cynapse®
Beckhoff PLC – ifm IO-Link master

Getting started

WITTENSTEIN alpha GmbH

Walter-Wittenstein-Straße 1
D-97999 Igersheim
Germany

Cybertronic support

If you have questions about this implementation example, please contact:
cybertronic-support@wittenstein.de

Customer service

		✉	✆
Deutschland	WITTENSTEIN alpha GmbH	service@wittenstein-alpha.de	+49 7931 493-12900
Benelux	WITTENSTEIN BVBA	service@wittenstein.biz	+32 9 326 73 80
Brasil	WITTENSTEIN do Brasil	vendas@wittenstein.com.br	+55 15 3411 6454
中国	威腾斯坦（杭州）实业有限公司	service@wittenstein.cn	+86 571 8869 5856
Österreich	WITTENSTEIN GmbH	office@wittenstein.at	+43 2256 65632-0
Danmark	WITTENSTEIN AB	info@wittenstein.dk	+45 4027 4151
France	WITTENSTEIN sarl	info@wittenstein.fr	+33 134 17 90 95
Great Britain	WITTENSTEIN Ltd.	sales.uk@wittenstein.co.uk	+44 1782 286 427
Italia	WITTENSTEIN S.P.A.	info@wittenstein.it	+39 02 241357-1
日本	ヴィッテンシュタイン株式会社	sales@wittenstein.jp	+81-3-6680-2835
North America	WITTENSTEIN holding Corp.	technicalsupport@wittenstein-us.com	+1 630-540-5300
España	WITTENSTEIN S.L.U.	info@wittenstein.es	+34 93 479 1305
Sverige	WITTENSTEIN AB	info@wittenstein.se	+46 40-26 50 10
Schweiz	WITTENSTEIN AG Schweiz	sales@wittenstein.ch	+41 81 300 10 30
台湾	威騰斯坦有限公司	info@wittenstein.tw	+886 3 287 0191
Türkiye	WITTENSTEIN Güç Aktarma Sistemleri Tic. Ltd. Şti.	info@wittenstein.com.tr	+90 216 709 21 23

Table of contents

1 About this manual

This guide contains procedures for the exemplary use of the WITTENSTEIN sensor cynapse®.

This guide uses example code. If you require any code examples, please contact: cybertronic-support@wittenstein.de

The original was prepared in German, all other language versions are translations of the original instructions.

1.1 Information symbols and cross references

The following information symbols are used:

- Indicates an action to be performed
- ⮐ Indicates the results of an action
- ⓘ Provides additional handling information

A cross reference refers to the chapter number and the header of the target section (e. g. chapter 2 "Hardware structure").

A cross reference to a table refers to the table number (e.g. table "Tbl - 1").

2 Hardware structure

The following hardware components were used for the sample project:

Control:	Beckhoff C6930
IO-LINK master:	ifm AL1330
IO-LINK device:	WITTENSTEIN cynapse®

Figure 1 shows the schematic representation of the structure. The IO-Link master AL1330 is connected to the PLC unit C6930 by means of EtherCAT (green). WITTENSTEIN cynapse® is connected to port 1 of the IO-Link master (black).

Figure 1: Schematic diagram

3 Commissioning in TwinCAT3

To commission cynapse® you need a new TwinCAT project. In addition, the following are required:

➲ A network port on the controller is configured as an EtherCAT port
➲ TwinCAT development environment is installed
➲ The hardware structure has been carried out according to Chapter 2

3.1 Configuration

Open the TwinCAT development environment (Visual Studio) and create a new TwinCAT project via "File" → "New" → "Project...". The new project is visible in the project folder explorer. It is possible to use the TwinCAT "locally" on the C6390 or "remotely" from an engineering PC. In this implementation example, TwinCAT is used "locally".
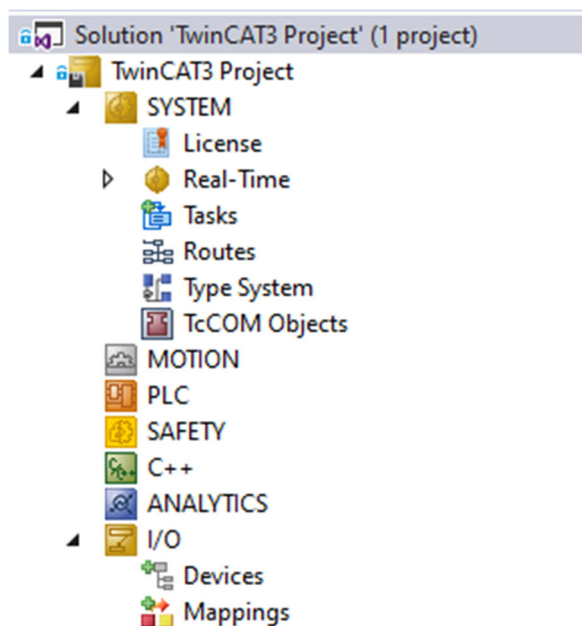
Figure 2: Project folder explorer

3.2 EtherCAT Slave Information – Install File

ifm provides an ESI file for imaging the AL1330 in TwinCAT at www.ifm.com.
The ESI file defines all parameter and process data and their valid value ranges.

