

教育部高等学校大学计算机课程教学指导委员会

中国大学生计算机设计大赛



软件开发类作品文档简要要求

作品编号：_____

作品名称：_____智能枕芯_____

版本编号：_____V 1.1_____

填写日期：_____2020.5.22_____

填写说明：

- 1、本文档适用于**所有**涉及软件开发的作品，包括：软件应用与开发、大数据、人工智能、物联网应用；
- 2、正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；
- 3、本文档为简要文档，不宜长篇大论简明扼要为上；
- 4、提交文档时，以 PDF 格式提交本文档；
- 5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目 录

第一章 需求分析	5
1.1 产品开发背景	5
1.2 产品的特色分析	6
1.3 产品竞品分析	7
第二章 概要设计	8
2.1 系统功能设计	8
2.2 系统架构设计	8
2.3 物理架构设计	10
第三章 详细设计	11
3.1 前端设计	11
3.1.1 前端架构与逻辑设计	11
3.1.2 登录注册页面	12
3.1.3 主界面	15
3.1.4 设备列表页面	16
3.1.5 闹钟列表页面	17
3.2 后端设计	21
3.2.1 总体业务流程	21
3.2.2 基于深度学习的睡眠状态预测算法	22
3.3 硬件设计	24
3.3.1 框架	24
3.3.2 电路示意图	24
3.3 嵌入式软件设计	26
3.3.1 框架	26
3.3.2 主要运行流程	27
3.3.3 本地睡眠算法以及滤波设计	30
3.3.4 其它细节设计	31
3.3.5 成本估计	33
第四章 测试报告	34
4.1 本地测试	34
4.1.1 传感器读数以及串口测试	34

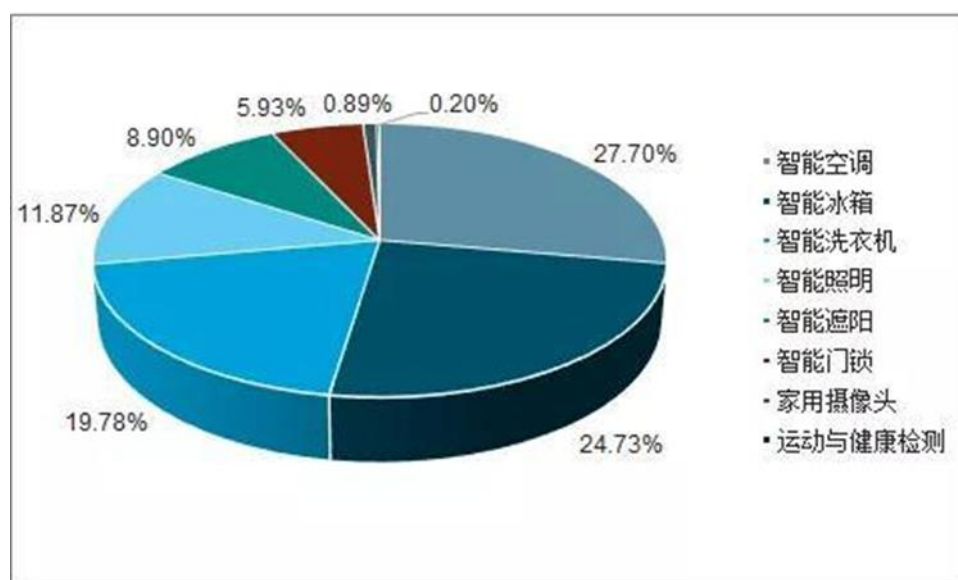
4.1.2 闹钟测试	35
4.1.3 多闹钟测试	36
4.1.4 与其它物联网设备联动	38
4.2 前后端测试	39
4.2.1 账户相关测试	39
4.2.2 全测试	43
第五章 安装及使用	45
第六章 项目总结	46
第七章 附录	47
7.1 项目任务书	47

第一章 需求分析

1.1 产品开发背景

智能家居主要指的是在智能家居控制系统当中对网络安全通信技术、自动控制技术与计算机网络技术进行充分的使用，让家居设备具有一定的智能特点，使用人员可以通过便携式设备，比如说手机对家居设备进行直接控制，比如说，通过无线终端设备、互联网等对家庭设备进行直接控制或者根据一定的场景需要来进行智能启动，另外家庭内部的设备可以进行相互通讯，并且配合操作，这样能够让家居设备更好的为人民服务，让人们获得一个更安全、更舒适、更便利的生活环境。

下图为智能家居主要产品市场占比



伴随当前物质水平进一步提高，智能家居会逐步广泛应用于人们日常生活中，为人们的生活提供一定的便利，让人们的生活质量得到进一步提高。而智能枕芯作为一个通过物联网技术，与其它智能家居进行互联互通的设备。能够家居设备更好的为用户服务。所以，其它各种智能家居，例如市场中占据大份额的智能空调，智能照明等产品，的发展也会为智能枕芯带来巨大的机遇。

1.2 产品的特色分析

智能枕芯是一个通过物联网技术，与其它智能家居互联互通的嵌入式设备。旨在利用最小的成本，最大化地提升用户生活水平与生活便利程度。我们作品的特色有如下几点：

一 高性价比

相比市场上价格上百的各种产品，我们的成本更加低廉，却可以有多的功能。

由于所有部件都是零售购买，成本会比厂家大量生产高很多。我们的成本如下：stm32c8t6 最小系统板 8.6 元，如果是芯片的话只需要 5.15 元（淘宝价），esp8266-01S 淘宝价 5.8 元，薄膜压敏传感器 49 元，传感器配套模块 14.8 元，震动以及蜂鸣器合计 5 元，锂电池 6.4 元，其余杂项硬件材料费用 3 元，合计 92.6 元。如果能够大量生产的话，通过更换传感器以及定制 PCB，预计成本可以到 40 元。

二 作为附件

虽然如今市场上已经有了很多智能枕头产品，但是它们的功能大多都是辅助分析睡眠状态，而没有帮助将人唤醒的功能。此外市面上的智能枕头大多都为整个枕头形式的成品，价格较贵，且不能制定。而我们打算做的智能枕芯只是作为枕头的一个附件，垫在枕头下面就可以让普通枕头智能化。而且用户可以自己选择自己喜欢的枕头，附加上我们的产品，就能达到市面上智能枕头的效果，相当于每个用户都能自定义属于自己的智能枕头，对用户有更大的吸引力。

三 互联互通

它通过单片机将各种传感器收集到的数据，通过 WIFI 模块与云端服务器交互，并结合用户在网站上设置的各种信息，做出提升用户生活便利程度的决策，例如在用户睡觉的时候，自动关闭房间中的灯。此外，我们还在做了硬件上做了离线处理机制，确保了在一段时间内断网的情况下，任然可以正常工作。

四 智能闹钟

市面上现有的智能枕头大多都将目标放在改善用户睡眠情况上，而忽视了一些别的适合集成在智能枕头上的功能，比如智能闹钟。而我们做的智能枕头则添加了这一功能。它能获取用户在网页上设置的闹钟信息，并且在用户该起床的时候智能做出反应。比如，如果用户已经起床了，闹钟就会自动关闭，如果用户一直不起床，闹钟就不会停止，防止睡过。

1.3 产品竞品分析

相比市场上已有的各种智能枕头，我们已经添加了智能闹钟和控制其它智能家电的功能。相比同样有着智能闹钟和控制其它智能家电功能语音助手(比如天猫精灵)，我们的设备更适合被放在卧室，因为它能监测睡眠状态，让闹钟更加智能。此外，我们的设备不会采集语音信息，对保护用户的隐私更友好。并且，我们的硬件成本是所有产品里面最低的。

	睡眠状态 监测	按摩功能	智能闹钟	控制其它 智能家居	价格
语音助手	X	X	✓	✓	100~500
智能枕头	✓	✓	X	X	300~1800
智能枕芯	✓	X	✓	✓	40~100

第二章 概要设计

2.1 系统功能设计

我们的智能枕芯及其在云服务器上的相关程序提供了如下功能：

- 1) 用户可以注册登录账户。由于智能枕芯十分适合与其他的智能家居协同工作，比如小米，华为等提供的智能设备，登录的最佳方案是与第三方的账号绑定。但由于我们并没有相关的授权，所以登录与注册仅仅作作为代替方案使用。
- 2) 管理设备，包括购买硬件之后根据指导添加设备到自己的账户，修改设备的名字方便记忆，与删除不用的设备。
- 3) 设置闹钟，通过闹钟设置的睡眠时间段，对同一用户的所有智能设备进行控制。比如在用户应该起床的时候，自动响铃，而在确保用户已经起床，或者是已经提前起床的情况下，自动关闭闹钟。这一点是目前其它智能家居很少做到的。此外，在用户晚上起床的时候，自动打开走廊上的灯，或者是在用户睡觉的时候，帮助用户进行自动关灯，解决了关灯后看不清的麻烦。

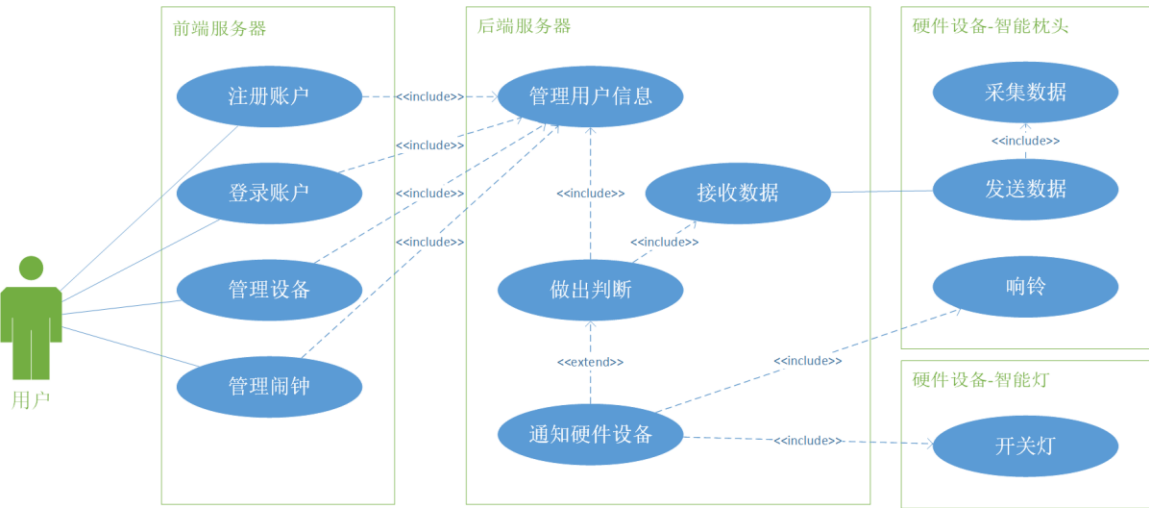


图 2-1 系统功能设计图

2.2 系统架构设计

我们系统架构一共分为 7 个层次，分别是 前端 UI，展示层，业务层，数据层，数据库，运行环境与设备模块。

前端 UI，展示层由前端发服务器来提供服务，主要是提供能够被用户访问的网页，并且将用户的操作传递给后端服务器。

业务层，数据层，数据库由后端服务器负责提供服务。业务层包括和前端服务器协作通讯完成用户管理，设备管理与闹钟管理，并且对数据库进行相应的读写操作。业务层还包括完成于硬件设备的通讯，接收硬件发送的数据，并且在做出预测之后将预测结果和用户在网站上的设置推送到硬件设备，以此完成智能的互联互通。同时后端服务器也包括对用户权限的控制与对必要过程的日志记录。

运行环境与设备模块则代表了系统功能与硬件的对应关系。其中前端服务器与后端服务器上的功能都是云服务器提供服务，确保能够 24 小时在线工作。而嵌入式设备则是我们的智能枕芯，设备模块着对应着智能枕芯在硬件上的组成。

