THE AMAZING RACE PROJECT

B03

SECTION 2

TEAM 2

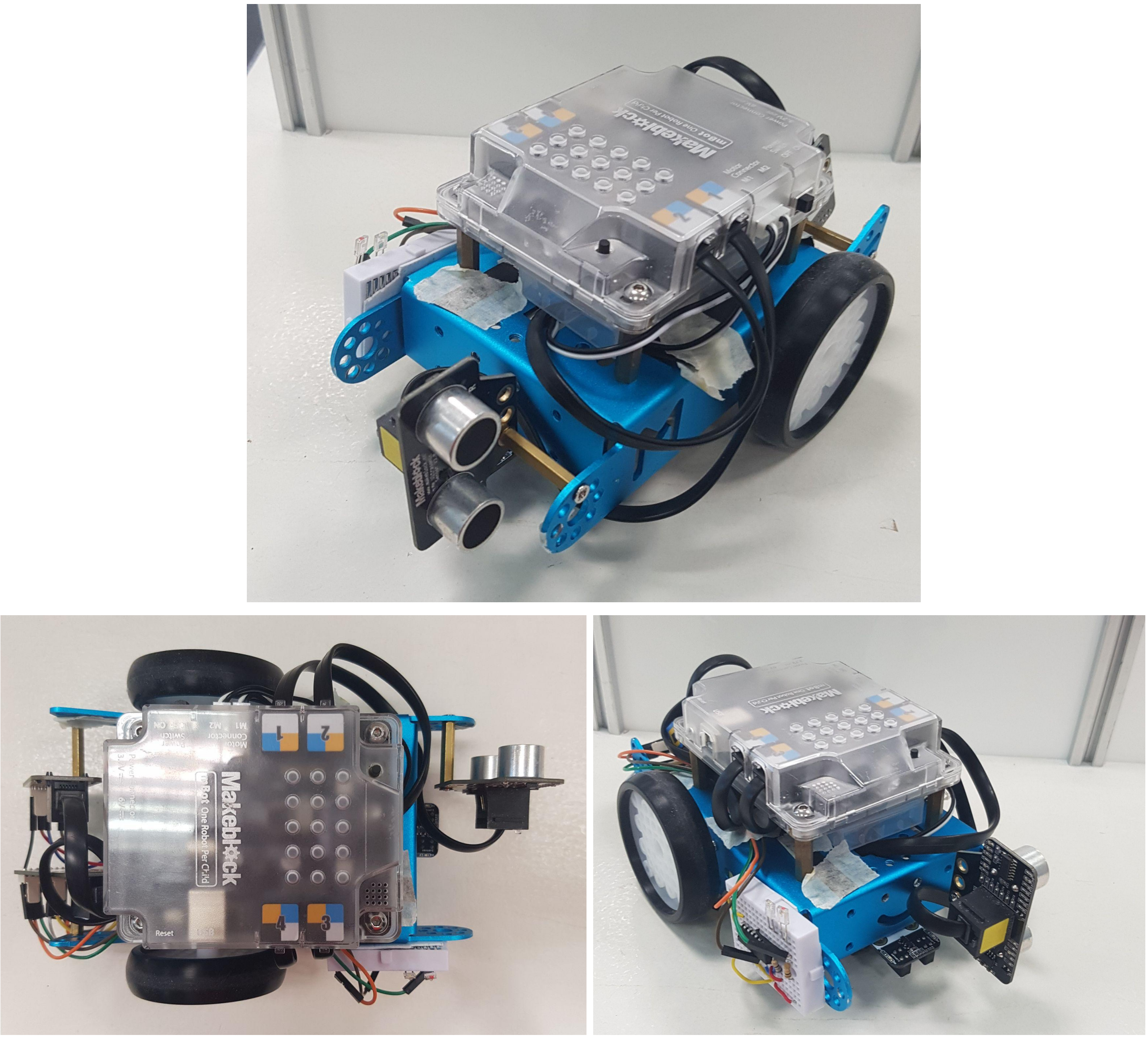
NIKHIL BABU

NICHOLAS TAN YUN YU

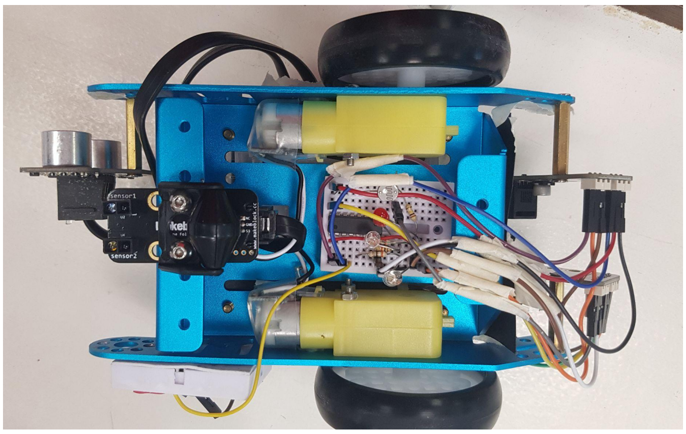
NG YAN HUI

NG XIN ZHE SEAN

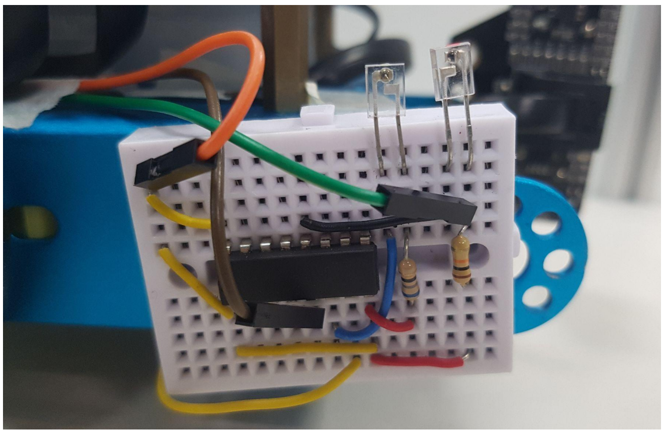
Pictures of mBot and Circuits



(clockwise from top) top view of robot, front right view, front left view



Colour Sensor Circuit



IR Sensor Circuit

Overall Algorithm

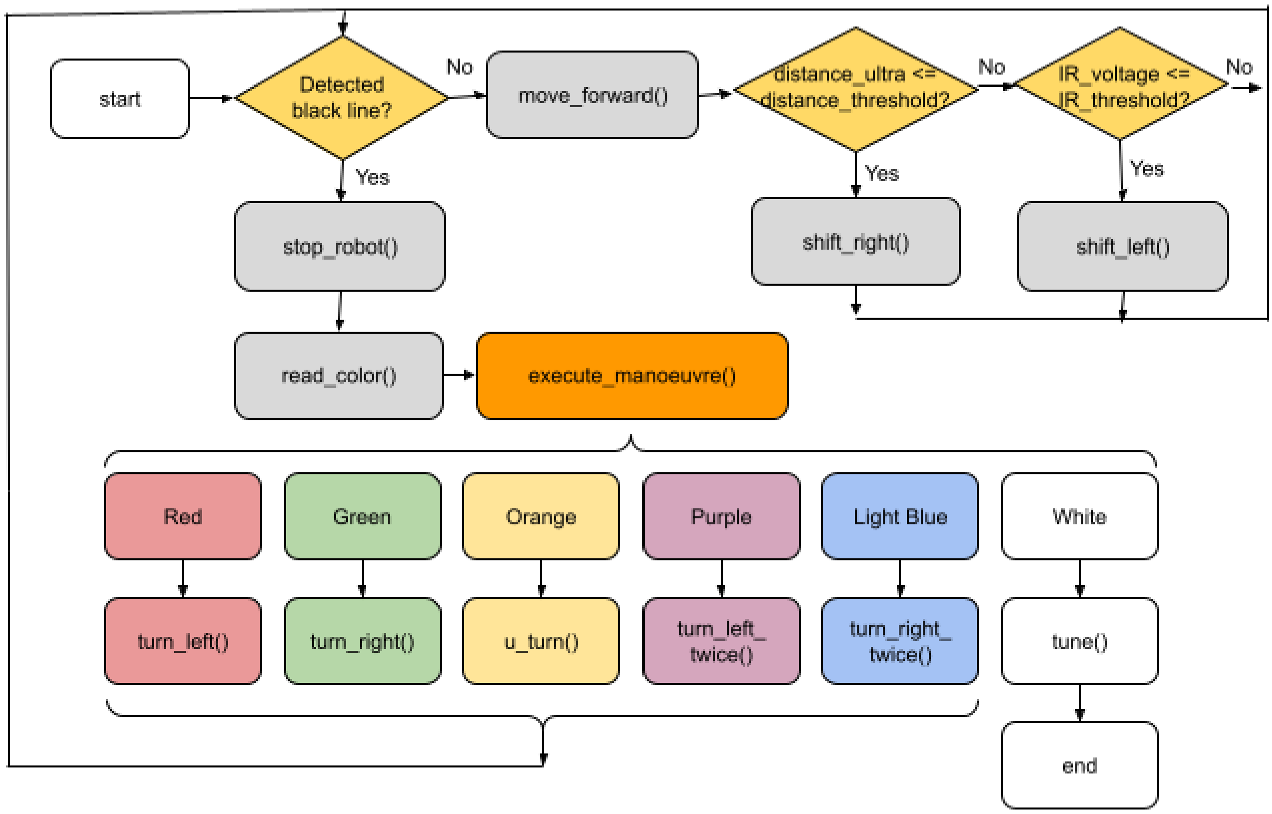


Figure 1 - Overall Algorithm

Our robot is first activated after the user engages the ON button. After which, the robot will run in a continuous loop as above. (Figure 1)

