

源木工坊编程课程

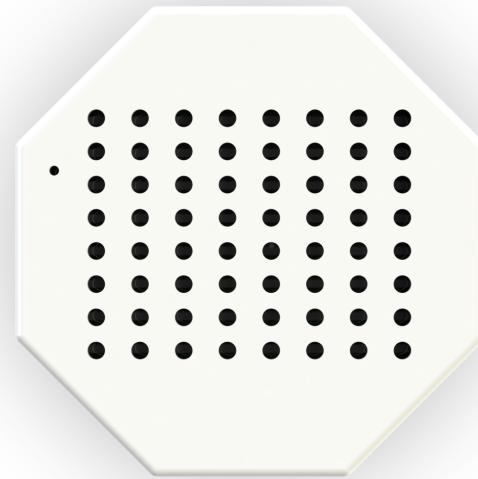
第八课 循环控制与逻辑判断



在第六课中，我们使用了变量来代替LED的X坐标，但按照上图的程序，我们多次使用了变量，十分繁琐，那有没有什么简单的方法呢？

接下来我们就要学习循环控制，来让系统自动运行部分程序。

前期准备



编程模块

DPM-1



基本控制库

BasicControl 3.0.0

循环的类型

循环控制可以让程序在某种情况下自动循环运行。常用的循环指令有以下三种：



①



②



③

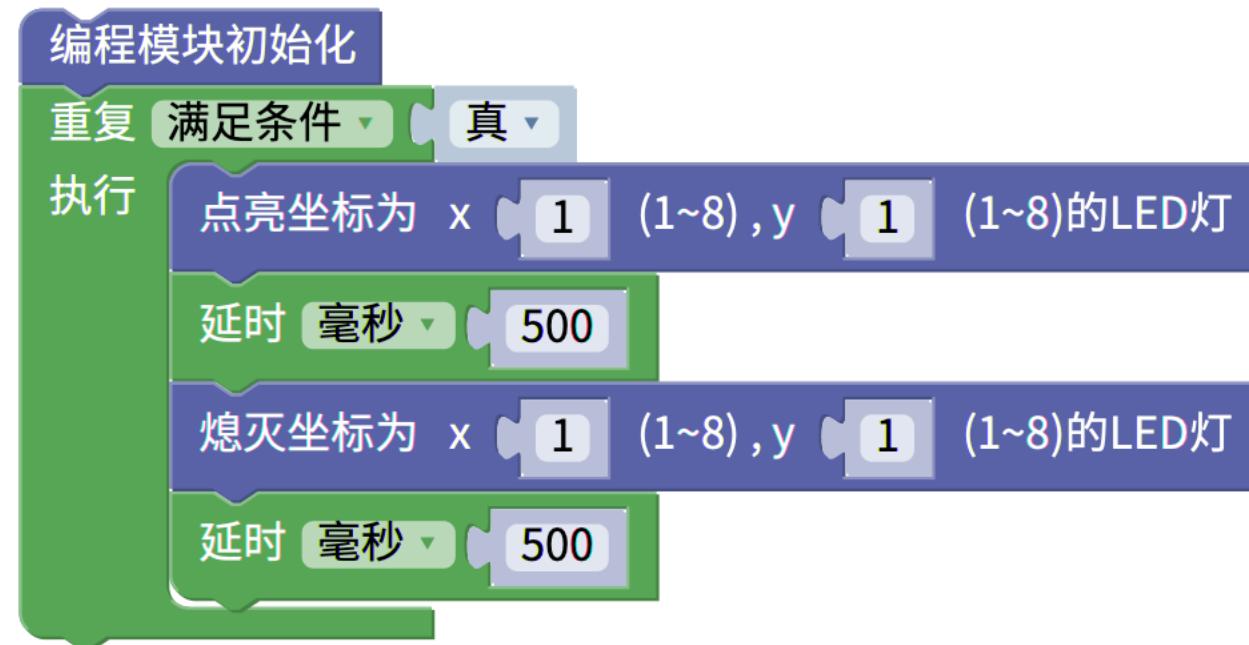
循环①：在程序语言中叫做while循环，当程序满足某种条件时，则不断循环执行模块内部程序，直到不再满足该条件；

