

华中科技大学

本科生毕业设计 [论文]

基于 Android 的 ECG 信号监测与记录应用
程序实现

院 系： 光学与电子信息学院

专业班级： 在这里写下你的专业名称和班级号

姓 名： 某某某

学 号： U201200000

指导教师： 某某某

2016 年 6 月 10 日

学位论文原创性声明

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，本论文不包括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

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密口，在 年解密后适用本授权书

2、不保密口

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

摘要

目前, Android 系统已成为广泛使用的嵌入式设备系统, 具备计算能力强、便携、对外接口丰富、开发简便等特点; 同时, 心电图 (Electrocardiogram, ECG) 监测设备也向着小型化、智能化发展。因此, 将 ECG 监测设备和 Android 设备融合成为了不少研究努力的方向。有不少研究提出了基于 Android 平台的 ECG 信号检测系统的方案, 也有研究提出了专门针对嵌入式平台的 ECG 特征识别算法, 这些研究的目的是推进 ECG 监测设备的小型化, 同时利用 Android 方便接入网络、计算能力强大的特点开发 ECG 监测设备的在线预警功能。然而受 Android 设备尺寸等因素所限, 这些研究成果往往缺乏对 ECG 信号的有效展示, 医护人员不能方便地获取自己所需要的信息。

本研究提出了一种使用蓝牙 (Bluetooth) 技术连接 ECG 传感器、有效显示波形的 Android 应用程序, 其特点如下: (1) 利用 Android 横竖屏切换功能高效地显示波形, 波形显示快速稳定; (2) 提供了与用户交互的波形测量功能; (3) 使用基于 SQLite 的数据库技术有效存储、管理 ECG 数据; (4) 提供了准确稳定的 R 峰检测、RR 间期计算、心率计算功能。经测试, 该应用程序能够正常运行, 各项功能正常, 运行无卡顿。其中 R 峰检测算法的敏感度为 99.7%, 准确度为 99.7%, 基本达到使用要求。

关键词: Android 应用程序; 心电图; R 峰检测;

Abstract

Android System has been widely used as an embedded operating system recently. It owns powerful calculating ability, abundant I/Os and it is quite easy to develop on it. Meanwhile, electrocardiogram(ECG) monitoring devices are changing its way to mobility and smart. Therefore, the combination of them has become the main aim of most of the related studies. Many Android-based ECG monitoring system have been performed and also, there are some researches aiming at the specific ECG waveform analyzing algorithms. The ultimate goal of these researches is to build a mini-type ECG monitor as well as an online warning system based on Android's powerful calculation and easy access to the Internet. However due to the limited size of most of Android devices and other facts, these designs hardly could display the ECG waveform effectively, which means that the health care workers are not able to fetch what they expected before.

In this study, a new type of Android ECG monitoring application has been developed. It exhibits the ECG waveform with high level of effectiveness, with the Bluetooth connection to a ECG sensor. Below are its four main features: (1) its effective display method based on portrait and landscape layout switching; (2) a user interface to measure the waveform; (3) SQLite-based ECG database which is easy to manage and (4) an accurate, robust R-peak detecting algorithm. The application has passed a series of tests with a stable and normal performance. Tests shown that the Sensitivity of the R-peak detecting algorithm is 99.7% and the Accuracy of it is 99.7%, too. The application can be a success in monitoring ECG signals.

Key Words: Android Application; Electrocardiogram; R-peak detection

目 录

1	绪论	1
1.1	研究背景	1
1.1.1	Android 系统发展状况	1
1.1.2	心电图及其监测技术	1
1.2	相关研究介绍及其局限	2
1.2.1	相关研究介绍	2
1.2.2	相关研究的局限性	3
1.3	本研究的成果和意义	4
1.4	论文的组织结构	4
2	Android 平台开发相关技术简介	5
2.1	Android 系统简介	5
2.2	Android 系统蓝牙功能介绍	6
2.2.1	蓝牙串口协议简介	6
2.2.2	在 Android 建立蓝牙串口连接	7
2.3	Android 系统绘图功能介绍	7
2.4	Android 系统 SQLite 数据库简介	9
2.5	本章小结	9
3	ECG 监测与记录应用的设计与实现	11
3.1	应用程序需求分析	11
3.1.1	ECG 信号的产生与特点	11
3.1.2	用户需求	12
3.2	系统结构设计	13
3.3	各模块的设计与实现	14
3.3.1	显示模块的设计与实现	14
3.3.2	蓝牙传输模块的设计与实现	16
3.3.3	数据库管理模块的设计与实现	17
3.3.4	数据分析模块的设计与实现	19

3.4	主线程与各模块的通信	20
3.5	本章小结	21
4	R 峰检测算法及其实现	23
4.1	相关研究中的算法实现	23
4.2	基于局部最大值的 R 峰检测算法	23
4.3	R 峰检测算法的 Java 实现	25
4.4	本章小结	26
5	测试与结论	27
5.1	R 峰峰值检测算法的测试与结果	27
5.2	ECG 监测与记录应用的运行测试	27
5.3	结论与展望	27
	致 谢	29
	参考文献	30

1 绪论

1.1 研究背景

1.1.1 Android 系统发展状况

Android 系统是由 Google 公司开发的一款面向智能手机、平板电脑等终端的操作系统。近年来,采用 Android 作为操作系统的手机持续占有 80% 以上的手机市场 [1],而其中采用 5.0 及以上版本 Android 系统的手机占 Android 平台设备总量的 43% [2]。可以说,基于 Android 系统(尤其是 Android 5.0 及以上版本系统)的智能设备已经广泛普及。因此,采用 Android 平台作为运行环境的应用程序能够获得较为广泛的使用。

作为一款智能设备操作系统,Android 提供了对蓝牙(Bluetooth)通信、无线局域网(Wireless LAN, WLAN)通信和近场通信(Near Field Communication, NFC)等通信模式的底层驱动支持和应用程序接口支持,方便其与外部设备的沟通 [3]。同时,采用 Android 系统的智能手机等设备的处理性能普遍高于一般的专用嵌入式设备,这也允许人们通过智能手机进行一些较为复杂的运算(如面部识别),减少了日常生活中的运算成本。通过 Android 平台,人们已经实现了对家用电器、个人电脑、显示器等设备的简单控制,同时一些简单的数据处理工作(如个人文档处理、面部识别、语音识别等)也能够借助 Android 平台实现。

与此同时,医疗设备的小型化、便携化成为了医学领域电子设备发展的趋势。由于近年来移动终端计算能力的提高,医学设备的小型化、便携化程度逐步提高。更加轻便的医学设备既简化了医学检测方法,又方便了未经医学训练的普通人对自己身体状况的关注,在体育运动领域也体现出了潜在的应用价值 [4]。在这种背景下,将医疗设备与 Android 设备结合是一个自然而然的想法:Android 显然有助于在将医疗设备便捷化的同时,保证医疗设备所需的计算能力。而对于医疗设备中形式多样的生物传感器,Android 设备也可选择使用蓝牙、WiFi、NFC 等多种通信方式与之连接。目前在 Android 平台,已有多种健康类应用供用户选择下载或购买。

1.1.2 心电图及其监测技术

心电图(Electrocardiograph, ECG)反映了心脏功能的电信号活动记录,这些电信号由心脏中心肌细胞的除极和复极过程产生,并且可以通过在人体表面贴加

电极（被称为导联）、测量不同部位的电势差而记录下来 [4][5]。通过观察 ECG 信号的形态和一些关键点的数据，医生能够判定出病人的心脏功能存在哪些异常，而正常人心脏跳动所形成的 ECG 信号能够提供诸如心率等主要的人体体征指标。

由于 ECG 监测设备要求使用者长时间佩戴，因此便携性成为该类监测设备的发展方向之一。目前的医疗用便携式 ECG 监测设备均具有 ECG 信号记录、简单的信号显示和心率测量功能，经过记录的 ECG 数据往往存入 SD Card 中，需要分析时，将 SD Card 中的数据拷贝到计算机中进行分析。这些设备均配备了专用的低噪声电路，并且对信号进行了预处理，保证了 ECG 信号记录的有效性。还有部分便携式 ECG 设备也具有测量血氧饱和度（SpO₂）的功能。目前使用的便携 ECG 监测仪主要用来完成长时间的 ECG 信号记录工作，并不能进行较为复杂的信号分析，且波形显示方法也缺乏交互性。

为了简化 ECG 数据的采集和处理过程、提高相关病情诊断效率，一些研究提出了借助智能手机平台处理 ECG 信号的设想。这些研究所提出的 ECG 监测仪大多由两部分组成：具备数据传输功能的可佩戴式 ECG 信号采集设备，以及涌来接收信号并加以处理的智能设备（比如一台 Android 智能手机）。已有研究证实了通过这种方法在 Android 设备上手机 ECG 等医疗数据的可行性，一些适用于 Android 平台的 ECG 信号处理算法也在验证后提出并应用。基于此，一些研究提出了专门处理 ECG 数据的 Android 平台应用这一构想。

1.2 相关研究介绍及其局限

1.2.1 相关研究介绍

已有研究主要着眼于两方面：基于 Android 平台 ECG 监测应用的开发，以及 Android 平台上对 ECG 信号的处理算法。

利用 Android 智能设备的蓝牙模块与 ECG 采集设备连接来获取数据、再加以处理，最后在 Android 端绘制波形的方案成为了相关研究的普遍选择。[6] 中的研究验证了 Android 设备上蓝牙连接的稳定性和速度：通过多线程技术实现了 Android 与最多 7 个蓝牙设备间的连接和数据传输，并指出在 Android 系统建立了 4 到 7 个基于 RFCOMM 协议的蓝牙连接后，传输速率保持在 20KB/s 到 50KB/s 之间，能够满足 10KHz 16bit 数据的传输。因此蓝牙连接完全能够满足两种设备间传输 ECG 信号的要求。参考文献 [4][7][8] 中的研究分别实现了 Android 设备对

