

“私家侦探” 编程设计

陈开元 莫昊宇

摘要

可曾想过，一款看似朴实、单调的单片机内竟暗藏玄机！你的大致行程方位将被放在车篮里、毫不起眼的 **MicroBit** 一一记录在案。它利用你的水平运动加速度，悄无声息地推算速度、位移，稍有不慎，旋即中招。当然，那些表面上的功能也是值得一用的。指南针可以助你迷途知返；计步器可以监督你打卡锻炼；计时器则更应是家中常备。一旦有了它，心机巧妙的你便可借其表面功能之名，行内部追踪之实。想必这次，你的那个他在劫难逃了吧.....

一、选题及创意介绍

本款产品的创意核心在于朴实、基本的 **MicroBit** 自带功能之下潜藏的行程追踪。我们利用 **A** 和 **B** 的按键，为用户提供了三种不同的功能选择：**A+B** 为计步器；**2A+B** 为秒表；**3A+B** 为指南针。

但实际上，真正的核心在于触摸 **logo** 之后再按 **A** 键，将开启“私家侦探”模式。这时，它仍然拥有指南针作为表面掩护，但实际后台已经开始更为隐秘的计算，将行踪一一记录，交给幕后黑手。

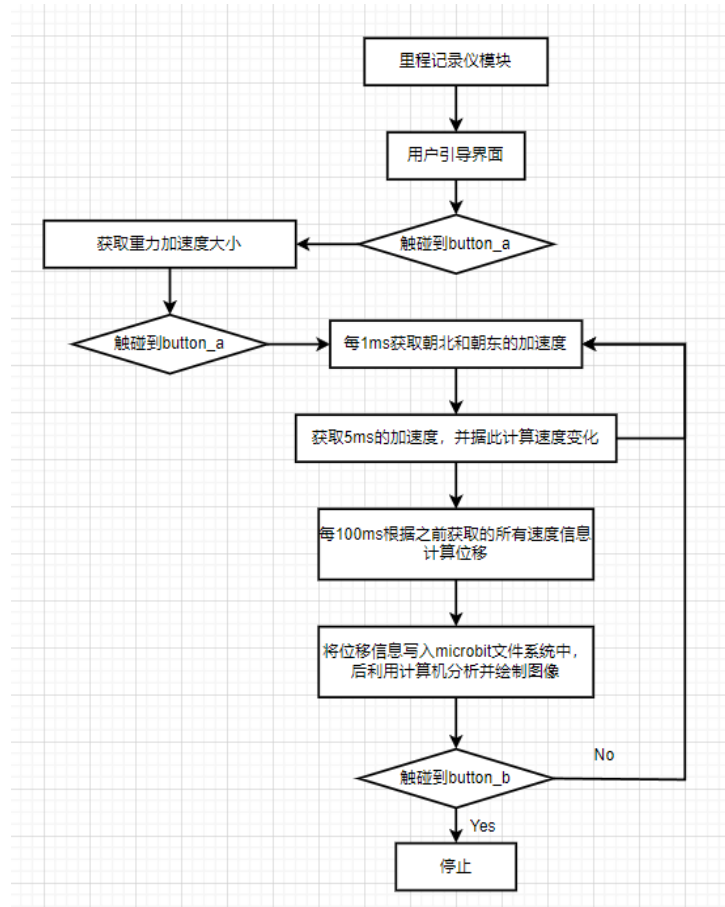
二、设计方案和硬件连接

默认起始坐标和速度为零，且运动大致保持在 **Accelerator** 的 **y** 轴方向（无左右晃动）。在重力加速度校正后，通过记录行程中的加速度变化，进行降噪平滑、矢量加减、积分计算等一系列措施后，返回当前的位移值。最终，将位移数据通过 **USB** 接口传输至电脑，本地运行 **MBit.py** 文件实现追踪轨迹的动态可视化。

所需硬件：**MicroBit** 单片机一块，需水平放置，随身携带；移动电源；安卓口数据传输线；电脑一台，用于记录数据后的轨迹动态可视化。

三、实现方案及代码分析

私家侦探追踪器



程序流程图如上。

程序主要通过 `accelerometer` 和 `compass` 模块进行分析计算，从而实现里程记录。

重力加速度的获取：保持 `microbit` 为静止状态，在一秒内测量一千次加速度大小并求得平均值，即得重力加速度 `abar`。

```
# 重力加速度的获取
if button_a.is_pressed():
    # 提示信息, 保持microbit 为静止状态
    display.show('gravity')
    sleep(1000)
```

```

# 多次测量
atotal = 0
for _ in range(int(1 / dt)):
    atotal += accelerometer.get_strength() # 测量加速度大小
    sleep(dt * 1000) # dt = 0.001(s)
# 求平均值
abar = atotal / int(1 / dt)
# 提示信息
display.show('done')

