Checkpoint # 3   Report

[ICN5406] Mobile Robot 2021

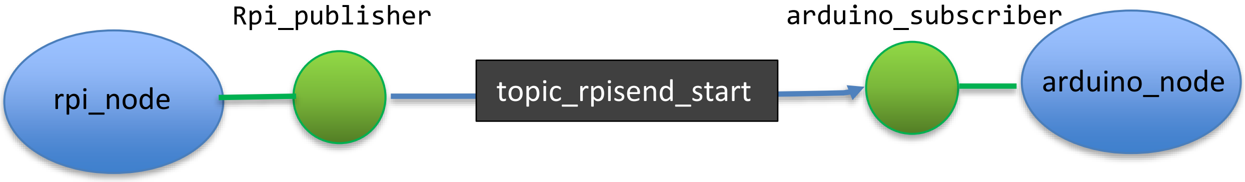
Student ID: 310515010 **Name:** 洪子茵 **Date:** **11/11**

1. Purpose:

make sure you can control your robot to move in the arena. The mobile robot needs to detect an obstacle in front of it and take action to avoid the obstacle in order to continue its motion. (use two touch sensors)

Finally, your robot can find the assigned target. In this checkpoint, the target is a ring of LED lights.(use one touch sensor, and a photo resistor sensor to detect light value)

2. Description of Design:

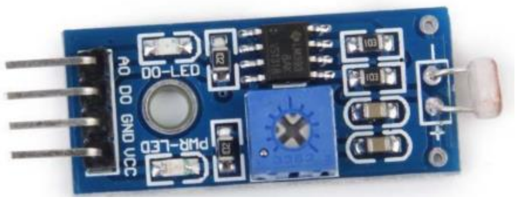


My design is simple on this checkpoint, first of all , I create a topic between raspberry pi and Arduino, so rpi can send a "start" signal to tell Arduino it can start running.

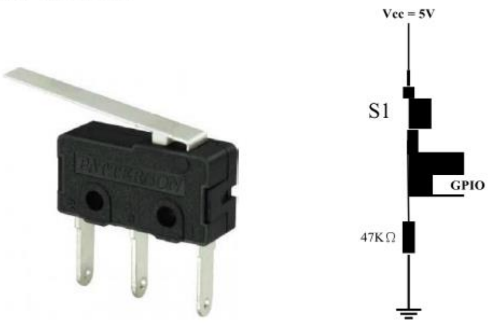
Then I do all the other sensing staff in Arduino, including **three** touch **sensors**(which I use three digital pin), and **one photo resistor sensor**(using analog pin), to get the current situation of the mobile car(the sensors used is like the picture below.)

So for example, when the car touch the obstacle, the sensor will get "0", otherwise, it will get "1", and the car will sense the environmental light value using photo resistor sensor, the values is in the range of [0, 1023], the brighter, the lower value it will get.

