Pi-Bot Wars

Build Instructions

Eller MIS

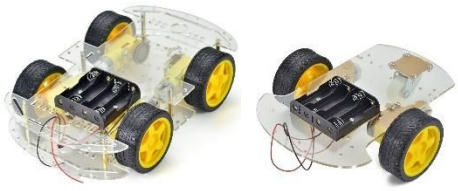
Kroll

There will be five main parts to this lab. Volunteers are available to help, please feel free to ask for assistance at any time. We are here to help make this an enjoyable experience for you. A summary of the steps are detailed below:

1) Assemble the Chassis
2) Connect the Motorhat
3) Install and configure the OS
4) Install the Motorhat libraries on the PI
5) Configure remote control with the Wii remote

Your complete robot will look something like the below examples. Feel free to place the components how you see fit on the robot. Experiment and have fun.

1. Assemble the Robot Chassis



The Chassis is the body of the robot. It will hold the wheels, Raspberry Pi, batteries and everything else.

Follow the instructions for your Chassis to assemble it. We have extra instructions available for the Two Wheel Chassis Kit and the Four-Wheel Chassis kit at the front of the room.

Tip: Placing some hot glue where the wires are connected to the motor can help protect the wires from breaking off. See a volunteer for help if needed.

2. Connect the Motorhat

The Motorhat is what connects the Raspberry Pi to the Motors. We have two motorhats we recommended for this lab, the Pololu DRV8835 that supports 2 wheels, and the Adafruit motorhat that supports 4 wheels. We also have instructions for the MotoZero motorhat. Each motorhat kit must be assembled before connecting to the Raspberry Pi.

Pololu DRV8835 Assembly

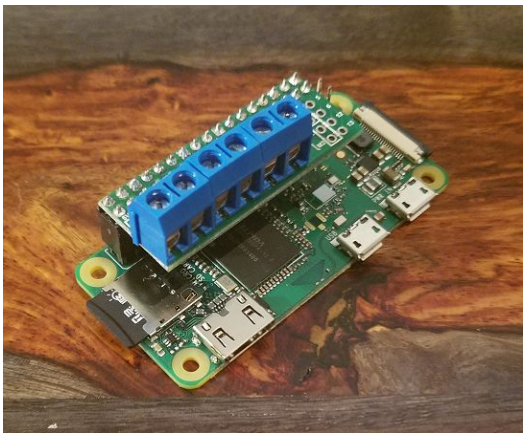
(https://www.pololu.com/product/2753)

The Pololu ships with female headers and three terminal blocks that must be soldered to the board. If you need help soldering this, please see a volunteer at the soldering table. Once soldered, it will look like the below picture:

https://www.pololu.com

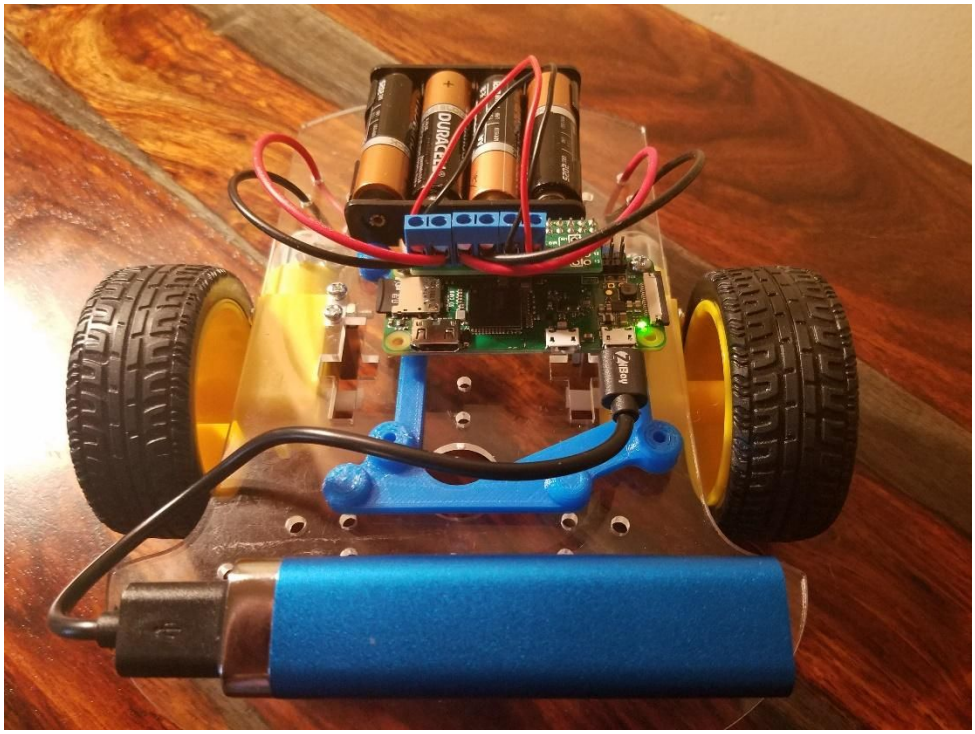
Once it has been assembled, it will need to be placed on your Raspberry Pi like the below pictures:

Raspberry Pi 3



Raspberry Pi Zero W

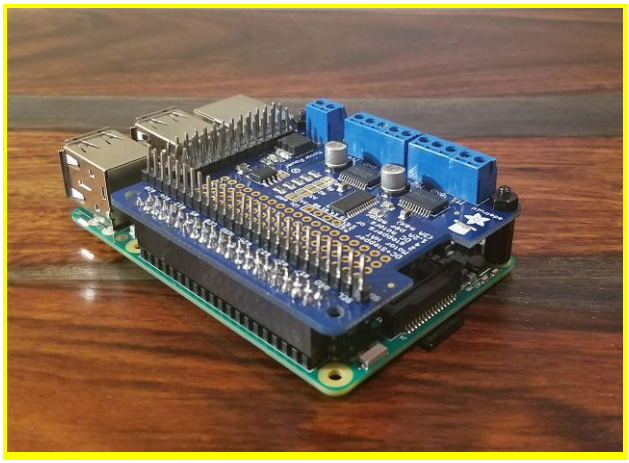
Next, mount it to the chassis and connect the motors and battery (for weight and balances reasons, the batteries are best placed over the canister wheel). You can use duct tape to secure it, or a 3D printed mounting bracket. Both are available at the table in the front of the room. Make sure you connect the battery correctly. The red goes into the far right, and the black next to it, like the picture below:



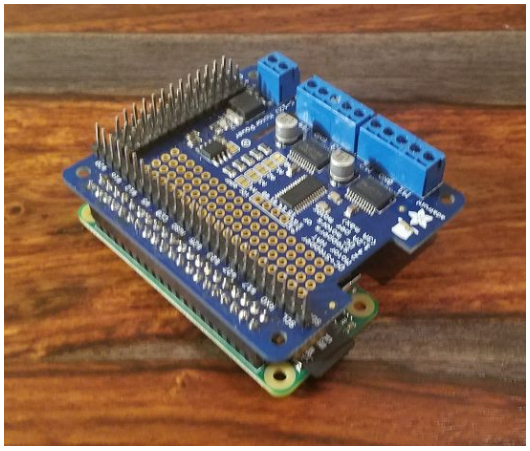
Adafruit Motor Hat

(https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi)

The Adafruit Motorhat ships with female headers and five terminal blocks that must be soldered to the board. If you need help soldering, please see a volunteer at the soldering table. Once soldered and assembled on the Raspberry Pi it will look like the below picture:



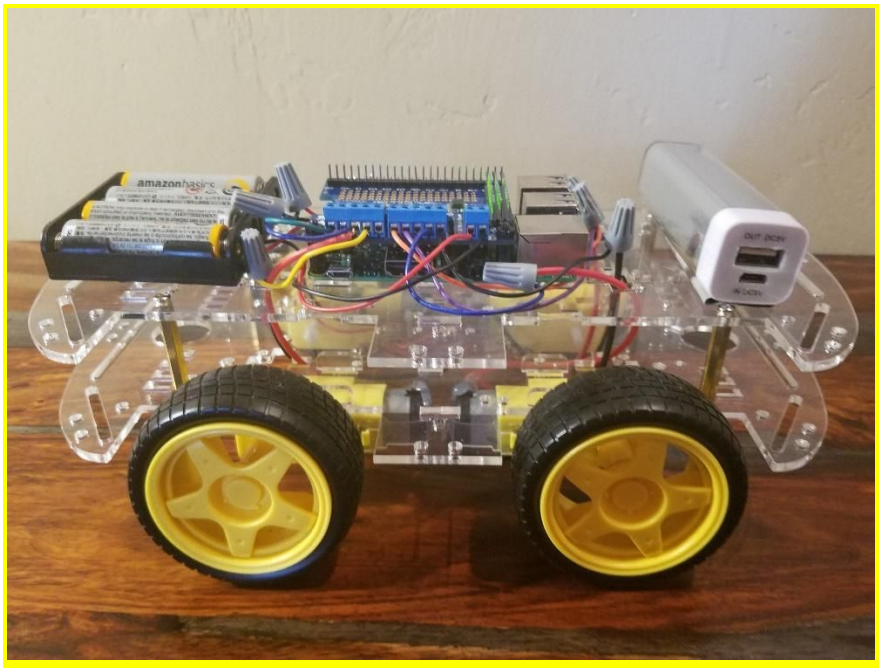
Raspberry Pi 3



Raspberry Pi W

Next, the motors and battery will need to be connected. They are label on the Motorhat by the blue terminal jacks. Make sure and connect the motors in the following way:

M1 = Right Front
M2 = Left Front
M3 = Right Rear
M4 = Left Rear



Tip: If your wires won't stay in the terminal jacks, a touch of solder on the tips will help.

3. Install and configure the OS

The Raspberry Pi does not come with any operating system so we will need to install the operating system on the SD Card. We will be using Raspbian Jessie Lite as the operating system. If you have a laptop, you will need to download the following software or grab a thumb drive from the front table with the software. We also have extra laptops available for use to for this step.

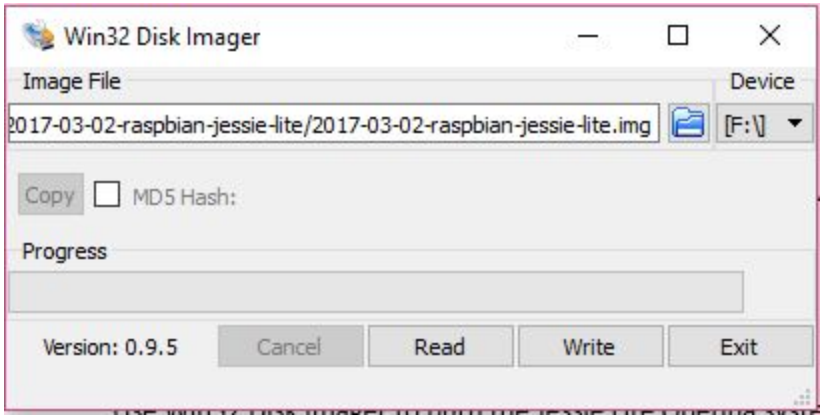
Software:

Win32 Disk Imager: https://sourceforge.net/projects/win32diskimager/

Jesse Lite: https://downloads.raspberrypi.org/raspbian_lite_latest

Putty: http://www.putty.org/

Place your SD Card into an SD Card reader (we have extra SD Card readers if you need one at the front table). Launch the Win32 Disk Imager which should be on the Computer Desktop. Use Win32 Disk Imager to burn the Jessie Lite Operating system to the SD Card. Select the raspbian-jessie-lite.img file from the desktop:



Place the SD card into the slot on the Raspberry Pi.

Now find an available keyboard and monitor station to use. Plug the network cable, monitor and keyboard into the Pi. Plug in the micro-usb power supply to turn on the Pi. The Pi should boot and you will see a terminal screen. Login with the default username "pi" and password "raspberry".

We will be using various commands in this lab. If you would like more details on what the commands do, please ask one of the helpers. Below is an explanation of what some of the commands mean:

sudo <command> : run the command as root/administrator

cd: change into the specified directory

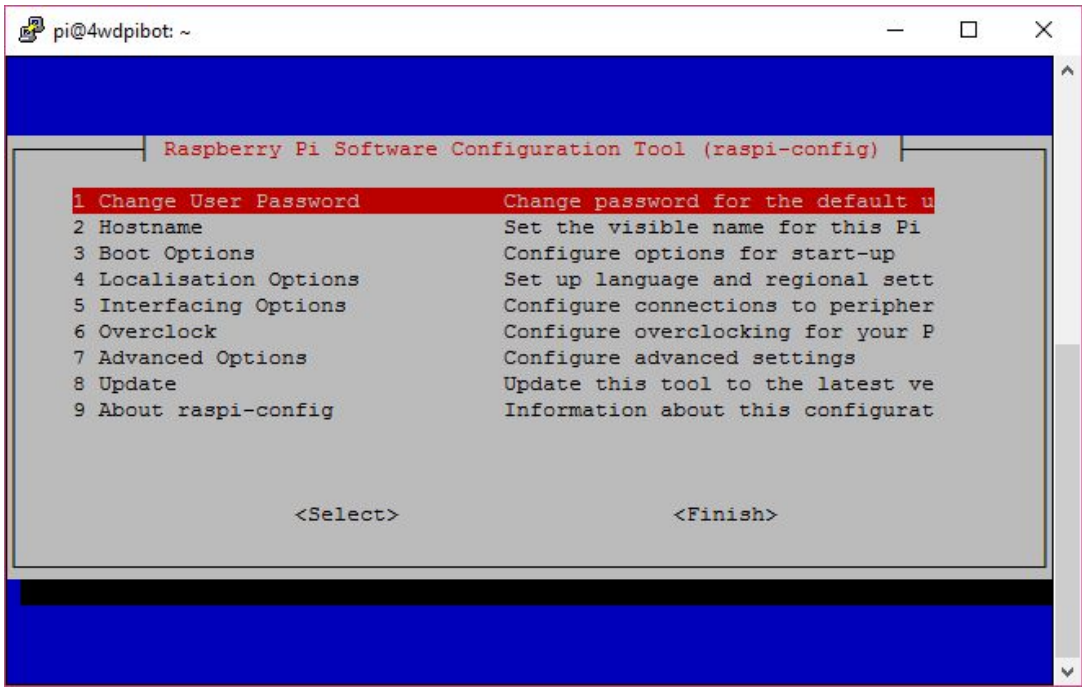
nano: run the nano text editor

The first thing we will do is a quick update:

```
sudo apt-get update
```

There are several configuration settings that will need to be made. We will use the Raspberry Pi Configuration tool to make these settings.

```
sudo raspi-config
```



First, to keep people from logging into your Raspberry Pi, change the default password. Choose “Change User Password” from the menu and follow the instructions.

You will need to change the hostname to something unique. You will use this hostname later to find your Raspberry Pi on the wireless network. Choose “Hostname” from the menu options and follow the instructions.

SSH is a way to connect to the Raspberry Pi in “headless mode” without the monitor and keyboard. SSH

is turned off by default and will need to be enabled for later use.

Choose “Interfacing Options” then “SSH” then “Yes”

If you are using the **ADAFRUIT motorhat** enable IC2:

Choose “Interface Options” > “I2C” > “Yes”

When all these have been completed, select Finish to exit the menu and reboot.

Optional: Configure Headless Mode

If you have your own laptop and want to connect to the Raspberry Pi using your laptop rather than the monitor and keyboard you can configure the Raspberry Pi to connect to our Ad-Hoc Wireless network. This will also free up some of the lab machines for other users while you continue with the next few steps. To do this, you will need to edit a file giving the Raspberry Pi the network ID and password:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

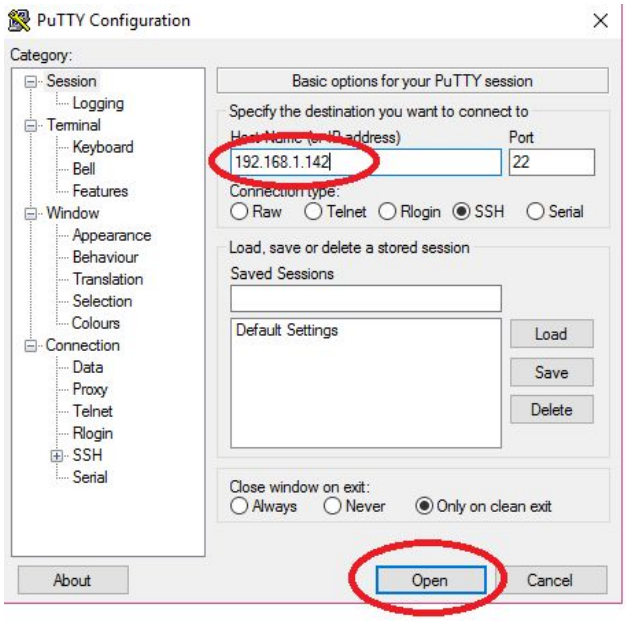
Go to the bottom of the file and add in the following:

```
network={
    ssid="NSA Van 4"
    psk="STEMROCKS"
}
```

When you are done editing the file, select Ctrl-X to exit, then “Y” and “Enter” to save. When you reboot, it will connect to the wireless network.

