

TSCINBUNY
ESP8266 WIFI CAR V2.0
ZYC0057

V2.02.308.31

教程简介

本教程包括如下图所示的文件内容帮助你深入学习小车套件和编程知识：

 1_Get_start-准备	2023/8/15 15:45	文件夹
 2_Arduino_Code-代码文件	2023/8/15 14:25	文件夹
 3_Graphical-图形化	2023/8/15 14:20	文件夹

“1_Get_start”：该文件夹存放机器人组装指南和必要的软件环境文件等。为了准确且快速的完成组装，请务必详细查看并按照手册进行组装。同时，实现程序功能的前提是要正确的创建软件环境，请查看该文件夹下的 PDF 文件。

“2_Arduino_Code”：该文件夹用来存放 Arduino 代码文件，将代码上传到控制板使用，每份代码功能独立，每次上传一份代码，后面上传的代码会覆盖前面的代码；

“3_Graphical”：该文件夹存放 mind+软件的图形化编程代码和指导文件。

完成小车组装和环境创建后跟着本教程逐步完成智能小车程序烧录和实现不同的功能吧！

对于具有较好编程基础的爱好者可以直接跳到第 4 节内容快速实现小车 WiFi 遥控功能。

目录

1.让汽车动起来.....	5
1.1 描述.....	5
1.2 电机驱动模块 L9110S.....	5
1.3 驱动模块 L9110S 如何工作?	5
1.4 代码烧录.....	7
1.5 代码分析.....	9
2.跟随小车	11
2.1 描述.....	11
2.2 超声波	11
2.3 代码分析.....	12
3.避障小车	14
3.1 描述.....	14
3.2 舵机.....	14
3.2 代码分析	14
4. ESP8266 WiFi Car	18
4.1 描述.....	18

4.2WiFi 遥控原理	18
4.3 代码上传	18
4.4 代码分析	19

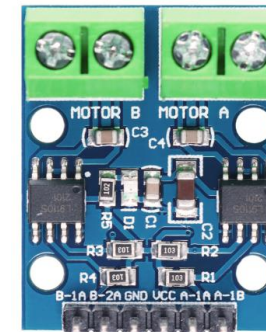
1.让汽车动起来

1.1 描述

本节内容主要了解到电机和电机驱动模块原理，掌握如何通过编程控制小车机器人完成前进、后退、左转弯、右转弯行走。

1.2 电机驱动模块 L9110S

- 1、双 L9110S 芯片
- 2、电机工作电压在 2.5v-12v 之间，最大工作电流为 0.8A;
- 3、可同时驱动两台直流电机，或一台 4 线 2 相步进电机。
- 4、PCB 尺寸:2.8cm *2.1cm 超小体积，适合组装
- 5、带固定安装孔，直径:3mm



1.3 驱动模块 L9110S 如何工作?

通过 B-1A(PIN 7)/B-2A(PIN 6)/A-1A(PIN 8)/A-1B(PIN 4)输出高低电平或输出 pwm 来控制电机正、反旋转。这里 pwm 数值范围（0~255），例如 analogWrite(PIN,180)。

State of motion	B-1A (PIN 7)	B-2A (PIN 6)	A-1A (PIN8)	A-1B (PIN 4)
Forward	HIGH	LOW	LOW	HIGH
Backward	LOW	HIGH	HIGH	LOW
TurnLeft	HIGH	LOW	HIGH	LOW
TurnRight	LOW	HIGH	LOW	HIGH

Stop	LOW	LOW	LOW	LOW
------	-----	-----	-----	-----

设置速度变量 speedCar 和前进函数 Forward()

```

13
14  int speedCar =180;

40  void Forward(){
41      digitalWrite(IN_1, LOW);
42      analogWrite(IN_2, speedCar+10);//当小车前进或后退走不直时可以修改参数来调节：speedCa
43      analogWrite(IN_3, speedCar);
44      digitalWrite(IN_4, LOW);
45  }

