EIE 331 Final Project

Contents

1. Outline

1.1 Team #: Group 1

1.2 Members: Liang Peng, Huo Zhifeng, Wu Yuze

1.3 Project Name: WIFI remote-control car with obstacle avoidance

1.4 Project Description: A remote-control car whose driving directions can be controlled by computer keyboard input using WIFI server-client mode, and equipped with obstacle avoidance by including esp32 cam and object detection model running on computer.

1.5 GitHub Repository: https://github.com/LiangPeng03/OD-WiFi-remote-car

1.6 Project Components: Hardware, software, and other resources needed for project implementation.

1.6.1 Hardware

- Power bank
- Power adapter board
- TM4C123 board
- Motors (2)
- Wheels (3)
- Framework of car
- esp8266 wifi
- esp32 cam
- Styrofoam with velcro fastener

1.6.2 Software

- **Platform**: CCS (PWM control) + VS Code (wifi remote control and object detection)
- **Programs**: PWM control, Wi-Fi remote control, object detection

1.7 Technical Challenges: Anticipated challenges and technical issues required to be overcome.

1.7.1 How to maintain power supply using power bank.

1.7.2 How to precisely control the speed of motor to move straight.

1.7.3 How to assemble all hardware nicely as a complete car.

1.7.4 How to decide when to avoid obstacles.

1.8 Duty

Group Leader: Liang Peng

Development/Testing: Liang Peng, Huo Zhifeng

Documentation/Review: Wu Yuze

1.9 Schedule

Week	Date Range	Sprint	Tasks	Responsible
1	11.18-11.24	**Sprint 1** (Days 1-3)	- Requirement Analysis: Determine the functionalities to implement	Liang Peng, Huo Zhifeng, Wu Yuze
			- Clarify project goals and scope	Wu Yuze, Liang Peng, Huo Zhifeng
			- Develop detailed project plan	Wu Yuze
			- Set up development environment	Liang Peng, Huo Zhifeng
			- Gather necessary resources	Liang Peng, Huo Zhifeng
		Sprint 2 (Days 4-6)	- Develop PWM signal program to test the usability of motors	Liang Peng
2	11.25-12.1	**Sprint 3** (Days 7-9)	- Develop remote control program	Liang Peng
		Sprint 4 (Days 10-12)	- Preliminary testing of driving modules	Liang Peng, Huo Zhifeng
			- Fix identified issues	Liang Peng, Huo Zhifeng
			- Documenting Bugs	Wu Yuze
3	12.2-12.8	**Sprint 5** (Days 13-15)	- Complete development of object detection module	Liang Peng, Huo Zhifeng
			- Optimize code and performance	
		Sprint 6 (Days 16-18)	- Module integration	Liang Peng, Huo Zhifeng
			- System integration testing	
			- Resolve issues found during integration	
4	12.9-12.15	**Sprint 7** (Days 19-21)	- Comprehensive testing and debugging	Liang Peng, Huo Zhifeng

			- Ensure all functionalities work properly	
		Sprint 8 (Days 22-24)	- Prepare project documentation and presentation materials	Wu Yuze, Huo Zhifeng, Liang Peng
			- Project presentation and demonstration	
5	12.16-12.25	**Sprint 9** (Days 25-34)	- Review project process and write final report	Wu Yuze, Huo Zhifeng, Liang Peng
			- Submit project deliverables	

2. Requirement Analysis

2.1 Functional Requirements

- **Remote Control Functionality**
  - Users should be able to control the car's movement (forward, backward, left, right) using a mobile app or web interface.
  - The system must support real-time control with minimal latency.
- **Wi-Fi Connectivity**
  - The car should connect to a local Wi-Fi network to enable remote operation.
  - It should provide a mechanism for pairing with a mobile device or web interface.
- **Power Management**
  - The system must monitor battery levels and provide alerts to the user when the battery is low.
  - Implement an automatic shutdown feature when battery levels fall below a certain threshold.
- **Camera**
  - Users can see the perspective of the car.
  - The car can automatically keep its distance with obstacles under different light conditions.

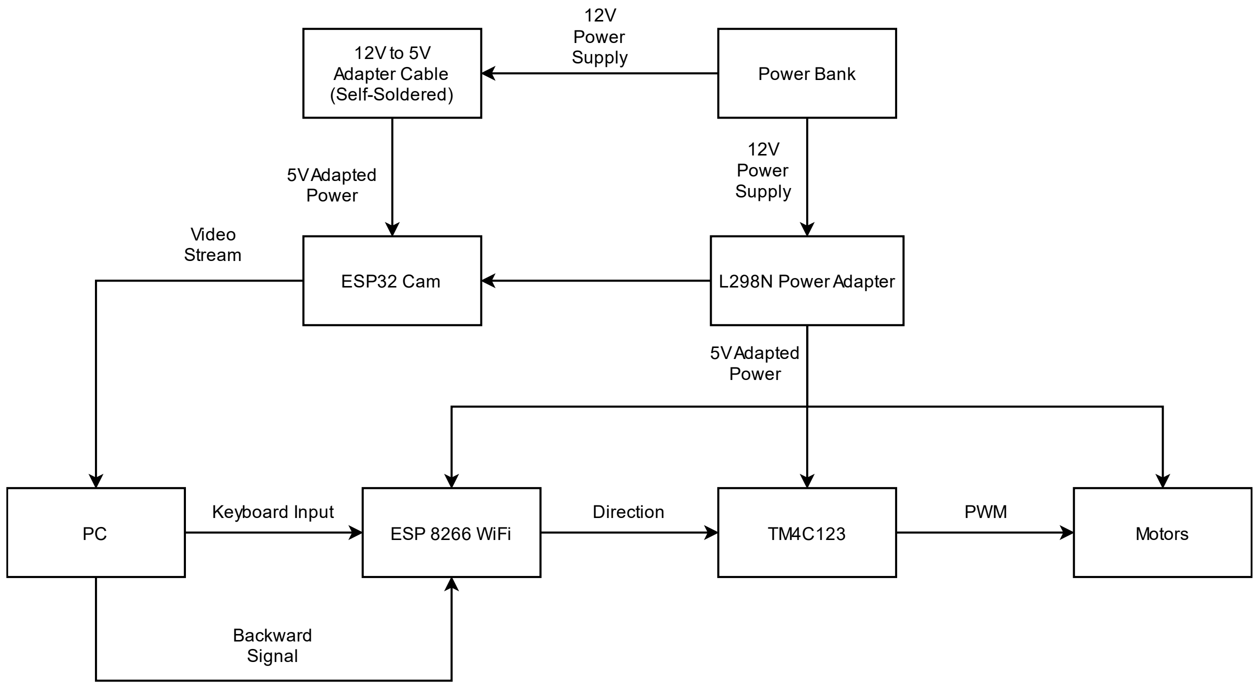
2.2 Non-functional Requirements

- **Performance**
  - The system should respond to user commands with a latency of less than 100 milliseconds.
- **Reliability**
  - The car should maintain a stable connection within a range of at least 30 meters under normal conditions.
  - The system should recover gracefully from disconnections and allow reconnection without user intervention.
- **Usability**

    - The user interface must be intuitive and easy to navigate for users of all ages.
    - Provide clear instructions and feedback to users during operation.
- **Scalability**
    - The system should be designed to support additional features in the future, such as GPS tracking or additional sensors.
- **Maintainability**
    - The codebase should be modular and well-documented to facilitate future updates and maintenance.
    - Provide diagnostic tools or logs to help troubleshoot issues.