整个系统的架构设计如如 2-2 所示。

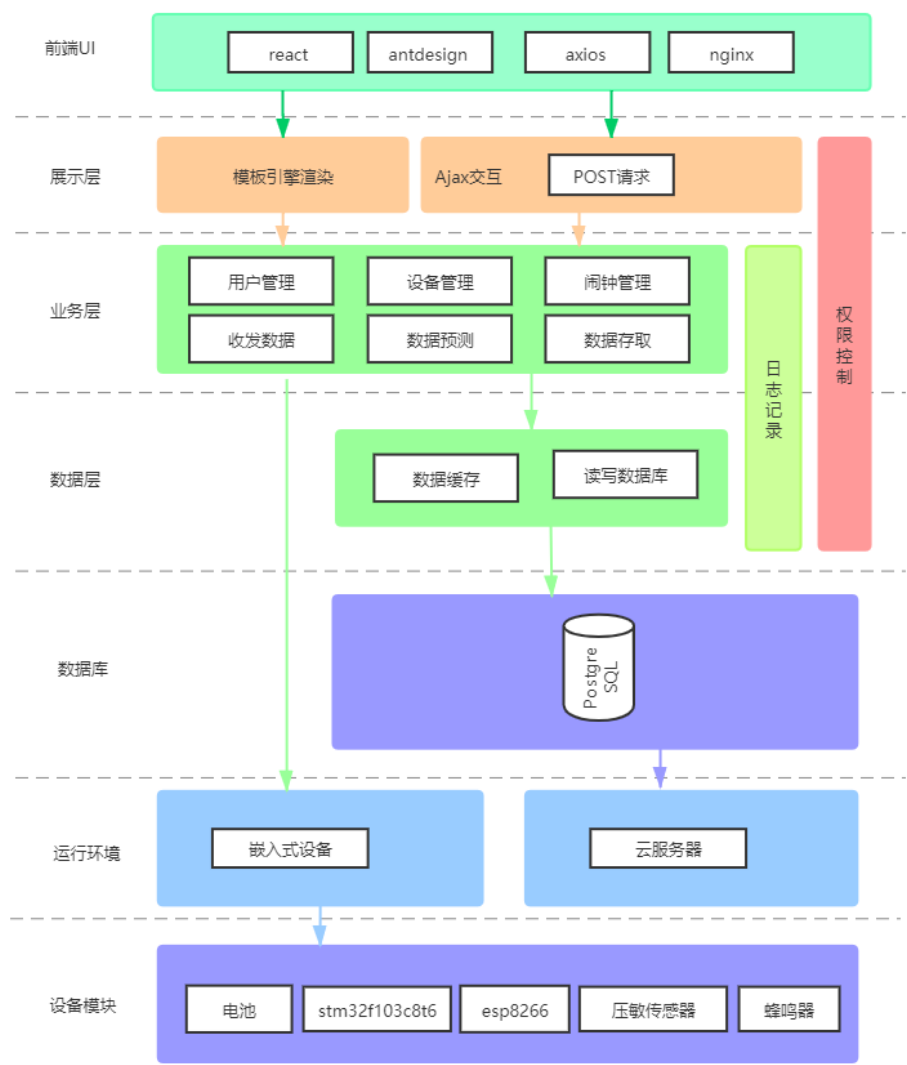


图 2-2 系统架构设计图

2.3 物理架构设计

为了最大化的方便用户使用，比如在手机，笔记本和台式电脑上都可以轻松地访问我们的设置界面，我们通过前端服务器提供了通过 web 界面直接登录并进行设置的功能。同时考虑到在用户增多的情况下，服务器的负载会增加，我们采用了前后端分离的架构，提高了整个系统的承载能力。

整个系统的物理架构设计图如图 2-3 所示。

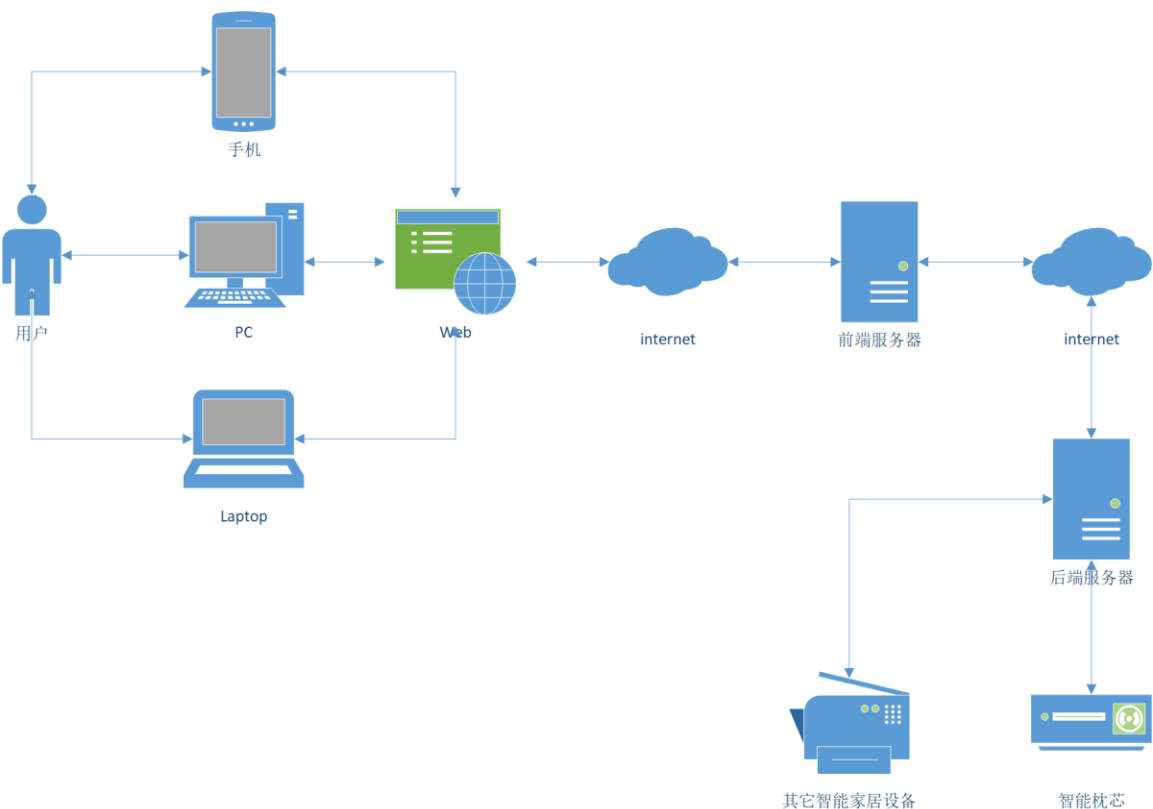


图 2-3 物理架构设计图

第三章 详细设计

3.1 前端设计

3.1.1 前端架构与逻辑设计

我们的前端由 react, antdesign 和 nginx 组成，采用组件化的设计模式，前后端分离的架构，而用户管理则采用 token 的认证模式，因为采用了前后端分离的设计方法，所以我们采取了两次 nginx 反向代理的方式来解决前后端跨域问题，最终我们将 react 和 antdesign 的代码运行在 nodejs 上，最终构建了主页面，登录注册页面，设备和闹钟列表页面，添加与修改闹钟、设备名称页面，数据分析页面，再使用 react-router 路由进行单页面应用中不同页面之间的跳转。

