

编号_____

南京航空航天大学

毕 业 论 文

题 目 基于 4G 通信模块的无人机无线通信研究与实现

学生姓名

罗鑫辉

学 号

161310420

学 院

计算机科学与技术学院

专 业

计算机科学与技术

班 级

1613105

指导教师

翟象平 讲师

二〇一七年六月

南京航空航天大学

本科毕业设计（论文）诚信承诺书

本人郑重声明：所呈交的毕业设计（论文）（题目：_____）是本人在导师的指导下独立进行研究所取得的成果。尽本人所知，除了毕业设计（论文）中特别加以标注引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写的成果作品。

作者签名：

年 月 日

（学号）：

基于 4G 通信模块的无人机无线通信研究与实现

摘 要

随着科学技术的蓬勃发展，人们开始畅想以“物联网”为核心的智能化未来生活，而这样一个“物物相连”的网络的构建需要可靠的通信方式作支持。本课题选取无人机作为具体的应用对象，它是物联网技术重要的应用领域之一，其对通信方式的选择和使用更是有着特殊的要求。本文首先对包括 4G 无线通信技术在内的热门的无线通信技术的特点进行简述和比较，并结合无人机这一特殊应用背景对这些通信技术的应用可行性进行分析。其次重点阐述无人机通过安装了 4G 模块的树莓派接入 4G 网络，并进一步通过采用 B/S 架构开发的应用程序实现用户与无人机之间的控制通信的设计和过程。

关键词：无线通信，4G，通信模块，树莓派，无人机

Research and Implementation of Wireless Communication of Drones Based on 4G Communication Module

Abstract

With the flourish development of science and technology, people begin to imagine an intelligent life with the core of “Internet of Things”, and the construction of such a “material connected” network needs reliable communication mode for support. This project chooses drones as a specific application object, which is one of the important application fields of Internet of Things technology, and it has special requirements for the choice and use of communication mode. In this paper, the characteristics of popular wireless communication technologies, including the 4G wireless communication technology, are briefly introduced and compared, and the application feasibility of these communication technologies is analyzed in combination with the special application background of drones. Second, focus on elaborating the design and process of drones connect to 4G network through a Raspberry Pi with the installation of 4G module, and further achieve the control of the communication between users and drones through an application developed in B/S architecture.

Key Words: Wireless Communication; 4G; Communication Module; Raspberry Pi; Drones

目录

摘 要 i

Abstract ii

第一章 引 言 1

 1.1 课题研究背景和意义 1

 1.2 无线通信技术 1

 1.2.1 热门无线通信技术 1

 1.2.2 4G 通信的关键技术及特点 3

第二章 无人机通信系统的总体设计 5

 2.1 需求分析 5

 2.2 系统结构 5

 2.2.1 硬件结构 5

 2.2.2 软件结构 6

 2.3 通信流程设计 7

第三章 4G 模块..... 9

 3.1 驱动安装 9

 3.1.1 编译环境和工具 9

 3.1.2 文件修改 9

 3.1.3 编译方法 11

 3.1.4 可能存在的问题和解决方法 13

 3.2 模块使用 13

 3.2.1 拨号方法 13

3.2.2 断开连接	17
3.2.3 关机	18
3.2.4 其它重要的 AT 指令及常见错误信息	18
第四章 软件系统设计与实现	20
4.1 系统总体设计	20
4.1.1 功能划分	20
4.1.2 技术应用	20
4.1.3 开发环境和工具	20
4.2 前端界面	21
4.2.1 界面设计	21
4.2.2 界面跳转逻辑	22
4.3 服务器	23
4.3.1 事物逻辑处理	23
4.3.2 Socket 通信程序	23
4.4 数据库	24
4.4.1 表结构及数据类型	24
4.4.2 数据库操作	26
4.5 树莓派	27
第五章 系统测试及分析	29
5.1 测试情况和结果分析	29
5.2 存在的问题和解决方法或建议	31
第六章 总结和展望	33
参考文献	34
致谢	35

第一章 引言

1.1 课题研究背景和意义

随着科学技术的发展，人们开始展望物联网时代的美好生活；作为物联网技术的重要应用领域之一，无人机的使用不再局限于少数特殊行业和场景，大量的民用无人机进入市场，并且在人们的生活和工作中发挥着越来越重要的作用，例如影视航拍、农药喷洒、快递运输和公共侦查等^[1]。而在无人机的应用过程中，可靠的通信技术的支持显得尤为重要，它不仅能为无人机的使用提供基本而必要的安全保障，更重要的是能够为无人机满足更多的应用需求、拓展应用空间以及形成良好的使用体验提供了可能，例如常见的图片、视频传输等。而近年来 4G 通信技术的日渐成熟和广泛的应用使其成为了一种值得关注的可能满足无人机应用需求的选择。本文将根据 4G 通信技术的特点以及相较于其它无线通信技术的优势分析其在无人机通信应用的可行性；并进一步提供基于 4G 通信模块的无人机无线通信的设计和应用方法。

1.2 无线通信技术

当前应用较为广泛或热门的无线通信技术主要有 Wi-Fi、Bluetooth、ZigBee、可见光通信、4G 等。这些技术都有各自的特点，当中的许多技术已经在物联网的某些应用领域发挥了重要的作用，如智能家居^{[2][3]}；因此，我们也希望从这些技术中能够找到适合应用于无人机通信的技术。本章接下来的篇幅将会从上述无线通信技术的技术特点出发，以无人机通信为应用对象作分析和比较；并且通过 4G 通信技术中所运用的关键技术，着重探讨 4G 通信技术在无人机无线通信应用的可能性和优势。

1.2.1 热门无线通信技术

Wi-Fi，是当前人们在日常生活和工作中使用最广泛的无线通信技术之一，其最主要的优势在于在实现高速的上网体验（其中 IEEE802.11b 最高速度为 11Mbps，IEEE802.11a 与 IEEE802.11g 的最高速度为 54Mbps，而 IEEE802.11n 的理论速度更是能达到 600Mbps）的同时，却因为其使用的是免许可频段、在频率资源上不存在限制，而只需要极低的使用成本。此外，Wi-Fi 网络组网便捷、设备成本较低、技术本身的健康安全性好（发射功率低）等也是其重要的应用优势^[4]。然而，Wi-Fi 技术本身也存在一些不足，主要在于频率干扰使得其传输

速率显著降低、传输质量不稳定，而且相似频段之间存在的无线电波将产生相互影响；同时 Wi-Fi 信号在遇到障碍物时的穿透性能较差^[5]。此外，相关的实验也表明在现有 802.11n 的标准下，无法较好地满足无人机高速移动下的通信需求，需要进一步地修订和改善^[6]。由此可见，虽然 Wi-Fi 的通信速度和信号覆盖范围等技术条件能够基本满足普通情况下的无人机的应用需求，但是存在的不足也将对其在无人机通信的应用中形成一定的制约，对用户的使用体验可能造成影响。

Bluetooth，中文名称蓝牙，作为一种在 20 世纪末创制的无线通信技术，它的出现为人们摆脱线缆限制、实现简便的无线传输提供了可能，如今绝大多数的电子设备都支持使用 Bluetooth 进行数据传输。而且，Bluetooth 在历经 20 年左右的今天依然是一种热门的无线技术与其具备许多应用优点是分不开的。它工作在全球通用的 2.4GHz 频段，同样不必经过申请即可使用，而且采用跳频技术来有效地避免干扰。它采用 FM 调制方式，设备的复杂性低。它组网方便，支持包括点对点、点对多点的方式，而且不分主从设备。此外，它还具备安全灵活、功耗低、辐射小等特点。即便如此，Bluetooth 仍有其不适合应用于无人机通信的缺陷，主要在于它是一种短距离无线传输技术，有效的通讯距离仅约为 10~100m；而且它的传输速率为 1Mb/s，与 Wi-Fi 等无线通信技术相比有巨大的差距，无法很好地满足在无人机应用过程中常见的图像、视频数据的传输需求^{[7][8]}。

近年来，ZigBee 在物联网领域中发挥了重大的作用，成为了热门的无线通信技术，广泛应用于自动控制和远程控制，如智能家居中的温度、照明、电器及各种计量表的控制等。它具备许多技术优点，例如数据传输可靠、网络容量大、安全性高等。而它在相应领域内得到广泛应用的更重要的原因在于它的成本低、功耗低、组网方式灵活多样，能够很好地支持小型模块进行大规模布局。但是，这也决定了 ZigBee 是一种近距离的无线通信技术，一般情况下有效通信范围仅有 10~75m^[9]；而且数据传输速率低（只有 10~250Kb/s），只适合于完成简单数据的传输，如温度、湿度、压力等数据的采集传输^[10]，因此无法应用于无人机通信。

