

东南大学

毕业设计(论文)报告

题 目 基于 Android 智能手机的个人健康信息终端软件设计

仪器科学与工程学院 (系) 测控技术与仪器 专业

学 号 22011311

学生姓名 刘森林

指导教师 林国余

起止日期 2015 年 3 月——2015 年 6 月

设计地点 东南大学中心楼 325 室

基于 Android 智能手机的个人健康信息终端软件设计

摘 要

信息和通信技术的发展,特别是生物医学传感器、移动智能终端、云计算、3G 等技术,使得远程医疗和基于家庭的医疗关爱服务变得可行。设计一个基于 Android 平台的手机应用,检测血压等人体健康指标将会非常具有实际运用意义。移动医疗当今正成为整个移动通信产业的热点,改变了过去人们只能前往医院检测健康状况的传统医疗方式。本课题属于移动医疗的一个应用,其课题任务即通过 Android 手机和蓝牙无线技术获取电子血压计采集到的人体健康数据。本课题主要的研究内容:

- (1) 通过蓝牙无线传输数据,使用智能手机接收医疗设备的采集结果。
- (2) 使用智能手机进行采集操作,并在手机屏幕上显示采集结果。
- (3) 使用数据库保存用户的健康信息,使用户可以查看健康数据的的历史记录。
- (4) 实现用户数据上传和多用户管理功能。

关键词: Java 穿戴式设备 Android 智能手机 健康数据

Personal Health Data Acquisition Software Based on Android Platform

Abstract

The development of information and communication technologies, Particularly the biomedical sensors, mobile intelligent terminals, cloud computing and 3G technology. Making the telemedicine and home-based health care services become available. Design a mobile application based on Android platform, detecting blood pressure and other health indicators will be very practical application of significance. Mobile medical is becoming a hot spot in the entire mobile communications industry today. It changed the traditional medical approach that people only go to the hospital to detect health conditions in the past. The subject is an application of mobile medical, the subject task is through Android phones and Bluetooth wireless technology, linking to electronic sphygmomanometer to collected health data. The main research contents of this subject:

1. Using the Bluetooth wireless translate health data, use the smartphone to receive medical equipment acquisition results.
2. Using the smartphone capture portable medical equipment's operation, and displays the capture results on the phone screen.
3. Using the database to save the user's health information, it allows users to review the history of health data.
4. Implementing user data uploading and multi-user management function.

Keywords: Java, portable devices, Android Smartphone, Health Data

目录

第一章 绪 论.....	5
1.1 研究背景及意义.....	5
1.2 研究现状.....	6
1.2.1 国内研究现状.....	7
1.2.2 国外研究现状.....	8
1.3 本论文研究工作及章节安排.....	9
1.3.1 主要研究工作.....	9
1.3.2 系统配置.....	10
1.3.3 文章结构安排.....	10
第二章 系统结构.....	11
2.1 个人健康数据的构成以及采集方法.....	11
2.2 系统的总体框架.....	12
2.3 蓝牙通信.....	13
2.3.1 蓝牙通信在本设计中的实现.....	14
2.3.2 蓝牙的连接步骤.....	15
2.4 数据传输流程.....	16
2.5 通讯协议.....	16
2.6 系统运行流程.....	19
第三章 开发环境.....	20
3.1 开发平台.....	20
3.1.1 Android 系统.....	20
3.1.2 Eclipse 软件.....	21
3.1.3 Java 语言.....	21
3.2 开发环境搭建.....	22
3.2.1 安装 JDK.....	22
3.2.2 下载 Eclipse.....	23
3.2.3 安装 JDK.....	23
3.2.4 为 Eclipse 安装 ADT.....	24
第四章 程序设计.....	25
4.1 软件需求分析.....	25
4.2 数据存储.....	25
4.2.1 创建数据库.....	27
4.2.2 数据的插入、删除.....	28
4.2.3 数据查询.....	28
4.3 数据解析.....	29
4.4 蓝牙通信.....	32
4.4.1 蓝牙命令与响应.....	32
4.4.2 蓝牙的操作步骤.....	33
4.5 数据上传.....	34
4.5.1 创建服务器端.....	34
4.5.2 创建 Android 端.....	35
4.6 用户管理.....	36

4.7 界面显示.....	38
4.8 本章小结.....	40
第五章 调试与实验结果.....	41
5.1 蓝牙模块模拟调试.....	41
5.2 手机与血压计联调.....	42
5.3 本章小结.....	44
结束语	45
致谢	45
参考文献.....	46

第一章 绪 论

1.1 研究背景及意义

目前, 智能手机拥有高速数据传输能力(例如 3G、4G), 而且拥有微处理器, 使其有能力通过无线连接到外部设备(例如 蓝牙)。在国内, 智能手机的普及率越来越高, 智能手机作为最常见的移动终端, 和普通的便携式计算机相比, 具有很多实际应用的优势。例如较高的人口渗透, 增肌保密性, 购买成本更低, 跟容易运输, 在总体上增加了个人使用的便利性。手机作为一种远程或者无线监测设备已近被广泛使用, 目前有很多医疗仪器拥有移动传感器和蓝牙传输模块, 例如: 血压计、血糖仪、胎心仪。这就使得将智能手机和便携式的医疗仪器相连接成为可能。本设计的目的是可以设计一个 Android 手机应用, 连接匹配的蓝牙便携医疗设备, 通过手机的微处理器对数据进行分类和处理, 对人的健康状况进行实时的观测, 对人体健康有指导意义^{[1][2]}。

健康问题已经成为当今社会一个重要的问题, 也成为人们在日常生活中不可避免的一个问题, 尤其是心血管问题已经成为困扰很多中老年人的重大隐患。在发达国家, 很早就有人意识到健康问题的重要性, 然而中国的经济刚刚起步不久, 人们的生活条件有了大幅的改善, 随着社会的发展, 一开始一些被人们忽视的健康问题, 现在慢慢被重视起来。尤其是近几年, 关注心脑血管健康的呼声越来越高, 慢性病对人的危害越来越被重视。除了逐渐改善的医疗条件, 对自身健康状况的了解, 也是预防慢性疾病非常总要的一个方面。

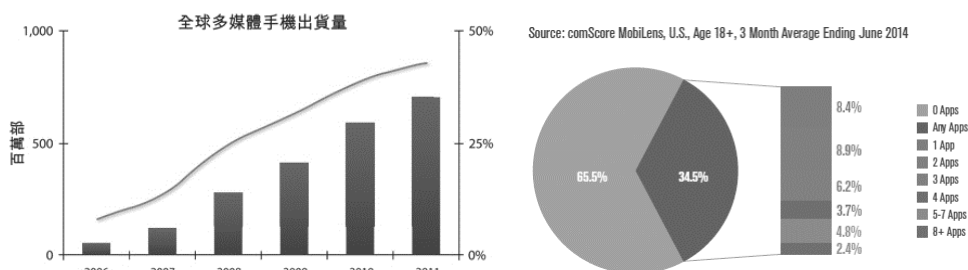


图 1.1 智能手机发展趋势

尤其是在二十一世纪 10 年代以后, 智能手机应用的发展异常迅猛, 如图 1.1。国外已近涌现出一批非常优秀的关注人体健康的手机应用。例如远程医疗的手机应用, 利用手机摄像头来进行医疗诊断的应用, 当然也有利用蓝牙技术连接便携设备的。相比于国外, 国内的应用开发起步较晚, 但是势头迅猛。尤其在个人健康领域还有待发展。

国内的智能手机行业在近几年发展迅速, 一大批智能手机厂商迈入了国际先进行列, 例如华为, 小米等厂商的出货量已经可以排在全球前列。不仅如此, 一些传统家电行业的巨头也加入了智能手机市场的争夺中来, 例如联想、海尔、海信等。也有一些互联网公司加入了争夺, 例如乐视等。而且国产智能手机行业已经形成了较为成熟的发展模式, 从研发到上市有一整套规范化的流程。除此之外, 中国已经成为了世界第一大移动终端生产国, 大量的智能手机的生产、代工甚至研发都是在国内完成^[3]。可以说, 国内的手机硬件已经发展到了一个很高的水平。虽然在核心的工艺, 例如微处理器的研发上对外依赖比较严重, 但在整体上已经有了非常不错的发展势头。相比手机硬件, 手机软件以及应用的开发还处于一个起步期, 尤其是国产智能手机的平台, 几乎是 Android 一家独大, 但是, Android 使用率高也为手机软件的通用性提供了便利, 如图 1.2 所示。

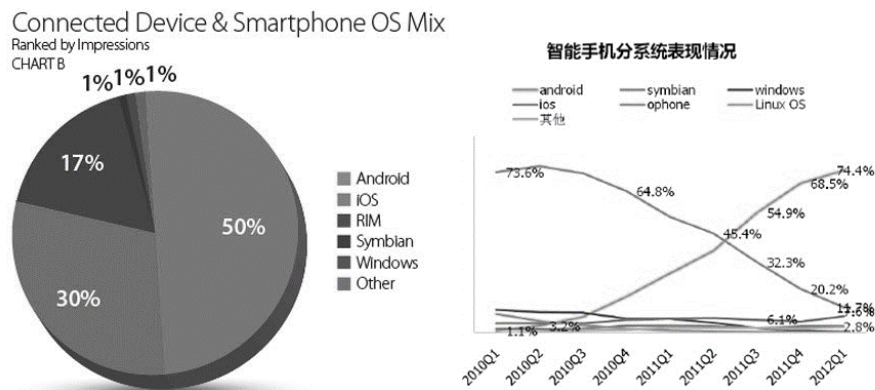


图 1.2 Android 平台的市场占有率

本课题的主要目标是基于 Android 系统开发一个人体健康数据采集系统软件。本设计有利于实时监测人们自身的健康状况，因为目前大多数心脑血管患者，临床症状不会非常明显，大多数患者也是采用保守的治疗方式，例如服用延缓性的药物，通过饮食和休息来调节血压和血糖。大多数时候人们不会去医院进行集中地治疗。由此可见，预防慢性病，有效地观察和记录自身的健康数据，对于慢性病的治疗是非常有作用的。如果可以便捷的监督自身的健康状况不仅可以及时改变自身的生活习惯以及工作节奏，也可以为医生的诊疗做出有价值的参考信息。

本设计是基于 Android 操作系统的个人健康数据采集的手机应用，将注重于简洁快速地采集健康数据，并进行有效的分析。面向具有蓝牙模块的便携式医疗设备（血压计、血糖仪），采集血压、血糖等最基本的同时也是最重要的健康信息。对于数据要求具有保密性和准确性，因此个人的智能手机是非常适合的平台。据调查，在中国市场，搭载 Android 操作平台的智能手机占到市场份额的 52%，安卓移动终端被越来越多的用户在接触^{[4][5]}。电脑上的应用程序会消耗人们的大量时间，于是，在琐碎的时间里人们越来越乐意在手机上用一些手机应用程序。基于安卓系统平台的应用程展现出非常大的市场潜力。Android 系统是一个多任务的操作系统，很好的做到实时采集，并且 Android 通过合理的优先级设置，可以达到较好的实时性。

1.2 研究现状

本设计是基于 Android 系统开发一款健康数据软件，Android 平台的特性和优点为本设计的实现提供了可能。Android 平台由操作系统、中间件、用户界面和应用软件组成。

Android 的核心系统服务真是依赖于 Linux 2.6 内核，包括内存管理，安全性，进程管理，驱动模型和网络协议栈。由于 Android 带有相当正宗的 Linux 血统，这就意味着他拥有开源 Linux 的大量好处和优点：运算速度快、安全、易于开发等。考虑到 Android 平台具有上述等优点，因此它在国内的竞争优势相当明显。青年网络的现任 CEO 潘长华曾经说：考虑到国内的开发环境，Android 作为一款十分优秀的移动开发平台，十分适合当前的开发。特别是中小企业开发团队以低成本，或者源代码的分享可以为企业节省大量资源，这对于中小企业来说无疑具有很大的吸引力。青年网络之前推出的“课程格子”，可以说是国内较为典型的低成本的 Android 平台产品^[6]。可以肯定的是 Android 已经形成一定的开发规模，并且在东南大学仪科学院已经有了相当的开发经验。

1.2.1 国内研究现状

手机健康软件的开发离不开便携式医疗器械地发展。国内近几年个人健康信息终端发展迅速，便携式、轻量化、可穿戴成为个人医疗仪器的发展方向。采集的生理指标主要包括：脉搏（心率）、血压数据（收缩压，舒张压）、心电图血糖浓度、(ECG)、体温、血氧饱和度（SpO2）以及运动参数。如图 1.4，目前有电子血压仪，快速体温计，快速心电监测仪，血糖监测仪，血样检测仪，运动分析仪等仪表等^{[7][8]}。

上海理工大学医疗器械与食品学院研发了一款基于 Android 平台的脉搏波监测系统，利用光电探测电路获取光容积脉搏波（photoplethysmography, PPG）信号，通过微处理器（MSP430）转换此信号为数字信号，然后使用蓝牙模块将数字信号发送至 Android 智能手机，实物如图 1.3；在 Android 智能手机上开发应用软件，实现对 PPG 信号的波形显示、数据存储功能和生理参数的计算^[9]。

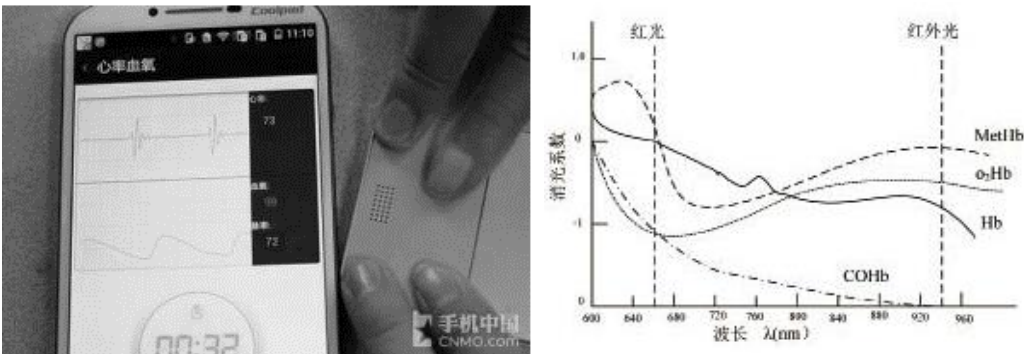


图 1.3 上海理工研制的 Android 脉搏测量系统

这种通过获取光容积脉搏波来计算人体健康数据的方法非常可行，很多的移动终端采集健康数据都是通过这种方法。这种方法可以利用手机自带的 LED 闪光灯，或者外加的光照设备，加上手机的摄像头，构成数据采集系统。

电子科技大学基于光容积脉搏波研发了一款蓝牙血压采集系统。心脏的波动导致脉搏的高低压力变化，会形成脉搏波。人体的血红蛋白和氧气做可逆结合，在光照下会长生不同的现象。通过智能手机的蓝牙，接收光电传感器的信号，可已检测患者的血氧饱和度，心率以及收缩压和舒张压^[10]。电子科大所做的系统框图如图 1.4 所示，对本设计有很大的借鉴意义。

