

Erweiterte Kompetenzübung

Datum: 10.01.2019

Autor: Mario Fentler 5CHIT

Teammitglieder:

Tesanovic, Özer, Rousavy, Pader-Barosch - 5AHIT

Ziel dieser Übung ist es über Codesys echte Hardware anzusprechen und über einen Button eine Lampe zum Leuchten zu bekommen.

Voraussetzungen

Folgende Dinge müssen im Vorfeld heruntergeladen werden:

- CODESYS V3.5 SP13 Patch1
- [CODESYS Controll for Raspberry Pi SL 3.15.13.20](#)
(Alle Versionen -> 3.15.13.20)
- [WAGO USB Treiber für Ethercat](#)
(Downloads -> Gerätedateien -> Gerätetreiber -> "WAGO USB Service Kabel und Treiber / Serie 750 und 857")

Arbeitsschritte

Codesys Controll for Raspberry Pi

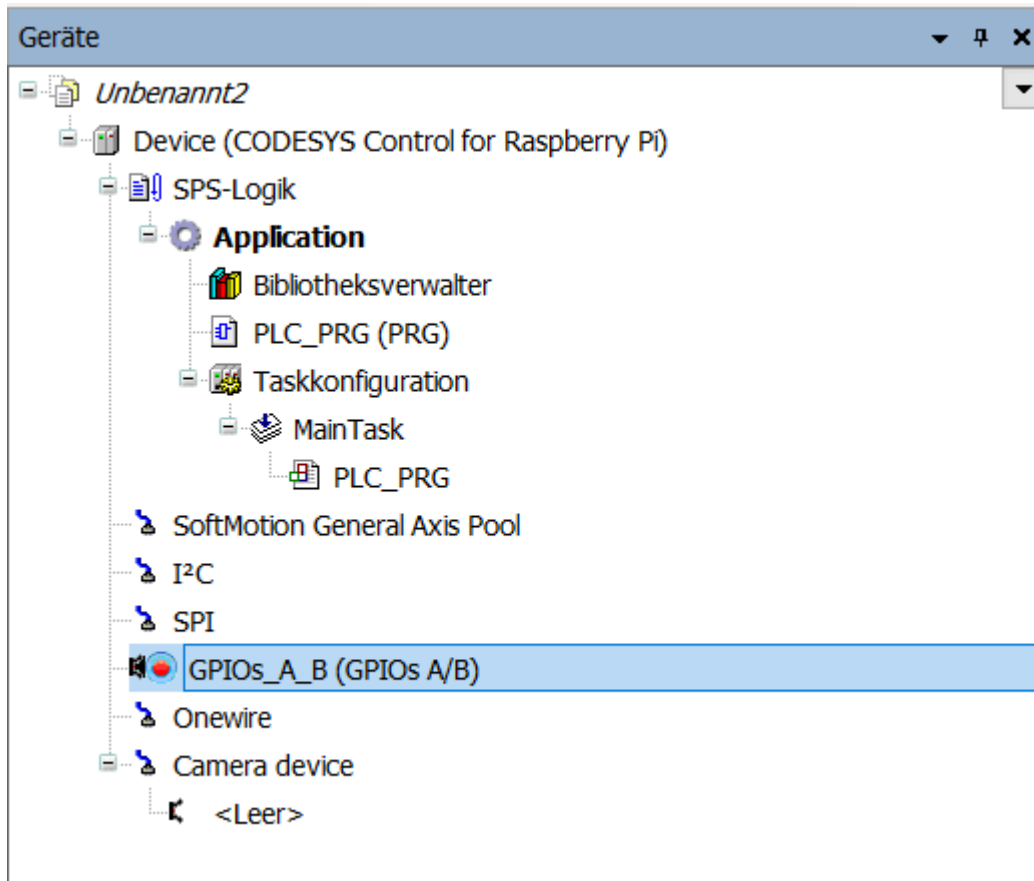
Als erstes wird ein neues Standardprojekt in Codesys erstellt. Anschließend wird das vorher heruntergeladene **Codesys Controll for Raspi** Package hinzugefügt.

