廣東工業大學

《传感器技术及应用》
项目报告

2020——2021 学年第 2 学期

题　　目　基于有限元差分方法的久坐提醒坐垫

学　　院　信息工程学院

专　　业　信息工程

年级班别　18 级 1 班

小组成员　陈雅清 3218002266

范林静 3218002268

林碧丽 3218002270

陈丹萍 3218002315

邓莹　 3218002317

指导教师　何家峰

2021 年 5 月 8 日

摘要

科技的进步、互联网的普及，更让人们的生活方式更加多样化，网上娱乐方式也日益增多。社会的进步也带动了教育的普及、脑力劳动的增加，压力的增大。更多学生不得不花费更多时间坐在椅子上完成作业，上班族不得不长时间待在办公室完成上司安排的任务，导致久坐时间越来越长。人们也因工作、学习或者休息而逐渐喜欢坐着或躺着，不愿离开原来的位置，使得国内外久坐情况日益严峻。研究表明，久坐不动或少动位列十大致死、致病因素的第 8 位，它可以增加几乎所有疾病的死亡率，全球每年有近 200 万人的死亡与此有关。可见，解决久坐情况刻不容缓。针对久坐的严峻情况，我们决定制作一款久坐提醒坐垫，及时提醒人们远离座位，进行适当运动。

在软件方面，通过建立非稳态热传导模型，根据环境温度不同以及衣服材质不同计算温度传感器放在大腿内测稳定后的温度，得到裤子沿着厚度方向的温度分布。为减少误差，选取棉织、涤纶材质、皮椅、木制椅子进行温度稳定性验证，实验证明，不会对稳定后的温度产生太大影响。考虑到感应温度在裤子边界以及室内环境温度存在温差，引入牛顿冷却定律，通过前向差分的形式建立有限元模型，进而通过有古典隐式方法求解得到传感器温度阈值。硬件与软件的相结合，使久坐提醒坐垫可根据衣服材质和室内环境温度自动调节传感器温度阈值。在人体与坐垫接触时，成功感应温度并得知温度已超过温度阈值 Temp，执行 alarm_cnt++,alarm_cnt 为时间，如果 alarm_cnt 不等于 0，则模拟坐的情景；温度若小于设定值，alarm_cnt=0，模拟人离开坐垫的情景；当 alarm_cnt 大于 1 小时后，表明久坐，则蜂鸣器响应，为报警模式，而且必须起身手动关闭开关才能使报警声音消失。

在硬件方面，从价格、性能、准确性等方面选择合适的传感器，市场上久坐提醒坐垫采用的是压力传感器，由于压力传感器容易受力不均、承受力不足、价格昂贵，导致市场受众不多。所以我们最终决定采用 DS18B20 温度传感器，且通过一根不锈钢探头测量温度，可以防水，防压，实现精准温度测量。并且 DS18B20 温度传感器感应温度稳定，价格便宜，供电电压仅为 3-5.5VDC，所需的供电电压很小，一般的蓄电池就足以供电，方便使用。电路方面，结合单片机进行电路设计，电路简洁易操作，适用大多数人群。本产品，设置定时提醒功能，应用场景多变，可操作性高而且，为了使产品更加一体化同时为了保护硬件模块，对作品整体进行外观设计，使用 UG 软件对产品外包装进行 3D 建模与实物制作。

从而，让更多久坐人群离开原来的位置，适当活动，远离久坐带给自身的危害。平价的温度传感器与舒适的坐垫相结合，成为一款合理舒适，注重健康的坐垫，这是科技的进步。然而，促进科学技术的发展的同时，也别忘了身体素质的提升。及时起身，拒绝久坐，注重长久健康。

关键字： 久坐提醒 坐垫 有限元模型 牛顿冷却定律 非稳态热传导模型 温度传感器

目录

一、 选题背景

1.1 市场需求

随着社会和经济的发展，社会分工越来越精细，脑力劳动岗位也日益增多，使得长时间坐在电脑前办公越来越普遍。九年义务教育的普及，让越来越多的青少年有书可读，知识多样性的增加和人才选拔难度的提高，使得青少年长时间坐在椅子上完成作业和上课。随着年龄的增大，老年人的运动条件和娱乐方式的减少，使得久坐现象也很严峻。

久坐行为（sedentary behavior,SB）指个体在清醒状态下进行的坐或躺的活动，其能量消耗范围为 1.0 1.5METs[1]。大多数学生都表示久坐是一种无意识形成的行为，很少会想到要起来走走，只有实在累了才会伸伸懒腰或是起身走走。而上班族表示有时一工作起来就会不自觉连坐几个小时，直到感觉累了才意识到久坐了，或是下班之后再做一些运动 [2]。只为了一时的欢快或者工作，对久坐的危害存在侥幸心理，殊不知，久坐导致的危害日益增加。长时间坐着不活动，再加上喝水少，血液循环减缓，导致大脑供血不足，伤神损脑，产生精神压抑，日久还会使心脏机能衰退，引起心肌萎缩；长期坐着不运动易造成身体肥胖，促进体内产生“胰岛素抵抗”，使胰岛素的降糖功能大打折扣；坐着工作、学习或玩电脑久了，缺乏运动，会激发“久坐基因”活跃起来，最终导致肠癌的发生 [3]。2002 年世界卫生组织健康报告显示，久坐不动或少动位列十大致死、致病因素的第 8 位，它可以增加几乎所有疾病的死亡率，全球每年有近 200 万人的死亡与此有关 [4]。

在如此多的危害下，仍有研究显示，儿童青少年久坐时间普遍偏高，以上课、家教、做功课等为主的教育类久坐占据儿童青少年上学日/休息日久坐的大部分时间 [5]。也有国内一家网站的一项调查显示，目前国内有 43% 的人每天在办公室至少 8 个小时，而美国大多数人每天至少要坐八九个小时，午休时间只有 1 个小时 [2]。我国的老年人久坐情况也不容乐观。郭海军等研究结果显示在 2010 年至 2012 年间，我国老年居民静态活动时间平均为 2.66 小时/天，77.5% 的老年居民平均每天久坐时间在 2 小时以上 [6]。

在如此严峻的久坐情况和巨多的久坐危害下，人们还未能意识到自己久坐或者不愿起身运动，因此及时提醒人们久坐时间成为一项很重要的任务。无论哪个年龄段，都存在着大量的久坐人群，可见具有久坐提醒功能的产品的市场极大。

1.2 久坐提醒装置的技术现状

通过搜索各大电商平台，发现关于具有久坐提醒功能的产品还掺杂着其他附属功能，使产品的价格上涨。加之本身采用的技术和昂贵的元件，让消费者望而却步。目前的久坐提醒装置，主要有以下 3 种：

（1）基于穿戴设备。如智能手环通过加速度传感器及算法来实现久坐提醒功能，但

它经常不能准确判断人的坐下及离开行为状态，其提醒功能经常不动作或误动作。

（2）基于压力传感器装置。如健康座椅通过体压检测进行坐姿提醒，但使用柔性力敏传感器成本较高，不利于系统地应用推广。市场上关于压力传感器的售价，随着承重力和耐受力的提高上涨飞快，加上压力传感器存在受力不均匀，需要采用多个应变片，使成本更加高昂，对于各个年龄段的消费者来说，这是极其不友好的。

（3）基于电脑提醒软件。目前存在通过在电脑上安装防久坐的相关软件，电脑定时在显示屏上显示久坐提醒的防久坐方式。但是该软件无法准确获取电脑用户处于坐着的状态，也无法判断是否是同一个人使用电脑。