Detected black line

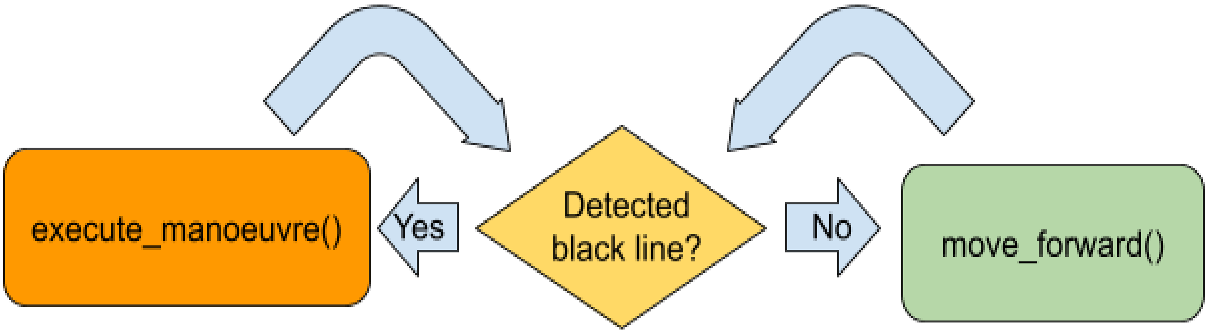


Figure 2 - Detected Black Line Algorithm

With the MakeBlock line sensor at the front of the robot, the robot will continuously detect the presence of a black line. Upon detection, the robot stops its movement and proceeds to do the Waypoint challenge. Otherwise, the Ultrasonic distance sensor and Infrared (IR) proximity sensors will be activated to ensure that the robot does not collide into the side walls as it moves forward.

