Checkpoint # 4 Report

[ICN5406] Mobile Robot 2021

Student ID: 310515010 **Name:** 洪子茵 **Date:** **11/26**

1. Purpose:

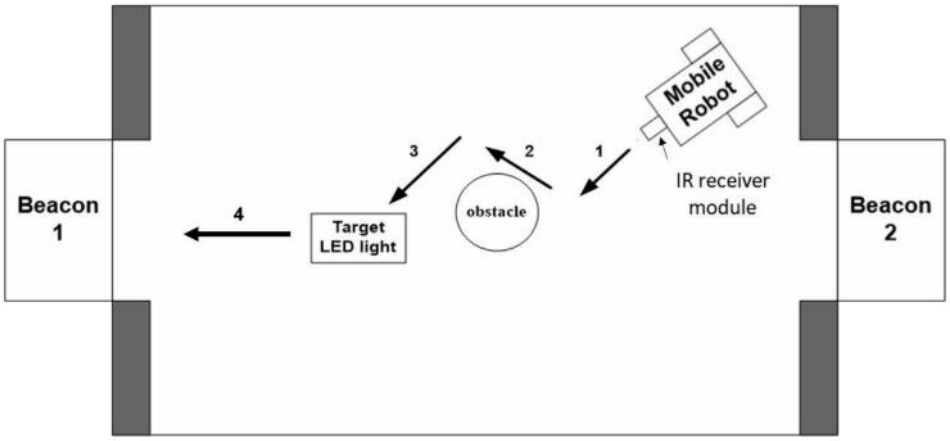
The purpose of this checkpoint has two goals:

1. making sure that your robot can detect a beacon signal and move towards it.
2. combine all the function together for robot hockey contest.

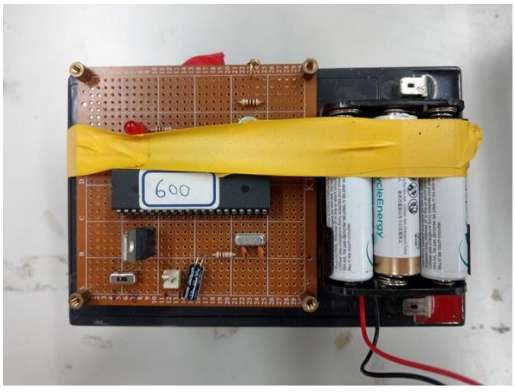
2. Description of Design:

For this assignment, two infrared diodes will be set up at opposite ends of an arena.

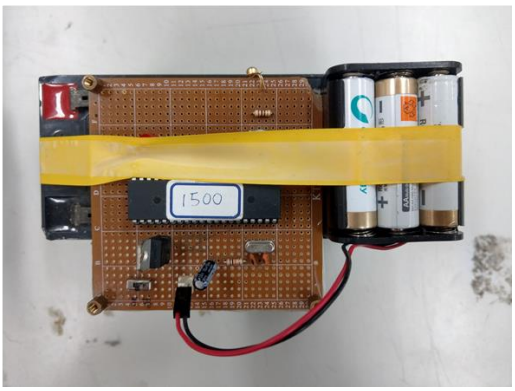
Each diode will be emitting light modulated at 38 KHz, but their pulse width are different when received by IR receiver module.



