



# Part 4

## 控制系统的设计与实践



# 目 录

4.1

驱动小车移动

4.2

驱动炮塔移动

4.3

遥控小车移动

4.4

自动循迹小车

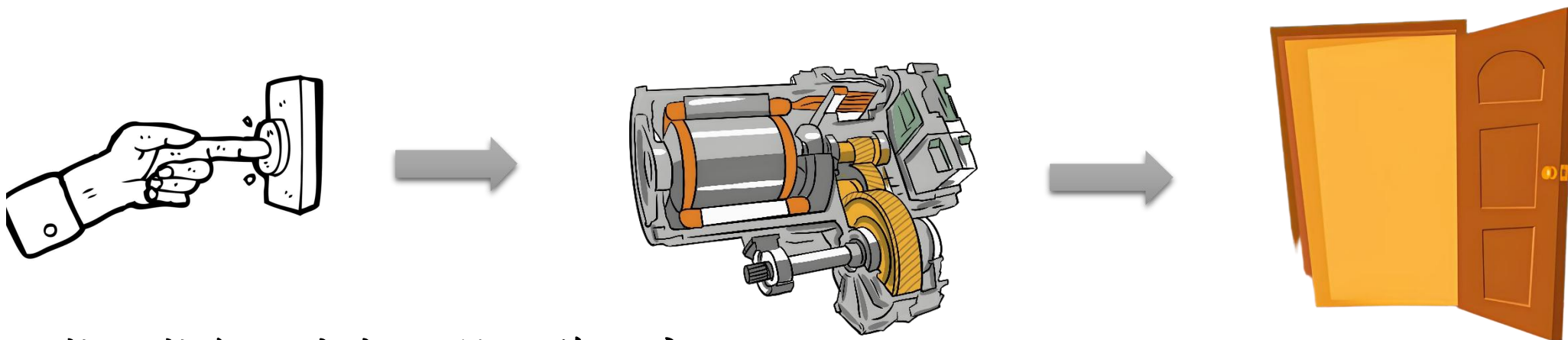


# 引入



## 开门

动作：按下按钮，电机带动门打开。



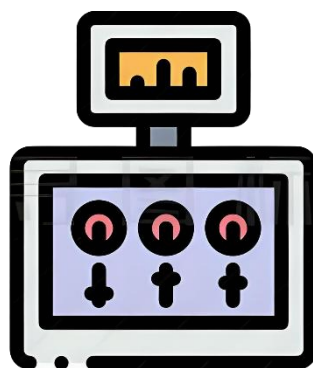
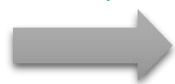
就按下按钮，电机开始工作而言：