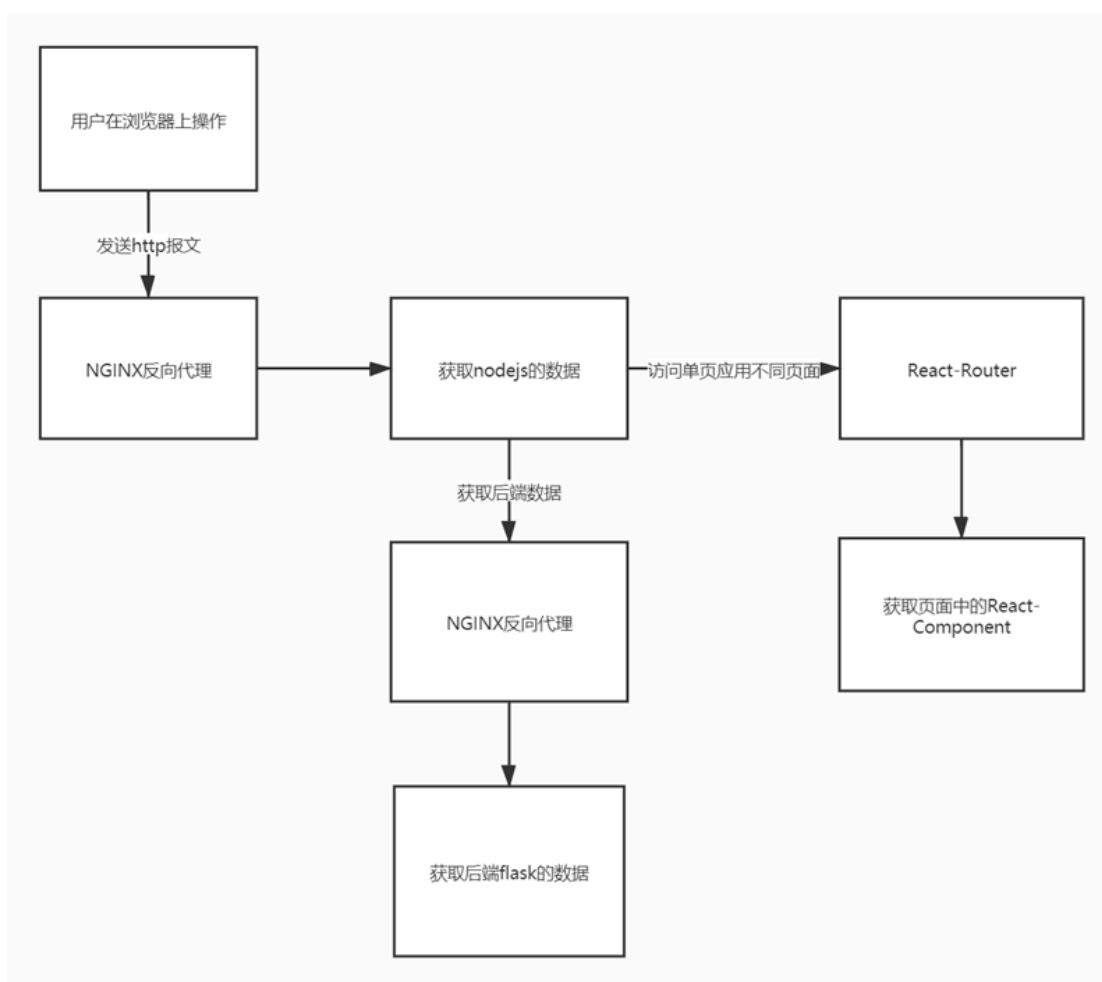


图 3-1-1 前端架构图

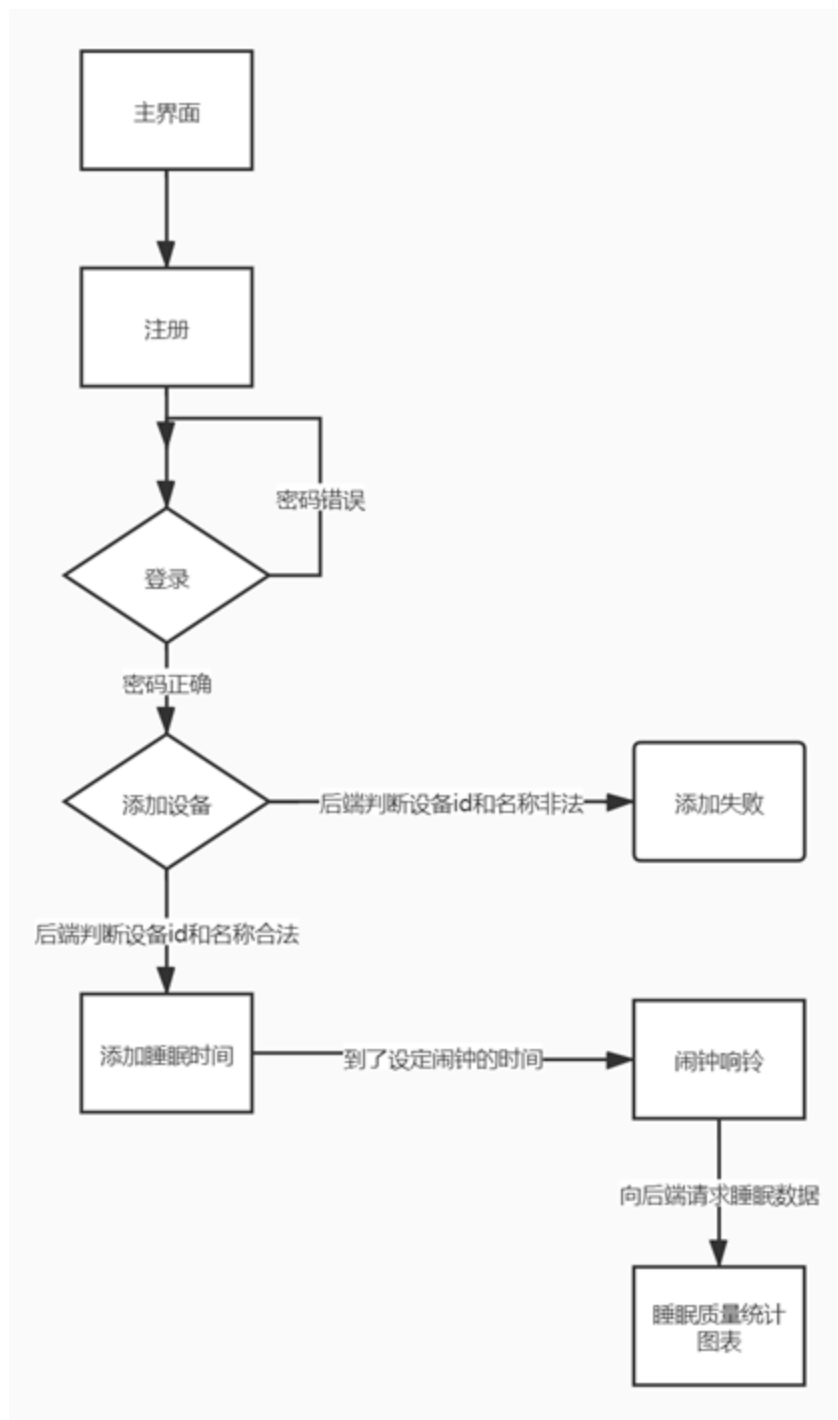


图 3-1-2 前端操作逻辑图

3.1.2 登录注册页面


首先进入主页面，可以看见登录注册按钮



图 3-1-3 主页面的登录和注册

我们点击登录按钮进入登录页面



 tom

 ...|

☐ Remember me

Log in

Or [register now!](#)

图 3-1-4 登录界面

3.1.3 主界面

登录结束以后，自动返回主页面，此时可以看到设备列表、闹钟列表和登出按钮。

Smart Pillow

Welcome,tom

Device Table

Alarm Table

Log Out

图 3-1-5 主界面

3.1.4 设备列表页面

点击 (Device Table) 可以进入设备列表页面，点击 (Alarm Table) 可以进入闹钟列表页面。

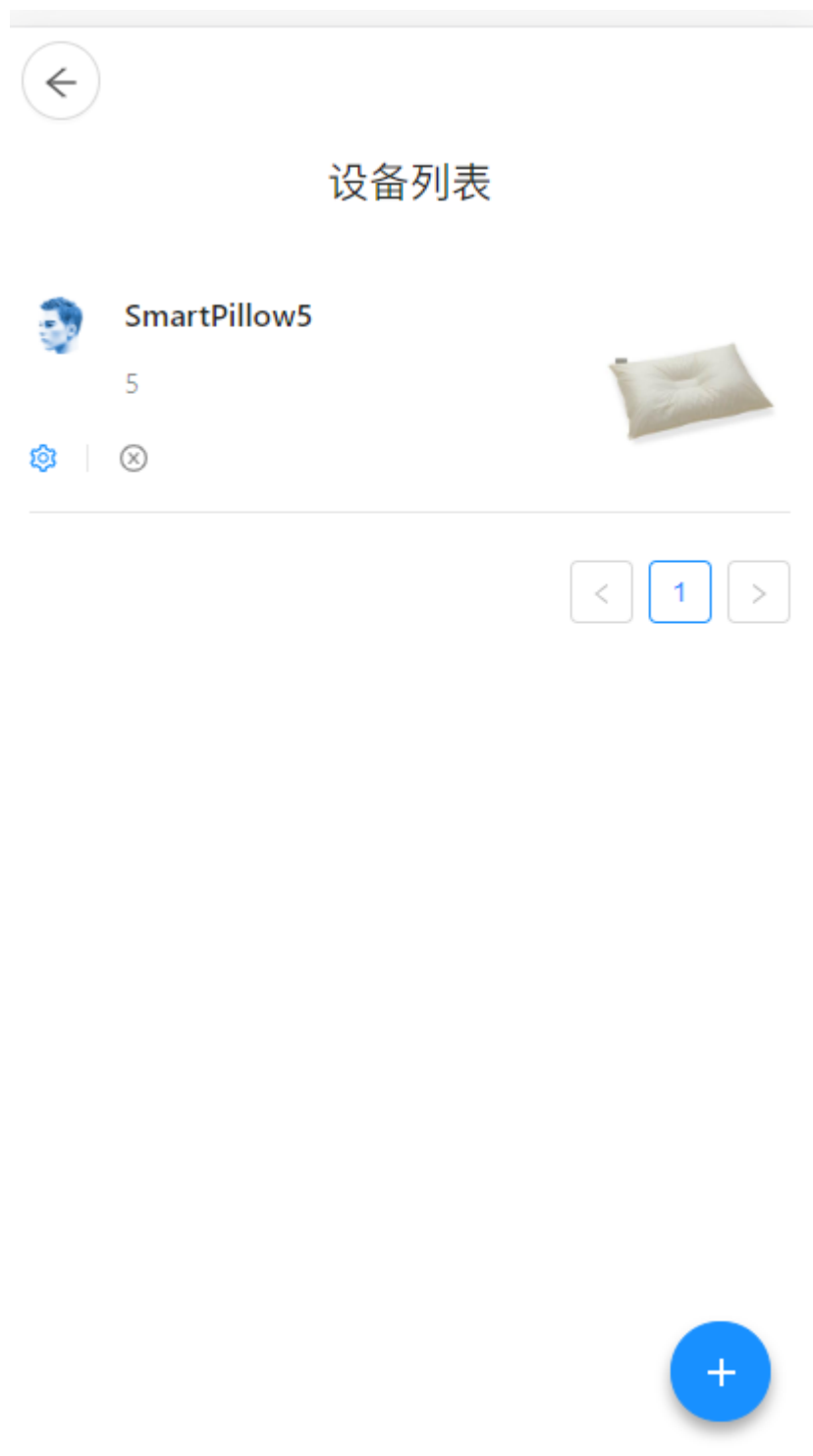
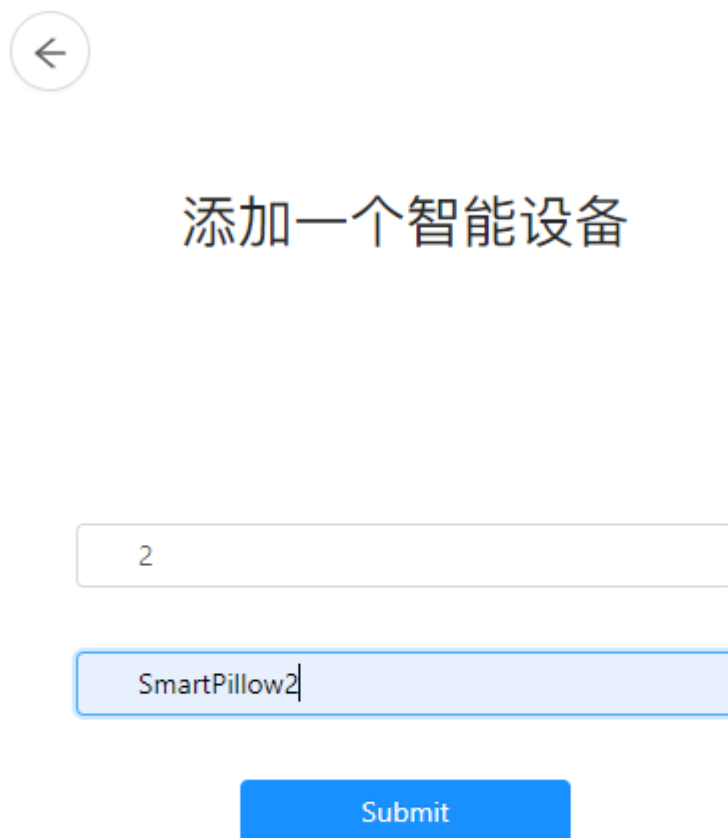


图 3-1-6 设备列表界面

在设备列表页面点击右下角的十号，可以进入添加设备页面



The image shows a mobile application interface for adding a smart device. At the top left, there is a circular button with a left-pointing arrow. Below this, the title '添加一个智能设备' (Add a smart device) is centered in a large, bold font. Underneath the title, there are two input fields. The first field contains the number '2'. The second field contains the text 'SmartPillow2'. Below these fields is a blue button with the word 'Submit' in white text.

