

TECH NOTE :: DSE_Checkweiger mit SPS (R-PI 4B)

Version: 2022-04-26

Autor: Michael Guckes, Patrick Schöpfer

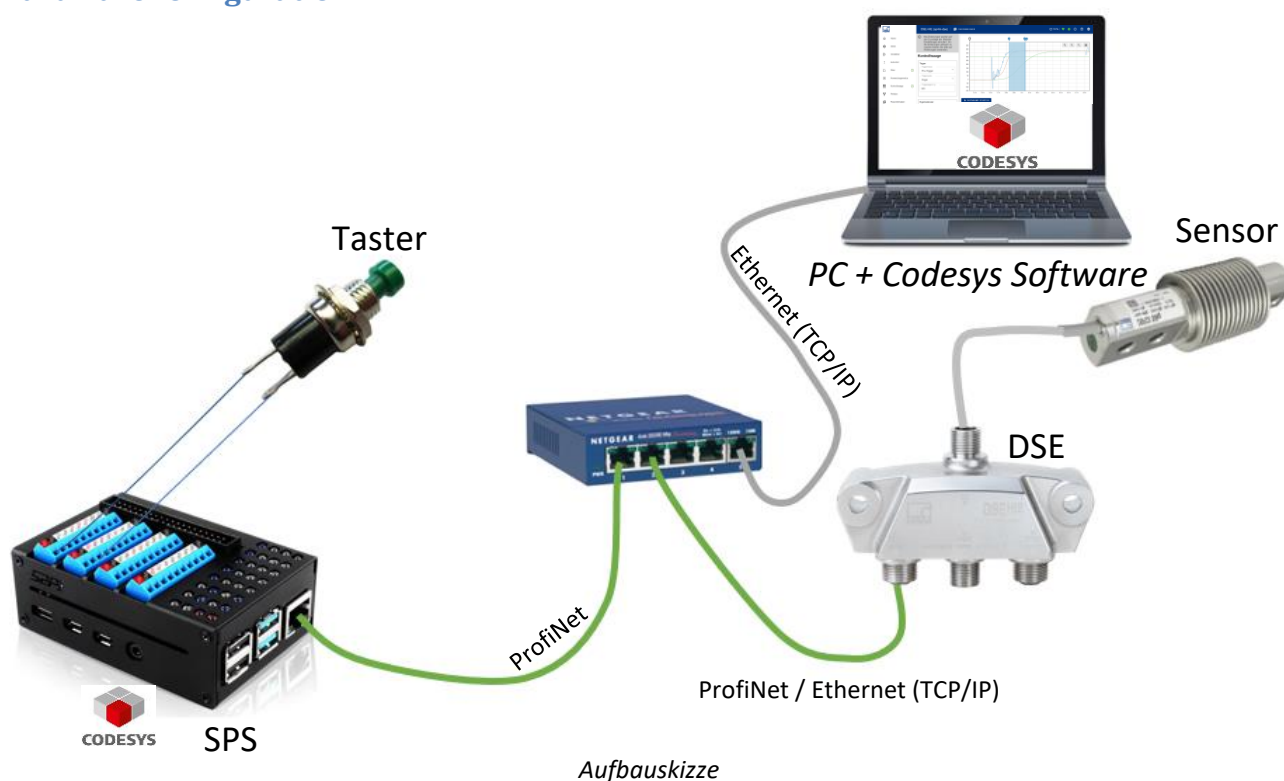
Status: HBM: Public

Kurzbeschreibung

Dies ist eine Schnellstart-Anleitung zum Aufbau einer Kontrollwaage (Checkweiger) mit dem DSE (ab FW 2.0). Die Kontrollwaage kann über zwei Wege angesteuert werden, zum einen über einen Pegel zum anderen über ein externes Signal (in diesem Beispiel einem Taster). Die DSE verfügt über keine hardwaremäßigen DIO's, so daß die Triggersignale von einer SPS erfasst und an die DSE übertragen werden, hier ein Raspberry Pi mit Codesys als SPS, über PROFINET und EtherCAT.

Dazu wird später für dieses diese einmalig eine Codesys Runtime-Lizenz und ein fertiges Codesys-Projekt „DSE-Checkweiger-R-PI.projectarchive“ in den R-PI geladen. Verwendete Codesys Version ist 3.5.17.10 Patch 3 oder höher.

Hardwarekonfiguration

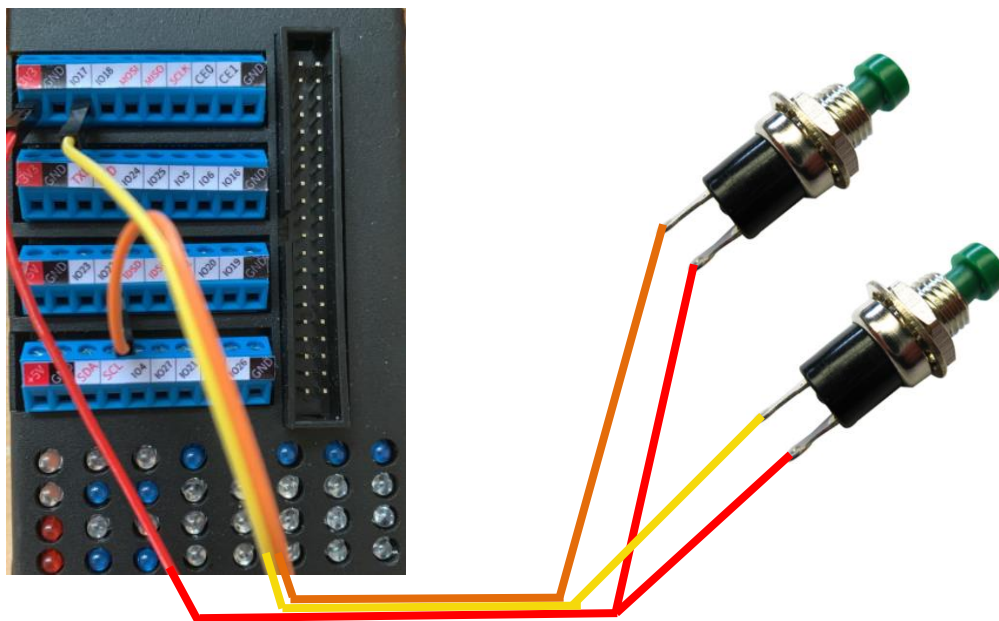


Aufbauskizze

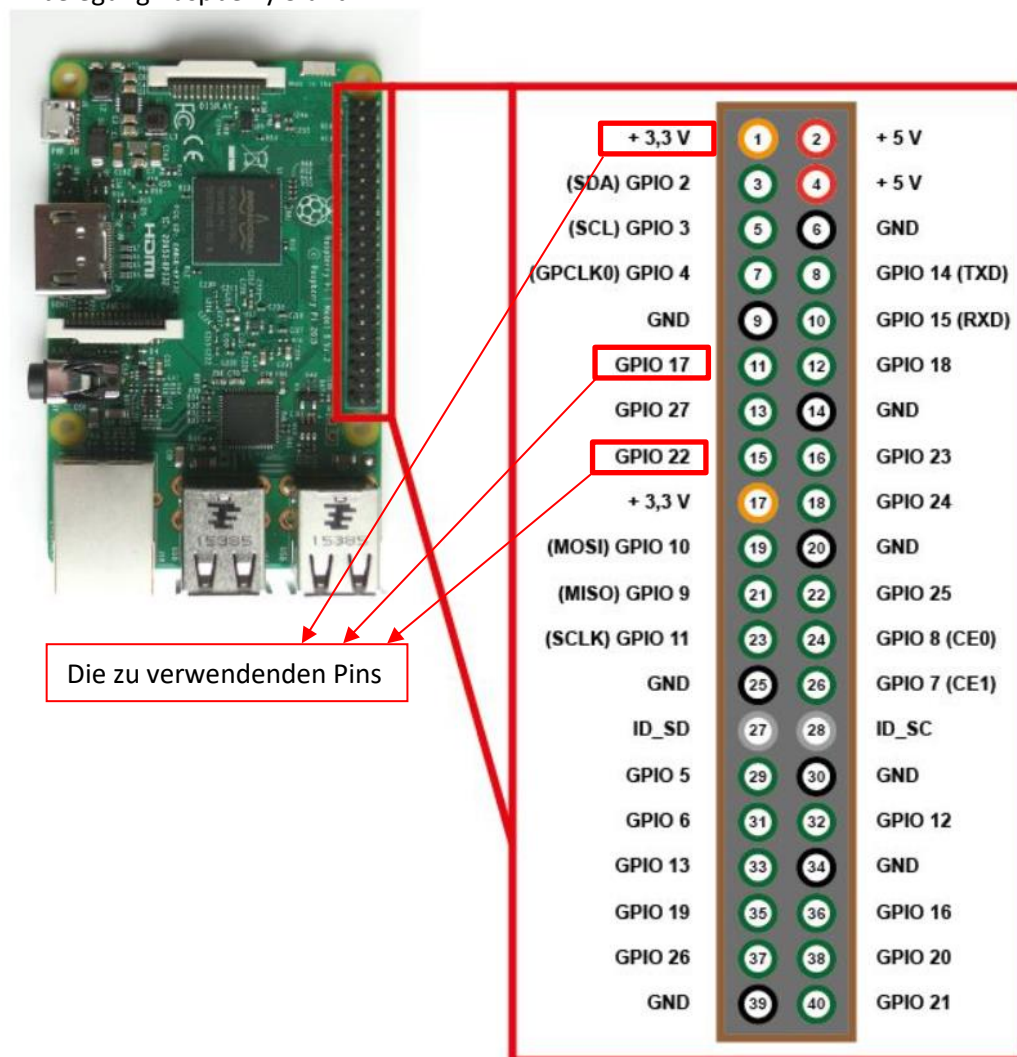
Erforderliche Komponenten

- 1x DSE System (inkl. Netzteil und Ethernetkabel)
- 1x kostenlose Software Codesys
- 1x Ethernet switch
- 1x SPS, R-PI Model 4B (Codesys Controle for Raspberry Pi V 4.4.0.0)
- 2x Taster
- 1x Wägezelle