物理：0V - 5V

LOW - HIGH

逻辑：0 - 1

信号



判断

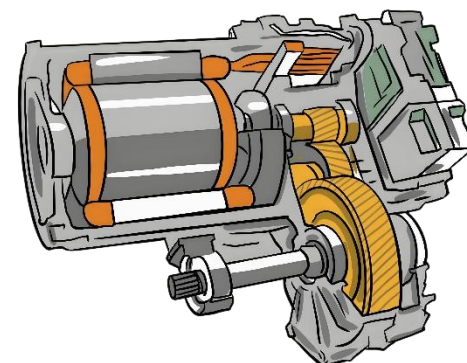
真

转动

假

停止

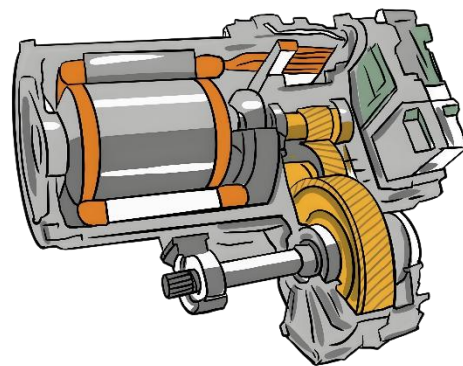
信号



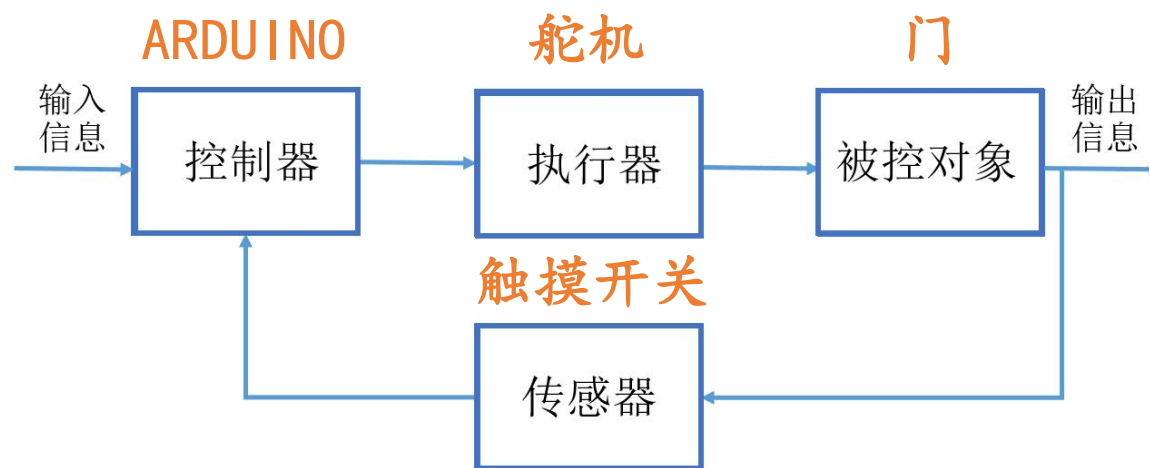


## 开门

**动作：**按下按钮，电机带动门打开。



逻辑框图及硬件组成







# 引入

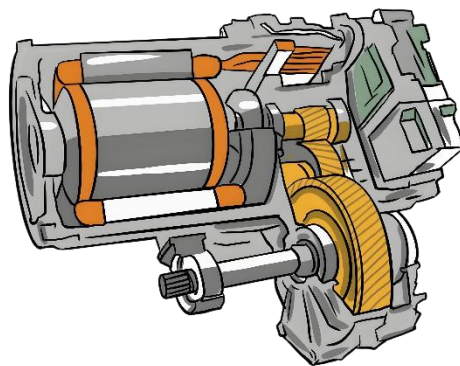


哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

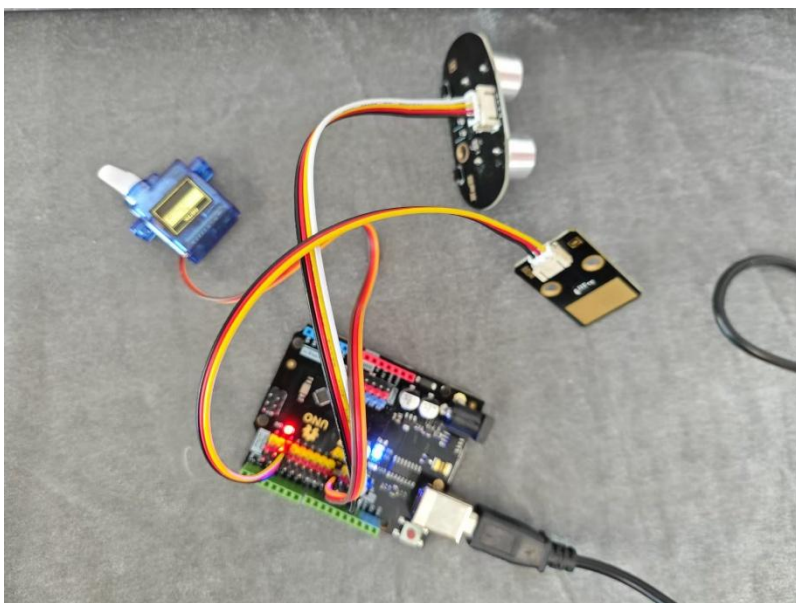


## 开门

**动作：**按下按钮，电机带动门打开。



硬件连接



**注意：**VCC，GND以及SIGNAL连线位置不要出错，特别是VCC。



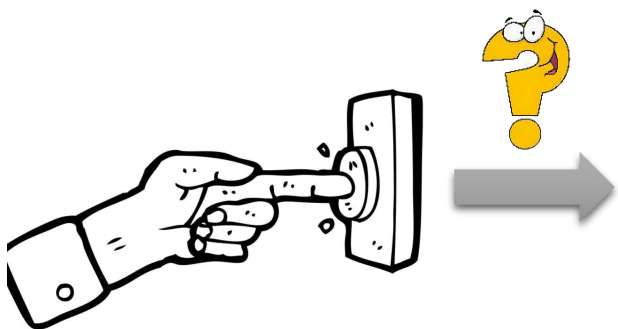
# 引入



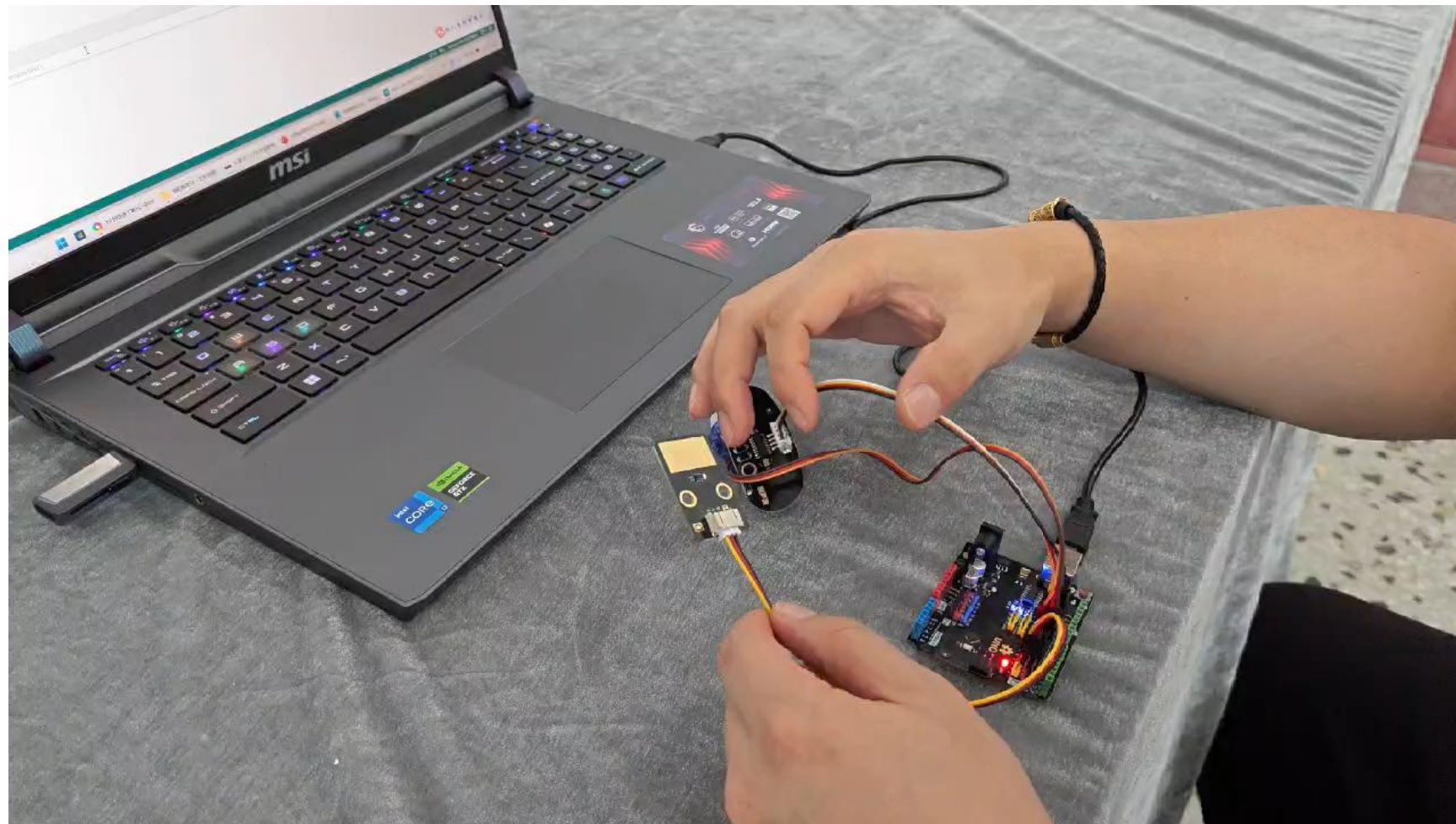
哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

## ➤ 开门

动作：按下按钮。



系统：1-0。





# 引入

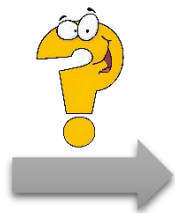


哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY



## 开门

**动作：**按下按钮，电机带动门打开。



```
void setup()
```

```
{
```

```
  Serial.begin(9600);  
  pinMode(2, INPUT);
```

```
}
```

主程序框架  
初始化+循环

初始化串口

设置2号端口为输入  
可调整信息

```
void loop()
```

```
{
```

```
  if (Serial.available())  
  {
```

```
    Serial.println(digitalRead(2));  
  }
```

```
}
```

如果串口有信息  
读取2号端口信息

获取输入信息程序







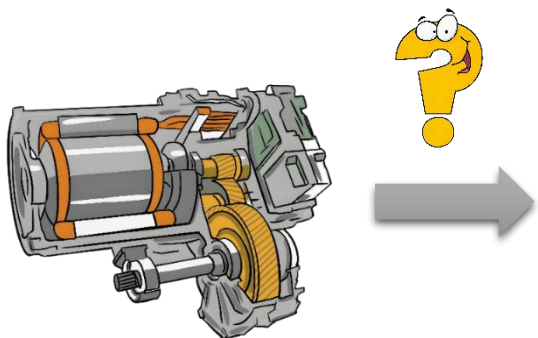
# 引入



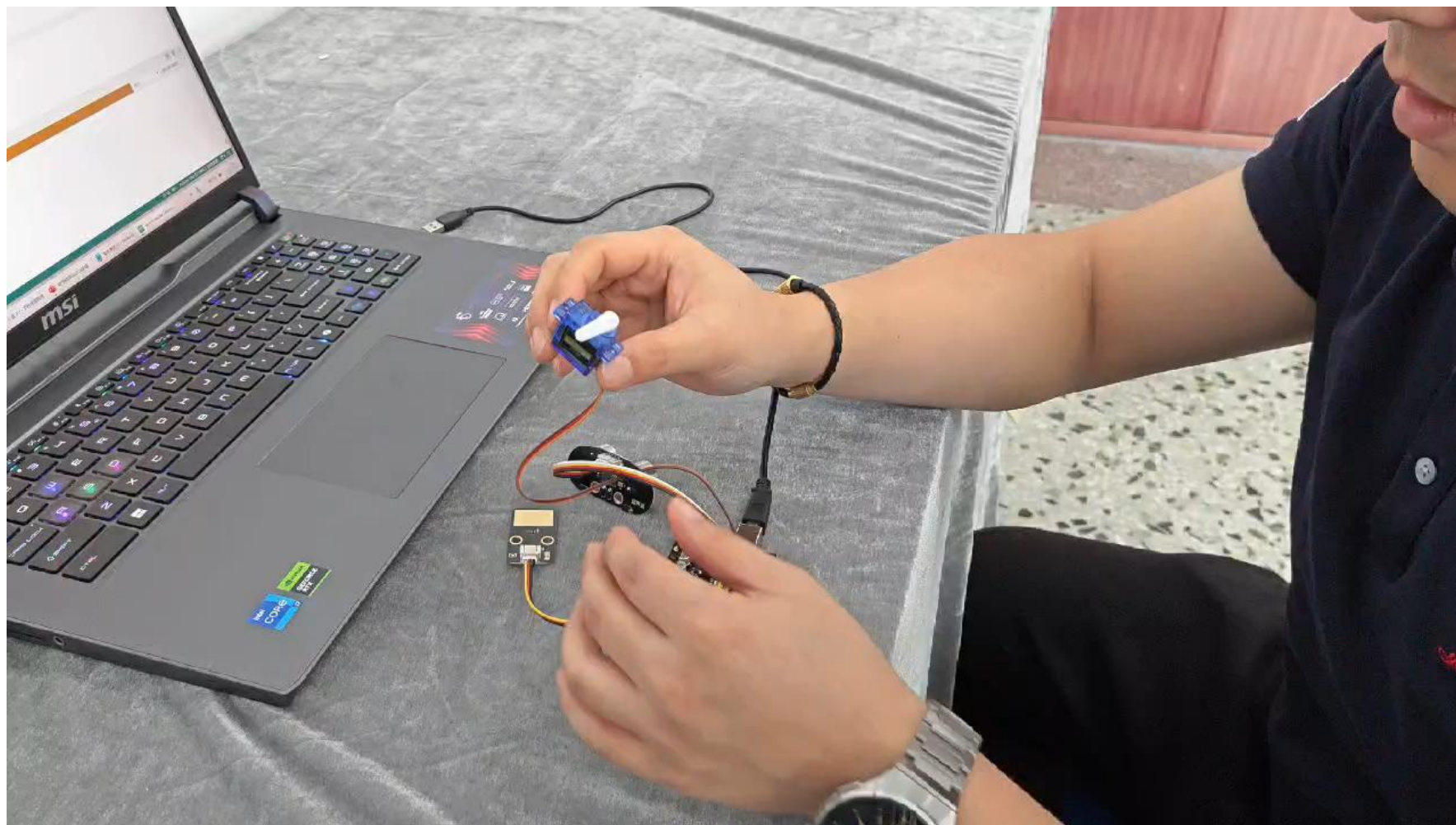
哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

## ➤ 开门

**动作：**电机带动门打开。



**系统：**舵机旋转指定角度。





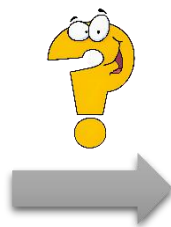
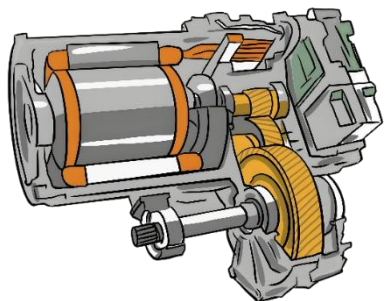


# 引入



## 开门

动作：按下按钮，电机带动门打开。



```
#include <Servo.h>
```

调用函数库

```
Servo servo_9;
```

```
void setup()
```

```
{  
  servo_9.attach(9);  
}
```

