



BAIJIA DAGU
百佳大谷

Microbit 初学者套件

Arduino starter kit

目 录

LED模块.....	1
RGB模块.....	4
按键模块.....	6
光敏模块.....	8
有源蜂鸣器模块.....	11
无源蜂鸣器模块.....	13
触摸按键模块.....	17
循迹模块.....	19
风扇模块.....	22
红外接收模块.....	24
麦克风模块.....	26
电位器模块.....	28
超声波模块.....	30

LED模块

（红黄绿蓝白）

模块介绍：LED是发光二极管的缩写。它通常是由砷化镓，磷化镓半导体材料。LED有两个电极，一个正极和一个负极，当正向电流通过时，它就会发光。它可以是红、蓝、绿或黄光，光的颜色取决于它所用的材料。

工作电压：3.3V到5V

输入类型：数字信号，对于0V到5V的模拟电压，值（0-255）也允许作为数字值输入。

接口模式：pH2.0-3P

模块尺寸：35*26.3mm

模块重量：47g

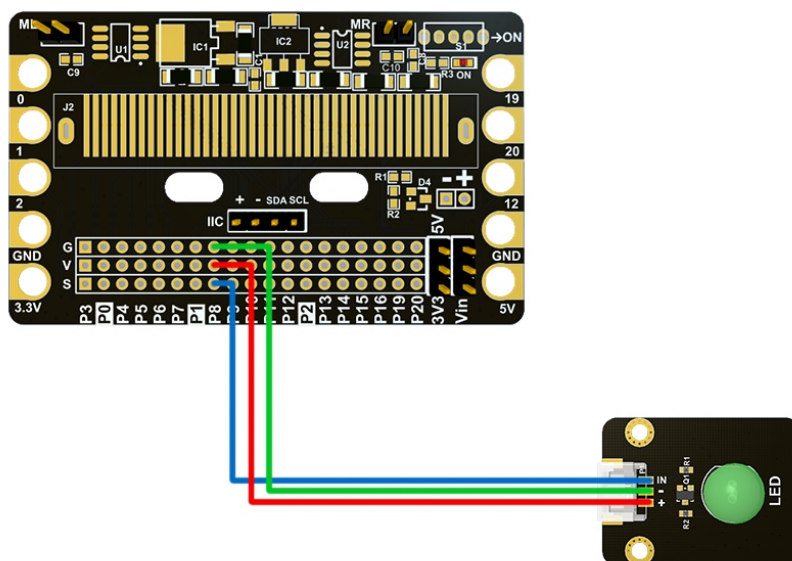
引脚定义：

IN	输入
+	VCC
-	GND

Micro:bit

样例一

连接图



代码

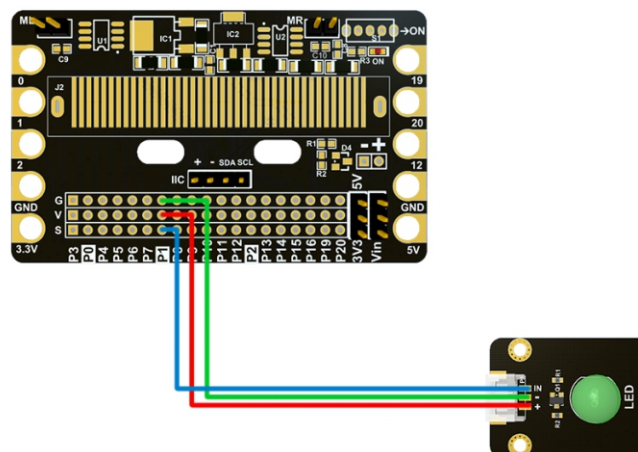
```
from microbit import *
#write your program:
while True:
    #写入接入 pin5 的 LED 数字值 1
    pin5.write_digital(1)
    sleep(500)
    #写入接入 pin5 的 LED 数字值 0
    pin5.write_digital(0)
    sleep(500)
```

Microbit图形化编程



结果：结果：引脚8插上LED灯后，LED灯会循环的亮1秒灭1秒。

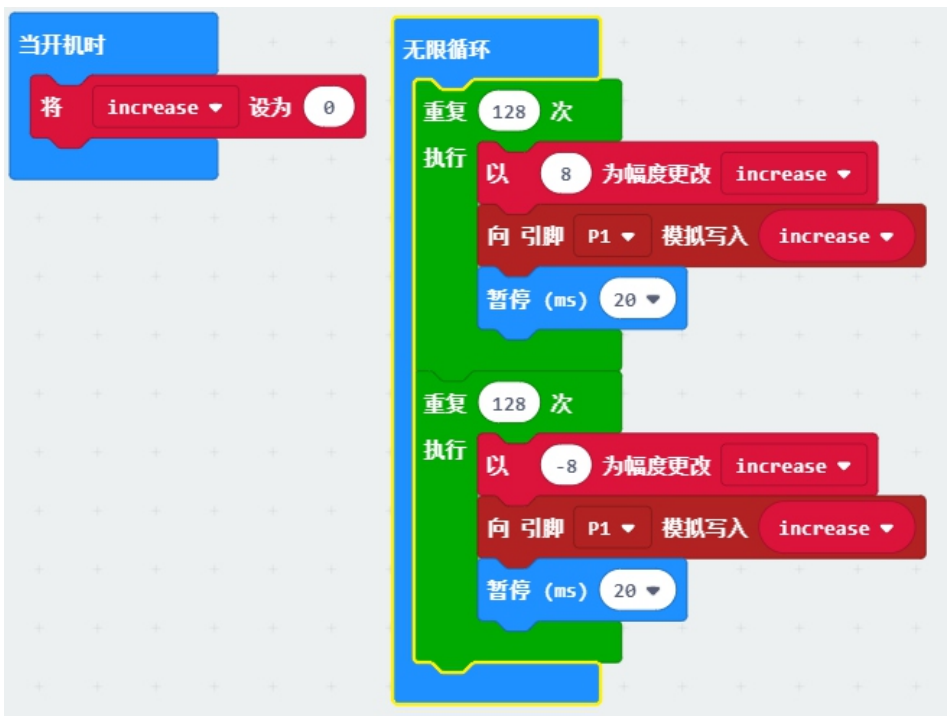
Micro:bit 样例二 连接图



代码

```
from microbit import *  
#write your program:  
increase = 0  
while True:  
    #循环 1 到 128, 每次写入接入 pin1 的 LED 模拟值加 8  
    for i in range(1,128):  
        increase += 8  
        pin1.write_analog(increase)  
        sleep(20)  
    #循环 1 到 128, 每次写入接入 pin1 的 LED 模拟值减 8  
    for i in range(1,128):  
        increase -= 8  
        pin1.write_analog(increase)  
        sleep(20)
```

Microbit图形化编程



结果：引脚1上的led灯有个逐渐由亮到灭的一个缓慢过程，而不是直接的亮灭。

RGB 模块

模块介绍：RGB 模块由三个LED组成。每个LED实际上有一个红光、一个绿光和一个蓝光。这三种颜色的LED能够产生任何颜色。RGB LED带有红色、绿色和蓝色发射器，通常使用三条线与一根公共导线（阳极或阴极）连接，该模块是共阴极LED。

