

项目八 震动传感器

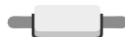
震动传感器，我们从名字中应该就可以判断，传感器能够检测震动中的物体。我们用什么来做震动传感器呢？那就是滚珠开关。滚珠开关，其内部含有导电珠子，器件一旦震动，珠子随之滚动，就能使两端的导针导通。

通过这个原理，我们可以做一些小玩具结合起来。最常见的，比如我们看到一些小孩子穿的一闪一闪的小鞋子！走动的过程，就能使内部珠子滚动。

只要传感器检测到东西震动，就会有信号输出。这里，我们想通过滚珠开关做个简单的震动传感器，并把震动传感器和 LED 的结合，当传感器检测到物体震动时，LED 亮起，停止震动时，LED 关闭。

所需元件

● 1× 滚珠开关 SW200D



● 1× 5mm LED 灯



● 1× 220 欧电阻



硬件连接

从滚珠开关这个名字，我们可以把它和什么联想在一起呢？就是按键开关，滚珠开关和我们项目三中介绍的按钮在硬件连接是完全相同的，原理也相似。只是使用方法不同而已。可以把下图对应项目三的一起看，你会发现很多相似之处。滚珠开关也需要一个下拉电阻，LED 需要一个限流电阻。

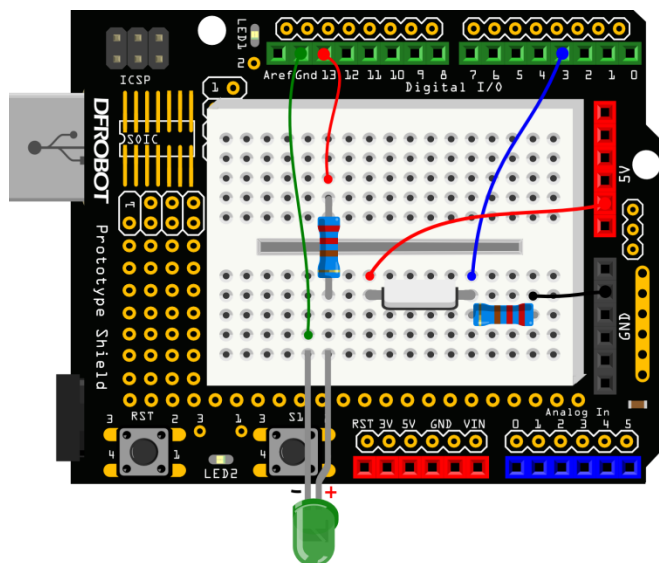


图 8-1 震动传感器连线图

输入代码

样例代码 8-1:

```
//项目八 - 震动传感器

int SensorLED = 13;          //定义 LED 为数字引脚 13
int SensorINPUT = 3;         //连接震动开关到中断 1, 也就是数字引脚 3
unsigned char state = 0;

void setup() {
    pinMode(SensorLED, OUTPUT);          //LED 为输出模式
    pinMode(SensorINPUT, INPUT);         //震动开关为输入模式

    //低电平变高电平的过程中, 触发中断 1, 调用 blink 函数
    attachInterrupt(1, blink, RISING);
}

void loop() {
    if(state!=0) {                  // 如果 state 不是 0 时
        state = 0;                  // state 值赋为 0
        digitalWrite(SensorLED, HIGH); // 亮灯
        delay(500);                 //延时 500ms
    }
    else{
        digitalWrite(SensorLED, LOW); // 否则, 关灯
    }
}

void blink() {                      //中断函数 blink()
    state++;                         //一旦中断触发, state 就不断自加
}
```

当我们晃动板子时, LED 灯也会随之亮, 一旦停止晃动, LED 灯又恢复到熄灭的状态。

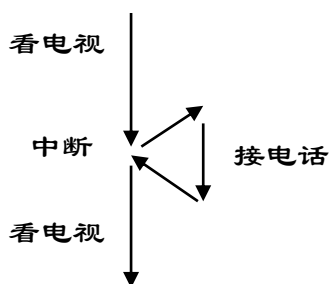
代码回顾

代码虽不长，但还是不太容易理解的。先大致说下代码的运行过程。

在没有任何打扰的情况下，程序在不断运行着…，让 LED 一直处于关闭。突然，被人打扰了（也就是晃动板子），就跳到中断函数 `blink()` 中（当然进入中断也是要条件的，我们后面说）。此时，`state` 不断自加，连锁反应的，主函数中 `if` 函数检测到 `state` 不为 0 了，那么就让 LED 亮起了，同时又重新让 `state` 为 0，等待下一次中断。如果没有中断的话，LED 有恢复到关闭的状态。

