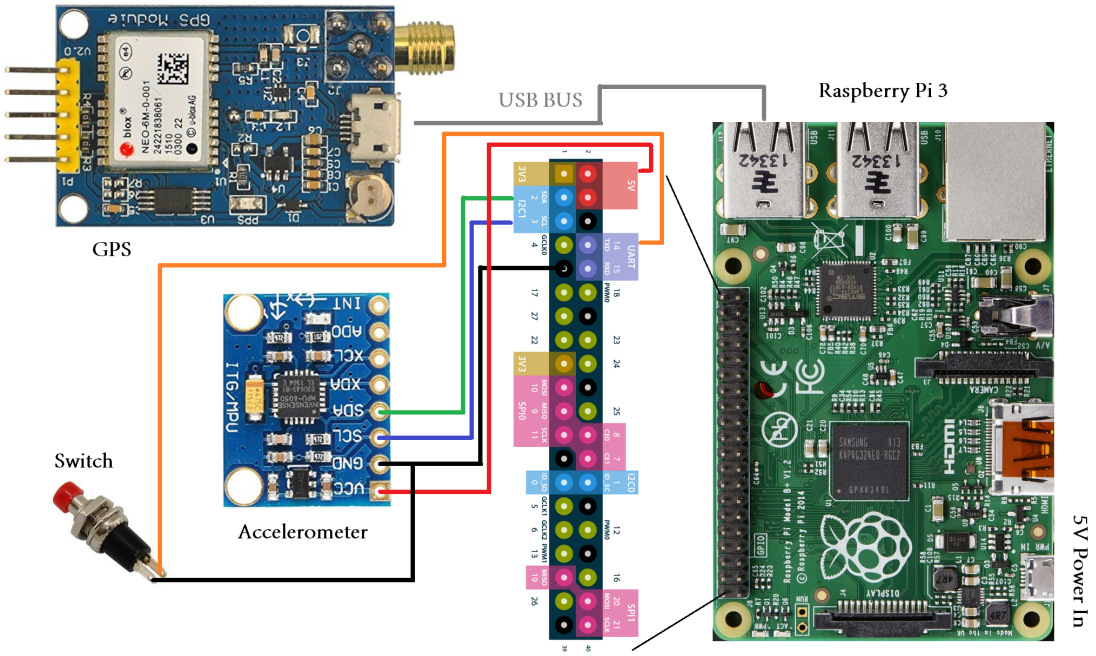
Electric Longboard Sensor Module Build Tutorial

Electrical

Parts List

Raspberry PI 3
https://www.amazon.com/Raspberry-Pi-RASPBERRYPI3-MODB-1GB-Model-Motherboard/dp/B01CD5VC92
GPS Module
https://www.amazon.com/gp/product/B01MRNN3YZ
Accelerometer
https://www.amazon.com/gp/product/B008BOPN40
Power Bank
https://www.amazon.com/gp/product/B00JM59JPG

Schematic



Power Supply

First off the power bank is much larger then it needs to be. It is housed in a large plastic case which can be disassembled.
In pretty much all USB power-banks the battery cells inside are in parallel. Thus we can just take all of the electronics and the cells out of their case, and detach 2 of the cells.
With only 2 cells we have half the power lifetime for out device but this still gives us close to 4 hours of runtime. More than enough for a few longboarding trips.

The output of the battery bank should be hooked up to the 5v input of the raspberry pi.

Accelerometer and Switch

The switch and accelerometer need to be soldered to the Raspberry Pi. Following the schematic we can see that the switch is attached across Pin 14 and GND. Meanwhile, The Accelerometer is powered by 5v and GND. It is attached to the I2C pins of the RasPi (Pin 2,3). Make sure that SDA on the accelerometer matches SDA on the RasPi.

GPS

The GPS can simply be plugged into one of the USB inputs of the RasPi.
The extending range antenna should be screwed onto the GPS Port of the GPS module.

Mechanical

Parts List

PLA or ABS Printer
3D printer filament
8 x 1/4” length plastic fastener screws
4 x 1” length plastic fastener screws
Double sided tape.

Instructions

The Enclosure is designed to be 3D printer with either ABS or PLA.

The STL files for the enclosure are provided (Enclosure Full Top.STL, Enclosure Full Bot.STL).

Once printed, use the 1/4” length screws to mount the GPS, Accelerometer, and Raspberry Pi.

