



EtherCAT Application Manual

November 2012 (Ver. 1.403)

Notice

This guide is delivered subject to the following conditions and restrictions:

- This guide contains proprietary information belonging to Elmo Motion Control Ltd. Such information is supplied solely for the purpose of assisting users of the EtherCAT Application Manual in its installation.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice.



Elmo Motion Control and the Elmo Motion Control logo are registered trademarks of Elmo Motion Control Ltd.



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

Document no. G-ETHERCATAM (Ver. 1.403)

Copyright © 2012

Elmo Motion Control Ltd.

All rights reserved.

Revision History

Version	Date	Details
1.40	21/05/2012	Document completely revised. Initial new document
1.401	12/11/2012	Updated to include EtherCAT and CAN Object tables
1.402	27/11/2012	Corrections to EtherCAT explanation and objects
1.403	29/11/2012	Addition of EtherCAT switch details

Elmo Worldwide

Head Office

Elmo Motion Control Ltd.

60 Amal St., POB 3078, Petach Tikva 49516
Israel

Tel: +972 (3) 929-2300 • Fax: +972 (3) 929-2322 • info-il@elmomc.com

North America

Elmo Motion Control Inc.

42 Technology Way, Nashua, NH 03060
USA

Tel: +1 (603) 821-9979 • Fax: +1 (603) 821-9943 • info-us@elmomc.com

Europe

Elmo Motion Control GmbH

Hermann-Schwer-Strasse 3, 78048 VS-Villingen
Germany

Tel: +49 (0) 7721-944 7120 • Fax: +49 (0) 7721-944 7130 • info-de@elmomc.com

China

Elmo Motion Control Technology (Shanghai) Co. Ltd.

Room 1414, Huawen Plaza, No. 999 Zhongshan West Road, Shanghai (200051)
China

Tel: +86-21-32516651 • Fax: +86-21-32516652 • info-asia@elmomc.com

Asia Pacific

Elmo Motion Control

#807, Kofomo Tower, 16-3, Sunae-dong, Bundang-gu, Seongnam-si, Gyeonggi-do,
South Korea

Tel: +82-31-698-2010 • Fax: +82-31-698-2013 • info-asia@elmomc.com

Table of Contents

Chapter 1: Introduction	6
1.1. What is EtherCAT?	6
1.2. Elmo EtherCAT	8
1.3. Elmo Slave Drives.....	9
1.3.1. CTT (Conformance Test Tool)	10
1.4. Elmo G-MAS Master	11
1.4.1.1. For EAS	13
1.4.2. G-MAS Operation Modes	13
1.4.2.1. NC Axes Motions.....	13
1.4.2.2. Distributed / Standard DS-402 (stand-alone) Drive.....	14
1.4.2.3. G-MAS to Servo Drive Interfaces	14
1.5. ELMO Application Studio (EAS) – Configuration Tool	15
Chapter 2: Elmo EtherCAT Slave Device	16
2.1. Indicators	16
2.1.1. Drive status LED Indicator	16
2.1.2. EtherCAT Link/Activity Indicators	16
2.1.3. EtherCAT Status Indicator.....	17
2.1.4. EtherCAT Address Switches (where present on servo drive)	18
2.2. CoE – CANopen Over EtherCAT	19
2.2.1. PDO (Process Data Object)	19
2.2.1.1. Receive PDO Mapping (Outputs).....	20
2.2.1.2. Transmit PDO Mapping (Inputs)	22
2.2.2. Emergency Requests.....	24
2.3. Synchronization Modes	25
2.3.1. Free Run.....	25
2.3.2. Distributed Clocks (DC Mode) - Synchronous with SYNC0	27
2.4. EoE – Ethernet Over EtherCAT	29
2.4.1. EoE Extensions	30
2.4.1.1. MAC Address Info	30
2.4.1.2. IP Address Info	30
2.5. FoE – File Access over EtherCAT	31
2.6. EEPROM	32
2.7. ESI (xml format)	34
Chapter 3: G-MAS Communication	35
3.1. EAS EtherCAT Quick Configuration.....	35
3.2. Downloading Firmware to G-MAS using FoE.....	44
3.3. G-MAS EoE Configuration.....	44
Chapter 4: TwinCAT Communication	47
4.1. Architecture.....	47

4.2.	Using TwinCAT Master	47
4.3.	Using a Switch Port.....	48
4.4.	Download Firmware using FoE (via TwinCAT)	49
4.4.1.	Setup Using the TwinCAT NC/PTP System Manager	49
4.4.2.	Setup Procedure	49
4.4.3.	The Firmware Download Procedure.....	58
Chapter 5: Gold Drive Object list		61
5.1.	Complete Object List	61
5.2.	CAN Only Object List.....	67
5.3.	EtherCAT Only Object List.....	68
5.4.	EtherCAT CoE - PDO Objects list.....	69
5.4.1.	Transmit PDOs object list.....	69
5.4.2.	Receive PDOs object list	71
Chapter 6: Elmo Emergency Error and Abort List.....		73
6.1.	Emergency Error Description.....	73
6.2.	Abort SDO Transfer Protocol	77

Chapter 1: Introduction

The ELMO EtherCAT environment is extensive and abounding in features. ELMO provides a comprehensive solution for the EtherCAT system, which includes the G-MAS Network Motion Controller EtherCAT Master, Servo Drives, EtherCAT Slaves and EtherCAT configuration tools. ELMO EtherCAT is standard compliant and is successfully EtherCAT conformance tested.

This manual describes the installation, setup, range of functions and EtherCAT protocols for Elmo's EtherCAT-ready products.

1.1. What is EtherCAT?

Ethernet for Control Automation Technology (EtherCAT) is an open high performance Ethernet-based fieldbus system, which uses the family of industrial computer network protocols used for real-time distributed control, now standardized as IEC 61158. It is a highly flexible Ethernet network protocol, running over a fast real time Master–Slave network.

The EtherCAT communication speed is up to 100 Mbps full duplex and can include a maximum of 65,535 stations in a single network configuration such as Ethernet star, line or tree without using switches.

Figure 1 describes a network of EtherCAT slaves in a ring topology. The Master controls the traffic in the network by initiating the transactions.

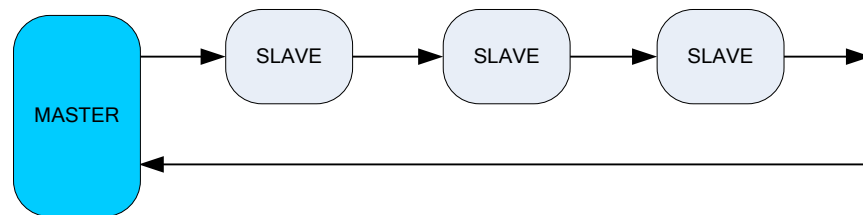


Figure 1: EtherCAT Network Configuration

Usually, a control system requires the following in periodic time intervals:

- **Inputs**
Messages from the ECAT device to the Master latches data such as Positions, Velocities, Currents, System Status, IO's etc,
 - **Outputs**
Messages from the ECAT Master to the device with data or commands such as Control word commands, Trajectory Information (set point), or Higher Drive Level Commands.
- The specific nature of the data transferred via the network depends on the operation mode of the slave drive. The Device Profile describes the application parameters and the functional behavior of the devices including the device class-specific state machines. A common standard for the Servo

Drive is the DS-402 for Drive and Motion Control device Profile, which can be addressed via CoE (Can Over EtherCAT).

The EtherCAT protocol is optimized for process data and is transported directly within the standard IEEE 802.3 Ethernet frame. Each Ethernet frame can include several EtherCAT frames, each serving another slave.

EtherCAT network uses a processing on the fly, whereby the Ethernet frame is received and processed, while the telegram passes through the device. The frames only delay by a fraction of a microsecond in each node. Using EtherCAT, the entire network can be addressed with just one frame.

The data sequence is independent of the physical order of the nodes in the network; addressing can be in any order. Broadcast, multicast and communication between slaves are possible and must be performed by the master device.

The EtherCAT protocol can be inserted into UDP/IP datagrams. This also enables any control with an Ethernet protocol stack to address EtherCAT systems.

Using the Master configuration tool, the Master scans the EtherCAT network and uses the EtherCAT Slave library (ESI: EtherCAT Slave Information in XML format) to compare the slave memory area that includes information about the slave such as Vendor ID, Product Code, and Slave Configuration.

1.2. Elmo EtherCAT

For a complete motion solution, the ELMO environment comprises of three levels:

- **EAS**; EtherCAT configuration tools
- **G-MAS**; EtherCAT G-MAS master
- **Gold Servo Drive**; Elmo EtherCAT slave drives

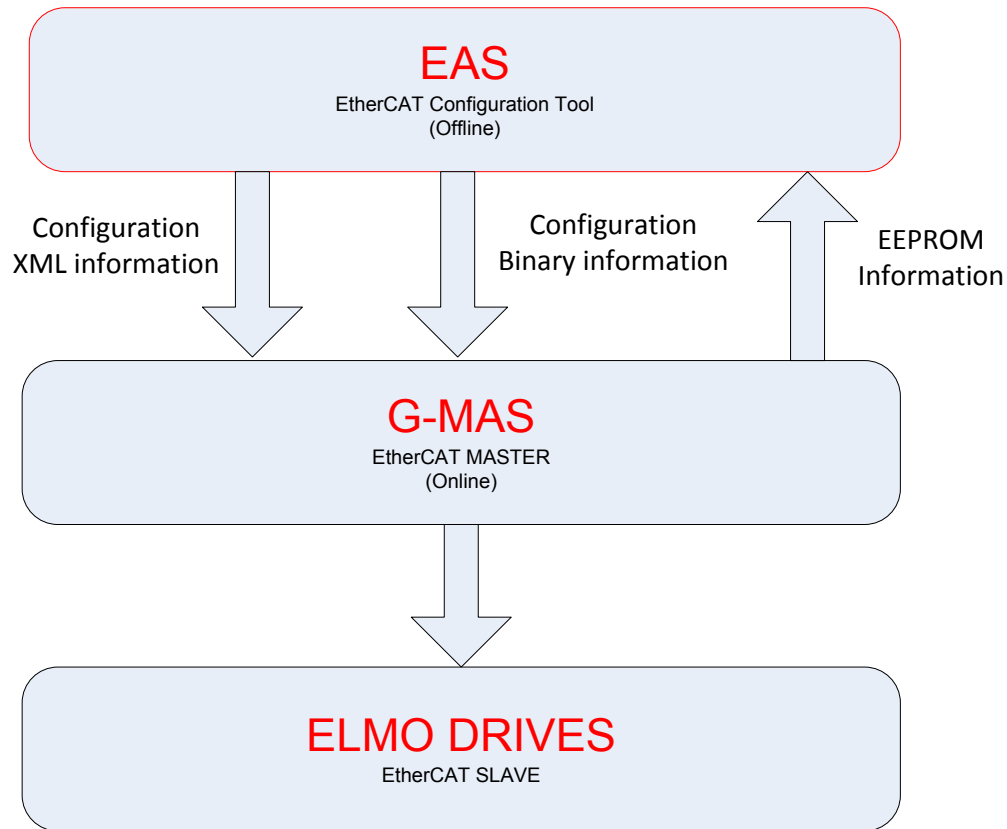


Figure 2: EtherCAT Environment

1.3. Elmo Slave Drives

The following diagram describes the EtherCAT communication of the drive.

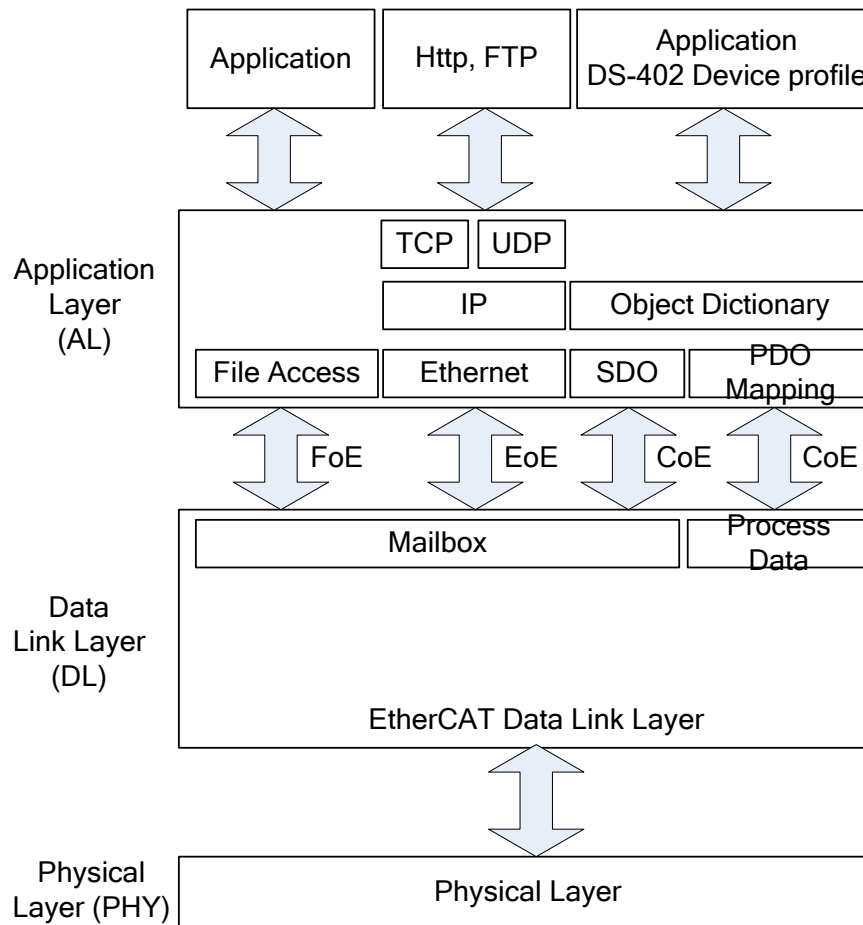


Figure 3: Layered Communication protocol in EtherCAT

Physical Layer

The Physical layer of the EtherCAT is a 100Mbps/sec Ethernet port over twisted per cable.

Data Link Layer

Supports two mechanisms of data transfer:

- Process data**
 Allows writing and reading data simultaneously. This mode is used to transfer the Process data objects (PDO). The PDO transfers via SYNC Manager 2 (rPDO) and SYNC Manager 3 (tPDO)
- Mailbox**
 The mailbox mechanism assures that the data will reach to the target without overlapping previous data. The mailbox is used to transfer the SDOs. The SDO transfers via SYNC Manager 0 (MailboxOut) and SYNC Manager 1 (MailboxIn). SDO objects are used for user triggered access.

With SDO services, all of the OD's entries can be accessed. The SDO transport works in asynchronous mode.

The Elmo drive supports the following communication protocols:

CoE (CANopen over EtherCAT)	Defines a standard way to access the CANopen protocol and includes an object dictionary, SDO, PDO and emergency messages.
EoE (Ethernet over EtherCAT)	Fully Ethernet compatible, defines a standard way to exchange or tunnel standard Ethernet frames. This is typically used to address the drive inherent parameters such as control parameters
FoE (File over EtherCAT)	Similar to TFTP, defines a standard way to download firmware to the servo drive.

The **Object Dictionary (OD)** contains parameters, application data and the mapping information between the Process Data Interface (PDI) and application data. Its entries can be accessed via the Service Data Object (SDO).

An Object Dictionary is a naming system that provides a unique identifier to each data item or "object" communicated over the CoE protocol. An object is identified by an index and sub-index. It contains variables, arrays and complex objects. CoE and EoE protocols require a set of mandatory objects. Elmo's OD is compliant with the DS402 V3 object list for Drive and Motion control device. The DS402 defines standard objects for the following motion modes; Profile Position, Profile Velocity, Profile Torque, Homing mode, Synchronous Cyclic Position, Synchronous Cyclic Velocity and Synchronous Cyclic Torque.

Elmo drive supports distributed clock in order to **synchronize** between the Master and Slaves on the EtherCAT network. Refer to the chapter 2.3 Synchronization Modes.

1.3.1. CTT (Conformance Test Tool)

The CTT (Conformance Test Tool) is the official tool used by the EtherCAT Technology Group (ETG) for EtherCAT conformance certification. The CTT includes thousands of tests to verify that the device complies with the EtherCAT definition, requirements and standard. The Gold drive firmware version has passed the CTT in several quality assurance (QA) stations, making sure that the ETG Certificate given to Elmo is valid.

1.4. Elmo G-MAS Master

While a single servo drive can run as a stand-alone drive using its inner profiler and filter, in order to perform synchronized multi axis motions in the system (such as circle, line etc.), a real time communication protocol must be used, and all drives must be synchronized to a specific SYNC signal in the system. The Gold Maestro Network Motion Controller (G-MAS) performs this task and operates as a master, independent of any host system. In operational mode, it periodically sends data to the slaves that may override the data that a user sends from a host system. Therefore, for example, the user cannot tune an axis if the axis is in operational mode.

Figure 4 describes the G-MAS general communication and API architecture. This diagram describes two types of communication mechanisms:

- Inter-process communication (IPC) which is a set of techniques to exchange data among multiple threads in one or more processes. C Programs located on the GMAS use the IPC mechanism to communicate, and the GDS can communicate directly with the G-MAS via C programs.
- A remote procedure call (RPC) is an inter-process communication allowing a program to initiate a subroutine or procedure to execute in another address space (the G-MAS server) without the programmer explicitly coding the details for this remote interaction. The EAS application uses RPC to communicate with the GMAS.

Both these mechanisms allow the GDS and Elmo Application Studio (EAS) to communicate using TCP/IP and EtherCAT over TCP/IP, with the G-MAS to perform master operations.

Important: It should be noted that connecting to the G-MAS is only allowed using one user application at a time. Connecting two applications to the G-MAS in parallel may cause serious problems to the G-MAS library. When performing multiple IPC connections to the G-MAS, the multiple connections must be opened from the same user application.

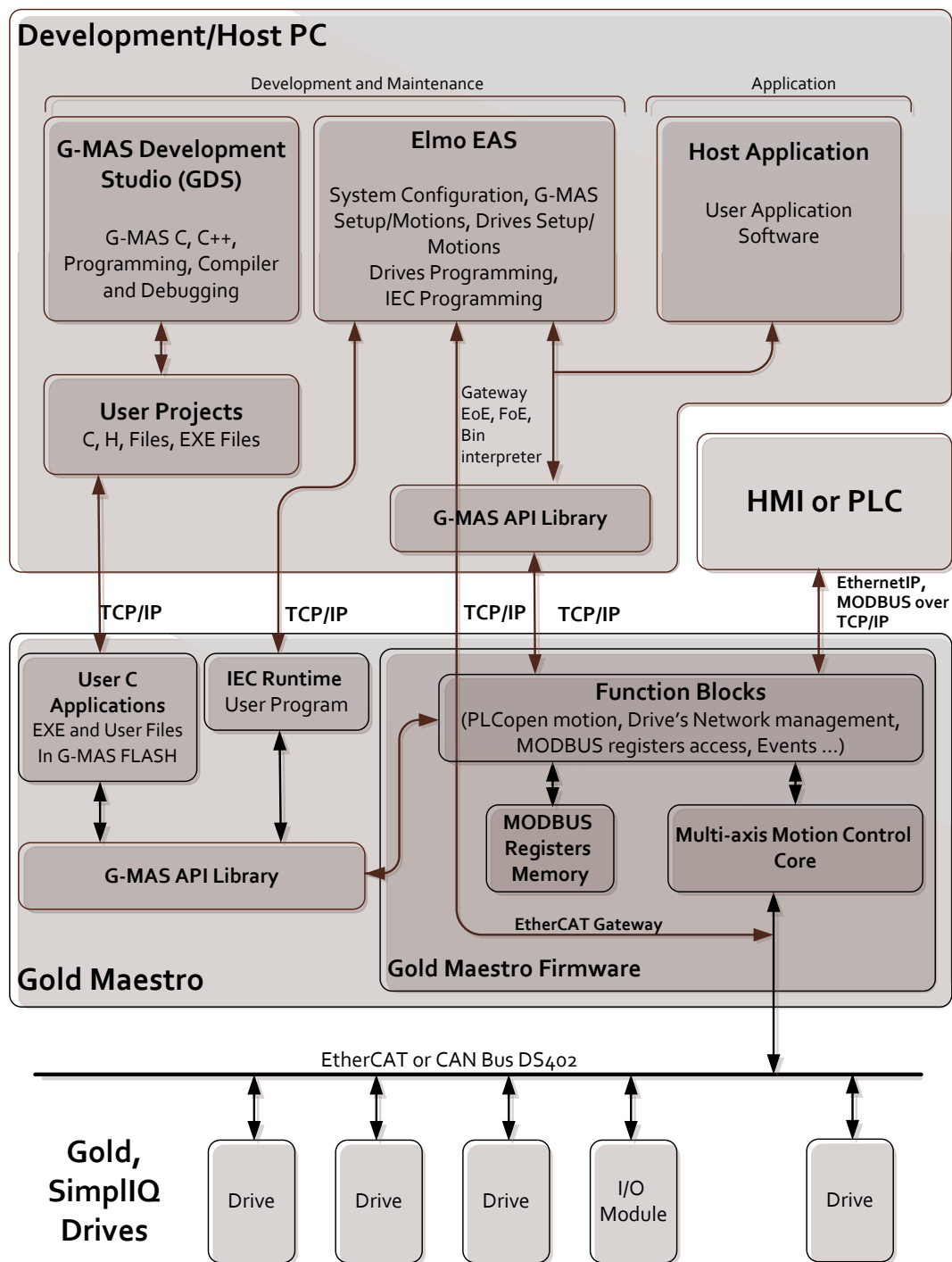


Figure 4: GoS System Software Structure –Host PC Development

In addition, since the G-MAS operates as a master, independent of any host system, in operational mode, it periodically sends data to the slaves, that may override the data that a user sends from a host system. Therefore, for example, the user cannot tune an axis if the axis is in operational mode. To prevent this and allow the EAS application (Figure 4) to operate via the G-MAS CANbus and

EtherCAT, specific API functions are called to change the G-MAS operation and allow these applications to function.

