BeagleBone Blue

BeagleBoard.org Foundation
Jun 29, 2025

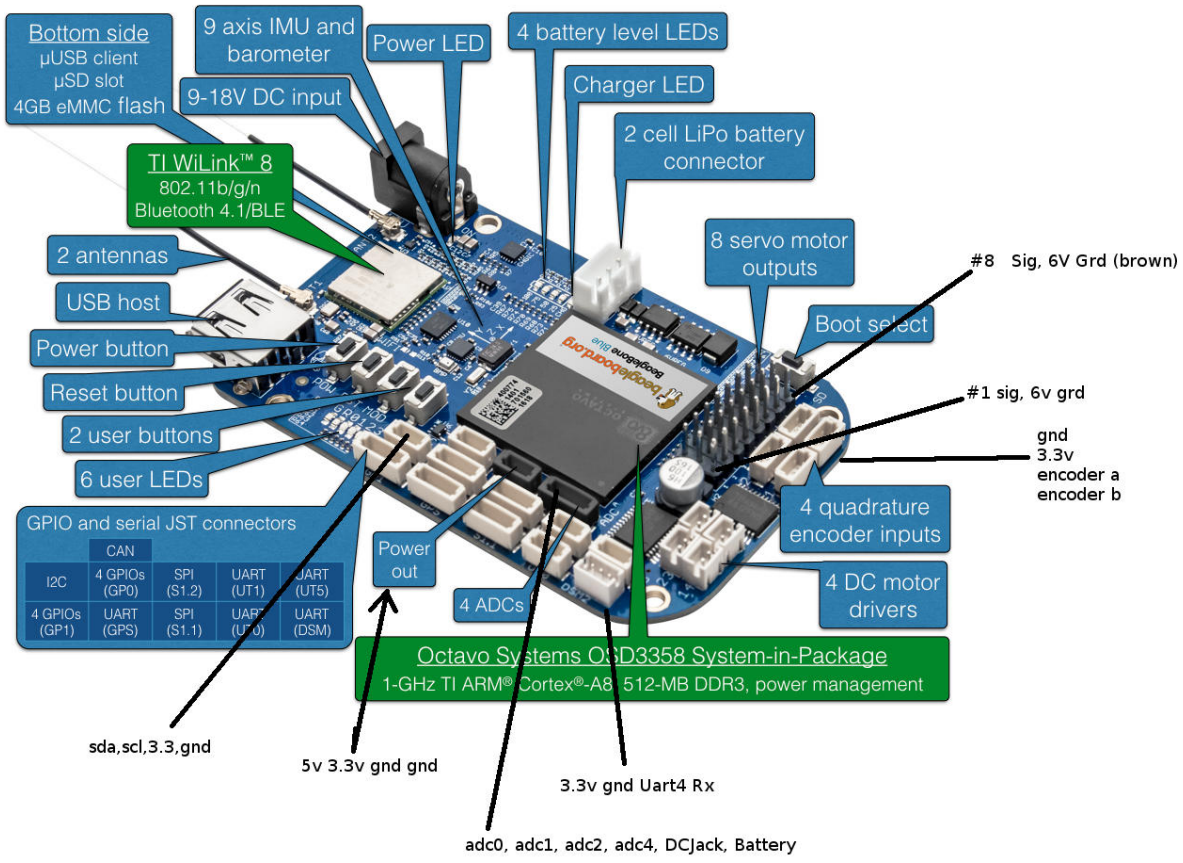
Table of contents

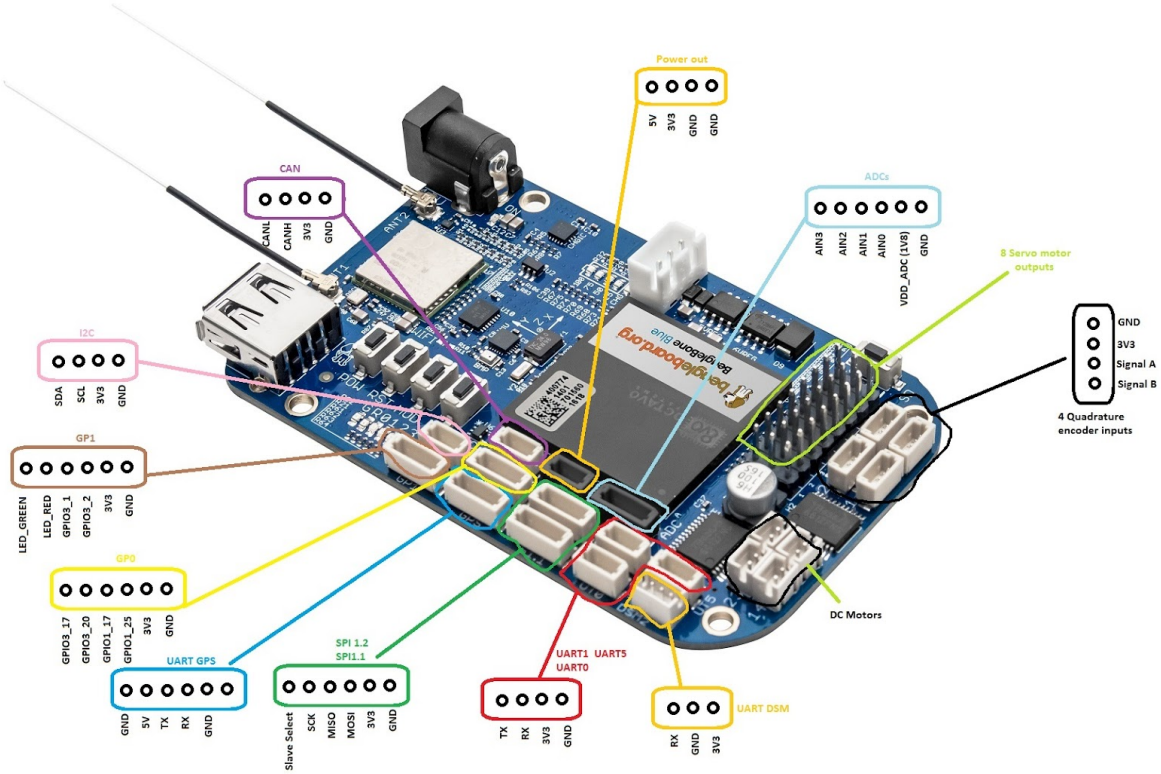
To optimize BeagleBone for education, BeagleBone Blue was created that integrates many components for robotics and machine control, including connectors for off-the-shelf robotic components. For education, this means you can quickly start talking about topics such as programming and control theory, without needing to spend so much time on electronics. The goal is to still be very hackable for learning electronics as well, including being fully open hardware.

BeagleBone Blue's legacy is primarily from contributions to BeagleBone Black robotics by UCSD Flow Control and Coordinated Robotics Lab, Strawson Design, Octavo Systems, WowWee, National Instruments LabVIEW and of course the BeagleBoard.org Foundation.

Chapter 1

BeagleBone Blue Pinouts





- Connector pinout details from schematic(s)
- Pin Table with some Blue : Black corelation.

1.1 UT1

1.1.1 UART (/dev/ttyS1)

```
config-pin P9.24 uart
config-pin P9.26 uart
```

1.2 GPS

1.2.1 UART (/dev/ttyS2)

```
config-pin P9.21 uart
config-pin P9.22 uart
```

Chapter 2

SSH

If you don't have ssh installed, install it. (google is your friend) Then *ssh debian@192.168.7.2* The board will tell you what the password is, on my it was *temppwd*.