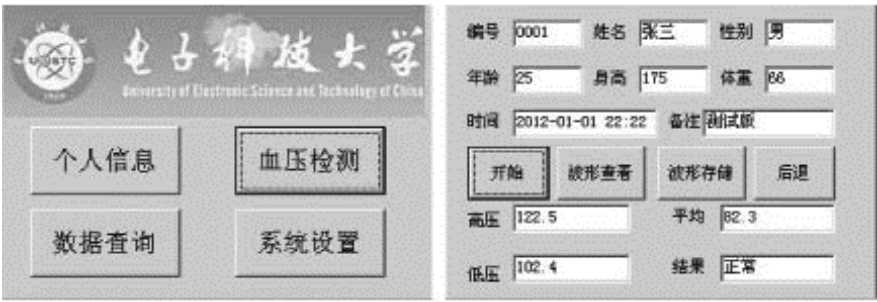


图 1.4 电子科大血压采集系统界面

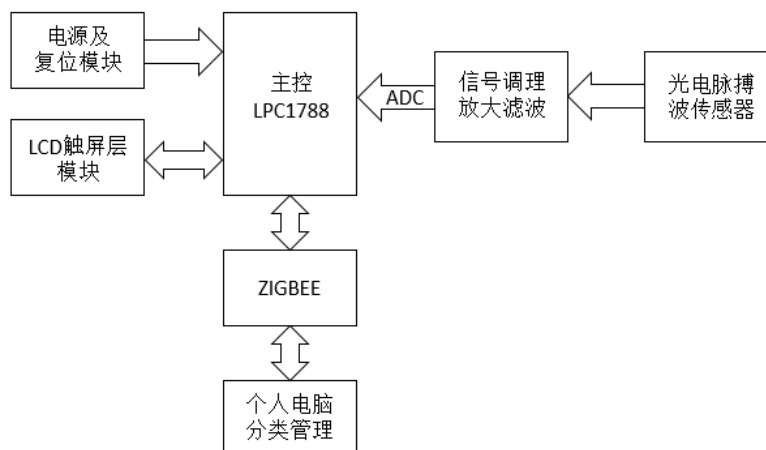


图 1.5 电子科大血压采集系统框图

医疗设备发展到现在，可穿戴医疗设备在诊断、信息监测等方面具有极大的优势，将会有很大的应用前景。目前，可穿戴医疗设备对血压、血氧、血糖等数据的监测不仅可以与智能手机或者其他移动终端相连，还可借助云存储技术，将收集的数据通过云技术进行存储和分析，并和医院的病例资源系统相连，有异常时可以及时提供警报以及做出诊治意见。不仅如此，可穿戴医疗设备作为移动互联网的新入口，利用收集到的医疗云端“大数据”，可以为用户提供个性化的远程医疗服务。前不久在博鳌亚洲论坛上，东软集团主要负责人刘积仁表示，在未来五年中，大部分人都会有可穿戴设备，不仅如此，大部分可穿戴设备都可以把相应的应用和软件移植到移动终端上，很多人可以通过手机的应用与客户或者医生进行交流和信息沟通，这是未来重要的发展趋势^{[11][12]}。

个人健康信息由便携式设备采集后，将会通过两种方式进行传输，交给采集终端或者管理平台处理。第一种也是本设计研究的是近距离传输方式。常见的有有线 USB 连接，无线蓝牙连接，或者是 ISM 频段的无线连接。另外一种就是远距离传输，也就是讲个人健康信息传输到服务器、云处理终端。远距离的信息传输通常是通过 2G(GSM/CDMA)或 3G(TD/CDMA2000/WCDMA),也可以通过 WLAN 连接至互联网。



图 1.6 个人健康信息的传输方式

本设计使用蓝牙传输个人健康数据，在短距传输中，蓝牙具有明显的优势。蓝牙协议结构简单，使用重传机制来保证链路的可靠性，在基带、链路管理和应用层中还可实行分级的多种安全机制，不仅如此，利用跳频技术，可以消除网络环境中来自其它无线设备的干扰。

1.2.2 国外研究现状

国外利用智能手机以及手机应用进行个人健康数据采集起步较早，而且具有较高的水平。调研中，了解到美国南卡罗来纳医科大学研发了一款手机应用，项目名称为：基于智能手机

的心率采集应用的开发与验证以及运用于促进健康与远程医疗。这是一个基于智能手机摄像头进行心律采集的一个应用软件，可以生成使用者的心电图。该设计运行在摩托罗拉 Droid 设备上，可以代替脉搏血氧仪的作用。著名的电子元器件生厂商 Digi-international 推出了一整套个人医疗设备，包括电子血压仪、电子血糖仪、快速心电仪等。并且在发达国家的医疗体系中，病人可以将采集得到的健康数据传输到千里之外的大医院，由著名的医生对数据进行分析 and 诊断，可以对病人的状况进行实时的监控。

最近几年，大多数智能手机都在大力推广手机健康应用软件，相关人士估计，一旦手机健康软件推广开来，到 2019 年，相关行业的产值将达到 30 亿美元。但也并非所有人都看好便携设备以及健康信息软件的发展前景。澳大利亚的一名医疗卫生专家对此并不乐观，认为目前的手机健康应用“毫无作用”。报道称，澳大利亚的西悉尼大学的乔治·马格里斯说，那些智能手机健康应用对于专业医生来说没有任何帮助，除非该软件检测到的数据，能够传送到相关的电子平台。“尽管相关产品非常流行，但是手机健康应用产业非常不成熟。”马格里斯说，迄今为止，手机健康应用还集中在一些低端的功能，如监控健康的基本数据等，但对于如何长期监控和控制疾病则无能为力。“尤其是对于一些慢性病患者来说，长期的监控和管理，对于他们是至关重要的，而这不是通过手机健康应用监测到的几个人体健康指标就可以实现的。”但我坚信随着手机应用的发展，这些问题都会一一得到解决，因此手机健康软件的研发团队应该要掌握更专业的医疗卫生知识，去深入地了解医疗卫生行业的真正需求，而不是匆匆忙忙研发出一些华而不实的软件^{[17][18]}。



图 1.7 国外基于手机的健康软件

1.3 本论文研究工作及章节安排

1.3.1 主要研究工作

本课题以智能手机和血压计为硬件基础，Java 语言为语言工具，Eclipse 软件为开发平台，充分研究了基于 Android 操作系统的智能手机个人健康信息采集终端的软件设计，设计了智能化的基于智能手机的采集系统。

未完成预期设计，本设计将在 Android 系统中开发一个 BPT 数据采集系统软件，本设计的主要研究工作如下：

- (1) 搭建 Android 开发平台，为本设计选择合适的开发版本。
- (2) 基于 SQLite 数据库存储用户的健康信息，使用户可以查询历史数据。
- (3) 将血压计返回的测试数据进行解析，从十六机制的报文中提取测试结果，以及将数据格式转换为可以存储和显示的格式。
- (4) 实现蓝牙数据通信，通过蓝牙通信进行命令和响应的传输。
- (5) 通过网络编程，实现用户数据上传。
- (6) 实现用户管理，通过用户管理实现多用户使用。

(7) 完成显示界面设计，将血压计状态以及健康数据在手机屏幕上实时显示。

1.3.2 系统配置

系统配置	参数
• 进行开发的主机操作系统:	Microsoft Windows 7
• 处理器 CPU:	Intel Core i5
• 内存:	6GB DDR2000
• 硬盘:	500G 硬盘
• 进行设计开发的平台:	Eclipse
• 进行设计开发的语言:	Java
• 设计的搭载器件:	HTC Android 智能手机

1.3.3 文章结构安排

本文共分为四个章节，各章内容安排如下：

第一章“绪论”主要介绍了课题的研究背景及意义，分析了基于 Android 智能手机个人健康数据采集系统的国内外研究现状，最后明确了本文所要研究的主要内容。

第二章“系统结构”介绍了基于 Android 智能手机的个人健康信息终端软件的整体构架和各模块组成，以及模块间是如何协调工作的。介绍了蓝牙的通信方法，以及蓝牙通信在本设计中如何实现，蓝牙的连接步骤。对本设计用到的通信协议做了说明，最后说明了本设计系统的运行流程。

第三章“开发环境”介绍了该采集软件的开发环境。首先介绍了本设计用到的平台系统 Android 系统，简单描述了 Android 系统的特点，以及选择 Android 系统的原因。最后介绍了本设计使用的 java 语言， java 作为一种完全面向对象的语言非常适合做嵌入式的开发，易于学习掌握，适合本设计使用。介绍了本设计使用的开发工具 eclipse，以及配置系统的方法，简单介绍了 eclipse 的框架。说明了 Android 软件的开发过程，介绍了 Android 开发平台的搭建方法，以及如何配置环境变量、SDK 和 ADT。

第四章“程序设计”从六个方面介绍了本设计程序部分的设计，分别为数据存储、数据解析、蓝牙通信、数据上传、用户管理以及界面显示。第四章也是本文的核心部分，详细介绍了基于 Android 系统的人体健康数据采集软件的实现过程，以及本设计各种预期功能的实现方法。

第五章“调试与试验结果”对本设计的成果进行展示，首先将手机和单独的蓝牙模块进行调试，验证本里实现的可行性。然后将血压计和手机软件进行联调，实现设计的功能。

第二章 系统结构

2.1 个人健康数据的构成以及采集方法

本设计首先以领泰医疗的的蓝牙血压计为设计的突破口,首先做出一个可以实现血压采集的系统。血压的定义:体循环动脉血压简称血压 (blood pressure, BP)。血压是血液在血管内流动时,作用于血管壁的压力,它是推动血液在血管内流动的动力。心室收缩,从而血液从心室流入动脉,此时血液对动脉的压力达到最高,为收缩压(systolic blood pressure, SBP)。心室舒张,使得动脉血管弹性回缩,血液仍慢慢向前流动,血压下降,称为舒张压(diastolic blood pressure, DBP)。

作为人体健康的重要参考指标,血压是我们着重要了解的部分。人们随着年龄的增长以及身体机能的减退,血压会呈现一个上升的趋势。见下表 2.1 (数据来源于网络)。

表 2.1 中国人平均正常血压参考值(mmHg)

年龄	收缩压(男)	舒张压(男)	收缩压(女)	舒张压(女)
16-20	115	73	110	70
21-25	115	73	110	71
26-30	115	75	112	73
31-35	117	76	114	74
36-40	120	80	116	77
41-45	124	81	122	78
46-50	128	82	128	79
51-55	134	84	134	80
56-60	137	84	139	82
61-65	148	86	145	83

目前比较主流的血压采集方法是示波法,是 90 年代发展起来的一种比较先进的电子测量方法。其原理简述如下:一开始把气囊捆在手臂上,对袖带进行充气,达到一定压力(一般会比收缩压高 30~50 mmHg)然后停止加压,开始放气,当气压达到一定程度,血流就能通过血管,并会有一定的振荡波,振荡波会通过气管传播到压力传感器,压力传感能实时检测到袖带内的压力及波动。逐渐放气,振荡波越来越大。气压再降低,由于袖带与手臂的接触越松,压力传感器所检测的压力及波动越来越小。选择波动最大的时刻为参考点,以这点为基础,向前寻找是峰值 0.45 的波动点,这一点为收缩压,向后寻找是峰值 0.75 的波动点,这一点所对应的压力为舒张压,而波动最高的点所对应的压力,被称为平均压。



图 2.1 血压的传统测量方法以及电子测量方法

2.2 系统的总体框架

人体具有复杂的生理结构，涉及的参数众多，本设计采集人体的收缩压，舒张压以及心率。在采集的过程中会实时采集血压计气囊中的气压。气囊的气压是分析血压的基础，通过心率和血压可以判断人体的一些基础的健康状况。使用本设计健康信息传输器 BPT(Blood Pressure Transducer) 可以从蓝牙血压计上接受血压、心率等健康信息，对人体的健康分析具有重要意义。本论文所研究的健康信息采集由以下几个部分组成：血压计模块、蓝牙传输模块以及手机显示模块^[13]。各个模块间的数据通讯如下图 2.2 所示：

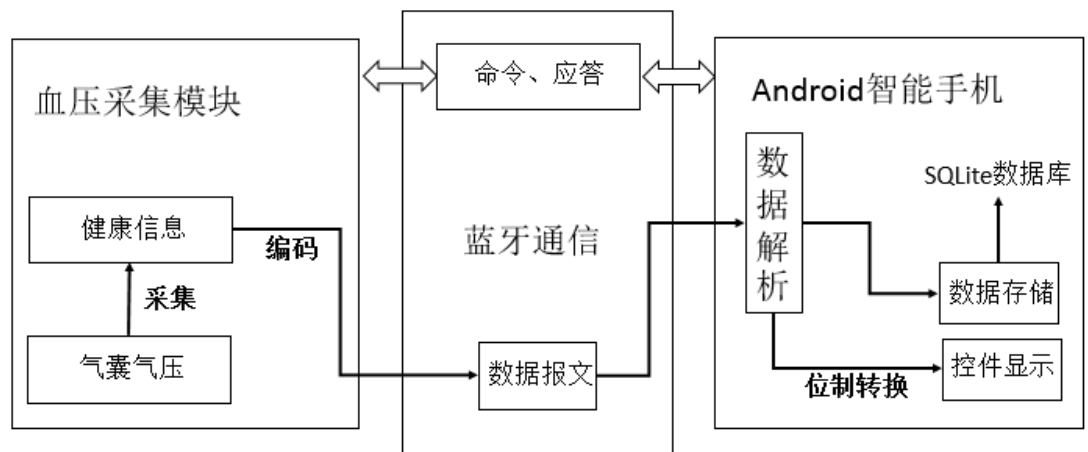


图 2.2 健康数据采集系统结构示意图

通过袖管加压感受到血管的震荡波，测出心率、收缩压以及舒张压，经过数据处理和编码器编码，将有效信息转化为十六进制编码。然后通过蓝牙通信发送至手机。手机上显示分析软件以 Android 系统为基础进行搭建。

综上所述，随着计算机技术的迅速发展，基于个人计算机控制的数据采集系统的应用日益受到重视，目前健康信息的采集系统大多是基于 PC 平台进行搭建的，利用 PC 机并行接口来进行数据采集有以下的优点：

(1) 设计严谨，设计人员只须考虑数据传输以及处理，设计人性化界面，将数据转化为有用的医疗数据，而不必花大的精力去考虑接口。

(2) 编程便利，只要输入输出语句就可以完成对多路信号的，不同精度要求的数据采集。

但是在本次设计中要求开发的系统具有携带方便、成本低、可携带性强等特点，所以基于 PC 平台的搭建并不能满足我们的需要。然而手机是人们日常生活中主要是通讯工具，这种传播方式是已经被消费者的接收产品信息，而利用手机来开发软件产品是将产品信息植于应用制作，供用户下载，通过应用达到信息传播，从而使软件得到广泛的传播。手机的应用程序中包含了一些图片、文字、视频，这样可以是用户全方位的感受产品，对于那些用户有行为差异的和不能快速了解新产品的来言具有很容易的上手性。图 2.3 基于 Android 手机的软件开发环境。

