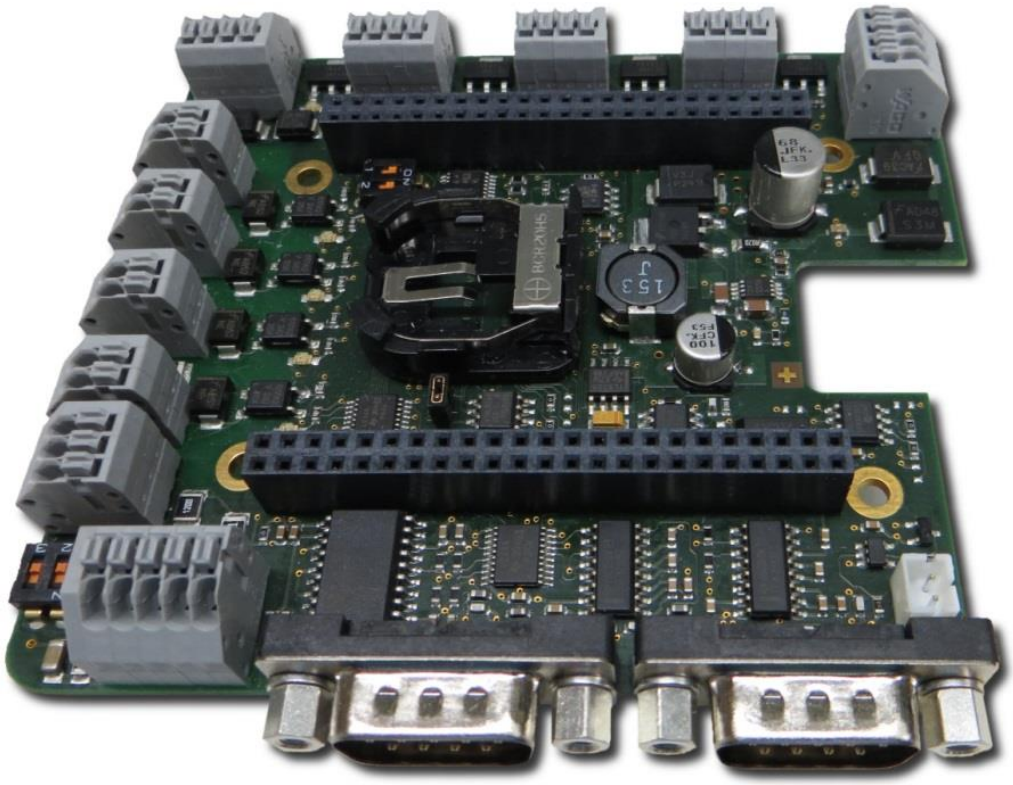
qCAPE-COM1 for Beaglebone Black
User Manual



1. Document revisions

Rev	Date	Author	Remark
0.0.1	08/19/2014	Claus Gindhart	Draft
1.0.0	08/20/2014	Claus Gindhart	Initial release version after proof reading by SKO
1.0.1	09/17/2014	Claus Gindhart	Added Support for Debian Linux and Beaglebone Rev C
1.0.2	09/30/2014	S. Kostelidis	Added new block diagram; Format changes
1.0.3	10/30/2015	Nadja Stärk	Corrected Reading of GPIOs
1.0.4	09/11/2016	Nadja Stärk	Adding instructions for using Debian package with all files for testing
1.0.5	05/02/2017	Frederik Funk	Added comment on power protection. Added comment on EEPROM write protection. Added chapter about testing RS485-Interface, internal checked and released by NST without changes

2. Copyright notice

3. Introduction

3.1 What is the qCAPE-COM1

The qCAPE-COM1 is an extension module specifically designed for the Beagle Bone Black (BBB). It provides different Interfaces and digital I/O for direct usage in industrial environment.

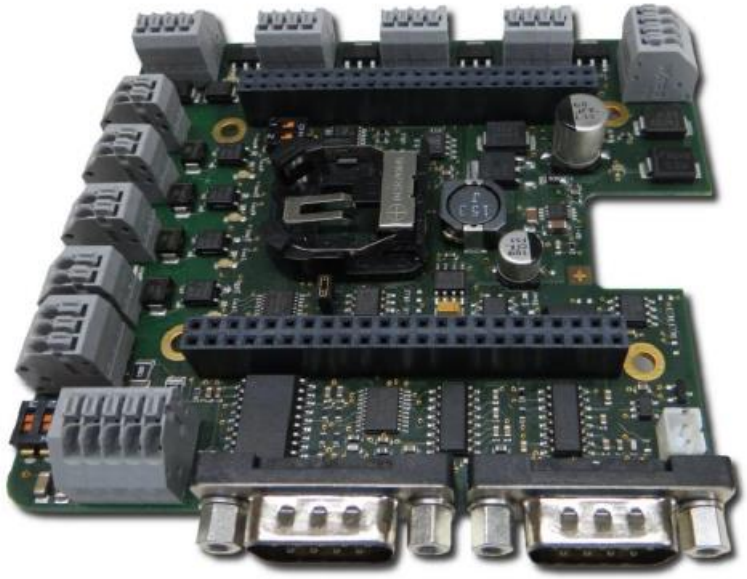
3.2 Scope of this document

This document describes how to use the various onboard devices of the Cape in Linux applications.

