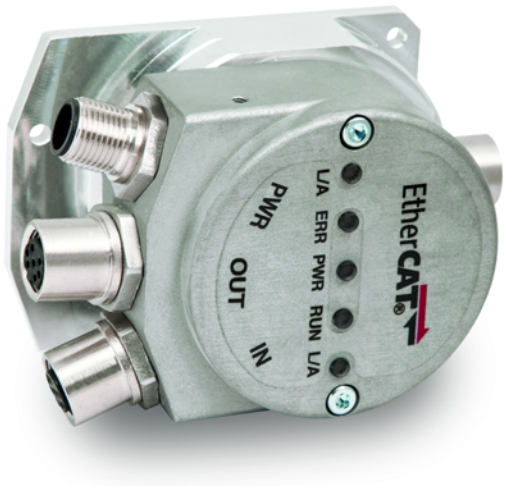
Smart encoders & actuators

User's guide

IF56-ROT-EC
IF56-LIN-EC



EtherCAT®

in compliance with ETG.1000

- SSI and BiSS to EtherCAT converter
- Suitable for SSI and BiSS rotary and linear encoders
- Singleturn resolution up to 18 bit; total resolution up to 30 bit
- Complies with ETG.1000 specifications
- Implements CoE, FoE, & EoE protocols and the EtherCAT State Machine
- M12 connectors

Suitable for the following models:

- IF56-ROT-EC
- IF56-LIN-EC

General Contents

This manual is periodically reviewed and revised. As required we suggest checking if a new or updated edition of this document is available at Lika Electronic s.r.l.'s website. Lika Electronic s.r.l. assumes no responsibility for any errors or omissions in this document. Critical evaluation of this manual by the user is welcomed. Your comments assist us in preparation of future documentation, in order to make it as clear and complete as possible. Please send an e-mail to the following address info@lika.it for submitting your comments, suggestions and criticisms.

General contents

Subject Index

Typographic and iconographic conventions

In this guide, to make it easier to understand and read the text the following typographic and iconographic conventions are used:

- parameters and objects both of the device and the interface are coloured in GREEN;
- alarms are coloured in RED;
- states are coloured in FUCSIA.

When scrolling through the text some icons can be found on the side of the page: they are expressly designed to highlight the parts of the text which are of great interest and significance for the user. Sometimes they are used to warn against dangers or potential sources of danger arising from the use of the device. You are advised to follow strictly the instructions given in this guide in order to guarantee the safety of the user and ensure the performance of the device. In this guide the following symbols are used:

	This icon, followed by the word **WARNING**, is meant to highlight the parts of the text where information of great significance for the user can be found: user must pay the greatest attention to them! Instructions must be followed strictly in order to guarantee the safety of the user and a correct use of the device. Failure to heed a warning or comply with instructions could lead to personal injury and/or damage to the unit or other equipment.
	This icon, followed by the word **NOTE**, is meant to highlight the parts of the text where important notes needful for a correct and reliable use of the device can be found. User must pay attention to them! Failure to comply with instructions could cause the equipment to be set wrongly: hence a faulty and improper working of the device could be the consequence.
	This icon is meant to highlight the parts of the text where suggestions useful for making it easier to set the device and optimize performance and reliability can be found. Sometimes this symbol is followed by the word **EXAMPLE** when instructions for setting parameters are accompanied by examples to clarify the explanation.

Preliminary information

This guide is designed to provide the most complete and exhaustive information the operator needs to correctly and safely install and operate the following **SSI / BiSS to EtherCAT gateways of the IF56 series**:

IF56-ROT-EC **(suitable for rotary encoders with singleturn resolution up to 18 bits, see the Singleturn resolution (bits) parameter of the 2201-00 Rotary Encoder Resolution object; and multiturn resolution -number of revolutions- up to 14 bits, see the Multiturn resolution (bits) parameter of the 2201-00 Rotary Encoder Resolution object**

IF56-LIN-EC **(suitable for linear encoders with resolution 1,000,000 nm -1 mm- down to 1 nm, see the 2201-00 Linear Encoder Resolution object; max. number of bits: 32, see the Max No of Information (bit) L parameter of the 2200-00 Linear Encoder Settings object)**

The **IF56** series gateways allow the **integration of SSI and BiSS encoders**, both rotary and linear, **into industrial Ethernet networks**. They offer completely new hardware with advanced technology and a simple, fully compliant configuration into the widest range of industrial Ethernet networks: **Profinet**, **EtherNet/IP**, **EtherCAT**, **POWERLINK**, **MODBUS-TCP**, and **CC-LINK**. For the integration of SSI encoders, both rotary and linear, into conventional fieldbuses (Profibus, CANopen, and DeviceNet), see the IF55 series gateways.
The present manual is specifically designed to describe the SSI/BiSS to EtherCAT IF56 model for rotary and linear encoders (ordering codes: IF56-ROT-EC and IF56-LIN-EC).
For information on the gateways designed for the integration with other protocols (for example: SSI/BiSS to Profinet: ordering codes IF56-ROT-PT and IF56-LIN-PT; SSI/BiSS to EtherNet/IP: ordering codes IF56-ROT-EP and IF56-LIN-EP; etc.), refer to the specific documentation.

Please note that the present manual does not prescind from the user's guide of the SSI or BiSS encoder the gateway has to be connected to. Please read carefully the encoder's documentation before installing, connecting, and operating the measuring system.

For technical specifications please refer to the technical catalogue.

To make it easier to read the text, this guide is divided into four main sections.
In the first section general information concerning the safety, the mechanical installation and the electrical connection are provided.
In the second section information on how to install and configure the converter in TwinCAT development environment from Beckhoff as well as tips for setting up and running properly and efficiently the unit are provided.
In the third section, entitled **EtherCAT Interface**, both general and specific information is given on the EtherCAT interface. In this section the interface features and the objects implemented in the unit are fully described.
In the fourth section the Integrated Web Server is described.

Glossary of EtherCAT terms

EtherCAT, like many other networking systems, has a set of unique terminology. Table below contains a few of the technical terms used in this guide to describe the EtherCAT interface. They are listed in alphabetical order.

Acknowledge telegram (AT)	Telegram, in which each Slave inserts its data.
Actual value	Value of a variable at a given instant.
Algorithm	Completely determined finite sequence of operations by which the values of the output data can be calculated from the values of the input data.
Application	Function or data structure for which data is consumed or produced. Software functional element specific to the solution of a problem in industrial-process measurement and control.
Application class	Configuration of a Drive Object with a set of functional objects and supported by standard telegrams.
Application mode	Type of application that can be requested from a PDS.
Application objects	Multiple object classes that manage and provide a run time exchange of messages across the network and within the network device.
Application process	Part of a distributed application on a network, which is located on one device and unambiguously addressed.
Application relationship	Cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities.
Attribute	Description of an externally visible characteristic or feature of an object, property or characteristic of an entity. The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behaviour of an object. Attributes are divided into class attributes and instance attributes.
Axis	Logical element inside an automation system (e.g. a motion control system) that represents some form of movement.
Basic Slave	Slave device that supports only physical addressing of data.
Behaviour	Indication of how an object responds to particular events.
Bit	Unit of information consisting of a 1 or a 0. This is the smallest data unit that can be transmitted.

CANopen	Application layer protocol as defined in EN 50325-4.
Channel	Representation of a single physical or logical management object of a Slave to control conveyance of data.
CIP™	Common Industrial Protocol (see IEC 61158 Type 2, IEC 61784-1 and IEC 61784-2 CPF2).
Class	Description of a set of objects that share the same attributes, operations, methods, relationships, and semantics.
Client	Object which uses the services of another (Server) object to perform a task. Initiator of a message to which a Server reacts.
Clock synchronization	Representation of a sequence of interactions to synchronize the clocks of all time receivers by a time Master.
Commands	Set of commands from the application control program to the PDS to control the behaviour of the PDS or functional elements of the PDS.
Communication cycle	Accumulation of all telegrams between two Master synchronization telegrams.
Communication object	Component that manages and provides a run time exchange of messages across the network.
Connection	Logical binding between two application objects within the same or different devices.
Consume	Act of receiving data from a provider.
Consumer	Node or sink receiving data from a provider.
Control	Purposeful action on or in a process to meet specified objectives.
Control device	Physical unit that contains – in a module/subassembly or device – an application program to control the PDS.
Control unit	Control device.
Control word	Two adjacent bytes inside the Master data telegram containing commands for the addressed drive.
Controller	Controlling device which is associated with one or more drives (axes) a host for the overall automation.
Conveyance path	Unidirectional flow of APDUs across an application relationship.
Cycle time	Time span between two consecutive cyclically recurring events.
Cyclic	Events which repeat in a regular and repetitive manner.
Cyclic data	Part of the telegram which does not change its meaning during cyclic operation of the interface. High priority real-time data that is transferred by a CIP Motion connection on a periodic basis.
Data	Generic term used to refer to any information carried over a

	fieldbus.
Data consistency	Means for coherent transmission and access of the input-or output-data object between and within Client and Server.
Data exchange	Demand dependent; non cyclic transmission (service channel).
Data type	Relation between values and encoding for data of that type according to the definitions of IEC 61131-3. Set of values together with a set of permitted operations.
Data type object	Entry in the object dictionary indicating a data type.
Default gateway	Device with at least two interfaces in two different IP subnets acting as router for a subnet.
Device	Field device. Networked independent physical entity of an industrial automation system capable of performing specified functions in a particular context and delimited by its interfaces. Entity that performs control, actuating and/or sensing functions and interfaces to other such entities within an automation system. Physical entity connected to the fieldbus composed of at least one communication element (the network element) and which may have a control element and/or a final element (transducer, actuator, etc.).
Device profile	Collection of device dependent information and functionality providing consistency between similar devices of the same device type. Representation of a device in terms of its parameters and behaviour according to a device model that describes the device's data and behaviour as viewed through a network, independent from any network technology.
Diagnosis information	All data available at the Server for maintenance purposes.
Distributed clocks	Method to synchronize Slaves and maintain a global time base.
DL	Data-link-layer.
DLPDU	Data-link-protocol-data-unit.
Drive Object	Functional element of a Drive Unit.
Drive Unit	Logical device which comprises all functional elements related to one central processing unit.
Error	Discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition.
Error class	General grouping for related error definitions and corresponding error codes.
Error code	Identification of a specific type of error within an error class.
EtherCAt State Machine	EtherCAT Slave is a state machine; communication and operating characteristics depend on the current state of the

	device.
Event	Instance of a change of conditions.
Event data	Medium priority real-time data that is transferred by a CIP Motion connection only after a specified event occurs.
Feed forward	Command value used to compensate the lag in the control loop.
Feedback variable	Variable which represents a controlled variable and which is returned to a comparing element.
Fieldbus memory management unit	Function that establishes one or several correspondences between logical addresses and physical memory.
Fieldbus memory management unit entity	Single element of the fieldbus memory management unit: one correspondence between a coherent logical address space and a coherent physical memory location.
Frame	Denigrated synonym for DLPDU.
FreeRun	Asynchronous communication mode.
Full Slave	Slave device that supports both physical and logical addressing of data.
Functional element	Entity of software or software combined with hardware, capable of accomplishing a specified function of a device.
HMI	Human Machine Interface.
Host	Device that covers the automation functionality of an automation device.
I/O data	Input data and output data that would typically need to be updated on a regular basis (e.g. periodic change of state), such as commands, set-points, status and actual values.
Identification number (IDN)	Designation of operating data under which a data block is preserved with its attribute, name, unit, minimum and maximum input values, and the data.
Index	Address of an object within an application process.
Input data	Data transferred from an external source into a device, resource or functional element.
Interface	Shared boundary between two entities defined by functional characteristics, signal characteristics, or other characteristics as appropriate.
Little endian	Data representation of multi-octet fields where the least significant octet is transmitted first.
Logical power drive system	Model which includes PDS and communication network accessible through the generic PDS interface.
Mapping	Correspondence between two objects in that way that one object is part of the other object.
Mapping parameters	Set of values defining the correspondence between application objects and process data objects.

Master	Device that controls the data transfer on the network and initiates the media access of the Slaves by sending messages and that constitutes the interface to the control system. Node, which assigns the other nodes the right to transmit.
Master data telegram (MDT)	Telegram, in which the Master inserts its data.
Medium	Cable, optical fibre or other means by which communication signals are transmitted between two or more points.
Message	Ordered series of octets intended to convey information. Normally used to convey information between peers at the application layer.
Model	Mathematical or physical representation of a system or a process, based with sufficient precision upon known laws, identification or specified suppositions.
Motion	Any aspect of the dynamics of an axis.
Motion Axis Object	Object that defines the attributes, services, and behaviour of a motion device based axis (or PDS) according to the CIP Motion specification, including Communications, Device Control, and Basic Drive FE elements as defined in IEC 61800-7.
Network	Set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways.
Node	Single DL-entity as it appears on one local link. End-point of a link in a network or a point at which two or more links meet [derived from IEC 61158-2].
Object	Abstract representation of a particular component within a device. An object can be: 1. an abstract representation of the capabilities of a device. Objects can be composed of any or all of the following components: ◦ data (information which changes with time); ◦ configuration (parameters for behaviour); ◦ methods (things that can be done using data and configuration); 2. a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behaviour.
Object dictionary	Data structure addressed by Index and Sub-index that contains description of data type objects, communication objects and application objects. List of objects with unique 16-bit index and 8-bit sub-index as defined in EN 50325-4.
Operating cycle	Period of the control loop within the drive or the control unit.
Operating mode	Characterization of the way and the extent to which the human operator intervenes in the control equipment.

Output data	Data originating in a device, resource or functional element and transferred from them to external systems.
P-Device	Field device and the host for the Drive Objects.
Parameter	Data element that represents device information that can be read from or written to a device, for example through the network or a local HMI.
PDO	Process Data Object.
PDS	Power Drive System.
Process data	Collection of application objects designated to be transferred cyclically or acyclically for the purpose of measurement and control.
Process Data Object (PDO)	Communication object with real-time capability. Structure described by mapping parameters containing one or several process data entities.
Producer	Node or source sending data to one or many consumers.
Profile	Representation of a PDS interface in terms of its parameters, parameter assemblies and behaviour according to a communication profile and a device profile.
Protocol	Convention about the data formats, time sequences, and error correction in the data exchange of communication systems.
Reference variable	Input variable to a comparing element in a controlling system which sets the desired value of the controlled variable and is deducted from the command variable.
Resource	Processing or information capability.
Segment	Collection of one real Master with one or more Slaves.
Server	Object which provides services to another (Client) object.
Service	Operation or function than an object and/or object class performs upon request from another object and/or object class.
Service data	Lower priority real-time data associated with a service message from the controller that is transferred by a CIP Motion connection on a periodic basis.
Set-point	Value or variable used as output data of the application control program to control the PDS.
Slave	DL-entity accessing the medium only after being initiated by the preceding Slave or the Master. Node, which is assigned the right to transmit by the Master.
Standard telegram	Set of input data and output data for an application mode.
Status	Set of information from the PDS to the application control program reflecting the state or mode of the PDS or a functional element of the PDS.
Status word	Two adjacent bytes inside the drive telegram containing status information.

Subindex	Sub-address of an object within the object dictionary.
Supervisor	Engineering device which manages provisions of configuration data (parameter sets) and collections of diagnosis data from P-Devices and/or controllers.
Switch	MAC bridge as defined in IEEE 802.1D.
Sync Manager	Sync Manager has the task of synchronizing data transfer between Master and Slave and prevents the same memory area from being written by different events. Collection of control elements to coordinate access to concurrently used objects.
Sync manager channel	Single control elements to coordinate access to concurrently used objects.
Synchronised	Condition where the local clock value on the drive is locked onto the Master clock of the distributed System Time.
Synchronous with DC SYNC0	In this operating mode data is sampled and then copied into Sync Manager buffer simultaneously at SYNC0 event generated by the ESC capture/compare unit.
Synchronous with SM3	In this mode data is sampled and then copied into Sync Manager buffer as soon as previous data was read from the Master (SM event); in this way new sampled data is synchronous with Master readings.
System Time	Absolute time value as defined in the CIP Sync specification in the context of a distributed time system where all devices have a local clock that is synchronised with a common Master clock.
Telegram	Message.
Time stamp	System Time stamp value associated with the CIP Motion connection data that conveys the absolute time when the associated data was captured, or that can also be used to determine when the associated data shall be applied.
Topology	Physical network architecture with respect to the connection between the stations of the communication system.
Type	Hardware or software element which specifies the common attributes shared by all instances of the type.
Use case	Class specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.
Variable	Software entity that may take different values, one at a time.

List of abbreviations

Table below contains a list of abbreviations (in alphabetical order) which may be used in this guide to describe the EtherCAT interface.

Term	Description
ADS	Automation Device Specification
AL	Application layer
AoE	ADS over EtherCAT
AP (-task)	Application (-task) on top of the stack
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CoE	CANopen over EtherCAT
COS	Change of State
DC	Distributed Clocks
DL	Data Link Layer
DPM	Dual port memory
E2PROM (EEPROM)	Electrically erasable Programmable Read-only Memory
EoE	Ethernet over EtherCAT
ESC	EtherCAT Slave Controller
ESM	EtherCAT State Machine
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control and Automation Technology
FMMU	Fieldbus Memory Management Unit
FoE	File Access over EtherCAT
IEEE	Institute of Electrical and Electronics Engineers
LFW	Loadable firmware
LOM	Linkable object modules
LSB	Least significant byte
MSB	Most significant byte
OD	Object dictionary
ODV3	Object dictionary Version 3
PHY	Physical Interface (Ethernet)
PDO	Process Data Object (process data channel)

Term	Description
RTR	Remote Transmission Request
RxPDO	Receive PDO
SDO	Service Data Object (representing an acyclic data channel)
SHM	Shared memory
SM	Sync Manager
SoE	Servo Profile over EtherCAT
SSC	SoE Service Channel
TxPDO	Transmit PDO
VoE	Vendor Profile over EtherCAT
XML	eXtensible Markup Language

References

[1] IEC 61158 Part 2-6 Type 12 documents (also available for members of EtherCAT Technology Group as specification documents ETG-1000)

[2] Proceedings of EtherCAT Technical Committee Meeting from February 9th, 2005.

[3] IEC 61800-7

[4] EtherCAT Specification Part 5 – Application Layer services specification. ETG.1000.5.

[5] EtherCAT Specification Part 6 – Application Layer protocol specification. ETG.1000.6.

[6] EtherCAT Indicator and Labeling Specification. ETG.1300.

[7] EtherCAT Protocol Enhancements. ETG.1020.

[8] EtherCAT Slave Information Annotation ETG 2001.

[9] EtherCAT Slave Information Specification ETG.2000.

1 - Safety summary

1.1 Safety

- Always adhere to the professional safety and accident prevention regulations applicable to your country during device installation and operation;
- installation and maintenance operations have to be carried out by qualified personnel only, with power supply disconnected and stationary mechanical parts;
- device must be used only for the purpose appropriate to its design: use for purposes other than those for which it has been designed could result in serious personal and/or the environment damage;
- high current, voltage and moving mechanical parts can cause serious or fatal injury;
- warning ! Do not use in explosive or flammable areas;
- failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment;
- Lika Electronic assumes no liability for the customer's failure to comply with these requirements.

1.2 Electrical safety

- Turn OFF the power supply before connecting the device;
- connect according to the explanation in the "4 - Electrical connections" section on page 28;
- connect +Vdc and 0Vdc and check the power supply is correct first before connecting the communication ports;
- in compliance with the 2014/30/EU norm on electromagnetic compatibility, the following precautions must be taken:

  - before handling and installing, discharge electrical charge from your body and tools which may come in touch with the device;
  - power supply must be stabilized without noise, install EMC filters on device power supply if needed;
  - always use shielded cables (twisted pair cables whenever possible);
  - avoid cables runs longer than necessary;
  - avoid running the signal cable near high voltage power cables;
  - mount the device as far as possible from any capacitive or inductive noise source, shield the device from noise source if needed;
  - to guarantee a correct working of the device, avoid using strong magnets on or near by the unit;
  - minimize noise by connecting the shield and/or the connector housing and/or the frame to ground. Make sure that ground is not affected by noise. The connection point to ground can be situated both on the device side and on user's side. The best solution to minimize the interference must be carried out by the user. Provide the ground connection as close as possible to the unit. We suggest

using the ground point provided in the connection cap (use one TCEI UNI M3 x 6 cylindrical head screw with two tooth lock washers).

1.3 Mechanical safety

- Install the device following strictly the information in the "3 - Mechanical installation" section on page 25;
- mechanical installation has to be carried out with power supply disconnected and stationary mechanical parts;
- do not disassemble the unit, unless otherwise indicated in the document;
- do not tool the unit, unless otherwise indicated in the document;
- delicate electronic equipment: handle with care;
- do not subject the device to knocks or shocks;
- respect the environmental characteristics declared by manufacturer.

2 - Identification

The device can be identified through the **ordering code** and the **serial number** printed on the label applied to its enclosure. Information is listed in the delivery document too. Please always quote the ordering code and the serial number when reaching Lika Electronic for purchasing spare parts or needing assistance. For any information on the technical characteristics of the product refer to the technical catalogue.

Warning: devices having ordering code ending with "/Sxxx" may have mechanical and electrical characteristics different from standard and be supplied with additional documentation for special connections (Technical info).

3 - Mechanical installation

WARNING
Installation and maintenance operations have to be carried out by qualified personnel only, with power supply disconnected. Shaft and mechanical components must be in stop.

For any information on the mechanical data and the electrical characteristics of the converter please refer to the technical catalogue.

3.1 Overall dimensions

(values are expressed in mm)

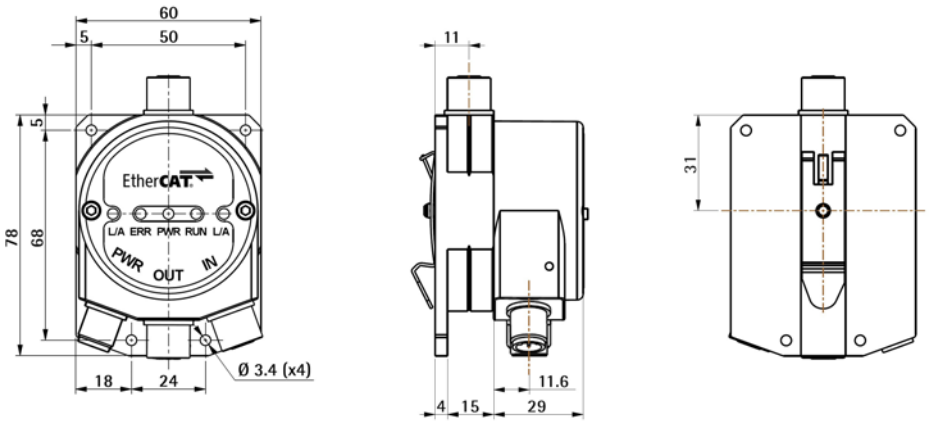


Figure 1 - Overall dimensions

3.2 Installation on panel (Figure 2)

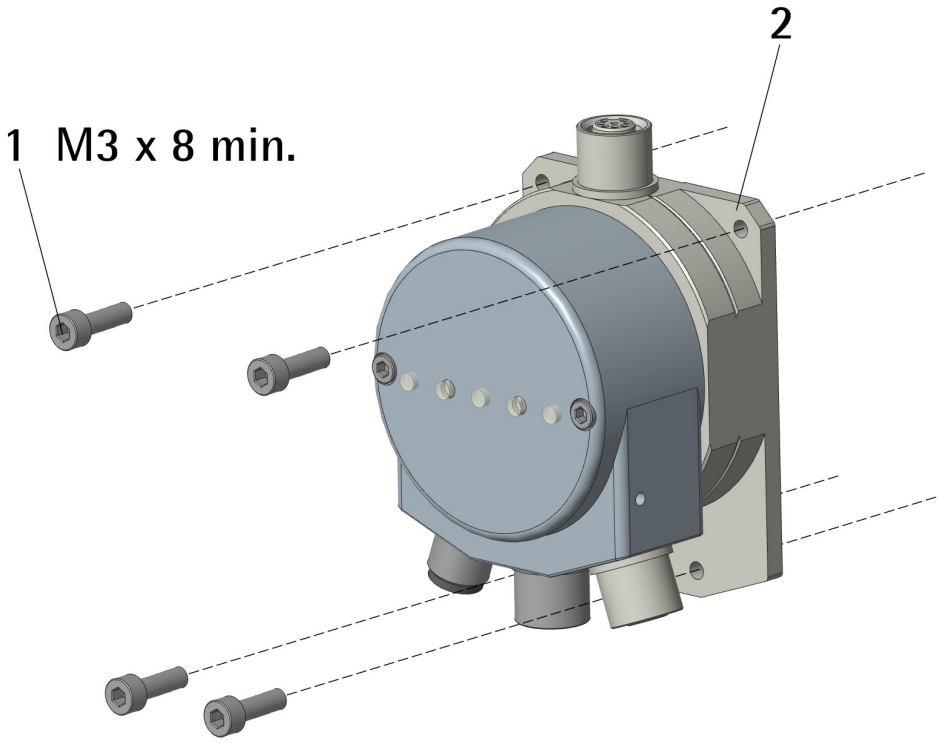


Figure 2 - Installation on panel

The unit is designed for installation on the even surface of a panel.
The back flange **2** is fitted with four holes for inserting the fixing screws **1**. Tighten the four fixing screws **1** until the unit is properly fastened to the support. Use **four M3 8 mm min. long cylinder head screws**. The recommended tightening torque is **1.1 Nm**.

3.3 Installation with DIN TS35 rail clip (Figure 3)

The unit can be installed on DIN profiles inside a rack. A clip **3** for direct fitting on DIN TS35 rails is supplied for free. It has to be fixed on the back of the flange **2** by means of the provided screw **4**.

WARNING

To mount the clip **3** you need to remove the cap **5** and drill a hole **A** in the back flange **2**. Delicate electronic circuits and wirings are located inside both the cap **5** and the back flange **2**. Thus this operation has to be accomplished by skilled personnel only. Please pay careful attention and observe great precaution when carrying out this operation.

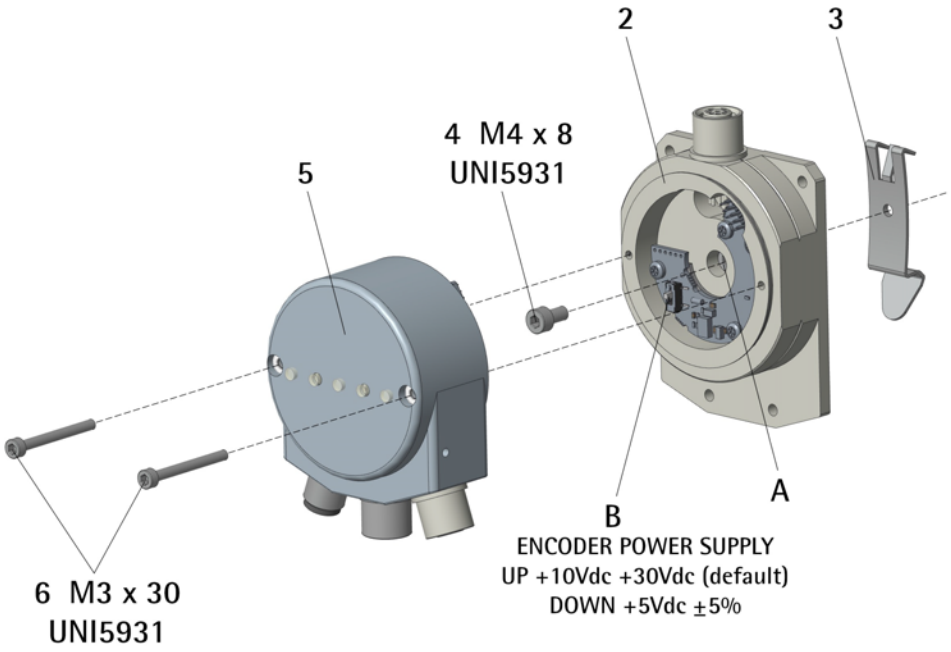


Figure 3 - Installation with DIN TS35 rail clip

- Loosen the two M3 UNI5931 screws **6** that fasten the cap **5** to the back flange **2**;
- open the cap **5** and separate it from the flange **2**; please pay attention to the internal wirings and connectors;
- drill a 4.5 mm diameter hole **A** in the flange **2**; use the notch in the inside of the flange **2** to guide the drill bit;

WARNING

Carefully remove the scrap material after drilling.

- mount the clip **3** on the back of the flange **2** and fix it by means of the provided M4 x 8 UNI5931 screw **4**; it has to be screwed on the inner side of the flange **2**;
- replace the cap **5** and fix it by means of the screws **6**.

4 - Electrical connections

WARNING
Power supply must be turned off before performing any electrical connection! Installation, electrical connection, and maintenance operations must be carried out by qualified personnel only, with power supply disconnected. Mechanical components must be in stop.

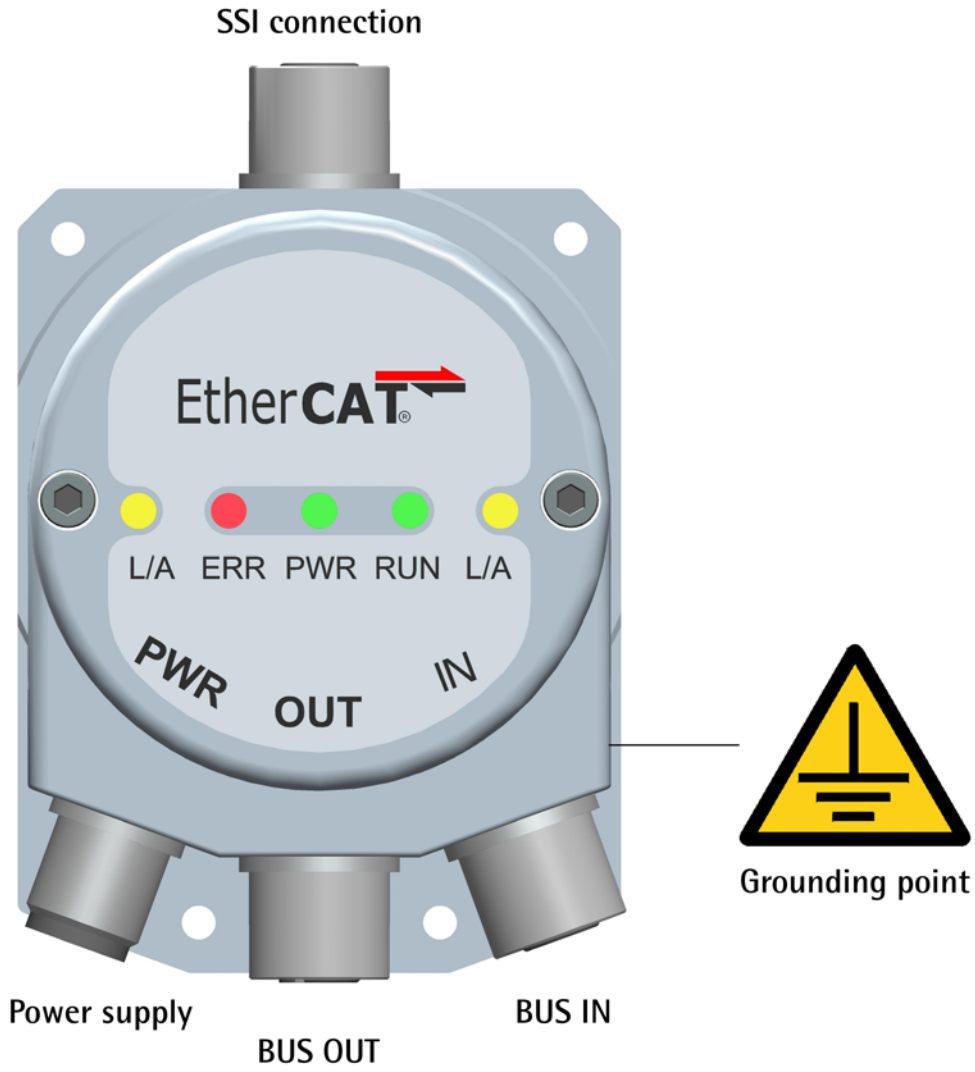
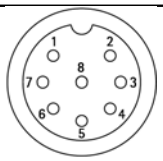


Figure 4 - Connectors and diagnostic LEDs

WARNING
Before switching the power on, please read carefully the "4.1.1 POWER SUPPLY DIP switch (Figure 5)" section on page 29.

4.1 SSI connector (Figure 4)

The connection cap is fitted with one M12 8-pin female connector to network the IF56 gateway and the SSI / BiSS encoder.

M12 8-pin	SSI connection
(frontal side)	 A coding female

Pin	Description
1	0Vdc power supply
2	+Vdc power supply *
3	Clock OUT + / MA +
4	Clock OUT - / MA -
5	Data IN + / SLO +
6	Data IN - / SLO -
7	not connected
8	not connected

* The power supply voltage level must be set through the POWER SUPPLY DIP switch located inside the enclosure of the converter, see the following section.

WARNING
The max. length of the SSI cable must not exceed 30 m / 98.425 ft.
The max. length of the BiSS cable must not exceed 1000 m / 3,281 ft.

4.1.1 POWER SUPPLY DIP switch (Figure 5)

WARNING
Power supply must be turned off before performing this operation!

The power supply voltage level to be provided to the connected encoder must be set through the POWER SUPPLY DIP switch **B** located inside the enclosure of the converter. It must be according to the power supply voltage level required by the connected SSI / BiSS encoder. To access the POWER SUPPLY DIP switch refer to the following section.

Set the POWER SUPPLY DIP switch to the UP position to provide +10Vdc +30Vdc power supply voltage level to the encoder (default setting); set the

POWER SUPPLY DIP switch to the DOWN position to provide +5Vdc ±5% power supply voltage level to the encoder.

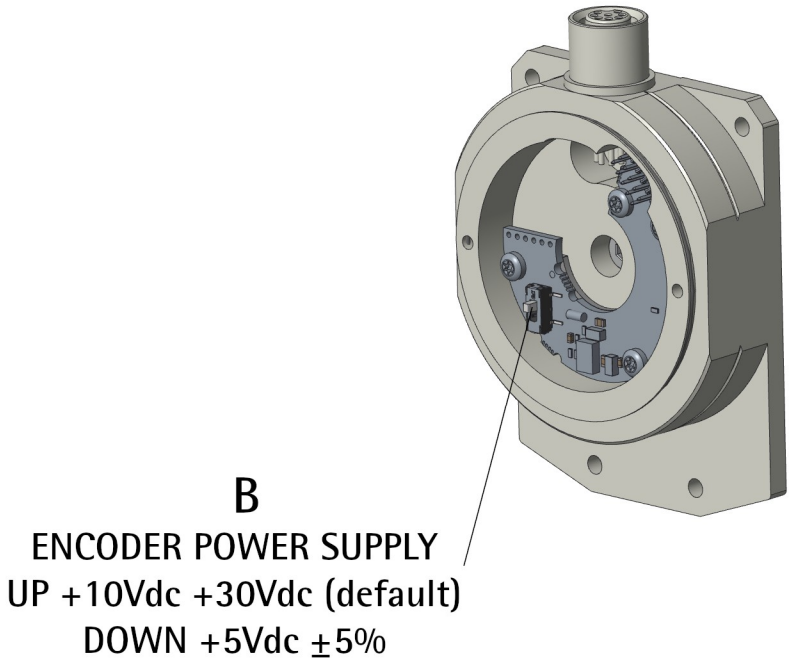


Figure 5 - POWER SUPPLY DIP switch

4.1.2 Connection cap of the converter (Figure 6)

WARNING
Do not remove or mount the connection cap with the power supply switched ON. Damage may be caused to the internal components.

The DIP switch meant to set the power supply of the connected SSI / BiSS encoder is located inside the converter. Thus you must remove the connection cap to access it.

NOTE
Be careful not to damage the internal components when you perform this operation.
To remove the connection cap loosen the two M3 x 30 UNI5931 screws **6** (Figure 6). Please be careful with the internal connector.
Always replace the connection cap at the end of the operation. Take care in re-connecting the internal connector. Tighten the screws **6** using a tightening torque of approx. 2.5 Nm.

WARNING
You are required to check that the back flange of the converter and the connection cap are at the same potential before replacing the connection cap!

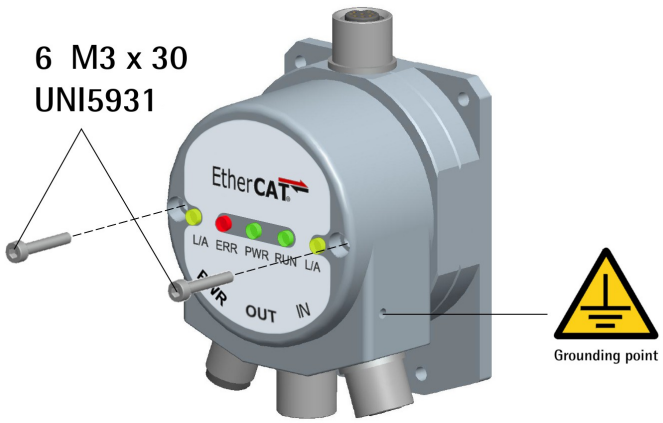
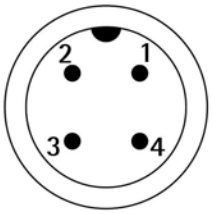


Figure 6 - Removing the connection cap

4.2 PWR Power supply connector (Figure 4)

The M12 4-pin male connector with A coding is used to supply the IF56 converter.



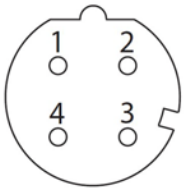
Description	Pin
+10Vdc +30Vdc	1
n.c.	2
0Vdc	3
n.c.	4

n.c. = not connected

WARNING
Connect +Vdc and 0Vdc and check the power supply is correct first before connecting the communication ports.

4.3 P1 Port 1 and P2 Port 2 connectors (Figure 4)

Two M12 4-pin female connectors with D coding are used for Ethernet connection through port 1 and port 2.



Description	Pin
Tx Data +	1
Rx Data +	2
Tx Data -	3
Rx Data -	4

WARNING
Connect +Vdc and 0Vdc and check the power supply is correct first before connecting the communication ports.

The M12 connectors have pin-out in compliance with the EtherCAT® standard. Therefore you can use standard EtherCAT cables commercially available.
The Ethernet interface supports 100 Mbit/s, fast Ethernet, full duplex operation.
Please note that input ECATIN (BUS IN) and output ECATOUT (BUS OUT) connectors are not interchangeable! BUS IN connector must be networked towards the EtherCAT Master.

4.4 Network configuration: topologies, cables, hubs, switches - Recommendations

Cables and connectors comply with the EtherCAT specifications. Cables are CAT-5 shielded cables.
Line, tree or star: EtherCAT supports almost any topology. The bus or line structure known from the fieldbuses thus also becomes available for Ethernet, without the quantity limitations implied by cascaded switches or hubs.
The Fast Ethernet physics (100BASE-TX) enables a cable length of 100 m (328 ft) between two devices. Since up to 65,535 devices can be connected, the size of the network is almost unlimited.
The Ethernet protocol according to IEEE 802.3 remains intact right up to the individual device; no sub-bus is required. In order to meet the requirements of a modular device like an electronic terminal block, the physical layer in the coupling device can be converted from twisted pair or optical fiber to LVDS (alternative Ethernet physical layer, standardized in [4.5]). A modular device can thus be extended very cost-efficiently. Subsequent conversion from the backplane physical layer LVDS to the 100BASE-TX physical layer is possible at any time – as usual with Ethernet.

For a complete list of the available cordsets and connection kits please refer to the product datasheet ("Accessories" list).

4.5 Addressing

It is not necessary to assign a physical address to the device because the addressing of the Slave is automatic at power-on during the initial scanning of the hardware configuration.
The field for addressing is 32-bit long, there are three kinds of addressing:

- Auto Increment Addressing = Position Addressing: 16 bits indicate the physical position of the Slave inside the network while 16 bits are scheduled for local memory addressing; when the Slave receives the frame then it increments the position address and the Slave receiving address 0 is the addressed device;

- Fixed Addressing = 16 bits indicate the physical address of the Slave inside the network while 16 bits are scheduled for addressing the local memory;
- Logical Address = the Slave is not provided with its own individual address, but it can read and write data in a section of the total memory space available (4 Gigabytes).

For complete information refer to the "7.1.5 Addressing" section on page 82.

4.6 Line Termination

EtherCAT network needs no line termination because the line is terminated automatically; in fact every Slave is able to detect the presence of the downstream Slaves. For complete information refer to the "7.1.4 Line Termination" section on page 81.

4.7 Ground connection

To minimize noise connect properly the shield and/or the connector housing and/or the frame to ground. Connect properly the cable shield to ground on user's side. Lika's EC- pre-assembled cables are fitted with shield connection to the connector ring nut in order to allow grounding through the body of the device. Lika's E- connectors have a plastic gland, thus grounding is not possible. If metal connectors are used, connect the cable shield properly as recommended by the manufacturer. Anyway make sure that ground is not affected by noise. It is recommended to provide the ground connection as close as possible to the device. We suggest using the ground point provided in the housing (see Figure 4, use one TCEI UNI M3 x 6 cylindrical head screw with two tooth lock washers).

4.8 Diagnostic LEDs (Figure 4)

Five LEDs located in the rear side of the converter (see Figure 4) are designed to show the operating or fault status of the EtherCAT® interface.
The LEDs operation is according to the EtherCAT specifications, see ETG1300_S_R_V1i1i0_IndicatorLabelingSpecification.pdf.

LED states	Definition
ON	The indicator shall be constantly ON.
OFF	The indicator shall be constantly OFF.
Flickering	The indicator shall turn ON and OFF iso-phase with a frequency of approximately 10 Hz to indicate high Ethernet activity: ON for approximately 50 ms and OFF for approximately 50 ms. The LED turns ON and OFF in irregular intervals to indicate low Ethernet activity.
Blinking	The indicator shall turn ON and OFF iso-phase with a frequency

	of 2.5 Hz: ON for 200 ms followed by OFF for 200 ms.
Single flash	The indicator shall show one short flash (200 ms) followed by a long OFF phase (1,000 ms).
Double flash	The indicator shall show a sequence of two short flashes (each 200 ms), separated by a short OFF phase (200 ms), and followed by a long OFF phase (1,000 ms).

