



CODESYS Control for Raspberry Pi (Demo)

General information

Order number: 000051	Supplier information 3S-Smart Software Solutions GmbH Memminger Straße 151 87439 Kempten Germany Support: Phone: +49 831 54031 66 support@codesys.com
Version: 1.0.0.0	
Short description CODESYS Control for Raspberry Pi (Demo)	

Requirements and restrictions

Programming system	CODESYS Development System Version 3.5.4.0 or higher
Target system	CODESYS Control Version 3.5.4.0 (this is part of this product)
Supported Platforms/ Devices	Raspberry Pi (s. http://www.raspberrypi.org/)
Additional requirements	SD-card (minimum 4GB)
Restrictions	Runtime limitation (2 hours)

Price

This product is for free.

Required accessory, purchasable in the CODESYS Store:

-

Product description

This product contains a runtime limited CODESYS Control application for Raspberry Pi (see <http://www.raspberrypi.org/>) as well as driver support for the extension modules Raspberry PiFace Digital, Raspberry Pi Camera and several devices/breakouts with I²C communication interface.

This product is delivered as SD card image based on a Linux distribution (Raspbian). This image must be copied onto the memory card of the device. After reboot CODESYS Control is started automatically and shuts down after 2 hours.

The runtime system does not have realtime behavior. Its Jitter depends on many factors, especially on parallelly executed Linux applications, and is ideally about 50µs with maximum values of 400µs.

This product supports the following functionalities

- EtherCAT Master
- Modbus TCP Master and Slave
- Ethernet/IP Scanner
- Web Visualization
- SoftMotion CNC

This product consists of:

- SD card image for operating system and CODESYS Control
- CODESYS device description files for Raspberry Pi, Raspberry PiFace Digital, Raspberry Pi Camera, several devices/breakouts with I²C communication interface (SRF02, Adafruit PWM, MPU6050, MPU9150, AK8975)

This product offers amongst other things the possibility to plug and control additional devices via SPI or I²C.

This product is provided for testing and training purpose and may not be used in the field.

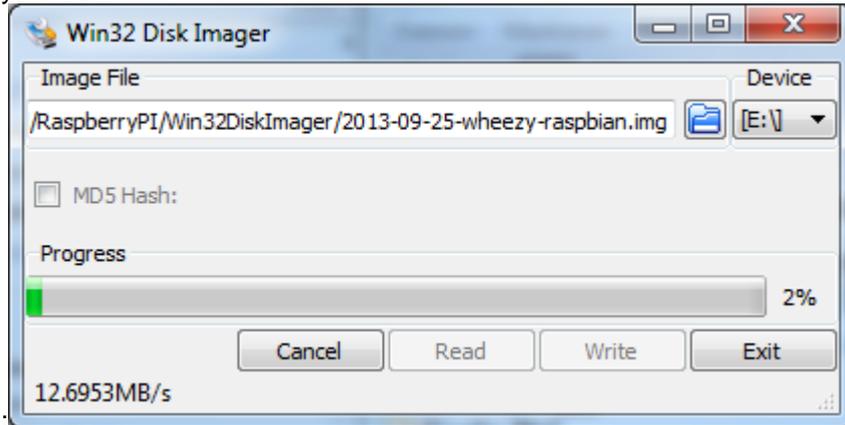
Technical description

Preparations

1. Download and unzip the product zip file from CODESYS Store
2. Install the included package file inside CODESYS (Menu Tools -> Package Manager -> Install).
Note down the installation path.
3. Generate a bootable SD card

Under Windows, the application *Win32diskimager* (download from <http://sourceforge.net/projects/win32diskimager/>) can be used to download the delivered image onto

your SD card:

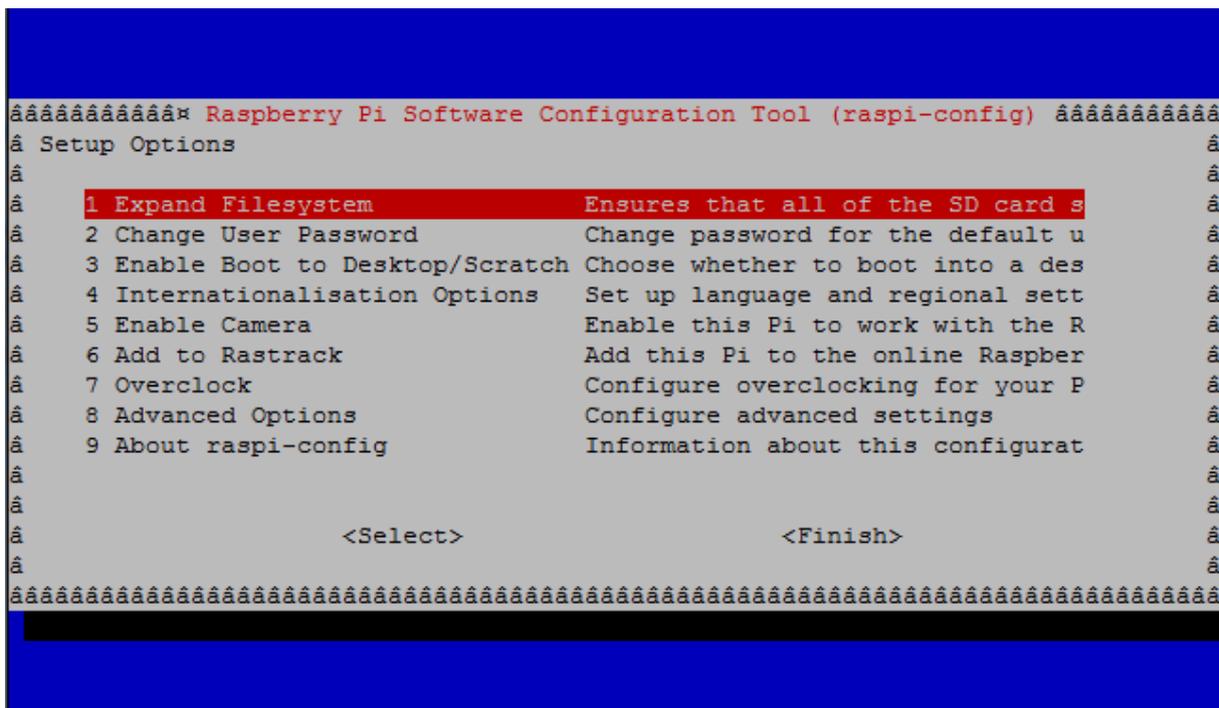


Reboot Raspberry Pi with this SD card afterwards. In order to obtain a valid IP address, the device should be connected to a network with a DHCP server. To find out this address, you may either plug a keyboard, mouse and a monitor to your device, log in (User: pi, Passwort: raspberry), open a console and issue the command `ifconfig`.

Alternatively, you can start CODESYS and proceed like explained in the first example application (see below), without logging into the device and starting the application. Knowing the IP address you can also log in with Windows remote desktop.

4. Expand the file system according to the size of your SD card

From a console run `sudo raspi-config` and execute the following action:



5. After a reboot of your device it is ready for use.

Example applications

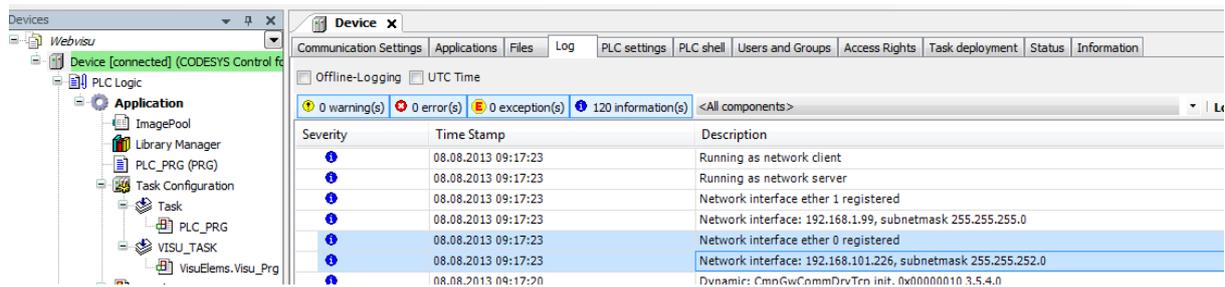
In the installation path (that you have noted down during installation of the package) you will find the following example projects:

1. Webvisu.project

This example shows how CODESYS web visualization can be used. Open the project and download it onto your Raspberry Pi by doubleclicking on the node "Device" in the device tree (left hand side). Then press "Scan network" in the communication dialog tab and select your device that should now appear under the name "RaspberryPi", if the device is in the same network with your programming PC. Select it and run "log in" from the menu "Online". Then start the application with F5.