Hardwarekonfiguration der GPIO's am PI

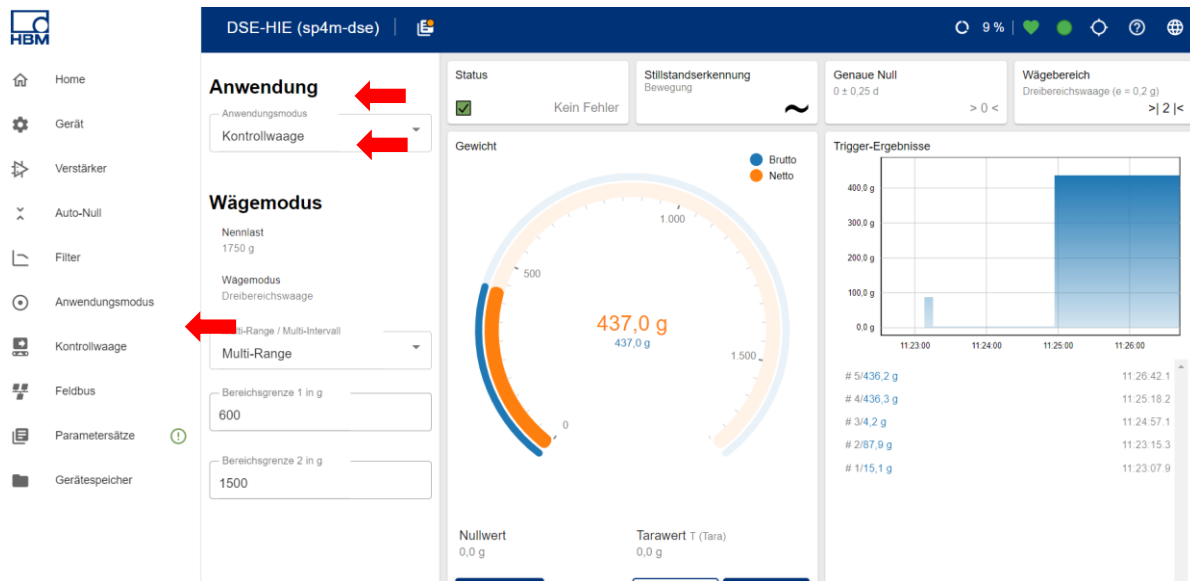


Pinbelegung Raspberry 3 und 4

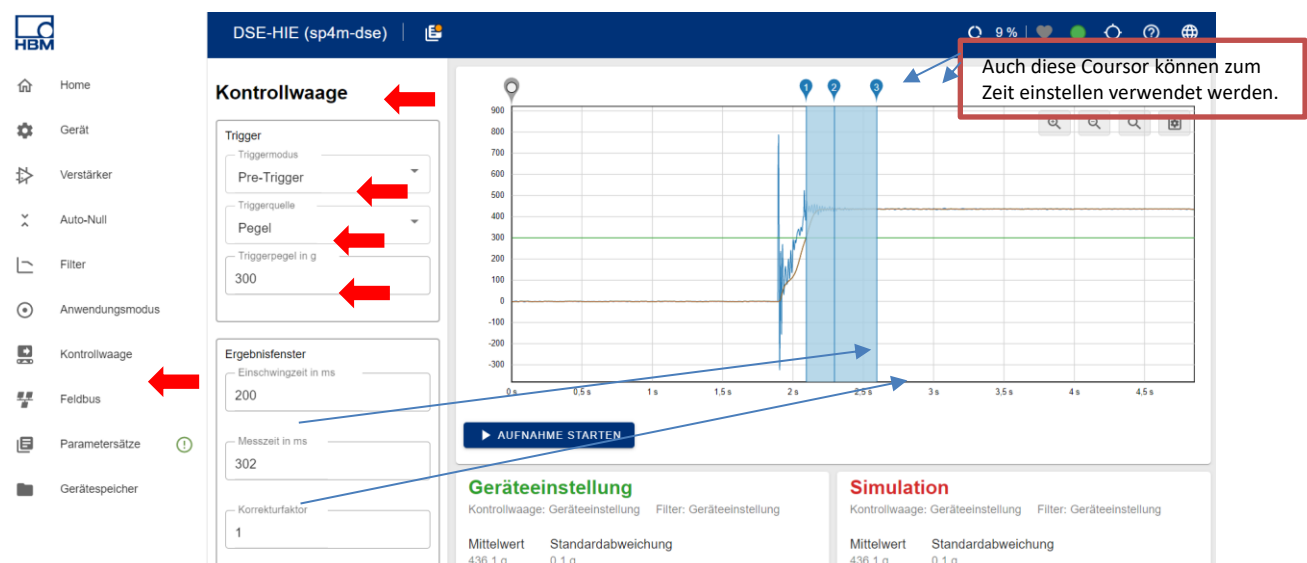


Pegelsteuerung

Bei der Pegelsteuerung wird der Check-Weighing Vorgang gestartet (getriggert), wenn die Last einen Pegel über und wieder unterschreiten wird.



Im Menü Applikation > Kontrollwaage auswählen. Unter „Triggerquelle“ muss „Pegel“ eingestellt werden. Die Höhe des Pegels ist in dem folgenden Feld einzustellen. Zum Testen empfiehlt sich ein niedriger Pegel, der auf jeden Fall bei der erwarteten Belastung überschritten wird. Der Pegel ist in dem Diagramm grün eingezeichnet. Im Ergebnissenfenster lässt sich zunächst die Einschwingzeit einstellen. Diese beginnt mit dem Überschreiten des Simulationsergebnisses (braun) und sollte nach dem Einschwingen enden, diese Zeit muss durch Testläufe ermittelt werden. Als nächstes kann die Messzeit erstellt werden, auch diese sollte möglichst gering sein aber noch so lang, dass sich ein korrektes Ergebnis einstellt.



In dem Fenster „**Ergebnisse/Statistik**“ werden die verschiedenen Werte, die mit dem System erfasst werden können, visualisieren.

Ergebnisse/Statistik

Trigger-Ergebnis
436,2 g

Mittelwert
195,9 g

Standardabweichung
198,3 g

Min. Wert
4,2 g

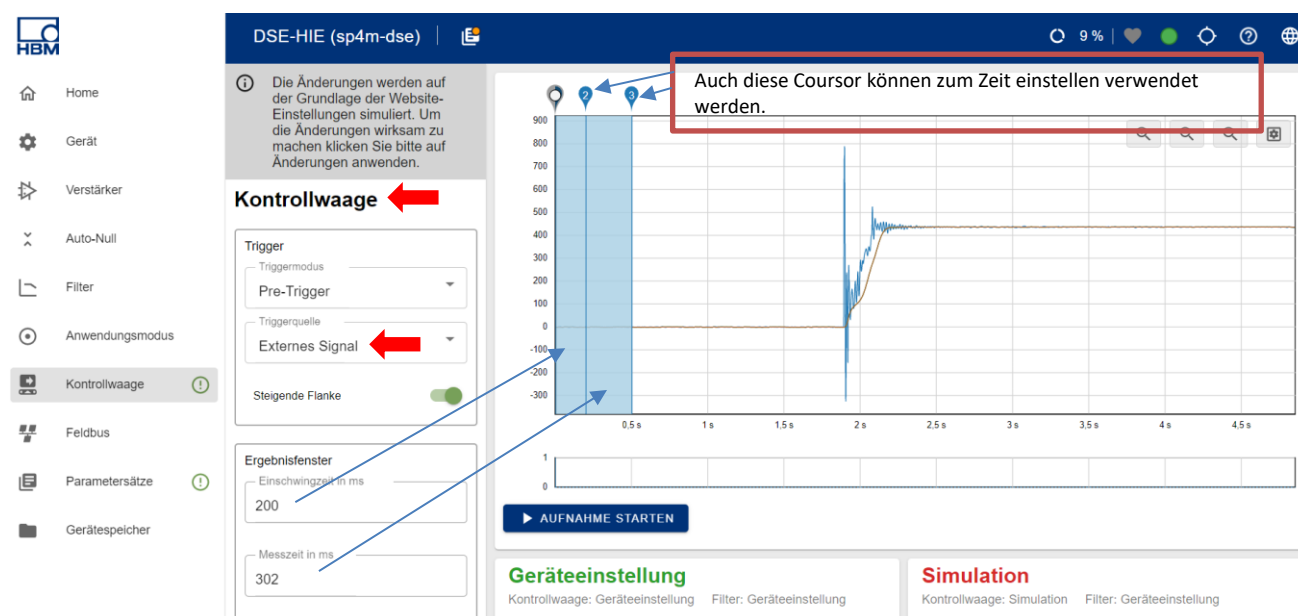
Max. Wert
436,3 g

Gesamtanzahl
5

EXTERNER TRIGGER

ZURÜCKSETZEN

Steuerung über externes Signal (Taster oder Lichtschranke)



Bei der Verwendung eines externen Signals als Trigger, ist der Start des Messvorgangs losgelöst vom Verlauf des Messsignals. Das externe Signal kann auf viele Wege erzeugt werden, z.B. durch eine Lichtschranke oder einen Taster, es handelt sich aber immer um den Datentyp Bool, das bedeutet „true or false“ oder „1 oder 0“. In dem Beispiel oben kann man sehen, dass der Messvorgang zusammen mit der Aufzeichnung gestartet wurde, auch wenn dort noch keine Veränderung des Verlaufs stattgefunden hat.

Der Check-Weighing Vorgang kann mit einem Pre-Trigger vor Belastung der Wägezelle oder mit einem Post-Trigger nach Belastung der Wägezelle ausgelöst werden.

Codesys installieren (einmalig nötig)

