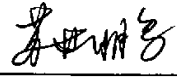


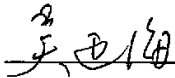
论文题目: 手机软件仿真平台的研究与实现

专 业: 通信与信息系统

硕 士 生: 苏世鹏

(签名) 

指导教师: 吴延海

(签名) 

摘 要

信息产业飞速发展,移动终端设备已成为各国发展的热点。本文主要是对移动终端设备(手机)软件仿真平台的研究与实现。结合所学理论知识,文中分析了GSM手机的硬件组成、软件结构、工作原理及实现方法,着重研究了实习公司手机开发平台的硬件芯片模组和软件模块划分及其设计方法。掌握了该平台的软硬件工作原理、设计思路和开发调试工具,为仿真平台的开发奠定了基础。

手机软件仿真平台的目的是实现手机软件界面部分的模拟,使得在模拟器中开发的代码可以移植到实际的手机上去。工作的重点分两部分:一是在PC上模拟手机软件运行所需的硬件部分,如模拟LCD的显示输出,键盘的模拟,flash ROM的模拟等;二是在PC上模拟MMI层软件运行所需的其它层软件,如通信模块的模拟,实时操作系统的模拟,计时器的模拟等。

文中首先提出了仿真平台的设计方案,接着论证其方案的可行性,在此基础上对仿真平台做了整体设计,然后对该平台做了模块划分,并对各个模块分别进行了详细设计和代码编写,最后,设计和实现了仿真平台的上层结构,成功的在PC机上搭建了手机软件的仿真平台。该仿真平台已经开始应用于实习公司的手机开发,并取得了较显著的成效。

关键词: GSM协议; MMI; Task; 仿真

研究类型: 应用研究

Subject : The Research and Development of Mobile Software Simulation Platform

Specialty : Communication and Information System

Name : SuShiPeng (Signature) Sushi peng

Instructor: WuYanHai (Signature) Wuyanhai

ABSTRACT

With the highly development of information industry, the mobile terminology has become a hotspot through the world. This essay mainly focused on the research and analysis of the mobile terminology software which is the mobile software simulation platform. The hardware structure, software structure, working theory and realization method are analyzed based on the theoretical information. Within which the partition and design of the hardware chip module and software module were the main concern. With the knowledge of hardware and software working theory, design method as well the development tool, the development of new functions are now of a stable base.

The main objective of mobile software simulation platform was to realize the interface simulation in order that the code developed by simulator can be immigrated to the real mobile. In other words, there are two important parts in this process: one is to simulate the hardware environment needed by the mobile software. For example: the simulation of output of the LCD, the simulation of flash ROM etc. Another is to simulate the other layer software needed by the MMI layer software on PC e.g. the simulation of telecommunication module, the simulation of real time operating system as well as the timer simulation etc.

Furthermore, a design proposal was issued in the essay. Based on which an overall design of the simulation platform was given. After divide the platform into different modules while detail design and code written applied to each module respectively. Finally the design and realization of the upper-structure of the platform enabled an updated mobile software simulation platform. What's more, the software has been applied to the development in the company while obvious accomplishment has been achieved.

Keywords: GSM protocol MMI Task Simulation

Thesis : Application Research

西安科技大学

学位论文独创性说明

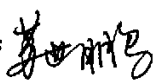
本人郑重声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中加以标注和致谢的地方外，论文中不包含其他人或集体已经公开发表或撰写过的研究成果，也不包含为获得西安科技大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

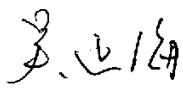
学位论文作者签名： 日期：2006.06.08

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西安科技大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西安科技大学。

保密论文待解密后适用本声明。

学位论文作者签名：

指导教师签名：

2006年4月20日

1 绪论

1.1 移动通信发展与现状

随着 20 世纪大规模集成电路和计算机软件技术的发展，现代社会已经进入信息时代。由于人类社会生活对通信技术，尤其是对“个人通信”的需求越来越高，移动通信目前已经成为各国通信技术发展的热点。移动通信已经历了数十年发展，但其真正的兴起还是在 20 世纪 80 到 90 年代。到目前，移动通信已经自成一个网络体系，与综合业务数字网（ISDN），公用电话交换网（PSTN），因特网（Internet）等连接互通。

最早的移动通信系统是模拟移动通信系统，到了 70 年代这种系统所能容纳的用户数目都还是相当低。蜂窝移动通信是在 70 年代才开始出现的名词。70 年代初，美国贝尔实验室提出了蜂窝系统的概念和理论。并按此理论进行了 AMPS（Advanced Mobile Phone Service）业务和系统性能的综合试验，并于 80 年代初投入商用。AMPS 系统的出现标志着第一代蜂窝移动通信系统（1G）的诞生。同时，其它各国研制的 TACS，NMT450/900，RadioComm2000，C-450 及 NTT 也属于第一代移动通信系统。

针对第一代模拟移动通信系统具有诸多缺点和不便利的因素，欧洲电信管理部门会议 CEPT 于 1982 年成立了 GSM（group special mobile —移动特别小组）专题小组，开始制定用于泛欧的一种数字移动通信系统的技术规范，1987 年最终确定了系统标准，并将标准命名为 GSM。

GSM 系统的出现标志着第二代移动通信系统的诞生。与第一代相比，第二代数字蜂窝移动通信系统（2G）具有如下特点：

- 先有标准，后有设备；
- 安全保密性好；
- 适应固定网的数字化发展；
- 数字方式传输，系统容量提高。

由于 GSM 标准的开放性，世界上许多著名的通信公司都在制作和提供 GSM 系统设备，因而系统获得了广泛的应用，取得了巨大的商业成功。GSM 系统已在世界上 100 多个国家和地区获得了应用。目前全世界有移动用户约 10 亿左右，并以每天 25 万的速度增加；中国的移动用户数近 4 亿，并且以 120 万/月的速度增加，在已有及增加的用户数中，绝大部分为 GSM 系统用户，其它的为 CDMA，D-AMPS，PDC 模式等。目前，中国的 GSM 用户远超过其它用户^[1]。

1.2 国产手机现状

目前中国手机保有量大约在 3.5 亿部, 相对于拥有 13 亿人口的国家来说, 手机普及率还不到 30%。与一些发达国家和地区相比, 例如西欧的 80.1%、北美的 59.8% 以及日本的 69.7%, 中国的这个数字是很小的。随着中国经济的不断发展, 中国的手机普及率也必然随之提高。在中国, 每增加一个百分点, 就代表着一个巨大数字的市场容量。

同时, 国产手机厂商经过整合调整能够逐步提高竞争能力。如果辅以适当的经营策略, 未来国产手机厂商是能够有所作为的。

赛迪顾问分析师蒋利峰就提出: “从长远来看, 我不仅看好中国手机市场的发展, 同时也看好中国手机企业的发展。对于国产手机厂商来讲, 要充分利用自己的优势, 形成符合自身发展的商业模式或赢利模式, 而不应该以自己的短处和国外厂商的长处来竞争。就相当长的时期看, 制造方面将始终是中国企业的一个优势, 同时, 国内企业也完全可以向韩国的企业学习, 在工业设计、制造工艺、相关领域配套产业建设以及品牌营销方面发挥更多的创造空间, 以“外围包围核心”的策略打造具有国际竞争力的著名企业。”

2005 年政府监管部门也通过出台一些政策为国产手机产业搭桥铺路, 如国家发改委把手机生产牌照的发放由原来的审批制变为核准制。“从政府监管角度讲是一种进步。过去审批制是保护落后, 现在的核准制更加符合市场经济的运行规律。从长期来看, 这对国内的手机产业会是一件好事。”陆刃波作出了这样的评价。

此外, 正在进行的“黑手机”整治行动将会进一步净化市场环境, 为国产手机产业创造良性的发展环境。

1.3 移动通信技术发展趋势

移动通信发展迅速, 终端设备不停更新换代, 其应用软件则种类繁多, 各显特色, 通信协议也在升级。

随着 GPRS、MMS 正式开通, 彩屏手机将开始展现其魅力, 从而完成手机从听到看的功能延伸。手机市场将进入了由彩屏、图像、MMS (多媒体信息服务) 和 Java 推动的具有先进功能和服务的新阶段。尽管 65536 色的显示屏足以令人炫目, 但在市场上已经出现了可达 26 万色显示的彩屏机。而且在日本和韩国用户可以真正的享受到 3G 服务。虽然 3G 泡沫破了, 但这是人们对它寄与过高期望的结果, 以后移动通信、整个信息产业的前景无可限量。GPRS、PDA 功能、红外、蓝牙、数码摄像头、MMS 和 Java 等功能, 更是被越来越多地应用在了彩屏手机上。

多普达 686 手机的问世, 引起轰动。那就是 3C 的融合 (手机、PDA 和 PC 三合一), 我们可以称之为“智能手机”。这种智能手机拥有看电影、看 Flash 动画、听广播、

听 MP3 等多媒体处理能力。3C 的融合只是一个基础，而搭建无线互联网平台，把现有的以语音为中心的服务概念彻底转变成以数据为中心才是手机未来的发展之路。

同时，在通信协议上，现今情况也是在扑朔迷离中，渐有一丝明朗。

2002 年 10 月份，饱受争议的大唐 3G 标准 TD-SCDMA 终于得到了强有力的政策支持。3G 频段划分中，中国留出了 155Mhz 的 TDD 频段。这件事情在业界随即带来巨大影响。

以 NTT DoCoMo、爱立信、诺基亚、西门子为首的 WCDMA 联盟率先出手，宣布不会提高 WCDMA 手机和基础设施设备的价格，而会使其“公平合理并具有竞争力”；国内的主要通讯设备商紧急召开内部会议，重新磋商对 TD-SCDMA 的研发定位。据透露，华为有一个 200 多人的研发队伍在跟踪 TD-SCDMA 的发展。中兴的 3G 研发队伍中，WCDMA 属于移动事业部，CDMA2000 隶属 CDMA 事业部，中兴方面表示近期将整合三种标准，对三种标准都将有所投入；菲利普公司、三星公司和大唐移动合资成立 TD-SCDMA 终端芯片研发中心，据称其很多研发人员来自菲利普和美国硅谷业内资深人员，每年的预算将超过 1.2 个亿人民币。

中国作为全球最大的信息通信技术市场之一，已经成为各通讯巨头的必争之地。2006 年中国移动用户可能超过 5 亿户，未来几年基础设施的投资总额将会超过 5000 亿元，移动终端的销售总额也将超过 5000 亿元，两者相加将会超过万亿元。而由移动通信产业带动的相关产业的产值将很可能超过 10 万亿元。

中国移动通信已经丢失了第 1 代，错过了第 2 代，在 3 代上，我们期望中国政府、运营商和制造商等密切配合、相互支持，以促进中国 3G 系统早日完善^{[2][3]}。

