



测一测你的睡眠质量

谢作如 金孜达 浙江省温州中学

我们如何得知自己的睡眠质量?比如夜里翻身几次,有没有梦游?或许大家第一个想到的就是智能手环。常见的智能手环的确都带有睡眠监测功能,但我们对其监测结果深表怀疑,因为手环毕竟是戴在手上的,未必准确。而最可靠的监测位置,应该在人体的中心,即肚子。于是,我们试着用加速度传感器做了一个能监测睡眠质量的小仪器。

● 材料选择

我们选择了Microduino。Microduino是Arduino的兼容开发板,体积很小,一块板子和1元硬币相近。因其具有电子积木的特点,堆叠方便,不需要焊接连线,做产品原型特别方便。制作这个作品大致需要如下材料,分别说明如表1:

表1

| 名称 | 说明 |
|--------------------------|-----------|
| Microduino Core | 主芯片模块 |
| Microduino USBTTF FT232R | USB数据交换模块 |
| Microduino 10DOF MPU6050 | 加速度传感器模块 |
| Microduino SD | SD卡读写模块 |

当然,SD卡和USB线是不可少的,这里不再赘述。这四个模块可以直接

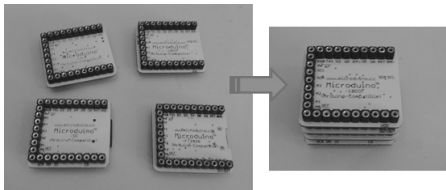


图1 组装好的Microduino模块

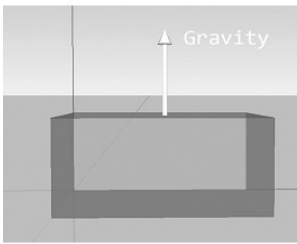


图2 仰卧

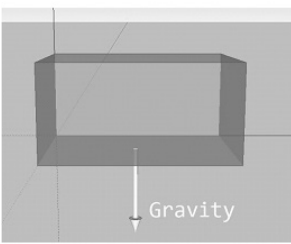


图3 俯卧

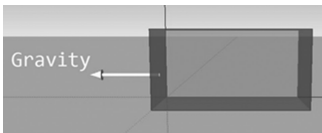


图4 左侧卧

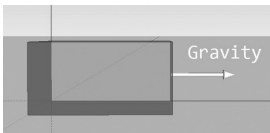


图5 右侧卧

堆叠,高度约2CM,如图1所示。

● 工作原理分析

睡眠监测仪的关键就是负责记录身体的朝向,并将其忠实地记录于SD卡上,以便于特制的数据分析器分析数据。我们先分析一下对身体朝向的定义。

图2至图5的视角是当你将其佩戴在腹部时,从头部往腹部看的视角。

为了更易观察,下一页图6、图7的视角发生了变动,请使用原先的相对视角看待它们。

然而事实上,几乎不可能得到如图2至下一页图7那样的监测值。实际上往往每次监测都会得到重力加速度的X、Y、Z方向的分量,且均不为零。对此,我们采用了一个非常简单的判断法:取模最

长的一个分量对应的方位为此次的方位,如下页图8所示。

此外,我们还可以顺便记录每相邻2次测得数量值的矢量差,这些都是有用的数据,可以用来判断是否入睡以及估算一段时间内的睡眠质量等。

● 算法优化

最简单的方法是周期性地监测重力加速度,不加任何处理地直接原始地记录入SD卡,将一切处理任务全部交付于数据分析器。然而,这种方法一个晚上会产生大量的数据,如果每100ms记录一次,记录8小时,则文件大小为5.5M,明显不是一种好方法。

第一种方法产生的文件之所以大,是因为存在大量的冗余数据。例如,一个人睡觉时一般会在10~15分钟保持同一朝向几乎不移动,而这段时间得到的数据十分接近,却都被记录了。所以应该要剔除相似数据,即设定一个阈值,阈值以内的数据不记录。

