EtherCAT®
Fieldbus Interface

ServoPac

CAUTION

This is a specific manual describing the EtherCat® fieldbus interface.

It must be used in conjunction with the Installation, User and Programming manuals of the ServoPac drives.

Instructions for storage, use after storage, commissioning as well as all technical details require the MANDATORY reading of the manual before getting the amplifiers operational.

Maintenance procedures should be attempted only by highly skilled technicians having good knowledge of electronics and servo systems with variable speed (EN 60204-1 standard) and using proper test equipment.

The conformity with the standards and the "CE" approval is only valid if the items are installed according to the recommendations of the amplifier manuals. Connections are the user's responsibility if recommendations and drawings requirements are not met.

Any contact with electrical parts, even after power down, may involve physical damage.
Wait for at least 5 minutes after power down before handling the amplifiers (a residual voltage of several hundreds of volts may remain during a few minutes).

ESD INFORMATION (ElectroStatic Discharge)
TRANSTECHNIK amplifiers are conceived to be best protected against electrostatic discharges. However, some components are particularly sensitive and may be damaged if the amplifiers are not properly stored and handled.
STORAGE
- The amplifiers must be stored in their original package.
- When taken out of their package, they must be stored positioned on one of their flat metal surfaces and on a dissipating or electrostatically neutral support.
- Avoid any contact between the amplifier connectors and material with electrostatic potential (plastic film, polyester, carpet…).

HANDLING
- If no protection equipment is available (dissipating shoes or bracelets), the amplifiers must be handled via their metal housing.
- Never get in contact with the connectors.

ELIMINATION
In order to comply with the 2002/96/EC directive of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE), all TRANSTECHNIK devices have got a sticker symbolizing a crossed-out wheel dustbin as shown in Appendix IV of the 2002/96/EC Directive.
This symbol indicates that TRANSTECHNIK devices must be eliminated by selective disposal and not with standard waste.

TRANSTECHNIK does not assume any responsibility for any physical or material damage due to improper handling or wrong descriptions of the ordered items.
Any intervention on the items, which is not specified in the manual, will immediately cancel the warranty.

TRANSTECHNIK reserves the right to change any information contained in this manual without notice.

Contents

Chapter 1 - Introduction

The purpose of this manual is to provide the user with a support when designing an application and configuring the EtherCAT®[1] communication.

The EtherCAT® fieldbus interface allows the ServoPac drives to be used in an EtherCAT® communication network using the CANopen over EtherCAT® protocol. These drives do not support the Ethernet over EtherCAT® protocol.

In this manual, the EtherCAT® slave is supposed to be:

- ServoPac-ETH.

---

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

Chapter 2 - BASIC INFORMATION ABOUT EtherCAT®

EtherCAT® ("Ethernet for Controller and Automation Technology") is a real-time Ethernet-based fieldbus.