1.4 手机软件仿真平台研究的重要性及意义

随着微处理芯片性能的不断提高，各种高性能的基于嵌入式系统的消费类电子产品逐渐出现在我们的日常生活中，例如手机、掌上电脑等。和已往的工业用嵌入式系统相比较，这些系统的处理器性能明显提高，需要丰富且易操作的图形用户界面。对应用的开发者来说，程序的复杂度大大提高，开发和测试所需的时间明显延长，又由于基于嵌入式平台开发软件需要更多的专门技术，需要对开发人员更多的培训，所以，嵌入式平台的软件开发成本较之基于 PC 的软件开发成本要高得多。与此形成鲜明反差的是：PC 上的 C 语言开发环境已经相当成熟，一些非常优秀的开发调试环境例如（Visual C++ IDE）已经提供了很多强大而实用的工具，使 PC 上的开发和调试变得非常方便和快捷。

那么是否能够采取某些方法，使得开发人员可以利用 PC 上良好的开发环境来开发和调试基于嵌入式平台的应用呢？答案是肯定的。这种方法的基础在于，嵌入式系统上的应用同样是用 C 语言开发的，如果我们能够在 PC 上模拟嵌入式应用的系统环

境，就可以做到在 PC 上编写和调试嵌入式系统的应用程序。

如果能够利用 PC 平台的开发调试环境来开发手机应用，可以较大地节约开发成本，缩短开发时间，并对应用进行更快、更完善的测试。

1.5 论文的工作和结构

在攻读研究生期间，本人有幸参加了深圳某公司自主开发的手机项目研究工作。在一年多的时间里，本论文工作可分为如下几个阶段：

第一阶段：在原有软硬件基础理论上熟悉 GSM 手机系统的软硬件构架。首先了解手机芯片构成，采用的是实时操作系统，然后是软件系统模块划分，模块间接口。了解了系统结构框架后，对手机系统进行了深入学习，重点是移动终端所涉及到的相关理论。

第二阶段：在理解了手机系统结构、模块划分的基础上，重点实现了对应用层分析和优化。在应用层软件设计中重点是人机交互（MMI：Man Machine Interface）的设计与实现，MMI 在系统中作为一个 Task 运行，其结构分为三层（Application，GUI，Framework）实现。

第三阶段：延续 MMI 模块接口设计，完成了手机软件仿真平台的设计与实现。

根据所完成的工作，以及技术方案将论文结构安排如下：

第一章：绪论。介绍移动通讯系统、国产手机发展现状、未来前景以及本论文研究的意义。

第二章：GSM 协议体系结构。在分析了 GSM 系统体系结构和其 L3 层软件细化说明基础上初步实现了主程序框架设计。

第三章：GSM 手机系统整体结构介绍，从硬件软件整体结构、协议、驱动等几方面阐述。

第四章：人机交互（MMI）的设计实现。重点分析了 MMI Task 的三层结构，以及 MMI Task 与其它 Task 的交互过程，并描述了 MMI Task 的运行机制。

第五章：手机软件仿真平台的设计与实现。首先给出了仿真平台的设计方案并论证其可行性，接着根据其设计方案对仿真平台做了整体设计，然后对各个模块分别进行了设计和实现，最后给出了仿真平台的运行流程和结果仿真图。

论文最后对工作进行了总结。阐述课题成果、不足之处以及展望。

2 GSM 协议体系结构

2.1 GSM 协议体系概述

2.1.1 GSM 协议体系

GSM 中采用了 OSI 的分层协议结构。其中下一层协议为上一层协议提供服务，上一层协议利用下一层所提供的功能，上下层之间通过原语进行通信。在建立连接之后，对等层之间形成逻辑上的通路。在信令系统中，可简单分为物理层、链路层、网络层。

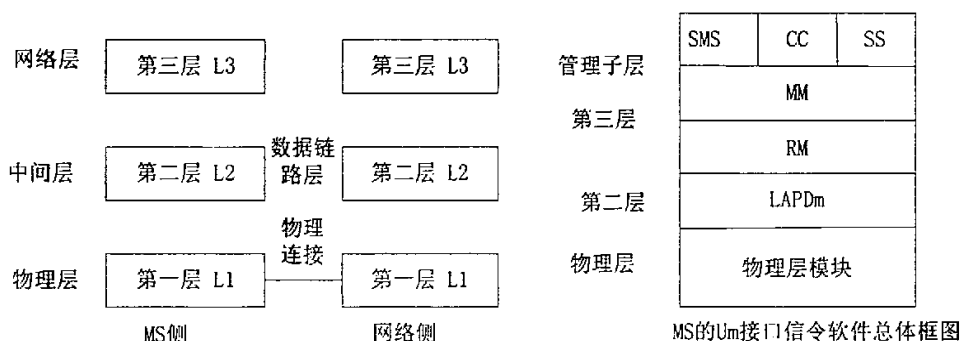


图 2.1 GSM 协议结构分层模型

第一层：物理层。物理层是 L3 协议体系中的最低层，它包括各类信道，为高层信息的传输提供基本的逻辑信道；每个逻辑信道都有自己的服务接入点（SAP），移动台的接入方式采用多址接入方式，可以在空闲时间检测周围的无线电环境，把检测结果定时的传给基站，确定是否进行小区切换。

第二层：链路层。链路层属于中间层（LAPDm：Link Access Protocol for Dm Channel Dm 信道链路连接协议）。它包括各种数据传输结构，对数据传输进行控制。LAPDm 的基本功能是把单个的 bit 构成一个集合，以便在移动台和基站之间提供可靠的无线数据链路，所有的链路功能都是建立在这个基本结构单元上，这个单元称为帧（frame），帧的结构采用 HDLC（高层数据链路控制）的定义方式。链路层协议主要由 GSM04.05、04.06 系列标准来描述。

第三层：网络层，包括无线资源管理子层、移动管理子层及呼叫控制子层

- 无线资源管理子层（RR）为公共控制信道（CCCH）和专用控制信道（DCCH）操作提供控制功能。
- 移动管理子层（MM）提供以下控制功能：
移动台号码的保密、确认身份、位置登记、用户登记和取消、定时登记

• 呼叫控制子层：包括呼叫控制（CC），短消息业务（SMS）和附加业务（SS）实体。

呼叫控制包括呼叫建立、释放和呼叫有关的补充业务处理等；

短消息业务实体功能是利用信令信道为用户提供短消息的服务；

附加业务实体是提供独立的附加业务^[4]。

2.1.2 GSM 协议栈结构

如前文所述，GSM 系统按照开放互联系统（OSI）模型采用分层的协议结构设计，并按通信过程分为三个层次，即物理层 L1，数据链路层 L2 和网络层 L3。我们将协议体系进行必要的转化，提出协议栈的概念，并按协议规范对协议栈的功能确定如下：

这里提到的协议栈是对应 GSM 标准的 L2 和 L3 层，其中信令 L3 层又被分为无限资源管理（RM），移动管理（MM）和连接管理（CM）。而连接管理（CM）又分为三个协议实体，分别是呼叫控制（CC），补充业务（SS）和短消息业务（SMS）。同时还实现 L3 层之上的短消息的第四层和第五层结构，分别为中继层（SM-RL）和传输层（SM-TL）。此外还需要支持 SIM 卡单元和小区广播的功能模块。

2.2 L3 层软件分析

由协议栈总体结构的描述可知，L3 层是整个协议栈的核心，它包括了大部分的手机功能实现。它提供一个蜂窝移动网和与其相连接的其他公众移动网之间的建立、维护和释放电路交换的功能；提供必要的支持补充业务、短消息业务和呼叫控制的功能；L3 还包括移动管理和无限资源管理的功能。此外，最新开发的 L3 层应当能支持 GPRS 业务，并提供相应复杂的控制功能。在软件设计过程中，L3 层主要由大量的程序模块构成，这些程序块在第三层各主体之间、第三层与相邻层以及相关层之间传递携带各种信息的信息单元。因此可以说，解决了 L3 层协议的设计也就基本上解决了整个协议栈软件的设计工作。下面将主要围绕 L3 层协议进行讨论。

GSM 标准规定 L3 层由如图 2.2 所示的协议结构。

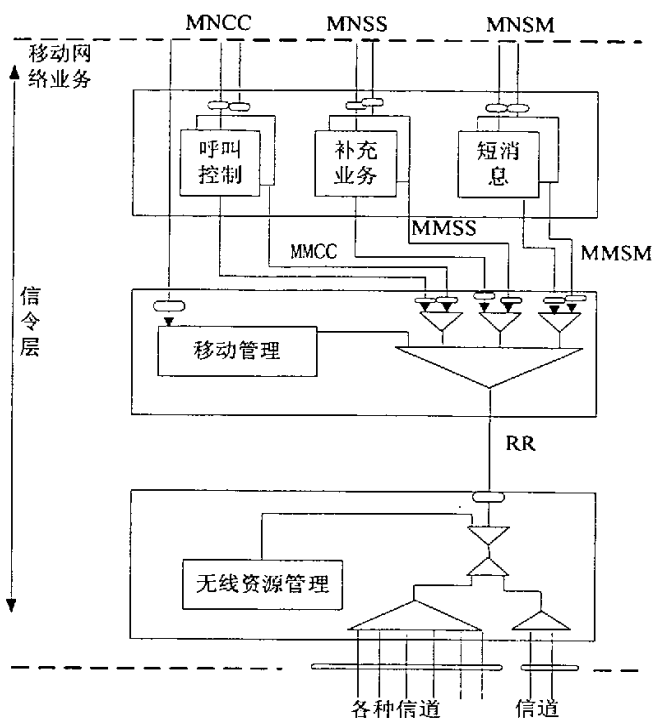


图 2.2 GSM 系统 L3 层移动台一侧协议结构

在实际软件实现中，根据具体功能模块和相关接口的分析，将图 2.2 中的 L3 协议层次结构转换为图 2.3 所示的软件结构图：

由图 2.3 可见，L3 层被分成了若干个功能单元，而这些功能单元通过相关接口进行原语通信，使用其他单元或服务，同时通过与应用层（AL）之间的接口，向应用层提供必要的支持，从而向用户城乡完整的功能实现。为了更好的理解 L3 层的模块，将分别对各个单元进行说明。

2.2.1 无线资源管理实体

无线资源管理实体（RR: Radio Resource）子层负责对无线链路连接（物理）和数据链路连接（逻辑）的建立、保持和释放。在空闲模式下，RR 子层负责协调小区的选择和重选。在专用模式下，RR 子层负责协调从报告检测结果到信道重新配置的整个切换过程。

从图 2.3 中可以看出：RR 子层通过 RR 接口向 MM 子层提供服务，同时，通过 DL 和 MPH 接口使用由 L2 层和 L1 层提供的服务。并且，RR 子层还通过 RRCB 接口，接收 SM-CB 单元发来的命令，来控制接收小区广播数据。图中箭头两侧是相关数据的交互，通过消息的交互实现 RR 子层的功能。

