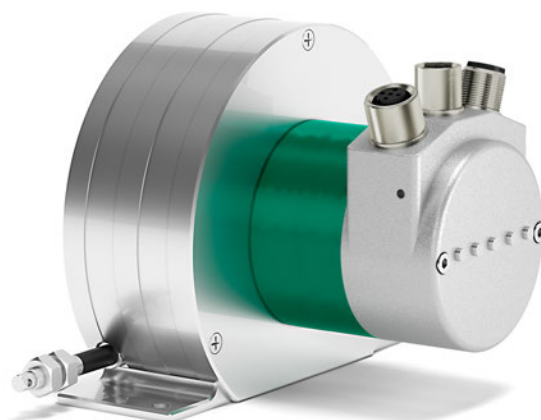


## SFAM3-05000-EC SFAM4-10000-EC



**EtherCAT®**

in compliance with ETG.1000

- 5,000 mm (196.85") & 10,000 mm (393.7") draw-wire encoder
- Integrated 27 bit absolute multiturn encoder
- Programmable resolution down to 24 µm
- Complies with ETG.1000 specifications
- Implements CoE, FoE, & EoE protocols and the EtherCAT State Machine
- With Energy Harvesting Technology

### Suitable for the following models:

- SFAM3-05000-EC2-08192-RM12
- SFAM4-10000-EC2-08192-RM12

### General Contents

Preliminary information	10
1 - Safety summary	22
2 - Identification	24
3 - Mechanical installation	25
4 - Electrical connections	30
5 - Getting started	37
6 - Quick reference with TwinCAT	40
7 - EtherCAT® interface	66
8 - Integrated Web Server	113
9 - Default parameters list	127

This publication was produced by Lika Electronic s.r.l. 2026. All rights reserved. Tutti i diritti riservati. Alle Rechte vorbehalten. Todos los derechos reservados. Tous droits réservés.

This document and information contained herein are the property of Lika Electronic s.r.l. and shall not be reproduced in whole or in part without prior written approval of Lika Electronic s.r.l. Translation, reproduction and total or partial modification (photostat copies, film and microfilm included and any other means) are forbidden without written authorisation of Lika Electronic s.r.l.

The information herein is subject to change without notice and should not be construed as a commitment by Lika Electronic s.r.l. Lika Electronic s.r.l. reserves the right to make all modifications at any moments and without forewarning.

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](mailto:info@lika.it) for submitting your comments, suggestions and criticisms.

The logo for Lika Electronic s.r.l. features the word "lika" in a bold, lowercase, sans-serif typeface. The letters are black and have a modern, clean appearance.

# General contents

User's guide.....	1
General contents.....	3
Subject Index.....	7
Typographic and iconographic conventions.....	9
Preliminary information.....	10
Glossary of EtherCAT terms.....	12
List of abbreviations.....	19
References.....	21
<b>1 – Safety summary.....</b>	<b>22</b>
1.1 Safety.....	22
1.2 Electrical safety.....	22
1.3 Mechanical safety.....	23
<b>2 – Identification.....</b>	<b>24</b>
<b>3 – Mechanical installation.....</b>	<b>25</b>
3.1 Overall dimensions.....	25
3.2 Mounting instructions.....	26
3.2.1 Handling the wire.....	26
3.3 Useful information.....	28
3.4 Maintenance.....	29
<b>4 – Electrical connections.....</b>	<b>30</b>
4.1 M12 connectors.....	31
4.1.1 PWR Power supply connector (Figure 1).....	31
4.1.2 BUS IN and BUS OUT connectors (Figure 1).....	31
4.2 Network configuration: topologies, cables, hubs, switches – Recommendations.....	32
4.3 Addressing.....	32
4.4 Line Termination.....	33
4.5 Ground connection.....	33
4.6 Diagnostic LEDs (Figure 1).....	33
4.7 EtherCAT states.....	36
<b>5 – Getting started.....</b>	<b>37</b>
5.1 Quick setting and main functions.....	37
5.1.1 Setting the scaling function and custom resolution.....	38
5.1.2 Reading the absolute position.....	38
5.1.3 Reading the velocity value.....	39
5.1.4 Setting and executing the preset.....	39
5.1.5 Saving data.....	39
5.1.6 Restoring defaults.....	39
<b>6 – Quick reference with TwinCAT.....</b>	<b>40</b>
6.1 Setting up EtherCAT® communication. Example with TwinCAT3.....	40
6.2 About TwinCAT.....	40
6.2.1 User interface components.....	41
6.3 Adding the XML file.....	42
6.4 System configuration using TwinCAT software system from Beckhoff.....	43
6.4.1 Creating a project.....	43
6.4.2 Adding an I/O device (network card).....	45
6.4.3 Adding an I/O module (Box).....	48

6.4.4 Tabbed pages.....	51
6.5 Setting the communication mode.....	55
6.5.1 Synchronous with SM3.....	55
6.5.2 Synchronous with DC (SYNCO).....	56
6.6 Process Data Objects.....	57
6.7 CoE Object Dictionary.....	59
6.8 Online Data.....	60
6.9 Firmware update.....	61
<b>7 - EtherCAT® interface.....</b>	<b>66</b>
7.1 Basic Information on EtherCAT® Protocol.....	66
7.1.1 Data transfer.....	67
7.1.2 ISO/OSI Layer model.....	68
7.1.3 Topology.....	68
7.1.4 Line Termination.....	69
7.1.5 Addressing.....	70
7.1.6 Communication mode.....	71
FreeRun.....	71
Synchronous with SM3.....	72
Synchronous with DC SYNCO.....	73
7.1.7 EtherCAT State Machine (ESM).....	73
7.1.8 Slave configuration.....	74
7.1.9 Timing and Synchronization.....	75
Sync Manager.....	75
Buffered Mode (3-Buffer Mode).....	75
Mailbox Mode (1-Buffer Mode).....	75
7.2 CANopen Over EtherCAT (CoE).....	77
7.2.1 XML file.....	77
7.2.2 Communication messages.....	78
7.2.3 Process Data Objects (PDO).....	78
7.2.4 Service Data Objects (SDO).....	79
7.2.5 Object dictionary.....	79
Communication Profile Area objects (DS301).....	81
<b>1000-00 Device Type.....</b>	<b>81</b>
<b>1008-00 Manufacturer Device Name.....</b>	<b>81</b>
<b>1009-00 Manufacturer Hardware Version.....</b>	<b>81</b>
<b>100A-00 Manufacturer Software Version.....</b>	<b>81</b>
<b>1010-01 Store parameters.....</b>	<b>82</b>
<b>1011-01 Restore default parameters.....</b>	<b>82</b>
<b>1018 Identity Object.....</b>	<b>83</b>
01 Vendor ID.....	83
02 Product Code.....	83
03 Revision Number.....	83
04 Serial Number.....	83
<b>10F8-00 Timestamp Object.....</b>	<b>84</b>
<b>1A00-00 TxPDO mapping parameter.....</b>	<b>84</b>
01 Mapped Object 001.....	84
02 Mapped Object 002.....	84
<b>1C00-00 Sync Manager Communication Type.....</b>	<b>85</b>
01 SM MailBox Receive (SM0).....	85
02 SM MailBox Send (SM1).....	85
03 SM PDO output (SM2).....	85

04 SM PDO input (SM3).....	85
<b>1C12-00 Sync Manager 1 RxPDO Assignment.....</b>	<b>85</b>
01 Subindex 001.....	85
<b>1C13-00 Sync Manager 1 TxPDO Assignment.....</b>	<b>86</b>
01 Subindex 001.....	86
<b>1C33-00 SM Sync Manager input parameter.....</b>	<b>86</b>
01 Sync mode.....	86
02 Cycle time.....	86
03 Shift Time.....	86
04 Sync modes supported.....	87
05 Minimum cycle time.....	87
06 Calc and copy time.....	87
0B SM event missed counter.....	87
0C Cycle exceeded counter.....	87
20 Sync error.....	87
Manufacturer Specific Profile Area objects.....	88
<b>3005-00 Velocity Format.....</b>	<b>88</b>
<b>3006-00 Velocity Value.....</b>	<b>88</b>
<b>3007-00 Wrong parameters list.....</b>	<b>88</b>
Units per revolution exceeded.....	89
Total Measuring Range exceeded.....	89
Preset value exceeded.....	89
Offset value exceeded.....	89
Speed Format exceeded.....	89
Standardised Profile Area objects (DS406).....	91
<b>6000-00 Operating parameters.....</b>	<b>91</b>
Code sequence.....	91
Scaling function.....	91
Mask Upgrade Firmware.....	92
<b>6001-00 Units per revolution.....</b>	<b>92</b>
<b>6002-00 Total measuring range.....</b>	<b>94</b>
<b>6003-00 Preset Value.....</b>	<b>97</b>
<b>6004-00 Position Value.....</b>	<b>99</b>
<b>6500-00 Operating Status.....</b>	<b>101</b>
Code sequence.....	101
Scaling function.....	101
<b>6501-00 Hardware counts per revolution.....</b>	<b>101</b>
<b>6502-00 Hardware number of turns.....</b>	<b>102</b>
<b>6503-00 Errors.....</b>	<b>102</b>
Position error.....	102
Flash memory error.....	103
<b>6504-00 Supported errors.....</b>	<b>103</b>
<b>6505-00 Warnings.....</b>	<b>103</b>
Machine data not valid.....	103
Position Warning.....	103
<b>6506-00 Supported warnings.....</b>	<b>104</b>
<b>6509-00 Offset.....</b>	<b>104</b>
7.2.6 SDO Abort codes.....	105
7.2.7 Emergency Error Codes.....	106
7.2.8 AL Status Error Codes.....	106

7.3 File Over EtherCAT (FoE).....	107
7.3.1 Downloading files.....	107
7.3.2 Improvements for better FoE performance.....	112
<b>8 – Integrated Web Server.....</b>	<b>113</b>
8.1 Integrated web server – Preliminary information.....	113
8.2 Web server Home page.....	114
8.3 Encoder position and speed.....	117
8.3.1 Specific notes on using Internet Explorer.....	118
8.4 Setting the objects.....	119
8.5 Encoder information (EtherCAT objects).....	121
8.6 Firmware update.....	122
<b>9 – Default parameters list.....</b>	<b>127</b>

# Subject Index

## 1

1000-00 Device Type.....	81
1008-00 Manufacturer Device Name.....	81
1009-00 Manufacturer Hardware Version.....	81
100A-00 Manufacturer Software Version.....	81
1010-01 Store parameters.....	82
1011-01 Restore default parameters.....	82
1018 Identity Object.....	83
10F8-00 Timestamp Object.....	84
1A00-00 TxPDO mapping parameter.....	84
1C00-00 Sync Manager Communication Type...	85
1C12-00 Sync Manager 1 RxPDO Assignment....	85
1C13-00 Sync Manager 1 TxPDO Assignment....	86
1C33-00 SM Sync Manager input parameter....	86

## 3

3005-00 Velocity Format.....	88
3006-00 Velocity Value.....	88
3007-00 Wrong parameters list.....	88

## 6

6000-00 Operating parameters.....	91
6001-00 Units per revolution.....	92
6002-00 Total measuring range.....	94
6003-00 Preset Value.....	97
6004-00 Position Value.....	99
6500-00 Operating Status.....	101
6501-00 Hardware counts per revolution.....	101
6502-00 Hardware number of turns.....	102
6503-00 Errors.....	102
6504-00 Supported errors.....	103
6505-00 Warnings.....	103
6506-00 Supported warnings.....	104
6509-00 Offset.....	104

## B

BOOTSTRAP.....	36, 74
----------------	--------

## C

Calc and copy time.....	87
Code sequence.....	91, 101
Cycle exceeded counter.....	87
Cycle time.....	86

## D

Diagnostic data.....	106
----------------------	-----

## F

Flash memory error.....	103
-------------------------	-----

## H

Hardware error.....	106
---------------------	-----

## I

INIT.....	36, 74
-----------	--------

## M

Machine data not valid.....	103
Mapped Object 001.....	84
Mapped Object 002.....	84
Mask Upgrade Firmware.....	92
Minimum cycle time.....	87

## O

Offset value exceeded.....	89
OPERATIONAL.....	36, 74

## P

Position error.....	102
Position Warning.....	103
PRE-OPERATIONAL.....	36, 74
Preset value exceeded.....	89
Product Code.....	83

## R

Revision Number.....	83
----------------------	----

## S

SAFE-OPERATIONAL.....	36, 74
Scaling function.....	91, 101
Serial Number.....	83
Shift Time.....	86
SM event missed counter.....	87
SM MailBox Receive (SM0).....	85
SM MailBox Send (SM1).....	85
SM PDO input (SM3).....	85
SM PDO output (SM2).....	85
Speed Format exceeded.....	89
Subindex 001.....	85p.
Sync error.....	87
Sync mode.....	86
Sync modes supported.....	87

## T

Total Measuring Range exceeded.....	89
-------------------------------------	----

## U

Units per revolution exceeded.....89

**V**  
Vendor ID.....83






# 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 Lika device and 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 <b>WARNING</b> , 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 <b>NOTE</b> , 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 <b>EXAMPLE</b> when instructions for setting parameters are accompanied by examples to clarify the explanation.

# Preliminary information

This guide is designed to provide comprehensive and detailed information that the operator needs to correctly and safely install and operate the **SFAM3-05000 and SFAM4-10000 absolute draw-wire encoders equipped with EtherCAT interface**.

The cable-pulling mechanism integrates a 13 x 14 bit absolute multiturn encoder (13 bits = singleturn resolution = 8,192 cpr; 14 bits = number of revolutions = 16,384 revolutions).

For technical specifications, please [refer to the technical catalogue](#).

SFAM3-05000 / SFAM4-10000 cable-pulling encoder is aimed at speed and position measurements and controls in a variety of industrial applications through the movement of a **5,000 mm (196.85 inches) or 10,000 mm (393.7 inches)** stainless steel wire. The typical back-and-forth travel of the axis causes the wire to reel and unreel, simultaneously rotating the encoder. Consequently, the linear motion is converted into rotational movement, which is detected by the encoder coupled to the drum.

The stroke per turn is always 200 mm (7.874 inches), the maximum number of turns is 25 for SFAM3-05000 and 50 for SFAM4-10000.

**EtherCAT** is the open standard for Industrial Ethernet. Its technology enables improved performances and allows to meet efficiency and productivity requirements in any complex industrial system. Real time communication, deterministic synchronization, high speed up to 100 Mbit/s full duplex over long distances, flexible network topologies, complete diagnostics, IT integration are among the key benefits. EtherCAT encoders comply with ETG.1000 specifications and implement the CoE (CANopen over EtherCAT), FoE (File over EtherCAT for firmware update), and EoE (Ethernet over EtherCAT) protocols as well as the EtherCAT State Machine. Thus they offer full scaling, preset, code sequence, position and velocity readout, diagnostic information, etc.

To make it easier to read the text, this guide is divided into three main sections.

In the first section some general information concerning the safety, the mechanical installation and the electrical connection as well as tips for setting up and running properly and efficiently the SFAM3-05000 / SFAM4-10000 cable-actuated encoder are provided.

In the second section, entitled **EtherCAT Interface**, you can find detailed information on the EtherCAT interface. In this section the interface features and the objects implemented in the unit are fully described.

In the third section the Integrated Web Server is described.

Before installation, please read the following notes carefully.

- To prevent failures, never exceed the maximum measuring length and ensure the wire does not become tangled.
- Never release the wire freely; always guide it to wind properly to avoid risk of personal injury and/or equipment damage.
- Always keep the wire aligned.
- The stroke per turn of the draw-wire unit is 200 mm (7.874 inches).

- This device is maintenance-free. However, if the wire becomes dirty due to adverse environmental conditions, it can be cleaned with a cloth.

Detailed information and instructions for installing the encoder and handling the wire can be found in the following sections.

# 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.

<b>Acknowledge telegram (AT)</b>	Telegram, in which each Slave inserts its data.
<b>Actual value</b>	Value of a variable at a given instant.
<b>Algorithm</b>	Completely determined finite sequence of operations by which the values of the output data can be calculated from the values of the input data.
<b>Application</b>	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.
<b>Application class</b>	Configuration of a Drive Object with a set of functional objects and supported by standard telegrams.
<b>Application mode</b>	Type of application that can be requested from a PDS.
<b>Application objects</b>	Multiple object classes that manage and provide a run time exchange of messages across the network and within the network device.
<b>Application process</b>	Part of a distributed application on a network, which is located on one device and unambiguously addressed.
<b>Application relationship</b>	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.
<b>Attribute</b>	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.
<b>Axis</b>	Logical element inside an automation system (e.g. a motion control system) that represents some form of movement.
<b>Basic Slave</b>	Slave device that supports only physical addressing of data.
<b>Behaviour</b>	Indication of how an object responds to particular events.
<b>Bit</b>	Unit of information consisting of a 1 or a 0. This is the smallest data unit that can be transmitted.

<b>CANopen</b>	Application layer protocol as defined in EN 50325-4.
<b>Channel</b>	Representation of a single physical or logical management object of a Slave to control conveyance of data.
<b>CIP™</b>	Common Industrial Protocol (see IEC 61158 Type 2, IEC 61784-1 and IEC 61784-2 CPF2).
<b>Class</b>	Description of a set of objects that share the same attributes, operations, methods, relationships, and semantics.
<b>Client</b>	Object which uses the services of another (Server) object to perform a task. Initiator of a message to which a Server reacts.
<b>Clock synchronization</b>	Representation of a sequence of interactions to synchronize the clocks of all time receivers by a time Master.
<b>Commands</b>	Set of commands from the application control program to the PDS to control the behaviour of the PDS or functional elements of the PDS.
<b>Communication cycle</b>	Accumulation of all telegrams between two Master synchronization telegrams.
<b>Communication object</b>	Component that manages and provides a run time exchange of messages across the network.
<b>Connection</b>	Logical binding between two application objects within the same or different devices.
<b>Consume</b>	Act of receiving data from a provider.
<b>Consumer</b>	Node or sink receiving data from a provider.
<b>Control</b>	Purposeful action on or in a process to meet specified objectives.
<b>Control device</b>	Physical unit that contains – in a module/subassembly or device – an application program to control the PDS.
<b>Control unit</b>	Control device.
<b>Control word</b>	Two adjacent bytes inside the Master data telegram containing commands for the addressed drive.
<b>Controller</b>	Controlling device which is associated with one or more drives (axes) a host for the overall automation.
<b>Conveyance path</b>	Unidirectional flow of APDUs across an application relationship.
<b>Cycle time</b>	Time span between two consecutive cyclically recurring events.
<b>Cyclic</b>	Events which repeat in a regular and repetitive manner.
<b>Cyclic data</b>	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.
<b>Data</b>	Generic term used to refer to any information carried over a

	fieldbus.
<b>Data consistency</b>	Means for coherent transmission and access of the input-or output-data object between and within Client and Server.
<b>Data exchange</b>	Demand dependent; non cyclic transmission (service channel).
<b>Data type</b>	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.
<b>Data type object</b>	Entry in the object dictionary indicating a data type.
<b>Default gateway</b>	Device with at least two interfaces in two different IP subnets acting as router for a subnet.
<b>Device</b>	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.).
<b>Device profile</b>	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.
<b>Diagnosis information</b>	All data available at the Server for maintenance purposes.
<b>Distributed clocks</b>	Method to synchronize Slaves and maintain a global time base.
<b>DL</b>	Data-link-layer.
<b>DLPDU</b>	Data-link-protocol-data-unit.
<b>Drive Object</b>	Functional element of a Drive Unit.
<b>Drive Unit</b>	Logical device which comprises all functional elements related to one central processing unit.
<b>Error</b>	Discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition.
<b>Error class</b>	General grouping for related error definitions and corresponding error codes.
<b>Error code</b>	Identification of a specific type of error within an error class.
<b>EtherCAT State Machine</b>	EtherCAT Slave is a state machine; communication and operating characteristics depend on the current state of the

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

<b>Master</b>	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.
<b>Master data telegram (MDT)</b>	Telegram, in which the Master inserts its data.
<b>Medium</b>	Cable, optical fibre or other means by which communication signals are transmitted between two or more points.
<b>Message</b>	Ordered series of octets intended to convey information. Normally used to convey information between peers at the application layer.
<b>Model</b>	Mathematical or physical representation of a system or a process, based with sufficient precision upon known laws, identification or specified suppositions.
<b>Motion</b>	Any aspect of the dynamics of an axis.
<b>Motion Axis Object</b>	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.
<b>Network</b>	Set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways.
<b>Node</b>	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].
<b>Object</b>	Abstract representation of a particular component within a device. An object can be: <ol style="list-style-type: none"> <li>1. an abstract representation of the capabilities of a device. Objects can be composed of any or all of the following components: <ul style="list-style-type: none"> <li>◦ data (information which changes with time);</li> <li>◦ configuration (parameters for behaviour);</li> <li>◦ methods (things that can be done using data and configuration);</li> </ul> </li> <li>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.</li> </ol>
<b>Object dictionary</b>	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.
<b>Operating cycle</b>	Period of the control loop within the drive or the control unit.
<b>Operating mode</b>	Characterization of the way and the extent to which the human operator intervenes in the control equipment.



<b>Output data</b>	Data originating in a device, resource or functional element and transferred from them to external systems.
<b>P-Device</b>	Field device and the host for the Drive Objects.
<b>Parameter</b>	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.
<b>PDO</b>	Process Data Object.
<b>PDS</b>	Power Drive System.
<b>Process data</b>	Collection of application objects designated to be transferred cyclically or acyclically for the purpose of measurement and control.
<b>Process Data Object (PDO)</b>	Communication object with real-time capability. Structure described by mapping parameters containing one or several process data entities.
<b>Producer</b>	Node or source sending data to one or many consumers.
<b>Profile</b>	Representation of a PDS interface in terms of its parameters, parameter assemblies and behaviour according to a communication profile and a device profile.
<b>Protocol</b>	Convention about the data formats, time sequences, and error correction in the data exchange of communication systems.
<b>Reference variable</b>	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.
<b>Resource</b>	Processing or information capability.
<b>Segment</b>	Collection of one real Master with one or more Slaves.
<b>Server</b>	Object which provides services to another (Client) object.
<b>Service</b>	Operation or function than an object and/or object class performs upon request from another object and/or object class.
<b>Service data</b>	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.
<b>Set-point</b>	Value or variable used as output data of the application control program to control the PDS.
<b>Slave</b>	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.
<b>Standard telegram</b>	Set of input data and output data for an application mode.
<b>Status</b>	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.
<b>Status word</b>	Two adjacent bytes inside the drive telegram containing status information.

<b>Subindex</b>	Sub-address of an object within the object dictionary.
<b>Supervisor</b>	Engineering device which manages provisions of configuration data (parameter sets) and collections of diagnosis data from P-Devices and/or controllers.
<b>Switch</b>	MAC bridge as defined in IEEE 802.1D.
<b>Sync Manager</b>	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.
<b>Sync manager channel</b>	Single control elements to coordinate access to concurrently used objects.
<b>Synchronised</b>	Condition where the local clock value on the drive is locked onto the Master clock of the distributed System Time.
<b>Synchronous with DC SYNC0</b>	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.
<b>Synchronous with SM3</b>	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.
<b>System Time</b>	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.
<b>Telegram</b>	Message.
<b>Time stamp</b>	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.
<b>Topology</b>	Physical network architecture with respect to the connection between the stations of the communication system.
<b>Type</b>	Hardware or software element which specifies the common attributes shared by all instances of the type.
<b>Use case</b>	Class specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.
<b>Variable</b>	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 encoder's power supply before connecting the device.
- Connect according to the explanation in the "4 – Electrical connections" section on page 30.
- 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. Ensure 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 stationary mechanical parts.
- Do not disassemble the unit, unless otherwise indicated.
- Do not try to open the device. The stored energy of the spring drive may lead to injuries when being mishandled.
- Do not tool the unit.
- Delicate electronic equipment: handle with care; do not subject the device to knocks or shocks.
- Respect the environmental characteristics of the product.
- We suggest installing the unit providing protection means against waste, especially swarf as turnings, chips, or filings; should this not be possible, please ensure that adequate cleaning measures are in place in order to prevent the wire from jamming.
- To prevent failures, never exceed the maximum measuring length and ensure that the wire does not become tangled.
- Do not touch the wire while operating the unit.
- Whenever possible, if the unit is operated in a humid environment, install it with the wire outlet downwards to prevent water from gathering inside the housing.
- Never release the wire freely; always guide it to wind properly to avoid risk of personal injury and/or equipment damage.
- Always keep the wire aligned not to damage the equipment.
- The stroke per turn of the draw-wire unit is 200 mm (7.874 inches).

## 2 - Identification

The device can be identified by the **order code** and the **serial number** printed on the label applied to its housing. This information is also listed in the delivery document. Please always provide the order code and the serial number when contacting Lika Electronic for purchasing spare parts or requesting assistance. For detailed information on the product's technical characteristics, refer to the technical catalogue.



**Warning:** encoders whose order code ends with "/Sxxx" may have mechanical and electrical characteristics different from standard and be supplied with additional documentation for special connections (Technical info).



### WARNING

Please note that since EtherCAT is not based on TCP/IP, there is no need to administer MAC addresses or IP addresses. However, you need the IP Address to reach, for instance, the web server. To know the IP address assigned by Beckhoff, see on page 114.