Dazu: *Tools/Packagemanager/Installieren*

Das Package braucht man um jetzt beim **Erstellen eines neuen Standardprojekts** das Gerät "*Codesys Controll for Raspberry Pi*" auswählen zu können.

GPIOs ändern

Anschließend ändert man die GPIOs. Dazu klickt man mit der **Rechten Maustaste auf GPIOs A/B -> Gerät aktualisieren -> GPIOs B+**



Wenn man jetzt nochmal mit der Rechten Maustaste auf die GPIOs klickt -> E/A-Abbild bearbeiten -> dann sind dort die GPIO In-/Output Adressen aufgelistet. Denen kann man Variablen zuweisen.

WAGO USB Treiber

Als nächstes wird der Wago USB Treiber installiert:

Die ZIP Datei wird entpackt (nur das File ohne "No_modules" ist von Bedeutung). Dieses wird dann in Codesys über **Tools/Gerätorepo, installieren** hinzugefügt.

Raspberry Pi hinzufügen

Die Steuerung wird über einen Raspberry Pi angesprochen. Diesen muss man natürlich auch noch als Gerät hinzufügen um ihn nutzen zu können.

Der Raspberry Pi befindet sich im Prio Netzwerk. Um damit zu kommunizieren muss man sich auch in diesem Netzwerk befinden. Anschließend geht man in Codesys auf **Tools/Update Raspberry -> Durchsuchen**. Dort kann man auch noch das installierte Codesyscontrol Package überprüfen und gegebenenfalls ändern.

oder den Raspberry Pi starten oder stoppen.

The screenshot shows a software interface titled "Raspberry Pi" with several sections:

- Login-Anmeldedaten:** Fields for "Benutzername" (containing "pi") and "Passwort".
- Zielgerät auswählen:** Field for "IP-Adresse" (containing "10.50.10.21") and a yellow "Durchsuchen" button.
- Package:** A "Version" dropdown menu showing "V3.5.13.20", with "Installieren" and "Entfernen" buttons below it.
- Package-Verzeichnis:** A text field containing "C:\Users\mario\CODESYS Control for Raspberry" and a browse button "...".
- System:** Two buttons: "System-Info" and "Zielgerät neu starten".
- Laufzeitsystem:** Three buttons: "Start", "Stop", and "Applikation deaktivieren".

Programme ausführen

Wenn man Programme dann auf der Hardware ausführen will klickt man auf **Online/Einloggen** -> **Gerät/Bevorzugte Geräte verwalten** -> **Neues hinzufügen** und gibt dort die IP-Adresse vom Pi an.