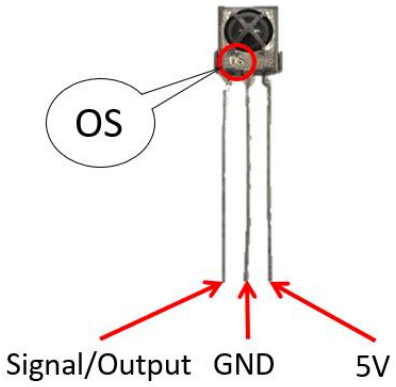
Beacon1
600



Beacon2
1500

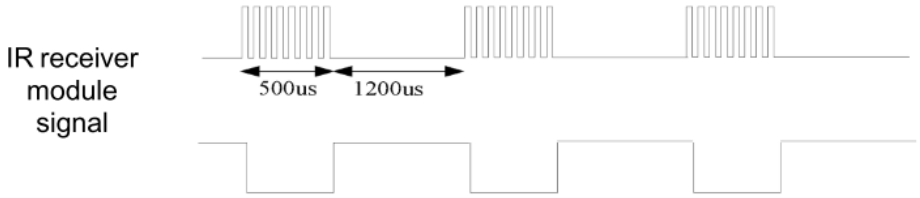


The goal of this checkpoint is to find beacons(Beacon-1 600 and Beacon-2 1500) using IR receiver.

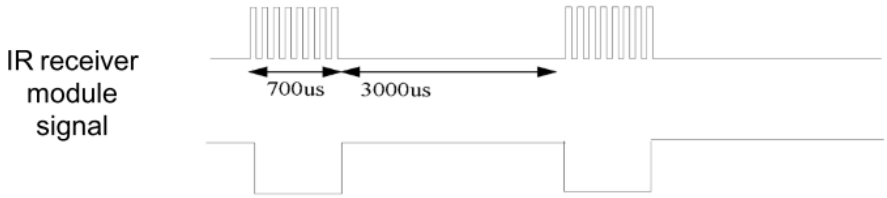


The IR receiver(PIC-428LM IR receiver) will get 0 when it receive IR signal, and 1 for opposite, so **the method to know whether it's door 1500 or door 600 is to calculate the ratio of "0"**, which demonstrate as below:
(but remind that when at implementation, the ratio need to be experimented to have the most accurate value.)

1. Beacon-1 600(50%)



2. Beacon-2 1500(50%)



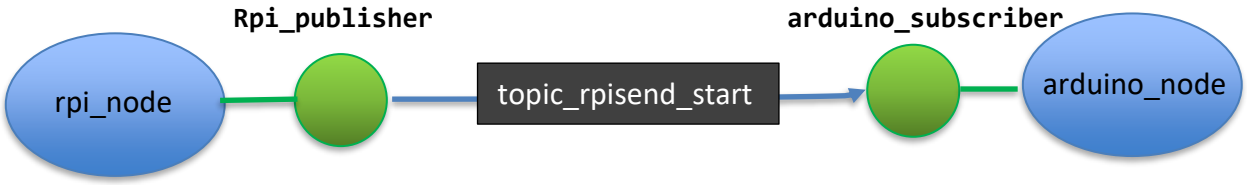
Receive ir data for a period of time and calculate the pulse proportion.

$$\textbf{Ratio} = \frac{\boldsymbol{number\ of\ 0}}{\boldsymbol{total\ data(include\ 0\ and\ 1)}}$$

If your goal door is 1500, the ratio is between 0.17 and 0.22.

If your goal door is 600, the ratio is between 0.27 and 0.32.

The Rpi and Arduino communication is like last checkpoint:



Code in rpi

```cpp
#include "ros/ros.h"
#include "std_msgs/Int32.h"
#include <iostream>
#include <stdio.h>
#include <wiringPi.h>

int flag = 0;

int main(int argc, char **argv)
{

    ros::init(argc, argv, "ch3_node");

    ros::NodeHandle node_obj;

    ros::AsyncSpinner spinner(0);
    spinner.start();

    ros::Publisher rpi_publisher_start = node_obj.advertise<std_msgs::Int32>("topic_rpisend_start", 10);


    while (ros::ok()) {
        std_msgs::Int32 start;

        std::cout << "when start give 1 :";
        std::cin >> start.data;
        flag = 1;

         if (flag == 1){
             rpi_publisher_start.publish(start);
             flag == 0;
         }

         ros::spinOnce();

    }

    return 0;
}
```

Code in arduino

It can divide into three part:

1. the main void loop part
2. avoid obstacle part
3. searching light part
4. **searching door part (the only part different from last checkpoint)**

