TECH NOTE :: DSE_Checkweigher with PLC (R-PI 4B)

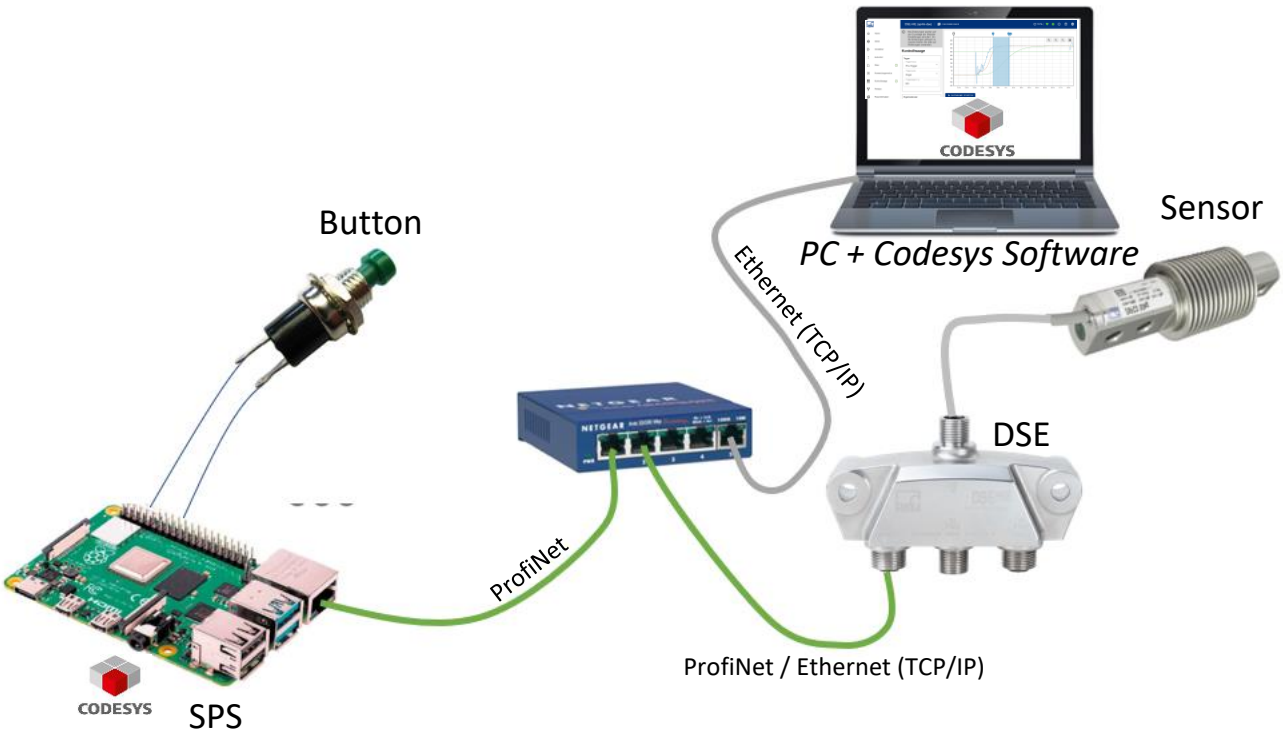
Version: 2022-04-26
Author: Michael Guckes, Patrick Schöpper
Status: HBM: Public

Brief description

This is a quick start guide for setting up a checkweigher with the DSEs (from FW 2. 0). The control scale can be controlled in two ways, on the one hand via a level and on the other via an external signal (in this example a pushbutton). The DSE has no hardware DIO's, so the trigger signals are captured by a PLC and transmitted to the DSE, here a Raspberry Pi with Codesys as PLC, via PROFINET and EtherCAT.

For this purpose, a Codesys Runtime license and a finished Codesys project "DSE-Checkweiger-R-PI. projectarchive" will be loaded into the R-PI. Used Codesys version is 3. 5. 17. 10 Patch 3 or higher.

Hardware configuration

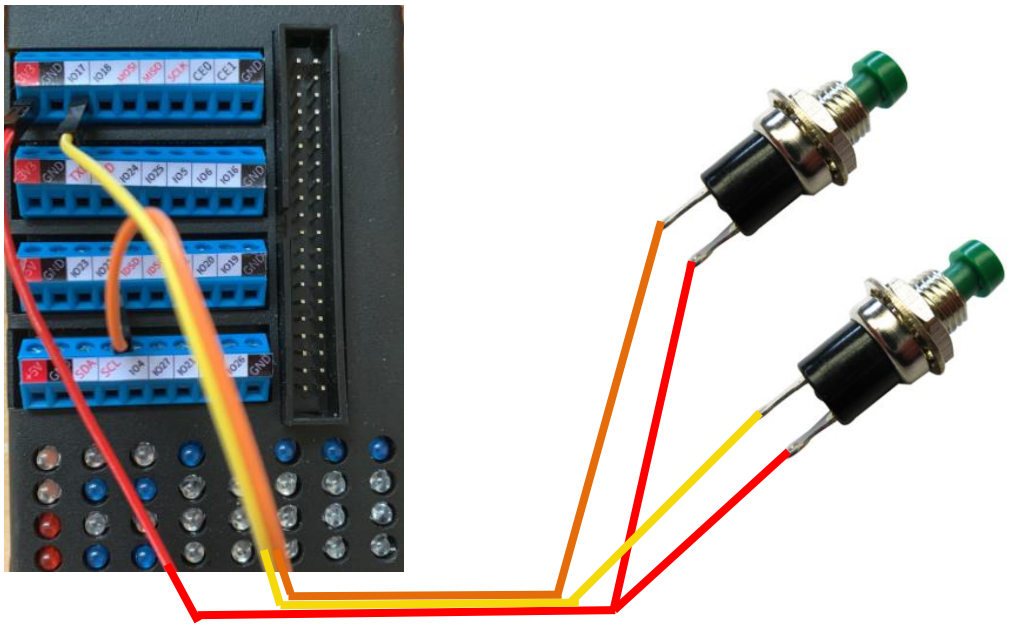


Aufbauskizze

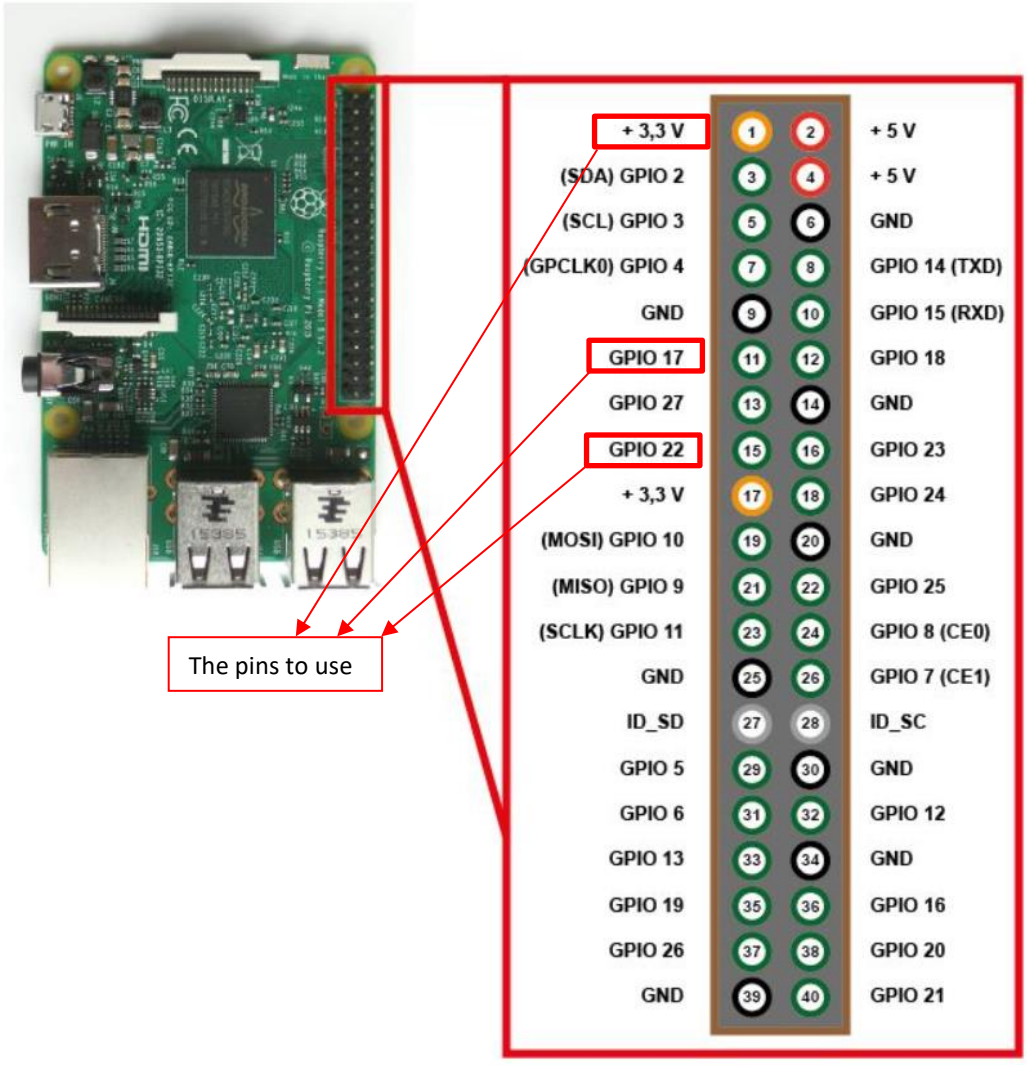
Components required

- 1x DSE system (incl. power pack and Ethernet cable)
- 1x free Codesys software
- 1x Ethernet switch
- 1x SPS, R-PI Model 4B (Codesys Controle for Rasperry Pi V 4.4.0.0)
- 2x Switch button
- 1x load cell