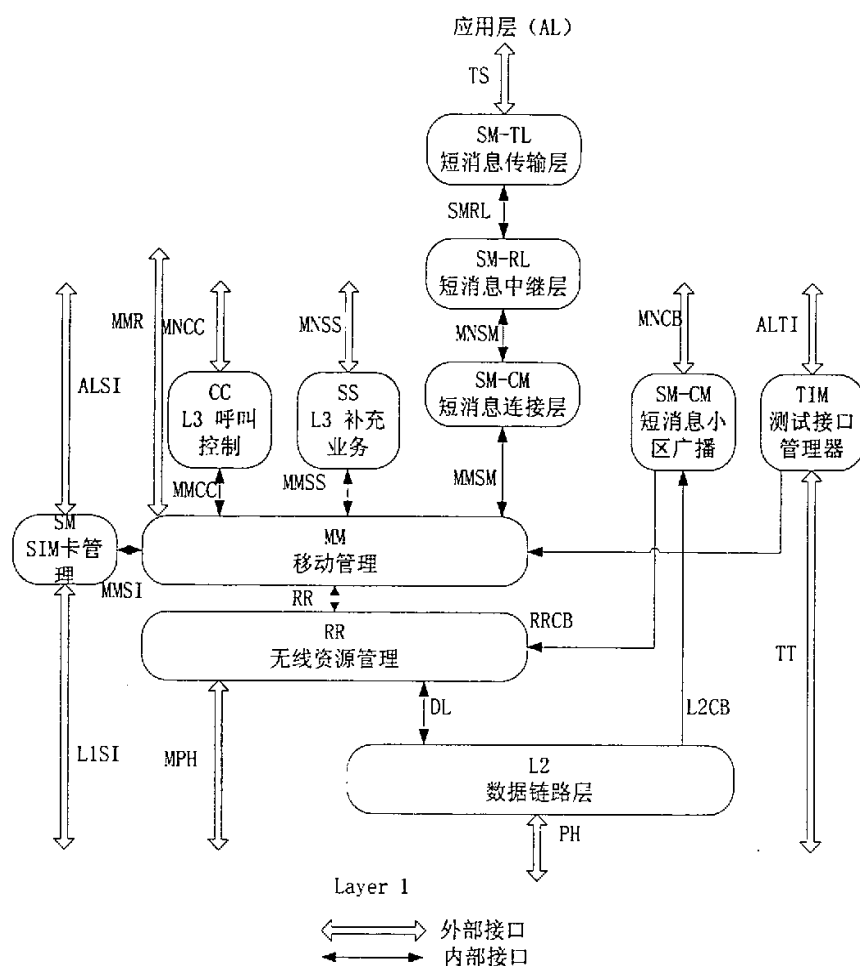


图 2.3 L3 软件结构图

2.2.2 移动管理实体

移动管理实体 (MM: Mobile Management) 子层主要考虑移动站在空闲模式下如何向网络通报它的地理位置。一个大的 PLMN 服务区域被划分为许多小的定位区域 LA, MM 子层的任务就是当移动站进入一个新的区域时通知网络, 以便能够继续跟踪移动站。此外, MM 子层还将所有其他上次实体的呼叫多路复用到一个单一的通道, 同时 MM 子层完成呼叫重建任务的过程相对用户是透明的。

由图 2.4 可知: MM 子层通过 MMCC、MMSS 和 MMSM 接口向 CM 子层实体 CC、SS 和 SM-CM 提供服务, 同时, MM 向应用层提供注册和取消注册的服务。MM 子层通过 RR 接口使用 RR 子层提供的服务, 通过 MMSI 接口使用 SIM 提供的服务。

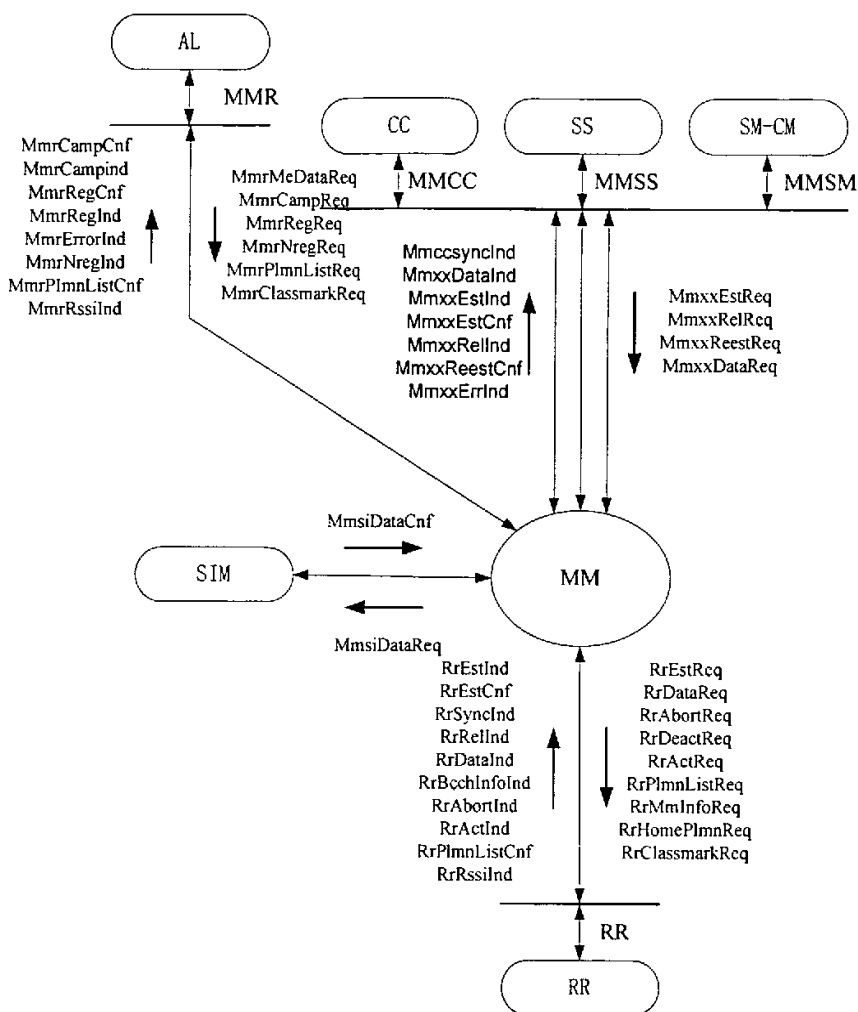


图 2.4 MM 单元结构

2.2.3 呼叫控制实体

呼叫控制实体 (CC: Call Control) 单元负责移动终端与网络一方的 MSC (移动交换中心) 之间的操作, 它管理与呼叫建立和保持相关的任务, 这些任务对用户来说是不可见的。CC 子层要为呼叫议定一条合适的通路, 并告知用户呼叫进行的情况 (即是否被叫已经振铃、占线或号码无效等)。每个呼叫, 不管呼入还是呼出, 都有自己专用的 CC 呼叫实体与之对应。CC 使得用户不必面对网络或无线接口的出错信息, 保证呼叫双方的控制信息按照正确的顺序进行。

由图 2.3 可知: CC 单元通过 MNCC 接口向应用层提供服务, 通过 MMCC 接口使用 MM 单元服务。每个 CC 实体负责一个声音或者数据呼叫。移动台可以并行发起 7 个

呼叫或者接收 7 个呼叫。但是, 在某一时刻, 只能有一个激活(只有一个传输信道存在), 其它处于保持状态。可设置同步呼叫的个数。

2.2.4 补充业务实体

补充业务实体(SS: Supplementary Services)是指在基本业务上的修改和扩充。主要是允许用户根据自己的需要改变网络对呼入和呼出的处理, 或者通过网络向用户提供某种信息让用户能够智能化的利用一些常规业务。绝大部分移动通信的补充业务是从电信网中继承过来的, 因此这些补充业务与固定网的补充业务没有本质的差别, GSM 数字移动通信的目标就是能够提供各种补充业务, 让用户有充分的选择余地。而 SS 单元则主要负责传输和接收设备消息中的呼叫独立补充业务的数据, 并且初始化 MM 连接的建立和释放。

由图 2.3 可知: SS 单元通过 MNSS 接口向应用层提供服务, 通过 MMSS 接口使用 MM 层提供的服务。同时只能建立 7 个连接。同步连接的实际个数由设定选项提供。

2.2.5 点对点短消息实体

点对点短消息(Short Message)业务使移动用户可以发出或接收长度有限的数字或文字消息。GSM 系统提供的短消息业务可以使网络端知道对方是否已经收到所发的消息。如果传送失败被叫方没有回答确认消息, 网络一侧会保留所传的消息; 一旦网络发现被叫方能被接通时, 消息将被重发以确保让被叫方收到。另一种增强的性能是主叫方能够得到消息发出以后的结果—被叫确认是否送到。

2.2.6 短消息小区广播单元

短消息小区广播(Short Message —Cell Broadcast)是另一种 GSM 系统特有的性能, GSM 移动中心可向确定移动台周期性地广播特定的数据信息。支持此业务的移动台能连续地监测广播信息并向用户显示。这一功能类似于寻呼台给用户发送的群呼信息, 如天气预报和股票信息。而 SM-CB 单元则在 L2 所提供的小区广播数据和应用层所要求的整个小区广播消息之间架起了桥梁, 将小区广播数据连接成应用层要求的完整信息并滤除应用层没有要求的小区广播数据。

由图 2.3 可知: SM-CB 单元通过 MNCB 接口向应用层提供服务, 通过 L2CB 接口从 L2 接收小区广播数据, 通过 RRCB 接口使用 RR 服务。

2.3 L3 层软件模块总体设计

通过以上说明, 可以看到, 每个单元根据其自身状态机, 对它所接收到的消息和将要发出的消息进行相关处理, 从而实现该单元的功能, 并向上层提供服务。可以说, 无

论一个单元的功能多么复杂，它都遵循这一基本原则。而正因为这些模块单元存在这样处理上的相似性，在程序结构中，除了具体的消息处理过程不同，每个单元的主框架是相同的。

设计单元主流程如图 2.5 所示：主流程为一循环机制，不断地去轮询有没有新的消息到来，如果有消息到来，即对此消息进行判断，并进行必要的消息处理，包括读取接收到消息中的参数，实现自身状态的转换，构造所需发送的消息并发送到其它模块。

在实际的 GSM 软件设计中，将图 2.3 中每个单元定义为一个 Task（作业），将消息定义为信号。将一个单元定义为一个 Task，实际上是对其进行模块化的设计，同时也便于程序的封装和连接，而将消息定义为信号，可以方便实现对消息的判断和具体操作。

Task: 每个 Task 具有一个输入队列，其中保存所有发送到此 task 的消息。且每个 task 拥有一块保留的存储区——堆栈。这个堆栈保存局部变量和函数返回地址。一个 task 被设计用于接收，处理和释放该信号，或者根据情况创建信号并将其发送到其它作业。

信号: 每个信号由两部分组成，一部分是信号的 ID，另一部分是信号的实际内容，称为信号体。定义信号缓冲区包含所有的信号，包括 2 个字节的信号 ID 和一个包括所有信号体的联合体。每个模块只定义包含自己所需信号的联合体。

