maxon

Servo Controllers

ESCON2

Application Notes

escon.maxongroup.com

ESCON2 Servo Controllers
Application Notes
CCMC | Edition 2026-01 | DocID rel13317

maxon

TABLE OF CONTENTS

READ THIS FIRST

These instructions are for qualified technical personnel only. Before you start any work:
- *Read this manual carefully.*
- *Make sure that you understand this manual.*
- *Follow all instructions in this manual.*

1 ABOUT

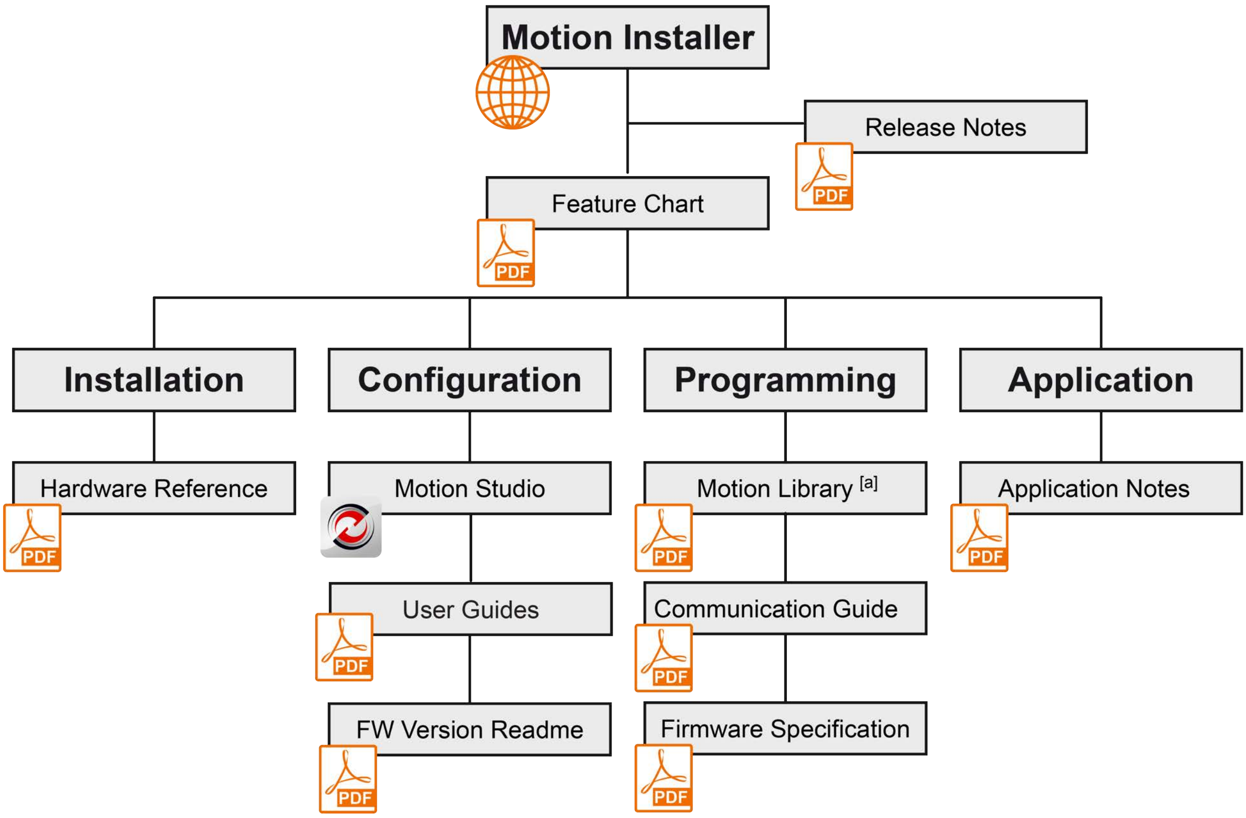
1.1 About this Document

1.1.1 Intended Purpose

The present document provides you with information to cover particular cases or scenarios that might come in handy during commissioning of your drive system. Do not use the equipment for any purpose other than the intended purpose. maxon, the manufacturer, is not liable for loss or damage that results from use outside the intended purpose.
The overview below shows the documentation hierarchy and how its parts are related. Follow the instructions:

- to avoid dangerous situations,
- to keep installation and/or commissioning time at a minimum,
- to increase reliability and service life of the described equipment.

This document is part of a documentation set. The overview below shows the documentation hierarchy and how its parts are related:



[a] including software programming examples

Figure 1-1 Documentation structure

Find the latest edition of this document, along with additional documentation and software for ESCON2 Servo Controllers, at: http://escon.maxongroup.com

1.1.2 Target Audience

This document is intended for trained and skilled personnel. It provides information on how to understand and perform the respective tasks and duties.

1.1.3 How to use

Follow these notations and codes throughout the document.

Notation	Meaning
ESCON2	stands for «ESCON2 Servo Controller»
«Abcd»	indicating a title or a name (such as of document, product, mode, etc.)
(n)	refers to an item (such as a part number, list items, etc.)
*	refers to an internal value
➔	denotes “check”, “see”, “see also”, “take note of” “refer to” or “go to”

Table 1-1 Notations used in this document

In later parts of this document, the following abbreviations and acronyms will be used:

Short	Description
CW	Clockwise
CAN	CAN Application Layer
CiA	CAN in Automation
CST	Cyclic Synchronous Torque Mode
CSV	Cyclic Synchronous Velocity Mode
ID	Identifier – the name by which a CAN device is addressed
NMT	Network Management
PDO	Process Data Object – object for data exchange between several devices
PVM	Profile Velocity Mode

Table 1-2 Abbreviations & acronyms used

1.1.4 Symbols & signs

This document uses the following symbols and signs:

Type	Symbol	Meaning
Safety alert DANGER		Indicates an **imminent hazardous situation**. If not avoided, it **will result in death or serious injury.**
WARNING		Indicates a **potential hazardous situation**. If not avoided, it **can result in death or serious injury.**
CAUTION		Indicates a **probable hazardous situation** or calls the attention to unsafe practices. If not avoided, it **may result in injury.**
Prohibited action	 (typical)	Indicates a dangerous action. Hence, **you must not!**

Type	Symbol	Meaning
Mandatory action	 (typical)	Indicates a mandatory action. Hence, **you must!**
Requirement, Note, Remark		Indicates an activity you must perform prior to continuing, or gives information on a particular point that must be observed.
Best practice		Indicates an advice or recommendation on the easiest and best way to further proceed.
Material Damage		Indicates information particular to possible damage of the equipment.

Table 1-3 Symbols and signs

1.1.5 Trademarks and brand names

All trademarks, brand names or other signs mentioned in this manual remain the property of their respective owners. They are protected by trademark, copyright, and/or other applicable laws. For easier reading, no symbols such as ® or ™ are being used with respect to the trademarks or brand names mentioned herein.
Nothing in this manual shall be construed as granting any rights with respect to the mentioned trademarks, brand named or other signs, including without limitation neither the right to use the mentioned trademarks, brand names or other signs without permission of the respective owners.

The trademarks, brand names or other signs are mentioned in this manual solely for information or identification purposes.

1.1.6 Sources for additional Information

For further details and additional information, please refer to the resources listed below:

Ref. no.	Reference
[1]	USB Implementers Forum: Universal Serial Bus Revision 2.0 Specification www.usb.org/developers/docs
[2]	CiA 102 V3 1.0: CAN physical layer for industrial applications www.can-cia.org
[3]	CiA 301 V4.2: CANopen application layer and communication profile www.can-cia.org
[4]	CiA 302 V4.1: CANopen additional application layer functions www.can-cia.org
[5]	CiA 305 V3.0: Layer Setting Services (LSS) and protocols www.can-cia.org
[6]	CiA 306 V1.4: CANopen electronic data sheet specification www.can-cia.org
[7]	CiA 402 V5.0: CANopen device profile for drives and motion control www.can-cia.org
[8]	CiA 801 V1.0.1: Automatic bit-rate detection www.can-cia.org
[9]	Bosch's CAN Specification 2.0 www.can-cia.org
[10]	Konrad Etschberger: Controller Area Network ISBN 3-446-21776-2
[11]	maxon: ESCON2 Communication Guide http://escon.maxongroup.com
[12]	maxon: ESCON2 Hardware Reference http://escon.maxongroup.com
[13]	maxon: ESCON2 Firmware Specification http://escon.maxongroup.com
[14]	IEC 61158-x-12: Industrial communication networks – Fieldbus specifications (CPF 12)
[15]	IEC 61800-7 Ed 2.0: Adjustable speed electrical power drives systems (Profile type 1)
[16]	EN 5325-4 Industrial communications subsystem based on ISO 11898 (CAN) for controller device interfaces Part4: CANopen

Table 1-4 Sources for additional information

1.1.7 Copyright

CCMC | ESCON2 Servo Controllers Application Notes | Edition 2026-01 | DocID rel13317

1.2 About the Devices

The ESCON2 line of products are small, powerful 4-quadrant PWM servo controllers. Their high power density allows flexible use for brushed DC motors and brushless EC (BLDC) motors. They support various feedback options, such as Hall sensors, incremental encoders, and absolute sensors for many drive applications.

The devices are designed to be controlled by analog and digital set values, or as a slave node in a CANopen network. You can also operate them via any USB or RS232 communication port of a Windows workstation. They have extensive analog and digital I/O functions.

Latest technology, such as field-oriented control (FOC), acceleration and velocity feed forward, in combination with highest control cycle rates allow sophisticated, ease-of-use motion control.

You can find the latest edition of this document on the Internet: ➔http://escon.maxongroup.com. This website also gives you access to related documents and software for ESCON2 servo controllers.

