EtherCAT Diagnostic Specification

FUB “ECD”

Customer: Vistaprint Winterthur

Project: Line ALADDIN

Projectnr.: 10116.04.01

Version: 0.01

Storage: D:\01\_Projekte\10116 Vista Print\04 Line - ALADDIN\01 Development\20\_Engineering\20\_Development\Specification\03 ECD Client\ECP client Spec V0.01docx.docx

# Table of contents

[1 Table of contents 2](#_Toc380483001)

[2 Introduction 3](#_Toc380483002)

[2.1 Scope 3](#_Toc380483003)

[2.2 Reference Documentation 3](#_Toc380483004)

[2.3 Version 3](#_Toc380483005)

[2.4 Abbreviations, definitions, glossary 3](#_Toc380483006)

[3 Common information’s 4](#_Toc380483007)

[3.1 Used software components and libraries 4](#_Toc380483008)

[3.2 Used ADS Index-Group 4](#_Toc380483009)

[3.3 ADS error codes 4](#_Toc380483010)

[3.4 Concept (Example configuration) 5](#_Toc380483011)

[4 Controller Description 6](#_Toc380483012)

[4.1 Level 0 Element: ADS client (Class ACL) 6](#_Toc380483013)

# Introduction

## Scope

This document gives an overview how to use the FUB “ECD” (EtherCAT Diagnostic) and is solely for the benefit of Vistaprint and all the persons that are involved at platform development. These are software developers, quality engineers and maintenance engineers.

## Reference Documentation

|  |  |  |
| --- | --- | --- |
| Documents | Version | Datum |
| Beckhoff InfoSys - [EtherCAT Diagnose](http://infosys.beckhoff.com/index.php?content=../content/1031/tcplclib_tc2_ethercat/html/tcplclibtcethercat_fb_ecgetallslaveaddr.htm&id=12612)  Infosys.beckhoff.com TwinCAT3 / TExxxx | TC3 Engineering / PLC / Libraries / TwinCAT 3 PLC Lib: Tc2\_EtherCAT |  |  |
| Beckhoff InfoSys - [AdsErrorCode](http://infosys.beckhoff.de/english.php?content=../content/1033/tc3_adsnetref/html/TwinCAT.Ads.AdsErrorCode.html&id=) |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Author | State | Version | Date |
| Start | AVME/PRE | d | 0.1 | 2014-04-25 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

State: **d** = design, **r** = released

## Abbreviations, definitions, glossary

|  |  |
| --- | --- |
| Designation | Name |
| ADS | "Automation Device Specification” Describes a device and fieldbus-independent interface which controls the communication between ADS device(s). |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Common information’s

## Used software components and libraries

|  |  |  |
| --- | --- | --- |
| Description | Version | Datum |
| TwinCAT system library (Tc2\_System) | V3.3.4.0 | 2013-09-02 |
| TwinCAT utilities library (Tc2\_Utilities) | V3.3.5.0 | 2013-07-18 |
| TwinCAT EtherCAT library (Tc2\_EtherCAT) | V3.3.2.0 |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Used EtherCAT Functions

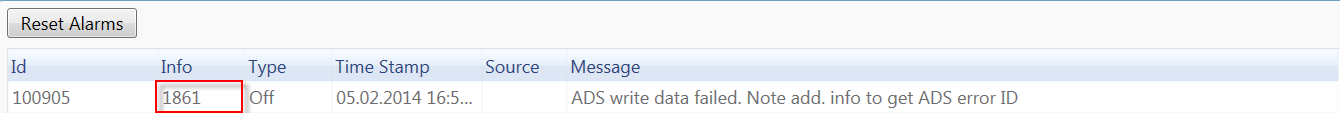
|  |  |
| --- | --- |
| Function | Description |
| FB\_EcGetConfSlaves | generates a List of all configured Slaves from the Master object directory |
| FB\_EcGetAllSlaveAddr | allows the addresses of all the slaves connected to the master to be read |
| FB\_EcGetMasterState | allows the EtherCAT state of the master to be read |
| FB\_EcGetSlaveState | allows the EtherCAT status and the Link status of an individual EtherCAT slave to be read |
| FB\_EcGetSlaveCrcError | allows the CRC error counters of the individual ports (A, B and C) of a slave to be read |
| FB\_EcMasterFrameStatisticClearCRC | Reset all CRC Errors on the slaves |
|  |  |
|  |  |
|  |  |

## ADS error codes

Follow the link for more details about ADS error codes: [AdsErrorCode](http://infosys.beckhoff.de/english.php?content=../content/1033/tc3_adsnetref/html/TwinCAT.Ads.AdsErrorCode.html&id=)

The error code is an additional information of alarms set by ECD controller.

