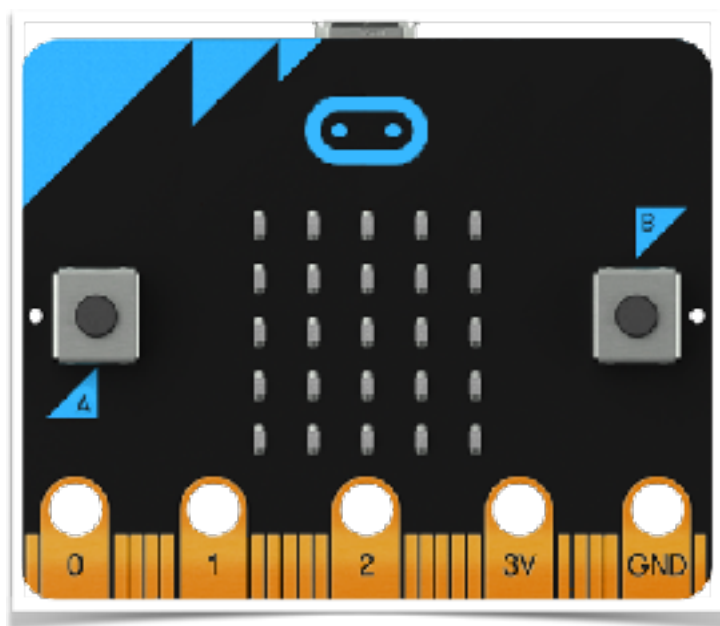


micro:bit基础课程 (makecode版本)

指尖新空间

指上创客课程开发团队



前言

首先很感谢micro:bit给了我们这么好的一个可以进行教学使用的开源硬件，感谢猫友会以及micro:bit协作组的几位老师让我接触到该硬件。

其次，本课程是指尖新空间的指上创客教育课程开发团队利用休息时间一天内完成的，所以难免有各种瑕疵。本课程旨在抛砖引玉，不存在任何商业目的，所以大家可以自由使用和分享。但版权属于原创团队，请大家务必遵守开源版权协议：切勿用于商业用途。

本课主要面向对象为小学生，是以大班教学为课程背景进行的开发。所以在器材上没有对micro:bit进行任何的扩展和补充，所以请大家在使用时根据各校情况自由加入相应的材料或者内容。

同时，也欢迎大家分享自己的优秀课程案例给我们。

关于课程分享及课程建议，大家可以email向老师：
xiangjin@xjtu.edu.cn。

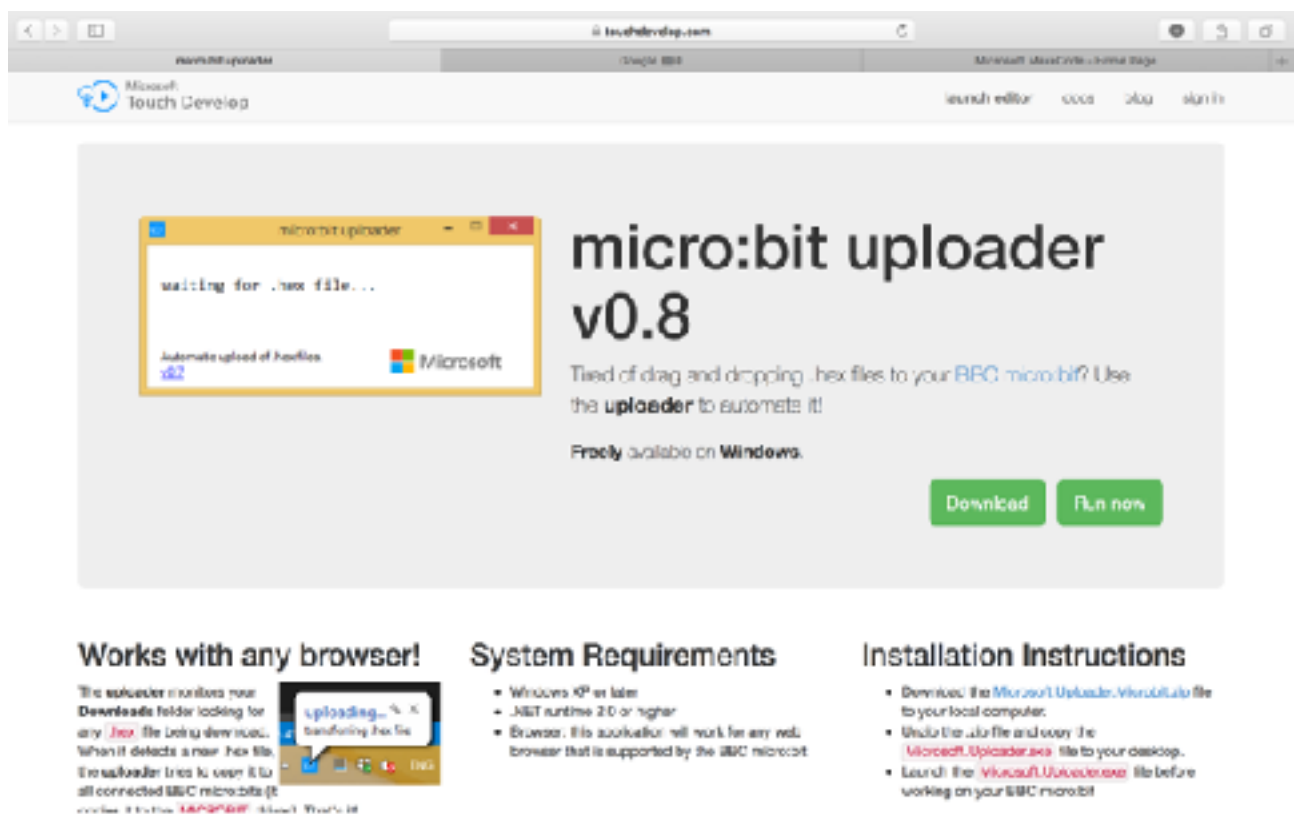
最后，再次感谢指尖、指上的小伙伴们为本课程做出的无私奉献。

分享一个技术诀窍：

自动下载程序到micro:bit

如何在Windows下自动将micro:bit的程序下载进去？

1、进入网站：<https://www.touchdevelop.com/microbituploader>（如下图）



2、下载软件micro:bit uploader

3、解压缩软件。

4、在准备下载之前请启动软件，这时候软件会自动监测是否有.hex后缀的文件下载下来，如果有，它会自动将该文件复制到micro:bit中。

5、说明：适用于所有的浏览器（是否如此，我还没有测试）。

课程目录

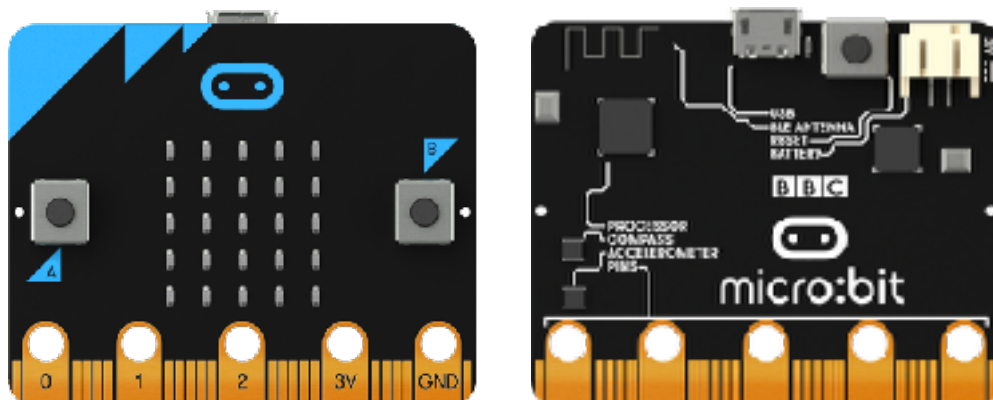
前言	2
自动下载程序到micro:bit	3
第一课：开始使用micro:bit	5
第二课：学会数数	11
第三课：屏幕橡皮擦	15
第四课：聪明的micro:bit	18
第五课：智能夜灯	21
第六课：温度计	23
第七课：指南针	26
第八课：骰子游戏	31
第九课：手表	36
第十课：状态指示器	39
第十一课：电报机	42
第十二课：计分器	45
第十三课：智能闹钟	47
第十四课：探究加速度（物理体验课）	50
第十五课：萤火虫（生物体验课）	54

第一课：开始使用micro:bit

大家可能在之前都没听过micro:bit. micro:bit是一个由英国广播公司BBC 推出的微型可编程计算机，类似于树莓派，但是体积更小，它的尺寸只有4厘米x 5厘米。micro:bit采用32位ARM Cortex处理器供电，配有内置的5x5 LED矩阵，为用户提供25个可单独编程的红色LED，用作显示屏，另外还有一些按钮，以使用户可以与游戏和程序进行交互，其它内置配件包括加速度计，磁力计和蓝牙天线。我们在今后的课程里就教大家如何使用micro:bit来做一些非常有趣的东西。

一.知识储备

1. 认识micro:bit

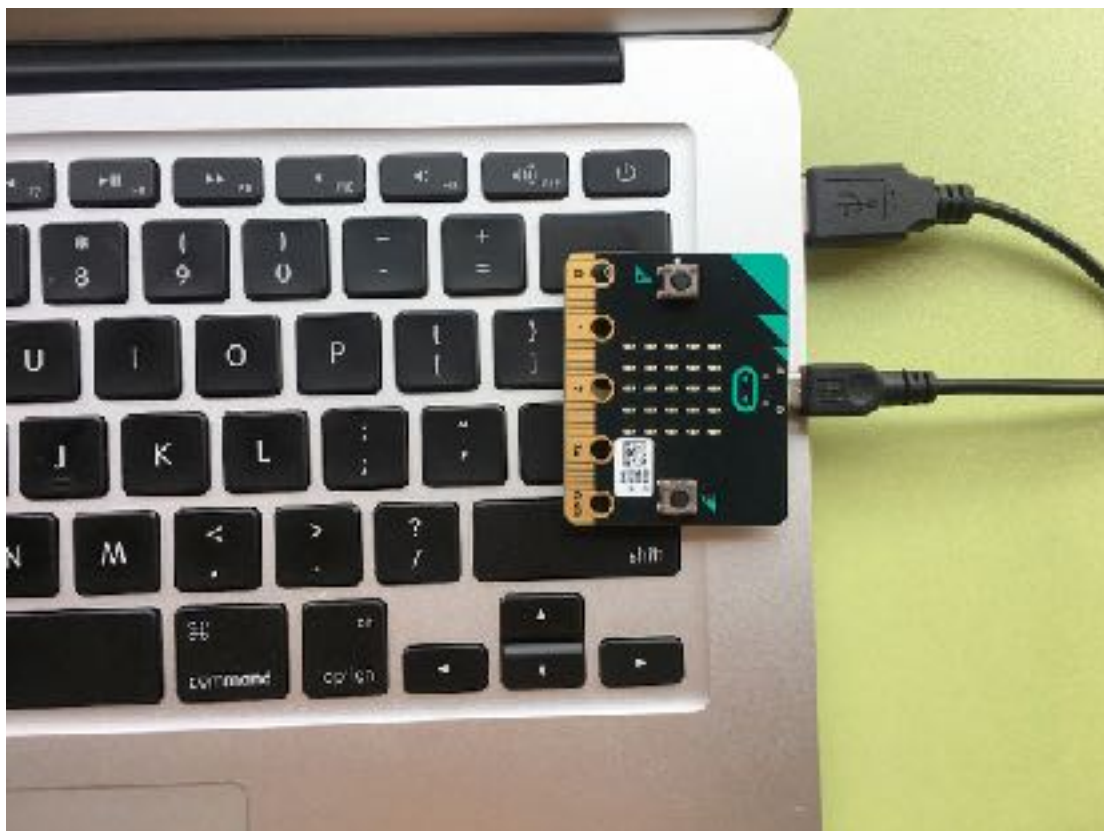


MicroBit是一块英国BBC针对青少年学习编程，集国际大企业共同设计开发的开源实验板，它因教育而生，是人工智能教育的开发工具，它具有独立的运算存储编译功能，板卡上集成了地磁、运动、光电、温度传感器，并有一个5X5的LED阵列显示字及图案，蓝牙及无线电通讯也是其亮点。通过USB与计算机连接就可以实现数据传送。微软、谷歌线上编程可使用可视、代码方式完成并配以虚拟演示功能，非常适合只能给小学生STEM项目学习使用。Python也为其开发出PythonEditor使用可视、代码两种编程方式。是目前基于硬件的编程学习最受老师和同学欢迎的实验板，如果使用图形化编程，没有任何程序语言学习就可以实现简单应用。

二.准备工作

一.连接micro:bit

首先，打开包装取出micro:bit。然后找到随机附带的usb连接线。用usb线把micro:bit连接到电脑usb接口上，



此时，打开电脑，会在我的电脑里发现多出一个硬盘名字叫microbit，如下图所示，这个驱动器我们到时候会用到。



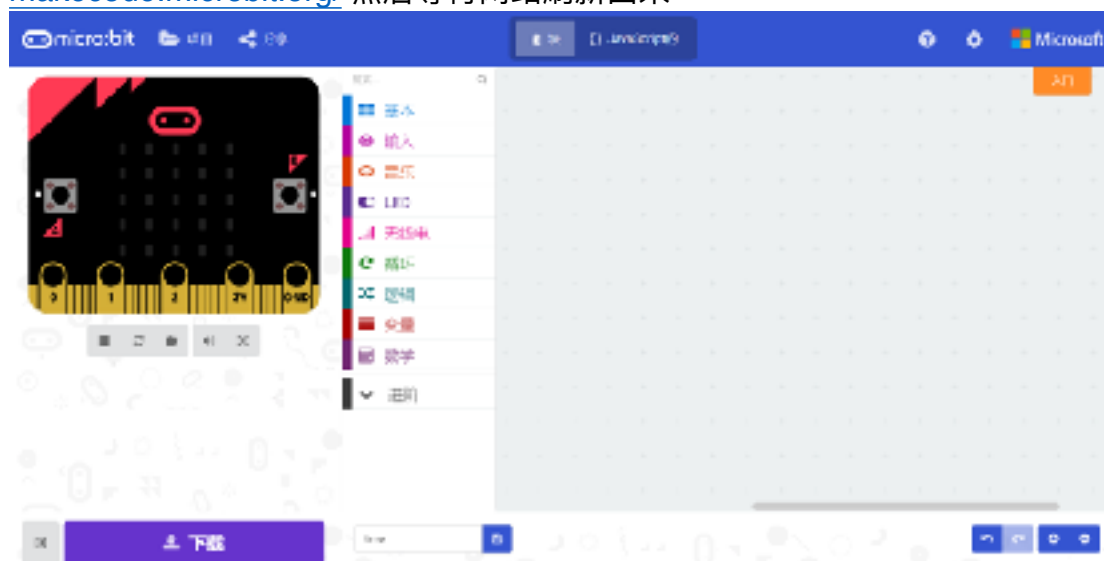
到此为止你就可以开始你的编程之旅了。有的同学可能会问，怎么没有安装驱动呀？对！我们这个micro:bit就是不需要驱动。