Start an internet browser (possibly also on your smartphone) and connect to <Netzwerk-Adresse>:8080/webvisu.htm.

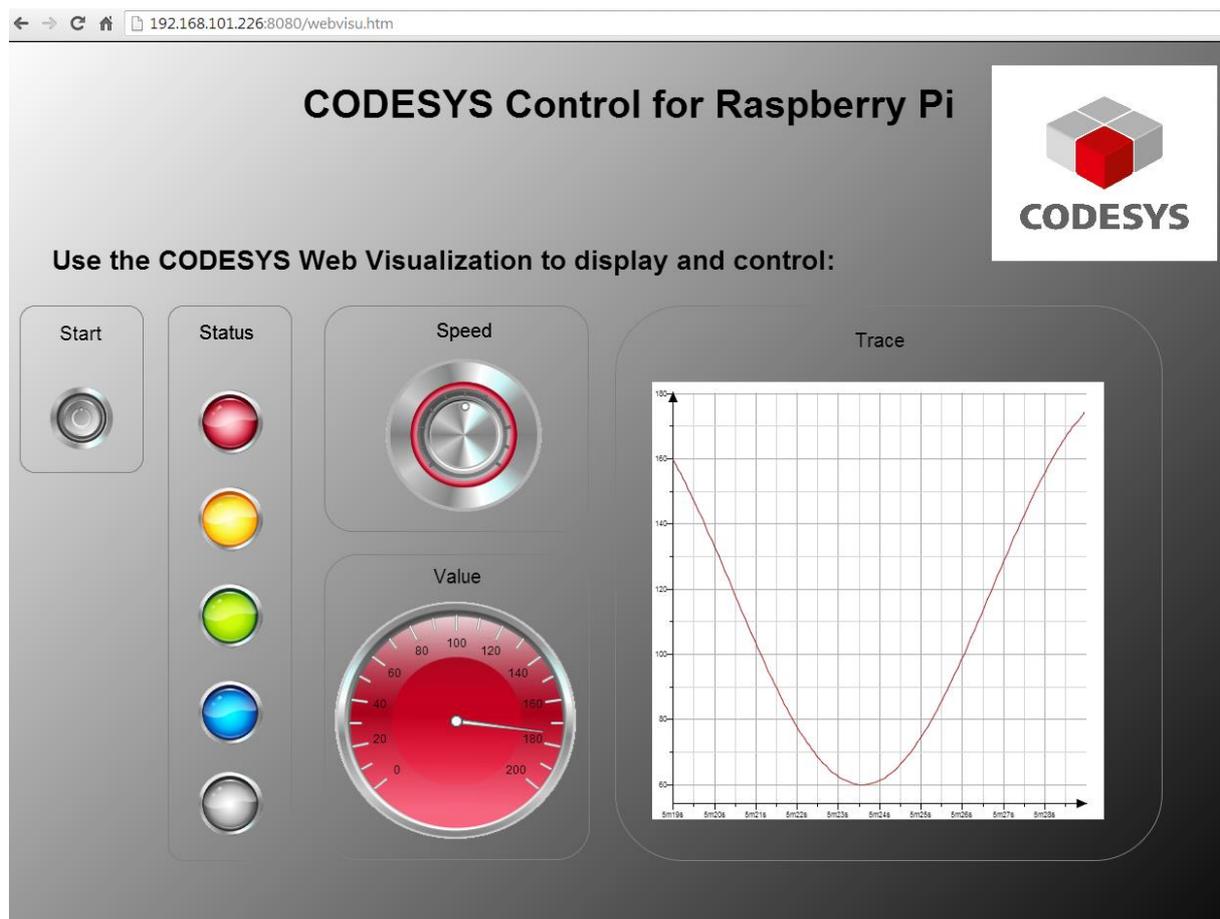
You can find out the IP address of your device from the log outputs on your device:



The screenshot shows the CODESYS Log window with a table of log entries. The table has columns for Severity, Time Stamp, and Description. The following table represents the data shown in the screenshot:

Severity	Time Stamp	Description
Information	08.08.2013 09:17:23	Running as network client
Information	08.08.2013 09:17:23	Running as network server
Information	08.08.2013 09:17:23	Network interface ether 1 registered
Information	08.08.2013 09:17:23	Network interface: 192.168.1.99, subnetmask 255.255.255.0
Information	08.08.2013 09:17:23	Network interface ether 0 registered
Information	08.08.2013 09:17:23	Network interface: 192.168.101.226, subnetmask 255.255.252.0
Information	08.08.2013 09:17:20	Dynamic: CmnGwCommDrvTrn init: 0x00000010 3.5.4.0

In your browser you will see the visualization that has been designed in the project:



The screenshot shows the CODESYS Control for Raspberry Pi web visualization interface. The interface features a dark grey background with a white header area containing the CODESYS logo and the text "CODESYS Control for Raspberry Pi". Below the header, there is a section titled "Use the CODESYS Web Visualization to display and control:" followed by several control elements:

- Start:** A circular button with a grey gradient.
- Status:** A vertical column of five circular buttons with red, yellow, green, blue, and grey gradients.
- Speed:** A circular speedometer with a red needle and a scale from 0 to 180.
- Value:** A circular gauge with a red needle and a scale from 0 to 200.
- Trace:** A line graph showing a parabolic curve on a grid. The y-axis ranges from 60 to 180, and the x-axis ranges from 8m198 to 8m238.

2. Camera.project (precondition: Raspberry Pi Camera is connected)

This project shows the usage of Raspberry Pi Camera (extension hardware) from your PLC:

Download the application to your controller, start the program and set the variable `xTakePicture` to `TRUE`. The camera will take a picture and store it in the local file system with the name „Picture.jpg“.

You can copy this file onto your programming PC with the help of the file dialog (doubleclick “Device”, tab “Files”, Button “Update” on the right side etc.).

3. PiFace.project (precondition: Raspberry PiFace Digital is connected)

Preparation: Please make sure that the system file `/etc/modules` on your RaspberryPi contains the line „`spi-bcm2708`“. This is necessary to activate the access to the SPI driver.

This example shows the usage of Raspberry PiFace Digital (8 digital inputs and outputs).

Open the project, download it to the controller and start it. The simple application in `PLC_PRG` controls the relay output K0 depending on the key button S1 (on-switching of K0 is delayed for 1s) and the relay output K1 depending on button S2 (off-switching is delayed for 500ms).

Please note that this driver allows to connect more than one PiFace devices (hardware address is set with the jumpers JP1, JP2) by setting the corresponding parameter in the PiFace device in the device tree.

The library `SPI_PiFace` that operates as driver is provided as source code and can be seen as example how to communicate to other devices via SPI. The communication bases are on the library `RaspberryPiPeripherals`, for which a reference documentation is provided (see online help (F1) -> Libraries).

4. PiFaceloDrv.project (precondition: Raspberry PiFace Digital is connected)

This example is similar to the one before. Instead of providing the I/O data as inputs of an automatically instanced function block, the example implements the data exchange as standard PLC IO driver via a process image, how it is typically done on PLCs.

The driver library `IoDrvPiFace.library` is provided as source code.

5. I2CExample.project (precondition: special hardware is connected via I²C)

Preparation: Please make sure that the system file `/etc/modules` on your RaspberryPi contains the line „`i2c-dev`“. This is necessary to activate the access to the I²C driver.

This example communicates with the following breakouts via I²C:

- Adafruit 16-channel/12-Bit PWM
- SRF02 (supersonic distance sensor)
- Drotek IMU 9DOF - MPU9150 (gyroskope, accelerometer, compass)

The libraries `I2C_*` that implement the data exchange are provided in source code and can be used as example for additional interface connections. The communication bases are on the library `RaspberryPiPeripherals`, for which a reference documentation is provided (see online help (F1) -> Libraries).

