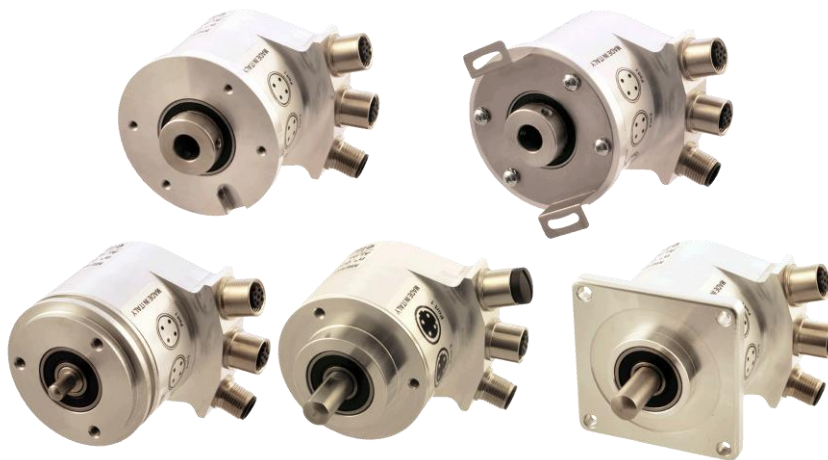


# ABSOLUTE ENCODER

**MEM-BUS** EtherCAT<sup>®</sup> 

## Instruction Manual



**EtherCAT<sup>®</sup>**   
Conformance tested



ELAP VIA VITTORIO VENETO, 4 • I-20094 CORSICO (MI) • TEL. +39.02.4519561  
FAX +39.02.45103406 • E-MAIL [INFO@ELAP.IT](mailto:INFO@ELAP.IT) • SITE [WWW.ELAP.IT](http://WWW.ELAP.IT)

## References

- ETG.1000.1 – Overview
- ETG.1000.2 – Physical Layer service definition and protocol specification
- ETG.1000.3 – Data Link Layer service definition
- ETG.1000.4 – Data Link Layer protocol specification
- ETG.1000.5 – Application Layer service definition
- ETG.1000.6 – Application Layer protocol specification
- ETG.1020 V1.2.0 – Protocol Enhancements
- ETG.1300 V1.1.1 – Indicator and Labeling
- ETG.2010 V1.0.0 – Slave Information Interface
- ETG.2100 V1.0.0 – Network Information
- ETG.2200 V2.1.2 – Slave Implementation Guide
- CiA Draft Standard 301
- CiA Draft Standard 406

## Trademarks and Patents

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

All other specified products, names and logos, serve exclusively for information purposes and may be trademarks of their respective owners, without any special marking to indicate this.



# 1 EtherCAT® Technology

## 1.1 General information

EtherCAT (**E**thernet for **C**ontrol and **A**utomation **T**echnology) is a real-time Ethernet technology, that is particularly suitable for communication between control systems and peripheral devices such as I/O systems, drives, sensors and actuators.

EtherCAT was developed in 2003 by Beckhoff Automation GmbH and it is available as an open standard. The ETG (**E**therCAT **T**echnology **G**roup) association was established for further development of this technology.

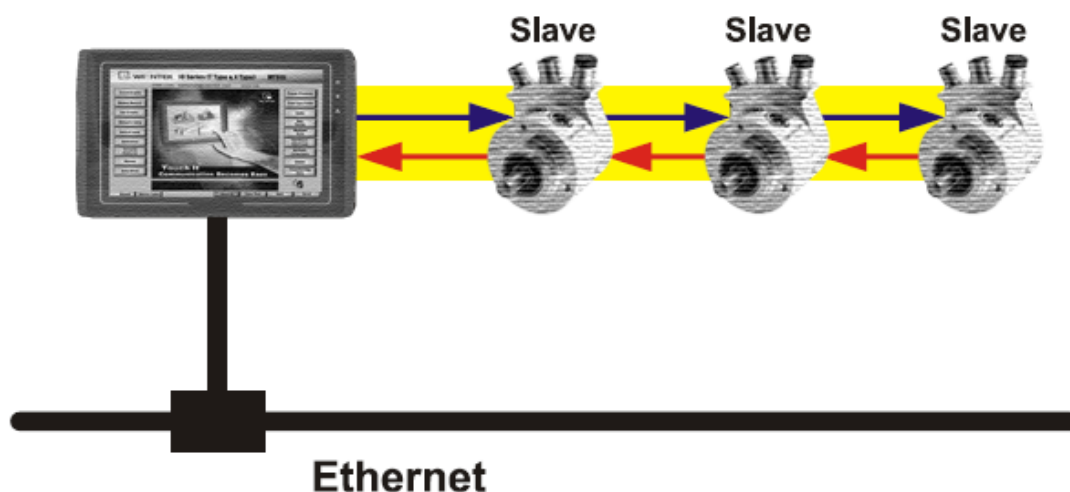
EtherCAT is a public accessible specification, which was published by the IEC (IEC/Pas 62407) in 2005 and it is part of ISO 15745-4. This part was integrated into the new editions of the international field bus standards IEC 61158 (Protocols and Services), IEC 61784-2 (Communication Profiles) and IEC 61800-7 (Drive Profiles and Communication).

## 1.2 EtherCAT® functional principle

The EtherCAT® technology overcomes the generally known limitations of other Ethernet solutions, where the Master Controller generally sends a data request through the network and then waits until data are elaborated by all the devices. This type of communication is not a real-time one, because the Master Controller receives the data at different time from the various slaves in the network.

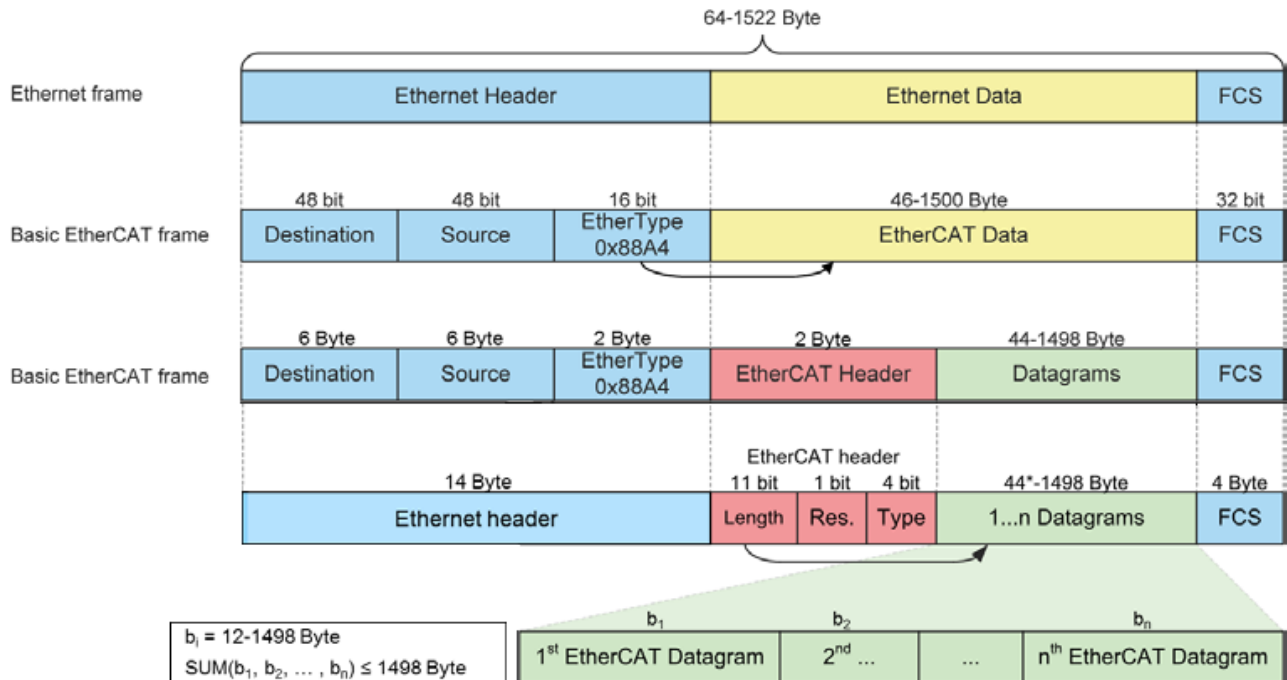
EtherCAT® technology overcomes this limitation because the same data request passes through all the slaves and each device, if addressed, inserts or takes its own data and then transmits the frame to the next device. The last device sends back to all the slaves and then to the Master a complete packet, with all the data requested.

A logical ring structure thus results for this communication. As Fast-Ethernet with Full Duplex, a physical ring structure results, too.



### 1.3 Protocol

The EtherCAT® protocol, optimized for process data, is transported directly in the Ethernet frame via a special Ether type. A complete transmission can consist of several sub-frames. The data sequence is independent of the physical sequence of the slaves in the network. The addressing can be freely selected: Broadcast, Multicast and lateral communications between slaves are possible. The protocol also supports acyclic parameter communication. The structure and meaning of the parameters is defined by the device profile **"CANopen Device Profile for Encoder CiA DS-406"**.



Data in the EtherCAT® frame are transmitted between Master and Slave as Process Data (PDO, **P**rocess **D**ata **O**bjects). Each PDO is associated with an address to one or more Slaves; this association data+address (with other elements and a validation check-sum) forms a telegram (Datagram). One EtherCAT® frame can contain more telegrams and a complete control cycle can need more than one frame.

