Seeing Like A Bike

Aastha Jalan | Nikhil Soraba | Sharon Ang

Georgia Institute of Technology | Project Studio: Seeing Like a Bike | Proximity Team | Spring 2017

What are we interested in?

Rider Stress

How close objects which might contribute to rider stress are on the road.

This information when combined with data about the rider's speed, road surface information, air quality information and explicit rider feedback can help us understand the riding experience better and work to improve the biking infrastructure so as to enhance the quality of the rider's experience.

How we approached the problem...

Planning

Incremental Development

Integration Test

Data Collection and Analysis

Possible Sensors for Target Detection

Selecting the right sensor is not a strict process. This is about eliminating all the wrong choices based on asking a series of questions aiming to eliminate first the technology underlying the sensor and then the product that doesn't fit the bike requirements.

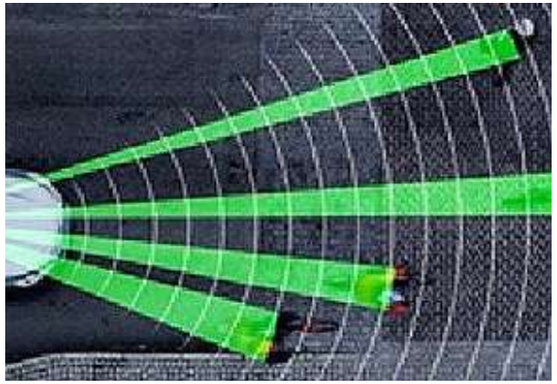
A good starting point is to begin questioning these about the sensors:

* Type of Sensor – the presence of an object can be detected with proximity sensors, and there are several kinds of sensor technologies including here image sensors, ultrasonic sensors, capacitive, photoelectric, inductive, or magnetic;
* Accuracy – it is useful to choose sensors with accuracy values between desired measurement margins;
* Resolution – a high resolution can detect smallest changes in the position of the target;
* Range – involves choosing the sensors based on measurement limits and compared with the desired detection range of the bike;
* Control Interface – to interface the sensor you have to know how it interfaces with the controllers e.g. I2C, PWM, Serial, Analog etc.;
* Cost – depending on the budget allocated to a project, you can select the sensor o that can be put on the bike.

Microphone

Maxbotix Ultrasound Sensor

Tera Ranger Time of Flight Sensor



Walabot Radar

RPLIDAR 360° Laser Scanner

Garmin LIDAR Lite V3

Our Requirements

* **Type of Sensor** – image sensors were avoided due to the high post-processing cost;

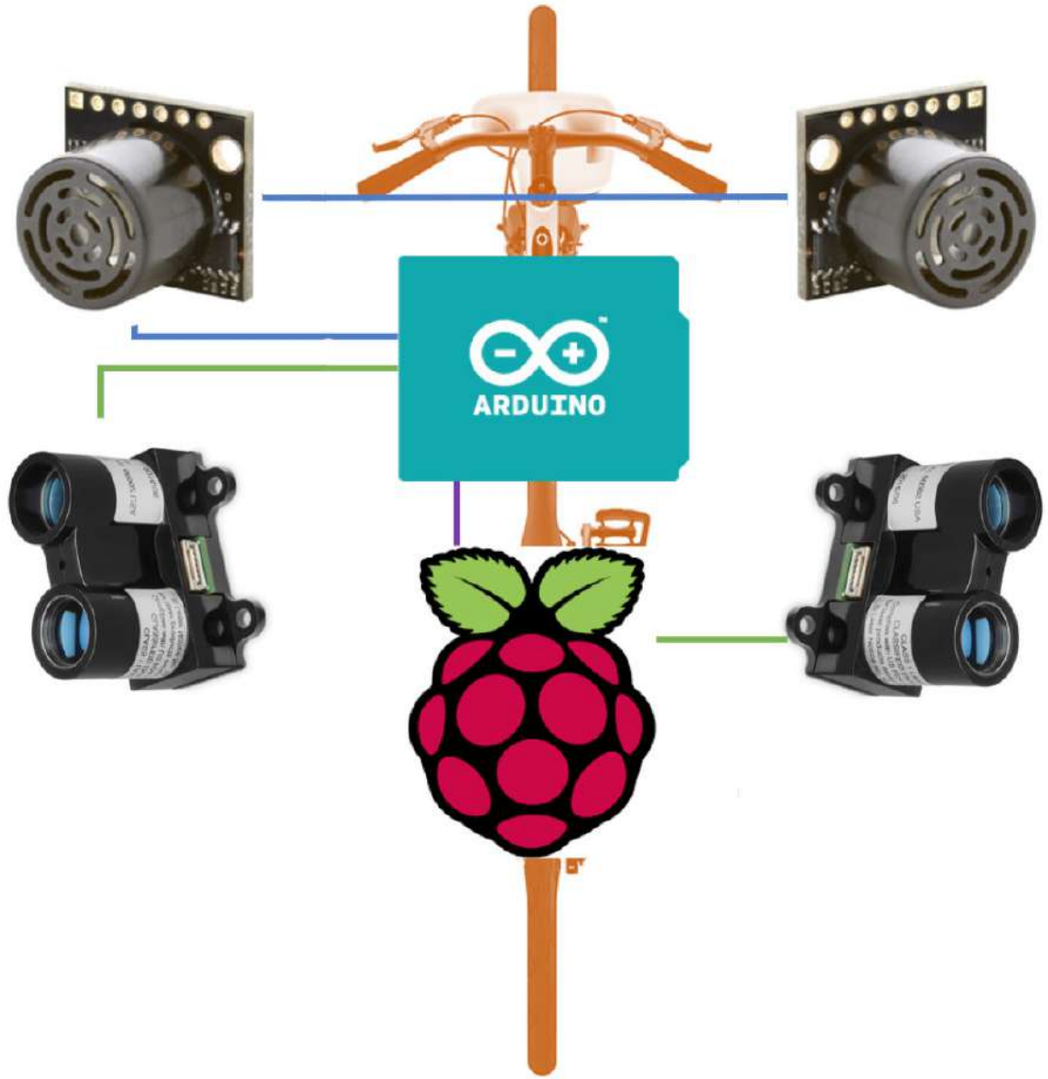
* **Accuracy** – since objects causing stress to the rider were relatively large (e.g. vehicles), accuracy requirements were not very stringent;

* **Resolution** – a high resolution is not required because the objects causing stress to the rider would have a net displacement above the centimeter-scale resolution;

* **Range** – the assessment was made that for objects to cause stress to a rider, they have to be within 3m at the sides and 40m from the back;

* **Control Interface** – the controllers used were primarily Arduino Uno and Raspberry Pi, therefore any sensor that could be controlled with these controllers would qualify;

* **Cost** – the total budget for this project is capped at $700, hence eliminating high cost items.



Sensors

Garmin LIDAR Lite V3

* Reliable and powerful ranging and proximity sensor
* Compact, lightweight with low power consumption
* Communication via I2C or PWM
* Requires a power source and an external micro-controller running an application
* 40 meter laser-based sensor
* 130 milliamps during an acquisition

Maxbotix MB1010 LV-MaxSonar-EZ1 Range Finder

* 42kHz Ultrasonic sensor measures distance to objects
* Read from all 3 sensor outputs: Analog Voltage, Serial Digital, Pulse Width
* Objects closer than 6 inches range as 6 inches
* Resolution of 1 inch
* Maximum Range of 254 inches (645 cm)
* Operates from 2.5-5.5V
* Low 2.0mA average current requirement
* 20Hz reading rate
* Small, light weight module

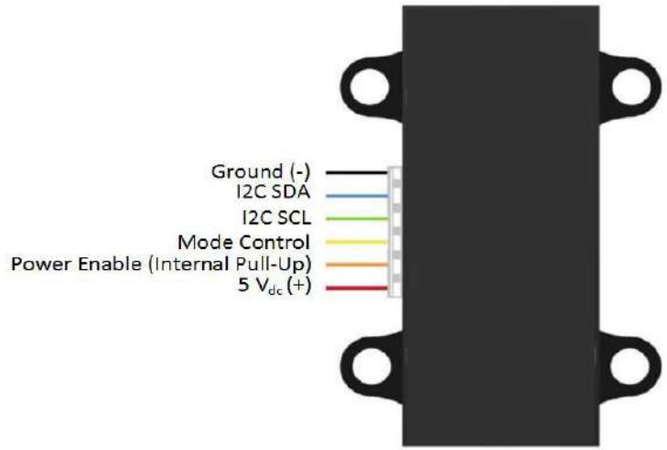
What Do You Need?

2 Ultrasonic Sensors
2 Lidar-Lite V3
1 Arduino Uno
1 Raspberry Pi v3
1 Power Bank
1 RGB LED
2 Single Pole On-Off Switch
Some jumper wires
1 A-B USB cable
1 Micro-B USB cable
1 LAN cable (RJ-45)
1 Monitor
1 HDMI cable
1 Mouse
1 Keyboard

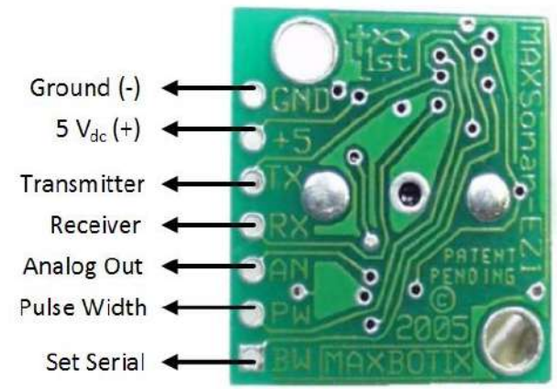
*Connection diagrams found in each step.
Reference codes can be found in the Appendix.

Understand Your Sensors

It is important to understand the sensors that we are working on. A simplified pin-out of the two sensors are shown above. For more information, please refer to the individual datasheets.