图 3-1-7 添加设备页面

3.1.5 闹钟列表页面

点击左上角的返回回到主页面，然后点击闹钟列表页面

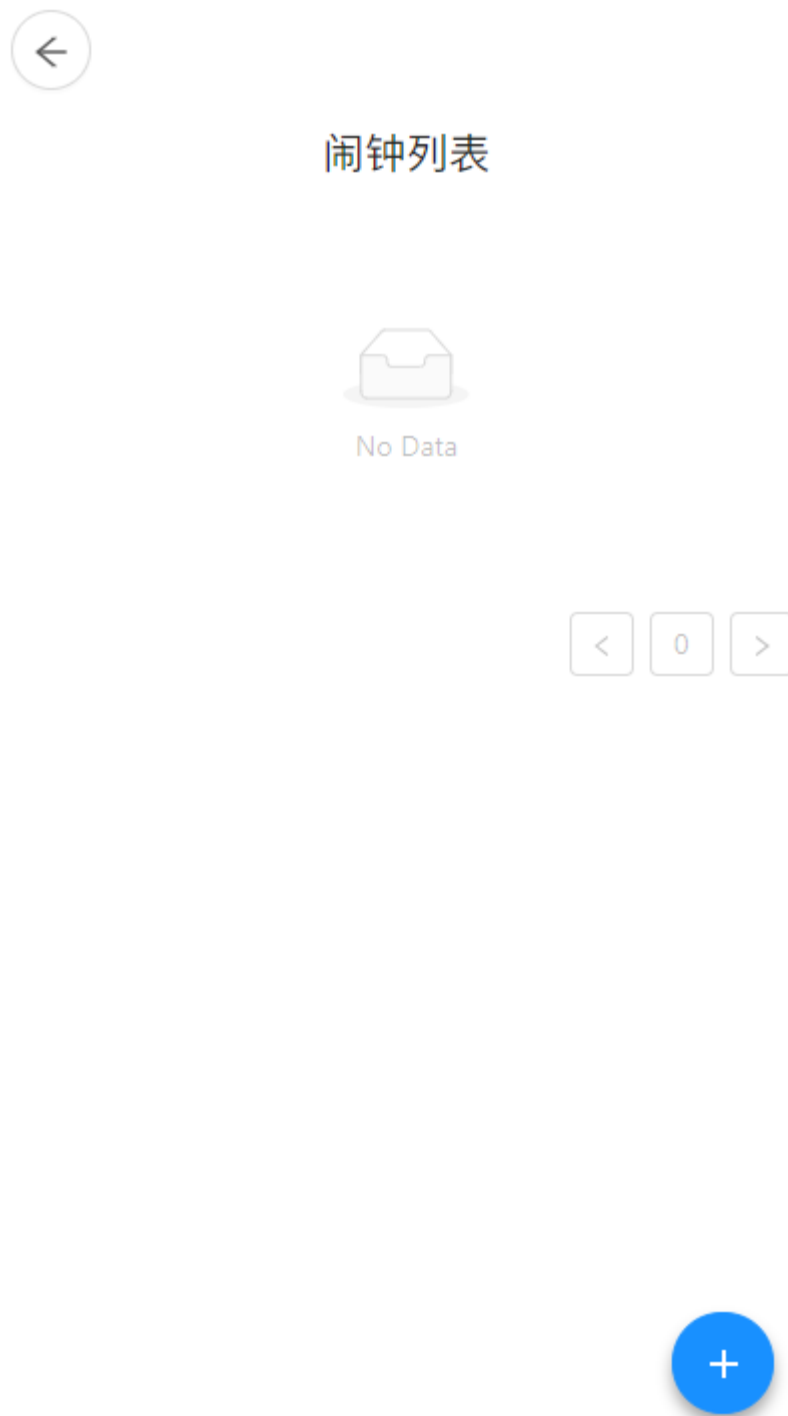



图 3-1-8 闹钟列表页面

接下来我们设置一个闹钟进行测试




设置一个闹钟

* SleepingTime

22:08:38 

* WakeupTime

07:08:40 

Submit

图 3-1-9 设置闹钟页面

返回闹钟列表，可以看到我们设置的闹钟

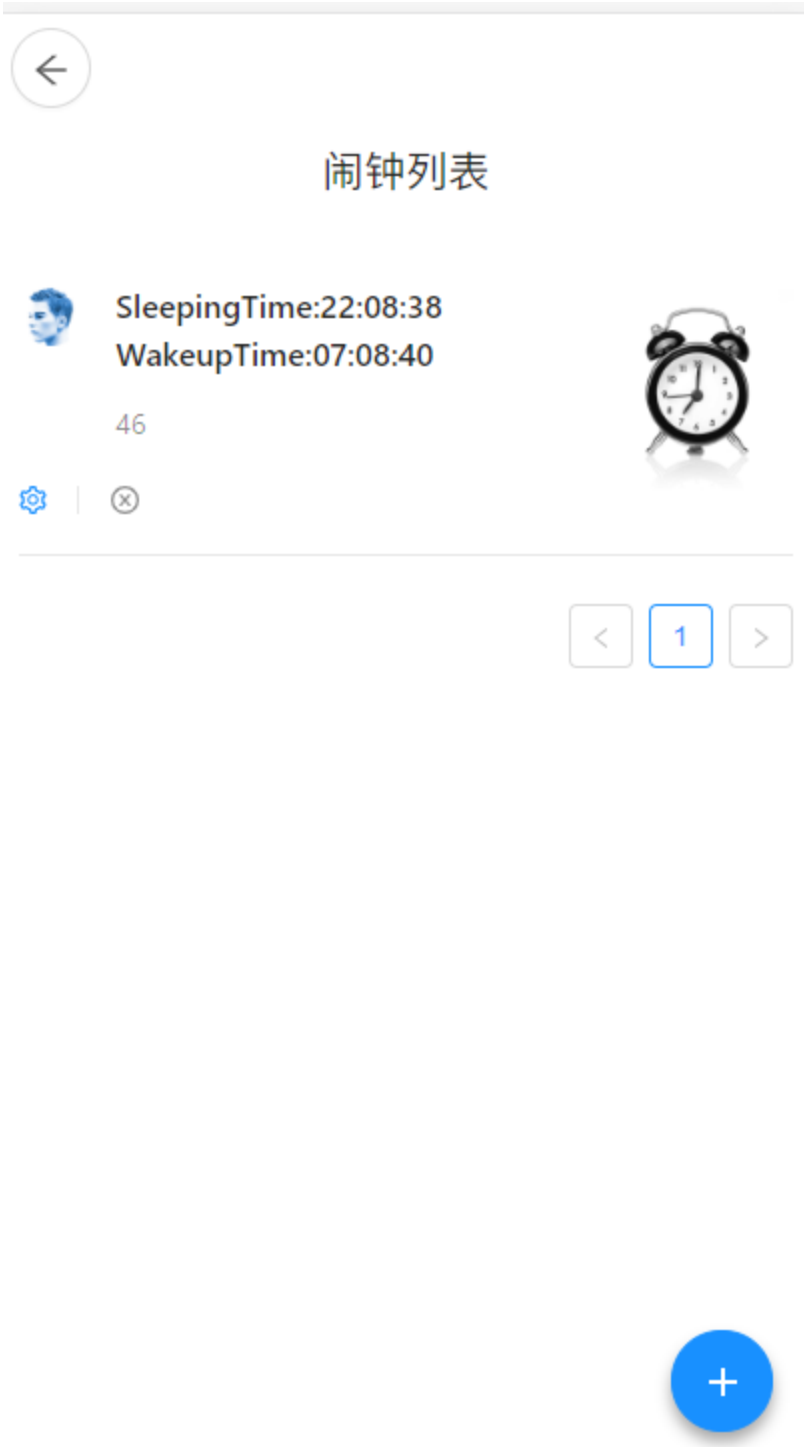


图 3-1-10 闹钟列表页面

3.2 后端设计

3.2.1 总体业务流程

后端服务器主要负责的任务有三个，分别是：

1 处理自硬件的连接请求，并且根据硬件发送的数据做出相应的决策，同时通知对应的硬件使其做出相应的响应。此外还应该在用户更改闹钟信息的时候向硬件推送新的闹钟时间表，确保硬件设备在短时间断网的情况下也可以正常工作。

2 监听前端服务器发送过来的 HTTP POST 请求，根据相应的请求执行相应的业务逻辑函数。此过程中包括了对数据库的访问，在特殊情况下，比如用户修改了闹钟的时间或者是新增加，删除了闹钟的时候，对该用户的硬件设备做出通知。

3 根据硬件设备和前端服务器发来的信息，做出对数据库进行相应的查询，添加，删除和修改的操作。

其总体业务流程如图 3-2-1 所示

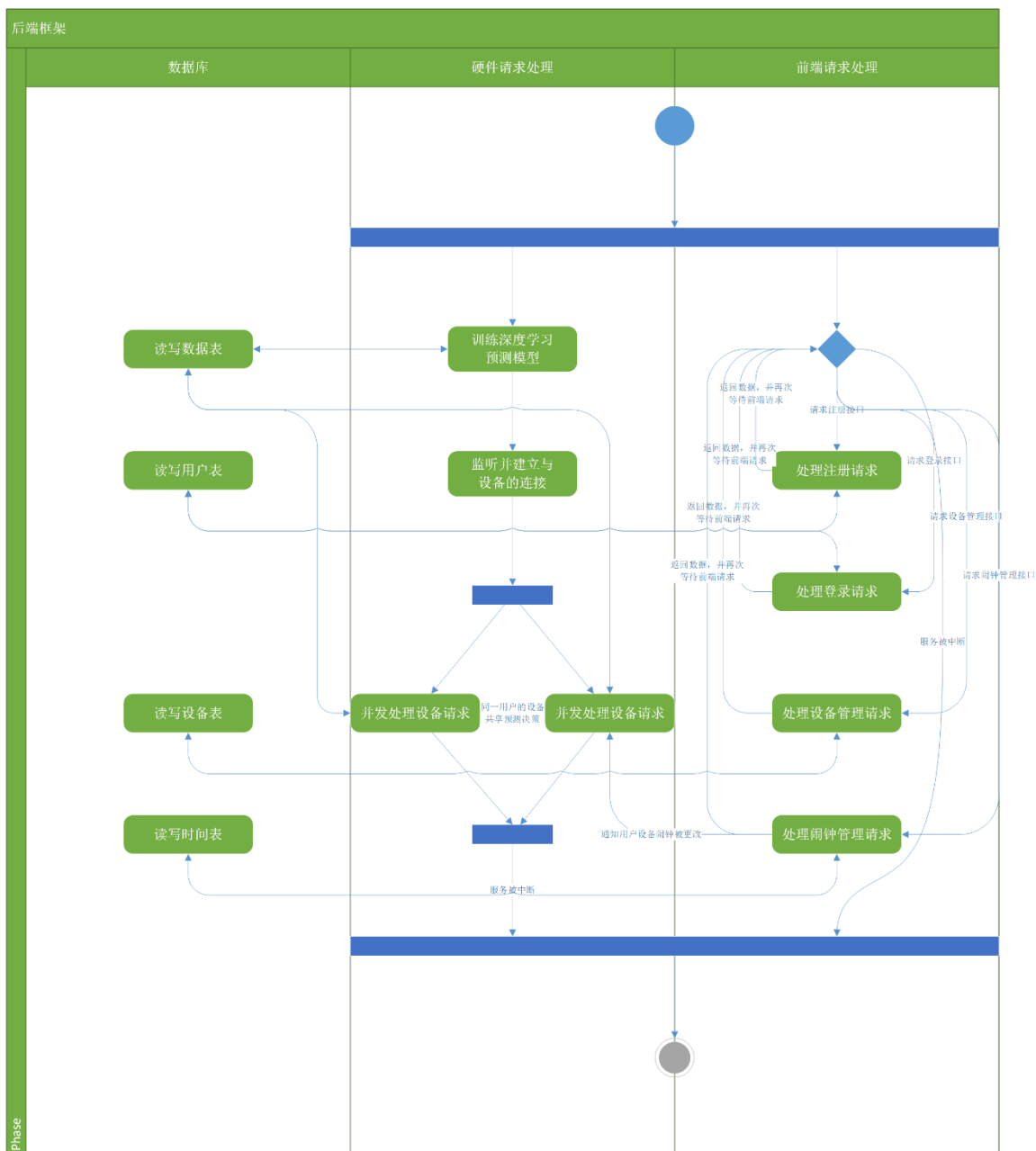


图 3-2-1 总体业务流程

3.2.2 基于深度学习的睡眠状态预测算法

考虑到不同用户可能会使用不同型号的枕头，而且不同用户的睡眠习惯，身体状况不相同，我们的硬件设备通过传感器收集到的数据，即使本应该处于相同的分类，也可能会有着较大的偏差。此外，在人呼吸，或者是翻身的时候，枕芯采集到的压力的变化不便于用常规算法表示，所以我们采用了神经网络算法，直接拟合我们收集标注的训练数据。并利用这些数据做一个科学的预测，来尽可能的降低这些偏差带来的负面影响。

由于我们只收集了压敏传感器这一项信息，建立一个带有 drop out 的循环神经网络就可以满足我们的预测需求。深度学习算法的流程如图 3-2-2 所示。模型在数据输入后，会先让数据通过一个全连接层，扩大数据的维度，以便从更多的角度分析数据。然后通过一个 drop out 层，防止模型的过拟合。之后数据会通过 GRU 单元构成的循环神经网络结构，完成对时序信息的处理，最后会再通过两个全连接层，对数据做一个非线性变换的同时进行预测。深度学习的模型让我们可以忽略数据的预处理，如特征提取等过程，并且在以后扩展的时候只需要添加输入的维度，所以相比于利用常规算法，神经网络算法更具有优势并且适合用在我们的项目上面。同时我们采样数据的频率很低，主要是在数据发生突变的情况下向服务发送数据硬件设备为了节约电量与降低服务器压力。

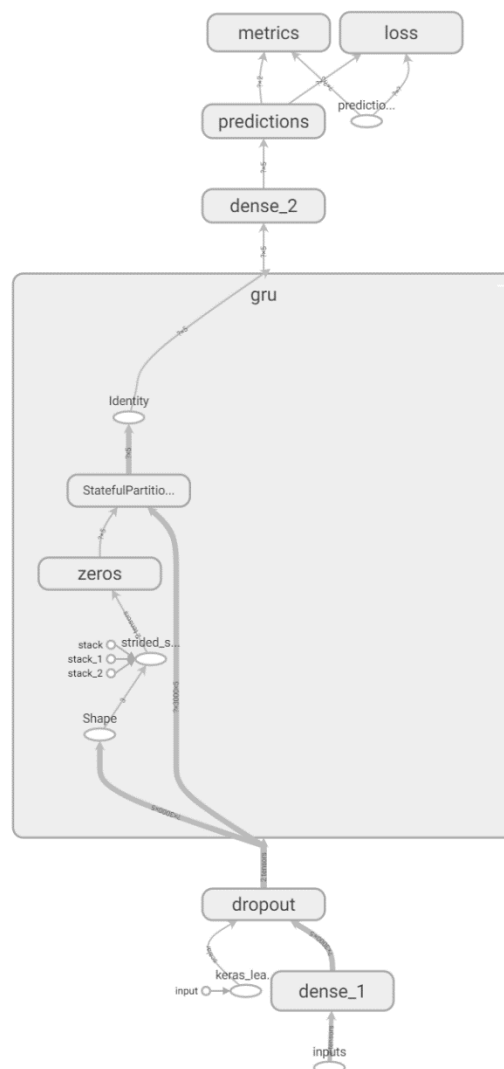


图 3-2-2 深度学习算法流程

3.3 硬件设计

3.3.1 框架

智能枕芯硬件框架以可裁剪、高性价比的思路设计，旨在以有限的资源完成所有的用户功能。使用高性能的 stm32f103c8t6 作为控制器，选择 esp8266 接入家庭网络，从而访问服务器。在外围电路上，我们使用了高敏感度的压敏传感器采集用户数据，唤醒功能上使用震动和大功率电子蜂鸣器，并且留有一定的扩展空间，可在后续迭代中更换为音乐等形式。实物图如 3-3-1。