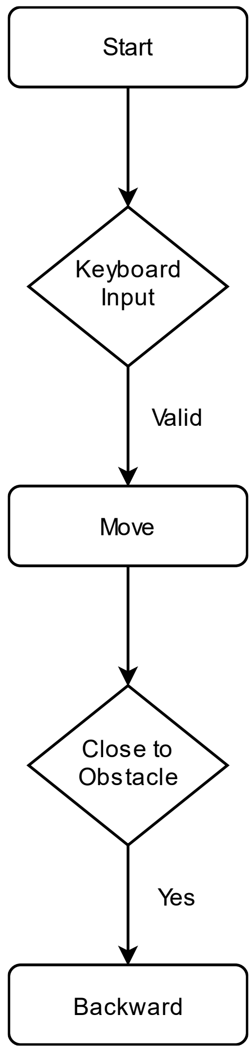
3. Design Document

3.1 Architecture



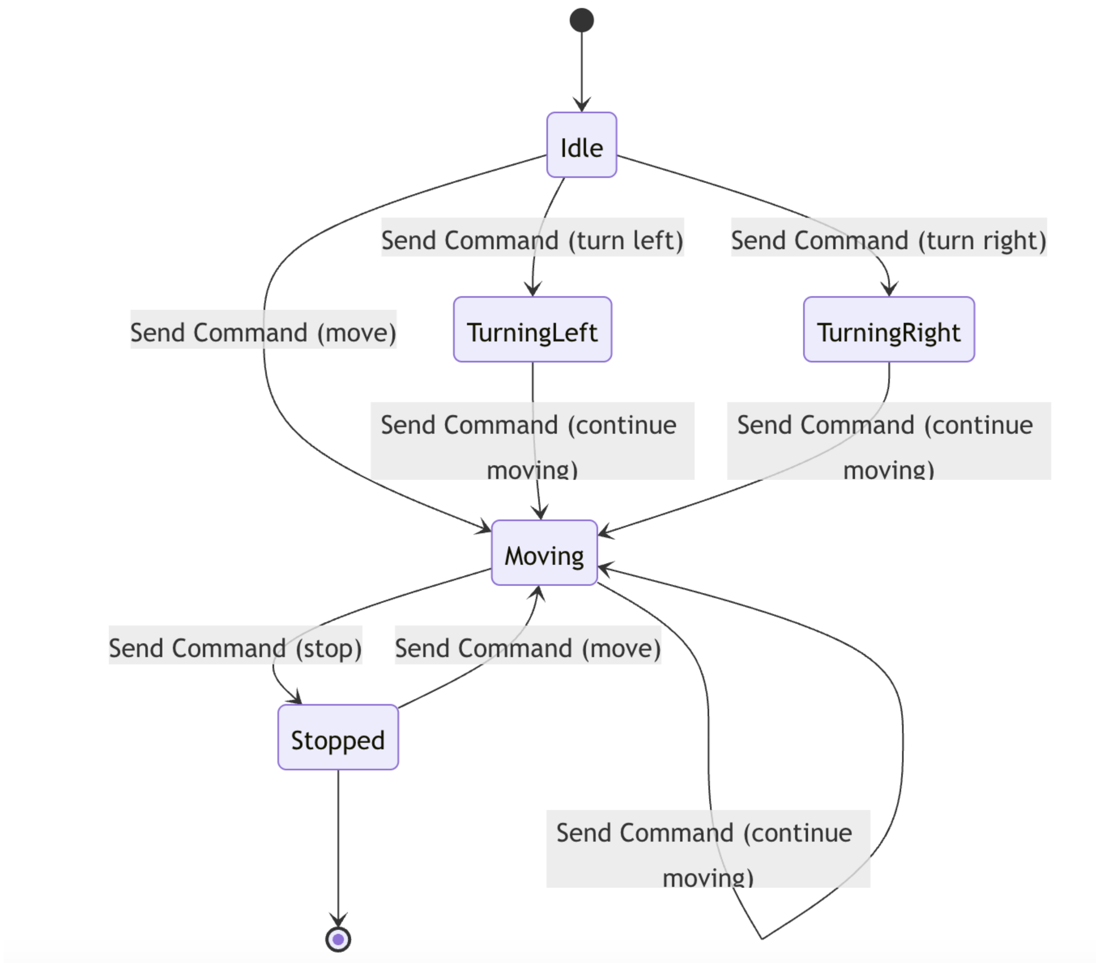
The architecture of our project is roughly as shown in the diagram: firstly, power is supplied by the power bank to ensure that each device can run at full capacity, and a 12V to 5V power adapter board is connected to stabilize the voltage. Camera and vehicle movement module is independent. For the vehicle movement module, first listen to the keyboard press operation from computer, if it is a valid command, will be sent to the motherboard connected to the esp8266 Wi-Fi, according to the direction of the input to the motor output different PWM duty cycle to achieve a different rotational speed, so as to realize the forward, backward, left right and other actions. Then comes the logic of the camera to identify obstacles, the camera transmits the video input to the computer, the computer runs the object recognition model to return the object's labeling box, by judging the approximate size of the object to determine whether to move back to avoid obstacles, and if necessary, then sends the corresponding move back signal to the Wi-Fi chip, and then through the motherboard to control the vehicle to move back.