1.4.1.1. For EAS

For the EAS application to monitor and perform motions, the G-MAS cannot operate in the background. A special API function changes the EtherCAT and CANbus communication from the G-MAS to Pre-Operation mode, causing the following:

EtherCAT **No process cycle operates**

CANbus **No outputs via CAN, and no state machines run**

These API functions change the GMAS mode so that the GMAS operation is transparent and no messages transfer between the GMAS and the drives.

In order to configure the EtherCAT network (EtherCAT Configuration Mode) via the EAS application, the G-MAS must be set to EtherCAT Configuration mode. The user is then able to perform the operations. The API then employs the specific functions to change the G-MAS back to operational mode.

1.4.2. G-MAS Operation Modes

To optimize the device network usage, the G-MAS supports two modes of operating axes present on the Device Network:

- NC Axes – for Numeric Control Axes
- Distributed – for axes not under strict numeric control

The main difference between these modes is the way the motion profile is calculated, and as a result, the synchronization level achieved.

In general, for axes not requiring low level (network) motion synchronization, the Distributed mode should be used, allowing the servo drives to generate their own motion trajectory, thus reducing network load. In this case, synchronized motions like ECAM, based on an external master encoder can still be executed. For highly synchronized motions, generated by the Master controller (referred to under the PLCopen definitions as group vector motions), the NC mode should be used.

1.4.2.1. NC Axes Motions

In this mode, the G-MAS controls the motion, handling the axis (and motion) State (as defined by the PLCopen Standard), and calculating the motion profile as part of its real-time loop process (NC Cycle). Servo drives operating with a G-MAS master under this mode will run under the DS-402 motion modes e.g.; Interpolated position, or one of the Cyclic Sync modes (Position/Velocity).

1.4.2.2. Distributed / Standard DS-402 (stand-alone) Drive

In this mode, the G-MAS uses the servo drives own DS-402 operation modes, where the drive itself controls its own profiling as part of its Real Time process. The G-MAS only synchronizes start/stop and general activation functions, but is not responsible to the low-level real-time profile generation.

The G-MAS can mix NC and Distributed axes in the same network configuration, thus optimizing usage of network and processor resources. The definition of the axis type (NC or Distributed), can be changed during operation using an operation mode command.

1.4.2.3. G-MAS to Servo Drive Interfaces

The G-MAS manages all motion commands sent to the servo drives, via the CANopen DS-402 standard (Refer to Figure 4). This is relevant to the G-MAS CAN hardware interface, and to the EtherCAT protocol implementing CoE (CAN Over EtherCAT).

For axes (Nodes) that operate in NC mode, G-MAS uses the DS-402 motion modes: Interpolated position, or one of the Cyclic Sync modes (Position/Vel).

For axes (Nodes) operating in Distributed mode, where the servo drive manages its own profiler and real-time motion execution, it is assumed that the servo drive supports the relevant requested motion modes.

Motion Modes that are part of the PLC Motion API definition, but are **not** supported by the DS-402 interface, will not be available in standard DS-402 servo drives when working in Distributed mode (unless specific Vendor Types objects are defined, e.g. ECAM in drive level, etc. as implemented for example in Elmo servo drives).

The G-MAS uses the EtherCAT communication protocol to enable synchronized motion of all the controllers to the same SYNC signal. Thus, all drives in the system are synchronized to the master clock, and all generate an interrupt at exactly the same time.

A profiler can run in the G-MAS, on the condition that the axis (axes) is defined as a vector axis (axes). A vector axis may consist of 1 - 16 axes. The Multi Axis Indexer (MAI) is the profiler that runs within the G-MAS, which sends (via a high priority interrupt routine) a calculated set point to the axes in the system and can perform vector calculations for up to 16 axes. The profiler EtherCAT outputs are points that are to be sent to the specific drives belonging to the vector. Therefore, a number of combination options are available:

- 1 x 16 axes (One vector profiler performing profiles for 16 axes),
- 16 x 1 axes (16 profilers for 16 vector axes), or,
- Any combination of M x N axes – as long as $M \times N < 16$

The SYNC interrupt signal to the drives is based on the ET1100 component in the servo drive. The master G-MAS does not receive this signal, but can calculate when the SYNC signal is generated. This is because the master EtherCAT in the G-MAS is responsible for updating the SYNC cycle time

in the servo drives, and therefore knows when the SYNC is generated. The MAI can operate at varying cycle times, dependent on a number of parameters, such as the:

- Desired response from the system
- Number of axes participating in the MAI. The more axes, the higher the cycle rate

1.5. ELMO Application Studio (EAS) – Configuration Tool

The EtherCAT configuration tools enable configuration and monitoring of the network from the EAS application.

Using the Master configuration tool, the Master scans the EtherCAT network and uses the EtherCAT Slave library to compare the slave memory area that includes information about the slave such as Vendor ID, Product Code, and Slave Configuration.

Chapter 2: Elmo EtherCAT Slave Device

2.1. Indicators

This section describes the drive and EtherCAT indicators used to support visual inspection and troubleshooting of the drive and networks.

2.1.1. Drive status LED Indicator

This LED is used for immediate indication of the drive status. The LED color varies between green or red. The LED states are defined in the table below.

Indicator state/color	Definition
Off	No power supply to drive
Temporary blinking red and then Off	The drive is in BOOT state. Firmware download is required and should be performed now.
Red	The drive is in fault state e.g. Safety switches or Low bus voltage)
Green	Drive is ready

2.1.2. EtherCAT Link/Activity Indicators

Each EtherCAT slave device includes two RJ-45 connectors; EtherCAT IN and EtherCAT OUT as shown in Figure 4:



Figure 5: EtherCAT ports

Figure 5 describes the two status LEDs for each RJ-45 connector. The link/activity indicators show the state of the physical link and activity on this link.

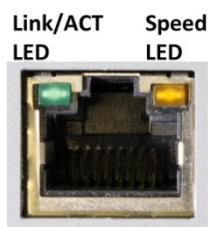


Figure 6: Status LEDs

Table 1 displays the LEDs' link and activity.

LED	State		Definition
Link and/or Activity	Off	No Connection	Green color, defines the state of the physical link/activity of the link.
	On	Connection Established (Link)	
	Flashing	Data transmission active (Act)	
Speed	On	100Mbps Connection (default) otherwise there is no EtherCAT connection	Orange color, define the speed of the EtherCAT line.
	Off	10Mbps connection	

Table 1: LED Functionality

2.1.3. EtherCAT Status Indicator

This indicator is one bi-colored led that combines the green RUN indicator (Table 2: RUN Indicator States) and the red ERROR indicator (Table 3: ERROR Indicator States) of the EtherCAT state machine. The LED indicator signals are based on ETG1300 of which Elmo drives are fully certified.

Indicator State	ESC State	Description
Off	Initialization	Elmo drive is in state INIT
Blinking	Pre-Operational	Elmo drive is in state Pre-OP
Single Flash	Safe-Operational	Elmo drive is in state Safe-OP
On	Operational	Elmo drive is in state OP
Flickering	Initialization or Bootstrap	Elmo device is booting and has not yet entered the INIT state, or Elmo device is in state Bootstrap. Firmware download operation in progress
Triple Flash	Device Identification	User sets this state from the master to locate the specific slave

Table 2: RUN Indicator States

Error State	Error Name	Description
On	Application controller failure	An critical communication or application controller error has occurred
Double Flash	Process data watchdog timeout / EtherCAT watchdog timeout	An application watchdog timeout occurred
Single Flash	Local error	Elmo device application has changed the EtherCAT state autonomously, due to local error.
Blinking	Invalid configuration	General configuration error
Flickering	Booting error	Booting error was detected
Off	No error	The EtherCAT communication of the device is in working condition.

Table 3: ERROR Indicator States

2.1.4. EtherCAT Address Switches (where present on servo drive)

The EtherCAT address of each axis is specified by two switches. The position of the EtherCAT switches, may either be found, inside the battery compartment as in the Gold Duet, or in other positions on the drive or drivers interface. Use a fine screwdriver to set the low and the high bytes of the EtherCAT address.

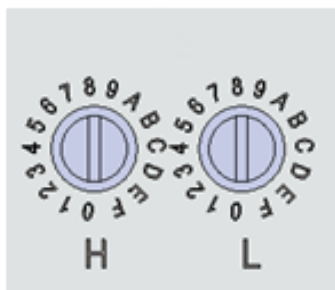


Figure 7: EtherCAT Address Switches

2.2. CoE – CANopen Over EtherCAT

Defines a standard way to access the CANopen protocol and includes an object dictionary, SDO, PDO Emergency and Abort messages.

2.2.1. PDO (Process Data Object)

The PDO protocol is used for communication with SYNC Manager 2 for RxPDO, and SYNC Manager 3 for TxPDO.

Each PDO consist of objects in the object dictionary, which is PDO map able. The PDO mapping objects describes how these objects are related to a PDO.

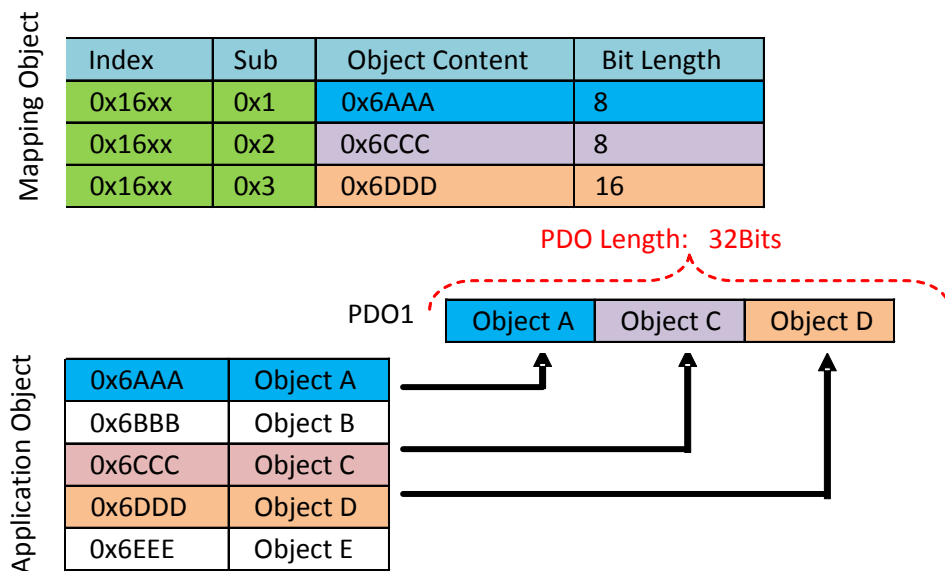


Figure 8: PDO Mapping

Each sync manager channel object describes a consistent area inside the EtherCAT process data and consists of several process data objects.

All PDO mappings are located in the object dictionary from index 0x1C10 to 0x1C2f.

2.2.1.1. Receive PDO Mapping (Outputs)

This defines a list of objects that include data from the master to the slave. All RPDOs are located in the object dictionary from index 0x1600 to 0x17FF.

Sub-index	Description	Data Type	Access	PDO Mapping	Value
0	Number of objects in this PDO	Unsigned8	RW, Mandatory	No	0-254 Writable if variable mapping is supported
1	First output object to be mapped	Unsigned32	R, depends upon setting	No	Bit 0-7: length of the mapped objects in bits. Bit 8-15: sub index of the mapped object. Bit 16-31: index of the mapped object
.....					
N	Last output object to be mapped	Unsigned32	R, depends upon setting	No	

Table 4 : Receive PDO Mapping Configuration

The following table describes Elmo RxPdo's objects:

PDO Index	Default Value	Bit Len	Description	Function Group	Exclude
0x1600	0x607A 0x60FE:1 0x6040	32 32 16	Target Position Digital Inputs Control Word	Position	0x1601-0x1606
0x1601	0x60FF 0x6040	32 16	Target Velocity Control Word	Velocity	0x1600, 0x1602-0x1606
0x1602	0x6071 0x6040	16 16	Target Torque Control Word	Torque	0x1600-0x1601 0x1603-0x1606
0x1603	0x607A 0x60FE:1 0x60B1 0x6040	32 32 32 16	Target Position Digital Outputs Velocity Offset Control Word	Position	0x1600-0x1602 0x1604-0x1606
0x1604	0x607A 0x60FF 0x6072 0x6040	32 32 16 16	Target Position Target Velocity Max. Torque Control Word	Position, Velocity	0x1600-0x1603 0x1605-0x1606

PDO Index	Default Value	Bit Len	Description	Function Group	Exclude
0x1605	0x607A 0x60FF 0x6071 0x6072 0x6040 0x6060	32 32 16 16 16 8	Target Position Target Velocity Target Torque Max. Torque Control Word Mode Of Operation	Position, Velocity, Torque	0x1600-0x1604 0x1606
0x1606	0x607A 0x60FE:1 0x60FF 0x60B1 0x60B2 0x6040	32 32 32 32 16 16	Target Position Digital Outputs Target Velocity Velocity Offset Torque Offset Control Word	Position, Velocity	0x1600-0x1605
0x160A	0x6040	16	Control Word		
0x160B	0x6060	8	Mode Of Operation		
0x160C	0x6071	16	Target Torque		
0x160D	0x6072	16	Max. Torque		
0x160E	0x6073	16	Max. Current		
0x160F	0x607A	32	Target Position		
0x1610	0x607F	32	Max. Profile Velocity		
0x1611	0x6081	32	Profile Velocity		
0x1612	0x6082	32	End velocity		
0x1613	0x6083	32	Profile Acceleration		
0x1614	0x6084	32	Profile Deceleration		
0x1615	0x6087	16	Torque Slope		
0x1616	0x60B0	32	Position Offset		
0x1617	0x60B1	32	Velocity Offset		
0x1618	0x60B2	16	Torque Offset		
0x1619	0x60B8	16	Touch Probe Function		
0x161A	0x60C1:1	32	Interpolated data record (1)		
0x161B	0x60C1:2	32	Interpolated data record (2)		
0x161C	0x60FF	32	Target Velocity		
0x161D	0x60FE:1	32	Digital Output		
0x161E	0x607F	8	Polarity		

Table 5: RxPDO Default Values

2.2.1.2. Transmit PDO Mapping (Inputs)

This defines a list of objects that include data from the slave to the master. All TPDOs are located in the object dictionary from index 0x1A00 to 0x1BFF.

Sub-Index	Description	Data Type	Access	PDO Mapping	Value
0	Number of objects in this PDO	Unsigned8	RW, Mandatory	No	0-254 Writable if variable mapping is supported
1	First output object to be mapped	Unsigned32	R, depends upon setting	No	Bit 0-7: length of the mapped objects in bits. Bit 8-15: sub index of the mapped object. Bit 16-31: index of the mapped object
...					
N	Last output object to be mapped	Unsigned32	R, depends upon setting	No	

Table 6 : Transmit PDO Mapping Configuration

The following table describes Elmo TxPDO's objects:

PDO Index	Default Value	Bit Len	Description	Function Group	Exclude
0x1A00	0x6064 0x60FD 0x6041	32 32 16	Position Actual Value Digital Inputs Status Word	Position	0x1A01-0x1A04
0x1A01	0x6064 0x606B 0x6074 0x6041	32 32 16 16	Position Actual Value Velocity Demand Value Torque Demand Value Status Word	Position	0x1A00, 0x1A02-0x1A04
0x1A02	0x6064 0x6077 0x6041 0x6061	32 16 16 8	Position Actual Value Torque actual Value Status Word Mode Of Operation Display	Position, Torque	0x1A00-0x1A01, 0x01A03-0x1A04
0x1A03	0x6064 0x60FD 0x606C 0x6041	32 32 32 16	Position Actual Value Digital Inputs Velocity Actual Value Status Word	Position, Velocity	0x1A00-0x1A02, 0x1A04

PDO Index	Default Value	Bit Len	Description	Function Group	Exclude
0x1A04	0x6064	32	Position Actual Value	Position, Torque	0x1A00-0x1A03
	0x60F4	32	Position Following error Value		
	0x6077	16	Torque Actual Value		
	0x6041	16	Status Word		
	0x6061	8	Mode Of Operation Display		
0x1A0A	0x6041	16	Status word		
0x1A0B	0x6061	8	Mode of operation display		
0x1A0C	0x6062	32	Position Demand [UU]		
0x1A0D	0x6063	32	Actual position [counts]		
0x1A0E	0x6064	32	Position actual value		
0x1A0F	0x6069	32	Velocity sensor actual value [counts/sec]		
0x1A10	0x606B	32	Velocity demand [cnt/sec]		
0x1A11	0x606C	32	Velocity actual value		
0x1A12	0x6074	16	Torque demand value		
0x1A13	0x6077	16	Torque actual value		
0x1A14	0x60B9	16	Touch Probe status		
0x1A15	0x60BA	32	Touch Probe Pos1 Positive		
0x1A16	0x60BB	32	Touch Probe Pos1 Negative		
0x1A17	0x60BC	32	Touch Probe Pos 2 Positive		
0x1A18	0x6079	32	DC link circuit voltage		
0x1A19	0x60F4	32	Position Following error		
0x1A1A	0x60FA	32	Control Effort [cnt/sec]		
0x1A1B	0x60FC	32	Position Demand Value [cnt]		
0x1A1C	0x60FD	32	Digital Inputs		
0x1A1D	0x2205	16	Analog input		
0x1A1E	0x20A0	32	Auxiliary position actual value		
0x1A1F	0x6078	16	Current actual value		

Table 7: TxPDO Default Values

2.2.2. Emergency Requests

Emergency messages are triggered by the occurrence of a device internal error situation. The transmission executes via mailbox interface. An emergency object is transmitted only once per 'error event'.

CoE Header		Emergency				
Number	RSVD	Service	Error Code	Error Register	Data	RSVD

Frame Part	Data Field	Value / Description
CoE Header	Number	0x00
	RSVD	0x00
	Service	0x01 – Emergency
Emergency	Error Code	Error Code
	Error Register	Error Register
	Data	0000-9FFF: Manufacture Specific A000-EFFF: Diagnostic Data F000-FFFF: Manufacture Specific
	RSVD	

Table 8: Emergency Data Fields

2.3. Synchronization Modes

Elmo devices support two different modes; Master and Slaves for synchronization:

- | | |
|---|--|
| Free run | The Master cycle time and Slave cycle time are independent. |
| Synchronization with distributed clocks | The Master cycle time and Slave cycle time are synchronized. |

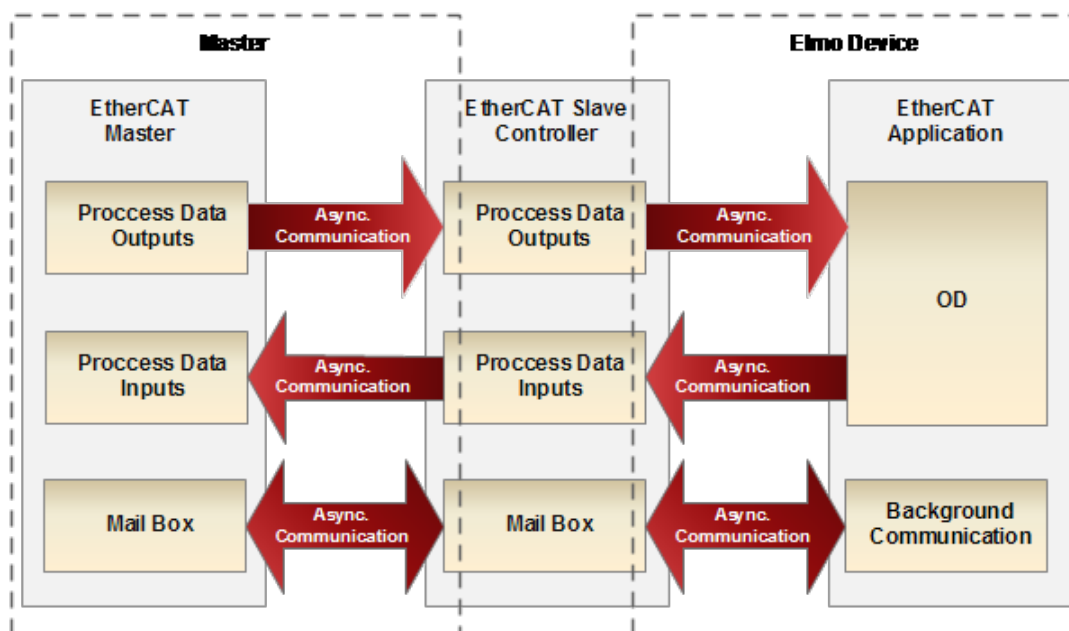


Figure 9: Frame Processing

2.3.1. Free Run

Configuration of the Master cycle time and the Slave cycle time are fully independent, and the slave's application is not synchronized to EtherCAT. Each slave device reads/writes its own process data according to its local time, independent of the Master's cycle time.

