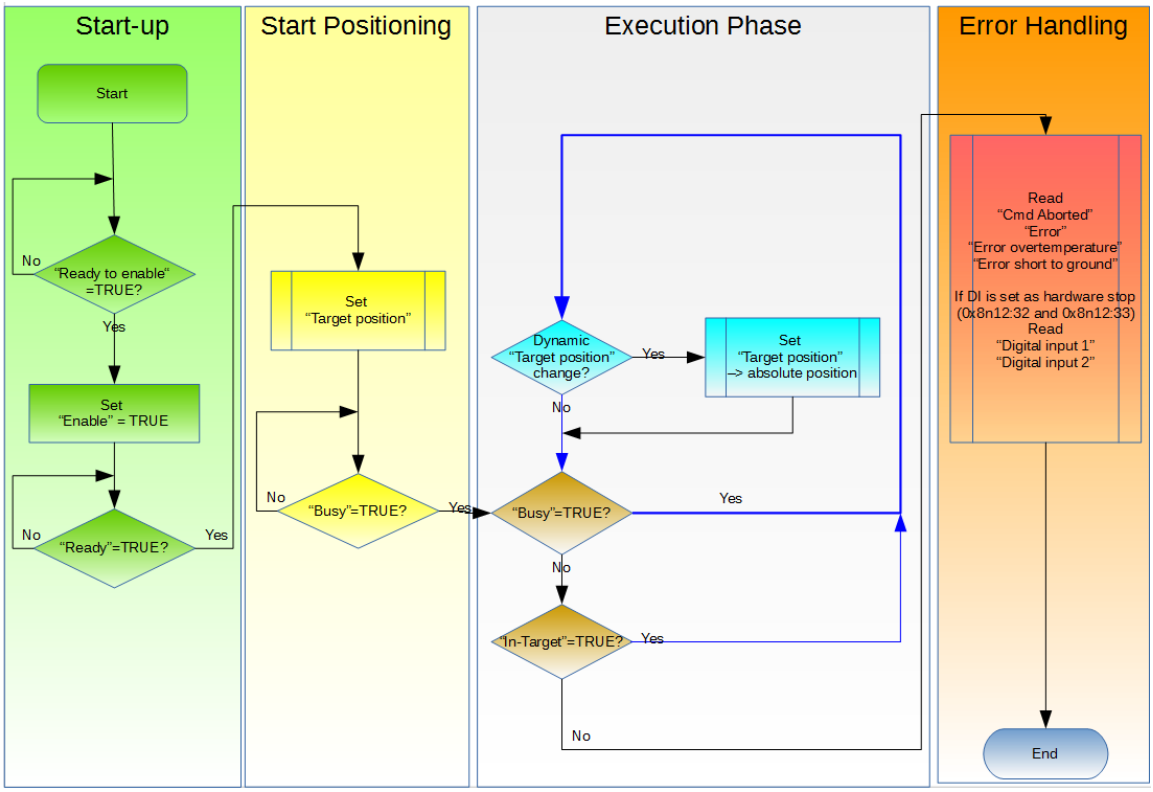


Figure 29: Variable execution sequence for the position control mode

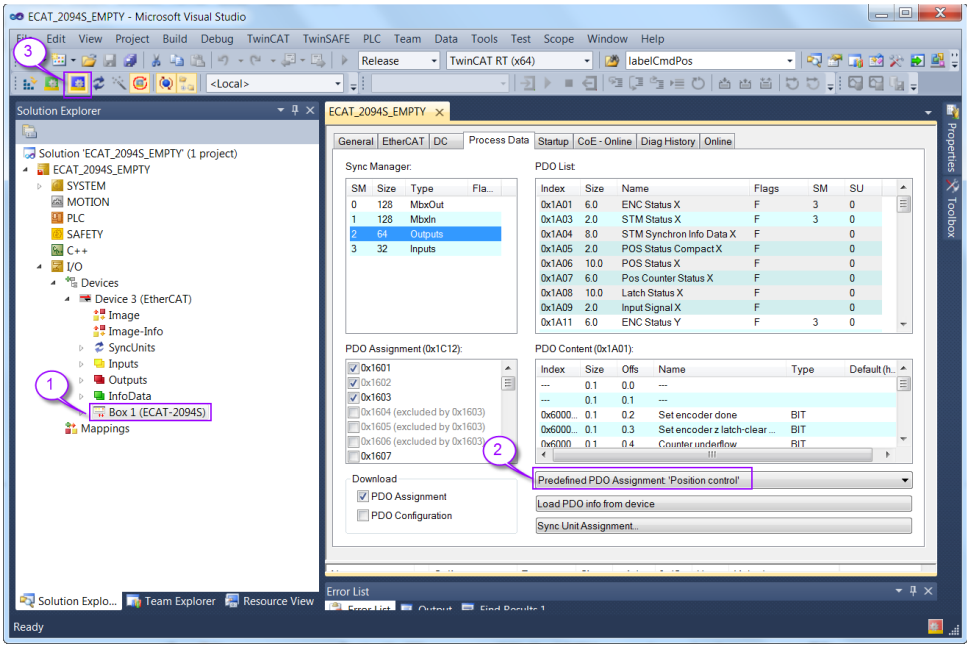
Operating procedure:

Step 1: PDO assignment

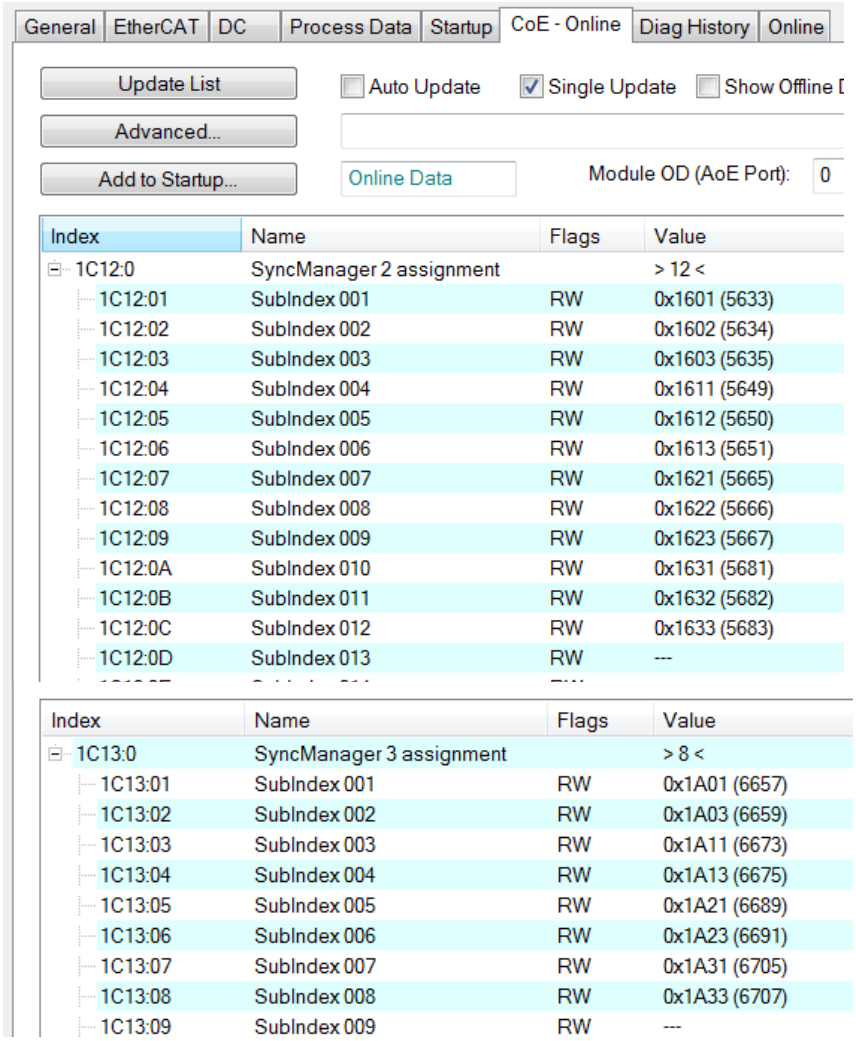
Select the function "Positioning control" predefined PDO assignment in the lower part of the Process data tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.

Procedure:

1. Select the "Process Data" tab of the ECAT-2094S
2. Select the "Predefined PDO Assignment: "Position control" from the combo box
3. Send the PDO assignment to the slave by clicking the "Reload I/O device" button



The SyncManager 2 and 3 in the "CoE-Online" tab displays the new PDO assignment:



Step 2: Set the motor torque (see "Positioning interface", chapter 6.2 Step 2:)

Step 3: Set the number of micro-steps per full step (8n10:07). The motor runs smoother and with less vibration with higher micro-steps value setting.

Step 4: Set the motion parameters for the system: max velocity, max acceleration, etc.

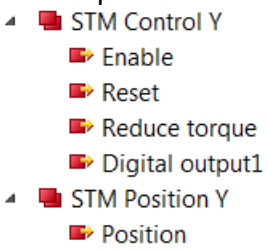
1. The velocity has to be set as described for the "Positioning interface" (chapter 6.2 Step 4:)
2. In addition the acceleration and deceleration time (unit: milliseconds) have to be set. The time for both the positive and negative directions are required.
   i. Acceleration pos (0x8n20:03)
   ii. Acceleration neg (0x8n20:04)
   iii. Deceleration pos (0x8n20:05)
   iv. Deceleration neg (0x8n20:06)

   The acceleration time is defined as the time needed to accelerate the motor from "Velocity min" (0x8n20:01) to "Velocity max" (0x8n20:02) and the deceleration time is defined as the time required to decelerate the

motor from "Velocity max" (0x8n20:02) to "Velocity min" (0x8n20:01).

