

项目八 火焰报警器

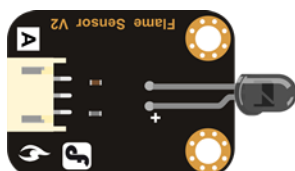
在厨房安装一个火焰报警器应该是非常管用的，如果不小心忘关煤气的话，只要有一点点的火苗，就能触发火焰报警器，探测距离可达 20cm。别看一个小小的报警器，说不定就能避免一场不必要的意外发生，何乐而不为呢？

所需材料

- 1× 数字蜂鸣器模块



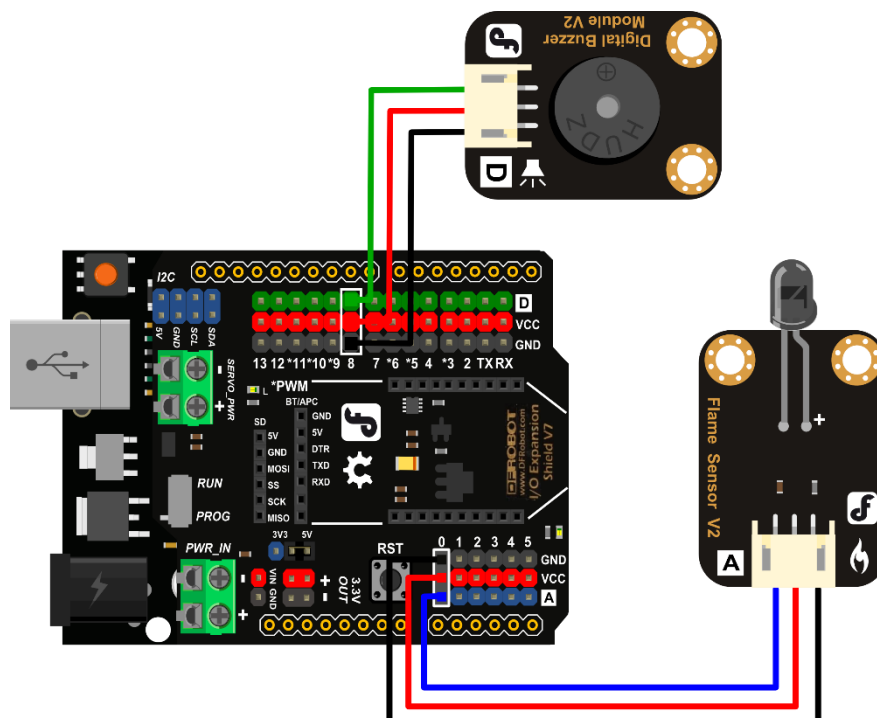
- 1× 火焰传感器



硬件连接

数字蜂鸣器模块 → 数字口 8

火焰传感器 → 模拟口 0



输入代码

样例代码 8-1:

```
//项目八 - 火焰报警器
float sinVal;
int toneVal;

void setup(){
    pinMode(8, OUTPUT);          // 蜂鸣器引脚设置
    Serial.begin(9600);          //设置波特率为 9600 bps
}

void loop(){
    int sensorValue = analogRead(0); //火焰传感器连到模拟口，并从模拟口读值
    Serial.println(sensorValue);
    delay(1);

    if(sensorValue < 1023){ // 如果数据小于 1023，说明有火源，蜂鸣器响
        for(int x=0; x<180; x++){
            //将 sin 函数角度转化为弧度
            sinVal = (sin(x*(3.1412/180)));
            //用 sin 函数值产生声音的频率
            toneVal = 2000+(int(sinVal*1000));
            //给引脚 8 一个
            tone(8, toneVal);
            delay(2);
        }
    } else { // 如果数据大于等于 1023，没有火源，关闭蜂鸣器
        noTone(8); //关闭蜂鸣器
    }
}
```

可以试下那个打火机慢慢靠近火焰传感器,看看蜂鸣器会不会报警。

代码回顾

首先，定义两个变量：

```
float sinVal;
int toneVal;
```

浮点型变量 sinVal 用来存储正弦值，正弦波呈现一个波浪形的变化，变化比较均匀，所以我们选用正弦波的变化来作为我们声音频率的变换，toneVal 从 sinVal 变量中获得数值，并把它转换为所需要的频率。

这里用的是 `sin()` 函数，一个数学函数，可以算出一个角度的正弦值，这个函数采用弧度单位。因为我们不想让函数值出现负数，所以设置 `for` 循环在 0~179 之间，也就是 0~180 度之间。

```
for(int x=0; x<180; x++){}
```

函数 `sin()` 用的弧度单位，不是角度单位。要通过公式 $3.1412/180$ 将角度转为弧度：

```
sinVal = (sin(x*(3.1412/180)));
```

之后，将这个值转变成相应的报警声音的频率：

```
toneVal = 2000+(int(sinVal*1000));
```

这里有个知识点——浮点型值转换为整型。

`sinVal` 是个浮点型变量，也就是含小数点的值，而我們不希望频率出现小数点的，所以需要有一个浮点值转换为整型值得过程，也就是下面这句语句就完成了这件事：

```
int(sinVal*1000)
```

把 `sinVal` 乘以 1000，转换为整型后再加上 2000 赋值给变量 `toneVal`，现在 `toneVal` 就是一个适合声音频率了。

之后，我们用 `tone()` 函数把生成的这个频率给我们的蜂鸣器。

```
tone(8, toneVal);
```

下面我们来介绍一下 `tone` 相关的三个函数

(1) `tone(pin, frequency)`

`Pin` 都是指连接到蜂鸣器的数字引脚，`frequency` 是以 Hz 为单位的频率值。

(2) `tone(pin, frequency, duration)`

第二个函数，有个 `duration` 参数，它是以毫秒为单位，表示声音长度的参数。像第一个函数，如果没有指定 `duration`，声音将一直持续直到输出一个不同频率的声音产生。

(3) `noTone(pin)`

`noTone(pin)` 函数，结束该指定引脚上产生的声音。

趣味练习

结合人体红外热释电传感器，红色 LED 发光模块可以做个防盗报警器。当然，如果要声音大一点的话，蜂鸣器的威力可能不太够，小型的报警器还是没有问题的。