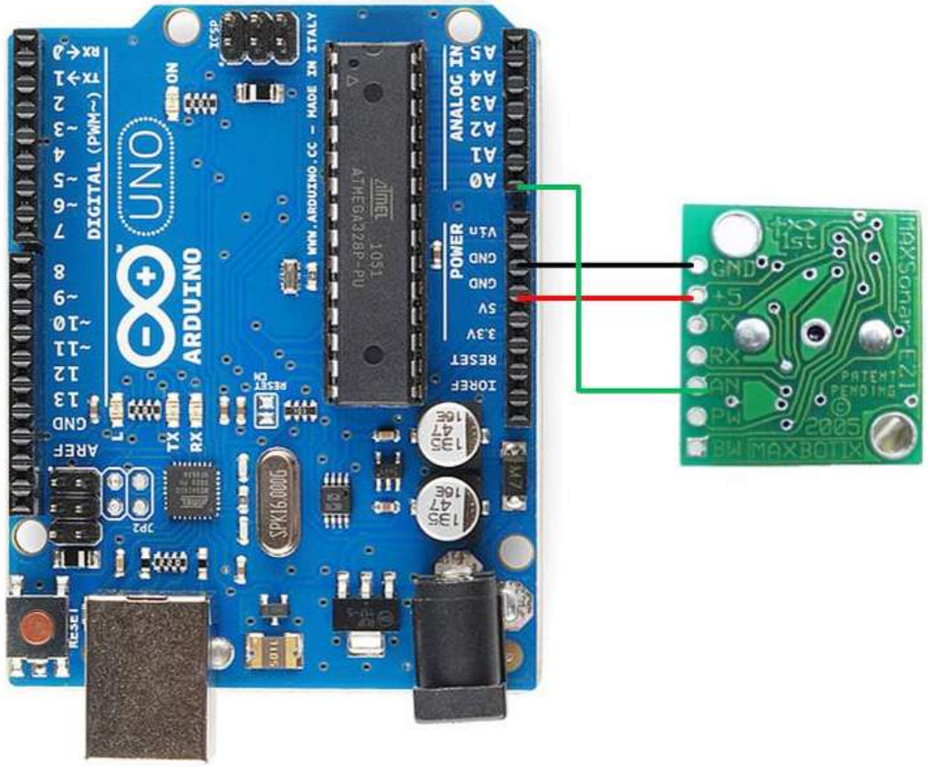
Pin-out for LIDAR Lite

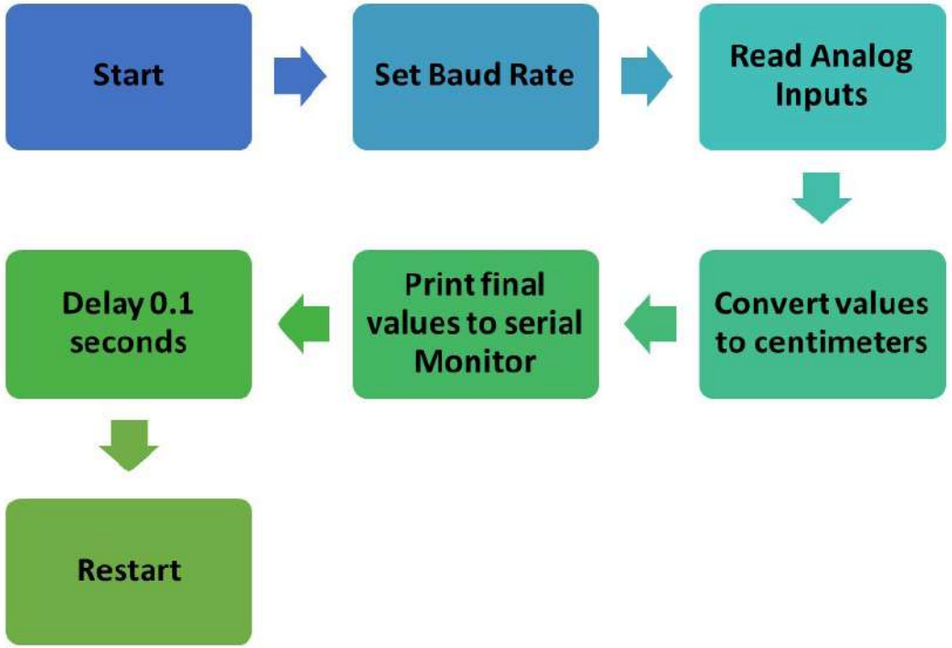


Pin-out for Ultrasound Sensor

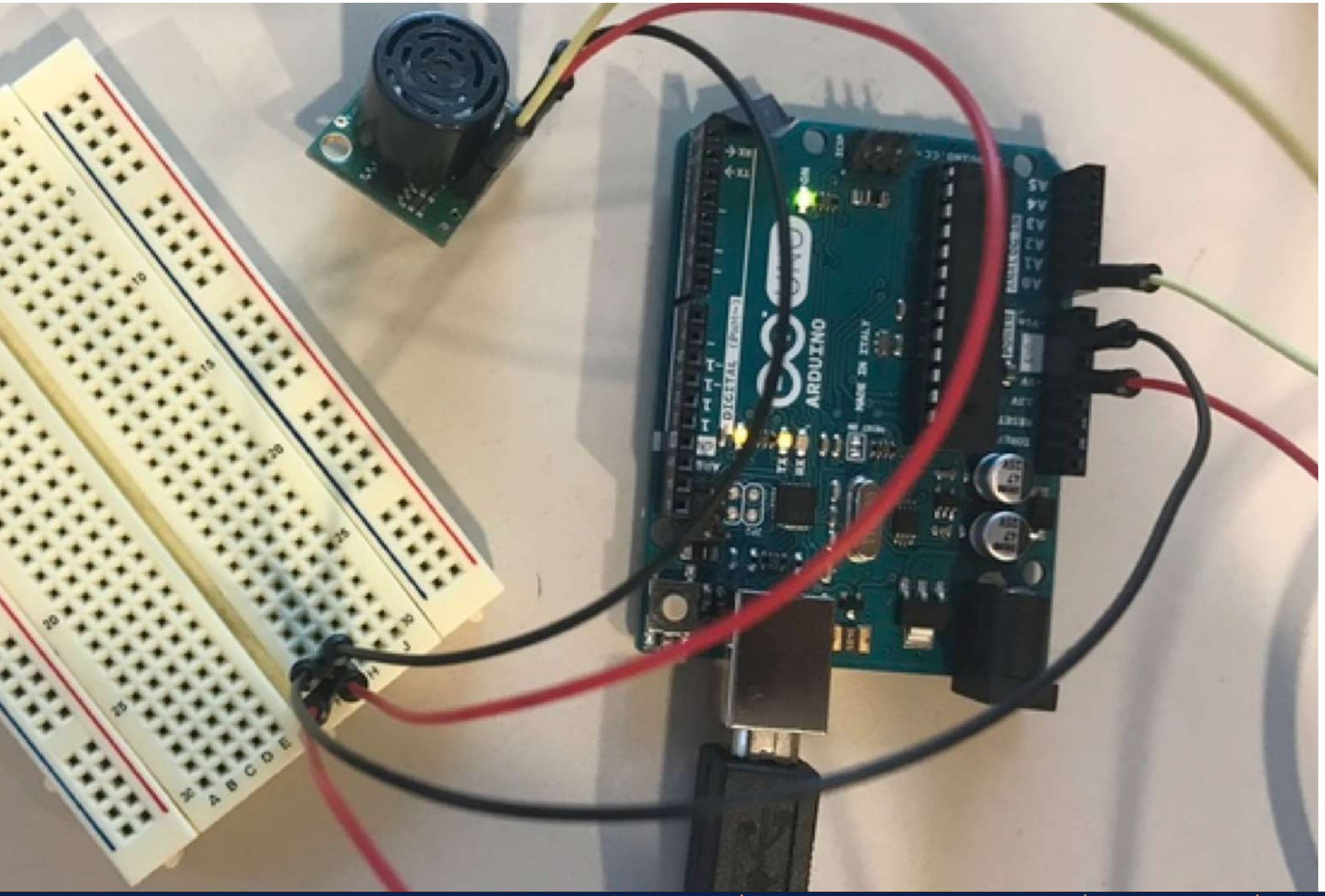
Step 1: Single Ultrasonic Sensor

Schematic and Logical Flow





Step 1: Single Ultrasonic Sensor

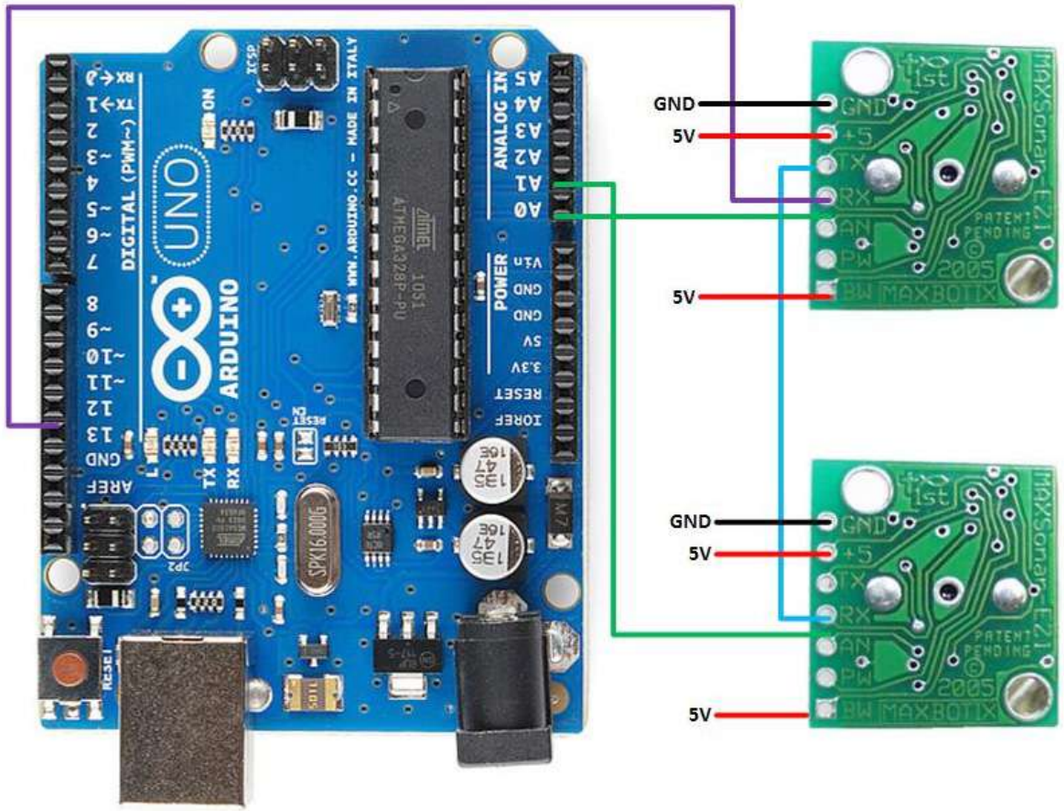


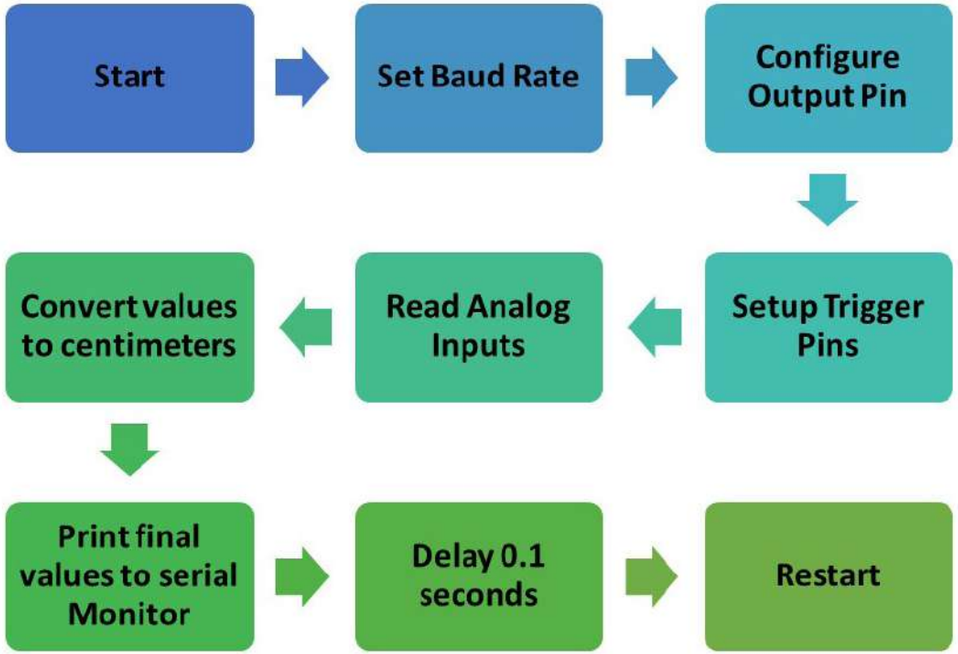
Note