并且利用手机软件来采集数据的有以下特点：

(1) 成本低，只有软件的开发成本；而利用 PC 则是成本昂贵，所需设备复杂。

(2) 高应用，直接被目标客户受用，而且免费；而利用 PC 则客户接受性较差，很容易让客户放弃使用。

(3) 精准性，不管是提供服务的精准度还是用户的竞争性上都很有优势；而 PC 不具

有这样的竞争力。

- (4) 持续性，一旦用户下载到手机成为客户端，那么持续性使用成为必然。
- (5) 携带方便。手机设备大多以“小、轻、薄”见长，对用户来说，同样采集数据，手机比 PC 要轻许多的重量，在想要随身携带大量数据时，当然手机会是首要之选了。
- (6) 可操作性强。手机软件大多是编译好的，需要采集数据时只需点击屏幕控件即可实现；而 PC 往往需要一些复杂的代码和繁琐的设备启动。

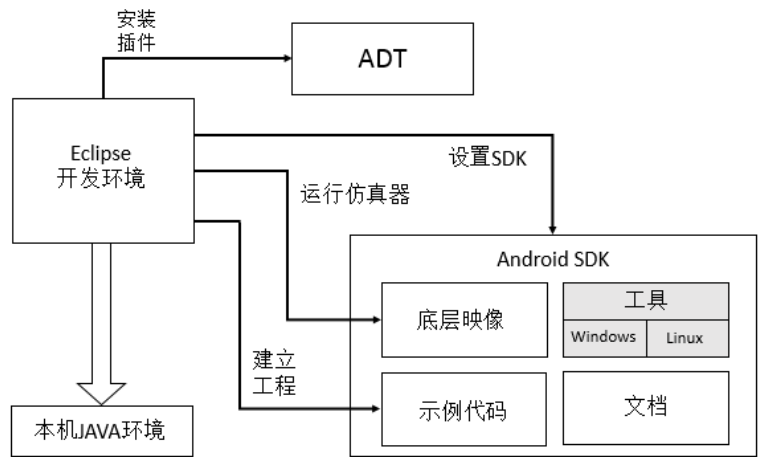


图 2.3 Eclipse 的开发环境

2.3 蓝牙通信

蓝牙是一种无线电技术，用来支持设备短距离通信（通常 10m 内）。本设计中蓝牙利用的是 RFCOMM 通道。RFCOMM 协议是一个基于 ETSI07.10 规程的串行线性仿真协议。此协议提供 RS232 的状态和控制信号，如 CTS 以及数据信号，基带上的损坏等，为上层业务提供了传送能力。RFCOMM 是一个简单传输协议，其作用是如何在两个不同设备上的应用之间保证一条完整的通信路径，并在它们之间保持一通信号。电子装置彼此可以通过蓝牙模块连接起来，摒弃了传统的电线。透过芯片上的接收器，配有蓝牙的电子产品能够在 10m 的距离内进行通信，传输速度可以达到 1Mb/s。短距离内，包括移动电话、PDA、笔记本电脑、相关外设等众多设备之间都可以进行无线信息交换。蓝牙的标准是 IEEE802.15，工作在 2.4GHZ 频带，带宽为 1Mb/s。本次设计前期调试时用到的蓝牙模块是 HC-06 系列如图 2.4 所示。在设计的初级阶段，本设计先使用了一个单独的蓝牙模块进行调试，使用计算机的串口和蓝牙连接，通过串口调试助手，先对血压计和上位机的通信协议进行模拟实验。验证手机软件的通讯编程的正确性^[14]。

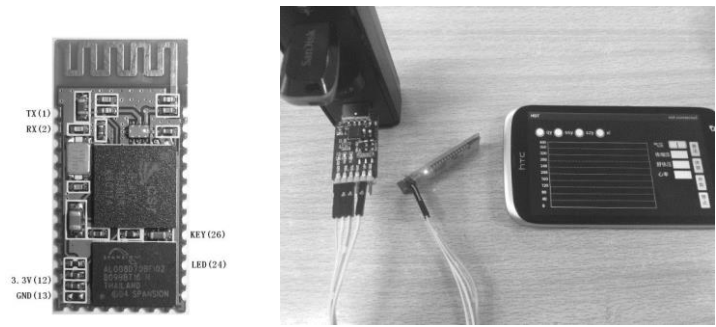


图 2.4 调试使用的蓝牙模块

蓝牙通信主要有以下的特征：

无线收发	灵敏度(误码率)达到 -80dBm , -4 -> 6dBm 功率可调输出
性能概要	<ul style="list-style-type: none">* 蓝牙 2.0 带 EDR, 2Mbps-3Mbps 调制度* 内置 2.4GHz 天线, 用户无需调试天线, 外置 8Mbit FLASH* 低电压 3.3V 工作 (3.1V~4.2V) 配对时 30~40MA 波动, 配对完毕通信 8MA* 可选 PIO 控制, 标准 HCI 端口(UART or USB), USB 协议: Full Speed USB1.1, Compliant With 2.0* 模块可以作为 SMD 贴片工艺* RoHS 制程, 引脚半孔工艺* 数字 2.4GHz 无线收发, CSR BC04 蓝牙芯片技术, 自适应跳频技术* 体积小,(27mm×13mm×2mm)* 简单的外围设计电路* 蓝牙 Class 2 功率级别* 存储温度: -40 至+85 度, 工作温度: -25 至+75 度, 谐波干扰: 2.4MHz, 发射功率 3 dBm, 误码率: 0, 但会在传输链路产生信号衰减, 才有误码, 如 RS232 和 TTL 线路处理线路中
低功耗、成本低	标准的 CLASS1 蓝牙模块发射功率为+20dbm, 即 100mw; 标准的 CLASS2 蓝牙模块发射功率<6dbm, 即小于 4mw, 蓝牙模块在查找、配对、通讯不同的过程中, 其发射功率会不同;
应用领域	<ul style="list-style-type: none">* 蓝牙车载免提* 蓝牙 GPS* 蓝牙 PCMCIA , USB Dongle* 蓝牙数据传送

表 2.2 蓝牙的主要特征

蓝牙技术在目前的设计领域大致可以有以下应用：

- (1) 代替串行线缆。当前的串行线缆可以由无线蓝牙代替。
- (2) 蓝牙与互联网技术结合。利用蓝牙的 CPU 处理能力, 可以给设备增加更多功能, 如内置的用户网络接口。
- (3) 接入点。通过接入点, 可以将数个蓝牙设备连接到传统的有线网络中, 比如 IP 网络或者工业现场总线网络(如 Controlnet、Profibus 等)。
- (4) 无线传感器和起动器。利用蓝牙来将与物理动作关系密切的设备 (例如: 传感器、数字 IO 设备、起动器和简单的模拟)连接到监控系统。

2.3.1 蓝牙通信在本设计中的实现

Android 平台支持蓝牙网络协议栈, 实现设备之间无线传输数据。利用 Android 平台提供的蓝牙 API 来实现蓝牙设备之间的通信, 主要包括了四个步骤: 设置蓝牙设备、搜寻局域网内可能匹配的设备、连接设备和设备之间进行数据传输^[15]。以下是本次设计中建立蓝牙连接的所需要的一些基本类:

- (1) BluetoothAdapter 类: 代表一个本地的蓝牙适配器, 是所有蓝牙交互的入口点。利用它你可以发现其他蓝牙设备, 查询绑定了的设备, 使用已知的 MAC 地址实例化一个

蓝牙设备和建立一个 `BluetoothServerSocket`（作为服务器端）来监听来自其他设备的连接。

（2）`BluetoothDevice` 类：代表一个远端的蓝牙设备，使用它请求远端蓝牙设备连接或获取远端蓝牙设备的名称、地址、种类和绑定状态。（其信息是封装在 `bluetoothsocket` 中）。

（3）`Bluetoothsocket` 类：代表了一个蓝牙套接字的接口（类似于 `tcp` 中的套接字），他是应用程序通过输入、输出流与其他蓝牙设备通信的连接点。

（4）`Blueboothserversocket` 类：代表打开服务连接来监听可能到来的连接请求（属于 `server` 端），为了连接两个蓝牙设备必须有一个设备作为服务器打开一个服务套接字。当远端设备发起连接连接请求的时候，并且已经连接到了的时候，`Blueboothserversocket` 类将会返回一个 `bluetoothsocket`。

（5）`Bluetoothclass` 类：描述了一个蓝牙设备的一般特点和能力。他的只读属性集定义了设备的主、次设备类和一些相关服务。

2.3.2 蓝牙的连接步骤

在这里，简单说明一下蓝牙模块的通信原理。第一，初始化，直到蓝牙模块初始化成功。然后对通信模块里面的各个端口进行定义。本次设计调试时用到的蓝牙模块为 HC-06 系列。其中的 AT91R40008 有 32 位的 I/O 口。设计时结合实际情况，考虑具体所用的端口，定义这些端口时结合其他通信模块。完成定义通信模块端口以后，立即与测试现场无线设备进行连接。随后，蓝牙接入点开始搜索现场设备，对现场设备筛选后进行配对，并对现场设备进行读写，接收蓝牙指令并更新^[16]。蓝牙通信的流程如图 2.5 以及图 2.6。

蓝牙接入点向本地自己的设备声明，然后，接收其他设备的声明。初始化结束后 5 秒钟内用来搜寻附近的设备，查找完成后，开始建立附近设备列表，并把附近设备列表内的其他设备添加到无线调度列表中。设备网络搭建完成后，蓝牙按照命令把采集到的数据传输给移动终端。本次设计中，在蓝牙模块之间建立的连接是，血压数据采集现场的血压传感器通过一个蓝牙专用的传输协议将数据传输到蓝牙接入点。数据经过转换，成为蓝牙能够识别和支持的格式，并传输到与蓝牙匹配的一个上位机中。上位机也可以通过蓝牙对数据进行实时监控。



图 2.5 蓝牙的连接步骤

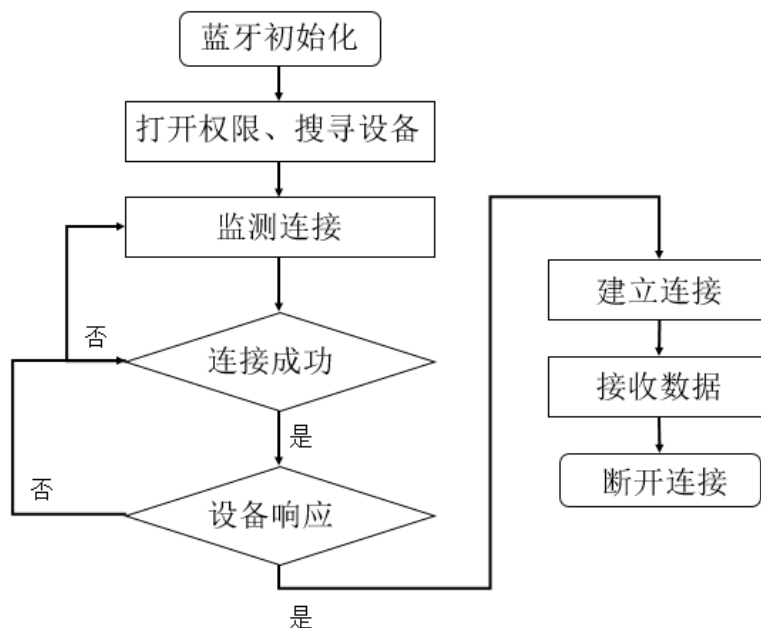


图 2.6 蓝牙通信流程

2.4 数据传输流程

为了明确数据采集系统在整个数据传输中的位置及作用，本论文对整个数据流程进行了分析，包括激活、休眠、开始采集和停止采集四种情况下的数据传输流程，以及蓝牙系统在这些流程中所需完成的工作。

(1) 激活流程。手机通过蓝牙向血压计发出激活命令，解码器对其进行解析，若命令有效，则通过蓝牙将此命令转发至相关硬件，然后在转发至数据采集模块，当采集模块收到激活命令后，执行激活功能，并向传输模块返回激活成功命令，命令响应发送到手机，手机收到激活成功的报文后，激活流程结束。此时采集模块处于激活状态。

(2) 休眠流程。当需要采集模块进入休眠状态时，手机向数据采集系统发出休眠命令，采集系统同样解析该命令，解析正确后转发至蓝牙模块，再由蓝牙模块转发给采集模块，采集模块启动休眠功能并发送成功休眠返回命令，再由蓝牙模块转发给采集装置，采集装置对命令进行解析，若接收到正确命令则转发给手机。此后手机处于休眠状态。

(3) 开始/停止采集流程。当手机通过蓝牙向数据采集系统发送开始采集命令时，命令通过蓝牙发送给血压采集系统，解析成功后采集系统开始采集血压。采集过程中，不断向手机发送气压数据，在采集结束之后向手机发送采集结果，手机在屏幕上显示采集结果。当采集模块成功接收到停止采集命令时才停止发送数据。

2.5 通讯协议

从上述流程图可以看出，上位机为了能准确控制下位机执行相关操作，需要下达不同的命令。而且数据采集系统和蓝牙传输模块，以及数据采集系统和数据显示分析系统之间分别采用蓝牙进行通讯，通讯数据量大，因此必须设定相应的通讯协议。整个通讯流程采用的通

讯协议也有所区别。本设计使用苏州领泰数字医疗的便携血压计，为臂式测量方式，具有测量精度高，操作简单等特点为。可广泛 应用于各类血压测量系统。下面简单介绍一下血压计与上位机的通讯协议。

数据显示分析系统要向嵌入式数据采集系统下达激活、休眠、开始采集或停止采集命令时，发送 5 个字节的数数据报，嵌入式数据采集系统再将该命令转发给数据传输模块，再由数据传输模块发送给数据采集模块执行以执行相关操作。

- (1) 通信内容：传感器唤醒、传感器休眠、启动测量、停止测量。
- (2) 通信：蓝牙 2.3 无线通信
- (3) 报文格式：1 个起始位、8 个数据位、1 个停止位、无奇偶校验位，波特率 115200。数据帧格式如图 2.7 所示。

帧头标识	固定为0xFF、0xCD
长度	包含长度、校验、命令、参数的字节数
校验	长度、命令、参数求和取低字节
命令及返回	
参数	
.....	