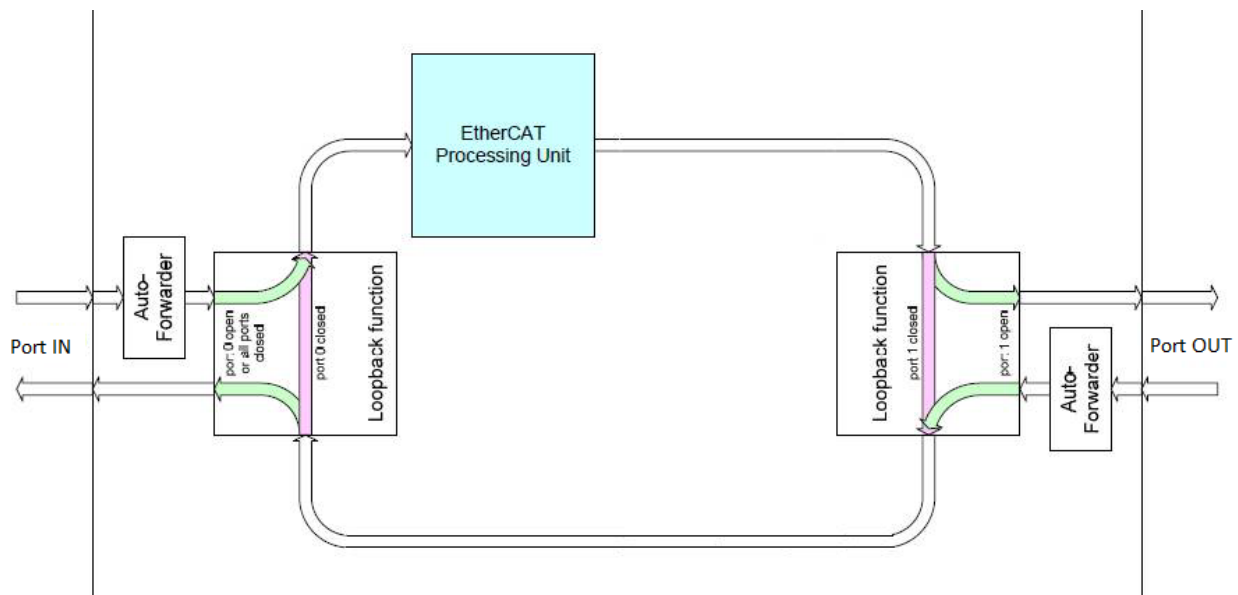
### 1.4 Device address

It is not necessary to assign a physical address to a device, since every Slaves are automatically addressed at power on, when scanning the hardware configuration.

### 1.5 Line termination

Line terminations are not required, since the EtherCAT® network is automatically closed; each Slave is able to feel the presence of the following Slave.

Any physical EtherCAT® network topology always forms a logical ring since the frame processing in a slave works like a roundabout. The ESCs (EtherCAT **S**lave **C**ontroller) are connected to upstream (Master) always via port IN and to downstream (following Slaves) via port OUT. The frame processing is done only once per ESC in the EtherCAT Processing Unit which is located after port IN. Thus, returning frames will not be processed again but are only passed to the next port or returned to port IN.



## 1.6 Communication modes

Elap encoder with EtherCAT® interface supports the following communication modes:

- **Free-Run:** Slave's application is not synchronized to EtherCAT. The local cycle is started by the local timer interrupt of the application controller. The cycle time can be modified by the Master, but this is a local cycle and it does not depend on communication and on Master's cycle.
- **Synchronous with SM3 Event:** Slave's application is synchronized to the SM3 Event (that is the cyclic inputs transmission to the Master). SM events are based on the time an EtherCAT® frame is received. This time ca jitter in the range of a few microseconds due to the EtherCAT Master implementation (delay in stack, PHY & MAC delay, etc.).
- **Synchronous with DC SYNC0 Event:** Slave's application is synchronized to the SYNC0 event, which is base on the Distributed Clocks Unit (DC). The jitter could be reduced to a few nanoseconds.

DC mode is generally used in closed ring networks, when a high real-time performance is required. For all the other cases, when real-time mode is not important, Free-Run or SM3 modes can be used. The **Sync Manager** is a mechanism to protect data in the DPRAM from being accessed simultaneously. EtherCAT Sync Managers can operate in two modes; mode selection and initialization can be made through the XML file or loading data from the corresponding SII section (in EEPROM).

- **Buffered Mode**

This is typically used for cyclic data exchange. Since the buffered mode allows access to the communication buffer at any time for both sides, EtherCAT Master and device application. The sender can always update the content of the buffer. If the buffer is written faster than it is read by the receiver, old data is dropped. Thus, the receiver always gets the latest consistent buffer content which was written by the sender. Sync Managers running in buffered mode need three times the process data size allocated in the DPRAM.

- **Mailbox Mode**

This mode implements a handshake mechanism for data exchange. EtherCAT Master and device application only get access to the buffer after the other one has finished its access. When the sender writes the buffer, the buffer is locked for writing until the receiver has read it out. The mailbox mode is typically used for application layer protocols and exchange of acyclic data (e.g. parameter settings).

## 1.7 Distributed clocks

The method of Distributed Clocks (DC) provides highly precise time synchronization between slaves in an EtherCAT® network. Since DC refers to the ESC internal clocks, slave synchronization between slaves corresponding to a DC is done in hardware and thus guaranteed to much better than 1 microsecond.

One of the slaves (generally the first one) receives the reference Master Clock and then synchronizes the clocks of all the other devices in the network; the Master controller cyclically sends a special synchronization telegram, in which, the reference slave writes its own current time. This telegram is then sent to all the other slaves, that synchronize their own clocks. Time synchronization is very important to obtain an instant picture of the system and execute simultaneous actions in special applications of servo axes control and coordination.

The EtherCAT Slave Controller (ESC) has got one compare unit, that is able to generate synchronization signals directed to the local controller (SYNCO or interrupt). These signals allow the controller itself to synchronize its own local time with the time of the slave.

## 1.8 Slave configuration

The slave communication mode (address, synchronization, PDO data mapping, etc.) can be configured with the ESI (EtherCAT Slave Information) file provided by Elap, in XML format, or loading data from the SII (Slave Information Interface) section of EEPROM of the encoder.

The figure below shows the SII section general structure.

Word

0	EtherCAT Slave Controller Configuration Area							
8	VendorId	ProductCode	RevisionNo	SerialNo				
16								
24	Hardware Delays		Bootstrap Mailbox Config					
	Mailbox Sync Man Config		Reserved					
64	Additional Information (Subdivided in Categories)							
...								
	Category Strings							
	Category Generals							
	Category FMMU							
	Category SyncManager							
	Category Tx- / RxPDO for each PDO							

## 1.9 ESI File

An ESI (EtherCAT Slave Information) description file is provided by Elap in XML format. It contains specific information such as manufacturer, product information, profile, object, process data, synchronization and sync manager settings. This XML file should be integrated in the EtherCAT® configuration tool.



## 2 Installation

The encoder must be installed by qualified experts only, with net voltage off and standstill shaft. Always observe the operating instructions of the machine manufacturer.

### 2.1 Safety

- Always observe prevention and safety norms during the installation and operation of the device.
- Use the encoder exclusively for its intended purpose.
- High voltage, current and rotating parts may cause serious or fatal injuries.
- Encoders must not be operated outside the specified limited values (see detailed product documentation).

### 2.2 Transport and storage

- Always transport or store encoders in their original packaging.
- Never drop encoders or expose them to major vibrations.

### 2.3 Mechanical assembly

- Do not open the device.
- Do not carry out mechanical changes on the device.
- Avoid impacts or shocks on the housing and shaft.
- Operate the device within its environmental specifications.

### 2.4 Electrical supply

- Carry out the wiring operations exclusively with unplugged voltage supply
- Do not operate on the electrical plant while the encoder is on.
- Ensure that the entire plant complies with EMC requirements. The installation environment and wiring affect the electromagnetic compatibility of the encoder. In particular:
  - before handling and installing the encoder, eliminate any electrostatic charge from your body and from any tool that will get in contact with the device;
  - supply the encoder with steady voltage free from electrical noise; if necessary, install EMC filters for the supply input;
  - always use shielded and, if possible, twisted cables;
  - do not use longer cables than necessary;
  - the device cable path should be away from power cables;
  - install the device away from possible interference sources, or shield it effectively;
  - connect the cable shield and the connector case to a protective earth and make sure that the protective earth is free from electrical noise; the connection to earth can be carried out at the encoder side and/or at the user side; it is up to the user to evaluate which is the best solution to keep the electrical interference as low as possible.

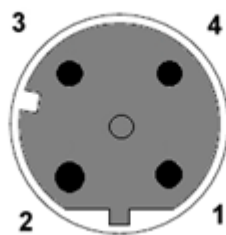
To achieve the highest possible noise immunity the EtherCAT® cable screen must be connected to ground on both ends. In certain cases current might flow over the screen, therefore it is recommended to use equipotential connections.

Input (IN) and output (OUT) connectors cannot be exchanged.

#### Connectors IN and OUT

M12 female type, D code

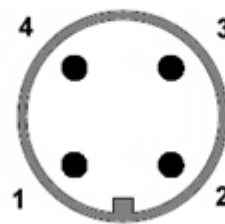
Pin	Signal
1	Tx +
2	Rx +
3	Tx -
4	Rx -



#### SUPPLY CONNECTOR

M12 male type, A code

Pin	Signal
1	Supply voltage (10-30 V DC)
2	N.C.
3	GND (0V)
4	N.C.



## 2.5 LED indicators

Four LEDs show the working conditions of the EtherCAT® interface. See the table below.