Ensure the correct baudrate is selected when using the Arduino Serial Monitor

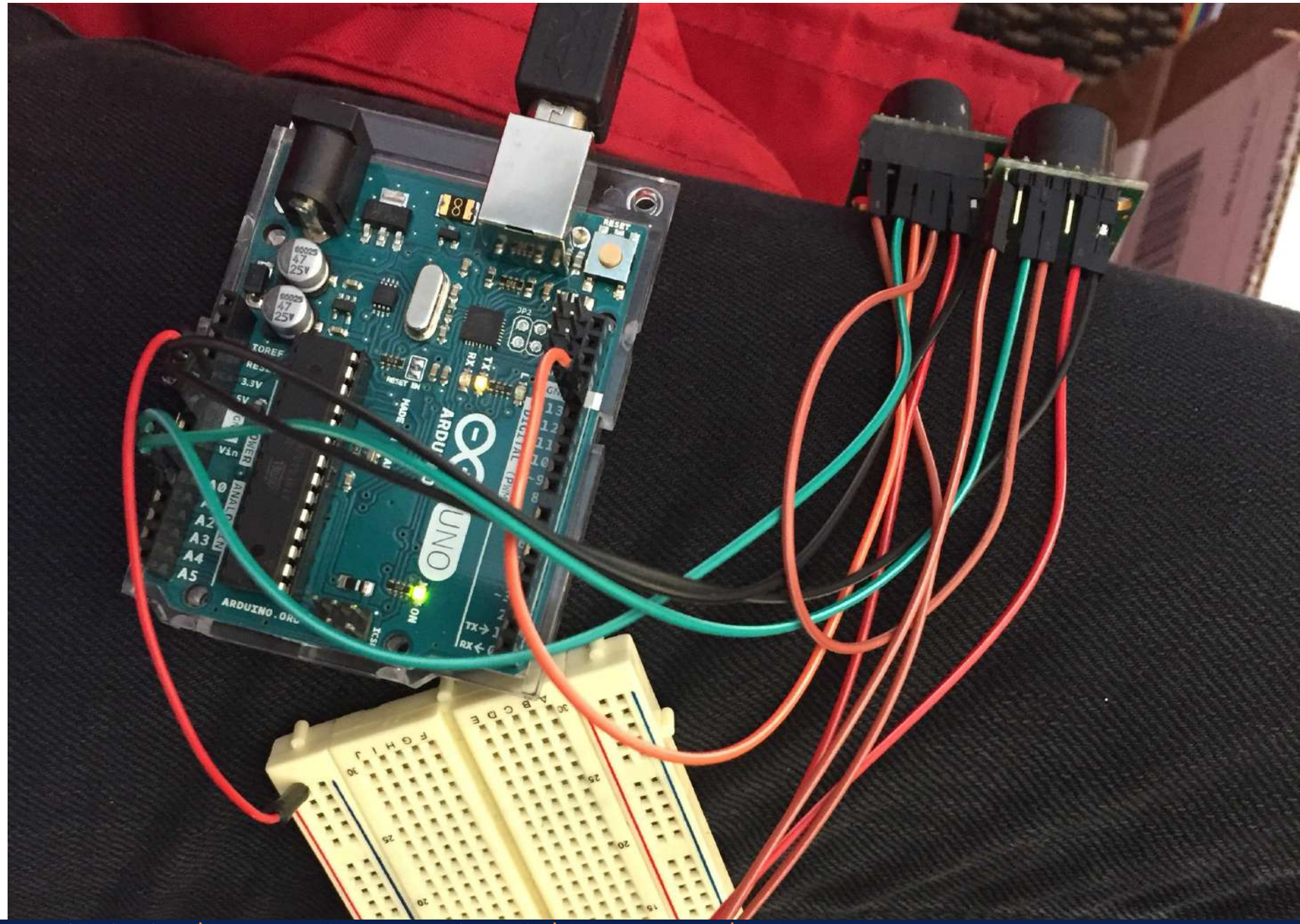
Step 2: Two Ultrasonic Sensors

Schematic and Logical Flow





Step 2: Two Ultrasonic Sensors

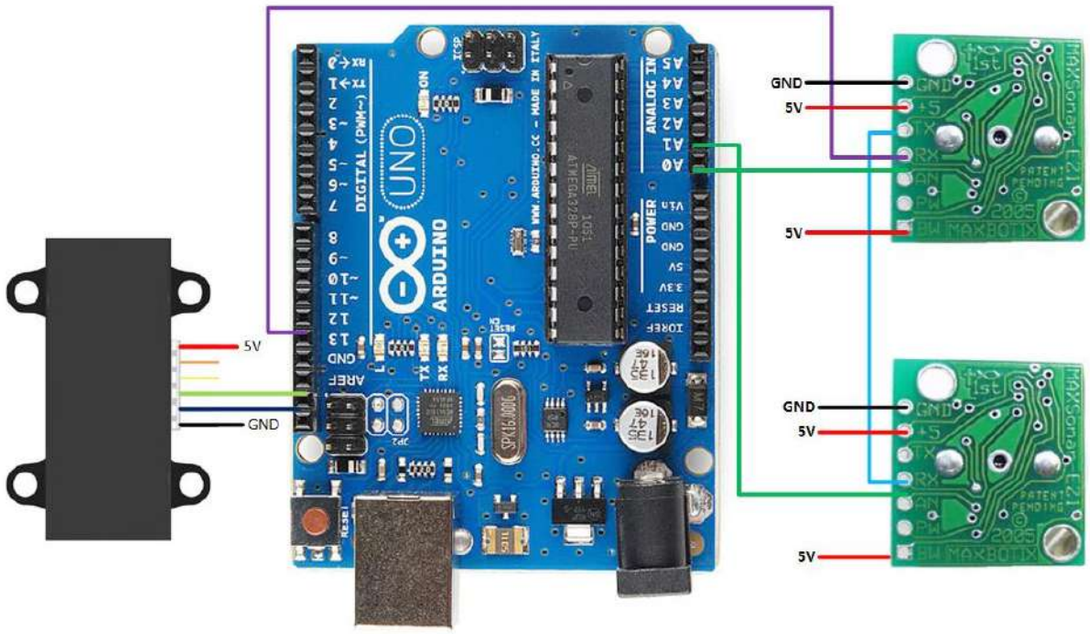


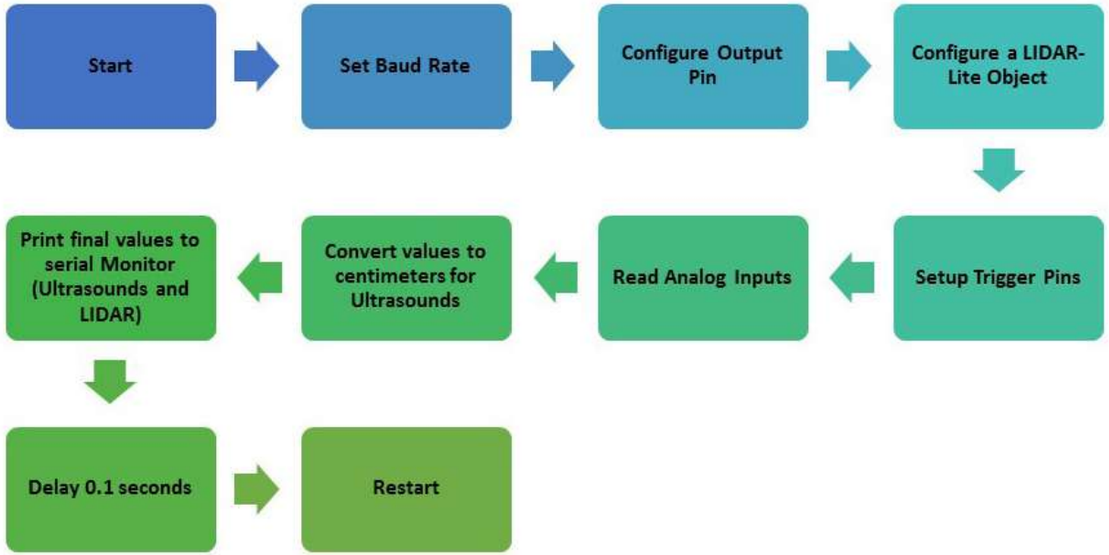
Note

To ensure that the sensors do not interfere, a special chaining method has to be used.