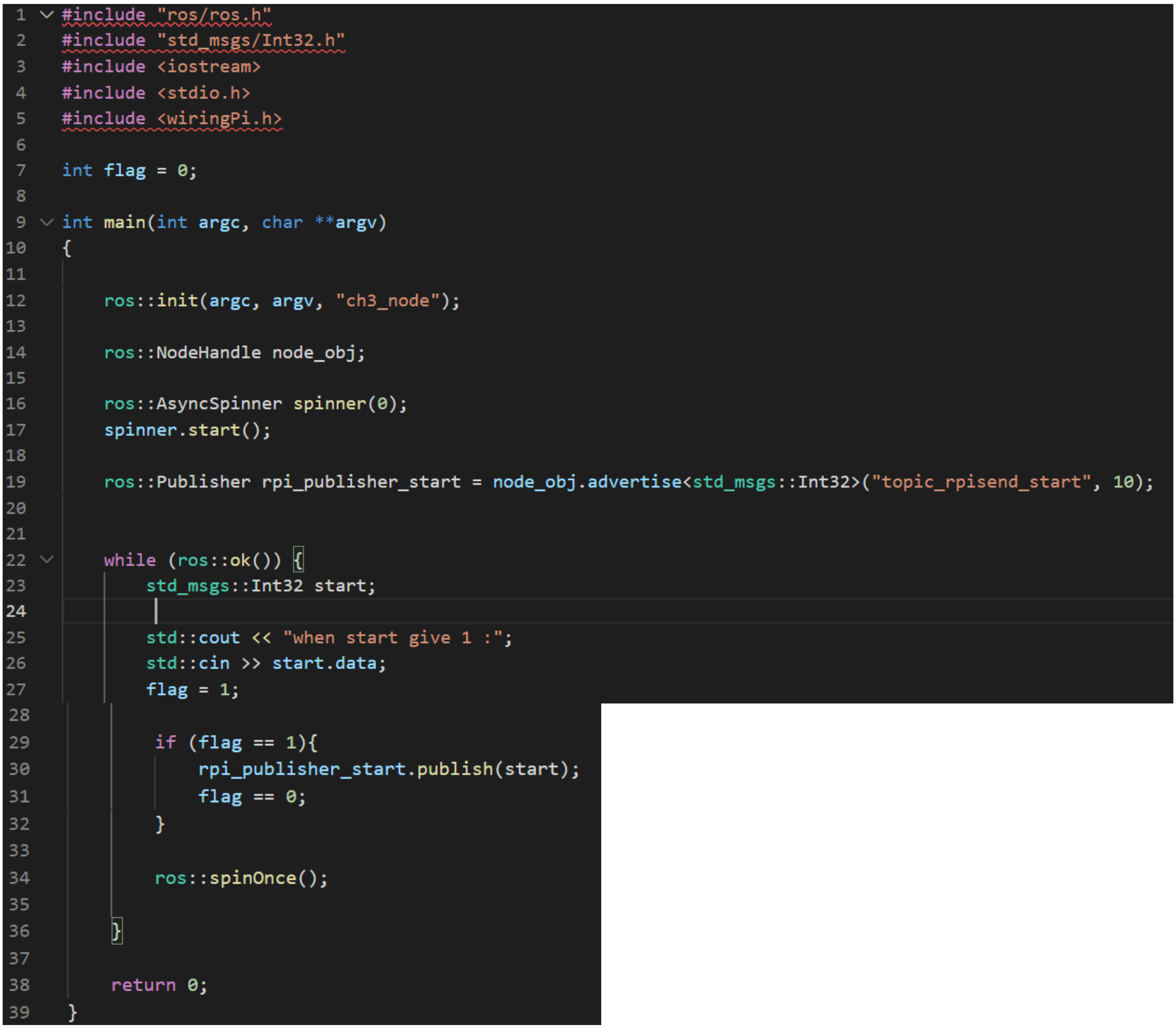
- **Photo resistor sensor:**



- **Touch sensor:**



Code in rpi



```
