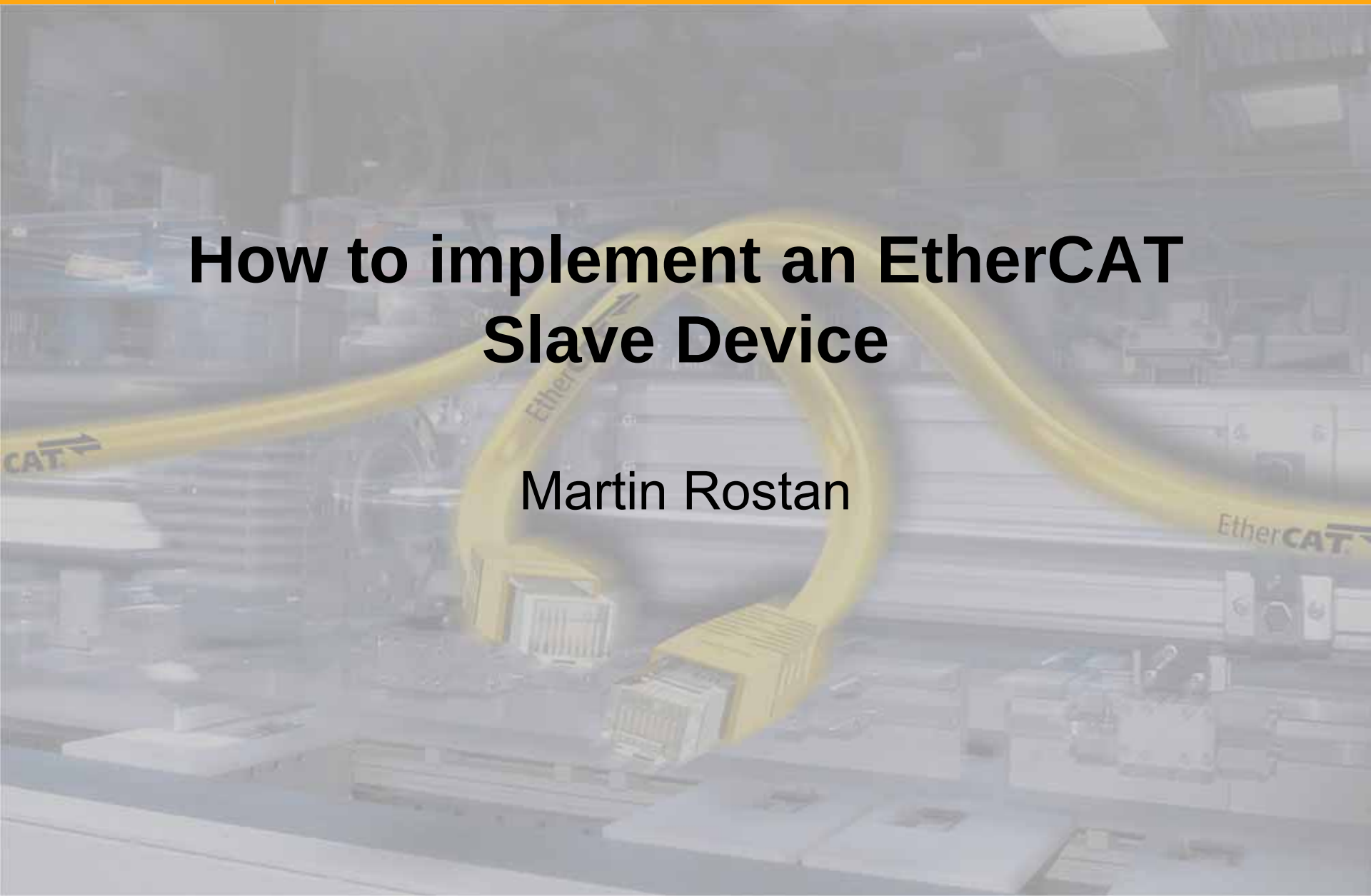
EtherCAT Technology Group

How to implement an EtherCAT Slave Device

Martin Rostan



Agenda

EtherCAT Slave Structure

Device Definition

- Integrated or Interface Device
- Hardware Selection
- Device Profile
- Process Data
- Synchronization