```

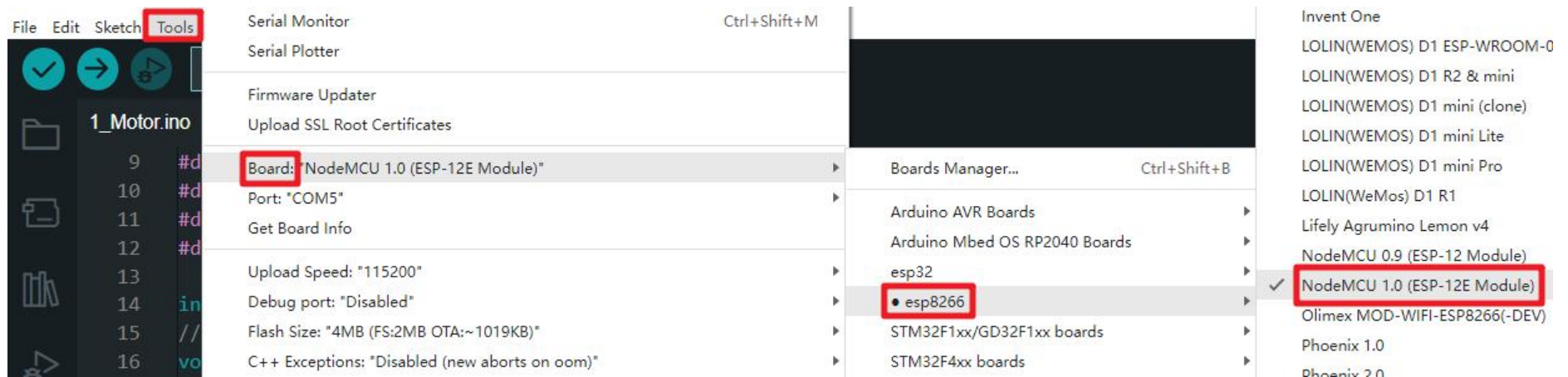
当小车前进或后退走不直时可以修改参数来调节：speedCar+10 相比原来的速度增加了 10 个单位，可以让小车往右调节一些，当(IN_2,speedCar-10)即可让小车往左调节；同理设置 IN_2 或 IN_3 的参数值都可以调节小车的速度差。