## 3 - Mechanical installation



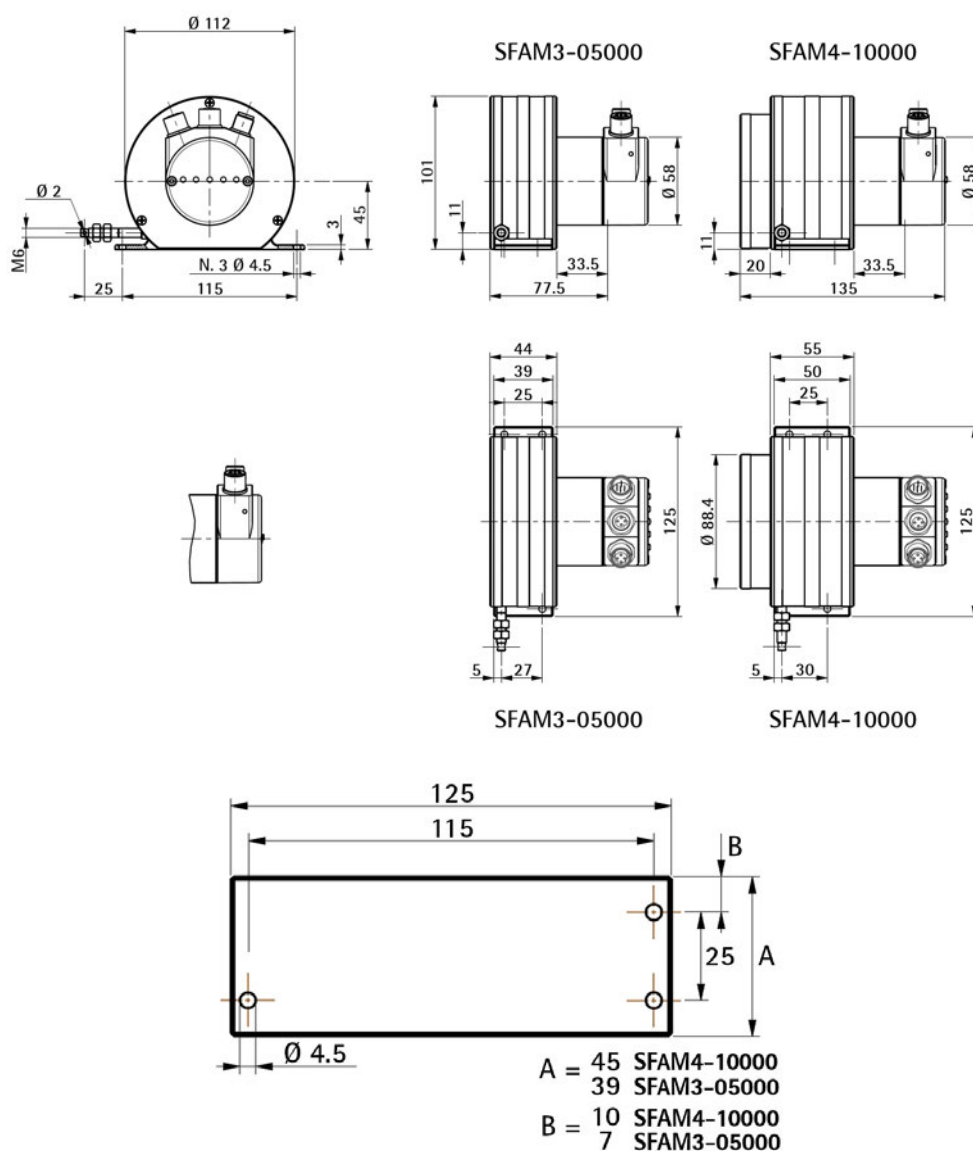
### WARNING

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.

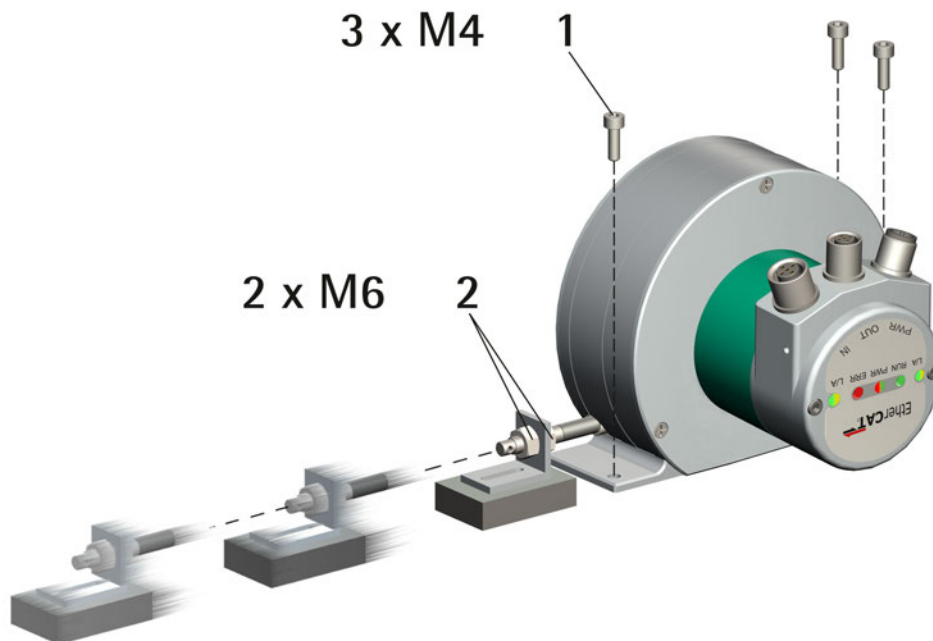
For any information on the mechanical data and the electrical characteristics of the encoder please refer to the technical catalogue.

### 3.1 Overall dimensions

Values are expressed in millimetres (mm)



### 3.2 Mounting instructions



- Mount the draw-wire unit before extracting the wire and attaching it to the measuring target.
- Secure the draw-wire unit to a fixed support using **three M4 screws 1**. Recommended tightening torque: 2.5 Nm.
- Once the encoder assembly is fully mounted, remove the transport safety wire that secures the end of the measuring wire.
- Carefully extract the measuring wire as needed and secure its end to the measuring target using the provided **M6 nuts 2**.



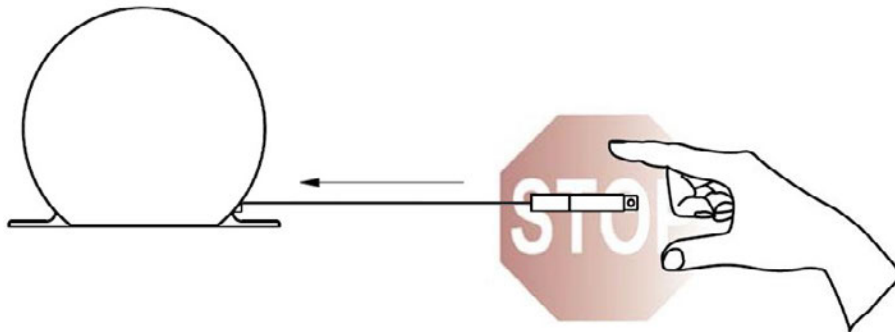
#### WARNING

We suggest installing the unit providing protection means against waste, especially swarf as turnings, chips, or filings; should this not be possible, please ensure that adequate cleaning measures are in place in order to prevent the wire from jamming.

To prevent failures, never exceed the maximum measuring length and ensure the wire does not become tangled.

#### 3.2.1 Handling the wire

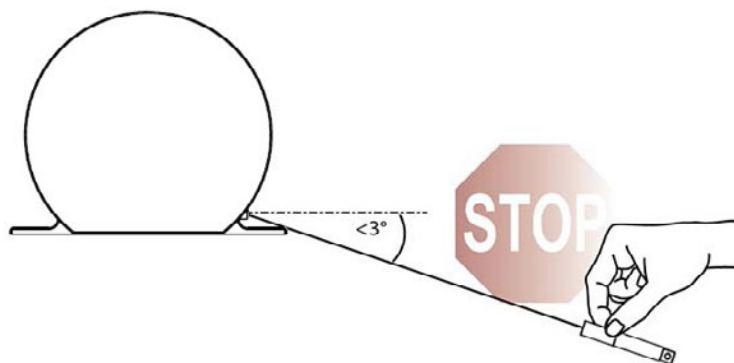
- When installing or operating the draw-wire encoder, take care not to allow the wire to snap back accidentally or sharply; or to pull the wire beyond the specified measurement range, as this may damage the unit.



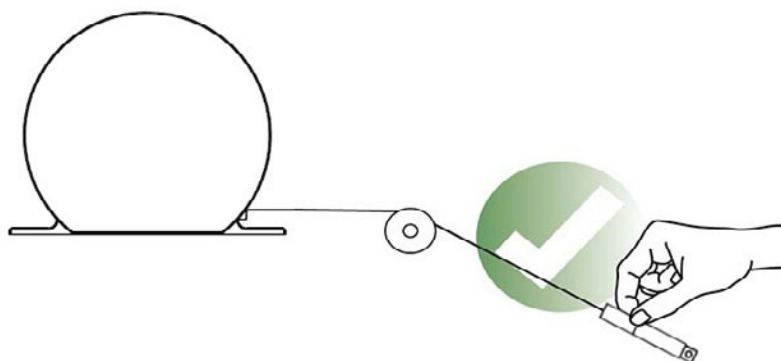
- The wire must be pulled out in a straight line from the unit.



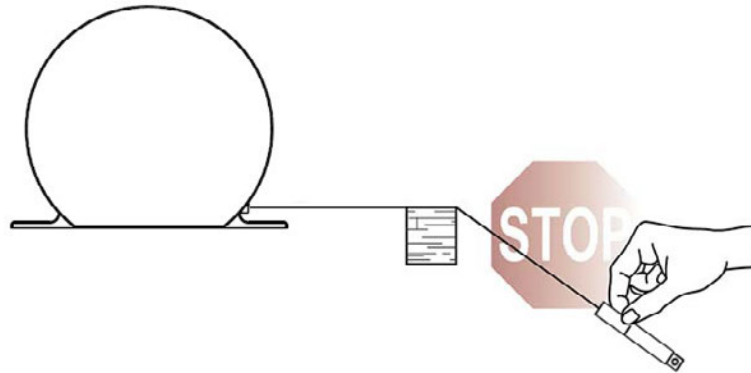
- The maximum deviation from the straight line is 3°. Avoid pulling the wire at an angle, as this will significantly reduce the device's durability.



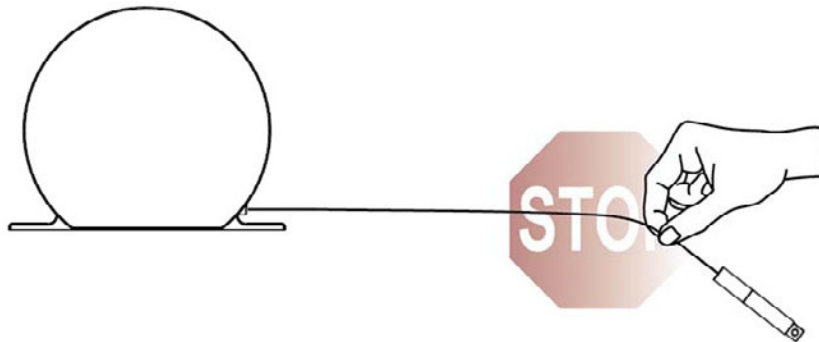
- If it is not possible to maintain the limit of 3°, a deflection pulley must be used.



- Avoid guiding the wire over edges or corners; instead, use a deflection pulley.



- Do not pull directly on the wire, as this may cause it to buckle. Do not operate the mechanical assembly if the wire is buckled or damaged. Tearing the wire could result in injury or damage to the unit.



- Guide the wire preferably along corners or within protected channels to prevent accidental contact or contamination.

### 3.3 Useful information

If you want to know the **maximum measuring length** and the **physical linear resolution** of the draw-wire encoder, please refer to the order code. The stroke per turn is always 200 mm (7.874 inches), the maximum number of turns is 25 for SFAM3-05000 and 50 for SFAM4-10000.



#### EXAMPLE 1

SFAM3-05000-EC2-08192-RM12 using the physical resolution (Scaling function bit 2 in **6000-00 Operating parameters** = 0)

Stroke per turn of the drum = 200 mm (7.874 inches)

Physical resolution per turn (**6501-00 Hardware counts per revolution**) = 13 bits = 8,192 cpr

Max. number of physical revolutions (**6502-00 Hardware number of turns**) = 16,384

Total physical resolution = 27 bits = 134,217,728 counts  
 Physical linear resolution = 0.024 mm = 24 µm (200 mm / 8,192 cpr)  
 Max. number of turns of the drum = 25  
 Max. measuring length = 5,000 mm (196.85 inches)  
 Max. number of information = 208,300 counts


**EXAMPLE 2**

SFAM4-10000-EC2-08192-RM12 using a custom resolution (**Scaling function** bit 2 in **6000-00 Operating parameters** = 1)

Stroke per turn of the drum = 200 mm (7.874 inches)

Physical resolution per turn = 13 bits = 8,192 cpr

Max. number of physical revolutions = 16,384

Custom resolution per turn = **6001-00 Units per revolution** = 2,000 cpr (example)

**6002-00 Total measuring range** = 8,192,000 information (example)

Custom number of encoder revolutions = 
$$\frac{\text{6002-00 Total measuring range}}{\text{6001-00 Units per revolution}} = 4,096$$

Custom linear resolution = 0.1 mm = 100 µm (200 mm / 2,000 cpr)

Max. number of turns of the drum = 50

Max. measuring length = 10,000 mm (393.7")

Max. number of information = 100,000 counts

**3.4 Maintenance**

The measuring system is maintenance-free and does not need any particular maintenance; anyway it has to be handled with the utmost care as any delicate electronic equipment. From time to time we recommend the following operations:

- the unit and the wire must to be cleaned regularly using a soft and clean cloth to remove dust, chips, moisture etc.; do not use oil to clean the wire. If they become dirty due to adverse environmental conditions, clean them as often as required.

## 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.  
Do not remove the connection cap of the encoder. Damage may be caused to internal components.



No user serviceable parts inside the encoder!

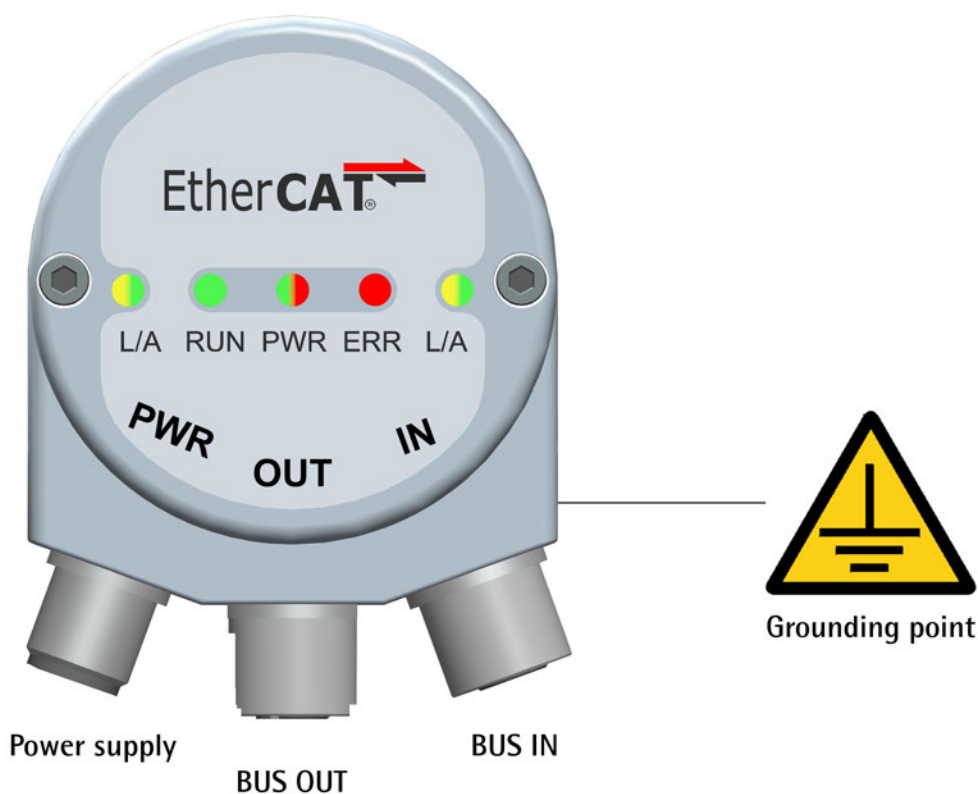


Figure 1 - Connectors and diagnostic LEDs

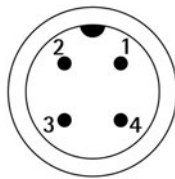
#### 4.1 M12 connectors

The connection cap is equipped with three M12 connectors with pin-out in compliance with the EtherCAT® standard. Therefore you can use standard EtherCAT cables commercially available.

Please note that input ECATIN (BUS IN) and output ECATOUT (BUS OUT) connectors are not interchangeable!

##### 4.1.1 PWR Power supply connector (Figure 1)

M12 4-pin male connector with A coding is used for power supply.



Description	Pin
+5Vdc +30Vdc	1
n.c.	2
0Vdc	3
n.c.	4

n.c. = not connected

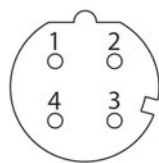


#### WARNING

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

##### 4.1.2 BUS IN and BUS OUT connectors (Figure 1)

Two M12 4-pin female connectors with D coding are used for Ethernet connection through BUS IN and BUS OUT ports.



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

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.



#### WARNING

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

## 4.2 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).

## 4.3 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 70.



### WARNING

Please note that since EtherCAT is not based on TCP/IP, there is no need to administer MAC addresses or IP addresses. However, you need the IP Address to reach, for instance, the web server. To know the IP address assigned by Beckhoff, see on page 114.



#### 4.4 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, please refer to the "7.1.4 Line Termination" section on page 69.

#### 4.5 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 1, use one TCEI UNI M3 x 6 cylindrical head screw with two tooth lock washers).

#### 4.6 Diagnostic LEDs (Figure 1)

Five LEDs located in the rear side of the connection cap (see Figure 1) 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
<b>ON</b>	The indicator shall be constantly ON.
<b>OFF</b>	The indicator shall be constantly OFF.
<b>Flickering</b>	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.
<b>Blinking</b>	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.
<b>Single flash</b>	The indicator shall show one short flash (200 ms) followed by a long OFF phase (1,000 ms).
<b>Double flash</b>	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 encoder 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.

**RUN LED (green)**

It shows the current communication state of the EtherCAT encoder.

ERR LED	Description
OFF	The encoder is in <b>INIT</b> state.
BLINKING (2.5 Hz) green	The encoder is in <b>PRE-OPERATIONAL</b> state.
SINGLE FLASH green	The encoder is in <b>SAFE-OPERATIONAL</b> state.
ON green	The encoder is in <b>OPERATIONAL</b> state.

**PWR Power LED (green / red)**

It shows the power supply and system state.

PWR LED	Description	Meaning
OFF	Power OFF	The encoder power supply is switched OFF. No supply voltage for the device or hardware fault.
ON green	Power ON	The encoder power supply is switched ON.

		The firmware is running.
<b>BLINKING red</b>	No firmware program installed, firmware update mode	At power ON the LED blinks red at 1 Hz. The firmware program is not installed, the encoder 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 61; or to the "8.6 Firmware update" section on page 122.

### ERR LED (red)

It shows the current error state of the EtherCAT encoder.

ERR LED	Description
<b>OFF</b>	No error. The EtherCAT communication of the encoder is in working condition.
<b>BLINKING (2.5 Hz) red</b>	Invalid configuration, general configuration error. Possible reason: state change commanded by the Master is not possible due to register or object settings.
<b>SINGLE FLASH red</b>	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 encoder enters the <b>SAFE-OPERATIONAL</b> state automatically.
<b>DOUBLE FLASH red</b>	Application watchdog timeout. An application watchdog timeout has occurred. Possible reason: Sync Manager Watchdog timeout.

### 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
<b>OFF</b>	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.
<b>ON green</b>	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 encoder

		does not send/receive Ethernet frames through port BUS IN.
<b>FLICKERING</b> <b>yellow</b>	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.

#### 4.7 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.

## 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 encoder 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 40 ff);
- or by means of the Integrated Web Server (see the "8 – Integrated Web Server" section on page 113).

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 electrical connection and switch on the +5Vdc +30Vdc power supply, see on page 30 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 30 ff; check the soundness of the connection;
- you do not need to set the node address and the transmission rate (see on page 70);
- you do not need to set any line termination (see on page 69);
- in the software tool install the XML file corresponding to the encoder to be installed, see on page 42 ff; the XML is the same for all models, it is:
  - **Lika\_SFAMx\_EC\_vx\_x.xml**: it is intended for installation of **all SFAM3-SFAM4 draw-wire encoders** ("SFAMx" is the general identification of the encoder series; "EC" is the Lika code that identifies the EtherCAT protocol; "vx\_x" is the version of the XML file);
- if you want to use the physical resolution (see the **6501-00 Hardware counts per revolution** object and the **6502-00 Hardware number of turns** object), please check that the **Scaling function** parameter is disabled (the bit 2 in the **6000-00 Operating parameters** object = 0; see on page 91);
- 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 91) and then set the resolution you need for your application next to the **6001-00 Units per revolution** and **6002-00 Total measuring range** objects (see on page 92);

- if needed, you can enter the Preset value next to the **6003-00 Preset Value** object and then set it in the desired position; see on page 97;
- save the new setting values (use the **1010-01 Store parameters** object; see on page 82).

### 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 91); in this case, the device uses the physical resolution (see the **6501-00 Hardware counts per revolution** and **6502-00 Hardware number of turns** objects) to arrange the absolute position value.

You can also use the Integrated Web Server, see the "8.4 Setting the objects" section on page 119; or a software tool, see the "6.7 CoE Object Dictionary" section on page 59.

- 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:
  - set the singleturn resolution next to the **6001-00 Units per revolution** object, see on page 92;
  - set the total resolution next to the **6002-00 Total measuring range** object, see on page 94.

You can also use the Integrated Web Server, see the "8.4 Setting the objects" section on page 119; or a software tool, see the "6.7 CoE Object Dictionary" section on page 59.

### 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** object on page 99; the **6004-00 Position Value** object is mapped in the **1A00-00 TxPDO mapping parameter** object, sub-index 001 **01 Mapped Object 001**, see on page 84.
- Open the Integrated Web Server, see the "8.3 Encoder position and speed" section on page 117; see the "8.5 Encoder information (EtherCAT objects)" section on page 121.
- Use the software tool, see the "6.6 Process Data Objects" section on page 57.

### 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 88; 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 84.
- Open the Integrated Web Server, see the "8.3 Encoder position and speed" section on page 117; see the "8.5 Encoder information (EtherCAT objects)" section on page 121.
- Use the software tool, see the "6.6 Process Data Objects" section on page 57.

### 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** object, see on page 97; the preset value is activated as soon as the value is confirmed.
- Open the **Set Encoder Objects** page in the Integrated Web Server, see the "8.4 Setting the objects" section on page 119.
- Use the software tool, see the "6.7 CoE Object Dictionary" section on page 59.

### 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 82.
- Use the **Save Param.** function in the **Set Encoder Objects** page of the Integrated Web Server, see the "8.4 Setting the objects" section on page 119.

### 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 82.
- Use the **Load Default** function in the **Set Encoder Objects** page of the Integrated Web Server, see the "8.4 Setting the objects" section on page 119.

## 6 – Quick reference with TwinCAT

Lika encoders are Slave devices and support "CANopen over EtherCAT" (CoE) mode for data transfer. In particular, they support the "CANopen DS 301 Communication profile".

For any omitted specification on EtherCAT® protocol, please refer to "ETG.1000 EtherCAT Specification" documents available at the address [www.ethercat.org](http://www.ethercat.org).

For any omitted specification on CANopen® protocol, please refer to "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](http://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 encoder 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](http://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.

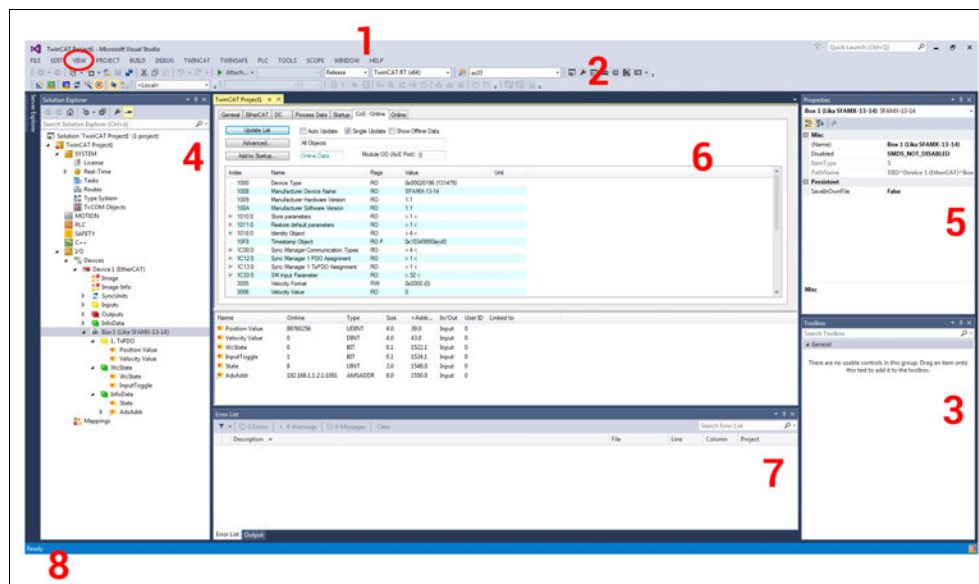


Figure 2 - User interface components

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 (Lika\_SFAMx\_EC\_vx\_x.xml) can be downloaded from Lika Electronic web site. 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...**

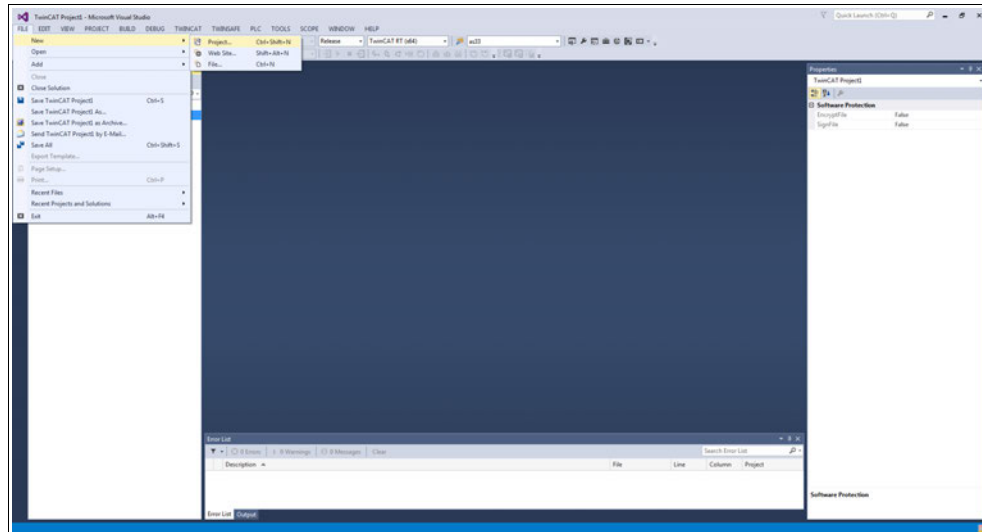


Figure 3 – Creating a new project

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.