3.2 Flow Chart

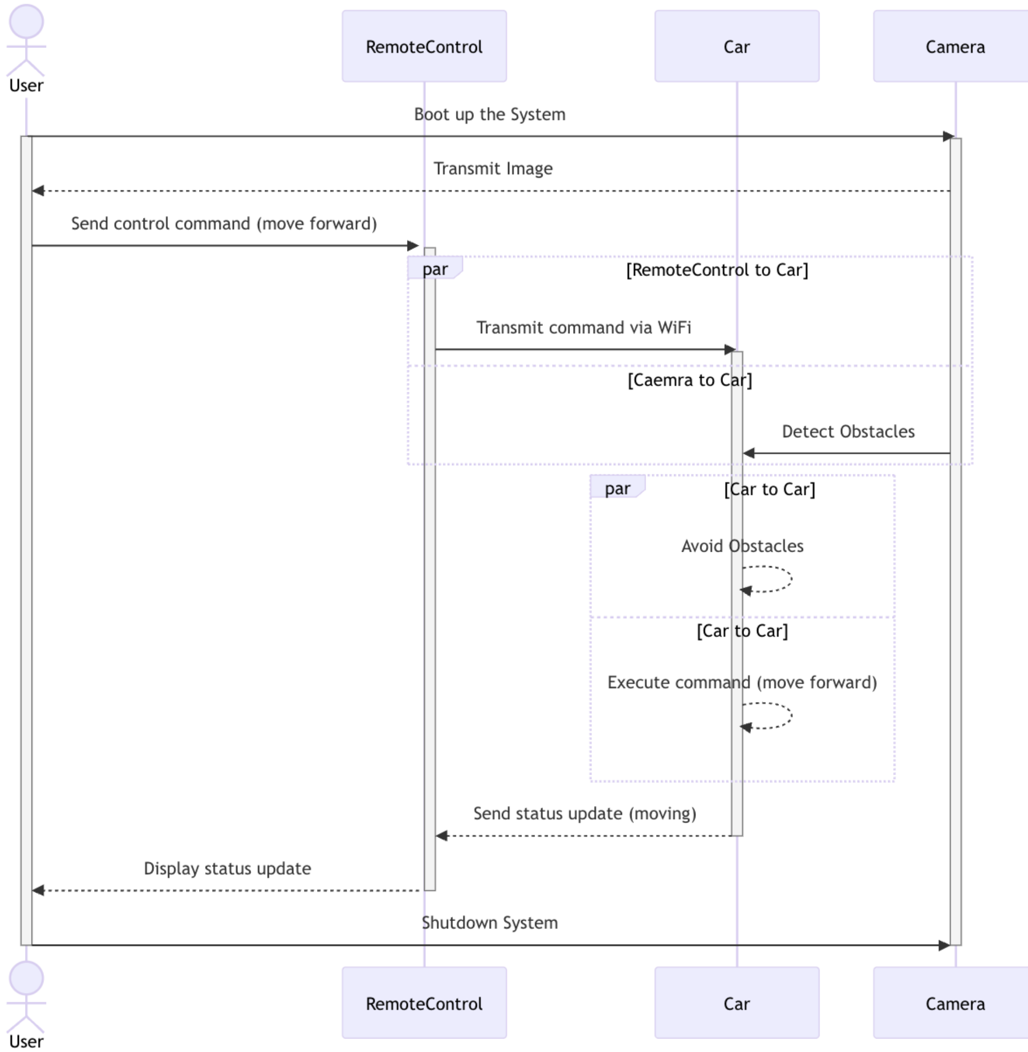


This is the general flow chart, after powering on, the vehicle control program control.py and camera object recognition program esp32cam.py require to be started, control.py is responsible for listening to the keyboard input, if it is a valid directions input, it will be sent to the Wi-Fi chip and then through the motherboard to control the motors, and esp32cam.py is responsible for determining the type and size of the object and deciding whether it needs to move backward for obstacle avoidance, and if it does, the backward signal will be sent to the main board.

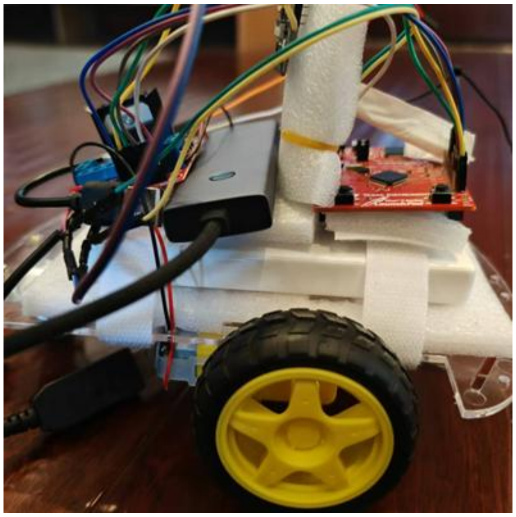
3.3 State Diagram

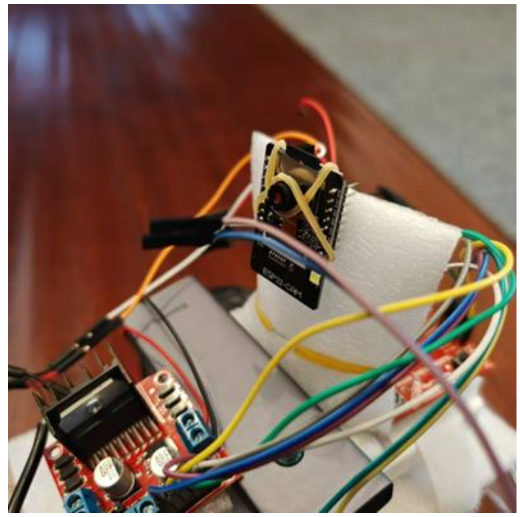


3.4 Sequence Diagram



3.1 Hardware Assembly and Exterior Design





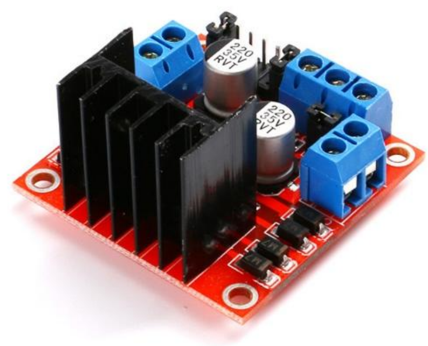
The base plate is not perfectly plain and has limited space, making it more difficult to secure all the parts. So, we took the method of hardware stacking and increasing friction to solve the problem.

First, put a layer of styrofoam on the base plate and put the power bank on the bottom, the power adapter board and motherboard were placed on top layer, and then use Velcro fastener to tie the power bank to the base plate to prevent it from falling off. Stick a layer of styrofoam underneath the parts, and then use Velcro fastener to stick it on top of the power bank. This not only fixes the power bank but also facilitates the connection of the upper development board and power adapter board, and modular disassembly. To ensure that the camera's view is not blocked by wires, we fixed it in the middle of the power supply board and the motherboard, and nailed it to the vertical Styrofoam to fix it at height.

4. Development Details

4.1 Hardware Introductions

4.1.1 Movement Module



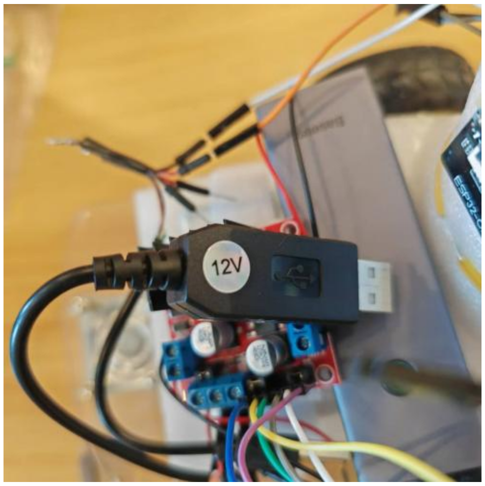
tt-motor **L298N Power Adapter board**

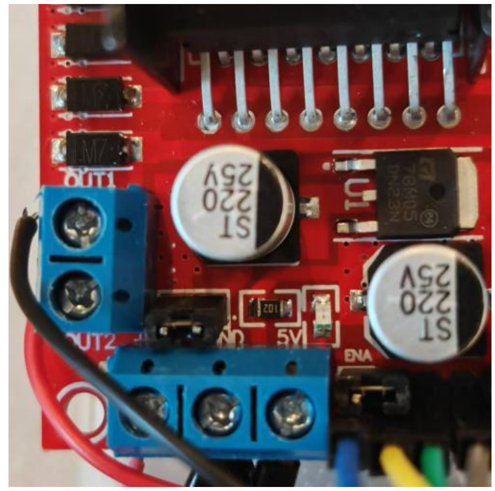
The vehicle movement part of the cart uses tt- motor and l298n charging board.

The tt-motor can be adjusted to different effective voltage levels to achieve different speed control by adjusting PWM duty cycle.

The L298n power adapter board can not only supply power to the motor, but also control the positive and negative direction of the motor through 4 direction control pins to control the directions of forward and backward, and there are also two PWM interfaces, to regulate the speed of the motor on both sides.

4.1.2 Power Supply





12V In **5V Out**

Driving the motor requires a higher current and therefore affects the power supply to the development board, so we used a higher voltage of 12V to power the power supply board first, and used the 5V step-down regulated power supply of the power adapter board to power the development board. By adopting the method we can power the motor and the board at the same time using a single power supply.

