Raspberry PI with Standard CODESYS V3

Erste Schritte

Version: 3.0
Template: templ_tecdoc_de_V1.0.docx
Dateiname: RaspberryPI_CodesysV3_FirstSteps_DE.doc

INHALT

1 Produktbeschreibung

Dieses Produkt beinhaltet ein CODESYS Control Laufzeitsystem für den Raspberry Pi (s. http://www.raspberrypi.org/), sowie Treiber-Unterstützung für die Erweiterungshardware Raspberry PiFace Digital, Raspberry Pi Camera und diverse Geräte/Platinen mit I²C-Schnittstelle.

Das empfohlene Betriebssystem ‚Raspian' kann über folgende Links bezogen werden:

Allgemeine Downloadseite: https://www.raspberrypi.org/downloads/

Aktuelle Version: https://downloads.raspberrypi.org/raspbian_latest

Das Produkt wird mittels des CODESYS Updatemanagers (CODESYS PlugIn RPIUpdate) in die Linux Distribution Raspbian installiert. Nach jedem Neustart des Raspberry Pi wird das Laufzeitsystem automatisch gestartet. Wenn keine gültige Lizenz vorhanden ist, arbeitet es für zwei Stunden ohne funktionale Einschränkung und beendet sich dann automatisch.

Das Laufzeitsystem verfügt über keine harten Echtzeiteigenschaften. Der Jitter ist abhängig von diversen Faktoren, u.a. den parallel ausgeführten Linux-Applikationen, liegt im Idealfall bei etwa 50µs mit Maximalwerten bei etwa 400 µs.

Hinweis: Das CODESYS Control Laufzeitsystem für den Raspberry Pi CODESYS unterstützt CODEYS BACnet. Eine Demoversion des BACnet-Stacks ist im Laufzeitsystem enthalten, eine Vollversion kann aus dem CODESYS Store bezogen werden.

Das enthaltene CODESYS Laufzeitsystem unterstützt folgende Funktionen:

- CODESYS EtherCAT Master
- CODESYS Profinet Master
- CODESYS Modbus TCP Master / Slave
- CODESYS Modbus RTU Master / Slave (serielle Schnittstelle muss im Betriebssystem bereitgestellt werden)
- CODESYS WebVisu
- CODESYS SoftMotion CNC
- CODESYS OPC/UA Server
- CANopen via EL6751 Gateway
- CODESYS EtherNet/IP Scanner
- CODESYS EtherNet/IP Adapter

Das CODESYS BACnet wird ebenfalls unterstützt. Eine Demoversion des BACnet-Stacks ist im Laufzeitsystem enthalten, eine Vollversion kann aus dem CODESYS Store bezogen werden.

Dieses Produkt besteht aus:

- Debian Packet mit CODESYS Control for Raspberry Pi
- CODESYS Plugin zum Aktualisieren der Software
- CODESYS-Gerätebeschreibungen für Raspberry Pi, Raspberry PiFace Digital, Raspberry PiFace Control&Display, Raspberry Pi Camera, Geräte/Platinen mit I²C-Schnittstelle (SRF02, Adafruit PWM, MPU6050, MPU9150, AK8975), SPI-Schnittstelle (MCP3008, MCP23S17) oder 1-wire-Schnittstelle (DS20B18).

Das Produkt bietet u. A. die Möglichkeit, weitere Geräte über SPI, I²C oder 1-wire anzuschließen.

Dieses Produkt ist für Test- und Lehrzwecke gedacht. Ein Einsatz für industrielle Zwecke wird nicht empfohlen.

2 Installation, Konfiguration und Lizenzierung

Starten Sie CODESYS. Installieren Sie über den Package Manager, den Sie im Menü Tools finden, die Package-Datei, die Sie vom CODESYS Store heruntergeladen haben.

Sie benötigen für die Installation von CODESYS Control for Raspberry SL ein operatives Raspbian-Betriebssystem. Sollten Sie dieses noch nicht haben, führen Sie bitte die unter *Vorarabeiten* beschriebenen Schritte durch. Ansonsten fahren Sie bei *Installation* fort.

2.1 Vorarbeiten

Folgen Sie den auf https://www.raspberrypi.org/downloads/ beschriebenen Schritten, um auf die Speicherkarte Ihres Raspberry Pi ein funktionierendes Betriebssystem zu installieren. Wir empfehlen das Verwenden von Raspbian.

Anschließend wird der Raspberry Pi mit der SD-Karte gestartet. Dabei sollte das Gerät über den Netzwerkanschluss Verbindung zu einem DHCP-Server haben, damit es eine gültige Netzwerkadresse erhält.

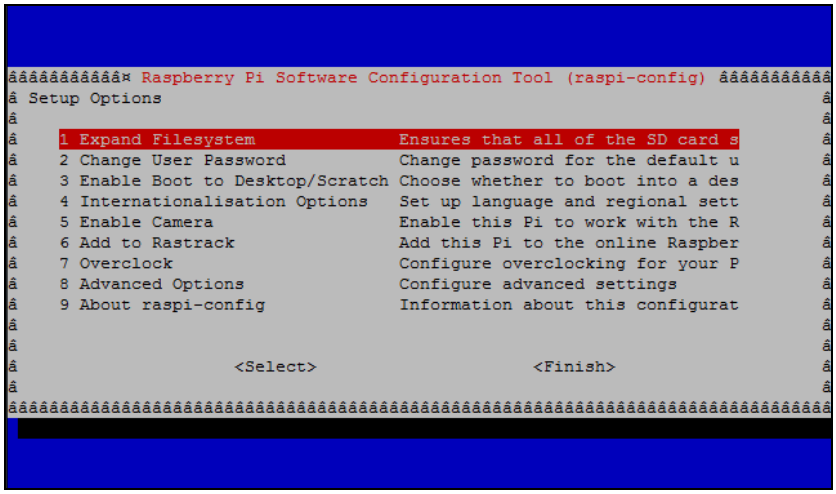
Um die Netzwerk-Adresse zu erfahren, verbinden Sie eine Tastatur und Maus sowie einen Monitor mit Ihrem Gerät, loggen Sie sich ein (User: `pi`, Passwort: `raspberry`[1]), und führen in einer Konsole `ifconfig` aus. Oder führen Sie aus der Windows-Konsole einen ping an den Gerätenamen (default: `RaspberryPi`) aus, der Ihnen die Netzwerk-Adresse liefert. Alternativ dazu können Sie in CODESYS im Menü Tools das Kommando „Update RaspberryPi“ ausführen und im sich öffnenden Dialog „Scan“ auswählen. Dieser liefert Ihnen die IP-Adressen der im Netzwerk verfügbaren RaspberryPi-Geräte zurück.

Mit der IP-Adresse können Sie sich über SSH mit einem entsprechenden Tool wie Putty einloggen (User: `pi`, Passwort: `raspberry`).

