

利用micro:bit探究物体的失重状态

李琦 浙江省诸暨市牌头中学

李旻 上海蘑菇云创客空间

谢作如 浙江省温州中学

加速度，顾名思义，即速度变化的快慢。当用力踩下汽车油门时，汽车速度突然变快，有着较大的向前的加速度；当轻轻点下油门时，汽车逐渐变快，有着较小的向前加速度；踩下刹车时，速度变慢，此时汽车有着向后的加速度。

正如牛顿第二运动定律所诠释的，质量一定的物体所受外力越大，加速度越大。当汽车在加速时，人也能感受到椅背的推力，这便是加速度所产生的作用力。〔加速度关系满足方程：加速度 a （米/平方秒）=合力 F （牛）/质量 m （千克）〕

重力，是一种由万有引力产生的作用力，质量越大的物体受到的重力越大。重力的测量非常简单，将弹簧秤在垂直于水平地面放置就能测得到重力的大小。〔重力关系满足方程：重力 G （牛）=质量 m （千克） \times 重力加速度 g （9.8米/平方秒）〕

然而，当初学者接触到重力加速度时，总会有这样的疑难点：人在静止的状态下能轻易感受到重力，但是却没有任何加速度，重力为何会与“加速度”有联系？

比较直观的解释是：当物体失去支撑的时候，会以 g （9.8米/平方秒）的加速度向下坠落，此时重力

产生了作用物体的加速度。

加速度能产生力的作用，重力又能产生重力加速度，那重力是不是能被加速度所抵消呢？答案是肯定的。

经常做升降电梯的人可能会有这样的经验：当电梯上升的一刹那，脚会感觉一沉；当电梯即将停止的一刹那，脚底感觉一飘。这种感觉持续的时间并不会很长，也不会很强烈，但是这两种感觉，就是所谓的“超重”与“失重”了。正是电梯上升与下降瞬间的微弱加速度产生的力，与一部分重力叠加或抵消，使乘客有了脚底一沉（合力变大）和脚底一飘（合力变小）的感觉。

顺着这个思路，我们再做一个假设：如果让一个装着物体的盒子自由落体，其向下的重力加速度 g （9.8米/平方秒）产生的力是否能和重力完全抵消，使得盒中物体受到的合力为0，从而在盒子里自由漂浮？

我们很难用传统的弹簧秤来验证这个假设，因为：①弹簧秤体积太大，无法轻易抛出并安全落地；②力的方向可能来自四面八方，而弹簧秤只能测量单个方向的力，自由落体状态下很难固定朝向；③弹簧秤振子质量很大，受到力的作用

后需要等待振动结束之后才能读取稳定的数值，无法实时记录并反馈结果。

但是micro:bit（英国BBC公司联合微软、三星、ARM等公司联合开发帮助学生编程的微型电脑）的出现，使得我们能运用其轻松地代替弹簧秤测量加速度，因为①micro:bit非常轻巧，能轻松安全的抛出；②micro:bit板载三轴加速度传感器，能测量上下、前后、左右三个不同方向的加速度；③将这三个方向的加速度叠加就能算出物体所受的合力；④micro:bit能够将数据实时储存下来，方便在实验之后研究结果。

接下来，我们将通过micro:bit来探究重力加速度和加速度之前的关系，以及在“失重”状态下物体受到的作用力究竟有多大。

● 实验一：静止状态下“感受”到的重力加速度

重力加速度描述了当物体在仅受重力作用的情况下产生的加速度。当物体静置的时候，物体并没有任何的加速度，这是因为地面的支持力与重力加速度产生的重力相抵消。所以测量静置物体的重力加速度，就能得到其受到的重力的大小。

实验方法:

在micro:bit中写入以下python代码,并利用串口打印进行测试:

```
from microbit import *

while True:
    x=accelerometer.get_x()
    y=accelerometer.get_y()
    z=accelerometer.get_z()
    print("x=%f,y=%f,z=%f"
    %(x,y,z))
    sleep(10)
```

以上代码不停地获取加速度计x、y、z轴的值,并用格式化字符串的形式在串口上打印出来。可以看到,当micro:bit正面向上平放时,输出的三轴上加速度值如图1所示。

