

Principles of Integrated Engineering (PIE)

Mini-Project 3 - Section 2: Group 21

Sparsh Gupta, Akshat Jain

Franklin W. Olin College of Engineering

October 15, 2023

Introduction

We built a robot that follows a line path and utilizes DC motors and IR reflectance sensors. The robot is run using Arduino code and the motors are controlled using a motor shield.

Implementation and Functionality

Sensor Calibration

To find the desired R_{sense} value for the IR reflectance sensor, we utilize the following formula:

$$5V = V_{out} + I_c R_{sense}$$

$$V_{out} = 5V - I_c R_{sense}$$

Essentially, we need to obtain the analog voltage values from the IR sensors when they detect the line or the floor, and we need to record the difference in these values for our computations.

The $I_c^{floor} = 0.001A$ and $I_c^{line} = 0.00001A$, so if we choose a value of $100,000\Omega$ for our R_{sense} , we obtain the following values for V_{out} :

$$V_{out}^{floor} = |5V - 0.001A \cdot 100000\Omega| = 95V$$

$$V_{out}^{line} = |5V - 0.00001A \cdot 100000\Omega| = 4V$$

As we can see, the V_{out}^{floor} and V_{out}^{line} vary by a factor of more than 20 and it gives us a reasonable range to observe the differences when the IR sensor detects the floor or the line. Once we tested the IR sensors with this $R_{sense} = 100000\Omega$ value, we obtained analog voltage read values ranging from 200 to 900 depending on if it was detecting the floor or the line, and we decided that this was a perfectly working range for us to do our computations.

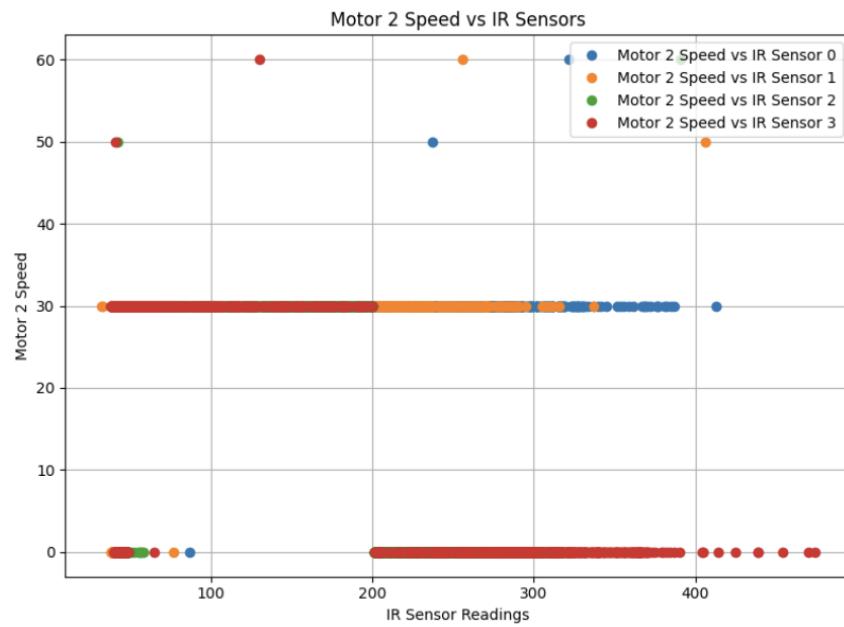
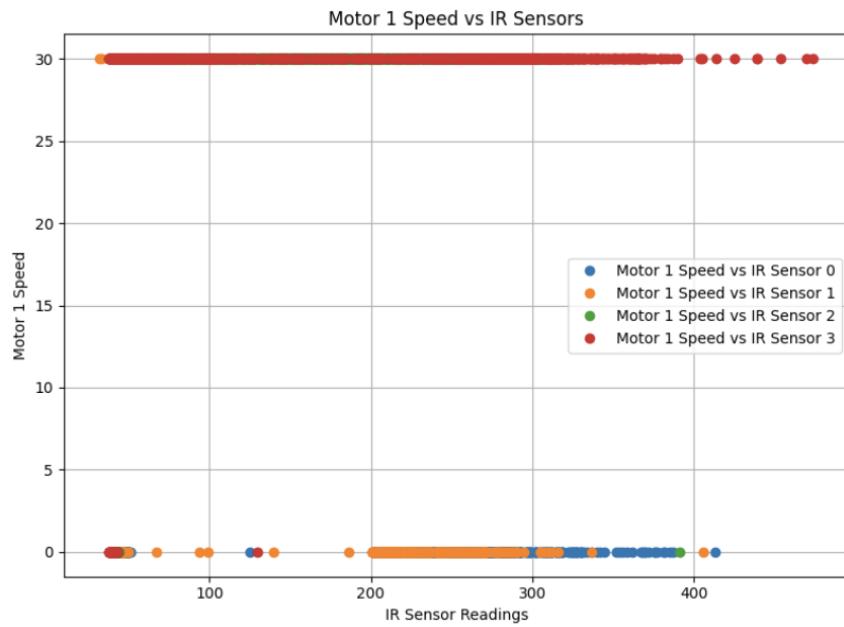
Control Logic

