Checkpoint #4 Report

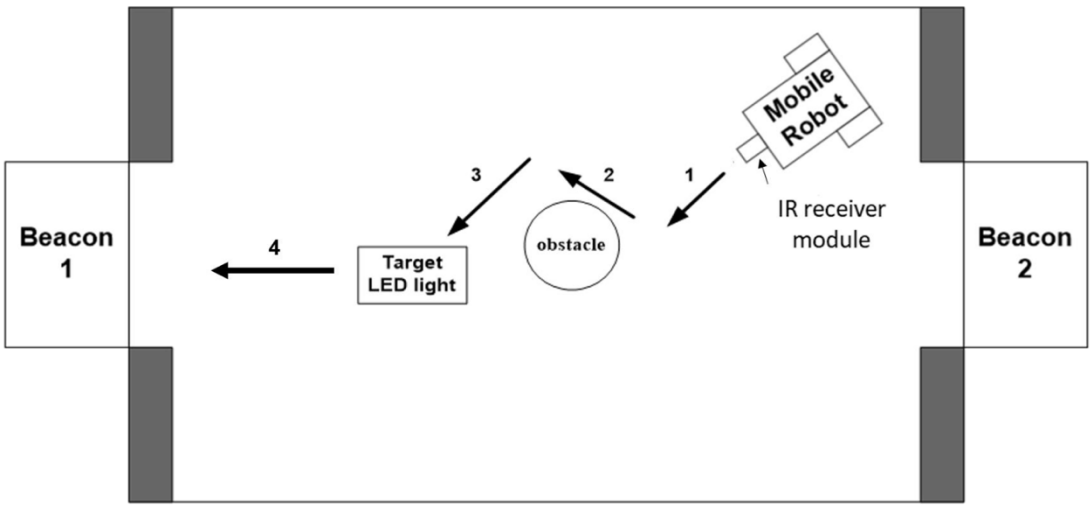
[EECN30169] Mobile Robot 2022

Student ID: 311605004 **Name:** 劉子齊 **Date:** 2022.12.08

1. Introduction:

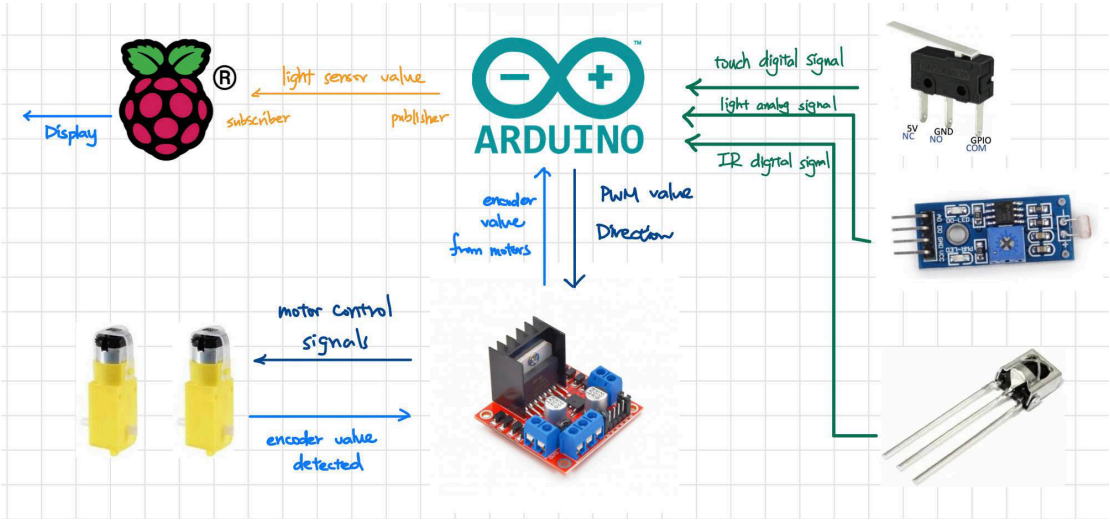
For checkpoint 4, our goal is to implement our mobile robot and make it able to move in the arena freely. Besides, to make our mobile robot able to avoid obstacles and detect the target hockey ball, we have to implement the feature of collision detection and light probing. In addition, in this checkpoint, we also need to make our hockey robot able to detect the 2 beacons which indicate the 2 goals of the arena.

The following figure is the hockey arena for this checkpoint. As we can see in the following figure, our goal is to make our robot to avoid the obstacle in the center of the arena. After avoided from the arena, the next task for our robot is to find the target hockey light by the light sensor. When detected the target hockey, our robot will have to approach the target hockey and catch it. Afterwards, our mobile robot has to approach and catch the target hockey, then we will start to find the hockey goal through detecting the beacon. Finally, as we find the goal, we will approach the goal and bring the hockey ball into the goal.



2. Description of Design:

Below is the workflow of my implementation to the communication between the Raspberry Pie, the Arduino, the L298N motor control module, the 2 DC motors, the 3 touch sensors, the light sensor, and the new added IR receiver.



To make our mobile able to detect the return signal of the touch sensors and the light sensor, we set the input pins as the following figure. We set the touch sensors as digital read, since we only need to know whether the touch sensors were pressed or not. However, we set the light sensor as analog read, since it will be a lot easier for us to detect the target hockey correctly by the exact light strength value returned from the light sensor.

```
#define left_touch_pin 3
#define right_touch_pin 2
#define under_touch_pin 4
#define light_sensor_pin A0 //A0
```

```
pinMode(left_touch_pin, INPUT);
pinMode(right_touch_pin, INPUT);
pinMode(under_touch_pin, INPUT);
pinMode(light_sensor_pin, INPUT);
```

```
int touch_left = digitalRead(left_touch_pin);
int touch_right = digitalRead(right_touch_pin);
int touch_under = digitalRead(under_touch_pin);
int light_sensor = analogRead(light_sensor_pin);
```

With all the return values from the sensors, we now move onto the motion control. The implementation of the motion control is shown in the following figures. Basically, I separated the whole process into 8 situations, which started from 0 as the initial situation.