To change your password use the command password it will ask you what your current password is, then ask for the replacement. Then it will say it was too simple and you have to do it again. Normal stuff.

If you want to insist on using your simple password, try this.

```
sudo -s
(become superuser/root)
enter your password
password debian
 (put your simple password in)
exit
(exit from superuser/root)
```

When you are running as root, password is more compliant and will accept simple password

Chapter 3

WiFi Setup

On my network, I'm set up as ip 192.168.1.*. To turn your wifi on, do the following.

```
sudo -s
(become superuser/root)
cd /etc/network/
ifconfig
(Note the wifi inet address, if it is already set, you are done!)
connmanctl
tether wifi off
enable wifi
scan wifi
services
(at this point you should see your network appear along with other stuff, in
↪my case it was "AR Crystal wifi_f45eab2f1ee1_6372797774616c_managed_psk")
nano interfaces
(or whatever editor you like)
remove the comment # from the wifi lines so it now appears like
##connman: WiFi
#
connmanctl
connmanctl> tether wifi off
connmanctl> enable wifi
connmanctl> scan wifi
connmanctl> services
connmanctl> agent on
connmanctl> connect wifi_f45eab2f1ee1_6372797774616c_managed_psk
connmanctl> quit
exit
note that you will need to fill in your own network data
```