舵机在9号端口

```
void loop()
```

```
{  
  servo_9.write(0);
```

舵机旋转到0度

```
  delay(1000);
```

```
  servo_9.write(180);
```

舵机旋转到180度

```
  delay(3000);
```

延迟3秒

```
}
```

获取输出信息程序

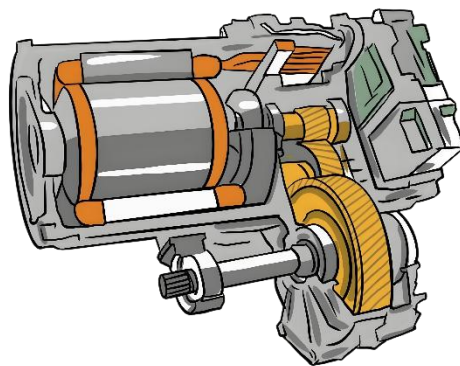


# 引入



## 开门

**动作：**按下按钮，电机带动门打开。



转换

按下按钮



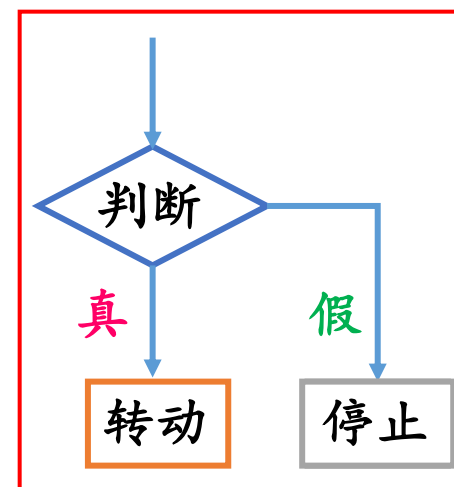
D2: 1-0



开 门



D9: 0° - 90°





## 自动开门<sup>1</sup>

检测到人来或  
人触摸开关，门自  
动打开。







## 开门

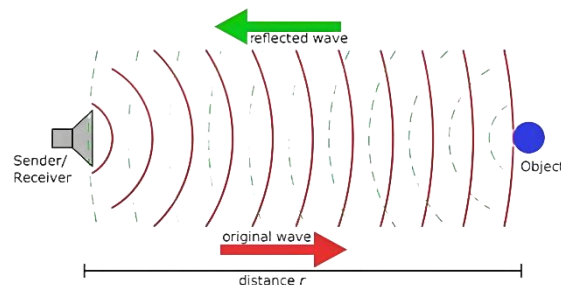
**动作：**检测到人接近到一定距离，电机带动门打开。

### 测试-程序

```
float checkdistance_12_13()
{
    digitalWrite(12, LOW);
    delayMicroseconds(2);
    digitalWrite(12, HIGH);
    delayMicroseconds(10);
    digitalWrite(12, LOW);
    float distance = pulseIn(13, HIGH) / 58.00;
    delay(10);
    return distance;
}

void setup(){
    Serial.begin(9600);
    pinMode(12, OUTPUT);
    pinMode(13, INPUT);
}

void loop(){
    //超声波测距\n超声波传感器trig接D2, Echo接D3\n串口输出超声波传感器检测的距离值
    Serial.println(String("距离:") + String(checkdistance_12_13()) + String("cm"));
    delay(1000);
}
```





# 引入

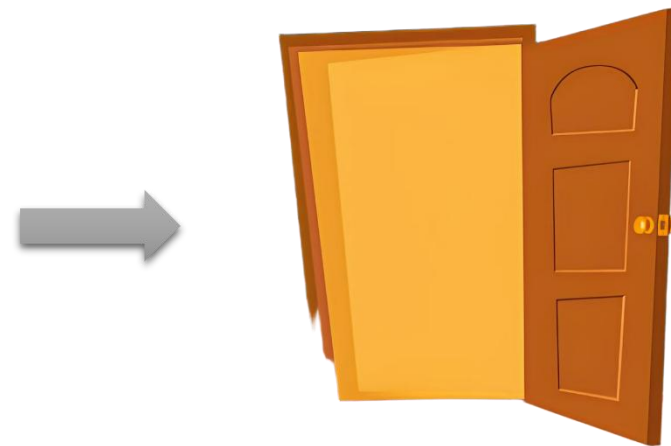
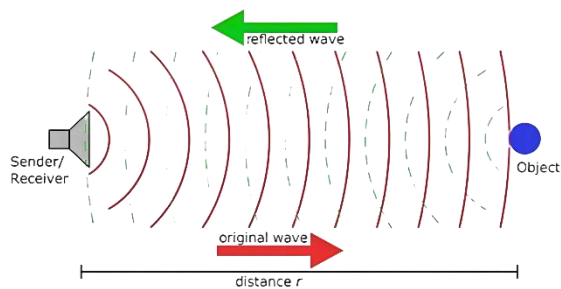
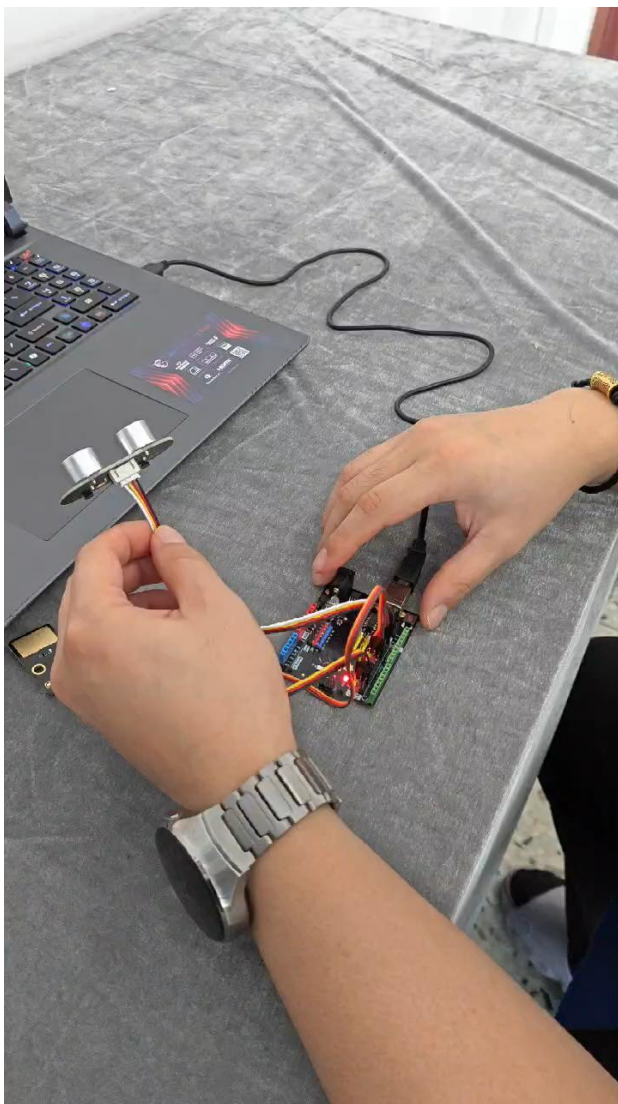


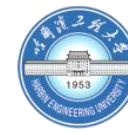
哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

## 开门

### 测试-演示

动作：检测到人接近到一定距离，电机带动门打开。





## 自动开门<sup>2</sup>

检测到有人接近到一定距离，门自动打开。







# 4.1 驱动小车移动



## 1) 明确目的及要求

对标任务：实现小车的前进、后退、左转、右转几个动作。



## 2) 工作逻辑分析



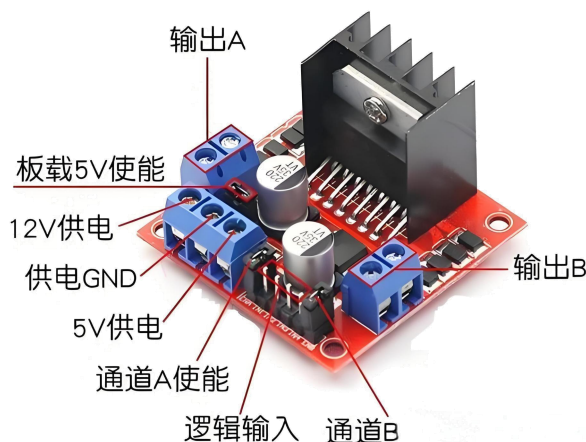
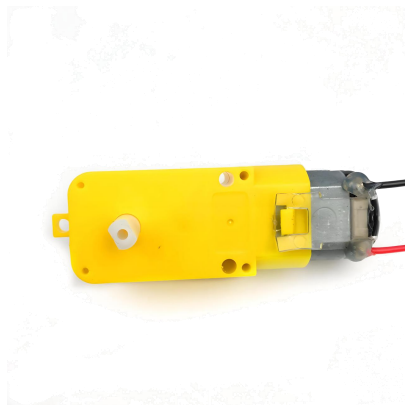


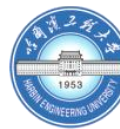
# 4.1驱动小车移动



## 3) 任务分解

- ① 硬件部分： ARDUINO;  
电机驱动模块+ 直流减速电机; } 管脚定义，性能参数是否匹配等!  
智能小车; → 明确被控对象结构特点，确保控制方案与机械部分相匹配。  
电路连接所需杜邦线等辅助器件;





## 4.1 驱动小车移动

### 3) 任务分解

- ① 硬件部分：ARDUINO;  
电机驱动模块+直流减速电机;  
智能小车;  
电路连接所需杜邦线等辅助器件;

`pinMode(pin, mode)` → 0-13引脚中的一个  
→ 输入 或 输出

- ② 软件部分：C语言基础;  
GPIO函数;  
直行，转弯对应的驱动信息;

- ③ 软、硬件联调：注意电路连接及程序语法正确;