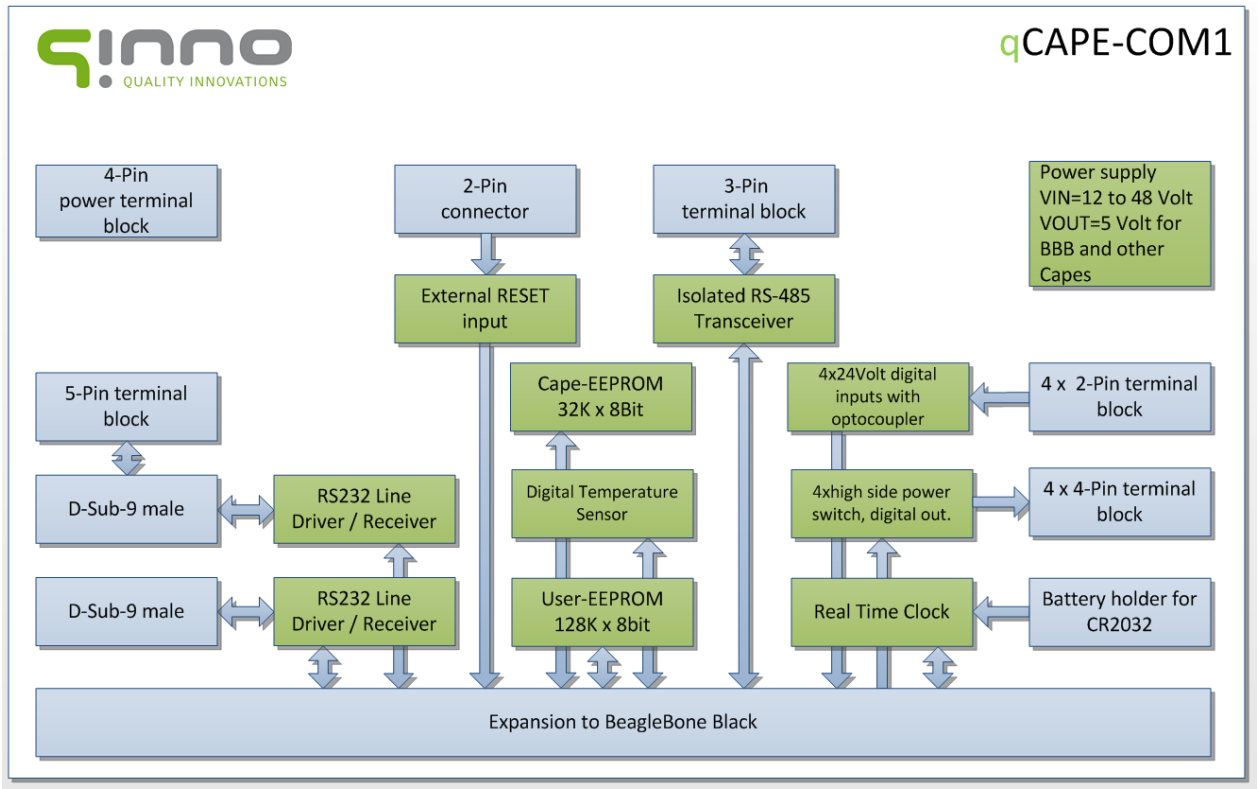
3.3 Contents

4. HW description

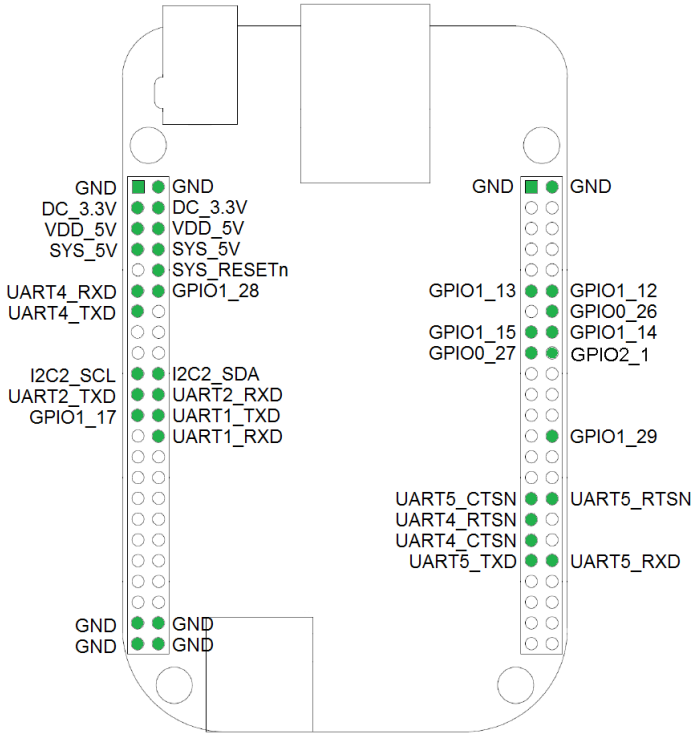
4.1 Product picture (component side)



