

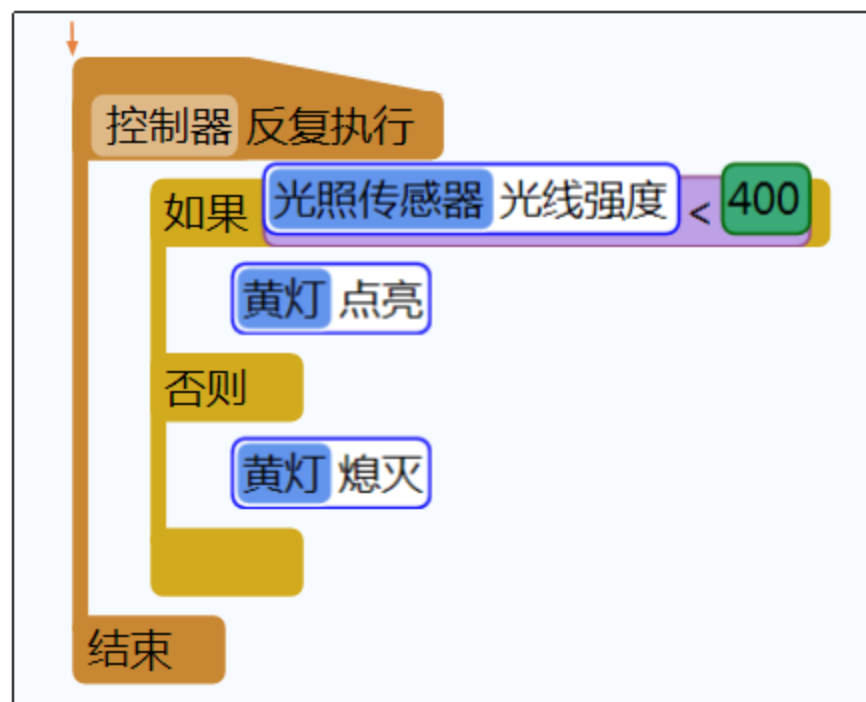
15



多线程（事件）

1. 了解事件的概念
2. 事件触发的应用
3. 了解多线程的概念

- 在前边的章节中我们讲到可以通过条件判断对特殊情况进行处理，但是条件判断适用于数值型类型的环境量，如超声波测到的距离、光照传感器测到的光线强度等。数值型环境量是一种连续型数据，所以可以通过循环检测判断来实现环境量的监测。如图中的程序可以实现对光线强度的监测，因为光线强度这个环境量一直都在，而且光线强度小于400，或者光线强度不小于400这两个条件总有一个是成立的。



- 但是还有一种环境量为触发型，这些环境量出现时间、频率不固定，而且每次出现时间很短，如按键按下、声音检测传感器检测到声音、光电避障传感器检测到障碍物等。如果使用循环检测判断，经常会捕捉不到事情的发生。因为只有在事件发生时，刚好执行到“如果...”条件判断指令，才能捕捉到。
- 如图中的程序，例如只有在程序刚好执行到“如果蓝色按钮按下”这条指令时，按下蓝色按钮才能检测到蓝色按钮按下，黄灯才会点亮。而且程序的执行是非常快的，所以很难检测到。



- 针对触发型环境量，在编写程序时通过编写事件触发程序来实现监测。只要事件发生，立即触发相应的程序。而且只有在事件触发时才会执行相应的程序。
- 以按钮为例，如图1所示，点击按钮图片，在弹出的对话框中，点击“按钮按下时”；出现如图2所示的指令；编写按钮按下时红灯点亮30秒的程序，如图3所示。



