

LoveMeOrNot *on micro:bit*

(是否爱我)

作者：李明远 沈烨

摘要：本项目通过 microbit 的按钮交互，实现在相亲饭局后的心意即时传递，决定两个人是否可以继续发展。除了本项目自身的实现之外，报告中还会探讨一些关于人机交互功能的实现问题。

一、选题及创意介绍

本选题来源于李明远同学的神奇脑洞。刚开始我们考虑到 microbit 自带的强大的交互功能，希望完全利用其功能实现一个感光的交互式闹钟，但是由于 microbit 音量有限且缺乏趣味性遂作罢。后来我们想到这样一块可爱的芯片如果可以在日常生活中实现更为**有趣**的实用性功能，也许更符合其设计初衷。

由于 microbit 可以实现简单的信息存储以及交互，于是我们设想将 microbit 用于相亲场景，以打破相亲后难以开口表达心意的尴尬局面。如今的年轻人都比较含蓄内敛，如果双方互存好感，往往因为害怕被拒绝而难以开口，最后一段缘分就此无疾而终；如果双方都没有好感却不好意思伤人心，磨磨唧唧地拖很久，还不如直截了当解决；如果一方有好感而一方没有，如果没好感的先拒绝也就罢了，有好感的好不容易鼓足勇气说出口却被残忍拒绝，最终也无法体面结束。最终，相亲在这种局面下成为了年轻人最不想参与的事情之一。所以利用 microbit 实现无痛相亲是很有必要的。

二、设计方案和硬件连接

实际上相亲尴尬局面的根源是两个人在不知道对方心意的情况下存在信息差，我们可以利用 microbit 的存储功能打破信息差。虽然两个人的输入顺序存在先后，但是这一先后关系由于放在了 microbit 这一黑箱中而不被显示出来，就打破了人们表达想法的心理障碍，从而让双方只需通过简单的按键就可以表达自己是否希望进行下一次约会的心意。

Microbit 上有 A、B 两个按键，选 A 表示 Yes，即同意进行下一次约会；选 B 表示 No，即不同意进行下一次约会。首先双方各有一次防误触的机会，即第一次按键是无效的，仅仅是判断本轮次已经按键，并且同时会传达出这一按键的性质，然后进入第二次按键，这次按键会被记录并最终决定了约会是否继续。第一个人完成两次按键后，将 microbit 交由第二个人，第二个人两次按键后，microbit 通过逻辑判断，输出两个结果的“与”结果，最后显示在屏幕上并发出相应的声音以提示双方。（视频中为了节省时间简化了第一次按键防误触的过程，即一次按键即视为有效并做出判断，与本文提交的完整代码不同。）

本功能的实现不需要额外的硬件设施，只需将程序导入 microbit 使之运行即可。如果可以实现两个终端的通信功能，则可能需要额外硬件。

三、实现方案及代码分析

1. 程序启动：由于 logo 可以接受到触屏信息，可以通过这一方法启动程序。
2. 双方情感的输入：microbit 自带按键交互功能，可以实现简单的逻辑判断，即 A 表示 Yes 而 B 表示 No，从而实现第一步，即情感判断的输入。难点：代码是立即执行的，所以必须通过无限循环判断是否按键，如未按键则继续等待。
3. 输入数据的处理：将输入处理成 True 和 False 的布尔值，就可以通过 python 自带的与判断得到最终结果。
4. 最终结果的输出：Microbit 有显示屏和扬声器，通过显示屏 display 的方法和 microbit 自带的 music 模块实现。Display 可以直接显示内置图片，music 可以直接播放内置音乐。这里我们想要寻找自己导入音频播放的方法，然而并没有成功。
5. 运行时间的调整：在程序进行过程中用 sleep 方法调整运行节奏，避免因为程序运行过快导致没有反应过来即显示结果。