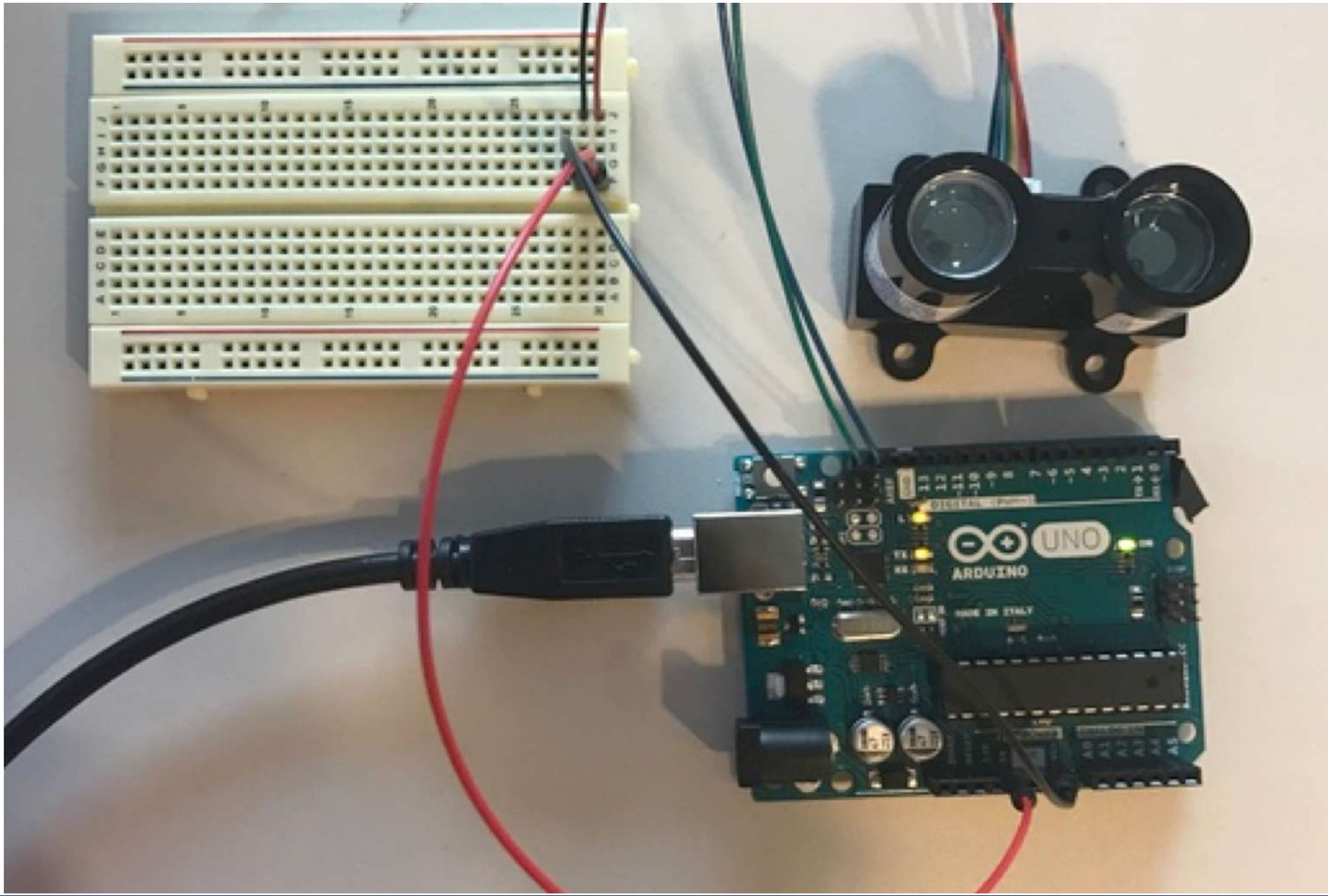
Step 3: Adding One LIDAR

Schematic and Logical Flow





Step 3: Adding One LIDAR



Note

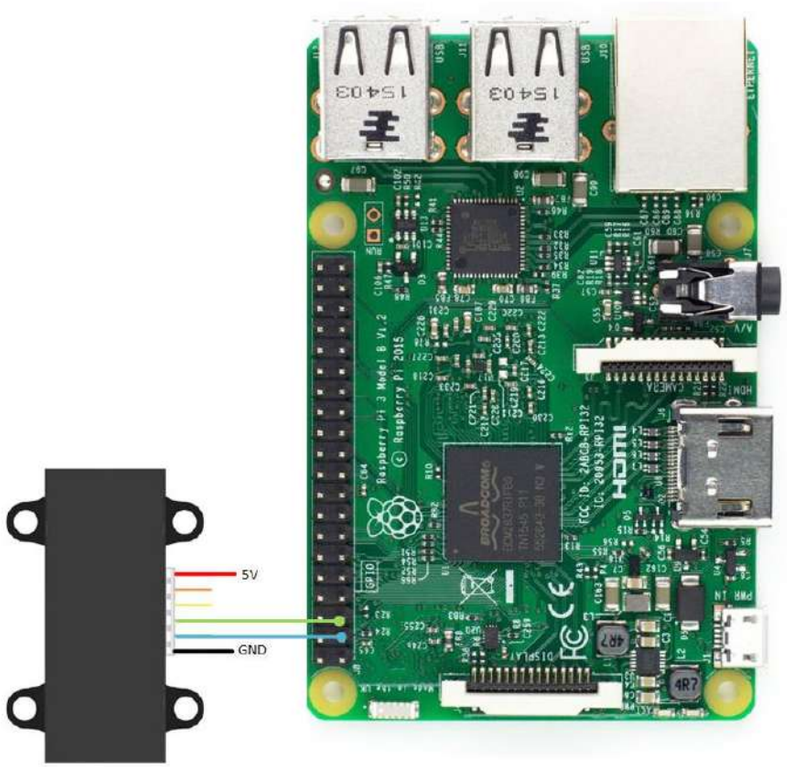
The Arduino sketch requires the libraries: <Wire.h> and <LIDARLite.h>

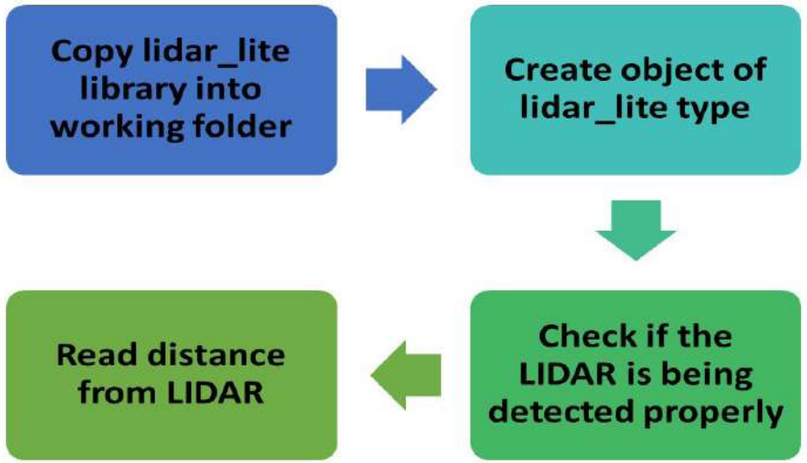
They can be downloaded from Github, and copied into the Libaries folder in the Arduino folder.

*When the LIDAR is out of range (i.e. beyond 40m), the output value is 1.