Waypoint Challenge

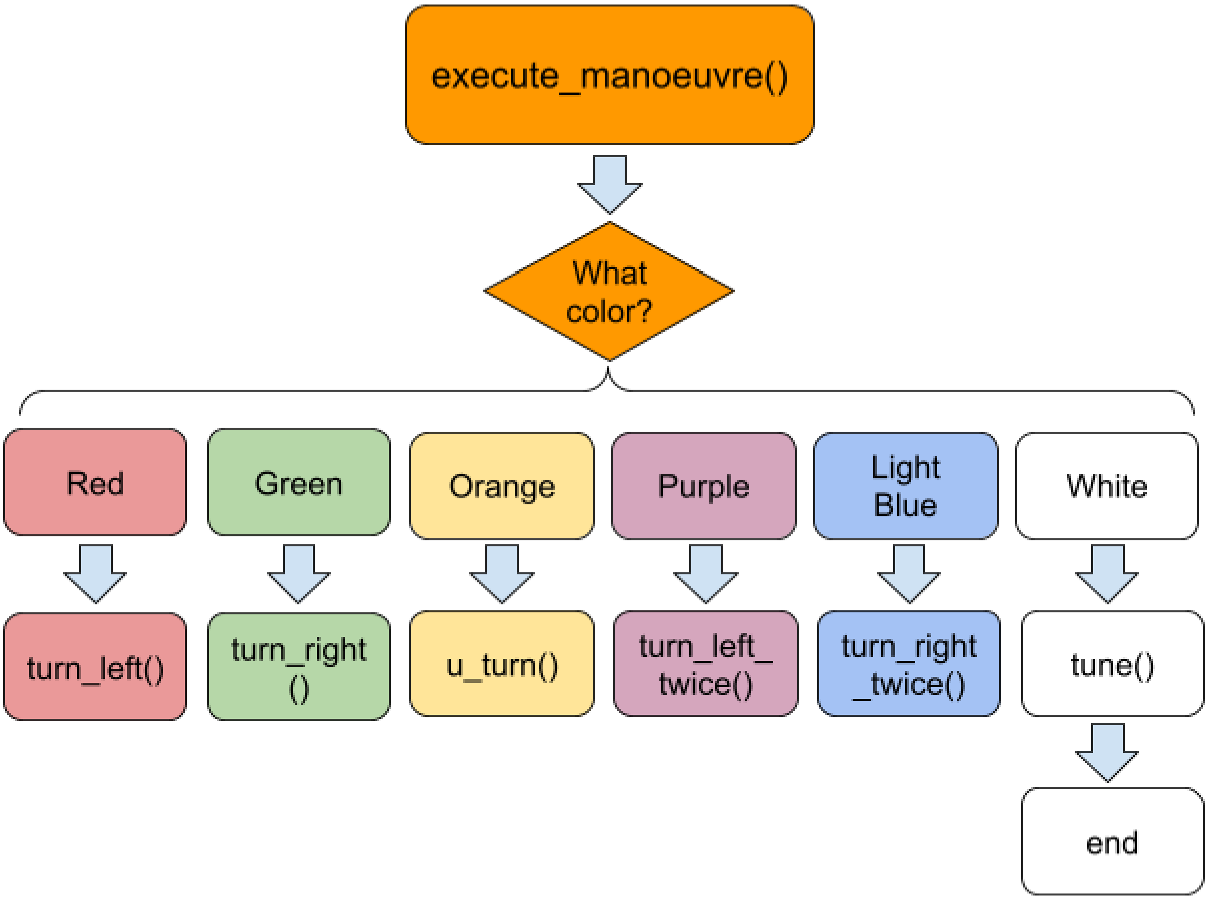


Figure 3 - Waypoint Challenge Algorithm

Figure 3 shows the algorithm flow behind the waypoint challenge. Using our colour sensor, which systematically switches on red, green and blue LEDs and determines the colour based on the Light Dependent Resistor (LDR) voltage, our robot moves accordingly.

Moving forward

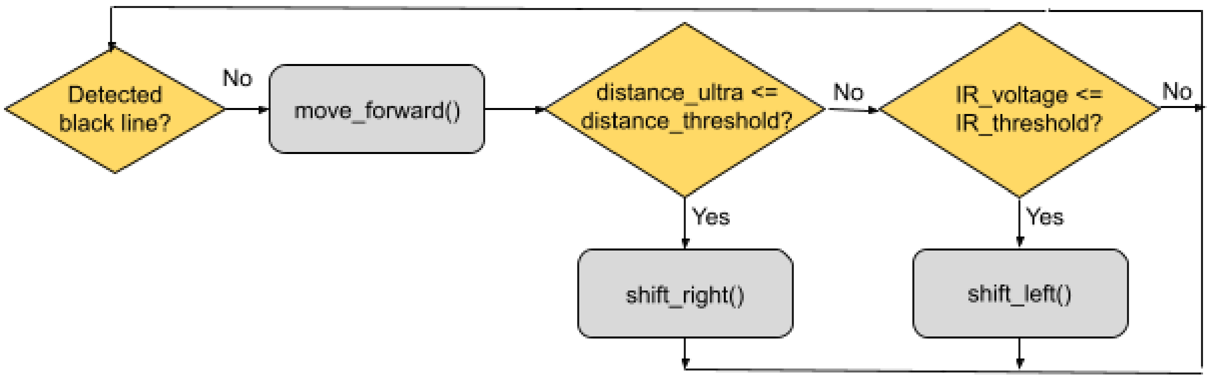


Figure 4 - Moving Forward Algorithm

Figure 4 shows how the ultrasonic (left side of the robot) and IR (right side of the robot) sensors guide the robot forward. The sensors will continuously scan for the distance away from both walls and execute the appropriate code depending on the distance measured.

Implementation of subsystems

1. Black Line Detector

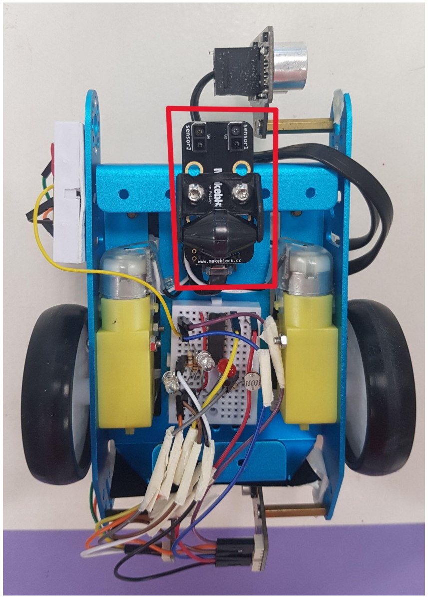


Figure 5 - MakeBlock Line sensor

The first component of our code is to detect the presence of a black line using the MakeBlock Line sensor. Using the *indirect incidence* method, the line sensor continuously emits IR rays from its 2 inbuilt IR emitters, and detects the amount of reflected IR with 2 IR detectors (S1, S2). As black absorbs IR, the IR detector will detect significantly less IR when the robot is on a black line. There are 4 possible outputs: S1_IN_S2_IN, S1_IN_S2_OUT, S1_OUT_S2_IN and S1_OUT_S2_OUT. We programmed the sensors such that the sensor will return S1_IN_S2_IN when both sensors detect the black line, and the waypoint challenge code will be activated, as shown below:

```
// Sensing black strip on ground
bool detected_black_line()
{
    // Return true when both sensors detect the black line
    int sensor_state = lineFinder.readSensors();
    return sensor_state == S1_IN_S2_IN;
}
```