综上所述，目前各大具有久坐提醒功能的产品各有缺点，市场受众不多，准确性不太高，外界影响因素较多。因此，综合考虑，为准确感应到人体的久坐情况，我们采用DS18B20温度传感器，通过接触温度传感器，传感器接受到人体温度信号，产生电信号并传给单片机。通过建立有限元模型，提前输入的程序对传感器和单片机进行控制。由于采用非稳态热传导模型，根据环境温度和衣服材质智能算出温度阈值，而温度传感器一旦并持续感应到人体温度，并且超过温度阈值，则记下相应的时间，超过提前设定的久坐时间，蜂鸣器则启动，发出提醒式声音。利用DS18B20温度传感器及有限元模型，及时提醒人们离开座位，我们将设计一个久坐提醒坐垫，一旦坐在椅子上超过一定时间，该坐垫会提醒人们离开座位，从而避免久坐带给人们的危害。考虑到该坐垫的市场和新颖性，我们也查阅了大量关于久坐和温度传感器的资料，以及进行了中国知网和百度的检索，检索证明见附录。由1.1市场需求可知，无论哪个年龄层，国内外都存在大量的久坐人群，关于久坐危害的研究也日益增多，人们对久坐危害也有了进一步的了解，所以目前急需合适的久坐提醒坐垫让人们远离椅子。而市场上的久坐提醒产品都没有很好地让消费者离开座位，而久坐提醒坐垫一旦发生声音，消费者需要起身去关闭开关，才能让自己的耳朵“清静”。我们采用DS18B20温度传感器，也是技术上的一种创新与改进，改善目前产品的缺点，寻找合适的技术去制造更好的产品。并且，市场上还未有久坐提醒坐垫，具有久坐提醒功能的大都是具有其他功能的复杂产品，体验感大打折扣。因此，久坐提醒坐垫，不但能够准确感应到人体温度，还有计时提醒功能，同时满足人们对舒适坐垫的需求，价格也很亲民。可见，久坐提醒坐垫的市场将会很大，也会让更多人远离久坐带来的危害。

二、 设计思路

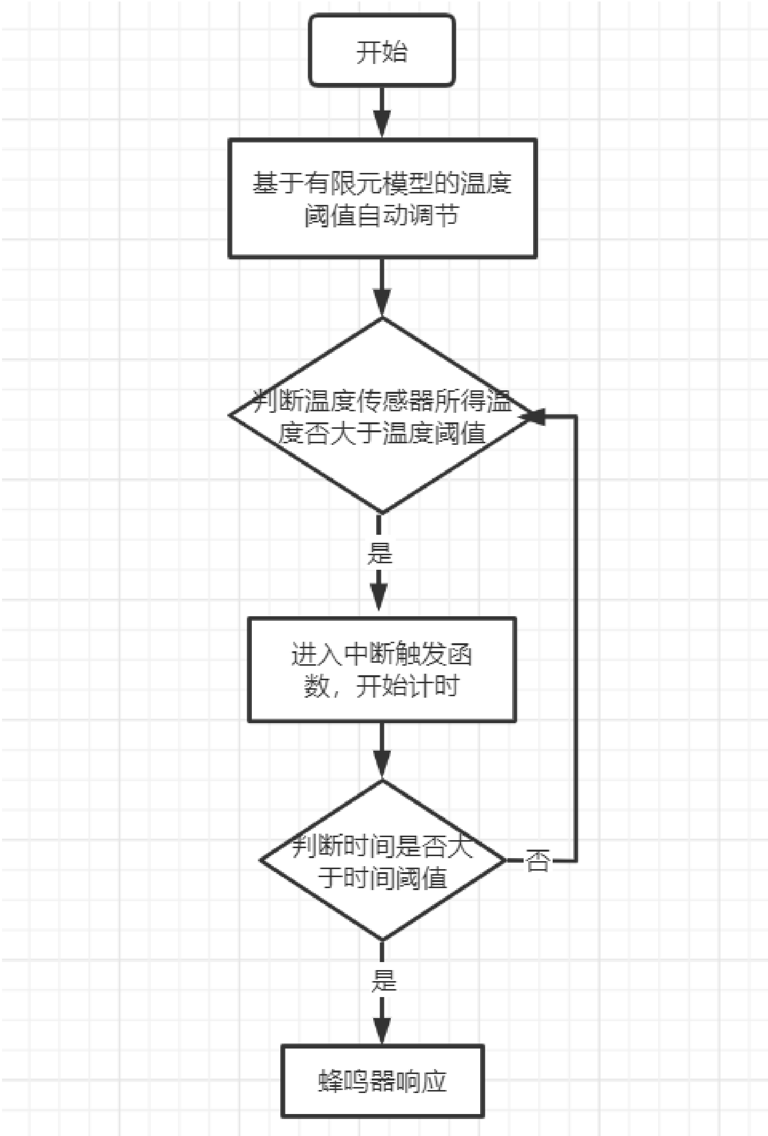

图 **1**　项目流程图

2.1 思路解析

基于温度传感器的久坐提醒坐垫，起关键判断作用的是不同环境中人体久坐后坐垫温度的设置以及久坐时间的设置。首先是温度阈值，我们基于非稳态热传导模型自动调节温度阈值得到不同环境温度以及不同衣服材质下温度稳定后的坐垫温度，将该温度设置为温度传感器的温度阈值，当传感器检测到坐垫的温度高于温度阈值时，模拟坐的情况，调用中断促发函数，计时器开始计时；温度低于阈值时，不调用调用中断促发函数，计时装置不启动，模拟离开的情况。

其次是设置时间阈值。根据现代研究，建议端正坐姿的基础上，每 1h 即进行站坐转换、全身伸展活动或其他轻体力活动 2 3 min，因此，我们设定计时时间阈值为 1h。

在同时满足温度阈值和时间阈值两个条件后，计时超过一个小时，也就是模拟久坐一个小时，蜂鸣器响应，提醒使用者离开座椅休息。

三、 条件假设与合理性分析

1. **假设人体温度维持在 37 摄氏度**

– 一般情况下，人体的温度恒定，且不会出现很大的偏差，波动范围在 ±0.3℃ 之间。波动范围小于 DS18B20 数字温度传感器的测量精度（±0.5℃），对实验结果不会造成很大影响，所以可以忽略人体温度的波动。

2. **不考虑极端环境**

– 结合人们的实际生活生产环境，我们基于非稳态热传导模型对温度进行自动调节，理论上，DS18B20 的测温范围（-55℃ +125℃）可以满足在大部分极端环境下的测温，但是在这些极端条件下，考虑到其他元器件（如一些非线性元件电阻等）的工作性能会改变，因此不予考虑极端环境如：极寒或者极热的条件下，产品是否能正常工作。

3. **不考虑温度剧烈突变**

– 针对普通人正常的生活环境的温度在不同的时间或者季节有差异，我们对此进行了室内室外不同温度下的温度阈值的计算。选取的温度分别为空调房间室内 16.0℃、24.0℃，室外 29.0℃，在不同的环境温度下，人体的温度呈现 ±0.1℃ 的差异，因此可以忽略人体温度的变化。在不同的环境温度下分别计算出的温度阈值分别为：30.5℃、31.3℃、31.4℃（计算结果精确到一位小数）；由计算结果可知，环境温度变化在小范围（±10.0°C）波动时，温度阈值的变化范围在 ±1.0℃；而这种温度变化是可以基本上满足我们的日常四季变换的温度变化的。除此之外，热传导是一个随时间变化的量，本次实验采用的 DS18B20 数字温度传感器温度探测器接口采用的是不锈钢，在 600.0℃ 以下，各种不锈钢的导热系数基本在 10 30W/（m•℃）范围内，导热性一般，因此导热部分不会温度的突变。因此综上所述，基于温度传感器的久坐提醒坐垫的实现是合理的。

另外，本次项目不考虑温度的大幅度的突变，因为与正常的生活环境温度变化相差太大，所以不作考虑。

四、 软件部分

4.1 基于非稳态热传导模型的温度阈值的自动调节

4.1.1 一维热传导方程的确定

对于热传导：热传导描述了当物体两端存在温度差时，热量自动的从高温部分传到低温部分的现象 [7]。热传导根据介质温度是否变化分为以下两种传热情景：非稳态热传导描述了热量传递时，存在热端温度逐渐下降，冷端温度逐渐上升，最终达到温度一

致的平衡。稳态热传导描述了热量传递时，介质在某一维空间上各处温度恒定。显然 1 层介质的温度变化属于非稳态热传导。

一般情况下的热传导为三维非定态热传导，温度随时间和三个空间坐标变化，可用热扩散方程描述如下：

$$\frac{\partial(k\frac{\partial u}{\partial x})}{\partial x}+\frac{\partial(k\frac{\partial u}{\partial y})}{\partial y}+\frac{\partial(\frac{\partial u}{\partial z})}{\partial z}+q=\rho C\frac{\partial u}{\partial t} \tag{1}$$