白光 LED 所具备的节能环保、使用寿命长、尺寸小等优点，使其在照明方面逐渐取代白炽灯、日光灯等而获得大量的应用，这也为以白光 LED 为基础的可见光通信提供了巨大的发展优势。对于当前应用十分广泛的射频通信，可见光通信因其所使用的是可见光波段，不会对现有的以射频为基础的其它系统产生干扰而显得十分友好，能够在对电磁波敏感的应用场景进行布局；而且不需要使用昂贵的射频单元也使得应用成本大大降低。此外，作为日常照

明使用的光源，可见光通信不会对人体健康产生危害。在保证了一定的传输速率的同时，可见光通信因其本身采用“光”作为传播载体的特性，使其具备易于在遮蔽空间内创造防“窃听”通信环境的优点，因此在提升安全性能方面具有特殊的优势。但是，这也成为了制约其在无人机通信中应用的重要因素，在使用过程中存在着因为遮蔽而导致设备“失联”的风险^[11]。同时，可见光通信目前仍存在技术层面上的制约，例如 LED 固有的非线性、带宽的限制、信号调制、收发器的功率输送和多重访问等发面的问题；而在非技术层面，可见光通信的推广也将受到现有成熟的射频通信的影响，仍需时日才能得到良好的布局应用^[12]。因此，可见光通信目前仍处于研究和实验阶段，即便已经获得可喜的研究成果，但还没有得到大规模成熟的应用，故现阶段也无法作为无人机通信的可选项。

综上所述，虽然这些热门的无线通信技术有的已经得到长时间的成熟应用，有的是技术本身具有特殊的前瞻性和广阔的应用前景，但都或多或少地存在着各自的技术限制而在现阶段无法很好地契合无人机通信的应用需求。

1.2.2 4G 通信的关键技术及特点

4G，即 the Fourth-Generation Mobile Communication Technology，指的是第四代移动通信技术，是当前得到广泛商用布局的无线通信技术。从技术层面看，4G 通信技术中运用的几大关键技术能够为实现可靠的通信和良好的用户体验提供保障。

首先是具备诸多优点的 OFDM 技术（正交频分复用），其基本思想是将高速信息流转换成多路低速率的并行数据流后在相互正交的子信道上传输。OFDM 的运用，使得 4G 通信系统的频谱利用率得到大大提高，并且具有很强的抗干扰和抗衰弱能力，适合进行高速数据传输^[13]。其次是智能天线技术，即通过算法智能地调节天线阵列的波束，使主瓣对准用户波达方向，旁瓣对准干扰信号的来源方向，实现充分获取和利用移动用户的目标信号的同时，抑制和消除干扰信号。除此之外，它还是一种空分多址技术，因此能够提高 4G 通信系统的频谱利用率、提升系统容量^[14]。再者是 MIMO（多输入输出）技术，它通过增加收发两端的天线数量来实现在不增加带宽的情况下，成倍地提高通信系统的容量和频谱利用率。对随机衰落和可能存在的多径传播形成有效的利用，使得信道的可靠性得到提高，降低了误码率^[15]。还有软件无线电技术，其基本思想在于，构造一个通用的硬件平台，然后用软件来实现具体的功能，并且将宽带 A/D 和 D/A 转换器尽可能靠近天线，使得无线通信系统具有高度灵活性和开放性。这种技术思想的运用能够为 4G 技术适应不同种类的产品需求打下基础，便于相

关产品的开发，甚至为其它 4G 关键技术的设计应用提供帮助^[16]。而在通信过程中所出现多址干扰以及随之产生的远近效应，则使用多用户检测技术来应对，通过实现从干扰信息中调制出特定用户的信号以达到提高系统的性能、增大系统的容量的目的^[17]。最后则是 IPv6 技术的应用，它不仅解决了 IPv4 现有的 IP 地址即将枯竭的问题，还具备能够实现端到端的对等通信、拥有较完善的安全机制、能够实现地址的自动配置、服务质量高、移动性好等优势^[18]。

在实际情况中，得益于运营商对 4G 通信系统的积极布局与建设，我们看到 4G 通信技术已经得到了广泛的应用；并且在上述关键技术的运用下，4G 通信网络相较于其它无线通信方式具备诸多优势。4G 网络在低速移动时的数据传输速率能够高达 100Mbps，达到目前军用数据链传输速率的要求；即便在高速移动的情况下也能满足对较高分辨率图像的传输需求^[19]。日益完善的网络建设使得信号覆盖面积大大增加，而且对于遮蔽物的穿透性相对较好；作为一种远距离通信技术，能够满足无人机高速、大范围移动的通信需求。因此，从技术和实际应用情况这两种方面看，4G 通信技术适合应用于无人机通信。

第二章 无人机通信系统的总体设计

本章将从总体的角度依次介绍本课题所实现的基于 4G 通信模块的无人机无线通信系统的应用需求、系统结构，以及通信流程的设计。

2.1 需求分析

本系统的设计目的在于实现基于 4G 通信模块的无人机无线通信的过程，进一步从实际的角度验证 4G 通信技术应用于无人机通信的可行性。故本系统在开发过程中将重点关注通信过程的设计和实现；而在功能方面暂不包含所有常见的无人机应用需求，仅为满足验证通信过程而选取部分基础功能进行设计和实现，包括对无人机的操作控制和信息采集。其中操作控制部分具体实现对无人机的“连接”和“断开连接”，以及无人机的“起飞”和“降落”这两个基本飞行动作。而信息采集部分将具体实现对无人机位置信息的采集，包括经度、纬度和高度。

本系统最终实现用户使用个人终端（手机、平板电脑、PC 等）通过浏览器访问系统，并在正确登录和连接后，在操作界面对无人机进行上述操作控制和信息采集功能的应用。同时实现对使用过程中产生的操作记录、指令、无人机的位置信息等进行存储，方便进一步管理和应用。

2.2 系统结构

2.2.1 硬件结构

硬件系统的构成包括 4G 模块（中兴 ME3760 V2/C）、树莓派（Raspberry Pi 3B）、无人机（大疆经纬 M100）、用户终端、服务器。本小节将重点介绍其中几个主要的硬件设备，然后在此基础上介绍本系统的硬件结构和各硬件之间的应用关系。

与现有 4G 网络的连接是通过 4G 模块实现的。根据查阅的相关资料和成功的应用案例，本课题选用的是中兴 ME3760 V2/C，这是一款 Mini PCI-E 接口的 LTE 模块，支持 LTE TDD band38(2.68GHz)、band39(1.9GHz)、band40(2.3GHz)、band41(2.5GHz)；向下兼容 TD-SCDMA A 频段(2.1GHz)和 F 频段(1.9GHz)以及 GSM900MHz、GSM1800MHz；支持最大 50M/150Mbps 的理论上下行数据传输速率。且该模块支持 SIM 卡业务（主要支持中国移动 4G 通信业务），故只需通过常用的 SIM 卡即可实现与现有 4G 网络进行连接，减少了课题应用的复杂度。

由于无人机本身无法提供良好的环境进行通信应用的开发，因此在实际开发过程中将使用一块树莓派作为中间平台。树莓派是只有信用卡大小的微型电脑（易于部署在无人机上），其系统基于 Linux，具有电脑的所有基本功能。它是一款基于 ARM 的微型电脑主板，以 SD/Micro SD 卡为存储介质，卡片主板周围有 4 个 USB 接口和一个 10/100 以太网接口，同时拥有 GPIO 接口、视频模拟信号的电视输出接口和 HDMI 高清输出接口等。因此可以方便地在树莓派上进行应用的开发，然后将具体的操作指令等传输至无人机，大大提升了无人机应用的拓展性。

本课题所选用的大疆经纬 M100 无人机具有较好的平台开放性，预留了 CAN 及 UART 双通信数据传输接口，便于接入扩展设备（树莓派）；同时提供了 SDK 满足开发通信应用的需求。

在原有以 PCI-E 为接口的 4G 模块上，为其配置包含 USB 公口（A 型插头）的底座（同时包含 SIM 卡插槽），即可通过 USB 接口实现 4G 模块和树莓派的连接。而将树莓派的 GPIO 接口与无人机预留的 UART 接口进行连接，即可实现树莓派与无人机进行数据传输；同时实现无人机通过配置 4G 模块的树莓派获得使用 4G 通信网络的能力，进一步通过 4G 网络与服务器进行数据交互。而另一端的用户终端则是通过 Web 服务与系统进行连接交互，最终实现从用户终端到无人机的通信链路。具体情况如图所示：

