






# 《信息系统与社会》micro:bit案例详解

## 一、读音

[micro](#) - 百度翻译

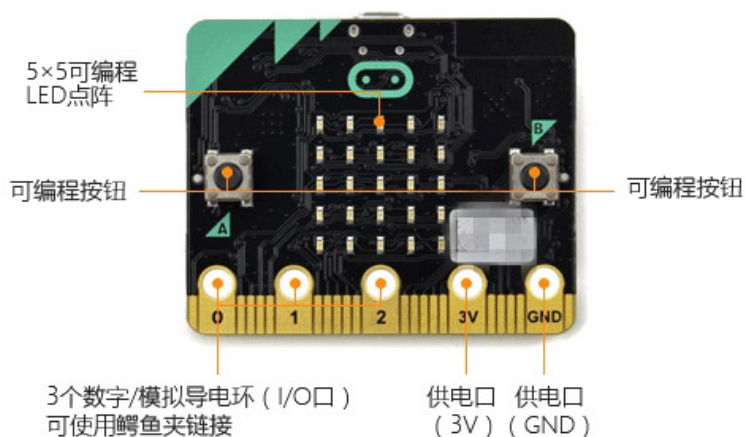
micro 英[ˈmaɪkrəʊ]   美[ˈmaɪkroʊ]  

adj. 微观的;

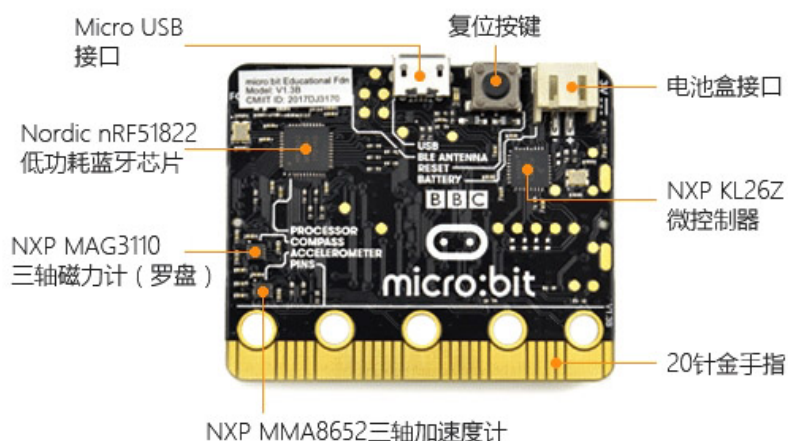
[例句] You haven't lived until you've used their new **micro** system.   
要是没用过他们新的微系统，这辈子就算白活了。