The GPS antenna can be left inside, and taped to the top part of the enclosure using double sided sticky tape or it can be wired to the outside.

Place a line of sticky tape at the far side of the bottom of the enclosure and attached the battery bank.

Using the 1” length screws fasten the enclosure together.

The final product should look like the following picture.

Code

Raspberry PI

Parts List

Raspberry PI with a micro SD Card
5v power supply
Ethernet Cable
Micro SD card Reader

Raspian

To run any programs on the raspberry PI, we will need to give it an operating system. We will be using Raspian.

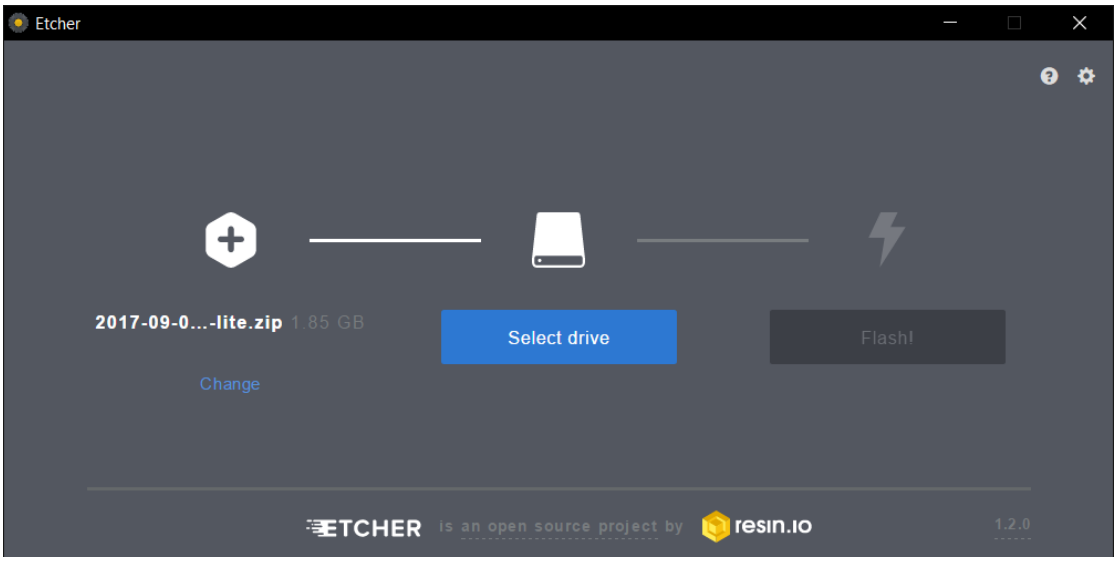
Raspian is a Linux OS distribution based on Debian Linux but updated specifically for raspberry pi.

Download “Raspian Lite” from the official Raspberry Pi website.
https://www.raspberrypi.org/downloads/raspbian/

We will then need to burn the ISO image onto the sd card. To do this we will use a flash software.

Download “Etcher” from the official etcher website.
https://etcher.io/

Select the downloaded Debian Image, the SD card and hit Flash!