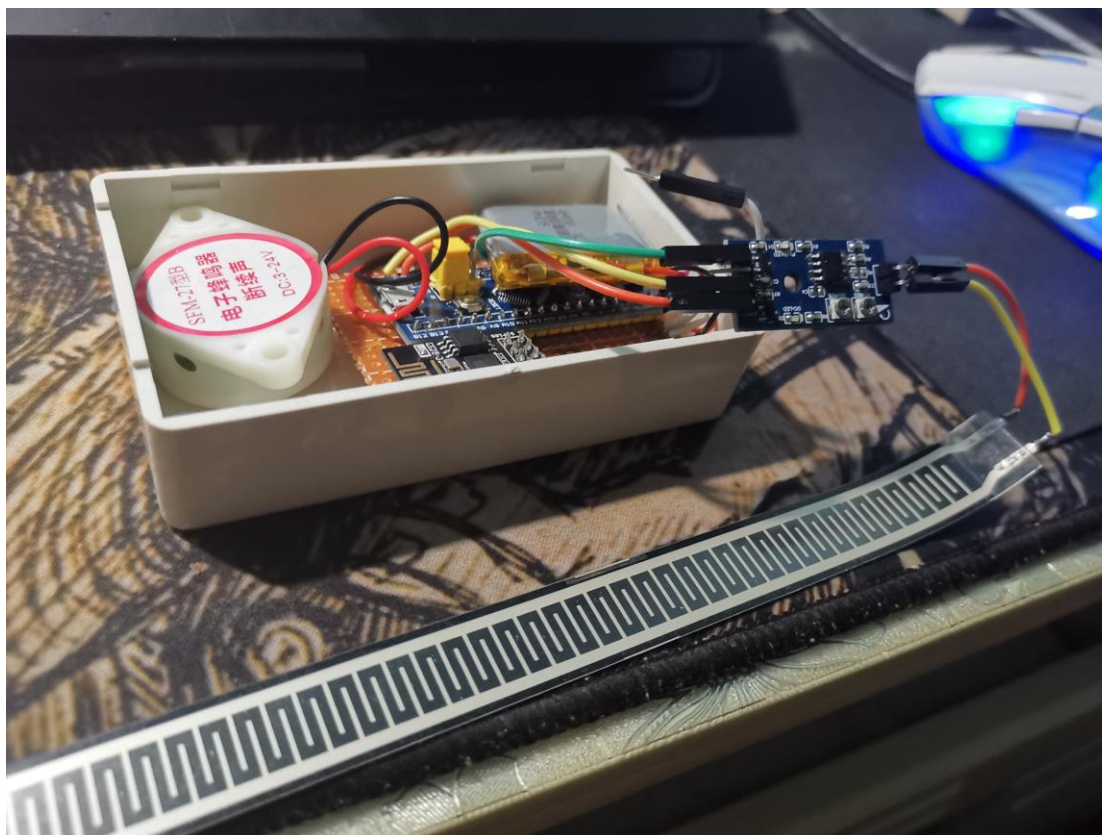


图 3-3-1

3.3.2 电路示意图

整体的示意图如图 3-3-2 所示，stm32 与 esp8266 连接来获得与服务器通信的能力，使用外部充电电池供电，通过模数转换器从传感器获得压敏数据，触发后通过蜂鸣器以及震动提醒用户。



图 3-3-2 电路示意图

stm32f103c8t6 最小系统板引脚如图 3-3-3 所示，使用 GPIOA9 与 GPIOA10 作为 USART 引脚与 esp8266 连接。GPIOA0 与蜂鸣器以及震动装置连接。GPIOA1 作为模数转换的采样引脚，与电压转换模块连接，电压转化模块把传感器的电阻转换为电压输入 GPIOA1。供电使用 VCC3.3 和 GND 连接充电电池。

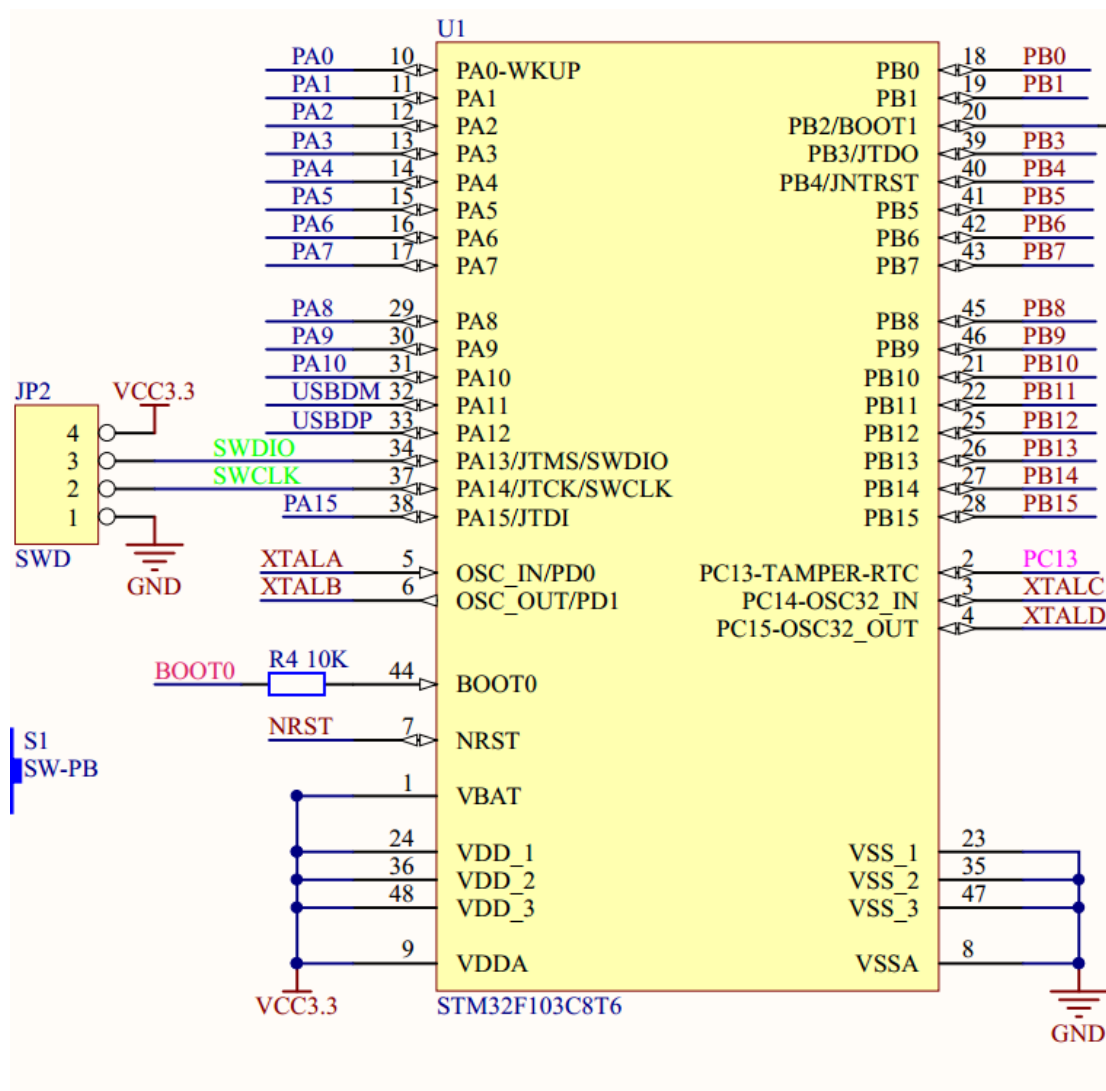


图 3-3-3 stm32 核心板引脚

3.3 嵌入式软件设计

3.3.1 框架

智能枕芯的嵌入式软件基于 stm32 的标准库，通过中断的方式来进行运行事件的触发。外设上，主要使用了 rtc 实时闹钟，TIM 触发器，串口通讯，模数转换以及基本的 I/O 输出功能，并且使用了部分低功耗功能以获得更长时间的待机。考虑到功耗等问题，我们没有使用嵌入式操作系统，而选择了裸板开发。工程展示如图 3-3-4。

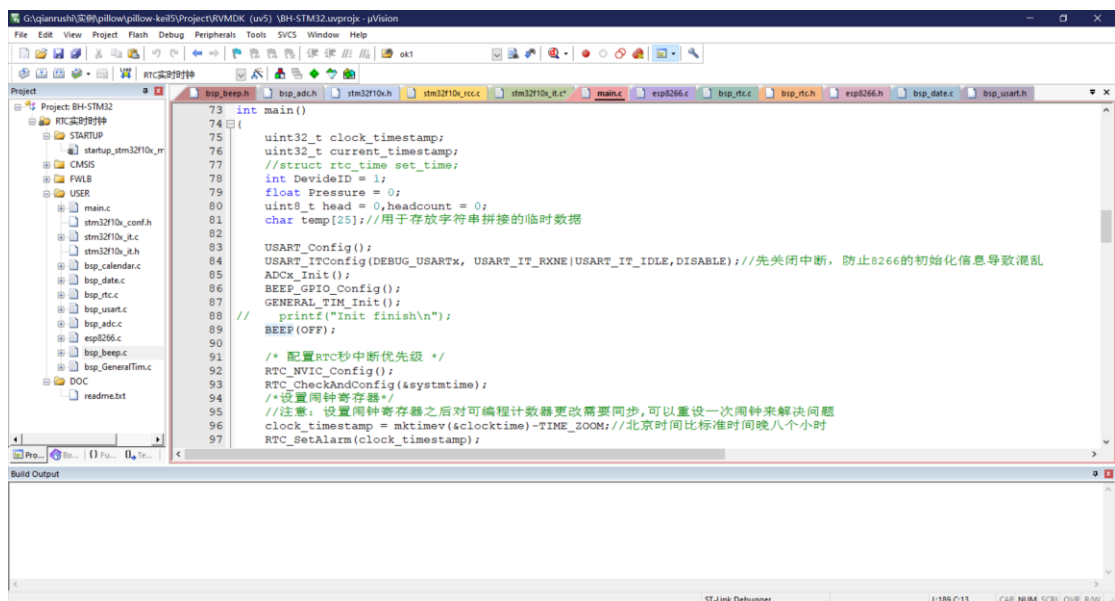


图 3-3-4 工程展示

3.3.2 主要运行流程

首先，stm32 上电之后完成自身串口、ADC、IO、TIM 以及 RTC 的初始化，然后使用 AT 指令，通过串口完成 esp82660 的初始化，进入主循环。

每 5 分钟，stm32 被唤醒一次，然后使用 TCP 发送数据。服务器收到数据后回复，stm32 矫正时间并且判断是否收到新的闹钟。如果收到了新的闹钟，stm32 通过对 RTC 相关寄存器的设置实现本地的闹钟功能，如图 3-3-5。