To integrate the ESI file into the EtherCAT configuration software TwinCAT 3.1:

- Download the device's ESI file.
- Copy the downloaded file to the following subdirectory of the TwinCAT installation directory: ..\3.1\Config\Io\EtherCAT
- Start TwinCAT

➲ TwinCAT loads the device description into the device catalog.

3.3 Inserting connected devices

In order to insert the connected devices, TwinCAT must be placed in "Config mode", if it is not already in it. To do this, click on the symbol ▨ or select "Restart TwinCAT (Config Mode)" from the "TWINCAT" menu →. Then, in the project folder explorer, you can select the "Devices" located within the "I/O" item and either right-click to open a context menu and select "Scan" or click on the icon ✳ in the menu bar to start the action.
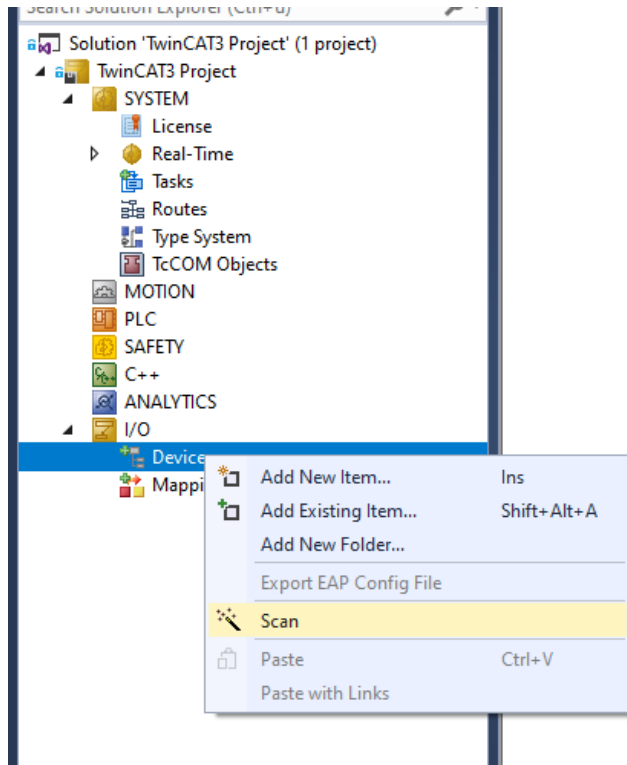

Figure 3: Select "Scan"

The following message must be confirmed and the "EtherCAT" devices must be selected in the dialog shown in Figure 4
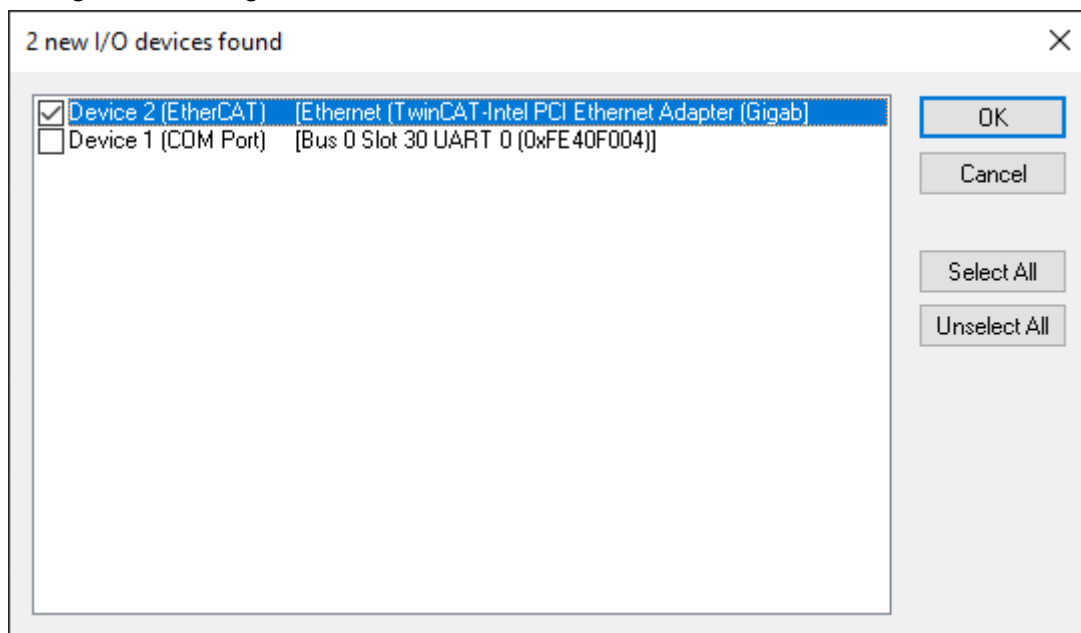

Figure 4: Selection of I/O devices to be integrated

The following message "Search for new boxes" must also be confirmed in order to search for the terminals connected to the devices.