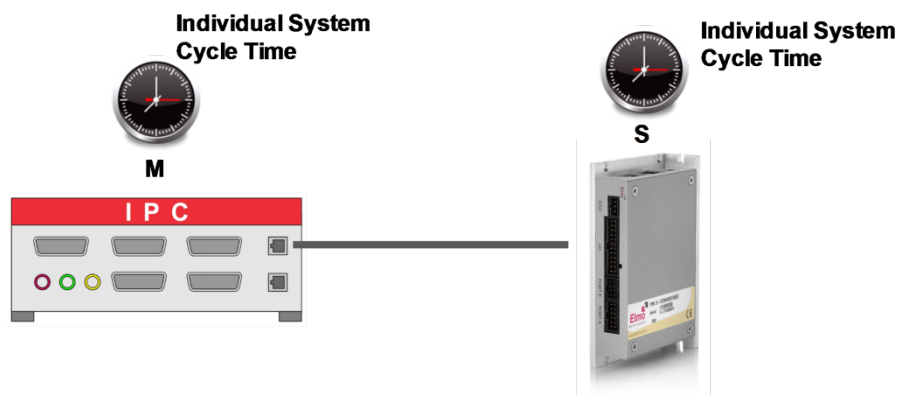


Figure 10: Master - Slave Cycle Times

The following diagram details the process timing in Free run mode:

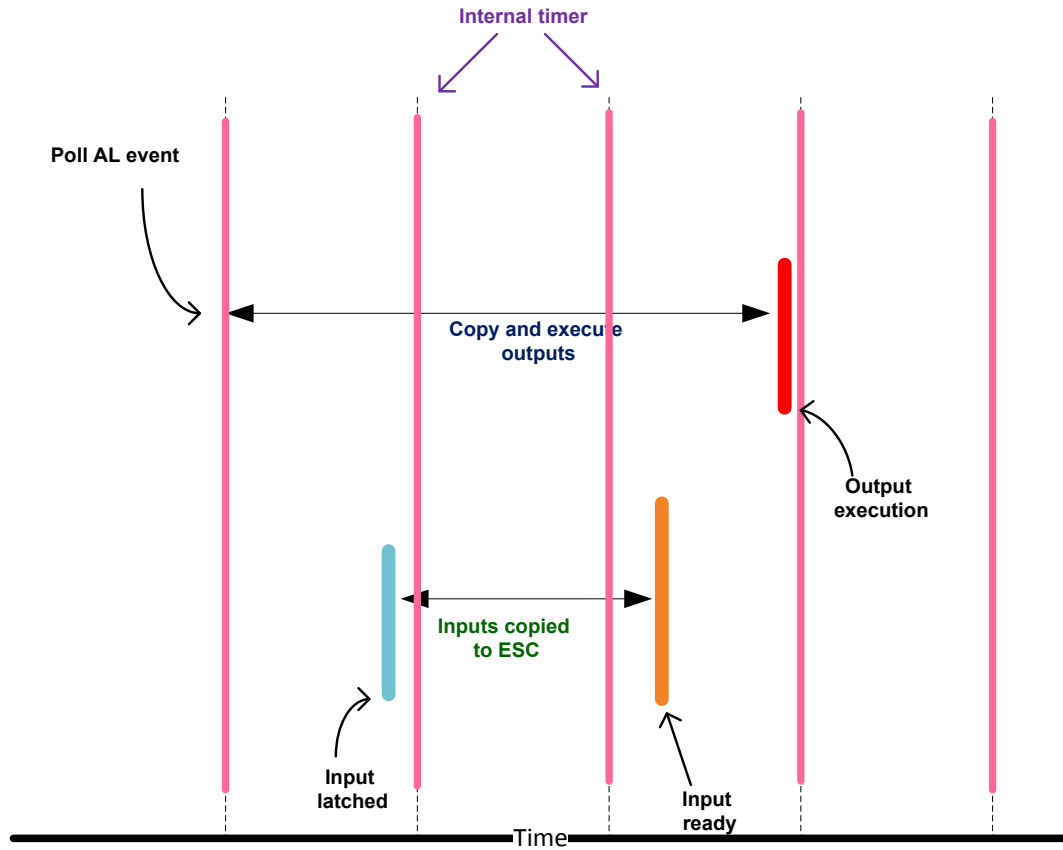


Figure 11: Frame Processing Free Run Mode

The above diagram is explained by the following definitions:

The **Poll AL event** is when the drive pools the AL event in its internal clock every 250µSec. In accordance with the AL indication, the outputs are **copied** to the relevant objects and **executed**. The **Input latched** are inputs which are latched every internal cycle and copied to the ESC. The **Outputs duration** from AL to execution is 750µSec, whereas the **Inputs duration** from latch time to ESC is 500µSec.

2.3.2. Distributed Clocks (DC Mode) - Synchronous with SYNC0

The Slave's application is synchronized to the SYNC0 event, which is based on the distributed clock unit to ensure that the time on all field bus devices is synchronized. Configuration of the Master and Slave cycle times are fully synchronized to the first Slave in the network that is used as a reference clock with the SYNC0 signal. This is then used to synchronize the slave clocks of the other devices and the controller.

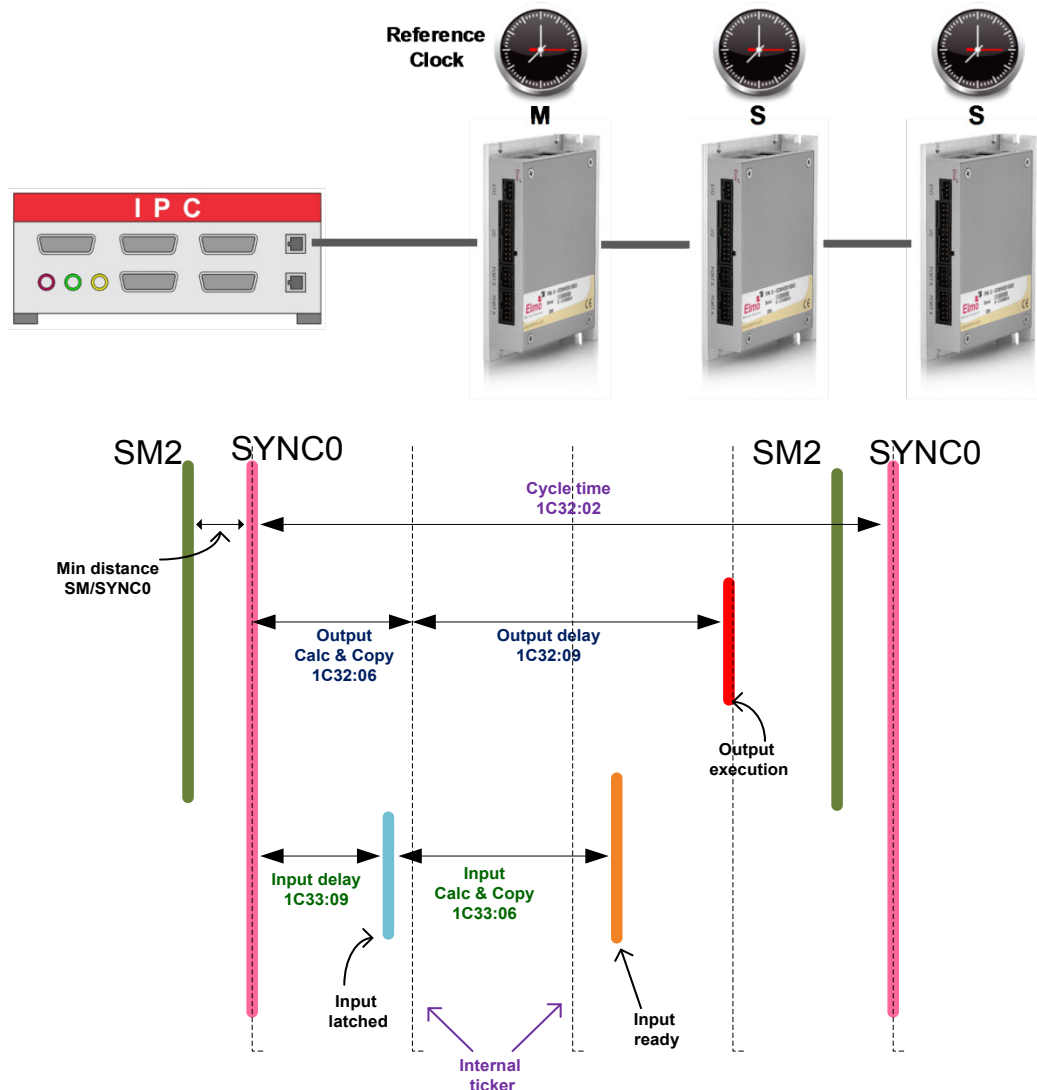


Figure 12: Frame Processing DC Mode Timing

The above Figure 11 details the DC mode timing, where the **Min distance SM/SYNC0** is the minimum time between SM2 arrival and the DC clock. Failing to attain this time may cause a delay in the output execution; typically 50uSec. The **Output Calc&Copy** is the time required to detect that AL arriving and obtain the relevant data from the ESC; up to 250uSec.

Output delay is defined as the time required for the application to perform the output, e.g. in cases of position set-point in cyclic synchronous position mode, this is the time from the set-point time interpolation until the actual torque as a result of this set point; up to 500uSec. The **Input delay** is the time from the SYNC0 signal until the actual inputs capture; 250uSec, and the **Inputs Calc&Copy** is the time taken to copy the inputs to the ESC.

All Slaves can be configured to a maximum of 250 μ s (up to 4 KHz) and the Master cycle time should be a multiple of that value (e.g. 1x250, 2x250, 3x250... nx250).

Synchronization that is carried out due to ESC (EtherCAT Slave Controller) registers configuration. For more information, see Beckhoff's ET1100 Hardware Data Sheet

(http://www.beckhoff.com/english.asp?download/ethercat_development_products.htm) Section II, 3.8 ESC Features (Version 1.8 obtainable from http://download.beckhoff.com/download/Document/EtherCAT/Development_products/EtherCAT_ET1100_Datasheet_all_v1i8.pdf).

2.4. EoE – Ethernet Over EtherCAT

The Ethernet Over EtherCAT (EoE) protocol enables the standard IP-based protocols TCP/IP and UDP/IP which are based on the higher level protocols HTTP and FTP, etc.

EoE can be used as layer 2 protocol (like an Ethernet switch) for transparently tunneling Ethernet frames over the EtherCAT communication.

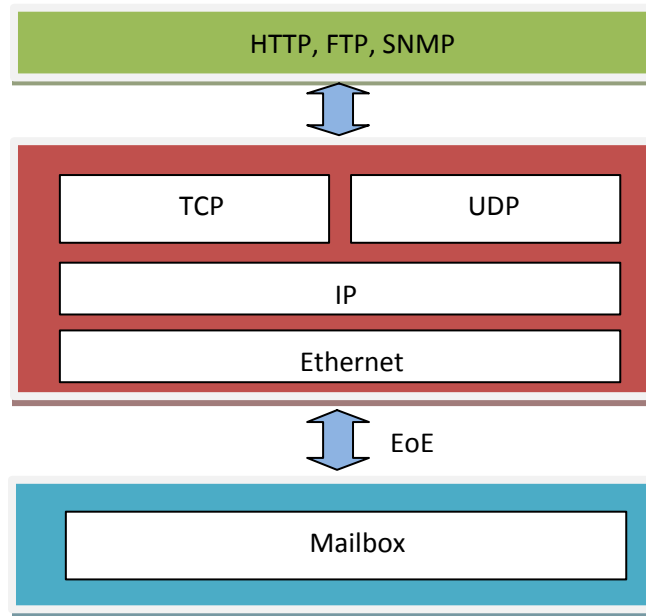


Figure 13: EoE Model

EoE has these services:

Initiate EoE	Used to transmit the first fragment of an Ethernet frame. It conveys the service specific parameters (such as MAC, IP, DNS, etc.) of the service request.
EoE fragment	Used to transmit the fragments of an Ethernet frame. This follows the Initiate EoE service, in case the Ethernet frame that is being transmitted is larger than the data parameter of the Initiate EoE service.
Set IP parameters	Used to transfer IP parameters from the client to the server. The server responds with the result of the Set operation. (The parameters are listed in section 5.3.2 – IP Address Info).
MAC Filter	Used if the EtherCAT device acts as bridge. Not currently supported.

2.4.1. EoE Extensions

2.4.1.1. MAC Address Info

Specify a virtual MAC address for the device according to ISO/IEC 8802-3. With this object the user can set or get the device's virtual MAC address.

Index	Sub-Index	Description	Data Type
0x1110	0	Virtual Ethernet MAC address	Unsigned 48

2.4.1.2. IP Address Info

With this object the user can set or get the device's virtual IP address.

Index	Sub-Index	Description	Data Type
0x1111	0	Number of entries	Unsigned 8
	1	IP Address	Unsigned 32
	2	Subnet Mask	Unsigned 32
	3	Default Gateway	Unsigned 32
	4	DNS Server	Unsigned 32
	5	DNS Name	String

2.5. FoE – File Access over EtherCAT

FoE specifies the standard method for downloading a firmware file from a Client machine (Master EtherCAT) to the Server (Slave) machine. This method uses the slave machine's mailbox interface to perform the firmware update. The FoE can also be used to update the PAL code in the drive.

The following services are defined within a FoE operation:

WRQ	Writing with an indication of the file name
PRQ	Reading with an indication of the file name
DATA	Data block
ACK	Acknowledgment of DATA and WRQ requests
ERR	Error notification
BUSY	Notification that the slave machine is currently busy

To transfer a new firmware version, the Slave initially must be set to "Boot" mode.

Notes

- Currently, only the write service is supported by the FoE operation. The service is mapped to mailbox transmission.
- The latest firmware files are located on Elmo's website.
- The G-MAS allows downloading of both firmware and PAL to the drive or several drives, via the EAS configurator, using the FoE.

2.6. EEPROM

Each node in the EtherCAT network contains an EEPROM device. The EEPROM stores device configuration and device description information.

All information, such as device identity, application data, PDI control and configuration settings are stored at a specific address. The addresses range from 0x00 to 0x3F.

The startup process includes a step where the EtherCAT Slave Controller must acquire and load configuration data from the EEPROMs.

Word

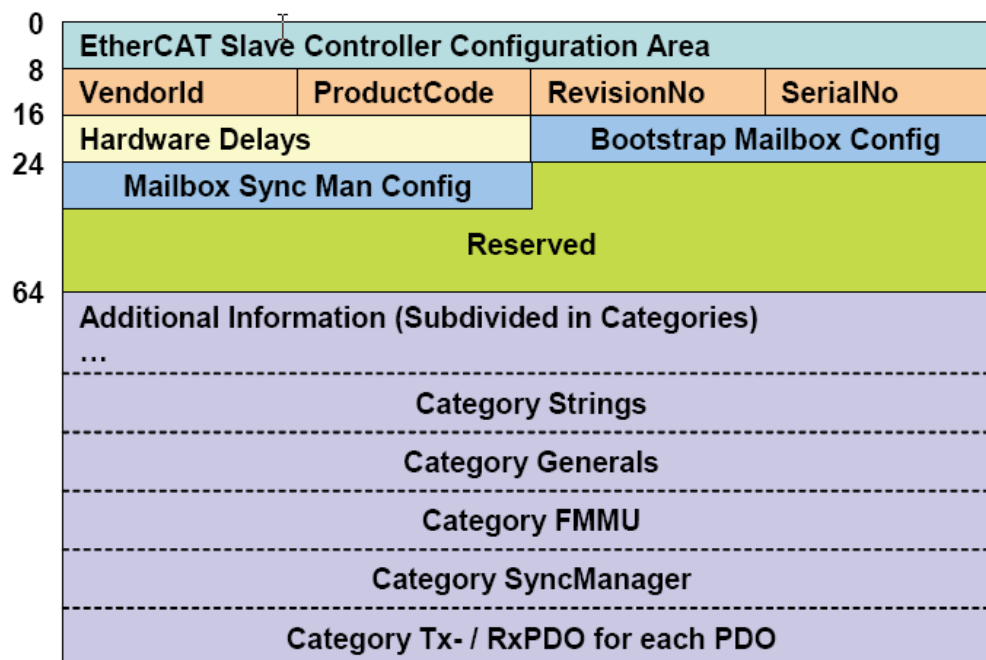


Figure 14: EEPROM Structure

Note: Write Access is not supported.

The information stored in the address range from word 0 to 63 (0x00 to 0x3F) is mandatory.

Word Address	Parameter	Word Address	Parameter
0x0	PDI Control	0x13	Reserved
0x1	PDI Configuration	0x14	Bootstrap Receive Mailbox Offset
0x2	Pulse Length of SYNC Signal	0x15	Bootstrap Receive Mailbox Size
0x3	Extended PDI Configuration	0x16	Bootstrap Send Mailbox Offset
0x4	Configure Station Alias	0x17	Bootstrap Send Mailbox Size
0x5 : 0x6	Reserved	0x18	Standard Receive Mailbox Offset
0x7	Checksum	0x19	Standard Receive Mailbox Size
0x8 : 0x9	Vendor ID	0x1A	Standard Send Mailbox Offset
0xA : 0xB	Product Code	0x1B	Standard Send Mailbox Size
0xC : 0xD	Revision Number	0x1C	Mailbox Protocol
0xE : 0xF	Serial Number	0x1D : 0x3D	Reserved
0x10	Execution Delay	0x3E	Size
0x11	Port0 Delay	0x3F	Version
0x12	Port1 Delay	0x40	

Table 9 : EEPROM Mandatory range memory

For a more detailed description of the EEPROM, refer to the “EtherCAT Slave Device Description” specification, which is available from the EtherCAT Technology Group at <http://www.ethercat.org>.

2.7. ESI (xml format)

ESI contains information about Elmo devices and their functionality. This information is as follows:

- Vendor: ID, Name, Logo
- Device Identity, Name, PDI type
- PDO mapping
- FMMU/SyncManger
- Group
- Types
- Etc.

Based on:

- EtherCATInfo.xsd
- EtherCATBase.xsd
- EtherCATDict.xsd

Notes

- The latest ESI file can be downloaded from Elmo's website at www.elmomc.com.
- Modification of the ESI file is prohibited.

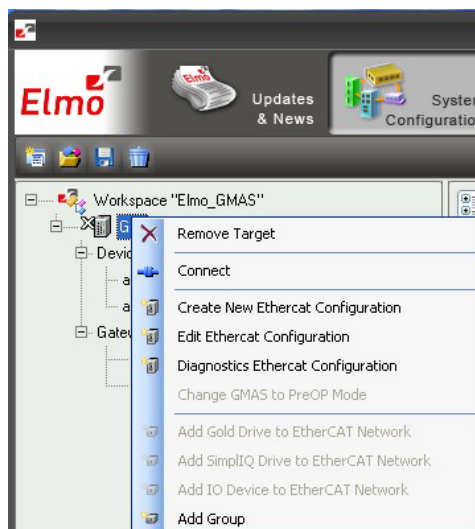
Chapter 3: G-MAS Communication

This chapter describes how to set up the G-MAS and drives using EtherCAT communication. The setup applies to a configuration where the G-MAS is not connected, but can equally apply to a G-MAS which is connected.

3.1. EAS EtherCAT Quick Configuration

This EtherCAT configuration allows the user to quickly setup the system for EtherCAT communication without have to configure individual drives. This setup assumes the G-MAS and /or Drives and connected and are visually seen in the Workspace by the EAS application.