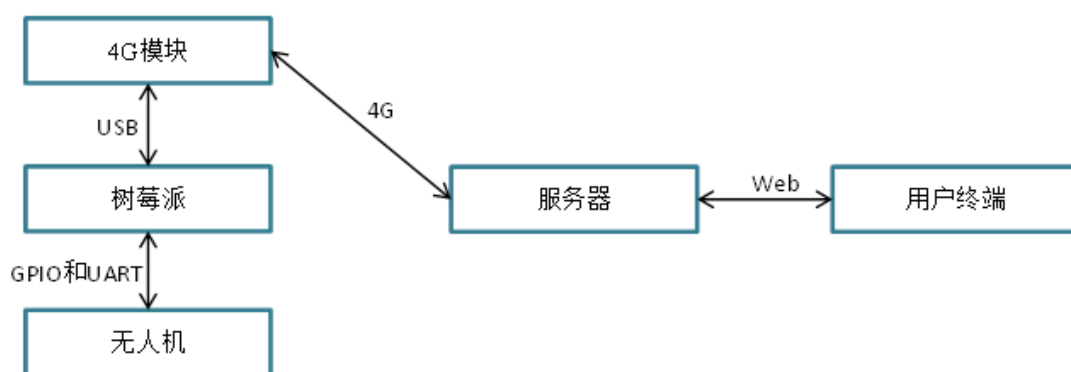


图 2.1 硬件结构关系

2.2.2 软件结构

软件系统的构成包括通信处理程序（服务器端）和无人机通信控制程序（树莓派端）。通信处理程序对用户提供了操作无人机及获取无人机信息的前端界面；同时对来自用户及无人机

的请求进行处理和响应，在处理过程中对维护的数据库进行相应的操作。无人机通信控制程序主要实现无人机与服务器之间的通信，包括向服务器发送查询最新指令的请求以及发送从无人机获取的位置信息，同时将服务器反馈的用户指令信息进行识别并调用相关方法将指令下发至无人机。如图所示：

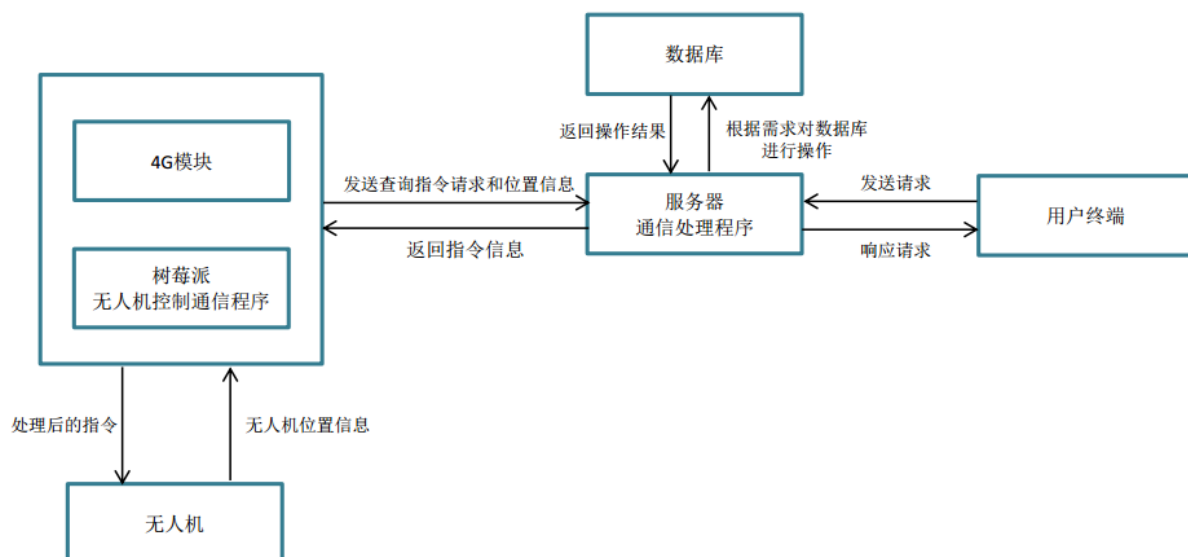


图 2.2 软件结构关系

2.3 通信流程设计

根据软件结构中所包含的两部分程序，通信流程也相对应地包括用户通信流程和无人机通信流程这两个部分。

首先是用户通信流程部分，用户使用终端设备的浏览器登录系统，然后连接至可用的无人机设备，进入无人机控制界面。用户在控制界面上点击按钮发出操作指令，指令信息以请求的方式发送至服务器，服务器接收请求后解析出对应的指令，并将指令添加至数据库，同时增加和修改数据库中存储的相关设备状态和飞行记录信息，然后返回响应结果，更新用户控制界面的信息。除了用户点击操作按钮发出的请求外，进入控制界面后，页面同时会以固定的频率对服务器发起“查询无人机位置信息”的异步请求，服务器响应后对数据库进行查询，然后反馈最新的位置信息，对用户控制界面的信息进行更新。此外，用户在连接至可用设备时会向无人机发出“连接”指令；而点击“断开连接”的按钮后，将会在一定时间内跳转至连接界面，可以连接其它可用设备。

其次是无人机通信流程部分，树莓派开机后，连接 4G 网络，然后启动应用程序，发起

对服务器的连接请求。成功与服务器建立连接后，则循环对服务器发起查询指令的请求，同时在请求信息中包含当前无人机的位置信息。服务器接收到无人机发送的数据后，从数据中识别出位置信息并添加至数据库；同时对数据库执行查询操作，将查询到的最新的指令信息发送给树莓派，树莓派再将指令处理后通过串口通信传递至无人机。整个通信过程持续至用户发出“断开连接”的指令为止。

第三章 4G 模块

本章将首先简述 Linux 环境下的驱动安装方法，进一步以此为基础阐述本课题所使用的 4G 模块的驱动安装过程。然后介绍 4G 模块的使用方法，包括如何使用 AT 指令进行拨号上网；同时总结部分重要的 AT 指令以便查询模块和网络信息，帮助对模块的了解和使用。

3.1 驱动安装

Linux 环境下的驱动安装主要有“静态加载”和“动态加载”这两种方式。静态加载即将驱动程序直接编译进内核，系统启动后可以直接调用；缺点是调试起来比较麻烦，每修改一个地方都要重新编译内核。动态加载则是利用了 Linux 的 module 特性，可以在系统启动后通过 insmod 命令添加模块，在不需要的时候使用 rmmod 命令卸载模块；采用动态加载的方式便于驱动调试，缺点是每次启动都需要使用命令加载所需的模块。因此，在本课题的应用过程中，先使用动态加载的方法进行调试，成功后再编译进内核。又由于树莓派硬件性能有限，故使用交叉编译的办法，利用 PC 的性能加快内核的编译速度。接下来将首先介绍驱动的编译环境和所需工具，然后介绍安装驱动和编译内核的方法，最后阐述编译过程中发现的问题并给出相应的解决办法。

3.1.1 编译环境和工具

交叉编译环境：Ubuntu 16.04 LTS 64 位

树莓派内核文件版本：linux-rpi-4.4.y.zip（根据实际需要选择，例如需要与原有系统的版本进行匹配以便让其它已有模块也能正常工作）

编译工具链：tools-master.zip

需要修改的内核文件：option.c, usb-serial.c, cdc_ether.c, Makefile, cdc_encap.c, cdc_encap.h

3.1.2 文件修改

此 4G 模块支持 ECM 及 RNDIS，可通过使用 AT 命令+ZNCARD/+ZNETCARDTYPE 命令进行切换；ECM 及 RNDIS 可以看成是在 USB 接口上实现网络接口卡（NIC, Network Interface Card）功能的不同协议规范。本文将对 RNDIS 方式下的驱动安装和加载方法作介绍。

将下载的内核源文件进行解压，然后进入相应的目录（具体路径根据实际情况作修改，本文默认内核文件解压至根目录下，即内核源文件所在的路径为/linux-4.4.y）对上述文件作

以下修改：

（1）增加具体设备驱动

文件：option.c

路径：/linux-4.4.y/drivers/usb/serial/option.c

修改内容：在 static struct usb_device_id option_ids[] = {} 中添加：

```
{USB_DEVICE(0x19d2,0x0199)},
```

（2）USB 串口驱动过滤 NDIS 接口

说明：由于 USB 串口跟 NDIS 都属于非标准 cdc 设备，需要防止 NDIS 口被 USB 串口驱动加载导致 NDIS 驱动无法顺利完成加载

文件：usb-serial.c

路径：/linux-4.4.y/drivers/usb/serial/usb-serial.c

修改内容：在函数 usb_serial_probe 中找到：

```
type = serach_serial_device(interface);
```

在其前面添加：

```
if((le16_to_cpu(dev->descriptor.idVendor) == 0x19d2)&&
    (le16_to_cpu(dev->descriptor.idProduct) == 0x0199)){
    if(1 == interface->cur_altsetting->desc.bInterfaceNumber){
        return -ENODEV;
    }
}
```