三.做一个漂亮的图案

大家在连接好micro:bit之后，一定很迫不及待想看一下是怎么工作的，现在我们就教大家来如何点亮我们的micro:bit。

首先我们编程是在一个网页编程里做的，大家打开浏览器，输入网址：<https://>

makecode.microbit.org/ 然后等待网站刷新出来

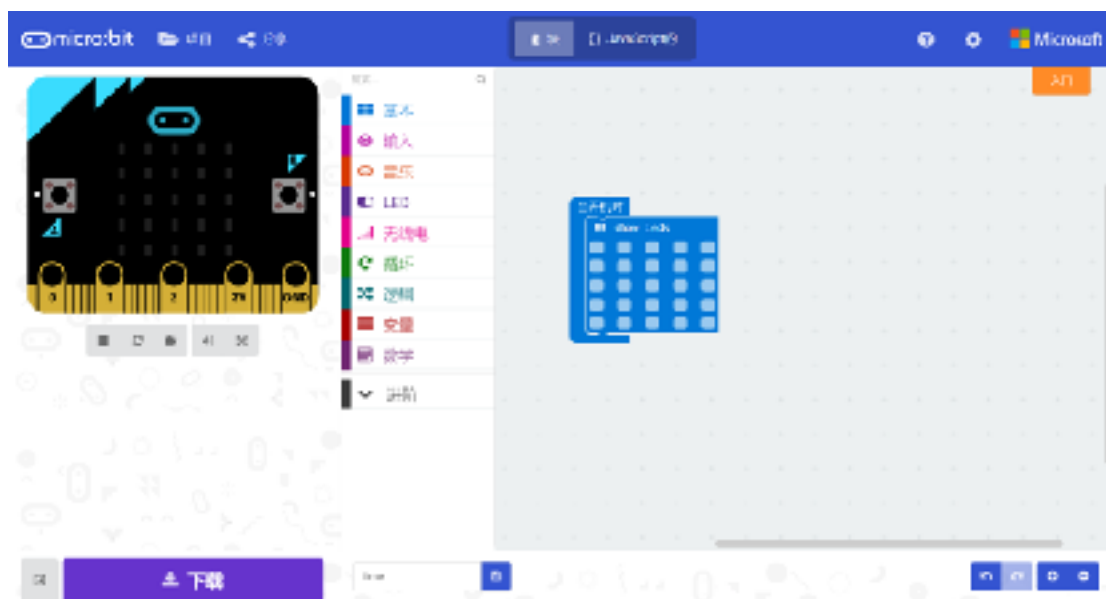


在网站的最左边是一个模拟窗口，它可以模拟我们micro:bit的工作，中间这一块是我们的指令区，里边有我们需要用到的各种程序模块。最右边空白区域是我们的脚本区，我们把指令拖拽到脚本区，就构成了我们的程序。接下来我们就用我们的程序做一个漂亮的图案。

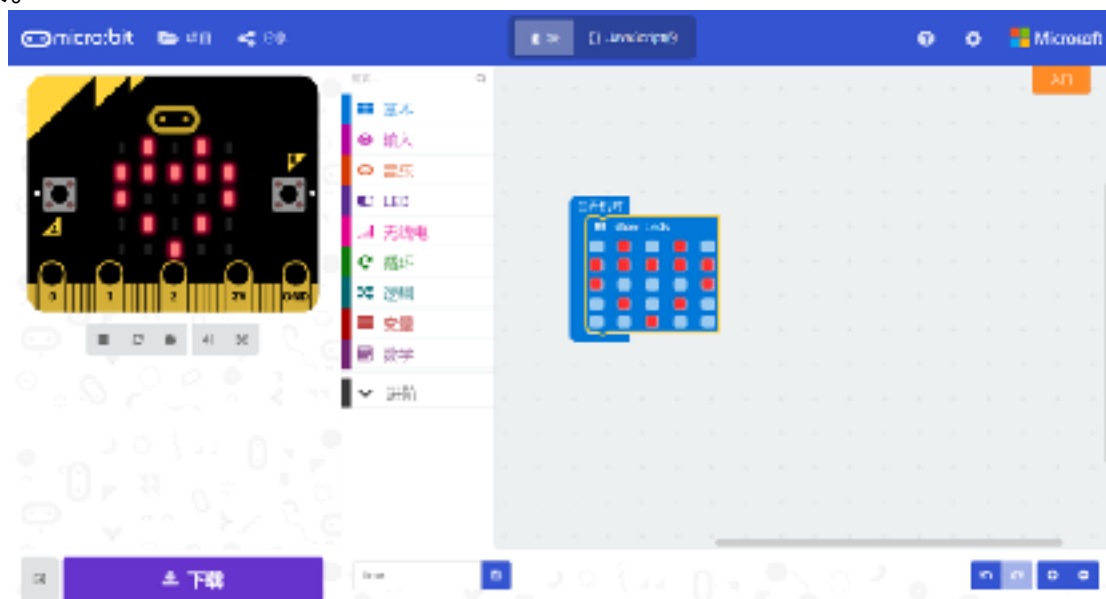
第一步，点击指令区的基本选项，然后选中弹出菜单的“当开机时”模块，并拖出



第二步，同样点击基本按钮，然后拖出“show leds”模块，并放入“当开机时”模块内



第三步，在“show leds”内点击出自己喜欢的图案，可以在左边模拟窗口看到模拟效果。



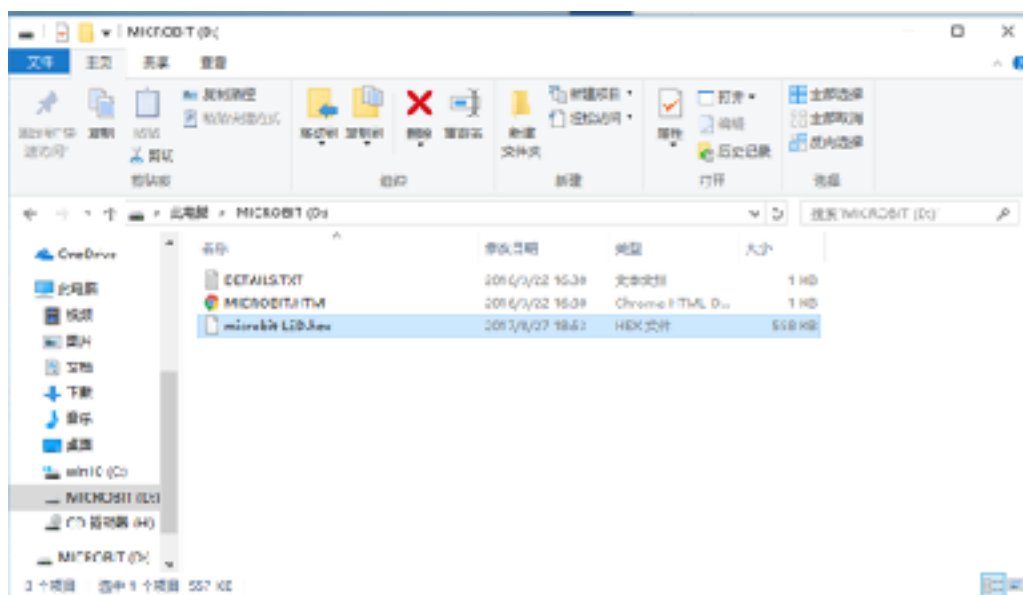
第四步，在如下图所示红框内修改文件名。



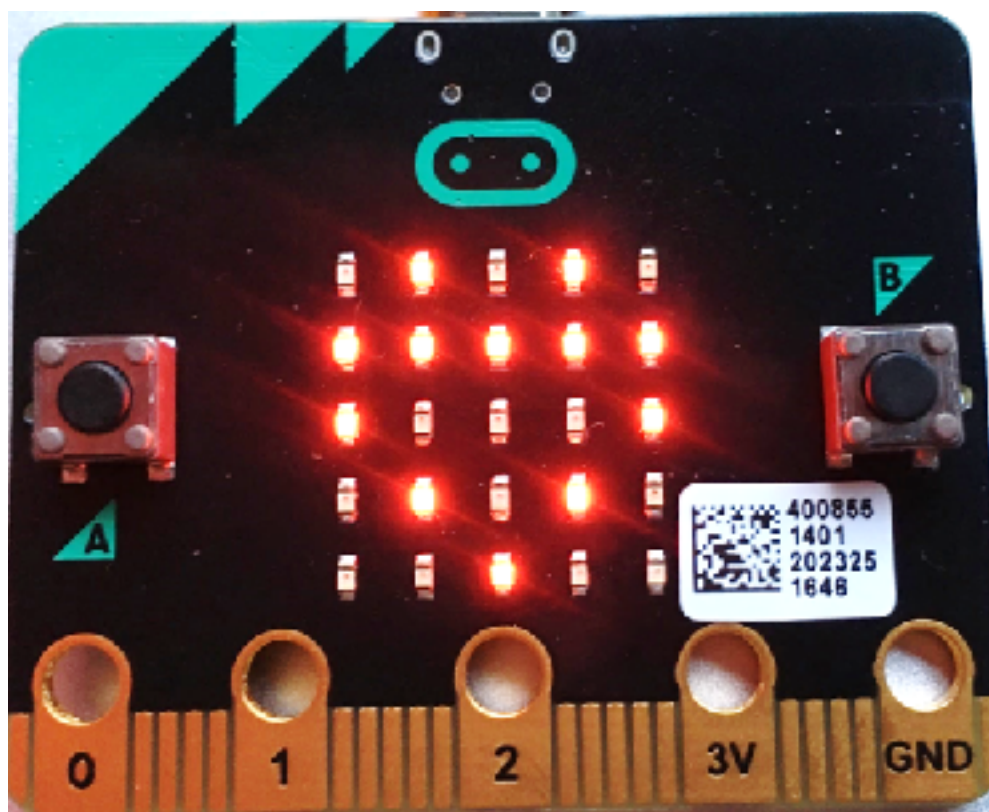
第五步，点击下载，



第六步，将下载好的文件拖入“MICROBIT”磁盘中，



现在看看你的micro.bit



到此为止，我们就完成了对micro.bit的一个简单的编程，是不是很简单？

第二课：学会数数

同学们还记得在刚上幼儿园的时候学会的第一件事是什么吗？我想大多数同学都是先学会数数的吧，我们的有的同学是先学会从0数到10，有的同学可以根据规律从0数到100。今天我们就教我们的micro.bit数数，大家一起比一比，看谁数的多。

【任务目标】

了解如何显示数字，“show number”生成数字。我们将学习如何使用“show number”以及简单的命令（如暂停）创建数字。

【知识点】

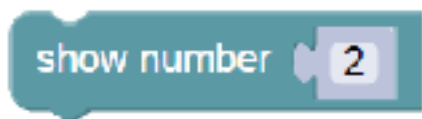
1. 学习“show number”命令的使用
2. 学习使用延迟模块

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “show number”显示数字指令



这个模块直接可以让我们的micro.bit显示不同数字，当我们输入的数字为个位数时，它是以常亮方式显示在我们的micro.bit上，如果我们输入的为两位数及以上时，它会滚动显示在我们的LED上。