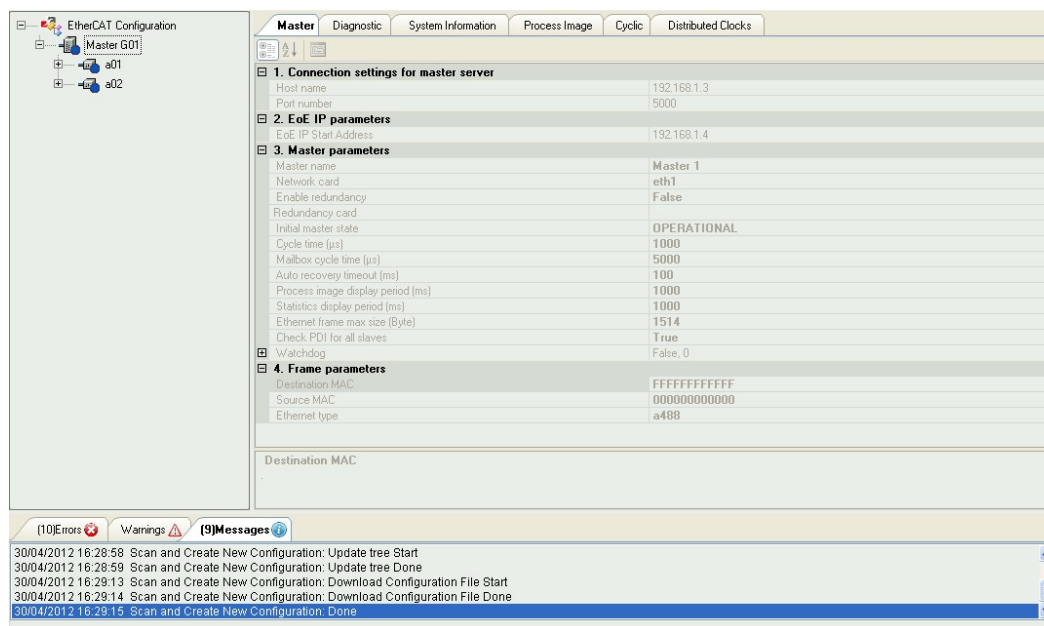
To Setup EtherCAT configuration



1. Right-click on the G-MAS icon in the Workspace window tab. A menu opens displaying options, which includes the following:

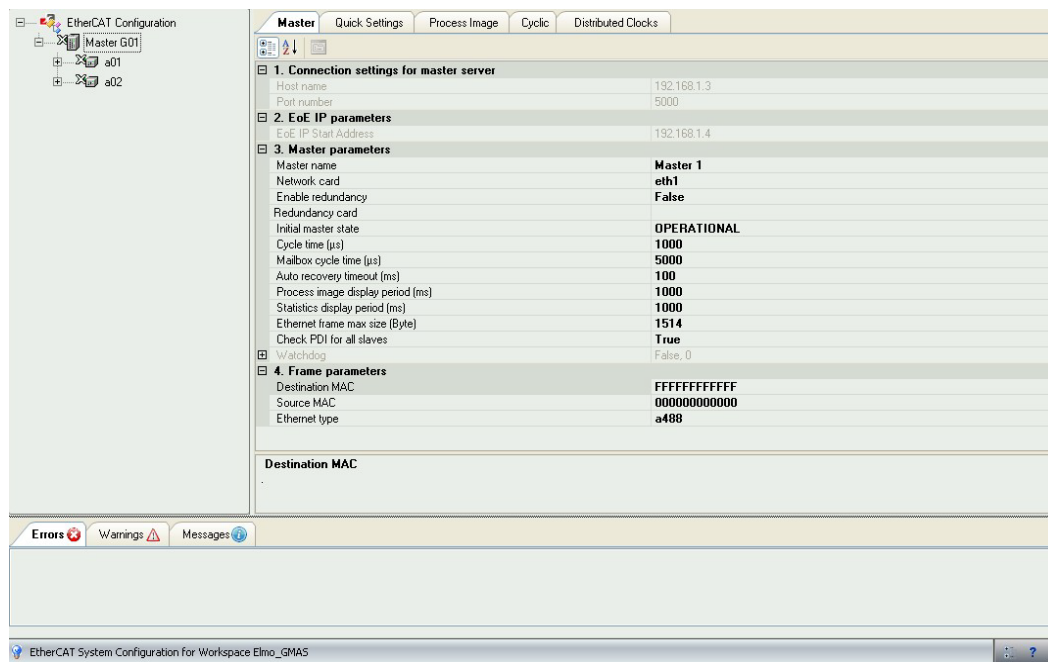
Create New EtherCAT configuration

Creates a new EtherCAT configuration by locating all connections, which are either connected or potentially connected via EtherCAT



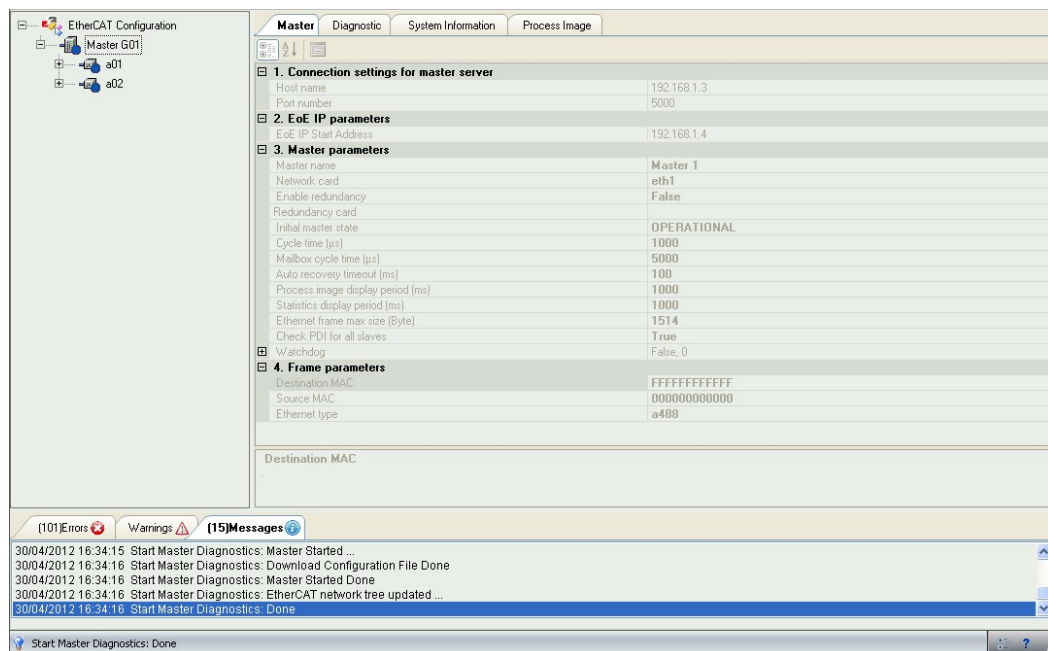
Edit EtherCAT Configuration

Edits the present configuration and allows editing of the configuration

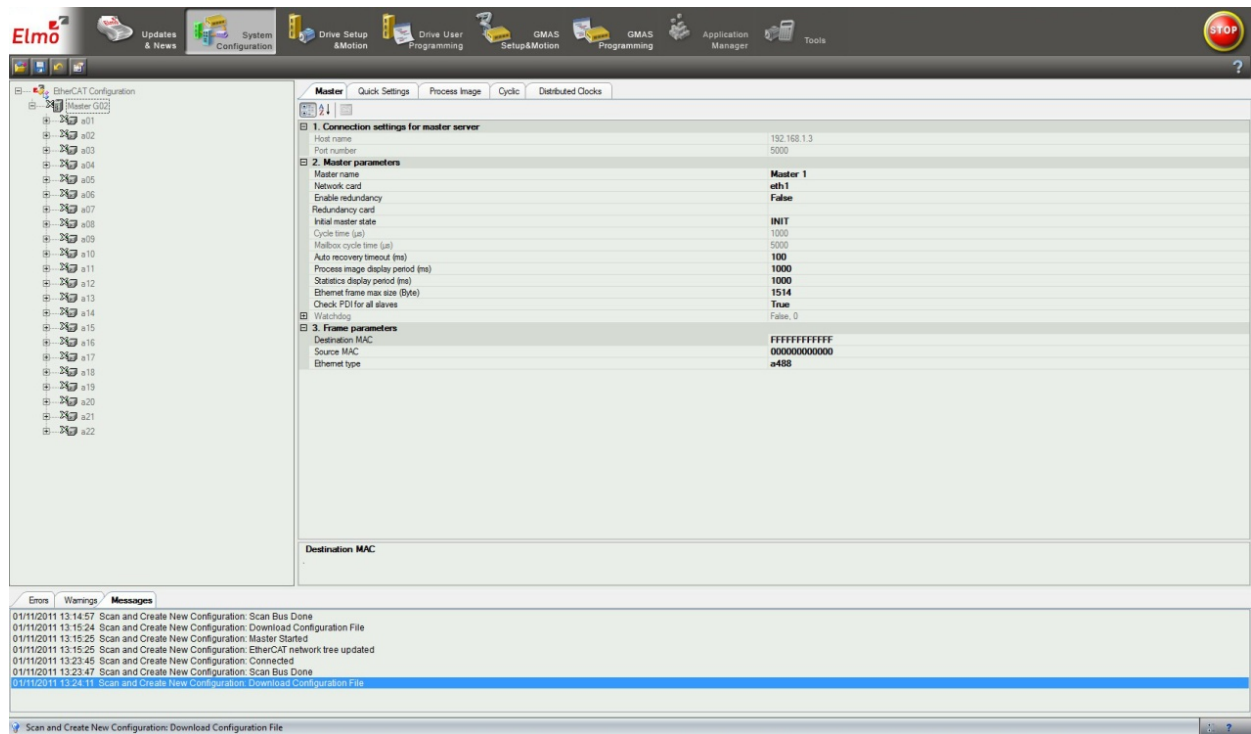


Diagnostics EtherCAT Configuration

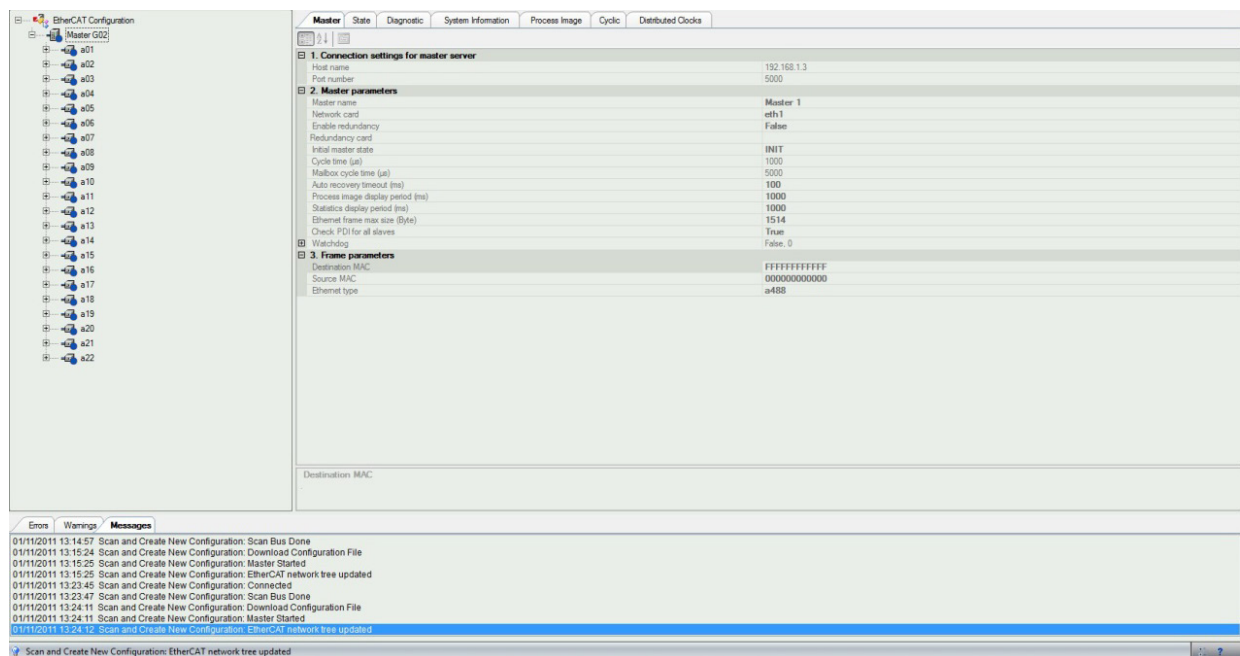
Adds a present or new drive to the EtherCAT Network



- To create a new EtherCAT configuration, select **Create New EtherCAT configuration**. The system automatically detects and enters EtherCAT configuration mode.



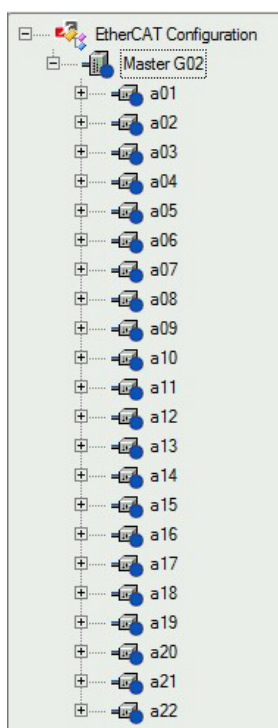
Finally appearing as:



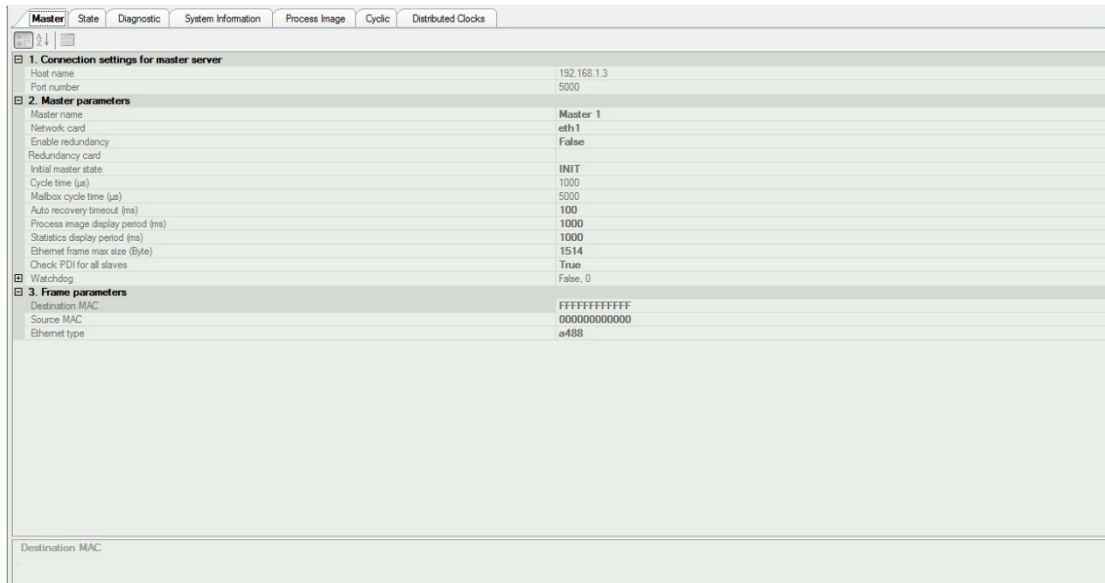
Notice that the Messages tab window displays the operations that the EtherCAT configuration has applied:

```
01/11/2011 13:14:57 Scan and Create New Configuration: Scan Bus Done
01/11/2011 13:15:24 Scan and Create New Configuration: Download Configuration File
01/11/2011 13:15:25 Scan and Create New Configuration: Master Started
01/11/2011 13:15:25 Scan and Create New Configuration: EtherCAT network tree updated
01/11/2011 13:23:45 Scan and Create New Configuration: Connected
01/11/2011 13:23:47 Scan and Create New Configuration: Scan Bus Done
01/11/2011 13:24:11 Scan and Create New Configuration: Download Configuration File
01/11/2011 13:24:11 Scan and Create New Configuration: Master Started
01/11/2011 13:24:12 Scan and Create New Configuration: EtherCAT network tree updated
```

In addition, the Workspace window pane, displays the EtherCAT configuration model of the Master G-MAS and its slave drives.

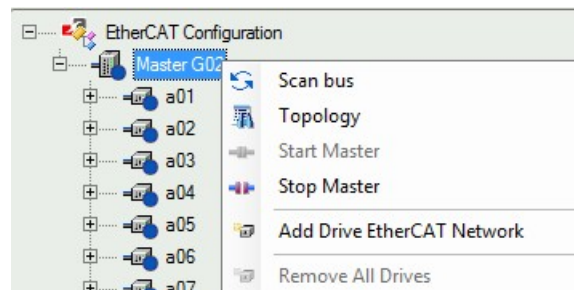


3. Click the Master XXX icon in the workspace, to gain information about the Master.



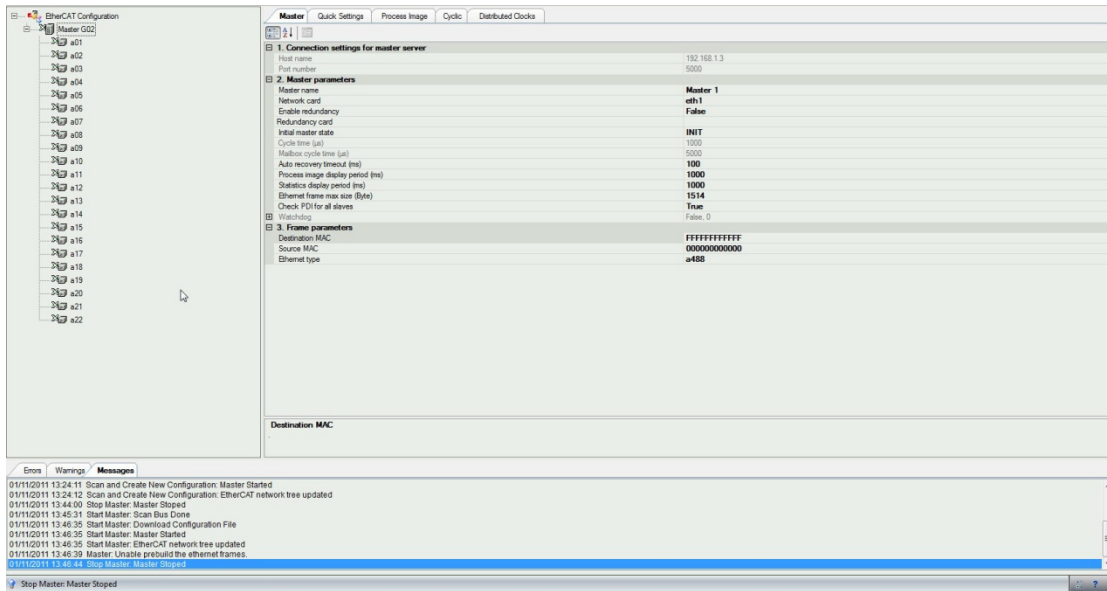
4. Similarly, click a Slave XXX icon in the workspace, to gain information about the Slave.
5. To view or change parameters in any of the main Master or Slave tab setting windows, **deactivate** the Master:

- a. At the EtherCAT Configuration tab, highlight the Master XXXX



- b. Right-click on the Master to open the Master menu.

- c. Select **Stop Master**. The Master is stopped and the parameters tabs become active to edit.



6. To view or change parameters in any of the other Master or Slave tab setting windows, **activate** the Master:
 - a. At the EtherCAT Configuration tab, highlight the Master XXXX
 - b. Right-click on the Master to open the Master menu.

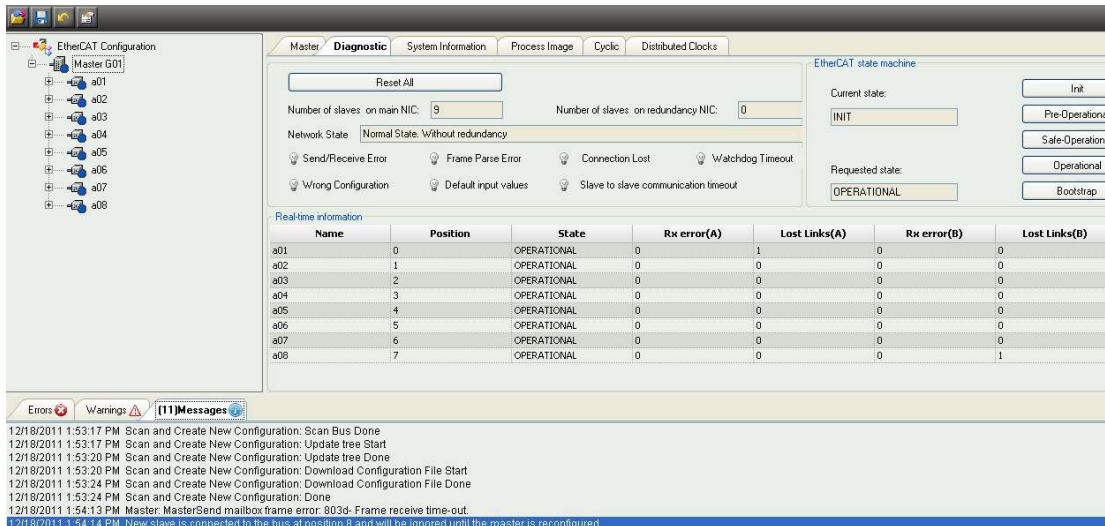


- c. Select **Start Master**. The Master is activated and the parameters tabs become inactive to edit. However, further tabs become available for viewing.

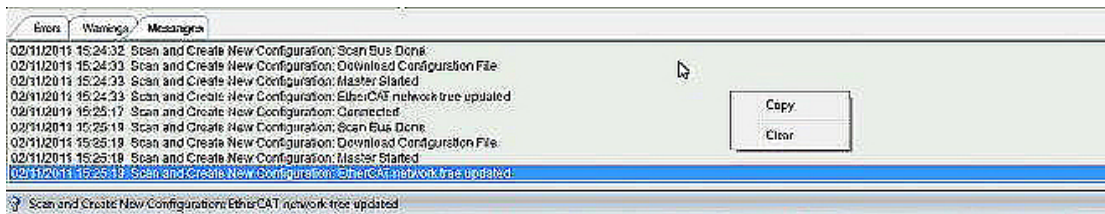
Master		State	Diagnostic	System Information	Process Image	Cyclic	Distributed Clocks
1. Connection settings for master server Host name 192.168.1.3 Port number 5000							
2. Master parameters Master name Master 1 Network card eth1 Enable redundancy False Redundancy card Initial master state INIT Cycle time (µs) 1000 Mailbox cycle time (µs) 5000 Auto recovery timeout (ms) 100 Process image display period (ms) 1000 Statistics display period (ms) 1000 Ethernet frame max size (Byte) 1514 Check PDI for all slaves True Watchdog False, 0 Enable False Check time (ms) 0							
3. Frame parameters Destination MAC FFFFFFFFFFFF Source MAC 000000000000 Ethernet type a488							
Watchdog							

7. To add a Slave drive to the EtherCAT Master G-MAS:
- Deactivate the Master is in step 5 above.
 - Switch off any power supply to the Master and Slave drives.
 - Physically connect the Slave drive to the Master G-MAS.
 - Switch on the power supply to the Master and Slave drives.
 - At the EtherCAT Configuration tab, highlight the Master XXXX, and right-click on the Master to open the Master menu.
 - From the Master menu, select **Scan and Create New Configuration**.

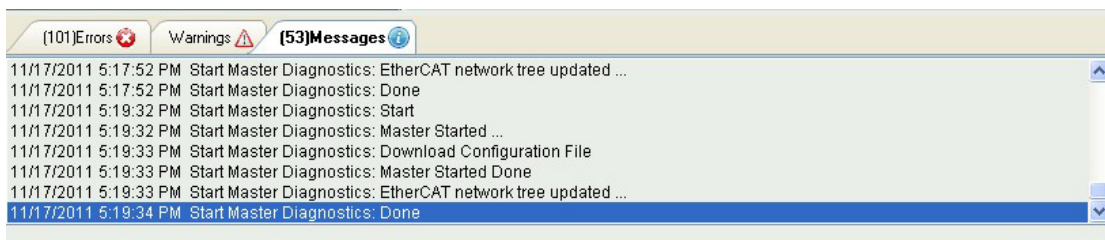
- g. Again, open and select from the Master menu, **Start Master**. The Master and Slaves are activated, and the new message appears that a new Slave has been added.



8. At any time, the Messages can be individually copied, or cleared by right-clicking on the message or selecting Copy or Clear from the popup menu.



In addition, the lower tabs record the number of Errors, Warnings, and Messages in brackets. This is useful for diagnostics, and as a warning if the number of errors becomes very large.



