



ETHERCAT SET UP GUIDE

Riccardo Piccinini
Mini Motor S.P.A.
FBS-ECT-EN
Firmware Revision: >4.039

Version 3, 07-2026

Table of Contents

1. EtherCAT	1
1.1. Overview	1
1.2. DS301 Communication Layer	1
1.3. Object Dictionary	3
2. Application Example - Omron	10
2.1. Tools	10
2.2. Overview	10
2.3. Install DBS ESI File	10
2.4. Profile Example	12
2.5. CSP Example	14
3. Application Example - Beckhoff	18
3.1. Tools	18
3.2. Overview	18
3.3. Profile Example	20
3.4. CSP Example	21
3.5. EoE Activation	22

1. EtherCAT

1.1. Overview



Note: available only for drivers equipped with CanOpen/Modbus RTU, CanOpen over EtherCAT or Ethernet PowerLink. (---/---/C; ---/---/ETH; ---/---/EPL).

The present document describes the adherence of the Minimotor stack to the following specification:

CanOPEN

- CiA DS301 v4.02 (CanOPEN DLL)
- CiA DS402 v3.0.1.15 (servodrive profile)

CanOPEN over EtherCAT:

- ETG1000-2 v1.0.1 (physical layer)
- ETG1000-3 v1.0.1 (DLL services)
- ETG1000-4 v1.0.1 (DLL protocols)
- ETG1000-5 v1.0.1 (AL services)
- ETG1000-6 v1.0.1 (AL protocols)
- CiA DS402 v3.0.1.15 (servodrive)
- ETG6010 v1.1.0 (CiA 402 impl. directives)

Ethernet PowerLink (EPL)

- EPSG DS 301 V1.2.0 (Powerlink DLL)

The purpose of the present document is to describe what optional features are implemented by the servo-motor CanOPEN, CoE or EPL.

Applicable ---/---/--- software release: $\geq v4.036$

1.2. DS301 Communication Layer

This paragraph describes the details of the DS301 layer of the CanOpen interface. It applies only to the CanOPEN over can-bus implementation.

1.2.1. Physical Layer

Supported baudrate:

- 1Mbps tq=71ns SamplingPoint=11tq
- 500Kbps tq=142ns SamplingPoint=12tq

- 250Kbps $t_q=284\text{ns}$ SamplingPoint=12 t_q
- 125Kbps $t_q=571\text{ns}$ SamplingPoint=12 t_q
- 50Kbps $t_q=1.43\mu\text{s}$ SamplingPoint=12 t_q
- 20Kbps $t_q=3.57\mu\text{s}$ SamplingPoint=12 t_q

1.2.2. SDO protocol

- Maximum supported object size: 32bit
- Expedited transfer supported SUPPORTED
- Segmented transfer: NOT SUPPORTED
- Block transfer: NOT SUPPORTED

1.2.3. TIME protocol

The time protocol is NOT supported.

1.2.4. PDO protocol

The implementation supports up to 4 Pdo TX and 4 Pdo RX; each object can map up to 8 objects; mapping can be done only using the whole size of the object (i.e. is not possible to map 8 bit of a 32 bit object).

The following transmit type are implemented for PDO TX:

Transmission Type	Description
0	PDO is emitted when changes occur after SYNC is received
1..240	PDO emitted after SYNC, depending on SYNC message occurrence (e.g. 1 = every SYNC, 2 = every 2 SYNC, etc.)
241..251	Not supported
252	PDO emitted after RTR reception * Data sampled on SYNC * Emitted after RTR
253	PDO emitted after RTR reception * Data sampled and emitted after RTR
254	Not supported
255	PDO emitted asynchronously Two features supported for emission control: * Inhibit time (emission only when PDO content changes) * Event timer (periodical asynchronous emission)

Defaults:

- inhibit time: 100ms
- event time: 0 (periodic emission on TT=255 disabled by default)

See application profile for default PDO configuration. ==== EMCY protocol

Emergency protocol is implemented for reporting alarms from the servodrive; to control the emission of

EMCY messages, the inhibit time (object 1015h) can be modified. See the application profile for detailed list of alarms. (See paragraph 8.3)

1.2.5. NMT protocol

The NMT implementation includes standard commands for NMT state machine management. For node alive checking, the NODE GUARDING protocol is implemented; the behaviour of node guarding is set by objects 100Ch (guard time) and 100Dh (life factor).

1.2.6. HEARTBEATING

Heartbeating Protocol is supported. Register 1016h defines the time in multiples of 1ms for the Heartbeat protocol consumer. Register 1017h determines the time in multiples of 1ms for the producer of the protocol. Object 1029h determines the behaviour when the error is triggered.

1.3. Object Dictionary

Object dictionary is divided in the following sections:

Range Objects	Functions
1000h .. 1FFFh	Communication profile area
2000h .. 21FFh	Manufacturer parameter area
3000h .. 3FFFh	Manufacturer specific object area
6000h .. 67FFh	Profiled objects

Other areas are unused; the object list is common between CanOPEN, CoE and EPL implementation.

1.3.1. Communication area

The following table shows the list of communication area objects; since some objects are relevant only for CanOPEN or CoE implementation, the table shows in which fieldbus the object is relevant.

Index	Description	CANopen	CoE	EPL
1000	Device type	X	X	X
1001	Error register	X	X	X
1003	Predefined error	X	X	
1006	Cycle length			X
100C	Guard time	X		
100D	Life factor	X		
1010	Store parameters ^[1]	X	X	X
1014	EMCY cob-id	X		
1015	EMCY inhibit time	X	X	

Index	Description	CANopen	CoE	EPL
1016	Consumer Heartbeat Time	X		
1017	Producer Heartbeat Time	X		
1018	Identity object	X	X	X
1020	CFM Verify Configuration			X
1029	Error Behaviour	X		
1030	NMT Interface Group 0h			X
1300	SDO Sequ Layer Timeout			X
1400	PDO RX 1 – config ^[2]	X	X	X
1401	PDO RX 2 – config	X		
1402	PDO RX 3 – config	X		
1403	PDO RX 4 – config	X		
1600	PDO RX 1 – mapping ^[2]	X	X	X
1601	PDO RX 2 – mapping	X		
1602	PDO RX 3 – mapping	X		
1603	PDO RX 4 – mapping	X		
1800	PDO TX 1 – config ^[2]	X	X	X
1801	PDO TX 2 – config	X		
1802	PDO TX 3 – config	X		
1803	PDO TX 4 – config	X		
1A00	PDO TX 1 – mapping ^[2]	X	X	X
1A01	PDO TX 2 – mapping	X		
1A02	PDO TX 3 – mapping	X		
1A03	PDO TX 4 – mapping	X		
1C00	Sync manager types		X	
1C0A	DLL_CnCollision REC			X
1C0B	DLL_CnLoss REC			X
1C0C	DLL_CnLossSoC_REC			X
1C0D	DLL_CnLossSoA REC			X
1C0E	DLL_CnSpCJitter REC			X
1C0F	DLL_CnCRCError REC			X
1C10	Sync manager 0 config		X	
1C11	Sync manager 1 config		X	
1C12	Sync manager 2 config		X	

Index	Description	CANopen	CoE	EPL
1C13	Sync manager 3 config ^[2]		X	
	DLL_CnSocJitterRange ^[2]			X
1C14	DLL_CnLossOfSocTolerance			X
1C32	Output sync parameters		X	
1C33	Input sync parameters		X	
1E40	NWL_IpAddrTable_0h_REC			X
1E4A	NWL_IpGroup_REC			X
1F82	NMT_FeatureFlags_U32			X
1F83	NMT_EPLVersion_U8			X
1F8C	NMT_CurrNMTState_U8			X
1F93	NMT_EPLNodeID_REC			X
1F98	NMT_CycleTiming_REC			X
1F99	NMT_CNBasicEthernetTimeout_U32			X
1F9A	NMT_HostName_VSTR			X
1F9E	NMT_ResetCmd_U8			X

1.3.2. Manufacturer parameter area

The object range 2000h ... 21FFh allows to access servodrive parameters; the following table summarize the details of a parameter object:

Object details

Campo	Valore
Subidx	0
Object type	16bit, signed integer
Access	Read/Write
PDO mappable	NO

Object dictionary for manufacturer parameters

Index	Description
2000h	Parameter 0
2001h	Parameter 1
...	...
21FFh	Parameter 511

For details of the parameter meaning and encoding refer to the servodrive manual. **NOTE:** write objects in the

2000h ... 21FFh range will alter only RAM image of the parameter; to actually save parameter to non-volatile storage a write to object 1010h.0 needs to be performed (see 1010h).

1.3.3. Manufacturer Specific Area

This area contains several objects that allow to read/write manufacturer specific data.