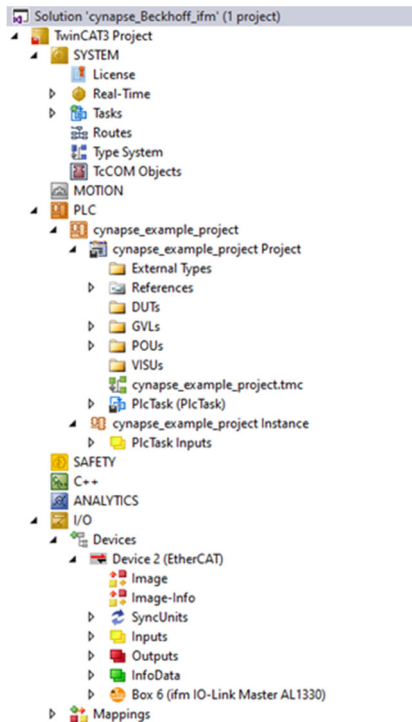Based on the configuration described here, the result is as follows:

Figure 5: Configuring the TwinCAT 3 environment

⮑ There is now an EtherCAT device in the "Devices" item (here: Device 2 (EtherCAT)).
The IO-Link master AL1330 is attached to this.

3.4 Configuration of the IO-Link master AL1330

When creating the IO-Link master, an additional tab is created called "Slots". Here are all available IO-Link ports of the IO-Link master. By default, the ports are assigned a process data image with four inputs and outputs each. WITTENSTEIN cynapse® sends process data with a length of 16 bytes and does not receive any process data.

In order to store this process data profile on the port, the currently stored process data profile must first be removed. To do this, select the desired port with one click, so that it is highlighted in blue, and then click on the symbol [X]
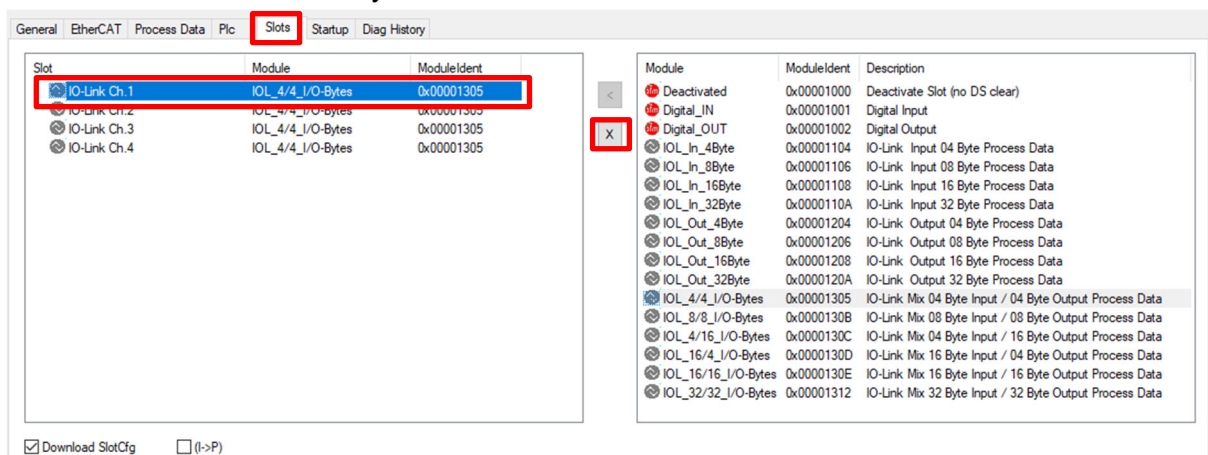
Figure 6: "IO-Link" tab

Then in the right table the appropriate process data image "IOL_In_16Byte" can be selected and assigned to the selected port by clicking on [<] .
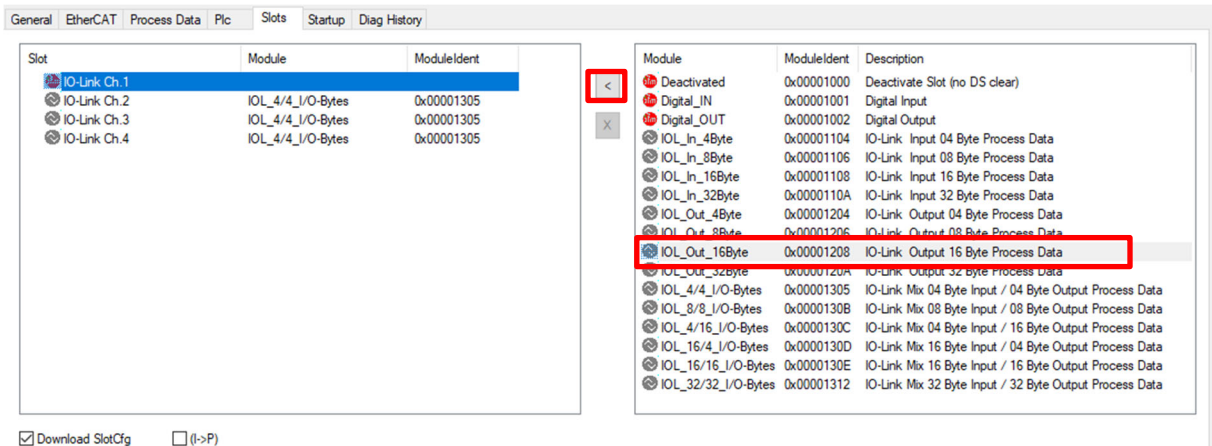
Figure 7: Process data image selection

