

归 档 号：

武汉工商学院毕业论文(设计)

学院： 人工智能学院

专业： 物联网工程 年级： 2017 级

学生： 学号：

指导教师： 职称：

题目： 基于 STM32 的运动监测系统的设计与实现

2021 年 5 月 7 日

武汉工商学院

本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年 月 日

目 录

摘 要	I
关键词	I
Abstract.....	II
Key words.....	II
1 绪论	1
1.1 系统设计背景	1
1.2 系统设计意义	1
1.3 系统功能设计要求	2
2 系统硬件设计	3
2.1 单片机核心电路设计	3
2.2 心率采集模块电路设计	5
2.3 LM393 滤波模块	5
2.4 运动状态检测模块电路设计	6
2.5 温度采集模块电路设计	7
2.6 Wi-Fi 通信模块电路设计	8
2.7 蜂鸣器报警电路设计	10
2.8 按键控制电路设计	11
3 系统软件设计	12
3.1 数据采集子程序设计	12
3.1.1 心率采集子程序设计	12
3.1.2 温度采集子程序设计	14
3.1.3 姿势获取子程序设计	16
3.2 安卓手机 APP 设计	17
3.2.1 活动组件 activity	17
3.2.2 Android APP 连接接口	18
3.2.3 Android APP 功能控制	19
3.2.4 无线通信子程序设计	20
4 系统测试	22

4.1 心率曲线滤波测试	23
4.2 测试数据对比	23
4.3 系统测试结果分析	26
5 总结	28
参考文献	29
附录	31

基于 STM32 的运动监测系统的设计与实现

摘 要

本系统利用ST公司基于 ARM Cortex M3 内核的 STM32F103C8T6 单片机，作为核心的运动检测系统。监测系统主要有心率采集模块、温度采集模块、Wi-Fi 模块、姿势检测模块以及APP控制模块五个模块相互配合工作。通过STM32单片机控制光电心率传感器采集心率值，通过ADXL345传感器获取运动状态及步数，通过DS18B20温度传感器获取体温数据，还具备万年历功能。系统能够对心率及温度设置阈值，如果超过阈值蜂鸣器报警提醒，同时闪烁红色LED灯警示用户，且设置阈值存储到单片机Flash中，掉电不丢失数据。Android APP共享信息，时刻掌握健康数据，了解自身的健康情况。本系统采用了模块分离的设计思想，各个模块之间独立互不影响，让整体程序简化。

关键词：STM32；ADXL345；Wi-Fi 模块；Android 模块；心率

The Design and Implementation of Motion Monitoring System Based on STM32

Abstract

This system uses STM32F103C8T6 MCU based on ARM Cortex M3 core of ST company, as the core of the motion detection system. The monitoring system mainly includes heart rate acquisition module, temperature acquisition module, Wi-Fi module, posture detection module and APP control module, which work together. The photoelectric heart rate sensor is controlled by STM32 microcontroller to collect the heart rate value, the ADXL345 sensor to obtain the motion state and the number of steps, and the temperature data to obtain the body temperature through the DS18B20 temperature sensor. It also has the function of the calendar. It can set thresholds for heart rate and temperature. If the threshold value is exceeded, the buzzer will alarm and alert the user with red LED light flashing. And set the threshold value stored in the MCU flash, power down no loss of data. Android APP shares information, grasps the health data at all times, and understands its own health situation. This system adopts the design idea of module separation, each module is independent of each other, so that the whole procedure is simplified.

Key words: STM32; ADXL345; Wi-Fi module; Android module; Heart rate

1 绪论

1.1 系统设计背景

随着国内人们的平均寿命变长以及新生儿出生率降低，老年人的比重越来越大，各种职业病发率的不断攀升，各种高危工作伴随的危险性，外加疫情等特殊情况，身体健康是社会需要加大关注的一个问题。反应身体健康最直观的数据就是心率和体温^[1]。

心脏病和高血压是人们谈论的焦点和危害人体健康的主要疾病之一。心率的变化与心脏病密切相关。处理过程需要更精确地掌握心率数据。一般情况下，人体体温相对恒定，当体温超过正常体温的上限时，称为发烧。当人体出现感染时，身体的炎症就会引发体温升高，所以实时了解自身体温的变化，便可以对疾病的预防做到未雨绸缪。因此将心率监测和体温检测集成在一起有着非常广阔的发展空间。

1.2 系统设计意义

1593 年，意大利科学家伽利略（Galileo）发明了水银温度计，该温度计利用了水银的热膨胀和冷收缩原理，但受外部大气压力等环境因素的影响，导致测量误差较大。至今已经有 428 年了。医用体温计正从原来的玻璃体温计逐渐发展为多功能电子红外体温计，精度越来越高，但体积越来越大，携带不便。同时，国家食品药品监督管理局于 2020 年 10 月 16 日明确要求，从 2026 年 1 月 1 日起，我国全面禁止含汞温度计和血压计产品的生产。

现在市面常见的心率设备给人的第一印象就是体积硕大。家用的电子心率监测设备携带也不方便，而且操作不当很容易导致信息不准确。在二十一世纪高速发展的今天，没有人愿意出门带着两个机器。为了改善携带不方便、操作不简洁、信息获取不及时缺点。本文研究的基于 STM32 的运动监测系统^[2]，将体温心率和运动状态集成到一起，并且通过板载 Wi-Fi 将信息共享到手机 APP。将体温计和心率监测设备小型化穿戴在身上，不需要任何操作打开手机就可以查看相关信息。

与此同时，它可以计算步数，测量一个人的心率、血氧和其他参数，然后通过 Wi-Fi 与 APP 通讯，在 Android 手机上显示和记录参数信息来帮助人们，长时间监测运动和健康状况。运动监测设备将是在万物互联的时代，每个人不可或缺的工具，基于 STM32 的运动监测系统有着重要的研究价值和发展意义。

1.3 系统功能设计要求

STM32 单片机用作运动监控系统的主控制器，主要模块为心率采集模块、温度采集模块、Wi-Fi 模块、姿势检测模块和 APP 控制模块。通过 STM32 控制各个模块相互配合工作，通过 Wi-Fi 模块和手机 APP 互联，实现 APP 控制。系统需满足要求如下：

- (1) STM32 单片机实时采集心率传感器输出的模拟量进行模数转换；
- (2) STM32 单片机通过温度传感器检测温度，温度显示精度 0.1 度；
- (3) STM32 单片机实时采集传感器 ADXL345 重力加速度分量变换状态，进行计步的同时根据一段时间内状态变换次数，分析出体态：比如躺、站、走、跑等状态；
- (4) 无线 APP 功能，APP 能够连接 Wi-Fi 模块，查看相应数据；
- (5) APP 能够对心率阈值、温度阈值进行设置，能够进行计步步数清零操作。

2 系统硬件设计

随着物联网的应用领域越发的广泛，智能医疗作为物联网主流产物，大大的提高了人们的生活质量。与之相应的设计了可以检测心率和体温的运动监测系统。该系统的主控制模块 STM32 单片机作为核心来控制温度传感器、光电心率传感器、加速度传感器和 Wi-Fi 模块相互工作。实现监测佩戴者心率、体温和运动状态等信息，然后信息通过 APP 可以查看，还添加了报警电路。具体的系统结构图如图 2-1 所示。

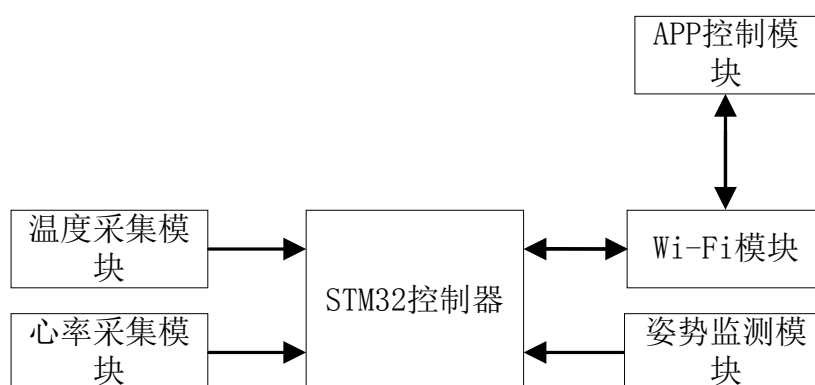


图 2-1 系统结构图

2.1 单片机核心电路设计

STM32F103C8T6 单片机作为本系统的核心控制芯片，电路整体相对复杂。主要通过各个 GPIO 串口控制各个传感器工作和获取数据^[3]。既能够保证最低成本、最低功耗，同时还可以提供非常丰富的接口以及功能。可支持重复擦除、下载、写入的 flash，方便各种场景下的系统实践。芯片独特的封装，有利于系统的维护、启动、外围电路的扩展。现在正是智能医疗中使用最多的芯片之一，价值不可估量。

数据总线主要采用全双工通信的同步串行总线，确保系统工作的稳定性和信息获取的准确性。单片机的 USART1 与 ESP8266 模块（Wi-Fi）串口通信，使用通用串行数据总线的串口和中断服务程序完成数据的接收。ADC 数模转换与光电心率传感器通信，通过串口控制信息回调实现数据的发送。

单片机其他 I/O 口与其他模块通信连接情况如下：

- (1) I/O 口 PC3 与蜂鸣器报警模块通信；
- (2) I/O 口 PA4-5 与 ADXL345 模块通信；
- (3) I/O 口 PA7 与温度监测电路通信；
- (4) I/O 口 PA8、PA11、PA2、PA5 是系统设置按键电路。通过按键手动设置

心率和体温的阈值。

STM32 单片机还集成了电源电路, IO 接口电路, 复位电路, 晶体振荡器电路, 串口下载电路和其他组件。 STM32F103C8T6 单片机接口电路如图 2-2 所示。