Index	Access	Data type	PDO mappable	Description
3000h	RO	INT16	YES	Ain0 value (-32768 to 32768)
3001h	RO	UINT16	YES	Digital input bits 0–4
3002h	RW	UINT16	YES	Digital output bit0
3003h	RO	INT16	YES	Heatsink temperature (x0.1 °C)
3005h	RO	INT16	YES	Motor temperature (x0.1 °C)
3008h	RO	UINT16	YES	Actual alarm code (not EMCY)
3009h	RO	UINT	NO	Events (0–16, see section 6.1)
3010h	RO	UINT16	YES	Power stage tension (x0.1 V)
3011h	RO	INT16	YES	Motor IQ current (mA)
3012h	RW	UINT16	YES	Motor IQ limit (mA)
3014h	RO	INT16	YES	Output power (1W/lb)
3020h	RW	INT32	YES	Touchprobe positioning offset
3021h	RW	UINT32	YES	Touchprobe positioning modulo
3030h	RO	UINT16	YES	Acceleration RMS modulus (milli-G)
3031h	RO	UINT16	YES	Acceleration RMS x (milli-G)
3032h	RO	UINT16	YES	Acceleration RMS y (milli-G)
3033h	RO	UINT16	YES	Acceleration RMS z (milli-G)
3034h	RO	UINT16	YES	Acceleration DC x (milli-G)
3035h	RO	UINT16	YES	Acceleration DC y (milli-G)
3036h	RO	UINT16	YES	Acceleration DC z (milli-G)
3037h	RO	INT16	YES	Instantaneous Acceleration x (milli-G)
3038h	RO	INT16	YES	Instantaneous Acceleration y (milli-G)
3039h	RO	INT16	YES	Instantaneous Acceleration z (milli-G)

1.3.4. Profiled Objects

The range 6000h ... 67FFh of objects are defined as specified by CiA402; refer to the application area for further information.

Index	Access	Type	PDO mappable	Description
6040h	RW	U16	YES	Control word
6041h	RO	U16	YES	Status word
605Ah	RW	U16		Quickstop option code
605Bh	RW	U16		Shutdown option code
605Ch	RW	U16		Disable operation option code
605Dh	RW	U16		Halt option code
605Eh	RW	U16		Fault reaction option code
6060h	RW	S8	YES	Mode of operation
6061h	RO	S8	YES	Mode of operation display
6062h	RO	S32	YES	Position demand value
6064h	RO	S32	YES	Position actual value
6065h	RW	U32		Following error window
6066h	RW	U16		Following error time
6067h	RW	U32		Position window
6068h	RW	U16		Position window time
606Bh	RO	S32	YES	Velocity demand
606Ch	RO	S32	YES	Velocity actual value
606Dh	RW	U16		Velocity window
606Eh	RW	U16		Velocity window time
606Fh	RW	U16		Velocity threshold
6070h	RW	U16		Velocity threshold time
6071h	RW	S16	YES	Target torque
6072h	RW	U16	YES	Max torque
6073h	RW	U16	YES	Max current
6074h	RO	S16	YES	Torque demand value
6075h	RO	U32	YES	Motor rated current
6076h	RO	U32	YES	Motor rated torque
6077h	RO	S16	YES	Torque actual value
6078h	RO	S16	YES	Current actual value
6079h	RO	U32	YES	DC link voltage
607Ah	RW	S32	YES	Target position
607Bh.0	RO	U8	YES	Position Range Limit – number of subindex

Index	Access	Type	PDO mappable	Description
607Bh.1	RW	S32	YES	Negative limit
607Bh.2	RW	S32	YES	Positive limit
607Ch	RW	S32	YES	Home offset
607Dh.0	RO	U8	YES	Software position limit – number of subindex
607Dh.1	RW	S32	YES	Negative limit
607Dh.2	RW	S32	YES	Positive limit
607Eh	RW	U8		Polarity
607Fh	RO	U32	YES	Max speed
6080h	RO	U32	YES	Max speed (duplicate index)
6081h	RW	U32	YES	Profile velocity
6083h	RW	U32	YES	Profile acceleration
6084h	RW	U32	YES	Profile deceleration
6085h	RW	U32	YES	Quick stop deceleration
6087h	RW	U32	YES	Torque slope
6098h	RW	S8	YES	Homing method
6099h.0	RO	U8	YES	Homing speed – number of subindex
6099h.1	RW	U32	YES	Switch speed
6099h.2	RW	U32	YES	Index speed
609Ah	RW	U32	YES	Homing acceleration
60B0h	RW	S32	YES	Position offset
60B1h	RW	S32	YES	Velocity offset
60B2h	RW	S16	YES	Torque offset
60B8h	RW	U16	YES	Touch-probe function
60B9h	RW	U16	YES	Touch-probe status
60BAh	RW	S32	YES	Touch-probe positive edge latched position
60BBh	RW	S32	YES	Touch-probe negative edge latched position
60C0h	RO	S16	NO	IpData submode selection
60C1h.0	RO	U8	YES	IpDataRecord – number of subindex
60C1h.1	RW	S32	YES	IpData record
60C2h.0	RO	U8		Interpolation time period – number of subindex
60C2h.1	RW	S8		Ip time units
60C2h.2	RW	S8		Ip time index

Index	Access	Type	PDO mappable	Description
60F2h	RW	U16	YES	Position Option Code
60F4h	RO	U32	YES	Following Error
60FDh	RO	U32	YES	Digital inputs
60FEh.0	RO	U8	YES	Digital outputs – number of subindex
60FEh.1	RW	U32	YES	Physical outputs
60FEh.2	RW	U32	YES	Output mask
60FFh	RW	S32	YES	Target velocity
6502h	RO	U32	NO	Supported mode of operation

[1] To save 2000h parameters to non-volatile storage a 32bit write to object 1010h.1 needs to be performed; in particular the value to be written to save parameter is 65766173h ("SAVE").

[2] PDO config and mapping have different encoding for CoE/CanOPEN and EPL. Please check DLL manual for details.

2. Application Example - Omron

2.1. Tools

[Sysmac Studio Profile Example](#)

[Sysmac Studio CSP Example](#)

[DBS/FC BSI Interface Software](#)

2.2. Overview

This section shows the creation of a DBS55 smart motor application over EtherCAT using an Omron controller, configured and programmed in the Sysmac Studio development environment.

References:

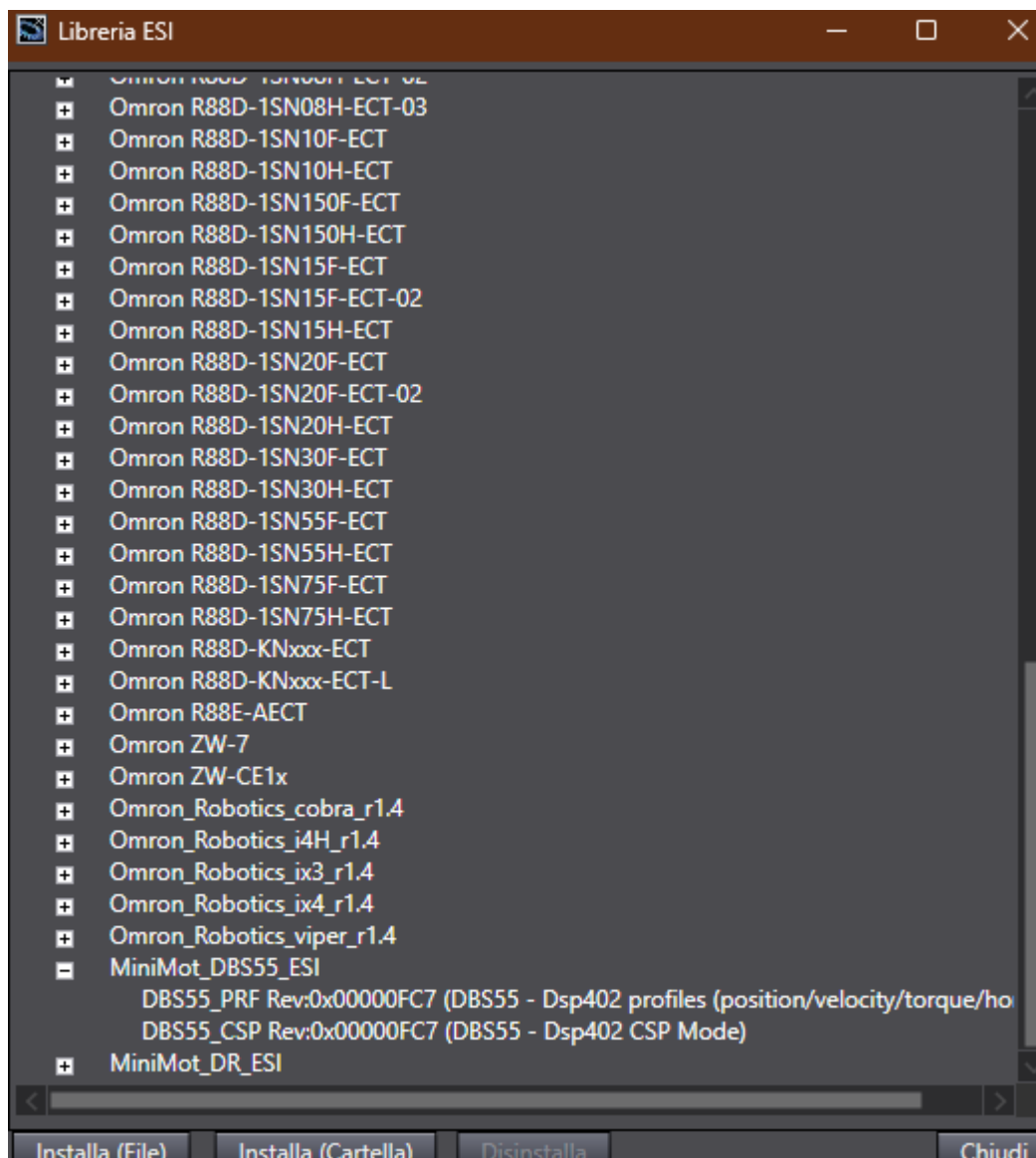
- DBS/FC Firmware version 4.039 or greater
- Sysmac Studio V 1.45



