CODESYS Control for BeagleBone SL

Getting Started

Version: 2.0
Template: templ_tecdoc_de_V1.0.docx
File name: CODESYS_Control_BBB_SL_First_Steps_DE.doc

© 3S-Smart Software Solutions GmbH

CONTENTS

Template: templ_tecdoc_de_V1.0.docx

© 3S-Smart Software Solutions GmbH

1 Product description

This product contains a CODESYS Control runtime system for BeagleBone Black SL (BBB). The recommended operating system "Debian" can be downloaded from the following website: http://beagleboard.org/latest-images.

You install this product from the CODESYS update manager included in the Debian distribution of Linux. After CODESYS Control for BeagleBone SL is started without a valid license, it will run for two hours with full functionality and it will stop automatically after the time has elapsed. Then you can start the runtime environment again manually.

The runtime system does not have real-time properties. Jitter depends on various factors, such as Linux applications running at the same time. Ideally it is approximately 50 µs with maximum values around 400 µs.

This product consists of the following items:

- Debian package with CODESYS Control for BeagleBone SL
- CODESYS plug-in for software updates
- CODESYS device descriptions for BeagleBone Black SL

The included CODESYS runtime system supports the following functions:

- CODESYS EtherCAT master
- CODESYS Modbus TCP master/slave
- CODESYS Modbus RTU master/slave
  (Required: Physical support for RS232/RS485 with control of the sender/receiver switching)
- CODESYS WebVisu
- CODESYS SoftMotion CNC+Robotics
- CODESYS OPC UA Server
- CANopen via EL6751 Gateway
- CODESYS EtherNet/IP scanner/adapter

© 3S-Smart Software Solutions GmbH

2 Installation, configuration, and licensing

Start CODESYS and the package manager located in the *Tools* menu. Install the package that you have downloaded from the CODESYS Store. In order to install CODESYS Control onto a BeagleBone Black device, you need a functioning Debian operating system. If you do not have this, then follow the steps described below in *Preparatory work*. Otherwise, continue to *Installation*.

2.1 Preparatory work

Follow the steps as described at http://beagleboard.org/getting-started to install a functioning operating system on the SD card of your device. We recommend that you use Debian.

Then start BeagleBone Black with this SD card. In order to be assigned a valid IP address, the device should be connected to a network with a DHCP Server.

To see the network address: connect a keyboard, mouse, and monitor to your device; log in (user: `root`, no password); open a console; and execute the `ifconfig` command. Optionally, you can also `ping` the network name of the device from the Windows console in order to see the IP address.

In CODESYS, you can also click *Update BeagleBone Black* in the *Tools* menu and then click *Scan* in the update dialog. In this way, you see a list of IP addresses of BeagleBone Black devices that are available in the network.

With an IP address, you can log in (user: `root`, no password) to a device via SSH by means of a tool, such as Putty (http://www.putty.org/).

Additional notes:

- If the BeagleBone Black is connected with a USB cable directly to your local system, then the default IP address is `192.168.7.2`.
- “CPUFreq Governor”: Provided by Linux as a frequency scaling mechanism. Frequency scaling allows for changing the clock speed of the CPU at runtime. This is a very good method for saving battery power, as the lower the CPU clock, the lower the CPU usage (from https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt).

  There are different kinds of governors; however, *ondemand* is used by default. For improved performance and decreased jitter, the governor should be set to *performance* for achieving static clocking. Set the performance of the governors easily with the following command when you are logged in to the shell of the BeagleBone Black:

  ```
  echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
  ```

  Read the governor currently in use with the command:

  ```
  cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
  ```

  Read the current CPU speed with the command:

  ```
  cat /sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_cur_freq
  ```

  After setting the performance governor, the CPU speed should be set to 1 GHz .