ECG 信号的图形化显示。这些研究中的 ECG 显示模仿了 ECG 定走纸的形式，在 Android 设备屏幕上绘制出带有坐标轴的 ECG 信号图形，并且实现了多个导联信号的同时显示。研究 [8] 还实现了心率波动波形的显示。

在 ECG 监测设备的功能性方面，许多研究利用 Android 平台的特点开发出了实用的功能。研究 [7][8] 根据 Android 设备方便接入互联网的特点，实现了通过 WiFi/3G 网络将 ECG 信号或其他生理学指标传输到远端服务器或另一 Android 设备，供医生远程诊断的功能。研究 [8] 中实现了 ECG 信号异常的警报功能，当警报触发时，应用能够自动输出长度为 10s 的 ECG 记录，并与医生的 Android 设备建立 HTTP 连接，传输图像。

同时，另一些研究尝试提出适合移动设备和可穿戴设备的信号分析算法。ECG 信号十分微弱，容易受到外部环境和其他生理活动的干扰，对此一些研究提出了 Android 平台和可穿戴设备上无效 ECG 信号的识别方法，成功地对带噪声的 ECG 信号进行了识别和分类，并进行信号质量评估 [9][10]。研究 [11] 提出了基于 RR 间期序列的心房颤动检测和基于机器学习的心室纤维性颤动检测算法。

1.2.2 相关研究的局限性

现有的 ECG 监测应用研究中，ECG 波形在小尺寸屏幕（比如大多数 Android 智能手机的屏幕）上的显示不尽人意。由于手机屏幕无法完全显示标准的 ECG 波形，部分现有研究选择了等比例缩小 ECG 波形后完全显示，这样一来 ECG 波形的细节就无法方便观察；而使用横向屏幕显示完整比例的 ECG 波形时，留给用户进行操作的区域面积则减少了。同时，大部分研究在显示 ECG 波形时模拟了 ECG 定走纸的样式，这妨碍了数据的快速读取。

ECG 信号处理算法方面，现有的算法着重解决某些疾病的自动检测及预警问题，已经能够有效检测出某些异常情况；另一些算法着重于实现移动设备和可穿戴设备上 ECG 信号的质量监测以及噪声消除 [10]。但是现有的 ECG 监测应用并没有很好地将 ECG 信号的重要特征提取并表示出来，而是在后台获取 ECG 特征后判断是否需要发出预警。造成的结果是对于临床应用来说，现有 ECG 监测应用并不能有效地减少医护人员在分析 ECG 信号时的负担。

1.3 本研究的成果和意义

基于以往研究,本论文实现了 Android 5.0 及以上版本的 ECG 监测与记录应用,该应用能够通过蓝牙与 ECG 传感器连接,并创建基于 SQLite 的数据库,方便数据读取。同时该应用采用了两种波形显示方法,兼顾用户使用和波形细节的展示。在细节显示方面,提出一种触控标尺功能,解决了传统坐标中读取数据繁杂的问题。

在数据处理方面,提出一种新的基于局部最大值的心率检测方法,该方法能不受基线漂移和振幅突变的影响,准确地检测到 ECG 信号中 R 峰的位置;同时该方法具有迭代次数少、计算量小的特点,方便在嵌入式平台上实现。经测试,该算法的准确率为 99.7%,敏感度为 99.7%。

通过提出新的适用于 Android 端 ECG 应用的交互界面,本研究在专用于波形显示的交互界面设计上做出了新的尝试;而通过提出新的 R 峰探测方法,本研究在 ECG 波形特征提取的关键步骤上做出了自己的探索。

1.4 论文的组织结构

本文内容按如下结构组织:

第一章介绍了医疗设备小型化的背景,总结出 Android 设备能够协助推进医疗设备小型化进程的观点,回顾了相关研究获得的成果、这些研究的关注点以及存在的不足,介绍了本研究获得的成果和其代表的意义;

第二章介绍了 Android 平台开发相关的背景技术,主要包括 Android 系统介绍、蓝牙技术在 Android 平台上的应用和其它 Android 平台支持的技术。这些技术在研究中均有涉及或实现;

第三章介绍了基于 Android 平台的 ECG 监测记录应用的设计、实现过程。从需求分析开始介绍了该应用完整的实现过程;

第四章介绍了应用中实现的心率检测算法及使用 Octave 进行算法建模的过程;

第五章介绍了应用中相关算法的测试方法和测试结果,总结了应用的特点与不足,并对未来的相关工作进行了规划。

2 Android 平台开发相关技术简介

2.1 Android 系统简介

Android 系统是由 Google 公司开发的一款主要面向触摸屏设备的直接控制式开源操作系统。该系统基于 Linux 内核，可运行在基于 32 位或 64 位 ARM 架构、x86 架构和 MIPS 架构的设备上。目前 Android 系统已经广泛应用于智能手机、平板电脑、笔记本电脑、智能电视和一些可穿戴式设备上。近年来，Android 系统的市场占有率持续领先其他同类型智能操作系统，是一款便于使用、便于开发的操作系统。

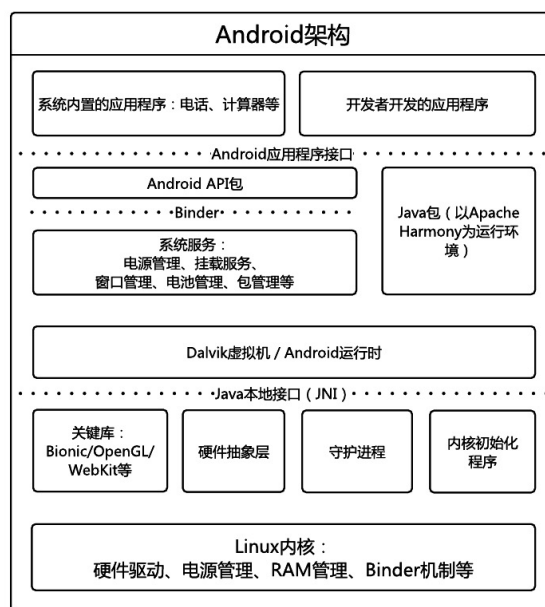


图 2-1: Android 系统架构

Android 系统由 Linux 内核及底层组件、Dalvik 虚拟机及运行时环境、系统服务、应用程序框架和应用程序四部分组成，其详细系统结构如图2--1所示。Linux 内核中包含 Wakelocks、Binder、RAM Console 等底层驱动和 IPC/RPC 机制。正是这些底层驱动实现了 Android 系统对音频、摄像头、显示、蓝牙、GPS 等外围设备的支持。在该内核上还运行着一些关键库，如 Bionic 标准 C 语言库（libc.so、libstdc++.so、libdl.so 等）、libhardware.so 硬件抽象层、libbinder.so Binder 标准库等。这些库函数通过 Java 本地接口和 Dalvik 虚拟机进行沟通。Dalvik 虚拟机是一种针对嵌入式系统设计的 Java 虚拟机，在更加节省运行所需空间的同时，还保证了程序运行的效率（这得益于 Dalvik 包含的 Just-in-Time 编译器可以将程序直接

转换为二进制指令后运行) [3]。系统服务为 Android 应用提供了与硬件抽象层沟通的方法, 这些服务既有用 Java 语言实现的, 也有用诸如 C、C++ 等其他语言实现的。当一个应用程序需要调用系统服务时, 它首先从服务管理器查得需要调用的系统服务的句柄, 然后通过 Binder 获得该系统服务并且使用它 (比如调用其中的某个方法)。最上层、也是最贴近用户的 Android 应用程序提供了人机交互接口, “计算器”、“录音机”等应用程序——也即人们平时所称的 APP——均属于应用程序。

2.2 Android 系统蓝牙功能介绍

蓝牙 (Bluetooth) 技术是一种基于 2.4GHz 频段设备的射频通讯标准, 使用蓝牙技术传输数据的功耗更低, 成本小, 易于使用 [12]。

2.2.1 蓝牙串口协议简介

蓝牙串口协议 (Serial Port Profile, SPP) 允许两台支持蓝牙技术的设备通过 RFCOMM 协议建立起虚拟的 RS232 串口传输通道 [13]。一台蓝牙设备可以选择性地兼容蓝牙技术中若干协议的某几种, 若要使用 SPP 协议传输数据, 两台蓝牙设备需要支持 LMP、L2CAP、RFCOMM、SDP 这四种协议, 同时需要串口仿真器作为应用层进行控制。SPP 的传输速度可达 128kbps 或更高。建立 SPP 通信的典型步骤如下:

- 通过 SDP 协议查找并发现附近所有 RFCOMM 服务器信道, 并向用户返回一个连接列表以供选择。如果有特定的选择对象, 可以在搜索时使用 UUID 进行筛选;
- 对将要连接的设备进行安全认证;
- 向将要连接的设备发送一个新的 L2CAP 信道连接请求;
- 在这个 L2CAP 信道上初始化两者间的 RFCOMM 连接;
- 使用服务器信道的 UUID 建立新的数据连接。

Android 平台一直以来都对蓝牙技术保持着良好的支持。使用 Google 提供的 Android API, 程序开发人员可以方便地完成如上所述的 SPP 通信连接。

2.2.2 在 Android 建立蓝牙串口连接

Android API 为开发者提供了 `BluetoothAdapter` 类和 `BluetoothDevice` 类来实现蓝牙串口链接的相关操作，使用这些类的步骤和建立一个 SPP 链接的步骤基本相同。首先在请求到蓝牙模块的控制权后，创建并实例化这两个类，使用 `BluetoothAdapter` 的方法开启蓝牙适配器并发起搜索，搜索到的设备将返回其蓝牙地址，使用这个蓝牙地址就能配对并建立连接 [14]。

