Setup and Use Instructions for RPI Drone

**Creating a new drone by copying the SD to a new RPI:**

1. Insert the new MicroSD card into the USB adapter, then the adapter into the RPI USB port.
2. In the RPI desktop, got to Accessories and open SD Card Copier utility.
3. Copy from /dev/mmcblk0
4. Copy to /dev/sdb (or something like that)
5. (don’t partition)
6. Takes about 20 minutes.
7. Then shut down the rpi and remove the new SD card.

**Setting up a new RPI from scratch:**

1. Need an SD card (sold separately). Plug into RPI
2. Flash 32-bit Debian Bullseye onto RPI SD card (do not use the newer version, it’s not compatible with Matlab)
   1. On computer, download Raspberry Pi Imager from Raspberry Pi website
   2. Device 4B, OS Debian Bullseye
   3. Yes to change settings, and put in a username/password
      1. Current Username is bridgeman. Current Password is bridgeman.
3. In Matlab, download and setup *(optional)*
   1. MATLAB Support Package for Raspberry Pi Hardware
   2. Simulink Support Package for Raspberry Pi

*Note: we used Debian Bullseye because we wanted it to be compatible with MATLAB. We aren’t currently using MATLAB anymore, so newer versions could work just as well.*

Connect RPI to:

1. Power (chord or LiPo)
2. Computer monitor (micro HDMI)
3. Keypad (USB)
4. Mouse (USB)

Then you can get into the Raspberry Pi’s interface. Connect to wifi (currently hotspot, working on Duke’s Wifi). To give the RPI access to duke wifi, go to the Terminal and type:

sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf

This gives you access to the wifi that the RPI has info on. Delete all and add the VICON\_Server network:

network={

ssid = “VICON\_Server”

scan\_ssid=1

psk=”duke-robotics1”

key\_mgmt=WPA-PSK

}

Then press Ctrl+O, Enter, Ctrl+X to save and exit. Then Restart the RPI with command

Sudo shutdown -r now

To enact the changes. Note: It’s necessary to connect to VICON\_Server wifi so that the rpi can get data from the Vicon system. A similar setup could be used to connect to other wifi, though the Duke wifi is more complicated and requires someone’s username and password, which can change. So it’s best to always use the Vicon wifi.

Next, we need to set up a static IP address (otherwise it changes most times you reboot, which is annoying and confusing). Type

hostname -I

to get the current IP address. Write it down as STATIC\_IP. [**Important note:** it may be advisable to choose a different static\_IP with a higher last number to avoid overlap with new devices. Each time a new device connects to the wifi, it is given an IP address that’s not currently taken. These start at XXX.XXX.X.100, and increments up to XXX.XXX.X.255. If you set the rpi to a low number, like 104, it’s probable a new device will be assigned that value while the rpi is disconnected. Which leads to an error and a security warning. However, if you choose a high number like 230, it’s unlikely that a new device will be assigned that while the rpi is disconnected. This seems like a problem that should have a better back-end solution, but it was observed in practice. Therefore, the current drone is set to 230.] Then type

ip r | grep default

to get the gateway IP address. Write down the first IP address that’s returned. Then type

sudo nano /etc/resolv.conf

to get the router’s DNS IP address. Write this down, then press Ctrl+X to close the file. Next, type

sudo nano /etc/dhcpcd.conf

This opens the file with the configuration file. At the bottom, add the following lines:

interface NETWORK

static ip\_address = STATIC\_IP/24

static routers=ROUTER\_IP

static domain\_name\_servers=DNS\_IP

For NETWORK, put wlan0 if using wifi or eth0 if using ethernet.. For STATIC\_IP, put the static IP address that you want. It’s recommended to use your current IP address, since it’s guaranteed to be unique at least locally. For ROUTER\_IP, put the gateway IP, and for DNS\_IP, put the DNS IP, which may be the same as the gateway IP. Then press Ctrl+X then Y then Enter to close and save. Then

sudo reboot

To implement the changes. Once repbooted, test the SSH connection using the following section.

The static IP address for the first rpi is 192.168.0.230. Its username and password are both bridgeman.

**Controlling the RPI Remotely:**

On any other computer, go to command prompt (terminal) and type:

ping 192.168.0.230

It should give a reply to let you know it’s connected correctly. Then SSH in by typing

ssh [bridgeman@192.168.0.230](mailto:bridgeman@192.168.0.230)

password: bridgeman

Now you can control the RPI from a separate computer

**Cloning a Git repo to the RPI**

*Alex Penne did this part, but Github is pretty well documented, so it shouldn’t be too hard to figure out if you need to do it again.*

**Using Github from commandline**

Before working on the code, pull so you know you’re working on the latest version:

git pull

After working, commit and push. If you created new files, you may need to add them before committing

Git add *file* *(you can do ‘git add .’ to add all the files in the current directory)*

git commit -m “*message*”

git push

**MATLAB Compatibility**

*(This section is archived because we aren’t using MATLAB anymore, but we may in the future).*

On the Windows 10 computer, the versions of Matlab and Bullseye are compatible, so the setup above works. On the Windows 7 computer, the version of Bullseye is not supported by Matlab (Matlab is too old). So I had to implement a patch provided here:

<https://www.mathworks.com/matlabcentral/answers/1946623-error-in-opening-matlab-device-file-for-led-trigger-source>

By the MathWorks MATLAB Hardware Team. Part of the patch is on the RPI terminal, and will need to be done on every SD card, probably after configuring on the Windows 10 computer.

cd /opt/MATLAB/

sudo pkill mwUnifiedSrvr

sudo rm -rf \*

The other half of the patch is on the desktop and probably only needed to happen once, so we won’t have to do it again unless we switch to another old computer.