1. Grundkonfiguration
    a. Starten Sie aus einer Konsole sudo raspi-config und führen folgende Aktionen aus

       Hinweis: Ab der Debian-Version „Jessie“ gibt es auch eine grafische Konfigurationsoberfläche mit denselben Einstellungen.

    b. Filesystem expandieren



    c. Optional: Geben Sie dem Gerät einen eigenen Namen (8 Advanced Options).
    d. Optional: Aktivieren Sie die Kamera (5 Enable Camera).
    e. Verlassen Sie das Konfigurationstool und führen einen Neustart aus.
2. Aktivieren der Peripherieschnittstellen i²c, spi, 1-wire
    a. Verbinden Sie sich über SSH (z.B. mittels Putty) mit Ihrem Gerät

[1] Bitte beachten Sie, wenn das englische Tastaturlayout gewählt ist, dass y und z vertauscht sind.

b. Editieren Sie die Datei /boot/config.txt z.B. mittels
sudo nano /boot/config.txt

c. Stellen Sie sicher, dass folgende Zeilen enthalten (und nicht mittels # auskommentiert) sind:

d. Optional: i²c
dtparam=i2c_arm=on

e. Optional: SPI
dtparam=spi=on

f. Optional: 1-wire
dtoverlay=w1-gpio-pullup,pullup=1

3. Optional: Camera aktivieren

a. Starten Sie raspi-config (s. 1) und führen “Enable Camera” aus

b. Verbinden Sie sich über SSH (z.B. mittels Putty) mit Ihrem Gerät (Standardzugang „pi“, Passwort „raspberry“) und führen folgende Kommandos in Ihrer Shell aus:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo rpi-update