L/A OUT Link/Activity LED for port BUS OUT (green / yellow)

It shows the state and the activity of the physical link (port BUS OUT).

L/A LED	Description	Meaning
OFF	No link No activity	The device has no link to the Ethernet, the link through port BUS OUT is not active. There is no activity on port BUS OUT, the device does not send/receive Ethernet frames through port BUS OUT.
ON green	Link active No activity	Port BUS OUT link active, the device is linked to the Ethernet, but there is no activity on port BUS OUT, the converter does not send/receive Ethernet frames through port BUS OUT.
FLICKERING yellow	Activity	Port BUS OUT link is active, there is activity on port BUS OUT and the device sends/receives Ethernet frames through port BUS OUT.

ERR LED (red)

It shows the current error state of the EtherCAT converter.

ERR LED	Description
OFF	No error. The EtherCAT communication of the converter is in working condition.
BLINKING (2.5 Hz) red	Invalid configuration, general configuration error. Possible reason: state change commanded by the Master is not possible due to register or object settings.
SINGLE FLASH red	Local error (see ETG1000.6, "EtherCAT Specification – Part 6"). Slave device application has changed the EtherCAT state autonomously. Possible reason 1: a host watchdog timeout has occurred. Possible reason 2: synchronization error, the converter enters the SAFE-OPERATIONAL state automatically.
DOUBLE FLASH	Application watchdog timeout. An application watchdog

red	timeout has occurred. Possible reason: Sync Manager Watchdog timeout.

PWR Power LED (green / red)

It shows the power supply and system state.

PWR LED	Description	Meaning
OFF	Power OFF	The converter power supply is switched OFF. No supply voltage for the device or hardware fault.
ON green	Power ON	The converter power supply is switched ON. The firmware is running.
BLINKING red	No firmware program installed, firmware update mode	At power ON the LED blinks red at 1 Hz. The firmware program is not installed, the converter enters the firmware update mode and waits for the firmware file to be installed. For complete information refer to the "6.9 Firmware update" section on page 73; or to the "8.6 Firmware update" section on page 156.

RUN LED (green)

It shows the current communication state of the EtherCAT converter.

ERR LED	Description
OFF	The converter is in INIT state.
BLINKING (2.5 Hz) green	The converter is in PRE-OPERATIONAL state.
SINGLE FLASH green	The converter is in SAFE-OPERATIONAL state.
ON green	The converter is in OPERATIONAL state.

L/A IN Link/Activity LED for port BUS IN (green / yellow)

It shows the state and the activity of the physical link (port BUS IN).

L/A LED	Description	Meaning
OFF	No link No activity	The device has no link to the Ethernet, the link through port BUS IN is not active.

		There is no activity on port BUS IN, the device does not send/receive Ethernet frames through port BUS IN.
ON green	Link active No activity	Port BUS IN link active, the device is linked to the Ethernet, but there is no activity on port BUS IN, the converter does not send/receive Ethernet frames through port BUS IN.
FLICKERING yellow	Activity	Port BUS IN link is active, there is activity on port BUS IN and the device sends/receives Ethernet frames through port BUS IN.

4.9 EtherCAT states

EtherCAT Slave is a state machine; the communication and the operating characteristics depend on the current state of the device:

- INIT: it is the default state after power-on; in this state there is not direct communication between the Master and the Slave on the Application Layer; some configuration registers are initialized and the Sync Managers are configured.
- PRE-OPERATIONAL (PREOP): in this state the mailbox is active; both the Master and the Slave can use the mailbox and its protocols for exchanging specific initialization parameters of the application. Exchange of Process Data (PDO) is forbidden. In this state the FoE protocol is used for firmware download.
- SAFE-OPERATIONAL (SAFEOP): in this state the Master and the Slave can issue only input process data, while the output process data is still in the SAFE-OPERATIONAL state;
- OPERATIONAL (OP): in this state the Master and the Slave are enabled to send both input process data and output process data.
- BOOTSTRAP (BOOT): no process data communication. Communication only via mailbox on Application Layer available. Special mailbox configuration is possible, e.g. larger mailbox size.

The current state of the Slave is signalled through the green **RUN** LED, see on page 33.

4.10 Tactile switch (Figure 7)

WARNING
Be careful not to press the tactile switch **C** unless specifically requested.

A tactile switch **C** is located inside the connection cap of the converter. Thus you must remove it to access the switch. It has no useful function to the operator under normal usage conditions, so never press it unless specifically requested by Lika Electronic's technicians. Electrical power must be provided for operation.
For complete information on accessing the inside of the converter please refer to the "4.1.2 Connection cap of the converter (Figure 6)" section on page 30.

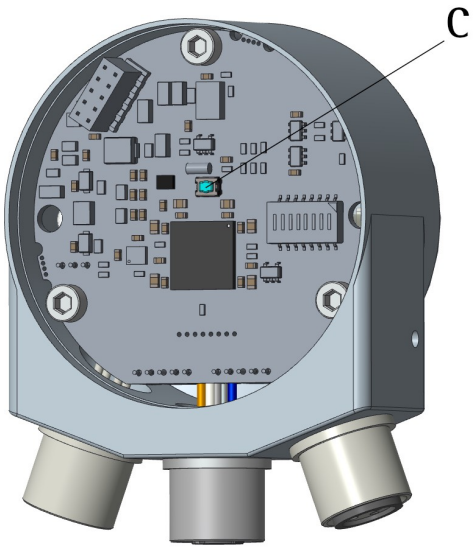


Figure 7 - Tactile switch

5 - Getting started

The following instructions are provided to allow the operator to set up the device for standard operation in a quick and safe mode.

5.1 Quick setting and main functions

The following instructions allow the operator to quickly and safely set up the device in a standard operational mode and to execute its main functions.
Sometimes a function or a procedure can be accomplished by using alternative ways:

- by means of a software tool such as TwinCAT from Beckhoff (see the "6 - Quick reference with TwinCAT" section on page 52 ff);
- or by means of the Integrated Web Server (see the "8 – Integrated Web Server" section on page 147).

They are all mentioned whenever available.
For complete and detailed information please read the mentioned pages thoroughly.

- Mechanically install the device (see on page 25);
- execute the power supply connection;
- check the position of the POWER SUPPLY DIP switch designed to set the voltage level of the power supply to be provided to the connected encoder; default setting = UP = +10Vdc +30Vdc; refer to the "4.1.1 POWER SUPPLY DIP switch (Figure 5)" section on page 29;
- switch on the +10Vdc +30Vdc power supply to the converter, see on page 28 ff; check the soundness of the connection;
- switch off the power supply and execute the network connection, then switch on the power supply again, see on page 28 ff; check the soundness of the connection;
- you do not need to set the node address and the transmission rate (see on page 82);
- you do not need to set any line termination (see on page 81);
- in the software tool install the XML file corresponding to the converter to be installed, see on page 60 ff; the XML is the same for all models, it is:
  - **Lika_IF56_EC_vx_x.xml**: it is intended for installation of **all IF56 converters** ("IF56" is the general identification of the converter series; "EC" is the Lika code that identifies the EtherCAT protocol; "vx_x" is the version of the XML file);
  - then you will be requested to install the specific module according to the type of encoder that is connected: **IF56-ROT-EC** for rotary encoders; **IF56-LIN-EC** for linear encoder;
- if you want to use the physical resolution (see the **6501-00 Hardware Singleturn Resolution** and **6502-00 Hardware number of revolutions** objects for rotary encoders; see the **6501-00 Measuring step** object for

linear encoders), please check that the **Scaling function** parameter is disabled (the bit 2 in the **6000-00 Operating parameters** object = 0; see on page 117);

- otherwise, if you need a custom resolution, enable the **Scaling function** parameter (the bit 2 in the **6000-00 Operating parameters** object = 1; see on page 117) and then set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects for rotary encoders (see on page 119); set the resolution you need for your application next to the **6005-01 Position step setting** object for linear encoders (see on page 129)
- if needed, you can enter the Preset value next to the **6003-00 Preset Value 32 bits** object and then set it in the desired position; see on page 126;
- save the new setting values (use the **1010-01 Store parameters** object; see on page 94).

NOTE

Please consider that, if the **Bypass** parameter (see the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object on page 102 -rotary encoder-; see the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** on page 106 -linear encoder-) is set to "0" = disabled, the position value read by the encoder can be processed according to needs, so the user can scale the value, set a preset, and change the counting direction. On the contrary, if the **Bypass** parameter is set to "1" = enabled, the information from the encoder is transmitted "as it is" and not processed in any way. The preset, scaling, and counting direction functions -even if set and enabled- are ignored; also the output code setting is ignored. If, for example, the user sets a preset while the bypass mode is enabled, the value is accepted, but not activated. As soon as the bypass mode is disabled, the preset, scaling, and counting direction functions -if set and enabled- become active and the position value will be accordingly.

5.1.1 Setting the scaling function and custom resolution

- If you want to use the physical resolution of the encoder, please check that the **Scaling function** parameter is disabled (the bit 2 in the **6000-00 Operating parameters** object is ="0", see on page 117); in this case, the device uses the physical resolution (see the **6501-00 Hardware Singleturn Resolution** and **6502-00 Hardware number of revolutions** objects for rotary encoders; see the **6501-00 Measuring step** object for linear encoders) to arrange the absolute position value.
  You can also use the Integrated Web Server, see the "8.4 Setting the objects" section on page 153; or a software tool, see the "6.7 CoE Object Dictionary" section on page 71.
- On the contrary, if you need a custom resolution, you must enable the scaling function by setting the **Scaling function** parameter (the bit 2 in

the **6000-00 Operating parameters** object) to ="1" first and then set the required resolution parameters:

- (rotary encoder) set the singleturn resolution next to the **6001-00 Units per revolution** object, see on page 119;
- (rotary encoder) set the total resolution next to the **6002-00 Total measuring range 32 bits** object, see on page 124;
- (linear encoder) set the linear resolution next to the **6005-01 Position step setting** object, see on page 129.

You can also use the Integrated Web Server, see the "8.4 Setting the objects" section on page 153; or a software tool, see the "6.7 CoE Object Dictionary" section on page 71.

5.1.2 Reading the absolute position

To read the position value you can choose among the following methods.

- To read the absolute position of the encoder see the **6004-00 Position Value 32 bits** object on page 128; the **6004-00 Position Value 32 bits** object is mapped in the **1A00-00 TxPDO mapping parameter** object, sub-index 001 **01 Mapped Object 001**, see on page 96.
- Open the Integrated Web Server, see the "8.3 Converter position and speed" section on page 151; see the "8.5 Converter information (EtherCAT objects)" section on page 155.
- Use the software tool, see the "6.6 Process Data Objects" section on page 69.

5.1.3 Reading the velocity value

To read the velocity value you can choose among the following methods.

- To read the velocity value of the encoder see the **3006-00 Velocity Value** object on page 112; the **3006-00 Velocity Value** object is mapped in the **1A00-00 TxPDO mapping parameter** object, sub-index 002 **02 Mapped Object 002**, see on page 96.
- Open the Integrated Web Server, see the "8.3 Converter position and speed" section on page 151; see the "8.5 Converter information (EtherCAT objects)" section on page 155.
- Use the software tool, see the "6.6 Process Data Objects" section on page 69.

5.1.4 Setting and executing the preset

To set and execute the preset you can choose among the following methods.

- Enter a suitable value next to the **6003-00 Preset Value 32 bits** object, see on page 126; the preset value is activated as soon as the value is confirmed.
- Open the **Set Converter Objects** page in the Integrated Web Server, see the "8.4 Setting the objects" section on page 153.
- Use the software tool, see the "6.7 CoE Object Dictionary" section on page 71.

5.1.5 Saving data

To save the parameters permanently you can choose among the following methods.

- Use the **1010-01 Store parameters** object, see on page 94.
- Use the **Save Param.** function in the **Set Converter Objects** page of the Integrated Web Server, see the "8.4 Setting the objects" section on page 153.

5.1.6 Restoring defaults

To restore the default parameters you can choose among the following methods.

- Use the **1011-01 Restore default parameters** object, see on page 94.
- Use the **Load Default** function in the **Set Converter Objects** page of the Integrated Web Server, see the "8.4 Setting the objects" section on page 153.

5.1.7 Connecting a rotary encoder

EXAMPLE 1
We need to connect the **EM58-10-14-BA2-...** rotary encoder.

The main features of the rotary encoder are:
Singleturn Resolution: **10 bits = 1,024 cpr** ("10", see the order code in the product datasheet).
Multiturn Resolution: **14 bits = 16,384 rev.** ("14", see the order code in the product datasheet).
Output code: **Binary code without error bit** (-BA2-, see the order code in the product datasheet).
SSI protocol: **25-bit "LSB Right Aligned" protocol** (-BA2-, see the order code in the product datasheet).

Protocol R (2200-00 Rotary Encoder Settings) = 00 = 25-bit "LSB Right Aligned" SSI protocol
SSI output code R (2200-00 Rotary Encoder Settings) = 0 = Binary code
SSI error bit R (2200-00 Rotary Encoder Settings) = 0 = No error bit
Number of clocks R (2200-00 Rotary Encoder Settings) = 25
Singleturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 10 (10 bits = 1,024 cpr)
Multiturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 14 (14 bits = 16,384 revolutions)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
Now set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects.

EXAMPLE 2
We need to connect the **ES58-13-00-GA2-...** rotary encoder.

The main features of the rotary encoder are:
Singleturn Resolution: **13 bits = 8,192 cpr** ("13", see the order code in the product datasheet).
Multiturn Resolution: **0 bits = 1 rev.** ("00", see the order code in the product datasheet).
Output code: **Gray code without error bit** (-GA2-, see the order code in the product datasheet).
SSI protocol: **13-bit "LSB Right Aligned" protocol** (-GA2-, see the order code in the product datasheet).

Protocol R (2200-00 Rotary Encoder Settings) = 00 = 13-bit "LSB Right Aligned" SSI protocol
SSI output code R (2200-00 Rotary Encoder Settings) = 1 = Gray code
SSI error bit R (2200-00 Rotary Encoder Settings) = 0 = No error bit
Number of clocks R (2200-00 Rotary Encoder Settings) = 13
Singleturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 13 (13 bits = 8,192 cpr)
Multiturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 0 (2^0 bits = 1 revolution)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
Now set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects.

EXAMPLE 3
We need to connect the **EHM36-12-13-GG4-...** rotary encoder.

The main features of the rotary encoder are:
Singleturn Resolution: **12 bits = 4,096 cpr** ("12", see the order code in the product datasheet).
Multiturn Resolution: **13 bits = 8,192 rev.** ("13", see the order code in the product datasheet).
Output code: **Gray code without error bit** (-GG4-, see the order code in the product datasheet).
SSI protocol: **"MSB Left Aligned" protocol** (-GG4-, see the order code in the product datasheet).

Protocol R (2200-00 Rotary Encoder Settings) = 01 = "MSB Left Aligned" SSI protocol
SSI output code R (2200-00 Rotary Encoder Settings) = 1 = Gray code
SSI error bit R (2200-00 Rotary Encoder Settings) = 0 = No error bit
Number of clocks R (2200-00 Rotary Encoder Settings) = 25
Singleturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 12 (12 bits = 4,096 cpr)
Multiturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 13 (13 bits = 8,192 revolutions)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
Now set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects.

EXAMPLE 4
We need to connect the **HM58-16-14-GA2-...** rotary encoder.

The main features of the rotary encoder are:
Singleturn Resolution: **16 bits = 65,536 cpr** ("16", see the order code in the product datasheet).
Multiturn Resolution: **14 bits = 16,384 rev.** ("14", see the order code in the product datasheet).
Output code: **Gray code without error bit** (-GA2-, see the order code in the product datasheet).
SSI protocol: **32-bit "LSB Right Aligned" protocol** (-GA2-, see the order code in the product datasheet).

Protocol R (2200-00 Rotary Encoder Settings) = 00 = 32-bit "LSB Right Aligned" SSI protocol
SSI output code R (2200-00 Rotary Encoder Settings) = 1 = Gray code
SSI error bit R (2200-00 Rotary Encoder Settings) = 0 = No error bit
Number of clocks R (2200-00 Rotary Encoder Settings) = 32
Singleturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 16 (16 bits = 65,536 cpr)
Multiturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 14 (14 bits = 16,384 revolutions)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
Now set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects.

EXAMPLE 5
We need to connect the **SMAR1-BG1-15M-...** rotary encoder.

The main features of the rotary encoder are:
Singleturn Resolution: **15 bits = 32,768 cpr** ("15", see the order code in the product datasheet).
Multiturn Resolution: **16 bits = 65,536 rev.** ("M", see the order code in the product datasheet).

Output code: **Binary code with error bit** (-BG1-, see the order code in the product datasheet).
SSI protocol: **"MSB Left Aligned" protocol** (-BG1-, see the order code in the product datasheet).

Protocol R (2200-00 Rotary Encoder Settings) = 01 = "MSB Right Aligned" SSI protocol
SSI output code R (2200-00 Rotary Encoder Settings) = 0 = Binary code
SSI error bit R (2200-00 Rotary Encoder Settings) = 1 = With error bit
Number of clocks R (2200-00 Rotary Encoder Settings) = 32
Singleturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 15 (15 bits = 32,768 cpr)
Multiturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 16 (16 bits = 65,536 revolutions)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
Now set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects.

EXAMPLE 6
We need to connect the **ASC85-25-00-SC1-...** rotary encoder.

The main features of the rotary encoder are:
Singleturn Resolution: **25 bits = 33,554,432 cpr** ("25", see the order code in the product datasheet).
Multiturn Resolution: **0 bits = 1 rev.** ("00", see the order code in the product datasheet).
Output protocol: **BiSS C-Mode** (-SC1-, see the order code in the product datasheet).

Protocol R (2200-00 Rotary Encoder Settings) = 10 = BiSS C-Mode protocol
Number of clocks R (2200-00 Rotary Encoder Settings) = 33 (Physical Total Resolution [bit] + 8 bits)
Singleturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 25 (25 bits = 33,554,432 cpr)
Multiturn resolution (bits) (2201-00 Rotary Encoder Resolution) = 0 (0 bit = 1 revolution)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
Now set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects.

5.1.8 Connecting a linear encoder

EXAMPLE 1
We need to connect the **SMA5-GA2-0050-...** linear encoder.

The main features of the linear encoder are:
Resolution: **0.05 mm** (-0050-, see the order code in the product datasheet).
Max. measuring length: **5,050 mm** (see the "Mechanical Specifications" in the product datasheet).
Output code: **Gray code without error bit** (-GA2-, see the order code in the product datasheet).
SSI protocol: **25-bit "LSB Right Aligned" protocol** (see the User's manual).

Protocol L (2200-00 Linear Encoder Settings) = 00 = 25-bit "LSB Right Aligned" SSI protocol
SSI output code L (2200-00 Linear Encoder Settings) = 1 = Gray code
SSI error bit L (2200-00 Linear Encoder Settings) = 0 = Without error bit
Number of clocks L (2200-00 Linear Encoder Settings) = 25
2201-00 Linear Encoder Resolution = 50,000 (0.05 mm resolution)
Max No of Information (bit) L (2200-00 Linear Encoder Settings) = 17 (= Max. measuring length / Resolution = 5,050 / 0.05 = 101,000 ≈ 2^{17} = 17 bits)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
6005-00 Measuring step setting ≤ 6501-00 Measuring step, the user can set a custom measuring step
6001-00 Total measuring range ≤ 131,072 (= 5,050 / 0.05 = 101,000 information; max. value 2^{17} = 131,072 dec); the user can set a custom measuring range

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 2^{17} - 1, i.e. 131,071 (assuming that 6001-00 Total measuring range = 131,072; **Max No of Information (bit) L** -2200-00 Linear Encoder Settings- = 17 = 2^{17} = 131,072).

...	131,069	131,070	131,071	0	1	2	...

EXAMPLE 2
We need to connect the **SMA5-GA2-0100-...** linear encoder.

The main features of the linear encoder are:
Resolution: **0.1 mm** (-0100-, see the order code in the product datasheet).
Max. measuring length: **5,050 mm** (see the "Mechanical Specifications" in the product datasheet).
Output code: **Gray code without error bit** (-GA2-, see the order code in the product datasheet).
SSI protocol: **25-bit "LSB Right Aligned" protocol** (see the User's manual).

Protocol L (2200-00 Linear Encoder Settings) = 00 = 25-bit "LSB Right Aligned" SSI protocol
SSI output code L (2200-00 Linear Encoder Settings) = 1 = Gray code
SSI error bit L (2200-00 Linear Encoder Settings) = 0 = Without error bit
Number of clocks L (2200-00 Linear Encoder Settings) = 25
2201-00 Linear Encoder Resolution = 100,000 (0.1 mm resolution)
Max No of Information (bit) L (2200-00 Linear Encoder Settings) = 16 (= Max. measuring length / Resolution = 5,050 / 0.1 = 50,500 ≈ 2^{16} = 16 bits)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
6005-00 Measuring step setting ≤ 6501-00 Measuring step, the user can set a custom measuring step
6001-00 Total measuring range ≤ 65,536 (= 5,050 / 0.1 = 50,500 information; max. value 2^{16} = 65,536 dec); the user can set a custom measuring range

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 2^{16} - 1, i.e. 65,535 (assuming that 6001-00 Total measuring range = 65,536; **Max No of Information (bit) L** (2200-00 Linear Encoder Settings) = 16 = 2^{16} = 65,536).

...	65,533	65,534	65,535	0	1	2	...

EXAMPLE 3

We need to connect the **SMAX-BG2-0100-...** linear encoder.

The main features of the linear encoder are:
Resolution: **0.1 mm** (-0100-, see the order code in the product datasheet).
Max. measuring length: **600 mm** (see the "Mechanical Specifications" in the product datasheet).
Output code: **Binary code without error bit** (-BG2-, see the order code in the product datasheet).
SSI protocol: **"MSB Left Aligned" protocol** (see the User's manual).

Protocol L (2200-00 Linear Encoder Settings) = 01 = "MSB Left Aligned" SSI protocol
SSI output code L (2200-00 Linear Encoder Settings) = 0 = Binary code
SSI error bit L (2200-00 Linear Encoder Settings) = 0 = Without error bit
Number of clocks L (2200-00 Linear Encoder Settings) = 13
2201-00 Linear Encoder Resolution = 100,000 (0.1 mm resolution)
Max No of Information (bit) L (2200-00 Linear Encoder Settings) = 13 (= Max. measuring length / Resolution = 600 / 0.1 = 6,000 ≈ 2^{13} = 13 bits)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
6005-00 Measuring step setting ≤ 6501-00 Measuring step, the user can set a custom measuring step
6001-00 Total measuring range ≤ 8,192 (= 600 / 0.1 = 6,000 information; max. value 2^{13} = 8,192 dec); the user can set a custom measuring range

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 2^{13} - 1, i.e. 8,191 (assuming that 6001-00 Total measuring range = 8,192; **Max No of Information (bit) L** (2200-00 Linear Encoder Settings) = 13 = 2^{13} = 8,192).

...	8,189	8,190	8,191	0	1	2	...

EXAMPLE 4

We need to connect the **SMA2-BG1-0005-...** linear encoder.

The main features of the linear encoder are:
Resolution: **0.005 mm** (-0005-, see the order code in the product datasheet).
Max. measuring length: **8,165 mm** (see the "Mechanical Specifications" in the product datasheet).

Output code: **Binary code with error bit** (-BG1-, see the order code in the product datasheet).
SSI protocol: **"MSB Left Aligned" protocol** (see the User's manual).

Protocol L (2200-00 Linear Encoder Settings) = 01 = "MSB Left Aligned" SSI protocol
SSI output code L (2200-00 Linear Encoder Settings) = 0 = Binary code
SSI error bit L (2200-00 Linear Encoder Settings) = 1 = With error bit
Number of clocks L (2200-00 Linear Encoder Settings) = 22
2201-00 Linear Encoder Resolution = 5,000 (0.005 mm resolution)
Max No of Information (bit) L (2200-00 Linear Encoder Settings) = 21 (= Max. measuring length / Resolution = 8,165 / 0.005 = 1,633,000 ≈ 2^{21} = 21 bits)

If you want to use the physical resolution:
Scaling function (6000-00 Operating parameters) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
6005-00 Measuring step setting ≤ 6501-00 Measuring step, the user can set a custom measuring step
6001-00 Total measuring range ≤ 2,097,152 (= 8,165 / 0.005 = 1,633,000 information; max. value 2^{21} = 2,097,152 dec); the user can set a custom measuring range

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 2^{21} - 1, i.e. 2,097,151 (assuming that 6001-00 Total measuring range = 2,097,152; **Max No of Information (bit) L** (2200-00 Linear Encoder Settings) = 21 = 2^{21} = 2,097,152).

...	2,097,149	2,097,150	2,097,151	0	1	2	...

EXAMPLE 5
We need to connect the **SMA21-SC1-0001-...** linear encoder.

The main features of the linear encoder are:
Resolution: **0.001 mm** (-0001-, see the order code in the product datasheet).
Max. measuring length: **32,749 mm** (see the "Mechanical Specifications" in the product datasheet).
Output protocol: **BiSS C-Mode** (-SC1-, see the order code in the product datasheet).

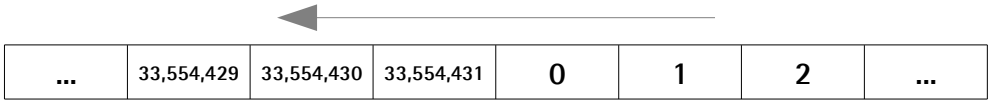
Protocol L (2200-00 Linear Encoder Settings) = 10 = BiSS C-Mode protocol
Number of clocks L (2200-00 Linear Encoder Settings) = 33 (= **Max No of Information (bit) L** + 8 bits)

2201-00 Linear Encoder Resolution = 1,000 (0.001 mm resolution)
Max No of Information (bit) L (**2200-00 Linear Encoder Settings**) = 25 (= Max. measuring length / Resolution = 32,749 / 0.001 = 32,749,000 ≈ 2^{25} = 25 bits)

If you want to use the physical resolution:
Scaling function (**6000-00 Operating parameters**) = 0

If you need a custom resolution:
Scaling function (**6000-00 Operating parameters**) = 1
6005-00 Measuring step setting ≤ **6501-00 Measuring step**, the user can set a custom measuring step
6001-00 Total measuring range ≤ 32,749,000 (= 32,749 / 0.001 = 32,749,000 information; max. value 2^{25} = 33,554,432 dec); the user can set a custom measuring range

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 2^{25} - 1, i.e. 33,554,431 (assuming that **6001-00 Total measuring range** = 33,554,432; **Max No of Information (bit) L** (**2200-00 Linear Encoder Settings**) = 25 = 2^{25} = 33,554,432).



EXAMPLE 6
We need to connect the **SMA1-SB1-0005-...** linear encoder.

The main features of the linear encoder are:
Resolution: **0.005 mm** (-0005-, see the order code in the product datasheet).
Max. measuring length: **5,050 mm** (see the "Mechanical Specifications" in the product datasheet).
Output protocol: **BiSS B-Mode** (-SB1-, see the order code in the product datasheet).

Protocol L (**2200-00 Linear Encoder Settings**) = 10 = BiSS B-Mode protocol
Number of clocks L (**2200-00 Linear Encoder Settings**) = 28 (= **Max No of Information (bit) L** + 8 bits)
2201-00 Linear Encoder Resolution = 5,000 (0.005 mm resolution)
Max No of Information (bit) L (**2200-00 Linear Encoder Settings**) = 20 (= Max. measuring length / Resolution = 5,015 / 0.005 = 1,003,000 ≈ 2^{20} = 20 bits)

If you want to use the physical resolution:
Scaling function (**6000-00 Operating parameters**) = 0

If you need a custom resolution:
Scaling function (6000-00 Operating parameters) = 1
6005-00 Measuring step setting ≤ 6501-00 Measuring step, the user can set a custom measuring step
6001-00 Total measuring range ≤ 1,048,576 (= 5,015 / 0.005 = 1,003,000 information; max. value 2^{20} = 1,048,576 dec); the user can set a custom measuring range

If you set a 0 preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 2^{20} - 1, i.e. 1,048,575 (assuming that 6001-00 Total measuring range = 1,048,576; **Max No of Information (bit) L** (2200-00 Linear Encoder Settings) = 20 = 2^{20} = 1,048,576).

...	1,048,573	1,048,574	1,048,575	0	1	2	...

6 - Quick reference with TwinCAT

Lika converters are Slave devices and support the "CANopen over EtherCAT" (CoE) mode for data transfer. In particular, they support the "CANopen DS 301 Communication profile".
For any omitted specification on the EtherCAT® protocol, please refer to the "ETG.1000 EtherCAT Specification" documents available at the address **www.ethercat.org**.
For any omitted specification on the CANopen® protocol, please refer to the "CiA Draft Standard Proposal 301. Application Layer and Communication Profile" and "CiA Draft Standard 406. Device profile for encoders" documents available at the address **www.can-cia.org**.

6.1 Setting up EtherCAT® communication. Example with TwinCAT3

The following sections show how to set up the EtherCAT® communication using Beckhoff TwinCAT3 software running on a PC. Some screenshots are shown to explain how to install and configure the converter in a supervisor. In the specific example the development environment is TwinCAT version v3.1.4022 from Beckhoff Automation. A 7 day free version of TwinCAT can be downloaded from www.beckhoff.de. For complete information please refer to the I/O TwinCAT 3 manual from Beckhoff.

6.2 About TwinCAT

The TwinCAT software system turns almost any PC-based system into a real-time controller, with a multi-programmable logic controller (PLC) system, numerical control (NC) axis control, programming environment and operating station. TwinCAT brings far greater levels of data visualisation to automation software and fieldbus control, to better handle large data quantities with increased data transparency.
It consists of reliable and powerful software run-time systems that execute control programs in real-time and develop environments for programming, diagnostics, and configuration.
EtherCAT in combination with TwinCAT offers the foundational technologies and tools needed to implement Industry 4.0 concepts and Industrial Internet of Things (IIoT) connectivity – all via PC-based control.

Not only up to 6,000 modules can be controlled from just a single Controller PC, but it can be done so through any Windows programs. TwinCAT brings widespread industry power to your PC through visualisation programs or Office programs, data access through Microsoft interfaces, command executions, an embedded IEC 61131-3 software for PLC, NC and computer numerical control (CNC).

TwinCAT 3 Highlights

- Only one software for programming and configuration

- Visual Studio® integration
- More freedom in selecting programming languages
- Support for the object-oriented extension of IEC 61131-3
- Use of C/C++ as the programming language for real time applications
- Link to MATLAB®/Simulink®
- Integrated motion control solution – from NC PTP, robotics to CNC
- Open interfaces for expandability and adaptation to the tools landscape
- Flexible runtime environment – active support of multi-core CPUs
- Support of 32- and 64-bit operating systems (Windows CE, Windows 7, Windows 10, TwinCAT/BSD)
- Migration of TwinCAT 2 projects
- Automatic code generation and project implementation with the TwinCAT Automation Interface
- Connection to all common fieldbuses
- PC interfaces (network, PCI, USB, hard disk) are supported
- Data connection to user interfaces and other programs using open standards (OPC, ADS etc.)

TwinCAT is the fully-featured and de facto reference EtherCAT Master implementation.

6.2.1 User interface components

TwinCAT 3 consists of various components. The appearance of the user interface is determined by the arrangement and configuration of the individual components. Use the **View** drop-down menu in the menu bar to customize the user interface.

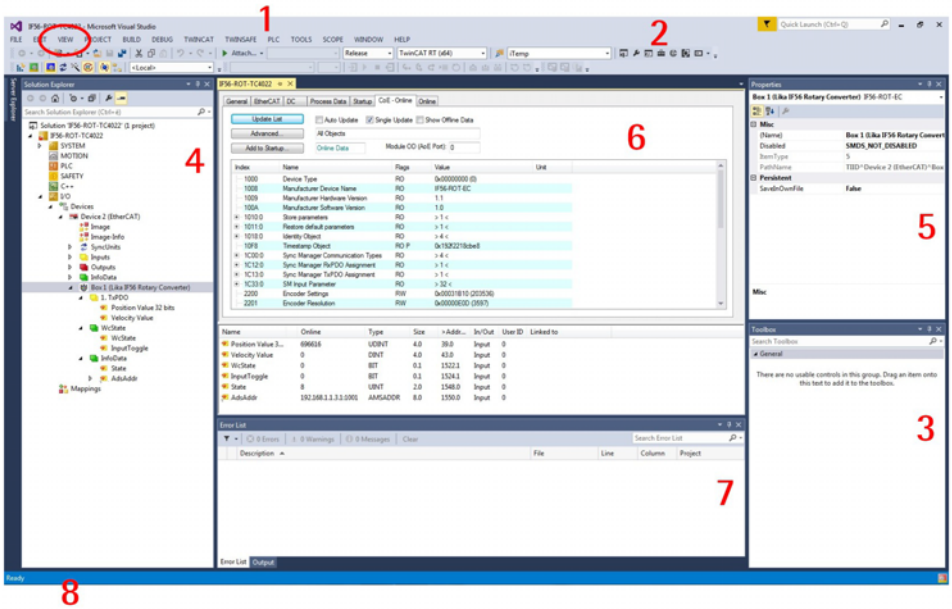


Figure 8 - User interface components

1. **Menu bar**: it shows the menus according to the settings in the Customize dialog.
2. **Toolbar**: it shows the commands as buttons identified with symbols according to the settings in the Customize dialog.
3. **Toolbox**: it shows the "tools" that are available for the currently active editor/project window (e.g., graphical programming elements).
4. **Solution Explorer**: it shows the TwinCAT 3 project with the associated project elements in a structured form. It is also named "tree view".
5. **Properties Window**: it shows the properties of the element that is currently selected in the Solution Explorer.
6. **Editor/Project Window**: it is used for defining and editing the project and the objects.

The following components provide information about the current processes in the project in offline or online mode:

7. **Message Window**: it shows current errors, warnings, and messages relating to syntax check, compile process etc.
8. **Information and status bar**: it shows the status of the TwinCAT 3 runtime. If an editor window is currently active, the current cursor position and the set editing mode are displayed. In online mode you see the current program status.

6.3 Adding the XML file

The .xml file can be downloaded from Lika Electronic web page. After TwinCAT installation is complete, copy the .xml file in the default location: C: \ TwinCAT \ 3.1 \Config \ Io \ EtherCAT. TwinCAT will need a restart in order to load the new file.

6.4 System configuration using TwinCAT software system from Beckhoff

6.4.1 Creating a project

Launch **TwinCAT** program.
You must create your own project first.
To do this, on the **File** menu, point to **New**, and then click **Project...** .

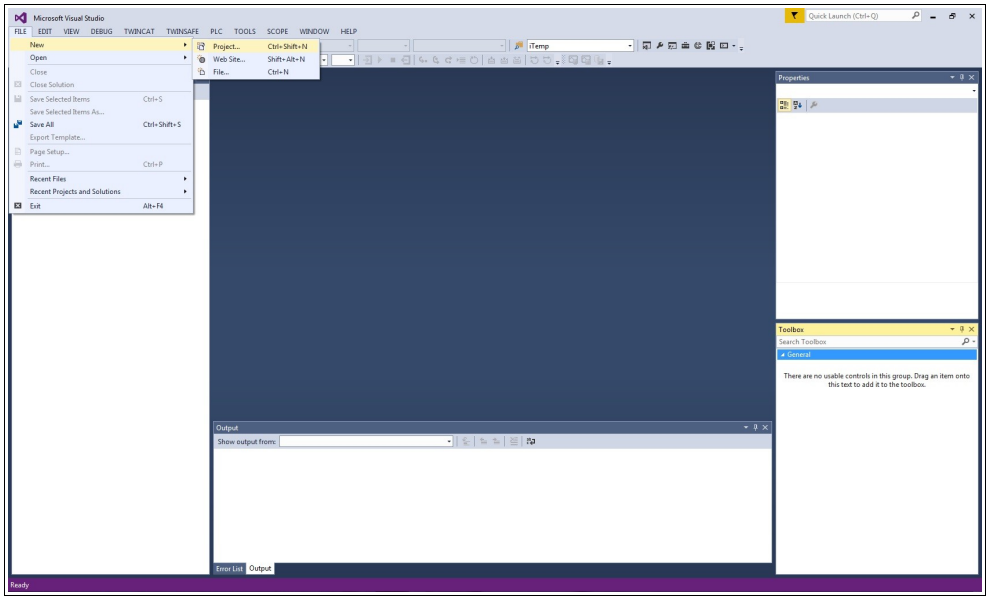


Figure 9 - Creating a new project

The **New project** page will be displayed.
Select the **TwinCAT Projects** under the **Templates** list, type the name of the project, and browse to the location where the project file has to be saved by means of the **BROWSE...** button. Finally press the **OK** button to confirm.

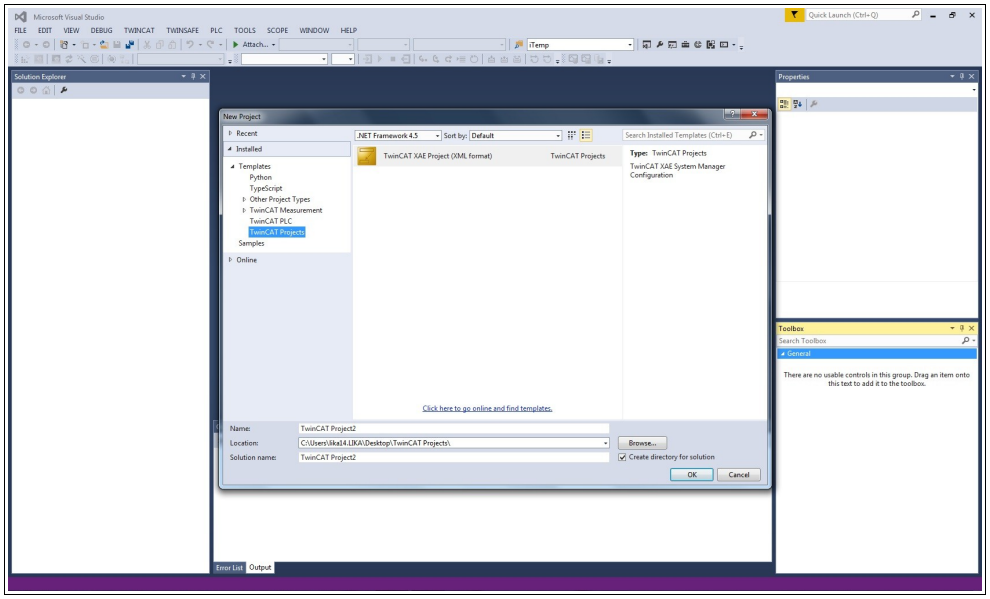


Figure 10 - Name and location of the project

The new TwinCAT project will be listed in the tree view.

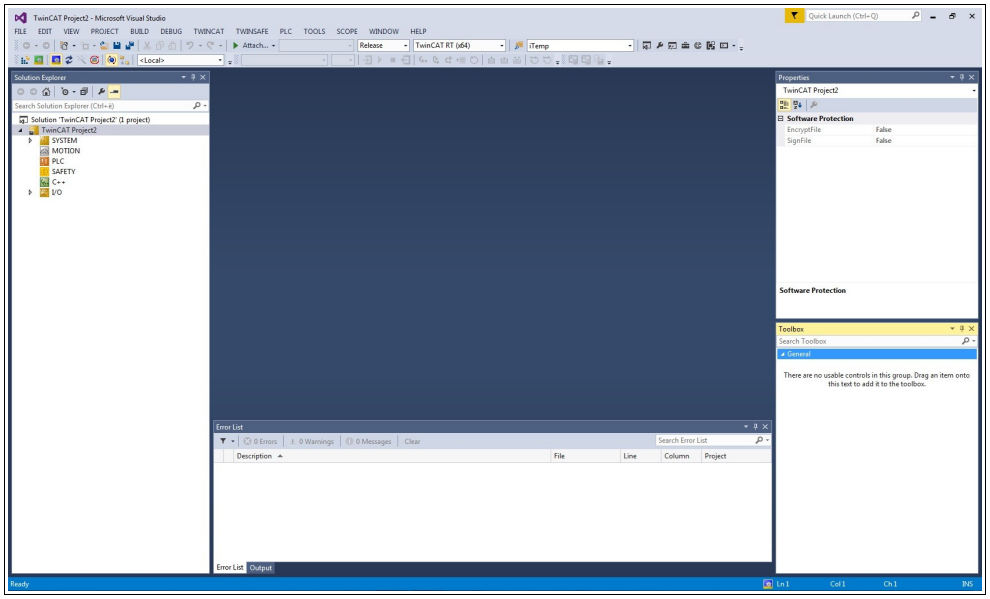


Figure 11 - New TwinCAT project created

6.4.2 Adding an I/O device (network card)

Now you must add an I/O device and set the network card (i.e. your Master).
In the left navigation bar extend the devices tree and select the **Devices** item under the **I/O** directory; right-click the **Devices** item and then press the **Add New Item...** command.

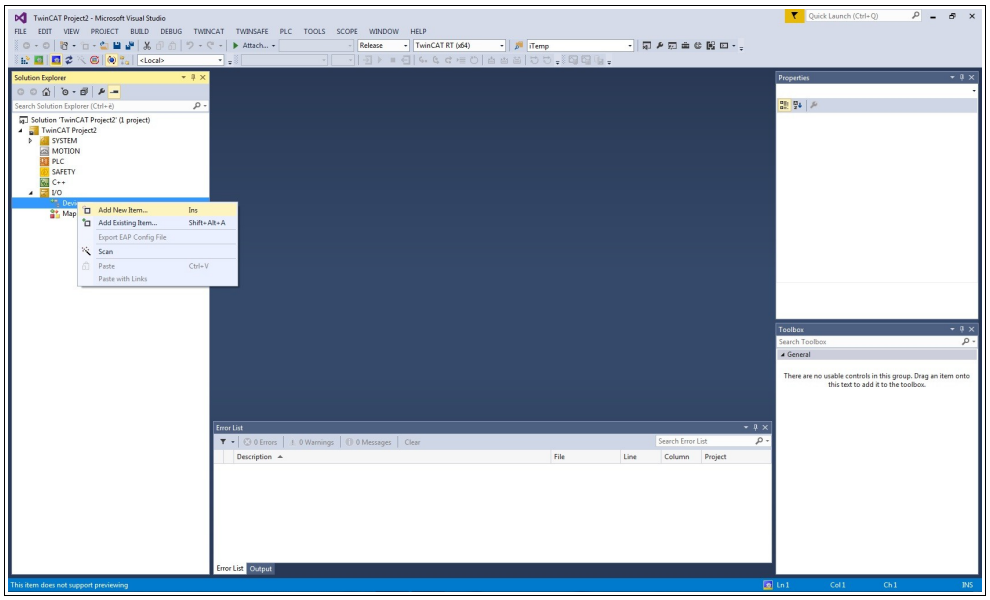


Figure 12 - Adding a new I/O device

The **Insert Device** window will appear.
In the **Insert Device** dialog box expand the **EtherCAT** list and then select the **EtherCAT Master** item and confirm pressing the **OK** button.