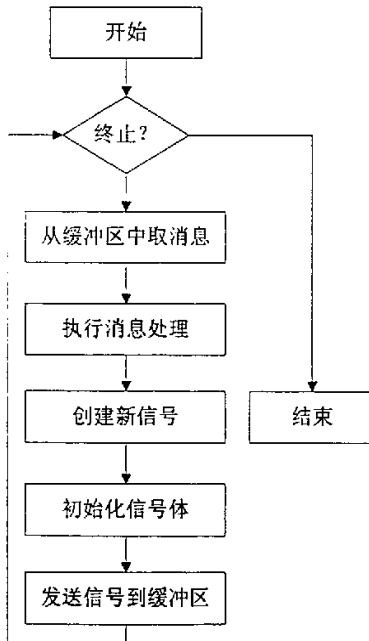


图 2.5 协议软件单元主流程

2.4 本章小结

本章在简单分析 GSM 协议的基础上，重点研究了 GSM 协议的网络层（L3 层），并对 L3 层的各个模块作了简单的描述。分析各个单元自身的特点，并总结其共同点，完成了软件模块的总体设计。

下一章将是本项目所开发的手机系统架构。

3 手机系统架构

3.1 概述

本项目所开发的手机支持电路交换和分组交换，可以提供语音和数据应用服务。同时，支持双频带接入（GSM900 和 DCS1800），工作频段如下所示：

EGSM900,Class4 (2W)

传输频率 880-915MHZ

接收频率 925-960MHZ

所有信道的灵敏度最好可达到-106dBm

GSM 1800,Class1 (1w)

传输频率 1710-1785MHZ

接收频率 1805-1880MHZ

所有信道的灵敏度最好可达到-104dBm

除了 GSM 协议规定的移动终端功能外，该移动终端还支持 MP3，MPEG4，照相等多媒体功能。以下章节将从硬件、软件两个方面对该开发平台进行概要描述。

3.2 系统硬件组成

手机硬件从电路结构看，分为逻辑系统、射频系统、电源系统三个部分。这 3 个部分相互配合，在逻辑控制系统统一指挥下，完成手机的各项功能。本论文所讨论的手机硬件系统主要由一些主芯片组和一些外围器件构成。主要的硬件模块包括：基带芯片 MT6219（该模组主要由 ARM7 微处理器、DSP 和一些外围设备组成）；RF 模块（由收发信机 MT6129、功率放大器 RF3146 和天线开关组成）；电源管理芯片 MT6305；人机接口设备：键盘、液晶 LCD；camera 模组；NOR flash 芯片；NAND flash。整个手机的硬件架构如图 3.1 所示。下面简要介绍硬件模组的特性。

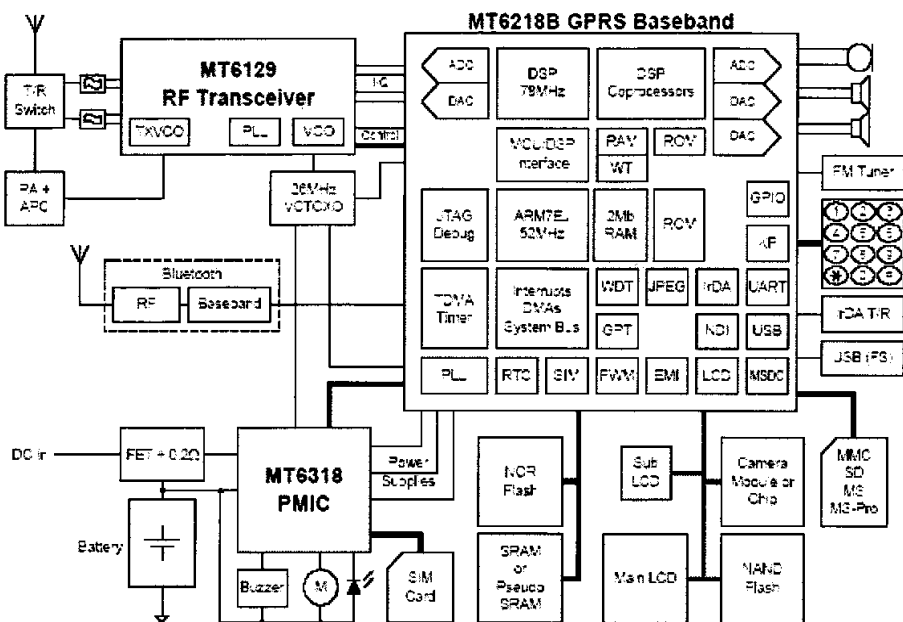


图 3.1 手机硬件架构图

3.2.1 基带芯片 (MT6219) 特性

MT6219 是双内核结构，不仅支持 GPRS Class 12 modem，同时提供全面、先进的多媒体解决方案。MT6219 集成了 ARM7EJ-S 内核和 DSP 内核。ARM7EJ-S 是主处理器，主要运行 GSM/GPRS 上层协议软件、多媒体应用、外设驱动程序等。DSP 处理低层的调制解调以及复杂的语音功能。除了一些混合信号电路，其他 MT6219 的内置电路或者和微控制器或者和数字信号处理器相联。MT6219 包括以下子系统：

- 微控制器子系统 (MCU)：包括 ARM7EJ-S 处理器、内存管理单元和中断控制器。
- DSP 子系统：包括 DSP、内存、内存管理单元和中断控制器。
- MCU/DSP 接口：MCU 和 DSP 交换软硬件信息。
- 微控制器外设：所有的用户接口模块和 RF 接口模块。
- 微控制器的协处理器：代替微控制器执行计算密集的操作。
- DSP 外设：GSM/GPRS 信道编解码加速硬件。
- 多媒体子系统：集成了几个先进的硬件加速器来支持多媒体应用。
- 语音前端：语音的模数、数模转换数据通道。
- 基带前端：RF 模块的数模、模数转换数据通路。
- 时序发生器：产生和 TDMA 帧时序相关的控制信号。
- 电源、复位和时钟子系统—管理电源，复位和时钟分配。

3.2.2 射频 (RF) 芯片特性

在本项目中所使用的射频芯片是 MT612X。MT612X 是为 GSM 和 GPRS 的多频带全球系统而设计的具有很高集成度的收发 IC。MT612X 包括四个 LNAs (低噪声放大器)。两个 RF 积分合成器, 一个集成的信道滤波器, 可编程的增益放大器 (PGA), 接收机的 IQ 解调器, 用于发射机的带锁相环的高精度 IQ 调制, 两个内部的 TX VCOs, 一个片上调整的 VCXO, 一个完全可编程的 s-d 部分-N 带片上 RF VCO 合成器。MT612X 也包括控制电路来支持不同的操作模块。下面对其三个主要部分进行介绍:

(1) 接收机: MT612X 的接收机包括四频段的微分输入低噪声放大器, 射频积分合成器, 一个片上信道滤波器, 可编程的增益放大器和一个低通滤波器。非常低的 1F MT612X 使用 image-rejection 和滤波器来消除干扰。使用准确的 RF 积分信号产生器和合成匹配技术, MT612X 的 image-rejection 在所有的频段都能达到 35dB。高集成度的信道滤波能够抵抗干扰, 块信号和没有其他任何成分的图像。同直接合成的接收机相比, MT612X 的超低 1F 结构能够提供块抑制, 调幅抑制和邻道干扰的抵消。此外, 超低 1F 结构消除了直接合成的接收机必须有的精确的 DC 缓冲标准。而且, 和生表面波中输入的普通模式平衡需求是相关的。MT612X 不需要任何其他频率合成成分就能提供模拟的 IQ 基带输出。

MT612X 包括四个差分的低噪声放大器用于 GSM850 (869MHz-893MHz) E-GSM (925MHz-960MHz), DCS1800 (1805MHz-1880MHz) 和 PCS1900 (1930MHz-1990MHz)。差分的输入通过 LC 网络和 200 Ω 的声表面滤波器相匹配。低噪声放大器的增益能够控制在附加的 35dB 的动态范围内。其特性如下:

- 具有非常低的 1F 结构 (1F 指二次变频时多出来的 VCO)
- 四频段的微分输入低噪声放大器
- 积分射频合成器
- 高度集成的信道滤波
- 大于 100dB 的增益
- 大于 110dB 的可控范围

(2) 发射机: MT612X 的发射机由两个片上的 TX VCO, 缓冲放大器, 下行链路的合成器, 四项的调制器, 一个模拟的相位检测和一个数字的相频检测组成, 每一个部分有一个 charge pump 输出和一个片上环路滤波。分路器和环路滤波器用来从下行链路合成器和四项调制器中得到理想的 1F 频率。对于一个给定的传输信道来讲, 发射机要从两个不同的 TX 号码中选择一个。这些成分同内部的压控振荡器和环路滤波实现转化的环路调制。TX VCO 适用于功率放大器。外部实现的一个控制环用来控制功率放大器的输出功率值。其特性如下:

- 精确的 IQ 调制
- 平移环结构
- 高度集成的宽度 TX VCO
- 高度集成的 TX 环滤波

(3) 频率合成器: MT612X 包括一个含有高集成度的 RF 压控振荡器的频率合成器来产生 RX 和 TX 的本地振荡频率。锁相环把 RF 压控振荡器的频率锁定在 26MHz。为了减少由部分-N 合成器引起的固有的影响, 带有抖动功能的 3 阶 $\delta - \dot{0}$ 调制来产生分频数 N。

为了减少捕获时间或者说为了多时隙的数据服务(比如 GPRS)快速设定时间, 在合成器中使用了带有快速捕获系统的数字环(振荡环)。一旦合成器设定后, 射频压控振荡器通过数字校准环设在理想频率的附近。校准后, 在一定时期内, 利用快速捕获系统来使锁定更准确。一旦捕获完成, 锁相环就会回到通常的状态。其特性如下:

- 单集成, 完全可编程的部分-N 合成器
- 高度集成的宽带 RF VCO
- 为多时隙 GPRS 应用快速准确的设置时间

3.2.3 电源管理芯片 (MT6305) 特性

MT6305 是为 GSM 手机, 尤其是为 MT620X 系统方案的 GSM 手机所优化的电源管理芯片。它包含 7 个 LDO (Low Drop Out), 每一个用于管理 GSM 子模块的电源, MT6305 的优化设计使电池具有最长的生命期, 在待机状态仅仅有 107 μ A 的低电流, 在工作状态也只有 187 μ A。MT6305 支持锂电和镍锌电池。MT6305 包含了 3 个开漏输出转换为 LED、alerter 和 vibrator。SIM 接口为 SIM 卡和微处理器提供了水平转换。MT6305 有 48 个管脚, 操作温度范围是 -25 摄氏度到 +85 摄氏度。其典型特点如下:

- 管理 GSM 基带芯片的所有电源
- 2.8V 到 5.5V 的输入电压
- 最高 15V 充电电压
- 为 GSM 子系统所优化的 7 个 LDOs
- 高操作效率和低待机电流
- 锂电和镍锌电充电功能
- SIM 卡接口
- 三个开漏输出控制 LED, Alerter 和 Vibrator
- 高温过载保护
- 高电压锁保护