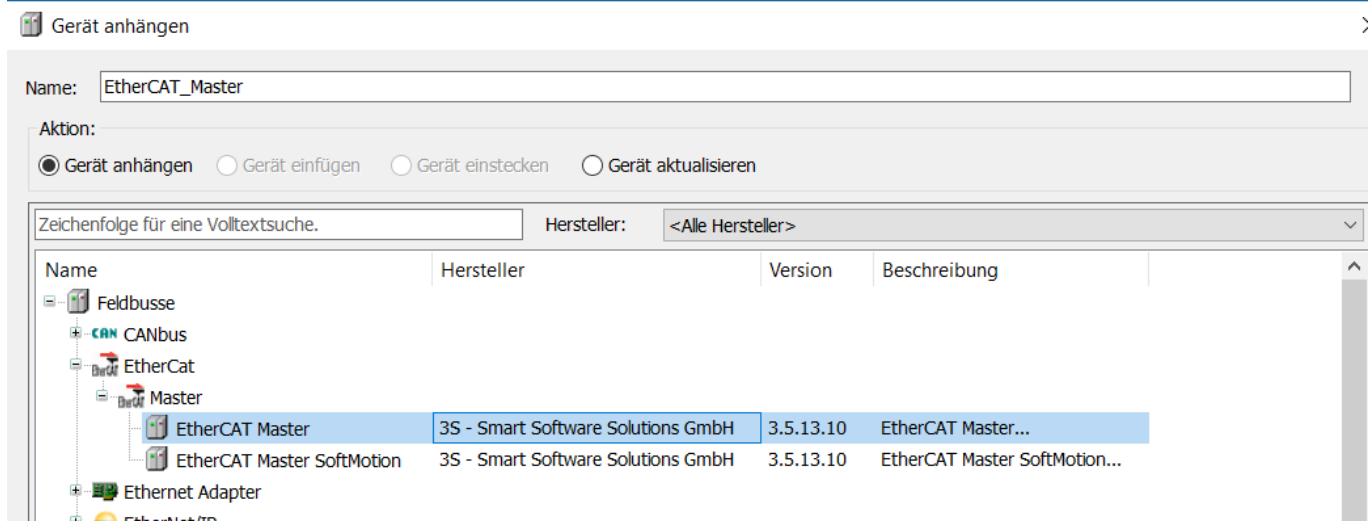
Anschließend kann man die Programme so ausführen, wie wir das auch schon bei den Übungen davor gemacht haben. Der einzige Unterschied ist, dass wir uns jetzt nicht mehr im Simulationsmodus befinden.

Master - Slave

Davor muss aber noch ein Master eingefügt werden. Der Master entscheidet welcher Slave das Signal empfangen darf. Der Master ist eine reine Softwareimplementierung. Die Slaves hingegen sind die angeschlossenen Hardwareslots.

Um nun einen neuen **Master** hinzuzufügen:

Rechte Maustaste auf Device -> Geräte anhängen -> Ethercat -> Master



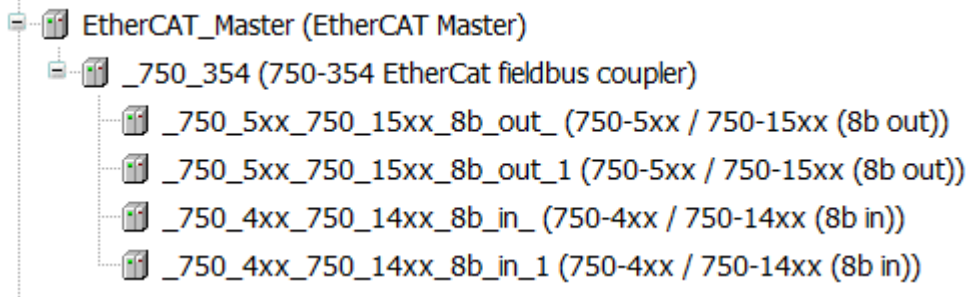
Als nächster Schritt werden nun die **Slaves** hinzugefügt. Die Slaves besitzen alle eine Nummer die auf der Hardware ersichtlich ist. Je nach Nummer kann man herausfinden ob es sich dabei um eine Input- oder eine Outputkarte handelt.

Rechte Maustaste auf EtherCAT Master -> Gerät anhängen -> 750-354 EtherCAT Fieldbus (Die Nummer muss man auch von der Hardware ablesen) -> **Gerät anhängen** -> Nach **WAGO** suchen.

Dort kann man jetzt die einzelnen In/Output Karten auswählen. Dabei muss man auch darauf achten, dass man sie in der Reihenfolge hinzufügt, in der sie auch gesteckt sind.

Für unsere Übung brauchen wir die Karten 530,530,430,430. Da es die so aber nicht in der Liste gibt müssen wir die **750-5xx/750-4xx** mit jeweils entweder **8b in** oder **8b out** nehmen, je nachdem ob wir Input oder Output haben.