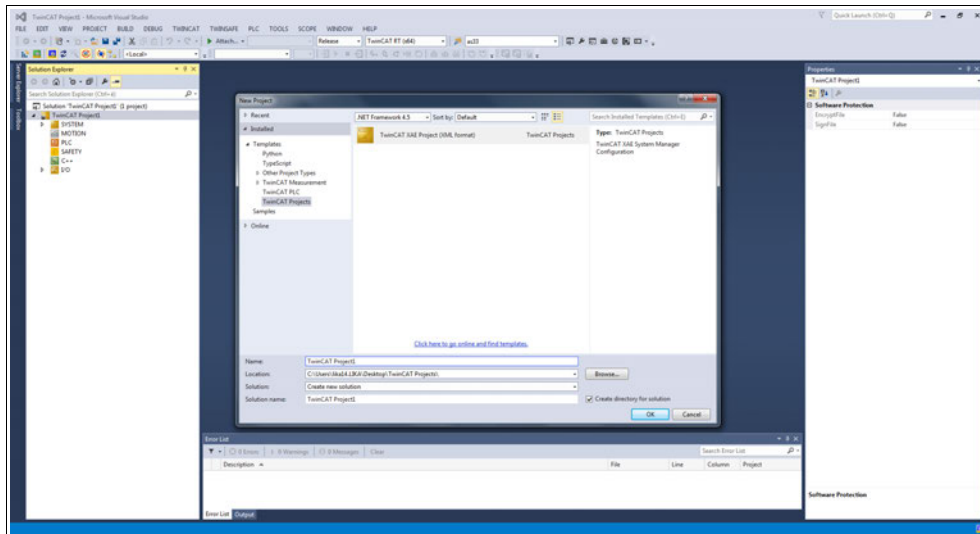


Figure 4 - Name and location of the project

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

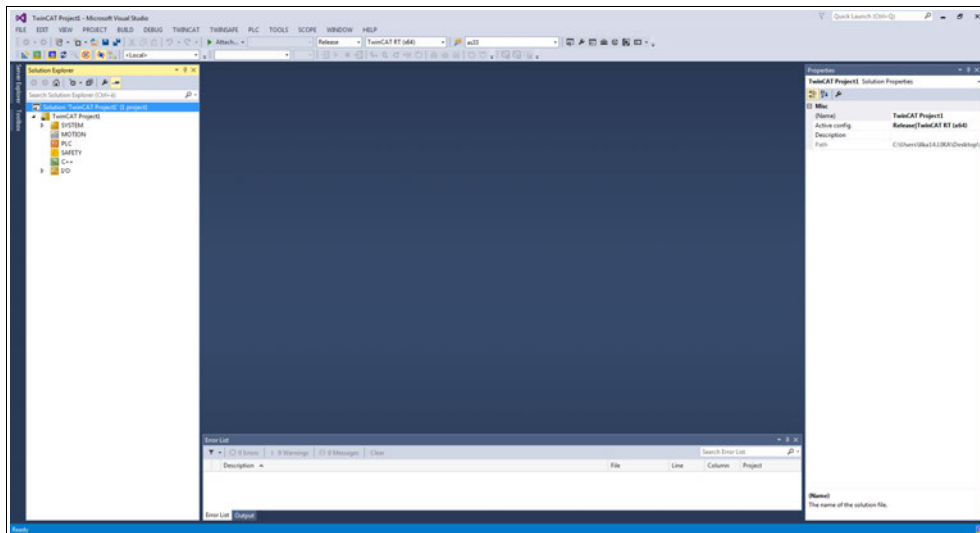


Figure 5 - New TwinCAT project created

## 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.

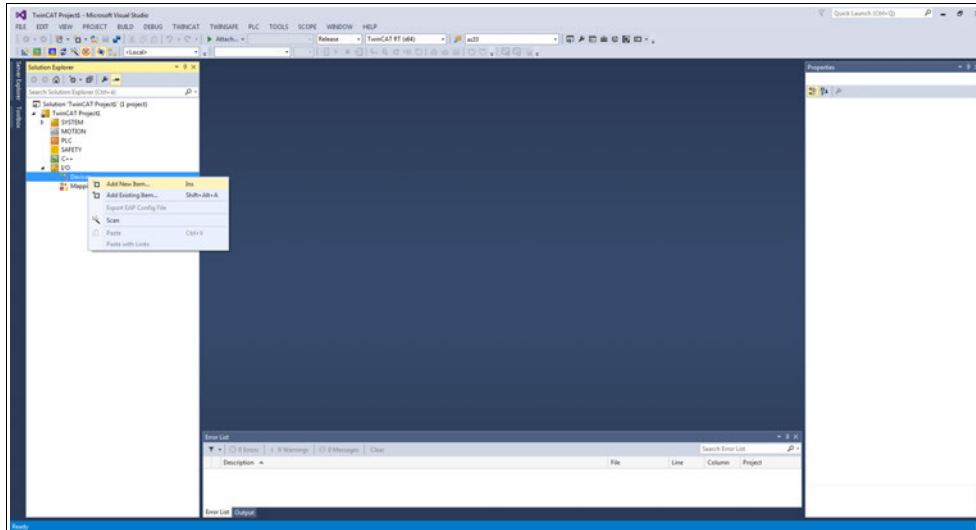


Figure 6 - Adding a new I/O device

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.

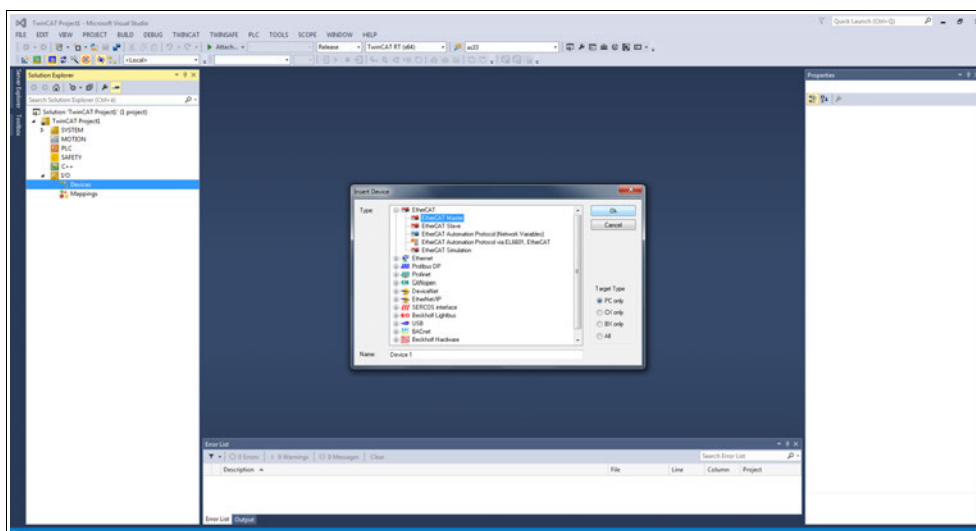


Figure 7 - Adding the EtherCAT Master

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.

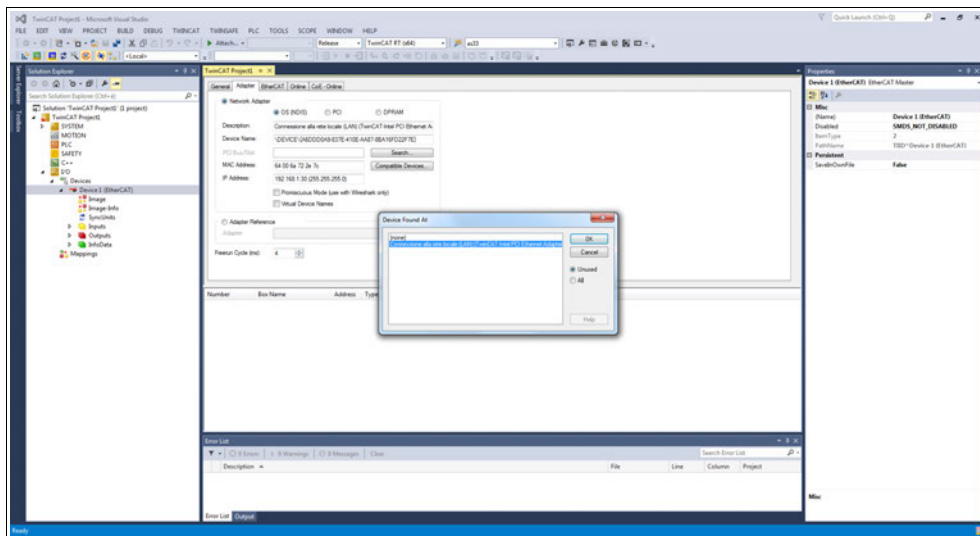


Figure 8 - Selecting the network card

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, on the menu bar, select the **TWINCAT** menu and then click **Show Real Time Ethernet Compatible Devices...**

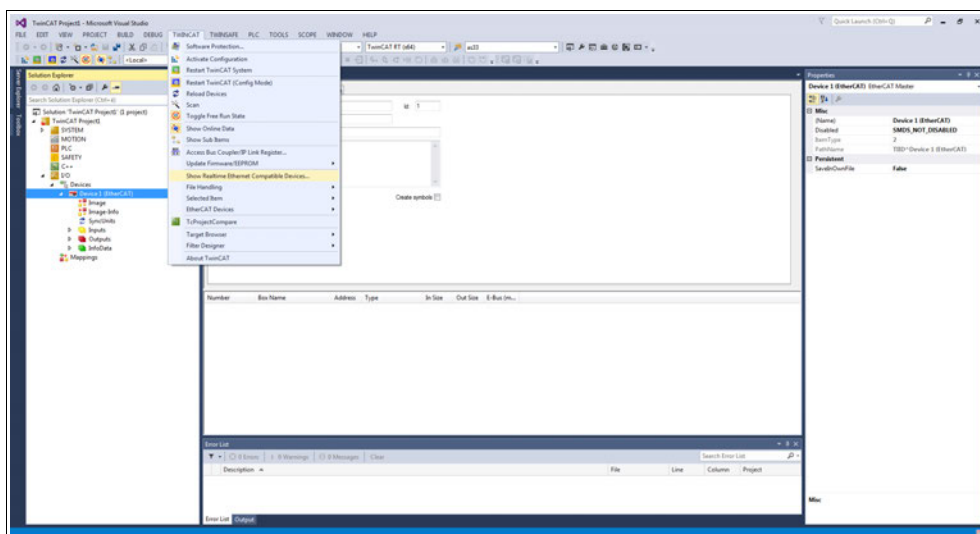


Figure 9 - Real Time Ethernet Compatible Devices

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

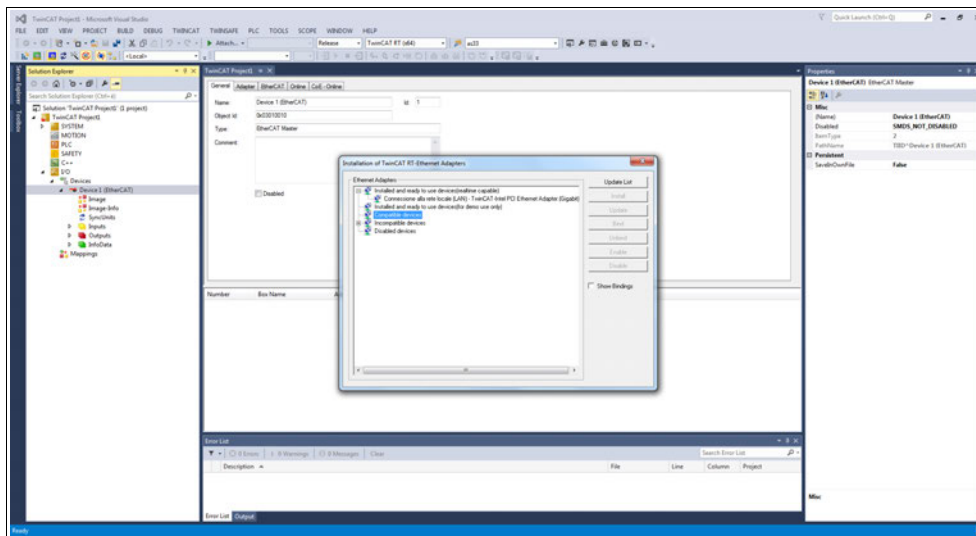


Figure 10 – Selecting the network card

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 will scan the PC for lower-level devices.

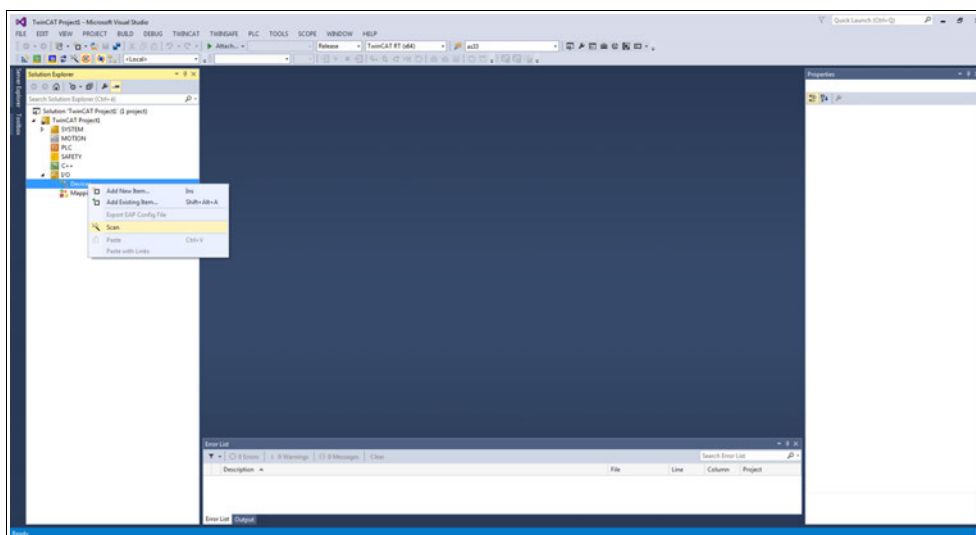


Figure 11 – Scanning the network for connected devices

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.

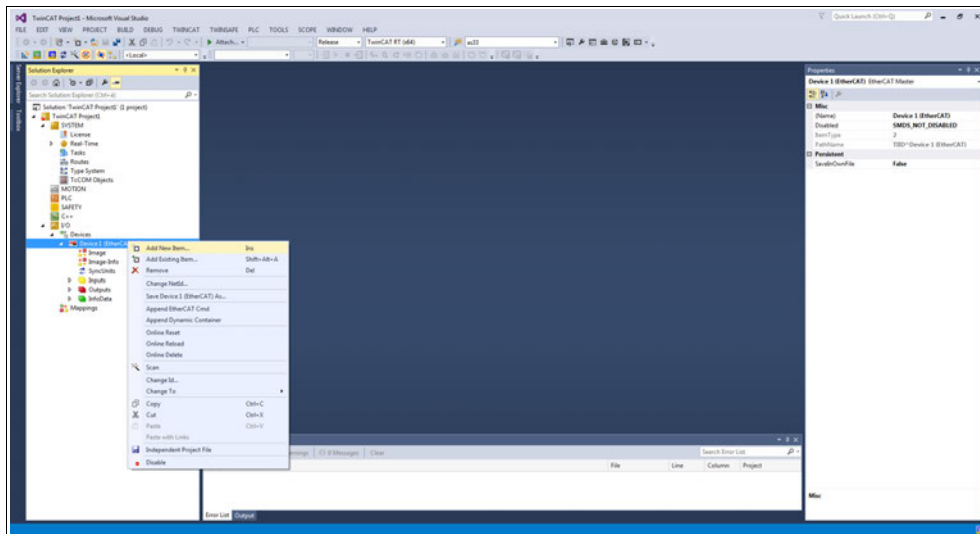


Figure 12 - Adding a new I/O module

The **Insert EtherCAT device** dialog box will be displayed.

Scroll to the **Lika Electronic srl** directory, expand the list and select the encoder you want to install (SFAMX-13-14, i.e. SFAM3 or SFAM4 27-bit multiturn draw-wire encoder in the example). Click **OK** to confirm the choice.

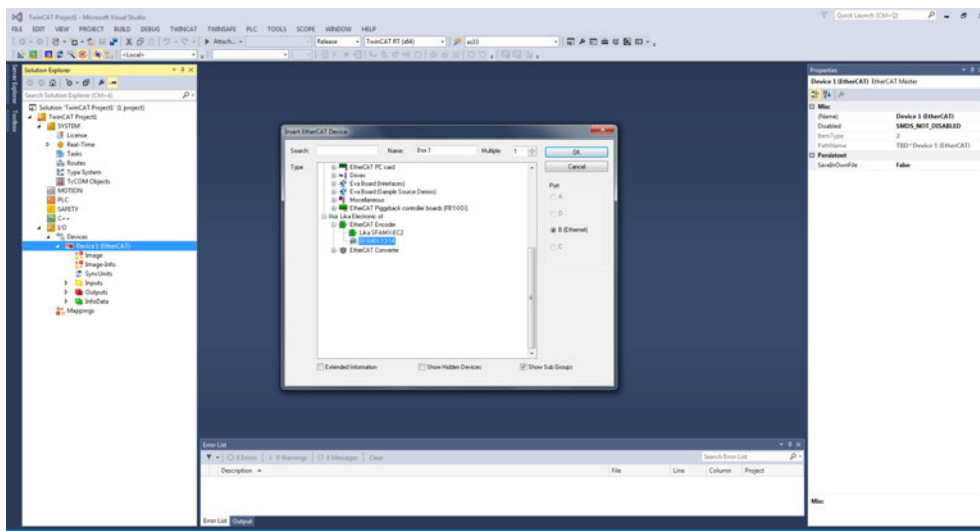


Figure 13 - Selecting the encoder



The **Box 1** device will be listed under **Device 1 (EtherCAT)**, Lika SFAMX-13-14 in the example.

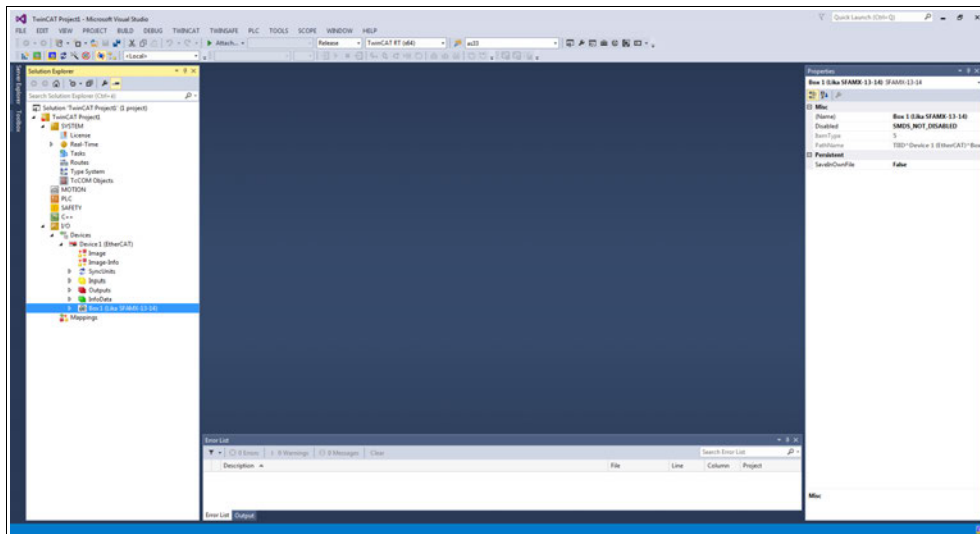


Figure 14 – Encoder installed

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

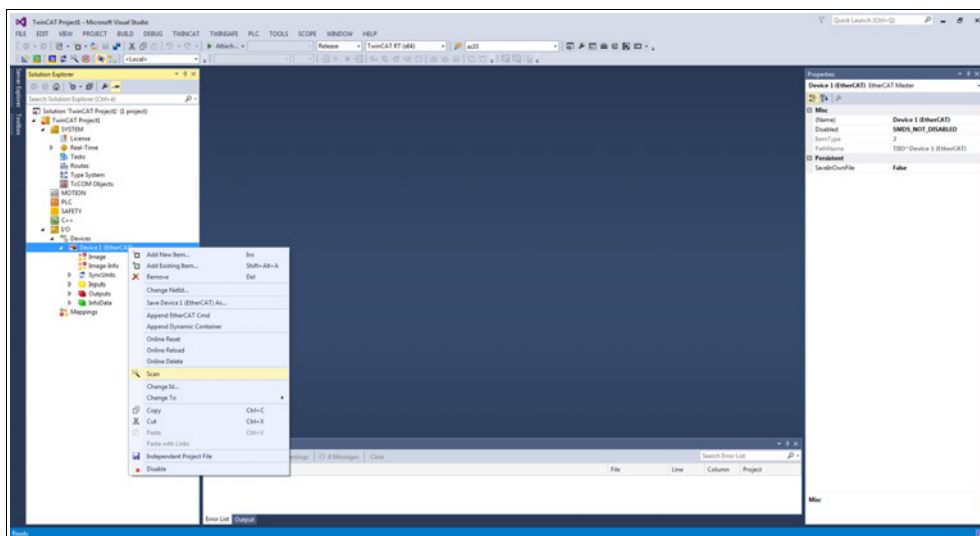


Figure 15 – Scanning the network for connected devices

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

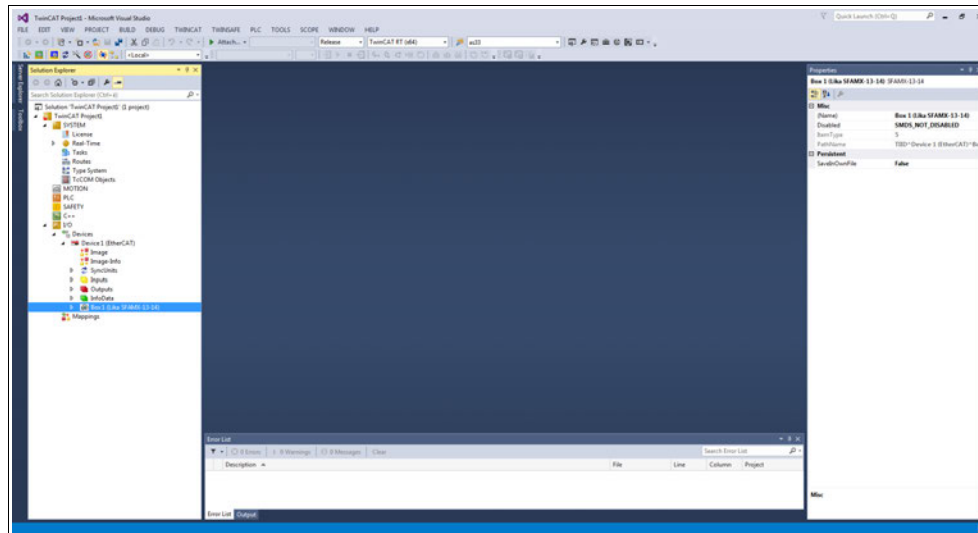


Figure 16 – Found boxes



### WARNING

The correct XML file supplied with the encoder must be available inside the TwinCAT \ 3.1 \ Config \ Io \ EtherCAT directory: **Lika\_SFAMx\_EC\_vx\_x.xml** (see at [www.lika.biz](http://www.lika.biz)).