工作电压：. 3V到5V

输入类型：PWM，值（0-255）作为数字值分别输入RGB三个接口。通过PWM来控制RGB发光二极管的颜色。

接口模式：pH2.0-4P

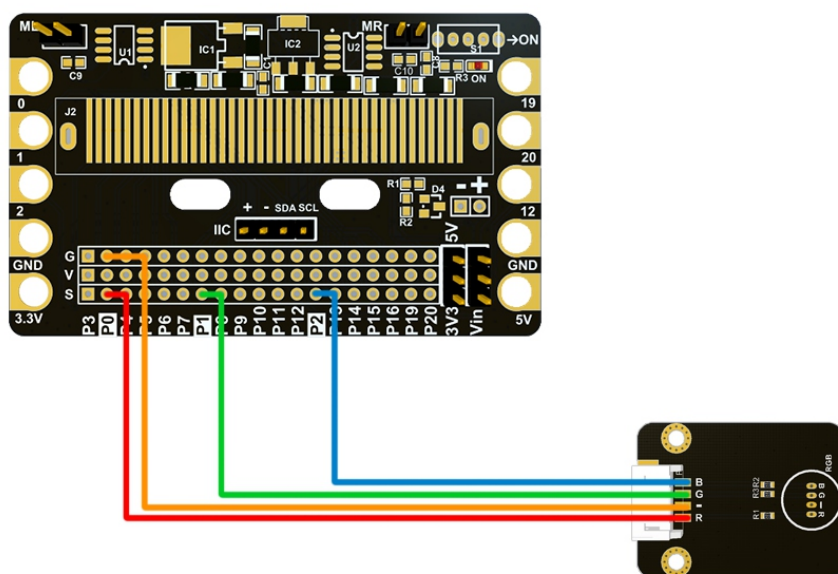
模块尺寸：35*26.3mm

模块重量：49g

引脚定义：

B	蓝色输入
G	绿色输入
-	GND
R	红色输入

Micro:bit
样例一
连接图



代码

```
from microbit import *
#设置 RGB 的红、绿、蓝数值
def set_rgb(red, green, blue):
    #设置 pin0 脚上的红色值，设置 pin1 脚上的绿色值，设置 pin2 脚上的蓝色值
    pin0.write_analog(red)
    pin1.write_analog(green)
    pin2.write_analog(blue)

while True:
    #RGB 显示红色
    set_rgb(1023, 0, 0)
    sleep(500)
    #RGB 显示绿色
    set_rgb(0, 1023, 0)
    sleep(500)
    #RGB 显示蓝色
    set_rgb(0, 0, 1023)
    sleep(500)
    #RGB 显示白色
    set_rgb(1023, 1023, 1023)
    sleep(5000)
```

Microbit图形化编程



结果：模块的R接入引脚0，G接入引脚1，B接入引脚2的RGB led灯以红绿蓝白为周期循环闪烁。

按键模块

模块介绍：按钮是用于控制电子设备的常见组件。他们是通常用作连接或断开电路的开关。在正常情况下，按钮的两个触点处于断开状态，只有按下按钮时它们才闭合。

工作电压：3.3V到5V

输出类型：数字信号。按键按下，高电平；按键弹起，低电平。

接口模式：pH2.0-3P

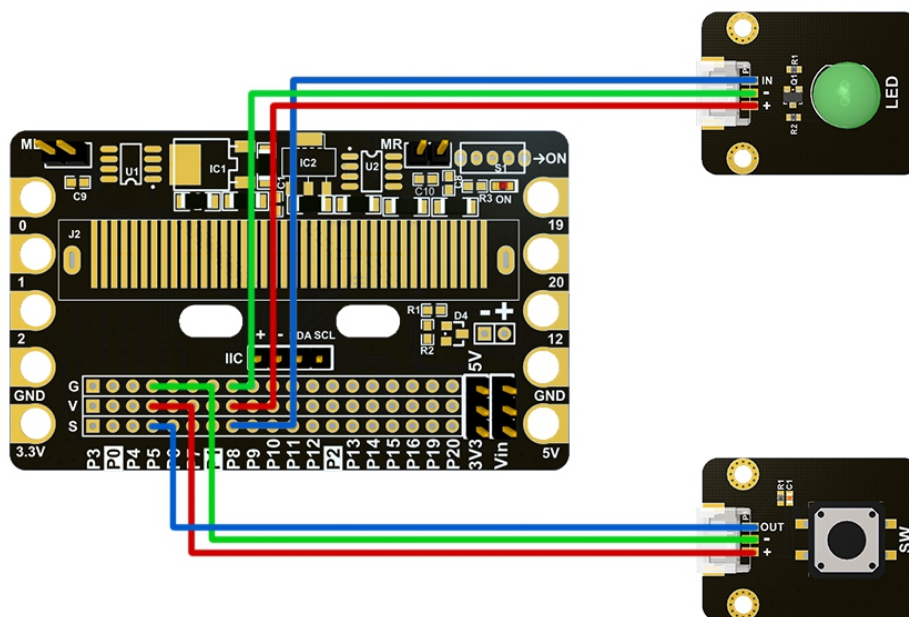
模块尺寸：35*26.3mm

模块重量：49g

引脚定义：

OUT	输出
+	VCC
-	GND

Micro:bit
样例一
连接图



代码

```
from microbit import *  
#write your program:  
while True:  
    #读取接入 pin5 的按键的数字值  
    state = pin5.read_digital  
    #数字值为 1 时，接入 pin8 的 led 灯输入数字值 1  
    #数字值为 0 时，接入 pin8 的 led 灯输入数字值 0  
    if state == 1:  
        pin8.write_digital(1)  
    else:  
        pin8.write_digital(0)
```

Micro:bit图形化编程



结果：引脚5接入按键，引脚8接入led灯，当按键被按下时，led灯亮起；松开按键时，LED灯熄灭。

光敏模块

模块介绍：光敏电阻是一种光控可变电阻器。光敏电阻具有光电导性，可用于光敏探测器电路。光敏电阻是由高电阻半导体制成的。在黑暗中光敏电阻的电阻可以高达几兆欧（MΩ），而在充足光线下，光敏电阻的电阻可以低至几百欧姆。如果光敏电阻上的入射光超过一定的频率，光子被半导体吸收给束缚电子足够的能量跳跃进入传导带。由此产生的自由电子导电，从而降低电阻。