#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;
}
```

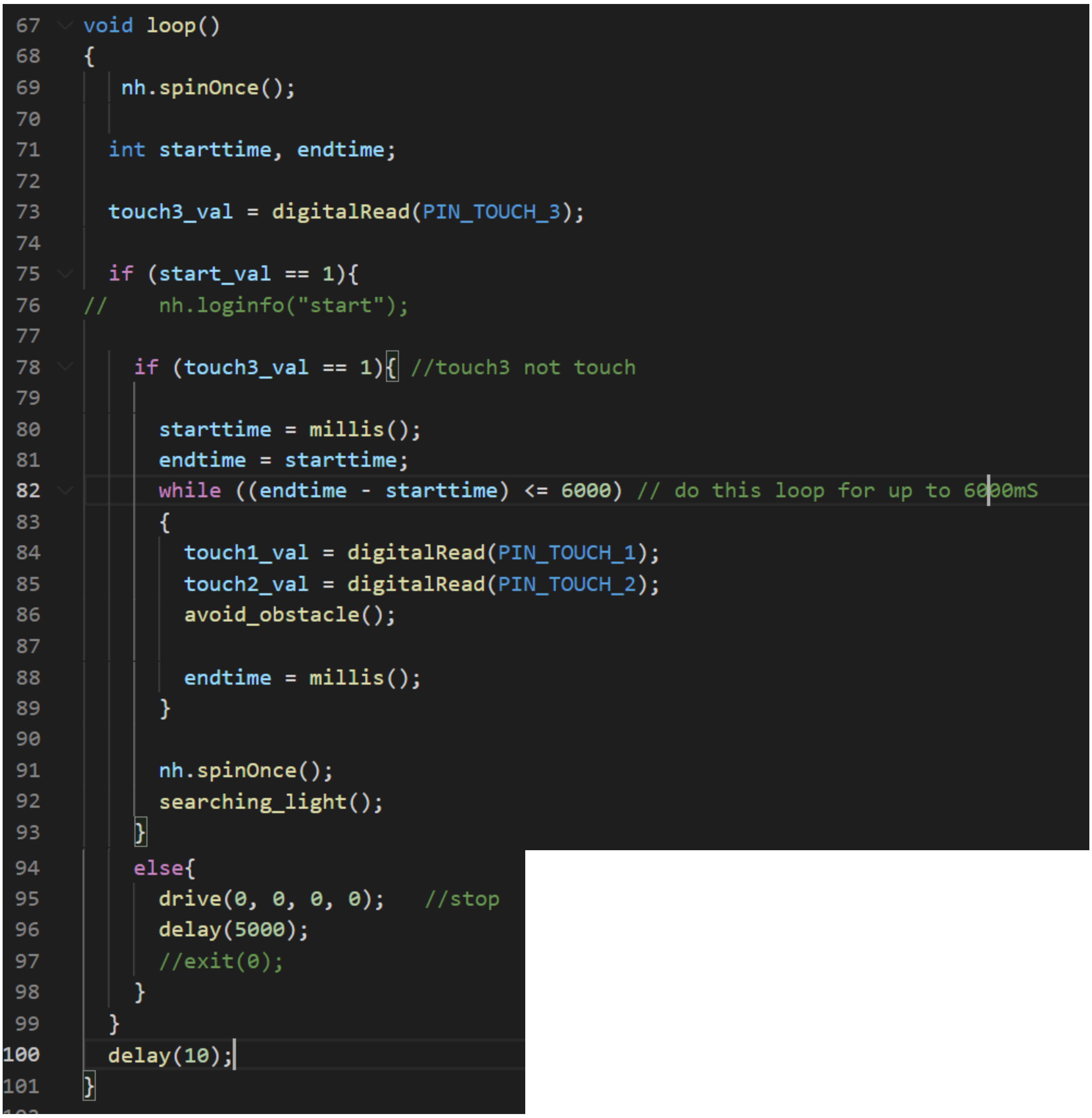
Code in arduino

It can divide into three part:

1. the main void loop part
2. avoid obstacle part
3. searching light part

1. the main void loop part

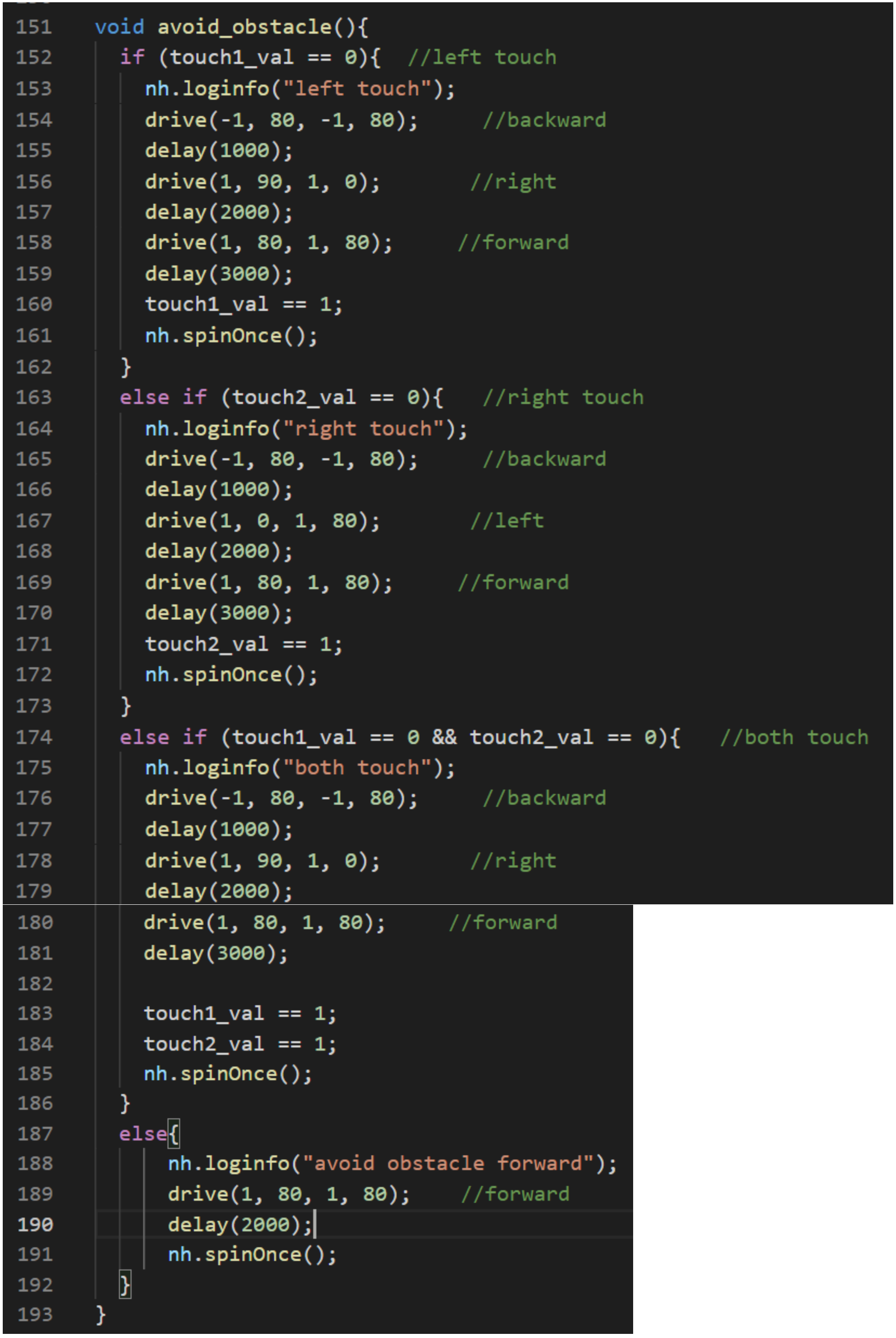
when Arduino get the start signal from raspberry pi, it will start looping **avoid obstacle** and **searching_light part.**
The avoid obstacle will take about 6 seconds to finish, and then Arduino will start searching light.



```
void loop()
{
   nh.spinOnce();

  int starttime, endtime;

  touch3_val = digitalRead(PIN_TOUCH_3);

  if (start_val == 1){
//    nh.loginfo("start");

    if (touch3_val == 1){ //touch3 not touch

      starttime = millis();
      endtime = starttime;
      while ((endtime - starttime) <= 6000) // do this loop for up to 6000mS
      {
        touch1_val = digitalRead(PIN_TOUCH_1);
        touch2_val = digitalRead(PIN_TOUCH_2);
        avoid_obstacle();

        endtime = millis();
      }

      nh.spinOnce();
      searching_light();
    }
    else{
      drive(0, 0, 0, 0);   //stop
      delay(5000);
      //exit(0);
    }
  }
  delay(10);
}
```

