

Python艺术编程07

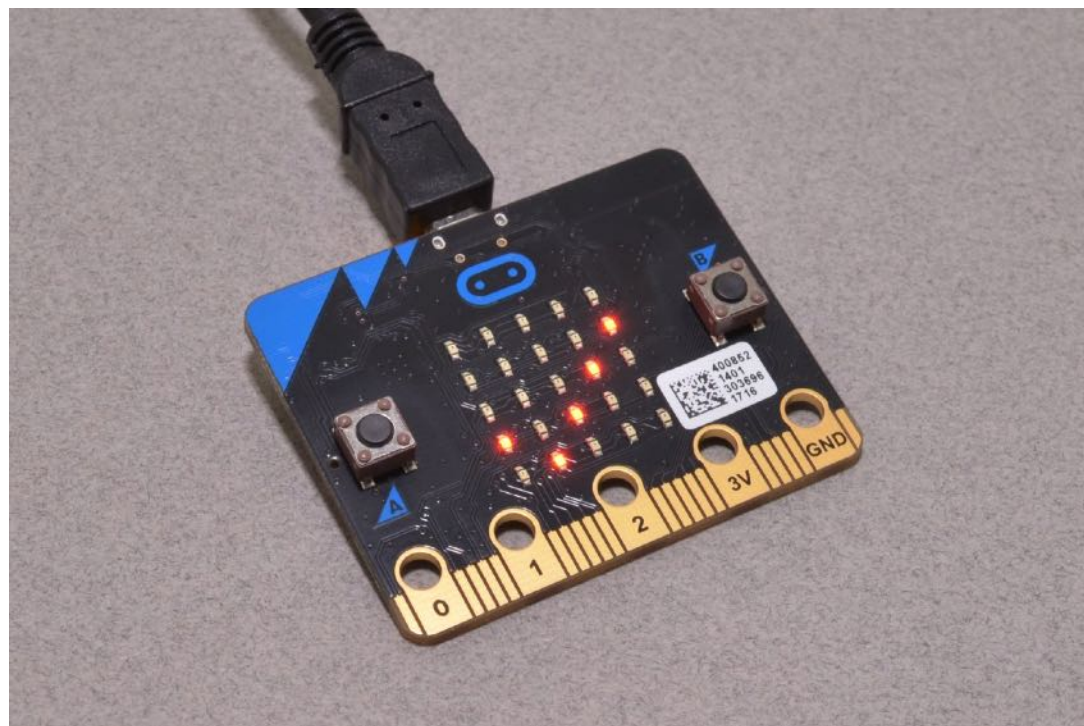
——开源硬件和灯光艺术

北京大学 陈斌

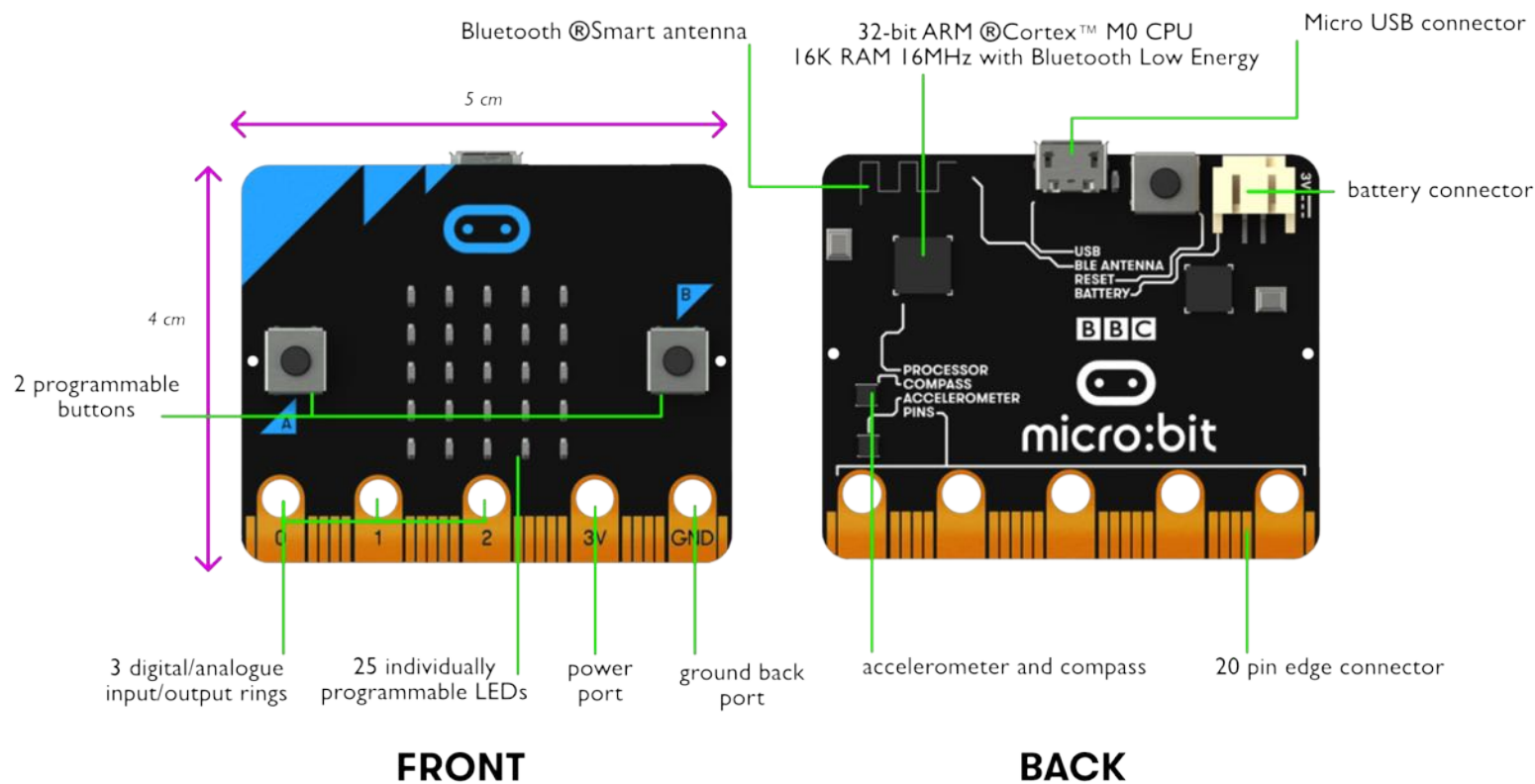
2018.10.29

目录

- microbit单片机介绍
- LED点阵控制
- NeoPixel灯光控制
- 按钮控制
- 灯光艺术