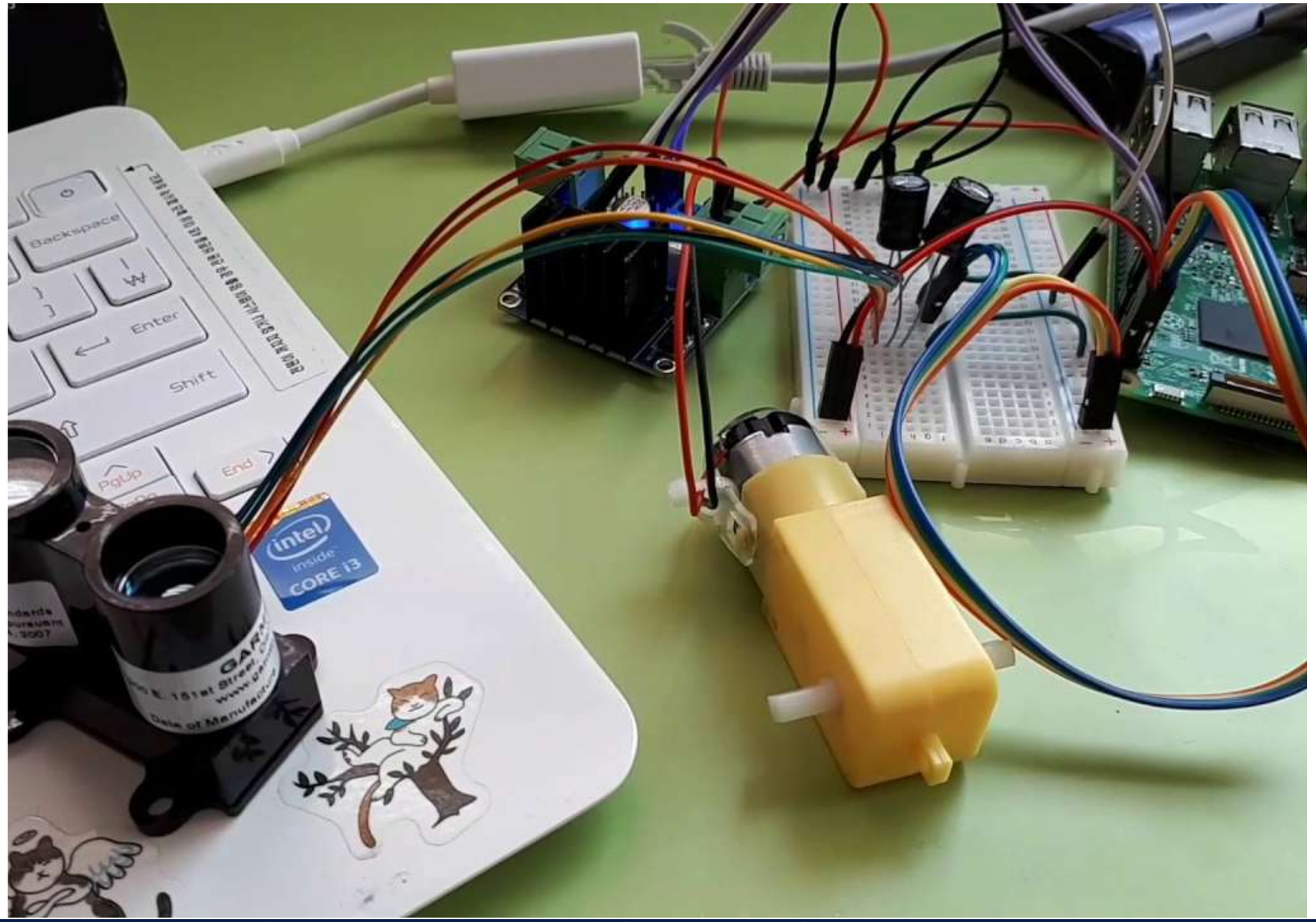
Step 4: Second LIDAR to Pi

Schematic and Logical Flow





Step 4: Second LIDAR to Pi

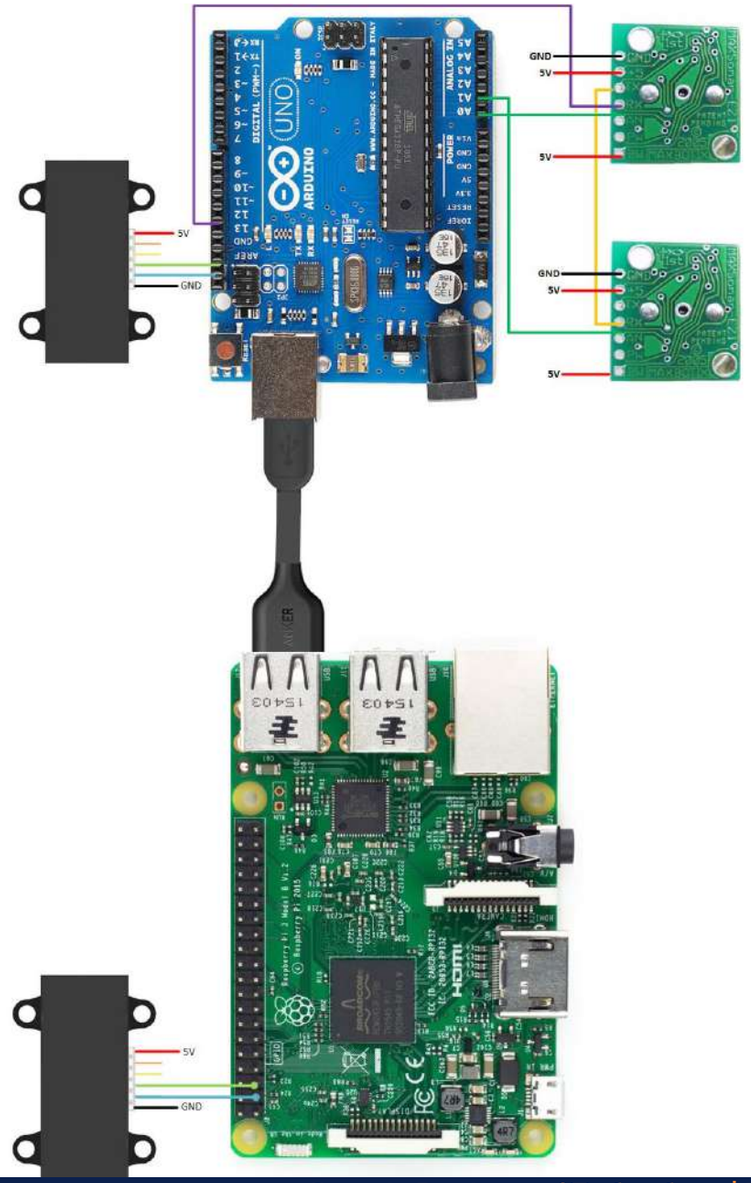


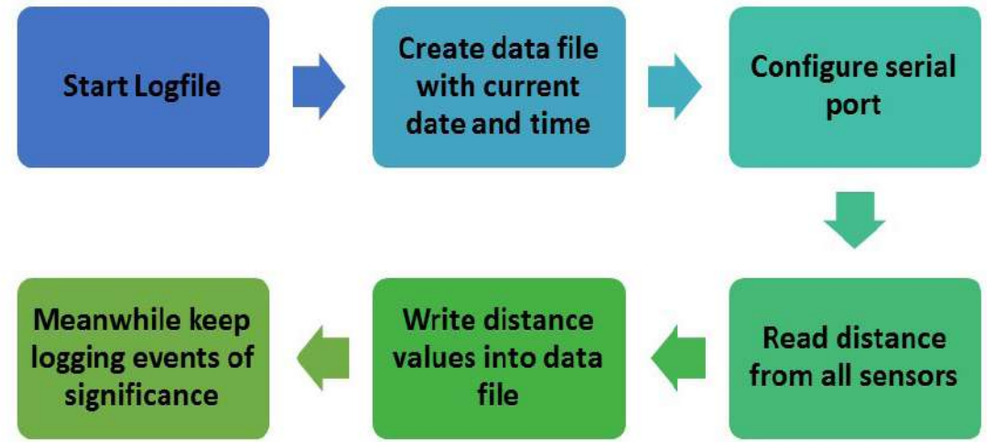
Note

The python code running on Raspberry Pi which acquires data from LIDAR requires Lidar_Lite module. It is available on the GitHub repository for the code. Ensure that you copy the files from the library into your working directory before you try running the code.

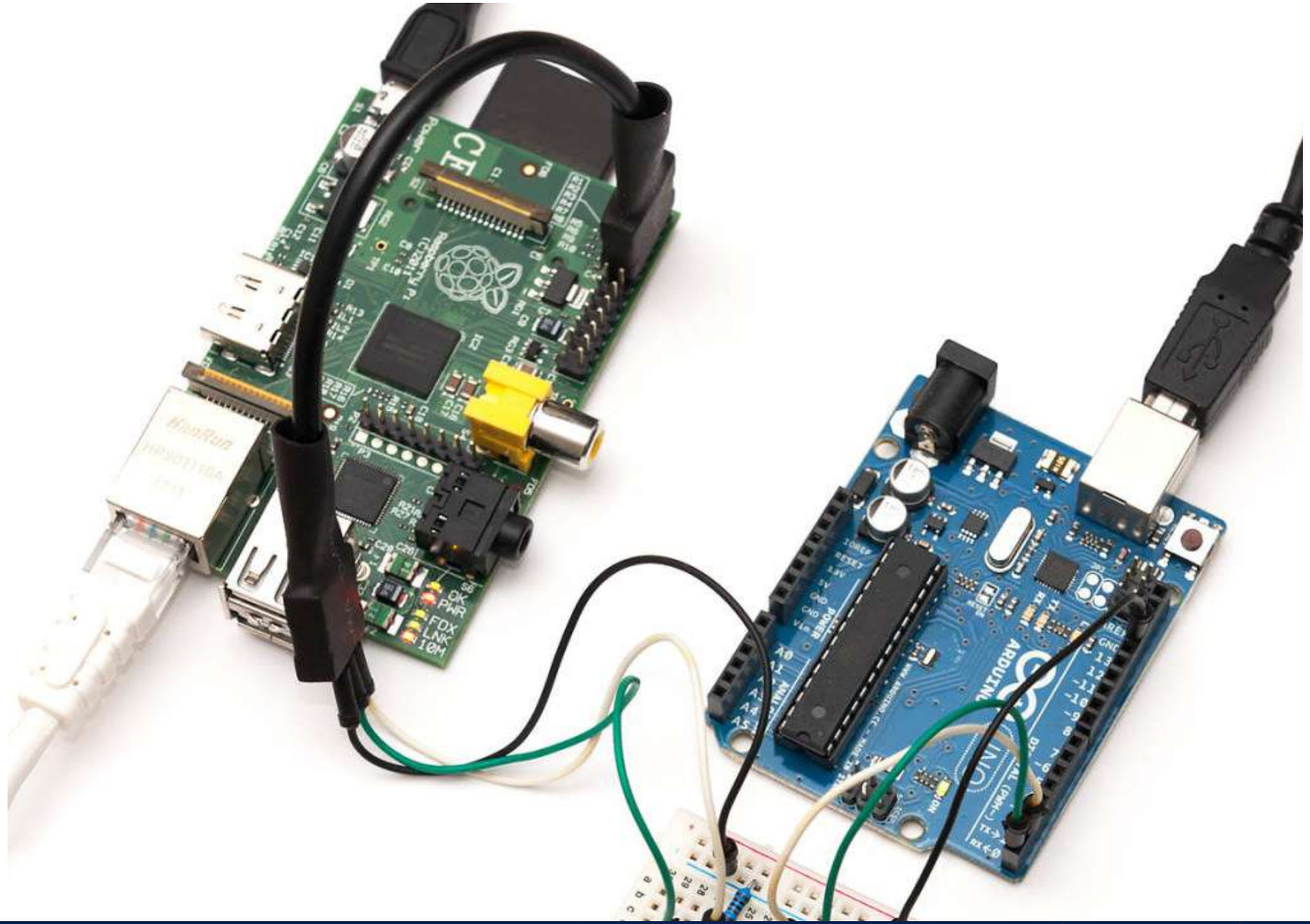
Step 5: Integrating Arduino and Raspberry Pi

Schematic and Logical Flow





Step 5: Integrating Arduino and Raspberry Pi



Note

Integrating Arduino with the Raspberry Pi:

(a) Connect the USB cable from the USB port of the Arduino to one of the USB ports of the Raspberry Pi.

(b) Download the Arduino sketch (serialTest.ino) from GitHub and program it in the Arduino.

(c) This will start serial communication between the two devices with a baud rate of 4800.