We employ an Adafruit motor shield to govern motor control. The initial motor speed is configured at 30 rpm. The analog sensor output is translated into a boolean result, hinging on a specified cutoff. When all four sensors uniformly report the same value, both motors proceed straight. When a single sensor exclusively detects the line, the opposing motor adjusts to steer the robot. Detection by 2 or 3 sensors implies an imminent sharp turn, prompting the robot to pivot until the leftmost or rightmost off-line sensor registers the line.

Serial Interface

The Serial interface is used to modify motor speeds by sending commands from the Arduino Serial port. For this, we identify if the Serial port is available first and then we can send commands through the Serial monitor command line. The commands we use are “W” to increase the speed by 10 and “S” to decrease the speed by 10.

Plot with motor commands and sensor data



Demo video with line following and behavior change

Line following: <https://youtu.be/6WaoDBJH2Vw>

Behaviour change: <https://youtu.be/TfLmgbF4W9s>

Mechanical Design

The chassis of the robotic car is composed of two DC motors attached to Omni-wheels for driving it. The Arduino and the Adafruit Motor Shield is mounted on top of the chassis. A breadboard is mounted on the bottom of the chassis such that the four IR reflectance sensors face the ground and the circuit is built on it. The electrical connections from the breadboard to the motor shield Analog Input pins and 5V/GND pins to obtain the IR sensor V_{out} values and have 5V/GND rails is done using jumper wires.

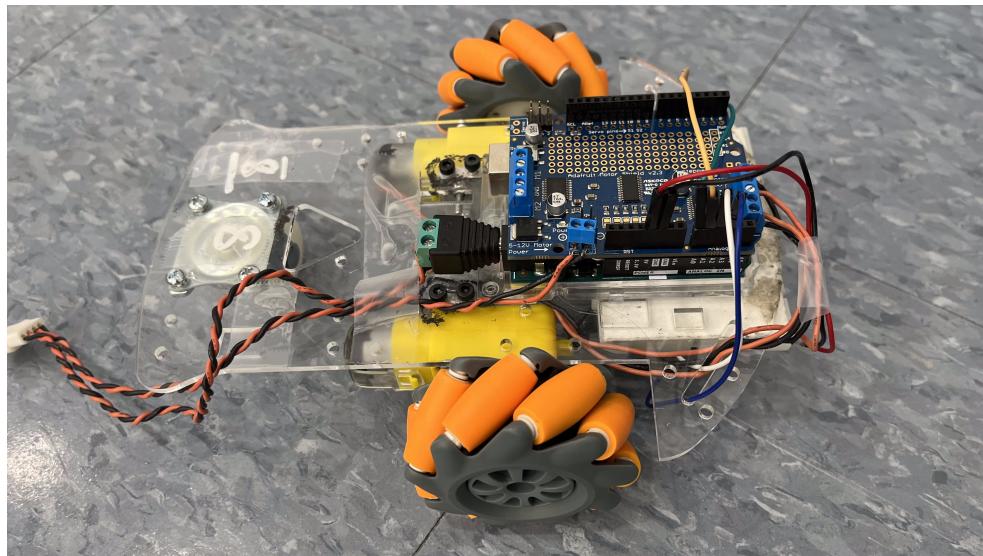


Figure 1: Robot - top view

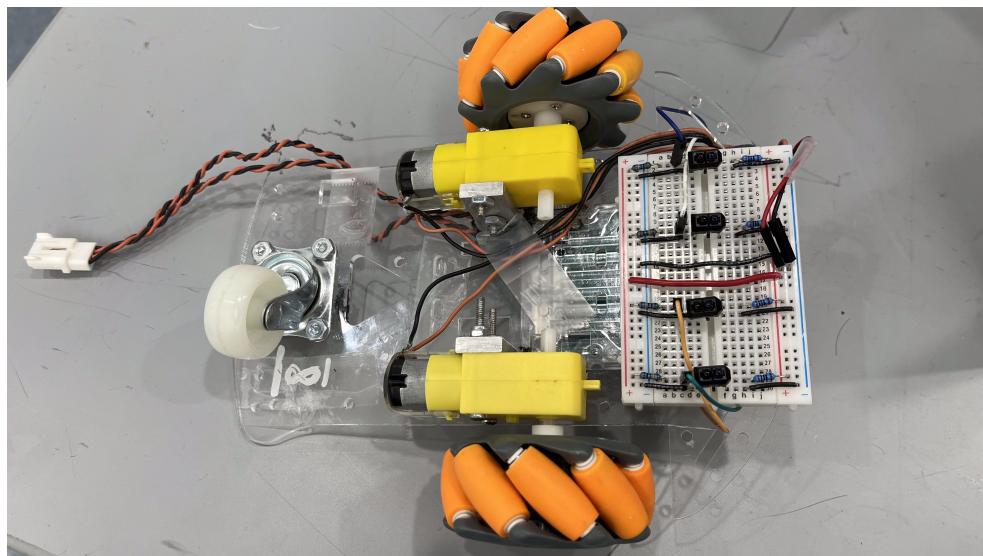


Figure 2: Robot - bottom view

Circuit Diagram

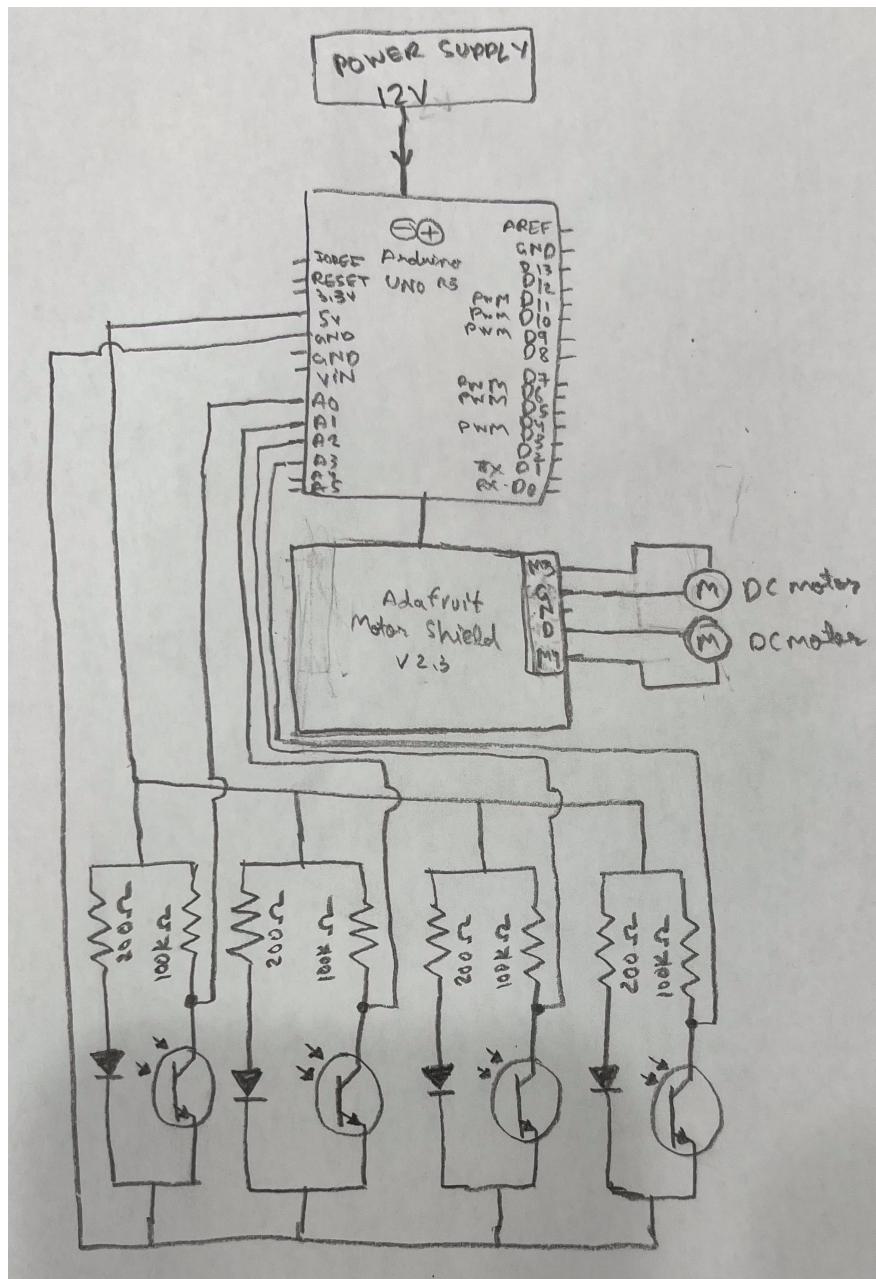


Figure 3: Circuit Diagram

Code

Arduino

```
1 #include <Adafruit_MotorShield.h>
2
3 Adafruit_MotorShield AFMS = Adafruit_MotorShield();
4 Adafruit_DCMotor *myMotor1 = AFMS.getMotor(3);
5 Adafruit_DCMotor *myMotor2 = AFMS.getMotor(4);
6
7 int IRSensor2 = A0; // IR sensor pin for left-most sensor
8 int IRSensor3 = A1; // IR sensor pin for middle-left sensor
9 int IRSensor4 = A2; // IR sensor pin for middle-right sensor
10 int IRSensor5 = A3; // IR sensor pin for right-most sensor
11
12 int cutoffValue = 200;
13 int speed = 30;
14
15 void setup() {
16
17     // Start motor with the default frequency 1.6KHz
18     AFMS.begin();
19
20     Serial.begin(9600); // Set Serial Baud Rate
21
22     // Initialize motors
23     myMotor1->setSpeed(0); // Set Motor 1 speed
24     myMotor2->setSpeed(0); // Set Motor 2 speed
25
26     // Initialize IR sensors
27     pinMode(IRSensor2, INPUT); // IR Sensor 2 pin INPUT
28     pinMode(IRSensor3, INPUT); // IR Sensor 3 pin INPUT
29     pinMode(IRSensor4, INPUT); // IR Sensor 4 pin INPUT
30     pinMode(IRSensor5, INPUT); // IR Sensor 5 pin INPUT
31 }
32
33 void loop() {
34     // IR sensor readings
35     int IR2 = analogRead(IRSensor2); // Set the IR Sensor 2 as Input
36     int IR3 = analogRead(IRSensor3); // Set the IR Sensor 3 as Input
37     int IR4 = analogRead(IRSensor4); // Set the IR Sensor 4 as Input
38     int IR5 = analogRead(IRSensor5); // Set the IR Sensor 5 as Input
39
40     // Print sensor readings
41     Serial.print(IR2);
42     Serial.print(",");
43     Serial.print(IR3);
```

```
44 Serial.print(",");
45 Serial.print(IR4);
46 Serial.print(",");
47 Serial.print(IR5);
48 Serial.print(",");
49
50 // Classify IR sensor readings as 0 or 1 based on cutoffValue
51 int IR2state = (IR2 > cutoffValue) ? 1 : 0;