Example: 1861 (ClientSyncTimeOut)



## Concept (Example configuration)

Once the element is startet with the command RUN, it cyclicle scans the EtherCAT Slaves and check all States and CRC errors. If something is not correct, a message would be generated.

Every EtherCAT Master needs an Element of ECT to check its slaves. At the configuration the NetId has to be given (for example: ‘192.168.70.24.2.1’). This NetId could be get in the configuration.

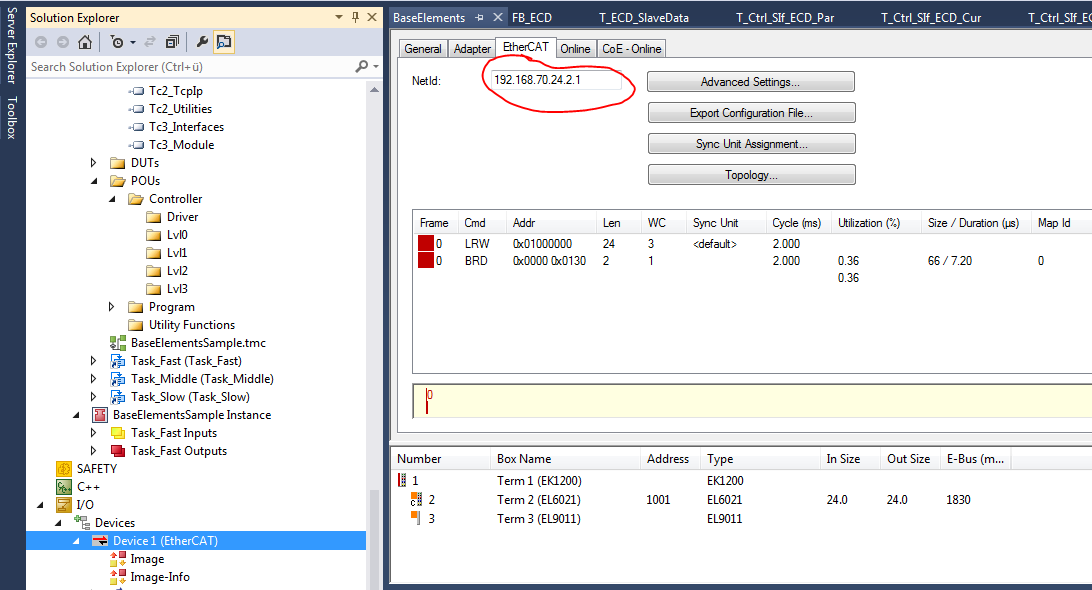


Figure 1: Get EtherCAT NetID

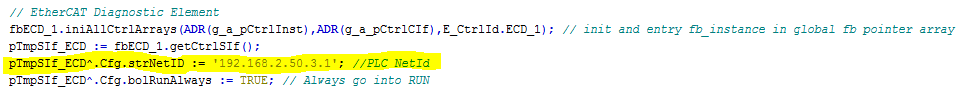


Figure 2: Configuration of NetID

The amount of slaves is now limited by 256. This is defined in the variable ECD\_Const.cuinMaxSlaves.