1. the main void loop part

when Arduino get the start signal from raspberry pi, it will start looping **avoid obstacle** and **searching_light part** until the automobile car get the light ball (which means that the third touch sensor at the below will be pushed successfully by light ball.)
Once it get light ball, it will start finding door using **searching_door()** function, if the robot find door in that function time(use IR receiver) Than it will return 1 for result, otherwise it will move forward little bit, and searching door again.

```
74  void loop()
75  {
76    nh.spinOnce();
77
78    int starttime, endtime;
79    int result;
80    char INFO[20];
81
82    touch3_val = digitalRead(PIN_TOUCH_3);
83
84    if (start_val == 1){
85      if (touch3_val == 1){ //touch3 not touch
86
87        starttime = millis();
88        endtime = starttime;
89        while ((endtime - starttime) <= 5500)
90        {
91          touch1_val = digitalRead(PIN_TOUCH_1);
92          touch2_val = digitalRead(PIN_TOUCH_2);
93          avoid_obstacle();
94
95          touch3_val = digitalRead(PIN_TOUCH_3);
96          if (touch3_val == 0){
97            break;
98          }
99
100         endtime = millis();
101       }
```

```
      nh.spinOnce();
      if (touch3_val == 1){
        searching_light();
      }
    }
    else{                  //touch3 touch, get the light ball, Goal door Seeking
      delay(100);
      nh.spinOnce();
      nh.loginfo("start searching door");
      result = searching_door();        // spin one round & move forward

      nh.spinOnce();
      if (result == 1){
        nh.loginfo("find door!!!!!!");
        drive(0, 0, 0, 0);
        delay(500);


        starttime = millis();
        endtime = starttime;
        while ((endtime - starttime) <=3000)
        {
          drive(1, 105, 1, 100);     //forward

          endtime = millis();
        }
      }
      else{
        starttime = millis();
        endtime = starttime;
        while ((endtime - starttime) <=1300)
        {
          avoid_obstacle_door();                  //move forward

          nh.spinOnce();
          endtime = millis();
        }
      }
    }
  }
  nh.spinOnce();
  delay(10);
}
```

2. avoid obstacle part

in this part I detect whether the car touch the obstacle, there are four possible situation, if each of one happens, the car will have corresponding action.(ex: if it doesn't touch anything, it will keep moving forward.)

```
151   void avoid_obstacle(){
152     if (touch1_val == 0){  //left touch
153       nh.loginfo("left touch");
154       drive(-1, 80, -1, 80);     //backward
155       delay(1000);
156       drive(1, 90, 1, 0);       //right
157       delay(2000);
158       drive(1, 80, 1, 80);     //forward
159       delay(3000);
160       touch1_val == 1;
161       nh.spinOnce();
162     }
163     else if (touch2_val == 0){   //right touch
164       nh.loginfo("right touch");
165       drive(-1, 80, -1, 80);     //backward
166       delay(1000);
167       drive(1, 0, 1, 80);       //left
168       delay(2000);
169       drive(1, 80, 1, 80);     //forward
170       delay(3000);
171       touch2_val == 1;
172       nh.spinOnce();
173     }
174     else if (touch1_val == 0 && touch2_val == 0){   //both touch
175       nh.loginfo("both touch");
176       drive(-1, 80, -1, 80);     //backward
177       delay(1000);
178       drive(1, 90, 1, 0);       //right
179       delay(2000);
180       drive(1, 80, 1, 80);     //forward
181       delay(3000);
182
183       touch1_val == 1;
184       touch2_val == 1;
185       nh.spinOnce();
186     }
187     else{
188         nh.loginfo("avoid obstacle forward");
189         drive(1, 80, 1, 80);    //forward
190         delay(2000);
191         nh.spinOnce();
192     }
193   }
```

3. searching light part

In this part I used this algorithm to find light :