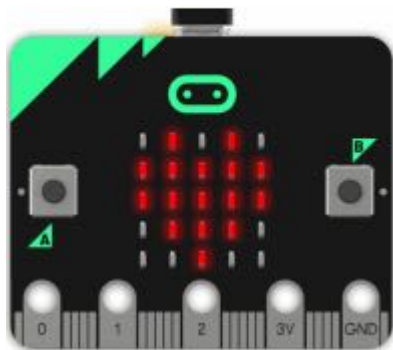
microbit from BBC介绍



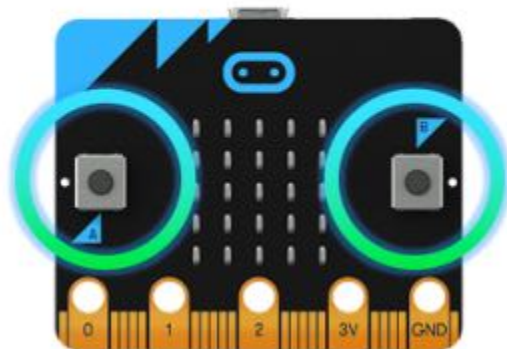
- 25个独立编程的LED
- 2个可编程的按钮
- 1个reset按钮
- microUSB接口
- 3V电源接口
- 光线传感器、温度传感器
- 加速计、电子罗盘
- 无线通信：射频以及蓝牙

