

项目二 S.O.S 求救信号器

本项目将继续使用项目 1 的搭建的电路，但我们这里将改变一下代码，就能让我们的 LED 变为 S.O.S 求救信号了。这是国际莫尔斯码求救信号。莫尔斯码是一种字符编码，英文的每个字母，都是由横杠和点不同的组合而成。这样的好处是，使用简单的两种状态，就能来传递所有的字母和数字，非常的简便！不得不佩服前人的聪明吧！

我们正好可以通过 LED 开关两种状态来拼出一个个字母。通过长闪烁和短闪烁来表示点和横杠。我们这个项目中，我们就拼写 S.O.S 这三个字母。

通过查阅莫尔斯码表，我们可以知道，字母“S”用三个点表示，我们这里用短闪烁替代，字母“O”则用三个横杠表示，用长闪烁替代。

有了前一个项目的基础，不难理解下面样例代码 2-1。但先不要急着输入这段代码，只是看一下。

样例代码 2-1:

```
int ledPin = 10;
void setup() {
    pinMode(ledPin, OUTPUT);
}
void loop() {
    // 三个快闪烁来表示字母“S”
    digitalWrite(ledPin, HIGH);
    delay(150);
    digitalWrite(ledPin, LOW);
    delay(100);

    digitalWrite(ledPin, HIGH);
    delay(150);
    digitalWrite(ledPin, LOW);
    delay(100);

    digitalWrite(ledPin, HIGH);
    delay(150);
    digitalWrite(ledPin, LOW);
    delay(100);

    delay(100);           //100 毫秒延时产生字母之间的间隙
```

```
//三个短闪烁来表示字母“O”
digitalWrite(ledPin, HIGH);
delay(400);
digitalWrite(ledPin, LOW);
delay(100);

digitalWrite(ledPin, HIGH);
delay(400);
digitalWrite(ledPin, LOW);
delay(100);

digitalWrite(ledPin, HIGH);
delay(400);
digitalWrite(ledPin, LOW);
delay(100);

delay(100);           //100 毫秒延时产生字母之间的间隙

//再用三个快闪烁来表示字母“S”
digitalWrite(ledPin, HIGH);
delay(150);
digitalWrite(ledPin, LOW);
delay(100);

digitalWrite(ledPin, HIGH);
delay(150);
digitalWrite(ledPin, LOW);
delay(100);

digitalWrite(ledPin, HIGH);
delay(150);
digitalWrite(ledPin, LOW);
delay(100);

delay(5000);          // 在重复 S. O. S 信号前等待 5 秒
}
```

输入代码

上面的写法固然正确，可是是不是觉得有点繁琐呢？如果有个 100 个，难不成还重复 100 遍吗？有没有更好的书写程序的方法呢？想必发明编程的人也考虑到这个问题了，所以有了我们更好的一种写法。我们先来看一下样例代码 2-2。

样例代码 2-2:

```
//项目二 -- S.O.S 信号

int ledPin = 10;

void setup() {
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // 三个快闪烁来表示字母“S”
    for(int x=0;x<3;x++){
        digitalWrite(ledPin, HIGH);    //设置 LED 为开
        delay(150);                    //延时 150 毫秒
        digitalWrite(ledPin, LOW);     //设置 LED 为关
        delay(100);                    //延时 100 毫秒
    }

    //100 毫秒延时产生字母之间的间隙
    delay(100);

    //三个短闪烁来表示字母“O”
    for(int x=0;x<3;x++){
        digitalWrite(ledPin, HIGH);    //设置 LED 为开
        delay(400);                    //延时 400 毫秒
        digitalWrite(ledPin, LOW);     //设置 LED 为关
        delay(100);                    //延时 100 毫秒
    }

    //100 毫秒延时产生字母之间的间隙
    delay(100);
```

```
// 再用三个快闪烁来表示字母“S”
for(int x=0;x<3;x++){
    digitalWrite(ledPin,HIGH);           //设置 LED 为开
    delay(150);                          //延时 150 毫秒
    digitalWrite(ledPin,LOW);            //设置 LED 为关
    delay(100);                          //延时 100 毫秒
}

// 在重复 S.O.S 信号前等待 5 秒
delay(5000);
}
```

在输入代码的时候,注意保持代码的一个层次感,除了美观外,也便于你日后检查代码。

确认正确后,下载代码到 Arduino 中,如果一切顺利的话,我们将看到 LED 闪烁出莫尔斯码 S.O.S 信号,等待 5 秒。重复闪烁。给 Arduino 外接电池,整个装到防水的盒子里,就可以用来发 S.O.S 信号了。S.O.S 通常用于航海或者登山。我们接着来分析下代码。