## 二、Micro:bit主控板

### 正面

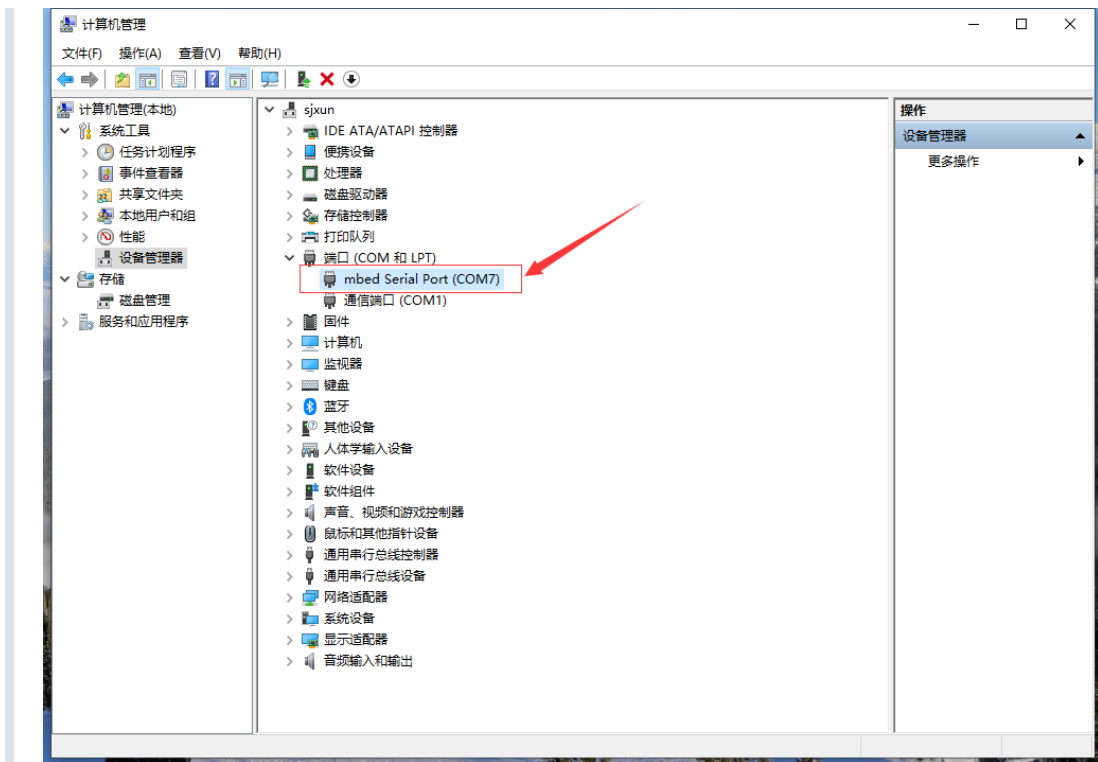


### 反面



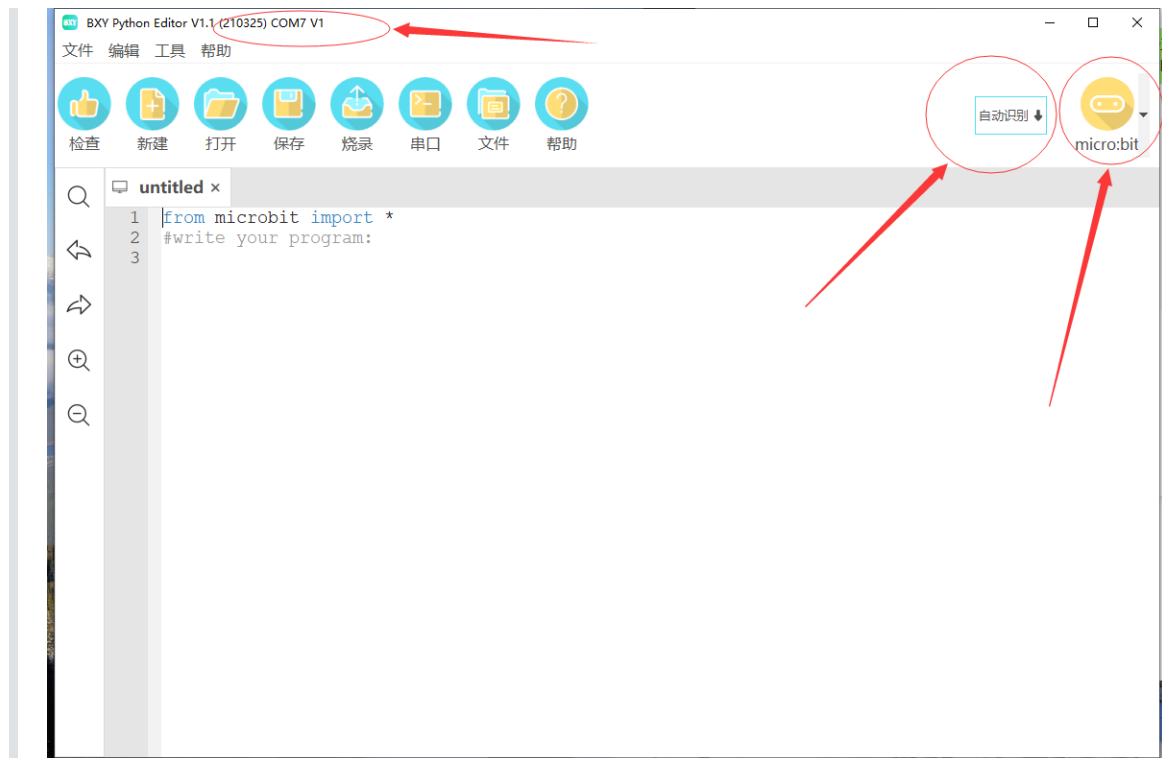
## 三、安装软件、连接Micro:bit

1. 下载BXY软件并安装。网址: <https://bxy.dfrobot.com.cn/>
2. 打开电脑“设备管理器”，展开“端口列表”