式中 u 为温度，x、y、z 为三个空间坐标方向，ρ 为衣服密度，C 为恒压热容，k 表示衣服热导率，q 为单位体积内热量产生的速率；其中，$\frac{\partial(k\frac{\partial u}{\partial x})}{\partial x}$、$\frac{\partial(k\frac{\partial u}{\partial y})}{\partial y}$、$\frac{\partial(\frac{\partial u}{\partial z})}{\partial z}$ 分别描述了热量沿 xyz 方向的传导速率，整个方程描述了单位体积内能量变化速率等于热量进入单位体积的净热传导速率与单位体积的热能产生速率之和。

本题的实验条件下，仅研究与厚度有关的一维空间，且各分层内不含热源，由此上式可化简为：

$$\alpha\frac{\partial^2 u}{\partial x^2}=\frac{\partial u}{\partial t} \tag{2}$$

式中，α 称为热扩散系数，α=$\frac{k}{\rho C}$，ρ 为密度，C 为恒压热容，k 表示热导率。

延裤子厚度方向的温度随时间而变化，最终达到动态平衡，为非稳态传热，即单位体积内能量变化速率等于热量进入单位体积的净热传导速率，方程如下：

$$\begin{cases}\alpha\frac{\partial^2 u}{\partial x^2}=\frac{\partial u}{\partial t}, x\in(0,d)\\ u\big|_{t=0}=37\\ u\big|_{x=0}=24\end{cases} \tag{3}$$

式中，α 为裤子的热扩散率，α=$\frac{k}{\rho C}$，ρ 为密度，C 为恒压热容，k 表示热导率，d 为裤子的厚度;$u\big|_{t=0}=37$ 表示 t=0s 时刻衣服初始温度为人体温 37ºC，$u\big|_{x=0}=24$ 表示环境温度为 24 摄氏度，模拟空调房温度。

4.1.2 有限元模型的建立

由于热传导微分方程组的解不易计算，故本文采用有限差分法计算其数值解，其基本思想是将定义域进行网格划分，在网格点上以差分表示微分，将原微分方程转化为差分格式求解。对时间-热传导方向所在的一维空间平面进行网格划分，分别取 Δt、Δx 为 t 方向与 x 方向的步长，则 $u_{i,j}$ 表示 $t=i\Delta t$、$x=j\Delta x$ 处的介质温度。

4.1.3 热传导方程的差分近似

对于一维热传导方程：$\alpha\frac{\partial^2 u}{\partial x^2}=\frac{\partial u}{\partial t}$，在网格内点 (i,j) 处，对 $\frac{\partial u}{\partial t}$ 采用向后差商公式，对 $\frac{\partial^2 u}{\partial x^2}$ 采用二阶中心差商公式，则一维热传导方程可表示为：

$$\frac{u_{i+1,j}-u_{i,j}}{\Delta t}-\alpha\times\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{\Delta x^2}=0 \tag{4}$$

式中，α 称为热扩散系数，$\alpha=\frac{k}{\rho C}$，ρ 为密度，C 为恒压热容，k 表示热导率。则衣服介质内热传导方程可差分近似表示为：

$$\begin{cases}\frac{u_{i,j}-u_{i-1,j}}{\Delta t}-\alpha_1\times\frac{u_{i-1,j+1}-2u_{i-1,j}+u_{i-1,j-1}}{\Delta x^2}=0, i\in(0,\frac{t}{\Delta t}], j\in(0,\frac{d}{\Delta x}) \\ u_{0,j}=37, j\in[0,\frac{d}{\Delta x}] \\ u_{i,0}=24, i\in[0,\frac{t}{\Delta t}]\end{cases} \tag{5}$$

式中，$u_{0,j}=37$ 表示 t=0s 时刻衣服初始温度为人体温 37ºC，$u_{i,0}=24$ 表示环境温度为 24ºC。

热传导方程有多种差分格式，其中古典隐式格式取不同步长时稳定性更高，故项目采用古典隐式格式描述和求解差分方程。

4.1.4 牛顿冷却定律的差分处理

在求得裤子厚度方向的温度分布后，由于传感器与裤子直接接触的交界面处需要单独考虑，还需要引入牛顿冷却定律，牛顿冷却定律描述了当周围环境与物体表面存在温度差时，单位面积变换的热量速率与该温度差成正比，比例系数名为热传递系数，公式如下：

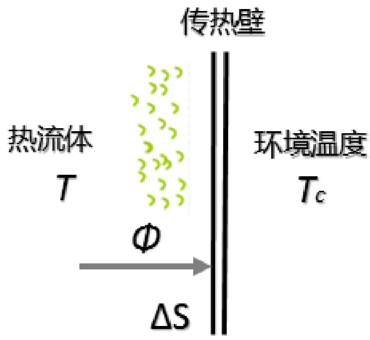

图 **2**　牛顿冷却定律原理图

$$\frac{u_{i,j}-u_{i,j-1}}{\Delta x}=\eta(u_{i,j-1}-u_{ren}), i\in(0,\frac{t}{\Delta t}], j=\frac{d}{\Delta x} \tag{6}$$

式中，u_{ren} 表示人皮肤温度,d 为裤子厚度。

4.1.5 古典隐式格式求解方程组

将差分处理后的耦合条件加入热传导方程组，解方程组，即可逐层解得随时间推移各位置的温度分布。由于不能直接计算各层介质不同时间各个位置上的温度，必须通过

解下列线性方程组才能计算 $u_{i,j}$，此方程组是三对角方程组：

$$
\begin{bmatrix}
1+2\alpha & -\alpha & & & & & \cdots\cdots & \\
-r & 1+2\alpha & -\alpha & & & & \cdots\cdots & \\
 & \cdots\cdots & & & & . & \cdots\cdots & \\
 & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \\
 & \cdots\cdots & & & & -\alpha & 1+2\alpha & -\alpha \\
 & \cdots\cdots & & & & & \eta+\frac{1}{\Delta x} & \frac{-1}{\Delta x}
\end{bmatrix}
\begin{bmatrix}
u_{i,1} \\ u_{i,2} \\ \cdots \\ \cdots \\ \cdots \\ u_{i,n-1}
\end{bmatrix}
=
\begin{bmatrix}
u_{i-1,2}+r_1 u_{i,1} \\ u_{i-1,3} \\ \cdots \\ \cdots \\ \cdots \\ 37\eta
\end{bmatrix}
$$

该方程组求解了裤子介质整体温度在时间、位置上的分布，其中，x 分别为裤子在空间维度上的网格坐标。

考虑到人体温维持在 37ºC，在整个传热过程中会降低人皮肤外侧的温度，即所求的空气层最外层温度略高于实际试验结果空气层最外层温度，因此需通过所求的对流传热系数 η 转换得空气层最外层的实际温度分布。具体求解步骤如下：