```

朝北和朝东的加速度、速度、位移的获取：根据物理知识，若不考虑 microbit 在竖直方向的加速度（这是合理的，因为使用时一般只有水平面的加速度），将测量到的加速度和重力加速度进行矢量减法即可得到水平面加速度的大小，再根据指南针计算 microbit 朝向和正北方向的夹角，根据几何知识即可简单的计算出朝北和朝东方向的加速度大小。得到一系列加速度后进行积分（散点列的近似积分计算）得到速度，再积分一次得到位移。

```

# 朝北和朝东的加速度、速度、位移的获取
aY=0
for _ in range(int(dT / dt)): # dT = 0.01(s), dt = 0.001(s)
    # microbit 朝向和正北方的夹角
    direction = math.radians(compass.heading())
    # 水平面加速的大小的平方 a_squared
    asq = (accelerometer.get_strength())** 2 - abar ** 2
    # 这一步操作是为取平均做准备
    a2 += asq
    aY += a_y
    cosum += math.cos(direction)
    sinum += math.sin(direction)
    sleep(dt * 1000)
    # 如果 a_2 为负数，则舍弃
if a2 <= 0:
    aH = 0
    # 否则，可计算
else:
    aH = (a2 / int(dT / dt)) ** (1 / 2) // 10*10
    # 水平方向加速度；扫底除，减小残差的影响。
if aY <= 0:
    axy = -aH * 0.001 * g
else:
    axy = -aH * 0.001 * g
    # 朝东、朝北方向速度
vN += axy * cosum / int(dT / dt) * dT

```

```

vE += axy * sinum / int(dT / dt) * dT
    # 朝东、朝北方向位移
N += vN * dT
E += vE * dT

```

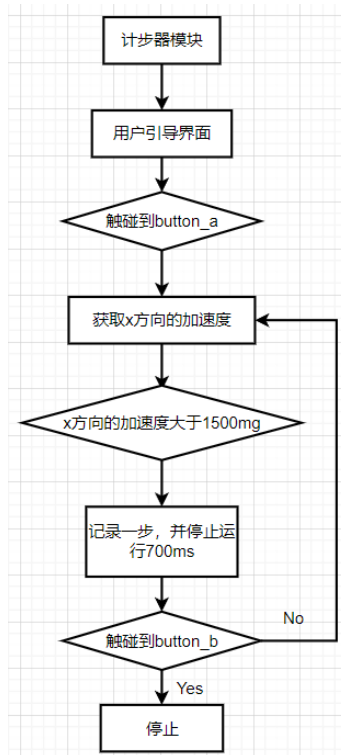
重要的是将位移数据写入文件中，以便之后可以由计算机进行分析并画图。

```

with open("position.txt", "w", encoding="utf8") as f:
    while True:
        # ... 获取数据
        # N 是朝北方向的位移, E 是朝东方向的位移
        f.write(str(N) + '/' + str(E))
        f.write("\n")

```

计步器



程序流程图如上。

程序主要通过 `accelerometer` 模块来实现计步。人走路时，起脚的一瞬间会有一个标志性的加速度（约为 1.5 个重力加速度），这个加速度在静息状态下是很难达到的；并且人走路的步频是相对固定的，即使不同人会有差别，此差异也是可以控制的。将 `microbit` 固定在鞋子的外侧（x 方向朝向前方），如果监测到大于等于

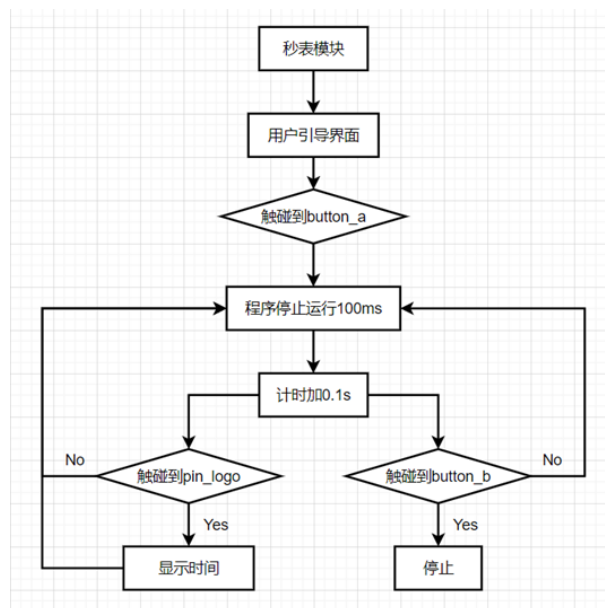
1.5g 的加速度（g 表示重力加速度，约为 $9.8m/s^2$ ），则记一次步数。但是由于 MicroBit 检测时间间隔是很短的，记完一次步数后则需要停止运行一段时间以等待下一步的到来。

```
# 记录走路的步数
if accelerometer.get_x() > 1500:
    count += 1
# 停止运行一段时间是必要的
sleep(700)
```