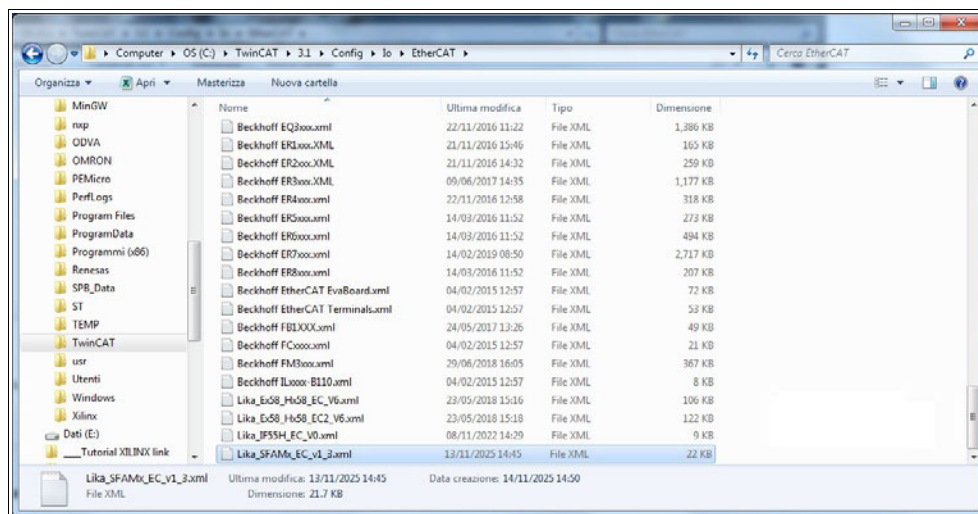


Figure 17 – XML files folder

## 6.4.4 Tabbed pages

Double click the **Box 1 (Lika SFAMX-13-14)** item, the tabbed pages that describe the encoder 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 encoder such as the name of the EtherCAT Slave device, the identification number, the TwinCAT object identification number, the EtherCAT device type, etc.

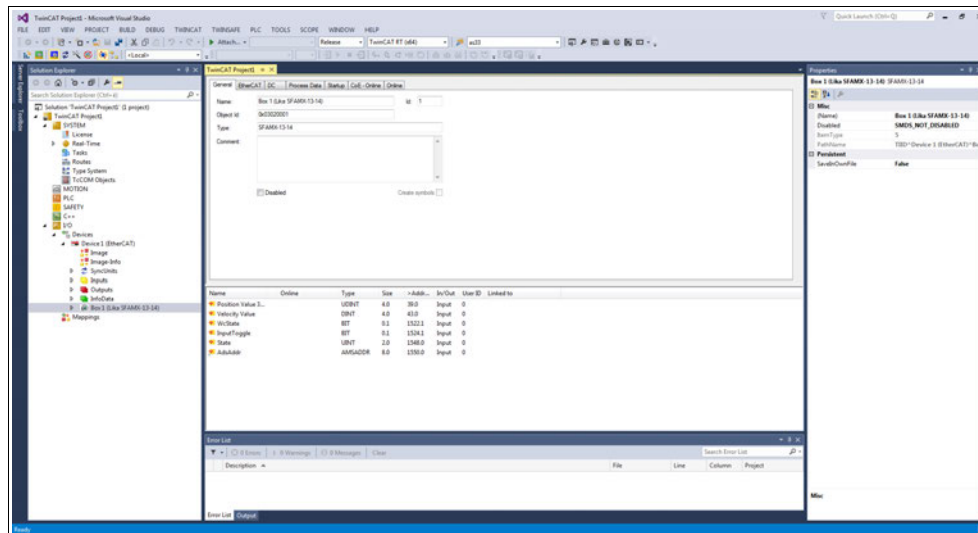


Figure 18 – General tabbed page

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.

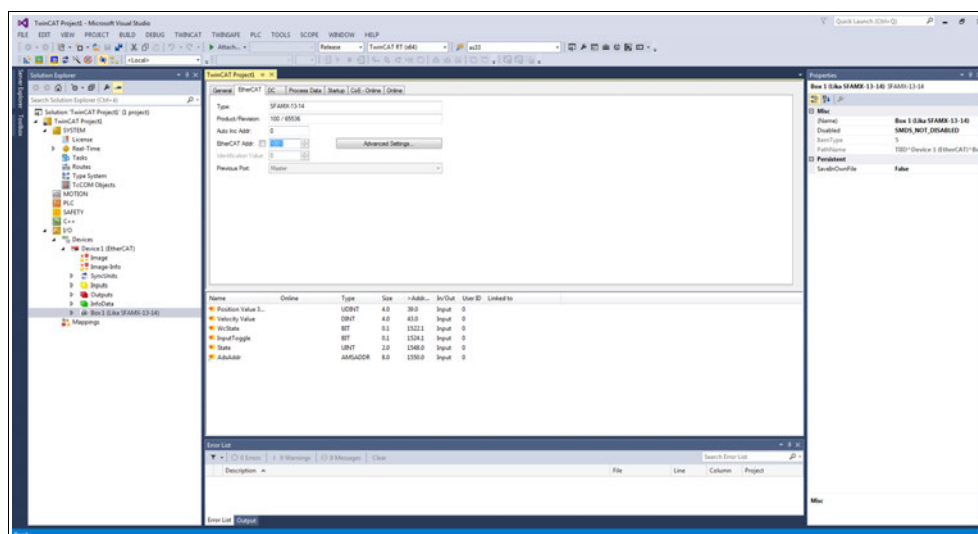


Figure 19 – EtherCAT tabbed page

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 55.

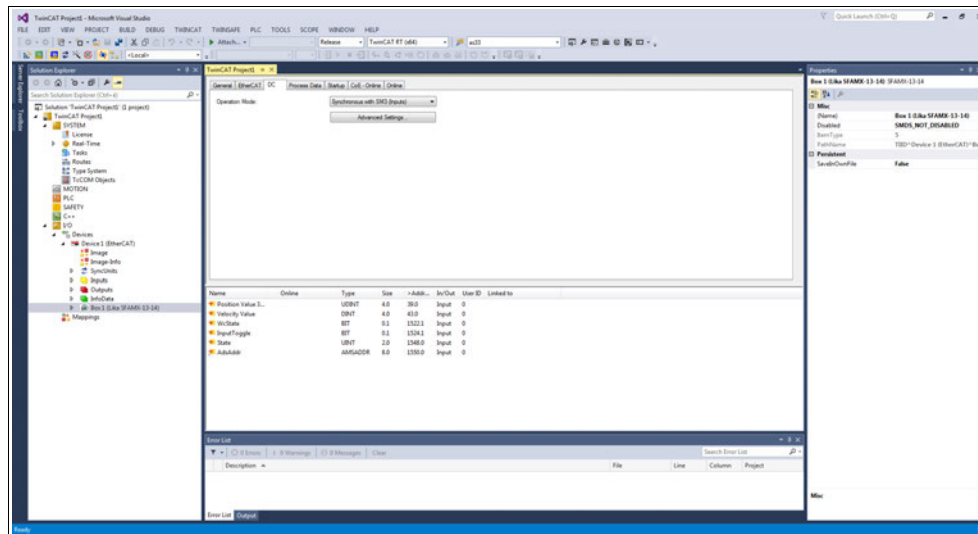


Figure 20 - DC tabbed page

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 57.

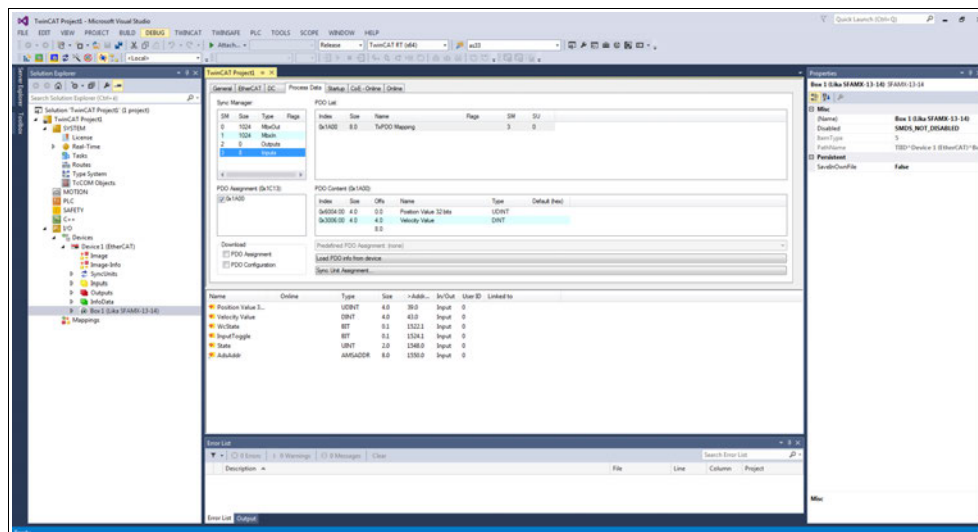


Figure 21 - Process Data tabbed page

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.

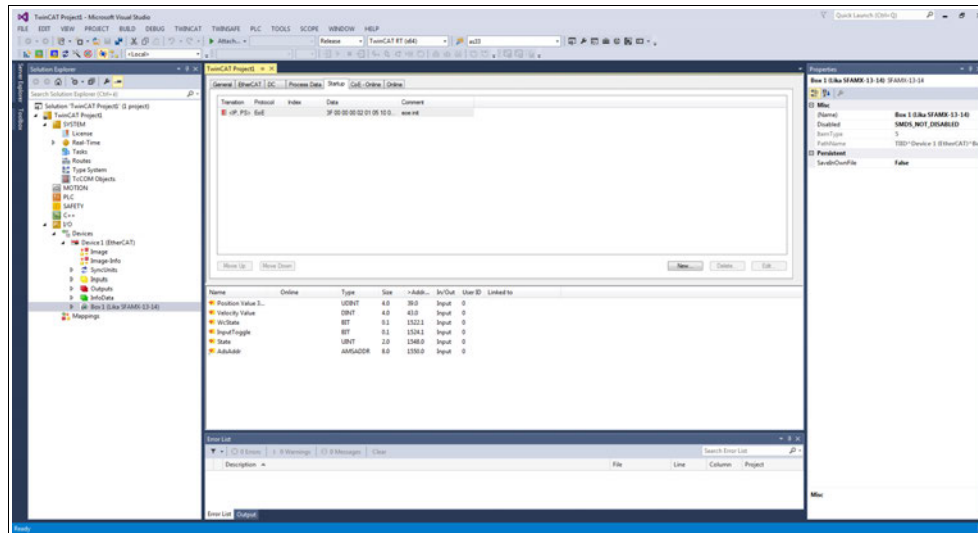


Figure 22 - Startup tabbed page

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 59.

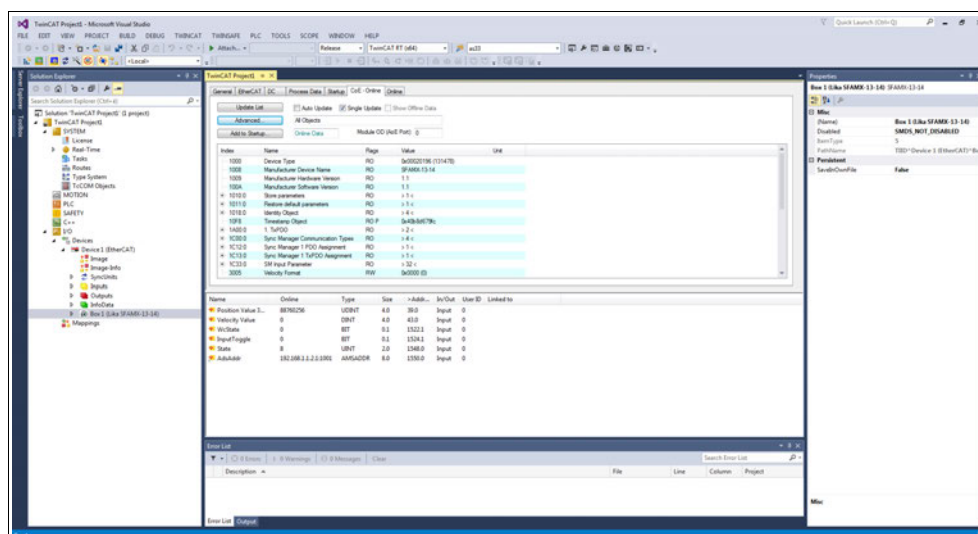


Figure 23 - CoE - Online tabbed page

The **Online** tabbed page allows the user to check and set the state of the encoder and also to update the firmware by using the File Access over EtherCAT protocol. Refer also to the "6.8 Online Data" section on page 60; and to the "6.9 Firmware update" section on page 61.

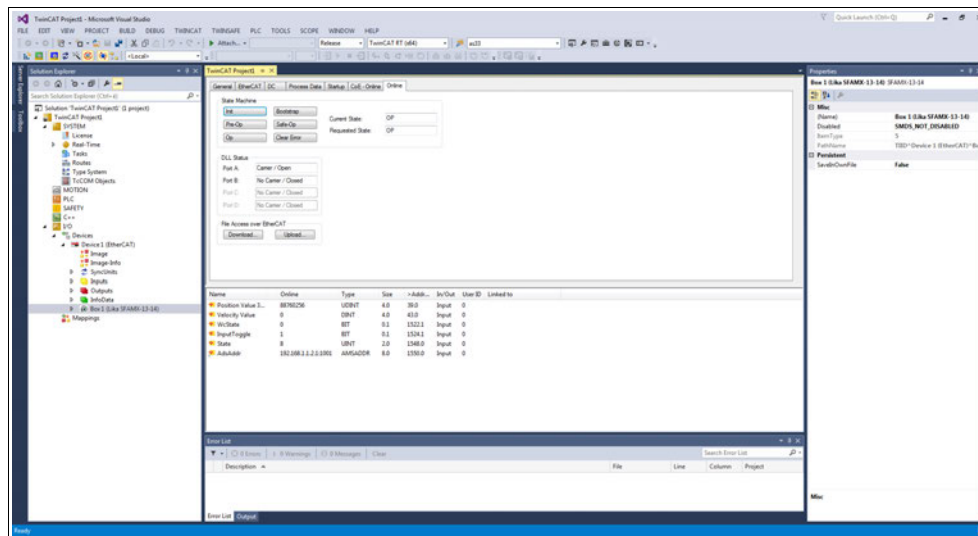


Figure 24 - Online tabbed page

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 objects **1009 Manufacturer Hardware Version (1009-00 Manufacturer Hardware Version)** and **100A Manufacturer Software Version (100A-00 Manufacturer Software Version)**.

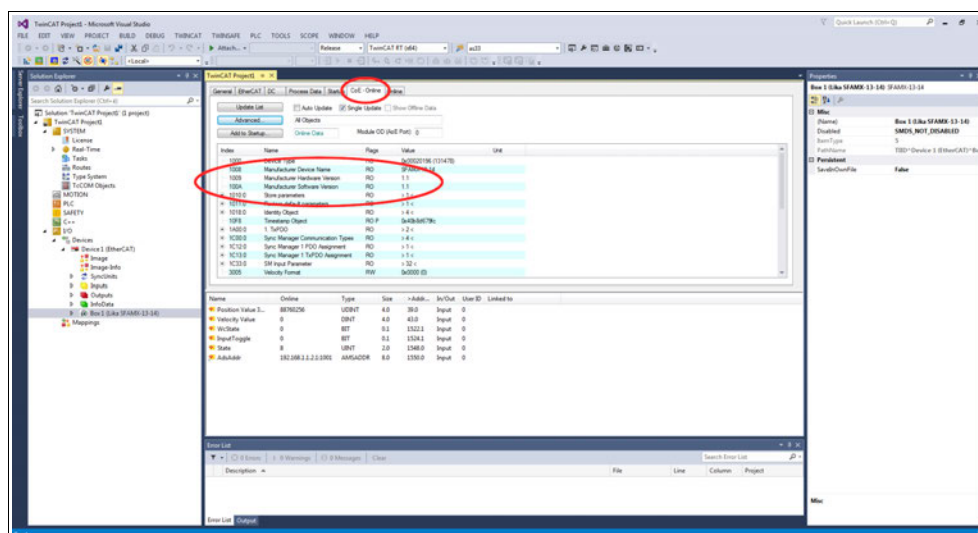


Figure 25 - Hardware and software versions

## 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.

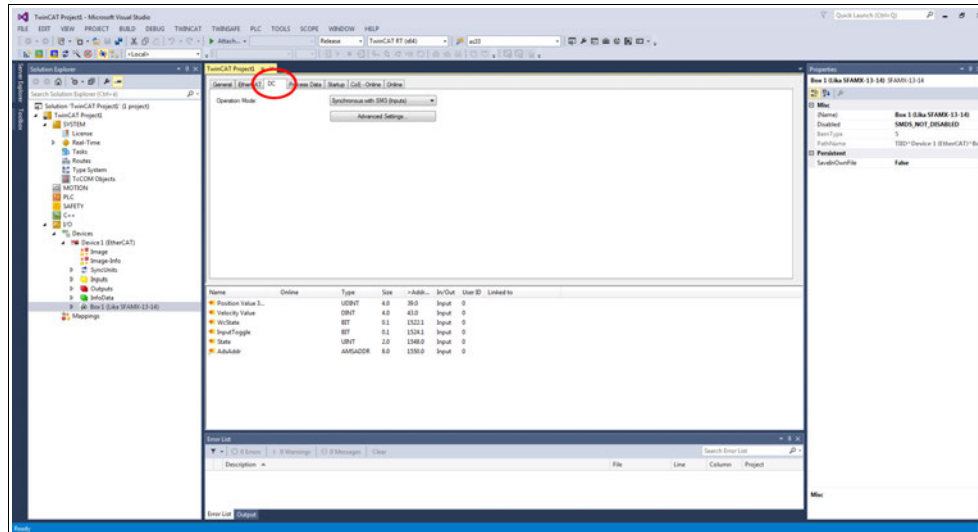


Figure 26 - DC tabbed page

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 (SYNCO)**.

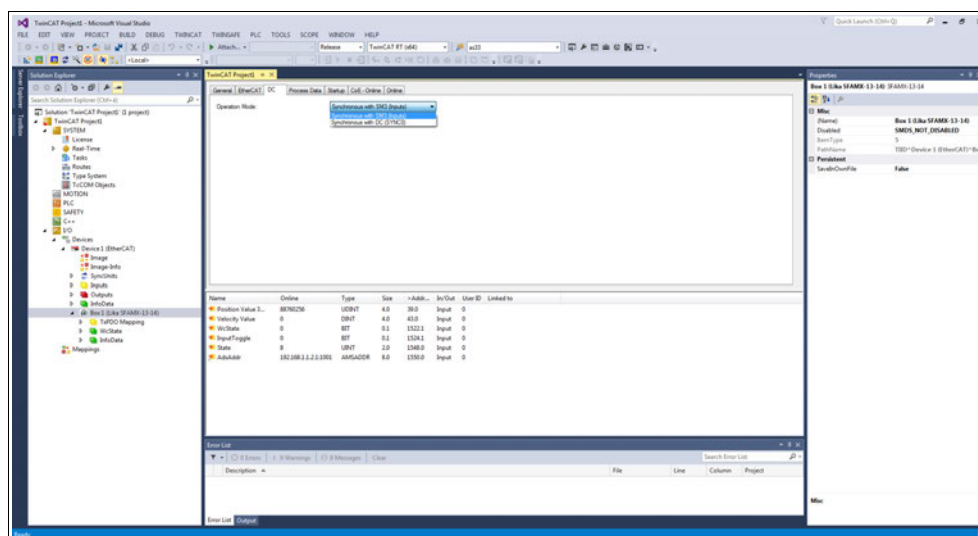


Figure 27 - Selecting the operation mode

For any further information on the Synchronous with SM3 operation mode, please refer to the "Synchronous with SM3" section on page 72 and to the [1C33-00 SM Sync Manager input parameter](#) object on page 86.



## 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.

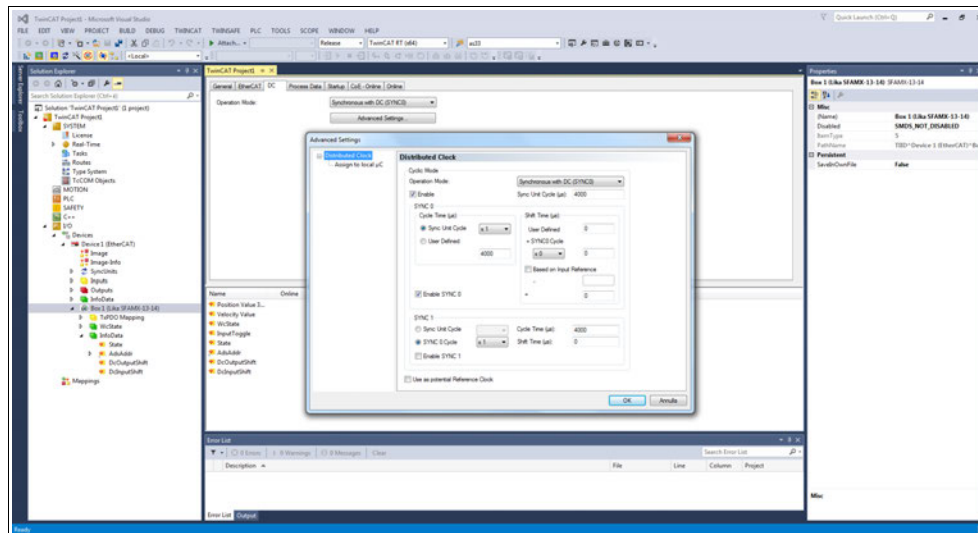


Figure 28 – Synchronous with DC operation mode

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 73 and to the [1C33-00 SM Sync Manager input parameter](#) object on page 86.



## 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.

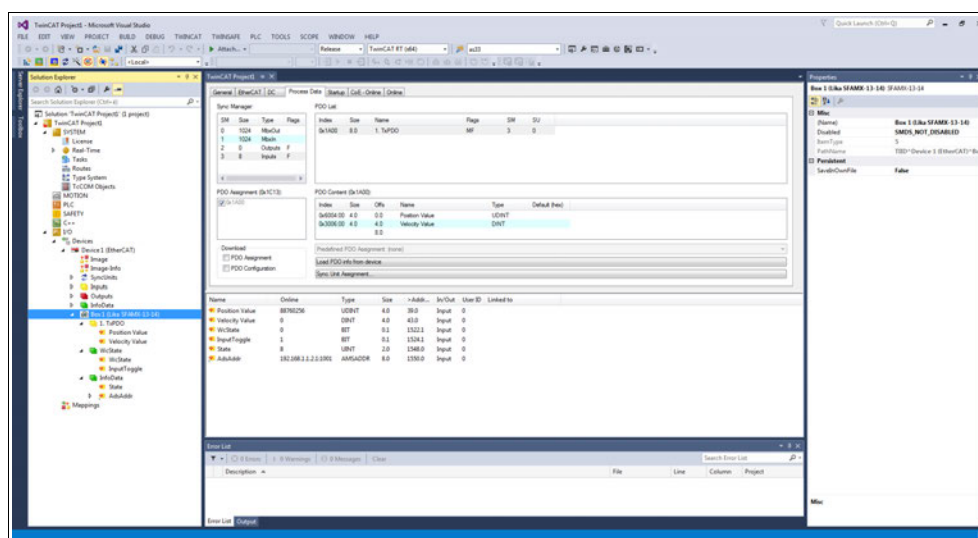


Figure 29 – Process data objects

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 encoder. 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 that 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** 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.

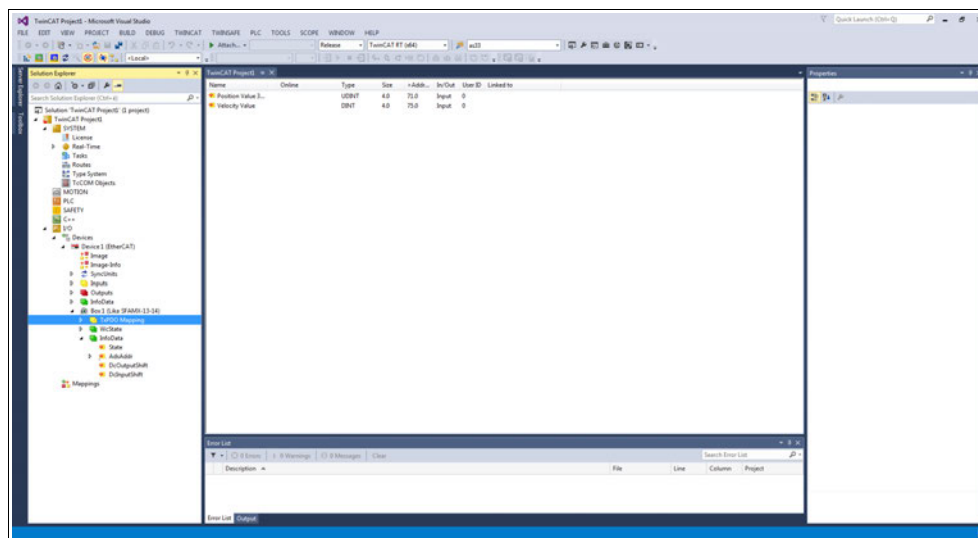


Figure 30 – TxPDO Mapping (offline)

If no values are 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.