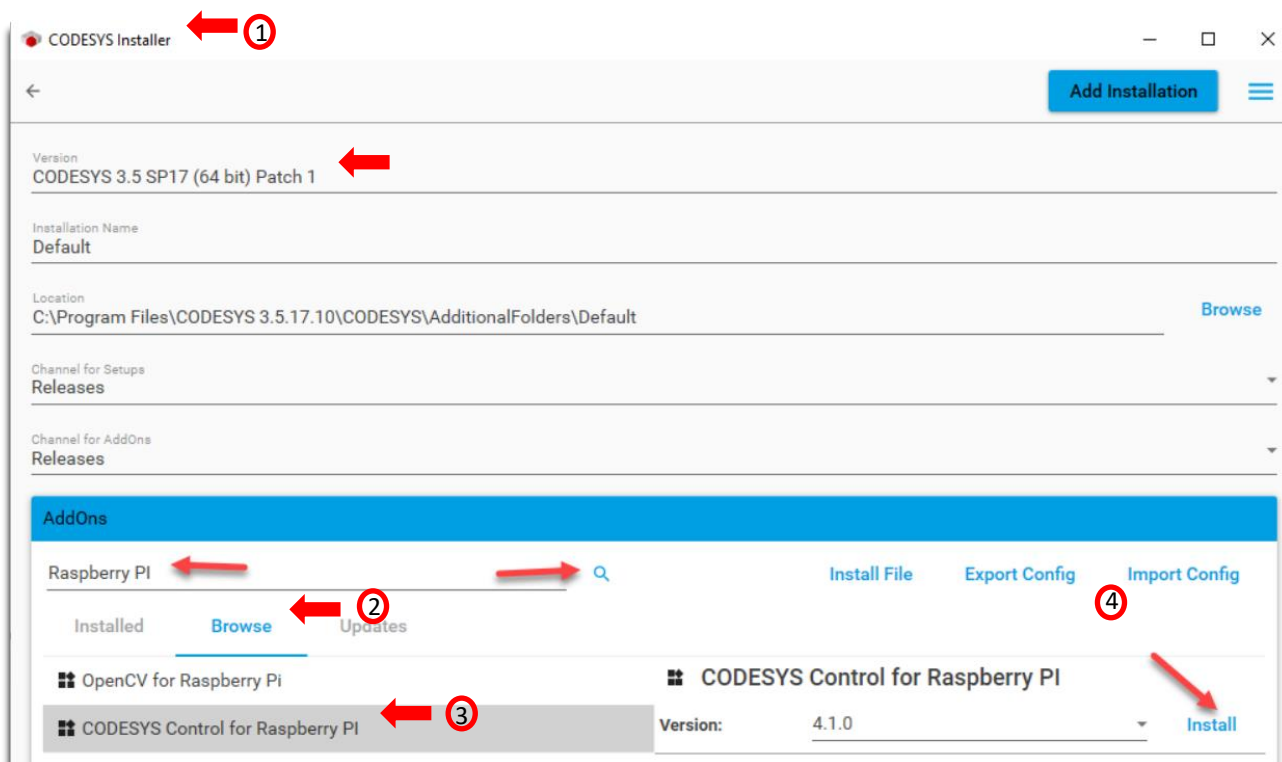
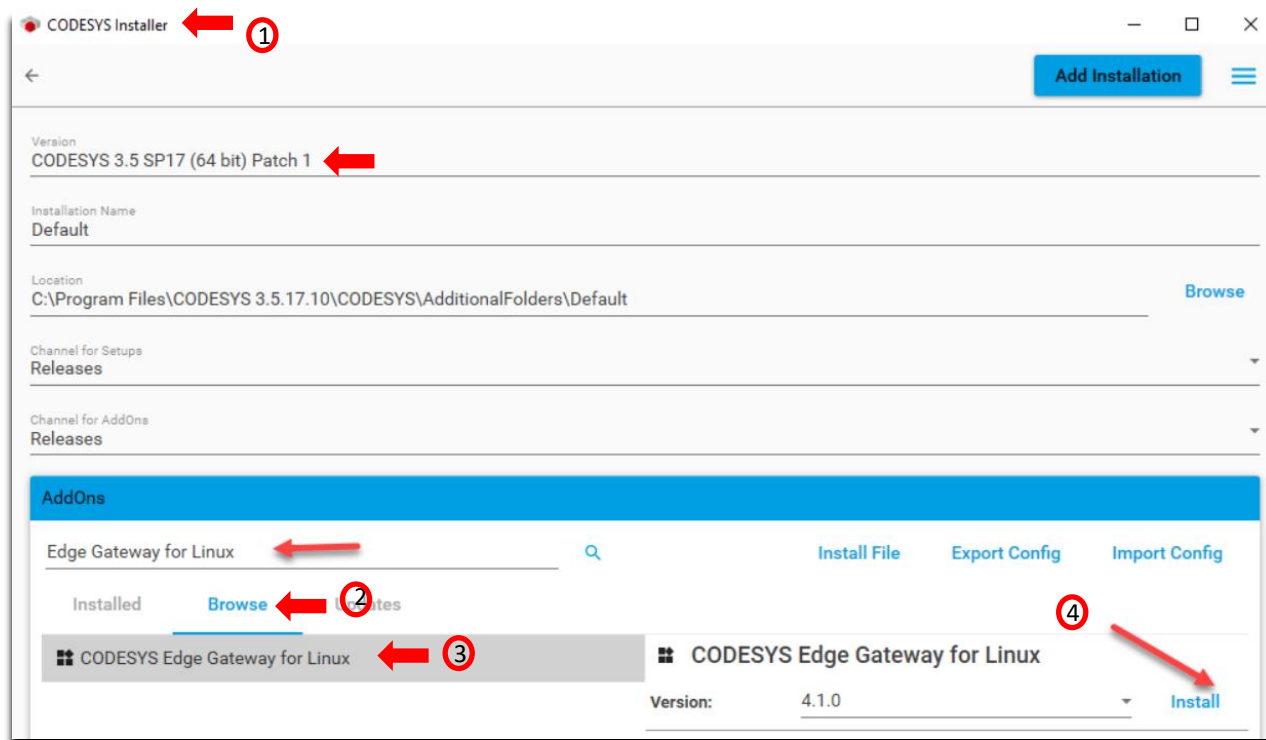
Die kostenlose Codesys Software kann hier heruntergeladen werden:

<https://store.codesys.com/de/codesys.html>

Alle weiteren Installationen über den **Codesys Installer**.

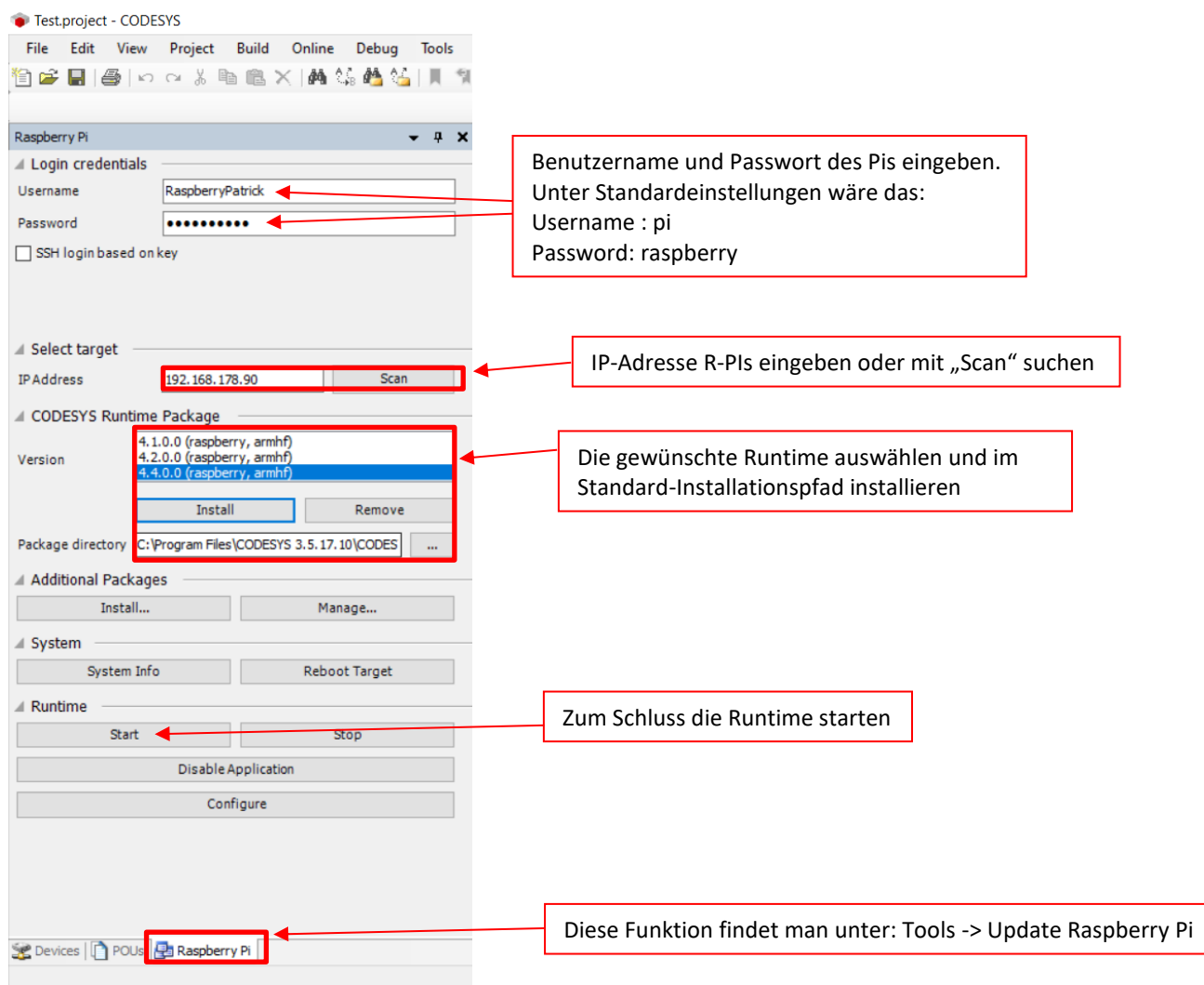
Öffnen Sie den Codesys Installer:

Zuerst Installieren Sie das „Edge Gateway for Linux“ dann das „Codesys Control for Raspberry Pi“.



Installation der Codesys-Runtime auf dem Raspberry Pi (einmalig nötig)

Nach der Installation starten Sie Codesys auf ihrem Computer mittels Klick auf den Icon. Nachdem Codesys gestartet ist, muss die Runtime auf dem Pi geladen werden.