Index	Name	Flags	Value
8010:0	STM Motor Settings X		> 18 <
8010:01	Maximum run current	RW	0x02EE (750)
8010:02	Reduced run current	RW	0x0177 (375)
8010:03	Maximum hold current	RW	0x02EE (750)
8010:04	Reduced hold current	RW	0x0177 (375)
8010:06	Motor fullsteps	RW	0x00C8 (200)
8010:07	Micro Steps	RW	256 (8)
8010:08	Power on motor current	RW	0x0177 (375)
8010:09	Max Start Velocity	RW	0x0064 (100)
8010:12	Safe motor current	RW	0x0177 (375)
8012:0	STM Features X		> 66 <
8020:0	POS Settings X		> 7 <
8020:01	Velocity min	RW	0x00000000 (0)
8020:02	Velocity max	RW	0x00002710 (10000)
8020:03	Acceleration pos	RW	0x03E8 (1000)
8020:04	Acceleration neg	RW	0x03E8 (1000)
8020:05	Deceleration pos	RW	0x03E8 (1000)
8020:06	Deceleration neg	RW	0x03E8 (1000)
8020:07	Emergency deceleration	RW	0x0000 (0)

Step 5: Motion execution procedure:



1. Activate the "Enable" (0x7n10:01) parameter
2. Set the absolute target position (unit: steps). The driver will output steps as soon as the actual and target position are not identical. Set the acceleration time and velocity (Step 4:) to a high value if the application if the motor needs to reach the target position at the end of each cycle.
3. Error: If an error occurred during driving (overheating, EtherCAT communication failed, Master sets slave from OP to none OP mode, etc.) the error flag is activated (0x6n10:04 Error). In order to clear this flag the "Reset" variable has to be activated (0x7n10:02 - Reset) for one cycle time.

7 Velocity Control Setting

In velocity control mode the motor accelerates to the target velocity and keeps running at this velocity until the user changes the velocity. When the user changes the velocity setting the controller will automatically accelerate/decelerate to the new value. In case of a rotation direction change the driver first slows the motor down to standstill before accelerating in the opposite direction. The motor will stop if the speed is set to zero. The acceleration and deceleration values have to be set via the configuration objects (0x8n20).

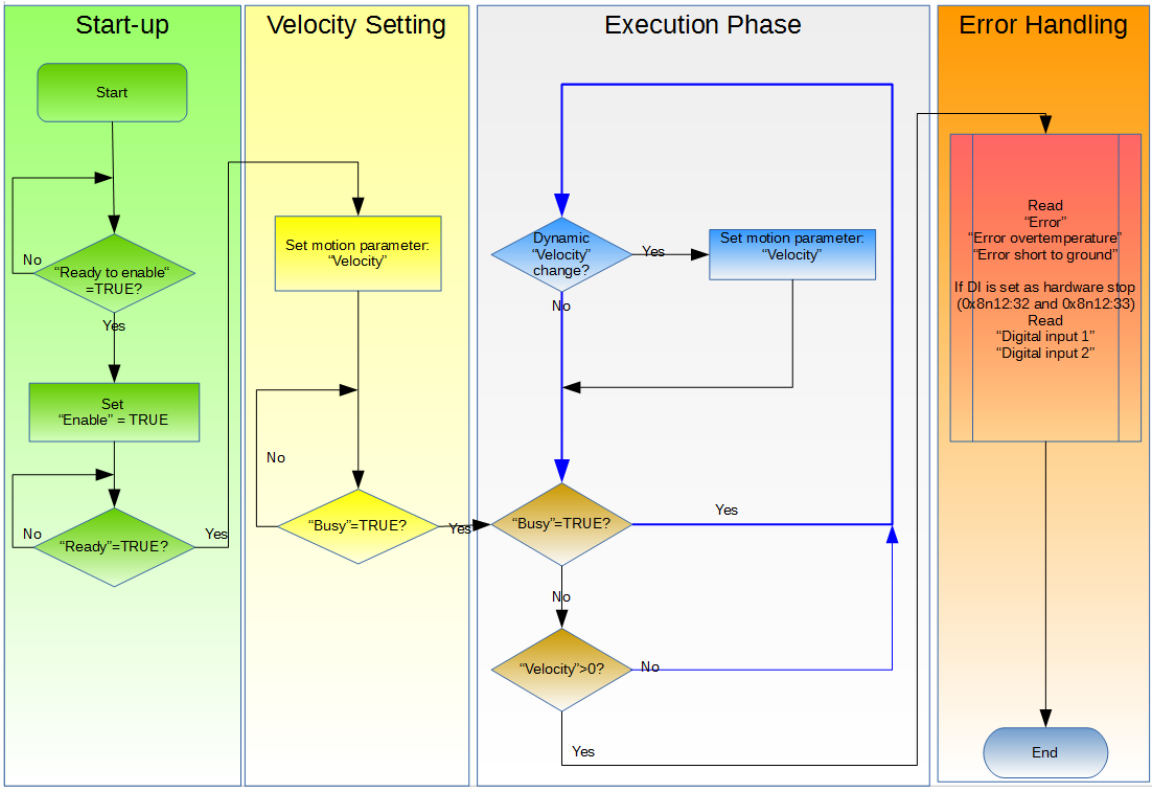


Figure 30: Velocity control settings

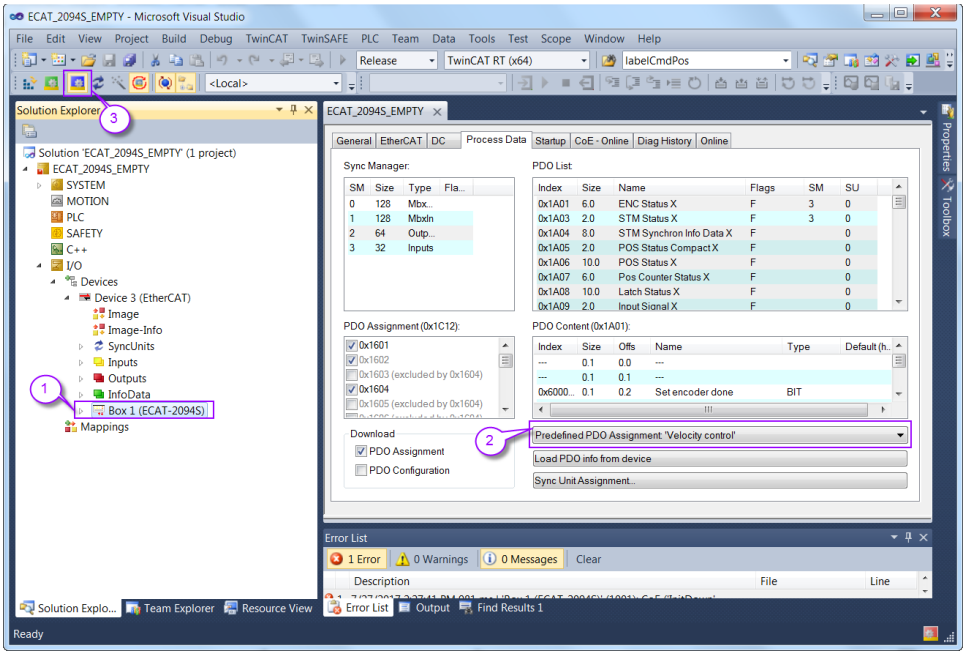
Procedure for the velocity control operation:

Step 1: PDO assignment

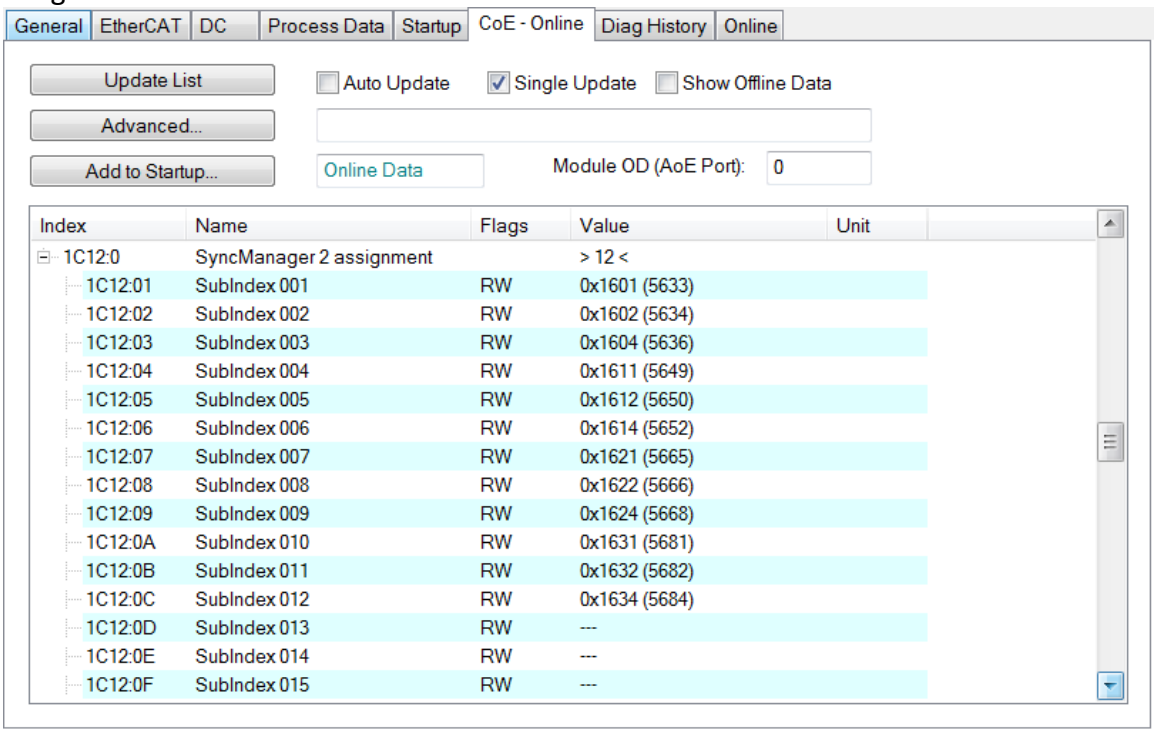
Select the function "Velocity control" predefined PDO assignment selection list in the lower part of the "Process data" tab:

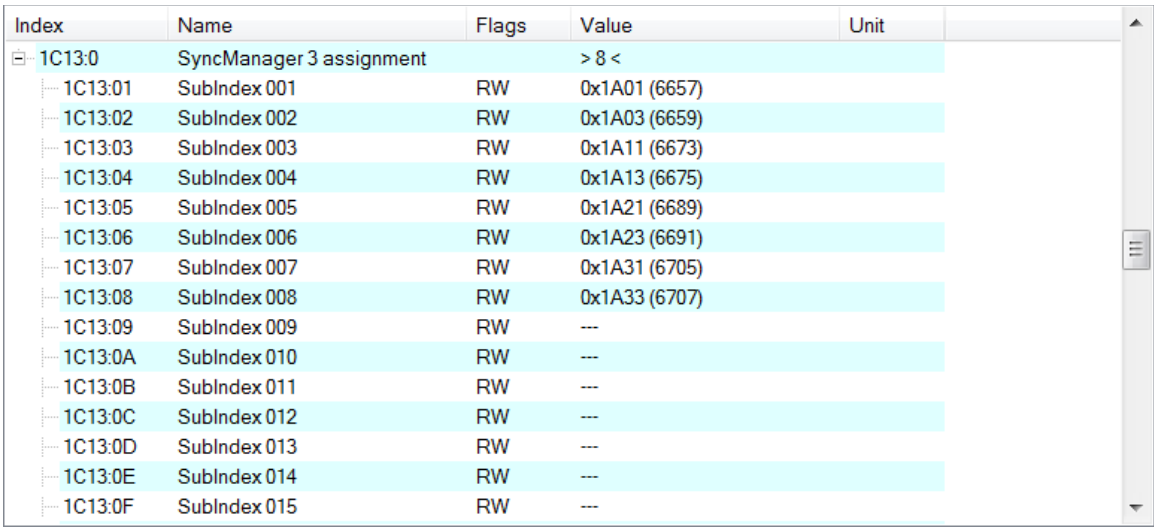
1. Select the "Process Data" tab of the ECAT-2094S
2. Select the "Predefined PDO Assignment: " Velocity control " from the combo box
3. Send the PDO assignment to the slave by clicking the "Reload I/O device"

button



The SyncManager 2 and 3 in the "CoE-Online" tab displays the new PDO assignment:





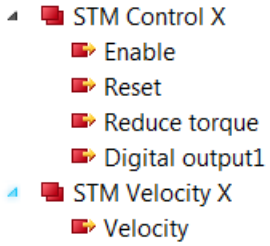
Index	Name	Flags	Value	Unit
1C13:0	SyncManager 3 assignment		> 8 <	
1C13:01	SubIndex 001	RW	0x1A01 (6657)	
1C13:02	SubIndex 002	RW	0x1A03 (6659)	
1C13:03	SubIndex 003	RW	0x1A11 (6673)	
1C13:04	SubIndex 004	RW	0x1A13 (6675)	
1C13:05	SubIndex 005	RW	0x1A21 (6689)	
1C13:06	SubIndex 006	RW	0x1A23 (6691)	
1C13:07	SubIndex 007	RW	0x1A31 (6705)	
1C13:08	SubIndex 008	RW	0x1A33 (6707)	
1C13:09	SubIndex 009	RW	---	
1C13:0A	SubIndex 010	RW	---	
1C13:0B	SubIndex 011	RW	---	
1C13:0C	SubIndex 012	RW	---	
1C13:0D	SubIndex 013	RW	---	
1C13:0E	SubIndex 014	RW	---	
1C13:0F	SubIndex 015	RW	---	

Step 2: Set the motor torque (see "Positioning interface", chapter 6.2 Step 2:)

Step 3: Set the number of micro-steps per full step (8n10:07). The motor runs smoother and with less vibration with higher micro-steps value setting.

Step 4: Set the motion parameters. Follow the steps described for the position control (chapter 6.2 Step 4:)

Step 5: Motion execution procedure:



1. Activate the "Enable" (0x7n10:01) parameter
2. Set the velocity (unit: step/second). The driver will immediately accelerate the motor to the set speed and continuously run at this speed until a new speed has been received. The motor will stop if the speed is set to zero or the "Enable" (0x7n10:01) flag has been put to FALSE or an error occurred.

8 CoE Interface

8.1 General Description

The CoE interface (CANopen over EtherCAT) is used for parameter management of EtherCAT devices. The CoE interface displays all the objects and parameters which are required for operating and diagnosing the ECAT-2094S device. Some parameters are fixed and can not be modified, they for example indicate the operating status of the device or the device properties. Motion related parameter need to be set before the actual motion control starts. These parameter setting are determined by the controlled stepper motor type and the setup of the motion application system.

CoE parameters has to be accessed via the CAN over EtherCAT protocol. The EtherCAT master accesses the local CoE lists of the slaves via CAN over EtherCAT. The user does not need to understand the CoE protocol when using the TwinCAT System Manager for CoE parameter configuration.

The CoE parameter describe a wide range of features such as manufacturer ID, device name, process data settings, calibration values for the stepper motor such as the current output, microsteps per full step, maximum velocity, etc..

The relevant ranges of the CoE list are:

- 0x1000: Stores fixed information of the device, including name, manufacturer, serial number etc.. In addition stores information about the current and available process data configurations.
  - 0x1600: RxPDO mapping
  - 0x1A00: TxPDO mapping
- 0x8000: Stores all the configuration data which are required for the stepper motor control.
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

The Figure 31 shows some of the CoE objects available for the ECAT-2094S device, ranging from 0x1000 to 0xF008. The parameters of the objects can be accessed by expanding the tree in the "CoE-Online" tab. The objects and their properties are described in chapter 9.

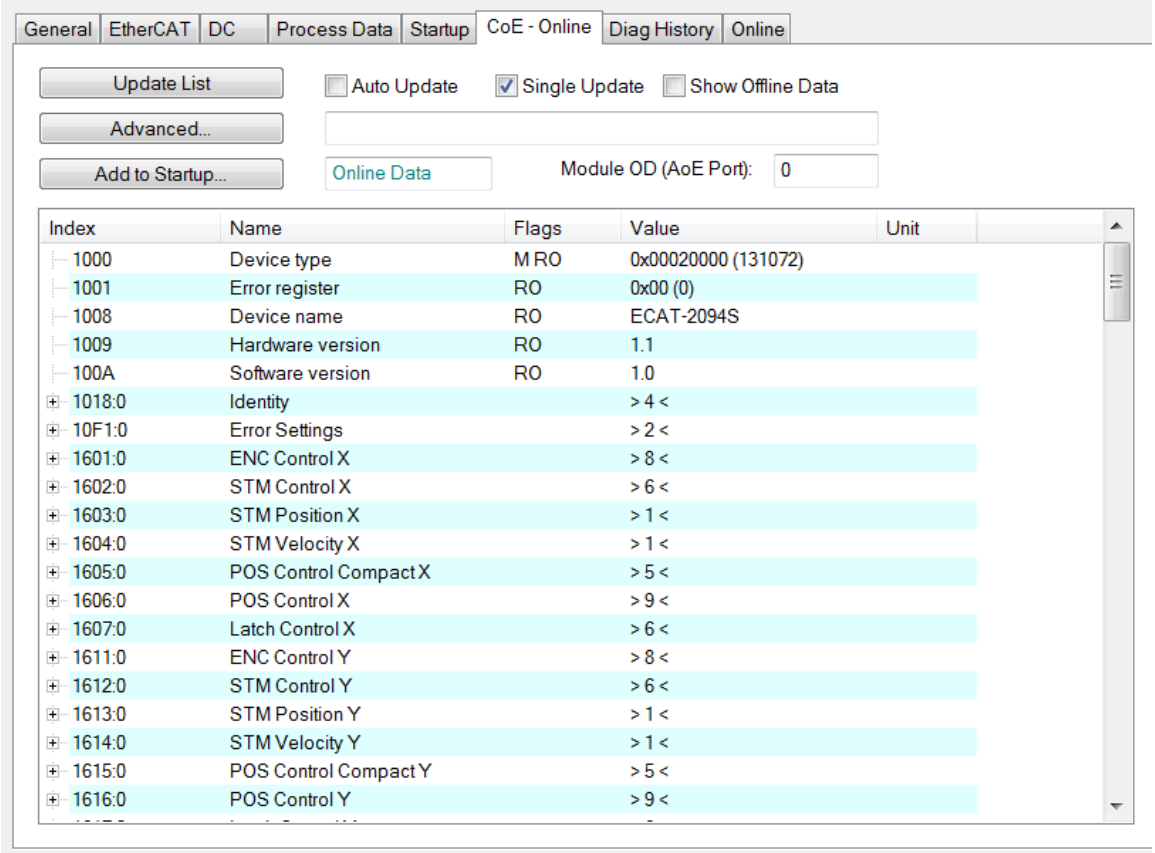


Figure 31: "CoE - Online " tab

8.2 Save Configuration Data to Memory

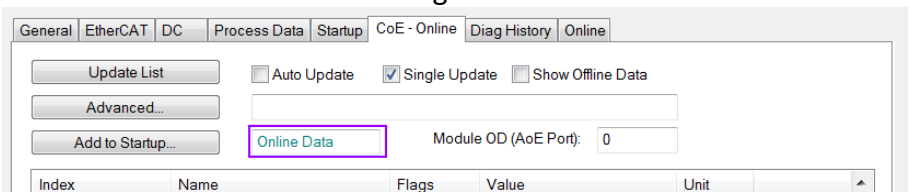
In this section the procedure of saving motion configuration parameters to the device non-volatile memory is being discussed.

The CoE object range 0x8000 to 0x8321 contains all the motion related parameters which are configurable and storable. TwinCAT allows the user to set the configuration parameters via the System Manager (Figure 31) or from a TwinCAT PLC via ADS (TcEtherCAT.lib library).