52 int IR3state = (IR3 > cutoffValue) ? 1 : 0;
53 int IR4state = (IR4 > cutoffValue) ? 1 : 0;
54 int IR5state = (IR5 > cutoffValue) ? 1 : 0;
55
56 // Modify motor speeds using serial commands
57 if (Serial.available() > 0) {
58     char command = Serial.read();
59     if (command == 'W') {
60         // Increase speed
61         speed += 10;
62     } else if (command == 'S') {
63         // Decrease speed
64         speed -= 10;
65     }
66 }
67
68 if (IR2state==0 & IR3state==0 & IR4state==0 & IR5state==0) {
69     int motor1Speed = speed;
70     int motor2Speed = speed;
71
72     myMotor1->setSpeed(motor1Speed);
73     myMotor2->setSpeed(motor2Speed);
74
75     Serial.print(motor1Speed);
76     Serial.print(",");
77     Serial.println(motor2Speed);
78 }
79 if (IR2state==1 & IR3state==1 & IR4state==1 & IR5state==1) {
80     int motor1Speed = speed;
81     int motor2Speed = speed;
82
83     myMotor1->setSpeed(motor1Speed);
84     myMotor2->setSpeed(motor2Speed);
85
86     Serial.print(motor1Speed);
87     Serial.print(",");
88     Serial.println(motor2Speed);
89 }
```

```
90  else if (IR2state==0 & IR3state==0 & IR4state==1 & IR5state==0) {  
91      int motor1Speed = speed;  
92      int motor2Speed = 0;  
93  
94      myMotor1->setSpeed(motor1Speed);  
95      myMotor2->setSpeed(motor2Speed);  
96  
97      Serial.print(motor1Speed);  
98      Serial.print(",");  
99      Serial.println(motor2Speed);  
100 }  
101  
102 else if (IR2state==0 & IR3state==0 & IR4state==0 & IR5state==1) {  
103     int motor1Speed = speed;  
104     int motor2Speed = 0;  
105  
106     myMotor1->setSpeed(motor1Speed);  
107     myMotor2->setSpeed(motor2Speed);  
108  
109     Serial.print(motor1Speed);  
110     Serial.print(",");  
111     Serial.println(motor2Speed);  
112 }  
113  
114 else if (IR2state==0 & IR3state==1 & IR4state==0 & IR5state==0) {  
115     int motor1Speed = 0;  
116     int motor2Speed = speed;  
117  