microbit概貌

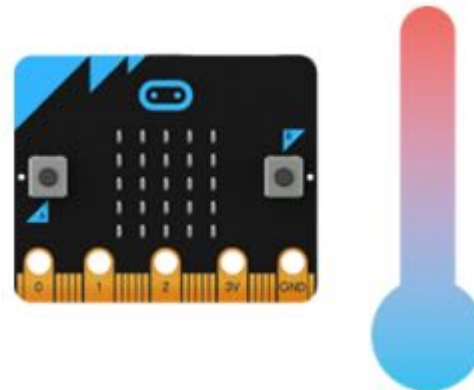
LED



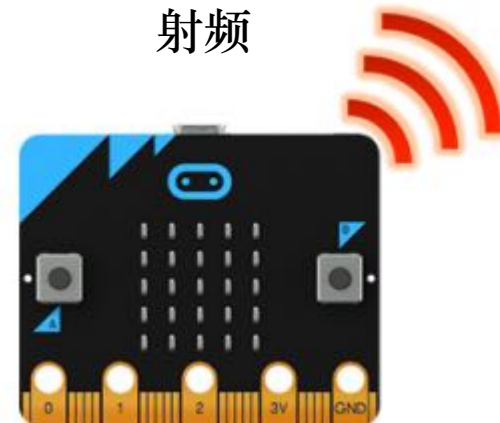
按钮传感器



温度传感器



射频



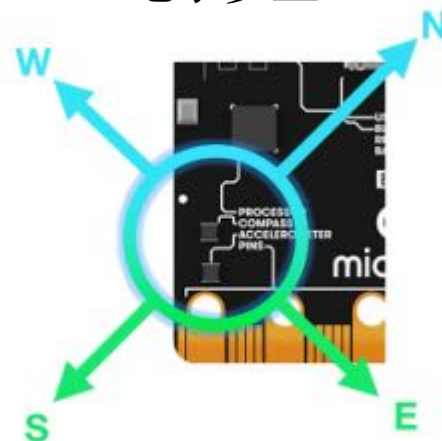
光线传感器



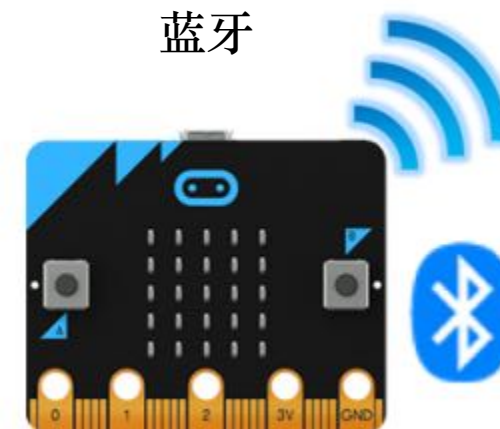
加速度计



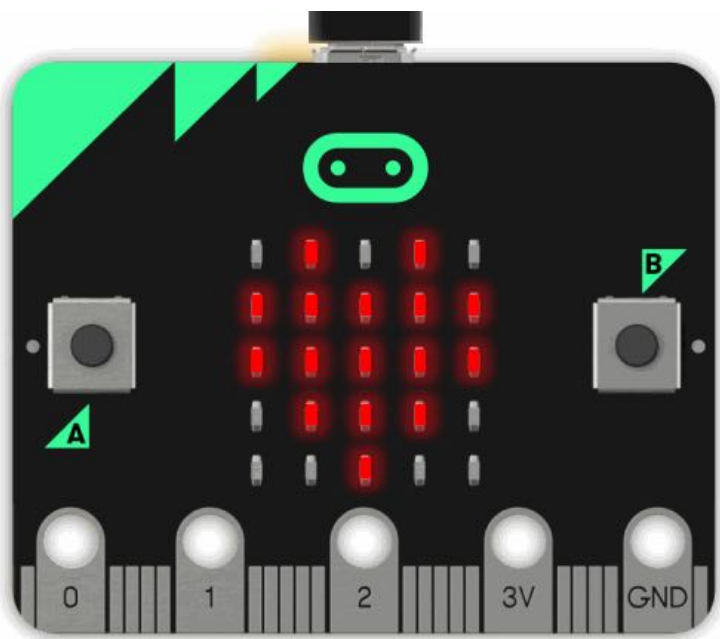
电子罗盘



蓝牙

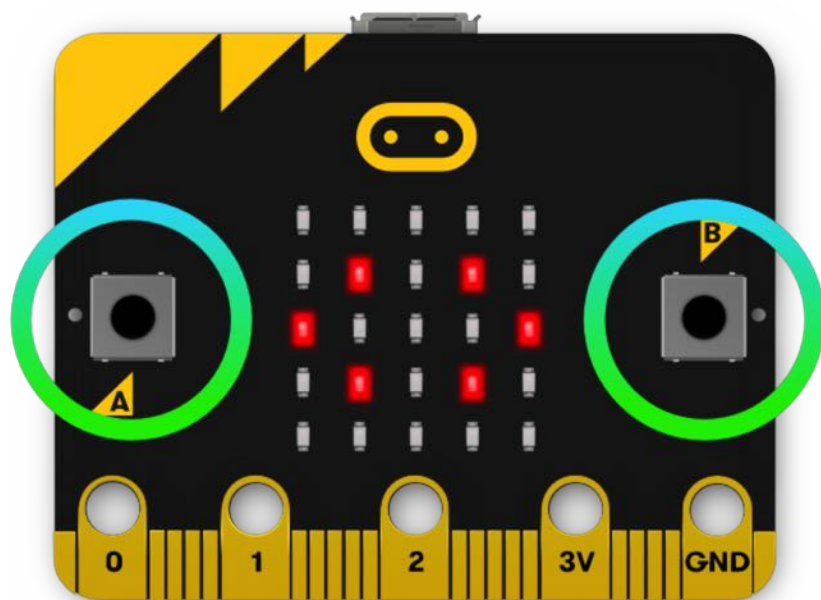


microbit特征： LED



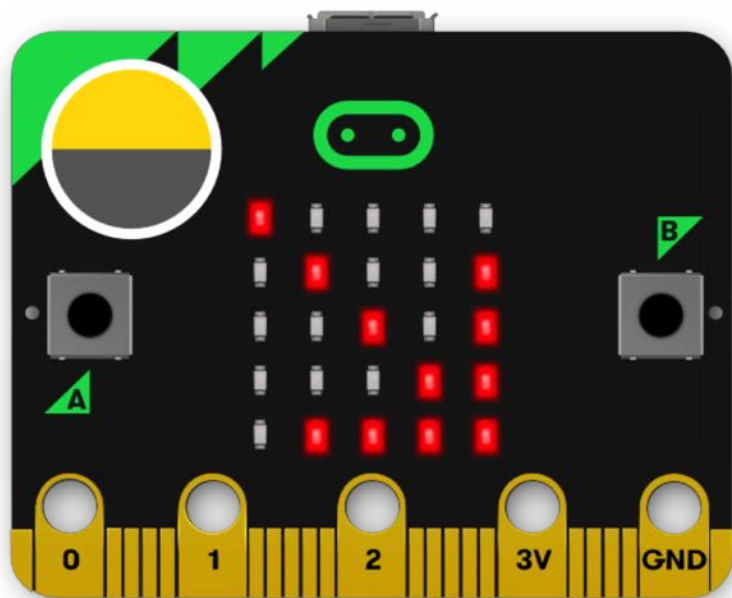