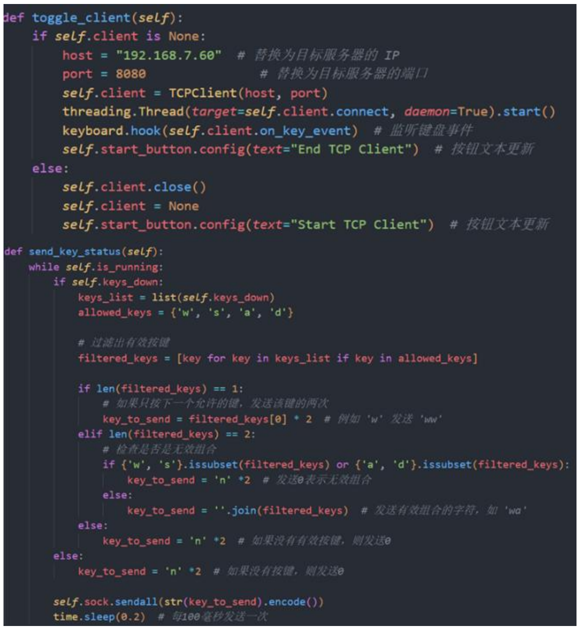
4.1.3 Camera

The object detection module adopts Arduino esp32-cam system. The original plan was to use the camera directly pin to the development board, but it requires 8 I2C pins for data transfer, which is the total amount on board. Considering our original plan will take two I2C interface to connect the accelerometer, we turned to use the system.

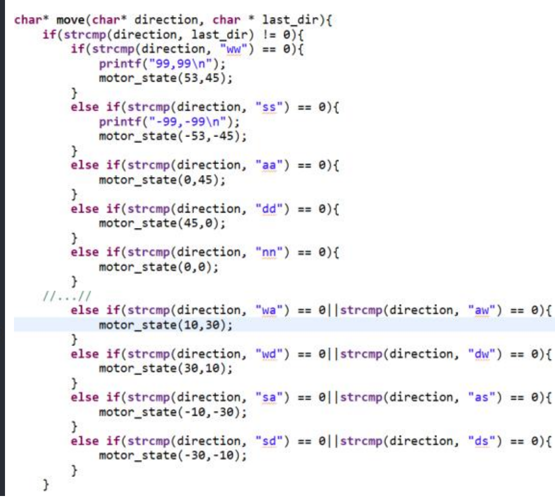
However, esp32-cam need 5V2A stabilized voltage current, which is the maximum output voltage of the power adapter board. As a result, the camera will not be able to be fully powered. So, we finally adopted the method of powering it by power bank.

4.2 Program Logic

4.2.1 Direction Control



```python
def toggle_client(self):
    if self.client is None:
        host = "192.168.7.60"  # 替换为目标服务器的 IP
        port = 8080            # 替换为目标服务器的端口
        self.client = TCPClient(host, port)
        threading.Thread(target=self.client.connect, daemon=True).start()
        keyboard.hook(self.client.on_key_event)  # 监听键盘事件
        self.start_button.config(text="End TCP Client")  # 按钮文本更新
    else:
        self.client.close()
        self.client = None
        self.start_button.config(text="Start TCP Client")  # 按钮文本更新

def send_key_status(self):
    while self.is_running:
        if self.keys_down:
            keys_list = list(self.keys_down)
            allowed_keys = {'w', 's', 'a', 'd'}

            # 过滤出有效按键
            filtered_keys = [key for key in keys_list if key in allowed_keys]

            if len(filtered_keys) == 1:
                # 如果只按下一个允许的键，发送该键的两次
                key_to_send = filtered_keys[0] * 2  # 例如 'w' 发送 'ww'
            elif len(filtered_keys) == 2:
                # 检查是否是无效组合
                if {'w', 's'}.issubset(filtered_keys) or {'a', 'd'}.issubset(filtered_keys):
                    key_to_send = 'n' *2  # 发送0表示无效组合
                else:
                    key_to_send = ''.join(filtered_keys)  # 发送有效组合的字符，如 'wa'
            else:
                key_to_send = 'n' *2  # 如果没有有效按键，则发送0
        else:
            key_to_send = 'n' *2  # 如果没有按键，则发送0

        self.sock.sendall(str(key_to_send).encode())
        time.sleep(0.2)  # 每100毫秒发送一次
```

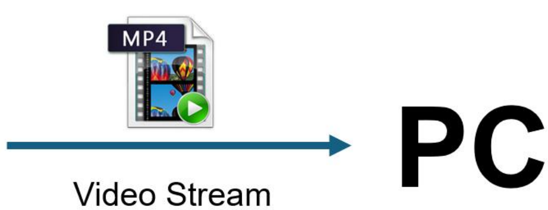


```c
char* move(char* direction, char * last_dir){
    if(strcmp(direction, last_dir) != 0){
        if(strcmp(direction, "ww") == 0){
            printf("99,99\n");
            motor_state(53,45);
        }
        else if(strcmp(direction, "ss") == 0){
            printf("-99,-99\n");
            motor_state(-53,-45);
        }
        else if(strcmp(direction, "aa") == 0){
            motor_state(0,45);
        }
        else if(strcmp(direction, "dd") == 0){
            motor_state(45,0);
        }
        else if(strcmp(direction, "nn") == 0){
            motor_state(0,0);
        }
    //...//
        else if(strcmp(direction, "wa") == 0||strcmp(direction, "aw") == 0){
            motor_state(10,30);
        }
        else if(strcmp(direction, "wd") == 0||strcmp(direction, "dw") == 0){
            motor_state(30,10);
        }
        else if(strcmp(direction, "sa") == 0||strcmp(direction, "as") == 0){
            motor_state(-10,-30);
        }
        else if(strcmp(direction, "sd") == 0||strcmp(direction, "ds") == 0){
            motor_state(-30,-10);
        }
    }
```

control.py **motor.c**

The direction control part of the cart is composed of two python scripts, firstly, control.py controls the vehicle's forward, backward, left and right movement. The cart's esp8266 Wi-Fi will serve as a server, and then the Python script on the PC will connect to the cart's server and send continuous commands of different actions, and the cart only needs to get the latest remote control signal every time, the and update the value of PWM duty cycle for its own motor speed pin and the h-bridge pin level to control the direction of the cart's motor.

4.2.2 Video Stream Transmit



The object detection module creates a server through esp 8266 Wi-Fi to transmit the video stream returned from camera to the PC for processing

4.2.3 Obstacle Avoidance

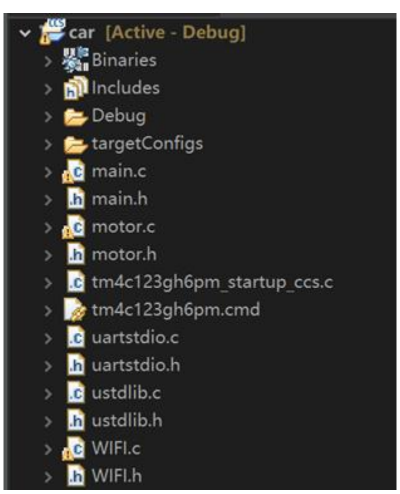
```python
# 发送c小车后退信号"ss"到esp wifi
try:
    # 创建一个 TCP/IP socket
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # 连接到服务器
    server_address = ('192.168.7.60', 8080)  # 使用服务器地址和端口
    sock.connect(server_address)

    # 发送信号
    tt = "ss"
    sock.sendall(tt.encode())

except Exception as e:
    print(f"Failed to send signal: {e}")
```

esp32cam.py: send backward signal

We use a python script with pre-built object detection models, through which we perform obstacle detection and judge its rough distance from the cart by assessing the size of returned labeling box. After detecting a specific kind of obstacle and detecting that its size exceeds a threshold, a back off signal will be sent to the vehicle and it can automatically move away from the obstacle.

4.3 Code Analysis

4.3.1 Project Structure



This is the programs of movement module which contains three main code files: **main.c** main function, **motor.c** with the motion state control function and **WIFI.c** server establishment function. The rest are dependent library function files.

4.3.2 main.c

```c
#include "main.h"
#include "motor.h"
#include "WIFI.h"

volatile char direction[2]={};

volatile char last_dir[2]={};

void delayUs(uint32_t us){
    SysCtlDelay((SysCtlClockGet() / 3000000) * us );
}
void delayMs(uint32_t ms){
    SysCtlDelay((SysCtlClockGet()  / 3000)* ms );
}

int main(void){
    SysCtlClockSet(SYSCTL_SYSDIV_2_5 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN | SYSCTL_XTAL_16MHZ);

    motor_init();

    esp8266Init();

    char *qq;

    while(1){

        *direction = getWASD();

        *last_dir = move(direction,last_dir);
    }
}

```

This is the **main.c** file, the main function includes the **initialization program** for the movement module including the **tt-motor** and the **L298n power adapter board**, the initialization program for the **esp8266 WIFI** communication chip and an infinite loop to **get the commands** sent from the remote control program on the computer side to **update the motor status**.

4.3.3 Motor.c

motor.c

```c

void motor_init(void){
    //PA 2~5 -> L298n in 1~4
    SYSCTL_RCGCGPIO_R |= 0x01; // 启用端口A时钟
    while ((SYSCTL_PRGPIO_R & 0x01) == 0); // 等待端口A可用
    GPIO_PORTA_DIR_R |= MOTOR_PIN; // 设置引脚为输出
    GPIO_PORTA_PUR_R |= MOTOR_PIN;        // 启用上拉电阻到PA 2~5
    GPIO_PORTA_DEN_R |= MOTOR_PIN; // 使能引脚

    //配置PWM时钟（设置USEPWMDIV分频器）
    SysCtlPWMClockSet(SYSCTL_PWMDIV_8);

    //使能时钟
    SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);

    //PWM信号分配
    // 3. 配置 PB6 和 PB7 为 PWM 功能
    GPIOPinConfigure(GPIO_PB6_M0PWM0); // PB6 作为 PWM0
    GPIOPinConfigure(GPIO_PB7_M0PWM1); // PB7 作为 PWM1

    GPIOPinTypePWM(GPIO_PORTB_BASE, GPIO_PIN_6 | GPIO_PIN_7);

    PWMGenConfigure(PWM0_BASE, PWM_GEN_0, PWM_GEN_MODE_UP_DOWN | PWM_GEN_MODE_NO_SYNC);
    PWMGenPeriodSet(PWM0_BASE, PWM_GEN_0, 100); // 设定周期，比如 1000 个时钟周期

    // 5. 设置 PWM 占空比
    PWMPulseWidthSet(PWM0_BASE, PWM_OUT_0, 0); // 设置 PB6 的占空比为 50%
    PWMPulseWidthSet(PWM0_BASE, PWM_OUT_1, 0); // 设置 PB7 的占空比为 25%

    // 6. 启用 PWM 输出
    PWMGenEnable(PWM0_BASE, PWM_GEN_0);
    PWMOutputState(PWM0_BASE, PWM_OUT_0_BIT | PWM_OUT_1_BIT, true); // 开启输出

}

```

This is the **motor_init()** function in the **motor.c**. Hardware using the TM4C123g PA2-5 connected to the L298n in1-4, with a combination of high and low levels through the “H-bridge” to control the direction of the two motors, PB6, PB7 are used for the PWM outputs to control the speed of the two motors. It is worth noting that the period is set to 100 system clock cycles, so in the subsequent setting of the effective range of the PWM duty cycle is [0,100), if the cycle is set to 100 then it becomes a general high level instead of PWM signals so that the L298n can’t receive.

```c
void motor_state(int out12_v,int out34_v){      //  (-100 , +100)
    if (out12_v == 0){
        MOTOR_PORT &= ~0x0C;     //拉低PB2,3
        PWMPulseWidthSet(PWM0_BASE, PWM_OUT_0, 0); // 设置 PB6 的占空比
    }
    if (out34_v == 0){
        MOTOR_PORT &= ~0x30;     //拉低PB4,5
        PWMPulseWidthSet(PWM0_BASE, PWM_OUT_1, 0); // 设置 PB7 的占空比
    }
    if (out12_v > 0){
        MOTOR_PORT |= 0x04;      //PB2   高
        MOTOR_PORT &= ~0x08;     //PB3   低
        PWMPulseWidthSet(PWM0_BASE, PWM_OUT_0, out12_v);
    }
    if (out34_v > 0){
        MOTOR_PORT |= 0x10;      //PB4   高
        MOTOR_PORT &= ~0x20;     //PB5   低
        PWMPulseWidthSet(PWM0_BASE, PWM_OUT_1, out34_v);
    }
    if (out12_v < 0){
        MOTOR_PORT |= 0x08;      //PB3   高
        MOTOR_PORT &= ~0x04;     //PB2   低
        PWMPulseWidthSet(PWM0_BASE, PWM_OUT_0, -out12_v);
    }
    if (out34_v < 0){
        MOTOR_PORT |= 0x20;      //PB5   高
        MOTOR_PORT &= ~0x10;     //PB4   低
        PWMPulseWidthSet(PWM0_BASE, PWM_OUT_1, -out34_v);
    }
    PWMGenEnable(PWM0_BASE, PWM_GEN_0);
    PWMOutputState(PWM0_BASE, PWM_OUT_0_BIT | PWM_OUT_1_BIT, true); // 开启输出
}
```

Next is **motor_state()** function, which is used to **modify the speed and direction** of the motor. In order to reduce the number of parameters for ease of use, allow negative numbers, the left and right two motors speed range is [-99,+99], [-99,+99], if the speed is set to negative it will be converted to reverse (backward), and then set the corresponding duty cycle.

```c
char* move(char* direction, char * last_dir){
    if(strcmp(direction, last_dir) != 0){
        if(strcmp(direction, "ww") == 0){
            //printf("99,99\n");
            motor_state(51,44);
        }
        else if(strcmp(direction, "ss") == 0){
            //printf("-99,-99\n");
            motor_state(-53,-44);
        }
        else if(strcmp(direction, "aa") == 0){
            motor_state(0,45);
        }
        else if(strcmp(direction, "dd") == 0){
            motor_state(50,0);
        }
        else if(strcmp(direction, "nn") == 0){
            motor_state(0,0);
        }
    //...//
        else if(strcmp(direction, "wa") == 0||strcmp(direction, "aw") == 0){
            motor_state(47,42);
        }
        else if(strcmp(direction, "wd") == 0||strcmp(direction, "dw") == 0){
            motor_state(50,43);
        }
        else if(strcmp(direction, "sa") == 0||strcmp(direction, "as") == 0){
            motor_state(-47,-45);
        }
        else if(strcmp(direction, "sd") == 0||strcmp(direction, "ds") == 0){
            motor_state(-55,-43);
        }
    }

    return * direction;
}
```

The **move()** function **compares and updates the direction signal** sent by the remote control with the previous signal, if it is the same then no change is needed to be made, else the **motor_state()** function with specific parameters will be called to modify the motor state. Since the remote control end uses nonlinear control, the motor state is also nonlinear control, but as long as the motor state is continuously updated in small increments over a period of time linear control can be achieved, which is also the original design. Although it is not realized, but theoretically it can be implemented.

4.3.4 WIFI.c

```c
WIFIDATA wifidata={0};

char direction[2]={};

void UART1IntHandler(void)
{
    uint32_t ui32IntStatus;
    uint8_t ui8RxBuffer[1024];
    uint8_t data = 0;
    uint8_t i = 0;

    ui32IntStatus = UARTIntStatus(UART1_BASE, true);

    UARTIntClear(UART1_BASE, ui32IntStatus);
//  UARTprintf("RNTER INTERRPUT");
    while (UARTCharsAvail(UART1_BASE))
    {

        data = (uint8_t) UARTCharGet(UART1_BASE);
        wifidata.recvbuf[wifidata.recvcnt] = data;
        if((char)data=='+'){
            direction[0]=wifidata.recvbuf[wifidata.recvcnt-4];
            direction[1]=wifidata.recvbuf[wifidata.recvcnt-3];
        }
        wifidata.recvcnt++;
        wifidata.recvcnt%=1024;

    }
```

This is the **UART1IntHandler()** function in **WIFI.c**, which is used to **generate an interrupt when a signal is received**, storing the information into the wifidata.recvbuf buffer and passing the 2-bit direction information in the signal into an array for status update for comparison with the previous direction information.

```c
uint8_t Wifi_Send_Cmd(char * cmd,char * recv,uint32_t timeout)
{
    uint32_t timecnt=0;
    delay_ms(10);
    memset(&wifidata,0,sizeof(wifidata));
    //delay_ms(10);
    UARTprintf("%s",(uint8_t *)cmd);
    //delay_ms(10);
    while(strstr((char *)wifidata.recvbuf,recv)==NULL){
        timecnt++;
        delay_ms(1);
        if(timecnt>=timeout){
            //UARTprintf("fail\n");
            UARTprintf("FAIL DATA = %s\n",wifidata.recvbuf);
            delay_ms(100);
    //      step_cnt = 1;
            return 0;
        }
    }
    //UARTprintf("this is :   %s\n",wifidata.recvbuf);
    delay_ms(100);
    return 1;

}
```

This is the **Wi-Fi send command** function used in experiment 3 to send AT commands as well as return error messages.

main.c WIFI.c

```c
void esp8266Init(void){
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART1);
        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);

        while(!SysCtlPeripheralReady(SYSCTL_PERIPH_UART1));
        while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOB));

         //配置对应的IO口
        GPIOPinConfigure(GPIO_PB0_U1RX);
        GPIOPinConfigure(GPIO_PB1_U1TX);
        GPIOPinTypeUART(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1);

          //配置串口1  8数据位，0校验位，1停止位，波特率115200
        UARTConfigSetExpClk(UART1_BASE, SysCtlClockGet(), 115200,
                            (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE));
        // UARTstdio
        UARTStdioConfig(1, 115200, SysCtlClockGet());

        UARTFIFOEnable(UART1_BASE);
        UARTFIFOLevelSet(UART1_BASE, UART_FIFO_TX2_8, UART_FIFO_RX2_8);
        UARTIntEnable(UART1_BASE, UART_INT_RX);
        IntPrioritySet(INT_UART1, 0x0);
        UARTIntRegister(UART1_BASE, UART1IntHandler);
        IntEnable(INT_UART1);
        IntMasterEnable();

        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
        //    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOF))
        //    {}
        GPIOPinTypeGPIOOutput(GPIO_PORTE_BASE, GPIO_PIN_5);
        GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_5, 0xFF);

        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
        //    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOF))
        //    {}
```

main.c WIFI.c

```c
        GPIOPinTypeGPIOOutput(GPIO_PORTE_BASE, GPIO_PIN_1|GPIO_PIN_2);

//        delay_ms(500);
        if(Wifi_Send_Cmd("AT\r\n","OK",1000)== 1)
        {
            //UARTprintf("1111111");
        }

        if(Wifi_Send_Cmd("AT+CWMODE=1\r\n","OK",1000)== 1){//Station模式
            if(Wifi_Send_Cmd("AT+CWJAP=\"LP1\",\"22222222\"\r\n","OK",10000)== 1){//连接热点
                //delay_ms(5000);

                if(Wifi_Send_Cmd("AT+CIPMUX=1\r\n","OK",3000)== 1){//单路连接模式
                    //delay_ms(1000);

                    if(Wifi_Send_Cmd("AT+CIPSERVER=1,8080\r\n","OK",2000)== 1)
                    {

                        //UARTprintf("this is:  %s   END\n",wifidata.recvbuf);
                        Wifi_Send_Cmd("AT+CIFSR\r\n","OK",2000);
                        return 1;
                     }

                 }

              }
           }
       return 0;
}
char* getWASD(void){
    char temp[2]={};
    temp[0]=direction[0];
    temp[1]=direction[1];
    //printf("temp:%s.\n",temp);
    return *temp;
}
```

The above is the initialization function for the esp8266, which is similar to the Wi-Fi initialization and network connection steps in Experiment 3.

On the esp32-cam side, an example program is used to save the picture to its own server for client to access.

4.3.5 esp32-cam.py

```python
if __name__ == '__main__':
    '''loop1 initialization'''
    # 初始化esp32cam url
    url='http://192.168.137.120//' # 改成串口收到的ESP32-CAM内网url，检查ESP32-CAM和上位机是不是在一个局域网网段上。

    # 初始化global img变量，存储esp32cam捕获到的图片（opencv格式）
    img = np.zeros((296, 400, 3), np.uint8)

    '''loop3 initialization'''
    # 导入模型和pipeline（loop3）
    '''image classification'''
    garbage_classification = pipeline(Tasks.image_classification, model='damo/cv_convnext-base_image-classification_garbage')
    dailylife_classification = pipeline(Tasks.image_classification, model='damo/cv_vit-base_image-classification_Dailylife-labels')
    general_recognition = pipeline(Tasks.general_recognition, model='damo/cv_resnest101_general_recognition')

    '''object detection'''
    object_detection = pipeline(Tasks.image_object_detection,model='damo/cv_tinynas_object-detection_damoyolo')
    #head_detection = pipeline(Tasks.domain_specific_object_detection, model='damo/cv_tinynas_head-detection_damoyolo')
    #facemask_detection = pipeline(Tasks.domain_specific_object_detection, model='damo/cv_tinynas_object-detection_damoyolo_facemask')
    phone_detection = pipeline(Tasks.domain_specific_object_detection, model='damo/cv_tinynas_object-detection_damoyolo_phone')

    # 初始化global result_c和global result_d变量，用于存储模型推理的结果（loop3）
    result_c = {'scores': [0,0,0,0,0], 'labels': ['','','','','']}
    result_d = {'scores': [], 'labels': [], 'boxes': []}

    stop = False

    '''keyboard_callback initialization'''
    # 初始化global model_sel变量，用于存储模型选择的标识符（keyboard_callback）
    model_sel = 'o'

    # 创建按键监听线程，按下任何按键时（包括长按），都会调用keyboard_callback，并传入键盘事件作为参量
    #skeyboard.on_press(keyboard_callback)

    # 初始化控制线程正常退出的标志位
    exit_flag = False

    # 创建线程
    t1 = threading.Thread(target=loop1)
    t2 = threading.Thread(target=loop2)
    t3 = threading.Thread(target=loop3)
    # 启动线程
```