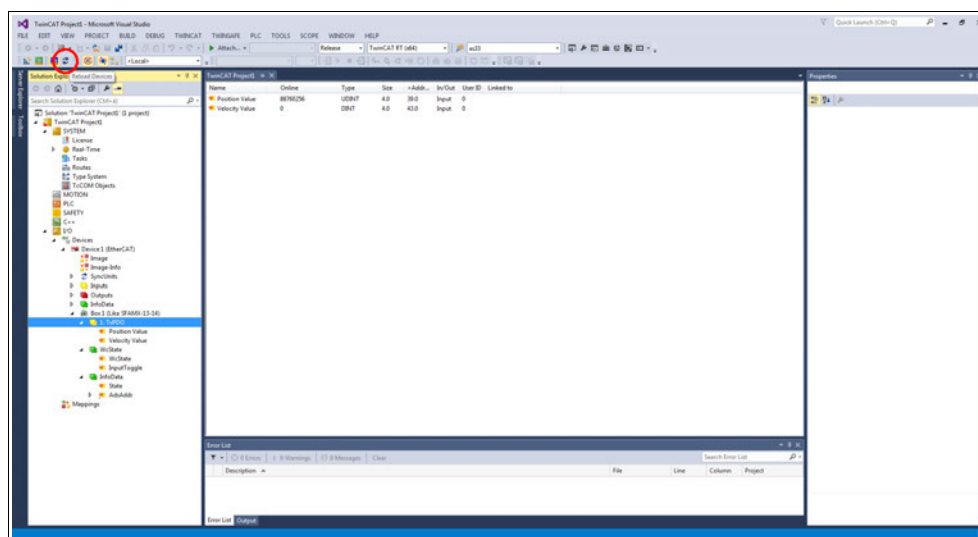


Figure 31 – TxPDO Mapping (online)

## 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.

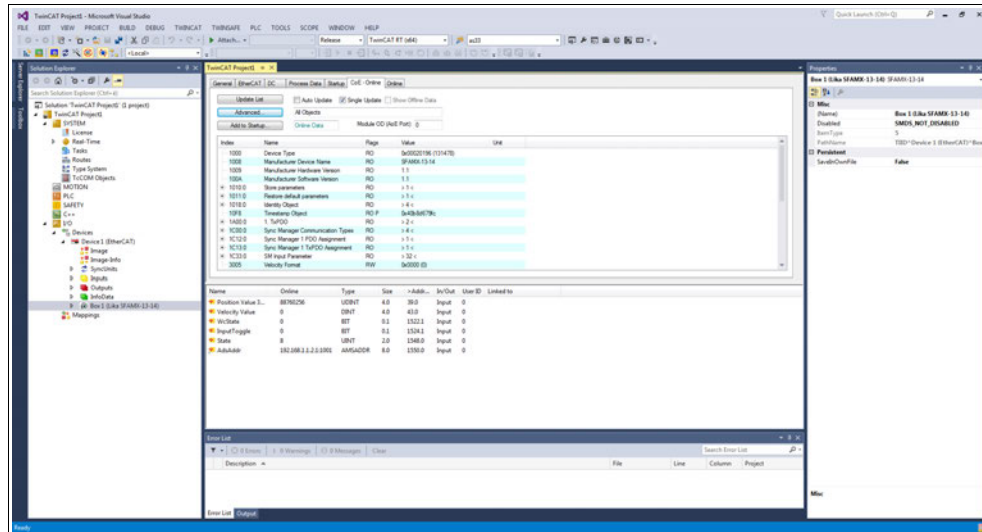


Figure 32 - CoE - Online tabbed page

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

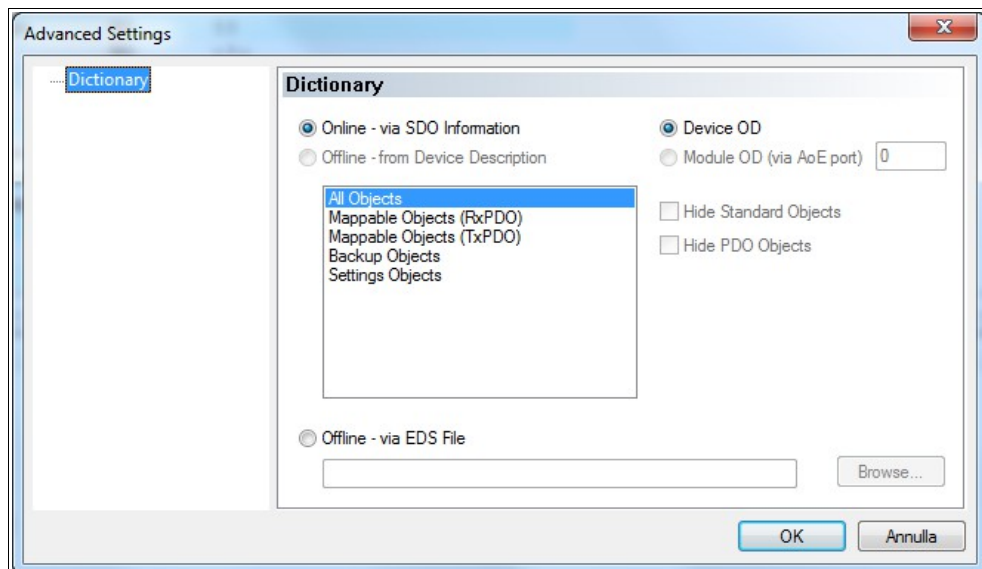


Figure 33 - CoE - Online - Advanced Settings

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 encoder 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.

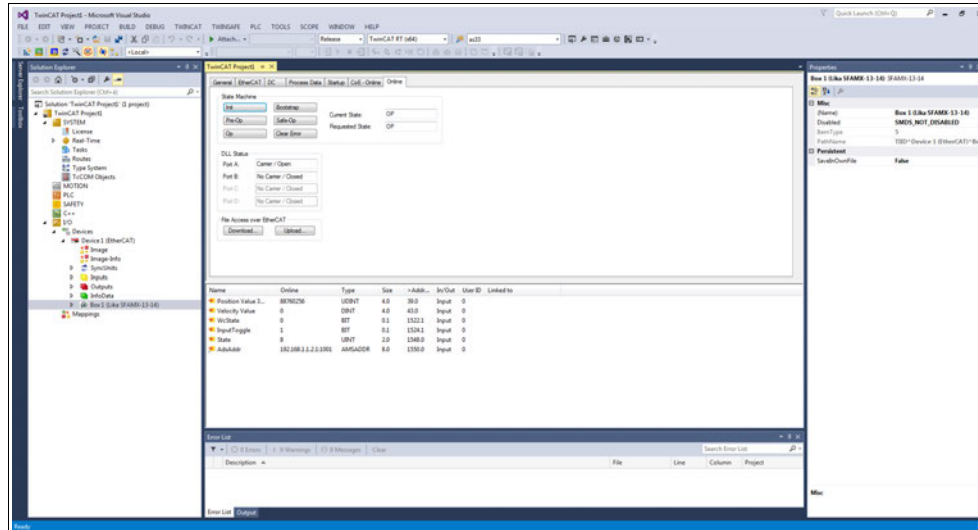


Figure 34 – Online tabbed page

To display the encoder 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 must be carried out by skilled and competent personnel. If the upgrade is not performed according to the provided instructions, or if an incorrect or incompatible firmware program is installed, the unit may not update correctly, which in some cases can prevent it from operating.

The firmware is a software program that controls the functions and operation of a device. This program, sometimes referred to as the "user program", is stored in the flash memory integrated within the unit. Lika encoders are designed so that the firmware can be easily updated by the user. This allows Lika Electronic to provide new and improved firmware versions throughout the product's lifetime. Typical reasons for releasing new firmware updates include the need to correct errors, enhance performance, and add new functionalities to the device.

The firmware upgrade program consists of a single file with a .EFW extension. It is released by Lika Electronic Technical Assistance & After-Sales 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 encoder you need to update: some tabbed pages for configuring and managing the device will appear. Navigate to the **Online** tabbed page.
3. Make sure that the Slave is in **Pre-Operational** state. To check the current state of the encoder, 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.

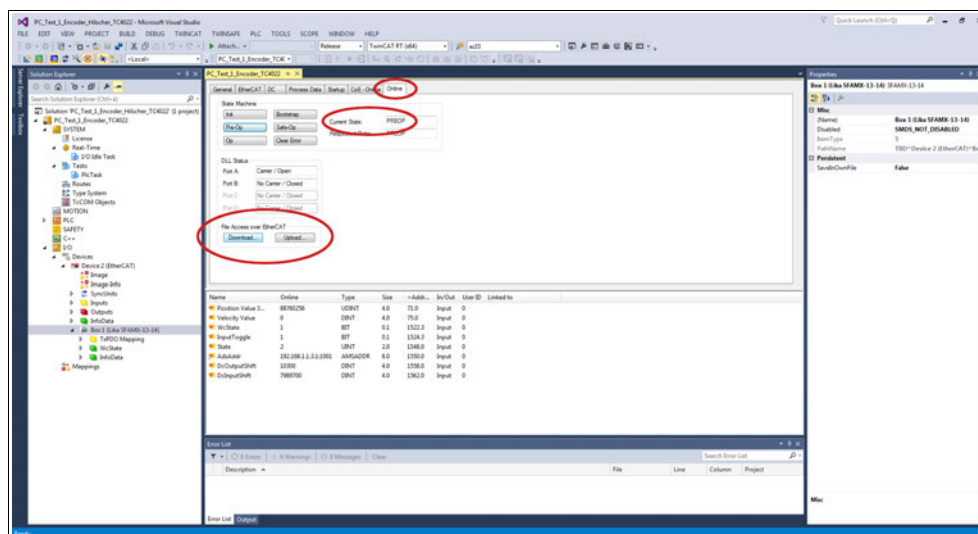


Figure 35 - Setting the State Machine

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 that 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.

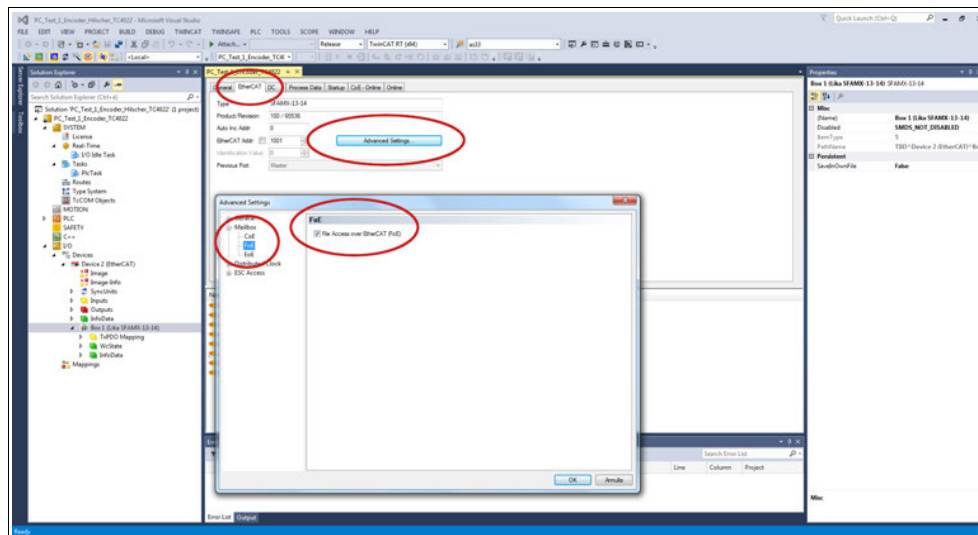


Figure 36 - Enabling the File Access over EtherCAT (FoE)

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

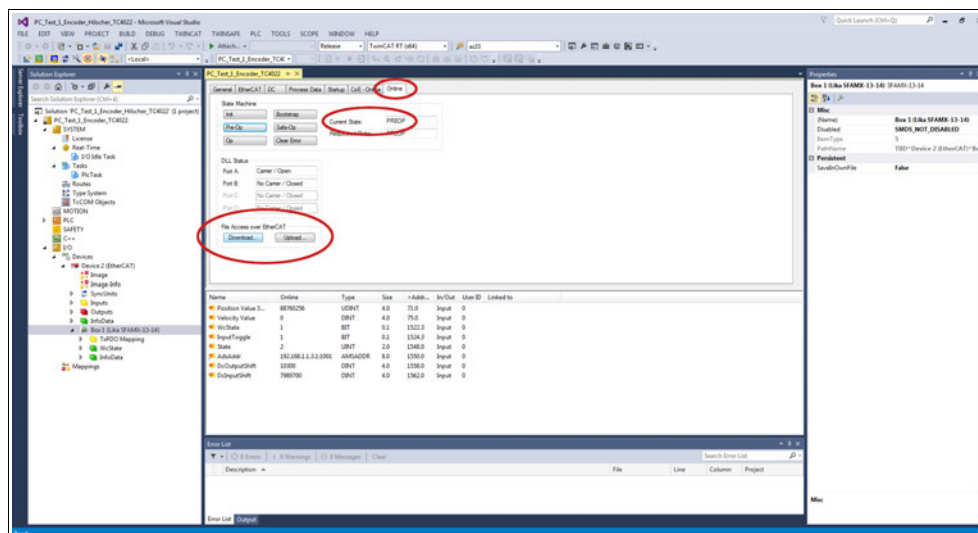


Figure 37 - Downloading a file via File Access over EtherCAT (FoE)

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.

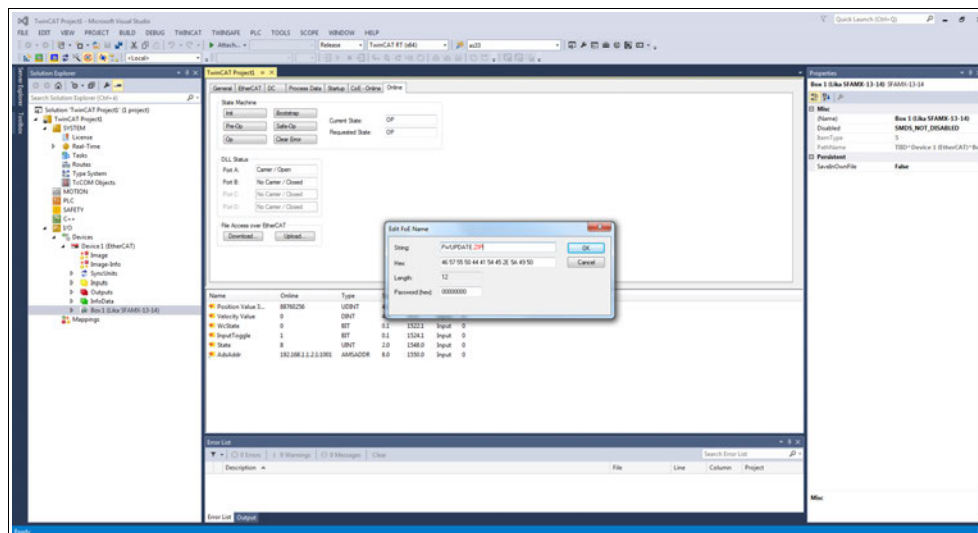


Figure 38 - Edit FoE Name dialog box

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

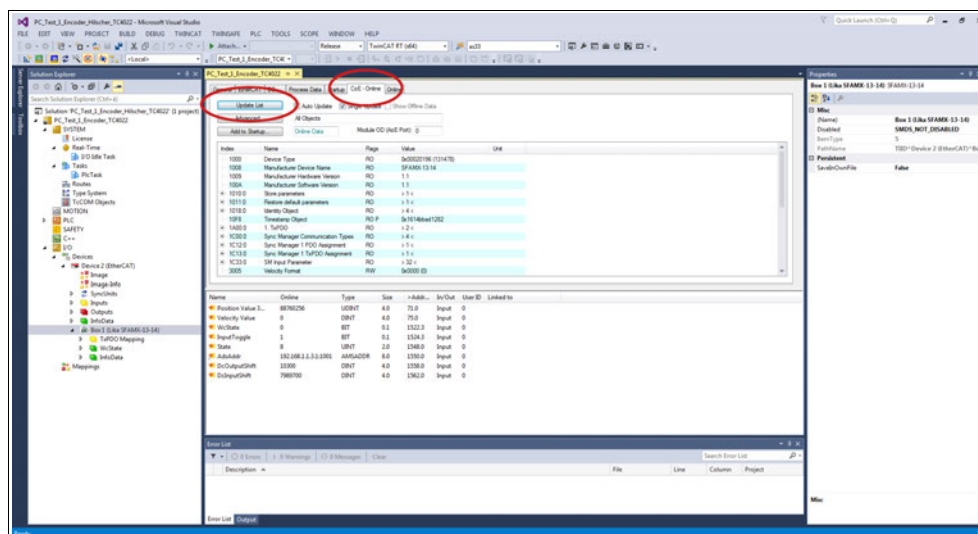


Figure 39 - Updating the object list



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.

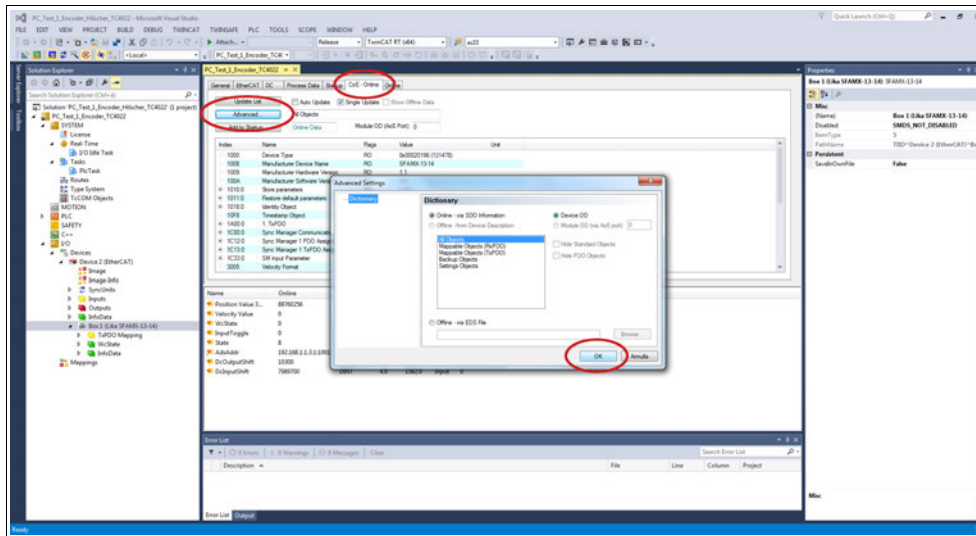


Figure 40 - Updating the object list

11. Scroll through the objects and double click on the object 6000 (see the **6000-00 Operating parameters** object on page 91).
12. Set the bit 15 **Mask Upgrade Firmware** to 1 (1000 0000 0000 0001 in binary notation = 32769 dec = 8001h), confirm pressing the **OK** button and check that the PWR LED starts flashing: it shall flash red while the update is in progress.

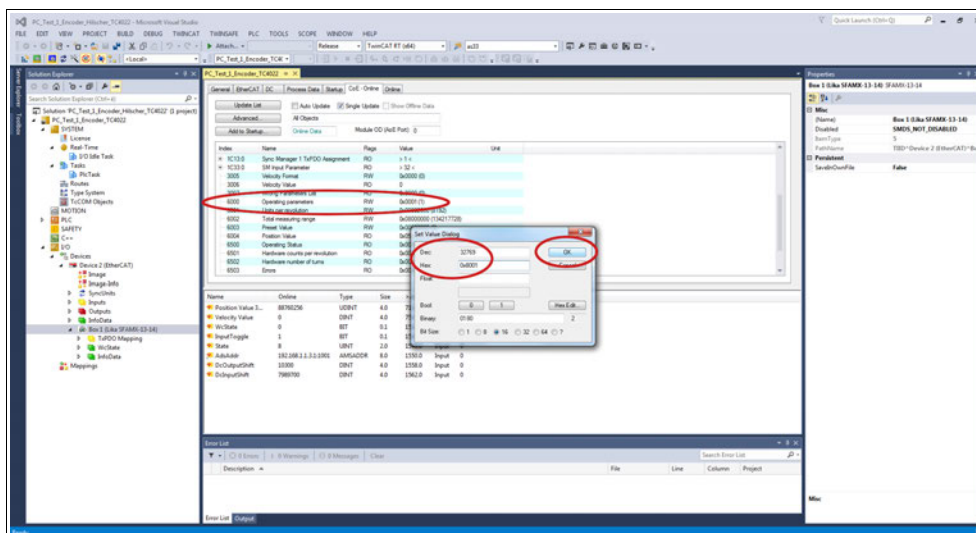


Figure 41 - Updating the object list



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**.

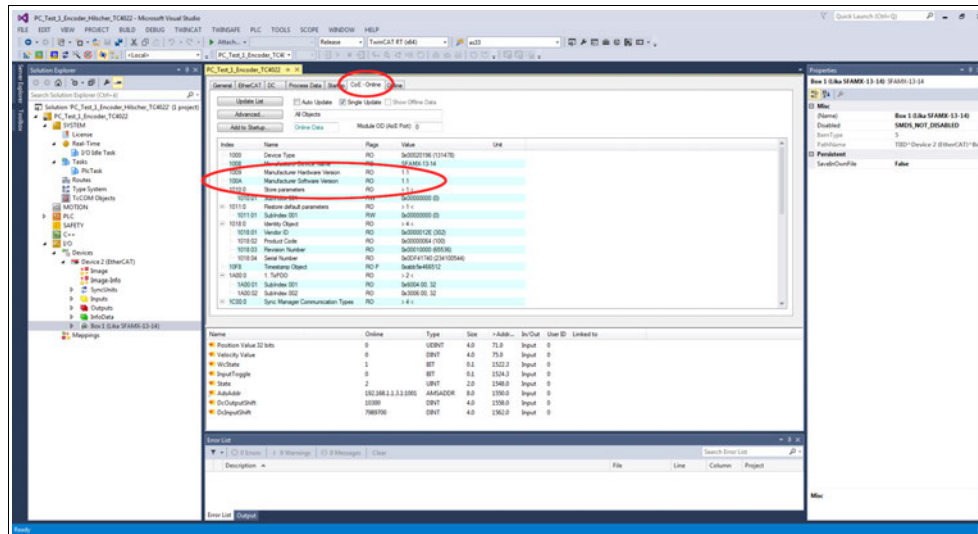


Figure 42 - Manufacturer Software Version

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

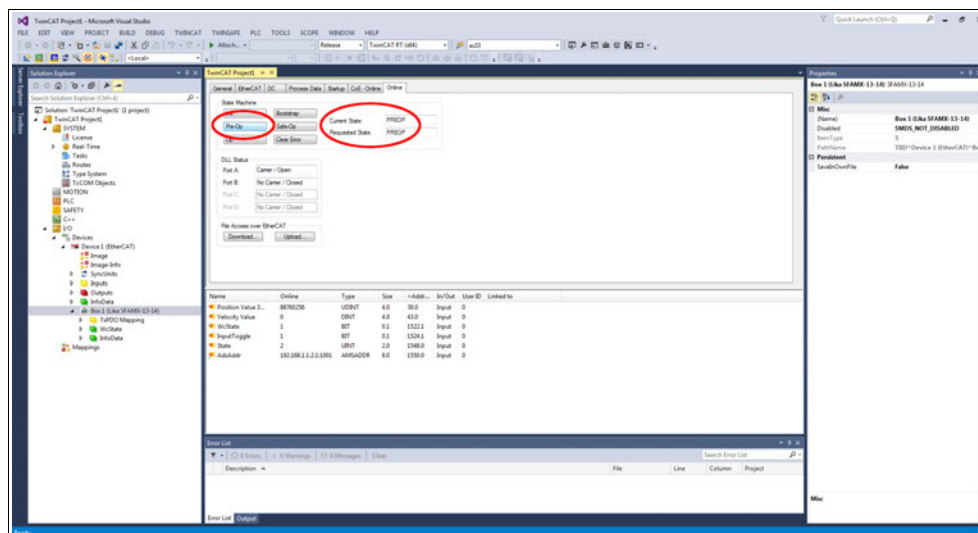


Figure 43 - Preoperational state

## 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.

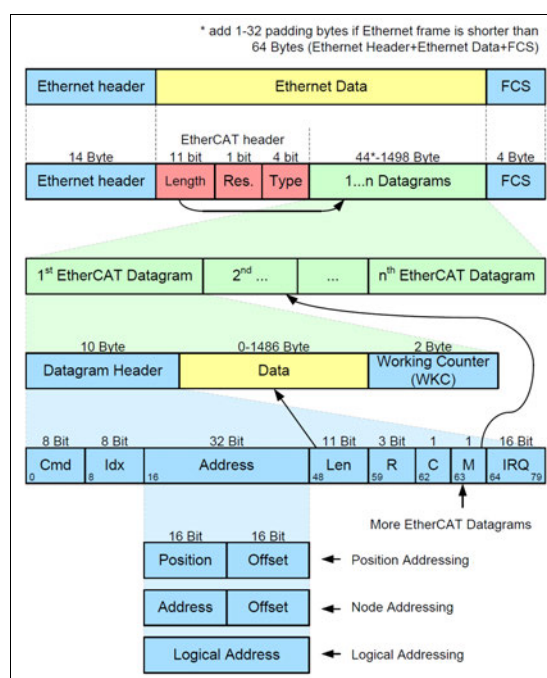
The main focus during the development of EtherCAT was on short cycle times ( $\leq 100 \mu\text{s}$ ), low jitter for accurate synchronization ( $\leq 1 \mu\text{s}$ ) and low hardware costs. EtherCAT was introduced in April 2003, and the EtherCAT Technology Group was founded in November 2003 - Meanwhile 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).

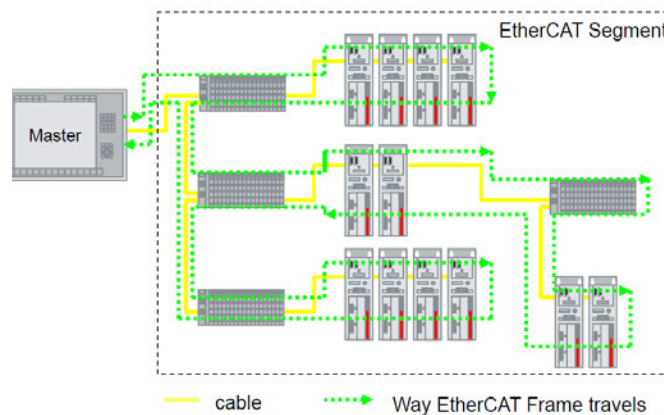
An EtherCAT frame can contain several telegrams and a complete control cycle often requires more than one frame.