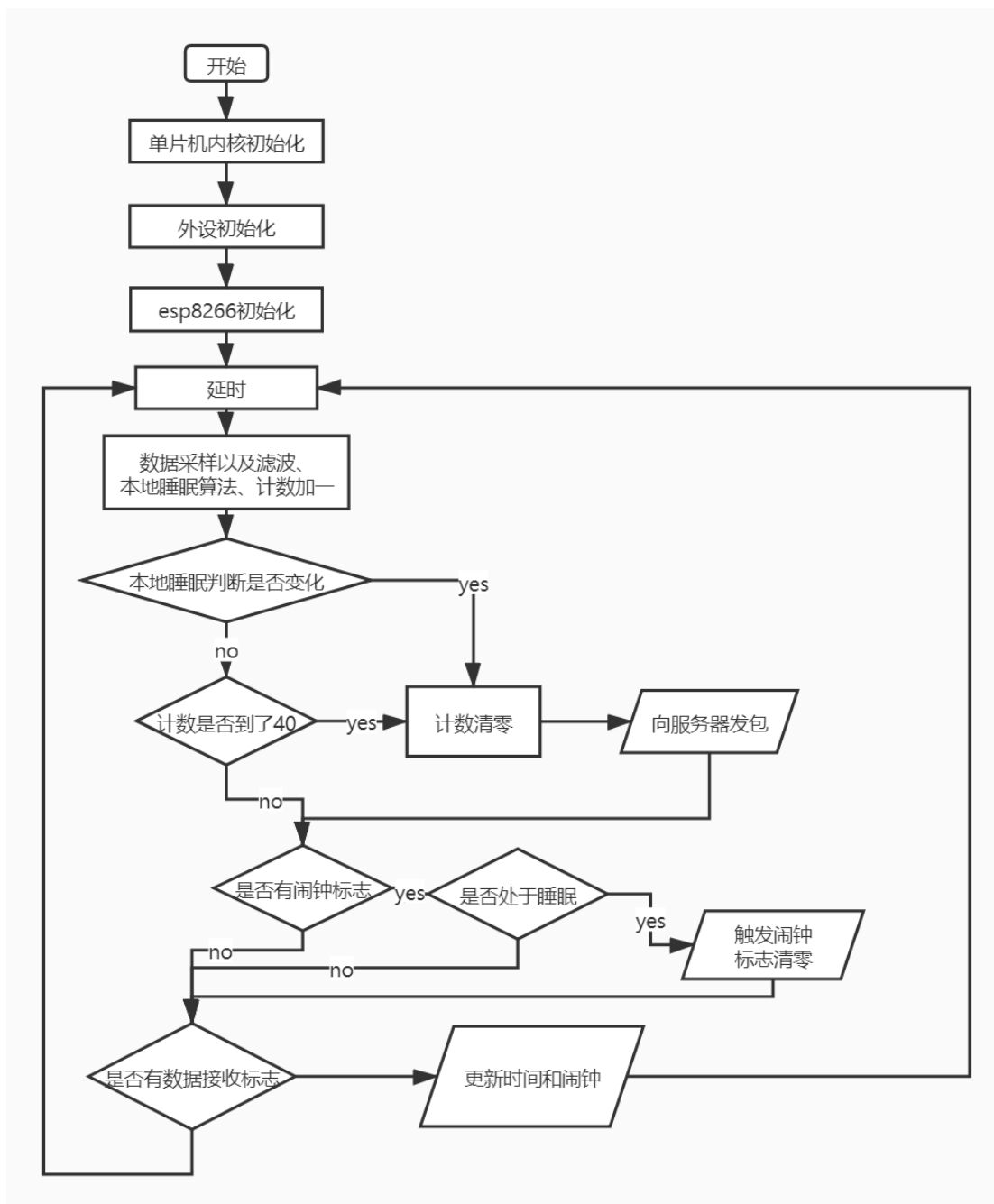


图 3-3-5

当闹钟到时，根据历史数据判断上面是否有人躺着，如果有人则触发蜂鸣器和震动直到离开枕头三次计数为止。如下图：

计数值	0	1	2	3	4	5	6	7	8
是否处于睡眠	无	无	无	有	有	有	有	有	有

当 stm32 接收到来自串口的信息时，通过中断的方式读取不定长数据，并且触发特定函数，对接收到的 json 进行手动处理（因为 cJSON 所需要的堆对于这款芯

片来说太奢侈了，而且 json 包也不大，权衡之下选择手动处理），包括设备 ID 的判断，接受来自服务器的对于是否处于睡眠状态的判断，校准时钟以及设置闹钟。
如图 3-3-6。

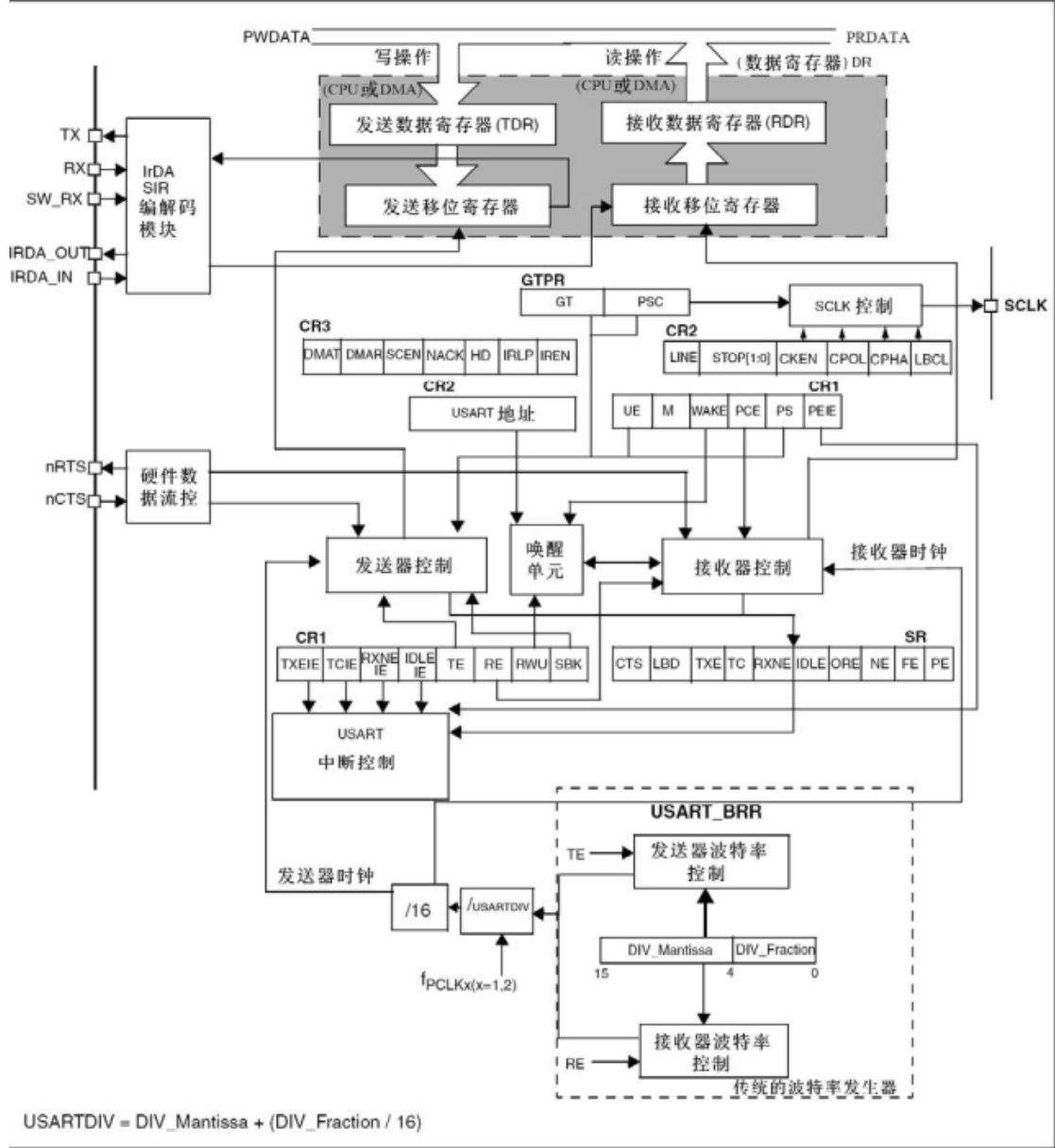


图 3-3-6 串口框图

stm32 的 rtc 框图如下图 3-3-7，我们设置 RTC_CNT 寄存器来设置本地时钟，通过设置 RTC_ALR 寄存器来实现闹钟的触发功能。

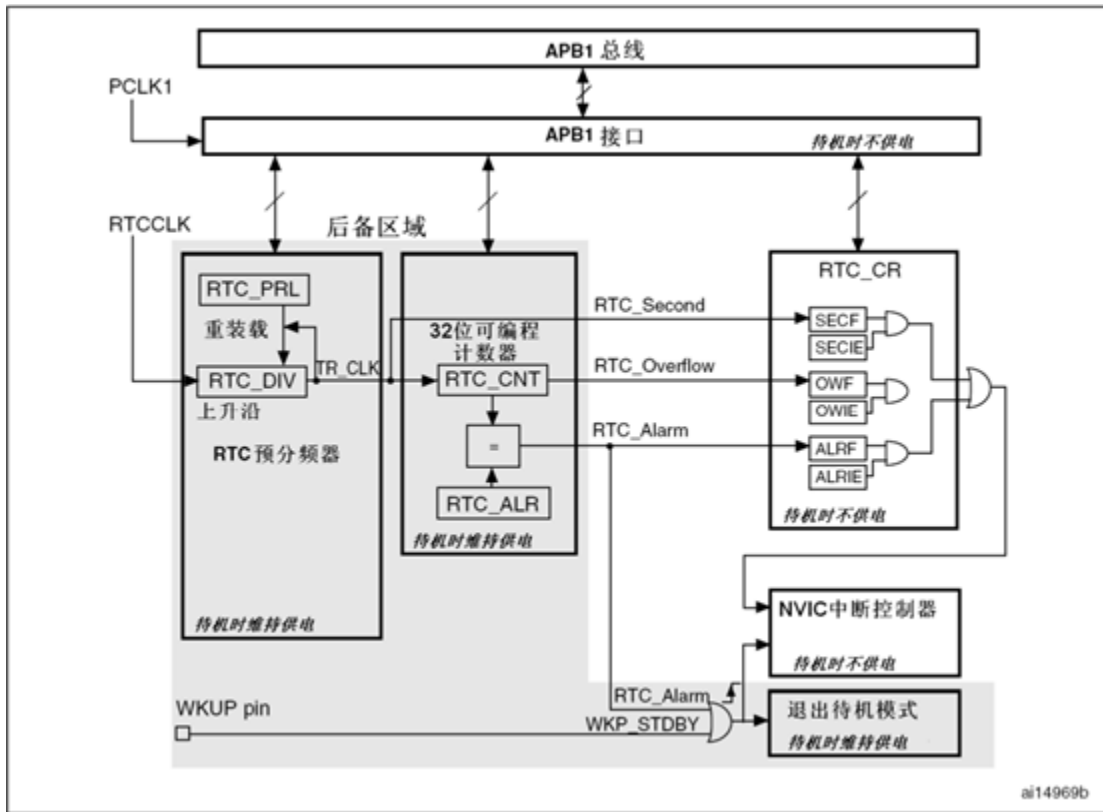


图 3-3-7 rtc 寄存器

我们在嵌入式平台上的设计过程中大量使用了中断以提高系统的实时性，可以对串口信息等外界输入快速做出响应，完成必要的工作，然后用标志位的方式把主要的程序工作放在主程序里面执行，防止在一个中断里的任务过多导致无法快速响应其它中断。

3.3.3 本地睡眠算法以及滤波设计

基于寄存器移位思路设计睡眠滤波算法是一个可以体现嵌入式设计专业素养的原创部分，我们使用一个 uint_8t 类型的变量来模拟一个寄存器，通过移位的方式完成每一次采样后的更新。只有在连续多次采样到相同数据时我们才会确定状态的变化。基于这种算法，在本地端，我们可以非常准确地判断睡眠的状态，实测结果也的确如此。如图 3-3-8。

优点：

- （1）时间代价低，生成的汇编指令很短，高效率。
- （2）空间代价低，内存开销极少，仅一个 uint_8t。
- （3）高容错率，经测试高准确率。

```

1.  if(time>500)//0.5s 一次处理
2.      {
3.          NewAdcData=ADCSTA;
4.          databuf[datacount++]=NewAdcData;//更新数据
5.          time = 0;
6.          headcount = headcount>>1;//headcount 是一个 8 位的软件寄存器
7.          if(NewAdcData>0.07)headcount |= 0x80;
8.          if(headcount>0xE0){//0b11100000
9.              if(head==0){
10.                  CreateJson(databuf);
11.                  for(int i=0;i<40;i++)databuf[i]=0;
12.                  datacount=0;
13.              }
14.              head=1;
15.          }
16.          else if(headcount<0x10){//0b00010000
17.              if(head==1){
18.                  CreateJson(databuf);
19.                  for(int i=0;i<40;i++)databuf[i]=0;
20.                  datacount=0;
21.              }
22.              head=0;
23.              BEEP(OFF);
24.          }
25.      }

```

图 3-3-8

3.3.4 其它细节设计

(1) 在 wifi 连接上，我们设想的方案是使用一个专用的串口上位机，后续将 esp8266 设计成可插拔的模式，在产品中加上一个串口转换模块（3 元左右），用户可以通过串口的方式，从电脑端快捷地为产品设置自己家的 wifi。