- micro:bit有25颗可独立编程的LED灯
- 可以用来显示文本，数字以及图像

microbit特征： 按钮



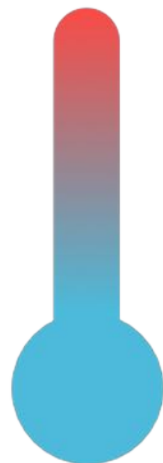
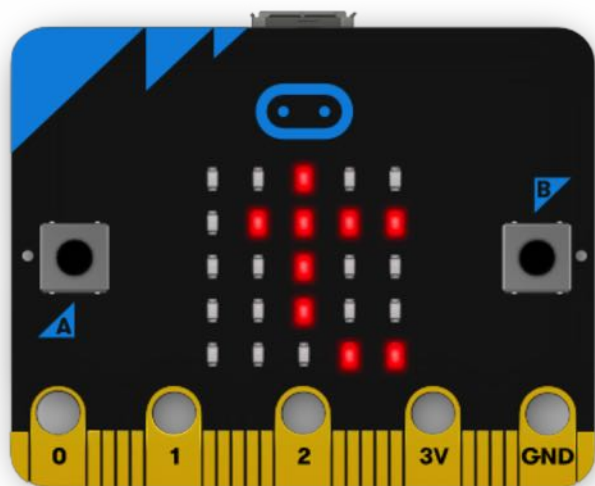
- 在micro:bit板子前面有2个按钮（标记了A和B）
- 可以检测按下这些按钮，运行代码
- 还可以检测这些按钮被按下的时间和次数

microbit特征： 光线传感器



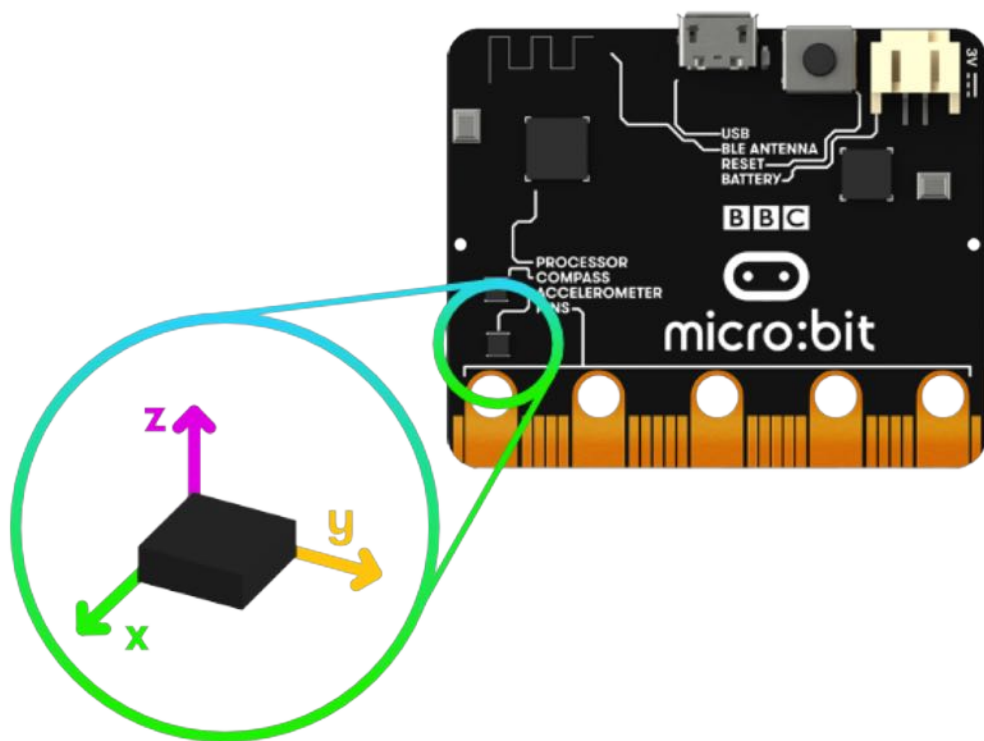
- 通过反转LED屏幕，micro:bit进入输入模式
- LED屏幕起到一个基础的光线传感器的作用
- 可以用来检测周围的光线