- 低电压保护
- 复位和开启计时器
- 48 管脚封装

3.2.4 系统硬件工作原理

手机硬件的三个部分：逻辑系统、射频系统、电源系统，在逻辑控制系统统一指挥下，相互配合，完成手机的各项功能。硬件模块基带芯片 MT6219 主要用于实现 GSM/GPRS 的链路层和网络层协议控制 RF 模组和 BSS 通信，同时运行其它软件实现手机的附加功能，例如：显示、拍照、播放 MP3 等；RF 模块主要完成 GSM/GPRS 协议栈的物理层功能、实现突发脉冲的发射接收、射频调制解调功能；电源管理芯片 MT6305 主要负责电源管理和给系统提供各种稳定的电压；人机接口设备：键盘、液晶 LCD 用于输入、输出显示；camera 模组提供照相和动画录像等多媒体功能；NOR flash 芯片用于存储整个手机软件和用户数据；NAND flash 用做 USB 存储器^{[5][6][7]}。

3.3 系统软件

手机软件包括运行软件和调试软件。整个手机运行软件主要由 GSM 协议栈、Nucleus 操作系统、设备驱动、MMI 和 WAP 模块组成。调试软件包括 Catcher 和 META，它们的作用在于监视软件的实时运行，加速手机运行软件的开发^{[8][9]}。整个软件的系统架构如图 3.2 所示：

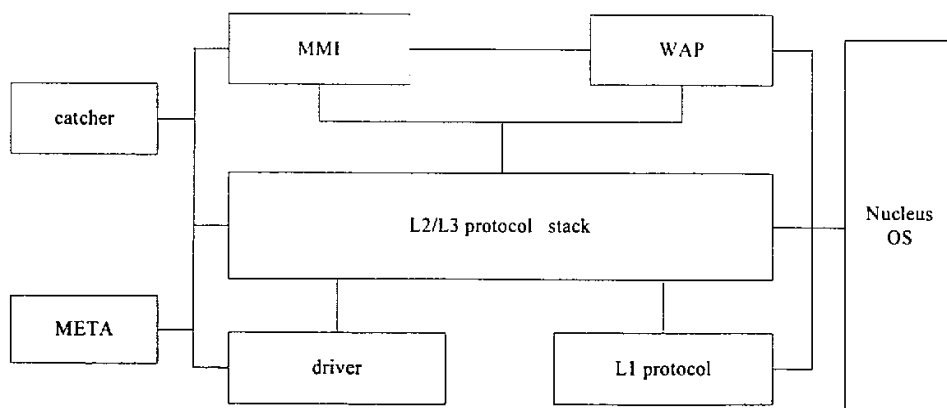


图 3.2 软件的系统结构框图

- Nucleus OS: Nucleus 实时嵌入式操作系统。
- L1 协议栈: GSM/GPRS 物理层协议。
- Drivers: 整个系统外围设备的驱动程序如: SIM,LCD,URAT,GPIO 等等。
- L2/L3 协议栈: GSM 协议栈链路层/网络层。

- MMI: 人机界面, 主要负责人机交互。
- WAP: 无线应用协议。

3.3.1 Nucleus 操作系统

由于手机系统面对的任务较复杂, 且对实时性要求很高, 所以必须有一个操作系统对各种复杂的任务根据一定的准则进行调度, 合理的安排 CPU 的时间, 使系统能够安全协调的运行。在本项目的系统中采用 Nucleus 操作系统。Nucleus 是一个实时的、多任务抢占式的嵌入式操作系统内核。采用嵌入式操作系统目的是降低软件开发的难度, 把整个软件划分为若干 task, 通过 OS 进行调度管理。OS 的作用主要表现在任务调度和向上层软件提供系统服务, 例如: 消息传递、信号量、时钟管理、中断管理等。本系统设计了一个操作系统适配层 (KAL) 使得上层的软件独立于 Nucleus 操作系统。

3.3.2 L1 协议栈

物理层或 Layer1 依照 GSM05 系列技术规范, 支持在无线媒质上传输比特流。它向上层提供服务、控制物理信道和逻辑信道之间的映射, 并且控制时序的安排, 同时执行 TDMA 的取帧和无线控制。

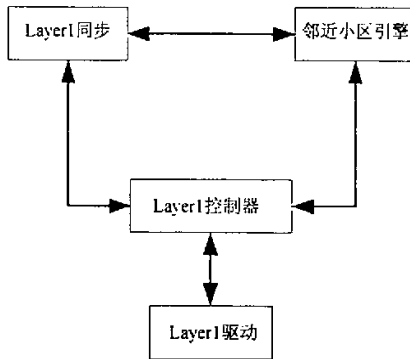


图 3.3 L1 层系统结构

L1 层各模块描述如表 3.1 所示:

表 3.1 L1 层各模块功能描述

模块名称	模块描述
L1 层异步控制	处理来自上层请求的消息, 同时把 L1 层产生的结果传向上层。
Surrounding cell engine	测试相邻蜂窝小区的功率和获取相邻蜂窝小区的信息。
Layer1 层控制	处理 TDMA 在无线环境中的时序安排, 包括来自服务基站的预定时间和公控命令。
Layer 1 driver	处理 DSP 和无线管理。

3.3.3 L2/L3/L4 协议栈

这部分包括许多 GSM/GPRS 所要求必备的条件，并为应用提供了一个强大的 GSM/GPRS 平台。这个软件平台易于通过 AT 命令实现远程 PC 控制。详细系统结构如图 3.4 所示。

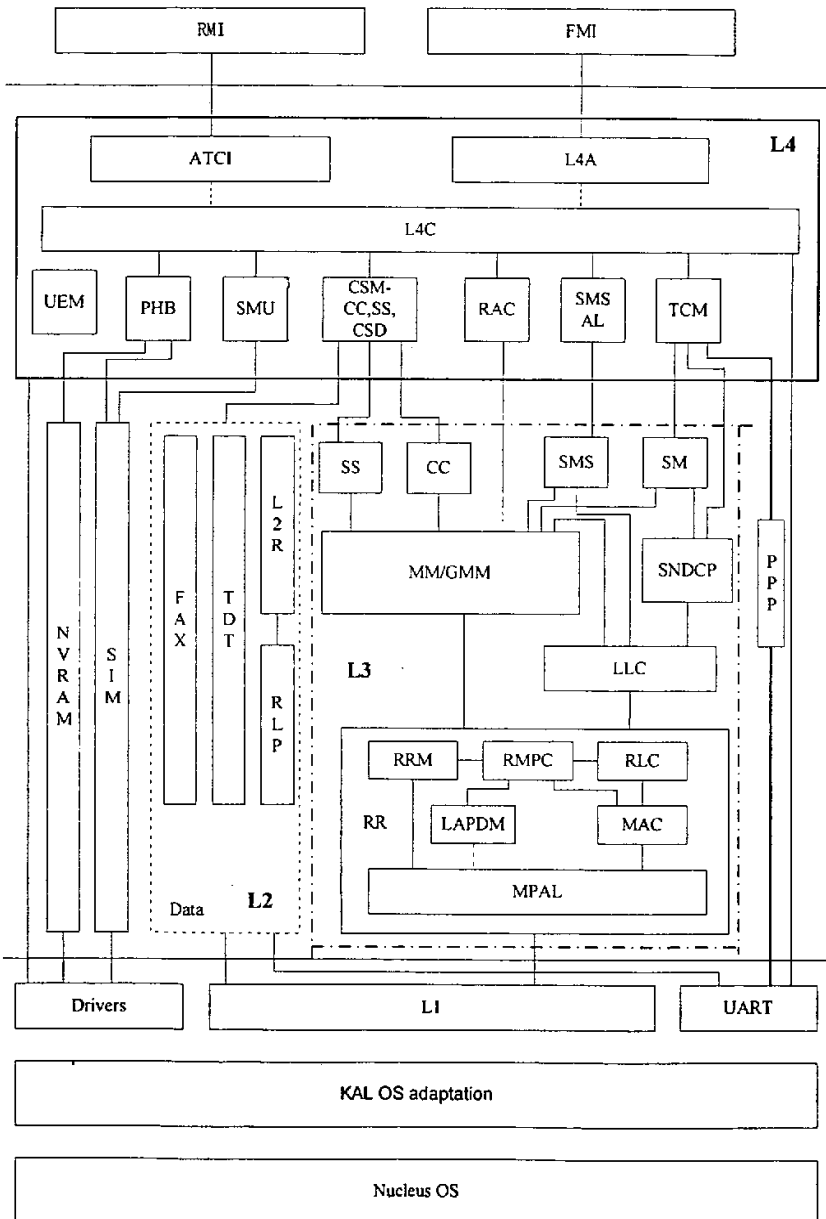


图 3.4 协议栈体系结构

在图 3.4 中, L2 是数据链路层; L3 是网络层; L4 是底层驱动和协议栈的封装层, 也是和 MMI 通信的接口层, MMI 通过 L4 和底层、协议栈进行信息交互, 完成上层应用的各个功能。图中各个节点的功能描述如表 3.2 所示:

表 3.2 协议栈各模块功能描述

模块名称	模块/子模块描述
MMI	<p>人机界面, 包括两个部分:</p> <ul style="list-style-type: none"> • RMI: 远程 MMI, 例如, PC 端。通过 UART 口利用 AT 命令与协议栈通信。 • MMI: 包括 MMI 的所有特性。
L4	<p>L4 是一个 TASK, 通过它 FMI 和 RMI 与协议栈进行通信, 一般将 L4 以下的部分称为下层, 上面的称为上层或者应用层。L4 task 包括以下的几个子模块:</p> <ul style="list-style-type: none"> • ATCI: AT Command 解释器。解释来自 PC 端的 AT 命令同时命令。L4C 执行相应的动作。 • L4A: MMI 的 L4 适配层, MMI 和 L4A 之间的服务节点是最基本的节点。 • L4C: L4 的控制实体。处理所有应用的请求和响应。 • UEM: 用户设备适配。驱动的适配, 如 KEYPAD, LED, AUDIO 等。 • PHB: 电话本管理。处理电话本的相关操作, 如存储等。 • SMU: SIM 管理单元。处理安全管理和 STK。 • CSM: 电路交换协议栈管理。 • RAC: 注册接入控制。 • SMSAL: 短消息服务应用层。 • TMC: 终端文本管理。
NVRAM	非易失性随机存储器。这是 flash 内存的适配层。
SIM	用户识别模块。如 ETSI 11.1 中描述的来处理 SIM。
DATA	<p>电路交换数据服务。数据 task 包括以下的几个子模块:</p> <ul style="list-style-type: none"> • FAX: 三组传真。 • TDT: 透明传输的电路交换数据 • L2R: 非透明传输的电路交换数据的 L2 中继协议。 • RLP: 非透明传输的电路交换数据的无线链路协议。
CC	电路交换呼叫控制。
SS	增加服务
SMS	短消息服务
SM	会议管理