Chapter 4

IP settings

You will usually want to have a fixed ip if you are doing robotics, so you have a standard ip to connect to. If you are already connected in dhcp you can borrow some of the settings from that to use in your new configurations.

```
route
```

make a note of the default one, (in the example below 192.168.1.1)

```
cat /etc/resolv.conf
```

make a note of the nameserver, (in the example below 8.8.8.8)

In my case I wanted 192.168.1.7 to do this,

```
sudo -s
connmanctl config wifi_f45eab2f1ee1_6372797774616c_managed_psk --ipv4 manual␣
↪192.168.1.7 255.255.255.0 192.168.1.1 --nameservers 8.8.8.8
exit
```

the –ipv4 says to use ipv4 settings (as opposed to ipv6), the manual means we are setting the values. 192.168.1.7 is the ip address we want. (use your own of course). 255.255.255.0 is the network mask 192.168.1.1 is the route to the internet. (You're might be different, but this is common). –nameservers 8.8.8.8 says where to find the ip address for a given domain name. the 8.8.8.8 says use's googles

Chapter 5

Flashing Firmware

5.1 Overview

Most Beaglebones have a built in 4 GB SD card known as a eMMC (embedded MMC). When the boards are made the eMMC is “flashed” with some version of the BeagleBone OS that is usually outdated. Therefore, whenever receiving the BeagleBone it is recommend that you update the eMMC with the last version of the BeagleBone OS or a specific version of it if someone tells you otherwise.

5.2 Required Items

1. Micro sd card. 4 GB minimum
2. Micro sd card reader or a built in sd card reader for your PC
3. BeagleBone image you want to flash.
4. Etcher utility for your PC’s OS.

5.3 Steps Overview

1. Burn the image you want to flash onto a micro sd card using the Etcher utility.
2. Boot the BeagleBone like normal and place the micro sd card into the board once booted.
3. Update the micro sd card image so its in “flashing” mode.
4. Insert micro sd card, remove power from the BeagleBone, hold sd card select button, power up board
5. Let the board flash

5.3.1 Windows PCs

1. Download the BeagleBone OS image you want to use.
2. Use the Etcher utility to burn the BeagleBone image you want to use on the micro sd card you plan on using.
3. Make sure you don’t have the micro sd card plugged into your board.
4. Boot the board
5. Connect to the board via serial or ssh so that your on the command prompt.
6. Plug the micro sd card into the board.

7. Type dmesg in the terminal window
8. The last line from the output should say something like (the numbering may differ slightly):
    - "[2805.442940] mmcblk0: p1"
9. You want to take the above and combine it together by removing the : and space. For the above example it will change to "mmcblk0p1"
10. In the terminal window enter the following commands:

```
mkdir sd_tmp
sudo mount /dev/mmcblk0p1 sd_tmp
sudo su
echo "cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh" >> sd_
↪tmp/boot/uEnv.txt
exit
sudo umount sd_tmp
```

11. Now power off your board
12. Hold the update button labeled SD (the one by itself) to boot off the sdcard.
13. Restart (RST button) or power up (while still pushing SD button).

Flashing can take some minutes.

5.3.2 Linux/Mac PCs

1. Download the BeagleBone OS image you want to use.
2. Use the Etcher utility to burn the BeagleBone image you want to use on the micro sd card you plan on using.
3. On the SD card edit the file /boot/uEnv.txt in order for the SD card contents to be flashed onto the firmware eMMC. (Otherwise the BBBL will do no more than boot the SD image.) Uncomment the line containing init-eMMC-flasher-v<number>.sh either manually or using these commands substituting X with what your SD card shows in /dev/:

```
sudo mount /dev/emmcblkXp1 /mnt
cd /mnt
sed -i 's_#[ ]*\(cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v[0-
↪9]\+.*\.sh\)_\1_' boot/uEnv.txt
```

4. Eject the sdcard from your computer.
5. Put it into your BeagleBoneBlue.
6. If your board was already powered on then power it off
7. Hold the update button labeled SD (the one by itself) to boot off the sdcard.
8. Restart (RST button) or power up (while still pushing SD button).

Flashing can take some minutes.

How to tell if it is flashing?