microbit特征： 温度传感器



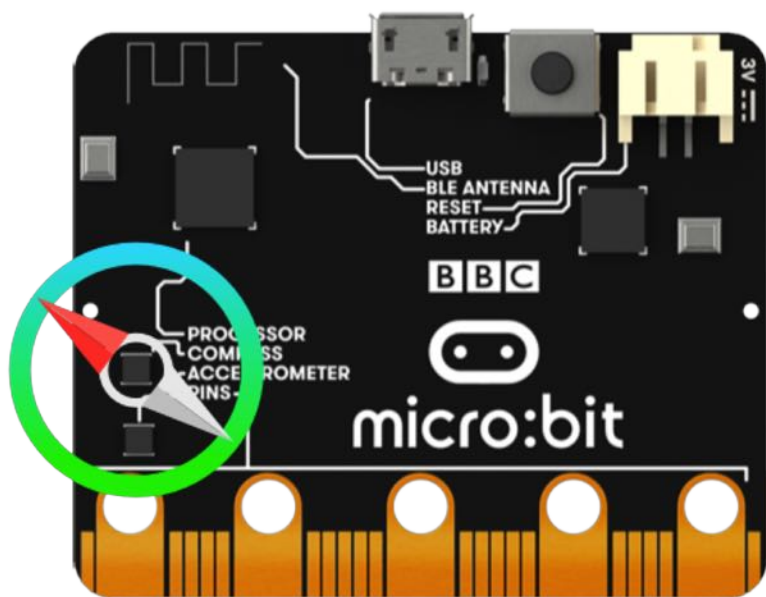
- 温度传感器可以让micro:bit检测当前环境温度(以摄氏度为单位)
- 温度传感器本来是用来检测处理器温度
- 如果处理器计算负载高的话，温度测量值也高

microbit特征： 加速度传感器



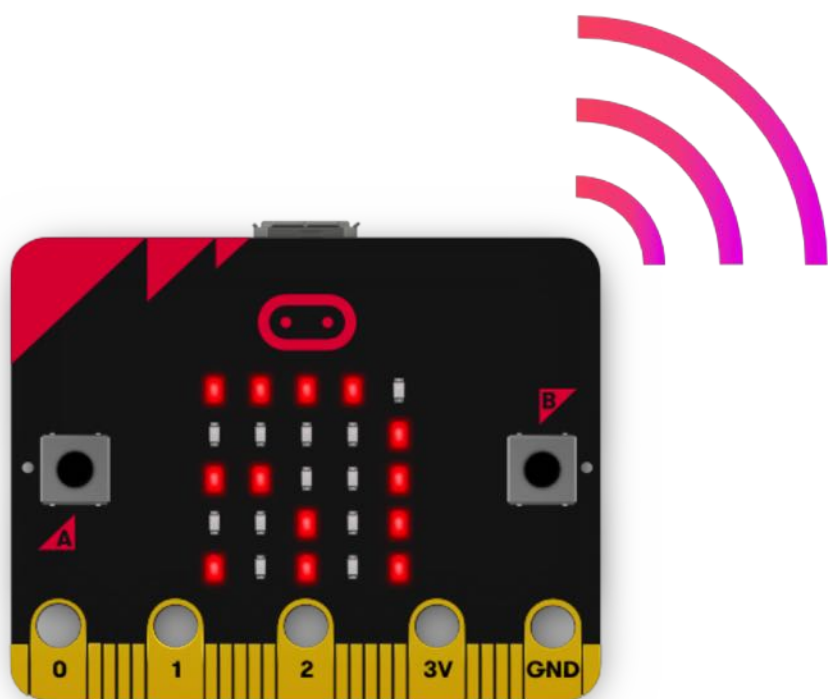
- 加速度传感器可以测量micro:bit的加速度
- 可以检测micro:bit的移动
- 也可以检测其他的动作
- 例如：摇动，倾斜以及自由落体。

