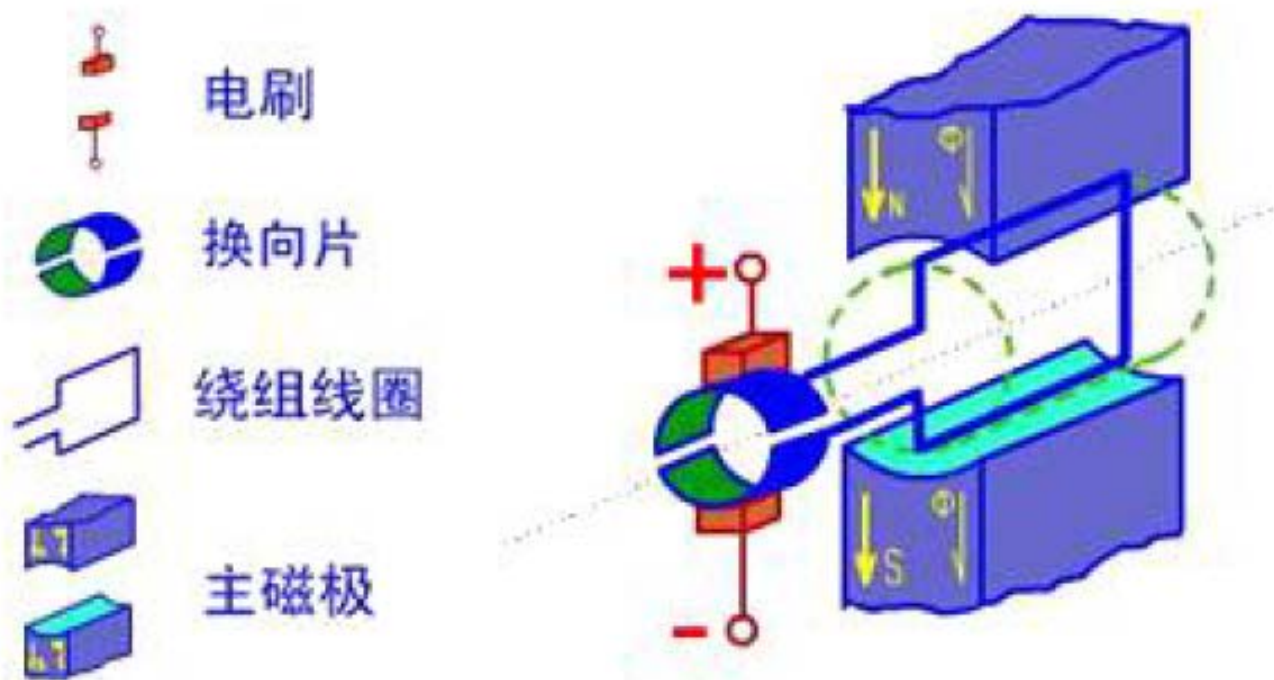




电机与运动控制

直流减速电机

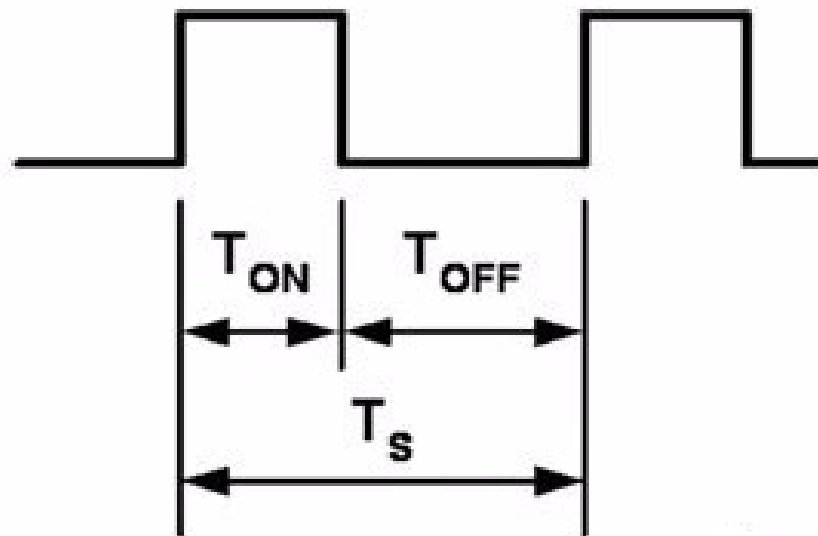
- 机器人的腿
- 直流电机+减速器
- 直流电机（铝壳）：
 - 直流通电后电机转动，极性决定转向，电压控制转速
- 减速器（黄色）：
 - 降低转速，增加扭矩