Android 使用 `Socket` 作为 SPP 协议中的应用层。建立连接后，使用 `BluetoothSocket` 类可以创建基于 RFCOMM 的无线通信。在通信中，Android 设备既可以作为服务器端，也可以作为客户端。作为客户端时，使用 UUID 指定需要连接的设备，并且建立连接，最后使用 `InputStream` 类获取服务端传递的讯息。此时该信息为 `byte` 类型变量，并存储在缓存中等待处理。Android 平台蓝牙连接的过程可见图2--2。

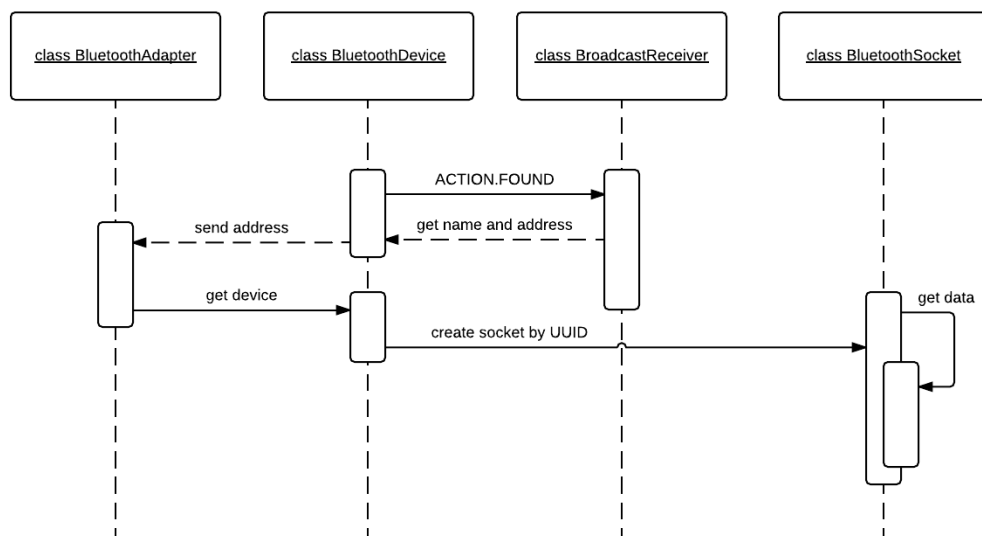


图 2-2: Android 平台蓝牙连接 UML 时序图

2.3 Android 系统绘图功能介绍

在绘制 ECG 波形时，不可避免地要接触到 Android 系统的绘图功能。在 Android 系统中，`View` 类承载了界面绘制的功能。诸如按钮（`Button` 类）、文本（`TextView` 类）和界面布局（各种 `Layout` 类）都以 `View` 作为基类。一个 Android 活动（`Activity`）通过获取 `View` 类来生成界面并且展示给用户，从而产生了使用者最终看到的画面。作为绘制界面的基石，`View` 类自然实现了可以自定义绘制的

方法 onDraw()。但 View 类的 onDraw 操作必须发生在主线程中，所以更新 View 类的时候，主线程必然是阻塞的，应用程序不会对用户的操作做出回应。这时，如果 View 更新过程超过 5 秒，耗时的更新操作还可能引发应用程序未响应的错误（Application Not Responding，ANR）从而强制停止该程序。

使用 SurfaceView 类绘制图形可以解决这个问题。顾名思义，该类是包含了一个“表面（Surface）”的 View 类（事实上 SurfaceView 确实继承了 View 类）。SurfaceView 绘图时，向“表面”，也即真正显示的区域提交的内容和绘画处理的内容可以是不同步的，即该类允许在后台进程中分别进行图像绘制和图像预处理（如加载一个 JPEG 文件）。因此在图像不更新时，可以先将下一次的绘图预处理，加快下次的显示时间。同时，该类还允许使用者通过重写一个回调函数来获取 SurfaceView 类对象的状态，并进行相应操作（如初始化绘图线程和销毁绘图线程）。

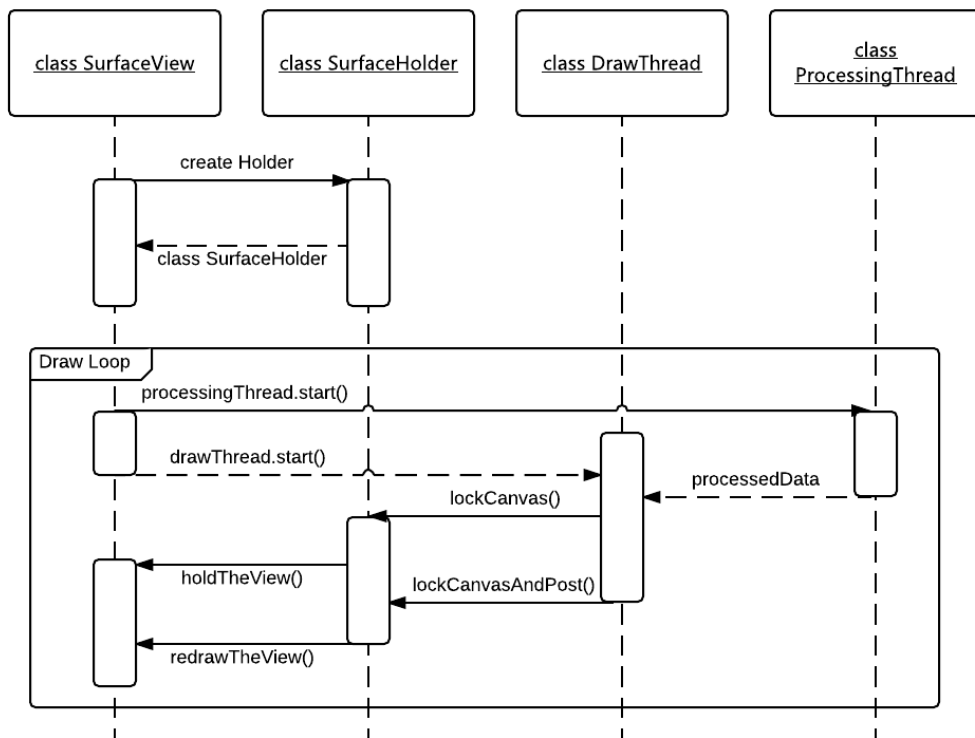


图 2-3: Android 平台 SurfaceView 绘图 UML 时序图

使用 SurfaceView 类绘图的过程如图2--3所示。在实例化各类后，使用 SurfaceView 获取 SurfaceHolder，之后就可以用实现好的回调函数启动图像绘制和处理进程了。这个过程可以循环，以实现图像的持续更新。

2.4 Android 系统 SQLite 数据库简介

通过使用 Android 系统中内嵌的 SQLite 数据库功能, ECG 波形数据可以被方便地组织为 SQLite 数据库的形态, 从而方便数据的查找和输出。

SQLite 是一款独立的、无服务器的、零配置的事务型 SQL 数据库引擎 [15]。SQLite 不含服务器端, 能够完全嵌入其他程序中, 对本地磁盘上的数据库文件进行管理。同时, SQLite 的体积十分小巧轻便。一个完全版的 SQLite 函数库不足 500KB, 对运行内存的大小也没有苛刻的限制, 这使得 SQLite 十分适合嵌入式平台的应用。作为一款数据库引擎, SQLite 十分稳定, 很好地保持了数据库事务执行的 ACID 原则 (指原子性、一致性、隔离性和持久性)。

Android 自带 android.database 包, 支持 SQLite 数据库的操作。通过实例化 SQLiteDatabase 类, 能够得到一个 SQLite 数据库。而通过继承 SQLiteOpenHelper 类, 能够打开已存在的数据库或创建新的数据库。SQLiteOpenHelper 还提供了一些回调方法, 便于开发者在数据库遭到修改时作出反应。

在 Android 应用中, 提交一个 SQLite 事务的步骤如下:

- 调用 SQLiteDatabase.beginTransaction() 方法开始事务;
- 进行事务操作 (如插入一条记录);
- 调用 SQLiteDatabase.endTransaction() 结束事务。

同时为方便数据库的查询, Android 提供了 Cursor 类简化诸如查询、遍历数据库等操作。Cursor 提供的方法包括返回数据、查询列名、查询列总数、查询数据、移动光标等。

2.5 本章小结

本章简要介绍了 Android 系统的结构、各结构间的关系、概括了各结构是如何协同运行、最终形成一个完整的系统的; 介绍了蓝牙技术的基本情况、蓝牙技术所包含的 SPP 协议是如何在 Android 平台上得到应用的; 最后概述了 Android 平台中的两个功能性 API 类: SurfaceView 类和 SQLiteDatabase 类的机制和简要实现方法。

本章所述内容在 ECG 监测与记录应用的实现中均有涉及, 并且构成了应用的几个主要类中的重要部分。可以说, 本章介绍的内容可视为应用的主要结构。正

因为不同内容所对应的是应用中需要对应的不同现实对象（如 Bluetooth 相关类对应了有关蓝牙设备的操作、SurfaceView 对应了绘图区、SQLiteDatabase 对应了 ECG 数据库等），本应用开发过程严格遵循”面向对象”的原则才得以贯彻，并且其合理性也更能得到印证。

3 ECG 监测与记录应用的设计与实现

3.1 应用程序需求分析

通过分析 ECG 波形的基本形态、关键特征并参考已有设计的特点后，能够归纳出用户对 ECG 监测与记录应用的需求。

3.1.1 ECG 信号的产生与特点

心脏通过心肌组织的受控收缩 / 舒张完成泵血的工作，而心肌组织由心肌细胞组成。每个心肌细胞含有钾、钠、钙等离子，通过输入和泵出这些离子，心肌细胞完成舒张和收缩的过程，相应的，由这些过程造成的细胞内外电位差变化被称为“复极”和“除极”[4]。而通过在人体表面施加电极（被称为导联），记录下这种电位差变化而形成的图形被称为心电图（Electrocardiogram, ECG）[5]。如图3--1所示，除极过程和复极过程表现出来的细胞内外电势差方向是相反的，因此两过程形成的波形方向也相反，心电图也因此包含正负两个方向的电位变化。