microbit特征：指南针



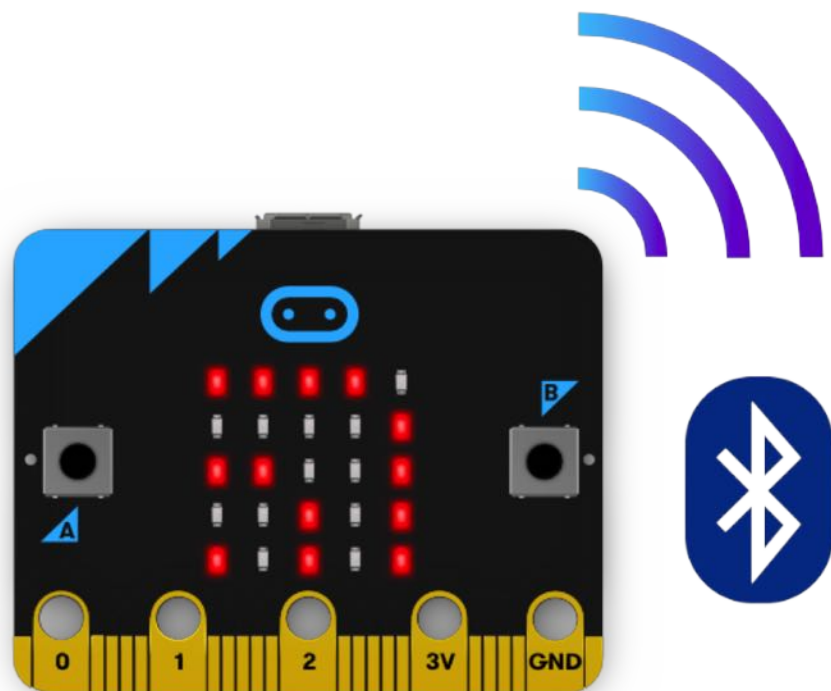
- 指南针用于检测地球磁场，可以探测到micro:bit面对的方向
- 在使用之前，需要校准指南针。
- “校准”是为了确保指南针的结果正确
- 在JavaScript积木块编辑器中，使用“指南针校准”积木块
- 在Python中用 `compass.calibrate()` 校准指南针

microbit特征：无线电



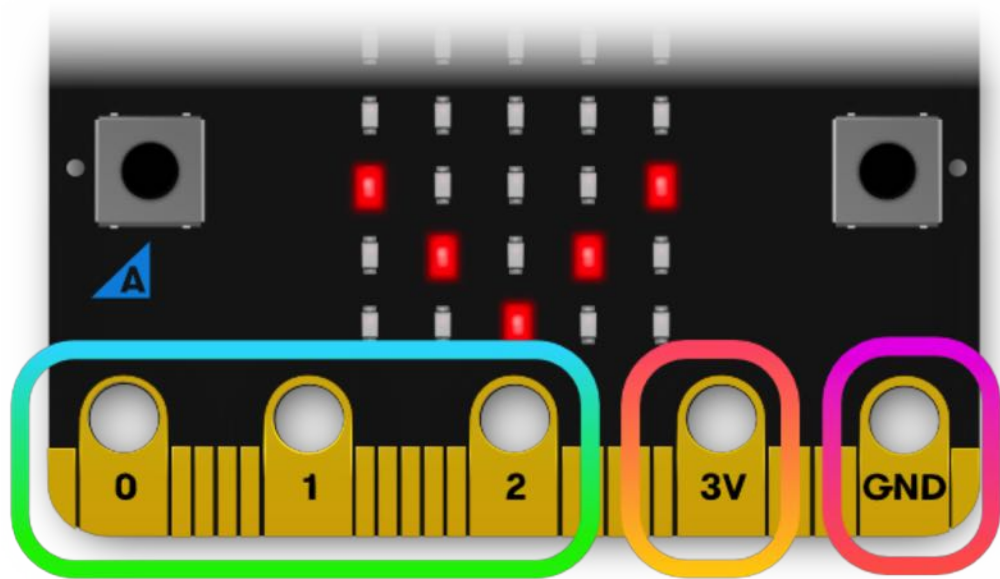
- 可以在2块甚至多块micro:bit板子之间进行无线通讯
- 用无线电发送信息到其他的micro:bit板子上
- 无线电可以有100个频道，调节发射功率，以免互相干扰
- 用于创建多人游戏以及更多有趣的发明！

microbit特征： 蓝牙（Python不可用）



- BLE(蓝牙低能量) 天线可以让micro:bit接收蓝牙信息
- 这可以让micro:bit和电脑，手机以及平板进行无线通信
- 可以用micro:bit控制手机
- 或者用手机发送无线代码到设备上

扩展引脚



- 在micro:bit连接器的边缘有25个外部接口
- 这些接口称作“引脚”
- 引脚可以扩展连接电机，LED灯
- 或者其他带引脚的电子元件编程
- 或者是连接外部传感器控制代码