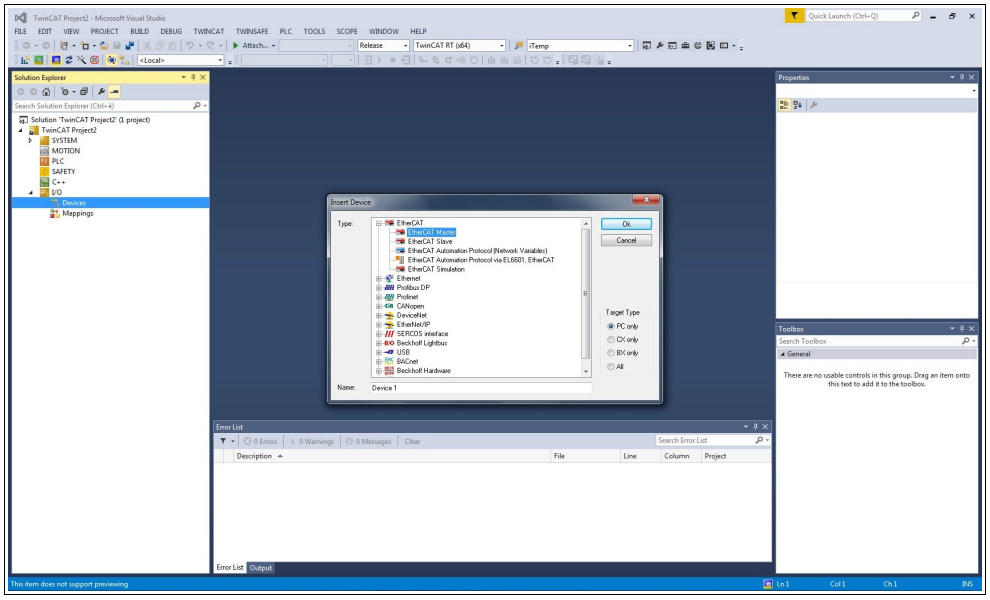


Figure 13 - Adding the EtherCAT Master

If a network card has been already installed properly, double click the installed **Device 1 (EtherCAT)** item under **Devices**, select the **Adapter** tabbed page, and press the **SEARCH...** button. The **Device Found At** dialog box will appear and show the list of the installed devices.

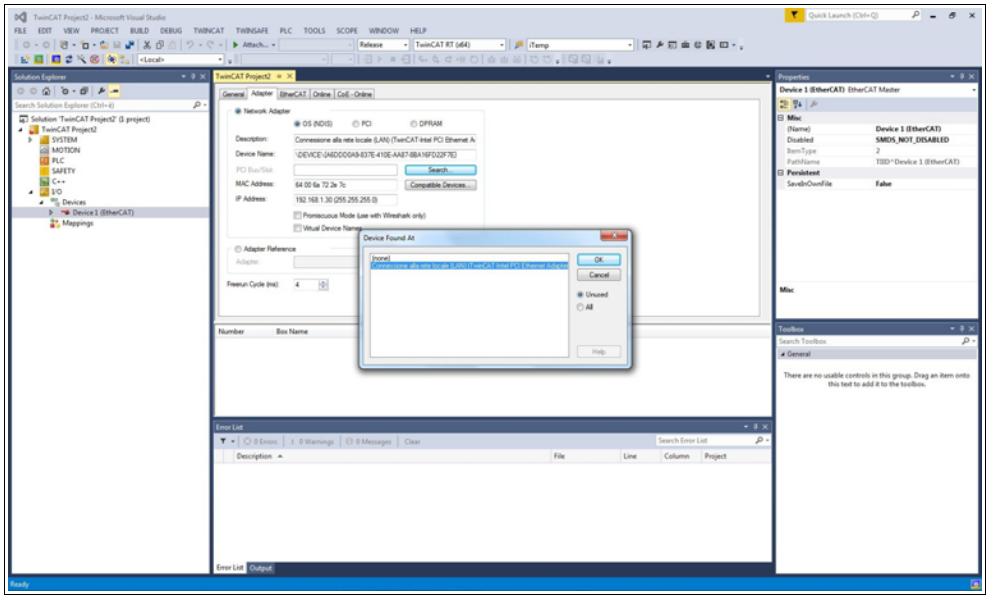


Figure 14 - Selecting the network card

Select the network card you want to use and then confirm the choice by pressing the **OK** button.

If there are no network cards installed, you must install one before proceeding. To do this, select the **TWINCAT** menu on the menu bar and then click **Show Real Time Ethernet Compatible Devices...** .

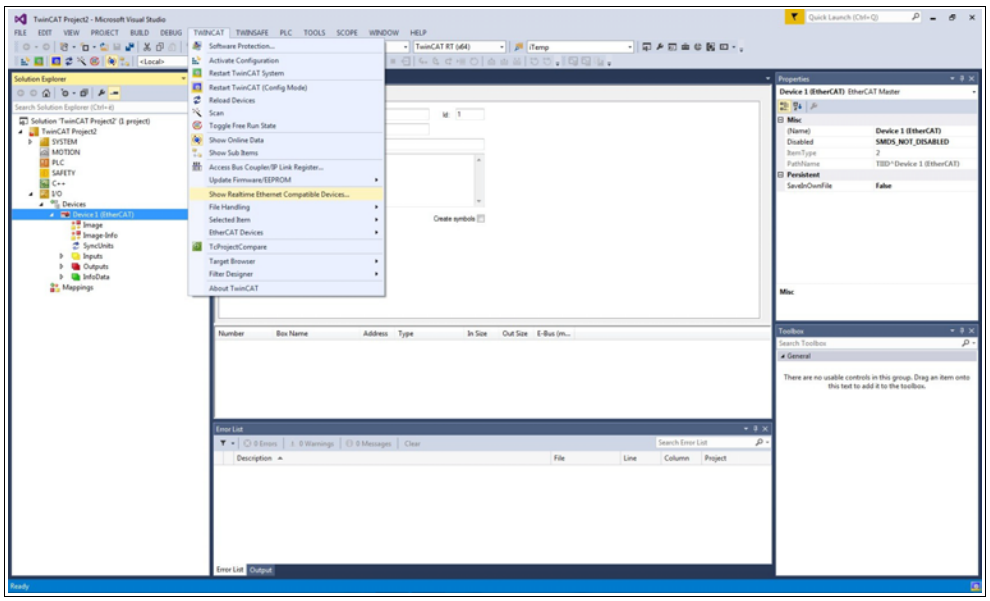


Figure 15 - Real Time Ethernet Compatible Devices

The **Installation of TwinCAT RT – Ethernet Adapters** dialog box will appear.

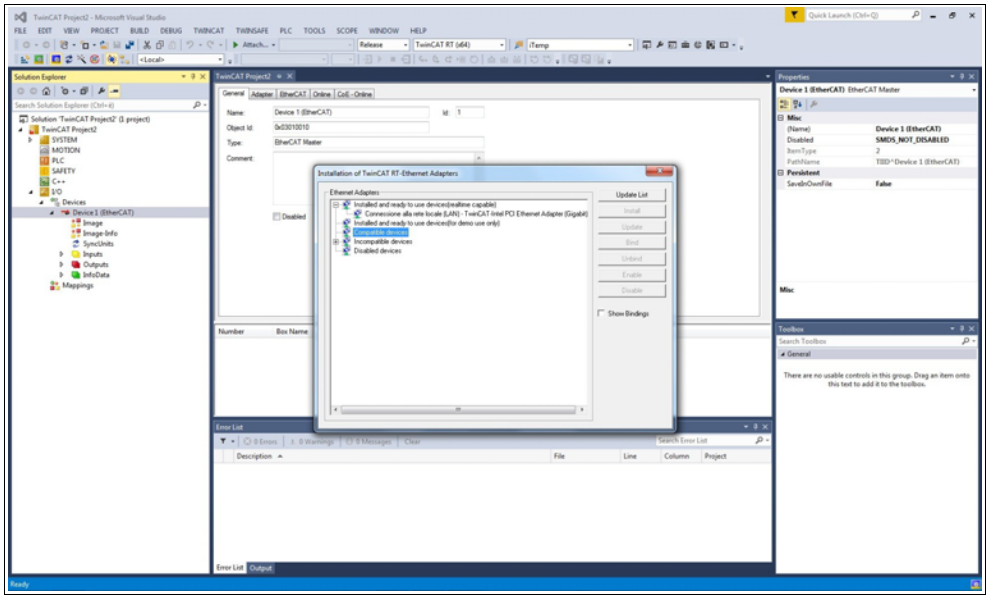


Figure 16 - Selecting the network card

Now expand the **Compatible Devices** list and choose the network card you want to install; finally press the **Install** button to confirm your choice.

As an alternative, you can right-click the **Devices** item in the tree view and press the **SCAN** command. It scans the PC for lower-level devices.

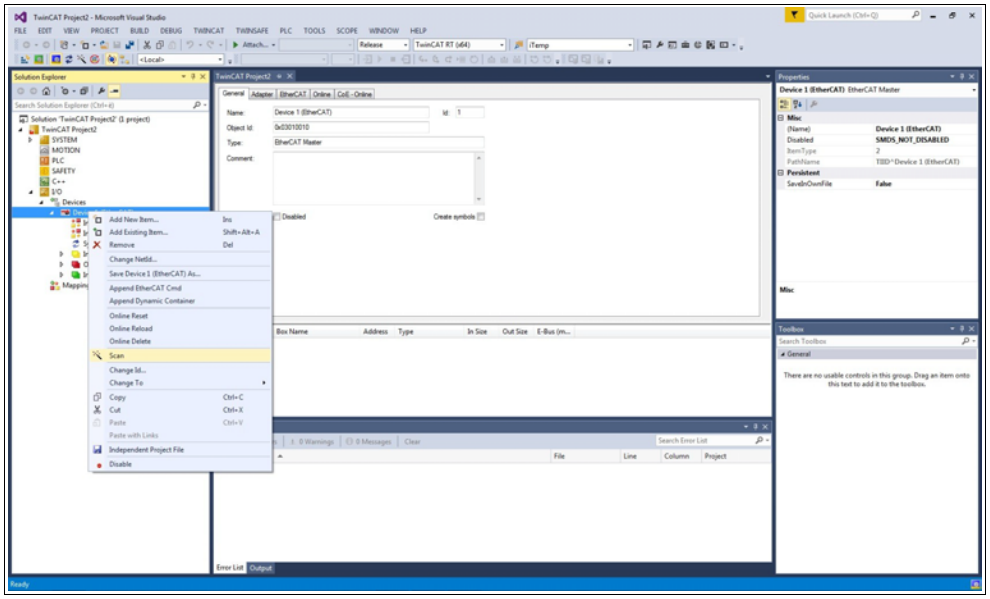


Figure 17 - Scanning the network for connected devices

Found devices are listed in the tree view below **I/O**.

6.4.3 Adding an I/O module (Box)

If one or more devices (i.e. Slaves) are connected to the network and switched ON, right-click the **Device 1 (EtherCAT)** item in the left navigation bar and press the **Add New Item...** command.

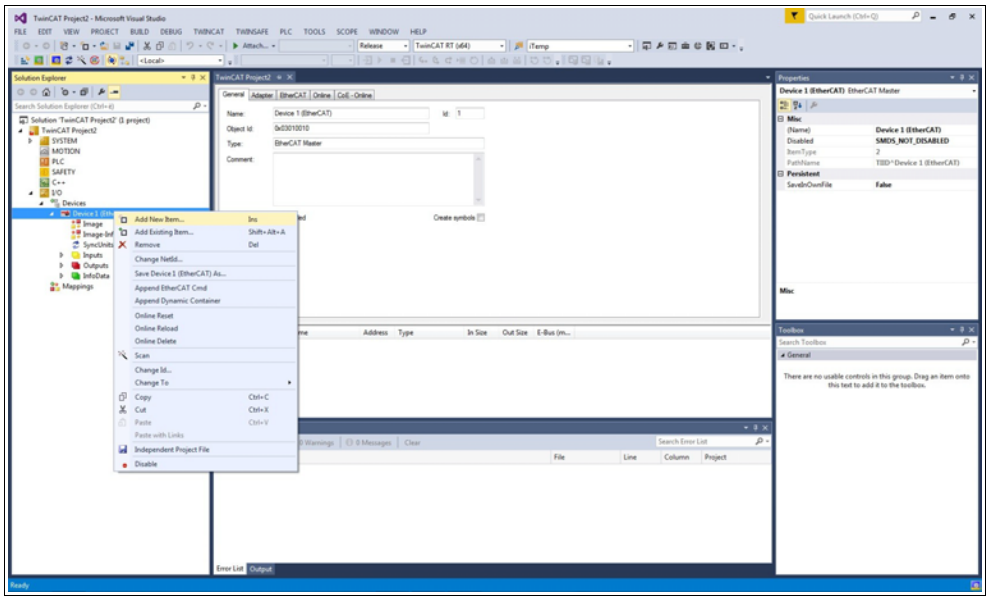


Figure 18 - Adding a new I/O module

The **Insert EtherCAT device** dialog box will be displayed.
Scroll to the **Lika Electronic srl** directory, expand the list and select the converter you want to install (IF56-ROT-EC, i.e. IF56 converter for rotary encoders in the example). Click **OK** to confirm the choice.

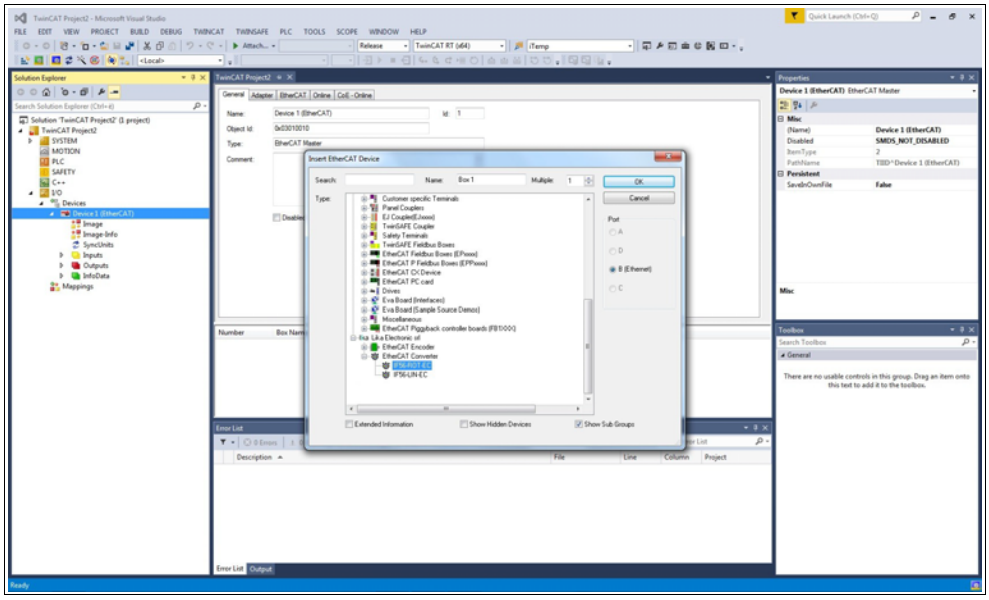


Figure 19 - Selecting the converter

The **Box 1** device will be listed under **Device 1 (EtherCAT)**, Lika IF56 Rotary Converter in the example.

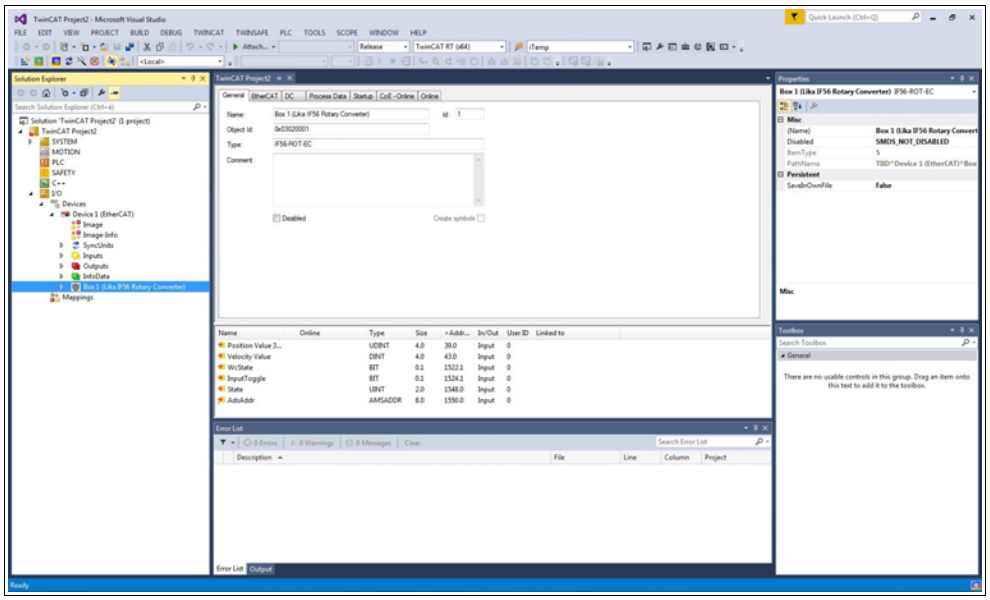


Figure 20 - Converter installed

As an alternative, you can right-click the **Device** item in the tree view and press the **SCAN** command. It scans the network for lower-level boxes.

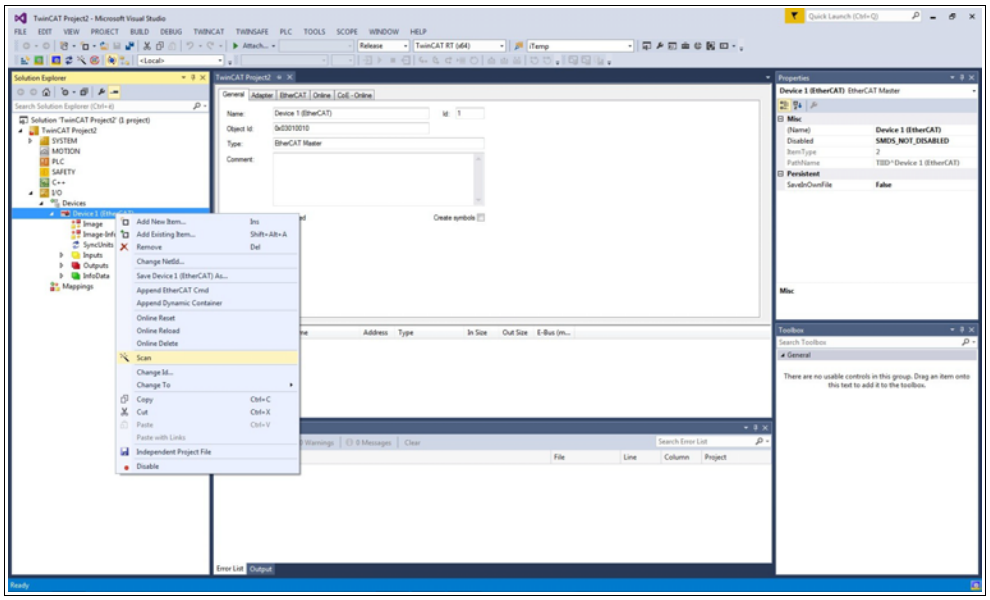


Figure 21 - Scanning the network for connected devices

Found boxes are listed in the tree view below the **Device**.

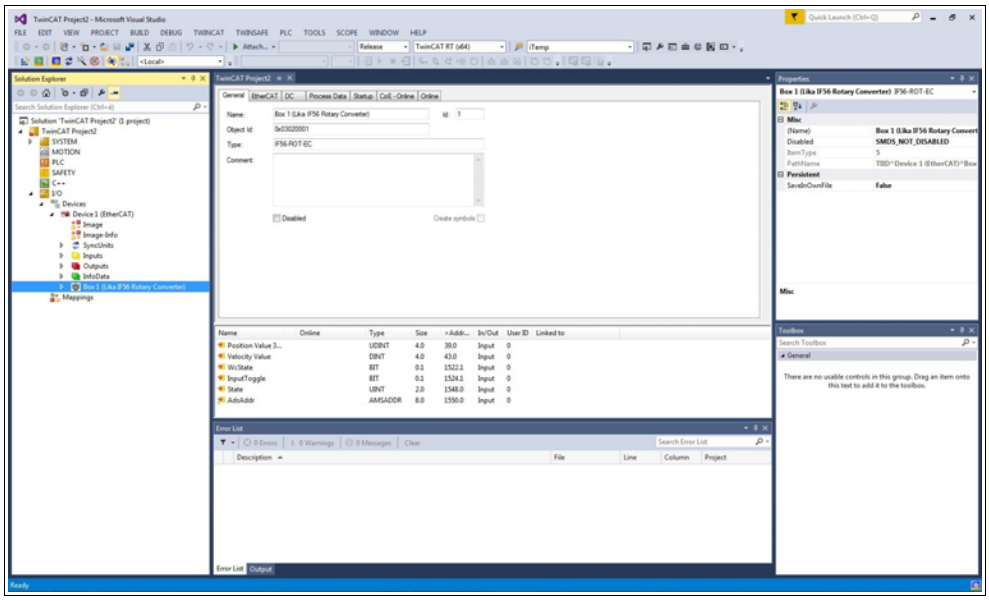


Figure 22 - Found boxes

WARNING

The correct XML file supplied with the converter must be available inside the TwinCAT \ 3.1 \ Config \ Io \ EtherCAT directory: **Lika_IF56_EC_vx_x.xml** (see at **www.lika.biz**).

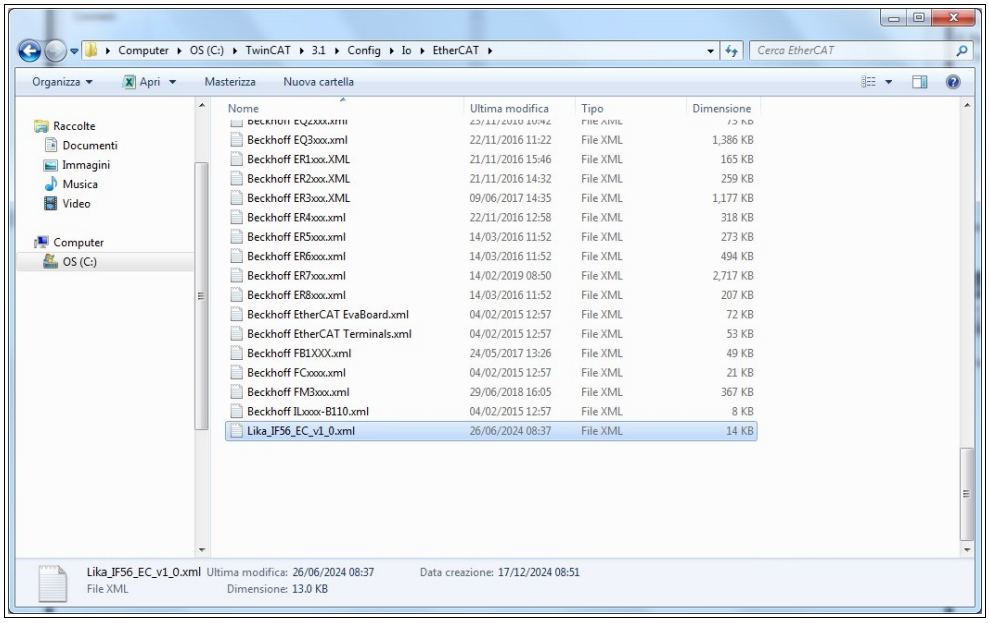


Figure 23 - XML files folder

6.4.4 Tabbed pages

Double click the **Box 1 (Lika IF56 Rotary Converter)** item, the tabbed pages that describe the converter and its operation will be displayed. For complete information please refer to the I/O TwinCAT 3 manual from Beckhoff.

The **General** tabbed page contains some general information on the converter such as the name of the EtherCAT Slave device, the identification number, the TwinCAT object identification number, the EtherCAT device type, etc.

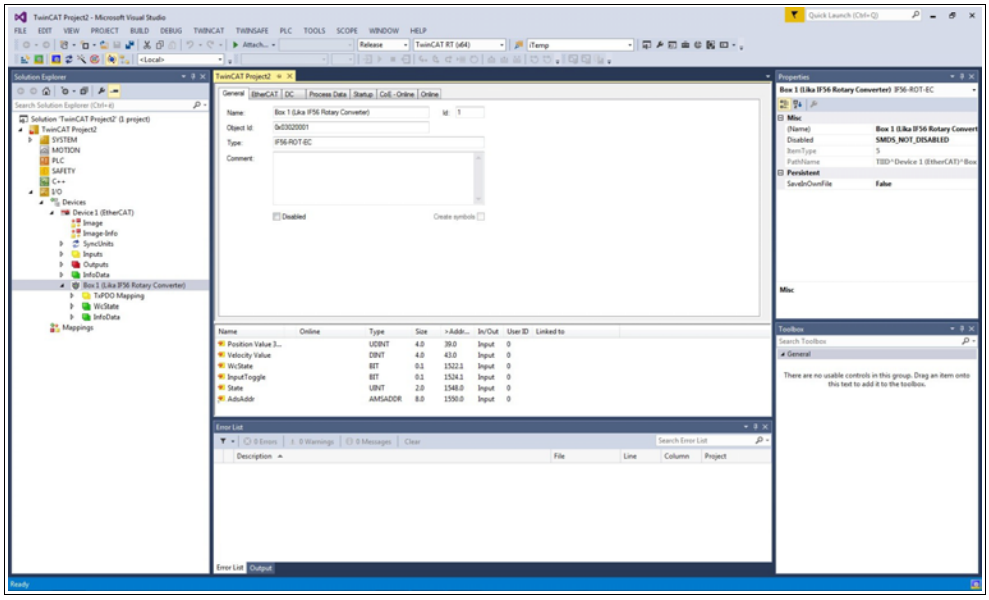


Figure 24 - General tabbed page

The **EtherCAT** tabbed page contains specific information on the EtherCAT configuration. It allows, for instance, to set the CoE, FoE, and EoE protocols parameters and to check the assigned IP address.

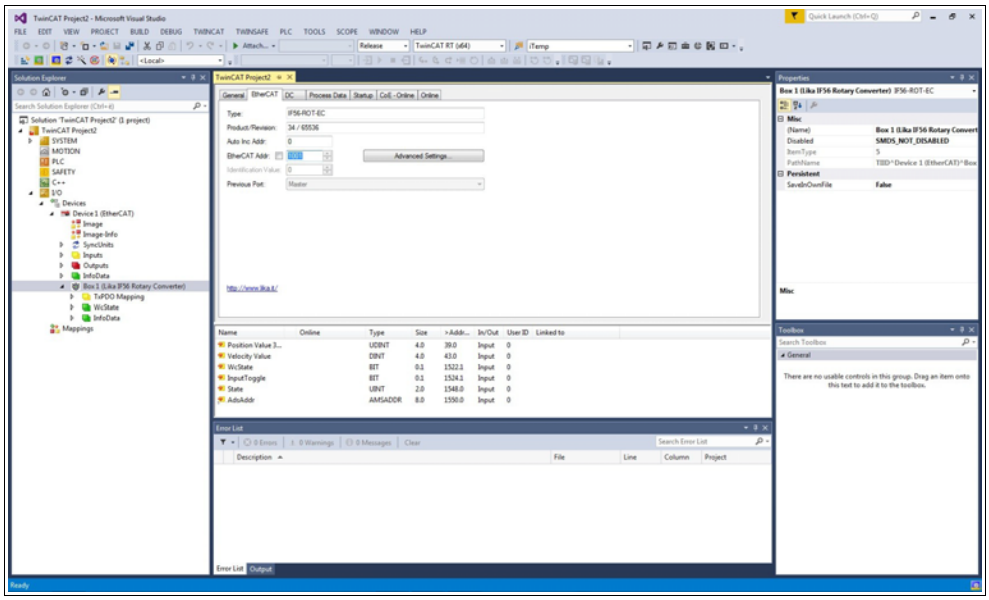


Figure 25 - EtherCAT tabbed page

The **DC** tabbed page allows the operator to select the operation mode of the Slave device. For complete information please refer to the "6.5 Setting the communication mode" section on page 67.

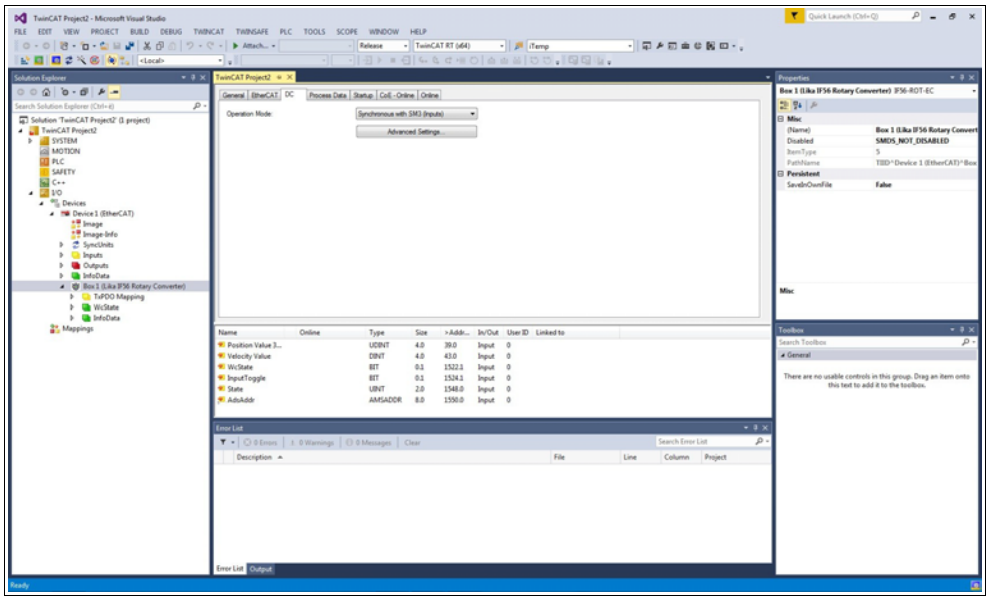


Figure 26 - DC tabbed page

The **Process Data** tabbed page shows the configuration of the process data. The input and output data of the EtherCAT Slave are represented as CANopen process data objects (PDO). Refer also to the "6.6 Process Data Objects" section on page 69.

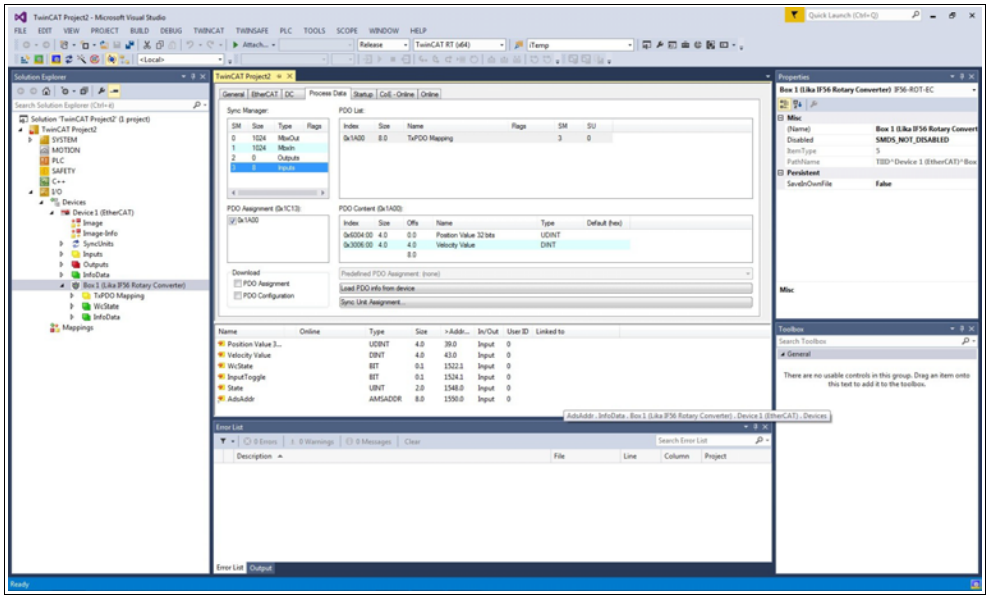


Figure 27 - Process Data tabbed page

The **Startup** tabbed page is displayed as the EtherCAT Slave has a mailbox and supports the mailbox protocol "CAN application protocol over EtherCAT" (CoE). This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the Slave in the same order as they are shown in the list.

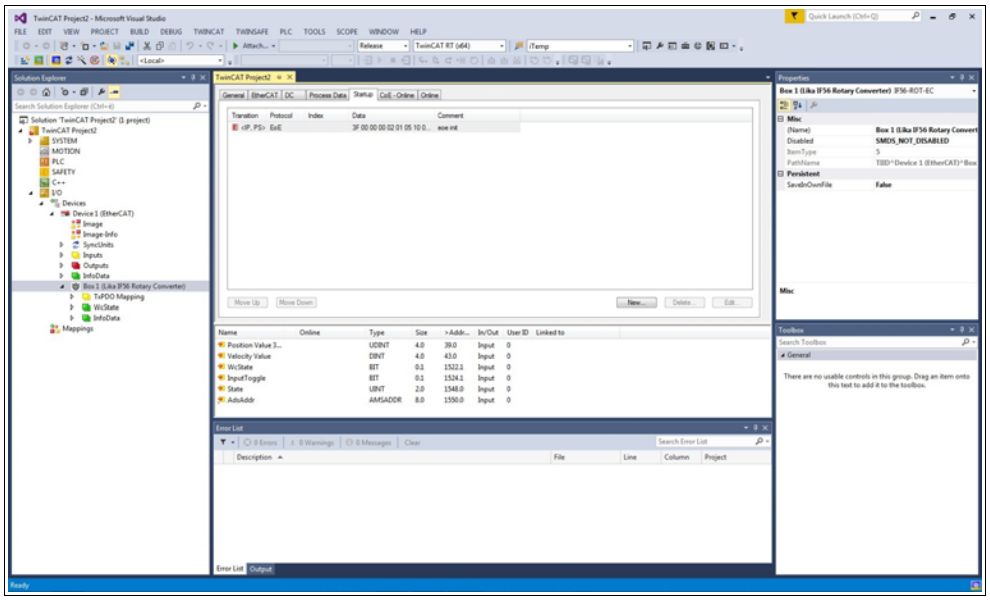


Figure 28 - Startup tabbed page

The **CoE - Online** tabbed page becomes available when you are connected to the target system and select the EtherCAT device in the IO tree. It lists the contents of the Object Dictionary of the Slave device and allows the user to change the contents of an object of this dictionary. Refer also to the "6.7 CoE Object Dictionary" section on page 71.

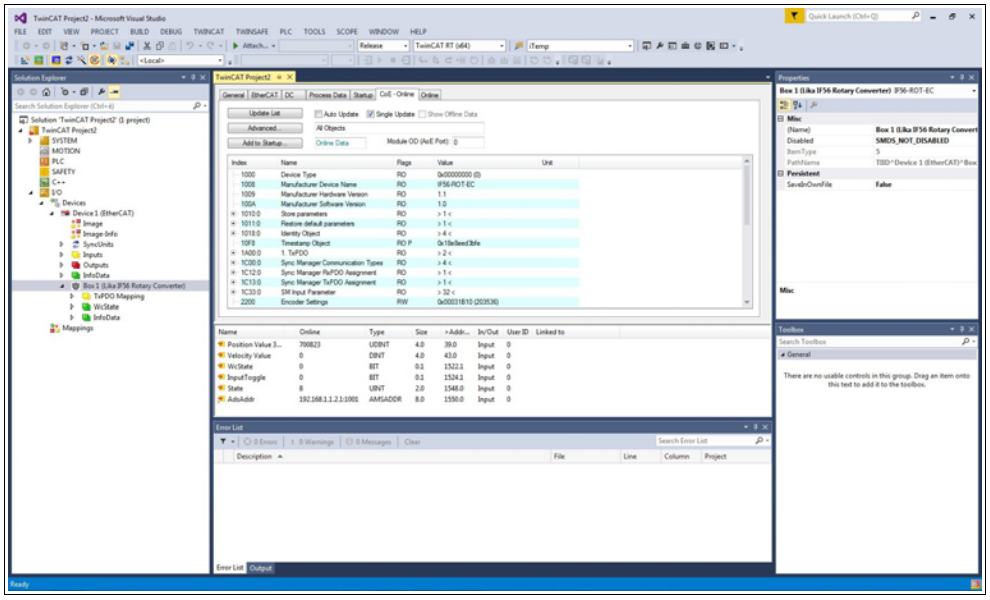


Figure 29 - CoE - Online tabbed page

The **Online** tabbed page allows the user to check and set the state of the converter as well as to update the firmware by using the File Access over EtherCAT protocol. Refer also to the "6.8 Online Data" section on page 72; and to the "6.9 Firmware update" section on page 73.

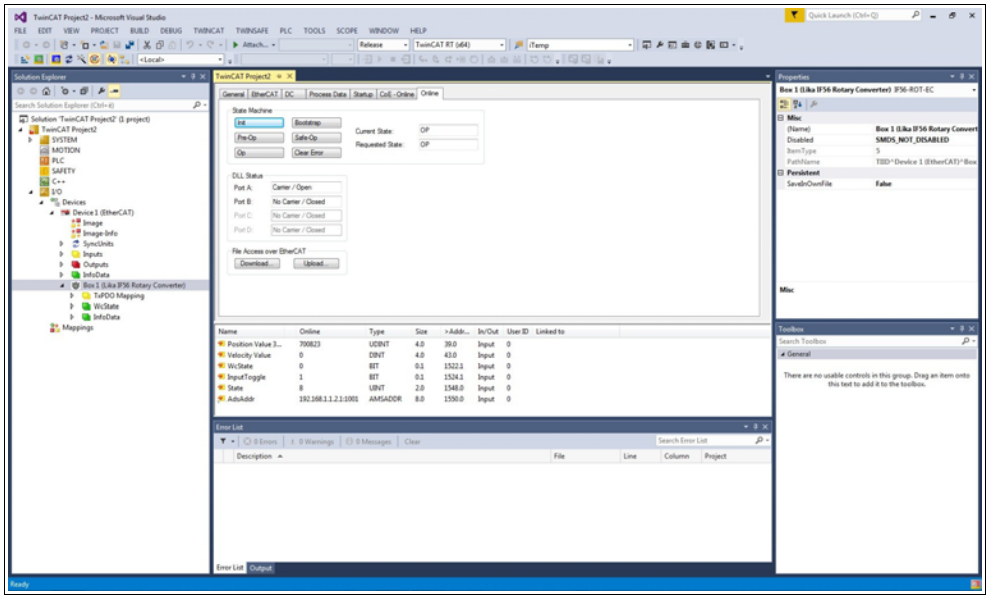


Figure 30 - Online tabbed page

If you want to know the current hardware and firmware versions of the device, select the installed **Box** and enter the **CoE - Online** tabbed page. Refer to the object **1009 Manufacturer Hardware Version** (1009-00 Manufacturer Hardware Version) and to the object **100A Manufacturer Software Version** (100A-00 Manufacturer Software Version).

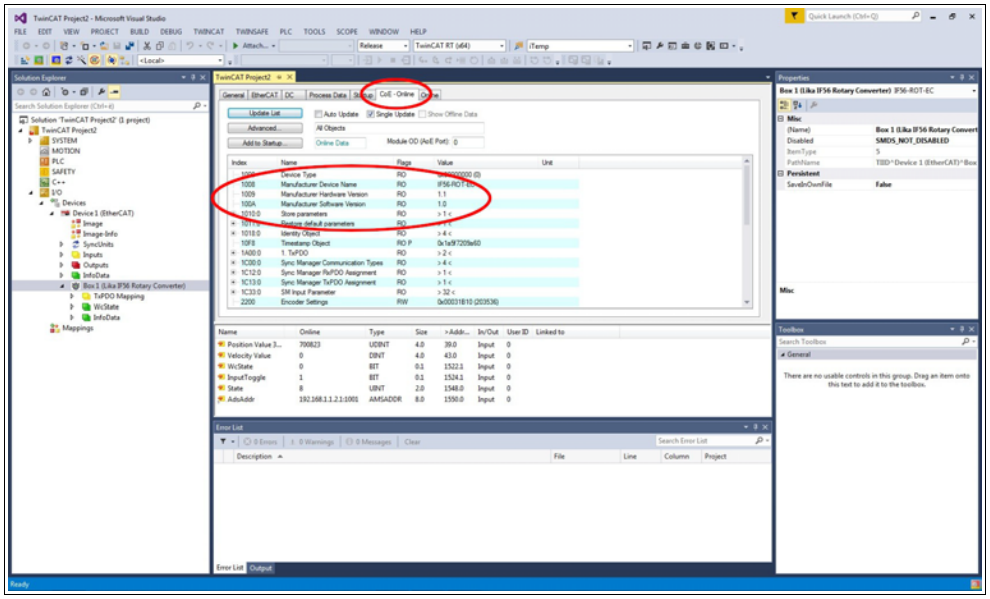


Figure 31 - Hardware and software versions

6.5 Setting the communication mode

6.5.1 Synchronous with SM3

The **DC** tabbed page allows the operator to select the operation mode of the Slave device. To enter the **DC** tabbed page select the **Box** item and the **DC** tab among the available pages.