3.连接Micro:bit到电脑USB接口，观察“端口列表”变化

4.打开BXY软件，确认连接正常



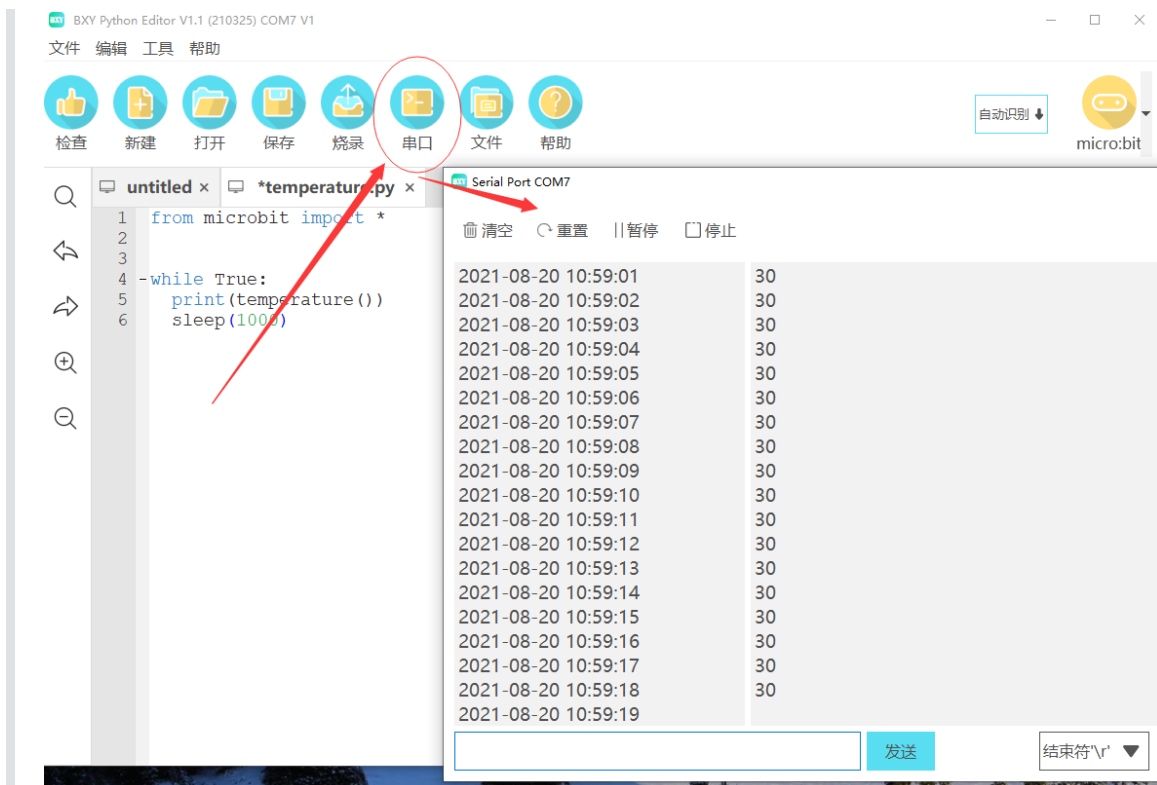
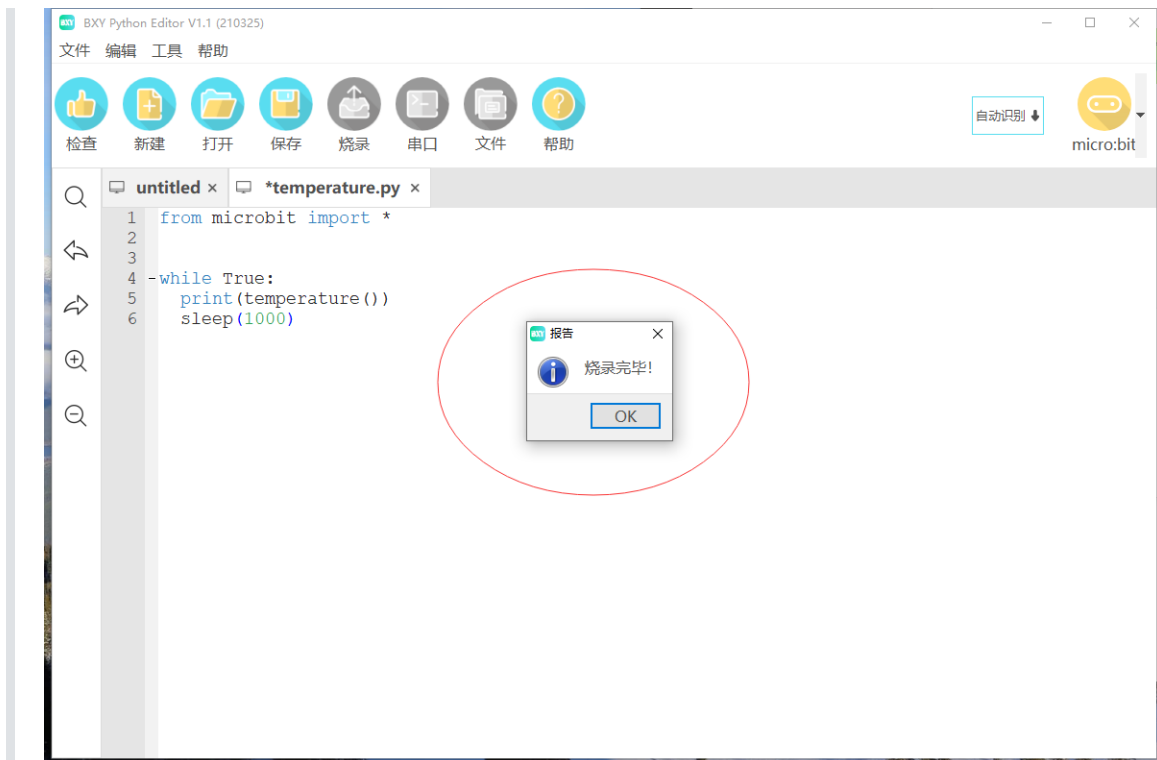
## 四、《信息系统与社会》案例详解