而把micro:bit竖放时,三轴上加速度值如图2所示。

不难发现,加速度传感器获取的值并非直接表示物体本身速度的变化方向,而是还包含了一个g的重力加速度值。而且x、y、z三轴值分别表示重力加速度值在图3三个轴向上的分量。

当micro:bit正面向上平放时,重力完全作用在z轴上,此时z轴的读数非常接近重力加速度g。但是当micro:bit的姿态非平放也非竖直放置时,三轴的读数实际上表示重力分别在三轴上的分量。而为了得到处于各种姿态的micro:bit所受的重力值,需要对重力在三轴向上的分量进行矢量合成并取模。因为三个轴向的加速度矢量两两正

交,因此,可以用公式 $g = \sqrt{x^2 + y^2 + z^2}$ 计算检测到的重力加速度g。

因此,可以将上述python代码改写如下:

```
from microbit import *
import math

while True:
    x=accelerometer.get_x()
    y=accelerometer.get_y()
    z=accelerometer.get_z()
    g=math.sqrt(x**2+y**2+z**2)
    print("%f" %(g))
```

程序更新后(如图4),可以看到无论micro:bit处于何种静止的姿态,串口输出的值都接近于地球表面的重力加速度g的值,即9.8。

```
x=-16.000000,y=-32.000000,z=-992.000007
x=-16.000000,y=-16.000000,z=-992.000007
x=-16.000000,y=-16.000000,z=-992.000007
x=-16.000000,y=-16.000000,z=-992.000007
x=-16.000000,y=-32.000000,z=-1008.000016
x=-16.000000,y=-32.000000,z=-1008.000016
x=-16.000000,y=-32.000000,z=-1008.000016
x=-32.000000,y=-32.000000,z=-992.000007
x=-32.000000,y=-32.000000,z=-992.000007
x=-16.000000,y=-32.000000,z=-992.000007
x=-16.000000,y=-32.000000,z=-992.000007
x=-16.000000,y=-32.000000,z=-992.000007
x=-16.000000,y=-16.000000,z=-992.000007
```

图1

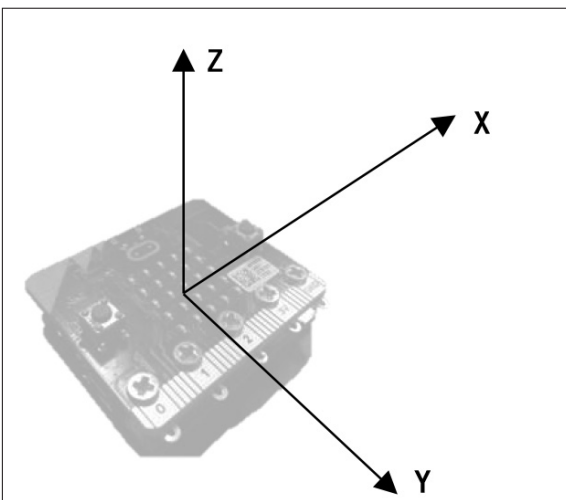


图3

● 实验二: 上下晃动时“感受”到的重力加速度

实验一验证了物体在静止状态下所受到恒定重力作用,以及重力加速度的存在。

那在垂直方向晃动micro:bit时,产生的这部分加速度是否又会和重力加速度抵消或者叠加呢?

通过串口绘图软件,可以得到如下页图5所示的曲线。

曲线记录了micro:bit不断上下晃动的状态。可以发现,曲线始终在1000数值上下晃动,正好接近实验一中测出的静止状态下的重力加速度数值。在micro:bit上下运动的过程中,因为加速度的方向不断改变,

```
x=16.000000,y=1024.000048,z=32.000000
x=16.000000,y=1024.000048,z=32.000000
x=16.000000,y=1024.000048,z=16.000000
x=0.000000,y=1024.000048,z=16.000000
x=0.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=16.000000
x=16.000000,y=1024.000048,z=32.000000
```

