

分类号_____

学校代码 10487

学号 M202073426

密级_____

华中科技大学

硕士学位论文

(学术型☐ 专业型☒)

智能手环关键技术设计与实现

学位申请人：葛俊旗

学 科 专 业：计算机技术

指 导 教 师：阳富民 教授

答 辩 日 期：2022 年 05 月 22 日

**A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Master Degree in the Professional Master Degree**

**Design and Implementation of Key Technologies of
Intelligent Bracelet**

Candidate : GE Junqi
Major : Computer Technology
Supervisor : Prof. YANG Fumin

Huazhong University of Science and Technology
Wuhan 430074, P. R. China
May, 2022

摘要

随着信息技术的不断发展,中小学教育的信息化也在逐步升级,一系列可穿戴设备也加入了中小学的日常教学、生活中。智能手环无疑是可穿戴设备中最亮眼、最普及的设备。从智能手环的诞生开始,心率检测以及步数统计就成为了智能手环最基本也是最广为人知的用途。除此之外,借助无线通信技术,可以定制手环一些在校园中应用的独特功能。

以主控芯片 STM32WB55CGU6、心率传感器、加速度传感器为基础,设计并实现了一款手环。硬件上给出了传感器与主控的连接配置方案。软件上给出了手环端在心率检测、步数统计、低功耗蓝牙通信、校园特色功能方面的工作流程。

对于手环心率检测功能,介绍了基于反射式光电容积脉搏波的心率检测原理,对获取的原始光电容积脉搏波信号运用自适应滤波技术进行了降噪处理,实现了一个利用干净光电容积脉搏波信号的心率计算方法。详细描述了心率传感器的驱动实现技术,实现了心率传感器初始化、获取原始光电容积脉搏波数据的功能。对于手环步数统计功能,介绍了三轴加速度计步原理,实现了基于三轴加速度的波峰检测计步算法。详细描述了加速度传感器的驱动设计,实现了加速度传感器初始化、获取三轴加速度数据流程。对于低功耗蓝牙通信功能,分别完成了基于连接、广播两种方式的低功耗蓝牙通信。基于连接的方式应用于手环与家长端 Android 手机 APP 之间;基于广播的方式基于教师端手机 APP、教室蓝牙网关、学生手环,应用于校园教学生活场景之中,完成校园特色功能的实现。

分别测试了手环的心率检测、步数统计、低功耗蓝牙通信、校园特色功能。测试结果表明,静态心率检测准确,但动态心率检测有一定误差。计步误差随着步数的增多逐渐减小,可以满足日常计步需求。手环与其他设备之间蓝牙数据通信交互正常。

关键词: 微控制器; 传感器; 心率检测; 步数统计; 低功耗蓝牙

Abstract

With the continuous development of information technology, the informatization of primary and secondary education is also gradually upgrading, and a series of wearable devices have also joined the daily teaching and life of primary and secondary schools. Smart bracelet is undoubtedly the most dazzling and popular wearable device. Since the birth of the smart bracelet, heart rate detection and step count have become the most basic and well-known uses of the smart bracelet. In addition, with the help of wireless communication technology, the bracelet can be customized with some unique functions applied in the campus.

Based on the main control chip STM32WB55CGU6, heart rate sensor and acceleration sensor, a bracelet is designed and implemented. In terms of hardware, the connection configuration scheme between sensor and main control is given. In terms of software, the workflow of the bracelet in heart rate detection, step count, Bluetooth low energy communication and campus characteristic functions is given.

For the heart rate detection function of bracelet, the heart rate detection principle based on photoplethysmography is introduced. The original photoplethysmography signal is denoised by using adaptive filtering technology, and a heart rate calculation method using clean photoplethysmography signal is realized. The driving technology of the heart rate sensor is described in detail, and the functions of initializing the heart rate sensor and obtaining the original photoplethysmography signal data are realized. For the step count function of bracelet, the step principle of three-axis accelerometer is introduced, and the step algorithm of wave crest detection based on three-axis acceleration is realized. The driving design of the acceleration sensor is described in detail, and the technological process of initializing the acceleration sensor and acquiring the three-axis acceleration data are realized. For the low-power Bluetooth communication function, the low-power Bluetooth communication based on connection and broadcast is completed respectively. The connection based method is applied between the bracelet and the parent mobile APP; The broadcast based method is based on the teacher's mobile APP, classroom Bluetooth gateway and student bracelet. It is applied to the scene of campus teaching and life to

complete the realization of campus characteristic functions.

The heart rate detection, step count , Bluetooth low energy communication and campus characteristic functions of the bracelet are tested respectively. The test results show that the static heart rate detection is accurate, but the dynamic heart rate detection has a certain error. The step counting error decreases gradually with the increase of the number of steps, which can meet the daily step counting requirements. The Bluetooth communication of between the bracelet and other device is normal.

Key words: Micro controller, Sensor, Heart rate detection, Step counting, Bluetooth low energy

目 录

1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 手环研究现状.....	2
1.3 本文主要工作.....	7
1.4 本文章节结构.....	8
2 智能手环总体设计	9
2.1 智能手环功能概述.....	9
2.2 智能手环硬件设计.....	9
2.3 智能手环软件设计.....	12
2.4 本章小结.....	17
3 基于心率传感器驱动的心率检测	18
3.1 PPG 获取心率方法	18
3.2 心率传感器数据传输方案	26
3.3 心率传感器驱动设计与实现	29
3.4 本章小结.....	35
4 基于加速度传感器的计步算法实现	36
4.1 计步原理及计步算法实现	36
4.2 加速度传感器驱动设计与实现	40
4.3 本章小结.....	44
5 低功耗蓝牙通信实现	45
5.1 BLE 协议栈	45
5.2 基于连接的 BLE 通信过程.....	47
5.3 手环端低功耗蓝牙通信实现	48
5.4 Android 端低功耗蓝牙通信实现	50
5.5 校园特色功能实现.....	53
5.6 本章小结.....	56
6 实验测试与结果分析	57
6.1 心率检测测试.....	57
6.2 计步测试.....	59
6.3 BLE 通信测试	60
6.4 本章小结.....	63
7 总结与展望.....	64
7.1 总结.....	64
7.2 展望.....	64

参考文献.....	66
-----------	----

1 绪论

1.1 研究背景与意义

近年来,随着科技和经济的逐步发展,智能设备越来越多的出现在人们的生活之中。从上个世纪七十年代以来,麻省理工学院陆续推出智能眼镜、智能手表、智能鞋子等可穿戴设备^[1],伴随着嵌入式芯片的逐步发展,各种智能可穿戴设备逐渐被研发出来走向市场,也成为了各大科技厂商产品线中不可或缺的一员^[2]。

智能手环价格低廉、功能强大、穿戴方便舒适,在各种可穿戴智能设备中脱颖而出,可以被认为是使用最为广泛的可穿戴式智能设备^[3]。因为其体积轻巧、佩戴舒适、功耗低等特点,在日常生活、锻炼中不会轻易离身。用户对于手环的长时间贴身佩戴,使得手环有了成为随身健康检测设备的可能。健康监测一般指医疗监测和运动监测两种,医疗监测是指通过监测一些人体健康指标来反应用户当前的身体状况,并且对于一些不良情况及时给出预警;运动监测则是可以在用户锻炼身体时实时采集一些运动数据,收集反馈给用户进行分析,以便进一步制定更好的运动计划^[4]。

当下科技的发展已经蔓延至社会中的各个角落,信息化建设广泛的应用于工业、农业、医疗、教育等领域。智慧校园近年来被提出,旨在通过信息化手段为教师、学生提供更创新高效的教学辅助、更方便周到的校园生活。目前在高校中信息化的建设相对已经比较成熟完善,但在中小学教育中信息化建设尚未得到过多关注。在2019年2月,国务院颁发的《中国教育现代化2035》对我国当前中小学教育信息化建设作出了要求,从此我国中小学教育的发展有了明确方向^[5],越来越多的关注来到了中小学的信息化建设上。

手环除了基本的检测功能之外,还可以通过一些无线通信技术与外界设备进行联动延伸出更多的功能。作为一款不携带娱乐功能的小终端也适合在校学生佩戴使用^[6]。本文来源于精伦电子股份有限公司的智慧校园项目,围绕开发设计一款针对中小学生在使用的智能手环而展开。除了完成基于心率传感器、加速度传感器的心率检测、步数统计功能以外,还通过利用低功耗蓝牙技术,配合智能手机、蓝牙网关等

设备完成手环上一些校园中特色功能的实现。

1.2 手环研究现状

1.2.1 心率测量方法

目前对于心率的检测方法主要有两种，一种是心电信号法，另外一种是光电体积法。

(1) 心电信号法

人体的心脏主要由窦房结来控制收缩和舒张来完成泵血的功能，而窦房结的神经信号可以被看做是一个电信号。这种测量方法需要在人体皮肤表面贴上电极来监控电势变化，将得到的电势差记录下来得到的动态曲线就是心电图，心电图曲线中的每一个波峰都代表一次心跳^[7-9]。但这种测量使用的高精度仪器体积大、功耗大、而且需要贴上电极，被检测人无法自由移动，不满足日常实时监测的需求。

(2) 光电体积法

光电体积法是一种借助人体皮肤对于光的反射或者透射作用测量心率的方法，也叫作光电容积脉搏波描记法（Photoplethysmography, PPG）^[10-11]。当光束打到皮肤表面时，表层皮肤、脂肪、血管、骨骼等人体组织对于光的吸收程度是不变的，变化的是血管中的血液容量。心脏收缩时血管中的血液容量最少，对于光的吸收也最少；当心脏舒张时血管中的血液容量最大，对光的吸收也最大。通过光电传感器感知经过透射或者反射之后接收到的光信号的周期性变化，这种变化与心跳的周期性变化相同。光电体积法又分为透射式和反射式，透射式由于接收的透射光线强度较低，只能用于人体上相对比、薄的组织，比如手指指尖、耳垂等，而反射式可以用于全部裸露部位的皮肤。

相较于心电信号法设备体积大、功耗高、无法随身携带，光电体积法具有功耗低、易携带的特点，更适合作为可穿戴设备测量心率的方法。

2015 年，来自美国的 Joe Zareff 博士做了一项智能手表心率测量准确度的测试，对比了医疗用专业心电仪和当时市面上主流的智能手表，分别做了静止时和运动时的对比测试，结果如表 1.1 所示。

表 1.1 智能手表心率测量准确度

设备名称	测量方法	测量部位	静止状态准确率	运动状态准确率
Garmin Vivofit	ECG	胸部	81.3%	99.3%
Withings Pulse O2	反射式 PPG	手腕	95.7%	42.5%
Samsung gear fit	反射式 PPG	手腕	95.1%	37.5%
Basis carbon steel	反射式 PPG	手腕	87.5%	45.2%
Samsung Galaxy S5	反射式 PPG	指腹	95.4%	91.2%

可以看出所有设备在用户休息状态下都能得到比较精准的静止心率，但是运用反射式 PPG 方法测量心率的设备在运动状态下效果则不甚理想。这是因为反射式 PPG 由于其测量的原理（光电技术），测得的信号不仅会受到肤色、汗液、毛发影响，在运动时受到的运动干扰最为严重，这种运动干扰也被叫做运动伪迹^[12]。对于利用反射式 PPG 原理测量的设备，如何消除原信号存在的运动伪迹，提取出干净的脉搏波信号也是研究的重点、难点。因为脉搏波信号和噪声信号的频率相近，所以传统的滤波信号处理方法无能为力。目前的研究主要运用自适应滤波器来对 PPG 原始信号进行滤波处理。自适应滤波器相较于传统滤波器优势在于，可以根据前一次滤波结果的反馈来动态调整自己的滤波相关参数，Tamu T 在 2014 年证实了在 PPG 信号降噪处理方面，自适应滤波器相对于传统滤波器性能更佳^[13]。

近些年来，众多学者都做了利用自适应滤波器对 PPG 信号运动伪迹消除的研究。第三军医大学的龚俞顺利用 PPG 信号做血氧饱和度采样时，利用自适应滤波器得到了较好的滤波效果^[14]，之后 Ram M R 等人则针对自适应滤波器的自适应算法入手，设计了 AS-LMS 算法^[15-16]，有效去除了运动伪迹。除了算法之外，对于自适应滤波器的参考信号选择也起到不可忽视的作用，参考信号需要和噪声信号具有高相关性，阻抗变化值、电极变形量、三轴加速度模值三者都被验证和噪声信号相关，对于滤波效果起到了提升作用^[17-20]。

1.2.2 Android 与可穿戴设备

Android 是目前广泛搭载于移动设备之上的开源操作系统。自 2007 年发布以来就迅速占领了大部分市场，到 2015 年，使用 Android 系统的智能手机已经占到整个移动设备市场的 78.1%。除了手机和平板电脑之外，Android 系统也广泛搭载于智能电视、数码相机等移动设备上，如今全球搭载 Android 系统的移动设备已达到 50 亿之巨^[21]。

随着 Android 市场的空前强大，众多可穿戴设备厂商纷纷为自家设备与 Android 手机提供互联服务。自 Android4.3 版本官方支持低功耗蓝牙（Bluetooth Low Energy, BLE）之后，Android 手机与智能设备之间的联系进一步加强，可穿戴设备关联安卓手机、安卓手机为设备安装适用 APP 的模式成为了常态^[22]。

可穿戴设备一般集成多种传感器，可以实现的功能越来越多，但是受限于便携性，体积一般不大，只能显示一些基本的简单信息，比如时间、步数、心率等等。用户要对设备实现个性化设置，只在设备上操作也很不人性化，所以一般都采用手机端 APP 操作再向设备发送命令的方式。同时，穿戴式设备算力一般也不够强大，无法进行复杂的信息处理，设备的续航能力也是考虑的重点因素。

1.2.3 通信总线传输协议

对于嵌入式设备，合理地选取主控和传感器芯片之间的通信协议可以大幅提高数据传输的效率。根据不同的传输需求，选择不同的传输协议是至关重要的，常见的串行总线协议有 SPI、I2C 等。

SPI 是常用的全双工高速同步串行设备通信协议^[23]，相比 I2C 和 UART 协议有着固定的位数限制，SPI 可以发送任意数量连续的数据。SPI 设备分为主机和从机，主机作为控制设备，而从机一般是显示设备、传感器等等。SPI 在数据传输时需要四条数据线，分别是主机输出从机输入线（Master Output Slave Input, MOSI）、主机输入从机输出线（Master Input Slave Output, MISO）、串行时钟信号线（Serial Clock, SCLK）以及片选信号线^[24]。由于有 MOSI 以及 MISO 两条数据线，所以 SPI 支持全双工通信。SPI 在每个 SCLK 线控制的时钟周期传输 1bit

的数据,时钟信号频率由主机来制订,从机遵循,所以通信是同步的。同时在 SPI 总线上可以挂载多个设备,主机可以与多个从机进行全双工的数据传输。

I2C 总线是飞利浦公司开发的一种双向同步串行总线,两条线即可实现通信,一条数据线 (SerialData, SDA) 负责双方的数据传输,一条时钟线 (SerialClock, SCL) 来控制时钟信号同步^[25]。I2C 传输数据是以包的形式来传输,每个包都有开始位和结束位。I2C 总线可以有一个主机控制多个设备,也可以多个主机控制多个设备。通信包中还有设备地址帧,每个从设备都有一个 7 位地址,用于主从之间的设备识别,还有一位读写位,高电平为发送数据,低电平就是请求数据。每次传输,数据帧之后都有 ACK 位来验证数据发送接收状态是否正确。

1.2.4 低功耗蓝牙协议

在 WiFi、蜂巢网络、RFID、ZigBee、蓝牙等众多无线通信技术中^[26-27],蓝牙应用广泛,无论是在智能手机上,还是各种智能家居、穿戴设备上都成为了标配。随着需求的不断提升,蓝牙技术也一直在更新换代,经历最初的蓝牙 1.0 版本到如今蓝牙 5.0 版本多次提升,蓝牙不仅数据传输效率得到了飞速的发展,信息传输的可靠性也有了极大的提升^[28]。蓝牙技术受限于距离的限制,主要用于个人无线网络,很多相关的应用对于传输速率并不敏感,反而对于运行和待机时功耗有更高的要求。为此,在蓝牙 4.0 规范中加入了 BLE 技术。

蓝牙 BLE 技术是蓝牙 4.0 规范相比以前的蓝牙规范做出的最重要的升级^[29]。最重要的特征就是低功耗,除此之外,还具有成本低廉,传输可靠,信息安全等优秀特性。BLE 相对于传统蓝牙主要实现以下几点来做到低功耗^[30]:

(1) 低功耗蓝牙的广播频段和广播射频持续时间相比传统蓝牙有所减少。传统蓝牙有 32 个频段进行广播,而低功耗蓝牙仅仅使用三个广播频道,每次广播时的射频持续时间也从几十毫秒下降到了几毫秒。

(2) 低功耗蓝牙每次通信只携带少量数据,但这也决定了它传输的数据不适用于高质量的视频、音频传输,但对于少量信息,比如心率计、血压计、温度传感器等来说是相当合适的。

(3) 低功耗蓝牙尽最大努力来保证发送和接收任务的尽快完成, 一次任务需要的扫描、链接、发送、认证、结束只需要 3 毫秒即可结束, 完成之后就会暂停发送, 相比于传统蓝牙, 每次连接任务需要几百毫秒, 而且之后还要继续保持连接, 极大地减少了能量的消耗。