At first a blue heartbeat is shown indicating the image is booted. On flash procedure start, the blue user LEDs light up in a "larson scanner" or "cylon" pattern (back and forth).

When finished, either all blue LEDs are on or the board is already switched off.

If the LEDs are on for a long time then it may indicate failure e.g. wrong image. Can be verified if boot fails, i.e. board turns off again shortly after power up.

Chapter 6

Play with the code

The board has some code built in to the system that can allow you to try out the various options. They all start with rc

```
rc_balance              rc_dsm_passthrough    rc_test_encoders
rc_battery_monitor      rc_kill               rc_test_filters
rc_benchmark_algebra    rc_spi_loopback       rc_test_imu
rc_bind_dsm             rc_startup_routine    rc_test_motors
rc_blink                rc_test_adc           rc_test_polynomial
rc_calibrate_dsm        rc_test_algebra       rc_test_servos
rc_calibrate_escs       rc_test_barometer     rc_test_time
rc_calibrate_gyro       rc_test_buttons       rc_test_vector
rc_calibrate_mag        rc_test_cape          rc_uart_loopback
rc_check_battery        rc_test_dmp           rc_version
rc_check_model          rc_test_drivers
rc_cpu_freq             rc_test_dsm
```

Try them out to try out the various functions of the board. The source code for these tests and demos is at Robotics cape installer at github

Chapter 7

BeagleBone Blue tests

7.1 ADC

- Grove Rotary Angle Sensor See output on adc_1 source

```
rc_test_adc
```

7.2 GP0

- Grove single GPIO output modules like LED Socket Kit

```
cd /sys/class/gpio;echo 49 >export;cd gpio49;echo out >direction;while sleep␣
↪1;do echo 0 >value;sleep 1;echo 1 >value;done
```

- Grove single GPIO input modules like IR Distance Interrupter or Touch Sensor

```
cd /sys/class/gpio;echo 49 >export;cd gpio49;echo in >direction;watch -n0␣
↪cat value
```

7.3 GP1

- Grove single GPIO output modules like LED Socket Kit

```
cd /sys/class/gpio;echo 97 >export;cd gpio97;echo out >direction;while sleep␣
↪1;do echo 0 >value;sleep 1;echo 1 >value;done
```

- Grove single GPIO input modules like IR Distance Interrupter or Touch Sensor

```
cd /sys/class/gpio;echo 97 >export;cd gpio97;echo in >direction;watch -n0␣
↪cat value
```

7.4 UT1

- Grove GPS

```
tio /dev/ttyO1 -b 9600
```

7.5 GPS

- GPS Receiver - EM-506

```
tio /dev/ttyO2 -b 4800
```

7.6 I2C

7.6.1 Grove I2C modules

The Linux kernel source has some basic IIO SYSFS interface documentation which might provide a little help for understanding reading these entries. The ELC2017 conference also had an IIO presentation.

- Digital Light Sensor

```
cd /sys/bus/i2c/devices/i2c-1;echo tsl2561 0x29 >new_device;watch -n0 cat 1-
↪0029/iio\:device0/in_illuminance0_input
```

- Temperature & Humidity Sensor

```
cd /sys/bus/i2c/devices/i2c-1;echo th02 0x40 >new_device;watch -n0 cat 1-
↪0040/iio\:device0/in_temp_raw
```

7.7 Motors

```
rc_test_motors
```

Chapter 8

Accessories

Todo: We are going to work on a unified accessories page for all the boards and it should replace this.

8.1 Chassis and kits

- EduMIP
- Pololu Romi Chassis with geared motors
  - Wheel encoders
  - Chassis - Black
- Sprout Runt Rover

8.2 Cases

8.3 Cable assemblies and sub-assemblies

Beware; purchased pre-made connector assembly wire colors may not reflect true pin designations. These assemblies are readily available from Digi-Key, SparkFun, Hobby King, Pololu and Cables and Connectors.

