

更多原创作品尽在电路城：<http://www.cirmall.com/>

## Arduino 项目教程(3)—Mini 台灯

台灯，是我们再常见不过的东西。“啪”一下开，“啪”一下关。Mini 台灯的功能就和台灯类似，按钮就像是 LED 的开关，每按一下，就会切换 LED 的状态。做完之后再给小灯来个壳儿，一定很 Q。

### 所需元件

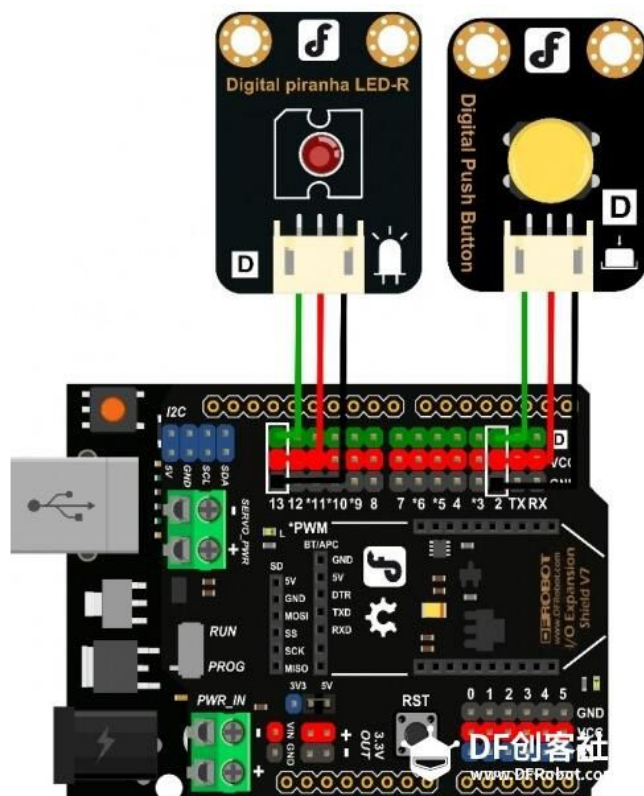
1× 数字食人鱼红色 LED 发光模块

1× 数字大按钮模块

### 硬件连接

数字大按钮 → 数字 2

数字食人鱼红色 LED 发光模块 → 数字 13



### 输入代码

样例代码 3-1：

更多原创作品尽在电路城: <http://www.cirmall.com/>

//项目三 —— 小台灯

```
int buttonPin = 2;           //按钮连接到数字 2

int ledPin = 13;             //LED 连接到数字 13

int ledState = HIGH;         // ledState 记录 LED 状态

int buttonState;             // buttonState 记录按键状态

int lastButtonState = LOW;    // lastbuttonState 记录按键前一个状态

long lastDebounceTime = 0;

long debounceDelay = 50;     //去除抖动时间

void setup() {

    pinMode(buttonPin, INPUT);

    pinMode(ledPin, OUTPUT);

    digitalWrite(ledPin, ledState);

}

void loop() {

    //reading 用来存储 buttonPin 的数据

    int reading = digitalRead(buttonPin);

    // 一旦检测到数据发生变化, 记录当前时间

    if (reading != lastButtonState) {

        lastDebounceTime= millis();

    }

    // 等待 50ms, 再进行一次判断, 是否和当前 button 状态相同

    // 如果和当前状态不相同, 改变 button 状态
```

更多原创作品尽在电路城：<http://www.cirmall.com/>

```
// 同时，如果 button 状态为高（也就是被按下），那么就改变 led 的状态
```

```
if ((millis() - lastDebounceTime) > debounceDelay) {
```

```
    if (reading != buttonState) {
```

```
        buttonState = reading;
```

```
        if (buttonState == HIGH) {
```

```
            ledState= !ledState;
```

```
        }
```

```
    }
```

```
}
```

```
digitalWrite(ledPin, ledState);
```

```
// 改变 button 前一个状态值
```

```
lastButtonState = reading;
```

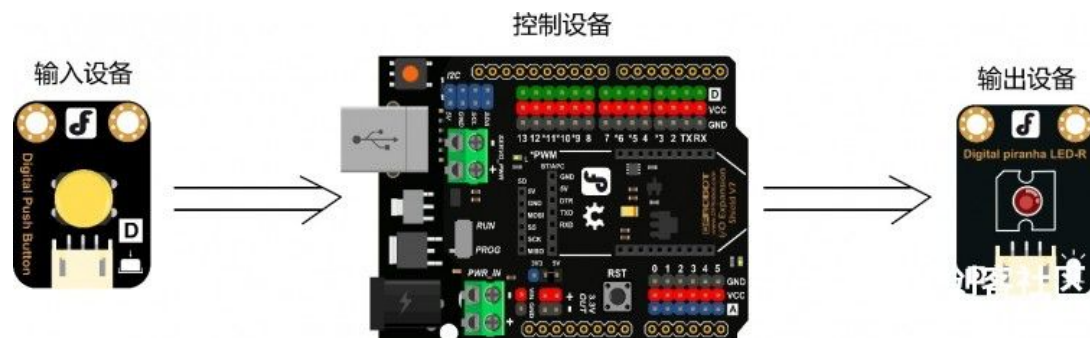
```
}
```

下载完代码，按下按钮，灯点亮。再按下按钮，灯熄灭。是不是很像个小台灯？

### 硬件分析（数字输入—数字输出）

很明显，大按钮是输入设备，LED 是输出设备。和前面感应灯类似，也是一个数字输入控

制一个数字输出。只是形式与代码有所不同。



### 代码回顾

更多原创作品尽在电路城：<http://www.cirmall.com/>

由硬件分析可以看出，按键是输入设备，LED 是输出设备。

```
pinMode(buttonPin, INPUT);
```

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

通过 `digitalWrite()` 读取按键的状态:

```
int reading = digitalRead(buttonPin);
```

按键在由低变高或者由高变低时，都会有个抖动的过程，时间非常的短，如下图所示：



为了避免由于抖动产生的错误信号，所以我们代码中有个去抖的过程。去抖的方法很简单，就是等到数据发生变化时，隔一段时间再检测一次。

一旦检测到读取的数据发生变化，通过 `millis()` 函数记下时间：

```
if (reading != lastButtonState) {  
    lastDebounceTime = millis();  
}
```

**`millis()`** 是一个函数，该函数是 Arduino 语言自有的函数，它返回值是一个时间，**Arduino 开始运行到执行到当前的时间，也称之为机器时间**，就像一个隐形时钟，从控制器开始运行的那一刻起开始计时，**以毫秒为单位**。

再等待 50ms，再进行一次判断，是否和当前 button 状态相同。如果和当前状态不相同，改变 button 状态。同时，如果 button 状态为高（也就是被按下），那么就改变 LED 的状态。

更多原创作品尽在电路城: <http://www.cirmall.com/>

```
if ((millis() - lastDebounceTime) > debounceDelay) {  
  
    if (reading != buttonState) {  
  
        buttonState = reading;  
  
        if (buttonState == HIGH) {  
  
            ledState = !ledState;  
  
        }  
  
    }  
  
}
```

### 趣味练习

#### 灯光门铃

现在越来越多年轻人回家就塞上耳机，即使在家都听不见门铃声，那就自制一个灯光门铃，有人来了，灯就开始狂闪，提醒里面的人，门口有人在按门铃了。这样的门铃也同样适用于那些耳朵不好的老人，又或者是那些聋哑人士。