图1

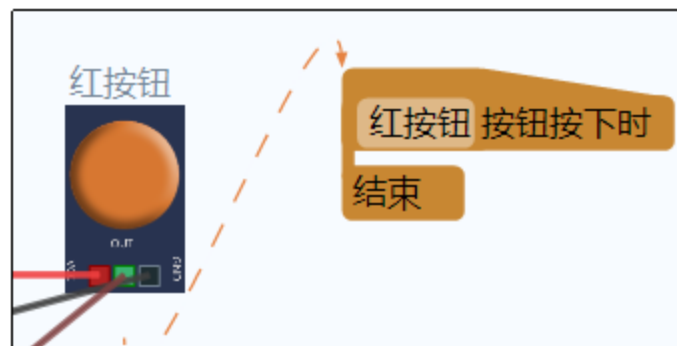


图2

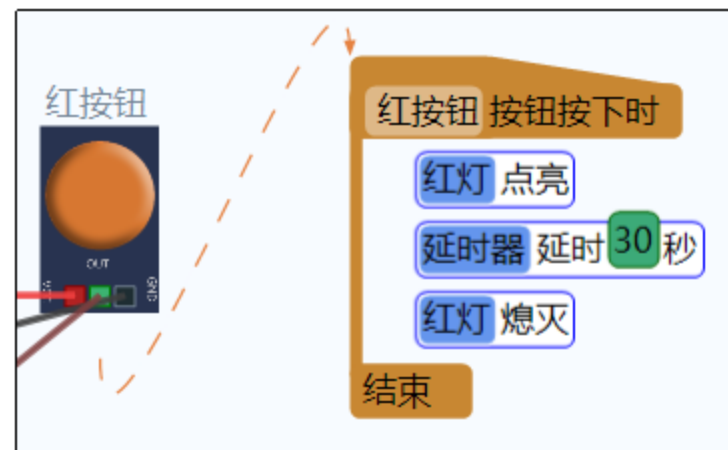
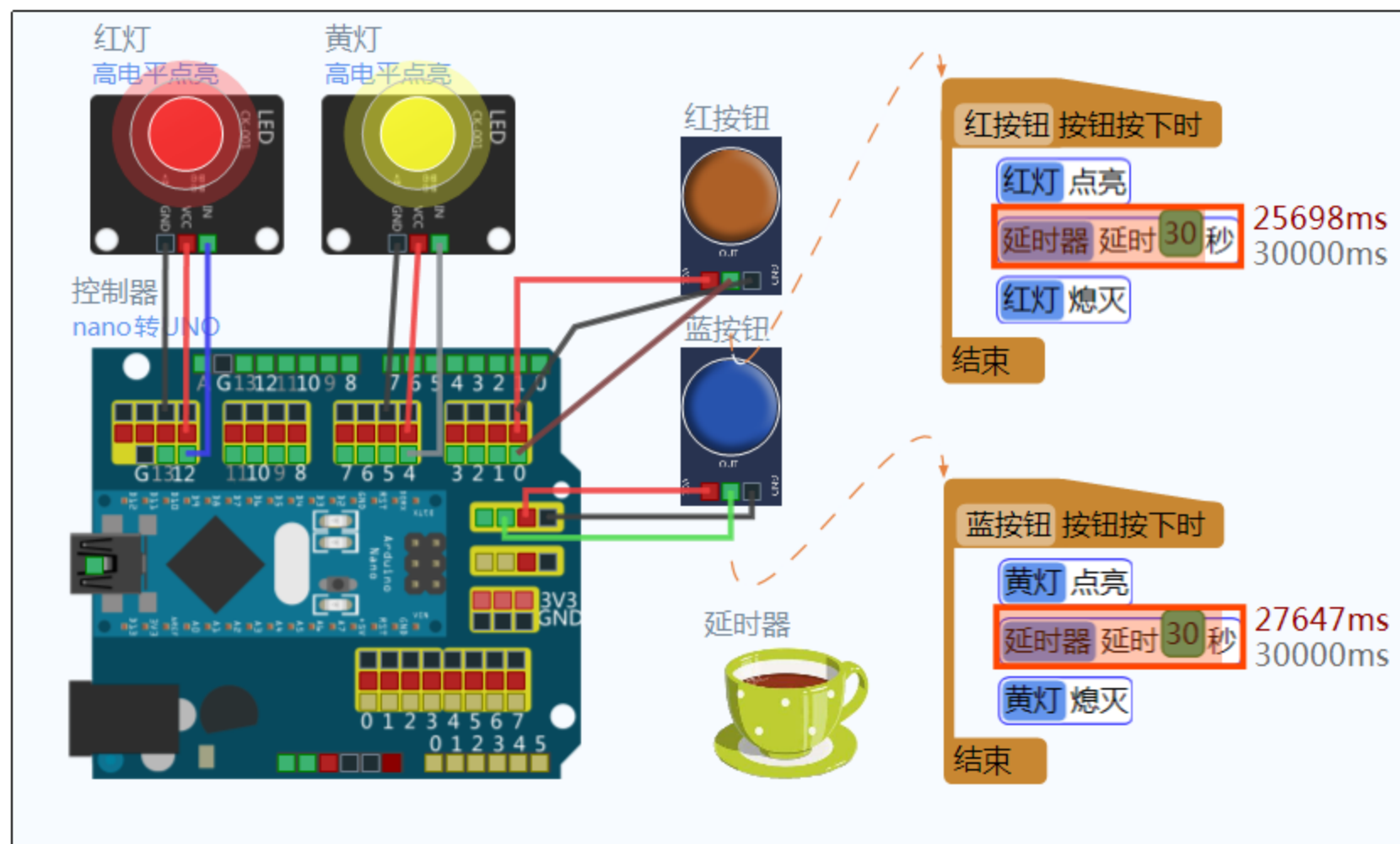


图3

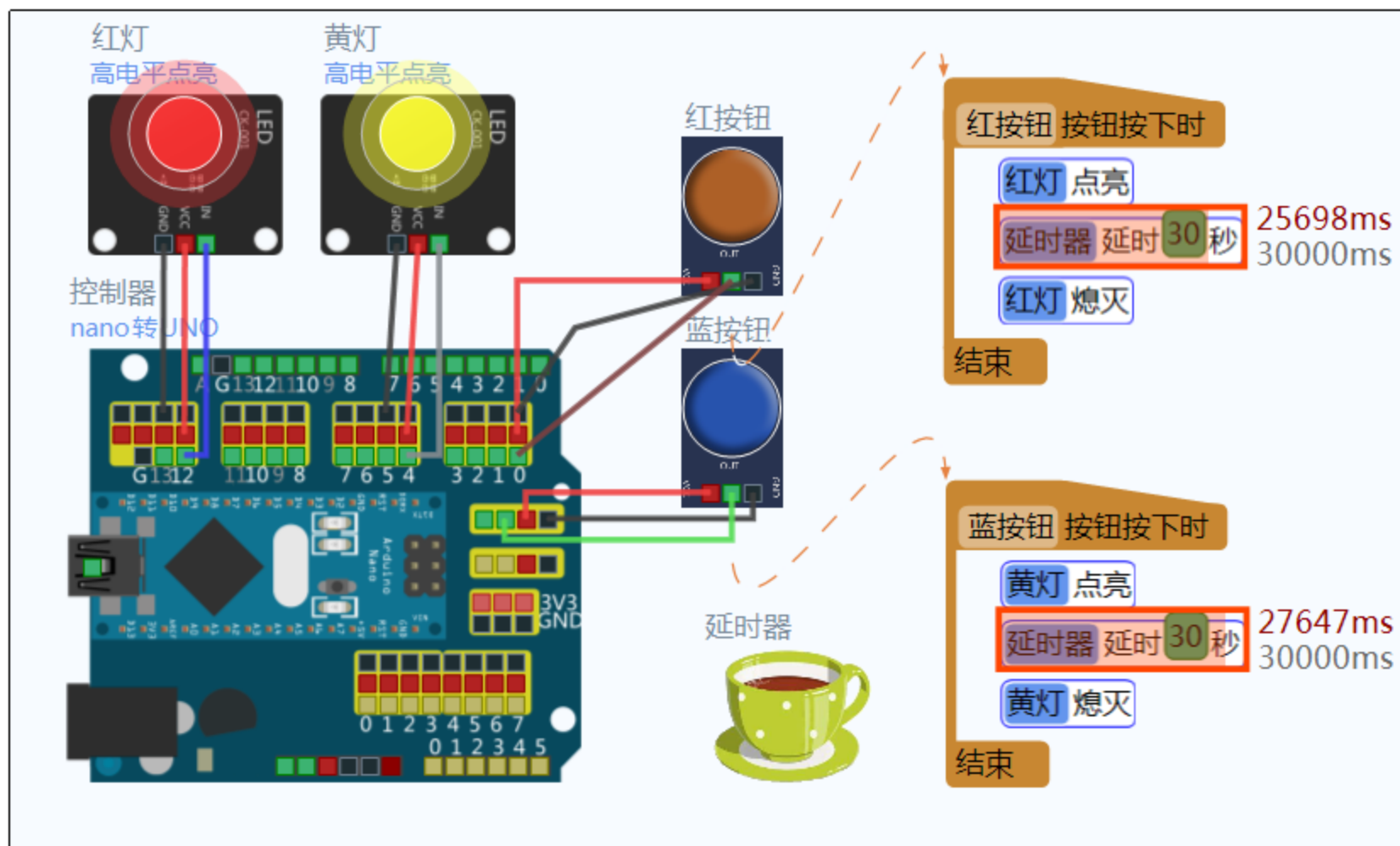
- 完整程序的仿真效果如图所示，两个按钮控制两个灯，可快速检测到按钮按下。而且互不影响，这就是多线程的应用，什么是多线程呢？



- 在短距离田径比赛中，多个运动员各自在自己的跑道上向终点奔跑。每个运动员都有专属自己的跑道，在奔向终点的过程中，他们互不影响。
- 程序也具有类似的功能，不同功能的程序相互独立执行，可同时进行，这称为多线程并行。



- 在按钮控制灯的程序中，就有两条线程：一、红按钮控制红灯；二、蓝按钮控制黄灯。我们可以发现两段程序是独立执行，互不影响的。



- 接下来我们再来体验一下多线程的应用，编写一段程序实现音乐播放过程中音符发生变化时，彩灯的颜色随机的改变。
- 这段程序有两个线程：一、循环播放音乐；二、彩灯的颜色随音符改变。
- 如果放在单线程程序中，是无法实现的。因为如果检测到音符发生改变，则要执行彩灯颜色改变指令，此时就不能播放音乐。单线程程序在任意时刻都只能执行一条指令。



- 首先我们来选择想要播放的歌曲。
- 在元素列表中找到音乐模块，如图1所示。
- 将其添加到编程区，点击图片，在弹出的对话框中点击“编辑”；
- 之后在弹出的窗口中，可以自己编写音乐，也可以选择“打开音乐库”，在库里挑选歌曲，最后点击确定。例如选择“天空之城”这首音乐，如图3所示

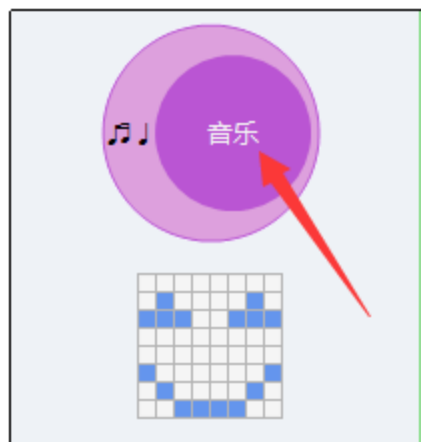


图1

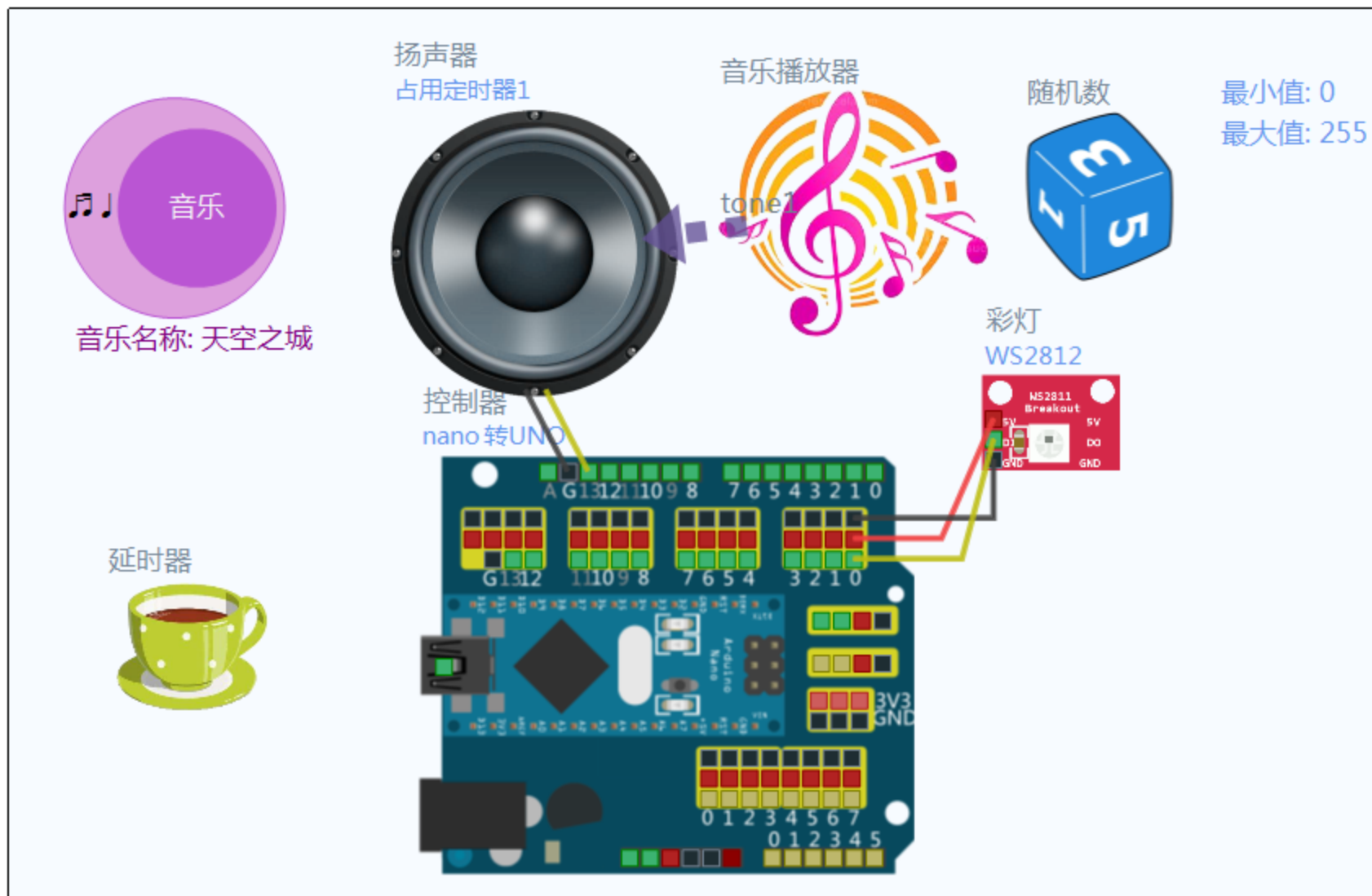


图2

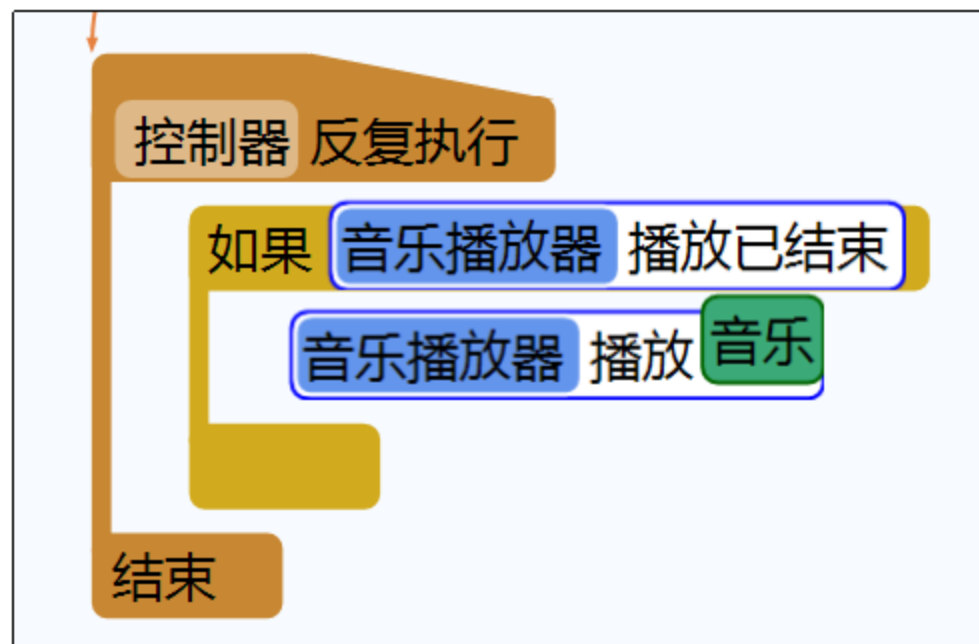


图3

- 编写音乐彩灯程序的所有模块如图所示：



- 音乐播放程序如图所示，为了避免音乐播放器不断重启，导致不能完整播放音乐，需要添加判断指令，只有当音乐播放结束的状态下才能重新开始播放音乐。



- 点击音乐播放器图片，在弹出的对话框中点击“通道1音符变化时”，如图1所示；
- 则会出现如图2所示的指令；
- 编写彩灯颜色变化程序，如图3所示；



图1



图2

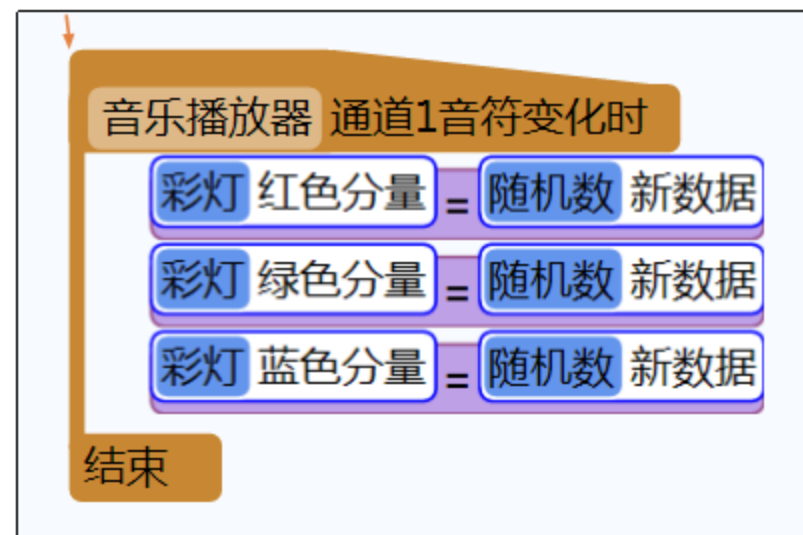


图3

- 完整程序如图所示，实现音乐播放和彩灯颜色改变同时执行。