2.2 Installation

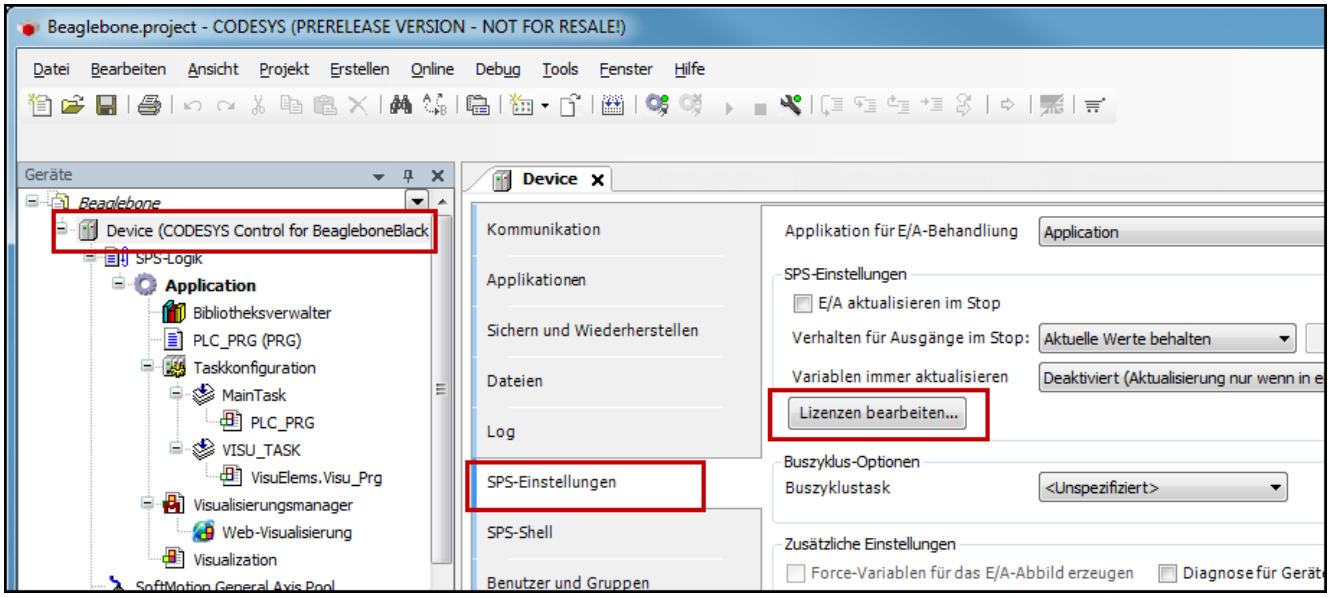
1. In CODESYS, click *Update BeagleBone Black* in the *Tools* menu.
   a. Select a version.
   b. Specify the correct login credentials (user: `root`, no password).
   c. Select the IP address of your device.
   d. Click OK and check in the messages view that the runtime system has been installed successfully.
2. After you restart the device, it will be ready for operation.

© 3S-Smart Software Solutions GmbH

2.3 Licensing in the CODESYS Development System

Requirements: PC with CODESYS Development System, Internet access, and a connected BeagleBone Black.

Licensing is performed on a PC or laptop with the CODESYS Development System and a connected BeagleBone Black. You can edit the license entries by double-clicking the controller in *PLC settings* / Edit licenses.



Licenses are activated in *Install licenses / Activate license* by specifying the ticket number and transferring the license to the CODESYS Software Key (soft container).

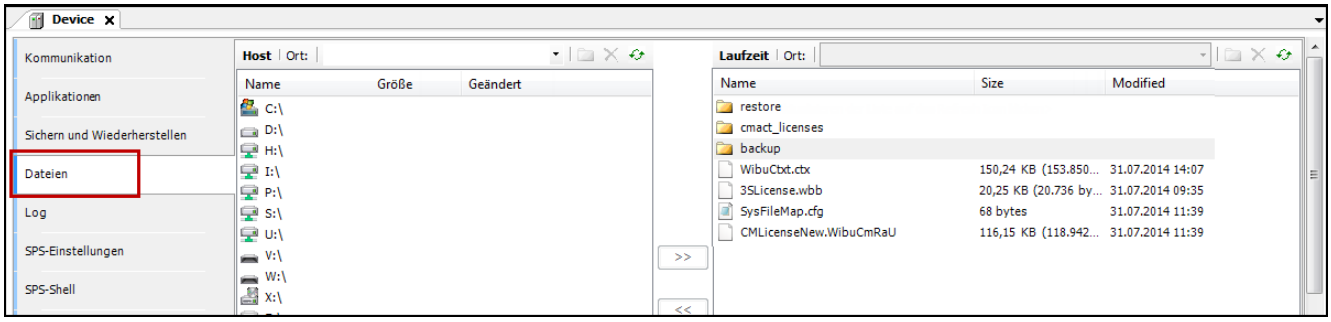
Note: License activation locks the license to the BeagleBone Black. Reinstallation is possible only on the same device.