目前主流的智能手机操作系统都对低功耗蓝牙协议作出了支持, 这对于 BLE 技术在各种移动设备上的推广起到了极大的助力作用, 消费者们逐步开始选择更多可以配合智能手机使用的可穿戴设备。除了可穿戴设备之外, 在医疗健康领域, BLE 也有众多的应用, 使用 BLE 作为无线通信技术的血压计、血糖计层出不穷。除此之外, 传统家用电器厂商也在寻求自家产品与智能化的趋势结合, 通过蓝牙 BLE 技术来控制家中的空调、加湿器、电灯等已得到了广泛的推广, 早已不是什么新鲜事^[31-32]。

1.2.5 当前存在问题

虽然手环自问世以来已经发展多年, 但仍存在问题如下:

(1) 心率准确率不易提高。获取 PPG 信号的质量与许多因素相关。手环佩戴于手腕处, 但手腕处不同于指尖或者耳垂, 此处皮肤组织更厚、血流速度更缓慢, 检测到的信号微弱得多。不同的个体差异如体毛、肤色等也是影响 PPG 信号质量的因素。除此之外, 运动状态下的心率测量相比静态下的心率测量更为困难, 这是由于静态下血液的流动仅仅由心脏收缩控制, 而在动态下运动产生的加速度也会引起血液在血管中的流动变化, 这种变化叫做运动伪迹, 运动伪迹的消除是检测动态心率的重难点。

(2) 计步容易误判。手环计步检测是基于三轴加速度的, 手环佩戴于手腕之上, 只要是手腕的移动都会引起三轴加速度的变化, 容易将其误判为走路。在不同场景下, 如走路、跑步、爬楼梯等, 加速度的变化也不尽相同, 需要通过大量的测试来不断调整不同场景下计步判断的标准。

(3) 设备功耗问题。手环是可穿戴设备, 如果因为续航问题需要经常取下充电, 会大大降低用户体验, 也丧失了可穿戴设备实时监测的意义。手环中主要耗电的部

分在于屏幕、光电心率传感器、蓝牙通信等。要在软件上进行合理的设计，比如采集外部设备数据采取中断模式而不采轮询、合理调度传感器的开启与关闭、选取功耗更低的通信协议等等。

(4) 手环适合作为中小学学生的日常学习辅助设备，但没有专用设备。手环不具备娱乐功能，但有触摸屏和蓝牙无线通信功能，适合作为中小学学生的日常学习辅助设备。当前市面上的手环均是针对普罗大众，并没有对于一些特定使用场景做出定制化。本文所研发的手环除了具备手环基本的功能之外，还实现了一些针对中小学生在生活上的特制化功能。

1.3 本文主要工作

本文基于精伦电子股份公司智慧校园项目，实现一款适用于中小学学生的手环。主要目标在于实现智能手环上的心率检测、步数统计、低功耗蓝牙通信功能，并且借助蓝牙实现一系列区别于其它手环的校园中专用功能。主要工作如下：

(1) 完成手环系统的总体设计。在硬件方面，介绍手环选用的主控芯片、心率传感器、加速度传感器，给出硬件设备的连接配置方式。软件上，利用主控芯片提供的定时器、调度器，完成心率模块、计步模块、蓝牙通信模块在手环系统上的总体设计。

(2) 完成心率检测功能。设计并实现手环中心率传感器的初始化流程、数据获取流程，从中获取到原始 PPG 数据。利用光电容积反射法测量心率原理，对于 PPG 数据进行滤波处理，并且实现一个利用滤波后 PPG 数据计算心率的算法。

(3) 完成步数统计功能。设计并实现加速度传感器的初始化、获取加速度数据流程。利用根据三轴加速度数据计步原理，实现一个基于波峰检测的步数统计算法。

(4) 完成 BLE 通信功能。给出手环与手机双方通信过程，设计双方通信交流的数据包格式。分别在手环端和手机端实现 BLE 通信模块。实现基于广播形式的校园特色功能。

(5) 设计实验对手环进行测试。选取不同人群，在不同场景下，对比其他市面

在售手环，对手环心率功能、计步功能测试，也对手环端与手机端之间的数据同步进行测试验证。

1.4 本文章节结构

全文分为 7 章，具体内容如下：

第一章 绪论：简要介绍智能手环当前研究背景，研究现状，并且给出全文主要的研究内容，章节安排。

第二章 智能手环总体设计：分别在手环硬件、手环软件方面给出智能手环的总体设计。硬件设计方面，给出手环主控芯片与外设（心率传感器、加速度传感器）的简要介绍，以及主控与外设之间的连接配置。软件设计方面，给出手环主程序对于心率检测任务、步数统计任务以及蓝牙通信的调度流程。

第三章 基于心率传感器驱动的心率检测：介绍利用 PPG 信号获取心率的原理，并且利用自适应滤波技术对 PPG 信号降噪处理，实现一个根据干净 PPG 信号获取心率的算法。完成心率传感器的初始化、获取原始 PPG 数据流程。

第四章 基于加速度传感器的计步算法实现：介绍根据三轴加速度计步原理，实现一个根据三轴加速度数据统计步数的算法。完成加速度传感器的初始化、获取三轴加速度数据的流程。

第五章 低功耗蓝牙通信实现：简要介绍 BLE 协议栈中对于程序编写直接接触的 GAP 层、GATT 层，给出 BLE 通信过程设计，分别在手环端与 Android 手机端实现 BLE 通信的程序，实现手环在校园中使用的特色功能。

第六章 实验测试与结果分析：与当前市面上其他在售手环作为对比，分别对心率检测、步数统计、BLE 通信功能进行测试。

第七章 总结与展望：总结全文做出的工作内容，总结尚且存在的不足之处，并对未来可以持续优化的工作做出展望。

2 智能手环总体设计

本章给出智能手环的需求分析和设计，给出手环系统硬件、软件方面的总体设计，以便后续章节详细对具体功能的实现进行描述。

2.1 智能手环功能概述

智能手环不仅具有手环最基本的功能，例如心率检测、步数统计等等，而且还针对学生在校园中的应用场景设计了一些方便实用的功能，具体设计功能如下：

(1) 心率检测

心率检测是手环的经典功能，心率作为人体重要的健康指标，对它进行监测不仅可以对日常心血管疾病进行预防，还能提示在剧烈运动中心脏的突发不适情况。除此之外，掌控自身心率也可以更合理地安排运动计划。

(2) 步数统计

步数统计也是手环的经典功能之一，相比于智能手机的步数统计功能，手机在日常生活中并不一定随时携带在身上，难免会有漏记的步数，而手环这类可穿戴设备一般很少会摘下，对于步数的统计更加全面。

(3) 无线传输功能

智能手环的屏幕和性能有限，想要实现更多的功能依赖于与智能手机之间进行联动。一些针对于校园特有的功能，比如教师通知发放、课程表发放、运动计划下发、课堂点名、抢答等功能都是基于手机和手环之间的无线传输，无线传输功能使用低功耗蓝牙技术实现。

2.2 智能手环硬件设计

智能手环内部集成心率传感器、加速度传感器、充电管理芯片以及4MB串行Flash。主控制芯片采用ST公司的STM32WB55CGU6，该芯片内部集成Cortex-M4核和蓝牙5.0模块。手环显示模块采用1.1寸126 x 294分辨率的彩色AMOLED显示屏，AMOLED屏显示区域可以滑动触摸，屏幕下方可单点触摸。手环内置132mAh可充

电锂电池。硬件总体结构如图 2.1 所示。

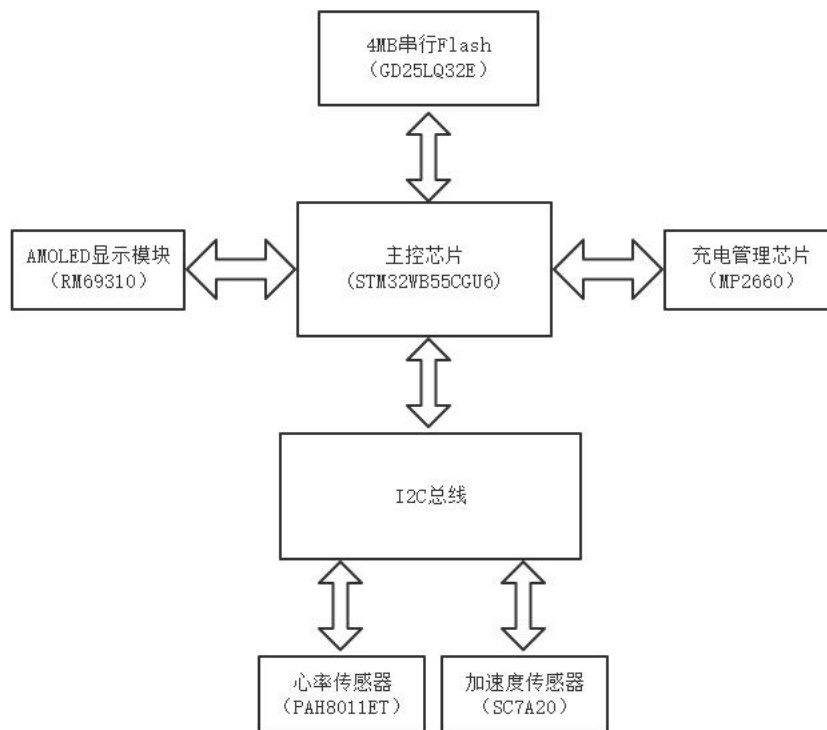


图 2.1 手环硬件架构图

(1) 主控芯片

主控芯片选用的 STM32WB55CGU6 是 ST 公司为了满足各种 BLE 应用场景推出的一款 2.4G 双核无线 MCU^[33]。该 MCU 具有双核架构，一颗 Cortex-M4 运行用户应用程序和控制外设，一颗 Cortex-M0 专门用来处理射频协议栈，这种应用和协议栈分开的架构使得开发灵活，开发者无需关心协议栈方面，也可以降低功耗。该芯片在无线功能方面的强大能力，资源使用的灵活性，功耗管理都做出了改进，为开发者提供了丰富的支持，十分适合作为智能手环的主控芯片来使用。

(2) 心率传感器

心率传感器采用 PAH8011ET。PAH8011ET 是一款低功耗光学心率感测芯片，它不仅整合了 DSP，而且还有两个绿色 LED，一个红外线 LED 结合在芯片之上。DSP 用于处理、获取光电容积脉搏波数据。PAH8011ET 在系统中始终作为从机来实现，需要微控制器或者处理器作为主机成为一个完整的系统设计。主机通过串行接口获取 PPG 数据，从而在应用层进行复杂的数据处理。

PAH8011ET 作为光学心率传感器,主要用来采集原始 PPG 数据提供给主控芯片,传感器的引脚需要和微控制器的引脚进行连接,建立好数据传输通道,就可以实现主控到传感器设备之间的信息交互。具体连接方案如表 2.1 所示。

表 2.1 主控芯片与 PAH8011ET 连接方式

MCU 引脚	心率传感器芯片 (PAH8011ET) 引脚	MCU 定义说明
PB4	HR_INT1	输入,数据准备好(ready)中断
PA13	HR_INT2	输入,接触检测中断(设计预留,引脚默认为 MCU_SWD 信号,HR_INT2 信号不使用)
PE4	HR_PDN	输出,控制 PAH8011ET 进入低功耗模式,高电平进入低功耗模式,低电平退出低功耗模式
PA10	I2C1_SDA	输入输出,I2C1 数据信号,与 AMOLED 触摸芯片,充电管理芯片,加速度芯片共用
PB6	I2C1_SCL	输出,I2C1 时钟信号,与 AMOLED 触摸芯片,充电管理芯片,加速度芯片共用

SCL 为 I2C 传输的时钟管理引脚,与主控芯片的 PB6 端口连接,主控通过这个端口来同步它和 PAH8011ET 的时钟周期。

I2C_SDA 为 I2C 数据传输信号,在每一个时钟周期传输一个比特的信号。

HR_PDN 与 PE4 端口连接,可以控制 PAH8011ET 的低功耗模式切换,拉高进入低功耗模式,拉低则退出。

HR_INT1 与 PB4 连接,当 PAH8011ET 处于中断模式下,PPG 数据准备好时触发该中断。

(3) 加速度传感器

加速度传感器采用 SC7A20,SC7A20 传感器是杭州士兰微电子推出的一款高精度三轴加速度传感器芯片。该传感器芯片体积小、功耗低、功能丰富。芯片通过 I2C 或者 SPI 协议与主控芯片进行通信。两个中断管脚提供多种内部检测中断信号,可以进行不同场景下的运动检测。内置 FIFO 缓冲区、高精度校准模块,可以补偿传感器的失调误差和增益误差。多种不同加速度范围 $\pm 2G$ 、 $\pm 4G$ 、 $\pm 8G$ 、 $\pm 16G$ 可选。

数据输出频率 1Hz、10Hz、25Hz、50Hz、100Hz、200Hz、400Hz 均可选择，可以灵活输出三轴加速度数据。

三种工作模式可选，分别是正常模式（Normal mode）、掉电模式（Power Down）、低功耗模式（Low power）。正常模式能保证更高的三轴输出数据精度，低功耗模式能减少电流消耗。在正常工作模式下，能保证更好的数据采样精度，XYZ 三轴的数据可按照设置好的数据更新率周期性的输出三轴数据寄存器。在掉电模式下，除了 I2C 和 SPI 以外所有的内部模块都被关闭以节省功耗，所有寄存器之前配置过的参数数据都被保存，而三轴输出的数据寄存器的数据不再更新，仍维持上一次正常工作时采样的数据。在低功耗模式下，主要特点是功耗低，数据输出情况和正常模式一致，三轴数据可按照设置好的数据更新频率周期性地输出三轴数据。

SC7A20 与主控芯片之间的引脚连接如表 2.2 所示。

表 2.2 SC7A20 与主控引脚连接配置

MCU 引脚	加速度传感器(SC7A20)引脚	MCU 定义说明
PA10	I2C1_SDA	输入输出，I2C1 数据信号，与 AMOLED 触摸芯片，心率传感器，充电管理芯片共用
PB6	I2C1_SCL	输出，I2C1 时钟信号，与 AMOLED 触摸芯片，心率传感器，充电管理芯片共用
PB1	GS_INT	输入，中断输入

加速度传感器提供 I2C 和 SPI 两种短距离数据传输方式，这里选取 I2C 作为主控与传感器的数据传输方式。加速度传感器的串行时钟线连接主控芯片的 PB6 引脚，串行数据线连接主控芯片的 PA10 引脚，与心率传感器共用。中断源信号引脚与主控芯片的 PB1 引脚相连。

2.3 智能手环软件设计

智能手环包含屏幕显示、屏幕触控、心率检测、步数统计、无线通信等功能，本节只介绍主控芯片的软件架构，心率任务、计步任务的调度，以及手环端与手机端通信的软件设计。

(1) 主控芯片支撑软件

手环程序属于单片机程序，没有操作系统，完成硬件初始化后，对任务的调度主要通过调度器和定时器来进行处理。调度器最多支持 32 个任务，支持主动请求执行任务、暂停任务、恢复任务，同时支持两个任务优先级，调度方式采取非抢占式基于优先级调度算法，相同优先级选取任务 ID 小的优先执行。调度器相关函数如表 2.3。

表 2.3 调度器相关函数

函数	方法
UTIL_SEQ_Idle	没有任务执行时调用，使 MCU 进入低功耗状态。
UTIL_SEQ_Run	请求调度器开始执行在 mask_bm 中注册等待执行的任务。
UTIL_SEQ_RegTask	在调度器中注册一个任务，将回调函数保存在数组之中。
UTIL_SEQ_SetTask	请求调度器执行给定 ID 任务

使用调度器时，注册好需要执行的任务，在 while(1) 循环中调用 UTIL_SEQ_Run 函数即可。

注册任务时需要提供任务 ID (0-31)、以及任务需要执行的函数。有一个大小为 32 的 TaskCb 函数指针数组，注册任务的实质就是将执行指定 ID 任务的函数保存在 TaskCb 数组之中，数组下表为任务 ID，数组元素为任务回调函数。

调度器支持两个优先级，使用两个 32 位整数 task_set_priority0、task_set_priority1 作为位图来保存需要执行的任务 ID。位图中比特位为 1 的索引表示需要执行的任务 ID，例如 task_set_sriority1 中第 5 个比特位与第 10 个比特位为 1，则要执行任务 ID 为 5、10 且优先级为 1 的两个任务。UTIL_SEQ_SetTask 函数需要提供需要执行任务的 ID 和优先级，实质就是将对应位图的比特位置为 1。

UTIL_SEQ_Run 函数每次调用可以将 task_set_priority0、task_set_priority1 中请求的任务执行完毕，调度方式采取非抢占式基于优先级调度算法，相同优先级选取任务 ID 小的优先执行。但这样运作时，如果同一优先级经常有两个任务同时请求调用，这样只会执行任务 ID 小的任务，任务 ID 大的任务会产生饥饿现象。故引入一个 round_robin_mask 位图结构表示可以执行的任务，对应 ID 比特位上为 1 时才可以执行该 ID 的任务，这样保证了每次 UTIL_SEQ_Run 调用时，如果同一 ID 的任务多次

