
MICRO:BIT

造物粒子入门教程



DFROBOT
DRIVE THE FUTURE

目录

目录.....	2
第一话：MakeCode 与 micro:bit	6
MakeCode 平台介绍	6
micro:bit 简介	6
初次使用 micro:bit	7
STEP1: 打开 makecode 在线编程平台	7
STEP2: 连接 micro:bit 至电脑	8
STEP3: 新建项目	9
STEP4: 下载程序至 micro:bit	10
Micro:bit 扩展板与 Micro:bit 的结合	12
第二话：小试牛刀.....	14
项目一 神秘的 micro:bit	14
麦克的故事	14
所需元件	14
实现功能	15
硬件连接	15
编写程序	15
能力提升	17
项目二 闪烁的 LED	18
麦克的故事	18
所需元件	18
任务一：点亮 LED 灯	19
实现功能	19
硬件连接	19
编写程序	20
实现效果	21
任务二：闪烁 LED 灯	21
实现功能	21
编写程序	21
能力提升	22
项目三 呼吸灯.....	23
麦克的故事	23
所需元件	23

任务一：按钮台灯	24
实现功能.....	24
硬件连接.....	24
编写程序.....	25
任务二：旋钮控制 LED 灯	27
实现功能.....	27
硬件连接.....	27
能力提升	28
项目四 变速风扇.....	29
麦克的故事	29
所需元件	29
任务一：按钮风扇.....	30
实现功能.....	30
硬件连接.....	30
编写程序.....	31
任务二：变速风扇.....	33
实现功能.....	33
硬件连接.....	33
编写程序.....	33
第三话：如虎添翼.....	35
项目一 电子蜡烛.....	35
麦克的故事	35
所需元件	35
任务一：声音强度探测.....	36
实现功能.....	36
硬件连接.....	37
编写程序.....	37
任务二：电子蜡烛.....	38
实现功能.....	38
编写程序.....	38
项目二 自动门.....	40
麦克的故事	40
所需元件	40
任务一：按钮控制舵机（伺服机构）	41
实现功能.....	41

硬件连接.....	42
编写程序.....	42
实验效果	43
任务二：运动传感器控制舵机（伺服机构）	44
实现功能.....	44
硬件连接.....	44
编写程序.....	44
实验效果	45
项目三 音乐盒	46
麦克的故事	46
所需元件	46
任务一：电子音乐盒	47
实现功能.....	47
硬件连接.....	47
编写程序.....	48
任务二：智能音乐盒	49
实现功能.....	49
编写程序.....	49
项目四 炫彩灯带	51
麦克的故事	51
所需元件	51
任务一：点亮彩虹灯带.....	52
实现功能.....	52
硬件连接.....	52
编写程序.....	52
实验效果.....	54
任务二：依次点亮彩虹灯带	55
实现功能.....	55
编写程序.....	55
任务三：声音控制彩虹灯带	57
实现功能.....	57
编写程序.....	57
实验效果.....	58
第四话：大展拳脚	59
项目一 自平衡仪	59

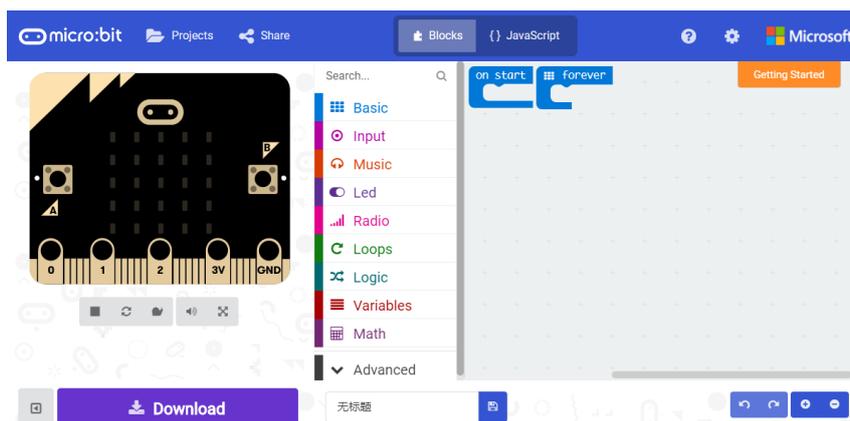
麦克的故事.....	59
所需元件	59
实现功能	60
硬件连接	60
编写程序	61
实验效果	63
能力提升	63
项目二 DJ 演奏台	64
麦克的故事.....	64
所需元件	64
实现功能	65
硬件连接	65
编写程序	66
实验效果	68
能力提升	69
项目三 可移动门铃	70
麦克的故事.....	70
所需元件	70
实现功能	71
硬件连接	72
编写程序	72
实验效果	75
能力提升	75
项目四 拆弹游戏	76
麦克的故事.....	76
所需元件	76
实现功能	77
硬件连接	77
编写程序	77
能力提升	83

*点击页面左上角 DF 创客社区，有意外惊喜哦！

第一话：MakeCode 与 micro:bit

MakeCode 平台介绍

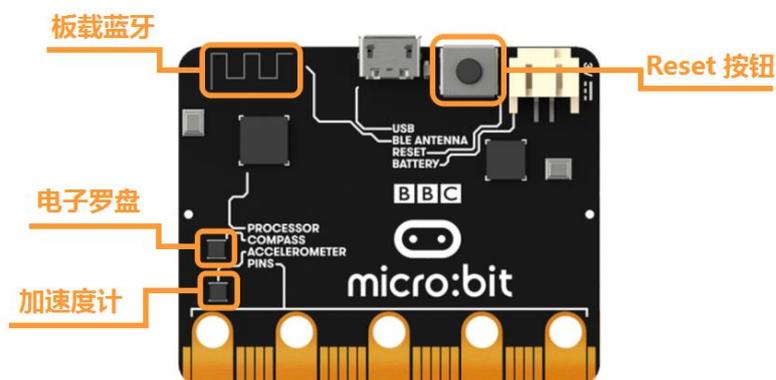
MakeCode for micro:bit 是 micro:bit 官网上使用最为广泛的图形化编程环境，它基于微软开源项目 MakeCode 开发的图形化编程环境。

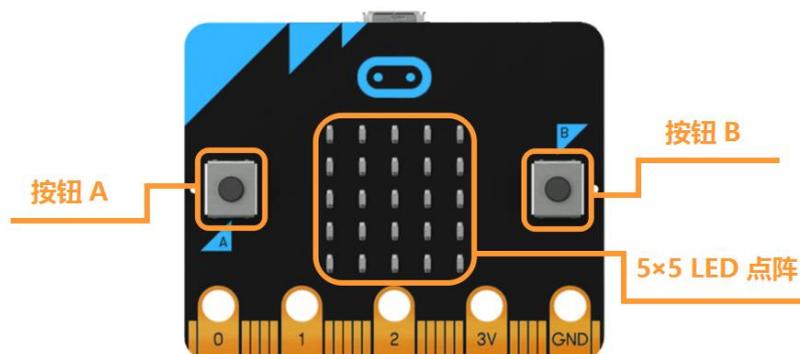


micro:bit 简介

micro:bit 是一款由英国 BBC 设计的 ARM 架构的单片机，板载蓝牙，加速度计，电子罗盘，三个按钮，5x5LED 点阵，主要用于青少年的编程教育。

利用 micro:bit 可以实现任何酷炫的小发明，无论是机器人还是乐器。micro:bit 拥有一系列新颖的功能，例如 25 个可显示消息的红色 LED 灯；有两个可编程按钮，可以用于控制游戏操作或者暂停/播放一首音乐。micro:bit 可以检测动作并且告知使用者动作进行的方向，同时它也可以通过低功耗蓝牙模块与其它设备或因特网互联。



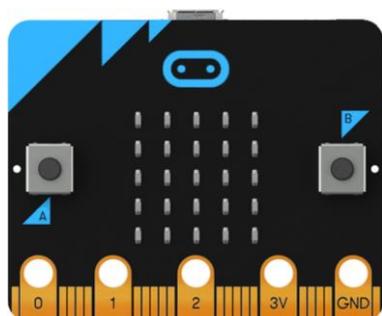


micro:bit 还自带光照度传感器和温度传感器等常见的感应装置，这意味着它自己本身也可以制作很多生活中常见的智能化产品。

初次使用 micro:bit

如果你是第一次接触 micro:bit，同样可以通过以下零基础教程学习如何使用 makecode 在线编程平台，以及如何为 micro:bit 编写程序。

开始之前，请确认你手边有如下物品：



Micro:bit 主板



microUSB 连接线

除此之外，你还需要一台运行 Windows/Mac OS/Linux 操作系统并且有网络连接的电脑。

STEP1: 打开 makecode 在线编程平台

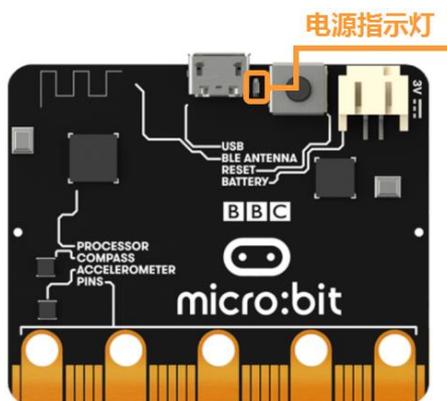
以下的步骤说明基于 Windows 操作系统，如果你使用的是其他操作系统，可以将其作为参考。

首先，你需要登录网页：<https://makecode.microbit.org/#>

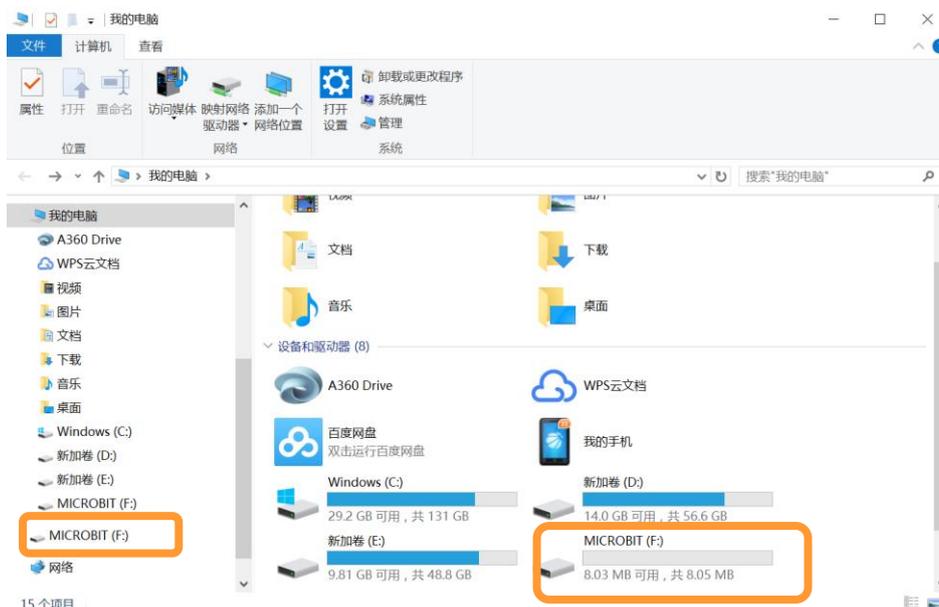


STEP2: 连接 micro:bit 至电脑

将 micro:bit 主板通过 MicroUSB 数据线连接至电脑。正确连接时 micro:bit 主板背面的电源指示灯会亮起。



在编程之前，我们需要确保主板被电脑识别。打开“我的电脑”。接上 micro:bit 主板的就会在磁盘列表中显示磁盘“MICROBIT”。



STEP3: 新建项目

现在让我们仔细看下 makecode 在线编程界面。



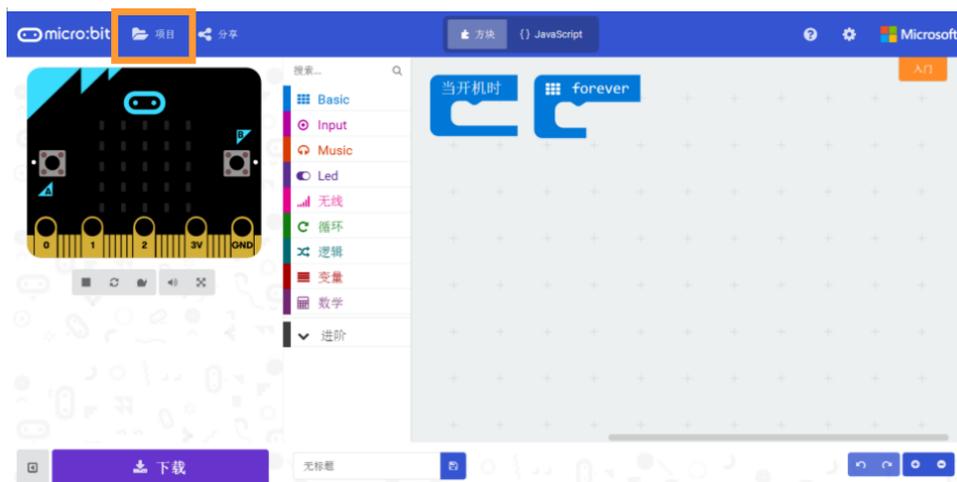
模拟窗口：模拟 micro:bit 工作的状态，在编程过程中，可以通过观察模拟窗口，观看初步的运行结果。

指令区：拖动指令区的指令可以对 micro:bit 进行编程控制。

脚本区：程序的编写区域，拖拽指令区的指令在此编写程序。

设置：常用功能为切换语言。

单击模拟窗口正上方的“项目”。

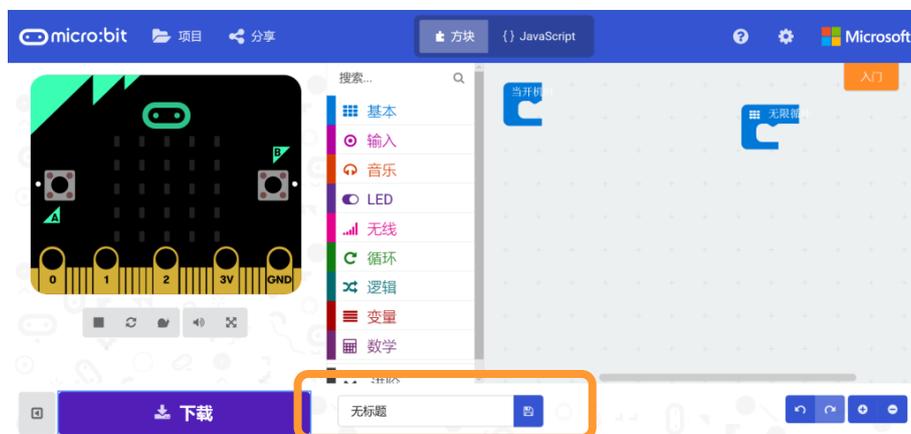


在跳转出来的页面单击“新建项目”即可。



STEP4: 下载程序至 micro:bit

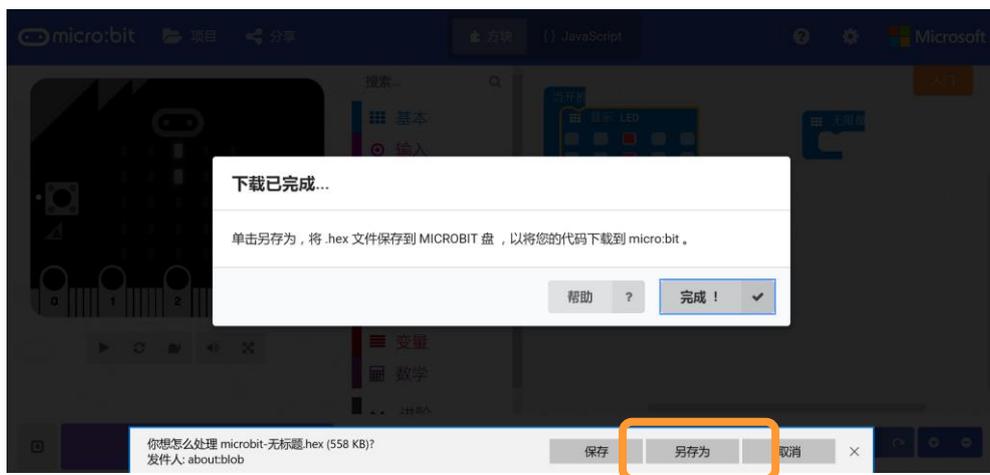
程序编写完成之后要下载到 micro:bit 主板里面，点击指令区正下方的“无标题”框，给程序命名。