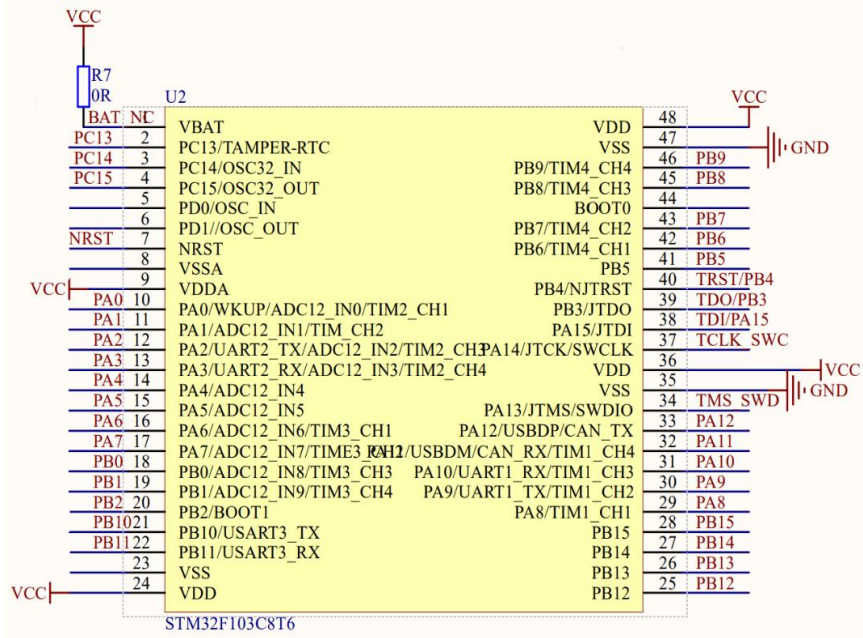


图 2-2 STM32F103C8T6 接口电路图

JTAG 接口作为嵌入式使用最多的下载调试接口,最基本的组成有 5 条线,TCK、TMS、TDI、TDO 和复位信号。它和专用设备（仿真器）协作与主机通信来交换数据。芯片上有一个专用的硬件部分与此接口配合工作,因此主机可以下载程序,控制芯片用于调试。读取程序将此端口的信息导出或存放到片上寄存器,关联的存储器和外围设备。本系统使用的 JTAG 接口电路如下图 2-3 所示。

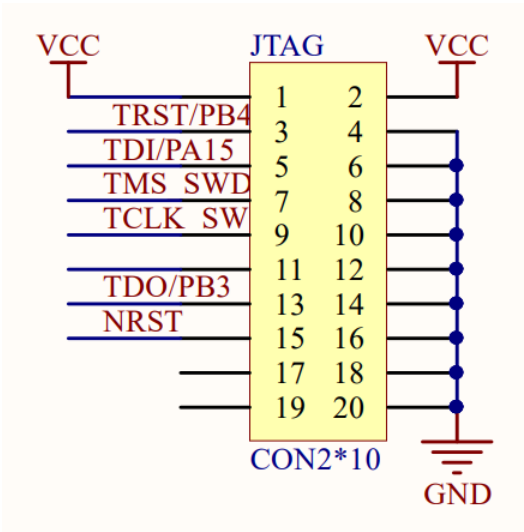


图 2-3 JTAG 下载接口电路图

2.2 心率采集模块电路设计

Pulsesensor 传感器是常用于心率检测，本质上是一种光学心率传感器，它具有放大器电路和噪声消除电路。传感器的核心有一个发光二极管作为发射端，光电晶体管作为设备的接收端。通过发光二极管发出的光通过人体的表皮组织会发生定期的变化，晶体管接收到变化的光信号，通过两个电路的作用转化为电信号。与此同时，在使用传感器的时候，最理想的测试位置是指尖和耳垂。模块实物如图 2-4 所示，相关参数配置特性表如表 2-1 所示。



图 2-4 脉搏心率传感器实物图

表 2-1 Pulsesensor 主要特性表

特性	说明
供电	4.8V-5.0V
引脚	外接 GND
输出	输出接口(0 和 1)
温度	-40- 85℃

2.3 LM393 滤波模块

LM393 是双电压比较器集成电路。模块有负载电阻保护，可以接入 2V~36V 的电源电压。而且没有高电压端电压值的束缚，输出时可以使用 SPS 接地开路，不受其他影响。但是在设备启动的都有反向电动势，输出端的灌电流受到电机或者其他电子设备影响，输出的电压可能在启动时瞬间达到峰值。电路参数配置特性表如表 2-2 所示。

表 2-2 LM393 比较器主要特性表

参数	说明
单电源	2~ 36V
ATSE	$\pm 1 \sim \pm 18V$
电流消耗	$I_{CC}=0.4mA$
失调输入电压	$V_{IO}=\pm 2mV$
共模输入电压	$V_{IC}=0 \sim V_{CC}-1.5V$
输出	TTL、DTL、MOS、CMOS 等兼容
备注	开路集电极连接“或”门

本系统中使用 LML393 通用双电压滤波模块，将脉搏传感器的输出电压接到 IN+引脚接口，两个输入接口但是心率曲线只有一个输出电压，所以 IN-引脚接地。通过 DO 小板数字量输出接口（0 和 1）。将输入电压与参考电压进行比较并实现开关输出，常用于各种类型的传感器，经过 AD 处理后的输出电压更加直观稳定。使用 LM393 滤波模块对脉搏和心率传感器模块的脉冲信号进行滤波，因为 LM393 集成了 ADC 数模转换模块可以直接连接到单片机引脚。用于滤波脉冲的电路原理图如图 2-5 所示。

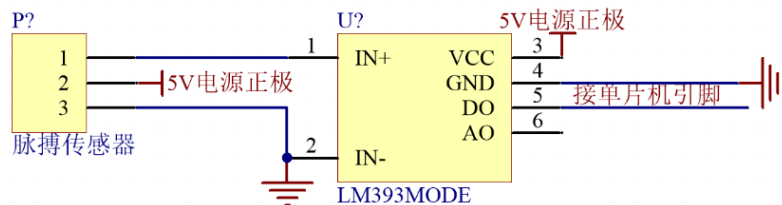


图 2-5 滤波电路图

2.4 运动状态检测模块电路设计

本系统中用来检测运动状态和统计步数的传感器是倾斜角传感器，使用的模块是 ADXL345，该模块具有体积小、集成度高和超低功耗的特点，电路配置参数特性表如表 2-3 所示。

表 2-3 ADXL345 主要特性表

参数	说明
分辨率	13bit
测量范围	16g
输出数据格式	16bit 二进制补码

设计中实时检测相关运动状态信息，选择倾斜传感器的 ADXL345 模块主要原因是：ADXL345 适合移动设备应用使用与开发。模组在倾斜检测感应中，不仅可以测量静态重力加速度，还可以测量在不同运动状态下以及非正常状态下的动态加速度。而且高分辨率的测量精度可以精确到正负 1 的倾斜角度变化。模块还具备独具一格的测试功能：

- （1）运动检测功能：可以自动或者手动检测 x 轴、y 轴、z 轴的加速度，并且与使用者设置的标准值进行对比，检测当前是否存在运动趋势；
- （2）敲击检测功能：可以监测来自所有方向的单和双震动；
- （3）自由落体检测功能：模块通过某一时间段突然的动态加速度变化，且变化的方向相对固定，可以检测设备是否坠落。

以上特色功能都可以独立出来，通过 GPIO 引脚中断控制，都互不影响。同时如果使用 32 级别的 First in First out 缓冲区来存储数据，可以减低控制主处理器芯片的整体功耗，最大程度的做到低功耗。ADXL345 模块的电路图如图 2-6 所示。

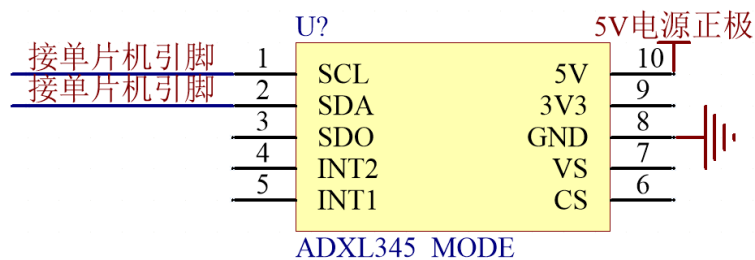


图 2-6 ADXL345 模块电路图

2.5 温度采集模块电路设计

温度检测主要使用具有体积小、成本低廉、抗干扰能力强和精度高等特点的 DS18B20 数字温度传感器。只有一根数据引脚线，非常方便的接线原理，包装以后的传感器可以通用于各种情况。常见的包装方式有管道型、螺纹型、磁体吸附型、不锈钢包装型、LTM8877、LTM8874 等。

传感器的通用性极高，可以根据不同的应用环境更改传感器的外形。其耐热性很高，常用于测量电缆沟温度，测量循环炉水的温度，学校食堂或者工厂的锅炉温度也是使用 DS18B20 进行测量。室温、体温、水温都可以使用该传感器进行测量，可以取代生活中的体温计、温控计等。弹药仓库温度测量和其他非限制性温度事件。耐磨、防撞、体积小、使用方便，适用于各种包装形式，各种狭窄空间设备的数字温度测量和控制。DS18B20 技术参数：

(1) DS18B20 接口模式为唯一的单线接口，与微处理器连接时，只需一根端口线即可实现核心处理器与 DS18B20 之间的控制和信息传输；

(2) 支持的测量范围为-55 度到+125 度，温度测量误差（传感器自身物理误差）为 1；

(3) 可以多点联网功能。多个 DS18B20 可以并联连接到 3 根导线，提示并联的最多导线数量为 8，如果特殊情况下需要超过 8 个，需要给恒定的电压，并联过多导致电源电压过低可能会信号传输丢失；

