Chi.Bio

Software Setup Guide V2.3

Harrison Steel

April 26, 2022

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 computer with a different operating system. In that case you will need to use different approaches to creating the SD card install media (guides exist for this online but 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.

Note that this is V2 of the Software Setup approach which is (I hope) much easier and straight-forward than the previous version.

Contents

1 Install Chi.Bio Operating System

This section provides instructions for installing an updated operating system on the Beaglebone Black control computer. Note that this operating system uses as its foundation a Debian 10.5/Linux 4.19 image for Beaglebone (assembled by Robert C Nelson), but several changes have been made in the Kernel to improve hardware drivers utilised by Chi.Bio.

1.1 Create SD card with operating system image

1. Download the Chi.Bio operating system image from `https://www.dropbox.com/s/ehj2sqh6v4omoht/ChiBio-OS-2021-06-18.img.xz?dl=0`.

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. Note that if your device has come from Labmaker then the login should instead be root:root.

6. Become super-user by executing sudo su, and using temppwd as password (password should be root if your device is pre set-up by Labmaker).

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) that says "cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh" 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. Note that if your device has come from Labmaker then this edit may already have been made.

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.

1.4 Update with Latest Github Image (recommended)

Though not essential, following the above steps we recommend you perform a **minor** update of the Chi.Bio operating system (i.e. load the latest core files from the project Github) as outlined in Section 2. This ensures any fixes that have not yet made it to the latest operating system image are present on your device.

1.5 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. The specific Chi.Bio operating system files required are in 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 set up as in Section 3.1.

2 Updating Chi.Bio Operating System

The operating system image used to flash the device in Section 1 above includes all software to run the Chi.Bio software *and* the Beaglebone control computer itself. This operating system image is updated frequently, but may be behind the latest version of the Chi.Bio operating system on Github (to which members of the user community can contribute).

As such, there are two methods for updating the Chi.Bio operating system, termed Major and Minor updates, which are outlined below. We recommend performing a Minor update frequently to ensure your system includes any recent bug fixes or features created by the community.

Minor: If you only wish to do a minor update - of the main Chi.Bio scripts or user interface - then:

1. Go to Github (`https://github.com/HarrisonSteel/ChiBio`) and download all files as a zip, and then unzip the files into a convenient directory on your computer.

2. Open up a FTP tool such as Filezilla (as in Section 3.1).

3. Copy the new version of app.py you just downloaded from Github into /root/chibio.

4. If you also want to update the user interface then copy over the files in the folders "static" and "template" into their corresponding folders under /root/chibio/.

Note that if you want to change the contents of these files yourself the easiest approach is to use the Cloud9 IDE that runs on the Beaglebone by navigating in a web browser (on your connected PC) to `http://192.168.7.2:3000/`

Major: If you wish to completely wipe the device and update to the latest operating system then follow the steps in Section 1.1 and 1.3 above.

3 Other Procedures

3.1 Transferring files to/from Chi.Bio with Filezilla

To get files on/off the device generally the easiest approach is to use an FTP software tool such as FileZilla (`https://filezilla-project.org/`). To connect to the Beaglebone in Filezilla use IP 192.168.7.2, port 22, username:root debian:root.

3.2 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 192.168.7.2 to 192.168.8.2. The 8 can be changed to a different number (≥ 8) of your choosing to connect more devices.

1. Navigate to /etc/default and edit the file "bb-boot" (e.g. run nano bb-boot).

2. Change all IP addresses in this file of the form 192.168.7 to 192.168.8. There should be two such entries, one on a line "USB0_SUBNET=192.168.7" (change this to 192.168.8) and another on a line "USB0_ADDRESS=192.168.7.2" (change this to 192.168.8.2).

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

4. Navigate to the main Chi.Bio folder /root/chibio and edit the file cb.sh (i.e. run nano cb.sh).

5. Change the IP address in this file of the form 192.168.7.2 to 192.168.8.2. There should be one entry on the only active line (i.e. the line beginning with command "gunicorn").

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 (i.e. it will become 192.168.8.2:3000) and the operating system GUI (now 192.168.8.2:5000). 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. Alternatively, re-write the operating system using a microSD card to get a fresh start.

3.3 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/.)