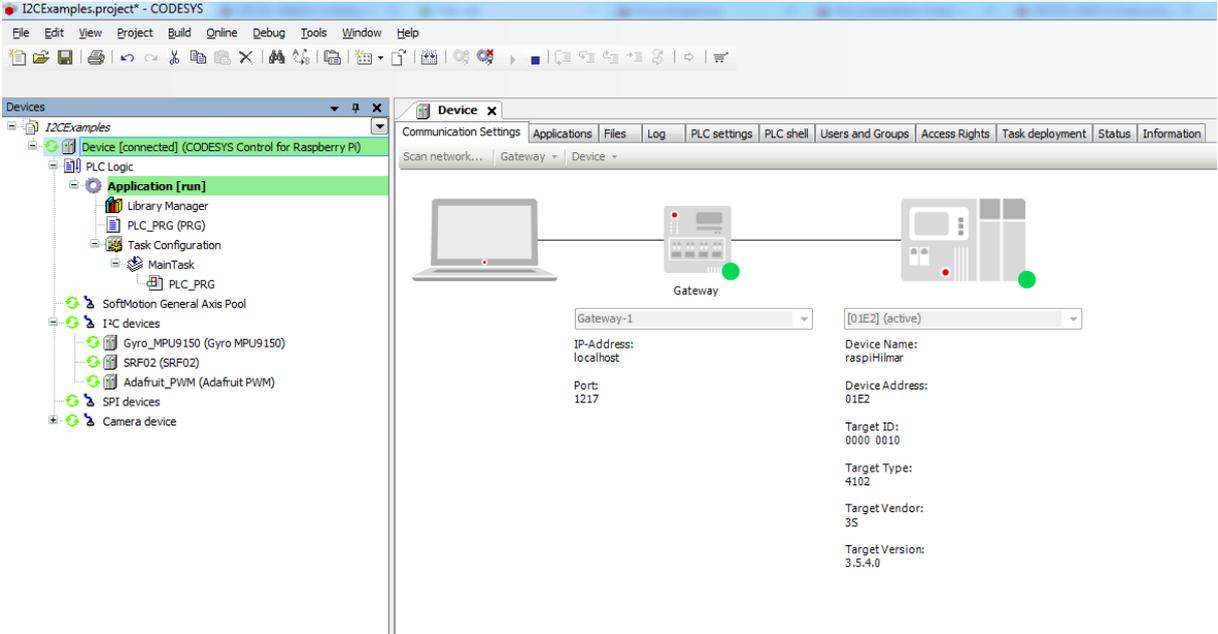
6. EtherCAT.project (precondition: The following devices are connected to the Pi's LAN adapter: Beckhoff EK1100 mit Beckhoff EL2008)

This example switches the eight available outputs on the connected hardware implementing an EtherCAT master.

Open the project, download and start it. The outputs of the clamp will change continuously.

Please note that the LAN port of your Raspberry Pi is then blocked for EtherCAT and hence cannot be used as communication and programming interface. It is recommended to use a USB WLAN adapter (e.g. Edimax N150) in this case.

Screenshots



CODESYS Control for Raspberry Pi



Use the CODESYS Web Visualization to display and control:

The visualization interface is divided into four main sections:

- Start:** A single circular button with a metallic texture and a central dot.
- Status:** A vertical column of five circular indicators. From top to bottom, they are red, yellow, green, blue, and grey.
- Speed:** A circular speedometer with a silver outer ring and a red inner ring. The needle is currently pointing to approximately 100.
- Value:** A circular gauge with a red face and a white needle. The scale ranges from 0 to 200 with major ticks every 20 units. The needle is pointing to approximately 180.
- Trace:** A line graph showing a parabolic curve. The y-axis ranges from 60 to 180, and the x-axis shows time from 8m19s to 8m28s. The curve starts at (8m19s, 180), reaches a minimum of approximately 60 at 8m24s, and ends at (8m28s, 180).