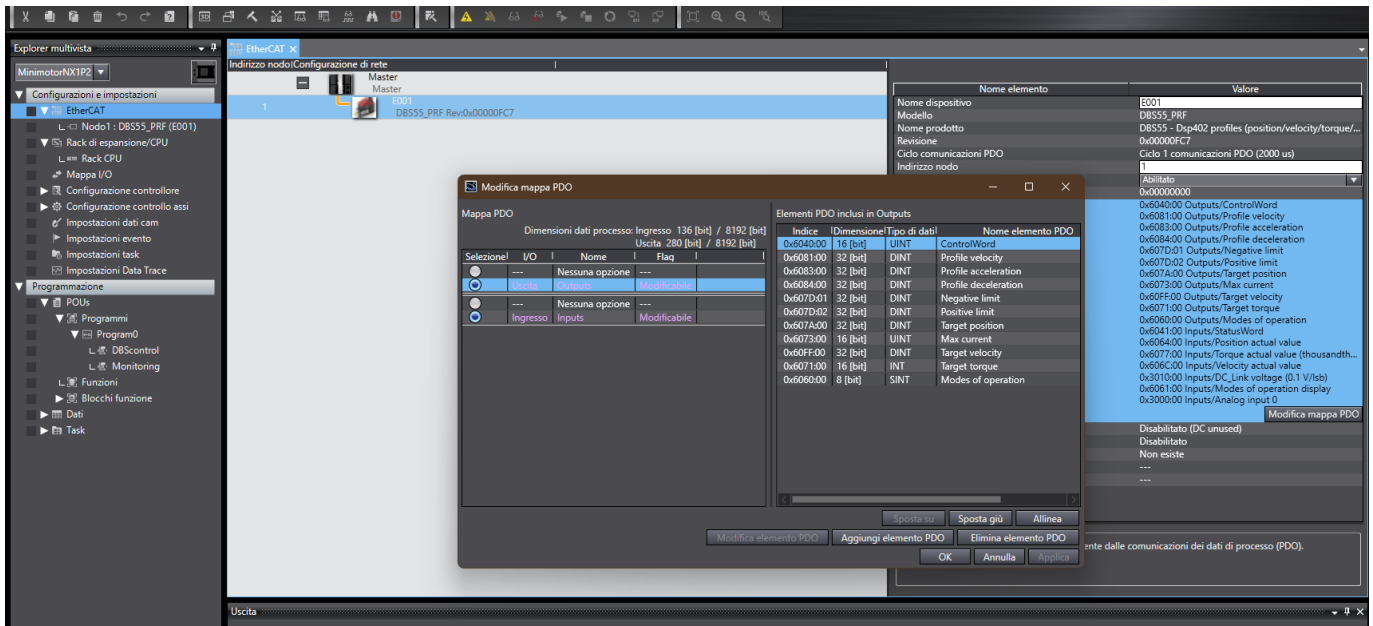
2.3. Install DBS ESI File

- Get the ESI file from the \MM-DBS_v4.xxx\DeviceDesc\EtherCAT folder of the BSI Interface Software.
- Copy it into C:\Program Files\OMRON\Sysmac Studio\IODeviceProfiles\EsiFiles\UserEsiFiles.

- Install the ESI file from the library (right click on the PLC in the EtherCAT node).



- If you want to use the motor as a node, use the DBS55_PRF ESI file.
- If you want to use the motor as an electrical axis, use the DBS55_CSP ESI file. The motor must be ordered with the CSP option for this to work and parameter P198 must be set on 1-CSP (default on CSP motors).
- In the EtherCAT field add the motor, assign the node and build the PDO. In the examples we will have pre-built PDOs to show all motor functions.



2.4. Profile Example

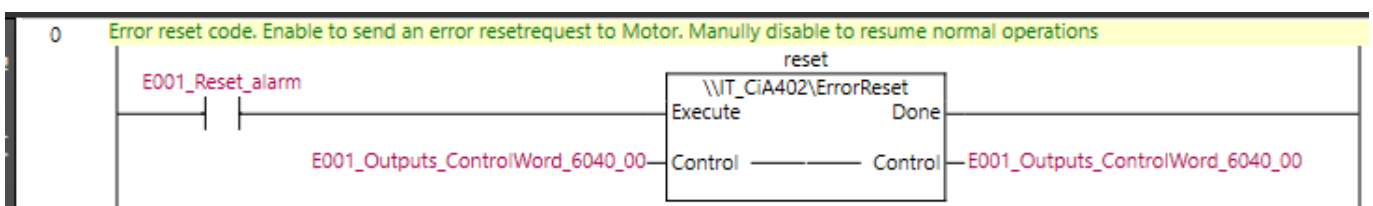
The Profile Example program shows the motor working with the CanOpen DS402 Profile Velocity, Profile Position, Profile Torque and Profile Homing modes, as described by the DS402 standard. You can download the DS402 library used in the example separately.

2.4.1. DBS Control

In this part of the program you have all of the ways to control the motor following the DS402 supported profiles. You can look inside the non-library block to see which data is transferred to the motor.

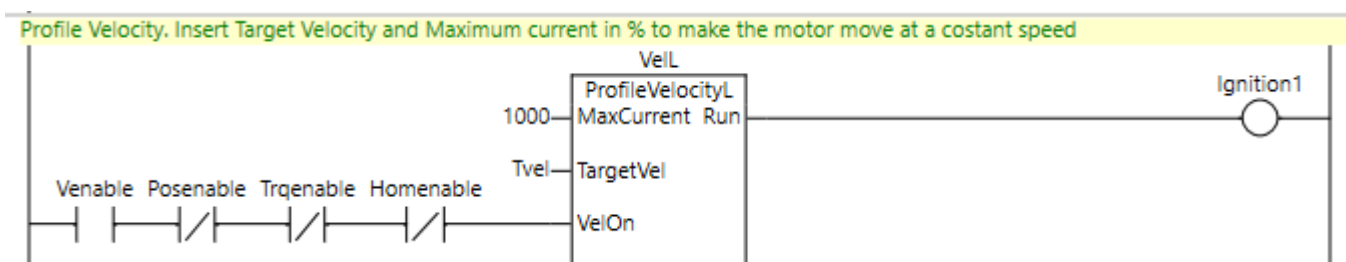
Alarm Reset

This rung is used to send an error reset to the motor. Used to clear alarms.



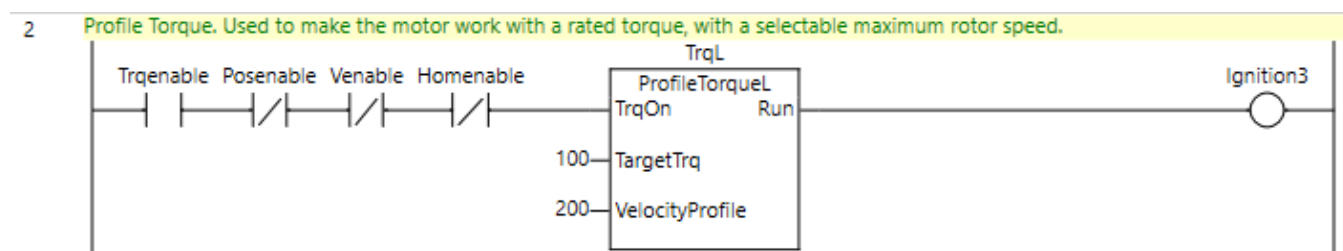
Profile Velocity

In profile velocity the motor keeps the target speed using up to the maximum current. Maximum Current is expressed as a percentage of rated current.



Profile Torque

In the profile torque section the motor will use up to the target torque, keeping a maximum profile speed. Target Torque is expressed as a percentage of rated torque.