An IO Device Description file cannot be stored. A direct interpretation of the process data in the IO-Link master is thus not possible and must be carried out in the following e.g. in the PLC program.

3.5   Creating a PLC program

To create a programming environment, a new PLC subproject must be added from the context menu of "PLC" in the project folder explorer by selecting "Add New Item...":

Figure 8: Creating a PLC program

In the subsequent dialog, a "Standard PLC Project" is selected and a project name (e.g. "cynapse®_example_project") is assigned. Selecting "Standard PLC Project" automatically creates the "Main" program. This can be opened by double-clicking on the PLC subproject "cynapse®_example_project" in "POUs". A global variable list (GVL) is also created. Here you have to create sample variables (e.g. as an array with a length of 16 bytes) for the continued procedure, which can then be linked with the input variables of cynapse®:

Figure 9: Creation of global variables

After the PLC project has been compiled via "Build", the variables marked "AT%" are present in the "Assignments" and can be linked to the EtherCAT input variables.
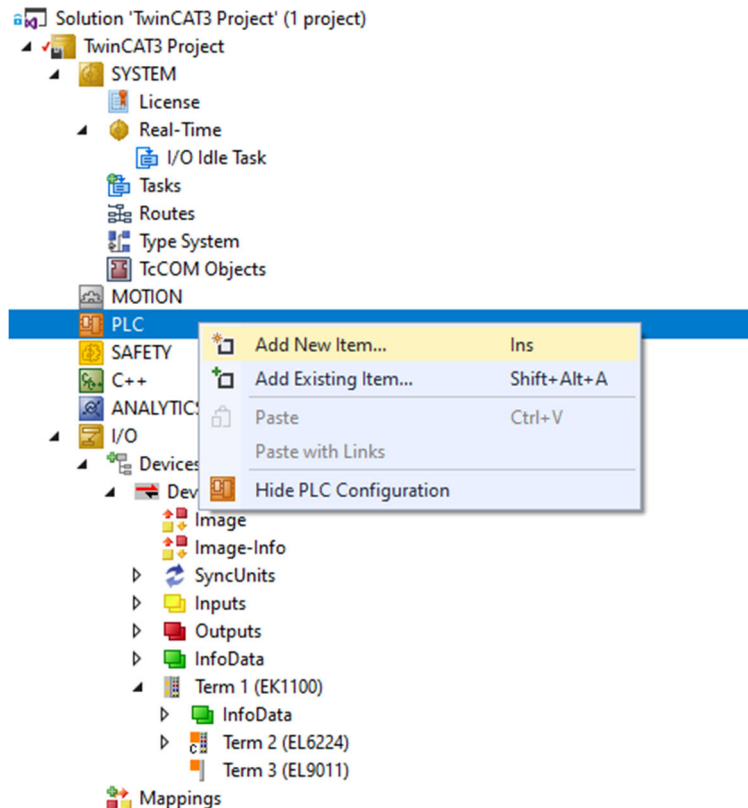
Through right-clicking on the respective variable, a window for selecting its link is opened via "Change link...".
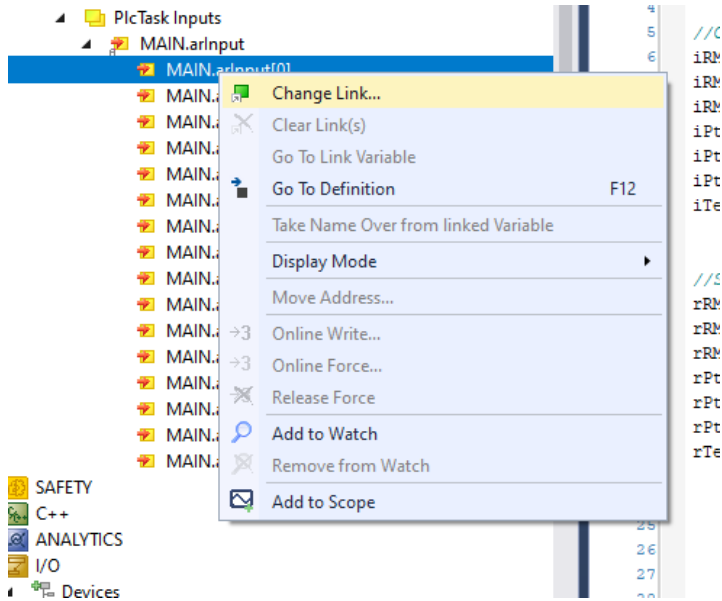

Figure 10: Create links between PLC variable and process objects

3.6 Activating the configuration

By activating the configuration via "TWINCAT" → "Activate configuration" or by clicking on the symbol ![symbol], the TwinCAT control is set to run mode and the process data from cynapse® is interrogated cyclically.