Figure 6 - Code to detect black line

2. Waypoint Challenge

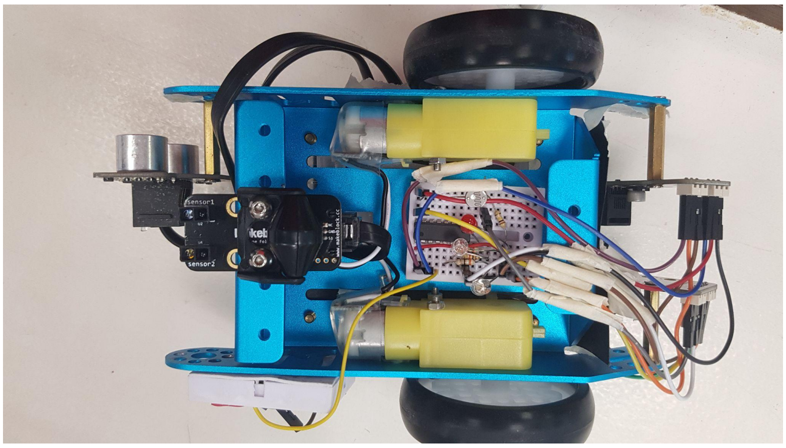


Figure 7 - Colour sensor

Upon detecting a black line, the robot will come to an immediate stop and activate the colour sensor. The colour sensor will systematically switch on the Red, Green, and Blue LEDs and measure the voltage across the Light Dependent Resistor (LDR) whose resistance, and hence voltage, varies according to the amount of light received. With this characteristic in mind, we first established the upper and lower bound of values: namely white and black respectively: white will reflect the most amount of light in R, G and B domains, resulting in the lowest voltage across the LDR (see Figure 8) and highest reading from analogRead(), while black will reflect the least amount of light and result in the lowest reading from analogRead().

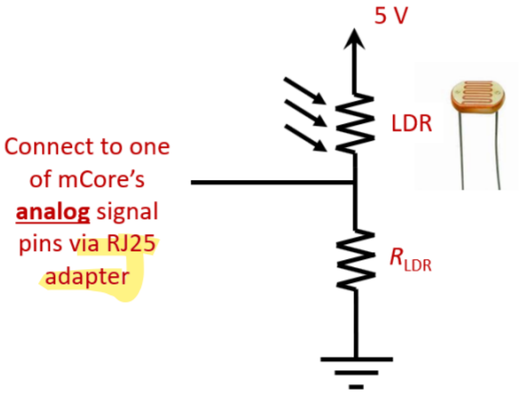


Figure 8 - from The A-maze-ing Race handout

As the RGB colour code is {0, 255}, we can get the RGB colour code for an unknown colour from:

```
// Get the average of 10 readings and rebase to 255
final_RGB[i] = (get_average_reading(10) - blackArray[i])/(greyDiff[i])*255;
```

Figure 9 - RGB Equation

where blackArray[ ] contains the analogRead values of black and greyDiff[ ]= whiteArray[ ] - blackArray[ ] , where whiteArray[ ] contains the analogRead values of white, and greyDiff contains the range of analogRead values.
Hence, from the RGB values, the robot can determine the unknown colour and execute the appropriate manoeuvre.

3. Forward Movement

In the case when the MakeBlock Line sensor does not return S1_IN_S2_IN (no black line detected), the robot will execute the forward movement code, relying on the ultrasonic sensor and IR sensor to avoid collision with the side walls. Even with a missing side wall, the robot continues traversing in a straight path.

Ultrasonic sensor

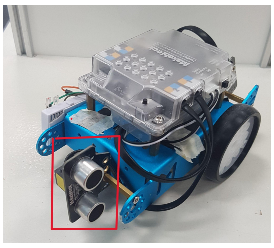


Figure 10 - Ultrasonic sensor

```
// Measure the distance from the left of the mBot to the wall
distance_ultra = ultraSensor.distanceCm();

// Robot is too close to the left wall
if (distance_ultra <= distance_threshold)
{
    Serial.println("Too close to left wall, shifting right...");
    shift_right();
}
```

Figure 11 - Ultrasonic sensor code

Located at the left side of our robot, the ultrasonic sensor continuously returns the distance of the robot from the left wall, and executes a shift_right() algorithm should the distance returned be smaller than our predetermined threshold distance of 7cm (see Figure 11). This was determined after measuring the maze's width (~30cm), our robot's width, as well as the average turning radius.

IR sensor

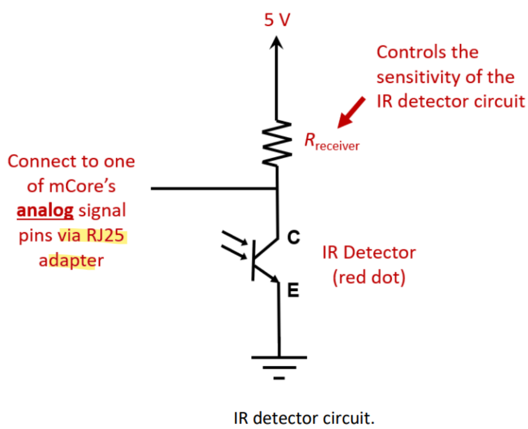


Figure 12 - IR detector circuit

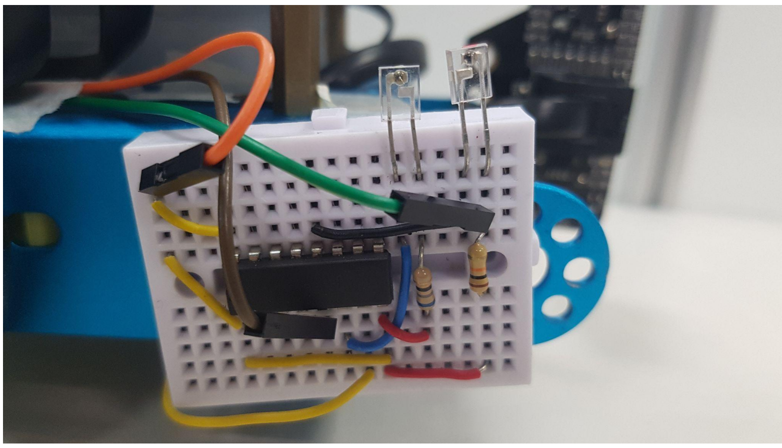


Figure 13 - IR sensor

```
// magnitude of IR_voltage increases as distance from the right wall increases
if (IR_voltage <= IR_threshold)
{
    Serial.println("Too close to right wall, shifting left...");
    shift_left();
}
```

Figure 14 - IR sensor code

Located at the right side of our robot, the sensor complements the ultrasonic sensor and returns the distance from the robot to the right wall, executing a shift_left() algorithm should the voltage across the IR detector be less than 3.15V (Figure 14). This corresponds to ~4 cm distance from the wall and was determined after multiple test runs.