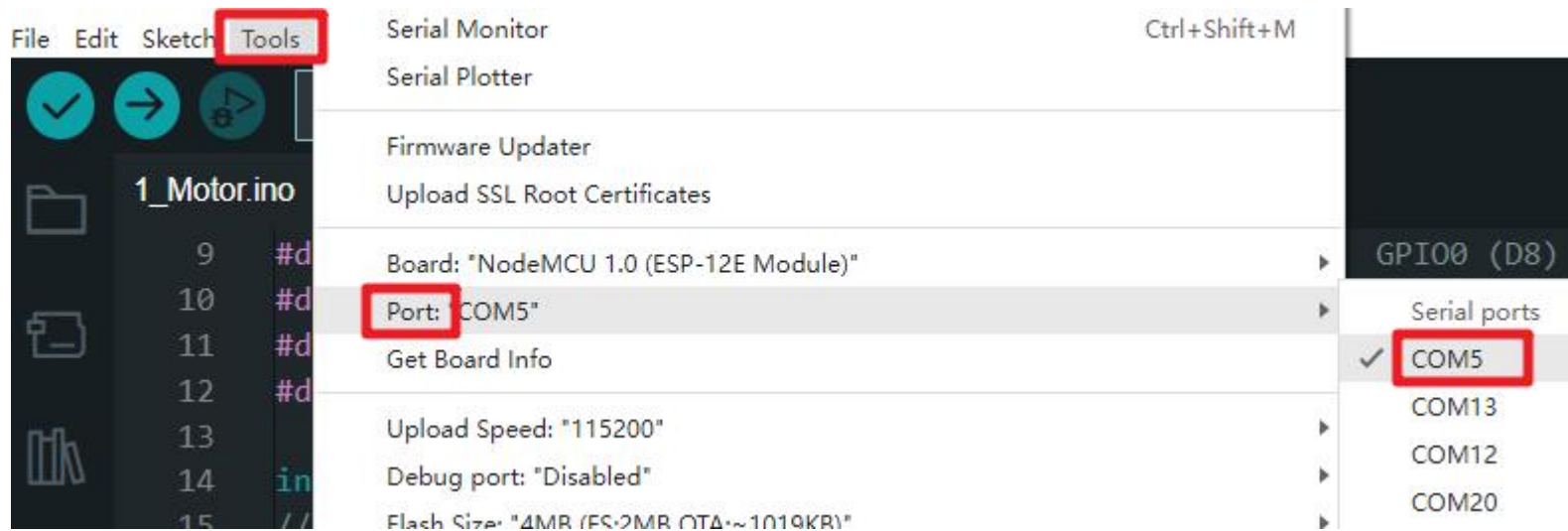
1.4 代码烧录

打开代码文件（文件夹路径：2_Arduino_Code\1_Motor\1_Motor.ino）

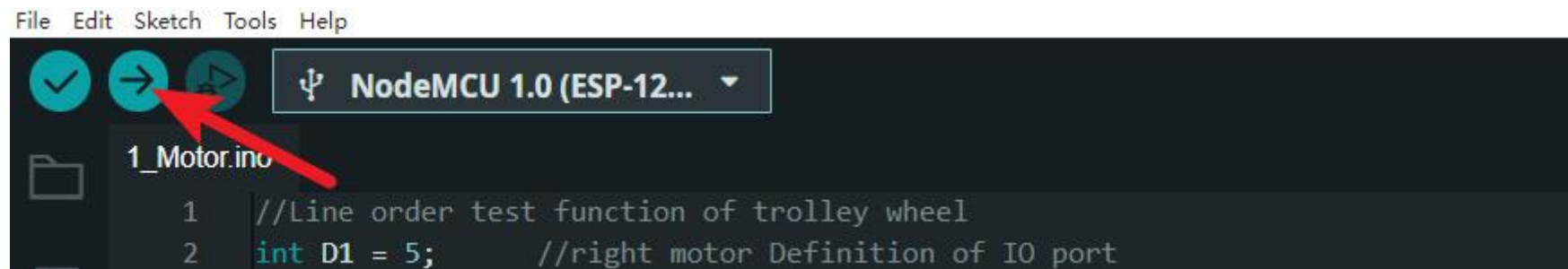
1_Motor	2023/8/15 14:25	文件夹
2_Follow	2023/8/15 14:24	文件夹
3_Obstacle_avoidance	2023/8/15 14:25	文件夹

使用 USB 连接线将控制板连接到计算机并检查 Board Type 和 Serial Port 设置正确。





注意:这里的板卡类型选择 **NodeMCU1.0** 和串口选择 **COM5**。实际上串行端口显示会不一样, 尽管这里选择了 COM5, 但它可能是您计算机上的 COM3 或 COM4。然后, 点击上传按钮等待上传完成即可。



代码烧录成功后能看到小车执行前进、后退、左转和右转运动。

1.5 代码分析

电机引脚定义

```

9  #define IN_1  0           // L9110S B-1A motors Left      GPIO0 (D8)
10 #define IN_2  4           // L9110S B-1B motors Left      GPIO4 (D4)
11 #define IN_3  13          // L9110S A-1A motors Right     GPIO13(D7)
12 #define IN_4  12          // L9110S A-2A motors Right     GPIO12(D6)

```

设置波特率 115200，电机引脚为输出，`setup()`只在程序启动时执行一次。

```

16 void setup()
17 {
18     Serial.begin(115200);
19     pinMode(IN_1, OUTPUT);
20     pinMode(IN_2, OUTPUT);
21     pinMode(IN_3, OUTPUT);
22     pinMode(IN_4, OUTPUT);
23 }

```

设置左右电机速度变量初始值 180，当小车前进走不直时可以增大或减小其中一边速度来调节走直线。

```

13
14 int speedCar =180;

```

根据前面电机驱动模块分析那样，创建小车各个方向移动的函数，包括前进函数 `forward()`，后退函数 `Backward()`、左转、右转和停止函数（代码已折叠）。

```
43 void Forward()  
44 > {...  
49 }  
50 void Backward()  
51 > {...  
56 }  
57 void TurnLeft()  
58 > {...  
63 }  
64 void TurnRight()  
65 > {...  
70 }  
71 void Stop()  
72 > {...  
77 }
```

loop()循环函数，调用小车各个方向移动的函数，通过 delay()函数控制执行时长。

```
26 void loop()  
27 {  
28     Forward();           //Car forward  
29     delay(2000);  
30     Backward();          //Car back  
31     delay(2000);  
32     TurnLeft();          //Car turnleft  
33     delay(2000);  
34     TurnRight();         //Car turnright  
35     delay(2000);  
36     Stop();              //Car stop  
37     delay(1000);         //delayed
```

如果整个过程都能实现证明电机驱动接线已经正确，接下来可以进行后面项目任务。

2.跟随小车

2.1 描述

超声波测距是一种非常有用且广泛使用的测量方式，本节内容主要了解超声波模块工作原理、掌握超声波电路的连接和通过编程实现小车的跟随功能。

2.2 超声波

超声波是如何工作的？

- 1.发射器（trig pin）发送信号：高频声音；
- 2.当信号碰到物体时，它会被反射；
- 3.接收器（echo pin）：接收从它反射回来的信号。

超声波测距的方法有很多种，该系统在超声波测量中的原理是：检测来自超声波发射器的超声波通过气体介质到接收器的传输时间，将此时间乘以空气中的声速，得出声音传播的距离。