注意：此程序只用于监测走路时的步数，若要记录跑步时的速度，则需要测量跑步时足部的加速度以及步频，并且需要进一步调整监测的内容以适应跑步时大幅度的体态变化。

程序还配有一些用户接口，可以随时查看已走过的步数。

秒表



程序流程图如上。

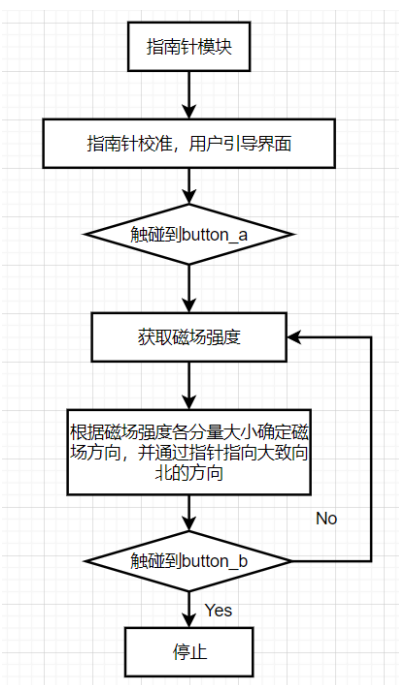
程序主要通过 sleep 函数实现计时功能，并通过 Image.ALL_CLOCKS 来实现计时可视化：认为除 sleep 函数外的所有操作时间可以忽略不计，则程序停止运行的时间

即可看作已经经过的时间。用一个指针标志量来显示时间，并通过 microbit 内置的时针图标展示给用户。

```
# 指针标志量
pointer = int(time / 5) % 12
display.show(Image.ALL_CLOCKS[pointer])
```

当然程序还配有一些音乐和引导性的文字以使得 microbit 更具有交互性。

指南针



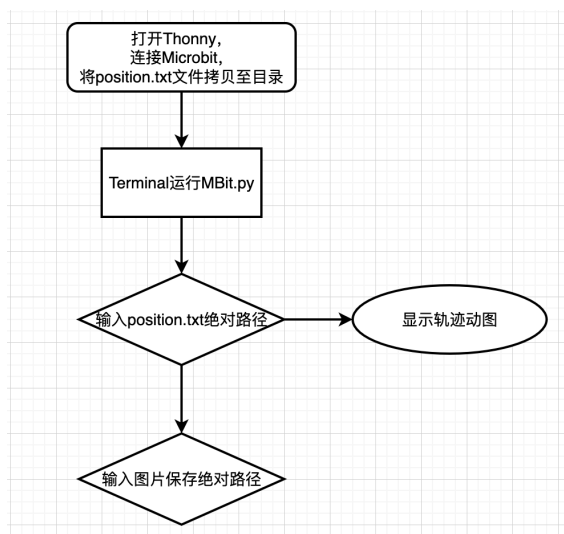
程序流程图如上。

程序主要通过 compass 模块实现监测磁场方向功能，并通过 Image.ALL_CLOCKS 来实现可视化：使用指针标志变量来操控 LED 屏上显示的指针方向，使指针一直保持指向大致为北的方向。

```
# 指针标志量
needle = ((15 - compass.heading()) // 30) % 12
display.show(Image.ALL_CLOCKS[needle])
```

行程轨迹读出（本地）

行程轨迹的读出方法如下流程图所示：



四、后续工作展望

首先，里程记录仪在实际使用时会出现一些误差。除了磁力传感器和加速度传感器的灵敏度限制，还受到数据处理方法的影响。如果能在处理数据的时候通过在更大时间范围的数据间进行分析并对数据进行更好的降噪操作，得到的结果会更加稳定、也会更加精确。其核心思想为“抑制残差项”，尽可能抑制二次积分后的放大作用。另外，目前的程序仅支持二维平面运动的记录，是比较理想的。这一限制导致了实际地形起伏带来的 z 方向加速度对 x 、 y 方向计算的影响。这一点需要在硬件的更新中进行强化升级，如加入陀螺仪功能。

其次，我们深知搭建一个强引导性、易于操作、用户友好的交互界面是十分重要的，但是受制于 `microbit` 自带 LED 灯的数量，我们实现的人机交互界面是相对粗糙的。可以对界面进行一定量的改造使其更适用于日常生活的需要，同时，在外接屏幕和音响的情况下，交互界面会显得更加人性化。当然这些不是一蹴而就的，而是要在深度分析不同用户的使用体验后才能不断地进行优化。

最后，虽然此次作业使用 **microbit** 实现种种功能都是比较初级的，但是这样的设计能够带来许多启发，希望在未来拥有更强算力和更精确传感器的基础上，我们能够实现更加实用、更加酷炫的功能。

五、小组分工合作

陈开元：修改各个接口并编写总程序，完成部分实习报告，拍摄视频，拍摄活动照片

莫昊宇：编写各个模块接口的程序，完成部分实习报告，制作海报，剪辑视频