3.2. Downloading Firmware to G-MAS using FoE

If the G-MAS and servo drives attached are not connected via USB to the EAS application, and the user wishes to employ the FoE connection to the EAS for downloading firmware, refer to Elmo for help in setting and configuring the G-MAS to download firmware via FoE.

3.3. G-MAS EoE Configuration

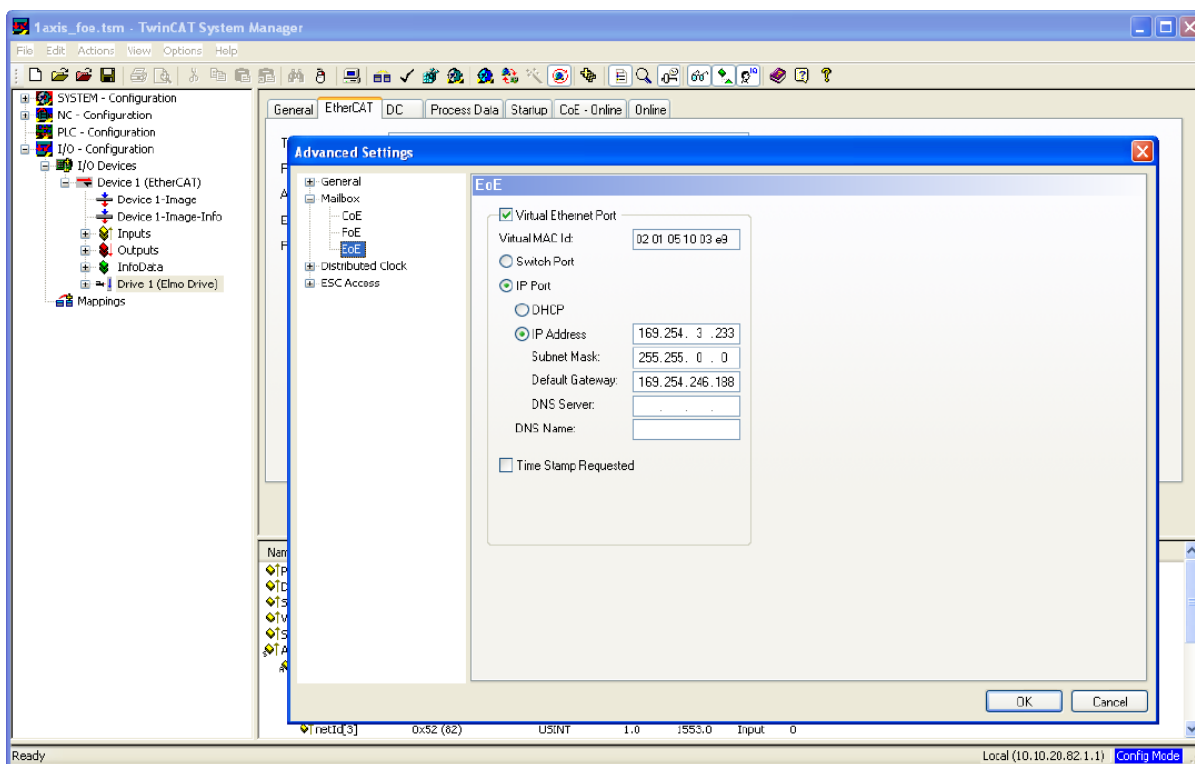
The following instructions demonstrate how to configure EAS application to Elmo drives using EoE protocol. Before starting make sure, the drive connected to a network card (section 5.1.1) or to a switch port (section 5.1.2).

Note: This example assuming there is no switch port connected.

9. Setup the Drive IP address:

a. Using the TwinCAT application, open drive advanced setting:

Device1 -> Drive1->EtherCAT ->Advanced Setting->Mailbox->EoE

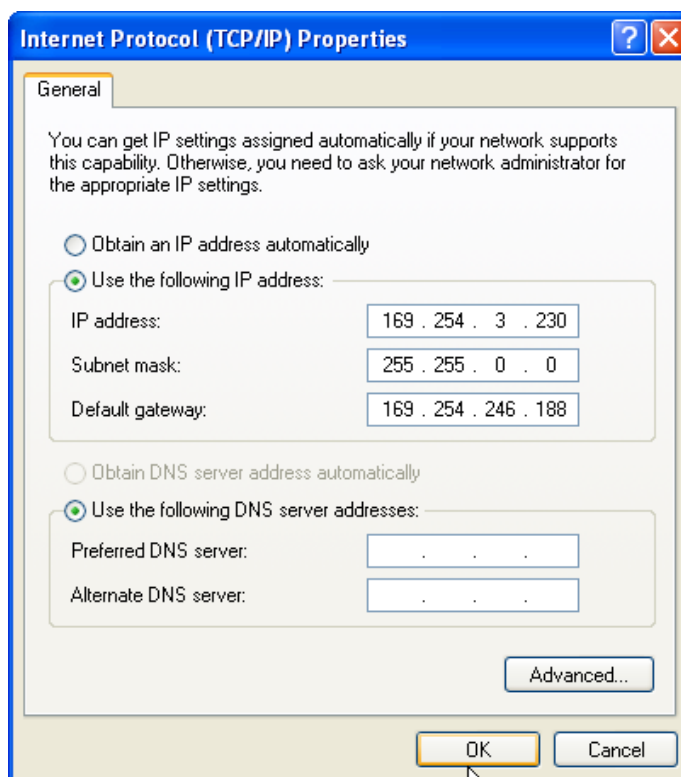


b. Select IP Port and IP Address (radio buttons), TSM automatically update the IP address, Subnet Mask and Default Gateway. Click **OK**.

10. Configure the PC Network Card:

a. To enable the network card using EoE protocol, the network card must configured as follows:

- i. From the Windows Control Panel, select **Network Connections**.
- ii. Right click the relevant network and select **Properties**.
- iii. In the list of items, select **Internet Protocol (TCP/IP)**. You may need to scroll down.
- iv. Click **Properties**.
- v. Select **Use the following IP address**, and enter the new IP address, Subnet mask and Default gateway for example:



- vi. Click **OK**.
 - b. Return to TSM and reload the devices (activate configuration).
11. Before setting the EAS application, open a command prompt window and using the Ping command check for a reply from the drive.

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\eran>ping 169.254.3.233

Pinging 169.254.3.233 with 32 bytes of data:

Reply from 169.254.3.233: bytes=32 time=192ms TTL=128
Reply from 169.254.3.233: bytes=32 time=37ms TTL=128
Reply from 169.254.3.233: bytes=32 time=62ms TTL=128
Reply from 169.254.3.233: bytes=32 time=91ms TTL=128

Ping statistics for 169.254.3.233:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 37ms, Maximum = 192ms, Average = 95ms