The first situation is situation 0, which the mobile robot has only 1 task, which is to go straight until it runs into the wall. As the 2 touch sensors attached at the front of the mobile robot detect the wall, the

robot will enter situation 1, which the robot will go backwards for 600 milliseconds. After moving backwards, the robot will enter situation 2, which it will start to turn left. During situation 2, when the light sensor attached at the bottom of the robot detects the target hockey, which is obviously brighter than the surroundings, the robot will stop turning and enter situation 3.

As the robot enters situation 3, it means the robot had found the target hockey. At this moment, there is only one mission for the mobile robot, which is rush towards the target hockey and catch it. If the robot catches the target hocket, which will determine on the touch sensor attached at the bottom of the robot, the robot will enter situation 4.

In situation 4, the robot will start turning counter-clockwise to search for the beacon, which is the goal. The implementation of the goal searching process is shown in the following figure. Basically, what we did is to calculate the ratio of the low signals to the total count of the signals, determine which beacon did the signal belong through the ratio we obtained, and return the result goal.

```
153  int find_goal(){
154       float rate = 0.0;
155       float count_low = 0.0;
156       float count_high = 0.0;
157
158       for(int i = 0; i < 120; i++){
159            if(digitalRead(IR_pin) == 1){
160                 count_high++;
161                 delay(1);
162            }
163
164            if(digitalRead(IR_pin) == 0){
165                 count_low++;
166                 delay(1);
167            }
168       }
```

```
170       rate = count_low / (count_high + count_low);
171
172       if(rate > 0.25 && rate < 0.37){
173            which_IR = 600;
174            return 600;
175       }
176       else if(rate > 0.17 && rate < 0.22){
177            which_IR = 1500;
178            return 1500;
179       }
180       else{
181            return 0;
182       }
183  }
```

As our hockey robot find the goal, at this moment, there is only one mission for the mobile robot, which is rush towards the goal and goal it, and this indicates the accomplishment of the task.

For the motion control of the mobile robot, I applied the achievements of the previous checkpoint. The motion signal is send form the Arduino to the L298N motor control module according to different cases, which I mainly made use of the 3 cases defined below.

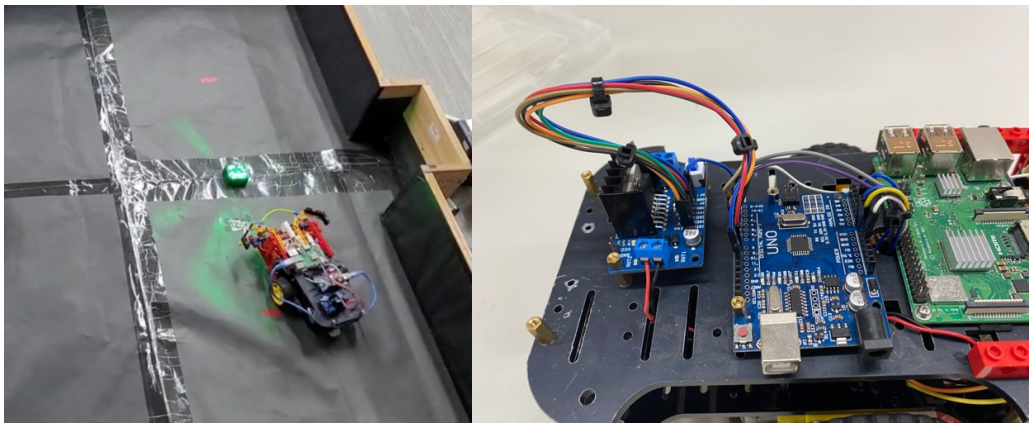
```
void Forward(){
    digitalWrite(L298N_IN1, HIGH);
    digitalWrite(L298N_IN2, LOW);
    digitalWrite(L298N_IN3, HIGH);
    digitalWrite(L298N_IN4, LOW);
    analogWrite(E_left, val_output_L);
    analogWrite(E_right, val_output_R);
}
```

```
void Backward(){
    digitalWrite(L298N_IN1, LOW);
    digitalWrite(L298N_IN2, HIGH);
    digitalWrite(L298N_IN3, LOW);
    digitalWrite(L298N_IN4, HIGH);
    analogWrite(E_left, -val_output_L);
    analogWrite(E_right, -val_output_R);
}
```

```
void Right(){
    digitalWrite(L298N_IN1, LOW);
    digitalWrite(L298N_IN2, HIGH);
    digitalWrite(L298N_IN3, HIGH);
    digitalWrite(L298N_IN4, LOW);
    analogWrite(E_left, val_output_L);
    analogWrite(E_right, -val_output_R);
}
```

Lastly, during the implementation, we found we would need to know the exact value which the light sensor and the IR receiver are receiving to adjust the threshold value we applied to determine whether we are facing the target hockey or not. Hence, we applied the subscriber and the publisher we learned in the previous checkpoint to send the light value detected from the Arduino to the Raspberry Pie.

3. Result



The figures above of the mobile robot we designed, and how the arena and the target hockey look like. This time we also spent plenty of time arranging the wires of our mobile robot, which can be found in the figures above.

4. Discussion

The obstacle we met in this checkpoint is the IR signal, which published by the beacon, is kind of unstable. Especially when detecting the Goal-600, our mobile robot always slightly offset to the right of the Goal-600. As a solution to this problem, when the goal our robot detects is the Goal-600, before rushing straight to the goal, we make our robot turn slightly left first, and this solved the problem.