Hardware configuration of the GPIO's at the PI

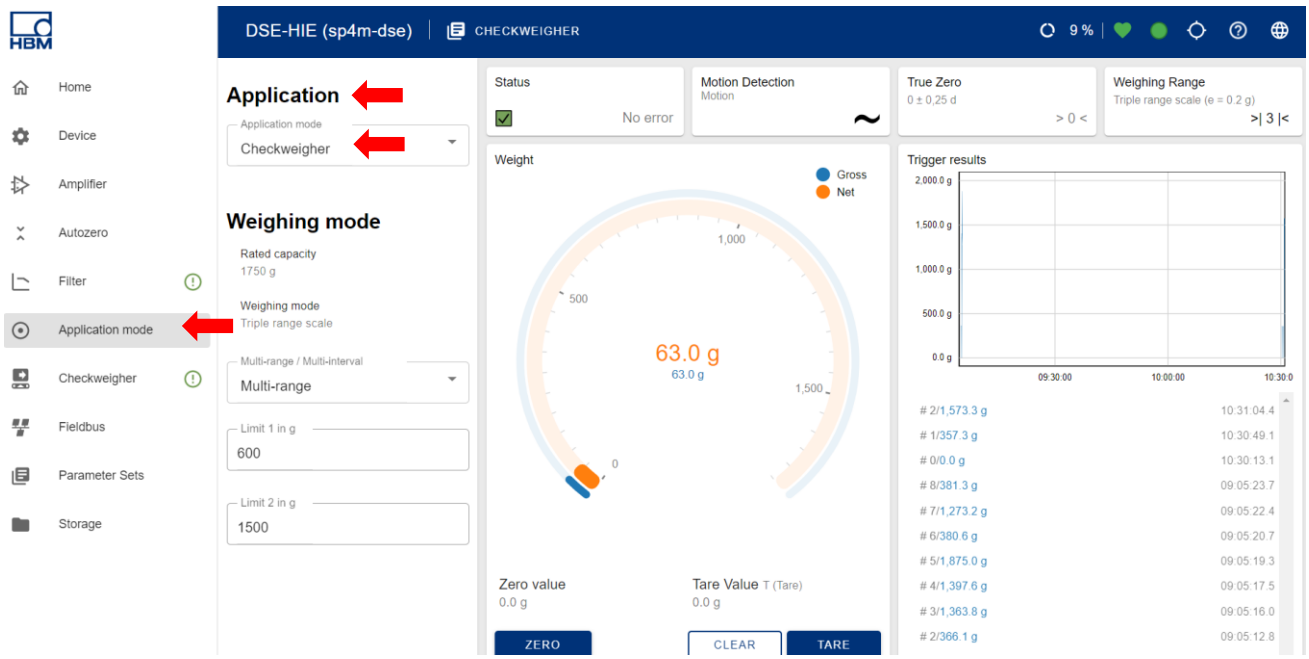


Pin Assignment of the Raspberry 3 and 4

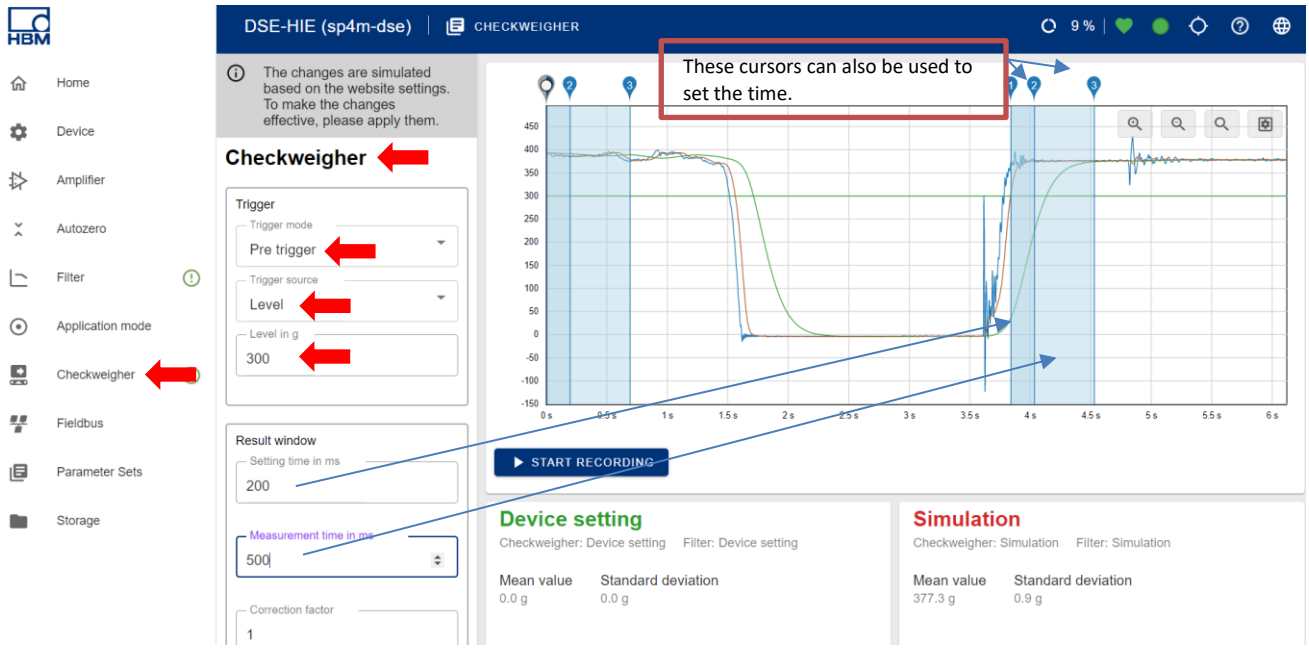


Level controlled

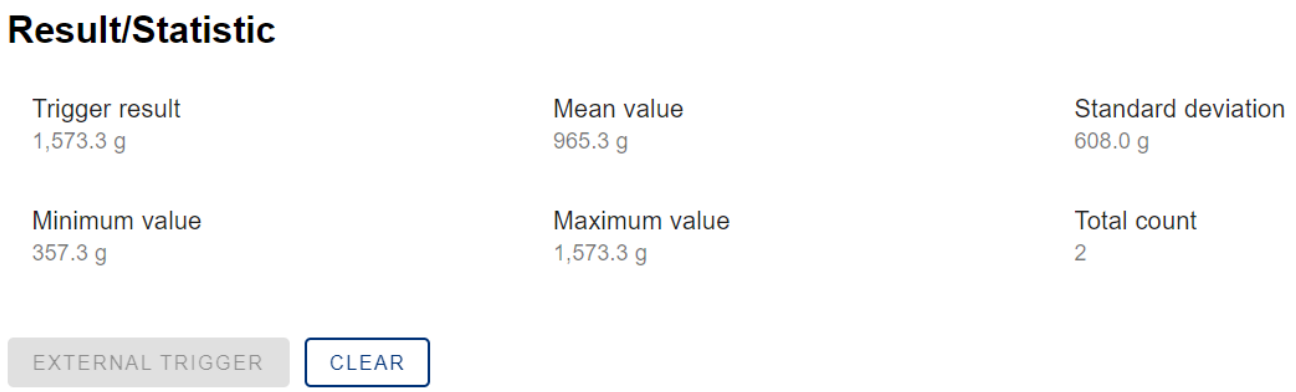
With level control, the measurement process is started when a certain value (level) is above or below. This means when the measurement starts depends very much on the preconfigured parameters.



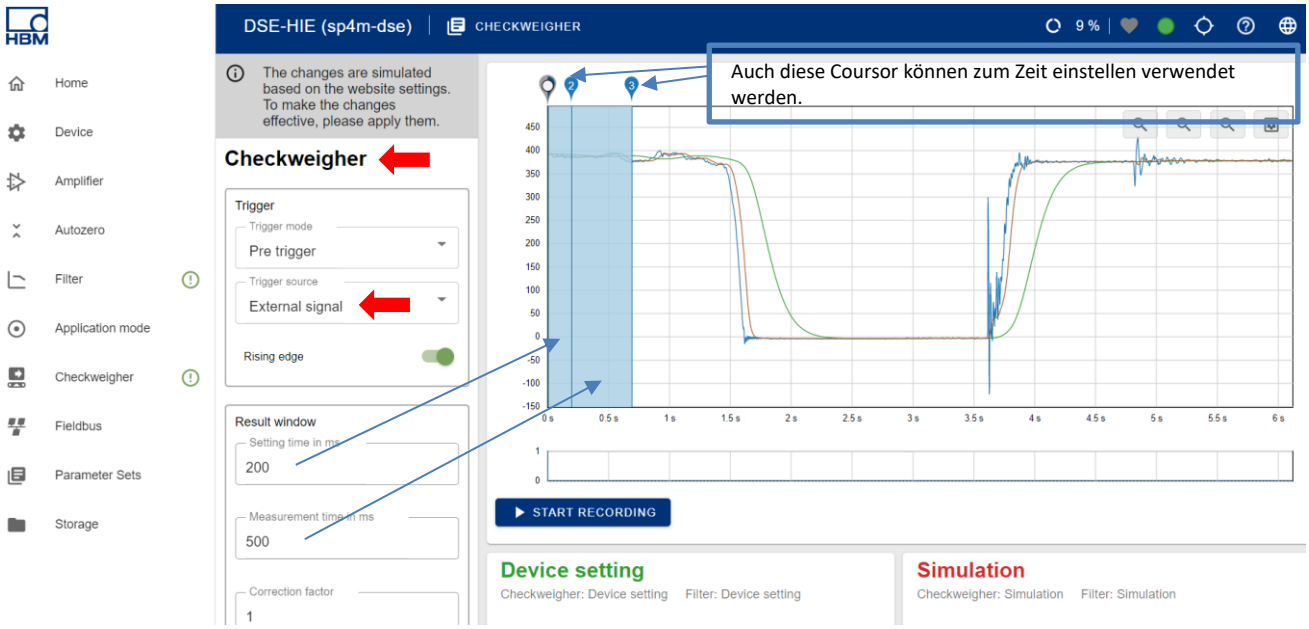
Now switch to the checkweigher. Under “Trigger source”, “Level” must be set. The height of the level shall be set in the following field. A low level is recommended for testing, which is definitely exceeded at the expected load. The level is shown in green in the diagram. In the result window you can first set the vibration time. This starts with the exceeding of the filter result (brown) and should end after the switching, this time must be determined by test runs. Next, the measurement time can be created, which should be as short as possible but still long as possible to obtain a correct result.



In the **“Results/Statistics”** window, you can visualize the different values that can be recorded with the system.