(d) Bootup the Raspberry Pi and navigate to /dev/

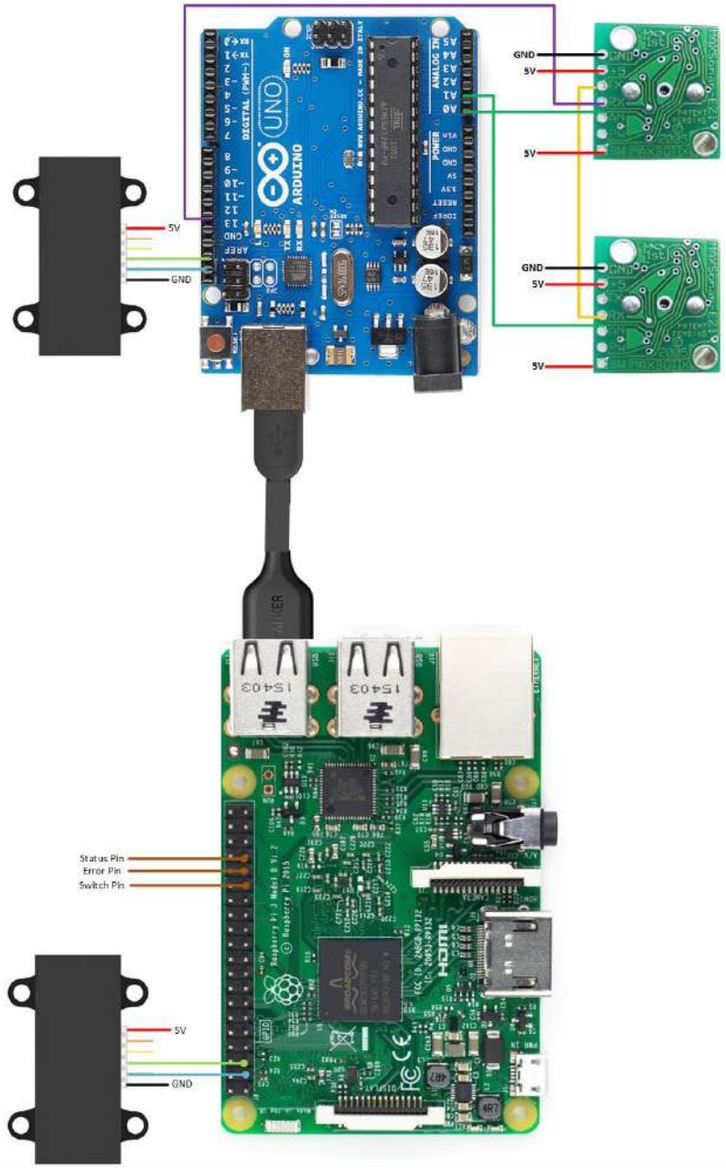
(e) Unplug and plug back the USB cable. Check the directory listing in this folder to identify the port name. (Usually /dev/ttyACM0)

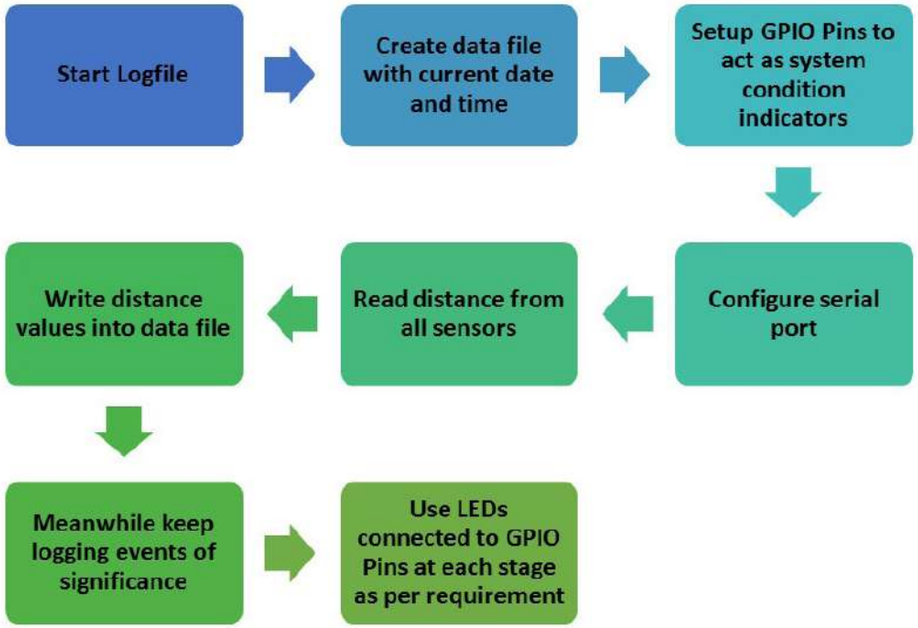
(f) Open a serial terminal (e.g. miniterm) with the port name and a baud rate of 4800.

(g) The sensor values from the Arduino will be printed on the console.

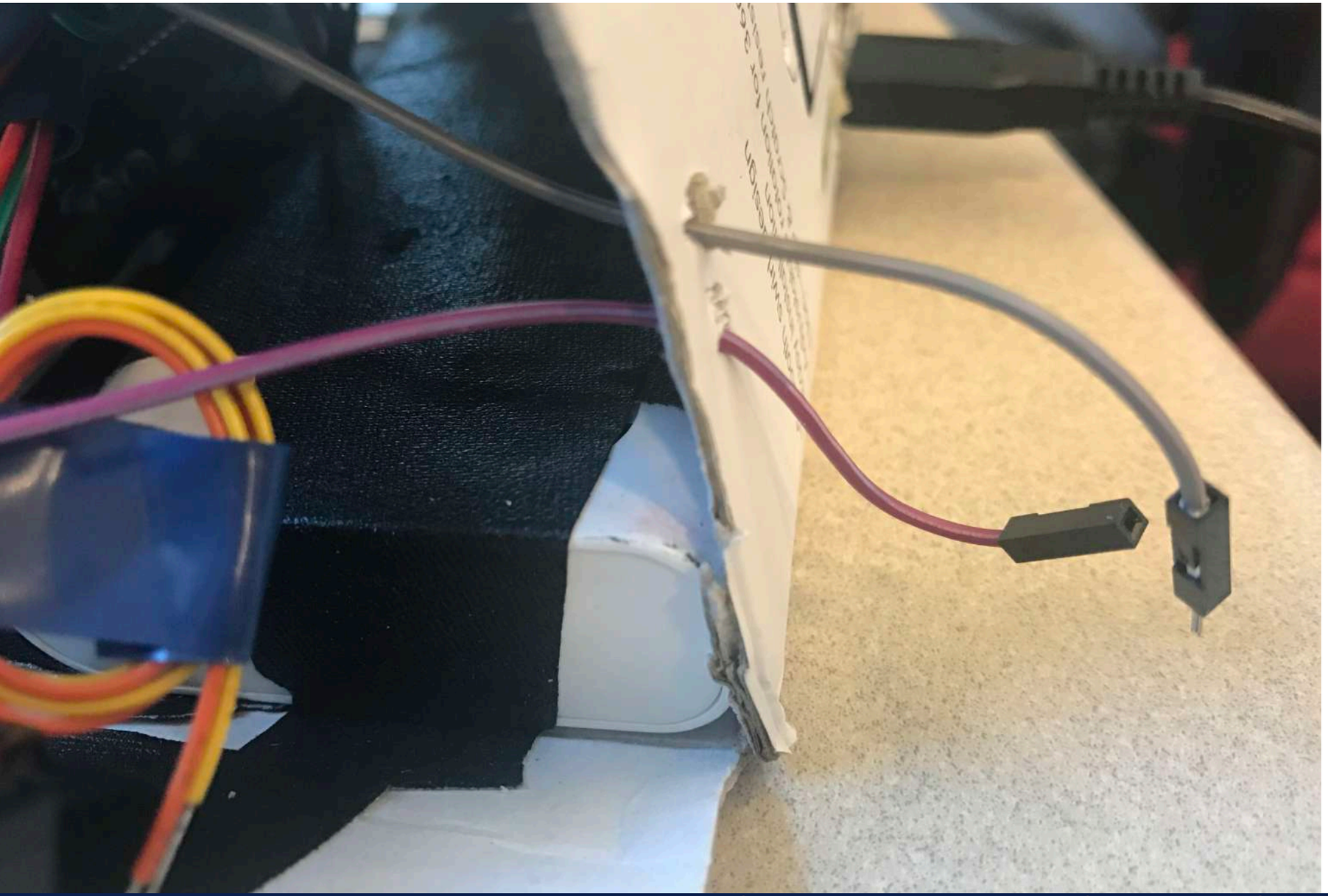
Step 6: Adding Switch and Indicators

Schematic and Logical Flow





Step 6: Adding Switch and Indicators



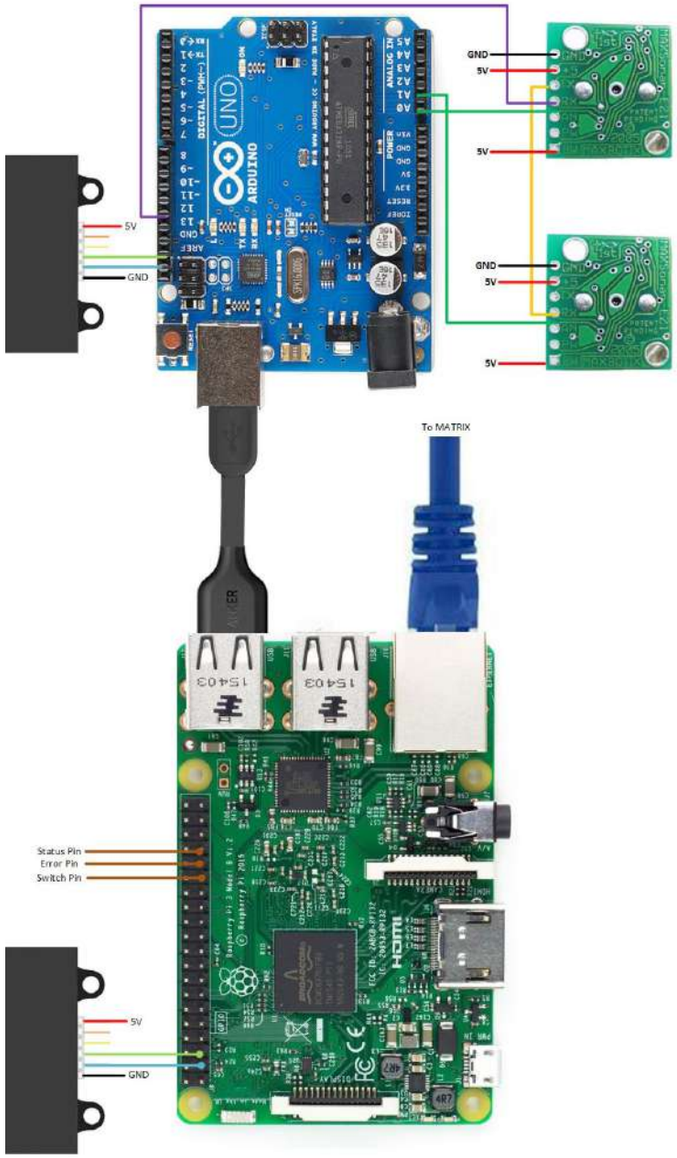
Note

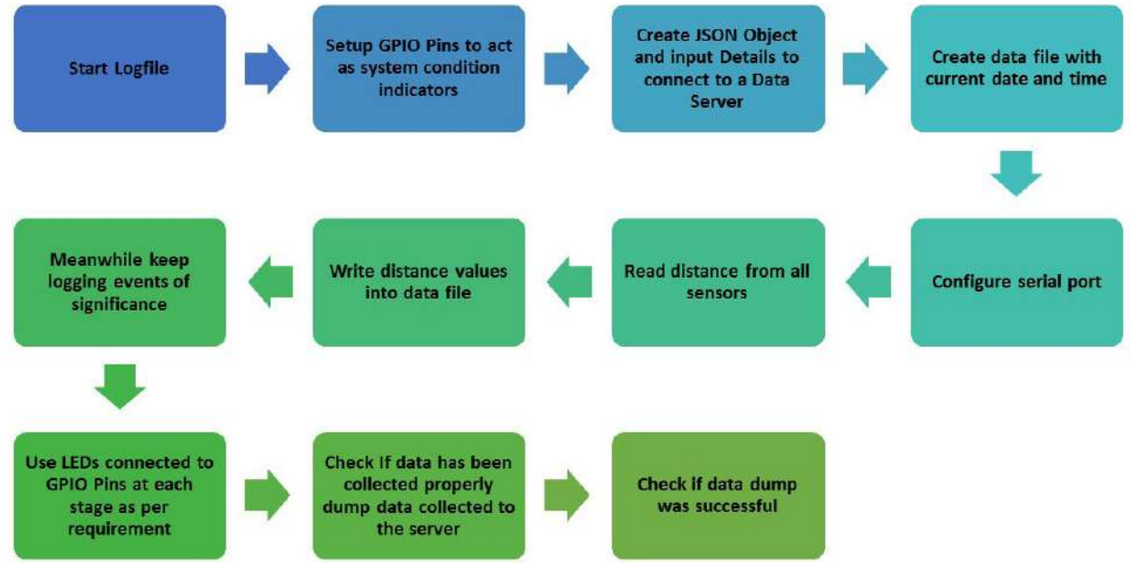
While this step is not absolutely necessary, it facilitates data collection in the field by having a status indicator LED and a switch to start data collection.