循环②：在程序语言中叫做do循环，与while循环稍有不同，在do循环中，系统会先执行一遍内部程序，再判断程序是否满足条件，如果满足则继续循环，如果不满足，则停止循环到后面的程序。

循环③：在程序语言中叫做for循环，for循环可以指定循环的次数，通过计步的方式，每执行一次程序，步数便加1，直到累计步数超过设定值，则循环停止，执行后面的程序。

简单的while循环

一个简单的while循环如下图所示。尝试上传到DPM-1编程模块，看看有什么效果？



可以看到，我们只需几行程序，就能让LED灯一直不停的闪烁，按照1秒的周期，LED灯循环点亮、熄灭。

循环判断

与条件判断语句相同，循环语句的执行也需要判定是否满足条件。我们看到默认的循环模块后面自带了一个值，也可以再拼接新的模块。



还记得变量类型中有一种类型叫做“布尔”吗？布尔型变量是一种只有两种值的变量，即“真”和“假”。

真，在布尔型变量中记做“1”；假，在布尔型变量中记做“0”；循环模块的判定条件就是一种布尔值。只有在特定的布尔变量值的情况下，循环模块才能执行内部程序。

而实际上，判定值只要不是“0”，系统都会判定为真，这也简化了我们的程序。

还记得上一课我们使用过的这个语句吗？这就是一个布尔型判断语句，当 $\text{LedX} \leq 4$ 的时候，该语句输出布尔值为真“1”，而当 $\text{LedX} > 4$ 时，该语句便输出布尔值为假“0”。

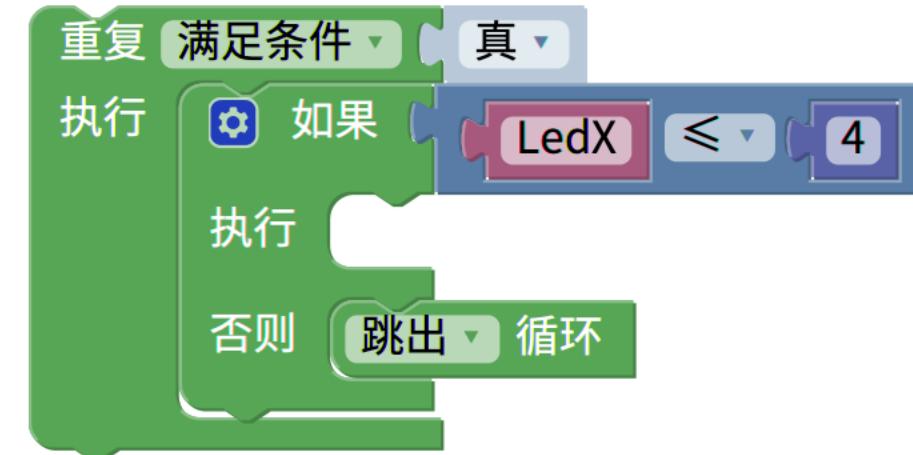
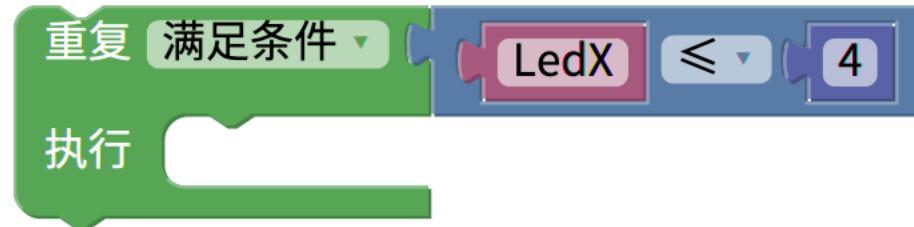