### 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.

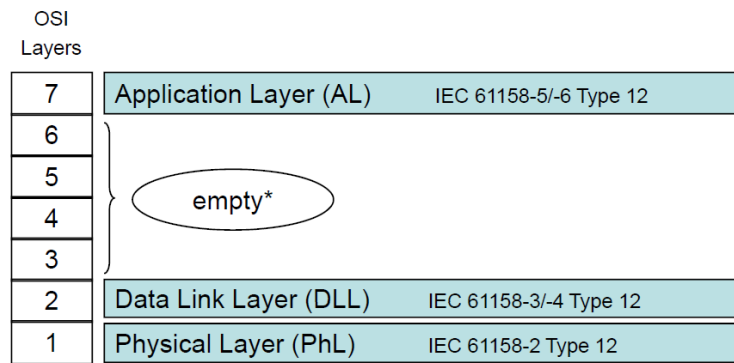
The last Slave issues back the complete frame to the Master with all the requested data (again passing through all the Slaves).



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



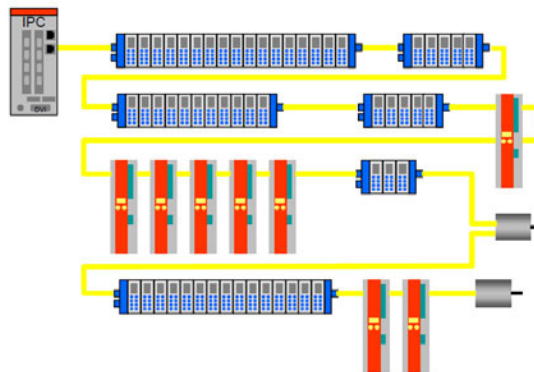
\* "Empty" means that the layer behaviour exists, but is not shown explicitly.

### 7.1.3 Topology

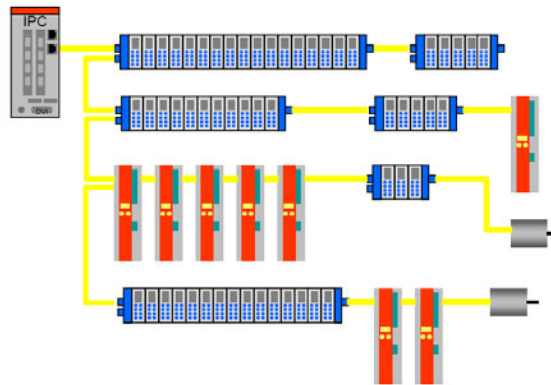
Several topologies of connection are supported by the EtherCAT networks: Line, Tree, Daisy Chain + Drop Line, Star, ...). EtherCAT networks can be configured in almost any topology in the same structure. The maximum length of the cable between two Slaves is 100 m / 328 ft; standard EtherCAT cables commercially 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:

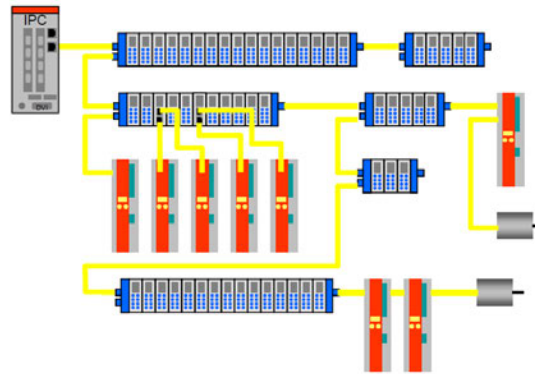
LINE topology:



TREE topology:



DAISY CHAIN with drop lines topology:



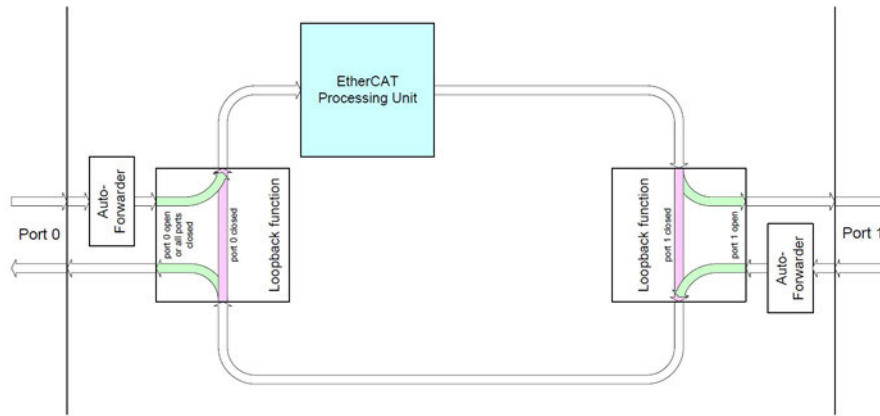
#### 7.1.4 Line Termination

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.

8 Bit	8 Bit	32 Bit		11 Bit	2	1	1	1	16 Bit
Cmd	Idx	Address		Len	R	C	R	M	IRQ
APxx		16 Bit	16 Bit						
		Position	Offset						
FPxx		Address	Offset						
Lxx		Logical Address							

← Auto Increment Addressing  
(Position addressing)

← Fixed Physical Addressing  
(Node addressing)

← Logical Addressing

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).



#### WARNING

Please note that since EtherCAT is not based on TCP/IP, there is no need to administer MAC addresses or IP addresses. However, you need the IP Address to reach, for instance, the web server. To know the IP address assigned by Beckhoff, see on page 114.

### 7.1.6 Communication mode

Lika encoders 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.

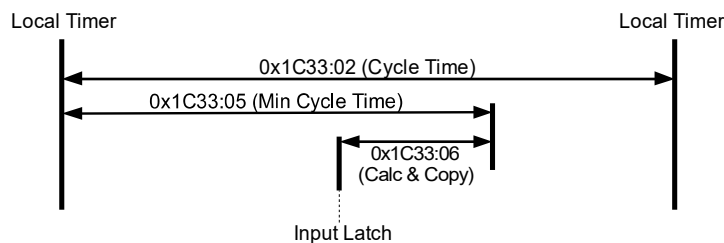
A reference parameter is the "Jitter": it represents the temporal fluctuation of the instant data sampling; in other words, data sampled by the micro-controller is available in ECAT DPRAM memory after a certain time and the measure of the variability over time is the "jitter".

#### FreeRun

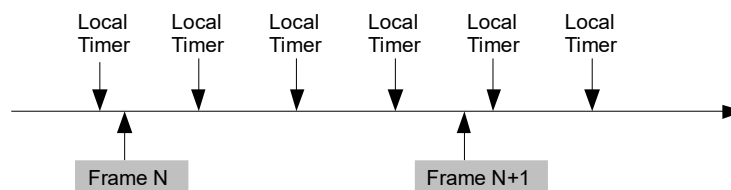
The FreeRun mode is supported only when the encoder is in **PREOP** state and the parameters (position and speed included) are read asynchronously. You can check and control IO by using the FreeRun mode without having to activate the configuration and write a PLC logic. This tool is very helpful during the commissioning stage to ensure that 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.

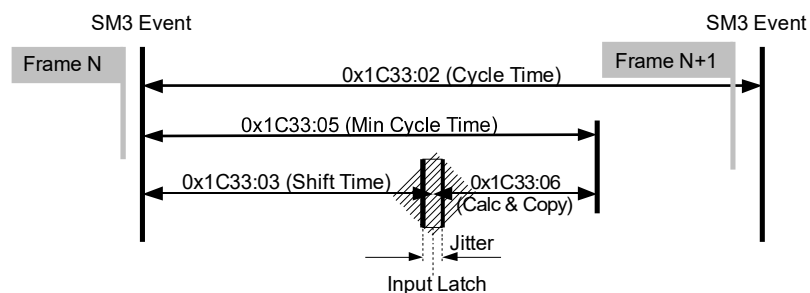


Description	Min	Typ	Max	
Jitter	0		500	μs
Cycle Time	500		64000	μs

See the [1C33-00 SM Sync Manager input parameter](#) entry on page 86.

### 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). 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

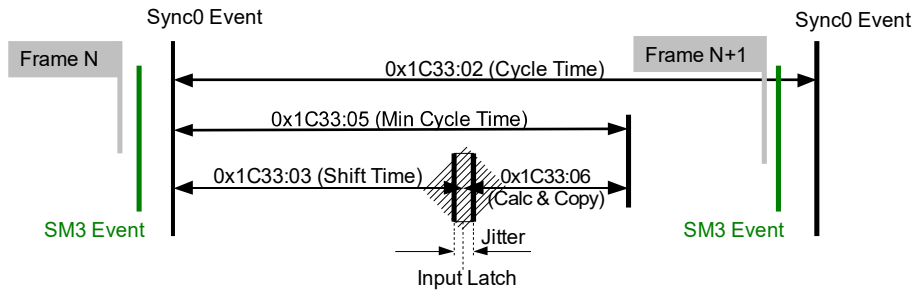
See the [1C33-00 SM Sync Manager input parameter](#) entry on page 86.



## 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.

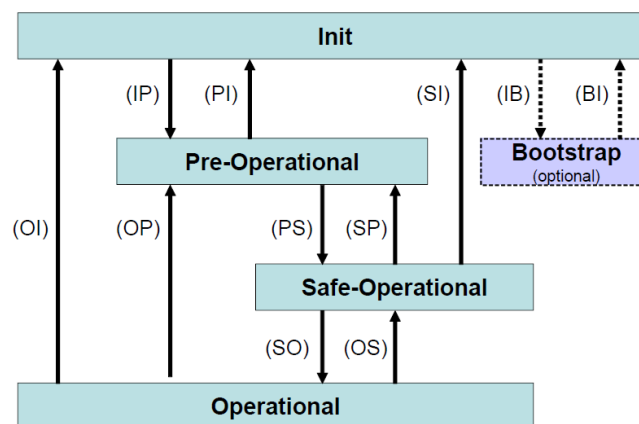
Time required for accomplishing these operations is set in object **1C33-00 SM Sync Manager input parameter** object; in particular, in the **03 Shift Time** entry (1C33hex, sub3) and in the **06 Calc and copy time** entry (1C33hex, sub6).



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.

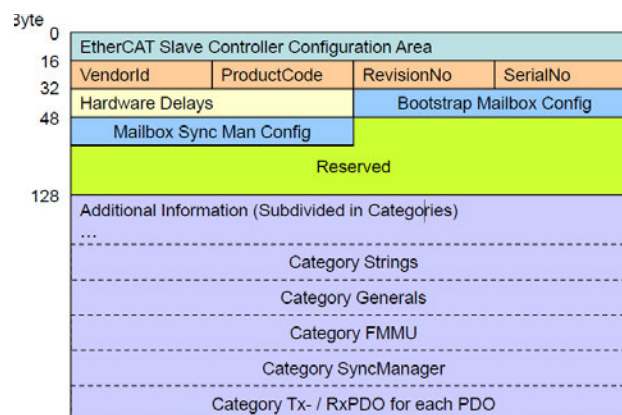
- **INIT**: it is the default state after power-on; in this state there is not direct communication between the Master and the Slave on the Application Layer; some configuration registers are initialized and the Sync Managers are configured.
- **PRE-OPERATIONAL** (PREOP): in this state the mailbox is 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 (SYNCO 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 encoder features four Sync Managers, see the [1C00-00 Sync Manager Communication Type](#) object on page 85:

- **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 1,024 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 1,024 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 encoders 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 EtherCAT® protocol, please refer to the "ETG.1000 EtherCAT Specification" document available at the address [www.ethercat.org](http://www.ethercat.org).

For any omitted specification on 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](http://www.can-cia.org).

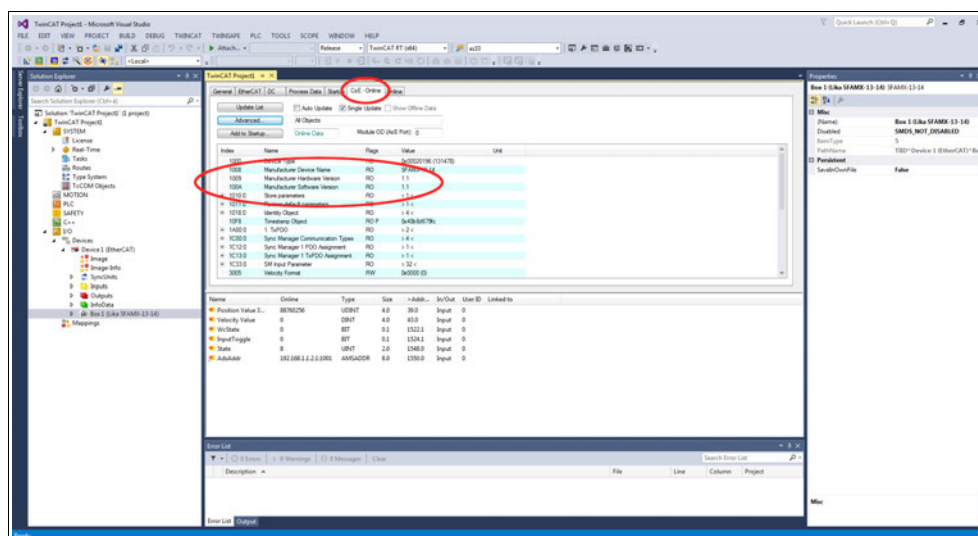
### 7.2.1 XML file

EtherCAT® encoders are supplied with their own XML file. It can be downloaded from Lika Electronic web page, see at [www.lika.biz](http://www.lika.biz) > **PRODUCTS** > **DRAW-WIRE ENCODERS**.

For more information, please refer to the "6.3 Adding the XML file" section on page 42.

For any information on the firmware update procedure, please refer to the "6.9 Firmware update" section on page 61; and to the "8.6 Firmware update" section on page 122.

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 objects **1009 Manufacturer Hardware Version (1009-00 Manufacturer Hardware Version)** and **100A Manufacturer Software Version (100A-00 Manufacturer Software Version)**.



### 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 the 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 81.

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 88.

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 91.

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 82.

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



### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values, and performing a preset operation. The machine must be compulsorily in stop.



## 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 = 0002 0196h = multiturn encoder, in compliance with DS 406

### 1008-00 Manufacturer Device Name

[String, ro]

It shows the manufacturer device name, expressed in hexadecimal ASCII code.

Default = 5346414D582D31322D3134h = "SFAMX-13-14" = SFAMx 27 bit multiturn draw-wire encoder

### 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

312E31h = 1.1 = Software version 1.1

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 → Encoder

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

Encoder → Master (confirmation)

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



#### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values, and performing a preset operation. The machine must be compulsorily in stop.

### 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 127.

Write **"load"** in hexadecimal ASCII code in the data bytes:

Master → Encoder

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

Encoder → Master (confirmation)

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



#### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values, and performing a preset operation. The machine must be compulsorily in stop.



## 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

[Unsigned8, ro]

This object contains 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 0096h SFAMx 27 bit multiturn draw-wire encoder

### 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 SFAMx series draw-wire encoder

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 ns of the local time at the send/receipt of a frame.

Default = 16687756753650 (0F2D 6BD0 3AF2h)

#### 1A00-00 TxPDO mapping parameter

[Unsigned8, 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 in bits. This may be used to verify the mapping.

7	0	15	8	23	16	31	24
Length		Sub-Index		Index			
LSB <span style="float:right">MSB</span>							

Default = 6004 0020h = **6004-00 Position Value** 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 in bits. This may be used to verify the mapping.

7	0	15	8	23	16	31	24
Length		Sub-Index		Index			
LSB <span style="float:right">MSB</span>							

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

### 1C00-00 Sync Manager Communication Type

[Unsigned8, ro]

This object contains the number and type of Sync Manager Communication Types supported by the encoder. Sub-index 00 specifies the number of Sync Manager channels. Refer also to the "Sync Manager" section on page 75.

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

**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 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 71.

0: FreeRun: see on page 71;

1: Synchronous with SM3 Event (no outputs available): see on page 72;

2: DC mode synchronous with SYNC0 event: see on page 73.

Default = 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 = 2,008,660

#### 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 = 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 = 7

#### 05 Minimum cycle time

[Unsigned32, ro]

Minimum duration of the encoder 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 = 125,000

#### 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 = 70,000

#### 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 output 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 82.

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

## Manufacturer Specific Profile Area objects

### 3005-00 Velocity Format

[Unsigned16, rw]

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 RPM

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** 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 84.

### 3007-00 Wrong parameters list

[Unsigned16, 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	<b>Units per revolution exceeded</b>	Warning not active	Warning active
1	<b>Total Measuring Range exceeded</b>	Warning not active	Warning active
2	<b>Preset value exceeded</b>	Warning not active	Warning active
3	<b>Offset value exceeded</b>	Warning not active	Warning active
4	<b>Speed Format exceeded</b>	Warning not active	Warning active
5 ... 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** 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** 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** 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.

**Speed Format exceeded**

bit 4 Wrong data has been set next to the **3005-00 Velocity Format** 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.

bits 5 ... 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 82.

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	<b>Code sequence</b>	Count up information rewinding the wire	<b>Count up information pulling the wire out</b>
1	not used		
2	<b>Scaling function</b>	<b>Disabled</b>	Enabled
3 ... 14	not used		
15	<b>Mask Upgrade Firmware</b>	<b>New firmware is not loaded</b>	Firmware is loaded

Default values are highlighted in bold

Default = 0001h

#### Code sequence

This is intended to set whether the count is increasing (count up information) when you rewind the wire or when you pull the wire out. Setting 0 (bit 0 **Code sequence** = 0) causes the encoder counting to increase when you rewind the wire; setting 1 (bit 0 **Code sequence** = 1) causes the encoder counting to increase when you pull the wire out.

To know whether the **Code sequence** is currently set to "Count up information rewinding the wire" or "Count up information pulling the wire out", you can read the bit 0 **Code sequence** in the **6500-00 Operating Status** object, see on page 101.



#### WARNING

Every time you change the **Code sequence**, then you are required to activate a new preset (see the **6003-00 Preset Value** 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**.

When the scaling function is disabled (bit 2 **Scaling function** = 0), the encoder uses the physical singleturn resolution and the physical multiturn resolution (i.e. the hardware counts per revolution and the number of hardware revolutions, see the encoder identification label and the objects **6501-00 Hardware counts per revolution** and **6502-00 Hardware number of turns**) to calculate the absolute position information; the **6001-00 Units per revolution** and **6002-00 Total measuring range** objects are ignored.

On the contrary, when the scaling function is enabled (bit 2 **Scaling function** = 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** object 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 101.



#### 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** objects), then you are required to activate a new preset (see the **6003-00 Preset Value** object) and finally save the new parameters (see the **1010-01 Store parameters** object).



#### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values, and performing a preset operation. The machine must be compulsorily in stop.

#### 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 107.

#### 6001-00 Units per revolution

[Unsigned32, rw]



#### 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 counts per revolution** and **6502-00 Hardware number of turns**) to calculate the position information.

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

To avoid counting errors, check that

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

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 counts per revolution** 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** object (see the bit 0 **Units per revolution exceeded**) signal the error.

Default = 0000 2000h = 8,192 (min. = 1, max. = 8,192)



#### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values, and performing a preset operation. The machine must be compulsorily in stop.



#### 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** object value and ensure that the resulting number of revolutions complies with the **Hardware number of revolutions** of the device (see the **6502-00 Hardware number of turns** object).

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

Let's suppose that the draw-wire encoder is programmed as follows:

**6001-00 Units per revolution**: 8,192 cpr

**6002-00 Total measuring range** = 33 554 432<sub>10</sub> = 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** value at the same time, we will get the following result:

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

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. 16,384. When this happens, the encoder falls into an error signalling the

faulty condition (see the [6505-00 Warnings](#) and [3007-00 Wrong parameters list](#) objects).



#### 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](#) 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](#) objects), then you are required to activate a new preset (see the [6003-00 Preset Value](#) object).

#### [6002-00 Total measuring range](#)

[Unsigned32, rw]



#### 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 values ([6501-00 Hardware counts per revolution](#) and [6502-00 Hardware number of turns](#)) to calculate the position information.

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 [6501-00 Hardware counts per revolution](#) and [6502-00 Hardware number of turns](#) objects). The overall hardware resolution results from:

[6501-00 Hardware counts per revolution](#) \* [6502-00 Hardware number of turns](#).

We recommend the **Number of revolutions** to be set to a power of 2.

The set **Number of revolutions** results from the following calculation:

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

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 requiring 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** object (see the bit 1 **Total Measuring Range exceeded**) signal the error.

Default = 0800 0000h = 134,217,728 (min. = 1, max. = 134,217,728)



#### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values and performing a preset operation. The machine must be compulsorily in stop.



#### WARNING

When you set a new value next to the **6002-00 Total measuring range** 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 (max. 16,384 revolutions).

Let's suppose that the encoder is programmed as follows:

**6001-00 Units per revolution**: 8,192 cpr

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

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

As the **6002-00 Total measuring range** 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** object) and finally save the new parameters (see the **1010-01 Store parameters** object).



#### EXAMPLE

We install the **SFAM3-05000-EC2-08192-RM12** draw-wire encoder.

The physical values are:

- Stroke per turn of the drum = 200 mm (7.874 inches)
- Physical single-turn resolution = **6501-00 Hardware counts per revolution** = 13 bits = 8,192 cpr ( $2^{13}$ )
- Physical multi-turn resolution = **6502-00 Hardware number of turns** = 14 bits = 16,384 turns ( $2^{14}$ )

- Total hardware resolution = **6501-00 Hardware counts per revolution** \* **6502-00 Hardware number of turns** = 27 bits = 134 217 728 ( $2^{13} \times 2^{14} = 2^{27}$ )
- Physical linear resolution: 0.024 mm = 24 µm (200 mm / 8,192 cpr)
- Max. number of turns of the drum = 25
- Max. measuring length = 5,000 mm (196.85 inches)
- Max. number of information = 208,300 counts (5,000 / 0.024)

Let's suppose that we need a **tenth of a millimetre linear resolution** for the specific installation:

- Enable the scaling function: **6000-00 Operating parameters**, bit 2 **Scaling function** = "1"
- Custom resolution per turn: **6001-00 Units per revolution** = 2,000 cpr
- Linear resolution: 0.1 mm = 100 µm

$$\text{Linear resolution} = \frac{\text{Stroke per turn mm}}{\text{6001-00 Units per revolution}} = \frac{200 \text{ mm}}{2,000 \text{ cpr}} = 0.1 \text{ mm}$$

The custom number of revolutions can be equal to the physical number of revolutions:

$$\text{Custom number of encoder revolutions} = \frac{\text{6002-00 Total measuring range}}{\text{6001-00 Units per revolution}} = 16,384$$

- **6002-00 Total measuring range** = 32 768 000



#### NOTE

Please note that if you set a preset along the path, when the wire is rewound and passes zero while moving backward, the value immediately after 0 will be 32 768 000 - 1, i.e. 32 767 999.

...	32,767,997	32,767,998	32,767,999	0	1	2	...
-----	------------	------------	------------	---	---	---	-----



#### EXAMPLE

Let's use the values in the previous example and let's suppose that the travel in the application is 2 m long. As the stroke per turn is 200 mm (7.874 inches), you will need 10 revolutions to cover the travel length.

- **6002-00 Total measuring range** = **6001-00 Units per revolution** \* custom number of revolutions = 2,000 \* 10 = 20,000



In fact:

$$\text{Custom number of encoder revolutions} = \frac{\text{6002-00 Total measuring range}}{\text{6001-00 Units per revolution}} = 10$$

In this case, you will obtain several 20,000 information long 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.

...	19,997	19,998	19,999	0	1	2	...	19,997	19,998	19,999	0	1	2	...
← max. measuring length →														



#### NOTE

We recommend 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** objects to avoid counting errors.



#### WARNING

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

#### 6003-00 Preset Value

[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 (i.e. to a position in the travel of the wire). 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 (i.e. the position of the wire) in the moment when the preset value is transmitted. We suggest setting the preset value when the encoder is in stop.

Default = 0000 0000h



### WARNING

Please be careful when saving the parameters, restoring the default parameters, setting new resolution values, and performing a preset operation. The machine must be compulsorily in stop.



### 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** 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** - **6509-00 Offset**.

If you never set the **6003-00 Preset Value** and you never performed the preset setting, then the transmitted value and the read position are necessarily the same as **6003-00 Preset Value** = 0 and **6509-00 Offset** = 0.

When you set the **6003-00 Preset Value** 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** are the same as read position - **6509-00 Offset** = 0; in other words, the value set next to the **6003-00 Preset Value** 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** 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** (= "50") - **6509-00 Offset** (= "1000") = 50.

The following transmitted value will be:

**Transmitted value = read position** (= "1001") + **6003-00 Preset Value** (= "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** (= 1000 = 03E8h)

Master → Encoder

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

Encoder → 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** must be less than or equal to the **total hardware resolution** – 1, i.e. (**6501-00 Hardware counts per revolution** \* **6502-00 Hardware number of turns**) – 1.
- 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** must be less than or equal to **6002-00 Total measuring range** – 1.



#### WARNING

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

#### 6004-00 Position Value

[Unsigned32, ro]

This object contains the information about the current position of the encoder. If the scaling function is enabled (see the bit 2 **Scaling function** in the **6000-00 Operating parameters** object), the output value is scaled according to the scaling parameters. The **6004-00 Position Value** object is mapped in the **1A00-00 TxPDO mapping parameter** object, sub-index 001 **01 Mapped Object 001**, see on page 84.



#### WARNING

Please note that the position value issued by the encoder is expressed in counts; therefore, you have then to convert the number of counts into a linear measuring unit.

To convert the position value into millimetres (mm) or micrometres (µm) you have to multiply the number of information by the linear resolution of the encoder expressed in millimetres or micrometres.