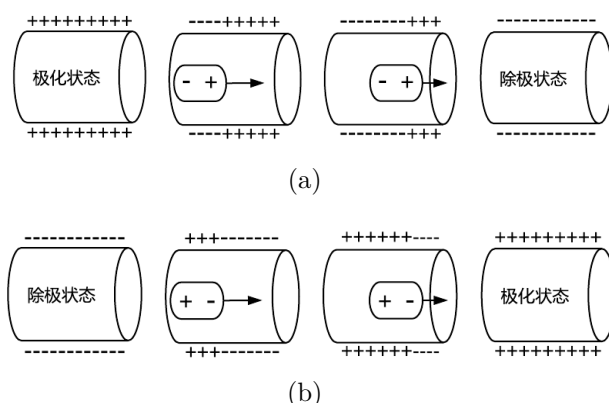


图 3-1: 心肌细胞的除极与复极过程: (a) 为除极过程, (b) 为复极过程

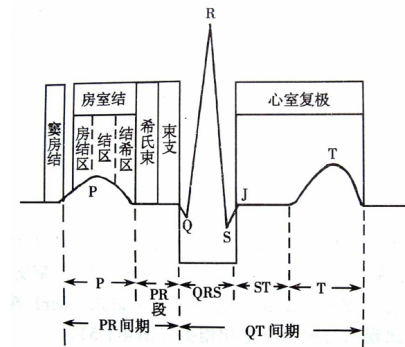


图 3-2: ECG 信号波形示意图

若将所有心肌细胞内外电势差看为一个电矢量，则它的位移局限在心肌内，可近似看作电偶固定在心脏中心，心脏中不同部位的心肌组织除极时间存在差异，从而造成了这个矢量方向的转动。同时，电矢量的强度也随心肌大小、薄厚的改变而改变。因此将导联放置在身体不同位置时，对应的电矢量变化不同，测得的心电图也不同。实际的心电图测量往往包含肢导联和胸导联，不同的导联放置位置不同。在临床实践中通常综合 12 个导联的 ECG 波形来判断心脏状态。

单个导联的心电图通常包含 P 波、QRS 波群和 T 波，这是根据波的大小和形

态而划分的（见图3--2）。通过观察一幅 ECG 图像的 PQRST 波及其特点，能够了解心脏某部分或整体的搏动状况。在医学生的训练中，以下 ECG 图像特征的观察备受强调：

- 两个 P 波或 R 波的间隔时间（PP 间期或 RR 间期）；
- PP 或 RR 间期的离散程度；
- P 波持续时间；
- P 波振幅；
- P 波与 R 波的间隔时间（PR 间期）；
- QRS 波群的间隔时间（QRS 间期）；
- S 波结束与 T 波开始之间的间隔时间；
- 经过矫正的 QT 间期（即 QT_c ，可通过 $QT_c = \frac{QT}{\sqrt{RR}}$ 计算，其中 QT 表示实际 QT 间期，RR 表示 RR 间期）

除此以外，还有一些基于波形形态的观测（如 QRS 波的主波方向）在诊断中也十分重要。这些波形形态能够反映心脏房室肥大情况、心肌缺血情况、心肌是否坏死等。

通过了解 ECG 信号的产生原理和主要特征，医护人员对于 ECG 监测记录应用的需求便可以总结出来。

3.1.2 用户需求

用户的需求直接决定了功能的设计和实现。本研究假定使用 ECG 监测记录应用的用户群体为医护人员，根据 ECG 信号的基本特征，可以将用户的需求总结如下：

业务需求：能够实时观察 ECG 波形的变化；能够快速了解到波形中某些时刻的幅值；能够观察多个周期 ECG 的波形以观察其形态；能够观察 ECG 波形中较为具体的细节。

功能需求：能够记录测量到的 ECG 波形；能够提取关键的 ECG 信号特征（如心率）并展示出来；能够提供测量 ECG 波形中两点间的时间间隔和幅值差；能够计算出某些特征值的波动情况（如心率的波动情况）。

非功能性需求：应用运行时应顺畅无卡顿；波形记录应能准确记录某时间点对应的数值；界面应该易于使用，没有误导性；应用的核心功能应该提供接口，以便二次开发。

根据上述业务及功能需求，总结出该应用的 UML 用例图如下：

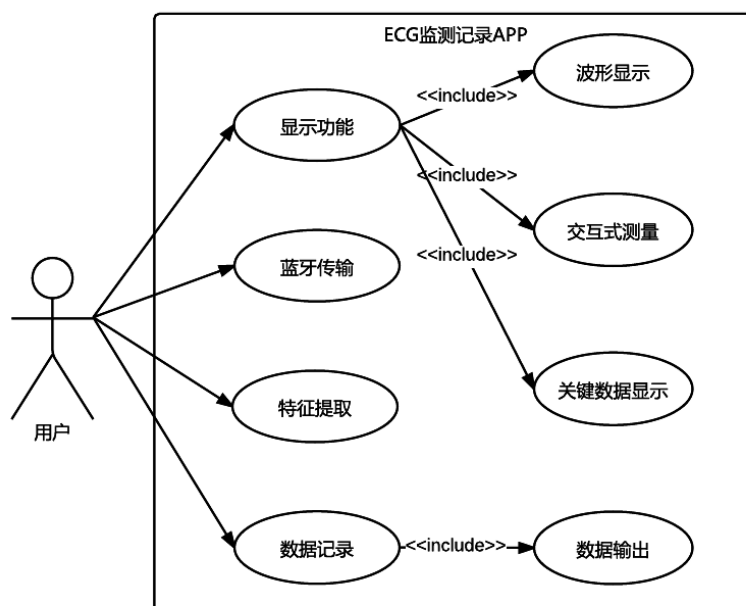


图 3-3: ECG 监测记录应用的 UML 用例图

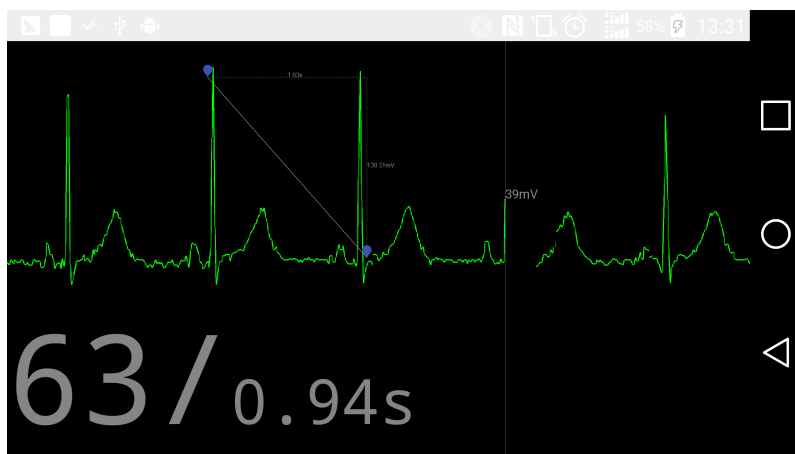
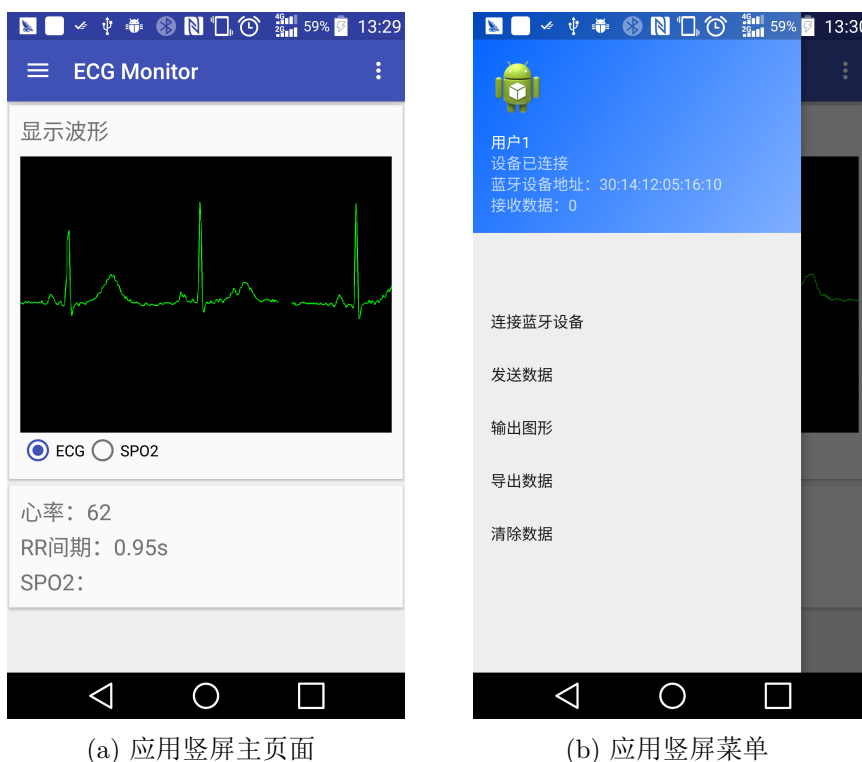
3.2 系统结构设计

以应用的功能为参考，本研究使用模块化结构实现应用的功能，保证每个类对相应对象的抽象更加准确。根据上文总结的各类需求，将应用的设计与实现分为蓝牙传输模块、显示模块、数据库管理模块和数据分析模块四个部分，各个模块之间呈现松耦合的关系，通过在主线程中接受各模块信息、调用各模块的成员函数实现应用的功能。

3.3 各模块的设计与实现

3.3.1 显示模块的设计与实现

显示模块在保证波形顺畅显示的同时,也需要满足用户观察波形的需求和交互式测量的需求。如 1.2.2 节所述,现有研究的显示方式无法解决用户操作和波形完全显示之间的矛盾。为解决此问题,本研究在设计显示模块时利用了横竖屏各自的特点,设计了两种波形显示方式:竖屏时概略显示波形,同时提供更多的操作选项;横屏时着重显示波形及其细节,辅以显示波形的关键特征。两种界面的示意图如下所示:



(c) 应用横屏显示

图 3-4: 应用的布局方式

在 Android 平台中实现横竖屏布局切换较为容易：在工程的 res 资源文件夹中准备两种布局文件 activity_main.xml 与 activity_main_land.xml；当屏幕状态改变时，系统会自动调用 onConfigurationChanged() 方法。通过重写该方法使系统根据不同的屏幕状态选择布局文件和适合屏幕的 SurfaceView 类。

对于波形的显示，创建 DrawSurfaceView 类用于管理和绘制显示波形的 SurfaceView 类。该类需要实现的功能包括：根据获取到的数据绘制波形、显示波形的特征、提供交互式的波形测量方法。针对每种绘制工作，该类使用一条线程实现。

对于绘制波形，每个数据点在 SurfaceView 中的坐标是关键。首先，按数据值占 ECG 信号最大值的比例计算出该数据点在显示框中的纵坐标；对于横坐标，以数据点的序号作为绘制的横坐标，当横坐标超出绘制区宽度则从头绘制。确定坐标以后启动绘制进程，锁定画布，根据屏幕方向更新对应的 SurfaceView 即可。但由于 Android 系统存在其他后台活动的原因，当 ECG 信号发送频率达到 300Hz 以上时波形显示将会出现严重的丢帧现象。虽然 ECG 采集速率一般低于 300Hz，但 ECG 传感器发送频率可能高于此值。因此在绘制波形时对收到的信号以 20Hz 重新采样，得到流畅的显示效果，同时对波形的观察没有太大影响，此时信号记录按接收速率保存数据，并不重新采样。总体绘制的步骤使用 Java 代码表示如下：

```

1 public DrawSurfaceView() {
2     x = 1;
3     shouldRefresh = false;
4     setTimer(50);
5 }
6 ...
7 public void drawPoint(int drawX, int drawY) {
8     if (shouldRefresh) {
9         calculateXandY(x,y);
10        shouldRefresh = false;
11        drawThread = new DrawThread();
12        drawThread.start();
13    }
14 }
15 ...
16 private class DrawThread extends Thread{
17     ...
18     @Override
19     public void run() {
20         try {
21             if(x>surfaceWidth) x = 0;
22             lockCanvas(lastx,0,x+100,surfaceHeight)
23             drawLine(lastX,lastY,x,y);
24         } finally {
25             unlockCanvasAndPost(canvas);
26         }
27     }
28 }

```

对于关键数据（如心率、赋值等），使用 `drawText()` 方法绘制在图上即可。对于测量标尺，需要检测 `onTouch` 触摸事件以做出响应反馈。对显示标尺的 `SurfaceView` 设置 `onTouchListener` 监听器，发生触摸时记录该点位置，发生触摸移动时绘制用户画出的直线；结束触摸时显示标尺量出的数据。标尺的绘制同样使用 `SurfaceView` 机制，在此不作赘述。标尺的显示效果见图3--4。

3.3.2 蓝牙传输模块的设计与实现

蓝牙传输模块保证了数据的正确接收，当接收到数据后还会向主线程发送信号、提醒主线程即时显示数据、将数据插入数据库。蓝牙传输模块的 UML 类图如下：

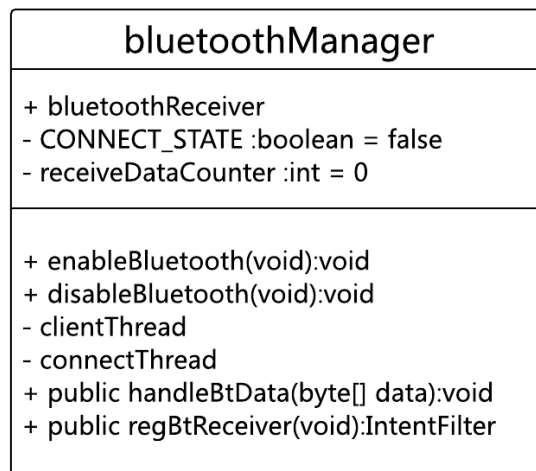


图 3-5: 蓝牙传输模块的 UML 类图

其中 `CONNECT_STATE` 负责表明当前连接状态，`receiveDataCounter` 表示接收的数据总数，`bluetoothReceiver` 表明搜索到的蓝牙种类及处理方式。在各项方法中，`enableBluetooth()` 与 `disableBluetooth()` 方法负责打开蓝牙、关闭蓝牙；`clientThread` 线程负责建立 RFCOMM 连接，`connectThread` 负责在已建立的 RFCOMM 上通过 `socket` 循环接收数据，并调用 `handleBtData()` 方法处理数据、传送给其他模块；`regBtReceiver()` 方法用来在主线程初始化时注册蓝牙广播接收器。

利用广播接收器 (`BroadcastReceiver`) 接收检测到的蓝牙设备是蓝牙连接的开端。在主线程注册蓝牙广播接收器后，若有蓝牙设备探测到，系统发出相应广播，`bluetoothReceiver`(继承自 `BroadcastReceiver`) 被触发，并通过蓝牙设备名称筛选蓝牙设备。当找到需要连接的设备后，获取其地址和设备，开启一个 `clientThread` 线程；若 `bluetoothReceiver` 收到的是诸如连接断开的异常广播，则采取相应措施。

此部分使用 Java 代码可简单表示如下:

```

1 public class bluetoothReceiver extends BroadcastReceiver {
2     @Override
3     public void onReceive(Context context, Intent intent) {
4         String action = intent.getAction();
5         String targetName = "hc-bluetooth";
6         if (action == ACTION_FOUND) {
7             currentDevice = getDevice();
8             if (currentDevice.getName() == targetName) {
9                 CONNECT_STATE = true;
10                new clientThread.start();
11            }
12        } else if (action == ACTION_DISCOVERY_FINISHED) {
13            if (!isConnected()) {
14                print搜索结束, 没有找到设备("");
15            }
16        } else if (action == ACTION_DISCOVERY_STARTED) {
17            print( 搜索开始 "");
18        } else if (action == ACTION_ACL_DISCONNECTED) {
19            CONNECT STATE = false;
20            print连接断开, 正在重试连接("");
21            refreshUI();
22        }
23    }
24 }

```

在 ClientThread 线程中, Android 手机将使用 UUID 与蓝牙设备建立一条 RFCOMM 信道, 并且开启 connectThread 线程。之后, connectThread 线程将建立 InputStream 变量持续接收 byte[] 类型的数据, 直至 RFCOMM 通道被关闭。由于

InputStream 带有缓存, 因此高速传输时数据不会丢失。该线程每接收 2byte 数据时调用 handleBtData 处理一次(将数据由 byte 类型补码形式转化为 int 类, 并插入数据库)。

3.3.3 数据库管理模块的设计与实现

为方便各类之间数据传输, 所测量数据的格式是经过专门设计的。首先创建 MeasuredData 类作为所得数据的基类, 再创建 EcgData 类继承 MeasuredData, 实现 ECG 数据的存储。两种类的关系如图 3-6 所示。

其中 ECG 专用数据结构在其父类的基础上增加了自动建立时间戳、标明数据传输速率的功能。在 EcgData 的构造函数中, 通过数据的序号和传输速率计算出时间戳并赋值

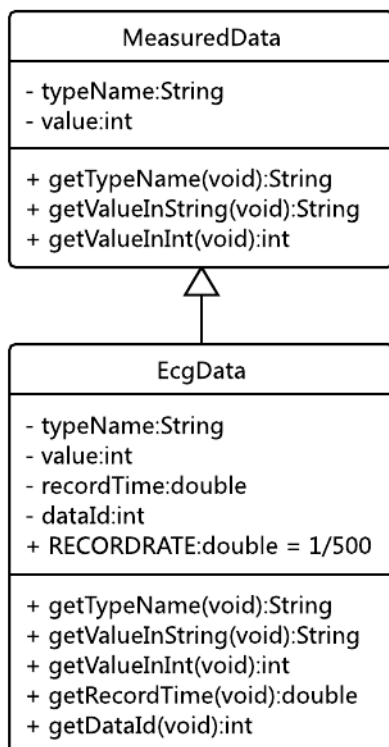


图 3--6: ECG 数据类的 UML 类图

给 recordTime。若在接收数据时获取系统时间并加盖时间戳,则由于系统后台程序等原因,时间的准确性无法保证。实验证明,以 500Hz 传输数据时,数据接收缓存中大约会积压 50 个左右待处理的数据,当之后为这些数据加盖时间戳时,获取的时间已经迟于真正接收它们的时间了(延迟时间约为 140ms,足以影响心率的判定)。而使用推算的方式能够保证时间戳的正确。

当一个 ECG 数据被蓝牙模块接收后,主线程会调用 EcgData 构造函数创建一个 EcgData 类的变量并调用 EcgDatabaseManager 类的成员函数将其插入数据库。该类主要负责数据库的管理和数据的输入输出工作,这些工作均通过对 SQLiteDatabase 类和 SQLiteOpenHelper 类的操作实现。当应用程序打开时,该类的构造函数会检查数据库的有效性,并创建或打开数据库 ecg.db。相关代码如下:

```

1 public HealthDatabaseHelper(Context context) {
2     super(context, DATABASE_NAME, null, 1);
3 }
4
5 @Override
6 public void onCreate(SQLiteDatabase db) {
7     db.execSQL("CREATE TABLE IF NOT EXISTS ecg" + "(_id
8         INTEGER PRIMARY KEY " + "AUTOINCREMENT, value INTEGER
9         , time REAL)");
10    //empty the ecg table
11    db.execSQL("DELETE FROM ecg");
12 }
13
14 @Override
15 public void onOpen(SQLiteDatabase db) {
16     super.onOpen(db);
17 }
18 ...
19 ecgDatabase = healthDatabaseHelper.getWritableDatabase();