简单说了下程序的运行过程，重复的知识点就不做说明了。就重点说下中断函数 `attachInterrupt()`。

什么是中断？打个比方吧，比如你在家好好的看电视，突然家里电话铃响了，那么你不得不停下看电视，先去接电话，等接完电话后，你又可以继续看电视啦！在整个过程中，接电话就是一个中断过程，电话铃响就是中断的标志，或者说是中断条件。



现在知道中断是什么意思了，再回到 `attachInterrupt()` 函数，它是一个当外部发生中断时，才被唤醒的函数。区别于其他函数，它依附于中断引脚才发生。大多数板子都有两个外部中断引脚：数字引脚 2（中断 0）和数字引脚 3（中断 1）。中断 0 与中断 1 是中断号，在函数中需要用到。不同板子，中断号对应引脚可能不同，可以查阅 Arduino 官方编程语法手册（<http://arduino.cc/en/Reference/AttachInterrupt>）。

`attachInterrupt()` 需要三个传递参数：



interrupt: 中断号 0 或者 1。如果选择 0 的话，连接到数字引脚 2 上，选择 1 的话，连接到数字引脚 3 上。

function: 调用的中断函数名。写中断函数时，需要特别说明以下三点：

- ① 我们在写中断函数的时候，该函数不能含有参数和返回值。也就是说，要是个无返回值的函数。
- ② 中断函数中不要使用 delay() 和 millis() 函数，因为数值不会继续变化。
- ③ 中断函数中不要读取串口，串口收到的数据可能会丢失。

mode: 中断的条件。只有特定的以下四种情况：

- ① LOW: 当引脚为低电平时，触发中断。
- ② CHANGE 当引脚电平发生改变时，触发中断。
- ③ RISING 当引脚由低电平变为高电平时，触发中断。
- ④ FALLING 当引脚由高电平变为低电平时，触发中断。

知道了 attachInterrupt() 函数的用法，回归到我们的代码中：

```
attachInterrupt(1, blink, RISING);
```

对应上面说明看。1，指中断号 1。所以滚珠开关接到数字引脚 3。blink 是我们下面要调用的中断函数。RISING，指引脚 3 在由低变为高的一瞬间，中断触发。

为什么要选 RISING 呢？由于硬件我们还没提到，我们就先滚珠开关想象成按键。在按键没按下的时，是断开的，引脚 3 处于低的状态。一旦被按下，就和 5V 导通，变为高。这个过程是引脚由低电平变高电平的过程，所以选择 RISING 模式。

硬件回顾

滚珠开关

滚珠开关，也叫做珠子开关，震动开关等等。虽然叫法不同，不过原理是相同的。就是通过珠子滚动接触导针的原理来控制电路的通断。看下结构图就明白了。

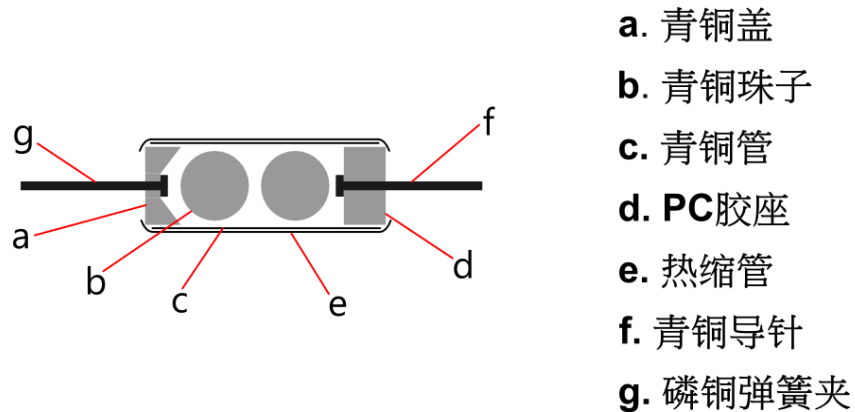


图 8-2 滚珠开关内部结构图

滚珠开关内部两个珠子，通过珠子滚动接触导针的原理来控制电路的接通或者断开。传感器震动或者晃动时，珠子就会接触导针，从而导通。还需要注意的一点是，由于滚珠开关的内部构造，滚珠开关只有一头是导通的，**金色导针一端是导通的，银色导针一端是不导通的**。这也就是为什么，往金色一端倾斜，灯会点亮，而偏向银色一端倾斜时，灯不会被点亮的原因。