端口	IN1	IN2	动作
参数	LOW	LOW	停止
	LOW	HIGH	反转
	HIGH	LOW	正转
	HIGH	HIGH	停止

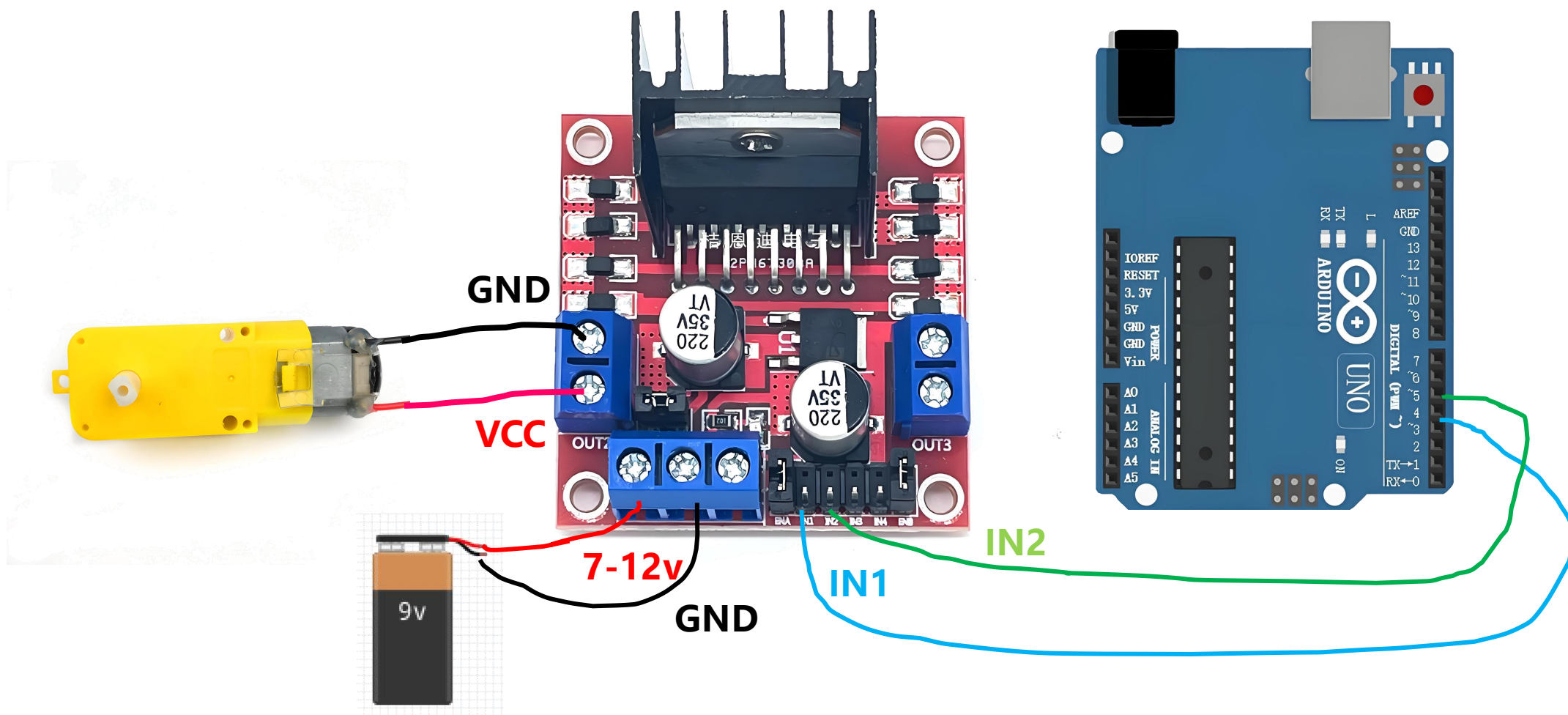


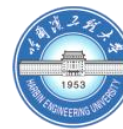


## 4.1 驱动小车移动

### 4) 任务实施

#### ① 电路连接





# 4.1 驱动小车移动

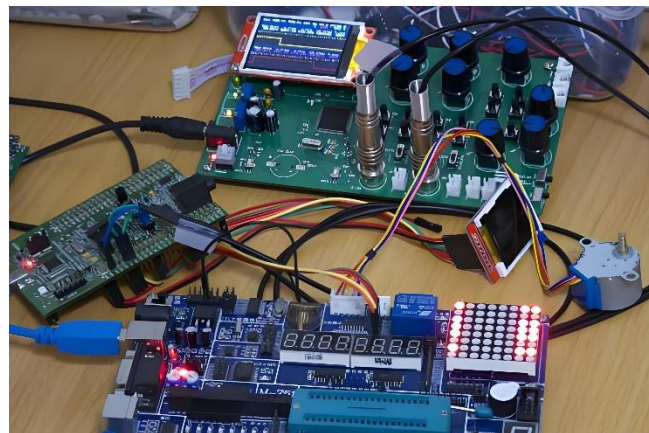
## 4) 任务实施

### ② 编写程序

```
1 void setup(){  
2     pinMode(4, OUTPUT);  
3     pinMode(5, OUTPUT);  
4  
5 }  
6  
7 void loop(){  
8     digitalWrite(4,HIGH);  
9     digitalWrite(5,LOW);  
10 }
```

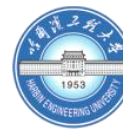
### 单电机驱动程序

### ③ 软、硬件联调 (教师指导、答疑)



### ④ 运行、演示 (教师指导、答疑)





# 4.1 驱动小车移动

## 4) 任务实施

### ⑤ 思考与拓展



测试效果如何?

电机运动方向与期望值相反，如何调整？  
有几种调整方式？

仿照例子，编写双电机**直行**控制程序！

那种方式更好？

1 调换导线正负极

2 调节I/O口控制参数调速

端口	IN1	IN2	动作
参数	LOW	LOW	停止
	LOW	HIGH	反转
	HIGH	LOW	正转
	HIGH	HIGH	停止



# 4.1 驱动小车移动



## 4) 任务实施

### ⑤ 思考与拓展

```
1 void setup(){
2     pinMode(4, OUTPUT);
3     pinMode(5, OUTPUT);
4     pinMode(6, OUTPUT);
5     pinMode(7, OUTPUT);
6 }
7
8 void loop(){
9     digitalWrite(4,HIGH);
10    digitalWrite(5,LOW);
11    digitalWrite(6,HIGH);
12    digitalWrite(7,LOW);
13
14 }
```



双电机的运行效果如何？两个电机**会同步运行吗？**如何同步？

**四电机独立驱动**有什么优点？

仿照例子，编写四电机控制程序！  
分别编写：前进、左转、右转、停止的程序！

**双电机驱动程序**





# 4.1 驱动小车移动



## 4) 任务实施

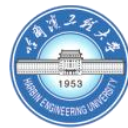
### ⑤ 思考与拓展

除了刚刚控制小车转弯的方式（一个不转，一个转），是否还能想到其他的转弯方式？





# 4.1 驱动小车移动



## 4) 任务实施

### ⑤ 思考与拓展



除了刚刚控制小车转弯的方式（一个不转，一个转），是否还能想到其他的转弯方式？

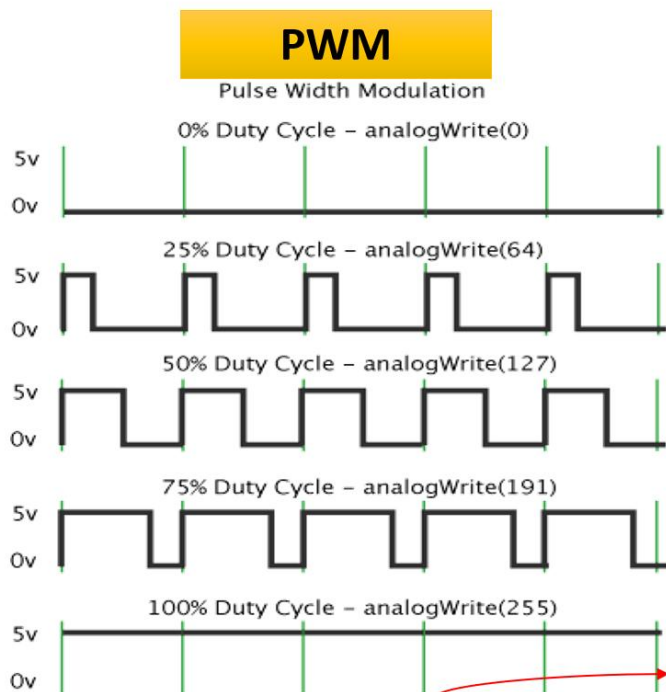
应用课程第三部分学习的**PWM**控制两轮转速差，实现转弯。

- (3,5-6,9-11)的引脚可以输出模拟信号。
- 他们实际是数字信号引脚，如何实现的输出模拟信号？

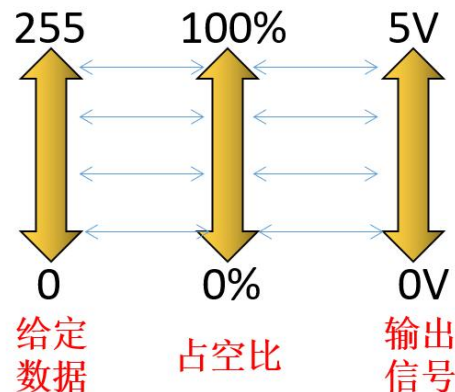
● PWM——脉宽调制

- 如何输出“x%”的占空比？

- Embedded function : `analogWrite(pin, value);`



- 调用`analogWrite()` 函数时，别忘了设置对应引脚为输出模式。



3,5-6,9-11引脚中的一个

数值为0~255



## 4.2驱动炮塔移动

### 1) 明确目的及要求

对标任务：实现炮塔的上、下俯仰、左转、右转几个动作。



### 2) 工作逻辑分析



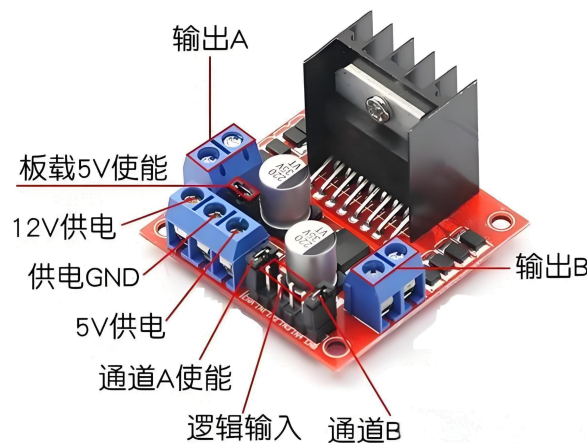


## 4.2驱动炮塔移动



### 3) 任务分解

- ① 硬件部分：ARDUINO；  
电机驱动模块+电机+舵机；  
智能小车；  
电路连接所需杜邦线等辅助器件；
- 管脚定义，性能参数是否匹配等！
- 明确被控对象结构特点，确保控制方案与机械部分相匹配。