git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git
cd RPi_Cam_Web_Interface
chmod u+x RPi_Cam_Web_Interface_Installer.sh
./RPi_Cam_Web_Interface_Installer.sh install
```

c. Starten Sie einen Webbrowser und öffnen Sie http://<network address>
Wenn die Installation erfolgreich war, wird die Benutzeroberfläche von RPi Cam Control geladen und Sie können Ihre Kamera konfigurieren.

2.2 Installation

1. Führen Sie in CODESYS im Menü Tools das Kommando „Update RaspberryPi“ aus.

a. Wählen Sie die gewünschte Version aus

b. Geben Sie die korrekten Login-Daten ein (default: pi/raspberry)

c. Wählen Sie die IP-Adresse Ihres Geräts aus

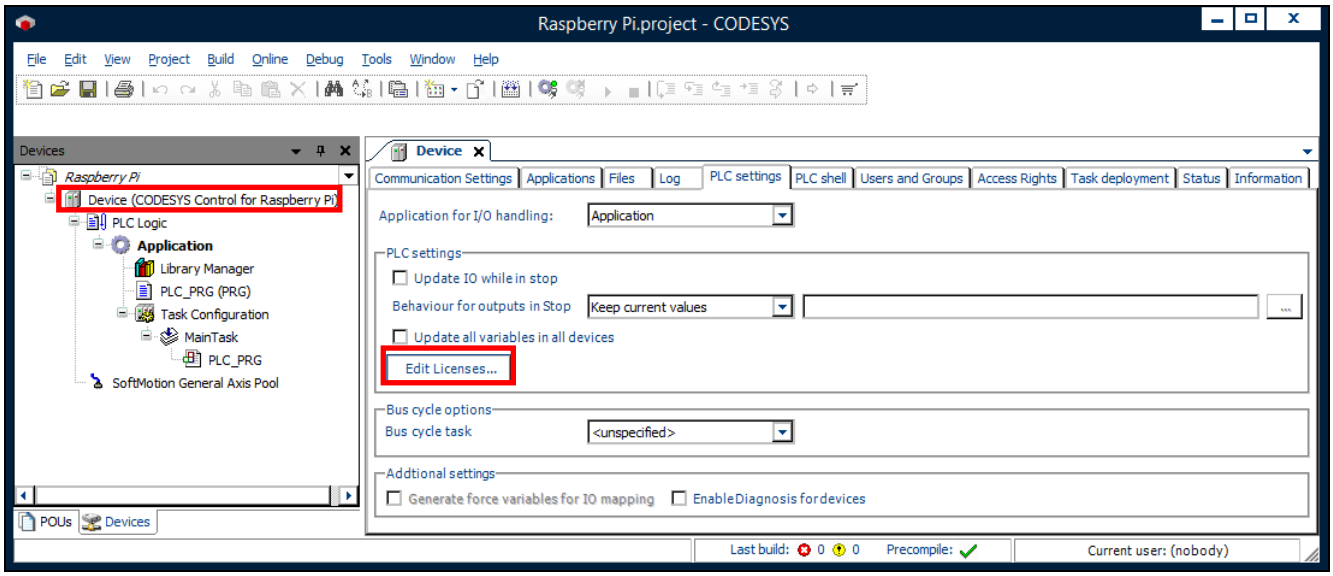
d. Wählen Sie OK und überprüfen Sie, ob im Meldungsfenster gemeldet wird, dass das Laufzeitsystem erfolgreich installiert wurde

2. Nach einem Neustart ist das Gerät betriebsbereit.

2.3 Lizenzierung über das CODESYS Development System

Voraussetzungen: PC mit CODESYS Development System, Internet-Zugang und verbundenem Raspberry Pi.

Die Lizensierung erfolgt über PC / Notebook mit dem CODESYS Development System und verbundenem Raspberry Pi. Die Lizenzeinträge werden über Doppelklick auf die Steuerung unter „SPS-Einstellungen“ / „Lizenzen bearbeiten…“ editiert.



Die Lizenzaktivierung erfolgt unter „Lizenzen installieren“ / „Lizenz aktivieren“ durch Eingabe der Ticketnummer und Übertragung der Lizenz auf den CODESYS Software Key (Softcontainer).

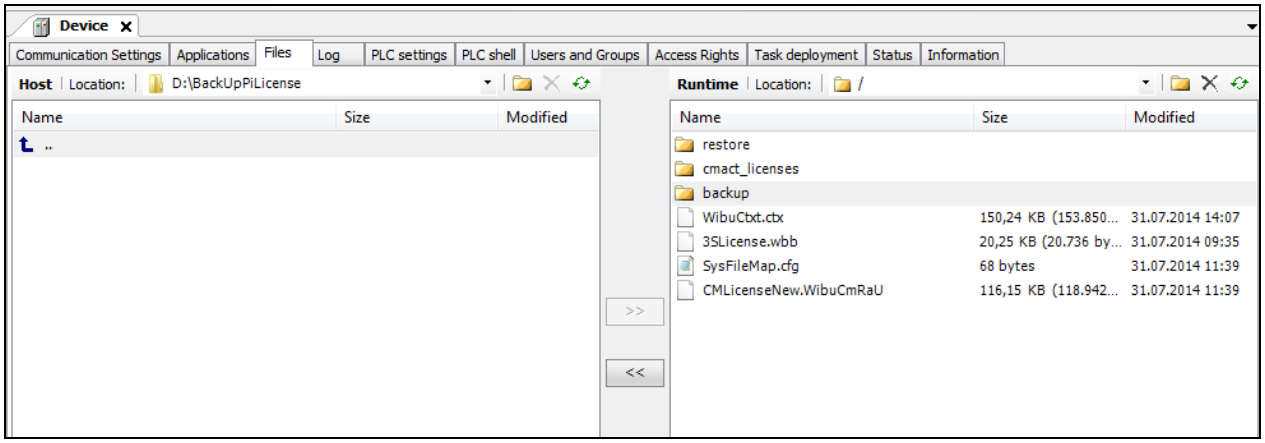
Achtung: Die Lizenz wird bei der Aktivierung an den Raspberry Pi gebunden und kann nur auf dem gleichen Gerät wieder eingespielt werden!

2.4 Sicherung der Lizenz

Verschiedene Vorgänge (z.B. plötzlicher Stromverlust) können zu einem korrupten Dateisystem auf dem Raspberry Pi führen. Zur Sicherung der Lizenz wird daher folgende Vorgehensweise empfohlen:

1. Aktivierung der Lizenz (siehe Beschreibung oben)
2. Re-Boot des Raspberry Pi
3. Backup der Lizenzdatei auf einem externen Datenträger

Um die Lizenzdatei zu sichern, wechseln Sie zum Ordner „Backup“ auf dem Raspberry Pi (zugänglich in CODESYS über Doppelklick auf das Gerät im Reiter „Files“).



Sichern Sie den Inhalt des Ordners („3SLicenseInfo.tar“) auf einen externen Datenträger.

2.5 Reaktivierung der Lizenz

Um die Lizenz zu reaktivieren muss die gesicherte Lizenzdatei in den Ordner „restore“ kopiert werden und im Anschluss ein Re-Boot des Systems erfolgen.

3 Beispielapplikationen

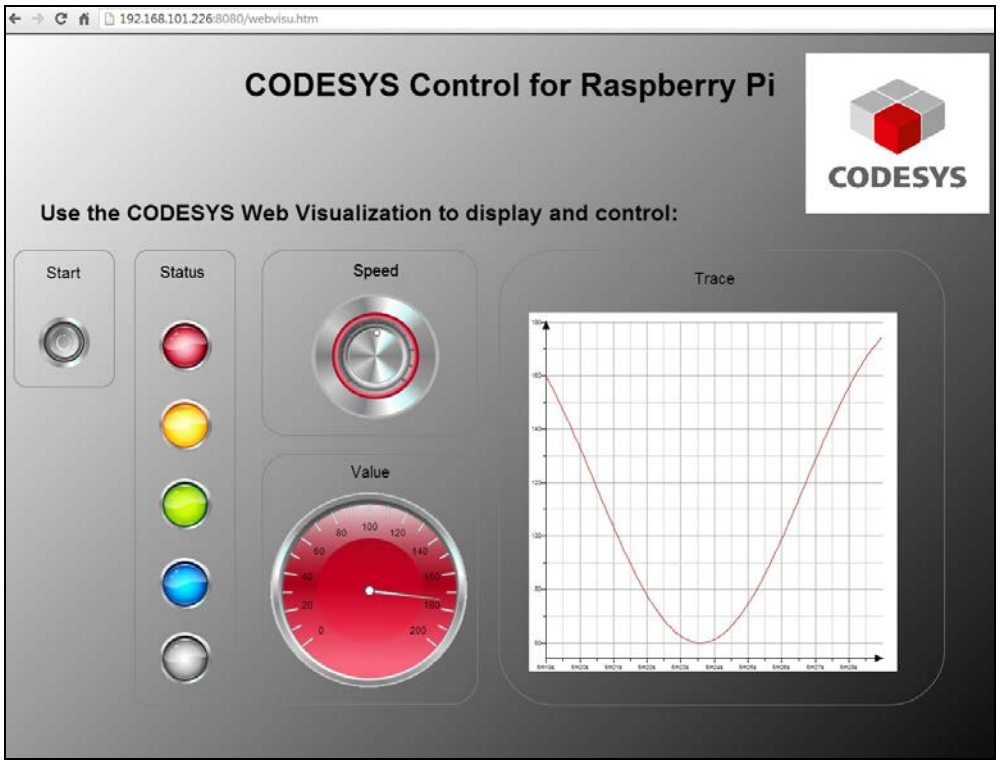
Im Installationsverzeichnis (das Sie sich während der Package-Installation notiert haben) finden Sie folgende Beispielprogramme:

3.1 Webvisu.project

Dieses Beispiel zeigt die Verwendung der CODESYS Visualisierung. Öffnen Sie das Projekt und laden es auf Ihren Raspberry Pi. Dazu doppelklicken Sie auf „Device“ im Gerätebaum links, wählen „Netzwerk durchsuchen“ im Kommunikationsdialog. Dort erscheint, wenn das Gerät sich im gleichen Netzwerk wie Ihr Programmier-PC befindet, Ihr Gerät zur Auswahl. Selektieren Sie es und wählen im Menü „Online“ den Befehl „Einloggen“. Dann starten Sie das Programm beispielsweise mit F5.

Verbinden Sie sich nun mit einem beliebigen Browser (evtl. auch Smartphone) auf die Adresse <Netzwerk-Adresse>:8080/webvisu.htm.

Im Browser sehen Sie dann die im Projekt beispielhaft angelegte Visualisierungsoberfläche:



3.2 Camera.project

(Voraussetzung: Raspberry Pi Camera ist angeschlossen und installiert, siehe 2.1)

Dieses Projekt zeigt wie Sie mit der Raspberry Pi Camera (Erweiterungshardware) einzelne Bilder aufnehmen und als Datei speichern können. Bitte beachten Sie, dass dazu auf manchen Modellen die installierte Applikation „RPi Cam Control“ deaktiviert sein muss. Dies erreichen Sie, indem Sie mit Ihrem Browser auf die Kamera-Konfigurationsseite `http://<Netzwerk-Adresse>` gehen und dort „stop camera“ ausführen.

Laden Sie die Applikation auf Ihre Steuerung, starten das Programm und setzen die Variable `xTakePicture` auf `TRUE`. Die Kamera nimmt daraufhin ein Bild auf und speichert es im lokalen Filesystem unter „Picture.jpg“.

Sie können das Bild im Anschluss über den Datei-Dialog (Doppelklick auf Device, Tab Files, Aktualisieren-Button rechts) auf Ihren Programmier-PC kopieren.

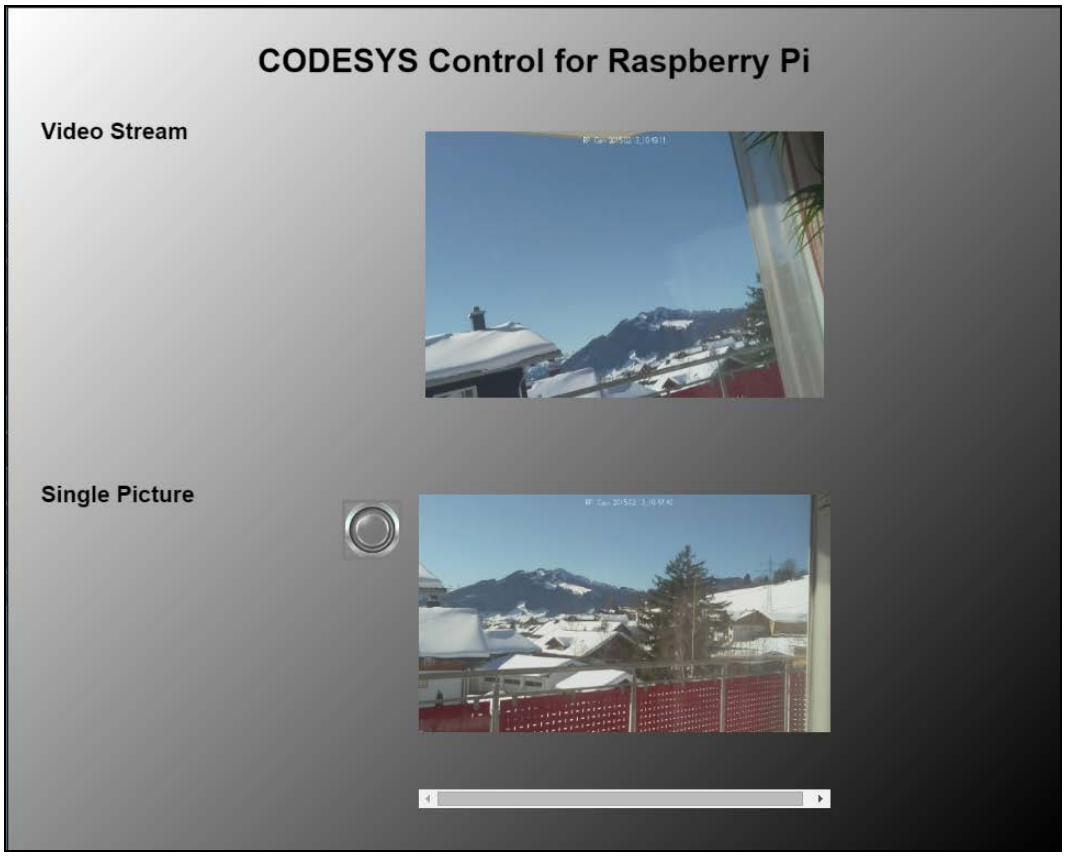
3.3 CameraStream.project

(Voraussetzung: Raspberry Pi Camera ist angeschlossen und installiert, siehe 2.1)

Dieses Projekt zeigt wie ein Kamera-Livestream oder ein Einzelbild in die Webvisualisierung eingebaut werden kann.

Laden Sie die Applikation auf Ihre Steuerung und starten das Programm. Verbinden Sie sich nun mit einem beliebigen Browser auf die Adresse `<Netzwerk-Adresse>:8080\webvisu.htm`. Dort wird Ihnen der Livestream der Kamera im oberen Teil und das letzte Einzelbild im unteren Bild dargestellt; letzteres können Sie mittels des Buttons aktualisieren.