Benutzername und Passwort des Pis eingeben. Unter Standardeinstellungen wäre das:
Username : pi
Password: raspberry

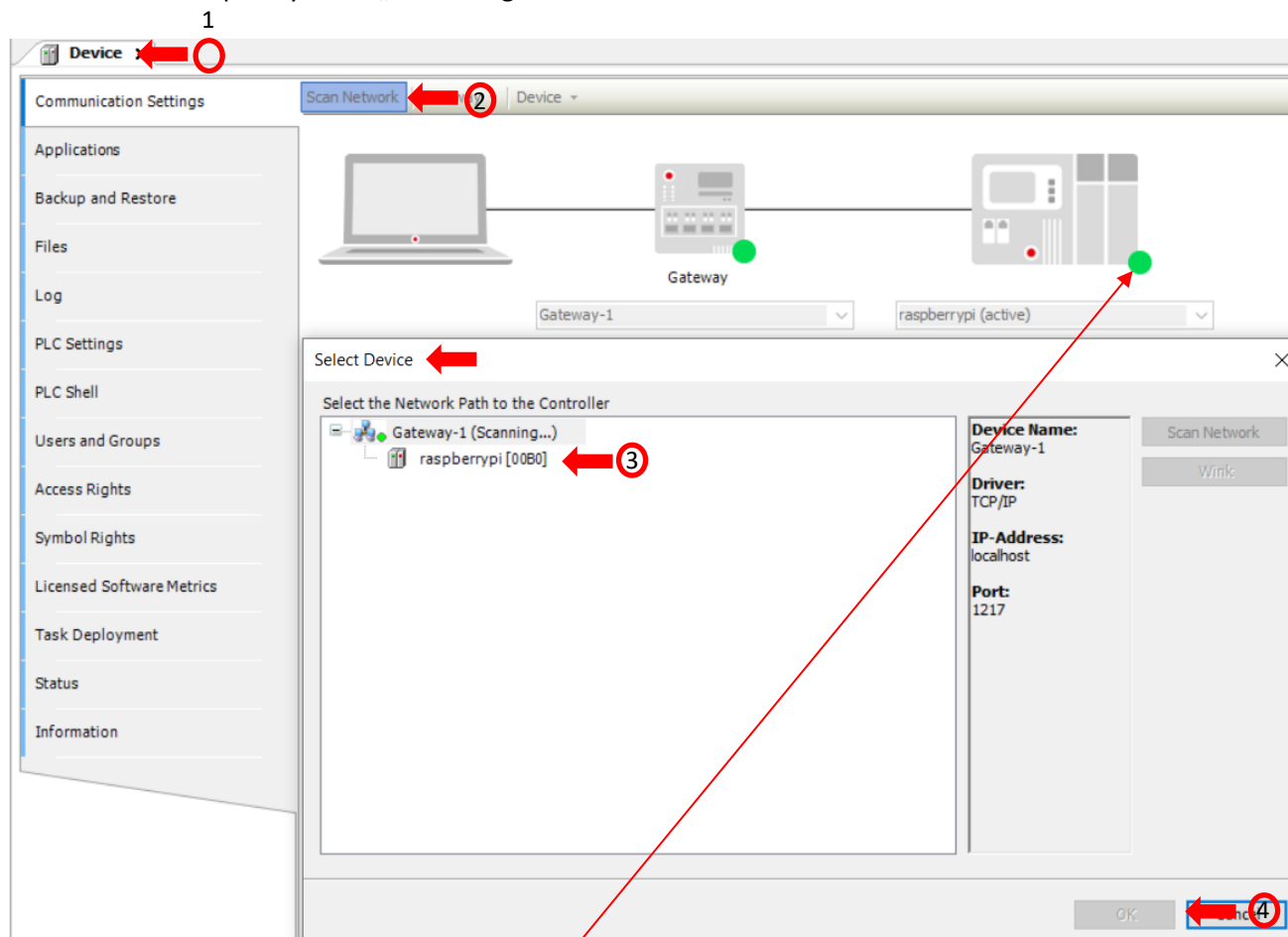
IP-Adresse R-Pis eingeben oder mit „Scan“ suchen

Die gewünschte Runtime auswählen und im Standard-Installationspfad installieren

Zum Schluss die Runtime starten

Diese Funktion findet man unter: Tools -> Update Raspberry Pi

Jetzt kann der Raspberry unter „Devices“ gefunden werden:

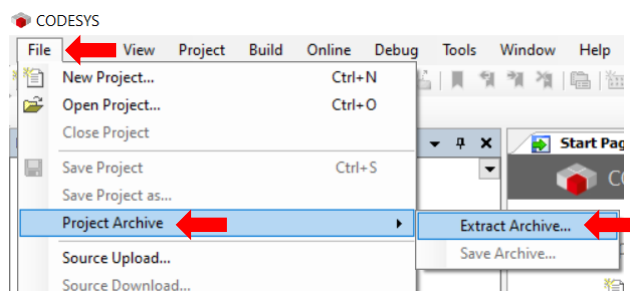


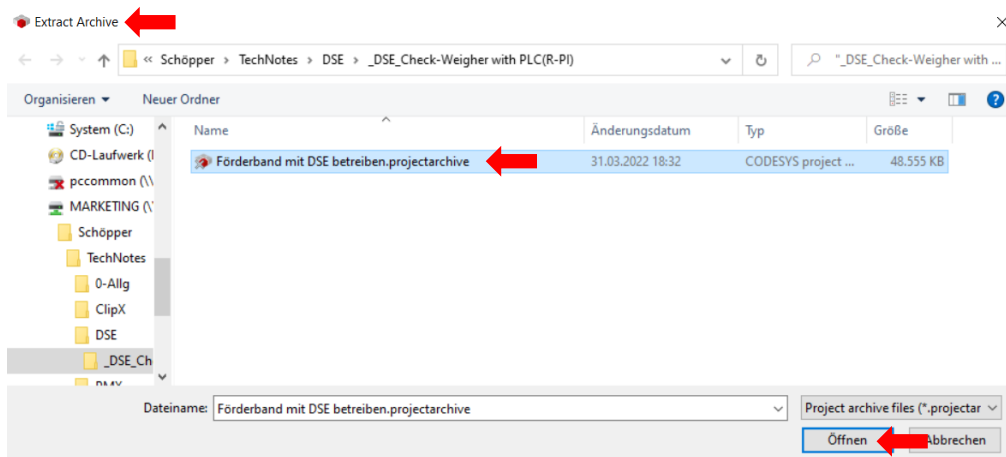
Nach der Betätigung von „OK“ muss man sich noch am Pi einloggen. Das funktioniert genauso wie bei der Installation der Runtime. Wenn nach diesem Vorgang die Led grün leuchtet, hat der Verbindungsaufbau funktioniert.

Projekt importieren (einmalig nötig)

Das Projekt enthält alle notwendigen Bibliotheken auch die Gerätebeschreibungsdatei (GSDML) der DSE und muss nur einmal importiert werden.

Öffnen Sie Codesys auf ihrem Rechner.

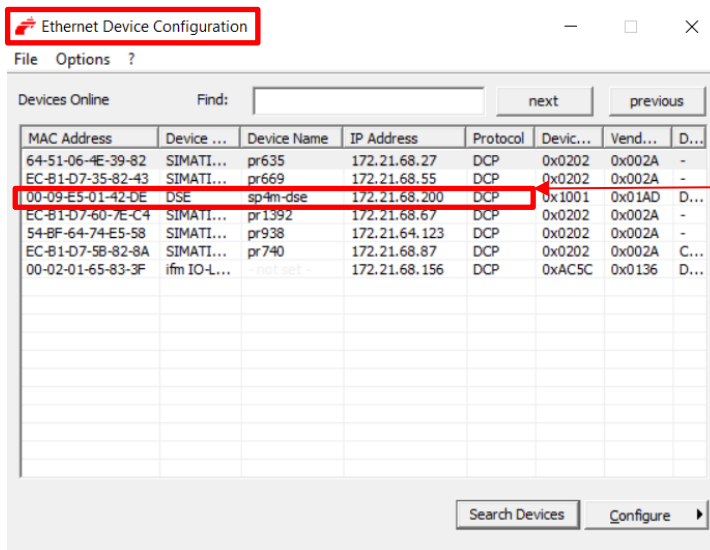




Nach dem Klick auf Öffnen wird das Projekt mit allen Bibliotheken und Geräten in Codesys importiert.

Netzwerk Konfigurationen

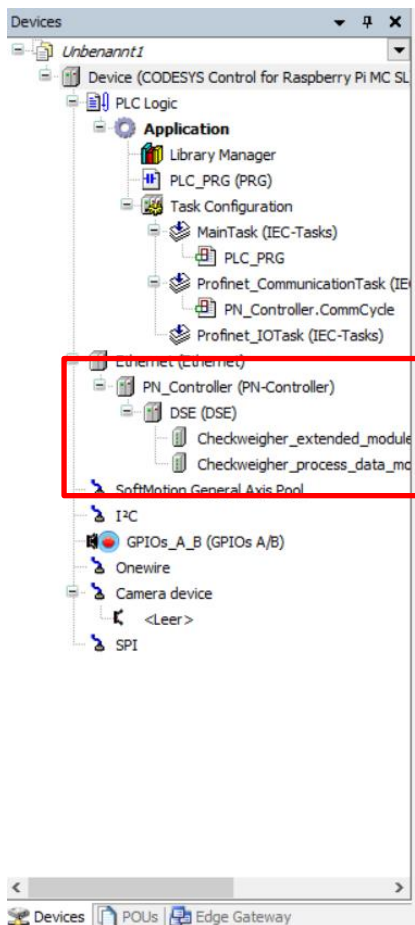
Hinweis: Die IP-Adresse sowie der Gerätenamen des DSE kann mittels des Tools „Ethernet-Geräte-Setup“ konfiguriert werden.



Rechtsklick -> Set IP/Device Name zum konfigurieren

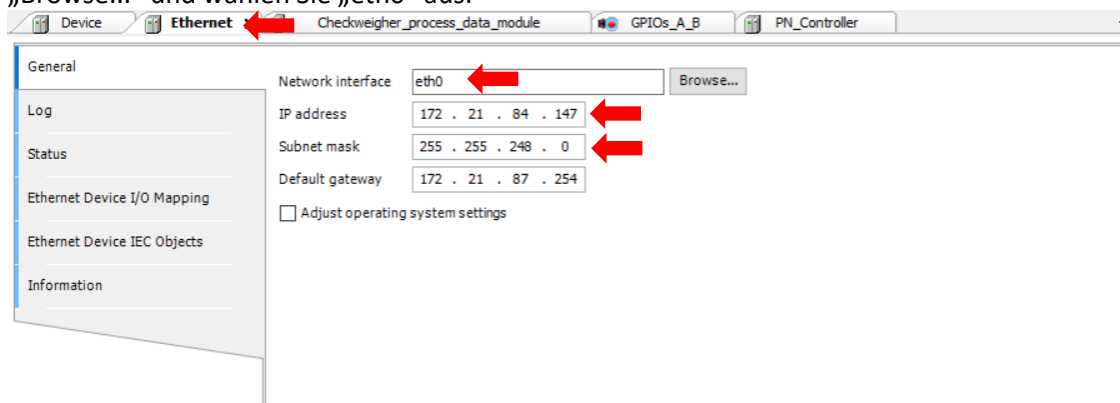
Hinweis: Die IP muss zu der IP-Adresse des R-Pis passen, d.h. ersten 3 Blöcke müssen gleich sein, der letzte unterschiedlich.

Hier kann man die verschiedenen Adapter sehen, um den DSE mit dem Codesys Controle zu verbinden. Diese müssen nun noch für das Netzwerk konfiguriert werden.

