

02

第二章
MAX的炫彩RGB灯



DFROBOT
DRIVE THE FUTURE

前情回顾

在上个章节里，学习了注释的使用、函数的定义、`setup()`函数和`loop()`函数、以及延时函数的使用。

本章内容

让MAX在行驶过程中，循环点亮4个RGB灯。

本章知识点

1. 变量的定义；
2. `for`循环语句的使用；
3. RGB灯控制函数的使用；



一、编写及下载程序

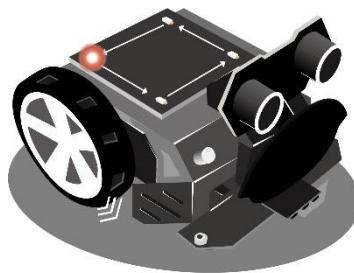
打开 **ArduinoIDE**，将下面的代码输入到编辑区中

```
/*
程序功能：主板上的 4 个 RGB 灯依次发出红色的灯光。
作 者：DFRobot
*/
#include <DFRobot_MAX.h> //载入 MAX 驱动库
DFRobot_MAX myMax; //载入 MAX 驱动程序
int i; //定义一个变量“i”

void setup() {
    myMax.begin(); //初始化 MAX 系统
    myMax.forward(150,150); //设置 MAX 按 150 的速度前进
}

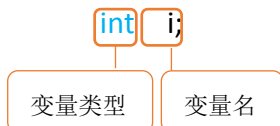
void loop() {
    for(i=0;i<4;i++){ //for 循环语句，只要变量“i”小于 4，循环一次就累加 1 。
        myMax.rgbLed(i,255,0,0); //for 循环语句内，逐个的点亮 RGB 灯为红色。
        delay(1000); //间隔 1 秒钟。
        myMax.rgbLed(i,0,0,0); //再将对应的 RGB 灯熄灭。
    }
}
```

点击“验证”按钮编译程序，用 USB 线将 MAX 和电脑连接在一起，点击“上传”按钮将程序上传到 MAX 机器人。打开电源开关后，我们能看见 MAX 主板上的 RGB 灯依次循环发出红色的灯光。



二、代码回顾

首先还是先载入 MAX 驱动库和驱动程序，接着定义一个变量“i”；



这就是变量声明，变量是用来储存数据的，而在本例中，我们所用的类型是 int 型，也就是所谓的整型变量。可以表示-32768 到 32767 之间的数。然而变量的类型，是取决于你储存的内容来决定的。

在声明的最后用一个“;”来表示这句话的结束，每句话的结束分号是必不可少的，而且必须是切换到英文输入法中的分号。

什么是变量？

变量好比一个盒子，盒子的空间用来存放东西的，想要放的东西一定要比盒子小，那样才放的下。变量也是一样，你存储的数据一定要在变量的范围内。

之所以叫变量，是因为程序运行过程中，可以改变它的值。程序中，有时候会对变量值进行数字计算，变量的值也会随之发生变化。

变量的申明方法

```
int i;  
int i=2;
```

以上就是变量的基本申明方法，**int 是变量的类型，i 是指变量的名称**。第二种方法中，我们直接给变量 i 赋予了一个默认值为 2 。

在给变量起名字的时候，必须以字母开头，之后可以包含字母、数字、下划线。最好起一些自己一眼能看懂的变量。**注意 C 语言认为大小写字母是不同的。C 语言中还有一些特定的名称也是不能使用的，比如 main, if, while 等。**为了避免这些特定名称作为变量名，所有这些名称在程序中显示为橙色。

变量的数据类型

为了合理的利用储存空间，我们需要给数据定一个类型，就像我们装不同的东西，需要用到不同大小的盒子一样的道理。我们现在最常用的几种数据类型，后面我们会接触到更多的数据类型。

基本变量的数据类型有 int（整型）、float（单精度浮点型）、double（双精度浮点型）、char（字符类型）、long（长整型）、short（短整型）。具体的范围，可以查看数据手册。

回到代码中，在 `setup()`函数中设置 MAX 的前进速度为 150。然后进入到 `loop()`循环。程序在运行的时候，我们可以看见 `rgbLed` 灯被循环点亮。能产生这样的效果主要是因为使用了 `for` 循环。