## 4.2驱动炮塔移动



### 3) 任务分解

① 硬件部分：ARDUINO；

电机驱动模块+直流减速电机+舵机；

智能小车；

电路连接所需杜邦线等辅助器件；

`pinMode(pin, mode)`

0-13引脚中的一个

输入 或 输出

② 软件部分：C语言基础；

GPIO函数；

上、下俯仰的驱动信息；

③ 软、硬件联调：注意电路连接及程序语法正确；

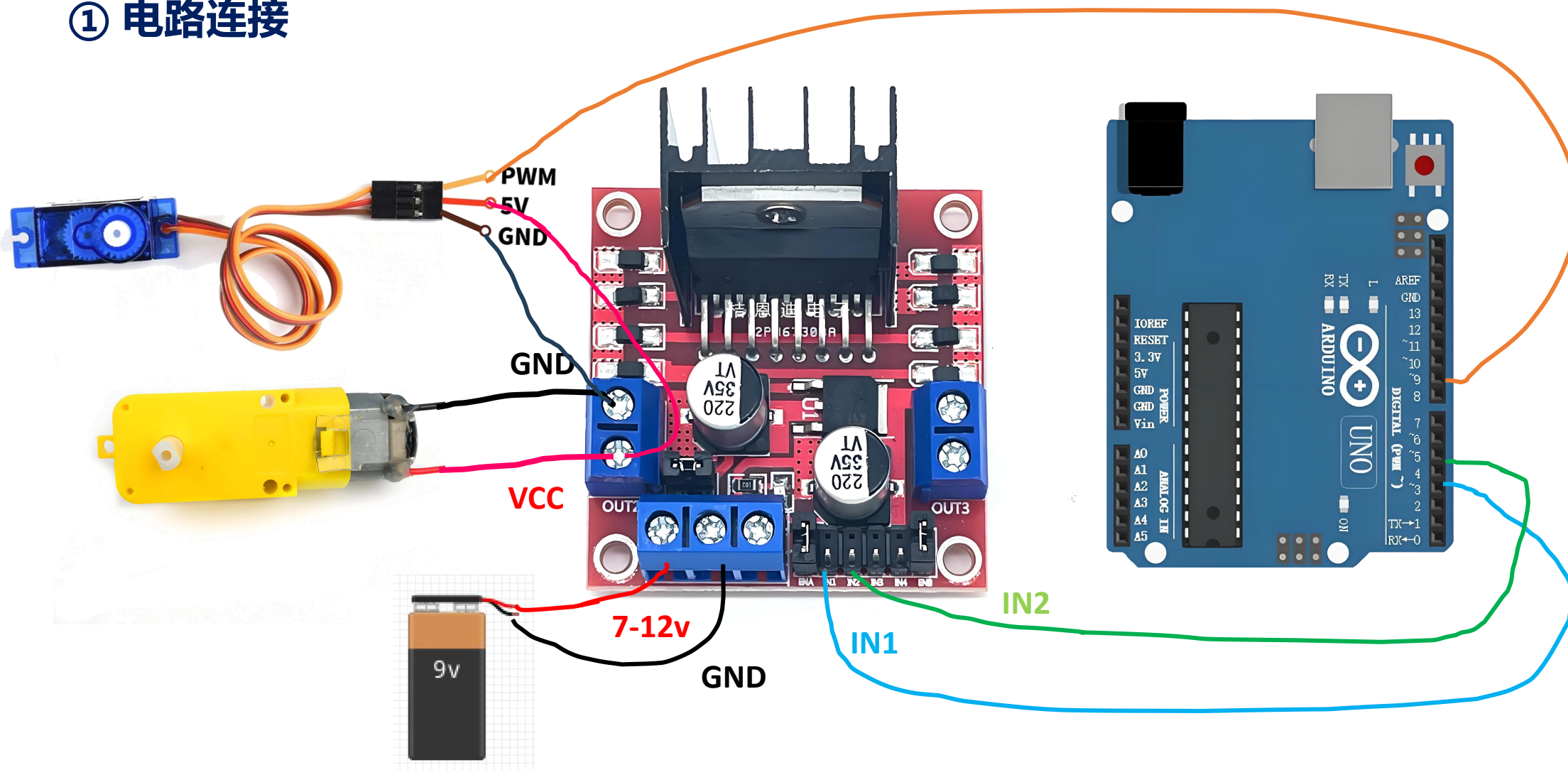
端口	IN1	IN2	动作
参数	LOW	LOW	停止
	LOW	HIGH	反转
	HIGH	LOW	正转
	HIGH	HIGH	停止



## 4.2 驱动炮塔移动

### 4) 任务实施

#### ① 电路连接





## 4.2驱动炮塔移动

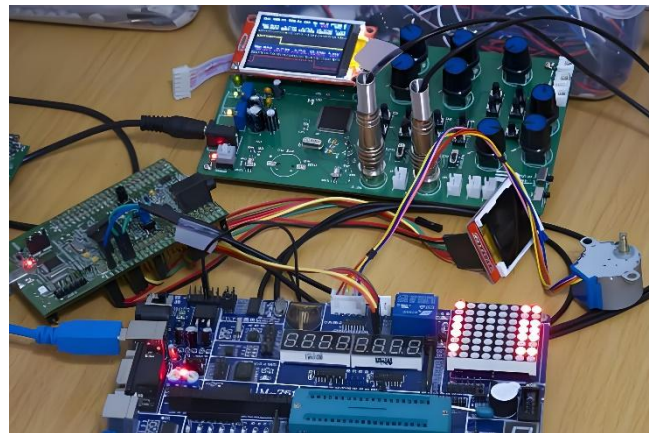
### 4) 任务实施

#### ② 编写程序

```
1  #include <Servo.h>
2  servo myServo;
3  int angle=90;
4
5  void setup(){
6      myServo.attach(9);
7      Serial.begin(9600);
8
9  }
10 void loop(){
11     while(Serial.available()==0);
12     angle=Serial.parseInt();
13     serial.println(angle);
14
15     myServo.write(angle);
16
17 }
```

舵机控制程序示例

#### ③ 软、硬件联调 (教师指导、答疑)



#### ④ 运行、演示 (教师指导、答疑)





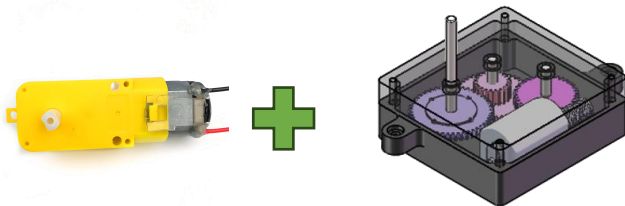
## 4.2驱动炮塔移动

### 4) 任务实施

#### ⑤ 思考与拓展



舵机和电机+齿轮传动两种方式都可以实现炮塔的左右旋转，二者有什么区别？

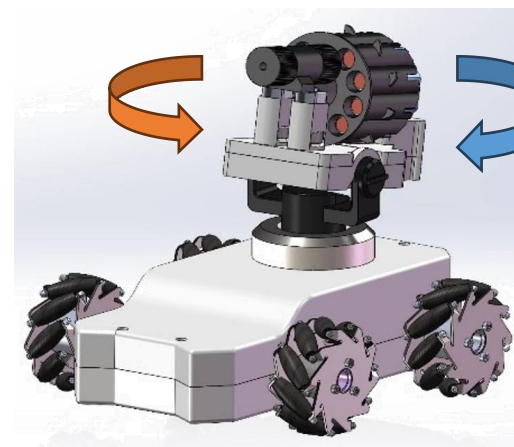
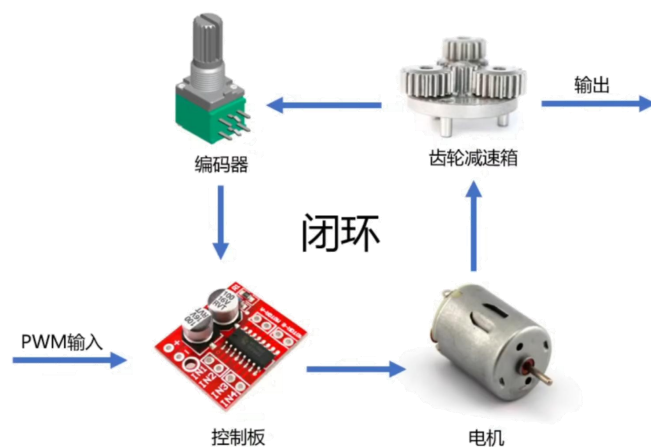


直流电机+齿轮传动

舵机可以控制转动角度，而后者不可以。



舵机

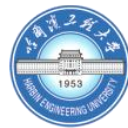


炮塔的左右旋转





## 4.2驱动炮塔移动



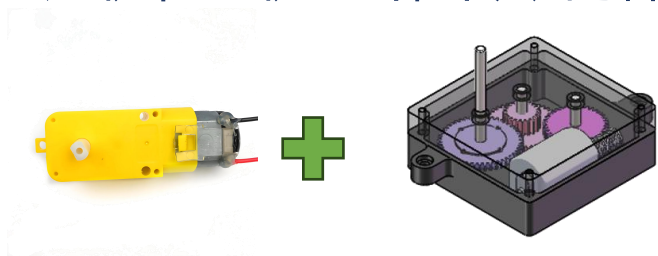
哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