（3）添加编译选项

文件：Makefile

路径：/linux-4.4.y/drivers/net/usb/Makefile

修改内容：在 obj-\$(CONFIG_USB_USBNET) += usbnet.o 的下一行添加：

```
obj-m += cdc_cncap.o
```

（4）将文件 cdc_encap.c 和 cdc_encap.h 复制到/linux-4.4.y/drivers/net/usb 目录下（内核源码中不存在这两个文件，可直接复制）

（5）将文件 cdc_ether.c 复制到/linux-4.4.y/drivers/net/usb 目录下（内核源码中存在此文件，又 cdc_ether.c 文件改动较多，如原系统中对应的文件未被改动，则可使用提供的资源直

接替换，否则需要进行比对修改）

3.1.3 编译方法

完成所需文件的修改和添加后，进入内核目录，执行 `make menuconfig`，依次进入 `device drivers`→`usb support`→`usb serial converter support`，将组件 `USB driver for GSM and CDMA modems` 作为模块进行编译。然后依次进入 `device drivers`→`Network device support`→`usb Network Adapters` 将组件 `Mutil-purpose USB Networking Framework` 作为模块进行编译。由于修改了上述编译选项，故接下来需要编译内核。

（1）解压并添加工具链目录至当前用户目录下的 `.bathrc` 文件中的 `PATH` 路径。假设工具链目录为 `/tools`，则：

32 位环境：`/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian/bin`

64 位环境：`/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi-hf-raspbian-x64/bin`

（2）加载默认配置脚本。进入内核源码目录 `/linux-4.4.y`，执行命令：

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- bcm2709_defconfig
```

（3）编译 Linux 内核。执行命令：

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- zImage modules dtbs
```

（4）安装到 SD 卡上。

插入 SD 卡后，使用 `lsblk` 命令查看到两个分区，其中 `sdb1` 为 `FAT (boot)` 分区，`sdb2` 为 `ext4` 文件系统 (`root`) 分区。如果是安装了 `NOOBS` 系统的 SD 卡，则会看到 `sdb1`，`sdb2`，`sdb5`，`sdb6`，`sdb7`，其中 `sdb6` 和 `sdb7` 分别为 `FAT` 和 `ext4` 分区。

将两个分区挂载到当前环境中（以 `sdb1` 和 `sdb2` 为例，若为 `NOOBS` 则修改相应的分区数字即可），在内核文件根目录 `/linux-4.4.y` 下执行以下命令：

```
mkdir mnt/fat32
mkdir mnt/ext4
sudo mount /dev/sdb1 mnt/fat32
sudo mount /dev/sdb2 mnt/ext4
```

然后安装模块到 SD 卡上。执行命令：

```
sudo make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-hf- INSTALL_MOD_PATH=
mnt/ext4 modules_install
```

最后使用 `mkknlimg` 命令制作内核镜像文件并拷贝至 SD 卡，同时根据需要进行命名，例

如 kernel-myconifg.img。执行命令：

```
sudo cp arch/arm/boot/zImage mnt/fat32/kernel-myconfig.img
sudo cp arch/arm/boot/dts/*.dtb mnt/fat32/
sudo cp arch/arm/boot/dts/overlays/*.dtb* mnt/fat32/overlays/
sudo cp arch/arm/boot/dts/overlays/README mnt/fat32/overlays/
sudo umount mnt/fat32
sudo umount mnt/ext4
```

安装完成后修改 boot 分区中的 config.txt 文件，增加一行：

```
kernel=kernel-myconfig.img
```

即可在不影响原有系统内核文件的基础上，使用新的内核。

安装完成后，重新启动树莓派，插入 4G 模块，使用 “dmesg” 命令能够看到 ttyUSB0，ttyUSB1，ttyUSB2 相关加载信息；且使用 “ls /dev/ttyU*” 命令显示多了三个 USB 串口 /dev/ttyUSB0，/dev/ttyUSB1，/dev/ttyUSB2；同时使用 ifconfig -a 命令后显示多出了一个新的网络连接，则说明模块驱动安装成功。具体情况如下图所示：

```
[ 7.970258] usb 1-1.3: Product: ZTE-Modem
[ 7.970264] usb 1-1.3: Manufacturer: ZTE
[ 7.970270] usb 1-1.3: SerialNumber: ZTE-Modem
[ 8.018754] cdc_ether 1-1.3:1.1: ZTE Ethernet Device: claiming interface 1
[ 8.018803] usb 1-1.3: bad CDC descriptors
[ 8.020064] cdc_ether 1-1.3:1.1 eth1: register 'cdc_ether' at usb-3f980000.us
b-1.3, ZTE Ethernet Device, 02:a8:a2:c2:57:35
[ 8.020349] usbcore: registered new interface driver cdc_ether
[ 8.049817] usbcore: registered new interface driver usbserial
[ 8.049880] usbcore: registered new interface driver usbserial_generic
[ 8.050171] usbserial: USB Serial support registered for generic
[ 8.050705] usbcore: registered new interface driver cdc_wdm
[ 8.065992] usbcore: registered new interface driver qmi_wwan
[ 8.084090] usbcore: registered new interface driver option
[ 8.084159] usbserial: USB Serial support registered for GSM modem (1-port)
[ 8.084369] option 1-1.3:1.0: GSM modem (1-port) converter detected
[ 8.084653] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[ 8.084724] option 1-1.3:1.2: GSM modem (1-port) converter detected
[ 8.084941] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[ 8.084992] option 1-1.3:1.3: GSM modem (1-port) converter detected
[ 8.085192] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2
```

图 3.1 显示开机信息

```
pi@raspberrypi:~ $ ls /dev/ttyU*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2
```

图 3.2 查看设备目录

```
eth1    Link encap:Ethernet  HWaddr 02:a8:a2:c2:57:35
        inet addr:10.108.2.225  Bcast:10.255.255.255  Mask:255.0.0.0
        inet6 addr: fe80::c602:e78c:6273:276d/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:3966 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3781 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:3637968 (3.4 MiB)  TX bytes:434068 (423.8 KiB)
```

图 3.3 显示网络配置

3.1.4 可能存在的问题和解决方法

(1) 出现 "implicit declaration of function 'kmalloc'" and "implicit declaration of function 'kfree'" 的错误。

解决方法：在 cdc_encap.c 中添加头文件：#include <linux/slab.h>。

(2) 函数 ioctl() 在 linux2.6.38 版本以后被 unlocked_ioctl() 和 compat_ioctl() 取代。

解决方法：将 cdc_encap.c 文件的 static struct file_operations cdc_encap_fops = {} 中的 .ioctl = cdc_encap_ioctl 替换为 .unlocked_ioctl = cdc_encap_ioctl。

(3) 函数 init_MUTEX() 在 linux2.6.38 版本以后被 sema_init() 取代。

解决方法：将 cdc_encap.c 文件的 init_MUTEX(&encap->sem) 替换为 sema_init(&encap->sem, 1)。

(4) 在编译过程中 cdc_ether.c 出现大量错误需要更改，可直接进行替换。

3.2 模块使用

3.2.1 拨号方法

硬件成功识别后，可以通过串口调试工具 minicom 使用 AT 命令进行拨号上网。AT 命令的标准定义是：从 TE（终端设备）或 DTE（数据终端设备）向 TA（终端适配器）或 DCE（数据电路终端设备）发送的特定格式的字符串。TE 通过 TA 发送 AT 命令来控制 MS（移动台）的功能，与网络业务进行交互。用户可以通过 AT 命令进行呼叫、短消息、电话本、数据业务、补充业务、传真等方面的控制。

使用 AT 命令时应注意：

(1) AT 命令的发送应该是串行的，一条命令没有返回，或者等待命令超时之前，主机不能向设备发送任何 AT 命令，否则可能会造成前一条命令中断处理或忽略第二条命令，上报 CME ERROR:6000（通道堵塞）。

(2) 一般的 AT 命令超时时常为 30 秒，在发送超时后，可以使用 AT 命令中的 "AT"

来判断设备是否正常。

第一次使用需要对 minicom 配置相应的参数信息：在 root 权限下输入 minicom -s 命令进入 minicom 配置界面，选择 Serial port setup 选项进行配置，具体配置如图所示：