```

对于数据的插入,使用单独的线程实现,防止阻塞主线程。dataInsertingThread 线程即为此实现。通过调用 SQLiteDatabase.execSQL() 方法能够以字符串的形式执行 SQLite 命令。插入数据的命令为 INSERT。在指定表名、数据类型和插入的数据后,提交本次事务数据即可存入表中。但实验表明,以 500Hz 接收数据时若为每条数据的插入都提交一次事务,将造成短时间内大量创建 dataInsertingThread 线程,仍然会导致阻塞,甚至引发 ANR 错误,导致程序崩溃。因此在 EcgDatabaseManager 类中实现数据缓存,可容纳 500 个 EcgData 类。当接受 500 个数据后,EcgDatabaseManager 创建线程,将这些数据作为一次事务提交,解决了线程阻塞的问题。经过修改后的提交事务可使用 Java 代码简要表示如下:

```

1 ecgDatabase.beginTransaction();
2 try {

```



```

3 |     for (int cnt = 0; cnt < 500; cnt++) {
4 |         insertData();
5 |     }
6 |     ecgDatabase.setTransactionSuccessful();
7 | } finally {
8 |     ecgDatabase.endTransaction();
9 |     ecgDataTemp[0] = newEcgData;
10| }

```

数据的输出使用 `Cursor` 类遍历当前数据库中的 `ecg` 表获取数据，并输出到 SD 卡中的某个纯文本文件中。将 `Cursor` 指向数据库后，创建 `DataOutputThread` 线程输出数据。此处使用 `Cursor` 的过程不算数据库事务，不需要调用 `beginTransaction()`。`Cursor` 类含有移动和获取数据的方法。通过这些方法获取到数据以后，使用 `FileWrite` 类可将数据输出。代码如下：

```

1 | if (cursor.moveToFirst()) {
2 |     while (cursor.moveToNext()) {
3 |         int ID = cursor.getInt(0);
4 |         int value = cursor.getInt(1);
5 |         double dateTime = cursor.getShort(2);
6 |         fileWriter.write(Integer.toString(ID) + "\t" + new
7 |             DecimalFormat("0.000").format(dateTime) + "\t" +
8 |             Integer.toString(value) + "\r\n");
9 |         bufferedFileWriter.flush();
10|     }
11| }

```

其中第 6 行的代码使用 `DecimalFormat` 方法将数据格式化，便于阅读。在启动输出线程前，调用 `Environment.getExternalStorageState()` 方法检查 SD 卡的可用性。在输出线程中，当调用 `cursor.moveToFirst()` 时，`if` 语句保证了数据的第一项是存在的；当调用 `cursor.moveToNext()` 时，`while` 语句保证了 `cursor` 将要指向的下一条记录是可用的，同时也保证了遍历结束时跳出循环。当 `cursor` 使用结束时，调用 `cursor.close()` 方法关闭游标保证了下次数据库的正常访问也节省了内存空间。

3.3.4 数据分析模块的设计与实现

数据分析模块通过创建 `EcgDataAnalyzer` 类实现，主要以接收的数据为基础计算出 ECG 信号 R 峰出现的位置，进而得出心率与 RR 间期值，并通知主线程显示在屏幕中。

数据分析模块通过创建处理线程 `BeatRateAndRpeakDetectionThread` 分离数据分析操作和主线程的更新操作。由于接收到的数据以 80 个为一组接受处理，因此有可能出现两个线程并行处理的情况，此时，为了防止两个线程对同一关键变量（如本算法中的局部最大值变量 `rpeakMaxTemp`）的并发访问，需要对相应变量进

行上锁。因此数据分析模块中创建新的临时变量类 `DataTemp`，`DataTemp` 中有关数据访问的方法——`getData()` 和 `setData()`——的属性均设置为 `synchronized`，保证了 `DataTemp` 类数据访问的安全性，确定了算法的正确执行。

当收到数据后，主线程调用 `EcgDataAnalyzer.beatRateAndRpeakDetection()` 方法使用数据分析模块处理类。数据分析模块内置长度为 80 的 `DataTemp` 类数组作为缓存，对数据进行存储和预处理（包括类型转换等）。收到 80 组数据后，数据分析模块创建处理线程。当线程检测到 R 峰出现时，调用 `EcgDataAnalyzer.rpeaksHandling(DataTemp recentRpeak)` 方法计算心率及 RR 间期，同时向主线程发送检测到 R 峰的消息，相关数据将传递给显示模块，在主界面显示心率与 RR 间期。该过程的 Java 代码可简单表示如下：

```

1 public void beatRateAndRpeakDetection(EcgData ecgData) {
2     int numberInTemp = (ecgData.getDataId() + 1) % 80;
3     if (numberInTemp == 0) {
4         tempSequence[79] = ecgData
5         beatRateAndRpeakDetectionThread.start();
6     } else {
7         tempSequence[numberInTemp - 1] = ecgData
8     }
9 }
10 ...
11 private class BeatRateAndRpeakDetectionThread extends Thread
12 {
13     @Override
14     public void run() {
15         detectRpeak(tempSequence);
16         if (rpeakDetected)
17             rpeaksHandling();
18     }
19 }

```

3.4 主线程与各模块的通信

通过对各个模块中方法的调用，主线程 `MainActivity` 得以传递数据、获取信息并在适当的时间将用户所需信息合理的显示出来。在 `Android` 中，信息的传递主要通过 `Handler` 类型变量和 `Message` 类型变量配合实现：主线程将 `android.os.Handler` 类型变量传递给各模块中，各模块将需要传递的数值以 `int` 型或列表的形式封装在 `Message` 中并设置信息类型，之后调用 `Handler.sendMessage(Message)` 方法传递给主线程即可。

本程序中，主线程需要处理的信息类型如表3--1所示。以上信息由主线程中的 `Handler` 接收后，根据类型标记筛选，主线程做出对应的决策。这种类型标记通过给 `Message.what` 成员变量赋值的方法设置，经 `switch` 语句处理能够筛选。该过程

表 3-1: 主线程处理的信息类型及处理方式

信息类型	信息作用
蓝牙断开通知	通知主线程蓝牙设备断开, 此时主线程更新菜单中的设备信息。
蓝牙连接通知	通知主线程蓝牙设备已连接, 此时主线程更新菜单中的设备信息。
数据接收通知	通知主线程蓝牙模块接收到数据, 此时主线程将数据交由数据库模块和数据分析模块处理。
R 峰峰值检测通知	通知主线程已检测到 R 峰出现, 此时主线程将心率与 RR 间期数据交由显示模块显示数据, 主线程更新主界面中的信息栏。

可以 Java 代码简要表示为:

```

1 public android.os.Handler uiRefreshHandler = new
2     android.os.Handler() {
3     @Override
4     public void handleMessage(Message msg) {
5         switch (msg.what) {
6             case 0:// Bluetooth online
7                 refreshUI();
8                 break;
9             case 1:// Bluetooth offline
10                refreshUI();
11                enableBluetooth();
12                break;
13             case 2:// get a new data
14                drawPoint();
15                insertIntoDatabase();
16                dataAnalyze();
17                break;
18             case 3:
19                BeatRateDisplay();
20                break;
21         }
22         super.handleMessage(msg);
23     }
24 };

```

3.5 本章小结

本章首先归纳了从医学诊断的角度出发, ECG 检测仪产品的用户需求, 总结出包括波形显示、波形测量、特征检测、数据存储与输出等多个方面的需求。接着本章根据这些需求设计出应用程序具体结构, 并且对这些结构的作用和关系进行了解释。在第三小节, 介绍了显示、蓝牙传输、数据库、数据分析四个模块的设计和实现方法, 并着重介绍了一些细节的实现方式, 而这些细节如何实现则决定了应用最终的运行效果。最后, 本章介绍了各种通信类型的含义和主线程对它们分别是是如何反馈的, 以此对各个模块与主线程之间的通信进行了详细的阐述。

本章所介绍的工作既是本研究的重点, 也代表了本研究的进行过程。对于一款

应用来说,从需求分析到最终的实现,所选择的实现方式是否遵从初始的结构规划在某种程度上决定了应用最终的运行效果。本研究中的应用实现过程严格遵循模块化、低耦合、不阻塞主线程的初衷(也是 Android 开发和面向对象设计的原则),最终较好地实现了预定功能。

4 R 峰检测算法及其实现

4.1 相关研究中的算法实现

作为 ECG 信号处理中重要的一部分，R 峰的检测算法虽然简单但也备受关注。相关研究往往将检测 R 峰作为信号处理的第一步，进行详细的阐述、实现和测试。

R 峰指 QRS 波群中 R 波的出现，根据 R 峰峰值、采样速率及两次 R 峰出现的间隔时间等数据，可计算得出信号的心率、RR 间期、心率波动情况等特征，同时 R 峰的检测也有助于对 QRS 波的定位。现有研究中涉及 R 峰检测的算法不胜枚举，其中大多数算法基于差分阈值法——即波形的斜率超过某一阈值则判定为出现 R 峰。同时为了减少各种干扰（肌电干扰、基线漂移等），这些算法均对信号进行了预处理。如 [16] 中采用四层连续小波变换和差分阈值法结合的方式确定 R 峰的出现；[17] 中使用带通滤波、时间平均、平方等预处理后，判断信号是否两次穿越基准点的方法；[18] 中创新性地使用了一种经过滤波后求取两次差值的方法。虽然这些算法不尽相同，但容易归纳出它们的共同点：这些方法均针对 R 峰最显著的特征，即幅值突变。因此，这些方法本质上都属于对信号斜率的筛选。

以上算法均通过使用不同类型的 ECG 信号库验证了准确率，具有一定的稳定性和可靠性。但信号的突然抖动、基线漂移（主要体现在研究 [17] 中）等异常可能影响这些算法的检测。因此本研究尝试实现一种更加稳定的 R 峰检测方法。