In addition, you can watch video tutorials in the ESCON video library. These tutorials show how to start with «Motion Studio». They also show how to set up communication interfaces, configure the controller, and give helpful tips, etc. Explore the video library on Vimeo: ➔https://vimeo.com/album/4646396

1.3 About the Safety Precautions

- Read and understand the note ➔«READ THIS FIRST»!
- Do not start any work unless you have the required skills ➔chapter “1.1.2 Target Audience” on page 1-3.
- Refer to ➔chapter “1.1.4 Symbols & signs” on page 1-4 to understand the symbols used.
- Follow all applicable health, safety, accident prevention, and environmental protection regulations for your country and work site.

DANGER

High voltage and/or electrical shock
Touching live wires can cause death or serious injuries.
- *Treat all power cables as live unless proven otherwise.*
- *Ensure neither end of the cable is connected to live power.*
- *Ensure the power source cannot be turned on while you work.*
- *Follow lock-out/tag-out procedures.*

Requirements
- *Install all devices and components according to local regulations.*
- *Electronic devices are not fail-safe. Install separate monitoring and safety equipment for each machine. If the machine has a failure, the drive system must go into a safe state and stay in this state. Possible failures include incorrect operation, failure of the control unit, failure of the cables, or other faults.*
- *Do not repair any components that maxon supplies.*

Electrostatic sensitive device (ESD)
- *Observe precautions for handling Electrostatic sensitive devices.*
- *Handle the device with care.*

••page intentionally left blank••

2 DEVICE PROGRAMMING

2.1 In brief

The ESCON2 servo controllers support several operating modes. These modes let you configure the drive system for current, torque, or velocity control. The integrated CANopen interface lets you connect multiple axes in a network. You can also control the device online with a CANopen master.

This application note explains typical command sequences for the different operating modes. The sequences are based on read and write access to the Object Dictionary. This application note applies to all ESCON2 catalog products.

For additional information, refer to the following documents:

For object descriptions, refer to the ➔«ESCON2 Firmware Specification» [13].

For command structure and communication details, refer to the ➔«ESCON2 Communication Guide» [11].

2.2 Initial setup

Before you operate the motor, you must configure the device. Set the following parameters:

- Motor parameters
- Position sensor parameters
- Controller parameters

Most parameter values are available in the motor and encoder data sheets. Also consider the limits of the mechanics and the electrical system. These limits include maximum speed, acceleration, and deceleration.

For a detailed description of the Object Dictionary entries and their dependencies, refer to the ➔«ESCON2 Firmware Specification» [13].

Note
- *Refer to chapter «Operating modes» in the ➔«ESCON2 Firmware Specification» [13].*
- *This chapter describes the relevant objects for each operating mode.*

The parameter units in this document use the default system configuration. Depending on the configuration of the following objects, the units can change:

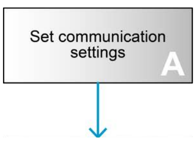
- Object **0x60A9** (Position unit)
- Object **0x60AA** (Velocity unit)

All unit values in this document refer to the default configuration.

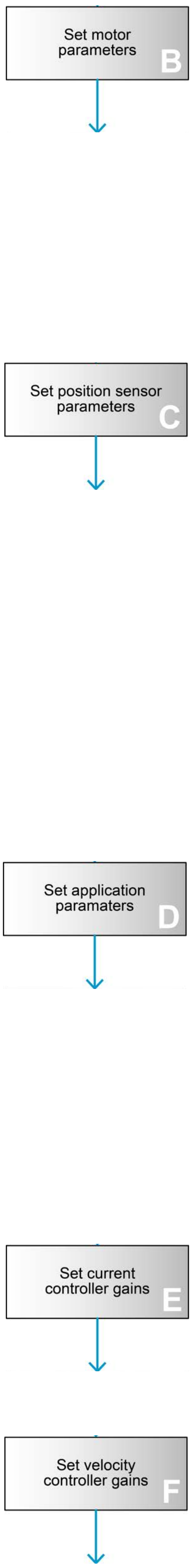
2.2.1 Device programming | First steps

The below table lists the parameters you must set before you operate the drive system. The parameters are grouped by function. Set the parameters in the order shown.

For detailed descriptions of the parameters, refer to the ➔«ESCON2 Firmware Specification» [13].



#	Object name	Object	Description [unit]
A	Node-ID	0x2000-00	Application-specific: typically configured by DIP switches
	CAN bit rate	0x2001-00	Application-specific: 0 = 1 Mbit/s
	Serial communication interface bit rate	0x2002-00	Application-specific: 5 = 115.2 kBit/s SCI is not supported by all ESCON2 types.



#	Object name	Object	Description [unit]
B	Motor type	0x6402-00	Motor-specific: 1= DC or 10 = EC / BLDC
	Nominal current	0x3001-01	Motor-specific [mA]
	Number of pole pairs	0x3001-03	Motor-specific [1]
	Thermal time constant winding	0x3001-04	Motor-specific [0.1 s]
	Torque constant	0x3001-05	Motor-specific [µNm/A]
	Max motor speed	0x6080-00	Drive train-specific [rpm]: depending on motor or mechanical limits
C	Axis configuration	0x3000-xx	Sensor-specific and system-specific
	Digital incremental encoder S2	0x3010-01 0x3010-02	Incremental encoder-specific: Check encoder data sheet! Number of pulses [pulses/revolution] = [cpt] Encoder type [0x0000] (= maxon without index, time measurement)
	SSI absolute encoder S2	0x3012-xx	SSI encoder-specific Check encoder data sheet
	BiSS C unidirectional absolute encoder S2	0x3013-xx	BiSS encoder-specific: Check encoder data sheet!
	Digital Hall sensor S1	0x301A-01	Hall-specific: Hall type [0x00] (=maxon, time measurement)
D	Output current limit	0x3001-02	Derivative & application-specific [mA] typ. 2 - 3 times of "Nominal current"
	Max acceleration	0x60C5-00	Application-specific [max. 4'294'967'295 rpm/s] depending on electrical and mechanical limits
	Max profile velocity	0x607F-00	Application-specific [120'000 rpm] depending on motor or mechanical limits
	Max temperature power stage	0x3201-03	Derivative & aplication-specific [0.1 °C] depending on ESCON2 product type
	Power supply undervoltage limit	0x2201-01	Derivative & application-specific [mV]
	Power supply overvoltage limit	0x2201-02	Derivative & application-specific [mV]
	Profile acceleration	0x6083-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
	Profile deceleration	0x6084-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
	Quick stop deceleration	0x6085-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
E	Current control parameter set: • Current controller PI gains	0x30A0-xx	Motor-specific: Determine optimal parameter using "Regulation Tuning" in «Motion Studio».
F	Velocity control parameter set: • Velocity controller PI gains • Velocity controller FF gains • Velocity controller filter cut-off frequency	0x30A2-xx	Motor-specific and load-specific: Determine optimal parameter using "Regulation Tuning" in «Motion Studio».

Table 2-5 Device programming | First steps

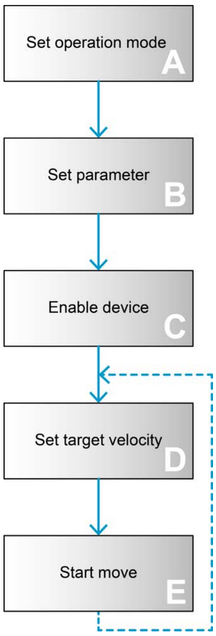
2.3 Profile Velocity Mode (PVM)

Note
For details about Controlword (0x6040), Statusword (0x6041), and related objects, refer to the ➔«ESCON2 Firmware Specification» [13].

START MOVEMENT

In Profile Velocity Mode, the axis moves at the commanded target velocity. The controller uses the configured acceleration and deceleration profile parameters.

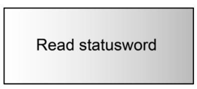
Set the parameters in the order shown.



#	Object name	Object	Description [unit]
A	Modes of operation	0x6060-00	0x03 = (Profile Velocity Mode)
B	Max profile velocity	0x607F-00	Motor- and application-specific [rpm]; depending on motor, mechanics, application
	Profile acceleration	0x6083-00	Application-specific [rpm/s]; default: 10’000 depending on electrical and mechanical limits
	Profile deceleration	0x6084-00	Application-specific [rpm/s]; default: 10’000 depending on electrical and mechanical limits
	Quick stop deceleration	0x6085-00	Application-specific [rpm/s]; default: 10’000
	Motion profile type	0x6086-00	Application-specific [0]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Target velocity	0x60FF-00	Desired velocity [rpm]
E	Controlword	0x6040-00	0x000F

Table 2-6 Device programming | Profile Velocity Mode (Start)

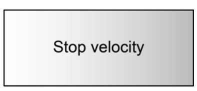
READ STATUS



Object name	Object	Description
Statusword (Target velocity reached)	0x6041-00	Target velocity is reached if bit 10 is set, refer to ➔*«ESCON2 Firmware Specification» [13].*

Table 2-7 Device programming | Profile Velocity Mode (Status)

STOP MOVEMENT



Object name	Object	Value
Controlword (Halt profile velocity mode)	0x6040-00	0x010F
	or	
Controlword (Quick stop)	0x6040-00	0x000B

Table 2-8 Device programming | Profile Velocity Mode (Stop)

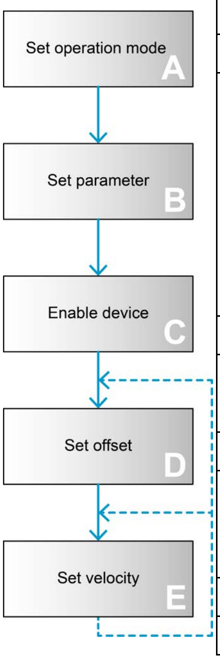
2.4 Cyclic Synchronous Velocity Mode (CSV)

Note
For details about Controlword (0x6040) and other related objects, refer to the ➔«ESCON2 Firmware Specification» [13].