直流电机原理示意

电机控制

- 调向
 - 上正下负 正转
 - 下负上正 反转
- 调速
 - 改变电压
 - 如何在数字电路实现电压的平滑改变?
 - PWM脉冲宽度调制
 - PWM占空比



PWM波形示意

系统库函数
设置输出引脚
PWM占空比

要设定的引脚编号
(引脚编号旁有~的
可以输出PWM)

PWM占空比
0-255 -> 0%-100%

```
analogWrite(6,200);
```

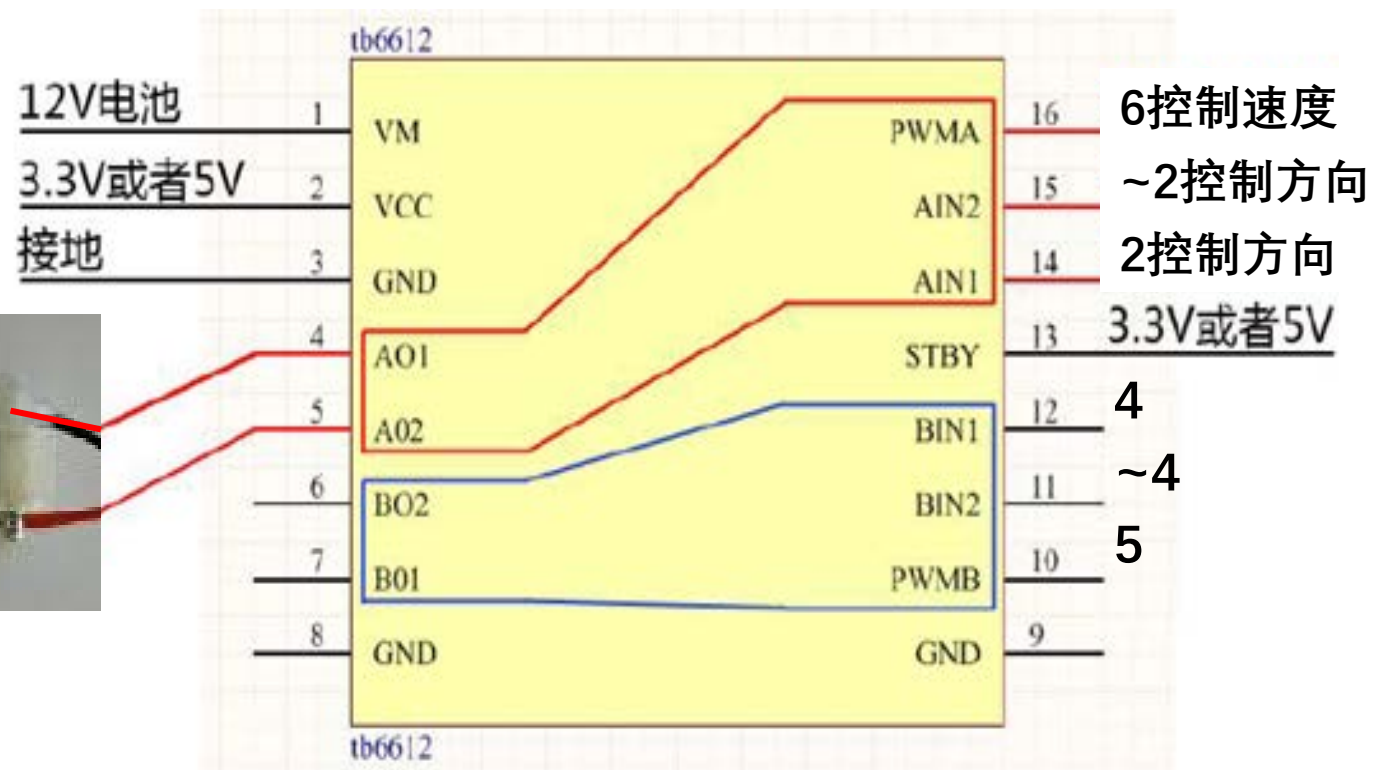
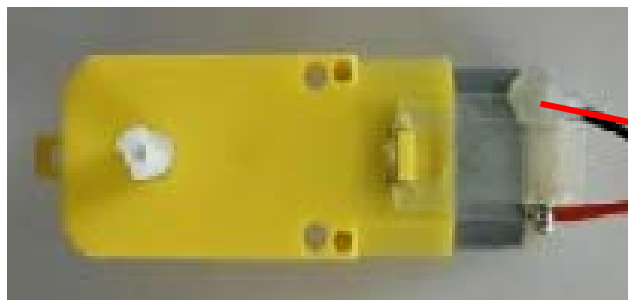