2. avoid obstacle part

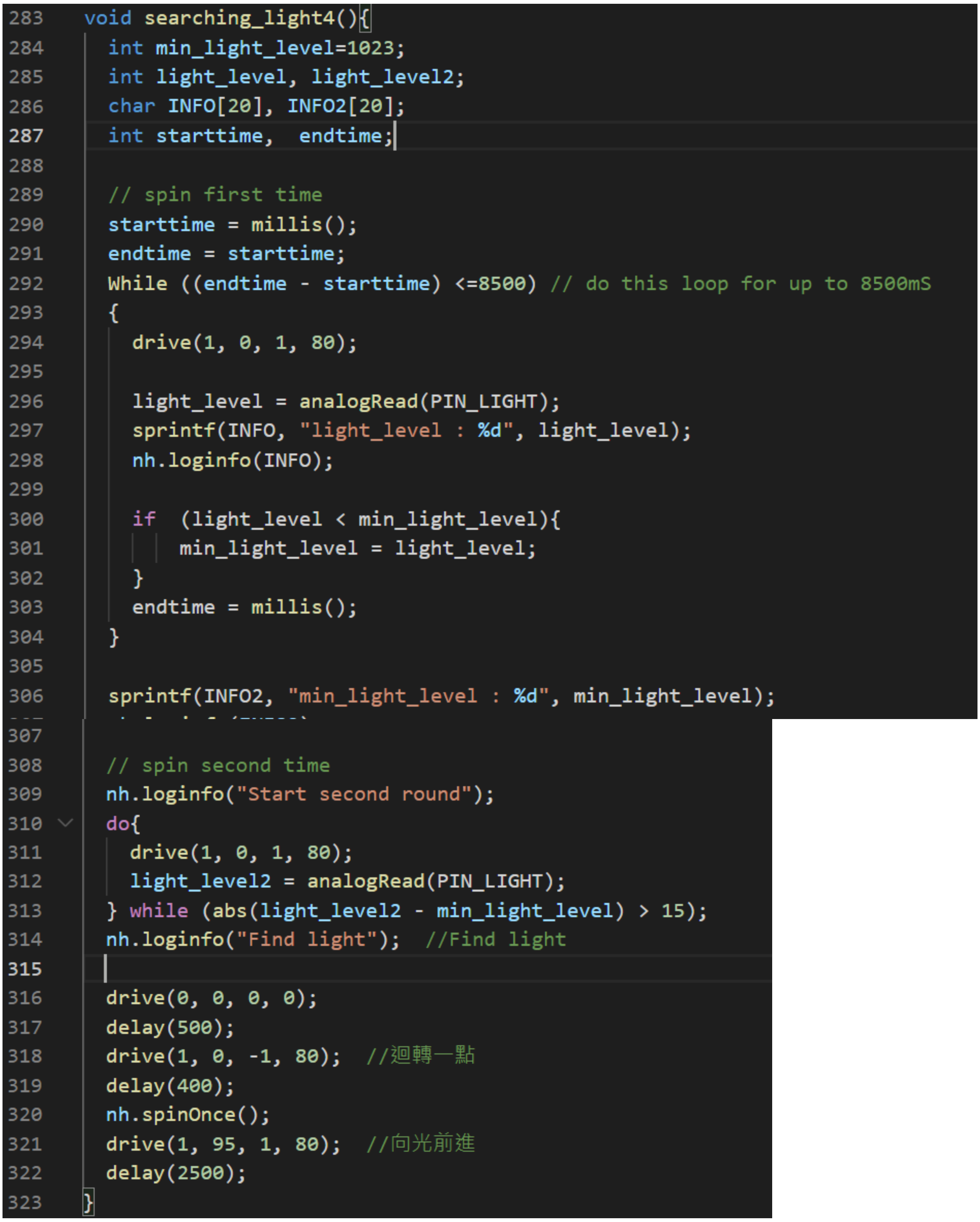
in this part I detect whether the car touch the obstacle, there are four possible situation, if each of one happens, the car will have corresponding action.(like if it doesn’t touch anything like normal, it will keep forward.)



```
void avoid_obstacle(){
  if (touch1_val == 0){  //left touch
    nh.loginfo("left touch");
    drive(-1, 80, -1, 80);     //backward
    delay(1000);
    drive(1, 90, 1, 0);       //right
    delay(2000);
    drive(1, 80, 1, 80);     //forward
    delay(3000);
    touch1_val == 1;
    nh.spinOnce();
  }
  else if (touch2_val == 0){   //right touch
    nh.loginfo("right touch");
    drive(-1, 80, -1, 80);     //backward
    delay(1000);
    drive(1, 0, 1, 80);       //left
    delay(2000);
    drive(1, 80, 1, 80);     //forward
    delay(3000);
    touch2_val == 1;
    nh.spinOnce();
  }
  else if (touch1_val == 0 && touch2_val == 0){   //both touch
    nh.loginfo("both touch");
    drive(-1, 80, -1, 80);     //backward
    delay(1000);
    drive(1, 90, 1, 0);       //right
    delay(2000);
    drive(1, 80, 1, 80);     //forward
    delay(3000);

    touch1_val == 1;
    touch2_val == 1;
    nh.spinOnce();
  }
  else{
      nh.loginfo("avoid obstacle forward");
      drive(1, 80, 1, 80);     //forward
      delay(2000);
      nh.spinOnce();
  }
}
```

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. Discussion

***Some issue I occur during this checkpoint**

1. The environment light will affect the photo resistor sensor to sense the LED light ring.
2. I have tried multiple find light algorithms(which will mention in the reference.), but after testing, none of it will work, or work perfectly as my current one. ☹
3. I was first using digital input pin to get photo resistor sensor's sensing value, but I found it will be more robust if I use analog value, so I switch at the end.
4. Can't use exit(0) to get out off void loop in Arduino.

4. Reference

Introduction to WiringPi (for Raspberry Pi)
https://roboticsbackend.com/introduction-to-wiringpi-for-raspberry-pi/

Use and compile WiringPi with ROS on Raspberry Pi
https://roboticsbackend.com/use-and-compile-wiringpi-with-ros-on-raspberry-pi/

Digital Pins With Interrupts
https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/

Light-Seeking Robot
https://learn.sparkfun.com/tutorials/light-seeking-robot/all

Simplest Light-Source Follower Robot
https://www.instructables.com/Simplest-Light-Source-Follower-Robot/