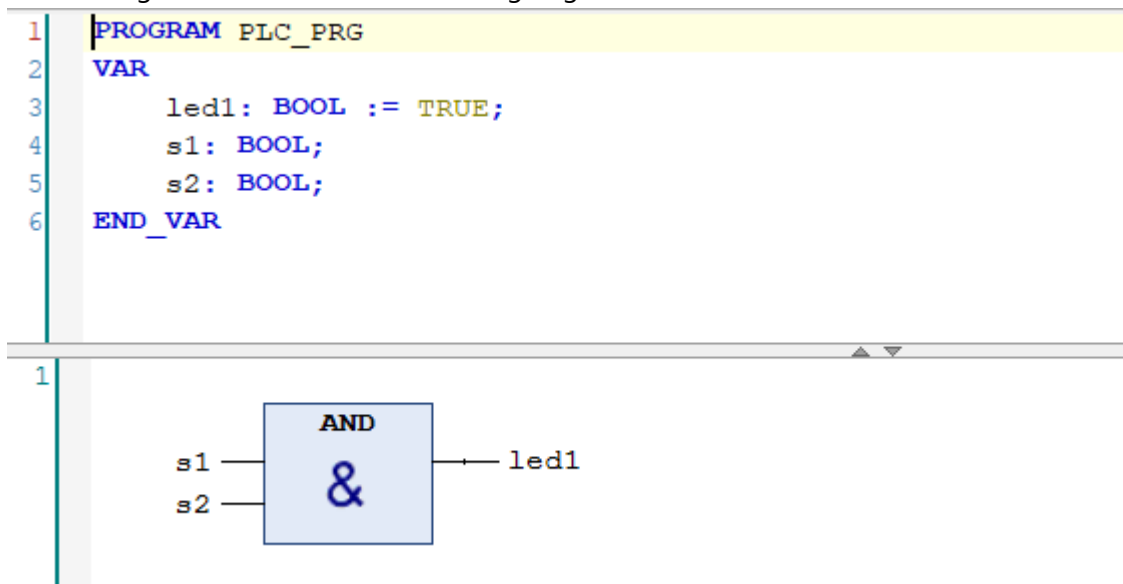
Das Resultat sieht dann so aus:



Hardware anschließen

Nun ist es an der Zeit die Schalter und die LED anzuschließen. Nachdem das gemacht wurde, wird ein **neues Programm (FUP)** erstellt welches auf Button Inputs reagiert und die Lampe einschaltet. Dieses Programm hat zwei Input (Buttons) und eine Output(LED) Variable.

Dieses Programm wird in den **Maintask** gefügt.



Hardware-Software Verbindung

Nun werden diese Variablen den Inputs und Outputs der Slaves zugeordnet. Dabei kommt es darauf an in welche Öffnungen die Kabel beim Slave gesteckt wurden. Diese sind von oben nach unten nummeriert.

In der Softwareumsetzung sieht das dann so aus:

Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
Application.PLC_PRG.s1		Channel 1 Data	%IX5.0	BIT		Channel 1 Data
Application.PLC_PRG.s2		Channel 2 Data	%IX5.1	BIT		Channel 2 Data
		Channel 3 Data	%IX5.2	BIT		Channel 3 Data
		Channel 4 Data	%IX5.3	BIT		Channel 4 Data
		Channel 5 Data	%IX5.4	BIT		Channel 5 Data
		Channel 6 Data	%IX5.5	BIT		Channel 6 Data
		Channel 7 Data	%IX5.6	BIT		Channel 7 Data
		Channel 8 Data	%IX5.7	BIT		Channel 8 Data

In dem Fall wurden die beiden Verbindungen zu den Buttons in den letzten Slave in die Öffnungen 1 und 2 gesteckt.

Überprüfen der Arbeit

Um die Arbeit zu überprüfen wechselt man den in den zuvor beschriebenen Modus (Online/Einloggen) und startet das Programm. Daraufhin kann man die Knöpfe drücken und sieht eine LED aufleuchten.