图2

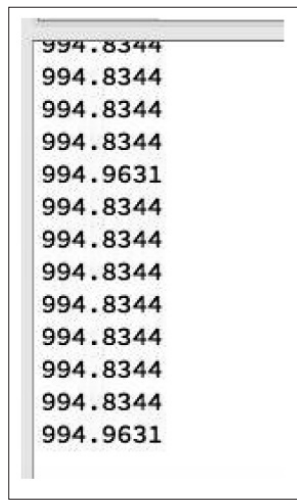


图4

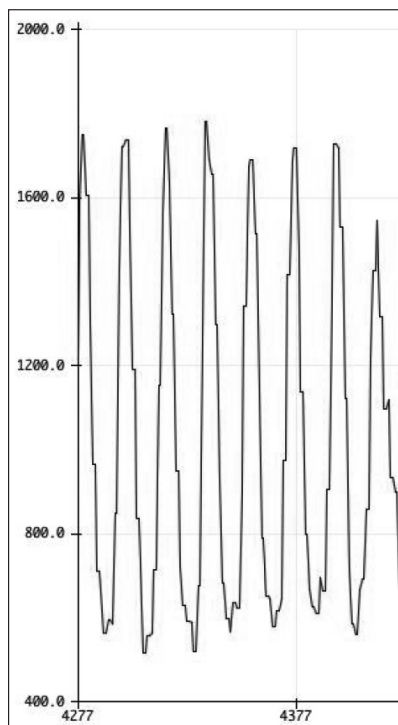


图5

板子受到的合力有时大于重力,有时小于重力,即前文所说的“超重”与“失重”状态。

● 实验三: 将micro:bit抛向空中, 观察其“感受”到的力

现在, 我们做一个大胆的尝试: 把micro:bit扔出去, 并记录这个过程中检测到的加速度值。micro:bit将要飞行的轨迹大约如图6所示。

为此, 需要如下这样一段完整的python代码让micro:bit去执行(如图7)。

micro:bit上电后, 会在LED矩阵中显示一次“ready”, 然后一直重复显示“go”, 这时候按住按钮A并准备将它丢出(当然你得为micro:bit准备一个较为柔软的着陆点, 如沙发或者床), 脱手之

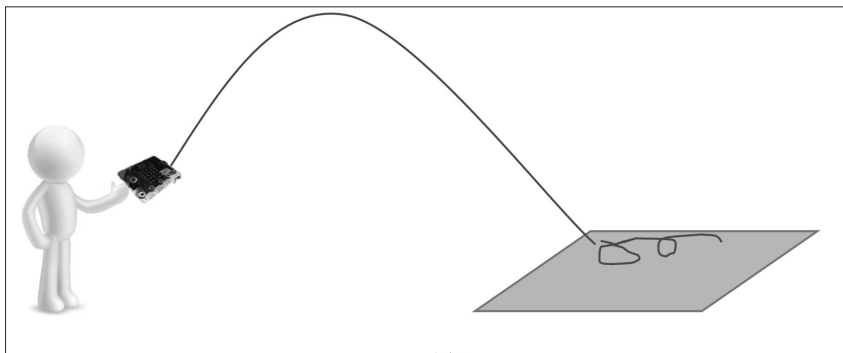


图6

```
from microbit import *
import math
import os
display.scroll("ready", delay=50)
while not button_a.is_pressed():
    display.scroll("go", delay=50)
while button_a.is_pressed():
    pass
ts=running_time()
t=ts
s='t\ta\r\n'
for i in range(180):
    t=running_time()
    x=accelerometer.get_x()
    y=accelerometer.get_y()
    z=accelerometer.get_z()
    g=math.sqrt(x**2+y**2+z**2)
    s=s+"d\t%0.2f\r\n" %((t-ts), g)
    print("%d\t%0.2f" %((t-ts), g))
    with open('data.txt', 'wt') as dt:
        dt.write(s)
display.scroll("done", loop=True)
```

#需要数学库以使用 sqrt 函数
#涉及文件操作的库
#显示“ready”
#等待按钮A被按下同时不停
#显示“go”
#等待按钮A被释放
#记录当前运行时间作为起始点
#生成标题行
#进行有限次的循环
#因为micro:bit资源有限
#因此需要控制循环次数
#180次的循环经过测试大约
#需要1.7秒左右
#从传感器得到加速度值
#把时间与加速度值保存到字符串
#串变量中
#把字符串写入data.txt文件
#循环显示“done”提示完成

