**Arduino课堂笔记**

**1.函数封装**

为了使程序更加清晰、易于设计、易于维护，我们需要对程序进行模块化设计，将程序切分为一个个模块，模块之间的联系尽可能少。

* + 通过自定义函数机制，将一部分代码从主函数中剥离，可以将一个模块压缩成主程序中的一个语句，将模块的核心语句移入子函数。
  + 语句块独立为函数之后，可以在任何地方通过函数名使用，不必再书写多余的语句。

**函数的定义：**

数据类型 函数名(形式参数表)

{

函数体; //执行语句

}

说明：数据类型是函数的返回值类型，如果没有返回值，则类型为void，如果有返回值则类型为返回值的类型。

函数名是标识符，按照标识符的取名规则命名，最好取有助于记忆的名字。

**函数的调用：**在程序运行过程中，可以调用自定义函数，调用函数时就会进入该函数，执行该函数中的语句，当执行到函数中的返回语句或执行完所有语句时，再回到调用函数的地方继续执行后面的语句。

形式：函数名(实参列表)

例如： f(x)或f()

**函数的参数：**

* 对于有参函数来说，函数调用时的传递的参数叫实际参数，函数定义时的参数叫形式参数。
* 在函数调用时，实际参数的值通过赋值传递给形式参数。
* 实际参数和形式参数要一一对应，第一个实际参数把值传递给第一个形式参数，第二个实际参数把值传递给第二个形式参数。通常要求形式参数和实际参数数量一致、一一对应。

**超声波函数封装示例：**

float getDistance() {

float temp = 0, cm = 0;

digitalWrite(Trig, LOW);

delayMicroseconds(2);

digitalWrite(Trig, HIGH);

delayMicroseconds(10);

digitalWrite(Trig, LOW);

temp = (float)(pulseIn(Echo, HIGH));

cm = temp \* 17 / 1000;

return cm;

}

**电机函数封装示例：**

void mForward(int spl, int spr) {//括号中有两个形式参数，都是int类型，

//spl表示左电机速度，spr表示右电机速度

//右上角电机正转

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

//左上角电机正转

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

**2.编程任务**

（1）小车跟随主人前进和后退，当人前进时小车也前进，当人后退时小车也后退

分析：

* + - 当超声波传感器检测到前方物体时小车后退。
    - 当超声波传感器检测不到前方物体时小车前进。

#define IN1 2

#define IN2 4

#define ENA 3

#define IN3 7

#define IN4 6

#define ENB 5

#define Trig A0

#define Echo A1

float cmMidlle = 0;

void setup() {

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(ENA, OUTPUT);

pinMode(Trig, OUTPUT);

pinMode(Echo, INPUT);

Serial.begin(9600);

delay(1000);

}

void loop() {

cmMidlle = getDistance();

if (cmMidlle < 30)

{ //后退

mBackward(110, 110);

} else {//前进

mForward(110, 110);

}

delay(100);

}

float getDistance() {//超声波传感器检测距离的函数

float temp = 0, cm = 0;

digitalWrite(Trig, LOW);

delayMicroseconds(2);

digitalWrite(Trig, HIGH);

delayMicroseconds(10);

digitalWrite(Trig, LOW);

temp = (float)(pulseIn(Echo, HIGH));

cm = temp \* 17 / 1000;

return cm;

}

void mForward(int spl, int spr) {//小车前进的函数

//右上角电机正转

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

analogWrite(ENA, spr);

//左上角电机正转

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

analogWrite(ENB, spl);

}

void mBackward(int spl, int spr) {//小车后退的函数

//右上角电机反转

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

analogWrite(ENA, spr);

//左上角电机反转

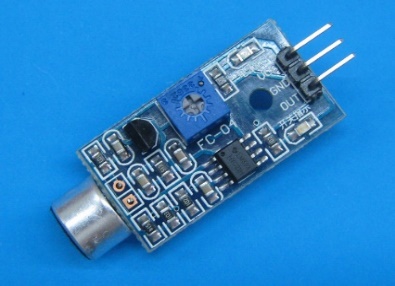
digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

analogWrite(ENB, spl);

}

**3.声音传感器**



* 可以检测周围环境的声音强度。
* 灵敏度可调（图中蓝色数字电位器调节）
* 工作电压3.3V-5V
* 输出形式 数字开关量输出（0和1高低电平）

模块接线说明

VCC 接3.3V-5V电压

GND 接GND

OUT Arduino的数字引脚

使用说明

(1)声音模块对环境声音强度比较敏感，一般用来检测周围环境的声音强度。

(2)声音模块在环境声音强度达不到设定阈值(模块上的蓝色旋转电阻调节)时，OUT引脚输出高电平，当外界环境声音强度超过设定阈值时，OUT引脚输出低电平；

(3)声音模块的OUT引脚可以与单片机直接相连，通过单片机来检测高低电平，由此来检测环境的声音；

程序示例：

void setup() {

pinMode(3,INPUT);//声音传感器连接到Arduino的数字引脚，例如3引脚

Serial.begin(9600);

}

int a=0;

void loop() {

a=digitalRead(3);//获取声音传感器的值

Serial.println(a);

}

**4.编程任务**

(1)当有声音比较大时，LED灯亮，否则就熄灭

void setup() {

pinMode(2, INPUT);

pinMode(3, OUTPUT);

}

void loop() {

if (digitalRead(2) == 0)

{

digitalWrite(3, HIGH);

delay(1000);

}

else

{

digitalWrite(3, LOW);

}

}

(2)当第一次拍手时，LED灯亮，并且一直亮

当第二次拍手时，LED灯熄灭，并且一直灭

当第三次拍手时，LED灯再次点亮

………

重复执行以上步骤，也就是拍手LED灯亮，再拍手LED灯灭

示例代码：

int wangsihan = 0;

void setup() {

pinMode(2,INPUT);

pinMode(3,OUTPUT);

}

void loop() {

if(digitalRead(2)==0 and wangsihan == 0)

{

digitalWrite(3,HIGH);

wangsihan=1;

}

else if(digitalRead(2)==0 and wangsihan == 1)

{

digitalWrite(3,LOW);

wangsihan=0;

}

}