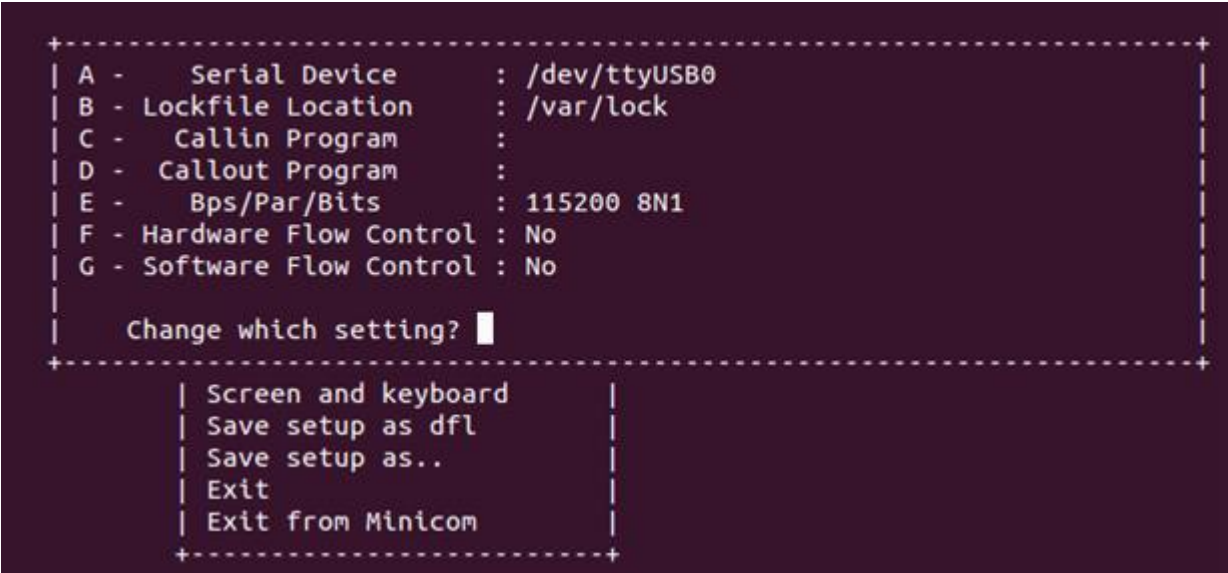


图 3.4 minicom 配置

配置完成后选择 Save setup as dfl 保存为默认设置。重新输入 minicom 进入串口通信界面，在键盘上按下 Ctrl+A 后再按 E（打开本地字符回显）即可开始输入 AT 指令进行拨号，根据模块 AT 指令手册总结出的具体使用的指令、返回的信息以及指令和返回信息各自对应的含义如下（根据拨号过程中指令的使用顺序列出，其中各表格第一列中以下划线标注的为实际输入指令，其余为返回的信息）：

（1）AT

表 3.1 测试 Module 与串口的联通状态

指令或信息	含义
<u>AT</u>	检测 Module 与串口是否联通，能否接收 AT 命令。
OK	

（2）IMSI 获取

表 3.2 IMSI 获取

指令或信息	含义
<u>AT+CIMI</u>	用于查询 USIM 或者 SIM 卡的 IMSI 值。
+ZICCID:	ICCID 上报。卡信息主动上报。

表 3.2 续

指令或信息	含义
+CIMI:	返回存储卡中的 IMSI 值，由 0~9 的十进制数构成的字符串。
OK	
+ZPBIC:1,0	卡初始化完成。在卡初始化完成时，会主动上报此命令，提示卡初始化是否成功。其中第一个参数值为“1”表示卡初始化成功。第二个参数表示业务类型：“0”为短消息，“1”为电话本，“2”为全部。
+ZPBIC:1,1	同上。

（3）终端报错格式设置

表 3.3 终端报错格式设置

指令或信息	含义
<u>AT+CMEE=1</u>	取值为“1”表示使用+CME ERROR:<err>result code 的格式报错，其中<err>采用错误编号值。
OK	

（4）字符回显设置

表 3.4 字符回显设置

指令或信息	含义
<u>ATE0</u>	用于设置 MS 是否会送从 TE 接收的字符。本版本仅支持 ATE0(不回送)的格式。
OK	

（5）PDP 设置环境

表 3.5 PDP 设置环境

指令或信息	含义
<u>AT+CGDCONT=1,"IP"</u>	第一个参数为<cid>，取值为“1~4”，为 PDP 设置环境的索引值，每一个值包含一组与 PDP 相关的参数。第二个参数为<PDP_type>：表示包交换协议类型；“IP”，意即 IPv4 协议。
OK	

(6) GPRS 自动附着设置

表 3.6 GPRS 自动附着设置

指令或信息	含义
<u>AT+ZGAAT=0</u>	对 PS 服务进行设置永久在线或用时在线。参数为<state>，此处取值为“0”即表示“自动附着”。
OK	

(7) 操作模式设置

表 3.7 操作模式设置

指令或信息	含义
<u>AT+CFUN=1</u>	用于设置 MS 的模式或重启 MS。参数为<fun>，此处取值为“1”表示“完全功能”。
OK	
+CREG:2	网络注册，参数<stat>的值为“2”，表示没有注册，但是 ME 正在寻找一个新的网络进行注册。
+CGREG: 2	GPRS 网络注册，参数<stat>的值为“2”，表示没有注册，但是 ME 正在搜寻要注册的新的运营商。
+CEREG: 2	EPS 注册状态，参数<stat>的值为“2”，表示没有注册，但是当前 MT 正在搜寻一个新的运营商进行注册。

此指令对应返回信息的后三项（+CREG、+CGREG、+CEREG）的值若为“1”和“5”则表示注册，其中“1”表示本地网络，“5”表示漫游。若值为“3”，表示注册被拒绝；“4”表示未知。

(8) EPS 网络注册状态

表 3.8 EPS 网络注册状态

指令或信息	含义
<u>AT+CEREG=1</u>	参数<n>的值为“1”，表示启用网络注册的主动上报结果码。
OK	

(9) PDP 上下文激活/去激活

表 3.9 PDP 上下文激活/去激活

指令或信息	含义
<u>AT+CGACT=1,1</u>	第一个参数<state>指示 PDP 上下文激活的状态此处值为“1”表示激活的。第二个参数<cid>表示一个指定特定 PDP 上下文的数字参数。
OK	
+ZGIPDNS:1,1,"IP",	PDP 激活之后的上报。此命令为主动上报命令，在 CGACT 命令激活成功后主动上报 IP，DNS 地址。其中参数（包括省略部分）依次表示：<cid_num>，已经激活的 CID 的个数（由 CGACT 命令激活）；<cid_num>，含义同前；<PDP_type>，协议类型，此处“IP”即指 IPv4；PDP 地址；网关；首要 DNS；次要 DNS。

(10) RNDIS 连接控制

表 3.10 RNDIS 连接控制

指令或信息	含义
<u>AT+ZGACT=1,1</u>	用于连接或断开指定<cid>标识 RNDIS 链路。第一个参数为<state>，此处值为“1”表示连接。第二个参数为<cid>，表示一个指定上下文的数字参数。
OK	
+ZCONSTAT:1,1	RANDIS 连接状态显示。第一个参数为<stat>，此处值为“1”，表示 RANDIS 连接。第二个参数为<cid>，含义同前文同名处所述。

依次使用上述 AT 指令完成拨号后,在命令行中输入 `sudo udhcpc -n -i eth1` 命令注册路由,其中 `eth1` 为执行前文所述的 `ifconfig -a` 命令所查看到的新增的网络连接。此命令执行完成后即可开始访问网络。

3.2.2 断开连接

如果需要断开连接,则可在 `minicom` 串口通信界面下输入以下指令（并接收对应的反馈信息）:

表 3.11 断开连接指令及返回信息

指令或信息	含义
<u>AT+ZGACT=0,1</u>	即将第一个参数指定为“0”。
+CGEV:ME PDN DEACT 1	PS 域事件上报。“ME”表示手机操作。“PDN”表示该激活或去激活的上下文是与 LTE 里 PDN 连接关联或者是与 GMS/UMTS 里的主要 PDP 上下文关联。“DEACT”表示去激活。最后一个参数为<cid>，含义同前文同名处所述。
+ZCONSTAT:0,1	RNDIS 连接状态显示。含义同前文同名处所示，其中第一个参数值为“0”表示 RANDIS 断开。

3.2.3 关机

如需将 4G 模块关机，则在 minicom 串口通信界面下输入以下指令（并接收对应的反馈信息）：

表 3.12 关机指令及返回信息

指令或信息	含义
<u>AT+CFUN=0</u>	值为“0”表示完全关机。另有 AT+CFUN=4 为飞行模式。
+CREG:0	没有注册，且 ME 目前也没有正在寻找新的网络进行注册，下同。
+CGREG:0	
+CEREG:0	
OK	

3.2.4 其它重要的 AT 指令及常见错误信息

表 3.13 重要的 AT 指令及含义

指令	示例	含义
USB 端口组合切换命令	<u>AT+ZNCARD=1</u>	实现 USB 在 Windows 及 Linux/Android 操作系统的端口组合及切换，重启后生效。
+ZNCARD/+ZNETCARDTYPE	OK	ME3760_V2/C（公网）模块出厂设置默认为带 NDIS 口。
获取信号强度命令 +CSQ	<u>AT+CSQ</u> +CSQ:165,99	用于检查当前网络信号强度，检测接收信号的强度指示<rss>和信道误码率<ber>。