图7

后按钮A自然就被松开了。一旦按钮A松开, micro:bit会记录当前的running time为起始时刻, 之后便是有限次的循环计算时间以及加速度值, 最后把这些值写进一个名为“data.txt”的文件中。

● 获取数据记录

micro:bit中的数据文件保存在它的闪存中, 我们可以将micro:bit用USB线连接至电脑, 并打开mu, 点击按钮, 此时在下方出现了两个

列表框(如下页图8)。

左边的列表框表示在micro:bit中的文件, 现在只需要把data.txt拖拽到右侧, 就可以在下页图9所示的文件夹中找到data.txt了。

在OS或者Linux系统中, 也可以通过microFS来实现, 这里就不具体介绍了。

● 数据分析

因为在写入数据时加入了制表符(\t)以及回车换行(\r\n), 我们可



图8

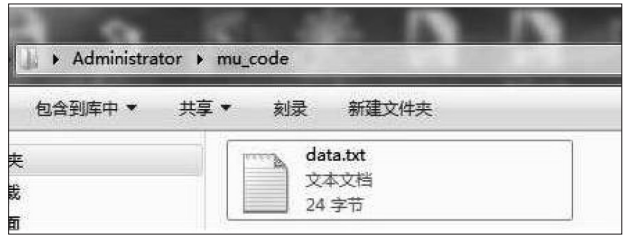


图9

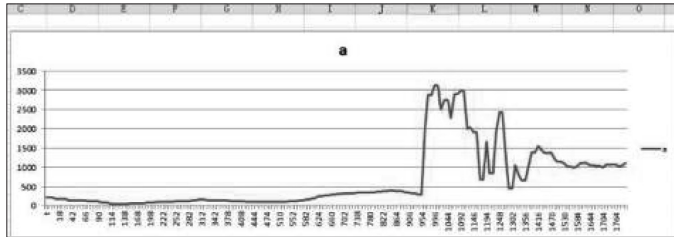


图10

以将data.txt文件中的数据全部复制到Excel中进行处理。一大堆数据用肉眼肯定没办法研究，还是先制作成图表以便于观察。

图10中横坐标表示扔出后经过的时间(ms)，纵坐标表示检测到的加速度值(mm/s²)。通过对图表的观察，我们将micro:bit的经历分为三个过程：过程一，从0时刻开始到大约950毫秒，此过程为空中飞行过程，无论是斜向上过程还是斜向下过程，micro:bit检测到的加速度值都很小；过程二，大约954毫秒时加速度值突然增大，直到1580毫秒之前都非常不稳定，此过程应该是micro:bit落地后翻滚造成的；过程三，大约1580毫秒后，加速度值稳定在1000左右，此时micro:bit已经停止了翻滚，得到的数值非常接近于地球表面重力加速度值。

● 问题与思考

物体的受力分析一直是高中物理教学中的重点与难点，因为这一知识点不仅包含了物理知识，还需

要结合数学中的空间几何知识。利用micro:bit的加速度传感器不仅可以直观验证一些力学问题，还可以开展很多有意思的探究活动。例如，把micro:bit放置到人偶中，通过在不同的高度丢下后落地时受到的最大加速度值探究高空坠落对人体的伤害；也可以在相同的高度用不同的方式落水，探究不同落水方式中人受到的加速度冲击有多大等。

micro:bit的内存有限，因此通过文件记录的方式无法满足记录大量数据的需求。当然，我们也有解决方案，如使用两个micro:bit，其中一个将采集到的数据实时通过蓝牙发送给另一个，而另一个就负责与计算机进行串口通信并实时将蓝牙收到的数据转发给计算机进行分析处理。

micro:bit不仅仅是一款帮助中小学生对编程的智能硬件，更是实施STEM教育的利器。利用micro:bit自带的传感器就能做一些科学实验，如果加上其他扩展模块，能做

的研究项目就更多了，值得在中小学中推广应用。*e*

如果对相关内容感兴趣，请关注主持人博客。