超声波发射器向一定方向发射超声波，同时 MCU 开始计时，超声波在空中发射，途中遇到障碍物立即返回，超声波接收器接收到反射波立即停止计时。根据定时器记录的时间 T ，可以计算出发射点到障碍物的距离（ S ）。

$$\text{公式: } S = V * T/2$$

四个因素限制了超声波系统的最大可测量距离：超声波的幅度、反射器的纹理、反射和入射声波之间的角度以及接收换能器的灵敏度。接收换能器直接接收声脉冲的能力将决定最小可测量距离。

2.3 代码分析

打开代码文件（文件夹路径：2_Arduino_Code\2_Follow\2_Follow.ino）

在上一节内容的基础上增加超声波发送（2）和接收（15）引脚的定义，超声波测量出的距离变量 distance 和 cm.

```
1  #define Trig 2          //GPIO2 (D9)
2  #define Echo 15         //GPIO15 (D10)
3  float distance;
4  float cm;
```

用来获取超声波检测距离的函数 GetDistance()

```
40 float GetDistance()
41 { // Send a low short pulse to Trig to trigger the ranging
42   digitalWrite(Trig, LOW); //Send a low level to Trig
43   delayMicroseconds(2);
44   digitalWrite(Trig, HIGH);
45   delayMicroseconds(10);
46   digitalWrite(Trig, LOW);
47   distance = pulseIn(Echo, HIGH) / 58.00;
48   Serial.print("Distance = ");
49   Serial.println(distance); //The serial output distance is converted into cm
50   return distance;
```

在主函数中调用获取超声波距离的函数并保存到变量，再进行数值比较。

```
void loop()
{
  cm = GetDistance();
  if(cm > 10 && cm < 25){
```

跟随小车的原理：

当超声波检测前方物体距离小于 5 时后退，在 10~25 之间时小车前进跟随，否则其他情况就停止。共有前进、后退和停止三种状态。

```
29     if(cm <5){  
30         Backward();  
31     }  
32     else if(cm > 10 && cm <25){  
33         Forward();  
34     }  
35     else{  
36         Stop();  
37     }
```

3.避障小车

3.1 描述

上节内容中我们已经了解和学习了超声波模块的相关知识，本节内容增加舵机控制超声波旋转功能，并掌握避障小车的原理，最后通过编程实现避障功能。

3.2 舵机



舵机（伺服电机）控制脉冲信号周期为 20MS 脉冲宽度调制信号（PWM），脉冲宽度从 0.5ms 到 2.5ms，对应的转向位置从 0 到 180 度，线性变化。

舵机内部有参考电路，产生周期为 20ms、宽度为 1.5ms 的脉冲信号。有比较器，将外部信号与参考信号进行比较，判断方向和大小，从而产生电机转动信号。

3.2 代码分析

打开代码文件（文件夹路径：2_Arduino_Code\3_Obstacle_avoidance\3_Obstacle_avoidance.ino）

1_Motor	2023/8/15 14:25	文件夹
2_Follow	2023/8/15 14:24	文件夹
3_Obstacle_avoidance	2023/8/15 14:25	文件夹

避障原理:

判断超声波与前方障碍的距离(变量 `cm`)小于设定值 (`< set_dis`) 时, 让小车停下 200ms, 也就是调用停止函数 `Stop()`, 舵机带动超声波左转并获取左侧超声波与障碍之间距离保存到变量 `leftDis`, 同理超声波再右转获取距离保存到变量 `rightDis`, 最后舵机回到正前方。通过比较左右侧距离大小, 让小车往距离大的一侧行驶, 如果最大距离都小于 10cm, 那么小车先后退再转弯, 从而避开障碍。

新建带参数的避障函数 `avoidance(int set_dis)`, 首先将超声波检测前方的距离保存到内部变量 `cm`

```
98 void avoidance(int set_dis) {  
99     myservo.write(90); //Steering engine back to center  
100     //Obtain the distance between the cart and the obstacle and store it in cm  
101     cm = GetDistance();
```

判断遇到前方障碍 ($cm < \text{set_dis}$), 将超声波获取到的左、右侧距离保存到 leftDis 和 rightDis。

```

104   if (cm <= set_dis) {
105       Stop();                //Stop the car
106       delay(200);
107       myservo.write(140);
108       leftDis = GetDistance(); // Measure the distance between the left obstacle and the car, and store the
109       delay(200);
110       myservo.write(90);
111       delay(100);
112       myservo.write(40);
113       rightDis = GetDistance(); // Record the range data on the right side of the trolley
114       delay(200);
115       myservo.write(90);