Step1：代入边界条件和耦合条件，通过非稳态热传导方程的离散形式逐层求解，得到裤子厚度方向温度随时间在各位置上的计算值；

Step2：结合传热系数 η，带入牛顿冷却定律方程，解的裤子外侧边界温度值，即为温度传感器感应值；

4.1.6 温度分布拟合结果

根据所得的对流传热系数 η，计算温度分布，绘制裤子厚度方与时间-空间的三维温度分布图如下如所示：

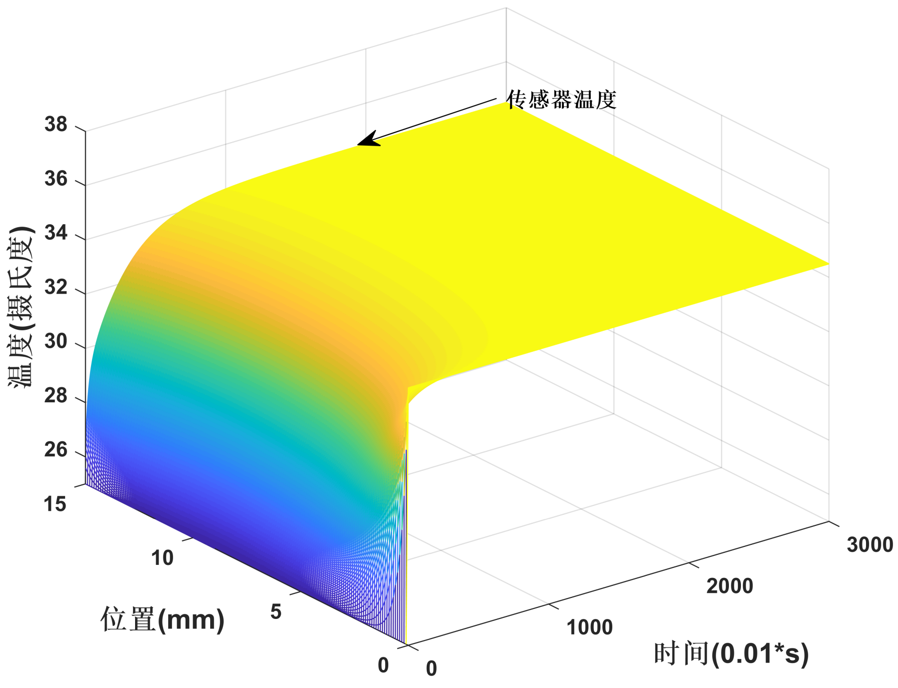

图 **3**　裤子厚度方与时间-空间的三维温度分布

4.2 久坐提醒代码设计

基于上述温度自动计算阈值的方法，可以根据不同环境温度、不同衣服材质的性质，自动得到的所该设计的温度阈值。所以我们定义将腿部感受到的温度超过温度阈值一定时间（可设为 1h）后即为久坐。具体设计思路如下：

Step1：通过隐式格式求解热传导模型得到不同环境温度以及不同衣服材质下温度稳定后传感器的温度 Temp;

Step2：通过 $Timer0\_Init()$ 函数配置 1ms 触发一次定时器中断，配置完之后，每 1ms 就会进入 $voidTimer0(void)interrupt1$；

Step3：每次进入 $voidTimer0(void)interrupt1$ 执行 $count++$, 当 count 大于 1000 的时候，也就是 1 秒钟

Step4：当传感器感应到的温度超过温度阈值 Temp 的时候，执行 $alarm\_cnt++$，$alarm\_cnt$ 为时间，如果 $alarm\_cnt$ 不等于 0，则模拟坐的情景；温度若小于设定值，$alarm\_cnt = 0$，模拟人离开凳子的情景;

Step5：当 $alarm\_cnt$ 大于 1 小时（60*60=3600）后，表明久坐，则蜂鸣器响应即报警;

4.3 误差分配及参数分配

本文最主要的误差是环境温度不同所导致的，所以我们建立了非稳态热传导模型，根据环境温度不同以及衣服材质不同计算温度传感器放在大腿内测稳定后的温度，首先我们模拟不同材质裤子对稳定后的温度的影响，选取了棉织、涤纶材质裤子进行模拟计算稳定后的温度，以及稳定所需要的时间，如下图所示：

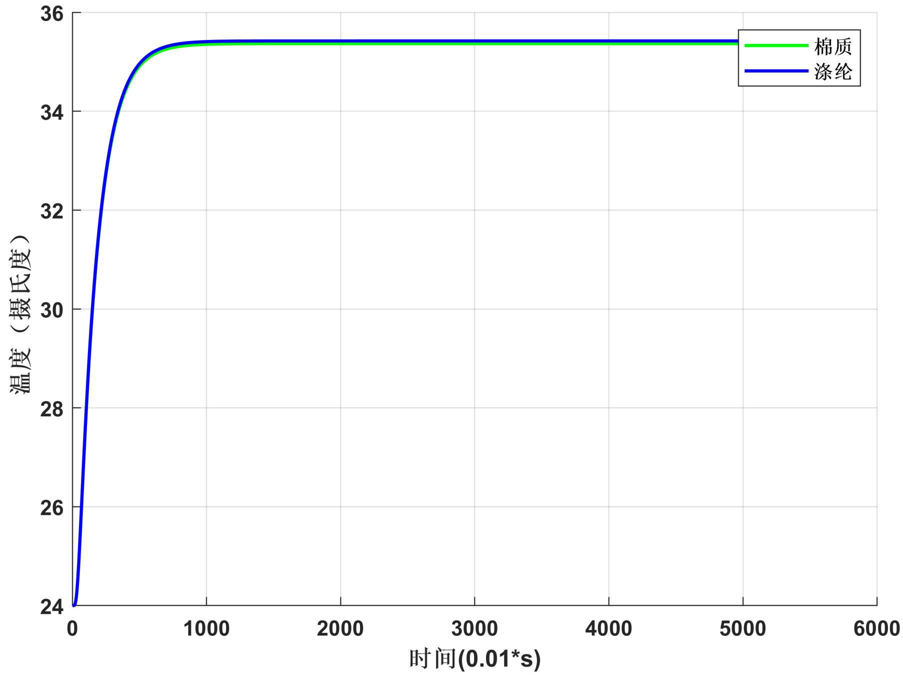

图 **4**　棉织、涤纶材质裤子进行模拟计算稳定后的温度对比

由此可见，由于衣服材质较薄，且热传导系数相差较小所以稳定的时间以及稳定后的温度相差均比较小。同时我们还考虑了不同材质凳子对稳定后温度的影响，如皮椅和木制凳子对稳定后温度以及稳定时间的影响。

表 **1**　温度分布

凳子材质	温度稳定所需时间（s）	稳定后的温度（摄氏度）
皮椅	5.92	34.4
木制椅子	6.12	34.2

由此可见，凳子材质并不会对稳定后的温度产生太大影响，对稳定时间影响也甚微。也进一步说明了我们系统的具有一定的稳定性以及鲁棒性。

4.4 抗干扰设计

为了使得温度传感器感应温度更加稳定，我们使用 DS18B20 防水温度传感器数字温度计探头 + 端子适配器模块 +3pin 电缆如下图所示，可将该模块放置于大腿下侧进行温度感应，经过我们对比发现相比普通 DS18B20 温度传感器稳定度具有了较大提升.