4 Access to IO-Link data

The Beckhoff IO-Link master terminal AL1330 is divided into two services. On the one hand, it provides an IO-Link master for the connected sensors, on the other hand, it is an EtherCAT slave of the higher-level TwinCAT master.

In principle, cyclic and acyclic data are exchanged between the IO-Link master and the IO-Link slave. From the perspective of the EtherCAT master, the cyclic process data can be accessed via the PDOs. The access to the acyclic data (blob, parameters and events) takes place via AoE.

4.1 Reading process data

After successful configuration of the respective IO-Link port, the process data is displayed in the "Process data" tab after a restart of the EtherCAT system. The process data are interrogated cyclically in the set system clock and can be linked to PLC variables for further use.

Figure 11: "Process data" tab

WITTENSTEIN cynapse® transmits the current temperature, as well as various acceleration parameters. Various process data formats are provided in order to offer different characteristic data with a constant process data length of 16 bytes. The process data format is configured as described in Chapter 4.2.

cynapse® does not use any outgoing process data (from the point of view of the IO-Link master).

For more information on the available process data formats, please refer to the cynapse® operating manual.

In order to convert the raw process data into measured values, they must be scaled in the PLC program. For this purpose, all incoming process data, as described in Chapter 3.5, must be linked to PLC variables in order to process them further in the PLC.

Depending on the selected process data profile, cynapse® supplies different measurement data. The various process data profiles can be found in the cynapse® operating manual.
The scaling of the process data for the process data profile "RMS, Peak to Peak, Temperature" (see table "Tbl - 1") is described below by way of example.

Byte	Description	Unit	Conversion factor
0	reserved	-	-
1	Process data profil	-	-
2 … 3	RMS value radial	m/s^2	0,01
4 … 5	RMS value axial	m/s^2	0,01
6 … 7	RMS value tangential	m/s^2	0,01
8 … 9	Peak to Peak radial	m/s^2	0,01
10 … 11	Peak to Peak axial	m/s^2	0,01
12 … 13	Peak to Peak tangential	m/s^2	0,01
14… 15	Temperature	°C	0,01

Tbl - 1: Process data profile 1 "RMS, Peak to Peak, Temperature"

The measured values coming from cynapse® are of the type Integer and thus have a length of 2 bytes. Consequently, 2 bytes of the raw process data must always be linked to a measured value. The following example shows the scaling of the process data in a PLC program. In order to convert the measured values into a decimal number, the conversion factor specified in table "Tbl - 1" can be used.

Declaration part:

```
PROGRAM MAIN
VAR
    arInput AT%I*  : ARRAY[0..15] OF SINT;

    iRMS_X         : INT;
    iRMS_Y         : INT;
    iRMS_Z         : INT;
    iPtP_X         : INT;
    iPtP_Y         : INT;
    iPtP_Z         : INT;
    iTemp          : INT;

    rRMS_X         : REAL;
    rRMS_Y         : REAL;
    rRMS_Z         : REAL;
    rPtP_X         : REAL;
    rPtP_Y         : REAL;
    rPtP_Z         : REAL;
    rTemp          : REAL;

END_VAR
```

Implementation:

```
//***********************************************
// ProcessData Read
//***********************************************

//Connect Processdata

iRMS_Z := arInput[2] * 256 + arInput[3];
iRMS_Y := arInput[4] * 256 + arInput[5];
iRMS_X := arInput[6] * 256 + arInput[7];
iPtP_Z := arInput[8] * 256 + arInput[9];
iPtP_Y := arInput[10] * 256 + arInput[11];
iPtP_X := arInput[12] * 256 + arInput[13];
iTemp := arInput[14] * 256 + arInput[15];

//Scale to Decimal
rRMS_X := iRMS_X * 0.01;
rRMS_Y := iRMS_Y * 0.01;
rRMS_Z := iRMS_Z * 0.01;
rPtP_X := rPtP_X * 0.01;
rPtP_Y := rPtP_Y * 0.01;
rPtP_Z := rPtP_Z * 0.01;
rTemp := iTemp * 0.01;
```

4.2 Configure the process data format

As described in Chapter 4.1, cynapse® supports different process data profiles with a constant process data length. The process data profile is selected using the parameter "Settings" Index 0x60 Subindex 0x09 ("Process data profile"). As described in Chapter 4.3, the parameter "Settings" can be adjusted by means of the function module "Write parameters" in the PLC program or via the tab "AoE – Online" in the TwinCAT development environment.