电机驱动

- 电机需要的电压、电流都远大于单片机IO口提供的电压和电流。需要一个电路，将单片机的控制信号转化为电机的驱动电流

- TB6612特性

- 最大100KHzPWM
- 5v 25mA → 15V 1.2A/3.6A
- 正反转(IN1,IN2为1,0或0,1)
- 过热保护
- 低压保护



单侧电机控制

- 根据硬件电路定义引脚

```
#define Right_motor_pwm 6 //右电机控速管脚 PWMB
#define Right_motor_go 2 //右电机前进管脚
#define Left_motor_pwm 5 //左电机控速管脚 PWMA
#define Left_motor_go 4 //左电机前进管脚
```

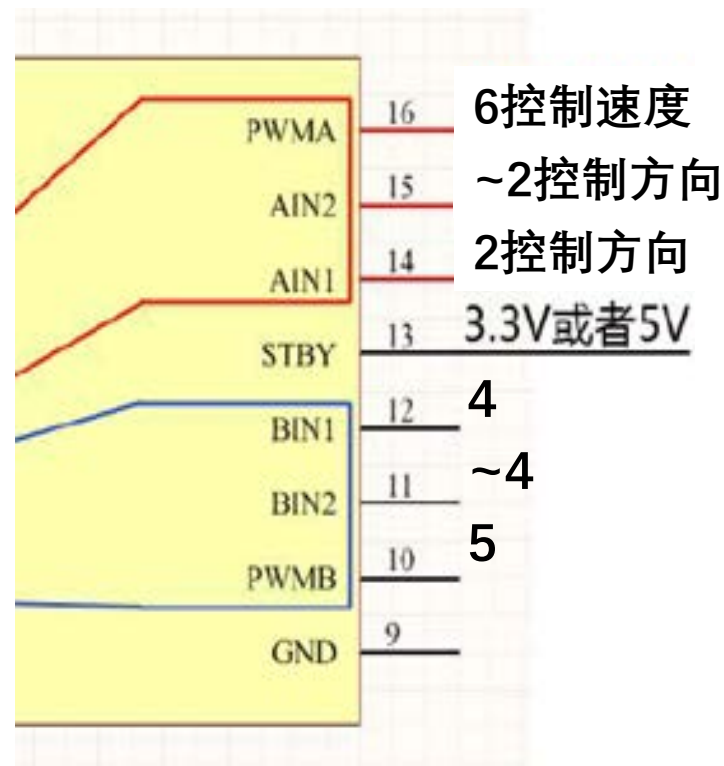
- 右前进，速度70

```
digitalWrite(Right_motor_go, LOW); //右电机前进使能
analogWrite(Right_motor_pwm, 70);
```