工作电压：3.3V到5V

输出类型：数字、模拟接口。此模块，入射光线越强，输出模拟值越低。

接口模式：pH2.0-4P

模块尺寸：35*26.3mm

模块重量：42g

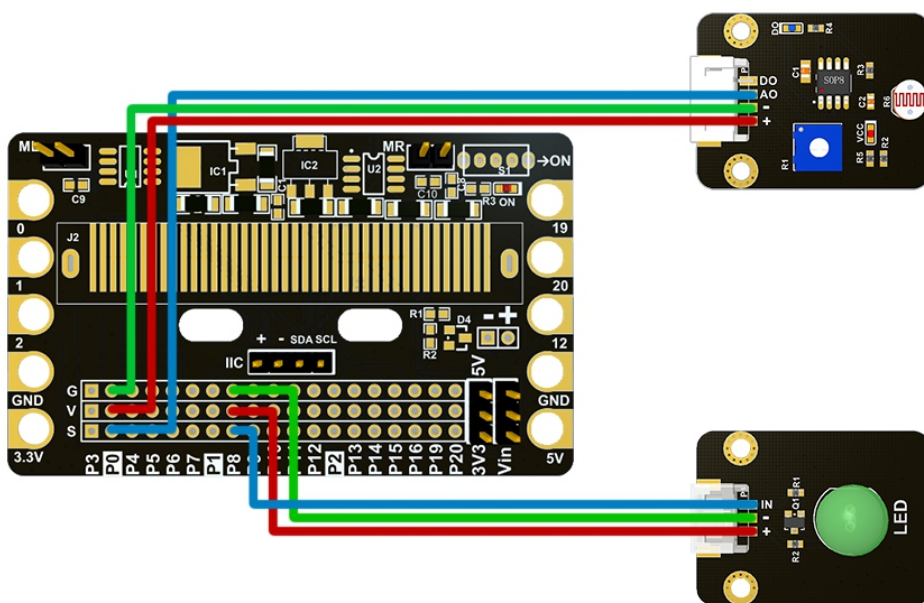
引脚定义：

DO	数字输出
AO	模拟输出
-	GND
+	VCC

Micro:bit

样例一

连接图



代码

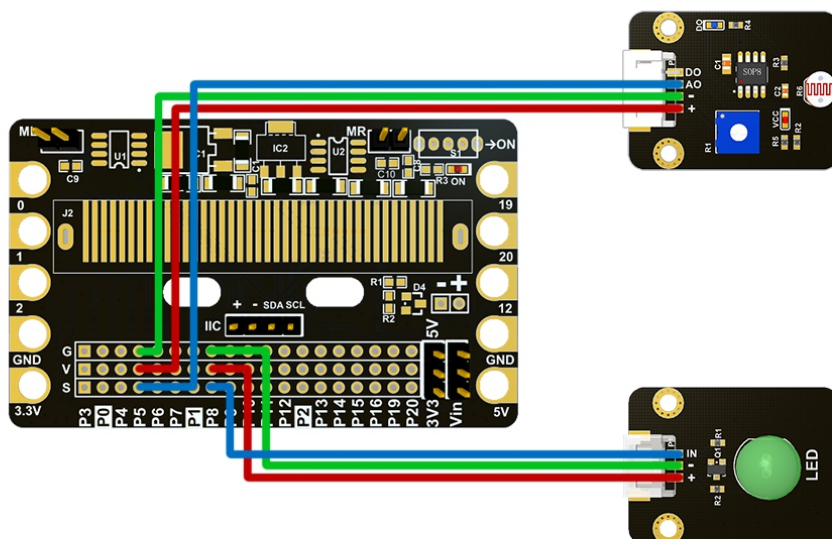
```
from microbit import *  
#write your program:  
while True:  
    #读取接入 pin5 的光敏电阻的模拟值  
    state = pin5.read_analog  
    #模拟值大于 600 时，写入接入 pin8 的 LED 数字值 1；反之，写入接入 pin8 的 LED 数字值 0  
    if state > 600:  
        pin8.write_digital(1)  
    else:  
        pin8.write_digital(0)
```

Micro:bit图形化编程



结果：引脚0接入光敏电阻的模拟输出，引脚8接入led灯，在模拟输出值高于600时，led亮；反之，led灯不亮。

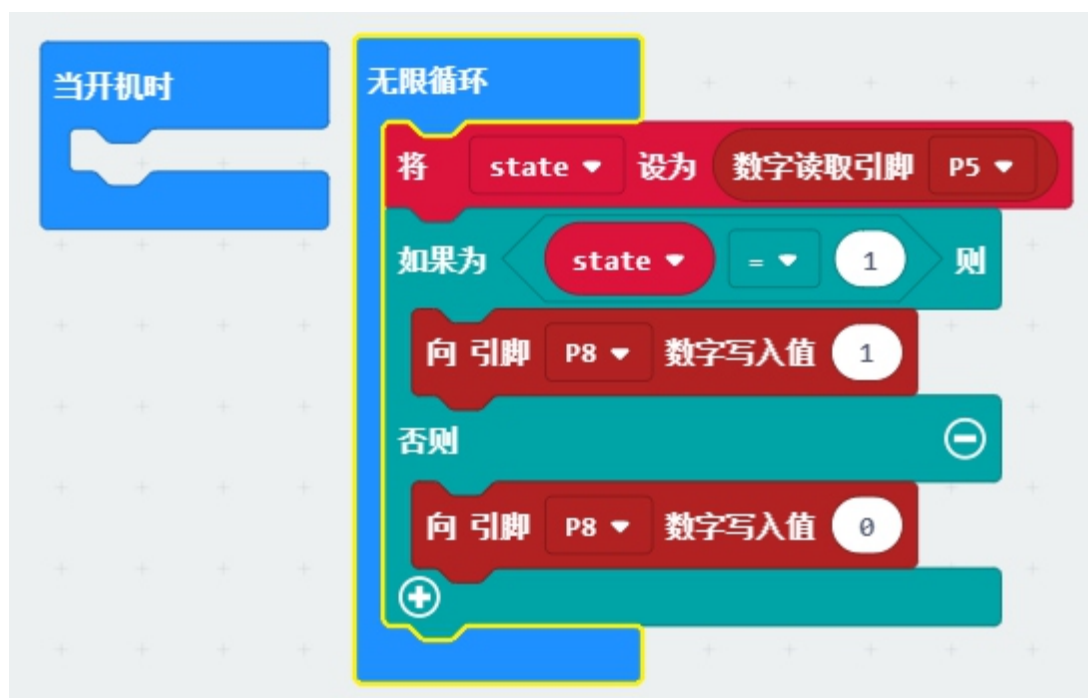
Micro:bit 样例一 连接图



代码

```
from microbit import *
#write your program:
while True:
    #读取接入 pin5 的光敏电阻的数字值
    state = pin5.read_digital
    #数字值为 1 时，写入接入 pin8 的 LED 数字值 1
    if state == 1:
        pin8.write_digital(1)
    #数字值为 0 时，写入接入 pin8 的 LED 数字值 0
    else:
        pin8.write_digital(0)
```

Micro:bit图形化编程



结果：引脚5接入光敏电阻的数字输出，引脚8接入Led灯，当引脚5输出的数字信号为0时，led灯亮；反之，led灯不亮。

提示：尝试用螺丝刀扭一下模块的R1上的十字口，在测试下上例，观察看看串口输出的内容有什么变化。

有源蜂鸣器模块

模块介绍：蜂鸣器是一种音频信号装置。作为一种集成结构的电子蜂鸣器，采用直流电压供电的，广泛应用于计算机、打印机、复印机、警报器、电子玩具、汽车电子设备、电话、计时器和其他电子产品语音设备。蜂鸣器按其驱动方式可分为有源蜂鸣器和无源蜂鸣器。

有源蜂鸣器和无源蜂鸣器的区别：这里的源不是指电源，而是指震荡源，也就是说，有源蜂鸣器内部带有震荡源，所以只要通电就会叫，而无源蜂鸣器内部不带震荡源，所以用直流信号是无法令其鸣叫的。