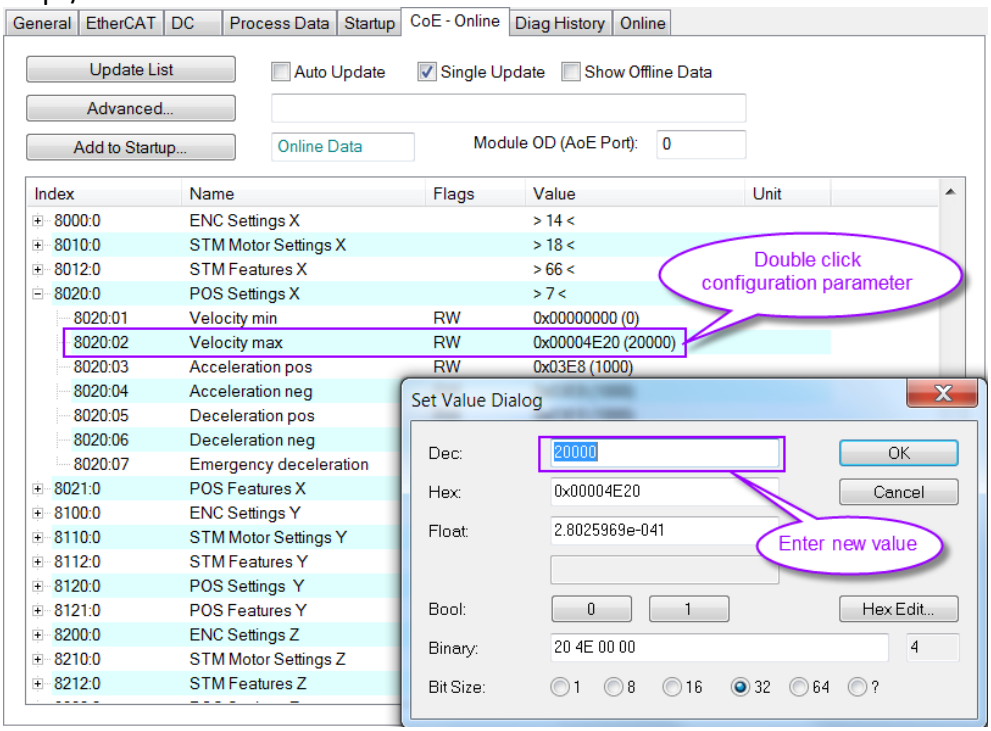
If slave CoE parameters are modified online, the ECAT-2094S device does not automatically store the data to a non-volatile memory. The data are lost if the device is switched off. The 0xF008 object provides functions to store the modified configuration data to the non-volatile memory of the device and the setting will be immediately available after a restart.

Procedure for storing configuration data to the local ECAT-2094S memory:

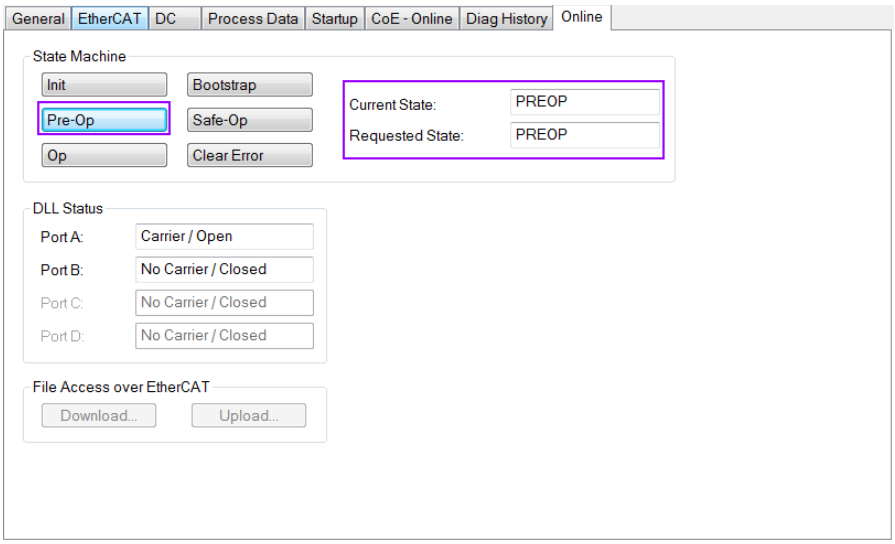
Step 1: Make sure the TwinCAT System Manager is connected to the ECAT-2094S and the "CoE-Online" tab is showing that the slave is online.



Step 2: Set all the necessary configuration objects (0x8n00, 0x8n10, 0x8n12, 0x8n21) for the X, Y, Z and U motor (n=0 to 3). Setting is being done by double clicking the configuration parameter and entering a new value in the popup window. In the following picture the maximum velocity of motor X is being set to 20000 steps/second.

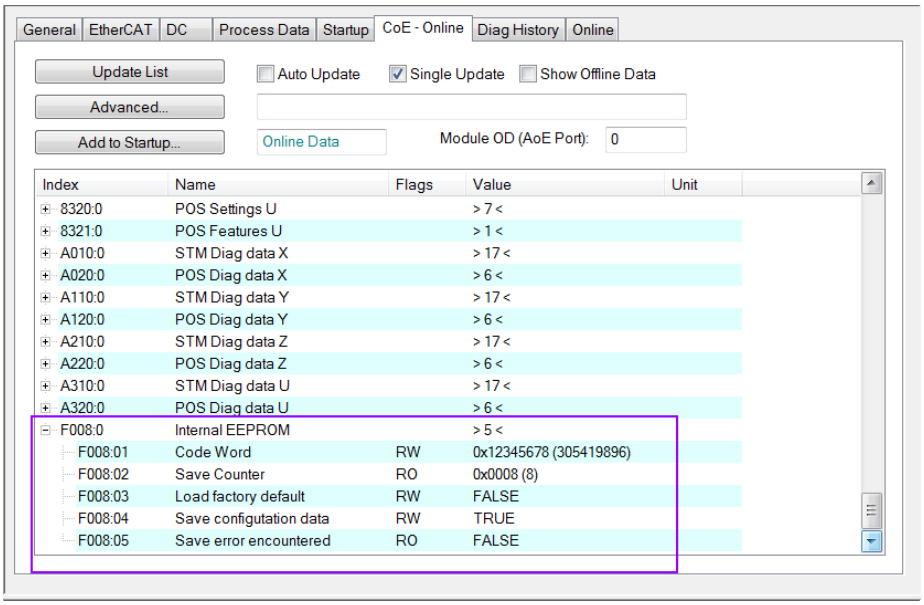


Step 3: After all the configurations have been done set the slave into Pre-Op mode. Data can only be stored to the local device if it is in Pre-Op mode. On the "Online" tab click the "Pre-OP" button to put the slave into Pre-OP mode.

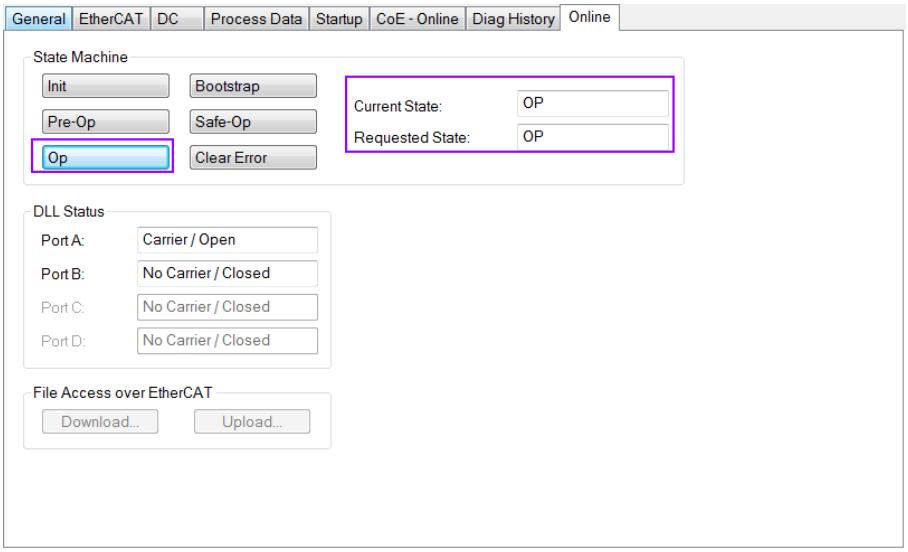


Step 4: The parameters of the 0xF008 object handles the save procedure.

1. Scroll to the end of the "CoE-Online" list and expand the tree view of the 0xF008 object
2. Enter the value 0x12345678 for the "Code Word"
3. Set the "Save configuration data" from FALSE to TRUE in order to save the configuration data to the internal EEPROM. The parameter "Save error encountered" (F008:05) indicates whether an error occurred during save process.
4. In order for the user configuration data to take effect after device restart set the "Load factory default" to FALSE. It is always possible to return to the factory default setting by setting this value back to TRUE.
5. The "Save Counter" (F008:02) shows how often configuration data has been stored to the local memory in the lifetime of the device.
   ATTENTION:
   The local memory only supports a limited number of save operations. Depending on the memory version once the save operation exceeds 10000 cycles it can no longer be guaranteed that data are reliably saved or are still readable. Therefore the "Save configuration data" (F008:04) and the "Load factory default" (F008:03) should not be continuously set from the controlling application program.



Step 5: Set the ECAT-2094S back into OP mode.



9 Object Description and Parameterization

9.1 Standard Objects

Index 1000 Device type

Index (hex)	Name	Description	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave	UINT32	RO	0x00020000

Index 1008 Device name

Index (hex)	Name	Description	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	ECAT-2094S

Index 1009 Hardware version

Index (hex)	Name	Description	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	1.1 (or greater)

Index 100A Software version

Index (hex)	Name	Description	Data type	Flags	Default
100A:0	Software version	Software version of the EtherCAT slave	STRING	RO	1.0 (or greater)

Index 1018 Identity

Index (hex)	Name	Description	Data type	Flags	Default
1018:0	Identity		UINT8	RO	0x04
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00494350
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x00209453
1018:03	Revision	Revision number of the EtherCAT slave	UINT32	RO	0x00010000
1018:04	Serial number	Serial number of the EtherCAT slave (not supported)	UINT32	RO	0x00000000

Index 10F1 Error settings

Index (hex)	Name	Description	Data type	Flags	Default
10F1:0	Error settings		UINT8	RO	0x02
10F1:01	Local error reaction	Not implemented	UINT32	RW	0x00000001
10F1:02	Sync error counter limit	For DC mode only: The Sync Error Counter is incremented with every missing Sync Management Event by three and decremented by one if an event is	UINT16	RW	0x0004

		received. If the Sync Error Counter exceeds this limit the system changes into the SAFEOP state with the 'Synchronization Lost' error. The Sync Error Counter is reset when the error was acknowledged.			

9.2 RxPDO Mapping Objects

The symbol **n** represents the axis number: 0 to 3.