Self-Spin twice time, the first spin is to get the min light value(which should be the brightest place.), then the second spin will compare the sensed light value to the min light value, and stop self-spinning when **abs(light value in second run- min light value) < 15, which would be the condition that it find light! ,**then move forward to the light.

```
void searching_light4(){
  int min_light_level=1023;
  int light_level, light_level2;
  char INFO[20], INFO2[20];
  int starttime,  endtime;

  // spin first time
  starttime = millis();
  endtime = starttime;
  While ((endtime - starttime) <=8500) // do this loop for up to 8500mS
  {
    drive(1, 0, 1, 80);

    light_level = analogRead(PIN_LIGHT);
    sprintf(INFO, "light_level : %d", light_level);
    nh.loginfo(INFO);

    if  (light_level < min_light_level){
        min_light_level = light_level;
    }
    endtime = millis();
  }

  sprintf(INFO2, "min_light_level : %d", min_light_level);

  // spin second time
  nh.loginfo("Start second round");
  do{
    drive(1, 0, 1, 80);
    light_level2 = analogRead(PIN_LIGHT);
  } while (abs(light_level2 - min_light_level) > 15);
  nh.loginfo("Find light");  //Find light

  drive(0, 0, 0, 0);
  delay(500);
  drive(1, 0, -1, 80);  //迴轉一點
  delay(400);
  nh.spinOnce();
  drive(1, 95, 1, 80);  //向光前進
  delay(2500);
}
```

3. searching door part

Add two main function in this part :
calculate_ratio(), searching_door()

searching_door is the main function here, I use the algorithm like searching light part, which is let robot spin while at the time calculate the door number, using calculate_ratio() function, if the door number is 600 or 1500, than the robot find goal door! And it will return the result = 1, and return back to void loop().

```
397  int searching_door(){
398    door_num = calculate_ratio();
399    if (door_num == 600 || door_num == 1500){
400      drive(0, 0, 0, 0);
401      return 1;
402    }
403
404    int starttime, endtime;
405    starttime = millis();
406    endtime = starttime;
407    while ((endtime - starttime) <=4850)
408    {
409      drive(1, 0, 1, 125);                    //spin one round
410      door_num = calculate_ratio();
411      if (door_num == 600 || door_num == 1500){
412        drive(0, 0, 0, 0);
413        return 1;
414      }
415      nh.spinOnce();
416      endtime = millis();
417    }
418  }
```

Calculate_ratio() use to calculate the ratio of “0” received by IR receiver, and depend on the ratio being calculated, to know which door robot encounter.

```
int calculate_ratio(){
  int zero_count, one_count, total_count;
  float ratio;
  int IR_value;
  char INFO[20], INFO1[20];
  int door_number;

  for(int i = 0 ; i<200; i++){
      IR_value = digitalRead(IR_PIN);

      if (IR_value == 1){
          one_count++;
      }
      else{
          zero_count++;
      }
      delay(1);
   }

  total_count = zero_count + one_count;
  ratio = (float)zero_count/(float)total_count;
  sprintf(INFO1, "zero_count : %d, one_count : %d", zero_count , one_count);
  nh.loginfo(INFO1);

  sprintf(INFO, "ratio : %.2f", ratio);
  nh.loginfo(INFO);
  if (ratio > 0.15 && ratio < 0.3){
      nh.loginfo("goal door 1500");
      door_number = 1500;
  }
  else if (ratio > 0.35 && ratio < 0.5){
    nh.loginfo("goal door 600");
    door_number = 600;
  }

  one_count = 0;
  zero_count = 0;

  return door_number;
}
```

3. Discussion

*Some issue I occur during this checkpoint

The sensor value is the most difficult part in this checkpoint, and due to different environment condition or interference, it is so hard to reproduce the same behavior each time, and not robust at all, especially for photo resistor sensor.

Besides, my motor also have problem when demonstrate the rest door, I have no idea whyI give the motors same speed every time , but it turn out performing different behaviors, or being too fast/too slow (still don’t know it’s the battery issue or because I didn’t use PID to control motors…)