此处讲解的蜂鸣器是有源蜂鸣器。只要有电源。我们输出高低电平交替，使蜂鸣器鸣响。

工作电压：3.3V-5V

输入类型：数字信号。

接口模式：pH2.0-3P

模块尺寸：35*26.3mm

模块重量：47g

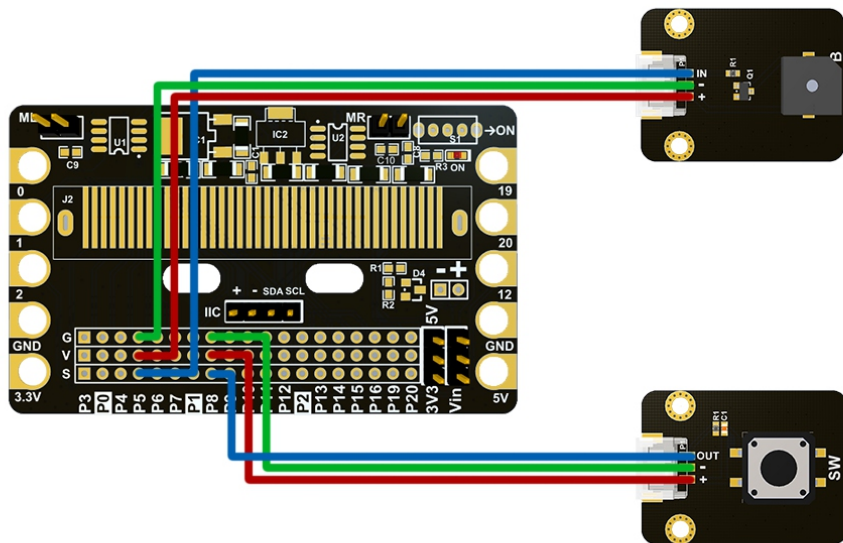
引脚定义：

IN	数字输入
+	GND
-	VCC

Micro:bit

样例一

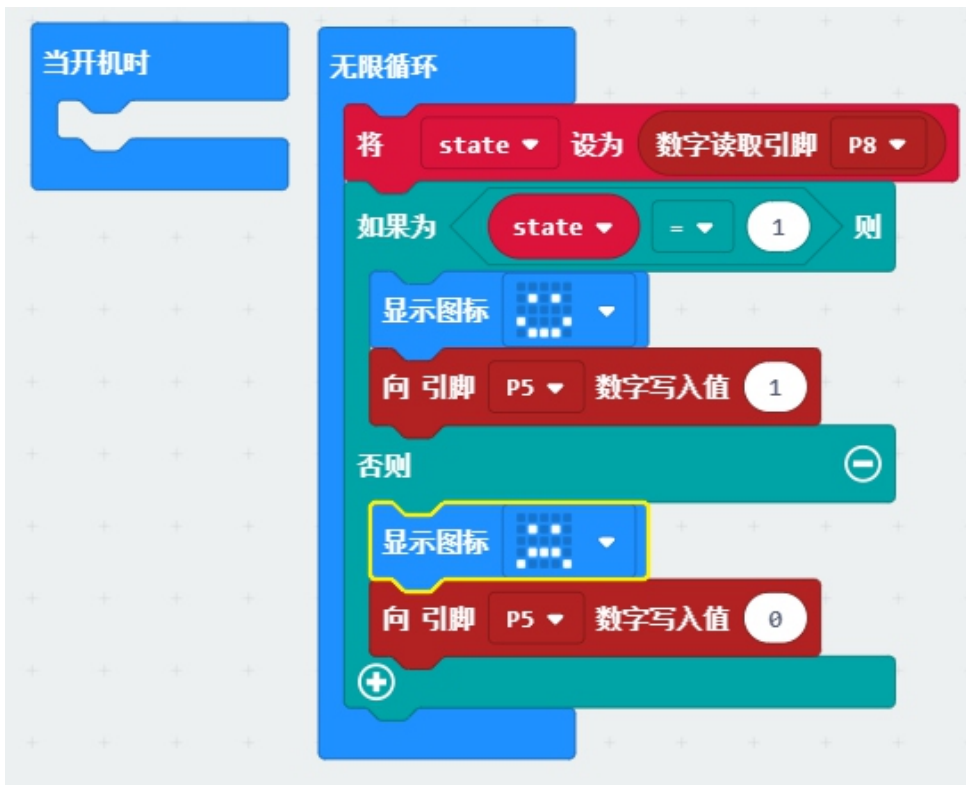
连接图



代码

```
from microbit import *  
#write your program:  
  
while True:  
    #读取接入 pin8 的按键的数字值  
    state = pin8.read_digital()  
    #数字值为 1 时显示笑脸，写入接入 pin5 的有源蜂鸣器数字值为 1  
    if state==1:  
        display.show(Image.HAPPY)  
        pin5.write_digital(1)  
    #数字值为 0 时显示苦脸，写入接入 pin5 的有源蜂鸣器数字值为 0  
    else:  
        display.show(Image.SAD)  
        pin5.write_digital(0)
```

Microbit图形化编程



结果：按键接入引脚8，有源蜂鸣器接入引脚5，当按下按键时，蜂鸣器发出响声并显示笑脸，释放按键时，蜂鸣器不发出响声并显示哭脸。

无源蜂鸣器模块

模块介绍：蜂鸣器是一种音频信号装置。作为一种集成结构的电子蜂鸣器，采用直流电压供电的，广泛应用于计算机、打印机、复印机、警报器、电子玩具、汽车电子设备、电话、计时器和其他电子产品语音设备。蜂鸣器按其驱动方式可分为有源蜂鸣器和无源蜂鸣器。是输入模块，数字接口。

有源蜂鸣器和无源蜂鸣器的区别：这里的源不是指电源，而是指震荡源，也就是说，有源蜂鸣器内部带有震荡源，所以只要通电就会叫，而无源蜂鸣器内部不带震荡源，所以用直流信号是无法令其鸣叫的。

