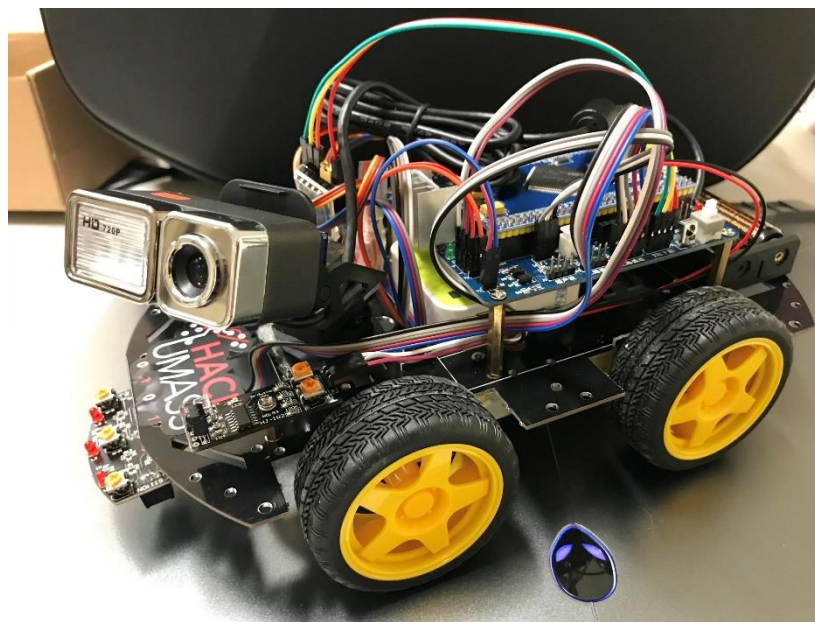


## Smart Robot Car Using STM32 Microcontroller



**Zhenyu Pan & Jiaming Nie**

November 5, 2017

# Outline

## Introduction

### Function

Function 1 Obstacle Avoidance

Function 2 Path Tracing

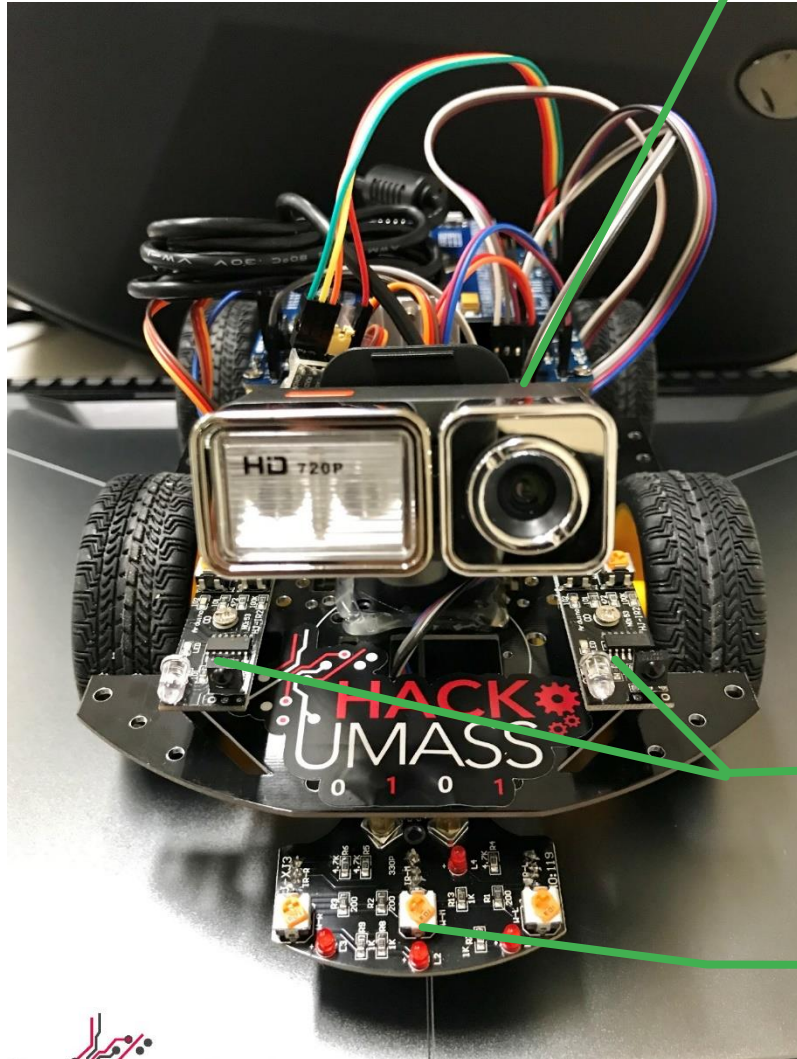
Function 3 PC Control using Wi-Fi and Real-time Video Transmission

