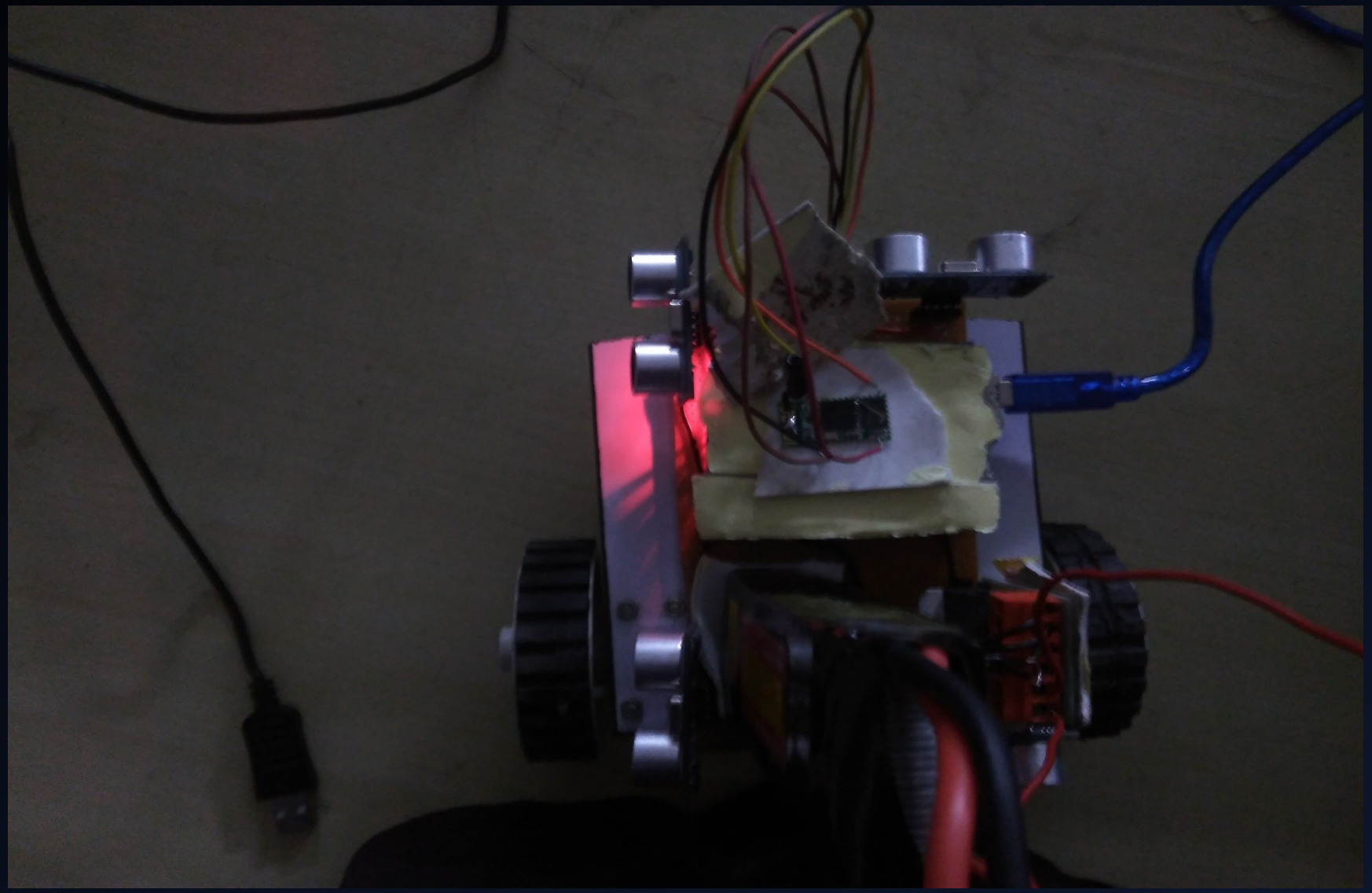
Indoor Mapping and Localisation

CS637 PROJECT

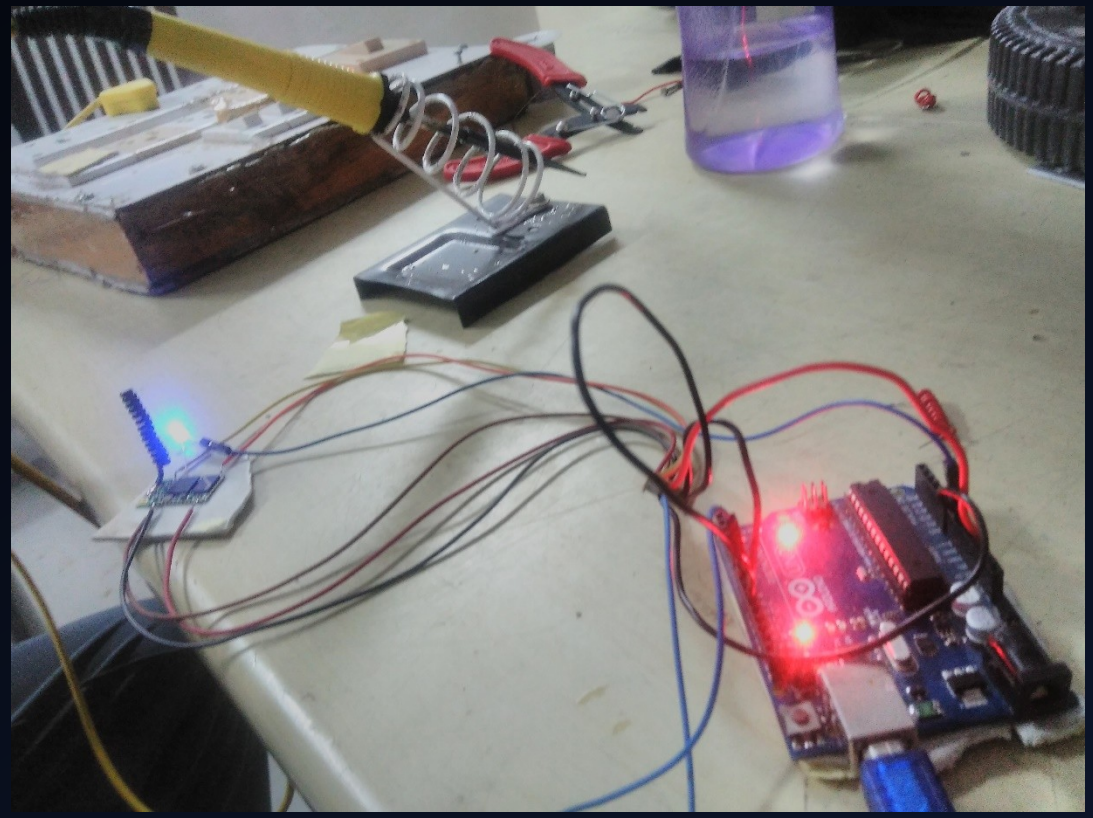
Shubh Gupta (14670), Swati Gupta (14742), Deepak Gangwar (14208)