START MOVEMENT

In Cyclic Synchronous Velocity Mode, the axis follows velocity set values that the master sends cyclically. The ESCON2 performs a linear interpolation between two velocity set values. The interpolation is based on the configured Interpolation Time Period.

Set the parameters in the order shown.



#	Object name	Object	Description [unit]
A	Modes of operation	0x6060-00	0x09 = (Cyclic Synchronous Velocity Mode)
B	Max motor speed	0x6080-00	Application-specific; depending on motor or mechanical limits [rpm]
	Profile deceleration [a]	0x6084-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
	Quick-stop deceleration [a]	0x6085-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
	Interpolation time period [b]	0x60C2-xx	Master's SYNC period (Cycle ticks) [ms]
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Velocity offset	0x60B1-00	Velocity offset [rpm]
	Torque offset	0x60B2-00	Torque offset [0.1 % of "Motor rated torque"; 0x6076]
E	Target velocity	0x60FF-00	Desired velocity [rpm]

[a] The deceleration values are used for stopping only. In this mode, the controller does not use these parameters for normal motion.

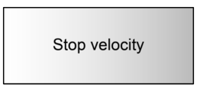
[b] Set the **Interpolation time period** to match the master PDO cycle time.

If the value is **0**, the ESCON2 applies the new set value in the next control cycle. The controller then holds this value until it receives the next set value from the master. This can cause irregular and noisy motion if the master sends set values not at short cycle times.

If you set the **Interpolation time period** correctly, the ESCON2 interpolates between set values. This results in smooth motion and stable control.

Table 2-9 Device programming | Cyclic Synchronous Velocity Mode (Start)

STOP MOVEMENT



Object name	Object	Value
Target velocity	0x60FF-00	0x0000
	or	
Controlword (Quick stop)	0x6040-00	0x000B

Table 2-10 Device programming | Cyclic Synchronous Velocity Mode (Stop)

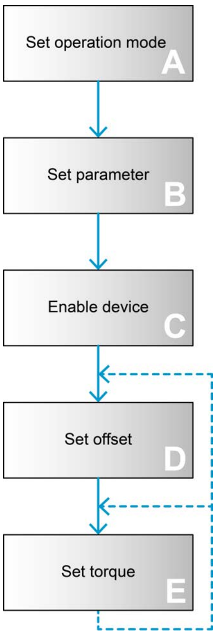
2.5 Cyclic Synchronous Torque Mode (CST)

Note
For details about Controlword (0x6040) and other related objects, refer to the ➔«ESCON2 Firmware Specification» [13].

START MOVEMENT

In Cyclic Synchronous Torque Mode, the controller applies the commanded torque to the motor. The motor torque equals the motor current multiplied by the torque constant.

Set the parameters in the order shown.



#	Object name	Object	Description [unit]
A	Modes of operation	0x6060-00	0x0A (Cyclic Synchronous Torque Mode)
B	Max motor speed	0x6080-00	Application-specific [rpm]; depending on motor or mechanical limits
	Profile deceleration [a]	0x6084-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
	Quick-stop deceleration [a]	0x6085-00	Application-specific [10'000 rpm/s]; depending on electrical and mechanical limits
C	Controlword (Shutdown)	0x6040-00	0x0006
	Controlword (Switch on & Enable)	0x6040-00	0x000F
D	Torque offset	0x60B2-00	Torque offset [0.1 % of "Motor rated torque"; 0x6076]
E	Target torque	0x6071-00	Desired torque [0.1 % of "Motor rated torque"; 0x6076]

[a] The deceleration values are used for stopping only. In this mode, the controller does not use these parameters for normal motion.

Table 2-11 Device programming | Cyclic Synchronous Torque Mode (Start)

STOP MOVEMENT

Object name	Object	Value
Target torque	0x6071-00	0x0000
	or	
Controlword (Quick stop)	0x6040-00	0x000B

Table 2-12 Device programming | Cyclic Synchronous Torque Mode (Stop)

2.6 State Machine

The state machine defines the operating states of the ESCON2. You control the state transitions with the Controlword.

For details about state transitions and bit definitions, refer to the ➔«ESCON2 Firmware Specification» [13].

CLEAR FAULT

If the device is in the Fault state, you must reset the fault before you continue.

Clear fault

Object name	Object	Value
Controlword (Fault reset)	0x6040-00	0x0000; 0x0080

Table 2-13 Device programming | State machine (clear fault)

SEND NMT SERVICE

Send NMT service

Object name	Command	
Node ID (Unique Node ID or “0” (zero) for all nodes)	0x01 0x02 0x80 0x81 0x82	Start remote node Stop remote node Enter pre-operational Reset node Reset communication

Table 2-14 Device programming | State machine (send NMT service)

2.7 Motion Information

GET MOVEMENT STATE

Read statusword

Object name	Object	Description
Read statusword	0x6041-00	The bits are partly depending on the operating mode (for details refer to the ➔«ESCON2 Firmware Specification» [13])

Table 2-15 Device programming | Motion information (Get movement state)

READ VELOCITY

Read velocity

Object name	Object	Description [unit]
Read velocity actual value	0x606C-00	Velocity actual value [rpm]
Read velocity actual value averaged	0x30D3-01	Velocity actual value averaged [rpm]

Table 2-16 Device programming | Motion information (Read velocity)

READ TORQUE

Read torque

Object name	Object	Description [unit]
Read torque actual value	0x6077-00	Torque actual value [0.1 % of “Motor rated torque”; 0x6076]
Read torque actual value averaged	0x30D2-01	Torque actual value averaged [0.1 % of “Motor rated torque”; 0x6076]

Table 2-17 Device programming | Motion information (Read torque)

2.8 Utilities

STORE ALL PARAMETERS

Saves all parameters persistently.

Object name	Object	Value
Store all parameters	0x1010-01	0x65766173 (i.e. “save”)

Table 2-18 Device programming | Utilities (Store all parameters)

RESTORE ALL DEFAULT PARAMETERS

Restores all parameters to factory settings.

Object name	Object	Value
Restore all default parameters	0x1011-01	0x64616F6C = (i.e. “load”)

Table 2-19 Device programming | Utilities (Restore all default parameters)

••page intentionally left blank••

3 ENCODER REQUIRED OR NOT REQUIRED

3.1 In brief

This application note explains when an encoder is required for ESCON2 servo controllers. It compares control performance with Hall sensor feedback and with encoder feedback. The information helps you decide if an encoder is necessary for your application.

The ESCON2 determines the rotor position with advanced algorithms based on Hall sensor signals. In many applications, Hall sensor feedback is sufficient for current control and speed control. An additional encoder is often not required for BLDC motors.

The ESCON2 supports Field-Oriented Control (FOC) with Hall sensor feedback only. The controller provides stable and precise speed control over the full motor speed range, including low speeds.

This application note:

- explains how ESCON2 algorithms improve rotor position estimation for FOC and speed control based on Hall sensor edges only.
- explains when the use of an encoder can be advantageous and is recommended.

3.2 Perfect commutation and smooth operation without an encoder

The ESCON2 provides high commutation accuracy and high control quality.

The ESCON2 uses Field-Oriented Control (FOC) to generate sinusoidal motor currents. This applies over the full speed range and works with Hall sensors or with encoders.

The ESCON2 uses an advanced method based on Hall sensor signals only. This method enables precise commutation and stable speed control without an encoder.

3.2.1 Hall sensor limitations and compensation

In BLDC motors without encoders, Hall sensors provide low position resolution per rotor revolution. The Hall sensor switching edges are not evenly spaced due to manufacturing tolerances.

In conventional servo controllers, this causes inaccurate commutation. It also causes deviations between measured speed and actual motor speed.

The ESCON2 analyzes the Hall sensor switching edges during the first commanded movement after power-on. The controller compensates the detected deviations and generates a virtual high-resolution position signal.

The ESCON2 uses this virtual signal for commutation and speed control over the full speed range. The controller interpolates between Hall sensor edges to support FOC commutation.

As a result, the ESCON2 provides stable speed control and smooth motor operation. An encoder is often not required.

3.2.2 Speed stability comparison

The Figure below compares the speed response of a conventional servo controller and an ESCON2.

For both controllers, a target velocity of **1000 rpm** is commanded. A maxon **EC-i 40** motor (P/N 449470) with Hall sensor feedback operates in speed control mode.

A high-resolution encoder with **16384 counts per revolution** measures the motor speed. The encoder is not used for control or commutation.

The ESCON2 shows significantly lower speed fluctuation than the conventional controller.

This comparison shows the speed stability of a conventional servo controller and an ESCON2 during speed-controlled operation of a BLDC motor with Hall sensor feedback.

