

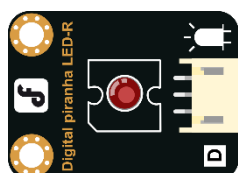
项目一 点亮一盏灯

在前面几篇前奏中，我们已经对 Arduino 有了简单的了解，整个装置工作是依赖于哪些部分。也了解了电子世界最重要两个量，数字量与模拟量。接下来我们就正式开始做东西了，第一个要做的必须是最经典的，最经典的莫过于“blink”。

其实，前面在一开始驱动安装的时候就用过这段代码了，区别在于这里将不使用板子上的 LED 13(也就是“L”灯)，而是在数字引脚 13 连接一个 LED。

所需元件

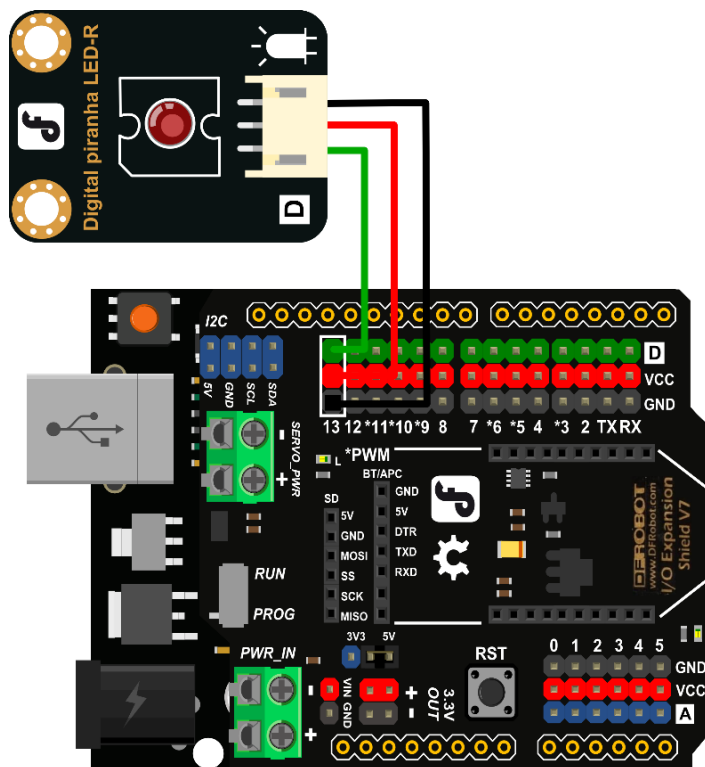
- 1× 数字食人鱼红色 LED 发光模块



*之后的项目将都不重复罗列 UNO 与 I/O 传感器扩展板，但是！每次都还是需要用到的。

硬件连接

直接把食人鱼红色模块连接到 Arduino UNO 的数字口 13。插上 USB 线，准备下载程序。



输入代码

打开 Arduino IDE，在编辑框中输入样例代码 1-1 所示代码。（输入代码也是一种学习编程的过程，虽然提供代码的压缩包，但还是建议初学者自己输入代码，亲身体验一下。）

样例代码 1-1:

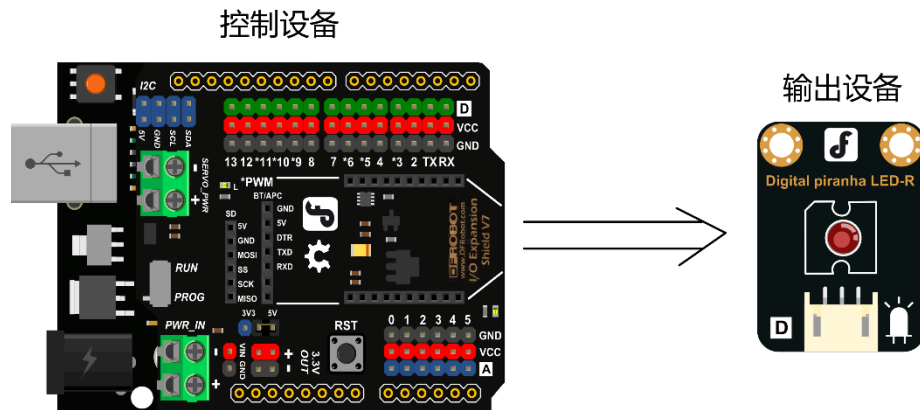
```
//项目一 —— LED 闪烁
/*
  描述：LED 每隔一秒交替亮灭一次
*/
int ledPin = 13;
void setup() {
    pinMode(ledPin, OUTPUT);
}
void loop() {
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```

输入完毕后，点击 IDE 的“校验 (Verify)”，查看输入代码是否通过编译。如果显示没有错误，单击“下载 (Upload)”，给 Arduino 下载代码。以上每一步都完成了的话，你应该可以看到面包板上的红色 LED 每隔一秒交替亮灭一次。

现在让我们来回顾一下代码，看看它们是如何工作的。

硬件分析（数字输出）

我们从前面几章说的输入输出的角度来看。整个装置只有两个部分，控制与输出。Arduino 就是控制设备，LED 发光模块就是输出设备。对的，这个整个装置是没有输入设备的。有了这么一分析，我们再看代码就不那么难理解了。



代码分析

先说下 Arduino 代码必须具备的两个组成部分：