Index 16n1 ENC Control (RxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
16n1:0	ENC Control	Encoder control	UINT8	RO	0x08
16n1:01	Status__Gap1	empty (1 Bit)	UINT32	RO	0x00000001
16n1:02	Control__Set encoder	Set encoder (1 Bit)	UINT32	RO	0x7n000201
16n1:03	Control__Set position counter	Set position counter (1 Bit)	UINT32	RO	0x7n000301
16n1:04	Control__Set encoder z latch-clear mode	Activate the encoder index latch-clear mode (1 Bit)	UINT32	RO	0x7n000401
16n1:05	Control__Gap2	BYTE padding (4 Bit)	UINT32	RO	0x00000004
16n1:06	Control_Encoder z latch-clear mode	Encoder index clear mode (8 Bit)	UINT32	RO	0x7n000608
16n1:07	Control__Set encoder value	Set encoder value (32-bit)	UINT32	RO	0x7n001120
16n1:08	Control__Set position counter value	Set position counter value (32-bit)	UINT32	RO	0x7n001220

Index 16n2 STM Control (RxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
16n2:0	STM Control	Stepper motor control	UINT8	RO	0x06
16n2:01	Control__Enable	Enable (1 Bit)	UINT32	RO	0x7n100101

16n2:02	Control__Reset	Reset (1 Bit)	UINT32	RO	0x7n100201
16n2:03	Control__Reduce torque	Reduce torque (1 Bit)	UINT32	RO	0x7n100301
16n2:04	Control__Gap1	BYTE padding (5 Bit)	UINT32	RO	0x00000005
16n2:05	Control__Digital output1	Digital output1 (1 Bit)	UINT32	RO	0x7n100C01
16n2:06	Control__Gap2	BYTE padding (7 Bit)	UINT32	RO	0x00000007

Index 16n3 STM Position (RxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
16n3:0	STM Position	Stepper motor position control	UINT8	RO	0x01
16n3:01	Control__Position	Position (32 Bit)	UINT32	RO	0x7n101120

Index 16n4 STM Velocity (RxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
16n4:0	STM Velocity	Stepper motor velocity control	UINT8	RO	0x01
16n4:01	Control__Velocity	Velocity (32 Bit)	UINT32	RO	0x7n102120

Index 16n5 POS Control Compact (RxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
16n5:0	POS Control Compact	Stepper motor compact control	UINT8	RO	0x05
16n5:01	Control__Execute	Execute (1 Bit)	UINT32	RO	0x7n200101
16n5:02	Control__Emergency stop	Emergency stop (1 Bit)	UINT32	RO	0x7n200201
16n5:03	Control__Gap1	BYTE padding (6 Bit)	UINT32	RO	0x00000006
16n5:04	Control__Gap2	WORD padding (8 Bit)	UINT32	RO	0x00000008
16n5:05	Control__Target position	Target position (32 Bit)	UINT32	RO	0x7n201120

Index 16n6 POS Control (RxPDO-Map)

Index	Name	Description	Data type	Flags	Default

(hex)					
16n6:0	POS Control	Position control	UINT8	RO	0x09
16n6:01	Control__Execute	Execute (1 Bit)	UINT32	RO	0x7n200101
16n6:02	Control__Emergency stop	Emergency stop (1 Bit)	UINT32	RO	0x7n200201
16n6:03	Control__Gap1	BYTE padding (6 Bit)	UINT32	RO	0x00000006
16n6:04	Control__Gap2	WORD padding (8 Bit)	UINT32	RO	0x00000008
16n6:05	Control__Target position	Target position (32 Bit)	UINT32	RO	0x7n201120
16n6:06	Control__Velocity	Max Velocity (32 Bit)	UINT32	RO	0x7n202120
16n6:07	Control__Start type	Start type (16 Bit)	UINT32	RO	0x7n202210
16n6:08	Control__Acceleration	Acceleration (16 Bit)	UINT32	RO	0x7n202310
16n6:09	Control__Deceleration	Deceleration (16 Bit)	UINT32	RO	0x7n202410

Index 16n7 Latch Control (RxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
16n7:0	Latch Control	Latch setting	UINT8	RO	0x06
16n7:01	Control__Enable latch active edge DI 1	Enable Latch extern on positive edge DI 1 (1 Bit)	UINT32	RO	0x7n300101
16n7:02	Control__Enable latch active edge DI 2	Enable Latch extern on positive edge DI 2 (1 Bit)	UINT32	RO	0x7n300201
16n7:03	Control__Enable Latch inactive edge DI 1	Enable Latch extern on negative edge DI 1 (1 Bit)	UINT32	RO	0x7n300301
16n7:04	Control__Enable Latch inactive edge DI 2	Enable Latch extern on negative edge DI 2 (1 Bit)	UINT32	RO	0x7n300401
16n7:05	Control__Gap1	BYTE padding (4 Bit)	UINT32	RO	0x00000004
16n7:06	Control__Gap2	WORD padding (8 Bit)	UINT32	RO	0x00000008

9.3 TxPDO Mapping Objects

The symbol **n** represents the axis number: 0 to 3.

Index 1An1 ENC Status (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An1:0	ENC Status	Encoder status	UINT8	RO	0x0D
1An1:01	Status__Gap1	empty (1 Bit)	UINT32	RO	0x00000001
1An1:02	Status__Gap2	empty (1 Bit)	UINT32	RO	0x00000001
1An1:03	Status__Set encoder done	Set position counter done (1 Bit)	UINT32	RO	0x6n000301
1An1:04	Status__Set encoder z latch-clear mode done	Set z latch clear mode done(1 Bit)	UINT32	RO	0x6n000401
1An1:05	Status__Counter underflow	Counter underflow (1 Bit)	UINT32	RO	0x6n000501
1An1:06	Status__Counter overflow	Counter overflow (1 Bit)	UINT32	RO	0x6n000601
1An1:07	Status__Index	Encoder index event (1 Bit)	UINT32	RO	0x6n000701
1An1:08	Status__Gap3	BYTE padding (1 Bit)	UINT32	RO	0x00000001
1An1:09	Status__Gap4	empty (5 Bit)	UINT32	RO	0x00000005
1An1:0A	Status__Sync error	Sync error (1 Bit)	UINT32	RO	0x6n000E01
1An1:0B	Status__Gap5	empty (1 Bit)	UINT32	RO	0x00000001
1An1:0C	Status__TxPDO Toggle	TxPDO Toggle (1 Bit)	UINT32	RO	0x6n001001
1An1:0D	Status__Encoder value	Encoder value (32-Bit)	UINT32	RO	0x6n001120

Index 1An3 STM Status (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An3:0	STM Status	Stepper motor status	UINT8	RO	0x10
1An3:01	Status__Ready to enable	Ready to enable (1 Bit)	UINT32	RO	0x6n100101
1An3:02	Status__Ready	Ready (1 Bit)	UINT32	RO	0x6n100201
1An3:03	Status__Warning	Warning (1 Bit)	UINT32	RO	0x6n100301
1An3:04	Status__Overtemperatur	Pre-Warning overtemperature (1 Bit)	UINT32	RO	0x6n100401
1An3:05	Status__Warning open load	Open load detected on	UINT32	RO	0x60100501

		phase A or B			
1An3:05	Status__Error	Error (1 Bit)	UINT32	RO	0x6n100601
1An3:06	Status__Error overtemperature	Error overtemperature (1 Bit)	UINT32	RO	0x6n100701
1An3:07	Status__Error short to ground	Error short to ground (1 Bit)	UINT32	RO	0x6n100801
1An3:09	Status__Moving positive	Moving positive (1 Bit)	UINT32	RO	0x6n100901
1An3:0A	Status__Moving negative	Moving negative (1 Bit)	UINT32	RO	0x6n100A01
1An3:0B	Status__Torque reduced	Torque reduced (1 Bit)	UINT32	RO	0x6n100B01
1An3:0C	Status__Digital input 1	Digital input 1 (1 Bit)	UINT32	RO	0x6n100C01
1An3:0D	Status__Digital input 2	Digital input 2 (1 Bit)	UINT32	RO	0x6n100D01
1An3:0E	Status__Sync error	Sync error (1 Bit)	UINT32	RO	0x6n100E01
1An3:0F	Status__Motor standstill	Motor is at standstill (1 Bit)	UINT32	RO	0x6n100F01
1An3:10	Status__TxPDO Toggle	TxPDO Toggle (1 Bit)	UINT32	RO	0x6n101001

Index 1An4 STM Synchron Info Data (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An4:0	STM Synchron Info Data		UINT8	RO	0x02
1An4:01	Status__Info data 1	Info data 1 (32 Bit)	UINT32	RO	0x6n101120
1An4:02	Status__Info data 2	Info data 2 (32 Bit)	UINT32	RO	0x6n101220

Index 1An5 POS Status Compact (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An5:0	POS Status Compact		UINT8	RO	0x0B
1An5:01	Status__Busy	Busy (1 Bit)	UINT32	RO	0x6n200101
1An5:02	Status__In-Target	In-Target (1 Bit)	UINT32	RO	0x6n200201
1An5:03	Status__Warning	Warning (1 Bit)	UINT32	RO	0x6n200301
1An5:04	Status__Error	Error (1 Bit)	UINT32	RO	0x6n200401
1An5:05	Status__Gap1	Padding (1 Bit)	UINT32	RO	0x00000001
1An5:06	Status__Accelerate	Accelerate (1 Bit)	UINT32	RO	0x6n200601
1An5:07	Status__Decelerate	Decelerate	UINT32	RO	0x6n200701

		(1 Bit)			
1An5:08	Status__SoftEmg	Software Emergency (1 Bit)	UINT32	RO	0x6n200801
1An5:09	Status__CmdRejected	Command rejected (1 Bit)	UINT32	RO	0x6n200901
1An5:0A	Status__CmdAborted	Command Aborted (1 Bit)	UINT32	RO	0x6n200A01
1An5:0B	Status__Gap2	BYTE padding (6 Bit)	UINT32	RO	0x00000006