2. “pause”延迟命令

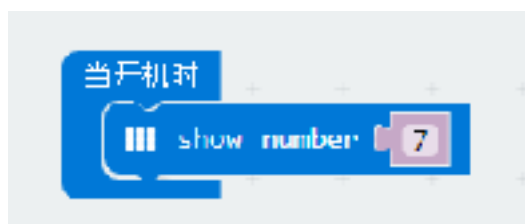


在模块里输入不同数字就是我们上一个指令需要延迟的时间，ms - 要暂停的毫秒数（100 = 1/10秒，1000毫秒= 1秒）。

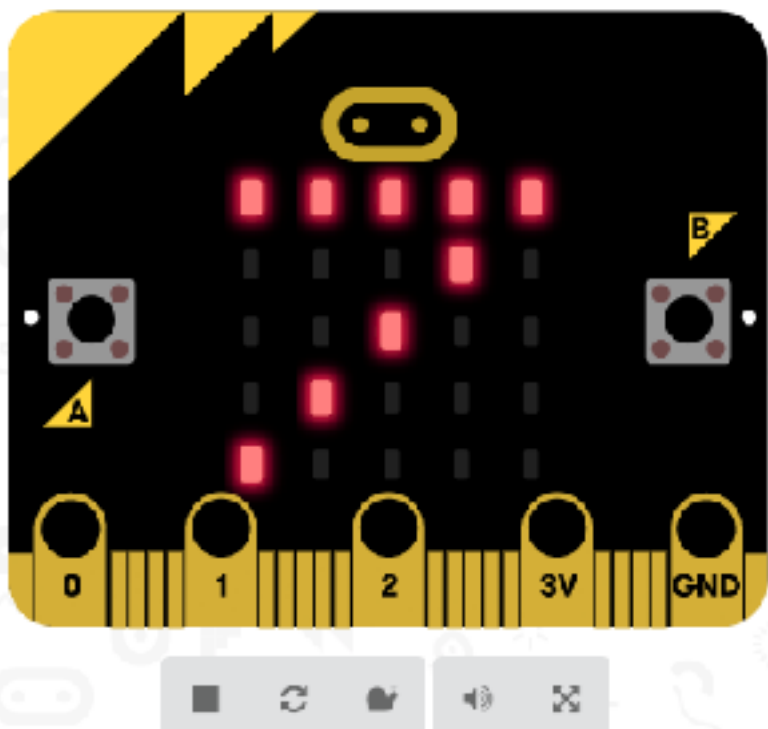
【动手实践】

任务描述1：让我们的micro.bit显示数字7

参考程序



上传程序后，我们可以在模拟窗口看的读数



任务描述2：让我们的micro.bit轮流显示1、2、3

参考程序



上传程序后，我们可以在模拟窗口看的读数

同学们也可以把写好的程序上传给micro.bit看看谁做的更酷炫。

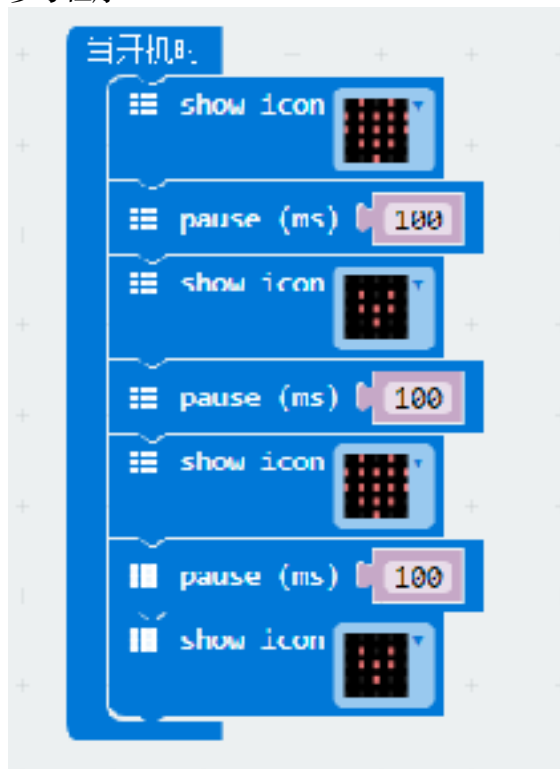
【探究思考】

如果我们想让从1显示到10，有几种办法？

【挑战自我】

同学们试着做一个心跳的图案，要有心跳的动作。

参考程序

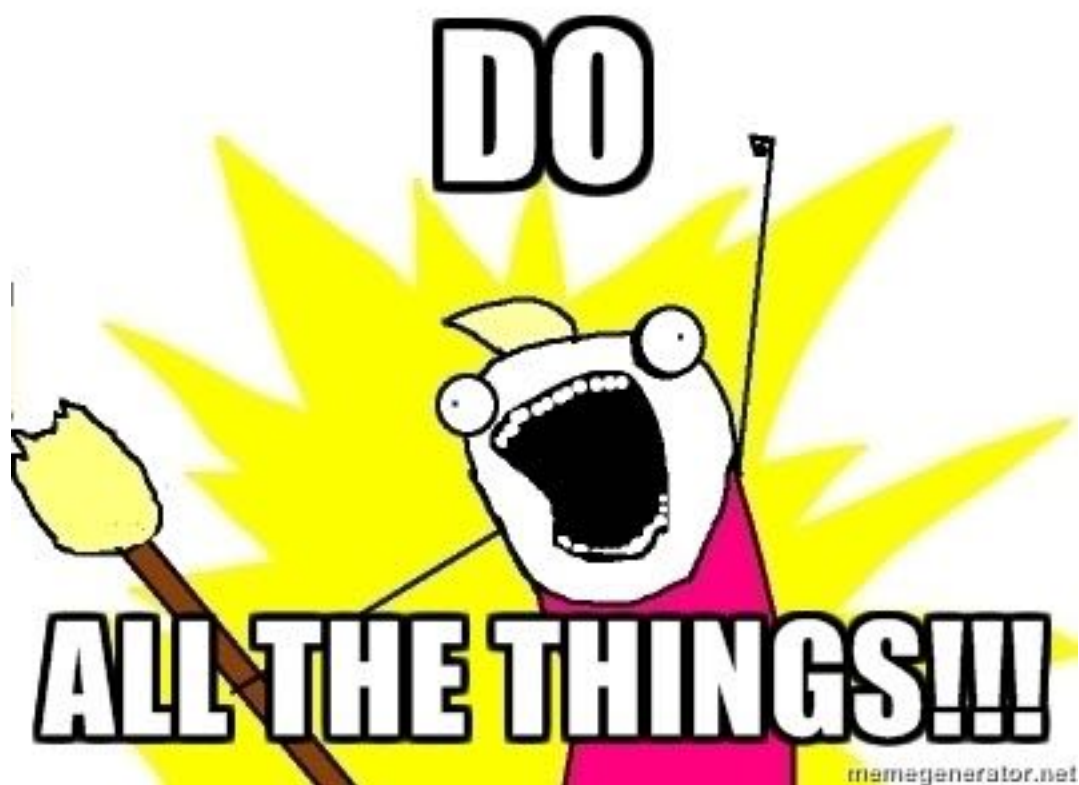


【扩展阅读】

什么是编程？

尽可能简单的讲，编程是编写代码，命令计算机去完成某项任务的艺术。这里讲的某项任务，可以是简单的两数相加，或是像把飞船送入轨道这样的复杂任务！

一个程序里面，最小的组成部分被称作语句（statement）——代表了对计算机做出的一条指令。



当你完成了自己的程序后，编译器会把你写的代码转换为机器码——计算机语言的最底层。机器码指示中央处理器工作（central processing unit），或者叫做CPU，这里面包含一些诸如加载某个值或是做数学运算的步骤。

第三课：屏幕橡皮擦

平时我们上课的时候老师写错字都要用黑板擦来把错误的内容擦掉，今天我们就介绍一下我们micro.bit里的屏幕橡皮擦，它可以很轻松的把显示屏上的内容擦掉，我们可以用按钮来当擦掉屏幕的指令

【任务目标】

了解如何清除屏幕，“clean screen”清除屏幕。我们将学习如何使用“on button A pressed do”控制我们两个按钮。

【知识点】

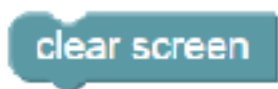
1. 学习“clean screen”清楚屏幕
2. 学习使用“on button A pressed do”模块来控制按钮

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “clean screen”显示数字指令



当这个模块执行时，我们的屏幕上的信息都会被清除

2. “on button A pressed do”按钮控制命令

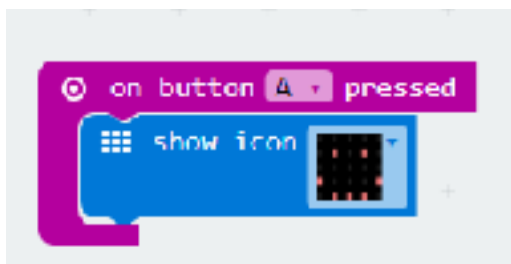


这个模块做简单的理解就是当按钮“A被按下，我们就执行程序里边的内容”

【动手实践】

任务描述1：让我们的micro.bit当按下“A”按钮时显示一个笑脸

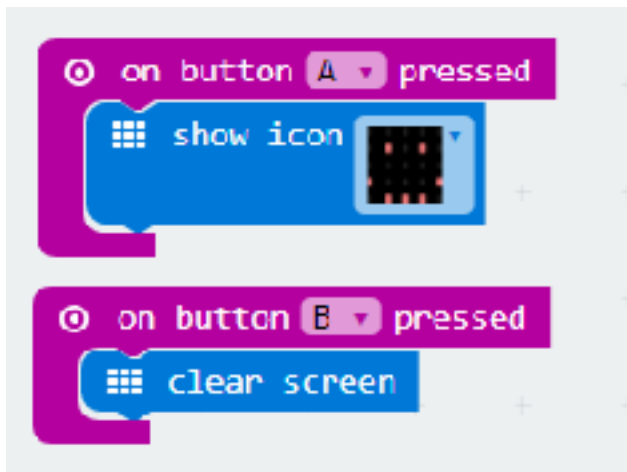
参考程序



上传程序后，我们可以在模拟窗口试验一下，如果可以成功就上传到我们的主控器。

任务描述2：让我们的micro:bit按下A显示笑脸，按下B清除屏幕。

参考程序



上传程序后，我们可以试试是不是我们程序写的那样

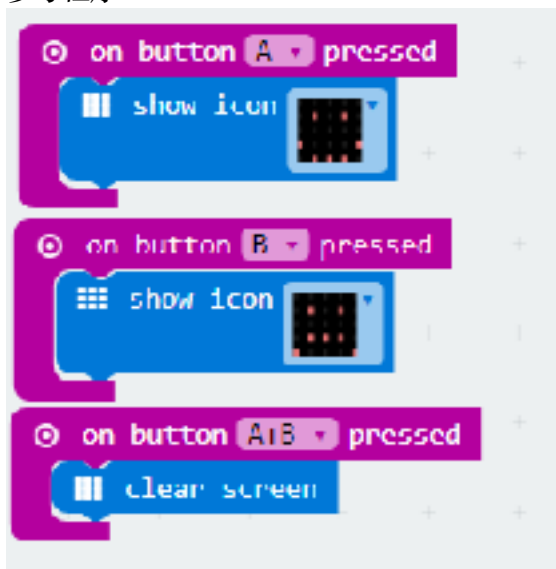
【探究思考】

如果我们想让同时按下A和B出现另外一个图案需要怎么操作？

【挑战自我】

同学们试着做一个按下A出现笑脸，按下B出现哭脸，同时按下A和B清除屏幕。

参考程序



【扩展阅读】

按钮消抖

按键消抖通常的按键所用开关为机械弹性开关，当机械触点断开、闭合时，由于机械触点的弹性作用，一个按键开关在闭合时不会马上稳定地接通，在断开时也不会一下子断开。因而在闭合及断开的瞬间均伴随有一连串的抖动，为了不产生这种现象

而作的措施就是按键消抖。

我们常用软件方法去抖，即检测出键闭合后执行一个延时程序，5ms~10ms的延时，让前沿抖动消失后再一次检测键的状态，如果仍保持闭合状态电平，则确认为真正有键按下。当检测到按键释放后，也要给5ms~10ms的延时，待后沿抖动消失后才能转入该键的处理程序。

一般来说，软件消抖的方法是不断检测按键值，直到按键值稳定。实现方法：假设未按键时输入1，按键后输入为0，抖动时不定。可以做以下检测：检测到按键输入为0之后，延时5ms~10ms，再次检测，如果按键还为0，那么就认为有按键输入。延时的5ms~10ms恰好避开了抖动期。

第四课：聪明的micro:bit

在我们的平时快要摔倒时，我们本能反应就可以让我们站立起来。在机器世界里机器人可以靠三轴陀螺仪来感知机子的身体姿态，今天我们就教大家如何使用三轴陀螺仪来判断我们的micro:bit的姿态，并且显示在屏幕上。

【任务目标】

当我们反转屏幕，在屏幕上显示我们的反转方向。

【知识点】

1. 学习“on logo up”、“on logo down”等来感知模块方向
2. 学习使用“on shake”模块感知micro:bit摇晃

【材料清单】

Micro:bit主控板、数据线、Makecode在线网站

【知识储备】

1. “on logo up”显示数字指令



当我们模块向上反转时，执行里边的程序

2. “on shake”摇晃检测命令



当我们的主控板摇晃时，执行里边的程序

3. “show arrow”方向显示命令



这里我们用“东南西北”来显示各种方向

【动手实践】

任务描述：让我们的micro:bit显示出反转方向，当摇晃时，清除屏幕

参考程序



上传程序后，我们可以在模拟窗口试验一下，如果可以成功就上传到我们的主控器。

任务描述2：让我们的micro:bit 自己能指示自己倾斜方向，并且当摇晃它时自己清屏。

参考程序



上传程序后，我们可以试试是不是我们程序写的那样

【探究思考】

我们可不可以试着做一个指南针？

【挑战自我】

试着用这个我们做一个计步器。

【扩展阅读】

三轴陀螺仪



三轴陀螺仪是惯性导航系统的核心敏感器件，其测量精度直接影响惯导系统的姿态解算的准确性。因此，如何减小三轴陀螺仪的测量误差，提高其测量精度，就成为了一个至关重要的问题。对于单轴陀螺仪来说，影响其静态测量精度的主要因素是该传感器的零偏误差、刻度系数误差和随机漂移误差；但对于三轴陀螺仪来说，其测量结果的精度与构成三轴陀螺仪的各单轴陀螺仪的零偏误差、刻度系数误差、随机漂移误差以及各单轴陀螺仪敏感轴之间的不正交安装误差相关。相比于单轴传感器，三轴传感器的校准参数更多，校准过程更为复杂。目前，陀螺仪的标定通常采用位置标定和速率标定方法，这2种方法具有原理简单、易于实现、精度较高等优点，但随着标定参数的增加，数据量剧增，耗时，且测试条件比较苛刻，需要高精度的测试设备，标定结果取决于测试设备的精度；此外，有采用系统级的标定方法，利用惯性仪表的输出直接进行导航解算，利用导航解算误差作为量测量来估算陀螺误差参数，这种方法不需要精密的测试设备，通常采用滤波算法对误差进行参数估计，但计算量大，可观测性分析复杂，标定时间较长。因此，本文提出了一种基于椭球拟合的三轴陀螺仪的快速校准方法。首先对三轴陀螺仪的制造误差进行全面的理论分析，建立相应的数学模型，然后根据椭球拟合算法，对包含制造误差的三轴测量数据进行椭球拟合，并对陀螺仪的制造误差进行参数标定与补偿。

第五课：智能夜灯

当今世界中，我们的生活越来越智能。很多传统开关都由智能硬件所代替。很多同学可能好奇为啥路灯在天黑的时候会自动打开，在天亮的时候自动熄灭？今天我们就在这里教大家如何使用micro:bit自带的光线传感器来制作一个智能夜灯，让它也能像路灯一样在光线暗的时候它变亮，当光线亮的时候灯可以自动变暗。

