Using UR67 Profinet IO-Link Master with Codesys

Abstract:

Learn how to set up and use u-remote UR67 Profinet IO-Link Master UR67-PN-HP-8IOL-12-60M in Codesys. The document describes all steps to create a Codesys project for a Weidmüller u-control with u-OS , add the IO-Link master, and connect an IO-Link device. For simple configuration of IO-Link devices, TMG's IO-Link Device-tool can be used.

Hardware reference

No.	Component name	Article No.	Hardware / Firmware version
1	UC20-M3000	2839150000	2.1.0
2	UR67-PN-HP-8IOL-12-60M	2426260000	1.00 / 2.1.1.16
3			

Software reference

No.	Software name	Article No.	Software version
1	Codesys	-	V3.5 SP19
2	IO-Link Device-tool	-	
3	CodeSys Control SL	-	4.9.0.0-1

File reference

No.	Name	Description	Version
1			
2			

Contact
Weidmüller Interface GmbH & Co. KG
Klingenbergstraße 26
32758 Detmold, Germany
www.weidmueller.com

For any further support please contact your local sales representative:
https://www.weidmueller.com/countries

Content

1 Warning and Disclaimer

Warning
Controls may fail in unsafe operating conditions, causing uncontrolled operation of the controlled devices. Such hazardous events can result in death and / or serious injury and / or property damage. Therefore, there must be safety equipment provided / electrical safety design or other redundant safety features that are independent from the automation system.

Disclaimer
This Application Note / Quick Start Guide / Example Program does not relieve you of the obligation to handle it safely during use, installation, operation and maintenance. Each user is responsible for the correct operation of his control system. By using this Application Note / Quick Start Guide / Example Program prepared by Weidmüller, you accept that Weidmüller cannot be held liable for any damage to property and / or personal injury that may occur because of the use.

Note
The given descriptions and examples do not represent any customer-specific solutions, they are simply intended to help for typical tasks. The user is responsible for the proper operation of the described products. Application notes / Quick Start Guides / Example Programs are not binding and do not claim to be complete in terms of configuration as well as any contingencies. By using this Application Note / Quick Start Guide / Example Program, you acknowledge that we cannot be held liable for any damages beyond the described liability regime. We reserve the right to make changes to this application note / quick start guide / example at any time without notice. In case of discrepancies between the proposals Application Notes / Quick Start Guides / Program Examples and other Weidmüller publications, like manuals, such contents have always more priority to the examples. We assume no liability for the information contained in this document. Our liability, for whatever legal reason, for damages caused using the examples, instructions, programs, project planning and performance data, etc. described in this Application Note / Quick Start Guide / Example is excluded.

Security notes
In order to protect equipment, systems, machines and networks against cyber threats, it is necessary to implement (and maintain) a complete state-of-the-art industrial security concept. The customer is responsible for preventing unauthorized access to his equipment, systems, machines and networks. Systems, machines and components should only be connected to the corporate network or the Internet if necessary and appropriate safeguards (such as firewalls and network segmentation) have been taken.

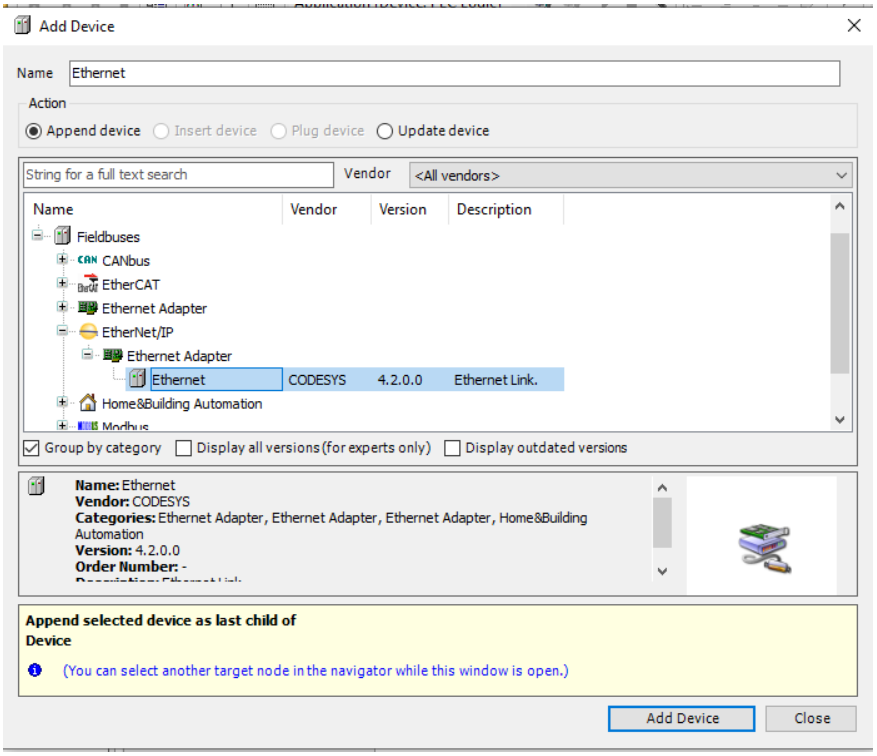
2 Codesys project with Profinet

The following steps briefly explain how to create a new project Codesys project and prepare the Profinet interface.

