

## 5.8 WIFI camera control car with speed and servo

### 1. Learning goal:

Learn about the serial communication method, analyze the communication protocol, use the APP to control the movement of the car, and the state machine switches the movement state of the car.

Added adjust control speed and servo on the basis of the previous lesson,

### 2. Experimental phenomena:

After we open the power of switch, the mobile phone connects to the WIFI signal of the car, then opens the APP to control the car forward, backward, left translation, right translation, spin left, spin right, left front, left rear, right front and right rear. And we can controls the car acceleration, deceleration and sliding slider to control the servo angle.

### 3. Analysis:

3.1 PCA9685 controls the principle of the servo:

The duration of a high level in a pulse with a period of 20ms determines the angle of rotation of the servo.

The correspondence of the 180°servo is as follows: 0.5ms-->0°, 1ms-->45°, 1.5ms-->90°, 2ms-->135°, 2.5ms --> 180°.

3.2 We can send data to WIFI camera by mobile APP.

3.2.1 Please use your mobile phone scan QR code on the cover of the Instruction manual, download and install the YahboomRobot APP.

**Note: During installation, If you find any prompts on your phone (for example: location permissions of your phone). You must select "Allow".**

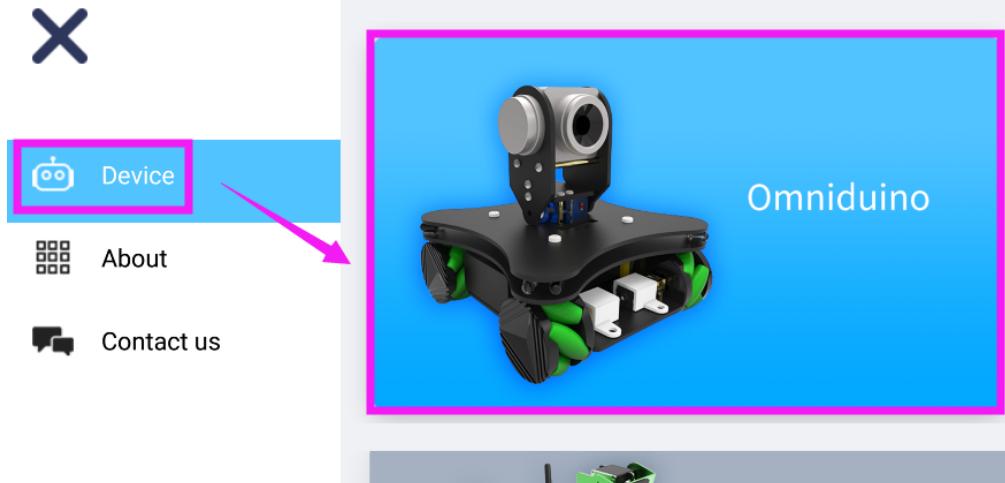
3.2.2 Turn on the phone to search for the WIFI signal and connect the Yahboom\_WIFI, no need password, just click on the connection. Some mobile phones may promptly disconnect without network data, and you can click again to connect.



### 3.2.3 Open the YahboomRobot APP.

**!Note: When you open it for the first time, it will be prompted to get some permissions, you must select “Allow”.**

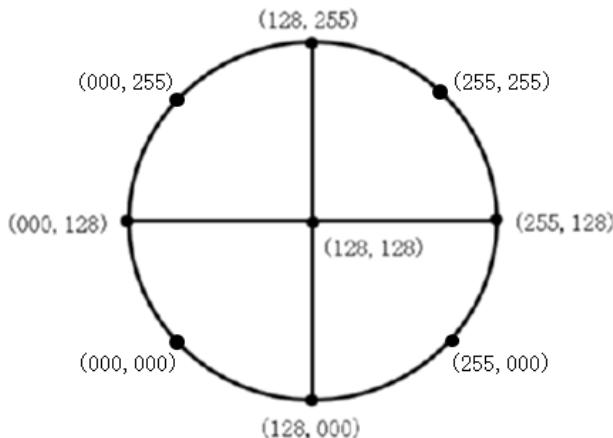
### 3.2.4 Click on the menu icon in the upper left corner of the APP In the device column, you need to select Omniduino. As shown below.