请求，但只能被执行一次。UTIL_SEQ_Run 函数伪代码如下：

算法 1 任务调度算法

输入：无

输出：无

```
1.  while (task_set_priority0>0 || task_set_priority1>0) do    //有任务需要执行
2.      if (task_set_priority0>0) then                        //高优先级任务优先执行
3.          cur_task_set = task_set_priority0
4.      else
5.          cur_task_set = task_set_priority1
6.      end if
7.      if(round_robin_mask & cur_task_set == 0) then        //重置 round_robin_mask
8.          round_robin_mask = ~0
9.      end if
10.     cur_task_idx =cur_task_set & round_robin_mask 的第一个不为 0 比特位位置
11.     执行 TaskCb[cur_task_idx]()                          //执行对应 ID 任务
12.     cur_task_set &= ~(1 << cur_task_idx)                //任务集合中移除执行过的任务
13.     round_robin_mask &= ~(1 << round_robin_mask) //标记本次执行过的任务
14. end while
15. 进入低功耗模式
16. return
```

除了调度器之外，还用到了定时器。STM32WB55CGU6 在内存允许的情况下最高支持 255 个虚拟定时器，支持单次模式和重复模式。定时器相关函数定义如表 2.4。

表 2.4 定时器相关函数

Function	Description
HW_TS_Init	初始化定时器服务
HW_TS_Create	创建一个虚拟定时器
HW_TS_Stop	停止一个虚拟定时器
HW_TS_Start	开始一个虚拟定时器

为了实现心率检测、蓝牙连接等功能的定时控制，使用定时器创建一个间隔为 1 秒的系统重复 heartbeat 定时器，在每 1 秒时做如下检测工作：

- 1) 每 60 秒开启心率检测。
- 2) 心率检测持续 20 秒后停止心率检测。
- 3) 每次建立蓝牙连接之后，开启 90 秒的计时。
- 4) 蓝牙连接 90 秒内有动作则重新计时，无动作自动断开连接。

(2) 心率任务

在调度器中设置心率任务，心率任务在心率传感器发生数据缓冲区中断时且当前开启心率检测时，运用 UTIL_SEQ_SetTask 函数向调度器发出执行请求。同时，心率任务不是始终执行的，每 60 秒开启一次，每次持续 20 秒结束。心率任务执行流程如下：

- 1) 读取采集到的 PPG 数据、三轴加速度数据。
- 2) PPG 数据送入心率算法之中计算心率。
- 3) 存储本次获取到的心率。

(3) 计步任务

在调度器中设置计步任务，在加速度传感器发生 FIFO WTM 中断时，运用 UTIL_SEQ_SetTask 函数向调度器发出执行请求，计步任务执行流程如下：

- 1) 读取采集到的三轴加速度数据。
- 2) 送入计步算法计算本次累积步数。
- 3) 存储本次三轴加速度数据，更新计步。

(4) 手环总控程序

1) 首先对手环设备进行初始化操作，包括外设传感器的初始化以及蓝牙通信层的初始化。向系统调度器中注册设置好的心率任务、计步任务，初始化一个重复的系统 heartbeat 定时器，每秒进行一次例行的任务检测。同时对手环 GATT 服务器进行初始化，设置好通信的服务和特征值，注册事件回调函数。

2) 设备调度器开始运作，当定时器检测到每过 60 秒时，开启心率检测，此时若发生心率传感器数据准备好中断，则执行心率任务，获取 PPG 数据送入心率算法计

算存储。心率检测持续 20 秒之后停止检测。

3) 当发生加速度传感器 FIFO 中断时执行计步任务。获取三轴加速度数据, 送入计步算法计算本次累积步数, 更新计步。

4) 接收到蓝牙客户端发送指令时, 执行注册好的回调函数, 根据定义的命令格式加以解析并且返回所请求服务的响应。手环端程序流程如图 2.2 所示。

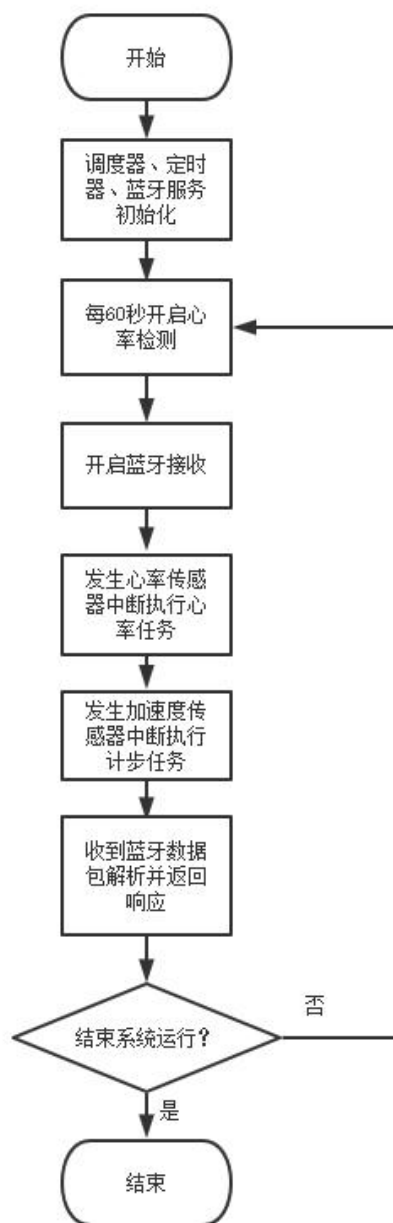


图 2.2 手环端程序流程图

2.4 本章小结

本章首先阐述了智能手环的功能需求，其次分别对智能手环的硬件、软件进行了总体设计。硬件设计方面，介绍了智能手环主要涉及到的 STM32WB55CGU6 主控芯片、PAH8011ET 心率传感器、SC7A20 加速度传感器，并且设计了主控和传感器的引脚连接。软件方面，使用调度器与定时器完成了对于心率检测、步数统计、蓝牙通信功能的流程控制。

3 基于心率传感器驱动的心率检测

介绍完手环整体设计之后，本章主要完成手环中心率检测这一重要功能的实现。首先说明了光电反射法测量心率的原理，利用基于变步长 LMS 算法的自适应滤波器对 PPG 信号做了降噪处理，实现了一个对于干净 PPG 信号获取心率的算法。其次对心率传感器与主控芯片之间使用的通信协议进行了选择，实现了双方的通信函数。最后完成了对于心率传感器的初始化，详细完成了心率传感器驱动程序的编写，给出了获取原始 PPG 数据的接口。

3.1 PPG 获取心率方法

3.1.1 反射法测量心率原理

心脏在收缩和膨胀时血管中的血容量是不同的，当血容量大时对于光线的吸收也更多，血容量少时对于光线的吸收也更少。其他组织诸如皮肤、血管、肌肉等对于光线的吸收是不变的。光电反射法通过向人体皮肤发射光线，利用光电传感器对于反射回来的光线强度进行获取，经过一系列数模处理之后，将其转换成电信号，这个电信号就是需要的 PPG 信号^[34-35]。它的波形可以认为是人体心脏周期性心跳的波形。图 3.1 展示了光线被组织吸收示意图。

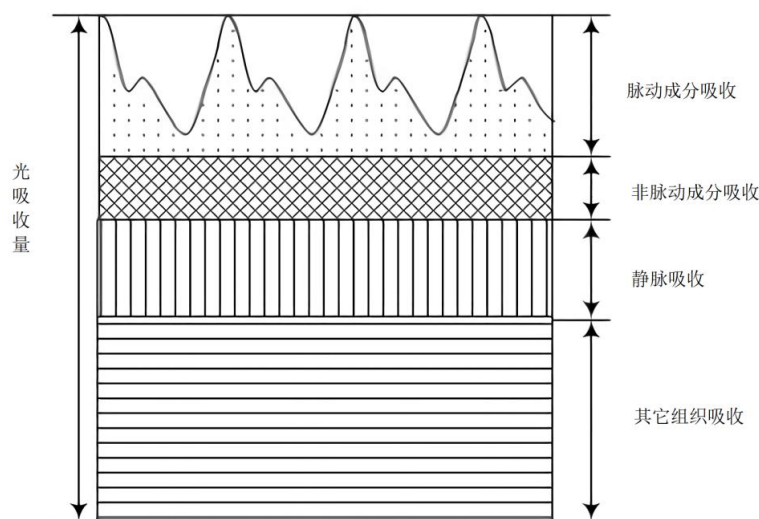


图 3.1 光线被组织吸收示意图

3.1.2 PPG 信号降噪处理

(1) 信号预处理

PPG 信号中具有实际应用价值的交流部分只占很小的比例,运用直接得到的 PPG 信号不能很好的获取到心率。应当通过滤波手段对 PPG 信号中具有实际意义的交流部分进行增强处理。另外人体心率极限范围在 40 次每分钟到 240 次每分钟之间,所以可能的 PPG 信号频率相对应在 0.66Hz 到 4Hz 之间。可以采用 Butterworth 低通滤波器对 PPG 信号滤波,将滤波器截止频率设置为 4Hz 来滤除高频噪声。

(2) 自适应滤波器

来自外界对于 PPG 信号的干扰之中,由于运动产生运动伪差的最为严重,但运动时心率的监测也是进行心率检测的重要目的。运动伪差是指人体运动时由于血管的充盈变化和光电路线的改变引起的伪差,产生的影响往往与心率信号的频谱相近甚至重叠。传统滤波器对于运动伪差的消除无能为力。

自适应滤波器相较于传统滤波器而言是非线性滤波器,不用了解信号的先验知识去设计滤波器系数,而是在滤波过程中根据每一次的滤波结果不断反馈调整滤波参数^[36]。自适应滤波器的更新方式被叫做目标函数,目标函数一般会采用最小误差的更新方式。自适应滤波器的种种特性正好符合 PPG 信号的滤波需求,在 PPG 信号去噪方面受到了广泛的关注。

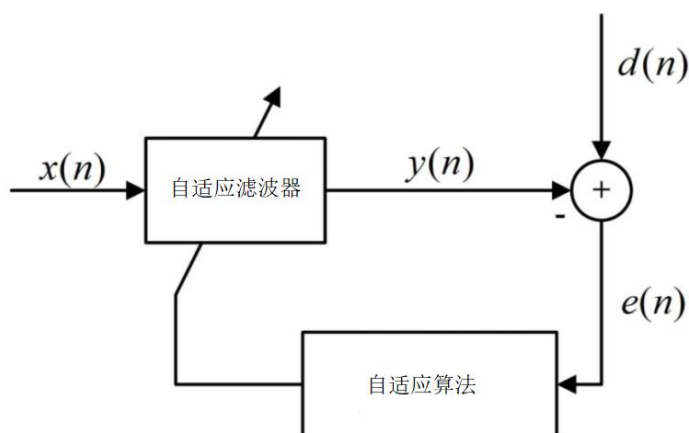


图 3.2 自适应滤波器原理

图 3.2 给出了自适应滤波器的基本原理, $x(n)$ 表示滤波器输入信号, $y(k)$ 表示经过

滤波处理之后的输出信号, $d(n)$ 表示期望信号, $d(n)$ 与 $y(n)$ 作差之后得到误差信号 $e(n)$, 作为反馈交给自适应算法来处理调整下一次处理的滤波器参数。经过历次迭代优化之后达到最优的滤波器参数, 使得输出信号 $y(n)$ 与误差信号 $d(n)$ 的误差越来越小。

(3) 基于 LMS 的自适应滤波算法

由以上可见, 自适应滤波器的性能很大程度取决于自适应算法的优劣, 自适应算法是指根据反馈对参数进行调整的算法。常用的自适应算法有两类, 分别是最小均方 (LMS) 算法与递归最小二乘 (RLS) 算法^[37]。LMS 算法尽力使得均方误差 $E[e^2(k)]$

达到最小值。RLS 算法最小化的目标是加权平方和 $\varepsilon(n) = \sum_{k=0}^n \lambda^{n-k} e^2(k)$, 其中 λ 表示遗忘因子, 范围是 $0 < \lambda \leq 1$ 。RLS 算法相较于 LMS 算法收敛速度更快, 但是运算时要求的存储空间和计算资源都高。LMS 算法计算复杂度更低, 要求的资源更少, 更适用于在单片机系统中使用。

LMS 算法应用于 PPG 信号自适应滤波之中, 输入为向量, 对于 PPG 信号而言, 对每个信号做出对应的时延就可以得到向量。输出是输入信号的线性组合, 输出信号的表达式有式 (3.1):

$$y(k) = \sum_{i=1}^N w_i(k)x(k-i) = W^T(k)X(k) \quad (3.1)$$

上式中, $X(k) = [x(k)x(k-1)\dots(k-N)]^T$ 为 PPG 原始信号输入, $W(k) = [w_0(k)w_1(k)\dots w_N(k)]^T$ 表示为自适应算法调整后的权系数。此时自适应滤波器得到的误差有式 (3.2):

$$e(k) = d(k) - y(k) = d(k) - W^T(k)X(k) \quad (3.2)$$

$e(k)$ 的绝对值越小则自适应滤波的效果越好。LMS 算法对 $e(k)$ 求得最小均方误差 $E[e^2(k)]$ 可以更显著的代表自适应滤波的性能优劣。当其达到最小时, 自适应算法经过历次迭代得到的滤波系数 $W(k)$ 最佳。 $E[e^2(k)]$ 经过推导之后有均方误差函数, 见式 (3.3):

$$E[e^2(k)] = E[d^2(k)] - 2W^T(k)E[d(k)X(k)] + W^T(k)E[X(k)X^T(k)]W(k) \quad (3.3)$$

根据 Wiener Hopf 方程可以推出 LMS 算法权系数迭代方程有式 (3.4) :

$$W(k+1) = W(k) + 2\mu e(k)X(k) \quad (3.4)$$

其中 μ 为更新步长, $W(k)$ 在每次迭代中的变化由其决定。要在多次迭代中权系数达到最优, 需要满足的条件如式 (3.5), 式中 λ_{\max} 是矩阵 $X(k)X^T(k)$ 的最大特征值。

$$0 < \mu < \frac{1}{\lambda_{\max}} \quad (3.5)$$

LMS 算法的收敛性和稳定性主要取决于步长因子 μ , 然而 μ 却是需要有取舍的。 μ 越小, 稳态误差会减小, 但是权系数更新迭代时间变长, 对于 PPG 信号这种非平稳信号追逐速度变慢; μ 越大, 收敛时间变短的同时也带来了失调量变大的副作用。

将 μ 固定为常数的 LMS 算法称作为定步长 LMS 算法, 此类算法总会有收敛速度慢或者失调大的缺点。为了解决定步长 LMS 算法的缺点, Kwong R. H. 等人提出了一种变步长 LMS 算法, 主要思想为在滤波初始阶段采用较大的步长因子加速收敛, 在收敛接近尾声时换用小的补偿因子避免失调^[38]。步长因子迭代公式如式 (3.6)、式 (3.7)。

$$\mu(k+1) = \alpha\mu(k) + \beta e^2(k) \quad (3.6)$$

$$\mu(k+1) = \begin{cases} \mu_{\max}, \mu(k+1) > \mu_{\max} \\ \mu_{\min}, \mu(k+1) < \mu_{\min} \\ \mu(k+1), \mu_{\min} \leq \mu \leq \mu_{\max} \end{cases} \quad (3.7)$$

式 (3.7) 中 μ_{\max} 由式 (3.5) 所决定, 用来保证 μ 在许可范围内, μ_{\min} 用来保证算法收敛速度不会太慢。式 (3.6) 中 α 决定本次的 μ 在多大程度上取决于上次, 一般有 $0 < \alpha < 1$, 当算法逐步收敛之后, $\beta e^2(k)$ 也随之趋于 0, 此时如果 α 太小会造成 μ 下降过快, 所以 α 一般小于 1 但接近 1。 β 为瞬时误差的权值, 当算法开始时, 误差较大, 此时如果 β 过大就会引起系统失调量增大, 故 β 一般取一个很小、大于 0 但接近 0 的值。综上所述, 使用变步长 LMS 算法可使得步长因子在开始收敛时, 步长较大收敛较快, 在即将结束收敛时减小步长获得更强的跟踪能力。

对 PPG 信号滤波时, 每次读取 20 个 PPG 数据作为一组向量, 初始化滤波器权

系数为 $W = [0, 0, 0, \dots, 0]^T$ ，计算前 20 个 PPG 数据向量的实对称矩阵，求出矩阵的最大特征值 λ_{\max} ，初始步长设置为符合式 (3.5) 的最大步长。

接着根据式 (3.2) 计算误差，根据式 (3.6)，式 (3.7) 确定本次更新的步长因子，再根据式 (3.4) 计算新的滤波器权系数，如此不断反复迭代。

随着权系数的不断更新，滤波器的输出结果也逐渐收敛。输出的式 (3.1) 中的 $y(k)$ 信号也不断逼近光电容积脉搏波信号中的运动伪差干扰，此时整个模型的输出 $e(n)$ 也就愈发接近干净的 PPG 信号。变步长 LMS 算法流程图如图 3.3。