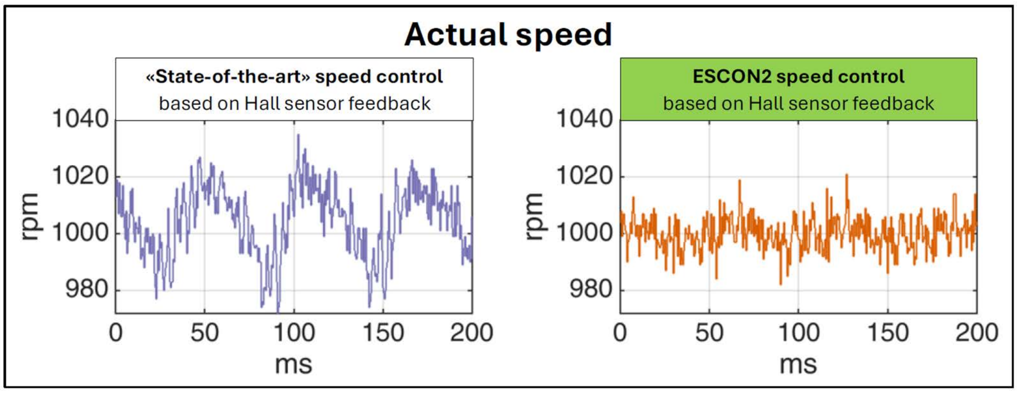


Figure 3-2 Speed stability with Hall sensor feedback

3.2.3 Motor current comparison

The Figure below compares the motor current during speed-controlled operation.

Both servo controllers operate under identical conditions. The maxon **EC-i 40** motor (P/N 449470) uses Hall sensor feedback without an encoder.

Under identical load conditions, the average motor current is approximately **100 mA**.

The conventional servo controller shows higher current ripple. The ESCON2 shows smooth motor current without cyclic oscillations.

This comparison shows the motor current of a conventional servo controller and an ESCON2 during speed-controlled operation of a BLDC motor with Hall sensor feedback.

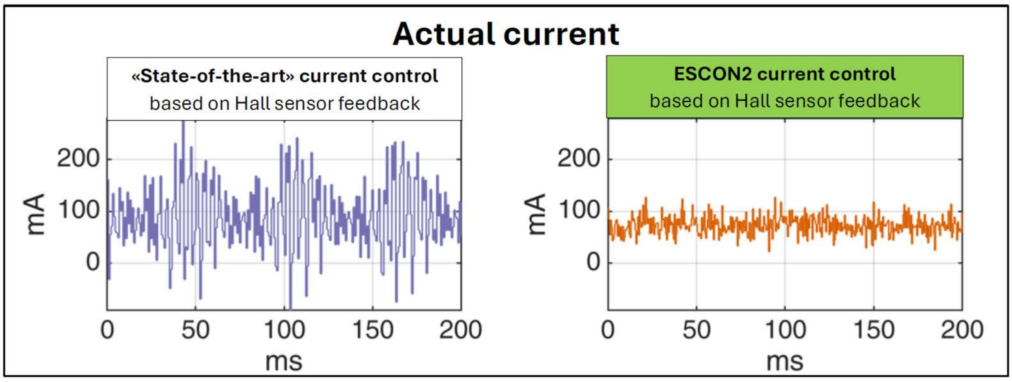


Figure 3-3 Motor current stability with Hall sensor feedback

3.3 When an encoder should be used

In some applications, an encoder improves control performance.

Hall-sensor-based control requires a short adjustment phase after each power-on. During this phase, the motor must reach a sufficient mechanical speed at least once.

At very low speeds, control accuracy can decrease. This can occur when:

- The set value changes dynamically, or
- The load varies strongly.

In these cases, use an encoder.

In many applications, the motor operates at different speed levels. The speed can increase to several hundred rpm and decrease to low rpm during acceleration or deceleration. The ESCON2 can control these speed changes stably with Hall sensor feedback only.

3.3.1 Electrical speed and mechanical speed

Two factors are relevant to decide if an encoder is necessary:

- The motor speed
- The number of motor pole pairs

You must consider the electrical speed and the mechanical shaft speed. The electrical speed equals the mechanical shaft speed multiplied by the number of pole pairs.

Hall sensor calibration

The ESCON2 performs an automatic Hall sensor calibration if the mechanical shaft speed exceeds 600 rpm at least once. This applies to single-pole-pair and multi-pole-pair motors. If the motor exceeds 600 rpm:

- Stable control is possible at lower speeds afterwards
- This applies only if there are no fast dynamic movements
- This applies only if the load does not vary strongly

Multi-pole-pair motors

For multi-pole-pair motors, sufficient control performance is possible at electrical speeds above 1000 rpm. This corresponds to a mechanical shaft speed of: 1000 rpm divided by the number of pole pairs.

3.3.2 Encoder usage guideline

Single-pole-pair motor

Use an encoder if the mechanical shaft speed never exceeds approximately 600 rpm.

Multi-pole-pair motor

Use an encoder if the main mechcanical motor shaft operating speed is below:
1000 rpm divided by the number of pole pairs.

Examples:

- 2 pole pairs: below 500 rpm
- 4 pole pairs: below 250 rpm

General guidelines (independent of pole pair number)

Use an encoder if:

- The application requires fast dynamic response at low speed, or
- The load varies strongly, or
- The application requires position feedback or position monitoring

Encoder resolution

If an encoder is required, select a minimum resolution of 500 counts per turn (cpt).

Recommended encoder types:

- Incremental encoder with 1024 to 4096 cpt
- Absolute encoder with a single-turn resolution of 12 to 16 bits

4 ADJUST THE COMMUTATION OFFSET OF AN ABSOLUTE ENCODER

4.1 In brief

The ESCON2 supports SSI and BiSS-C absolute encoders for motor commutation without Hall sensor feedback.

If you use a maxon motor with a maxon SSI or BiSS-C absolute encoder, the encoder zero position is factory aligned with the rotor position. In this case, the commutation offset value is **0**, and no adjustment is required.

If you use third-party motors or encoders, the encoder zero position may not match the rotor position. In this case, you must adjust the commutation offset during commissioning.

This application note explains the required adjustment steps.

Important

- *This adjustment applies only to **SSI or BiSS-C absolute encoders** with **ESCON2 servo controllers**.*
- *Some absolute encoders can store an internal **offset** or **addition** value. Make sure that the encoder does **not** store such a value.*
- *Follow the instructions in the specified order.*

4.2 Preconditions

Motion Studio

1) Install Motion Studio, version 1.2 or later, on your PC. If the software is not installed, download the latest version from: ➔https://escon.maxongroup.com
2) Connect **Motion Studio** to the ESCON2 with **USB, SCI (RS232),** or **CANopen**.

ESCON2 Servo Controller

3) Install ESCON2 Firmware, version 0x120 or later, on the ESCON2. If the firmware is not available, download the latest version from: ➔https://escon.maxongroup.com
4) Supply the ESCON2 with the specified power supply voltage.

Motor and SSI or BiSS-C absolute encoder

5) Connect the motor and the absolute encoder correctly to the ESCON2.
6) If the parameter settings of your ESCON2 are unknown, execute **«Restore all default parameters»** before you continue. Refer to ➔Figure 4-4.

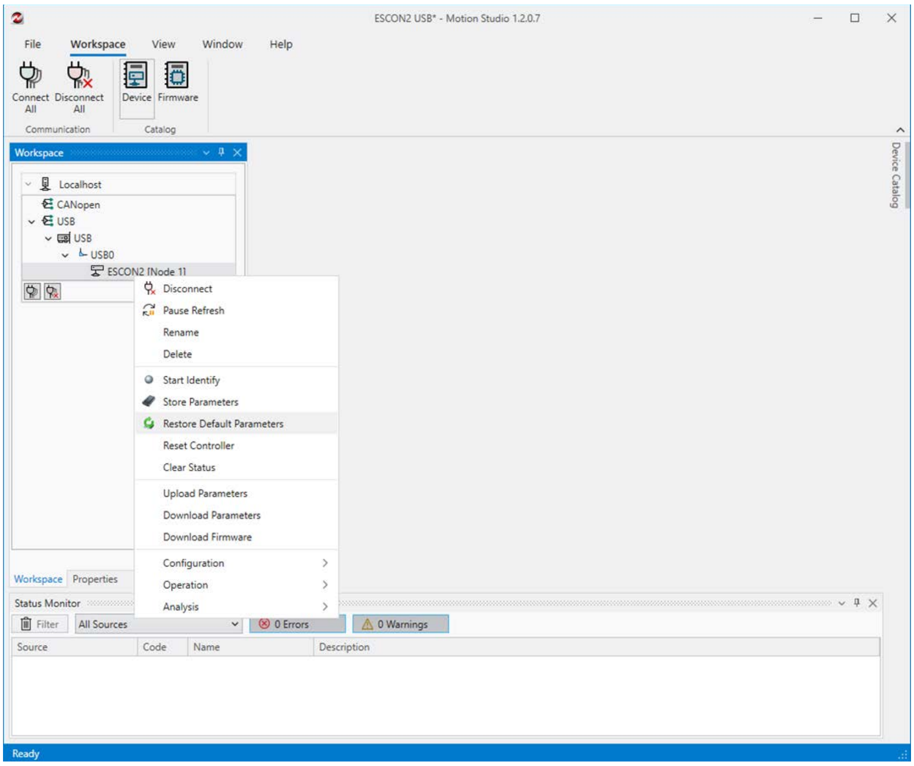


Figure 4-4 Adjustment of SSI & BiSS-C commutation offset value | Restore all default parameters

7) Configure the motor and the absolute encoder with «Motion Studio»: Go to Startup > Drive System > Motor or Startup > Drive System > Sensors. Refer to ➔Figure 4-5.