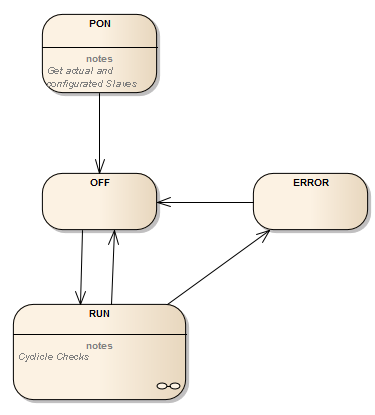
# Controller Description

## Level 0 Element: ECD client (Class ECD)

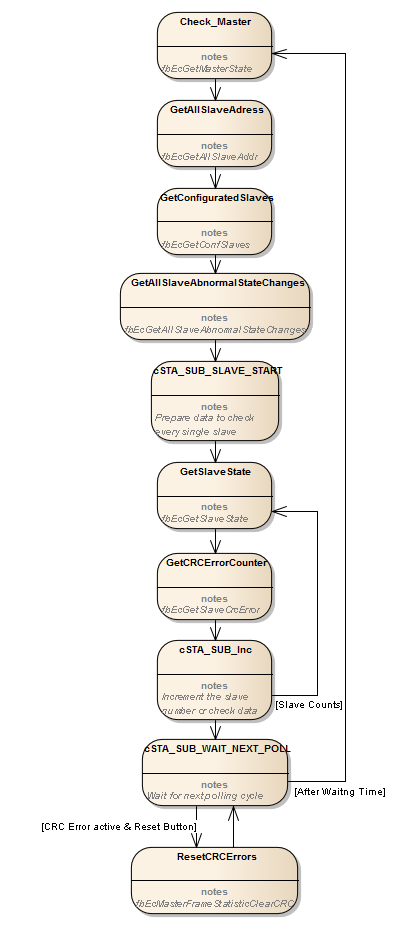
**Functionality**

Once the element is startet with the command RUN, it cyclicle scans the EtherCAT Slaves and check all States and CRC errors. If something is not correct, a message would be generated.

**Main Sequence**



**RUN Cequence, cyclicle checks**



**Commands**

* PON (Determines the handle IDs of all connections from the internal list)
* OFF (No slave polling)
* RUN (Checks enabled)

**Configuration**

|  |  |  |
| --- | --- | --- |
| *Variable* | *Datatype* | *Description* |
| strNetID | STRING(30) | NetID |
| bolRunAlways | BOOL | Run allways the scanning without set into run |
|  |  |  |

**Parameter**

|  |  |  |
| --- | --- | --- |
| *Variable* | *Datatype* | *Description* |
| udiTOPowerOn\_ms | UDINT | [ms] Timeout power on. Timeout for initialize all ADS connections. |
| udiTOCheckAll\_ms | UDINT | [ms] Timeout function over all slaves |
| udiTOCheckOne\_ms | UDINT | [ms] Timeout function for one slave |
| udiPollingInterval\_ms | UDINT | [ms] Polling interval for States |
| udiMaxCRC | UDINT | Maximum CRC Value one slave |
|  |  |  |

**Current Values**

|  |  |  |
| --- | --- | --- |
| *Variable* | *Datatype* | *Description* |
| uinNumOfSlaves | UINT | Shows the number of EtherCAT Slaves |
| stSlaveData | ARRAY[0..ECD\_Const.cuinMaxSlaves] OF T\_ECD\_SlaveData | Information of every slave |
| worMasterState | WORD | State Of Master |
| udiCRCSumm | UDINT | All CRC Errors |
| uinNumOfSlaveErrors | UINT | Counter of Slave Errors State |

**Slave data T\_ECD\_SlaveData**

|  |  |  |
| --- | --- | --- |
| *Variable* | *Datatype* | *Description* |
| stConfigData | Tc2\_EtherCAT.ST\_EcSlaveConfigData | Slave Configuration |
| uinAddress | UINT | Address of the slave (as UINT, beacuase in stConfigData is only as WORD) |
| stState | Tc2\_EtherCAT.ST\_EcSlaveState | State of the slave |
| stCRCError | Tc2\_EtherCAT.ST\_EcCrcErrorEx | CRC Error totally |

**Errors**

The Errortexts would be written in the FBinit, but complemented in the Software to get more informations into the texts.