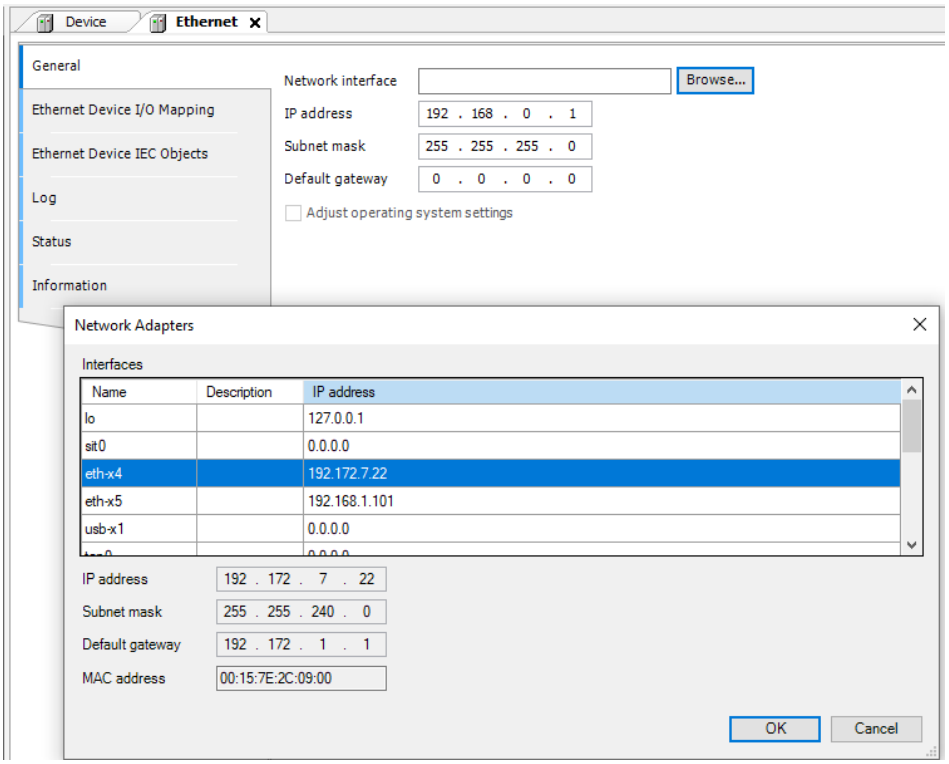
- Start by opening Codesys and creating a new standard project.
- Select the controller to be used, e.g., u-control M3000.

Once the project is created, a project tree will appear on the left side. We will now add the Profinet interface.

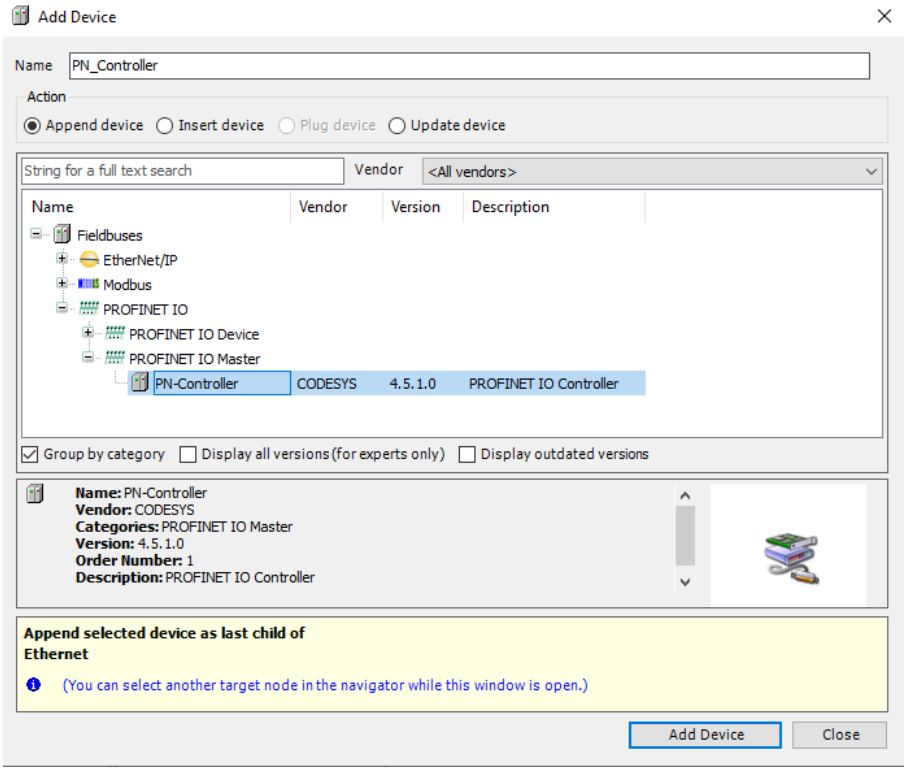
- Right click on the device (“UC20-M3000”, in the example)
- Select “Add Device…”
- Look for and select “Ethernet” then click on “Add Device”:



- Once the device is added, it will appear in the project tree under our device. Double click on the new adapter to assign a physical network interface.
- Click on “Browse” choose the network adapter of the controller. Then click “OK”.



- Next step is to add a Profinet Master. Right click on our ethernet adapter in the device tree and select “add device”:
- Select “PN-Controller” under “Profinet Master”, then click on “Add Device”.

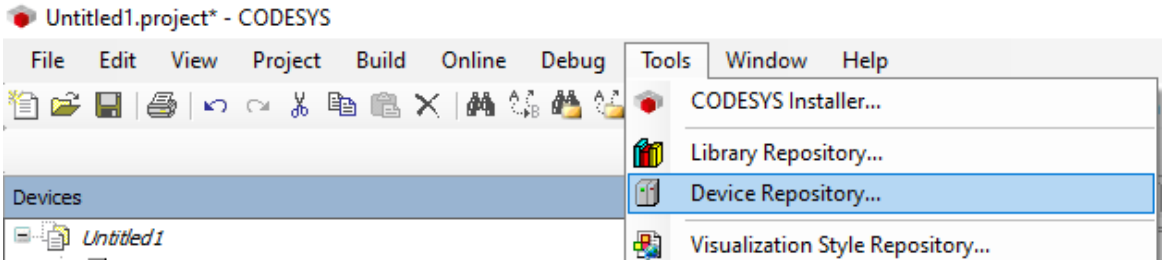


- Now, the selected Ethernet port can be used for Profinet.