You may now connect to the Raspberry Pi using the program Putty from your Laptop. To do so, you will need to see what your IP address is once you reboot the Pi and it connects to the wireless network. There will be a monitor at the front table that lists out the IP addresses and Hostnames. Come find out what your IP address is by looking up the unique hostname you gave your system earlier.

Launch Putty, and supply your IP Address then select Open:



You will now be able to log into the Raspberry Pi using the username “pi” and the password you supplied earlier:

4. Install the Motorhat libraries on the PI

****Follow the relevant section for your Motorhat****

Populu Motorhat DRV8835

(https://www.pololu.com/product/2753)

The Populu DRV8835 can control two DC Motors. In order to use it, the Wiring Pi python library must be installed:

```
sudo apt-get install python-dev python-pip
sudo pip install wiringpi
```

Next, install the Populu DRV8835 python library

```
sudo apt-get install git
git clone https://github.com/pololu/drv8835-motor-driver-rpi.git
cd drv8835-motor-driver-rpi
sudo python setup.py install
```

Test your motors by using the example script included in the Populu DRV8835 library

```
sudo python example.py
```

******If the wheels turn the opposite way as expected, simply switch the wires for that wheel in the terminal blocks. Your wire colors may not match the picture, and that is OK.******

Adafruit Motor Hat

(https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi)

Install the Adafruit Motorhat library:

```
sudo apt-get install python-dev
sudo apt-get install git
git clone
https://github.com/adafruit/Adafruit-Motor-HAT-Python-Library.git
cd Adafruit-Motor-HAT-Python-Library
sudo python setup.py install
```

Test your motors by using the example script for this lab:

```
cd ..
git clone https://github.com/mdegrazia/MotorHats
cd MotorHats/Adafruit/
sudo python Adafruit_test.py
```

Ensure that the wheels are turning the correct way. If not, simply switch the wires for that wheel in the terminal blocks.

MotoZero

(https://cdn.shopify.com/s/files/1/0176/3274/files/MotoZero_User_Guide_1.2.pdf)

The MotoZero uses the python GPIOZERO library. This can be installed by using the following:

```
sudo apt install python-dev
sudo apt-get install python-pkg-resources
sudo apt-get install python-gpiozero
```

Test your motors by using the example script for this lab:

```
sudo apt-get install git
git clone https://github.com/mdegrazia/MotorHats
cd MotorHats/MotoZero
sudo python motozero_test.py
```

5. Remote Control with the Wii Remote

We will be using a Wii Remote to control the wheels. The Wii remote will connect to the Raspberry Pi using Bluetooth. The Raspberry Pi 3 comes with Bluetooth. If you have an older Raspberry Pi, make sure your Bluetooth dongle is plugged in before proceeding. Note – Third party Wii remotes and Wii remote plus MAY NOT WORK. Make sure you have an original Nintendo Wii Remote.

To verify that Bluetooth is up and running correctly type the following:

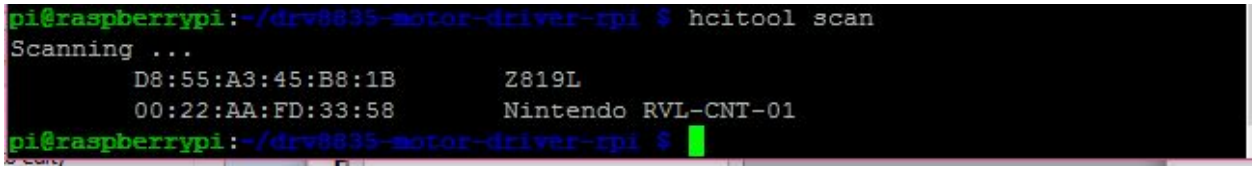
```
hcitool dev
```

If Bluetooth is working correctly, you should see something like:

Next, check to see if Bluetooth will recognize your Wii Remote. Power on your Wii Remote and press the 1+2 on it which will put the Wii Remote in discovery mode.

```
hcitool scan
```

After a few seconds (it may take a couple of times hitting the 1+2) you should see something similar to the following. Look for the line that says “Nintendo RVL-CNT-01” or similar.