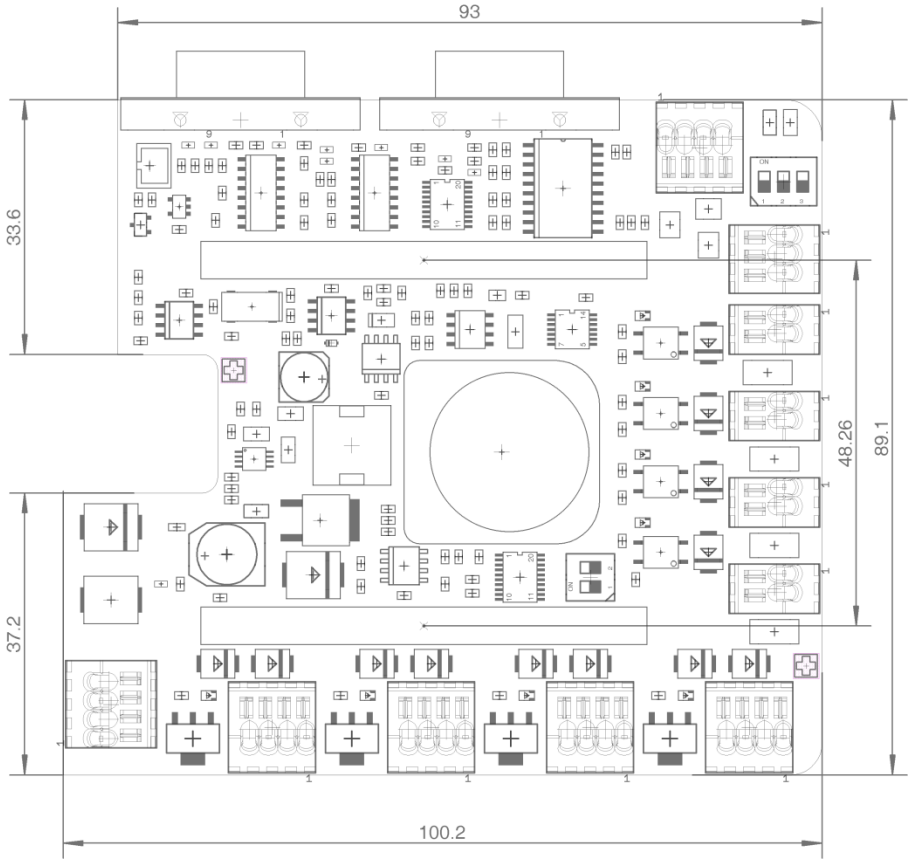
4.2 Block diagram



4.3 BBB's pins occupied by qCAPE-COM1



4.4 Mechanical dimensions of qCAPE-COM1

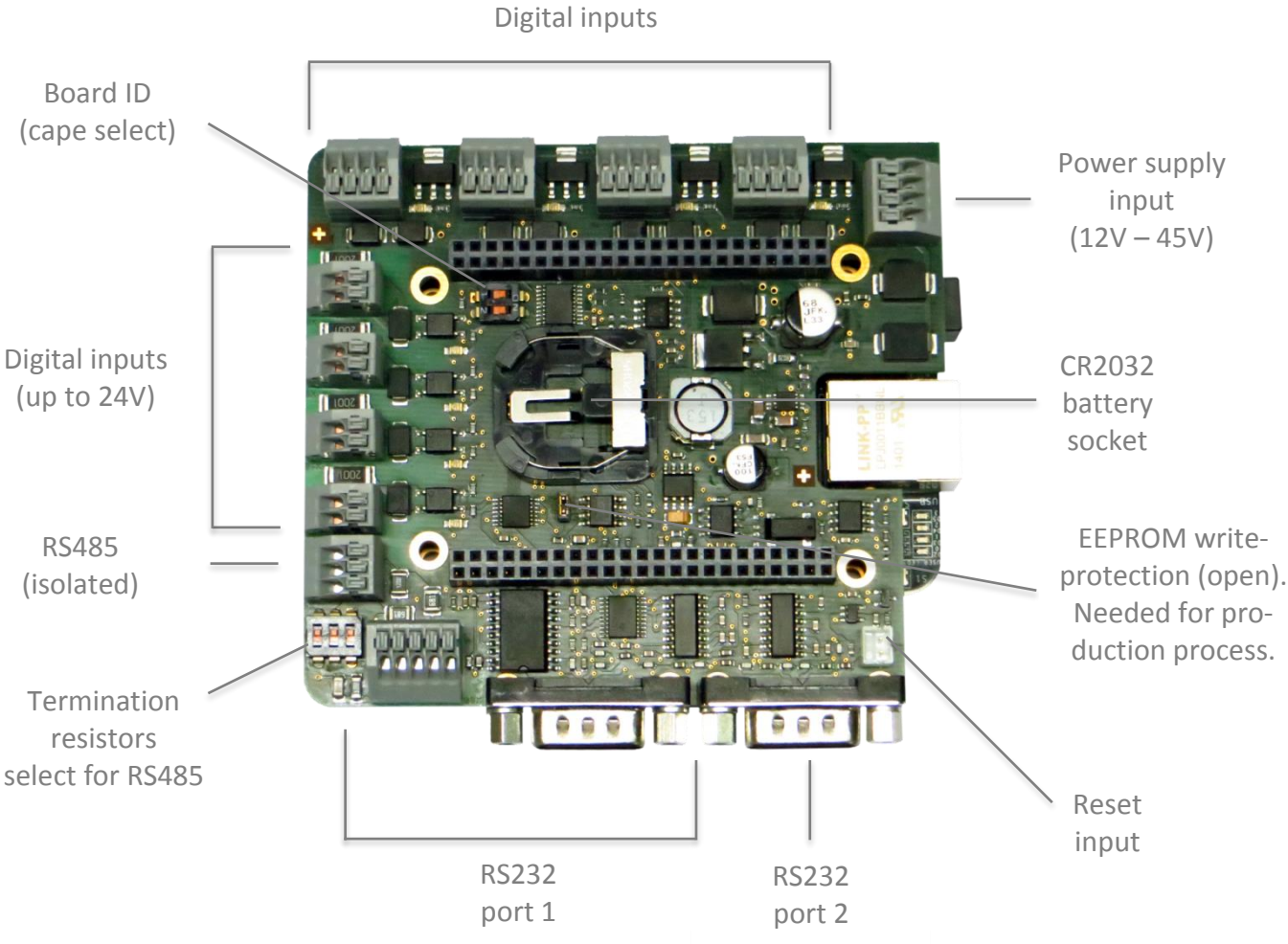


All dimensions in mm

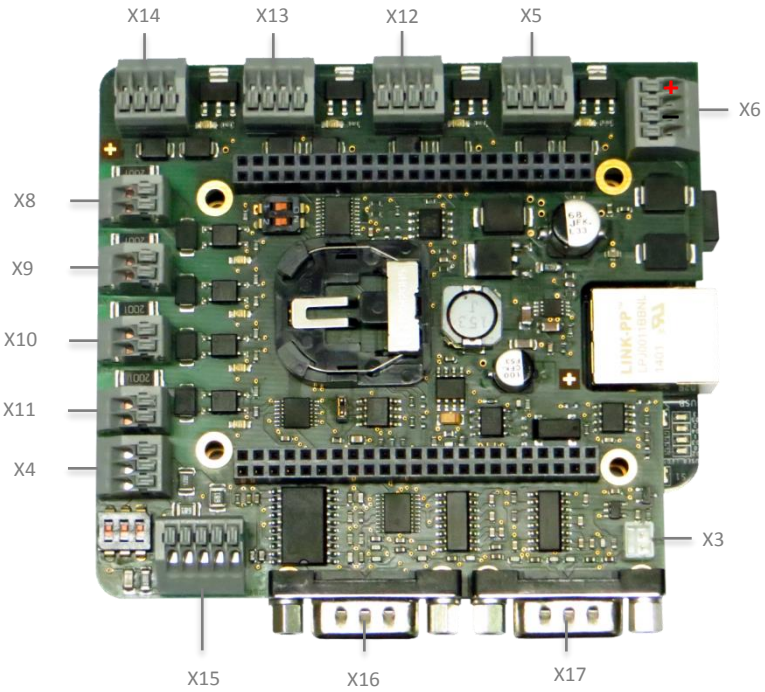
4.5 Features

- Power supply input (12V – 45V min. 0.5A. Required to work with cape). Power protection is implemented using a transient-voltage-suppression (TVS) diode and a reverse power protection diode.
- 4 digital outputs (up to 1.4A @ 24V @ 25 °C), operating Voltage 12..45V
- 4 digital inputs (up to 24V, threshold @ 6.5V)
- 2 RS232 interfaces (TXD, RXD, CTS, RTS)
- RS485 interface with 15kV ESD protection and switchable termination resistors
- Real Time Clock with leap-year compensation and CR2032 battery socket
- 1Mbit EEPROM memory with I2C interface
- On-board temperature sensor with I2C interface (-55°C to +125°C)
- Reset input connector for external switch (JST PH type)

4.6 Connector locations



4.7 Connector table and pinout



Name	Description (Board V1.1)
Power Input	
X6-1, 2	Power Input VCC (+12..+45V)