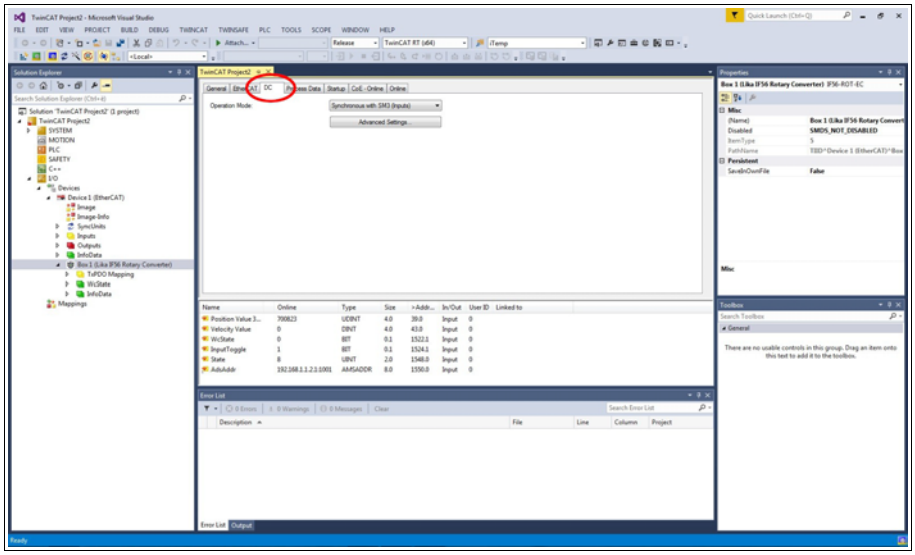


Figure 32 - DC tabbed page

The **Operation Mode** drop-down menu allows the user to select among the operation modes available in the EtherCAT Slave: **Synchronous with SM3 (Inputs)** and **Synchronous with DC (SYNC0)**.

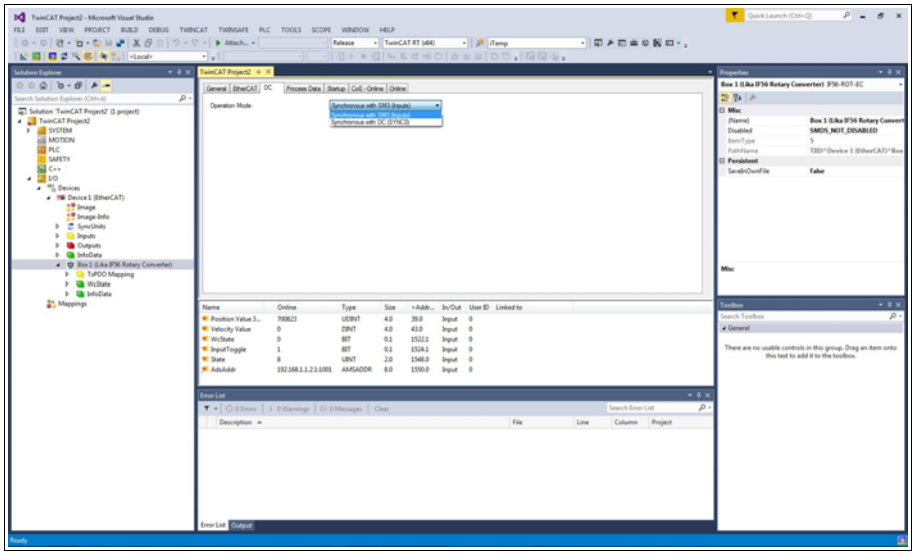


Figure 33 - Selecting the operation mode

For any further information on the Synchronous with SM3 operation mode please refer to the "Synchronous with SM3" section on page 84 and to the 1C33-00 SM Sync Manager input parameter object on page 98.

6.5.2 Synchronous with DC (SYNC0)

If you need to select the Synchronous with DC operation mode enter the **DC** tabbed page by selecting the **Box** item and the **DC** tab among the available tabbed pages. The **Operation Mode** drop-down menu allows the user to select among the operation modes available in the EtherCAT Slave: **Synchronous with SM3 (Inputs)** and **Synchronous with DC (SYNC0)**.
Select the **Synchronous with DC (SYNC0)** option.
Then press the **ADVANCED SETTINGS...** button. The **Advanced Settings** page will appear.

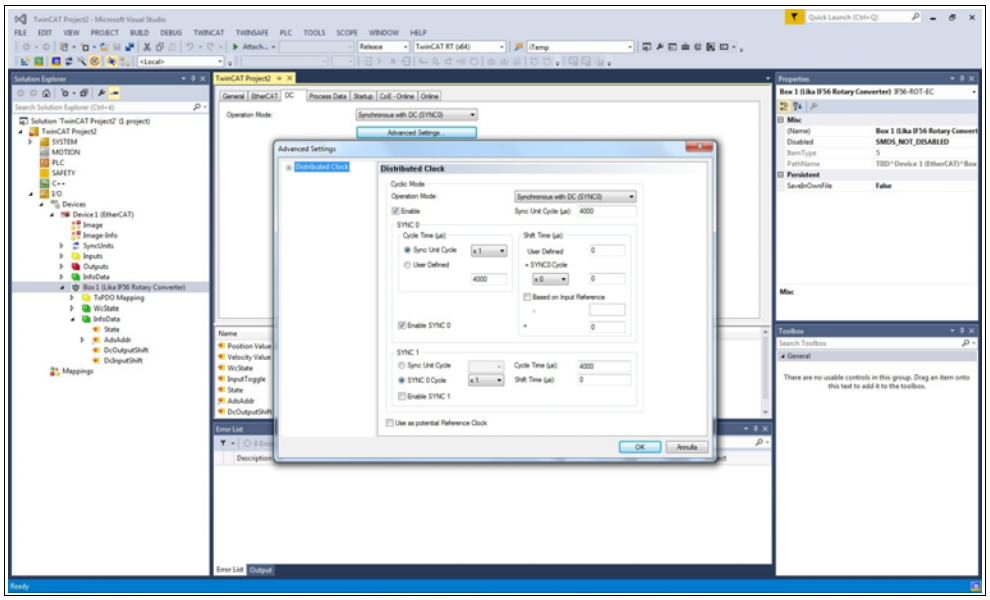


Figure 34 - Synchronous with DC operation mode

In the **SYNC 0** group box set the cycle time next to the **Sync Unit Cycle** box; the sync time is calculated as multiple (or sub-multiple) of the value set in the **Sync Unit Cycle (µs)** item right above.

For any further information on the Synchronous with DC operation mode please refer to the "Synchronous with DC SYNC0" section on page 85 and to the 1C33-00 SM Sync Manager input parameter object on page 98.

6.6 Process Data Objects

If you select the **Box** item and expand it, you can see the list of the Process Data Outputs (TxPDO). Then enter the **Process Data** tabbed page.
The **Process Data** tab shows the process data objects (TxPDO Mapping), i.e. the configuration of the process data. The input and output data of the EtherCAT Slave are represented as CANopen process data objects (PDO). This dialog allows the user to select a PDO via the **PDO List** and to vary the content of the individual PDO in the **PDO Content** list view.

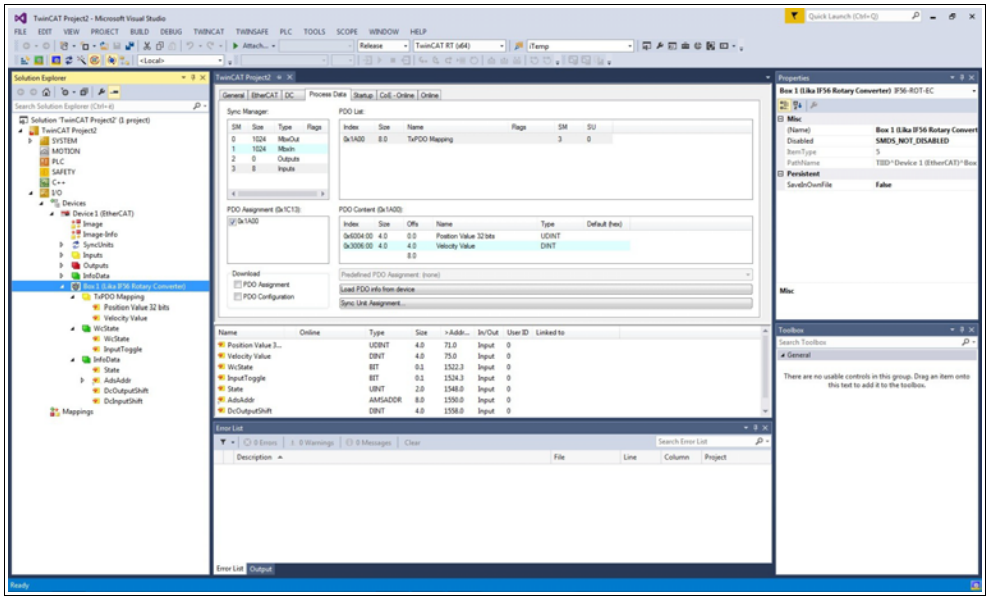


Figure 35 - Process data objects

The **Sync Manager** list view lists the configuration of the Sync Managers (SM). SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn). SM2 (outputs) is used for the output process data and SM3 (inputs) for the input process data. The **Size** column shows the size of the Sync Manager in bytes.
If an entry is selected, the corresponding PDO assignment is displayed in the **PDO Assignment** list below.

The **PDO Assignment** list contains the PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here. If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed. Please note that no RxPDOs are available in this converter. If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.
The selected entries are the PDOs participating in the process data transfer. These PDOs are displayed in the I/O tree as variables of the EtherCAT device. The names of the variables are identical to the Name parameter of the PDOs, as displayed in the **Name** column of the **PDO List**. If an entry in the **PDO assignment** list is disabled, i.e. not selected and greyed out, it indicates that this entry is excluded from the PDO assignment. To be able to select a greyed out PDO, you must deselect the PDO that prevents the greyed out PDO from being selected.

The 1A00-00 TxPDO mapping parameter object (see the **PDO Content** table) contains the mapping parameters for the PDOs the EtherCAT device is able to transmit. Sub-index 001 **01 Mapped Object 001** contains the information of the mapped application object 001: 6004-00 Position Value 32 bits object. Sub-index 002 **02 Mapped Object 002** contains the information of the mapped application object 002: 3006-00 Velocity Value object.

Process data objects can be displayed also by selecting the **TxPDO Mapping** item in the tree view.

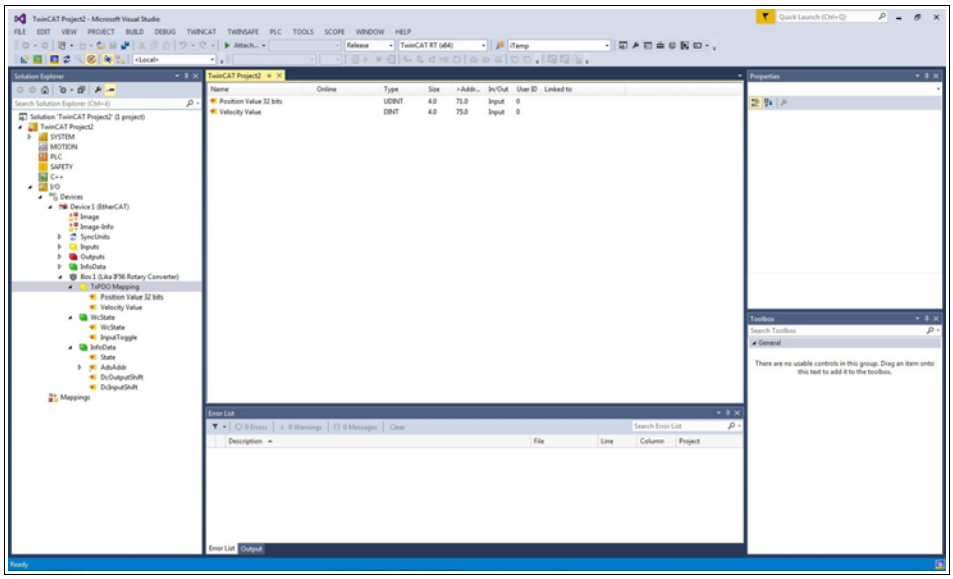


Figure 36 - TxPDO Mapping (offline)

If no value is shown under the **Online** column, press the **RELOAD DEVICES** button in the toolbar to load the created I/O devices. The current position and speed values will be displayed.

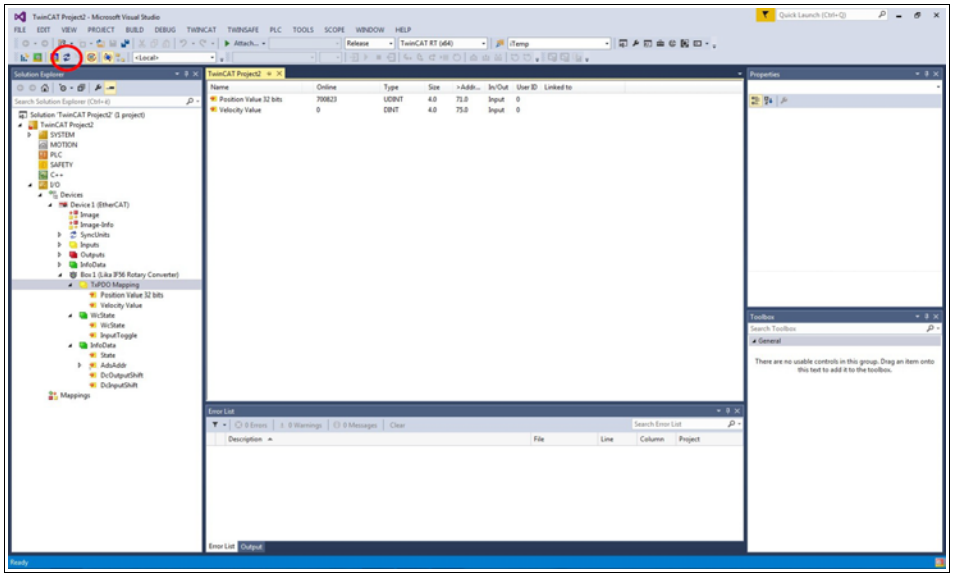


Figure 37 - TxPDO Mapping (online)

6.7 CoE Object Dictionary

The **CoE - Online** tab lists the contents of the Object Dictionary of the Slave device and allows the user to change the contents of an object of this dictionary. This is the offline version of the object dictionary as read from the XML file. To enter the **CoE - Online** tabbed page, select the installed **Box** and press the **CoE - Online** tab.

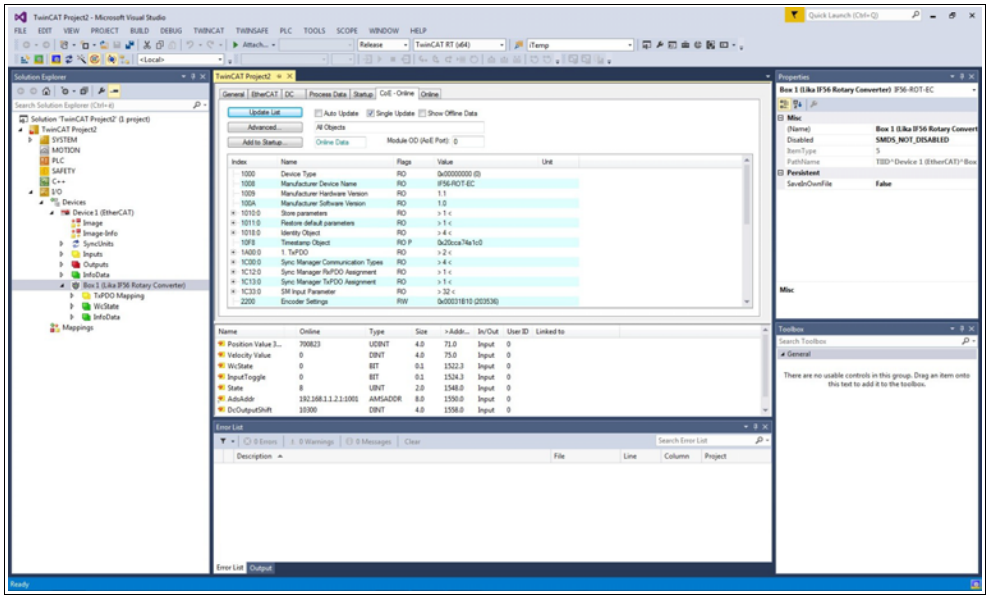


Figure 38 - CoE – Online tabbed page

Objects can also be read directly from the converter; to do this click the **Advanced...** button: the **Advanced Settings** dialog box will appear.

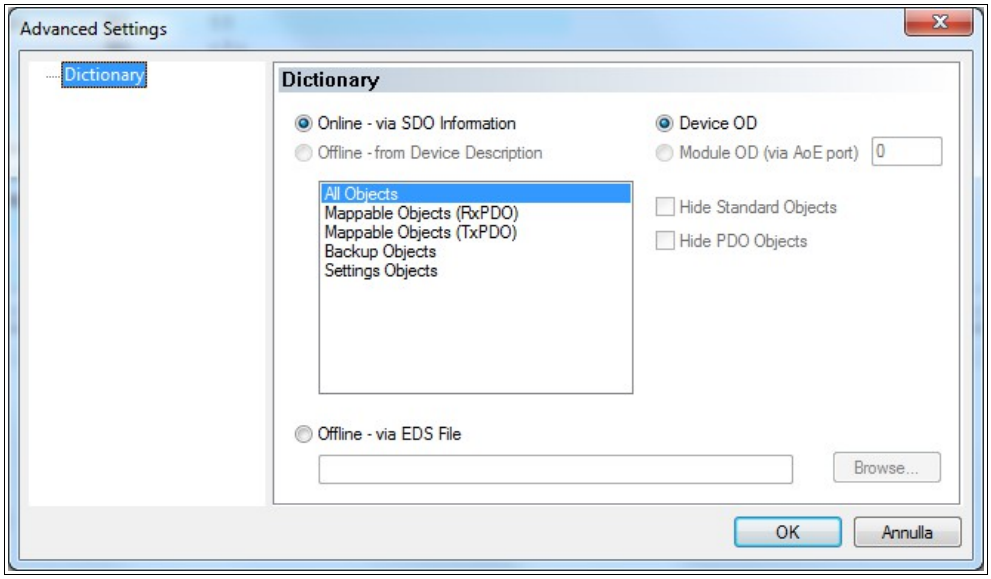


Figure 39 - CoE – Online - Advanced Settings

Select the **Dictionary** item in the tree view and then click the **Online - via SDO Information** option button in the **Dictionary** group box; press the **OK** button to confirm.

6.8 Online Data

The **Online** tabbed page allows the user to check and set the state of the converter as well as to update the firmware by using the File Access over EtherCAT protocol. To enter the **Online** tabbed page, select the installed **Box** and press the **Online** tab.

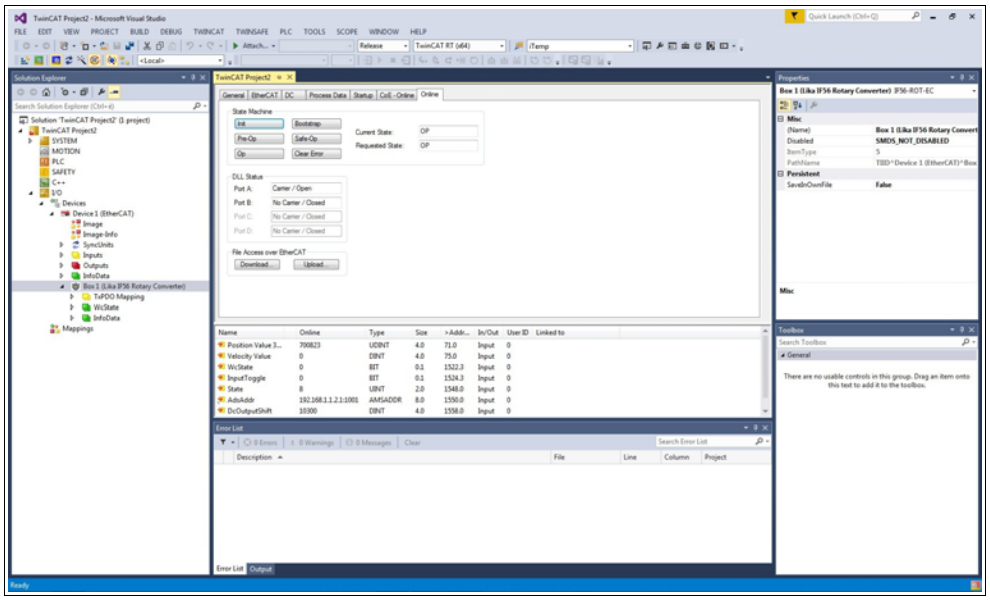


Figure 40 - Online tabbed page

To display the converter process data in real time, click the **SAFE-OP** button if you want to display inputs only; click the **OP** button if you want to display both inputs and outputs.

WARNING

The structure of Data Objects (PDOs and SDOs) requires bytes to be sent from the Least Significant Byte (LSB) to the Most Significant Byte (MSB).
On the contrary in TwinCAT write and read data from MSB to LSB.
Furthermore in TwinCAT also strings must be entered in the reverse order:

- read default values: Data byte = 64 61 6F 6Chex = "daol" in ASCII code (means "load" if read in reverse);
- save parameters: Data byte = 65 76 61 73hex = "evas" in ASCII code (means "save" if read in reverse).

6.9 Firmware update

WARNING
The firmware upgrade process has to be accomplished by skilled and competent personnel. If the upgrade is not performed according to the instructions provided or a wrong or incompatible firmware program is installed, then the unit may not be updated correctly, in some cases preventing the unit from operating.

The firmware is a software program which controls the functions and operation of a device; the firmware program, sometimes referred to as "user program", is stored in the flash memory integrated inside the unit. Lika converters are designed so that the firmware can be easily updated by the user himself. This allows Lika Electronic to make new improved firmware programs available during the lifetime of the product.
Typical reasons for the release of new firmware programs are the necessity to make corrections, improve, and even add new functionalities to the device.
The firmware upgrading program consists of a single file having .ZIP extension. It is released by Lika Electronic Technical Assistance & After Sale Service.

To update the firmware using Beckhoff TwinCAT 3 and the File Access over EtherCAT protocol proceed as follows.

1. Connect to the Slave.
2. Highlight the Slave by pressing the **Box** item of the converter you need to update: some tabbed pages for configuring and managing the device will appear. Navigate to the **Online** tabbed page.
3. Make sure the Slave is in Pre-Operational state. To check the current state of the converter see the **Current State** information field in the **State machine** group box. If required, press the **PRE-OP** button in the **State machine** group box.

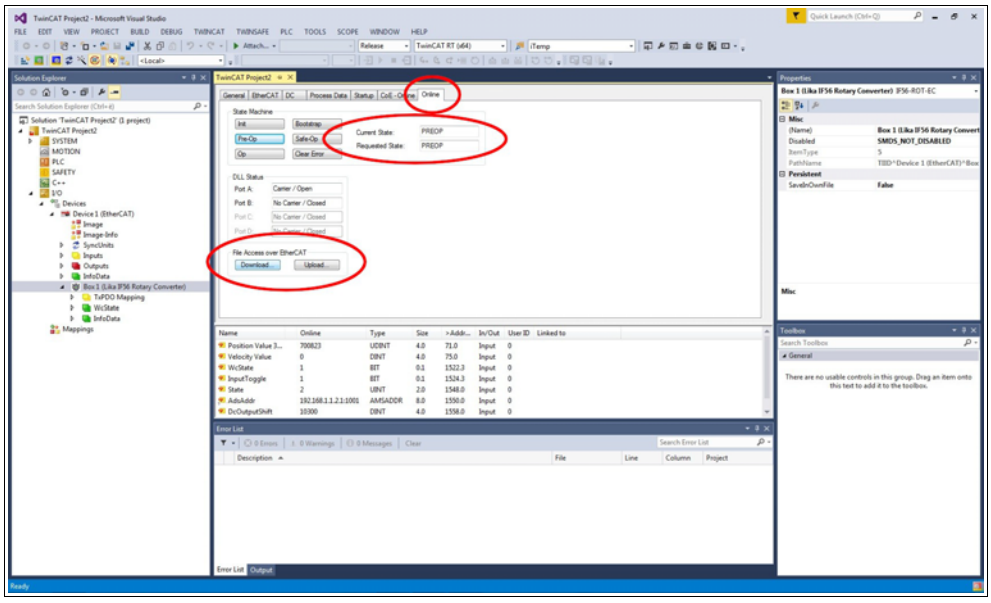


Figure 41 - Setting the State Machine

4. In the same page refer to the **File Access over EtherCAT** group box. If the **DOWNLOAD** / **UPLOAD** buttons are greyed out (disabled), make sure the **File Access over EtherCAT (FoE)** checkbox is selected.
5. To do this, enter the **EtherCAT** tabbed page first and then press the **ADVANCED SETTINGS...** button.
6. In the **Advanced Settings** page open the **Mailbox** list and then press the **FoE** command: the **FoE** group box will be displayed. Check the **File Access over EtherCAT (FoE)** checkbox is selected.

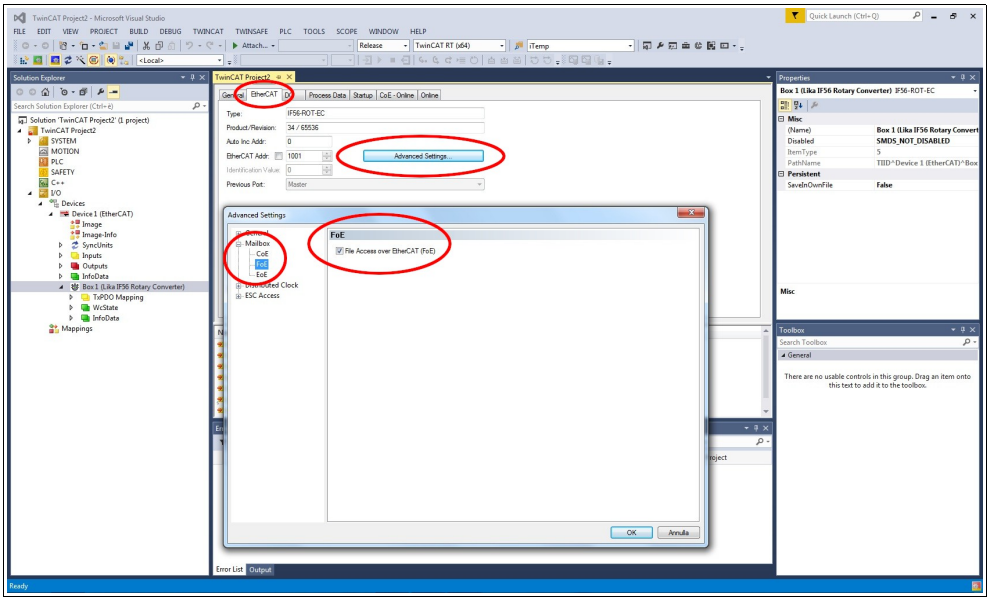


Figure 42 - Enabling the File Access over EtherCAT (FoE)

7. To download the firmware file press the **DOWNLOAD...** button in the **File Access over EtherCAT** group box in the **Online** tabbed page.

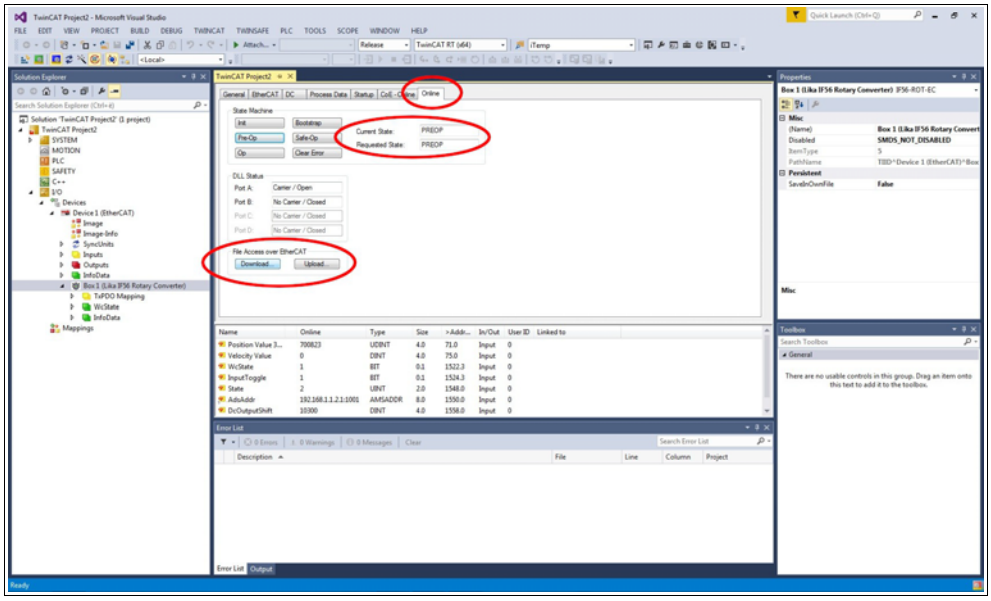


Figure 43 - Downloading a file via File Access over EtherCAT (FoE)

8. Browse for the firmware file (select "All Files (*.*)" as extension) and then click **OPEN** and **OK** to download the file. The name of the file to be downloaded must be compulsorily FWUPDATE.ZIP. ZIP files with different name are not accepted.

WARNING

After selecting the FWUPDATE.ZIP file in the **Open** page and confirming it by pressing the **OPEN** button, the **Edit FoE Name** dialog box will appear on the screen. In the **String** field the file extension will be omitted. Please add the .ZIP extension to the file name. Enter the password 00000000hex next to the **Password (hex)** item below in the page and then press the **OK** button to confirm.

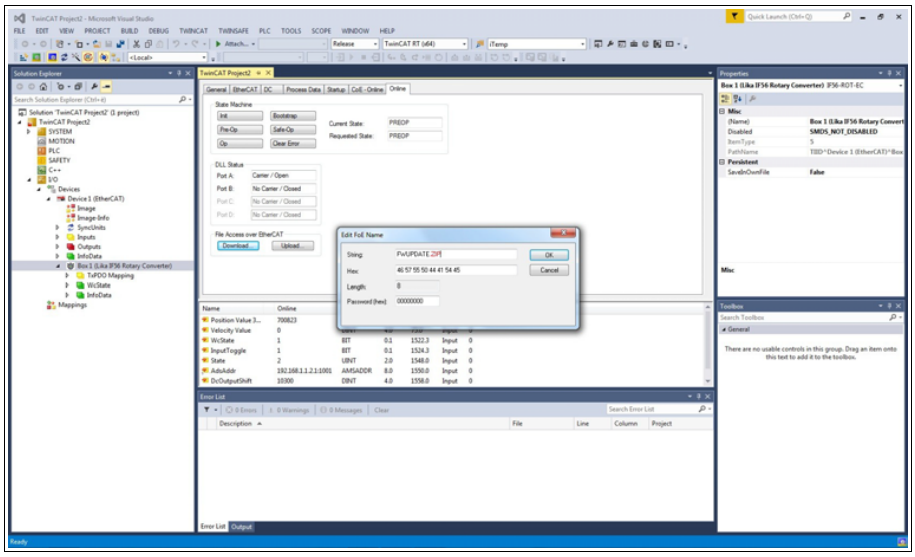


Figure 44 - Edit FoE Name dialog box

9. To trigger the new update go to **CoE - Online** tabbed page and re-scan the objects by pressing the **UPDATE LIST** button.

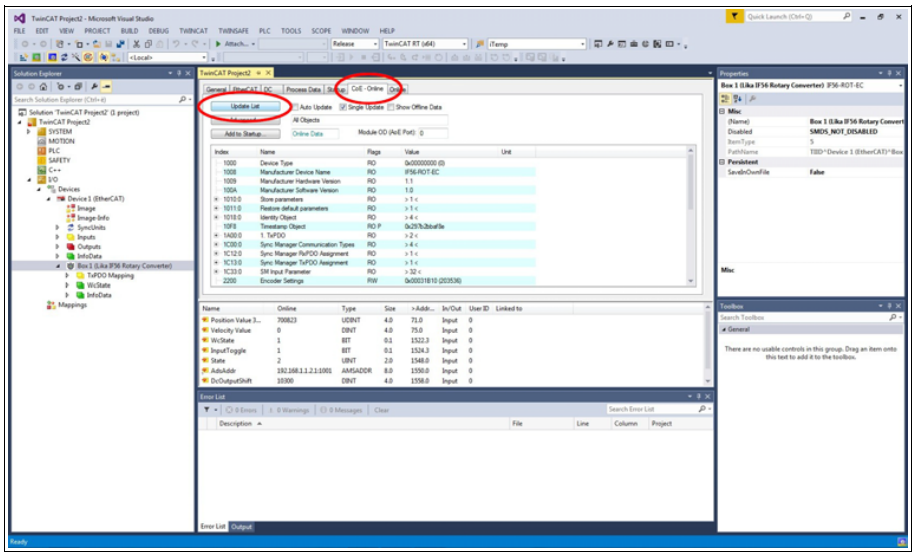


Figure 45 - Updating the object list

10. Or, in the same **CoE - Online** tabbed page, press the **ADVANCED...** button and then press the **OK** button in the **Advanced Settings** page that appears.

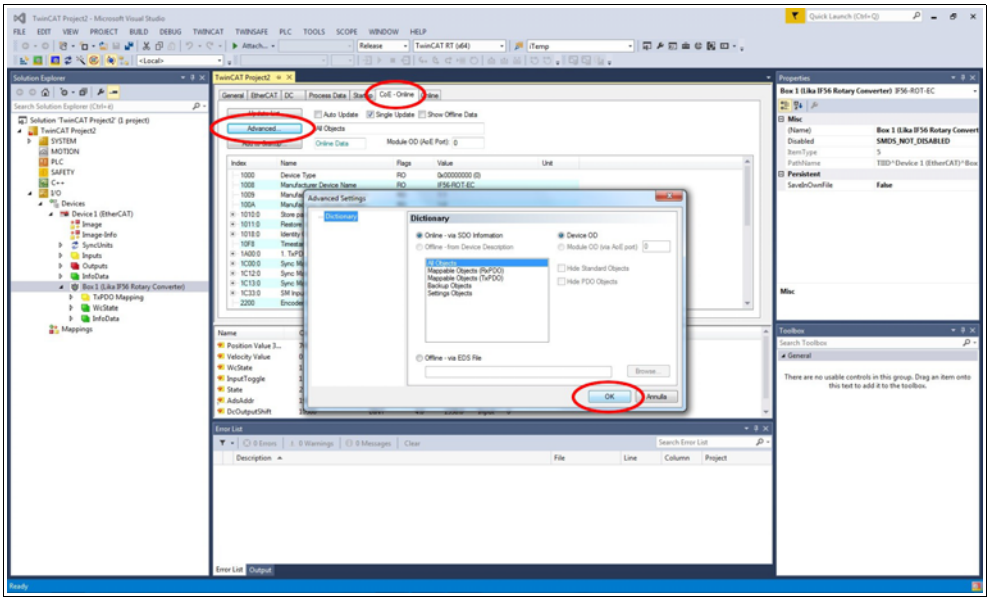


Figure 46 - Updating the object list

11. Scroll through the objects and double click on the object 6000 (see the 6000-00 Operating parameters object on page 117).
12. Set the bit 15 **Mask Upgrade Firmware** to 1 (1000 0000 0000 0000 in binary notation = 32768 dec = 8000h), confirm pressing the **OK** button and check that the PWR LED starts flashing: it shall flash red while the update is in progress.

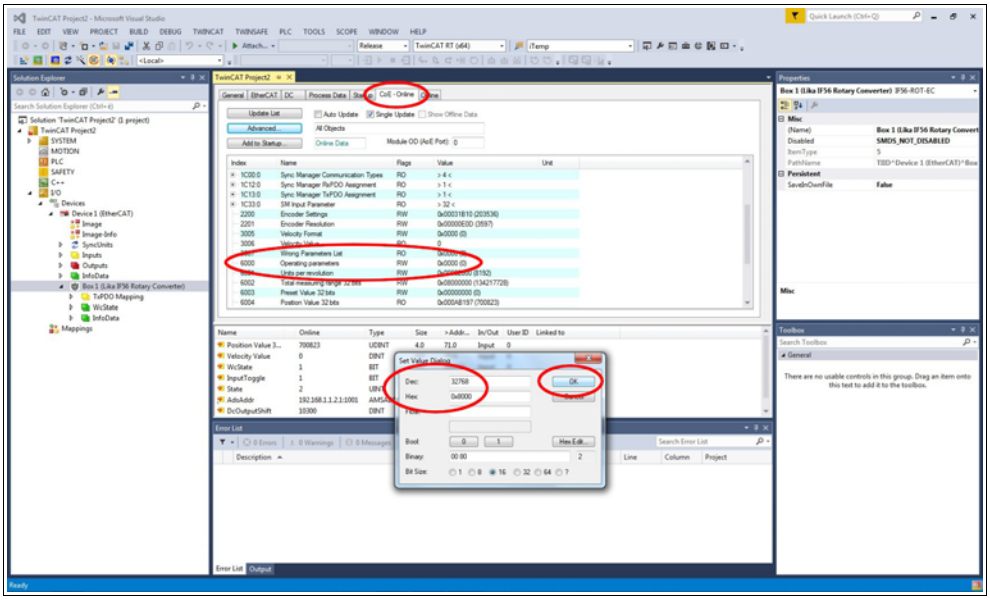


Figure 47 - Updating the object list

12. To check whether the firmware update procedure has been completed successfully enter the **CoE - Online** tabbed page and check the value next to the object **100A Manufacturer Software Version**.

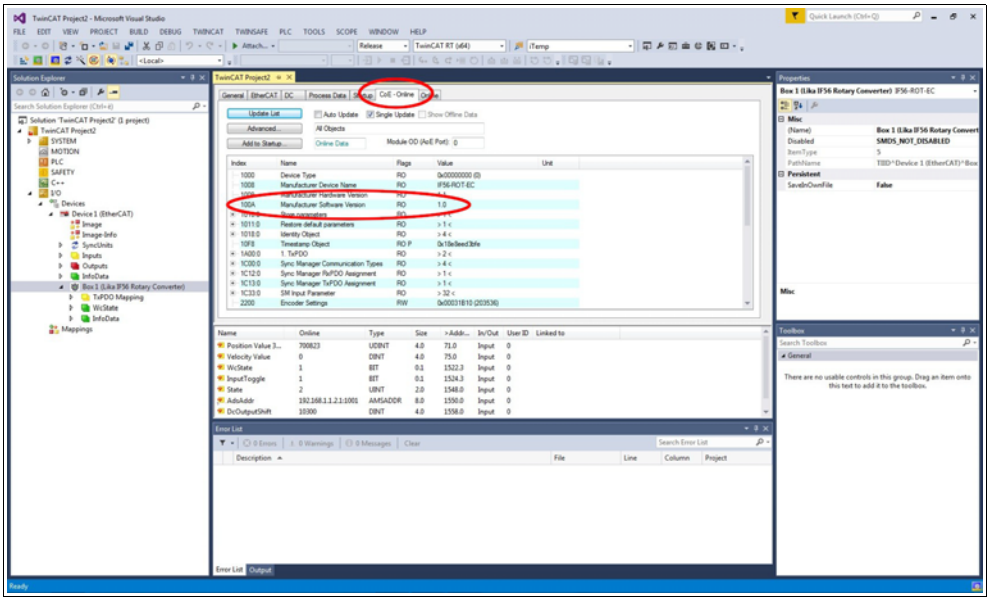


Figure 48 - Manufacturer Software Version

13. Otherwise, enter the **Online** tabbed page and press the **PRE-OP** button in the **State Machine** group box; if everything is ok, the converter enters the PREOPERATIONAL state (the **PREOP** message appears next to the **Current State** item in the same box).

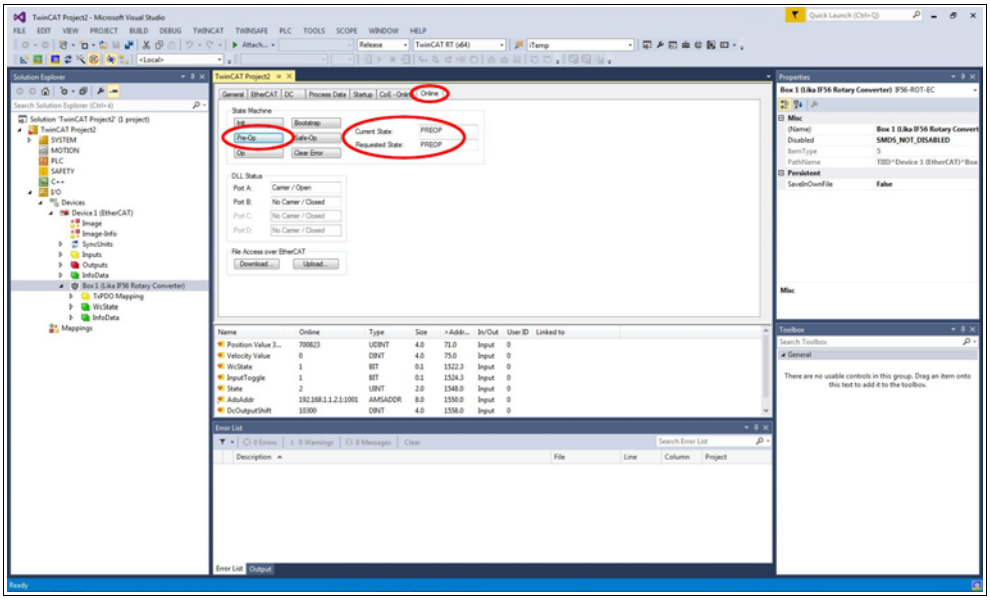


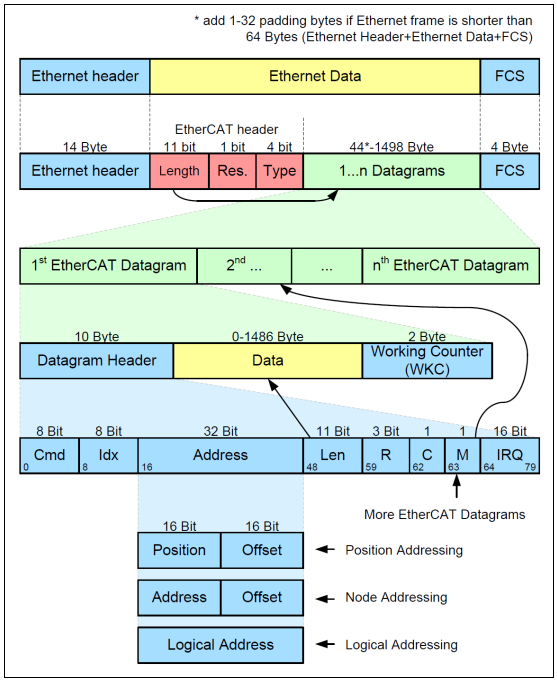
Figure 49 - Preoperational state

7 - EtherCAT® interface

7.1 Basic Information on EtherCAT® Protocol

EtherCAT is the ETHERnet for Control Automation Technology. It is a real-time Industrial Ethernet technology originally developed by Beckhoff Automation. The EtherCAT protocol which is disclosed in the IEC standard IEC61158 is suitable for hard and soft real-time requirements in automation technology, in test and measurement and many other applications.
The main focus during the development of EtherCAT was on short cycle times (≦ 100 µs), low jitter for accurate synchronization (≦ 1 µs) and low hardware costs.
EtherCAT was introduced in April 2003, and the EtherCAT Technology Group was founded in November 2003 - Meanwhile ETG has grown into the world's largest Industrial Ethernet and fieldbus organization. The ETG brings together manufacturers and users, which contribute in technical working groups to the advancement of the EtherCAT technology.