图 2.7 数据帧格式

(4) 控制命令及数据格式说明：控制命令及数据的格式总共有三种，上位机对传感器的控制命令为 5 位，传感器收到命令后将会给上位机应答，数据串长度也为 5 位。传感器激活并开始采集后，传感器向上位机发送采集到的数据。采集过程中传感器向上位机发送的气压数据包为 7 位，采集结束后，传感器会向上位机发送一串长度为 10 位的数据，包括测量结果（收缩压、舒张压、心率），具体内容如下表所示。

表 2.3 上位机控制传感器数据格式

控制命令	特征代码	DATA0	DATA1	DATA2	DATA3	DATA4
		报文头		长度	校验位	命令
传感器唤醒	0xAA	0xFF	0xCD	0x03	CKSUM	0xAA
传感器休眠	0xAB	0xFF	0xCD	0x03	CKSUM	0xAB
启动测量	0xA0	0xFF	0xCD	0x03	CKSUM	0xA0
停止测量	0xA3	0xFF	0xCD	0x03	CKSUM	0xA3

表 2.4 传感器应答上位机数据格式

传感器应答	特征代码	DATA0	DATA1	DATA2	DATA3	DATA4
		报文头		长度	校验位	命令
传感器唤醒	0x5A	0xFF	0xCD	0x03	CKSUM	0x5A
传感器休眠	0x5B	0xFF	0xCD	0x03	CKSUM	0x5B

表 2.5 采集过程传感器应答上位机数据格式

传感器应答	特征代码	DATA0	DATA1	DATA2	DATA3	DATA4
		报文头		长度	校验位	
采集过程	0X54	0XFF	0XCD	0X05	CKSUM	0X54
采集结果	0X55	0XFF	0XCD	0X07	CKSUM	0X55
		DATA5	DATA6	DATA7	DATA8	DATA9
采集过程		QY 高 8 位	QY 低 8 位			
采集结果		SSY 高 8 位	SSY 低 8 位	SZY 高 8 位	SZY 低 8 位	XL8 位

表 2.5 通讯协议内容说明：

- (1) QYH 为测量过程中气压输出的高字节，其中 QYH.4=1 表示有心跳，QYH.4=0 表示无心跳。
- (2) QYL 为气压输出的低字节。
- (3) SZYH, SZYL 为舒张压输出的高字节和低字节。
- (4) SSYH、SSYL 为收缩压输出的高字节和低字节，其中 SSYH.7=1 表示心率不齐，SSYH.7=0 表示心率正常。
- (5) XL 为心率输出的 8 位数据。

在测量过程中，如果血压传感器未能正常工作，血压计将会向上位机返回错误报告，错误报告的格式以及内容如下表 2.6 所示。0XFF 0XCD 0X04 CKSUM 0X56 X

表 2.6 错误报告格式

	特征代码	DATA0	DATA1	DATA2	DATA3	DATA4	DATA5
		报文头		长度	校验位	命令	错误参数
错误报告	0X56	0XFF	0XCD	0X03	CKSUM	0X56	X

错误报告的内容通过参数 X 反应，X 具体代表的事件如下表 2.7 所示。

表 2.7 参数 X 代表的事件

X 的值	错误报告事件
X=0	表示测量不到有效的脉搏。
X=1	表示 11 秒内打气不上 50mmHg(气袋没绑好)
X=2	表示测量结果数值有误
X=3	表示气袋压力超过 295mmHg，进入超压保护
X=4	表示干预过多（测量中移动、说话等）

2.6 系统运行流程

智能手机平台的个人健康数据采集软件设计基于 Android 操作系统，当启动手机软件时候包括四个步骤：

- (1) 软件的登录界面。它用来进行用户的登录；
- (2) 进行蓝牙搜索，与下位机的蓝牙进行匹配；
- (3) 蓝牙配对好，开始进行数据的采集和处理；
- (4) 保存数据至数据库文件中。

其中蓝牙通讯和数据的采集与处理为主线程，主线程采用的是蓝牙协议和通讯协议。运行过程中通过上位机的命令来执行相关操作，根据已经编程好的按键来完成开始采集和停止采集的工作。为避免数据混乱，读写数据的事件会打包请求主线程响应，主线程会依次处理读写命令，避免了数据的丢失。

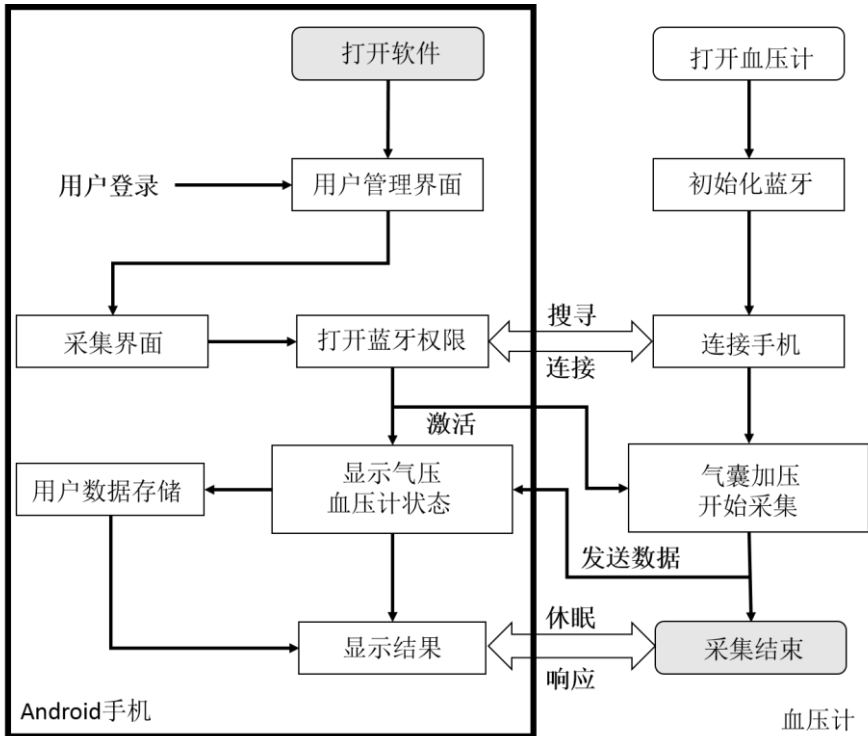


图 2.8 本设计中系统运行步骤

系统开始时，首先初始化各变量，打开登录界面，进行登录。打开蓝牙，设置蓝牙连接，接着调用蓝牙协议，使手机与下位机蓝牙连接好，然后启动蓝牙之间的线程，图 2.7 所示。监听网络命令在主线程中完成，其流程为：当接收到上位机的命令后，首先对命令进行解析，并转发该命令给蓝牙；若收到的是激活命令，则将激活标志置为 1，若是休眠命令则置为 0；若收到的是采集命令，且此时采集模块处于激活状态，则创建本地数据文件，并将采集标志置为 1；若是停止采集命令，则需要关闭对应的数据文件并将采集标志置为 0。接收的流程为：首先判断采集系统是否处于激活状态，若不是则退出此次循环；若是，则开始接收数据缓存区数据，然后判断采集标志是否为 1，若是则将采集到的蓝牙数据存储到数据库文件中；接着判断蓝牙连接是否连接上，若是再将这些数据通过蓝牙发送给手机，此时流程结束。

第三章 开发环境

3.1 开发平台

3.1.1 Android 系统

Android: 一词的本义指机器人, 是 Google 发布的基于 Linux 核心的开源移动终端软件平台。Android 平台由操作系统、用户界面、中间件和应用软件组成, 是第一个为移动终端设计的真正开放和完整的移动终端开发软件。Android 是以智能手机开发为主要的对象, 但不限于开发手机操作系统, 包括手机游戏和手机其它多种功能的开发平台。

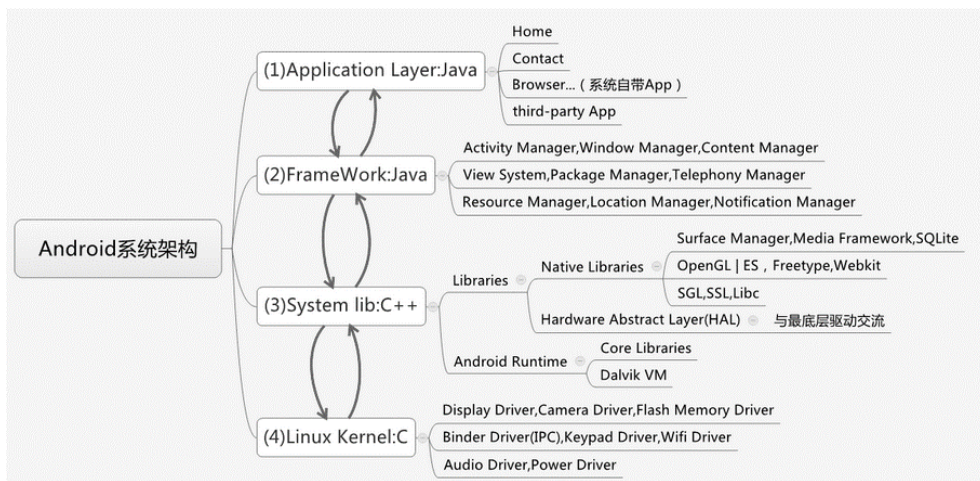


图 3.1 Android 系统架构

从宏观的角度来看, Android 是一个开放的软件系统, 它包含了众多的源代码。从下至上, Android 系统分成 4 个层次:

- 第1层次: Linux操作系统及驱动;
- 第2层次: 本地代码 (C/C++) 框架;
- 第3层次: Java框架;
- 第4层次: Java应用程序。

Android 的第 1 层次由 C 语言实现, 第 2 层是由 C 和 C++ 来实现, 第 3 层和 4 层次主要由 Java 语言来实现。

第 1 层和第 2 层之间, 从 Linux 操作系统的角度来看, 是内核空间与用户空间的分界线, 第 1 层运行于内核空间, 第 2、3、4 层次运行于用户空间。

第 2 层和第 3 层之间, 是本地代码层和 Java 代码层的接口。

第 3 层和第 4 层之间, 是 Android 的系统 API 的接口, 对于 Android 应用程序的开发, 第 3 层以下的内容是不可见的, 仅考虑系统 API 即可。

由于 Android 系统需要支持 Java 代码的运行, 这部分内容是 Android 的运行环境 (Runtime), 由虚拟机和 Java 基本类组成。

对于 Android 应用程序的开发, 主要关注第 3 层和第 4 层之间的接口。

Android 是由 Linux Kernel、Android Runtime、Libraries、Application Framework、Applications 五部分组成。Android 系统的高级编程语言是 Java, 很多 Java 和 C# 程序员用来做

基于 Android 系统的软件开发，比如基于 Android 系统的 QQ 等。Android 的上述编译语言对于我们仪器科学与工程学院的学生来说都接触过，并且 Android 系统是开源的，“开源”是用于描述那些源码可以被公众使用的软件，并且此软件的使用、修改和发行也不受许可证的限制。Android 操作系统的开源意味着开放的平台允许任何移动终端厂商加入到 Android 联盟中来。因为 Android 的开源可以使我利用开放的源代码来进行二次开发打造出个性化的 Android 软件应用。

3.1.2 Eclipse 软件

Eclipse 是著名的跨平台的自由集成开发环境（IDE），如图 3.2。主要用来 Java 语言开发，Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Kit, JDK）。虽然大多数用户很乐于将 Eclipse 当作 Java IDE 来使用，但 Eclipse 的目标不仅限于此。Eclipse 还包括插件开发环境（Plug-in Development Environment, PDE），这个组件主要针对希望扩展 Eclipse 的软件开发人员，因为它允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。

Android 开发环境目前只支持 XP(32-bit)、Win7(32/64-bit)、Mac OS X 10.4.8 或更新版本(x86)、Linux，Android 官方的 SDK 就是基于 Eclipse 的，具上陈述决定使用 Eclipse 进行开发。

- 对于本次设计 Eclipse 开发 android 程序的有下面一些优点：
- （1）Eclipse，简单。换了系统之后，里面的东西都不需要变，照样继续工作。
 - （2）Eclipse 是开源的，并且 android 也是开源的，为软件设计提供了便利。
 - （3）Eclipse 中 DDMS 透视图能控制和监控模拟器，这是目前最好的模拟器管理控制台。
 - （4）Eclipse 有强大的社区来开发插件不断的补充 Eclipse 的基本的功能。
 - （5）Eclipse 对于新手来说容易学习，上手速度，并且有广泛的教学视频和书籍。

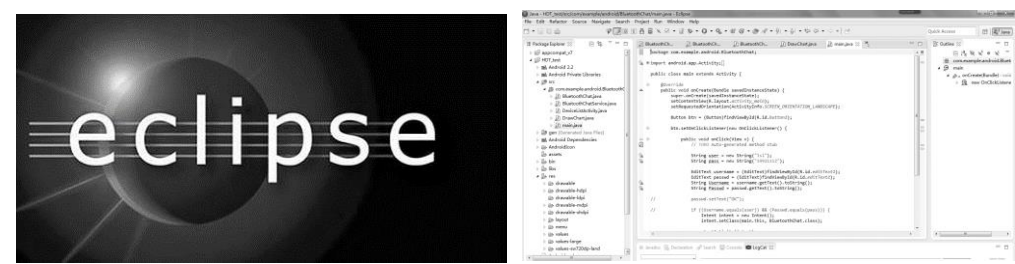


图 3.2 用于 Java 开发的 eclipse 软件

3.1.3 Java 语言

Java 是一种可以设计跨平台应用的面向对象的程序设计语言。Sun Microsystems 公司于 1995 年 5 月发布的 Java 程序设计语言和 Java 平台（即 JavaSE, JavaEE, JavaME）的合称。Java 技术具有非常好的高效性、通用性、平台移植性和安全性，非常广泛地应用于个人 PC、游戏控制中心、大型计算机、数据中心、移动终端和互联网。也拥有世界上最大的开发者专业群体。

3.2 开发环境搭建

3.2.1 安装 JDK

JDK（Java Development Kit）是编写 Java 程序必备的工具，是 Java 开发工具箱，SE 表示标准版。JDK 是 Java 的核心，包含了 Java 的运行环境（Java Runtime Environment），Java 的工具和给开发者开发应用程序时调用的 Java 类库。

（1）打开网页：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>，下载 Oracle 公司的 JDK，选择自己系统对应的版本，如图 3.3 所示。












Java SE Development Kit 8u20		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	135.24 MB	 jdk-8u20-linux-i586.rpm
Linux x86	154.87 MB	 jdk-8u20-linux-i586.tar.gz
Linux x64	135.6 MB	 jdk-8u20-linux-x64.rpm
Linux x64	153.42 MB	 jdk-8u20-linux-x64.tar.gz
Mac OS X x64	209.11 MB	 jdk-8u20-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.02 MB	 jdk-8u20-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.09 MB	 jdk-8u20-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.16 MB	 jdk-8u20-solaris-x64.tar.Z
Solaris x64	94.22 MB	 jdk-8u20-solaris-x64.tar.gz
Windows x86	161.08 MB	 jdk-8u20-windows-i586.exe
Windows x64	173.08 MB	 jdk-8u20-windows-x64.exe