C:\Documents and Settings\eran>_
```

12. Now open the EAS application and configure the Drive communication as follow:
 - a. Connection Type : EtherCAT EoE
 - b. IP Address: Same IP address as configure in TwinCAT application (see Section 8.1).
 - c. IP Port: 5001 (Default)

1. General	
1.1 Target Name	Axis_USB
1.2 Target Version	Guitar 01.01.04.31 05Sep2010
1.3 Project	C:\Program Files\Elmo Motion Control\Elmo Application Studio\workspaces\1Axis\Fw_Tests\Fw_Tests.ElmoPRJ
1.4 Active	True
1.5 Target Type	NG Drive
2. Connection	
2.1 Connection Type	EtherCAT EoE
2.2 IP Address	169.254.3.233
2.4 IP Port	5001

13. Press **Connect**.

Chapter 4: TwinCAT Communication

4.1. Architecture

Due to flexibility of EoE there is more than one way to configure an EtherCAT network that includes EoE devices. Each configuration uses the EAS for downloading parameters, axis tuning, recorder, Ethernet communication, etc.

4.2. Using TwinCAT Master

In this configuration, the Master EtherCAT device and EAS run on IPC using the same Ethernet port. EoE communication goes directly from the Master EtherCAT device to the network device (i.e., a switch port is not required).



Figure 15: TwinCAT and EAS EoE Configuration

4.3. Using a Switch Port

In this configuration, the EoE communication goes via a switch port to and from the device using the Ethernet protocol.

EAS (Elmo Application Studio) runs on a remote PC using the switch port terminal. It can run all the EoE services (download parameters, recording tuning, etc.).

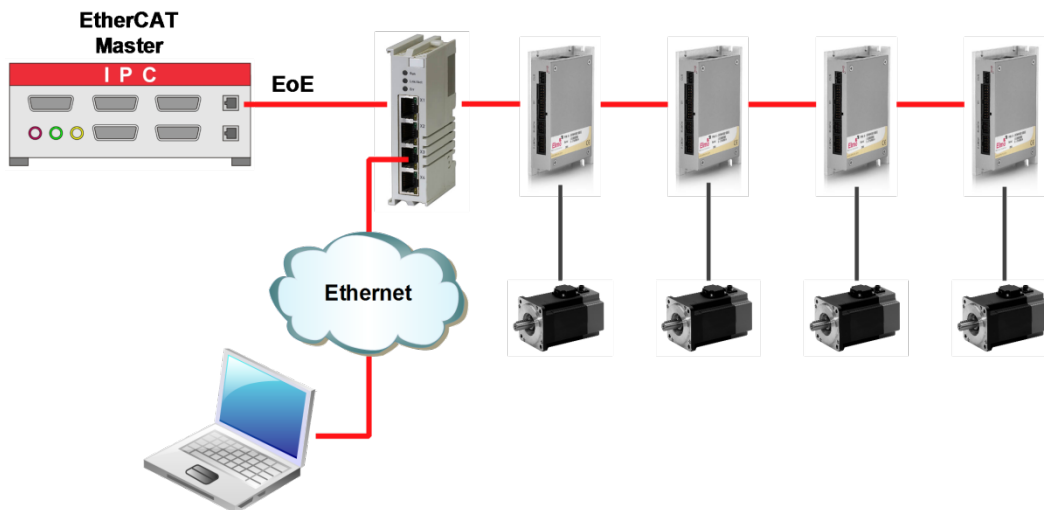


Figure 16: TwinCAT and EAS EoE Configuration Using a Switch Port

4.4. Download Firmware using FoE (via TwinCAT)

4.4.1. Setup Using the TwinCAT NC/PTP System Manager

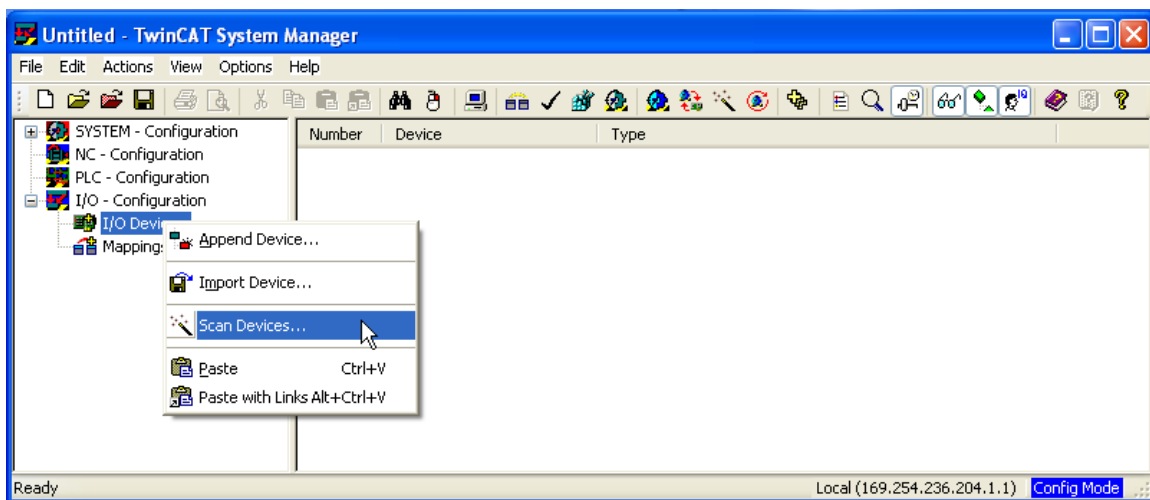
Before starting the drive setup procedures, the following must be complete:

- The drive has been configured by the EAS and the servo motor can move.
- TwinCAT software from Beckhoff (NC/PTP mode setup) is installed.
- The latest ESI (xml format) file is available and has been copied to the TwinCAT slave library (usually found in C:\TwinCAT\IO\EtherCAT). It is then possible to scan the network for new devices.
- The drive is connected to an EtherCAT Master PC/IPC.
- The TwinCAT system manager resides in configuration mode.
- The TwinCAT system manager must be configured with a network card.

4.4.2. Setup Procedure

The following example demonstrates the automatic EtherCAT network setup procedure. Network setup may also be performed manually – refer to the TwinCAT manual for more information.

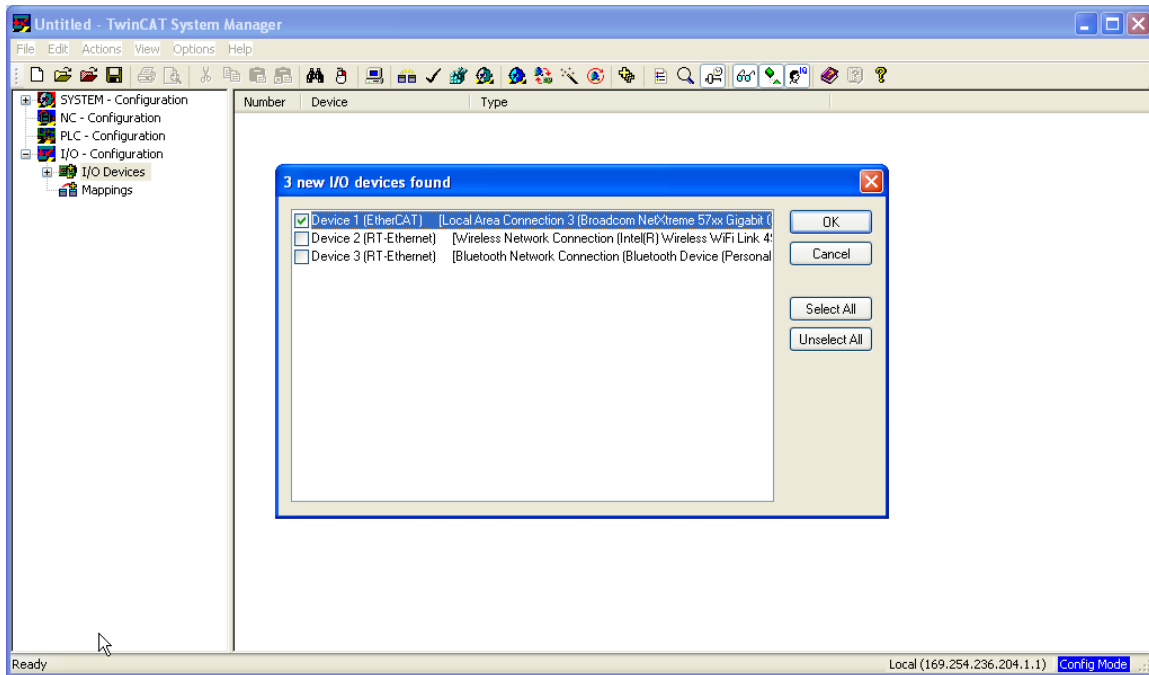
1. Scan Devices:
 - a. Ensure the EtherCAT network is fully connected, and the power is switched ON.
 - b. Create a new project, then right-click and select I/O Devices. Choose **Scan Devices** from the menu. The scan is performed.



A pop-up window informs you that not all devices can be detected by the TwinCAT software. Click **OK** to continue.

2. Select New Devices

An EtherCAT card is found during the scan. Select the EtherCAT checkbox, and then click **OK** to continue.

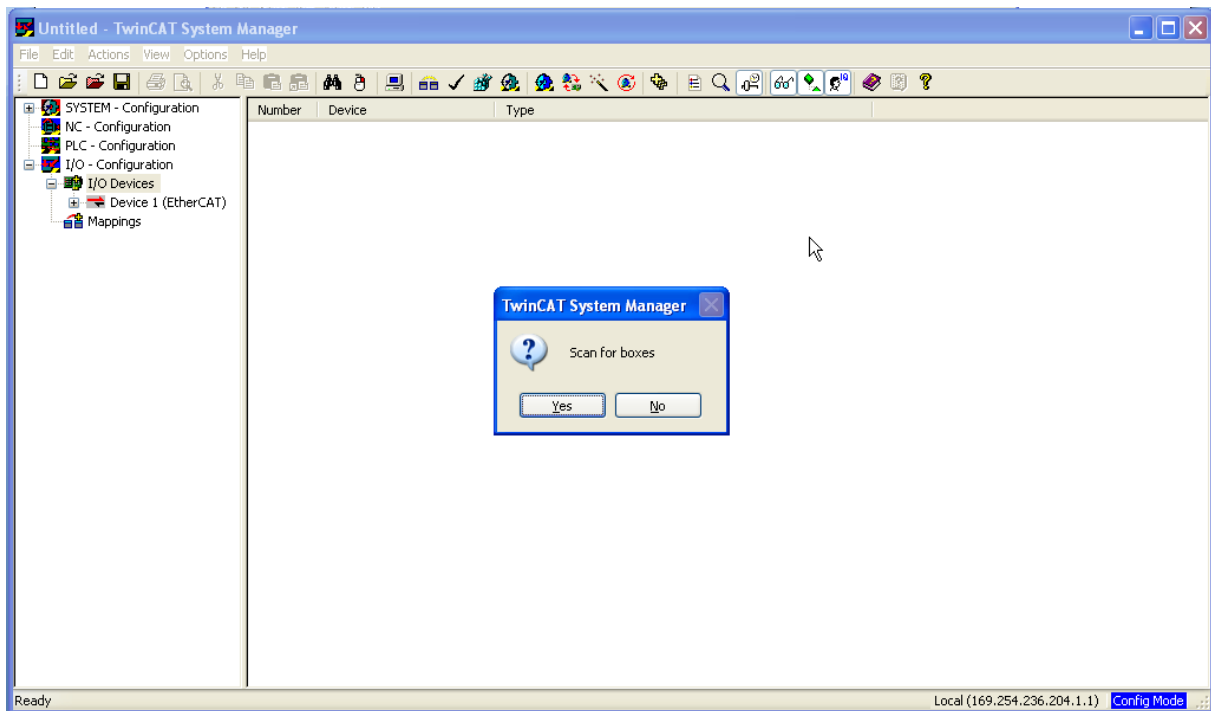


If more than one Ethernet devices is found, select the EtherCAT device only and click **OK** to continue.

If no EtherCAT card is found during the scan, verify that the EtherCAT slave is connected to the correct EtherCAT network that was already installed. Otherwise, TwinCAT will find a real time Ethernet card.

3. Scan for Boxes

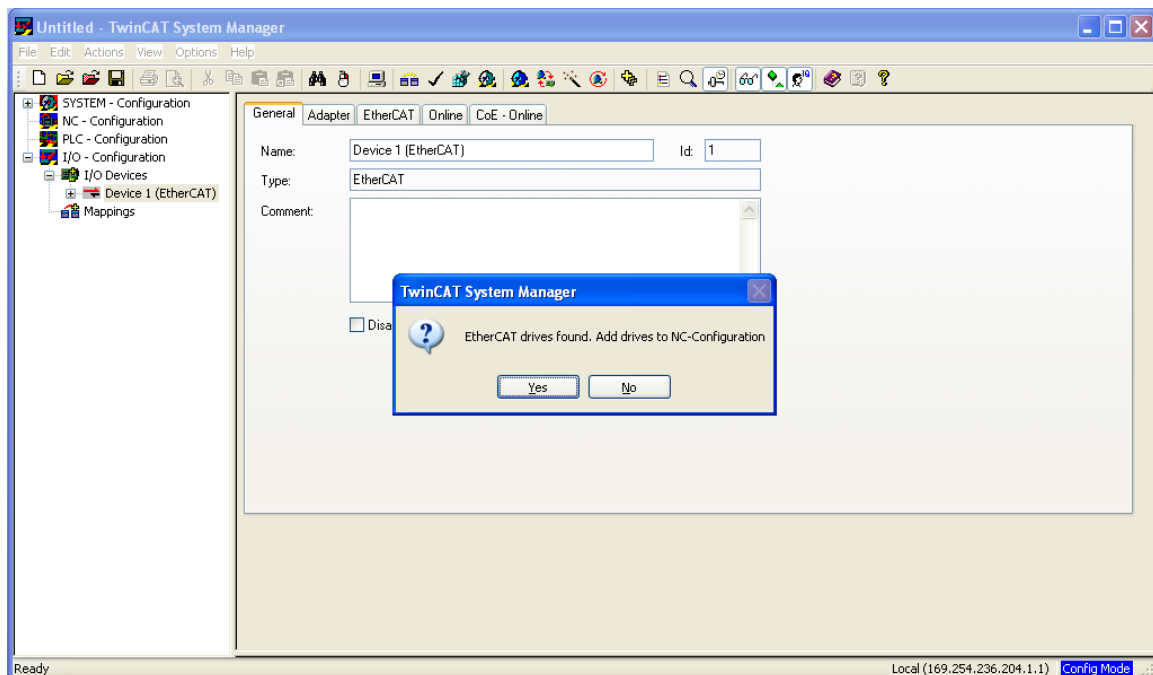
To allow TwinCAT to scan for boxes, click **Yes** in the pop-up window that appears.



4. Adding to the NC Task

After the Box scan is complete, a pop-up window appears. The pop-up window requests that the device in question be added to the NC – Configuration.

Click **Yes**, TwinCAT adds the device to the NC – Configuration.

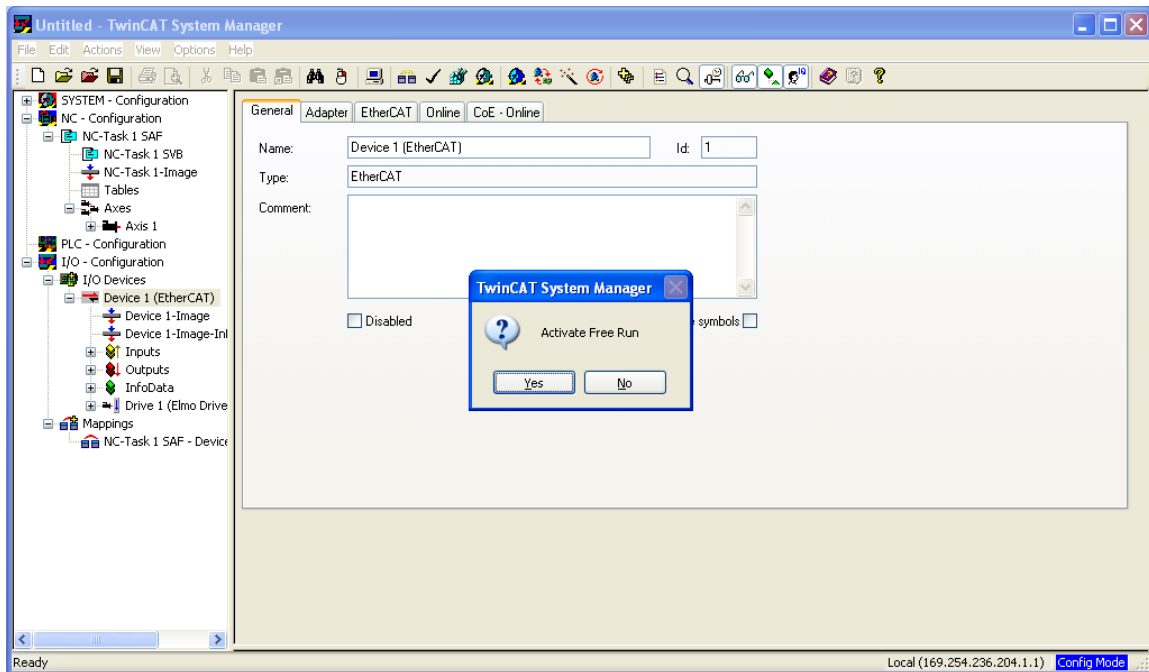


5. Activate the TwinCAT Configuration

After TwinCAT adds the device to the NC – Configuration, a pop-up window appears.

Click **Yes** to stay in the EtherCAT system manager's configuration mode. The TwinCAT system manager automatically changes to an OP state.

Otherwise, click **No** to automatically change the TwinCAT system manager to a PO state and continue to the next step.



6. Enable NC Configuration

7. Enable Axis

To define the download process for new firmware files to Elmo's drive, use the FoE communication link. Each process starts with a user request while the drive is in a BOOT state. Each process finishes after all files are sent to the drive and the Master machine receives an ACK for the last packet.

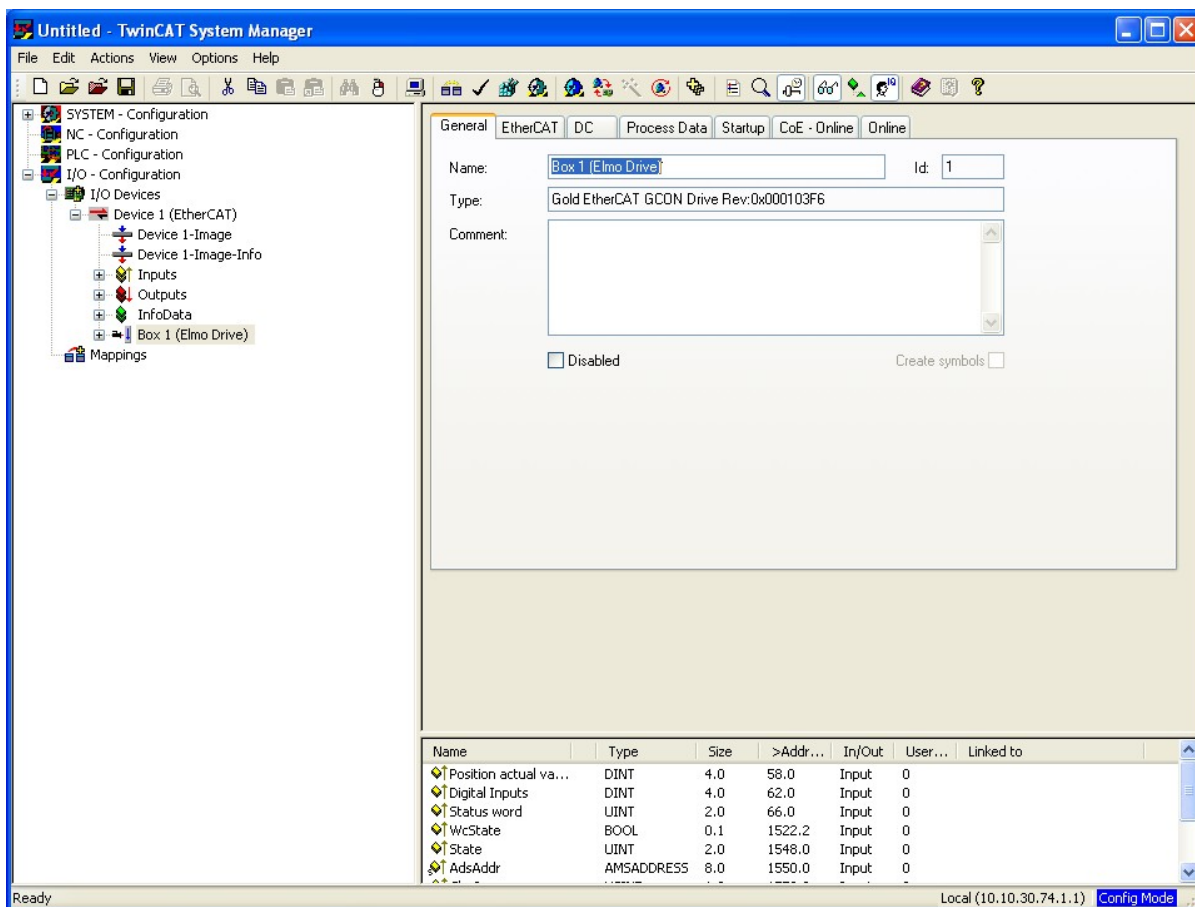
Only FoE communications are handled by the Master machine during a firmware download.

The following steps must be executed in order to update the Elmo devices' firmware.

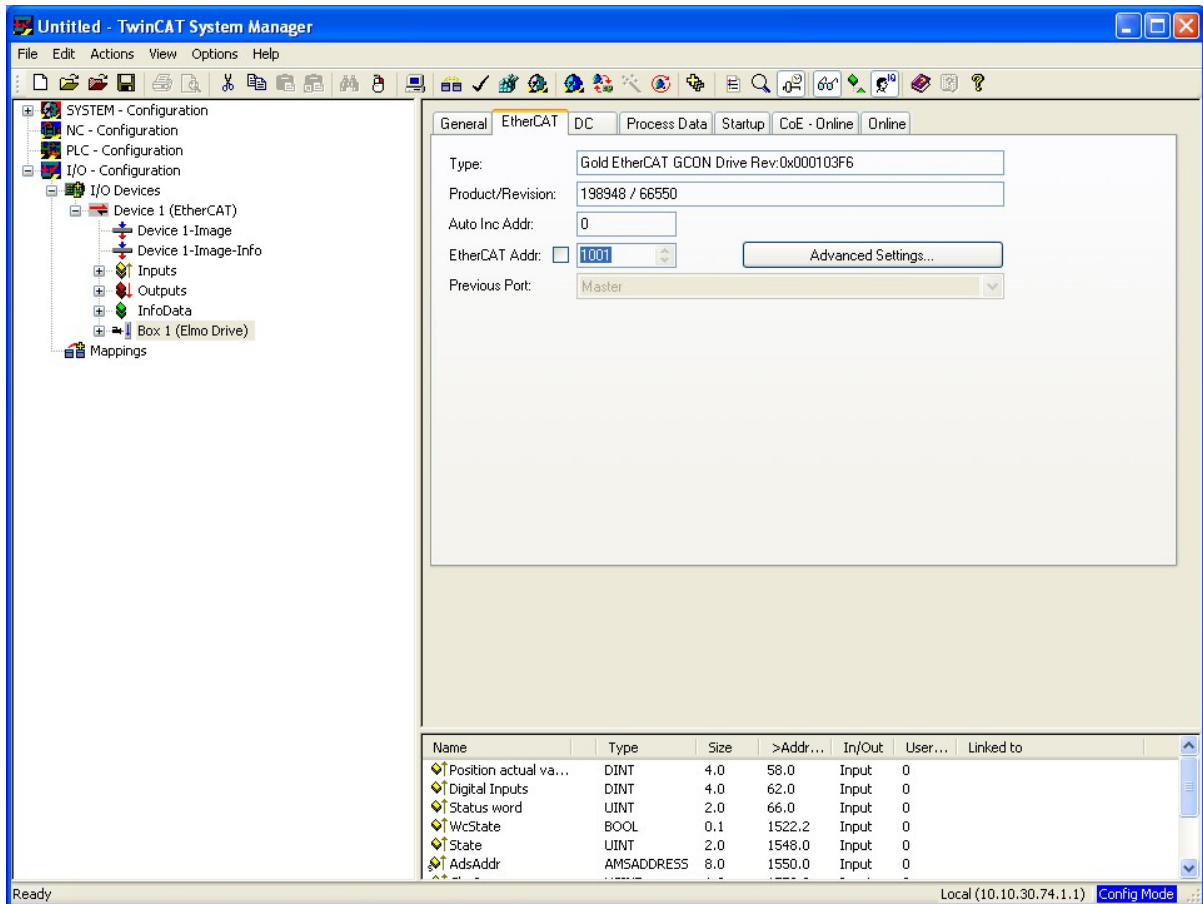
- The USB cable is disconnected.
- All user programs must be closed.
- The mailbox cycle time is configured to 50 msec by default.

To update the firmware

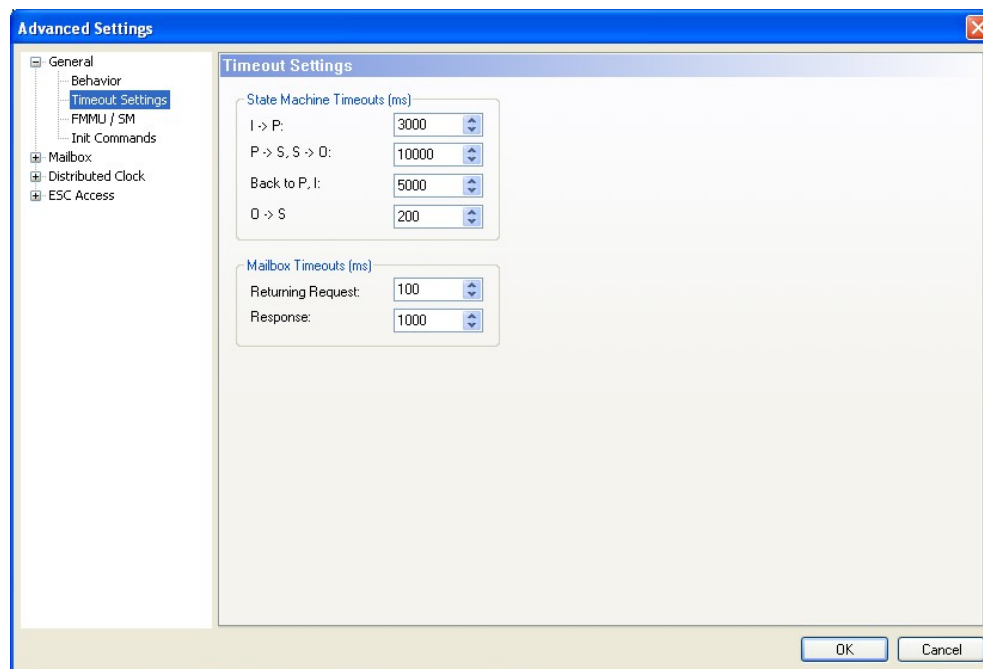
1. Perform the procedural steps described in section 5.2.1 Setup Using the TwinCAT NC/PTP System Manager on page 49 to configure the devices and Boxes attached to the I/O Configuration.



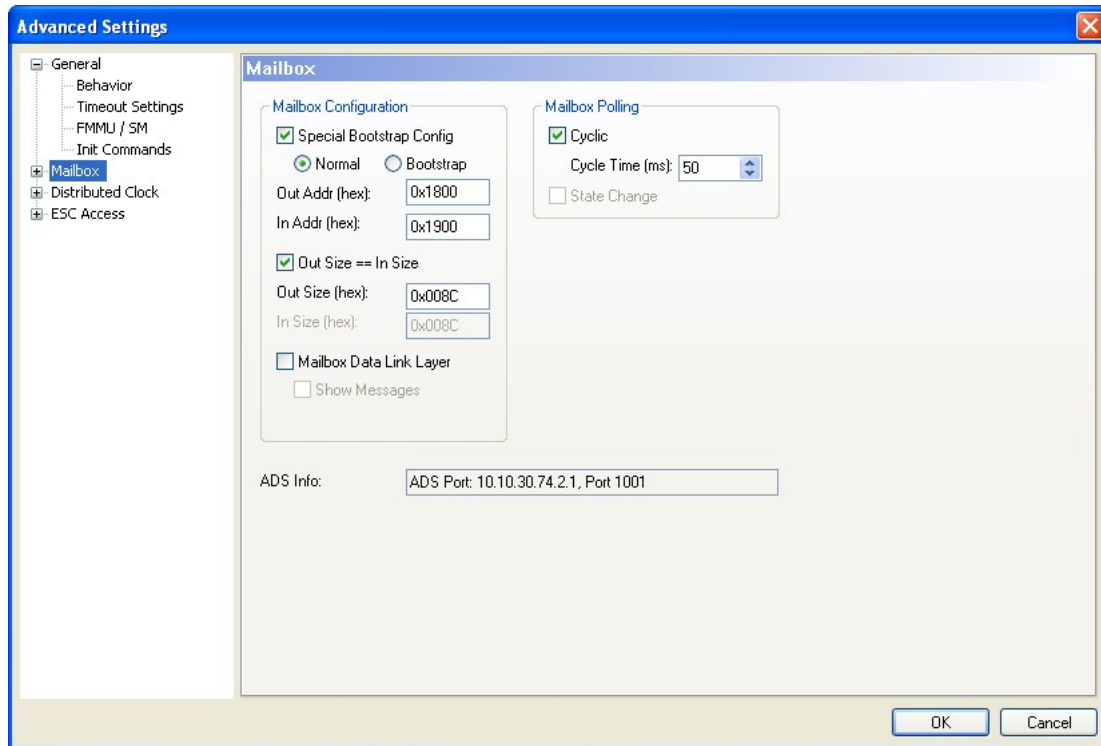
2. Select the EtherCAT tab, and click **Advanced Settings**.



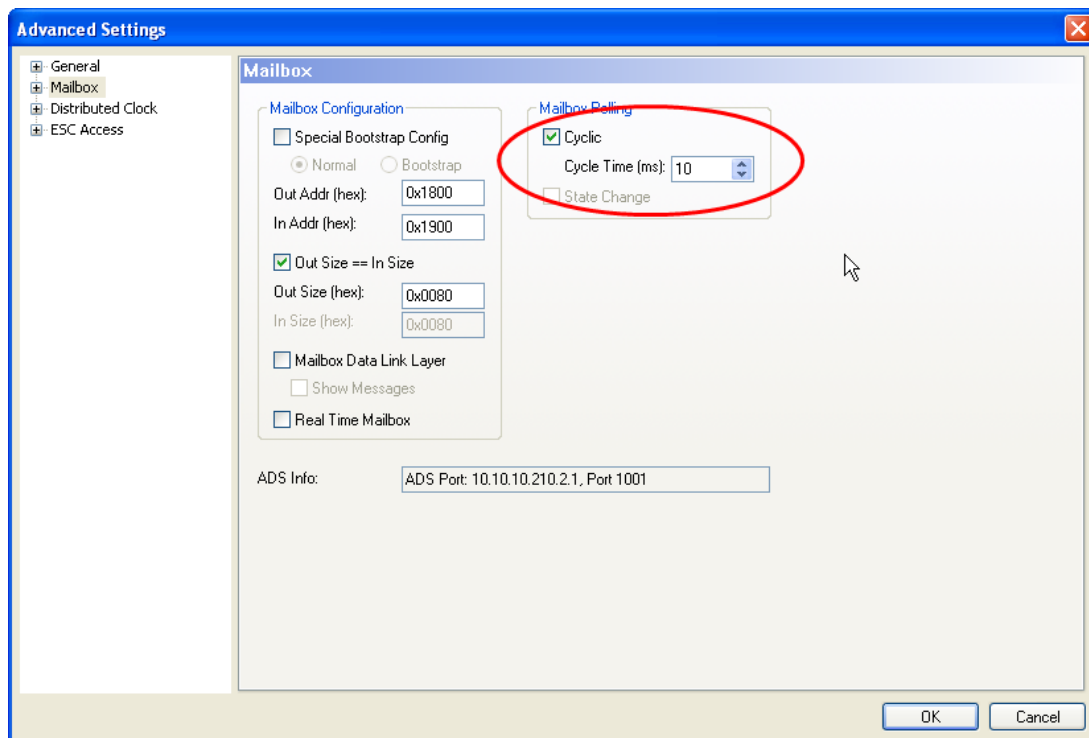
The Advance Settings window opens.



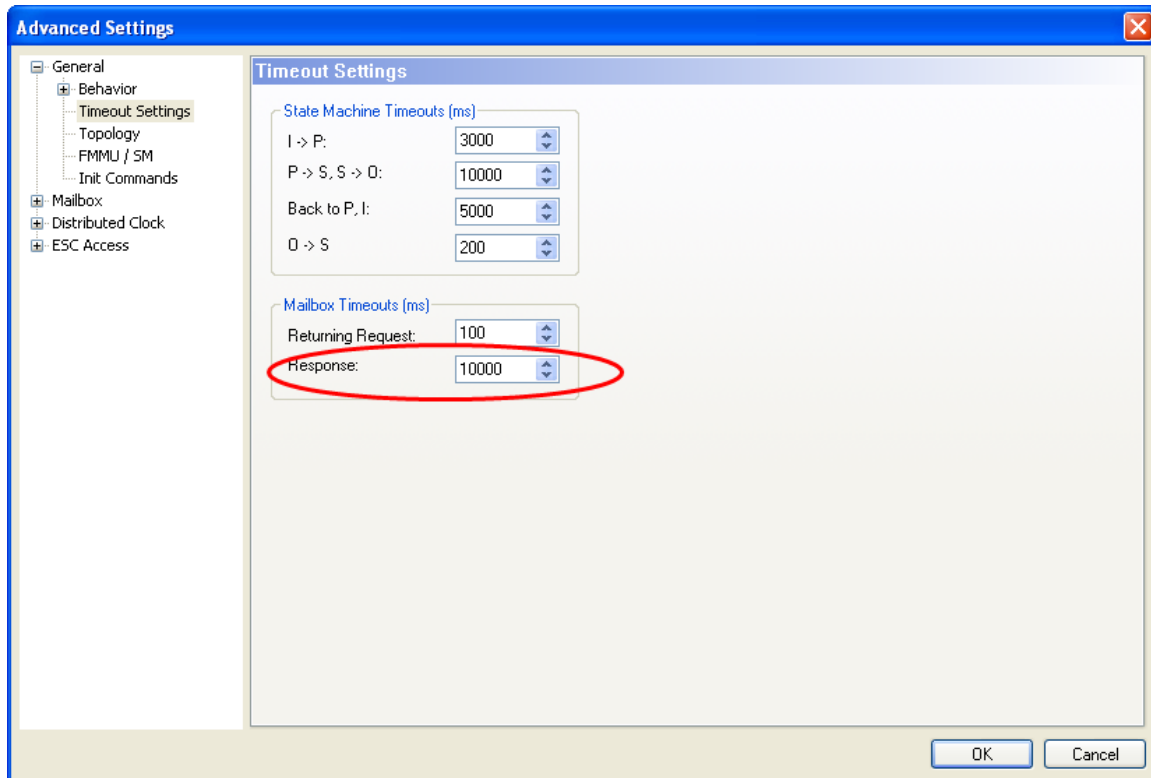
3. At the left hand explorer, click **Mailbox**.



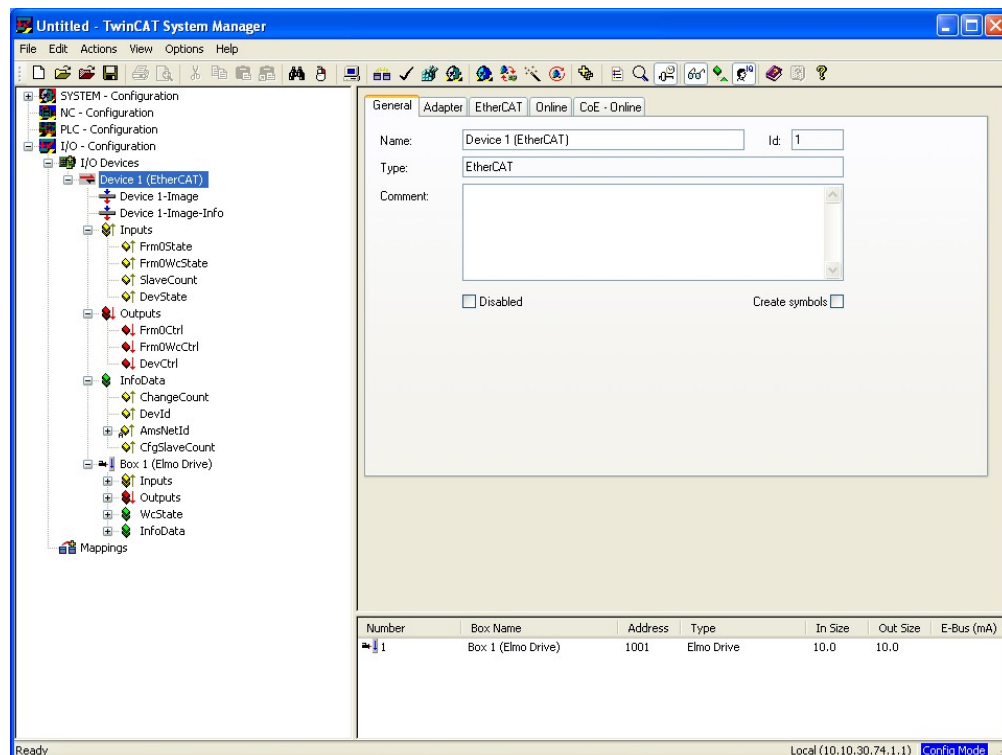
4. At the Mailbox Polling option, click the **Cyclic** checkbox and set the mailbox cycle time to **50 msec** by default.



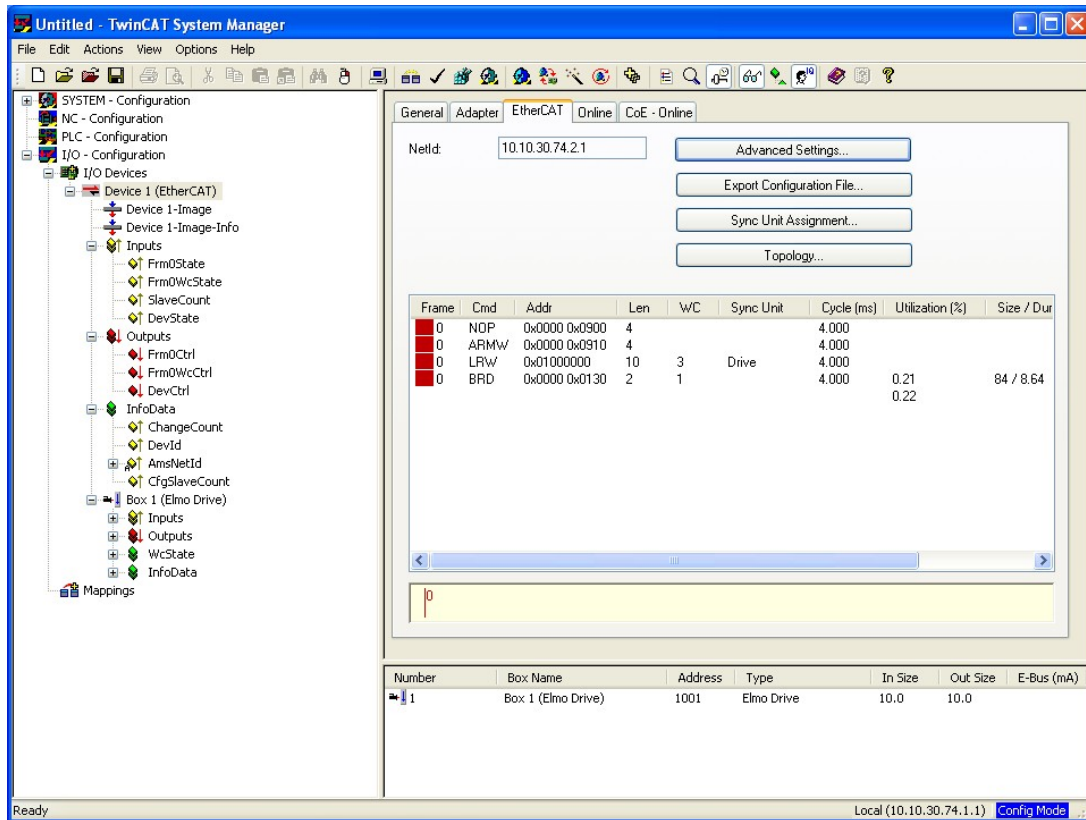
5. In the Timeout Settings of each mailbox, under General, set the Mailbox Timeout Response time to a minimum of **20,000 msec**.



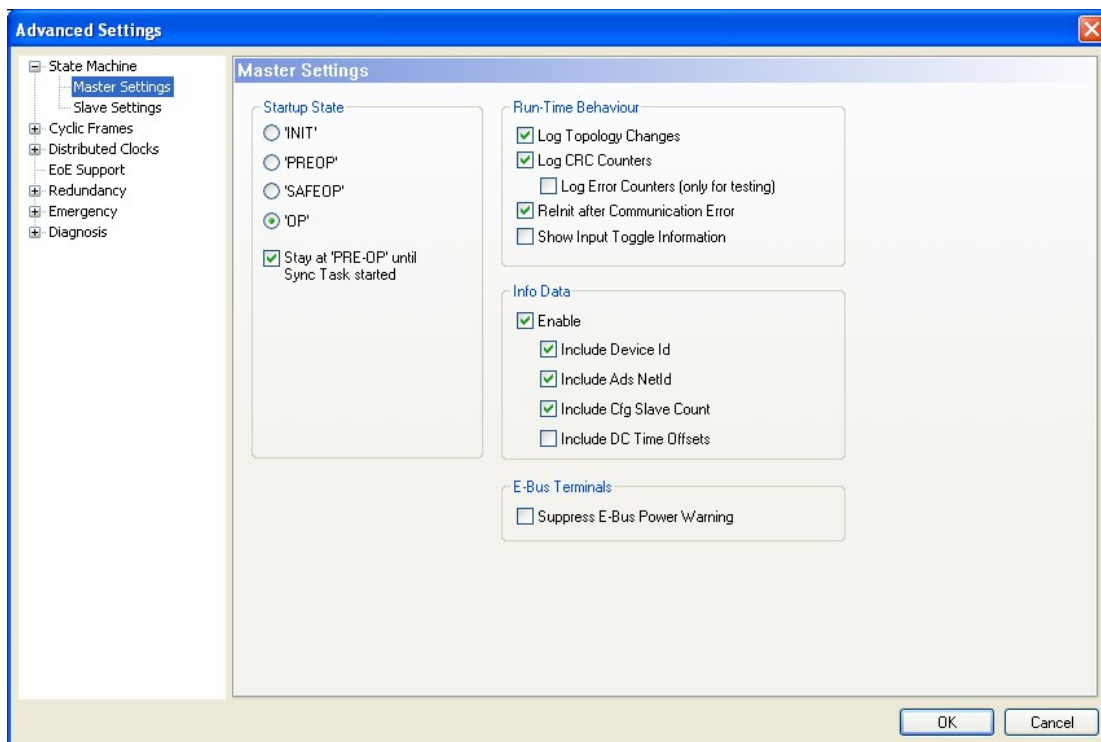
6. Click the Master **Device 1 (EtherCAT)** in the main explorer window at the left side.



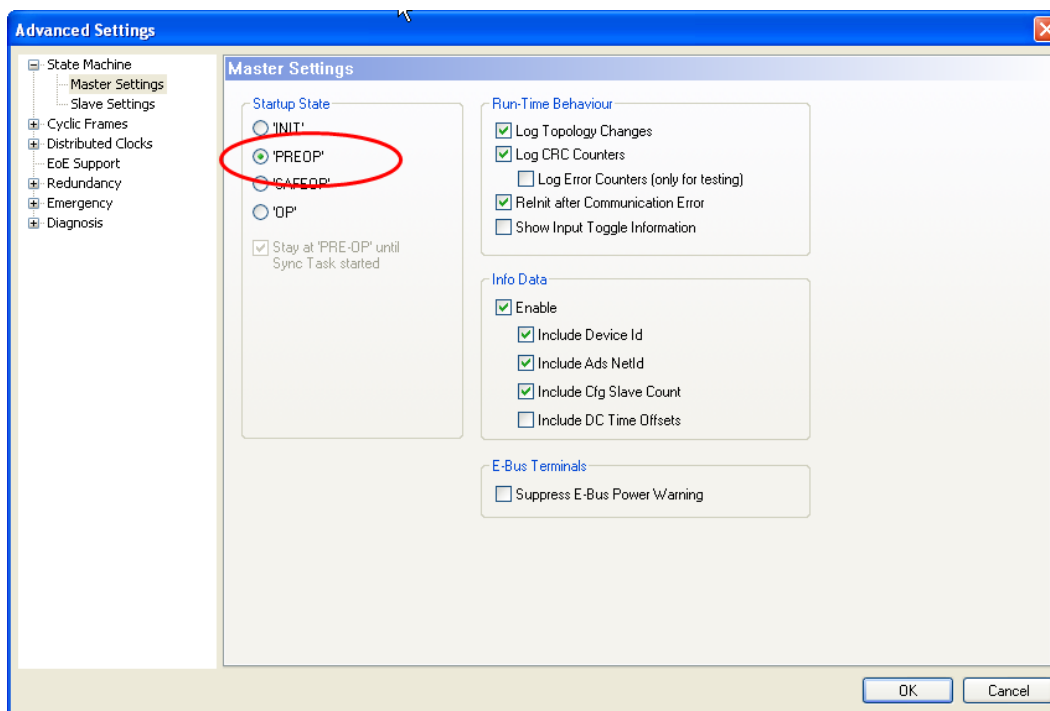
7. Click the EtherCAT tab, and select **Advanced Settings**.



8. From the explorer at the left side, select **Master Settings**.



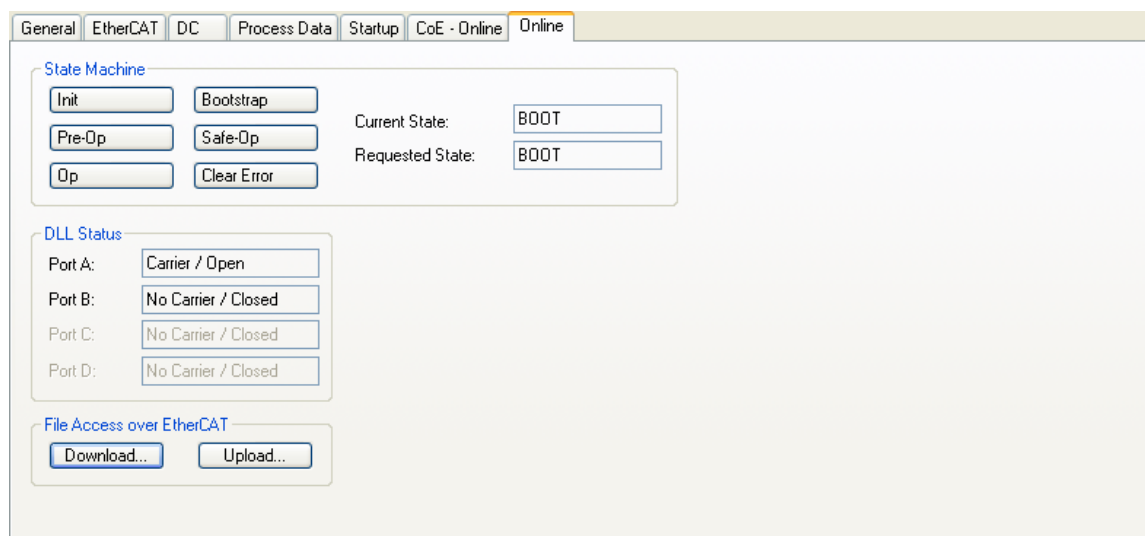
- Set the EtherCAT Master to the **PreOP** startup state.



4.4.3. The Firmware Download Procedure

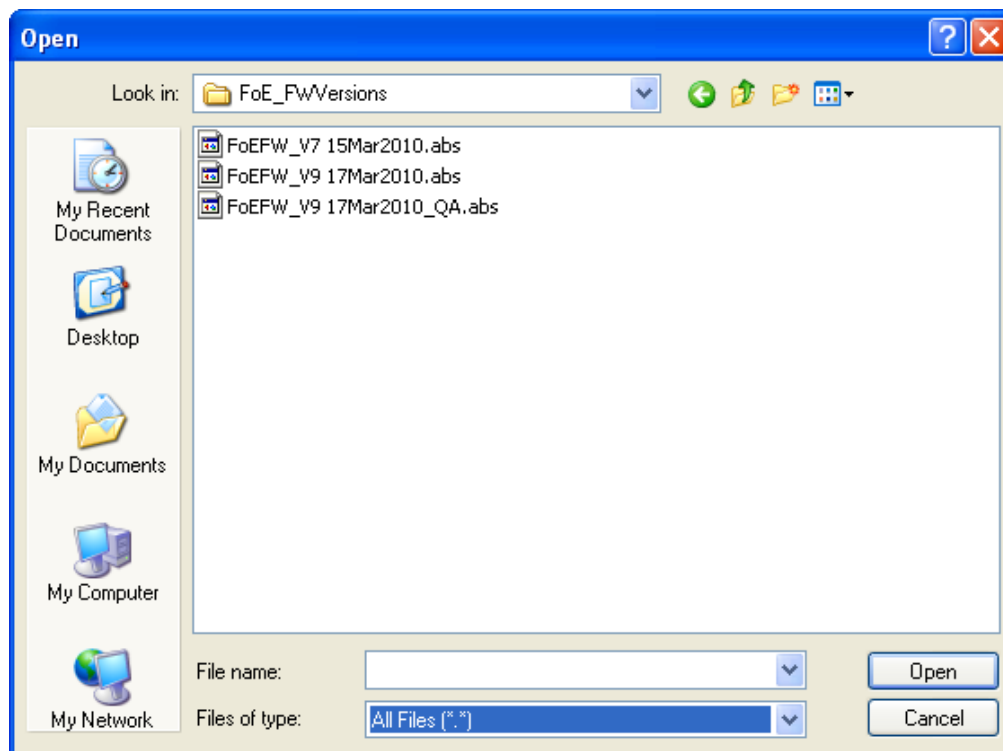
To perform firmware download

- Restart all the drives in the network (including the target drive for download). Select the **Online** tab, set the target drive state to **BOOT**, and then click **Download**.

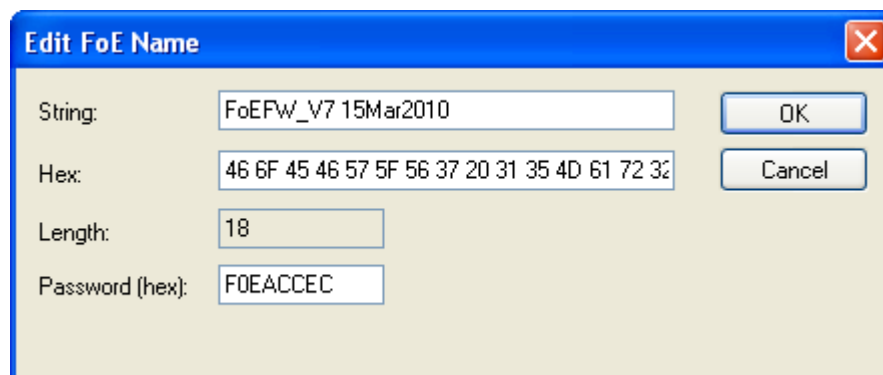


Note: If the state has not been changed to **BOOT**, wait 10 seconds and try again. If the state has still not changed, run: “set/reset TwinCAT to config mode”, wait 10 seconds and try again.

2. Click **Download** and an Open dialog box will appear. Change “Files of type” to “All Files”.

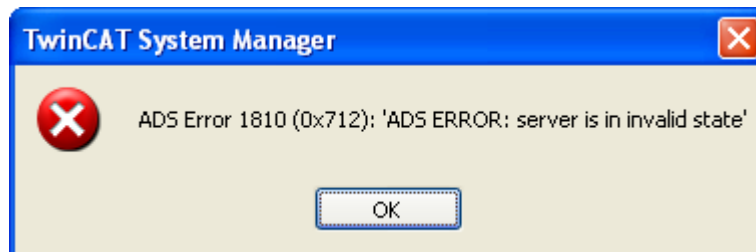


3. Select the file that will be downloaded (the filename prefix must start with 'FoEFW_V').
4. Click **Open**, and an Edit FoE Name dialog window appears.

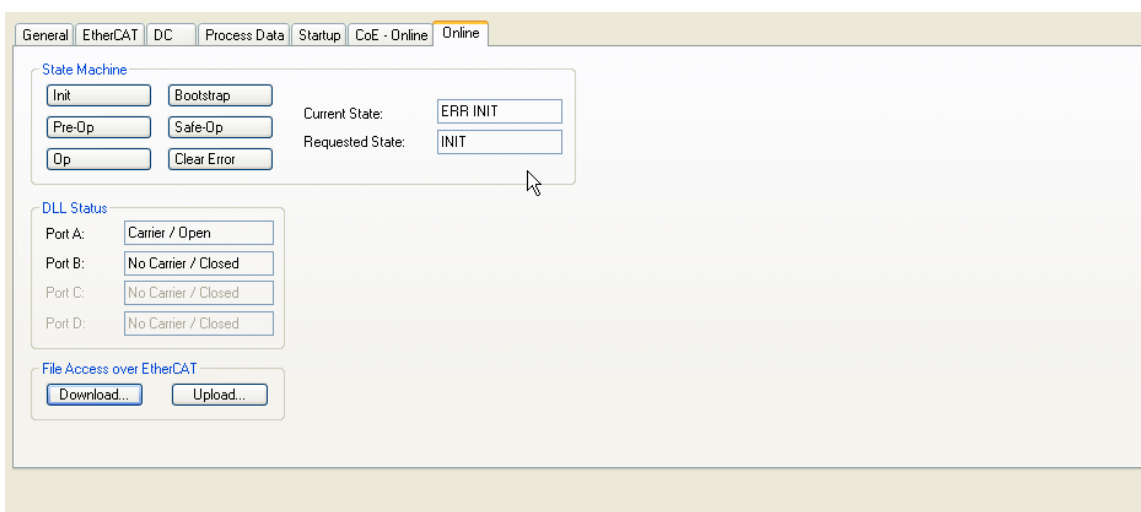


5. Enter the password: **F0EACCEC** and click **OK**. The download starts immediately.

6. After the download procedure is complete, do the following:
 - a. When the TwinCAT progress bar finishes, an error “init state” message appears. Click **OK**.



- b. Verify that the EtherCAT state has changed to **Err INIT**.



- c. Reset the drive's hardware. When a PLC program is running, the whole network of drives require a hardware reset.
 - d. Reload the TwinCAT I/O devices (press F4).

Chapter 5: Gold Drive Object list

This chapter describes the Gold drive supported objects in a concise table. For detailed information on the objects, refer to the relevant Elmo standard document. Both EtherCAT CoE protocol and CANopen use a similar object list, but some objects used in one, are not in the other. The EtherCAT CoE specific object list is listed in the section.

The following sub-section details the Complete Object List (Section 5.1), which includes CAN, EtherCAT, and the Common Objects. In contrast, section 5.2 details the CAN only Objects, and section 5.3, the EtherCAT only Objects list.

5.1. Complete Object List

UU: User defined Units

Cnt/sec: counts per second

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
1000	Device type	UINT32	RO	No	Return 0x192
1001	Error Register	UINT8	RO	No	
1002	Status Register	UINT32	RO	Yes	Similar to SR command
1003/16	Pre-Defined Error Field	UINT32	RO	No	Up to 16 last transmitted EMCY messages
1006	Communication cycle period	UINT32	RW	NO	
1008/16	Device Name	STRING	RO	No	Drive given name
1009	Hardware Version	STRING	RO	No	HW identification number
100A	Software Version	STRING	RO	No	Similar to VR command
100B	CANopen Node ID	UINT8	RO	No	Similar to PP[13] command
1016/3	Consumer heartbeat time	UINT32	RW	No	
1017	Producer heartbeat time	UINT16	RW	No	
1018	Identity Object	RECORD	RO	No	Used in LSS for drive identification
1023/3	OS command	RECORD	RW	No	See OS Interpreter chapter in DS301 Manual ##
1024	OS Command mode	UINT8	RW	No	

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
1029	Error Behavior	UINT8	RW	No	Lost of heartbeat communication response.
10E0	Device ID Reload	INT16	RW		ECAT only
10F1/2	SYNC monitor settings	UINT32	RW	No	
1400 – 1403/6	rPDO mapping communication parameters	UINT32	RW	No	Receive PDO mapping parameter of PDO1 to PDO4.
1600 – 1603/8	rPDO mapping parameters	UINT32	RW	No	Receive PDO mapping objects of PDO1 to PDO4. Up to 8 entries.
1800 - 1803/6	tPDO mapping communication parameters	UINT32	RW	No	
1A00 – 1A03/8	tPDO mapping parameters	UINT32	RW	No	