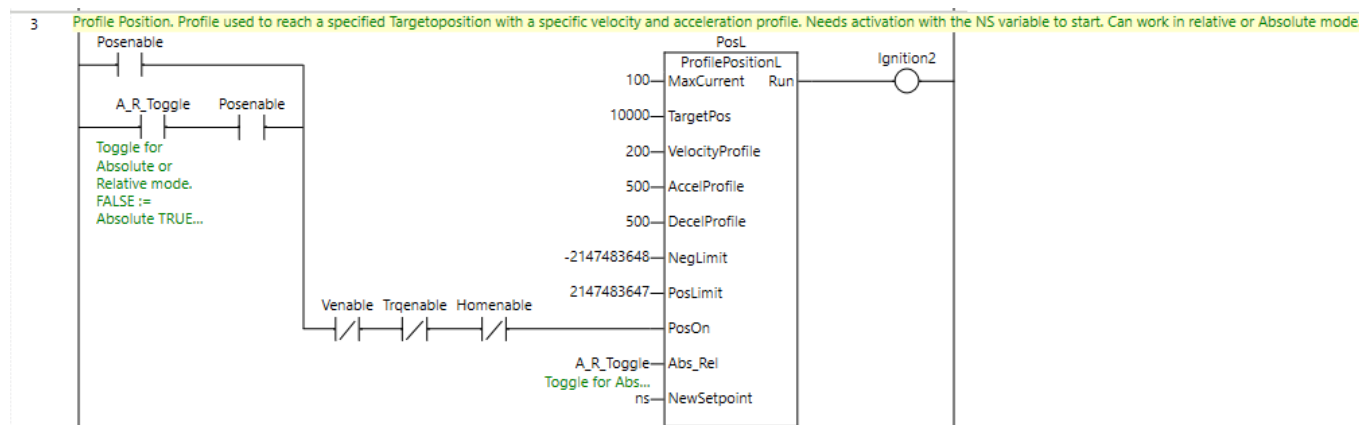


Profile Position

The profile position takes the profile velocity together with acceleration and deceleration, all in rpm and rpm/s, and uses them to generate a trapezoidal trajectory going from the actual position to the target position, using up to the maximum current. The software limits are used as software end of travel. At their maximum value (signed 32 bit register) they are disengaged.

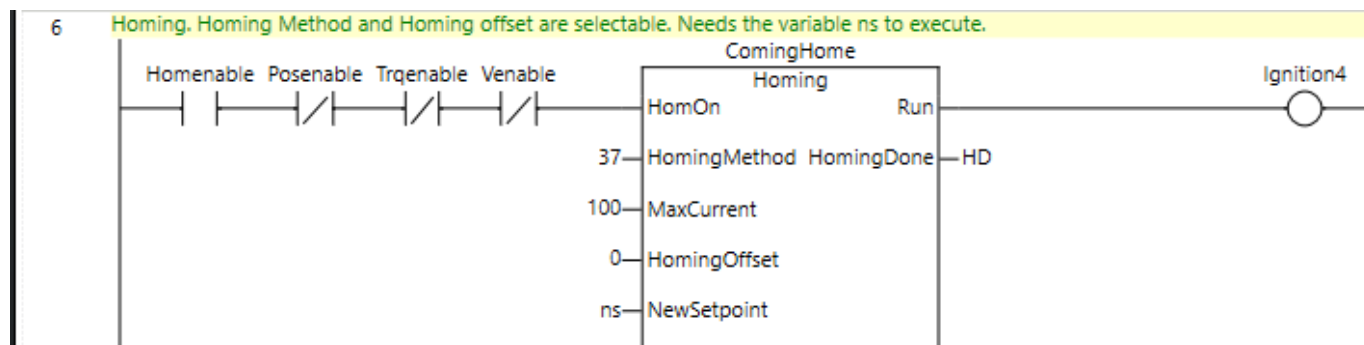
After enabling the block, the motor will enable but will not move. To do so, it requires a rising edge on the New Setpoint bit of the control word, handled by rung 4. Activating this bit will make the motor act on the target position and start moving.

The HALT command is used to momentarily stop the motor command for as long as the halt command is held. Once set to 0 again, the motor will resume the previous command, be it a velocity command or a positioning.



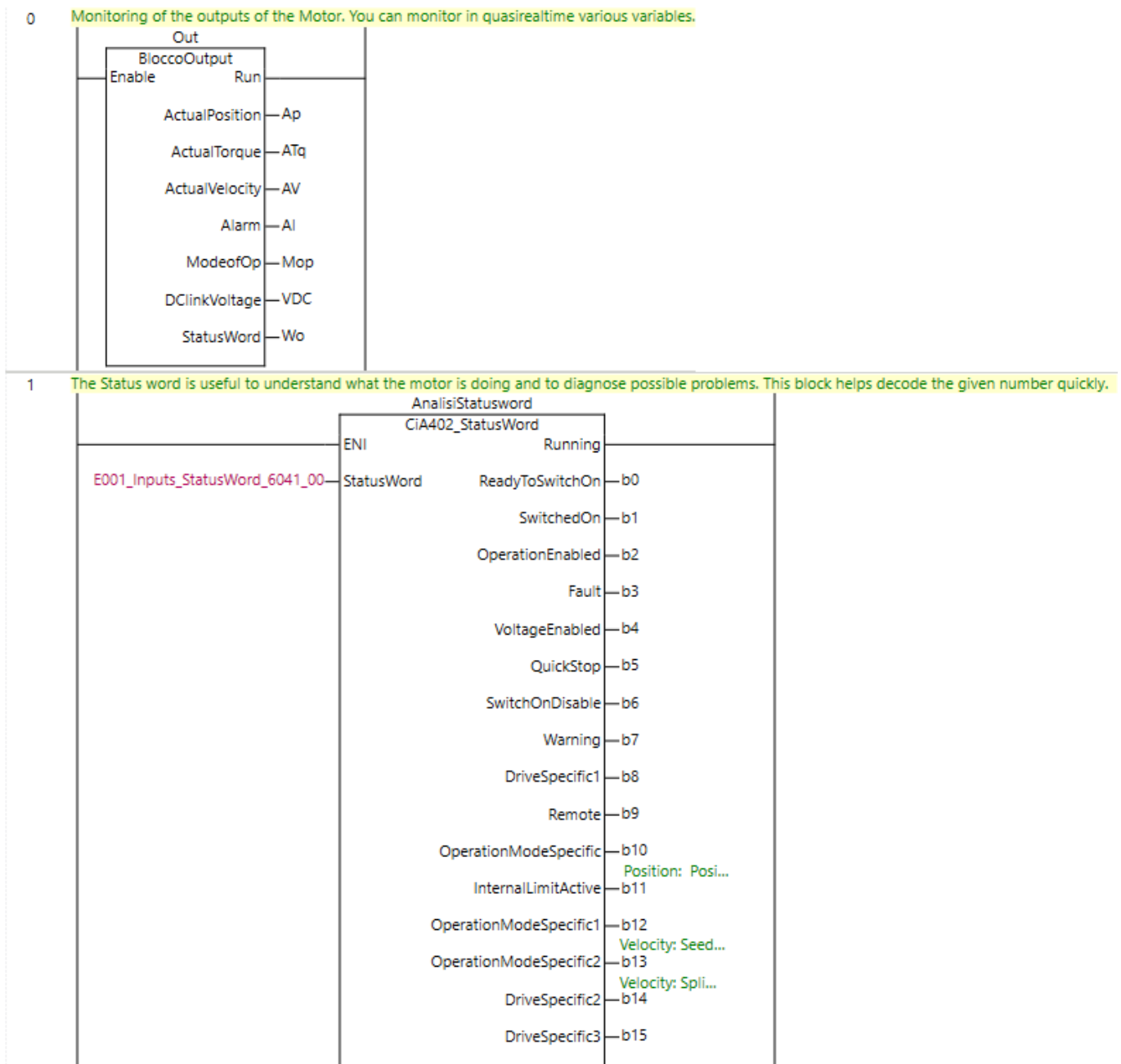
Homing

The homing procedure makes it possible to home the motor position. The default homing method here is to set the current position at 0. Check the manual for different homing procedures. To start the homing you need to activate the New Setpoint in rung 4, same as the profile position.



2.4.2. Monitoring

In the monitoring part you can check all data coming from the motor. In rung 0 you get the macro quantities, like speed, position and torque used. In rung 1 you can read each single bit of the status word to get a proper understanding of the state of the motor.