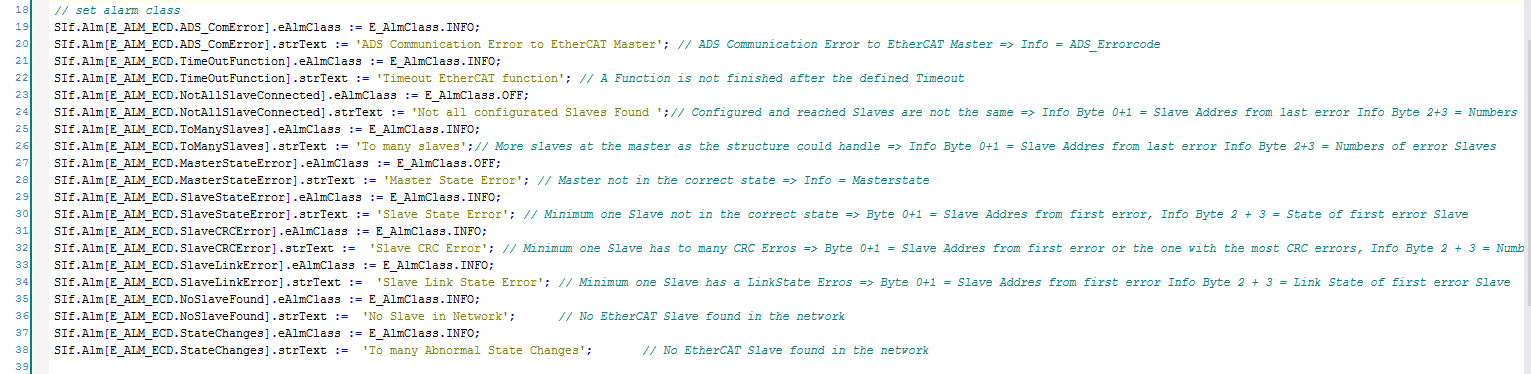


Figure 3: Definition in FB\_init

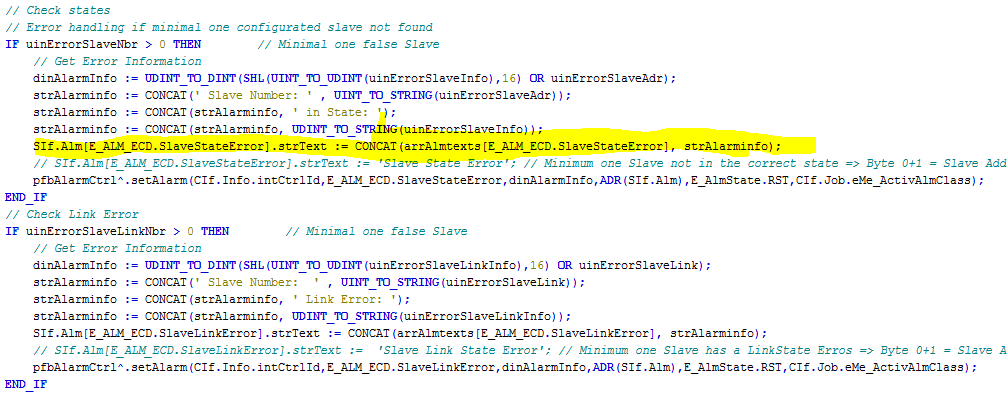


Figure 4: Adding informations to Errortexts

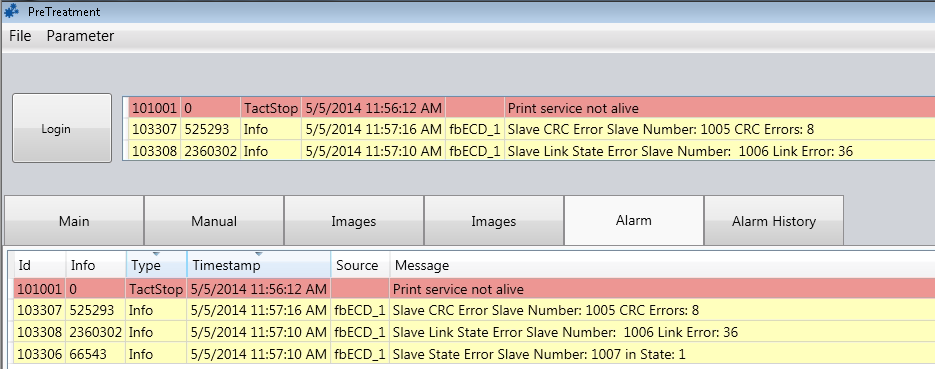


Figure 5: Display of errortexts

* ADS Communication Error to EtherCAT Master category: INFO  
  Info = ADS\_Errorcode
* Timeout EtherCAT function category: INFO
* Not all configurated Slaves Found category: OFF  
  Info Byte 0+1 = Slave Addres from last error Info Byte 2+3 = Numbers of error Slaves
* To many slaves category: INFO  
  Info Byte 0+1 = Slave Addres from last error Info Byte 2+3 = Numbers of error Slaves
* Master State Error category: OFF  
  Info = Masterstate
* Slave State Error category: INFO  
  Byte 0+1 = Slave Addres from first error, Info Byte 2 + 3 = State of first error Slave
* Slave CRC Error category: INFO  
  Byte 0+1 = Slave Addres from first error or the one with the most CRC errors, Info Byte 2 + 3 = Number of CRC Errors from first Slave with error
* Slave Link State Error category: INFO  
  Byte 0+1 = Slave Addres from first error Info Byte 2 + 3 = Link State of first error Slave
* No Slave in Network category: INFO
* To many Abnormal State Changes category: INFO

# Appendix

**TYPE ST\_EcSlaveState**

The structure ST\_EcSlaveState contains the EtherCAT status and the Link status of an EtherCAT slave device.

TYPE ST\_EcSlaveState:

STRUCT

deviceState :BYTE;

linkState :BYTE;

END\_STRUCT

END\_TYPE

**deviceState:** EtherCAT status of a slave. The status can adopt one of the following values:

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| EC\_DEVICE\_STATE\_INIT | 0x01 | Init State |
| EC\_DEVICE\_STATE\_PREOP | 0x02 | Pre-Operational State |
| EC\_DEVICE\_STATE\_BOOTSTRAP | 0x03 | Bootstrap State |