## Future Work



# Introduction

Front



720P HD Camera

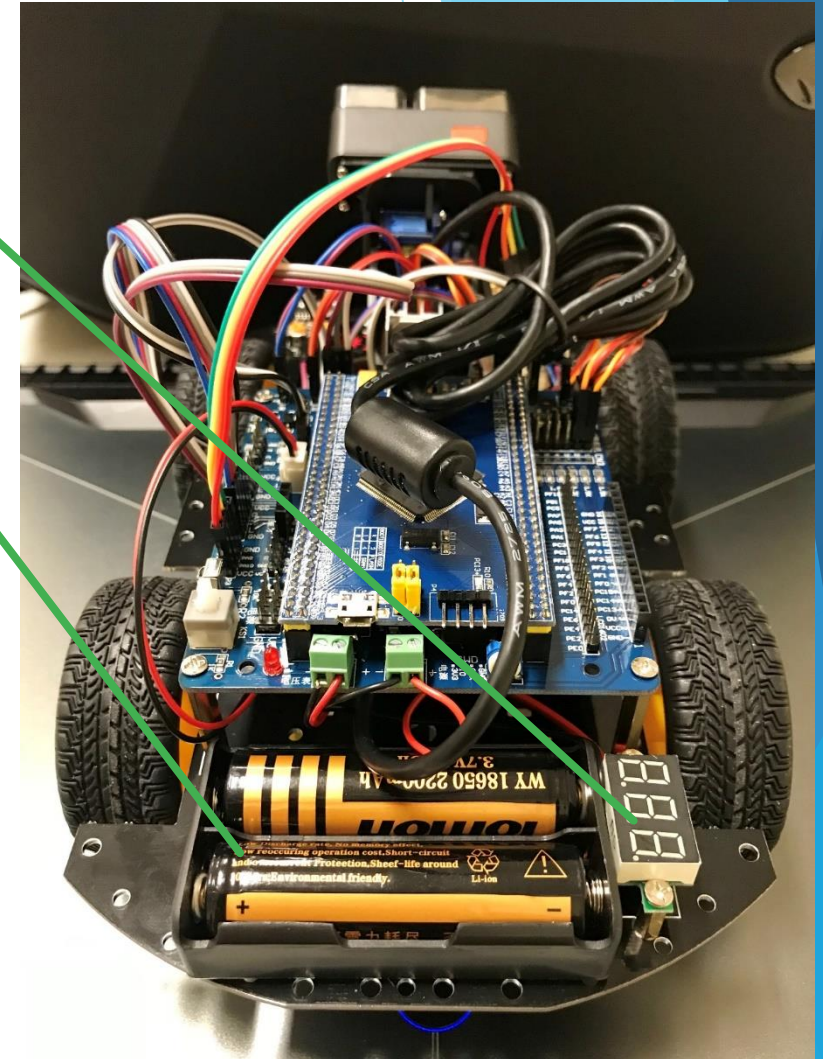
Digital Display  
Voltmeter

18650 Battery Holder  
and Lithium Battery

HJ-IR2 Infrared Obstacle  
Avoidance Module

HJ-XJ3 Three-way Path  
Tracing Module

Back