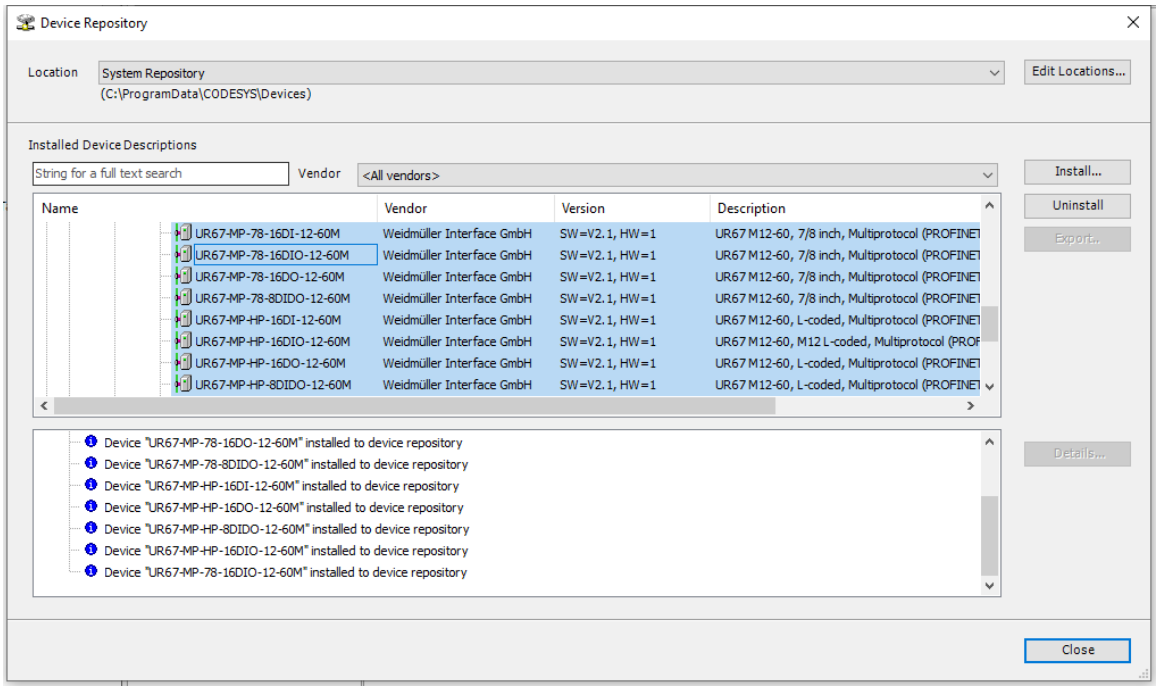
3 Connection and setup of the UR67 IO-Link Master Module

We will now add the UR-67-PH-HP-8IOL-12-60M module to our project. This module will serve as IO-Link master. IO-Link slaves like sensors can be connected to this device. Before adding the device, we need its GSDML file, which is available from the Weidmueller product catalog. After downloading and extracting the file, we can import it to our Codesys project.

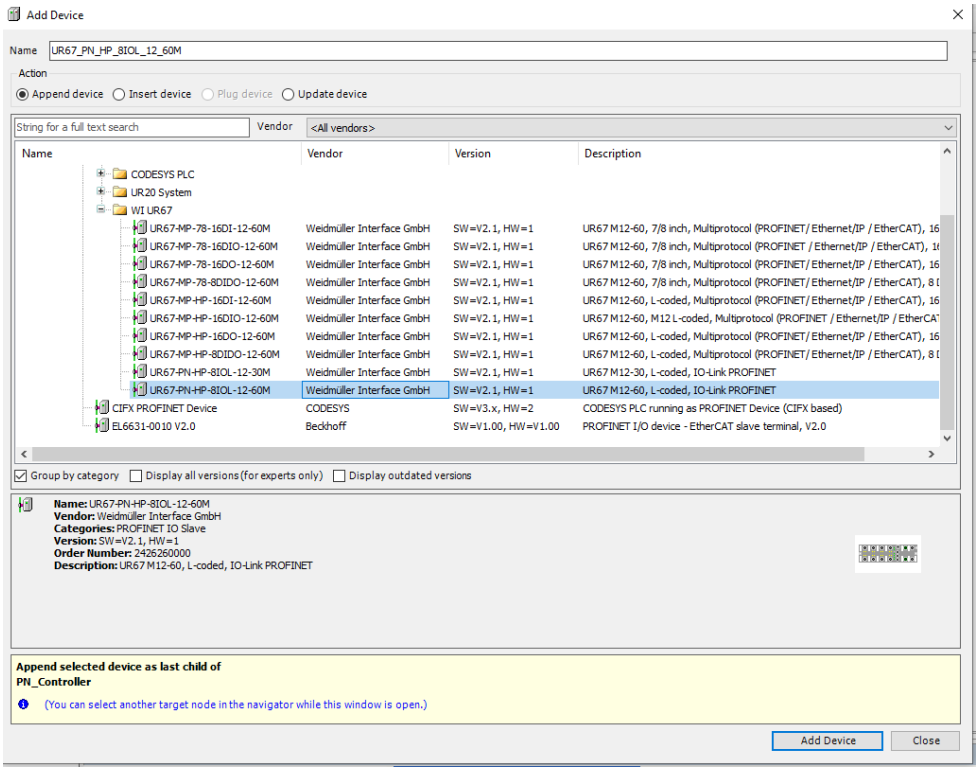
- Head to “tools” and then to “Device Repository”.



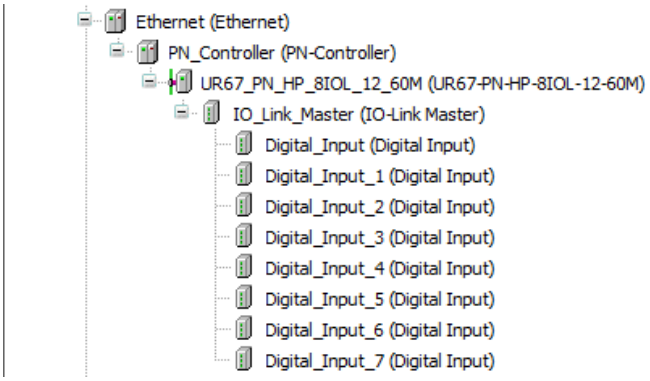
- Click “Install” and look for the downloaded file.
- Select the file and click “Open”
- After the installation was successful, we see multiple new entries in the device repository and click on “Close”.