**In order for the program to run upon Power Up, the script should be invoked in the LXDE file of the Pi. To do this on the Raspberry Pi v3:
-In Terminal, navigate to ~/.config/lxsession/LXDE-pi
-Open the autostart file in that folder:
$ sudo nano autostart
-To run the script, put @python (your path)/(your script name).py

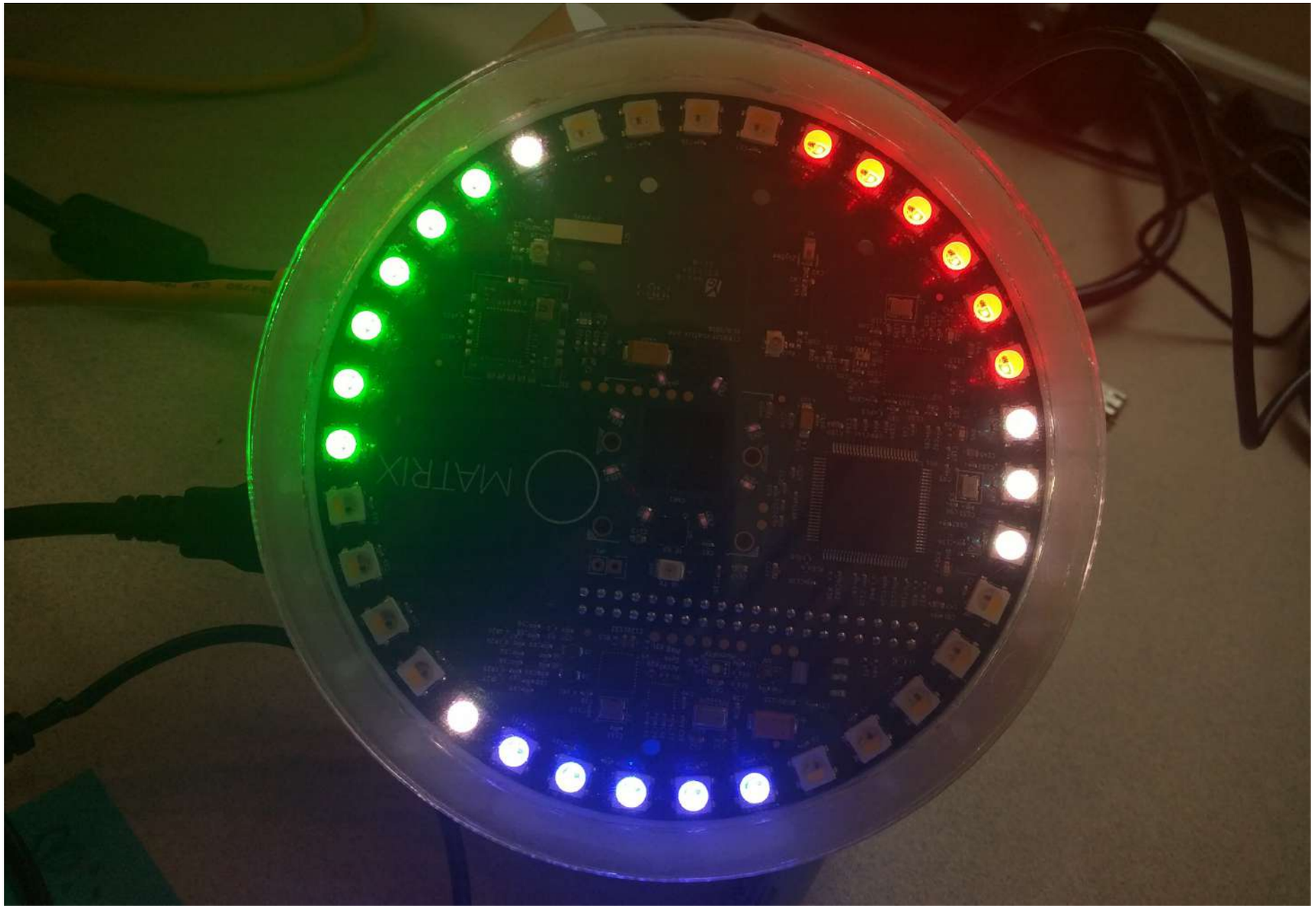
Step 7: Transferring Data

Schematic and Logical Flow





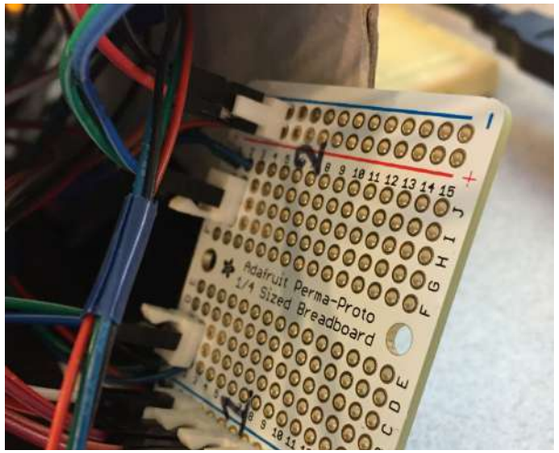
Step 7: Transferring Data



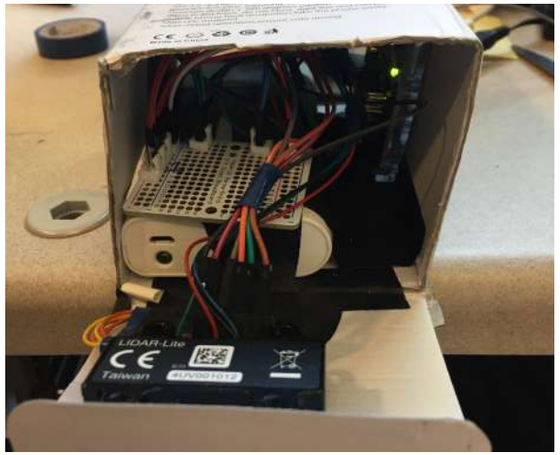
Note

This step connects the Proximity sub-system to a MATRIX Creator, that runs on another Raspberry Pi via a LAN cable.

Getting it on a bike



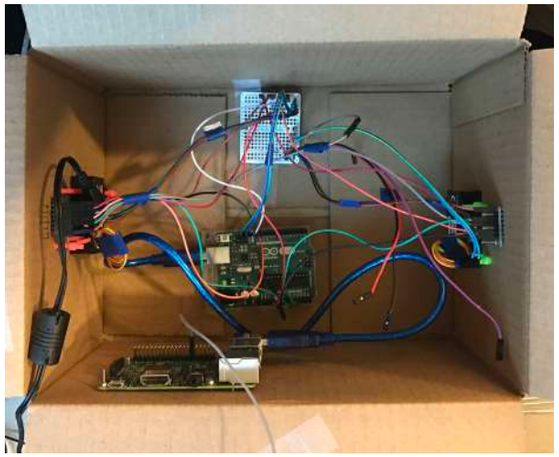
Ensuring wires are secure and do not fall out mid-way.



Packaging both sensors into a (small) box.

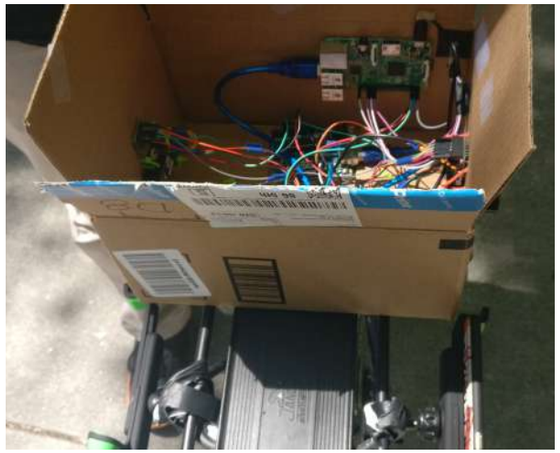
Sealing the box and placing it on a mount.

Ensuring secure and stable mounting.



Transferring the set-up into a bigger box for ease of access.

Side view of box.



Placing the box on the bike carrier rack.

The sensors mounted on a bike, ready to go!

Going for a ride

Note

Safety first!
Watch out for Parked cars and Traffic.

Ensure that the box is fastened securely, and that the LED indicators are working as expected.