(4) 正常的工作电源: 3.0~5.5V/DC （支持数据线寄生电源）；

(5) 不需要任何外围元件辅助使用；

(6) 测量结果传输方式以 9bit 到 12bit 二进制数字信号进行串行传输。

DS18B20 温度传感器具有 9 到 12 位的可编程分辨率，需要耗时 750 毫秒才能转换为 12bit 数字格式的最大温度，应用场景是用户自定义的温度报警设置，而且是非易失性的。主要应用在恒温器控制，消费电子产品温度计和工业级的热敏系统。测试证明 DS18B20 温度传感器符合本设计要求。

通常，有两种类型的 DS18B20 温度传感器封装，它们均被同等使用。可以根据使用环境自由选择。第一个是主要用于空气温度感应的直接暴露的芯片。第二种通常封装在防水不锈钢管中，用于在水温灯中进行液温感测。

DS18B20 温度传感器的示意图如图 2-7 所示。为了实现 DS18B20 传感器的数据获取的准确性通常会在电路中加一个 10k 的上拉电阻，既可以保护电路也可以保证数据的准确性。

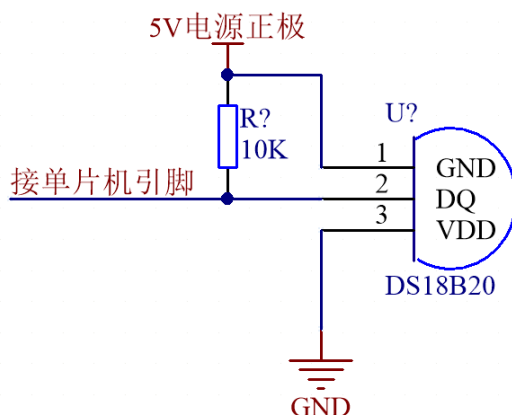


图 2-7 DS18B20 温度传感器原理图

2.6 Wi-Fi 通信模块电路设计

ESP8266 模块是一款超低功耗，高效率的 USART 同步透明传输模块^[4~5]。通

用的包装大小和低功耗的集成技术，多出现在移动终端或者物联网应用中，智能家居、智能医疗、智能农业都有它的身影。可以执行 Internet 或者 LAN 通信实现联网操作，具有 Wi-Fi 杀手的称号。

模块有多种封装方式，天线可以支持三种板载 PCB 天线，即 IPEX 接口、邮票孔接口和外接接口。

工作温度范围大，且能够保持稳定的性能，能适应各种操作环境。可广泛应用于智能电网、交通、家居和工业控制等领域。

无线保真特点：

- (1) 采用无线 802.11 b/g/n 标准；
- (2) 三种工作模式选择：STA / AP / STA+AP；
- (3) 包含 TCP/IP 协议，同时连接多个 TCP 客户端；
- (4) Socket AT 指令；
- (5) 使用 USART/GPIO 数据通信接口,传输数据不掉帧；
- (6) Smart Link 自动联网；
- (7) 可以远程升级固件（OTA）；
- (8) 具有独立的 MCU（32 位），可作为应用处理器；
- (9) 电池供电；
- (10) 单电源供电 3.3V；

可以实现的主要功能包括串口透明传输，PWM 控制和 GPIO 控制：

- (1) 串口透明传输：数据传输，传输可靠性好，最大传输速率为 460800bps；
- (2) PWM 控制：照明控制，3 色 LED 控制，电机速度控制等；
- (3) GPIO 控制：控制开关，继电器等。

ESP8266 模块常见的三种工作模式：作为服务器，当作客户端，服务器+客户端：

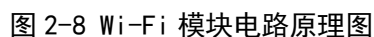
(1) STA 模式：作为客户端连接到局域网，通过 APP 或者 PC 对设备进行远程控制；

(2) AP 模式：ESP8266 模块作为服务器，设置热点，用户通过终端连接热点直接实现无线控制；

(3) STA + AP 模式：可以通过 Internet 远距离控制，操作方式多样。

应用领域：

- Wi-Fi 模块电路图如下图 2-8 所示。



有源蜂鸣器是电子蜂鸣器的集成结构，利用直流电压供电，蜂鸣器是生活中最常见的电子器件。电饭锅开关跳开以后会“滴”一下，烟雾报警器检测到有烟时会长鸣，倒车雷达除了语音提醒还会发出蜂鸣警报等。这些都是蜂鸣器的应用。在本系统中使用 5V 直流电源供电的有源蜂鸣器传感器，传感器电路使用 9012 晶体管作为驱动。当引脚为高电平时，蜂鸣器会发出警报，当引脚电压为 0，则蜂鸣器不会发出声音警报。单芯片引脚的方波输出形状可以控制蜂鸣器的工作模式（蜂鸣模式）。电阻使用是一个限流电阻，蜂鸣器保护报警电路原理图如图 2-9 所示。

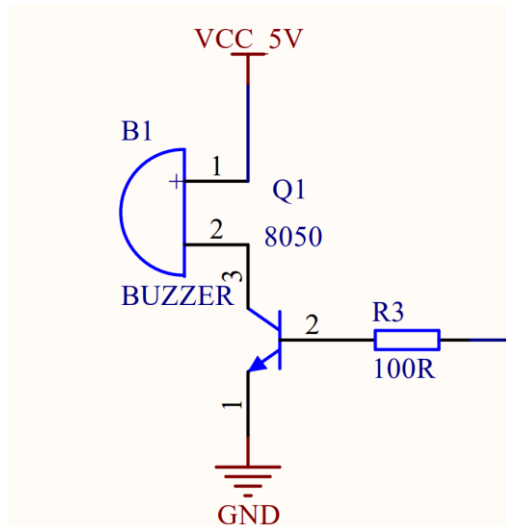


图 2-9 蜂鸣器报警保护电路原理图

2.8 按键控制电路设计

系统中使用最简单的四角按键作为按键电路开关，这种开关不能够自锁，但是每当按下去一次，信号就发生一次跳变。未按下时只有两组引脚连接，按下之后，四个引脚连接在一起。然后实现对系统的手动输入。使用了四个按键来控制系统操作。S4 按键与引脚 PA5 连接，实现系统功能选择，设置对应的阈值，选定以后可以通过与引脚 PA11 相连的 S2 按键增加数值，S1 按键减小数值，S3 按键返回操作，具体电路原理图如下图 2-10 所示。

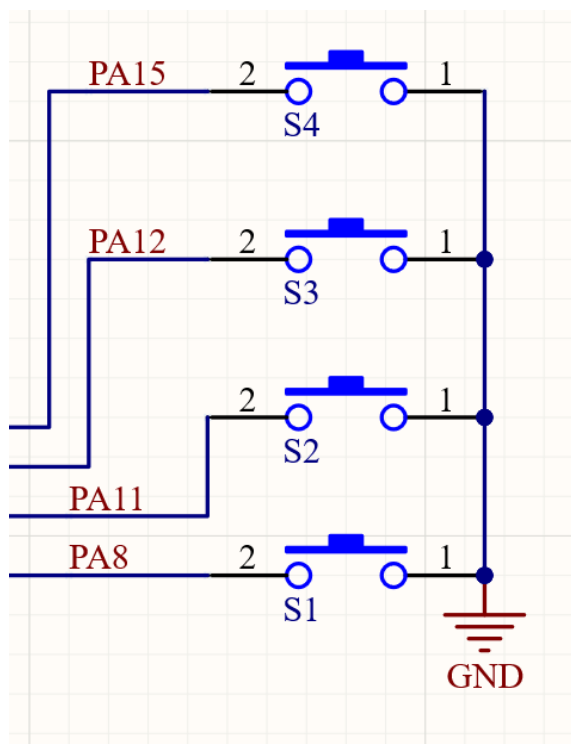


图 2-10 按键电路原理图

3 系统软件设计

系统软件设计主要分为两个部分：**Android APP 设计**和**硬件驱动代码设计**。系统通过软硬结合，驱动代码使用库函数编写，通过调用相关函数进行底层工作。主函数调用各个模块的子函数进行相互配合工作^[6~8]。函数分模块化的好处是简化程序设计，在可读性和修改容易度上面有着明显的益处，主函数流程图下图 3-1 所示。