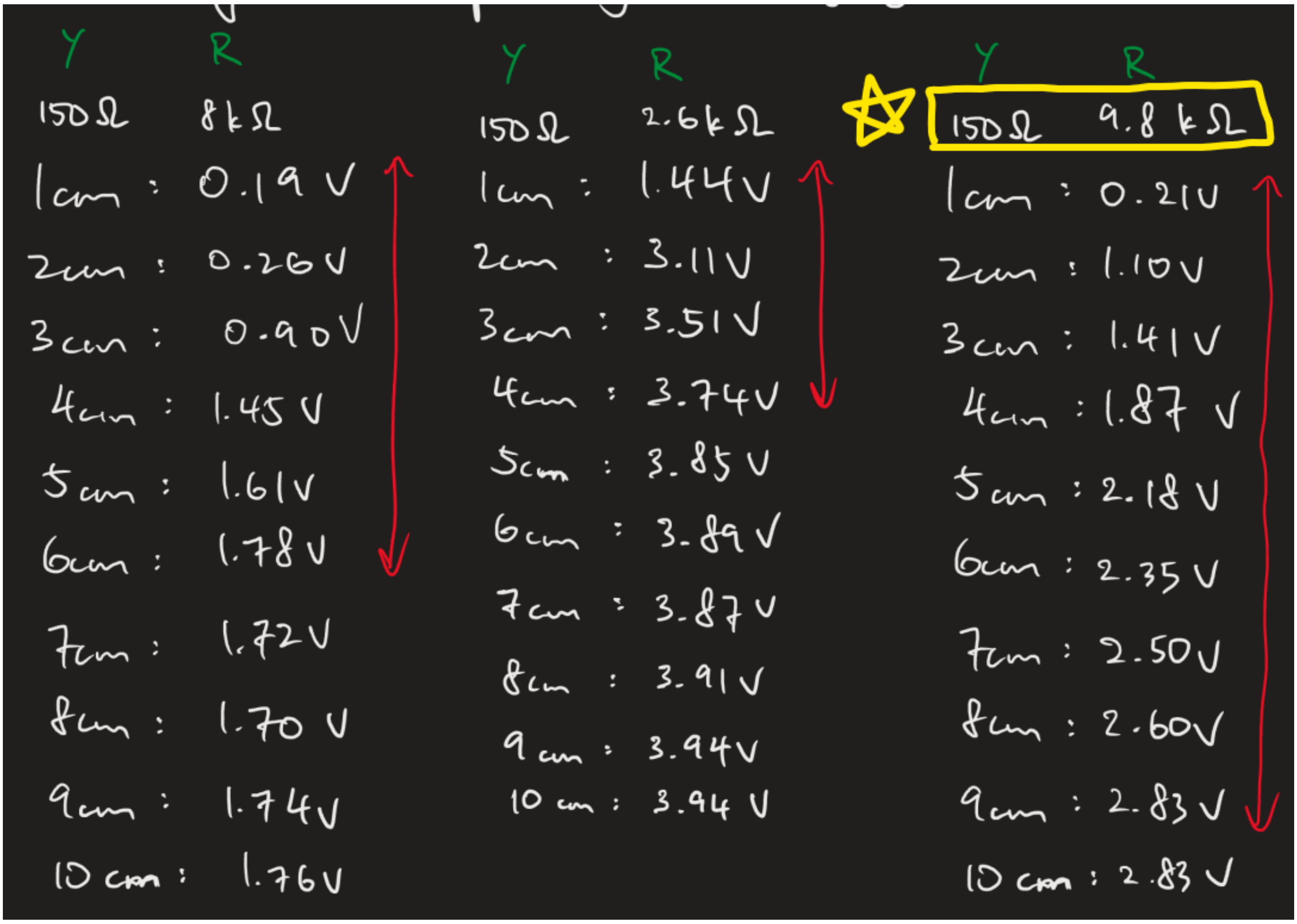
Calibrating and Improving Robustness of Sensors

IR sensor

We had 2 considerations: 1. To increase the sensitivity and 2. To reduce the effect of ambient IR to improve robustness of our IR sensor.

<u>IR sensor sensitivity (IR detector)</u>

First, increasing the sensitivity of our IR sensor meant for the same change in distance, there is a greater change in voltage. In addition, we also had to ensure the voltage remains unsaturated while the distance is below 6 cm (refer to red arrows in Figure 15): the distance from the IR sensor to the maze wall when our robot is at the centre of the path. Hence, we varied the resistor values of the IR detector. Figure 15 shows 3 of the many resistors we experimented with:



Y	R	Y	R	Y	R
150Ω	8kΩ	150Ω	2.6kΩ	150Ω	9.8kΩ
1cm :	0.19V	1cm :	1.44V	1cm :	0.21V
2cm :	0.26V	2cm :	3.11V	2cm :	1.10V
3cm :	0.90V	3cm :	3.51V	3cm :	1.41V
4cm :	1.45V	4cm :	3.74V	4cm :	1.87V
5cm :	1.61V	5cm :	3.85V	5cm :	2.18V
6cm :	1.78V	6cm :	3.89V	6cm :	2.35V
7cm :	1.72V	7cm :	3.87V	7cm :	2.50V
8cm :	1.70V	8cm :	3.91V	8cm :	2.60V
9cm :	1.74V	9cm :	3.94V	9cm :	2.83V
10cm :	1.76V	10cm :	3.94V	10cm :	2.83V

Figure 15 - Resistor values for IR detector

We chose the IR detector resistor to be 9.8kΩ as it fulfilled the criteria of having a significant increase in voltage for a change in distance, while the distance is 6 cm and below, and giving us the flexibility to have a greater IR threshold distance should we decide to increase beyond 6 cm.