The detailed description of the process data formats can be found in the "cynapse®" operating manual. Table "Tbl - 2:" shows an overview of the available process data profiles.

Process data Profile		
Data type	UIntegerT8	
Authorization	rw	
Profile	RMS, Peak to Peak, Temperature	1
	Acceleration, Peak to Peak, Temperature	2
	RMS, Acceleration, Temperature	3

Tbl - 2: Process data profile

ⓘ When writing and reading parameters via the tab "AoE – Online", it should be noted that the port numbering of Beckhoff and ifm are different!

4.3 Writing and reading parameters

cynapse® supports parameterization by ISDU (Indexed Service Data Unit). These acyclic parameters must be explicitly requested or sent via the PLC. Access is via ADS or CoE. The following describes how to read and write parameters using ADS.

An ADS address always consists of NetID and port number. An ADS command is sent from TwinCAT via AoE (ADS over EtherCAT) to the IO-Link master AL1330 and from there to the demand data channel.

AoE NetID

The AL1330 has its own NetID for communication. This can be looked up in the terminal configuration in the "EtherCAT" tab under "Advanced settings" -> "Mailbox" -> "AoE".
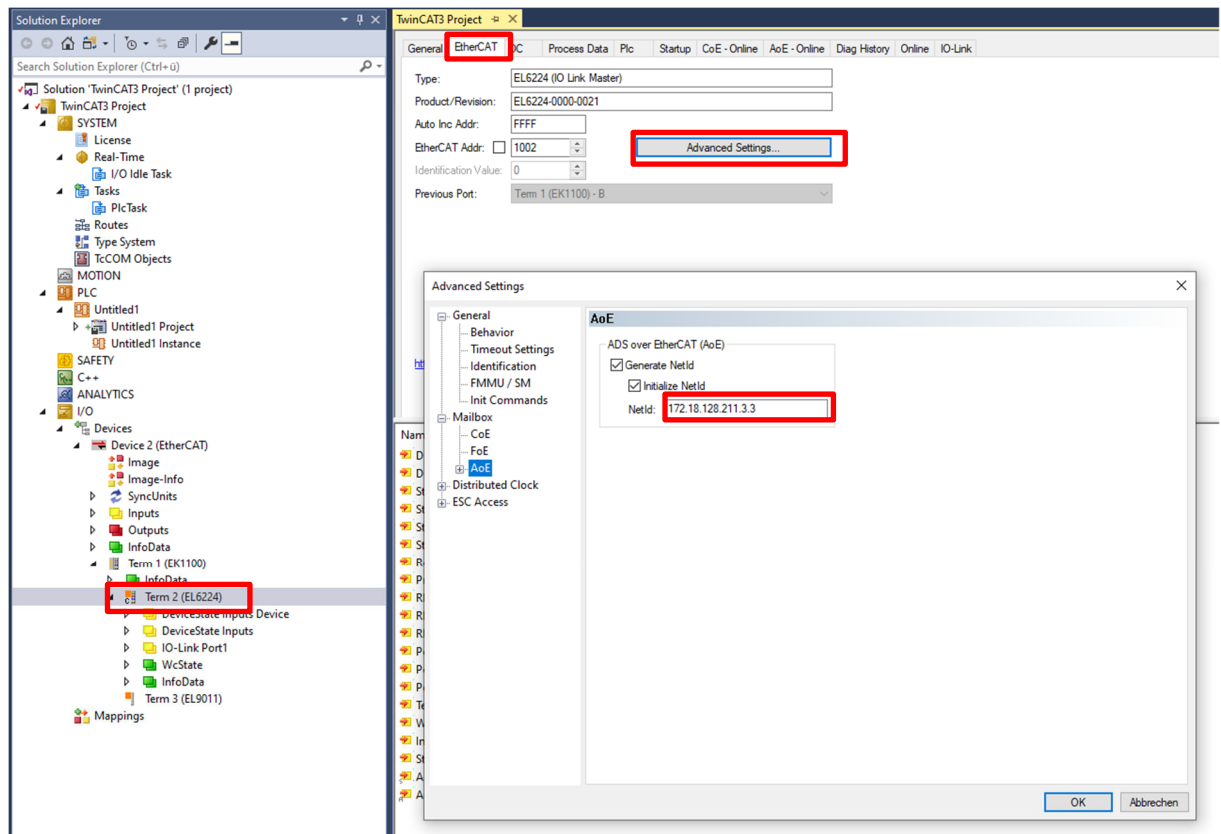
Figure 12: ADS address

Port number

The assignment of the query to a single IO-Link port takes place via the port number.
The port number is assigned ascending from 0x1001. IO-Link port 1 thus corresponds to 0x1001 and IO-Link port n corresponds to port number 0x1001+n-1.