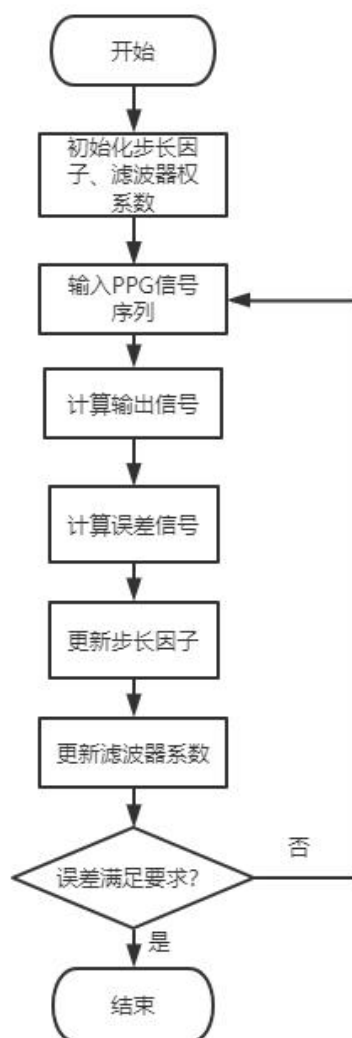


图 3.3 变步长 LMS 算法流程图

(4) 自适应滤波器在 PPG 信号的应用

利用自适应滤波器对 PPG 信号进行滤波的实际应用中，输入信号为未经去噪的 PPG 信号，要做自适应滤波还需要一个参考信号。参考信号要求与需要的信号或者噪声信号中的一个线性相关，与另一个线性无关。在实际得到的采样信号中，可以认为由三部分构成，真实需要的脉搏波信号 $S(n)$ ，由于运动产生的运动伪差信号 $N(n)$ ，环境影响产生的高频噪声 $F(n)$ ，环境产生的高频噪声经过低通滤波器处理之后可以认为已经极大衰减，设经过处理之后的信号为 $D(n)$ 有式 (3.8)。

$$D(n) = S(n) + N(n) \quad (3.8)$$

由于脉搏波信号和运动伪差信号相互之间独立且不相关，想要得到干净的脉搏波信号，可以间接通过参考信号得到运动伪差信号，再用低通滤波之后的采样信号减去运动伪差信号来得到干净的脉搏波信号。在 2007 年，Yoon 等人的研究中证实了加速度信号和运动伪差之间存在强相关性，可以作为运动伪差的参考信号^[39]。一般加速度计为三轴加速度计，分别代表 x, y, z 三轴上的加速度大小，为了避免采取单一轴上加速度作为参考信号的局限性，可以取三轴上加速度的模值作为参考信号，参考信号有式 (3.9)。

$$acc(n) = \sqrt{acc_x(n)^2 + acc_y(n)^2 + acc_z(n)^2} \quad (3.9)$$

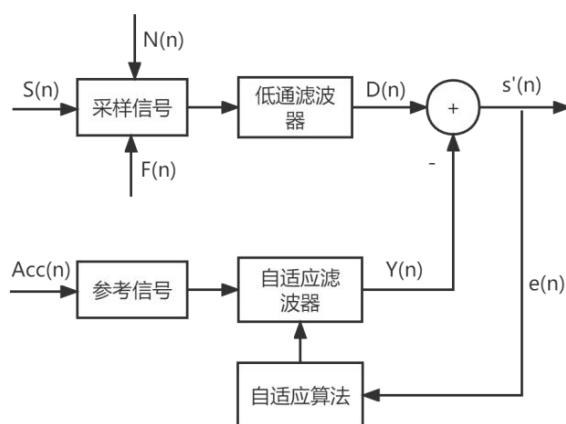


图 3.4 自适应滤波器在 PPG 信号上的应用

图 3.4 给出了自适应滤波器在 PPG 信号上的应用。在实际应用之中, 采样信号经过低通滤波器之后输出信号 $D(n)$ 之中包含实际脉搏波信号 $S(n)$ 和运动伪差 $N(n)$, $Acc(n)$ 作为参考信号输入到自适应滤波器之中, 得到输出信号 $Y(n)$, 将 $D(n)$ 与 $Y(n)$ 作差获得 $e(n)$ 作为误差反馈信号, 经过自适应算法处理之后得到新的滤波器系数反馈给滤波器, 经过数次迭代之后使误差 $e(n)$ 越来越小, 此时 $Y(n)$ 也越来越接近 $N(n)$, 同样的系统输出 $s'(n)$ 与 $s(n)$ 也越来越接近, $s'(n)$ 即为滤波获得的干净 PPG 信号

3.1.3 心率计算方法

在干净的脉搏波信号之中, 每一对波峰波谷可以被看做是心脏的一次收缩运动, 故可以通过计算波峰之间的间隔来获得心率。为了达到实时的心率监控效果, 本文选取每四个波峰为一组数据来计算心率, 具体心率可由式 (3.10) 计算得到

$$BPM = \frac{60 * 10^3}{t_1 - t_4} * 3 \quad (3.10)$$

式 (3.10) 中, t_1 表示最后一个检测到的波峰时刻, t_4 表示倒数第四个检测到的波峰的时刻, 单位都是毫秒。

但是人的脉搏波波形并没有这么简单, 还与人体一些健康情况相关, 比如血管壁弹性、血压、血脂含量等等。这些人体的生理因素与人体年龄息息相关。不同年龄段特征的人的脉搏波波形可大致分为低阻型、中阻型、高阻型三类。低阻型波形一般常见于年轻人中, 该类型重搏波波峰与主波波峰区别明显。中年人中常见中阻型, 这类波形主波波峰仍然明显, 但是从主波波峰到重搏波波峰变化趋势放缓, 重搏波不明显或者已经消失。在老年人中, 这类人群一般血管弹性变差, 阻力变大, 高阻型脉搏波中主波波峰和重搏波波峰区别不明显, 重搏波或已经消失。

由于脉搏波中总有重搏波的存在, 不同年龄段个体的脉搏波波形也不尽相同, 上述心率算法无法区分主波波峰波峰和重搏波波峰, 为了避免因为重搏波的误判和不同年龄个体的不同情况, 需要设计一个具有通用性的心率计算算法。重搏波波谷明显往往高于主波波谷, 而主波波峰也显著高于重搏波波峰。故主波波峰波谷差值显著高于重搏波波峰波谷, 可以根据主波波峰波谷和重搏波波峰波谷的差值来分辨主

波波峰和重搏波波峰。具体的阈值需要根据测得的数据来动态进行更新，当差值小于阈值时则不作为主波波峰记录时间。

(1) 顺序读入 PPG 信号的值，获取当前系统时间，与上一点的数据做对比，判断该点处于上升趋势还是下降趋势，并记录下来。

(2) 如果前一点是上升趋势且当前数据小于之前的数据，则前一点属于波峰点。

(3) 通过波峰和波谷差值来判断当前波峰是主波波峰还是重搏波波峰，如果是主波波峰就记录下当前时间，并且改趋势为下降，重搏峰波峰则忽略。

(4) 如果是下降趋势，且当前点数据大于上一点数据，则更新波谷为上一点的数据，趋势改为上升。

(5) 如此迭代来记录需要的时间点。当记录的波峰时刻满足 4 个时，则带入公式 (3.10) 计算心率。心率算法伪代码如下：

算法 2 心率计算算法

输入: 当前 PPG 值、系统时间

输出: 心率

```
1.  if(isUp) then                                //当前处于上升趋势
2.      if(curValOfPPG < preValOfPPG)
3.          if (preValOfPPG - ValleyVal > threshold) then //上一点为波峰点
4.              Heartrate=(60*103)/(t1-t4)*3 //记录波峰时间点，累积 4 个计算心率
5.              update threshold                    //更新阈值
6.          end if
7.          isUp = false                            //趋势改为下降
8.      end if
9.  else                                          //当前处于下降趋势
10.     if(curValOfPPG > preValOfPPG) then        //上一点为波谷点
11.         ValleyValue = preValOfPPG            //更新波谷点值
12.         isUp = true                            //趋势改为上升
13.     end if
14. end if
15. preValOfPPG = curValOfPPG                    //更新上一点值为当前点值
```

16. return heartrate

//返回心率值

3.2 心率传感器数据传输方案

PAH8011ET 同时支持 SPI 通信协议与 I2C 通信协议,这两种协议都适合心率传感器与主控芯片之间通信。考虑到主控芯片上引脚紧张,I2C 相比 SPI 只需要两条线就可以完成通信,具有优势;虽然 SPI 的传输效率比 I2C 高出很多,但是不同于要求传输速率的音频、视频数据,传感器数据每秒仅仅传输几十个字节,I2C 通信支持的吞吐量已经足够;尤其是智能手环 I2C 总线上不仅搭载了心率传感器,还有加速度传感器,AMOLED 触摸芯片等等,I2C 更适合主控和心率传感器芯片的通信。

对于 STM32 的开发方式,最早时期是使用直接配置寄存器的方式。但是这种方式在早期芯片功能不够强大、寄存器数量少时还可行,如今随着寄存器越来越多,这种方式逐渐被摒弃^[40]。ST 公司随之推出了标准库,库函数开发代替寄存器开发成为了主流。每款芯片都有自己的标准库文件,标准库文件中包含一些宏定义,也把一些外设通过结构体来封装了起来,以后再操作外设时只需要配置结构体成员变量就可以。但是这种方式由于每一个芯片都有自己的库文件,无法实现不同芯片之间的程序移植。

HAL 库是 ST 公司最新推出的 STM32 开发库,HAL 库不仅完成了配置寄存器方式的标准化,而且做到了一些特定功能的标准化。使用 HAL 库编写的程序,不同芯片之间只要使用的是相同的外设,是可以不用修改直接完全移植的^[41]。基于以上优点,本文 I2C 初始化与 I2C 通信函数都基于 HAL 库提供的。

3.2.1 I2C 初始化

(1) 配置 I2C 初始化结构体

在 STM32 中,如果要初始化一个外设,比如 I2C、SPI、UART 之类,初始化它们的寄存器是第一步^[42]。I2C 的初始化结构体 I2C_InitTypeDef 字段定义如表 3.1。

其中 I2C_ClockSpeed 设置为 10KHz, I2C_Mode 一般直接选取 I2C 模式; I2C_DutyCycle 占空比配置为 2:1,表示低电平高电平比例为 2:1, I2C_OwnAddress1 为 I2C 总线上的 I2C 主机地址,配置为 0x38; I2C_Ack 配置为打开 I2C 应答,根据

连接到总线上的外设情况，I2C_AcknowledgedAddress 配置寻址地址为 7 位地址。

表 3.1 I2C_InitTypeDef 字段描述

字段	描述
I2C_ClockSpeed	I2C 传输速率
I2C_Mode	I2C 使用方式
I2C_DutyCycle	I2C 时钟线占空比
I2C_OwnAddress1	自身 I2C 设备地址
I2C_Ack	打开或者关闭响应
I2C_AcknowledgedAddress	指定设备地址长度

(2) 调用库函数写入寄存器

初始化结构体设置好之后，直接调用 HAL_I2C_Init 函数，传入 I2C_InitTypeDef 结构体句柄，即可完成 I2C 的初始化。

3.2.2 通信函数实现

对于 I2C 的通信，HAL 库一共提供超时、中断、DMA 三种通信方式，这里选取中断模式作 I2C 的通信方式。主要使用 HAL 库中的 HAL_I2C_Master_Transmit_IT 和 HAL_I2C_Master_Receive_IT 这两个函数来进行数据的传输。

在使用 HAL_I2C_Master_Transmit_IT 进行数据传输时，需要提供设备地址、发送数据缓冲区指针、需要发送数据的长度。

在使用 HAL_I2C_Master_Receive_IT 进行数据接收时，提供的参数有设备地址、接收缓冲区指针、需要读取的数据长度。

除了这两个函数以外，还有 HAL_I2C_GetState 函数来判断当前 I2C 所处的状态，主要用于通过判断当前 I2C 总线是否可用来确定通信是否完成；HAL_I2C_GetError 函数返回通信过程中的错误代码，如果发生错误则要尝试重新发送或者重新接收。

(1) 写入操作实现。实现函数 heartrate_sensor_comm_write(uint8_t addr, uint8_t data)来完成寄存器的写操作，addr 是寄存器地址，data 是需要写进寄存器的数据。流程如图 3.5 所示。

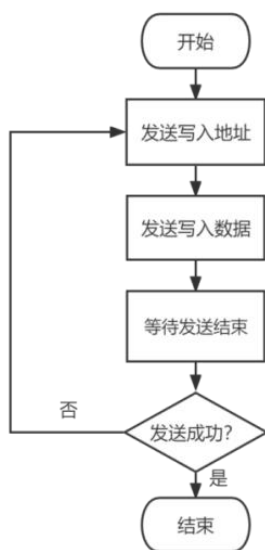


图 3.5 写入寄存器函数流程图

其中，发送操作使用 `HAL_I2C_Master_Transmit_IT` 完成，等待发送结束使用函数 `HAL_I2C_GetState` 判断当前状态是否为 `HAL_I2C_STATE_READY`，是否发送成功通过 `HAL_I2C_GetError` 返回错误代码来判断，为 `HAL_I2C_ERROR_NONE` 则表示发送成功，其它则失败重发。

(2) 读取操作实现。实现函数 `hearttrate_sensor_comm_read(uint8_t addr, uint8_t *data)` 来完成寄存器，其中 `addr` 是需要读取的寄存器地址，`data` 指向存储读取数据的接收缓冲区。读取操作流程如图 3.6 所示。

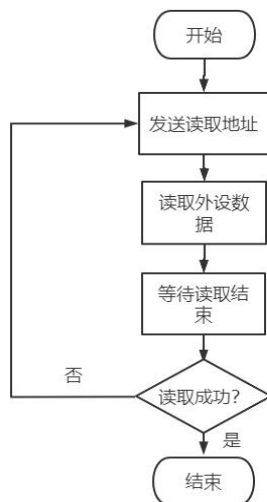


图 3.6 读取寄存器流程图

其中，发送操作使用 HAL_I2C_Master_Transmit_IT 完成，读取操作使用 HAL_I2C_Master_Receive_IT 完成，等待读取结束使用函数 HAL_I2C_GetState 判断当前 I2C 状态是否为 HAL_I2C_STATE_READY，是否读取成功通过 HAL_I2C_GetError 返回错误来判断，结果为 HAL_I2C_ERROR_NONE 则表示接收成功，其它则失败重新读取。

3.3 心率传感器驱动设计与实现

心率传感器内置有数据缓冲区，经过数字处理之后的 PPG 数据可以保存在自己的数据缓冲区之中。心率传感器有两种方式来获取数据，分别是轮询模式和中断模式。轮询模式是由处理器主动发起询问外设的收集信息方式，外部设备属于被动接收，这种模式缺点很明显，对处理器资源的占用极大。中断方式则本质上属于一种“硬件轮询”，在这种模式下，数据的 I/O 是由外设来主动发起的，当数据缓冲区准备好数据时向处理器发出中断信号，此时主控制器保存当前工作现场，再进入中断处理程序处理。这种方式不仅使得处理器的利用率有了大幅的提高，也实现了多个外部设备的并行 I/O 操作。心率传感器在中断模式下总体工作流程如图 3.1。

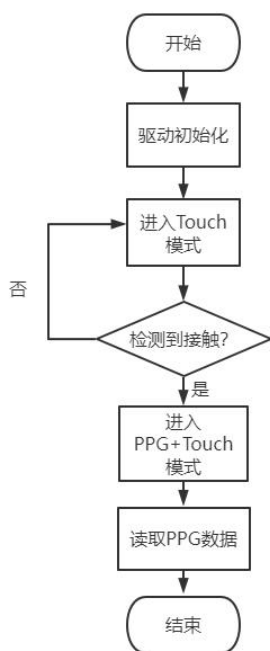


图 3.7 心率传感器中断模式下工作流程图

3.3.1 驱动初始化

对于 PAH8011ET 的初始化，主要就是根据数据手册设置需要的寄存器，再加以验证是否设置正确。初始化流程如下：

（1）验证通信正常

首先要对主控芯片与心率传感器之间的通信进行验证，否则之后所有的操作都是无意义的，该操作通过读取产品 ID 来完成。产品 ID 位于 bank1 的地址为 0x00, 0x01，为两个只读寄存器，如果读取数据为 0x8011 则说明当前通信正常。

（2）曝光时间与帧周期设置

为了更好的适应不同人群的肤色、肌肉等，心率传感器可以自行调整曝光时间、LED DAC 电流大小、帧周期以保证光线在最佳亮度范围。同时也可以设置自动调整的范围，这些寄存器位于 bank 0 之中，由于数量众多且并不重要，不再一一具体介绍具体实现。这里均打开自动调整，范围根据官方推荐默认值设置。

（3）中断模式选择

心率传感器的中断模式根据位于 bank 1 地址为 0x36 的 Interrupt Mode Selection Register 来控制，该寄存器描述如表 3.2。

表 3.2 中断模式选择寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0						INT_Mode_Sel[1:0]	

通过最低两位 INT_Mode_Sel 来选择中断模式，手环硬件设计中只使用一个 INT1 引脚作为中断源，这里设置 INT_Mode_Sel 为 01，此时表示禁用 INT2 中断源，INT1 中断源同时被缓冲区数据准备好中断、缓冲区数据上溢中断、缓冲区数据下溢中断、触摸中断所共享。

（4）使能中断

心率传感器的 INT1 中断源与主控芯片的 PB4 引脚相连，通过 HAL 库提供的函数 HAL_NVIC_EnableIRQ 在系统中使能该引脚的中断。完成此步之后，心率传感器的初始化操作就此结束。

3.3.2 模式切换实现

心率传感器具有的七种工作模式如表 3.3 所示。

表 3.3 心率传感器七种工作模式描述

工作模式	描述
OFF	没有电源供应，所有电压轨和时钟都是选通的。
RESET	通电后，各个功能块的所有轨道均应在内部设置闸门。
READY	可以将此模式视为加载初始设置的模式。
Power-down	低功耗模式
PPG	该模式使用 I2C 或 SPI 从就绪模式获取心率 PPG 原始数据。
Touch	此模式使用 I2C 或 SPI 从就绪模式进行触摸检测。