### 4) 任务实施

#### ⑤ 思考与拓展



舵机和电机+齿轮传动两种方式都可以实现炮塔的左右旋转，二者有什么区别？



直流电机+齿轮传动



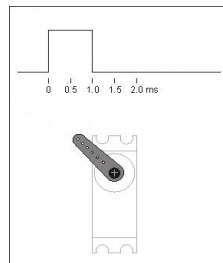
舵机

舵机可以控制转动角度，而后者不可以。

能否让直流电机+齿轮传动的组合方式，也能实现角度控制呢？（加计数装置）

**拓展任务：**在串口输入端，输入不同的角度，实时控制舵机转向；

输入角度

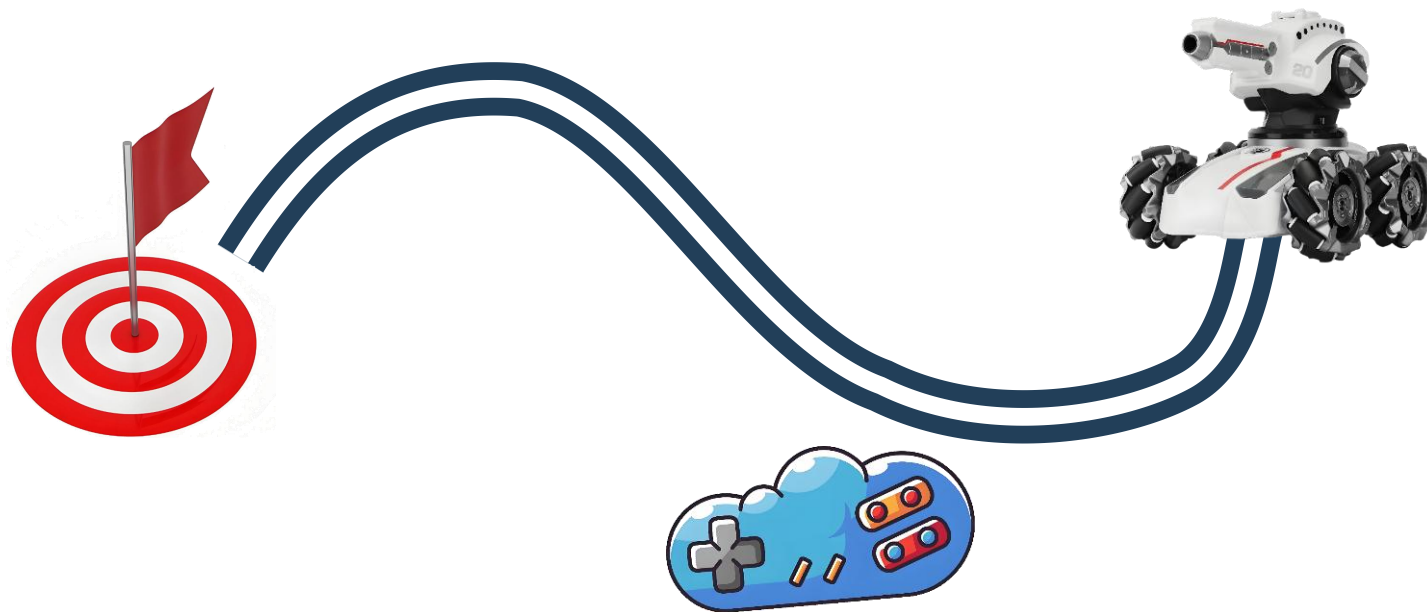




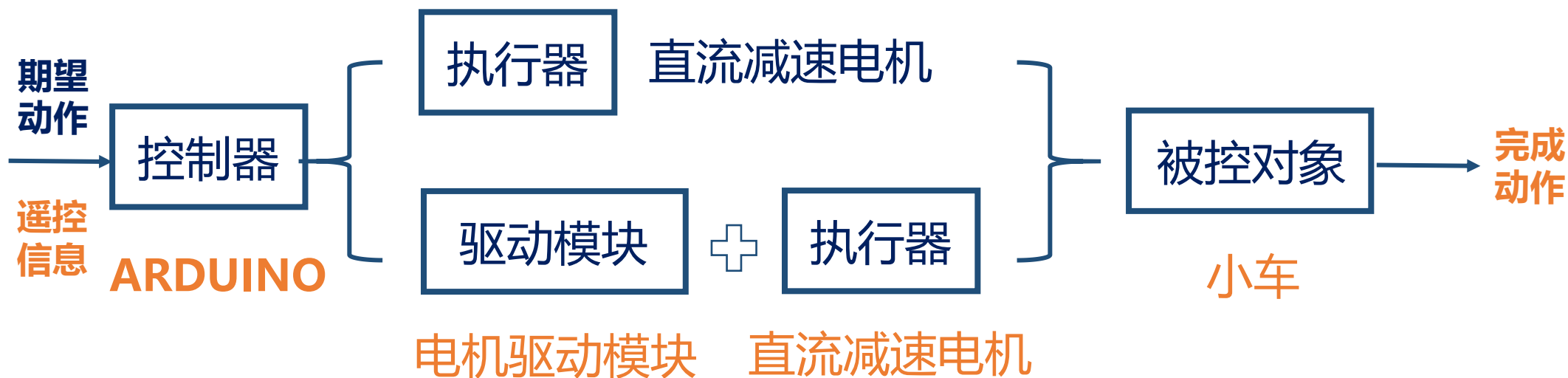
## 4.3 遥控小车移动

### 1) 明确目的及要求

**对标任务：**无线遥控  
小车行进到指定位置。



### 2) 工作逻辑分析



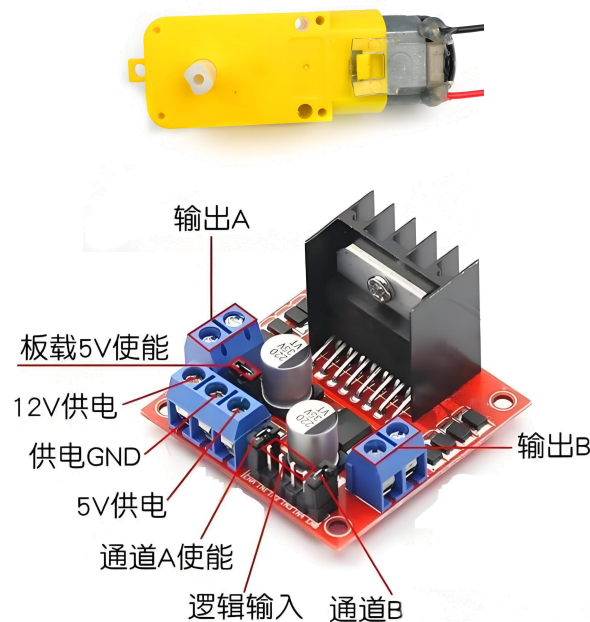
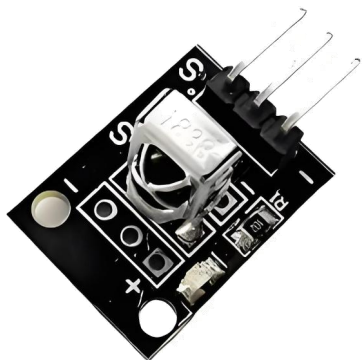


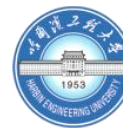
## 4.3 遥控小车移动



### 3) 任务分解

- ① 硬件部分：ARDUINO；  
电机驱动模块+直流减速电机+遥控；  
智能小车；  
电路连接所需杜邦线等辅助器件；
- 管脚定义，性能参数是否匹配等！  
明确被控对象结构特点，确保控制方案与机械部分相匹配。





## 4.3 遥控小车移动

### 3) 任务分解

- ① 硬件部分：ARDUINO;  
电机驱动模块+直流减速电机;  
智能小车;  
电路连接所需杜邦线等辅助器件;

`pinMode(pin, mode)` 0-13引脚中的一个  
输入 或 输出

- ② 软件部分：C语言基础;  
GPIO函数;  
直行，转弯对应的驱动信息;

- ③ 软、硬件联调：注意电路连接及程序语法正确;

端口	IN1	IN2	动作
参数	LOW	LOW	停止
	LOW	HIGH	反转
	HIGH	LOW	正转
	HIGH	HIGH	停止



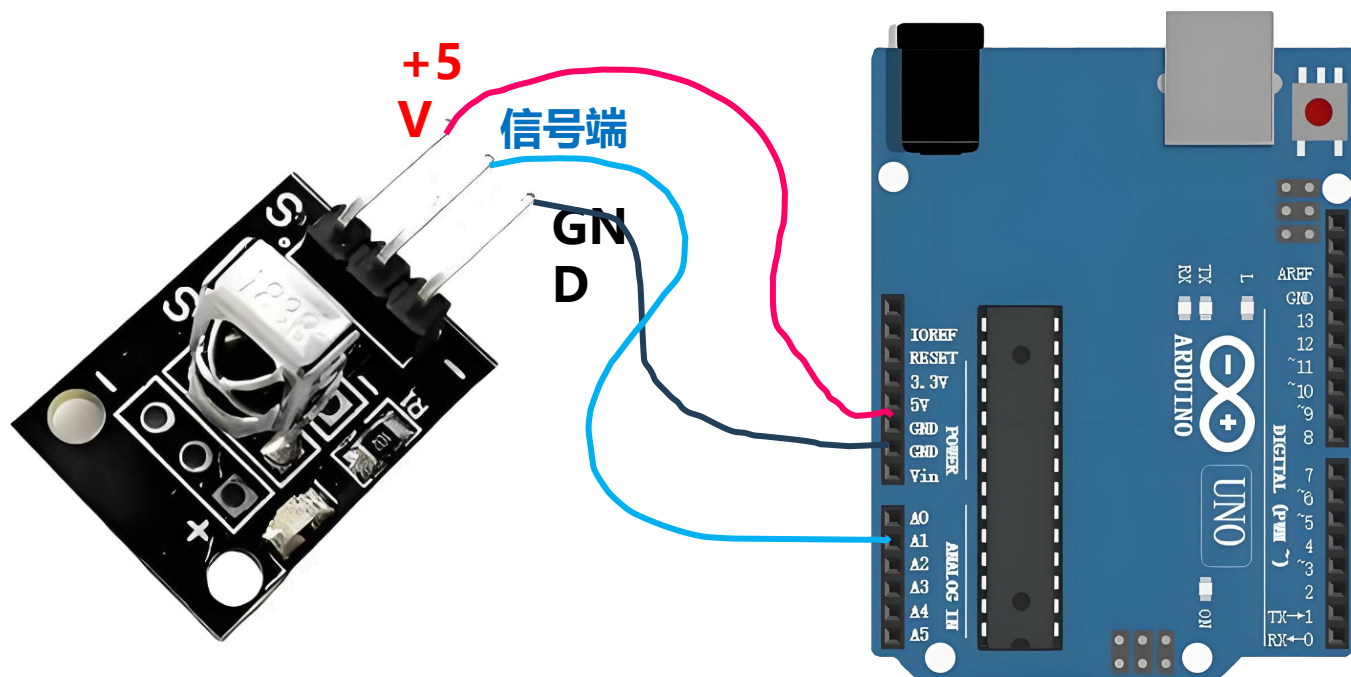


## 4.3 遥控小车移动



### 4) 任务实施

#### ① 电路连接



红外无线接收模块与ARDUINO接线示意图

按键	编码	按键	编码
1	0xFFA25D	9	0xFF960F
2	0xFF629D	0	0xFF9867
3	0xFFE21D	*	0xFF6897
4	0xFF22DD	#	0xFFB04F
5	0xFF02FD	上	0xFF18E7
6	0xFFC23D	下	0xFF4AB5
7	0xFFE01F	左	0xFF10EF
8	0xFFA857	右	0xFF5AA5