### 1.案例一：温度计（2.4.3传感器信息的获取）

①.新建文档，输入下列代码并“保存”。然后点击“烧录”，出现“烧录完毕”，打开串口工具查看数据：

```
#sample1_1.py
from microbit import *

while True:
    print(temperature())
    sleep(500)
```



②.用手指捂住，观察串口监视器数值变化。（也可以使用通用串口工具查看信息）

③.简单改进：

- display.scroll(): 滚动显示“信息”到led阵列
- MicroPython基本语法：<https://bxy.dfrobot.com.cn/mciroPythonJiChu>

- Micro:bit教程: <https://bxy.dfrobot.com.cn/microbit>

```
#sample1_2.py
from microbit import *

while True:
    print(temperature())
    display.scroll(temperature())
    sleep(1000)
```

## 2.案例二：Python读取串口数据（2.4.3传感器信息的获取）

案例一是通过BXY这个软件给Micro:bit写程序，程序在Micro:bit上运行，并完成读取温度，输出到串口。Python IDE想要得到这些数据，可以使用“serial”库连接到串口然后读取信息。

### ①使用serial库读取串口数据

```
#sample2_1.py
import serial
ser=serial.Serial()
#设置通信波特率，需要与Micro:bit中设定的通信速率一致
ser.baudrate = 115200
#设置串口号
ser.port='COM7'
ser.open()
while True:
    print(ser.readline())
```