表 3.13 续

指令	示例	含义
PIN 码输入 +CPIN	<u>AT+CPIN?</u> +CPIN:READY	设置命令用于校验、解锁 PIN 和 PIN2。查询命令用来指示是否存在密码输入请求。可借此用于测试 SIM 卡是否被识别。
厂商信息获取 +CGMI	<u>AT+CGMI</u> +CGMI:ZTE	用于查询厂商信息。
产品名称获取 +CGMM	<u>AT+CGMM</u> ZTE-A370	用于查询产品的模型 ID。
运营商选择 +COPS	<u>AT+COPS=?</u> +COPS:.....	设置命令用于选择运营商。查询命令用于查询当前注册的网络状态、网络选择模式。执行命令能完成 GSM/UMTS 网络的自动与手动选择。读命令返回当前的网络选择模式，如果注册成功，则返当前的运营商信息。测试命令返回当前网络中存在的运营商的信息列表，最多不超过 20 个。
系统信息获取命令 ^SYSINFO	<u>AT^SYSINFO</u> ^SYSINFO:2,3,0,15,1	执行命令返回当前系统信息。返回信息对应的参数依次为: <srv_status>, <srv_domain>, <roam_status>, <sys_mode>, <sim_state>。 如示例中返回值的具体含义为: 有服务, PS 和 CS 服务, 非漫游状态, TD-SCDMA 模式（若为 17 则为 LTE 模式），卡状态有效。

表 3.14 常见错误信息及含义

错误信息	含义
+CME ERROR:3	不允许的操作。即命令正确但不是当前情况下可执行的命令。
+CME ERROR:6003	命令不识别。即可能命令名称、格式等有误。
+CMEERROE:6004	命令参数不对。

第四章 软件系统设计与实现

本章将详细介绍第二章中阐述的无人机通信系统中的软件系统部分。首先将介绍软件系统的总体设计情况，包括对软件整体进行功能模块的划分和简述、各部分所运用的主要技术以及软件的开发环境和使用工具等；然后再介绍各部分应用的具体设计和实现情况。

4.1 系统总体设计

如第二章所述，本系统主要由部署于服务器端的通信处理程序和部署于树莓派端的无人机通信控制程序这两部分组成，两部分应用程序之间采用 Socket 通信的方式进行交互。

其中部署于服务器端的程序采用 B/S 架构进行设计和开发。B/S 架构即 Browser（浏览器）/Server（服务器）架构，是一种区别于 C/S（Client/Server）架构的设计开发模式。这种模式统一了客户端，将主要的功能实现集中到了服务器上，开发人员不必再为系统适配不同的用户终端而耗费大量的开发和测试时间，同时也便于系统进行更新和维护。用户也无需在个人终端上安装客户端，节省了存储空间且免去了频繁的应用更新工作。用户只需要使用浏览器访问相关的页面即可通过服务器与后台的数据库进行数据交互。

4.1.1 功能划分

本系统主要包括用户管理、设备管理、飞行信息管理、指令信息管理和位置信息管理这五个功能模块。其中用户管理主要实现了用户登录操作和用户信息查询方面的功能。设备管理主要实现了无人机连接操作和无人机信息查询方面的功能。飞行信息管理主要实现了飞行记录的添加、查询和修改的功能。指令信息管理主要实现了指令的添加和查询功能。位置信息管理主要实现了无人机位置信息的添加和查询功能。

4.1.2 技术应用

从技术层面上看，使用 HTML, CSS 和 JavaScript 对前端页面进行开发和简单逻辑实现；使用 AJAX 对服务器发起异步请求同时进行页面信息的更新；使用 Java 编写 Servlet 对请求进行响应和处理；使用 C++ 开发对无人机的控制应用程序；使用 Socket 实现树莓派和服务端之间的通信。

4.1.3 开发环境和工具

OS: Windows 8.1, Ubuntu-Mate 16.04.2 LTS for aarch32(ARMv7) computers

Eclipse: Eclipse Java EE IDE for Web Developers (Version: Neon.3 Release (4.6.3))

Java Development Kit: jdk1.8.0_91

Tomcat: 7.0.70-windows-x64

MySQL: Version 14.14 Distribute 5.7.16, for Win64

Navicat for MySQL: 10.1.7 – Enterprise

Browser: Chrome 58.0.3029.96, UC V11.5.1.944

DJI SDK: Onboard-SDK-ROS 3.2

4.2 前端界面

前端界面是提供给用户与系统进行应用和数据交互的接口，用户在浏览器中输入正确的地址信息即可访问。用户通过前端界面发起请求，实现登录、连接设备以及控制设备等操作。

4.2.1 界面设计

前端界面主要包括用户登录界面（login.jsp）、设备连接界面（connect.jsp）和设备控制界面（control.jsp）。具体界面设计如下图所示：



图 4.1 用户登录界面



图 4.2 设备连接界面



图 4.3 设备控制界面

4.2.2 界面跳转逻辑

用户在登陆界面输入用户名和密码，点击确认按钮，若信息正确则跳转至设备连接界面，否则返回登录界面。在设备连接界面中输入设备名称和连接码后，点击确认按钮，若信息正确则跳转至设备控制界面，否则返回设备连接界面。

用户在设备控制界面点击返回按钮则跳转至设备连接界面。用户在设备连接界面点击返回按钮则跳转至登录界面。

4.3 服务器

根据前端页面的设计和树莓派的通信需求，服务器端的处理程序主要包括事物逻辑处理和 Socket 通信程序这两大部分。其中事物逻辑处理包括 LoginServlet, ConnectServlet 和 ControlServlet 这三个模块，分别对应前端的三个页面的操作和请求。Socket 通信程序包括 SocketListener, SocketThread, SocketOperate 这三个模块，实现从 Socket 监听的建立到响应、处理和反馈的完整过程。

4.3.1 事物逻辑处理

LoginServlet 对应于前端的 login.jsp（用户登录界面）。主要实现用户登录操作过程中的验证和判断，通过对数据库访问接口的调用，对提交的用户登录信息进行验证，对包含正确的用户名和密码组合的请求进行跳转，同时在服务器中生成会话信息（Session）；对其它不正确的情况进行提示。

ConnectServlet 对应于前端的 connect.jsp（设备连接界面）。主要实现对用户连接无人机设备的操作的验证和判断，通过对数据库信息的查询，对提交的设备连接信息进行验证，对包含正确的设备名称和连接码组合的请求进行跳转，修改保存在服务器中的会话信息，并在数据库中添加飞行记录、添加“连接”指令以及修改对应设备的状态；对其它不正确的情况进行提示。此外，实现用户点击“返回”按钮的操作（即退出登录状态），控制页面跳转至 login.jsp，同时清除会话信息。

ControlServlet 对应于前端的 control.jsp（设备控制界面）。主要实现对用户点击无人机操作按钮的事物处理，包括添加指令信息、查询位置信息并反馈。此外，实现用户点击“断开连接”的操作，控制页面跳转至 connect.jsp，修改数据库中的飞行记录，同时修改会话信息。

4.3.2 Socket 通信程序

主要实现服务器端的 Socket 通信程序。在 web.xml 中添加<listener></listener>标签并关联 SocketListener 监听类以配置相关的监听服务。SocketListener 实现随服务器启动的监听程序，

具体通过 SocketThread 创建线程启动监听，并在监听到客户端的请求后，创建包含新 Socket 的线程进行通信，并调用 SocketOperate 实现与客户端的数据交互以及数据库的操作，包括将客户端发送的位置信息保存至数据库和发送指令信息等。

4.4 数据库

服务器对来自客户端的请求进行响应后，根据逻辑需要对数据库中的数据进行相应的“增加”、“删除”、“修改”、“查询”等操作，然后将操作结果做进一步判断后反馈至客户端。根据本课题的应用需求，数据库设置五张表对数据进行记录，分别是：用户信息表（users）、无人机信息表（drones）、飞行记录表（flights）、指令信息表（instructions）和位置信息表（positions）。

4.4.1 表结构及数据类型

（1）users 用户信息表

表 4.1 users（用户信息表）

属性名称	数据类型	备注
name	varchar(255)	用户名
password	varchar(255)	密码

（2）drones 无人机信息表

表 4.2 drones（无人机信息表）

属性名称	数据类型	备注
drone_name	varchar(255)	无人机名称
connect_code	varchar(255)	连接码
state	varchar(255)	无人机状态
information	varchar(255)	无人机信息

其中 state 定义三种取值情况分别对应三种无人机状态：0-离线（无人机未开机），1-在线待机（无人机已开机但未被使用），2-在线使用（无人机正被使用中）。

（3）flights 飞行记录表

表 4.3 flights（飞行记录表）

属性名称	数据类型	备注
flight_num	varchar(5)	飞行编号
drone_name	varchar(255)	无人机名称

表 4.3 续

属性名称	数据类型	备注
username	varchar(255)	用户名
start_time	datetime	飞行起始时间
end_time	datetime	飞行结束时间