一.知识储备

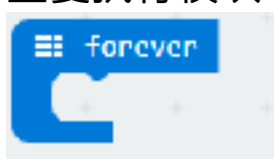
3. 光线传感器



MicroBit的光线传感器是内置在主控器里的，的数值范围我们可以通过程序来测得，数字高说明光线越亮，数字越低光线越暗。

通过图二的程序测得我们的光线传感器量程范围是0~255。当光线越亮的时候数字越高。

4. 重复执行模块



在模块内的程序会一直重复执行。

5. “如果”、“则”模块

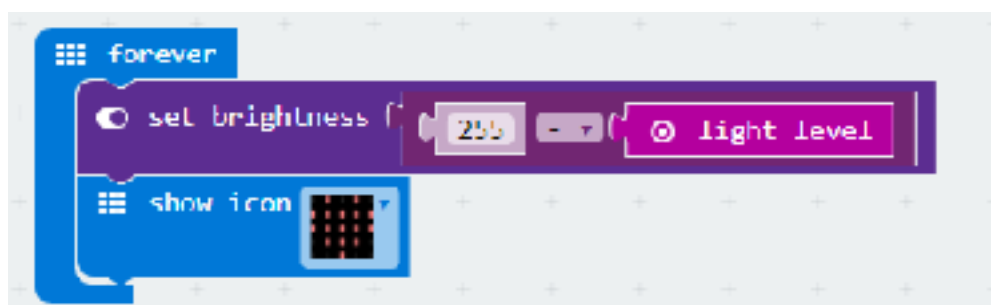


当如果后边的内容满足时，执行程序内部的内容。

【动手实践】

任务描述1：让我们的micro.bit显示图案随着光线越亮灯越暗

参考程序

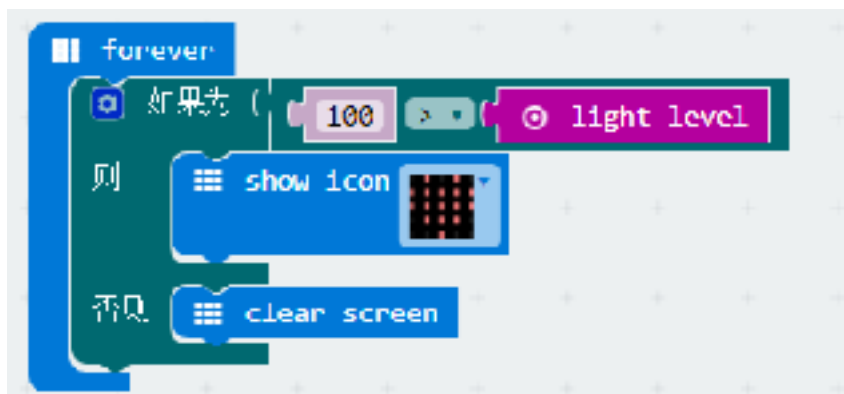


上传程序后，我们可以试验一下，当有光照时，LED灯变暗，当没有光照时，LED灯变亮

很多同学可能会问为什么要用255减去现在的光感数值？

任务描述2：当光线小于100时，灯亮，当光线大于100时，灯灭。

参考程序



上传程序后，试着把主控器挡住，看看灯会不会亮

【探究思考】

我们是不是可以用读出光线传感器的方法来读其他传感器读数？

【挑战自我】

试着做一个边显示当前光线传感器读值，亮度也随着光线传感器读值变化。

【扩展阅读】

光线传感器原理

光线感应器是由两个组件即投光器及受光器所组成，利用投光器将光线由透镜将之聚焦，经传输而至受光器之透镜，再至接收感应器，接收感应器将收到之光线讯号转变成电信号，此电信讯号更可进一步作各种不同的开关及控制动作，其基本原理即对投光器受光器间之光线做遮蔽之动作所获得的信号加以运用以完成各种自动化控制。

第六课：温度计

如今人们越来越关注自己的生活环境质量，家里也几乎都会准备一个温度计来看家里实时温度，今天我们就教大家如何使用micro:bit来做一个温度计，他可以实时显示当前温度，并且在温度过于高或者过于低的时候报警。

【任务目标】

制作一个温度计，当温度高于30度显示HOT，当温度低于20度显示COLD，当温度在20度到30度之间显示实时温度。

【知识点】

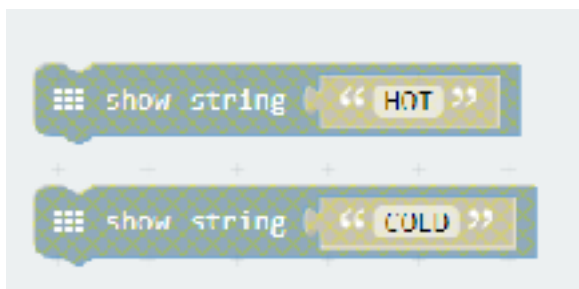
1. 学习“show string”来显示字符
2. 学习使用“temperature”模块感知环境温度

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “show string”显示字符指令



我们可以自定义字符让我们的micro:bit来显示（只能是英文）

2. “temperature”温度检测命令



温度传感器模块。它直接显示当前环境温度，显示单位为摄氏度。

3. “与”“或”“非”

“且”运算，也就是我们计算机里讲到的“与”运算。只有满足左右两项条件输出值才为真，其他都为假。

“或”运算，只要满足其中一项为真。

“非”运算，也就是我们的“（ ） 不成立”，判断（ ）不成立为真。

基本公式如下：

注：此处“+”表示“或”，“·”表示“与”，“'”表示“非”

(1) $0 \cdot A = 0$ (10) $1' = 0$; $0' = 1$

(2) $1 \cdot A = A$ (11) $1 + A = 1$

(3) $A \cdot A = A$ (12) $0 + A = A$

(4) $A \cdot A' = 0$ (13) $A + A = A$

- (5) $A \cdot B = B \cdot A$ (14) $A + A' = 1$
 (6) $A \cdot (B \cdot C) = (A \cdot B) \cdot C$ (15) $A + B = B + A$
 (7) $A \cdot (B + C) = A \cdot B + A \cdot C$ (16) $A + (B \cdot C) = (A + B) \cdot C$
 (8) $(A \cdot B)' = A' + B'$ (17) $A + B \cdot C = (A + B) \cdot (A + C)$
 (9) $(A')' = A$ (18) $(A + B)' = A' \cdot B'$

【动手实践】

任务描述：让我们的micro.bit显示出当前环境温度

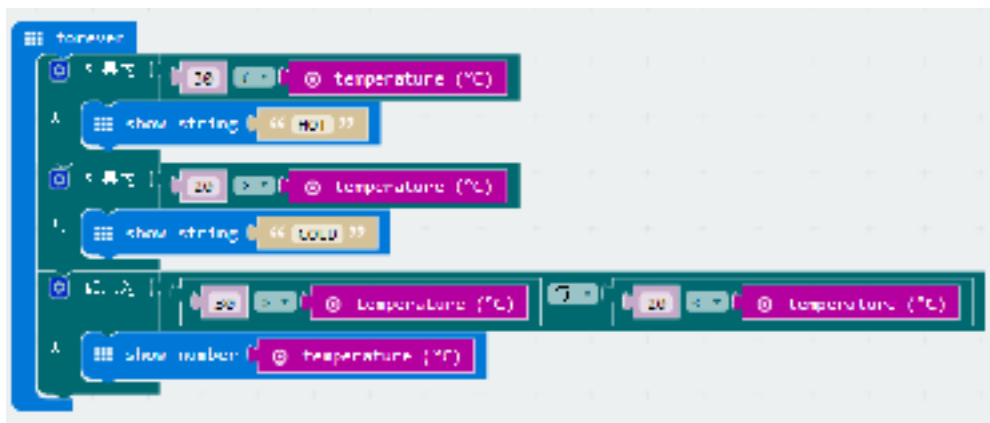
参考程序



此时我们要注意到“show number”和“show string”的区别

任务描述2：让我们的micro.bit 自己能指示当前温度，当温度高于30度显示HOT，当温度低于20度显示COLD，当温度在20度到30度之间显示实时温度

参考程序



上传程序后，我们可以试试是不是我们程序写的那样

【探究思考】

为什么我们用的是“与”运算？用其他可以吗？

【挑战自我】

试着用雪花和太阳代替我们的英文单词。

【扩展阅读】

温度传感器

温度传感器（temperature transducer）是指能感受温度并转换成可用输出信号的传感器。温度传感器是温度测量仪表的核心部分，品种繁多。按测量方式可分为接触式和非接触式两大类，按照传感器材料及电子元件特性分为热电阻和热电偶两类。



第七课：指南针

在我们到了一个陌生的地方或者是荒郊野外时，我们没有办法辨别方向，这个时候就需要用的户外必备的一个工具：指南针。指南针可以永远指着北方，告诉我们前进的方向。指南针又称司南，主要组成部分是一根装在轴上的磁针，磁针在天然地磁场的作用下可以自由转动并保持在磁子午线的切线方向上，磁针的北极指向地理的北极，利用这一性能可以辨别方向。

常用于航海、大地测量、旅行及军事等方面。物理上指示方向的指南针的发明由三部件组成，分别是司南、罗盘和磁针，均属于中国的发明。据《古矿录》记载最早出现于战国时期的磁山一带。

今天我们就教大家如何使用micro:bit来制作指南针。

【任务目标】

制作一个指南针，用“N”“S”“W”“E”来指示方向

【知识点】

1. 学习“show string”来显示字符
2. 学习使用“temperature”模块感知环境温度

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “compass heading”读取方向指令



我们的micro: bit有一个内置的磁力计，所以它可以使你的方向相对于北磁极。

2. 将“item”设定为 ()