8.3.1 JST Connector Bundle

Renaissance Robotics JST Jumper Bundle

Four of the 2-pin JST ZH (1.5mm pitch) connectors, with 150mm 28AWG wires, for motors,
Eight of the 4-pin JST SH (1mm pitch) connectors, with 150mm 28AWG wires, for encoders, UART, I2C, CAN, PWR, and
Four of the 6-pin JST SH (1mm pitch) connectors, with 150mm 28AWG wires, for SPI, GPS, GPIO, ADC.
Renaissance Robotics JST Jumper Bundle

Conrad BeagleBoard Kabel BB-Blue-Kabelset

10x 4-Pin JST-SH
6x 6-Pin JST-SH

4x 2-Pin JST-ZH
1x 3-Pin JST-ZH
BeagleBoard Kabel BB-Blue-Kabelset (Conrad.de)

8.3.2 UART, I2C, CAN, Quadrature encoders, PWR

4-wire JST-SH (1mm pitch)

- 4-wire Grove cable (Digi-Key)
- Hobby King SKU 258000190-0
- SparkFun PN 10359
- Cables and Connectors 4” ribbon PN #4904
- Digi-Key wires
- Digi-Key housings

8.3.3 SPI, GPIO, ADC

6-wire JST-SH (1mm pitch)

- Hobby King SKU 258000192-0
- SparkFun PN 10361
- Cables and Connectors 50cm length PN #49406
- Digi-Key wires
- Digi-Key housings
- 6-wire Grove cable (4 populated) (Digi-Key)

8.4 Motors

2-wire JST-ZH (1.5mm pitch)

- Digi-Key wires
- Digi-Key receptacle

8.5 DSM

3-wire JST-ZH (1.5mm pitch)

- Pololu PN# 2411

8.6 microUSB

standard

8.7 Batteries

2S1P LiPo with 3-wire JST-XH (2.5mm pitch) charge connection

- Hobby King 1000mAh 2S 20C LiPo
- Hobby King 1600mAh 2S 20C LiPo

8.8 Power supplies

12V with 5.5mm/2.1mm center positive

- Jameco: supply and power cord
- Hobby King 12V 3A supply

8.9 Motors

8.10 Servo motors

6V DC

- Parallax Inc. 900-00005 Standard Servo
- Hobby King SKU HD-1900A
- TowerPro SG92R-7

8.11 DC motors

6V, typically geared

- SparkFun Hobby Gearmotor - 200 RPM (Pair)
- SparkFun Hobby Motor - Gear

8.12 Radio remotes

- Hobby King OrangeRX satellite receiver
- Spektrum DSM2 Remote Receiver

8.13 GPS

- Sparkfun GPS Receiver - EM-506 (48 Channel)
- Adafruit Ultimate GPS breakout
- Ublox Neo-M8N GPS with Compass
- SeeedStudio Grove - GPS

8.14 Replacement antennas

- LSR PIFA
- LSR Dipole: antenna and cable
- Anaren U.FL 2.4GHz 6MM Antenna
- TI approved antennas

8.15 USB devices

8.15.1 USB cameras

- Logitech C270
- Logitech C920

8.16 SPI devices

8.16.1 SPI TFT displays

- Adafruit 2.4” LCD breakout

8.17 I2C devices

- See *One Liner Module Tests*
- See beagle101_i2c

8.18 UART devices

8.18.1 Computer serial adapters

- Sparkfun FTDI Cable 5V VCC-3.3V I/O
- Adafruit FTDI Serial TTL-232 USB Cable

8.19 Bluetooth devices

- WowWee Groove Cube Speaker

Chapter 9

Frequently Asked Questions (FAQs)

Important: All support for BeagleBone Blue design is through BeagleBoard.org community at BeagleBoard.org forum.

9.1 Production board boot media

Todo: Add production boot media link in `_static/epilog/production.image` and reference it here.

9.2 Are there any books to help me get started?

The book BeagleBone Robotic Projects, Second Edition specifically covers how to get started building robots with BeagleBone Blue.

For more general books on BeagleBone, Linux and other related topics, see https://beagleboard.org/books.

9.3 What system firmware should I use for starting to explore my BeagleBone Blue?

Download the latest 'IoT' image from https://www.beagleboard.org/distros. As of this writing, that image is https://debian.beagleboard.org/images/bone-debian-9.5-iot-armhf-2018-10-07-4gb.img.xz.

Use http://etcher.io for writing that image to a 4GB or larger microSD card.

Power-up your BeagleBone Blue with the newly created microSD card to run this firmware image.

9.4 What is the name of the access point SSID and password default on BeagleBone Blue?

SSID: BeagleBone-XXXX where XXXX is based upon the board's assigned unique hardware address
Password: BeagleBone

9.5 I've connected to BeagleBone Blue's access point. How do I get logged into the board?

Browse to http://192.168.8.1:3000 to open the Visual Studio Code IDE and get access to the Linux command prompt.