IR sensor sensitivity (IR emitter)

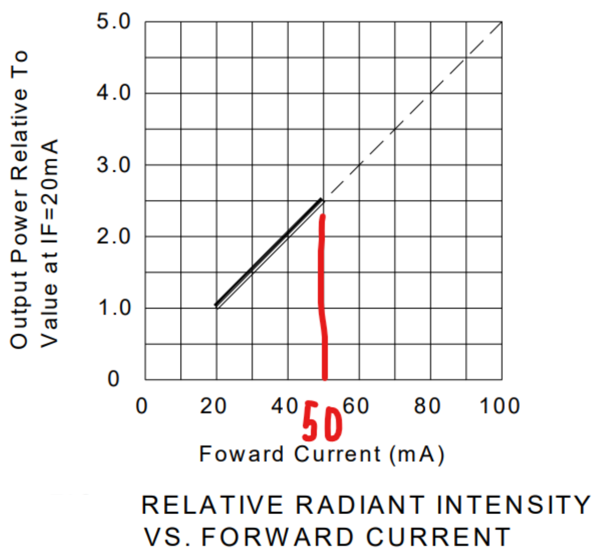


Figure 16 - I_{max} of IR emitter

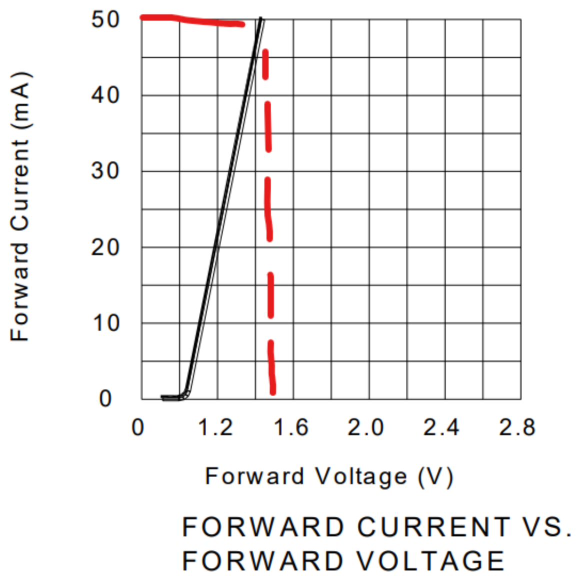


Figure 17 - approximate V_{max} at I_{max}

Another way of increasing the sensitivity of the IR detector is by increasing the amount of IR emitted by the IR emitter: with more IR from the emitter, the maximum amount of IR detected by the IR detector is greater. Consequently, the corresponding change in voltage will be greater. To emit the maximum amount of IR, the current across the IR emitter must be the maximum.

As the maximum voltage and current across the IR emitter without damaging it are 1.5V (Figure 17) and 50mA (Figure 16) respectively, from $R_{max} = V_{max} / I_{max}$, the resistor value to achieve this is (5 - 1.5)V / (50mA) = 70Ω . (Figure 18)

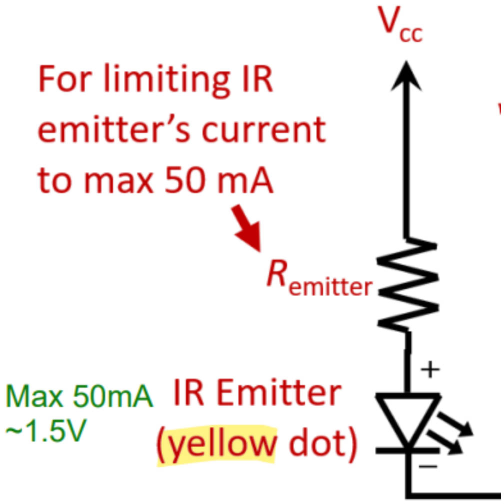


Figure 18 - IR emitter circuit

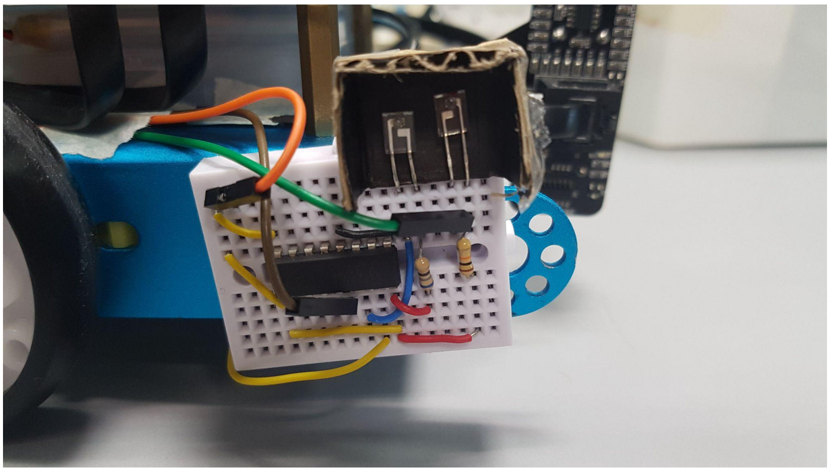


Figure 19 - dome shaped IR cover

Reducing effect of ambient IR to improve robustness

We had 2 ideas to improve the robustness of our IR sensor (ie less affected by ambient IR) via physically blocking most ambient IR with a dome shaped cover (Figure 19) and through the derivation of the IR voltage code. (Figure 20)

```
colour(3);  // on IR emitter;
delay(1000); //for IR detector to stabilize;
int IR_measured = analogRead(IRPin); //IR from emitter & ambient;

colour(0); //off IR emitter;
delay(1000); //for IR detector to stabilize;
int IR_ambient = analogRead(IRPin); //IR from ambient only;

IR_actual = IR_measured - IR_ambient;
Serial.println(IR_actual);
if (IR_actual < IR_threshold) //too close to right wall;
{
    Serial.println("Too close to right wall, shifting left...");
    shift_left();
}
```

Figure 20 - IR sensor code to reduce effects of ambient IR

Colour Sensor

Table III: Recommended Operating Conditions of the HD74LS139P 2-to-4 decoder IC Chip

Item	Symbol	Min	Typ	Max	Unit
Supply voltage	V_{CC}	4.75	5.00	5.25	V
Output current	I_{OH}	—	—	–400	µA
	I_{OL}	—	—	8	mA
Operating temperature	Topr	–20	25	75	°C