1C00	SM Communication type	UINT8	RW	No	
1C10	SM0 PDO assignment	REC	RW	No	NOT TO BE USED (CTT only)
1C11	SM1 PDO assignment	REC	RW	No	NOT TO BE USED (CTT only)
1C12	SM2 (Outputs) PDO assignment	REC	RW	No	
1C13	SM3 (Inputs) PDO assignment	REC	RW	No	
1C32/31	Sync Manager 2 parameters	UINT32	RW	No	ECAT Outputs
1C33/31	Sync Manager 3 parameters	UINT32	RW	No	ECAT Inputs
2012	Binary Interpreter reply object	UINT64	WO	Yes	Map to tPDO2
2013	Binary Interpreter receive object	UINT64	RO	Yes	Map to rPDO2
2020/3	Home Block limit parameters	UINT32	RW	No	
2030	Upload recording data	UINT64	RO	No	
2035	Upload data param	UINT32	RW		
2036	Upload data (UL)	UINT64	RO		
2041	Time stamp uSec resolution	UINT32	RO	Yes	
2045	Block upload Inhibit time parameter	UINT16	RW	No	
2046	Distributed clock inhibit time	INT16	RW	No	in mSec

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
2051	Download data (DL)	UINT64	WO	No	
207B/2	Additional Position range limit	INT32	RW	No	Modulo range
2081/4	Extended error code	INT32	RO	No	Reflects EE[]
2082	CAN controller status	UINT32	RO	Yes	
2085	Extra Status register	INT32	RO	Yes	
2090	CAN DF implementation	STRING	WO	No	
20A0	Additional Position in UU	INT32	RW	Yes	
20B0/10	Socket additional function	UINT32	RW	No	
20FD	Digital input (0x60FD alias)	INT32	RW	No	Allows write function
2205/1	Analog input	INT16	RO	Yes	In mV
2206	5V DC supply	INT16	RO	Yes	In mV
22A0	Digital output	UINT8	RW	Yes	GP output only
22A2	Drive Temperature in C	UINT16	RO	Yes	Legacy object
22A3/4	Temperature	UINT16	RO	Yes	Similar to TI[]
2E00	Position compensate index	UINT8	RW	Yes	
2E0A	Torque compensate value	UINT16	RW	No	Not in use
2E10	Set HOME Position according to last Touch Probe capture.	UINT16	RW	No	
2E15	Gantry YAW offset	INT32	RW	No	Reflected in TW[14]
2F00/25	General purpose User Integer array	INT32	RW	Yes	Reflects UI[]
2F01/25	General purpose User Float array	FLOAT	RW	Yes	Reflects UF[]
2F05	Get drive control board type	UINT16	RO	No	Similar to WS[8]
2F20/5	tPDO Asynchronous events	UINT32	RW	No	
2F41	Configuration object	UINT32	RW	No	
2F42	Internal use	UINT32	WO	No	Do not modify
2F45/4	Threshold parameter object	INT32	RW	No	
2F75	Extrapolation Cycles Timeout	INT16	RW	No	
6007	Abort connection code	UINT16	RW	No	
6040	Control word	UINT16	RW	Yes	

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
6041	Status word	UINT16	RO	Yes	
6502	Supported Drive Modes	UINT32	RO	No	
605A	Quick stop option code	INT16	RW	No	
605B	Shut down option code	INT16	RW	No	
605C	Disable operation option code	INT16	RW	No	
605D	Halt option code	INT16	RW	No	
605E	Fault reaction option code	INT16	RW	No	
6060	Modes of Operation	INT8	RW	Yes	
6061	Modes Of operation display	INT8	RO	No	
6062	Position demand value	INT32	RO	No	
6063	Actual Position internal units	INT32	RO	No	
6064	Actual Position in user units	INT32	RO	Yes	
6065	Position following error window	UINT32	RW	No	
6066	Position following error time out	UINT16	RW	No	
6067	Position Window	UINT32	RW	No	
6068	Position Windows time	UINT16	RW	No	
6069	Velocity sensor actual value cnt/sec	INT32	RO	Yes	
606A	Velocity sensor selection code	UINT16	RW	No	
606B	Velocity demand value (vel command)	INT32	RO	Yes	
606C	Velocity from speed or position sensor	INT32	RO	Yes	According to 606A
606D	Velocity window	UINT16	RW	No	
606E	Velocity window time	UINT16	RW	No	
606F	Velocity threshold	UINT16	RW	No	
6070	Velocity threshold time	UINT16	RW	No	
6071	Target - Torque	INT16	RW	Yes	
6072	Maximal torque	INT16	RW	No	
6073	Maximal current	INT16	RW	No	
6074	Torque Demand Value	INT16	RO	Yes	
6075	Motor rated current	UINT32	RW	No	

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
6076	Motor rated torque	UINT32	RW	No	
6077	Motor actual torque	INT16	RO	Yes	
6078	Motor actual current	INT16	RO	Yes	
6079	DC link voltage	UINT32	RO	Yes	
607A	Profile target Position	INT32	RW	Yes	
607B/3	Position range limit	INT32	RW	No	
607C	Home offset	INT32	RW	No	
607D/3	Software position limit	INT32	RW	No	
607E	Polarity (speed & position)	UINT8	RW	Yes	
607F	Max profile velocity	UINT32	RW	No	
6080	Max motor speed	UINT32	RW	No	
6081	Profile velocity	UINT32	RW	Yes	
6082	End velocity	UINT32	RW	Yes	
6083	Profile acceleration	UINT32	RW	Yes	
6084	Profile deceleration	UINT32	RW	Yes	
6085	Quick stop deceleration	UINT32	RW	Yes	
6086	Motion profile type	INT16	RW	No	
6087	Torque slope	UINT32	RW	Yes	
6089	Position notation index	INT8	RO	No	
608A	Position dimension index	UINT8	RO	No	
608B	Velocity notation index	INT8	RO	No	
608C	Velocity dimension index	UINT8	RO	No	
608D	Acceleration notation index	INT8	RO	No	
608E	Acceleration dimension index	UINT8	RO	No	
608F/3	Position encoder resolution	UINT32	RW	No	
6090/3	Velocity Encoder resolution	UINT32	RW	No	
6091/3	Gear ratio	UINT32	RW	No	
6092/3	Feed constant	UINT32	RW	No	
6093	Position factor	UINT32	RO	No	Compatibility

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
6094	Velocity encoder factor	UINT32	RO	No	Compatibility
6095	Velocity factor 1	UINT32	RO	No	Compatibility
6096	Velocity factor	UINT32	RW	No	
6097	Acceleration factor	UINT32	RW	No	
6098	Homing Method	INT8	RW	No	
6099/3	Home speed	UINT32	RW	No	
609A	Homing acceleration	UINT32	RW	No	
60B0	Position offset	INT32	RW	Yes	
60B1	Velocity offset	INT32	RW	Yes	
60B2	Torque offset	INT16	RW	Yes	
60B8	Touch probe function	UINT16	RW	Yes	
60B9	Touch probe status	UINT16	RO	No	
60BA	Touch probe pos1 positive value	INT32	RO	No	
60BB	Touch probe pos1 negative value	INT32	RO	No	
60BC	Touch probe pos2 positive value	INT32	RO	No	
60BD	Touch probe pos2 negative value	INT32	RO	No	
60C0	interpolated position mode	INT16	RW	No	
60C1/3	interpolated data record	INT32	RW	Yes	
60C2/3	interpolated time period	INT8	RW	Yes	
60C4/7	interpolation data configuration	INT16	RW	Yes	
60C5	Max acceleration	UINT32	RW	No	
60C6	Max deceleration	UINT32	RW	No	
60F2	Position option code	UINT16	RW	No	
60F4	Following error actual value	INT32	RO	No	
60FA	Control effort	INT32	RO	No	
60FC	Position demand value	INT32	RO	Yes	
60FD	Digital input	UINT32	RO	Yes	
60FE/3	Digital outputs	UINT32	RW	No	
60FF	target velocity	INT32	RW	Yes	

5.2. CAN Only Object List

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
1006	Communication cycle period	UINT32	RW	NO	
100B	CANopen Node ID	UINT8	RO	No	Similar to PP[13] command
1016/3	Consumer heartbeat time	UINT32	RW	No	
1017	Producer heartbeat time	UINT16	RW	No	
1023/3	OS command	RECORD	RW	No	See OS Interpreter chapter in DS301 Manual ##
1024	OS Command mode	UINT8	RW	No	
1029	Error Behavior	UINT8	RW	No	Lost of heartbeat communication response.
1400 – 1403/6	rPDO mapping communication parameters	UINT32	RW	No	Receive PDO mapping parameter of PDO1 to PDO4.
1600 – 1603/8	rPDO mapping parameters	UINT32	RW	No	Receive PDO mapping objects of PDO1 to PDO4. Up to 8 entries.
1800 - 1803/6	tPDO mapping communication parameters	UINT32	RW	No	
1A00 – 1A03/8	tPDO mapping parameters	UINT32	RW	No	
2012	Binary Interpreter reply object	UINT64	WO	Yes	Map to tPDO2
2013	Binary Interpreter receive object	UINT64	RO	Yes	Map to rPDO2
2030	Upload recording data	UINT64	RO	No	
2035	Upload data param	UINT32	RW		
2036	Upload data (UL)	UINT64	RO		
2041	Time stamp uSec resolution	UINT32	RO	Yes	
2045	Block upload Inhibit time parameter	UINT16	RW	No	
2051	Download data (DL)	UINT64	WO	No	
2082	CAN controller status	UINT32	RO	Yes	
2090	CAN DF implementation	STRING	WO	No	
2F20/5	tPDO Asynchronous events	UINT32	RW	No	

5.3. EtherCAT Only Object List

UU: User defined Units

Cnt/sec: counts per second

Object(Hex) Sub(Dec)	Name	Type	Attribute	Mappable?	Comment
10E0	Device ID Reload	INT16	RW	No	
10F1/2	SYNC monitor settings	UINT32	RW	No	
1400-14FF	See EtherCAT PDO list	UINT32	RW	No	
1600-16FF		UINT32	RW	No	
1800 – 18FF		UINT32	RW	No	
1A00 – 1AFF		UINT32	RW	No	
1C00	SM Communication type	UINT8	RW	No	
1C10	SM0 PDO assignment	REC	RW	No	NOT TO BE USED (CTT only)
1C11	SM1 PDO assignment	REC	RW	No	NOT TO BE USED (CTT only)
1C12	SM2 (Outputs) PDO assignment	REC	RW	No	
1C13	SM3 (Inputs) PDO assignment	REC	RW	No	
1C32/31	Sync Manager 2 parameters	UINT32	RW	No	ECAT Outputs
1C33/31	Sync Manager 3 parameters	UINT32	RW	No	ECAT Inputs
2046	Distributed clock inhibit time	INT16	RW	No	in mSec

5.4. EtherCAT CoE - PDO Objects list

The CoE includes a predefined static list of PDO mappings. The table below described the process data objects.

Note: The total bytes which can be mapped by the Gold line EtherCAT is 32 bytes. Objects between 0x1A00 (PDO1) to 0x1A09 are excluded from each other. Only one can be selected at a time.

The drive performs a single process in a cycle. In a situation where the same object is mapped twice, the last received object will be processed.

5.4.1. Transmit PDOs object list

PDO Index	Default Value	Bit Len	Description	Exclude
0x1A00	0x6064	32	Position Actual Value	0x1A01- 0x1A04
	0x60FD	32	Digital Inputs	
	0x6041	16	Status Word	
0x1A01	0x6064	32	Position Actual Value	0x1A00, 0x1A02- 0x1A04
	0x606B	32	Velocity Demand Value	
	0x6074	16	Torque Demand Value	
	0x6041	16	Status Word	
0x1A02	0x6064	32	Position Actual Value	0x1A00- 0x1A01, 0x01A03- 0x1A04
	0x6077	16	Torque actual Value	
	0x6041	16	Status Word	
	0x6061	8	Mode Of Operation Display	
0x1A03	0x6064	32	Position Actual Value	0x1A00- 0x1A02, 0x1A04
	0x60FD	32	Digital Inputs	
	0x606C	32	Velocity Actual Value	
	0x6041	16	Status Word	
0x1A04	0x6064	32	Position Actual Value	0x1A00- 0x1A03,
	0x60F4	32	Position Following error Value	
	0x6077	16	Torque Actual Value	
	0x6041	16	Status Word	
	0x6061	8	Mode Of Operation Display	
0x1A0A	0x6041	16	Status word	
0x1A0B	0x6061	8	Mode of operation display	
0x1A0C	0x6062	32	Position Demand [UU]	
0x1A0D	0x6063	32	Actual position [cnts]	

PDO Index	Default Value	Bit Len	Description	Exclude
0x1A0E	0x6064	32	Position actual value	
