

03

第三章

使用按键点亮RGB灯



DFROBOT
DRIVE THE FUTURE

前情回顾

在上个章节里，我们了解了什么是变量，如何使用for循环语句和RGB灯控制函数

本章内容

实现按一次按键，点亮MAX背上的1个RGB灯；

本章知识点

1. 按键的原理；
2. 如何使用if语句；
3. “==”和“=”的区别；



一、编写并下载程序

打开 **ArduinoIDE**，将下面的代码输入到编辑区中

```
/*  
程序功能：按一次按键，主板上的 1 个 RGB 灯发红光。  
作 者：DFRobot  
*/  
#include <DFRobot_MAX.h> //载入 MAX 驱动库  
DFRobot_MAX myMax; //载入 myMax 函数  
  
void setup() {  

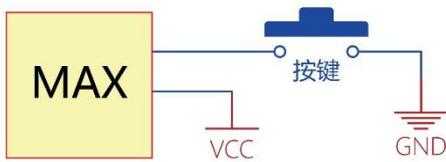
```

上传成功后，打开电源开关，看见 RGB 灯是熄灭的，按下按键之后，RGB 灯亮红色，松开按键 RGB 灯熄灭。

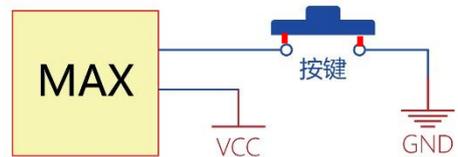


二、按键的原理

按键，可实现电路的导通与断开。MAX 按键的状态分别为“按键按下”和“按键未按下”。图中 VCC (1) 表示电源，GND (0) 表示接地。然而在数字逻辑电路中，我们把高电平用“1”表示，低电平用“0”表示。



按键未按下



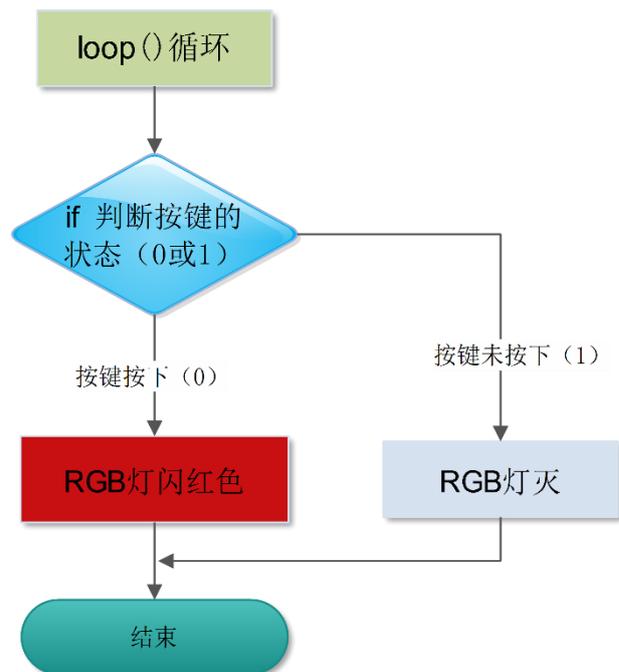
按键按下

如上图所示，按键有两种状态，按下和未按下。当 MAX 检测到信号“1”时，按键处于断开状态，如图 5.1 所示；当 MAX 检测到信号“0”时，按键处于闭合状态，如图 5.2 所示，即可导通电路，点亮 LED 灯。由此可以看出，MAX 是低电平有效。

注意：此处的“0”和“1”不再是数学概念上的数量表示方法，而是数字信号中的两种状态。

三、代码回顾

经过前面几章的学习，我们已经知道程序的运行过程是：载入驱动库和驱动函数→程序初始化→进入 loop 循环……，其主要流程见右图。



通过前面两章的学习，我们应该能理解本章代码的大部分内容了。进入 `loop()` 函数，发现了一个新的函数：

```
myMax.buttonValue();
```

这个函数是用来读取按键的状态，当状态为“0”时，说明按键被按下了，当状态为“1”，表明按键未按下。所以可以直接检查 `myMax.buttonValue()` 的值来判断按键是否被按下：

```
例程：if(myMax.buttonValue()==0){
    myMax.rgbLed(1,255,0,0);
}
```

从例程知道判断按键的状态时用到一个新的语句——if 语句。

```
if (表达式) {
    语句;
}
```

表达式是指我们的判断条件，通常为一些关系式或者逻辑式，也可以直接表示某一数值。如果 if 表达式条件为真则执行 if 中的语句。表达式条件为假，则跳出 if 语句。

例程语义：如果 MAX 按键按下(表达式条件为真)，RGB 灯就亮红色。如果 MAX 按键未按下(条件为假)，就跳出 if 语句。

“==”和“=”运算符的区别

1、“==”是一种比较运算符，用于判断两边是否相等。注意是“双等号”哦！

例：if (a==1)，就是如果 a 等于 1，这个条件才成立。

2、“=”是赋值运算符，把等号右边的值赋给左边。

例：a=1，就是将 1 赋值给变量 a。

一般常用的运算符有：

- +、-、*、/、%（加、减、乘、除、求余）
- =（赋值）
- <（小于）
- >（大于）
- <=（小于等于）
- >=（大于等于）
- ==（等于）
- !=（不等于）
- &&（逻辑与）
- ||（逻辑或）
- !（非运算）

这几种运算符之间不能留有空格，否则会编译不通过。

具体用法可参考[语法手册](#)

课后思考

将下面的代码输入到编辑区中，编译并上传到 MAX 上，然后按一下功能按键，看看它能实现什么功能？并试着分析这些代码的含义。

```
#include <DFRobot_MAX.h>
DFRobot_MAX myMax;
int a;
int b=0;

void setup() {
  myMax.begin();
}

void loop() {
  a=myMax.buttonValue();
  if(a==0 && b<4){
    myMax.rgbLed(b,255,0,0);
    b++;
    delay(300);
  }
  if(b==4){
    b=0;
    for(int i=0;i<4;i++){
      myMax.rgbLed(i,0,0,0);
    }
  }
}
```

在实际生活中，有很多地方都用到了按键，比如键盘。它的玩法有很多样，然而应该怎样玩？就要靠你的想象了。