此处讲解的蜂鸣器是无源蜂鸣器。

工作电压：3.3V到5V

输入类型：数字信号。

接口模式：pH2.0-3P

模块尺寸：35*26.3mm

模块重量：40g

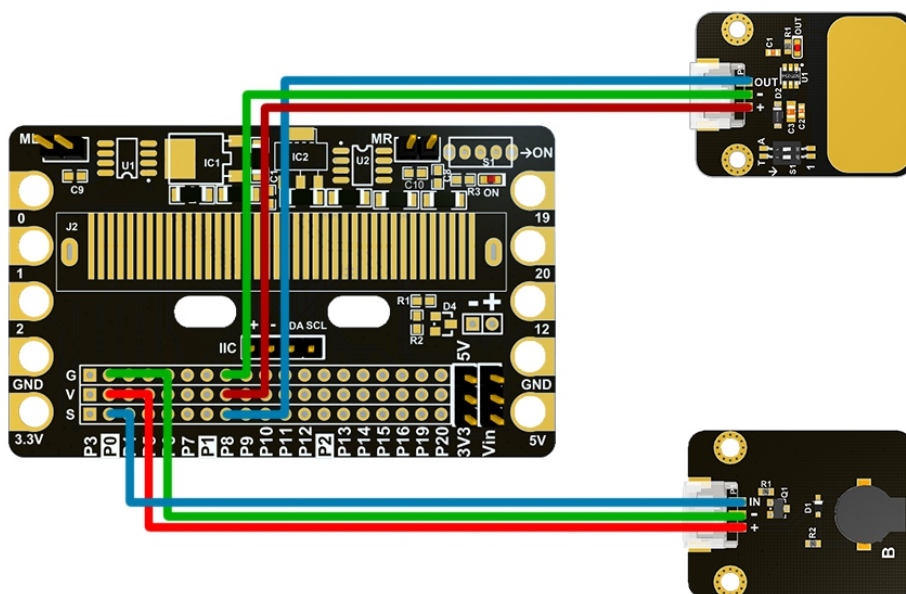
引脚定义：

IN	输入
+	GND
-	VCC

Micro:bit

样例一

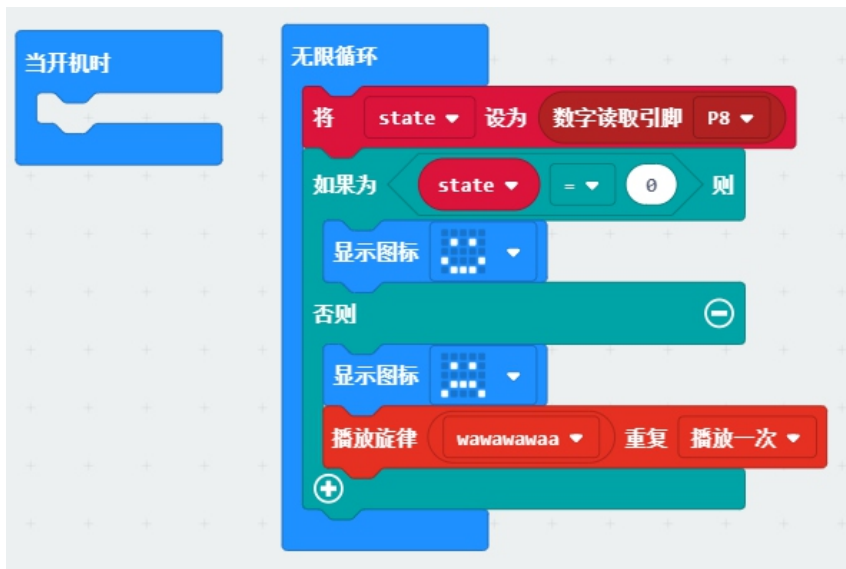
连接图



代码

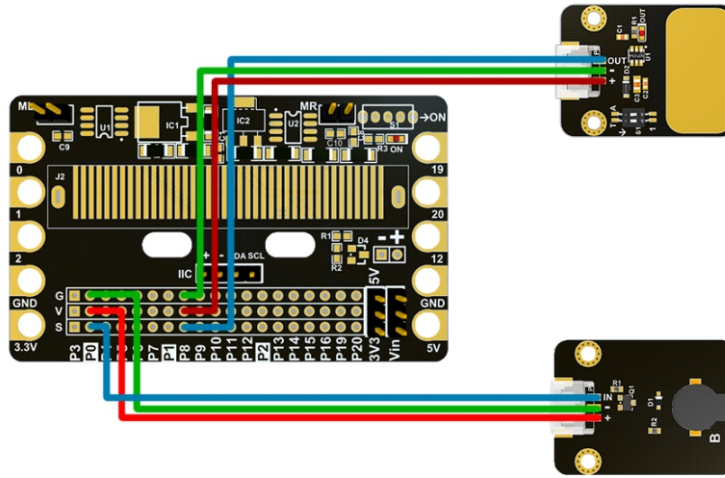
```
from microbit import *  
#write your program:  
#导入音乐库  
import music  
while True:  
    #读取接入 pin8 的触摸模块的数字值  
    state = pin8.read_digital()  
    #数字值为 0 时，显示笑脸；反之显示哭脸并播发音乐 WAWAWAWAA  
    if state ==0:  
        display.show(Image.HAPPY)  
    else:  
        display.show(Image.SAD)  
        music.play(music.WAWAWAWAA)
```

Microbit图形化编程



结果：引脚8接入触摸模块，引脚0接入无源蜂鸣器，当触摸模块输出高电平时，Microbit板上显示哭脸并且无源蜂鸣器发出wawa的声响；当触摸模块输出低电平时，Microbit板上显示笑脸并无源蜂鸣器不发出声响。

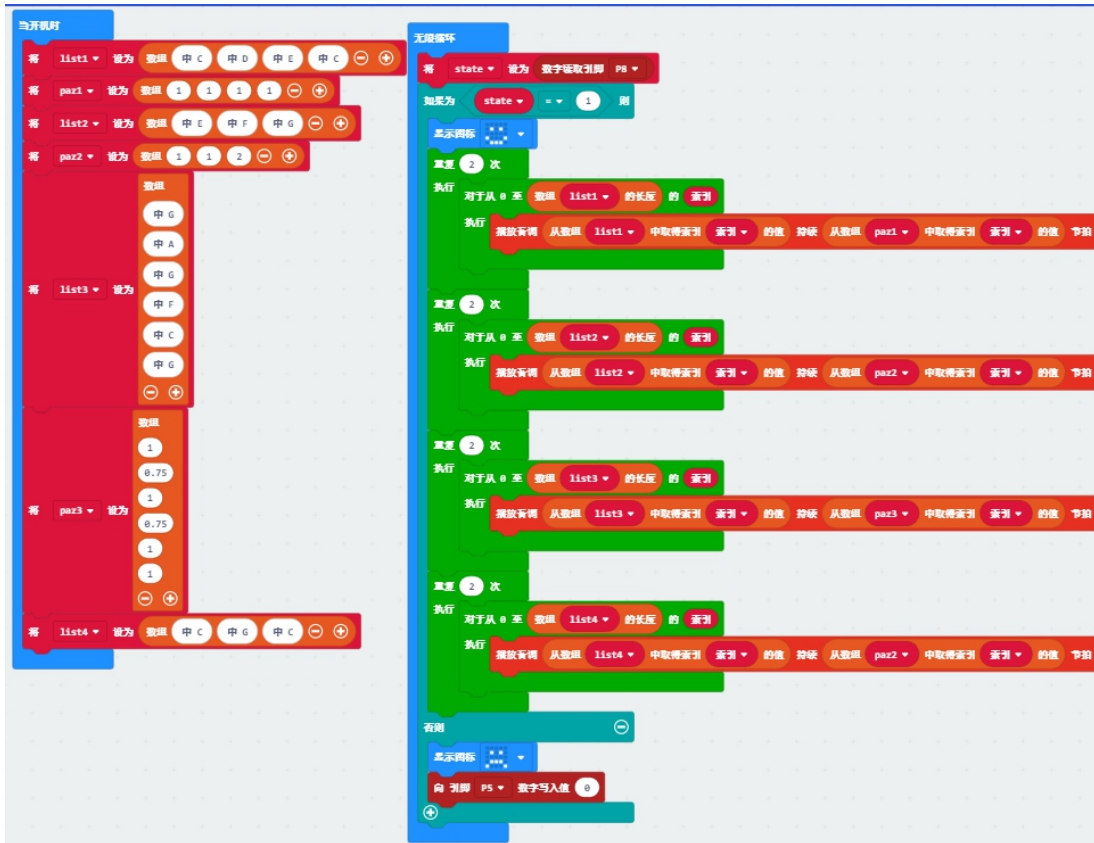
Micro:bit 样例二 连接图



```
from microbit import *
#write your program:
#导入音乐库
import music
#两只老虎音乐音调音节数组
tune = ["C4:", "D", "E", "D",
        "C", "D", "E", "D",
        "E", "F", "G:8",
        "E", "F", "G:8",
        "G", "A:3", "G", "F:3", "E", "C",
        "G", "A:3", "G", "F:3", "E", "C",
        "C", "G", "C:8",
        "C", "G", "C:8"]

while True:
    #读取接入 pin8 的触摸模块的数字值
    state = pin8.read_digital()
    #数字值为 1 时，显示笑脸并响起音乐；反之显示哭脸并停止播发音乐
    if state == 1:
        display.show(Image.HAPPY)
        music.play(tune)
    else:
        display.show(Image.SAD)
        music.stop()
```

Micro:bit图形化编程



结果：引脚8接入触摸模块，引脚0接入无源蜂鸣器。当触摸模块输出高电平的时候，蜂鸣器会播放歌曲；当触摸模块输出低电平的时候，蜂鸣器会停止播放歌曲。

触摸按键模块

模块介绍：触摸模块是一个基于触摸检测的电容式点动型触摸开关模块。常态下，模块输出低电平；当用手指触摸相应位置时，模块会输出高电平。可以将模块安装在非金属材料如塑料、玻璃的表面，另外将薄薄的纸片（非金属）覆盖在模块的表面，只要触摸的位置正确，即可做成隐藏在墙壁、桌面等地方的按键。该模块可以让你免除常规按压型按键的烦恼。