2.1 - MAIN ETHERCAT® SLAVE FUNCTIONS

The following EtherCAT® slave node reference model makes it easy to locate the impact of the EtherCAT® specific functions on the communication architecture.
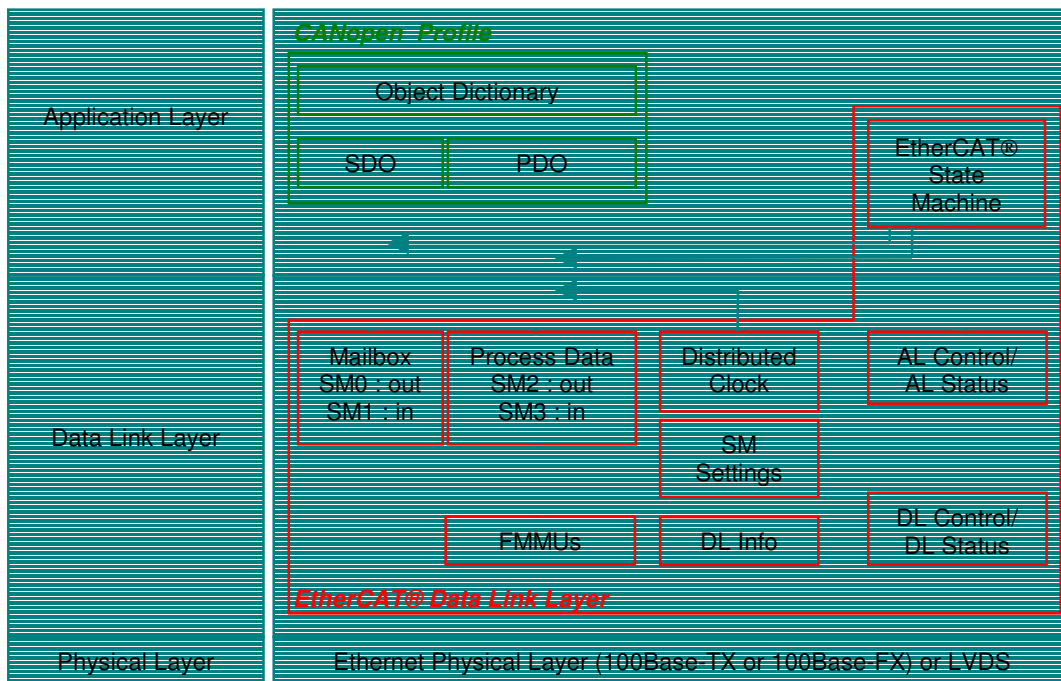
The green box contains the modules belonging to the CANopen communication profile. This profile is implemented in the ServoPac drives (please see "ServoPac User Guide" for more details about this communication profile).
The red box contains the modules that belong to the EtherCAT® Slave Controller (ESC) on the ServoPac drive.

The EtherCAT® frame is initiated by the EtherCAT® master which sends this frame to the first EtherCAT® slave. An EtherCAT® frame can contain several telegrams. If one of the telegrams is assigned the first slave, it will process this telegram by inserting or extracting data respectively in or from the frame. The frame is then transmitted to the next EtherCAT® slave which makes the same. The last EtherCAT® slave receiving the frame, will also process it, and then returns the frame to the master via all slaves. To perform such a processing, two full-duplex communication ports are controlled.

2.2 - ETHERCAT® STATE MACHINE

The EtherCAT® master controls the following state machine in every slave:
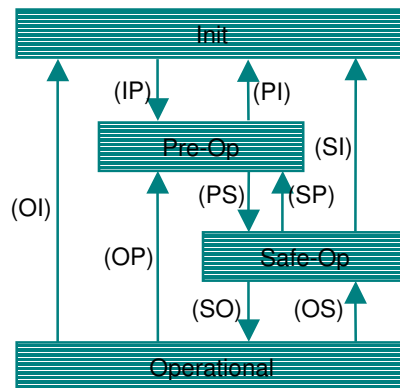
Description of the states and the transitions:

State	Description
Init	Initial state at the power on. Amplifier configuration is performed. No communication.
Pre-Op	SM0 and SM1 are active : SDO exchange is possible.
Safe-Op	SM0, SM1, SM3 are active : SDO and TPDO exchange is possible. DC generates event.
Operational	SM0, SM1, SM2 and SM3 are active : SDO, TPDO and RPDO exchange is possible. DC generates

Transition	Actions
IP	Start SDO
PI	Stop SDO
PS	Start TPDO
SP	Stop TPDO
SO	Start RPDO
OS	Stop RPDO
OP	Stop RPDO and TPDO
SI	Stop TPDO and SDO
OI	Stop RPDO, TPDO and SDO

The internal register called "AL Status Code" (Address 0x134:0x135) indicates the eventual error cause of an unsuccessful transition. This code can be read by using the object called "Extension Board Access" with index 0x2320, sub-index 0. The following table lists the supported AL status error codes:

Code	Description	Transition	Resulting state
0x0000	no error	any	current state
0x0011	invalid request change	I->S, I->O, P->O	current state
0x0012	unknown request change		current state
0x0013	bootstrap not supported	I->B	I
0x0016	invalid mailbox configuration	I->P	I
0x001D	invalid output configuration	0, S, P->S	S
0x001E	invalid input configuration	0, S, P->S	P
0x0026	inconsistent settings	P->S	P
0x0029	free run needs 3-buffer mode	P->S	P
0x0030	invalid DC SYNCH configuration	0, S	S
0x0036	invalid DC SYNC0 cycle time	0, S	S
0x8000	mother board watchdog expired	I->P	I

2.3 - ETHERCAT® FIELDBUS TOPOLOGY

In motion control applications, EtherCAT® slaves are connected together in one line. The physical layer technology employed by the EtherCAT® fieldbus is the 100BASE-TX. The EtherCAT® master is connected at one end of the line.
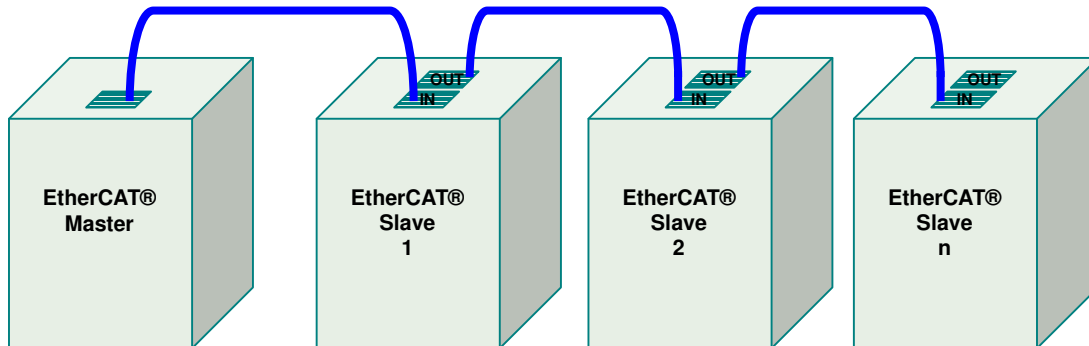
RJ-45 sockets on EtherCAT® slaves are labelled "IN" and "OUT".
The EtherCAT® master should be connected to the first EtherCAT® slave using the "IN" RJ-45 socket of the EtherCAT® slave.

The cables used for an EtherCAT® network need to be of category 5e quality. Maximum cable length should be 100 m. The following table indicates the pin function and the wire colors (see T 568-B standard for more information).
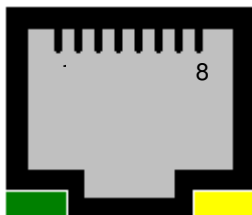
Pin	T568B Color	Function
1	white/orange stripe	TxData +
2	orange solid	TxData -
3	white/green stripe	RecvData +
4	blue solid	Not used
5	white/blue stripe	Not used
6	green solid	RecvData -
7	white/brown stripe	Not used
8	brown solid	Not used

Chapter 3 - Electrical installation

3.1 - CONNECTORS

On the ServoPac drive, the EtherCAT® fieldbus connectors X6 and X7 are located on the top of the drive.

3.1.1 "IN" CONNECTOR: X6

PIN	SIGNAL	DESCRIPTION
1	Tx Data+	Differential signals
2	Tx Data-	
3	Rx Data+	Differential signals
6	Rx Data-	
Others		Reserved

3.1.2 "OUT" CONNECTORS: X7

PIN	SIGNAL	DESCRIPTION
1	Tx Data+	Differential signals
2	Tx Data-	
3	Rx Data+	Differential signals
6	Rx Data-	
Others		Reserved

⚠ On each EtherCAT® slave, the incoming cable should always be connected to the "IN" RJ-45 socket and the outcoming should also be connected to the "OUT" RJ-45 socket.

3.2 - INDICATORS

Four LEDs are available for quick diagnosis purposes

3.2.1 - LINK/ACTIVITY INDICATORS

These two LEDs are integrated in RJ-45 sockets.