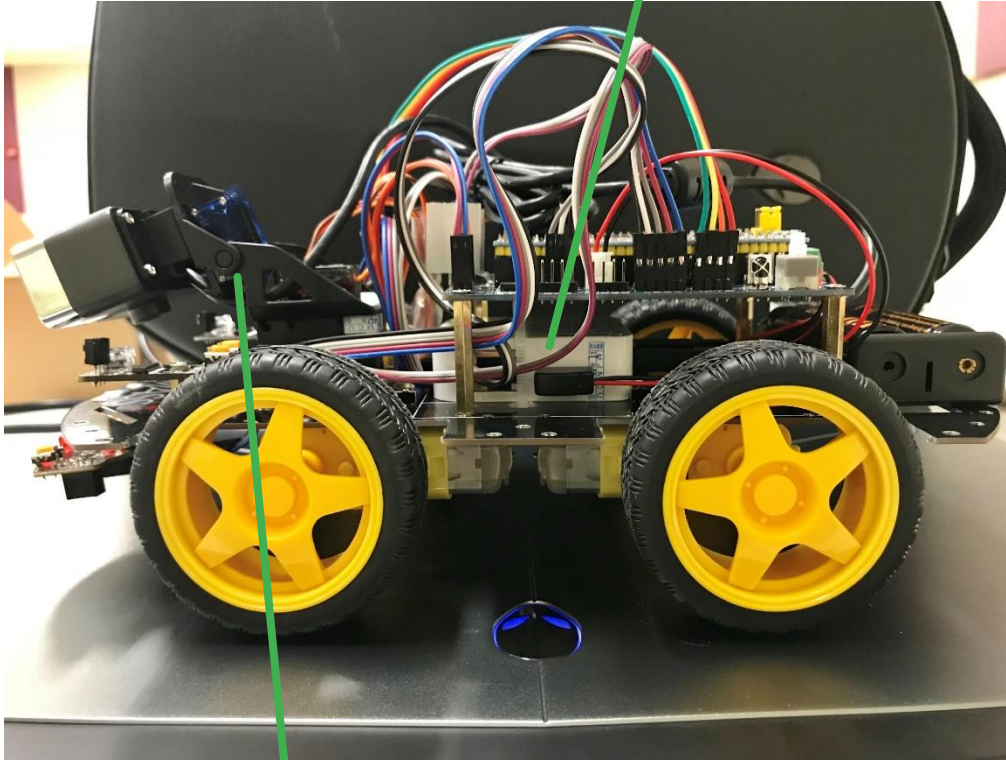




# Introduction

60MB Memory Wi-Fi Data  
Transmission Module

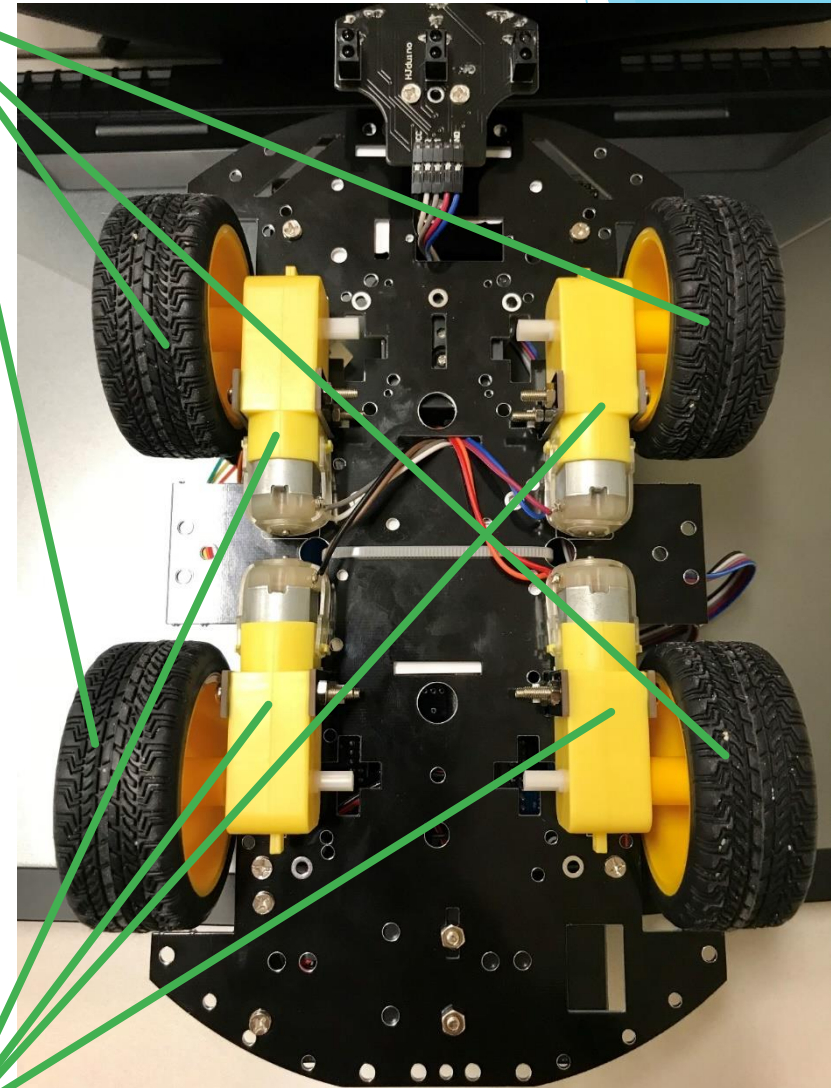
Side



Two-dimension Servo  
Controller

Tires

Bottom



Anti-interference  
Carbon Brush TT  
Reduction Motor