图 3.3 下载 JDK

（2）配置 Windows 上 JDK 的环境变量。

为了配置 JDK 的系统变量环境，我们需要设置三个系统变量，分别是 JAVA_HOME，Path 和 CLASSPATH。下面是这三个变量的设置方法。

1、JAVA_HOME：设置这个系统变量名称，变量值为 JDK 在你电脑上的安装路径：C:\Program Files\Java\jdk。创建好后则可以利用%JAVA_HOME%作为 JDK 安装目录的统一引用路径。

2、PATH：属性已存在直接编辑，在原来变量后追加：

;%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin。

3、CLASSPATH：设置系统变量名为：CLASSPATH 变量值为：

.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar。变量值字符串前面有一个"."表示当前目录，设置 CLASSPATH 的目的，在于告诉 Java 执行环境，在哪些目录下可以找到您所执行的 Java 程序所需要的类或者包。设置环境变量如图 3.4 所示。



图 3.4 设置环境变量

3.2.2 下载 Eclipse

Eclipse 为 Java 应用程序及 Android 开发的 IDE（集成开发环境）。Eclipse 不需要安装，下载后把解压包解压后，剪切 eclipse 文件夹到预期安装的地方，打开时设置工作目录。在 <http://www.eclipse.org> 下载 Eclipse 如图 3.5 所示。

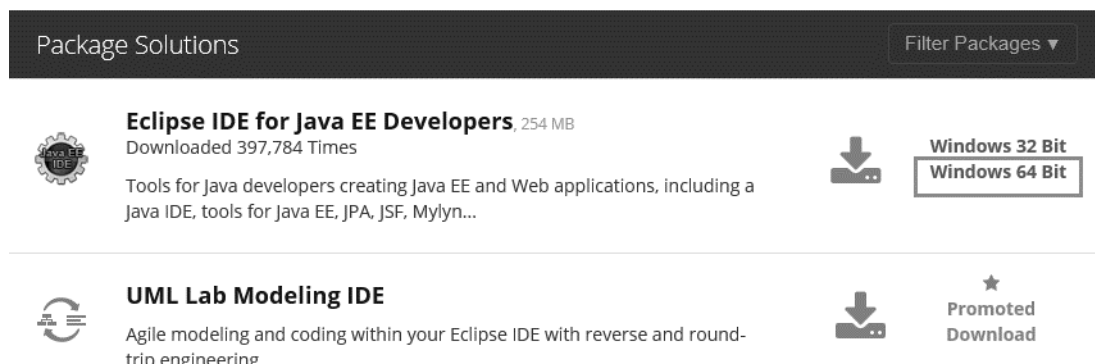


图 3.5 下载 Eclipse

3.2.3 安装 SDK

配置了 JDK 变量环境，安装了 Eclipse，已经具备了普通 Java 软件开发的环境。通过 Eclipse 开发 Android 应用程序，需要下载 Android SDK（Software Development Kit）和在 Eclipse 安装 ADT 插件，这个插件能让 Eclipse 和 Android SDK 关联起来。Android SDK 提供了开发 Android 应用程序所需的 API 库和构建、测试和调试 Android 应用程序所需的开发工具。打开 <http://developer.android.com/sdk/index.html>，选择单独下载 SDK。

打开 Android SDK Manager, Android SDK Manager 负责下载或更新不同版本的 SDK 包，选择本次设计需要使用的 Android 版本，如图 3.6 所示。

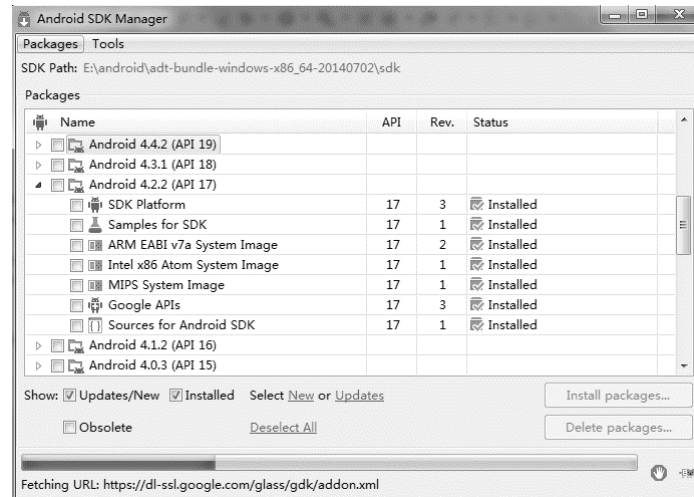


图 3.6 Android SDK Manager 下载所需的 Android 版本

3.2.4 为 Eclipse 安装 ADT

为了使得 Android 应用的创建，运行和调试更加方便快捷，Android 的开发团队专门针对 Eclipse IDE 定制了一个插件：Android Development Tools（ADT）。在 Eclipse 里在线安装 ADT 如图 3.7 所示。

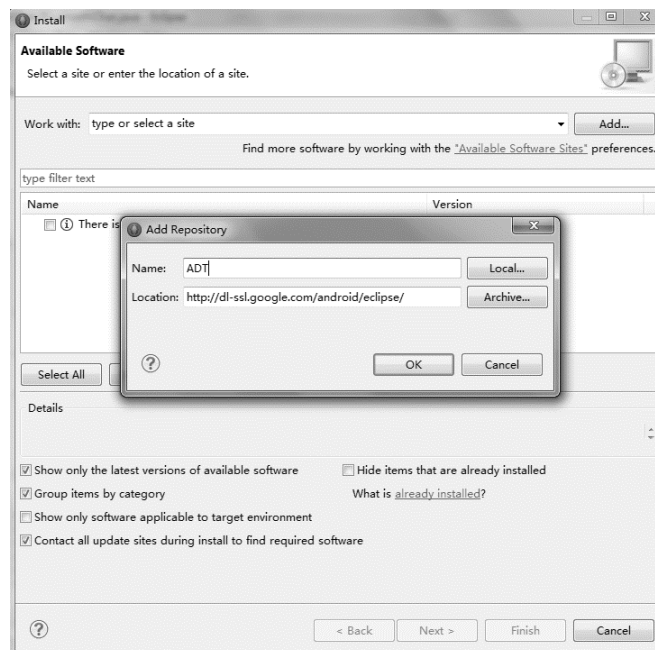


图 3.7 在线下载 ADT

第四章 程序设计

本设计以领泰医疗的蓝牙血压计为硬件基础，Java 汇编语言为语言工具，Eclipse 软件为开发平台，充分研究了基于 Android 操作系统智能手机平台的人体健康数据采集软件设计，设计了智能化的基于智能手机的蓝牙采集系统。其程序设计的主要内容包括以下几个部分：数据存储、数据解析、蓝牙通信、界面显示、用户管理和数据上传组成，如图 4.1 所示。

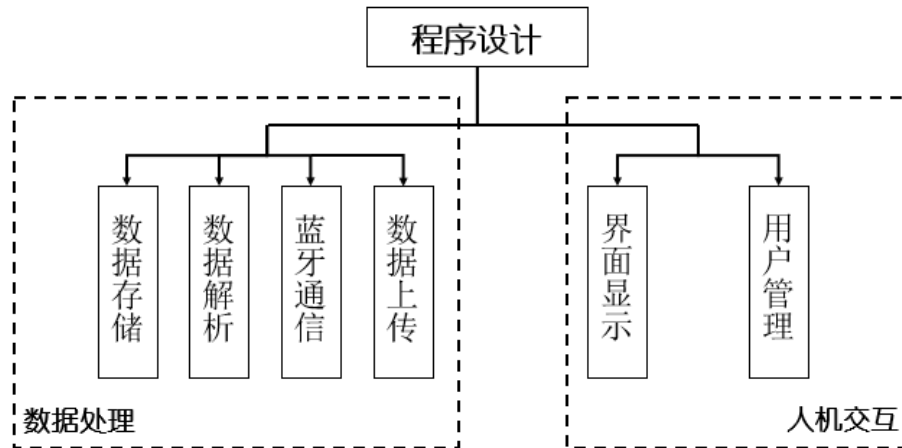


图 4.1 程序设计主要内容

4.1 软件需求分析

基于 Android 操作系统智能手机平台的人体健康采集软件要实现的功能是数据的采集以及命令转发、数据保存工作。在软件运行时，用户通过手机输入数据采集系统的激活、休眠、开始与停止采集指令，嵌入式数据采集软件向蓝牙模块转发相应的指令，并从蓝牙模块采集数据，并接收蓝牙模块返回的相应指令。同时系统将采集到在手机的 SD 卡中存储。血压传感器采集的是手臂动脉的收缩压（SSY）、舒张压（SZY）和心率（XL）。该血压传感器将血压和心率采集后，通过蓝牙输出，其输出频率为 200Hz，过程输出 7 个字节的数据，测量结束后输出 10 个字节的数据，为防止数据丢包，必须做到较好同步性和实时保存。因此，数据采集软件应满足以下要求：

- （1）能够通过命令参数实现手机（数据显示分析系统）的数据采集，采集动作启停功能（即要求能够单独完成数据采集工作）；
- （2）能通过蓝牙接收来自上位机的控制命令，并进行解析，接着转发至数据传输模块；
- （3）在手机数据库文件中存储人体健康数据数据；
- （4）将健康数据传感器数据重新组成报文通过蓝牙发送至手机，采用蓝牙协议。

4.2 数据存储

本设计采用 Android 自带的轻型数据库 SQLite，SQLite 是一款轻量级数据库，它的设计目的是嵌入式，而且它占用的资源非常少，在嵌入式设备中，只需要几百 KB。

SQLite 的特性：

(1) 轻量级：使用 SQLite 只需要带一个动态库，就可以享受它的全部功能，而且那个动态库的尺寸想当小。

(2) 独立性：SQLite 数据库的核心引擎不需要依赖第三方软件。

(3) 隔离性：SQLite 数据库中所有的信息（比如表、视图、触发器等）都包含在一个文件夹内，方便管理和维护。

(4) 跨平台：SQLite 目前支持大部分操作系统，不至电脑操作系统更在众多的手机系统也是能够运行，比如：Android。

(5) 多语言接口：SQLite 数据库支持多语言编程接口。

(6) 安全性：SQLite 数据库通过数据库级上的独占性和共享锁来实现独立事务处理。这意味着多个进程可以在同一时间从同一数据库读取数据，但只能有一个可以写入数据。

优点：一、能存储较多的数据。二、能将数据库文件存放到 SD 卡中。

SQLite 数据库存储体统的结构图如下图 4.2 所示：

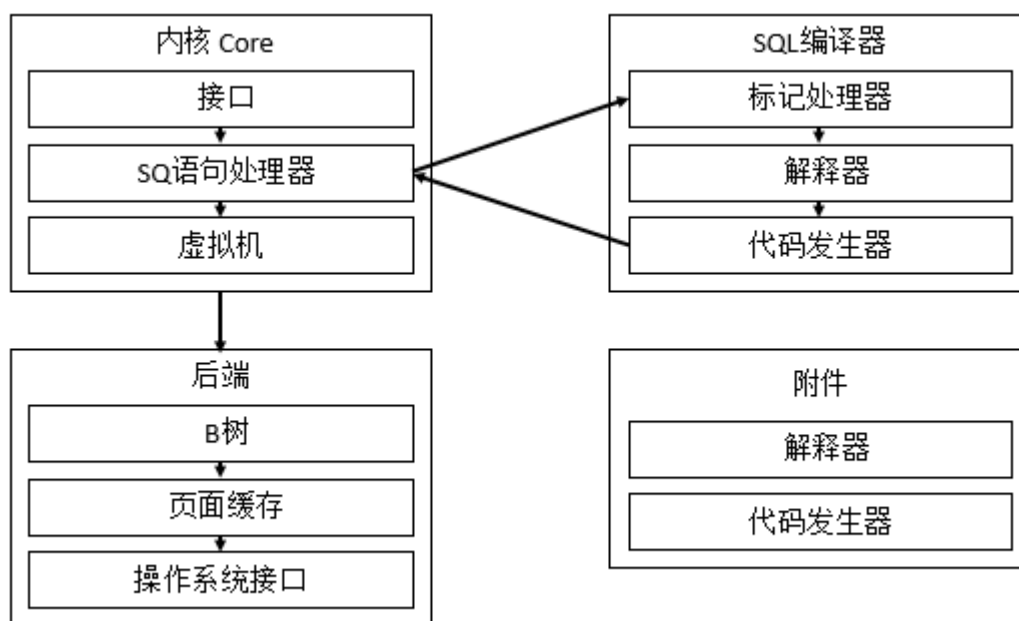


图 4.2 SQLite 数据库结构

下面介绍 SQLite 数据库里面最基本的几个类：

SQLiteDatabase 类：一个 SQLiteDatabase 的实例代表了一个 SQLite 的数据库，通过 SQLiteDatabase 实例的一些方法，我们可以执行 SQL 语句，对数据库进行增、删、查、改的操作。需要注意的是，数据库对于一个应用来说是私有的，并且在一个应用当中，数据库的名字也是惟一的。

SQLiteOpenHelper 类：根据这名字，我们可以看出这个类是一个辅助类。这个类主要生成一个数据库，并对数据库的版本进行管理。当在程序当中调用这个方法 `getWritableDatabase()`，或者 `getReadableDatabase()` 方法的时候，如果当时没有数据，那么 Android 系统就会自动生成一个数据库。SQLiteOpenHelper 是一个抽象类，我们通常需要继承它，并且实现里边的 3 个函数。

ContentValues 类：ContentValues 类和 Hash map/Hash table 比较类似，它也是负责存储一些名值对，但是它存储的名值对当中的名是一个 String 类型，而值都是基本类型。

Cursor 类：Cursor 在 Android 当中是一个非常有用的接口，通过 Cursor 我们可以对从数据库查询出来的结果集进行随机的读写访问。

4.2.1 创建数据库

(1) 创建一个数据库类 `BloodPressureDB`，创建的数据库类首先要继承 `SQLiteOpenHelper` 类，`SQLiteOpenHelper` 类是一个抽象类，主要的作用是生成数据库。继承这个类的主要目的是重写 `SQLiteOpenHelper` 类中的两个方法，`onCreate()` 和 `onUpgrade()`。当数据库首次被创建时，需要将创建表等初始化操作放在 `onCreate()` 方法中进行。当需要更新数据库时，需要调用 `onUpgrade()` 方法。

本例从 `SQLiteOpenHelper` 类继承的数据库类是 `BloodPressureDB`，使用它创建一个数据库对象，具体实现的代码如下图 4.3 所示。

```
public class BloodPressureDB extends SQLiteOpenHelper {

    private final static String DATABASE_NAME = "bp_db";//数据库名称
    private final static int DATABASE_VERSION = 1;
    private final static String TABLE_NAME = "bp_table";//表名称
    public final static String FIELD_id = "_ID";
    public final static String FIELD_USER = "_USER";
    public final static String FIELD_DATE = "_DATE";
    public final static String FIELD_DIAS = "_DIAS";
    public final static String FIELD_SYST = "_SYST";
    public final static String FIELD_ARRAY = "_ARRAY";

    public BloodPressureDB (Context context){
        super(context,DATABASE_NAME,null,DATABASE_VERSION);
    }
}
```