```
void setup() {  
    // 写入 setup 代码，只运行一次：  
}  
  
void loop() {  
    // 写入 main 代码，重复运行：  
}
```

Arduino 代码必须包含 `setup()` 和 `loop()` 这两个函数。`setup` 英文中是“设置”的意思。所以 `setup()` 函数是用于一些初始化设置的，只在代码一开始时，运行一次。`loop` 是“循环”的意思，只要 Arduino 不掉电，`loop` 就会不停的重复运行。

由于 LED 是输出设备，所以不难看出，在 `setup()` 函数中先初始化 LED 为输出模式。

函数格式如下：

```
pinMode(pin, mode)
```

这个函数是用来设置 Arduino 数字引脚的模式，只用于数字引脚定义是输入(INPUT)还是输出(OUTPUT)。pin 指数字引脚号，mode 指引脚模式(OUTPUT/INPUT)。

回头看下代码中，

```
pinMode(ledPin, OUTPUT);
```

这句话的意思就是，将 ledPin 设置为输出模式，中间的逗号可不能省。那 ledPin 是什么呢？

看下代码的第一句话：

```
int ledPin = 13;
```

我们在一开始的时候给 13 号引脚起了个名字叫做 ledPin，所以 ledPin 就代表了 13 号引脚。**前面的 int 可不能少！** int 代表了 ledPin 是个整数。

明白了这两句话的意思了，如果我们现在需要换个引脚，LED 不连接到 13 号引脚，连接到 10 引脚，可以怎么写：

```
pinMode(10, OUTPUT);
```

只需把 pin 换成对应的引脚号就行了。

再看下 loop() 函数，loop 函数中就只用到了一个函数 digitalWrite()。

函数格式如下：

digitalWrite(pin, value)

这个函数的意义是：引脚 pin 在 pinMode() 的中被设置为 OUTPUT 模式时，其电压将被设置为相应的值，HIGH 为 5V (3.3V 控制板上为 3.3V)，LOW 为 0V。

```
digitalWrite(ledPin, HIGH); //LED 被点亮
```

```
digitalWrite(ledPin, LOW); //LED 被熄灭
```

代码中的，ledPin 同样指引脚。写入 HIGH 时，引脚 13 就被至高，LED 被点亮。写入 LOW 时，引脚 13 就被拉低，LED 被熄灭。

亮与灭直接还有句语句：

```
delay(1000);
```

delay 是延时的意思。括号中写入的是毫秒(ms)。所以，delay(1000)就是延时 1s 的意思。最后实现的就是 LED 亮一秒，灭一秒，一直无限循环。

细心的朋友可能注意到，代码开始部分有段带 “//” 和 “/*...*/” 的文字：

```
//项目一 — LED 闪烁
/*
    描述：LED 每隔一秒交替亮灭一次
*/
```

这是代码中的说明文字，可以叫做注释。是以 “//” 开始，这个符号所在行之后的文字将不被编译器编译。

还有另外一种写注释的方式，用 “/*...*/”，这个符号的作用是可以注释多行，这也是与上一种注释方式的区别之处。在 /*和*/ 中间的所有内容都将被编译器忽略，不进行编译。IDE 将自动把注释的文字颜色变为灰色。

趣味练习

能否试试变换 LED 的亮灭速度，让 LED 保持关闭 5 秒，然后快速闪烁一下（250 毫秒），就像汽车报警器上的 LED 指示灯那样。

通过改变 LED 开和关的时间，可以产生不同的效果，开关时间短，则感觉动感，开关时间长，则感觉柔和。