Chi.Bio

Software Setup Guide V1.4

Harrison Steel

March 31, 2021

Overview

This document provides instructions for setup of the software and operating system on the Chi.Bio control computer. Required materials are a PC, internet connection, micro-SD card (from 4 to 16 GB) and the control computer itself. These instructions are targetted toward using a windows PC, but can in practice use a PC with a different operating system. In that case you will need to use different approaches to creating the SD card install media and connecting the Beaglebone to the internet (guides exist for both of these on the internet, they are not specific to Chi.Bio). Commands that must be directly input by the user are highlighted in blue, and must be executed as-is (i.e. type them out and press enter). A rudimentary knowledge of linux may be helpful, in particular commands for navigating file directories. Useful ones include `ls` (list the contents of a folder), `cd FOLDERNAME` to enter a particular folder, `cd ..` to move up a folder in the directory (i.e. out of the current folder), `pwd` to see the directory that you are currently in.

Contents

1 Install Beaglebone Operating System

This section provides instructions for installing an updated operating system (Debian 10 with Linux 4.19) on the Beaglebone Black control computer. These instructions are not specific to Chi.Bio, and so if you encounter difficulties you will likely be able to find help on a range of internet guides/forums.

1.1 Create SD card with updated Linux Debian install

1. Download eMMC flasher for Debian 10 for AM3335x Beaglebone version (bone-eMMC-flasher-debian-10.0-iot-armhf-2019-07-28-4gb.img.xz), available from https://elinux.org/Beagleboard:BeagleBoneBlack_Debian#Debian_10_.28Buster.29_-_Weekly. As new versions are released this distribution date/number will change; use the latest version under the "Debian Buster IOT Snapshot" and "AM335x based Beaglebone's and PocketBeagle" subheadings.

2. Use BalenaEtcher to flash the SD card with this operating system, following the instructions here: https://beagleboard.org/getting-started (starting from step # 0.B).

1.2 Set up Beaglebone for Flashing

1. Install drivers for the Beaglebone on your PC. These can be downloaded for various operating systems here: https://beagleboard.org/getting-started#troubleshooting.

2. Download and install Putty (https://www.putty.org/).

3. Plug Beaglebone into PC using mini-USB cable, wait 1 minute for it to boot up.

4. Connect to Beaglebone using Putty; IP address is 192.168.7.2, port 22.

5. Log in to device in Putty terminal, username:password is debian:temppwd.

6. Become super-user by executing sudo su, and using temppwd as password.

7. Navigate to /boot/ (this will require moving up directories until you are in "/", and then executing cd boot).

8. Execute nano uEnv.txt to edit the file uEnv.txt.

9. Make the changes described here under "Flashing eMMC": https://elinux.org/Beagleboard:BeagleBoneBlack_Debian#Flashing_eMMC. Don't worry about the optional update to flasher scripts. In short, you need to un-comment a line (delete the # character) near the bottom of the file (navigate there using the arrow keys). Once this is done press ctrl+x to exit and type y to save.

10. Turn off the Beaglebone by executing shutdown now and unplug it from the USB.

1.3 Flash Beaglebone

1. Put the micro SD card prepared in Section 1.1 into the control computer (with it unplugged).

2. Locate the switch "S2" on the Beaglebone Black; this is the third switch, not the ones labelled "Reset" or "Power". It is located approximately below the text "T1" on the blue control PCB (which you will need to remove to access the switch). While holding this switch plug the Beaglebone into the PCB with its USB cable. Release the switch after 20 seconds.

3. The Beaglebone should now be flashing its onboard memory with the new operating system. The four indicator lights should flash backward and forth until this process is complete.

4. When flashing is complete (should take approximately 15 minutes) the four indicator lights will go off. Now unplug the Beaglebone from the PC, and remove the micro SD card.

2 Installing Packages and Chi.Bio Operating System

2.1 Connecting to the Internet

In order to get internet access on the Chi.Bio computer it needs to be set up to share the internet connection from the connected PC.

1. Log into the Beaglebone using Putty as in Section 1.2.

2. Become super-user by executing sudo su, and using temppwd as password.

3. Set up internet connection sharing with the Beaglebone as described in the "Connecting the Beaglebone Black to the Internet via the USB Connection " section here: `http://ofitselfso.com/BeagleNotes/HowToConnectBeagleboneBlackToTheInternetViaUSB.php`. Note you can stop following these steps after the "Configuring a Nameserver" section. To test the internet is working as expected run command ping google.com and you should see a response (top stop pinging press ctrl+c).