循环的终止

循环控制会在每次循环的时候检查判定条件是否满足，如果判定条件不再满足，则循环停止。

因此，只要我们让LedX小于或等于4，就可让循环执行，通过程序让LedX大于4，就能让循环终止。

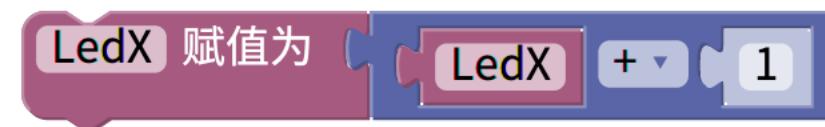
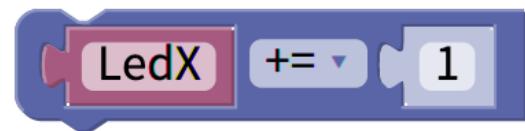
我们还可以通过跳出循环指令来提前终止循环，结合if语句，也能实现循环判断的效果，因此，下面的两种循环写法是相同效果的。



自增运算

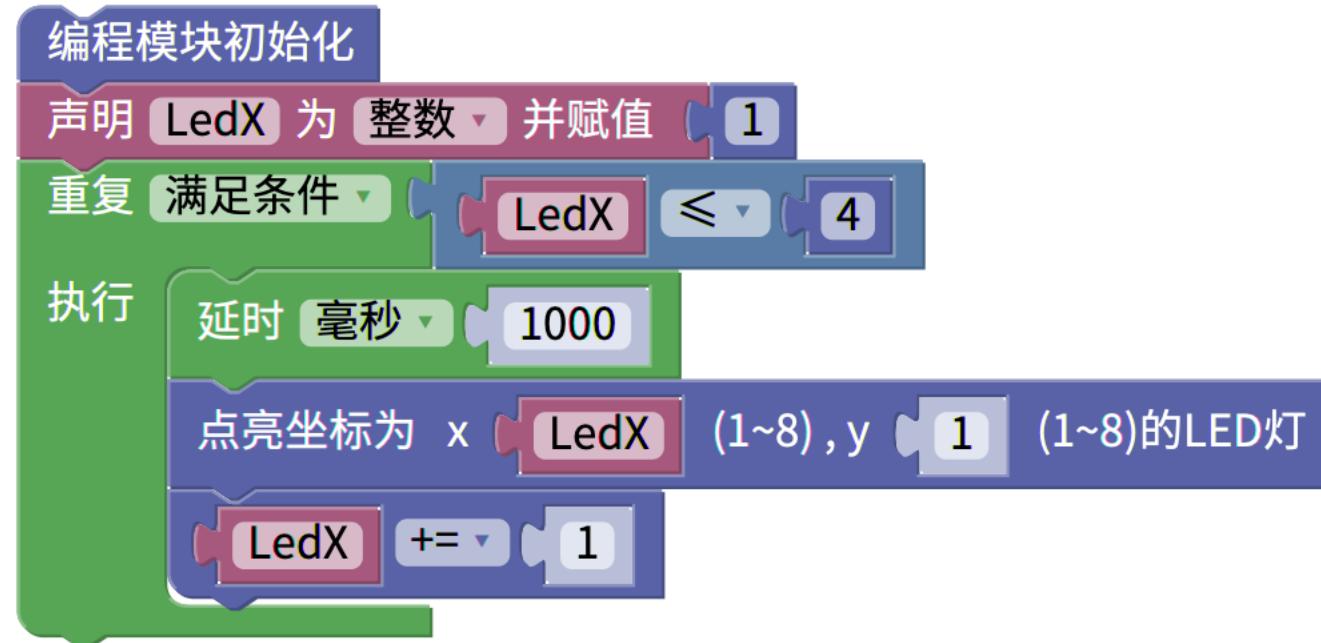
我们在初始的时候赋值LedX为1，我们希望每一个循环，LedX自动增加1，循环执行4次后，LedX的值为 $5 > 4$ ，则终止循环。

让LedX在每个循环内自动增加的运算叫做自增预算，如下左图模块所示。该模块每执行依次便增加1。自增运算还有一种写法，如下右图所示。



另外还有自减、自乘、自除运算，可以通过选择模块中间的运算符来设置。

最终程序



那么，回到最初的程序，我们就可以用这样的循环控制来实现依次点亮LED等。我们只需将4改成8，也便能实现依次点亮8盏灯。

循环②：do循环

了解了循环、逻辑和判定的相关知识后，自己尝试一下使用do循环来改写你的程序吧。



谢谢观看

MIU