The Robot



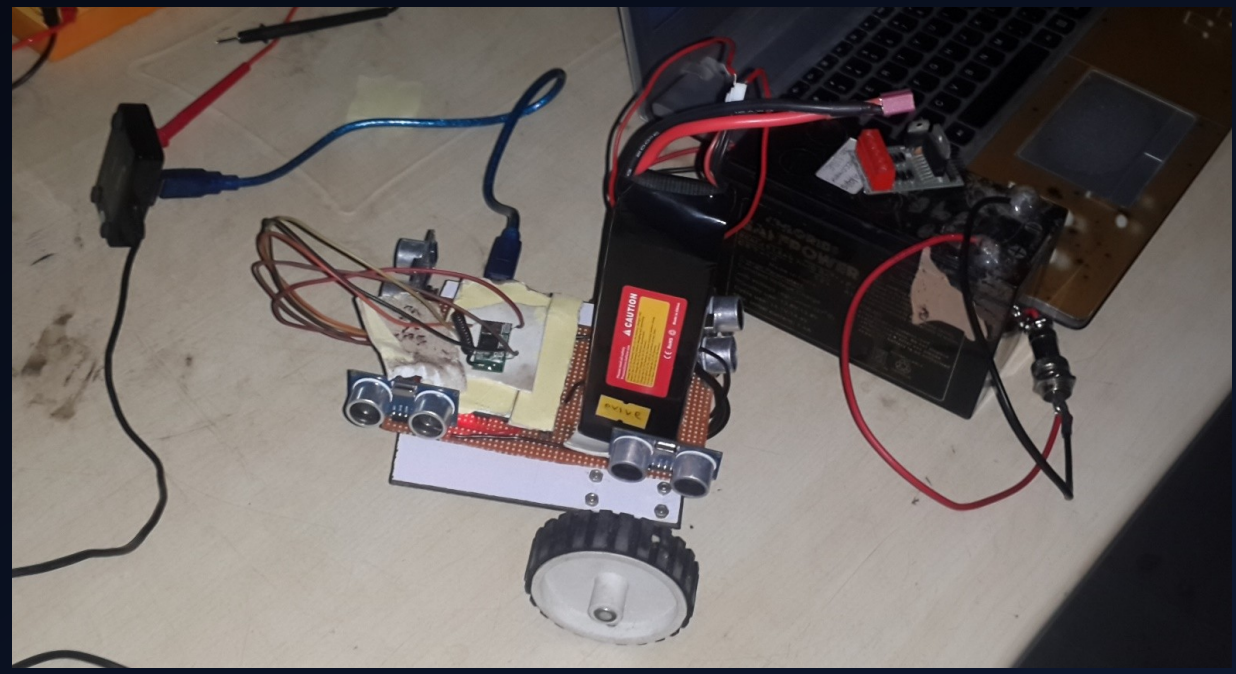
Problem Statement

To localize a robot in an indoor environment by mapping the setting and positioning via Bluetooth.