变量模块，其中“item”可以更改。变量就相当于我们数学中学到的“X”，它可以随意改变。使用这个模块我们可以把变量定义为某一个数。

3. “show string”显示字符串命令



让可以在LED屏幕上一次显示一个字符（从左到右滚动）。

【动手实践】

任务描述：让我们的micro.bit显示出所指向的方向

参考程序

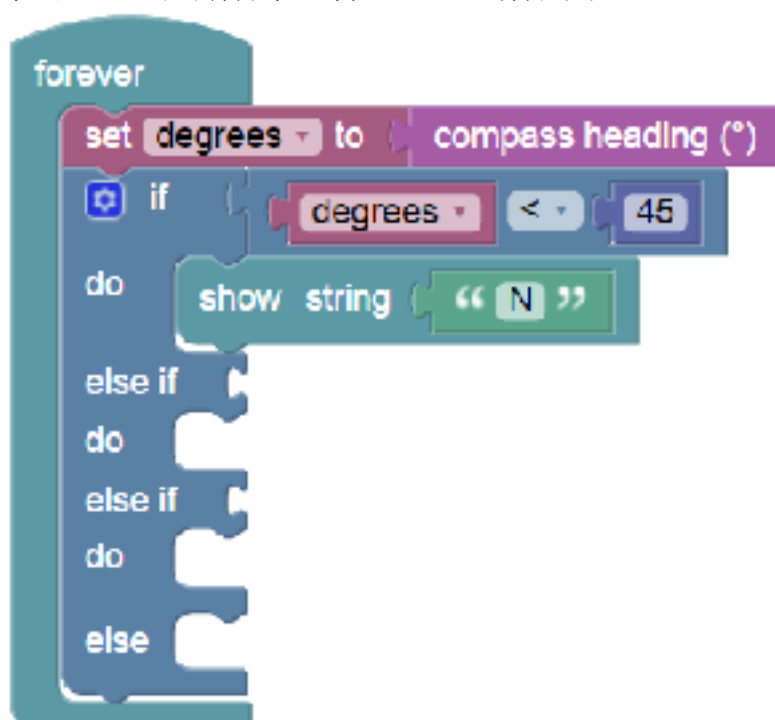
创建一个循环，将不断更新指南针的读数。



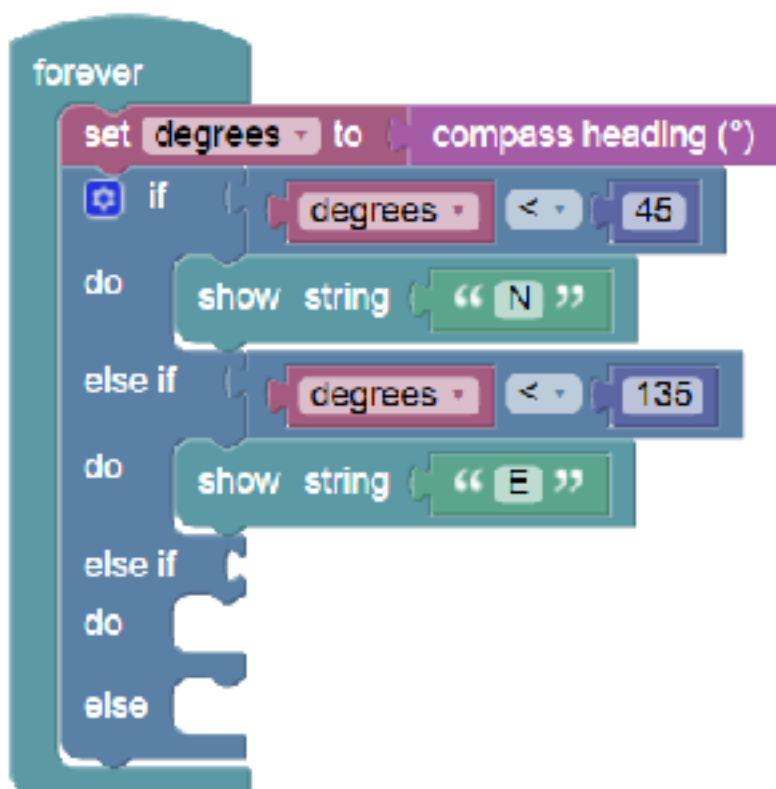
将micro: bit的读取存储在一个变量中degrees。



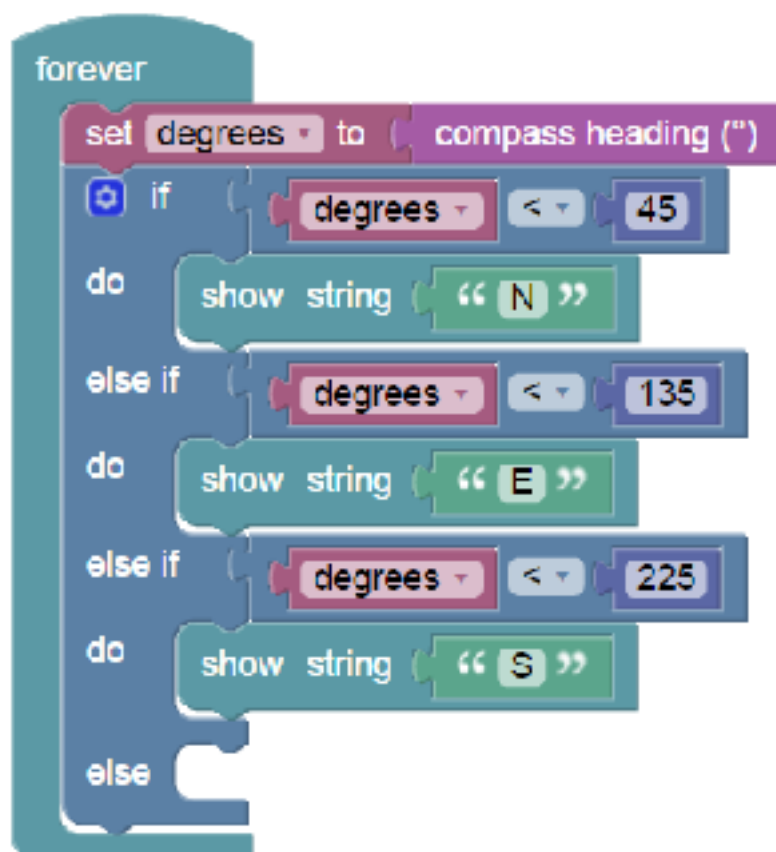
如果degrees小于45，则指南针标题主要指向北。显示N



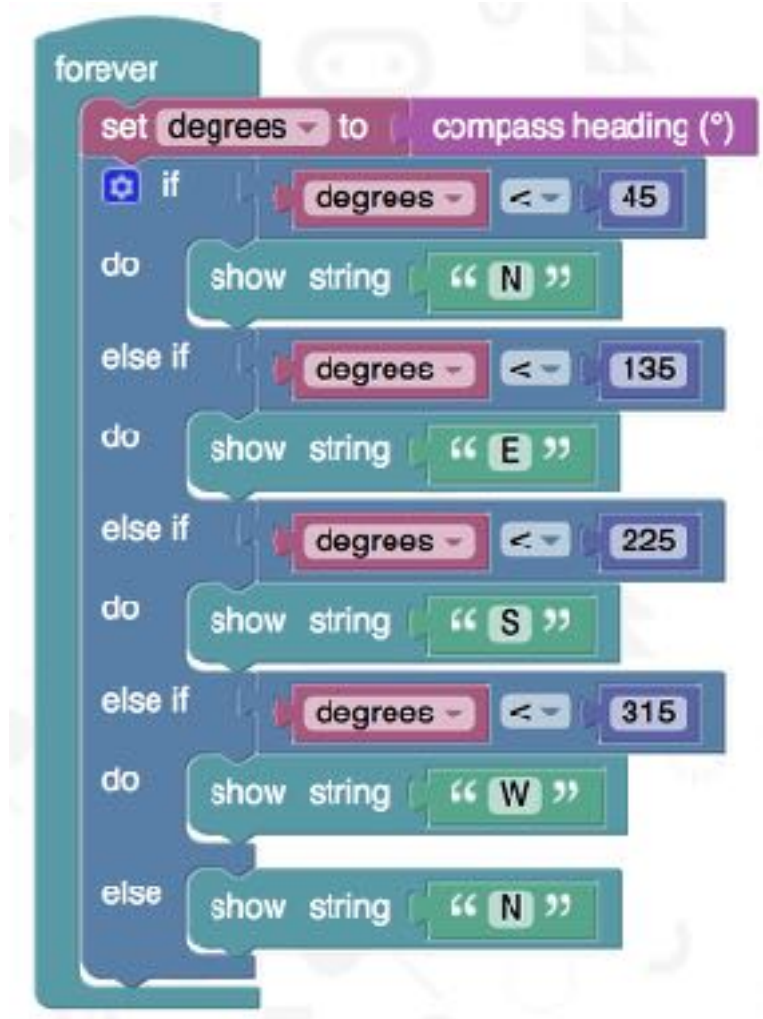
如果degrees小于135，则微：位大部分指向东。显示E



如果degrees小于225，则微：大多数指南南。显示S



如果没有这些条件返回true，那么必须指向West。显示W



【探究思考】

我们所测得角度需不需要初始化？

【挑战自我】

试着使用方向箭头代表方向，箭头始终指向北方。

【扩展阅读】

指南针的发明

指南针的发明应当是在一个很漫长的时间中，慢慢地改进的结果，而不同时期的形式，应以不同的形式出现。唐代堪舆家的活动相当活跃，并开始强调方向的选择，寻找比磁勺更方便的指向器成了当务之急。于是指南铁鱼或者蝌蚪形铁质指向器及水浮磁针应运而生。

活动于唐开元年间（713—741年）的山西堪舆家丘延瀚，被后世堪舆家推崇为堪舆术三针（正针、缝针和中针）中最早出现的正针法的创始人。明代雅好科技的宁献王朱权则在其《神机秘籍》中说：“针法古无所传，自昔玄真始制。”玄真即唐代浙江金华道家张志和（约730—约810年），他的道号叫玄真子，所著《玄真子》十二卷，残存三卷。从这残卷中知道他颇好物理，但

未找到与指南针或针法直接有关的记载。然而，这个道号不禁使人联想到旧题晋崔豹所作的《古今注》中把蝌蚪称为“玄针”。其文曰：“虾蟆子，曰蝌蚪，一曰玄针，一曰玄鱼，形圆而尾大，尾脱即脚生。”十世纪时马缟的《中华古今注》中也有类似的记载。这项记载似乎把磁针与指南鱼的发明和应用从形状和性能上联系到了一起。

第八课：骰子游戏

骰子，古代汉族民间娱乐用来投掷的博具。相传是三国时魏国曹植所造。通常作为桌上游戏的小道具，最常见的骰子是六面骰，它是一颗正立方体，上面分别有一到六个孔（或数字），其相对两面之数字和必为七。中国的骰子习惯在一点和四点漆上红色。骰子是容易制作和取得的随机数产生器。今天我们就自己教大家制作一个数字骰子。

【任务目标】

制作一个骰子，可以每摇一下随机显示数字1-6的点数

【知识点】

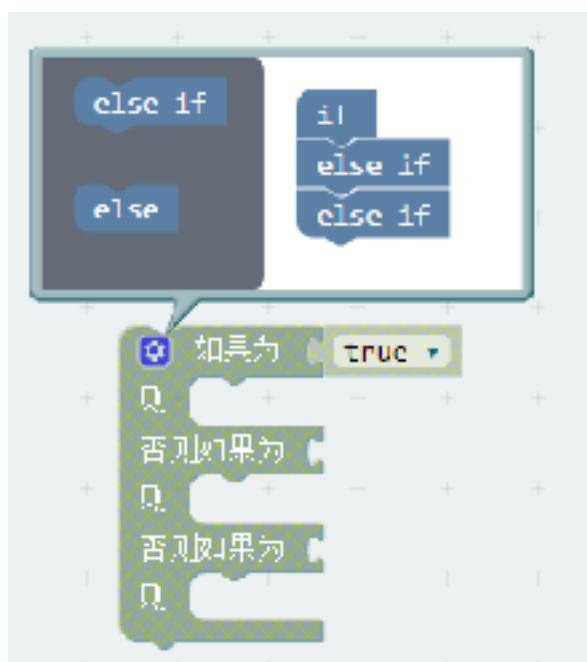
1. 学习“show string”来显示字符
2. 学习使用“temperature”模块感知环境温度

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “如果为（）则”条件判断模块



在之前我们只是用到最简单的如果条件满足，就执行一个条件，今天我们用到的稍微复杂一点，点击模块上的齿轮，就会出现图上所示图框，把“else if”拖入之后，就会出现多个条件判断。

2. 选择0到（）的随机数

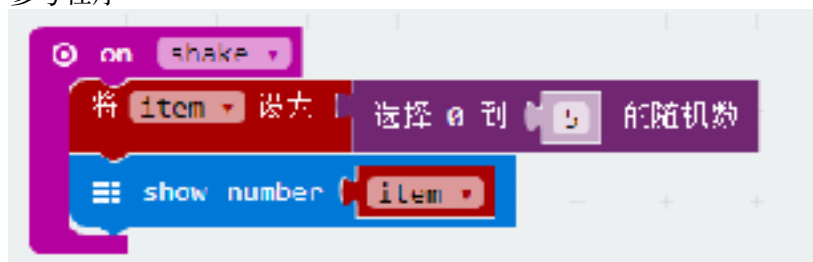


顾名思义，就是从0到（）中选择一个随机数

【动手实践】

任务描述1：让我们的micro:bit显示每摇一下的随机数

参考程序

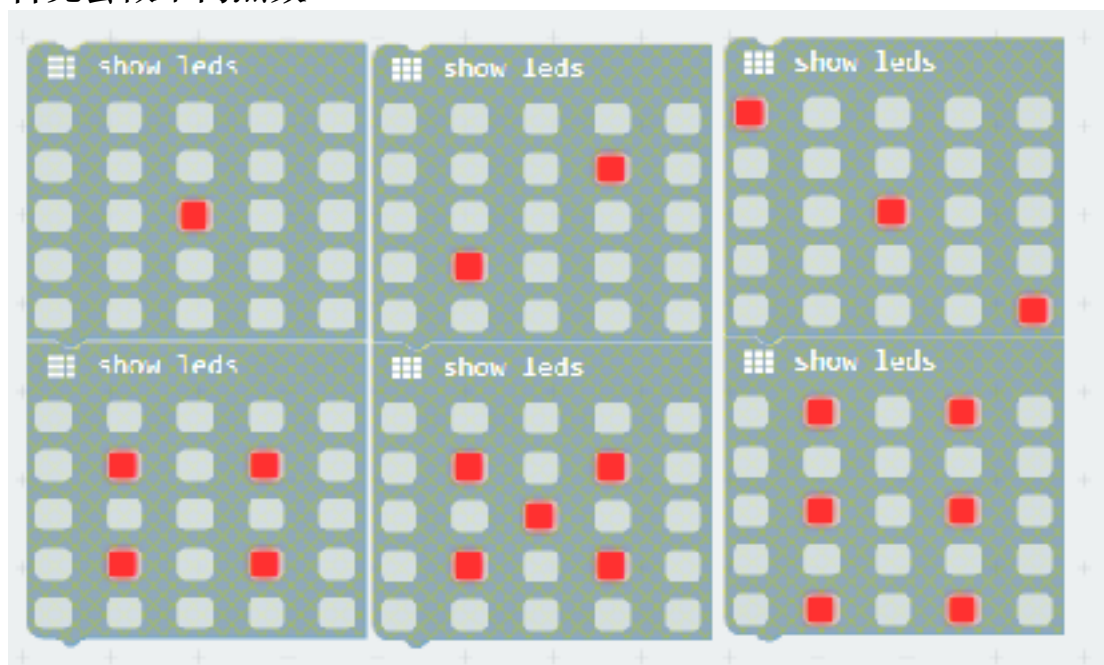


将变量设定为从0到5的随机数，每摇动一下随机yici，然后显示

任务描述2：把摇动的随机数用不同点数显示出来

参考程序

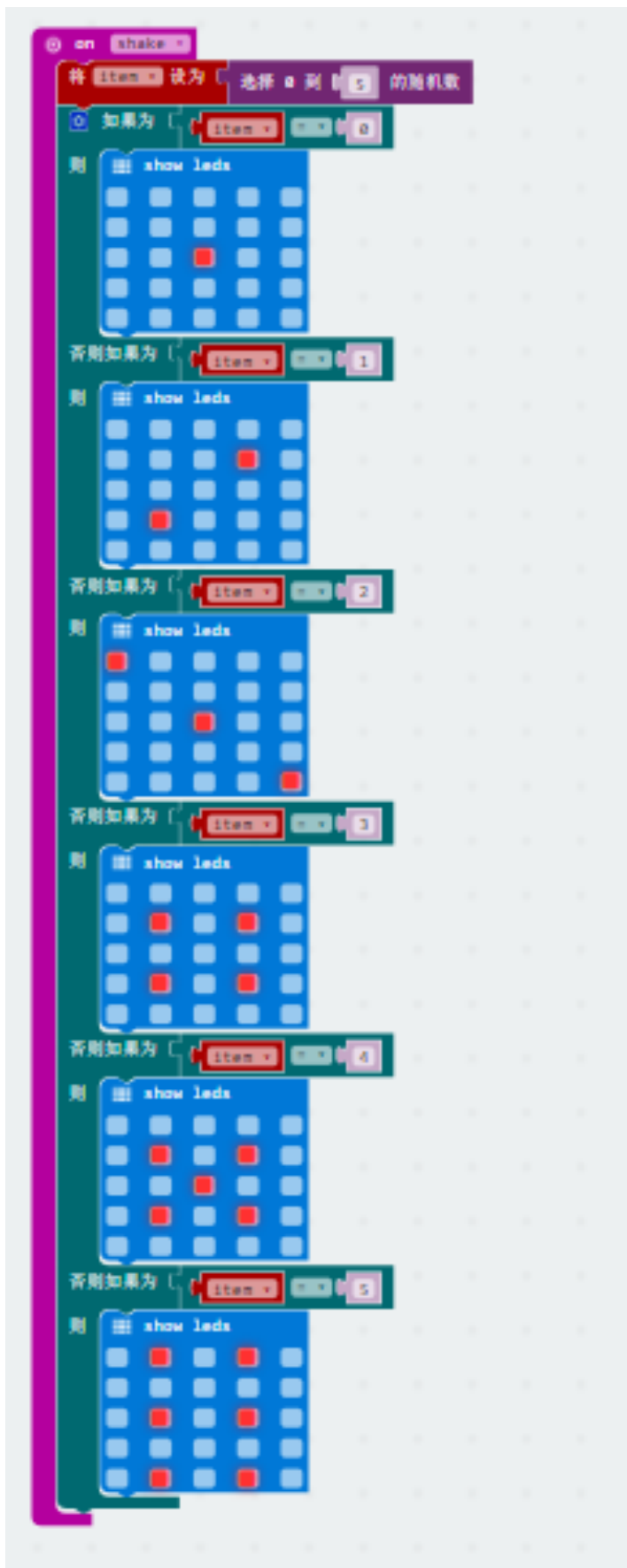
首先会做不同点数



然后用如果做出嵌套程序



最后组合起来，就是我们的程序



将变量设定为从到5的随机数，每摇动一下随机一次，然后显示

【探究思考】

为什么我们点数对应的随机数字每次都相差1

【挑战自我】

试着可以做一个更大的骰子，例如1到10

【扩展阅读】

随机数

随机数是专门的随机试验的结果。在统计学的不同技术中需要使用随机数，比如在从统计总体中抽取有代表性的样本的时候，或者在将实验动物分配到不同的试验组的过程中，或者在进行蒙特卡罗模拟法计算的时候等等。产生随机数有多种不同的方法。这些方法被称为随机数生成器。随机数最重要的特性是它在产生时后面的那个数与前面的那个数毫无关系。