因为要节约能量，心率传感器完成初始化之后，还处于低功耗模式无法工作，需要合理的控制心率模式的转换来控制传感器的工作。心率传感器工作流程之中，模式切换主要就是低功耗模、PPG 模式、PPG+Touch 模式之间的切换。单独的 PPG 模式下需要关闭 Touch 模式，同理 Touch 模式下也要关闭 PPG 模式，而 PPG+Touch 模式下则同时进入 PPG 模式和 Touch 模式。PPG 模式和 Touch 模式两种模式的区别主要在于是否使能获取相应数据的通道。

(1) 低功耗模式的进入和退出

心率传感器控制开启和关闭低功耗模式由 Power Down Mode Register 来决定，地址为 bank4 的 0x69，这是一个八位的寄存器，其中只有 bit[0]位控制低功耗模式，拉低表示退出低功耗模式，拉高表示进入低功耗模式，其它位标记为 0。寄存器描述如表 3.4 所示。

表 3.4 低功耗模式寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0							PD_Mode

(2) PPG Mode 的进入和退出

PPG 模式与 Touch 模式二者设置寄存器序列的区别主要在于设置上报数据的频率不同以及是否使能相应的通道。在 PPG Mode 下，一般上报 Touch 数据频率为 1Hz，PPG 数据的频率为 20Hz。心率传感器中一共有 4 个通道用来上报数据，可从通道

A,B,C 中选取一个来上报 PPG 数据，通道 T 则用来上报 touch 数据。各个 Channel 上报数据的频率由 Golden_Report_Divier, ChA_Report_Divider, ChB_Report_Divider, ChC_Report_Divider 以及 ChT_Report_Divider 决定。Golden_Report_Divier 由 bank2 的 0x26, 0x27 两位一起组成，一共取 13 位数据，每一位数值表示为 31.25us。其他 Channel 的频率由全局上报频率做除法得到。如果要设置通道 A,B,C 为 20Hz，通道 T 为 1Hz，具体方法如下：

Golden Report Frequency=Golden Report Divider [12: 0]* 31.25us =1600*31.25us = 50000us=20Hz

Channel A Frequency= Golden Report Frequency/CHA Report Divider[7:0]=20/1=20Hz

Channel B Frequency=Golden Report Frequency/CHB Report Divider[7:0]=20/1=20Hz

Channel C Frequency=Golden Report Frequency/CHC Report Divider[7:0]=20/1=20Hz

Channel T Frequency=Golden Report Frequency/CHT Report Divider[7:0] =20/20=1Hz

以上为 PPG Mode 下的寄存器设置示例。对于 PPG Mode 的开启和关闭，主要是对相应通道的使能与关闭来完成，有通道 A、B、C 分别有自己的控制寄存器，地址分别为 bank2 的 0x21、0x23、0x25。将当前使用的 Channel 使能完成就相当于进入了 PPG Mode，每个寄存器使用 bit[0]来表示是否使能对应通道，拉低表示为禁用，拉高表示为使能。在开启和关闭时，首先要读取当前的寄存器值，再通过位操作对 bit[0]进行设置，写回到寄存器中，不能改变其它位的状态。以通道 A 为例，寄存器描述如表 3.5。

表 3.5 通道 A 功能寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	1	BS_EnH	0	1	On_Frame_Group[1:0]		ChA_EnH

(3) Touch Mode 的进入与退出

关于寄存器序列的设置，在 Touch Mode 下则设置 ChT_Report_Divider 寄存器为 0x01，设置当前使用的 Channel 控制寄存器为 0x14 即可。这种设置下，上报 touch 数据的频率为 20Hz，上报 PPG 数据的频率为 1Hz。

Touch Mode 的关闭和开启由 Touch Channel Enable Register 来控制，其中 bit[0]来决定是否开启 Touch Mode，拉低为关闭，拉高为开启。Bit[1]来决定是否开启 Touch

Fast Mode，拉低为关闭，拉高为开启。在 Touch Fast Mode 下接触上报的频率与全局上报频率相同，关闭时则为通道 T 的上报频率。寄存器描述如表 3.6。

表 3.6 通道 T 使能寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0						Touch_Fast	Touch_Det

(4) PPG+Touch Mode 的进入与退出

在 PPG + Touch 模式下,无论是 PPG 数据还是 Touch 数据上报的频率都为 20Hz,此时 Golden_Report_Divier 不变,设置当前使用的 PPG Channel 寄存器以及 Touch Channel 寄存器为 0x01,并且同时使能当前使用的 PPG 通道以 Touch 通道即可进入 PPG+Touch 模式。此时关闭 PPG 模式即可进入 Touch 模式,同理关闭 Touch 模式即可转换到 PPG 模式

3.3.3 读取数据流程设计

设计编写 heartrate_task 函数执行任务的函数,用来执行读取数据的流程,该函数作为心率传感器 INT1 中断源发生时的回调函数。函数流程如图 3.8。

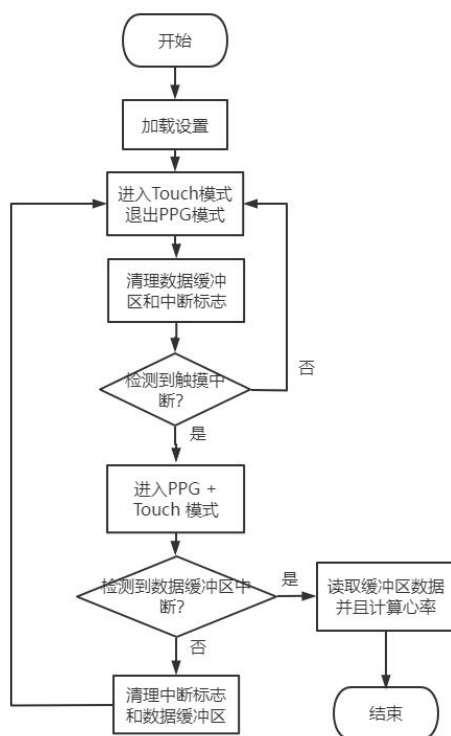


图 3.8 读取 PPG 数据流程图

整体流程如下：

(1) 进入 Touch Mode 进行触摸检测。

(2) 读取中断状态，中断状态寄存器地址为 bank1 的 0x1B。该寄存器高 4 位未使用，低 4 位中 bit[0]用来表示数据缓冲区数据是否准备好，bit[1]为接触检测中断，bit[2]，bit[3]分别表示数据上溢和下溢，情况通常发生于处理函数在发生中断后被调用过晚。中断状态寄存器描述如表 3.7。

表 3.7 中断状态寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0				FIFO_UND	FIFO_OVF	Touch_Det	FIFO_DR

(3) 如果未发生触摸中断，则继续等待触摸中断，否则进入 PPG+Touch 模式读取数据缓冲区 FIFO 中断。

(4) 进入 PPG +Touch 模式下，如果未发生数据缓冲区中断，则返回 Touch 模式继续等待触摸中断。如果发生了数据准备好中断则要进行数据的读写。

(5) PPG 数据存储于 bank3 之中，进行读写首先要知道具体数据的个数，这个数量由 FIFO Data Number Register 保存，该寄存器位于 bank2 的 0x25 和 0x26 之中，对其进行读取可以获得本次读取的数据数量。这是一个 9 位数据，bit[0-7]存储在 0x25 中，bit[8]存储在 0x26 中。寄存器描述如表 3.8。

表 3.8 数据缓冲区数量寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_RPTNUM_SYNC[7:0]							
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FIFO_RPTNUM_SYNC[8]							

(1) 读取每个 PPG 数据是 32 位整数，心率传感器的 bank3 位专门存储 PPG 数据的缓冲区，从 bank3 的 0x00 开始，读取本次数据个数*4 个字节数据到数据缓冲区中就是本次读取获得的 PPG 数据，之后可以利用数据进行心率计算。

(7) 任务执行完毕之后，要及时执行清理中断标识和数据缓冲区操作。

3.3.4 心率数据存储

完成原始 PPG 数据的获取和心率计算之后即可获取心率数据，获得的心率数据暂时存储到结构体 `daily_body_data` 之中，下文会获取的步数数据也是存储到这里。结构体设计如表 3.9。

表 3.9 `daily_body_data` 结构体

字段	描述
Year, month, day	当前日期
Heartrate_max	最高心率
Heartrate_min	最低心率
Current_heartrate	当前最近心率
Current_stepcounts	当前步数
Heartrate_history	历史心率数据

每次读取到新的心率之后，分别更新 `current_heartrate` 字段，并且判断最新获取的心率是否是最大或者最小心率。`Heartrate_history` 是一个长度为 500 的无符号的八位整数数组，存储当天历史心率。当其他应用需要心率数据时从该结构体中直接读取。该结构体会在每天结束或者关机之前写入到外设串行 Flash 之中长期保存。

3.4 本章小结

本章主要是对心率检测功能的实现。首先给出了如何利用原始 PPG 数据获取心率的方法，重点做了 PPG 信号的降噪处理。之后详细介绍了主控与传感器芯片之间的 I2C 通信，最后根据传感器数据手册完成了传感器的初始化、模式切换、PPG 数据获取流程的实现。

4 基于加速度传感器的计步算法实现

加速度传感器是手环项目之中必不可少的存在,利用加速度传感器可以实现步数统计、运动检测、抬腕亮屏、方向指示等等实用功能,在实现 PPG 信号滤波时,也用到了加速度值作为参考信号。加速度传感器与主控之间 I2C 传输流程与心率传感器只有设备地址不同,其他完全一致,步数数据存储方式也与心率数据存储基本相同,故本章不再赘述加速度传感器具体数据传输方式以及数据存储方式。本章主要给出三轴加速度数据计步原理,实现一个计步算法,以及完成加速度传感器的初始化、数据获取流程。

4.1 计步原理及计步算法实现

4.1.1 三轴加速度计步原理

人体在走路或者跑步之中,手臂前后摆动会产生在水平方向加速度上的周期性变化,脚步的抬起和下落会产生垂直方向加速度的周期性变化。在跨步伸脚时,水平方向加速度增加,收脚时水平方向加速度减小;抬脚时垂直方向加速度增加,落脚时垂直方向加速度减小^[43]。将三轴加速度数据经过图表可视化之后如图 4.1。

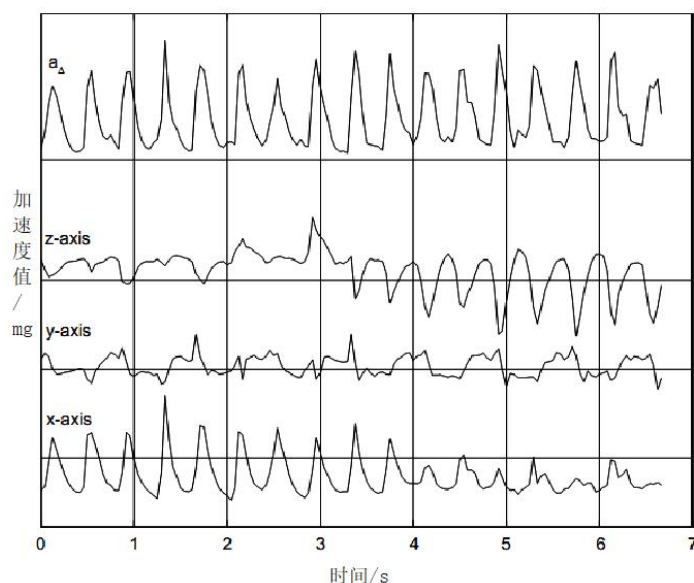


图 4.1 行走时三轴加速度值变化

如图 4.1 可以看出,水平和垂直方向上的加速度都呈现周期性变化,每一个周期都可以认为是用户走了一步,对于周期数量的统计可以通过统计峰值点数量来实现。但是考虑到手环设备用户可能放置方向的不同,可能会出现 X 轴或者 Z 轴统计的加速度值并不是水平或者垂直方向上的加速度,故采用单一轴作为参考是不合理的。考虑将三个加速度轴的模值变化曲线作为参考,对其做峰值数量统计是合理的。

4.1.2 计步算法实现

设计 StatusOfPoint 和 StatusOfGlobal 两个结构体来分别表示采集到的点状态和当前全局状态。

StatusOfPoint 结构体设计如表 4.1。

表 4.1 StatusOfPoint 结构体设计

字段	描述
x,y,z	x,y,z 三轴加速度值
sensorValue	加速度模值
IsUp	该点相较前一点上升趋势
continueUpCount	到该点持续上升次数

StatusOfGlobal 结构体设计如表 4.2。

表 4.2 StatusOfGlobal 结构体设计

字段	描述
ContinueUpCount	当前全局持续上升次数
PeekValue	当前最新波峰值
ValleyValue	当前最新波谷值
TimeOfLastPeek	最新波峰点时刻
StepCount	本次统计到的步数
ThresholdArray	存储波峰波谷差值的数组

下面开始详细介绍算法流程。

(1) 静止点判断

算法开始时传入三轴加速度值,将本次三轴加速度值与上一点三轴加速度值进

行比较,如果三轴都与上次差别很小,则可以认为是静止状态直接返回。否则计算本次三轴加速度的模值,进行波峰、波谷的判断。

(2) 升降趋势判断

将本次获得的模值与上次记录的模值相比较,如果本次大于上次,则表示还在上升,持续上升次数加一,如果本次小于上次,则本次记录为下降,将当前全局持续上升次数赋值给上一点持续上升次数,同时当前全局上升次数清零。

(3) 波峰波谷点判断

在本次点呈下降趋势、上一点为上升趋势、上一点持续上升次数大于 5 或者上次点模值超过设定阈值三个条件同时满足时,可以记录上次点为波峰点。相反,如果上次点为下降趋势,本次点为上升趋势,则记录上一点为波谷点。记录波谷值是为了和波峰值作差比较,排除短时间小波动的波峰值。

(4) 高频噪声消除

如果没有检测到波峰就什么都不做。检测到波峰的话,首先对当前时间和上次波峰点时间作差进行判断,如果差值小于 250ms,则认为本次波峰为连续抖动波峰,则忽略,否则继续判断当前波峰波谷的差值,如果大于阈值的话继续下一步防误判。

(5) 防误判计步

防误判操作主要针对用户日常生活中手臂抬起、放下、晃动等动作会造成计步的误计设计。此处会判断当前波峰时间和上次波峰时间的差值,如果超过 3 秒,则此前累积的步数清零,之后从 1 开始计算,如果小于 3 秒,则将全局状态中的 stepCount 加一,如果此时 stepCount 大于 5 则最终记录次数加一,小于 5 则清零。此处主要表示只有在 3 秒内连续累积了 5 步,之后还在运动的步数才有效,3 秒内运动不超过 5 步的记录都将失效。

(6) 阈值更新

阈值的计算主要依靠历次波峰波谷差值的平均值再乘一个系数来获得,这个阈值在不同场景下大小会有区别,比如在慢走场景下阈值会小,在跑步或者剧烈运动下阈值会大。系数的设定是经验值的累积,需要经过多次测试来设置系数,多判断了就尝试提高系数,少判断了就尝试降低系数。计步算法伪代码如下:

算法3 计步算法

输入:三轴加速度值、当前系统时间

输出:本次累加计步

```
1. if(abs(preX-x)>100 || (preY-y)>100||(preZ-z)>100) then //判断当前是否与上次差距明显
2.   curVal = sqrt(x*x+y*y+z*z)
3.   if(curVal > oldVal) then
4.     isUp = true; continueUpCount ++;
5.   else
6.     isUp=false; continueUpCount=0;
7.   end if
8.   if(isUp&&!lastIsUp&&(continueUpCount>4||preVal>30000)) then //波峰判断
9.     peekVal = preVal; isPeek = true; //更新波峰
10.  else if(!isUp&&lastIsUp) //波谷判断
11.    valleyVal = preVal //更新波谷
12.  end if
13.  lastIsUp=isUp //更新上升趋势
14.  if(isPeek) then
15.    if(peekVal - valleyVal > threshold && curSysTime - prePeekTime > 250
&&curSystem-time-prePeekime < 3000 ) then //波峰合法性检测
16.      tmpStepCount++; //临时计步增加
17.      if(tmpStepCount > 5) then
18.        ret = 1 //累计步数超过 5 步，本次算一步
19.      else if(tmpStepCount == 5)
20.        ret = 5 //累积步数等于 5 步，本次累积 5 步
21.      else
22.        ret = 0 //3 秒内步数不超过 5 步，累积到临时计步。
23.        prePeekTime = curSysTime //更新上一次波峰时间
24.      end if
25.    else //波峰不合法，清空临时计步
26.      tmpStepCount=0
27.    endif
28.    update threshold; //根据本次波峰波谷差值更新阈值
```

```
29.   endif
30.   endif
31.   return ret
```

4.2 加速度传感器驱动设计与实现

SC7A20 的工作流程相对简单，只需要按照数据手册对传感器完成一些必要的配置（比如数据量程、数据更新率、中断阈值、工作模式等）完成初始化，再从相应的寄存器中读取获得三轴加速度值即可。

获取三轴加速度数据可以通过查询状态寄存器完成，当有 X、Y、Z 轴有数据更新时，对应状态标识寄存器 STATUS_REG(27h)的 ZYXDA、ZDA、YDA、XDA 位会自动置 1，此时可直接读取三轴数据寄存器 OUT_X_L(28H)、OUT_X_H(29H)、OUT_Y_L(2AH)、OUT_Y_H(2BH)、OUT_Z_L(2CH)、OUT_Z_H(2DH)，以上寄存器全部读取操作完成后，状态标识寄存器 STATUS_REG 的 ZYXDA、ZDA、YDA、XDA 位才能自动清 0。若在数据读取前，数据发生再次更新，上次数据会被自动覆盖，且此时 STATUS_REG 的 ZYXOR、ZOR、YOR、XOR 会自动置 1，而读完三轴数据寄存器内容后会自动清 0。