图 4.3 继承 `SQLiteOpenHelper` 类创建数据库

(2) 重写创建数据库所需要的两个方法。首先重写 `onCreate()` 方法，在方法中创建一张表，并且确定表的每一列的内容，包括用户名、主键、气压、收缩压、舒张压和心率。之后使用 `db.execSQL()` 执行一条 sql 语句，进行一张表的创建。然后重写 `onUpgrade()` 方法，这个方法可以更新数据库的版本号，并在这个方法里删除表和新建表，在方法的最后执行 sql 语句，进行数据库的更新，具体代码见下图 4.4 所示。

```
@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub
    String sql = "create table bp_table ( '_ID' integer primary key autoincrement, '_USER' integer, '_DATE' test, " +
        "'_DIAS' integer, '_SYST' integer, '_ARRAY' integer)";
    db.execSQL(sql);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    String sql = "DROP TABLE IF EXISTS " + TABLE_NAME;
    db.execSQL(sql);
    onCreate(db);
}
```

图 4.4 重写创建和更新方法

(3) 在主程序 `BluetoothChat.java` 中创建数据库。调用 `BloodPressureDB` 创建数据库 `bpdb`，再创建一个数据库辅助类的实例，代码如图 4.5 所示。

```
//database
private BloodPressureDB bpdb;

bpdb = new BloodPressureDB(this);
//创建数据库辅助类实例
```

图 4.5 创建数据库实例

4.2.2 数据的插入、删除

(1) 在 BloodPressureDB 类中定义插入数据和删除数据的方法，分别命名为 insert () 和 delete ()。具体实现的代码如图 4.6 所示。

```
public long insert(int user, String date, int diaval, int sysval, int arrval ) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(FIELD_USER, user);
    cv.put(FIELD_DATE, date);
    cv.put(FIELD_DIAS, diaval);
    cv.put(FIELD_SYST, sysval);
    cv.put(FIELD_ARRAY, arrval);
    long row = db.insert(TABLE_NAME, null, cv);
    return row;
}

public void delete(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    String where = FIELD_id + " = ?";
    String[] whereValue = { Integer.toString(id) };
    db.delete(TABLE_NAME, where, whereValue);
}
```

图 4.6 数据插入与删除的方法

在 insert () 方法中，ContentValues 是一个哈希表，键值是字段的名称（例如用户名、收缩压等），存入哈希表的值就是字段的值。通过 ContentValues 的 put 指令，可以将数据先放入哈希表中，再放到数据库的表中。在 delete () 方法中，参数为需要删除的数据的 ID，如果传入 null，就将数据全部删除。

(2) 将采集到的人体健康数据插入到表中。在 BluetoothChat.java 中，当采集结果读取完毕后，将采集结果插入数据库。数据库表中数据的主键（data）是当时的时间，便于历史记录的查询，并且将收缩压、舒张压和心率插入到表中。具体代码如图 4.7 所示。

```
SimpleDateFormat DateFormat;
DateFormat = new SimpleDateFormat("yyyyMMddHHmmss");
String date = DateFormat.format(new Date());
bpd.insert(001, date, diaPval, SysPval, arrval);
```

图 4.7 将采集结果插入表中

4.2.3 数据查询

(1) 使用 SQLite 数据库自带的 Cursor 接口来实现数据查询的方法。先获取一个数据库的实例，再获得查询接口 Cursor，就可以实现数据查询的方法。具体代码如下图 4.8 所示。

```
public Cursor select() {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db
        .query(TABLE_NAME, null, null, null, null, null, null);
    return cursor;
}
```

图 4.8 数据查询的方法

(2) 在历史查询界面中调用查询的方法，查询了当前表中存储了多少组测量结果，显

示在 HistoryActivity 界面上，具体代码如图 4.9 所示。

```
bpdb = new BloodPressureDB(this);
myCursor = bpdb.select();
String countNumber = Integer.toString(myCursor.getCount());
tv.setTextColor(Color.BLUE);
tv.setText(countNumber);
```

图 4.9 查询已有测量结果的数量

4.3 数据解析

血压计传送给手机的报文是以十六进制报文形式发送给手机，手机接收报文之后要对其进行数据解析，数据解析的流程如下图 4.10 所示。

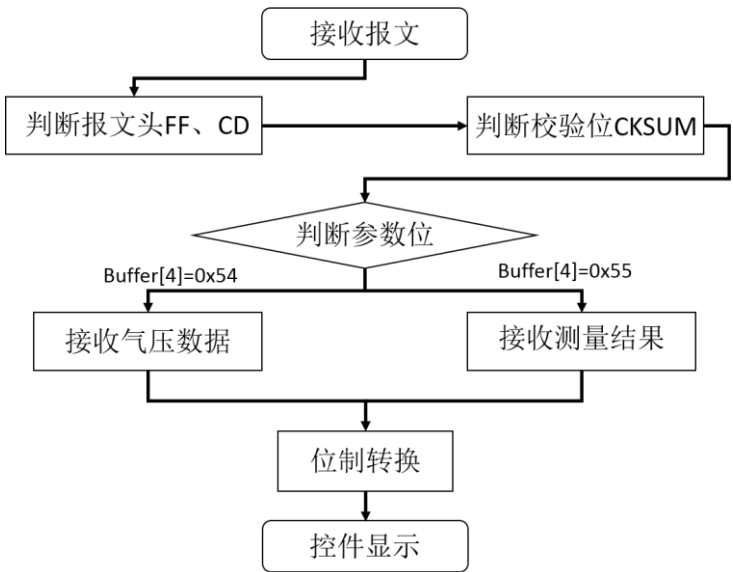


图 4.10 数据解析流程

(1) 将读取到的报文转化为字节数组，具体代码为 byte[] readBuf = (byte[]) msg.obj，然后判断报文头是否正确，正确的报文因该是长度为 7 字节，开头以 0xFF、0xCD 开始，源码如下图 4.11 所示。

```
public int BloodPressure(int j, int Index, byte[] Buffer){
    int cIndex = j;
    //Blood Pressure Test
    for(; j<(Index-1);j++){
        if((Buffer[j]==(byte)0xFF)&&(Buffer[j+1]==(byte)0xCD)){
            System.out.println("package header confirmed!");
            int t = ((int)Buffer[j+2])-1;
            if((t>7)|| (t<2)){
                System.out.println("package amount wronged!");
                continue;
            }
        }
    }
}
```

图 4.11 判断正确报文

(2) 判断校验位是否正确，本设计中的校验采用的是和校验，即校验位等于长度、命令、参数求和取低字节。校验报文校验位的代码如下图 4.12 所示。


```

} else if(Buffer[j+4]==(byte)0x55){
    System.out.println("BP test result!");
    //BP test result
    if((Buffer[j+5]&((byte)0x80))==1){
        //arrhythmia is detected
        Toast.makeText(getApplicationContext(), "arrhythmia is detected", Toast.LENGTH_SHORT);
    } else{
        //arrhythmia is not detected
        Toast.makeText(getApplicationContext(), "arrhythmia is NOT detected", Toast.LENGTH_SHORT);
    }
    //BP value
    statusET.setTextColor(Color.GREEN);
    statusET.setText(ProtocolConstants.Results);
    String SSYH = Integer.toHexString((Buffer[j+5]&0x000000FF)|0xFFFFFFFF00).substring(6);
    String SSYL = Integer.toHexString((Buffer[j+6]&0x000000FF)|0xFFFFFFFF00).substring(6);
    int SysPval = Integer.valueOf(SSYH.substring(1,16)*256+Integer.valueOf(SSYL,16));
    systolicpressureET.setText(Integer.toString(SysPval)+ProtocolConstants.Unit);
    System.out.println("systolicpressureET"+processingBuffer[j+5]+" "+processingBuffer[j+6]);
    System.out.println("systolicpressureET"+Integer.toHexString(SysPval));
    String SZYH = Integer.toHexString((Buffer[j+7]&0x000000FF)|0xFFFFFFFF00).substring(6);
    String SZYL = Integer.toHexString((Buffer[j+8]&0x000000FF)|0xFFFFFFFF00).substring(6);
    int diaPval = Integer.valueOf(SZYH.substring(1,16)*256+Integer.valueOf(SZYL,16));
    diastolicpressureET.setText(Integer.toString(diaPval)+ProtocolConstants.Unit);
    System.out.println("diastolicpressureET"+processingBuffer[j+7]+" "+processingBuffer[j+8]);
    System.out.println("diastolicpressureET"+Integer.toHexString(diaPval));
    String XL = Integer.toHexString((Buffer[j+9]&0x000000FF)|0xFFFFFFFF00).substring(6);
    int arrval = Integer.valueOf(XL, 16);
    arrhythmiaET.setText(Integer.toString(arrval)+"bpm");
}

```

图 4. 15 测量结果的接收与转换

(6)接收到的报文内容是十六进制整形,现将十六进制整形转化为十六进制的字符串,再将其转化为十进制的字符串显示出来。下面代码是以气压为例:

```

String QYH = Integer.toHexString((Buffer[j+5]&0x000000FF)|0xFFFFFFFF00).substring(6);
String QYL = Integer.toHexString((Buffer[j+6]&0x000000FF)|0xFFFFFFFF00).substring(6);
int SPval = Integer.valueOf(QYH.substring(1,16)*256+Integer.valueOf(QYL,16);

```

Buffer[]& 0x000000FF 的作用是,如果 Buffer[]是负数,则会清除前面 24 个零,正的 byte 整型不受影响。(...)| 0xFFFFFFFF00 的作用是,如果 Buffer[]是正数,则置前 24 位为一,这样 toHexString 输出一个小于等于 15 的 byte 整型的十六进制时,倒数第二位为零且不会被丢弃,这样可以通过 substring 方法进行截取最后两位即可。流程如下图 4.16 所示。

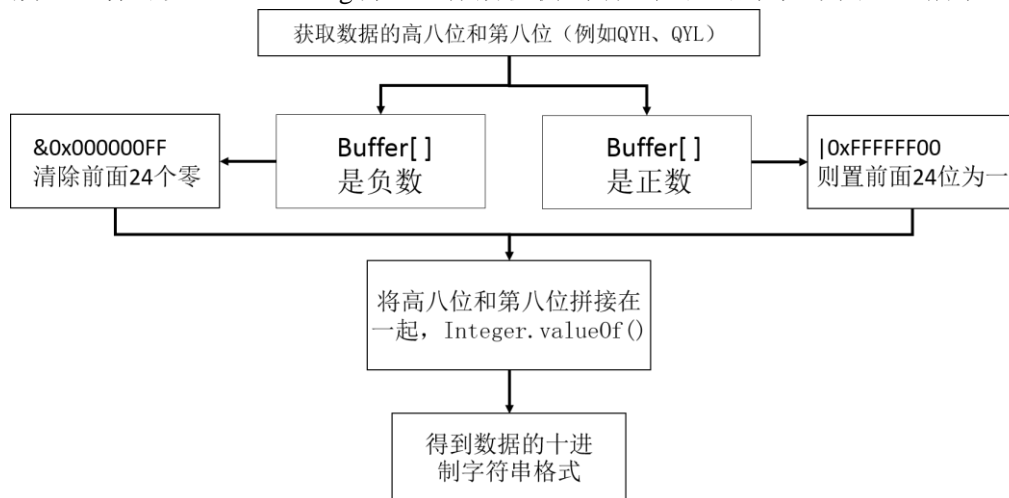


图 4. 16 取报文转换为可以显示的格式

4.4 蓝牙通信

当手机的蓝牙与血压传感器连接好以后，软件开始运行，系统开始时，初始化变量与线程。此时的系统主要存在 4 个模式。分别为激活、休眠、开始和停止。系统激活后，可以开始采集数据，当点击休眠时候，系统进入休眠模式，不进行数据的采集。当点击开始，系统会发出开始采集的命令，并且开始采集的时候系统还牵扯到十六进制整形与十六进制字符串的转换，再进行十六进制字符串和整形的转换，所以必须转换为十进制整形才能顺利的显示出来。当完成采集工作后，可以点击停止按钮，来结束程序，这样一个完整的采集流程便结束了。

4.4.1 蓝牙命令与响应

一、手机向血压传感器发送指令

指令包括激活、休眠、开始采集和停止采集。在协议常量（ProtocolConstants.java）中进行了统一的定义，如图 4.17 所示。

```
public static final byte[] wake = {(byte) 0xFF, (byte) 0xCD, (byte) 0x03, (byte) 0xAD, (byte) 0xAA};  
public static final byte[] sleep = {(byte) 0xFF, (byte) 0xCD, (byte) 0x03, (byte) 0xAE, (byte) 0xAB};  
public static final byte[] start = {(byte) 0xFF, (byte) 0xCD, (byte) 0x03, (byte) 0xA3, (byte) 0xA0};  
public static final byte[] stop = {(byte) 0xFF, (byte) 0xCD, (byte) 0x03, (byte) 0xA6, (byte) 0xA3};
```

图 4.17 命令定义

手机向血压传感器发出开始采集的指令之后，首先先向传感器发出激活传感器的指令，经过 500ms 的延迟之后，向传感器发出采集指令，将两个索引号置零 bufferIndex = 0; currentIndex = 0。然后发出采集命令，具体代码如下图 4.18 所示。

```
start.setOnClickListener(new OnClickListener(){  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        start.setEnabled(false);  
        stop.setEnabled(true);  
        sendMessage(ProtocolConstants.wake);  
        try{  
            Thread.sleep(500);  
        }catch (InterruptedException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
        bufferIndex = 0;  
        currentIndex = 0;  
        sendMessage(ProtocolConstants.start);  
        System.out.println("start");  
    }  
});
```

图 4.18 开始采集（start）

手机发出停止采集的命令，start 键变灰不可操作，屏幕输出采集结果之后，将采集结果清零，将索引号置零，结束采集，源码如图 4.19。


```

stop.setOnClickListener(new OnClickListener(){

    public void onClick(View v) {
        // TODO Auto-generated method stub
        start.setEnabled(true);
        stop.setEnabled(false);
        System.out.println("all: ");
        for(int i=0; i<bufferIndex;i++){
            String temp = Integer.toHexString((processingBuffer[i]&0x000000FF)|0xFFFFFF00).substring(6);
            System.out.print(" "+temp);
        }
        System.out.println("\r\n");

        bufferIndex = 0;
        currentIndex = 0;
        sendMessage(ProtocolConstants.stop);
        System.out.println("stop");
        stopCommand = true;
        Process();
        isProcessCompleted = false;
        pressureoutputET.setText(" ");
        diastolicpressureET.setText(" ");
        systolicpressureET.setText(" ");
        arrythmiaET.setText(" ");
        statusET.setText(" ");
    }

});

```

图 4.19 停止采集 (stop)