在下载之前，你可以在模拟窗口看到程序运行之后的模拟动作，通过模拟窗口下方的按钮可以控制模拟器。

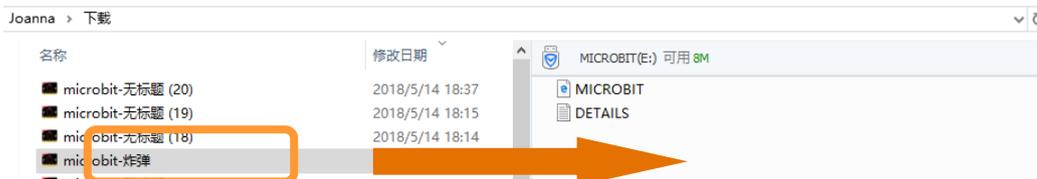


点击模拟窗口正下方的“下载”，弹出的对话框下选择“另存为”。



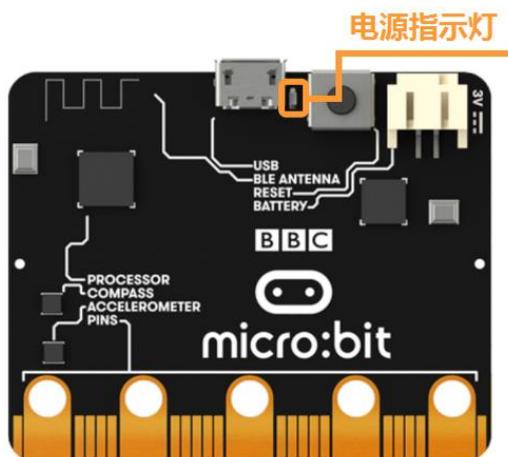
在弹出的对话框中, 选择把.hex 文件保存到“MICROBIT 磁盘”, 再点击“保存”。

如果没有出现另存为, 需要你找到下载的.hex 文件, 复制或者拖入到“MICROBIT 磁盘”。





下载过程中 micro:bit 主板背面的电源信号灯会闪烁，当下载完成后电源信号灯停止闪烁，保持长亮。



Micro:bit 扩展板与 Micro:bit 的结合

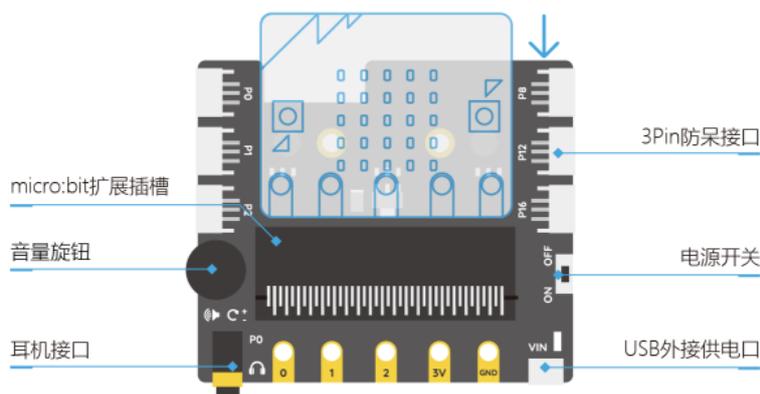
Micro:bit 的使用限制

Micro:bit 需要通过金手指插槽来连接更多外部设备。因其自身的驱动能力较差，无法直接驱动电机，舵机等大电流设备。

另外，micro:bit 本身的供电和信号电压为 3.3V，无法直接和常用的 5V 电子元件一起使用。

Micro:bit 扩展板的好处

1. micro:bit Boson 扩展板能轻松将 micro:bit 额外的引脚引出，即插即用，简单快捷；
2. 扩展接口，连接稳定性更强；
3. 外接的 USB 供电口 VIN 能为外部元件提供额外的供电，更好地支持灯带，舵机等大电流设备；
(程序下载时，还是需要连接在 Micro:bit 上)
4. 扩展板还附带了 3.5mm 耳机接口和音量旋钮，方便耳机直插。



扩展板最大额定电流为 500mA，请在使用前确定元件的最大功率。请勿将扩展板与大型舵机一起使用，可能引起板载元件损坏！

以上就是使用 micro:bit 的一些基本的方法。如果在使用中存在任何疑问或者建议，欢迎访问我们的论坛联系我们。

论坛链接: <http://www.dfrobot.com.cn/community/forum.php>

记得来逛 DFRobot 的社区看看更多的教程和精彩的项目哦。我们同样希望你能够把你自己的项目或者想法发在论坛上分享。欢迎成为我们的一员。

第二话：小试牛刀

经过前面的知识介绍，你是不是跃跃欲试了呢？现在，我们就开始 Boson for micro:bit 的神奇之旅吧！

项目一 神秘的 micro:bit

麦克的故事

麦克会在周末的时候会跟小朋友用网络沟通，常常在手机上或者电脑上面会见到一些萌萌的，很可爱的表情包，比如这样的：

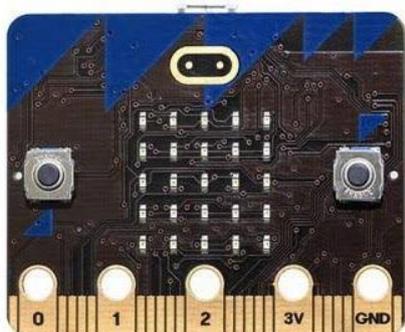


麦克问小米老师，可不可以自己设计表情包，随时随地给其他小朋友展示呢？

小米老师说：带你认识一位新的小朋友吧，它的名字叫 micro:bit。当看到这个小小的很精致的 micro:bit 的小板子的时候，麦克很好奇，这个小板子真的可以做出表情包么？麦克已经迫不及待的动手制作表情包啦，接下来，由小米老师带着大家一起玩起来。

所需元件

1 × micro:bit 主控板



1 × MicroUSB 连接线

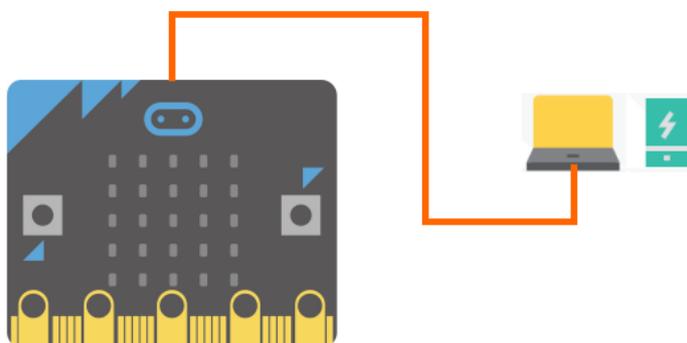


实现功能

micro:bit 板子上面有 5*5 (25 个) 板载的小灯，我们可以通过控制小灯的亮暗，制作小灯亮出不同的造型，展示表情包的效果。

硬件连接

将 micro:bit 主控板用 USB 连接线连接在电脑上。



编写程序

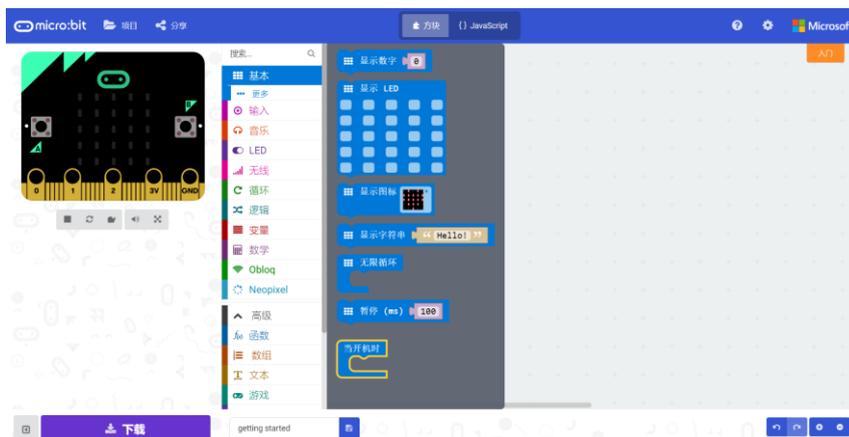
打开 MakeCode 在线编程界面。

STEP1: 新建项目，具体操作可参考第一话“初次使用 micro:bit”的 STEP3。

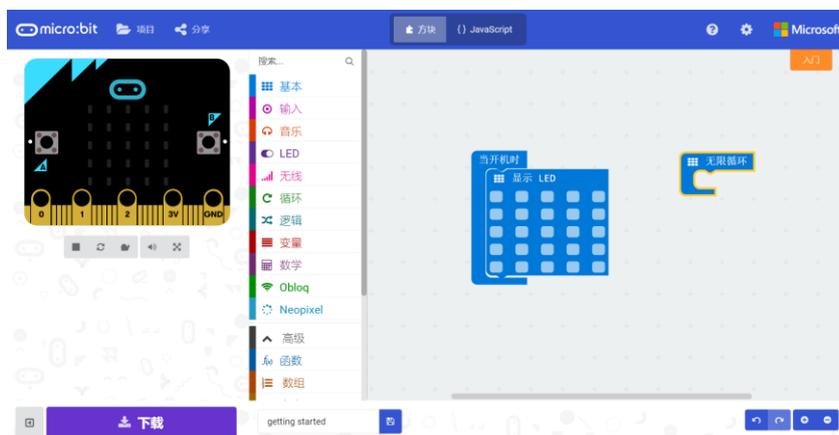
STEP2: 单击指令区最上面的“基本”，在指令区的右侧会出现一排模块，在右侧模块中找到“显示 LED”模块。



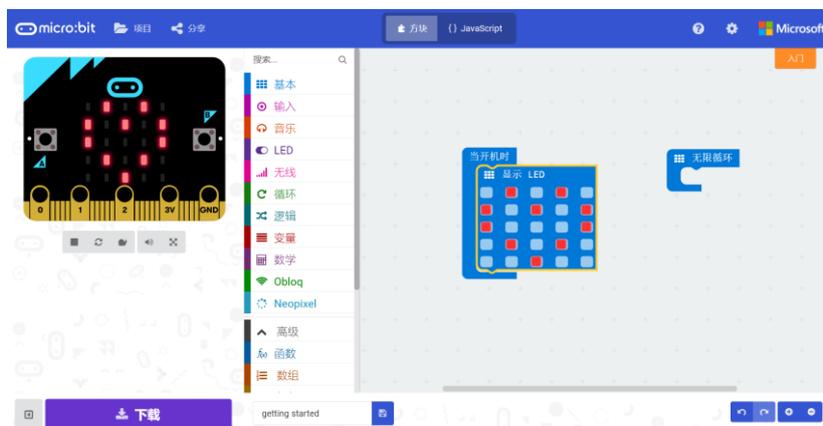
STEP3: 打开 MakeCode 在线网站之后，界面上会自动出现“当开机时”模块，或者在“基本”右侧模块的最底部可以找到“当开机时”模块，将“当开机时”模块拖至脚本区。



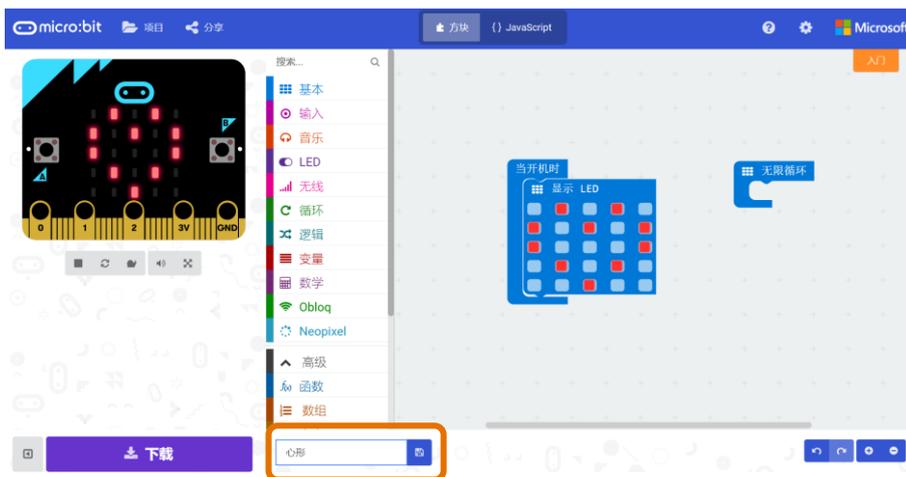
STEP4: 将“显示 LED”模块拖动至脚本区，并放入“当开机时”模块内。



在脚本区可以看到已经拖动的“显示 LED”模块，单击浅蓝色的小方块，可以开始设计表情包，首先，先来绘制一个萌萌的“心型”，可以在左边的模拟窗口可以看到绘制的模拟效果。



STEP5: 在下方红框内，编辑为“心形”（可以自己命名）。



STEP6: 单击“下载”，将文件保存在电脑上“micro:bit”硬盘上，即可完成保存程序，下载至 micro:bit 主板。这样，micro:bit 板子上就会显示“心形”。（具体下载步骤请参考第一话“初次使用 micro:bit”的 STEP4）



能力提升

试着制作可以切换两个表情的表情包。

项目二 闪烁的 LED

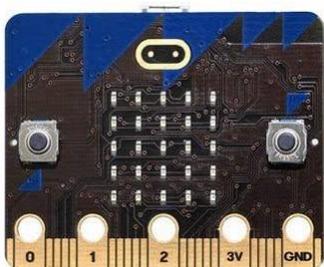
麦克的故事

一周的时间很快就过去啦。在周五放学的时候，麦克收到了新朋友 micro:bit 的邀请去家中做客，micro:bit 来自于遥远的 DF 星球。为了感谢好朋友的盛情邀约，麦克决定在出发前一天做一个小礼物送给 micro:bit，到底做什么好呢？麦克不由得纠结了起来...麦克看着大街上五颜六色的小彩灯，心中突然有了灵感。不如就做一个闪闪发光的星星灯吧。一到家麦克就拨通了小米老师的电话开始请教起来。

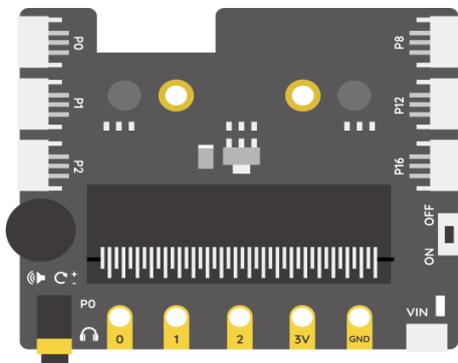


所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × LED 灯模块



1 x MicroUSB 连接线



任务一：点亮 LED 灯

实现功能

当在 Makecode 在线网站编写程序后，烧录至 micro:bit 小板子，点亮连接在 micro:bit 小板子上的外接 LED 灯。

硬件连接

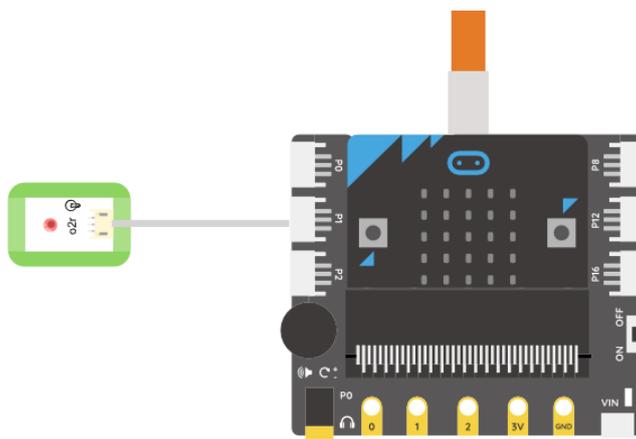
我们需要连接以下部分：

- micro:bit 主控板与 micro:bit 扩展板连接；



注意：接下来要学习的项目中都是将 micro:bit 主控板与扩展板结合在一起使用的，后续这一部分的连接将不会指出。

- LED 灯模块连接到扩展板的 P1 号接口



编写程序

打开 MakeCode 在线编程界面。

STEP1: 新建项目——点亮 LED 灯，具体操作可参考第一话“初次使用 micro:bit”的 STEP3。

STEP2: 硬件连接中运用到了外接 LED 小灯，编程过程中，通过设置 micro:bit 的引脚值（连接 LED 小灯的引脚）来控制小灯的亮暗。控制 LED 小灯呈现“亮”或“暗”两种状态。

LED 小灯在这个实验中属于**数字输出**，单击“高级”，会出现“引脚”，调用指令的流程为：引脚（高级指令）---- “向 引脚 P0 数字写入值 0”。（拖动后，将 P0 改为 P1）



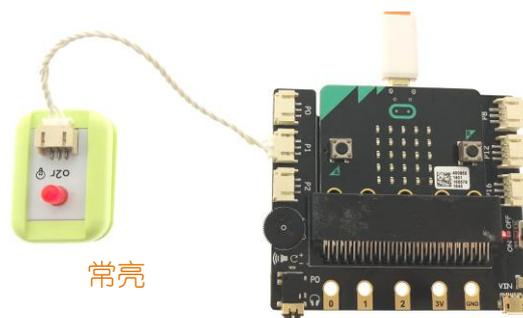
STEP3: 将其指令拖动入“无限循环”中，当烧录代码至 micro:bit 小板子时，可以点亮连接在 P1 号引脚的外接 LED 小灯。（拖动后，将 P0 改为 P1）引脚值只有 0 和 1 两种情况，当值为 1 时，表示小灯被点亮；当值为 0 时，表示小灯呈现熄灭状态。