工作电压：3.3V到5V

输出类型：数字信号

工作模式：4种

接口模式：pH2.0-3P

模块尺寸：35*26.3mm

模块重量：36g

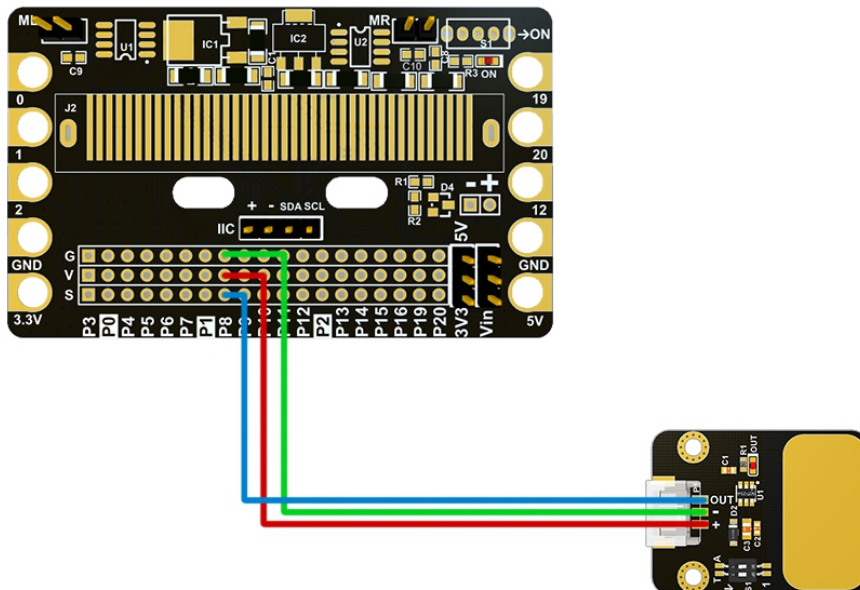
引脚定义：

OUT	输出
-	GND
+	VCC

Micro:bit

样例一

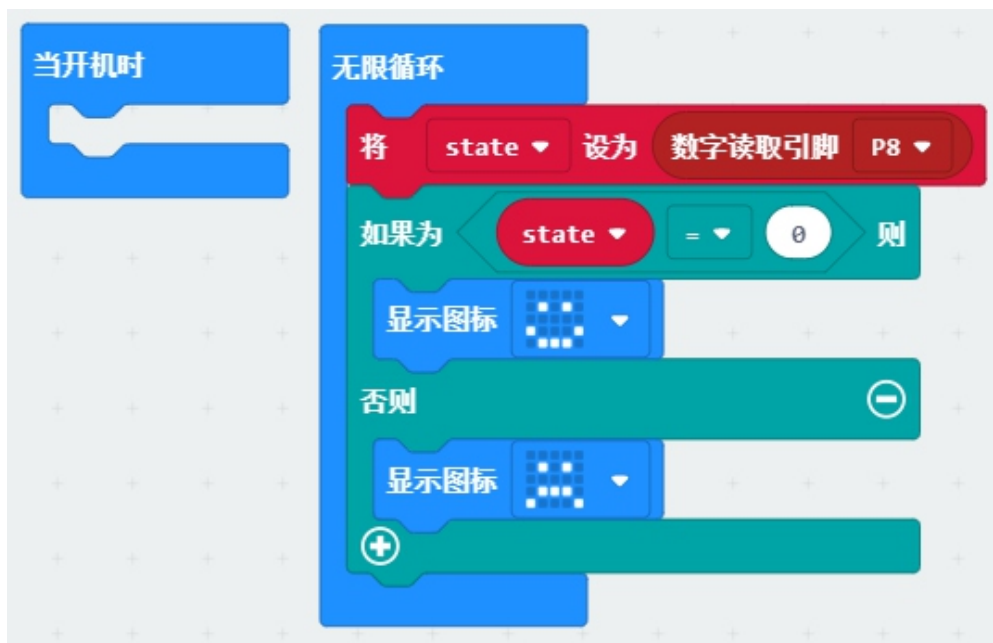
连接图



代码

```
from microbit import *  
#write your program:  
while True:  
    #读取接入 pin8 的触摸模块的数字值  
    state = pin8.read_digital()  
    #数字值为 0 时，显示笑脸；数字值为 1 时，显示哭脸  
    if state==0:  
        display.show(Image.HAPPY)  
    else:  
        display.show(Image.SAD)
```

Micro:bit图形化编程



结果：引脚8接入触摸模块。每当触摸模块输出高电平的时候，5*5点阵led显示苦脸，直到触摸模块输出低电平的时候，5*5点阵led显示笑脸。

补充：该模块上有个两格的拨码盘，拨动拨码盘，一共有4个组合，分别00、01、10、11，即触摸模块根据编码器的组合，实现4个触摸模式。00和01是每当触摸该模块时，会改变当前电平，两者之间的区别是，00是每当一通电时，该模块会先以输出高电平开始，而01是每当通电时，该模块会先以输出低电平开始；10是每当触摸该模块时，会输出低电平，反之输出高电平；11是每当触摸该模块时，会输出高电平，反之输出低电平。注意的是，由于模块上优格电容，所以每次修改模块的触摸模式后，断开电源让模块放放电哦！

循迹模块

模块介绍：循迹模块是红外发射管发射光线到路面，红外光遇到黑色时则被吸收，接收管没有接收到反射光，输出高电平。当红外光遇到其他颜色时，接收管接收到反射光，输出低电平。该模块在使用模拟输出的时候，可以当作灰度传感器，灰度传感器是模拟传感器，可以感知地面或桌面不同的颜色而产生相应的信号。

工作电压：3.3V到5V

输入类型：数字信号、模拟信号

接口模式：pH2.0-6P

模块尺寸：35*26.3mm

模块重量：51g

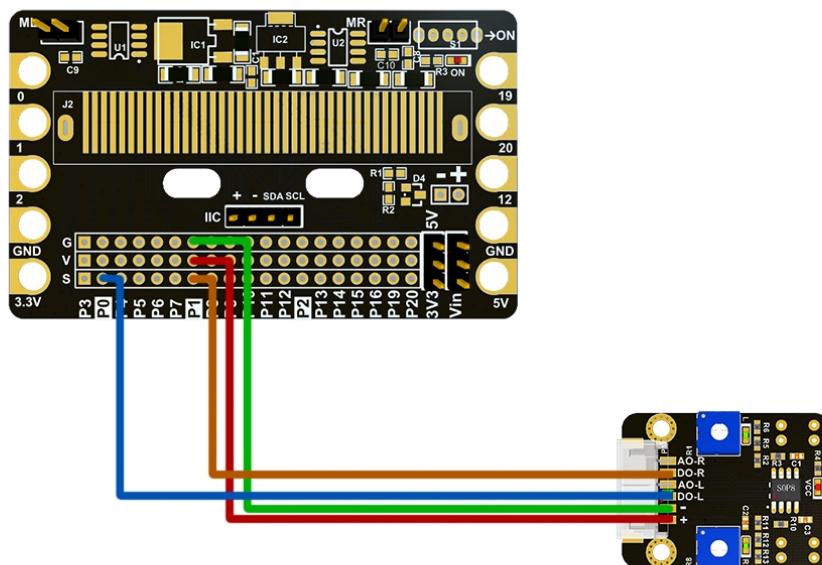
引脚定义：

AO-R	左边模拟输入
DO-R	右边数字输入
AO-L	左边模拟输入
DO-L	左边数字输入
-	GND
+	VCC

Micro:bit

样例一

连接图



代码

```
from microbit import *
#write your program:
while True:
    #left、right 两变量分别记录接入寻迹模块的 DO-L 和 DO-R 的 pin0 和 pin1 的数字值
    left = pin0.read_digital()
    right = pin1.read_digital()
    #pin0、pin1 的数字值为 1 时，显示交叉
    if left==1 and right==1:
        display.show(Image.NO)
    #pin0、pin1 的数字值为 0 时，显示对勾
    if left==0 and right==0:
        display.show(Image.YES)
    #pin0 的数字值为 0、pin1 的数字值为 1 时，显示笑脸
    if left==0 and right==1:
        display.show(Image.HAPPY)
    #pin0 的数字值为 1、pin1 的数字值为 0 时，显示苦脸
    if left==1 and right==0:
        display.show(Image.SAD)
```


Microbit图形化编程



结果：模块的DO-L接入引脚0，DO-R接入引脚1，打开串口窗口。当两边红外被阻挡时，接收不到信号，输出高电平，提示“stop”；当两边红外都没被阻挡时，接收到信号，提示“go”；当左边红外被阻挡时，左边接收不到信号，提示“left”；当右边红外被阻挡时，右边接收不到信号，提示“right”。

风扇模块

模块介绍：直流电动机是把直流电能转换成机械能的一类电机。最常见的类型依赖于磁场产生的力。几乎所有类型的直流电动机都有一些内部机制，无论是机电的还是电子的，以周期性地改变部分电动机的电流流向。大多数类型产生旋转运动；直线电机直接产生力和直线运动