图 **5**　温度传感器模块

五、 硬件部分

5.1 电路结构形式

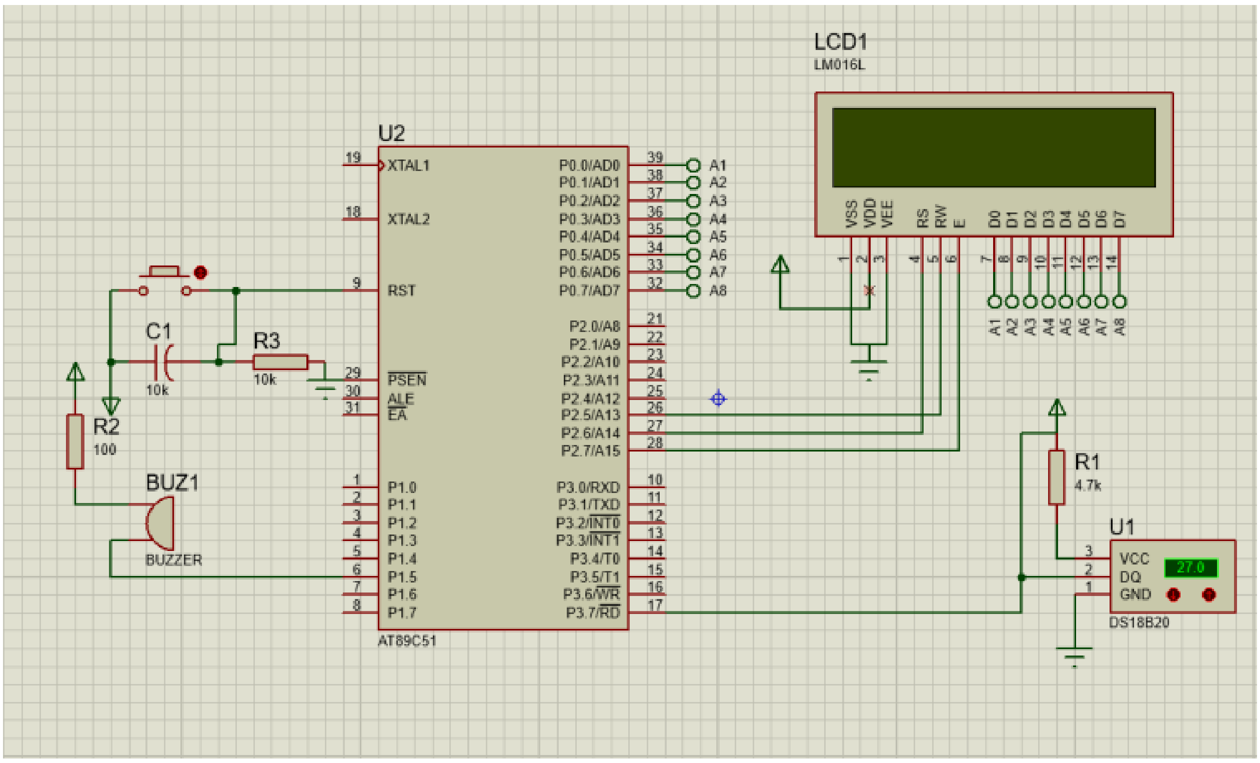

图 **6**　电路原理图

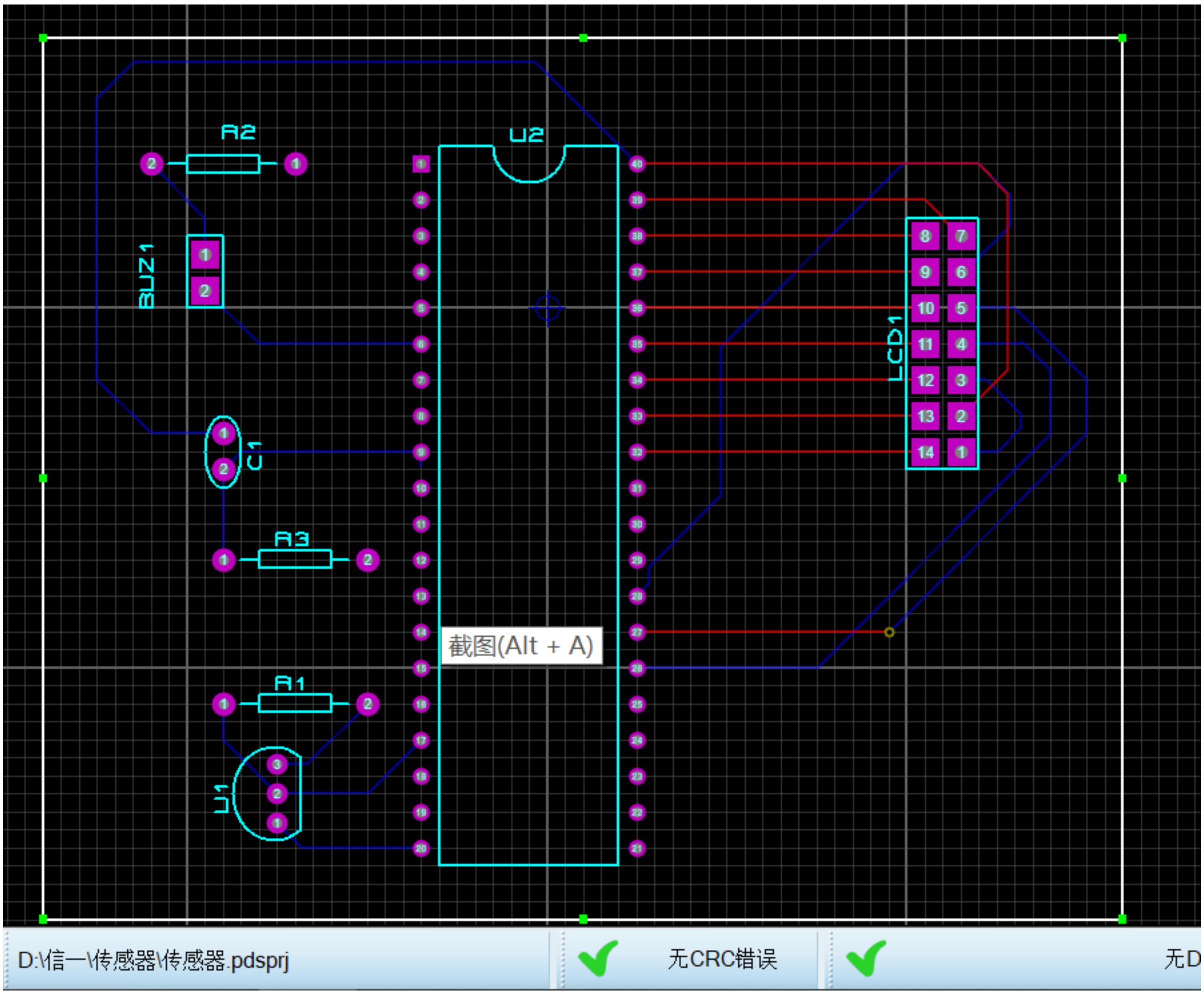

图 7　**PCB** 封装图

5.2 元件选择

关于温度传感器的选择，我们小组进行了对现有技术的分析（具体可参考背景部分），最终经过小组讨论我们决定了使用温度传感器来实现我们小组的项目想法。

测温在目前有数字温度传感器和热敏电阻两种选择，然而热敏电阻的适用范围在40℃ 90℃，而人体温度一般在 30℃ 40℃，所有热敏电阻并不符合我们所要测试的人体温度，并且热敏电阻可靠性差，测量温度准确率低，对于 1 摄氏度的信号是不适用的，还得经过专门的接口电路转换成数字信号才能由微处理器进行处理。显然，数字温度传感器，会更加适合我们小组的项目实验。

当然温度传感器也是有很多种选择，现在市面上常见的有 DS18B20，PT100, 为了方便对比，我们制作了如下表格（表一），来说明两者优劣性。通过表格我们可以对比得出，DS18B20 虽然在测量范围和精度上不如 PT100，但是对于我们小组的项目来说也足够了，并且 DS18B20 相较于 PT100，直接输出数字信号，也更加便于我们的项目，也更加便宜，实用。

表 **2** 元件参数

	测量范围	精度	数字信号	价格
DS18B20	-55℃-+125℃ 直	0.5℃	直接输出	较低
PT100	-200℃-+150℃	0.1℃	需要 AD 转换	较高

所以，根据以上的综合分析，我们小组最终选择用于测温度元件是 DS18 B20 温度传感器。并且为了更方便我们小组进行座垫对人体温度的测量，我们选择了探头性的 DS18B20。

5.3 组装调试结果

因为我们选择了温度探头,所以需要在连接时增加一个4.7KΩ的电阻,并且,DS18B20独特的单线接口仅需一个端口引脚进行通讯，且因其直接输出数字信号，不需要使用 AD 转换器，所以我们仅需将其接入单片机的温度传感器组件即可，连接的图解为下图3，可供参考。但仍需注意的，就是 DS18B20 连接有正反之分，一旦接反，就会发热，然后烧坏。当然，为了更好的固定 DS18B20 传感器，我们借助了杜邦线连接，并且在连接之前，通过万用表检测杜邦线是否能正常使用。但是，我们忽视了单片机的检测，连接电脑时没有检测到单片机，一度以为是使用的数据线，仅能充电，并不能传输数据，当然最终还是有惊无险的完成的这个项目。

5.4 性能测试与分析

基于非稳态热传导模型对温度阈值进行计算，输入环境、衣服布料的参数，得到我们所设定的阈值为 32.1℃。一开始我们用手心握着传感器，测出手心的温度是 34.9°C 左右，又考虑到座垫的因素限制我们将传感器用大腿压住，监测大腿的温度，我们发现，通过布料和人体部位温度不同，发现大腿的稳定温度事实上比手心低了，大概在 32.3℃左右，考虑到如果下天，空调房，室温的因素，我们又调整了之前代码编写的参数，得到了最终的设定阈值为 31.1℃。然后我们经过多次测量，不同人，不同环境，最终都可以得到我们想要的结果。

我们采用的温度传感器是 DS18B20 数字温度传感器，具有耐腐蚀性、防水性好等特点，能够在潮湿、水浸、腐蚀、碰撞、摩擦等等的严苛的工作场合中运用。参考它的技术参数进行分析：