Bitte beachten Sie, dass sich in Abhängigkeit der Version des „RPi_Cam_Web_Interface“ und der enthaltenen Apache-Installation der Default-Pfad ändert von „`/var/www/`“ zu „`/var/www/html/`“. Dieser muss im PLC_PRG in Zeile 3 ggf. angepasst werden.



3.4 GPIO.project

Dieses Projekt zeigt die Verwendung freier GPIOs. In der Konfiguration des Geräts GPIO hat man die Möglichkeit, die Verwendung freier GPIOs als digitalen Input oder Output festzulegen:

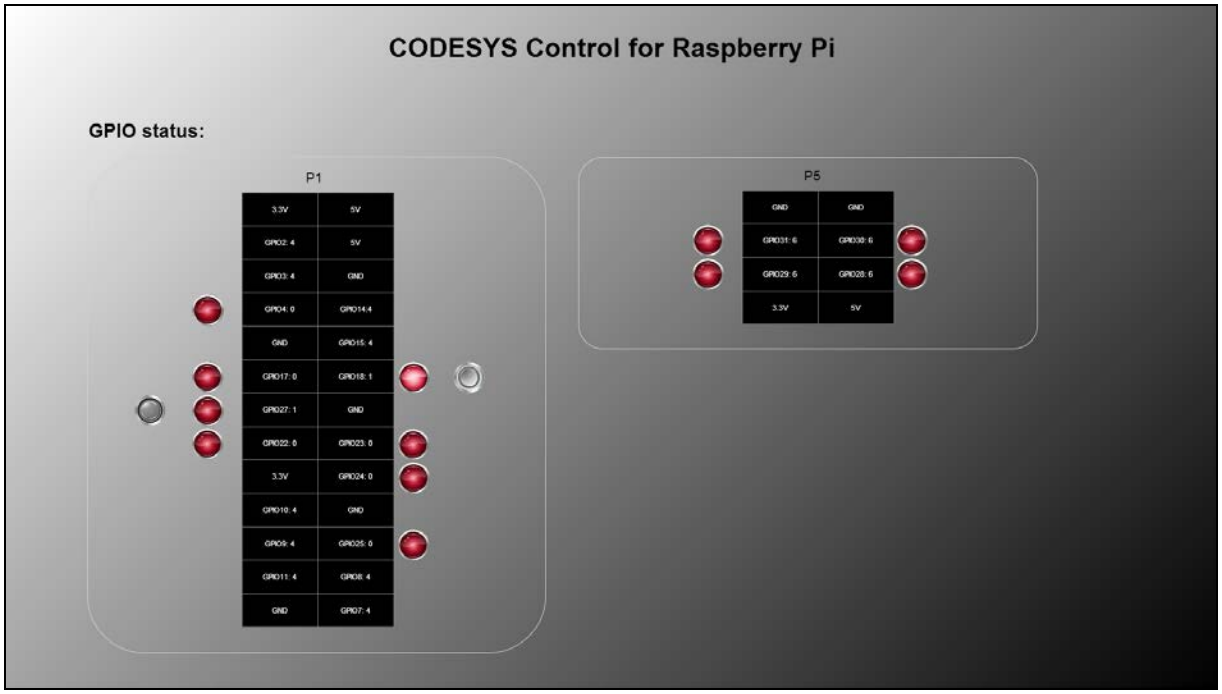
GPIOs

GPIOs Configuration | GPIOs I/O Mapping | Status | Information

Parameter	Type	Value	Default Value	Unit	Description
GPIO4	Enumeration of BYTE	not used	not used		configuration of GPIO4
GPIO17	Enumeration of BYTE	not used	not used		configuration of GPIO17
GPIO18	Enumeration of BYTE	Output	not used		configuration of GPIO18
GPIO22	Enumeration of BYTE	not used	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	not used	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

Um die Ein- und Ausgänge zu verwenden, ist im Reiter „GPIOs I/O Mapping“ je ein DWORD definiert. Deren Bit <X> steht für den Wert des GPIO<X>.

Im Beispiel wurde der GPIO18 als Ausgang definiert und durch das Programm PLC_PRG mit einem Blinksignal belegt. Eine enthaltene Visualisierung stellt den Wert der Eingänge dar und erlaubt das Setzen der Ausgänge:



Um die zusätzlichen GPIOs der Modellvariante Raspberry Pi B+ zu verwenden gibt es ein eigenes, erweitertes GPIO-Gerät, welches man im Gerätebaum in den Slot „GPIOs“ setzen kann.

Bitte beachten Sie, dass in Abhängigkeit von den geladenen Treibern manche GPIOs permanent mit anderen Funktionen belegt sind und evtl. nicht zur Verfügung stehen.

3.5 PiFace.project

(Voraussetzung: Raspberry PiFace Digital ist angeschlossen)

Dieses Beispiel zeigt die Verwendung der E/A-Hardware Raspberry PiFace Digital (8 digitale Ein- und Ausgänge).

Öffnen Sie das Projekt, laden es auf Ihren Raspberry Pi und starten es. Das einfache Programmbeispiel PLC_PRG bewirkt, dass durch Drücken des Tasters S1 der Relaisausgang K0 mit einer Sekunde Zeitverzögerung geschaltet wird. Drücken von S2 bewirkt das sofortige Schalten von K1, welches für eine halbe Sekunde nach dem Loslassen von S2 nachgehalten wird.

Beachten Sie auch, dass Sie mehrere Instanzen (Hardware-Adresse über Jumper JP1, JP2 änderbar) verwenden können, indem Sie den Geräteparameter beim PiFace-Gerät im Gerätebaum entsprechend anpassen.

Die für die Anbindung verantwortliche Bibliothek SPI_PiFace wird im Sourcecode zur Verfügung gestellt und kann als Beispiel für andere Anschaltungen dienen. Basis für die Kommunikation über SPI ist die Bibliothek RaspberryPiPeripherals, deren Schnittstellen dokumentiert sind (siehe Onlinehilfe (F1) -> Bibliotheken).

3.6 PiFaceIoDrv.project

(Voraussetzung: Raspberry PiFace ist angeschlossen)

Dieses Beispiel ähnelt dem vorangegangenem mit dem Unterschied, dass anstelle einer impliziten FB-Instanz ein E/A-Treiber verwendet wird, der den Austausch der Ein- und Ausgänge in SPS-üblicher Art und Weise über das Prozessabbild realisiert.

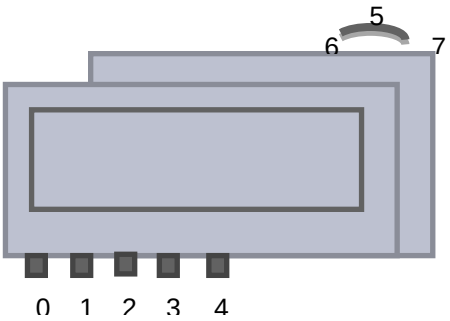
Dieser E/A-Treiber wird in Form der Bibliothek IoDrvPiFace.library im Sourcecode bereitgestellt.

3.7 PiFaceDisplayAndControl.project

(Voraussetzung: Raspberry PiFace Control&Display ist angeschlossen)

Dieses Beispiel zeigt, wie die Inputs und das zweizeilige Textdisplay dieses Geräts verwendet werden können, um Applikationsparameter einzustellen.

Man beachte, dass das PiFace Control&Display über den SPI-Port 1 ('/dev/spidev0.1'), einzustellen im SPI Master, erreicht wird. Durch das Einhängen des Geräts in den Gerätebaum wird eine FB-Instanz angelegt, welche diverse Methoden und Properties zur Ansteuerung des Geräts bereitstellt und den Zustand der Taster zurückgibt. Die Bits 0 bis 7 des Ausgangs bySwitches stehen dabei für folgende Taster:



Im Beispiel wird der FB PiFace_Control_Display an eine Instanz des Bausteins ParamListPiFace übergeben, welcher einen Parametereditor implementiert. Im Ansichtsmodus kann man über den Navigationstaster (6/7) die Parameterliste durchlaufen. Durch Drücken des Navigationstasters (5) sieht man den Parameter der ersten Zeile im Detail. Die Detailansicht verlässt man über Taster 4; durch erneutes Drücken von 5 gelangt man in den Editiermodus, in dem man den Wert im zulässigen Bereich mit den Tastern 0 und 1 verändern kann. Taster 2 speichert den Wert; über Taster 4 verlässt man den Editiermodus.

3.8 I2CExample.project

(Voraussetzung: spezielle Hardware ist über I²C angeschlossen)

Dieses Beispiel zeigt, wie diverse Chips/Breakouts über I²C angeschlossen werden. In diesem Beispiel wurde folgende Hardware angeschaltet:

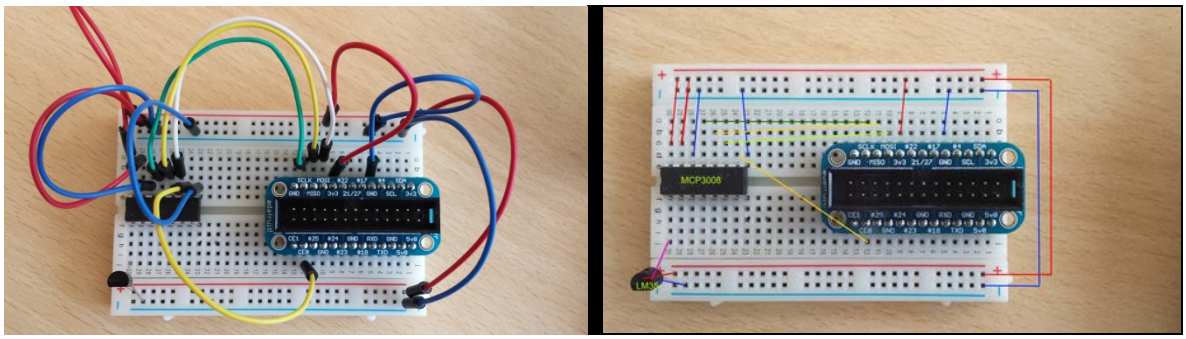
- Adafruit 16-channel/12-Bit PWM
- SRF02 (Ultraschall-Entfernungssensor)
- Drotek IMU 9DOF - MPU9150 (Gyroskop, Beschleunigungsmesser, Magnetoskop)

Die für die Anbindung verantwortlichen Bibliotheken I2C_* werden als Sourcecode zur Verfügung gestellt und können als Beispiel für andere Anschaltungen dienen. Basis für die Kommunikation über I²C ist die Bibliothek RaspberryPiPeripherals, deren Schnittstellen dokumentiert sind.

3.9 MCP3008_Temperature.project

(Voraussetzung: spezielle Hardware ist über SPI angeschlossen)

Dieses Beispiel zeigt, wie ein analoger Temperatursensor (LM35), der an einen A/D-Wandler-Chip (MCP3008) angeschlossen wurde, über SPI ausgelesen werden kann. Der MCP3008 kann 8 analoge Kanäle auswerten, wobei in diesem Beispiel nur einer benutzt wird. Folgender Testaufbau wird verwendet:

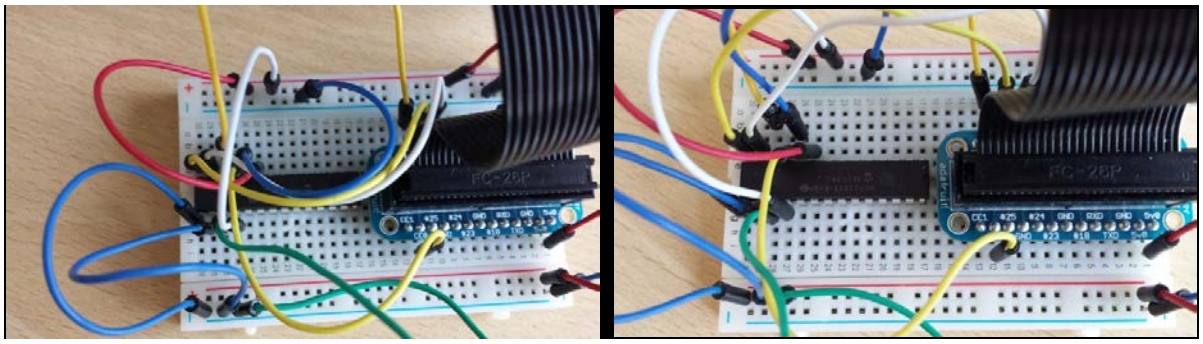


Die für die Anbindung verantwortlichen Bibliotheken SPI_MCP3008.library werden als Sourcecode zur Verfügung gestellt und können als Beispiel für andere Anschaltungen dienen. Basis für die Kommunikation über SPI ist die Bibliothek RaspberryPiPeripherals, deren Schnittstellen dokumentiert sind.

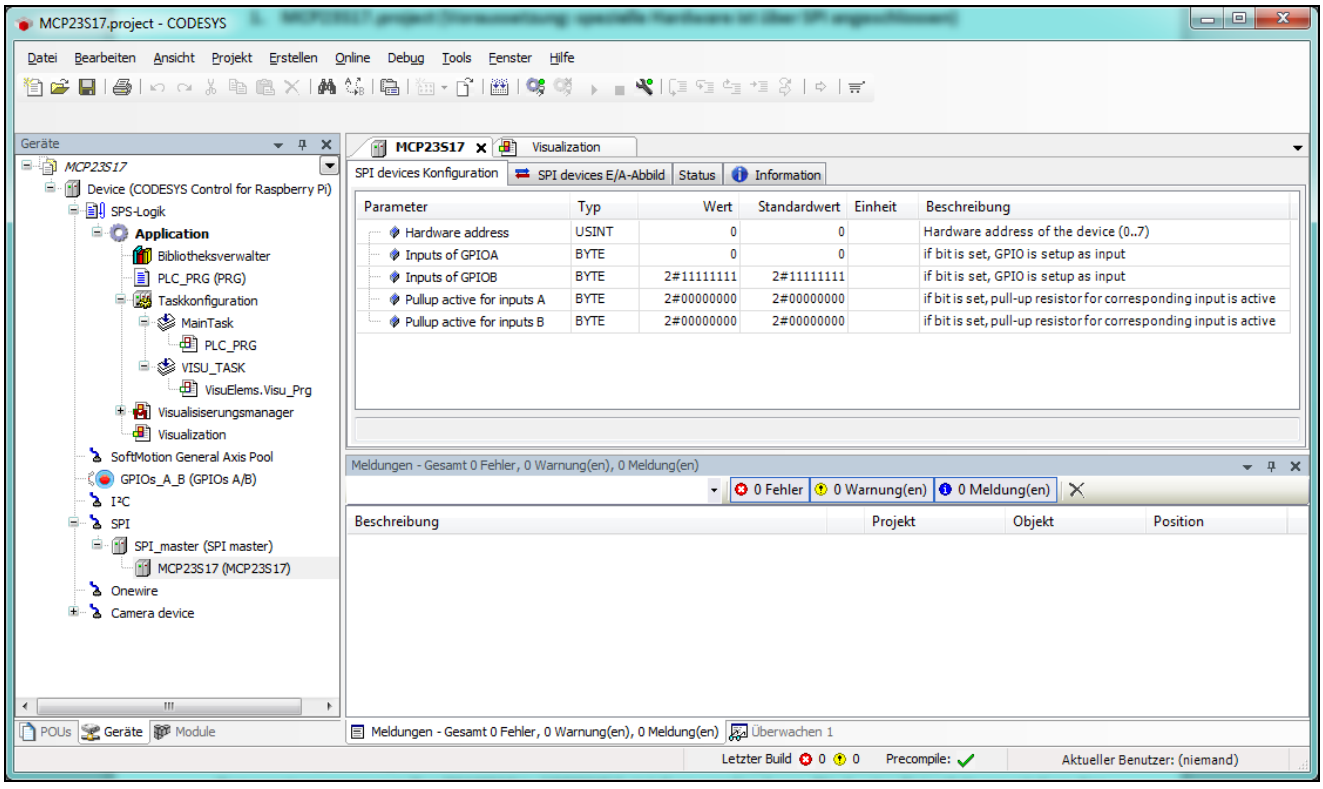
3.10 MCP23S17.project

(Voraussetzung: spezielle Hardware ist über SPI angeschlossen)

Dieses Beispiel zeigt, wie ein Portexpander-Chip (MCP23S17) über spi angebunden werden kann. Die Hardware sollte dazu folgendermaßen verdrahtet sein:



Bei den Geräte-Einstellung kann man auswählen, welche der GPIO-Pins als Input und welche als Output verwendet werden sollen. Für die Inputs kann man zudem einen Pullup-Widerstand aktivieren. Je nach Belegung der Adress-IO-Pins muss die Hardware-Adresse richtig eingestellt sein.



3.11 OneWire.project

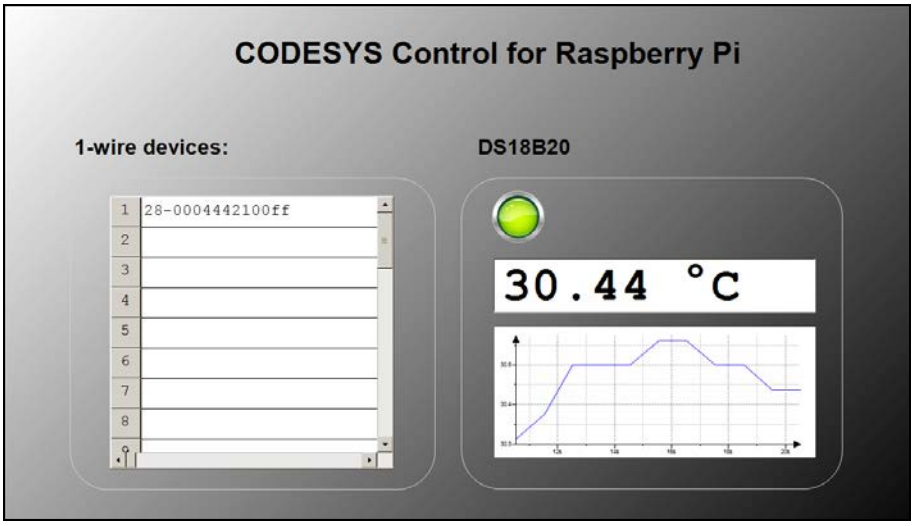
(Voraussetzung: 1-wire Temperatursensor ist angeschlossen)

Dieses Beispiel zeigt, wie das Gerät die über 1-wire angeschlossenen Geräte erkennt und ein Temperatursensor vom Typ DS18B20 in CODESYS betrieben werden kann. Die 1-wire-Datenleitung wird an den GPIO4 angeschlossen.

Jeder Sensor verfügt über eine eindeutige ID, über die er identifiziert und angesprochen wird. Deshalb muss man nach dem Einfügen eines 1-wire-Geräts (z.B. Temperatursensor DS18B20) in dessen Konfiguration die ID setzen.

Um die ID zu erfahren, kann diese Beispielapplikation verwendet werden. Diese zeigt zwei Funktionen:

1. Sie stellt das Ergebnis des Scans in der Visualisierung dar. Dadurch erkennt man die ID der angeschlossenen Geräte.
2. Ist darunter ein DS18B20 und wurde dessen ID richtig konfiguriert, zeigt die Applikation die gemessene Temperatur.



Man beachte, dass der Datenaustausch über 1-wire viel Zeit in Anspruch nimmt, und die Task währenddessen blockiert ist. Bei zeitlich kritischen Anwendungen empfiehlt es sich deshalb, die 1-wire Feldgeräte einer anderen Task zuzuordnen (s. Buszyklus-Optionen im E/A-Abbild-Tab).

3.12 SoftMotion Servo Example

(Voraussetzung: über I²C ist eine Adafruit 16-channel/12Bit PWM-Platine angeschlossen, auf deren ersten PWM-Kanal ein Modellbau-Servo verkabelt ist.)

Dieses Beispiel zeigt, wie das Funktionspaket CODESYS SoftMotion in Verbindung mit Modellbau-Servomotoren verwendet werden kann. Als Kommunikationsinterface wird eine über I²C verbundene Erweiterungsplatine (Adafruit ID 815) verwendet.

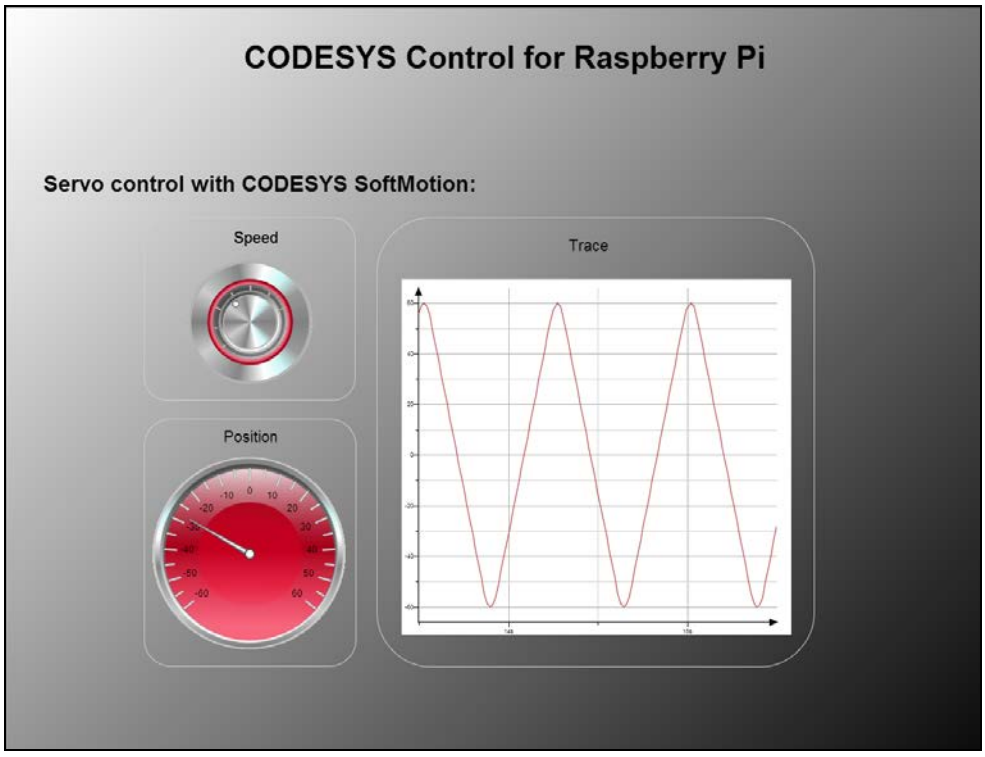
Öffnen Sie das Projekt, laden Sie es auf die Steuerung und starten es. Der Servomotor beginnt, sich stetig von links nach rechts zu drehen. Dafür verantwortlich ist das in SFC programmierte PLC_PRG, welches die Achse zunächst einschaltet und hernach stets zwischen den Positionen -60 und +60, die in der Konfiguration der Achse SM_Drive_Servo als Endlagen angegeben wurden, bewegt.

Die Vorgabe der Position wird dabei über die bei Modellbau-Servos übliche PWM-Schnittstelle transportiert, wobei im festen Takt (default: 50Hz; Parameter des Adafruit PWM Softmotion-Geräts) ein HIGH-Puls gesendet wird, der zwischen 1ms und 2ms lang ist. 1ms steht hierbei für die untere, 2ms für die obere Endlage. Der Bewegungsbereich ist von Motortyp zu Motortyp unterschiedlich. Will man den Motor in Winkelgrad steuern, muss man diesen ausmessen, indem man auf die Endlagen (im Beispiel -60, +60) fährt, und die gemessenen Positionen im Konfigurationsbildschirm einträgt:

SoftMotion Drive: Basic | SM_Drive_Servo: Configuration | SM_Drive_Servo: I/O Mapping | Status | Information

Parameter	Type	Value	Default Value	Unit	Description
AXIS_REF: Standard					
wDriveID	WORD	1	1		ID of drive
bVirtual	BOOL	FALSE	FALSE		drive is simulated
dwRatioTechUnitsDenom	DWORD	1	1		conversion inc./tech.units denominator
iRatioTechUnitsNum	INT	1	1		conversion inc./tech.units numerator