Index 1An6 POS Status (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An6:0	POS Status		UINT8	RO	0x0D
1An6:01	Status__Busy	Busy (1 Bit)	UINT32	RO	0x6n200101
1An6:02	Status__In-Target	In-Target (1 Bit)	UINT32	RO	0x6n200201
1An6:03	Status__Warning	Warning (1 Bit)	UINT32	RO	0x6n200301
1An6:04	Status__Error	Error (1 Bit)	UINT32	RO	0x6n200401
1An6:05	Status__Gap1	Padding (1 Bit)	UINT32	RO	0x00000001
1An6:06	Status__Accelerate	Accelerate (1 Bit)	UINT32	RO	0x6n200601
1An6:07	Status__Decelerate	Decelerate (1 Bit)	UINT32	RO	0x6n200701
1An6:08	Status__SoftEmg	Software Emergency (1 Bit)	UINT32	RO	0x6n200801
1An6:09	Status__CmdRejected	Command rejected (1 Bit)	UINT32	RO	0x6n200901
1An6:0A	Status__CmdAborted	Command Aborted (1 Bit)	UINT32	RO	0x6n200A01
1An6:0B	Status__Gap2	BYTE padding (6 Bit)	UINT32	RO	0x00000006
1An6:0C	Status__Actual motor position	Actual position (32 Bit)	UINT32	RO	0x6n201120
1An6:0D	Status__Actual motor velocity	Actual velocity (32 Bit)	UINT32	RO	0x6n202120

Index 1An7 Pos Counter Status (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An7:0	Pos Counter Status	Position counter status	UINT8	RO	0x02
1An7:01	Status__Set position counter done	Set position counter done (1 Bit)	UINT32	RO	0x6n202301

1An7:02	Status__Sync error	Sync error (1 Bit)	UINT32	RO	0x6n202401
1An7:03	Status__TxPDO Toggle	TxPDO Toggle (1 Bit)	UINT32	RO	0x6n202501
1An7:04	Status__Gap1	BYTE padding (5 Bit)	UINT32	RO	0x00000005
1An7:05	Status__Gap2	WORD padding (8 Bit)	UINT32	RO	0x00000008
1An7:06	Status__Position counter value	Position counter value (32-Bit)	UINT32	RO	0x6n201120

Index 1An8 Latch Status (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An8:0	Latch Status		UINT8	RO	0x09
1An8:01	Status__Latch extern valid DI 1	Latch DI 1 extern valid (1 Bit)	UINT32	RO	0x6n300101
1An8:02	Status__Latch extern valid DI 2	Latch DI 2 extern valid (1 Bit)	UINT32	RO	0x6n300201
1An8:03	Status__Status of extern latch DI 1	Status of the ext. latch input DI 1 (1 Bit)	UINT32	RO	0x6n300301
1An8:04	Status__Status of extern latch DI 2	Status of the ext. latch input DI 2 (1 Bit)	UINT32	RO	0x6n300401
1An8:05	Status__Gap1	BYTE padding (4 Bit)	UINT32	RO	0x00000004
1An8:06	Status__Gap2	empty (7 Bit)	UINT32	RO	0x00000007
1An8:07	Status__TxPDO Toggle	TxPDO Toggle (1 Bit)	UINT32	RO	0x6n301001
1An8:08	Status__Encoder latched value	Latched encoder value (32-Bit)	UINT32	RO	0x6n301220
1An8:09	Status__Position counter latched value	Latched position counter value (32-Bit)	UINT32	RO	0x6n301320

Index 1An9 Input Signal (TxPDO-Map)

Index (hex)	Name	Description	Data type	Flags	Default
1An9:0	Input Signal		UINT8	RO	0x08
1An9:01	Status__Left reference input	Left reference input (1 Bit)	UINT32	RO	0x6n101301
1An9:02	Status__Right reference input	Right reference input (1 Bit)	UINT32	RO	0x6n101401
1An9:03	Status__Encoder A channel input	Encoder A channel input (1 Bit)	UINT32	RO	0x6n101501

1An9:04	Status__Encoder B channel input	Encoder B channel input (1 Bit)	UINT32	RO	0x6n101601
1An9:05	Status__Encoder Z channel input	Encoder Z channel input (1 Bit)	UINT32	RO	0x6n101701
1An9:06	Status__Driver enable	Driver enabled signal (1 Bit)	UINT32	RO	0x6n101801
1An9:07	Status__Gap1	BYTE padding (2 Bit)	UINT32	RO	0x00000002
1An9:08	Status__Gap2	WORD padding (8 Bit)	UINT32	RO	0x00000008

9.4 Sync Manager Objects

Index 1C00 Sync manager type

Index (hex)	Name	Description	Data type	Flags	Default
1C00:0	Sync manager type	Using the sync managers	UINT8	RO	0x04
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04

Index 1C12 RxPDO assign

Index (hex)	Name	Description	Data type	Flags	Default*
1C12:0	RxPDO assign	SyncManager 2 assignment: PDO Assign Outputs	UINT8	RO	0x1C
1C12:01	SubIndex 001	default assignment: Velocity control	UINT16	RW	0x1601
1C12:02	SubIndex 002	default assignment: Velocity control	UINT16	RW	0x1602
1C12:03	SubIndex 003	default assignment: Velocity control	UINT16	RW	0x1604
1C12:04	SubIndex 004	default assignment: Velocity control	UINT16	RW	0x1611
1C12:05	SubIndex 005	default assignment: Velocity control	UINT16	RW	0x1612
1C12:06	SubIndex 006	default assignment: Velocity	UINT16	RW	0x1614

		control			
1C12:07	SubIndex 007	default assignment: Velocity control	UINT16	RW	0x1621
1C12:08	SubIndex 008	default assignment: Velocity control	UINT16	RW	0x1622
1C12:09	SubIndex 009	default assignment: Velocity control	UINT16	RW	0x1624
1C12:0A	SubIndex 010	default assignment: Velocity control	UINT16	RW	0x1631
1C12:0B	SubIndex 011	default assignment: Velocity control	UINT16	RW	0x1632
1C12:0C	SubIndex 012	default assignment: Velocity control	UINT16	RW	0x1634
1C12:0D	SubIndex 013	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:0E	SubIndex 014	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:0F	SubIndex 015	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:10	SubIndex 016	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:11	SubIndex 017	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:12	SubIndex 018	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:13	SubIndex 019	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:14	SubIndex 020	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:15	SubIndex 021	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:16	SubIndex 022	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:17	SubIndex 023	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:18	SubIndex 024	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:19	SubIndex 025	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:1A	SubIndex 026	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:1B	SubIndex 027	reserve space for additional RxPDO assignment	UINT16	RW	0x0000
1C12:1C	SubIndex 028	reserve space for additional RxPDO assignment	UINT16	RW	0x0000

*Sub index 001 to 028 contains the index of the associated RxPDO mapping object

Index 1C13 TxPDO assign

Index (hex)	Name	Description	Data type	Flags	Default*
1C13:0	TxPDO assign	SyncManager 3 assignment:	UINT8	RO	0x20

		PDO Assign Inputs			
1C13:01	SubIndex 001	default assignment: Velocity control	UINT16	RW	0x1A01
1C13:02	SubIndex 002	default assignment: Velocity control	UINT16	RW	0x1A03
1C13:03	SubIndex 003	default assignment: Velocity control	UINT16	RW	0x1A11
1C13:04	SubIndex 004	default assignment: Velocity control	UINT16	RW	0x1A13
1C13:05	SubIndex 005	default assignment: Velocity control	UINT16	RW	0x1A21
1C13:06	SubIndex 006	default assignment: Velocity control	UINT16	RW	0x1A23
1C13:07	SubIndex 007	default assignment: Velocity control	UINT16	RW	0x1A31
1C13:08	SubIndex 008	default assignment: Velocity control	UINT16	RW	0x1A33
1C13:09	SubIndex 009	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:0A	SubIndex 010	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:0B	SubIndex 011	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:0C	SubIndex 012	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:0D	SubIndex 013	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:0E	SubIndex 014	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:0F	SubIndex 015	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:10	SubIndex 016	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:11	SubIndex 017	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:12	SubIndex 018	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:13	SubIndex 019	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:14	SubIndex 020	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:15	SubIndex 021	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:16	SubIndex 022	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:17	SubIndex 023	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:18	SubIndex 024	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:19	SubIndex 025	reserve space for additional TxPDO assignment	UINT16	RW	0x0000

1C13:1A	SubIndex 026	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:1B	SubIndex 027	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:1C	SubIndex 028	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:1D	SubIndex 029	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:1E	SubIndex 030	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:1F	SubIndex 031	reserve space for additional TxPDO assignment	UINT16	RW	0x0000
1C13:20	SubIndex 032	reserve space for additional TxPDO assignment	UINT16	RW	0x0000

*Sub index 001 to 032 contains the index of the associated TxPDO mapping object

Index 1C32 Sync Manager (SM) output parameter

Index (hex)	Name	Description	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20
1C32:01	Synchronization Type	Current synchronization mode: • 0: Free Run • 1: Synchronous without SM 2 event • 2: DC-Mode - Synchronous with SYNC0 Event • 3: DC-Mode - Synchronous with SYNC1 event	UINT8	RO	0x0001
1C32:02	Cycle Time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT8	RO	0x00000000
1C32:04	Synchronization Types supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode)	UINT8	RO	0x8007
1C32:05	Minimum Cycle Time	Minimum cycle time (in ns)		RO	0x00000000
1C32:06	Calc and Copy Time	Minimum time between		RO	0x00000000

		SYNC0 and SYNC1 event (in ns, DC mode only)			
1C32:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started Set parameter to 1 in order to update the Cycle Time (1C32:02, 1C33:02) parameter with the maximum measured value		RW	0x0000
1C32:09	Delay Time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)		RO	0x00000000
1C32:0A	Sync0 Cycle Time			RW	0x00000000
1C32:0B	SM-Event Missed	Number of missed SM events in OPERATIONAL (DC mode only)		RO	0x0000
1C32:0C	Cycle Time Too Small	Cycle was not completed in time or the next cycle began too early		RO	0x0000
1C32:20	Sync Error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)		RO	FALSE

Index 1C33 Sync Manager (SM) input parameter

Index (hex)	Name	Description	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20
1C33:01	Synchronization Type	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT8	RO	0x0022
1C33:02	Cycle Time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT8	RO	0x00000000
1C33:04	Synchronization Types supported	Supported synchronization modes:	UINT8	RO	0x8007

		• Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode)			
1C33:05	Minimum Cycle Time	Minimum cycle time (in ns)		RO	0x00000000
1C33:06	Calc and Copy Time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)		RO	0x00000000
1C33:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started Set parameter to 1 in order to update the Cycle Time (1C32:02, 1C33:02) parameter with the maximum measured value		RW	0x0000
1C33:09	Delay Time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)		RO	0x00000000
1C33:0A	Sync0 Cycle Time			RW	0x00000000
1C33:0B	SM-Event Missed	Number of missed SM events in OPERATIONAL (DC mode only)		RO	0x0000
1C33:0C	Cycle Time Too Small	Cycle was not completed in time or the next cycle began too early		RO	0x0000
1C33:20	Sync Error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)		RO	FALSE

9.5 Input Data

The symbol **n** represents the axis number: 0 to 3.