Operating Instructions

To operate the Proximity Sub-System

1. Ensure that all 4 sensors are connected correctly as shown in Steps 1 to 7.

2. Connect the Arduino to the Raspberry Pi with a USB cable.

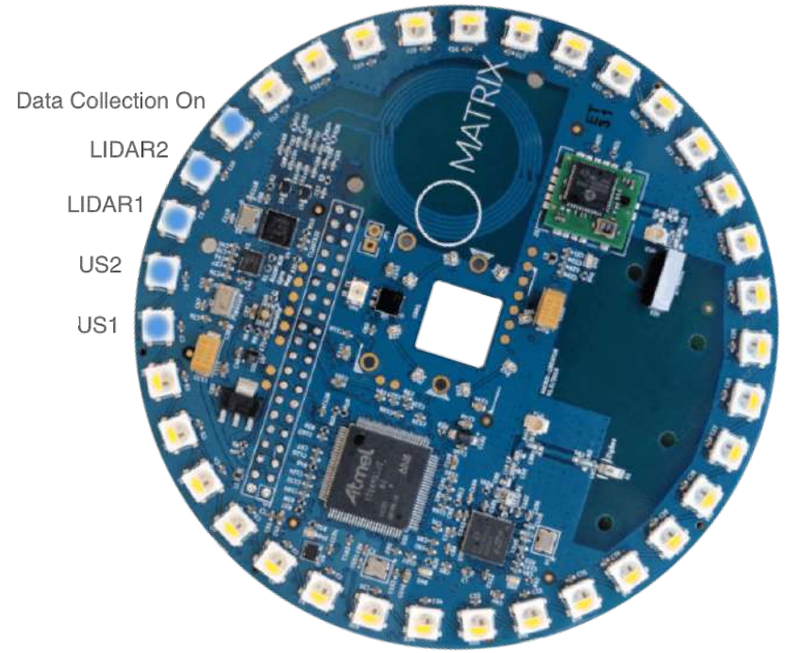
3. Connect the Raspberry Pi via the USB cable with the Switch to the Power Bank, ensuring that the Switch is in the OFF position.

4. Connect the Raspberry Pi to the Raspberry Pi with the MATRIX Creator, using a LAN cable.

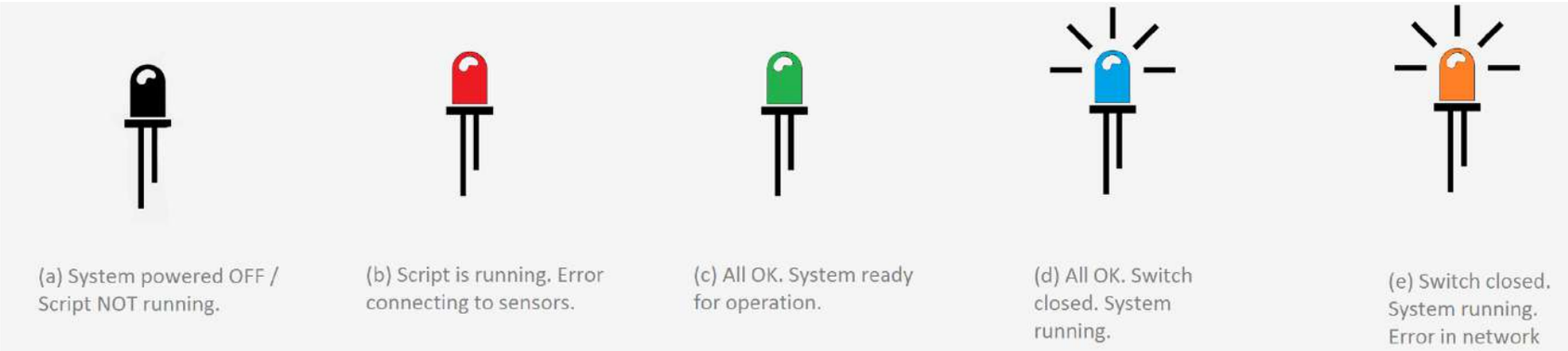
5. When everything is connected properly, turn the Switch to the ON position to power on the system. The LED on the Proximity box will be Green.

6. When ready to collect data, flip the switch for data collection to ON position. The LED on the Proximity box will flash Blue. All 5 LEDs on the MATRIX as shown in the figure, should be ON. Otherwise, check connections.

Proximity Sub-System Indicators on the MATRIX



LED indicator on the Proximity Box



The Types of Data

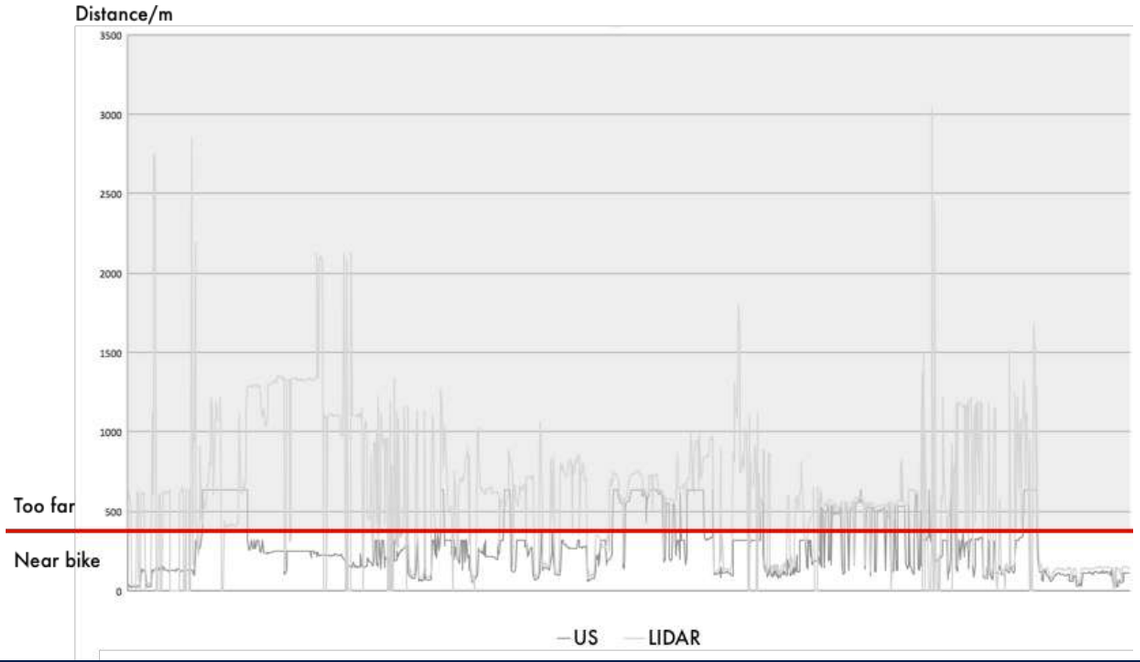
Data that is collected is appended into a new csv file each time the program begins. The data format is mentioned below. The csv file can be opened in Microsoft Excel to plot a graph easily.

data_04-10-2017-17-32-04-2.csv

```
04-10-2017-17-57-31,6,0.00,576,730
04-10-2017-17-57-32,124.46,0.00,580,739
04-10-2017-17-57-39,121.92,0.00,608,773
04-10-2017-17-57-40,35.56,0.00,1143,706
04-10-2017-17-57-40,124.46,0.00,605,746
04-10-2017-17-57-42,121.92,0.00,611,766
04-10-2017-17-57-42,121.92,0.00,613,723
04-10-2017-17-57-42,124.46,0.00,612,739
04-10-2017-17-57-42,124.46,0.00,613,764
04-10-2017-17-57-43,121.92,0.00,614,756
04-10-2017-17-57-43,124.46,0.00,603,45
04-10-2017-17-57-43,121.92,0.00,572,32
04-10-2017-17-57-43,121.92,0.00,571,35
04-10-2017-17-57-43,121.92,0.00,597,30
04-10-2017-17-57-44,33.02,0.00,613,20
04-10-2017-17-57-44,27.94,0.00,592,34
04-10-2017-17-57-44,25.40,0.00,577,23
04-10-2017-17-57-44,25.40,0.00,42,1
04-10-2017-17-57-45,20.32,0.00,38,712
04-10-2017-17-57-45,25.40,0.00,34,720
04-10-2017-17-57-45,27.94,0.00,28,744
04-10-2017-17-57-45,22.86,0.00,32,769
04-10-2017-17-57-45,20.32,0.00,29,754
04-10-2017-17-57-46,20.32,0.00,30,688
04-10-2017-17-57-46,25.40,0.00,614,677
04-10-2017-17-57-46,35.56,0.00,622,687
04-10-2017-17-57-46,35.56,0.00,69,689
04-10-2017-17-57-46,27.94,0.00,48,690
```

Data format:
Timestamp, Ultrasound1, Ultrasound2, LIDAR1, LIDAR2

The data for sensors mounted on the same side (i.e. Ultrasound1 and LIDAR1) can be plotted on a graph. Because we are only interested in near objects, the ones that are too far are ignored.



Processing Data

Because the nature of the two sensors are different (Ultrasound sensor has a wider beam width, whereas LIDAR often measures a point), both sensors are used to determine whether an object of interest exists. In the figure below, circled in red are possible parked cars, while circled in green are possibly infrasturcture (e.g. posts) or human on the walkway.

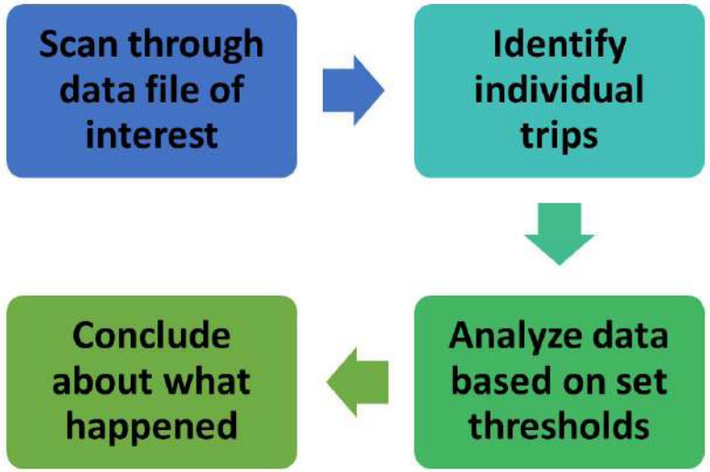
Understanding the Ride

```
 Trip 1 Summary ************
Number of parked cars: 0
Number of parked trucks:0
Approx. number of pedestrians / other short obstacles: 0

 Trip 2 Summary ************
Number of parked cars: 6
Number of parked trucks:1
Approx. number of pedestrians / other short obstacles: 50

 Trip 3 Summary ************
Number of parked cars: 14
Number of parked trucks:3
Approx. number of pedestrians / other short obstacles: 86

 Trip 4 Summary ************
Number of parked cars: 12
Number of parked trucks:2
Approx. number of pedestrians / other short obstacles: 97
[Finished in 0.6s]
```



Note

Python script for post-processing is available in Appendix H.

Seeing Like A Bike

From the output, a report of the what the bike sees can potentially be plotted on a map with location parameters, as shown.

The team would like to thank Professor Christopher A. Le Dantec for his leadership and guidance, and to the rest of the Bike Studio for their constructive feedback and support during the course of the Project.