Figure 21 - Current output from HD74LS139P 2-to-4 decoder IC Chip

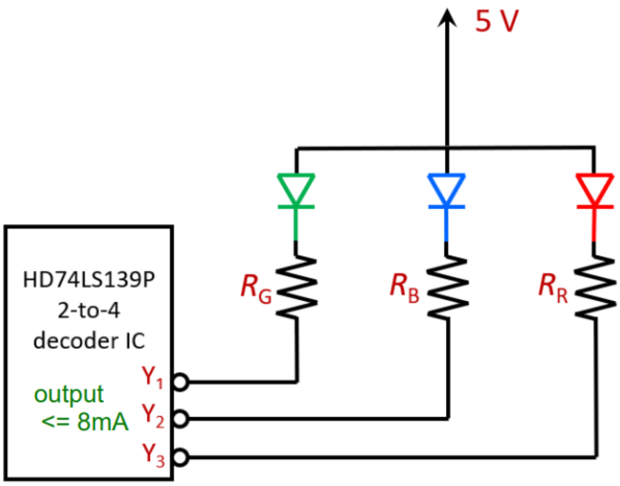


Figure 22 - LED circuit

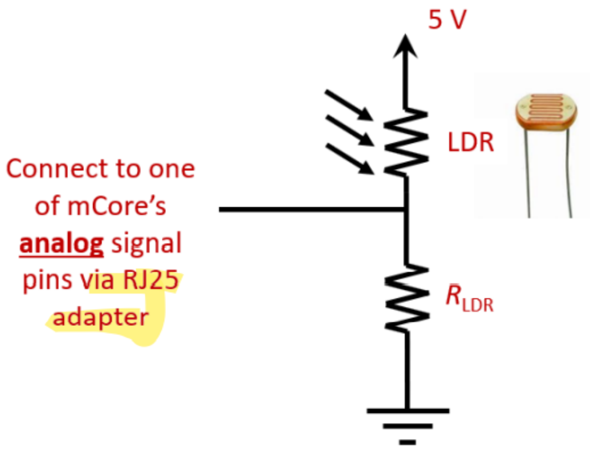


Figure 23 - LDR circuit

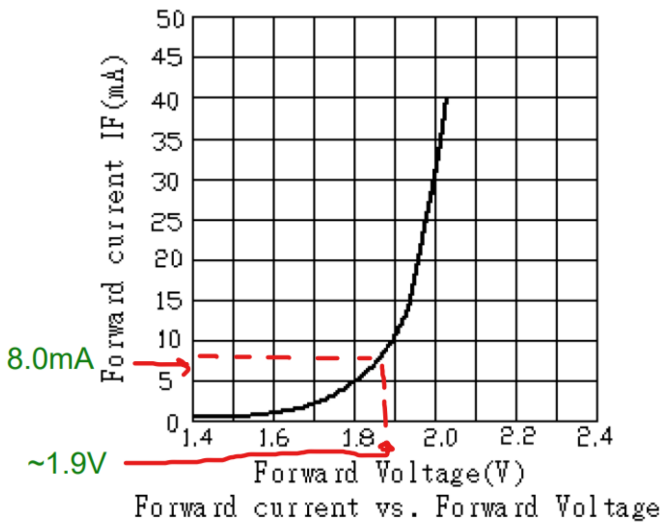


Figure 24 - I_{LED} v V_{LED} (Red LED)

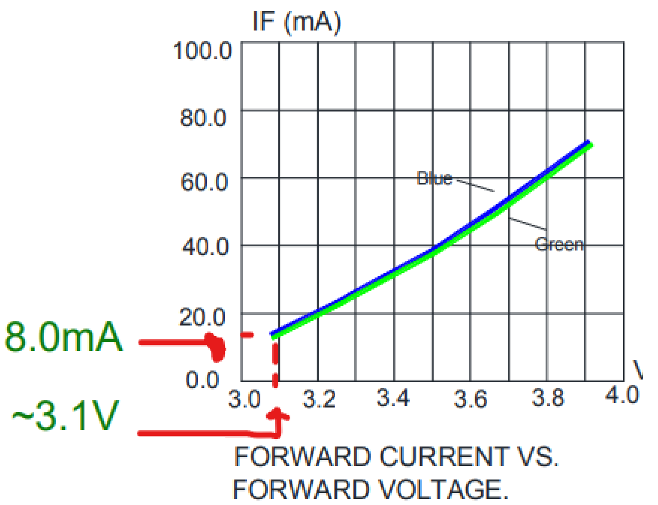


Figure 25 - I_{LED} v V_{LED} (Blue, Green LED)

Colour Sensor Responsiveness

To maximize the responsiveness of our LDR, we initially maximized the intensity of the LED by maximizing I_{LED}.From above, the current output from the HD74LS139P 2-to-4 decoder IC Chip is 8mA (Figure 19), while the Red LED voltage at 8.0mA is ~1.9V (Figure 22). From $R_{max} = V_{max} / I_{max}$, resistor value to achieve the maximum voltage of ~1.9V at 8.0mA is (5.0 - 1.9)V / (8mA) = 387Ω (Figure 22). For blue and green LED (Figure 25), based on same formula, ideal resistor value to use would be (5.0 - 3.1)V / (8mA) = 238Ω.

However, after taking readings from our colour sensor, the RGB values for blue and green were significantly higher than that of red, regardless of the unknown colour. Hence, we had to reduce the current for the blue and green LED by increasing the resistor values. We arrived at 985Ω and 382Ω for blue and green LED respectively, whereby the RGB values for all 3 colours were comparable to each other.

Improving robustness of colour sensor

Additionally, we also wanted to reduce the effect of ambient light as it significantly affected our colour sensor's colour detection by inflating/deflating the RGB values. We achieved this by covering the entire base with black paper, coupled with an additional chimney-like opening with an extra protrusion towards the ground (Figure 27) to further reduce the ambient light coming from the sides.