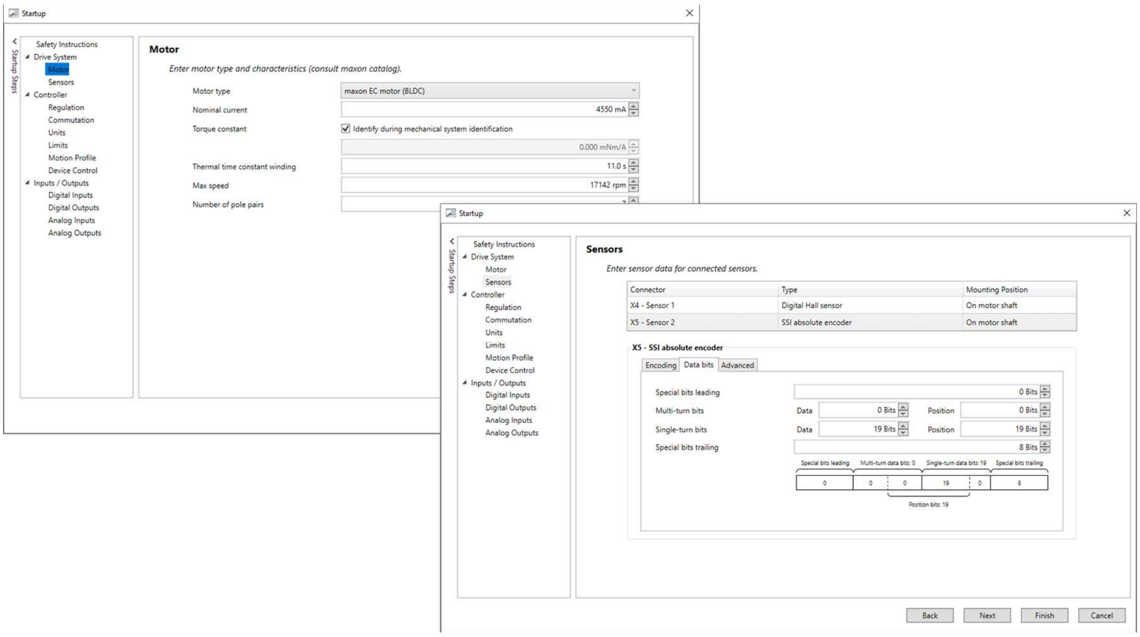


Figure 4-5 Adjustment of SSI & BiSS-C commutation offset value | Set motor and sensor data

8) Configure the SSI or BiSS-C absolute encoder as follows:

- Set the encoder as **Main sensor in Controller > Regulation**.
- Set the encoder as commutation sensor in **Controller > Commutation**.

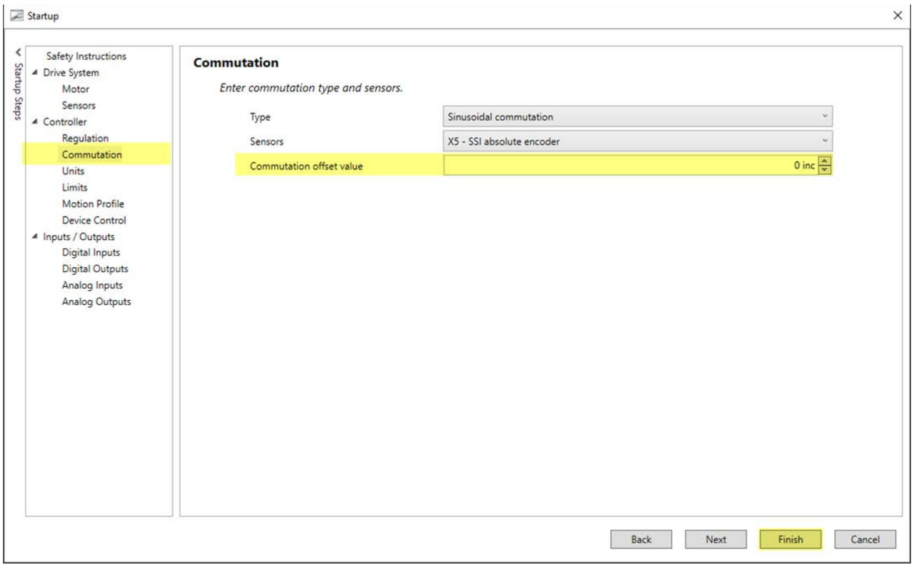


Figure 4-6 Adjustment of SSI & BiSS-C commutation offset value | Set commutation sensor

Incorrect configuration can damage the motor.
Incorrect configuration settings can permanently damage the motor during the alignment process. Before the next step, configure the motor data for the exact motor type. This includes the ***Nominal current****.*

9) Make sure that the motor can rotate freely. If a brake is installed, make sure that the brake does not engage. Disconnect all mechanical loads from the motor shaft.

10) Set the **User level** to **Expert** in **Properties**.

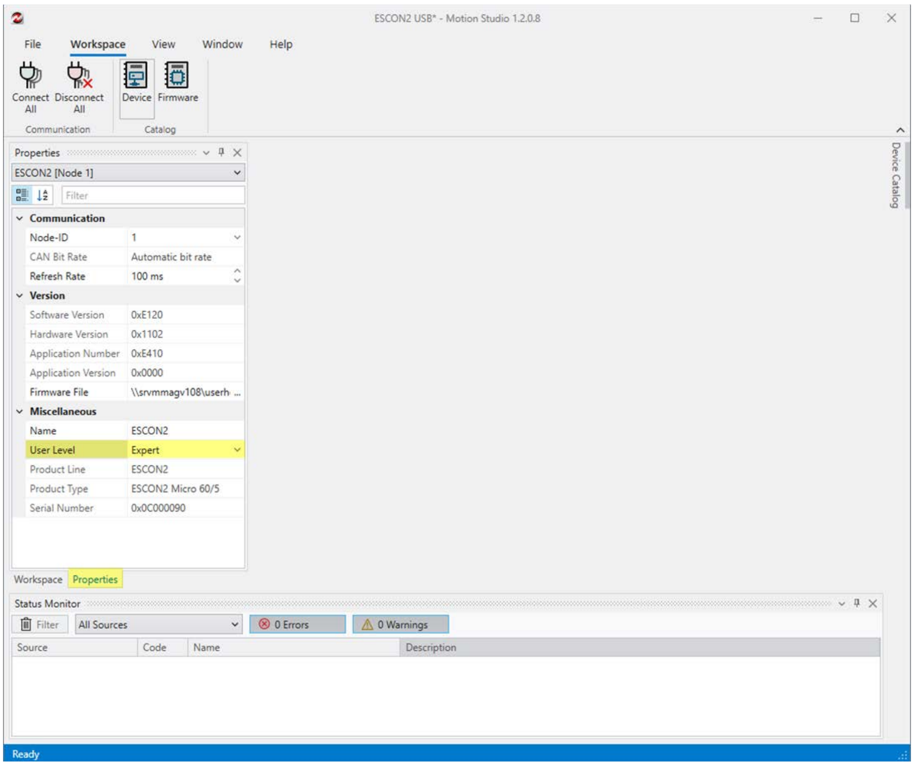


Figure 4-7 Adjustment of SSI & BiSS-C commutation offset value | Switch user level to Expert

11) Verify the direction of the absolute encoder:

a) Open the Object Dictionary tool.
Right-click the ESCON2 node in **Workspace**, then select
Analysis > Object Dictionary.
Refer to ➔Figure 4-8.

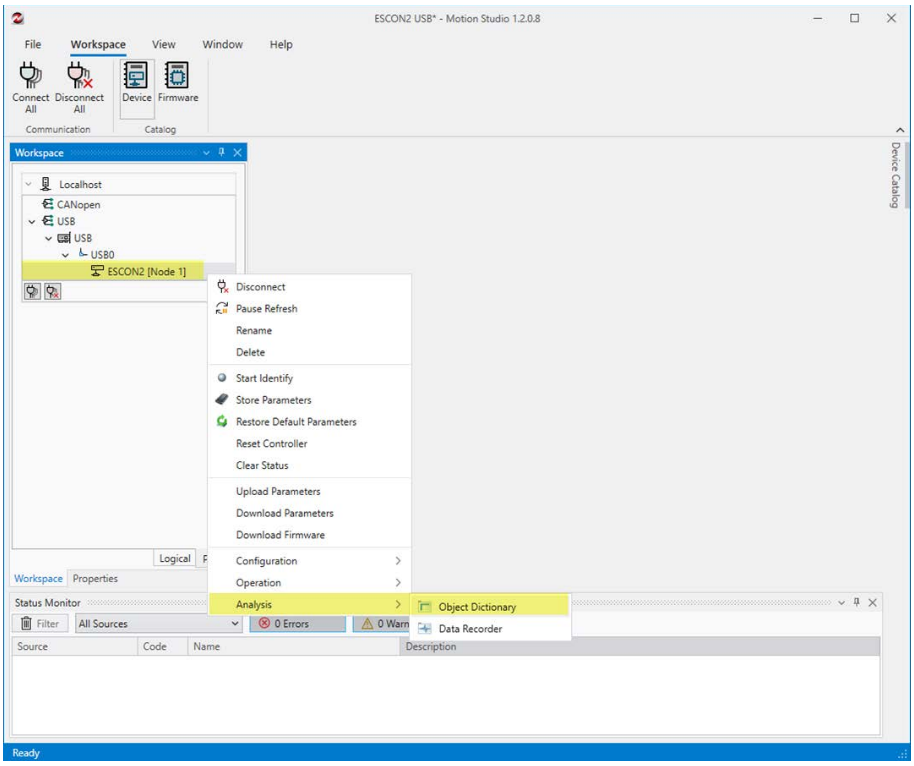


Figure 4-8 Adjustment of SSI & BiSS-C commutation offset value | Open Object Dictionary

b) Turn the motor shaft by hand **clockwise (CW)** as seen from the motor mounting flange. Refer to ➔Figure 4-9.