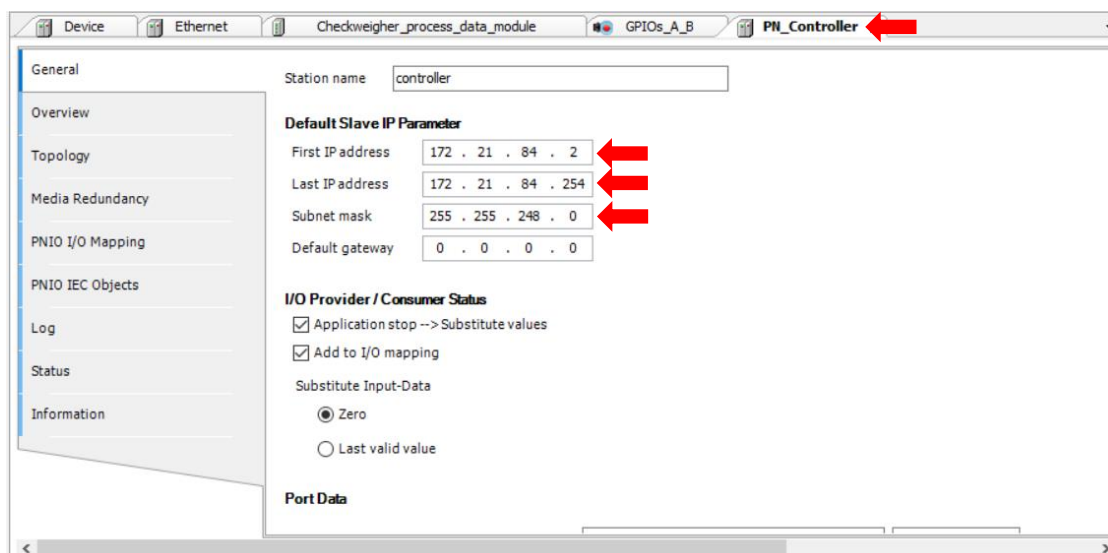


Da das Codesys Hauptprogramm mit der Applikation auf dem Raspberry kommunizieren muss, müssen diese mittels PROFINET miteinander verbunden werden. Dies wird im Folgenden erläutert.

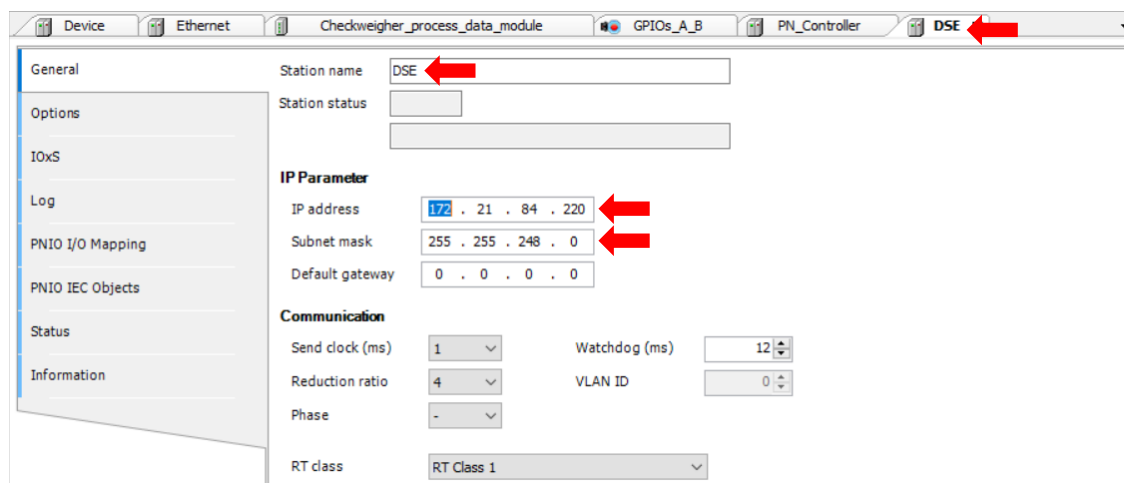
Zunächst muss die Netzwerk-Adresse des Ethernet Adapter angepasst werden. Dafür klicken Sie auf „Browse...“ und wählen Sie „eth0“ aus.



Danach muss auch die IP-Adresse des „PN_controller angepasst werden, dafür ist wichtig, dass die ersten 3 Spalten identisch zu den ersten 3 Spalten von „eth0“ sind. Auch die Subnetzmaske muss mit der von „eth0“ übereinstimmen.



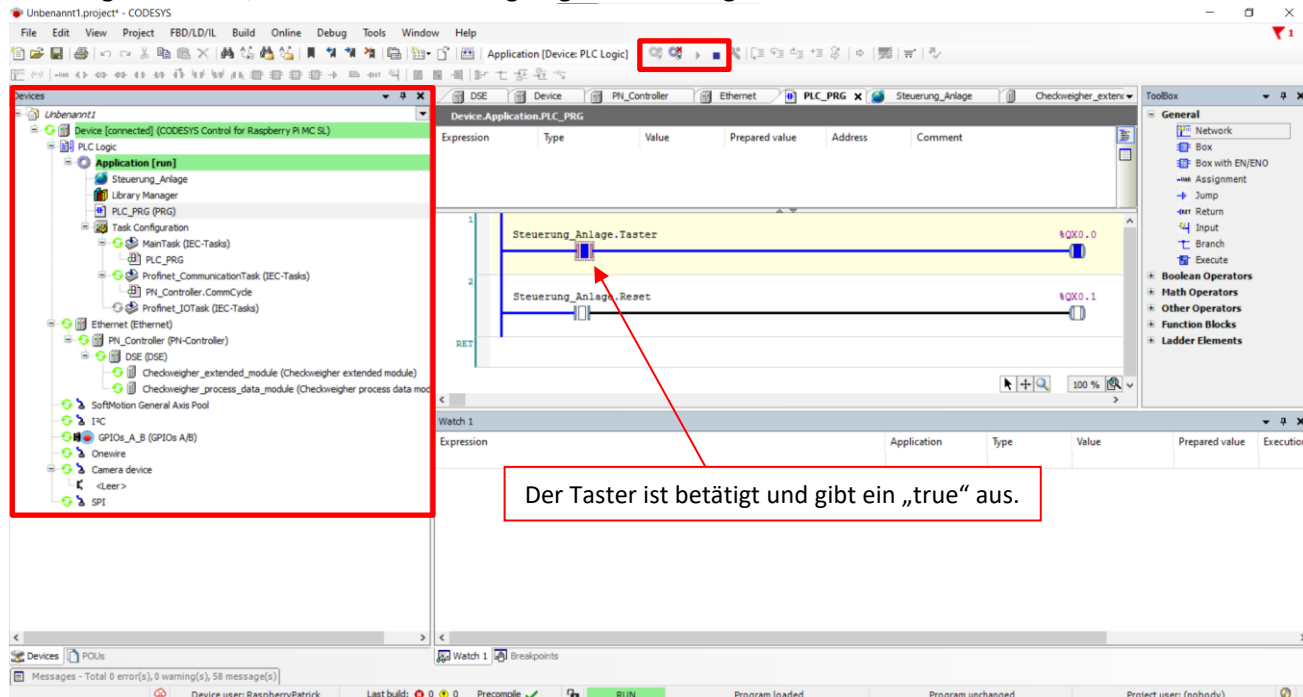
Die Subnetzmaske muss wieder mit der von „eth0“ übereinstimmen, außerdem muss die IP- Adresse des DSE eingetragen werden. Auch muss der Gerätenamen korrekt eingetragen werden.



Nun muss das Projekt nur noch gestartet werden.

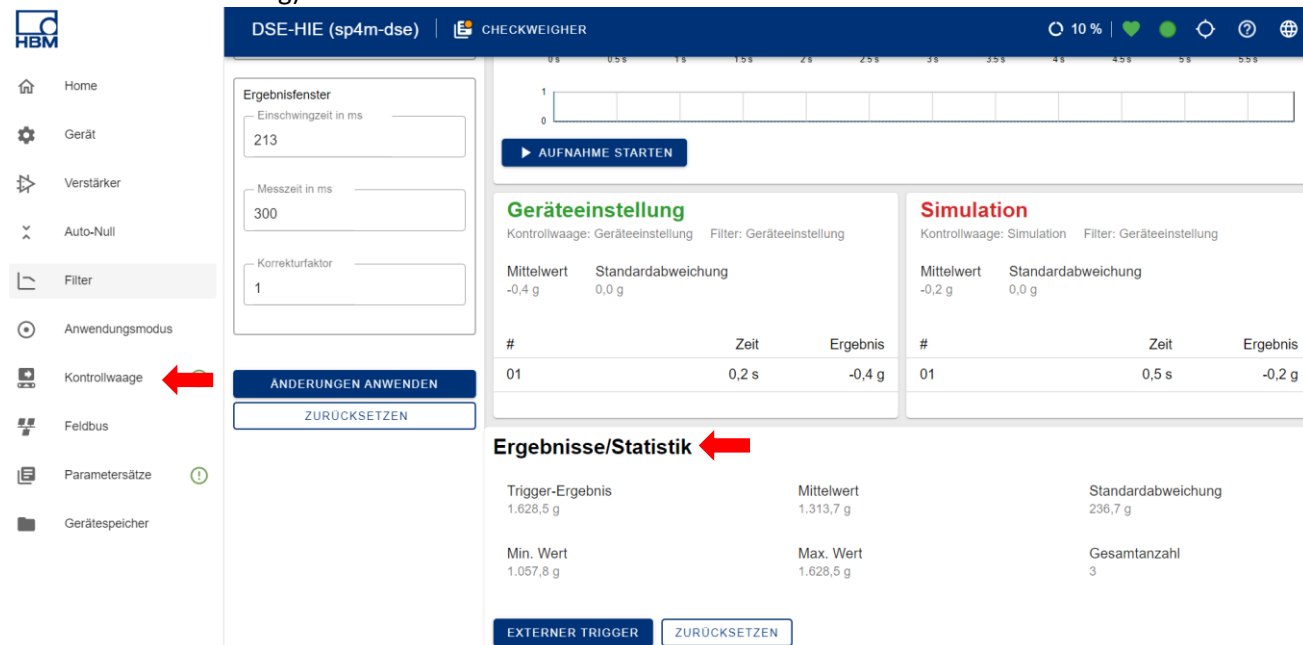


Das Programm läuft, wenn neben den eingefügten Geräten grüne Kreise sind. Siehe Bild:



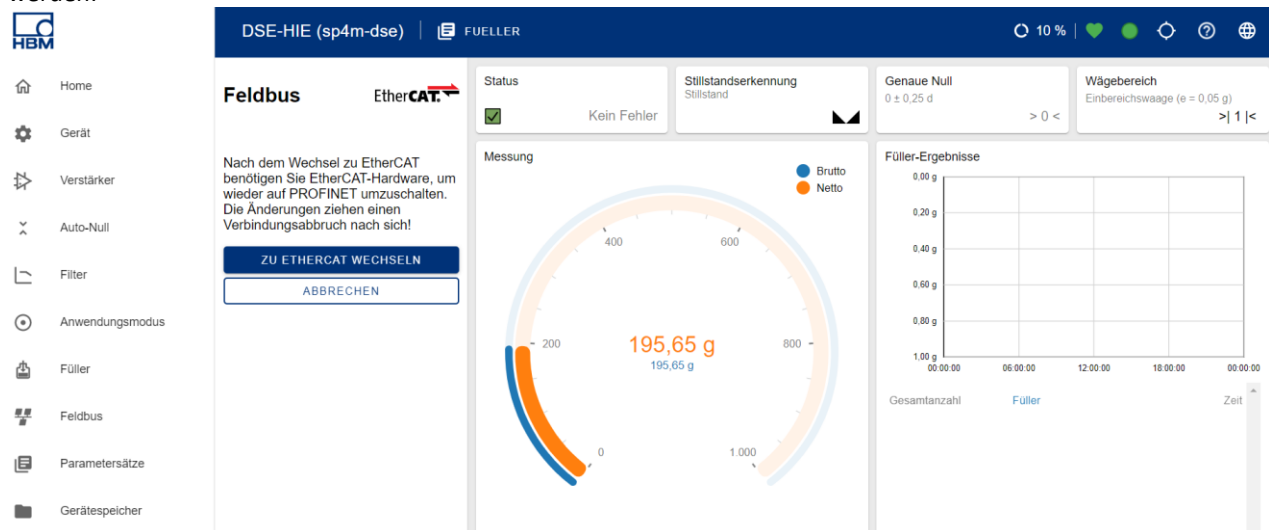
Hinweis: Mit dem Hochladen bleibt der Code im Raspberry gespeichert. Auch nach dem Abschalten des PI's und schließen der Codesys Software am Computer.

Nun erhält das DSE die Signale des Tasters und zählt mit jedem Drücken hoch (erkennt nur Tastendrucke außerhalb der Messung). Dies lässt sich in der Weboberfläche des DSE beobachten.



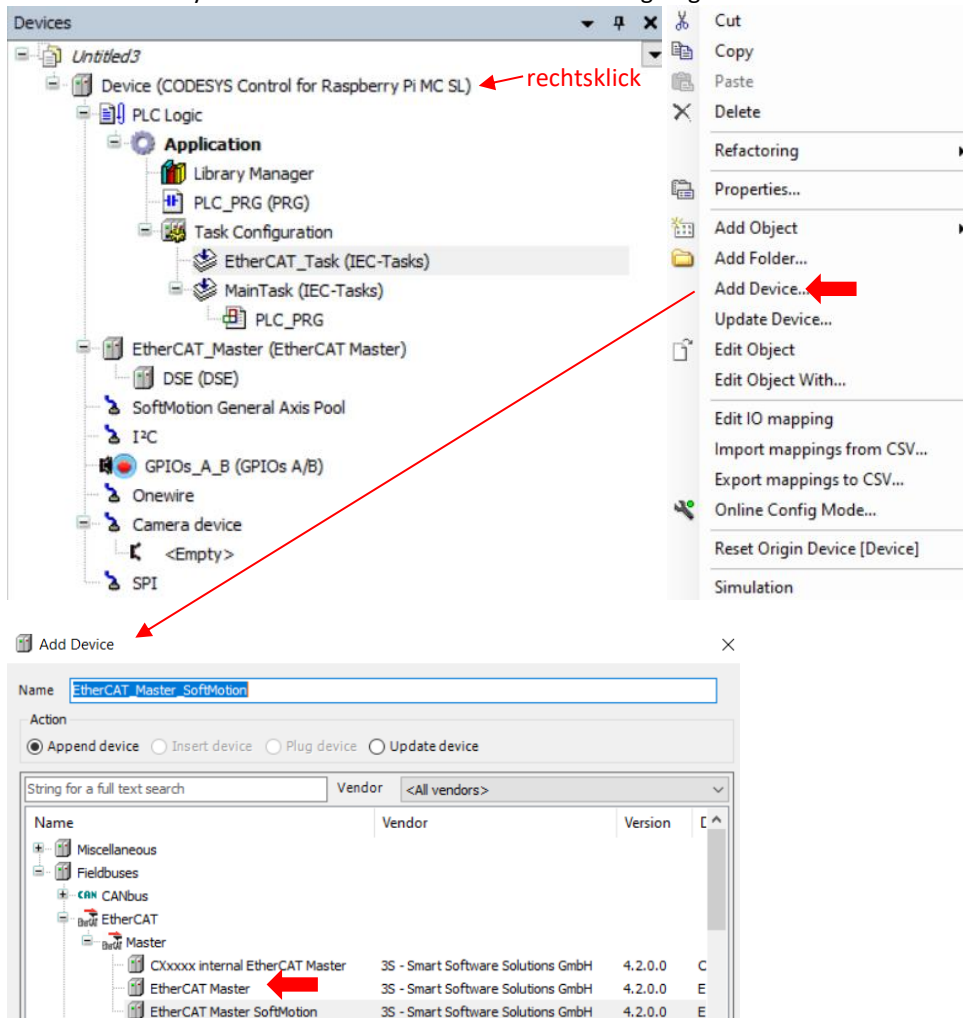
Alternativ: Verbindung via EtherCAT

Optional kann die Verbindung auch über EtherCAT aufgebaut werden. Zunächst muss der DSE auf EtherCAT umgestellt werden:

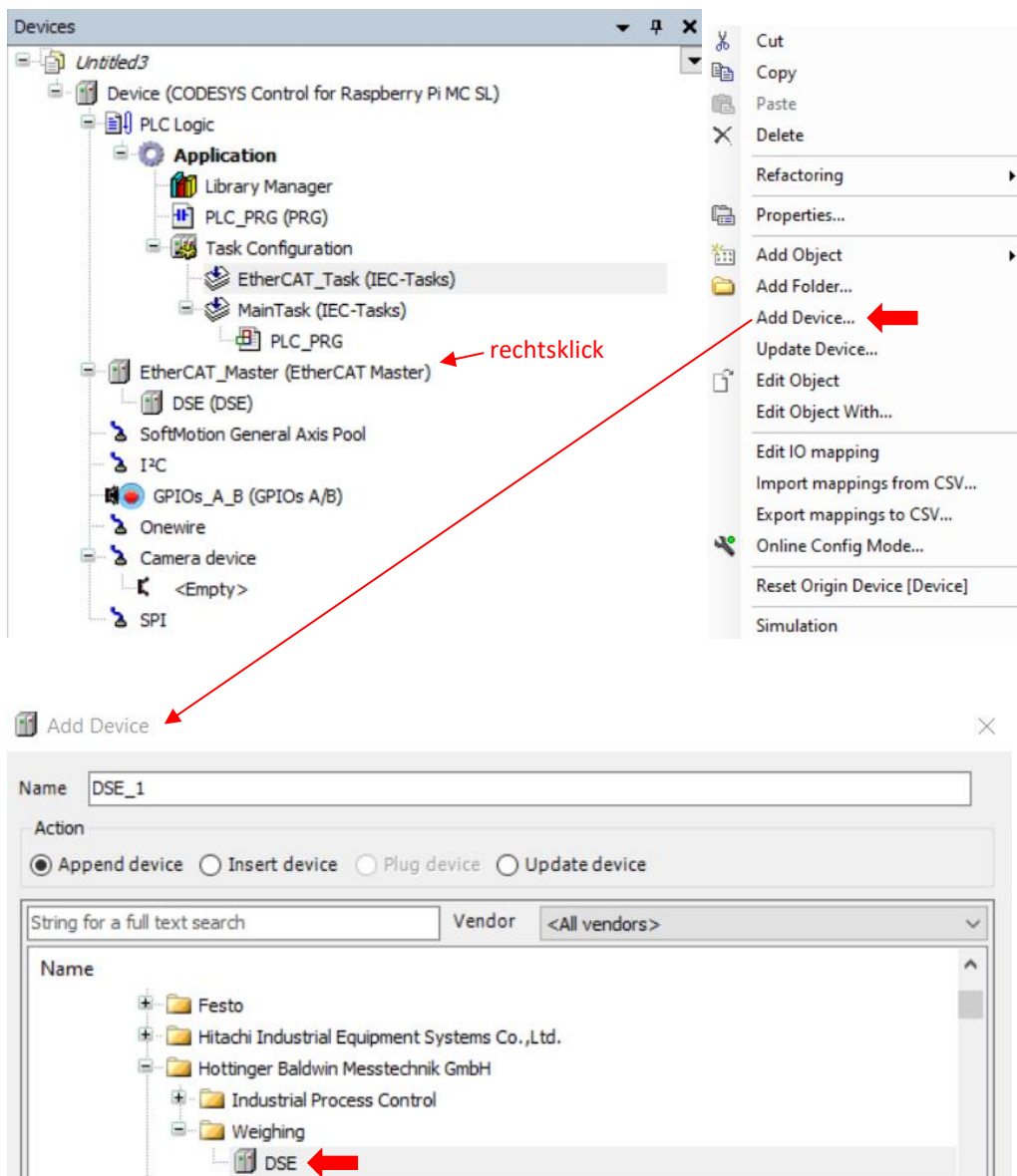


Nun die XML-Datei des DSE herunterladen

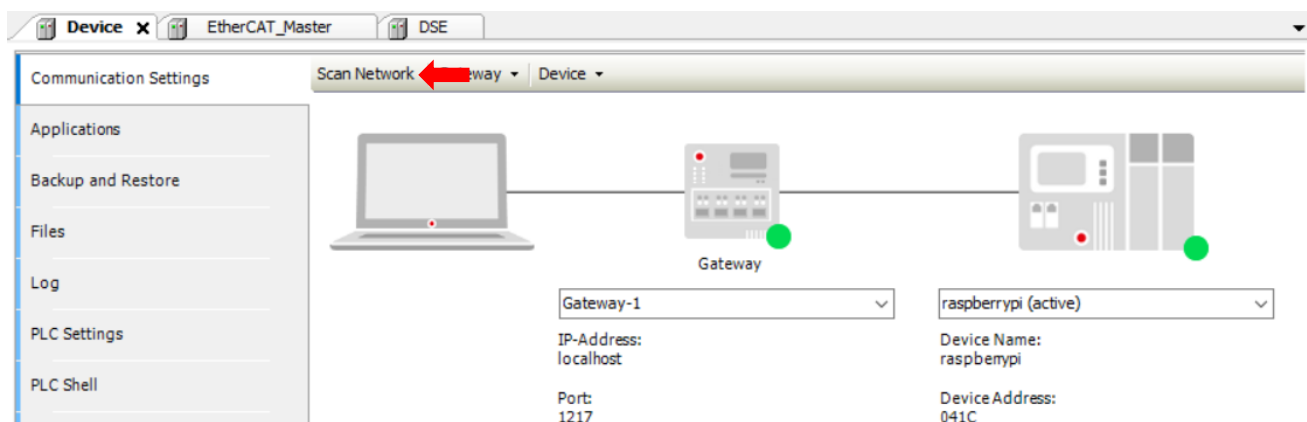
Es muss in Codesys ein Ethernet Master und der DSE hinzugefügt werden.