4.4.2 蓝牙的操作步骤

蓝牙使用的基本步骤可以归纳为 3 步：

(1) 使用蓝牙的响应权限

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

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

(2) 配置本机蓝牙模块

在这里首先要介绍这次设计中对蓝牙操作一个核心类 BluetoothAdapter

1、直接打开系统的蓝牙设置面板：

```
BluetoothAdapter adapter = BluetoothAdapter.getDefaultAdapter()
```

```
Intent intent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE)
```

```
startActivityForResult(intent, 0x1)
```

2、直接打开蓝牙：adapter.enable()、关闭蓝牙：adapter.disable()。

3、打开本机的蓝牙发现功能：

```
Intent discoveryIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE)
```

4、设置持续时间（最多 300 秒）：

```
discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300)
```

打开蓝牙的权限代码如下图 4.20 所示。

```

mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
// If the adapter is null, then Bluetooth is not supported
if (mBluetoothAdapter == null) {
    Toast.makeText(this, "Bluetooth is not available", Toast.LENGTH_LONG).show();
    finish();
    return;
}

```

图 4.20 打开蓝牙权限

(3) 搜索蓝牙设备

使用 `BluetoothAdapter` 的 `startDiscovery()`方法来搜索蓝牙设备 `startDiscovery()`方法是一个异步方法，调用后会立即返回。该方法会进行对其他蓝牙设备的搜索，该过程会持续 12 秒。该方法调用后，搜索过程实际上是在一个 `System Service` 中进行的，所以可以调用 `cancelDiscovery()`方法来停止搜索（该方法可以在未执行 `discovery` 请求时调用）。

请求 `Discovery` 后，系统开始搜索蓝牙设备，在这个过程中，系统会发送以下三个广播：

- 1、`ACTION_DISCOVERY_START`：开始搜索
- 2、`ACTION_DISCOVERY_FINISHED`：搜索结束
- 3、`ACTION_FOUND`：找到设备，这个 `Intent` 中包含两个 `extra fields`：`EXTRA_DEVICE` 和 `EXTRA_CLASS`，分别包含 `BluetoothDevice` 和 `BluetoothClass`。

搜索蓝牙设备的源码如下图 4.21 所示。

```
private class AcceptThread extends Thread {
    // The local server socket
    private final BluetoothServerSocket mmServerSocket;
    private String mSocketType;

    public AcceptThread(boolean secure) {
        BluetoothServerSocket tmp = null;
        mSocketType = secure ? "Secure":"Insecure";

        // Create a new listening server socket
        try {
            if (secure) {
                tmp = mAdapter.listenUsingRfcommWithServiceRecord(NAME_SECURE,
                    MY_UUID_SECURE);
            } else {
                tmp = mAdapter.listenUsingRfcommWithServiceRecord(
                    NAME_INSECURE, MY_UUID_INSECURE);
            }
        } catch (IOException e) {
            Log.e(TAG, "Socket Type: " + mSocketType + "listen() failed", e);
        }
        mmServerSocket = tmp;
    }
}
```

图 4.21 搜索蓝牙设备

4.5 数据上传

目前的智能手机普遍具备连接互联网的能力，将用户的健康数据上传到服务器，可以远程的了解患者的健康状况。服务器存储用户数据也就是目前兴起的云技术，如果用户数量足够多，就可以对人群的健康状况进行大数据计算。本设计简单地利用 `GET` 方式将数据上传到服务器。`GET` 上传的方式比较简单，对安全的要求比较低。本设计的数据基本以字节较小的文件，`GET` 方式上传的数据大小不高于 2KB，可以满足本设计的要求。

4.5.1 创建服务器端

- （1）首先创建服务器端项目，服务器端项目如图 4.22。

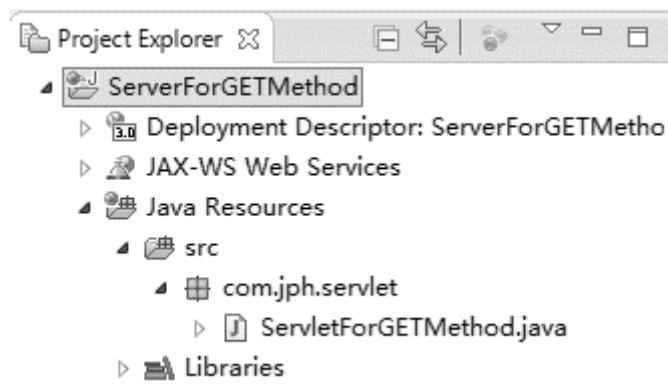


图 4.22 服务器端项目

(2) 创建控制器 Servlet。所有的 http 请求都会发给 Servlet 控制器，Servlet 控制器来决定谁来处理该请求的分派，创建控制器代码如下图 4.23 所示。

```
@WebServlet("/ServletForGETMethod")
public class ServletForGETMethod extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ServletForGETMethod() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        // TODO Auto-generated method stub
        // 获取请求的参数（使用utf-8进行解码，然后用进行ISO8859-1编码）
        String name=new String(request.getParameter("name").getBytes("ISO8859-1"),"utf-8")
        String name=request.getParameter("name");
        String pwd=request.getParameter("pwd");
        System.out.println("name:"+name+"    pwd:"+pwd);
    }
}
```

图 4.23 控制器的创建

(3) 测试 Servlet 控制器，发布项目并在浏览器中输入：
http://localhost:8080/ServerForGETMethod/ServletForGETMethod?name=aa&pwd=124 并观察控制台的输出。

4.5.2 创建 Android 端

第一步：创建 Android 端业务逻辑层，核心部分代码是 SendDateToServer.java，通过 GET 方式向服务器发送数据，代码如下图 4.24 所示。

```

public class SendDataToServer {
    private static String url="http://10.219.61.117:8080/ServerForGetMethod/ServletForGetMethod";
    public static final int SEND_SUCCESS=0x123;
    public static final int SEND_FAIL=0x124;
    private Handler handler;
    public SendDataToServer(Handler handler) {
        // TODO Auto-generated constructor stub
        this.handler=handler;
    }
    /**
     * 通过Get方式向服务器发送数据
     * @param name 用户名
     * @param pwd 密码
     */
    public void SendDataToServer(String name,String pwd) {
        // TODO Auto-generated method stub
        final Map<String, String> map=new HashMap<String, String>();
        map.put("name", name);
        map.put("pwd", pwd);
        new Thread(new Runnable() {
            @Override
            public void run() {
                // TODO Auto-generated method stub
                try {
                    if (sendGetRequest(map,url,"utf-8")) {
                        handler.sendMessage(SEND_SUCCESS);//通知主线程数据发送成功
                    }else {
                        //将数据发送给服务器失败
                    }
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        }).start();
    }
}

```

图 4.24 通过 GET 方式向服务器发送数据

4.6 用户管理

本设计不仅面向单一的用户，可以提供用户登录界面，对多用户进行管理。建立新类 login.java，在类中创建一个用于用户登录的 activity，用户可以设置自己的用户名（user）以及密码（password）。不同的用户登录到软件上会将用户的健康信息存储到不同的数据库文件里，以此实现对不同用户的数据管理，login.java 的源码如图 4.25 所示。

```

public void onClick(View v) {
    // TODO Auto-generated method stub

    String user = new String("lsl");
    String pass = new String("19921112");

    EditText username = (EditText)findViewById(R.id.editText1);
    EditText passwd = (EditText)findViewById(R.id.editText2);
    String Username = username.getText().toString();
    String Passwd = passwd.getText().toString();

    passwd.setText("OK");

    if ((Username.equals(user)) && (Passwd.equals(pass))) {
        Intent intent = new Intent();
        intent.setClass(login.this, BluetoothChat.class);

        startActivity(intent);

        login.this.finish();
    }
}

```

图 4.25 登陆界面源码

为登陆界面设计一个 layout，在 res 文件夹中新建一个 xml 文件，命名为 login.xml，添加控件并调整控件的布局，对应之前 login.java 中的定义。源码如下图 4.26 所示。

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#808080"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="User"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword" />

```

图 4.26 登陆界面的布局文件

4.7 界面显示



图 4.27 软件界面设计

本设计主要的交互界面有两个，分别是进行用户管理的登陆界面和数据采集界面，如图 2.27 所示。界面的布局主要由三种控件组成，Button 控件、Textview 控件和 Edittext 控件。空间定义在各个 Activity 对应的布局 xml 文件中，以采集界面的 xml 文件为例，如图 4.28 所示。

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#808080"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="User"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" />
```

图 4.28 登陆界面的布局文件

Textview 控件为静态变量，主要负责显示各部分的功能以及提示作用。Button 控件是各种命令在布局上的表现，通过设置控件监听 `setOnClickListener(new OnClickListener(),` 来对空间的状态进行监听，如果对 Button 控件进行了点击操作，监听就会返回控件的状态。例如本设计中如果点击了 Start 按钮，就会激活蓝牙设备，并向血压计发送采集命令，在监听对象中包含了点击动作之后需要完成的事件，源码如下图 4.29 所示。

```

start.setOnClickListener(new OnClickListener(){
    public void onClick(View v) {
        // TODO Auto-generated method stub
        start.setEnabled(false);
        stop.setEnabled(true);
        sendMessage(ProtocolConstants.wake);
        try{
            Thread.sleep(500);
        }catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        bufferIndex = 0;
        currentIndex = 0;
        sendMessage(ProtocolConstants.start);
        System.out.println("start");
    }
});

```

图 4.29 Start 事件的监听对象

Activity 之间的跳转需要 Intent 对象来实现, Intent 对象可以实现例如打开图片或者是发送消息之类的操作。例如登陆界面和数据采集界面的跳转如下图 4.25 所示。

```

passwd.setText("OK");

if ((Username.equals(user)) && (Passwd.equals(pass))) {
    Intent intent = new Intent();
    intent.setClass(login.this, BluetoothChat.class);

    startActivity(intent);

    login.this.finish();
}

```

图 4.30 Intent 实现界面跳转

EditText 控件可以实时显示数据, 包括测量的状态和采集的结果。通过 (控件 ID).setText 可以在 EditText 控件中显示要现实的数据或者文本。例如在控件中显示收缩压测量结果: systolicpressureET.setText(Integer.toString(SysPval)+ProtocolConstants.Unit)。

最后关闭软件的时候点击 Android 系统手机的退出按钮, 会跳出是否退出的选项, 代码以及界面如图 4.31、图 4.32 所示。



图 4.31 软件退出界面


```

public boolean onKeyDown(int keyCode, KeyEvent event)
{
    if (keyCode == KeyEvent.KEYCODE_BACK )
    {
        //
        AlertDialog isExit = new AlertDialog.Builder(this).create();
        //
        isExit.setTitle("退出");
        //
        isExit.setMessage("是否确认退出");
        //
        isExit.setButton("确定", listener);
        isExit.setButton2("取消", listener);
        //
        isExit.show();
    }
    return false;
}

```

图 4.32 软件退出界面代码

4.8 本章小结

本章首先对基于 Android 操作系统智能手机平台软件进行了需求分析，然后从两大块、六个方面介绍了本设计预期的功能如何实现。

（1）数据存储：使用 SQLite 数据库将采集到的健康数据保存在了手机本地的数据库文件中。介绍了 SQLite 的结构以及使用方法，以及如何在设计中实现。

（2）数据解析：实现了手机与血压计之间的命令与响应，将血压及传输回来的采集结果报文解析，并进行位制装换，并显示在控件上。

（3）蓝牙通信：利用 Android 自带的蓝牙操作类与对象，实现了手机和血压计之间的数据通信，实现了如何激活蓝牙，搜寻设备，连接设备，发送数据以及休眠设备。介绍了本设计中蓝牙通信的操作步骤。

（4）数据上传：用网络编程实现数据上传。本两部分实现数据上传，第一部分是服务器端的编程，第二部分是 Android 端的编程。使用 GET 方法实现了数据上传到服务器。

（5）用户管理：设计了用户登录界面进行用户管理，将不同用户的健康数据存储在不同的数据库文件中，实现多用户的使用。

（6）界面显示：使用 Android 提供的布局以及控件实现了本设计的界面显示，实现了登陆界面与采集界面，将采集到的结果显示在界面上，并在界面上实时反映采集的状态。

通过这六部分的功能的编程实现，基本达到了本设计的预期效果。

第五章 调试与实验结果

5.1 蓝牙模块模拟调试

使用蓝牙血压计调试需要样机测试，而且程序调试次数多，无法做到每次测试都进行样机测试，因此有必要先使用独立的蓝牙模块进行模拟调试，确定通讯协议的正确性以及数据解析的正确性。

蓝牙模块为 HC-06 系列，使用串口调试助手（sscom3.2）接收手机的命令，再通过串口向手机发出报文。

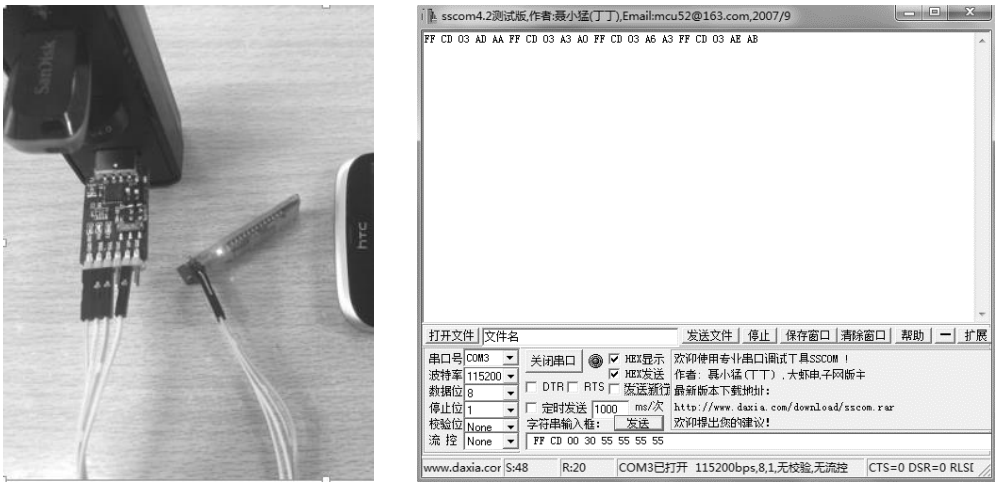


图 5.1 串口调试助手收到手机指令



图 5.2 将数据发送到手机上

通过使用单独的蓝牙模块模拟测试，通讯协议准确，手机设备可以搜索并连接到蓝牙模块，打开手机蓝牙权限后，连到蓝牙后可以接收和发送数据，并且显示数据为发送的十六进制数据对应的十进制数据，说明数据解析正确。