for 循环的一般形式为：

```
for(表达式 1 ;表达式 2 ;表达式 3 )
{
  循环体语句;
}
```

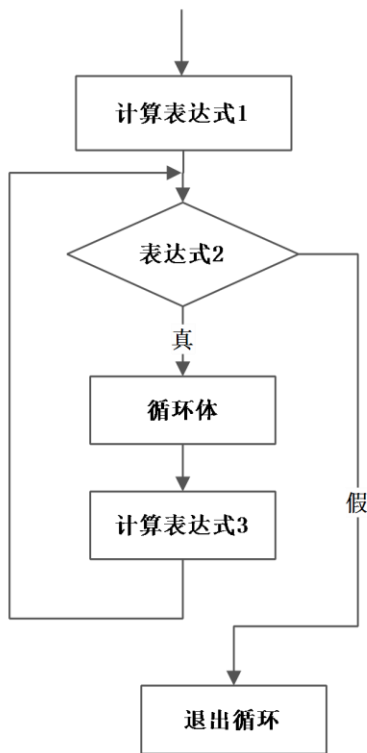
表达式 1：一般为赋值表达式，给控制变量赋初值；

表达式 2：关系表达式或逻辑表达式，循环控制条件；

表达式 3：一般为赋值表达式，给控制变量增量或减量；

语句：循环体，当有多条语句时，必须使用复合语句。

执行过程如下图：



for循环

在本例中，我们是这样循环的：

```
for(i = 0; i < 4; i ++){
  .....
}
```

第一步：初始化变量 `i=0`。

第二步：判断 `i` 是否 `<4`。

第三步：判断第二步成立，`for` 循环中执行 RGB 灯的亮与灭。

第四步：`i` 自加，变为 2。（`i++`这句话表示把 `i` 的值增加 1，等同于写成 `i=i+1`，也就是把 `i` 当前的值变为 `i+1`，再赋给 `i` 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 `i=2`，判断 `i` 是否 `<4`，再次自加，`i=3`。

.....

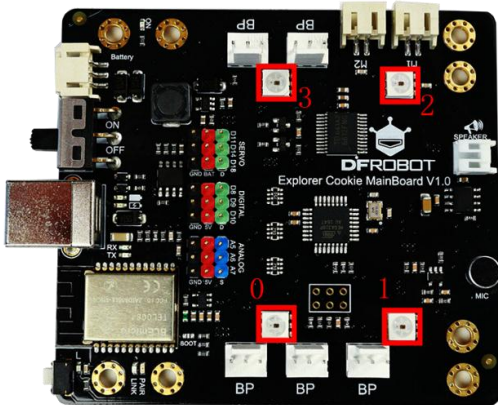
直到 `i` 循环到 4 时，判断 `i<4` 不成立，自动跳出 `for` 循环，程序继续往下走。

这样就能循环点亮 MAX 身上的 RGB 灯了。

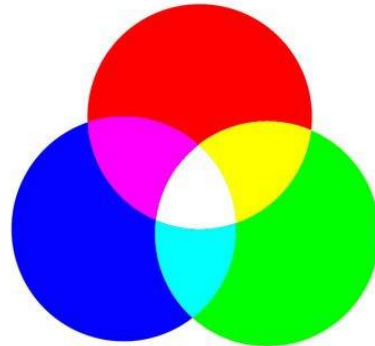
三、RGB 灯函数的使用

在本例中，我们用到了 `myMax.rgbLed(i, 255, 0, 0)`；这样一条语句。看到上面的注释，想必大家对这条语句的作用也有大致的了解了。我们来详细介绍下这条语句的用法：

`myMax.rgbLed`(序号, 红色值, 绿色值, 蓝色值);
函数中序号对应着 MAX 主板上的 4 个 RGB 灯，
如图所示：序号从 0 到 3，共 4 个 RGB 灯。



在函数的参数里，需要填上三个颜色值，我们知道颜色是由红、绿、蓝三基色构成，他们的范围是 0~255 之间，通过三种不同颜色的组合，就可以显示各种各样的色彩。



如果让 0 号 rgbLed 灯依次循环显示紫色、黄色、青色？

```
代码例程: #include <DFRobot_MAX.h>           //载入 MAX 驱动库
           DFRobot_MAX myMax;                 //载入 MAX 驱动程序

           void setup() {
               myMax.begin();                 //初始化 MAX 系统
           }

           void loop() {
               myMax.rgbLed(0,138,43,226);    //0 号灯显示紫色
               delay(1000);                   //间隔 1 秒钟
               myMax.rgbLed(0,255,255,0);    //0 号灯显示黄色
               delay(1000);                   //间隔 1 秒钟
               myMax.rgbLed(0,124,252,0);    //0 号灯显示草坪绿
               delay(1000);                   //间隔 1 秒钟
           }
```

通过上面的代码例程，了解到通过调整 R、G、B 三个基准颜色的值，可以组合出了更多的颜色。

课后思考

1、如果想要循环 100 次呢，for 语句应该怎样书写呢？

在本章我们学习了利用 for 循环语句从 0 开始计算，0 到 3，循环了 4 次点亮 rgbLed 灯，如果你手上有 100 颗 rgbLed 灯，你怎样让它循环点亮呢？

2、依次循环点亮 4 颗灯，并且让每颗灯亮的颜色不一样。

在本章节我们学习了通过 R、G、B 三个基准颜色点亮了 0 号 rgbLed 灯，并让它循环显示三种颜色，如果想要依次循环点亮 4 颗灯，并且每颗灯的颜色不一样。应该怎么做呢？