(4) instructions 指令信息表

表 4.4 instructions（指令信息表）

属性名称	数据类型	备注
flight_num	varchar(5)	飞行编号
time	datetime	指令产生时间
instruction	varchar(255)	指令

(5) positions 位置信息表

表 4.5 positions（位置信息表）

属性名称	数据类型	备注
flight_num	varchar(5)	飞行编号
time	datetime	位置信息更新时间
longitude	decimal(12,8)	经度
latitude	decimal(12,8)	纬度
height	decimal(10,4)	高度

根据上述表格间的关系所作 ER 图如下所示：

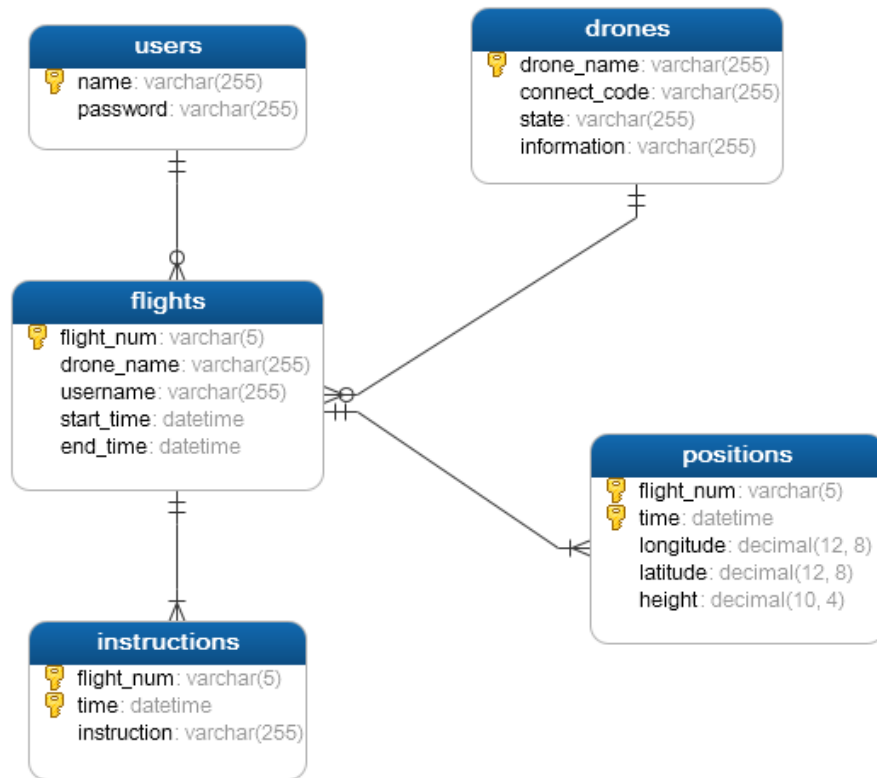


图 4.1 ER 图表

4.4.2 数据库操作

对数据库中的五张表分别定义相应的 DAO（数据访问对象）以实现对不同表进行所需的数据交互操作。分别为 UserDao, DroneDao, FlightDao, InstructionDao, PositionDao，对应操作的表为 users, drones, flights, instructions, positions。具体包含的操作方法如下：

（1）UserDao 包含方法：

1. 根据用户名查找密码

```
public String findPasswordByUsername(String username)
```

（2）DroneDao 包含方法：

1. 根据无人机名称查找所有无人机信息

```
public List<Map<String,Object>> findAllByDroneName(String droneName)
```

2. 修改无人机状态

```
public void modifyState(String droneName, String state)
```

3. 查询无人机状态

```
public String QueryState(String droneName)
```


（3）FlightDao 包含方法：

1. 查询所有记录中最后一个飞行编号

```
public String queryLastFlightNum()
```

2. 添加飞行记录

```
public void addFlight(String flightNum,String droneName, String username, Timestamp  
startTime)
```

3. 根据飞行编号查询飞行记录

```
public List<Map<String, Object>> queryFlight(String flightNum)
```

4. 根据飞行编号和实际结束时间修改结束时间

```
public void modifyEndTime(String flightNum, Timestamp time)
```

5. 根据无人机名称查询其上次飞行编号

```
public String queryLastFlightNumByDroneName(String droneName)
```

（4）InstructionDao 包含方法：

1. 添加对应飞行编号的新指令

```
public void addInstruction(String flightNum, String instruction, Timestamp time)
```

2. 查询某次飞行中（flightNum）某时间节点之后的最后一条指令

```
public List<Map<String, Object>> queryInstruction(String flightNum, Timestamp time)
```

（5）PositionDao 包含方法：

1. 根据飞行编号查询当前无人机位置信息

```
public List<Map<String, Object>> queryPosition(String flightNum)
```

2. 添加最新位置信息

```
public void addPosition(String flightNum, Timestamp time, String longitude, String latitude,  
String height)
```

4.5 树莓派

部署于树莓派端的通信控制程序主要实现树莓派与服务器的通信和树莓派控制无人机这两大部分的功能。其中通信部分由 Socket 实现，并在通信过程中根据接收到的具体指令调用相关的无人机操作方法。

启动树莓派通信控制程序后，进行相关的初始化操作，然后建立 Socket，循环不断地向

服务器发起查询指令信息的请求（直至接收到“disconnect 断开连接”的指令），并在请求信息中包含当前无人机的位置信息。而在接收到服务器返回的相应信息后，解析出其中的指令信息并调用相应的无人机 SDK 所提供的接口和方法通过串口通信完成对无人机的操作，具体指令和方法对应如下：

（1）connect 连接

启动验证信息

```
void activate(ActivateData *data, Callback callback = 0, UserData userData = 0);
```

获得无人机控制权

```
void setControl(bool enable, Callback callback = 0, UserData userData = 0);
```

（2）takeOff 起飞

```
ackReturnData monitoredTakeoff(CoreAPI* api, Flight* flight, int timeout)
```

（3）land 降落

```
ackReturnData landing(CoreAPI* api, Flight* flight, int timeout)
```

（4）disconnect 断开连接

```
cleanup(serialDevice, api, flight, &read);
```

（5）无人机获取相关飞行信息（包括位置信息）

```
FlightStatus getFlightStatus() const;
```

第五章 系统测试及分析

本章将简单介绍测试的情况和测试结果，并对系统性能和可能存在的问题进行分析以及给出相应的解决方案或建议。

5.1 测试情况和结果分析

主要针对完整应用流程的实现进行测试，进一步考察系统的性能。具体测试项目如下：

（1）页面显示测试

1. 不同屏幕尺寸的个人终端的显示情况（包括手机、PC、平板电脑等主流终端产品）。

测试结果：页面均显示正常，能按实际屏幕大小进行自动匹配缩放。

（2）登录页面功能测试

表 5.1 登录页面功能测试结果

测试项目	预期结果	实际结果
输入用户名和密码且正确	跳转至连接页面	正确
输入用户名和密码且密码错误	提示密码错误	正确
输入用户名和密码且用户名不存在	提示用户名不存在	正确
输入用户名且未输入密码	提示密码错误	正确
未输入用户名且输入密码	提示用户名不存在	正确
用户名和密码均未输入	仍处于登录页面且无提示	正确

（3）连接页面功能测试

表 5.2 连接页面功能测试结果

测试项目	预期结果	实际结果
输入设备名和连接码且正确	跳转至控制页面，数据库添加飞行记录，添加“connect”指令，修改无人机状态	正确
输入设备名和连接码且连接码错误	提示连接码错误	正确
输入设备名和连接码且设备名不存在	提示设备名不存在	正确
输入设备名且未输入连接码	提示连接码错误	正确

表 5.2 续

测试项目	预期结果	实际结果
未输入设备名且输入连接码	提示设备名不存在	正确
设备名和连接码均未输入	仍处于连接页面且无提示	正确
点击“exit”按钮	跳转至登录界面	正确

(4) 控制页面功能测试

表 5.3 控制页面功能测试结果

测试项目	预期结果	实际结果
飞行信息显示	显示当前飞行编号、设备名、用户名、开始时间等信息	正确
位置信息更新	初始显示“等待中”，然后以固定频率更新显示	正确
初始按钮状态	“takeOff”和“disconnect”按钮可点击，“land”按钮不可点击	正确
点击“takeOff”按钮	显示“takeOff”，数据库增加指令，“takeOff”和“disconnect”按钮不可点击，无人机起飞	正确
点击“land”按钮	显示“land”，数据库增加指令，“takeOff”和“disconnect”按钮可点击，“land”按钮不可点击，无人机降落	正确
点击“disconnect”按钮	短暂显示“disconnect”，所有按钮均不可点击，然后跳转至连接页面，数据库增加指令，修改飞行记录的结束时间，修改无人机状态	正确

(5) 其它项目测试

表 5.4 其它项目功能测试结果

测试项目	预期结果	实际结果
无人机位置信息发送	更新数据库数据，控制界面周期性更新显示	正确

经上述测试流程，本系统所规定的基本功能能够正常应用，同时能够保证基本的性能需求。在实际测试信号强度和访问速度上均有可靠和稳定的表现，其中信号强度能够维持在较高水平（在 LTE 模式下，信号强度<rssI>的取值范围为 100~199，测试时数值维持在 165~170），

而使用 PING 命令测试网络联通情况的结果显示网络联通情况好、数据传输完整、延时低，具体测试结果如图所示：

```
AT+CSQ
+CSQ: 165,99
OK
AT+CSQ
+CSQ: 165,99
OK
AT+CSQ
+CSQ: 165,99
OK
```

图 5.1 使用 AT 命令查询信号接收强度

```
pi@raspberrypi:~$ ping www.baidu.com
PING www.a.shifen.com (111.13.100.92) 56(84) bytes of data.
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=1 ttl=52 time=49.3 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=2 ttl=52 time=58.6 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=3 ttl=52 time=56.8 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=4 ttl=52 time=54.8 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=5 ttl=52 time=53.8 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=6 ttl=52 time=52.8 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=7 ttl=52 time=50.8 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=8 ttl=52 time=49.9 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=9 ttl=52 time=58.8 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=10 ttl=52 time=58.5 ms
64 bytes from promote.cache-dns.local (111.13.100.92): icmp_seq=11 ttl=52 time=57.8 ms

--- www.a.shifen.com ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 49.321/54.768/58.852/3.492 ms
```

图 5.2 使用 PING 命令测试网络联通情况

5.2 存在的问题和解决方法或建议

1. 前期开发时发现 4G 模块信号接收情况较差，无法满足应用需求。

解决方法：模块本身的硬件设计预留了天线接口，为模块添加天线后信号接收情况良好且稳定，网络访问速度快，满足应用需求。

2. 用户指令下达和无人机执行存在轻微时延。

原因：系统设计是无人机按固定频率请求服务器查询最新指令，可能出现查询时刻和指令产生时刻重叠的情况，导致新指令需要在下一个查询周期才能被查询。

解决方法：可采用适当减小请求周期，优化代码结构以减小代码执行时间，优化服务器

硬件质量、提高处理速度等办法；进一步可探寻使用性能更好的开发技术和框架，或设计更好的通信流程和方法。

3. 系统应对非正常通信情况的应对能力有待完善。

本系统着重实现正常应用条件下的基于 4G 模块的无人机无线通信，后期需要进一步设计和完善非正常情况下的应对预案，如：用户和服务器或无人机和服务器断开连接的重连设计、无人机遇到 4G 网络信号较弱的环境的安全飞行保障（如失去连接情况下采用自动返回至预设地点、悬停等待、选择备用连接方式连接服务器等解决方法）。

第六章 总结和展望

本课题研究和实现了基于 4G 模块的无人机无线通信，达到了课题预期的研究目的，但仍有许多地方需要进一步的完善和优化以获得更好的使用体验。

随着 4G 网络的深入建设，4G 信号的覆盖范围将不断的扩大，通信质量将会逐步提升，4G 通信技术必将会成为更加可靠的通信方式。应用 4G 技术的无人机将能够在更多领域为人们发挥更大的应用价值，满足人们不同的通信和应用需求，无人机将因此迎来更广阔的应用和发展空间。例如在某些特殊室内场景应用环境下，无人机的通信是通过接入 802.11n 标准的无线网络实现的，由于搭载信号发射源的设备的移动受环境制约且信号的穿透性较差，无人机的活动范围受到影响^[20]。而如果使用 4G 通信技术，则不仅能保证与场外人员的通信，同时能够使无人机能够摆脱信号源的距离限制进行更自主的飞行和探索；进一步地，可以减少使用相关的地面设备，降低应用成本和复杂度。而且随着运营商对套餐资费的进一步降低，人们所担心的应用成本问题也将会得到更好地解决，为 4G 通信技术应用于无人机通信提供更加有利的因素。

参考文献

- [1]黄泽满, 刘勇, 周星, 等. 民用无人机应用发展概述[J]. 赤峰学院学报, 2014, (24): 30~32.
- [2] RA Ramlee, DHZ Tang, MM Ismail. Smart home system for Disabled People via Wireless Bluetooth[A]. In: Dr-Tech Ary Setijadi Prihatmanto, Reza Darmakusuma, Rifki Wijaya et al ed. International Conference on System Engineering and Technology[C]. Bandung, Indonesia: IEEE, 2012: 1~4.
- [3]D Yan, Z Dan. ZigBee-based Smart Home system design[A]. In: Wen Desheng, Wang Ruofeng, Xie Yi ed. 2010 3rd International Conference on Advanced Computer Theory and Engineering[C]. Chengdu, China: IEEE, 2010, (2): 650~653.
- [4]唐思敏. WIFI 技术及其应用[J]. 福建电脑, 2009, (10): 59, 82.
- [5]周洁. 基于 WiFi 传输与接入技术的发展[J]. 信息通信, 2012, (2): 115~116.
- [6] Mahdi Asadpour, Domenico Giustiniano, Karin Anna Hummel. From ground to aerial communication: dissecting WLAN 802.11n for the drones[A]. In: Albert Banchs, Sunghyun Choi ed. 8th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTech 2013)[C]. Miami, Florida, USA: ACM, 2013: 25~32.
- [7]张秀爱. 蓝牙技术综述及其发展前景[J]. 中国科技信息, 2008, (16): 128,131.
- [8]符鹤, 周忠华, 彭智朝. 蓝牙技术的原理及其应用[J]. 微型电脑应用, 2006, 22 (7): 60~61, 6.
- [9]梁旭. ZigBee 技术的特点及其在无线网络中的应用[J]. 信息通信, 2014, (11): 186.
- [10]张兰. 浅谈 Zigbee 无线通信技术特点及应用[J]. 计算机光盘软件与应用, 2014, (1): 297, 299.
- [11]王虹, 蔡喜平. 基于白光 LED 的可见光通信研究进展[J]. 半导体光电, 2014, 35 (1): 5~9.
- [12] H Burchardt, N Serafimovski, D Tsonev et al. VLC: Beyond point-to-point communication[J]. Communications Magazine IEEE, 2014, 52(7): 98~105.
- [13]朱月秀, 林野. 正交频分复用技术及应用研究[J]. 杭州电子科技大学学报, 2007, (5): 21~24.
- [14]王亚军, 张艳. 4G 通信中的关键技术之智能天线技术[J]. 信息通信, 2015, (1): 213~215.
- [15]庞进, 陈金鹰, 陈斌. 4G 的核心——MIMO 及其关键技术[A]. 见: 中国通信学会编. 四川省通信学会 2005 年学术年会论文集[C]. 成都: 四川省通信学会, 2005.
- [16]陈忠民, 田增山. 浅谈软件无线电技术及其在 4G 中的应用[J]. 电信快报, 2006, (1): 44~46.
- [17]代延村. MIMO-OFDM 系统中的多用户检测技术[D]. 西安: 西安电子科技大学, 2010.
- [18]牟亚南, 陈金鹰, 朱军. 移动 IPv6 技术在下一代互联网的应用探讨[J]. 通信与信息, 2013, (6): 72~74.
- [19]陶金牛, 钱坤, 刘凯, 等. 基于 4G 移动网络的无人机及其应用前景分析[A]. 见: 中国航空学会编. 2014 (第五届) 中国无人机大会论文集[C]. 北京: 航空工业出版社, 2014.
- [20] Nathan Michael, Shaojie Shen, Kartik Mohta et al. Collaborative Mapping of an Earthquake-Damaged Building via Ground and Aerial Robots[J]. Journal of Field Robotics, 2012, 29(5): 832~841.

致 谢

这次毕业设计是在我的指导老师翟象平老师的悉心指导下完成的。在课题研究的过程中，翟老师不仅帮助我对课题进行深入地分析，指导我尽快掌握课题研究目的和研究方法，更是在我遇到困难、停滞不前时，跟我一起探讨和分析解决问题的办法，使我能够更好地完成毕业设计工作。在此，我要由衷地感谢翟老师对我的关心和指导。

在学习使用设备及相关知识的过程中，邹馨仪同学给了我大量的帮助，提供了许多学习资料，在此表示衷心的感谢。

另外也感谢我的室友，在我的学习和生活中提供了许多帮助，他们的理解和支持使我能够更加顺利地完成这次毕业设计。