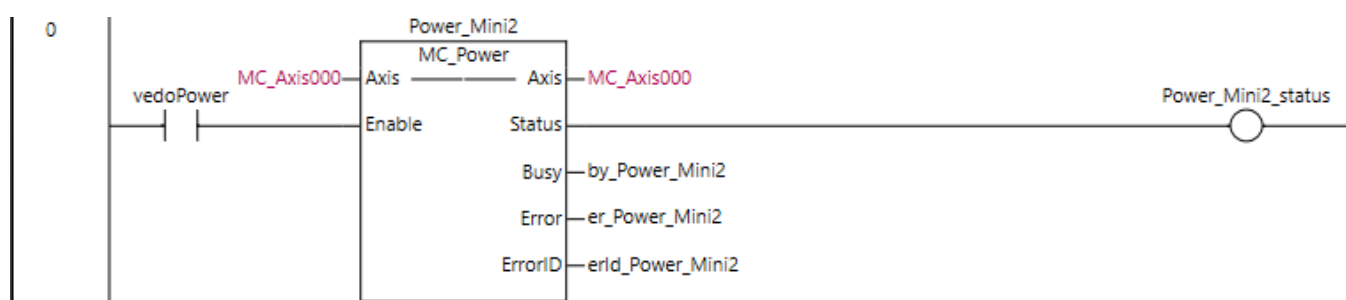
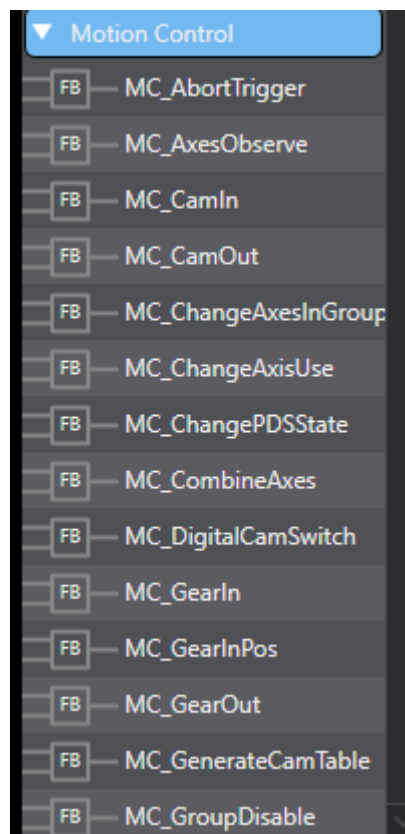


2.5. CSP Example

In the CSP example, the Motion Control blocks directly from Omron are used to get the motor running.

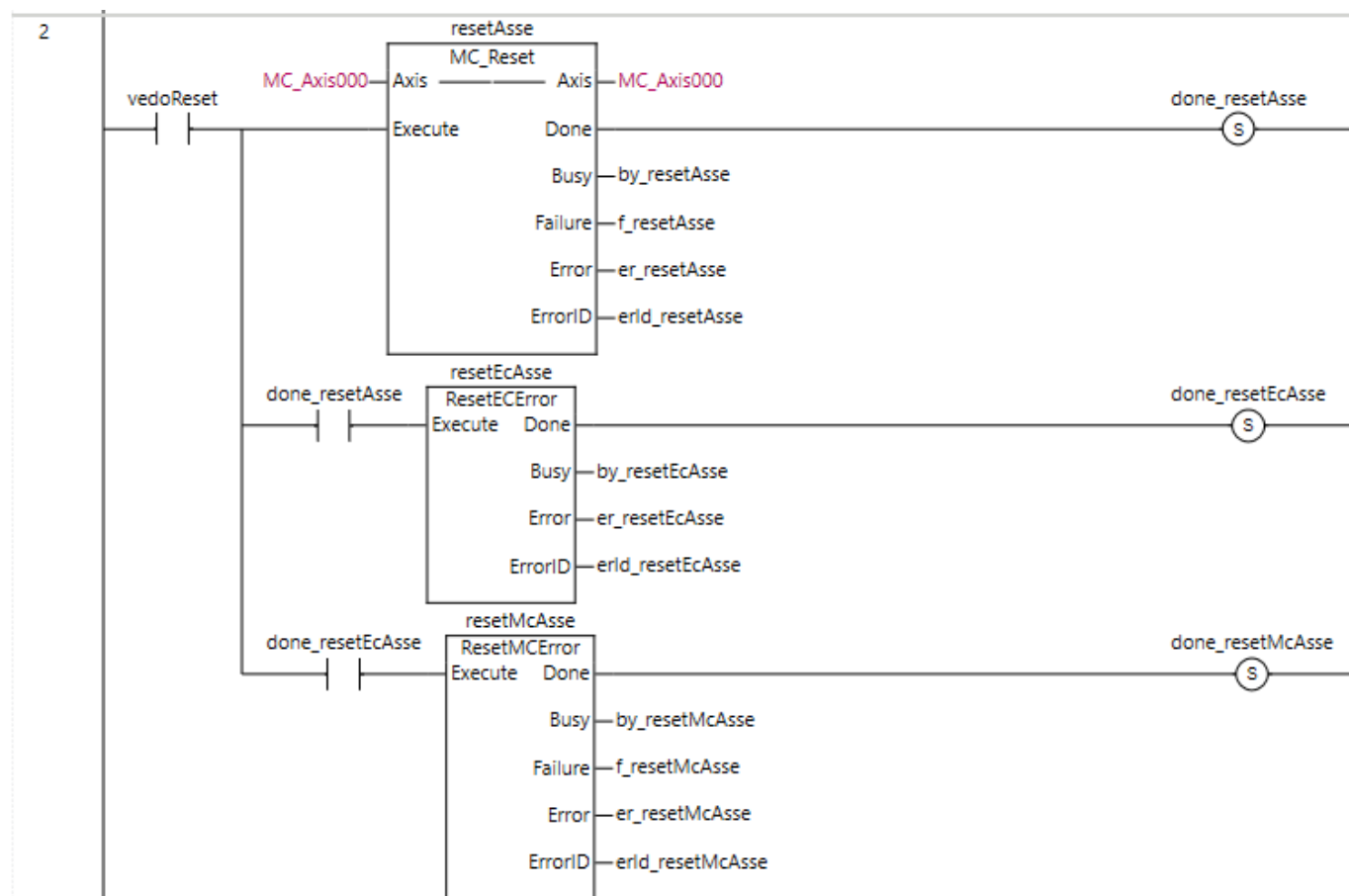
Power

The first rung is used to enable the system. It uses the Motion Control library provided by Omron.



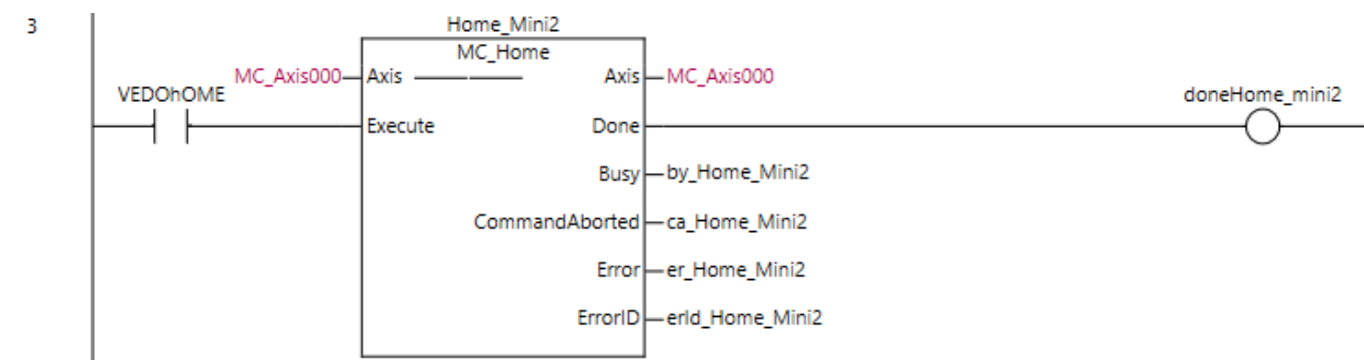
Reset

This rung is used to reset the axis.