指令	查询指令： AT+CWJAP? 功能：查询 ESP32 Station 已连接的 AP 信息。	设置指令： AT+CWJAP=<ssid>,<pwd>[,<bssid>] 功能：设置 ESP32 Station 需连接的 AP。
响应	+CWJAP:<ssid>,<bssid>,<channel>,<rssi> > OK	OK 或者 +CWJAP:<error code> ERROR
参数说明	<ul style="list-style-type: none"> • <ssid>: 字符串参数, AP 的 SSID • <bssid>: AP 的 MAC 地址 • <channel>: 信道号 • <rssi>: 信号强度 	<ul style="list-style-type: none"> • <ssid>: 目标 AP 的 SSID • <pwd>: 密码最长 64 字节 ASCII • [<bssid>]: 目标 AP 的 MAC 地址, 一般用于有多个 SSID 相同的 AP 的情况 • <error code>: (仅供参考, 并不可靠) <ul style="list-style-type: none"> ▸ 1: 连接超时 ▸ 2: 密码错误 ▸ 3: 找不到目标 AP ▸ 4: 连接失败 ▸ 其他值: 未知错误 <p>参数设置需要开启 Station 模式, 若 SSID 或者 password 中含有特殊符号, 例如, 或者 " 或者 \ 时, 需要进行转义, 其它字符转义无效。</p>
提示信息	// If ESP32 station connects to an AP, it will prompt messages: WIFI CONNECTED WIFI GOT IP // If the WiFi connection ends, it will prompt messages: WIFI DISCONNECT	
注意	本设置将保存在 NVS 区	
示例	AT+CWJAP="abc","0123456789" 例如, 目标 AP 的 SSID 为 "ab\,c", password 为 "0123456789"\", 则指令如下: AT+CWJAP="ab\\,c","0123456789\\"" 如果有多个 AP 的 SSID 均为 "abc", 可通过 BSSID 确定目标 AP: AT+CWJAP="abc","0123456789","ca:d7:19:d8:a6:44"	

图 3-3-9 wifi 连接指令

(2) esp8266 的 AT 指令细节: 支持将 wifi 信息以及服务器信息保存到本地 flash, 因此我们只需要修改一次就可以长期作用。

设置指令	AT+CWAUTOCONN=<enable>
响应	OK
参数说明	<enable>: ▶ 0: 上电不自动连接 AP ▶ 1: 上电自动连接 AP ESP32 Station 默认上电自动连接 AP。
注意	本设置保存到 NVS 区域。
示例	AT+CWAUTOCONN=1

图 3-3-10 wifi 配置保存指令

(3) 功耗问题：根据乐鑫的官方文档，esp8266-01s 的平均工作电流为 12mA，如果在待机模式下，可以降低到 200uA 一下，因此，设计唤醒机制可以大大降低系统的功耗，在用户睡眠状态相对稳定并且离闹钟时间较远时进入低功耗状态可以获得更长的待机时间。

硬件参数	硬件接口	UART, IIC, PWM, GPIO, ADC
	工作电压	3.3V
	GPIO驱动能力	Max: 15mA
	工作电流	持续发送下= 平均值: ~70mA,峰值: 200mA 正常模式下= 平均: ~12mA,峰值: 200mA 待机: <200uA,
	工作温度	-40℃~125℃
	存储环境	温度: <40℃, 相对湿度: <90%R.H.
	尺寸	板载 PCB 天线: 14.3mm*24.8mm*1mm;

图 3-3-11 官方硬件手册功耗参数

(4) 供电问题：成熟的方案应该是使用 AMS1117 降压，从而获得稳定的 3.3V 电压为 stm32 以及 esp8266 供电，电池本身也是可以充电的，因此只需要在盒子上开一个口，通过 stm32 最小系统板上面的 microusb 即可对电池充电。

3.3.5 成本估计

stm32c8t6 最小系统板 8.6 元，如果是芯片的话只需要 5.15 元（淘宝价），esp8266-01S 淘宝价 5.8 元，薄膜压敏传感器 49 元，传感器配套模块 14.8 元，震动以及蜂鸣器合计 5 元，锂电池 6.4 元，其余杂项硬件材料费用 3 元，合计 92.6 元。如果能够大量生产的话，通过更换传感器以及定制 PCB，预计成本可以到 40 元。