### 3.2.5 Click on the picture to enter the remote control interface of the omniduino car.



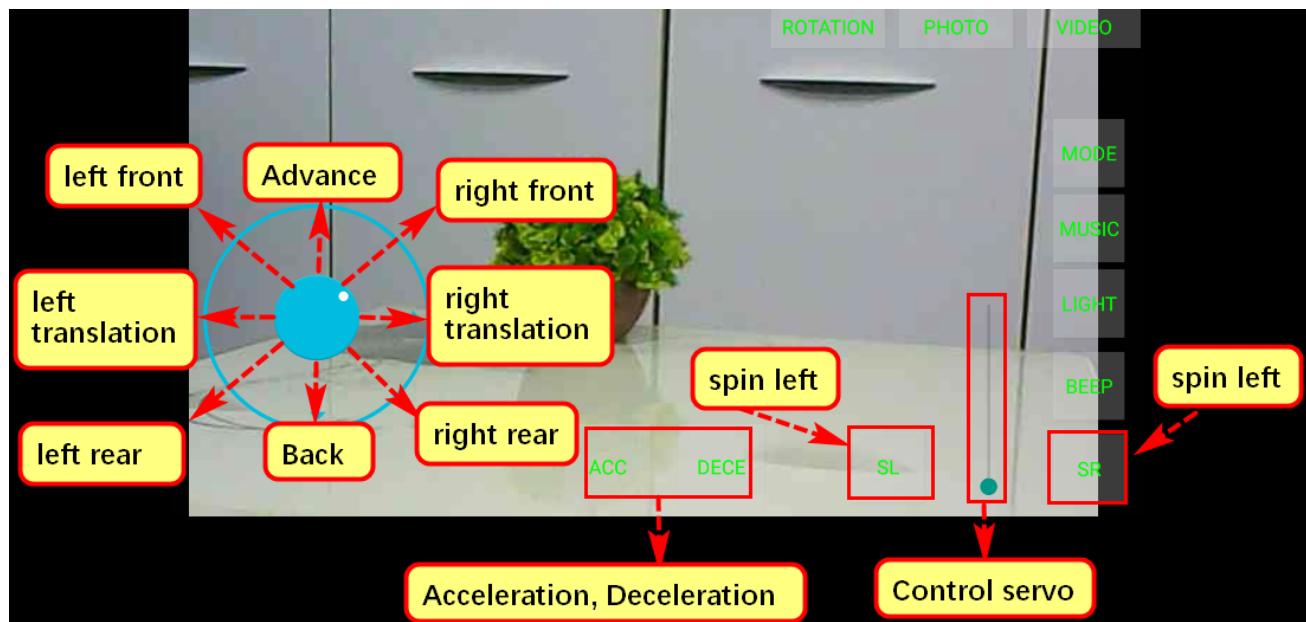
### 3.3 When the APP simulation joystick is sent, the X axis gradually increases from left to right (0~255), the Y axis gradually increases from down to up (0~255), and the middle position is (128, 128).



3.3 “**SL**” button in APP, press send: \$Spin, 11#, release send: \$Spin, 00#.

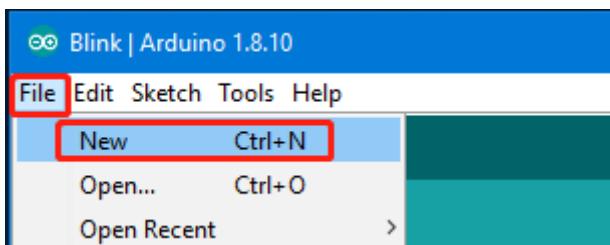
3.4 “**SR**” click button in APP, press send: \$Spin, 21#, release send: \$Spin, 00#.