Index 6n00 ENC Inputs

Index (hex)	Name	Description	Data type	Flags	Default
6n00:0	ENC Status	Encoder status inputs	UINT8	RO	0x11
6n00:03	Set encoder done	The encoder value has	BOOLEAN	RO	FALSE

		been set			
6n00:04	Set encoder z latch-clear mode done	Indicates whether the encoder index latch-clear mode was set successfully	BOOLEAN	RO	FALSE
6n00:05	Counter underflow	Counter underflow	BOOLEAN	RO	FALSE
6n00:06	Counter overflow	Counter overflow	BOOLEAN	RO	FALSE
6n00:07	Encoder index event	Encoder index event detected	BOOLEAN	RO	FALSE
6n00:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle	BOOLEAN	RO	FALSE
6n00:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated	BOOLEAN	RO	FALSE
6n00:11	Actual encoder value	The counter value	INT32	RO	0x00000000

Index 6n10 STM Inputs

Index (hex)	Name	Description	Data type	Flags	Default
6n10:0	STM Inputs	Stepper motor inputs	UINT8	RO	0x18
6n10:01	Ready to enable	Driver stage is ready for enabling	BOOLEAN	RO	FALSE
6n10:02	Ready	Driver stage is ready for operation	BOOLEAN	RO	FALSE
6n10:03	Warning	A warning has occurred	BOOLEAN	RO	FALSE
6n10:04	Warning over temperature	Over-temperature pre-warning	BOOLEAN	RO	FALSE
6n10:05	Warning open load	Open load detected on phase A or phase B • In motor stand still, open load cannot be measured, as the coils might eventually have zero current	BOOLEAN	RO	FALSE
6n10:06	Error	An error has occurred	BOOLEAN	RO	FALSE
6n10:07	Error over temperature	Over-temperature error	BOOLEAN	RO	FALSE
6n10:08	Error short to ground	Short to ground phase A or phase B	BOOLEAN	RO	FALSE
6n10:09	Moving positive	Motor turns in positive direction	BOOLEAN	RO	FALSE
6n10:0A	Moving negative	Motor turns in	BOOLEAN	RO	FALSE

		negative direction			
6n10:0B	Torque reduced	Reduced torque is active	BOOLEAN	RO	FALSE
6n10:0C	Digital input 1	Digital input 1	BOOLEAN	RO	FALSE
6n10:0D	Digital input 2	Digital input 2	BOOLEAN	RO	FALSE
6n10:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	RO	FALSE
6n10:0F	Motor standstill	Indicates the whether motor is in standstill (TRUE - standstill)	BOOLEAN	RO	FALSE
6n10:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	FALSE
6n10:11	Info data 1	Synchronous information (selection via sub index 8n12:11)	INT32	RO	0x00000000
6n10:12	Info data 2	Synchronous information (selection via sub index 8n12:19)	INT32	RO	0x00000000
6n10:13	Left reference input	Signal of the left reference input	BOOLEAN	RO	FALSE
6n10:14	Right reference input	Signal of the right reference input	BOOLEAN	RO	FALSE
6n10:15	Encoder A channel input	Signal of the encoder A channel	BOOLEAN	RO	FALSE
6n10:16	Encoder B channel input	Signal of the encoder B channel	BOOLEAN	RO	FALSE
6n10:17	Encoder Z channel input	Signal of the encoder Z channel	BOOLEAN	RO	FALSE
6n10:18	Driver disabled	Indicates whether the driver has been enabled	BOOLEAN	RO	FALSE

Index 6n20 POS Inputs

Index (hex)	Name	Description	Data type	Flags	Default
6n20:0	POS Inputs		UINT8	RO	0x25
6n20:01	Busy	A travel command is active	BOOLEAN	RO	FALSE
6n20:02	In-Target	Motor has arrived at target	BOOLEAN	RO	FALSE
6n20:03	Warning	A warning has occurred	BOOLEAN	RO	FALSE

6n20:04	Error	An error has occurred	BOOLEAN	RO	FALSE
6n20:05	Calibrated_xx	Motor is calibrated (not supported)	BOOLEAN	RO	FALSE
6n20:06	Accelerate	Motor is in the acceleration phase	BOOLEAN	RO	FALSE
6n20:07	Decelerate	Motor is in the deceleration phase	BOOLEAN	RO	FALSE
6n20:08	Soft Emg	Emergency stop has been triggered by software	BOOLEAN	RO	FALSE
6n20:09	Cmd rejected	Motion command has been reject	BOOLEAN	RO	FALSE
6n20:0A	Cmd aborted	Motion command has been aborted	BOOLEAN	RO	FALSE
6n20:11	Actual motor position	Current target position of the travel command generator	INT32	RO	0x00000000
6n20:21	Actual motor velocity	Current velocity of the travel command generator	INT32	RO	0x00000000
6n20:22	Actual drive time_xx	Travel command time information (see subindex 8n21:11) (Not supported)	UINT32	RO	0x00000000
6n20:23	Set position counter done	The position counter has been set	BOOLEAN	RO	FALSE
6n20:24	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle	BOOLEAN	RO	FALSE
6n20:25	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	FALSE

Index 6n30 Latch Inputs

Index (hex)	Name	Description	Data type	Flags	Default
6n30:0	Latch Inputs		UINT8	RO	0x13
6n30:01	Latch extern valid DI 1	The counter value was stored via the external latch DI 1	BOOLEAN	RO	FALSE
6n30:02	Latch extern valid DI 2	The counter value was stored via the external latch DI 2	BOOLEAN	RO	FALSE
6n30:03	Status of extern latch DI 1	Status of the ext. latch	BOOLEAN	RO	FALSE

		input DI 1			
6n30:04	Status of extern latch DI 2	Status of the ext. latch input DI 2	BOOLEAN	RO	FALSE
6n30:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated	BOOLEAN	RO	FALSE
6n30:12	Latched encoder value	Latched encoder value	INT32	RO	0x00000000
6n30:13	Latched position counter value	Latched position counter value	INT32	RO	0x00000000

9.6 Output Data

The symbol **n** represents the axis number: 0 to 3.

Index 7n00 ENC Outputs

Index (hex)	Name	Description	Data type	Flags	Default
7n00:0	ENC Outputs		UINT8	RO	0x12
7n00:02	Set encoder	Assigned value to encoder counter • By setting the bit from FALSE to TRUE the encoder value (7n00:11) will be set	BOOLEAN	RO	FALSE
7n00:03	Set position counter	Assigned value to position counter • By setting the bit from FALSE to TRUE the position counter value (7n00:12) will be set	BOOLEAN	RO	FALSE
7n00:04	Set encoder z latch-clear mode	Set the latch and clear mode of the encoder counter in case of an index (z) event • By setting the bit from FALSE to TRUE the Encoder z latch-clear mode (7n00:06) will be set	BOOLEAN	RO	FALSE
7n00:06	Encoder z latch-clear mode	Latch-clear mode of the encoder for an index (z) event • Valid range: 0x00 ~ 0x04	BIT8	RO	0x00

		• 0: Disable index latch • 1: Latch index once. The encoder value will be latched by the index signal only once after 7n00:06 has been set to 1 • 2: Latch continuous: After setting 7n00:06 to 2 the encoder value will be latched each time an index latch occurs. • 3: Latch and clear once: After setting 7n00:06 to 3 the encoder value will be latched and the encoder counter will be cleared at the first index signal encountered • 4: Latch and clear continuous: After setting 7n00:06 to 4 the encoder value will be latched and the encoder counter will be cleared after each index signal event			
7n00:11	Set encoder value	Encoder value to be set via "Set encoder"	INT32	RO	0x00000000
7n00:12	Set position counter value	Position counter value to be set via "Set position counter"	INT32	RO	0x00000000

Index 7n10 STM Outputs

Index (hex)	Name	Description	Data type	Flags	Default
7n10:0	STM Outputs	Stepper motor outputs	UINT8	RO	0x21
7n10:01	Enable	Activates the output stage • Enable = TRUE Output "Maximal current" (0x8n10:01) • Enable = FALSE Output to "Power on motor current"	BOOLEAN	RO	FALSE

		(0x8n10:08)			
7n10:02	Reset	All errors that may have occurred are reset by setting this bit (rising edge) • By setting the bit from FALSE to TRUE errors which occurred during motion execution (e.g. over-temperature) will be cleared	BOOLEAN	RO	FALSE
7n10:03	Reduce torque	Activation of reduced torque (coil current) (sub index 8n10:02)	BOOLEAN	RO	FALSE
7n10:0c	Digital output1	Digital output1	BOOLEAN	RO	FALSE
7n10:11	Position	Set position; Absolute target position for the "Position control" mode (see chapter 6.4)	INT32	RO	0x00000000
7n10:21	Velocity	Set velocity The target velocity for the "Velocity control" mode (see chapter 7)	INT32	RO	0x00000000

Index 7n20 POS Outputs

Index (hex)	Name	Description	Data type	Flags	Default
7n20:0	POS Outputs		UINT8	RO	0x24
7n20:01	Execute	Start travel command (rising edge), or prematurely abort travel command (falling edge)	BOOLEAN	RO	FALSE
7n20:02	Emergency stop	Prematurely abort travel command with an emergency ramp (rising edge)	BOOLEAN	RO	FALSE
7n20:11	Target position	Specification of the target position (unit: steps). • Depending on the "Start type" (0x7n20:22) the position can either be relative, absolute or additive	INT32	RO	0x00007FFF
7n20:21	Velocity	Specification of the	INT32	RO	0x00000000

		maximum set velocity (unit: steps/second)			
7n20:22	Start type	Specification of the start types (see Table 9: Start type definition)	UINT16	RO	0x0000
7n20:23	Acceleration	Acceleration time (unit: milliseconds) • The acceleration time (unit: milliseconds) is defined as the time to accelerate the motor from "Velocity min" (0x8n20:01) to "Velocity max" (0x8n20:02)	UINT16	RO	0x0000
7n20:24	Deceleration	Deceleration time (unit: milliseconds) • Deceleration time is defined as the time required to decelerate from "Velocity max" (0x8n20:02) to "Velocity min" (0x8n20:01)	UINT16	RO	0x0000

Index 7n30 Latch Outputs

Index (hex)	Name	Description	Data type	Flags	Default
7n30:0	Latch Outputs		UINT8	RO	0x04
7n30:01	Enable latch active edge DI 1	DI 1 trigger level: rising edge	BOOLEAN	RO	FALSE
7n30:02	Enable latch active edge DI 2	DI 2 trigger level: rising edge	BOOLEAN	RO	FALSE
7n30:03	Enable latch inactive edge DI 1	DI 1 trigger level: falling edge	BOOLEAN	RO	FALSE
7n30:04	Enable latch inactive edge DI 2	DI 2 trigger level: falling edge	BOOLEAN	RO	FALSE

9.7 Configuration Data

The symbol **n** represents the axis number: 0 to 3.