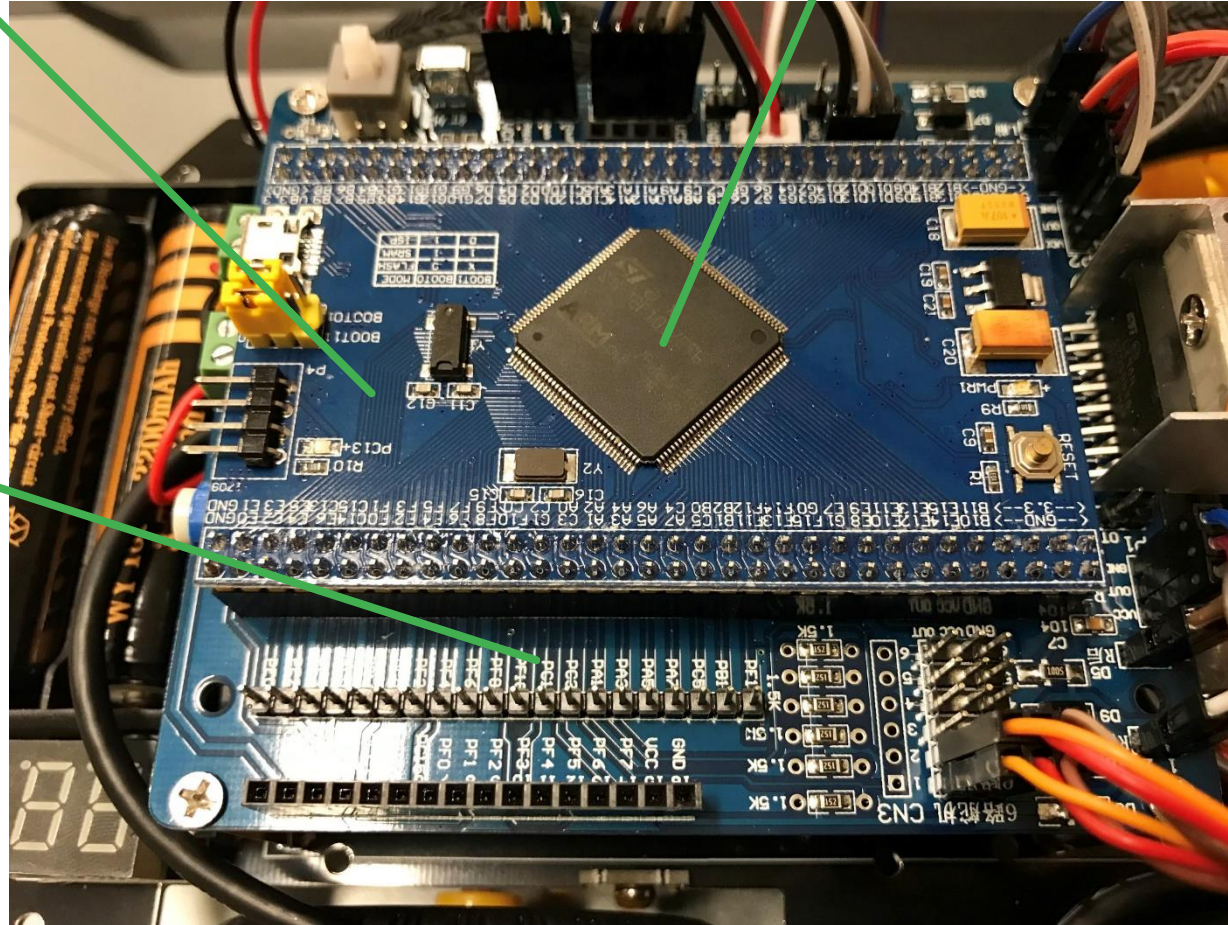
# Introduction

STM32 Core-board

Top

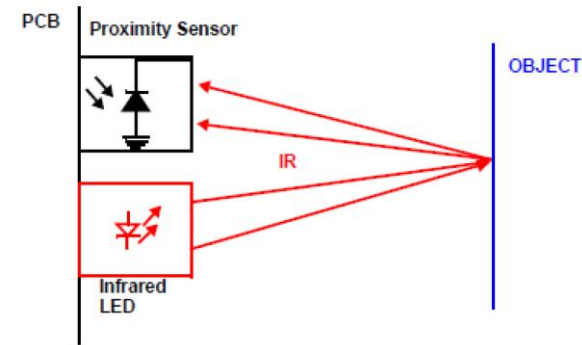
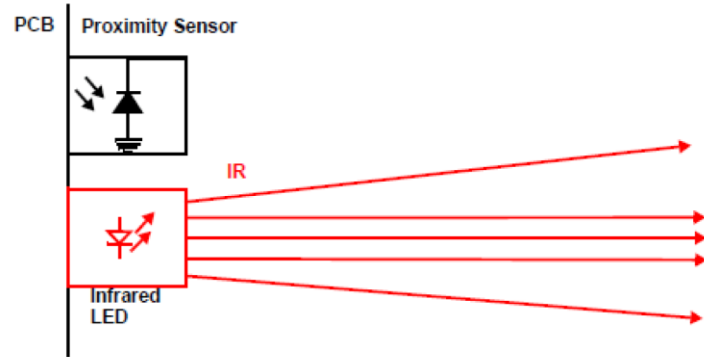
STM32F103ZET6  
Microcontroller

STM32-L298N  
Driver Board





# Function 1 Obstacle Avoidance



Algorithm:

If no infrared sensor senses obstacles, then go straight forward

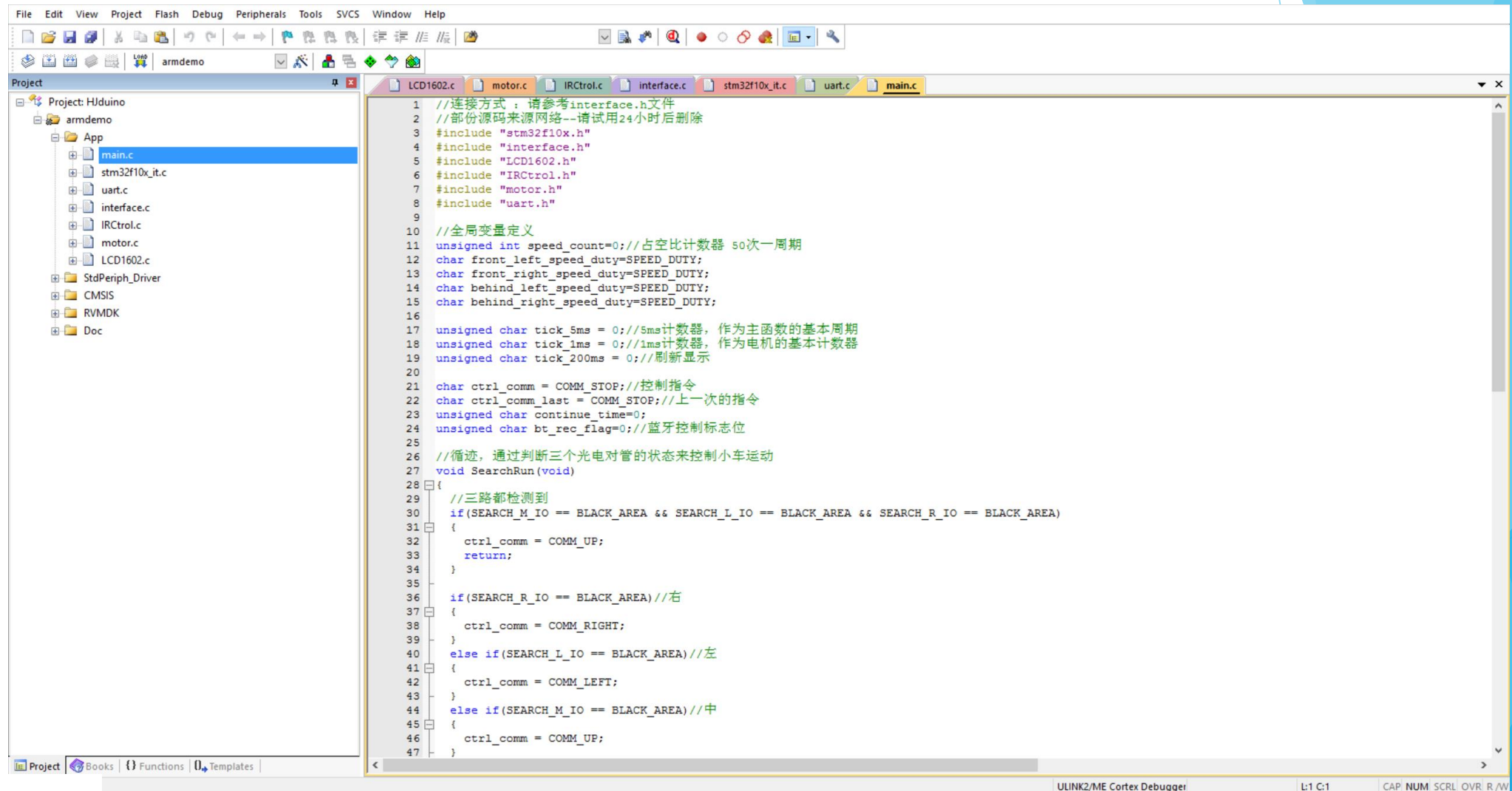
If left infrared sensor senses obstacles, then go backward 500ms and right turn 500ms

If right infrared sensor senses obstacles, then go backward 500ms and left turn 500ms

If both infrared sensors sense obstacles, then go backward 700ms and right turn 500ms

# Function 1 Obstacle Avoidance

## Obstacle Avoidance Code



The screenshot displays an IDE window with a project named 'HJduino'. The left sidebar shows the project structure, including files like 'main.c', 'stm32f10x\_it.c', 'uart.c', 'interface.c', 'IRCtrl.c', 'motor.c', and 'LCD1602.c'. The main editor window shows the code for 'main.c'. The code includes various headers and defines global variables for speed control and timing. It also contains a function 'SearchRun' that implements obstacle avoidance logic based on sensor readings.

```
1 //连接方式：请参考interface.h文件
2 //部份源码来源网络--请试用24小时后删除
3 #include "stm32f10x.h"
4 #include "interface.h"
5 #include "LCD1602.h"
6 #include "IRCtrl.h"
7 #include "motor.h"
8 #include "uart.h"
9
10 //全局变量定义
11 unsigned int speed_count=0;//占空比计数器 50次一周期
12 char front_left_speed_duty=SPEED_DUTY;
13 char front_right_speed_duty=SPEED_DUTY;
14 char behind_left_speed_duty=SPEED_DUTY;
15 char behind_right_speed_duty=SPEED_DUTY;
16
17 unsigned char tick_5ms = 0;//5ms计数器, 作为主函数的基本周期
18 unsigned char tick_1ms = 0;//1ms计数器, 作为电机的基本计数器
19 unsigned char tick_200ms = 0;//刷新显示
20
21 char ctrl_comm = COMM_STOP;//控制指令
22 char ctrl_comm_last = COMM_STOP;//上一次的指令
23 unsigned char continue_time=0;
24 unsigned char bt_rec_flag=0;//蓝牙控制标志位
25
26 //循迹, 通过判断三个光电对管的状态来控制小车运动
27 void SearchRun(void)
28 {
29     //三路都检测到
30     if (SEARCH_M_IO == BLACK_AREA && SEARCH_L_IO == BLACK_AREA && SEARCH_R_IO == BLACK_AREA)
31     {
32         ctrl_comm = COMM_UP;
33         return;
34     }
35
36     if (SEARCH_R_IO == BLACK_AREA) //右
37     {
38         ctrl_comm = COMM_RIGHT;
39     }
40     else if (SEARCH_L_IO == BLACK_AREA) //左
41     {
42         ctrl_comm = COMM_LEFT;
43     }
44     else if (SEARCH_M_IO == BLACK_AREA) //中
45     {
46         ctrl_comm = COMM_UP;
47     }
48 }
```