5.2 手机与血压计联调

在测试成功之后，将手机与领泰医疗的蓝牙血压计进行联调，并进行样机测试。苏州领泰医疗的蓝牙血压计如下图 5.3。



图 5.3 领泰医疗蓝牙血压计

手机界面上的测试软件 apk 图标以及样机测试现场如图 5.3 所示。测试步骤如下：

一、将臂带绑在受测人的左臂上，绑紧臂带，将血压计电源打开，红灯变绿并闪烁，表示等待蓝牙设备连接。

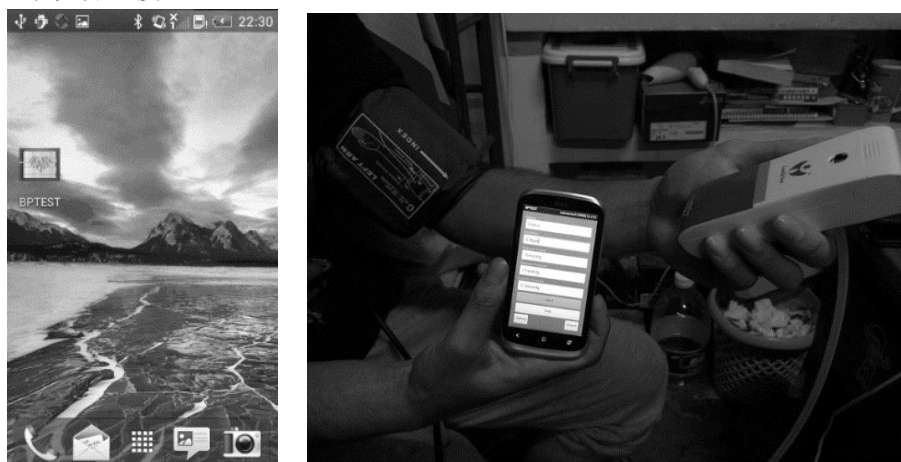


图 5.4 APK 图标（左）样机测试（右）

二、用户登录和蓝牙连接

(1) 打开手机应用，进入登录界面。输入用户名 (lsl)，密码 (921112)，进行登录。

(2) 进入之后，长按菜单键，弹出蓝牙菜单，从上至下依次是连接到可靠的设备 (connect a device- secure)，连接至不可靠设备 (connect a device- insecure)。最下面是使自身的设备可以被周围的蓝牙设备检测到 (make discoverable)。

(3) 检测到设备 LT BPG_007 就是本设计的领泰医疗血压计，连接成功后应用程序右上角会显示 connected LT BPG_007。

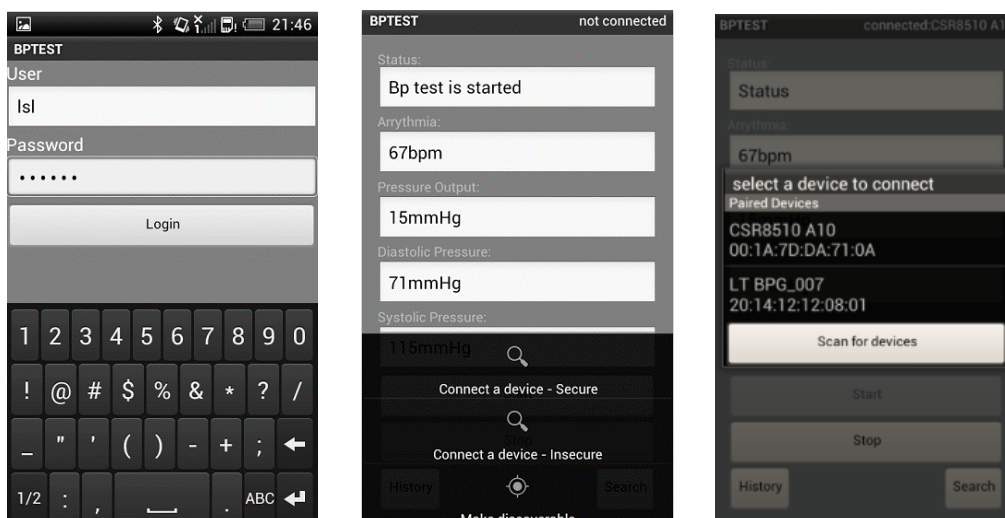


图 5.5 测试步骤 登录（左）、打开蓝牙权限（中）、选择设备（右）

三、开始采集

(1) 手机发送命令激活蓝牙设备，创建数据库、表。开始采集数据后，气囊开始充气，在 Pressure Output 一栏中，实时显示气囊里的气压。在状态栏 Status 里，显示当前的状态，如果手臂气囊正常绑紧的话，显示"BP test is started!", "Pumping Air Bag!"。

(2) 采集结束后，控件 Arrhythmia 心率、Diastolic pressure 收缩压、Systolic pressure 舒张压中，会显示测量结果，如图心率 67bpm，收缩压 71mmHg，舒张压 115mmHg。

(3) 采集结束后，点 stop 按钮，清空所有控件的显示，并准备下一次测量。

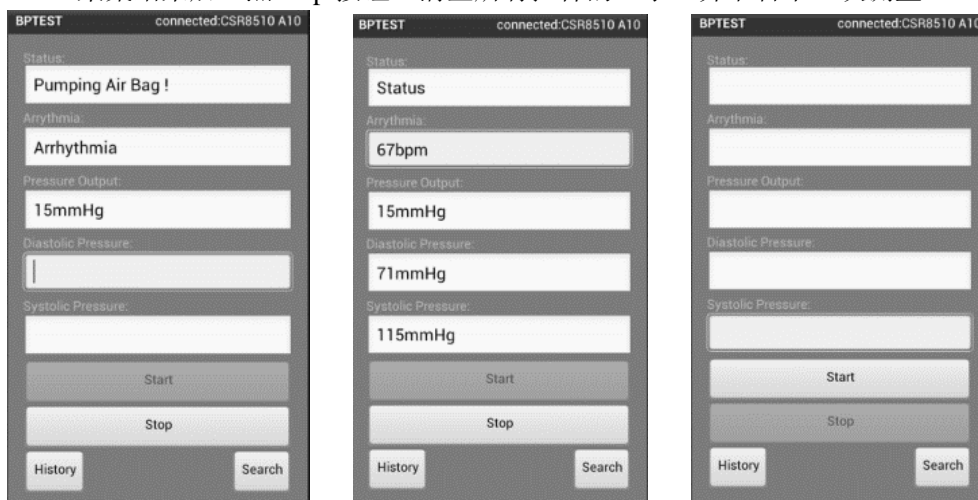


图 5.6 测试步骤 开始采集（左）、显示结果（中）、控件清除（右）

三、如果采集中出现异常情况，例如走动，说话，气囊未绑紧等情况，传感器会发出错误报文，手机应用将会在状态栏里显示这些状况。状况的常量定义在 ProtocolConstants.java 中，如下图 5.7 所示。

```

public static final String NotStart = "BP test is not started!";
public static final String Stop = "BP test is stopped!";
public static final String NotStop = "BP test is not stopped!";
public static final String CommFailed = "Communication is failed!";
public static final String ING = "Testing !";
public static final String Results = "Results are as follows:";
public static final String StatusError1 = "Cannot detect accurate pulse!";
public static final String StatusError2 = "Air bag did not tight correctly!";
public static final String StatusError3 = "Results contains wrong info!";
public static final String StatusError4 = "Air bag over-pressured!";
public static final String StatusError5 = "Interference !";
public static final String StatusError6 = "Unknown error !";

```

图 5.7 异常情况报告

例如臂带没有绑紧导致气囊内气压不足，或者由于在测试过程中受到干扰（例如说话、走动），软件会在状态栏里显示出错误报告。臂带未绑紧的错误报告以及测试受到干扰的报告如图 5.8 所示。

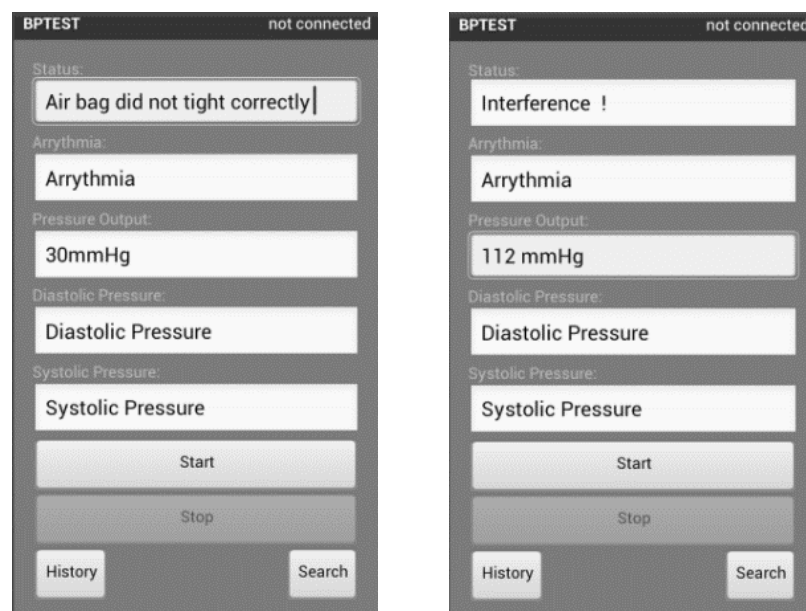


图 5.8 臂带未绑紧（左）、测试干扰（右）

5.3 本章小结

本章介绍了手机软件和硬件调试的过程以及结果。在测试血压计之前，先使用了单独的蓝牙模块，利用电脑的串口，与手机进行模拟调试，测试通信协议是否正确，是否能接收到手机的指令，是否可以收到蓝牙传输的数据，以及最后数据的转换以及存储是否正确，能否在控件中显示出来。

第二部分将血压计与手机进行联调，并进行样机测试。验证了本设计的可行性以及方案的正确性、数据的准确性。实现了用户管理、数据存储、界面显示等功能。

结束语

基于 Android 智能手机的个人健康信息终端软件这个设计的核心部分是基于 Android 平台的 BPTEST，核心部件是智能手机，在整个设计中都是围绕着怎样能使智能手机发挥出穿戴式医疗设备的优点进行的。穿戴式医疗设备最重要的特点就是便捷，贴近生活。智能手机恰恰有相同的特点。因此，二者的结合将是非常有实用意义的。

本文从穿戴式医疗设备的应用和已有的技术积累出发，设计了基于 Android 操作系统智能手机平台的人体健康采集软件设计，介绍了在移动终端上健康应用的发展，阐述了人体健康数据采集理论和技术基础以及本课题研究的意义。该采集软件能够以良好的性能运行于 Android 系统，并且实现了课题的要求。

在本课题的研究过程中，总共实现了以下功能：

一、编写了数据传输和转换的程序，实现了采集数据传输模块数据的功能。

二、使用 Java 语言，基于手机平台，实现了数据的实时采集、输出、并显示在手机屏幕上的功能。

三、实现了使用 Android 自带的 SQLite 轻型数据库，将数据保存在了数据库中。

四、蓝牙通讯协议实现了和上位机的命令交互，包括激活命令、休眠命令、开始采集命令和停止采集命令。对命令进行解析并通过蓝牙进行数据的实时传输。

致谢

本文是在林国余老师的悉心指导下完成的。“悉心”二字，绝非可有可无。因为，从选题到今天成文，林老师给予了我很大帮助，付出了大量的心血。可以说，没有林老师的指导，就没有今天的顺利成文。在这段时间里，我感到学到很多东西。林老师治学严谨、做事认真，大到结构，小至标点，一一从严要求，决不允许敷衍；林老师为人真诚、待人热情。很多次，都是林老师给不厌其烦的给我改正错误，及时传道授业、答疑解惑。在此，向林老师致以最诚挚的谢意。

感谢在血压测试中愿意帮助我测试的同学，以为一个人反复测量血压会有不适，因此很多同学愿意帮助我完成测试任务，对我测试工作提供了很大的帮助，他们是刘经纬、孙嘉辉、孙业飞、俱子研、张劭、李奕其、骆凡平。感谢他们对我工作的支持。

我还要感谢在大学四年来所有曾经教过及帮助过我的老师，是他们的谆谆教诲、无私奉献，使我增加了知识、提高了能力！没有他们，也就没有我四年学业的顺利完成！

另外，我还要感谢四年来朝夕相处的同学，和他们切磋知识、交流思想，也使我感到受益匪浅！最后，向所有关心和帮助过我的老师和同学们致以最诚挚的谢意！

参考文献

- [1]滕晓菲,张元亨. 移动医疗:穿戴式医疗仪器的发展趋势[J]. 中国医疗器械杂志, 2006, 05:330-340.
- [2]曹沁颖. 面向病患的穿戴式健康医疗产品设计研究[D]. 沈阳航空航天大学, 2015.
- [3]明轩,徐玉炎,张瑞,李奇越,孙秀丽. 基于 ZigBee 的穿戴式医疗监护系统节点的设计与实现[J]. 电子设计工程, 2014, 21:81-83+87.
- [4]王景灿,郭兴明,李立策,谭新. 基于射频技术的穿戴式医疗仪器的设计[J]. 现代科学仪器, 2008, 01:90-92.
- [5]黎阳子. 基于 Android 平台的移动医疗系统的研究与设计[D]. 武汉科技大学, 2014.
- [6]王景灿. 可穿戴远程医疗系统用户端的硬件研究[D]. 重庆大学, 2008.
- [7]李黎国. Android 健康服务终端无线技术的研究与设计[D]. 西安电子科技大学, 2012.
- [8]陈建刚. 基于 Android 客户端和 Web 服务器的个人健康档案系统[D]. 北京邮电大学, 2012.
- [9]张秀香. 基于 Android 的健康管理系统客户端的设计与实现[D]. 大连理工大学, 2012.
- [10]李高尚,沈巍. 一种应用于移动健康医疗的 Android 客户端的研究[J]. 电子测试, 2012, 02:64-67.
- [11]李黎国,张辉,程号. 基于 Android 健康服务终端蓝牙传输软件的设计[J]. 电子科技, 2012, 05:115-118.
- [12]熊狮. 基于 Android 系统健康信息移动监测技术的研究[D]. 华南理工大学, 2013.
- [13]花正东. 基于 Android 手机健康管理平台的研究[D]. 华南理工大学, 2013.
- [14]黄灿添. 基于 Android 移动互联的健康监护技术研究[D]. 华南理工大学, 2014.
- [15]冯艳红,何加铭,杨任尔,冯波. 基于 Android 蓝牙技术的健康服务系统设计[J]. 无线电通信技术, 2014, 01:61-64.
- [16]杨泽军. 基于 Android 平台的健康感知信息采传系统研究与实现[D]. 山东师范大学, 2013.

[17] Rajani S. Sadasivam, Vaibhav Gathibandhe, Murat M. Tanik, James H. Willig. Development of a Point-of-Care HIV/Aids Medication Dosing Support System Using the Android Mobile Platform[J]. Journal of Medical Systems, 2012

[18] Anonymous. Research and Markets: The World of Health and Medical Apps: 5,820 Medical, Health and Fitness Apps Available for Smartphones Today by iPhone, BlackBerry, Android, Nokia and Palm[J]. M2 Presswire, 2010