MM/GMM	移动管理/GPRS 移动管理
SNDCP	子网依靠转换协议
LLC	逻辑链路控制
RR	无线资源管理。RR task 包括的子模块： <ul style="list-style-type: none"> • RRM：处理蜂窝选择和 PLMN 选择。 • RMPC：处理空闲状态或者检测特定的状态包括周围小区的配置和测量报告。 • LAPDM：处理 GSM L2 中定义的情况。 • RLC：无线链路控制协议。 • MAC：媒质访问控制协议 • MPAI：无线资源和 L1A 的适配层。
PPP	点对点协议层。这个是客户 PPP 层。

3.3.4 设备驱动

设备驱动主要用来解释来自 MCU 的命令，实现对外设的控制。这个模块处理许多用户可观察的手机行为，例如键盘处理、LCD 显示等。该模块通过函数调用或与 L4 实体进行消息交互的方式被引用，通过访问设备寄存器来提供必要的控制功能。目前，手机系统所包括的外围设备如图 3.5 所示：

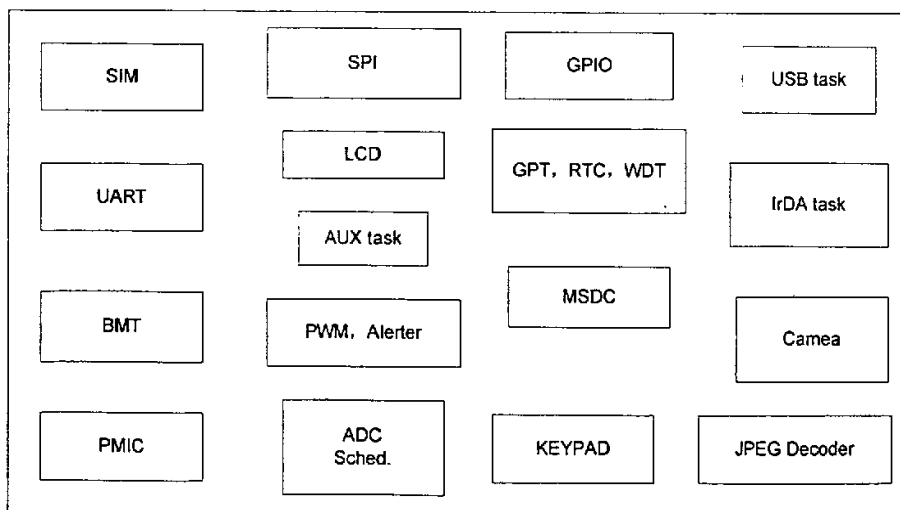


图 3.5 MCU 外围设备

表 3.3 描述了各个外设的基本功能:

表 3.3 MCU 外围设备模块功能描述

模块名称	模块描述
SIM	用户识别模块, 写入用户信息
UART	通用异步收发设备, 用于 MCU 和外设之间的通信
SPI	串行接口
LCD	液晶显示器
GPIO	通用输入输出接口, 用于控制外设
RTC	实时时钟, 用于提供时间
WDT	看门狗, 检测 MCU 的运行情况
GPT	通用定时器
PWM	脉冲宽度调制
Alerter	报警设备
Keypad	该模块处理所有从键盘发送来的信息, 处理后并把按键信息发给上层
PMIC	电源管理芯片
BMT	电池充/放电管理
AUX task	耳机 task
USB task	USB1.1 协议和驱动
IRDA task	IRDA 和驱动
MSDC	内存卡驱动, 支持 SD,MMC 卡和内存片
Camera	照相机
JPEG Decoder	控制硬件 JPEG 解码器的软件

3.3.5 MMI 模块

MMI 模块即人机界面模块, 主要负责人机界面的显示、屏幕流的控制以及和 L4 层进行通信完成人机交互。MMI 模块主要由三部分组成: Application (应用层软件), Framework (框架) 和 GUI (图像用户接口)。MMI 层的应用软件包括: 电话本, 消息 (短消息, 增强型短消息 4.0, CB), 呼叫历史, 设置 (电话设置, 网络设置, 安全设置), 情景模式等。Framework (框架) 是 MMI 模块的核心, 提供 MMI 层的事件处理机制, 屏幕流管理, 以及和其他模块的消息交互。GUI 主要负责人机界面元素, 例如: 标题, 按钮, 图片等的组织和调用 LCD 驱动显示人机界面。它们之间的关系如图 3.6 所示:

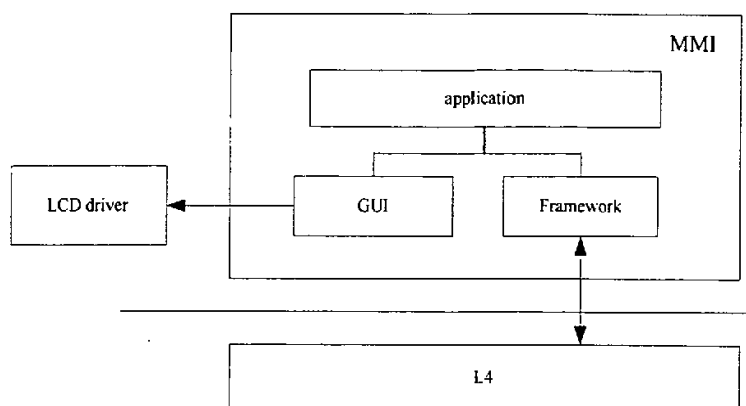


图 3.6 MMI 架构图

3.4 本章小结

这一章主要是对本项目所开发的 GSM 手机体系结构进行了总体描述。首先整体介绍了手机系统的硬件结构和软件结构。因为本论文工作主要是软件部分，所以重点讲解了手机软件运行的基础——Nucleus 操作系统以及手机体系软件部分的协议和驱动。软硬件结构概述是以两个结构框图为准，操作系统是按其功能特性介绍的，协议部分则按四层结构做了分层简介，驱动是按模块功能介绍，应用层主要介绍了 MMI 模块。本章是后面工作的基础。有了这些手机架构的基本了解，才可以进一步完成自己的设计重点——人机交互软件和软件仿真平台。

接下来的一章将是对人机交互软件的设计与实现。

4 人机交互软件的设计与实现

4.1 人机交互软件简介

人机交互软件 MMI (Man Machine Interface) 是整个系统软件体系结构的最上层, 主要实现人机交互的功能, 即解释用户的各种命令并发消息到相应的处理模块等待系统控制及处理, 同时将系统接收到的各种消息或系统所处的状态以友好的方式反馈给用户。MMI 可以看作是手机系统功能级的调度管理中心, 对手机的各个模块和部件的调度管理是通过发送相应的消息来实现的^[11]。

MMI 的主要功能有:

- 输入用户操作信息;
- 以声、光、振动等手段显示手机及网络的状态、输出操作结果;
- 实现 GSM 相关协议, 如 DRAFT pr ETS 300 906 移动台特性, ETS300 907 移动台 MMI 等标准对 MMI 的各项要求;
- 提供合理、方便的操作方法;
- 对众多的功能进行分类、组织, 以方便用户使用;
- 增加计算器等附加功能, 提高手机的实用性;
- 加入动画、图标、问候语等修饰功能, 使产品界面美观、友好。

4.2 MMI 软件设计思想

根据 GSM 手机 MMI 软件开发的具体情况, 采用面向对象的程序设计技术, 用 C 语言以注册事件和消息驱动机制来组织软件结构, 可以满足对 MMI 软件开发工作的各项要求。并将整个结构分为三个层次设计开发, 这样组织软件使得程序结构清晰, 代码效率高, 模块可再用性好, 便于增加产品功能, 便于开发小组编程时分工合作。

在这个技术方案中, Framework 是整个 MMI 软件的核心结构。Framework 为对话框、菜单、编辑及其它用户输入输出操作提供了一个基本框架, 为消息处理提供了基本手段。由于使用了面向对象的程序设计方法, MMI 软件采用消息驱动机制。Framework 收集所有的输入信息, 然后把这些输入信息以消息的形式送往合适的窗口。这些消息的执行结果又可能产生新的消息, 新消息的传送仍由 Framework 负责。Framework 通过调用窗口函数, 向窗口函数传递消息参数, 把消息送到目的窗口。

4.3 MMI 工作原理

在 MMI 应用内部, 按照功能可以划分为不同的子模块, 这些子模块由 MMI 主控模

块管理。MMI 主控模块负责各个子模块的初始化和调度，以及在 MMI 应用内部的消息寻迹。各个子模块对应不同手机功能的人机界面，通过调用底层的服务模块完成特定的功能。

MMI 的基本结构如图 4.1 所示，其中，应用层是用户定义的所有的应用，如短消息应用、MP3、呼叫管理、话机设置等。framework 层是对消息收发管理和事件的处理。UI 层管理与用户界面相关的所有功能。可以这样理解：如果涉及到功能性的实现则通过 framework 进行调度和管理，而涉及到用户界面的显示需要通过 UI 层来实现。

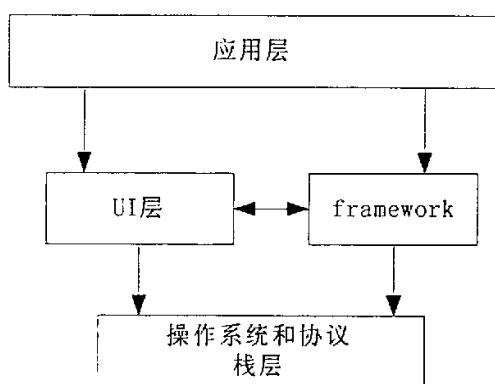


图 4.1 MMI 的基本结构框图

4.4 MMI Task

如 3.3.1 节中所介绍，该系统是一个多任务、抢占式的嵌入式操作系统。MMI 软件在系统中作为一个任务在运行，统一服从操作系统的管理。MMI Task 的结构如图 4.2 所示，它是一个多层的结构，可以分为 Framework、UI 和应用层。下面将详细介绍各层的主要功能及设计思想。

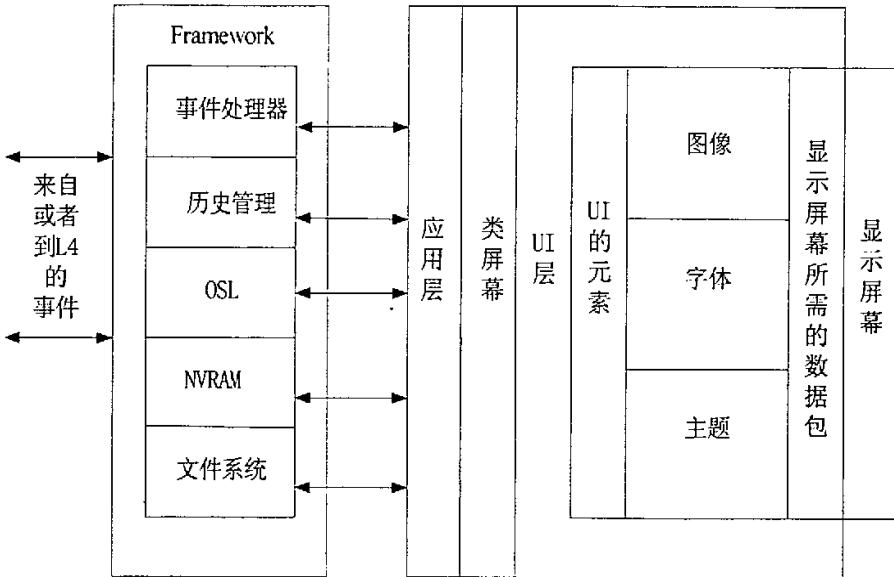


图 4.2 MMI Task 的结构

4.4.1 Framework 层

Framework 用于接收来自所有 Tasks 的事件，帮助应用层控制显示屏幕的流程，且为系统提供调用应用时所需的数据包，例如消息分发时写入到队列中的数据包。

Framework 的元素：

- 事件处理器：为各种事件注册和执行应用中的回调函数。
事件的分类：
 - A. 协议栈事件；
 - B. Highlight 事件；
 - C. 按键事件；
 - D. 计时器事件。
- 历史管理：帮助应用保留屏幕流程和存储屏幕数据。
- OSL：根据应用的命令向所有的操作系统提供数据包。
- NVRAM：保存手机中非易失的数据。
- 文件系统：存储数据并且从中恢复数据。