Control via external signal (button or light barrier)



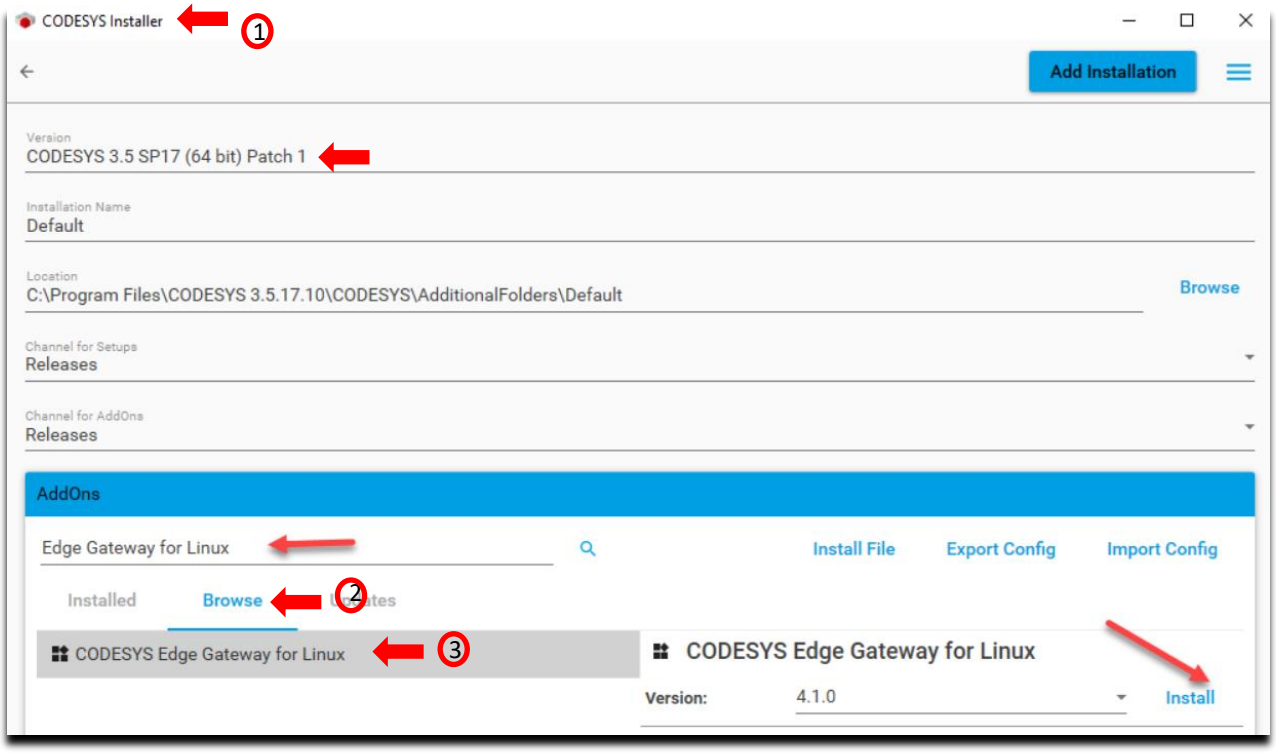
When using an external signal trigger, the start of the measurement process is detached from the course of the measurement signal. The external signal can be generated in many ways, e. g. by a light barrier or a button, but it is always the data type Bool, which means “true or false” or “1 or 0. ” In the example above you can see that the measurement process was started together with the recording, even if there has not yet been any change in the course.

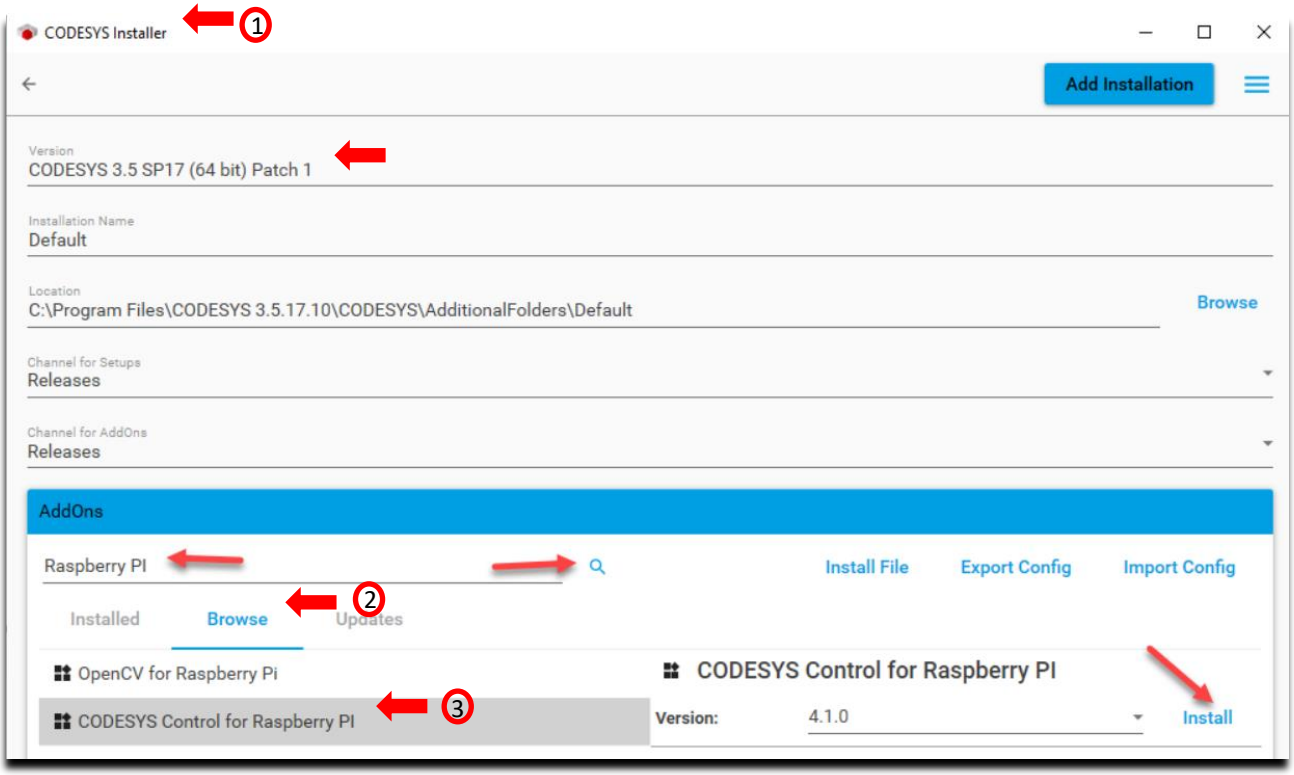
The check-weighing process can be triggered with a pre-trigger before the load cell is loaded or with a post-trigger after the load cell is loaded.

Install Codesys (required once)

The free Codesys software can be downloaded here:
https://store.codesys.com/de/codesys.html

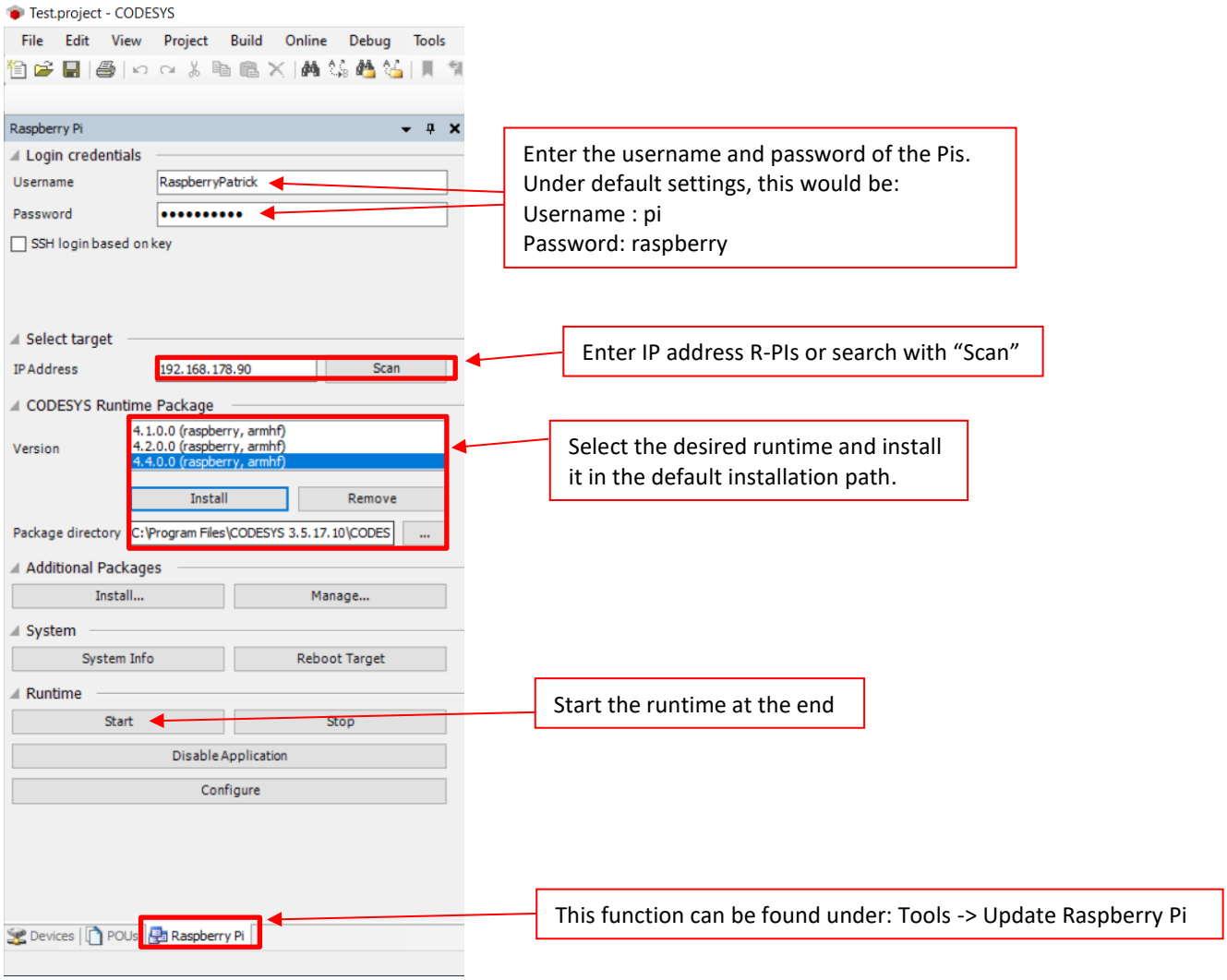
All further installations via the Codesys Installer.
Open the Codesys Installer:
First install the “Edge Gateway for Linux” then the “Codesys Control for Rasperry Pi ”.