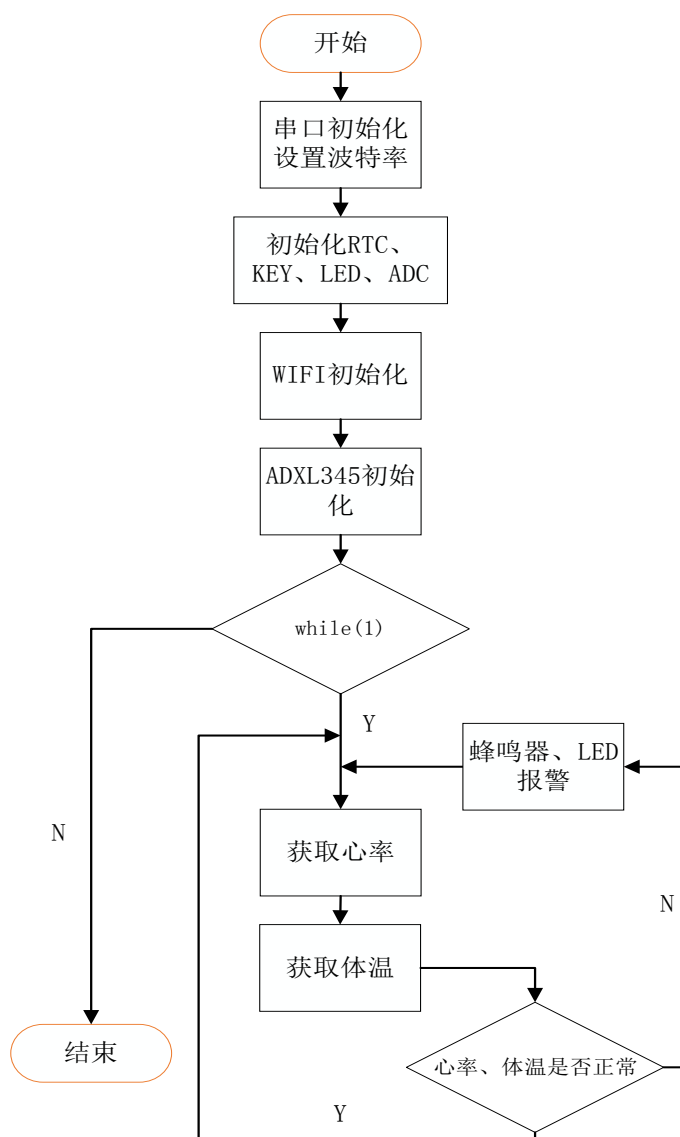


图 3-1 主函数流程图

3.1 数据采集子程序设计

数据由 **Pulsesensor 脉搏心率传感器**，**DS18B20 温度传感器**，**ADXL345 倾角传感器**收集，从外界获取数据。

3.1.1 心率采集子程序设计

Pulsesensor 脉冲传感器对光电信号进行滤波和放大，最后输出一个模拟电压

值^[9]。单片机可以将采集到的模拟信号值转换为数字信号，然后通过简单的计算即可获得心率值。只需要控制信号引脚就可以工作。首先初始化心率值为零，调用滤波获取函数，进入滤波获取函数首先调用中断延时函数，做到滤波防止抖动 300ms，也就获取滤波周期大于 300ms 才能进入心率检测，滤波滤掉手指未放情况就可以进行滤波处理，在整个子函数中除了心率计算函数就是滤波函数滤掉手指未放的情况，函数的实现主要调用中断延时，每次的滤波回波少于 2000ms 的时候就表示没有物体去监测，这个情况下心率值为零，延时清空。在符合监测情况下，滤波处理每次 60 秒获取四段滤波求平均值，心率值=平均滤波*5*5ms^[10]。如图 3-2 所示。具体实现代码如下：

if(heartRateDelay>(300/My_SysTick_GetPeriod()))//滤波防止抖动 300ms，timerPeriod 为定时器周期，在 my_timer.c 中

```
{
    if(heartRateDelay>(2000/My_SysTick_GetPeriod()))//滤波滤掉手指未放情况
    {
        heartRateDelay = 0;//此时心率为 0
    }
    else
    {
        heartRateTemp[count++]=heartRateDelay;//记录前后两次事件间隔
        heartRateDelay=0;
        if(count>=4)//记录超过 4 次进行滤波处理
        {
            heartRate
            =60000/((heartRateTemp[0]+heartRateTemp[1]+heartRateTemp[2]+heartRateTemp[3])/4*My_SysT
            ick_GetPeriod());//60000 单位 ms 表示 60s /4 求平均 *5 每次定时 5ms
            count=0;//清除本次记录
        }
        heartRateDelay =0;//延时清零
    }
}
```

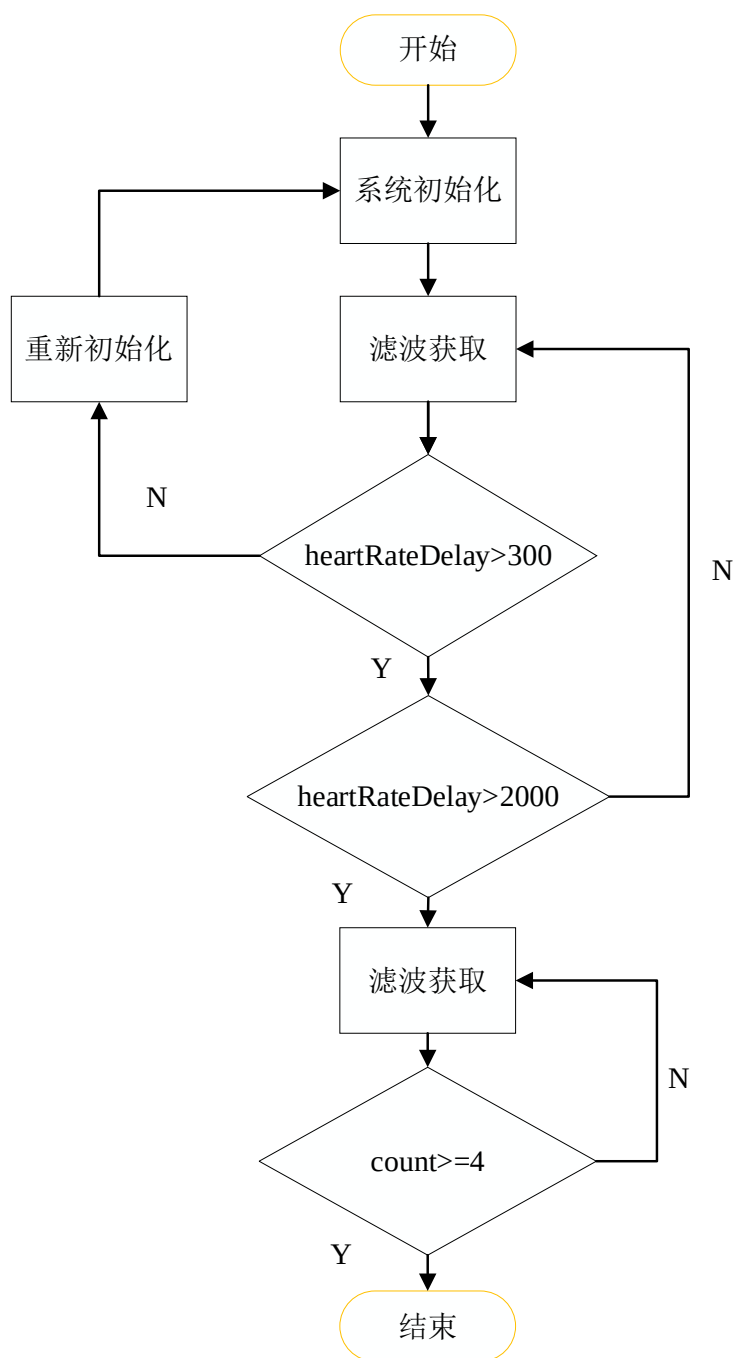


图 3-2 心率采集流程图

3.1.2 温度采集子程序设计

DS18B20 温度传感器同样只有一个 DQ 引脚作为单片机控制的接口和数据输出的接口，温度采集主要是串口初始化，然后复位 DS18B20，复位是为了等待 DS18B20 响应，相当于一个延时给温度获取提供时间，写一个字节命令到 DS18B20 设置对应的传感器工作模式，获取温度并将最终的温度转换成数字模式，通过串行端口发送到 MCU。具体流程图如图 3-3 所示。

温度获取代码如下：

```

float My_DS18B20_GetTemp(MyPinDef pin)
{
    Temp_L=My_DS18B20_Read_Byte(pin); // LSB
    Temp_H=My_DS18B20_Read_Byte(pin); // MSB
    if(Temp_H>7)
    {
        Temp_H=~ Temp_H;Temp_L =~ Temp_L;
        temp=0; // Temperature is negative
    }else temp = 1; // Temperature is positive
    Tem = Temp_H; // Get the high eight
    tem <<= 8;
    tem += Temp_L;// Get the low eight
    tem = (float)tem * 0.625;//exchange
    if(temp) return (float)tem / 10; //return Temperature
    else return -(float)tem/10;
}

```

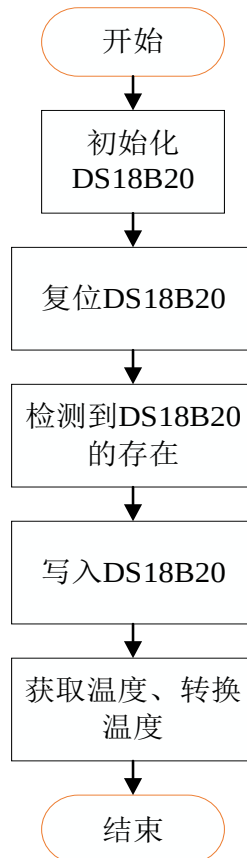


图 3-3 温度采集流程图

3.1.3 姿势获取子程序设计

本系统中的步数和姿势都是通过 ADXL345 数字倾角传感器来获取,ADXL345 数字倾角传感器主要通过 IIC 总线通信,首先通过寄存器初始化对应的设置,读取到复位值 0xE5 的话,说明器件连接,以便进行以下操作;低电平中断输出,13 位全分辨率,右对齐输出数据,16g 范围,数据输出速率为 100Hz,链接使能和测量模式,不使用中断。然后读取 X、Y、Z 三轴的数据。

读取 ADXL345 的数据 100 次,再取平均值。每一次数据通过 IIC 总线发送完后自动校准 X、Y、Z 轴的角度。根据角度 X、Y、Z 方向的重力加速度分量分析检测当前运动状态的次数,具体流程图如下图 3-4 所示。

设置 ADXL345 传感器工作模式,寄存器配置代码如下:

```
bool My_ADXL345_Init(void)
{
    if(IIC_ReadOneByte(Address_ADXL345, DEVICE_ID)==0XE5)
    {
        IIC_WriteOneByte(Address_ADXL345,DATA_FORMAT,0X2B);/
        IIC_WriteOneByte(Address_ADXL345,BW_RATE,0x0A);
        IIC_WriteOneByte(Address_ADXL345,POWER_CTL,0x28);
        IIC_WriteOneByte(Address_ADXL345,INT_ENABLE,0x00);
        IIC_WriteOneByte(Address_ADXL 345, OFSX,0x00);
        IIC_WriteOneByte(Address_ADXL 345, OFSY ,0x00);
        IIC_Write One Byte(Address_ADXL 345,OFSZ,0x00);

        return true;
    }
    return false;
}
```