To know the linear resolution of the encoder, please consider that **the stroke per turn of the drum is 200 mm (7.874 inches)**.

The linear resolution results from the following calculation:

$$\text{Linear resolution} = \frac{\text{Stroke per turn of the drum mm}}{\text{Singleturn resolution cpr}}$$

If you want to know the linear position value, you will need to multiply the transmitted position value by the linear resolution.

**Linear position value** = transmitted position \* linear resolution



**NOTE**

Please note that the encoder's physical linear resolution can be read also in the order code next to the rotary resolution. Refer to the product datasheet.



**EXAMPLE 1**

Let's suppose that we are using the physical resolution of the SFAM3-05000-EC2-08192-RM12 draw-wire encoder (the bit 2 **Scaling function** in the **6000-00 Operating parameters** object = 0).

The physical singleturn resolution of the measuring device is 8,192 cpr (= 0.024 mm, see the order code in the product datasheet).

As stated, the linear resolution results from the following calculation:

$$\text{Linear resolution} = \frac{\text{Stroke per turn of the drum mm}}{\text{Resolution cpr}}$$

$$\text{Linear resolution} = \frac{200 \text{ mm}}{8,192 \text{ cpr}} = 0.024 \text{ mm} = 24 \text{ }\mu\text{m}$$

Let's say that the transmitted position value is 123.

Therefore, the linear position value will be as follows:

**Linear position value** = transmitted position \* linear resolution

$$\text{Linear position value} = 123 \text{ counts} * 0.024 \text{ mm} = 2.952 \text{ mm} = 2,952 \text{ }\mu\text{m}$$



**EXAMPLE 2**

Let's suppose that we are using the SFAM3-05000-EC2-08192-RM12 draw-wire encoder. The singleturn resolution is set to the custom value of 4,000 cpr (**6001-00 Units per revolution** = 4000). The transmitted position value is 1,569.

The linear resolution can be easily calculated as follows:

$$\text{Linear resolution} = \frac{200 \text{ mm}}{4,000 \text{ cpr}} = 0.05 \text{ mm} = 50 \text{ }\mu\text{m}$$

Therefore, the linear position value will be as follows:

$$\text{Linear position value} = 1,569 \text{ counts} * 0.05 \text{ mm} = 78.45 \text{ mm} = 78\,450 \text{ }\mu\text{m}$$

## 6500-00 Operating Status

[Unsigned16, ro]

Bit	Function	bit = 0	bit = 1
0	Code sequence	Count up information rewinding the wire	Count up information pulling the wire out
1	not used		
2	Scaling function	Disabled	Enabled
3 ... 15	not used		

### Code sequence

It shows the value that is currently set through the bit 0 **Code sequence** in the **6000-00 Operating parameters** object. If the bit is "0", the output encoder position value has been set to increase when you rewind the wire; if the bit is "1" instead, the output encoder position value has been set to increase when you pull the wire out. To set the code sequence to either value, you must set the bit 0 **Code sequence** in the **6000-00 Operating parameters** object to 0 / 1. For any further information on setting and using the counting direction function, please refer to the **6000-00 Operating parameters** object on page 91.

### Scaling function

It shows the value that is currently set through the bit 2 **Scaling function** in 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 any further information on setting and using the scaling function, please refer to the **6000-00 Operating parameters** object on page 91.

## 6501-00 Hardware counts per revolution

[Unsigned32, ro]



### 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**) to calculate the position information.

This object is intended to show the number of physical distinguishable steps provided per each turn by the hardware (physical singleturn resolution, see the hardware counts per revolution in the encoder identification label).

If you want to set a custom singleturn resolution, please see the **6001-00 Units per revolution** object.

Default = 8,192

### 6502-00 Hardware number of turns

[Unsigned32, ro]



#### 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**) to calculate the position information.

This object is intended to show the number of physical turns provided by the hardware of the encoder (physical multi-turn resolution, see the hardware revolutions in the encoder identification label).

The **Total hardware resolution** results from **6501-00 Hardware counts per revolution** \* **6502-00 Hardware number of turns**.

If you want to set a custom number of turns see the **6001-00 Units per revolution** and **6002-00 Total measuring range** objects.

Default = 16,384

### 6503-00 Errors

[Unsigned16, ro]

The corresponding bits of supported errors are set (see the **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	<b>Position error</b>	Alarm not active	Alarm active
1 ... 11		not used	
12	<b>Flash memory error</b>	Alarm not active	Alarm active
13 ... 15		not used	

#### Position error

Error bit of the position read system. Fault and malfunction of the encoder 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 encoder. Refer to the previous **6503-00 Errors** object.

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

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

## 6505-00 Warnings

[Unsigned16, ro]

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

The **6505-00 Warnings** object indicates that tolerances for certain internal parameters of the encoder 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	<b>Machine data not valid</b>	Warning not active	Warning active
13	not used		
14	<b>Position Warning</b>	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** object, see on page 88.

### Position Warning

Warning bit of the position read system. Fault and malfunction of the encoder position measurement system or the measured value processing unit. This

warning does not cause an invalid position and speed actual value, 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	<b>Machine data not valid</b>	Not supported	<b>Supported</b>
13	not used		
14	<b>Position Warning</b>	Not supported	<b>Supported</b>
15	not used		

Default = 5000h (= 0101 0000 0000 0000 = warnings at bits 12 and 14 are supported and displayed next to 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 82).

For any further information on the preset function and the meaning and use of the related objects **6003-00 Preset Value** and **6509-00 Offset** refer to page 97.



#### NOTE

To save the new parameters execute the store parameters function (see the **1010-01 Store parameters** object on page 82).

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 "4.6 Diagnostic LEDs (Figure 1)" 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 <b>PRE-OPERATIONAL</b> to <b>SAFE-OPERATIONAL</b> not successful A001hex: transition from <b>SAFE-OPERATIONAL</b> to <b>OPERATIONAL</b> not successful Encoder errors: 5000hex: <b>Hardware error</b> 5001hex: <b>Diagnostic data</b> (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 "4.6 Diagnostic LEDs (Figure 1)" 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 encoders 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. Ensure that the Slave is in **Pre-Operational** state. To check the current state of the encoder, 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.

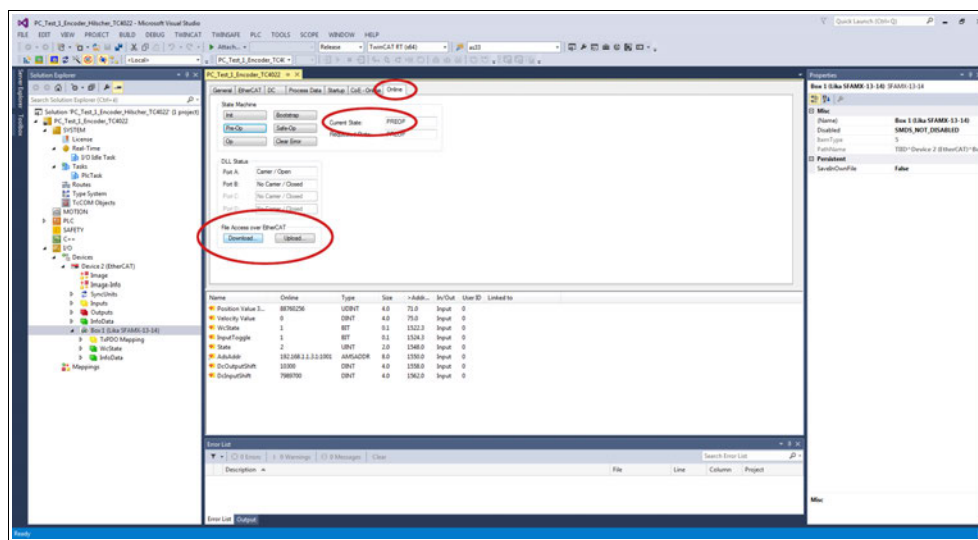


Figure 45 - Setting the State Machine

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.

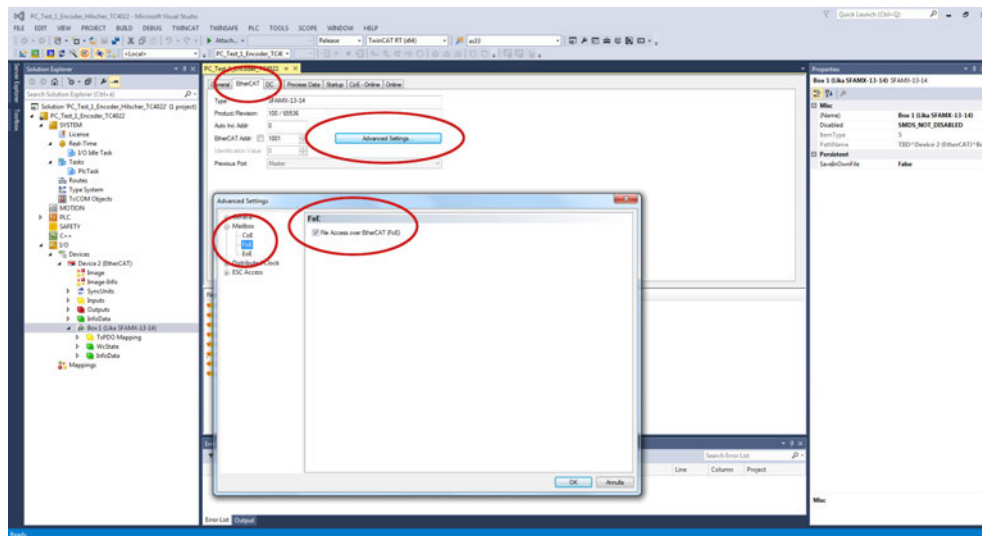


Figure 46 - Enabling the File Access over EtherCAT (FoE)

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

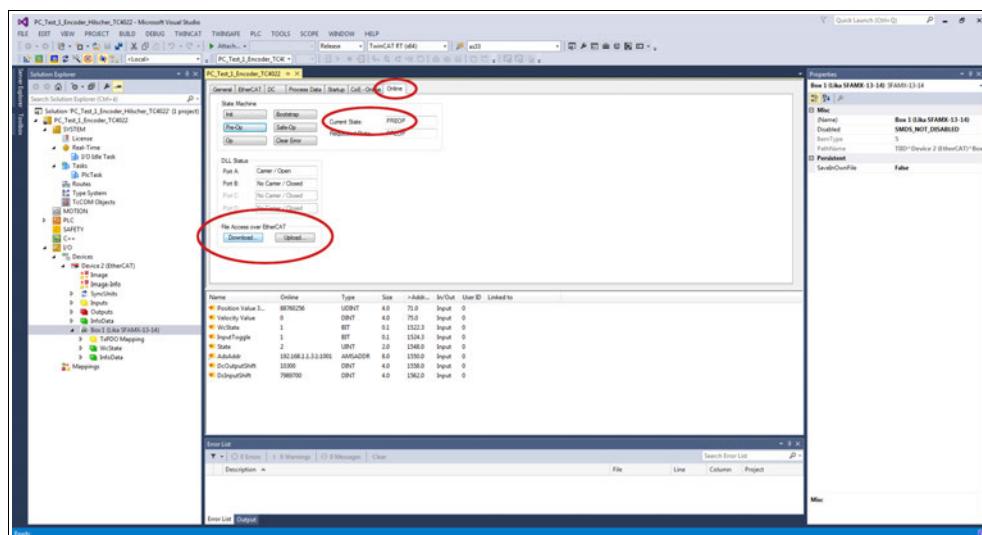


Figure 47 - Downloading/Uploading a file via File Access over EtherCAT (FoE)

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.

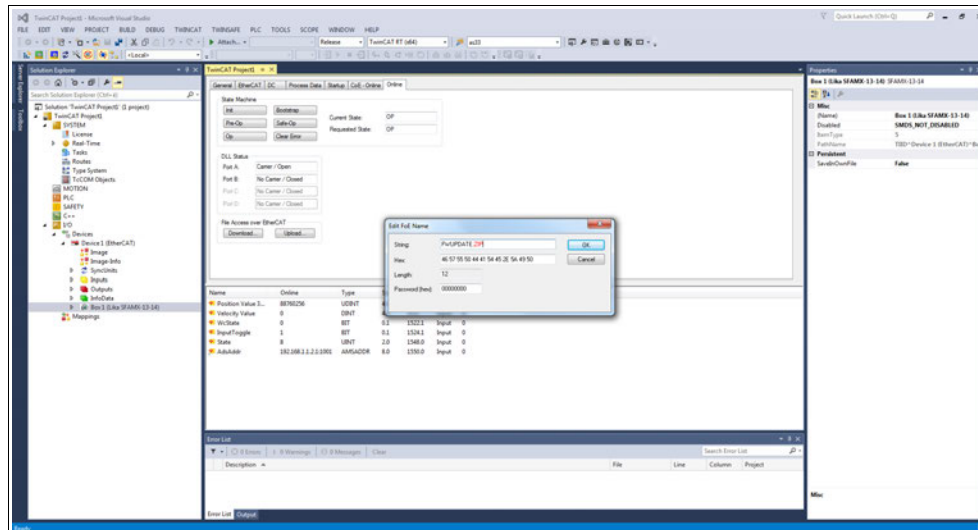


Figure 48 - Edit FoE Name dialog box

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

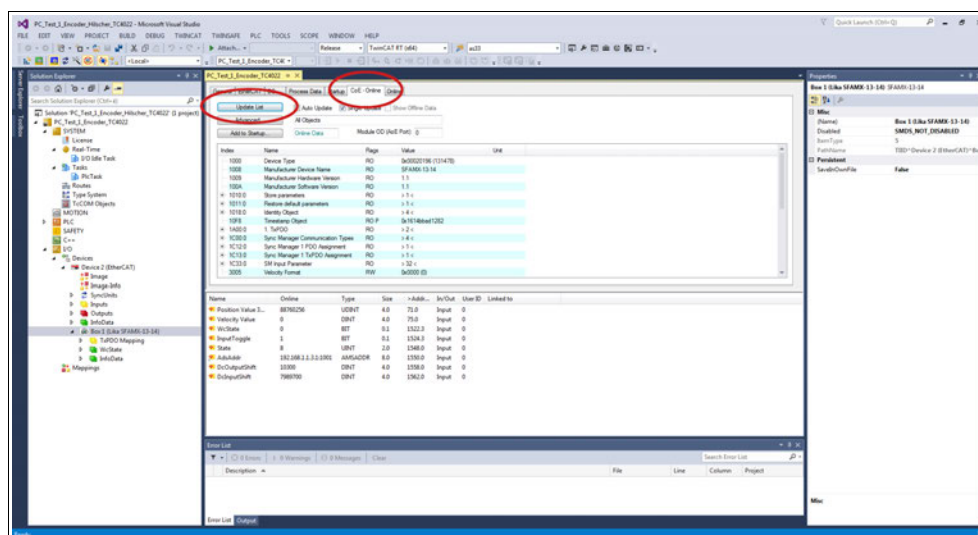


Figure 49 - Updating the object list

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.

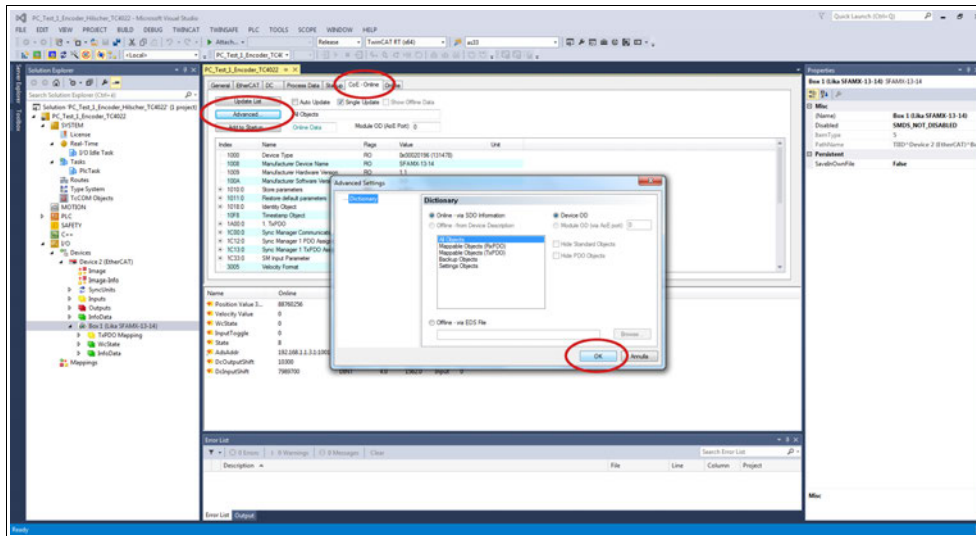


Figure 50 - Updating the object list

11. Scroll through the objects and double click on the object 6000 (see the **6000-00 Operating parameters** object on page 91).
12. Set the bit 15 **Mask Upgrade Firmware** to 1 (1000 0000 0000 0001 in binary notation = 32769 dec = 8001h), confirm pressing the **OK** button and check that the PWR LED starts flashing: it shall flash red while the upgrade is in progress.

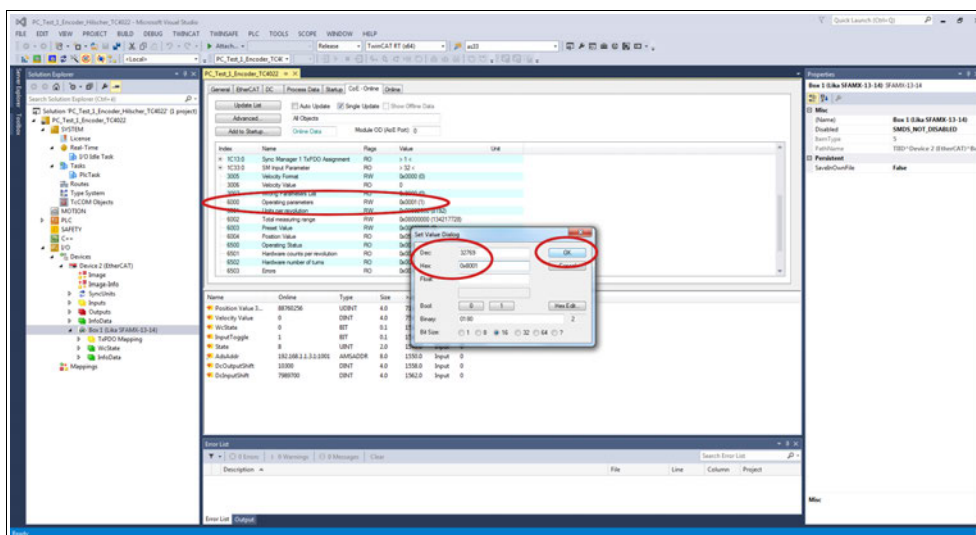


Figure 51 - Setting the Mask Upgrade Firmware bit

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 object **100A Manufacturer Software Version**.

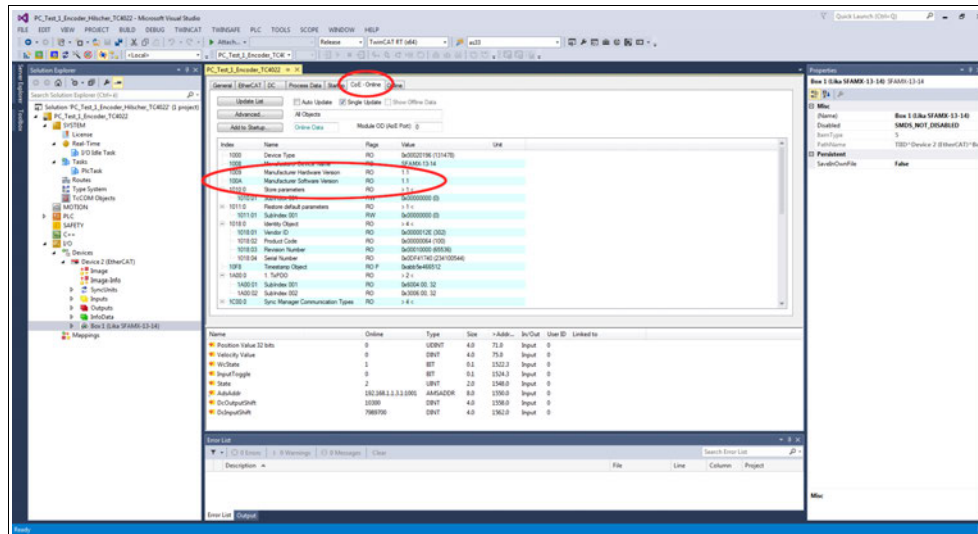


Figure 52 - Manufacturer Software Version

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

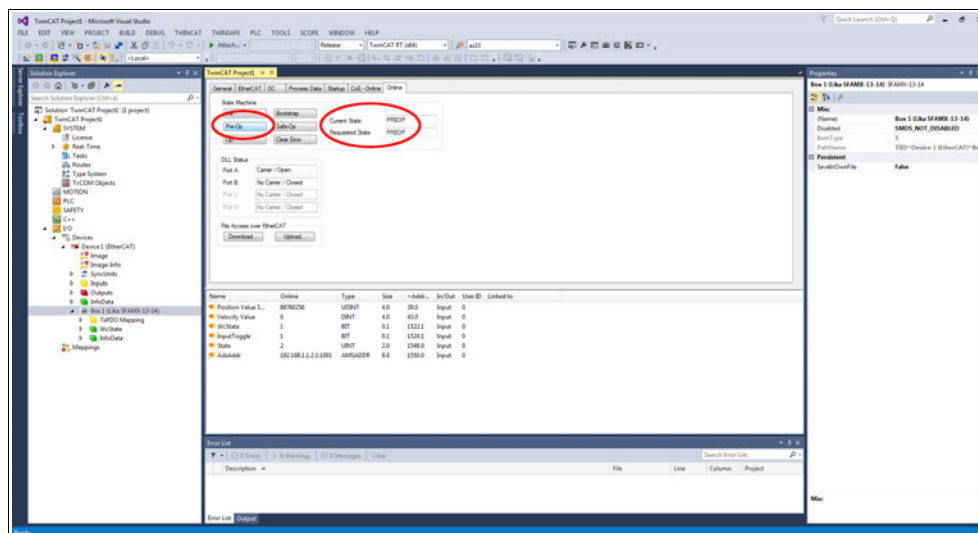


Figure 53 - Preoperational state



### 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).

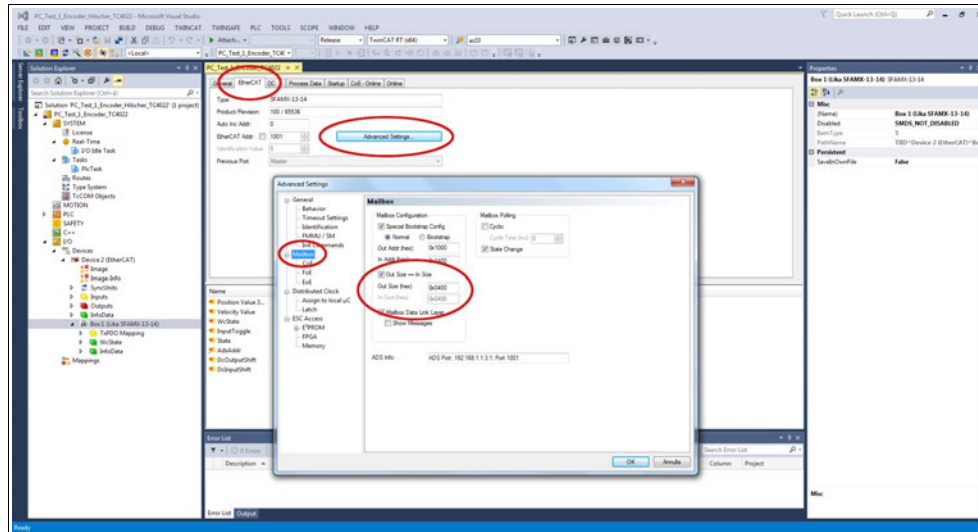


Figure 54 - Increasing the Mailbox sizes

For any specification on the FoE protocol, please refer to "ETG.1000 EtherCAT Specification" document available at the address [www.ethercat.org](http://www.ethercat.org). Please refer also to the "6.9 Firmware update" section on page 61.



## 8 – Integrated Web Server

### 8.1 Integrated web server – Preliminary information

EtherCAT encoders 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 set currently;
- to monitor the encoder;
- 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 encoder please ascertain that the following requirements are fully satisfied:

- the encoder is connected to the network;
- the encoder has 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 encoder is in the **Pre-Operational** or **Operational** or **Safe-Operational** states. Enter the **Online** tabbed page to check the current state of the encoder (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.

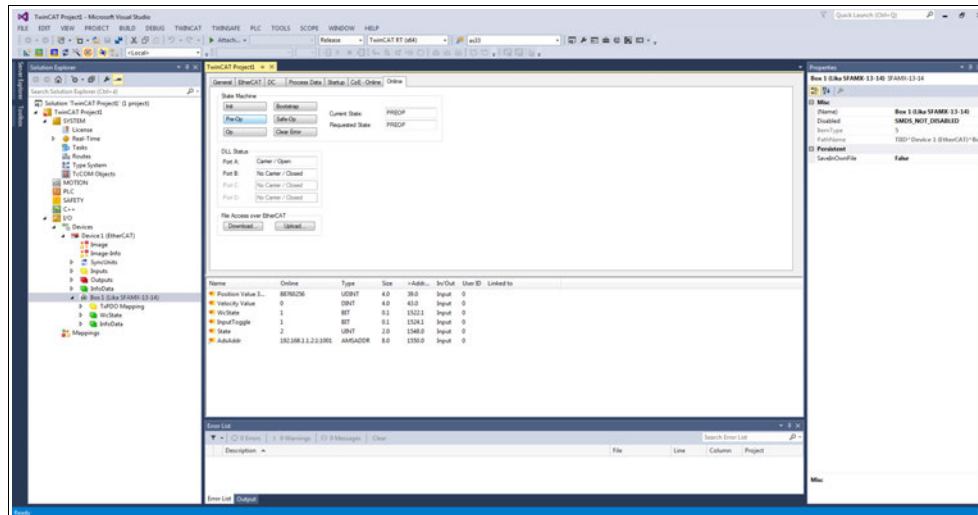


Figure 55 - Setting the State Machine

## 8.2 Web server Home page

To open the EtherCAT encoder web server, proceed as follows.