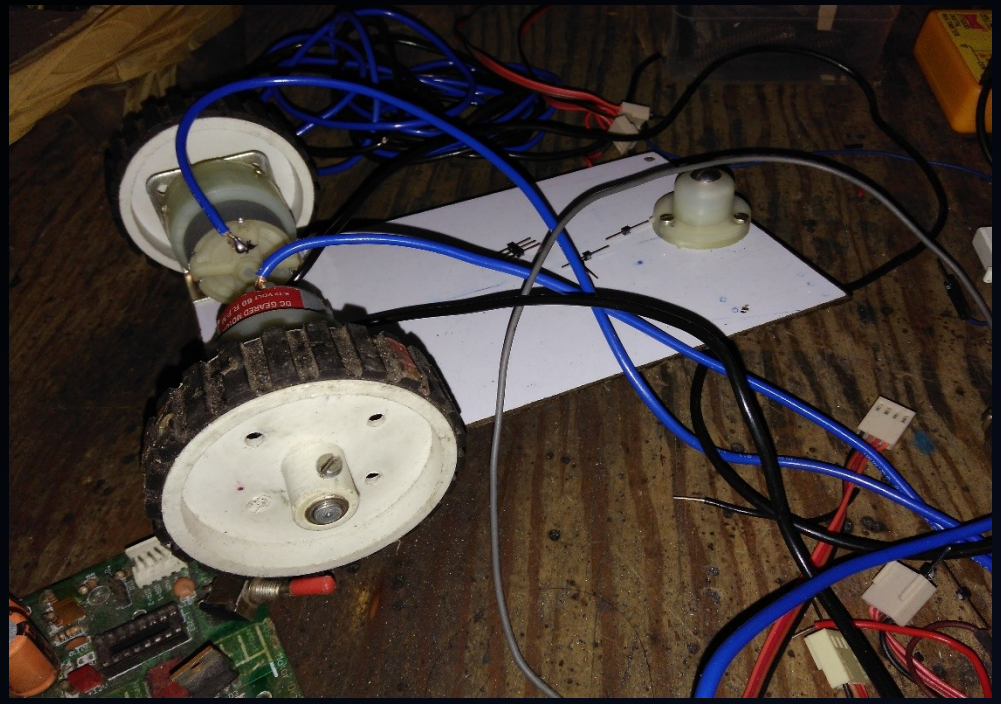
Features

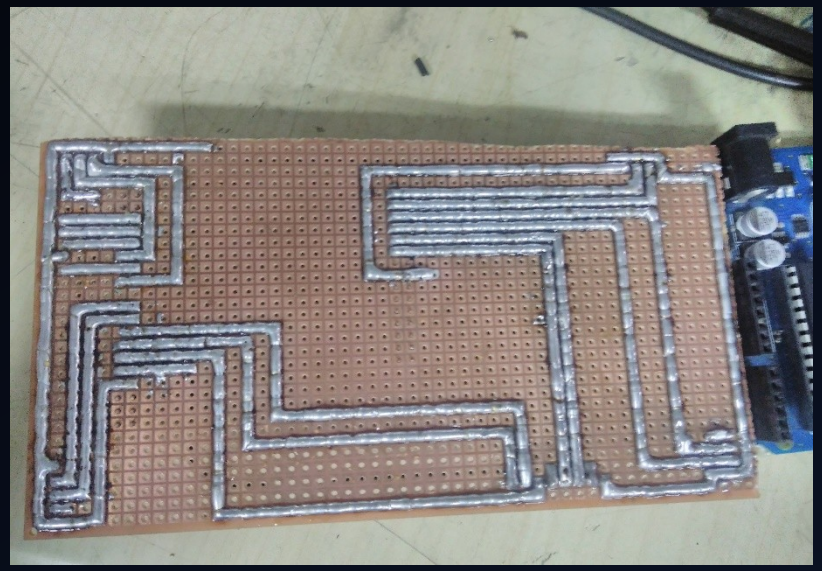
- Mapping of a room using ultrasonic sensors via wall following.
- Positioning of the bot to any selected coordinate in the room.



Components

- Electronics
  - Arduino UNO
  - GPB Board
  - Motor driver
  - Ultrasonic Sensors X4
  - Bluetooth modules X5
  - Battery 12V
  - Soldering tools
  - Breadboard wires
  - FRC Cable X5
  - Button cells
- Testing Arena
  - Wooden box 1m X 1m
- Chassis
  - Mica Sheet
  - Wheels X2
  - Motor X2
  - Castor wheel
- Softwares
  - Arduino IDE
  - Processing IDE

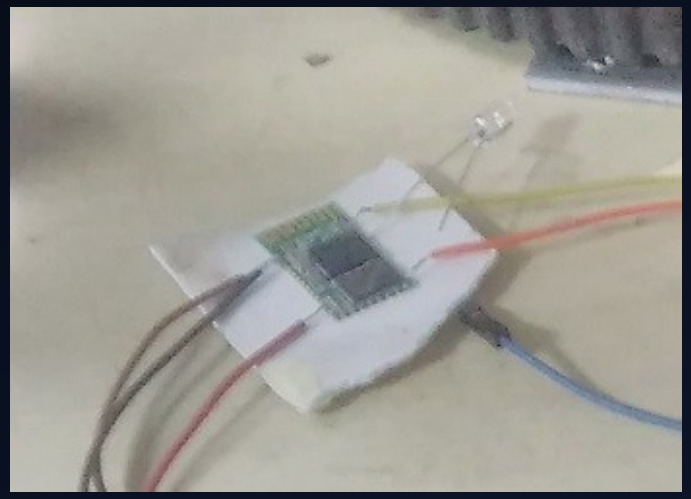





100% GENUINE QUALITY
100% GENUINE QUALITY

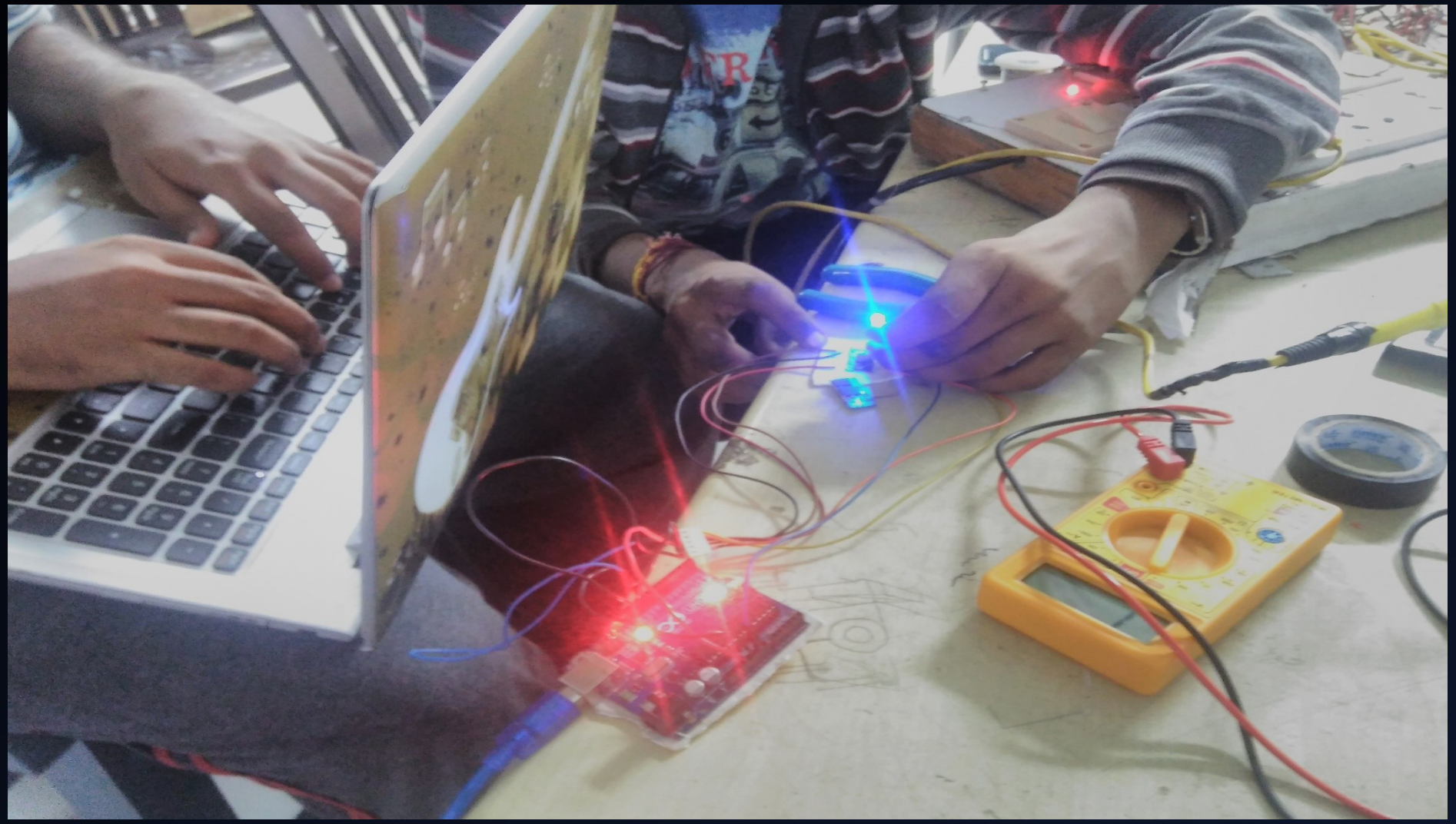
Testing Phase – Bluetooth

- Since the HC-05 modules came without a shield, they had to be operated upon for usage.
- After several failed attempts at making connections, we ended up with using FRC cables to solder to required slots in the modules.
- FRC cables are lightweight and more flexible as compared to breadboard wires, berg sticks or copper wire.