The following definition therefore applies to the IO-Link master AL1330 with 4 IO-Link ports used here

IO-Link port 1 → port number 0x1001

IO-Link port 2 → port number 0x1002

IO-Link port 3 → port number 0x1003

IO-Link port 4 → port number 0x1004

ADS index group

The index group of an ADS command is set to 0xF302 for the IO-Link demand data channel.

ADS index offset

In index offset, the IO-Link addressing is encoded with index and subindex.
The index offset is 4-byte in size and is divided as follows:

- Bit 16-31:   Index
- Bit 8-15:   reserved
- Bit 0-7:   Subindex

Example: Index 0x005D and subindex 0x02 equals index offset 0x005D0002

The available acyclic parameters can be read in the cynapse® operating manual.

4.4   Sample project: Reading/writing parameters

The following describes how parameters can be read or written via AoE using the function modules "ADSRead" and "ADSWrite".

The two modules "ADSRead" and "ADSWrite" are part of the Beckhoff own library "TC2_Standard". By default, the library is loaded when a PLC project is rebuilt.

The following example demonstrates how to write and read the "Operating temperature threshold" parameter using the "ADSREAD" and "ADSWRITE" function blocks. First, a new threshold value is written with the instance "fbIOLWrite" of the "ADSWrite" module and then read for verification with the instance "fbIOLRead".

ⓘ   The complete sample project is available on request at: cybertronic-support@wittenstein.de

Declaration part:

```
PROGRAM MAIN
VAR
    fbIOLRead       : ADSRead;
    fbIOLWrite      : ADSWrite;

    iState          : INT;
    bExecute        : BOOL;

    rTemperatureThresholdWrite      : REAL := 40; //°C
    rTemperatureThresholdRead       : REAL;

    bBusy           : BOOL;
    bError          : BOOL;
    nErrID          : UDINT;
END_VAR
```

Implementation:

```
CASE iState OF
0 : IF bExecute THEN
        bBusy := TRUE;
        bError := FALSE;
        nErrID := 0;

        // Write new Process Data Profile to cynapse®

        fbIOLWrite(
          NETID:='172.18.128.211.3.3',    //AoE-NetID AL1330
          PORT:= 16#1000,                 //PortNo IO-Link Port
          IDXGRP:= 16#F302,               //Defined by Beckhoff
          IDXOFFS:= 16#00520000,          //Index = 0x0052 and Subindex = 0x00
          LEN:= SIZEOF(rTemperatureThresholdWrite),
          SRCADDR:= ADR(rTemperatureThresholdWrite),
          WRITE := TRUE );

        iState := 1;
    END_IF


1 : fbIOLWrite( WRITE := FALSE, BUSY=>bBusy, ERR=>bError, ERRID=>nErrID );

    IF NOT bBusy THEN
      IF NOT bError THEN
         iState := 2; //Success
      ELSE
         iState := 100; //Error
      END_IF
    END_IF

2 : //Read Process Data Profile
    fbIOLRead(
      NETID:='172.18.128.211.3.3',    //AoE-NetID AL1330
      PORT:= 16#1000,                 //PortNo IO-Link Port
      IDXGRP:= 16#F302,               //Defined by Beckhoff
      IDXOFFS:= 16#00520000,          //Index = 0x0052 and Subindex = 0x00
      LEN:= SIZEOF(rTemperatureThresholdRead),
      DESTADDR:= ADR(rTemperatureThresholdRead),
      READ := TRUE );

    iState := 3;


3 : fbIOLRead( READ := FALSE, BUSY=>bBusy, ERR=>bError, ERRID=>nErrID );

    IF NOT bBusy THEN
      IF NOT bError THEN
         iState := 4; //Success
      ELSE
         iState := 100; //Error
      END_IF
    END_IF

4 : //Compare Read and Write Value
    IF rTemperatureThresholdRead = rTemperatureThresholdWrite THEN
       iState := 0; //Success
    ELSE
       iState := 100; //Error
```

```
    END_IF

    bExecute := FALSE;

100 :  //Implement Error Handler here

END_CASE
```

4.5 Blob transfer

The BLOB transfer is not supported by IO-Link masters of the ifm organisation via EtherCAT.

ⓘ If you would like to read out BLOB data, please contact cybertronic-support@wittenstein.de to find a BLOB transfer option.

4.6 Events

cynapse® supplies so-called IO-Link events for selected operating conditions, for example when vibration or temperature threshold values are exceeded. This can be evaluated by the higher-level control system.

In IO-Link there are 3 different types of events (Error, Warning, Information). Error and warning events always have a start (Appear) and an end (Disappear). Thus, event types are two time-shifted events that are sent by the IO-Link device. Information events are so-called singleshot events. There is only one event here.

ⓘ The events supported by cynapse® are listed in the operating manual.

ⓘ In order to send events, these must be activated in cynapse®. This release is made via index 0x60. A general event release (subindex 0x01) is necessary and a parameter dependent release (subindex 0x02 - 0x07) is possible.