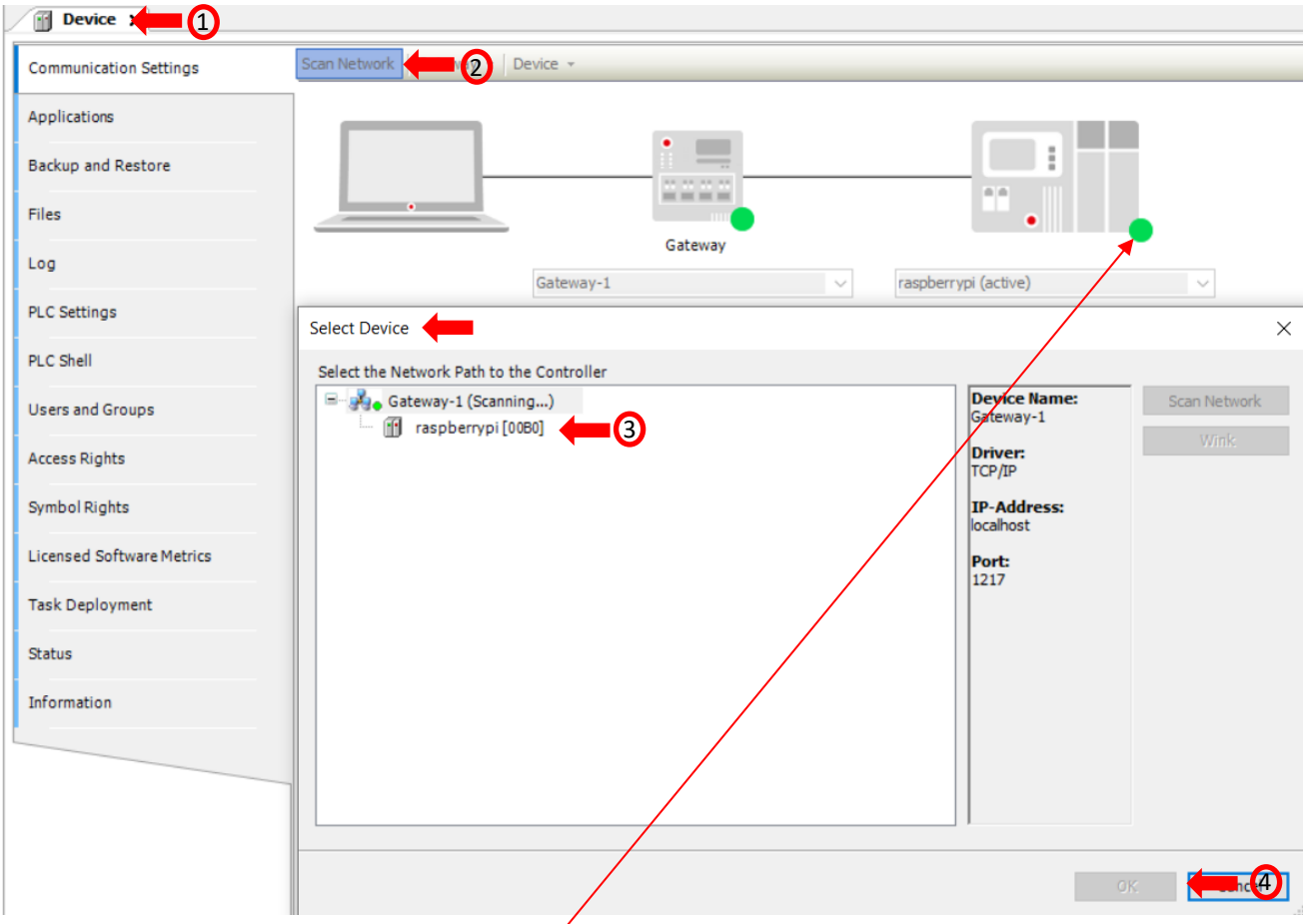


Installation of the Codesys Runtime on the Raspberry Pi (required once)

After the installation, start Codesys on your computer by clicking on the icon.
After Codesys is started, the runtime must be loaded on the Pi.



Now the Raspberry can be found under “Devices”:

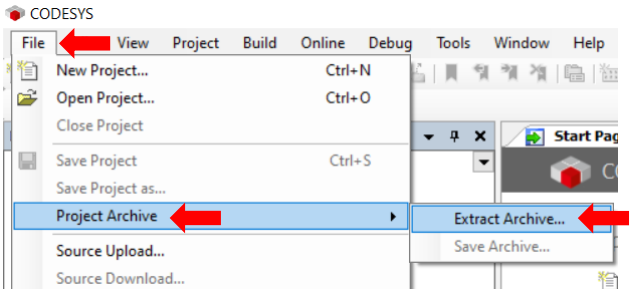


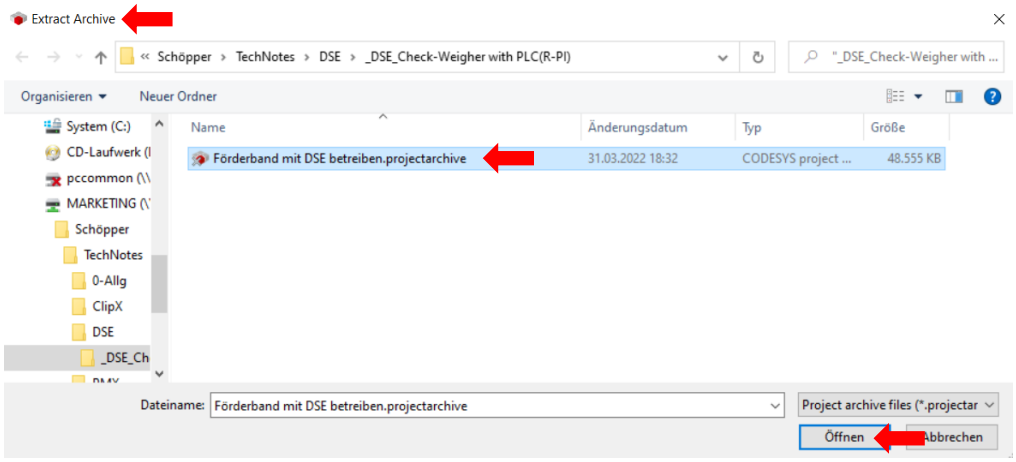
After pressing “OK” you still have to log in to the Pi. This works the same way as when installing the runtime. If after this process the LED lights up green, the connection setup has worked.

Import Project (one-time need)

The project contains all necessary libraries including the device description file (GSDML) of the DSE and only needs to be imported once.

Open Codesys on your computer.

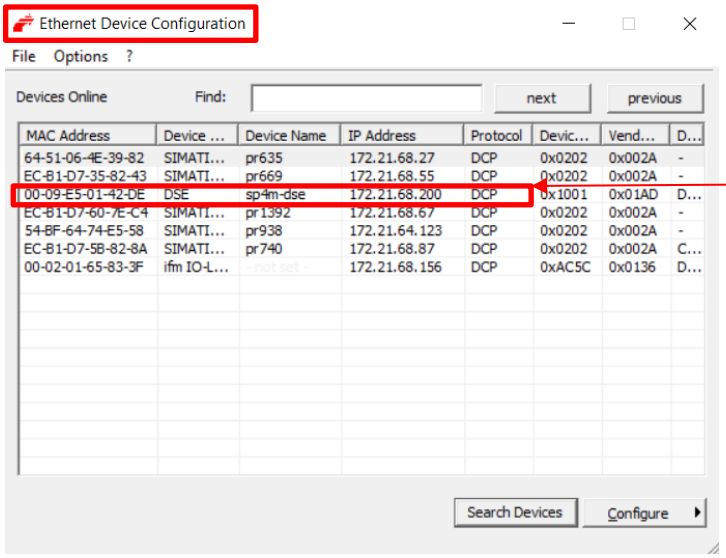




After clicking Open, the project is imported into Codesys with all libraries and devices.

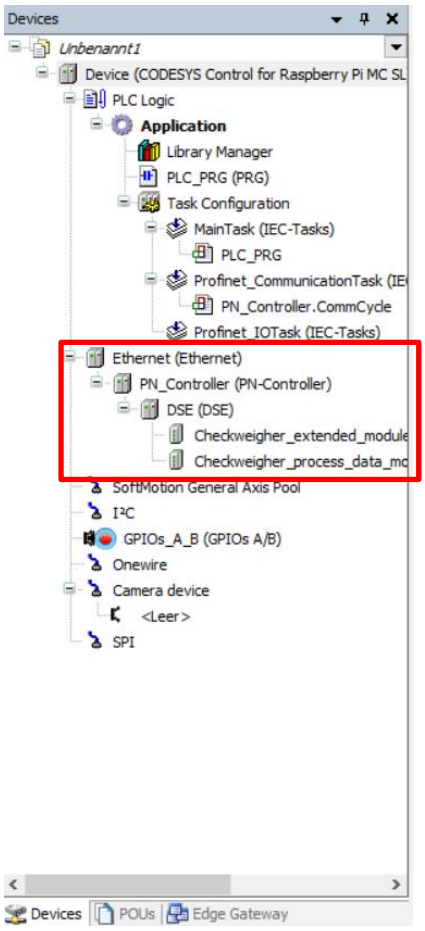
Network configurations

Note: The IP address and device name of the DSE can be configured using the “Ethernet Device Setup” tool.

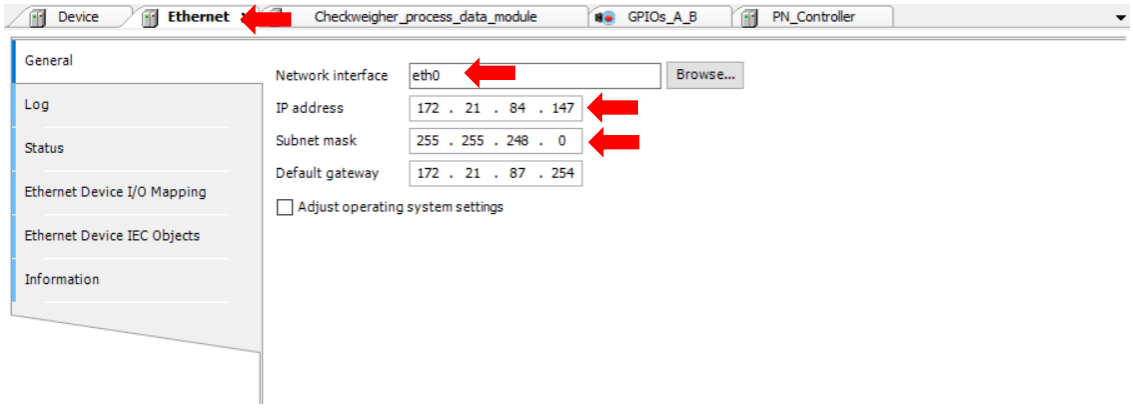


Rechtsklick -> Set IP/Device Name zum konfigurieren
Hinweis: Die IP muss zu der IP-Adresse des R-Pis passen, d.h. ersten 3 Blöcke müssen gleich sein, der letzte unterschiedlich.