按键对应编码



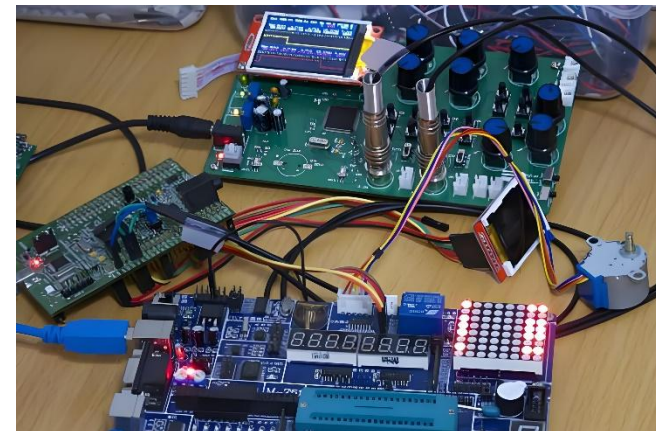
## 4.3 遥控小车移动

### 4) 任务实施

#### ② 编写程序

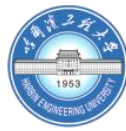
```
1  #include <IRremote.h>
2  IRrecv irrecv_A0(A0);
3  decode_results results_A0;
4  long ir_item;
5  void go() {
6      digitalWrite(4,HIGH);
7      digitalWrite(5,LOW);
8      digitalWrite(6,HIGH);
9      digitalWrite(7,LOW);
10 }
11 void setup(){
12     Serial.begin(9600);
13     pinMode(4, OUTPUT);
14     pinMode(5, OUTPUT);
15     pinMode(6, OUTPUT);
16     pinMode(7, OUTPUT);
17     irrecv_A0.enableIRIn();
18 }
19 void loop(){
20     if (irrecv_A0.decode(&results_A0)) {
21         ir_item=results_A0.value;
22         String type="UNKNOWN";
23         String typelist[14]={"UNKNOWN", "NEC", "SONY", "RC5", "RC6", "DISH", "SHARP", "PANASONIC", "JVC", "SANYO", "MITSUBISHI", "SAMSUNG", "LG", "WHYNTER"};
24         if(results_A0.decode_type>=1&&results_A0.decode_type<=13){
25             type=typelist[results_A0.decode_type];
26         }
27         Serial.print("IR TYPE:"+type+" ");
28         Serial.println(ir_item,HEX);
29         irrecv_A0.resume();
30     } else {
31         if (ir_item == 0xFF18E7) {
32             go();
33         }
34     }
35 }
```

#### ③ 软、硬件联调 (教师指导、答疑)



#### ④ 运行、演示 (教师指导、答疑)





## 4.3 遥控小车移动

### 4) 任务实施

#### ⑤ 思考与拓展

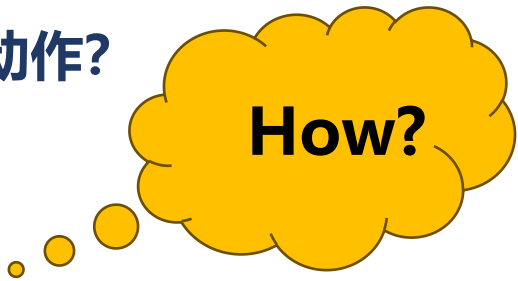
仿照例子，编写后退、左转、右转的遥控程序！

尝试遥控小车实现前进、后退、左转、右转的动作！

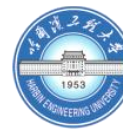
是否可以让它自主完成预期动作？



人工控制



自动控制



## 4.4 自动循迹小车

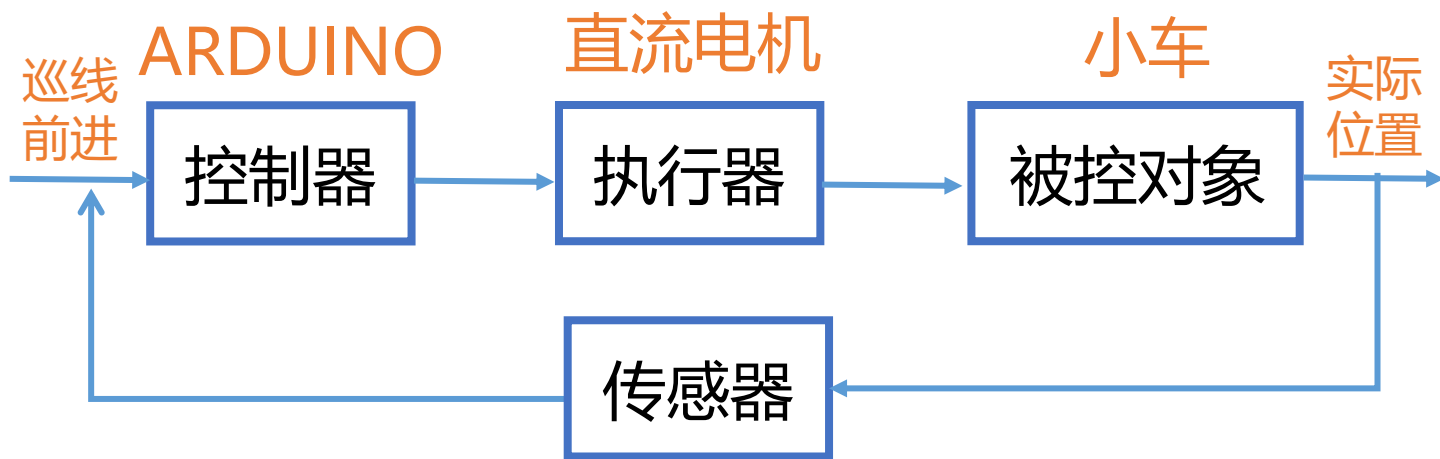
### 1) 明确目的及要求

**对标任务：**实现小车自动的按照预定路线行走。



自动巡线

### 2) 工作逻辑分析



红外循迹传感器





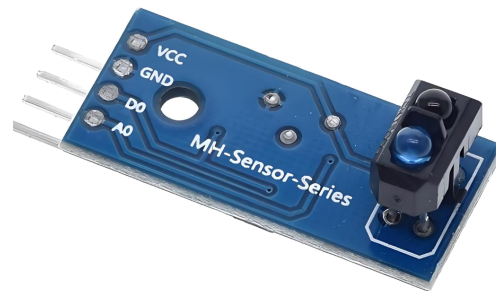
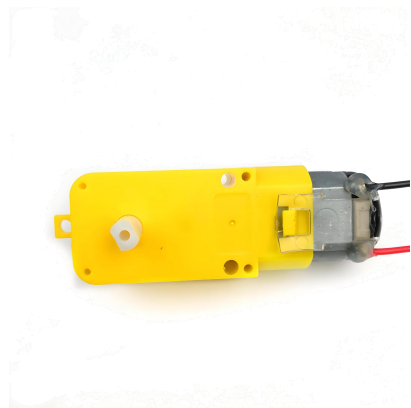
## 4.4 自动循迹小车

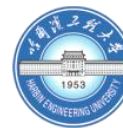


### 3) 任务分解

- ① 硬件部分：ARDUINO；  
电机驱动模块+直流减速电机；  
红外循迹传感器；  
智能小车；→ 明确被控对象结构特点，确保控制方案与机械部分相匹配。  
电路连接所需杜邦线等辅助器件；

管脚定义，性能参数是否匹配等！





## 4.4 自动循迹小车

### 3) 任务分解

#### ① 硬件部分：ARDUINO；

电机驱动模块+直流减速电机；

红外循迹传感器；

智能小车；

电路连接所需杜邦线等辅助器件；

#### ② 软件部分：C语言基础；

GPIO函数；

直行，转弯对应的驱动信息；

#### ③ 软、硬件联调：注意电路连接及程序语法正确；

`pinMode(pin, mode)` 0-13引脚中的一个  
输入 或 输出

端口	IN1	IN2	动作
参数	LOW	LOW	停止
	LOW	HIGH	反转
	HIGH	LOW	正转
	HIGH	HIGH	停止

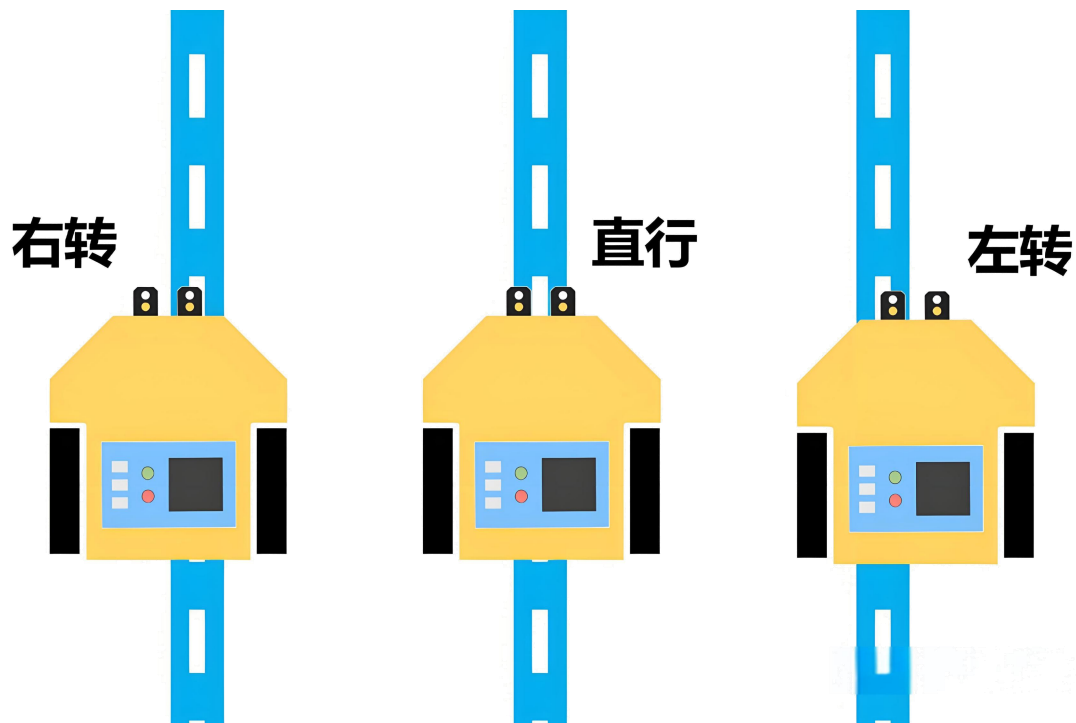


## 4.4 自动循迹小车

### 循迹原理

当传感器对准深色线路时，其发射管发射出的红外线被线路**吸收**，而接收接收到的红外线非常弱，那么接收管截止(电阻大)，导致接收管端的输出**电压变小**。反之，收管端的输出电压变大。