2.4 Backing up licenses

Some events, such as a sudden loss of power, can corrupt the file system on the BeagleBone Black. To back up a license, we recommend the following procedure:

1. Activate the license (see description above).
2. Reboot the BeagleBone Black.
3. Back up the license file to an external drive.

To back up the license file, switch to the *backup* folder on the BeagleBone Black. This folder is accessible from CODESYS by double-clicking the device and selecting the *Files* tab.



Back up the contents of the folder (`3SLicenseInfo.tar`) to an external drive.

2.5 Reactivating licenses

To reactivate a license, the backed up license file must have been copied to the "restore" folder. Then the system must be rebooted.

© 3S-Smart Software Solutions GmbH

3 Configuring the CAN/serial cape

3.1 Introduction

This document describes the steps required for configuring a BeagleBone Black that will access the serial cape and CAN cape from Logic Supply. These instructions were written for kernel version 4.1. Some steps may vary slightly for other kernel versions.

Note: You can find your kernel version (e.g. connected by SSH) with the command:

```
uname –a
```

3.2 Inserting the cape

Insert the appropriate cape as described in the user manual on the BeagleBone Black.

After the cape in inserted, power up the BeagleBone Black and check the connection with the command:

```
cat /sys/devices/platform/bone_capemgr/slots
```

You should receive the following information (example):

```
0: P---L-   0 cape-CBB-Serial,r01,Logic Supply,cape-CBB-Serial
1: PF----  -1
2: PF----  -1
3: PF----  -1
```

You can also check the kernel output:

```
dmesg | grep cape
```

The output should look like this:

```
…
[3.719868] bone_capemgr bone_capemgr: Baseboard: 'A335BNLT,000B,1614BBBK0256'
[3.719904] bone_capemgr bone_capemgr: compatible-baseboard=ti,beaglebone-black - #slots=4
[3.762237] bone_capemgr bone_capemgr: slot #0: 'cape-CBB-Serial,r01,Logic Supply,cape-CBB-Serial'
[3.818178] bone_capemgr bone_capemgr: slot #1: No cape found
[3.878171] bone_capemgr bone_capemgr: slot #2: No cape found
[3.938165] bone_capemgr bone_capemgr: slot #3: No cape found
[3.944194] bone_capemgr bone_capemgr: initialized OK.
[3.963568] bone_capemgr bone_capemgr: slot #0: dtbo 'cape-CBB-Serial-r01.dtbo' loaded; overlay id
#0
…
```

This shows you that a cape has been found and a respective device tree overlay is loaded.

© 3S-Smart Software Solutions GmbH

3.3 CAN interface

3.3.1 CAN configuration

A BeagleBone Black combined with DCAN0 would clash with the I2C-2 functionality. Therefore, DCAN1 is normally used with P9 pin24 (Rx) and P9 pin26 (Tx). To configure and test the CAN manually, you can use the commands:

```
sudo modprobe can
sudo modprobe can-dev
sudo modprobe can-raw
```

You also use these commands to load the required kernel modules.

Now you should be able to use the standard Linux mechanism *socketcan*. By default, this is supported by CODESYS Control for BeagleBone Black.

Set the baud rate:

```
sudo ip link set can0 up type can bitrate 125000
```

Activate the interface:

```
sudo ifconfig can0 up
```

3.3.2 Testing CAN

To test the CAN directly from the command line, you can use *can-utils* for the Debian package:

Installation:

```
sudo apt-get install can-utils
```

Sending a message:

```
cansend can0 5A1#ABCDABCD
```

Receiving a message:

```
candump can0
```

3.4 Serial

3.4.1 Configuring UARTs

UARTs on this cape (UART2 and UART4) can be accessed from the devices `/dev/ttyO2` and `/dev/ttyO4`. Refer to the respective Linux documentation if you want to access the serial port yourself (outside of CODESYS). For access to UART from CODESYS, specify the SysCom configuration in the file CODESYSControl.cfg.

```
[SysCom]
Linux.Devicefile=/dev/ttyO
```

3.4.2 Testing UARTs

Execute the following Python code in an interactive Python shell, or create a short text file with the extension `.py` and execute the file with the Python interpreter:

```
import serial
import time
uart2_file='/dev/ttyO2'
baud=115200
ser = serial.Serial(uart2_file, baud)
while True:
    ser.write("Testing")
    time.sleep(1)
```

This opens UART2 with a baud rate of 115200 and sends the character string "Testing" every second. Then you can connect a serial cable and test the output on your PC (e.g. with Putty, http://www.putty.org/).