- We can now add the device to our project tree by right-clicking on “PN_Controller” and selecting “Add Device”.
- Look for the correct device in the list and click on “Add Device”.



- Our new device will appear in the project tree under “PN_Controller”. Note that all channels/ports are by default set up as “Digital_Input”



- Since we want to use the device as an IO-Link Master, we need to replace the default channels by IO-Link channels. Therefore, delete the at least one (or all) of the channels by right-clicking and choosing “Delete”. In the next chapter, we will see how to add the IO-Link channels.

4 Connecting and using IO-Link devices

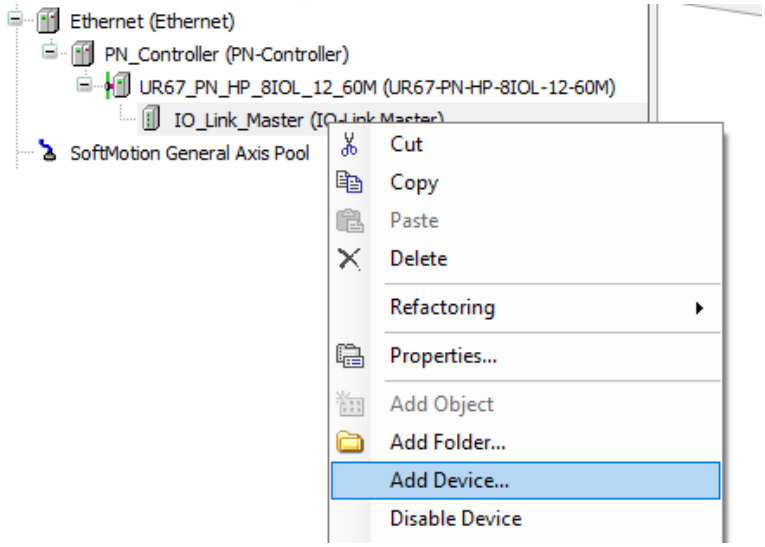
As an example, we will now connect an IO-Link vibration sensor to our IO-Link master and read the sensor data with our Codesys Application.

4.1 Adding an IO-Link Port

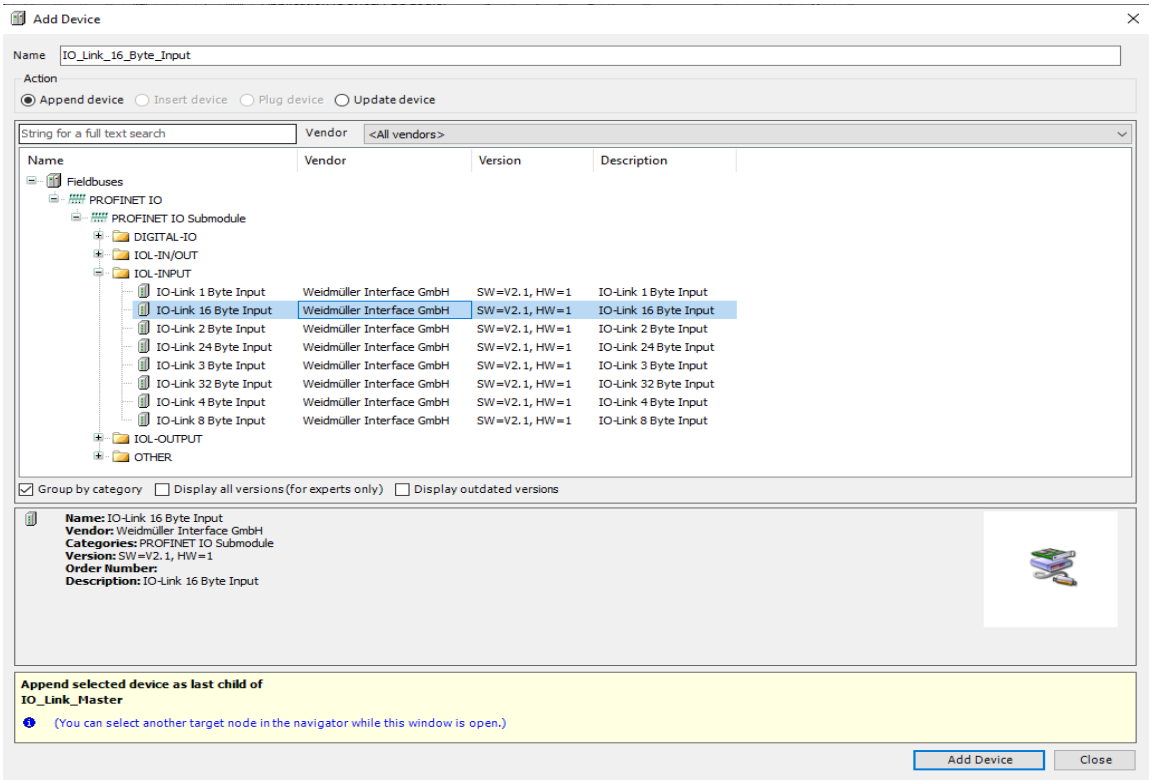
We can now add an IO-Link to our configuration. The UR-67-PH-HP-8IOL-12-60M handles each port as a submodule that can be added to the device. There are different submodules depending on the needed process data size.
Thus, we now need to know the process data size of our IO-Link device, which can be found in its manual.

For our example, we use a sensor which provides read-only process data with the size of 16 bytes. So, we need to add a 16 Bytes input IO-Link channel.

