



# **Avionics System I User Manual**

**EI00001**

**Rev: A01**  
Cem Eden  
2019-07-18

# Table of Contents

<b>1</b>	<b>Beaglebone Black</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Connecting to the Beaglebone . . . . .	2
1.2.1	Using a Mac . . . . .	2
1.2.2	Using Windows . . . . .	4
1.2.2.1	Using Powershell . . . . .	4
1.2.2.2	Using PuTTY . . . . .	4
1.3	Using Debian . . . . .	6
1.3.1	Basic commands . . . . .	6
1.3.1.1	ls . . . . .	6
1.3.1.2	cd . . . . .	6
1.3.1.3	cp . . . . .	6
1.3.1.4	mv . . . . .	7
1.3.1.5	rm . . . . .	7
1.3.1.6	pwd . . . . .	7
1.3.2	Command Line Text Editors . . . . .	8
1.3.2.1	nano . . . . .	8
1.3.2.2	vim . . . . .	8
1.3.3	Bash Scripting . . . . .	9
1.3.4	Further Reading . . . . .	10
<b>2</b>	<b>Telemetry Board</b>	<b>11</b>
<b>3</b>	<b>Actuator Board</b>	<b>11</b>
<b>4</b>	<b>Scate Board</b>	<b>11</b>
<b>5</b>	<b>Floor Board</b>	<b>11</b>
<b>6</b>	<b>Water Board</b>	<b>11</b>
<b>7</b>	<b>Advisory Board</b>	<b>11</b>
<b>8</b>	<b>Mid-Sized Staffed with Volunteering Board</b>	<b>11</b>
<b>9</b>	<b>3 ply construction Board</b>	<b>11</b>

# 1 Beaglebone Black

## 1.1 Introduction

The Beaglebone Black is essentially a full mini-computer that can run Unix systems. In our case, we run a command-line only Debian distribution. This means that we can easily use almost any programming or scripting language that suits our needs.

Most of this guide will be focusing on the programs that are being or have been written for the Beaglebone and how to use them. In addition, you will find a short guide on how to use Unix systems, with emphasis on Debian.

## 1.2 Connecting to the Beaglebone

In most cases, connecting to the Beaglebone will be done over a protocol called "ssh" (short for Secure SHell), which is a secure connection to remotely access a shell session.

A remote connection is required since in most cases the Beaglebone will not have a screen and keyboard connected to it. While this can be set up, usually there won't be space or access to the Beaglebone when it is built into a rocket or enclosed within a command console.

### 1.2.1 Using a Mac

While the Mac operating system is not the ideal OS to work on engineering projects, it can easily be used to connect via ssh. However, there are some things to consider. Since Apple has changed the USB ports on some of their computers to USB-C, it is required that a way to connect to the Beaglebone's USB mini-B port is available.

This can be achieved with either a dongle and a USB-A to USB-mini-B cable or a USB-C to USB-mini-B cable. The former seems to be the more available option since USB-C to USB-mini-B cables are likely not widely available.

The connection itself is rather easy, however

First open the Terminal, which can be found in the Utilities folder under Applications. A white window with black text should pop up, where commands can be entered.

To connect to the Beaglebone, type the command:

---

```
ssh debian@192.168.2.7
```

---

A prompt may ask to accept an ECDSA key fingerprint. For this answer "yes", and the prompt should not come up again for the same Beaglebone.

Depending on how the Beaglebone was configured previously, either a welcome text should be displayed, or a new line of text reading "Password:" will be prompting for a password.

If a password is required, then enter "temppwd". When typing, it may seem as if nothing is getting typed, but rest assured since this is intended, and no text will be displayed until the enter key is pressed. (Note: this is an engineering club and not a computer security club. Just make sure the rocket is never connected to the internet when primed and ready to launch!)

At this point a connection to a shell session should be established. If not, it may be that some settings need to be changed under network settings.

## 1.2.2 Using Windows

With recent updates to windows, an ssh client has been added to Windows itself. Alternatively, you may also use a separate program to do this. The latter tends to be more convenient since these programs can save sessions and thus require less to be typed to start a session.