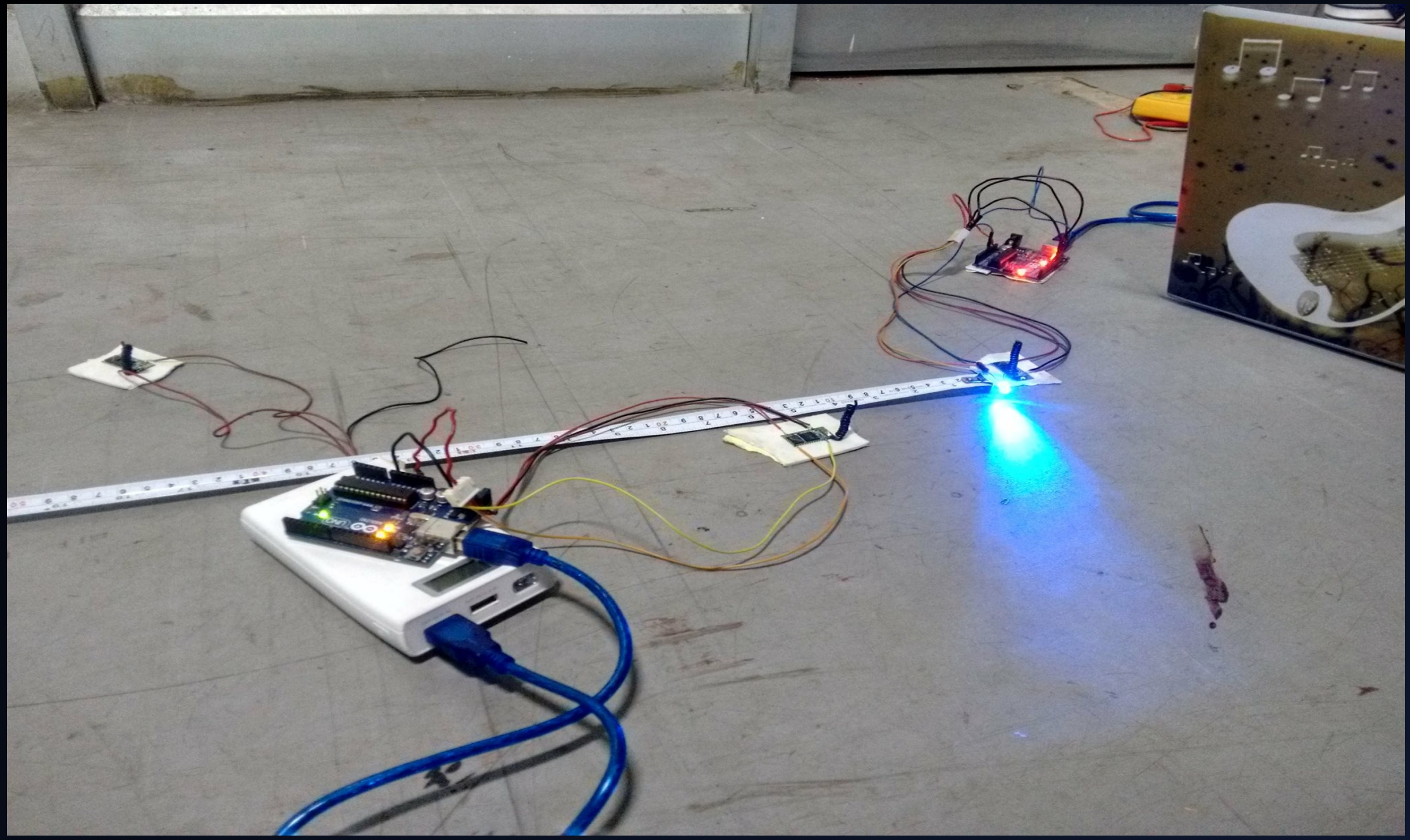
Testing Phase – Bluetooth Communication

- Bluetooth modules can operate in two modes – Master and Slave.
- Slave modules are capable of replying to requests or inquiries but incapable of making them, unlike Master modules.
- HC-05 modules come by default in Slave mode. To switch them to master mode a certain protocol has to be followed.
- After rigorous research on the matter, we were successfully able to put a module in Master mode and make inquiries from it to the Slave module.

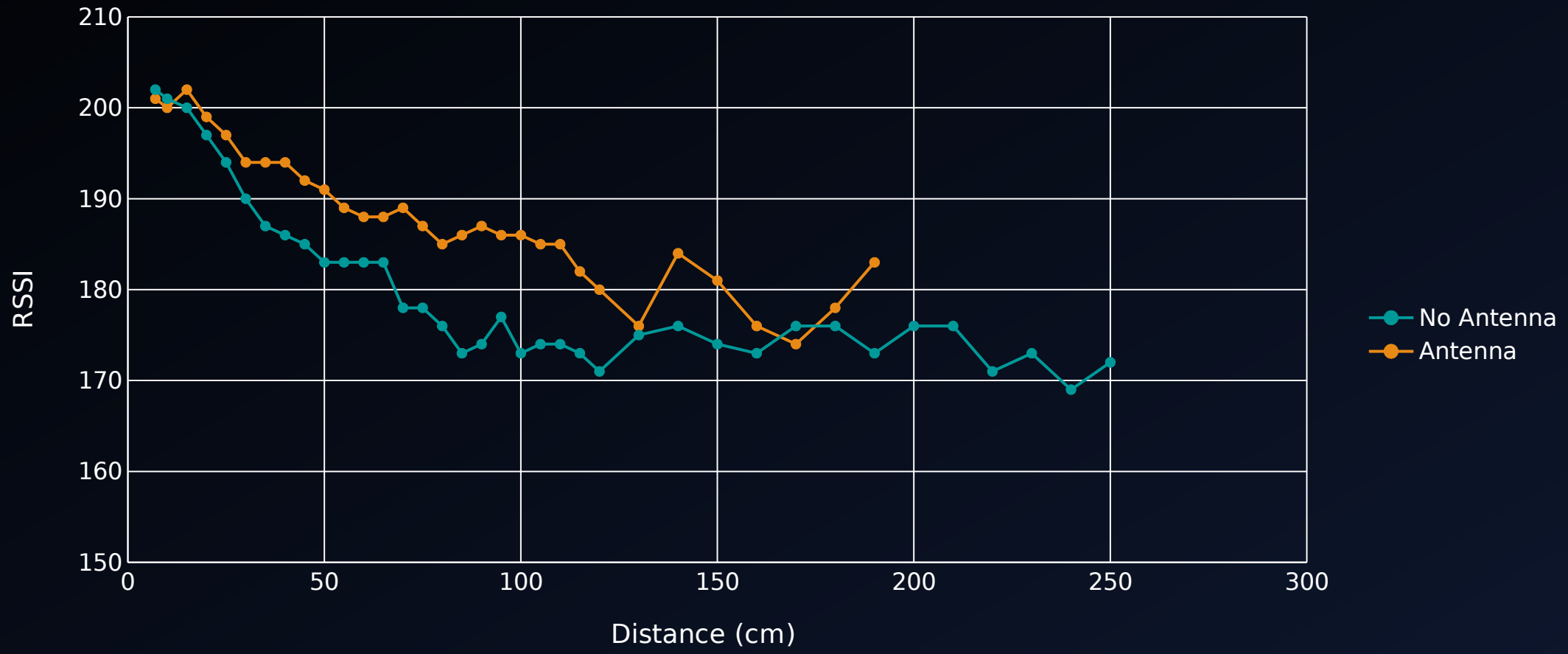


Testing Phase – Bluetooth RSSI

- Bluetooth Slave modules reply with a received signal strength indicator (RSSI) value to the inquiry from the master along with their address.
- RSSI is a function of distance to some extent, and hence can be used to locate a nearby Bluetooth device.
- We tested this feature of Bluetooth devices to see if we could use it for localisation with reasonable accuracy. Unfortunately, RSSI could only give differentiable values upto about 1m and that too in a single direction.
- In order to overcome this, we tried to increase the range of our modules by attaching an antenna. The antenna was designed to be a vertical helix so as to maintain symmetry in the radial plane.

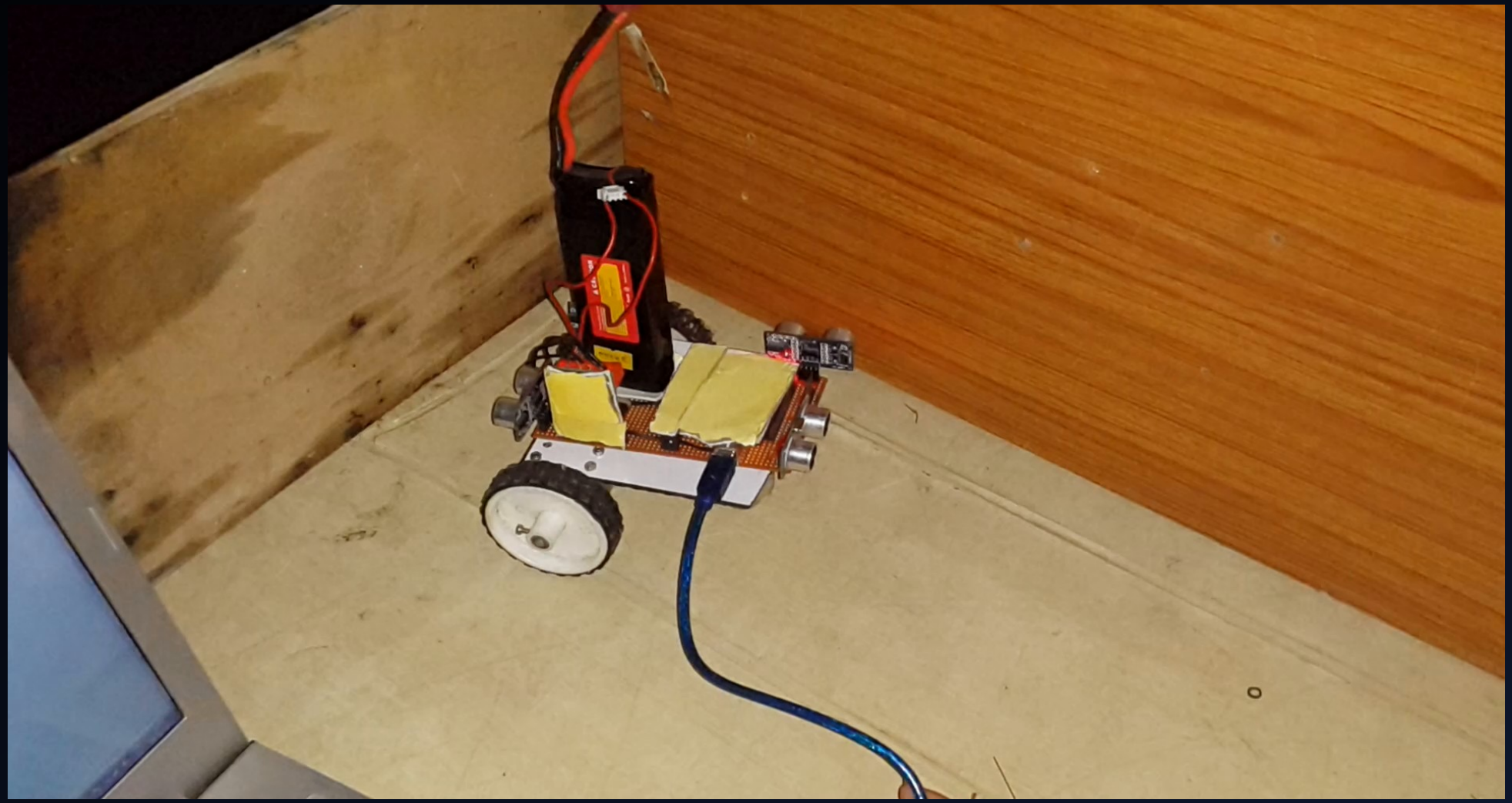


RSSI vs distance plot



Testing Phase – Wall following

- Ultrasonic sonic sensors are used for proximity detection. They work by sending a pulse of ultrasonic noise and calculating the distance based upon the time required to receive the echo.
- For wall following, we used two parallelly placed ultrasonic sensors separated by 5cm. We used PID control to track the error generated by taking difference of the two sensor values in order to keep the bot parallel to the wall.
- A pre alignment of the bot prior to wall following ensured that the bot moved at a nearly constant speed. At start and stopping of wall following as well as rotation, a start, stop and rotate bit is sent over the serial port to the mapper.