If you've connected via USB instead, the address will be either http://192.168.6.2:3000 or http://192.168.7.2:3000, depending on the USB networking drivers provided by your operating system.

9.6 How do I connect BeagleBone Blue to my own WiFi network?

From the bash command prompt in Linux:

```
sudo -s (become superuser/root)

connmanctl
   connmanctl> tether wifi off (not really necessary on latest images)
   connmanctl> enable wifi (not really necessary)
   connmanctl> scan wifi
   connmanctl> services (at this point you should see your network
   appear along with other stuff, in my case it was "AR Crystal wifi_
↪f45eab2f1ee1_6372797774616c_managed_psk")
   connmanctl> agent on
   connmanctl> connect wifi_f45eab2f1ee1_6372797774616c_managed_psk
   connmanctl> quit
```

9.7 Where can I find examples and APIs for programming BeagleBone Blue?

Programming in C: http://www.strawsondesign.com/#!manual-install

Programming in Python: https://github.com/mcdeoliveira/rcpy

Programming in Simulink: https://www.mathworks.com/hardware-support/beaglebone-blue.html

9.8 My BeagleBone Blue fails to run successful tests

You've tried to run rc_test_drivers to ensure your board is working for DOA warranty tests, but it errors. You should first look to fixing your bootloader as described http://strawsondesign.com/docs/librobotcontrol/installation.html#installation_s5

9.9 I'm running an image off of a microSD card. How do I write it to the on-board eMMC flash?

Refer to the "Flashing Firmware" page: https://git.beagleboard.org/beagleboard/beaglebone-blue/-/wikis/Flashing-firmware

Meanwhile, as root, run the /opt/scripts/tools/eMMC/bbb-eMMC-flasher-eewiki-ext4.sh script which will create a copy of the system in your microSD to a new single ext4 partition on the on-board eMMC.

9.10 I've written the latest image to a uSD card, but some features aren't working. How do I make it run properly?

It is possible you are running an old bootloader off of the eMMC. While power is completely off, hold the SD button (near the servo headers) while applying power. You can release the button as soon the power LED comes on. This will make sure the bootloader is loaded from microSD and not eMMC.

Verify the running image using version.sh via:

```
sudo /opt/scripts/tools/version.sh
```

The version.sh output will tell you which version of bootloader is on the eMMC or microSD. Future versions of version.sh might further inform you if the SD button was properly asserted on power-up.

One you've booted the latest image, you can update the bootloader on the eMMC using /opt/scripts/tools/developers/update_bootloader.sh. Better yet, read the above FAQ on flashing firmware.

9.11 I've got my on-board eMMC flash configured in a nice way. How do I copy that to other BeagleBone Blue boards?

As root, run the /opt/scripts/tools/eMMC/beaglebone-black-make-microSD-flasher-from-eMMC.sh script with a blank 4GB or larger microSD card installed and wait for the script to complete execution.

Remove the microSD card.

Boot your other BeagleBone Blue boards off of this newly updated microSD card and wait for the flashing process to complete. You'll know it successfully started when you see the "larson scanner" running on the LEDs. You'll know it successfully completed when it shuts off the board.

Remove the microSD card.

Reboot your newly flashed board.

9.12 I have some low-latency I/O tasks. How do I get started programming the BeagleBone PRUs?

There is a "Hello, World" app at https://gist.github.com/jadonk/2ecf864e1b3f250bad82c0eae12b7b64 that will get you blinking the USRx LEDS.

The libroboticscape software provides examples that are pre-built and included in the BeagleBone Blue software images for running the servo/ESC outputs and fourth quadrature encoder input. You can use those firmware images as a basis for building your own: https://github.com/StrawsonDesign/Robotics_Cape_Installer/tree/master/pru_firmware

You can find some more at https://beagleboard.org/pru

9.13 Are there available mechanical models?

A community contributed model is available at https://grabcad.com/library/beaglebone-blue-1

9.14 What is the operating temperature range?

'0..70' due to processor, else '-20..70'

9.15 What is the DC motor drive strength?

This is dictated by the 2 cell LiPo battery input, the TB6612FNG motor drivers and the JST-ZH connectors

- Voltage: 6V-8.4V (typical)
- Current: 1A (maximum for connectors) / 1.2A (maximum average from drivers) / 3.2A (peak from drivers) per channel