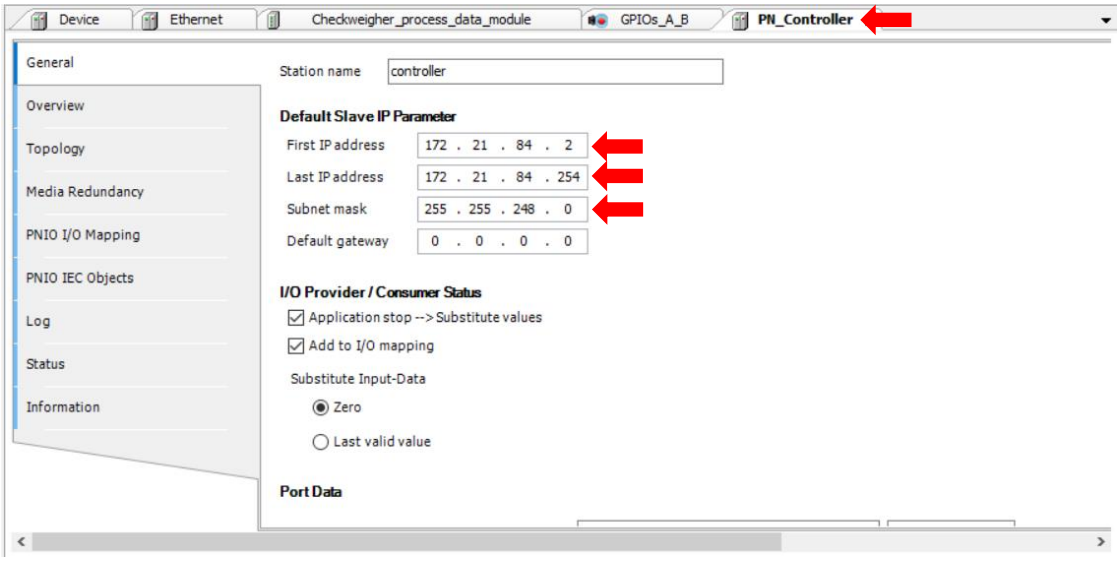
Here you can see the different adapters to connect the DSE to the Codesys Control. These must now be configured for the network.



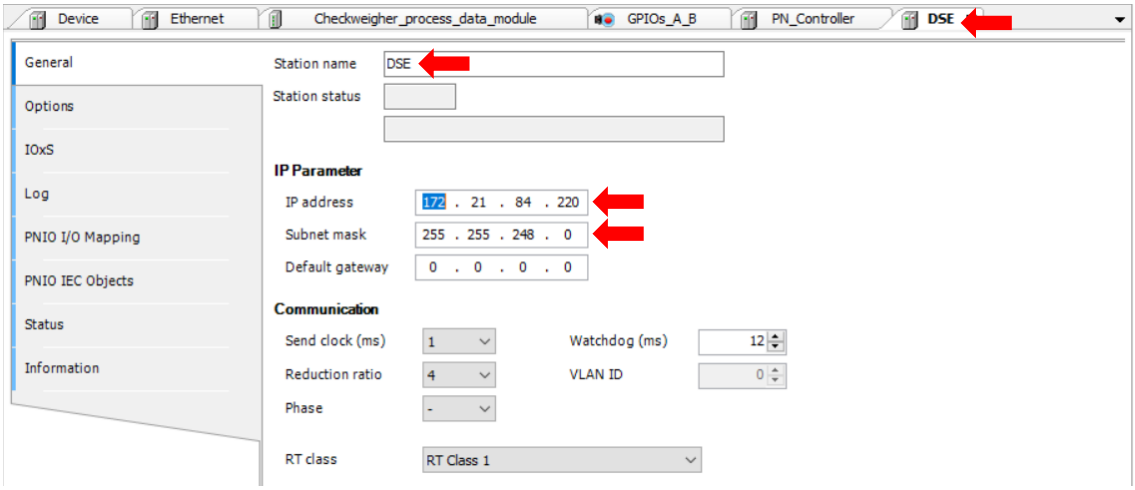
Since the Codesys main program must communicate with the application on the Raspberry, they must be connected via PROFINET. This is explained below.
First, the network address of the Ethernet adapter needs to be adjusted. To do this, click on “Browse”. . . and select “eth0. ”



After that, the IP address of the “PN_controller has to be adjusted, so it is important that the first 3 columns are identical to the first 3 columns of “eth0. ” The subnet mask must also match that of “eth0. ”.

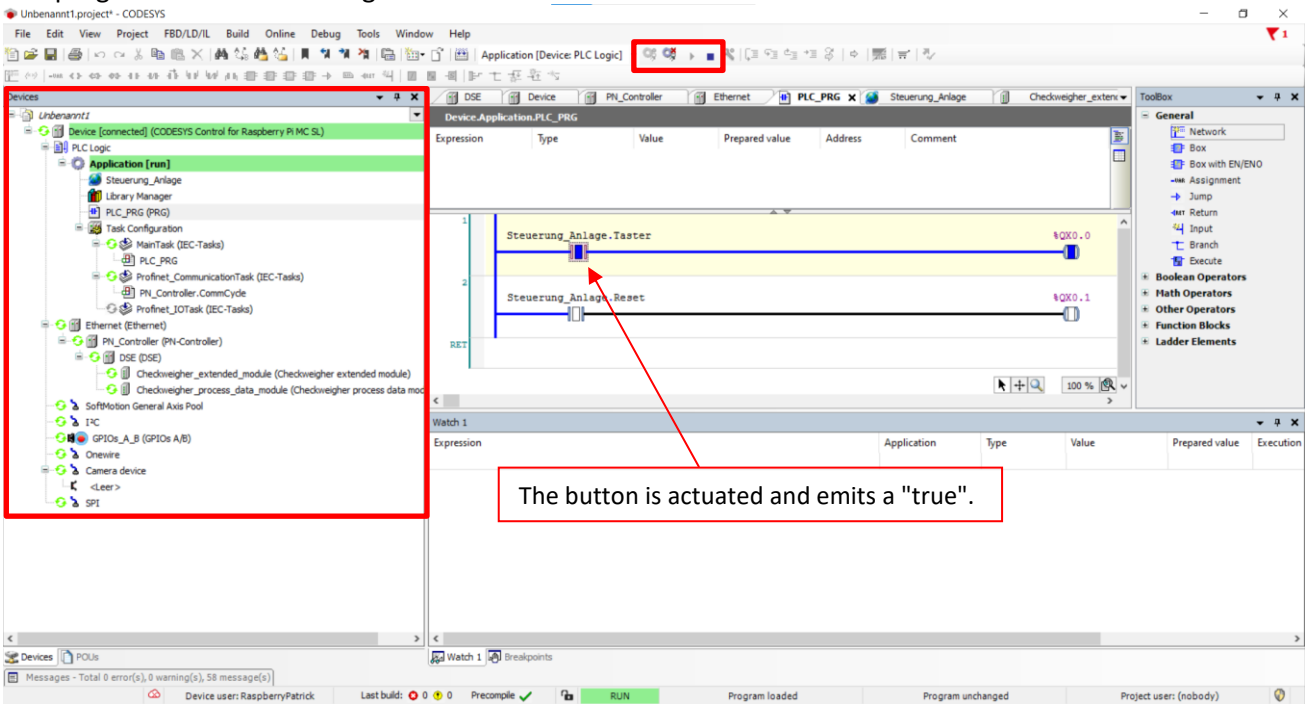


The subnet mask must match that of “eth0” again, and the IP address of the DSE must be entered. Also, the device name must be entered correctly.



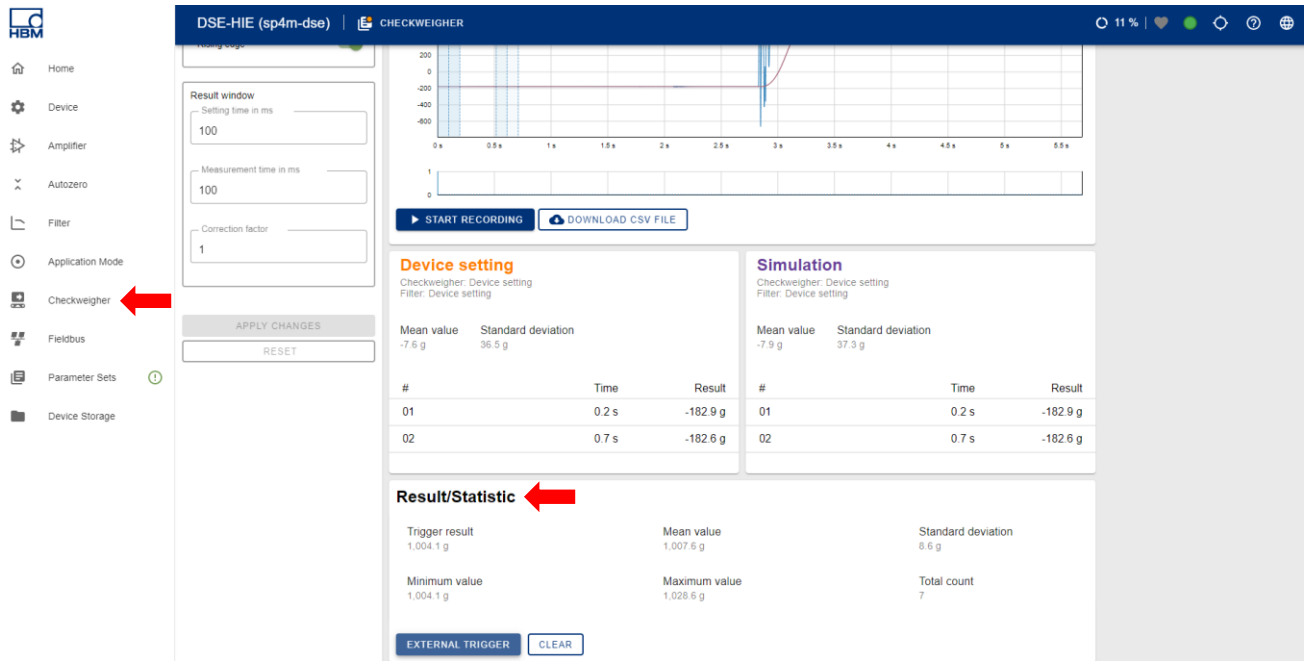
Now the project only needs to be started.