Link / Activity (green)	The Link/Activity indicators show the state of the physical link and activity on this link.
OFF	State → port closed, Link → NO, Activity → Non Applicable
Blinking	State → port open, Link → YES, Activity → YES
ON	State → port open, Link → YES, Activity → NO
RUN (green)	The RUN indicator shows the status of the ESM (EtherCAT State Machine).
OFF	The encoder is in the state INIT
Blinking	The encoder is in the state PRE-OPERATIONAL
Single Flash	The encoder is in the state SAFE-OPERATIONAL
ON	The encoder is in the state OPERATIONAL
ERR (red)	The ERROR indicator the presence of errors.
OFF	No error
Blinking	Invalid configuration
Single Flash	Local error, causing a change in the ESM (see ETG1000.6)
Double Flash	Watchdog timeout
ON	Application controller failure, ESC is not responding any more

### 3 CANopen over EtherCAT (CoE)

#### 3.1 General information

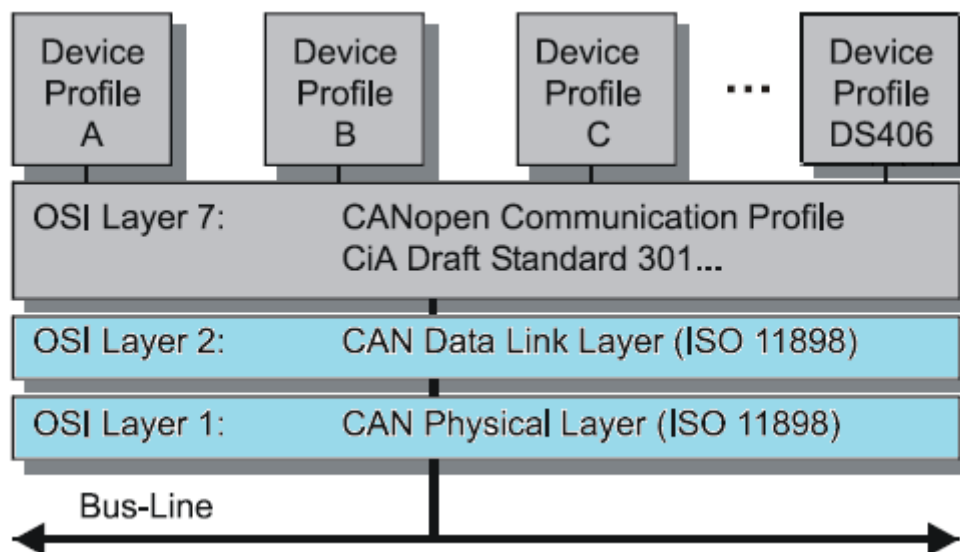
Elap encoders are slave devices, using CoE (CANopen over EtherCAT) protocol for data exchange. The Encoder Profile class 2 (CiA DS-301 and CiA DS-406) is supported.

EtherCAT can provide the same communication mechanisms as those known from CANopen:

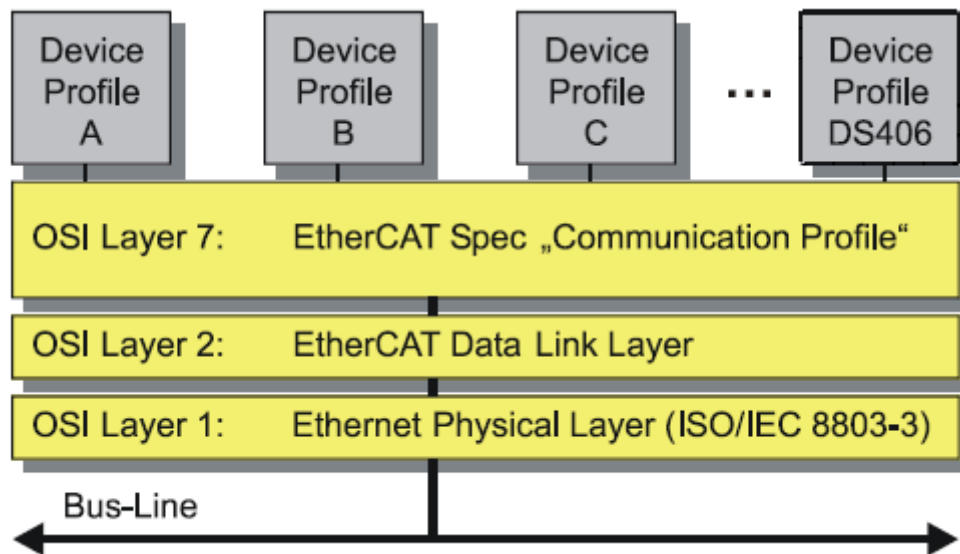
- ESM (EtherCAT State Machine)
- Process data PDO (Process Data Objects)
- Acyclic data SDO (Service Data Objects)
- Object dictionary

The pictures below show a comparison of CANopen / EtherCAT in the ISO /OSI layer model.

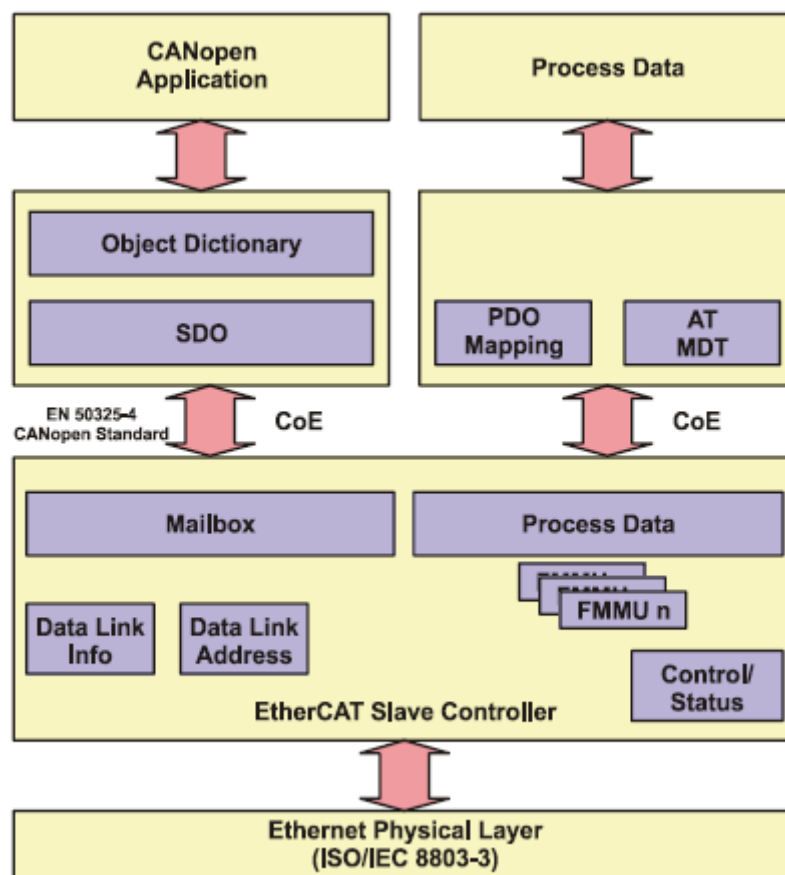
For CANopen:



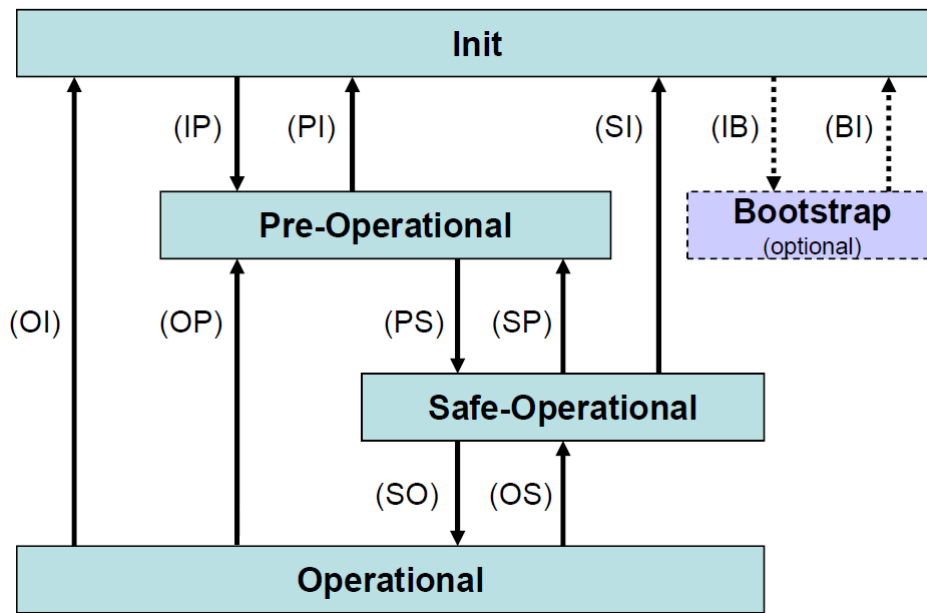
For EtherCAT:



The device profile describes the application parameters and the functional behavior of the device, including the device class-specific state machine. With EtherCAT it is not necessary to develop individual device profiles for device classes, since simple interfaces are provided for existing device profiles. The measuring system the CANopen over EtherCAT (CoE) mailbox protocol, and then the Device Profile for Encoders (CiA DS-406) from CANopen. The picture below shows CoE Communication mechanism.



### 3.2 EtherCAT State Machine (ESM)



The Application Management contains the EtherCAT State Machine, which describes the states and state changes of the slave application. Apart from a few details, the ESM corresponds to the CANopen Network Management (NMT). In order to enable reliable starting behavior the “safe-Operational” state has been introduced in EtherCAT. In this state valid entries are transmitted, while the outputs remain in safe status.

- **INIT:** it is the default state at power-on. No communication on the application layer is available. The Master has access only to the DL-information registers.
- **PRE-OPERATIONAL:** mailbox communication on the application layer is available, but no process data communication is available.
- **SAFE-OPERATIONAL:** mailbox communication on the application layer and process (input) data communication are available. Only inputs (Slave → Master) are evaluated, while outputs (Master → Slave) are kept in safe state.
- **OPERATIONAL:** process data inputs (Slave → Master) and outputs (Master → Slave) are valid.