0x1A0F	0x6069	32	Velocity sensor actual value [cnt/sec]	
0x1A10	0x606B	32	Velocity demand [cnt/sec]	
0x1A11	0x606C	32	Velocity actual value	
0x1A12	0x6074	16	Torque demand value	
0x1A13	0x6077	16	Torque actual value	
0x1A14	0x60B9	16	Touch Probe status	
0x1A15	0x60BA	32	Touch Probe Pos1 Positive	
0x1A16	0x60BB	32	Touch Probe Pos1 Negative	
0x1A17	0x60BC	32	Touch Probe Pos 2 Positive	
0x1A18	0x6079	32	DC link circuit voltage	
0x1A19	0x60F4	32	Position Following error	
0x1A1A	0x60FA	32	Control Effort [cnt/sec]	
0x1A1B	0x60FC	32	Position Demand Value [cnt]	
0x1A1C	0x60FD	32	Digital Inputs	
0x1A1D	0x2205	16	Analog input	
0x1A1E	0x20A0	32	Auxiliary position value	
0x1A1F	0x6078	16	Current actual value	
0x1A20	0x60BD	32	Touch Probe Position 2 values	
0x1A21	0x2F45	32	Extra Status register	

5.4.2. Receive PDOs object list

PDO Index	Default Value	Bit Len	Description	Exclude
0x1600	0x607A	32	Target Position	0x1601 - 0x1606
	0x60FE	32	Digital outputs	
	0x6040	16	Control Word	
0x1601	0x6040	16	Control Word	0x1600, 0x1602 - 0x1606
	0x60FF	32	Target Velocity	
0x1602	0x6040	16	Control Word	0x1600,0x1601, 0x1603 - 0x1606
	0x6071	16	Target Torque	
0x1603	0x607A	32	Target Position	0x1600 - 0x1602, 0x1604 - 0x1606
	0x60FE	32	Digital outputs	
	0x60B1	32	Velocity offset	
	0x6040	16	Control Word	
0x1604	0x607A	32	Target Position	0x1600 - 0x1603.0x1605, 0x1606
	0x60FF	32	Target Velocity	
	0x6072	16	Max Torque	
	0x6040	16	Control Word	
0x1605	0x607A	32	Target Position	0x1600 - 0x1604. 0x1606
	0x60FF	32	Target Velocity	
	0x6071	16	Target Torque	
	0x6072	16	Max Torque	
	0x6040	16	Control Word	
	0x6060	16	Modes Of Operation	
0x1606	0x607A	32	Target Position	0x1600 - 0x1605
	0x60FF	32	Target Velocity	
	0x60FE	32	Digital outputs	
	0x60B1	32	Velocity offset	
	0x60B2	16	Torque Offset	
	0x6040	16	Control Word	
0x160A	0x6040	16	Control Word	
0x160B	0x6060	16	Modes Of Operation	
0x160C	0x6071	16	Target Torque	
0x160D	0x6072	16	Max Torque	
0x160E	0x6073	16	Max current	
0x160F	0x607A	32	Target Position	
0x1610	0x 607F	32	Max Profile Velocity	

PDO Index	Default Value	Bit Len	Description	Exclude
0x1611	0x 6081	32	Profile Velocity	
0x1612	0x6082	32	End Velocity	
0x1613	0x6083	32	Profile Acceleration	
0x1614	0x6084	32	Profile Deceleration	
0x1615	0x6087	32	Torque Slope	
0x1616	0x60B0	32	Position Offset	
0x1617	0x60B1	32	Velocity Offset	
0x1618	0x60B2	16	Torque Offset	
0x1619	0x60B8	16	Touch Probe Function	
0x161A	0x60C1	32	Interpolated data record (1)	
0x161B	0x60C1	32	Interpolated data record (2)	
0x161C	0x60FF	32	Target Velocity	
0x161D	0x60FE	32	Digital Output	
0x161E	0x607E	8	Polarity	

Chapter 6: Elmo Emergency Error and Abort List

This chapter describes the Elmo Emergency Error and Abort codes. An Emergency Error at the drive sends a callback event, which can be interpreted and solved. These error will vary in their definition and description. In addition, further Errors may be added to the list.

An Abort Error is different. This error can occur due to a misreading of a command, causing the drive to abort and send an Abort Code via a handshake protocol. They are fixed by the CANopen standardization, and cannot be edited or added.

6.1. Emergency Error Description

The following list the Emergency Error codes and their description with the relevant Elmo code where applicable.

Description	Error Code	Error Register	Elmo Error Code	Data Field
Object mapped to an RPDO returned an error during interpretation or a referenced motion failed to be performed.	6300	01		
Reserved	2311	03		
Short circuit: motor or its wiring may be defective, or drive is faulty.	2340	03		
Reserved.	8312	03		
Reserved	3100	05		
Under-voltage: power supply is shut down or it has too high an output impedance.	3120	05		
Over-voltage: power-supply voltage is too high or servo drive could not absorb kinetic energy while braking a load. A shunt resistor may be required.	3310	05		
Safety switch is sensed – drive in safety state	FF20	05		
Temperature: drive overheating. The environment is too hot or heat removal is not efficient. Could be due to large thermal resistance between drive and its mounting.	4310	09		

Description	Error Code	Error Register	Elmo Error Code	Data Field
CAN message lost (corrupted or overrun)	8110	11		
Heartbeat event	8130	11		
Protocol error (unrecognized NMT request)	8200	11		
Cannot start motor.	5400	21		
Disabled by switch	5441	21		
Disable by switch that is defined as external abort	5442	21		
Cannot start because of inconsistent database. The type of database inconsistency is reflected in the status SR report, and in the CD CPU dump report.	6320	21		
Motor stuck - the motor is powered but is not moving according to the definition of CL[2] and CL[3] .	7121	21		
Reserved.	7306	21		
Attempt to access a non-configured RPDO	8210	21		
The peak current has been exceeded. Possible reasons are drive malfunction or bad tuning of the current controller.	8311	21		
Position tracking error DV[3] - PX (UM=5) or DV[3] - PY (UM=4) exceeded position error limit ER[3] . This may occur due to: 1) Bad tuning of the position or speed controller 2) Too tight a position error tolerance 3) Abnormal motor load, or reaching a mechanical limit	8611	21		
DS 402 IP Underflow	FF02	21		
Reserved	1000	81		
Too large a difference in ECAM table.	5280	81		
Timing Error	5281	81		
Reserved	5282	81		

Description	Error Code	Error Register	Elmo Error Code	Data Field
Fatal CPU error: stack overflow	6180	81		
CPU exception - fatal exception. Something such as an attempt to divide in zero or another fatal firmware error has occurred. Use the CD command to get the CPU dump and report to your service center.	6181	81		
User program aborted by an error	6200	81		
1) Resolver feedback is not ready – Resolver angle was not found yet. 2) Analog encoder or Resolver feedback is either lost or with too low amplitude.	7300	81		
Feedback loss: no match between encoder and Hall location. Available in encoder + Hall feedback systems.	7380	81		
Two digital Hall sensors were changed at the same time. Error occurs because digital Hall sensors must be changed one at a time.	7381	81		
Failed to find the electrical zero of the motor in an attempt to start it with an incremental encoder and no digital Hall sensors. The reason may be that the applied motor current did not suffice for moving the motor from its position.	8380	81		
Speed tracking error DV[2] - VX (for UM=2 or UM=4, 5) exceeded speed error limit ER[2] . This may occur due to: 1) Bad tuning of the speed controller 2) Too tight a speed error tolerance 3) Inability of motor to accelerate to the required speed due to too low a line voltage or not a powerful enough motor	8480	81		
Speed limit exceeded: VX<LL[2] or VX>HL[2] . (Compatibility only)	8481	81		

Description	Error Code	Error Register	Elmo Error Code	Data Field
Position limit exceeded: PX<LL[3] or PX>HL[3] (UM=5), or PY<LL[3] or PY>HL[3] (UM=4). (Compatibility only)	8680	81		
Queue is low. Number of yet unexecuted PVT table rows has dropped below the value stated in MP[4] .	FF00	81	56	Field 1: Write pointer, Field 2: Read pointer
Write pointer is out of physical range ([1...64]) of PVT table. Reason may be an improper setting of MP[6] .	FF00	81	5b	Value of MP[6]
(Reserved for Compatibility reason)	FF00	81	5c	
An attempt has been made to program more PVT points than are available in queue.	FF00	81	34	Field 1: Index of PVT table entry that could not be programmed
Cannot initialize motion due to bad setup data. The write pointer is outside the range specified by the start and end pointers.	FF00	81	07	
Mode terminated and motor has been automatically stopped (in MO=1).	FF00	81	08	Data field 1: Write pointer; Data field 2: 1) End of trajectory in non-cyclic mode 2) A zero or negative time specified for a motion interval 3) Read pointer reached write pointer
A CAN message has been lost.	FF00	81	09	
Request by user program "emit" function	FF01	81		
Cannot start motor.	FF10	81		

6.2. Abort SDO Transfer Protocol

This protocol is used to implement the Abort SDO Transfer service.

Client to server or server to client

0	1	4	8
7...5	4...0		
cs = 4	x	m	d (data)

where:

cs Command specifier 4: Abort transfer request

x Not used; always 0.

m Multiplexor. Represents index/sub-index of SDO.

d Four-byte abort code giving reason for abort, encoded as Unsigned32 value.

The SDO Abort codes are listed in the following table:

Abort Code	Description
0503 0000h	Toggle bit not alternated.
0504 0001h	Invalid or unknown client/server command specifier.
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	Object does not exist in object dictionary.
0604 0041h	Object cannot be mapped to PDO.
0604 0042h	Number and length of objects to be mapped exceeds PDO length.
0604 0043h	General parameter incompatibility.
0604 0047h	General internal incompatibility in device.
0606 0000h	Access failed due to hardware error.
0607 0010h	Data type does not match, length of service parameter does not match.
0607 0012h	Data type does not match, service parameter too long.
0607 0013h	Data type does not match, service parameter too short.

Abort Code	Description
0609 0011h	Sub-index 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. When the abort code is 0800 0000h, the actual error can be retrieved using the EC command.
0800 0020h	Data cannot be transferred to or stored in application.
0800 0021h	Data cannot be transferred to or stored in application due to local control.
0800 0022h	Data cannot be transferred to or stored in application due to present device state.
0800 0023h	Object dictionary dynamic generation failed or no object dictionary is present (for example, object dictionary is generated from file and generation has failed due to a file error).