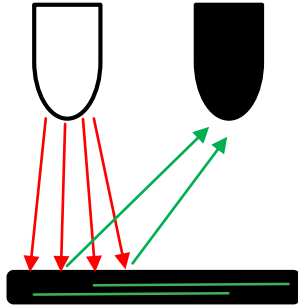
# Function 1 Obstacle Avoidance

## Live Demonstration

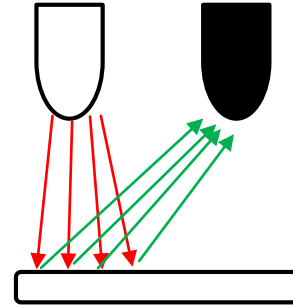




## Function 2 Path Tracing



Black absorbs infrared ray



White reflects infrared ray

Algorithm:

If only left infrared sensor senses black line, then slow down and turn left

If left and middle infrared sensors sense black line, then keep speed and turn left

If only right infrared sensor senses black line, then slow down and turn right

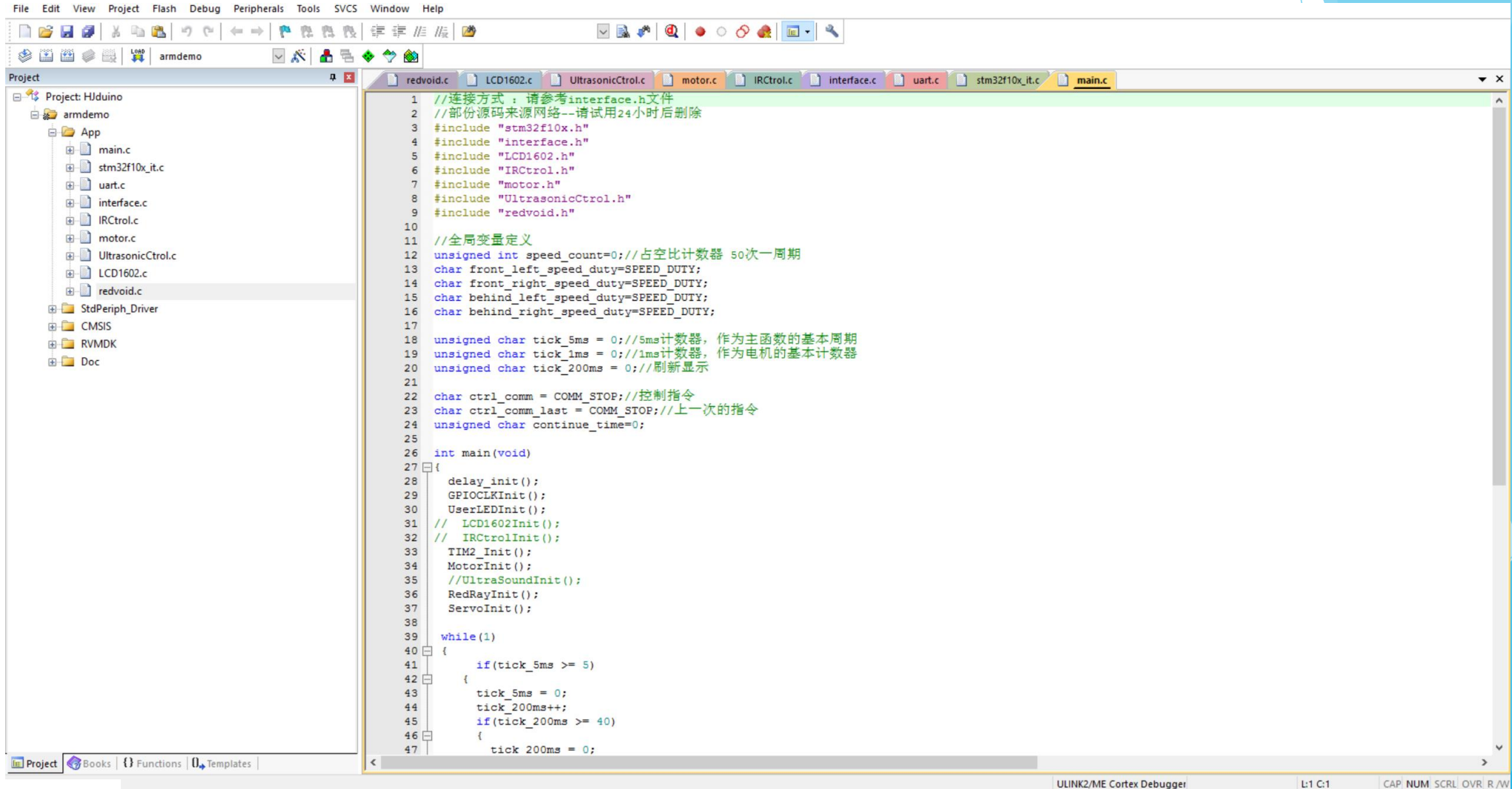
If right and middle infrared sensors sense black line, then keep speed and turn right

If only middle infrared sensors sense black line, then speed up and go straight forward