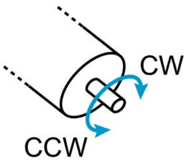


Figure 4-9 Adjustment of SSI & BiSS-C commutation offset value | Illustration of directions

c) Observe the position raw value in the **Object Dictionary**:
- SSI absolute encoder position raw value (0x3012-0B), or
- BiSS-C absolute encoder position raw value (0x3013-0B).
Check if the value increases or decreases while you turn the shaft CW. Refer to ➔Figure 4-9.

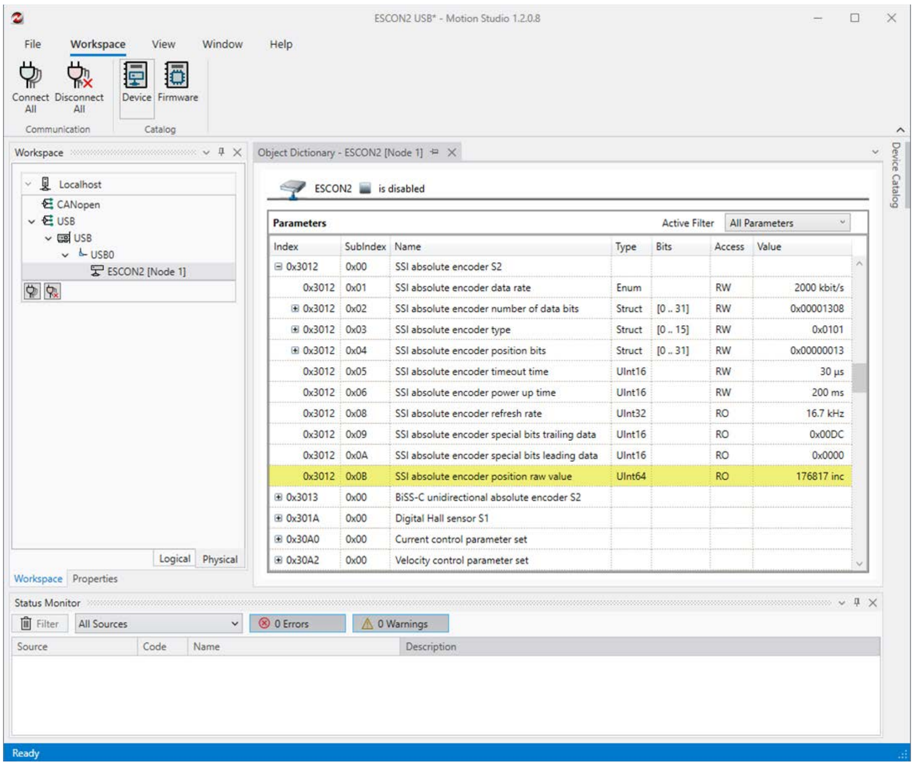


Figure 4-10 Adjustment of SSI & BiSS-C commutation offset value | Read SSI position raw value

d) Open the Startup wizard and select Drive System > Sensors.
If the position raw value increases, select maxon for Direction.
If the position raw value decreases, select Inverted.
Click Finish to apply the setting and verify the result. Refer to ➔Figure 4-11

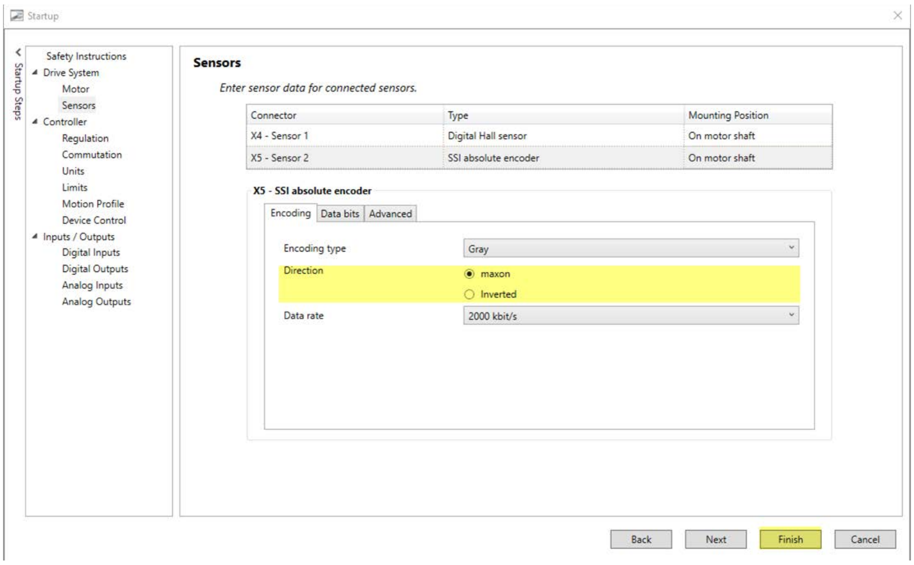


Figure 4-11 Adjustment of SSI & BiSS-C commutation offset value | Toggle direction of rotation

No indication or abnormal behavior

*If the **SSI absolute encoder position raw value** or the **BiSS-C absolute encoder position raw value** does not change, check the wiring of the absolute encoder. If the behavior is abnormal, verify the encoder configuration settings.*

4.3 Determination of the «Commutation offset value»

To determine the «Commutation offset value», consider the following criteria:

General parameter

- Number of pole pairs (0x3001-03)

For SSI absolute encoders

- SSI encoder single-turn bits (configuration object 0x3012-02; bits 8…15)
- SSI encoder direction (configuration object 0x3012-03; bit 4)
- SSI encoder position raw value (object 0x3012-0B)

For BiSS-C absolute encoder

- BiSS-C encoder single-turn bits (configuration object 0x3013-02, bits 0...7)
- BiSS-C encoder direction (configuration object 0x3013-03, bit 4)
- BiSS-C encoder position raw value (object 0x3013-0B)

Step 1 – Tune the current control parameters

- If the current control parameters are not tuned, open the Regulation Tuning wizard (Refer to ➔Figure 4-12).

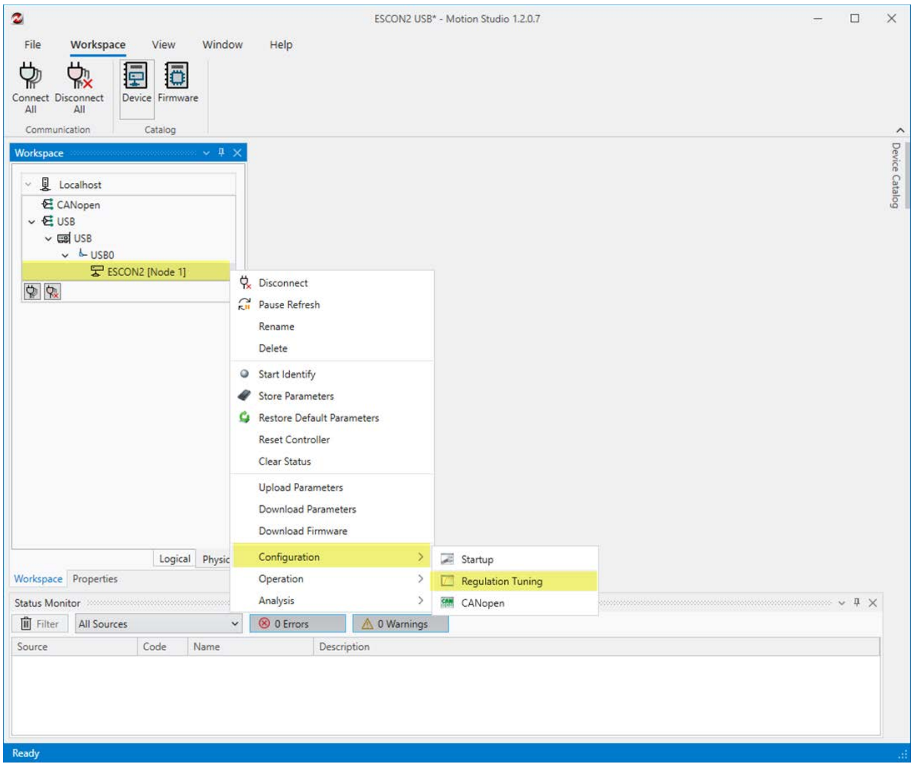


Figure 4-12 Adjustment of SSI & BiSS-C commutation offset value | Regulation Tuning

- Select Electrical Loop > Current and click Auto tune (Refer to ➔Figure 4-13).

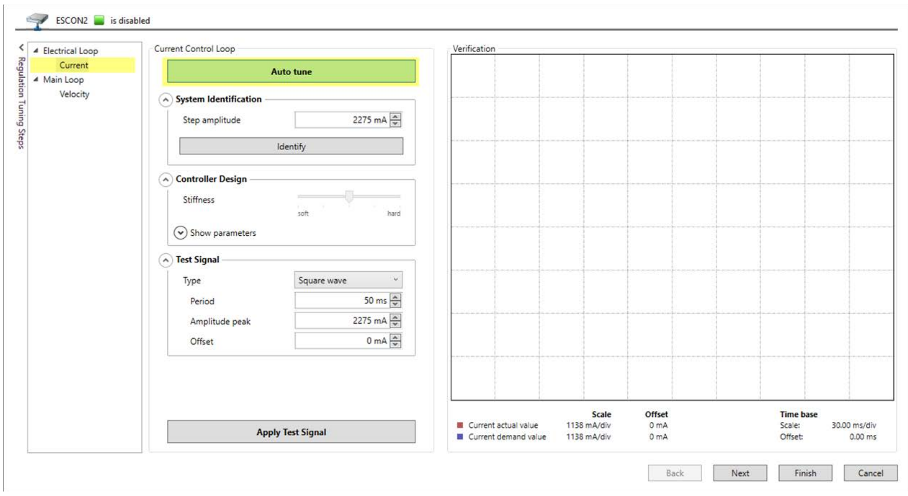


Figure 4-13 Adjustment of SSI & BiSS-C commutation offset value | Auto tuning for current control

Step 2 – Configure the motor type

- Open the **Startup** wizard and select **Drive System > Motor.**
  Set **Motor type** to **maxon DC motor.**
  Click **Finish** to close the wizard.

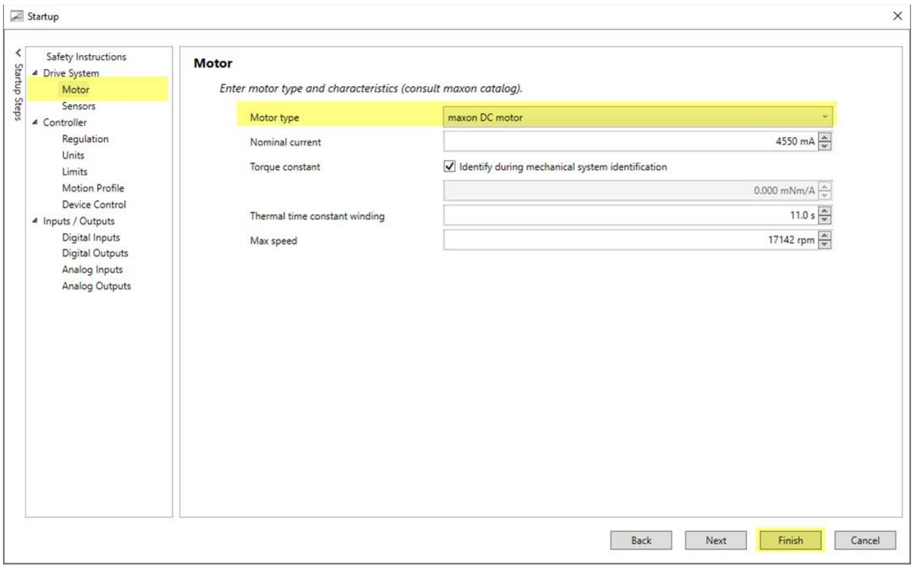


Figure 4-14 Adjustment of SSI & BiSS-C commutation offset value | Select motor type maxon DC motor

Step 3 – Activate Cyclic Synchronous Torque Mode

- Open the Cyclic Sync Torque Mode tool (Refer to ➔Figure 4-15).
  Click Activate Cyclic Synchronous Torque Mode, then click Enable (Refer to ➔Figure 4-16).

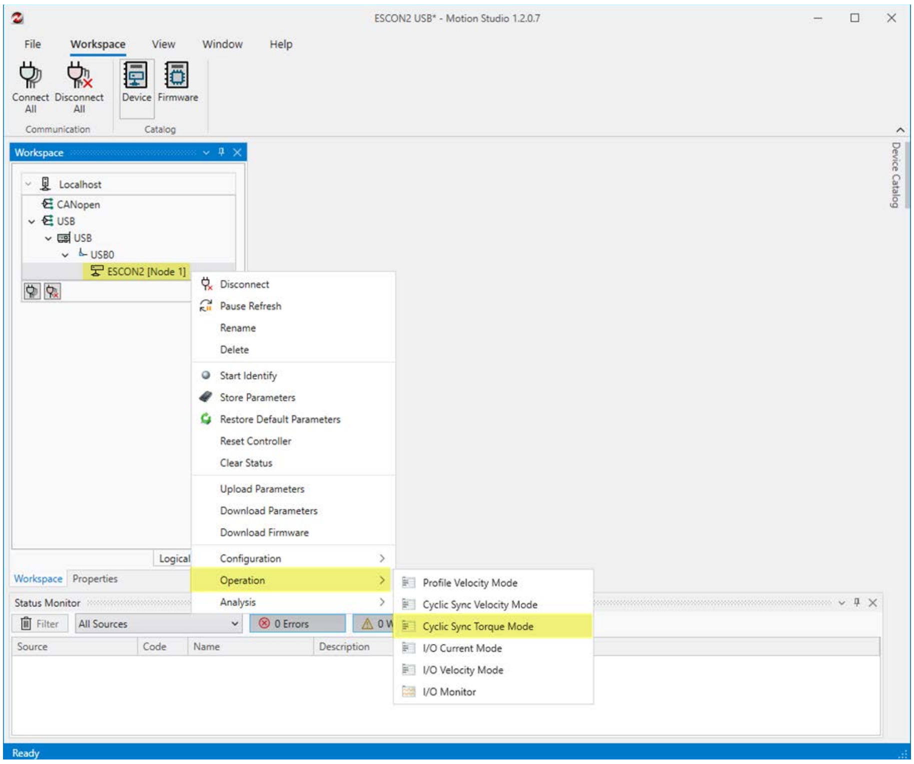


Figure 4-15 Adjustment of SSI & BiSS-C commutation offset value | Open Cyclic Sync Torque Mode (CST)

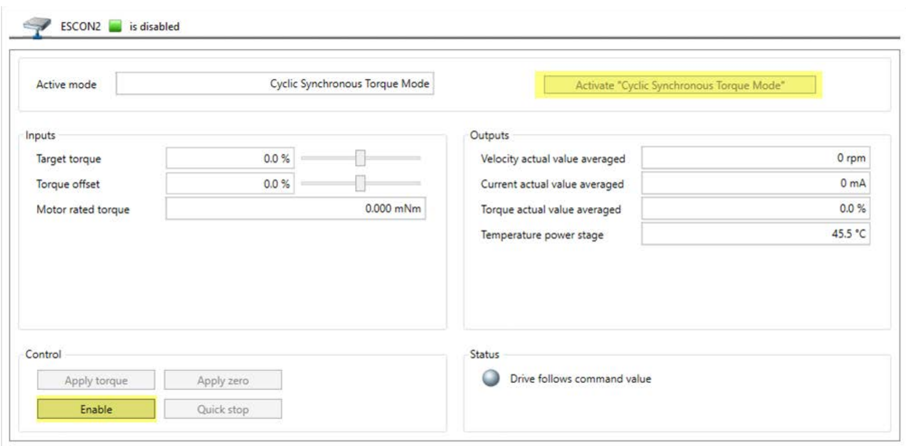


Figure 4-16 Adjustment of SSI & BiSS-C commutation offset value | Activate CST

Step 4 – Align the motor

- Set **Target torque** to **30.0%** and click **Apply torque**.
  The motor aligns but does not rotate continuously.
  Verify that **Current actual value averaged** is approximately **30%** of the configured **Nominal current** for the motor. If accessible, gently move the motor shaft by hand and release it.

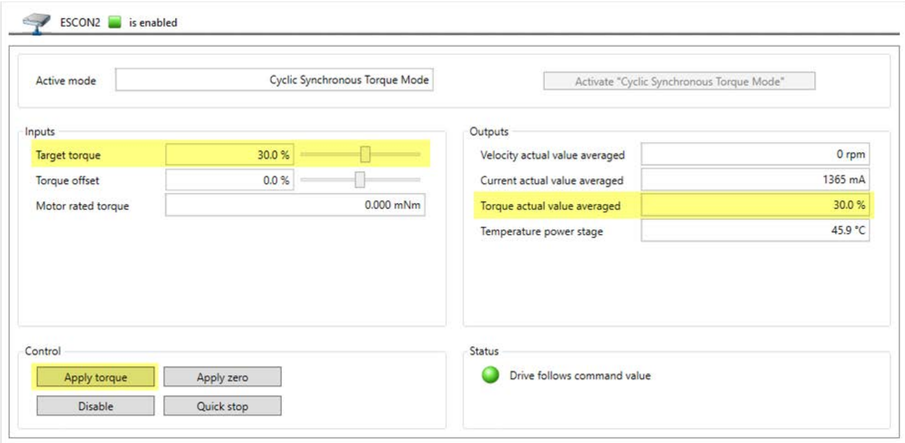


Figure 4-17 Adjustment of SSI & BiSS-C commutation offset value | Apply target torque

Best Practice
*If the motor has significant cogging torque or high static friction, you may need more than **30% Target torque**. For example, set **80% Target torque** to improve the alignment quality.*

Step 5 – Read the encoder position raw value

- Read the position raw value from the **Object Dictionary** (refer to ➔Figure 4-10) and record it.
  - For SSI: **0x3012-0B**, *SSI absolute encoder position raw value*
  - For BiSS-C: **0x3013-0B**, *BiSS-C absolute encoder position raw value*

Step 6 – Disable the motor

- Click **Disable** in the **Cyclic Sync Torque Mode** tool.

Step 7 – Calculate the commutation offset

- Calculate the commutation offset value.
  For a detailed example, refer to ➔“Calculation example «step-by-step instruction»” on page 4-34.

To calculate the commutation offset value, use the following parameters:

- **Number of pole pairs** (Object 0x3001-03)
- **Single-turn bits** of the absolute encoder
  - SSI: Object 0x3012-02, bit 8..15
  - BiSS-C: Object 0x3013-02, bit 0..7
- **Absolute encoder position raw value**
  - SSI: Object 0x3012-0B
  - BiSS-C: Object 0x3013-0B
- **Sensor direction**
  - SSI: Object 0x3012-03, bit 4
  - BiSS-C: Object 0x3013-03, bit 4

The result is the **commutation offset value**, which you must write to **Object 0x3241-03**.