- 右后退，速度70

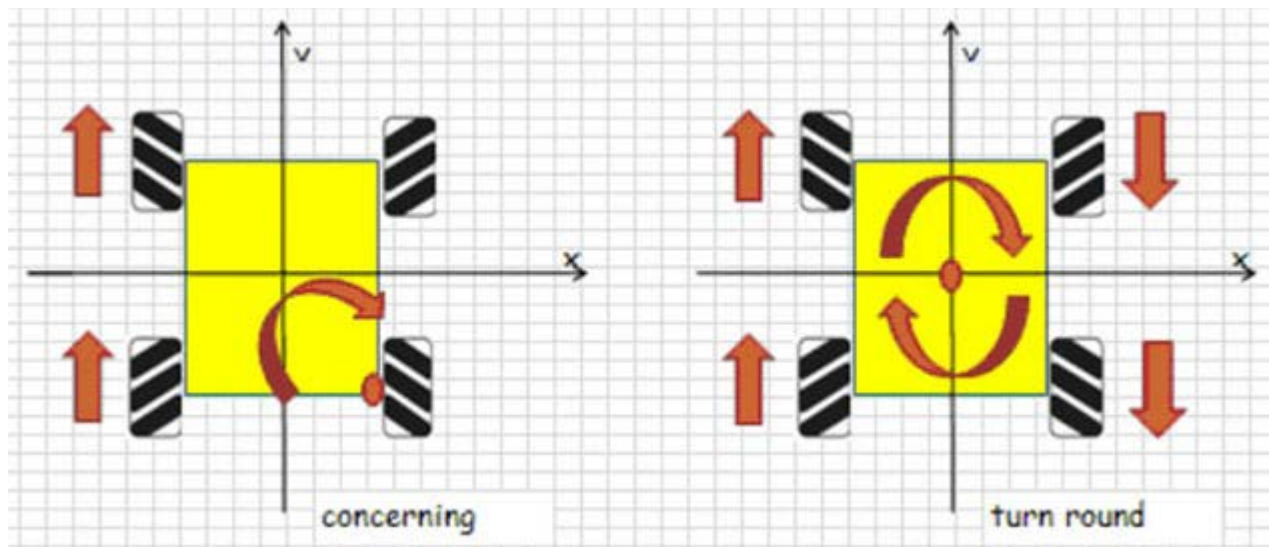
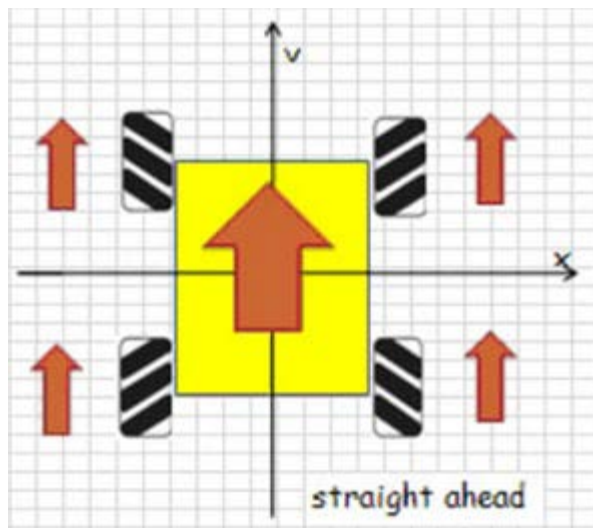
```
digitalWrite(Right_motor_go, ____); //右电机前进使能
analogWrite(Right_motor_pwm, 70);
```