```

当右边距离更大时右转，如果最大空间距离小于 10 则先后退再右转

```

117   //The right is more distant from the obstacle than the Left
118   if (leftDis < rightDis) {
119       if (rightDis < 10) {
120           Backward();
121           delay(300);
122           TurnRight();
123           delay(200);
124       } else {
125           TurnRight();
126           delay(200);

```

同理，可以得到左转时的代码


```
130     else if (leftDis > rightDis) {  
131         if (leftDis < 10) {  
132             Backward();  
133             delay(300);  
134             TurnLeft();  
135             delay(200);  
136         } else {  
137             TurnLeft();  
138             delay(200);
```

最后在没有遇到障碍时保持直行。

```
141     } else {  
142         Forward();  
143     }
```

4. ESP8266 WiFi Car

4.1 描述

Esp8266 主控带有强大的 WiFi 功能，通过编程设置 AP 模式即可释放 WiFi 热点，用另一台设备来连接该热点实现遥控功能。

4.2 WiFi 遥控原理

带有 esp8266 主控的小车在上传相应代码后能释放出 WiFi 热点，例如本项目 WiFi 热点名称为：ZY ESP8266 Car，用带有指令功能 APP 的设备连接该热点后即可发送遥控指令，小车接收到指令后执行相应的动作。在连接的过程中，你需要输入一个 IP 地址以完成设备连接，并且一旦连接成功后就意味着你的网络服务已经切换到 esp8266 的热点而不是你的 GPRS 网络，你处于仅能遥控小车而不能上网的情况。

关于 APP 的安装在不同设备上会有不同的安装方式，注意查看文件夹“4_APP”中的文档说明。

4.3 代码上传

打开代码文件（路径：2_Arduino_Code\4_esp8266_Car\4_esp8266_Car.ino）

在 ArduinoIDE 中选择正确的板卡类型 NodeMCU1.0 和串口 COM5。

点击上传按钮“upload”等待上传完成。

4.4 代码分析

导入所需库文件

```
7  #include <ESP8266WiFi.h>
8  #include <WiFiClient.h>
9  #include <ESP8266WebServer.h>
```

设置接收指令的字符变量 `String command` 和两个速度变量 `speedCar`（一大一小形成速度差转弯，其中 1.5 是相差倍数关系，可以根据实际来调节）

```
16 String command;           //String to store app command state.
17 int speedCar = 200;        // 0~255.
18 int speed_low = speedCar/1.5;
```

设置 WiFi 热点名称变量和实例化网络服务，端口为 80

```
20 const char* ssid = "ZY ESP8266 Car"; //name
21 ESP8266WebServer server(80);
```

设置 WiFi 模式为 AP 模式和 WiFi 名称设置

```
30 // Connecting WiFi
31 WiFi.mode(WIFI_AP);
32 WiFi.softAP(ssid);
```

链接成功之后打印 IP 地址，在串口监视器可查看到 IP 地址。

```

34  IPAddress myIP = WiFi.softAPIP();
35  Serial.print("AP IP address: ");
36  Serial.println(myIP);

```

启动 Web 服务端及接收返回 html 文件函数 HTTP_handleRoot

```

38  // Starting WEB-server
39  server.on ( "/", HTTP_handleRoot );
40  server.onNotFound ( HTTP_handleRoot );
41  server.begin();

```

```

131 void HTTP_handleRoot(void) {
132     if( server.hasArg("State") ){
133         Serial.println(server.arg("State"));
134     }
135     server.send ( 200, "text/html", "" );
136     delay(1);
137 }

```

在循环函数 loop()内执行服务交互，将获取到的指令状态值保存到变量 command

```

107 void loop() {
108     server.handleClient();
109     command = server.arg("State");

```

根据接收到的指令值执行相应的运动，不同字母代表不同的运动方向，0~9 代表不同的速度

```
110     if (command == "F") Forward();
111     else if (command == "B") Backward();
112     else if (command == "L") TurnLeft();
113     else if (command == "R") TurnRight();
114     else if (command == "I") goAheadRight();
115     else if (command == "G") goAheadLeft();
116     else if (command == "J") goBackRight();
117     else if (command == "H") goBackLeft();
118     else if (command == "0") speedCar = 165;
119     else if (command == "1") speedCar = 175;
120     else if (command == "2") speedCar = 185;
121     else if (command == "3") speedCar = 195;
122     else if (command == "4") speedCar = 205;
123     else if (command == "5") speedCar = 215;
124     else if (command == "6") speedCar = 225;
125     else if (command == "7") speedCar = 235;
126     else if (command == "8") speedCar = 245;
127     else if (command == "9") speedCar = 255;
128     else if (command == "S") Stop();
```