Setting Up a Raspberry Pi with WPLibPi and MonsterVision

# Install WPILibPi onto a microSD Card

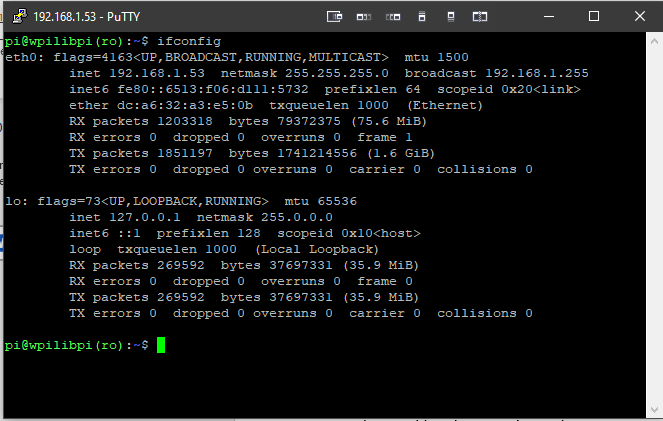
Note: you may need an adapter to mount the microSD card in your computer.

* Download the WPILibPi image from: <https://github.com/wpilibsuite/WPILibPi/releases>. Scroll to the bottom of the page and find WPILibPi\_image-v2021.2.1.zip. Extrace the .img file and put it somewhere convenient.
* Download and install Win32 Disk Imager from: <https://sourceforge.net/projects/win32diskimager/>. Note: this is a Windows program.
* Launch Win32 Disk Imager and use it to write the WPILibPi image to the microSD card. WARNING: Choose the correct output device. Choosing your main hard drive will wipe out your system!
* Once the card has been written, be sure to properly eject it before removing it from your computer.

# Connecting to the Raspberry Pi

The WPILib image is very minimal and FRC-legal. This means it has **no** GUI and both the Bluetooth and Wi-Fi radios are disabled. In order to talk to the RPi, we’ll use PuTTY over wired Ethernet.

* Install the above microSD card into the Raspberry Pi.
* Plug your RPi into your home network using Ethernet. The RPi does not have to be in the same room, though it would be convenient. The first boot of the RPi will take a minute or so as it resizes the file system to the actual size of the microSD card. Subsequent reboots require about 20 seconds.
* Log into your router and look up the IP address that was assigned to the RPi.
  + If you cannot determine the RPi’s IP address from your router, you can plug in a keyboard and monitor. Log into the RPi and type the command: **ifconfig**. You should see something like the following:

Under the eth0 heading, find the inet entry. This will be the RPi’s IP address (192.168.1.53, in this example).

* Now use PuTTY to log into the RPi with the IP address just discovered. The username is **pi** and the password is **raspberry**.
* Open a web browser and browse to the RPi’s IP address. You should see the WPILibPi’s configuration page.
* Note that at the top of the configuration page there is a pair of radio buttons labelled **Read-Only** and **Writeable**. Select **Writeable** in order to be able to modify further settings.
* Using the WPILib web interface, go to the Vision Settings tab.
  + Set the **Team Number** to 2635. This setting will only be visible if the **Client** switch is turned on.
  + If there is no roboRIO, turn off the **Client** switch. Otherwise, leave the **Client** switch on.
  + Remove the entry for the RPi Camera 0, if it is present (assuming there’s no physical device).
  + Click **Save**

# Installing Support Software

* Using the WPILib web interface, make the SD card writable as described previously.
* Use Putty to log into the Raspberry Pi and run the following commands:

sudo apt-get install git python3-opencv libatlas-base-dev

pip3 install depthai

git clone <https://github.com/luxonis/depthai.git>

cd depthai

python3 install\_requirements.py

echo 'SUBSYSTEM=="usb", ATTRS{idVendor}=="03e7", MODE="0666"' | sudo tee /etc/udev/rules.d/80-movidius.rules

sudo udevadm control --reload-rules && sudo udevadm trigger

# Installing MonsterVision

* On the build machine, get the MonsterVision repository from GitHub at <https://github.com/Lakemonsters2635/MonsterVision>
* On the build machine, create resources.zip by compressing the MonsterVision/resource directory. Be sure to include all subdirectories.
* Using the WPILib web interface, go to the Application tab.
  + Upload and unpack resources.zip.
  + Upload cone\_nt.py (or whatever the app is called).

SSH into the Raspberry Pi machine and edit runCamera in the /home/pi directory. After the “sleep 5” line, add these 2 lines:

Kill -9 python3  
python3 cone\_nt.py &

Note: the final & is important.

chmod +x cone\_nt.py

Using the WPILib web interface, go to the Vision Status tab and click “Kill.” After about 10 seconds, the Shuffleboard should come to life. If you want to see console output from the RPi, turn on the **Enable** switch.

As a final test, reboot the Raspberry Pi and make sure the MonsterVision subsystem restarts.

# Accessing the Raspberry Pi When Connected to roboRIO

Once the RPi has been configured and is on the robot’s network with radio and roboRIO, you can access it as **wpilibpi.local**. When using the browser, it may be necessary to entry the fully-decorated URL: [**http://wpilib.local**](http://wpilib.local). You should also be able access it using Putty using the same name.

# References:

<https://docs.luxonis.com/en/latest/pages/troubleshooting/>

<https://docs.luxonis.com/en/latest/pages/api/>

<https://github.com/wpilibsuite/WPILibPi/releases>

<https://frcdocs.wpi.edu/en/latest/docs/software/vision-processing/wpilibpi/the-raspberry-pi-frc-console.html>

<https://robotpy.readthedocs.io/projects/wpilib/en/2018.0.13/wpilib/SmartDashboard.html>

# Setting up VSCode for Remote Development on WPILibPi

These instructions assume you are using Windows as the local machine and a Raspberry Pi running the WPILibPi image on the target.

* On your local machine, install the VSCode extension: Remote Development
* Create your local SSH key pair via:

ssh-keygen -t rsa -b 4096

When asked about a passphrase, leave it blank.

* On your Windows machine, execute these commands in a **local PowerShell**.

$USER\_AT\_HOST="your-user-name-on-host@hostname" 🡸 Please substitute pi@<local\_ip\_address>

$PUBKEYPATH="$HOME\.ssh\id\_rsa.pub"

$pubKey=(Get-Content "$PUBKEYPATH" | Out-String); ssh "$USER\_AT\_HOST" "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '${pubKey}' >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"

Now test the connection by entering: ssh pi@<local\_ip\_address> You may be asked the raspberry pi’s password the first time.. If so, enter it and then log out and try the ssh command again. It should log you in with no questions.