### 1.2.2.1 Using Powershell

PowerShell is the new and updated command line interface that comes with windows since windows 7, which is meant to replace the old cmd.exe.

To open a PowerShell session, search for "PowerShell" in the corresponding windows search bar (found in the start menu or directly on the bottom right).

An application called "Windows PowerShell" should appear. Click on it to launch it. (Make sure that the application is not called "Windows PowerShell ISE", which is a debug/ide style wrapper to the command shell and not needed in this case).

Once the shell window has opened you should see a blue window with white text on it. The last line should be reading something like:

---

```
PS C:\Users\Your_Username>
```

---

At this point type the command:

---

```
ssh debian@192.168.2.7
```

---

A prompt may ask to accept an ECDSA key fingerprint. For this answer "yes", and the prompt should not come up again for the same Beaglebone.

Depending on how the Beaglebone was configured previously, either a welcome text should be displayed, or a new line of text reading "Password:" will be prompting for a password.

If a password is required, then enter "temppwd". When typing, it may seem as if nothing is getting typed, but rest assured since this is intended, and no text will be displayed until the enter key is pressed. (Note: this is an engineering club and not a computer security club. Just make sure the rocket is never connected to the internet when primed and ready to launch!)

### 1.2.2.2 Using PuTTY

If you choose to use a separate application, then a popular ssh client is "PuTTY", which can be downloaded under:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

After installing or extracting PuTTY, launch "PUTTY.EXE", which should popup a configuration window. Here type "192.168.2.7" under Host Name, "22" under Port and select SSH under Connection type.

Once done, either save the session to expedite future connections, and press Open.

At this point the Configuration window should have closed, and a new window with black background and white text should appear. In addition, a prompt may ask to accept an ECDSA key fingerprint. For this press Accept, and the prompt should not come up again for the same Beaglebone.

The new window should now be prompting for a username. For this enter "debian" and press enter.

Depending on how the Beaglebone was configured previously, either a welcome text should be displayed, or a new line of text reading "Password:" will be prompting for a password.

If a password is required, then enter "temppwd". When typing, it may seem as if nothing is getting typed, but rest assured since this is intended, and no text will be displayed until the enter key is pressed. (Note: this is an engineering club and not a computer security club. Just make sure the rocket is never connected to the internet when primed and ready to launch!)

## 1.3 Using Debian

Once you have successfully connected and logged into a shell session, you can start using Debian.

At first it may seem daunting to use a command line only operating system, however, once you get used to it, it may seem to be more convenient to use when doing repetitive tasks. Instead of having to repeat the same set of mouse clicks and typing, now everything can be easily automated and streamlined thanks to command line scripting. More on that later.

### 1.3.1 Basic commands

Until you get to advanced use cases, however, it is beneficial to learn some basic commands:

#### 1.3.1.1 ls

The command "ls" will print all the files and directories in the current directory (LiSt files). This command will probably be one of the most used, since you will need to know what is available to you where you currently are.

If you type "ls -a", now you will also see hidden files and directories as well as the directory "." and "..". The two aforementioned directories are not really directories, but rather shorthand for the current directory and parent directory respectively.

If you type "ls -l", you will now be shown more detail about each file and directory, such as its access privileges and last modification date.

#### 1.3.1.2 cd

The command "cd" allows you to change your working directory (Change Directory). Again, one of the most used commands, since it enables you to traverse your files.

For example, after you used "ls" to find out what directories are available, you can type "cd coolFolder" to enter the "coolFolder" directory.

Also typing "cd /" will bring you to the root directory, and "cd" on its own will bring you to your home directory.

#### 1.3.1.3 cp

The command "cp" allows you to copy files and directories (CoPy). To copy a file named "file1" to a new file called "file2" enter the command "cp file1 file2".

To copy entire directories, enter "cp -r directory1 directory2", which will copy the entirety of the directory "directory1" to a new directory called "directory2".

#### **1.3.1.4 mv**

The command "mv" allows you to move and rename files and directories (MoVe). To move a file called "file1" to a directory called "directory1" and also rename the file to "file2", enter the command "mv file1 directory1/file2".

#### **1.3.1.5 rm**

The command "rm" allows you to remove files and directories (ReMove). To remove a file called "file1", type "rm file1".