通过读取红外传感器的输出电压是否发生变化就可以知道小车是否检测到**路径**。当Arduino模拟引脚读到的电压变小时，则说明检测到**路径**，反之,则说明偏离黑线。



左传感器	右传感器	动作
0	0	停止
0	1	右转
1	0	左转
1	1	直行

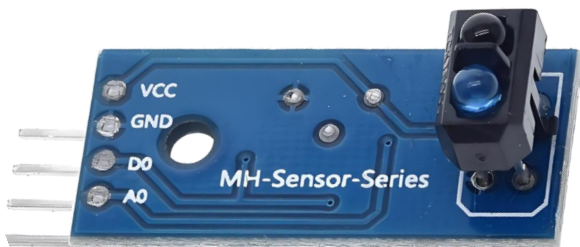


## 4.4 自动循迹小车



### 4) 任务实施

#### ① 电路连接



**Vcc:** 接电源正极 (3-5V)

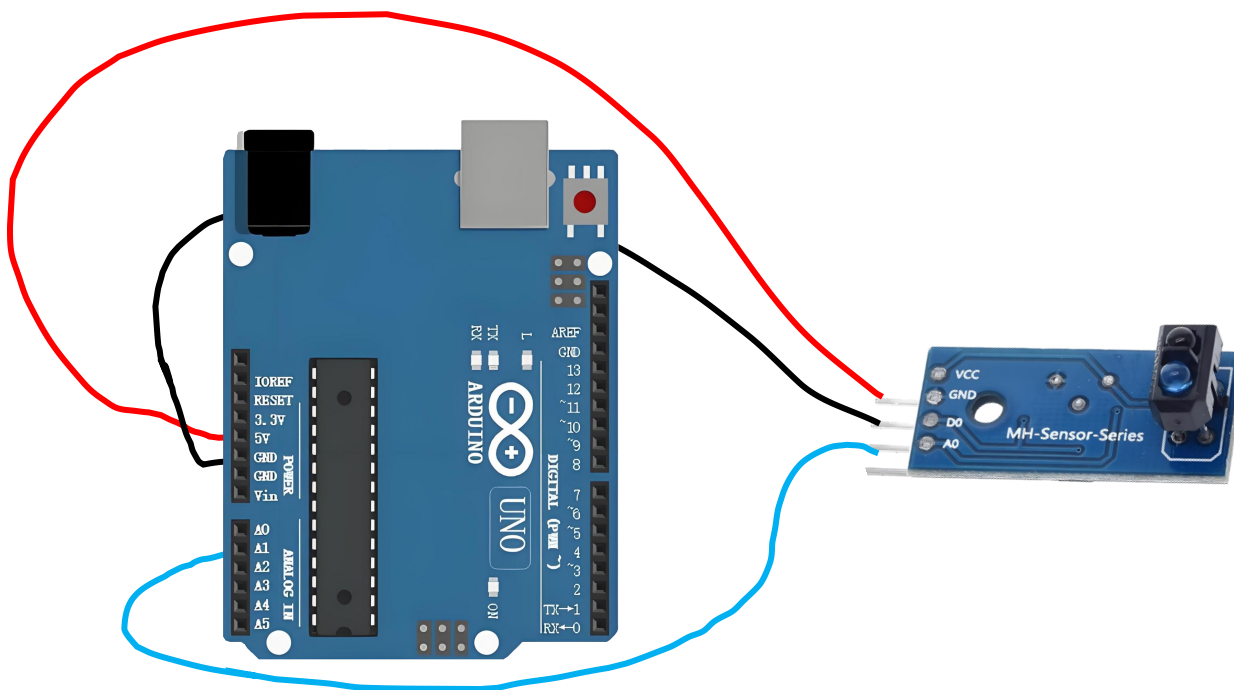
**GND:** 接电源负极

**D0:** TTL开关信号输出

**A0:** 模拟信号输出

#### 工作原理

传感器的红外发射二极管会不断发射红外线，当发射的红外线没有被反射回来时，红外接收端一直处于关闭状态，此时模块的输出端为高电平，指示二极管一直处于熄灭状态。







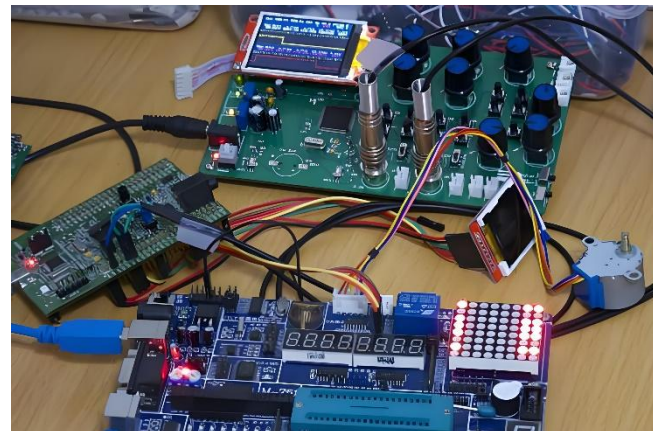
## 4.4 自动循迹小车

### 4) 任务实施

#### ② 编写程序

```
1 void go() {  
2     digitalWrite(4,HIGH);  
3     digitalWrite(5,LOW);  
4     digitalWrite(6,HIGH);  
5     digitalWrite(7,LOW);  
6 }  
7 void left() {  
8     digitalWrite(4,HIGH);  
9     digitalWrite(5,LOW);  
10    digitalWrite(6,LOW);  
11    digitalWrite(7,LOW);  
12 }  
13 void right() {  
14     digitalWrite(4,LOW);  
15     digitalWrite(5,LOW);  
16     digitalWrite(6,HIGH);  
17     digitalWrite(7,LOW);  
18 }  
19 void stop() {  
20     digitalWrite(4,LOW);  
21     digitalWrite(5,LOW);  
22     digitalWrite(6,LOW);  
23     digitalWrite(7,LOW);  
24 }  
25  
26 void setup(){  
27     pinMode(4, OUTPUT);  
28     pinMode(5, OUTPUT);  
29     pinMode(6, OUTPUT);  
30     pinMode(7, OUTPUT);  
31     pinMode(A2, INPUT);  
32     pinMode(A1, INPUT);  
33 }  
34 void loop(){  
35     if (digitalRead(A2) == 0 && digitalRead(A1) == 0) {  
36         stop();  
37     } else if (digitalRead(A2) == 1 && digitalRead(A1) == 0) {  
38         right();  
39     } else if (digitalRead(A2) == 0 && digitalRead(A1) == 1) {  
40         left();  
41     } else if (digitalRead(A2) == 1 && digitalRead(A1) == 1) {  
42         go();  
43     }  
44 }
```

#### ③ 软、硬件联调 (教师指导、答疑)



#### ④ 运行、演示 (教师指导、答疑)





## 4.4 自动循迹小车



### 4) 任务实施

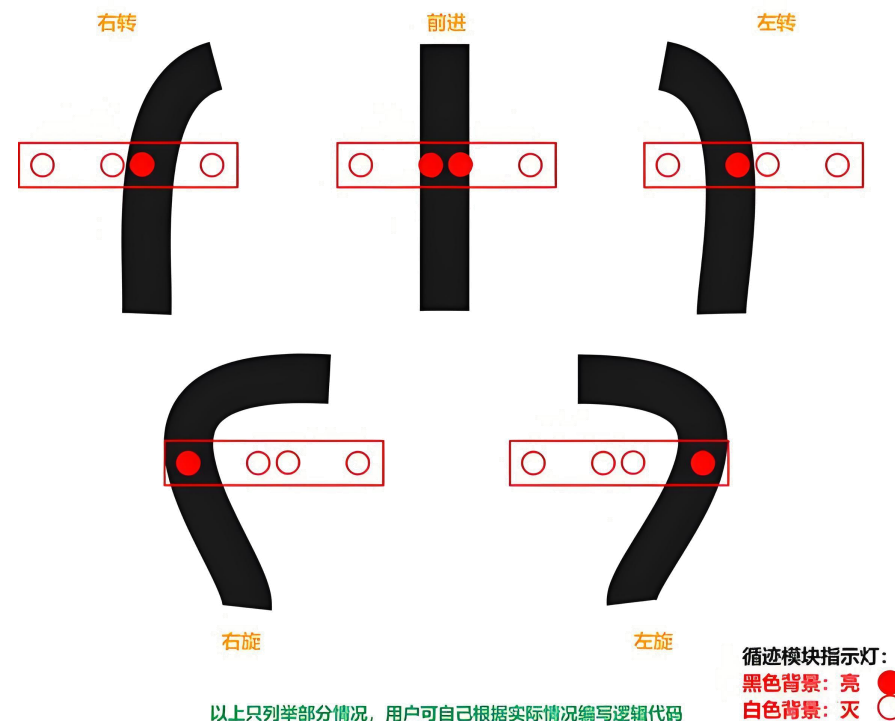
#### ⑤ 思考与拓展

仿照示例，上机调试；

寻迹效果如何？如何改进？

拓展:尝试3-5红外传感器，改进小车跟踪效果；

拓展:尝试在2传感器基础上，使用PID算法改进小车跟踪效果；



路径跟踪——四传感器



# 作业（预习）及参考资料



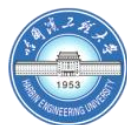
哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

## 作业

1 设计一套控制系统，实现小车自动行驶到指定位置后，炮塔锁定目标位置；

## 参考资料

- 1 《大话自动化》，机械工业出版社，2024；
- 2 《什么是自动化》，大连理工大学出版社，2021；
- 3 <https://www.arduino.cc/>
- 4 《学ARDUINO玩转机器人制作》，人民邮电出版社，2014；



哈尔滨工程大学  
HARBIN ENGINEERING UNIVERSITY

下课!