### 3.3 Process Data Objects (PDO)

Process Data Objects manage the process data exchange (for example, the cyclic transmission of the position value). The process data are defined in the *PDO Mapping* and they are handled by the *Sync Manager*.

### 3.4 Service Data Objects (SDO)

Service Data Objects manage the parameter data exchange (for example the acyclic execution of the preset function). The SDO provides an efficient communication mechanism for parameter data of any size. A service data channel is formed between the configuration Master and the connected devices for this purpose. The device parameters can be written or read from the device object dictionary with a unique frame handshake.

### 3.5 Object dictionary

The following table provides a summary of all SDO objects supported by the encoder.

- **Object**            object address in hex format (index and sub-index)
- **Name**
- **Type**            data type
- **Attr**            access attributes: ro → read only, rw → read / write
- **Default**        default value on first init

Object	Name	Type	Attr	Default	Information
1000H	Device Type	Unsigned32	ro	0002 0196H	Byte0-1 = 0196H (Profile 406) Byte2-3 = 2 Absolute multiturn encoder
1008H	Device Name	Visible String	ro		Device name in ASCII
1009H	Hardware Version	Visible String	ro	"n.nn"	Hardware version in ASCII
100AH	Software Version	Visible String	ro	"n.nn"	Software version in ASCII
1011H	Restore Parameters	ARRAY			
00H	Max sub-index	Unsigned8	ro	01H	
01H	Restore all param.	Unsigned32	rw	0000 0001H	6461 6F6CH for restoring default values.
1018H	Identity	RECORD			
00H	Max sub-index	Unsigned8	ro	04H	
01H	Vendor ID	Unsigned32	ro	0000 02B4H	Manufacturer identifier (from ETG)
02H	Product code	Unsigned32	ro	0000 000AH	10 → absolute multiturn encoder
03H	Revision	Unsigned32	ro	0000 0100H	Current revision
04H	Serial Number	Unsigned32	ro	0000 0000H	Not used
1A00H	TxPDO 1 Mapping	ARRAY			
00H	Max sub-index	Unsigned8	ro	01H	
01H	1^ object in PDO1	Unsigned32	ro	6004 0020H	Object 6004.00, 32 bit length
1A01H	TxPDO 2 Mapping	ARRAY			
00H	Max sub-index	Unsigned8	ro	02H	
01H	1^ object in PDO2	Unsigned32	ro	6004 0020H	Object 6004.00, 32 bit length
02H	2^ object in PDO2	Unsigned32	ro	3030 0020H	Object 3030.00, 32 bit length
1A02H	TxPDO 3 Mapping	ARRAY			
00H	Max sub-index	Unsigned8	ro	02H	
01H	1^ object in PDO3	Unsigned32	ro	6004 0020H	Object 6004.00, 32 bit length
02H	2^ object in PDO3	Unsigned32	ro	6030 0010H	Object 6030.00, 16 bit length
1C00H	Sync Manager Type	ARRAY			
00H	Max sub-index	Unsigned8	ro	04H	
01H	SM0 type	Unsigned8	ro	01H	1: Receive Mailbox (Master → Slave)
02H	SM1 type	Unsigned8	ro	02H	2: Send Mailbox (Slave → Master)
03H	SM2 type	Unsigned8	ro	03H	3: PDO outputs (Master → Slave)
04H	SM3 type	Unsigned8	ro	04H	4: PDO inputs (Slave → Master)
1C12H	RxPDO Assign	ARRAY			
00H	Max sub-index	Unsigned8	ro	00H	No receiving PDO (no outputs)
1C13H	TxPDO Assign	ARRAY			
00H	Max sub-index	Unsigned8	rw	01H	
01H	TxPDO index	Unsigned16		1A00H	

Object	Name	Type	Attr	Default	Information
1C33H	Sync Man. Parameter	RECORD			
00H	Max sub-index	Unsigned8	ro	06H	
01H	Sync. Type	Unsigned16	ro	0022H	Synchronization mode setting
02H	Cycle Time	Unsigned32	ro	0000 0000H	Cycle time in ns
03H	Shift Time	Unsigned32	ro	0000 0000H	Shift time in ns
04H	Sync. supported	Unsigned16	ro	0007H	Supported modes
05H	Min Cycle Time	Unsigned32	ro	0000 0000H	Minimum cycle time in ns
06H	Calc and Copy Time	Unsigned32	ro	0000 0000H	Calc and copy time in ns
2005H	Speed Unit	Unsigned8	rw	00H	Speed measuring unit
3030H	Speed 32-bit	Integer32	ro		Rotation speed (32-bit value)
6000H	Operating parameters	Unsigned16	rw	0000H	Bit0 → Rotation direction 0 = clockwise 1 = counter clockwise Bit2 → Scaling function 0 = disabled 1 = enabled
6001H	Measuring Units per Revolution	Unsigned32	rw	0000 2000H	13 bit resolution; all values from 01H to 2000H are accepted.
6002H	Total Measuring Range	Unsigned32	rw	2000 0000H	29 bit resolution
6003H	Preset	Unsigned32	rw	0000 0000H	Preset value.
6004H	Position	Unsigned32	ro		Absolute position of the encoder, including offset from preset value.
6030H	Speed 16-bit	Integer16	ro		Rotation speed (16-bit value)
6500H	Operating Status	Unsigned16	ro	0000H	Bit0 → Rotation direction Bit2 → Scaling function
6501H	Single Turn Resolution	Unsigned32	ro	2000H	13 bit resolution
6502H	Number of revolutions	Unsigned32	ro	0001 0000H	Maximum number of distinguishable revolutions
6503H	Alarms	Unsigned16	ro	0000H	Bit0 → Position error
6504H	Supported Alarms	Unsigned16	ro	0001H	The following alarms are supported: Bit0 → Position error
6505H	Warnings	Unsigned16	ro	0000H	Bit4 → Low battery charge
6506H	Supported Warnings	Unsigned16	ro	0010H	The following warnings are supported: Bit4 → Low battery charge

### 3.6 Standard objects

#### 1000.00 Device Type

[Unsigned32, ro]

Default = 0002 0196H

absolute multiturn encoder, with encoder profile CiA DS-406.

#### 1008.00 Device Name

[Visible String, ro]

Default = "ELAP EtherCAT MEM-BUS"

It is the encoder name, in ASCII format.

#### 1009.00 Hardware Version

[Visible String, ro]

It is the encoder hardware version, in ASCII format "n.nn".

#### 100A.00 Software Version

[Visible String, ro]

It is the encoder software version, in ASCII format "n.nn".

#### 1011.01 Restore all Default Parameters

[Unsigned32, rw]

Default = 0000 0001H

the encoder supports the restoring of all default parameter values.

This object allows the user to restore the default values of all writable parameters.

In order to avoid restoring of parameter values by mistake, restoring is only executed when a specific signature is written to sub-index 1. The signature is "load", in ASCII format.

MSB		LSB	
d	a	o	l
64H	61H	6FH	6CH

By read access on sub-index 1, the device provides information about its restoring capability.

Note: All restored values are immediately used, but they are shown in the corresponding CoE objects only after a new power-on.

#### 1018.xx Identity

**01H** – [Unsigned32, ro] Vendor ID

Default = 0000 02B4H

Elap

**02H** – [Unsigned32, ro] Product code

Default = 0000 000AH

Absolute multiturn encoder, 29 bit resolution.

**03H** – [Unsigned32, ro] Revision

**04H** – [Unsigned32, ro] Serial Number



## 1A00.xx TxPDO 1 Mapping

### 01 – [Unsigned32, ro]

Default = 6004 0020H

This object defines a PDO telegram, including the encoder position value.

Position (object 6004.00)			
Byte0	Byte1	Byte2	Byte3

## 1A01.xx TxPDO 2 Mapping

### 01 – [Unsigned32, ro]

Default = 6004 0020H

### 02 – [Unsigned32, ro]

Default = 3030 0020H

This object defines a PDO telegram, including the encoder position value and the speed 32 bit value.

Position (object 6004.00)				Speed 32-bit (object 3030.00)			
Byte0	Byte1	Byte2	Byte3	Byte0	Byte1	Byte2	Byte3

## 1A02.xx TxPDO 3 Mapping

### 01 – [Unsigned32, ro]

Default = 6004 0020H

### 02 – [Unsigned32, ro]

Default = 6030 0010H

This object defines a PDO telegram, including the encoder position value and the speed 16 bit value.

Position (object 6004.00)				Speed 16-bit (object 6030.00)	
Byte0	Byte1	Byte2	Byte3	Byte0	Byte1

## 3.7 Synchronization objects

### 1C00.xx Sync Manager Communication Type

#### 01H – [Unsigned8, ro]

Default = 1 SM0 is mailbox receive (Master → Slave)

#### 02H – [Unsigned8, ro]

Default = 2 SM1 is mailbox send (Master ← Slave)

#### 03H – [Unsigned8, ro]

Default = 3 SM2 is PDO output (RxPDO, Master → Slave)

#### 04H – [Unsigned8, ro]

Default = 4 SM3 is PDO input (TxPDO, Master ← Slave)

### 1C12.00 RxPDO Assign

[Unsigned8, ro]

Default = 0 There are no output PDO.

## 1C13.01 TxPDO Assign

[Unsigned32, ro]

Default = 0000 1A00H Input PDO are mapped into the object 1A00H.

## 1C33.xx Sync Manager Parameter

### 01H – [Unsigned16, ro] Synchronization Type

Default = 0022H

0000H → Free Run (not synchronized)

0022H → Synchronous with SM3 event

0002H → Synchronous DC-SYNCO (distributed clock mode)

### 02H – [Unsigned32, ro] Cycle Time in ns

Free Run → Time between two local timer events in ns

Synchronous with SM3 → Minimum time between two SM events in ns

Synchronous DC-SYNCO → SYNCO cycle time in ns

### 03H – [Unsigned32, ro] Shift Time in ns

Time between related event and the associated action (inputs latched from hardware) in ns.