最终程序，下载到 micro:bit 板子中。



实现效果



任务二：闪烁 LED 灯

实现功能

当在 Makecode 在线网站编写程序后，烧录至 micro:bit 小板子，外接 LED 小灯呈现亮一秒，暗一秒的闪烁效果。

编写程序

STEP1: 新建项目——闪烁 LED 灯，具体操作可参考第一话“初次使用 micro:bit”的 STEP3。

STEP2: 设置 micro:bit 的引脚值来控制小灯的亮暗，控制 LED 小灯呈现“亮”或“暗”两种状态。



STEP3: 怎么实现闪烁呢，需要让外接 LED 小灯保持亮的状态 1 秒，保持熄灭的状态 1 秒，呈现亮 1 秒，暗 1 秒相交替的状态，需要调用指令的流程为：基本-----暂停 (ms) (100)。

“暂停”指令可以通过改变后面的参数，控制 LED 小灯呈现亮或灭的状态保持一定时间。



STEP4: 将“暂停”指令拖入“无限循环”中，调整数字后，最终代码见下图。表示：P1 引脚为高电平（点亮状态）保持 1000ms，P1 引脚为低电平（熄灭状态）保持 1000ms，循环呈现闪烁效果。



能力提升

通过查找资料，了解一下 SOS 紧急救援信号灯的原理，与同学分享，尝试制作 SOS 信号灯吧！

项目三 呼吸灯

麦克的故事

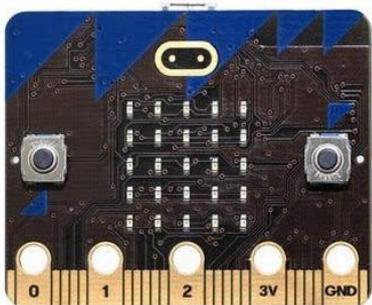
周六早上到了约定好的出发时间，麦克跟着 micro:bit 登上了前往 DF 星球的星际飞船，通过空间跃迁预计需要几个小时才能到达。

过了大约一个小时候左右，突然飞船内响起了呜呜呜的警报声，原来是空间跃迁仪出现了故障，飞船偏离了航向，在茫茫星海中迷了路。麦克急得满头大汗，倒是 micro:bit 还算镇定，对麦克说，这种时候千万不要慌，我们先想办法让别的飞船注意到我们，这样就容易获救啦。

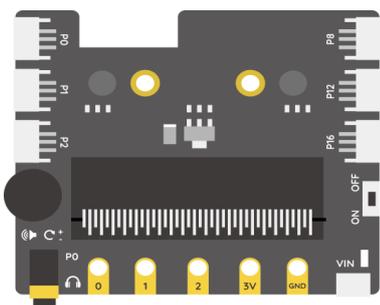
听了 micro:bit 的话，麦克也慢慢镇定了下来，他想起有时候晚上会看到天上的飞机有一亮一暗的灯光，在夜空中十分醒目，于是对 micro:bit 说：“让我们来做一个呼吸灯吧。”

所需元件

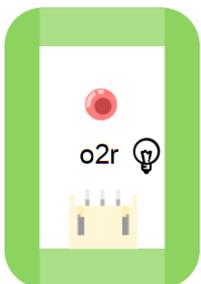
1 × micro:bit 主控板



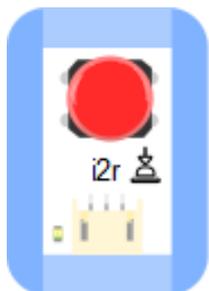
1 × micro:bit 扩展板



1 × LED 灯模块



1 × 按钮模块



按钮就像我们家用按钮台灯的按钮一样，当按下的时候，可以传递一个命令，让我们的小灯被点亮，不过需要连接编写好程序的 micro:bit 小板子才可以哦。

1 × 旋钮模块



旋钮作为**输入模块**，可以通过旋转不同的角度，传递不同的信号值。旋钮与按钮的区别：按钮处于按下状态时，电路接通，旋钮旋转在一定范围内，电路处于接通状态。

1 × MicroUSB 连接线



任务一：按钮台灯

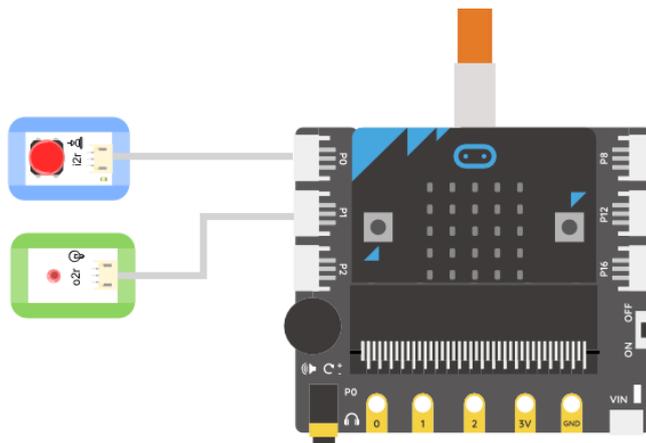
实现功能

当按下按钮的时候，外接 LED 小灯被点亮，松开按钮以后，外接 LED 小灯熄灭。

硬件连接

把按钮模块（旋钮模块）接到扩展板 P0 号接口上；

把 LED 灯模块接到扩展板 P1 号接口上。



编写程序

STEP1: 通过检测 P0 号引脚的值判断按钮是否被按下。

当按钮被按下时，设置 LED 小灯引脚为高电平，LED 灯被点亮，当松开按钮时，设置 LED 小灯引脚为低电平，LED 灯熄灭。

按钮模块属于数字输入，单击“高级”，会出现“引脚”，调用引脚时，通常会运用到这个指令，调用指令的流程为：引脚（高级指令）---数字读取引脚 P0。



STEP2: 逻辑指令中的“=”可以判断按钮是否被按下，如果按钮值为“1”成立，说明按钮被按下，否则，按钮没有被按下。

调用指令的流程为：逻辑----“=”。



将“数字读取引脚”嵌套在等式中



STEP3: 条件判断——怎样知道按钮是否按下？

如果按钮被按下，外接 LED 小灯被点亮；否则，外接 LED 小灯熄灭。

调用指令的流程为：逻辑——如果为 (true) 一则。



STEP4: 需要用到的指令结合在一起，实现：当按下按钮的时候，外接 LED 小灯被点亮，松开按钮以后，外接 LED 小灯熄灭。



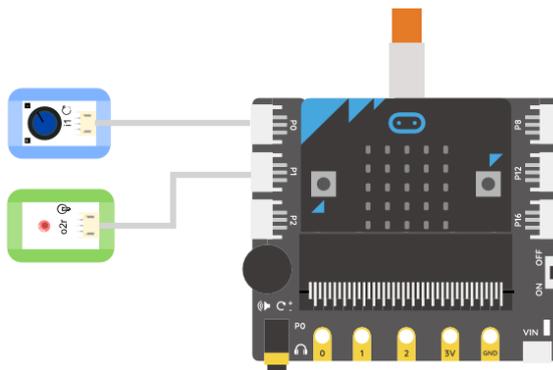
任务二：旋钮控制 LED 灯

实现功能

当旋钮旋至较大值时，micro:bit 控制外接小灯越亮；当旋钮旋至较小值时，micro:bit 控制外接小灯越暗。旋钮的旋转值不同，对应的小灯亮度不同，旋转时，灯的亮度连续变化。

硬件连接

把任务一按钮台灯中使用的按钮模块换为旋钮模块。



STEP1: 通过读取旋钮的值，来决定 LED 灯的亮度。

旋钮的值为模拟值，取值范围在 0-1023 之间，将其值映射为 LED 灯对应的亮度值。

旋钮连接板子的 P0 号引脚，旋钮值调用指令的流程为：引脚（高级指令）----模拟读取引脚（P0）。



STEP2: LED 灯的亮度由 P0 对应的旋钮值决定，将 P0 的值赋值给 P1，可以控制 LED 灯的改变，

调用指令的流程为：引脚（高级指令）----“向引脚 P0 模拟写入 1023”，将 P0 改为 P1。

之后将获取的旋钮值（P0）放入 P1 值中。



将两个指令嵌套在一起



【小贴士】

这段代码可以这样理解：获取旋钮当前旋转位置的值，这个值对应的是 LED 小灯的亮度值。

STEP4: 最终程序。



能力提升

利用家里的一些纸盒、纸杯、彩纸等，制作出精致的小灯外形，让它成为你自己的独一无二的作品。

项目四 变速风扇

麦克的故事

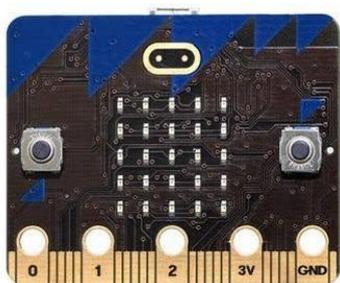
星际飞船在宇宙中随着惯性飘荡着，看着舷窗外明暗交替的灯光。

船舱内的温度也慢慢升高了起来，麦克的额头上都渗出了汗珠。麦克擦了一把汗说：“我们来做个小风扇凉快凉快吧。” micro:bit 立刻表示赞成。

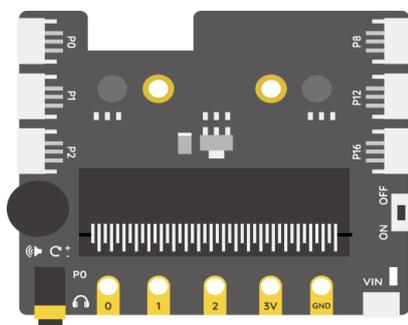
但是两人却有点无从下手，突然麦克灵机一动，想到了小米老师上节课所教给同学们的内容，呼吸灯的明暗变化不就是一个风扇的变速原理吗。于是，麦克和 micro:bit 开始了小风扇的制作。

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 风扇模块



与家用风扇的原理相似，内部有马达，扇叶可以旋转，转速可以调整。

1 × 按钮模块



1 × 旋钮模块



1 × MicroUSB 连接线



任务一：按钮风扇

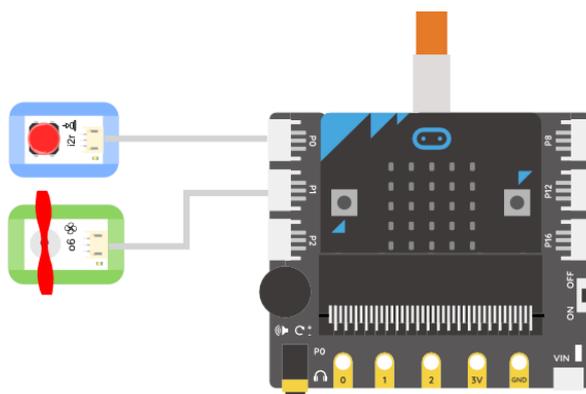
实现功能

当按下按钮的时候，外接风扇转动，松开按钮以后，风扇停止转动。

硬件连接

把按钮模块（旋钮模块）接到扩展板 P0 号接口上；

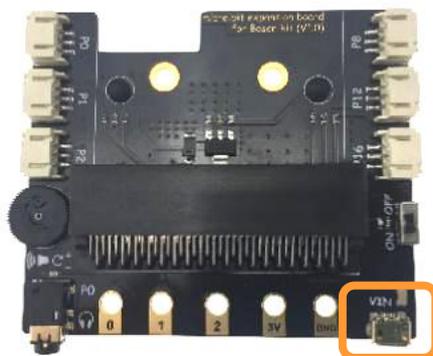
把风扇模块接到扩展板 P1 号接口上。



风扇模块的使用

硬件连接时，我们将 micro:bit 与电脑相连接，进行程序调试和下载；

实验中，由于风扇或舵机等传感器的耗电量较大，所以，需要外接电源（或电脑 USB □）独立供电。外接电源接在扩展板的外接电源接口 VIN。



编写程序

STEP1: 通过检测 P0 号引脚的值判断按钮是否被按下。

当按钮被按下时，设置风扇连接引脚为高电平，风扇转动，当松开按钮时，设置风扇连接引脚为低电平，风扇停止转动。

按钮模块属于数字输入，单击“高级”，会出现“引脚”，调用引脚时，通常会运用到这个指令，调用指令的流程为：引脚（高级指令）----数字读取引脚 P0。



STEP2: 逻辑指令中的“=”可以判断按钮是否被按下，如果按钮值为“1”成立，说明按钮被按下，否则，按钮没有被按下。调用指令的流程为：逻辑----“=”。



将“数字读取引脚”嵌套在等式中



STEP3: 如果按钮被按下，风扇转动；否则，风扇停止转动。调用指令的流程为：逻辑----如果为 (true) ----则----否则。



STEP4: 需要用到的指令结合在一起，实现：当按下按钮的时候，外接风扇转动，松开按钮以后，风扇停止转动。



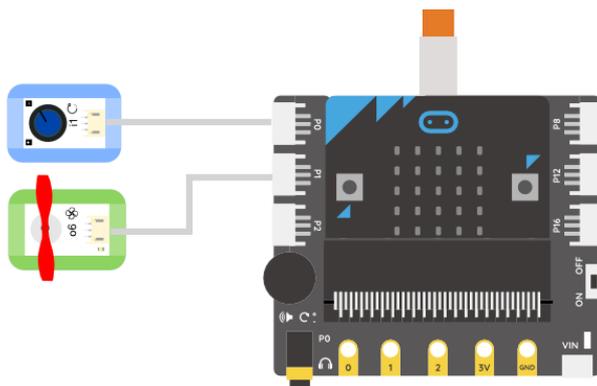
任务二：变速风扇

实现功能

当旋钮旋至较大值时，micro:bit 控制风扇转速增大；当旋钮旋至较小值时，micro:bit 控制风扇转速变小。旋钮的旋转值不同，对应的风扇转速不同。

硬件连接

将任务一按钮风扇用到的按钮模块换成旋钮模块。



编写程序

STEP1: 通过读取旋钮的值，来决定风扇的转速。

旋钮的值为模拟值，取值范围在 0-1023 之间，将其值映射为风扇对应的转速。

旋钮连接板子的 P0 号引脚，旋钮值调用指令的流程为：引脚（高级指令）----模拟读取引脚（P0）。



STEP2: 风扇的转速由 P0 对应的旋钮值决定，将 P0 的值赋值给 P1，可以控制风扇转速的改变。

调用指令的流程为：引脚（高级指令）----“向 引脚 P0 模拟写入（1023）”，将 P0 改为 P1。之后将获取的旋钮值（P0）放入 P1 值中。



将两个指令相互嵌套



STEP3: 最终程序。



第三话：如虎添翼

在学习了按钮模块、旋钮模块控制输出端之后，你是不是觉得好像还少了点什么呢？我们常说的传感器还有很多，在这一章节的内容学习中，我们将认识更多的传感器使用，也将制作出更加智能化、好玩和实用的作品。

项目一 电子蜡烛

麦克的故事

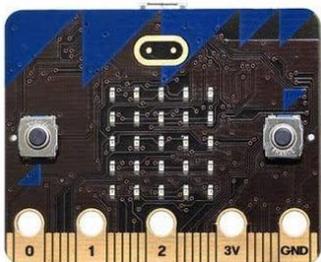
经历了一番太空历险之后，麦克和他的小伙伴终于有惊无险地回到了地球。

有一天，麦克看见新闻上关于温室效应的报道：人类的任何活动都可能造成碳排放，比如普通百姓简单的烧火做饭都能造成碳排放，任何物体被火烧后的废气都会产生碳排放。麦克深受启发要从小事做起减少碳排放，保护环境。

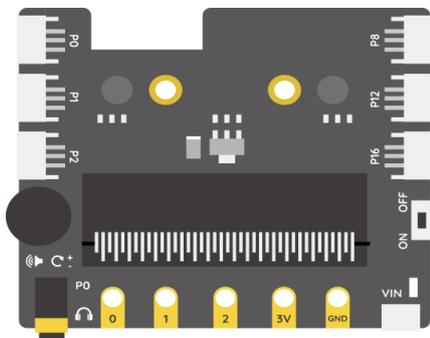
转眼间到了麦克生日这天，麦克突发奇想：能不能来做一个电子蜡烛，既特别还能保护环境呢？想着，麦克就开始倒腾起来：做一个能用声音吹灭的电子蜡烛。

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × LED 灯模块

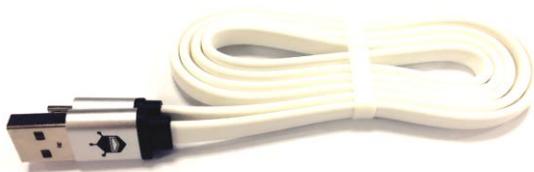


1 × 声音传感器模块



能够通过它来感知外界声音的强弱，编写代码实现声音强弱控制输出模块 LED 小灯的亮灭。

1 × MicroUSB 连接线



任务一：声音强度探测

实现功能

当声音强度模拟值小于 50 的时候，板载小灯显示数字“1”，表示声音较小，强度为 1 级；否则，声音强度模拟值大于 50 的时候，板载小灯显示数字“2”，表示声音较大，强度为 2 级。