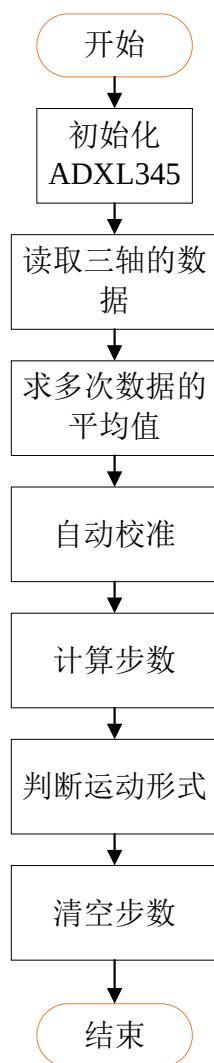


图 3-4 步数数据获取程序流程图

3.2 安卓手机 APP 设计

本设计中的 Android APP 主要使用四大组件，APP 可以连接硬件的 Wi-Fi，查看对应的 RTC 日期、RTC 时间，查看实时的心率值、温度值、步数。同时可以清空步数，查看对应的运动状态（躺、走、跑），设置对应的心率和温度阈值，可以自动校准时间^[11]。APP 和硬件系统的通信依靠 TCP 协议和 socket 接口，TCP 的确认应答机制和超时重发可以保证软硬件在通信的过程中数据的准确性、安全性和可靠性。与此同时同时支持 UDP 协议，UDP 协议作为一个备选方案应用于本文设计中，利用 UDP 存在接收缓冲区的特点，在 Wi-Fi 较差的时刻也能保证 APP 有数据。

3.2.1 活动组件 activity

活动是四个最基本且最常用的 Android 组件之一。Android 的四个主要组件是“活动”、“服务”、“内容提供者”和“广播接收器”。活动是 Android 最重要

的组成部分。在本系统中是编写接口时使用的主要方法。“服务”负责用户界面设计。内容取自“内容提供者”的布局文件。“广播接收器”是一个页面管理器对象，并且用户交互的容器和显示页面必须都与活动对象相对应。当页面跳转时，将调用 Activity Finish()方法来终止正在进行的接口，调用 Activity Manager()方法激活另一个接口（下一个活动），实现页面的跳转。Android 具有三个核心组件：服务，广播接收器和内容提供者，它们之间的关系如图 3-5 所示。

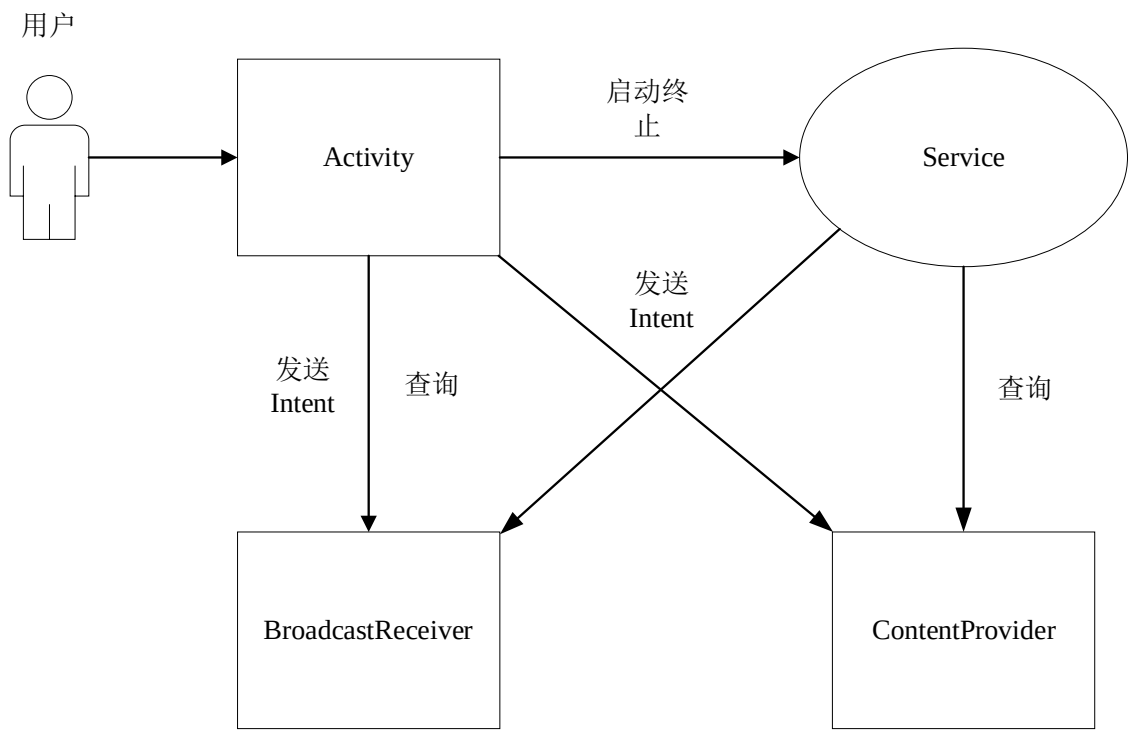


图 3-5 Android 组件及通信机制

3.2.2 Android APP 连接接口

在正常情况下，硬件 Wi-Fi 的 IP 地址和端口号是固定的。当 Android 手机连接到板载 Wi-Fi 硬件时，connect()函数会将文件描述符 sockfd 引用的活动套接字连接到该地址。如果 addr 和指定的 addrlen 连接成功，则 APP 立即从硬件向客户端请求数据并显示数据。流程图如图 3-6 所示。

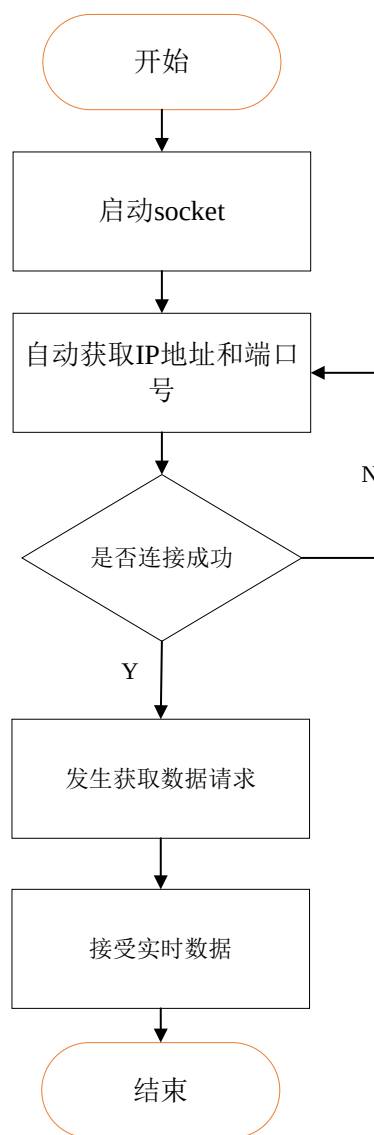


图 3-6 Wi-Fi 连接流程图

3.2.3 Android APP 功能控制

APP 连接 Wi-Fi 以后，就会显示心率值和温度值以及当前的运动状态，板子在第一次使用的时候，RTC 日期和时间会自动刷新，获取的是当前 Android 手机的日期和时间，现在的手机日期和时间都是联网校准的，所以时间为最新时间；在运动的状态下，会显示对应的运动状态，在走和跑的运动状态下，可以查看实时的步数，同时可以清空步数，可以设置对应的心率阈值和温度阈值，阈值可以自定义，但是在设置的过程中，需要在正常连接的情况下才能输入，在 APP 界面最下面有两个手动的 activity 交互按钮，用于在自动刷新日期和时间失败的时候，手动校准时间，更加准确的显示当时的健康数据。具体实现流程如下图 3-7 所示。

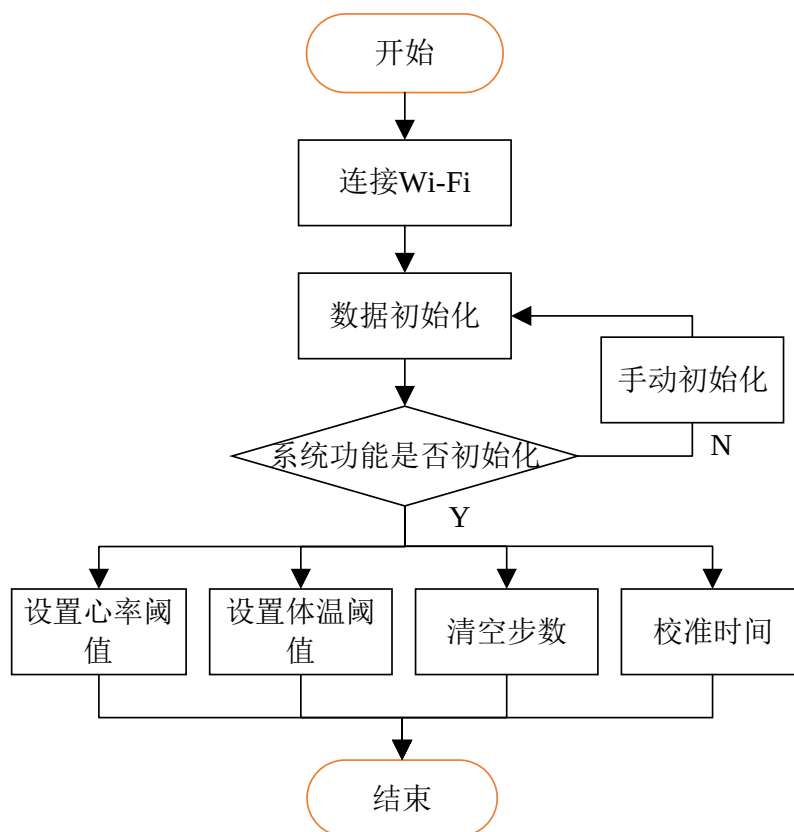


图 3-7 控制模块流程图

3.2.4 无线通信子程序设计

无线通信由 Wi-Fi 模块来完成，Wi-Fi 模块主要采用 ESP8266 模组，ESP8266 只使用两个引脚控制也就是支持 USART 协议。时钟信号可有可无，有一个奇偶校验位，只要双方设置后，就可以改变数据包的结构。首先是 Wi-Fi 热点的初始化，IO 接口以及各个通信总线的初始化，最后设置波特率以及相关数据包的配置。