X6-3, 4	Power Input GND
Reset Input	
X3-1	GND
X3-2	Reset Input
Digital Output (max 45V)	
X5-1, 2	Positive power supply voltage
X5-3	Output to the load (GPIO0_27)
X5-4	Load GND
X12-1, 2	Positive power supply voltage
X12-3	Output to the load (GPIO0_26)
X12-4	Load GND
X13-1, 2	Positive power supply voltage
X13-3	Output to the load (GPIO1_17)
X13-4	Load GND
X14-1, 2	Positive power supply voltage
X14-3	Output to the load (GPIO2_1)
X14-4	Load GND
Digital Input (max 24V)	
X8-1	Input VCC (GPIO1_12)
X8-2	Input GND
X9-1	Input VCC (GPIO1_13)
X9-2	Input GND
X10-1	Input VCC (GPIO1_14)
X10-2	Input GND
X11-1	Input VCC (GPIO1_15)
X11-2	Input GND
RS485	
X4-1	GND
X4-2	485TX/RX-
X4-3	485TX/RX+
	GPIO1_28 = high enable the 485 transmitter
RS232	
X15	RS232 Port 1 (Clamp)
X15-1	TXD
X15-2	RXD
X15-3, 4	GND
X16	RS232 Port 1 (SubD9)
X17	RS232 Port 2

qCAPE-COM1 is compatible only with Beaglebone Black. It is not recommended to use it with any other version of Beaglebone.

qCAPE-COM1 must be powered from its input for correct work.

RoHS Compliant

qCAPE-COM1 fulfills the *RoHS* Directive requirements.