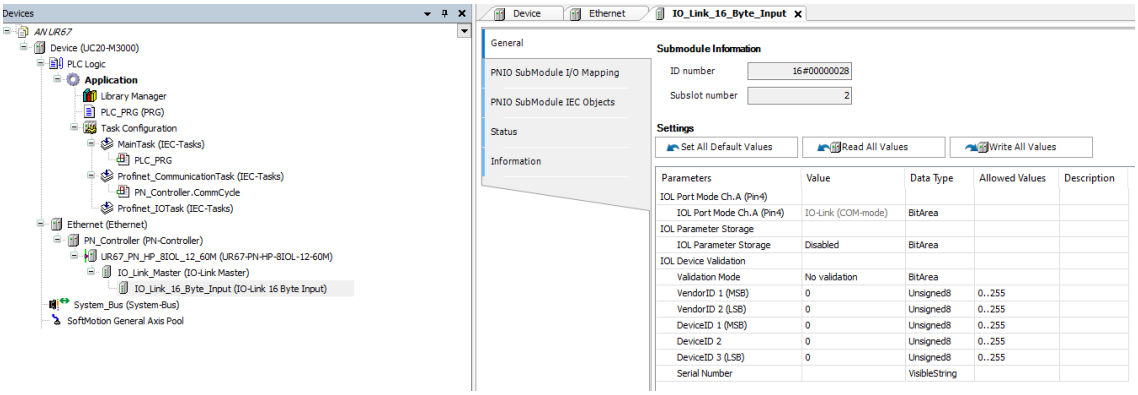
- Right-click on the “IO_Link_Master” and select “Add Device”.



- We look for the “IO-Link 16 Byte Input”, select it and click on “Add Device”.



- The Module will appear in the device tree:



4.2 Reading the data

There are two types of data that can be read from an IO-Link device:

- Process data which is sent cyclically.
- Parameter Data that needs to be requested.

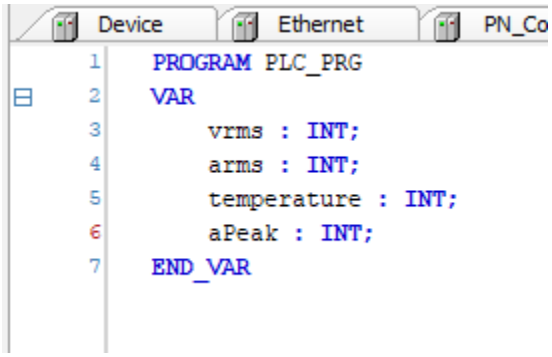
Process data (cyclic)

We will now create variables in the PLC program to be able to read the process data sent from our example sensor. We will monitor four variables:

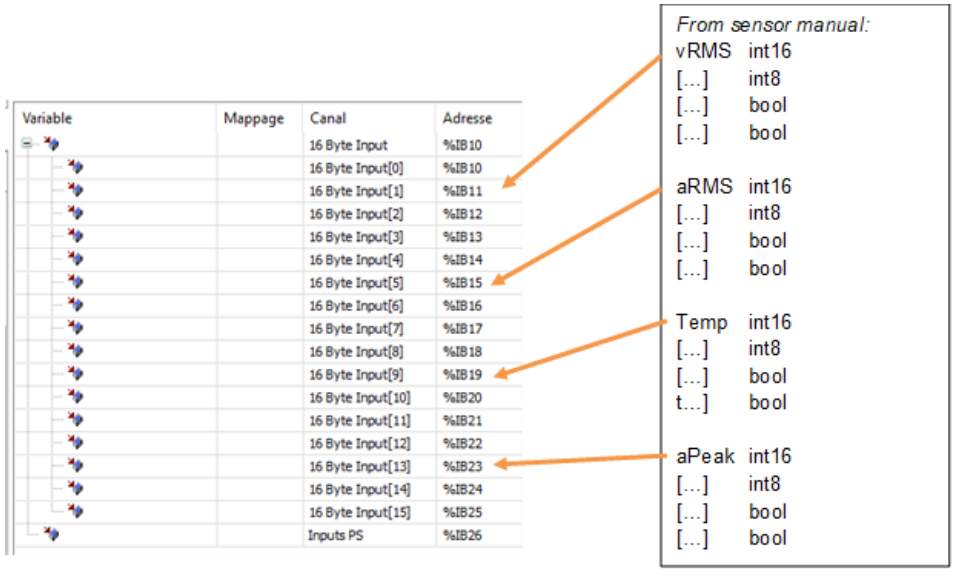
- vRMS
- aRMS
- aPeak
- temperature

According to the data sheet of our sensor, all four are of type INT16:

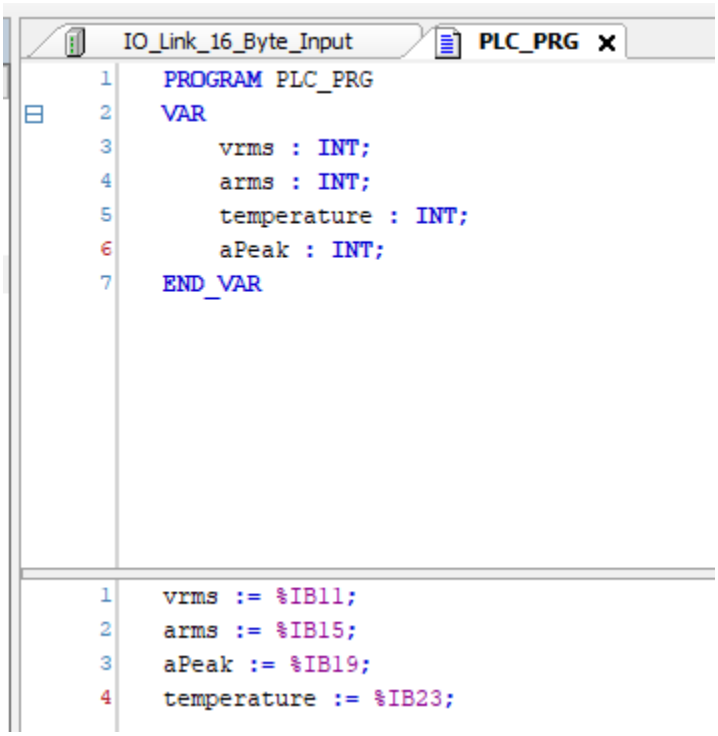
So, the four variables we will create will all be of the same type. We head to our PLC Program on CODESYS and declare four variables:



We now need to map the variable. In the Mapping section of the Submodule, we can find 16 inputs. Referring to the manual of our sensor, we can see which data coincides with which input:



So, a simple PLC Program might look like this:



Parameter data (acyclic data)

To read the parameter data we will need to use the IOL_Call function which is used in the context of PROFINET and IO-Link communication.
It is part of the Codesys ProfinetCommon library and is specifically used for interacting with IO-Link devices. You can read more about it here.