118     myMotor1->setSpeed(motor1Speed);  
119     myMotor2->setSpeed(motor2Speed);  
120  
121     Serial.print(motor1Speed);  
122     Serial.print(",");  
123     Serial.println(motor2Speed);  
124 }  
125  
126 else if (IR2state==1 & IR3state==0 & IR4state==0 & IR5state==0) {  
127     int motor1Speed = 0;  
128     int motor2Speed = speed;  
129  
130     myMotor1->setSpeed(motor1Speed);  
131     myMotor2->setSpeed(motor2Speed);  
132  
133     Serial.print(motor1Speed);  
134     Serial.print(",");  
135     Serial.println(motor2Speed);
```

```
136     }
137
138     else if (IR2state==1 & IR3state==1 & IR4state==1 & IR5state==0) {
139         int motor1Speed = 0;
140         int motor2Speed = speed+30;
141
142         Serial.print(motor1Speed);
143         Serial.print(",");
144         Serial.println(motor2Speed);
145
146         while (IR5state==0){
147             IR5 = analogRead(IRSensor5); // Set the IR Sensor 5 as Input
148             IR5state = (IR5 > cutoffValue) ? 1 : 0;
149             myMotor1->setSpeed(motor1Speed);
150             myMotor2->setSpeed(motor2Speed);
151         }
152         myMotor1->setSpeed(speed);
153         myMotor2->setSpeed(0);
154         delay(200);
155     }
156
157     else if (IR2state==1 & IR3state==1 & IR4state==0 & IR5state==0) {
158         int motor1Speed = 0;
159         int motor2Speed = speed+20;
160
161         Serial.print(motor1Speed);
162         Serial.print(",");
163         Serial.println(motor2Speed);
164
165         while (IR5state==0){
166             IR5 = analogRead(IRSensor5); // Set the IR Sensor 5 as Input
167             IR5state = (IR5 > cutoffValue) ? 1 : 0;
168             myMotor1->setSpeed(motor1Speed);
169             myMotor2->setSpeed(motor2Speed);
170         }
171
172         myMotor1->setSpeed(speed);
173         myMotor2->setSpeed(0);
174         delay(200);
175     }
176
177     // Run motors
178     myMotor1->run(FORWARD);
179     myMotor2->run(FORWARD);
180 }
```