| EC\_DEVICE\_STATE\_SAFEOP | 0x04 | Safe-Operational State |
| EC\_DEVICE\_STATE\_OP | 0x08 | Operational State |

Additionally following bits can be set:

|  |  |  |
| --- | --- | --- |
| **Konstante** | **Value** | **Description** |
| EC\_DEVICE\_STATE\_ERROR | 0x10 | Statemachine error in the EtherCAT slave |
| EC\_DEVICE\_STATE\_INVALID\_VPRS | 0x20 | Invalid VendorId, Product Code, RevisionsNo or SerialNo |
| EC\_DEVICE\_STATE\_INITCMD\_ERROR | 0x40 | Error occured while sending initialization commands. |

**linkState:** Link status of an EtherCAT slave. The Link status can consist of an ORing of the following bits.

|  |  |  |
| --- | --- | --- |
| **Konstante** | **Value** | **Description** |
| EC\_LINK\_STATE\_OK | 0x00 | Link status ok |
| EC\_LINK\_STATE\_NOT\_PRESENT | 0x01 | No communication with the EtherCAT-Slave |
| EC\_LINK\_STATE\_LINK\_WITHOUT\_COMM | 0x02 | Error at port X(defined by EC\_LINK\_STATE\_PORT\_A/B/C/D). The port has a link, but no communication is possible. |
| EC\_LINK\_STATE\_MISSING\_LINK | 0x04 | Missing link at port X(defined by EC\_LINK\_STATE\_PORT\_A/B/C/D). |
| EC\_LINK\_STATE\_ADDITIONAL\_LINK | 0x08 | Additional link at port X(defined by EC\_LINK\_STATE\_PORT\_A/B/C/D). |
| EC\_LINK\_STATE\_PORT\_A | 0x10 | Port 0 |
| EC\_LINK\_STATE\_PORT\_B | 0x20 | Port 1 |
| EC\_LINK\_STATE\_PORT\_C | 0x40 | Port 2 |
| EC\_LINK\_STATE\_PORT\_D | 0x80 | Port 3 |

**Master States**

|  |  |  |
| --- | --- | --- |
| **Constant** | **Value** | **Description** |
| EC\_DEVICE\_STATE\_INIT | 0x01 | Master is in the state Init |
| EC\_DEVICE\_STATE\_PREOP | 0x02 | Master is in the state Pre-Opereational |
| EC\_DEVICE\_STATE\_SAFEOP | 0x04 | Master is in the state Safe-Operational |
| EC\_DEVICE\_STATE\_OP | 0x08 | Master is in the state Operational |