模块需要充当多重身份，Wi-Fi 模块初始化后。AP 模式作为 server，设置 AP 的参数，SSID 和 Password。同时 Wi-Fi 模块初始化，作为客户端设置 Wi-Fi 模块使用的串口、要连接的 Wi-Fi 无线信号名称、要连接的 Wi-Fi 无线密码、要连接的 IP 地址、要连接的 TCP 端口号、连接超时时间等参数配置^[12~13]。

最后还需要发送数据给所有用户函数、发送数据给指定用户函数。Wi-Fi 接收状态机，在串口中断中调用，并处理接收到的数据。在主循环中调用，关闭和打开 TCP 或 UDP 服务函数等^[14]，无线通信子程序流程如图 3-8 所示。

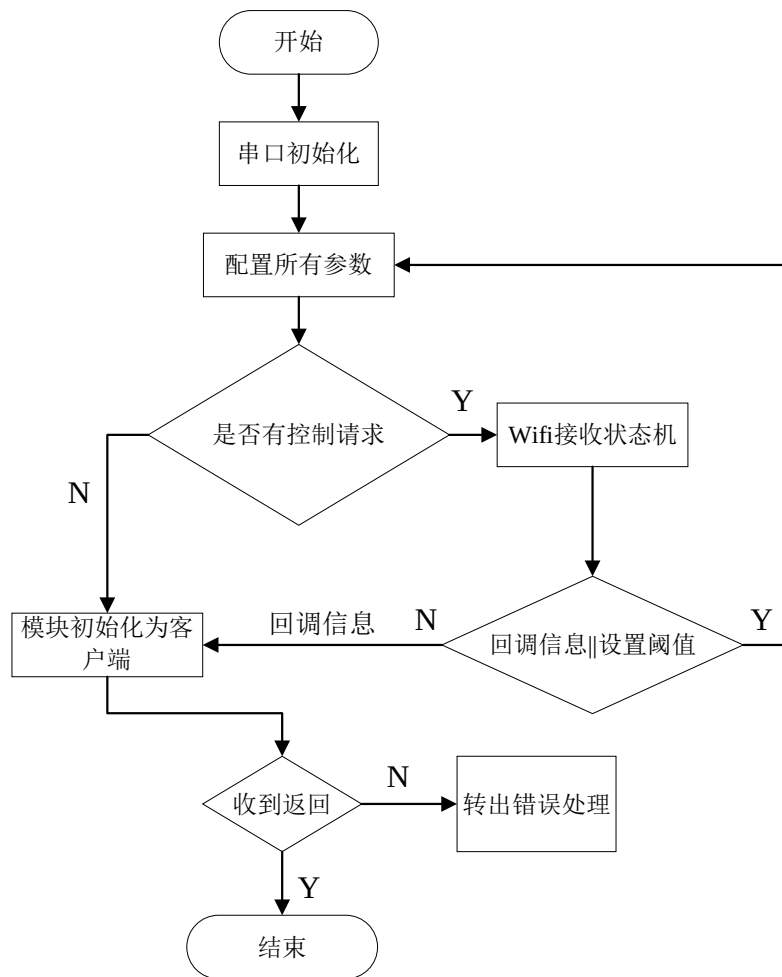


图 3-8 无线通信子程序流程图

ESP8266 模块所有函数如下图 3-9 所示。

```

- My_ESP8266_SaveToBuffer(u8 dat)
- My_ESP8266_WaitFlag(u16 flag, u16 t)
- READ_BIT(EspFlag, flag)
- Esp8266_Init_StationAP(const char *ssid, const char *pwd)
- Esp8266_Init_ConnectTCP(const char *ssid, const char *pwd, const char *ipAddr, u16 port, u16 timeout)
- My_ESP8266_SendData(u8 id, const u8 *dat, u16 length)
- READ_BIT(EspFlag, FLAG_ESP_MULTI_LINK)
- My_ESP8266_SendDataToAll(const u8 *dat, u16 length)
- My_ESP8266_SendStr(u8 id, const char *str)
- My_ESP8266_SendStrToAll(const char *str)
- My_ESP8266_StateMachine(u8 msgByte)
- My_ESP8266_Process(void)
- OnGetEspMessage(const esp_msg_obj *espMsgRec)
- My_ESP8266_SetEspCWSAP(const char *ssid, const char *pwd, bool def)
- My_ESP8266_SetUart_CUR(u32 baud)
- My_ESP8266_SetMultiLink(bool multi)
- My_ESP8266_StartSever(u16 port)
- My_ESP8266_StopSever(void)
- My_ESP8266_SendCmd(const char *cmd)
- My_ESP8266_SetMode(esp_mode mode, bool def)
- My_ESP8266_ConnectAP(const char *ssid, const char *pwd, u16 timeout_ms, bool def)
- My_ESP8266_DisconnectAP(void)
- My_ESP8266_SetIP_AP(const char *ip, bool def)
- My_ESP8266_SetIP_STA(const char *ip, bool def)
- My_ESP8266_ConnectSocket(const char *type, const char *remotelp, u16 remotePort, u16 localPort, u16 timeout)
- My_ESP8266_CloseClient(u8 id)
- My_ESP8266_SetTransMode(bool direct)
- My_ESP8266_SetTransLink(bool enable, const char *ipAddr, u16 port, const char *type)
- My_ESP8266_SetServerTimeOut(u16 t)
- My_ESP8266_SeverIsConnected(void)
- My_ESP8266_ClientIsConnected(u8 clientNum)
  
```

图 3-9 Wi-Fi 函数列图

4 系统测试

本系统组成主要是 STM32 单片机、Pulsesensor 脉搏光电心率传感器、LM393 电压比较器、ADXL345 倾斜角传感器、DS18B20 温度传感器、ESP8266、LED、蜂鸣器和按键。单片机控制各个模块相互工作。硬件使用一层贴片制版作为电路连接，汲取多次烧坏电路板的经验，历时四个月完成该系统的整体设计。