The object detection program **esp32-cam.py** consists of **three main loop loops**, which are responsible for **storing the input video stream as an image file**, opencv preprocessing (**displaying the video stream window and the selected model**) and **calling the model for inference** respectively.

```python
def loop1():
    '''esp32cam image mjpeg'''
    global img
    r = requests.get(url + 'cam.mjpeg', stream=True)
    print('test')

    if r.status_code == 200:
        bytes = builtins.bytes()
        for chunk in r.iter_content(chunk_size=1024):
            if chunk:  # 确保获取到的块不为空
                bytes += chunk

            a = bytes.find(b'\xff\xd8')
            b = bytes.find(b'\xff\xd9')

            if a != -1 and b != -1:
                # print('find jpg')
                jpg = bytes[a:b + 2]
                bytes = bytes[b + 2:]

                # 确保jpg不为空
                if len(jpg) > 0:
                    img = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8), cv2.IMREAD_COLOR)
                    if img is None:
                        print("Image decoding failed, created an empty image.")
                else:
                    print("No valid JPEG data found.")

            if exit_flag:
                break
    else:
        print("Received unexpected status code {}".format(r.status_code))
    print('loop1 exit normally')
```

Loop1: **Get video stream** from the URL and **store as picture**.

```python
def loop2():
    '''opencv processing'''
    while not exit_flag:
        img_puttext = img.copy()
        result_c_puttext = result_c.copy()
        result_d_puttext = result_d.copy()
        if model_sel == 'g' or model_sel == 'd' or model_sel == 'r':
            '''image classification'''
            if model_sel == 'g':
                img_puttext = cv2AddChineseText(img_puttext, '垃圾分类', (4, 6), (0, 255, 0), 20)
            elif model_sel == 'd':
                img_puttext = cv2AddChineseText(img_puttext, '日常用品', (4, 6), (0, 255, 0), 20)
            elif model_sel == 'r':
                img_puttext = cv2AddChineseText(img_puttext, '万物识别', (4, 6), (0, 255, 0), 20)
            for i in range(len(result_c_puttext['labels'])):
                text = str(result_c_puttext['labels'][i]) + ' ' + str(result_c_puttext['scores'][i])
                img_puttext = cv2AddChineseText(img_puttext, text, (4, 26 + 20 * i), (0, 255, 0), 20)
        elif model_sel == 'o' or model_sel == 'h' or model_sel == 'f' or model_sel == 'p':
            '''object detection'''
            if model_sel == 'o':
                img_puttext = cv2AddChineseText(img_puttext, '通用目标检测', (4, 6), (0, 255, 0), 20)
            elif model_sel == 'h':
                img_puttext = cv2AddChineseText(img_puttext, '人头检测', (4, 6), (0, 255, 0), 20)
            elif model_sel == 'f':
                img_puttext = cv2AddChineseText(img_puttext, '口罩检测', (4, 6), (0, 255, 0), 20)
            elif model_sel == 'p':
                img_puttext = cv2AddChineseText(img_puttext, '手机检测', (4, 6), (0, 255, 0), 20)
            try:
                img_puttext = vis_det_img(img_puttext, result_d_puttext)
            except cv2.error:
                pass
        cv2.imshow('esp32cam-test',img_puttext)
        if ord('q')==cv2.waitKey(10):
            break
    cv2.destroyAllWindows()
    print('loop2 exit normally')
```

Loop2: Set video stream **window** and **label selected model**.

```python
def loop3():
    '''modelscope processing'''
    global result_c
    global result_d
    global stop
    while not exit_flag:
        if model_sel == 'g' or model_sel == 'd' or model_sel == 'r':
            '''image classification'''
            if model_sel == 'g':
                result_c = garbage_classification(img)
            elif model_sel == 'd':
                result_c = dailylife_classification(img)
            elif model_sel == 'r':
                result_c = general_recognition(img)
            print(result_c)
        elif model_sel == 'o' or model_sel == 'h' or model_sel == 'f' or model_sel == 'p':
            '''object detection'''
            if model_sel == 'o':
                result_d = object_detection(img)
            elif model_sel == 'h':
                result_d = head_detection(img)
            elif model_sel == 'f':
                result_d = facemask_detection(img)
            elif model_sel == 'p':
                result_d = phone_detection(img)
            print(result_d)
            if result_d['labels'] == ['bottle']and result_d['boxes'][0][2]-result_d['boxes'][0][0]>80:
                # 发送信号"nn"到服务器
                try:
                    # 创建一个 TCP/IP socket
                    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

                    # 连接到服务器
                    server_address = ('192.168.137.217', 8080)  # 使用服务器地址和端口
                    sock.connect(server_address)

                    # 发送 HTTP 请求
                    tt = "ss"

                    sock.sendall(tt.encode())
                    time.sleep(0.1)
```

Loop3: Pass the image to the model for **inference**, and **send backward command** when the size of returned labelled box is above the threshold.

4.3.6 control.py

```python
import socket
import threading
import tkinter as tk
import keyboard
import time

class TCPClient:
    def __init__(self, host, port):
        self.server_address = (host, port)
        self.sock = None
        self.is_running = False
        self.keys_down = set()  # 存储当前按下的键

    def connect(self):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.connect(self.server_address)
        self.is_running = True
        print("连接到服务器:", self.server_address)
        self.send_key_status()

    def send_key_status(self):
        while self.is_running:
            if self.keys_down:
                keys_list = list(self.keys_down)
                allowed_keys = {'w', 's', 'a', 'd'}

                # 过滤出有效按键
                filtered_keys = [key for key in keys_list if key in allowed_keys]

                if len(filtered_keys) == 1:
                    # 如果只按下一个允许的键，发送该键的两次
                    key_to_send = filtered_keys[0] * 2  # 例如 'w' 发送 'ww'
                elif len(filtered_keys) == 2:
                    # 检查是否是无效组合
                    if {'w', 's'}.issubset(filtered_keys) or {'a', 'd'}.issubset(filtered_keys):
```

```python
class TCPClient:
    def send_key_status(self):
                if len(filtered_keys) == 1:
                    # 如果只按下一个允许的键，发送该键的两次
                    key_to_send = filtered_keys[0] * 2  # 例如 'w' 发送 'ww'
                elif len(filtered_keys) == 2:
                    # 检查是否是无效组合
                    if {'w', 's'}.issubset(filtered_keys) or {'a', 'd'}.issubset(filtered_keys):
                        key_to_send = 'n' *2  # 发送0表示无效组合
                    else:
                        key_to_send = ''.join(filtered_keys)  # 发送有效组合的字符，如 'wa'
                else:
                    key_to_send = 'n' *2  # 如果没有有效按键，则发送0
            else:
                key_to_send = 'n' *2  # 如果没有按键，则发送0

            self.sock.sendall(str(key_to_send).encode())
            time.sleep(0.1)  # 每100毫秒发送一次

    def close(self):
        self.is_running = False
        if self.sock:
            self.sock.close()
            print("断开与服务器的连接")

    def on_key_event(self, e):
        if e.event_type == keyboard.KEY_DOWN:
            self.keys_down.add(e.name)  # 添加按下的键
        elif e.event_type == keyboard.KEY_UP:
            self.keys_down.discard(e.name)  # 移除抬起的键

```

The control.py program creates a socket connection to the esp8266 on the cart and constantly listen to keyboard input and sends direction commands.