For support please contact: support@qcape.de

5. Definitions / Prerequisites

5.1 Supported Beaglebone revisions

All functions described below were executed with Beaglebone Black in Rev C.
Functionality cannot be guaranteed for other versions of the Beaglebone

5.2 Supported Linux distribution versions

All functions described below were executed with the following Linux distribution version:

- bone-debian-8.7-lxqt-4gb-armhf-2017-03-19-4gb.img
  Available under http://beagleboard.org/latest-images
  abbreviated as Debian-Image in the following text

Functionality cannot be guaranteed for other Linux distribution versions.

Please follow the instructions in the internet for flashing the Linux versions onto the Beaglebone.
http://beagleboard.org/getting-started#update

This is not described within this user manual.

5.3 Development host requirements

Standard PC or Notebook with a recent Debian or Ubuntu version.
Please follow the recommendations of the various Beaglebone information pages in the Web.

6. First steps with the qCAPE-COM1

6.1 Typographic conventions

The following typographic conventions are used in the following text:
Commands with a blue background, as shown below, are executed on a Linux development host:

```
qinno@ubu120404vbox:~$ nmap -p 22 192.168.2.140-190
```

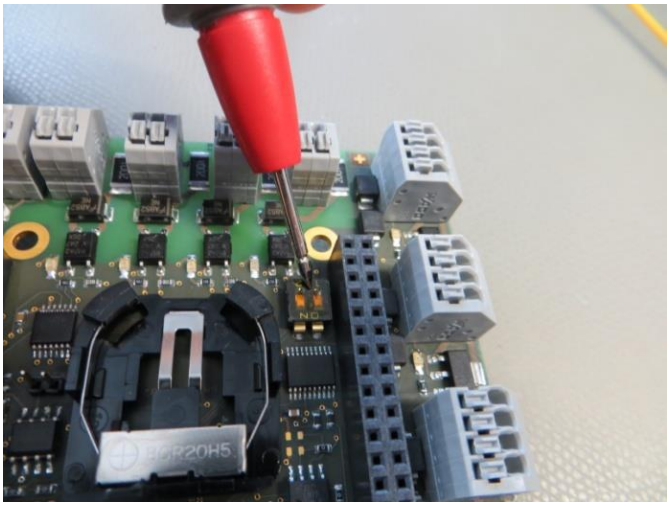
Commands with a green background have to be executed on the Beaglebone command line. Log into the Beaglebone via ssh (described later in this text), in order to execute these commands:

```
root@beaglebone:~/13_dg3_dev_si/trunk/qinno_cape1/dtb# cat /sys/devices/bone_cape-mgr.8/slots
```

Note: For initial tests, Windows users might use the Windows tool ***putty*** as a replacement for ***ssh***, and ***WinSCP*** as a replacement for the Linux tool ***scp***.
For development work it is recommended to have a Linux installation on the development host, probably in a virtual machine.

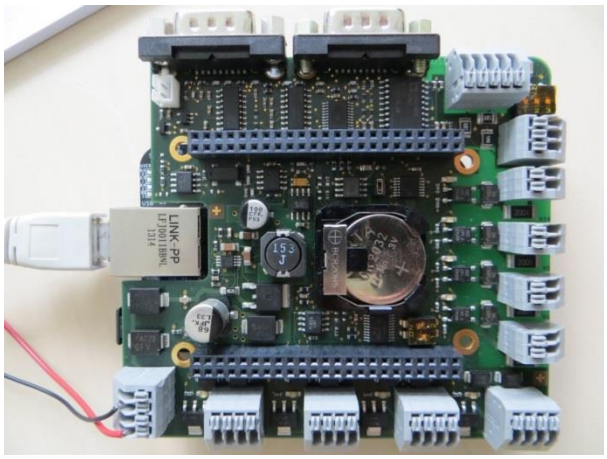
6.2 Board ID configuration DIP-Switch

2 DIP-Switches configure the Capes board ID.
Recommended setting: Switch both switches to the ON position.
Different settings might be necessary, if multiple Capes are stacked on top of each other. In this case, all modules have to be configured to different board IDs.



6.3 Apply Power

Apply power (+12V to +48V) to the Power input terminal of the Cape. The Cape generates the regulated 5V power to supply the Beaglebone and potentially other Capes. Pin1/2 = (+12V to +48V), Pin3/4=GND



As soon, as power is applied, You will recognize by the blue LEDs, that the Beaglebone starts operating.

6.4 Determine IP-Address of Beaglebone

If the Beaglebone is attached to an Ethernet network with a DHCP-Server (this is the case in most company and household networks), the server will automatically assign an IP-address to the Beaglebone.
In order to find out, which IP-address the DHCP-Server has assigned, You can either ask Your system administrator to look into the logfile of the DHCP-Server, or You can simply search for it, using the Linux tool nmap. Beaglebone has a running ssh server, which is running on Port 22. This responsiveness can be detected with nmap:

```
nmap –p 22 [ip address range]
```

For example:

```
nmap –p 22 192.168.2.1-255
```

It will respond with all IP addresses found in the specified subnet, for example:

```
qinno@ubu120404vbox:~$ nmap -p 22 192.168.2.140-190
Starting Nmap 5.21 ( http://nmap.org ) at 2014-04-30 08:17 CEST
Nmap scan report for 192.168.2.153
Host is up (0.0012s latency).
PORT STATE SERVICE
22/tcp open ssh
Nmap scan report for 192.168.2.154
Host is up (0.00079s latency).
PORT STATE SERVICE
22/tcp open ssh
Nmap scan report for 192.168.2.165
Host is up (0.00084s latency).
PORT STATE SERVICE
22/tcp open ssh
```