© 3S-Smart Software Solutions GmbH

3.5 Known issues

3.5.1 Serial RS485

When using RS485 of the specified CAN/serial cape, there is a problem with the RS485 transceiver. As the Linux driver cannot toggle between send and receive modes, applications may not function as expected.

Specifically, the OMAP serial driver of the BeagleBone Black does not support the control necessary for toggling between send and receive (DE/RE and RTS). Instead, it has an RS485 mode that uses a dedicated assigned GPIO pin for controlling the DE/RE lines. It also has a modified structure (serial_rs485 struct) as compared to the standard serial driver. You must use `ioctl` on the respective `devtty` file in order to set the serial driver in RS-485 mode and tell it which pin it has to use. You also have to take care of the multiplexing of the pin in GPIO mode.

© 3S-Smart Software Solutions GmbH

4 Using GPIOs and analog inputs

4.1 Introduction

This section describes the restrictions when using the standard image of a BeagleBone Black for accessing the GPIOs and analog inputs.

4.2 Pin access to P8/P9

The P8 and P9 pins allow for access to many GPIOs and the analog inputs. However, not all can be use without restriction by default. The standard HDMI and base device tree slots use some pins for HDMI, eMMC, and MCASP0. The following list shows which pins can be used as GPIOs and analog inputs, and which are already in use by the system.

Header pin	Pin name	Supported by CODESYS	Used by system
P9_01	*GND*	*NO*	*NO*
P9_02	*GND*	*NO*	*NO*
P9_03	*DC_3.3V*	*NO*	*NO*
P9_04	*DC_3.3V*	*NO*	*NO*
P9_05	*VDD_5V*	*NO*	*NO*
P9_06	*VDD_5V*	*NO*	*NO*
P9_07	*SYS_5V*	*NO*	*NO*
P9_08	*SYS_5V*	*NO*	*NO*
P9_09	*PWR_BUT*	*NO*	*NO*
P9_10	*SYS_RESETn*	*NO*	*NO*
P9_11	UART4_RXD	YES	NO
P9_12	GPIO1_28	YES	NO
P9_13	UART4_TXD	YES	NO
P9_14	EHRPWM1A	YES	NO
P9_15	GPIO1_16	YES	NO
P9_16	EHRPWM1B	YES	NO
P9_17	I2C1_SCL	NO	PINMUX I²C
P9_18	I2C1_SDA	NO	PINMUX I²C
P9_19	I2C2_SCL	NO	PINMUX I²C
P9_20	I2C2_SDA	NO	PINMUX I²C
P9_21	UART2_TXD	YES	NO
P9_22	UART2_RXD	YES	NO
P9_23	GPIO1_17	YES	NO
P9_24	UART1_TXD	YES	NO
P9_25	GPIO3_21	NO	MCASP0
P9_26	UART1_RXD	YES	NO
P9_27	GPIO3_19	YES	NO
P9_28	SPI1_CS0	NO	MCASP0
P9_29	SPI1_D0	NO	MCASP0
P9_30	SPI1_D1	NO	MCASP0
P9_31	SPI1_SCLK	NO	MCASP0
P9_32	*VADC*	*NO*	*NO*
P9_33	AIN4	YES	NO
P9_34	*AGND*	*NO*	*NO*