If all infrared sensors sense black line, then go straight forward

# Function 2 Path Tracing

## Path Tracing Code



The screenshot displays an IDE window with a project named 'HJduino'. The left sidebar shows the project structure, including a folder 'App' containing files like 'main.c', 'stm32f10x\_it.c', 'uart.c', 'interface.c', 'IRCtrl.c', 'motor.c', 'UltrasonicCtrl.c', 'LCD1602.c', and 'redvoid.c'. The main editor window shows the code for 'main.c'. The code includes various headers and defines global variables for speed counting and timing. It also includes a main function that initializes hardware and enters a while loop for the main logic.

```
1 //连接方式：请参考interface.h文件
2 //部份源码来源网络--请试用24小时后删除
3 #include "stm32f10x.h"
4 #include "interface.h"
5 #include "LCD1602.h"
6 #include "IRCtrl.h"
7 #include "motor.h"
8 #include "UltrasonicCtrl.h"
9 #include "redvoid.h"
10
11 //全局变量定义
12 unsigned int speed_count=0;//占空比计数器 50次一周期
13 char front_left_speed_duty=SPEED_DUTY;
14 char front_right_speed_duty=SPEED_DUTY;
15 char behind_left_speed_duty=SPEED_DUTY;
16 char behind_right_speed_duty=SPEED_DUTY;
17
18 unsigned char tick_5ms = 0;//5ms计数器，作为主函数的基本周期
19 unsigned char tick_1ms = 0;//1ms计数器，作为电机的基本计数器
20 unsigned char tick_200ms = 0;//刷新显示
21
22 char ctrl_comm = COMM_STOP;//控制指令
23 char ctrl_comm_last = COMM_STOP;//上一次的指令
24 unsigned char continue_time=0;
25
26 int main(void)
27 {
28     delay_init();
29     GPIOCLKInit();
30     UserLEDInit();
31     // LCD1602Init();
32     // IRCtrlInit();
33     TIM2_Init();
34     MotorInit();
35     //UltraSoundInit();
36     RedRayInit();
37     ServoInit();
38
39     while(1)
40     {
41         if(tick_5ms >= 5)
42         {
43             tick_5ms = 0;
44             tick_200ms++;
45             if(tick_200ms >= 40)
46             {
47                 tick_200ms = 0;
```

# Function 2 Path Tracing

## Video Demonstration



Zhenyu Pan & Jiaming Nie November 5, 2017 Hackathon UMass Amherst

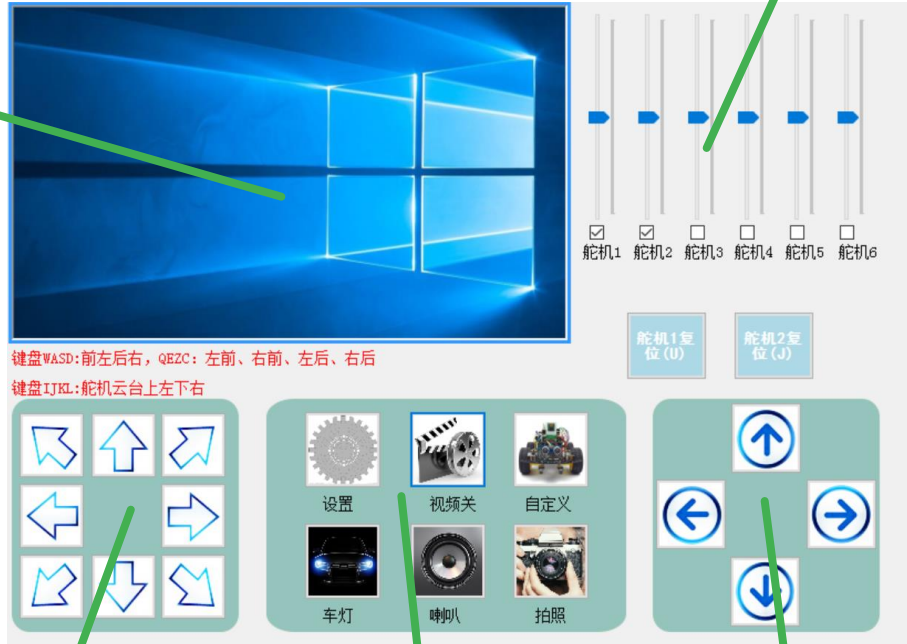


# Function 3 PC Control using Wi-Fi and Real-time Video Transmission

## Smart Robot Car Control Application on PC

Real-time Video

Two-dimension Servo Controller status



Connection settings should be done:

- videourl=<http://192.168.8.1:8083/?action=snapshot>
- controlUrl=192.168.8.1
- controlPort=2001