Next, we will install the python library that will control the Wii remote, CWiiD.

```
sudo apt-get install python-cwiid
```

Once you have that running, you can grab the code that will read the commands from the Wii Remote and control the motors. Follow the instructions for your particular motorhat. Once it has connected, you will be able to steer you robot forwards, backwards, left and right by using the cross on the Wii Remote.

Populu MotorHat DRV8835

```
git clone https://github.com/mdegrazia/MotorHats
cd MotorHats/Pololu
sudo python pololu_wii.py
```

Adafruit Motor Hat

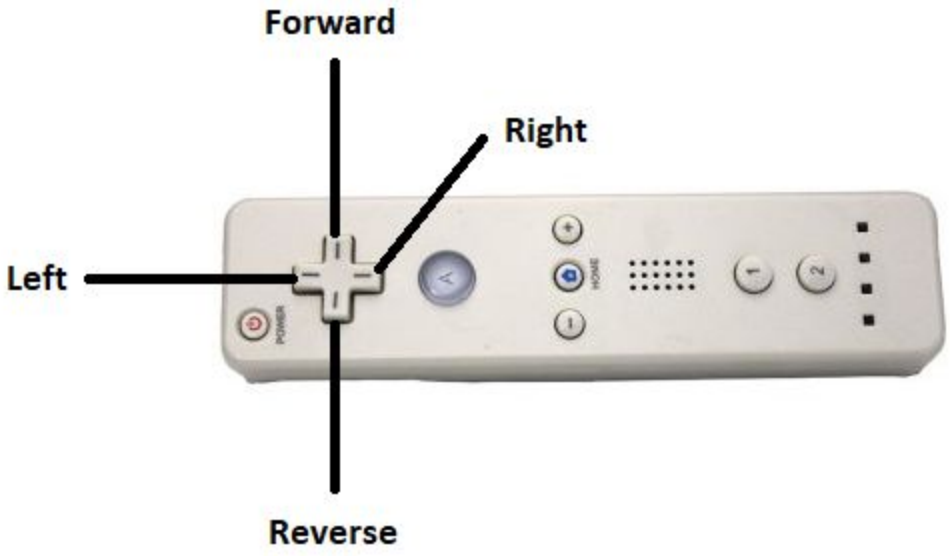
```
sudo python Adafruit_wii.py
```

MotoZero

```
sudo python motozero_wii.py
```

You should see some text scroll on the screen. This is the “position” or what button is currently being pressed on the Wii remote. If you hit the Forward button, you should see the values “512” scroll on the screen quickly. The code reads in this value, and that is how the code tells the robot to move forward.

Hold the Wii remote like the following to control the robot:



Coding Exercises

The code if very simple, which will allow you to configure it and tweak it to your liking. After you have the code working, try to see if you can make changes or tweak the code and answer/implement the follow questions. First, create a backup copy of the code. Substitute motorhat_will.py with the name of your motorhats code from above, IE pololu_wii.py.

```
cp motorhat_wii.py motorhat_wii_backup.py
nano motorhat_wii.py
```

Use Ctrl-X, Y to exit and save to when you are done. Then run the code to see what happens.

0) Hit the various buttons to see what values show on the screen. Can you program the “A” button to do something when it is pressed?

1) Can you make the wheels turn slower?

2) How about faster?

3) Can you make the robot turn sharper?

4) Can you make the robot turn more gradual?

5) How would you go about using one of the buttons to increment or decrement the speed?

6) Can you make the robot keep moving forward when the forward button is selected?

7) Can you program a button to make the robot stop?

Links to documentation on what the buttons on the Wii remote do:

http://www.brianhensley.net/2012/08/wii-controller-raspberry-pi-python.html

wii remote code examples:

http://talk.maemo.org/s2howthread.php?t=60178

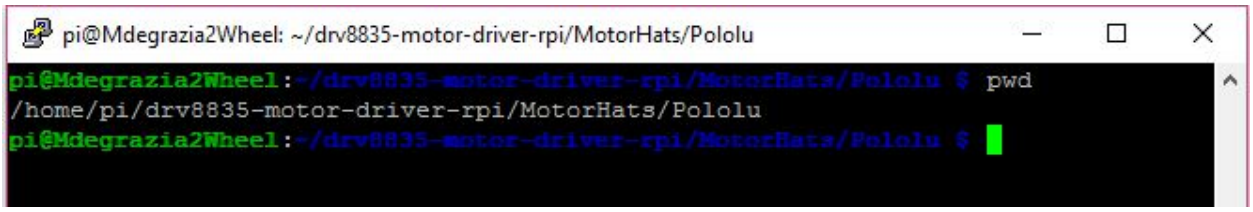
Try googling “python cwiid examples” for code examples.