4.6.1 Reading out events using "Diag History"

cynapse® forwards events to the IO-Link master. This signals this by setting the status bit "Device Diag". Further information about the events can be found in the Diag History tab.
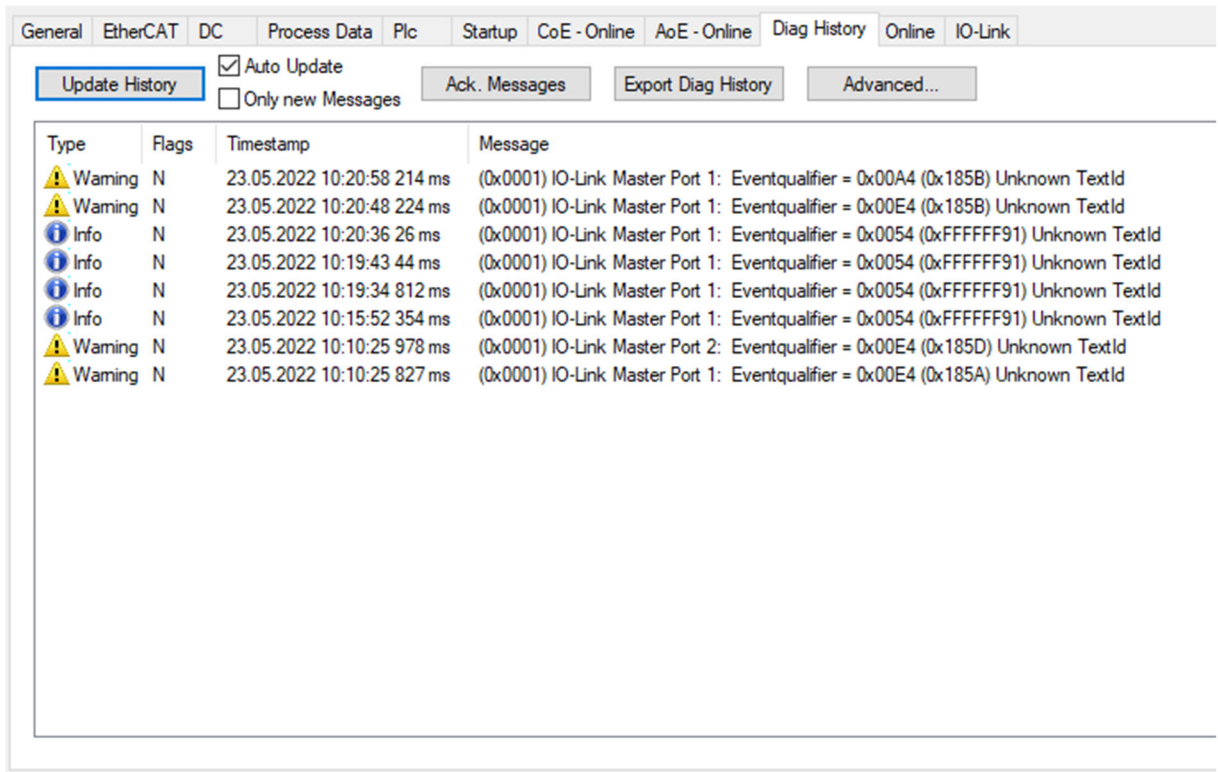

Figure 13: Diag History tab

The events that occur are determined by type (information, warning, error), flag, occurrence of the event (timestamp) and message (port number & event code) (see Figure 13). The IO-Link device can be uniquely assigned based on the port number.

4.6.2 Read out events using the "Detailed Device Status" parameter

Events of the type Error or Warning can additionally be read out using the index 0x25 "Detailed Device Status". The parameter contains only events that occur (Appear).
The parameter consists of a series of data packages, each with a length of 3 bytes.

cynapse® provides a list of 11 entries. If the values are NULL, no event is active.
The first empty entry can be canceled because the active events are at the top of the list.

Each 3 byte entry is divided into Event Qualifier (byte 1) and Event Code (bytes 2 and 3).
The interpretation of the event code can be found in the cynapse® operating manual.

Example

The cyclic query of the parameter "Detailed Device Status" index 0x25 yields the following result for the first 9 bytes:

➲ 0xE4185AE4185D000000

If you divide the answer into packets with a size of 3 bytes, you get the following result:

➲ 0xE4185A   0xE4185D 0x000000

There are two events. The third data package is empty and does not supply an entry, so the search for events can be aborted here. The first two packages contain upcoming events.
The first byte provides information about the EventQualifier. In both cases, this is 0xE4 and means that an occurring event (Appear) of the "Warning" type was sent by the device cynapse®.

ⓘ A detailed description of the EventQualifier can be found in the IO-Link specification.

The following two bytes contain the event code described in the cynapse® operating manual.

➲ 0x185A   →   The user's upper temperature threshold has been exceeded
➲ 0x185D   →   The user's vibration threshold has been exceeded

Revision history

Revision	Date	Comment	Chapter
01	08/16/2022	New version	All

WITTENSTEIN – one with the future

www.wittenstein-alpha.de