第九课：手表

时间是一个较为抽象的概念，是物质的运动、变化的持续性、顺序性的表现。时间概念包含时刻和时段两个概念。时间是人类用以描述物质运动过程或事件发生过程的一个参数，确定时间，是靠不受外界影响的物质周期变化的规律。例如月球绕地球周期，地球绕太阳周期，地球自转周期，原子震荡周期等。爱因斯坦说时间和空间是人们认知的一种错觉。大爆炸理论认为，宇宙从一个起点处开始，这也是时间的起点。今天我们就教大家如何使用我们的micro:bit来制作一个手表。

【任务目标】

制作一个手表，时、分、秒分别显示。

【知识点】

1. 学习“show string”来显示字符
2. 学习使用“temperature”模块感知环境温度

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. 时钟的进制

1天等于24小时

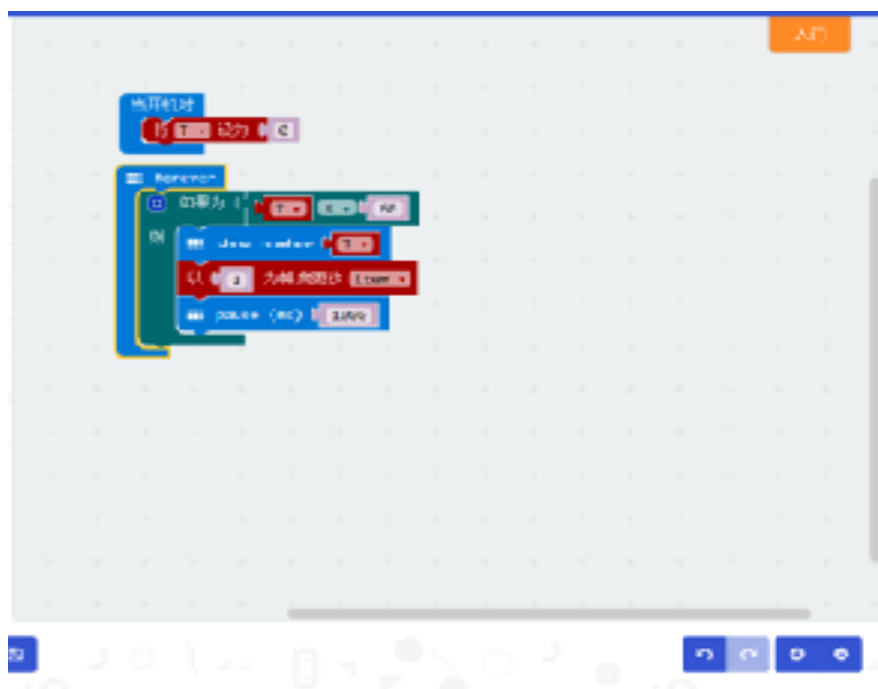
1小时等于60分

1分钟等于60秒

【动手实践】

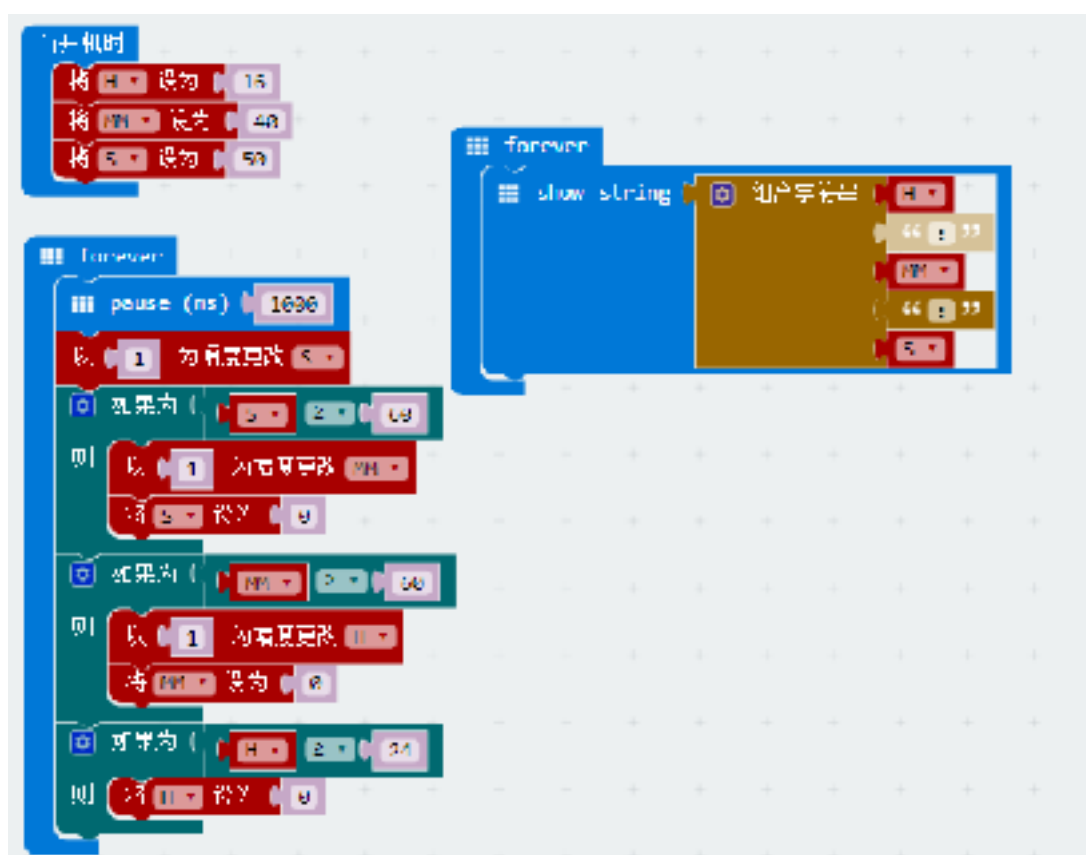
任务描述1：让我们的micro.bit从1数到60，每一秒数一个数

参考程序



任务描述2：让我们的micro.bit当前时间

参考程序



首先在变量初始化当前时间，然后把程序上传。

【探究思考】

我们为什么要用两个forever?

【挑战自我】

试着把时间指示用钟表上的时针、分针来表示。

【扩展阅读】

时区划分

将地球表面按经线划分的24个区域。当我们在上海看到太阳升起时，居住新加坡的人要再过半小时才能看到太阳升起。而远在英国伦敦的居民则还在睡梦中，要再过8小时才能见到太阳呢。世界各地的人们，在生活和工作中如果各自采用当地的时间，对于日常生活、交通等会带来许许多多的不便和困难。为了照顾到各地区的使用方便，又使其他地方的人容易将本地的时间换算到别的地方时间上去。有关国际会议决定将地球表面按经线从东到西，划成一个个区域，并且规定相邻区域的时间相差1小时。在同一区域内的东端和西端的人看到太阳升起的时间最多相差不过1小时。当人们跨过一个区域，就将自己的时钟校正1小时（向西减1小时，向东加1小时），跨过几个区域就加或减几小时。这样使用起来就很方便。现今全球共分为24个时区。由于实用上常常1个国家[6]，或1个省份同时跨着2个或更多时区，为了照顾到行政上的方便，常将1个国家或1个省份划在一起。所以时区并不严格按南北直线来划分，而是按自然条件来划分。例如，中国幅员宽广，差不多跨5个时区，但实际上在只用东八时区的标准时即北京时间为准。

第十课：状态指示器

在之前的课程中，我们做过温度计，指南针，夜灯等等，但是我们做的时候状态都是显示在制作的那个主板上的。很多人都会疑问，在生活中，如果我在家里就想看到室外的温度，那该怎么办？今天我们就教大家如何利用简单的通讯，让我们的micro:bit连到一起，显示出其他小伙伴的状态。

【任务目标】

制作一个远程监控的温度计，通过无线连接，实时显示远方的温度数据。

【知识点】

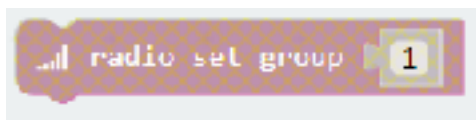
1. 学习“radio set group”设置我们的通信组
2. 学习“radio send number”来通过无线发送数字
3. 学习“on radio received”来接受数字

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “radio set group”设定信号组



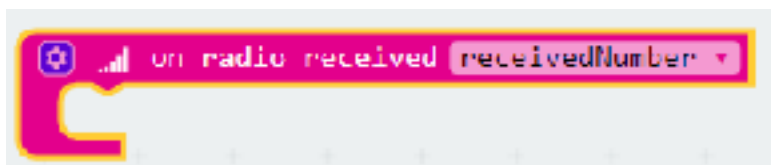
我们这里一共可以有0到255个信号组，每一个信号组就相当于一个班级一样，班级内部才可以通讯，所以我们两块板要在信号组相同的情况下才能发送和接受命令。

2. “radio send number”通过无线发送数字命令



我们的无线可以发送文字和数字，图上所示的命令是发送数字命令。如果我们的变量也是数字，也可以把变量模块拖入。

3. “on radio received”接收信号命令



在我们的接收信号中，我们也分为接受数字和文字的区别，图上所示就是将接收的数字命名为“receivednumber”然后来读取这个变量就可以。

【动手实践】

任务描述：让我们的micro.bit显示出当前环境温度

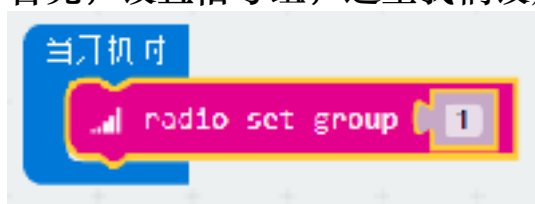
参考程序



此时我们要注意到“show number”和“show string”的区别

任务描述2：让我们可以远程检测到其他micro:bit的温度

首先，设置信号组，这里我们设定为1信号组



然后，将我们温度传感器所感知的温度作为数字发送出去

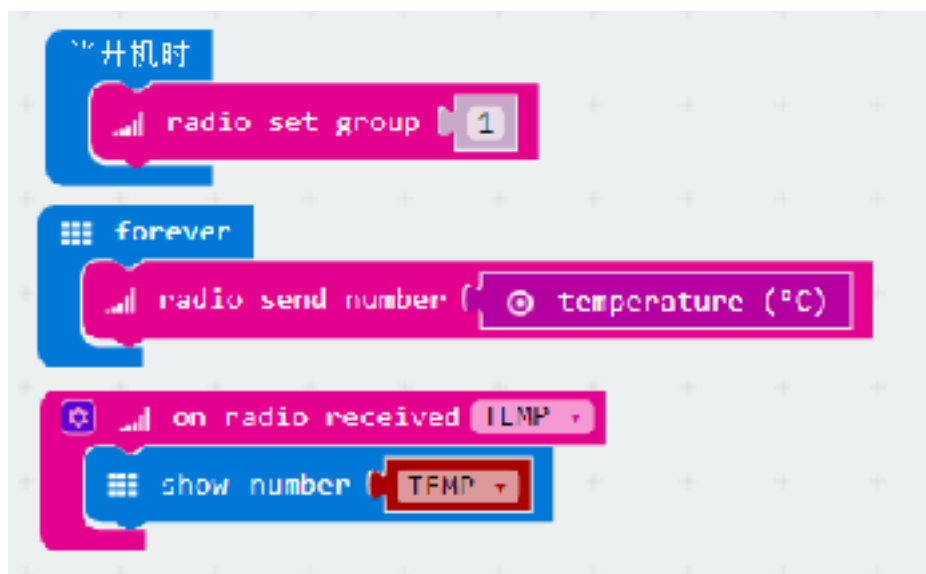


然后需要有一个接收模块，接收我们的温度信号



我们把接收到的数字信号定义为“TEMP”然后直接显示出来。

参考程序



上传程序后，我们可以试试是不是我们程序写的那样

【探究思考】

有没有其他方式读取别的传感器信号？

【挑战自我】

试着在我们显示的温度前加上我们板子的名称。

【扩展阅读】

无线通信

无线通信主要包括微波通信和卫星通信。微波是一种无线电波，它传送的距离一般只有几十千米。但微波的频带很宽，通信容量很大。微波通信每隔几十千米要建一个微波中继站。卫星通信是利用通信卫星作为中继站在地面上两个或多个地球站之间或移动体之间建立微波通信联系。



第十一课：电报机

我们在一些抗战片中经常可以看到电报机，电报机可以将人们想表达的内容传到很远的地方，甚至有的电报机可以加密，将人们说的话转变成另外一些密文来传播，这样可以有效的避免被人监听截取。今天我们就教大家做一个简单的电报机。