Python Plotting code

```
181 import serial
182 import matplotlib.pyplot as plt
183
184 arduinoComPort = "/dev/cu.usbmodem12101"
185 baudRate = 9600
186
187 serialPort = serial.Serial(arduinoComPort, baudRate, timeout=1)
188
189 # Initialize data lists
190 motor1_speeds = []
191 motor2_speeds = []
192 sensor0_readings = []
193 sensor1_readings = []
194 sensor2_readings = []
195 sensor3_readings = []
196
197 # Set the maximum number of readings
198 max_readings = 4000
199 reading_count = 0
200
201 # Main loop to read data from the Arduino, then display it
202 while reading_count < max_readings:
203     lineOfData = serialPort.readline().decode()
204
205     if "end\r\n" in lineOfData:
206         break
207
208     elif len(lineOfData) > 0:
209         print(lineOfData)
210         data = list(map(float, lineOfData.split(',')))
211
212         sensor0_readings.append(data[0])
213         sensor1_readings.append(data[1])
214         sensor2_readings.append(data[2])
215         sensor3_readings.append(data[3])
216         motor1_speeds.append(data[4])
217         motor2_speeds.append(data[5])
218
219         reading_count += 1
220
221 # Create subplots for Motor 1 speed vs all IR sensors and Motor 2 speed vs
222 # all IR sensors
223 plt.figure(figsize=(16, 6))
224
225 # Motor 1 Speed vs IR Sensor Data
226 plt.subplot(1, 2, 1)
227 plt.plot(sensor0_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
Sensor 0')
228 plt.plot(sensor1_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
Sensor 1')
```

```
    Sensor 1')
228 plt.plot(sensor2_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
      Sensor 2')
229 plt.plot(sensor3_readings, motor1_speeds, 'o', label='Motor 1 Speed vs IR
      Sensor 3')
230 plt.xlabel('IR Sensor Readings')
231 plt.ylabel('Motor 1 Speed')
232 plt.legend()
233 plt.title('Motor 1 Speed vs IR Sensors')
234 plt.grid(True)
235
236 # Motor 2 Speed vs IR Sensor Data
237 plt.subplot(1, 2, 2)
238 plt.plot(sensor0_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
      Sensor 0')
239 plt.plot(sensor1_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
      Sensor 1')
240 plt.plot(sensor2_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
      Sensor 2')
241 plt.plot(sensor3_readings, motor2_speeds, 'o', label='Motor 2 Speed vs IR
      Sensor 3')
242 plt.xlabel('IR Sensor Readings')
243 plt.ylabel('Motor 2 Speed')
244 plt.legend()
245 plt.title('Motor 2 Speed vs IR Sensors')
246 plt.grid(True)
247
248 plt.tight_layout()
249 plt.show()
```

Reflection

Software: We started by using the inbuilt PID library for the project but then faced problems with the library so switched to simple logic and conditional statements for the code.

Electrical: We faced several electrical problems because we unfortunately got defective motors and motor shield. SO, we had to replace the DC motors four times to finally obtain motors which were working. When using the motor shield, our Serial communication was not working because we would stop getting IR sensor data as soon as we would keep the robot on the ground because the sensors were connected to A2 through A5, and we found out that the shield uses the SDA and SCL i2c pins on pins A4 and A5 to control DC and stepper motors and that was causing this problem. So, we switched to using A0 and A1 instead after removing stuff that was blocking those pins in the motor shield.

Mechanical: The mechanical design was pretty simple where we attached a breadboard with sensors on the bottom of the robot, and the motors were attached using plates to the chassis of the robot. Because the old motors that we were using were defective, one of the motors in this process started smoking because it was unable to rotate. But, everything else was very smooth in terms of the mechanical aspect.

Teaming: The teaming was pretty smooth and we contributed equally to the project while achieving our learning goals for the project. We both did software, electrical and mechanical things equally as well.