打开Python IDE 输入上面代码，运行，查看效果

### ②将串口数据保存到文本文件

```
#sample2_2.py
import serial
ser=serial.Serial()
ser.baudrate = 115200
ser.port='COM7'
ser.open()
f=open('microbit.txt','wb')
a=20
while a>0:
    a-=1
    line=ser.readline()
    f.write(line)
    print(line)

f.close()
ser.close()
```

**关键点：**serial模块需要安装、串口号设置要正确、调试时，需要关闭BXY（或断开BXY和Micro:bit连接）

### 3.案例三：用计算机控制Micro:bit的LED阵列（2.4.5信息系统中的控制）

①打开BXY软件，输入如下代码，并烧录

```
#sample3_1.py
from microbit import *
while True:
    if uart.any():
        incoming=str(uart.readall(),"UTF-8")
        incoming=incoming.strip('\n')
        if incoming == 'H':
            display.show(Image.HAPPY)
            print("I am happy")
        elif incoming == 'S':
            display.show(Image.SAD)
            print("I am sad")
        else:
            print("err")
```

- 使用BXY的串口监视器分别发送'H'，'S'，和其他字符，关注Micro:bit上的LED点阵图形变化。



- 通过Python shell模式与Micro:bit交互

关闭BXY软件，打开Python IDE 在Shell模式分别输入以下代码：

```
import serial
ser=serial.Serial()
ser.baudrate=115200
ser.port='COM7'
#串口号以实际值为准
ser.open()
```

拿起Micro:bit，继续在Python Shell中输入以下指令，并观察Micro:bit上LED点阵图案变化

```
ser.write('H'.encode())
```

可以看到阵列显示‘高兴’的表情

```
ser.write('S'.encode())
```

可以看到阵列显示‘难过’的表情

## ②用Python编写程序，实现更多功能（计算机控制）

- 实现LED阵列有规律地切换表情

```
#sample3_2.py
import serial,time
ser=serial.Serial()
ser.baudrate=115200
ser.port='COM7'
ser.open()
while True:
    time.sleep(1)
    ser.write('H'.encode())
    time.sleep(1)
    ser.write('S'.encode())
```

- 实现交互控制LED阵列

```
#sample3_3.py
import serial,time
ser=serial.Serial()
ser.baudrate=115200
ser.port='COM7'
ser.open()
while True:
    name=input()
    ser.write(name.encode())
    line=ser.readline()
    print(line.strip().decode())
```

## ③创意作品（案例）

- 作品功能：程序运行时，随机出10道100以内加减法的数学题。每出现一道题都会等待用户输入，用户回答正确则连接计算机的Micro:bit显示‘高兴’的图案，用户回答错误则显示‘难过’的表情，10道题答完后，屏幕显示所得分数。
- Micro:bit代码：

```
#sample3_4.py
from microbit import *
while True:
    if uart.any():
        incoming=str(uart.readall(),"UTF-8")
        incoming=incoming.strip('\n')
        if incoming == 'H':
            display.show(Image.HAPPY)
            print("I am happy")
        elif incoming == 'S':
            display.show(Image.SAD)
            print("I am sad")
```

```
else:
    for i in range(5):
        display.scroll(incoming)
    print("Over")
```

- Python IDE 中输入代码:

```
#sample3_5.py
import serial,time,random
ser=serial.Serial()
ser.baudrate=115200
ser.port='COM7'
ser.open()
score=0;num=10
operator=['+', '-']
for i in range(num):
    t1=random.randint(1,100)
    t2=random.randint(1,100)
    if t1<t2:
        t1,t2=t2,t1
    op=random.choice(operator)
    print(i+1,"、",t1,op,t2,"=")
    print("请输入答案: ")
    u_ans=int(input())
    if op=='+' and u_ans==t1+t2:
        ser.write('H'.encode())
        score+=1
        print("回答正确! 太棒了! ")
    elif op=='-' and u_ans==t1-t2:
        ser.write('H'.encode())
        score+=1
        print("回答正确! 太棒了! ")
    else:
        ser.write('S'.encode())
        print("很遗憾, 回答错误! ")
time.sleep(1)
ser.write(str(score).encode())
print("您本次答题得分是: ",score)
```

测试看下效果吧