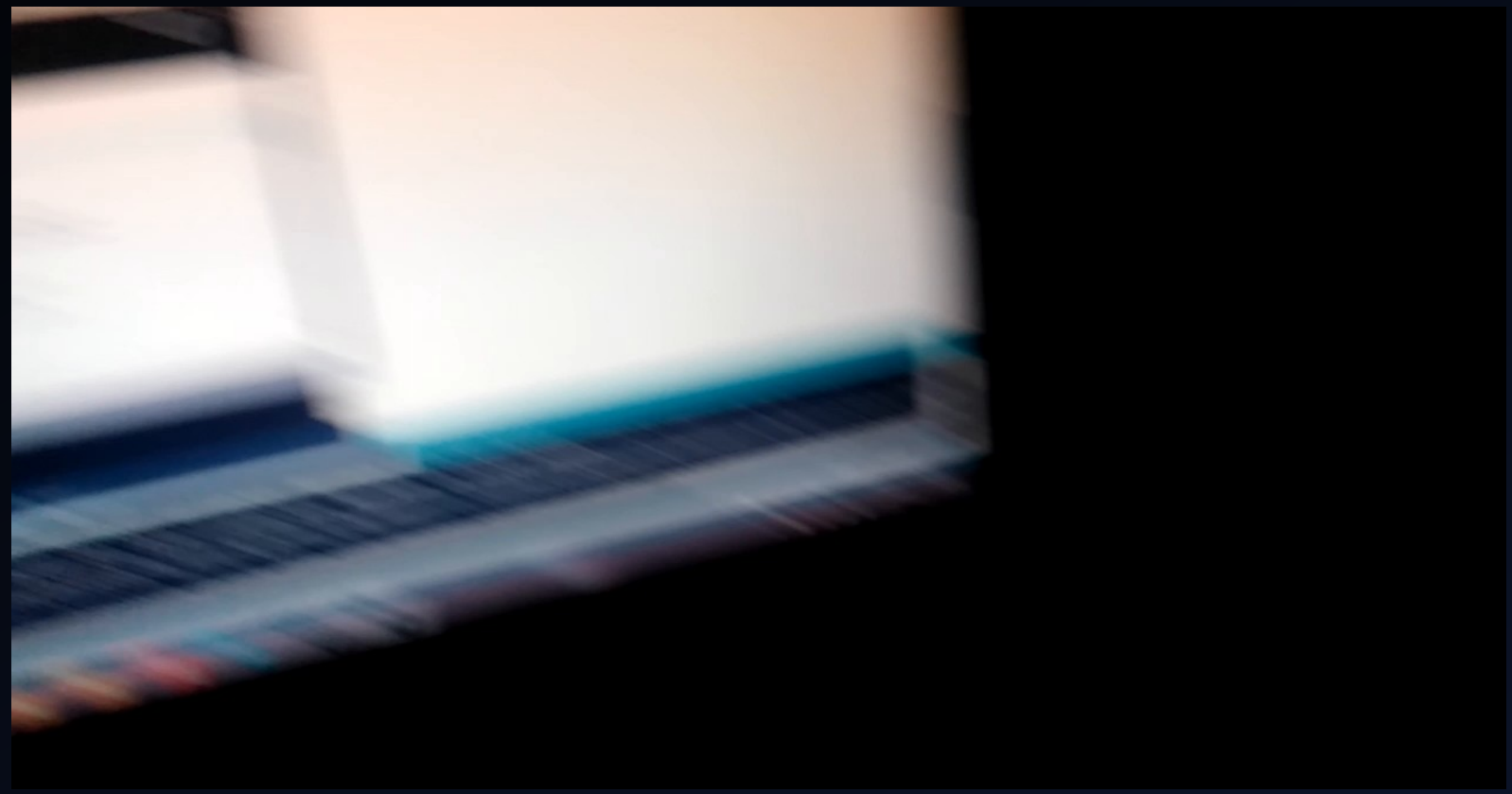
Localisation Plan - Mapping

- Realising the various shortcomings associated with the RSSI Values, we decided to use ultrasonic (wall following) for the map and then use RSSI for locating the nearest Bluetooth device to localise.
- For mapping, robot is to be placed at a predefined corner. The robot autonomously aligns itself to the nearest wall and backtracks to 3cm from the wall behind.
- Next, it moves parallel to the wall using wall following till it is 3cm away from the next wall. It then turns at a right angle and continues the process till it reaches its original position. All walls encountered during this motion are mapped.
- Corners of the room are then identified and Bluetooth modules are associated to them.