We need to declare the following variables for a simple acyclic reading of a string:

```
PROGRAM Main
VAR
    al_read : ProfinetCommon.IOL_CALL;
    err : ProfinetCommon.IOL_Error;
    productText : STRING(16);
    req_var : BOOL;
```

- "al_read" and "err" are instances of the needed library function blocks.
- "req_var" will be used to enable the call, while "productText" will hold the data to be read.

In our program, we only need to call the function block "al_read" to read data from our sensor. The required parameters for the call depend on the specific IO-Link devices. For "IOL_Index", "IOL_Subindex", length, and datatype, we refer to the manual of our IO-Link slave. Parameter "ID" is the ID of the IO-Link Master connected via Profinet and can be read via "GetID". "CAP" is always 255 for the UR-67-PH-HP-8IOL-12-60M. Note that the call takes multiple cycles, so we need to check the status to see when it is done. Here is some example code:

```
al_read.REQ := req_var;

al_read.ID := UR67_PN_HP_8IOL_12_60M.GetID(0,1,1);        // ID of IO-Link Device (in P
al_read.CAP := 255;                                       // Fix for Ed. 2 devices. For legacy devices get va
al_read.Port := 1;                                         // IO-Link device is plugged in Port 1
al_read.RD_WR := FALSE;                                   // Read data
al_read.IOL_Index := 18;                                   // IOL-Index = 20 means 'Product Text', this wi
al_read.IOL_Subindex := 0;                                 // 0 = read complete data-item
al_read.LEN := TO_INT(SIZEOF(productText) - 1);
al_read.IOL_Data := ADR(productText);                     // an ARRAY[] OF BYTE will be r

al_read();
IF(al_read.ERROR) THEN
    IF(al_read.STATUS <> 0) THEN
        // Profinet Communication-Error
        // e.g. DF80A100 (= PNIORW, application: write error) will be returned if a wrong 'ID' /
    ELSE
        // get IO-Link related error:
        err := ProfinetCommon.DECODE_IOL_STATUS(al_read.IOL_STATUS);
    END_IF
ELSIF(al_read.DONE_VALID) THEN
    ;// --> the variable 'productText' will contain some text
END_IF
```

Writing parameter data to the IO-Link device works in a similar way.

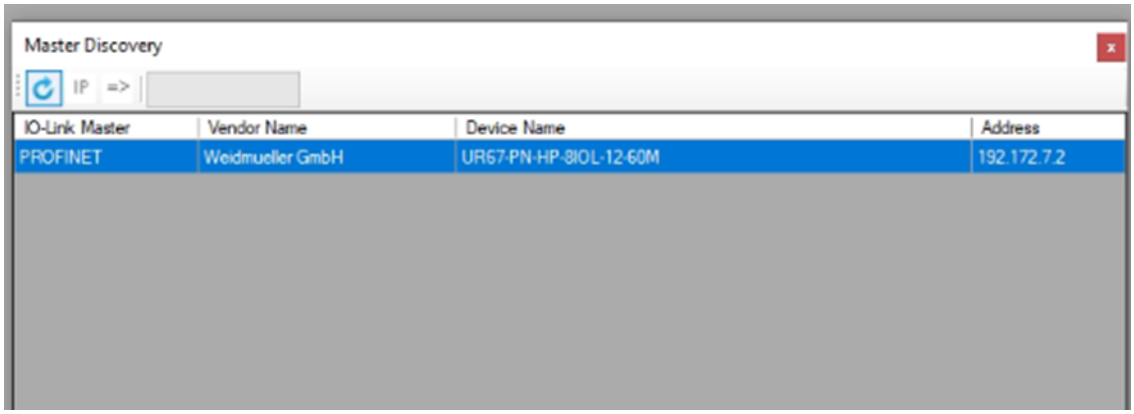
5 IO-Link Device configuration with an external tool

As seen in the previous chapter, reading, and writing acyclic data (parameters) of IO-Link devices from within a PLC program is quite simple.
Sometimes, it is more convenient to use an IO-Link configuration software for setting up and testing IO-Link devices, without Codesys or another PLC Engineering tool.

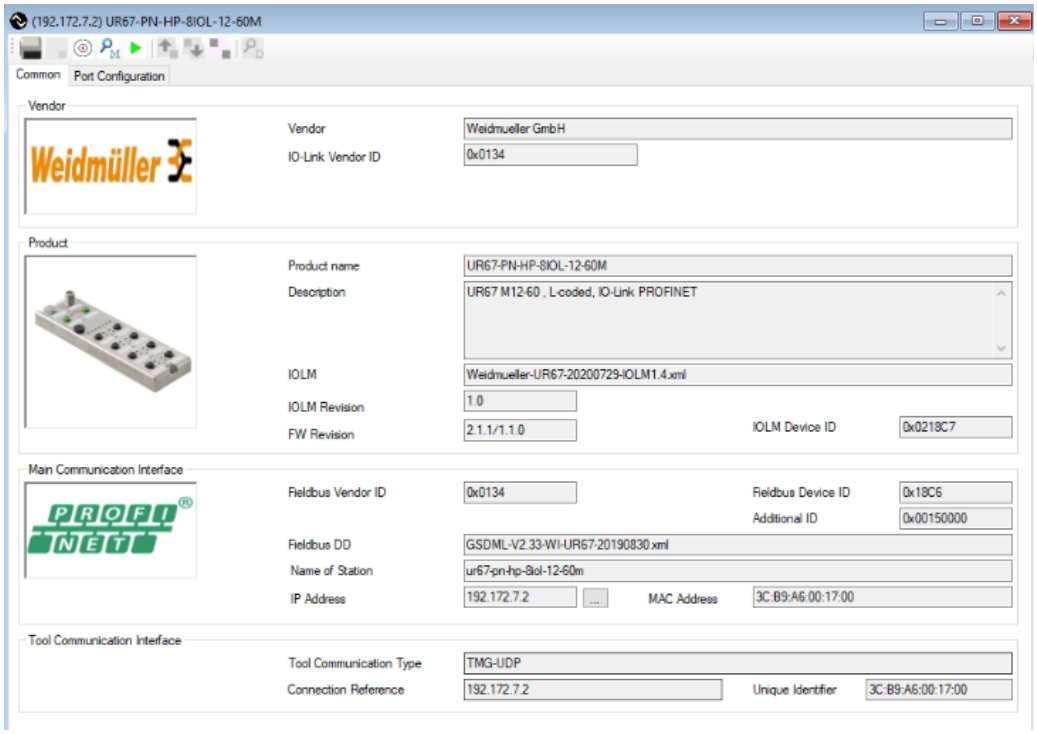
In this example, we use the IO-Link Device-Tool, a 3rd party software that can be downloaded from TMG's website and offers a free trial period.
In addition to the software tool, we need to download the IOLM file for the UR-67-PH-HP-8IOL-12-60M from the Weidmüller website and the IODD file for our IO-Link Slave device.