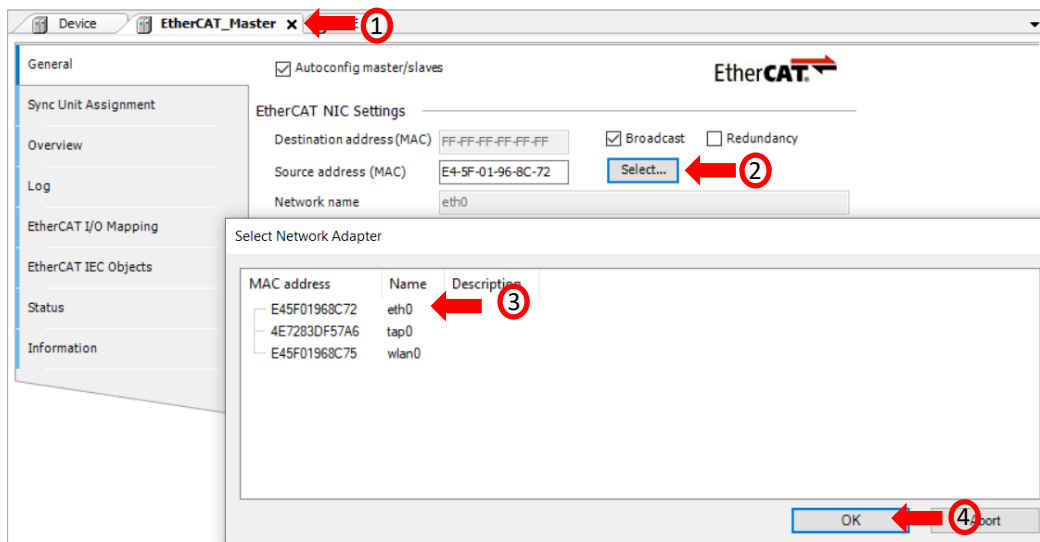
Nun ist der Ethercat Master hinzugefügt, jetzt muss nur noch der DSE eingefügt werden.



Nun kann die Verbindung zum RaspberryPi aufgebaut werden.



Ist diese Verbindung aufgebaut, kann die EtherCAT Master konfiguriert werden. Nach dem Klick auf „select...“ muss nochmal der Benutzername und Passwort des RaspberryPi eingegeben werden, dann erscheint folgendes Bild:

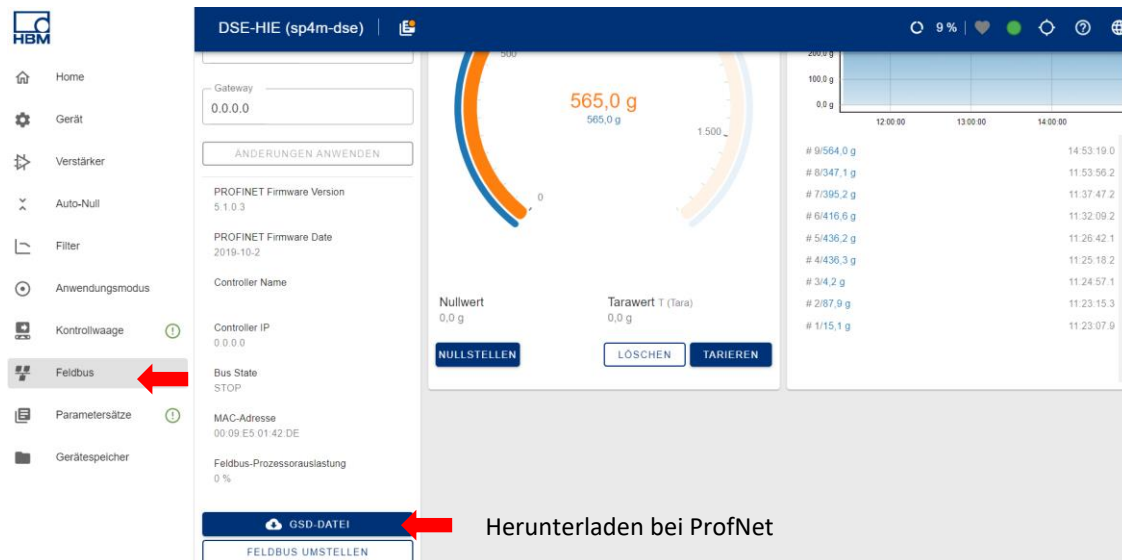


Nun muss das Projekt nur noch gestartet werden.

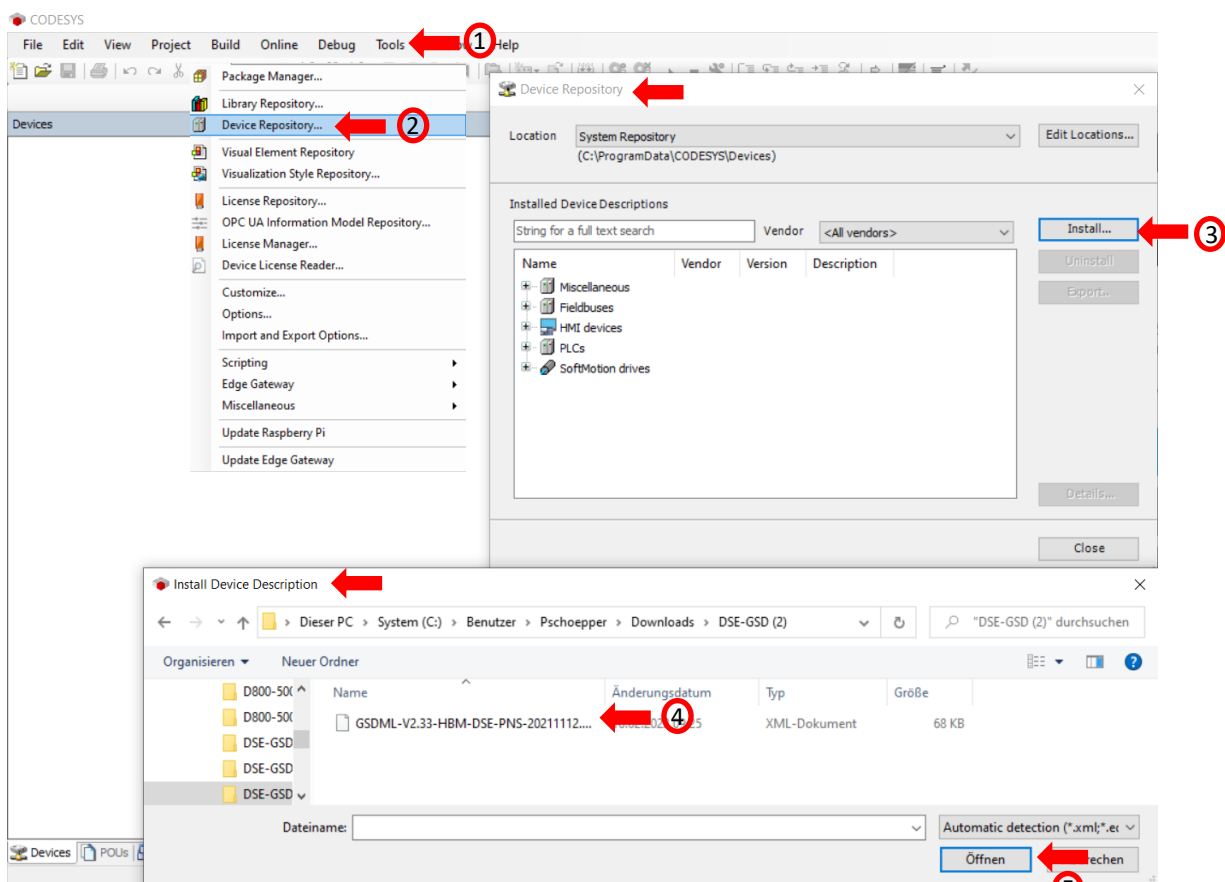


Zusatzinformation: der verwendete Codesys Programmcode

Bei der Profinet-GSDML-Datei handelt es sich um eine Gerätebeschreibungsdatei. Diese enthalten alle Geräte-Funktionalitäten werden für die Implementierung in Programmierungsumgebungen wie TIA-Portal oder Codesys benötigt. Nur so können die, nicht im Standardkatalog hinterlegten Geräte in den virtuellen Aufbau hinzugefügt werden und auch von den Steuerungen angesteuert werden.



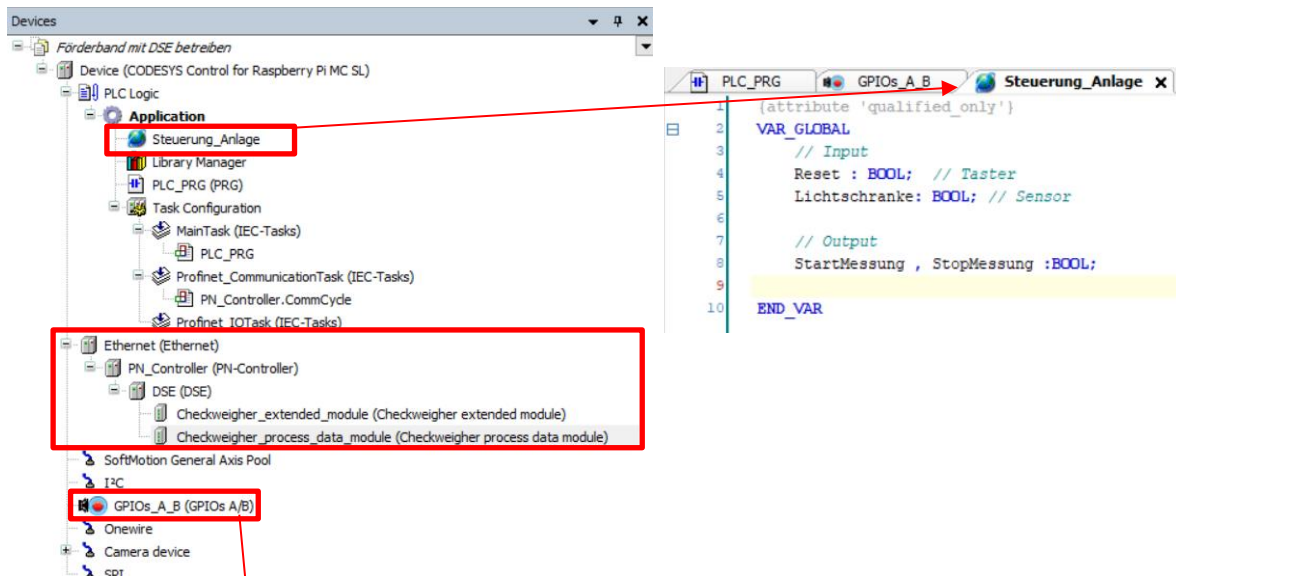
Die GSD-Datei wird in der Programmierungsumgebung hochgeladen, bei Codesys geht das wie folgt:



Nachdem „Öffnen“ gedrückt wurde, ist das Gerät dem „Repository“ hinzugefügt.