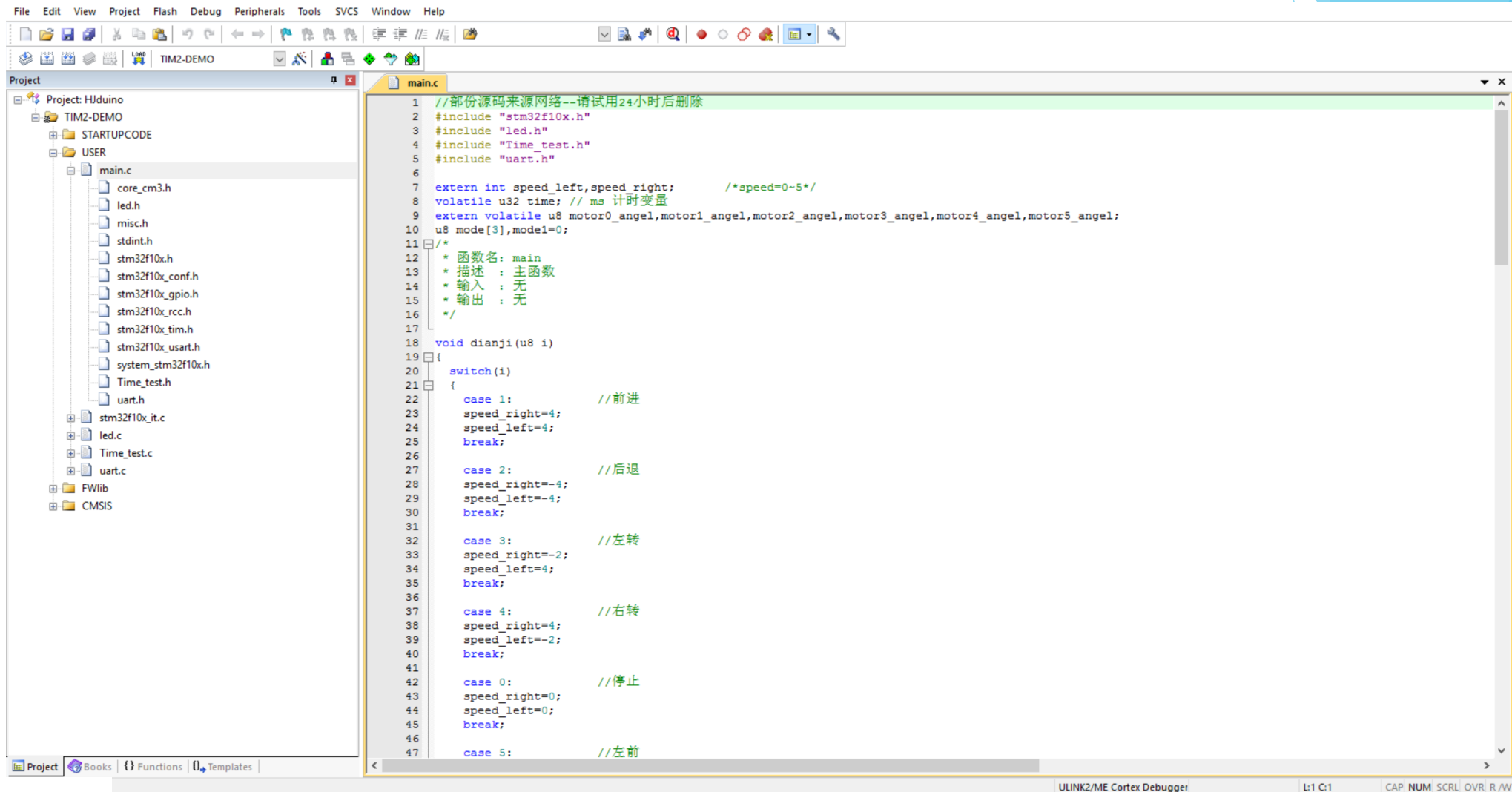
Smart Robot Car control

Settings, Video Stream on or off, Flash light, Horn, Take picture function

Two-dimension Servo Controller control

# Function 3 PC Control using Wi-Fi and Real-time Video Transmission

## Function 3 Code



```
1 //部份源码来源网络--请试用24小时后删除
2 #include "stm32f10x.h"
3 #include "led.h"
4 #include "Time_test.h"
5 #include "uart.h"
6
7 extern int speed_left, speed_right; /*speed=0-5*/
8 volatile u32 time; // ms 计时变量
9 extern volatile u8 motor0_angel, motor1_angel, motor2_angel, motor3_angel, motor4_angel, motor5_angel;
10 u8 mode[3], model=0;
11
12 /*
13  * 函数名: main
14  * 描述 : 主函数
15  * 输入 : 无
16  * 输出 : 无
17  */
18 void dianji(u8 i)
19 {
20     switch(i)
21     {
22     case 1: //前进
23         speed_right=4;
24         speed_left=4;
25         break;
26     case 2: //后退
27         speed_right=-4;
28         speed_left=-4;
29         break;
30     case 3: //左转
31         speed_right=-2;
32         speed_left=4;
33         break;
34     case 4: //右转
35         speed_right=4;
36         speed_left=-2;
37         break;
38     case 0: //停止
39         speed_right=0;
40         speed_left=0;
41         break;
42     case 5: //左前
```

# Function 3 PC Control using Wi-Fi and Real-time Video Transmission

## Video Demonstration





# Future Work

☺ Achieve automatic driving technology using Artificial Intelligence, Machine Learning and Computer Vision knowledge

☺ Design and build software using Automatic Control Theory, PID Algorithm, Fuzzy Control, Route Memorize, and different Filtering Algorithms like Kalman Filtering



**Thank you ~**

**Q & A Time**