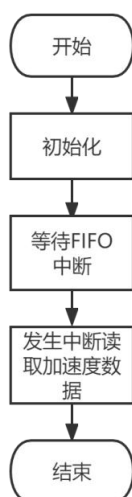


图 4.2 加速度传感器工作流程图

也可以配置相应的中断寄存器，FIFO 寄存器，设置 FIFO 通道中数据的阈值来触发中断。传感器内置的 FIFO 数据达到设定阈值时会触发 INT 脚的极性（高电平有

效或者低电平有效），此时可以读取三轴数据寄存器 OUT_X_L(28H)、OUT_X_H(29H)、OUT_Y_L(2AH)、OUT_Y_H(2BH)、OUT_Z_L(2CH)、OUT_Z_H(2DH)，来获取三轴加速度值。为了节省 CPU 时间，采用中断模式，中断模式下加速度传感器工作流程如图 4.2 所示。

4.2.1 加速度传感器驱动初始化

加速度传感器的初始化主要完成数据量程，数据更新速率，FIFO 模式，传感器中断配置。

(1) 加速度传感器量程配置

加速度传感器量程从 $\pm 2\text{G}$ 、 $\pm 4\text{G}$ 、 $\pm 8\text{G}$ 、 $\pm 16\text{G}$ 可选，默认为 $\pm 2\text{G}$ 。人在行走或者跑步时手臂摆动产生的加速度一般不会超过 $\pm 2\text{G}$ 的区间（即使超出对于步数统计应用来说也无甚影响），选取 $\pm 2\text{G}$ 的区间也可以获得更好的数据精度。量程的设置放在 CTRL_REG4（23h）寄存器中，寄存器描述如表 4.3。

表 4.3 控制寄存器 4

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BDU	BLE	FS1	FS0	DLPF_EN	ST1	ST0	SIM

其中 FS1、FS0 位共同控制量程选择，将其设置为 00 表示选择 $\pm 2\text{G}$ 量程，其它位使用默认值。

(2) 加速度传感器数据更新率与低功耗模式配置

SC7A20 提供了从 1Hz 到 400Hz 之间丰富的数据更新速率可供选择。三轴加速度数据在手环中主要用于步数统计，以及作为 PPG 信号自适应滤波中的参考信号。其中 PPG 信号频率为 20Hz。作为它的滤波参考信号，加速度信号的频率必须大于等于 20Hz。对于步数统计需求，人体跑步频率最快不超过 5Hz，根据香农采样定理，采样频率必须大于等于被采样信号频率的二倍，该需求要求加速度信号采样频率大于等于 10Hz。故加速度传感器的数据更新率必须大于 20Hz，但采样频率并不是越高越好，更高的采样频率会有更大的功耗，而且由于步数统计的需求，加速度传感器必须一直工作，有更高的能耗要求，故选取 25Hz 作为加速度传感器的数据更新率。

数据率和低功耗模式的使能由 CTRL_REG1 (20h) 来控制。寄存器描述如表 4.4 所示。

表 4.4 控制寄存器 1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ODR3	ODR2	ODR1	ODR0	LPen	Zen	Yen	Xen

其中 ODR3、ODR2、ODR1、ODR0 用来控制数据更新率，LPen 控制低功耗模式，Xen、Yen、Zen 分别表示 X，Y，Z 轴的使能。对该寄存器写入 0x3F，表示使用低功耗 25Hz 模式，并且使能三个加速度轴。

(3) 加速度传感器 FIFO 配置

FIFO_CFG (2Eh) 寄存器控制数据通路的模式以及中断引脚的选择和中断的触发条件，寄存器具体如表 4.5 所示。

表 4.5 FIFO 控制寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FM1	FM0	TR	FTH4	FTH3	FTH2	FTH1	FTH0

通过 FM1 和 FM0 两位选择 FIFO 模式，00 为 By-Pass 旁路模式，即不使用 FIFO 功能；01 位 FIFO 模式，缓存满未及时读取时，新数据丢弃；10 为 Stream 模式，缓存满后，最早数据丢弃，加入新数据。低五位是 FIFO 功能 WTM 阈值设置，当达到设置阈值时，FIFO 状态寄存器相应状态位会置为 1。将这个阈值设置为 25（此时在 25Hz 数据更新率下，约 1 秒发生一次），故该寄存器写入 0x59。

(4) 加速度传感器 FIFO 中断配置

最后需要完成 FIFO WTM 中断在 INT1 引脚上的使能，在控制寄存器 CTRL_REG3 (22h) 上完成，该寄存器如表 4.6 所示。

表 4.6 控制寄存器 3

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
II_CLICK	II_AOI1	II_AOI2	II_DRDY1	II_DRDY2	II_WTM	II_OVERRUNN	-

其中 bit[2]表示将 FIFO WTM 中断使能在 INT1 引脚上，将该位置为 1。完成该步设置之后，加速度传感器的初始化就此完成。

4.2.2 三轴加速度数据获取实现

(1) 中断标志读取

三轴加速度数据的获取在发生 FIFO WTM 中断时执行, 此时表示 FIFO 缓冲区的三轴加速度数据已经达到之前预设的阈值 (25 个)。由于获得的数据数量可能会超过 25, 首先要获取本次读取到的数据数量, 这个值存储在 FIFO_SRC(2Fh) 寄存器之中, 该寄存器描述如表 4.7 所示。

表 4.7 FIFO 控制寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WTM	OVER	EMPTY	FSS4	FSS3	FSS2	FSS1	FSS0

当 FIFO 中的数据个数超过设定阈值时, WTM 位置 1。当 FIFO 中的数据溢出时, OVER 位置 1。当 FIFO 中无数据时, EMPTY 位置 1。低五位则存储当前尚未读取的三轴加速度数据组数, 最多存储 32 组数据。要先判断 OVER 位是否被置为 1, 如果被置 1 则读取 32 组, 如果没有则将寄存器值与 0x1f 进行与操作, 获得读取数据数量。

(2) 加速度数据读取

采样后的加速度数据输出到寄存器 OUT_X_H (29H)、OUT_X_L(28H)、OUT_Y_H(2BH)、OUT_Y_L(2AH)、OUT_Z_H(2DH)、OUT_Z_L(2CH), 输出数据以二进制格式输出, 最高位为符号位, 故读取三轴数据需存储到无符号字符型变量后再做数据合并 (高低位合并), 然后再强制转换为有符号整型数据输出。三轴寄存器以 16 位数据输出, 默认: 最高位在左, 最低位在右。当读取完一组数据之后, 寄存器中存储的数据会自动切换到下一组, 故取多组数据时对寄存器循环读到到定义好的加速度数据数组中即可。

加速度数据如何来表示实际加速度大小, 主要是通过寄存 CTRL_REG4(23H)的 FS1、FS0 位来设置加速度传感器的量程, 不同的量程选择和不同工作模式组合对应的不同的灵敏度 (单位: mg/digit) 和量程。当选取 $\pm 2G$ 作为加速度传感器的量程时, 在低功耗模式下, 对应的灵敏度为 1mg/digit。例如, 当某轴数据输出为 -1000 时, 将此数据乘以对应的灵敏度 1mg/digit, 那么此时的重力加速度为 -1000mg, 也就是一

个重力加速度大小。

4.3 本章小结

本章完成了手环计步功能的实现。首先，利用三轴加速度进行步数统计的原理，实现了一个计步算法。其次，在驱动方面根据加速度传感器的数据手册以及自身项目实际需要对传感器进行了初始化。最后，完成了三轴加速度值获取流程的编写。

5 低功耗蓝牙通信实现

受限于屏幕尺寸大小和性能有限，单靠手环自身并不能很好的展现和处理信息。自从 BLE 协议问世以来，越来越多的智能设备都与智能手机相结合，智能设备负责采集信息，智能手机负责展示和处理信息。智能手环与智能手机之间的通信，对速率并不敏感，而对功耗及其敏感，使用 BLE 技术作为通信手段再适合不过。本章主要介绍 BLE 协议栈、设计手环端和 Android 智能手机端之间的 BLE 通信过程，并且在双端分别实现；设计基于广播形式的校园特色功能并且实现。

5.1 BLE 协议栈

协议主要是定义了数据格式、通信方式、数据含义、沟通时序的标准。BLE 协议就是这样一种无线通信标准，定义了开发低功耗蓝牙产品软硬件层面的标准和要求。BLE 协议栈是实现 BLE 协议的基础，任何 BLE 产品都要在 BLE 协议栈之上开发自己的应用^[44]。BLE 协议栈主要包含了主机、控制器以及应用程序，整体架构如图 5.1。



图 5.1 BLE 协议栈

(1) BLE 协议栈 GAP 层

GAP 层规定了不同设备之间如何交互以及广播。它使一个设备可以被其他设备发现。GAP 层也定义了 BLE 通信中设备可以扮演的四种角色：Broadcaster（广播者）、Observer（观察者）、Central（中心设备）、Peripheral（外围设备）。

GAP 层规定了两种设备之间的交互方式，分别是基于广播的方式和基于连接的方式。广播方式不需要设备之间连接，只需要广播者发出自己的数据就可以。这种方式多用于传感器的信息采集，广播者将自己收集到的信息发送给观察者。这种方式下，一个广播者发出信息可以被多个观察者同时接收。基于连接的方式，一般是外围设备通过发送广播让中心设备可以扫描到自己，两者之间建立连接之后的数据通信即可。这种方式下，一个外围设备只能被一个中心设备连接，而一个中心设备可以连接多个外围设备。

(2) BLE 协议栈 GATT 层

如果说 GAP 层规范了设备之间的广播和连接，那设备连接之后具体的数据交换则是在 GATT 层完成的。ATT 层定义数据封装对外暴露为“属性”，“属性”是一种结构简单，数据长度很短的结构，专门为低功耗蓝牙使用。GATT 则在 ATT 的基础上更进一步对“属性”进行逻辑封装，定义数据的交互方式和含义。

GATT 的层次结构从上到下依次为：Profile、Service、Characteristic。Profile 可以被认为是设计者预先设计的应用场景，是 Service 的集合。Service 是 Characteristic 的集合，包含有一个或者多个 Characteristic。Characteristic 是最基本的数据单位，也是在 BLE 通信中直接进行操作（读、写或者订阅通知）的单位。

对于特征值的操作，主要有读、写、接收通知三种。读写很好理解，就是从特征值之中读取和写入数据。接收通知的方式有两种，一种是 notify，一种是 indicate，它们之间的区别主要在于 notify 通知的数据可能会丢失，而 indicate 一定会发送成功。Indicate 具有 ACK 机制，如果没有收到客户端设备的确认，会一直重传数据直到成功。而 notify 没有失败重传机制。但 notify 具有速度快的优点。通常在外围设备需要不断发送数据给中心设备的时候使用 notify 形式。而发送一条短小但是重要的数据时使用 indicate。

5.2 基于连接的 BLE 通信过程

BLE 通信在应用层的程序设计主要分为建立连接与数据通信两个部分。

建立连接的过程，首先是由手环设备主动发起可连接广播使其被其他设备可见，智能手机开启广播扫描之后可以发现手环设备，智能手机可以向手环设备请求扫描回复，之后双方可以进行连接的建立。这种连接是独占的，建立连接之后手环会立即停止可连接广播。建立连接进行通信也是设备进行双向通信的唯一方式。建立 BLE 连接过程流程如图 5.2。

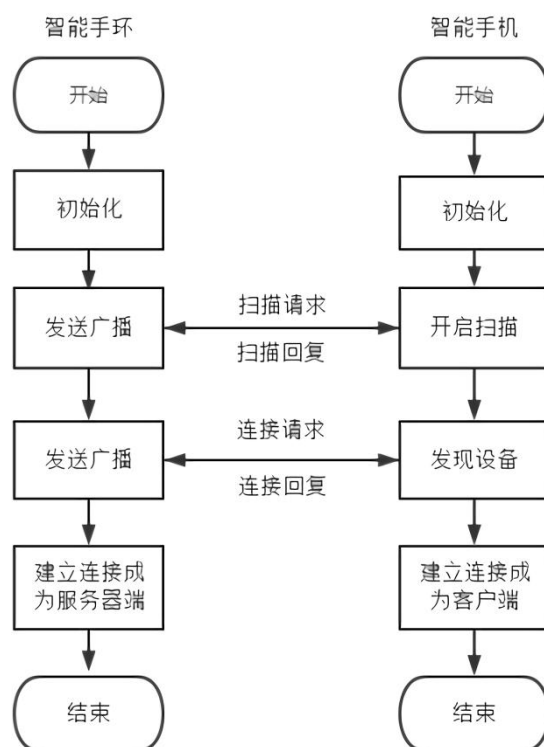


图 5.2 BLE 连接建立过程

手环与手机的通信虽然有多种数据需要传输（心率、步数、天气、运动计划等），但本文设计中只用一个服务以及一个特征值来进行通信。这个特征值之中存储智能手机端下发的指令以及手环端回复的数据。除了获取手环采集数据之外，还要通过智能手机下发指令完成手环的设置。所有通信都采取客户端（智能手机）请求、服务器（智能手环）响应的方式。图 5.3 展现通信采用的 C/S 架构方式。

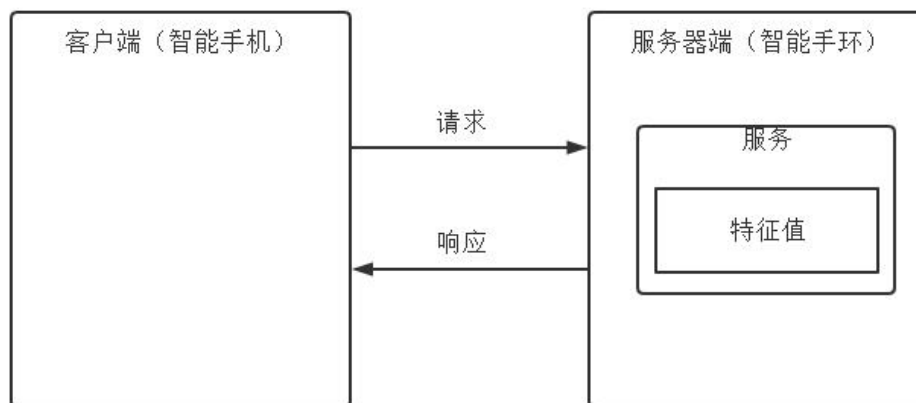


图 5.3 BLE C/S 架构

请求指令和响应数据都通过读写特征值来完成，数据包格式设计如表 5.1。

表 5.1 数据包格式设计

Business Cmd	Data Len	Data	8 bits CRC
0	1 - 4	5 - n-1	n

假设一个数据包为 n 个字节，则第一个字节指示具体是什么业务的请求\回复，1-4 位一共 4 个字节表示回复数据的长度，随之 5 - $n-1$ 个字节表示具体的请求\回复数据，最后一个字节是 8 位 CRC 校验。

5.3 手环端低功耗蓝牙通信实现

智能手环主控芯片 STM32WB55CGU6 是 ST 公司专门为了 BLE 应用推出的 SOC，ST 公司也为其专门打造了强大的低功耗蓝牙库。本节主要使用 ST 公司提供的库函数完成手环端低功耗蓝牙通信的实现。

5.3.1 GATT 服务器初始化

对 GATT 连接中服务器方的初始化，就是对于服务以及特征值的添加以及回调函数的注册。

添加服务利用 `aci_gatt_add_service` 函数，自定义服务 128-bit UUID 为 00000000-000F-000E-000F-000000000000，该函数返回指向被添加服务句柄的指针，

该句柄用于在应用层程序中定位服务。

添加特征值利用 `aci_gatt_add_char` 函数，要指定被添加进哪个服务，UUID（自定义特征值 128-bit UUID 为 00000000-000F-000E-000E-000000000000）以及声明该特征值可以被读写。该函数返回指向该特征值句柄的指针，该句柄用于定位应用层程序中的特征值，之后的通信就是通过读写该特征值进行的。

回调函数的注册使用 `SVCCTL_RegisterSvcHandler` 函数，当收到 GATT 事件时，会在回调函数中收到通知，通知包括连接句柄，被写入特征值句柄，写入数据长度，写入数据等等。在回调函数之中处理具体业务逻辑。

5.3.2 广播与连接

在建立一个 GATT 连接中，需要外围设备发送可连接广播，中心设备扫描发现之后建立连接。调用 `aci_gap_set_discoverable` 函数，该函数使设备可以进入普通可被发现状态，此时手环可以被智能手机扫描发现，并且可以通过该函数设置广播设备名，最长、最短时间间隔，可发现状态将持续到调用 `aci_gap_set_non_discoverable` 函数为止。广播数据的发送调用 `aci_gap_update_adv_data` 函数，传入广播数据缓冲区指针和数据长度，即可自动开始广播。

5.3.3 数据接收与发送

当 GATT 服务器有任何特征值被改写时，初始化中设置的回调函数 `TTManagement_Event_Handler` 会被调用。其返回的数据包含有发生事件的连接句柄、特征值句柄、数据长度、以及具体数据，判断该特征值句柄是先前设置交互的特征值之后，从中可以获得智能手机发送的数据。

解析手机传输的数据，再根据具体业务做出对应的响应，外围设备向中心设备发送数据使用 `aci_gatt_update_char_value` 函数完成，需要指定服务句柄，特征值句柄，以及发送数据长度以及发送缓冲区指针。需要注意的是，BLE 每次收发数据长度不超过 20 字节，如果超过 20 字节需要分多次接收或者发送。