（1）测温范围为-55°C-+125°C。一般情况下，久坐后温度稳定下来时的温度大约为 31°C 左右，即使在炎热或者寒冷天气，这个温度范围也适用。

（2）测量精度为：±0.5°C。久坐坐垫所需要的温度的精确度不需要很高。

（3）供电电压：3-5.5VDC。所需的供电电压很小，一般的蓄电池就足以供电，不需要再进行电压的调制。作用于坐垫，安全可靠。

5.5 电路改进

5.5.1 利用电机散热

我们使用的是温度传感器，对温度会比较敏感，为了防止单片机过热导致测量温度产生误差，我们可以增加一个散热系统进行电路改进。原理图如下：

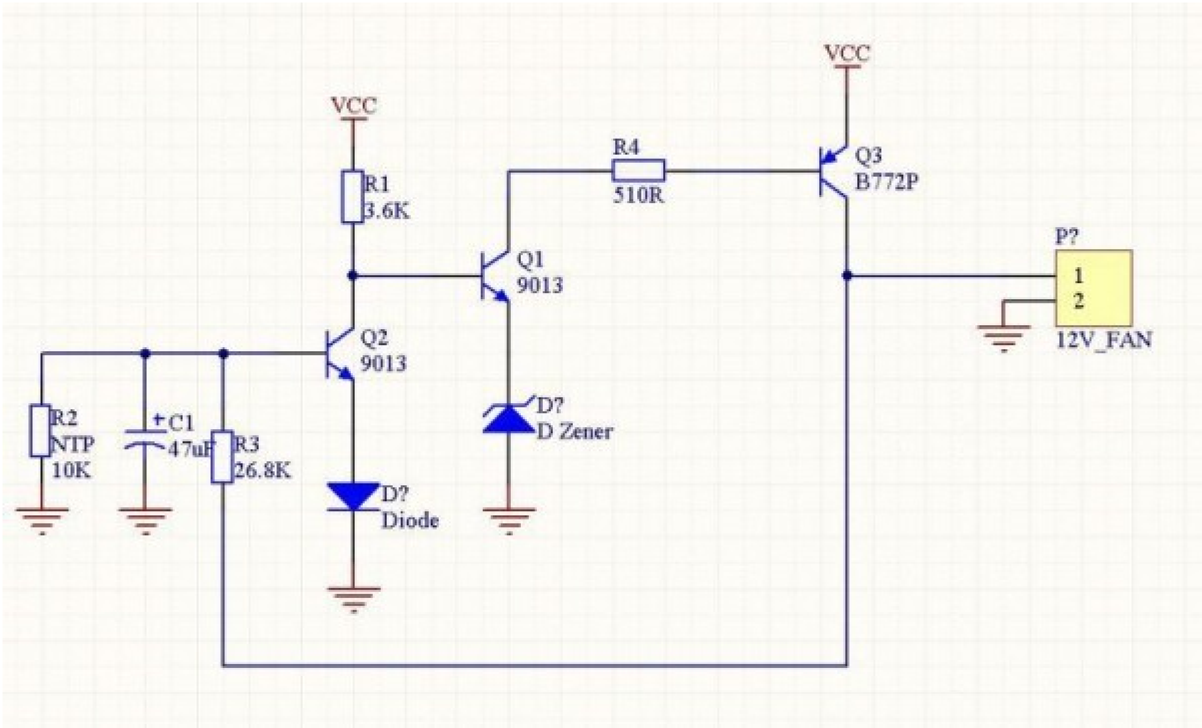

图 **8**　散热系统电路原理图

5.5.2 应用场景改进

我们这款座垫有定时提醒功能，所以也可以在其基础上添加时钟的电路结构，将其当成计时装置来使用。

5.6 外观设计

为了使产品更加一体化同时也为了保护硬件模块，我们对作品整体进行外观设计，使用 UG 软件对产品外包装进行 3D 建模与实物制作，设计图如下所示：

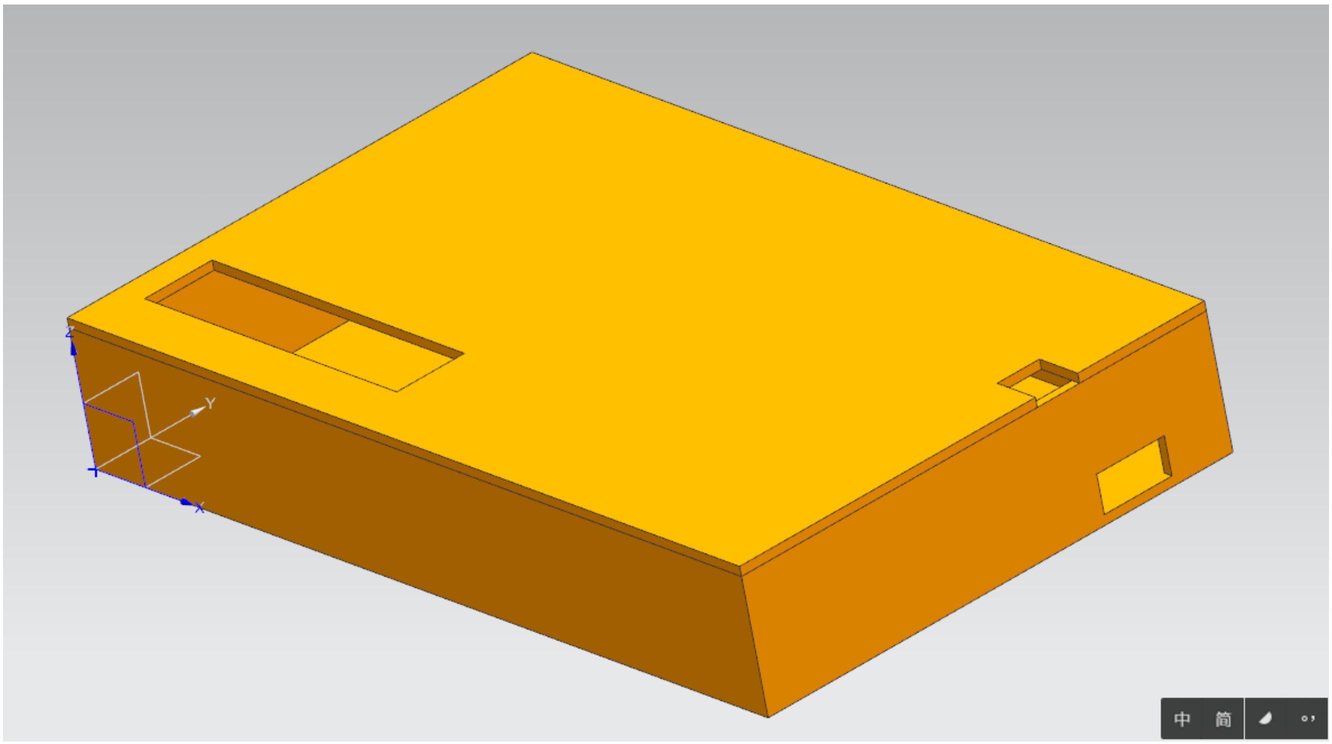

图 **9**　外观设计

六、 小组分工

林碧丽 (组长)	负责市场调研、对产品进行查重，文献检索，提供资料，参与各部分的协助，跟进项目的进度
陈雅清	编写软件程序，选择、购买元件，进行拼装连接成成品
范林静	绘制硬件方面的电路原理图，封装调试成品。
邓莹	PPT 制作与汇报演讲
陈丹萍	文献检索、报告编写

参考文献

[1] wen N, Healy G N, Matthews C E, et al. Too much sitting:the population health science of sedentary behavior[J]. Exerc Sport Sci Rev, 2010, 38(3):105-113.

[2] 赵斌, 何思萌.“久坐族”坐姿监测提醒设备研究 [J]. 设计,2016(21):108-109.

[3] 张欢欢. 久坐危害知多少 [J]. 家庭医学,2021(01):37.

[4] World Health Oraganization.World Health Report[R].Geneva:World Health Organization,2002.

[5] 杨剑, 吴铭, 邱服冰, 李安巧, 姜静远, 朱婷.12～17 岁儿童青少年健康相关久坐行为研究 [J]. 中国康复理论与实践,2020,26(12):1365-1372.