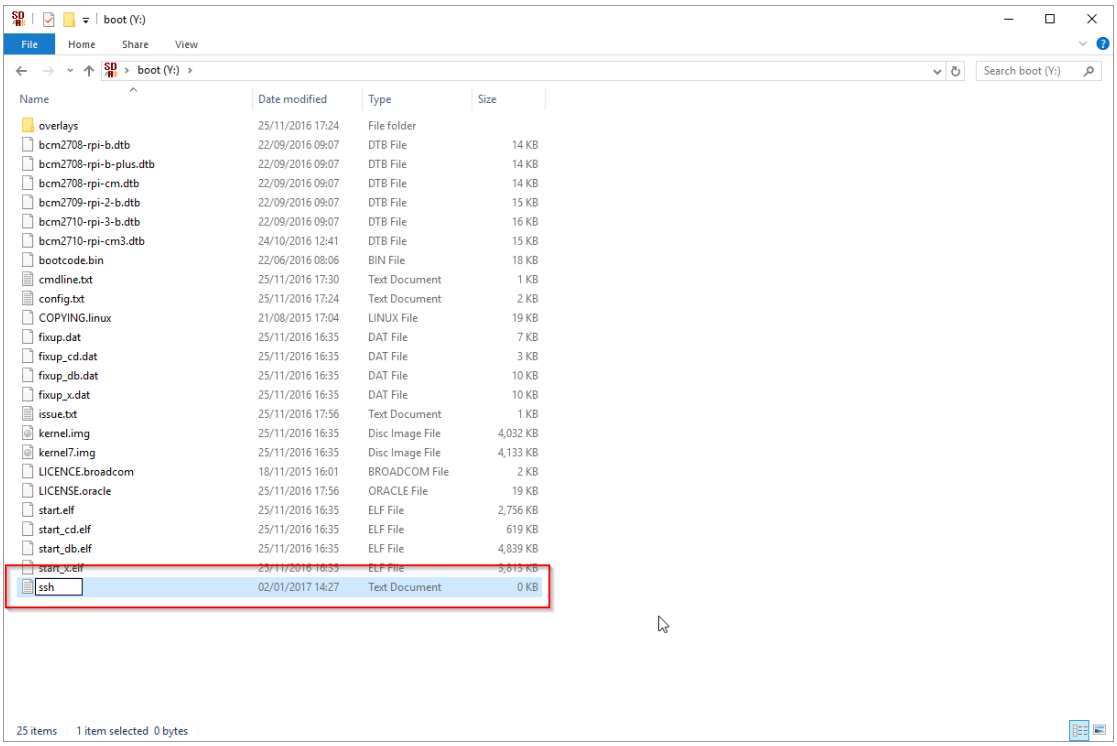
For a more thorough set of instructions check out the official Raspi flashing tutorial.
https://www.raspberrypi.org/documentation/installation/installing-images/

SSH

SSH or “secure shuttle” is a program use to access a computer remotely. We will need this in order to be able to control the raspberry PI from a laptop, with no extra hardware such as a keyboard or monitor.

SSH is installed by default on any UNIX system. Thus we just need to enable it.

The simplest way is to create a blank “ssh” document in the boot file on the newly flashed SD card.



To then access the Pi, connect it to your laptop over ethernet.

If you are using a windows computer type the command “arp -a” into cmd in order to search your local LAN network for IPs.

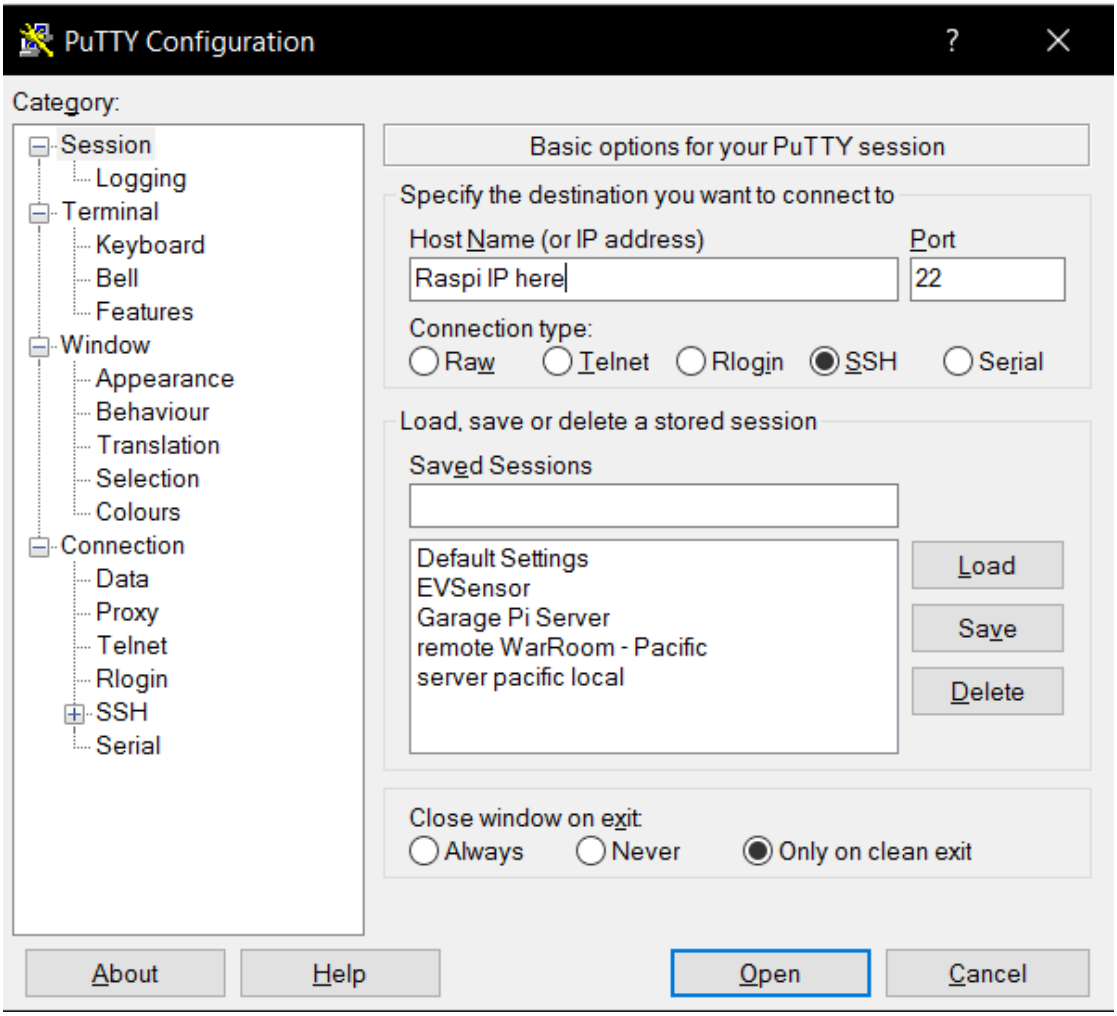
You should be able to identify the IP of the RasPi.

```
C:\>arp -a

Interface: 192.168.1.6 --- 0x12
  Internet Address      Physical Address      Type
  192.168.1.1           9c-3d-cf-c5-e2-c8     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

If you do not already have Putty installed do so now.
http://www.putty.org/

Putty is a program used by windows to communicate over a few different protocols. It allows us to establish a ssh tunnel.



Open PuTTY and input the IP of the pi where is says “Raspi IP here”. The default SSH port is “22”. Hit “Open”

This should establish a connection to the RasPi and it will ask for login credentials.
The default login credentials are:
Username: Pi
Password: Raspberry

Congrats, you should now be in full control of the RasPi remotely.

Python

In the RasPi terminal type “sudo apt-get install python2” and hit enter.
After a short installation process you should be able to type “python” in the terminal and receive the following:

```
pi@raspberrypi ~ $ python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

This means that python has been installed.

Git

Git is a program used to pull files from repositories such as GitHub, we will use this program to download all of the necessary code to power the project.

In the RasPi terminal type “sudo apt-get install git” and hit enter.

Installing Dependencies

Now that we have Git, we can start installing the code actually used for our project.

Run the following commands.
Run each of these lines in the console:
pip install colour
pip3 install gpsd-py3
pip install smbus2
sudo apt-get install mysql-server python-mysqldb

This will install all of the libraries and dependencies so that our project code can run.

Setting Up and Running the Code

Clone the repo (specifically googlemaps.py, dboutput.py) from https://github.com/samragsdalewustl/-LongboardSensors

Pic a folder and enter the command:
git clone https://github.com/samragsdalewustl/LongboardSensors

Follow the instructions here:
http://raspberrywebserver.com/sql-databases/using-mysql-on-a-raspberry-pi.html
To set up the SQL server.
(note that dboutput.py is currently setup for the a DB named “test” and table “tempdat” within “test”)

Setup Apache by following this tutorial:
https://www.stewright.me/2015/08/tutorial-install-apache-php-and-mysql-on-a-raspberry-pi-2/

enter:
sudo python dboutput.py
This should start our code.

The RasPi will now be waiting for a button press to start recording. If you’re ssh’d or connected to the RasPi over HDMI, you will see the data that is being added to the DB on your screen.

Hit the record button and data will begin recording.
Whenever you are done recording data hit the button again.

Connect to your ICS enabled device over ethernet.

Navigate to 192.168.2.2 (the default Apache IP address) on your laptop or other device (When RasPi has been connected)

The GPS route, as well as acceleration and velocity should display in a nice GUI on the web browser.

GPS Module on Boot

To set up the Pi to run the code on boot do the following.

Create a file named launcher.sh in your project folder

This file should be a list of bash commands which navigate from root to the project folder, followed by “sudo python dboutput.py”.

Ex:
cd /
cd /home/pi/LongboardSensor
sudo python dboutput.py
cd /

Type chmod 755 launcher.sh into the console to compile the script

Type sh launcher.sh into the console

Create a folder called “logs” in the home directory

Type “sudo crontab -e”

Now the Pi will automatically start the GPS tracking on wakup.