手环端蓝牙程序整体流程如图 5.4。

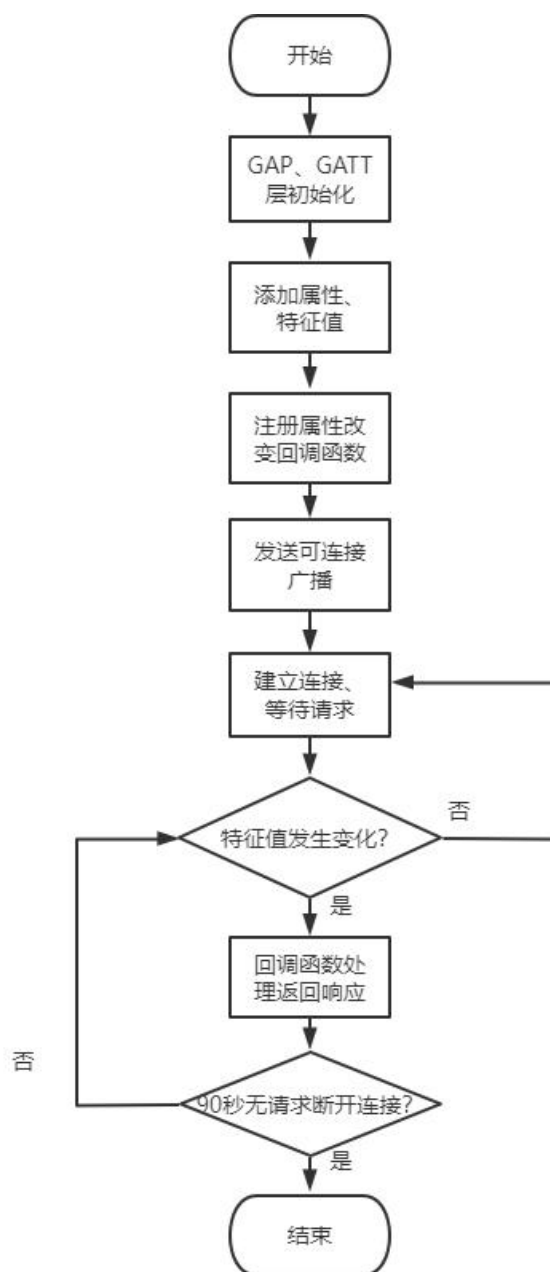


图 5.4 手环端蓝牙程序流程

5.4 Android 端低功耗蓝牙通信实现

在 Android4.3 版本正式支持低功耗蓝牙之后, 安卓官方也提供了一系列低功耗蓝牙相关的类来完成 BLE 通信的编写。下表 5.2 是 Android 官方提供的所有 BLE 编程相关类, 层级从上到下依次递减, 呈现包含关系。

表 5.2 低功耗蓝牙相关类

类名	描述
BluetoothManager	安卓手机蓝牙管理器
BluetoothAdapter	基本蓝牙操作，如扫描、连接等
BluetoothDevice	代表一个外围蓝牙设备
BluetoothGatt	提供 GATT 协议基本功能，获得服务、特征等
BluetoothGattService	GATT 服务器提供的服务
BluetoothGattCharacteristic	GATT 服务器提供的特征值，通过读写它来读写数据