iMovementType	INT	1	1		movement type: 0: rotary/modulo, 1: linear
fPositionPeriod	LREAL	360.0	360.0		modulo value for rotary drives
eRampType	INT	0	0		selects the velocity ramp type used by the FBs
fSWMaxVelocity	LREAL	1e3	1e3		maximum velocity value used for limit check at SMC_ControlAxisByPos
fSWMaxAcceleration	LREAL	1e5	1e5		maximum acceleration value used for limit check at SMC_ControlAxisByPos
fSWMaxDeceleration	LREAL	1e5	1e5		maximum deceleration value used for limit check at SMC_ControlAxisByPos
fRampJerk	LREAL	0	0		jerk used for bringing acceleration to 0 when sin² ramp is interrupted
fSWLimitPositive	LREAL	1000.0	60.0		software limit position in positive direction
fSWLimitNegative	LREAL	0.0	-60.0		software limit position in negative direction
fSWLimitDeceleration	LREAL	0	0		deceleration value used to stop when a software error has been detected
bSWLimitEnable	BOOL	FALSE	TRUE		activate/deactivate software limit
fSWErrorMaxDistance	LREAL	0	0		maximum distance that may be travelled for ramping down after a software error has been detect...
Servo: Configuration					
negative position [units]	LREAL	-60	-60.0		
positive position [units]	LREAL	60	60.0		
startup position [units]	LREAL	0.0	0.0		

Verbinden Sie sich mit einem Browser über <Netzwerk-Adresse>:8080/webvisu.htm auf Ihr Gerät, sehen Sie die erzeugten Sollpositionen und beeinflussen Sie die Geschwindigkeit:



3.13 EtherCAT.project

(Voraussetzung: An den Ethernet-Adapter sind folgende Geräte angeschlossen: Beckhoff EK1100 mit Beckhoff EL2008)

Dieses Beispiel zeigt die Verwendung eines EtherCAT-Felbusgeräts und schaltet die acht vorhandenen Ausgänge.

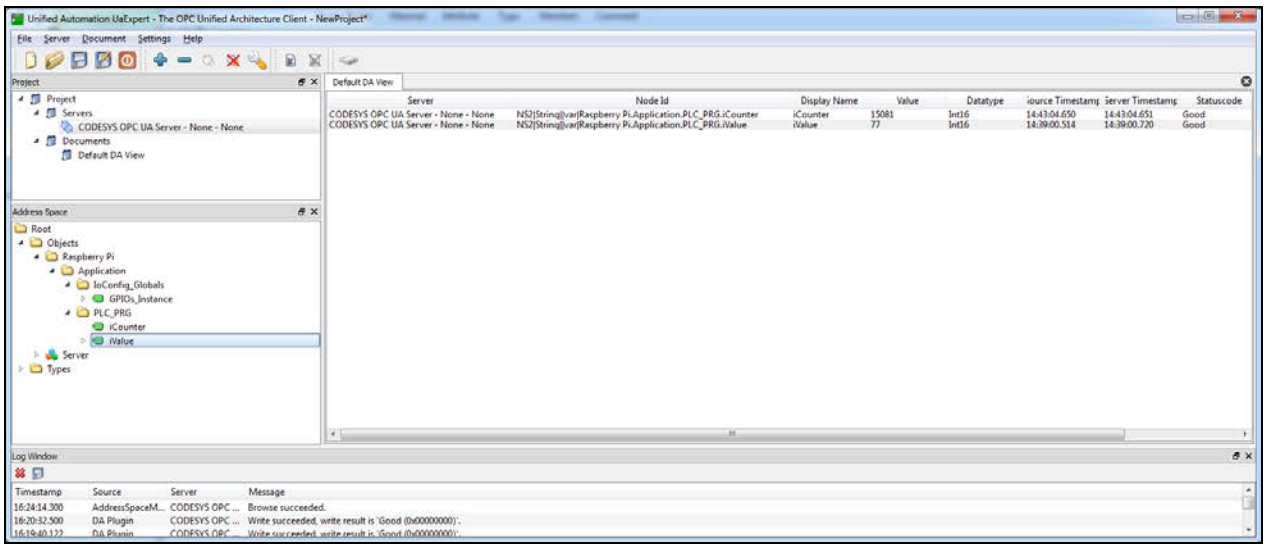
Öffnen Sie das Projekt, laden Sie es auf die Steuerung und starten es. Die Ausgänge der Klemme EL2008 ändern sich daraufhin laufend.

Bitte beachten Sie, dass Sie, wenn Sie den LAN-Port des Raspberry Pi für den EtherCAT-Feldbus verwenden, eine alternative Programmierschnittstelle benötigen. Es empfiehlt sich dafür die Verwendung eines USB-WLAN-Adapters (z.B. Edimax N150).

3.14 OPCUA.project

Dieses einfache Beispiel zeigt, wie sie Variablen Ihrer Applikation für den Zugriff eines OPC/UA-Clients vorbereiten. Dazu werden im Objekt „Symbol Configuration“ die beiden Variablen des Objekts PLC_PRG veröffentlicht.

In einem geeigneten OPC/UA-Client (z.B. Unified Automation UaExpert) können Sie sich zu Ihrem RaspberryPi über die Url „opc.tcp://<Netzwerkadresse>:4840“ verbinden, sehen dessen Objekte und können deren aktuelle Werte anzeigen bzw. – sofern die Konfiguration Schreibrechte erteilt – schreiben:



4 Anschließen weiterer Peripherie über I²C und SPI

Es gibt drei typische Möglichkeiten, zusätzliche Geräte über die Systemschnittstellen i²c, spi oder 1-wire anzuschließen und zu betreiben:

1. Programmierung eines Funktionsblocks (FB) und manuelle Deklaration und Aufruf einer Instanz im Programm
2. Programmierung eines Funktionsblocks (FB) und einer Gerätebeschreibung, um ein spezielles Gerät in den CODESYS-Gerätebaum einhängen und konfigurieren zu können
3. Programmierung eines IO-Treibers

Dabei stellt 1 die einfachste und schnellste Art der Anbindung dar. 2 beschreibt eine besser integrierte, anwenderfreundliche Möglichkeit. 3 richtet sich nach der für SPSen üblichen Methode und kopiert die E/A-Daten in bzw. aus dem Prozessabbild, welches das Mapping der Daten auf bestehende oder neue Variablen erlaubt und die Zykluskonsistenz der Daten übernimmt.

Zur besseren Veranschaulichung vergleichen Sie bitte die Beispielprojekte PiFace_FB (entspricht Variante 1), PiFace (Variante 2) und PiFaceIoDrv (Variante 3), welche alle die SPI-Anbindung des PiFace – Erweiterungsboards implementieren.

Variante 1 kann man mithilfe dieses Beispiels und der in der Bibliothek RaspberryPiPeripherals (s. Bibliotheksmanager) enthaltenen FB-Referenzdokumentation des Bausteins spiMaster (analog i2cMaster) verstanden und auf andere Geräte angewendet werden.

Variante 3 erfordert detailliertere Kenntnisse über CODESYS und das IO-Treiber-Konzept und würde den Rahmen dieses Dokumentes sprengen.

Die für Variante 2 nötigen Schritte werden im Folgenden erklärt:

Für ein neues Gerät sollten Sie eine neue Gerätebeschreibungsdatei (.devdesc.xml) sowie eine neue Bibliothek (.library) anlegen. Wählen Sie als Vorlage dazu ein Gerät, welches die gleiche Kommunikationsschnittstelle verwendet. Sie finden die Beispiele im Dateipfad, der auch die Beispielprojekte enthält. Führen Sie folgende Schritte aus:

(A) Gerätebeschreibung

- Legen Sie eine Kopie einer bestehenden Gerätebeschreibung an. Ändern Sie deren Namen in <myDeviceName>.devdesc.xml.
- Ändern Sie die ID des Geräts, indem Sie das obere Wort auf FFFF, das untere lokal eindeutig wählen:[2]