The program runs if there are green circles next to the inserted devices. See Picture:



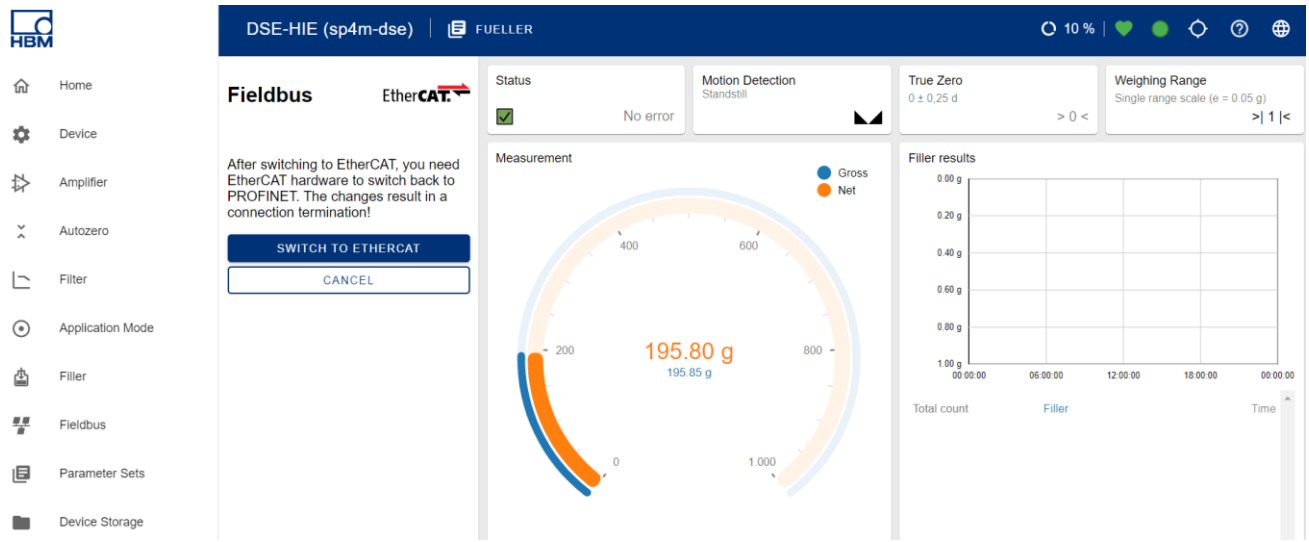
Note: When uploading, the code remains stored in the Raspberry. Even after switching off the PI´s and closing the Codesys software on the computer.

Now the DSE receives the signals of the button and counts up with each press (recognizes only button presses outside the measurement). This can be seen in the web interface of the DSE.



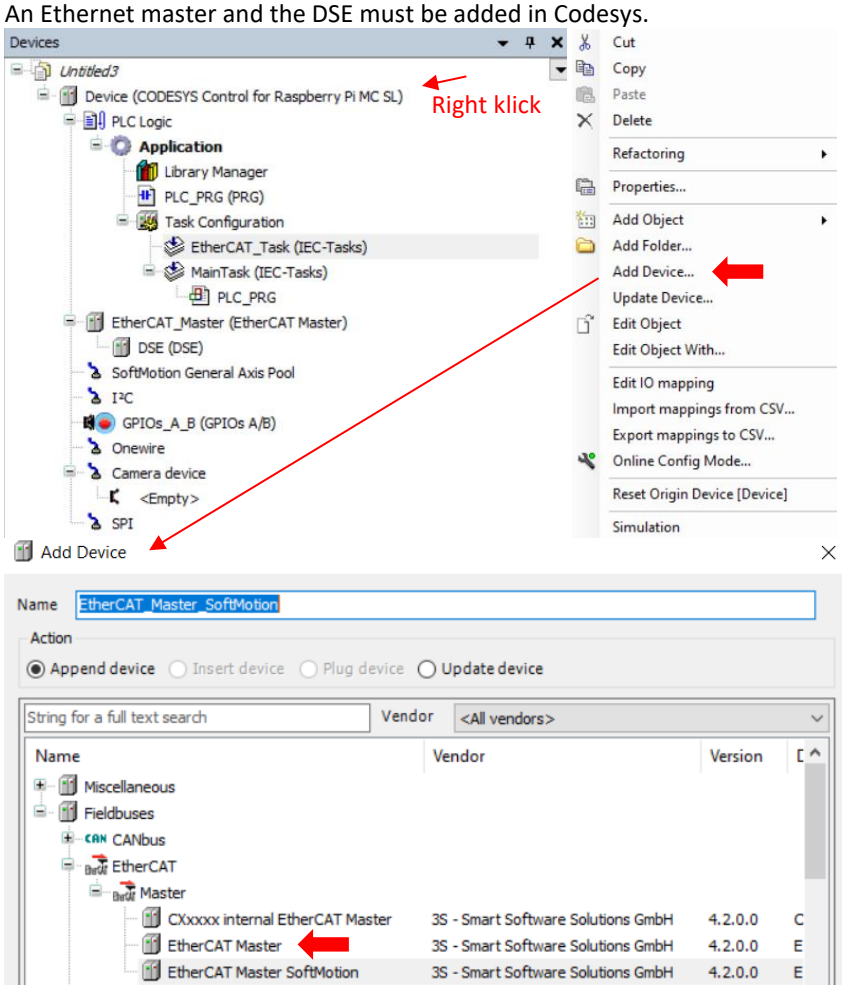
Alternative: Connection via EtherCAT

Optionally, the connection can also be established via EtherCAT. First of all, the DSE must be converted to EtherCAT:

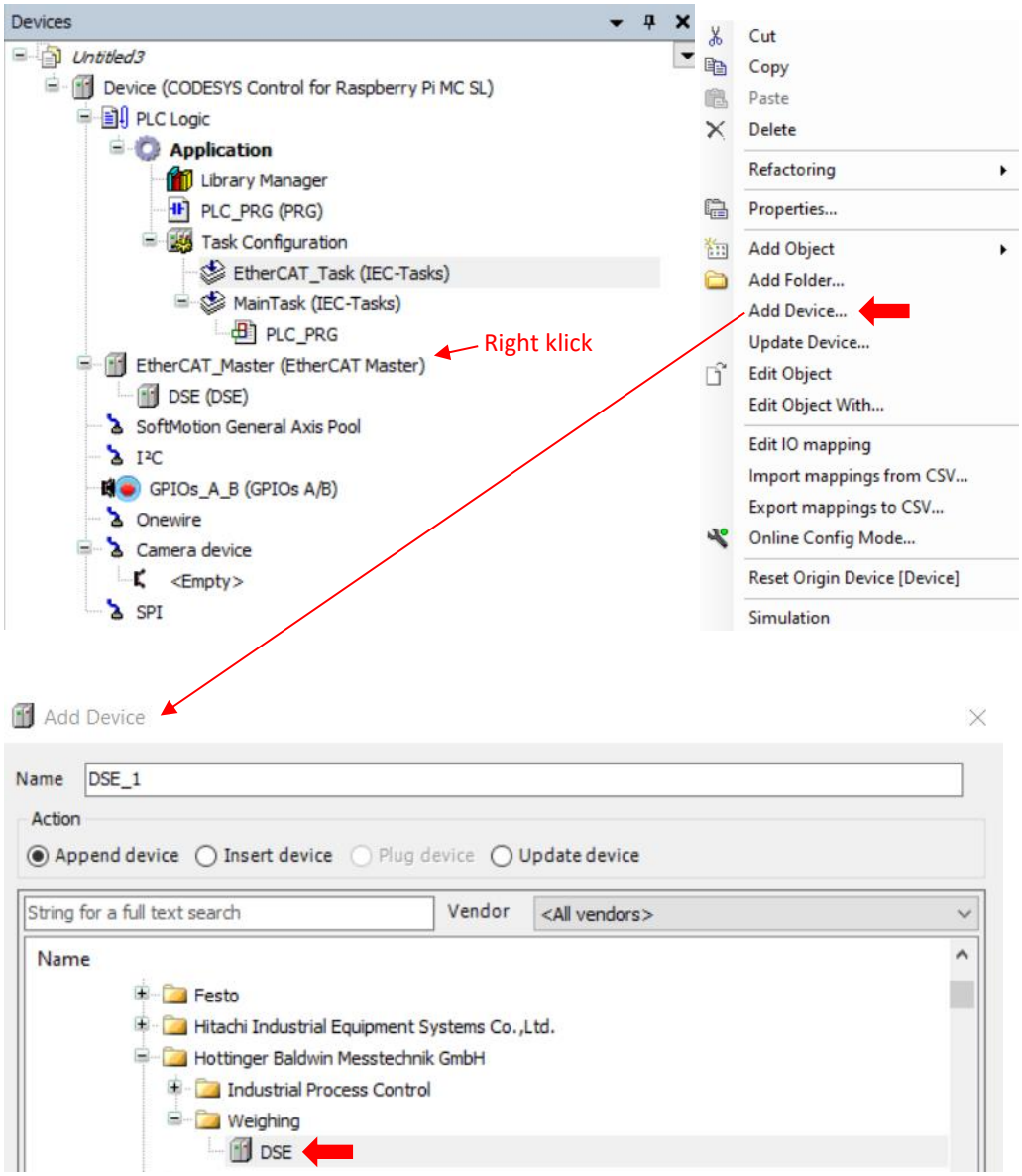


Now download the XML file of the DSE.

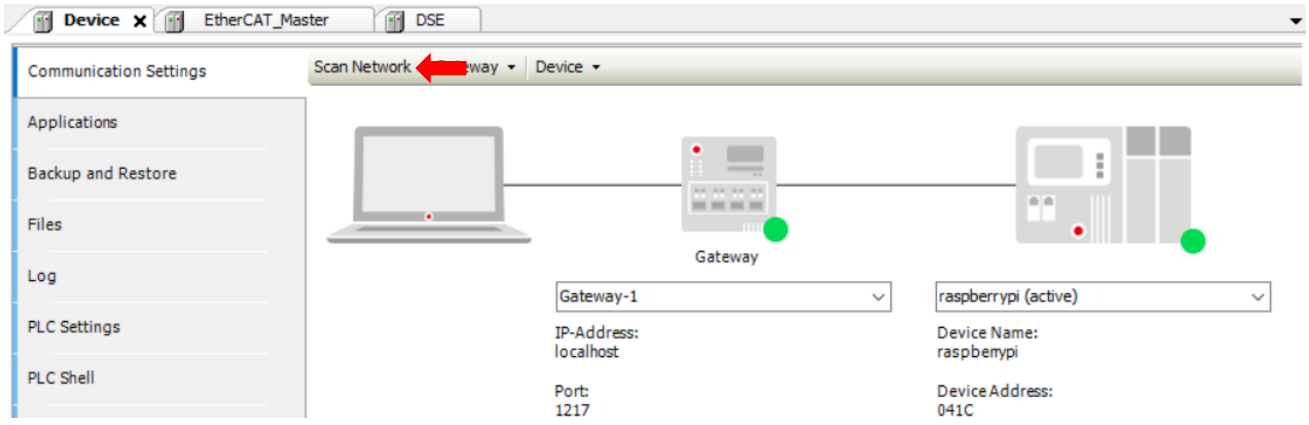
An Ethernet master and the DSE must be added in Codesys.