### 04H – [Unsigned16, ro] Synchronization types supported

Default = 0007H

Bit0 = 1 → Free Run supported

Bit1 = 1 → Synchronization with SM-Event supported

Bit2 = 1 → Synchronization with DC-SYNCO supported

### 05H – [Unsigned32, ro] Minimum Cycle Time in ns

Maximum duration time of the local cycle in ns. It is used in Synchronous or DC mode.

### 06H – [Unsigned32, ro] Calc and Copy Time in ns

Time needed by the application controller to perform calculations on the input values and to copy the process data from the local memory to the Sync Manager before the data is available for EtherCAT. Used in DC mode.

### 3.8 Manufacturer and Profile Specific Objects (CiA DS-406)

#### 2005.00 Speed Unit

[Unsigned8, rw]

Default = 0

This object indicates the measuring unit whereby the speed is calculated and output (see objects 3030.00 and 6030.00).

Selectable values:

- 0 → pulse / sec
- 1 → pulse / 100 ms
- 2 → pulse / 10 ms
- 3 → RPM

#### 3030.00 Speed 32-bit

[Integer32, ro]

This object shows the output speed value, with 32 bit resolution. Its measuring unit is set in object 2005.00.

#### 6000.00 Operating Parameters

[Unsigned16, rw]

Default = 0

Bit0 → Rotation direction, seen from the shaft side.

- Bit0 = 0 → Clockwise.  
The position value increases with a clockwise rotation.
- Bit0 = 1 → Counter clockwise.  
The position value increases with a counter clockwise rotation.

Bit2 → Scaling function.

- Bit2 = 0 → Disabled.  
Physical resolution is used for calculations (13 bit for single turn and 16 bit for number of revolutions).
- Bit2 = 1 → Enabled.  
Resolution set in objects 6001.00 and 6002.00 is used for calculations.

#### 6001.00 Measuring Units per Revolution

[Unsigned32, rw]

Default = 0000 2000H (8192)

This object sets the number of pulses for a single turn. It is active when the scaling function is enabled (that is Bit2 = 1 in object 6000.00). Values lower or equal to physical resolution (that is 8192 pulses per turn) are accepted.

**Note:** When the scaling parameters change, it is recommended to execute a Preset function (writing of parameter 6003.00) in order to reset the encoder and restart from a correct position.

## 6002.00 Total Measuring Range

[Unsigned32, rw]

Default = 2000 0000H (536870912)

This object sets the total measuring range for position values. It is active when the scaling function is enabled (that is Bit2 = 1 in object 6000.00).

The setting value is calculated as:

Measuring Units per Revolution (6001.00) x Number of Revolutions

Values lower or equal to the global physical resolution (that is 536870912) are accepted.

Note: When the scaling parameters change, it is recommended to execute a Preset function (writing of parameter 6003.00) in order to reset the encoder and restart from a correct position.

## 6003.00 Preset

[Unsigned32, rw]

Default = 0000 0000H

This object sets the Preset value. The Preset function, that is executed when writing the 6003.00 object, is used to adjust the measuring system to any position value within a range to the total measuring length - 1.

Note:

1. When the scaling parameters change, it is recommended to execute a Preset function in order to reset the encoder and restart from a correct position.
2. When the scaling function is disabled (Bit2 = 0 in object 6000.00), the Preset value shall be lower than the global physical resolution (536870912).
3. When the scaling function is enabled (Bit2 = 1 in object 6000.00), the Preset value shall be lower than the total resolution written in object 6002.00.

## 6004.00 Position

[Unsigned32, ro]

This object shows the output position value of the encoder, that is eventually modified by the scaling parameters (see objects 6000.00, 6001.00 and 6002.00).

## 6030.00 Speed 16-bit

[Integer16, ro]

This object shows the output speed value, with 16 bit resolution. Its measuring unit is set in object 2005.00.

## 6500.00 Operating Status

[Unsigned16, ro]

- Bit0 → Information about rotation direction is given (0 → clockwise, 1 → counter clockwise). Object 6000.00 shall be used to set the counting direction.
- Bit2 → Information about scaling function is given (0 → Disabled, 1 → Enabled). Object 6000.00 shall be used to enable/disable the scaling function.

## 6501.00 Single Turn Resolution

[Unsigned32, ro]

Default = 0000 2000H (8192)

This Object contains the number of measuring steps per revolution which can be output by the measuring system. See object 6001.00 in order to set a specific resolution.

## 6502.00 Number of Distinguishable Revolutions

[Unsigned32, ro]

Default = 0001 0000H (65536)

This object contains the number of distinguishable revolutions that the measuring system can output.

The global physical resolution of the Elap encoder is calculated as follows

$$\text{Single Turn Resolution} \times \text{Number of revolutions} = 8192 \times 65536 = 536870912$$

See objects 6001.00 and 6002.00 in order to set a specific total resolution.

## 6503.00 Alarms

[Unsigned16, ro]

Default = 0000H

Additionally to the emergency message, this object provides further alarms. An alarm is set when a malfunction in the measuring system could lead to an incorrect position value. If an alarm occurs, the corresponding bit is set to logical high until the alarm is cleared and the measuring system is able to provide an accurate position value again.

## 6504.00 Supported Alarms

[Unsigned16, ro]

Default = 0001H

This object provides information about the alarms that are supported by the measuring system.

Bit0 → Position error.

## 6505.00 Warnings

[Unsigned16, ro]

Default = 0000H

This objects provides the warnings and indicate that tolerance for certain internal parameters of the encoder have been exceeded. In contrast to alarms and Emergency messages, warnings do not imply incorrect position values.

## 6506.00 Supported Warnings

[Unsigned16, ro]

Default = 0010H

This object provides information about the warnings that are supported by the encoder.

Bit4 → Battery charge too low.

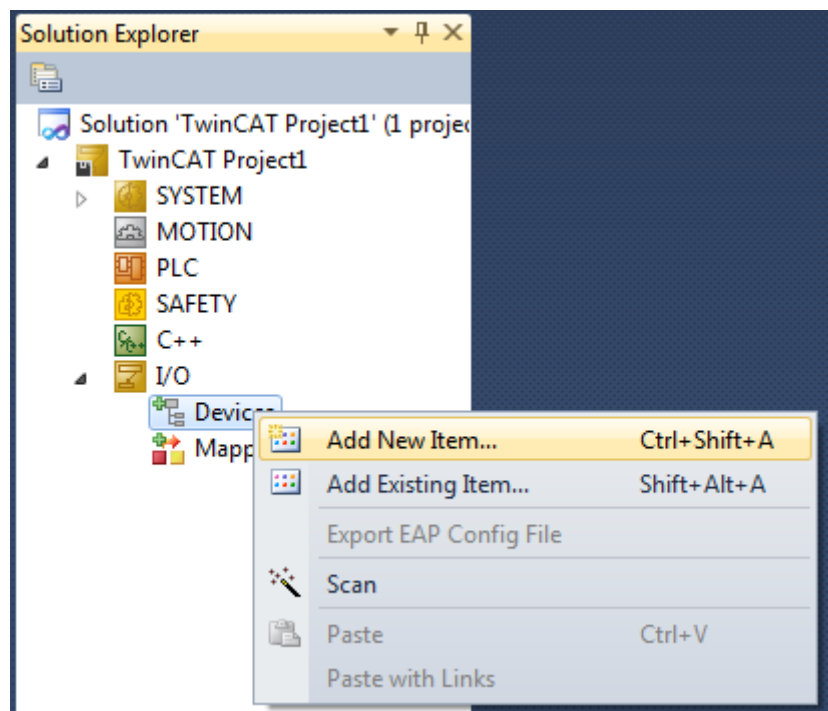
## 4 System set up with TwinCAT® 3 (Beckhoff software system)

In this chapter a short introduction for encoder device operation with a generic office PC environment is described. First of all, the XML file provided by Elap shall be copied into the directory C:\TwinCAT\3.1\Config\Io\EtherCAT.

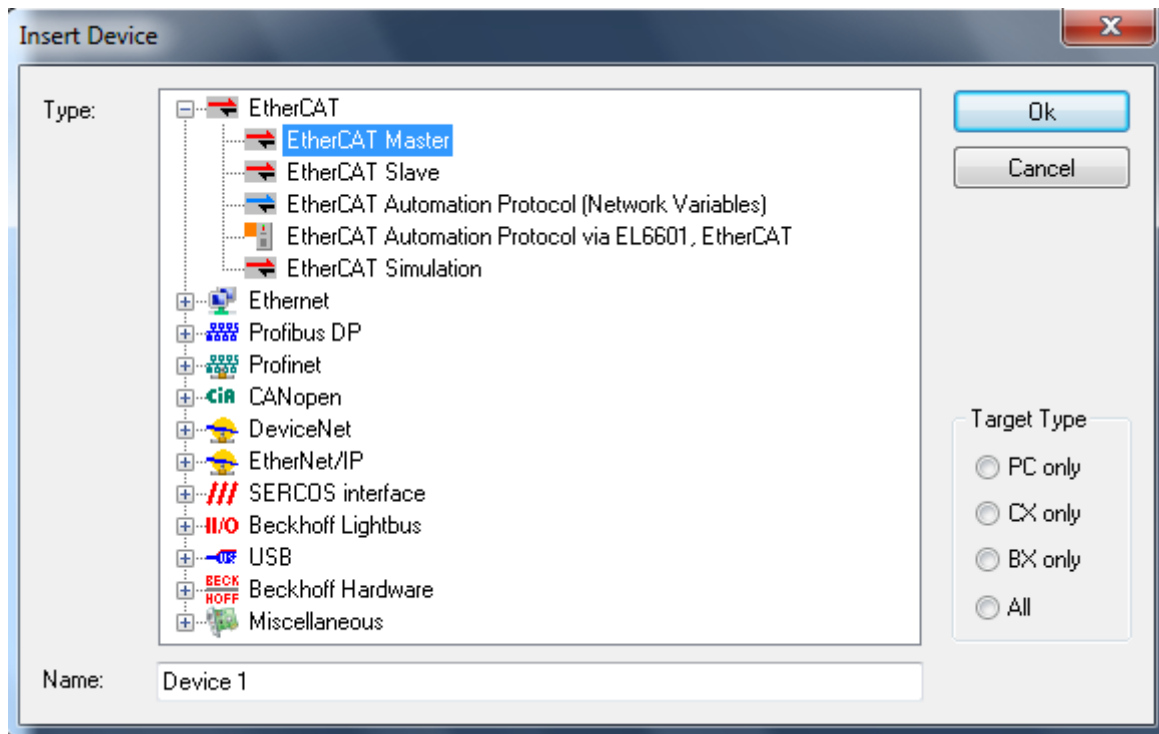
### 4.1 Setting up the network adapter

Start the *TwinCAT® System Manager* software and create a new project.