[6] Chomistek A K,Manson J E, Stefanick M L, et al. Relationship of sedentary behavior and physical activity to incident cardiovascular disease[J]. J Am Coll Cardiol, 2013, 61(23):2346-2354.

[7] 黄玉燕：《论“久坐伤肉”之“坐”》

知网和浏览器的网络检索

图 **10**　关键词“久坐提醒坐垫”知网检索

图 **11**　关键词“久坐提醒”知网检索

图 **12** 关键词“久坐提醒坐垫”百度检索

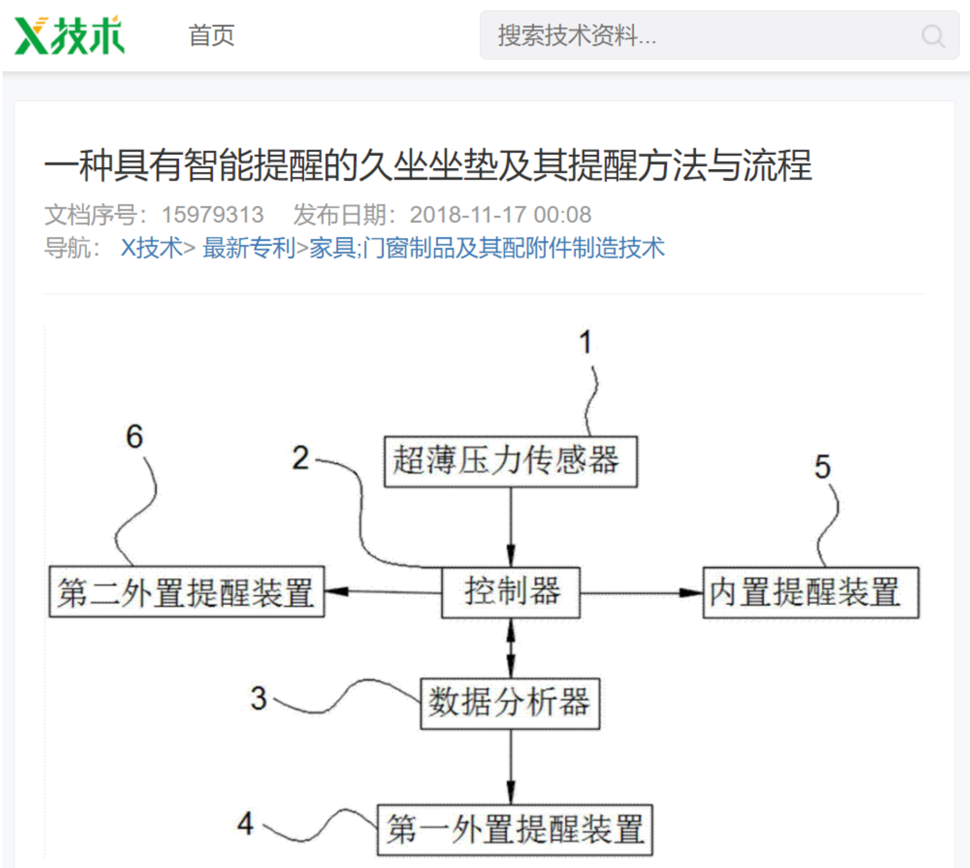

图 **13** 与本项目相类似的浏览器检索结果

附录 A　代码展示

main 文件

```
//main.c
/*
@2021/5/4
@funtion: 久坐提醒

实验说明:
将宏#define LCD1602_4PINS打开，我们这里使用的LCD1602是8位，所以默认将该宏注释。
实验接线:
1，LCD1602液晶模块-->单片机管脚
参考LCD1602液晶显示实验接线
2，独立按键模块-->单片机管脚
K1-->P34
K2-->P35
K3-->P32
3，蜂鸣器模块-->单片机管脚
BEEP-->P15
4，LED模块-->单片机管脚
D1-->P24
5，LED模块-->单片机管脚
D1-->P24
6，DS18B20模块-->单片机管脚
参考DS18B20温度传感器实验接线，这里我们使用的是可以坐防水的DS18B20模块

实验现象及解释:

Step1: 通过隐式格式求解热传导模型得到不同环境温度以及不同衣服材质下
温度稳定后传感器的温度Temp
Step2: Timer0_Init();是配置1ms触发一次定时器中断，配置完之后，每1ms
```

```
就会进入void Timer0(void) interrupt 1
每次进入就count++;当count大于1000的时候，也就是1秒钟
Step3: 当传感器温度超过31摄氏度的时候，判断alarm_flag == 1,
开始判断一下温度是不是大于设定值,
温度若大于设定值，alarm_cnt++，alarm_cnt为时间,
如果alarm_cnt 不等于0，则模拟坐的情景;
温度若小于设定值，alarm_cnt=0，模拟人离开凳子的情景
step4: 当alarm_cnt大于1小时（60*60=3600）后，表明久坐，则报警
*/

#include "public.h"
#include "lcd.h"
#include "temp.h"
#include "i2c.h"


sbit k3=P3^2; //设置温度上下限
sbit k1=P3^4; //加
sbit k2=P3^5; //减

sbit led=P2^4; //报警指示灯

sbit beep=P1^5; //蜂鸣器报警
sbit relay=P1^4; //加热设备

u16 temp_val; //检测的实际温度
u8 mode; //温度模式
u16 count = 0;
char alarm_falg = 0;
int alarm_cnt = 0;

void Temp_DataPros()
{
short temp;
u8 temp_buf[5];
```

```c
temp=Ds18b20ReadTemp();
temp_val=temp;
if(temp_val>310)
alarm_falg = 1;
else
alarm_falg = 0;
if(temp<0)
{
temp=-temp;
LCD_Dispstring(2+5,0,"-");
}
else
{
LCD_Dispstring(2+5,0," ");
}

temp_buf[0]=temp/100+0x30;
temp_buf[1]=temp%100/10+0x30;
temp_buf[2]='.';
temp_buf[3]=temp%100%10+0x30;
temp_buf[4]='\0';
LCD_Dispstring(2+6,0,temp_buf);  //显示检测的温度xx.x

temp_buf[0]=alarm_cnt/1000+0x30;
temp_buf[1]=alarm_cnt%1000/100+0x30;
temp_buf[2]=alarm_cnt%100/10+0x30;
temp_buf[3]=alarm_cnt%100%10+0x30;
temp_buf[4]='\0';
LCD_Dispstring(11,1,temp_buf);
}

void Timer0_Init(void)
{
TMOD |= 0x01;//设置定时器为工作方式1
TL0 = (65536 - 1000)%256; //装初值，低8位
```

```
TH0 = (65536 - 1000)/256; //高8位
ET0 = 1; //开定时器的中断
TR0 = 1; //开定时器
EA = 1; //开总中断
}

void Timer0(void) interrupt 1
{
TL0 = (65536 - 1000)%256; //装初值, 低8位//初值是1000即1ms
TH0 = (65536 - 1000)/256; //高8位
count++;
if(count == 1000)
{
count = 0;
if(alarm_falg == 1)
{
alarm_cnt++;
if(alarm_cnt>=30) //为了方便测试, 所以设置30s
{
alarm_cnt=30;
}
}
else
{
alarm_cnt = 0;
}
}
}

void sound()
{
u8 i=50;
while(i--)
{
beep=!beep;
```

```
delay(100);
}
}

void kai_display()
{
LCD_Dispstring(2,0,"Temp: C");
LCD_Dispstring(0,1,"Sit time:");
}

void main()
{
Timer0_Init();
LCD_Init();
kai_display();
while(1)
{
Temp_DataPros();
if(alarm_cnt>=30)
sound();
}
}
```