© 3S-Smart Software Solutions GmbH

Header pin	Pin name	Supported by CODESYS	Used by system
P9_35	AIN6	YES	NO
P9_36	AIN5	YES	NO
P9_37	AIN2	YES	NO
P9_38	AIN3	YES	NO
P9_39	AIN0	YES	NO
P9_40	AIN1	YES	NO
P9_41	GPIO0_20	YES	NO
P9_42	GPIO0_7	YES	NO
P9_43	*GND*	*NO*	*NO*
P9_44	*GND*	*NO*	*NO*
P9_45	*GND*	*NO*	*NO*
P9_46	*GND*	*NO*	*NO*
P8_01	*DGND*	*NO*	*NO*
P8_02	*DGND*	*NO*	*NO*
P8_03	GPIO1_6	NO	eMMC
P8_04	GPIO1_7	NO	eMMC
P8_05	GPIO1_2	NO	eMMC
P8_06	GPIO1_3	NO	eMMC
P8_07	TIMER4	YES	NO
P8_08	TIMER7	YES	NO
P8_09	TIMER5	YES	NO
P8_10	TIMER6	YES	NO
P8_11	GPIO1_13	YES	NO
P8_12	GPIO1_12	YES	NO
P8_13	EHRPWM2B	YES	NO
P8_14	GPIO0_26	YES	NO
P8_15	GPIO1_15	YES	NO
P8_16	GPIO1_14	YES	NO
P8_17	GPIO0_27	YES	NO
P8_18	GPIO2_1	YES	NO
P8_19	EHRPWM2A	YES	NO
P8_20	GPIO1_31	NO	eMMC
P8_21	GPIO1_30	NO	eMMC
P8_22	GPIO1_5	NO	eMMC
P8_23	GPIO1_4	NO	eMMC
P8_24	GPIO1_1	NO	eMMC
P8_25	GPIO1_0	NO	eMMC
P8_26	GPIO1_29	YES	NO
P8_27	GPIO2_22	NO	HDMI
P8_28	GPIO2_24	NO	HDMI
P8_29	GPIO2_23	NO	HDMI
P8_30	GPIO2_25	NO	HDMI
P8_31	UART5_CTSN	NO	HDMI
P8_32	UART5_RTSN	NO	HDMI
P8_33	UART4_RTSN	NO	HDMI

© 3S-Smart Software Solutions GmbH

Header pin	Pin name	Supported by CODESYS	Used by system
P8_34	UART3_RTSN	NO	HDMI
P8_35	UART4_CTSN	NO	HDMI
P8_36	UART3_CTSN	NO	HDMI
P8_37	UART5_TXD	NO	HDMI
P8_38	UART5_RXD	NO	HDMI
P8_39	GPIO2_12	NO	HDMI
P8_40	GPIO2_13	NO	HDMI
P8_41	GPIO2_10	NO	HDMI
P8_42	GPIO2_11	NO	HDMI
P8_43	GPIO2_8	NO	HDMI
P8_44	GPIO2_9	NO	HDMI
P8_45	GPIO2_6	NO	HDMI
P8_46	GPIO2_7	NO	HDMI

Note: Please be aware of the different voltages and currents allowed the for inputs and outputs.

© 3S-Smart Software Solutions GmbH

5 Configuring external storage devices

5.1 Introduction

This section describes the steps necessary for configuring a BeagleBone Black so that it is possible to access a USB key or a µSD card from CODESYS Control for BeagleBone SL. The compatibility of specific USB keys and µSD cards depends on the hardware. However, almost every USB key and µSD card works with the BeagleBone Black.

5.2 Accessing USB keys and µSD cards

5.2.1 Configuring Linux for automount

Debian provides a package that is responsible for automounting plug-on storage devices. This package can also be installed on your BeagleBone Black.

Installing *usbmount* from the Debian directory:

```
command>  sudo apt-get install usbmount
```

Note: The device must be connected to the Internet and it must be configured so that it can access the Debian server.

5.2.2 Accessing USB keys

If a USB key is connected, *usbautomount* creates the folder */media/usb* automatically for your to access the files on the USB key. The CODESYS Control runtime system is configured for displaying this kind of USB memory. When a USB key is connected, it is set up at ”`/media/usb`”. This folder can be accessed from CODESYS by means of the placeholder `$USB$`.

You can access the placeholder by means of file transfer in the CODESYS Development System:

- Double-click the PLC in the device tree.
- Select the "Files" tab.
- Click the "Refresh" button.

You can access the same files from your IEC application for using the "CAA_File" library. But please note that the files may not be accessible because the media is pluggable. Therefore, check the result of `CAAFileOpen`.

© 3S-Smart Software Solutions GmbH