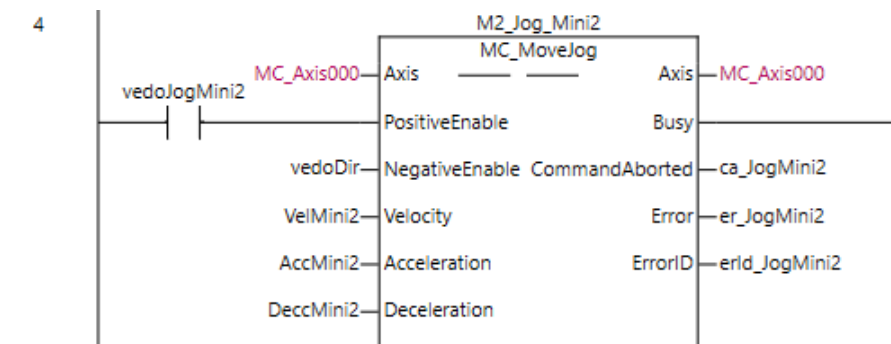
Home

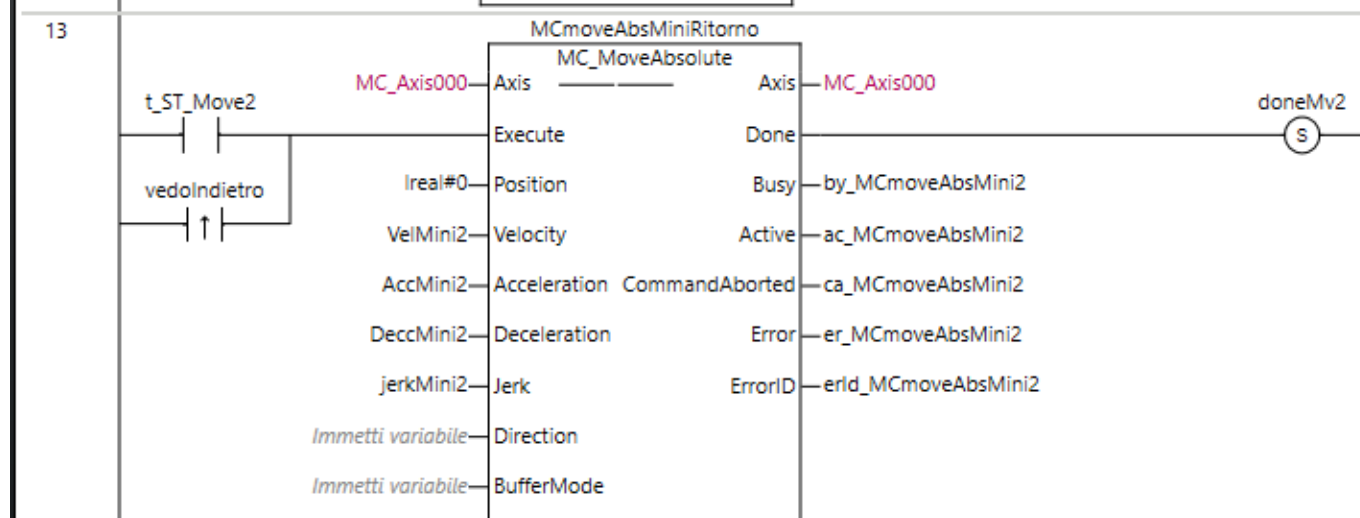
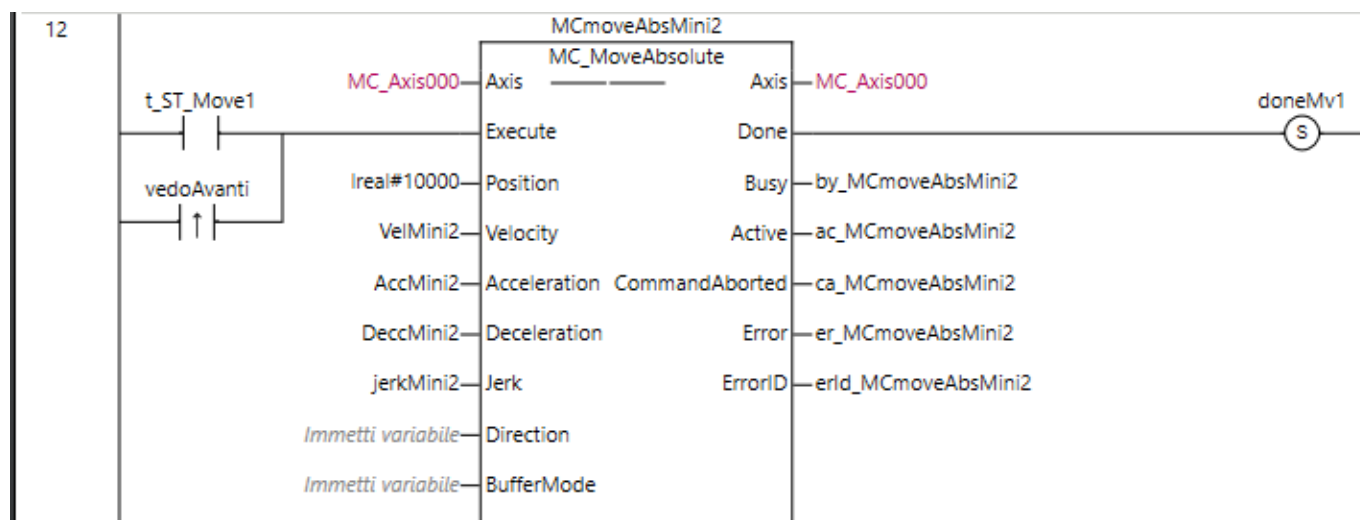
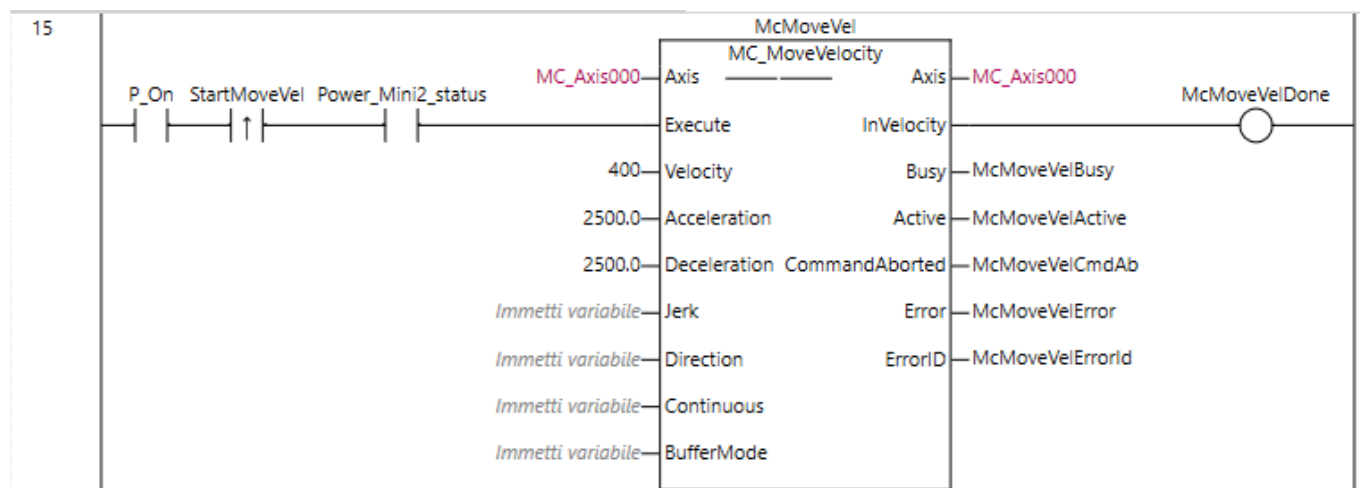
This rung is used to home the axis.



Movement

The movement in this example is shown with the Jog command, with Move Velocity and with positioning between two positions.





3. Application Example - Beckhoff

3.1. Tools

[TwinCAT 3 Profile Example](#)

[TwinCAT 3 CSP Example](#)

[DBS/FC BSI Interface Software](#)

3.2. Overview

This section shows the creation of a DBS55 smart motor application over EtherCAT using a Beckhoff controller, configured and programmed in TwinCAT 3 (TcXaeShell, Visual Studio shell).

References:

- TcXaeShell 15.0.28307.1300 D15.9

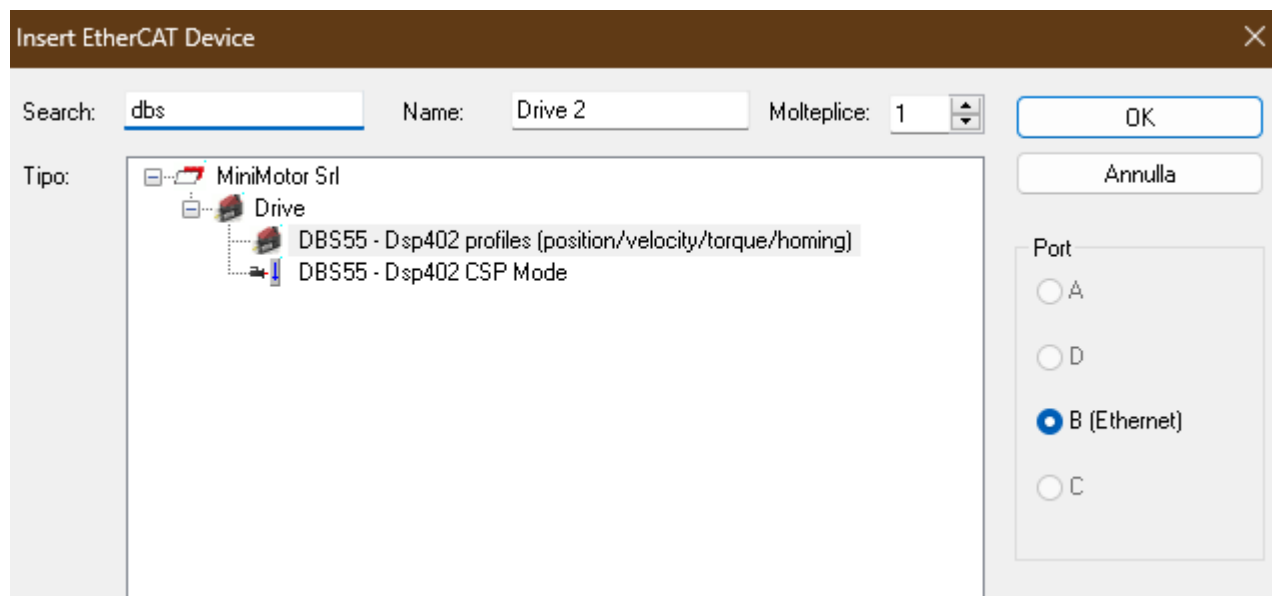


3.2.1. Install ESI File

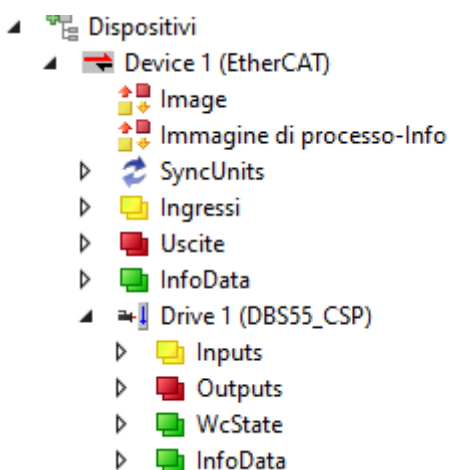
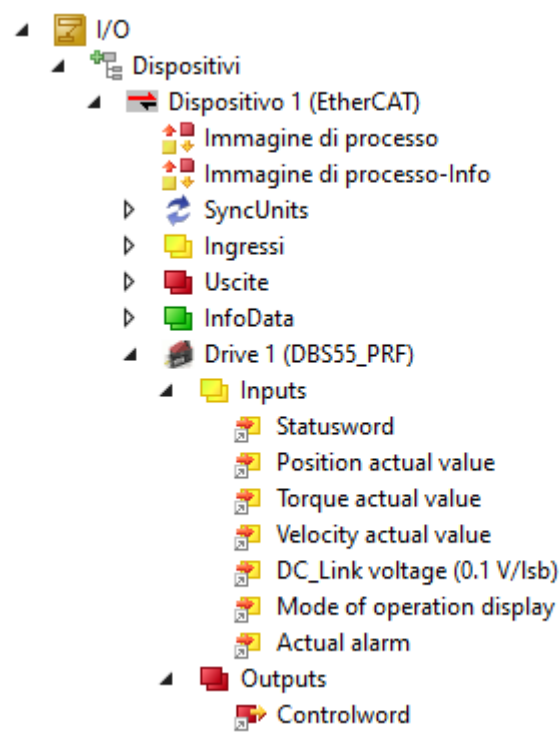
To install the ESI file, simply copy it from the Device Description folder of the BSI Interface Software to the \TwinCAT\3.1\Config\Io\EtherCAT folder.

3.2.2. Import Motor and PDO Configuration

Simply add a Profile device or a CSP device under the EtherCAT device.



Depending on whether you need to command the motor as a node using Profile Speed / Torque / Position, or use it as an electrical axis in interpolated mode, choose the corresponding device.



In interpolated mode it can be linked to an NC axis with the CANopen DS402 / Profile MDP 742 command profile.

In the device properties, in the Process Data tab, you can build your cyclic data exchange to suit your needs.

Nome	Online	Tipo	Gran...	> Indiri...	Entra...	User ID	Collegato a
Statusword	X	UINT	2.0	71.0	Ingre...	0	nState1, nState2
Position actual v...	X	DINT	4.0	73.0	Ingre...	0	nDataIn1 . In . Ingressi
Velocity actual v...	X	DINT	4.0	77.0	Ingre...	0	nDataIn7 . In . Ingressi
Torque actual va...	X	INT	2.0	81.0	Ingre...	0	nDataIn3[0] . nDataIn3 . l...
DC_Link voltage...	X	UINT	2.0	83.0	Ingre...	0	.DcLink . Standard Input...
WcState	X	BIT	0.1	1522.3	Ingre...	0	nState4, nState4
InputToggle	X	BIT	0.1	1524.3	Ingre...	0	nState4, nState4
State	X	UINT	2.0	1548.0	Ingre...	0	
AdsAddr	X	AMSADDR	8.0	1550.0	Ingre...	0	
Chn0	X	USINT	1.0	1558.0	Ingre...	0	
DcOutputShift	X	DINT	4.0	1559.0	Ingre...	0	nDcOutputTime . In . In...
DcInputShift	X	DINT	4.0	1563.0	Ingre...	0	nDcInputTime . In . Ingr...
Controlword	X	UINT	2.0	71.0	Uscita	0	nCtrl1, nCtrl2
Target position	X	DINT	4.0	73.0	Uscita	0	nDataOut1 . Out . Uscite...

3.3. Profile Example

In the profile example provided, the system is already rigged to work with a single motor and to show all the possible basic functions of the drive. The PDO is already built with all the necessary variables to test all the available profiles.

In the visualization provided you can check the Control Word and Status Word state at every step of the test, and you have two main command columns. The one on the far left has all the keys for enabling each operation profile, with the relevant target value below the button. In the central columns you can read all kinds of variables and write the relevant inputs, such as the acceleration and deceleration used to build the

trajectory.

ETHERCAT - Rev 4.037

Fault Reset

Maximum Torque %

Profile Velocity Enable

Target Vel (rpm)

Profile Torque Enable

Target Torque (%)

Profile Pos Enable

Target Pos (pulses)

New Pos Absolute **New Pos Relative**

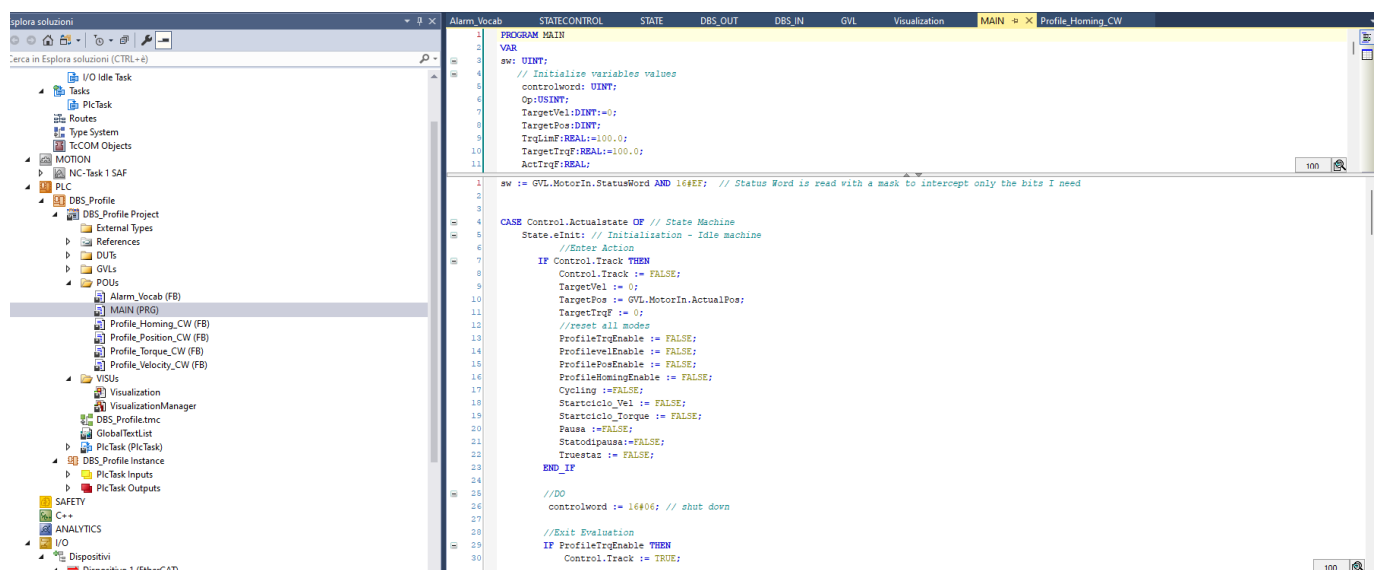
ChangeNow

Profile Homing Enable

Home

Status Word: 16#%x	CW Bit0 Switch On	SW Bit0 Ready to Switch On
Control Word: 16#%x	CW Bit1 Enable Voltage	SW Bit1 Switched On
Mode of Operation: %d	CW Bit2 Quick Stop	SW Bit2 Operation Enabled
Mode of Operation Display: %d	CW Bit3 Enable Operation	SW Bit3 Fault
Actual Vel: %d rpm	CW Bit4 New Setpoint	SW Bit4 Voltage Enabled
Torque: %4.1f%%	CW Bit5 Change Set Immediately	SW Bit5 Quick Stop
Actual Pos: %d pulses	CW Bit6 Abs-Rel	SW Bit6 Switch On Disabled
Actual Alarm: %d	CW Bit7 Fault Reset	SW Bit7 Warning
Profile Vel: %d rpm	CW Bit8 Halt	SW Bit8 M/S
Profile Acc: %d rpm/s	CW Bit9 Change on setpoint	SW Bit9 Remote
Profile Dec: %d rpm/s	CW Bit10 Reserved	SW Bit10 Target Reached
Position Limit NEG: %d pulses	CW Bit11 Touch Probe Offset mode	SW Bit11 Internal Limit Active
Position Limit POS: %d pulses	CW Bit12 Touch Probe Module mode	SW Bit12 Zero Speed / Set Point Ack
Homing Offset: %d pulses	CW Bit13 MS	SW Bit13 Slipage / Following Er
Homing Method: %d	CW Bit14 MS	SW Bit14 Touch Probe Armed
DC Link V: %d	CW Bit15 MS	SW Bit15 Touch Probe Saved

The program is handled by the state machine built in the MAIN (PRG), and the satellite Function Blocks which handle the control word / status word dialogue between the motor and the PLC to transition to the correct state for operations.