然而,我们本来就是为了记录身体的朝向,所以需要记录的其实仅仅是身体朝向改变的事件。因此,数据还可

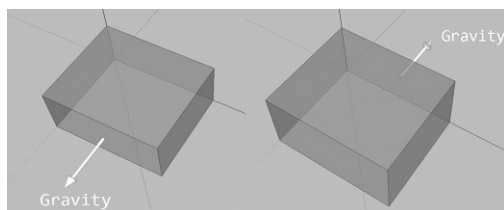


图6 站立

图7 倒立

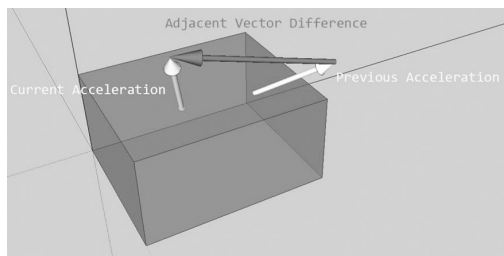


图8 方位取样

以进一步优化。经过测试,一个晚上记录的文件的大小一般小于3K。

需要注意的是,并不是任何原始数据都是可信的,除了无法预测的数据噪音,更值得关注的还有如下两种情况:①用户根本没有进入睡眠状态。我们无法期待用户在睡着前一瞬间启动产品,因此启动产品的时机都是睡着前的一段时间,而这段时间用户的行为被认为是相对活跃的。此时的数据根本不当被记入,否则会对数据产生一定的干扰。②用户已经进入睡眠状态,但是身体正在运动。虽然一般情况测得的加速度可直接视为重力加速度,然而当用户转身或者有大幅度的运动时,就不能如此轻率地将测得的加速度用于确定当前用户朝向的数据来源。

第一种情况的解决方法是并不急于记录数据,而是将监测仪分为“监视状态”和“记录状态”。一开始监测仪处于“监视状态”,该状态仅仅将数据写入内存而不写入文件。我们认为,若一段时间内数据变化不大且朝向主要不为

站立时,则用户已经进入睡眠状态,随后切入“记录状态”并新建数据文件。在“记录状态”,数据不仅被写入内存,还会经过初步处理写入文件。我们认为,若一段时间内用户几乎一直处于站立状态,则用户已经离开睡眠状态,随后返回“监视状态”并终止数据文件。

第二种情况的解决方法是综合考虑附近的数据。在这种处理方式下,我们可以较轻松地排除个别的突变数据,而不将之错误地作为有效数据进行处理。但如果用户确实发生了朝向改变等大动作,我们也能够正确地认知到这种变化并将其予以考虑。

● 睡眠指数的实现

睡眠好不好,总要弄个指标吧。睡眠质量指数是我们为了增加设备的功能而设计的一个参考指数。一般来说,在相等的一段时间内,身体活动越少,睡眠质量肯定越好。我们通过获取这段时间内任意相邻2次测得加速度的矢量差的模的平方并求和,来衡量身体如何活动。显然,在相等一段时间内,模的平方和越大,身体的活动就越剧烈。

我们每次对朝向相同的一段连续时间计算睡眠质量指数,考虑到这些时间不尽相同,还需要将其除以时间差。这是睡眠质量指数的计算公式:

$$Q = 2000 \left[\frac{1}{2} - \frac{1}{\pi} \arctan \left(\frac{\sum_{n=1}^{t_e-t_s} |\vec{a}_n - \vec{a}_{n-1}|^2}{100(t_e - t_s)} \right) \right]$$

其中 $\vec{a}_i (0 \leq i \leq \frac{t_e-t_s}{T})$ 表示这段时间内第 i 次测得的加速度, t_s 是这段时间相对于启

动仪器的开始时刻, t_e 是这段时间相对于启动仪器的结束时刻, T 是相邻2次测量的周期。

由于一共测量了 $\frac{t_e-t_s}{T}$ 次,故这段时间内的“平均相邻加速度差的模的平方”的值为 $\frac{\sum_{n=1}^{t_e-t_s} |\vec{a}_n - \vec{a}_{n-1}|^2}{t_e - t_s}$,除以100是数据上的需求(防止溢出)。

接着对计算得到的值进行映射。因为原先的值域为 $[0, +\infty)$,故对其进行一次反正切运算并除以圆周率,就可以将其映射到一个上下有界的区间 $[0, \frac{1}{2}]$ 。由于一般情况下我们觉得这个值越高睡眠质量越好,因此将其取负。为了方便观察,再加上 $\frac{1}{2}$ 。最后,乘以2000将其映射到 $(0, 1000]$,且此时睡眠质量指数与睡眠质量刚好成正相关,符合要求。

● 程序编写

代码编写并不是很难,但是只能用Arduino的代码写,图形化工具做不到。因为篇幅的缘故,下页表2仅提供“监视状态”模式的代码和简要解说,完整内容请访问文尾二维码链接的博客。

即使是经过初步处理的数据,其格式对一般用户来讲依然晦涩难懂,且格式不友好。因此,将数据转变为用户易于直观读取和理解的内容就成为了一项重要的任务。我们采用VB编写分析器的源代码和界面。虽然外表简陋,不过已经能将数据显示得足够直观。

首先,我们点击菜单中的文件选项打开文件选择框,选择一份文件(该文件是一份生成数据文件,格式为*.rd,仅供演示),然后,数据将被处理与显示,分写界面如下页图9所示。