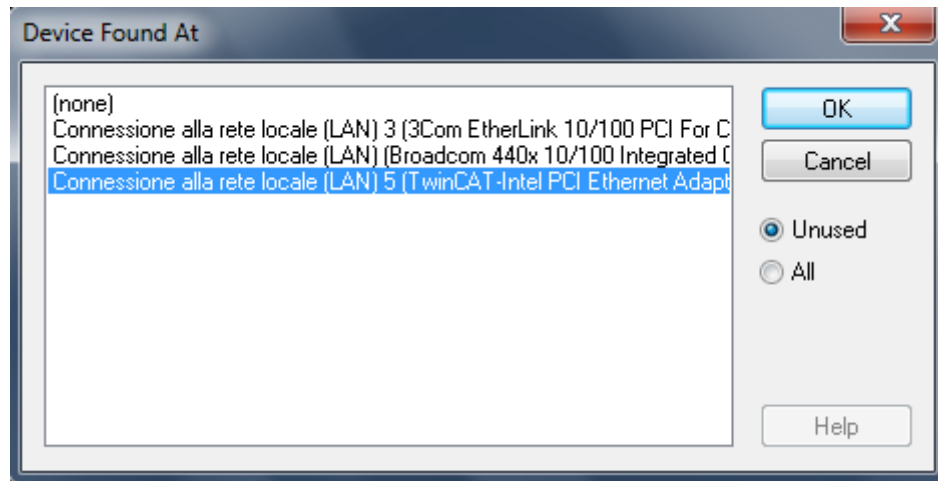
Open the **I/O** tree in the “Solution Explorer” window and select **Devices**; click the right button of the mouse and select the option **Add New Item...**



The “Insert Device” window appears. Select “EtherCAT Master” and then press OK.



Select the network card to use.

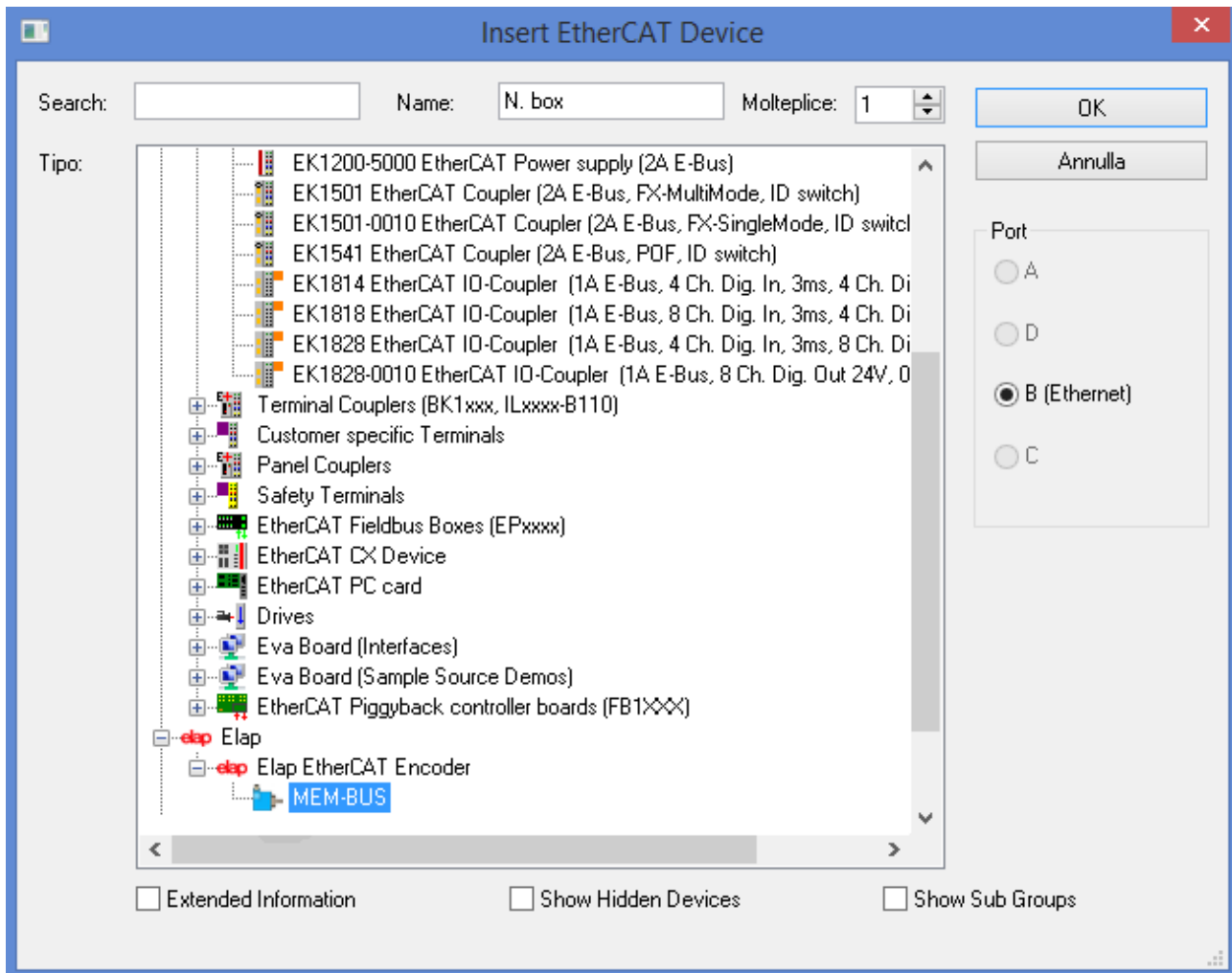


**Note:** If no network card is available you have to install one. This is possible under **TwinCAT** → **Show Real Time Ethernet Compatible Devices...** → **Install** “Compatible devices”. After this command, your network card should move from “Compatible devices” to “Installed and ready to use devices”. Confirm dialog with OK.

## 4.2 Inserting Input / Output Modules (Box)

If Elap encoder is connected to the PC, you can click your mouse right button on **Device 1 (EtherCAT)** and select the command **Scan Boxes**, otherwise select the command **Add New Item...**

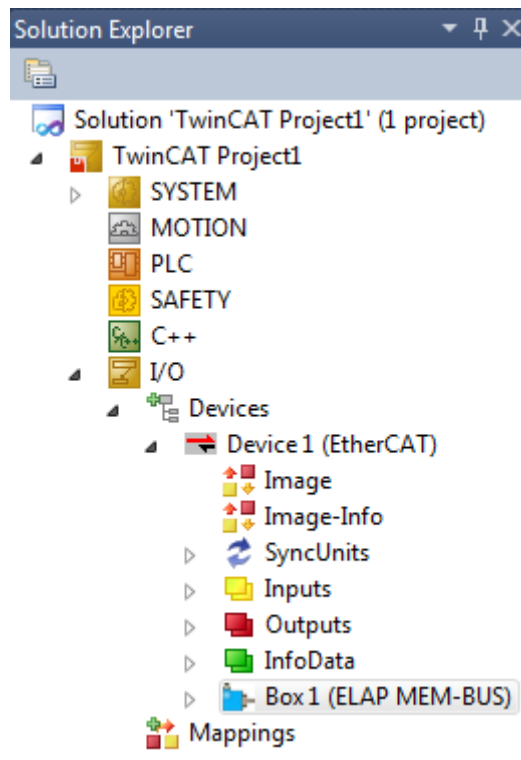
Select Elap encoder MEM-BUS in the window and press the OK button.





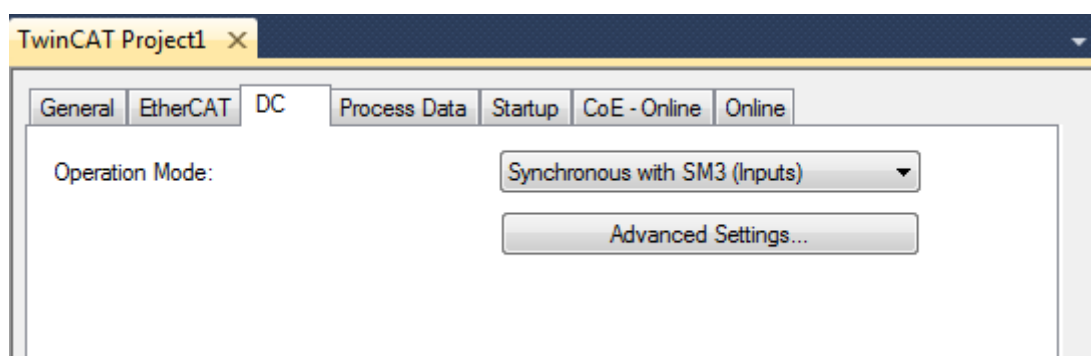
### 4.3 Setting up working mode

Select Elap MEM-BUS encoder in the “Solution Explorer” window and double click your mouse.