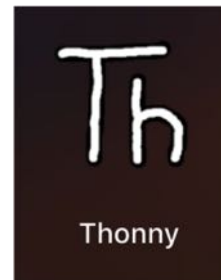
俄罗斯方块

俄罗斯方块
改编的
microbit方块

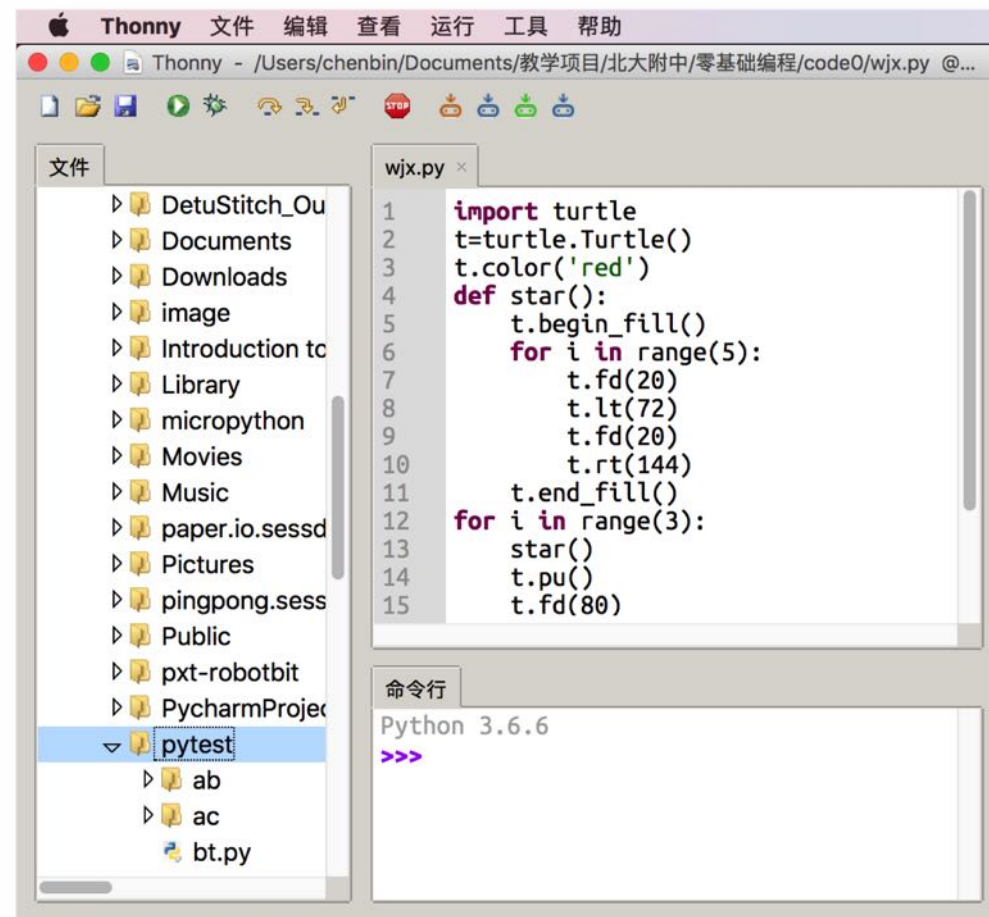
26组

陈丹丘&马涵聪&冉瑾瑜

集成开发环境：Thonny



- 跨平台Windows/macOS/Linux
- 自带最新版本Python3，无需安装Python
- 体积小巧，功能齐全
- 安装第三方模块很方便
- 可以连接microbit单片机编程
- 地小空开放实验室汉化版本
 - <https://github.com/chbpku/dxkStickIDE/tree/master/Setup>



实例 I：电子骰子



```
from microbit import *
from random import randint

while True:
    display.show('*')
    if accelerometer.was_gesture('shake'):
        display.clear()
        display.show(str(randint(1,6)))
        sleep(1000)
    sleep(10)
```