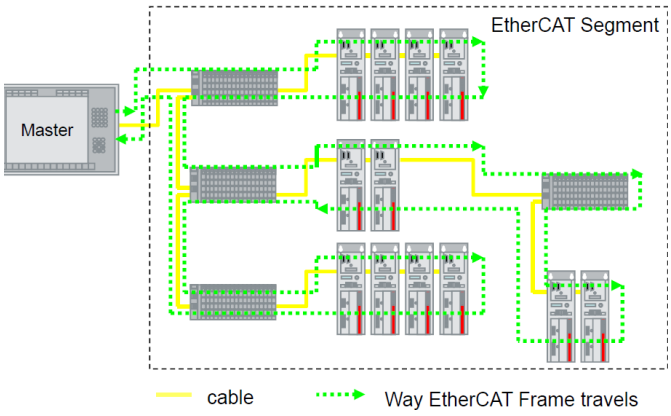
The EtherCAT protocol is designed to use the standard Ethernet dataframes for issuing data; in addition, and as regards the hardware, it is not necessary to install dedicated Masters for establishing and managing the EtherCAT communication because standard Ethernet network cards can be used. This results in a great advantage in terms of lower costs and simplicity of use because Ethernet network cards are used in standard personal computers and are easily commercially available.
An EtherCAT bus can be viewed as a single and large Ethernet device that receives and sends Ethernet telegrams; it can be considered an Ethernet subnet supported by an Ethernet dataframes structure.
However this "subnet" must be fitted with one only EtherCAT Master controller and several EtherCAT Slaves, but no Ethernet controller with downstream microprocessor must be present.
Here follows the structure of the Ethernet frame with EtherCAT:



Inside the Ethernet frames, data is transmitted among Master and Slaves using PDO (Process Data Objects) protocol. Each PDO message has inside one or more addresses for issuing data to the Slaves; data + address/es (and additional elements such as a validation checksum) joined together form an EtherCAT telegram (Datagram).
An EtherCAT frame can contain several telegrams and a complete control cycle often requires more than one frame.

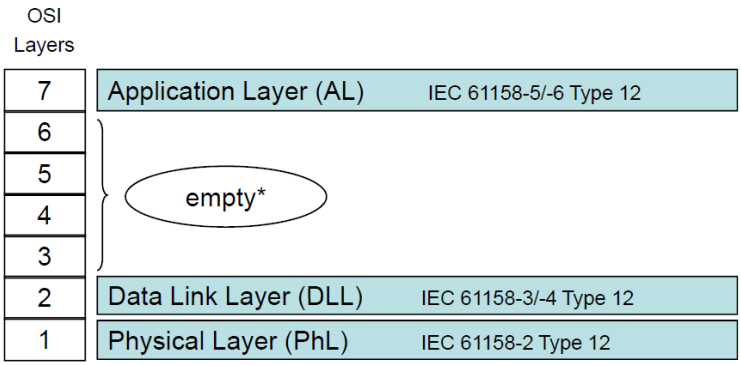
7.1.1 Data transfer

Usually, in a data bus system, Master controller sends online a data request and then waits for data to be processed and sent back from each Slave node; this does not comply with a real-time system because the Master receives data from the Slaves in different moments and the whole system cannot be synchronized.
In EtherCAT the real-time characteristic of the system is quite improved because data is processed "on-the-fly", using one single frame to acquire all data from all Slaves.
In fact each Slave node which is addressed the data reads the frame sent by the Master while the telegram passes through the device; similarly, input data is inserted while the telegram passes through. Then the telegram is forwarded to the next device. Telegrams are only delayed by a few nanoseconds.
The last Slave issues back the complete frame to the Master with all the requested data (again passing through all the Slaves).



This efficient data flow is guaranteed by the 100BASE-TX full-duplex structure of the EtherCAT bus which is equipped with two separated lines for transmitting and receiving data.
Moreover the protocols exchange takes place inside the hardware and it is thus independent from the CPU and the software processing.

7.1.2 ISO/OSI Layer model

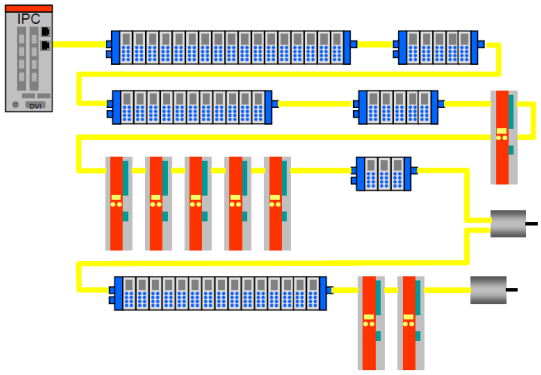


* "Empty" means that the layer behaviour exists, but is not shown explicitly.

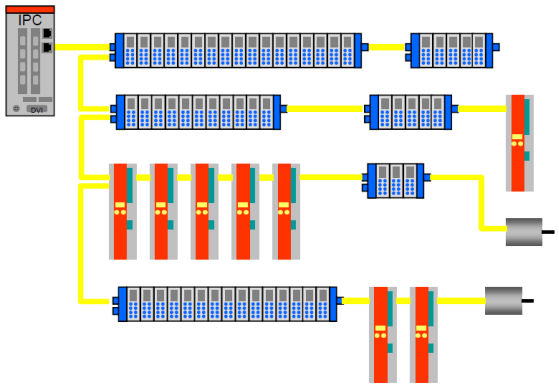
7.1.3 Topology

Several topologies of connection are supported by the EtherCAT networks: Line, Tree, Daisy chain, Star, ...). EtherCAT networks can be configured in almost any topology in the same structure. The maximum length of the cable between two Slaves is 100 m / 328 ft; standard EtherCAT cables commercially available can be used.
The choice of the topology depends on the structural characteristics of the plant and it is made in order to reduce the complexity and time for cabling.
Inside an EtherCAT network up to 65,535 devices can be connected.
Some topology examples are shown in the Figures below:

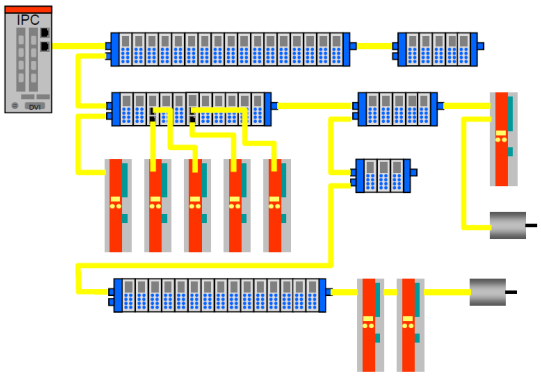
LINE topology:



TREE topology:



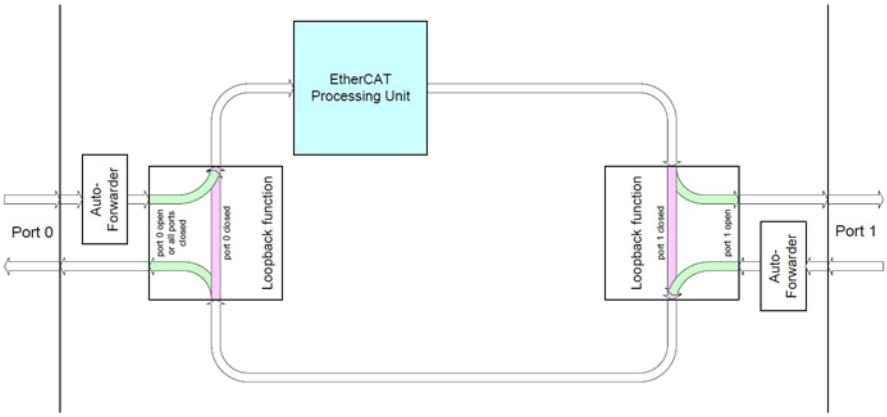
DAISY CHAIN with drop lines topology:



7.1.4 Line Termination

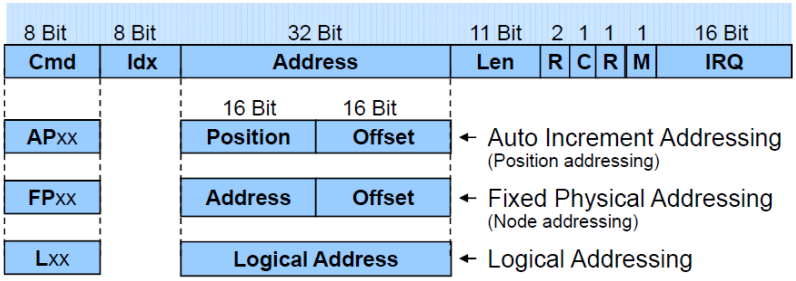
The EtherCAT network needs no line termination because the line is terminated automatically; in fact every Slave is able to detect the presence of downstream Slaves.
An EtherCAT Slave is able to detect the presence of the signal in the outgoing line (Port 0) or in the return line (Port 1).
If the Slave is not able to detect the signal in its return line, then it closes the communication ring by short-circuiting the TX signal of its outgoing line with the RX signal of its return line; in this way a telegram received through the outgoing line is processed and sent back through the TX of the return line.

The Slave sends a "carrier signal" or a telegram on TX of the outgoing line continuously and, once the next Slave is connected again, a signal on RX of the return line is detected again; so the short circuit is removed and the telegrams are sent on TX of the outgoing line.



7.1.5 Addressing

It is not necessary to assign a physical address to the device (for instance using a DIP switch) because the addressing of the Slave is automatic at power on during the initial scanning of the hardware configuration.



The field for addressing is 32-bit long; there are three kinds of addressing:

- Auto Increment Addressing = Position Addressing: 16 bits indicate the physical position of the Slave inside the network while 16 bits are scheduled for local memory addressing; when the Slave receives the frame then it increments the position address and the Slave receiving the address 0 is the addressed device;
- Fixed Addressing = 16 bits indicate the physical address of the Slave inside the network while 16 bits are scheduled for addressing the local memory;
- Logical Addressing = the Slave is not provided with its own individual address, but it can read and write data in a section of the total memory space available (4 Gigabytes according to EtherCAT specifications).

7.1.6 Communication mode

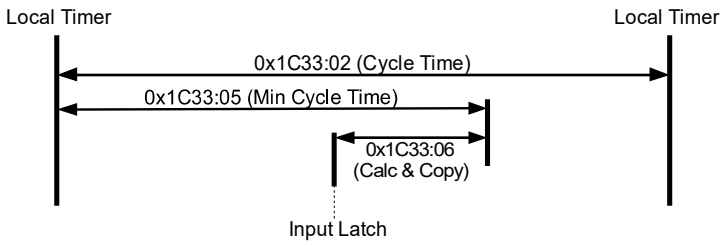
Lika converters with EtherCAT interface support the following operating modes:

- FreeRun: asynchronous mode;
- SM3 event: synchronous mode;
- DC: distributed clock synchronization mode (synchronous mode).

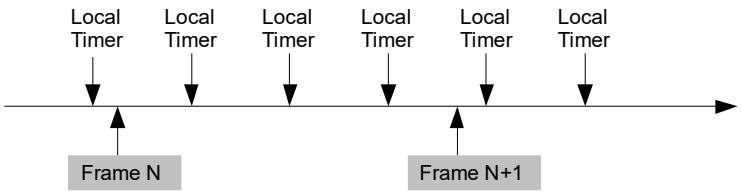
For a system that requires high performances in real time (closed-loop applications) we suggest using DC mode; if real time requirements are not so mandatory SM3 mode can be used instead.
A reference parameter is the "Jitter": it represents the temporal fluctuation of the instant data sampling; in other words data sampled by the micro-controller is available in ECAT DPRAM memory after a certain time and the measure of the variability over time is the "jitter".

FreeRun

The FreeRun mode is supported only when the converter is in **PREOP** state and the parameters (position and speed included) are read asynchronously. You can check and control IO by using the FreeRun mode without having to activate the configuration and write a PLC logic. This tool is very helpful during the commissioning stage to ensure all the IOs are functioning properly.
When the FreeRun mode is active, you will see that TwinCAT state indicator will flash blue and red continuously. Simply open the **Online** tabbed page of the IO you would like to control and write a value.
Asynchronous mode: the encoder position is sampled directly from EtherCAT frame sent by the Master; the position update is performed by an internal timer of the controller every 500 microseconds.



This operating mode has a sampling jitter up to 500 microseconds and can be chosen only when cycle times are quite longer than the jitter if we want to ensure a sufficient real-time system performance.

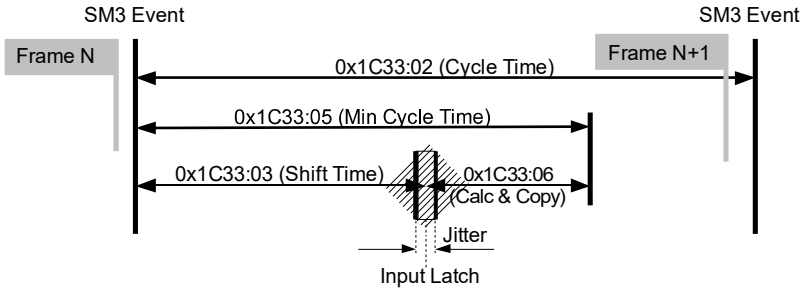


Description	Min	Typ	Max	
Jitter	0		500	µs
Cycle Time	500		64000	µs

See the 1C33-00 SM Sync Manager input parameter entry on page 98.

Synchronous with SM3

In this mode data is sampled and then copied into the Sync Manager buffer as soon as previous data was read from the Master (SM event); in this way new sampled data is synchronous with the Master readings.



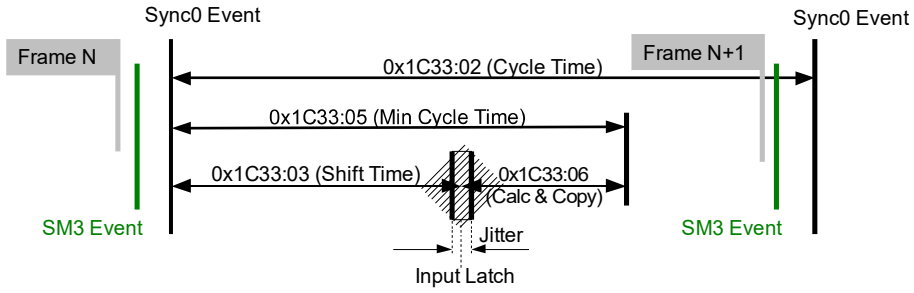
New data will be read by the Master at the next cycle (following SM3 event), so if the cycle time is too long, data could be relatively old for a real-time system.
The main advantage is that data is updated exactly when Master is reading (synchronous mode).

Description	Min	Typ	Max	
Jitter	0		7.2	µs
Cycle Time	125		64000	µs

See the 1C33-00 SM Sync Manager input parameter entry on page 98.

Synchronous with DC SYNC0

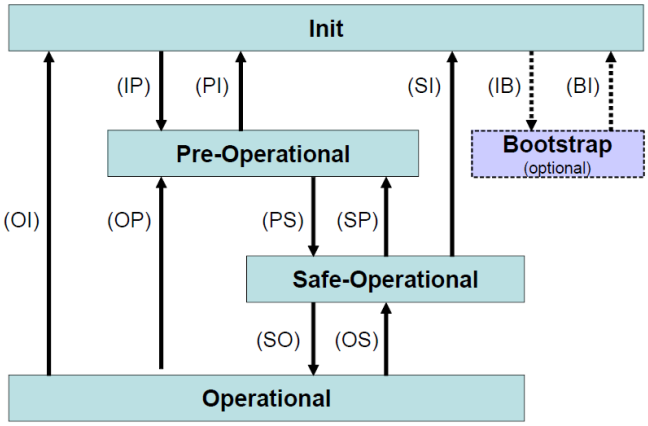
In this operating mode data is sampled and then copied into the Sync Manager buffer simultaneously at SYNC0 event generated by the ESC capture/compare unit.
Time required for accomplishing these operations is set in the **1C33-00 SM Sync Manager input parameter** object; in particular in the **03 Shift Time** entry (1C33hex, sub3) and in the **06 Calc and copy time** entry (1C33hex, sub6).



In this operating mode "Jitter" is a fundamental parameter in the sampling of two consecutive data. The main advantage of this mode is that there is a direct relation between the sampling instant and the absolute time of the system; in this way, if we know the shift times of the Slaves, we can have an exact image of the system at a given moment (with a tolerance equal to the jitter).

Description	Min	Typ	Max	
Jitter	0	100	200	µs
Cycle Time	125		64000	µs

7.1.7 EtherCAT State Machine (ESM)



EtherCAT Slave is a state machine; the communication and the operating characteristics depend on the current state of the device:

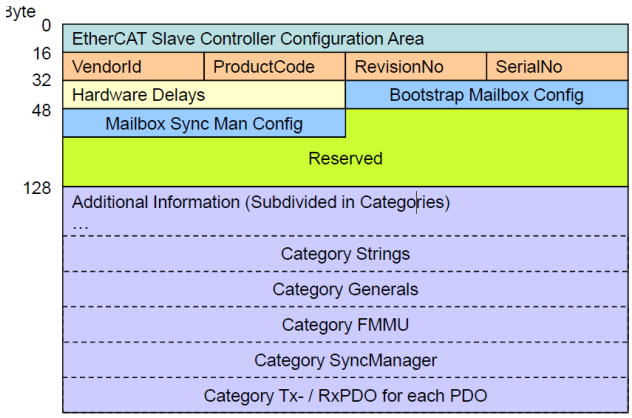
- INIT: it is the default state after power-on; in this state there is not direct communication between the Master and the Slave on the Application Layer; some configuration registers are initialized and the Sync Managers are configured.
- PRE-OPERATIONAL (PREOP): in this state the mailbox is active; both the Master and the Slave can use the mailbox and its protocols for exchanging specific initialization parameters of the application. Exchange of Process Data (PDO) is forbidden. In this state the FoE protocol is used for firmware download.
- SAFE-OPERATIONAL (SAFEOP): in this state the Master and the Slave can issue only input process data, while the output process data is still in the SAFE-OPERATIONAL state;
- OPERATIONAL (OP): in this state the Master and the Slave are enabled to send both input process data and output process data.
- BOOTSTRAP (BOOT): no process data communication. Communication only via mailbox on Application Layer available. Special mailbox configuration is possible, e.g. larger mailbox size.

The current state of the Slave is signalled through the green **RUN** LED, see on page 33.

7.1.8 Slave configuration

The configuration of the Slave communication characteristics (Sync Manager configuration, addresses, synchronization modes, PDO mapping, ...) can be made both by using the XML file (EtherCat Slave Information - ESI) or by loading data directly from EEPROM (Slave Information Interface SII).

EEPROM content (SII):



7.1.9 Timing and Synchronization

The main feature of EtherCAT is its almost ideal representation of a real-time system.
Hence the Master has to synchronize all the Slaves at the same time in order to build a system where all nodes have the same reference time, this goal can be achieved by using "distributed clocks".
The Master downloads its clock into one of the Slaves (customarily the first Slave) which becomes the reference clock for all the Slaves in the network; so it has the task of synchronizing the other Slaves. The Master controller periodically sends a special synchronization-telegram where the reference Slave writes its own "current time". This telegram is then sent to all the other Slaves that, in this way, provide for a new re-synchronization of their own clock in order to avoid possible drifts.
This synchronization of the reference time is very important in order to have an up-to-date "snapshot" of the system and accordingly take simultaneous actions in high sensitive applications such as the coordination in axis control operations.
Besides, the EtherCAT Slave Controller (ESC) is fitted with a capture/compare unit that provides accurate synchronization signals (SYNC0 or interrupts): they are sent to the local micro-controller so that it is able to synchronize its own clock to the Slaves clock.

Sync Manager
Sync Manager has the task of synchronizing data transfer between the Master and the Slave and prevents the same memory area from being written by different events.
There are two synchronization modes:

- 3-Buffer Mode;
- 1-Buffer Mode.

Synchronisation mode is initialized through the XML file or by loading data directly from EEPROM (SII).

Buffered Mode (3-Buffer Mode)
In this mode new data can be accessed at any time by both the EtherCAT Master and the ESC controllers; no timing restrictions are imposed.
Three buffers are necessary (three consecutive memory areas); one buffer is always available to the ESC controller for writing and one buffer always contains updated data to be read by the Master.
Customarily this mode is used for cyclic data exchange, i.e. process data communication.

Mailbox Mode (1-Buffer Mode)
In this mode a "handshake" between the Master and the Slave must be used; in fact one only memory buffer is available to both the Master and the Slave for writing and reading; the Master (or the Slave) is enabled to write only when the buffer is empty, that is when the Slave (or the Master) has finished reading the data buffer. And vice versa: the Master (or the Slave) is enabled to read only when the buffer is empty, that is when the Slave (or the Master) has finished

writing the data buffer. The mailbox mode is typically used for application layer protocols and exchange of acyclic data (e.g. parameter settings).

The converter features four Sync Managers, see the 1C00-00 Sync Manager Communication Types object on page 97:

- **Sync Manager 0 - 01 SM MailBox Receive (SM0)**
  Used for mailbox write transfers (Master to Slave).
  The module has a configurable write mailbox size with default size of 1024 bytes, including 10 bytes of the relevant protocol headers and padding.
- **Sync Manager 1 - 02 SM MailBox Send (SM1)**
  Used for mailbox read transfers (Slave to Master).
  The module has a configurable read mailbox size with default size of 1024 bytes, including 10 bytes of the relevant protocol headers and padding.
- **Sync Manager 2 - 03 SM PDO output (SM2)**
  It contains the RxPDOs (i.e., Sync Manager 2 holds the Read Process Data).
- **Sync Manager 3 - 04 SM PDO input (SM3)**
  It contains the TxPDOs (i.e., Sync Manager 3 holds the Write Process Data).

7.2 CANopen Over EtherCAT (CoE)

Lika converters are Slave devices and support the "CanOpen Over EtherCAT" (CoE) mode for data transfer. In particular, they support the "CANopen DS 301 Communication profile", Class 2 and the "CANopen DS 406 Device profile for encoders".
For any omitted specification on the EtherCAT® protocol, please refer to the "ETG.1000 EtherCAT Specification" document available at the address **www.ethercat.org**.
For any omitted specification on the CANopen® protocol, please refer to the "CiA Draft Standard Proposal 301. Application Layer and Communication Profile" and to the "CiA Draft Standard 406. Device profile for encoders" documents available at the address **www.can-cia.org**.

7.2.1 XML file

EtherCAT® converters are supplied with their own XML file. It can be downloaded from Lika Electronic web site, see at **www.lika.biz > ENCODER INTERFACES > ENCODER INTERFACES**.
For more information please refer to the "6.3 Adding the XML file" section on page 54.
For any information on the firmware update procedure refer to the "6.9 Firmware update" section on page 73; and to the "8.6 Firmware update" section on page 156.

If you want to know the current hardware and firmware versions of the device, select the installed **Box** and enter the **CoE - Online** tabbed page. Refer to the object **1009 Manufacturer Hardware Version** (1009-00 Manufacturer Hardware Version) and to the object **100A Manufacturer Software Version** (100A-00 Manufacturer Software Version).

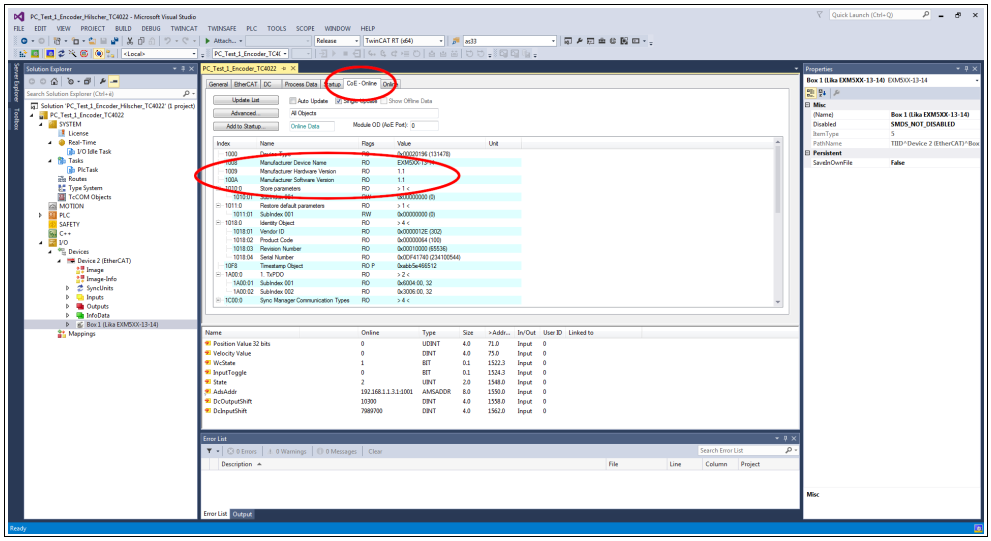


Figure 50 - Hardware and software versions

7.2.2 Communication messages

EtherCAT Datagram of CoE mode has the following structure:

Mbx Header	CoE Cmd			Cmd specific data
type = 3	Number	res	Type	
6 bytes	9 bits	3 bits	3 bits	0 ... 1478 bytes

Mbx Header = 3 CoE mode

Number = 0 in case of SDO messages
≠ 0 in case of PDO messages, it defines the type of service

res reserved bits

Type = 0 reserved
= 1 Emergency message
= 2 SDO request
= 3 SDO response
= 4 Transmitted PDO (TxPDO)
= 5 Received PDO (RxPDO)
= 6 Remote transmission request of TxPDO
= 7 Remote transmission request of RxPDO
= 8 SDO information
= 9 ... 15 reserved

Cmd specific data PDO messages: are the process data, e.g. position value
SDO messages: standard CANopen frame

Transmit (tx) or receive (rx) "Type" is viewed from the Slave side.

7.2.3 Process Data Objects (PDO)

PDO messages are used for transmitting or receiving process data in real time; data to be transmitted or received is defined in PDO Mapping and managed by Sync Manager PDO Mapping.

7.2.4 Service Data Objects (SDO)

SDO messages are issued via Mailbox (low priority data); Segmented SDO Service and SDO Complete Access are not supported (transfer of low size data and one sub-index at a time).

"CoE Cmd type" = 2 or 3

Structure of "Cmd specific data":

Cmd specific data				
SDO control	Index	Sub index	Data	Data optional
8 bits	16 bits	8 bits	32 bits	1 ... 1470 bytes

SDO control	standard CANopen SDO Service
Index	parameter index
Sub index	parameter sub-index
Data	parameter value
Data optional	optionally, more than 4 bytes of data can be sent in one frame. Full mailbox size usable.

Index and sub-index values are described in the "Object dictionary".

7.2.5 Object dictionary

The most important part of a device profile is the Object Dictionary. The Object Dictionary is essentially a grouping of objects accessible via the network in an ordered, pre-defined mode. Each object within the dictionary is addressed using a 16-bit index.
The Object Dictionary can contain a maximum of 65,536 entries.
The user-related objects are grouped in three main areas: the Communication Profile Area, the Manufacturer Specific Profile Area and the Standardised Device Profile Area. The objects are all described in the XML file.

The **Communication Profile Area** at indexes from 1000h to 1FFFh contains the communication specific parameters for the EtherCAT network. These entries are common to all devices. PDO objects and SDO objects are described in this section. The Communication Profile Area objects comply with the "CiA Draft Standard Proposal 301. Application Layer and Communication Profile". Refer to the "Communication Profile Area objects (DS301)" section on page 93.

The **Manufacturer Specific Profile Area** at indexes from 2000h to 5FFFh is free to add manufacturer-specific functionality. Refer to the "Manufacturer Specific Profile Area objects" section on page 100.

The **Standardised Device Profile Area** at indexes from 6000h to 9FFFh contains all data objects common to a class of devices that can be read or written via the network. The device profiles may use entries from 6000h to 9FFFh to describe the device parameters and the device functionality. The Standardised Device Profile Area objects comply with the "CiA Draft Standard

406 CANopen Device profile for encoders". Refer to the "Standardised Profile Area objects (DS406)" section on page 117.

In the following pages the objects implemented are listed and described as follows:

Index-subindex Object name
[data types, attribute]

- Index and sub-index are expressed in hexadecimal notation.
- Attribute:

ro = read only access
rw = read and write access

Signed8 / Unsigned8 data type:

Process data bytes							
byte 4							
7	6	5	4	3	2	1	0
MSbit		...					LSbit

Signed16 / Unsigned16 data type:

Process data bytes	
byte 4	byte 5
LSByte	MSByte

Signed32 / Unsigned32 data type:

Process data bytes			
byte 4	byte 5	byte 6	byte 7
LSByte	...	...	MSByte

NOTE
Always save the new values after setting in order to store them in the non-volatile memory permanently. Use the **1010-01 Store parameters** object, see on page 94.
Should the power supply be turned off all data that has not been saved previously will be lost!

Communication Profile Area objects (DS301)

1000-00 Device Type

[Unsigned32, ro]
It contains the information about the device type. The object describes the type of device and its functionality.
Default = 0000 0000h = Generic device

1008-00 Manufacturer Device Name

[String, ro]
It shows the name of the device assigned by the manufacturer, expressed in hexadecimal ASCII code.
Default = 494635362D524F542D4543h = "IF56-ROT-EC" = IF56 converter with EtherCAT interface for SSI and BiSS rotary encoders
494635362D4C494E2D4543h = "IF56-LIN-EC" = IF56 converter with EtherCAT interface for SSI and BiSS linear encoders

1009-00 Manufacturer Hardware Version

[String, ro]
It shows the hardware version of the device, expressed in hexadecimal ASCII code.

EXAMPLE
312E31h = 1.1 = Hardware version 1.1

Default = device dependent

100A-00 Manufacturer Software Version

[String, ro]
It shows the software version of the device, expressed in hexadecimal ASCII code.

EXAMPLE
312E30h = 1.0 = Software version 1.0

Default = device dependent

1010-01 Store parameters

[Unsigned32, rw]
Use this object to save all parameters on the non-volatile memory.
Write "**save**" in hexadecimal ASCII code in the data bytes:

Master → Converter

Cmd specific data							
Cmd	**Index**		**Sub**	**Data**			
23	10	10	01	73	61	76	65
				s	a	v	e

Converter → Master (confirmation)

Cmd specific data							
Cmd	**Index**		**Sub**	**Data**			
60	10	10	01	00	00	00	00

1011-01 Restore default parameters

[Unsigned32, rw]
This object allows the operator to restore all parameters to default values. The default parameters are set at the factory by Lika Electronic engineers to allow the operator to run the device for standard operation in a safe mode. A list of machine data and relevant default parameters preset by Lika Electronic engineers is available on page 161.
Write "**load**" in hexadecimal ASCII code in the data bytes:

Master → Converter

Cmd specific data							
Cmd	**Index**		**Sub**	**Data**			
23	11	10	01	6C	6F	61	64
				l	o	a	d

Converter → Master (confirmation)

Cmd specific data							
Cmd	**Index**		**Sub**	**Data**			
60	11	10	01	00	00	00	00

NOTE
To save the default values execute the "Store parameters" function (see the 1010-01 Store parameters object). When the power is turned off, parameters not saved are lost.

1018 Identity Object

[Usigned8, ro]
This object contains some general information about the device. Sub-index 00 contains the number of the entries.
Default = 4

01 Vendor ID
[Unsigned32, ro]
It provides the manufacturer-specific vendor ID. The EtherCAT vendor ID is the same as the CANopen vendor ID.
Default = 0000 012Eh

02 Product Code
[Unsigned32, ro]
The manufacturer-specific product code identifies a specific device version.
Default = 0000 0022h IF56 converter for rotary encoders
0000 0023h IF56 converter for linear encoders

03 Revision Number
[Unsigned32, ro]
The manufacturer-specific revision number consists of a major revision number and a minor revision number. The major revision number identifies a specific device behaviour. The minor revision number identifies different version with the same device behaviour.

Default = 0001 0000h Lika EtherCAT IF56 series converter

7	...	0	15	...	8	23	...	16	31	...	24
Minor revision number						Major revision number					
LSB			...			...			MSB		

04 Serial Number
[Unsigned32, ro]
It provides the Serial Number of the device. It is 0 if no serial number is provided.
The Serial Number is shown in the following format: YYwwnnnnn.
YY = Year
ww = week
nnnnn = unique number in ascending order assigned by Lika Electronic
Default = device dependent

EXAMPLE
234100544 has to be intended as follows:
23 = Year of production = 2023
41 = Week of production = week 41
00544 = unique number in ascending order assigned by Lika Electronic

Default = device dependent

10F8-00 Timestamp Object

[Unsigned64, ro]
This optional attribute specifies the timestamp in units of nanoseconds (ns) of the local time at the send/receipt of a frame.
Default = 831224469310 (C188D6D73Eh)

1A00-00 TxPDO mapping parameter

[Usigned8, ro]
This object contains the mapping parameters for the PDOs the EtherCAT device is able to transmit. Sub-index 00 contains the number of entries.
Default = 2

01 Mapped Object 001

[Unsigned32, ro]
Sub-index 001 contains the information of the mapped application object 001. The object describes the content of the PDO by its index, sub-index, and length. The length contains the length of the application object expressed in bits. This may be used to verify the mapping.

7 0	15 8	23 16	31 24
Length	Sub-Index	Index	
LSB			MSB

Default = 6004 0020h = **6004-00 Position Value 32 bits** object, length 32 bits

02 Mapped Object 002

[Unsigned32, ro]
Sub-index 002 contains the information of the mapped application object 002. The object describes the content of the PDO by its index, sub-index, and length. The length contains the length of the application object expressed in bits. This may be used to verify the mapping.

7 0	15 8	23 16	31 24
Length	Sub-Index	Index	
LSB			MSB

Default = 3006 0020h = **3006-00 Velocity Value** object, length 32 bits

1C00-00 Sync Manager Communication Types

[Unsigned8, ro]
This object contains the number and type of Sync Manager Communication Types supported by the converter. Sub-index 00 specifies the number of Sync Manager channels. Refer also to the "Sync Manager" section on page 87.
Default = 4

01 SM MailBox Receive (SM0)
[Unsigned8, ro]
Used for mailbox write transfers (Master to Slave).
Default = 01h

02 SM MailBox Send (SM1)
[Unsigned8, ro]
Used for mailbox read transfers (Slave to Master).
Default = 02h

03 SM PDO output (SM2)
[Unsigned8, ro]
It contains the RxPDOs (i.e. Sync Manager 2 holds the Read Process Data).
Default = 03h

04 SM PDO input (SM3)
[Unsigned8, ro]
It contains the TxPDOs (i.e. Sync Manager 3 holds the Write Process Data).
Default = 04h

1C12-00 Sync Manager 1 RxPDO Assignment

[Unsigned8, ro]
This object specifies whether the device uses Receive PDO messages. Sub-index 00 specifies the number of entries, i.e. the number of assigned RxPDOs.
Default = 1

01 Subindex 001
This device uses Receive PDO messages to receive no data.
Default = 0000 1600h = 1600-00 RxPDO mapping parameter object is empty

1C13-00 Sync Manager 1 TxPDO Assignment

[Unsigned32, ro]
This object specifies whether the device uses Transmit PDO messages. Sub-index 00 specifies the number of entries, i.e. the number of assigned TxPDOs.
Default = 1

01 Subindex 001
This device uses TxPDO messages to send both the position and the speed values.

Default = 0000 1A00h = **1A00-00 TxPDO mapping parameter** object

1C33-00 SM Sync Manager input parameter

The **1C33-00 SM Sync Manager input parameter** object contains the input synchronization parameters. Some of them are calculated dynamically and depend on both the encoder/converter configuration (programmed resolution, counting direction, ...) and the synchronization mode (SM or DC). Sub-index 00 contains the number of entries.
Default = 32

01 Sync mode
[Unsigned16, rw]
It allows to select the synchronization mode. For more information refer to page 83.
0: FreeRun: see on page 83;
1: Synchronous with SM3 Event (no outputs available): see on page 84;
2: DC mode synchronous with SYNC0 event: see on page 85.
Default = 0001h (1)

02 Cycle time
[Unsigned32, ro]
This parameter depends on the **01 Sync mode** selected. Application cycle time, i.e. interval between two position samplings (internal timer). The value is expressed in nanoseconds (ns).
If 0 = "FreeRun": interval between two position samplings (internal timer).
If 1 = "Synchronous with SM3 Event": minimum interval between two SM3 events.
If 2 = "DC mode synchronous with SYNC0 event": SYNC0 cycle time.
Default = 001E A9E2h (2,009,570)

03 Shift Time
[Unsigned32, ro]
Interval between the SYNC0 synchronization event and the moment of inputs latching from hardware. This parameter is calculated dynamically and expressed in nanoseconds (ns). DC mode only.
Default = 0000 2FA8h (12,200)

04 Sync modes supported
[Unsigned16, ro]
It shows the list of the supported synchronization modes.
bit 0: FreeRun mode is supported
bit 1: Synchronous with SM3 Event mode is supported (no outputs available)
bit 2: DC mode synchronous with SYNC0 event is supported
Default = 0007h (7)

05 Minimum cycle time
[Unsigned32, ro]
Minimum duration of the converter internal cycle time. This parameter is calculated dynamically and depends on the operating parameters and the position value. It is expressed in nanoseconds (ns).
Default = 000F 4240h (1,000,000)

06 Calc and copy time
[Unsigned32, ro]
Time the internal micro-controller (DSP) needs to make calculations on latched optical reading of position and then copy updated data from local memory to ESC memory (Sync Manager) before they are available to EtherCAT. This parameter is calculated dynamically and depends on the operating parameters and the position value. It is expressed in nanoseconds (ns). DC mode only.
Default = 0000 EA60h (60,000)

0B SM event missed counter
[Unsigned16, ro]
Number of missed SM events in OPERATIONAL state (DC mode only).
Default = 0000h

0C Cycle exceeded counter
[Unsigned16, ro]
Number of occasions the cycle time was exceeded in OPERATIONAL state (cycle was not completed in time or the next cycle began too early).
Default = 0000h

20 Sync error
[Bool, ro]
The synchronization was not correct in the last cycle (outputs were outputted too late; DC mode only).
Default = FALSE

NOTE
Always save the new values after setting in order to store them in the non-volatile memory permanently. Use the **1010-01 Store parameters** object, see on page 94.
Should the power supply be turned off all data that has not been saved previously will be lost!

Manufacturer Specific Profile Area objects

2200-00 Rotary Encoder Settings

[Unsigned32, rw]

Byte 3	bits 31 ... 24	Not used
Byte 2	bits 23 ... 20	Not used
	bits 19 ... 16	**Sensor communication frequency R**
Byte 1	bits 15 ... 8	**Number of clocks R**
Byte 0	bit 7	**Bypass R**
	bits 6 and 5	Not used
	bit 4	**SSI output code R**
	bit 3	Not used
	bit 2	**SSI error bit R**
	bits 1 and 0	**Protocol R**

Default = 0003 1910h (0000 0000 0000 0011 0001 1001 0001 0000)

Protocol R

[Byte 0, bits 0 and 1]

It sets the type of communication protocol the connected encoder is equipped with. More in detail, it sets whether the SSI or BiSS protocol is installed and the type of SSI protocol is used by the SSI encoder to arrange the absolute position information. The SSI protocol can be the "LSB Right Aligned" protocol (**Protocol R** = 00) or the "MSB Left Aligned" protocol (**Protocol R** = 01). For any information on the SSI / BiSS protocol please refer to the "User's manual" of the connected encoder.

The available options are:

00 = SSI LSB Right Aligned protocol (see e.g. ...-BAx-..., ...-GAx-..., ... ordering codes)

01 = SSI MSB Left Aligned protocol (see e.g. ...-BGx-..., ...-GGx-..., ... ordering codes)

10 = BiSS C-mode protocol (see ...-SCx-... ordering code)

Default = 00 (min. = 00, max. = 10)

EXAMPLE

We need to connect the following rotary encoder: **AMT6-13-12-BA2-...** .

The AMT6-...-BA2-... rotary encoder uses the 25-bit "LSB Right Aligned" protocol and the Binary code to arrange the absolute position information. Thus you have to set the value 00 = "LSB Right Aligned" in this parameter. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE
We need to connect the following rotary encoder: **EHO36-24-00-GG4-...** .
"GG" in the ordering code means that the "MSB Left Aligned" protocol and Gray code are used to arrange the absolute position information. Thus you have to set the value 01 = "MSB Left Alignment" in this parameter. For further information refer to the "User's manual" of the connected encoder.

SSI error bit R
[Byte 0, bit 2] [SSI encoders only]
It enables / disables the management of the error bit transmitted by the connected SSI encoder. The error bit is always considered to be the bit 0 of the SSI word. The setting of the error bit also affects the setting of the number of clocks, see the **Number of clocks R** byte 1 below. The available options are:
0 = error bit not available
1 = error bit available
Default = 0 (min. = 0, max. = 1)

EXAMPLE
We need to connect the following modular encoder: **SMAR1-BG1-...** .
The SMAR1 modular encoder provides the error bit. Thus you can set either the value 0 = "error bit not available" or the value 1 = "error bit available" (suggested) in this bit according to needs. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE
We need to connect the following rotary encoder: **EHM36-12-13-BG4-...** .
The EHM36-...-BG4-... rotary encoder does not provide the error bit. Thus you have to set the value 0 = "error bit not available" in this bit. For further information refer to the "User's manual" of the connected encoder.