Label name	Color	Function	Link	Activity	LED coding
L/A IN	Green	Link and activity on the IN port	Yes	No	On
L/A OUT	Green	Link and activity on the OUT port	Yes	Yes	Flickering
			No	(Not Applicable)	Off

3.2.2 - RUN INDICATOR

Label	Color	Function	State	LED coding
RUN	Green	State of the EtherCAT state machine	Init	Off
			Pre-Op	Blinking
			Safe-Op	Single Flash
			Operational	On

3.2.3 - ERROR INDICATOR

Label	Color	Function	State	LED coding
ERR	Red	Errors	No error	Off
			Booting error	Flickering
			Invalid Configuration	Blinking
			Unsollicited state change	Single Flash
			Application watchdog timeout	Double Flash

3.3 SLAVE ADDRESSING MODES

The EtherCAT® master has the possibility to use 3 addressing modes:

Logical addressing: this addressing mode is only and mandatory used for the process data exchange (PDOs).

Position addressing: this addressing mode is used for example during the start up of the EtherCAT® system to scan the fieldbus and during the distributed clocks compensation.

Node addressing: This addressing mode is used for the setup of the communication parameters of the ESC and also during the SDO exchanges. Two addresses are available in this mode:
- The configured station address is assigned by the master during start up and cannot be modified by the slave;
- The configured station alias is assigned by the slave but needs to be enabled by the EtherCAT® master.

The 6 switches labelled "node ID" on the amplifier front panel are decoded at power-on of the drive and stored in the object Device ID (index 200Ah sub-index 1) and in the configured station alias.

Chapter 4 - Specifications of the SDOs and PDOs for the EtherCAT® Interface

All four Sync Manager channels need to be setup by the EtherCAT® master during the communication initialisation phases according to the device description file "Transtechnik_TT-ETH_Vx_x.xml".

4.1 - SDO

All of the ServoPac parameters can be accessed through the SDO communication channel which is carried in the EtherCAT® telegrams. Please see "ServoPac User Guide" for more details about the available objects. SDO services encapsulated into the EtherCAT® telegrams are only 8 byte long.

SPECIFIC OBJECTS OF THE ETHERCAT®

Index	Object code	Name	Type	Attr.
0x1C00	ARRAY	Sync Manager Communication type	Integer8	ro
0x1C10	VAR	Sync Manager channel 0 PDO Assignment	Integer8	ro
0x1C11	VAR	Sync Manager channel 1 PDO Assignment	Integer8	ro
0x1C12	ARRAY	Sync Manager channel 2 PDO Assignment	Unsigned16	rw
0x1C13	ARRAY	Sync Manager channel 3 PDO Assignment	Unsigned16	rw

Manufacturer Specific Objects:

Index	Object code	Name	Type	Attr.
0x2012	VAR	Synchronisation clock object	Unsigned16	rw
0x2320	VAR	Read access to the ESC registers	Unsigned16	rw
0x2321	VAR	RPDO Watchdog time	Unsigned32	rw
0x3022	ARRAY	Error word	Unsigned16	ro
0x3024	VAR	Warning code	Unsigned32	ro

4.1.1 – SYNC MANAGER COMMUNICATION TYPE

Index	0x1C00
Name	Sync Manager Communication type
Object Code	ARRAY
Number of elements	0…4

Value description

Sub Index	0
Name	Number of Sync Manager
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	4

Sub Index	1
Name	Communication type Sync Manager channel 1
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	1 : mailbox receive (master to slave)

Sub Index	2
Name	Communication type Sync Manager channel 2
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	2 : mailbox send (slave to master)

Sub Index	3
Name	Communication type Sync Manager channel 3
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	3 : process data output (master to slave)

Sub Index	4
Name	Communication type Sync Manager channel 4
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	4 : process data input (slave to master)

4.1.2 – SYNC MANAGER CHANNEL 0 PDO ASSIGNMENT

Value description

Index	0x1C10
Name	Sync Manager channel 0 PDO assignment
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	0 : no PDO assigned

4.1.3 – SYNC MANAGER CHANNEL 1 PDO ASSIGNMENT

Value description

Index	0x1C11
Name	Sync Manager channel 1 PDO assignment
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	ro
PDO Mapping	No
Default Value	0 : no PDO assigned

4.1.4 – SYNC MANAGER CHANNEL 2 PDO ASSIGNMENT

Index	0x1C12
Name	Sync Manager channel 2 PDO assignment
Object Code	ARRAY
Number of elements	0…4

Value description

Sub Index	0
Name	Number of assigned RxPDOs
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	rw
PDO Mapping	No
Default Value	4

Sub Index	1
Name	PDO mapping object index of assigned RxPDO
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0x1600 : RxPDO1

4.1.5 – SYNC MANAGER CHANNEL 3 PDO ASSIGNMENT

Index	0x1C13
Name	Sync Manager channel 3 PDO assignment
Object Code	ARRAY
Number of elements	0…4

Value description

Sub Index	0
Name	Number of assigned TxPDOs
Object Code	VAR
Data Type	Integer8
Object Class	all
Access	rw
PDO Mapping	No
Default Value	4

Sub Index	1
Name	PDO mapping object index of assigned TxPDO
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0x1A00 : TxPDO1

4.1.6 – SYNCHRONISATION CLOCK OBJECT

Index	0x2012
Name	Synchronisation Type
Object Code	ARRAY
Number of elements	1

Value description

Sub Index	1
Name	Synchronisation clock
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	4

Synchronisation clock	Description
0	No synchronisation of the servo loop; no PDO mapping
1	No synchronisation of the servo loop; PDO mapping performed
2	Soft synchronisation of the servo loop; PDO mapping performed
4	Hard synchronisation of the servo loop; PDO mapping performed

4.1.7 – READ ACCESS TO THE FIELDBUS CONTROLLER REGISTERS

Value description

Index	0x2320
Name	Read access to the Fieldbus controller (ESC) registers
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0

ESC register read sequence has to be performed as follows:
1- write into the object (index 0x2320 sub-index 0) the address of the ESC register to be read
2- read from the object the data stored at the previously written address

4.1.8 – RPDO WATCHDOG TIME

Value description

Index	0x2321
Name	RPDO watchdog time
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0 (0: not used)
Unit	milliseconds

Starting at the reception of the first RPDO, if no RPDO is received after the RPDO watchdog time, bit 13 of the error monitoring object value (index 3022h sub-index 2) is set to one. The EtherCAT® slave autonomously changes to the Safe-Op state.

4.1.9 – FIELDBUS SPECIFIC ERROR BITS IN THE ERROR CODE 1 OBJECT (0x3022)

Index	0x3022
Name	Error word
Object Code	ARRAY
Number of Elements	3

Value Description

This object contains two 32-bit words in which one bit is assigned to different errors.
The Error code is the value which will be sent as an emergency message (EMCY). Only bit 13 of the value of sub index 2 is detailed in this manual:

Sub Index	2
Description	Error monitoring
Data Type	Unsigned32
Object Class	all
Access	ro
PDO Mapping	No
Value	See below
Default value	No

Bit	Value	Error Code	Function
13	0x00002000	46	RPDO watchdog overflow

4.1.10 – FIELDBUS SPECIFIC WARNING BITS IN THE WARNING CODE OBJECT (0x3024)

Value description

Index	0x3024
Name	Warning Code
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	ro
PDO Mapping	Possible
Default Value	0

Only bits 13 and 14 of the value of the object are detailed in this manual:

Bit	Mask	Function
13	0x00002000	EtherCAT® firmware not compatible
14	0x00004000	Fieldbus Controler (ESC) not accessible

4.2 - PDO PARTICULARITIES

The Sync Manager PDO Assign objects need to be setup in addition to the usual xPDO mapping objects.

Principle of PDO mapping:

CANopen usual PDO mapping setup

1rst Receive PDO mapping (object 1600h)		
0	2 (nb of entries)	
1	yyyyh (index)	yyh (sub-index)	20h (size)
2	xxxxh	xxh	10h

2nd Receive PDO mapping (object 1601h)		
0	1 (nb of entries)	
1	zzzzh (index)	zzh (sub-index)	20h (size)

Object Dictionary		
xxxxh	xxh	Application object 1
yyyyh	yyh	Application object 2
zzzzh	zzh	Application object 3

EtherCAT® specific Sync Manager PDO assignment

Sync Manager Channel 2 PDO Assignment (object 1C12h)	
0	2 (nb of entries)
1	1601h (index)
2	1600h

RPDO2	RDPO1	
Application object 3	Application object 2	Application object 1

These setups can be performed by using the SDO communication channels either with the **Gem Drive Studio** software or with an EtherCAT® master.

⚠ Sync Manager Channel 2 should be setup to process at least one RPDO.

Chapter 5 - Synchronisation of ServoPac EtherCAT® slaves

The EtherCAT® fieldbus is well-suitable for multiaxis applications with time-critical constraints.

An internal software PLL is used in the ServoPac drive to synchronize the servo loops with a fieldbus-based clock source.

Before entering the Safe-Op and Op states, the communication cycle period object (1006h) and the synchronisation clock object (2012h) need to be carefully setup according to the slave's setup performed by the EtherCAT® master. These objects are detailed in the "ServoPac User Guide".

SYNC TYPE 0: SYNCHRONISATION AND PROCESS DATA EXCHANGE DISABLED

This is the default value of the synchronisation type at power-on of the drive. The servo loop is not synchronized with the fieldbus. No PDO mapping is possible.

SYNC TYPE 1: SYNCHRONISATION DISABLED AND PROCESS DATA EXCHANGE ENABLED

No synchronisation of the servo loops takes place whereas the PDO mapping is performed.

SYNC TYPE 2: SOFT SYNCHRONISATION

A clock event is generated every time an EtherCAT® telegram which carries RPDO is received by the ServoPac drive. This clock event is the source of the internal software PLL to synchronize the servo loops. This clock event is physically generated by the ESC Sync Manager channel 2 interruption.
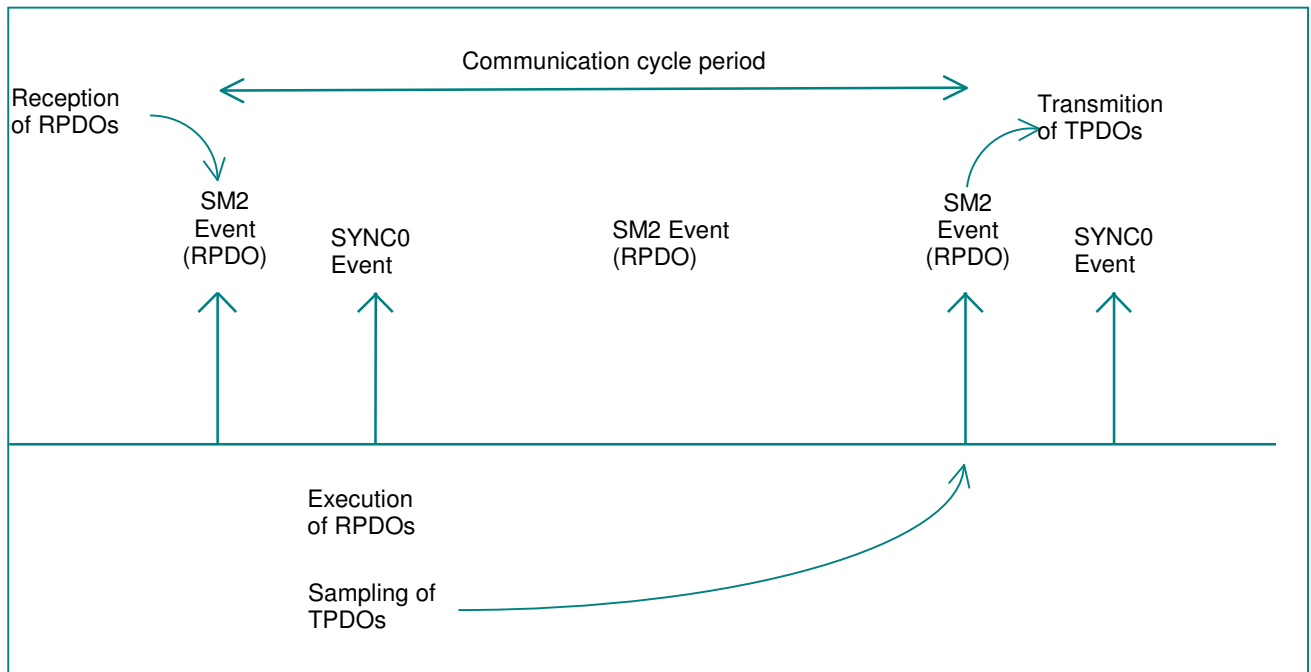The PDO mapping is performed in the following way:
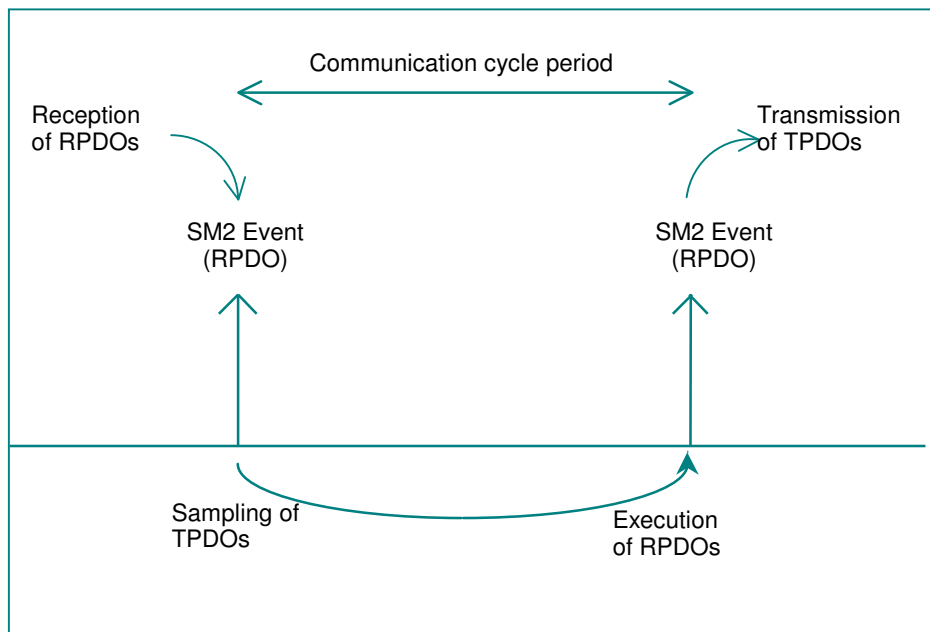
In this synchronisation type, the EtherCAT® master is supposed to embed TPDO and RPDO in the same EtherCAT® telegram.
The minimum communication cycle period with the soft synchronisation type using the EtherCAT® fieldbus is 2 ms.

> ⚠ With this type of synchronisation, distributed clocks should not be enabled by the master, otherwise the transition Preop to Safe-Op will not be performed and the AL Status Code will be set to 26h, meaning an inconsistent setting.

The distributed clock is the source of the internal software PLL to synchronize the servo loops. This clock event is physically the SYNC0 signal generated by the distributed clock of the ESC.
The PDO mapping is performed in the following way:

In this synchronisation type, the EtherCAT® master is also supposed to embed TPDO and RPDO in the same EtherCAT® telegram.

The EtherCAT® master is responsible for the setup of the SYNC0 cycle time parameter during the communication start-up phase (PS transition).

With this type of synchronisation care must be taken during the setup of the master :
Distributed clocks should be enabled by the master.
The SYNC0 cycle time and the cycle task communication period needs to be equal to the communication cycle period object 1006h.
If one of these conditions is not fulfilled, the transition Preop to Safe-Op will not be performed and the AL Status Code will be set to 26h, meaning an inconsistent setting.