WE SHAPE YOUR IDEAS

Localisation Plan - Positioning

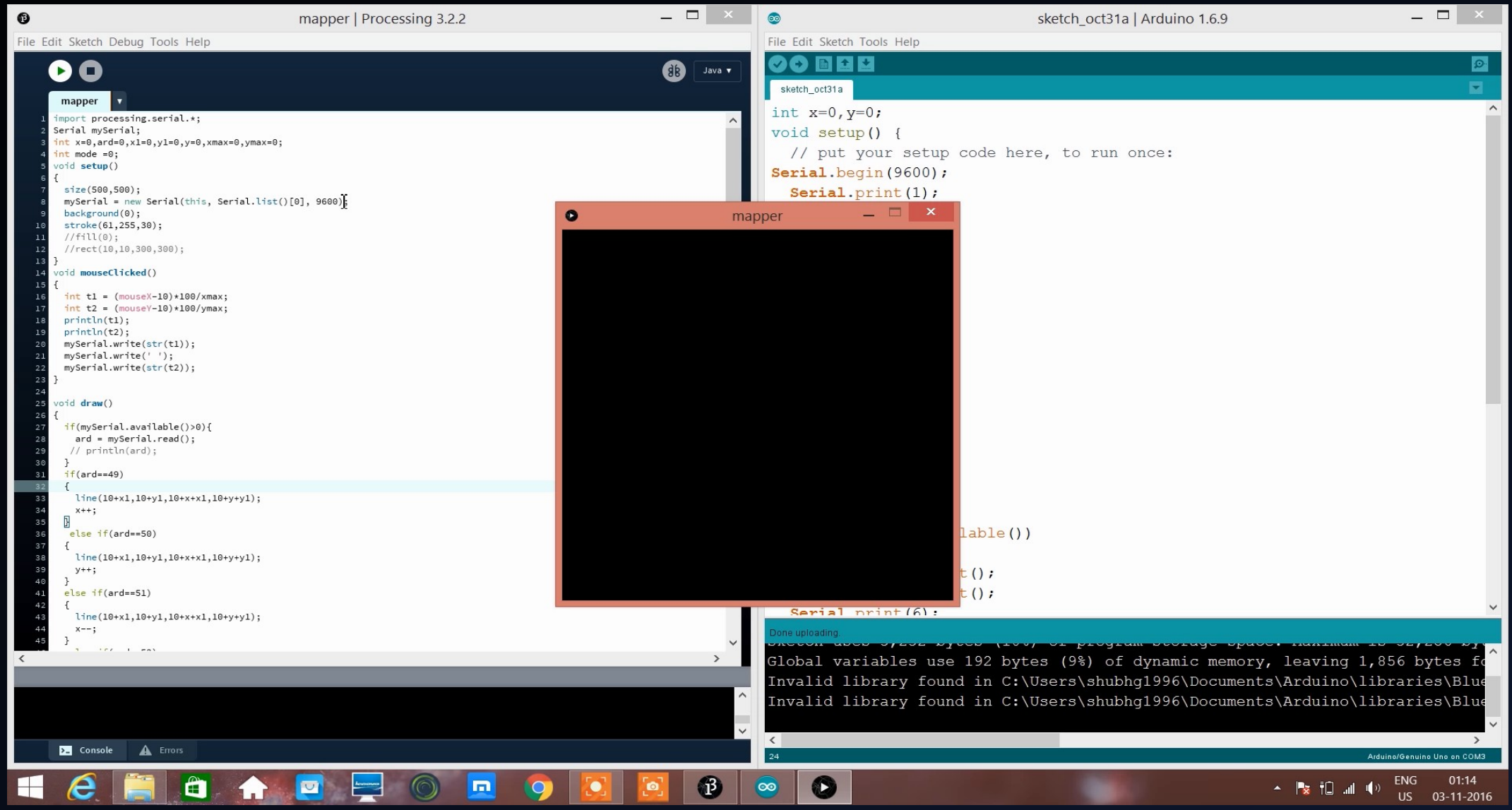
- After the map is generated, the robot can now be placed anywhere in the mapped room (power may be on or off). User then clicks on a position on the generated map where the bot is to be positioned.
- Bot on switching on in this mode will keep on moving straight till it encounters a wall. It will then align itself parallel to it and traverse to the first corner it encounters.
- Now, the bot will run a Bluetooth inquiry to locate the nearest Bluetooth module. Since each module is mapped to a corner, bots current location is identified and path to final position is determined.
- The bot then proceeds to reach the final position.



Mapping Software

- Processing IDE has been used to generate a map of the room. All communication between Arduino UNO and Processing has to be done over Serial port and hence custom protocols were developed for the process.
- For localisation, the mouse coordinates on the generated map are sent over to arduino as a ratio of the max dimension encoded to an integer.

Simulation


mapper | Processing 3.2.2
File Edit Sketch Debug Tools Help
Java
mapper
import processing.serial.*;
Serial mySerial;
int x=0,ard=0,x1=0,y1=0,y=0,xmax=0,ymax=0;
int mode =0;
void setup()
{
size(500,500);
mySerial = new Serial(this, Serial.list()[0], 9600);
background(0);
stroke(61,255,30);
//fill(0);
//rect(10,10,300,300);
}
void mouseClicked()
{
int t1 = (mouseX-10)*100/xmax;
int t2 = (mouseY-10)*100/ymax;
println(t1);
println(t2);
mySerial.write(str(t1));
mySerial.write(' ');
mySerial.write(str(t2));
}
void draw()
{
if(mySerial.available()>0){
ard = mySerial.read();
// println(ard);
}
if(ard==49)
{
line(10+x1,10+y1,10+x+x1,10+y+y1);
x++;
}
else if(ard==50)
{
line(10+x1,10+y1,10+x+x1,10+y+y1);
y++;
}
else if(ard==51)
{
line(10+x1,10+y1,10+x+x1,10+y+y1);
x--;
}
Console
Errors
mapper
sketch_oct31a | Arduino 1.6.9
File Edit Sketch Tools Help
sketch_oct31a
int x=0,y=0;
void setup() {
// put your setup code here, to run once:
Serial.begin(9600);
Serial.print(1);
Done uploading.
Global variables use 192 bytes (9%) of dynamic memory, leaving 1,856 bytes fo
Invalid library found in C:\Users\shubhg1996\Documents\Arduino\libraries\Blue
Invalid library found in C:\Users\shubhg1996\Documents\Arduino\libraries\Blue
24
Arduino/Genuino Uno on COM3
ENG
US
01:14
03-11-2016

Implementation of course contents

- Feedback control : PID
- Sensors and actuators : PWM
- Scheduling: Inquiry of Bluetooth module
- Quantisation: Mouse coordinates to Arduino serial
- I/O Hardware: Arduino UNO
- Serial Communication
- Parallel interfaces: Arduino, Bluetooth and laptop

Thank you!