工作电压：5V。

输入类型：数字信号、PWM

接口模式：pH2.0-4P

模块尺寸：35*26.3mm

模块重量：

引脚定义：

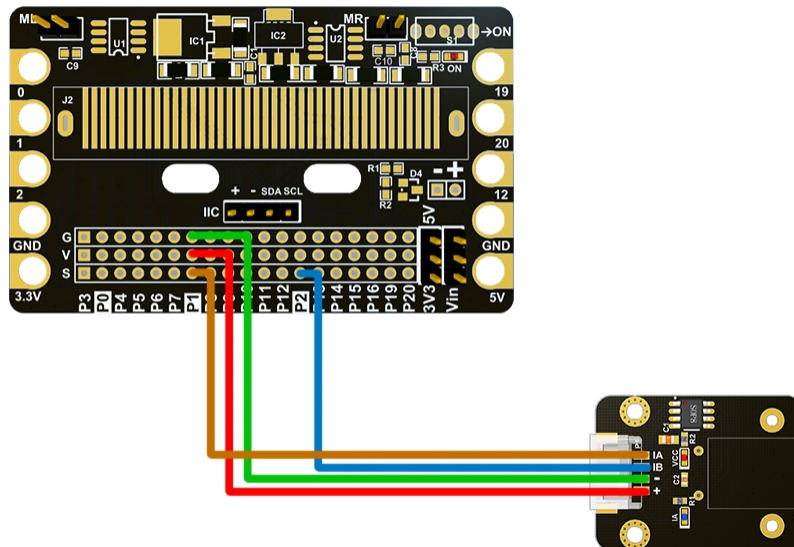
IA	正转数字输入
IB	反转数字输入
-	GND
+	VCC

控制方式

IA	IB	功能
0	0	停下
0	1	反向转
1	0	正向转

Micro:bit

样例一
连接图



代码

```
from microbit import *  
#write your program:  
while True:  
    #按下按键 A 时， 风扇正向转动  
    if button_a.is_pressed():  
        pin1.write_analog(1)  
        pin2.write_analog(0)  
    #按下按键 B 时， 风扇反向转动  
    elif button_b.is_pressed():  
        pin1.write_analog(0)  
        pin2.write_analog(1)  
    #同时按下按键 A B 时， 风扇停止转动  
    elif button_a.is_pressed() and button_b.is_pressed():  
        pin1.write_analog(0)  
        pin2.write_analog(0)
```

Micro:bit图形化编程



结果：IB接口接入引脚2，IA接口接入引脚1。当按键A按下时，电机正向旋转；当按键B按下时，反向旋转；同时按下A、B按键时，电机停止旋转。

红外接收模块

模块介绍：外接收管就是接收红外光的电子器件。比如我们电视机，空调这些家电，其实它们都需要用到红外接收管。我们都知道遥控器发射出来的都是红外光，电视机上势必要有红外接收管，才能接收到遥控器发过来的红外信号。

工作电压：3.3V到5V

输出类型：数字信号。

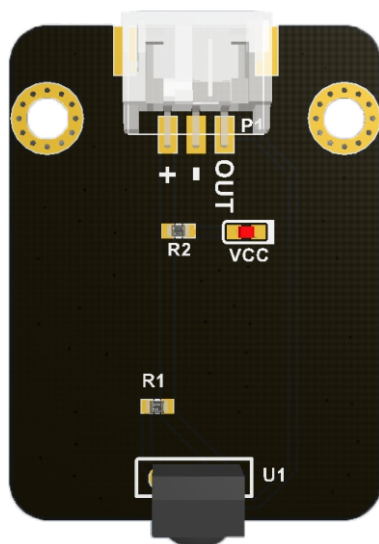
接口模式：pH2.0-3P

模块尺寸：35*26.3mm

模块重量：40g

引脚定义：

OUT	输出
-	GND
+	VCC



Micro:bit

样例一

代码

```
from microbit import *
#导入红外接收库
import necir
#对接收到的信息进行输出
def cb(addr,cmd):
    print('addr=',hex(addr))
    print('cmd=',hex(cmd))
#初始化红外接收，红外接收接入 pin16
necir.init(16)
#将读取的地址值和命令传入回调函数中显示出来
necir.read(cb)

while True:
    pass
```

根据串口输出的16进制数据，我们可以对代码进行修改

```
from microbit import *
#导入红外接收库
import necir
#对接收到的地址值和命令进行输出和处理
def cb(addr,cmd):
    print('addr=',hex(addr))
    print('cmd=',hex(cmd))
    if addr == 0xff00:
        if cmd == 0xef10:
            display.show(Image.HEART)
        if cmd == 0xee11:
            display.show(Image.HEART_SMALL)

#初始化红外接收，红外接收接入 pin16 和调用处理数据函数
necir.init(16,cb)
```

结果：引脚16接入红外接收模块，当按下“1”时，5*5点阵显示大心；当按下“2”时，5*5点阵显示小心。

麦克风模块 (声音传感器模块)

模块介绍：输出模块，数字模拟接口。其作用相当于话筒。用来接收声波，显示振动图像，但不能对噪声的强度进行测量。声波使话筒内的驻极体薄膜振动，导致电容的变化，从而产生与之对应变化的微小电压，经过数字/模拟转换被数据采集器接收，并传给控制板。

AO模拟量实时输出麦克风的接收到声量信号

DO当声音强度到达某个阈值，则输出低电平信号（可通过模块上的R2来调节），由于输出低电平的时间非常的短，为了捕捉到有用的信息使用信息过滤或中断来呈现，此处用信息过滤。

工作电压：3.3V到5V

输出类型：数字输出、模拟输出。

接口模式：pH2.0-4P

模块尺寸：35*26.3mm

模块重量：50g

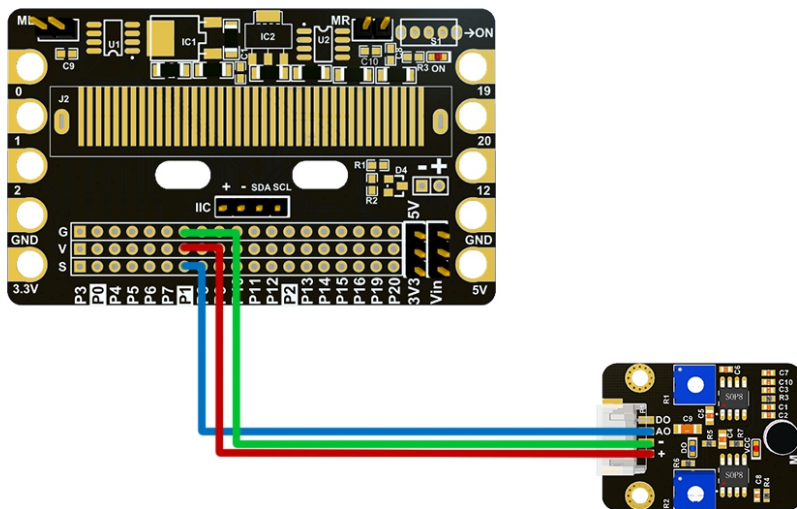
引脚定义：

DO	数字输出
AO	模拟输出
-	GND
+	VCC

Micro:bit

样例一

连接图



代码

```
from microbit import *  
#write your program:  
while True:  
    #读取接入 pin1 的麦克风 DO 的数字值  
    state = pin1.read_digital()  
    #滚动播放数字值  
    display.scroll(state)
```

Mirobit图形化编程



结果：代码编程中模块的D0脚接入PIN1，屏幕显示模块当前数字数值。图形化编程中模块的A0脚接入PIN1，打开窗口，查看模块在不同音量下的模拟数值变化。

电位器模块

模块介绍：电位器是一种三端电阻元件，其电阻可根据一定的规则变化。它通常由一个电阻元件和一个可移动电刷组成。当电刷沿着元件移动时，在端子处会产生相对于其移动距离的电阻或电压。