One of the IP-Addresses shown above, will be the IP-address of the Beaglebone.
Which one, can only be determined by try and error. Run nmap with the Beaglebone connected to the network. Then run nmap again without connected Beaglebone. Compare the two outputs, the missing IP is most likely the IP of the Beaglebone.
Execute the following command for the IP address.

```
ssh root@[ip address]
```

If You find the IP address of the Beaglebone, it will allow You to login.
Username will be ***root***, and no password:

```
Debian GNU/Linux 7
BeagleBoard.org Debian Image 2015-11-12
Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
Default username:password is [debian:temppwd]
Last login: Thu Nov 12 19:01:02 2015 from 192.168.7.1


root@beaglebone:~$
```

Please remember the IP-Address for all described commands below, and replace the IP-Address used in the examples by the IP-address of Your target.
It is also possible to connect the Beaglebone to the pc over Ethernet-over-USB, see https://beagleboard.org/getting-started#step2 for the drivers to install. Your IP address will then be 192.168.7.1 and the Beaglebone's 192.168.7.2.

6.5 Configure Beaglebone for the qCAPE-COM1

Every Beaglebone Cape must be supplied with an EEPROM. The format of this EEPROM is standardized for all Beaglebone Capes. It provides information about the features and capabilities of this Cape, and some other logistical information, for example the serial number.

WARNING: The qCAPE-COM1 is delivered with a properly configured Cape EEPROM. Please do NOT override the contents of this EEPROM. This will make Your Cape unusable.

In order to make the Cape known to the Beaglebone, a so-called DTB (Device Tree Blob) has to be loaded into the Kernel. This DTB-File describes the Cape-HW, and ensures that appropriate Kernel drivers are loaded for the Cape.

This requires the following steps:

Step 1: Copy debian package to target file system

Copy the debian package qcape_COM1_[version].deb to the Beaglebone, for example via scp:

```
…/qCape-COM1$ scp qcape_COM1_2.0.0.deb
```

Install the package on the beaglebone with:

```
root@beaglebone:~# dpkg -i qcape_com1_2.0.0.deb
Selecting previously unselected package qcape-com1.
(Reading database ... 59130 files and directories currently installed.)
Unpacking qcape-com1 (from qcape_com1_2.0.0.deb) ...
Setting up qcape-com1 (2.0.0) ...
root@beaglebone:~#
```

Check if the following files exists in /lib/firmware:

```
root@beaglebone:~# ls -l /lib/firmware/ | grep BB
-rw-r--r-- 1 root root   1613 Sep  3 23:48 BB-QINNOCAPE1-00A0.dtbo
-rw-r--r-- 1 root root   1880 Sep  3 23:48 BB-QINNOCAPE1-00A0.dts
root@beaglebone:~#
```

If not, something went wrong with the installation of the debian package.

Step 2: Tell the Kernel to load the DTB file

File uEnv.txt configures the Kernel command line of the Beaglebone. Default uEnv.txt has to be modified, in order to load the DTB for qCAPE-COM1, and in order to prevent the virtual Cape for HDMI to be loaded (Pins used by the qCAPE-COM1 conflict with the HDMI pins).

- Open file "/boot/uEnv.txt" with a text editor and add the following lines to the file

  cape_disable=capemgr.disable_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN
  cape_enable=capemgr.enable_partno=BB-QINNOCAPE1

- Manually Load DTB File:

  Note: Depending on the kernel and distribution version, Cape firmware might be automatically loaded by the uEnv.txt configuration, as described above. Unfortunately, in some kernel versions for the Beaglebone this does not work, and the firmware has to be loaded manually.

  Execute the following command on the Beaglebone (with newer versions of BBB it can be bone_capemgr.9)

  ```
  # echo BB-QINNOCAPE1 > /sys/devices/bone_capemgr.8/slots
  ```

  If an error occurs restart the Beaglebone and enter the command again.
  It is recommended to add this command to the start script, which starts the user application.

Step 4: Verify correct loading of qCAPE-COM1

Cape manager shows all cape DTBs loaded, with the following command:

```
# cat /sys/devices/bone_capemgr.8/slots
1: 55:PF---
2: 56:PF---
3: 57:PF---
4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
5: ff:P-O-- Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
6: ff:P-O-- Bone-Black-HDMIN,00A0,Texas Instrument,BB-BONELT-HDMIN
7: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-QINNOCAPE1
```

If the BB-QINNOCAPE1 appears in this list, the previous steps were successfully.
If it does not appear in the list, You will not be able to use any of the features of this cape.

6.6 Serial Interfaces

qCAPE-COM1 provides 3 serial interfaces, 2 x RS232, and 1 x RS485. Together with the onboard serial debug port of the Beaglebone, there are 4 serial interfaces in total.