【任务目标】

制作电报机，一个用来发送消息，一个用来接收消息。

【知识点】

1. 学习“radio send string”来通过无线发送字符串
2. 学习“on radio received”来接受字符串

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

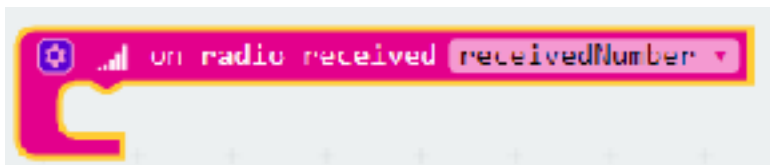
【知识储备】

1. “radio send string”通过无线发送字符串



我们的无线可以发送文字和数字，图上所示的命令是发送数字命令。如果我们的变量也是数字，也可以把变量模块拖入。

2. “on radio received”接收信号命令

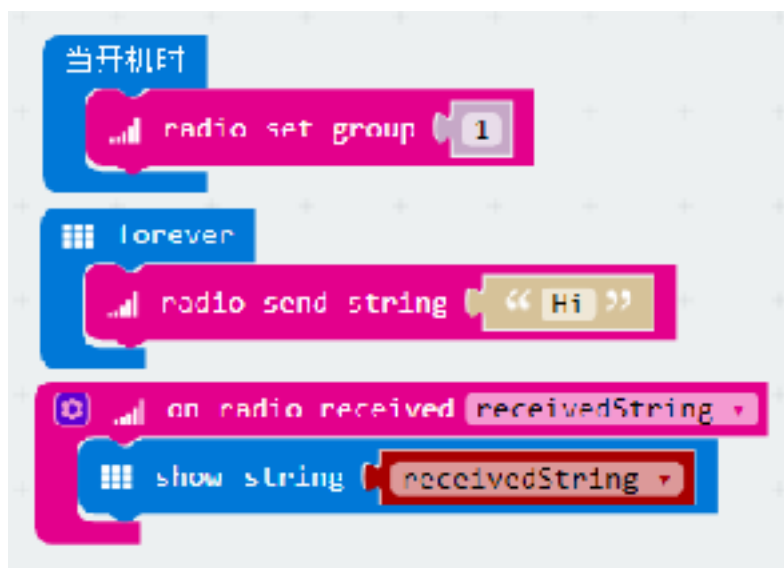


使用这个命令来接收别的信号发来的字符串

【动手实践】

任务描述1：让我们的micro.bit一个发送一个接收信号

参考程序



任务描述2：让我们的micro:bit按下A选择发送内容，按下B确定发送
参考程序



按A选择数字，按B发送，两个之间可以相互发送接收

【探究思考】

这个变量还可以以什么方式变更

【挑战自我】

试着把我们发送的内容改为字符串。

【扩展阅读】

电报机

在1832年，俄国外交家希林制作出了用电流计指针偏转来接收信息的机器。1835年美国画家莫尔斯经过3年的钻研之后，成功地用电流的“通断”和“长短”来代替了人类的文字进行传送，这就是鼎鼎大名的莫尔斯电码，第一台电报机问世。1837年6月英国青年库克获得了第一个电报发明专利权。

用电来实现远距离高速传递信息的技术可以追溯到18世纪70年代。1774年，一个瑞典发明家发明了一种装置，在装置中每根电线代表字母表中的一个字母。当电流从代表某个字母的电线流过，它会给与之相连的一个小球充电，而后者随后会敲响一个小铃，发出与这个字母相应的音符。这个装置当然很难在实践中得到真正的应用，直到7年之后，一个叫萨缪尔·芬利·莫尔斯的美国失意画家才真正创造出能够实际应用的高效的信息传递系统。莫尔斯把已有的所有零散发明组合了起来，从而发明了电报，莫尔斯唯一的原创就是发明了高效的代码，但是他也花费了好几年时间才能说服政府出资在华盛顿和巴尔的摩之间进行演示。莫尔斯在华盛顿通过电报把“上帝创造了什么”的信息发送给了他在巴尔的摩的同伴阿尔弗雷德·威尔，威尔随即将同样的信息反馈给他——电报获得成功，就立刻像蜘蛛网一样在全美国扩散。就在那一年，莫尔斯和他的合作伙伴成立了电磁电报公司（Magnetic Telegraph Company）来经营纽约和费城之间的电报线路。到1846年，公司已经开始盈利并分红了。10年之内，总长为23000英里电报线连接美国主要大城市。

第十二课：计分器

随着时代的发展，文明在进步，在生活的竞争也在日益的增加，还有许许多多体现人类社会文化进步的文明比赛，为了这些竞争或比赛的公平、公正和明朗化，随着时代的改变，人类发明创造了强大计分器。抢答记分器顾名思义就是在相同的时间内，参与比赛的人谁先回答出且回答正确主持人的问题谁就得分，那么究竟是谁先拥有这样的机会呢？

【任务目标】

制作计分器，回答正确加一分，回答错误减一分

【知识点】

1. 学习使用“A”“B”按钮来结合变量增加或减少来计算积分

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

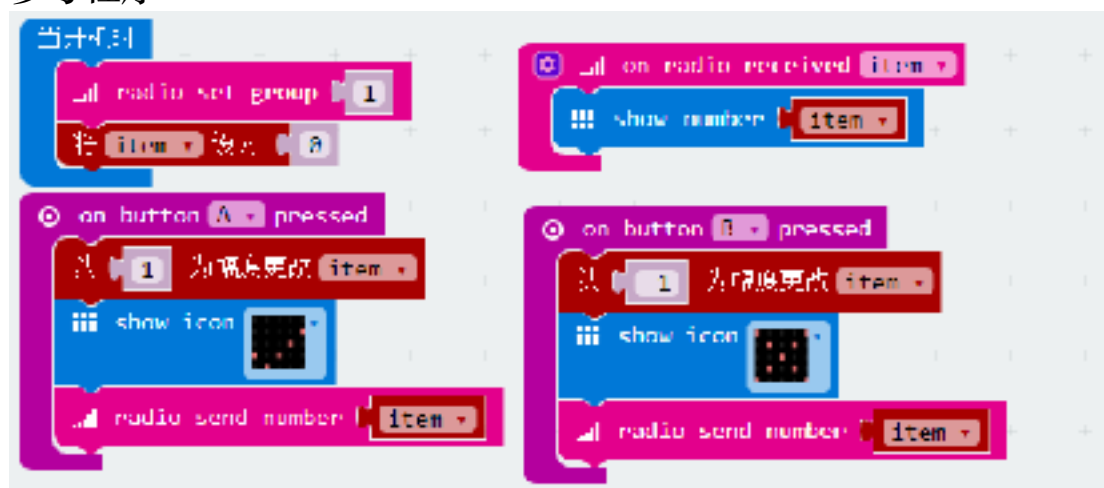
【知识储备】

因为不同场合计分根据的规则不同，例如有的考试答题正确一道题加3分，错误一道题扣1分。不同比赛计分规则不同，所以我们在制作计分器之前需要了解计分规则，这样才能制作出适合我们使用的计分器。

【动手实践】

任务描述：简单按键控制加减分并显示程序

参考程序：



这个程序里我们首先将信号组设定为1，然后设定接收信号接收变量数字，然后按键A和B各自为增加和减少变量，然后分别两边显示正确和错误符号。

【探究思考】

如果加分规则是对一道加5分，错一道扣3分，需要怎么改？

【挑战自我】

可不可以让两个人抢答，用时少者加分。

【扩展阅读】

抢答器

在知识竞赛、文体娱乐活动（抢答赛活动）中，能准确、公正、直观地判断出抢答者的座位号。

传统抢答器只是大概判断出抢答成功或犯规选手台号，无法显示出每个选手的抢答时间。而今抢答器可以通过数据来说明裁决结果的准确性、公平性。使比赛大大增加了娱乐性的同时，也更加公平、公正。

第十三课：智能闹钟

我们每天早晨都伴随着闹钟叫醒，很多同学可能第一反应就是赶快把闹钟关掉。今天我们就教大家做一个智能闹钟，在天亮的时候闹钟开始响，并且闹钟必须按特定次数键的时候才可以停止响铃。现在就让我们开始做吧！

【任务目标】

制作一个闹钟，开关由光线传感器控制，当响铃时，只有按特定次数按键才可以关闭响铃。如果没有蜂鸣器我们可以用自带的模拟界面来体验程序。

【知识点】

1. 学习“play tone”来让我们的micro:bit播放声音
2. 学习使用“light level”模块感知环境光线

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

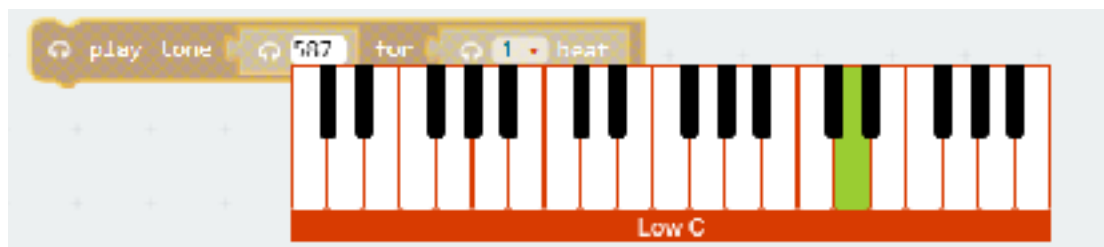
1. 蜂鸣器

蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器主要分为压电式蜂鸣器和电磁式蜂鸣器两种类型。蜂鸣器在电路中用字母“H”或“HA”（旧标准用“FM”、“ZZG”、“LB”、“JD”等）表示。

2. “play tone”模块



如上图所示，左边选择音调我们可以根据弹出的钢琴键盘模拟，如下图



右边选择节拍，大家可以尝试听声音。

【动手实践】

任务描述1：让我们的micro.bit当光线变亮时，自动发出声音

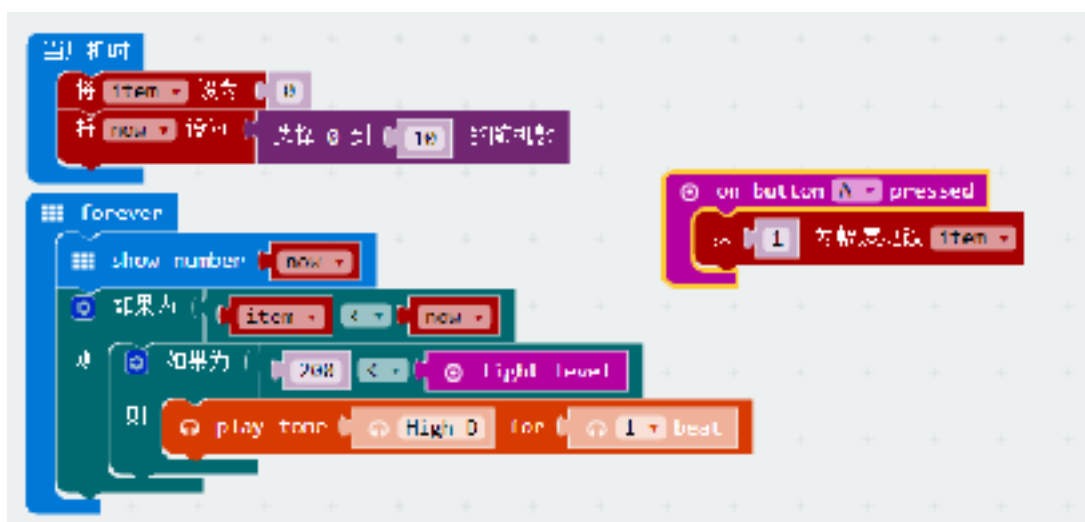
参考程序



当光线大于200时，自动发出声音

任务描述2：让我们的micro:bit只有按下随机次数的A键时，闹钟才停止响铃

参考程序



首先将按键次数变量设为0，然后将now指定一个随机数，将now的随机数显示出来，只有当按下次数大于随机数时，闹钟才跳出循环，停止响铃。

【探究思考】

可不可以用一个变量来实现程序？

【挑战自我】

试着把时间指示也显示出来。

【扩展阅读】

蜂鸣器发声基本知识

用蜂鸣器机播放音乐，或者弹奏电子琴，实际上是按照特定的频率，输出一连串的方波。为了输出合适的方波，首先应该知道音符与频率的关系。

1. 音名

从常见的电子琴的键盘谈起。

在下表中，可以看到一系列黑白相间的琴键（示意）。主要分成低音、中音和高音三个区域，每个区域都有12个琴键。其中的白键，简谱音符标为1、2、3、4、5、6、7，大家一般都读成哆、来、咪、发、嗦、拉、西。

2. 频率

注意看一下几个6（拉）的频率，它们是整数，容易看出规律——是成2倍的关系。其它的音符，也有同样的规律。这些频率，如220、440等，它们在琴键上的位置是世界统一的，无论是钢琴、手风琴，还是电子琴，都是一样的。

包括黑键和白键在内的全部音符的频率数值，是成“等比数列”的关系，它们之间有个公比，可以按照“2倍”的规律推算出来。

第十四课：探究加速度（物理体验课）

牛顿说：物体丢出到落地过程中，若不考虑空气阻力，物体将处于失重状态，当我们平时把物体抛出去的时候，基本上轨迹也会是一条抛物线。我们今天不讨论物体轨迹，只研究物体加速度。加速度也就是物体速度变化的速率，我们就用micro:bit的加速度传感器来测量抛出去一个物体加速度的变化，然后统计下来。

【任务目标】

制作一个加速度仪，可以读出当前加速度，并且通过串口输出

【知识点】

1. 学习“plot bar graph of”来让我们的micro:bit将传感器数值图形化
2. 学习使用“serial write number”读取串口数据

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. “plot bar graph of”图形化绘制模块