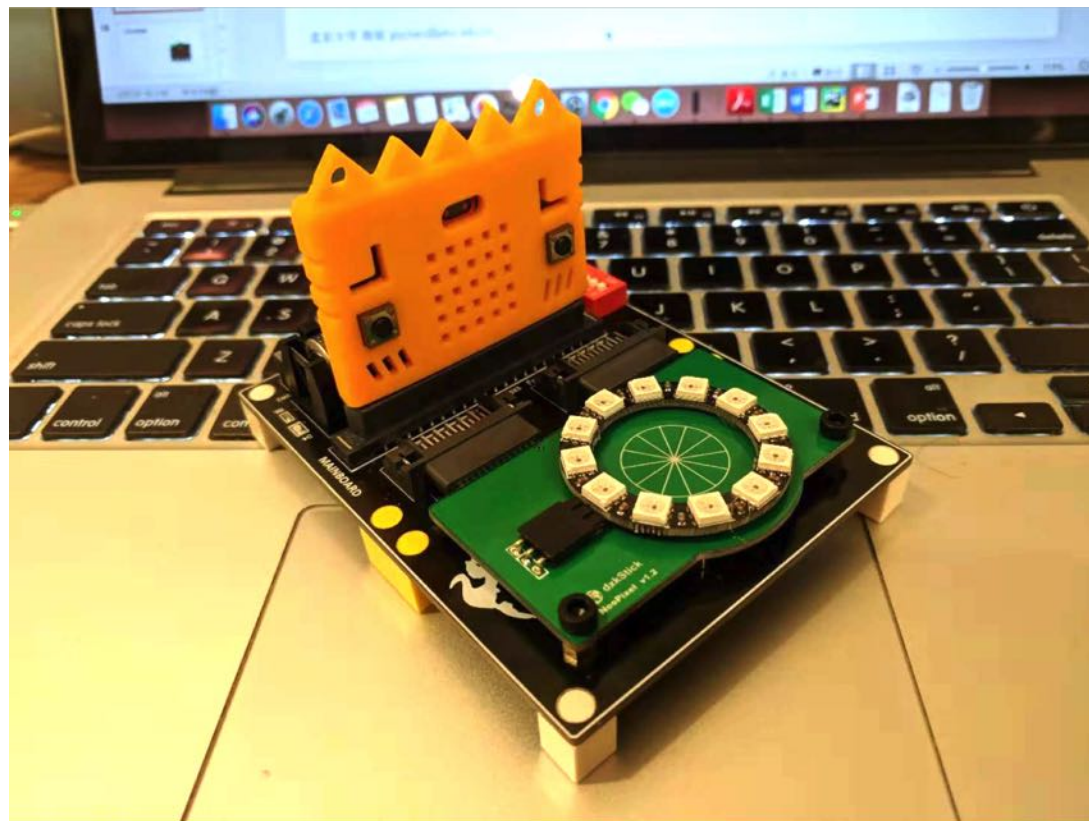

初识microbit-micropython



- USB线连接microbit
- 打开Thonny
- 输入程序，保存为py文件
- 点击“写入运行环境”
 - 稍等一下，闪烁结束
 - 出现成功Done字样
- 点击“写入当前代码”
 - 出现成功Done字样
- 立刻运行，可按RESET重启

microbit-micropython编程

- 导入microbit模块
 - `from microbit import *`
- 常用的函数和接口
 - 按钮: `button_a`, `button_b`
 - LED屏: `display`
 - LED图像: `Image`类
 - 加速计: `accelerometer`
 - 指南针: `compass`
- 彩灯模块NeoPixel
 - `import neopixel`



按钮控制button_a / button_b

- 有两个对象和三个方法
 - button_a, button_b
- 一直按着按钮
 - is_pressed()
- 按下放开按钮
 - was_pressed()
- 按过按钮的次数（距上次调用）
 - get_presses()

```
1 from microbit import *
2
3 while True:
4     if button_a.is_pressed() and button_b.is_pressed():
5         display.scroll('AB')
6         break
7     elif button_a.is_pressed():
8         display.show('A')
9     elif button_b.is_pressed():
10        display.show('B')
11    sleep(100)
12
```

LED点阵屏控制display

- 开关显示屏
 - `display.on()`, `off()`
- 显示字符串或者图像
 - `display.show(s)`
- 滚动显示字符串
 - `display.scroll(s)`
- 清除显示
 - `display.clear()`
- 点亮一个像素($b=0\sim9$)
 - `display.set_pixel(x,y,b)`

```
1  from microbit import *
2
3  display.show('HELLO')
4  sleep(1000)
5  display.scroll('WORLD')
6  sleep(1000)
7  display.clear()
8  for i in range(5):
9      display.set_pixel(i, i, 9)
10     sleep(200)
11 display.clear()
12 display.scroll('BYE!')
13
```


LED图像Image类

- 系统内置的图片
 - Image.HEART
 - HEART_SMALL, HAPPY, SMILE
 - SAD, CONFUSE, ANGRY, SILLY
 - MEH, YES, NO, CHESSBOARD等
- 创建一个Image对象
 - 通过字符串创建: Image(str)
 - '00000:22222:44444:77777:99999'
- 操作Image对象
 - set_pixel(x,y,n)
 - shift_left(n)/right/up/down

```
1 from microbit import *
2
3 display.show(Image.HAPPY)
4 sleep(1000)
5 a = Image('00000:22222:44444:77777:99999')
6 display.show(a)
7 s = [Image.YES, Image.NO, Image.SMILE, Image.DUCK, Image.CHESSBOARD]
8 for i in s:
9     display.show(i)
10    sleep(500)
11 s2 = []
12 for n in range(6):
13     s2.append(a.shift_up(n))
14 display.show(s2, delay=200, wait=True, loop=True)
```

彩灯NeoPixel模块

- 创建neopixel对象
 - `from microbit import *`
 - `import neopixel`
 - `np = neopixel.NeoPixel(pin16,12)`
- 设置彩灯 (0~11)
 - `np[0]=(255,0,0)`
 - `np[-1]=(0,255,0)`
- 点亮/清除彩灯
 - `np.show()`
 - `np.clear()`

```
1  from microbit import *
2  import neopixel
3  from random import randint
4
5  # 创建一个彩灯对象
6  np = neopixel.NeoPixel(pin16, 12)
7  while True:
8      # 循环每个灯珠
9      for pixel_id in range(len(np)):
10         red = randint(0, 60)
11         green = randint(0, 60)
12         blue = randint(0, 60)
13         # 设置随机颜色RGB
14         np[pixel_id] = (red, green, blue)
15         # 显示彩灯
16         np.show()
17         sleep(100)
18     # 清除彩灯
19     np.clear()
```

上机作业（周一周四）

- 编程实现灯光艺术作品
- 可以选择LED屏，或者NeoPixel彩灯，或两者综合
- 可以选用按钮进行灯光控制
- 要求上传带注释的源代码和运行照片
- 根据创意和实现效果评分