This can be visualized with the following command:

```
root@beaglebone:~# ls -l /dev/ttyO*
crw-rw---- 1 root tty 249, 0 Jan 1 15:48 /dev/ttyO0
crw-rw---- 1 root dialout 249, 1 Jan 1 15:49 /dev/ttyO1
crw-rw---- 1 root dialout 249, 4 Jan 1 15:49 /dev/ttyO4
crw-rw---- 1 root dialout 249, 5 Jan 1 15:49 /dev/ttyO5
```

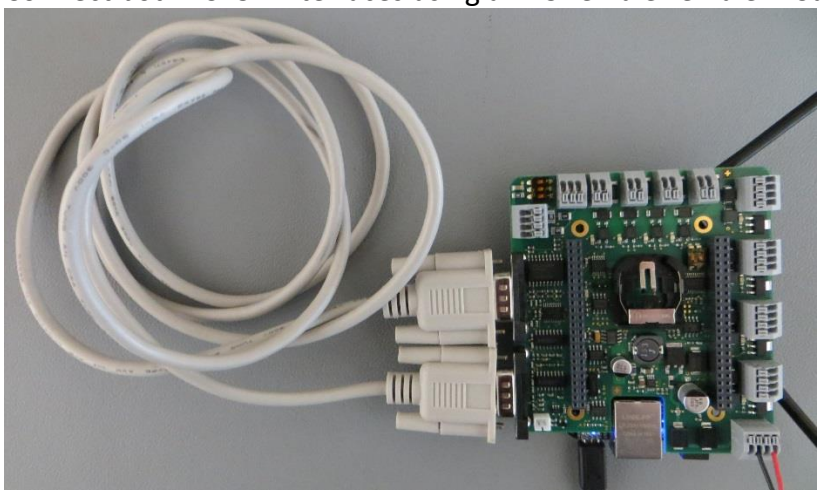
Relation between Linux device node and physical connectors:

Device	Connector	Signal	Location / Remark
/dev/ttyO0	Pinrow on Bea-glebone	Logic Level	Debug connector on Beagle-bone. Not accessible, if any Cape is attached
/dev/ttyO1	X4	RS485	qCAPE-COM1
/dev/ttyO4	X17	RS232 Port 2	qCAPE-COM1
/dev/ttyO5	X16 or X15	RS232 Port 1	qCAPE-COM1

SW usage of serial ports: These serial ports are supported by the serial driver of the Beaglebone. Standard Linux tools, like ***stty*** or ***minicom***, can be used on these ports.
For programming under C, C++, the standard serial driver functions read(), write(), and ioctl() can be used.

6.6.1 Testing the RS232 interfaces

Connect both RS232 interfaces using a DE9 female-female D-Sub cable.



Then start listening to /dev/ttyO4 running the following command:

```
# cat /dev/ttyO4
```

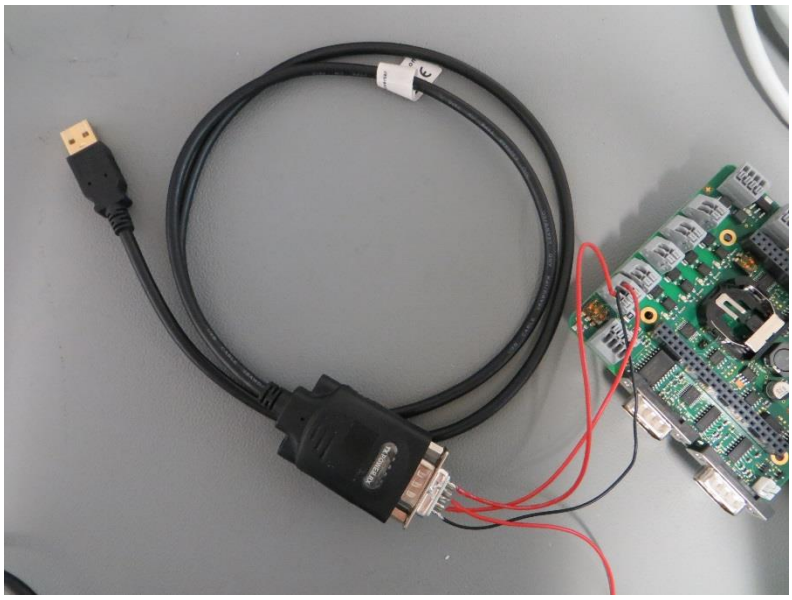
Open a second terminal, log in to the Beaglebone and start sending messages from */dev/ttyO5*.

```
# echo “test” > /dev/ttyO5
```

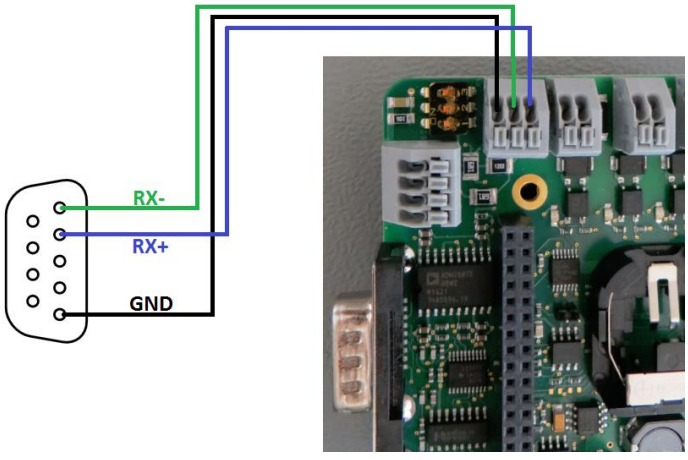
The messages should appear in the terminal listening to */dev/ttyO4*. To ensure that the serial connection is running bidirectional, you can listen to */dev/ttyO5* and send messages from */dev/ttyO4*.