我们可以通过这个模块将一些变量用图形化绘制出来，显示在我们的LED屏幕上，上边插入要读取的内容，下边记录量程。

2. “serial write number”串口写入模块

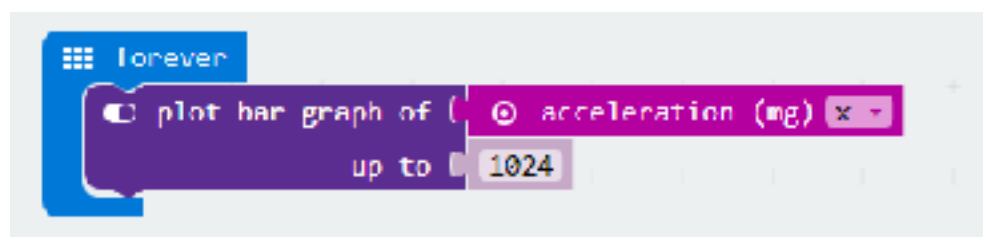


我们可以通过串口数据写入到串口里，然后通过usb线读取

【动手实践】

任务描述1：将加速度图形化

参考程序



我们的加速度传感器量程是0到1024，通过这个程序我们可以在LED显示出加速度图形

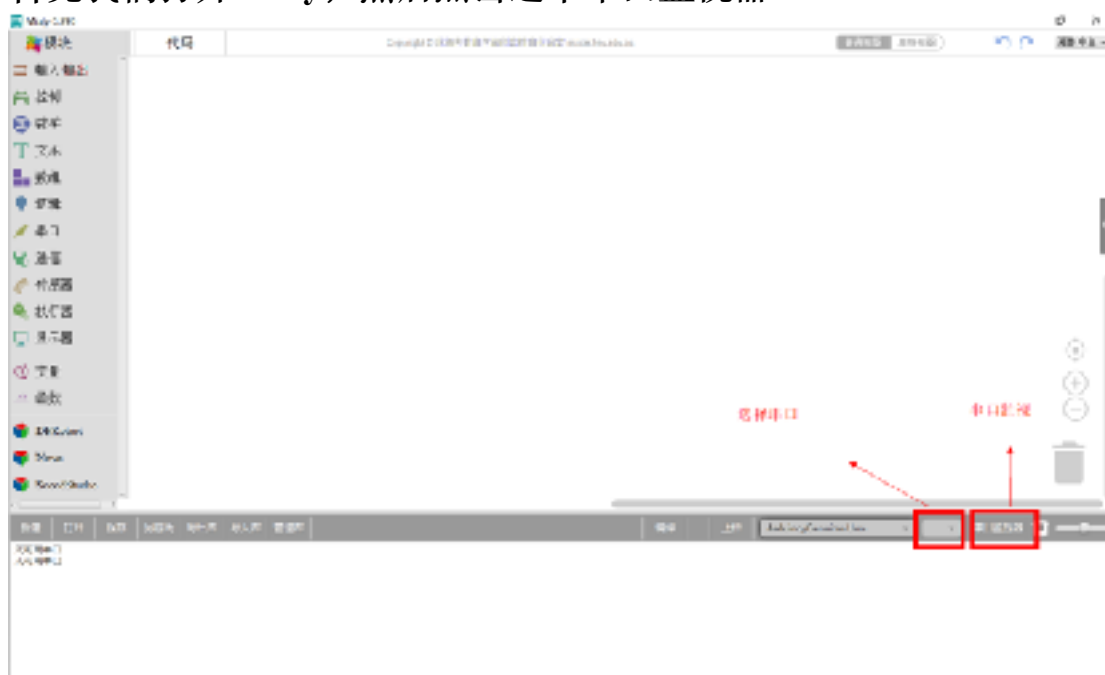
任务描述2：通过串口读取加速度数值

参考程序

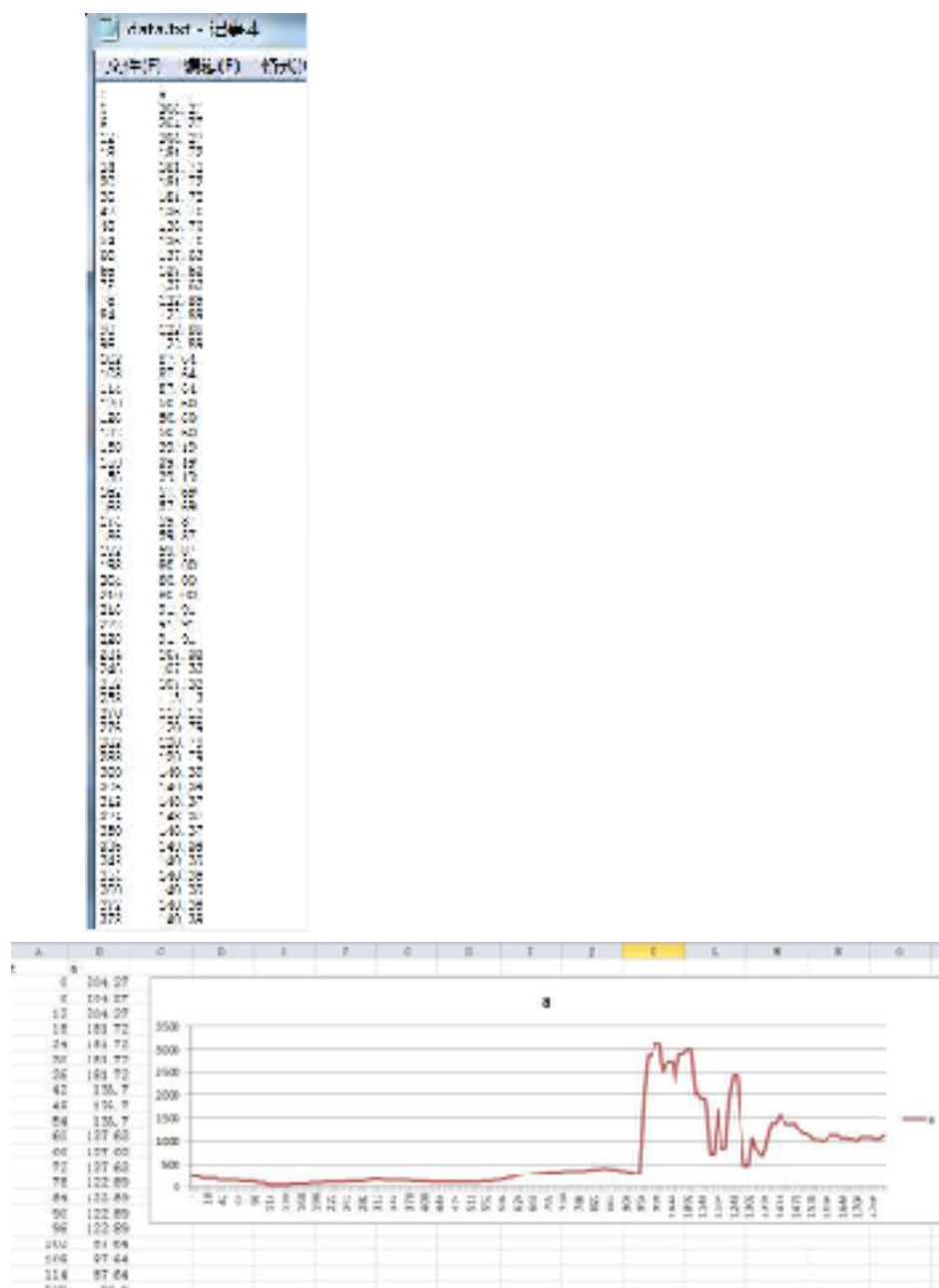


首先将这个程序上传，上传后我们需要通过软件来读串口数据，理论上任何可以都串口软件都可以读出我们串口的数据，在本文中我们使用mixly来读取串口

首先我们打开mixly，然后点击这个串口监视器



此时我们一定要注意选择我们对应的串口，不然软件会提示没有串口，进去之后我们就可以看到我们想要的的数据，如果出现乱码，就把波特率设置为115200。我们通过这个软件也可以把串口数据图形化。如下两幅图所示：



【探究思考】

物体在抛出去什么时候加速度为0

【挑战自我】

试着我们用加速度的变化值来测量物体空中降落时间。

【扩展阅读】

加速度的物理意义

表示质点速度变化的快慢的物理量。

举例：假如两辆汽车开始静止，均匀地加速后，达到10m/s的速度，A车花了10s，而B车只用了5s。它们的速度都从0变为10m/s，速度改变了10m/s。所以它们的速度变化量是一样的。但是很明显，B车变化得更快一些。我们用加速度来描述这个现象：B车的加速度（ $a=\Delta v/\Delta t$ ，其中的 Δv 是速度变化量）>A车的加速度。

显然，当速度变化量一样的时候，花时间较少的B车，加速度更大。也就是说B车的启动性能相对A车好一些。因此，加速度是表示物体速度变化快慢的物理量。

第十五课：萤火虫（生物体验课）

在夏天的夜晚，我们经常可以看到夜空中一些微弱的光在闪动，那就是萤火虫。今天我们就利用micro:bit的交互来做一群萤火虫，大家看一下他们是怎么通讯的。今天我们主要需要了解昆虫的生活习性，对程序不做要求。



【任务目标】

制作萤火虫，让班级里所有人的萤火虫一起交互，大家看一下有什么特点。

【知识点】

1. 了解萤火虫的生活习性，编写程序

【材料清单】

Micro.bit主控板、数据线、Makecode在线网站

【知识储备】

1. 萤火虫

萤火虫（英文：Firefly）又名夜光、景天、如熠耀、夜照、流萤、宵烛、耀夜等，属鞘翅目萤科，是一种小型甲虫，因其尾部能发出荧光，故名为萤火虫。这种尾部能发光的昆虫，约有近2000种，我国较常见的有黑萤、姬红萤、窗胸萤等几种。

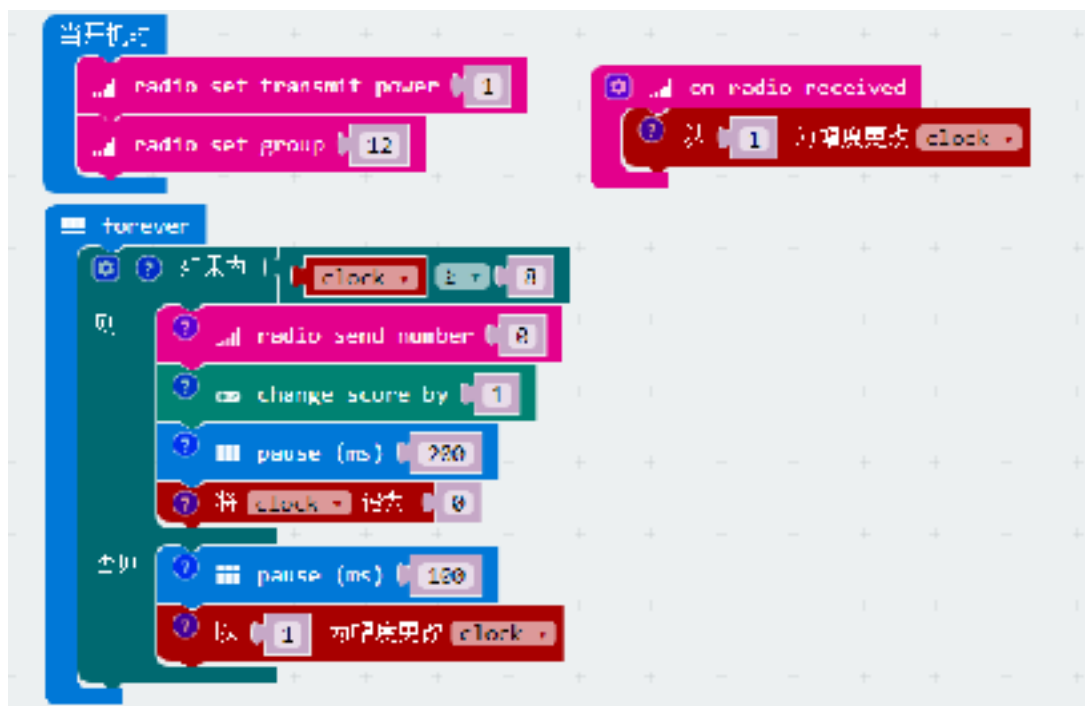
萤科昆虫的通称，全世界约2000种，分布于热带、亚热带和温带地区。根据中国几位专家的统计现发现的种类约有100余种，再加上未发现的种类，总共有150多种。夜间要发光，可分为水生类和陆生类两种。体型小至中型，长而扁平，体壁与鞘翅柔软。前胸背板平坦，常盖住头部。头狭小。眼半圆球形，雄性的眼常大于雌性。腹部

7~8节，末端下方有发光器，体内的荧光素和荧光素酶反应后生成的黄绿色荧光

【动手实践】

任务描述1：制作萤火虫，看看萤火虫在很多的情况下的发光特点

参考程序



这个程序大家可以试着读一下这个程序，看看这个程序的原理

将程序上传后，全班同学看看大家闪动的特点是什么？有没有发现大家的开机时间并不一样，但是闪烁的频率是一样的？

【探究思考】

这个clock变量在程序中起到什么作用？

【挑战自我】

试着让萤火虫用两种或者三种节奏来闪动。

【扩展阅读】

萤火虫的发光原理

萤火虫的发光是生物发光的一种。萤火虫的发光原理是：萤火虫有专门的发光细胞，在发光细胞中有两类化学物质，一类被称作萤光素（在萤火虫中的称为萤火虫萤光素(Firefly luciferin)），另一类被称为荧光素酶。荧光素能在荧光素酶的催化下消耗ATP，并与氧气发生反应，反应中产生激发态的氧化荧光素，当氧化荧光素从激发态回到基态时释放出光子。

反应中释放的能量几乎全部以光的形式释放，只有极少部分以热的形式释放，反应效率为95%，甲虫也因此而不会过热灼伤。人类到目前为止还没办法制造出如此高效的光源。

在虫的腹部下部有着很多白色斑块。其实是它的甲壳中对光透明的部分。在内部有一块白色

的膜，可以反射光。所以在日间这个部位呈现白色。

发光的生物学意义：

成虫利用物种特有的闪光信号来定位并吸引异性，借此完成求偶交配及繁殖的使命，少数萤火虫成虫利用闪光信号进行捕食，还有一种作用是作为警戒信号，即当萤火虫受到刺激时会发出亮光。