Damit die DSE über ProfiNet mit der Codesys-SPS kommunizieren kann, wird der „PN_Controller“ eingefügt. Außerdem werden die unten gezeigten Checkweiger Module aus der bereits geladenen GSDML-Datei benötigt.

Zunächst müssen Variablen erzeugt werden, in diesem Beispiel werden diese global unter „Steuerung_Anlage“ gespeichert:

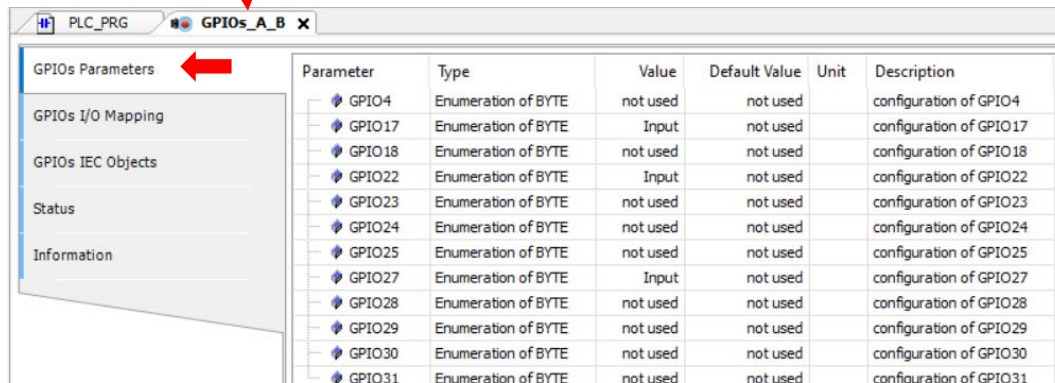


The top screenshot shows the project tree on the left. The 'Steuerung_Anlage' application is selected. Under 'Ethernet (Ethernet)', the 'PN_Controller (PN-Controller)' and 'DSE (DSE)' are highlighted. Under 'I2C', the 'GPIOs_A_B (GPIOs A/B)' is highlighted. The 'GPIOs_A_B' window on the right shows the following variable declaration:

```

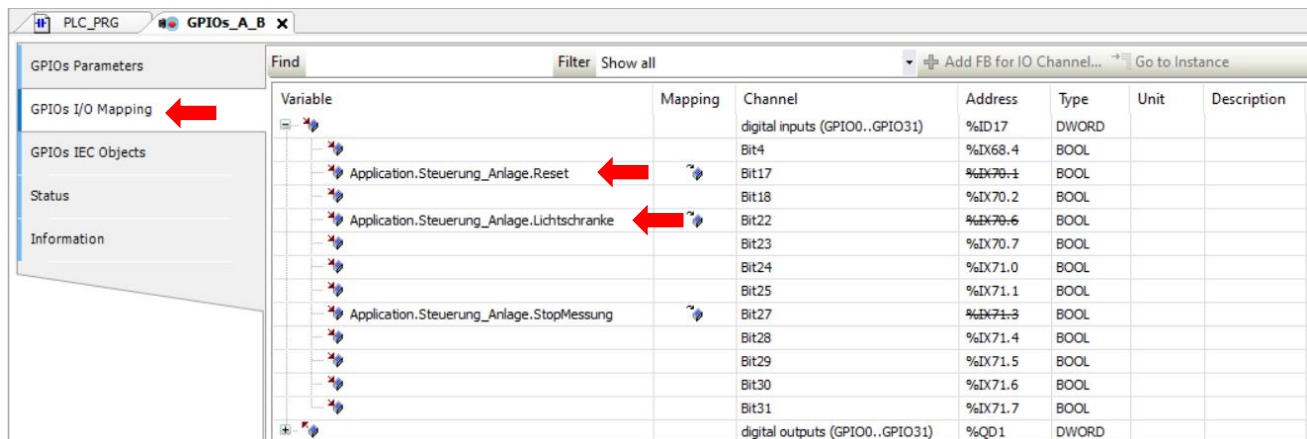
1 {attribute 'qualified_only'}
2
3 VAR_GLOBAL
4   // Input
5   Reset : BOOL; // Taster
6   Lichtschranke: BOOL; // Sensor
7
8   // Output
9   StartMessung , StopMessung :BOOL;
10
11 END_VAR

```



The middle screenshot shows the 'GPIOs Parameters' window. The 'GPIOs I/O Mapping' tab is selected. The table below lists the parameters for the GPIOs.

Parameter	Type	Value	Default Value	Unit	Description
GPIO4	Enumeration of BYTE	not used	not used		configuration of GPIO4
GPIO17	Enumeration of BYTE	Input	not used		configuration of GPIO17
GPIO18	Enumeration of BYTE	not used	not used		configuration of GPIO18
GPIO22	Enumeration of BYTE	Input	not used		configuration of GPIO22
GPIO23	Enumeration of BYTE	not used	not used		configuration of GPIO23
GPIO24	Enumeration of BYTE	not used	not used		configuration of GPIO24
GPIO25	Enumeration of BYTE	not used	not used		configuration of GPIO25
GPIO27	Enumeration of BYTE	Input	not used		configuration of GPIO27
GPIO28	Enumeration of BYTE	not used	not used		configuration of GPIO28
GPIO29	Enumeration of BYTE	not used	not used		configuration of GPIO29
GPIO30	Enumeration of BYTE	not used	not used		configuration of GPIO30
GPIO31	Enumeration of BYTE	not used	not used		configuration of GPIO31



The bottom screenshot shows the 'GPIOs I/O Mapping' window. The table below lists the mapping of variables to channels.

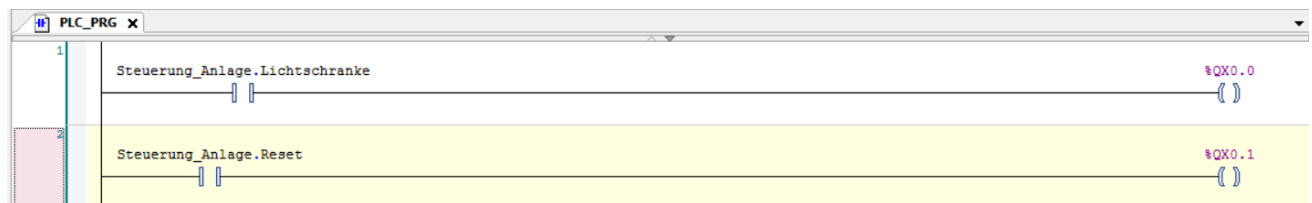
Variable	Mapping	Channel	Address	Type	Unit	Description
		digital inputs (GPIO0..GPIO31)	%ID17	DWORD		
		Bit4	%IX68.4	BOOL		
Application.Steuerung_Anlage.Reset		Bit17	%IX70.1	BOOL		
Application.Steuerung_Anlage.Lichtschranke		Bit18	%IX70.2	BOOL		
		Bit22	%IX70.6	BOOL		
		Bit23	%IX70.7	BOOL		
		Bit24	%IX71.0	BOOL		
Application.Steuerung_Anlage.StopMessung		Bit25	%IX71.1	BOOL		
		Bit27	%IX71.3	BOOL		
		Bit28	%IX71.4	BOOL		
		Bit29	%IX71.5	BOOL		
		Bit30	%IX71.6	BOOL		
		Bit31	%IX71.7	BOOL		
		digital outputs (GPIO0..GPIO31)	%QD1	DWORD		

Externes Signal verarbeiten

Die Adresse des „externen Signals“ ist %QX0.0 und für den Reset der Statistik ist %QX0.1.

Variable	Mapping	Channel	Address	Type	Unit	Description
Checkweiger result (2000/03)			%ID 10	REAL		
Inputs PS			%IB44	Enumeration of BYTE		
Checkweiger result status (2000/04)			%IW23	UINT		
Inputs PS			%IB48	Enumeration of BYTE		
Checkweiger result count (2020/06)			%ID 13	UDINT		
Inputs PS			%IB56	Enumeration of BYTE		
Checkweiger status (2023/05)			%IW29	UINT		
New checkweiger result available (toggles)			%IX58.0	BOOL		
Checkweiger settling			%IX58.1	BOOL		
Checkweiger measuring			%IX58.2	BOOL		
Inputs PS			%IB60	Enumeration of BYTE		
Checkweiger commands (2023/07)			%IW31	UINT		
External trigger (Light barrier)			%IX62.0	BOOL		
Clear checkweiger statistic			%IX62.1	BOOL		
Inputs PS			%IB64	Enumeration of BYTE		
Checkweiger commands (2023/07)			%QW0	UINT		
External trigger (Light barrier)			%QX0.0	BOOL		
Clear checkweiger statistic			%QX0.1	BOOL		
Outputs CS			%IB65	Enumeration of BYTE		

Diese müssen nun von unseren Inputs angesteuert werden.



Nun muss das Programm nur noch gestartet werden.



Man sieht, dass das Programm funktioniert, wenn neben den Eingefügten Devices grüne Kreise sind. Siehe Bild:

Der Taster ist betätigt und ein „true“ aus.

Watch 1	Expression	Application	Type	Value	Prepared value	Execution

Nun erhält das DSE die Signale des Tasters und zählt mit jedem Drücken hoch (erkennt nur Tastersignale außerhalb der Messung). Dies lässt sich auch in der Weboberfläche der DSE beobachten.

Ergebnisse/Statistik

Trigger-Ergebnis 381,3 g	Mittelwert 925,5 g	Standardabweichung 576,2 g
Min. Wert 366,1 g	Max. Wert 1.875,0 g	Gesamtanzahl 8

Hinweis: Mit dem Hochladen bleibt der Code im Raspberry gespeichert. Auch nach dem Abschalten des Pis und schließen der Codesys-Software am Computer.

Rechtlicher Hinweis

Dieses Beispiel dient lediglich der Veranschaulichung. Es unterliegt keinen Gewährleistungen oder Haftungsansprüchen.