4. If you are working with a different PC operating system have a look on google, there are many guides to connecting a Beaglebone Black to the internet.

2.2 Setting up Chi.Bio Operating System

1. Download a FTP software such as FileZilla (`https://filezilla-project.org/`), connect to the Beaglebone using IP 192.168.7.2, port 22, username:password debian:temppwd.

2. Go to Github (`https://github.com/HarrisonSteel/ChiBio`) and download all files.

3. Copy the files downloaded from Github into /home/debian on the Beaglebone (this should be the default directory Filezilla starts in).

4. Log into the Beaglebone using Putty, make sure you are in the /home/debian directory.

5. Become super-user by executing sudo su, and using temppwd as password.

6. Execute bash setup.sh which is located in /home/debian. This will automatically download and set up a bunch of packages, which may take 30 minutes. It will then reboot the device (which will give a “Server unexpectedly closed network connection” error, don’t be alarmed!).

3 Wrap-up

The sofware side of Chi.Bio should now be ready to go. The above setup process will have enabled the root account, meaning in future you can log in (over Putty or FileZilla using username:password root:root). It will have put all the operating system files required into the directory /root/chibio. To actually run the Chi.Bio operating system you need to navigate to this directory and execute bash cb.sh. Experimental data files will be saved in this directory, and can easily be copied off the device at any time using FileZilla.

4 Other Procedures

4.1 Changing IP Address

If you want to connect multiple Beaglebone Black's to a single PC via USB you may need to change their default USB IP address, such that you can communicate with each independently. The below instructions outline how to change the IP address of the USB connection from 192.168.7.X to 192.168.8.X. The 8 can be changed to a different number (≥ 8) of your choosing to connect more devices.

1. Navigate to /etc/network and edit the file "interfaces" (e.g. run nano interfaces).

2. Change all IP addresses in this file of the form 192.168.7.X to 192.168.8.X (there should be three such entries).

3. Save and close the file (press ctrl+x, type y and press enter).

4. Navigate to /opt/scripts/boot and edit the file am355x_evm.sh.

5. Change all IP addresses in this file of the form 192.168.7.X to 192.168.8.X. There should be three such entries; they are some way down the file so find them by pressing ctrl+w (opens search feature) then searching for 192.168.7.

6. Save and close the file.

7. Run restart now; the Beaglebone should now be accessible via 192.168.8.2 over Putty.

Following the above steps you will need to update the IP address you use in all contact with the device, including when accessing cloud9, the operating system GUI, and also in the command stored in the script cb.sh which is used to start the operating system. If for whatever reason you cannot access the Beaglebone at the expected IP after following the above steps you still access it using an Ethernet cable (and Putty) to figure out what went wrong.

4.2 Setting up Remote Access

If you want to remotely access the host PC that is connected to the Chi.Bio control computer, then it is useful to employ one of the many remote access software packages that have been designed for such a purpose. These allow you to log into the host PC remotely over the internet and see its screen and send keyboard/mouse commands to control it as you would if you were there in person. This is really helpful for checking on experiments remotely. Two effective software packages for achieving this are Teamviewer and Chrome Remote Desktop (though other alternatives exist).

Teamviewer is effective and easy to set-up, and has a free version which you can use fairly reliably. However, they have a very trigger-happy system for detecting what they perceive to be commercial use of their software, which can lead to them unpredictably disabling your account (obviously quite inconvenient). To set it up you need to install their software on both computers (the lab computer, and the one you intend to access it from remotely), which can be Windows/Linux/Mac. You then get a user address/password from the lab computer, which allows you to log in remotely.

Chrome Remote Desktop doesn't have all of the features provided by Teamviewer, but it has pretty much everything you will likely need for use with Chi.Bio, and it is completely free. It is essentially a browser addon for Google Chrome; again you must set it up on both computers, and in this case (I believe) they need to be logged into the same google account. Chrome Remote Desktop is also available on Windows/Linux/Mac, though on certain versions of Linux it can be a bit finicky to set up (because by default it wants to open a new X-session, meaning you cant see the desktop of the computer you hope to access, there are instructions to overcome this here: https://researchxuyc.wordpress.com/2014/07/30/to-show-the-same-display-session-in-ubuntu-by-chrome-remote-desktop/.)