硬件设计正反面如图 4-1 和 4-2 所示。

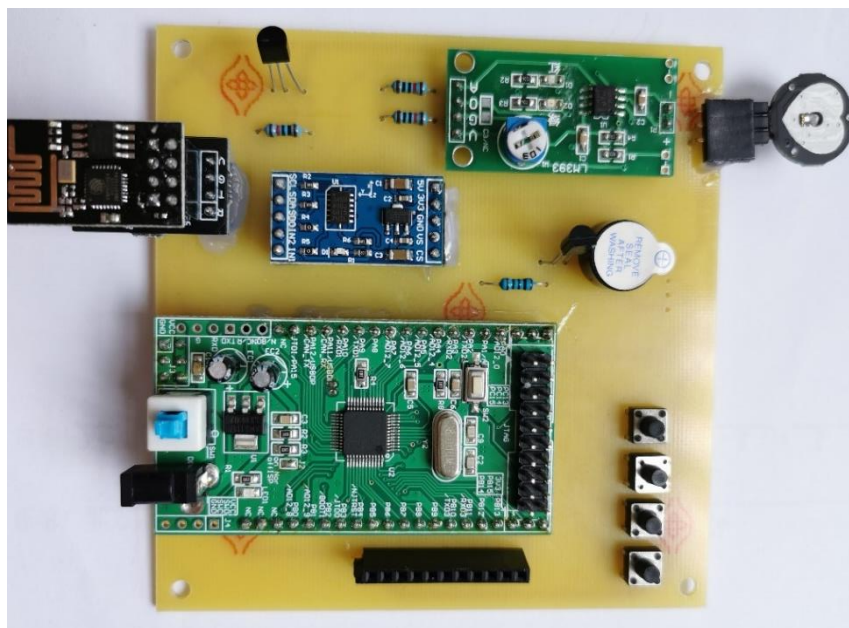


图 4-1 硬件实物正面图

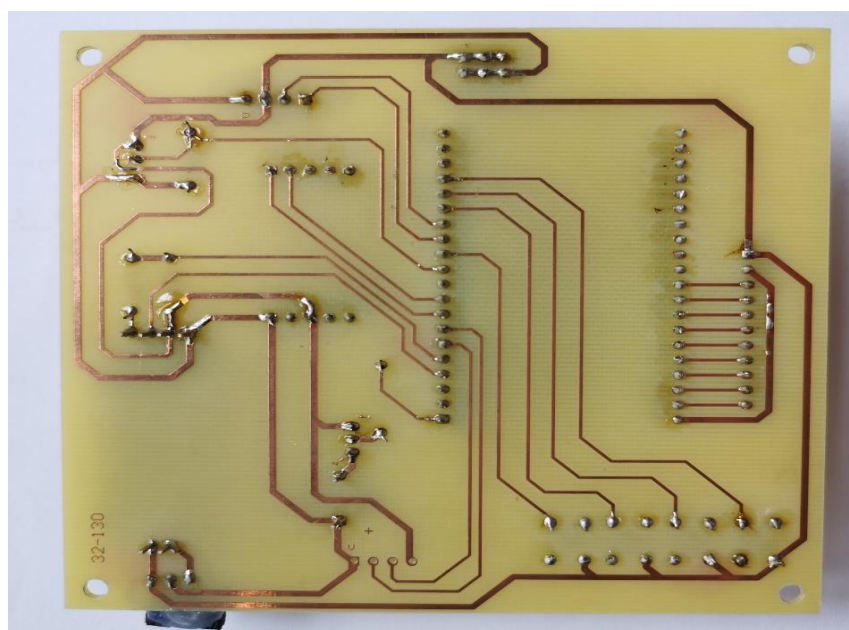


图 4-2 硬件实物反面图

4.1 心率曲线滤波测试

首先受用示波器对硬件进行测试，主要测试的是心率曲线，通过对 PA5 引脚的电压测试，PA5 引脚的电压曲线就是心率变化曲线，测得波形如下图 4-3 所示。

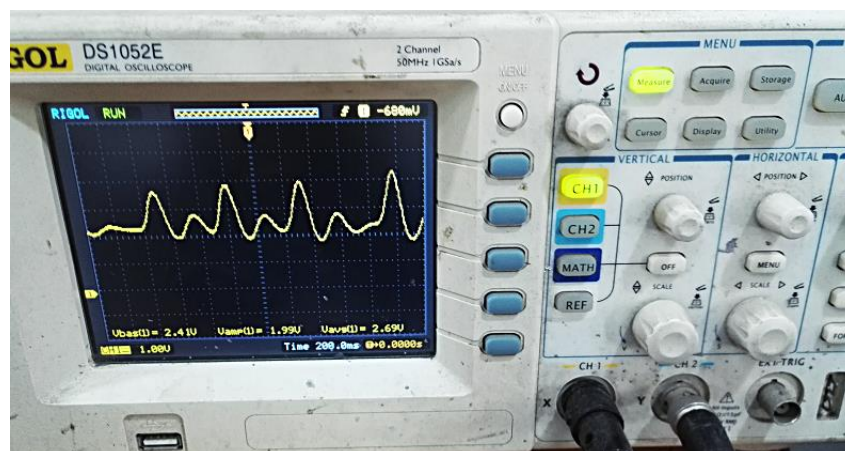


图 4-3 PA5 引脚电压实际输出信号波形图

根据图 4-3 可以看出心率波形图为不规则的波形图，使用 LM393 比较器对波形进行过滤，对信号进行处理可以提高数据的准确性，将曲线滤波成方波。然后输入信号为高电平 1 和低电平 0，确保处理器获取的信号不会因为电压不稳发生失真。LM393 比较器滤波后的波形如下图 4-4 所示。

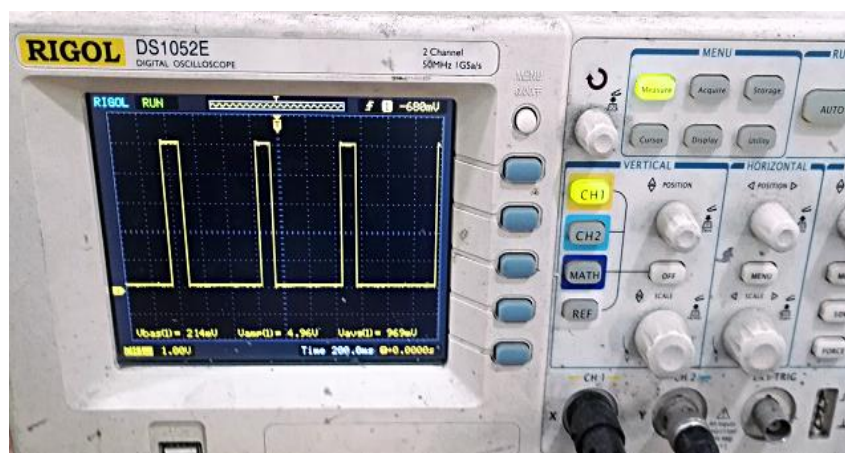


图 4-4 心率滤波后波形图

4.2 测试数据对比

使用手机 APP 通过 Wi-Fi 连接设备，查看相应的日期、时间、心率、体温、步数值及运动状态，同时通过 APP 设置阈值。查看得知当时的人体心率为 63 次/min，测得体温为 36.2 摄氏度，当前的运动状态为“躺”，APP 测试如图 4-5 所示。

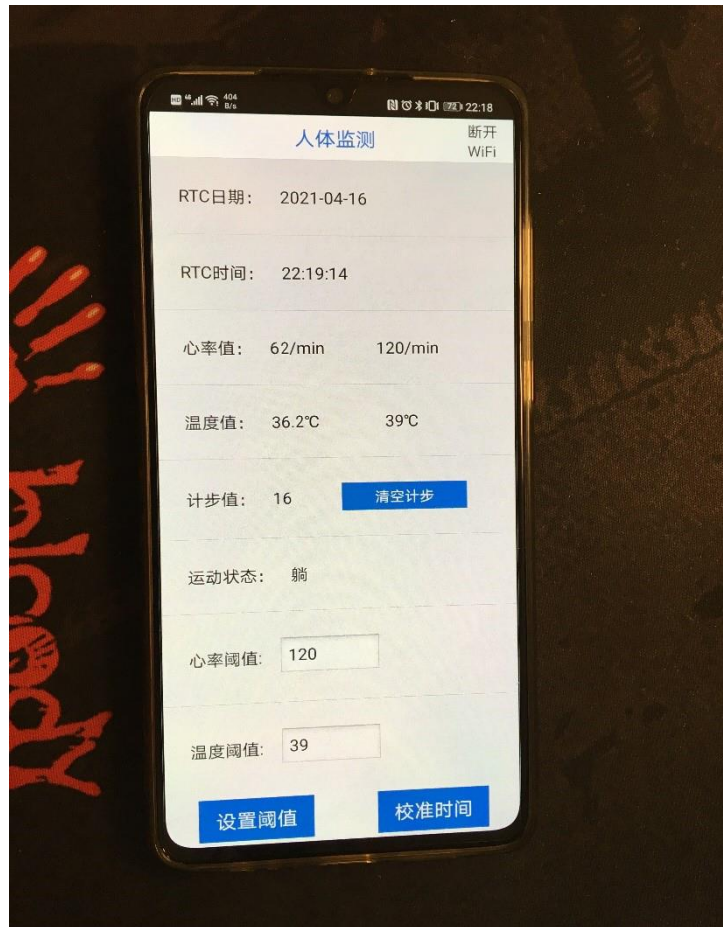


图 4-5 心率检测模块 APP 测试图

心率检测模块测试图如下图 4-6 所示。

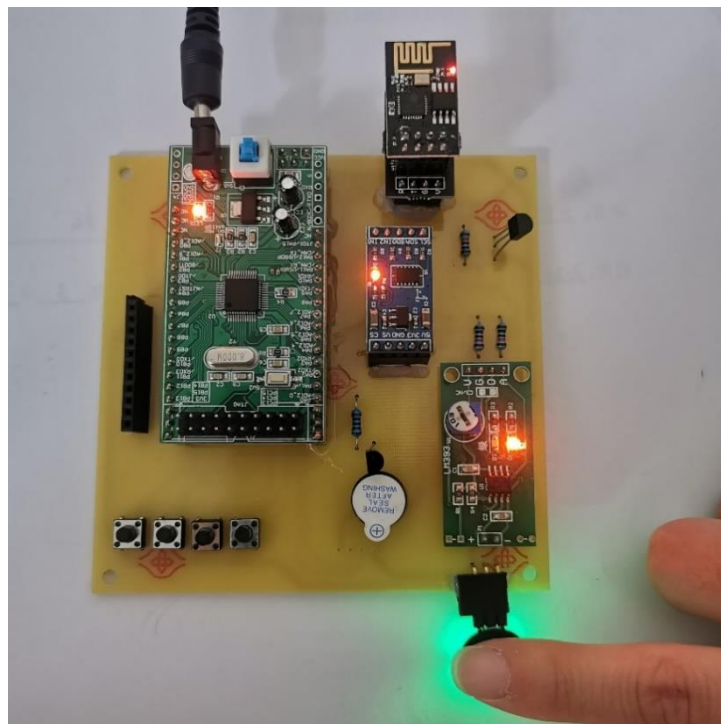


图 4-6 心率检测模块测试图

与此同时为了验证系统测试数据的准确性，还使用红外体温计和电子血压仪对人体体温和心率进行测量，测得数据与系统显示体温(36.2℃)和心率(62 次/min)一样，实验结果表明在稳定的环境下，本系统可以提供实时、准确的心率和体温以及步数，红外体温计和电子血压仪测得数据如图 4-7、4-8 所示。



图 4-7 红外体温计测温图



图 4-8 电子血压计测量图

4.3 系统测试结果分析

在系统通电的情况下，打开 SMT32 核心板的电源开关，让整个系统初始化，然后用 Android 手机连接名为 AUAISOUT000 的 Wi-Fi，打开人体检测 APP 进行测试，主要的测试内容与本系统的实现目标如下表 4-1 所示。

表 4-1 系统功能测试用例表

测试的模块	测试目的	实际情况
Wi-Fi 连接模块	通过手机可以搜索到 Wi-Fi（网络名：AUAISOUT000）	与期望的结果情况一致
心率检测模块	在一般环境下可以监测体温数据	由于光电原理传感器对不同环境的光照强度很敏感，在使用的过程中尽可能在偏暗的环境下测量或者带上屏蔽带
体温监测模块	在接触到人体时显示体温数据	与期望的结果情况一致
姿势监测模块	获取运动状态（躺、走路、跑步），同时在走路和跑步状态下获取步数信息	与期望的结果情况一致
按键测试	通过按键手动设置阈值	与期望的结果情况一致
LED 闪烁报警	温度、心率超过阈值闪烁报警	与期望的结果情况一致
蜂鸣器长鸣报警	心率、温都达到阈值长鸣报警	与期望的结果情况一致
压力测试	反复开关机，暴力开关机（拔电源）100 次（实际使用中，请勿暴力开关机）	实测 120 次，运行情况正常
量产测试	通过大量的数据采集，求整个系统的信息的准确性（测心率和体温数据以及在走路和跑步状态下的步数各 200 次）	心率误差值为+5 次/min；体温精度为 0.1 度，在正常使用情况下 0.1 误差；步数误差为 +7.5