- 使用这些引脚之前别忘了设置它们的输入输出模式
- 想一想，如果电机转动方向与我们希望的相反，会是什么原因，要怎么解决？



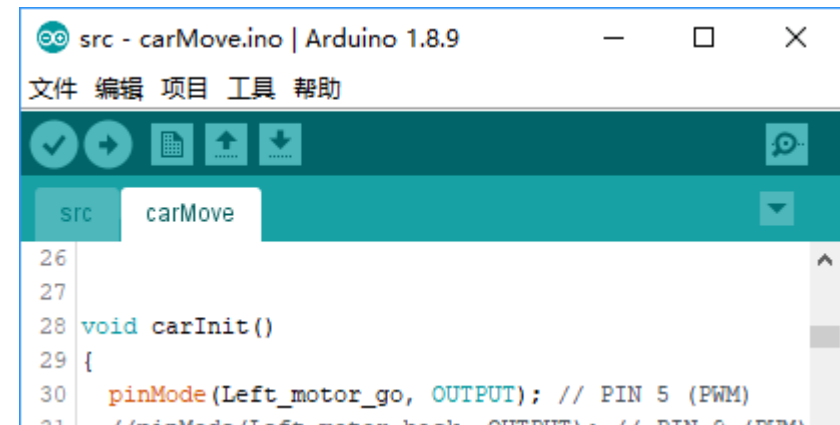
运动控制

- 四个轮子分为左右两组
- 前进
 - 左右正转
- 后退
 - 左右反转
- 左转
 - 左反右正
 - 左速 < 右速
- 右转
 - 左正右反
 - 左速 > 右速



模块化程序

- Arduino和C语言中允许将若干个函数、变量、宏等放置在其他文件中，我们可以利用这一特性将某一类用途的代码放在一个单独的文件中，封装成模块，模块间可以互相调用函数。
- carMove的范例代码中就将车辆运动控制的代码单独放进carMove.ino文件中
 - 主文件：src.ino
 - void setup()
 - void loop()
 - 小车运动模块： carMove.ino
 - 相关引脚宏定义
 - void carInit()//初始化
 - void carMove(int,int)//电机控制



不同的模块可以通过IDE中代码区上方的标签来选择

carMove模块

逻辑与，两侧条件都为真时结果才为真

```
//右电机前进管脚
#define Right_motor_go 2
//右电机控速管脚 PWMB
#define Right_motor_pwm 6
//左电机控速管脚 PWMA
#define Left_motor_pwm 5
//左电机前进管脚
#define Left_motor_go 4
void carInit()
{
    pinMode(Left_motor_go, OUTPUT);
    pinMode(Right_motor_go, OUTPUT);
    pinMode(Left_motor_pwm, ?);
    pinMode(Right_motor_pwm, ?);
}
```

```
void carMove(int leftSpeed, int rightSpeed)
{
    if (leftSpeed > 255)//如果用户给定速度大于255
        leftSpeed = 255;
    if (leftSpeed < -255)//如果用户给定速度小于-255
        leftSpeed = -255;
    if (leftSpeed < 10 && leftSpeed > -10)
        leftSpeed = 0;
    if (leftSpeed >= 0 && leftSpeed <= 255){
        digitalWrite(Left_motor_go, LOW); //左电机前进
        analogWrite(Left_motor_pwm, leftSpeed);
    }
    else if (leftSpeed < 0 && leftSpeed >= -255){
        leftSpeed = 0 - leftSpeed;
        digitalWrite(Left_motor_go, HIGH); //左电机后退
        analogWrite(Left_motor_pwm, leftSpeed);
    }
    //.....右侧电机控制
}
```


src主模块

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600); // 串口初始化, 设置波特率  
  carInit(); // 车辆电机初始化  
}
```

• 任务

- 以 10.2 的程序为基础, 完成对“?” 部分的内容替换, 并修改其他部分程序。让机器人可以走出一个正方形

```
void loop() {  
  Serial.println("Forward ");  
  carMove(?,?); // 左右侧轮前进, 车辆前进  
  delay(?); // 延时1s  
  Serial.println("Back ");  
  carMove(-?,-?); // 左右侧轮后退, 车辆后退  
  delay(1000);  
  Serial.println("Left ");  
  carMove(-?,?); // 左轮后退, 右轮前进, 车辆左转  
  delay(?);  
  Serial.println("Right ");  
  carMove(?,-?); // 左轮前进, 右轮后退, 车辆右转  
  delay(1000);  
  Serial.println("Stop ");  
  carMove(0,0); // 左轮0, 右轮0, 车辆停止  
  delay(1000);  
}
```