(1) 事件处理器：为应用提供事件处理的 API，在运行过程中管理事件的处理。包括：

① 键盘事件处理 API：主要被用于应用和类函数，如为特殊的按键建立按键处理、为一组按键建立按键事件、为按键事件执行当前的按键处理、为特殊的按键清除按键处

理、为所有的按键清除按键处理、启动和结束键的专门处理等。

② 协议事件处理 API：主要被用于应用，如建立协议事件处理、执行当前的协议事件处理、清除专门的协议事件、清除所有的协议事件处理等。

(2) 历史管理：历史节点的结构包括：

- . Screen ID：一个屏幕的 ID
- . 进入函数的指针：重新绘制一个屏幕
- . 输入缓冲器：存储这个屏幕上运行的文本数据
- . GUI 缓冲器：存储这个屏幕上与 UI 相关的信息

历史的 API 包括：

- . 添加节点到历史中
- . 从历史中删除第 N 个节点
- . 在历史中回到第 N 个节点
- . 恢复一个屏幕的历史
- . 恢复屏幕的输入缓冲器
- . 恢复屏幕的 UI 缓冲器
- . 在运行的过程中推出历史
- . 初始化历史

(3) OSL 层：该层主要有两个功能，一个是创建消息、发送消息和读取消息；另一个是创建计时器、开始和停止计时器。调用该层的消息 API 函数可以实现 MMI Task 和其它 Task 之间的信息交换，调用该层的计时器 API 函数可以实现 MMI Task 中所有计时器的功能。

(4) NVRAM（非易失性随机存储器）：NVRAM 保存着手机中非易失性的数据，手机中所有参数的设置值都在 NVRAM 中保存着，其大小为 2Mbytes，数据项的管理方式像文件系统管理文件一样。

MMI 在 NVRAM 中保存的数据类型如下：

Byte Data	1 byte
Short Data	2 bytes
Double Data	8 bytes
Application specific Record	Max 2K bytes

4.4.2 UI 层

UI 层主要的功能是给手机所有的屏幕提供数据，它有很多的种类屏幕组成，而每个类屏幕又有很多 UI 元素组成。

(1) 类屏幕：类屏幕层包含了一组定义用户界面的函数。每个类屏幕的结构都是相

近的，主要由以下函数组成：

- 显示一个类屏幕的函数；
 - 退出一个类屏幕的函数，当类屏幕不再需要显示时调用该函数；
 - 得到一个类屏幕历史数据的函数，得到类屏幕的状态信息，这个状态信息能被存储，能被恢复，可以通过调用显示函数来返回到它以前的状态；
 - 得到历史的大小函数，该函数得到存储屏幕的历史数据所需要的空间的比特数。
- (2) UI 元素：用于建立用户界面，它们是利用数据结构和函数实现的。

UI 元素的数据结构：

- 包括所有的 UI 需要显示的信息
- 包括与绘图相关的部分
- 绘图信息包括坐标，尺度和与主题相关的信息

一个简单的按钮 UI 元素：

```
button
{ INT      x, y;          /*位置*/
  INT      width,height; /*尺度*/
  STRING   text;        /*标志*/
};
```

4.4.3 应用层

应用层包含着手机提供给用户的各种功能，每个功能自己组成一个模块，可以根据具体的需要增加或删除子模块。该层是独立于 Framework 和 UI 层的，这样极大的增强了软件的可扩展性。应用层包括以下子模块：

- (1) 电话本
- (2) 信息 (SMS, EMS4.0, CB)
- (3) 呼叫记录
- (4) 设置 (话机设置, 通话设置, 安全设置)
- (5) 情景模式
- (6) 娱乐与游戏
- (7) 工具箱
- (8) 服务
- (9) 快捷功能
- (10) 呼叫管理
- (11) 中断事件, 如充电, 闹钟

4.5 MMI 与协议栈之间的接口设计

MMI 与协议栈的接口如图 4.3 所示，MMI 通过此接口和协议栈进行消息的发送和接收，完成信息的交互，实现和协议栈相关的功能。下面将详细说明通信方式、通信的数据结构和通信的全过程。

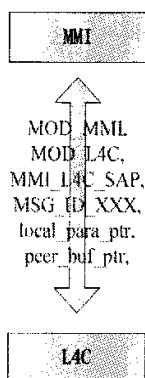


图 4.3 MMI 与协议的接口

- 通信方式：MMI 和协议栈（L4C）各自有一个外部消息队列，通过此消息队列来发送或者接收消息。

- 通信过程中传送的数据结构如下：

```
typedef struct ilm_struct {
    oslModuleType   oslSrcId;   // 源模块 ID
    oslModuleType   oslDestId;  // 接收模块 ID
    oslMsgType      oslSapId;   // 服务接入点
    oslMsgType      oslMsgId;   // 消息 ID
    oslParaType     *oslDataPtr; // 本地参数缓冲器
    oslPeerParaPtr  *oslPeerBuffPtr; // peer 缓冲器指针
}ilm_struct;
```

- 从 MMI 队列中读取一个消息：从 Task 的创造到进入一个消息的循环。接口函数为：OslReceiveMsgExtQ (mmi_qid, &mmi_message)；

- 向 L4C 发送一个消息，分下面四个步骤完成：

第一步：建立一个本地参数的缓冲器。

第二步：分配需要的值到本地参数缓冲器。

第三步：在 ilm_struct 中填写信息。

第四步：发送消息到 L4C 模块。

- 从 L4C 中接收到一个消息：注册一个消息响应的回调函数：

SetProtocolEventHandler(回调函数, msg_id), 在得到这个消息时根据消息的 ID(msg_id) 调用相应的处理函数也就是回调函数。

4.6 MMI Task 与其它 Task 的信息交互

由上面的章节我们知道，一套软件系统的运行是需要很多 Task 来完成的，而 MMI 作为一个 Task 在系统中运行，需要和其它 Task 不停的交互信息才能完成其所有的功能。下面将详细说明 MMI Task 和其它 Task 的信息交互过程以及本软件中 MMI Task 的运行机制。

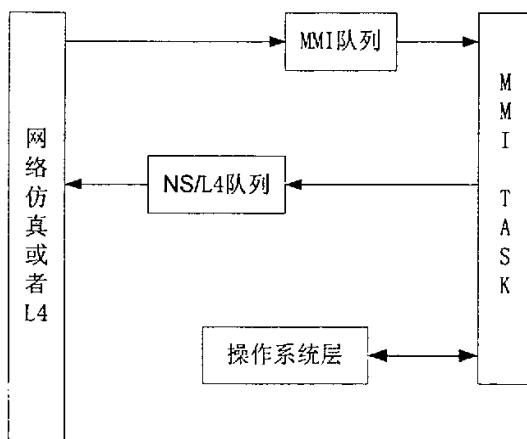


图 4.4 Task 交互的简单说明

Task 交互的简单说明如图 4.4 所示，由图可以看到：协议栈或者 L4 将事件消息写入到 MMI 队列中， MMI Task 从这个队列中读取事件消息。然后 MMI Task 向 L4/NS 队列中写入 MMI 事件消息， L4 Task 或者是网络仿真器从这个队列中读取事件消息。

下面举例说明 Tasks 之间的事件流程：

例一：按键事件（底层向 MMI 发送消息）

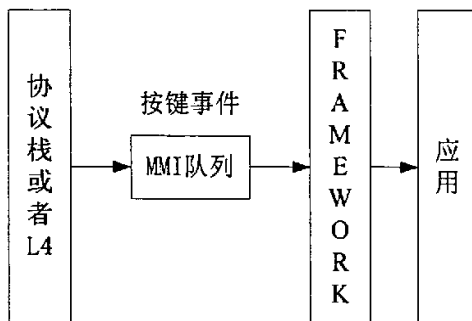


图 4.5 按键事件

- 协议栈向 MMI 队列中写入这个按键事件
- MMI Task 从 MMI 队列中读取该事件，包括以下两个过程：
 - Framework 调用应用函数在应用层中注册该事件
 - 应用层接收到这个按键事件并且做出响应

例二：应用请求底层播放声音（MMI 向底层发送消息）

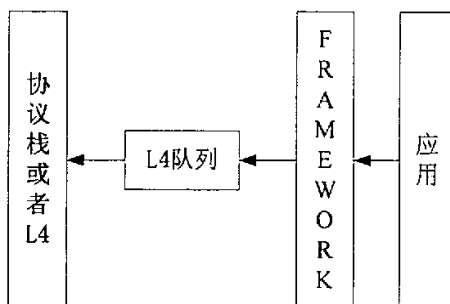


图 4.6 请求事件

- 应用传呼 Framework 层传送消息到 L4 队列中
- Framework 向 L4 写入消息
- L4 Task 从 L4 队列中读取事件并且播放请求的声音。

通过以上 MMI Task 和其它 Task 的消息交互过程，可以得出 MMI Task 的运行机制：

- (1) 等待来自 MMI 队列中的消息（这个队列中的消息是协议栈写入的）。
- (2) framework 层处理消息事件。
- (3) framework 层触发应用层中的回调函数。
- (4) 应用层利用 UI 层的类屏幕函数显示屏幕。

4.7 本章小结

本章详细描述了 MMI 软件的设计与实现。首先对 MMI 软件做了简单介绍，接着描述了其设计思想和工作原理，然后对 MMI Task 的三层结构（Framework 层、UI 层和应用层）作了详细的说明。之后是 MMI 与协议栈之间的接口设计，最后，论文重点分析了 MMI Task 与其它 Task 的交互过程，以按键事件和播音事件为例，讲解了实现过程及特点，并由此得出 MMI Task 的运行机制。

接下来一章将要讲述手机软件仿真平台的设计与实现。

5 手机软件仿真平台的设计与实现

5.1 仿真平台简介

在手机的研发过程中，用户界面部分的发展是必不可少的部分。由于直接在硬件上进行开发调试非常的费时耗力，用户界面部分属于人机交互界面，只要能够模拟出手机的屏幕显示和按钮输入等，就可以在开发界面部分时采用在 PC 机上的模拟器进行开发。界面开发模拟器的目的主要是实现手机软件界面部分的模拟，使得在模拟器中开发的代码可以复用到实际的手机上去。可以较大地节约开发成本，缩短开发时间，并对应用进行更快、更完善的测试。