4.2 基于局部最大值的 R 峰检测算法

考虑到 R 峰的出现伴随着波形幅值的突变，通过检查 ECG 波形中的最大值，必然可以得到 R 峰。以此为基础设计 R 峰检测算法。首先使用长度为 10 的均值滤波器滤波，得到平滑的图形。该过程可表示为：

$$e_{avg}[j-5] = \frac{\sum_{j=6}^n e[j-5] + e[j-4] + \dots + e[j+4]}{10} \quad (4.1)$$

其中 $e_{avg}[n]$ 表示经过滤波的信号序列，表示原始信号序列。此处理使得波形更加平滑，易于分析。之后检测信号中第一次 R 峰的出现：跳过信号序列前 0.4s 的数据后，选取长度为 80 的菲重叠窗口并取此窗口中信号的最大幅值，将其与上次窗口中的最大值比较，将较大值存入最大值缓存中，并将较大值对应窗口的数据平均值存入平均值缓存中，即：

$$\begin{aligned} t_{\max} &= \max(e_{\max}[i], e_{\max}[i-1]); \\ t_{\text{avg}} &= \sum_{i=0}^{79} e_{\text{avg}}[j+i]/80 \end{aligned} \quad (4.2)$$

其中 t_{\max} 表示最大值缓存, $e_{\max}[i]$ 表示滑动窗口取得的最大值序列, t_{avg} 表示平均值缓存。判断每次窗口中的数据是否满足:

$$\frac{t_{\max}}{t_{\text{avg}}} > 1.2 \quad (4.3)$$

若上述条件满足, 则判定第一次 R 峰出现, 记录下该次峰值, 平移窗口进行后续判断。若不满足, 则平移窗口, 继续检验, 直到出现满足条件的 R 峰。检测到第一次 R 峰出现后, 剩余的步骤主要判断新出现的最大值是否与第一次 R 峰类似:

$$\left| \frac{t_{\max} - t_{\text{avg}}}{l_{\max} - l_{\text{avg}}} - 1 \right| < 0.5 \quad (4.4)$$

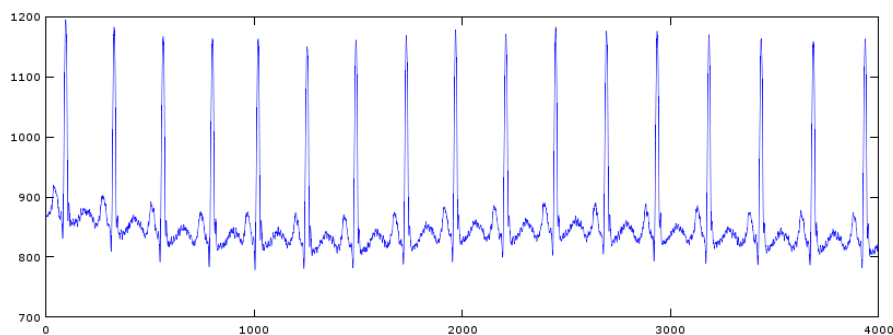
其中 l_{\max} 和 l_{avg} 分别代表最近一次窗口中取得的最大值与平均值。当上述条件满足时, t_{\max} 中的值替换为 l_{\max} 的值, l_{\max} 所在的点记录为一次 R 峰出现。继续这一步骤, 直到整段信号分析完毕。至此, 所有的 R 峰峰值及其出现时间均被记录, 通过以下算法计算 RR 间期及心率:

$$\begin{aligned} RR &= R_{tl} - R_{te} \times f_{\text{trans}}; \\ B &= \frac{RR}{60}; \end{aligned} \quad (4.5)$$

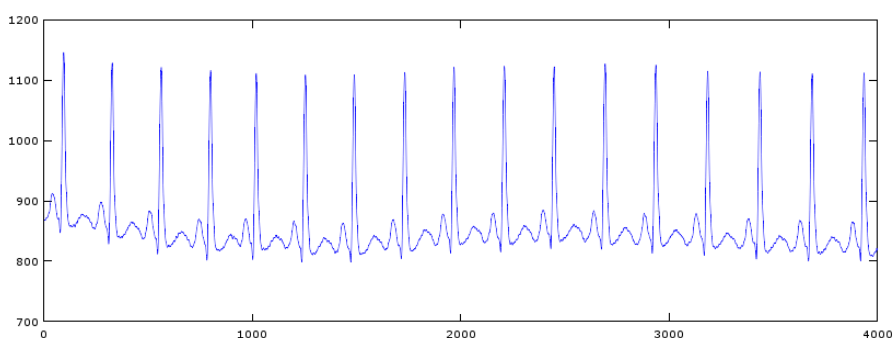
其中 RR 代表 RR 间期, R_{tl} 代表后一次 R 峰出现时间, R_{te} 代表前一次 R 峰出现时间, f_{trans} 代表数据的传输速率, B 代表心率。

分析4.3和4.4可看出, 由于该算法比较了同一窗口中最大值与平均值的差值, 而平均值随窗口的移动而波动, 因此基线漂移所造成的 R 峰异常增高并不会影响到算法的判断——评判标准是基于“相对的”最大值, 而不是“绝对的”。同时, 由于第一次出现的 R 峰被采用为随后检测 R 峰的基准, 那么之后的一些异常情况造成的突变也可以被避开。

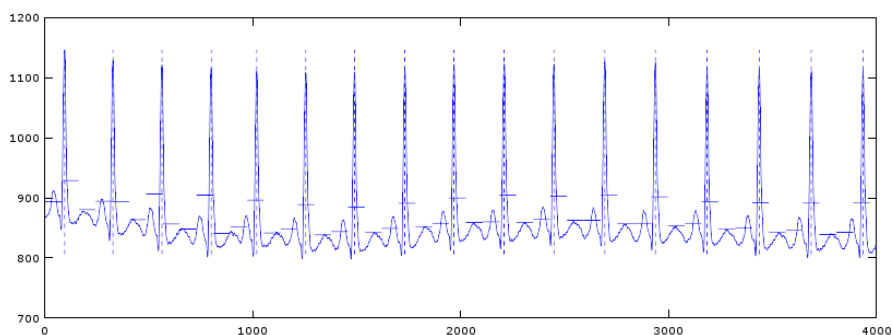
图4--1分别展示了经过预处理和 R 峰探测后的 ECG 波形。可以看出, 经过了平均滤波的波形明显更加平滑, 同时基线漂移并没有影响到 R 峰的判定, 因为每



(a) 数据源



(b) 经过平均滤波的波形



(c) 算法识别出的 R 峰

图 4-1: ECG 波形图的处理过程

个窗口的平均值是随基线波动的。

本算法并未涉及小波变换等数据分析方法，在嵌入式设备等计算资源不那么丰富的平台上也能轻松运行；同时，算法的迭代次数并不高。因此该算法具有较理想的时间和空间复杂度，容易在 Android 平台实现。

4.3 R 峰检测算法的 Java 实现

如前所述，在 `EcgDataAnalyzer` 类中实现成员类 `BeatRateAndRpeakDetection-Thread` 线程和长度为 80 的数据分析缓存，当缓存空间满时启动线程，分析数据。在同一个窗口的处理中，单次 for 循环即可实现平均数和最大值的求取，其代码

片段如下:

```

1 for (cnt = 0; cnt < 80; cnt++) {
2     dataAvgValue += dataTemp[cnt].getData();
3     if (rpeakLocalMax.getData() < dataTemp[cnt].getData()) {
4         rpeakLocalMax = dataTemp[cnt];
5     }
6 }
7 dataAvgValue = dataAvgValue / 80;

```

其中数据的获取方法 `getData()` 经 `synchronized` 关键字修饰, 可防止两次 R 峰检测对最大值缓存的并发访问, 也因此防止了某几次耗时稍长的检测影响下一次检测。至此, 信号的预处理完成。接下来线程进行 R 峰的检测。

```

1 double RRinterval = (recentRpeak.getDataId() - lastRpeak.
2     getDataId()) * EcgData.RECORDRATE;
3 beatRate = 60 / RRinterval;
4 lastRpeak = recentRpeak;
5 Message uiRefreshMessage = Message.obtain();
6 uiRefreshMessage.what = 3;
7 uiRefreshMessage.arg1 = (int) beatRate;
8 uiRefreshMessage.arg2 = (int) (RRinterval * 100);
9 uiRefreshHandler.sendMessage(uiRefreshMessage);

```

当线程检测到 R 峰出现时调用 `rpeaksHandling(DataTemp)` 方法处理数据。`DataTemp` 数据类包含了 R 峰峰值及其出现的时间, 使用前一节所述方法计算出 RR 间期与心率后, 使用 `Handler` 机制传递给主线程数据, 主线程负责稍后的显示。相关代码如上所示。

4.4 本章小结

本章从介绍现有 R 峰检测算法开始, 简述了现行大多数 R 峰检测算法, 并对这些算法进行了归纳。在第二节, 简述了所提出算法的原理, 并且展示了算法在实际应用中的效果。最后简要介绍了算法的 Java 实现过程中进行优化的部分, 也介绍了 RR 间期与心率计算的 Java 实现代码。

本章算法测试部分将在第五章介绍。

5 测试与结论

本章主要介绍对 R 峰检测算法与应用的测试，并且根据结果给出结论。

5.1 R 峰峰值检测算法的测试与结果

选取 PhysioNet 提供的 MIT-BIH 心律失常库进行测试 [19][20]。该 ECG 数据库提供了真实收集的 ECG 数据，并且提供了心率数据以供参考。通过对 10 条持续时间为 1min 的 ECG 记录进行分析，得出的结果见下表。

表 5-1: R 峰检测算法的测试结果

数据	100	101	103	105	106	114	115	116	118	122	总计
TP	74	71	70	82	66	54	63	79	71	87	717
FN	0	0	0	1	0	0	0	0	1	0	2
Se(%)	100	100	100	98.8	100	100	100	100	98.6	100	99.7
Ac(%)	100	100	100	98.8	100	100	100	100	98.6	100	99.7