Index 8n00 ENC Settings

Index (hex)	Name	Description	Data type	Flags	Default
8n00:0	ENC Settings	Encoder settings	UINT8	RO	0x0E
8n00:0E	Reversion of rotation	Activates reversion of rotation of the encoder	BOOLEAN	RW	FALSE

Index 8n10 STM Motor Settings

Index (hex)	Name	Description	Data type	Flags	Default
8n10:0	STM Motor Settings	Stepper motor settings	UINT8	RO	0x12
8n10:01	Maximum run current	Peak motor coil current for driving (unit: mA), default: 750, max: 1500	UINT16	RW	0x02EE (750)
8n10:02	Reduced run current	Reduced peak motor coil current for driving(reduced torque, unit: mA), default: 375, max: 1500 Will be activated when "Reduced torque" (0x7n10:03) has been set to true	UINT16	RW	0x0177 (375)
8n10:03	Maximum hold current	Motor standstill current (unit: mA), default: 750, max: 1500	UINT16	RW	0x02EE (750)
8n10:04	Reduced hold current	Reduced Motor standstill current (unit: mA), default: 375, max: 1500. Will be activated when "Reduced torque" (0x7n10:03) has been set to true	UINT16	RW	0x0177 (375)
8n10:06	Motor fullsteps	Motor full steps per revolution (not supported)	UINT16	RW	0x0000
8n10:07	Micro Steps	Number of microsteps per full step. • Supported values: 256, 128, 64, 32, 16, 8, 4, 2, 1	DT0801EN16	RW	0x0008 ("256")
8n10:08	Power on motor current	Motor coil current output directly after power on (unit: 1 mA)	UINT16	RW	0x0177 (375)
8n10:09	Max Start Velocity	Maximum possible	UINT16	RW	0x0064

		start velocity of the motor			(100)
8n10:12	Safe motor current	Set the safe motor coil current (will be applied if state changes from OP to a different state)(unit: 1 mA)	UINT16	RW	0x0177 (375)

Index 8n12 STM Features

Index (hex)	Name	Description	Data type	Flags	Default
8n12:0	STM Features	Stepper motor features	UINT8	RO	0x42
8n12:01	Operation mode	Operating mode, • 0: automatic • 2: velocity control • 3: position control	DT0802EN04	RW	0x00 ("Automatic")
8n12:09	Invert motor polarity	Activates reversal of the motor rotation direction.	BOOLEAN	RW	FALSE
8n12:11	Select info data 1	Select "Info data 1": • 3: Motor coil current A • 4: Motor coil current B • 7: Motor velocity • 8: Encoder position • 9: Position counter	DT0803EN08	RW	0x03 ("Motor coil current A")
8n12:19	Select info data 2	Select "Info data 2": • 3: Motor coil current A • 4: Motor coil current B • 7: Motor velocity • 8: Encoder position • 9: Position counter	DT0803EN08	RW	0x04 ("Motor coil current B")
8n12:2A	Power on DO 1	Set the power on DO 1 (will be applied directly after switching the device on)	BOOLEAN	RW	FALSE
8n12:2B	Safety DO 1	Set the safe DO 1 (will be applied if state changes from OP to different state)	BOOLEAN	RW	FALSE
8n12:30	Invert digital input 1	Inversion of digital input 1	BOOLEAN	RW	FALSE
8n12:31	Invert digital input 2	Inversion of digital input 2	BOOLEAN	RW	FALSE

8n12:32	Function for input 1	Select the digital input 1 type: • 0: Normal input • 1: Hardware stop enable	DT080AEN04	RW	0x00 ("Normal input")
8n12:36	Function for input 2	Select the digital input 2 type: • 0: Normal input • 1: Hardware stop enable	DT080AEN04	RW	0x00 ("Normal input")
8n12:37	Limit switch stop mode	Soft stop not supported	DT080BEN01	RW	0x00 ("Limit switch hard stop")
8n12:40	Encoder index latch trigger	Latch trigger setting for the encoder index (z) pulse: • 0: Level trigger • 1: Rising edge trigger • 2: Falling edge trigger • 3: Rising and falling edge trigger	DT0811EN03	RW	0x00 ("Level trigger")
8n12:42	Encoder index polarity	Active polarity of the encoder index (z): • 0: Low active • 1: High active	DT0813EN01	RW	0x01 ("High active")

Index 8n20 POS Settings

Index (hex)	Name	Description	Data type	Flags	Default
8n20:0	POS Settings	Position settings	UINT8	RO	0x07
8n20:01	Velocity min	Minimum set velocity	UINT32	RW	0x00000000
8n20:02	Velocity max	Maximum set velocity • Maximum velocity supported by the system	UINT32	RW	0x00002710 (10000)
8n20:03	Acceleration pos	Acceleration time in positive direction of rotation (unit: milliseconds)	UINT16	RW	0x03E8 (1000)
8n20:04	Acceleration neg	Acceleration time in negative direction of rotation (unit: milliseconds)	UINT16	RW	0x03E8 (1000)
8n20:05	Deceleration pos	Deceleration time in positive direction of rotation (unit: milliseconds)	UINT16	RW	0x03E8 (1000)
8n20:06	Deceleration neg	Deceleration time in	UINT16	RW	0x03E8

		negative direction of rotation (unit: milliseconds)			(1000)
8n20:17	Emergency deceleration	Emergency deceleration time (both directions of rotation, unit: milliseconds)	UINT16	RW	0x0000

Index 8n21 POS Features

Index (hex)	Name	Description	Data type	Flags	Default
8n21:0	POS Features		UINT8	RO	0x01
8n21:01	Start type	Standard start type: • 0: Idle, • 1: Absolute, • 2: Relative, • 6: Additive, • 1001: absolute change, • 1002: relative change, • 1006: additive change	DT080FEN16	RW	0x0002 ("Relative")

9.8 Information and Diagnostic Data

The symbol **n** represents the axis number: 0 to 3.

Index An10 STM Diag data

Index (hex)	Name	Description	Data type	Flags	Default
An10:0	STM Diag data	Stepper motor diagnostic data	UINT8	RO	0x11
An10:02	Over temperature	Driver IC temperature has reached more than 80 °C • ATTENTION: This error message must be acknowledged by the user (see index 0x7n10:02)	BOOLEAN	RO	FALSE
An10:03	Torque overload	Not supported	BOOLEAN	RO	FALSE
An10:04	Under voltage	Indicates an undervoltage on the charge pump. The driver	BOOLEAN	RO	FALSE

		is disabled in this case • ATTENTION: This error message must be acknowledged by the user (see index 0x7n10:02)			
An10:05	Over voltage	Not supported	BOOLEAN	RO	FALSE
An10:06	Short circuit A	Short to GND detected on phase A. The driver becomes disabled.	BOOLEAN	RO	FALSE
An10:07	Short circuit B	Short to GND detected on phase B. The driver becomes disabled.	BOOLEAN	RO	FALSE
An10:08	No control power	Not supported	BOOLEAN	RO	FALSE
An10:09	Misc error	Driver has been shut down due to - overtemperature - short circuit detection - undervoltage "uv_cp" ATTENTION: This error message must be acknowledged by the user (see index 0x7n10:02)	BOOLEAN	RO	FALSE
An10:0a	Configuration	Not supported	BOOLEAN	RO	FALSE
An10:11	Actual operation mode	Not supported	DT0809EN04	RO	0x00 ("undefined")

Index An20 POS Diag data

Index (hex)	Name	Description	Data type	Flags	Default
An20:0	POS Diag data	Stepper motor diagnostic data for position control	UINT8	RO	0x06
An20:01	Command rejected	Dynamic change of the target position was not accepted	BOOLEAN	RO	FALSE
An20:02	Command aborted	Command aborted due to internal error or emergency stop	BOOLEAN	RO	FALSE
An20:03	Target overrun	Change the target position on the fly may lead to an overshoot of the position and therefore change in direction of rotation may be necessary	BOOLEAN	RO	FALSE

An20:04	Target timeout	Not supported	BOOLEAN	RO	FALSE
An20:05	Position lag	Not supported	BOOLEAN	RO	FALSE
An20:06	Emergency stop	Emergency stop	BOOLEAN	RO	FALSE

9.9 Configuration Parameters Storage

Index F008 Internal EEPROM

Index (hex)	Name	Description	Data type	Flags	Default
F008:0	Internal EEPROM	Storing CoE parameters to the internal EEPROM.	UINT8	RO	0x05
F008:01	Code Word	Password for saving CoE configuration data to the EEPROM Password: 0x12345678	UINT32	RW	0x00000000
F008:02	Save Counter	Total number of save sequence	UINT16	RO	0x0000
F008:03	Load factory default	Load factory default configuration immediately after power on. • By setting this parameter to FALSE the user set configuration data (0x8000 to 0x8321) will be loaded after power on	BOOLEAN	RW	TRUE
F008:04	Save configuration data	Save all configuration setting to local non-volatile memory. • Set to TRUE in order to save the configuration data (0x8000 to 0x8321) to the memory of the ECAT-2094S	BOOLEAN	RW	FALSE
F008:05	Save error encountered	Indicates whether data has been successfully written to memory	BOOLEAN	RO	FALSE