SSI output code R
[Byte 0, bit 4] [SSI encoders only]
It sets the output code used by the SSI encoder to output the absolute position information. The output code can be:
0 = Binary code
1 = Gray code
For any information on the output code you must refer to the "User's manual" of the connected encoder.
Default = 1 = Gray code (min. = 0, max. = 1)

EXAMPLE

We need to connect the following modular encoder: **SMAR1-BG1-...** .
The SMAR1 modular encoder uses the Binary code to output the absolute position information Thus you must set the value 0 = "Binary code" in this bit. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE

We need to connect the following rotary encoder: **EHM36-12-13-BG4-...** .
The EHM36-...-BG4-... rotary encoder uses the Binary code to output the absolute position information. Thus you have to set the value 0 = "Binary code" in this bit. For further information refer to the "User's manual" of the connected encoder.

Bypass R

[Byte 0, bit 7]

If the bit 7 **Bypass R** = 0 = "Bypass disabled", the Bypass mode is disabled, that is: the position value (refer to the **6004-00 Position Value 32 bits** object on page 128) read by the encoder can be processed according to needs, so the user can scale the value, set a preset, and change the counting direction.

If **Bypass R** = 1 = "Bypass enabled", the Bypass mode is enabled, that is: the information from the encoder is transmitted "as it is" and not processed in any way. The preset, scaling, and counting direction functions -even if set and enabled- are ignored. If, for example, the user sets a preset while the Bypass mode is enabled, the value is accepted, but not activated. As soon as the Bypass mode is disabled, the preset, scaling, and counting direction functions -if set and enabled- become active and the position value will be arranged accordingly.

Default = 0 = Bypass disabled (min. = 0, max. = 1)

Number of clocks R

[Byte 1]

It sets the number of clocks required by the connected encoder to send the complete data word. The number of clocks depends on the resolution / the max. number of information of the encoder and, e.g., on the type of SSI protocol. The value has to be comprised between 1 and 64. Furthermore, if the error bit management is enabled (SSI encoders only: **SSI error bit R** bit 1 = 1 = "error bit available"), you must consider this bit and add an additional clock. For any information on the clocks required please refer to the "User's manual" of the connected encoder.

Default = 19h (min. = 01h, max. = 40h)

WARNING

SSI encoders

If **Protocol R** bits 0 and 1 = 00 = "LSB Right Aligned" protocol:

- The **Number of clocks R** must always be "=13" when the overall resolution of the connected encoder is lower than or equal to 13 bits.

- The **Number of clocks R** must always be "=25" when the overall resolution of the connected encoder is between 14 and 25 bits.
- The **Number of clocks R** must always be "=32" when the overall resolution of the connected encoder is between 26 and 32 bits.

If **Protocol R** bits 0 and 1 = 01 = "MSB Left Aligned" protocol:

- The **Number of clocks R** must be equal to the sum of the bits of the single- and multiturn physical resolutions (see the **2201-00 Rotary Encoder Resolution** object).

BiSS encoders

If a BiSS encoder is connected (see the **Protocol R** bits 0 and 1 on page 100 = 10), the **Number of clocks R** must equal the number of bits required by the encoder for the SCD data. So you must always add 8 bits (1 error bit, 1 warning bit, 6 CRC bits) to the bits required by the overall resolution of the encoder.

EXAMPLE

We need to connect the following rotary encoder: **ES58-12-00-BA2-...** .
The ES58-...-BA2-... uses the 13-bit "LSB Right Aligned" protocol to arrange the absolute position information as its overall resolution is ≤ 13 bits (13 + 0 bits). It always requires 13 clocks (the length of the word is always 13 bits, regardless of the max. number of information to provide). Thus you have to set "13" in this parameter. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE

We need to connect the following rotary encoder: **EM58-10-14-GA2-...** .
The EM58-...-GA2-... uses the 25-bit "LSB Right Aligned" protocol to arrange the absolute position information as its overall resolution is ≤ 25 bits (12 + 13 bits). It always requires 25 clocks (the length of the word is always 25 bits, regardless of the max. number of information to provide). Thus you have to set "25" in this parameter. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE

We need to connect the following rotary encoder: **HM58-16-14-BG2-...** .
The HM58-...-BG2-... uses the "MSB Left Aligned" protocol to arrange the absolute position information. Its overall physical resolution is 30 bits (16 + 14 bits). It requires 30 clocks at least (the length of the word is 30 bits at least). Thus you have to set "30" in this parameter. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE

We need to connect the following rotary encoder: **ASC85-25-00-SC1-...** .
The ASC85-...-SC1-... rotary encoder is equipped with the BiSS C-mode interface. As the position value needs 25 bits and the SCD also includes 1 error bit, 1 warning bit, and 6 CRC bits, then you have to set 25 + 8 = "33" in this parameter. For further information refer to the "User's manual" of the connected encoder. When the BiSS interface is set (see the **Protocol R** bits 0

and 1), you must always add 8 bits (1 error bit, 1 warning bit, and 6 CRC bits) to the bits required by the overall resolution of the encoder.

Sensor communication frequency R
[Byte 2, bits 0 ... 3]
It sets the communication frequency (clock frequency) of the connected encoder. Typically the clock frequency of an SSI encoder is between 100 kHz and 1 / 2 MHz; the clock frequency of a BiSS encoder is between 200 kHz and 10 MHz. For detailed information please refer to the documentation of the connected encoder.
The available options are:
0 = 125 kHz
1 = 250 kHz
2 = 500 kHz
3 = 1 MHz
4 = 2 MHz
5 = 5 MHz
6 = 10 MHz
other = 500 kHz
Default = 3h (min. = 0h, max. = 0Fh)

2200-00 Linear Encoder Settings

[Unsigned32, rw]

Byte 3	bits 31 ... 28	Not used
	bits 27 ... 24	**Sensor communication frequency L**
Byte 2	bits 23 ... 16	**Number of clocks L**
Byte 1	bits 15 ... 8	**Max No of Information (bit) L**
Byte 0	bit 7	**Bypass L**
	bits 6 and 5	Not used
	bit 4	**SSI output code L**
	bit 3	Not used
	bit 2	**SSI error bit L**
	bits 1 and 0	**Protocol L**

Default = 0319 1310h (0000 0011 0001 1001 0001 0011 0001 0000)

Protocol L
[Byte 0, bits 0 and 1]
It sets the type of communication protocol the connected encoder is equipped with. More in detail, it sets whether the SSI or BiSS protocol is installed and the type of SSI protocol is used by the SSI encoder to arrange the absolute position information. The SSI protocol can be the "LSB Right Aligned" protocol (**Protocol L** = 00) or the "MSB Left Aligned" protocol (**Protocol L** = 01). For any information on the SSI / BiSS protocol please refer to the "User's manual" of the connected encoder.
The available options are:
00 = SSI LSB Right Aligned protocol (see e.g. ...-BAx-..., ...-GAx-..., ... ordering codes)
01 = SSI MSB Left Aligned protocol (see e.g. ...-BGx-..., ...-GGx-..., ... ordering codes)
10 = BiSS C-mode protocol (see ...-SCx-... ordering code)
Default = 00 (min. = 00, max. = 10)

EXAMPLE
We need to connect the following linear encoder: **SMA1-GA2-...** .
The SMA1 encoder uses the 25-bit "LSB Right Aligned" protocol to arrange the absolute position information. Thus you have to set the value 00 in this bit. For further information refer to the encoder's "User's manual".

EXAMPLE
We need to connect the following linear encoder: **SMAX-BG2-...** .
"BG" in the ordering code means that "MSB Left Aligned" protocol and Binary code are used to arrange the absolute position information. Thus you have to set the value 01 in this bit. For further information refer to the encoder's "User's manual".

SSI error bit L
[Byte 0, bit 2] [SSI encoders only]
It enables / disables the management of the error bit transmitted by the connected SSI encoder. The error bit is always considered to be the bit 0 of the SSI word. The setting of the error bit also affects the setting of the number of clocks, see the **Number of clocks L** byte 1 below. The available options are:
0 = error bit not available
1 = error bit available
Default = 0 (min. = 0, max. = 1)

EXAMPLE
We need to connect the following linear encoder: **SMA2-BG1-...** .
The SMA2-BG1-... linear encoder provides the error bit. Thus you can set either the value 0 = "error bit not available" or the value 1 = "error bit available"

(suggested) in this bit according to needs. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-...** .
The SMA5 linear encoder does not provide the error bit. Thus you have to set the value 0 = "error bit not available" in this bit. For further information refer to the "User's manual" of the connected encoder.

SSI output code L
[Byte 0, bit 4] [SSI encoders only]
It sets the output code used by the SSI encoder to output the absolute position information. The output code can be:
0 = Binary code
1 = Gray code
For any information on the output code you must refer to the "User's manual" of the connected encoder.
Default = 1 = Gray code (min. = 0, max. = 1)

EXAMPLE
We need to connect the following linear encoder: **SMA2-BG1-...** .
The SMA2-BG1-... linear encoder uses the Binary code to output the absolute position information. Thus you must set the value 0 = "Binary code" in this bit. For further information refer to the "User's manual" of the connected encoder.

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-...** .
The SMA5 linear encoder uses the Gray code to output the absolute position information. Thus you have to set the value 1 = "Gray code" in this bit. For further information refer to the "User's manual" of the connected encoder.

Bypass L
[Byte 0, bit 7]
If the bit 7 **Bypass L** = 0 = "Bypass disabled", the Bypass mode is disabled, that is: the position value (refer to the **6004-00 Position Value 32 bits** object on page 128) read by the encoder can be processed according to needs, so the user can scale the value, set a preset, and change the counting direction.
If **Bypass L** = 1 = "Bypass enabled", the Bypass mode is enabled, that is: the information from the encoder is transmitted "as it is" and not processed in any way. The preset, scaling, and counting direction functions -even if set and enabled- are ignored. If, for example, the user sets a preset while the Bypass mode is enabled, the value is accepted, but not activated. As soon as the Bypass mode is disabled, the preset, scaling, and counting direction functions -if set and enabled- become active and the position value will be arranged accordingly.

Default = 0 = Bypass disabled (min. = 0, max. = 1)

Max No of Information (bit) L
[Byte 1]
It sets the max. number of information (expressed in bits) the encoder can output for the max. measuring length, i.e. the total physical resolution. The value depends on the encoder resolution and the max. measuring length. As soon as you confirm the value, the system automatically sets the default value of the **6001-00 Total measuring range** and **6002-00 Total measuring range** objects accordingly. For any information on the max. number of information please refer to the "User's manual" of the connected encoder.
Default = 13h (min. value = 01h, max. value = 20h)

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-0050-...** . Its resolution is **0.05 mm** (see the ordering code).
The max. measuring length of the the SMA5 linear encoder on the MTA-A096 scale is **5,050 mm**.
The max. number of information the encoder can output results from the following calculation:

$$\text{Max. No of Information} = \frac{\text{Max. measuring range}}{\text{Resolution}}$$

$$\text{Max. No of Information} = \frac{5{,}050}{0.05} = \mathbf{101{,}000}$$

Now you have to "round up" the result to the next highest power of 2, that is: 131,072 = 2^{17}. Thus the number of bits is "17". The value to set in this byte is 11h.

EXAMPLE
We need to connect the following linear encoder: **SMAX-BG2-0100-...** . Its resolution is **0.1 mm** (see the ordering code).
The max. measuring length of the SMAX linear encoder on the MTAX-A301 scale is **600 mm**.
The max. number of information the encoder can output results from the following calculation:

$$\text{Max. No of Information} = \frac{\text{Max. measuring range}}{\text{Resolution}}$$

$$\text{Max. No of Information} = \frac{600}{0.1} = \mathbf{6{,}000}$$

Now you have to "round up" the result to the next highest power of 2, that is: 8,192 = 2^{13}. Thus the number of bits is "13". The value to set in this byte is 0Dh.

Number of clocks L
[Byte 2]
It sets the number of clocks required by the connected encoder to send the complete data word. The number of clocks depends on the resolution / the max. number of information of the encoder and, e.g., on the type of SSI protocol. The value has to be comprised between 1 and 32. Furthermore, if the error bit management is enabled (SSI encoders only: **SSI error bit L** bit 1 = 1 = "Error bit available"), you must consider this bit and add an additional clock. For any information on the clocks required please refer to the "User's manual" of the connected encoder.
Default = 19h (min. = 01h, max. = 20h)

WARNING
SSI encoders
If **Protocol L** bits 0 and 1 = 00 = "LSB Right Aligned" protocol:

- The **Number of clocks L** must always be "=13" when the overall resolution of the connected encoder is lower than or equal to 13 bits.
- The **Number of clocks L** must always be "=25" when the overall resolution of the connected encoder is between 14 and 25 bits.
- The **Number of clocks L** must always be "=32" when the overall resolution of the connected encoder is between 26 and 32 bits.

If **Protocol L** bits 0 and 1 = 01 = "MSB Left Aligned" protocol:

- The **Number of clocks L** must be equal to the number of bits required by the encoder to send the complete data word.

BiSS encoders
If a BiSS encoder is connected (see the **Protocol L** bits 0 and 1 on page 105 = 10), the **Number of clocks L** must equal the number of bits required by the encoder for the SCD data. So you must always add 8 bits (1 error bit, 1 warning bit, 6 CRC bits) to the bits required by the position value.

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-0050-...** .
The SMA5 linear encoder uses the "LSB Right Aligned" protocol and always requires 25 clocks (the length of the word is always 25 bits, regardless of the max. number of information to provide). Thus you have to set "25" in this byte. For further information refer to the "User's manual".

EXAMPLE
We need to connect the following linear encoder: **SMA2-BG1-0005-...** .
The SMA2-BG1-0005-... linear encoder uses the "MSB Left Aligned" protocol and the Binary code to output the absolute position information. It also provides the error bit. You set the value 2 = "Error bit available" in the **SSI error bit L** bit 2. As the position value needs 21 bits and the error bit needs 1 bit, then you must set "22" in this parameter.

EXAMPLE
We need to connect the following linear encoder: **SMAX-BG2-0100-...** .
The SMAX-BG2-0100-... linear encoder uses the "MSB Left Aligned" protocol and the Binary code to output the absolute position information. As the position value needs 13 bits, then you must set "13" in this parameter.

EXAMPLE
We need to connect the following linear encoder: **SMA21-SC1-0001-...** .
The SMA21-SC1-0001-... linear encoder is equipped with the BiSS C-mode interface. As the position value needs 25 bits and the SCD also includes 1 error bit, 1 warning bit, and 6 CRC bits, then you must set 25 + 8 = "33" in this parameter. When the BiSS interface is set (see the **Protocol L** bits 0 and 1), you must always add 8 bits (1 error bit, 1 warning bit, 6 CRC bits) to the bits required by the the overall resolution of the encoder.

Sensor communication frequency L
[Byte 3, bits 24 ... 27]
It sets the communication frequency (clock frequency) of the connected encoder. Typically the clock frequency of an SSI encoder is between 100 kHz and 1 / 2 MHz; the clock frequency of a BiSS encoder is between 200 kHz and 10 MHz. For detailed information please refer to the documentation of the connected encoder.
The available options are:
0 = 125 kHz
1 = 250 kHz
2 = 500 kHz
3 = 1 MHz
4 = 2 MHz
5 = 5 MHz
6 = 10 MHz
other = 500 kHz
Default = 3 (min. = 0, max. = 15)

2201-00 Rotary Encoder Resolution

[Unsigned32, rw]

Byte 3	Byte 2	Byte 1	Byte 0
Not used	Not used	**Multiturn resolution (bits)**	**Singleturn resolution (bits)**

Default = 0000 0C0Dh

Multiturn resolution (bits)

It sets the physical multiturn resolution (the number of physical revolutions) of the connected encoder expressed in bits.

The value has to be comprised between 01h and 10h. The physical resolution can be read in the ordering code (see the product datasheet). As soon as the user confirms the value, the system automatically sets the value in the **6502-00 Hardware number of revolutions** object accordingly. For any information on the multiturn resolution please refer to the "User's manual" of the connected encoder.

Default = 0Ch

Min. value =00h

Max. value = 18h **Singleturn resolution (bits)** + **Multiturn resolution (bits)** ≤ 30 (1Eh)

EXAMPLE

We need to connect the following rotary encoder: **ES58-12-00-...** .

In the ordering code the hardware multiturn resolution (number of revolutions) is expressed in bits. The value to be set in this byte is 00h. For further information refer also to the encoder's "User's manual".

EXAMPLE

We need to connect the following rotary encoder: **HM58-16-14-...** .

In the ordering code the hardware multiturn resolution (number of revolutions) is expressed in bits. The value to be set in this byte is 0Eh. For further information refer also to the encoder's "User's manual".

Singleturn resolution (bits)

It sets the physical singleturn resolution (the number of physical distinguishable steps per each revolution) of the connected encoder expressed in bits.

The value has to be comprised between 1h and 12h. The physical resolution can be read in the ordering code (see the product datasheet). As soon as the user confirms the value, the system automatically sets the value in the **6501-00 Hardware Singleturn Resolution** object accordingly. For any information on the singleturn resolution please refer to the "User's manual" of the connected encoder.

Default = 0Dh
Min. value =01h
Max. value = 16h **Singleturn resolution (bits)** + **Multiturn resolution (bits)** ≤ 30 (1Eh)
Max. value = 12h if **Multiturn resolution (bits)** = 00h

EXAMPLE
We need to connect the following rotary encoder: **EHM36-12-13-...** .
As you can easily infer from the ordering code, the hardware singleturn resolution is 12 bits. Thus the value to be set in this entry is 0Ch. For further information refer also to the encoder's "User's manual".

EXAMPLE
We need to connect the following rotary encoder: **HM58-16-14-...** .
As you can easily infer from the ordering code, the hardware singleturn resolution is 16 bits. Thus the value to be set in this entry is 10h. For further information refer also to the encoder's "User's manual".

2201-00 Linear Encoder Resolution

[Unsigned32, rw]
It sets the physical resolution of the linear encoder expressed in nanometres (nm). The physical resolution can be usually read in the ordering code and is expressed in millimetres (see the product datasheet). As soon as the user confirms the value, the system automatically sets the default value of the 6005-01 Position step setting and 6501-00 Measuring step objects accordingly.
Default = 0000 2710h
Min. value = 0000 0001h
Max. value = 3B9A CA00h (= 10 cm)

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-0050**-... .
As you can see in the product datasheet, "0050" in the ordering code means 0.05 mm resolution = 50,000 nanometre resolution. Thus you have to set the value 00 00 C3 50h in this object. For further information refer also to the encoder's "User's manual".

EXAMPLE
We need to connect the following linear encoder: **SMAX-BG2-0100**-... .
As you can see in the product datasheet, "0100" in the ordering code means 0.1 mm resolution = 100,000 nanometre resolution. Thus you have to set the value 00 01 86 A0h in this object. For further information refer also to the encoder's "User's manual".

3005-00 Velocity Format R

[Unsigned16, rw] [Rotary encoder]

It sets the engineering unit of the velocity value provided in the **3006-00 Velocity Value** object, according to the following table.

Value	Measuring unit
0	Speed is expressed in counts per second
1	Speed is expressed in rotations per minute (RPM)

Default = 0

3005-00 Velocity Format L

[Unsigned16, rw] [Linear encoder]

It sets the engineering unit of the velocity value provided in the **3006-00 Velocity Value** object, according to the following table.

Value	Measuring unit
0	Speed is expressed in counts per second
1	Speed is expressed in millimetres per second

Default = 0

3006-00 Velocity Value

[Unsigned32, ro]

It shows the current velocity value. It is calculated every 100 ms and expressed using the engineering unit set next to the **3005-00 Velocity Format R** / **3005-00 Velocity Format L** object (see above). The **3006-00 Velocity Value** object is mapped in the **1A00-00 TxPDO mapping parameter** object, sub-index 002 **02 Mapped Object 002**, see on page 96.

3007-00 Wrong parameters list R

[Unigned16, ro]

The operator has entered invalid data and the **Machine data not valid** warning in the **6505-00 Warnings** object has been triggered. This variable is meant to show (bit value = HIGH) the list of the wrong parameters, according to the following table.

Please note that the normal work status can be restored only after having set proper values.

Bit	Function	bit = 0	bit = 1
0	Units per revolution exceeded	Warning not active	Warning active
1	Total Measuring Range exceeded	Warning not active	Warning active
2	Preset value exceeded	Warning not active	Warning active
3	Offset value exceeded	Warning not active	Warning active
4	Rotary Encoder Settings exceeded	Warning not active	Warning active
5	Rotary Encoder Resolution exceeded	Warning not active	Warning active
6	Speed Format exceeded	Warning not active	Warning active
7 ... 15	not used		

Byte 0

Units per revolution exceeded

bit 0 Wrong data has been set next to the **6001-00 Units per revolution** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Total Measuring Range exceeded

bit 1 Wrong data has been set next to the **6002-00 Total measuring range 32 bits** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Preset value exceeded

bit 2 Wrong data has been set next to the **6003-00 Preset Value 32 bits** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Offset value exceeded

bit 3 Wrong data has been set next to the **6003-00 Preset Value 32 bits** object and the calculated **6509-00 Offset** is out-of-tolerance. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Rotary Encoder Settings exceeded

bit 4 — Wrong data has been set next to the **2200-00 Rotary Encoder Settings** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Rotary Encoder Resolution exceeded

bit 5 — Wrong data has been set next to the **2201-00 Rotary Encoder Resolution** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Speed Format exceeded

bit 6 — Wrong data has been set next to the **3005-00 Velocity Format R** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

bit 7 — Not used

Byte 1 — **Not used**

3007-00 Wrong parameters list L

[Unigned16, ro] [Linear encoder]

The operator has entered invalid data and the **Machine data not valid** warning in the **6505-00 Warnings** object has been triggered. This variable is meant to show (bit value = HIGH) the list of the wrong parameters, according to the following table.

Please note that the normal work status can be restored only after having set proper values.

Bit	Function	bit = 0	bit = 1
0	Total Measuring Range exceeded	Warning not active	Warning active
1	Position Step Setting exceeded	Warning not active	Warning active
2	Preset value exceeded	Warning not active	Warning active
3	Offset value exceeded	Warning not active	Warning active
4	Linear Encoder Settings exceeded	Warning not active	Warning active
5	Linear Encoder Resolution exceeded	Warning not active	Warning active

6	Speed Format exceeded	Warning not active	Warning active
7 ... 15	not used		

Byte 0

Total Measuring Range exceeded

bit 0 Wrong data has been set next to the **6001-00 Total measuring range** / **6002-00 Total measuring range** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Position Step Setting exceeded

bit 1 Wrong data has been set next to the **6005-01 Position step setting** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Preset value exceeded

bit 2 Wrong data has been set next to the **6003-00 Preset Value 32 bits** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Offset value exceeded

bit 3 Wrong data has been set next to the **6003-00 Preset Value 32 bits** object and the calculated **6509-00 Offset** is out-of-tolerance. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Linear Encoder Settings exceeded

bit 4 Wrong data has been set next to the **2200-00 Linear Encoder Settings** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Linear Encoder Resolution exceeded

bit 5 Wrong data has been set next to the **2201-00 Linear Encoder Resolution** object. The tolerances for the parameter have been exceeded. Set proper values to restore

the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

Speed Format exceeded

bit 6 — Wrong data has been set next to the **3005-00 Velocity Format L** object. The tolerances for the parameter have been exceeded. Set proper values to restore the normal work condition. The warning is cleared if the tolerances are within normal parameters again.

bit 7 — Not used

Byte 1 — **Not used**

NOTE

Always save the new values after setting in order to store them in the non-volatile memory permanently. Use the **1010-01 Store parameters** object, see on page 94.

Should the power supply be turned off all data that has not been saved previously will be lost!

Standardised Profile Area objects (DS406)

6000-00 Operating parameters

[Unsigned16, rw]

Bit	Function	bit = 0		bit = 1	
0	**Code sequence**	Count up information with **CW** clockwise **rotation**	Count up information with **standard direction movement**	Count up information with CCW counter-clockwise rotation	Count up information with reverse direction movement
1	not used				
2	**Scaling function**	**Disabled**		Enabled	
3 ... 14	not used				
15	**Mask Upgrade Firmware**	**New firmware is not loaded**		Firmware is loaded	

Default values are highlighted in bold
Default = 0000h

Code sequence
[Rotary encoder] This is intended to set whether the count is increasing (count up information) when the shaft of the encoder rotates clockwise (CW) or counter-clockwise (CCW). Setting 0 (bit 0 = 0) causes the encoder counting to increase when the encoder shaft rotates clockwise; setting 1 (bit 0 = 1) causes the encoder counting to increase when the encoder shaft rotates counter-clockwise. CW and CCW rotations are viewed from the shaft end.

[Linear encoder] This is intended to set whether the count is increasing (count up information) when the linear encoder moves in the standard direction (it is indicated in the encoder's manual) or when the encoder moves in reverse of the standard direction. Setting 0 (bit 0 = 0) causes the encoder counting to increase when the encoder moves in the standard direction; setting 1 (bit 0 = 1) causes the encoder counting to increase when the encoder moves in reverse of the standard direction. For any information on the standard and inverted counting direction please refer to the specific manual of the linear encoder.

To know whether the **Code sequence** is currently set to CW / Standard or CCW / Reversed, you can read the bit 0 **Code sequence** in the **6500-00 Operating Status** object, see on page 133.

WARNING
Every time you change the **Code sequence**, then you are required to activate a new preset (see the **6003-00 Preset Value 32 bits** object) and finally save the new parameters (see the **1010-01 Store parameters** object).

Scaling function

This is meant to disable (0) / enable (1) the scaled parameters **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** (rotary encoder): or **6001-00 Total measuring range** and **6002-00 Total measuring range** (linear encoder).

When the scaling function is disabled (bit 2 = 0), the encoder uses the physical resolution (see the **6501-00 Hardware Singleturn Resolution** and **6502-00 Hardware number of revolutions** objects, see also the **Multiturn resolution (bits)** and **Singleturn resolution (bits)** parameters in the **2201-00 Rotary Encoder Resolution** object -rotary encoder-; see the **2201-00 Linear Encoder Resolution** object and the **Max No of Information (bit) L** entry; the **6005-01 Position step setting** and **6501-00 Measuring step** objects are automatically set accordingly -linear encoder-) to calculate the absolute position information; the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects (rotary encoder) and **6005-01 Position step setting**, **6001-00 Total measuring range**, and **6002-00 Total measuring range** objects (linear encoder) are ignored.

On the contrary, when the scaling function is enabled (bit 2 = 1), the user is allowed to enter the custom singleturn resolution in the **6001-00 Units per revolution** object and the custom total resolution in the **6002-00 Total measuring range 32 bits** object (rotary encoder) / the custom resolution in the **6005-01 Position step setting** object and the custom number of information in the **6001-00 Total measuring range** / **6002-00 Total measuring range** objects (linear encoder) and these values are used to calculate the position information.

To know whether the **Scaling function** is currently enabled, you can read the bit 2 **Scaling function** of the **6500-00 Operating Status** object, see on page 133.

WARNING

Every time you enable the scaling function and/or change the scaling values (see the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects -rotary encoder-; the **6001-00 Total measuring range** / **6002-00 Total measuring range** and **6005-01 Position step setting** objects -linear encoder-), then you are required to activate a new preset (see the **6003-00 Preset Value 32 bits** object) and finally save the new parameters (see the **1010-01 Store parameters** object).

WARNING (linear encoder only)

When you enable the scaling function (**Scaling function** = 1), a counting error, i.e. a jump in the position count, may occur if the following conditions arise:

- a physical zero setting has been performed in the linear sensor;
- the **6005-01 Position step setting** object value is not a multiple of the physical resolution as set next to the **2201-00 Linear Encoder Resolution** object;

- the measuring range (**6001-00 Total measuring range** and **6002-00 Total measuring range** objects) is not a power of 2 submultiple of the maximum measuring range.

If the above described conditions arise, a counting error may occur when the sensor crosses the physical zero point.
If the scaling function is disabled (**Scaling function** = 0), the transmitted position values are always consistent.
If the scaling function is enabled (**Scaling function** = 1) yet no physical zero setting has been performed in the linear sensor, the transmitted position values are always consistent.
If the scaling function is enabled (**Scaling function** = 1), the **6005-01 Position step setting** object value is a multiple of the physical resolution and the measuring range (**6001-00 Total measuring range** and **6002-00 Total measuring range** objects) is a power of 2 submultiple of the maximum measuring range, the transmitted position values are consistent, regardless of the physical zero setting.

NOTE
Please consider that if the **Bypass** parameter (see the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object on page 102 -rotary encoder-; see the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** on page 106 -linear encoder-) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored.

Mask Upgrade Firmware
This bit must be set high (="1") to trigger the firmware update process through the File Access over EtherCAT protocol. For complete information on updating the firmware through the File Access over EtherCAT protocol please refer to the "7.3 File Over EtherCAT (FoE)" section on page 141.

6001-00 Units per revolution

[Unsigned32, rw] [Rotary encoder only]

WARNING
This object is active only if the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=1"; otherwise it is ignored and the system uses the physical resolution values (**6501-00 Hardware Singleturn Resolution** and **6502-00 Hardware number of revolutions**) to calculate the position information.
Furthermore, if the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object (see on page 102) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object sets a custom number of distinguishable steps per revolution (custom singleturn resolution).

To avoid counting errors, check that

$$\frac{\text{6501-00 Hardware Singleturn Resolution}}{\text{6001-00 Units per revolution}} = \text{integer value}$$

You are allowed to set whatever integer value less than or equal to the **maximum number of physical steps per revolution** (see the hardware counts per revolution in the encoder identification label and the **6501-00 Hardware Singleturn Resolution** object). If you enter an out-of-range value, the number of measuring units per revolution is forced to the physical singleturn resolution and the **6505-00 Warnings** object (see the bit 12 **Machine data not valid**) as well as the **3007-00 Wrong parameters list R** object (see the bit 0 **Units per revolution exceeded**) signal the error.

Default = 0000 2000h (8,192 cpr)

WARNING
When you set a new value next to the **6001-00 Units per revolution** object, please always check also the **6002-00 Total measuring range 32 bits** object value and be sure that the resulting number of revolutions complies with the **Hardware number of revolutions** of the device (see the **6502-00 Hardware number of revolutions** object).

$$\frac{\text{6002-00 Total measuring range 32 bits}}{\text{6001-00 Units per revolution}} \leq \textbf{Number of physical revolutions}$$

Let's suppose that the connected encoder has a number of physical revolutions = 65,536 and is programmed as follows:
6001-00 Units per revolution: 8,192 cpr
6002-00 Total measuring range 32 bits = 33 554 432$_{10}$ = 8,192 (cpr) * 4,096 (rev.)
Let's set a new singleturn resolution, for instance: **6001-00 Units per revolution** = 360 cpr.
If we do not change the **6002-00 Total measuring range 32 bits** value at the same time, we will get the following result:

$$\textbf{Number of revolutions} = \frac{33\,554\,432\ (\text{6002-00 Total measuring range 32 bits})}{360\ (\text{6001-00 Units per revolution})} = 93{,}206.755\ldots$$

As you can see, the encoder is required to carry out more than 93,000 revolutions, this cannot be because the hardware number of revolutions can be max. 65,536. When this happens, the encoder falls into an error signalling the faulty condition (see the **3007-00 Wrong parameters list R** and **6505-00 Warnings** objects, see also the diagnostic LEDs).

WARNING
When you enable the scaling function (bit 2 **Scaling function** = 1), please enter scaled values next to the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects that are consistent with the physical values. In the case of inconsistent values, the system will warn about the wrong parametrization and fault condition by means of the dedicated objects.

WARNING
Every time you change the scaled values (see the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects), then you are required to activate a new preset (see the **6003-00 Preset Value 32 bits** object).

6001-00 Total measuring range

[Unsigned32, rw] [Linear encoder only]

WARNING
This register is active only if the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=1"; otherwise the user cannot enter any value in this object. As soon as the user confirms the value in the **Max No of Information (bit) L** parameter, the program automatically sets the default value of the **6001-00 Total measuring range** and **6002-00 Total measuring range** objects accordingly.
Furthermore, if the **Bypass L** parameter in the **2200-00 Linear Encoder Settings** object (see on page 106) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

If the **Scaling function** is disabled (the bit 2 in the **6000-00 Operating parameters** object is set to "=0"), then **6001-00 Total measuring range** / **6002-00 Total measuring range** = $2^{\text{Max No of Information (bit) L}}$.

It sets the length of the travel the encoder has to measure. The value is expressed in number of information.
It can be either the number of information for the max. measuring length (for instance, if the application needs the whole path); or the number of information for just a part of the scale if the application only uses a section of the scale.

Thus this value must be less than or equal to the number of information resulting from the scale max. measuring length ($2^{\text{Max No of Information (bit) L}}$).
We suggest setting a value that is a power of 2 submultiple of the maximum measuring range (**Max No of Information (bit) L**) not to cause a counting error, i.e. a jump in the position count when the sensor crosses the physical zero point (see the WARNING below).
Default = $2^{\text{Max No of Information (bit) L}}$
Min. value = 0000 0002h
Max. value = $2^{\text{Max No of Information (bit) L}}$

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-0050**-... .
As you can see in the product datasheet, "0050" in the ordering code means a **0.05 mm** resolution. Let's say the mechanical travel of our application is the max. measuring length the SMA5 linear encoder is allowed to run on the MTA-A096 scale, i.e. **5,050 mm**. Thus the max. number of information is **101,000** ≈ **17 bits** (for the complete explanation refer to the **Max No of Information (bit) L** parameter on page 107). After setting the **Max No of Information (bit) L** parameter, the system automatically sets the value 0002 0000h = 131,072 = 2^{17} in this object. If you need a custom measuring range, you need to enable the **Scaling function** and then set a value less than 2^{17} = 131,072 here.
If you set a preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be $2^{\text{Max No of Information (bit) L}}$ - 1, i.e. 131,071.

...	131,069	131,070	131,071	0	1	2	...

EXAMPLE
We need to connect the following linear encoder: **SMAX-BG2-0100**-... .
As you can see in the product datasheet, "0100" in the ordering code means a **0.1 mm** resolution. Let's say the mechanical travel of our application is the max. measuring length the SMAX linear encoder is allowed to run on the MTAX-A301 scale, i.e. **600 mm**. Thus the max. number of information is **6,000** ≈ **13 bits** (for the complete explanation refer to the **Max No of Information (bit) L** parameter on page 107). After having set the **Max No of Information (bit) L** parameter, the system automatically sets the value 0000 2000h = 8,192 = 2^{13}. If you need a custom measuring range, you need to enable the **Scaling function** and then set a value less than 2^{13} = 8,192 here.
If you set a preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be $2^{\text{Max No of Information (bit) L}}$-1, i.e. 8,191.

...	8,189	8,190	8,191	0	1	2	...

EXAMPLE

We need to connect the **SMA5-GA2-0050-...**, its physical resolution is **0.05 mm**. Let's say the mechanical travel of our application is **1000 mm**. Thus the max. number of information is **20,000 ≈ 15 bits** (for the complete explanation refer to the **Max No of Information (bit) L** parameter on page 107). Thus you must enable the **Scaling function** parameter and set the value 0000 4E20h in this parameter (instead of the default value 0002 0000h).
In this way you will obtain several 20,000 information sections following each other all along the whole measuring length. The position information will be from 0 to 19,999; then again from 0 to 19,999 and so on.

...	19997	19998	19999	0	1	2	...	19997	19998	19999	0	1	2	...

← max measuring length →

WARNING

When you enable the scaling function (**Scaling function** = 1), a counting error, i.e. a jump in the position count, may occur if the following conditions arise:

- a physical zero setting has been performed in the linear sensor;
- the **6005-01 Position step setting** object value is not a multiple of the physical resolution as set next to the **2201-00 Linear Encoder Resolution** object;
- the measuring range (**6001-00 Total measuring range** and **6002-00 Total measuring range** objects) is not a power of 2 submultiple of the maximum measuring range.

If the above described conditions arise, a counting error may occur when the sensor crosses the physical zero point.
If the scaling function is disabled (**Scaling function** = 0), the transmitted position values are always consistent.
If the scaling function is enabled (**Scaling function** = 1) yet no physical zero setting has been performed in the linear sensor, the transmitted position values are always consistent.
If the scaling function is enabled (**Scaling function** = 1), the **6005-01 Position step setting** object value is a multiple of the physical resolution and the measuring range (**6001-00 Total measuring range** and **6002-00 Total measuring range** objects) is a power of 2 submultiple of the maximum measuring range, the transmitted position values are consistent, regardless of the physical zero setting.

WARNING

When you change the value next to **6001-00 Total measuring range** or **6002-00 Total measuring range** objects, then you must check the value in the **6003-00 Preset Value 32 bits** object and perform the preset operation.

NOTE

The 6001-00 Total measuring range and 6002-00 Total measuring range objects are exactly the same thing. When you change the value next to this 6001-00 Total measuring range object you also change the value next to the 6002-00 Total measuring range object; and vice versa.

6002-00 Total measuring range 32 bits

[Unsigned32, rw] [Rotary encoder only]

WARNING

This object is active only if the bit 2 **Scaling function** in the 6000-00 Operating parameters object is set to "=1"; otherwise it is ignored and the system uses the physical resolution values (6501-00 Hardware Singleturn Resolution and 6502-00 Hardware number of revolutions) to calculate the position information.
Furthermore, if the bit 7 **Bypass R** in the 2200-00 Rotary Encoder Settings object (see on page 102) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object sets a custom number of distinguishable steps over the total measuring range. The total resolution of the encoder results from the product of 6001-00 Units per revolution by the required **Number of revolutions**.
You are allowed to set whatever integer value less than or equal to the **overall hardware resolution** (see the encoder identification label as well as the 6501-00 Hardware Singleturn Resolution and 6502-00 Hardware number of revolutions objects). The overall hardware resolution results from:
6501-00 Hardware Singleturn Resolution * 6502-00 Hardware number of revolutions.
We recommend the **Number of revolutions** to be set to a power of 2.
The set **Number of revolutions** results from the following calculation:

$$\text{Number of revolutions} = \frac{\text{6002-00 Total measuring range 32 bits}}{\text{6001-00 Units per revolution}}$$

Setting the **Number of revolutions** to a value which is a power of 2 is meant to avoid problems when using the device in endless operations that require the physical zero to be overstepped. If you set the **Number of revolutions** which is not a power of 2, a counting error is generated before the physical zero.
If you enter an out-of-range value, the number of measuring units per revolution is forced to the physical singleturn resolution and the 6505-00 Warnings object (see the bit 12 Machine data not valid) as well as the 3007-00 Wrong parameters list R object (see the bit 1 Total Measuring Range exceeded) signal the error. See also the diagnostic LEDs.

Default = 0800 0000h (134,217,728)

WARNING

When you set a new value next to the **6002-00 Total measuring range 32 bits** object, please always check also the **6001-00 Units per revolution** object value and be sure that the resulting number of revolutions complies with the Hardware number of revolutions of the device.

Let's suppose that the encoder has a number of physical revolutions = 16,384 and is programmed as follows:

6001-00 Units per revolution: 8,192 cpr

6002-00 Total measuring range 32 bits = 134 217 728_{10} = 8,192 (cpr) * 16,384 (rev.)

Let's set a new total resolution, for instance: **6002-00 Total measuring range 32 bits** = 360.

As the **6002-00 Total measuring range 32 bits** must be greater than or equal to the **6001-00 Units per revolution**, the above setting is not allowed.

WARNING

Every time you change the value in this object then you are required to activate a new preset (see the **6003-00 Preset Value 32 bits** object) and finally save the new parameters (see the **1010-01 Store parameters** object).

EXAMPLE

We connect a multiturn rotary encoder having a singleturn resolution of 16 bits and a multiturn resolution of 14 bits.

The physical resolution is as follows:

- **Physical singleturn resolution**: **6501-00 Hardware Singleturn Resolution** = 65,536 (2^{16})
- **Physical multiturn resolution**: **6502-00 Hardware number of revolutions** = 16,384 turns (2^{14})
- **Total hardware resolution**: **6501-00 Hardware Singleturn Resolution * 6502-00 Hardware number of revolutions** = 1 073 741 824 ($2^{16} * 2^{14} = 2^{30}$)

In the specific installation **2,048 counts/rev. * 1,024 turns** are required:

- Enable the scaling function:**6000-00 Operating parameters**, bit 2 **Scaling function** = "1"
- Singleturn resolution: **6001-00 Units per revolution** = 2,048 (0000 0800h)
- Total resolution: **6002-00 Total measuring range 32 bits** = 2,048 * 1,024 = 2 097 152 (0020 0000h)

NOTE

We suggest values which are a power of 2 (2^n: 2, 4, ..., 2048, 4096, 8192,...) to be set in the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects to avoid counting errors.

WARNING

If **6001-00 Units per revolution** and/or **6002-00 Total measuring range 32 bits** values change, the **6003-00 Preset Value 32 bits** object must be updated according to the new resolution. A new preset operation is required.

6002-00 Total measuring range

[Unsigned32, rw] [Linear encoder only]

This object is exactly the same as the **6001-00 Total measuring range** object. When you change the value next to this **6002-00 Total measuring range** object you also change the value next to the **6001-00 Total measuring range** object; and vice versa. For complete information refer to the **6001-00 Total measuring range** object.

Default = $2^{\text{Max No of Information (bit) L}}$

Min. value = 0000 0002h

Max. value = $2^{\text{Max No of Information (bit) L}}$

6003-00 Preset Value 32 bits

[Unsigned32, rw]

This object allows to set the encoder position to a Preset value. The Preset function is meant to assign a desired value to a physical position of the encoder. The chosen physical position will get the value set next to this object and all the previous and following positions will get a value according to it. This function can be useful, for instance, when the zero position of the encoder and the zero position of the axis need to match. The preset value will be set for the position of the encoder in the moment when the preset value is sent. We suggest setting the preset value when the encoder is in stop.

Default = 0000 0000h

EXAMPLE

Let's take a look at the following example to better understand the preset function and the meaning and use of the related objects and commands: **6003-00 Preset Value 32 bits** and **6509-00 Offset**.

The encoder position which is transmitted results from the following calculation:

Transmitted value = **read position** (it does not matter whether the position is physical or scaled) + **6003-00 Preset Value 32 bits** - **6509-00 Offset**.

If you never set the **6003-00 Preset Value 32 bits** and you never performed the preset setting, then the transmitted value and the read position are

necessarily the same as **6003-00 Preset Value 32 bits** = 0 and **6509-00 Offset** = 0.
When you set the **6003-00 Preset Value 32 bits** and then execute the preset setting, the system saves the current encoder position in the **6509-00 Offset** object. It follows that the transmitted value and the **6003-00 Preset Value 32 bits** are the same as read position - **6509-00 Offset** = 0; in other words, the value set next to the **6003-00 Preset Value 32 bits** object is paired with the current position of the encoder as you wish.
For example, let's assume that the value "50" is set next to the **6003-00 Preset Value 32 bits** object and you execute the preset setting when the encoder position is "1000". In other words, you want to receive the value "50" when the encoder reaches the physical position "1000".