表2

| 代码 | 注释 |
|---|--|
| <pre>void monitor() //Monitor Mode { #ifdef DEBUG_MODE Serial.println("Starting Monitoring..."); #endif int inital_count=0; for(; (double)oog_count/CACHE>OOG_PASS- RATE (double)direct_count[3]/CACHE>RAS- PASS_RATE inital_count<CACHE; current- index=(current_index+1)%CACHE) { oog_count-=distSqr(current- index, (current_index+1)%CACHE)>GATE; direct_count[acceleration[current- index].direct]--; getAcc(current_index); oog_count+=distSqr((current- index+CACHE-1)%CACHE, current_index)>GATE; direct_count[acceleration[current- index].direct]++; if(inital_count<CACHE) inital_count++; } previous_event_time=millis(); #ifdef DEBUG_MODE Serial.println("Ending Monitoring..."); #endif return; }</pre> | <p>[当进入“监视状态”时运行的函数]</p> <p>[调试模式下使用的代码]</p> <p>["监视模式"的终止条件为:在最近10分钟内连续的两次测量值的矢量差超过一个阈值(GATE=6000)的计数值比率小于80%(OOG_PASS_RATE=80%),且最近10分钟内测得为站立状况的计数值比率小于2.5%(RAS_PASS_RATE=2.5%),且至少进入该模式10分钟。如果有一个没有满足,会重新回到循环,每次循环的间隔为约1s.]</p> <p>[将RAM中最旧的一次数据抹除并清除其影响。]</p> <p>[记录新的测量值带来的影响。(如果这次的测量值与上次的测量值之间的矢量差是大于一个阈值(GATE=6000)则oog_count加1,此次测量值对应的方位的direct_count加1。)]</p> <p>[调试模式下使用的代码]</p> |

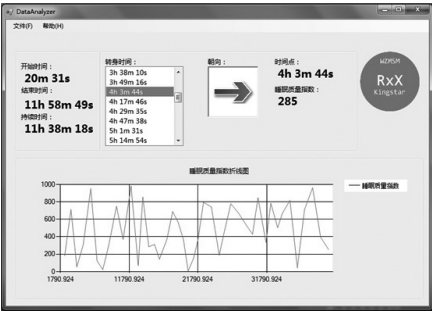


图9 分析界面

左上角表示睡眠的时间,一般情况只需了解最下面的“持续时间”即可。“起始时间”是自产品启动到开始记录文件的时间,“终止时间”是自产品启动到结束记录文件的时间。

右上角有一个按时间升序排列的

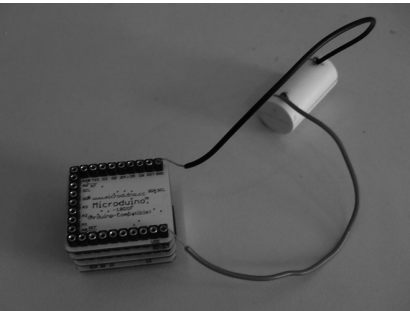


图10 连接电池的模块

列表,分别记录每一次身体转向的时间,和转向完毕后身体的朝向,以及这个朝向的这段时间内睡眠质量指数。睡眠质量指数是一个量化数值,可以反映你的睡眠质量,值在0~999内浮动,数值越高,睡眠质量越好。

下方是一个睡眠质量指数图表,直接完全地显示了每个转向的时刻及此段时间的睡眠质量指数,可清晰直观地了解一次睡眠的总体质量与变化。

● 作品原型

做一个作品原型吧,于是我们找了三个小电池并连接起来(如图10),以方便脱机运行。我们还采用3D打印外壳,将芯片与电池放入其中,最后封口。考虑到要以腰带的形式固定在肚子上测试,我们最终拆了一个毛绒绒的公仔,放在里面。

睡眠质量的监测仪器并不复杂,制作简易,探索历程却十分有趣。设计过程用到了单片机、编程和3D打印,体现了科技协作的力量。但这个作品还存在很多需要进一步考虑的问题,如加入一些简易有效的滤波算法,以便高效精准地处理原始数据;通过一段时间的数据更好地推测当前用户的睡眠状况;改进数据分析软件,使其更加易懂并具有更良好的交互功能等。

此外,睡眠数据的记录使用了SD卡。考虑到现今使用的数据格式并不会占用大块空间,因此还可以改良数据的传输方式,如使用蓝牙或者WiFi将数据直接实时传送到数据分析软件,这些都是下一步要研究的重点。e

如果对相关内容感兴趣,请关注主持人博客。