第四章 测试报告

4.1 本地测试

4.1.1 传感器读数以及串口测试

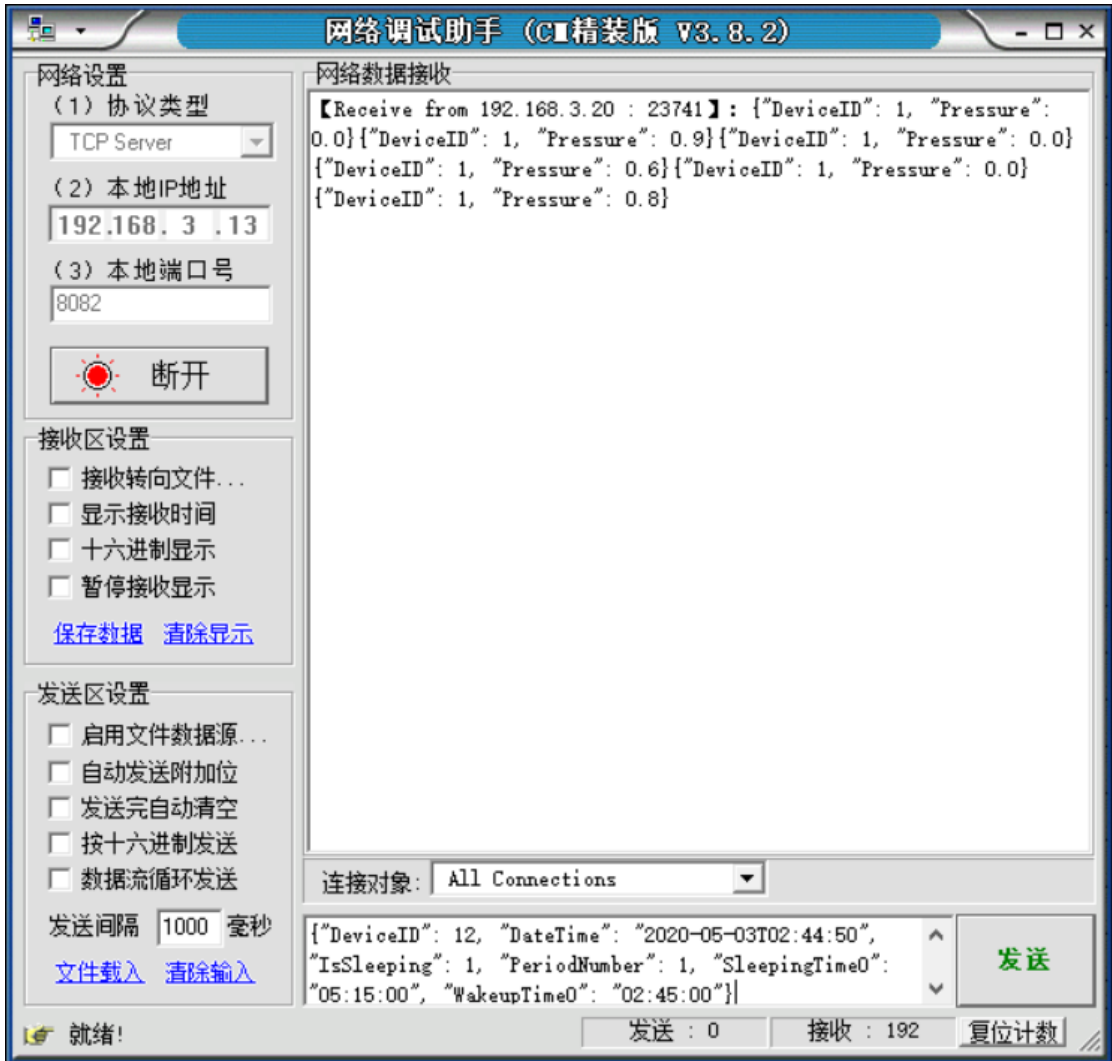


图 4-1 串口测试

将 8266 连接到本地搭建的调试网络，可以收到定时发送的以及睡眠情况确定改变时的传感器数据

4.1.2 闹钟测试

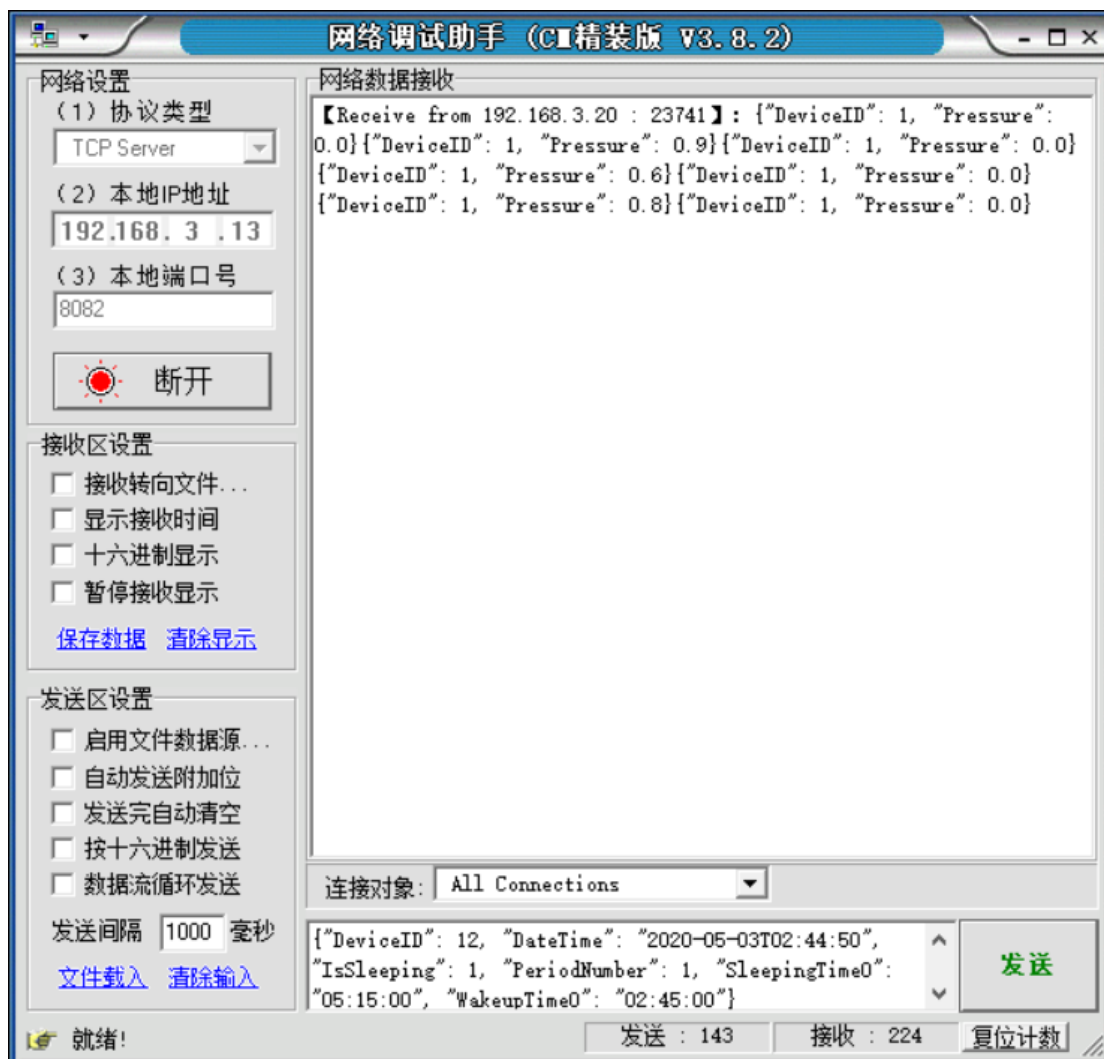


图 4-2 闹钟测试

发送当前时间并设置一个十秒之后的闹钟，发现当传感器上有持续压力时，蜂鸣器响了起来，无压力后，蜂鸣器和传感器延时关闭。

4.1.3 多闹钟测试



图 4-3 多闹钟测试

设置了一分钟后和两分钟后两个闹钟之后，蜂鸣器如我们所料地在正确的时间两次响起，松开传感器几秒后停止。

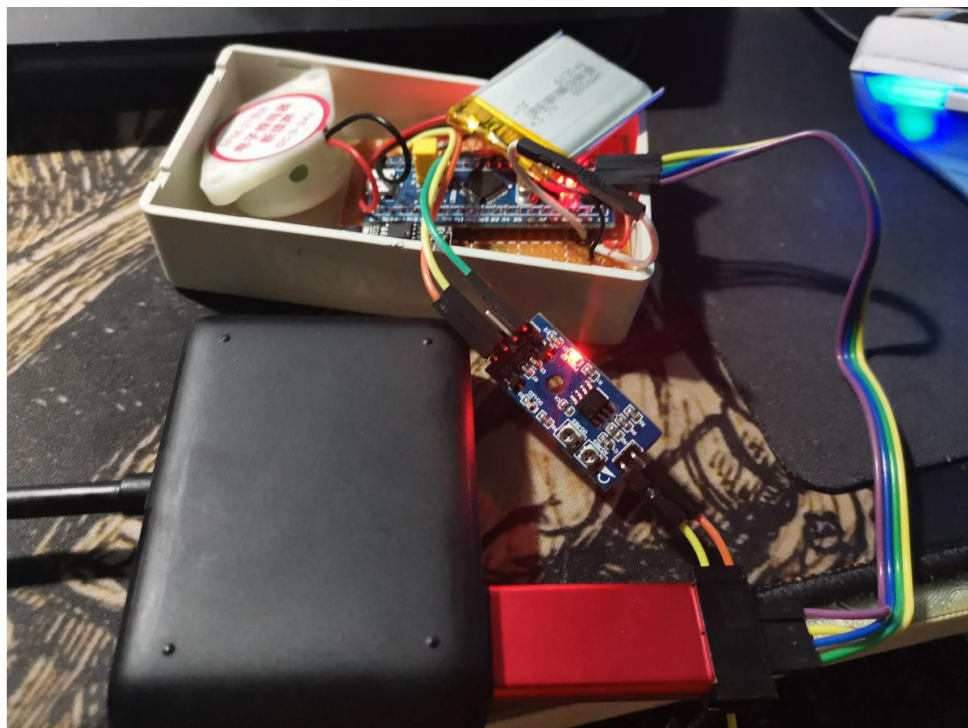


图 4-4 本地测试中的硬件

4.1.4 与其它物联网设备联动



图 4-5 联动功能

开启联动功能之后，当头放在枕头上时，智能灯关闭，头抬起来时智能灯打开。

4.2 前后端测试

4.2.1 账户相关测试

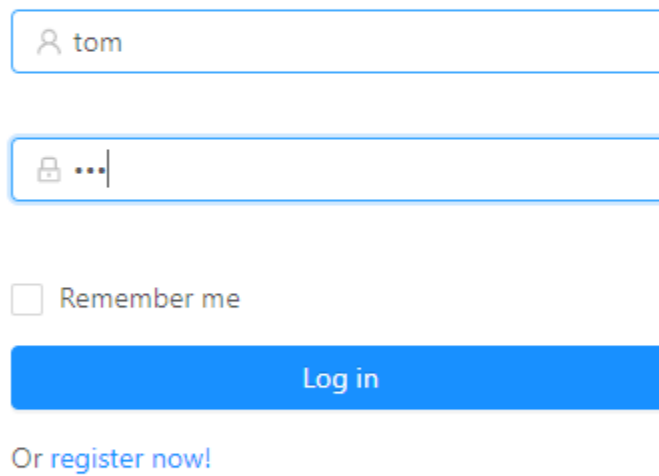
A login form with two input fields. The first field contains the text 'tom' preceded by a person icon. The second field contains three dots preceded by a lock icon. Below the fields is a checkbox labeled 'Remember me'. At the bottom is a blue 'Log in' button and a link that says 'Or register now!'.

图 4-6 登录测试

点击登录按钮，登录成功

Smart Pillow

Welcome,tom

Device Table

Alarm Table

Log Out

图 4-6 登录成功页面

点击登出按钮，进行登出测试

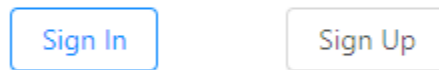


图 4-7 登出成功页面

点击 (Sign Up), 进行注册测试

A registration form with a light grey background. At the top left is a circular back button with a left-pointing arrow. Below it are three form fields, each with a red asterisk and a label: '* Username' with a text input containing 'ccdragon'; '* Password' with a password input (dots) and a toggle icon (eye) and a green checkmark; '* Confirm Password' with a password input (dots) and a toggle icon (eye) and a green checkmark. The 'Confirm Password' field is highlighted with a blue border. At the bottom is a blue 'Register' button.

图 4-8 注册页面

点击注册按钮，然后以新的账户进行登录

Smart Pillow

Welcome,ccdragon

Device Table

Alarm Table

Log Out

图 4-9 登录成功，注册测试完毕

4.2.2 全测试

我们以 tom 作为测试账户，前端后端和硬件共同完成一个闹钟响铃的测试

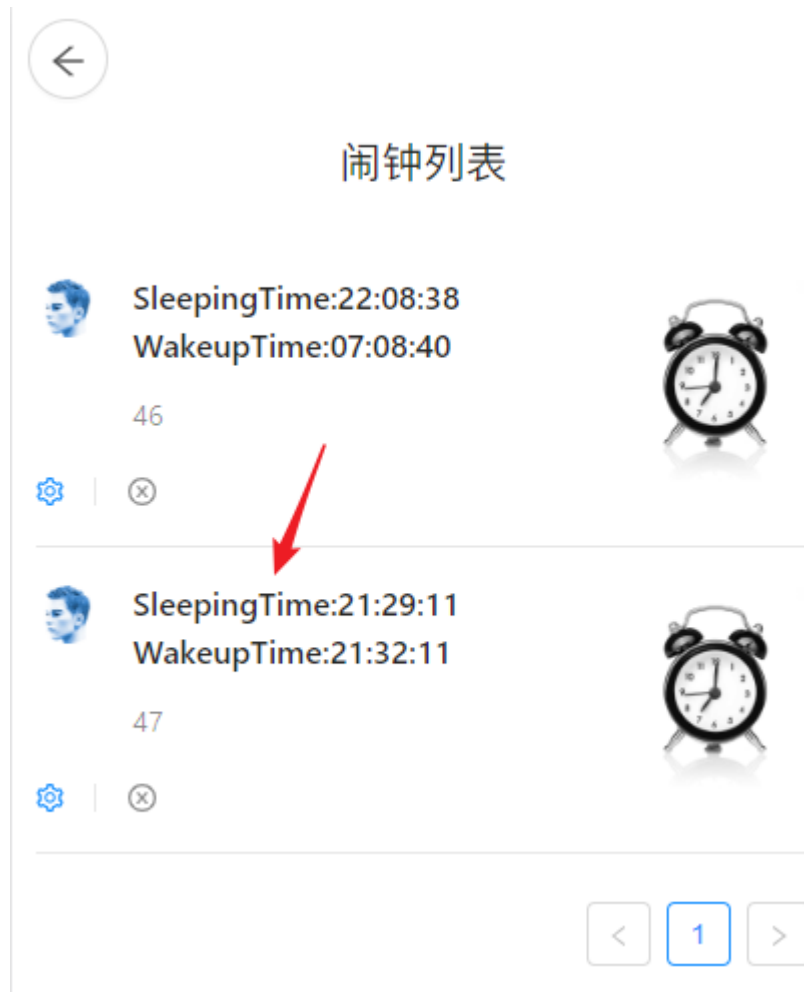


图 4-10 设置一个真实时间一分钟后的闹钟



图 4-11

通过在网页设定一个真实时间一分钟后的闹钟，将头压在枕头上，一分钟后蜂鸣器响了起来，起来之后几秒闹钟停止。

第五章 安装及使用

安装环境：本产品中，智能枕芯作为嵌入式设备无需安装。而管理系统也可以通过浏览器直接访问，无需安装。

使用流程：

- 1) 打开浏览器，访问 <http://smartpillow.jimheisenberg.xyz:89/>即可转到首面。
- 2) 在首页，点击 sign up，输入用户名与密码，点击 register，即可注册。注册成功后，回自动跳转到登录页面，输入之前的用户名，密码，点击 log in，即可登录并转到个人主页。
- 3) 在个人主页，点击 Device Table 即可转到设备管理页面，在其中可以显示，添加，删除，重命名设备。
- 4) 在个人主页，点击 Alarm Table 即可转到闹钟管理页面，在其中可以显示，添加，删除，更改闹钟。
- 5) 在个人主页，点击 log out 即可退出。

第六章 项目总结

本项目原本是工作室的一个老是睡过的老哥提出来的问题。他设了闹钟，但有几次在意识还不清醒的时候拍掉闹钟，导致睡过头错过上课。我们针对他的问题设计出来了这样的一套解决方案，如果你意识不清醒的时候拍掉了手机的闹钟，到了最后时限，如果你还在睡，那么这个智能枕芯就会无情地把你轰起来。如果你早就起床干别的事情去了，智能枕芯也不会打扰你。所以说呢，就适用性而言，我们的作品是立足于解决现实问题进行设计的，体现了工科生的基本素养：技术，为改变生活。

在设计的过程中，我们的团队在讨论之后确定了合理而明确的分工：硬件、前端、后端。在开发的过程中，我们从最开始的 API 确定开始，一点一点螺旋迭代到现在这样一个完整的作品，过程自然是充满趣味的，也能够锻炼人的内在表征。比如在调试全过程的时候，硬件始终收不到数据，但是能够发送数据，开发硬件的同学和后端的同学沟通寻找问题的时候，硬件同学本来觉得能发数据但是不能收数据，不太可能是硬件上的问题，大概率是软件上的。但是沟通的过程中突然想起 usart 是全双工通讯的，于是看了看焊接，发现是引脚焊错了，把本应该连 PA10 的连接到了 PA11 上，导致数据只能发不能收，问题明确之后就很简单了。渐渐地，随着前后端越来越完善，从用户到底层，我们体现了一个完整的过程。

当然，这个作品目前还不是完全体形态，还有很多东西可以迭代，比如机器学习的算法，通过历史数据可以对睡眠情况进行一个评估并反馈给用户；再比如硬件的迭代：成本降低、绘制 PCB 进行定制化以及增加音乐模块，把简单粗暴的蜂鸣器换下去；甚至还可以玩点更花的，比如加一个简单的语音助手，类似于小爱同学，这样很有可能获得用户的好感。这些实现理论上也不是很困难，只是需要一些时间。还有很多好玩的点子可以在后续的迭代中体现出来。窃以为在这个年轻人睡觉越来越晚的时代，这个想法是有商业化空间的。如果能够得到支持，后续能够做成什么样子是值得期待的。

总而言之，这次设计的目的算是达到了，我们实现了我们想要实现的东西，并且在这个过程中认识了新的人，学习了新的技能，沟通变得更加高效，这是一个软硬实力一起成长的过程。感谢一路上为我们提供帮助的人，也感谢我们的指导老师廖老师。

第七章 附录

7.1 项目任务书

电子科技大学大学生创新创业训练计划项目任务书

为了保证大学生创新创业训练计划的正常实施，确保经费的有效使用，学校教务处按照《电子科技大学大学生创新创业训练计划项目实施办法》要求与获得资助的项目负责人特订立本任务书，双方共同遵照执行，自签字之日起生效。

项目编号：202010614410

项目名称：智能枕芯

项目负责人：龚奕铭 学号：2018091609026

项目成员：来翔(2018091620028)、侯泰宇(2018091613025)

指导教师：廖勇 所在学院：信息与软件工程学院（示范性软件学院）

项目预计完成时间：2020 年 10 月

提交项目结题验收表等时间：项目负责人保证于 2020 年 9 月 30 日前向教务处提交项目结题验收表报告、结题论文及其他成果实物。

拟提供成果及成果形式：论文、实物作品、软件、设计文件、竞赛获奖

批准经费（元）：2000

项目研究进度安排：立项至中期检查：硬件的设计，购买，组装，驱动，和与云服务器的通讯

中期到结题验收：完成具有基本功能的初代产品

本项目组承诺：1、保证做到节约开支、专款专用，报销时经学院审核，由学院统一组织报销；2、保证按进度提交材料，参与中期检查和结题验收；3、本项目所形成的固定资产、无形资产等按照国家有关规定和学校的相关规定执行。

项目负责人签字：龚奕铭