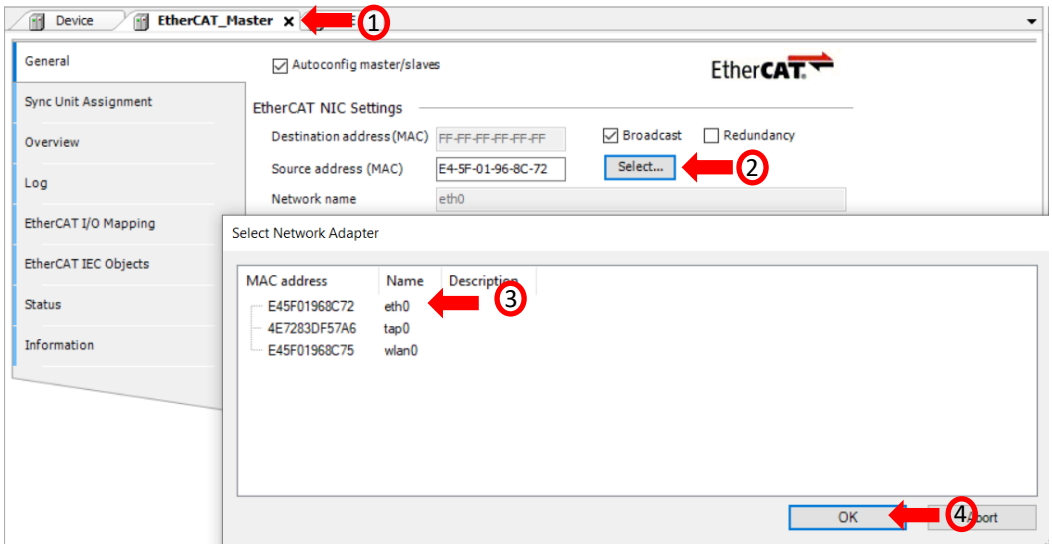
Now the Ethercat Master is added, now only the DSE has to be inserted.



Now the connection to the RaspberryPi can be established.



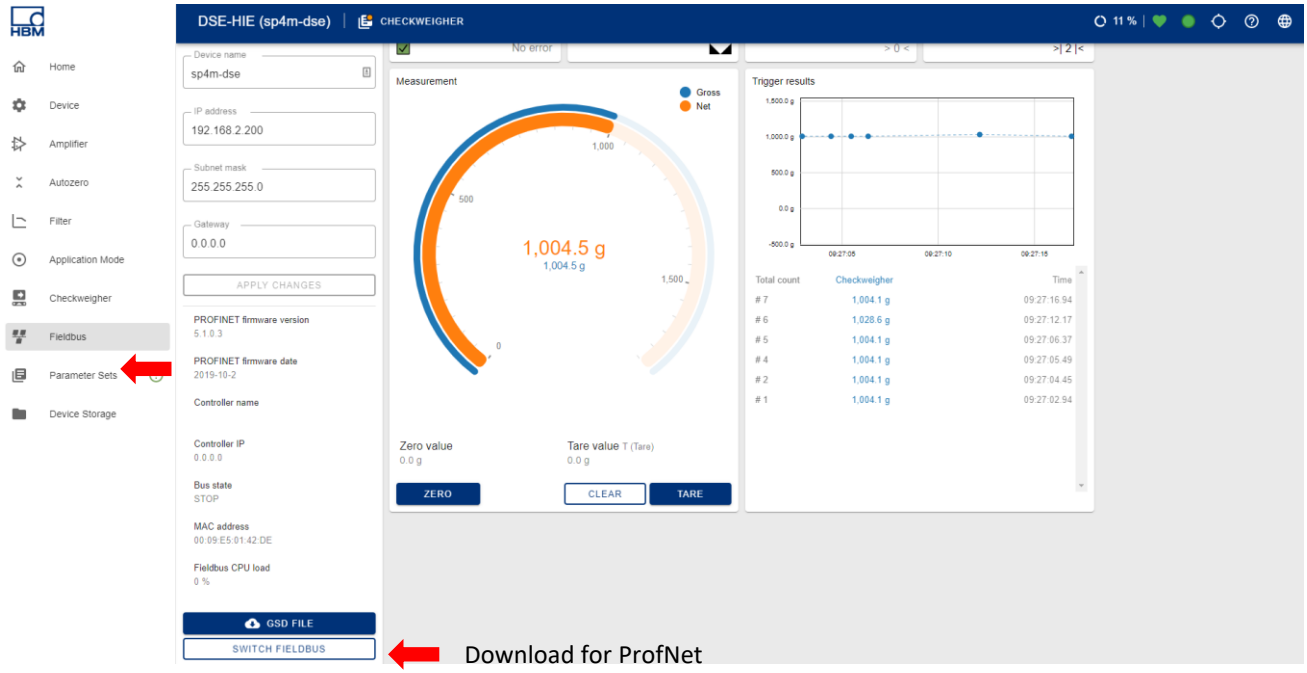
If this connection is established, the EtherCAT Master can be configured. After clicking on “select”. . . you have to enter the username and password of the RaspberryPis again, then the following picture appears:



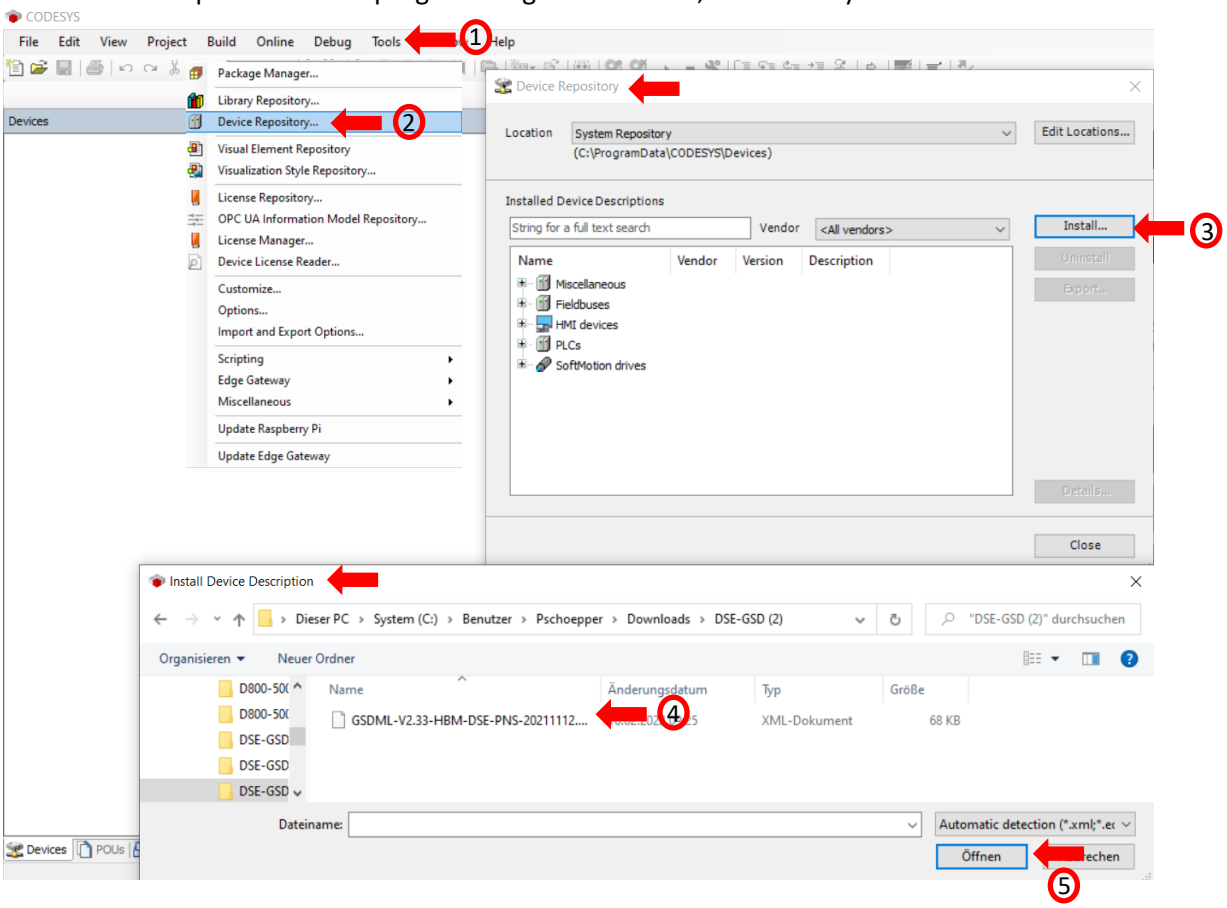
Now the project only needs to be started.

Additional information: the Codesys program code used

The Profinet GSDML file is a device description file. These include all device functionalities required for implementation in programming environments such as TIA Portal or Codesys. This is the only way to add the devices that are not stored in the standard catalog to the virtual setup and can also be controlled by the controllers.