The main window for Box configuration appears. Select DC table.

See section 1.6 for further details about working modes.



The default working mode is Synchronous with SM3 Event.

In order to work in real time mode, with higher performance, select **Synchronous with DC SYNC0** and then press **Advanced Settings...**

In **SYNC0** section, set the cycle time value in box **Sync Unit Cycle**; this time value is a multiplier for the value specified into the **Sync Unit Cycle** field (in micro-seconds), near the higher right corner.

**Advanced Settings**

**Distributed Clock**

Cyclic Mode  
 Modo operativo: Synchronous with DC (SYNC0)   
☒ Enable Sync Unit Cycle (µs): 4000

SYNC 0  
 Cycle Time (µs):  
☒ Sync Unit Cycle x 1 4000  
☐ User Defined  
☒ Enable SYNC 0

Shift Time (µs):  
 User Defined 0  
 + SYNC0 Cycle x 0 0  
☐ Based on Input Reference  
 +  
 = 0

SYNC 1  
☐ Sync Unit Cycle  
☒ SYNC 0 Cycle x 1  
☐ Enable SYNC 1

☐ Use as potential Reference Clock

OK Annulla

## 4.4 PDO Assignment

Select the **Process Data** table from the window dedicated to the encoder configuration.

In **PDO Assignment (0x1C13)** section it is possible to choose the appropriate process data for the application.

Generale EtherCAT DC Process Data Avvio

Sync Manager:

SM	Size	Type	Flags
0	128	MboxOut	
1	128	MboxIn	
2	0	Outputs	
3	4	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	4.0	TxPDO 1		3	0
0x1A01	8.0	TxPDO 2			0
0x1A02	6.0	TxPDO 2			0

PDO Assignment (0x1C13):

☒ 0x1A00  
☐ 0x1A01 (excluded by 0x1A00)  
☐ 0x1A02 (excluded by 0x1A00)

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Defa
0x6004:00	4.0	0.0	Position Value	UDINT	
	4.0				

Download

☒ PDO Assignment  
☐ PDO Configuration

Predefined PDO Assignment: (nessuno)

Load PDO info from device

Sync Unit Assignment...

**PDO Content** section shows data distribution in the telegram selected. In particular:

- Telegram 0x1A00

Position (object 6004.00)			
Byte0	Byte1	Byte2	Byte3

- Telegram 0x1A01

Position (object 6004.00)				Speed 32-bit (object 3030.00)			
Byte0	Byte1	Byte2	Byte3	Byte0	Byte1	Byte2	Byte3

- Telegram 0x1A02

Position (object 6004.00)				Speed 16-bit (object 6030.00)	
Byte0	Byte1	Byte2	Byte3	Byte0	Byte1

## 4.5 CoE Object dictionary

Select the **CoE-Online** table from the window dedicated to the encoder configuration.

Index	Nome	Flags	Valore
1000	Device Type	M RO	0x00020196 (131478)
1008	Device Name	RO	ELAP EtherCAT MEM-BUS
1009	Hardware Version	RO	1.00
100A	Software Version	RO	1.00
1011:0	Restore parameters	RO	
1018:0	Identity Object	M RO	> 4 <
1A00:0	TX PDO 1	RO	> 1 <
1A01:0	TX PDO 2	RO	> 2 <
1A02:0	TX PDO 3	RO	> 2 <
1C00:0	Sync Manager	M RO	> 4 <
1C12:0	Sync Manager 2 RXPDO Assignment	R/W	
1C13:0	Sync Manager 3 TXPDO Assignment	R/W	
1C33:0	Sync Manager 3 Synchronization	RO	> 6 <

It is possible to read specific values (Device Type, Software Version, etc.) and to change the encoder working parameters (rotation direction, single turn resolution and measuring range).

## TECHNICAL SPECIFICATIONS

MECHANICAL & ENVIRONMENTAL SPECIFICATIONS		
	MEM-BUS 620 / 520 / 540	MEM-BUS 440 / 450
Materials:	case shaft	Aluminium Stainless steel
Weight		500 g
Shaft/joint hole Ø	6, 8, 10 mm	8, 10, 12, 14, 15 mm
Revolutions/minute		6000
Starting torque		≤ 0.8 Ncm
Inertia		≤ 25 g cm <sup>2</sup>
Max load		80 N axial/100 N radial
Vibrations resistance (10÷2000 Hz)		100 m/sec <sup>2</sup>
Shock (11 ms)		50 G
Protection degree		IP67 – IP65 shaft side
Operating temperature		-30 ÷ 70°C
Stocking temperature		-30 ÷ 85°C

ELECTRICAL & OPERATING SPECIFICATIONS	
Operating principle	Magnetic
Resolution/revolution	8192 steps/rev – 13 bit
Revolutions no.	65536 - 16 bit
Initializing time	< 1 s
Data memory	> 20 years
Field bus	No motion – Power off
Supply	EtherCAT®
Power consumption	10 ÷ 30 Vdc
Accuracy	Protection against polarity reversal
Connection	2 W
Interference immunity	± ½ LSB
Emitted interference	2 M12 female connectors +1 M12 male connector
	EN 61000-6-2
	EN 61000-6-4

EtherCAT® IEC 61158-1-6 & IEC 61784-2	
Physical Layer	EtherCAT® 100Base-TX, Fast Ethernet, ISO/IEC 8802-3
Output code	Binary
Device profile	CANOpen over EtherCAT (CoE), CiA DS-406
Cycle time	≥ 1 ms
Transmission rate	100 Mbit/s
Transmission	CAT-5 cable, shielded (STP), ISO/IEC 11801
Programmable via EtherCAT	<ul style="list-style-type: none"> <li>○ Counting direction</li> <li>○ Measuring steps per revolution</li> <li>○ Total measuring length in steps</li> <li>○ Preset value</li> <li>○ Speed resolution</li> </ul>

## ORDERING INFORMATION

MEM520B	ECT	M	10
TYPE	<u>INTERFACE</u> ECT = EtherCAT	<u>No. of TURNS</u> M = Multiturn	<u>SHAFT/JOINT HOLE Ø</u> Shaft: 6 – 8 – 10 mm Joint: 8 – 10 – 12 – 14 – 15 mm

**MEM520-Bus** = Round flange Ø 58 mm

**MEM540-Bus** = Round flange Ø 58 mm

**MEM620-Bus** = Square flange 63.5x63.5 mm

**MEM440-Bus** = Blind hollow shaft for motor coupling

**MEM450-Bus** = Blind hollow shaft, fixing by elastic support

# Certificate

## EtherCAT Conformance Test

### ELAP s.r.l.

EtherCAT Technology Group hereby confirms the above named company that the following device is successfully **EtherCAT Conformance Tested**.

#### Device under Test

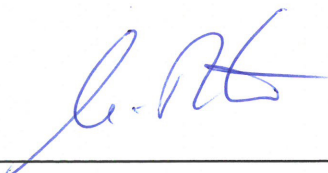
Product Name: MEM-BUS  
Product Code: 0xA  
Revision Number: 0x100

Assigned Vendor ID: 0x2B4  
Test Report Number: 0x2B4\_001  
EtherCAT Test Center: Beckhoff Automation, Nuremberg, Germany

#### Supported features tested:

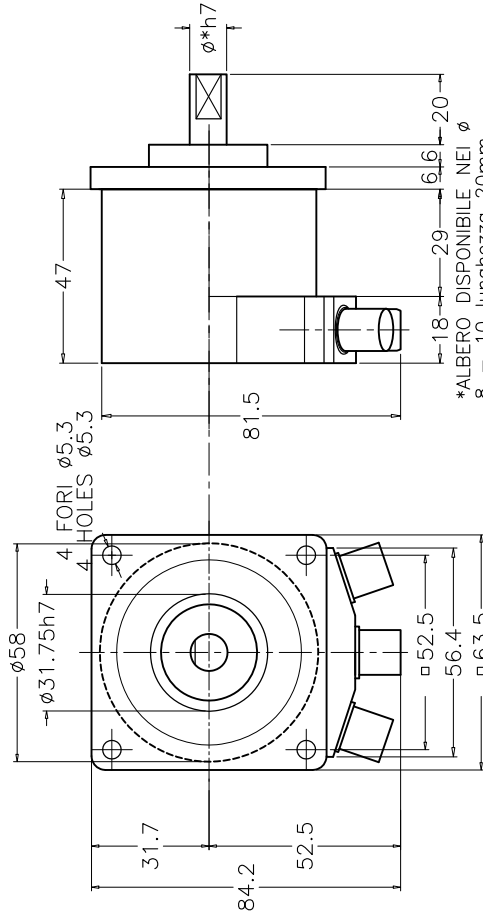
- ✓ Conformance Test Tool 2.1.0.0
- ✓ EtherCAT State Machine
- ✓ Indicator and Labeling
- ✓ Distributed Clocks
- Explicit Device ID
- ✓ CoE Mailbox Protocol
- CiA402 Profile
- Semi Device Profile

Nuremberg, March 02, 2017



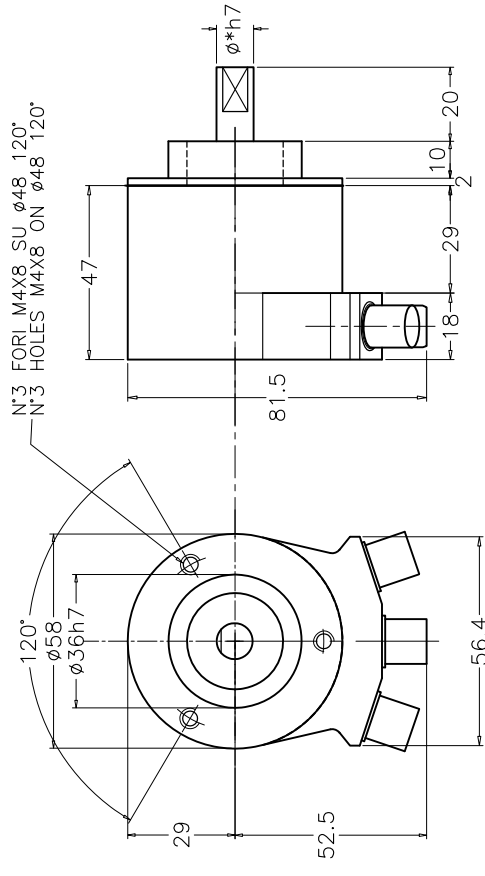
Martin Rostan, Executive Director  
EtherCAT Technology Group