3.4. CSP Example

In the CSP example the motor is commanded as an electrical NC axis. The drive is attached to an SDR_Axis and the motor is commanded using the MC_XXX function blocks. Using the interface it is possible to launch the motor in jog mode, in position mode, to have it flip between two positions and to home the motor position.

MiniMotor - DBS55 demo
CSP Interpolation

%s
RUN

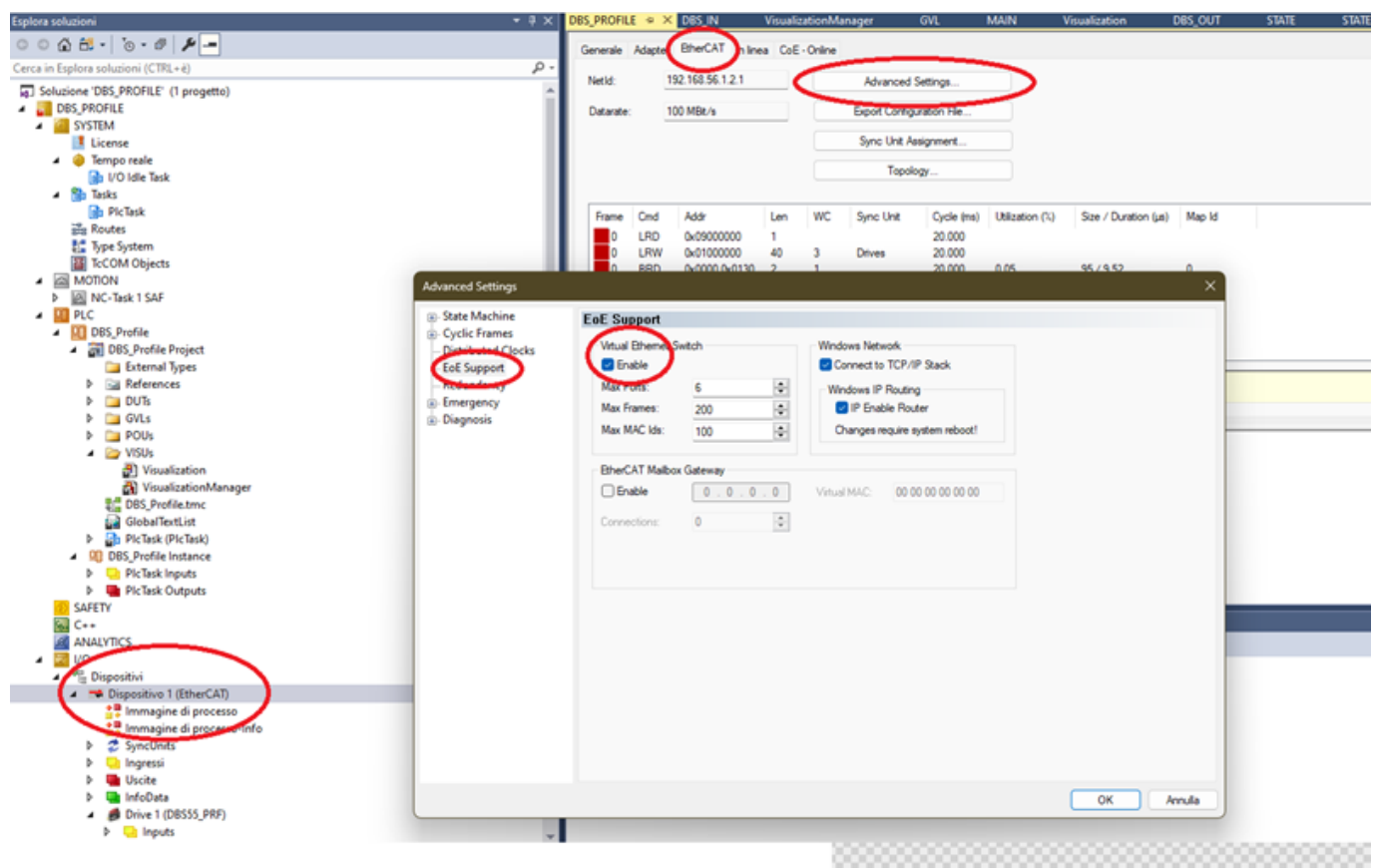
Servodrive Enable

Move Velocity	Move Position	Move Between 2 positions	Home
CC/CCW	POS %6.2f	POS1 %6.2f	
		POS2 %6.2f	
SPEED %6.0f	SPEED %6.0f	SPEED %6.0f	
ACC %6.0f	ACC %6.0f	ACC %6.0f	
DEC %6.0f	DEC %6.0f	DEC %6.0f	
JERK %9.0f	JERK %9.0f	JERK %9.0f	

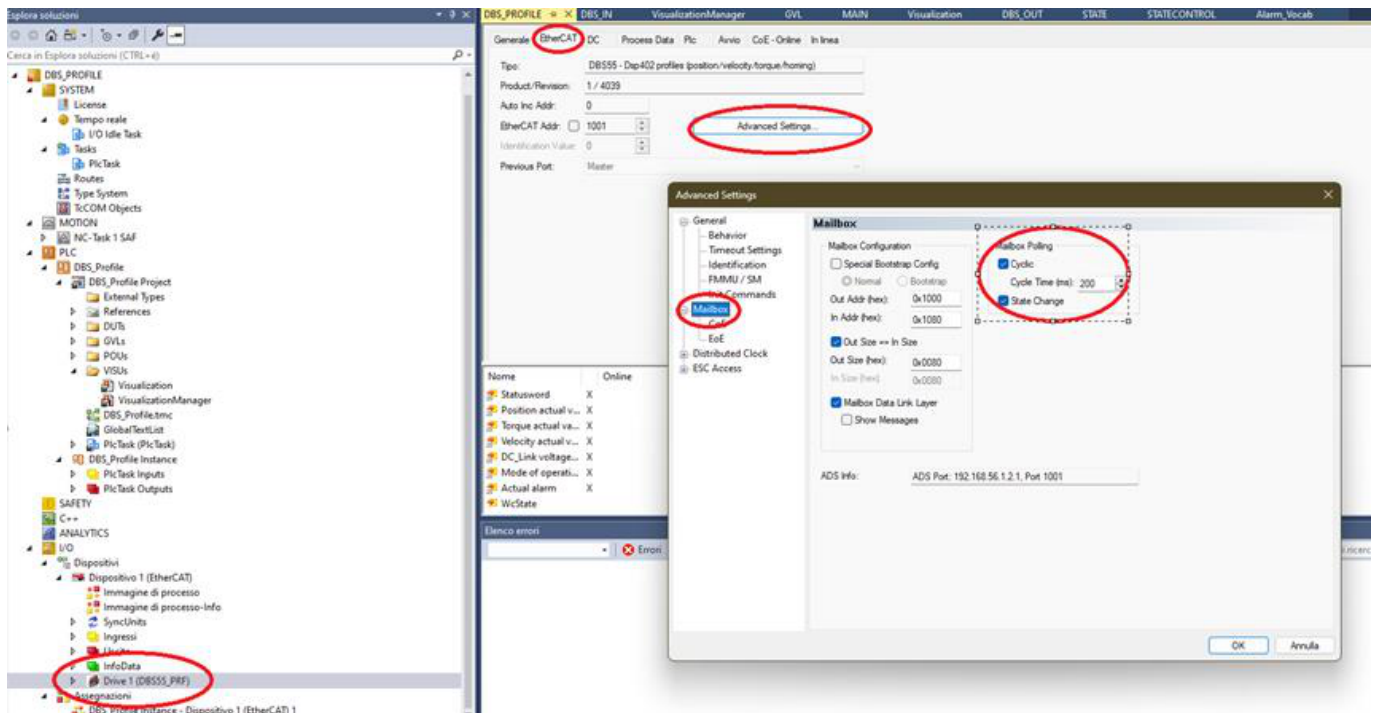
3.5. EoE Activation

If you want to activate the Ethernet over EtherCAT (EoE) feature on Beckhoff, follow this procedure:

- Select the EtherCAT Master → EtherCAT tab → Advanced Settings → EoE Support → Enable.



- Select each device → EtherCAT tab → Advanced Settings → Mailbox → Activate Cyclic polling (50-100 ms).



- Then on the Device → EtherCAT → Advanced Settings → Mailbox → EoE tab, select and choose the IP address you want for it.

Advanced Settings

- General
- Mailbox
 - CoE
 - EoE
- Distributed Clock
- ESC Access

EoE

- ☒ Virtual Ethernet Port
 - Virtual MAC Id: 02 01 05 10 03 e9
 - ☐ Switch Port
 - ☒ IP Port
 - ☐ DHCP
 - ☒ IP indirizzo
 - 192.168.10.10
 - Subnet Mask: 255.255.255.0
 - Default Gateway: 192.168.10.100
 - DNS Server: . . .
 - DNS Name: Drive_1__DBS55_
- ☐ Time Stamp Requested

- Build the program, download it and it will work.



Sometimes Windows can get complicated with routing. In that case check how things are set with the route print command in the Windows CMD; you could add a route with route add [EoE device IP address] [PLC IP address].