Autostart Configuration

Now that you have your robot up and running, it would be nice to have the code start automatically when the Pi turns on so you don’t have to connect to the Raspberry Pi, login and start the code each time. These next steps will start the python script automatically whenever the Pi reboots. You can then just turn on the Raspberry Pi and sync the Wii remote to get your robot going.

First, we will need the full path to where the python script is that you would like to run. Make sure you are in the same directory as the script and type:

```
pwd
```



Highlight the path text with your cursor. This will copy the path.

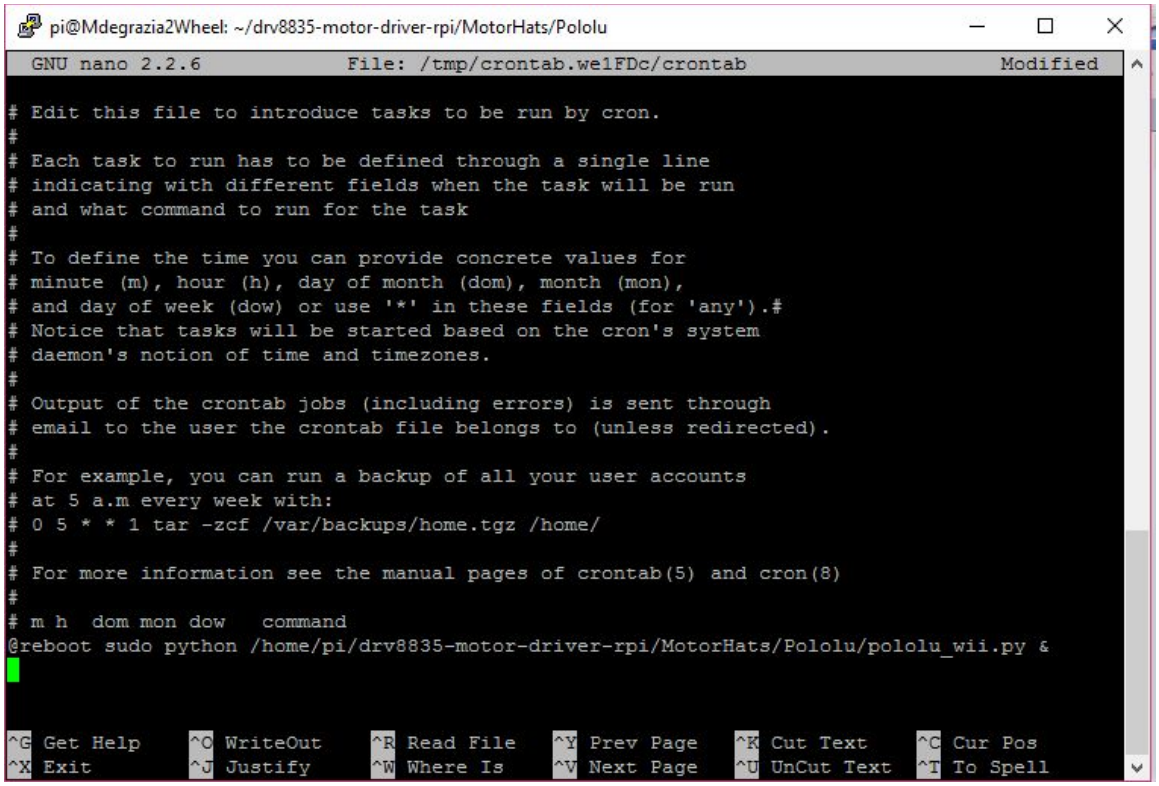
Next, we are going to edit a file called crontab. This will be the file that will automatically start the script on reboot:

```
sudo crontab -e
```

Choose 2 for the nano editor

Add in the following line at the end of the file. To paste the path you copied above, right click your mouse. Make sure and change the python script to match your motorhat wii script.

@reboot sudo python /home/pi/drv8835-motor-driver-rpi/MotorHats/Pololu/pololu_wii.py &



Ctrl-X and Y to save

Now you should be able to reboot your robot and when it starts, you can sync your wii remote and start driving!

Document Author: Mari DeGrazia