MEM620Bus



\*ALBERO DISPONIBILE NEI Ø  
8 - 10 lunghezza 20mm  
6 lunghezza 10mm  
\* AVAILABLE SHAFT DIAMETERS  
8 - 10 length 20mm  
6 length 10mm

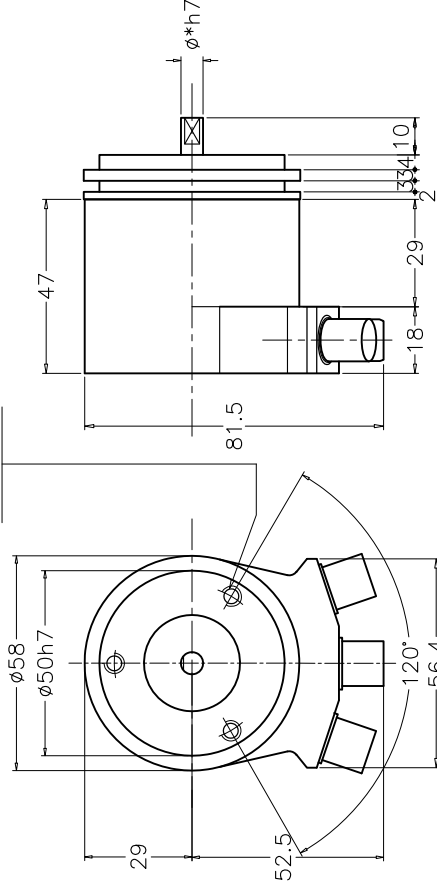
MEM540Bus



\*ALBERO DISPONIBILE NEI Ø  
8 - 10 lunghezza 20mm  
6 lunghezza 10mm  
\* AVAILABLE SHAFT DIAMETERS  
8 - 10 length 20mm  
6 length 10mm

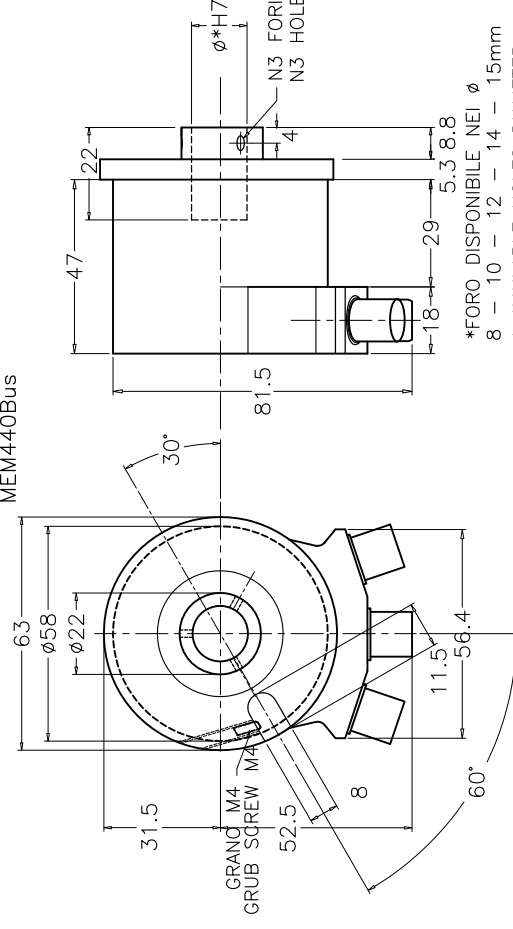
MEM520Bus

3 FORI M4X8 SU Ø42  
3 HOLES M4X8 Ø42



\*ALBERO DISPONIBILE NEI Ø  
8 - 10 lunghezza 20mm  
6 lunghezza 10mm  
\* AVAILABLE SHAFT DIAMETERS  
8 - 10 length 20mm  
6 length 10mm

MEM440Bus

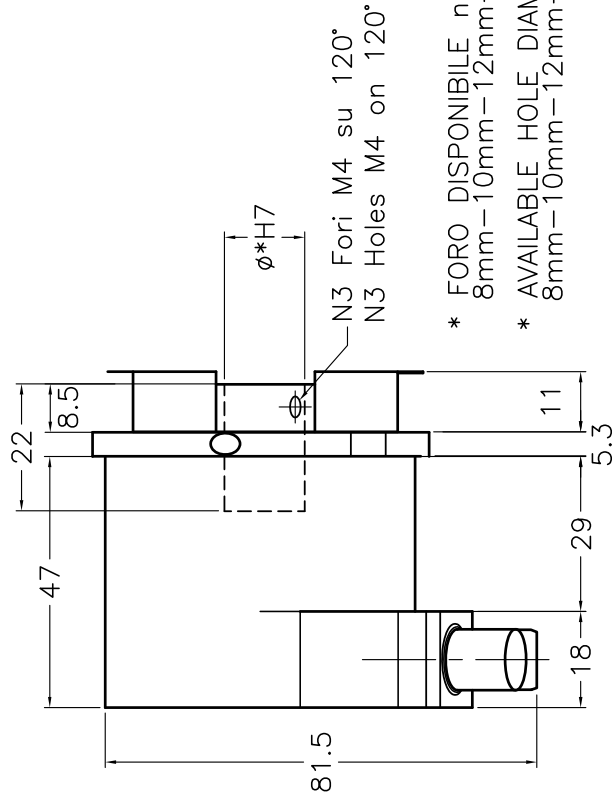
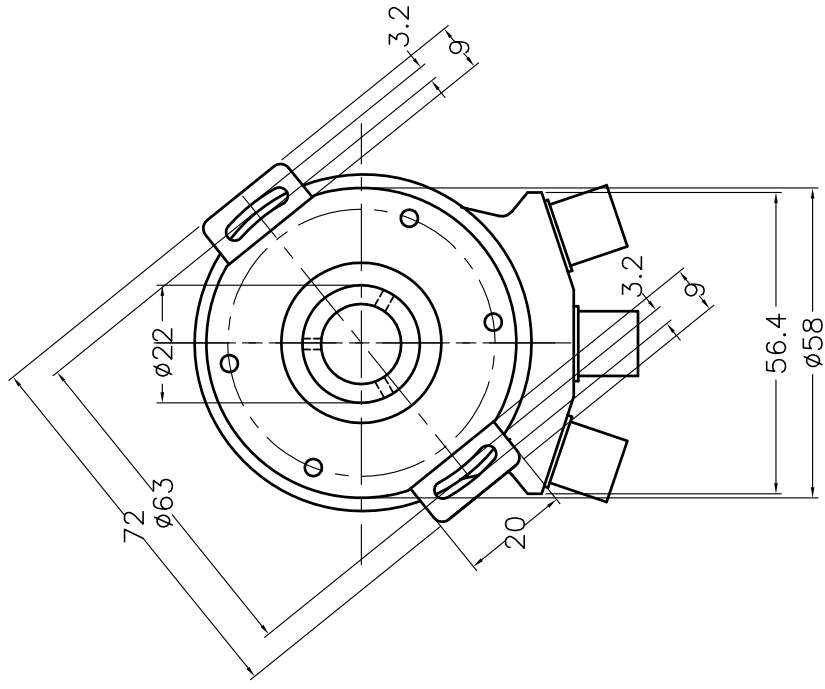


\*FORO DISPONIBILE NEI Ø  
8 - 10 - 12 - 14 - 15mm  
\* AVAILABLE HOLES DIAMETER  
8 - 10 - 12 - 14 - 15mm

DISEGNO N.	SCALA	FOGLIO	SOSTITUITO DA	DATA	DESSINATO	DATA	SOFTWARE
M1551B	1:1	1 di 1			T.R.		
REVISIONE	TOLL.	MATER.	SOSTITUISCE	DATA	FRMA	FILE	
	V.rote						
DESCRIZIONE	DATA	FRMA					
MEM BUS INT/EXT DIMENSIONI ENCODER PROFINET ETHERCAT	Albero mem bus 440 H8.8 16.1.17	5.7.16	7.2				
NOTE	Profinet ed Ethercat						
CLIENTE							

elap

VIA VITTORIO VENETO, 4  
20094 CORSECO (Milano)  
TEL. 024518561 FAX. 0245103408



\* FORO DISPONIBILE nei diametri  
8mm-10mm-12mm-14mm-15mm

\* AVAILABLE HOLE DIAMETERS  
8mm-10mm-12mm-14mm-15mm

DISEGNO N.	CODICE	SCALA	FOLIO	SOSTITUITO DA	DATA	FIRMA	DISEGNATO	DATA	SOFTWARE
M1553B		1:1	1 di 1				T.R.		
DENOMINAZIONE	MEM450-PNT ECT	TOLL.	MATER.	SOSTITUISCE	DATA	FIRMA	VISTA	FILE	
Dimensione d'ingombro		V.note	Vari					M1553A.DWG	
Profinet Ethercat		DESCRIZIONE			DATA	FIRMA			
NOTE		Corretto			6.7.16	99			
Tolleranza gen.li ISO 2768 - m									
CLIENTE									

elap  
SRL

VIA VITTORIO VENETO, 4  
20094 CORSO (MI)  
TEL. 02/4519561 R.A. FAX. 02/45103146