以下为完整带注释的代码：

```
from microbit import *
import music
#导入所需模块

while True:
    if pin_logo.is_touched():#触碰 logo 则开始游戏
        display.show(Image.HAPPY)
        music.play(music.POWER_UP)
        sleep(500)
        display.show(1, delay=1000)
        #显示轮到第一个
        player_1, player_2 = None, None
        #设置两个玩家的初始状态，当状态为 True 时表示有好感，False 表示无感
        loop_1, loop_2 = True, False
```

```
while loop_1:#通过无穷循环等待 A 或者 B 被按下
    if button_a.is_pressed():
        music.play(music.BA_DING)#每次按键均会有提示音响起
        display.show(Image.YES, delay=2000)
        loop_1 = False
        loop_2 = True
        #如果 A 被第一次按下，不再等待，进入第二重循环
```

```
while loop_2:
    if button_a.is_pressed():
        loop_2 = False
        music.play(music.BA_DING)
        display.show(Image.YES)
        player_1 = True
        #如果前后操作均为选 A，则输出结果为 Yes
```

```
elif button_b.is_pressed():
    loop_2 = False
    music.play(music.BA_DING)
    display.show(Image.NO)
    player_1 = False
    #如果第二次选 B，则输出结果为 No
```

```
elif button_b.is_pressed():
    loop_1 = False
    loop_2 = True
    music.play(music.BA_DING)
    display.show(Image.NO)
    #如果 B 被第一次按下，不再等待，进入第二重循环
```

```
while loop_2:
    if button_b.is_pressed():
        loop_2 = False
        music.play(music.BA_DING)
        display.show(Image.NO)
        player_1 = False
        #如果前后操作均为选 B，则输出结果为 No
    elif button_a.is_pressed():
        loop_2 = False
        music.play(music.BA_DING)
        display.show(Image.YES)
        player_1 = True
        #如果第二次选 A，则输出结果为 Yes
```

```
sleep(1000)
display.show(2, delay=1000)
loop_1 = True
#显示轮到第二个人
```

```
while loop_1:
    if button_a.is_pressed():
        loop_1 = False
        loop_2 = True
        music.play(music.BA_DING)
        display.show(Image.YES)
```

```
while loop_2:
    if button_a.is_pressed():
        loop_2 = False
```

```
        music.play(music.BA_DING)
        display.show(Image.YES)
        player_2 = True
    elif button_b.is_pressed():
        loop_2 = False
        music.play(music.BA_DING)
        display.show(Image.NO)
        player_2 = False
```

```
elif button_b.is_pressed():
    loop_1 = False
    loop_2 = True
    music.play(music.BA_DING)
    display.show(Image.NO)
```

```
while loop_2:
    if button_b.is_pressed():
        loop_2 = False
        display.show(Image.NO)
        player_2 = False
    elif button_a.is_pressed():
        loop_2 = False
        display.show(Image.YES)
        player_2 = True
#判断方式与第一个人完全相同
```

```
if all([player_1, player_2]):
    sleep(1000)
    display.show(Image.HEART)
    music.play(music.WEDDING)
#如果两人均表示 Yes，则输出爱心并播放婚礼进行曲，可以进行下一次约会
```

```
else:
    sleep(1000)
    display.show(Image.SAD)
    music.play(music.WAWAWAWAA)
#一旦有人表示 No，则输出哭脸并播放悲伤音乐，无法进行下一次约会
```

四、后续工作展望

现在我们设计的场景中只用到了一个 microbit，需要两个人在同一芯片上先后操作，而在我们构想的人手一个 microbit 的世界，如果可以实现多个终端同时操作，则更为方便。我们了解到 microbit 带有通信功能，通过无线电的信息传递可以实现两个 microbit 的信息共享，

只需要两人都分别选择“Yes”或“No”之后再稍作等待即可在自己的设备上看到结果，这个过程甚至可以不用面对面完成，让该项目的实现更为方便。

这个项目不仅仅只局限于相亲场景，还可以用于多个人表达意的场景，比如说用 microbit 实现投票系统，使其拓展成为一个多人平等贡献信息的平台。

这个项目的构想是在一个架空的大家都拥有 microbit 的世界中实现的。如果将其运用于现实世界中其实存在一个问题，就是需要有一方了解这一产品然后再将其向另一方推广，而这一推广的过程又是一个打破信息差的令人尴尬的场景。也许对方会认为提出使用这一方法的人有不愿意表露真实心意的心思，在相亲开始前印象就已经大打折扣。所以如果能够在社会上建立起这样一个所有人都知道该设计并且使用这种方法的氛围，那么这一方法才是行之有效的。

我们现在看来，当代年轻人确实越来越多地将自己的问题求助于计算机，所以也许将来这样方法的普及极有可能是大势所趋。即使没有人手一个 microbit，我们用手机也可以实现这样的功能，或许那时，人们的生活方式会因为思维观念的变化产生极大的改变。这种寻求于计算机的极简方法，会成为人类生活状态的归宿吗？这究竟是否有利于社会的发展？这些问题值得深思。

五、小组分工合作

李明远：创意提供，代码编写，视频剪辑，资料整理

沈烨：实习报告撰写

共同完成：代码修改完善，视频构思及拍摄