As shown below:

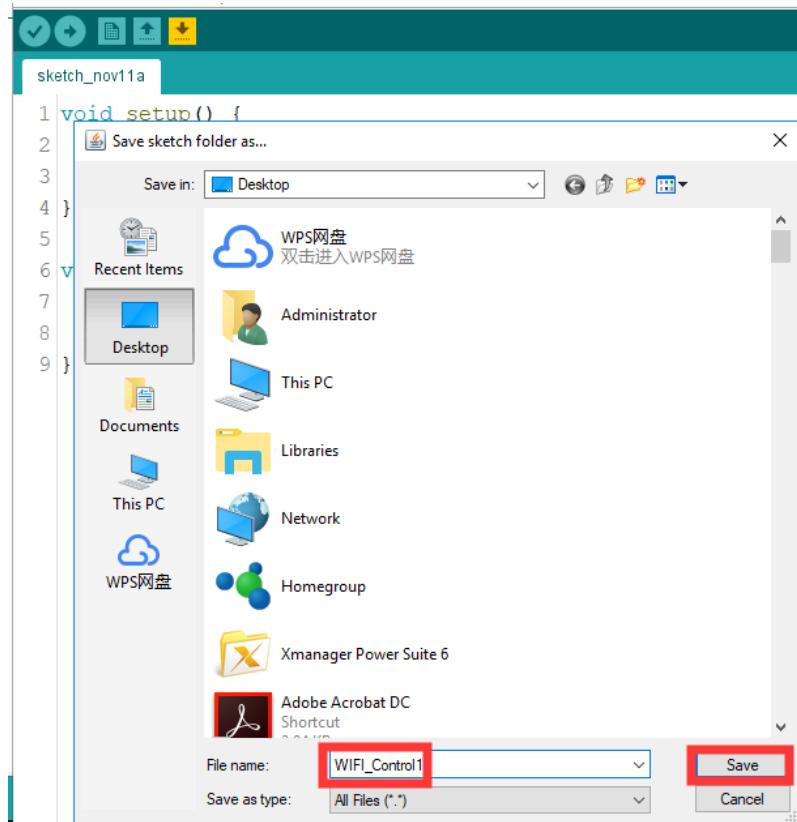


#### 4. Create new project

4.1 Click 【File】-->【New】.



4.2 Press **Ctrl+S** to save and rename WIFI\_Control02. As shown below.



4.3 We can see that there is a Serial folder with **WIFI\_Control02.ino** on the computer desktop.

4.4 We will **WIFI\_Control02.ino** as shown below.

```
void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeatedly:
}
```

The `setup()` function only runs once when the car is turned on or when the reset button is pressed, and the program for initializing the relevant content can be written;

The `loop()` function is the main loop function of the car and most of the data processing and logic processing are done in this function.

## 5. Programming

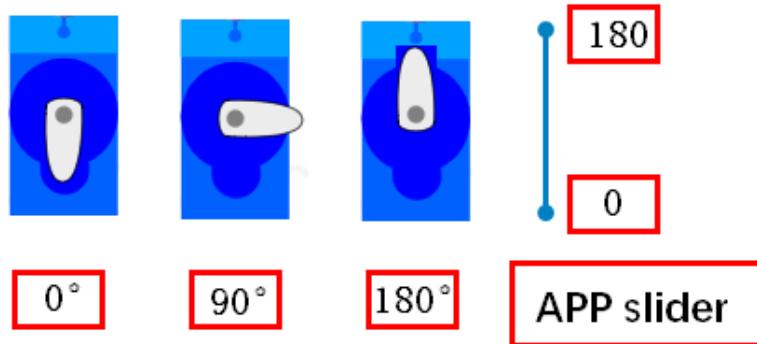
5.1 Create a new servo function to control the servo angle. We know that 0.5ms~2.5ms can control the servo from 0° to 180°. It is considered that some servos will shake at 0° and 180°, so it recommended range is 0.6ms~2.4ms. From the hardware manual, the servo is connected to pin 7 of the PCA9685.

```

void servo(int degree)
{
    long us = (degree * 1800 / 180 + 600); // 0.6 ~ 2.4
    long pwmvalue = us * 4096 / 20000;      // 50hz: 20,000 us
    pwm.setPWM(7, 0, pwmvalue);
}