代码回顾

代码的第一部分与上个项目是完全一样的。也是初始化一个变量,设置数字引脚 10 的模式为输出模式。在主函数 loop()中,你可以看到与上一个项目中类似的语句用来控制 LED 的开和关,并保持一段时间。然而,这次不同的是,主函数包含了三个独立的代码段。

第一段代码是输出是三个点:

```
for(int x=0;x<3;x++){
    digitalWrite(ledPin,HIGH);           //设置 LED 为开
    delay(150);                          //延时 150 毫秒
    digitalWrite(ledPin,LOW);            //设置 LED 为关
    delay(100);                          //延时 100 毫秒
}
```

LED 开关的语句是包含在一对花括号内的,因此为一组代码段。**必须说明的,花括号必须成对出现,如有遗留编译器编译时将不通过。**有个小技巧大家可以学一下,在开始写花括号的时候,就先把“{”“}”都写上,之后再在两个括号之间输入代码,这样就不会出现写到最后括号对应不上的情况。

当程序运行后我们可以看到，灯闪了 3 次而不是只闪了 1 次。产生这样效果的是因为使用了 for 循环。for 语句通常在程序中用作循环使用。我们来看一下：

for 语句格式如下：

```
for (①循环初始化; ②循环条件; ④循环调整语句){  
  ③循环体语句;  
}
```

for 循环顺序如下：

第一轮：1 → 2 → 3 → 4

第二轮：2 → 3 → 4

...

直到 2 不成立，for 循环结束。

来看下我们程序中的 for 循环：

```
for(int x=0;x<3;x++){  
    .....  
}
```

第一步：初始化变量 x=0。

第二步：判断 x 是否小于 3。

第三步：判断第二步成立，for 循环中执行 LED 开与关。

第四步：x 自加，变为 2。

(x++这句话表示把 x 的值增加 1, 等同于写成 x=x+1, 也就是把 x 当前的值变为 x+1, 再赋给 x 一遍。0 变为 1, 第二轮循环则 1 变 2。)

第五步：回到第二步，此时 x=2，判断是否小于 3。

第六步：重复第三步。

.....

直到 x 循环到 3 时，判断 x<3 不成立，自动跳出 for 循环，程序继续往下走。

我们这里需要它循环 3 次，所以设置为 x<3。从 0 开始计算，0 到 2，循环了 3 次。那如果要循环 100 次的话呢？答案：for(int x=0;x<100;x++){ }

我们在写一些判断语句的时候会经常用到一些比较运算符，比如大于，小于等等。下面就说下常用的比较运算符。

比较运算符

“<” 称之为比较运算符。比较运算符在代码中是用作判断的，比较两个值。我们常用的比较运算符有：

- == (等于)
- != (不等于)
- < (小于)
- > (大于)
- <= (小于等于)
- >= (大于等于)

特别要说明一下，等于必须是两个等号。还有像小于等于和大于等于，<和=之间不能留有空格，否则编译不通过。

当然，除了比较运算符外，程序也可以用的+、-、*、/（加、减、乘、除）这些常用的算术运算符。

现在知道 for 循环是如何运作吧！我们代码中有 3 个 for 循环：第一个 for 循环 3 次，长闪烁 3 次，代表输出 3 个点，也就是字母“S”。第二个 for 循环同样循环 3 次，短闪烁 3 次，代码输出 3 个横杠，也就是字母“O”。第三个 for 循环又来输出个“S”。

必须要注意的，我们这里要引用一个新的概念，是局部变量和全局变量。**局部变量，这类变量只在自己的代码内起作用。**就像我们这里 for 循环中的变量 x，它就是个局部变量，所以说，虽然每个 for 循环中都有一个变量 x，但它们不冲突就是这样原因，它们只在自己的循环中执行。还有一种变量叫**全局变量，不同之处是，它能在整个程序中起作用，但条件是，必须在 setup()、loop()函数外声明。**就像我们这里的 ledPin，能在整个程序中起作用。

在每 for 循环之间有个小延时 100 毫秒，是 S.O.S 字母之间有个清晰的停顿说明。最后，在回到主函数 loop 重新执行一遍之前，有个相对较长的延时，为 5 秒。

好了，我们 S.O.S 信号源项目就算告一个段落了。有所收获吗？

课后作业

我们学习了两个项目的基础，现在做个课后习题吧~做个交通信号灯，下图是整个一个运行过程，虚线框的是程序循环的部分。

提示：以上我们是只点亮的一个 LED 灯，现在需要点亮三个 LED 灯。电路连接的原理是和一个灯相同，程序中需要改变的用三个数字口来分别控制 3 个 LED 灯。自己动手试一下吧！