工作电压：3.3V到5V

输出类型：获得电位器开关的数字输出和获取电位器值（0-1023）的模拟输出。

接口模式：pH2.0-4P

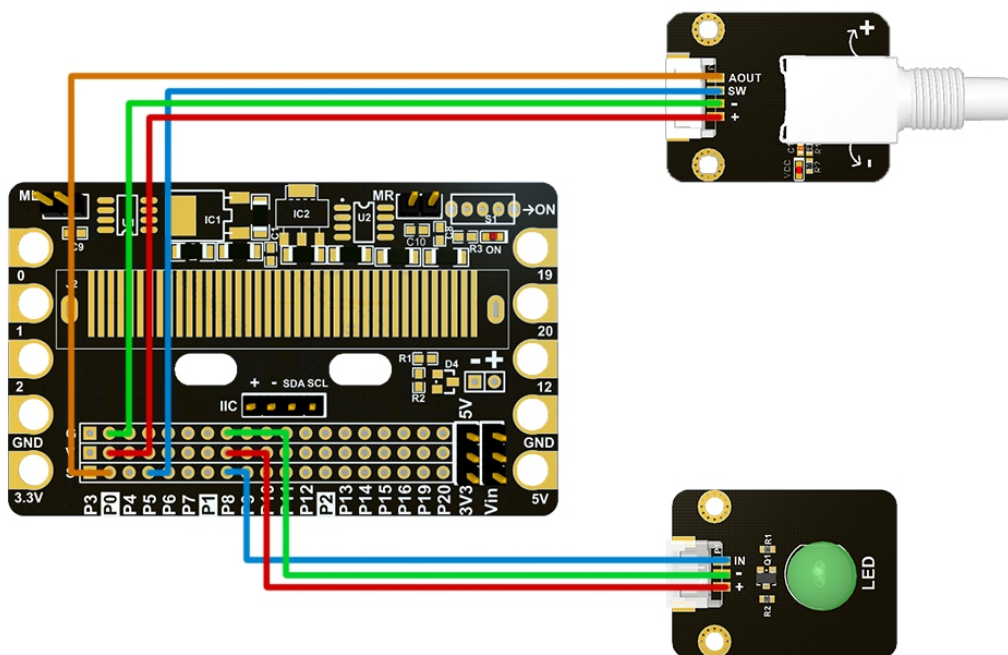
模块尺寸：35*26.3mm

模块重量：93g

引脚定义：

AUTO	模拟输出
SW	数字输出
-	GND
+	VCC

Micro:bit
样例一
连接图



代码

```
from microbit import *
#write your program:
while True:
    #读取接入 pin5 的电位器 sw 开关的数字值
    state = pin5.read_digital()
    #当数字值为 0 时，显示笑脸，读取接入 pin0 的电位器 AOUT 的模拟值，将读到的模拟值映射 10 到 1023 并写入接入 pin8 的 LED 灯
    if state==0:
        display.show(Image.HAPPY)
        num = pin0.read_analog()
        pin8.write_analog(num*0.99+10)
    #当数字值为 1 时，显示哭脸，写入接入 pin8 的 LED 灯数字值 0
    else:
        display.show(Image.SAD)
        pin8.write_digital(0)
```

Microbit图形化编程



结果：引脚5接入电位器的数字接口，引脚0接入电位器的模拟接口，引脚8接入led灯。Led灯的亮度随电位器的扭动而变化。Micro: bit板上的5*5点阵图随电位器的开关而改变。

超声波模块

模块介绍：用来测量距离，通过发送和接收超声波，测量声音从物体上反弹并返回传感器所需的时间，利用时间差和声音传播速度，计算模块到前方障碍物的距离。

工作电压：3.3V或5V

最小测量距离测量：2cm

最大测量距离测量：350cm

类型：Echo是输入数字信号，Trig是输出数字信号。

接口模式：pH2.0-4P

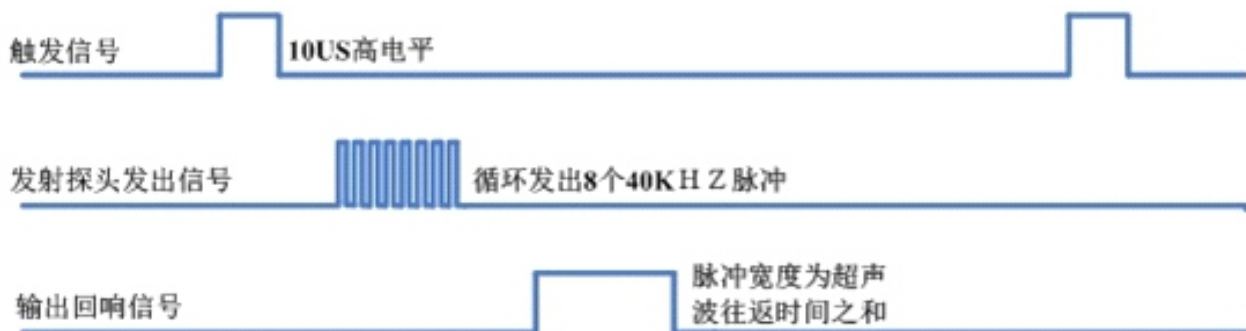
模块尺寸：35*26.3mm

模块重量：86g

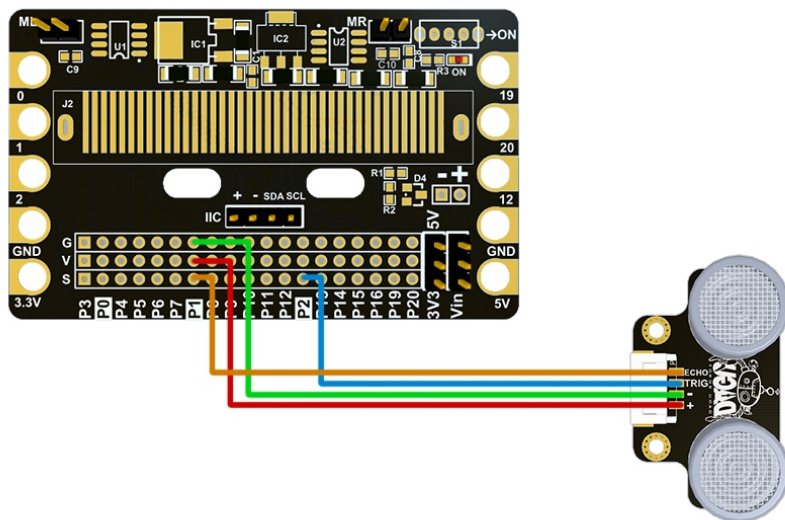
引脚定义：

Echo	接收输入
Trig	发送输出
-	GND
+	VCC

工作原理：向Trig管脚输入一个10uS以上的高电平，可触发模块测距。当测距结束时，Echo管脚会输出一个高电平，电平宽度为超声波往返时间之和。



Micro:bit 样例一 连线图



代码

```
from microbit import *  
#导入超声波库  
import urm10  
while True:  
    #urm10.read(ECHO,TRIG)读取当前超声波与障碍物的距离  
    a = urm10.read(2,1)  
    sleep(100)  
    #滚动显示距离值  
    display.scroll(a)  
    display.scroll("cm")
```

Mixly图形化编程

在图形化编程之前需要添加扩展，点击扩展，在“搜索或输入项目网址...”处搜索“Sonar”点击下图，并等待页面刷新完毕。



接下来图形化编程



结果：引脚1接入Trig，引脚2接入Echo。打开串口，移动超声波测距模块前的障碍物，输出到串口的数据发生变化。