To remove an empty directory called "directory1", type "rm directory1". If you want to remove a full directory called "directory2", type "rm -rf directory2".

NOTE: Do be careful when removing files and directories, since Linux does not forgive accidental removals. Once a file is gone, it is gone and any recovery is either extremely difficult or dependent on specific circumstances.

#### **1.3.1.6 pwd**

The command "pwd" will print out the path to your current working directory (present Working Directory). Useful if you want to copy files from a faraway destination to another, since in those instances a full path may be needed.



### 1.3.2 Command Line Text Editors

Sometimes it is not enough to simply move around files and directories, and files need to be modified. In graphical operating systems, this is easily done using a simple text editor. Well, in Debian there also are some text editors that are purely command line based.

#### 1.3.2.1 nano

Nano is the most user-friendly text editor in the command line interface. To open a file called "file1", simply type "nano file1" and the nano interface should pop up with whatever was in that file. If file1 didn't exist, nano also creates it once saved.

In nano, a couple of helpful bars are shown at the bottom. These show the commands available to you, such as how to save and how to exit the editor. For instance, the command "O" called "Write Out", writes all changes to the current file and the command "X" called "Exit" will exit the editor. When exiting, if changes have been made since the last save, nano will prompt if the current file should be saved or not.

Also, there is a single line at the top of the editor, that shows the current nano version and the opened file.

#### 1.3.2.2 vim

Vim is the command line text editor that is closer to a graphical text editor called "Emacs". It is regarded as the quicker editor, since it comes with a lot of commands to use to manipulate text.

To open a file called "file1" with vim, simply type "vim file1". You may immediately notice that there is not convenient list of commands at the bottom, unlike in nano.

In fact, it is not obvious how to even exit or save. For this press the escape key and type ":q". Similarly, to save type ":w".

This is because unlike editors you may be familiar with, vim has two modes, namely the command mode and the insert mode.

While it may seem like the harder to use text editor, this is partly due to its simplicity. While nano has a nicer user interface, it also relies on special instructions that get sent over your shell connection. If your setup only supports text-only, then vim may be the better choice.

### 1.3.3 Bash Scripting

While manually typing out commands is all well and good, but when certain commands have to be done in a specific order periodically, it can be helpful to write a script that does all this for you.

This is where bash scripting comes in. Basically, it is possible to create a simple file that contains all the commands that should be executed in a particular order.

To do this, simply create a file ending with ".sh". Although it's not strictly required to do so, the ".sh" extension signifies that the file is indeed a script.

In that file the first line is commonly "#!/bin/bash" which signifies what command line interface to use. This line can be left out, but it is advised to have it, since systems with other scripting interfaces may interpret commands differently.

Beyond this, any combination of commands can be written. As an example, see the contents of the file below:

---

```
#!/bin/bash
clear
echo "Hello World"
pwd
cd ..
pwd
```

---

This script will do the following:

- First, it will clear the screen.
- Next, it will print out the text "Hello World".
- Next, it will print out the current working directory.
- Next, it will switch to the parent directory, but won't be printing out anything to the user.
- Finally, it will print out the new current working directory.

You may realize that we have seen some of these commands before. Since this is not a complete guide on Debian, we have not gone over every command available.

You may also notice that all these lines can be typed directly into the command line and will have the same effect.

That is because, by design, a bash script is simply made to autonomously execute its contents on the command line.

### 1.3.4 Further Reading

To find out more about how to use Debian, simply search on the internet using your favorite search engine.

In fact, (while a large part of the Debian community may look down on me for the following statement) you can usually find simple to understand guides and Q&As if you search for something for Ubuntu instead of strictly Debian. This is since Ubuntu is one of the more popular flavors of Debian since it is more akin to windows out of the box, when used with its graphical user interface.

Speaking of Ubuntu, if you want to play around with Linux on your everyday computer, you may want to install Ubuntu on a virtual machine and just play around.

**2 Telemetry Board**

**3 Actuator Board**

**4 Scate Board**

**5 Floor Board**

**6 Water Board**

**7 Advisory Board**

**8 Mid-Sized Staffed with Volunteering Board**

**9 3 ply construction Board**