由测试得知，在系统测试中，心率传感器对光的灵敏度特别强，需要特别注意在使用过程中环境光照强度变化，可以使用黑色的屏蔽胶带将检测模块绑在耳垂或者手指，同时在测量中不要剧烈运动，佩戴方式如图 4-9 所示。

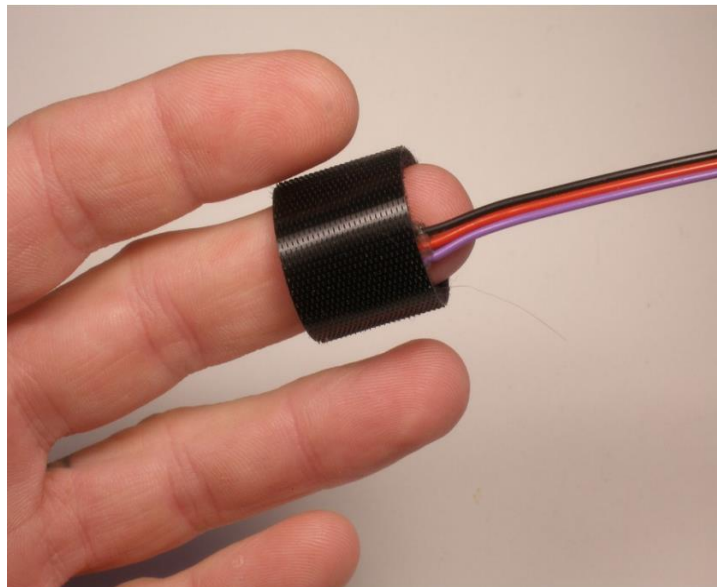


图 4-9 心率检测佩戴方式图

5 总结

本系统采用 STM32F103C8T6 核心板单片机、无线保真模块、Pulsesensor 脉搏心率传感器、ADXL345 传感器、DS18B20 温度传感器、LM393 比较器滤波电路、蜂鸣器报警电路、按键电路组成。上电自动初始化，开始工作，且设置阈值存储到单片机 Flash 中，掉电不丢失数据。系统主要使用 C 编写硬件程序和底层驱动程序，Java 和 Android 用于编写 APP，用了简单的 Makefile 作为工程管理^[15]；使用到的开发工具有 Keil uVision5、eclipse、JDK1.8、Android studio。经过为期四个月的设计和调试，实现了实时运动数据检测的功能。

在设计实现的过程中难免遇见各种问题，第一次使用 Altium Designer 设计原理图和 PCB 电路板，从做硬件的选配到最后的焊接，一边学习一边实践。在设计中，通过网络调试助手、模拟调试器、Android 模拟器一步一步的模拟实现，将函数模块化去验证。在此过程中认识到要学的东西还有很多。

设计中还要改善的问题：

（1）心率采集模块硬件设计上的稳定性和程序上的完整性，采集信息时需要非常细微的动作才能保证检测的数据的准确性，需要完善来适应多环境应用；

（2）数据可以通过 Wi-Fi 直接上传到云服务中，避免了每次使用都需要用手机连接特定的网络；

（3）APP 存在闪退的情况，不影响正常的使用，代码可能隐含 bug 需要进一步学习，进行解 bug；

本次设计来给我很多收获，让我对一个产品从底层硬件到上层的接口和应用都有了初步的了解。人生就是不断学习的过程。

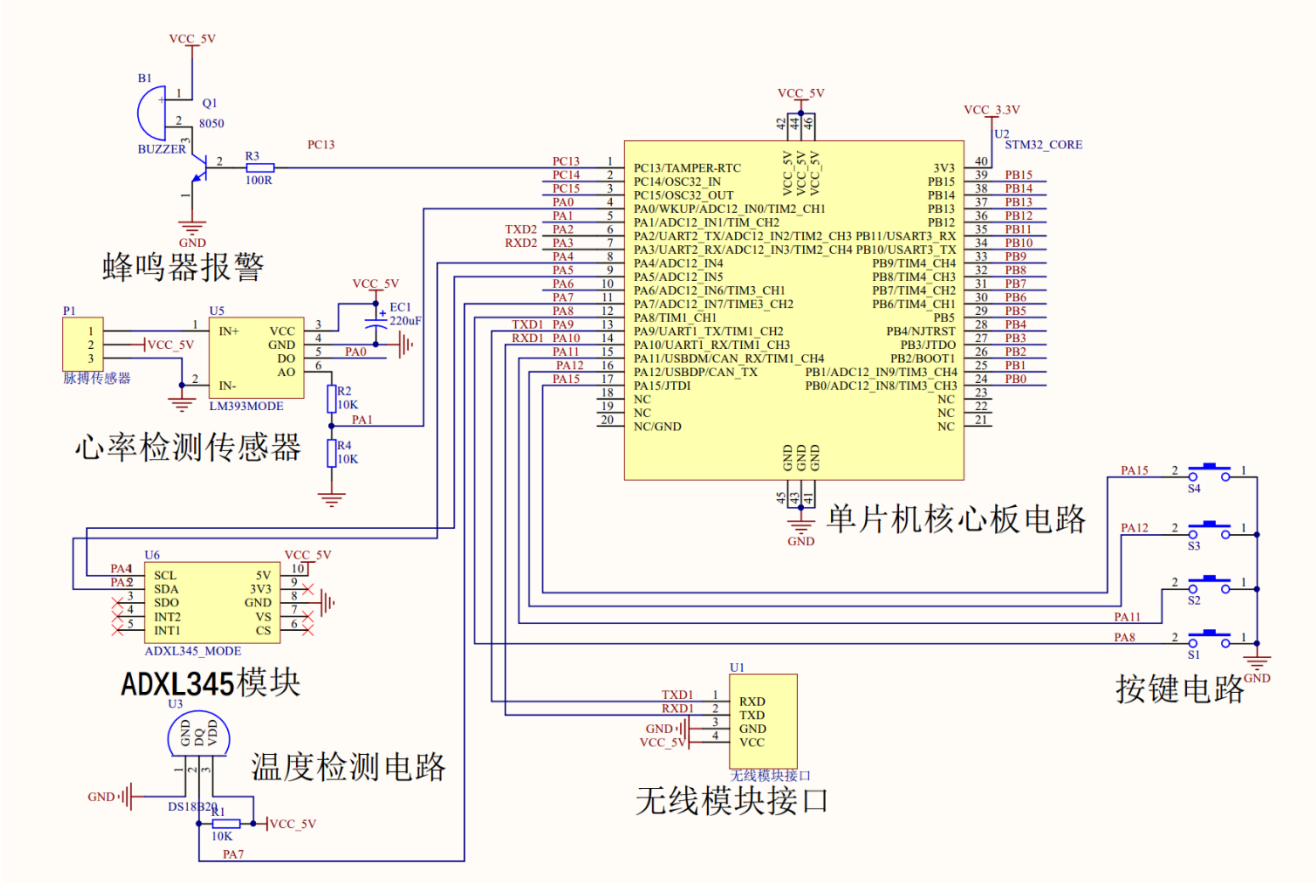
参考文献

- [1] 王忠远, 薛慧君, 胡秀丽. 老年公寓老年人身体运动姿态检测系统的设计[J]. 电脑编程技巧与维护, 2021(02):110-112.
- [2] 秦宏伟, 陈春雨, 成宝芝. 基于 Cortex-M3 内核的 STM32 微控制器研究与电路设计[J]. 大庆师范学院学报, 2013, 33(06):44-47.
- [3] 李欣洋, 韩付荣, 刘正轩, 凌翔. 基于 STM32 的踝关节步态采集处理系统设计[J]. 实验科学与技术, 2020, 18(06):40-45.
- [4] 张琥石, 林伟龙, 杨发柱, 黄向军, 金星伊, 陈洁, 罗晓春, 张盈盈, 甘辛. 基于 ESP8266 Wi-Fi 模块的物联网体温监测系统[J]. 物联网技术, 2020, 10(12):32-35.
- [5] 冯娜娜, 冯娟娟, 杨延宁, 安厚儒, 郝介一. 基于 ESP8266 的可视化温湿度实时监测器的设计[J]. 延安大学学报(自然科学版), 2020, 39(04):54-57+62.
- [6] 张宏伟. 基于 STM32 的智能环境监测系统设计与实现[J]. 大庆师范学院学报, 2020, 40(06):91-98.
- [7] 姜志宪, 张伟, 王福虎. 基于 STM32 物联网控制器方案设计[J]. 科技风, 2020(17):13-15.
- [8] 丁莹芝, 毕凌志. 基于 STM32 的无线心率实时监测装置[J]. 云南化工, 2020, 47(12):160-161+164.
- [9] 王毅, 杨志, 黄晓英, 聂云浩, 徐建. 基于 STM32 单片机的人体心率采集系统[J]. 电子世界, 2016(20):51-52.
- [10] 刘劲松, 朱纯午, 杨莉. 一种基于 STM32 和 PPG 技术的腕戴式心率计设计[J]. 电子技术应用, 2016, 42(10):62-64+68.
- [11] 杨其城, 阮银萍. 基于 APP 远程控制的智能家居设计[J]. 物联网技术, 2021, 11(03):68-69+73.
- [12] 刘靓. 浅谈嵌入式 TCP/IP 协议单片机技术在网络通信中的应用[J]. 计算机产品与流通, 2020(10):74-84.
- [13] 刘木泉. 基于 TCP/IP 协议多通道双向端口状态互传设备[J]. 现代信息科技, 2020, 4(14):46-49+53.
- [14] 钱幸, 王思谨, 李基瑞. 基于 TCP/IP 的网络健康在线监测系统设计[J]. 通信电源技术, 2020, 37(06):48-49.

[15]孙志勇, 龚琼, 苏大明, 王新生, 冯亦宁. 基于 STM32 的多参数手环的系统设计[J]. 科技视界, 2016(27):444-445.

附录

附录 1 基于 STM32 运动检测系统的总硬件电路图：



附录 2 系统硬件 PCB 图:

