



# 广东工业大学

## 本科毕业设计（论文）

### 户外智能运动服装功能设计

学    院     自动化学院  
专    业     物联网工程  
年级班别     2012 级（1）班  
学    号     3112001409  
学生姓名     黄庆麟  
指导教师     王春茹

2016 年 6 月

户外智能运动服装功能设计

黄庆麟

自动化学院

## 摘 要

户外智能运动服装功能采用非入侵人体的智能传感器来实时采集生命特征现象并进行数据存储来监测影响我们健康的关键信息。户外智能运动服装功能系统是一个集成系统，这系统包括智能传感器、GPS 模块地理信息定位功能、4G 与蓝牙 4.0 无线通信模块、云计算平台、手机端及 Web 管理端多个功能模块。本文描述了智能运动服装的系统框架的体系结构设计、通信网络及通信协议设计、云计算平台通信接口设计、众多智能传感器布局设计，构建了整个智能服装系统，最后以树莓派主控板为处理器，完成了脉搏等信号的采集及与蓝牙 4.0 通信等功能；最后开发了客户端 Android 应用软件，实现了完整的户外智能运动服装功能。

**关键词：** 智能传感器，户外智能运动服装，集成系统

## **Abstract**

Outdoor smart sportswear with noninvasive smart sensors handling vital signs are expected to be key elements for our health management in the future. A outdoor smart sportswear system is an integrated system including such smart clothes with sensors acquiring bio-signals, GPS modules , 4G and Bluetooth 4.0 wireless communication module, cloud-computing, mobile phone side and Web management and multiple function modules. Firstly, the paper introduces the architecture of the system framework of intelligent sportswear design. The system consists of many intelligent sensor, communication network and cloud-computing platform. Then the paper presents the design of whole system. Finally, the paper introduces the implementation of the pulse sensor, Raspberry Pi, Bluetooth 4.0 and Android APP for this project design.

**Key words:** Smart Sensors, Outdoor Smart Sportswear, Integrated System

# 目 录

1	绪论 .....	1
1.1	论文背景及研究意义 .....	1
1.2	国内外研究状况 .....	2
1.3	论文研究内容及章节安排 .....	3
2	智能服装关键技术研究 .....	5
2.1	体系结构 .....	5
2.1.1	信息检测层 .....	5
2.1.2	信息处理层 .....	6
2.1.3	信息反馈层 .....	7
2.2	通信网络 .....	7
2.2.1	硬件模块间的数据通信接口 .....	9
2.2.2	终端之间的通信方式 .....	9
2.3	云计算平台 .....	10
2.4	Android 系统简介 .....	10
2.4.1	Android 手机操作系统 .....	10
2.4.2	Android 平台优势 .....	12
3	智能运动服装功能设计 .....	13
3.1	户外智能运动服装功能设计需求分析 .....	13
3.1.1	感知环境 .....	13
3.1.2	感知人体生理指标 .....	13
3.1.3	地理位置信息 .....	13
3.1.4	通信网络稳定 .....	13
3.1.5	合身舒适安全 .....	13
3.1.6	方便移动能源 .....	14
3.1.7	双向防丢功能 .....	14
3.2	户外智能运动服装功能总体设计 .....	14
3.2.1	智能运动功能方案研究和设计 .....	14

3.2.2 智能运动服装功能设计系统框架 .....	15
3.3 户外智能运动服装功能各模块设计 .....	18
3.3.1 智能传感器 .....	18
3.3.2 4G 模块及 GPS 模块 .....	18
3.3.3 树莓派（包括移动电源） .....	19
3.3.4 蓝牙 4.0 模块 .....	19
3.3.5 Android APP .....	20
3.3.6 云计算平台通信接口设计 .....	20
3.3.7 系统通讯协议设计 .....	20
4 智能运动服装功能实现 .....	22
4.1 智能运动服装功能各模块实现 .....	22
4.1.1 血压传感器 .....	22
4.1.2 体温传感器 .....	23
4.1.3 脉搏传感器 .....	23
4.1.4 蓝牙 4.0 模块 .....	24
4.1.5 树莓派 .....	24
4.1.6 Android APP .....	27
4.2 智能运动服装功能实现展示 .....	28
结 论 .....	32
展 望 .....	34
参 考 文 献 .....	35
致 谢 .....	36
附录 A 树莓派程序代码 .....	37
附录 B Android 应用主要程序 .....	40

# 1 绪论

近年来，家庭医疗保健和远程医疗快速增长成为医疗实践的新趋势。在这趋势下，数字服装和智能衣服最为合适和突出。因为它们作为非入侵人体传感器结合的服装成为我们生活中的一部分，所以有助于无时无地采集各种生命特征信息，比如体温、心电图（Electrocardiograph）、呼吸、血压等等。在这些方面，这类衣服可以通过潜在的可穿戴设备来提供家庭医疗保健管理和预防疾病<sup>[1]</sup>。

## 1.1 论文背景及研究意义

智能服装原属尖端领域，随着服装行业日益红火，嗅觉灵敏的商家们拓宽了其原先狭小的市场空间。在未来几年里，我们的衣柜里将会塞满这样的智能衣物——能够读出人体心跳和呼吸频率的“聪明衬衫”；能够自动播放音乐的外套；能够在胸前显示文字与图像的 T 恤衫……美国科技媒体预测，未来的服装将成为真正的“多功能便携式高科技产品”，一件衣服能同时播放音乐、视频、调节温度，甚至上网冲浪。

就以目前的研究方向分析，智能服装大体上分为两个范畴。一是对服装材料进行改良，通过化学、物理手段改变纺织材料的结构，使之具备普通材料所不具有的更适用于我们需要的功能。例如：以形状记忆材料织成的服装，可以根据外界温湿度的变化来自自动调整结构，以便穿着者能够抵御冷或热气候的侵袭；以储能纤维织成的服装能够辐射红外线，从而达到保暖的目的。智能服装的另一个范畴是与信息技术紧密交织在一起，将信息技术以不影响穿着舒适性和不被察觉的方式嵌入到服装中，使之具有信息感知、计算和通信功能，满足各种不同情况的应用需求。这就是所谓的高级智能服装<sup>[2]</sup>。

本论文着重研究智能服装的第二范畴。其实，真正的智能服装应该是一个复杂的信息载体。纺织、材料、服装与信息技术的成功结合使得智能服装的开发成为可能。将电子信息技术嵌入人体日常穿着的服装中，一方面将赋予服装除了保暖、美观外的更多的功能，另一方面也将信息系统紧密地、方便地服务于我们。本论文基于智能服装领域的第二应用范畴，自拟构建户外智能运动服装功能系统，该系统主要采用材料、纺织、服装等领域先进技术，将检测、处理、存储、通信和输入输出等电子元器件进行微型化、柔性化以非入侵人体的形式植入服装内，在不影响服装穿着舒适性的条件下，实现用户在户外运动时的实时身体状况监测，向用户提供智能分析决策支持和反馈控制功能，以

保障日常健康生活质量。使得用户在室外可以达到在医院监测的效果。通过自拟构建户外智能运动服装功能系统,为进一步开展智能服装结合生物信息、微机电、可再生能源等多学科领域,面向军事、航天、救灾等多应用领域的研究打下基础。随着智能服装的发展,智能服装逐渐走进我们日常生活。尤其是在我们进行一些户外活动时,我们常常会不注意自身的状态而进行不合理的运动,进而在我们无意识下导致我们损害自己的身体健康,这不仅得不到锻炼身体的好处,反而成了恶性运动对自身不健康的隐患。针对此现象,当代人越来越关注自己的身体健康,而且也很愿意花费时间去户外运动,但往往我们并不知道自己的身体状态是否适于运动,所以本论文研究户外智能运动服装设计将变得更有意义。

## 1.2 国内外研究状况

人们对智能服装的开发研究最早源于 70 年代末,当时电子技术还比较落后,所以智能服装的发展也十分缓慢,其体积庞大而累赘,且功能单一。因此智能服装的应用领域也受到很大局限,主要应用在航空航天、军事军工等特殊行业领域。而今天,智能纺织品成为从纺织纤维和消费品中派生出来的新一代纺织新品种,是一种具有形态记忆能力的纺织品,它具有对外界环境感知和反应的功能,并且还保留有美学和技术特性,成为 21 世纪新服装时尚的基础。用于工业、技术、医药或军用纺织品的发展,成为开启未来时尚之门的一把钥匙。作为“第二皮肤”的生物纺织品或 e 纺织品将会有非常好的市场前景,它们被看作是纺织工业的未来。随着国际纺织品贸易自由化体制的重新建立,各国纺织业将有更多机会利用全球资源从事纺织面料和服装设计更新换代,把梭织、针织和印花衣料与全新的传导材料混纺,制造出一系列崭新的“智能服装”<sup>[2]</sup>。

到目前为止,在市面上已经出现了不少比较实用的可穿戴智能服装,包括内置生物识别技术的 T 恤、胸罩和专业运动服装。而在 2015 年里,智能服装领域将进入到完全以生物识别技术为主的阶段。下面本文将对国外在智能服装领域的相关研究情况进行简单的介绍。

Hexoskin 是智能运动服装里最著名的产品之一。这款 Hexoskin 运动背心集成了生物传感器,可以让经常参加体育锻炼的用户能监测自己的锻炼进度。这款 T 恤集成有多种传感器,每分钟能收集 42000 个数据,因此可以负责任地说,Hexoskin 将能够收集与用户和用户锻炼进度有关的详细数据。在白天,Hexoskin 可以测量心率、心率变化/恢复、



步数、卡路里消耗和呼吸等数据;而到了晚上,它还能追踪睡眠和环境,包括睡觉的姿势,以及心跳和呼吸活动。所有这些数据会通过蓝牙同步到配套的应用程序当中,或者是在线上传,以供远程教练实时查看。

而当前国内也有不少的企业朝向智能服装领域方向研发。其中, **Leist smart** 智能皮衣是国内市场上领先的智能服装领域产品。**Leist smart** 将智能模块与服装结合,在不影响外观和舒适度的情况下可实现一键控制、蓝牙通信、音乐播放、防丢失、无线拍摄、智能恒温功能。**Leist Smart** 智能皮衣的关键点除了能够塑形之外,“智能”两字才是最重中之重,虽然其功能不算顶尖科技,但却很实用,**Leist Smart** 智能皮衣整合了众多科技元素,让我们在繁忙中解放了双手,娱乐中更具吸引力、更加便捷,同时,滴滴的提示,帮助我们别忘记携带手机。而智能保温功能的应用,即使身处凛冽的冬天,也不惧怕寒风吹袭。**Leist smart** 重心在于用户智能服装的功能体验,而忽视了用户更为关注的一些重要的生理大数据分析来关注自身健康状况,这对用户来说更具有意义。相信在不久的将来,智能服装更多的重视用户健康方面,并且在该方向不断进步。

综上所述,户外智能运动服装在国内外市场上已存在不少。户外智能运动服装主要功能包括血压测量、心率测量、体温测量、呼吸频率、**EMG** 传感器肌肉测量、步数、卡路里消耗。市场上的大部分产品功能都显单一化,集成度不高。造成其现象的重要原因是集成化功能逻辑复杂影响性能稳定、功耗大、低功耗微处理器能力有限。

### 1.3 论文研究内容及章节安排

将发展迅速的物联网技术、智能传感技术运用到运动服装的设计中,能够实时了解到运动者生理指标情况、周围环境及运动者位置等信息。本论文在分析需求的基础上,确定户外智能运动服装的功能(例如生理指标监测、定位、周围环境监测等),并提出硬件方面和软件方面的需求,给出智能运动服装设计方,并在技术上实现方案中的各个功能模块的功能,并对智能运动服装信息获取与处理的关键技术进行研究与探讨,为运动服装的功能可扩展性及相关行业的发展提供有益的经验。

本论文章节安排如下:

第 1 章阐述了论文研究背景及研究意义、国内外研究情况,并介绍本论文的研究内容以及论文章节安排。

第 2 章围绕论文智能服装系统的关键技术进行了研究,内容包括总体框架、通信网

络、云计算平台及 Android 系统的详细介绍。

第 3 章针对论文户外智能运动服功能，提出系统性设计方案，分别对系统需求分析、总体设计进行了详细阐述，展现设计过程。

第 4 章针对论文户外智能运动服功能，陈述方案关键技术实现过程。

## 2 智能服装关键技术研究

### 2.1 体系结构

针对户外智能运动服装功能设计的需求，逻辑上可将其功能划分为信息检测层、信息处理层、和信息反馈层三层体系结构。如图 2.1 所示。

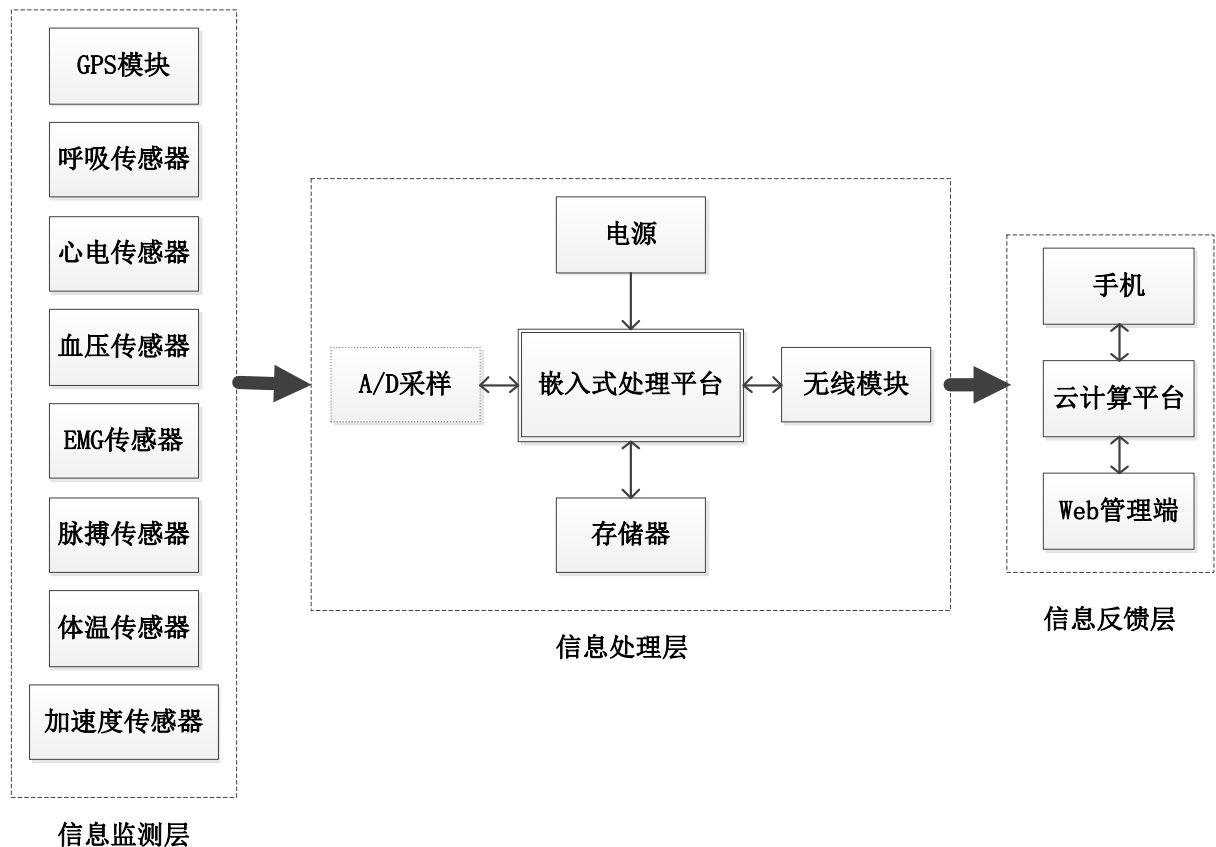


图 2.1 户外智能运动服装功能设计的总体框架

#### 2.1.1 信息检测层

信息检测层包含智能传感器、放大器、滤波器等硬件设备，主要实现对人体生理信号及其它外部信息的检测，是系统与外界环境信息交互的接口。传感器是服装与人体进行交互的第一环节。近十几年来，随着科技的发展，生物医学用传感器得到快速发展。传统的电阻式、电容式、电感式、压电式、光电式传感器与先进的光纤传感器、生物传感器等为生物医学信号的准确测量提供了方便<sup>[3]</sup>，各种各样的生物传感器已经在各种生物医学检测领域中得到了广泛的应用。涉及到的关键技术在于智能传感器的通信协议，

智能传感器通信协议主要是指智能传感器与主控板的通信协议，其主要通信方式有 I2C 通信、USB 串口通信、模拟信号通信等等，这部分主要解决智能传感器实时采集的生命特征现象以数字信号或者模拟信号的形式告知主控板，该部分数据为原始数据。当同时使用多种智能传感器，需要处理好数据同步与程序异步所引起的逻辑问题。目前，嵌入在服装中的传感器一般可分为三类：生理传感器、运动传感器和环境传感器。

生理传感器利用先进的信息和仪器设备制造技术，将原本大型的医疗检测设备如心电图 ECG、肌电图 EMG 等小型微型化，便于携带和嵌入到服装中，采集人体各种生理信息。其他常用的生理传感器如血氧、呼吸、脉搏、皮肤温度、血压等，还有声音传感器识别心音和肺音；汗腺排泄传感器、皮肤脱屑物传感器等。生理传感器的研究和开发应考虑能耗和信息分辨率的权衡<sup>[4]</sup>。

运动传感器有加速计、陀螺、倾角传感器等，如感知人的动作的可穿戴加速计（accelerometer），两种应于感知人体体态、行为的传感器：脊椎型的加速度计（spin, accelerate meter）和豪猪型的球形开关，各有其使用场合。

感知环境的传感器能感知光线、温度、压力、声音、图像、视频等，如何将其微型化以便嵌入服装，不影响穿着，且具有耐久性、耐洗涤、耐拉伸是可研究的课题。可穿戴的传感器还可以和材料和生物技术结合。如用新型压电材料研制的传感器用以感知变形、声音，并开发了手套原型感知手的运动<sup>[5]</sup>。基于泡沫塑料的压力传感器用于测试呼吸频率，易于嵌入服装。目前还没有开展的可穿戴生物传感器，如 DNA、血糖、血氧传感器可以帮助诊断治疗糖尿病、心脏病等慢性疾病，也将在可穿戴计算领域大有前途，其它还有化学传感器嵌入在救火服、井下作业服、战斗服中能检测各种化学物质，保护人体。

另外，户外智能服装最重要的一个功能是其全球定位功能。GPS 模块能获取全球经纬度，从而来确定当然该户外智能服装所在位置，因此其在此系统中发挥着重要作用。

### 2.1.2 信息处理层

信息处理层是整个系统的中枢，通过内置的微处理器对信息检测层所得到的生理信号进行特征提取、模式识别、分析和处理。这一层集成有智能决策系统，可以基于预先“学习”的病理知识，对获取的信息进行诊断，将抽象的外界信号翻译成为易于理解的语义，是整个系统的核心部分。涉及到的关键技术在于对主控板优劣势以及对其开发环境

的了解。主控板主要职责是数据处理、数据存储、数据传递，该部分主要充当系统的中间件的作用。这部分下与智能传感传感器进行原始数据通信，上与客户端应用层软件进行经处理的数据通信。因此，选择合适的微处理器和熟悉主控板编程环境是非常重要的。

针对服装可穿戴系统传感点所需的电源系统近年逐渐成为了热点。除了传统的高效能电池，还有研究如何采集人体能量，太阳能为系统供电。可穿戴系统供电的方式可以分为分布式电池，集中式电池供电两种。可穿戴系统通常包括多电源系统，尚缺乏模型和模拟系统。

### 2.1.3 信息反馈层

信息反馈层是系统与用户交互的接口，这一层可以将微处理器分析所得到的结果通过有线或无线的方式传递给服装上的终端设备，并及时反馈给用户，以使服装的穿着者根据这些结果适时地做出响应。涉及到的关键技术在于优秀的交互方式以及客户端软件的合理设计，当前客户端应用软件包括 PC 桌面软件、智能手机应用软件、智能手表软件、平板软件、指定遥控设备软件等等。其中，最受欢迎的是当前发展前沿的智能手机应用软件，并且在当前市场已存在大量类似的客户端应用软件。

人机交互设备 (Human Computer Interface, HCI) 是目前可穿戴计算领域热点和难点。智能服装需要新颖的交互模式，以满足如下性质：穿着舒适性、移动性、隐藏、不干扰人的其他活动、多媒体等。Infineon Tech 公司的 MP3 服装的输入设备采用缝制在衣袖中的键盘衬垫 (keypad)，点击输入设备新型鼠标满足可穿戴环境下的使用<sup>[7]</sup>。另外，输入设备还可以利用声音、手势信号等特殊的输入方式。例如，FreeDigiter 可以快速通过手势输入数字。输入设备主要有头戴式，手表式和眼镜式显示屏。为了不妨碍用户工作，以特殊的方式提醒用户，进行事件或信息告知，也是研究得热点。

手机通过无线通信方式与智能服装进行连接通信，并以人性理念将数据可视化，使得用户可以直观了解自身的身体状态。云计算平台通过无线模块与信息处理层进行连接通信，来实现嵌入式处理平台数据上传、存储及分析。并且云计算平台作为手机、Web 管理端的数据处理后台，为后期实现大数据平台搭建奠定基础。

## 2.2 通信网络

对于户外智能运动服装功能设计来说，通信网络是非常重要的。其中包括以下几个

方面通信：服装内部各组件之间的通信；服装与穿着者之间的通信；服装与外界环境之间通信。

服装内部各个组件之间的通信，目前是采用光纤或者可导纤维，它们都具备服装纺织的一些特性，能够很好地与服装无缝结合。它们有各自的优缺点：光纤质量轻，对电磁辐射不敏感，传导过程中不会产生热量，但是信号必须在至少有一端需要转换成电信号，增加了系统的复杂度。

服装与穿着者之间的通信是非常重要的部分，即所谓的人机交互接口。它常常采用压敏纺织材料，譬如软开关，能够接受用户的输入信息，服装内的微处理器翻译该信息并进行相应处理。

服装与外界环境之间通信一般采用无线通信来实现。通过在服装中内置无线模块和天线，完成和外界的交流。目前，国外有些研究机构正在研究如何利用服装足够的表面织成天线<sup>[6]</sup>，这样穿着者根本觉察不到它的存在。为了实现该通信技术，可利用现有的 4G、GPRS 移动通信网络和 Internet 等公共通信网络，实现相关数据的无线传输。随着无线通信技术、网络技术、嵌入式计算机技术以及智能传感器技术的发展，无线传感器网络（Wireless Sensor Network, WSN）已成为近年来国内外热门研究领域<sup>[7]</sup>。无线传感器网络由部署在监测区域内大量的微型传感器节点组成，通过无线通信方式形成的一个多跳的自组织的网络系统，目的是协作地感知、采集和处理网络覆盖区域中感知对象的信息，并发送给观察者。无线传感器网络最早应用于冷战时期的军事领域。2003 年，美国将其列为未来新兴十大技术的第一位。同时，由 2003 年底标准确立的低速低功耗无线个域网协议 IEEE 802.15.4 和 ZigBee 联盟共同制订完成的新兴的无线传感器网络技术标准 ZigBee，具有通用性强、低功耗、低成本和低复杂性度等特点。在国外，除军事领域外，政府已经开始着手将无线传感器网络应用于医疗保健、防灾救灾、智能家居等领域。为实现智能服装在实际中心中的轻捷、简便，无线传感器网络通信技术将成为未来理想的服装监控设备的关键技术。涉及到的关键技术在于通信网络的流畅度最大化、异常处理方式，当前无线通信随着科技不断进步逐渐取代有线通信，其中无线通信方式包括 WIFI、ZigBee、蓝牙、3G/4G 等等。该部分主要职责是解决主控板与客户端应用软件的数据通信，使用无线通信方式的原因是方便用户对客户端用户软件的使用，而不造成有线带来的使用不便<sup>[8]</sup>。

### 2.2.1 硬件模块间的数据通信接口

硬件模块间的数据通信接口需要确定通信接口协议和连接拓扑结构。为了抗干扰和高带宽性能，需采用数字传输<sup>[9]</sup>。现有的可选择通用接口技术被归纳如表 2.1 所示。

表 2.1 主要接口标准在智能服装中适用性分析

类型	优点	缺点
1-wire	简单	数据速率低
I <sup>2</sup> C	微控制器支持，4-线接口（+，-，时钟，数据）	面向板内，距离有限，电磁干扰
SPI	微处理器支持，数据时钟线分离	干扰
异步总线 (RS-232/CAN-bus)	面向板间的连接，在噪声环境中较可靠	需收发器
高速总线 (USB/Firewire)	高数据传输率，支持高层协议	硬件复杂，需要电缆
并行	高带宽	专用连接，线路数量多、复杂，电磁干扰

### 2.2.2 终端之间的通信方式

主控板与客户端软件的通信方式可分为无线通信和有线通信。针对智能服装功能设计需求，无线通信优势远大于有线通信。目前无线通信网络相关技术有：PAN(Personal area network 个域网)，UWB(超带宽)，Bluetooth，IEEE802.15.4。无线微传感器网络的应用领域十分广泛，包括超短距离的人体健康监护<sup>[10]</sup>、近距离的智能家居和长距离的环境监控，近年来受到极大地关注。设计无线传感器网络和制定其网络协议是一个极具挑战性的工作。尽管面向不同的应用，无线传感器网络有其共同制约条件，如密度大、分布式的流量和能源受限。微小型的节点无法携带大型的电池，数量多带来更换电池的不便。典型的能耗控制在 100uw，使其能够使用能量收集（energy scavenging）的方法从环境中获得能源。近年来，此研究领域致力于设计射频电路和传输协议来满足上述要求。IEEE 802.15.4 标准的颁布是一个发展的里程碑，规定了物理和媒体访问层的传输标准，主要针对小型、低速率、依靠电池供电的低能耗传感器<sup>[11]</sup>。较多的研究对其吞吐量和能耗效率通过仿真的方法进行了评估，并进一步进行优化，如针对高密度传感器网络，利用射频使能策略和链路适应方法，降低能耗。其他可穿戴网络设备，如个人移动集线器可作为可穿戴设备的通信代理，支持多种无线协议，使身上的传感器也能上 Internet。

## 2.3 云计算平台

云计算（cloud-computing）是基于互联网的相关服务的增加、使用和交付模式，通常涉及通过互联网来提供动态易扩展且经常是虚拟化的资源。本论文将采用云计算采集、存储、分析数据，为后期作大数据平台分析做准备。图 2.2 所示为云计算在户外智能服装功能设计的角色地位。

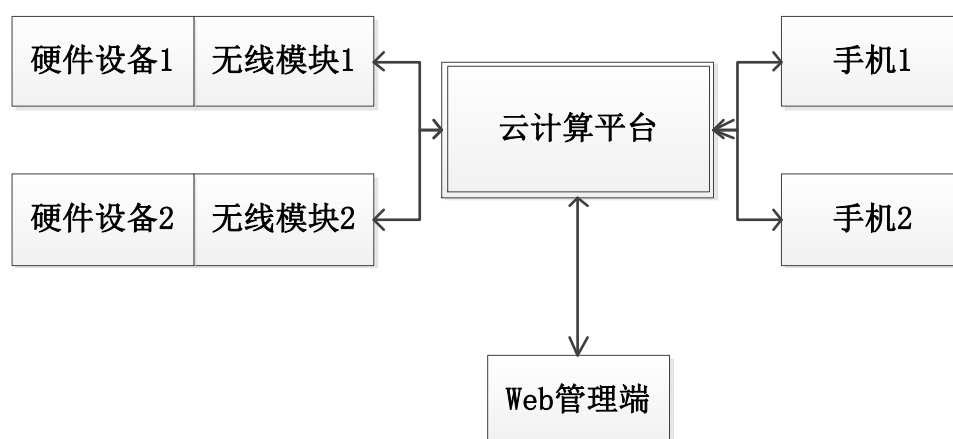


图 2.2 云计算在户外智能服装功能设计的角色地位

本论文云计算平台主要职能是对大量户外智能服装进行数据系统性管理，此平台主要是运用于群组形式，比如以家庭、班级、公司、旅游团等大小不一的群组分级管理。因此 Web 管理端可根据管理者权限限制来确定其组织管理权力，从而方便简单实现自由群组管理方式。云计算平台设计主要有三个数据通信通道，分别包括一个 Web 管理端、多个手机客户端、多个硬件设备。硬件设备需要具备接入互联网的能力，因此要添加 GPRS 模块或 WIFI 模块。这样不仅可以实现三端数据通信，而且能方便保持数据唯一性、数据管理便捷性。

## 2.4 Android 系统简介

### 2.4.1 Android 手机操作系统

Android 是 Google 于 2007 年 11 月 05 日宣布的基于 Linux 平台的开源手机操作系统的名称，该平台由操作系统、中间件、用户界面和应用软件组成。框架如图 2.3 所示，简单地说，就是将一个内核和一个库集合提供给整个应用程序框架，而该应用程序框架为运行时和应用程序提供服务，并对它们进行管理。

1) Linux 内核：核心服务（包括硬件驱动程序、进程和内存管理、安全、网络 and 电



影管理）都由一个内核处理。内核还在硬件和软件的其他部分之间提供了一层抽象。

2) 库：运行在内核之上，包含了各种核心库，例如，**libc** 和 **SSL**，以及用来回放音频和视频媒体的媒体库、提供管理显示的外观管理器、包含了和的的图形库、支持本地数据库的、集成了 **Web** 浏览器和 **Internet** 安全的 **SSL** 和 **WebKit**。

3) **Android** 运行时：**Android** 运行时可以让一个手机从本质上与一个移动 **Linux** 实现的手机区分开来。由于 **Android** 运行时包含了核心库和 **Dalvik** 虚拟机，因此，它是向应用程序提供动力的引擎，它和库一起形成了应用程序框架的基础。

4) 应用程序框架：应用程序框架提供了用来创建 **Android** 应用程序的类。它还对硬件访问和对用户界面及应用程序资源的管理提供了一般抽象。

5) 应用层：所有的应用程序，包括本地的和第三方的，都使用相同的库来构建与应用层之上。应用层运行在 **Android** 运行时之内，并且使用了其中可用的类和服务。

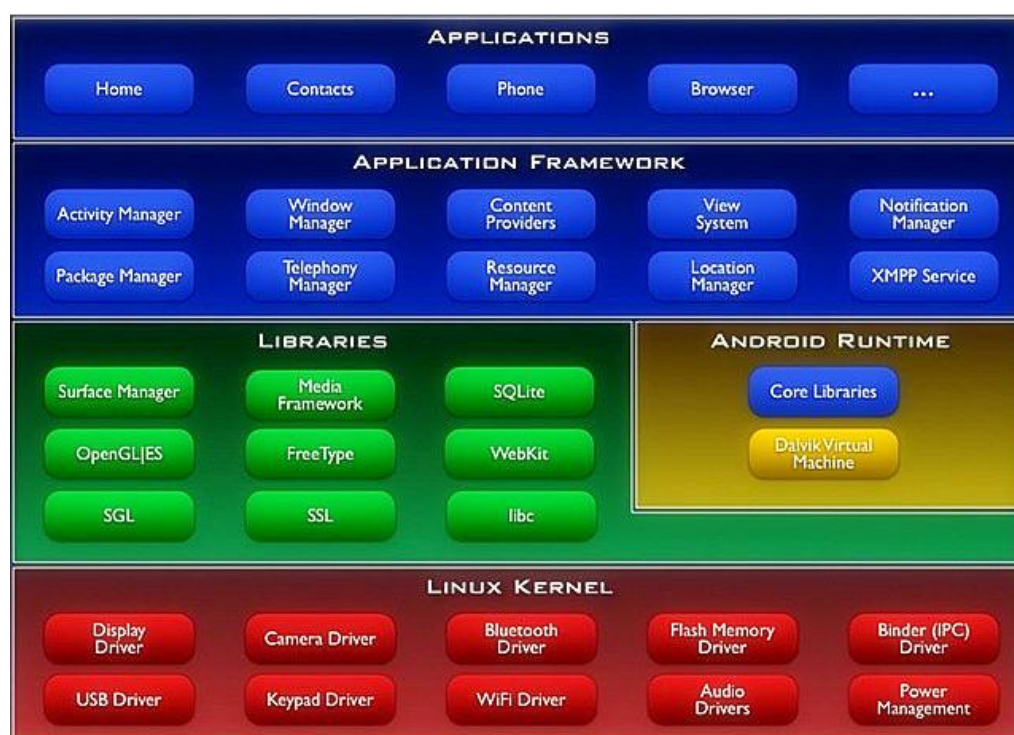


图 2.3 Android 系统框架

**Android** 的一个关键元素就是虚拟机。**Android** 使用了特定的虚拟机来保证多个应用可以高效地运行在一个设备上，而不是使用传统的 **Java** 虚拟机，比如 **Java ME**。**Dalvik VM** 使用设备的底层 **Linux** 内核来处理基本的功能，包括安全、线程以及进程和存储管理。编写直接运行在底层 **Linux OS** 上的 **C/C++** 应用程序也是可以的。尽管可以这样做，但大部分情况下没有这个必要。所有的 **Android** 硬件和系统服务访问都是使用作为中间层

的 Dalvik 来进行管理的。通过使用一个 VM 来承载应用程序的执行，开发人员就可以获得一个抽象层来保证他们永远都不需要考虑特定的硬件实现。Dalvik VM 执行 Dalvik 可执行文件，优化后的格式可以保证能最小限度地占用内存。使用 SDK 提供的工具，可以把 Java 语言编译的类转换为.Dex 可执行文件。

#### 2.4.2 Android 平台优势

1) 开放性。在优势方面，Android 平台首先就是其开发性，开发的平台允许任何移动终端厂商加入到 Android 联盟中来。显著的开放性可以使其拥有更多的开发者，随着用户和应用的日益丰富，一个崭新的平台也将很快走向成熟。开发性对于 Android 的发展而言，有利于积累人气，这里的人气包括消费者和厂商，而对于消费者来讲，最大的受益正是丰富的软件资源。开放的平台也会带来更大竞争，如此一来，消费者将可以用更低的价格购得心仪的手机。

2) 丰富的硬件。这一点还是与 Android 平台的开放性相关，由于 Android 的开放性，众多的厂商会推出千奇百怪，功能特色各具的多种产品。功能上的差异和特色，却不会影响到数据同步、甚至软件的兼容，如同从诺基亚 Symbian 风格手机一下改用苹果 iPhone，同时还可将 Symbian 中优秀的软件带到 iPhone 上使用、联系人等资料更是可以方便地转移。

4) 方便开发。Android 平台提供给第三方开发商一个十分宽泛、自由的环境，不会受到各种条条框框的阻扰，可想而知，会有多少新颖别致的软件会诞生。但也有其两面性，血腥、暴力、情色方面的程序和游戏如何控制正是留给 Android 难题之一。

5) Google 应用。在互联网的 Google 已经走过 10 年度历史，从搜索巨人到全面的互联网渗透，Google 服务如地图、邮件、搜索等已经成为连接用户和互联网的重要纽带，而 Android 平台手机将无缝结合这些优秀的 Google 服务。

## 3 智能运动服装功能设计

### 3.1 户外智能运动服装功能设计需求分析

#### 3.1.1 感知环境

户外活动对周围环境天气的关注是必不可少的，为了能让穿着者对当前环境有所了解，户外智能服装需要具备对周围环境温湿度的感知，以直观的方式让穿着者得知，还可以当周围环境超出运动适宜条件的时候，来对穿着者进行反馈。

#### 3.1.2 感知人体生理指标

户外运动最受关注就是穿着者的生理状态，基本包括血压、心率、体温、呼吸、肌肉状况、血糖情况，还有步数、卡路里消耗，将这些相关智能传感器多功能集成化，然后综合分析得出实时人体生理指标，用以判断当前穿着者状态是否适合运动。

#### 3.1.3 地理位置信息

GPS 全球定位功能是户外必不可少，此功能是预防穿着者发生意外或迷失方向事故，为穿着者提供户外运动安全的保障。

#### 3.1.4 通信网络稳定

在信息化的时代，稳定的通信网络显得非常重要，这将决定一个产品是否实用可靠的重要标志之一。通信网络的稳定性主要取决于通信信道带宽、数据通信合理以及通信异常处理方式。选择成熟稳定畅通的通信网络方式及其重要。

#### 3.1.5 合身舒适安全

既然是服装，那穿在身上合身舒适是非常重要的。安全是我们每个人都不可忽视的，因为智能服装要做到合身舒适安全，这也是最基本的要求。

### 3.1.6 方便移动能源

除非固定不动的，一般的智能服装都面临一个这样的问题，就是它的能源需要与自身本体一起移动。这样就迫使设计者不可避免地选择移动电源。国外的研究者在智能服装系统的研究中采用柔性太阳能电池。本论文主要是对户外智能服装功能设计的研究，对系统供电方面涉及不是很深入，采用移动电源完成系统的框架设计。

移动电源不是一次性设备，它可以反复使用数百次以上。当移动电源电能使用完后，必须给移动电源充电。因此其内部必须有充电管理系统。充电管理系统能根据锂电的电压，自动调节充电电流。该过程包括：预充，恒压充电和浮充充电等。

### 3.1.7 双向防丢功能

遗忘是每个人的有具备的一种不良习惯，出门在外遗忘携带是非常正常的现象，为此智能服装具备双向防丢功能便成为一种必要的实用功能，增加智能服装友好度。

## 3.2 户外智能运动服装功能总体设计

### 3.2.1 智能运动功能方案研究和设计

由于整个系统以服装为载体，具有可穿戴性，因此其设计与传统便捷式检测仪有着很大的不同。系统与服装结合的穿戴方式直接影响到系统的可穿戴性、用户的感受性和系统的物理结构。作为人机交互的一个重要的方面，穿戴方式的设计必须综合考虑人体工学和系统的逻辑、物理结构。本节对该方案设计准则进行初步探讨，并提出大体设计方案。

#### 1) 设计准则<sup>[12]</sup>

第一，基于人体可穿戴面积的考虑，包括成年人之间差异不大的区域、运动时相对固定或弹性较大的区域、承载表面积较大。

第二，人体运动对可穿戴方式的影响，在人体运动和保持各种常用姿势时，智能服装不应使人体产生严重的不适感，不影响人的关节活动。

第三大脑对穿戴设备的认可度，对于人体周边一定区域内的附着物，大脑会将它们当做人体的一部分。如果将系统微型处理器分布在这一区域内，则大脑将会准确的感知它们的位置，进而可以使它们避免被碰撞。

第四，基于与人体接触面舒适性和牢固性的考虑，可穿戴系统在于人体接触的内表面不应该有可能对人体造成伤害的锋利物，应该能与人体柔软接触，且接触应牢固附着人体。

第五，重量不应影响到人体的运动和平衡，在进行可穿戴方案的具体布局时，应尽量使前后，左右的重量相当，避免使人体产生一边重的感觉。在重量分布上，应该遵循重量分散的原则。较重的部分应分布在人体的几个承力点上，例如：腰部。

第六，根据就近原则对各模块进行布局，在遵循上述原则之后还需要考虑将连接较多的部分放在一起，减少连线，进行布局优化。

第七，美观考虑，可以尽量将体积较大的部分安排在不明显的地方，以减少对穿着者外观影响。

## 2) 服装可穿戴方案设计

第一，穿戴位置的选择，选择穿戴位置时遵循上述原则，因此我们选择了颈下、胸部下方和腰部三个环形区域以及上臂尾端和前臂作为主穿戴区域。

第二，各模块组件的布局，如图 3.1 所示。

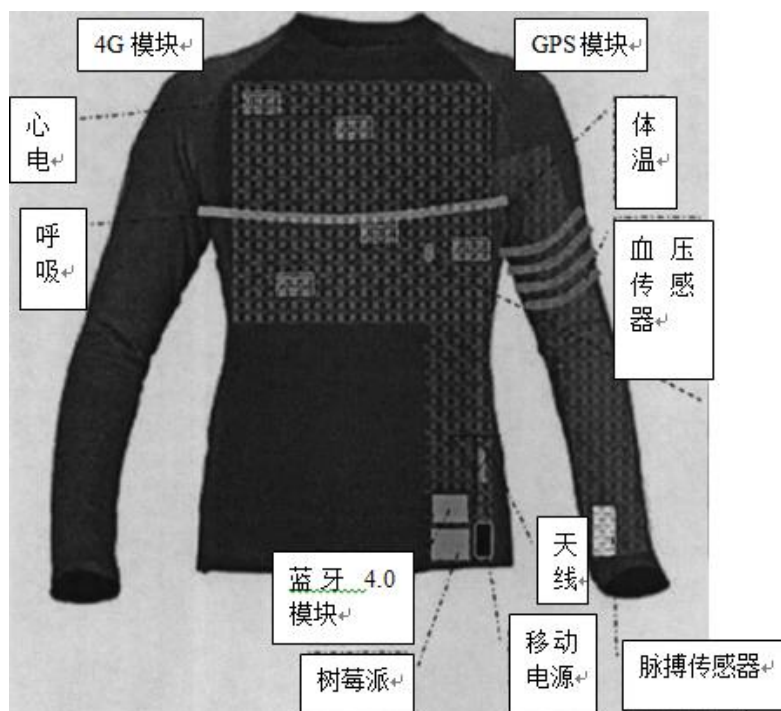


图 3.1 户外智能运动服装可穿戴设计方案图

### 3.2.2 智能运动服装功能设计系统框架

针对需求，如图 3.2 为智能运动服装功能设计的自拟方案的系统框架。

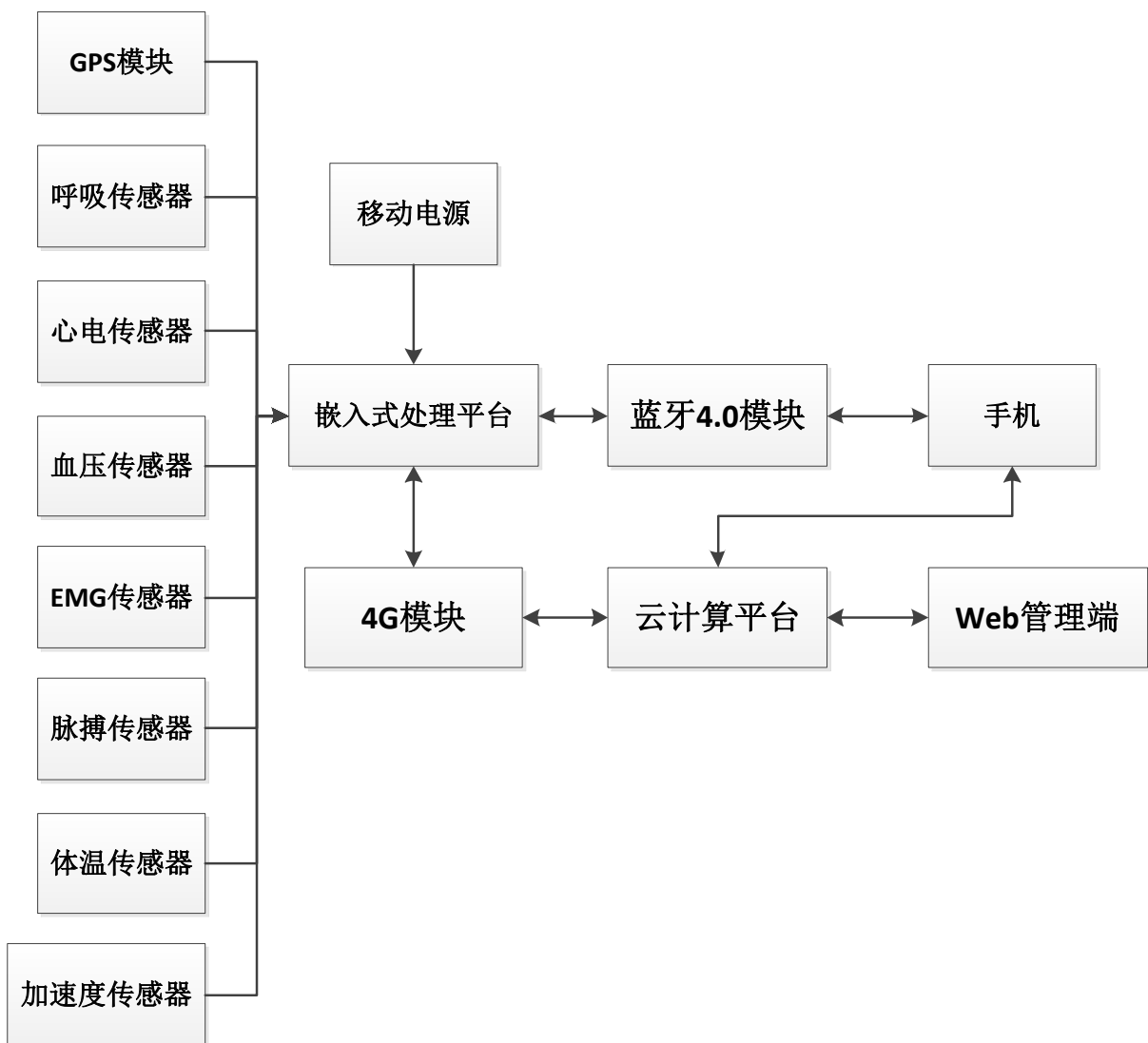


图 3.2 智能运动服装功能设计系统框架

针对户外智能运动服装功能设计系统框架，在具体硬件实施时有两种数据采集模式可供选择，包括集中式采集模式和集散式采集模式。

集中式采集模式是一个单处理器系统。嵌入服装的各种生理信号的采集模块独立采集信息，并通过编织在服装内的导体来传递生理信号。该模式中，中央处理器通过控制一个多路选择开关来选择当前所采集的信号，并对所有生理信号进行数字信号处理、信号分析。因此，在此模式下中央处理器应采取高性能的数字信号处理器。

集散式采集模式是一个多处理器系统。由一个中央处理器和多个信号采集协处理器组成<sup>[13]</sup>。每个协处理器负责采集相应的生理信号，并对生理信号做去噪和滤波等预处理。处理完毕后，通过嵌入服装的无线模块将结果送往中央处理器做统一处理和存储。

通过对两种采集系统的比较不难发现，集中式模式组成简单，成本较低，系统功率

低，但是存在效率低，抗干扰性差，可扩充性差等缺点；而集散式采集模式能够实现资源共享、效率高、抗干扰性强，可扩充性好，但是成本较高，系统功率较高。高可靠性和低功率是智能服装系统是否实用的关键指标。当智能服装系统用于对人体健康状况的监测时，我们根据具体实际应用场景力争在高可靠性与低功率达到相对理想的平衡点来最大限度满足需求。本论文使用集中式模式。

为了使智能运动服装功能设计项目的顺利进行，下面来描述一下该项目方案的具体内容。以上是根据目前市面上多种智能可穿戴设备的功能实施方式设计的。该方案是通过心电传感器、呼吸传感器、脉搏传感器、体温传感器、血压传感器、EMG 传感器、加速度传感器和温湿度传感器，采用采用高数据传输率，支持高层协议的高速总线（USB、Firewire）方式、集中式采集模式与目前市面上热门的微型计算机树莓派作为主控板进行数据通信，并且树莓派可携带数据库存储功能，使得传感器的数据存储得到不丢失的信息安全保障。在电源条件允许的情况下，树莓派一直实时对传感器所采集的数据进行处理、存储、传输。

GPS 模块主要解决地理位置信息问题，使穿着者户外运动不幸出现意外造成行动不便或迷失方向，也能被找回。4G 模块主要是解决智能服装与云计算通信连接的问题，使得云计算数据实时更新。由于 GPS 模块与 4G 模块的使用会造成耗电巨大，因此需合理制定运行计划方案来达到节能，增加设备续航能力。

Android APP 与树莓派的通信方式之所以选择蓝牙 4.0，是因为该通信方式是目前最流行且极其稳定的低功耗的通信方式，尤其是适用于目前市场上大部分手机的需求。蓝牙 4.0 优点有以下优点：1、蓝牙 4.0 将三种规格集一体，包括传统蓝牙技术、高速技术和低功耗技术，与 3.0 版本相比最大的不同就是低功耗。2、低功耗版本使蓝牙技术得以延伸到采用纽扣电池供电的一些新兴市场。蓝牙低功耗技术是基于蓝牙低功耗无线技术核心规格的升级版，为开拓钟表、远程控制、医疗保健及运动感应器等广大新兴市场的应用奠定基础。3、这项技术将应用于每年出售的数亿台蓝牙手机、个人电脑及掌上电脑。以最低耗能提供持久的无线连接，有效扩大相关应用产品的覆盖距离，开辟全新的网络服务。低功耗无线技术的特点在于超低的峰期、平均值及待机耗能；使装置配件和人机介面装置(HIDs)具备超低成本和轻巧的特性；更能使手机及个人电脑相关配件的成本降至最低、体积更小；全球适用之外，更具使用直觉，且能确保多种设备连接的互操作性。4、蓝牙 4.0 在个人健身和健康市场的影响很大。无论是在跑步机上，或在办公

室的小工具。Fitbit 无线师，耐克公司的新 Fuelband，摩托罗拉 MOTACTV，和时尚的基带是都是很好的例子。而且健身手表也承诺使用蓝牙跟踪体力活动和心率。另外蓝牙 4.0 依旧向下兼容，包含经典蓝牙技术规范 and 最高速度 24Mbps 的蓝牙高速技术规范。三种技术规范可单独使用，也可同时运行。其应用也极为广泛，这项技术可为制造商及用户提供三种无线连接方式，包括用于多个类别电子消费产品的传统蓝牙技术；用于手机、相机、摄像机、PC 及电视等视讯、音乐及图片传输的蓝牙高速技术；以及用于保健及健康、个人设备、汽车及自动化行业的低功率传感设备和新的网络服务的蓝牙低耗能技术。蓝牙 4.0 还可以检测信号强度，这就可以计算连接设备之间的距离，从而实现防丢功能。

移动终端选择 Android 系统。Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。该系统占据市场上大部分智能手机。该平台的优势在于其开放性、丰富的硬件、方便开发以及拥有丰富的 Google 应用支持。Android APP 开发涉及蓝牙 4.0 通信、数据库增删改查操作、数据展示方式三方面技术。

### 3.3 户外智能运动服装功能各模块设计

#### 3.3.1 智能传感器

针对需求，本论文使用的智能传感器如表 3.1 所示。

表 3.1 使用的智能传感器

智能传感器类型	型号	通信接口	用途
呼吸传感器	HKH-11B	模拟信号	检测呼吸深度及频率
心电传感器	HKD-10C	USB 串口	检测心跳活动情况
血压传感器	HKB-08B	USB 串口	检测血压
EMG 传感器	Duinopeak	IO 口	检测肌肉活动情况
脉搏传感器	HKG-07D	USB 串口	检测脉搏
体温传感器	HKT-09	USB 串口	检测体温
加速度传感器	ADXL345	I <sup>2</sup> C /SPI	检测步数，计算卡路里消耗

#### 3.3.2 4G 模块及 GPS 模块

4G 模块主要解决智能服装在无 WIFI 条件下通过互联网连接到云计算，并进行数据通信，正常情况下户外活动都没有 WIFI 提供。GPS 模块主要是通过卫星定位获取当前



经纬度，然后通过 4G 模块告知云计算平台，云计算平台推送到其它终端。

### 3.3.3 树莓派（包括移动电源）

针对需求，本论文采用树莓派（Raspberry Pi）作为主控板。树莓派由注册于英国的慈善组织“Raspberry Pi 基金会”开发，Eben Upton/埃·厄普顿为项目带头人。2012 年 3 月，英国剑桥大学埃本·阿普顿（Eben Epton）正式发售世界上最小的台式机，又称卡片式电脑，外形只有信用卡大小，却具有电脑的所有基本功能，这就是 Raspberry Pi 电脑板，中文译名“树莓派”。它是一款基于 ARM 的微型电脑主板，以 SD/MicroSD 卡为内存硬盘，卡片主板周围有 1/2/4 个 USB 接口和一个 10/100 以太网接口（A 型没有网口），可连接键盘、鼠标和网线，同时拥有视频模拟信号的电视输出接口和 HDMI 高清视频输出接口，以上部件全部整合在一张仅比信用卡稍大的主板上，具备所有 PC 的基本功能只需接通电视机和键盘，就能执行如电子表格、文字处理、玩游戏、播放高清视频等诸多功能。Raspberry Pi B 款只提供电脑板，无内存、电源、键盘、机箱或连线。并且树莓派官方还提供所有电路原理图，商家可以根据自家产品特性来选择裁剪没有必要的引脚，以达到电子最微小化。树莓派的主要功能包括与智能传感器的数据通信、数据处理、数据存储以及与无线通信模块的数据通信。本论文系统电源部分可携带移动电源作为整个系统的供电，达到可方便携带的目的。

### 3.3.4 蓝牙 4.0 模块

针对需求，本论文采用蓝牙 4.0 无线通信模块，主要用于树莓派与手机客户端的数据通信。蓝牙 4.0 是蓝牙 3.0+HS 规范的补充，专门面向对成本和功耗都有较高要求的无线方案，可广泛用于卫生保健、体育健身、家庭娱乐、安全保障等诸多领域。它支持两种部署方式：双模式和单模式。双模式中，低功耗蓝牙功能集成在现有的经典蓝牙控制器中，或再在现有经典蓝牙技术(2.1+EDR/3.0+HS)芯片上增加低功耗堆栈，整体架构基本不变，因此成本增加有限。单模式面向高度集成、紧凑的设备，使用一个轻量级连接层(Link Layer)提供超低功耗的待机模式操作、简单设备恢复和可靠的点对多点数据传输，还能让联网传感器在蓝牙传输中安排好低功耗蓝牙流量的次序，同时还有高级节能和安全加密连接。主要负责树莓派与手机客户端的数据通信。

3.3.5 Android APP

针对需求，选择 Android 4.4 版本，Android 系统采用 AChartEngine 库来绘制美观图表，采用 SQLite 数据库来进行数据增删改查操作以及蓝牙 4.0 通信编程来完成手机客户端软件开发。手机客户端主要功能包括实时显示用户的脉搏活动情况、对历史数据的更新、存储和查询以及防丢报警。

3.3.6 云计算平台通信接口设计

云计算平台是为系统大数据平台的前提基础，其主要职能是为系统存储所有户外智能服装采集的数据，并对大量数据进行合理分析来得到对穿着者有用建议，这是一种类似于人工智能的体现，通过大量数据让专业人士给予运动意见，改善运动计划，让穿着者在科学分析的条件下进行最佳运动，得到最佳运动效果，而且还可以防止穿着者一些非健康式运动，这样大大减少穿着者不健康运动行为的出现。云计算平台主要提供三个通信接口，包括与硬件设备 4G 模块通信接口、与 Web 管理端通信接口以及手机通信接口。云计算平台主要是采用可视化数据进行一系列操作。图 3.3 所示为云计算平台接口设计。

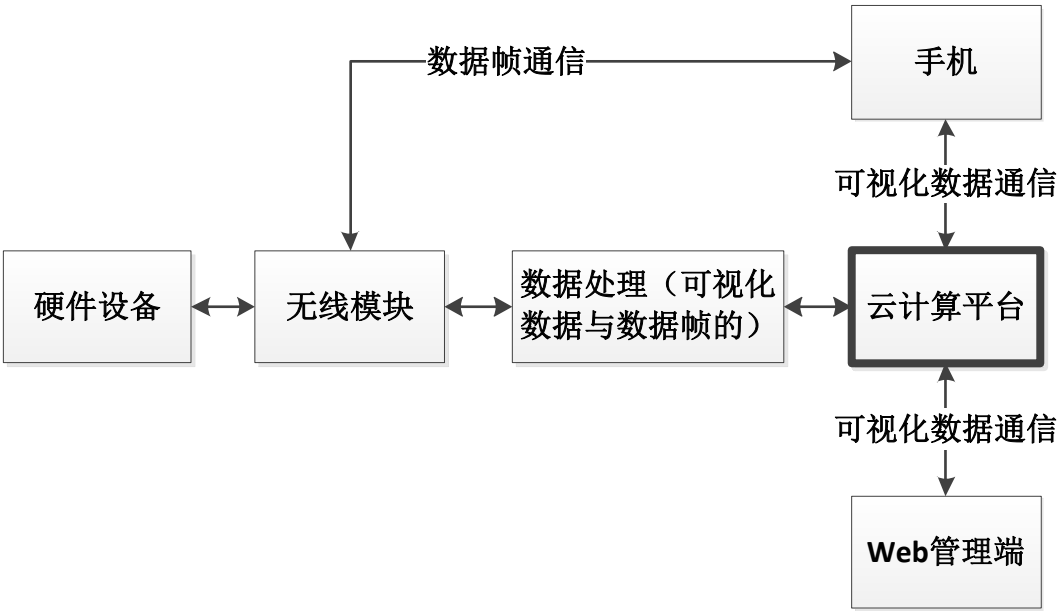


图 3.3 云计算通信接口

3.3.7 系统通讯协议设计

本论文设计的通讯方式主要通过蓝牙 4.0 模块该模块进行通讯特点，决定了整个系

统的通讯协议，以此达到系统通讯协议统一及规范。蓝牙 4.0 模块收发数据包为 20 个字节，如果超出则会自动进行拆包分发处理。为此，我们根据该特点来设计合理的通信协议，包括字符串形式和数据帧形式。字符串形式，该形式数据直观，方便数据库直接存储获取，但不利于大数据传输。由于本论文实现在早期未考虑形式问题，因此采用了该通信协议方式。数据帧形式，该形式采用十六进制统一规范所有数据，可包含更多的信息量，但终端需要对数据进行处理才可获得直观信息。实际应用中，该形式在智能硬件中更适用。表 3.2 所示为通信协议数据帧设计，表 3.3 所示为数据帧字段含义，表 3.4 所示为自定义智能传感器类型指令，以及根据各类智能传感器对控制字与数据进行具体设计。

表 3.2 通信协议数据帧设计

帧头标识	数据源头	智能传感器类型	控制字	数据	数据帧长度	校验和
1 字节	1 字节	1 字节	1 字节	多字节	1 字节	1 字节

表 3.3 数据帧字段含义

数据帧字段	含义
帧头标识	0xf0（固定）
数据源头	0x01(回馈)或 0x00(发送)
智能传感器类型	自定义智能传感器类型指令
控制字	根据智能传感器自定义功能指令
数据	根据智能传感器获取的具体数据
数据帧长度	一个数据帧的字节长度
校验和	所有字段的和的低字节

表 3.4 自定义智能传感器类型指令

智能传感器类型	类型指令
呼吸传感器	0x01
心电传感器	0x02
血压传感器	0x03
EMG 传感器	0x04
脉搏传感器	0x05
体温传感器	0x06
加速度传感器	0x07

## 4 智能运动服装功能实现

### 4.1 智能运动服装功能各模块实现

#### 4.1.1 血压传感器

1) 该通信协议采用全双工串行通讯接口，一个起始位、8 个数据位、1 个停止位、无奇偶校验位，波特率 115200bps。

2) 通信协议数据帧设计如表 4.1 所示。

表 4.1 通信协议数据帧设计

帧头标识	数据帧长度	校验和	控制字	数据
2 字节	1 字节	1 字节	1 字节	多字节

3) 血压传感器指令表如表 4.2 所示。

表 4.2 血压传感器指令表

指令	发送	回馈
休眠指令	0xff,0xcd,0x03,CKSUM,0xab	0xff,0xcd,0x03,CKSUM,0x503
唤醒指令	0xff,0xcd,0x03,CKSUM,0xaa	0xff,0xcd,0x03,CKSUM,0x5a
启动指令	0xff,0xcd,0x03,CKSUM,0xa0	0xff,0xcd,0x05,CKSUM,0x54, QYH,QYL
停止指令	0xff,0xcd,0x03,CKSUM,0xa3	0xff,0xcd,0x03,CKSUM,0x53
测试结果		0XFF 0XCD 0X08 CKSUM 0X55 SSYH SSYL SZYH SZYL XL
错误报告		0XFF 0XCD 0X04 CKSUM 0X56 X

4) 测量结果数据含义表如表 4.3 所示。

表 4.3 测量结果数据含义表

数据	含义
QYH	测量过程中气压输出高字节，其中 QYH.4=1 表示有心跳，QYH.4=0 表示无心跳
QYL	气压输出低字节
SZYH、SZYL	舒张压
SSYH、SSYL	收缩压，其中 SSYH.7=1 表示心率不齐，SSYH.7=0 表示心率正常
XL	心率

5) 错误报告数据含义表如表 4.4 所示。

表 4.4 错误报告数据含义表

数据	含义
X=0	测量不到有效的脉搏
X=1	11S 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	气袋压力超过 295mmHg .进入超压保护
X=4	干预过多（测量中移动、说话等）

#### 4.1.2 体温传感器

1) 该通信协议采用全双工串行通讯接口，一个起始位、8 个数据位、1 个停止位、无奇偶校验位，波特率 57600bps。

2) 通信协议数据帧设计如表 4.5 所示。

表 4.5 通信协议数据帧设计

帧头标识	控制字	数据	校验和
1 字节	1 字节	多字节	1 字节

3) 体温传感器指令表如表 4.6 所示。

表 4.6 体温传感器指令表

指令	发送	回馈
启动上传指令	0xf0,0xc0,0xb0	0xf0,0xc0,0xAA,0xaa,0xb0
关闭上传指令	0xf0,0xc1,0xb1	0xf0,0xc1,0xb1

4) 接收数据格式表如表 4.7 所示。

表 4.7 接收数据格式表

帧头字节	控制字	数据高位	数据低位	校验位
F0	C0	AA	aa	CKSUM

0xAAaa = 当前体温（单位：0.1 摄氏度）

#### 4.1.3 脉搏传感器

1) 该通信协议采用全双工串行通讯接口，一个起始位、8 个数据位、1 个停止位、无奇偶校验位，波特率 19200bps。

2) 通信协议数据帧设计如表 4.8 所示。

表 4.8 通信协议数据帧设计

帧头标识	控制字	数据	校验和
1 字节	1 字节	多字节	1 字节

3) 脉搏传感器指令表如表 4.9 所示。

**表 4.9 脉搏传感器指令表**

指令	发送	回馈
启动上传指令	0xf0,0xc0,0xb0	0xf0,0xc0,0xAA,0xaa,0xb0
关闭上传指令	0xf0,0xc1,0xb1	0xf0,0xc1,0xb1

4) 接收数据格式表如表 4.10 所示。

**表 4.10 接收数据格式表**

帧头字节	控制字	数据高位	数据低位	校验位
F0	C0	AA	aa	CKSUM

0xAAaa = 当前脉搏（单位：次数）

5) 性能参数：硬件接口为高数据传输率，支持高层协议的高速总线（USB）。数据发送方式为每次心跳输出一个数据，非测试状态或人体动作过大造成的无效信号输出 0。供电方式为 USB 供电。

#### 4.1.4 蓝牙 4.0 模块

##### 1、蓝牙 4.0 收发服务属性

- 1) 发送数据属性：0000fff2-0000-1000-8000-00805f9b34fb
- 2) 接收数据属性：0000fff1-0000-1000-8000-00805f9b34fb

##### 2、蓝牙通信协议

数据形式：采用字符串形式，如表 4.11 所示。

**表 4.11 蓝牙通信协议**

指令语义	指令格式	备注
实时脉搏数据	pul:80	当前脉搏：80 次
更新数据库	update20160101000000	从 2016-01-01 00:00:00 更新到最后
数据库更新完毕回馈 或者主动结束更新	finish	
数据库更新方式（以每 条数据为一帧）	20160227000000,100	2016-02-27 00:00:00 为 100 脉搏

#### 4.1.5 树莓派

##### 1、树莓派的编程环境

1) `pySerial` 封装了对串口的读写访问方式，在支持的平台上有一个统一的接口。通过 `python` 属性访问串口设置。支持不同的字节大小、停止位、校验位和流控设置。可以有或者没有接收超时。类似文件的 API,例如 `read` 和 `write`，也支持 `readline` 等。支持二进制传输，没有 `null` 消除，没有 `cr-lf` 转换。

2) `MySQL` 是一个关系型数据库管理系统，由瑞典 `MySQL AB` 公司开发，目前属于 `Oracle` 旗下公司。`MySQL` 最流行的关系型数据库管理系统，在 `WEB` 应用方面 `MySQL` 是最好的 `RDBMS` (`Relational Database Management System`，关系数据库管理系统) 应用软件之一。`MySQL` 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。`MySQL` 所使用的 `SQL` 语言是用于访问数据库的最常用标准化语言。`MySQL` 软件采用了双授权政策（本词条“授权政策”），它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 `MySQL` 作为网站数据库。由于其社区版的性能卓越，搭配 `PHP` 和 `Apache` 可组成良好的开发环境。`MySQL` 主要是用于数据存储。

3) `Python MySQL` 是 `Python` 数据库接口支持 `MySQL` 数据库的库。`Python` 标准数据库接口为 `Python DB-API`，`Python DB-API` 为开发人员提供了数据库应用编程接口。而 `Python MySQL` 就是其中之一。`Python MySQL` 主要是用于 `pySerial` 与 `MySQL` 数据库的结合开发。

## 2、树莓派的编程功能

- 1) 与智能传感器通过 `USB` 串口通信实现收发数据。
- 2) 与蓝牙模块通过有线串口通信实现收发数据。
- 3) 把数据帧处理成自定义数据，并数据存储和传递。
- 4) 树莓派的编程逻辑如图 4.1 所示。
- 5) 设置开机自动启动该系统功能程序。

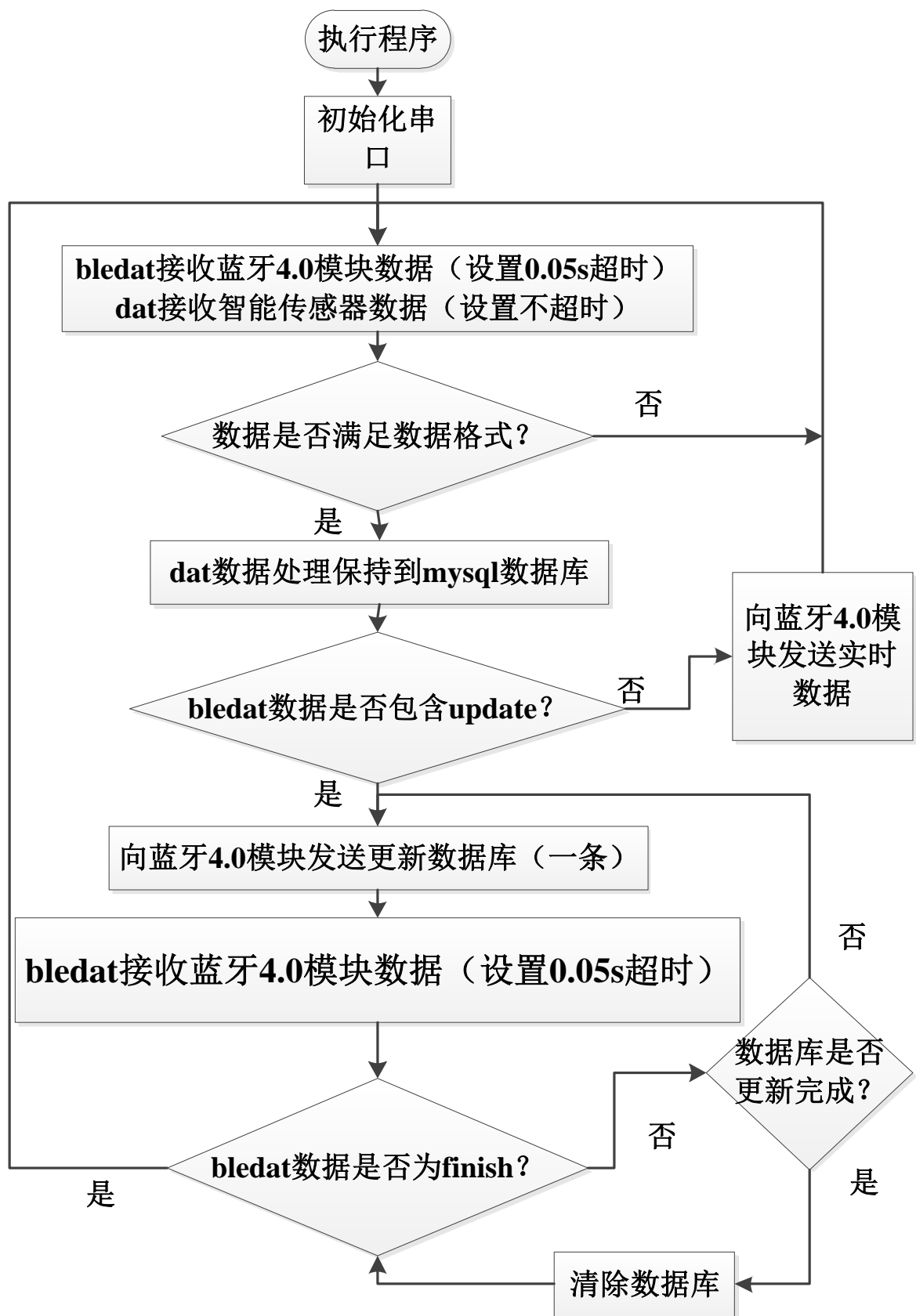


图 4.1 树莓派编程逻辑图



3、树莓派的数据库表格式如表 4.12 所示。

表 4.12 test 表的格式

字段名称	id	pulse_times	date_time	log
字段类型	int(10)	smallint(5)	timestamp	tinytext

4.1.6 Android APP

1、 Android 应用软件逻辑示意如图 4.2 所示。

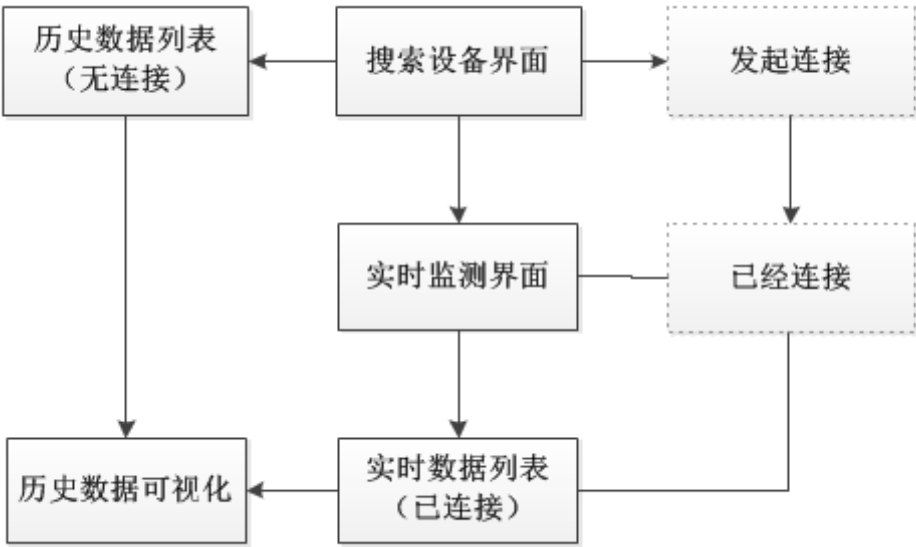
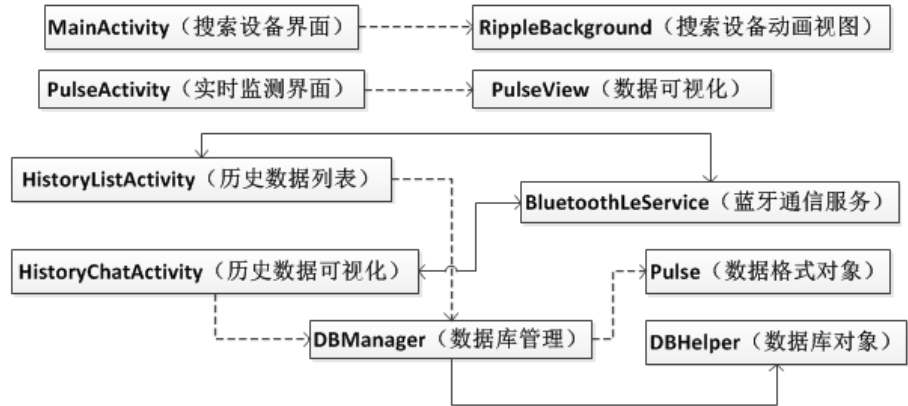


图 4.2 Android 应用软件逻辑示意图

2、 Android 应用软件类如图 4.3 所示。



备注：关联是一种拥有关系，依赖是一种使用关系。

图 4.3 Android 应用软件类图

3、 Android 应用软件数据库表格式如表 4.13 所示。

表 4.13 pulse 表的格式

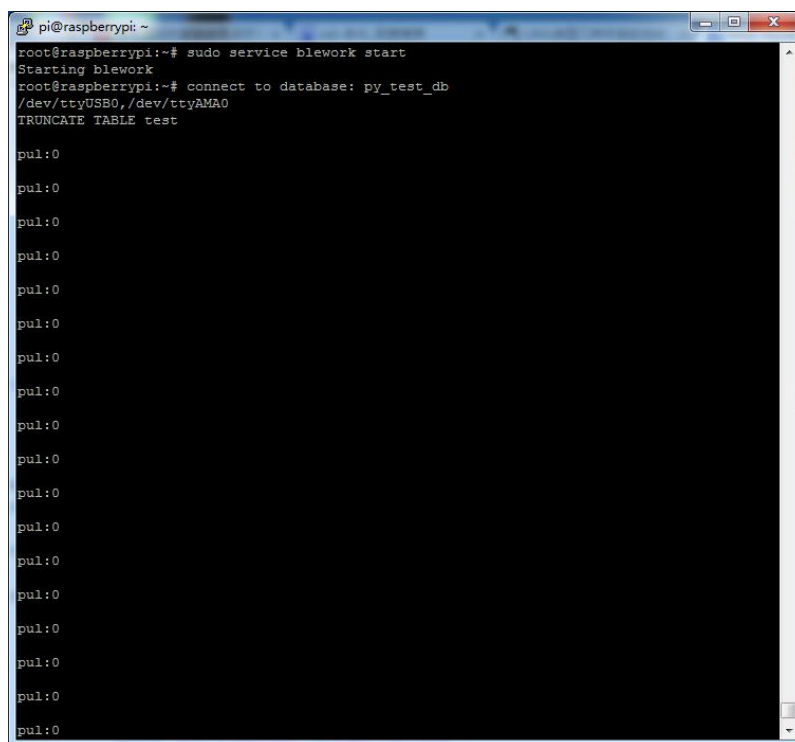
字段名称	_id	value	time
字段类型	INTEGER	INTEGER	TIMESTAMP

#### 4、Android 应用软件实现功能

- 1) 智能服装防丢报警功能
- 2) 智能服装脉搏实时测量
- 3) 智能服装脉搏历史翻查

## 4.2 智能运动服装功能实现展示

- 1) 程序启动后进行实时监测如图 4.4 所示。



```

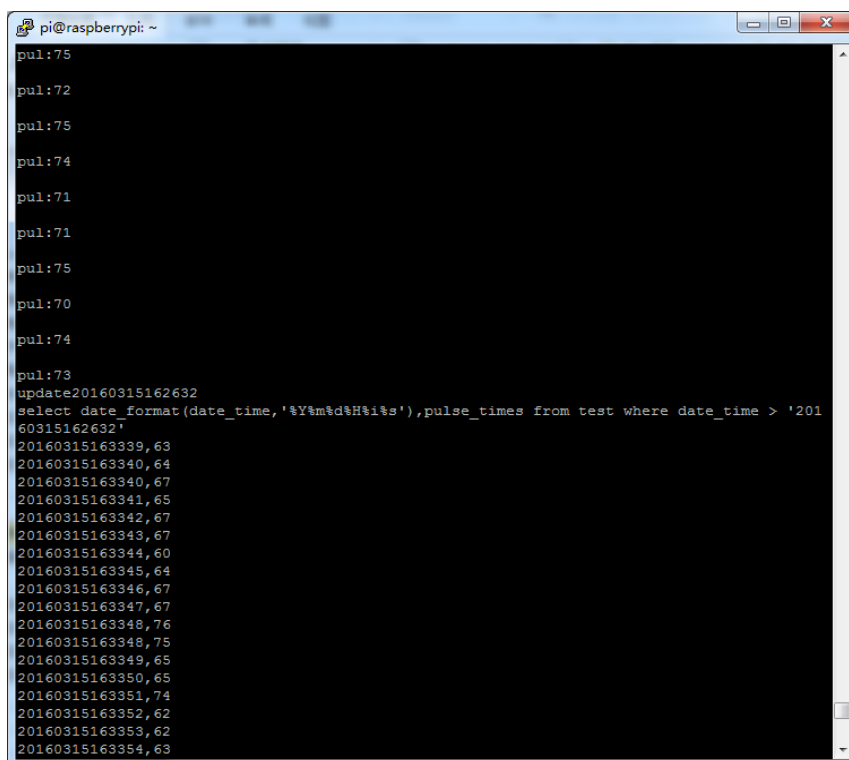
pi@raspberrypi: ~
root@raspberrypi:~# sudo service blework start
Starting blework
root@raspberrypi:~# connect to database: py_test_db
/dev/ttyUSB0,/dev/ttyAMA0
TRUNCATE TABLE test

pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0
pul:0

```

图 4.4 程序启动后进行实时监测

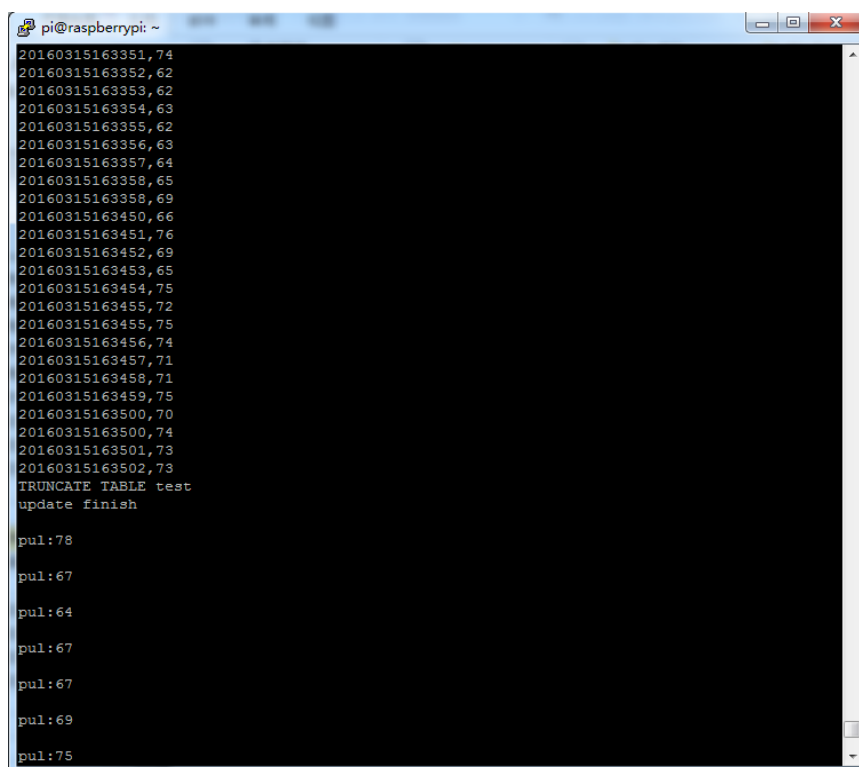
2) 正在更新客户端的历史数据如图 4.5 所示。



```
pi@raspberrypi: ~  
pul:75  
pul:72  
pul:75  
pul:74  
pul:71  
pul:71  
pul:75  
pul:70  
pul:74  
pul:73  
update 20160315162632  
select date_format(date_time,'%Y%m%d%H%i%s'),pulse_times from test where date_time > '20160315162632'  
20160315163339,63  
20160315163340,64  
20160315163340,67  
20160315163341,65  
20160315163342,67  
20160315163343,67  
20160315163344,60  
20160315163345,64  
20160315163346,67  
20160315163347,67  
20160315163348,76  
20160315163348,75  
20160315163349,65  
20160315163350,65  
20160315163351,74  
20160315163352,62  
20160315163353,62  
20160315163354,63
```

图 4.5 正在更新客户端的历史数据

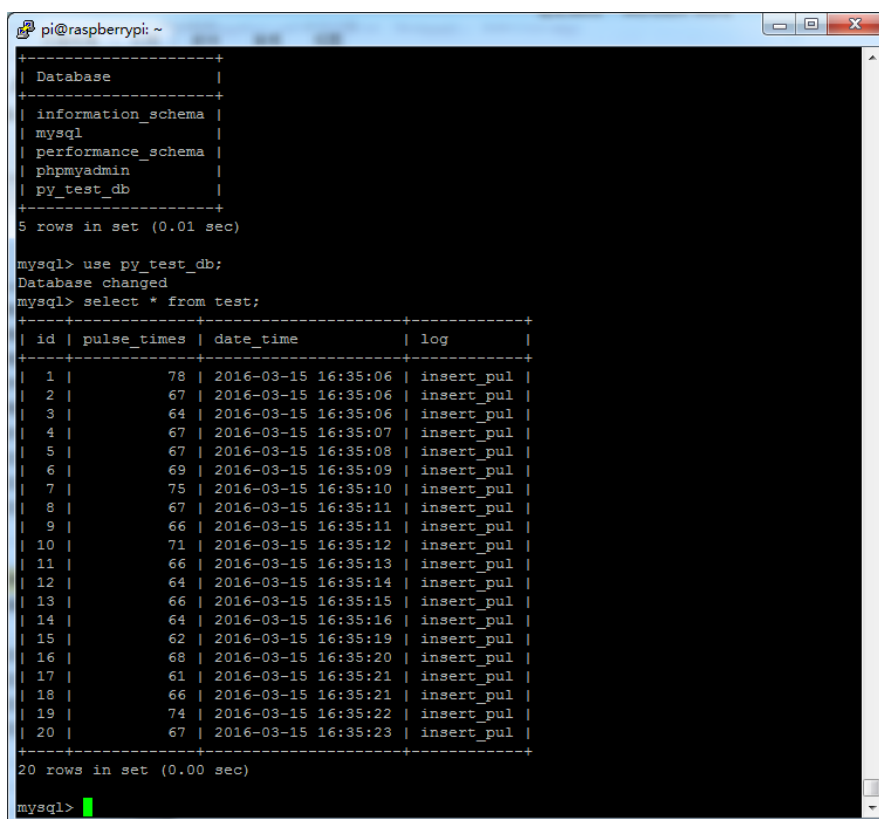
3) 客户端的历史数据更新完毕如图 4.6 所示。



```
pi@raspberrypi: ~  
20160315163351,74  
20160315163352,62  
20160315163353,62  
20160315163354,63  
20160315163355,62  
20160315163356,63  
20160315163357,64  
20160315163358,65  
20160315163358,69  
20160315163450,66  
20160315163451,76  
20160315163452,69  
20160315163453,65  
20160315163454,75  
20160315163455,72  
20160315163455,75  
20160315163456,74  
20160315163457,71  
20160315163458,71  
20160315163459,75  
20160315163500,70  
20160315163500,74  
20160315163501,73  
20160315163502,73  
TRUNCATE TABLE test  
update finish  
pul:78  
pul:67  
pul:64  
pul:67  
pul:67  
pul:69  
pul:75
```

图 4.6 客户端的历史数据更新完毕

4) 查询表数据如图 4.7 所示。



```
pi@raspberrypi: ~  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| phpmyadmin |  
| py_test_db |  
+-----+  
5 rows in set (0.01 sec)  
  
mysql> use py_test_db;  
Database changed  
mysql> select * from test;  
+-----+-----+-----+-----+  
| id | pulse_times | date_time | log |  
+-----+-----+-----+-----+  
| 1 | 78 | 2016-03-15 16:35:06 | insert_pul |  
| 2 | 67 | 2016-03-15 16:35:06 | insert_pul |  
| 3 | 64 | 2016-03-15 16:35:06 | insert_pul |  
| 4 | 67 | 2016-03-15 16:35:07 | insert_pul |  
| 5 | 67 | 2016-03-15 16:35:08 | insert_pul |  
| 6 | 69 | 2016-03-15 16:35:09 | insert_pul |  
| 7 | 75 | 2016-03-15 16:35:10 | insert_pul |  
| 8 | 67 | 2016-03-15 16:35:11 | insert_pul |  
| 9 | 66 | 2016-03-15 16:35:11 | insert_pul |  
| 10 | 71 | 2016-03-15 16:35:12 | insert_pul |  
| 11 | 66 | 2016-03-15 16:35:13 | insert_pul |  
| 12 | 64 | 2016-03-15 16:35:14 | insert_pul |  
| 13 | 66 | 2016-03-15 16:35:15 | insert_pul |  
| 14 | 64 | 2016-03-15 16:35:16 | insert_pul |  
| 15 | 62 | 2016-03-15 16:35:19 | insert_pul |  
| 16 | 68 | 2016-03-15 16:35:20 | insert_pul |  
| 17 | 61 | 2016-03-15 16:35:21 | insert_pul |  
| 18 | 66 | 2016-03-15 16:35:21 | insert_pul |  
| 19 | 74 | 2016-03-15 16:35:22 | insert_pul |  
| 20 | 67 | 2016-03-15 16:35:23 | insert_pul |  
+-----+-----+-----+-----+  
20 rows in set (0.00 sec)  
  
mysql>
```

图 4.7 查询表数据

5) Andorid 应用软件搜索设备如图 4.8 所示。

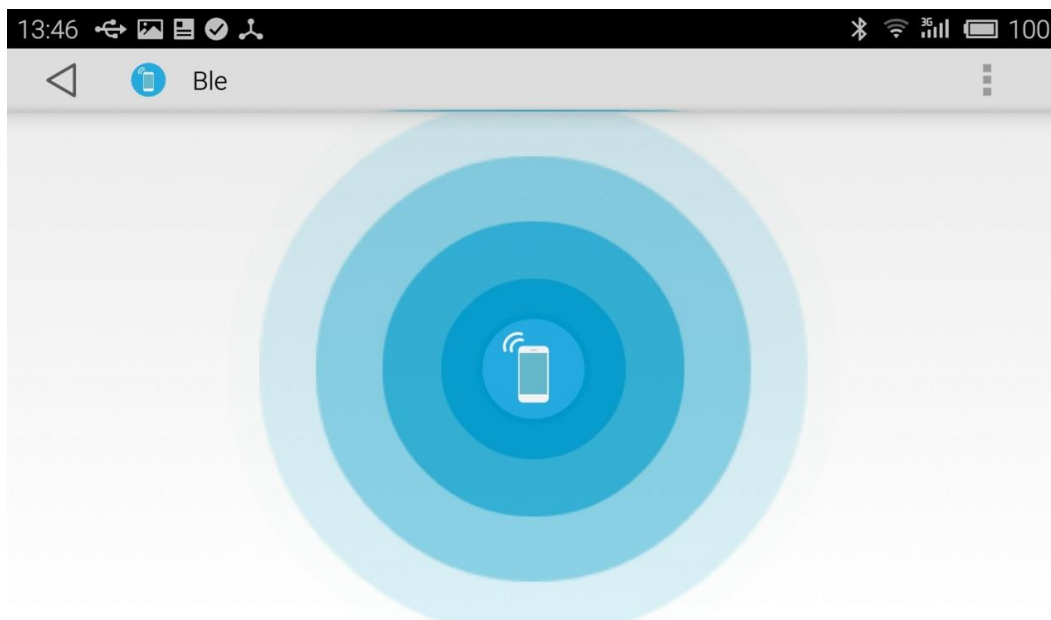


图 4.8 Android 应用软件搜索设备

6) Andorid 应用软件实时监测如图 4.9 所示。

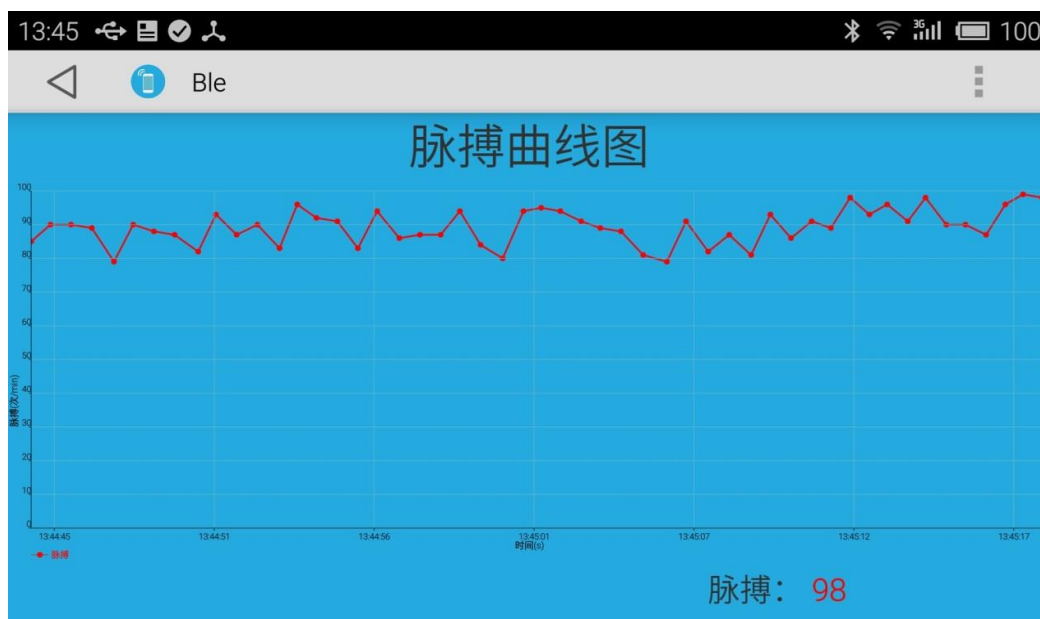


图 4.9 Android 应用软件实时监测

7) Andorid 应用软件历史数据列表如图 4.10 所示。

Figure 4.10 shows a screenshot of an Android application interface displaying a list of historical pulse data. The interface includes a status bar at the top with the time 13:45 and battery level 100. Below the status bar is a navigation bar with a back arrow, a 'Ble' label, and a 'STOP' button. The list of data is as follows:

2016-03-14 13:37:52	
~	
2016-03-14 13:38:33	
~	
2016-03-14 13:36:45	
~	
2016-03-14 13:37:52	
~	
2016-03-14 13:34:25	
~	
2016-03-14 13:36:45	

图 4.10 Android 应用软件历史数据列表

8) Andorid 应用软件历史数据可视化如图 4.11 所示。

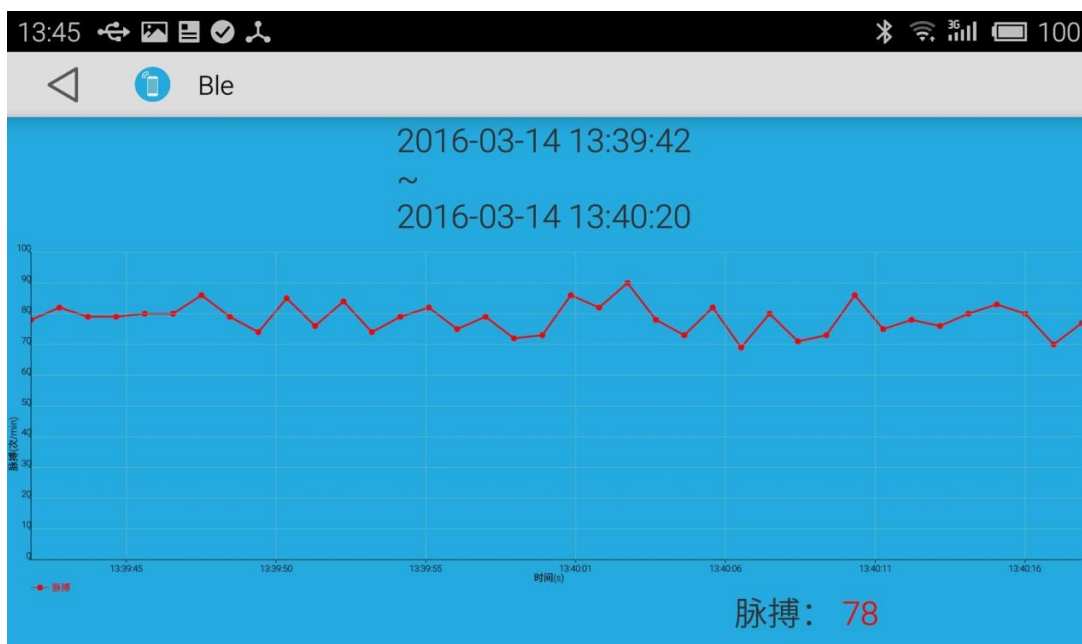


图 4.11 Android 应用软件历史数据可视化

9) 脉搏检测实物如图 4.12 所示。

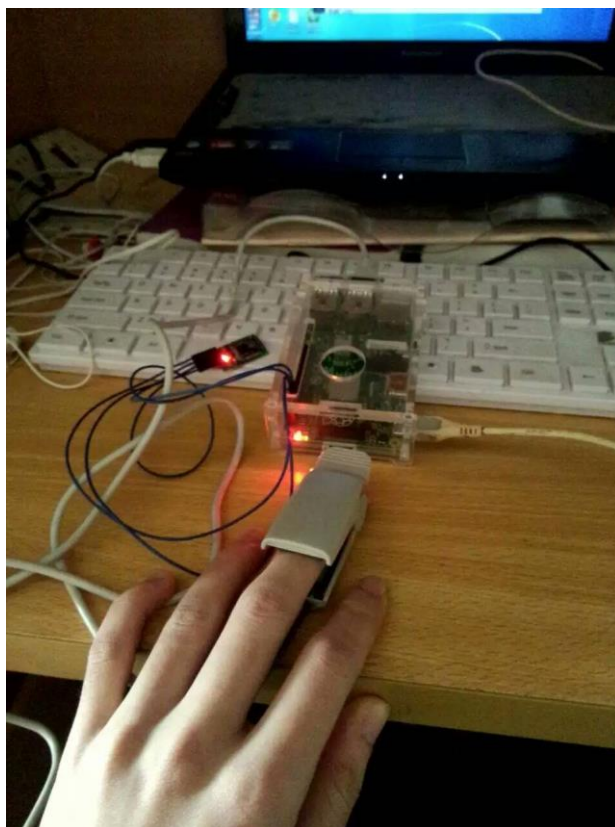


图 4.12 脉搏检测实物图

## 结 论

本论文针对人们对户外智能运动服装功能设计的需求，综合运用智能传感器、嵌入式处理主控板、无线通信技术、Android 应用软件开发等技术，设计和开发面向运动监测的智能服装系统的框架，并针对系统中主要模块设计与实现进行了探讨。论文的主要工作有：

首先，分析了智能服装的研发现状，给出了户外智能运动服装的研究目标与系统需求，介绍了智能服装的关键技术，根据系统体框架的体系结构分布对信息检测层、信息处理层和信息反馈层进行阐述。

其次，针对户外智能运动服装功能作详细设计，根据系统框架的体系结构分别对信息检测层、信息处理层和信息反馈层进行设计阐述以及针对该论文系统实现需求制定设计方案，依据人体力学及服装美学对户外智能服装原型进行设计，阐述了系统中每个功能模块的设计方案。

最后，针对户外智能运动服装功能设计来阐述实现情况，并且介绍涉及智能传感器、4G 模块与 GPS 模块、树莓派、蓝牙 4.0 模块、Android APP、云计算平台通信接口设计以及系统通信协议设计的关键细节。该设计是以可实施性为前提进行设计，其中关键技术已掌握，但数据格式处理以及后期数据分析还需进一步规范和改进。

## 展 望

本论文所研究和设计的户外智能运动服装功能系统,虽然基本达到预期的设计目标,但是这是一个复杂的系统,它涉及多个研究领域并且都在这些领域对研究者提出了挑战。

### 1) 新型智能材料

智能材料是一种对所给的特别的激励能进行判别并按预定方式做出反应的材料,具有感知功能、信息处理功能和执行功能,并且可以自动调解并具有自诊断、自适应、自修复、损伤抑制、寿命预报等能力,表现出动态的自适应性。

### 2) 智能传感器

智能传感器是指带微处理器、兼有信息检测和信息处理能力的传感器。智能传感器的最大特点就是传感器检测信息的功能与微处理器的信息处理功能有机地融合在一起。目前传感器正朝着智能化、小型化和低功耗的方向发展。只有传感器符合了这些特点,才能真正与智能有机结合在一起。

### 3) 低功耗芯片的研究

目前采用的只是现有的各大芯片厂商近年来相继推出功能强大、功耗低的 DSP、MCU 等芯片。但这些芯片离真正的应用于智能服装还有一定距离,这给芯片的研发带来了巨大的挑战。同时我们还要在现有低功耗器件的基础上研究优化的算法降低系统功耗。

### 4) 电源和能耗

智能服装往往是一个庞大的系统,传感器、处理器、通信等等这些都需要电能,因此一个好的系统设计应尽量降低系统的能耗,采用高性能和新型的电源。目前电池已经变得越来越小和轻,多种柔性电池也处在不断的研究当中。

面向户外智能运动服装功能系统是一个新兴的交叉领域,它涉及了材料、生理、医学、电子、信号处理、半导体、信号处理等多个领域,为我们研究和开发带来了难度。但是智能服装的美好前景以及面向大众应用的强大实力是不容忽视的,我们必须研究它的关键理论和关键技术,待其涉及的各个学科更为成熟后,我们将有可能开发功能强大的、可用于日常生活的智能服装。



## 参 考 文 献

- [1] L. Lauter, “Personal health care in Philips: Status and ambition” in Proceedings of 25th Annual International Conference on IEEE-EMBS[C].Cancun, Mexico, September. 17-21, 2003, pp.
- [2] H.V.S.Murthy,智能纺织品概述[J].国外纺织技术,2003,12:1-5.
- [3] 彭承琳,生物学传感器--原理与应用[M].重庆:重庆大学出版社,1996.
- [4] Proda M, Reina-Tosina J,Roa L.,Distributed intelligent architecture for falling detection and physically analysis in the elderly[C].Proc.2<sup>nd</sup> EMBS/BMES,Conf.;2003,9:191-197.
- [5] Klemm M,Locher I,Troster G.A Novel Circularly Polarized Textile Antenna for Wearable Applications[C].European Microwave Conference, Amsterdam, October 2004.
- [6] Lisettia C, Nasoza F, LeRougeb C, OzyeraO, Alvarezc K. Developing multimodal intelligent affective interfaces for tele-home healthcare[J].J Hum Compute Stud,2003,245-255.
- [7] 田文果,李立宏,向继东.传感器网络与自组织网络应用潜力分析[J].中兴通讯技术.2005,4(11):57-61.
- [8] 李建中,李金宝,石胜飞,传感器网络及其数据管理的概念,问题与进展[J].软件学报,2003,14(010):1717-1727.
- [9] R.L.Ashok,D.P.Agrawal.Next-generation wearable network[J].Computer,2003,36(11): 31-39.
- [10] Francine Gemperle, Chris Kasabach, John Stivoric,Malconlm Bauer,Richard Martin, Design for Wearability.[EB/OL]<http://www.ices.cmu.edu/design/wearability/>
- [11] Akay M,Tamura T,Higashi Y, FujimotoT. Unconstrained Monitoring of Body Motion During Walking[J].IEEE Eng Med Bio May,2003,22(3):104-109.
- [12] Stein J,Wearable Sensor Technology for Functional Assessment After Stroke[J]. IEEE Eng Med Bio May 2003,22(3):26-27.
- [13] Healey JA, Wearable and Automotive Systems for Affect Recognition from Physiology[J].[PhD Thesis]MIT,2000.

## 致 谢

本毕业设计论文是在我的指导教师王春茹老师的亲切关怀和悉心指导下完成的。她严肃的科学态度，严谨的治学精神，精益求精的工作作风，细心认真的教学态度，深深地感染和激励着我。从题目的选择到最终完成，王老师都始终给予我细心的指导和不懈的支持。衷心感谢王春茹老师的用心教导。

另外，非常感谢学校给予毕业设计论文所需要的智能传感器、无线模块、主控板等硬件设备的支持，使得我可以顺利完成本毕业设计论文，这对我是极大的鼓励和支持。还要感谢我舍友在这段忙碌的时间内给予我包容和帮助。

我要感谢李嘉铭、谭国颖、许昊铭、林灿坚等同学给予我毕业设计系统上开发的悉心指导，他们对智能服装系统开发的新想法和新思路给我毕业设计论文带来很多信息，让我毕业设计论文更加充实完善。最后，感谢我父母给予毕业设计论文支持与鼓励，在这段时间，给予我细心的照顾。

## 附录 A 树莓派程序代码

### A.1 sql\_tools.py 文件

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import MySQLdb
HOST = 'localhost'
USER = 'will'
PASS = 'ljm'
DB_NAME = 'py_test_db'

class SQLObj(object):
    """docstring for SQLObj"""
    def __init__(self):
        self.db = MySQLdb.connect(HOST, USER, PASS, DB_NAME)
        self.cursor = self.db.cursor()
        print 'connect to database: %s' % DB_NAME

    def execute_sql(self, sql):
        try:
            self.cursor.execute(sql)
            self.db.commit()
        except Exception, err:
            print err
            self.db.rollback()

    def fetchall_sql(self, sql):
        try:
            self.cursor.execute(sql)
            return self.cursor.fetchall()
        except Exception, err:
            print str(err) + ' when fetchall_sql'
            self.db.rollback()

    def fetchone_sql(self, sql):
        results = self.fetchall_sql(sql)
        for row in results:
            yield row

    def sql_from_table(self, sql, tables):
        return str(sql) + str(tables) + ';'

    def sql_finish(self):
        self.db.close()
```

### A.2 sql.py 文件

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import sql_tools

tables = 'test'
sql_obj = sql_tools.SQLObj()
```

```

def create_insert_sql(pulse_times, log = '-'):
    insert_data_sql = 'INSERT INTO ' + \
        tables + \
        ' (pulse_times,log) VALUES (%i,
"%s")' % (pulse_times, log)
    return insert_data_sql

def insert_data(pulse_times, log):
    if type(pulse_times) is int:
        insert_sql = create_insert_sql(pulse_times, str(log))
        sql_obj.execute_sql(insert_sql)

def get_all_data(sql):
    return sql_obj.fetchall_sql(sql)

def get_one_data(sql):
    return sql_obj.fetchone_sql(sql)

def get_time_data():
    sql = sql_obj.sql_from_table('select date_time from ', tables)
    return get_all_data(sql)

def get_one_time_data():
    sql = sql_obj.sql_from_table('select date_time from ', tables)
    return get_one_data(sql)

def get_pulse_time_data():
    sql = sql_obj.sql_from_table('select pulse_times from ', tables)
    return get_all_data(sql)

def get_day_data(date_time):
    sql = "select date_format(date_time,'%Y%m%d%H%i%s'),pulse_times from " +
    str(tables) + ' where date_time > ' + str(date_time)
    print sql
    return get_all_data(sql)

def delect_all_data():
    sql = "TRUNCATE TABLE " + str(tables)
    print sql
    sql_obj.execute_sql(sql)

def get_log_data():
    sql = sql_obj.sql_from_table('select log from ', tables)
    return get_all_data(sql)

def finish():
    sql_obj.sql_finish()

```

### A.3 PulseDevice.py 文件

```

import serial
import time
import sql

startCmd = bytes([0xf0,0xc0,0xb0])
endCmd = bytes([0xf0,0xc1,0xb1])

```

```

ser = serial.Serial('/dev/ttyUSB0', 19200)
serble = serial.Serial('/dev/ttyAMA0', 9600, timeout=0.05)
print(ser.name+', '+serble.name)

def output_to_ble(ser, data):
    ser.flushInput()
    ser.write(data)
#(datetime.datetime(2016, 2, 28, 16, 36, 7), 100) -> '20160228163607,100'
def changeDatas(day_data):
    return (str(day_data)[2:16] + ',' + str(day_data)[19:]).rstrip(')')

output_to_ble(ser, startCmd)
sql.delect_all_data()
work = 1

try:
    while(1):
        time.sleep(0.1)
        dat = serial.to_bytes(ser.read(5))
        bledat = serial.to_bytes(serble.read(20))
        print(bledat)
        datLen = len(dat)

        if 'update' in bledat:
            work = 0

        if datLen == 5 and ord(dat[0]) == 240 and ord(dat[1]) == 192:
            pulse = ord(dat[2])*256 + ord(dat[3])
            pulseStr = 'pul:' + str(pulse)

            sql.insert_data(int(pulse), 'insert_pul')

            if work == 1:
                output_to_ble(serble, pulseStr)
                print pulseStr

            elif work == 0:
                date_time = bledat[6:] #20160101000000
                day_datas = sql.get_day_data('"' + date_time + '"')

                for data in day_datas:
                    serble.flushInput()
                    serble.write(changeDatas(data)) #20160227000000,100
                    print(changeDatas(data))
                    bledat = serial.to_bytes(serble.read(6))
                    if bledat == 'finish':
                        work = 1
                        break

                serble.write('finish') #update finish
                sql.delect_all_data()
                print('update finish')
                work = 1

finally:
    ser.write(endCmd)

```

## 附录 B Android 应用主要程序

### B.1 android 蓝牙发送信息方法

/dataMode = 0:data is string

//dataMode = 1:data is hex

```
private void bleSend(String data, int dataMode){
    byte[] value = new byte[20];
    value[0] = (byte) 0x00;
    if (data.length() > 0 && dataMode == 0) {
        //write string
        WriteBytes = data.getBytes();
    } else if (data.length() > 0 && dataMode == 1) {
        WriteBytes = hex2byte(data.getBytes());
    }
    mWriteCharacteristic.setValue(value[0],
        BluetoothGattCharacteristic.FORMAT_UINT8, 0);
    mWriteCharacteristic.setValue(WriteBytes);

    mBluetoothLeService.writeCharacteristic(mWriteCharacteristic);
}

public static byte[] hex2byte(byte[] b) {
    if ((b.length % 2) != 0) {
        throw new IllegalArgumentException("长度不是偶数");
    }
    byte[] b2 = new byte[b.length / 2];
    for (int n = 0; n < b.length; n += 2) {
        String item = new String(b, n, 2);
        // 两位一组, 表示一个字节,把这样表示的16进制字符串, 还原成一个进制字节
        b2[n / 2] = (byte) Integer.parseInt(item, 16);
    }
    b = null;
    return b2;
}
```

### B.2 android 蓝牙接收信息处理方法

//data为接收到数据

//1、实时脉搏数据: pul:80

//当前脉搏: 80次

```
if(data.contains("pul:")){
    int addy = Integer.parseInt(getSubstring(data, 4));
    pulseView.updateChart(addy);
}

//2、结束历史数据更新
if(data.contains("finish")){
    mUpdating = false;
    invalidateOptionsMenu();
    mgr.add(pulses);
    pulses.clear();
    DisplayListTask task = new DisplayListTask(HistoryListActivity.this);
    task.execute();
}

//3、更新每一条历史数据
if(data.contains(",")){
```

```

        String[] datas = data.split(",");
        String dateTime = changeDateTime(datas[0],0);
        System.out.println("dateTime:" + datas[1]);
        Pulse pulse = new Pulse(datas[1], dateTime);
        pulses.add(pulse);
    }

    //4、每更新完50条历史数据刷新一次显示列表
    if(pulses.size()==50){
        mgr.add(pulses);
        pulses.clear();
        DisplayListTask task = new DisplayListTask(HistoryListActivity.this);
        task.execute();
    }

```

### B.3 android 数据库操作类

```
package com.lin.dbhelper;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
```

```
public class DBManager {
    private DBHelper helper;
    private SQLiteDatabase db;

    public DBManager(Context context) {
        helper = new DBHelper(context);
        //因为getWritableDatabase内部调用了mContext.openOrCreateDatabase(mName,
        0, mFactory);
        //所以要确保context已初始化,我们可以把实例化DBManager的步骤放在
        Activity的onCreate里
        db = helper.getWritableDatabase();
    }

    /**
     * add pulses
     * @param pulses
     */
    public void add(List<Pulse> pulses) {
        db.beginTransaction();    //开始事务
        try {
            for (Pulse pulse : pulses) {
                db.execSQL("INSERT INTO pulse VALUES(null, ?, ?)", new
                Object[]{pulse.getValue(), pulse.getTime()});
            }
            db.setTransactionSuccessful();    //设置事务成功完成
        } finally {
            db.endTransaction();    //结束事务
        }
    }

    /**

```

```

    * query all pulses, return list
    * @return List<Pulse>
    */
    public List<Pulse> query() {
        ArrayList<Pulse> pulses = new ArrayList<Pulse>();
        Cursor c = queryTheCursor();
        while (c.moveToNext()) {
            Pulse pulse = new Pulse();
            pulse.set_id(c.getString(c.getColumnIndex("_id")));
            pulse.setValue(c.getString(c.getColumnIndex("value")));
            pulse.setTime(c.getString(c.getColumnIndex("time")));
            pulses.add(pulse);
        }
        c.close();
        return pulses;
    }

    public List<Pulse> query(String date1, String date2){
        ArrayList<Pulse> pulses = new ArrayList<Pulse>();
        Cursor c = db.rawQuery("SELECT * FROM pulse where time >= ? and time
< ?",new String[]{ date1, date2});
        while (c.moveToNext()) {
            Pulse pulse = new Pulse();
            pulse.set_id(c.getString(c.getColumnIndex("_id")));
            pulse.setValue(c.getString(c.getColumnIndex("value")));
            pulse.setTime(c.getString(c.getColumnIndex("time")));
            pulses.add(pulse);
        }
        c.close();
        return pulses;
    }

    /**
     * query all pulses, return cursor
     * @return Cursor
     */
    public Cursor queryTheCursor() {
        Cursor c = db.rawQuery("SELECT * FROM pulse", null);
        return c;
    }

    /**
     * close database
     */
    public void closeDB() {
        db.close();
    }
}

```