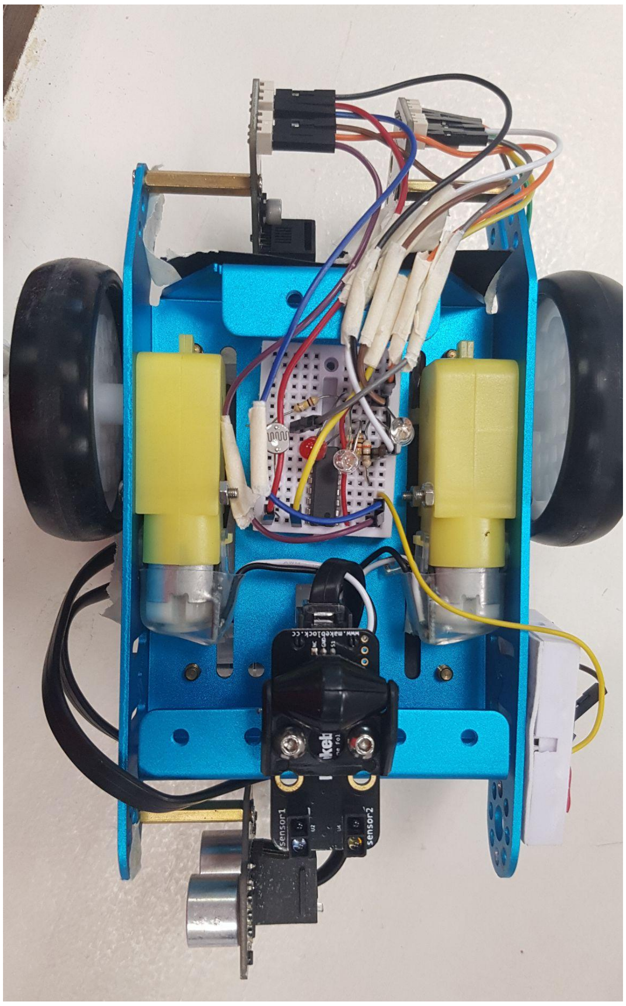


Figure 26 - Before chimney

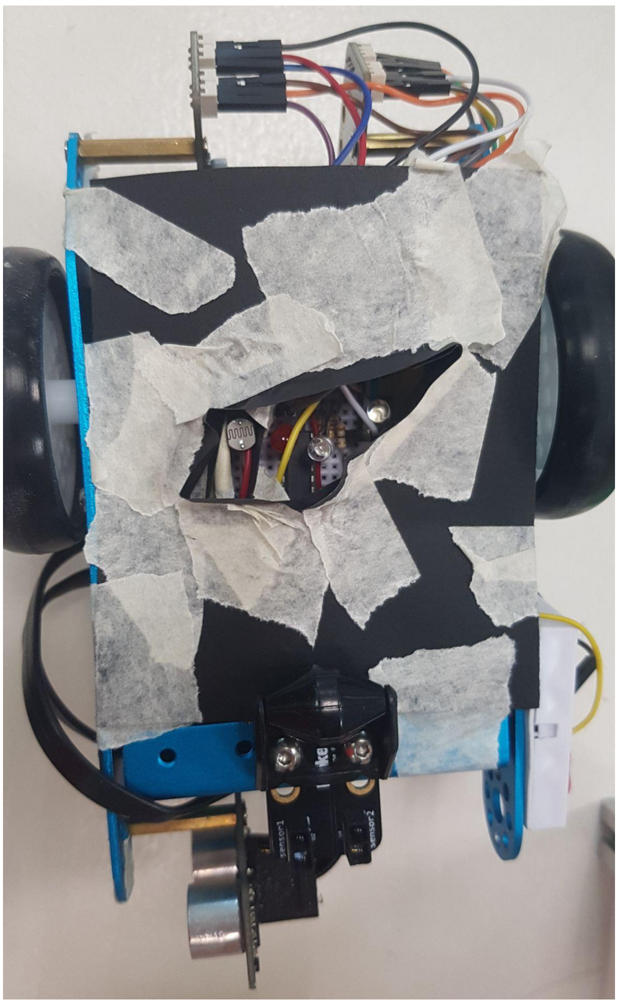


Figure 27 - After chimney

Challenges and Solutions

Challenges	Solutions
Frequent colliding into the front and side walls when the robot made two consecutive right and left turns at the light blue and purple colour challenges respectively.	We changed the relevant code functions to alter the robot's turning angle and distance travelled between 2 turns, arriving at the optimal values after multiple trial runs.
Occasionally, the angle of the robot after a colour challenge, especially after 2 consecutive turns, would hinder the robot's ability to detect a side wall, either causing a collision or rendering the robot off course when there is no wall.	We increased the threshold distance of our IR sensor to be able to detect the side walls no matter the angle of the robot after a colour challenge.
Initial calibration of the colour sensor did not yield optimal results - sensors were not able to detect the correct colour when ambient light intensity was changed slightly.	Taking multiple RGB values in different ambient light conditions, we engineered suitable RGB boundary values for each colour. Coupled with the other measures (Figure 22-25), this ensured our colour sensor was able to detect the correct colour in different environments.
Loose wire connections hindered voltage readings in our IR and colour sensor circuits, which was hard to diagnose given the sporadic nature of the issue.	Changed wires to solid wires wherever possible, which were able to be attached more securely and less likely to break inside. Trimmed wires to ensure wires remain secure and general tidiness of circuit connections for easy debugging/ spotting of loose wires.
When connected to the HD74LS139P 2-to-4 decoder IC Chip, the LEDs did not light up according to the logic behaviour of the chip, although the "Enable" (G) and "Select Inputs" (A/B) pins were at the correct setting.	Using our multimeter, we tested the voltages at each node to verify our assumptions. For example, the multimeter reading of 0V at the Enable pin confirms a logic LOW. Eventually, we determined that our current 2-to-4 decoder set was faulty after confirming the correct logic at each node. Changing to the second set of 2-to-4 decoder solved this problem.
The ultrasonic sensor was initially installed at the outer left edge of our robot. However, the operating range of 3 to 400 cm meant the robot was unable to detect the wall if it were within 3 cm of the wall.	To ensure our robot was still able to detect the wall within 3 cm of its body, we attached the ultrasonic sensor inwards to its front centre (Figure 8) to address this limitation.

Work Division

Team Member	Roles And Responsibilities
Nikhil Babu	- Overall IC for the mBot's code for black line detection and waypoint challenge - Involved in the mBot's code, including the movement code, black line detection, and waypoint challenge - Calibrated the movement code in line with the obstacle course - Setting up and installing the black line detector and colour sensor
Nicholas Tan Yun Yu	- Overall IC for the mBot's code for movement code and waypoint challenge - Involved in the mBot's code, including the black line detection and waypoint challenge - Calibrated the movement code in line with the obstacle course - Setting up and experimenting with various resistors for the IR sensor
Ng Yan Hui	- Setting up and installing the black line detector and colour sensor - Troubleshooting and experimenting various resistors for the colour sensor - Connecting the various sensors and circuits together - Involved in the waypoint challenge code and IR calculation
Ng Xin Zhe Sean	- Setting up and experimenting with various resistors for the IR sensor and colour sensors - Troubleshooting and experimenting various resistors for the colour sensor - Connecting the various sensors and circuits together - Contributed to the mBot's code with feedback