We will obtain the following information sequence:
Transmitted value = **read position** (="1000") + **6003-00 Preset Value 32 bits** (="50") - **6509-00 Offset** (="1000") = **50**.
The following transmitted value will be:
Transmitted value = **read position** (="1001") + **6003-00 Preset Value 32 bits** (="50") - **6509-00 Offset** (="1000") = **51**.
And so on.

To set the preset value you must send the following command:
Set the Preset value **6003-00 Preset Value 32 bits** (= 1000 = 3E8h)

Master → Converter

Cmd specific data							
Cmd	**Index**		**Sub**	**Data**			
23	03	60	00	E8	03	00	00

Converter → Master (Set confirmation)

Cmd specific data							
Cmd	**Index**		**Sub**	**Data**			
60	03	60	00	00	00	00	00

NOTE

- If the scaling function is disabled (see the bit 2 **Scaling function** in the **6000-00 Operating parameters** object = 0), then **6003-00 Preset Value 32 bits** must be less than or equal to the **total hardware resolution** (i.e. **6501-00 Hardware Singleturn Resolution** * **6502-00 Hardware number of revolutions**) – 1 (rotary encoder); less than or equal to $2^{\text{Max No of Information (bit) L}} - 1$ (for instance: **Max No of Information (bit) L** = 13 bits; $2^{13} - 1 = 8{,}191$ -linear encoder-).
- If the scaling function is enabled (see the bit 2 **Scaling function** in the **6000-00 Operating parameters** object = 1), then **6003-00 Preset Value 32 bits** must be less than or equal to **6002-00 Total measuring range 32 bits** - 1 (rotary encoder); less than or equal to **6001-00 Total measuring range** / **6002-00 Total measuring range** - 1 (linear encoder).

WARNING

Check the value in the **6003-00 Preset Value 32 bits** object and perform the preset operation every time you change the value next to the **Code sequence** parameter or the **6001-00 Units per revolution** and/or **6002-00 Total measuring range 32 bits** objects (rotary encoder) / **6001-00 Total measuring range** or **6002-00 Total measuring range** (linear encoder).

NOTE

Please consider that if the **Bypass** parameter (see the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object on page 102 -rotary encoder-; see the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** on page 106 -linear encoder-) is set to "1" = Bypass enabled, the preset function -even if enabled and activated- is ignored.

6004-00 Position Value 32 bits

[Unsigned32, ro]

This object contains the information about the current position of the encoder. The output value is scaled according to the scaling parameters, if the scaling function is enabled, see the bit 2 **Scaling function** of the **6000-00 Operating parameters** object and according to the setting in the **Bypass** parameter (see the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object on page 102 -rotary encoder-; see the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** on page 106 -linear encoder-). The **6004-00 Position Value 32 bits** object is mapped in the **1A00-00 TxPDO mapping parameter** object, sub-index 001 **01 Mapped Object 001**, see on page 96.

NOTE

Please consider that if the **Bypass** parameter (see the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object on page 102 -rotary encoder-; see the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** on page 106 -linear encoder-) is set to "0" = disabled, the position value read by the encoder can be processed according to needs, so the user can scale the value, set a preset, and change the counting direction. On the contrary, if the **Bypass** parameter is set to "1" = enabled, the information from the encoder is transmitted "as it is" and not processed in any way. The preset, scaling, and counting direction functions -even if set and enabled- as well as the output code are ignored. If, for example, the user sets a preset while the "Bypass mode" is enabled, the value is accepted, but not activated. As soon as the "Bypass mode" is disabled, the preset, scaling, and counting direction functions -if set and enabled- become active and the **6004-00 Position Value 32 bits** will be accordingly.

NOTE (linear encoders)
To convert the read position value into nanometres [nm] (and into micrometres or millimetres or any other engineering unit afterwards) you must multiply the read position by the value set next to the 6501-00 Measuring step object (if the bit 2 **Scaling function** in the 6000-00 Operating parameters object is disabled = 0); otherwise you must multiply the read position by the value set next to the 6005-01 Position step setting object (if the bit 2 **Scaling function** in the 6000-00 Operating parameters object is enabled = 1).

EXAMPLE
We have the following linear encoder: **SMA5-GA2-0050-...** .
Scaling function = 0
6501-00 Measuring step = 0000 C350h = 50,000 nm = 0.05 mm
6004-00 Position Value 32 bits = 0001 1005h = 69,637 dec
Position = 6004-00 Position Value 32 bits * 6501-00 Measuring step = 0001 1005h * 0000 C350h = CF88 D090h = 3,481,850,000 nm
3,481,850,000 nm = 3,481,850 µm = 3,481.85 mm

EXAMPLE
We have the following linear encoder: **SMA5-GA2-0100-...** .
Scaling function = 1
6005-01 Position step setting = 0001 86A0h = 100,000 nm = 0.1 mm
6004-00 Position Value 32 bits = 0000 1760h = 5,984 dec
Position = 6004-00 Position Value 32 bits * 6005-01 Position step setting = 0000 1760h * 0001 86A0h = 23AA DC00h = 598,400,000 nm
598,400,000 nm = 598,400 µm = 598.4 mm

6005-00 Measuring step setting

[Unsigned8, ro] [Linear encoder only)
This object is intended to show the number of subindexes that are included in the object 6005.
Default = 1h

6005-01 Position step setting

[Unsigned32, rw] [Linear encoder only]

WARNING
This object is active only if the bit 2 **Scaling function** in the 6000-00 Operating parameters object is set to "=1"; otherwise it is ignored, the user cannot enter any value in this object and the system uses the physical resolution, see the 2201-00 Linear Encoder Resolution and 6501-00 Measuring step objects. As soon as the user confirms the value in the 2201-00 Linear Encoder Resolution object, the program automatically sets the

default value of the **6005-01 Position step setting** and **6501-00 Measuring step** objects accordingly.
Furthermore, if the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** object (see on page 106) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

If the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=0", then **6005-01 Position step setting** = **2201-00 Linear Encoder Resolution**.

This object is used to set a custom resolution (otherwise referred to as measuring step) expressed in nanometres [nm].
The resolution can be defined as the smallest change in the underlying quantity that produces a response in the measurement, the response being the information that is provided to output.
The custom resolution value must be greater than or equal to the physical resolution of the connected encoder.
We suggest setting a value that is a multiple of the physical resolution as set next to the **2201-00 Linear Encoder Resolution** object not to cause a counting error, i.e. a jump in the position count when the sensor crosses the physical zero point (see the WARNING below).
Default = according to **6501-00 Measuring step**
Min. value = according to **6501-00 Measuring step**
Max. value = 3B9A CA00h (=10 cm)

EXAMPLE
We need to connect the following linear encoder: **SMA5-GA2-0050-...** .
As you can see in the product datasheet, "0050" in the ordering code means a **0.05 mm** resolution = 50,000 nanometres resolution. As soon as the user confirms the value in the **2201-00 Linear Encoder Resolution** object, the system automatically sets the default value of the **6005-01 Position step setting** object accordingly (0000 C350h). If needed, after enabling the **Scaling function** parameter, the user is allowed to set a custom resolution: it must be greater than or equal to 0000 C350h.

EXAMPLE
We need to connect the following linear encoder: **SMAX-BG2-0100-...** .
As you can see in the product datasheet, "0100" in the ordering code means a **0.1 mm** resolution = 100,000 nanometres resolution. As soon as the user confirms the value in the **2201-00 Rotary Encoder Resolution** object, the system automatically sets the default value of the **6005-01 Position step setting** object accordingly (0001 86A0h). If needed, after enabling the **Scaling function** parameter, the user is allowed to set a custom resolution: it must be greater than or equal to 0001 86A0h.

WARNING

When you enable the scaling function (**Scaling function** = 1), a counting error, i.e. a jump in the position count, may occur if the following conditions arise:

- a physical zero setting has been performed in the linear sensor;
- the **6005-01 Position step setting** object value is not a multiple of the physical resolution as set next to the **2201-00 Rotary Encoder Resolution** object;
- the measuring range (**6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects) is not a power of 2 submultiple of the maximum measuring range.

If the above described conditions arise, a counting error may occur when the sensor crosses the physical zero point.
If the scaling function is disabled (**Scaling function** = 0), the transmitted position values are always consistent.
If the scaling function is enabled (**Scaling function** = 1) yet no physical zero setting has been performed in the linear sensor, the transmitted position values are always consistent.
If the scaling function is enabled (**Scaling function** = 1), the **6005-01 Position step setting** object value is a multiple of the physical resolution and the measuring range is a power of 2 submultiple of the maximum measuring range, the transmitted position values are consistent, regardless of the physical zero setting.

NOTE

If you have set and activated the preset, when you change the value next to the **6005-01 Position step setting** object, then you must check the value in the **6003-00 Preset Value 32 bits** object and perform the homing operation.

EXAMPLE

The main and default features of the **SMAX-BG2-0100-...** linear encoder are as follows:

1 - **Default resolution** = 0.1 mm = 100,000 nm
2 - **MTAX-A301 max. measuring length** = 600 mm
3 - **Max. number of information** = 6,000 (13 bits)

As stated, the max. number of information provided to output is calculated as follows:

$$\text{Number of information} = \frac{\text{Max. measuring length}}{\text{Resolution}}$$

Thus, in a default configuration the number of information is:

$$\textbf{Number of information} = \frac{\textbf{Max. measuring length}}{\textbf{Resolution}} = \frac{600}{0.1} = \textbf{6,000}$$

Let's assume that you need **2,000 information** to be provided to output for the max. measuring length. It follows that you need to calculate and then set a custom resolution.

The resolution value results from the following calculation:

$$\textbf{Resolution} = \frac{\textbf{Max. measuring length}}{\textbf{Number of information}}$$

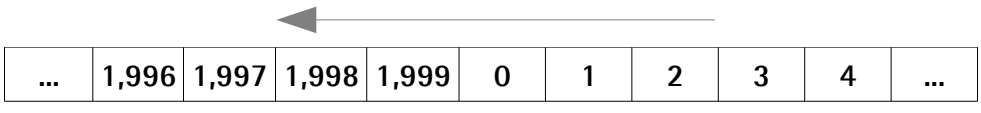
Thus, in the example the resolution will be:

$$\textbf{Resolution} = \frac{\textbf{Max. measuring length}}{\textbf{Number of information}} = \frac{600}{2{,}000} = \textbf{0.3}$$

As the value next to the 6005-01 Position step setting object has to be expressed in nanometres, then you have to enter the value **300,000**.
The complete programming sequence will be:

1. Enable the **Scaling function**: 6000-00 Operating parameters, bit 2 = 1
2. Set the custom resolution: 6005-01 Position step setting = 0004 93E0 hex (300,000 dec)
3. Save the set parameters (1010-01 Store parameters object; see on page 94)

NOTE
Please note that, if you set a preset along the path, when the encoder moves back and cross the zero, the value immediately after 0 will be 1,999 as shown below.



6500-00 Operating Status

[Unsigned16, ro]

Bit	Function	bit = 0		bit = 1	
0	**Code sequence**	Count up information with CW clockwise rotation	Count up information with standard direction movement	Count up information with CCW counter-clockwise rotation	Count up information with reverse direction movement
1	not used				
2	**Scaling function**	Disabled		Enabled	
3 ... 15	not used				

Code sequence

[Rotary encoder] It shows the value that is currently set in the bit 0 **Code sequence** of the 6000-00 Operating parameters object. If the bit is "=0" the output encoder position value has been set to increase when the encoder shaft rotates clockwise; if the bit is "=1" instead the output encoder position value has been set to increase when the encoder shaft rotates counter-clockwise.

[Linear encoder] It shows the value that is currently set in the bit 0 **Code sequence** of the 6000-00 Operating parameters object. If the bit is "=0" the output encoder position value has been set to increase when the linear encoder moves in the standard direction (it is indicated in the encoder's manual); if the bit is "=1" instead the output encoder position value has been set to increase when the encoder moves in reverse of the standard direction.

To set the code sequence to either CW / standard direction or CCW / reverse direction you must set the bit 0 **Code sequence** in the 6000-00 Operating parameters object to 0 / 1. For complete information on setting and using the counting direction function refer to the 6000-00 Operating parameters object on page 117.

Scaling function

It shows the value that is currently set in the bit 2 **Scaling function** of the 6000-00 Operating parameters object. In other words, it is intended to show whether the scaling function is enabled or disabled. If the bit is "=0", the scaling function is disabled; if the value is "=1" instead the scaling function is enabled.

To disable / enable the scaling function you must set the bit 2 **Scaling function** in the 6000-00 Operating parameters object to 0 / 1. For complete information on setting and using the scaling function refer to the 6000-00 Operating parameters object on page 117.

6501-00 Hardware Singleturn Resolution

[Unsigned32, ro] [Rotary encoder only]

WARNING

This object is active only if the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=0"; otherwise it is ignored and the system uses the custom values (**6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits**) to calculate the position information.
Furthermore, if the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object (see on page 102) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object is intended to show the number of physical distinguishable steps provided per each turn by the hardware of the connected encoder (physical singleturn resolution, see the hardware counts per revolution in the encoder identification label). The physical singleturn resolution of the encoder must be set next to the **Singleturn resolution (bits)** parameter of the **2201-00 Rotary Encoder Resolution** object, see on page 110. As soon as the user confirms the value in the value in the **Singleturn resolution (bits)** parameter of the **2201-00 Rotary Encoder Resolution** object, the program automatically sets the value in this object accordingly.
If you want to set a custom singleturn resolution, see the **6001-00 Units per revolution** object on page 119.

Default = according to the **Singleturn resolution (bits)** parameter of the **2201-00 Rotary Encoder Resolution** object

6501-00 Measuring step

[Unsigned32, ro] [Linear encoder only]

WARNING

This object is active only if the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=0"; otherwise it is ignored and the system uses the custom values (**6001-00 Total measuring range** / **6002-00 Total measuring range**) to calculate the position information.
Furthermore, if the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** object (see on page 106) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object is intended to show the physical resolution of the connected encoder expressed in nanometres [nm]. The physical resolution must be set next to the **2201-00 Linear Encoder Resolution** object on page 111. As soon as the user confirms the value in the **2201-00 Linear Encoder Resolution** object, the program automatically sets the value in this object accordingly.
If you want to set a custom resolution, see the **6005-01 Position step setting** object.

Default = according to the setting in the **2201-00 Linear Encoder Resolution** object

6502-00 Hardware number of revolutions

[Unsigned32, ro] [Rotary encoder only]

WARNING

This object is active only if the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=0"; otherwise it is ignored and the system uses the custom values (**6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits**) to calculate the position information.
Furthermore, if the bit 7 **Bypass R** in the **2200-00 Rotary Encoder Settings** object (see on page 102) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This object is intended to show the number of physical turns provided by the hardware of the connected encoder (physical multiturn resolution, see the hardware revolutions in the encoder identification label). The physical multiturn resolution of the encoder must be set next to the **Multiturn resolution (bits)** parameter of the **2201-00 Rotary Encoder Resolution** object, see on page 110. As soon as the user confirms the value in the **Multiturn resolution (bits)** parameter of the **2201-00 Rotary Encoder Resolution** object, the program automatically sets the value in this object accordingly.
If you want to set a custom number of turns, see the **6001-00 Units per revolution** and **6002-00 Total measuring range 32 bits** objects on page 119 ff.

Default = according to the **Multiturn resolution (bits)** parameter of the **2201-00 Rotary Encoder Resolution** object

6502-00 Hardware number of revolutions

[Unsigned32, ro] [Linear encoder only]

WARNING

This object is active only if the bit 2 **Scaling function** in the **6000-00 Operating parameters** object is set to "=0"; otherwise it is ignored and the system uses the custom values (**6001-00 Total measuring range** / **6002-00 Total measuring range**) to calculate the position information.
Furthermore, if the bit 7 **Bypass L** in the **2200-00 Linear Encoder Settings** object (see on page 106) is set to "1" = Bypass enabled, the scaling function -even if enabled- is ignored and the position information is outputted as it is.

This register is not available in the converter for linear encoders, it is listed for full compliance with the EtherCAT specifications.
Default = 0000 0001h

6503-00 Errors

[Unsigned16, ro]

The corresponding bits of supported errors are set (see the following **6504-00 Supported errors** object).

An error indicates that a malfunction has occurred which could lead to an incorrect position value. It is set when a bit indicating a fault is set to true (high). The alarm remains active until the error is cleared and the device is able to provide an accurate position value. The object is defined according to the following table.

Refer also to the following **6504-00 Supported errors** object.

Bit	Function	bit = 0	bit = 1
0	Position error	Alarm not active	Alarm active
1 ... 11	not used		
12	Flash memory error	Alarm not active	Alarm active
13 ... 15	not used		

Position error

Error bit of the position reading system. Fault and malfunction of the encoder/converter position measurement system or the measured value processing unit. This error causes an invalid position and speed actual value, it may be due to the hardware or the signal quality.

Flash memory error

Internal error, it cannot be restored. The flash memory contains corrupted data; or maybe the flash memory is damaged.

6504-00 Supported errors

[Unsigned16, ro]

This object contains the information on the error alarms supported by the converter. Refer to the previous **6503-00 Errors** object.

Bit	Function	bit = 0	bit = 1
0	Position error	Not supported	Supported
1 ... 11	not used		
12	Flash memory error	Not supported	Supported
13 ... 15	not used		

Default = 1001h (= 0001 0000 0000 0001 = alarms at bits 0 and 12 are supported and available in the previous **6503-00 Errors** object).

6505-00 Warnings

[Unsigned16, ro]

The corresponding bits of supported warnings are set (see the following **6506-00 Supported warnings** object).

The **6505-00 Warnings** object indicates that tolerances for certain internal parameters of the converter have been exceeded. It does not imply incorrect position values. The warning is cleared if the tolerances are again within normal parameters. The object is defined according to the following table.

Refer also to the following **6506-00 Supported warnings** object.

Bit	Function	bit = 0	bit = 1
0 ... 11	not used		
12	**Machine data not valid**	Warning not active	Warning active
13	not used		
14	**Position Warning**	Warning not active	Warning active
15	not used		

Machine data not valid

An out-of-tolerance parameter has been set. For more details about the specific out-of-tolerance parameter refer to the **3007-00 Wrong parameters list R** (rotary encoder) / **3007-00 Wrong parameters list L** (linear encoder) object, see on page 112.

Position Warning

Warning bit of the position reading system. Fault and malfunction of the encoder/converter position measurement system or the measured value processing unit. This warning does not cause a invalid position and speed actual values, it may be due to the hardware or the signal quality.

6506-00 Supported warnings

[Unsigned16, ro]

This object contains information on the supported warnings. Refer to the previous **6505-00 Warnings** object.

Bit	Function	bit = 0	bit = 1
0 ... 11	not used		
12	**Machine data not valid**	Not supported	Supported
13	not used		
14	**Position Warning**	Not supported	Supported
15	not used		

Default = 5000h (= 0101 0000 0000 0000 = warnings at bits 12 and 14 are supported and available in the previous **6505-00 Warnings** object).

6509-00 Offset

[Unsigned32, ro]

This object contains the Offset value. As soon as you activate the preset, the current position of the encoder is saved in this object. The offset value is then used in the preset function in order to calculate the encoder position value to be transmitted. To zero set the value in this object you must upload the factory default values (see the **1011-01 Restore default parameters** object on page 94).

For any further information on the preset function and the meaning and use of the related objects **6003-00 Preset Value 32 bits** and **6509-00 Offset**, refer to page 126.

NOTE

To save the new parameters execute the store parameters function (see the **1010-01 Store parameters** object on page 94).

When the power is turned off, parameters not saved are lost.

7.2.6 SDO Abort codes

SDO transfer could be unsuccessful; causes of error are listed and described in the SDO Abort Codes. Here follows the list of the available SDO Abort Codes. For complete information see ETG1000.6 "EtherCAT Specification – Part 6. Application Layer protocol specification", par. 5.6.2.7.2 table 40.

Abort code	Description
0503 0000h	Toggle bit not changed.
0504 0000h	SDO protocol timeout.
0504 0001h	Client/Server command specifier not valid or unknown.
0504 0005h	Out of memory.
0601 0000h	Unsupported access to an object.
0601 0001h	Attempt to read a write only object.
0601 0002h	Attempt to write a read only object.
0602 0000h	The object does not exist in the object dictionary.
0604 0041h	The object cannot be mapped into the PDO.
0604 0042h	The number and length of the objects to be mapped would exceed PDO length.
0604 0043h	General parameter incompatibility reason.
0604 0047h	General internal incompatibility in the device.
0606 0000h	Access failed due to a hardware error.
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Subindex does not exist.
0609 0030h	Value range of parameter exceeded (only for write access).
0609 0031h	Value of parameter written too high.
0609 0032h	Value of parameter written too low.
0609 0036h	Maximum value is less than minimum value.
0800 0000h	General error
0800 0020h	Data cannot be transferred or stored to the application.
0800 0021h	Data cannot be transferred or stored to the application because of local control.
0800 0022h	Data cannot be transferred or stored to the application because of the present device state.
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present.

Refer also to the "4.8 Diagnostic LEDs (Figure 4)" section on page 33.

7.2.7 Emergency Error Codes

Emergency Service is used by the Server for transmitting diagnostic messages to the client using MailBox; Error Codes are listed and described in the ETG1000.6 "EtherCAT Specification – Part 6. Application Layer protocol specification", par. 5.6.4.2 table 50.

Error Code		Error Register	Diagnostic Data				
Byte (0)	Byte (1)	Byte (2)	Byte (3)	Byte (4)	Byte (5)	Byte (6)	Byte (7)

Error Code — ESM State Transition Errors of State Machine:
(for detailed description see ETG1000.6 par. 5.6.4.3)
A000hex: transition from **PRE-OPERATIONAL** to **SAFE-OPERATIONAL** not successful
A001hex: transition from **SAFE-OPERATIONAL** to **OPERATIONAL** not successful
Encoder/converter errors:
5000hex: **Hardware error**
5001hex: **Diagnostic data** (wrong parameters loaded from flash memory)

Error Register — EtherCAT state machine current status (ESM)

Diagnostic Data — information about possible error causes (see ETG1000.6 par. 5.6.4.3.2-5).

Refer also to the "4.8 Diagnostic LEDs (Figure 4)" section on page 33.

7.2.8 AL Status Error Codes

If the state transition requested by the Master through the "AL Control Register" is unsuccessful, the Slave sets to 1 the "Error Indicator Bit" in "AL Status Register" and writes the cause of the error in "AL Status Code Register".
Values and descriptions of "AL Status Code" are available in ETG1000.6 "EtherCAT Specification – Part 6. Application Layer protocol specification", par.5.3.2 Table 11.

7.3 File Over EtherCAT (FoE)

Lika converters are devices that allow the firmware update using the "File Access over EtherCAT (FoE)" protocol.

7.3.1 Downloading files

To download files using Beckhoff TwinCAT 3 proceed as follows:

1. Connect to the Slave.
2. Highlight the Slave and navigate to the **Online** tabbed page.
3. Make sure the Slave is in Pre-Operational state. To check the current state of the converter see the **Current State** information field in the **State machine** group box. If required, press the **PRE-OP** button in the **State machine** group box.

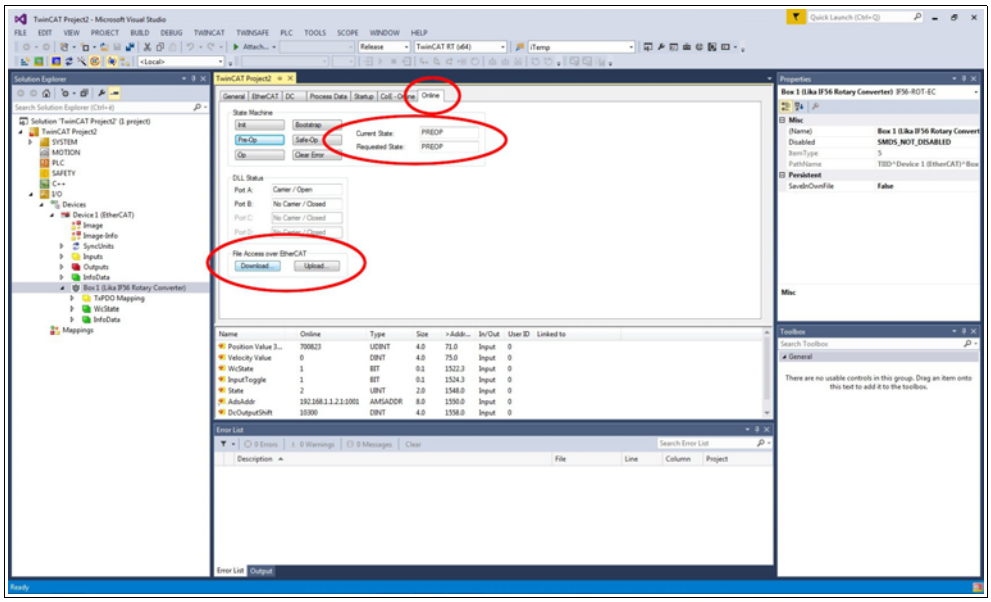


Figure 51 - Setting the State Machine

4. In the same page refer to the **File Access over EtherCAT** group box. If the **DOWNLOAD / UPLOAD** buttons are greyed out (disabled), make sure the **File Access over EtherCAT (FoE)** checkbox is selected.
5. To do this, enter the **EtherCAT** tabbed page first and then press the **ADVANCED SETTINGS...** button.

6. In the **Advanced Settings** page open the **Mailbox** list and then press the **FoE** command: the **FoE** group box will be displayed. Make sure the **File Access over EtherCAT (FoE)** checkbox is selected.

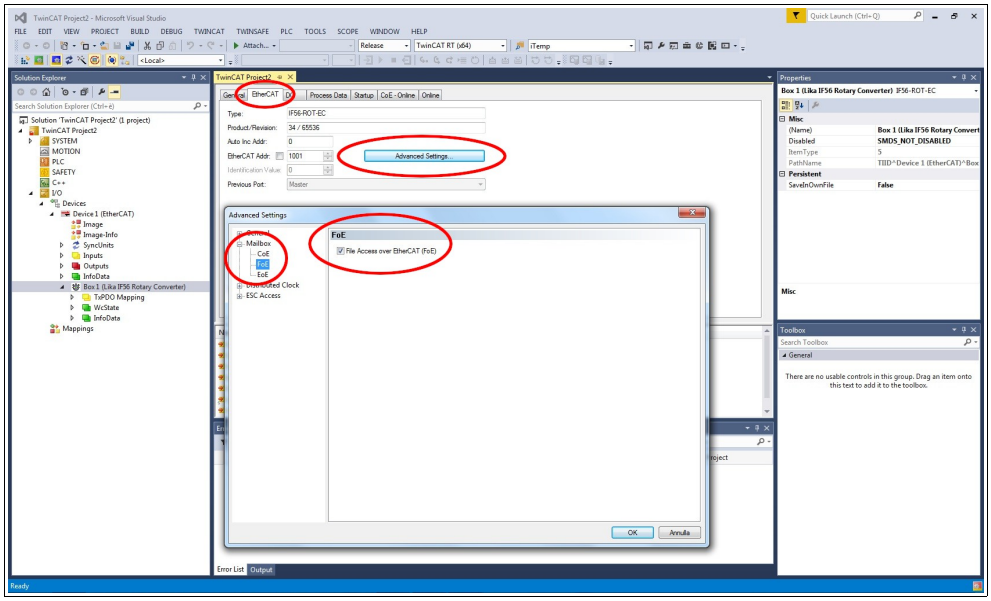


Figure 52 - Enabling the File Access over EtherCAT (FoE)

7. To download the firmware file press the **DOWNLOAD...** button in the **File Access over EtherCAT** group box in the **Online** tabbed page.

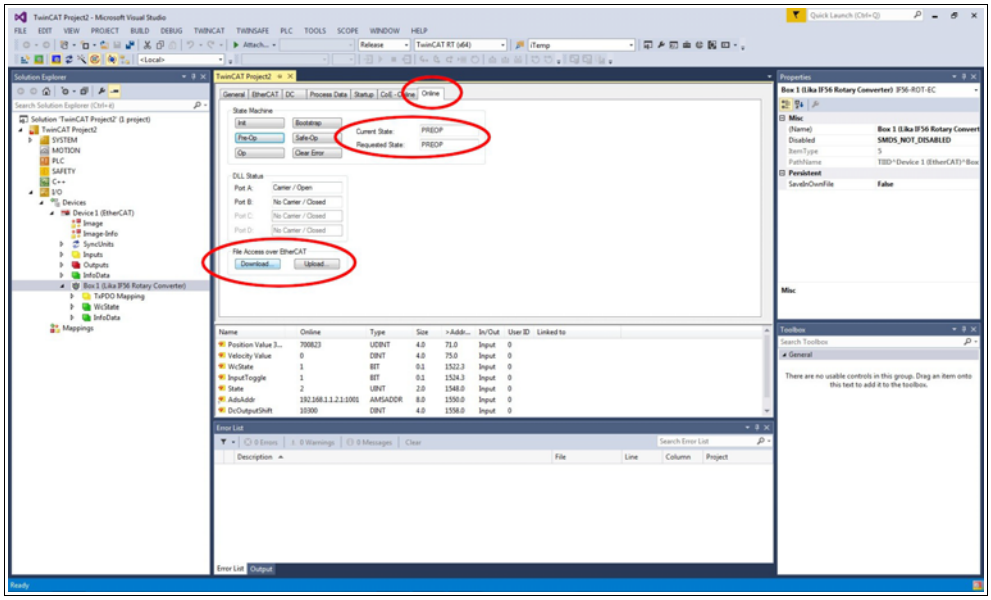


Figure 53 - Downloading/Uploading a file via File Access over EtherCAT (FoE)

8. Browse for the firmware file (select "All Files (*.*)" as extension) and then click **OPEN** and **OK** to download the file. The name of the file to be

downloaded must be compulsorily FWUPDATE.ZIP. ZIP files with different name are not accepted.

WARNING

After selecting the FWUPDATE.ZIP file in the **Open** page and confirming by pressing the **OPEN** button, the **Edit FoE Name** dialog box will appear on the screen. In the **String** field the file extension will be omitted. Please add the .ZIP extension to the file name. Enter the password 00000000hex next to the **Password (hex)** item below in the page and then press the **OK** button to confirm.

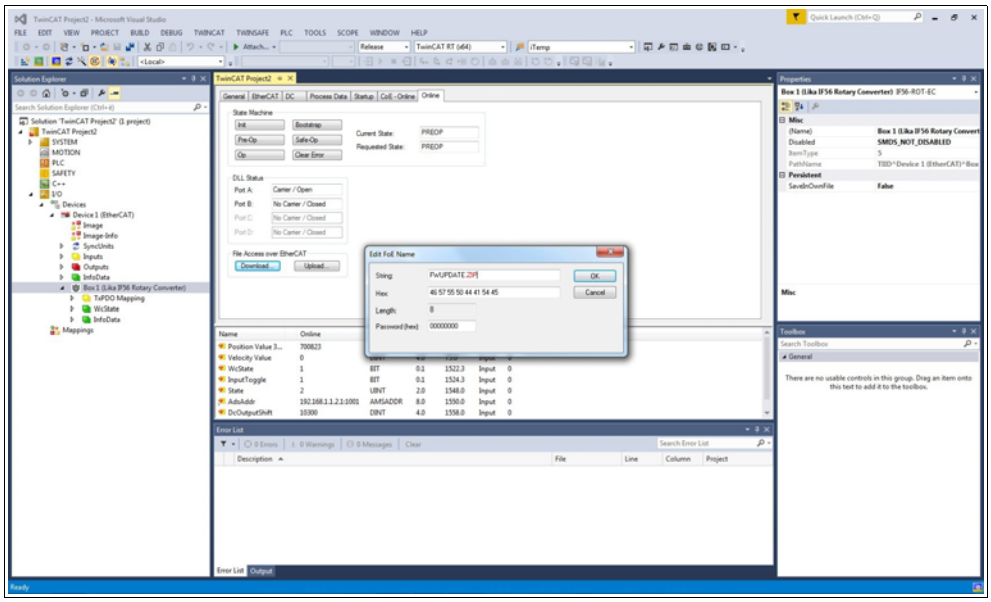


Figure 54 - Edit FoE Name dialog box

9. To trigger the new upgrade go to **CoE - Online** tabbed page and re-scan the objects by pressing the **UPDATE LIST** button.

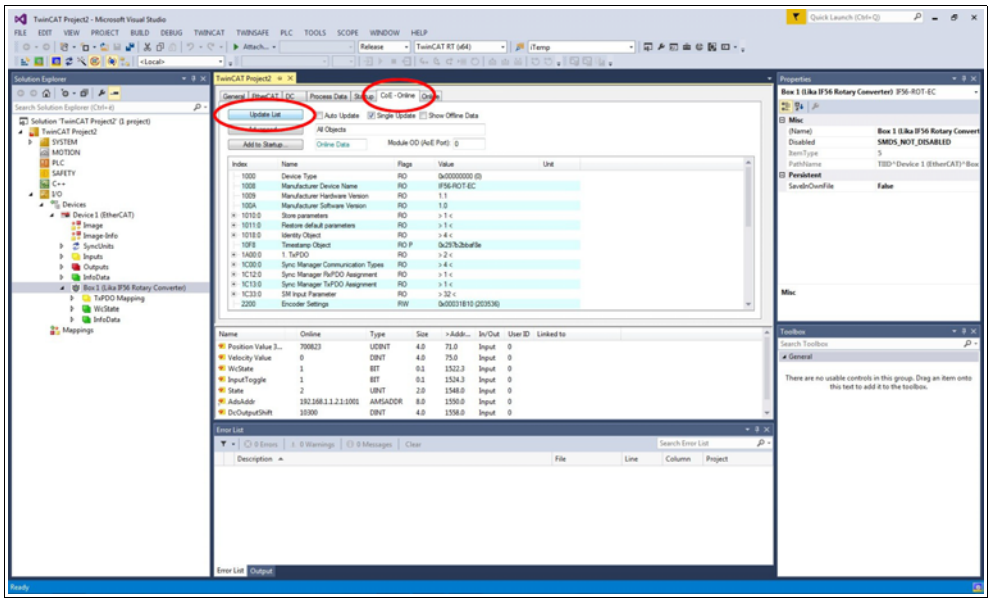


Figure 55 - Updating the object list

10. Or, in the same **CoE - Online** tabbed page, press the **ADVANCED...** button and then press the **OK** button in the **Advanced Settings** page that appears.

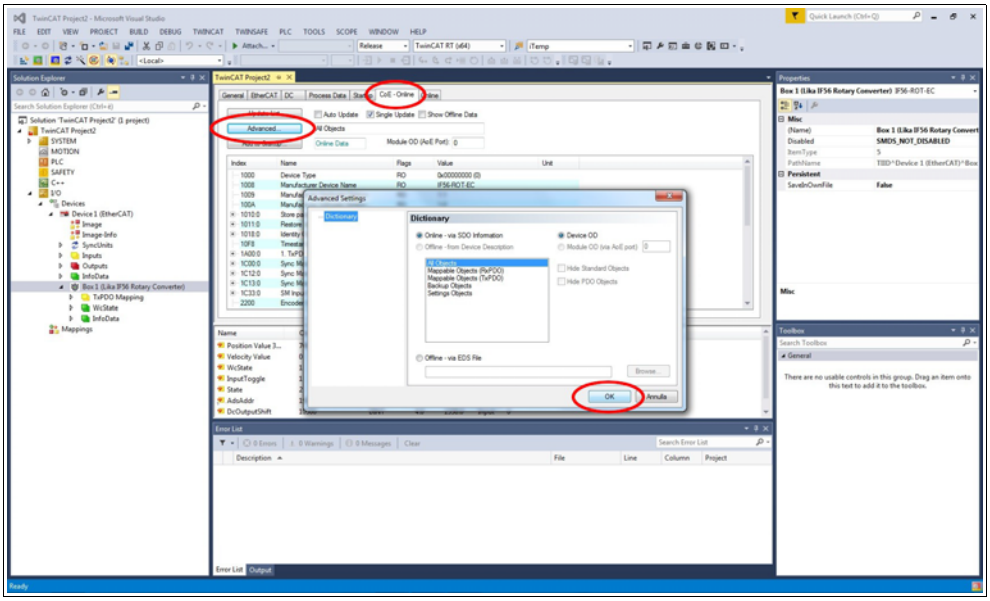


Figure 56 - Updating the object list

11. Scroll through the objects and double click on the object 6000 (see the 6000-00 Operating parameters object on page 117).
12. Set the bit 15 **Mask Upgrade Firmware** to 1 (1000 0000 0000 0000 in binary notation = 32768 dec = 8000h), confirm pressing the **OK** button and check that the PWR LED starts flashing: it shall flash red while the upgrade is in progress.

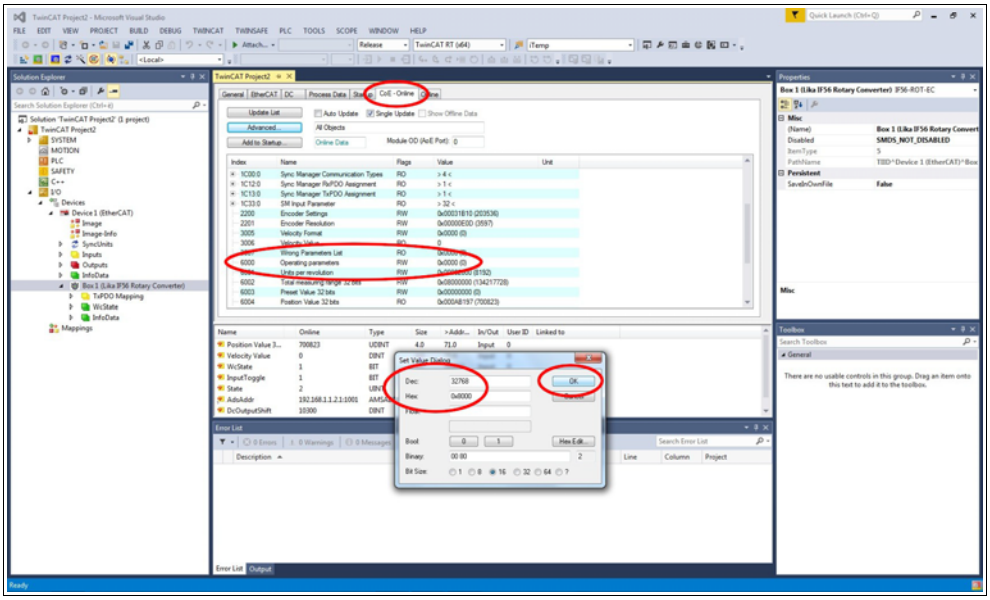


Figure 57 - Setting the Mask Upgrade Firmware bit

13. To check whether the firmware upgrade procedure has been completed successfully enter the **CoE - Online** tabbed page and check the value next to the **100A Manufacturer Software Version** object.

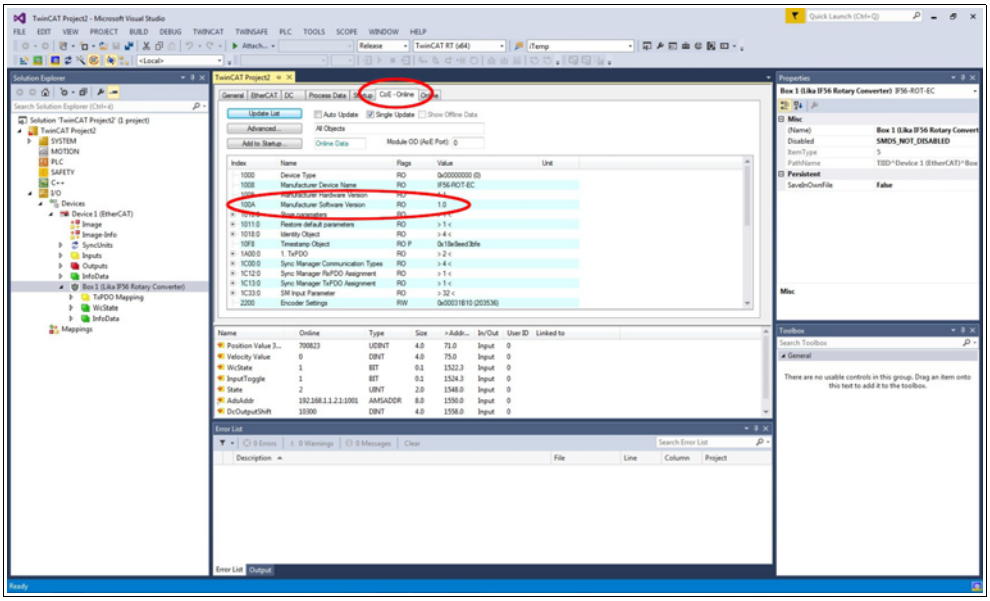


Figure 58 - Manufacturer Software Version

14. Otherwise, enter the **Online** tabbed page and press the **PRE-OP** button in the **State Machine** group box; if everything is ok, the converter enters the PREOPERATIONAL state (the **PREOP** message appears next to the **Current State** item in the same box).