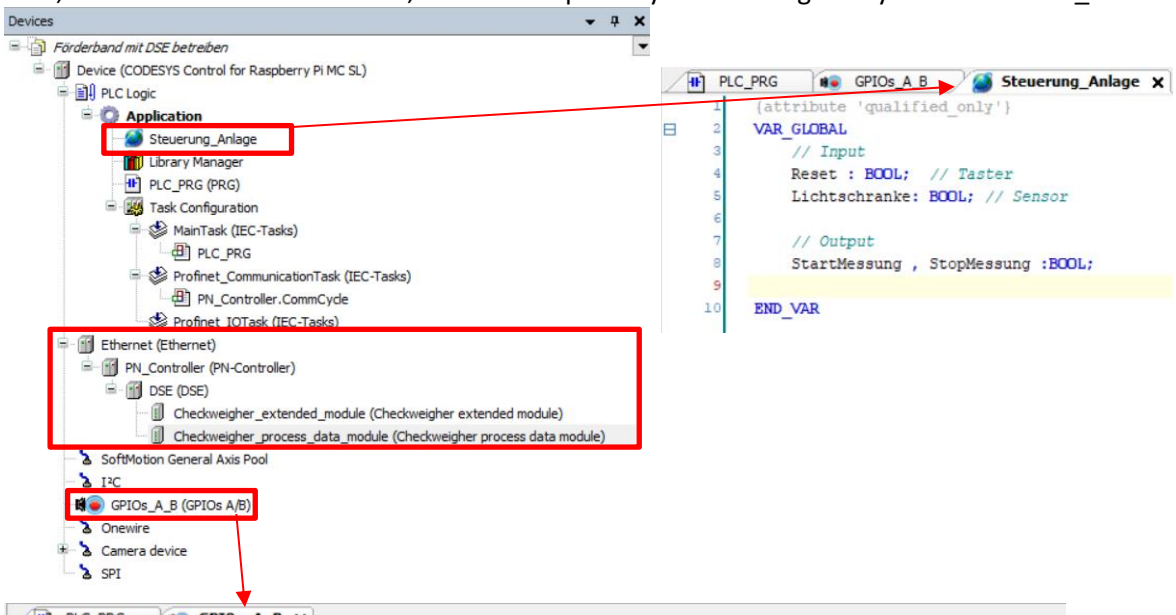


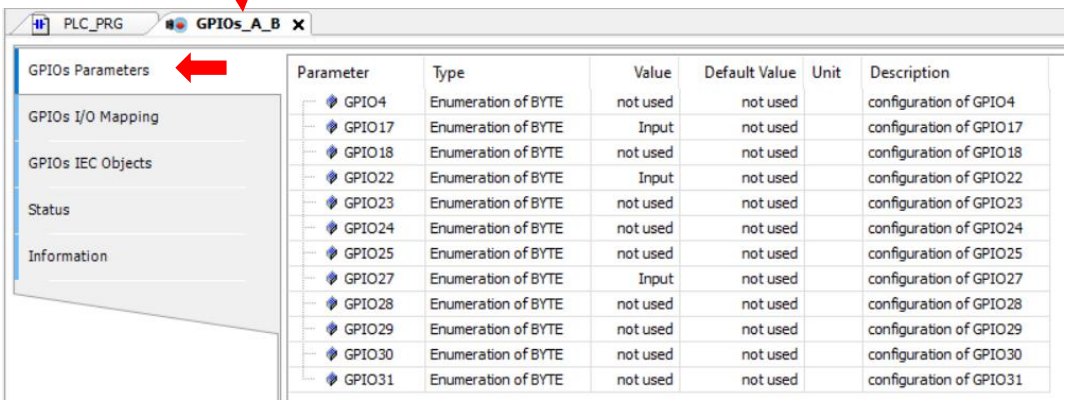
The GSD file is uploaded in the programming environment, with Codesys it works as follows:

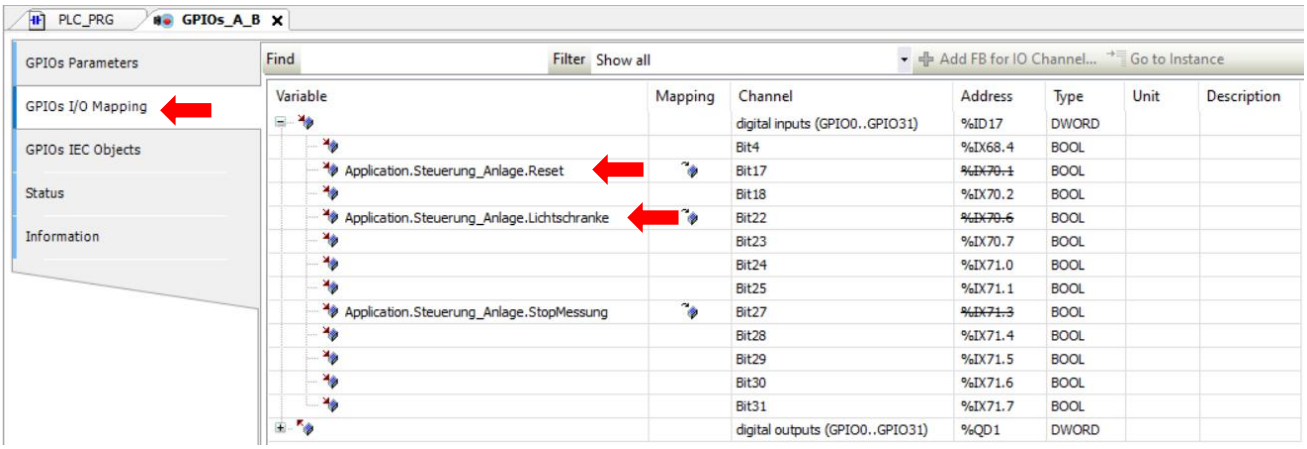


After pressing “Open”, the device is added to the “Repository. ”
To enable the DSE to communicate with the Codesys PLC via ProfiNet, the “PN_Controller” is inserted. In addition, the Checkweigher modules shown below are required from the already loaded GSDML file.

First, variables have to be created, in this example they are stored globally under “Control_Attachment”:

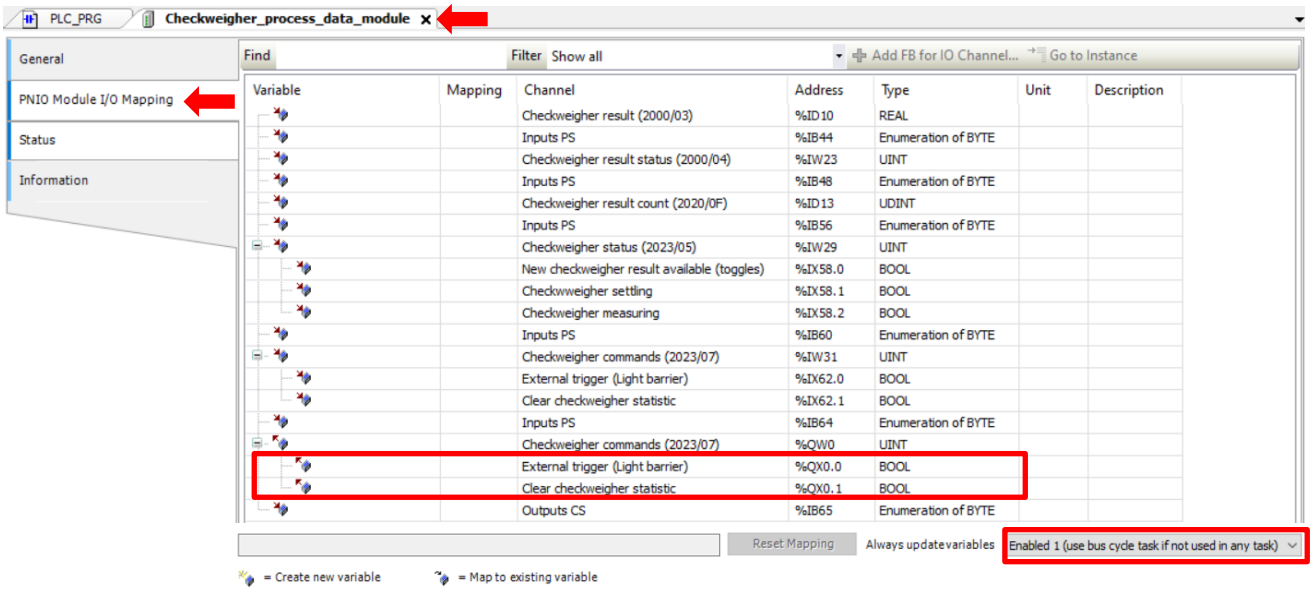




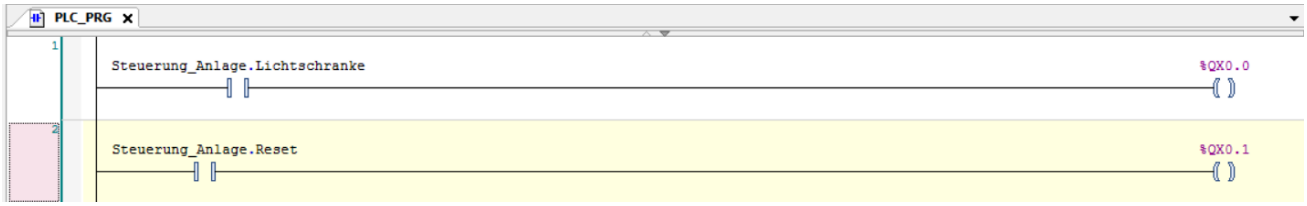


External Signal process

The address of the “external signal” is %QX0. 0 and for the reset of the statistics is %QX0. 1.

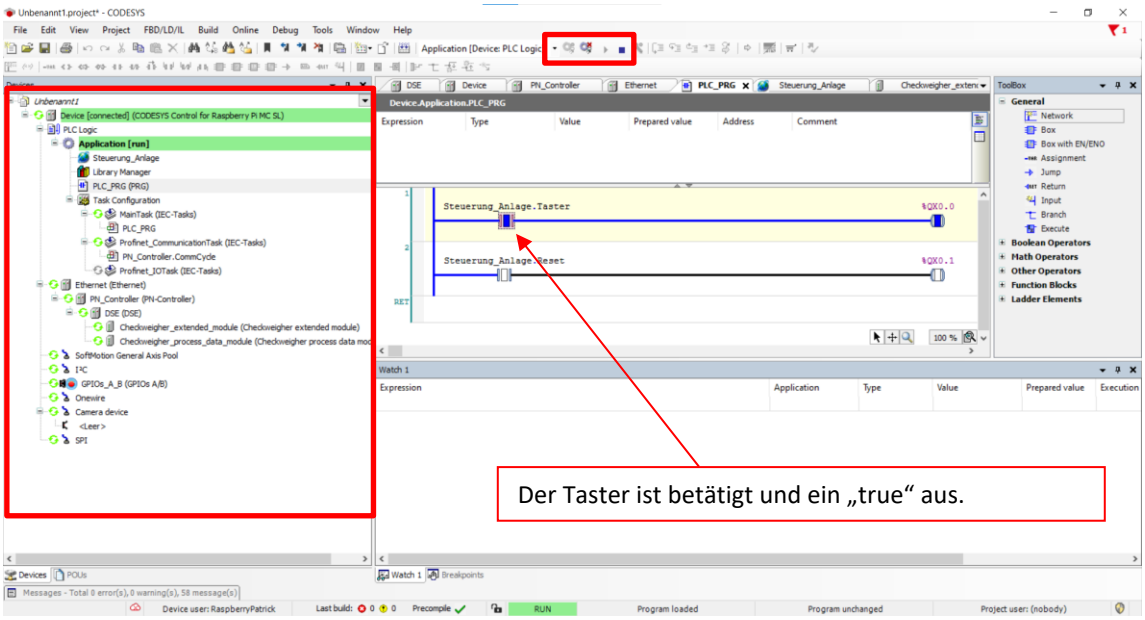


These must now be controlled by our inputs.



Now the program only needs to be started.

You can see that the program works if there are green circles next to the inserted devices. See picture:



Now the DSE receives the signals of the probe and counts up with each push (recognizes only probe signals outside the measurement). This can also be seen in the DSE web interface.

Note: When uploading, the code remains stored in the Raspberry. Even after switching off the Pis and closing the Codesys software on the computer.

Legal note

This example is for illustrative purposes only. It is not subject to any warranties or liability claims.