其中 TP、FN 均为病理统计中的方法，Se 指敏感度，Ac 指准确度 [21]。表中 TP 指正确识别的 R 峰数量，FN 指未识别出的 R 峰数量。由数据可看出 R 峰检测算法识别准确稳定，能够达到医疗应用的要求。

5.2 ECG 监测与记录应用的运行测试

分别在 LG G3 (D857) 和小米 2S 两种 Android 设备上安装 ECG 监测与记录应用，测试应用的各项功能。蓝牙发送功能由 PC 模拟。编写 Python 程序使用串口发送一段 ECG 信号记录，两种机型均能正常接收，且能正常识别 R 峰的出现。使用“导出数据”功能能够正常生成含有 ECG 记录的 txt 文本文件，说明数据库功能也运行正常。

应用识别一个周期内 R 峰出现、计算 RR 间期及心率的过程如图 5--1 所示。可以看到，在探测到 R 峰后，应用显示出了即时的心率值和 RR 间期。同时用户可在屏幕上触摸以获取准确的 R 峰峰值。

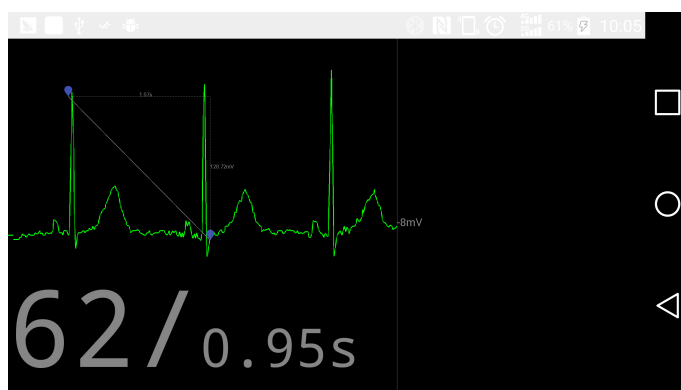
5.3 结论与展望

本研究是对 Android 平台有效展示 ECG 信号的一次尝试。综合来看，本研究所达到的结果有：

1. 提出了适用于 Android 平台的 ECG 波形显示方法，能够更有效地显示波形；



(a) 探测到 R 峰前



(b) 探测到 R 峰并测量

图 5-1: 探测 R 峰的过程

2. 提出了能够与用户交互的波形测量方法，方便波形测量；
3. 提出了一种新的、适用于嵌入式平台的 R 峰检测算法，并且能够稳定、准确地检测 R 峰；
4. 实现了基于 SQLite 的 ECG 信号数据库，方便管理。

在未来的研究中，基于 Android 的 ECG 监测记录应用应向更加实用、更加智能的方向发展。针对不同的用户，应用应能够提供不同的服务来加以适应：如针对医务工作者，应用提供更多的数据以供分析；针对患者，应用提供更加全面的监控或警报功能；针对不同年龄的用户，应用设立不同的监控标准。同时，其采用的信号分析方法也应该具有更高的效率和准确度，以确保在高配置和低配置的设备上均能够流畅运行。最后，相关应用应完成对 12 个导联的同步支持，这意味着实现更高的蓝牙串口传输速率，或更多的蓝牙设备连接。

致 谢

在这里写下你的致谢

2016 年 6 月

作者姓名

参考文献

- [1] IDC Research Inc. Smartphone OS Market Share, 2015 Q2[EB/OL]. .
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp#>.
- [2] Statista Inc. Android Version Market Share 2015[EB/OL]. .
[http://www.statista.com/statistics/271774/
share-of-android-platforms-on-mobile-devices-with-android-os/](http://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/).
- [3] YAGHMOUR K. 构建嵌入式 Android 系统 [M]. 秦云川, 肖淇, trans //. Sebastopol, CA: 中国电力出版社, 2013: 33–83.
- [4] MORENO M A. An Android hosted Bluetooth ECG monitoring device[D]. [S.l.]: University of Texas at Austin, 2012.
- [5] 万学红, 卢雪峰. 诊断学 [M] //. 8th ed. [S.l.]: 人民卫生出版社, 2013: 484–527.
- [6] TANG X, HU C, LIN W. Android Bluetooth multi-source signal acquisition for multi-parameter health monitoring devices[C/OL] // Information and Automation, 2015 IEEE International Conference on. [S.l.]: IEEE, 2015: 1790–1794.
[http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=
7279577](http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7279577).
- [7] LOU D, CHEN X, ZHAO Z, et al. A Wireless Health Monitoring System based on Android Operating System[J/OL]. {IERI} Procedia, 2013, 4: 208 – 215.
[http://www.sciencedirect.com/science/article/pii/
S2212667813000336](http://www.sciencedirect.com/science/article/pii/S2212667813000336).
- [8] GUO X, DUAN X, GAO H, et al. An ECG Monitoring and Alarming System Based On Android Smart Phone[J/OL]. Communications and Network, 2013, 05(03): 584–589.
[http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/cn.
2013.53B2105](http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/cn.2013.53B2105).

- [9] CHUDÁČEK V, ZACH L, KUŽILEK J, et al. Simple scoring system for ECG quality assessment on Android platform[C] // Computing in Cardiology, 2011. 2011 : 449 – 451.
- [10] SATIJA U, RAMKUMAR B, MANIKANDAN M S. A simple method for detection and classification of ECG noises for wearable ECG monitoring devices[C/OL] //. [S.l.]: IEEE, 2015: 164 – 169.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7095425>.
- [11] OSTER J, BEHAR J, COLLOCA R, et al. Open source Java-based ECG analysis software and Android app for Atrial Fibrillation screening[C] // Computing in Cardiology Conference (CinC), 2013. 2013 : 731 – 734.
- [12] Bluetooth SIG. Top Things to Know About Bluetooth Technology[EB/OL]. .
<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics>.
- [13] Bluetooth SIG. Serial Port Profile[K/OL]. 2012.
https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=260866&vId=290097&_ga=1.15428981.2109710189.1463300620.
- [14] Google Inc. Bluetooth|Android Developers[EB/OL]. .
<http://developer.android.com/guide/topics/connectivity/bluetooth.html>.
- [15] The SQLite Development Team. About SQLite[EB/OL]. .
<http://www.sqlite.org/about.html>.
- [16] 范晨灿. 基于蓝牙 4.0 传输的 Android 手机心电监护系统 [D]. [S.l.]: 浙江大学, 2013.
- [17] CLIFFORD G. Signal Processing Methods for Heart Rate Variability[D/OL]. [S.l.]: University of Oxford, 2002.
<http://web.mit.edu/~gari/www/papers/GDCliffordThesis.pdf>.

- [18] SADHUKHAN D, MITRA M. R-Peak Detection Algorithm for Ecg using Double Difference And RR Interval Processing[J/OL]. Procedia Technology, 2012, 4: 873 – 877.
<http://linkinghub.elsevier.com/retrieve/pii/S2212017312004227>.
- [19] GOLDBERGER A L, AMARAL L A N, GLASS L, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals[J/OL]. Circulation, 2000, 101(23): 215 – 220.
<http://circ.ahajournals.org/content/101/23/e215.abstract>.
- [20] MOODY G B, MARK R G. The impact of the MIT-BIH Arrhythmia Database[J/OL]. IEEE Engineering in Medicine and Biology Magazine, 2001, 20(3): 45 – 50.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=932724>.
- [21] Department of Statistics Online Programs. Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value[EB/OL]. .
<https://onlinecourses.science.psu.edu/stat507/node/71>.

成绩评定

毕业答辩及成绩评定说明

一、 毕业答辩

1. 答辩前，答辩小组应详细审阅每个学生的毕业设计（论文），为答辩做好准备。
2. 严肃认真组织答辩，开好答辩会。
3. 指导教师应参加所指导学生的答辩会，但评定其成绩时宜回避。
4. 答辩中要做好记录以供成绩评定时参考。

二、 成绩评定

1. 答辩前每个学生都要将自己的毕业设计（论文）在指定时间内交给指导教师，由指导教师审阅，写出评语并预评分。
2. 答辩工作结束后，答辩小组应举行专门会议进行讨论，在参考指导教师预评结果的基础上，结合学生毕业设计（论文）质量和学生答辩情况，综合评定每个学生的成绩。
3. 院（系）对专业答辩小组提出的优秀和不及格的毕业设计（论文），要组织院（系）级答辩，最终确定成绩，并向学生公布。
4. 各专业学生的最后成绩应符合正态分布规律。
5. 请用蓝、黑钢笔手写或五号宋体字编辑，签名须手写，A4 纸双面打印。

毕业设计（论文）成绩评定

班号：


学生姓名：

综合成绩：_____分（折合等级_____）

评分小组长（签名）：_____ 年 月 日

指导教师评定意见

一、对毕业设计（论文）的学术评语（应具体、确切、实事求是）



二、对毕业设计（论文）评分

(1) 理工医科评分表

评分项目	调研论证	外文翻译	设计(论文) 撰写质量	学习态度	基本理论和 基本技能	创 新	合 计
(分值)	(10 分)	(5 分)	(10 分)	(10 分)	(50 分)	(15 分)	(100 分)
得分							

(2) 文科评分表

评分项目 (分值)	文献阅读与 文献综述 (10 分)	外文翻译 (10 分)	论文撰写质量 (10 分)	学习态度 (10 分)	学术水平、论证 能力和创新 (60 分)	合 计 (100 分)
得分						

指导教师签字:

年 月 日

答辩小组评定意见

一、评语（根据学生答辩情况及其设计（论文）质量综合评价）

二、评分

评分项目 (分值)	答 辩 情 况		论 文 质 量		合 计
	答辩情况 (15 分)	回答问题情况 (25 分)	规范要求与文字表达 (20 分)	学术水平 (40 分)	(100 分)
得分					

答辩小组长签字：_____

年 月 日