HW Design

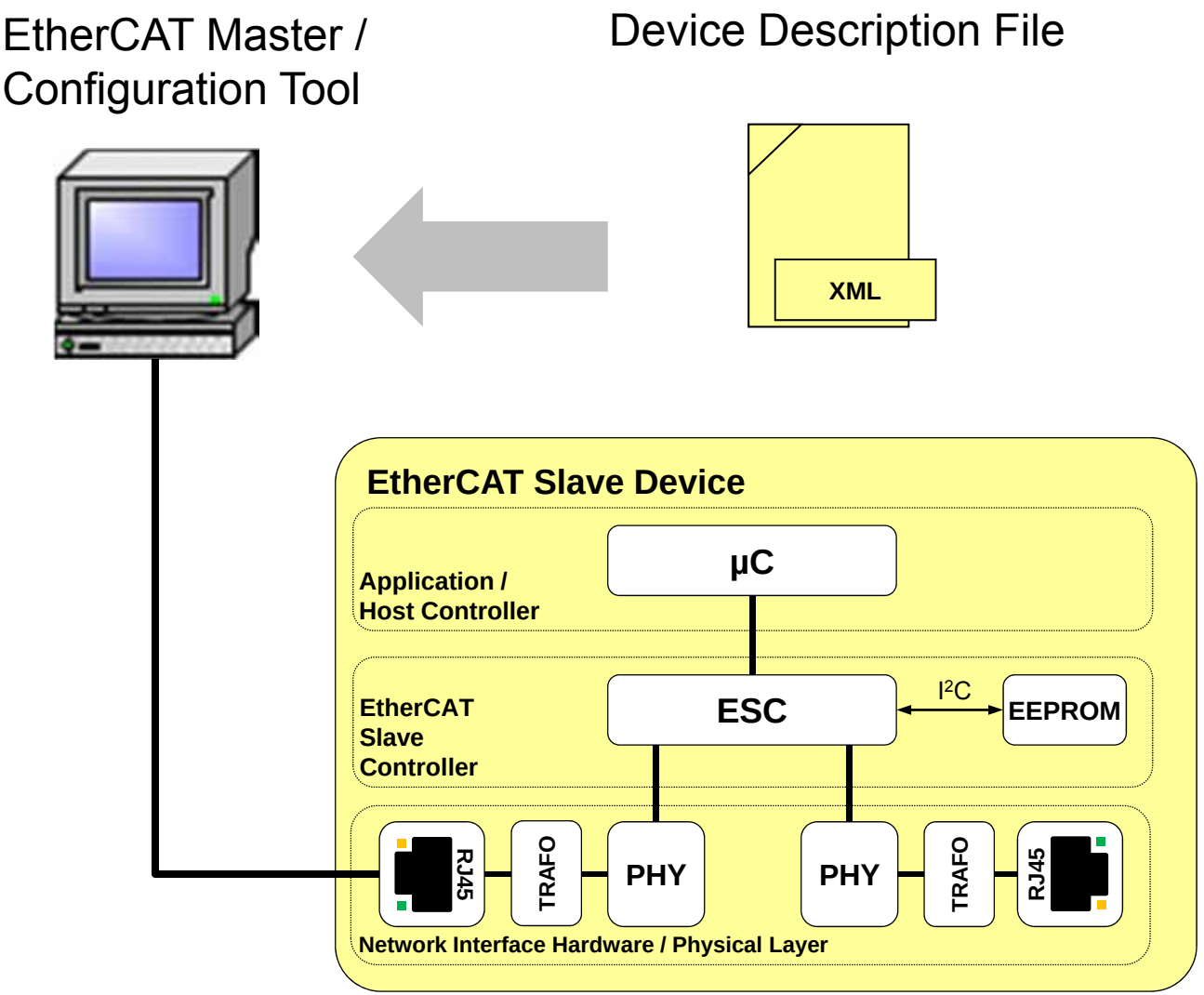
SW Development

Conformance Testing

Common Issues – and how to avoid them

1. EtherCAT Slave Structure Overview
2. First Steps: Device Definition
3. Hardware Design
4. Software Development
5. Conformance Testing
6. Common Issues – and how to avoid them

EtherCAT Slave Structure Overview



First: Device Definition

Define / Select:

1. Fully integrated Design or Interfacing Device
2. Interface Hardware
3. Device Profile
4. Parameter + Process Data
5. Synchronization and Time Stamping Requirements

Fully integrated or Interfacing Device?

Fully integrated

PRO:

- Lower hardware costs
- Most flexible solution
- Full control of all features

CON:

- Higher development costs

Interfacing device

PRO:

- Lower development costs
- Time to market
- Less network know-how required

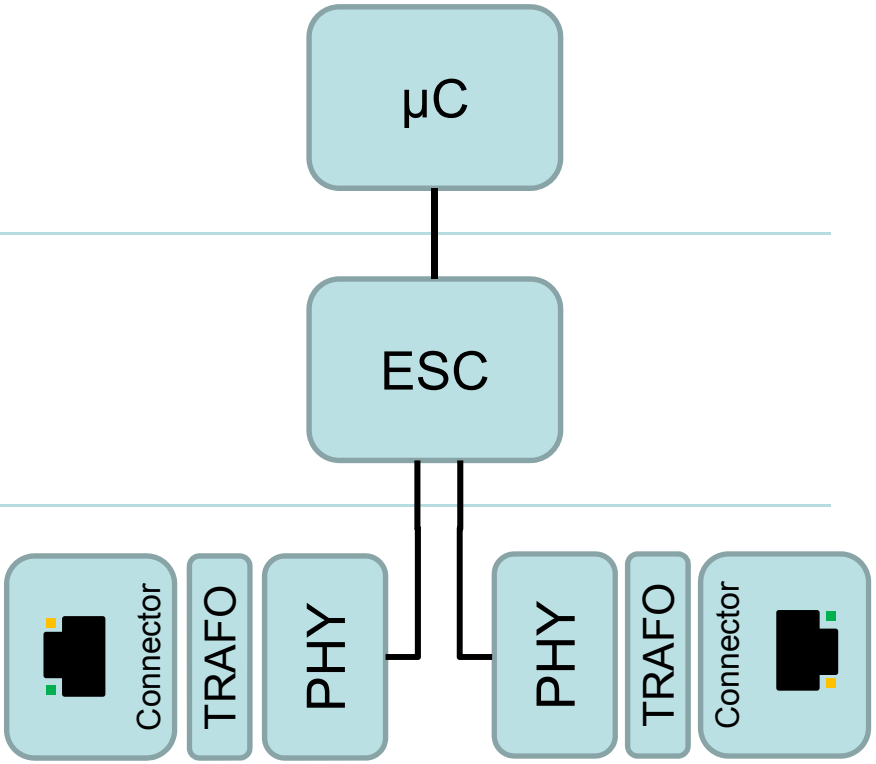
CON:

- Higher hardware costs
- Form factor restrictions

Interface Hardware?

Fully integrated Design*:

- Host / Application Controller
- EtherCAT Slave Controller
- Physical Layer, Network Interface



* Interfacing Device Hardware Selection: no generic rules due to the diverse architectures of the various solutions

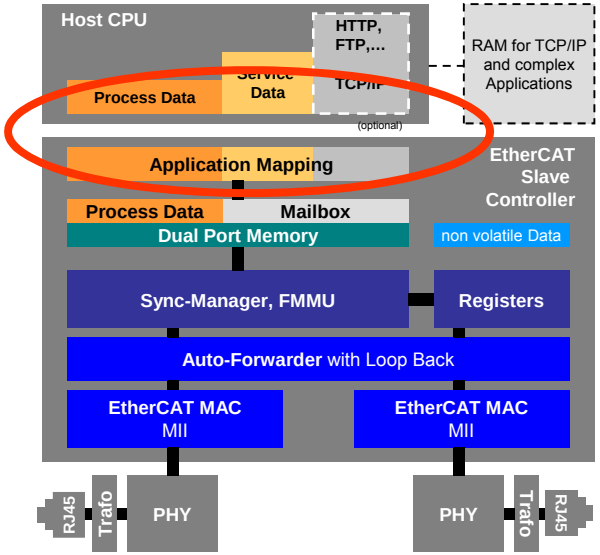
EtherCAT Host Controller?

- Simple (I/O) Devices do not require a µC at all
- Tasks of Host µC in more complex devices:
  - Process data – Exchange with the Application
  - Object Dictionary Handling
  - Handling of Application Parameter (Communication Parameter are handled by ESC)
  - TCP/IP Stack Handling – if required

- Host Controller Performance is determined by Device Application, not by EtherCAT
  → In many cases an 8bit µC is sufficient

EtherCAT Host Controller Interface?

The host controller may determine the interface to internal DPRAM of the EtherCAT Slave Controller



Example: Beckhoff ASICs:

- 8/16 Bit µC Interface
  - Demultiplexed
  - Intel Signal Types
  - Polarity configurable (BUSY, INT)
  - Typical µC: ARM, Infineon 80C16x, Hitachi SH1, ST10, TI TMS320 Series, …

- Serial – Interface (SPI)
  - Up to 10 MBaud
  - µC is SPI Master
  - Typical µC: Microchip PIC, DSPic, Intel 80C51, Atmel AVR…

EtherCAT Slave Controller?

ASIC

PRO:

- Low costs, small
- netx: Multiple networks supported

CON:

- Less flexible

FPGA

PRO:

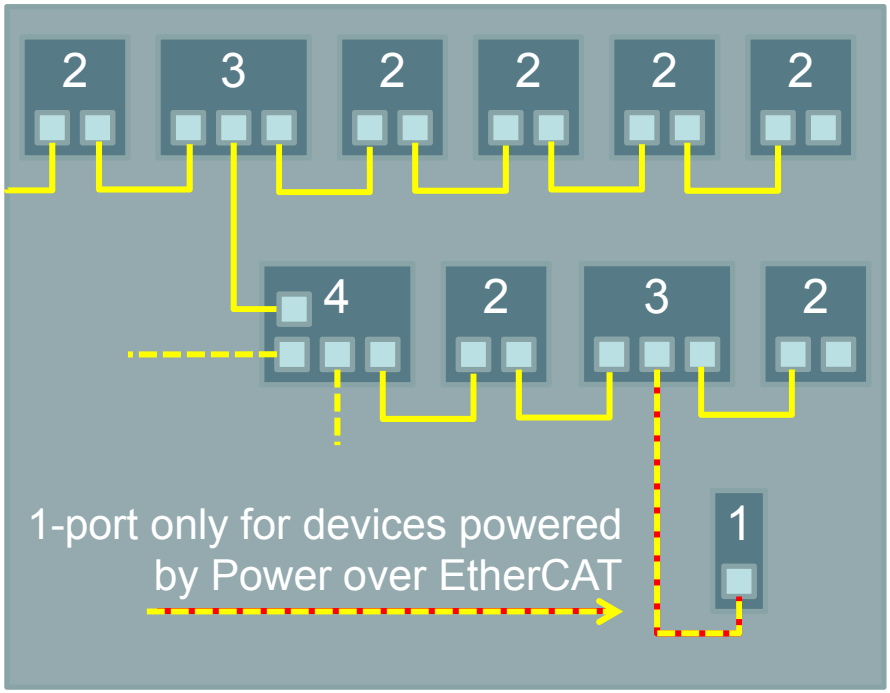
- Most flexible: FPGA can integrate application functionality as well
- Low costs especially if FPGA is used anyhow
- Can support multiple Ethernet flavors

CON:

- Requires VHDL programming know-how

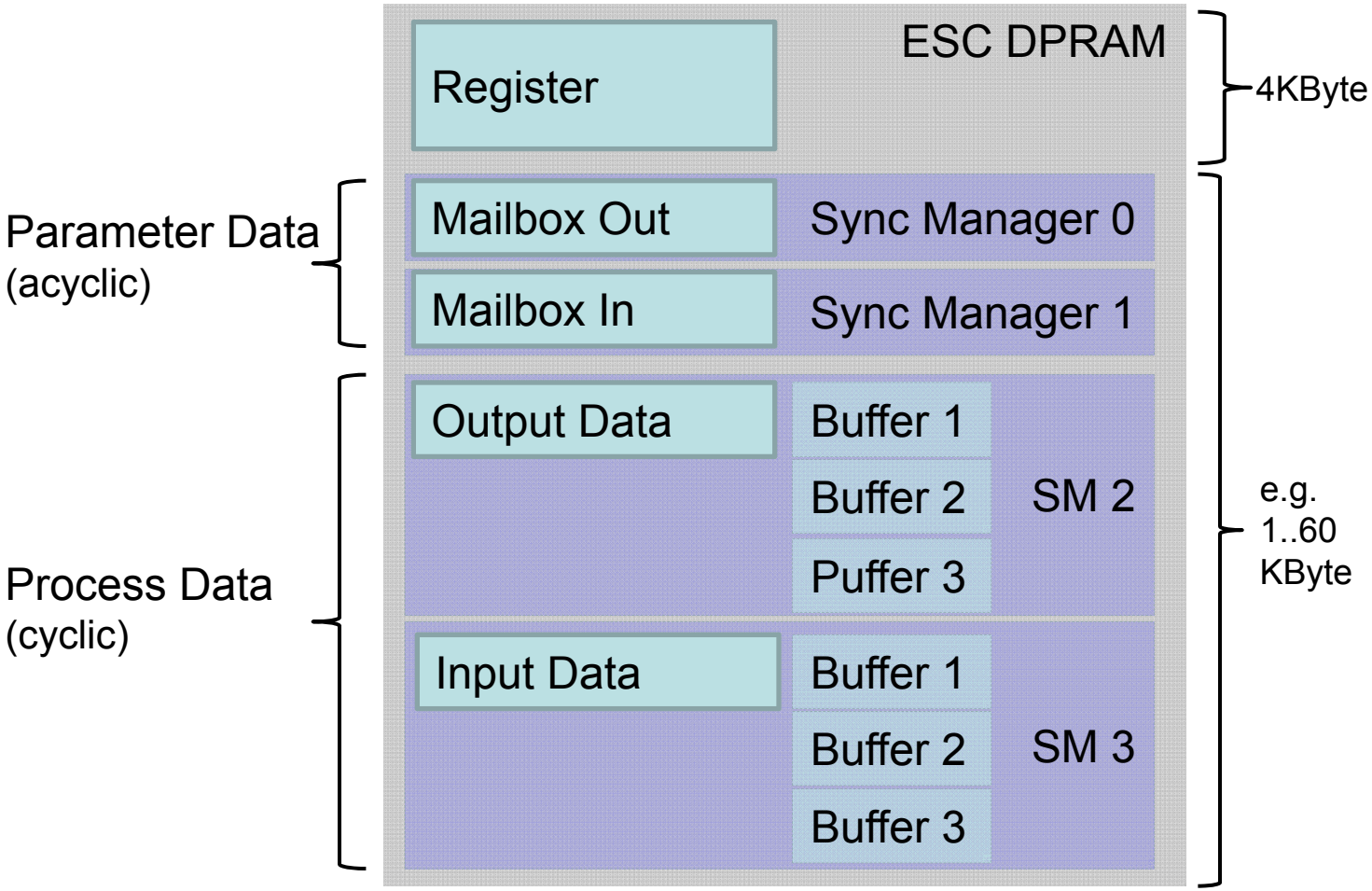
Selection Criteria EtherCAT Slave Controller

- No (and type) of Ports
- Typical: 2-port devices, for line and ring topologies
- 3+4-port devices cater for topology options



Selection Criteria EtherCAT Slave Controller

- Size of DPRAM and no. of Sync Manager entities



Selection Criteria EtherCAT Slave Controller

- No. of Fieldbus Memory Management Units (FMMU)
- FMMU: copies process data from EtherCAT datagram to DPRAM – and ensures data consistency
- Mechanism for further optimization of resources (bandwidth, CPU power)
- Typical requirement: minimum of 3.

FMMU Number	Usage
1	Output Data
2	Input Data
3	Status check of Mailbox Response

Selection Criteria EtherCAT Slave Controller

- Price?
- Local Support?
- Housing?
- Size?
- Integrated PHYs?
  - Hilscher netX
- Integrated CPU?
  - Hilscher netX
  - FPGA solutions (optional: softcore)
- Need to disclose quantities?
  - FPGA solutions (buy out licenses available)
- Multi Protocol Support
  - Hilscher netX
  - FPGA solutions

Physical Layer, Network Interface?

EtherCAT Physical Layer is 100BASE-TX or –FX*

EtherCAT PHYs have to support

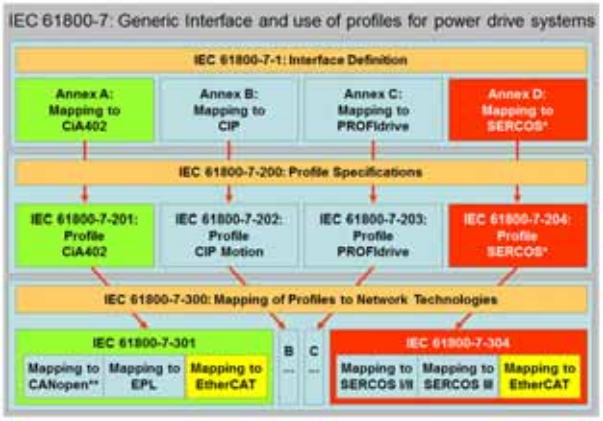
- Full Duplex Communication
- Auto-Negotiation, MDI/MDI-X auto-crossover
- MII with MII management interface
- PHY link loss reaction time (link loss to link signal/LED output change) shorter than 15µs (for short redundancy switchover)

For further details see the ESC Datasheets or the corresponding PHY Selection Guide

* + LVDS for modular devices, supported by Beckhoff ASICs only

Device Profile?

- Which device profile shall be supported?
- Drive: both CiA402 (the CANopen drive profile, IEC 61800-7-201) and the Sercos-Drive-Profile (IEC 61800-7-204) are mapped on EtherCAT



- If the device can be described as hardware modules or as logical modules:
  - Modular Device Description recommended
  - Modular Device Profile (ETG.5001)

Parameter + Process Data?

Device Profile determines Parameters and Process Data setup

But: decision if the Process Data Layout shall be:

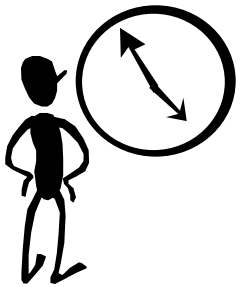
→ **Fixed**: cannot be changed by user.
Example: simple I/O device.

→ **Selectable**: user can select between several predefined process data layouts.
Example: drive where process data layout depends on the selected drive operation mode

→ **Determined by module combination (Dynamic)**: determined at device bootup by actual hardware modules;
Example: bus coupler with modular I/O.

Synchronization and Time Stamping?

What level of Synchronization is required?

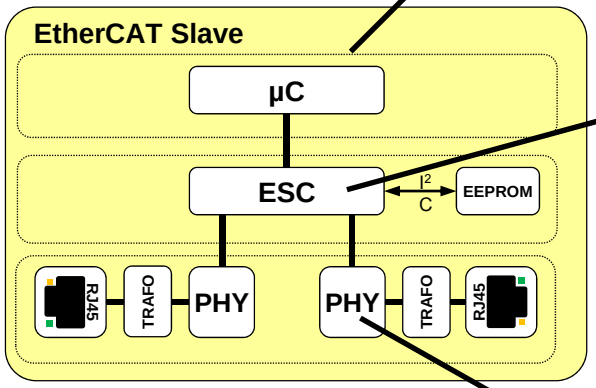


1. Freerun:
   local timer controls application, no synchronization with network

2. Synchronized with network cycle:
   local application triggered by reception of process data ("SM-event"). Jitter mainly depends on master accuracy.

3. Synchronized by Distributed Clocks:
   local application triggered by high precision and fully synchronized hardware interrupt generated by local clock; accuracy in the order of nanoseconds

Hardware Design



Application / Host Controller
According to the Application Requirements

EtherCAT Slave Controller
Acording to the ESC Selection Criteria

Network Interface / Physical Layer
Standard Ethernet Interface, Requirements according to PHY Selection Guide / ESC Data Sheet

Software Development

Typical Software-Structure:

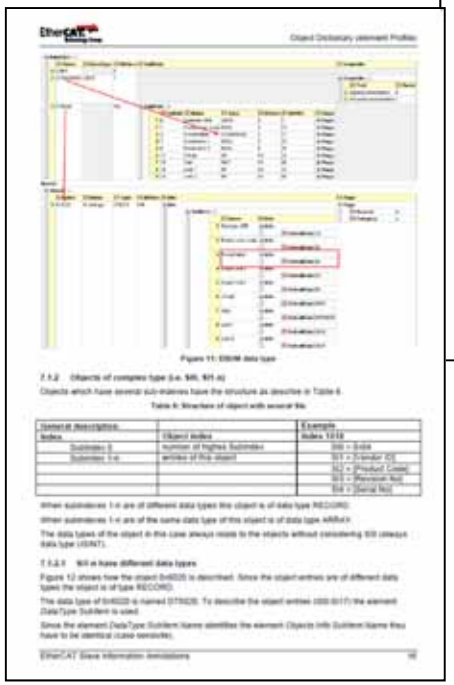
- Applications-Program/Firmware
- Communication-Stack with the following elements:
  - EtherCAT State Machine
    - Verification of the configuration settings done by master
    - Handling of synchronization + configuration errors
  - Mailbox-Protocol Handling
    - Most common protocol: CoE
    - Error Handling (e.g. Parameter cannot be read or written)
  - Access to ESC memory (DPRAM)
  - Synchronization

The listed functionality is supported by most available stacks, such as

- Beckhoff Slave Sample Code
- Hilscher EtherCAT Slave Stack)

Device Description: ESI File

- Each EtherCAT Slave device is described by an „EtherCAT Slave Information“ (ESI) File in XML Format
- The ESI Format is defined in the ETG.2000 spec
- Of course there are also a schemas, example files etc. on the EtherCAT website
- The ESI also supports the description of modular devices



Conformance Testing

- The EtherCAT Conformance Test Tool (CTT) is helpful throughout the implementation – and afterwards
- Having the CTT and testing with it is a requirement
- Recommended Procedure:
  - If not yet ETG member: Join ETG (free of charge)
  - Obtain EtherCAT Vendor ID (free of charge)
  - Subscribe to Conformance Test Tool
  - Conformance Test Record (ETG.7000-2) is Test Guideline
    - Test with CTT
    - Test of the LED behavior (ETG.1300)
    - Marking and Trademark Hints (ETG.9001)
    - Further tests
- Test in official EtherCAT Test Center (and Certification) is optional, but recommended

Common Issues – and how to avoid them!

The 5 „Killer“ for passing the conformance test:

1. Logo:
   Neither on the device nor in the documentation the EtherCAT logo is shown
2. Trademark:
   Trademark hint is missing in the documentation
3. Indicator and Port Marking:
   Marking is missing or misleading
4. Watchdog Behavior:
   If sending of process data is stopped the device does not show the required behavior
5. DC-Signal Monitoring:
   If the interrupt for the synchronization is disabled the device does not show the required behavior