The configuration of the UR-67-PH-HP-8IOL-12-60M is possible either via the integrated web interface (see manual) or with the IO-Link Device-Tool.

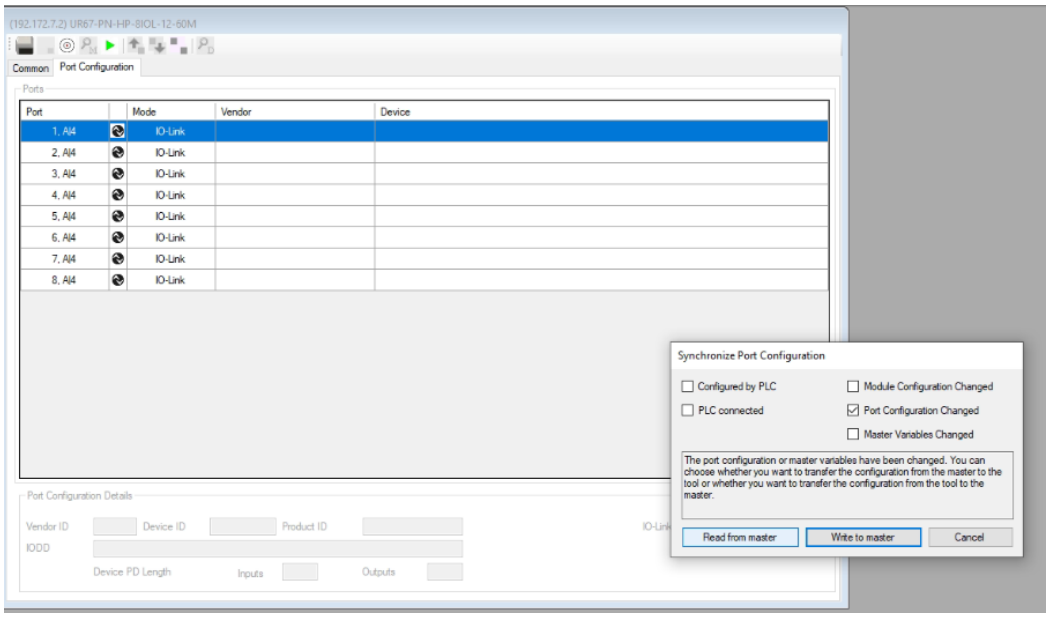
1. Make sure the IO-Link slave is connected to the IO-Link Master.
2. Open the IO-Link Device Tool.
3. First, we must import the device description of the IO-Link Master. Go to "Options", then "Import IOLM file"
4. Search for the IOLM file and import it.
5. Now, we will import the IO-Link slave device descriptions. Go to "Options", then "Import IODD file".
6. Import IODD files for the IO-Link slaves.
7. We can now start configuring our device. We click on "Search Master" to detect the Io-Link devices in our network. The tool will list all IO-Link devices in the network:



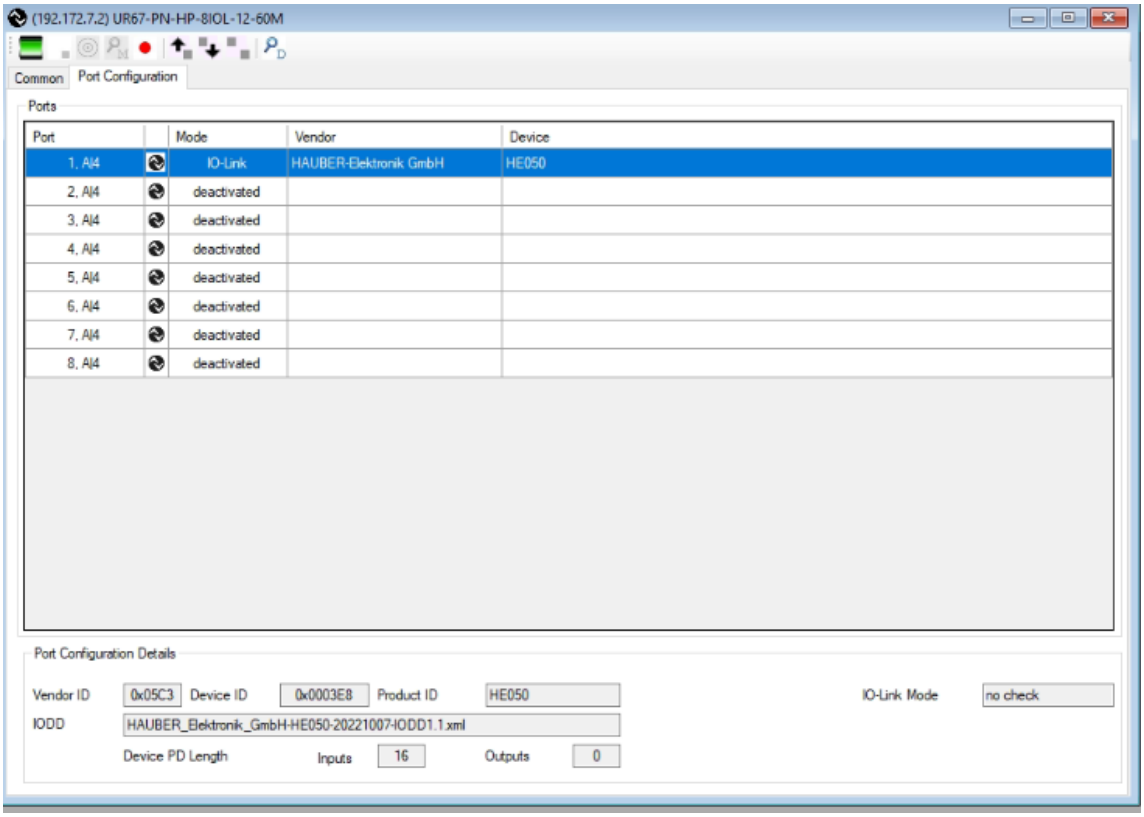
8. Double-click on the device opens a window with all the info about the device:

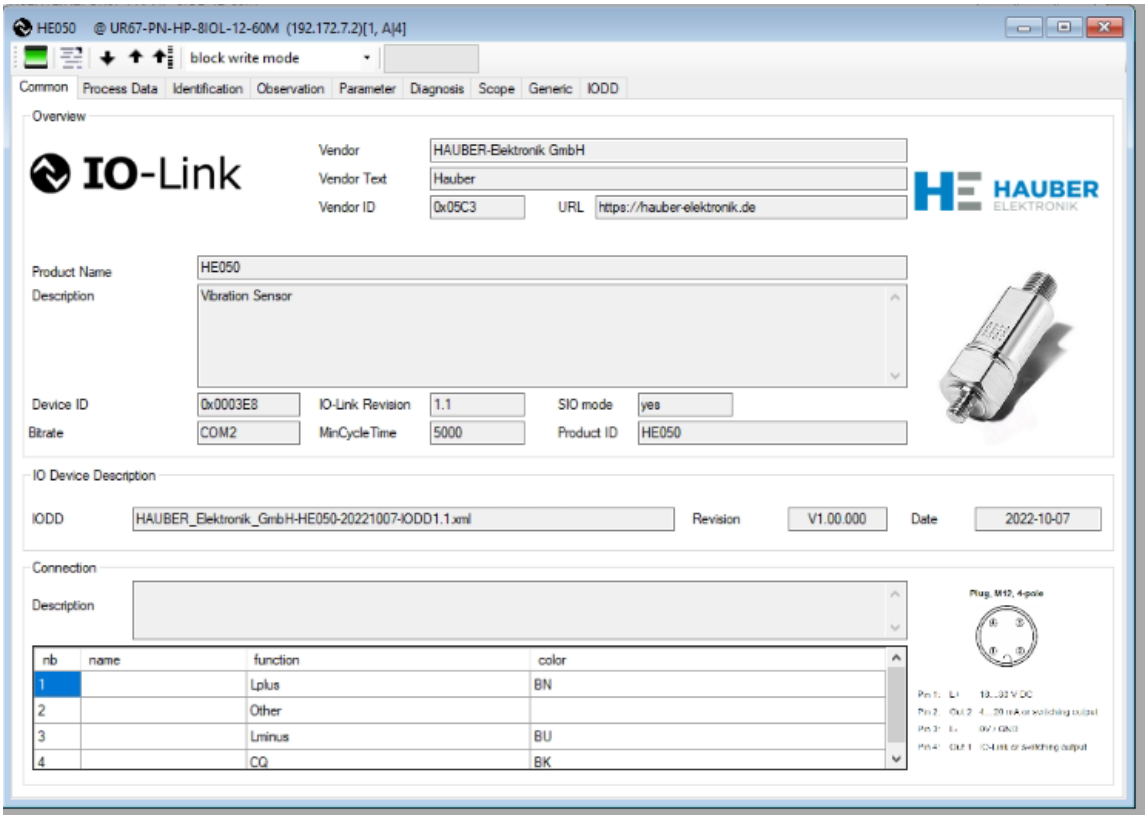


9. Head to “Port Configuration” and set the right mode for each port. By default, the ports are set as “DI”. For IO-Link, the used ports must be set to IO-Link. (right-click on the channel, select IO-Link)
10. We can now go online by clicking on the green play button and look for connected devices. Upon clicking on the run button, a window appears and we must select “Read from Master”.



11. We can now click on “check devices”. A window appears and we can see IO-Link devices connected to the master. We then click on “Take devices into engineering”
12. Our IO-Link device will now be listed at the corresponding port. Double clicking on it’s name opens a new window with device specific info. Here, we can configure the device or check parameters.


(192.172.7.2) UR67-PN-HP-8IOL-12-60M
Common
Port Configuration
Ports
Port
Mode
Vendor
Device
1, AI4
IO-Link
HAUBER-Elektronik GmbH
HE050
2, AI4
deactivated
3, AI4
deactivated
4, AI4
deactivated
5, AI4
deactivated
6, AI4
deactivated
7, AI4
deactivated
8, AI4
deactivated
Port Configuration Details
Vendor ID
0x05C3
Device ID
0x0003E8
Product ID
HE050
IO-Link Mode
no check
IODD
HAUBER_Elektronik_GmbH-HE050-20221007-IODD1.1.xml
Device PD Length
Inputs
16
Outputs
0


HE050 @ UR67-PN-HP-8IOL-12-60M (192.172.7.2)[1, AI4]
block write mode
Common
Process Data
Identification
Observation
Parameter
Diagnosis
Scope
Generic
IODD
Overview
IO-Link
Vendor
HAUBER-Elektronik GmbH
Vendor Text
Hauber
Vendor ID
0x05C3
URL
https://hauber-elektronik.de
HAUBER
ELEKTRONIK
Product Name
HE050
Description
Vibration Sensor
Device ID
0x0003E8
IO-Link Revision
1.1
SIO mode
yes
Bitrate
COM2
MinCycleTime
5000
Product ID
HE050
IO Device Description
IODD
HAUBER_Elektronik_GmbH-HE050-20221007-IODD1.1.xml
Revision
V1.00.000
Date
2022-10-07
Connection
Description
nb
name
function
color
1
Lplus
BN
2
Other
3
Lminus
BU
4
C/Q
BK