下面具体介绍使用这些类开发 Android BLE 通信的流程。

5.4.1 声明蓝牙使用权限

声明蓝牙权限，需要在 AndroidManifest.xml 中声明需要使用的权限，需要添加的配置如下

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

需要注意的是，如果 Android 版本在 6.0 以上，开启蓝牙的同时还需要模糊位置权限。由于位置权限是敏感权限，会在用户打开应用后询问是否开启，继续添加配置如下：

```
<uses-permission  
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

5.4.2 初始化工作

要使用低功耗蓝牙，首先需要确认当前设备是否支持 BLE，如果支持 BLE，再确定当前是否打开蓝牙开关，如果没有可以利用 Android Intent 组件来引导用户打开蓝牙。这些操作都可通过 BluetoothAdapter 类来实现。BluetoothAdapter 的获取可以通过 BluetoothManager 来完成。

5.4.3 扫描与连接

GATT 连接通过外围设备发送广播，中心设备扫描来建立。扫描可以通过调用蓝牙适配器的 `startLeScan` 方法开启专门的低功耗蓝牙扫描，该方法需要传入扫描设备成功之后的回调函数，回调函数之中会提供一个 `BluetoothDevice` 对象来代表一个扫描到的蓝牙设备，以及蓝牙信号指标和蓝牙广播数据。可以在这里把扫描到的设备对象保存起来，以便过滤重复出现的设备对象。调用 `BluetoothAdaptor` 的 `stopLeScan` 方法可以停止扫描，蓝牙扫描比较消耗电量，扫描一段时间之后必须关闭。

选择需要连接的设备，可以通过 `BluetoothDevice` 的 `connectGatt` 方法进行设备连接，这个方法需要传入是否自动连接参数，如果设为自动连接，则断开后会不断尝试重连，以及连接成功之后的回调函数。该方法会返回一个 `BluetoothGatt` 对象代表本次连接。

5.4.4 服务、特征值发现

`BluetoothGatt` 对象作为中心设备与外围设备连接的桥梁，可以通过它来获取外设设备的所有信息。通过它的 `getServices` 方法可以获取 GATT 服务器定义的所有服务对象 `BluetoothGattService` 以及服务 UUID。服务和特征值属于包含关系，特征值可以通过 `BluetoothGattService` 的 `getCharacteristic` 方法来获取。

5.4.5 特征值读写

写入过程调用特征值对象的 `setValue` 方法设置需要写入的指令，再调用 `BluetoothGatt` 连接对象的 `writeCharacteristic` 方法可以进行写入操作。在写入操作的回调函数，会返回本次写入的数据，如果和期待发送的数据相同，则代表发送指令成功，此时要进行读取特征值操作来读取服务器响应，如果发送失败则尝试重发指令。

读取操作调用的 `BluetoothGatt` 对象 `readCharacteristic` 方法传入要读取的特征值对象，命令系统异步去读取指定的特征值，如果读取到，则会调用 `onCharacteristicRead` 回调函数，回调函数的参数会有读取到的特征值对象，通过该对象的 `getValue` 方法获取读取到的数据。

手机端蓝牙程序流程如图 5.5。

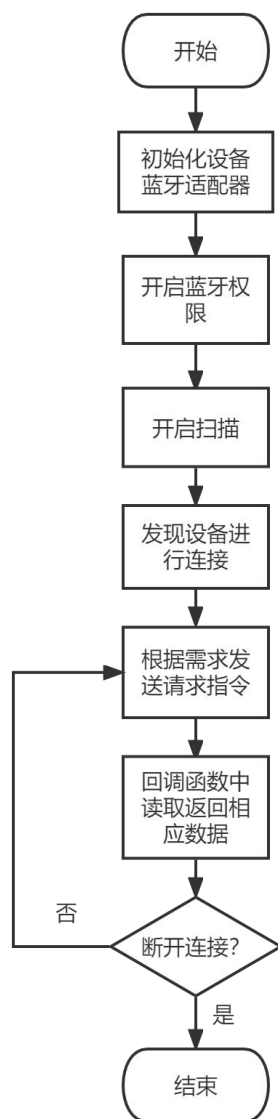


图 5.5 手机端蓝牙程序流程图

5.5 校园特色功能实现

基于连接的方式只支持一部手机与一个手环之间的交互，显然并不适用于校园使用场景中教师一部手机与学生多个手环之间的数据通信。故学生在校场景下，低功耗蓝牙采取基于广播的方式。

5.5.1 广播数据包格式设计

蓝牙广播包最大可以支持 37 个字节，其中 6 个字节被设备 MAC 地址占用，另外 31 个可用字节又有其特有的组织形式来传送数据。这种组织形式被叫做 AD structure，可以存在多个 AD structure。AD structure 格式如表 5.3。

表 5.3 AD structure 结构

Len	Type	Data
0	1	2-n

其中 Len 表示 AD structure 的长度（不包含其本身）；Type 表示 AD structure 的类型，由蓝牙官方定义；data 则表示具体数据。常用的 Type 含义如表 5.4。

表 5.4 常用 AD Type 含义

AD Type	含义
0x01	设备标识
0x08	缩略设备名称
0x09	完整设备名称
0x0A	发射功率
0xFF	厂商自定义数据

本文中蓝牙通信数据表设计如表 5.5。

表 5.5 蓝牙数据格式设计

ADV_LEN	ADV_TYPE	ADV_CRYP	ADV_ID	ADV_CMD	ADV_DATA
0	0xFF	2	3	4	5-n

其中 ADV_LEN 标识数据包长度，ADV_TYPE 固定为 0xFF 标识数据包为厂商自定义数据，ADV_CRYP 标识数据包是否加密，ADV_ID 标识广播发出数据的设备标识，ADV_DATA 为广播数据。

5.5.2 课堂答题实现

本文中设计处理的题目一共分为论述题、判断题、选择题三种类型，其中论述题的完成学生可以通过手环进行确认抢答，判断题和选择题学生可以通过手环发送答案。手环端通过广播数据包收到具体请求后，根据 ADV_CMD 位判断题目类型进行相应处理。对于每次答题过程，设计有答题上下文结构体如表 5.6 所示。

表 5.6 答题上下文结构体设计

字段	描述
begin_time	答题开始时间
spend_time	答题结束时间
question	题目 ID
question_type	题目类型
answer	学生作答答案

教师发送的题目广播包并不携带具体题目信息，只携带题目类型信息与题目 ID。手环端收到题目之后，设置好答题上下文相应字段，并且跳转到相应的屏幕界面进行答案的选取。

学生选取答案之后，拼接返回数据包，其中 ADV_CMD 与题目广播包相同，ADV_DATA 中保存学生答案与答题耗时，通过广播发送返回。

5.5.3 课表分发实现

手环具有课表功能，每天可以通过蓝牙网关向学生下发课表数据。课表数据广播的 ADV_CMD 为 0xA5，设计课表数据字段如表 5.7。

表 5.7 课表数据字段

字段	描述
course_list	存储课表数组
course_num	课程数量
week_type	单双周类型
day	枚举类型，当前星期日
curr_week	当前周数

广播数据受限于广播包大小限制，需要合理分配广播包中片段位置。广播数据中使用一个字节来表示 week_type、day、curr_week 字段，分别占据一个字节中的第 1 个 bit，第 2-4 个 bit，第 5-8 个 bit。随后的一个广播字节存储课程数量。具体课程在之后按序存储，课程在广播包中的表示形式是一个编码，手环的 flash 中存储了具体课程名称，可以从 flash 中根据具体编码获取课程名称到手环屏幕之上显示。

5.5.4 小红花数据分发

教师可以通过蓝牙广播向学生下发小红花表彰。该指令的 ADV_CMD 为 0xA8。下发小红花的蓝牙广播包含学生获得小红花数量、表现欠佳数量以及当前学生累积的总分。小红花下发虽然是从教师到学生的点对点，但是采用广播形式是为了节省蓝牙建立连接和断开的开销。顾在广播之中还要附加学生手环的 MAC 地址。手环收到广播包之后和自身 MAC 地址相比对，如果是自己的命令则进行处理。

5.6 本章小结

本章首先介绍了 BLE 协议栈，重点介绍了在开发 BLE 通信程序时主要接触的 GAP 层与 GATT 层。说明了智能手机与智能手环之间的蓝牙通信方式基于 C/S 架构，设计了双方进行通信的数据包格式。分别完成了服务器端（智能手环）与客户端（智能手机）的基于连接方式的低功耗蓝牙通信模块编写。完成了基于低功耗蓝牙广播形式的手环在校园中特色功能实现。

6 实验测试与结果分析

本章主要对本文做出的主要工作，心率检测、步数统计、低功耗蓝牙通信功能设计实验进行测试，并给出最终结果。

6.1 心率检测测试

心率检测测试需要考虑不同情况，设计了分别在静止状态、运动状态下对不同年龄段人群的测试。选取 10 名不同年龄段的志愿者，分别在静止站立、慢速行走、慢跑的情况下进行测试。进行对比的设备有当前市面上成熟的可穿戴心率检测设备，分别是小米手环 6，华为手环 6，Apple watch7。设备佩戴于距离手腕 1-2cm 处的手臂下方位置，并且与皮肤紧密贴近。下图 6.1 为智能手环与小米手环心率检测对比图。



图 6.1 智能手环与小米手环心率检测对比图

静止状态下采集结果如表 6.1:

表 6.1 静止状态下心率结果

测试者 编号	年龄	小米手环 6 数据 /BPM	华为手环 6 数据 /BPM	APPLE WATCH 数据/BPM	智能手环数据/BPM
1	23	68	68	66	67
2	28	84	82	82	83
3	23	72	75	74	73
4	24	68	69	70	68
5	32	62	63	62	63
6	43	65	64	64	65
8	55	59	60	58	59
9	48	67	66	66	68
10	25	78	79	77	78

从上表可以看出, 本文研制手环在静止状态下和市面上其他成熟手环差距很小, 微乎其微, 在不同年龄段人群都有很好的表现, 可以精准测出静止状态下受测者的心率。

行走状态下采集结果如下表 6.2:

表 6.2 行走状态下心率结果

测试者 编号	年龄	小米手环 6 数据 /BPM	华为手环 6 数据 /BPM	APPLE WATCH 数据/BPM	智能手环数据/BPM
1	23	75	74	76	75
2	28	93	92	91	92
3	23	83	84	85	84
4	24	85	86	85	88
5	32	84	84	85	86
6	43	80	79	78	81
8	55	75	76	74	77
9	48	75	75	76	77
10	25	91	92	90	89

上表中可以看出, 行走状态下, 本文研制手环仍旧与其它手环之间几乎没有差距, 可以说达到了同等的水平。

慢跑状态下测试结果如下表 6.3:

表 6.3 慢跑状态下心率结果

测试者 编号	年龄	小米手环 6 数据 /BPM	华为手环 6 数据 /BPM	APPLE WATCH 数据/BPM	智能手环数据/BPM
1	23	101	99	100	102
2	28	112	105	119	118
3	23	125	125	114	132
4	24	125	120	117	131
5	32	98	92	105	96
6	43	125	114	120	108
8	55	102	99	101	112
9	48	112	109	115	108
10	25	120	118	122	125

从上表之中可以看出,在慢跑状态下,本文研制手环和其他手环的测试结果产生了相比静止、行走状态下较大的出入,但也能大致反应出了当前受测者的心率状态。

6.2 计步测试

想要获取精准的步数很容易,只需要受测者在心中默记下所走的步数即可,故本测试不再选取其他产品作为对照。依然沿用心率测试的受测者,测试行走不同步数的计步情况,需要注意的事,测试状况下正常摆臂即可,不可故意摆平晃动手臂。下图 6.2 展示了手环计步界面。



图 6.2 智能手环计步界面

下表 6.4 展示了计步测试中的测试结果。

表 6.4 计步测试结果

测试者编号	实际步数	手环显示步数差	误差步数	误差精度
1	50	48	-2	4%
2	100	105	+5	5%
3	50	52	+2	4%
4	100	98	+2	2%
5	200	197	-3	1.5%
6	300	305	+5	1.6%
7	200	202	+2	1%
8	300	298	-2	1.5%
9	500	503	+3	0.6%
10	500	498	-2	0.4%

从上表可以看出，在不同步数下，本手环实际测得步数都与实际步数相差不大，并且测试步数越多，精度越高。而用户日常步数一般都在几千步以上，且步数只是要给用户一个大致的参考量，故在实际应用中，本手环计步功能完全可以满足用户需求。

6.3 BLE 通信测试

打开手环 APP 的连接界面，通过利用手环的加速度传感器，通过摇晃手环会自动发送可连接广播，此时设备可以被手机扫描到。手环连接 APP 界面如图 6.3 此时选择要连接的手环，即可完成手环与手机之间的蓝牙连接。



图 6.3 蓝牙配对连接界面

BLE 通信测试主要测试手环端到 Android 手机端的数据发送和接收,以及手机端到手环端的数据发送和接收。确保双方之间通信的正常。由于手环和手机之间采取 C/S 架构进行通信,故手机向手环发送请求数据命令,手环返回数据手机接收,就可以同时测试双方的接收发送都在正常状态。

在 Android 手机 APP 打开手环界面时,手机会自动向手环发起请求数据的命令,手环相应显示在 APP 界面之上。

经过反复多次测试,从图 6.4, 6.5 可以看出,手环采集的心率、步数、消耗卡路里等数据都已经同步到 Android 手机,并且正确显示在 APP 界面之上。

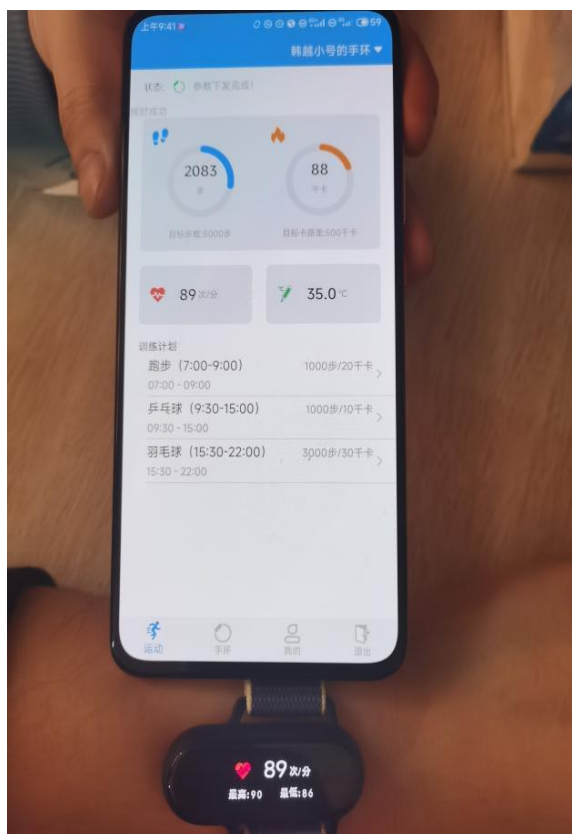


图 6.4 手机手环心率同步

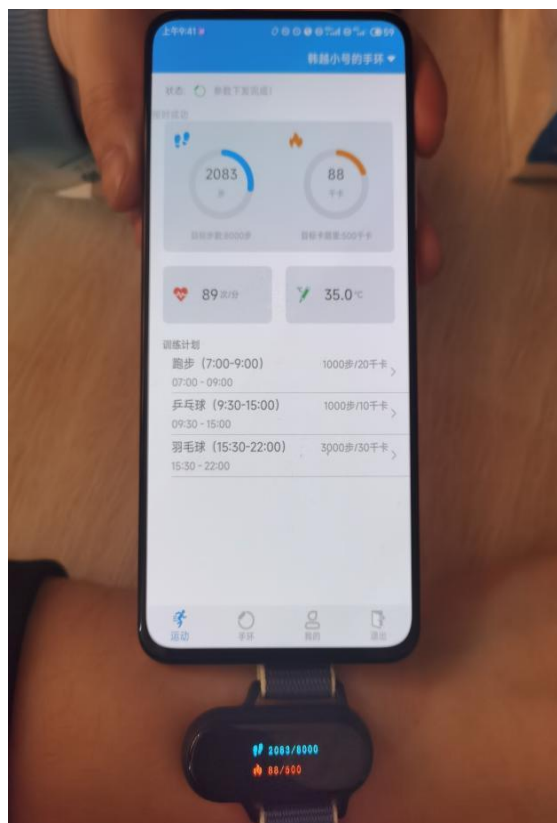


图 6.5 手机手环计步同步

以上为学生端下的基于连接的 BLE 通信测试,接下来测试校园场景下基于广播的 BLE 通信测试。

答题功能分为选择题与判断题,由 APP 教师端向蓝牙网关发送指令,蓝牙网关向教室中学生的手环发起广播。从图 6.6, 图 6.7 中可以看出,教师下发题目的指令成功显示到手环,手环也将学生做出的选择也返回给了教师端。

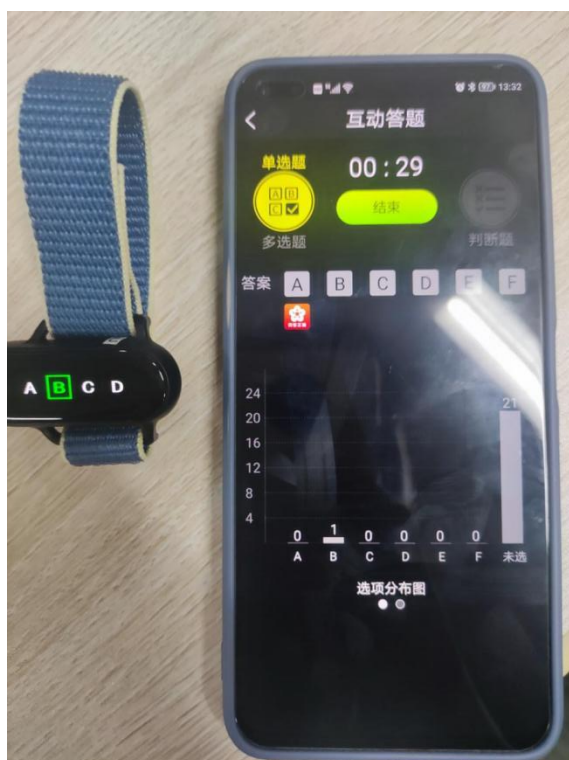


图 6.6 选择题答题测试



图 6.7 判断题答题测试

从图 6.8 中可以看出手环上成功同步了教师端 APP 下发的课表。

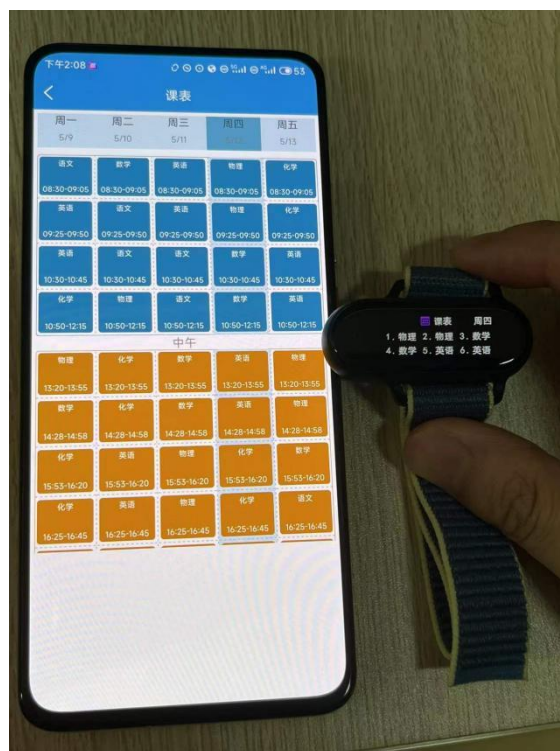


图 6.8 手环接收课表测试



图 6.9 教师下发小红花前



图 6.10 教师下发小红花后

图 6.9, 图 6.10 分别给出了教师下发小红花前后的变化, 可以看出手环可以正确接收教师下发的小红花蓝牙广播。

6.4 本章小结

本章主要对手环的心率检测、步数统计、BLE 通信进行了测试。心率检测功能在静止或者运动不激烈的状态下, 与市面上在售其它成熟手环之间几乎无差距, 在运动激烈的情况下有一定差距, 但也能正确反应当前心率情况。计步功能误差极小, 完全符合了用户计步的需求。最后对手环与手机 APP 之间的数据通信进行了测试, 分别测试了基于连接方式的家长端 APP 与手环的 BLE 通信、基于广播方式的在校场景教师端 APP 与手环 BLE 测试, 测试结果表明功能正常。

7 总结与展望

7.1 总结

可穿戴设备在人们的生活之中越来越常见,智能手环作为可穿戴设备的代表,最基础的两个功能就是心率检测与步数统计,无论在健康监测还是运动健身方面都有不可忽视的作用。本文来源于精伦电子股份公司合作智慧校园项目,开发出一款用于学生的智能手环,负责其中心率、计步以及蓝牙通讯的模块。主要工作包括:

(1) 给出了智能手环的软硬件设计。硬件方面给出了主控芯片和外设传感器的引脚连接配置;软件方面,利用主控芯片提供的调度器、定时器实现手环上心率模块、计步模块、低功耗蓝牙模块的总体设计。

(2) 完成心率检测功能实现。给出 PPG 信号降噪处理方案,实现了一个利用干净 PPG 信号计算心率的算法。完成光电心率传感器 PAH8011ET 的驱动程序,包括 I2C 通信实现、驱动初始化、PPG 数据获取流程的编写。

(3) 完成步数统计功能实现。实现了基于三轴加速度的数据波峰检测计步算法。完成加速度传感器 SC7A20 的驱动程序,包括驱动初始化、三轴加速度数据获取流程的编写。

(4) 完成了基于 C/S 架构的手环端与 Android 智能手机端低功耗蓝牙通信;完成基于低功耗蓝牙广播的校园特色功能实现。

(5) 分别测试了心率检测、步数统计、低功耗蓝牙通信功能,心率检测在静态心率下表现良好,动态心率测量虽然能体现出大致范围,但不够精准,步数统计功能符合预期,低功耗蓝牙通信正常。

7.2 展望

本文所研制手环距离成熟产品尚有一段差距,尚且存在的问题以及改进方向如下:

(1) 动态心率测量仍然不够准确,考虑是 PPG 信号滤波处理不够良好造成的,

自适应滤波方面可以考虑采取更先进的自适应算法，以及考虑除三轴加速度值之外的其他参考信号。

（2）心率检测测试部分没有专业的心率检测设备作为对比，只与市面上其它可穿戴心率检测设备做了对比。

（3）手环在校园中的应用仍在开发之中，还可以配合其它蓝牙设备完成更多如签到、上下课提醒、家长留言等实用功能。

参考文献

- [1] 韦哲, 薛翔, 吕克难. 可穿戴设备在医院诊疗中的应用研究进展. 中国医学装备, 2017, 14(8):170-173
- [2] Elizabeth C. Nelson, Tibert Verhagen, Matthijs Noordzij. Health empowerment through activity trackers. Computers in Human Behavior, 2016, 62(Sep.):364-374
- [3] 金英伟, 尚晓航. 技术创新产品购买意愿分析及预测——以智能手环为例. 消费经济, 2017, 33(1):70-77
- [4] CheukKei Lao, Binglong Wang, Richard S. Wang, Chang HsiaoYun Chang. The Combined Effects of Sports Smart Bracelet and Multi-Component Exercise Program on Exercise Motivation among the Elderly in Macau. Medicina (Kaunas, Lithuania), 2021, 57(1):342-346
- [5] 张梦琦, 刘宝存. 新时代我国开创教育对外开放新格局的政策走向——基于《中国教育现代化 2035》的解读. 中国电化教育, 2020(1):25-32
- [6] 陈诗茵, 施依娴, 李婷枫. 可穿戴设备在中小学智慧校园建设中的应用研究. 软件导刊(教育技术), 2019, 18(11):79-80
- [7] Miyashita R, Hirata Naoyuki, Sugino Shigekazu, Mimura M, Yamakage Michiaki. Improved non-invasive total haemoglobin measurements after in-vivo adjustment. Anaesthesia, 2014, 69(7):752-756
- [8] Jost B. Garrett, Podbielski M. Jeanette, Swartz D. Michail, Wade E. Charles, Moore J. Laura. Evaluation of Noninvasive Hemoglobin Measurements in Trauma Patients: A Repeat Study. Journal of Surgical Research, 2021, 26(6):213-221
- [9] Baulig Werner, Seifert Burkhardt, Spahn D. Rudolf, Theusinger M. Oliver. Accuracy of non-invasive continuous total hemoglobin measurement by Pulse CO-Oximetry in severe traumatized and surgical bleeding patients. Journal of Clinical Monitoring & Computing, 2017, 31(1):177-185
- [10] 彭福来. 基于光电容积脉搏波描记法的无创血红蛋白浓度检测技术的研究[博士学位论文]. 北京: 北京理工大学, 2016

- [11] 李晓霞. 人体血液成分无创检测的动态光谱理论分析及实验研究[博士学位论文]. 天津: 天津大学, 2005
- [12] 庞宇, 陈亚军, 汪立宇. 一种改进的变步长最小均方算法滤除心电信号运动伪迹的研究. 科学技术与工程, 2020, 20(8):3083-3087
- [13] Tamura Toshiyo, Maeda Yuka, Sekine Masaka, Yoshida Masaka. Wearable Photoplethysmographic Sensors—Past and Present. Electronics, 2014, 3(2):282-302
- [14] 龚渝顺, 吴宝明, 高丹丹, 闫庆广, 冯正权. 一种抗干扰穿戴式血氧饱和度监测仪的研制. 传感技术学报, 2012, 25(1):6-10
- [15] Ram M. Raghu, Madhav K. Venu, Krishna E. Hari, Komalla R. Nagarjuna, Reddy K. Ashoka. On the performance of AS-LMS based adaptive filter for reduction of motion artifacts from PPG signals. In: Instrumentation & Measurement Technology Conference (I2MTC 2011), Hangzhou, China, May 5-10, 2011, IEEE, 2011:135-142
- [16] Ram M. Raghu. A Novel Approach for Motion Artifact Reduction in PPG Signals Based on AS-LMS Adaptive Filter. IEEE Transactions on Instrumentation & Measurement, 2012, 61(5):1445-1457
- [17] Romero Indina, Danorl Geng, Berset Torfinn. Adaptive filtering in ECG denoising: A comparative study. In: Computing in Cardiology (CinC 2012), Krakow, Poland, September 9-12 2012. IEEE, 2012:45-48
- [18] Yan Liu, Michael Pecht. Reduction of skin stretch induced motion artifacts in electrocardiogram monitoring using adaptive filtering. In: International Conference of the IEEE Engineering in Medicine & Biology Society, (EMBC 2006), New York, USA, August 30 - September 3, 2006, IEEE, 2016:6045-6048
- [19] Yoon S. Won, Min S. Dong, Yong H. Yun, Seungpyo Lee. Adaptive Motion Artifacts Reduction Using 3-axis Accelerometer in E-textile ECG Measurement System. Journal of Medical Systems, 2008, 32(2):101-106
- [20] Lim C. Young, Jang K. Jin, Kim Hyun-woo, Kim Y. Hwan. A wearable healthcare system for cardiac signal monitoring using conductive textile electrodes. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2013), Osaka, Japan, July 3-7, 2013, IEEE, 2013:7500-7503

- [21] Mallinson Keith. Smartphone Revolution: Technology patenting and licensing fosters innovation, market entry, and exceptional growth. IEEE Consumer Electronics Magazine, 2015, 4(2):60-66
- [22] Moon Hyeonah , Park Sooyong , Choi Kwanghoon. An Android BLE Emulator for Developing Wearable Apps. KIISE Transactions on Computing Practices, 2018, 24(2):67-76
- [23] 李桐宇, 杨家玮. 用 SPI 总线实现 DSP 和 MCU 之间的高速通信. 电子元器件应用, 2006, 8(11):28-30
- [24] 易志明, 林凌, 李刚, 郝丽宏. SPI 总线在 51 系列单片机系统中的实现. 国外电子元器件, 2003(09):21-23
- [25] 何亚军, 邓飞其. 嵌入式 Linux 中 I2C 总线驱动程序设计. 计算机工程与设计, 2008, 29(10):41-43
- [26] 蔡型, 张思全. 短距离无线通信技术综述. 现代电子技术, 2017(3):65-67
- [27] 胡思雨. 2.4GHz ZigBee 与 WiFi 以及蓝牙系统间干扰分析[博士学位论文]. 西安: 西安电子科技大学, 2014
- [28] Decuir Joseph. Introducing Bluetooth Smart: Part II: Applications and updates. Consumer Electronics Magazine, IEEE, 2014, 3(2):25-29
- [29] Jung Jaehyo , Shin Shin , Kang Mingu, Kang Kyeong . Development of Wearable Wireless Electrocardiogram Detection System using Bluetooth Low Energy. Electronics, 2021, 10(5):608-613
- [30] 梁敏, 胡曦明, 李鹏, 马苗. “手机+可穿戴设备”的低功耗蓝牙安全实验技术. 计算机技术与发展, 2020, 30(11):111-116
- [31] 李兆亮, 张靖宇, 李立刚. 基于 MSP430 单片机和蓝牙的智能家居控制系统设计. 电子设计工程, 2021, 29(01):181-186
- [32] Changlong Yang, Ji Fang, Weiwei Wang. Design of Bluetooth-Based Smart Home Intelligent Controller. Applied Mechanics and Materials, 2013, 336-338(2):1540-1544
- [33] 单祥茹. 意法半导体 STM32 家族首次引进片上无线通信功能. 中国电子商情(基础电子), 2018(11):17-18

- [34] Ishikawa Takanori,Hyodo Yasuhide,Miyashita Ken,Yoshifuji Kazunari,ImaiYutaka. Wearable Motion Tolerant PPG Sensor for Instant Heart Rate in DailyActivity. In:10th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2017), Proto,Portugal,February 21 - 23, 2017:515-519
- [35] Ruisheng Lei,Bingo K. Ling,Peihua Peng Jinrong Chen.Estimation of HeartRate and Respiratory Rate from PPG Signal Using Complementary Ensemble Empirical Mode Decomposition with both Independent Component Analysis and Non Negative Matrix Factorization. Sensors, 2020, 20(11):3238-3251
- [36] Shynk J. John . Frequency-domain and multirate adaptive filtering. IEEE Signal Processing Magazine, 2002, 9(1):14-37
- [37] 夏晓. 自适应滤波器中 LMS 算法的研究及应用[博士学位论文].北京: 北京邮电大学, 2013
- [38] 李竹, 杨培林, 行小帅. 一种改进变步长 LMS 算法及其在系统辨识中的应用. 仪器仪表学报,2007,28(7):1340-1344
- [39] Yoon S. Won, Hangsik Shin, Min S. Dong. Adaptive motion artifacts reduction algorithm for ECG signal in textile wearable sensor. Ieice Electron Express, 2007,4(10):312-318
- [40] 刘军. 原子教你玩 STM32,寄存器版.第 2 版.北京: 北京航空航天大学出版社, 2015.
- [41] 王史春.基于 STM32 的 HAL 库和固件库嵌入式产品开发研究.电子测试, 2020(10): 71-72
- [42] 张洋, 刘军, 严汉字. 原子教你玩 STM 32:库函数版.北京: 北京航空航天大学出版社,2013.
- [43] 张志为, 曾婵. 基于三轴加速度计的双带通滤波计步方法及电子设备.中国, 发明专利, 202011518441.2, 2021
- [44] 徐叶菡, 史文婷, 陈茜.基于 BLE 及 Android 的蓝牙通信系统设计与实践.机电信息, 2016(12):168-169