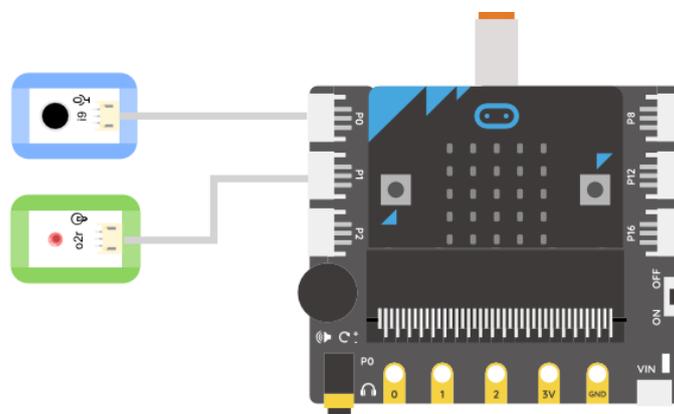
【小知识】

想一想，是不是有点像我们之前做过的按钮风扇呢？在这里我们将再次用到如果...则...(否则) ...的条件判断语句。

硬件连接

把声音传感器模块接到扩展板 P0 号接口上；

把 LED 灯模块接到扩展板 P1 号接口上。



编写程序

要想制作电子蜡烛，首先要认识声音传感器。我们可以将声音传感器与 micro:bit 板连接，通过显示数字来观察声音强弱程度。

STEP1: 将逻辑中的条件语句代码模块和逻辑判断数值大小的代码模块指令结合拖动入“无限循环”中，条件语句代码模块的功能在于设定触发显示板载数字的条件，如果满足右边逻辑判断语句。则执行在“则”右侧的代码语句，反之，如果不满足逻辑判断语句，则执行的是在“否则”右侧的代码。



STEP2: 模拟读取引脚读取 P0 引脚连接的声音传感器的模拟信号数值，模拟数值范围在 0—1023 之间。



STEP3: 设定“声音强度探测”的实现程序。

逻辑指令中的“<”可以判断声音强度的模拟值是否达到设定的标准。

如下图，表示模拟读取引脚 P0 的数值“<50”成立的时候，声音强度为 1 级的时候，板载小灯显示数字“1”，否则，模拟读取引脚 P0 的数值“>50”时，声音强度为 2 级，板载小灯显示数字“2”。



对着声音传感器说话或者吹气，看你的板载 LED 点阵会不会显示相应的数字。

如果你已经很用力说话或吹气了却还是不能显示相应的数字，该怎么调节临界值 50 这个强度呢？改为更大还是更小？

任务二：电子蜡烛

实现功能

当在 Makecode 在线网站编写程序后，烧录至 Micro: bit 小板子，通过声音传感器控制 LED 小灯的亮灭，就像过生日的时候吹蜡烛一样，对着声音传感器吹气，当声音达到一定强度的时候，比如之前设定的 2 级强度时，小灯就熄灭。

编写程序

STEP1: 编程过程中，通过设置 Micro: bit 的 P0 引脚值（连接声音传感器的引脚）来控制小灯的亮暗。

当声音达到设定的强度的时候，小灯熄灭；反之，当声音未达到设定强度时候，小灯保持亮起的状态。

将如图所示的判断语句代码模块指令拖动入“无限循环”中。



STEP2: 设定“电子蜡烛”的实现程序。

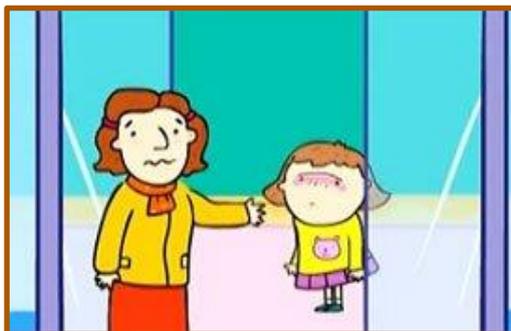
当声音传感器模拟值小于设定值 50 的时候，小灯保持亮起的状态，反之，轻轻一吹，声音强度大于 50 时，小灯将保持 3 秒的熄灭状态。



项目二 自动门

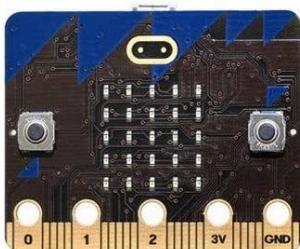
麦克的故事

科技发展的很快，麦克好奇的问小米老师，每次经过图书馆的时候，门会自动的打开，这是为什么的？是门口会有人专门守着，会帮每个人开门么？小米老师说：没有人专门守着，门很神奇的，它会监测到有人经过，自动的把门打开。

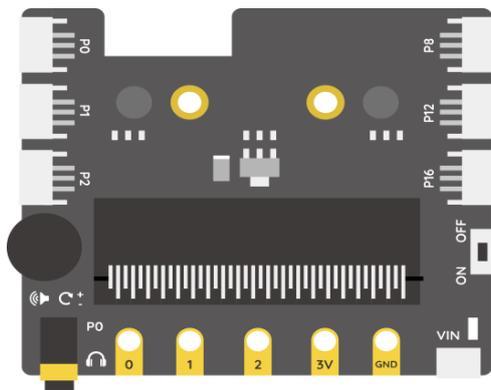


所需元件

1 x micro:bit 主控板



1 x micro:bit 扩展板



1 x 按钮模块

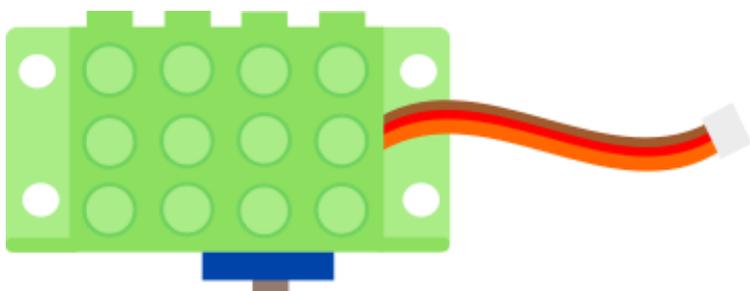


1 × 运动传感器模块



又称红外热释电运动传感器，能检测运动的人或动物身上发出的红外线，一定范围内，有人或动物，可以传递信号，就像按钮被按下传递信号一样。

1 × 舵机（伺服机构）



1 × MicroUSB 连接线



任务一：按钮控制舵机（伺服机构）

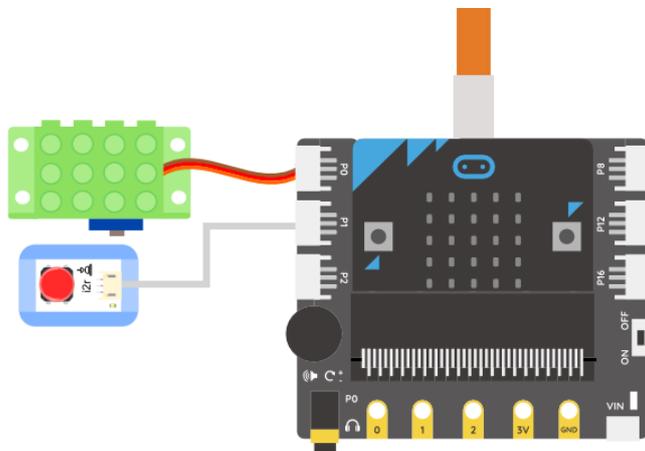
实现功能

当按下按钮的时候，舵机（伺服机构）转动 100° 并且 micro:bit 显示“O”，按钮没有被按下时，舵机旋转 0° 并且 micro:bit 显示“X”。

硬件连接

把舵机接到扩展板 P0 号接口上；

把按钮模块（运动传感器）接到扩展板 P1 号接口上。



编写程序

STEP1: 通过调用 P0 号引脚可以读取舵机（伺服机构）的值。

调用指令的流程为：引脚（高级指令）---->向伺服机构 引脚 P0 写入(180)。将“180”改成“100”。
(舵机连接 P0 引脚，故不用修改引脚值)



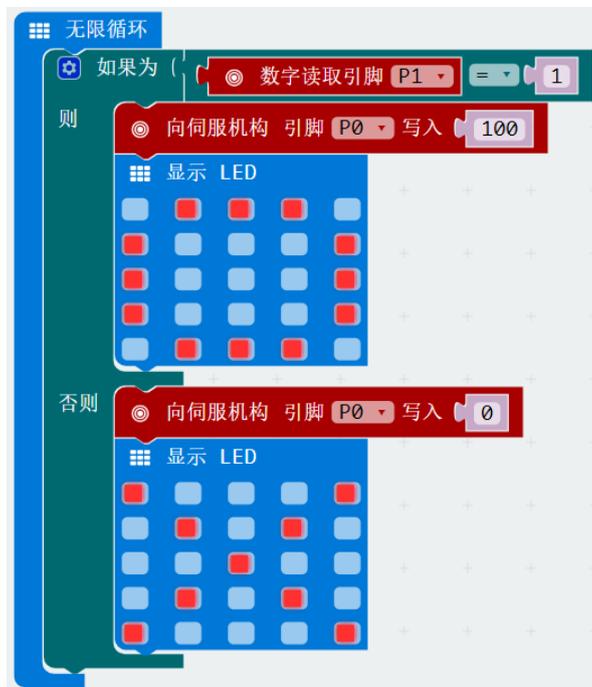
STEP2: 如果按钮按下, 实现舵机转动 100° 并且 micro:bit 显示“O”; 否则舵机旋转 0° 并且 micro:bit 显示“X”。

需要用到的指令为：如果为 (true) --则一否则。



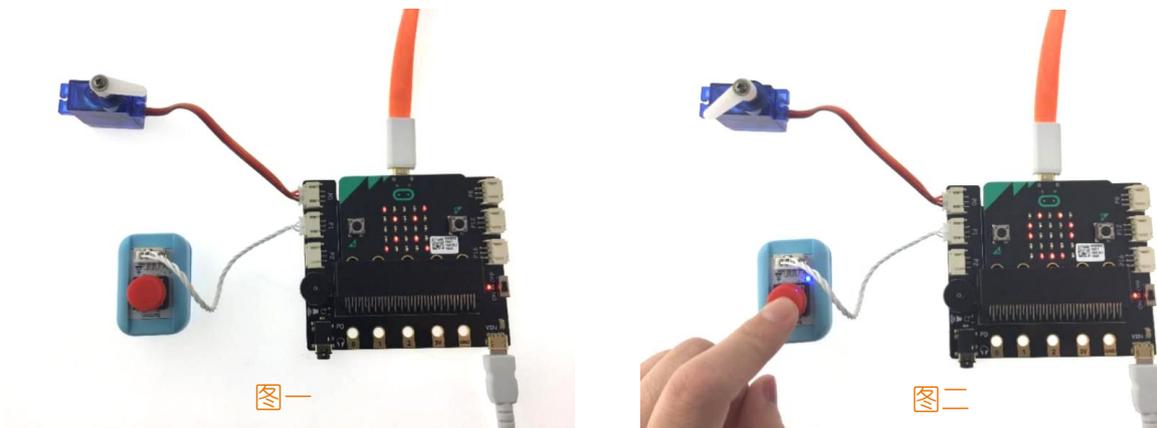
STEP3: 最终程序。

注意: 由于舵机 (伺服机构) 耗电量大, 需要外接电源。



实验效果

图一: 按钮没有按下; 图二: 按钮被按下



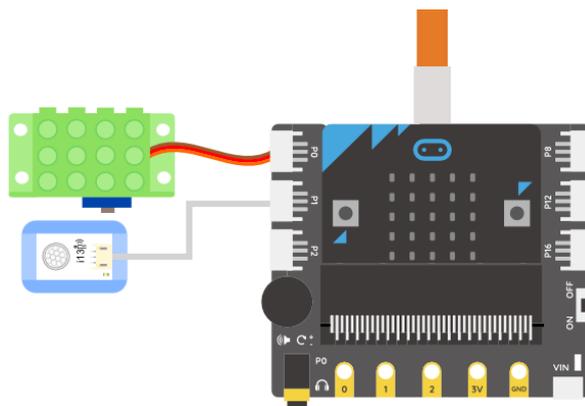
任务二：运动传感器控制舵机（伺服机构）

实现功能

当运动传感器检测到有人经过时，舵机转动 100° 并且 Micro.bit 显示 “O”，否则，舵机旋转 0° 并且 micro:bit 显示 “X”。

硬件连接

将任务一按钮控制舵机中的硬件按钮模块换成运动传感器模块。



编写程序

STEP1：通过调用 P1 号引脚可以读取运动传感器的值。

调用指令的流程为：引脚（高级指令）---数字读取引脚 p0 值（0）。（拖动后，将 P0 改为 P1）



STEP2：当运动传感器检测到有人经过时，实现舵机转动 100° 并且 micro:bit 显示 “O”；否则，舵机旋转 0° 并且 micro:bit 显示 “X”。

需要用到的指令为：如果为（true）--则--否则。如下图：



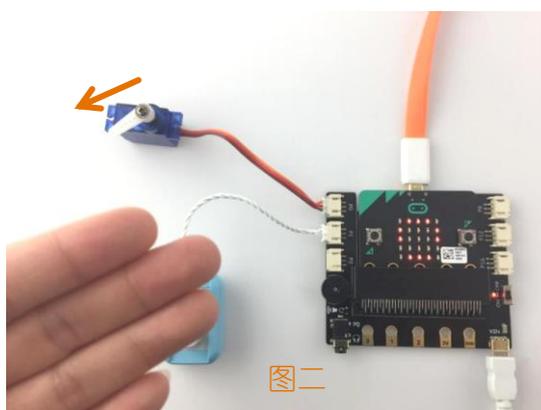
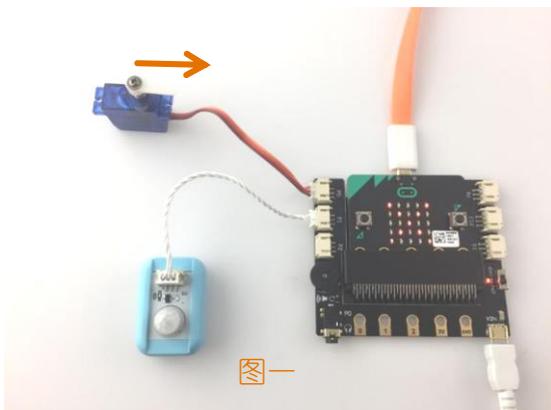
STEP3: 最终程序。



实验效果

图一：无人时；图二：有人时

注意：由于舵机（伺服机构）耗电量大，需要外接电源。



项目三 音乐盒

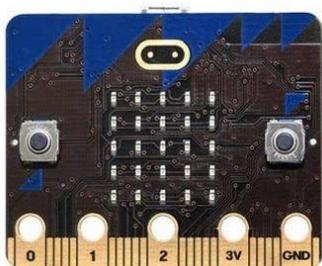
麦克的故事

“一闪一闪亮晶晶，满天都是小星星……” 麦克在学校最喜欢的就是音乐课啦！这天刚好上完音乐课学到了小星星这首歌，回到家中，麦克突然想起之前学习 micro:bit 的时候听小米老师讲通过编程也能让 micro:bit 板演奏出动听的音乐，这是真的吗？麦克开始认真研究起来，让 micro:bit 板也学会唱小星星。

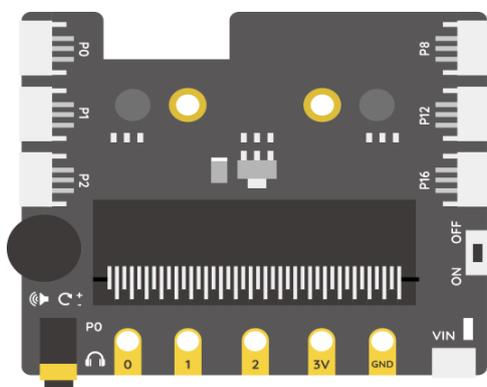


所需元件

1 × micro:bit 主控板



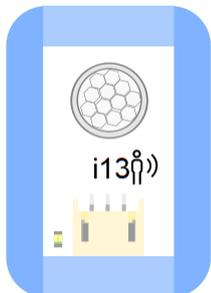
1 × micro:bit 扩展板



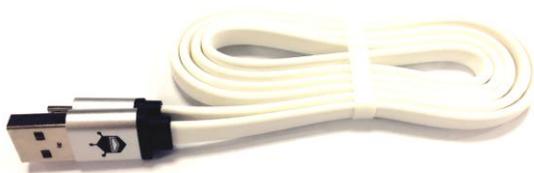
1 × 普通耳机



1 × 运动传感器模块



1 × MicroUSB 连接线



任务一：电子音乐盒

实现功能

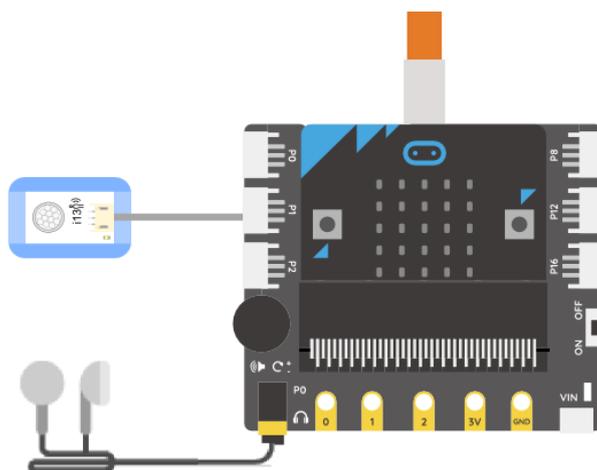
让 micro:bit 主板播放音乐。

硬件连接

用数据线将 micro:bit 主板与电脑连接起来，将耳机插入耳机插孔；

将运动传感器模块接在扩展板的 P1 号接口。

注意：播放音频的耳机插孔数据通信占用了 P0 端口，因此播放声音的时候，P0 端口不可用。



编写程序

STEP1: 将 music 模块集合中的播放旋律模块，拖动到无限循环中。

播放旋律模块中已经有内置的很多旋律可供选择，我们在这里选择的是第一段旋律“dadadum”重复播放一次。

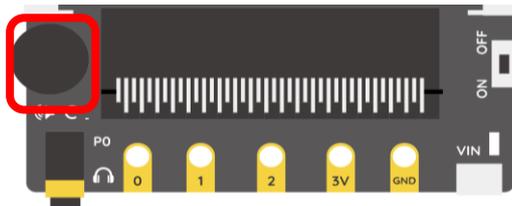


STEP2: 将选择的旋律“dadadum”一段完整的占 4 个节拍，所以在播放旋律模块的下方需要设定旋律播放的时长为 4 个节拍。

最终程序。



STEP3: 在扩展板耳机插孔的上方紧挨着的“小转盘”，试着旋转它，可以调节 micro:bit 板播放声音的大小。



任务二：智能音乐盒

实现功能

当有人靠近时，让 micro:bit 主板播放自己编的一段音乐；当没有人靠近时，不播放音乐。

编写程序

STEP1：编写一段歌曲《小星星》的音乐。

先尝试编写能演奏小星星第一句的代码，先看看这首歌的简谱是怎样的，音调不同，音符不同，需要的发声频率也不同。



以中音 C 调为例，发声频率与其对应的音符如下表所示：

音符	1	2	3	4	5	6	7
字母	C	D	E	F	G	A	B
频率	262	294	330	350	393	441	495

STEP2：打开 music 代码集中的代码模块，现在用到的是播放音调模块。在这里可以设置发出的音调和所占节拍。



STEP3: 编写好小星星的曲子之后我们的 micro:bit 已经学会唱歌啦，但是如何让它变得智能呢？这时候，我们就需要用到运动传感器了。其实很简单，只需要添加一个我们之前已经学过的条件判断语句就可以让智能音乐盒得以实现啦！一起来看看代码是怎样的吧？



项目四 炫彩灯带

麦克的故事

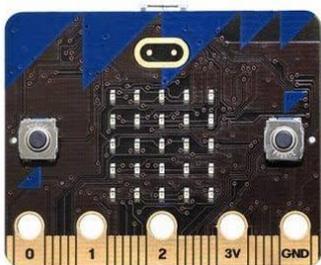
上节课结束之后麦克做出了音乐盒，能发出美妙的音乐又怎么能没有酷炫的外形呢，比如说这样的还有这样的：



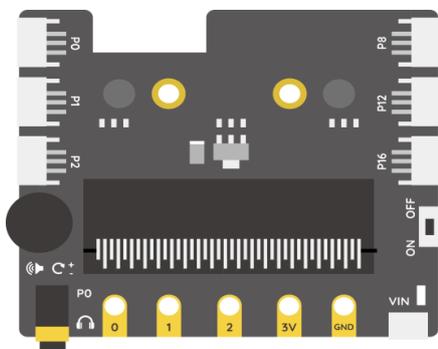
今天，麦克要给音乐盒增添一些色彩了，让 micro:bit 音乐盒变得好看起来。快来和小米老师一起动手吧。

所需元件

1 × micro:bit 主控板



1 × micro:bit 扩展板

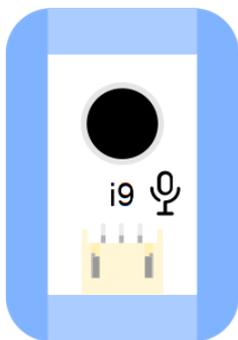


1 × 彩虹灯带



彩虹灯带可以通过调节色彩调出不同颜色的灯效

1 × 声音传感器模块



1 × MicroUSB 连接线



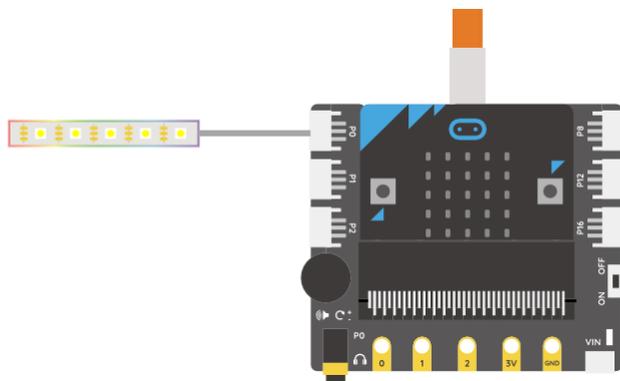
任务一：点亮彩虹灯带

实现功能

通过编写程序，将程序烧录至板子后，可以控制将彩虹灯带点亮并呈现七彩的效果。

硬件连接

将彩虹灯带接在扩展板的 P0 号接口。

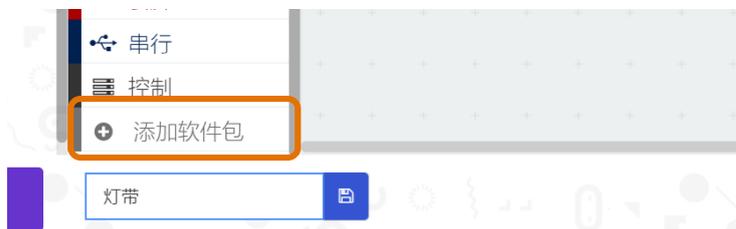


编写程序

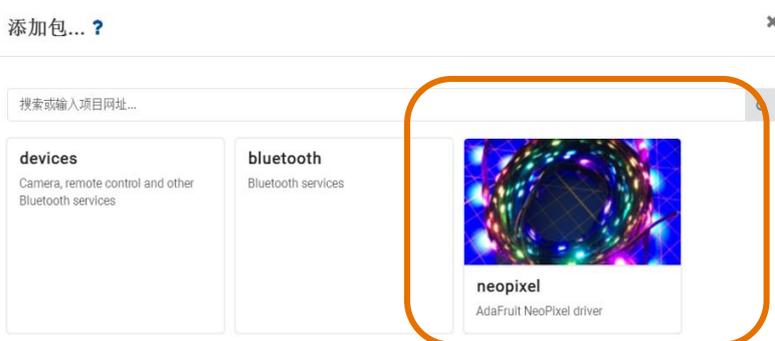
STEP1：扩展灯带的指令库。

使用灯带需要调用 Neopixel 指令，Neopixel 指令的调用需要滚动至指令区的最低部，单击“添加软件包”，出现“添加包”的菜单栏。

调用指令的流程为：高级----添加软件包----neopixel。单击 neopixel 会增加 neopixel 模块。



添加软件包 neopixel



STEP2: 灯带常用指令

在使用彩虹灯带的过程中，首先选定灯带连接 micro:bit 的引脚、灯带 LED 灯的数目及颜色模式，需要调用的指令如下：



之后要确定 LED 灯使用的范围，需要调用的指令是：

```
item range from 0 with 4 leds
```

最后要确定彩虹灯效色调范围，需要调用的指令是：

```
item show rainbow from 1 to 360
```

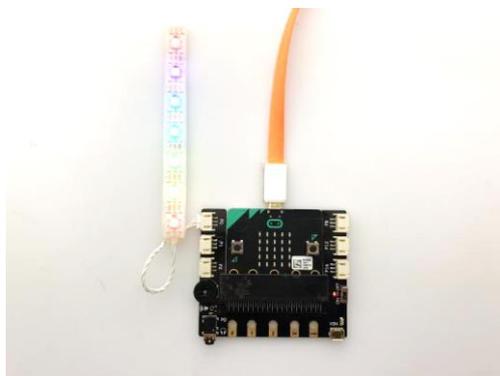
使用灯带的相关指令及说明如下：（在使用过程中，对参数会进行修改）



STEP3: 灯带连接引脚为 P0，包含 7 个 LED 小灯，设置彩虹灯效色彩范围 1-360（红色--绿色--蓝色），彩灯呈现七彩效果。调整参数后，最终程序。

```
无限循环  
NeoPixel at pin P0 with 7 leds as RGB (GRB format) range from 0 with 7 leds show rainbow from 1 to 360
```

实验效果



任务二：依次点亮彩虹灯带

实现功能

通过编写程序，将程序烧录至板子后，可以控制彩虹灯带的 LED 小灯每隔一秒点亮一盏 LED 小灯，呈现炫彩灯效。

编写程序

STEP1: 控制彩虹灯带的 LED 小灯每隔一秒点亮一盏 LED 小灯

需要运用可以存放变量的指令，每次来控制小灯被点亮的数目，需要调用指令的流程为：变量---设置变量，任意设定变量名（如：点亮小灯数），再单击确定。

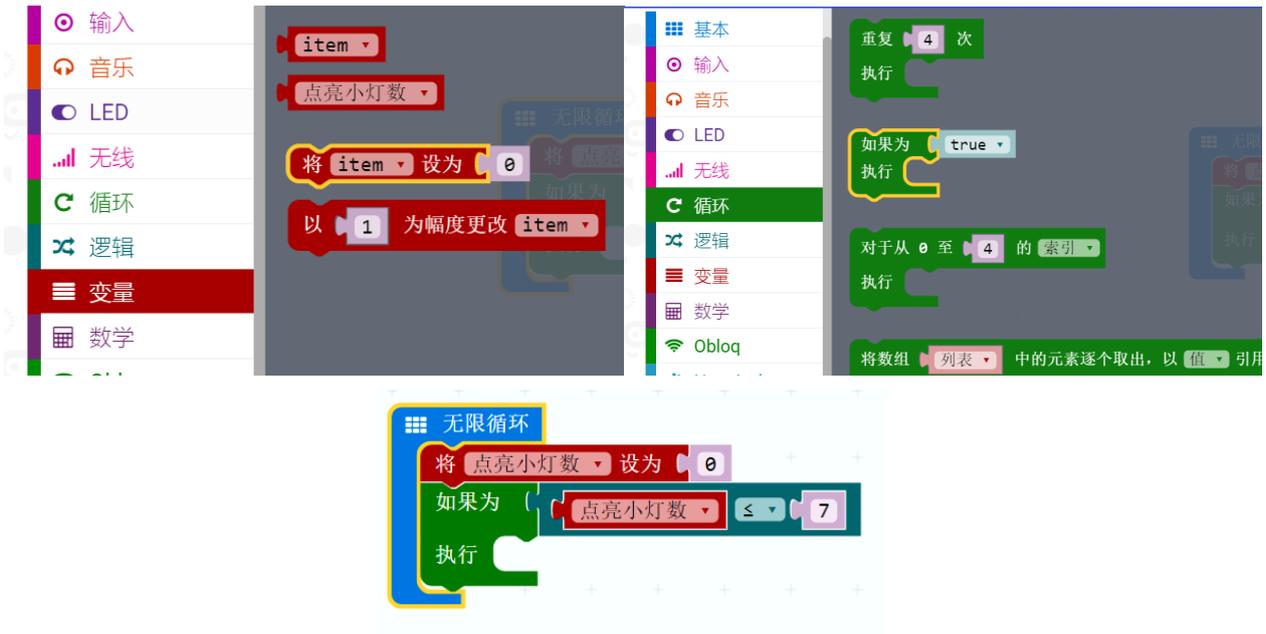


New variable name:

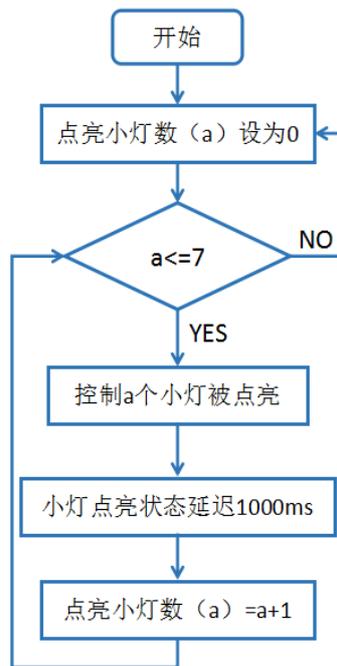
点亮小灯数

确定 ✓
取消 ✕

STEP2: “点亮小灯数”（也可以简称为“a”）最初设定变量的初值为 0（最初没有小灯被点亮），然后控制变量“点亮小灯数”在原有的基础上增加 1，需要运用到“循环”，当“点亮小灯数”不到 7 个的时候，继续点亮小灯。需要调用的指令如下：

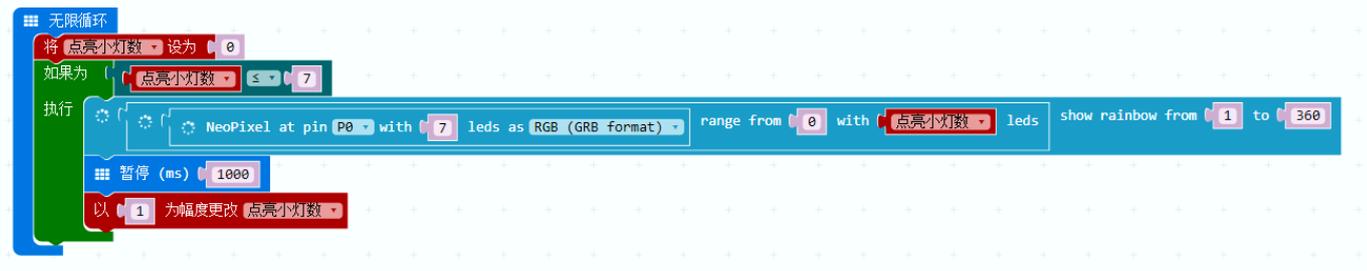


STEP3: 当“点亮小灯数” (a) 不到 7 时，可以点亮 a 个小灯，延迟 1000ms，之后让小灯数增加 1，流程图如下。



STEP4: 将点亮小灯数 (a) 最初设置为 0，点亮小灯数 (a) ≤ 7 时，控制点亮 a 个小灯，并延迟 1000ms，接着将点亮小灯数增加 1，流程为：变量---以 1 为幅度更改点亮小灯数。

如此循环，直到点亮小灯数大于 7，所有小灯熄灭，控制彩虹灯带的 LED 小灯每隔一秒点亮一盏 LED 小灯。



任务三：声音控制彩虹灯带

实现功能

最初灯带为熄灭状态，当发出声音时，点亮一盏 LED 灯，当没有声音时，灯带保持点亮部分小灯状态。

编写程序

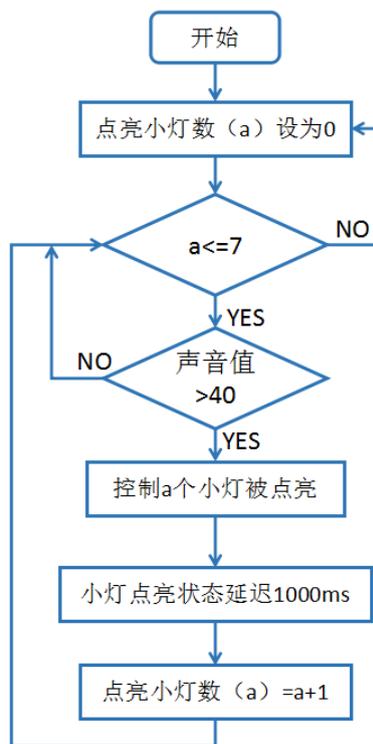
STEP1: 这个任务需要运用到声音传感器，当发出声音时，声音传感器可以检测到，控制灯带点亮 LED 小灯。之前章节学过声音传感器的用法，我们再来巩固一下调用方法。

调用指令的流程为：引脚（高级指令）---模拟读取引脚（P0），将 P0 修改为 P1。



STEP2: 当声音传感器检测到声音>40 且点亮灯的数量不超过 7 盏时，控制灯带点亮一盏 LED 灯，之后持续不断检测声音是否超过 40。

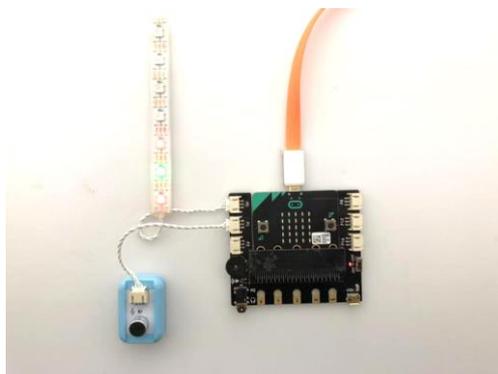
流程图：



STEP3: 最终程序。



实验效果



第四话：大展拳脚

经过前面这么多个项目的学习，你是不是对 micro:bit 已经了如指掌了呢？来挑战一下吧。
在这一章节的内容学习中，运用 micro:bit 主控板和传感器，制作出非常炫酷好玩的作品！

项目一 自平衡仪

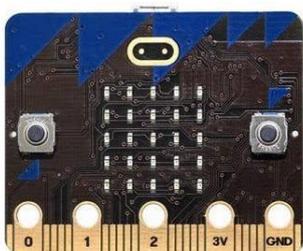
麦克的故事

麦克发现，你本来把手机拿在手里时是竖着的，将它转 90 度，页面就跟随你的重心自动旋转，也就是说页面也转了 90 度，这是一项极具人性化的设计。这是怎么做到的呢？

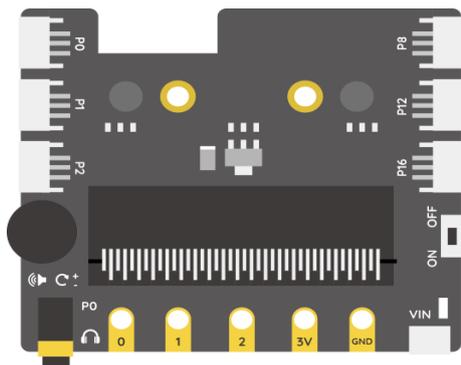
这节课的项目我们就是要来模拟这样的功能哦！

所需元件

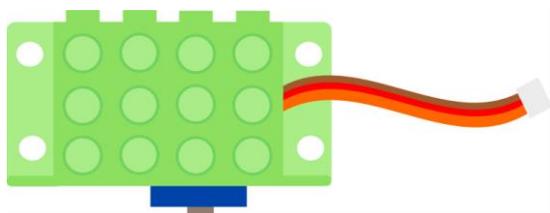
1 × micro:bit 主控板



1 × micro:bit 扩展板



1 × 舵机（伺服机构）



1 × MicroUSB 连接线



实现功能

通过编写程序，将程序烧录至板子后，可以让舵机（伺服机构）指针保持指向竖直上方，如果舵机上面支撑某个物体的话，那么物体始终保持竖直向上。

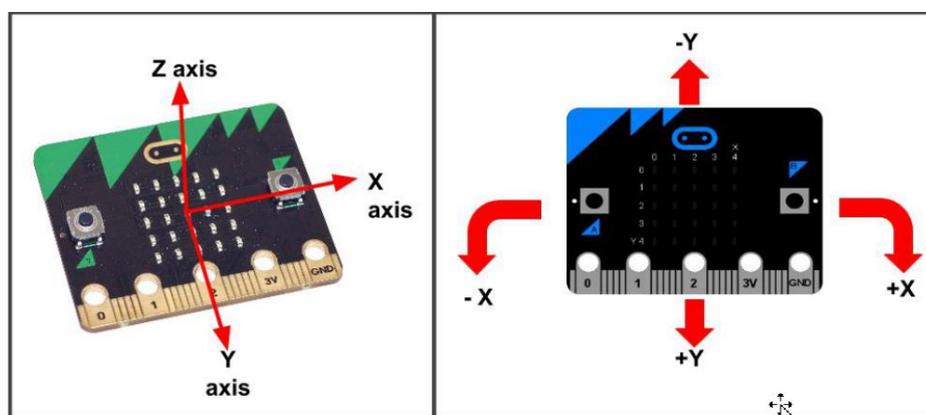
什么是重力加速度？

地球表面附近的物体因受重力产生的加速度叫做重力加速度，也叫自由落体加速度，用 g 表示。

重力加速度可以做些什么呢？

Micro:bit 自带加速度计，能实时检测姿态，从而控制舵机指针一直指向正上方。

Micro:bit 可以检测 X,Y,Z 三个方向的重力加速度，其中 X 沿着左右方向，Y 沿着前后方向，Z 垂直于板，沿着上下方向。本项目中主要检测 X 轴上的重力加速度。



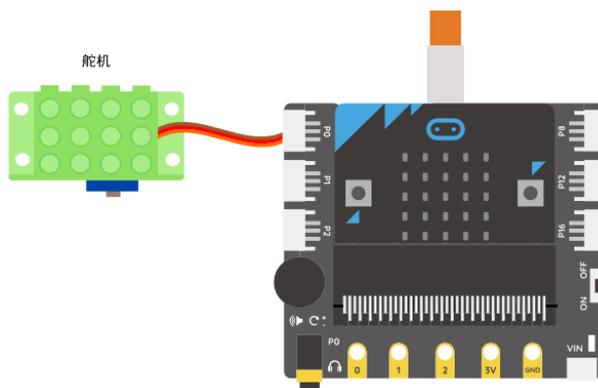
当主板平放在桌面上时，micro:bit 只受到竖直方向上的重力（Z 轴），所以 X,Y 都接近于 0。

而当板子产生左右倾斜时，X 的数值会发生变化。向左侧倾斜则 X 为负值，反之为正值。

所以我们可以获取 micro:bit 的倾斜角度，控制舵机旋转一个反向的角度，让舵机的指针一直朝上。

硬件连接

将舵机（伺服机构）与 micro:bit 的 P0 号接口相连接。



编写程序

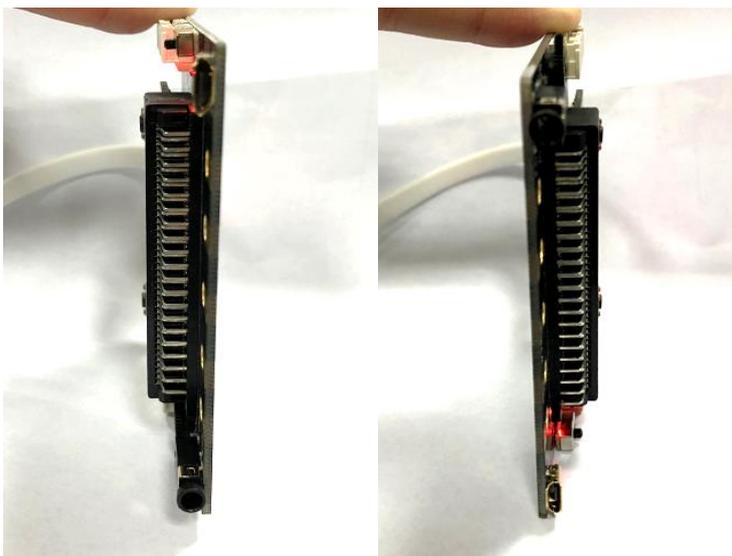
STEP1: 获取 X 轴的加速度

X,Y,Z 轴的数值可以在功能模块“输入”下找到。



加速度传感器的数值范围在-1023~1023 之间变化，你可以通过显示模块来查询变化范围。





STEP2: 数据处理

当 micro:bit 板子向左倾斜时，加速度传感器的 X 方向会产生一个负向的输出值，舵机（伺服机构）需要向右转动来抵消这个角度。

需要用到映射模块，将加速度 X 轴的值 (-1023~1023)，映射到舵机（伺服机构）的转动角度 (0~180)。映射之后，当 micro:bit 板子平放在桌面（桌面是水平的）时，舵机（伺服机构）角度为 90 度。

0 度	90 度	180 度
-1023	0	1023

在引脚中找到映射模块

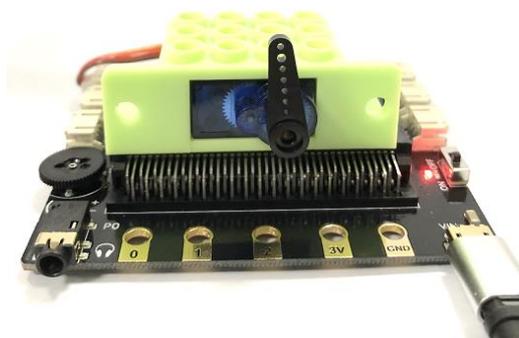


最终程序

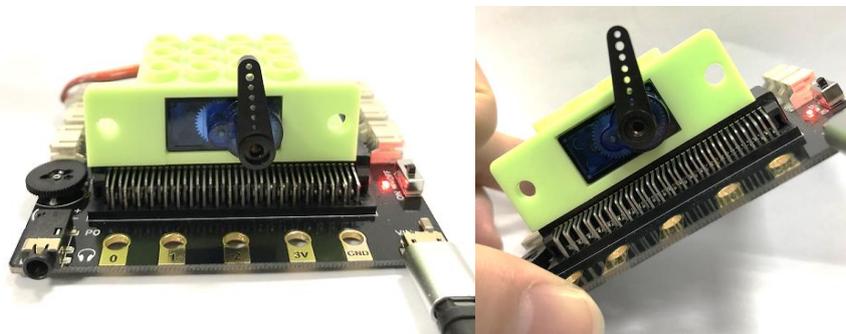


STEP3: 结构制作

用双面胶将舵机（伺服机构）和扩展板固定起来，使两者保持相对静止。



实验效果



能力提升

尝试使用 micro:bit 自带的电子罗盘功能制作一个指南针。



项目二 DJ 演奏台

麦克的故事

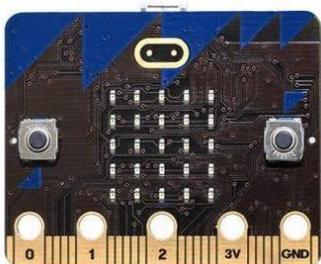
马上要到圣诞节了，“要在圣诞晚会上表演个什么节目呢？”麦克很苦恼。如果能够像一个 DJ 一样在舞台上演奏，那一定很棒吧。



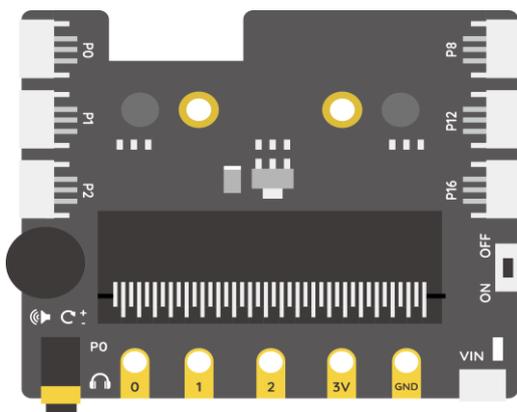
麦克决定用 micro:bit 与 BOSON 做个可以像 DJ 一样演奏台的装置，一起来认识一下神奇的 micro:bit 与 BOSON 吧。

所需元件

1 × micro:bit 主控板



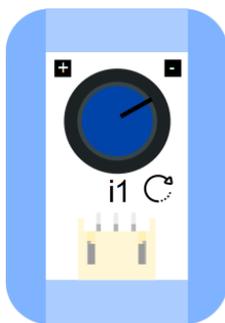
1 × micro:bit 扩展板



1 × 彩虹灯带



1 × 旋钮模块



1 × MicroUSB 连接线



实现功能

当用手调整旋钮的转动角度时，实现音乐演奏，同时配合灯带的效果实现。

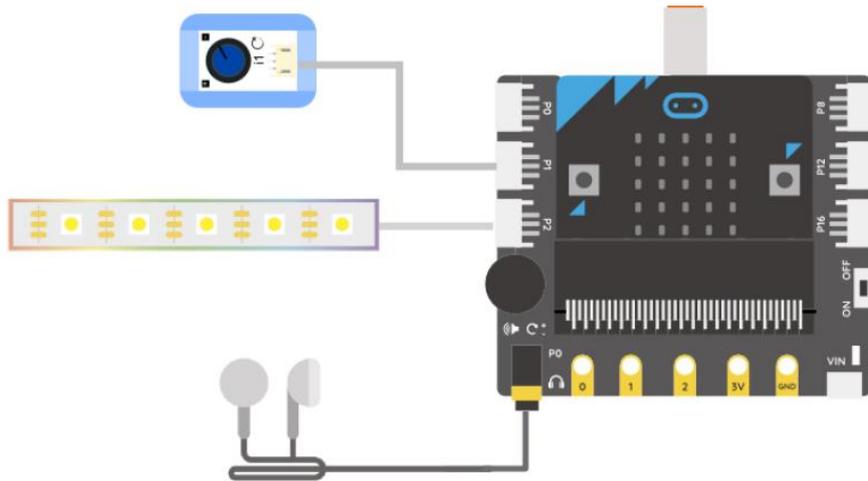
当用手拨动旋钮时，对应的 P1 号引脚读出的模拟值在 0~1023 范围变化，分隔成 8 个小范围，对应音乐中的一个八度 “Do~Do’ ”，对应灯带上个数的变化。

旋钮 (P1) 的输入值	声音	灯带
0~128	Do	1 颗灯亮
128~256	Re	2 颗灯亮
256~384	Mi	3 颗灯亮
384~512	Fa	4 颗灯亮
512~640	So	5 颗灯亮
640~768	La	6 颗灯亮
768~996	Xi	7 颗灯亮
996~1023	Do’	7 颗灯亮，都为红色

硬件连接

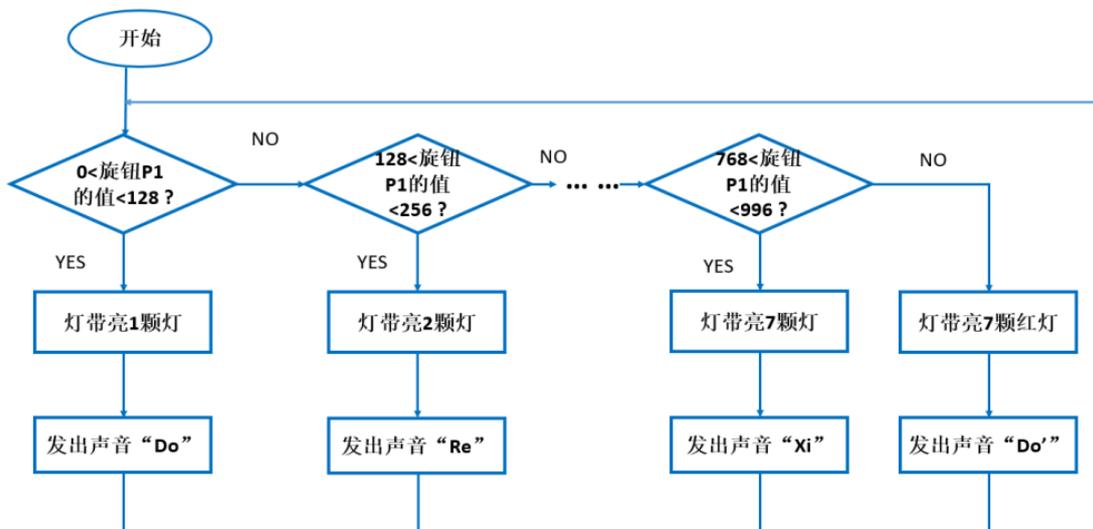
将旋钮模块接在扩展板的 P1 号接口；

将彩虹灯带接在扩展板的 P2 号接口。

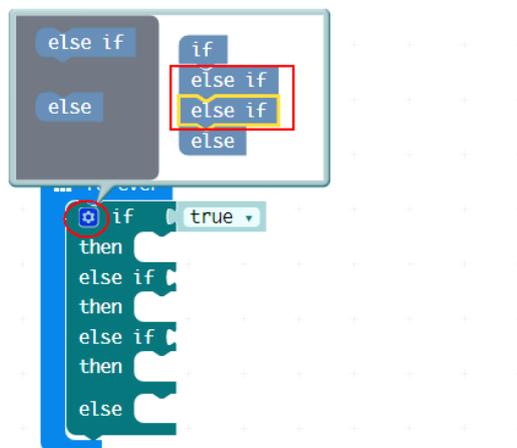


编写程序

STEP1: DJ 演奏台的实现逻辑图。



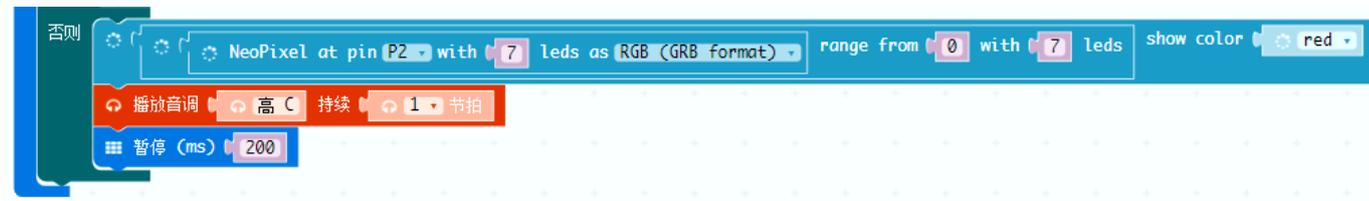
STEP2: 逻辑模块“if”的编辑，逻辑模块可以进行以下编辑操作。



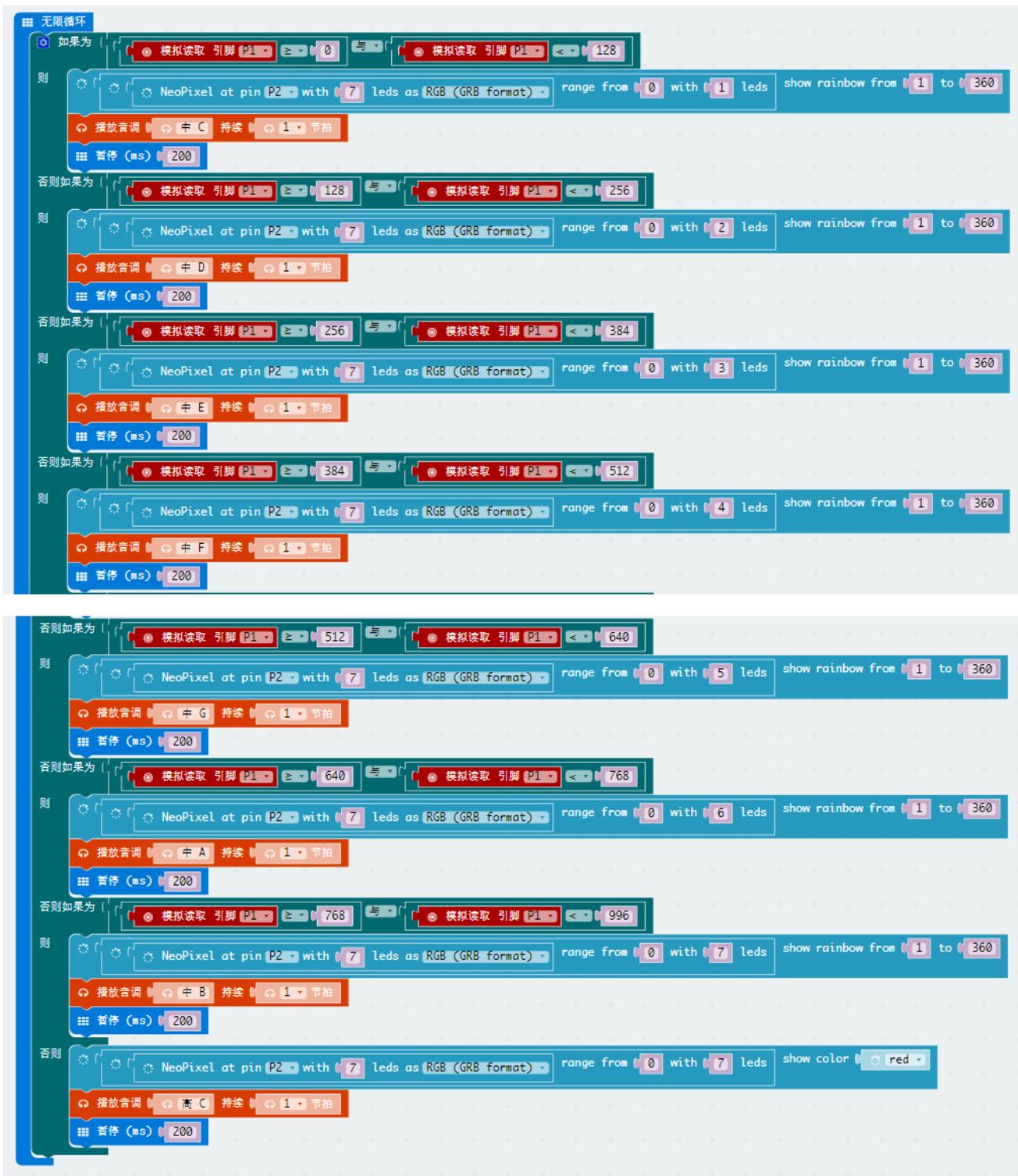
STEP3: 判断旋钮输入 (P1) 的值的范围，对应灯带与音效。以下是第 1 个区域的控制程序。



STEP4: 由于只有 7 颗灯，在第 8 个区域中，可以通过不同的灯带表现形式来表达 “Do”。



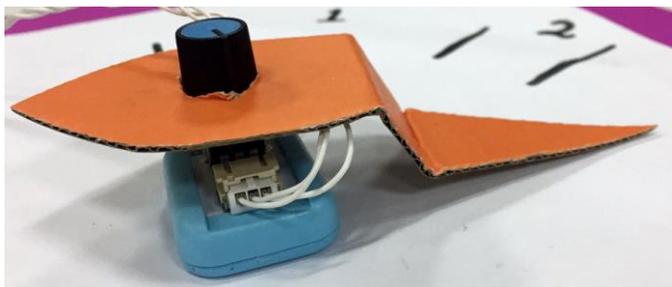
STEP5: 最终程序。



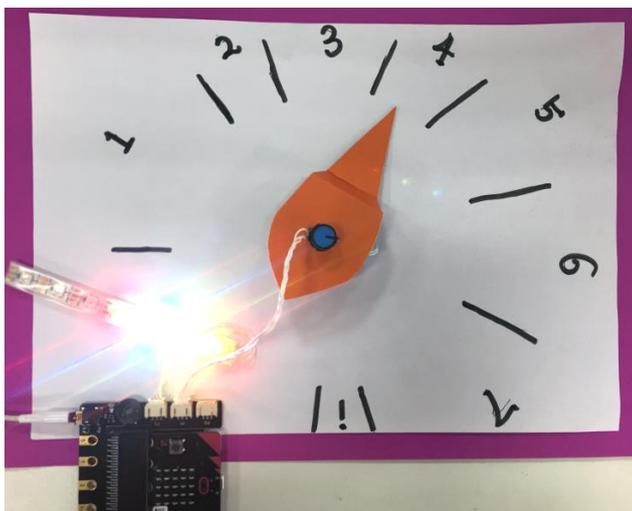
实验效果

为了可以方便进行演奏，需要给 DJ 演奏台做一个便于操作的外观。

1. 给旋钮做个手柄，方便进行操作



2. 将旋钮固定好之后，将每个音的范围标注出来。



能力提升

把 micro:bit 主板的 5x5LED 点阵也利用起来，当演奏出不同的音节时，对应显示该音节。尝试一下制作一下吧！

项目三 可移动门铃

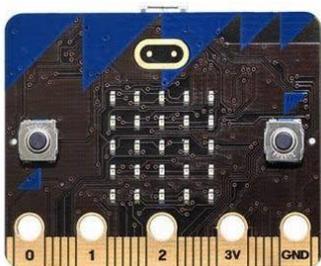
麦克的故事

麦克约了小伙伴来家里玩耍，结果小伙伴按了几次门铃，都没有人来开门，生气的走了。麦克知道后，感觉很郁闷，自己在卧室听音乐，并没有听到门铃的声音。麦克决心解决这个问题，做一个可以远程移动的门铃，不用担心自己在房间中听不到门铃声音。

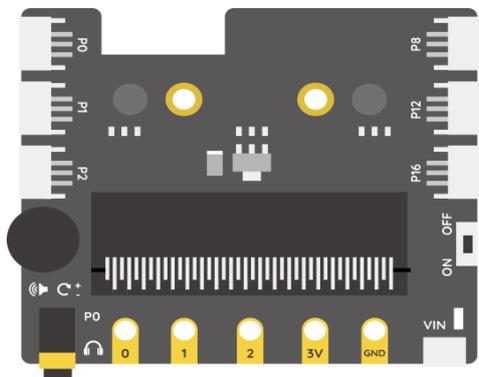


所需元件

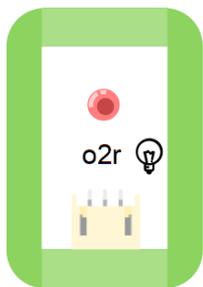
2× micro:bit 主控板



2× micro:bit 扩展板



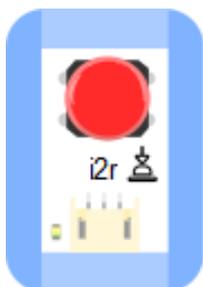
1× LED 灯模块



1 × 运动传感器模块



1 × 按钮模块



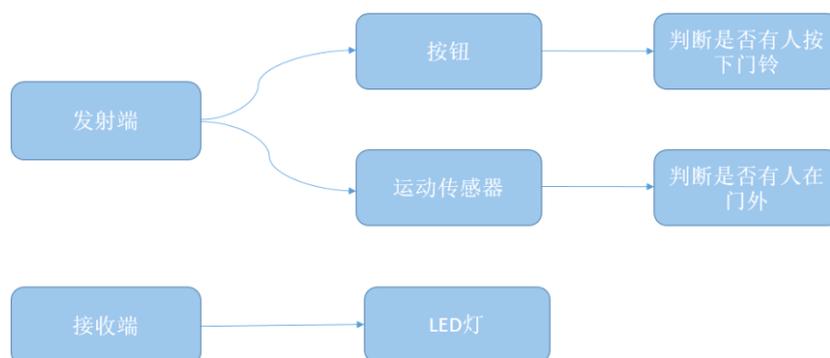
2 × MicroUSB 连接线



实现功能

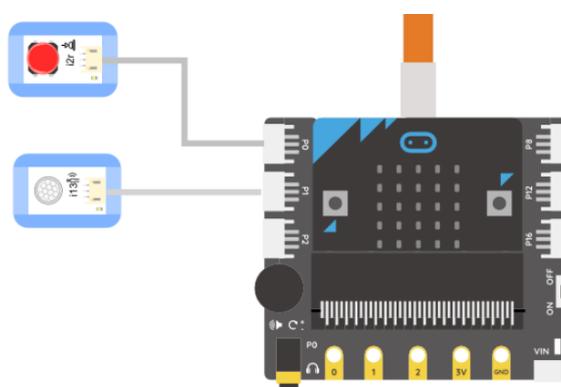
可移动门铃基于 micro:bit 的 Redio 功能，需要两块 micro:bit，一块是发射端，另外一块是可以移动接收端。

综合运用了闪烁的 LED、呼吸灯和自动门中所学到的知识。

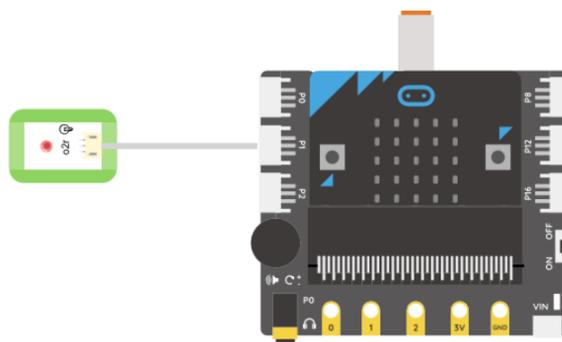


硬件连接

发射端硬件连接：



接收端硬件连接：



编写程序

设置发射端程序

STEP1：基于已有的编程基础，我们已经掌握对于按钮与运动传感器这两种输入模块的编程方法。可移动门铃的发射端需要将是否门外有人的消息，远程发射出去，怎么实现呢？这就是这个项目中需要重点学习的知识：无线传输。

在指令区中有“无线”指令模块。



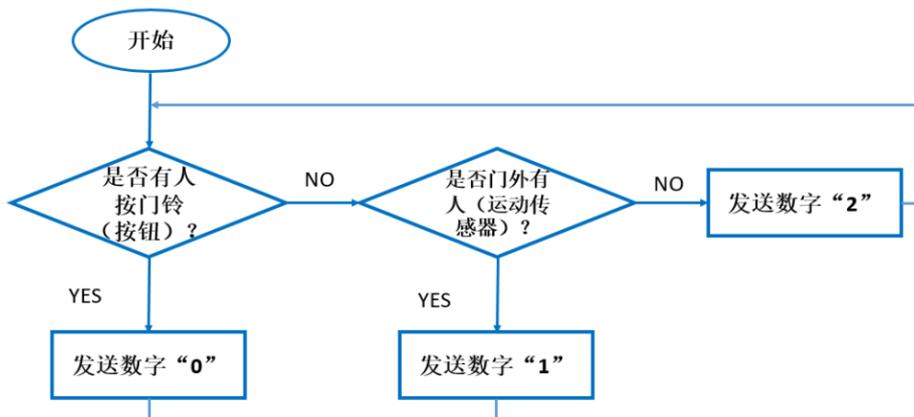
无线功能的使用中有三个常见的模块：



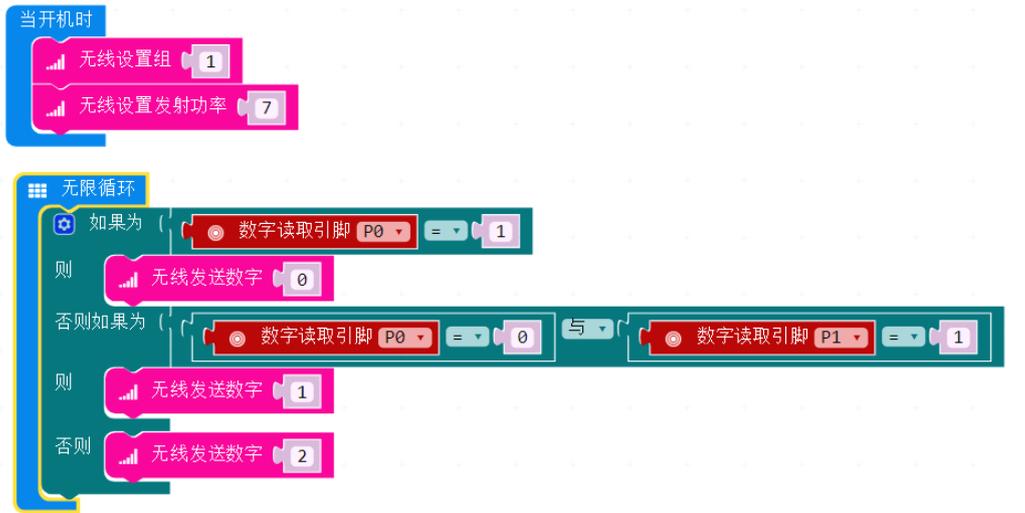
接收端需要识别接收到的信号，做出对应的动作，这里对应发射端有数字和文字两种情况。

注意，“receivedNumber”代表接收到的当前数字。

STEP2: 了解了无线功能的运用方法，对发射端进行程序编辑。编程流程图。

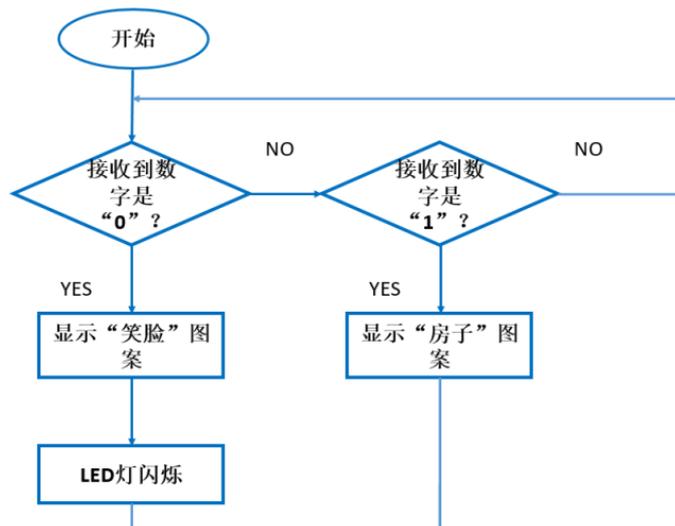


STEP3: 发射端完整程序。



设置接收端程序

STEP1: 接收端功能流程图。



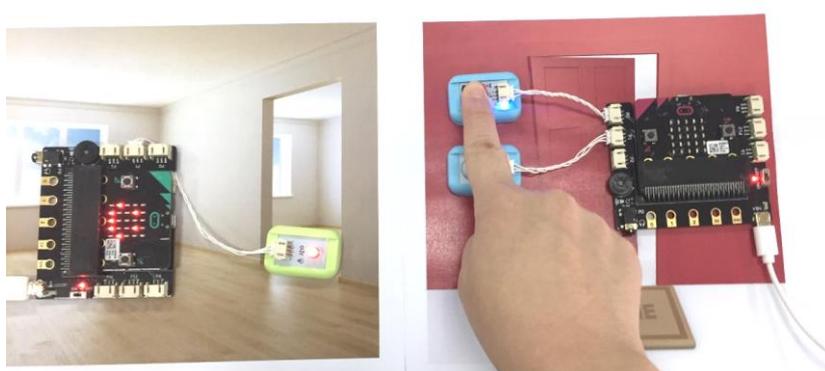
STEP2: 接收端完整程序。

实验效果

1) 当门外有人时



2) 当门外有人按门铃时



能力提升

在门外按门铃的人一定很焦急，请为门外的门铃，设定“请稍等”的声音。

项目四 拆弹游戏

麦克的故事

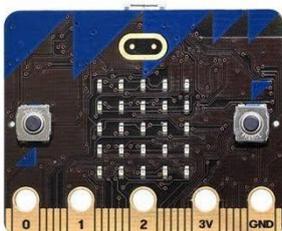
小朋友们要来麦克家里做客了，麦克很高兴，麦克想把自己最喜欢的玩具拿出来和小朋友们分享，有遥控汽车，遥控飞机

麦克觉得这些玩具没有什么新奇的，怎样才能够做一个属于麦克的游戏呢？麦克决定用 micro:bit 和 BOSON 做一个炫酷的游戏——拆弹游戏，考验一下小伙伴们的反应能力吧。

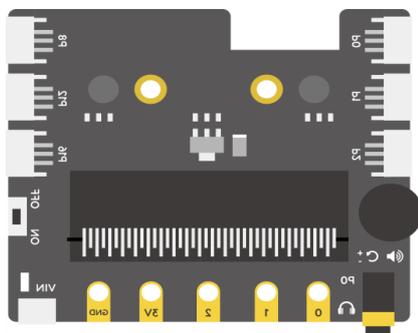


所需元件

1 x micro:bit 主控板



1 x micro:bit 扩展板



1 x 彩虹灯带



1 x MicroUSB 连接线



实现功能

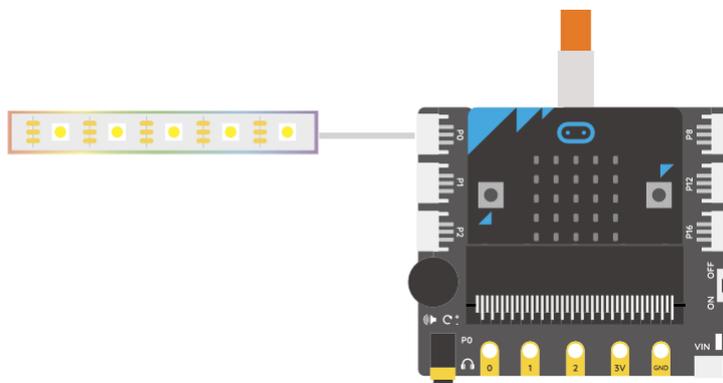
启动 micro:bit 之后，LED 灯模拟的引线会开始燃烧，当引线烧至最后时炸弹会引爆。麦克需要在炸弹引爆前，根据 micro:bit 提示的方向依次将板子向不同的方向倾斜，快速解锁炸弹才能避免“爆炸”。

这样的拆弹装置由三个部分组成：

1. 动作识别：用来识别倾斜的方向是否和指示的方向一致。
2. 灯光交互：为 LED 灯设置引线燃烧的效果，经过时间越长剩余点亮的 LED 灯越少。
3. 计分系统：在 LED 灯全部熄灭之前完成 20 个动作则拆弹成功；LED 灯全部熄灭时仍然没有完成，则判定游戏结束。

硬件连接

将彩虹灯带接在扩展板的 P0 号接口。

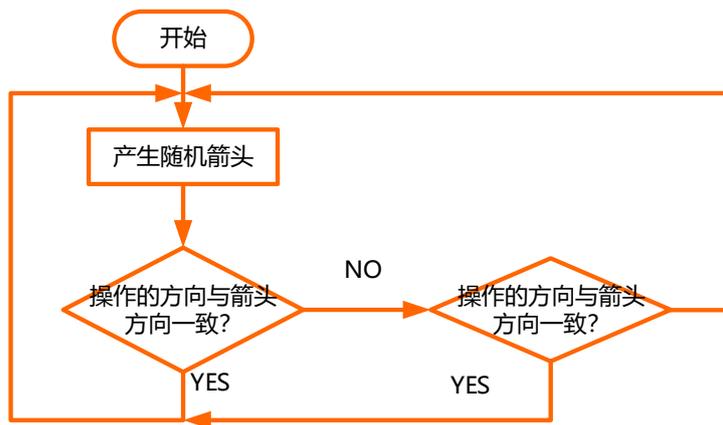


编写程序

STEP1：动作识别

首先我们需要让 micro:bit 能够产生并显示随机的方向，在我们向该方向倾斜操作时能识别出来，随后随机显示下一个方向。这样的一个程序的流程图大致如下。

在游戏过程中，只有当我们的操作与随机产生的指示箭头一致时，才会进行下一步操作，否则一直保持当前的箭头方向。

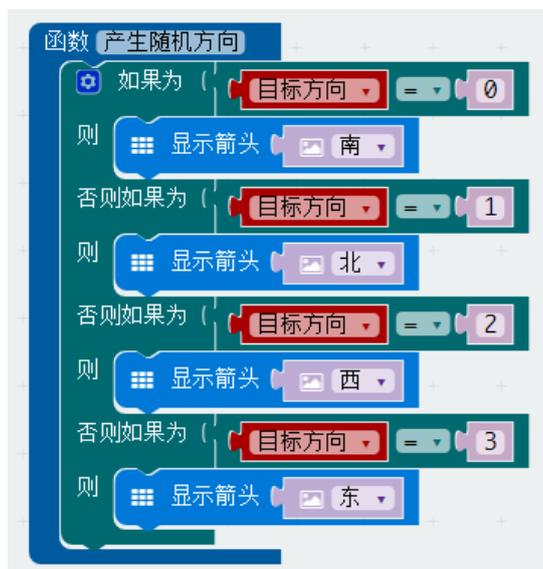


怎样产生随机方向呢？

可以用 0~3 四个数字代表四个方向，使用数学 Math 下的子模块产生随机数模块“选取 0 至()”来从 0~3 四个数字中随机选择一个数字。



建立一个函数“产生随机方向”，对应产生的随机数，显示不同的箭头。



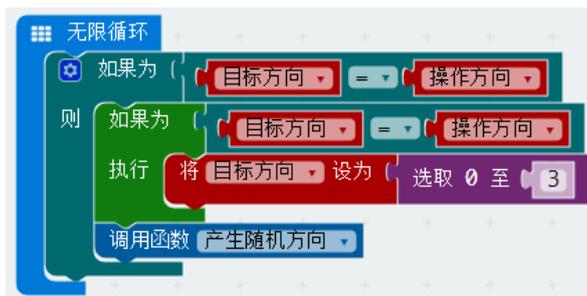
怎样判读倾斜操作的实际方向呢？

通过在输入 input 下的姿势识别模块来读取 micro:bit 的倾斜方向，并且分别对应到 0~3 中。

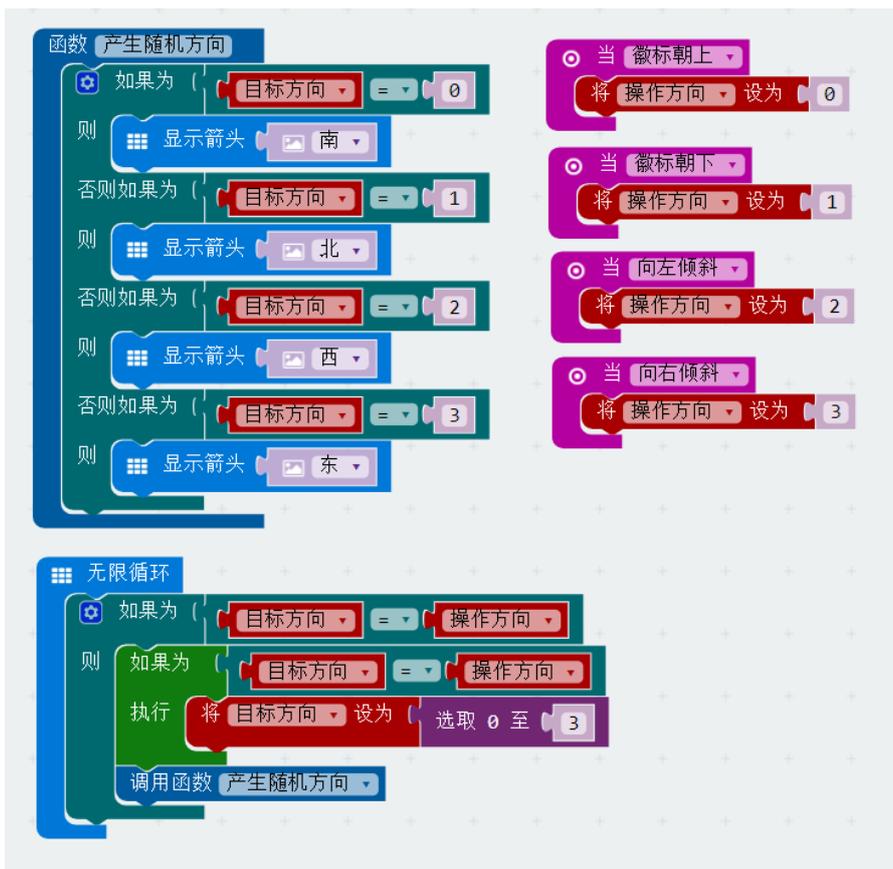




通过一个如果 if 语句条件判断，操作方向与目标方向是否一致。



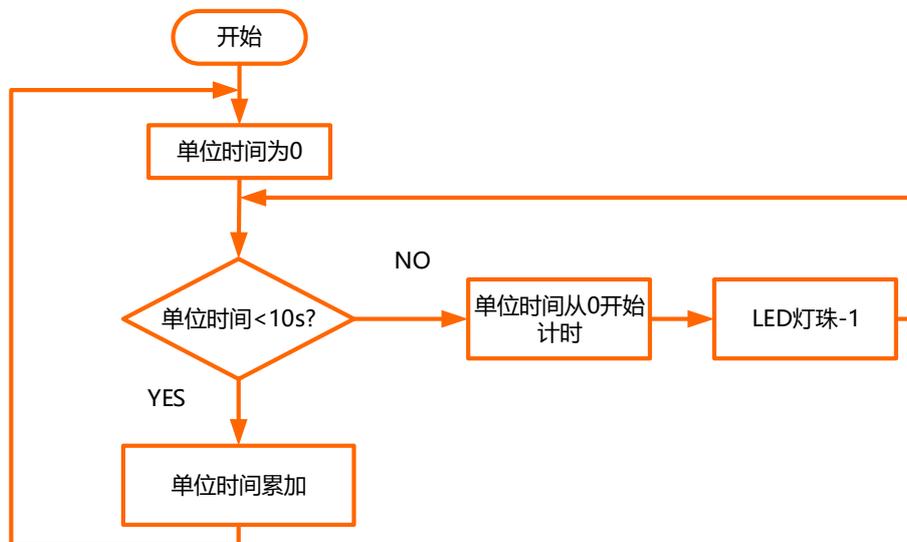
这个部分的完整程序为：



注意：“如果为 true 执行”语句用于保证两次随机出现的方向不相同。

STEP2: 制作炸弹程序

炸弹的效果通过彩色灯带显示，使用灯带来模拟炸弹的保险丝。随着时间的推移，保险丝上的 led 灯珠将会以稳定的速度相继熄灭。所以反应速度越快，小灯剩余数量越多。



怎么实现倒计时？

建立时间函数，通过计数的形式，设置每次计时的时间间隔，为 $(1 \times 100) \times 100\text{ms}$ 。一次计时间隔结束，小灯数量减 1。

注意：因为 micro:bit 在处理数据时会消耗到一定的时间，所以倒计时模块实际运行时走得慢一些。

时间倒计时的同时，小灯的显示部分：

这个部分的完整程序

```
当开机时
  将 时间 设为 0
  将 小灯数量 设为 7
  将 灯效控制 设为 NeoPixel at pin P0 with 7 leds as RGB (GRB format)

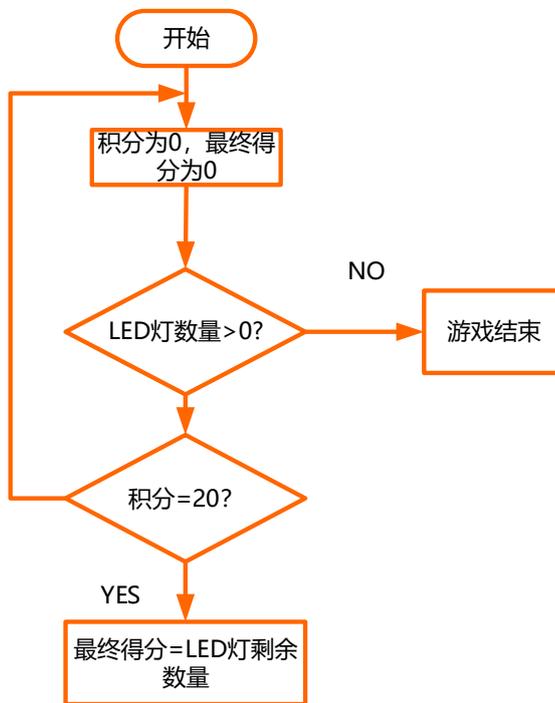
无限循环
  如果为 (时间 <= 100)
  则
    以 1 为幅度更改 时间
    暂停 (ms) 100
  否则
    将 时间 设为 0
    以 -1 为幅度更改 小灯数量

无限循环
  灯效控制 clear
  灯效控制 range from 0 with 小灯数量 leds show color red
  灯效控制 show
```

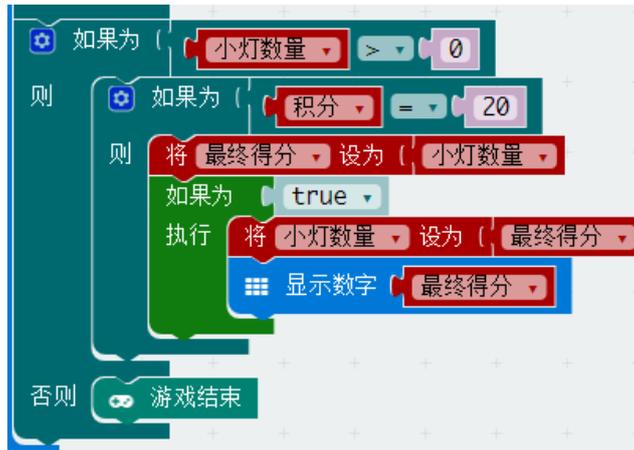
STEP3: 增加计分功能

在操作过程中，每次操作方向和目标方向一致，积一分，直到积满 20 分，并且此时 LED 灯珠数量大于 0，那么挑战成功，拆单成功，最终得分为 LED 灯珠剩余个数；否则，失败。

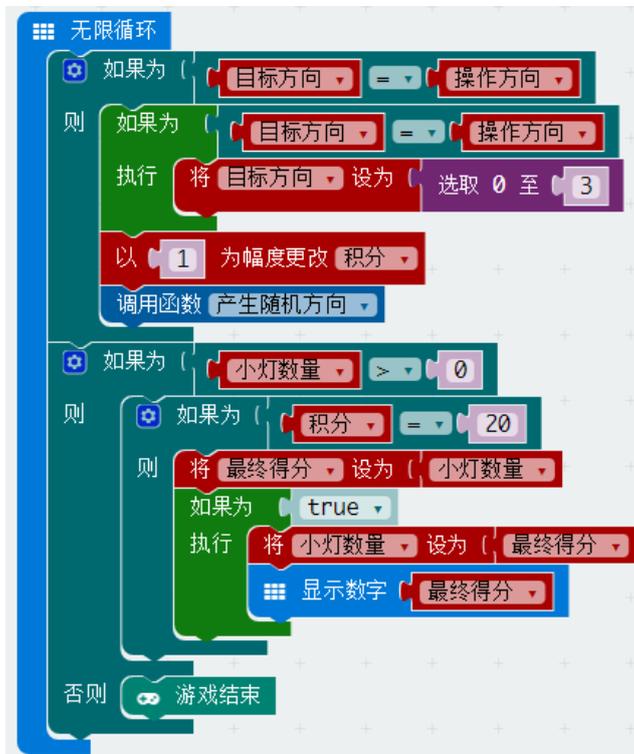
流程图大致如下：



判断是否成功及失败的程序如下：



计分部分可以合并在主程序中



STEP4: 最终程序。



能力提升

1. 尝试加快 LED 灯的熄灭速度，增加游戏难度；
2. 将上下左右方向用其他图形替代，挑战更高难度的下的反应力。

希望你的 micro:bit 之旅不会止步，用你的奇思妙想，玩出更多新颖有创意的作品。如果你愿意与我们分享的话，也可以直接登陆我们的论坛，让我们的社区论坛记录下你的点点滴滴！

欢迎登陆 DFRobot 创客社区！

DFRobot 创客社区: www.dfrobot.com.cn