6.7 Testing the RS485 interface

Connect the RS485 interface to the computer using an USB to RS485 converter (e.g. this one: http://www.lindy.de/USB-RS485-Konverter.htm?websale8=ld0101&pi=42845).



Be sure that the three pins are connected correctly to the adapter. For the adapter named above the connection is visualized in the following picture:



Turn on the three DIP-Switches in order to activate the termination resistors.

Then open a serial terminal (e.g. Tera Term) listening to the serial port. Make sure, that you set the baud rate correctly. You can find out the baud rate by running

```
# stty -F /dev/ttyO1
```

on the beaglebone.
Set GPIO_1-28 to “high” (see chapter 6.8) in order to make transmitting possible (gpionr = 60).
Then send messages from the beaglebone to the computer using the following command:

```
# echo “Test” > /dev/ttyO1
```

The messages should now appear in the serial listener.
After that, change to receiving mode by setting GPIO_1-28 to “low” (see chapter 6.8). Listen to incoming messages:

```
# cat /dev/ttyO1
```

Send messages from the computer to the beaglebone by writing into the serial console (Tera Term). They should now appear in the beaglebone’s command line.

6.8 GPIOs

qCAPE-COM1 provides 4 Digital Inputs and 4 Digital Outputs.
Status of each input and output is visualized by an accompanying LED.
All Beaglebone GPIOs can be accessed from Linux command line via the /sys filesystem. Examples of using this are located in the gpio scripts in the root directory after installation of debian package.
Controlling GP-Outputs from the Linux command line (Replace [gpionr] with the number of the table below):

Make the GPIO visible in /sys file system and configure it as an output

```
# echo [gpionr] > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio[gpionr]/direction
```

Set value to 1

```
# echo 1 > /sys/class/gpio/gpio[gpionr]/value
```

Set value to 0

```
# echo 0 > /sys/class/gpio/gpio[gpionr]/value
```

Reading GP-Input from the Linux command line (Replace [gpionr] with the number of the table below):
Make the GPIO visible in /sys file system and configure it as an input

```
# echo [gpionr] > /sys/class/gpio/export
# echo in > /sys/class/gpio/gpio[gpionr]/direction
```

Read Value (will return 0 or 1)

```
# cat /sys/class/gpio/gpio[gpionr]/value
```

Relations between Connectors and Beaglebone GPIO numbers:

Connector	Direction	Beaglebone PIN	gpionr
X5	Output	GPIO0_27	27
X12	Output	GPIO0_26	26
X13	Output	GPIO1_17	49

X14	Output	GPIO2_1	65
X8	Input	GPIO1_12	44
X9	Input	GPIO1_13	45
X10	Input	GPIO1_14	46
X11	Input	GPIO1_15	47

Accessing GPIO from application coded in C/C++:
GPIOs can be accessed from application code by reading and writing the /sys files, as described above. There are various example programs and libraries available on the Internet.

Using scripts for testing:
After installing debian package, there are three scripts located in the root directory. Use gpio_init.sh for initializing the GPIOs, gpio_out.sh for testing outputs and gpio_in.sh for testing inputs.

6.9 I2C-Devices

Beaglebone has 2 I2C-Busses, which are routed to the qCAPE-COM1.
Bus 0 is used for internal devices, as well as for the Cape EEPROM.

Warning: Never touch I2C bus 0 in your applications. This may lead to a nun-functional system.

Bus 1 contains the user devices.
Beaglebone Kernel provides the I2C-Dev driver, which provides access to the I2C-Bus from the user space. You can use the i2c-tools (i2cdetect, i2cdump, …) on these devices.

Example: Detection of all I2C-Devices on Bus 1:

```
root@beaglebone:~# i2cdetect -r -a 1
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-1 using read byte commands.
I will probe address range 0x00-0x7f.
Continue? [Y/n] y
0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: 50 51 -- -- UU UU UU UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@beaglebone:~#
```

The following table shows this information in a human readable format:

I2C-Address	Device	Remark
0x68	RealtimeClock DS1338Z	

0x50,0x51	EEPROM 24LC1026	
0x48	Temperature sensor LM77	See chapter LM77 below

6.10 LM77 Temperature sensor

BSP contains a demo program for reading temperatures from the LM77 temperature sensor. You can use the binary for a quick functionality check, and you are free to use the provided source code for integration in own applications.
At first, you have to set the binary executable. Change in the root directory and run

```
root@beaglebone:~# chmod +x lm77_test
```

Execute program on Beaglebone, it is installed in the /root directory after installing debian package.
Get Help:

```
root@beaglebone:~# ./lm77_test -?
./lm77_test: invalid option -- '?'
usage: lm77_test -i [i2c_fname] -y [temp] -l [temp] -h [temp] -c [temp]
-i i2c_dev i2c device file. Default /dev/i2c-0 or /dev/i2c/0
-y temp Write hysteresis register
-c temp Write critical temp register
-l temp Write low register
-h temp Write high register
```

Read temperatures:

```
root@beaglebone:~# ./lm77_test -i /dev/i2c-1
TEMP: 36.5C T_HYST_REG: 2.0C T_CRIT_REG: 2.0C T_HIGH_REG: 2.0C T_LOW_REG: 2.0
```