4.3.1 Step-by-step calculation

1) **Calculate the mechanical resolution**
The mechanical resolution corresponds to one full mechanical revolution:

$$ResMech = 2^{\text{SingleTurnBits}}$$

2) **Calculate the electrical resolution**
Divide the mechanical resolution by the number of pole pairs:

$$ResElec = \frac{ResMech}{PolePairs}$$

3) **Calculate the single-turn encoder position**
Apply a modulo operation to wrap the raw encoder value into one mechanical turn:

$$PosSingleTurn = modulo(PosRaw, ResMech)$$

4) **Determine the electrical reference angle**

- If the sensor direction is inverted:

$$Ref = 0$$

- Otherwise:

$$Ref = \frac{5}{6} x ResElec$$

5) **Calculate the preliminary offset**
Subtract the encoder position from the reference and wrap it into one electrical revolution:

$$Offset = modulo(Ref - PosSingleTurn, ResElec)$$

6) **Correct negative values**
If the result is negative, add one electrical revolution:

$$IfOffset < 0\text{: } Offset = Offset + ResElec$$

7) **Round the result**
The final commutation offset is the rounded value of the calculated result.

Step 8 – Restore the correct motor type

- Open the **Startup** wizard and select **Drive System > Motor**.
  Set **Motor type** to **maxon EC motor (BLDC)** (refer to ➔Figure 4-18).

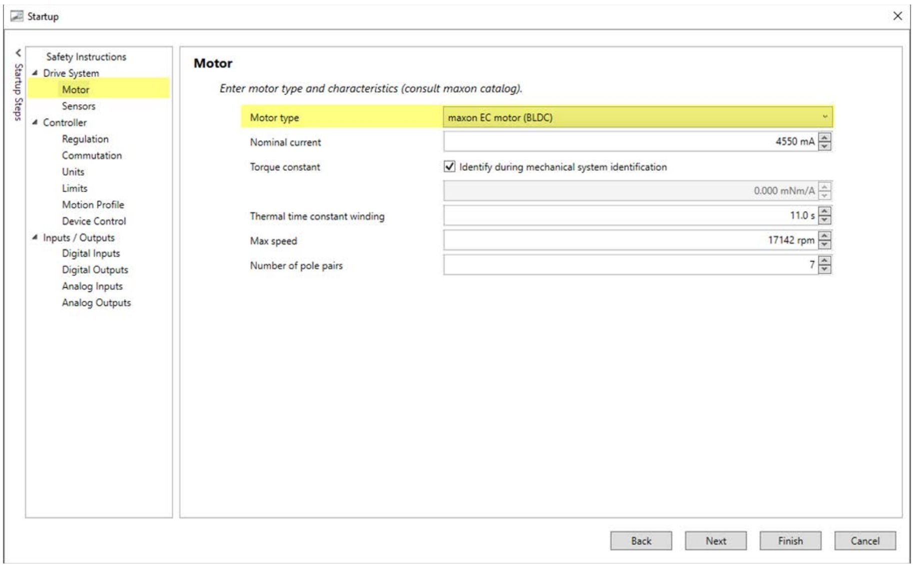


Figure 4-18 Adjustment of SSI & BiSS-C commutation offset value | Select motor type maxon EC motor

Step 9 – Enter the commutation offset

- Open the **Commutation** section and enter the calculated value in **Commutation offset value**.

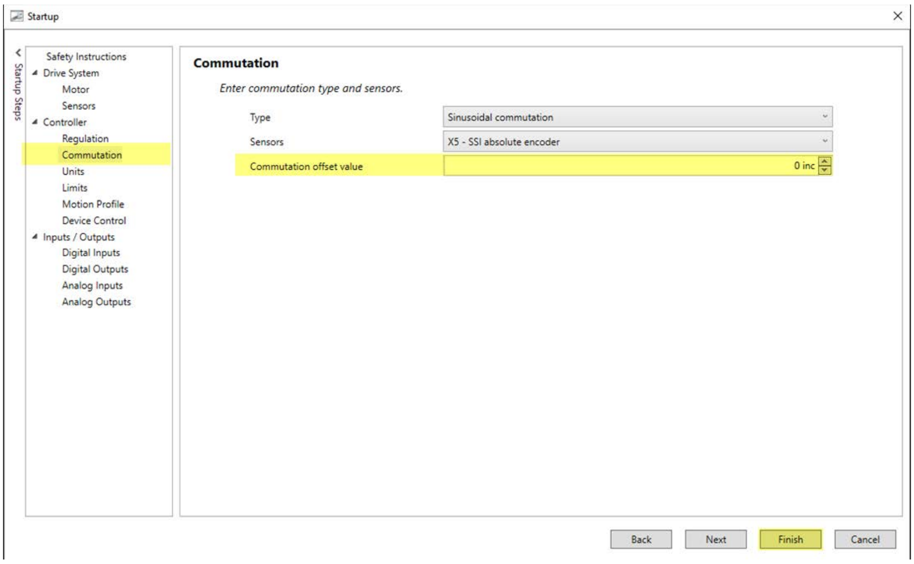


Figure 4-19 Adjustment of SSI & BiSS-C commutation offset value | Set Communication offset value

Step 10 – Tune the controller

- Open the **Regulation Tuning** wizard and complete all tuning steps.

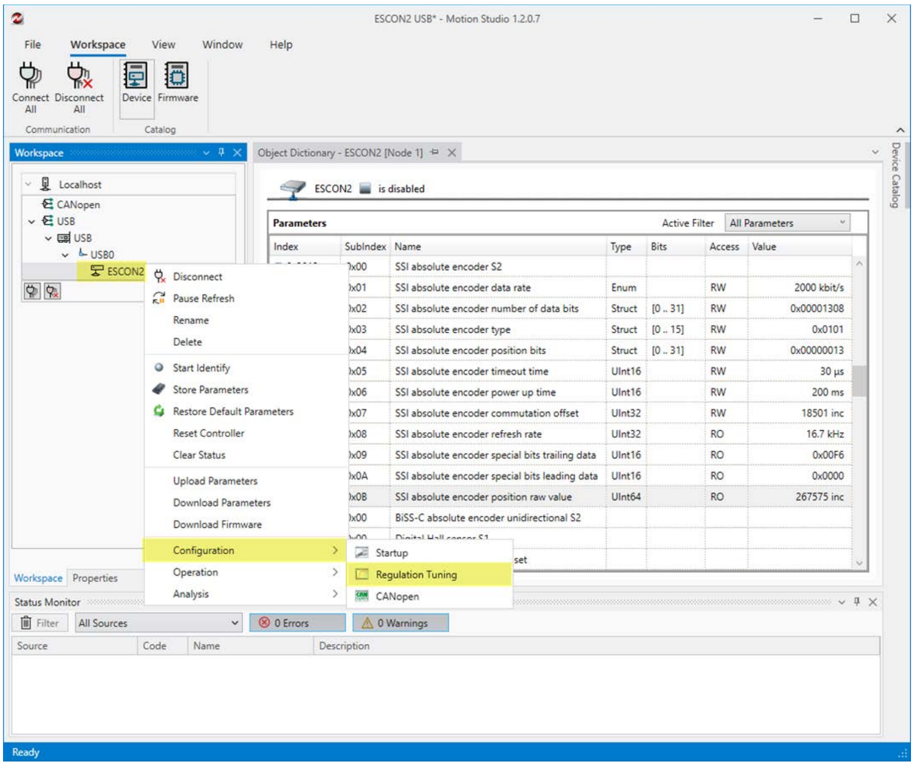


Figure 4-20 Adjustment of SSI & BiSS-C commutation offset value | Identify parameters

Step 11 – Verify the result

- Run the motor in **Profile Velocity Mode**.

If **Current actual value averaged** is unusually high, or if the motor velocity does not match the commanded value, readjust the **Commutation offset value** and repeat the procedure.

4.4 Calculation example «step-by-step instruction»

The following values are used in this example:

- **Number of pole pairs:** 7
- **Single-turn bits:** 12
- **Encoder direction:** maxon (0 = not inverted)
- **SSI position raw value:** 3 205 inc

1) **Step 1 – Calculate the mechanical resolution**
The mechanical resolution corresponds to one full mechanical revolution:

$$ResMech = 2^{\mathrm{SingleTurnBits}} = 2^{12} = 4096 inc|rev$$

2) **Step 2 – Calculate the electrical resolution**
The electrical resolution equals the mechanical resolution divided by the number of pole pairs:

$$ResElec = \frac{ResMech}{PolePairs} = \frac{4096}{7} = 585.1429 inc|electrical, rev$$

3) **Step 3 – Determine the single-turn encoder position**
Apply a modulo operation to wrap the raw value into one mechanical turn:

$$PosSingleTurn = mod(PosRaw, ResMech) = mod(3205, 4096) = 3205$$

4) **Step 4 – Determine the reference angle**
Because the encoder direction is not inverted, the reference angle is:

$$Ref = \frac{5}{6} x ResElec = \frac{5}{6} x 585.1429 = 487.6190 inc$$

5) **Step 5 – Calculate the preliminary offset**
Subtract the encoder position from the reference angle:

$$Offset = mod(Ref - PosSingleTurn, ResElec)$$

Insert the values:

$$Offset = mod(487.6190 - 3205, 585.1490)$$

$$Offset = mod(-2717.381, 585.1429)$$

Modulo operation:

$$-2718.381 + 5x585.1429 = 208.3333$$

Thus:

$$Offset = 208.3333$$

6) **Step 6 – Round the result**

$$Offset_{final} = round(208.3333) = 208$$

7) **Final commutation offset value: 208**

LIST OF FIGURES

LIST OF TABLES

INDEX

CCMC | ESCON2 Servo Controllers Application Notes | Edition 2026-01 | DocID rel13317