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

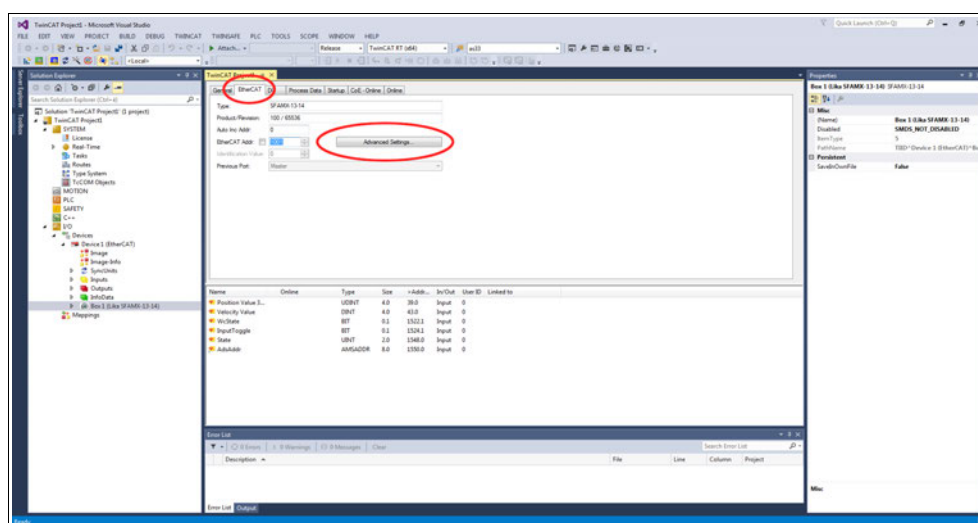


Figure 56 - EtherCAT tabbed page

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 (192.168.1.10 in Figure 57).

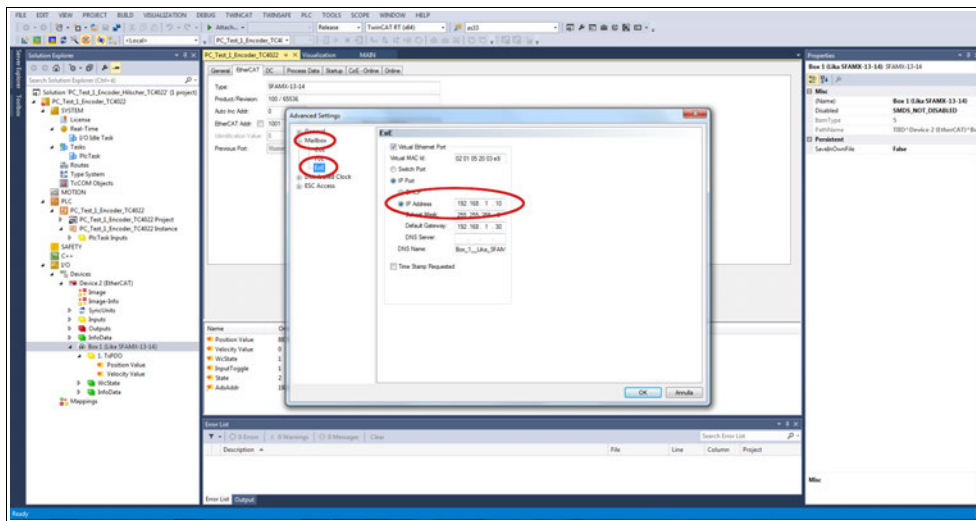


Figure 57 - Advanced Settings page

3. Type the IP address of the encoder you want to connect to (in the example: 192.168.1.10) in the address bar of your web browser and confirm by pressing **ENTER**.

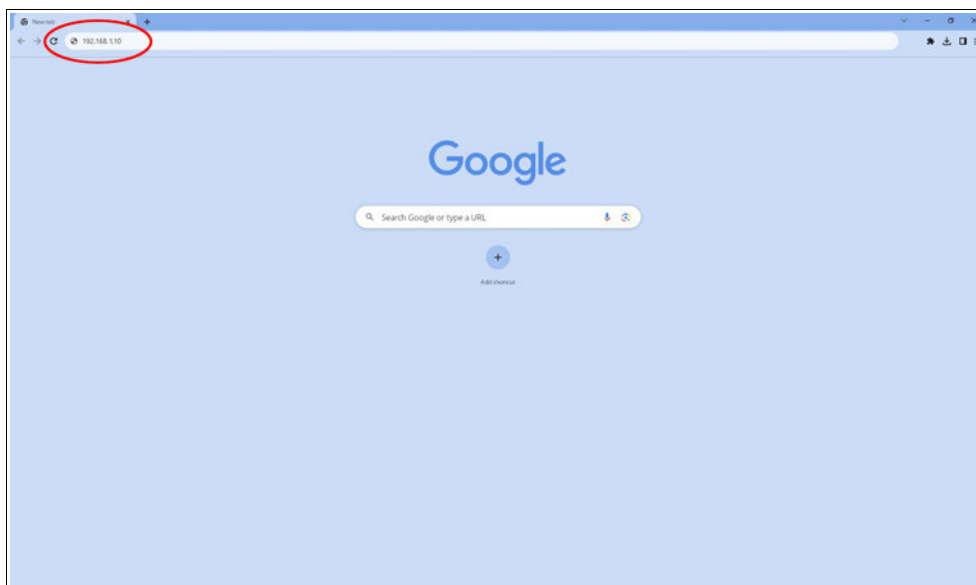


Figure 58 - Opening the web server

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

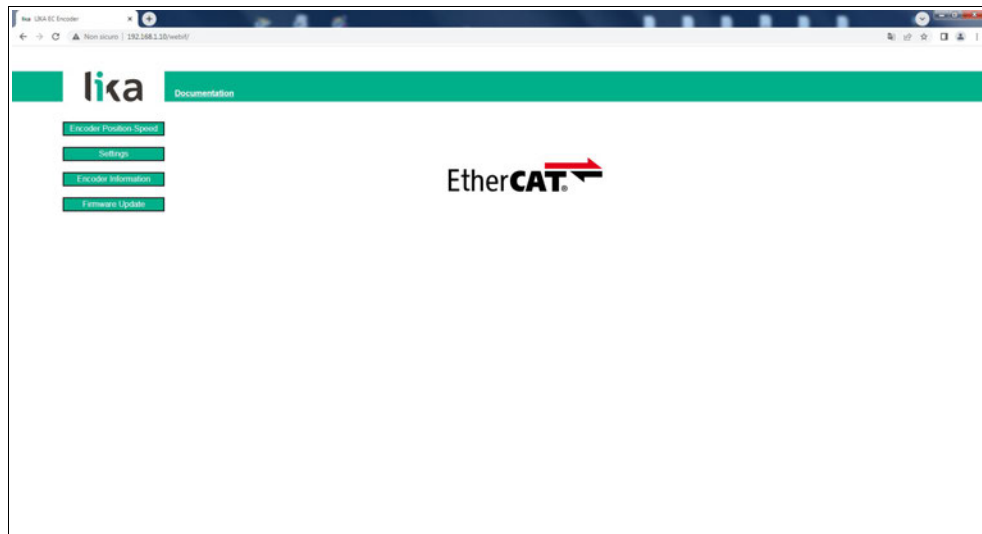


Figure 59 – Web server Home page

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](http://www.lika.biz)).

Press the **DOCUMENTATION** button to enter the EtherCAT encoder technical documentation page available on Lika's web site (<https://www.lika.it/eng/products/draw-wire-encoders/-1>) where specific technical information and documentation concerning the EtherCAT draw-wire encoder 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 encoder as well as useful functions can be achieved. They are described in the following sections.

### 8.3 Encoder position and speed

Press the **ENCODER 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.

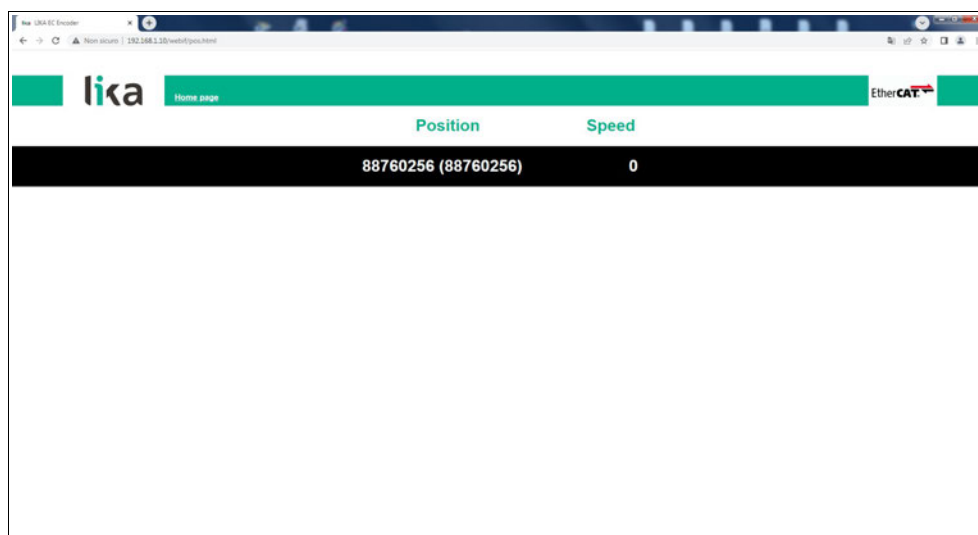


Figure 60 - Encoder position and speed page

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** object on page 99.

The current encoder speed (under the Speed item) is expressed according to the setting next the **3005-00 Velocity Format** object on page 88 (by default it is expressed in counts per second). For any information refer to the **3006-00 Velocity Value** object on page 88.



#### NOTE

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

Press the **HOME PAGE** 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 **Encoder 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 Encoder 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 encoder are displayed and their value can be changed.

For complete information on the encoder objects please refer to the "Manufacturer Specific Profile Area objects" on page 88; and to the "Standardised Profile Area objects (DS406)" section on page 91.

Parameter	Current Value	Action
Preset value	0	Send
Dir.Count.Toggle	1	Send
Scaling Control	0	Send
Units/Span	8192	Send
Total Meas. Range	134217728	Send
Velocity Format	0	Send
Load Default	0	Load
Save Param.		Save

Figure 61 - Page for setting the Encoder Objects

The values that are currently set in the encoder 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. The values have to be set in decimal notation.

For complete information on the available objects please refer to the "Manufacturer Specific Profile Area objects" on page 88; and to the "Standardised Profile Area objects (DS406)" section on page 91.



### EXAMPLE

The **6001-00 Units per revolution** object is currently set to **"8192"** (see the box next to the **Units/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 82.

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 encoder 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 82.

**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 Encoder information (EtherCAT objects)

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

Encoder information	
MAC address	00:00:00:00:00:00
Product Name	LIKA-10000
Product Serial Type	10000
Product Code	100
Serial Number	00000000
Hardware Version	1.0
Software Version	1.0
Network Firmware Ver.	00000000
Production Date	2021-10-13 00:04:00
Direction Counting Toggle	0-CW
Scaling Function Control	0-Disabled
Total Measuring Range	10000000
Measuring Units per Turn	1000
Physical Units	0
Measuring Error	0.000000
Operating Status	0000
Physical Resolution Type	0000
Number of Spots	1000
Alarm	0000
Supported Alarms	0000
Warnings	0000
Supported Warnings	0000
Offset Value	0
Writing Parameters List	0000

this page is not updated automatically

Figure 62 – Encoder Information page

For a complete description of the available encoder objects, please refer to the "Manufacturer Specific Profile Area objects" on page 88; and to the "Standardised Profile Area objects (DS406)" section on page 91.



### NOTE

Please note that the values shown in the **Encoder 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 **Encoder Information** page cannot be changed even though they are read-write access objects. To change the set values, please enter the **Set Encoder Objects** page (see on page 119).

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

The firmware update process must be carried out by skilled and competent personnel. It is mandatory to perform the update in accordance with the instructions provided in this section.

Before installation, always ensure that the firmware is compatible with the device's hardware and software. Furthermore, never turn off the power supply during the firmware update. In the event of an update error, the program will be irreversibly lost (as there is no bootloader), and the device will need to be returned to Lika Electronic for restoration.

This operation allows the unit firmware to be updated by downloading update data to the flash memory.

Firmware is a software programme that controls the functions and operation of a device. The firmware, sometimes referred to as the "user programme", is stored in the flash memory integrated within the unit. These encoders are designed so that the firmware can be easily updated by the user. This allows Lika Electronic to provide new and improved firmware programmes throughout the product's lifetime.

Typical reasons for releasing new firmware updates include the need to correct errors, enhance performance, and even add new functionalities to the device.

The firmware upgrade programme consists of a single file with a .ZIP extension. It is provided by Lika Electronic Technical Assistance & After-Sales Service.

If the latest firmware version is already installed on the unit, there is no need to proceed with a new firmware installation. The firmware version that is currently installed can be found next to the **Software revision** item on the **Encoder Information** page after connecting to the web server (see on page 121).



### NOTE

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

Before proceeding with the firmware update, please ensure that the following requirements are fully met.

- The encoder is connected to the Ethernet network.
- The encoder has a valid IP address.

- The PC is connected to both the network and the I/O controller.
- A web browser (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, etc.) is installed on the PC or device used for the connection.
- You have the .ZIP file for the 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 required to enter a password before starting the firmware update procedure.

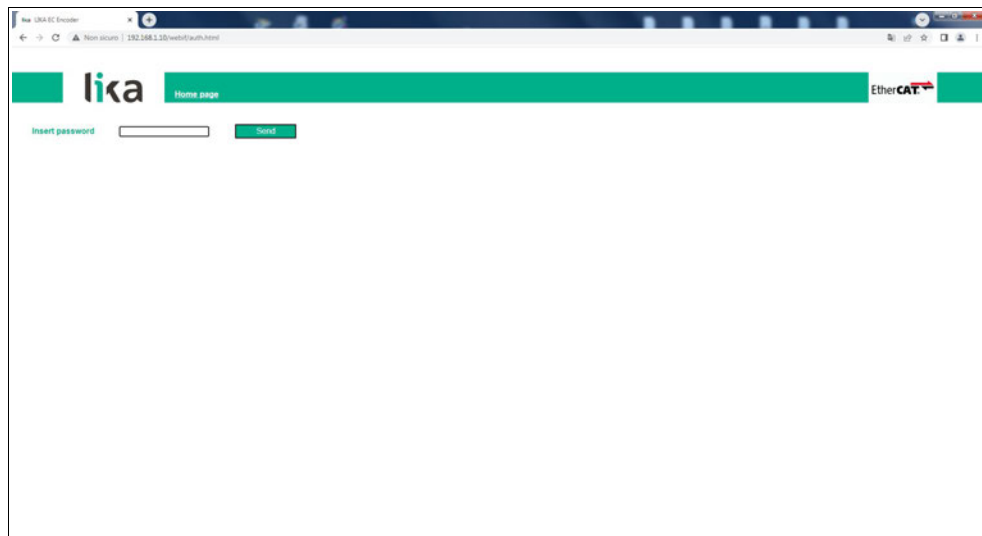


Figure 63 – Firmware Update page

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 entered is correct, the **Firmware Update** page will appear on the screen.

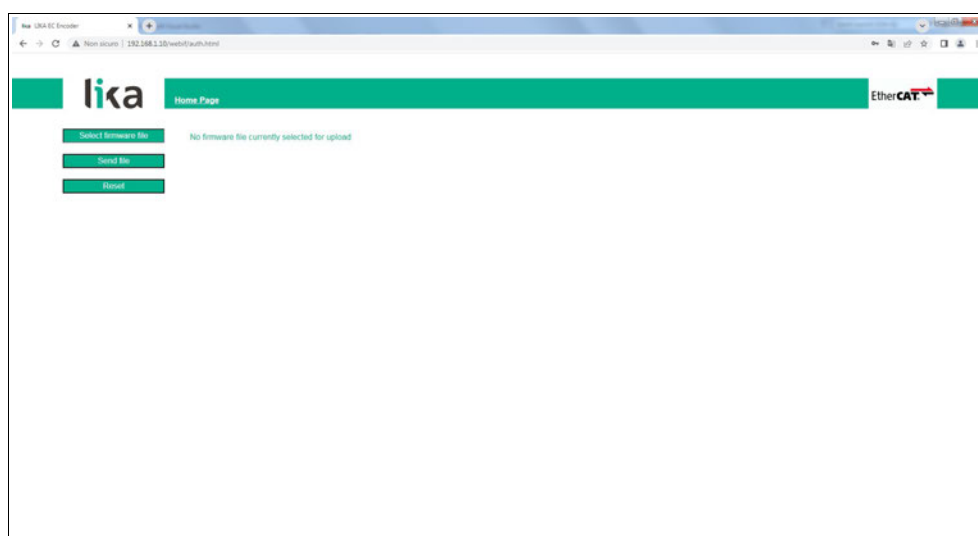


Figure 64 – Firmware Update page

6. Press the **SELECT FIRMWARE FILE** button. Once pressed, an **OPEN** dialog box will appear on the screen. Navigate to the folder containing the firmware update .ZIP file released by Lika Electronic, select the file, and confirm. Please check the file properties and ensure that you are installing the correct update file.

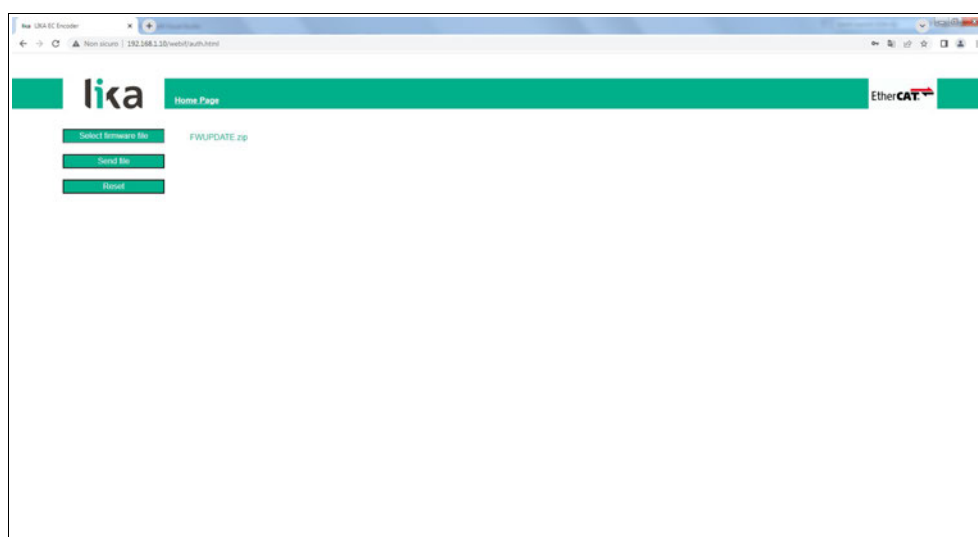


Figure 65 – Selecting the firmware update .zip file



## WARNING

Before installation, always ensure that the firmware is compatible with the device's hardware and software.

Never turn off the power supply during the firmware update.

7. Press the **SEND FILE** button to begin uploading the firmware program.
8. During the operation, and immediately after it is successfully completed, some messages will appear on the screen.

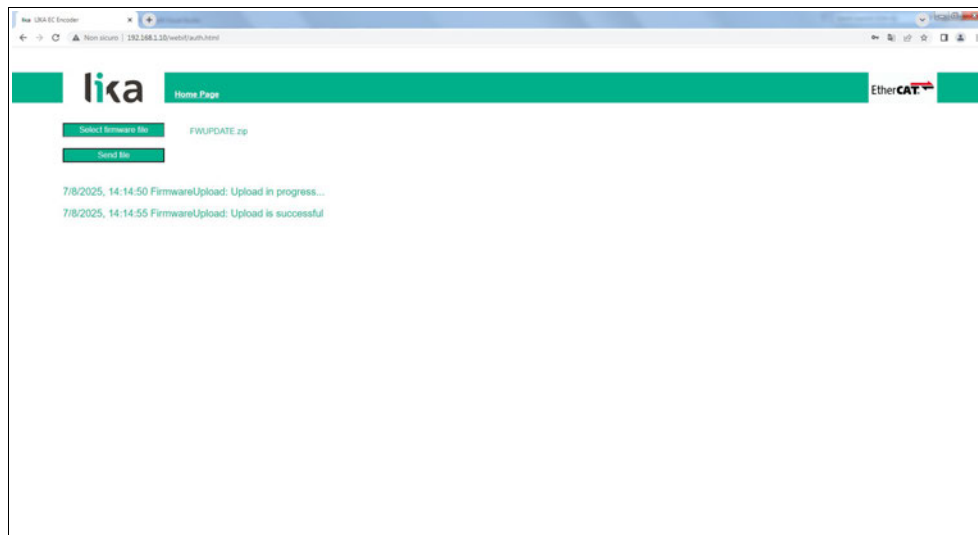


Figure 66 – Messages during firmware upload

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

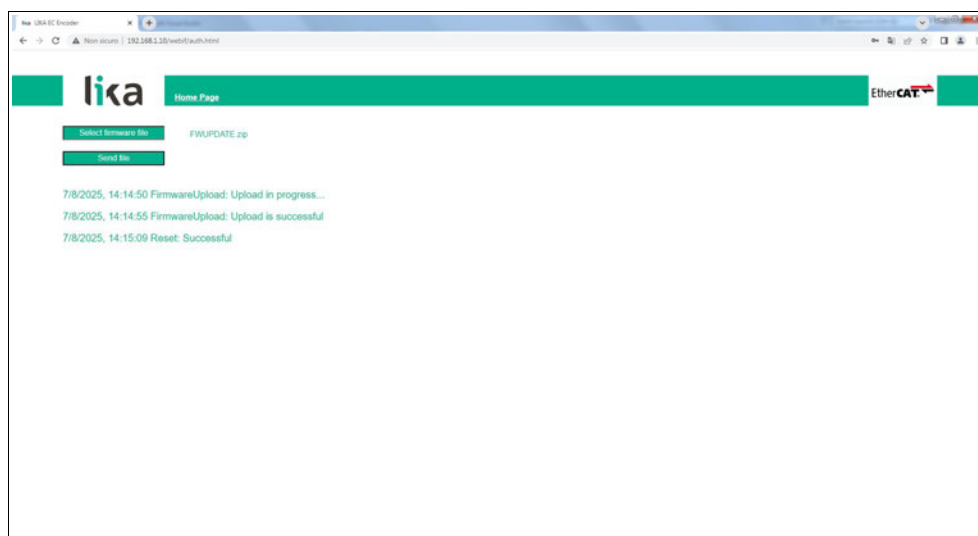


Figure 67 – Firmware update process accomplished

**NOTE**

While downloading the firmware update program, unexpected issues may arise that could cause the installation process to fail. If this occurs, the download cannot be completed successfully, and the operation will be aborted. In the event of a flash update error, please turn the encoder off and then back on before retrying 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		
<b>1000-00 Device Type</b>	0002 0196 = multiturn encoder		
<b>1008-00 Manufacturer Device Name</b>	5346414D582D31322D3134 = "SFAMX-13-14" = SFAMx 27 bit multiturn draw-wire encoder		
<b>1009-00 Manufacturer Hardware Version</b>	device dependent		
<b>100A-00 Manufacturer Software Version</b>	device dependent		
<b>1018 Identity Object</b>	4		
<b>01 Vendor ID</b>	0000 012E		
<b>02 Product Code</b>	0000 0096 = SFAMx 27 bit multiturn draw-wire encoder		
<b>03 Revision Number</b>	0001 0000 = Lika EtherCAT SFAMx series draw-wire encoder		
<b>04 Serial Number</b>	device dependent		
<b>1A00-00 TxPDO mapping parameter</b>	02		
<b>01 Mapped Object 001</b>	6004 0020		
<b>02 Mapped Object 002</b>	3006 0020		
<b>3005-00 Velocity Format</b>	0 = counts per second		
<b>6000-00 Operating parameters</b>	0001		
<b>Bit 0 Code sequence</b>	1 = count up information when pulling the cable out		
<b>Bit 2 Scaling function</b>	0 = disabled		
<b>Bit 15 Mask Upgrade Firmware</b>	0 = disabled		
<b>6001-00 Units per revolution</b>	0000 2000 (8,192 cpr)		
<b>6002-00 Total measuring range</b>	0800 0000 (134,217,728 counts)		
<b>6003-00 Preset Value</b>	00000 0000		
<b>6501-00 Hardware counts per revolution</b>	0000 2000 (8,192 cpr)		

6502-00 Hardware number of turns	0000 4000 (16,384 revolutions)		
-------------------------------------	-----------------------------------	--	--



This page intentionally left blank

This page intentionally left blank

This page intentionally left blank

Document release	Release date	Description	HW	SW	XML file version
1.0	10.02.2026	First issue	1.1	1.0	V1_3



**lika**

**Lika Electronic**

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

Tel. +39 0445 806600

Fax +39 0445 806699



info@lika.biz • www.lika.biz