C 源程序

```
#include "i2c.h"

/******************************
* 函数名 : Delay10us()
* 函数功能  : 延时10us
* 输入 : 无
* 输出  : 无
******************************/
```

```

void Delay10us()
{
    unsigned char a,b;
    for(b=1;b>0;b--)
        for(a=2;a>0;a--);

}
/*****************************
* 函数名 : I2cStart()
* 函数功能  :
    起始信号：在SCL时钟信号在高电平期间SDA信号产生一个下降沿
* 输入 : 无
* 输出  : 无
* 备注 : 起始之后SDA和SCL都为0
*****************************/

void I2cStart()
{
    SDA=1;
    Delay10us();
    SCL=1;
    Delay10us();//建立时间是SDA保持时间>4.7us
    SDA=0;
    Delay10us();//保持时间是>4us
    SCL=0;
    Delay10us();
}
/*****************************
* 函数名 : I2cStop()
* 函数功能  :
    终止信号：在SCL时钟信号高电平期间SDA信号产生一个上升沿
* 输入 : 无
* 输出  : 无
* 备注 : 结束之后保持SDA和SCL都为1；表示总线空闲
```

```
*****************************/

void I2cStop()
{
    SDA=0;
    Delay10us();
    SCL=1;
    Delay10us();//建立时间大于4.7us
    SDA=1;
    Delay10us();
}
/*****************************
* 函数名 : I2cSendByte(unsigned char dat)
* 函数功能  :
    通过I2C发送一个字节。在SCL时钟信号高电平期间，保持发送信号SDA保持稳定
* 输入 : num
* 输出  : 0或1。发送成功返回1，发送失败返回0
* 备注 : 发送完一个字节SCL=0,SDA=1
*****************************/

unsigned char I2cSendByte(unsigned char dat)
{
    unsigned char
        a=0,b=0;//最大255，一个机器周期为1us，最大延时255us。
    for(a=0;a<8;a++)//要发送8位，从最高位开始
    {
        SDA=dat>>7; //起始信号之后SCL=0，所以可以直接改变SDA信号
        dat=dat<<1;
        Delay10us();
        SCL=1;
        Delay10us();//建立时间>4.7us
        SCL=0;
        Delay10us();//时间大于4us
    }
    SDA=1;
```

```
    Delay10us();
    SCL=1;
    while(SDA)//等待应答，也就是等待从设备把SDA拉低
    {
        b++;
        if(b>200)
            //如果超过2000us没有应答发送失败，或者为非应答，表示接收结束
        {
            SCL=0;
            Delay10us();
            return 0;
        }
    }
    SCL=0;
    Delay10us();
    return 1;
}
/****************************
* 函数名 : I2cReadByte()
* 函数功能  : 使用I2c读取一个字节
* 输入 : 无
* 输出  : dat
* 备注 : 接收完一个字节SCL=0,SDA=1.
****************************/

unsigned char I2cReadByte()
{
    unsigned char a=0,dat=0;
    SDA=1;   //起始和发送一个字节之后SCL都是0
    Delay10us();
    for(a=0;a<8;a++)//接收8个字节
    {
        SCL=1;
        Delay10us();
        dat<<=1;
```

```
        dat|=SDA;
        Delay10us();
        SCL=0;
        Delay10us();
    }
    return dat;
}


/****************************
* 函数名 : void At24c02Write(unsigned char addr,unsigned char
   dat)
* 函数功能  : 往24c02的一个地址写入一个数据
* 输入 : 无
* 输出  : 无
****************************/

void At24c02Write(unsigned char addr,unsigned char dat)
{
    I2cStart();
    I2cSendByte(0xa0);//发送写器件地址
    I2cSendByte(addr);//发送要写入内存地址
    I2cSendByte(dat); //发送数据
    I2cStop();
}
/****************************
* 函数名 : unsigned char At24c02Read(unsigned char addr)
* 函数功能  : 读取24c02的一个地址的一个数据
* 输入 : 无
* 输出  : 无
****************************/

unsigned char At24c02Read(unsigned char addr)
{
    unsigned char num;
```

```
    I2cStart();
    I2cSendByte(0xa0); //发送写器件地址
    I2cSendByte(addr); //发送要读取的地址
    I2cStart();
    I2cSendByte(0xa1); //发送读器件地址
    num=I2cReadByte(); //读取数据
    I2cStop();
    return num;
}
```

```
#include "public.h"

void delay(u16 i)
{
    while(i--);
}
```

```
#include "temp.h"
/*****************************
* 函数名 : Delay1ms
* 函数功能  : 延时函数
* 输入 : 无
* 输出  : 无
*****************************/

void Delay1ms(unsigned int y)
{
    unsigned int x;
    for(y;y>0;y--)
        for(x=110;x>0;x--);
}
```

```
/*******************************
* 函数名 : Ds18b20Init
* 函数功能  : 初始化
* 输入 : 无
* 输出  : 初始化成功返回1，失败返回0
*******************************/

unsigned char Ds18b20Init()
{
    unsigned int i;
    DSPORT=0;   //将总线拉低480us~960us
    i=70;
    while(i--);//延时642us
    DSPORT=1;
       //然后拉高总线，如果DS18B20做出反应会将在15us~60us后总线拉低
    i=0;
    while(DSPORT) //等待DS18B20拉低总线
    {
        i++;
        if(i>5000)//等待>5MS
            return 0;//初始化失败
    }
    return 1;//初始化成功
}

/*******************************
* 函数名 : Ds18b20WriteByte
* 函数功能  : 向18B20写入一个字节
* 输入 : com
* 输出  : 无
*******************************/

void Ds18b20WriteByte(unsigned char dat)
{
    unsigned int i,j;
```

```c
    for(j=0;j<8;j++)
    {
        DSPORT=0;   //每写入一位数据之前先把总线拉低1us
        i++;
        DSPORT=dat&0x01; //然后写入一个数据，从最低位开始
        i=6;
        while(i--); //延时68us，持续时间最少60us
        DSPORT=1;
            //然后释放总线，至少1us给总线恢复时间才能接着写入第二个数值
        dat>>=1;
    }
}
/*******************************************************
* 函数名 : Ds18b20ReadByte
* 函数功能  : 读取一个字节
* 输入 : com
* 输出  : 无
*******************************************************/


unsigned char Ds18b20ReadByte()
{
    unsigned char byte,bi;
    unsigned int i,j;
    for(j=8;j>0;j--)
    {
        DSPORT=0;//先将总线拉低1us
        i++;
        DSPORT=1;//然后释放总线
        i++;
        i++;//延时6us等待数据稳定
        bi=DSPORT; //读取数据，从最低位开始读取
        /*将byte左移一位，然后与上右移7位后的bi，注意移动之后移掉那位补0。*/
        byte=(byte>>1)|(bi<<7);
        i=4;  //读取完之后等待48us再接着读取下一个数
```

```
        while(i--);
    }
    return byte;
}
/*******************************************************
* 函数名 : Ds18b20ChangTemp
* 函数功能  : 让18b20开始转换温度
* 输入 : com
* 输出  : 无
*******************************************************/

void Ds18b20ChangTemp()
{
    Ds18b20Init();
    Delay1ms(1);
    Ds18b20WriteByte(0xcc);  //跳过ROM操作命令
    Ds18b20WriteByte(0x44); //温度转换命令
// Delay1ms(100);  //等待转换成功，而如果你是一直刷着的话，就不用这个延时了

}
/*******************************************************
* 函数名 : Ds18b20ReadTempCom
* 函数功能  : 发送读取温度命令
* 输入 : com
* 输出  : 无
*******************************************************/

void Ds18b20ReadTempCom()
{

    Ds18b20Init();
    Delay1ms(1);
    Ds18b20WriteByte(0xcc); //跳过ROM操作命令
    Ds18b20WriteByte(0xbe); //发送读取温度命令
```

```c
}
/*****************************************************
* 函数名 : Ds18b20ReadTemp
* 函数功能  : 读取温度
* 输入 : com
* 输出  : 无
*****************************************************/

short Ds18b20ReadTemp()
{
    unsigned char temp=0;
    unsigned char tmh,tml;
    short tem;

    Ds18b20ChangTemp();    //先写入转换命令
    Ds18b20ReadTempCom();  //然后等待转换完后发送读取温度命令
    tml=Ds18b20ReadByte(); //读取温度值共16位，先读低字节
    tmh=Ds18b20ReadByte(); //再读高字节

    if(tmh>7)
    {
        tmh=~tmh;
        tml=~tml;
        temp=0;//温度为负
    }
    else
    {
        temp=1;//温度为正
    }

    tem=tmh; //获得高八位
    tem<<=8;
    tem|=tml;//获得底八位
    tem=(double)tem*0.625;//转换 放大10倍 精度0.1
    if(temp)
```

```
        return tem; //返回温度值
    else
        return -tem;
}
```