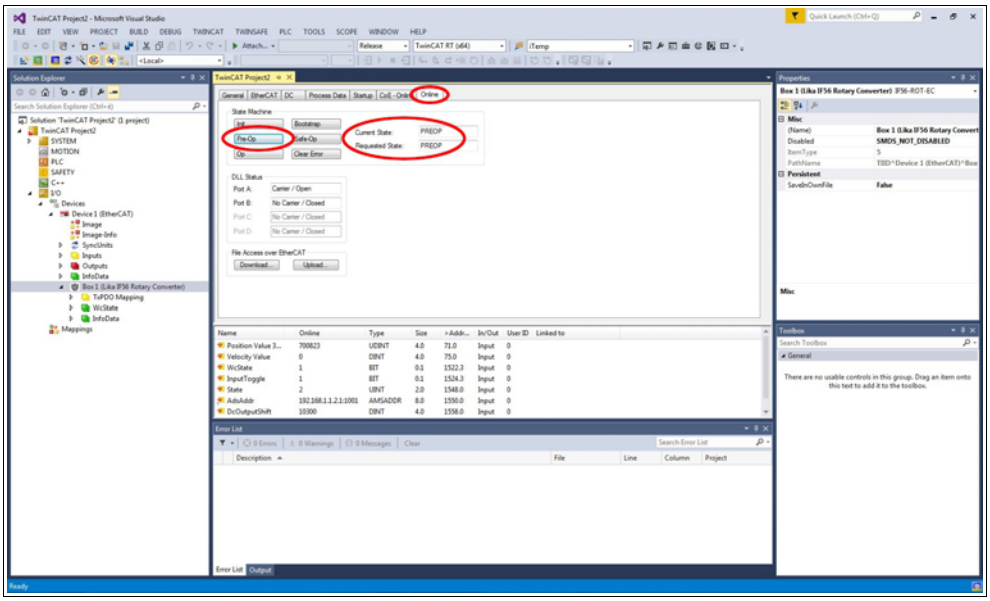


Figure 59 - Preoperational state

7.3.2 Improvements for better FoE performance

If required it is possible to increase the Mailbox sizes on the Slave. To do this press the **ADVANCED SETTINGS** button under the **EtherCAT** tabbed page and navigate to **Mailbox**. Edit the value in the **Out Size (hex)** field and/or the **In Size (hex)** field. The maximum size is 1,486 bytes (0x05CE).

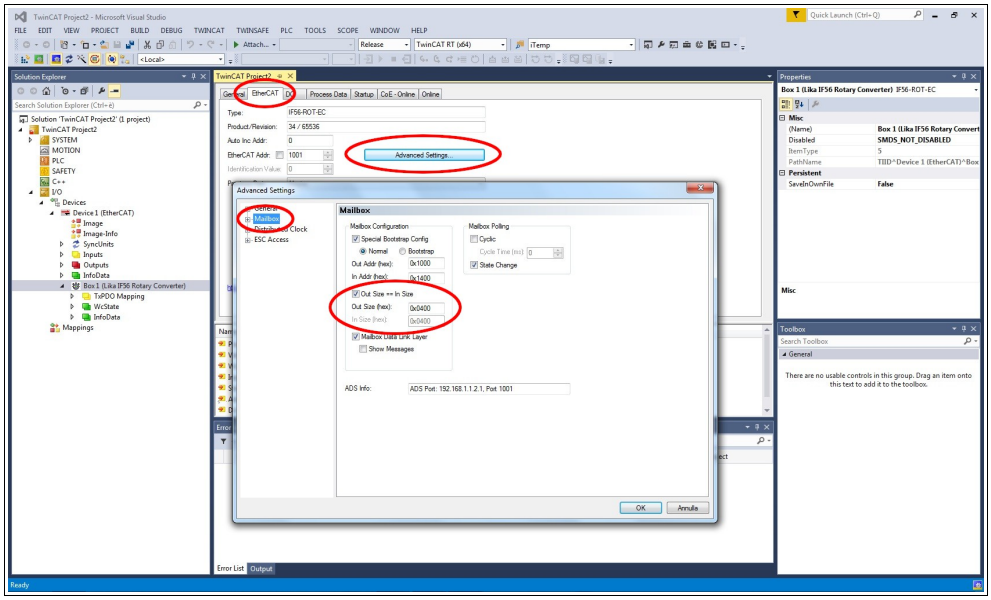


Figure 60 - Increasing the Mailbox sizes

For any specification on the FoE protocol, please refer to "ETG.1000 EtherCAT Specification" document available at the address **www.ethercat.org**.
Please refer also to the "6.9 Firmware update" section on page 73.

8 – Integrated Web Server

8.1 Integrated web server – Preliminary information

EtherCAT converters from Lika Electronic integrate a web server. This web-based user interface is designed to offer helpful functions and deliver complete information on the device that can be accessed through the Internet.
In particular it allows:

- to display the current position and speed values;
- to set some parameters such as the preset and the code sequence;
- to display and check the parameters that are set currently;
- to monitor the converter;
- to update the firmware.

The web server can be accessed from any PC running a web browser. Since its only requirement is a HTTP connection between the web browser and the web server running on the device, it is perfectly fitted also for remote access scenarios.

Before opening the web server of the EtherCAT converter please ascertain that the following requirements are fully satisfied:

- the converter is connected to the network;
- the converter has a valid IP address;
- the PC is connected to the network;
- a web browser (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, ...) is installed in the PC or in the device used for connection.

NOTE
This web server has been tested and verified using the following web browsers:

- Internet Explorer IE11 version 11.1593.14393.0
- Mozilla Firefox version 116.0.1
- Google Chrome version 115.0.5790.111
- Opera version 68.0.3618.165

NOTE
Please note that the appearance of the snapshots may vary depending on the web browser used. The following snapshots were taken from Google Chrome.

WARNING
Please note that since EtherCAT is not based on TCP/IP, there is no need to administer MAC addresses or IP addresses. But you need the IP Address to reach the web server. To know the IP address assigned by Beckhoff see the following pages.

WARNING

The web server can be accessed only if the converter is in the Pre-Operational or Operational or Safe-Operational states. Enter the **Online** tabbed page to check the current state of the converter (see the **Current State** information field in the **State machine** group box). To activate the Pre-Operational or Operational or Safe-Operational states, press the **PRE-OP** or **OP** or **SAFE-OP** buttons in the **State machine** group box respectively.

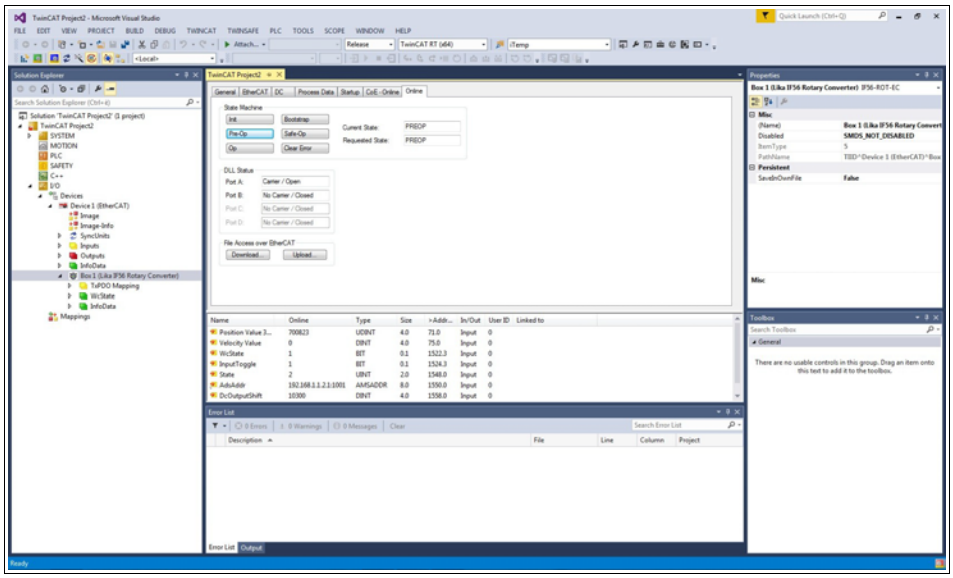


Figure 61 - Setting the State Machine

8.2 Web server Home page

To open the web server of the EtherCAT converter proceed as follows.

1. Check the IP address of the converter first. To do this, enter the **EtherCAT** tabbed page and then press the **ADVANCED SETTINGS...** button.

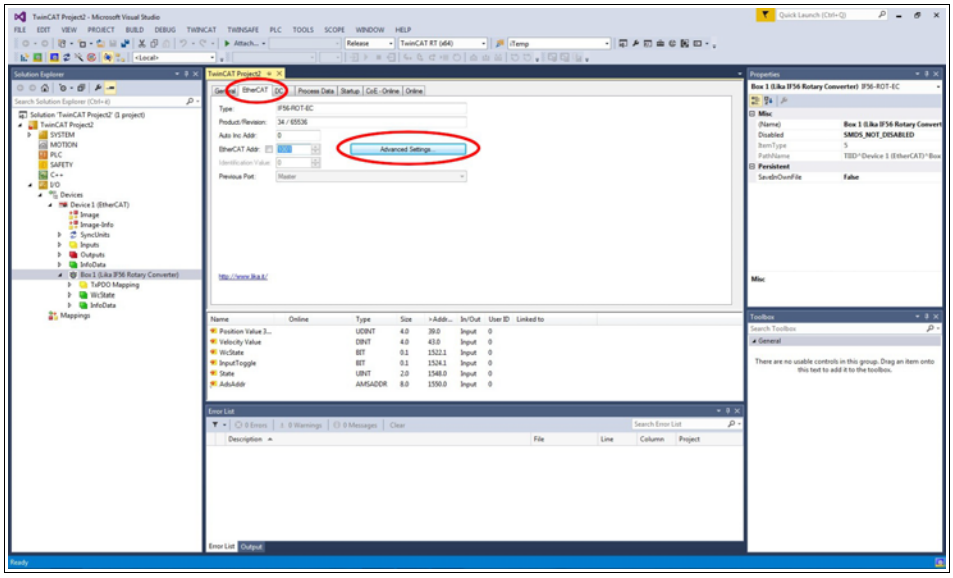


Figure 62 - EtherCAT tabbed page

2. In the **Advanced Settings** page open the **Mailbox** list and then press the **EoE** command: the **EoE** group box will be displayed. Check the assigned IP address next to the **IP Address** item (192.168.1.0 in Figure 63).

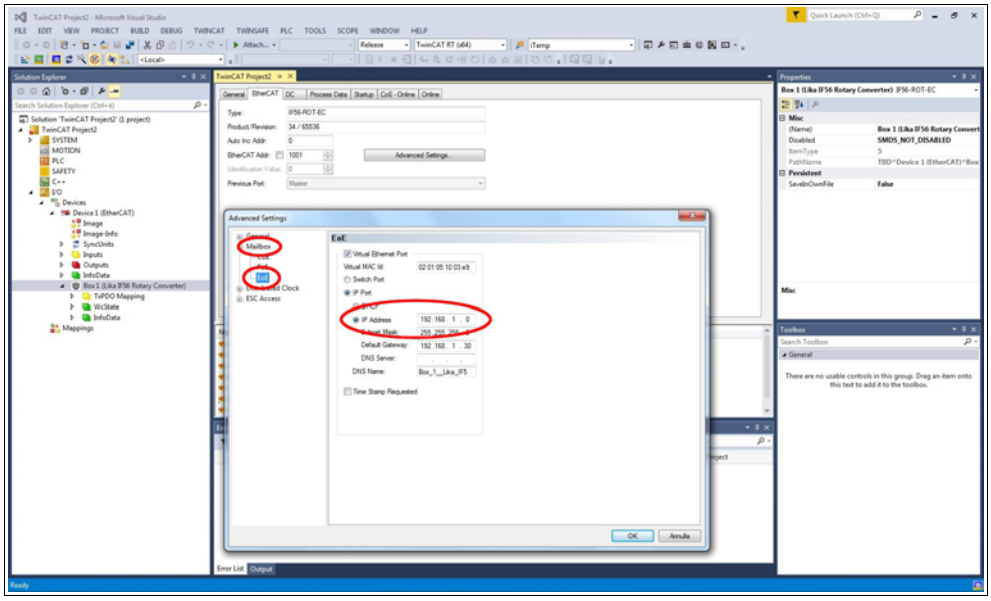


Figure 63 - Advanced Settings page

3. Type the IP address of the converter you want to connect to (in the example: 192.168.1.0) in the address bar of your web browser and confirm by pressing **ENTER**.

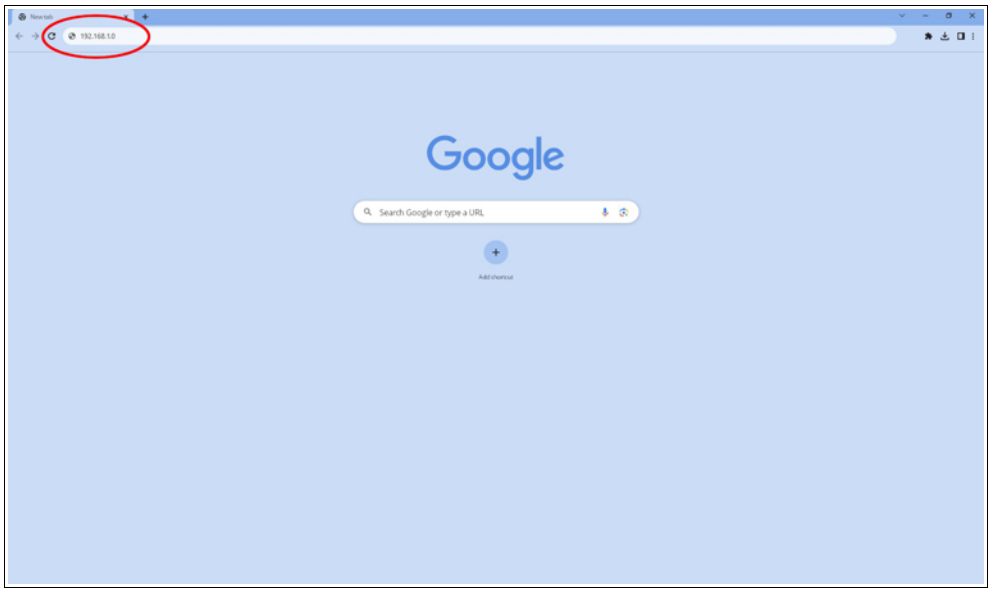


Figure 64 - Opening the web server

4. As soon as the connection is established, the web server **Home** page will appear on the screen.

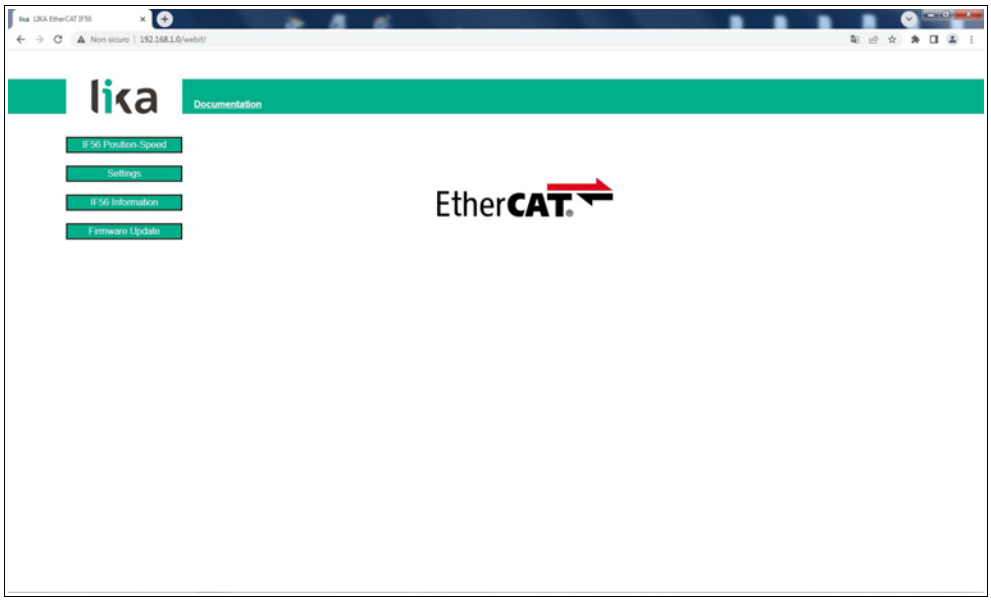


Figure 65 - Web server Home page

Some commands are available in the menu bar of the **Home** page.

Press the **Lika logo** to enter Lika's web site (www.lika.biz).

Press the **DOCUMENTATION** button to enter the EtherCAT converter technical documentation page available on Lika's web site (https://www.lika.it/eng/products/encoder-interfaces/encoder-interfaces) where specific technical information and documentation concerning the EtherCAT converter can be found.

Furthermore some buttons are available in the left navigation bar. All the pages except the **Firmware Update** page are freely accessible through the buttons in the bar. The **Firmware Update** page is protected and requires a password.
These buttons allow to enter specific pages where information and diagnostics on the connected converter as well as useful functions can be achieved.
They are described in the following sections.

8.3 Converter position and speed

Press the **IF56 POSITION-SPEED** button in the left navigation bar of the Web server **Home** page to enter the page where the current encoder position and the current encoder speed are displayed.

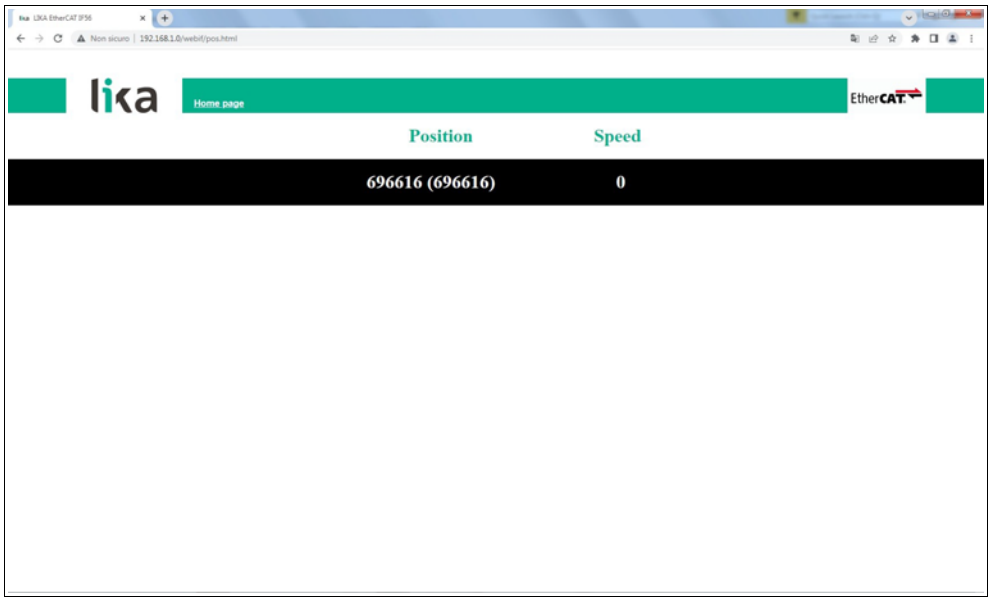


Figure 66 - Encoder position and speed page

The first value (under the Position item) is the absolute position calculated considering scaling and preset functions, if activated; the value in brackets is the raw value (physical absolute position). Both encoder positions are expressed in counts. For any information refer to the **6004-00 Position Value 32 bits** object on page 128.

The current encoder speed (under the Speed item) is expressed according to the engineering unit set next the **3005-00 Velocity Format R** / **3005-00 Velocity Format L** object on page 112 ff (by default it is expressed in counts per second). For any information refer to the **3006-00 Velocity Value** object on page 112.

NOTE
The current encoder position and speed values are real-time processed and updated continuously (every 200 msec. on the screen).

Press the **HOMEPAGE** button to move back to the Web server **Home** page.

8.3.1 Specific notes on using Internet Explorer

The following options must be set properly on Internet Explorer in order to get the **IF56 position and speed** page to be updated continuously.

- Open the **Settings** menu;
- open the **Internet Options** property sheet;
- in the **General** tabbed page, press the **Setting** button available in the **History Browsing** section;
- under **Check for newer versions of stored pages**, click **Every time I visit the webpage**;
- press the **OK** button to confirm whenever requested.

8.4 Setting the objects

Press the **SETTINGS** button in the left navigation bar of the Web server **Home** page to enter the **Set Converter Objects** page. In this page the read-write access objects available in the Manufacturer Specific Profile Area and in the Standardised Profile Area of the EtherCAT converter are displayed and their value can be changed.
For complete information on the converter objects please refer to the "Manufacturer Specific Profile Area objects" on page 100; and to the "Standardised Profile Area objects (DS406)" section on page 117.

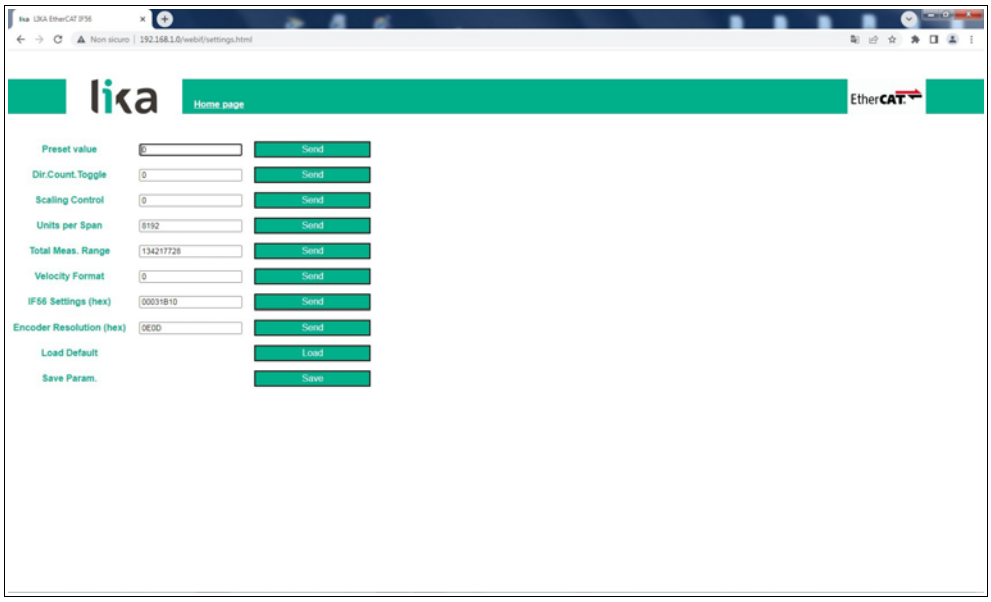


Figure 67 - Page for setting the Converter Objects

The values that are currently set in the converter are displayed in the fields.

To change any value enter a suitable value next to the desired parameter and then press the **SEND** button on the right. Values have to be set in either decimal or hexadecimal notation, as indicated.
For complete information on the available objects please refer to the "Manufacturer Specific Profile Area objects" on page 100; and to the "Standardised Profile Area objects (DS406)" section on page 117.

EXAMPLE
The 6001-00 Units per revolution object is currently set to "**8192**" (see the box next to the **Units per Span** item the Figure above). To change the set value enter a suitable value in the box and then press the **SEND** button on the right in the same line to confirm.

NOTE
Please note that, after pressing the **SEND** button, the set value is saved temporarily in the objects. To save it permanently, please press the **SAVE** button next to the **Save Param.** item. Should the power supply be turned off without saving data, the values that have not been saved on the Flash EEProm will be lost! For more information refer to the **1010-01 Store parameters** object on page 94.

Press the **LOAD** button next to the **Load Default** item to restore all parameters to default values. Default values are set at the factory by Lika Electronic engineers to allow the operator to run the device for standard operation in a safe mode. This function can be useful, for instance, to restore the factory values in case the converter is set incorrectly and you are not able to resume the proper operation. For more information refer to the **1011-01 Restore default parameters** object on page 94.

WARNING
The execution of this command causes all parameters which have been set previously to be overwritten!

NOTE
At each confirmation of the set parameters, a message will appear under the buttons. It informs whether the operation has been accomplished properly or an error occurred (for example **Setting executed correctly!** if everything went well).

Press the **HOMEPAGE** button to move back to the Web server **Home** page.

8.5 Converter information (EtherCAT objects)

Press the **IF56 INFORMATION** button in the left navigation bar of the Web server **Home** page to enter the **IF56 Information** page. In this page the list of the most useful EtherCAT objects available for the converter is displayed. Values of the objects are expressed in either hexadecimal or decimal notation or in a string format.

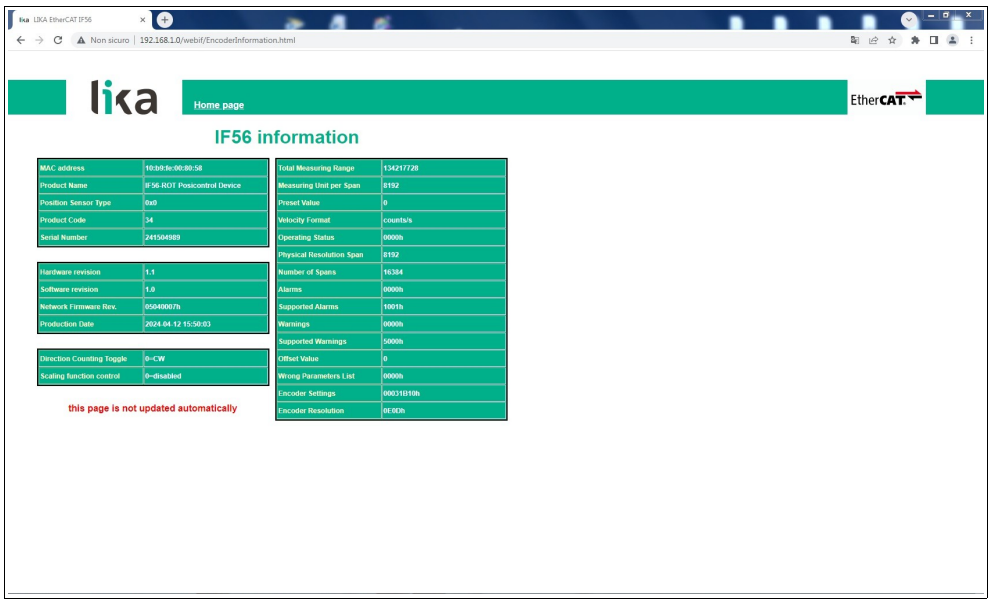


Figure 68 - IF56 Information page

For a complete description of the available converter objects please refer to the "Manufacturer Specific Profile Area objects" on page 100; and to the "Standardised Profile Area objects (DS406)" section on page 117.

NOTE
Please note that the values shown in the **IF56 Information** page are "frozen" in the moment when the page is displayed. To update the values you must refresh the web page.

NOTE
The objects in the **IF56 Information** page cannot be changed even though they are read-write access objects. To change the set values please enter the **Set Converter Objects** page (see on page 153).

Press the **HOMEPAGE** button to move back to the Web server **Home** page.

8.6 Firmware update

Press the **FIRMWARE UPDATE** button in the left navigation bar of the Web server **Home** page to enter the **Firmware Update** page. Please note that this is a password protected page, thus a password is requested to access the page.

Password: **LiKa** ("L" and "K" in uppercase letters; "i" and "a" in lowercase letters)

WARNING
Firmware updating process has to be accomplished by skilled and competent personnel. It is mandatory to perform the update according to the instructions provided in this section.
Before installation always ascertain that the firmware program is compatible with the hardware and software of the device. Furthermore never turn off the power supply during the flash update. In case of flash update error, the program is lost irreversibly (there is not a bootloader) and the device must be sent back to Lika Electronic for restoring.

This operation allows to update the unit firmware by downloading updating data to the flash memory.
Firmware is a software program which controls the functions and operation of a device; the firmware program, sometimes referred to as "user program", is stored in the flash memory integrated inside the unit. These converters are designed so that the firmware can be easily updated by the user himself. This allows Lika Electronic to make new improved firmware programs available during the lifetime of the product.
Typical reasons for the release of new firmware programs are the necessity to make corrections, improve, and even add new functionalities to the device.
The firmware upgrading program consists of a single file having .ZIP extension. It is released by Lika Electronic Technical Assistance & After Sale Service.

If the latest firmware version is already installed in the unit, you do not need to proceed with any new firmware installation. The firmware version currently installed can be read next to the **Software revision** item in the **Converter Information** page after connection to the web server (see on page 155).

NOTE
If you are not confident that you can perform the update successfully please contact Lika Electronic Technical Assistance & After Sale Service.

Before proceeding with the firmware update please ascertain that the following requirements are fully satisfied:

- the converter is connected to the Ethernet network;
- the converter has a valid IP address;
- the PC is connected both to the network and to the IO controller;

- a web browser (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, ...) is installed in the PC or device used for connection;
- you have the .ZIP file for firmware update.

To update the firmware program please proceed as follows.

1. Press the **FIRMWARE UPDATE** button in the left navigation bar of the Web server **Home** page to enter the **Firmware Update** page.
2. The operator is requested to submit a password before starting the firmware update procedure.

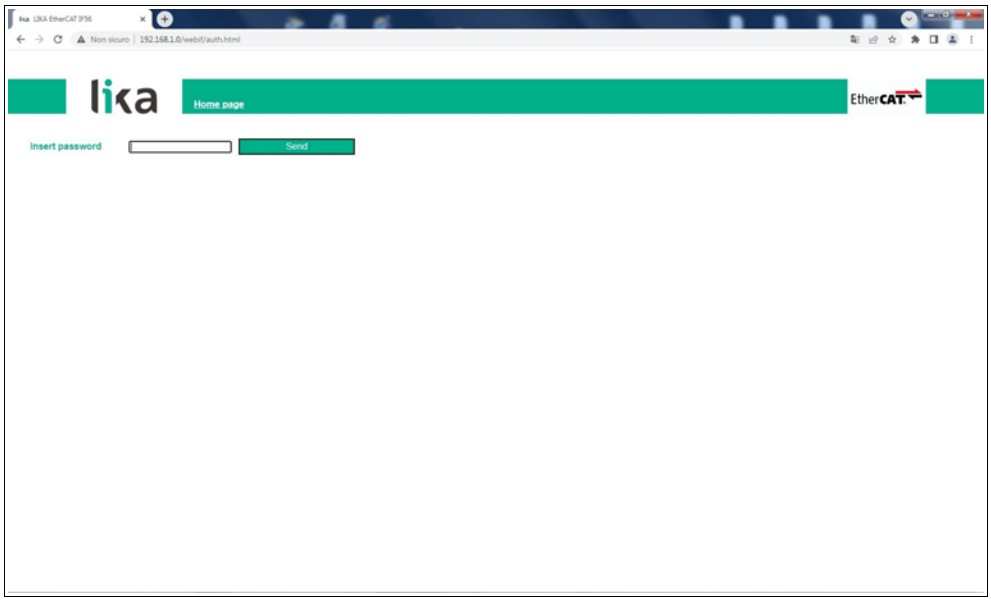


Figure 69 - Firmware Update page

3. In the **Insert password** text box type the password **LiKa** ("L" and "K" in uppercase letters; "i" and "a" in lowercase letters) and then press the **SEND** button.
4. If the password you typed is wrong, the following warning message will appear on the screen: **WRONG PASSWORD INSERTED. RETRY**. Please retype the password and confirm.

5. If the password you typed is correct, the **Firmware Update** page will appear on the screen.

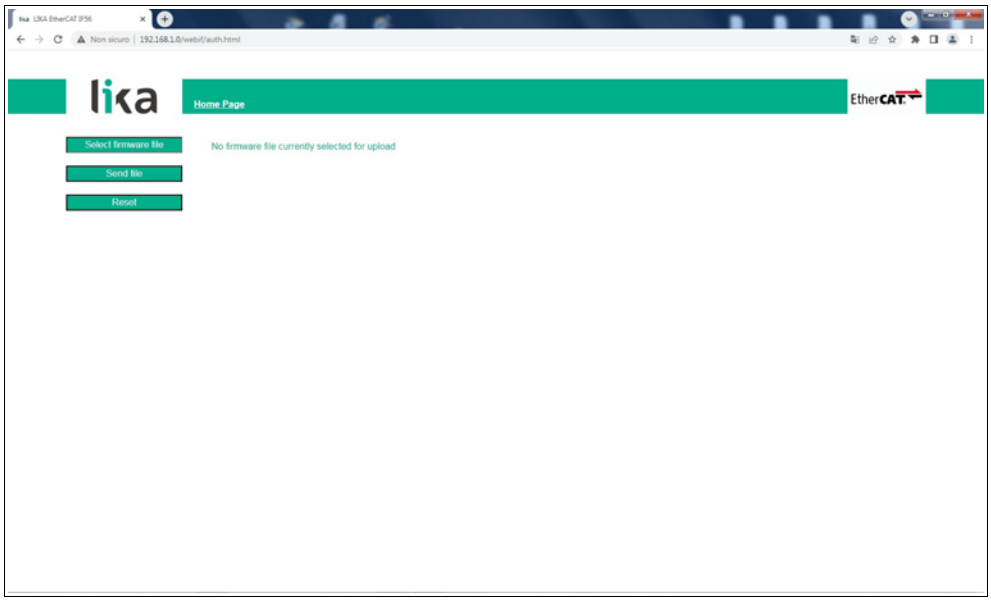


Figure 70 - Firmware Update page

6. Press the **SELECT FIRMWARE FILE** button; once you press the **SELECT FIRMWARE FILE** button an **OPEN** dialog box appears on the screen: open the folder where the firmware updating .ZIP file released by Lika Electronic is located, select the file and confirm. Please check the file properties and ascertain that you are installing the correct update file.

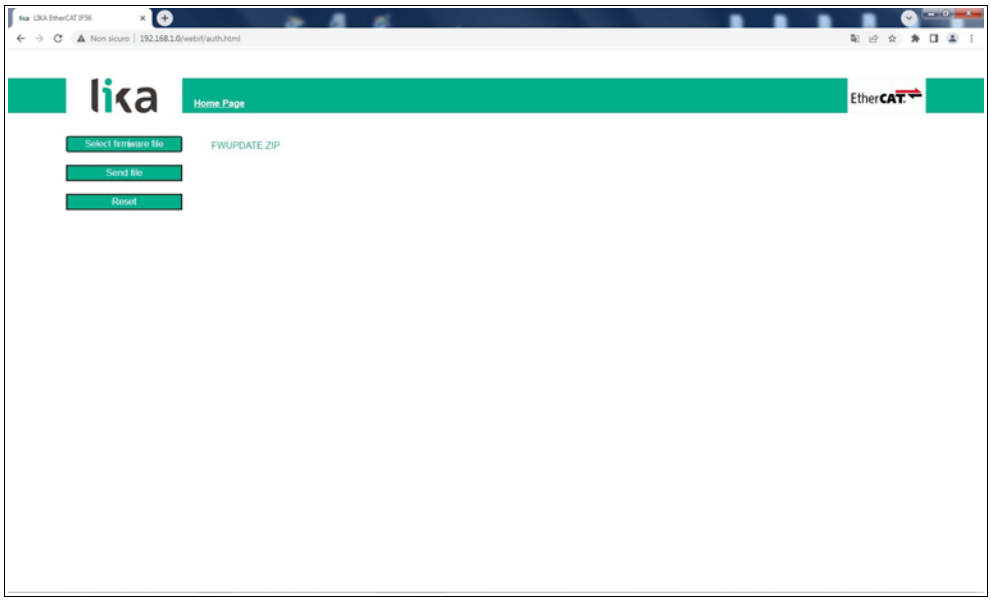


Figure 71 - Selecting the firmware update .zip file

WARNING
Before installation always ascertain that the firmware program is compatible with the hardware and software of the device.
Never turn the power supply off during the flash update operation.

7. Press the **SEND FILE** button to start the upload of the firmware program.
8. During the operation and as soon as the operation is carried out successfully, some messages will appear on the screen.

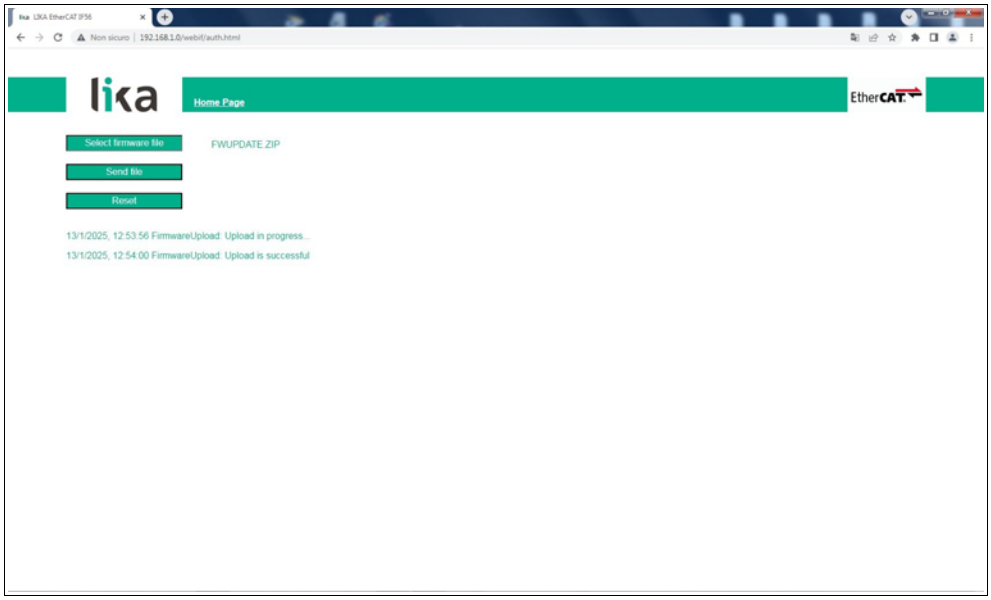


Figure 72 - Messages during firmware upload

9. Finally press the **RESET** button to automatically reset and restart the converter and complete the operation.

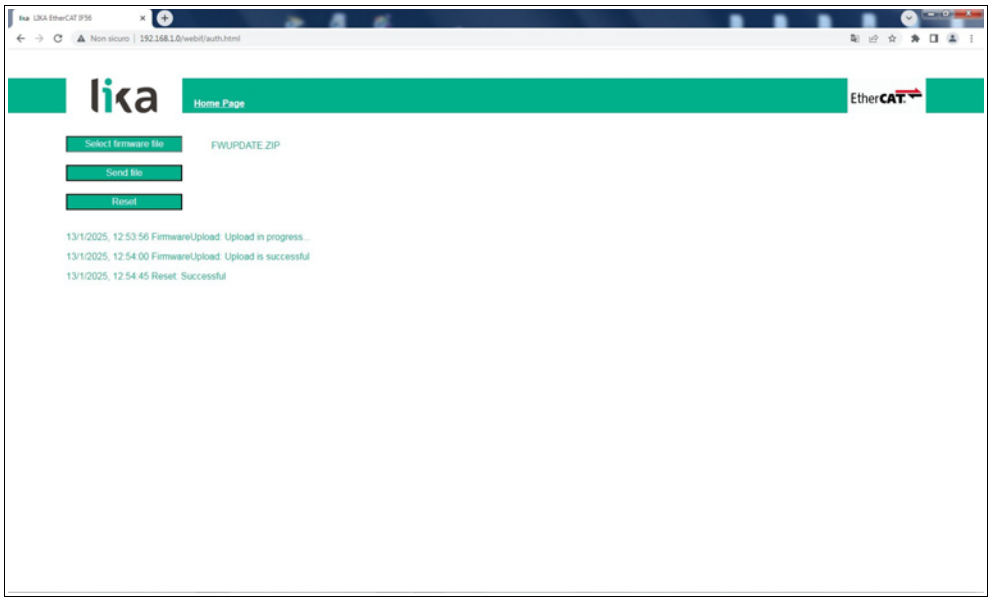


Figure 73 - Firmware update process accomplished

NOTE
While downloading the firmware updating program, unexpected conditions may arise which could lead to a failure of the installation process. When such a matter occurs, the download process cannot be carried out successfully and thus the operation is aborted. In case of flash update error, please switch the converter off and then on again and retry the operation.

Press the **HOMEPAGE** button to move back to the Web server **Home** page.

9 – Default parameters list

Default values are expressed in hexadecimal notation.

Parameters list	Default values		
1000-00 Device Type	0000 0000 = generic device		
1008-00 Manufacturer Device Name	494635362D524F542D4543 = "IF56-ROT-EC" = IF56 converter with EtherCAT interface for SSI and BiSS rotary encoders 494635362D4C494E2D4543 = "IF56-LIN-EC" = IF56 converter with EtherCAT interface for SSI and BiSS linear encoders		
1009-00 Manufacturer Hardware Version	device dependent		
100A-00 Manufacturer Software Version	device dependent		
1018 Identity Object **01 Vendor ID** **02 Product Code** **03 Revision Number** **04 Serial Number**	4 0000 012E 0000 0022 = IF56 converter for rotary encoders 0000 0023 = IF56 converter for linear encoders 0001 0000 = Lika EtherCAT IF56 series converter device dependent		
1A00-00 TxPDO mapping parameter **01 Mapped Object 001** **02 Mapped Object 002**	02 6004 0020 3006 0020		
2200-00 Rotary Encoder Settings **Bits 0-1 Protocol R** **Bit 2 SSI error bit R** **Bit 4 SSI output code R** **Bit 7 Bypass R** **Bits 8 ... 15 Number of clocks R**	0003 1910 00 0 1 0 19		

Bits 16 ... 19 Sensor communication frequency R	3		
2200-00 Linear Encoder Settings Bit 0-1 Protocol L Bit 2 SSI error bit L Bit 4 SSI output code L Bit 7 Bypass L Bits 8 ... 15 Max No of Information (bit) L Bits 16 ... 23 Number of clocks L Bits 24 ... 27 Sensor communication frequency L	0319 1310 00 0 1 0 13 19 3		
2201-00 Rotary Encoder Resolution Multiturn resolution (bits) Singleturn resolution (bits)	0000 0C0D 0C 0D		
2201-00 Linear Encoder Resolution	0000 2710		
3005-00 Velocity Format R	0 = counts per second		
3005-00 Velocity Format R	0 = counts per second		
6000-00 Operating parameters Bit 0 Code sequence Bit 2 Scaling function Bit 15 Mask Upgrade Firmware	0000 0 = disabled 0 = CW 0 = disabled		
6001-00 Units per revolution	0000 2000 (8,192)		
6001-00 Total measuring range	$2^{\text{Max No of Information (bit) L}}$		
6002-00 Total measuring range 32 bits	0800 0000 (134,217,728)		
6002-00 Total measuring range	$2^{\text{Max No of Information (bit) L}}$		
6003-00 Preset Value 32 bits	00000 0000		
6005-01 Position step setting	according to 6501-00 Measuring step		

6501-00 Hardware Singleturn Resolution	according to **Singleturn resolution (bits)**		
6501-00 Measuring step	according to 2201-00 Linear Encoder Resolution		
6502-00 Hardware number of revolutions	according to **Multiturn resolution (bits)**		
6502-00 Hardware number of revolutions	0000 0001		

Document release	Release date	Description	HW	SW	XML file version
1.0	29.01.2025	First issue	1.1	1.0	V1_0

Lika Electronic
Via S. Lorenzo, 25 • 36010 Carrè (VI) • Italy

Tel. +39 0445 806600
Fax +39 0445 806699

info@lika.biz • www.lika.biz