```

5.2 Increase the servo\_code\_parse() function to analyze the servo control code. Since the slider of the APP adjusts the steering angle is gradually increased from down to up (0~180), and we need to use the map function to limit the angle and swap the control direction.



```

// control servo
if (InputString.indexOf("Servo") >= 0)
{
    // Extract servo angle values and limit angles
    servo_degree = InputString.substring(7, 10).toInt();
    servo_degree = map(servo_degree, 0, 180, 150, 30);
    servo(servo_degree); //set servo angle
}

```

5.3 The serial\_data\_parse() function adds parsing acceleration and deceleration button codes. Since the car speed cannot be set too fast, it cannot be set too slowly. It is recommended that the speed range is 40~160.

```

//Parses the APP ADD, DECE button protocol(eg:Accelerate):$Speed,11#
if (InputString.indexOf("Speed") >= 0)
{
    if (InputString[8] == '1')
    {
        if (InputString[7] == '1') //Accelerate, add 20 each time
        {
            CarSpeedControl += 20;
            if (CarSpeedControl >= 160)
                CarSpeedControl = 160;
        }
        else if (InputString[7] == '2') //Slow down, minus 20 each time
        {
            CarSpeedControl -= 20;
            if (CarSpeedControl <= 40)
                CarSpeedControl = 40;
        }
    }
}

```

## 6. Compiling and downloading code

6.1 After the code is written, press Ctrl+S to save, then click the “√” button to compile. If there is no problem, click “→” to upload (the car must be connected to the computer via the USB cable).

```

File Edit Sketch Tools Help
[Icons]
LED
10 */
11 //Define LED light(D9)pin
12 #define LED_PIN 5
13
14 void setup() {
15     // put your setup code here, to run once:
16     // set LED pin to output mode
17     pinMode(LED_PIN, OUTPUT);
18 }
19
20 void loop() {
21     // put your main code here, to run repeatedly:
22     digitalWrite(LED_PIN, LOW);      //LED is on
23     delay(500);
24     digitalWrite(LED_PIN, HIGH);    //LED is off
25     delay(500);
26 }

```

6.2 If there is an error like the following, it means that the library file is missing. Please copy the library file provided by the omniduino omnibus to the library file directory compiled by arduinolIDE.

please refer to **【3.Development Environment Construction】----【3.4 Add additional library files】**

```

Adafruit_PWMSServoDriver.h: No such file or directory

CarRun:2:10: error: Adafruit_PWMSServoDriver.h: No such file or directory
#include <Adafruit_PWMSServoDriver.h>
^
compilation terminated.

exit status 1
Adafruit_PWMSServoDriver.h: No such file or directory

```

6.3 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.

```

An error occurred while uploading the sketch
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
An error occurred while uploading the sketch
avrduude: ser_open(): can't open device "\.\COM32": The system can

```

**Solution:** Open the device manager to see if there is a serial port with CH340 tag. If not, please restart the Omniduino car, then, re-plug the USB cable or replace a USB cable; If there is a serial port number, we need to close the other serial port or assistant software, avoid serial port occupation, and then re-select the serial port to ArduinoIDE 【Tool】-->【Port】. 6.4 After clicking the upload button, the upload is always displayed, but it can't be uploaded successfully for a long time.

```
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions. Copy error messages
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 4 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 5 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 6 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 7 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 8 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 9 of 10: not in sync: resp=0xec
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 10 of 10: not in sync: resp=0xec
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#upload for suggestions.
```

Because the uploading program and the WIFI camera communication is realized through the serial port, when the serial port is occupied by the WIFI camera, and the program cannot be uploaded.

**Solution:**

- ①Unplug the USB cable, turn off the power of the car, wait for the D2 indicator to go out.
- ②Then, plug in the USB data cable. At this time, your mobile phone should not connect the WiFi signal of the car.
- ③You can upload the program to the car according to the normal steps.
- ④After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. The corresponding experimental phenomenon will appear.  
*(Tip: If you upload APP control program. After the program is successfully uploaded, unplug the USB data cable, open the power switch of the car. Mobile phone connect the car to the WIFI signal, and then open the APP to control.)*