```
<Device hideInCatalogue="false">
  <DeviceIdentification>
    <Type>501</Type>
    <Id>FFFF 4711</Id>
    <Version>1.0.0.0</Version>
  </DeviceIdentification>
```

- Passen Sie die Geräte-Info an:

```
  <DeviceInfo>
    <Name name="local:ModelName">MCP3008</Name>
    <Description name="local:DeviceDescription">MCP3008</Description>
    <Vendor name="local:VendorName">3S - Smart Software Solutions GmbH</Vendor>
    <OrderNumber>-</OrderNumber>
```

- Tragen Sie den Namen, den Hersteller und die Version der Bibliothek, die Sie für den Treiber bereitstellen werden (s.u.), ein:

```
<RequiredLib libname="Raspberry SPI MCP3008" vendor="3S - Smart Software Solutions GmbH" version="1.0.0.0" identifier="deviceLib">
```

[2] Please note, that if you want to distribute your new device support you need a registered customer ID at 3S - Smart Software Solutions GmbH

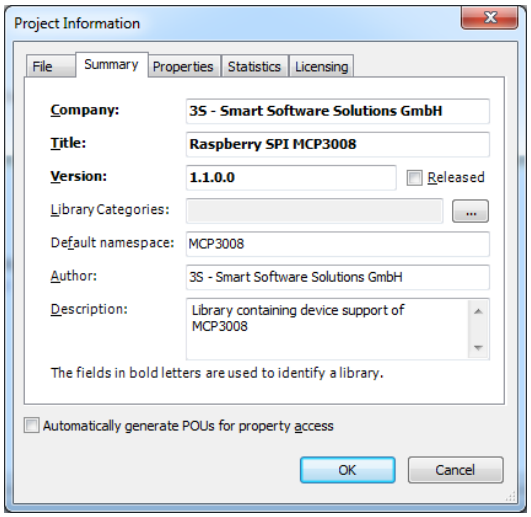
- Tragen Sie den Namen des Bausteins in der Bibliothek, der für die Kommunikation verantwortlich ist, ein:

```
<FBInstance basename="$(DeviceName)" fbname="MCP3008">
  <Initialize methodName="Initialize" />
  <CyclicCall methodname="AfterReadInputs" task="#buscycletask" whentocall="afterReadInputs" />
  <CyclicCall methodname="BeforeWriteOutputs" task="#buscycletask" whentocall="beforeWriteOutputs" />
</FBInstance>
```

- Installieren Sie die Gerätebeschreibung in Ihr Geräte-Repository in CODESYS. Ab jetzt können Sie Ihr Gerät unter den Ordner „I²C devices“ bzw. „SPI devices“ einfügen.

(B) Bibliothek

- Legen Sie eine Kopie einer bestehenden Bibliothek an. Ändern Sie deren Namen in <myDeviceName>.library.
- Öffnen Sie die neue Bibliothek in CODESYS und passen Sie die Projektinformation an:



  Company, Title und Version müssen mit den Einträgen in der Gerätebeschreibung übereinstimmen.
- Benennen Sie den FB in der Bibliothek um. Der neue Name muss mit dem Eintrag in der Gerätebeschreibung übereinstimmen.
- Die Zustandsmaschine programmieren Sie im Rumpf des Bausteins. Dabei steht _iState=0 für den Initialisierungszustand, _iState=10 für den normalen Betrieb, _iState=1000 für einen Fehler. Zwischenzustände können Sie falls nötig verwenden.
- In der Methode AfterReadInputs sollten Sie die Eingangsdaten des Geräts auslesen. Verwenden Sie für die Übermittlung die Basismethoden des FBs i2c (read8, write8, etc.) bzw. spi (transfer).
- In der Methode BeforeWriteOutputs sollten Sie die Ausgangsdaten übertragen.
- Speichern Sie die Bibliothek und installieren Sie sie in das Bibliotheks-Repository.

(C) Verwendung

- Sie können nun in ein neues Projekt unter dem entsprechenden Kommunikationsbus Ihr neues Gerät einfügen. Dadurch wird eine FB-Instanz erzeugt, die Sie in Ihrer Applikation verwenden können.

5 Screenshots