5. Testing

5.1 Testing Scope

- **Functional Testing:** Verify each functional requirement is implemented correctly.
- **Non-Functional Testing:** Assess performance, reliability, usability, scalability, and maintainability.

5.2 Testing Environment

- **Hardware Setup:** Remote control car, Wi-Fi module, PC for control.
- **Software Setup:** Control application, serial port listener, under same LAN network.
- **Testing Locations:** Enough plain ground (indoor) to evaluate connectivity and mobility.

5.3 Test Cases and Schedule

Requirement	**Test Case**	**Timing**
Wi-Fi Connectivity	1. Test connection within LAN network 2. Verify the availability of transmitting signals	Days 4-6 (Sprint 2)
Remote Control	1. Test movement commands (forward, backward, left, right and combinations) 2. Verify minimal latency in real-time control	Days 7-8 (Sprint 3)
Power Management	Observe and record the output of motors and camera under different battery levels	Days 10-12 (Sprint 4)

Camera	1. Observe the accuracy of detecting object 2. Test the usability of video stream transmitting within a period of time	Days 13-15 (Sprint 5)
Performance	Measure command latency	Days 16-18 (Sprint 6)
Reliability	Test connection stability	Days 19-21 (Sprint 7)
Usability	Evaluate the integral running results in all aspects	Days 22-24 (Sprint 8)
Scalability	Assess system capability for future feature integration	Ongoing during development
Maintainability	1. Review code for modularity and documentation 2. Test diagnostic tools and logs for troubleshooting	Ongoing during development

5.4 Testing Procedures

- **Functional Testing:**
  - Execute movement commands and log actions variance.
  - Test connection stability.
- **Non-Functional Testing:**
  - Measure latency during control commands.
  - Conduct range tests to ensure connectivity.
- **Power Management:**
  - Observe the usability under low battery conditions.
- **Performance and Reliability:**
  - Run tests over extended periods to monitor stability and performance.

5.5 Data Collection and Analysis

- **Data Collection:** Log all test results, performance metrics, and bug reports.
- **Analysis:** Evaluate data to identify issues, track performance with corresponding requirements, and improve the programs.

5.6 Risk Management

- **Identified Risks:**
  - Network disconnection, insufficient power supply, turbulence of camera.
- **Mitigation Strategies:**
  - Use PC as hotspot server to ensure stability of LAN connection between the components.
  - Make sure the power bank has plenty of power capacity.
  - Add support structure for the camera.

6. Problems and Proposed Solutions

6.1 Hardware

6.1.1 Motors

- **Problem:** The motors have very large tolerances and inputting the exact same PWM value and direction, they can vary very much in spinning speed.
- **Solution:** Adjust according to the observation during actual testing

6.1.2 Wheels

- **Problem**: The shape of the wheel is not a standard circle, but shaped like ellipse, while its center is not on the axle, and the wheel is not perpendicular to the axle, so the speed of left and right motor will change by times even if the values remain the same.
- **Solution**: Include acceleration and angles as input (not implemented since time limit)

6.1.3 Speed Control

- **Problem:** There is also the problem of the car going too fast, even if the current speed is already the slowest case. Continuing to reduce the duty cycle can’t drive the motors since not enough voltage, but even the lower bound outputs an excessive speed level.
- **Solution**: Adjust the PWM value linearly, but this relied on accelerometers and angle sensors to sense the deflection to the left and right while making dynamic adjustments to direction and speed. But it was not realized since time limit.

6.2 Assemble and Exterior Design

6.2.1 Hardware Secure

- **Problem:** Base plate’s mounting clips are uneven resulting difficulty to secure upper hardware
- **Solution:** Styrofoam padding is used to elevate the hardware and increase friction, and Velcro fasteners are used for modular mounting and dismounting of the upper development board and power adapter board.

6.2.2 Camera view

- **Problem:** The wiring of the development board and power adapter board may affect the vision of camera.
- **Solution:** The camera height was increased by sticking on vertical Styrofoam particle.

6.2.3 Camera Swaying

- **Problem:** When the camera is elevated, the lens will shake violently when the vehicle moves, thus affecting the accuracy of identification
- **Solution:** Include support structure to minimize the swaying of camera.

6.2.4 Center of Gravity

- **Problem:** Vehicle hardware center of gravity is at backward instead of center resulting in a tendency to buckle at high speed.
- **Solution:** Lower the moving speed.

6.2.5 Insufficient pin and power supply

- **Problem**: Insufficient pin and power supply
- **Solution**: Split wires through soldering

6.3 Wi-Fi Connection

6.3.1 IP changes

- **Problem:** When connecting to a different hotspot, the IP will change accordingly, causing the connection failure.
- **Solution:** Use the PC as a hotspot server, and check the IP of all devices within the subnet on the PC, and then the esp866 and camera WiFi are turned on as server for client's connection. So that there is no need to use a fixed IP on the hardware side, even if connecting to a new hotspot. It is only required to modify the script on the control terminal.

6.3.2 Can’t check IP

- **Problem**: Xiaomi phone can't check the IP of the device connected to hotspot, which causes difficulties in communication within the LAN.
- **Solution**: Use the PC as a hotspot server.

6.3.3 Frequent disconnection

- **Problem:** When the computer connects to the cell phone hotspot, the connection will fail frequently, which affects our development and debugging progress.
- **Solution:** Use the PC as a hotspot server.

6.4 Signal Transmit

6.4.1 Block

- **Problem:** When sending commands to the vehicle, there is no specific protocol added to the esp8266 Wi-Fi, so sending post commands from the pc console to the cart will result in blocking.
- **Solution:** Adopt the traditional transfer method, first create a socket, then bind, and only transfer the information after the connection is established.

6.4.2 Delay

- **Problem:** Low command sending frequency results in control delays and inability to quickly update the vehicle's motion status.
- **Solution**: Increase signal sending frequency

6.5 Object Detection

6.5.1 Excessive Classes

- **Problem**: This recognition model was not trained by our own, and it contains too many classes, resulting in poor accuracy when recognizing objects of a specific class and easy to lose label tracking.
- **Solution**: Reducing the speed of vehicles approaching objects.

6.5.2 Camera Viewing Angle

- **Problem:** The camera's viewing angle is very narrow, resulting in obstacles being more likely to go out of view and lose the target.
- **Solution:** Caused by the hardware's own characteristics, and therefore cannot be resolved.

7. Possible Improvements

7.1 Can’t move straight

Since the ground resistance is not uniform and the tolerances of the wheels and motors are very large, the use of accelerometers and angle sensors mpu6050 is very necessary. In the future we may add them to achieve perfect straight line forward and backward movement as well as a linear start.

7.2 Frequent plugging of power

Forget to add the switch during the soldering process, so each time of powering needs to plug and unplug the power wires. To avoid frequent plugging and unplugging, we will add the switch to control the power supply.

7.3 Low accuracy of object detection

The accuracy of the model for specific objects is not satisfying, subsequently if we have time, we will collect data and retrain this model in order to achieve an improvement in its accuracy for specific obstacle recognition.

7.4 Tremble of camera

Camera shake can also seriously affect the accuracy of image recognition, so we consider adding anti-shake devices to the camera module.