5.2 仿真平台设计方案和工作原理

本项目所开发的手机软件采用 C 语言进行编写，所以软件仿真平台采用 VisualC++ 进行编写，以便实现软件代码的复用；采用 MFC 单文档视图实现程序的主框架，程序运转的内核部分（包括 BIOS，AP，FRAMEWORK，RES 资源）基本上同手机实际使用的源代码完全一致。仿真平台程序的目录结构也和手机源代码界面模块的目录结构相同。

在手机软件的架构中，界面部分是构建在 BIOS 模块（BIOS 模块主要完成对手机硬件的驱动）之上的，同时界面部分也和通信部分存在着消息传递（如呼出、挂断等）。

界面部分的代码调用 BIOS 模块提供的 API 函数来实现各种逻辑（如界面状态的转移、界面上的各种控件等），其实界面部分的代码并不知道也不关心 BIOS 提供的 API 是如何实现的，它只需要知道某个 API 实现了什么功能就可以了。因此，模拟 BIOS 模块在 PC 机上实现是仿真平台的重要任务。

界面部分还和通信部分存在信息交互，因此必须存在一个虚拟的通信部分，它可以模拟手机软件中的通信模块接收界面发出的消息并对其作出正确的回应。

手机软件是基于实时多任务操作系统的，界面任务和通信部分都是独立的任务，因此仿真平台还应实现对任务的模拟。

界面的另一个重要单元是定时器，界面的动画、各种延时等要靠定时器才能完成。综上所述，仿真平台应实现以下几方面的模拟功能：

1. 对 BIOS 模块涉及的硬件如 LCD、FLASH ROM、键盘进行模拟。
2. 对通信模块的模拟。
3. 对实时操作系统的任务机制实现模拟。
4. 实现定时器的模拟。

5.3 仿真平台总体设计

根据仿真平台的设计思想和工作原理需求，在 PC 模拟环境中，仿真平台用三个主要的线程模拟操作系统的多任务，并建立起系统的主框架。三个不同的线程分别实现了模拟器框架、L4 和服务模块、MMI 应用程序。系统总体结构如图 5.1 所示：

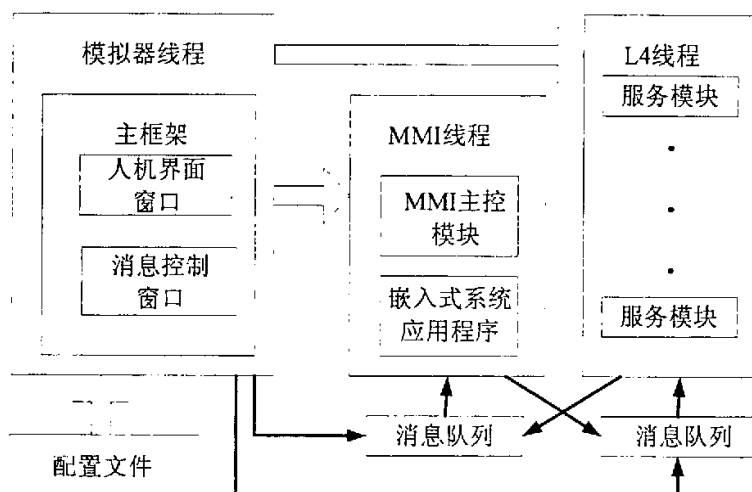


图 5.1 仿真平台体系结构

(1) 模拟器线程：该线程负责模拟环境的控制，负责把 MMI 应用作为动态链接库显示加载到模拟环境中并启动 L4 线程和 MMI 线程。该线程包含了人机界面窗口和消息控制窗口。人机界面窗口中实现了模拟的键盘输入界面和 LCD 输出界面；消息控制窗口为用户提供了发送定制消息能力。在调试程序时，用户可以用手动的方式向 MMI 线程发送特定内容的消息。

(2) L4 线程：该线程模拟了 RTOS 的系统调用和各个底层服务模块的消息接口的功能。在手机环境中，RTOS 负责接收 MMI 应用程序发往底层服务模块的消息。

(3) MMI 线程：在手机系统中，MMI 应用作为一个任务运行在系统中，在这个任务中包含了不同的应用程序。在模拟系统中，MMI 应用程序以动态链接库的方式调入模拟环境，并用一个单独的线程实现。该线程中包括 MMI 主控模块和手机系统的应用程序，其中 MMI 主控模块模拟了 RTOS 对进程的调度和消息的分发。模拟环境中的 MMI 应用程序与实际手机系统中的应用程序由同样的代码构成。每个应用程序模块包括：模块初始化函数，模块内部的消息接收和分发函数，特定消息的处理函数。

5.4 模块设计与实现

5.4.1 LCD 的模拟

在仿真环境中，用同样尺寸的位图来模拟 LCD 显示屏。在读入配置文件后，模拟器线程将创建该位图并分配一块显存。任何 MMI 应用中对显示驱动的调用将被模拟为对该显存的操作。LCD 的模拟是通过模拟向 Windows 程序的一个控件上画点来实现的。部分代码如下：

```
void lcd_fb_update (lcd_frame_update_struct *lcd_para)
{
    /* draw to dc */
    if (lcd_id == MAIN_LCD)
    {
        /* draw to simulator's main LCD region */
        hdc = GetDC (hWnd) ;
        hdcMem = CreateCompatibleDC (hdc) ;
        hbmp = CreateCompatibleBitmap (hdc, LCD_WIDTH, LCD_HEIGHT) ;
        SelectObject (hdcMem, hbmp) ;
        for (i = 0 ; i < LCD_HEIGHT; i++)
        {
            for (j = 0 ; j < LCD_WIDTH; j++)
            {
                c.r = HW_TO_RGB_R (simulator_shadow_buffer[i * WIDTH + j]) ;
                .....
                SetPixel (hdcMem, j, i, RGB (c.r, c.g, c.b)) ;
            }
        }
        BitBlt (hdc, X, Y, X + WIDTH, Y + HEIGHT, hdcMem, 0, 0, SRCCOPY) ;
        .....
    }
}
```

其中使用了 SetPixel 这个 Windows API 完成在屏幕上画点的工作。

5.4.2 FLASH ROM 的模拟

FLASH ROM 的模拟比较简单。手机采用的是 2Mbyte 的 FLASH, 在程序源代码中, 在 FlashConfig.c 文件中定义了 FLASH 的开始地址和总空间大小, 如:

```
#define FLASH_BASE_ADDRESS    0x0E00000
#define ALLOCATED_FAT_SPACE    0x200000 //0x200000 == 2Mbytes
```

在实际的手机软件中 FLASH_BASE_ADDR 是个确定的地址值, 在模拟器中却是一个指针变量, 在 flash.h 中如下:

```
#ifndef MMI_ON_WIN32
#include "define.h"
extern U8 * FLASH_BASE_ADDRESS;
```

这样, 模拟 FLASH ROM 的方案就是: 在模拟器开始运行时, 分配一块 2Mbyte 的内存, 如果存在 CUSTPACK_ROM 这个文件, 就将该文件中的内容复制到这块内存中, 否则按照默认值进行处理。模拟器运行结束后将这块 2Mbyte 的内存保存到 CUSTPACK_ROM 这个文件中。

由于 FLASH ROM 采用内存的方式模拟, 所以 BIOS 中关于 FLASH ROM 操作的函数, 如 NvramInterface.c 中的函数都直接变成了 windows 平台上操作内存的函数, 与实际手机中使用的 flash 操作函数大不相同。

5.4.3 键盘的模拟

手机对键盘的处理是采用键盘信息队列的方式, 即按下一个键, 就将键盘信息存入队列, 由键盘任务读取键盘并处理之。键盘队列本质上是个结构数组, 因此模拟按键信息就是按照规定将键盘信息放入键盘队列中即可。

仿真平台采用按钮来模拟按键, 如果用户按下了某个按钮, 程序就会将相应的按键发往按键队列当中去。在 SimPSView.cpp 中, CSimPSView::OnButtonPSKeys (UINT nID) 是按钮事件的响应函数。

5.4.4 通信模块的模拟

为了模拟在手机上 MMI 应用模块和各个底层服务模块的消息通信, 在模拟环境中为 L4 线程和 MMI 线程各分配了一个消息队列。创建过程如下:

```
for (i=0;i<TOTAL_TASKS-9;i++)
{
    task_info_g1[i+9].task_ext_qid=OslCreateMsgQ (task_create_tbl[i].task_qname,
                                                sizeof (MYQUEUE) ,task_create_tbl[i].task_ext_qsize);
}
```

这里调用了消息创建函数 `OslCreateMegQ`，第一个参数是消息队列名字，第二个参数是消息队列所占的内存空间大小，第三个参数是消息队列所能容纳的最大消息个数。利用这两个消息队列可以实现 MMI 应用模块和底层服务模块的通信，模拟器线程也通过 MMI 线程应用的消息队列向其发送模拟的系统超时消息。两个队列中的信息通过图 5.2 显示：

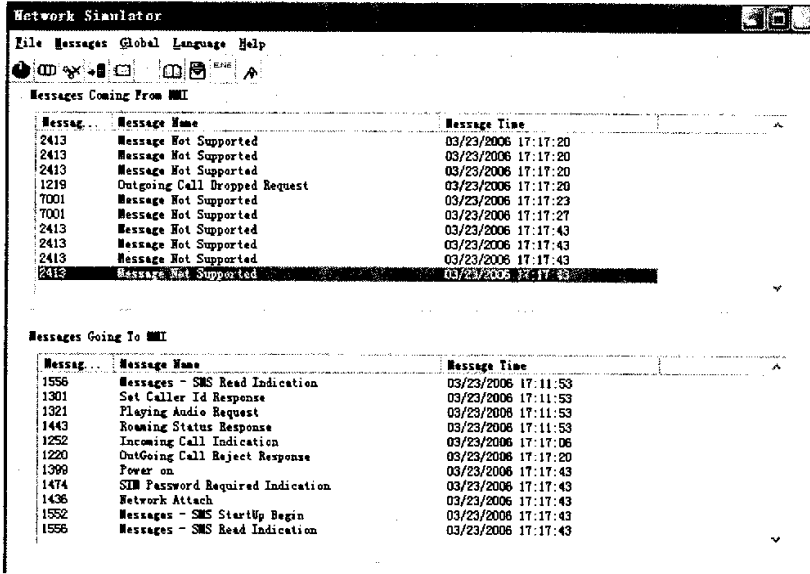


图 5.2 MMI 和 L4 通信的消息显示图

注：图 5.2 的上半部分为 MMI 发往 L4 的消息

下半部分为 L4 发往 MMI 的消息

在模拟环境的界面上将提供用户发送特定消息的功能，用户可以选择内建的各种消息类型，输入消息参数，并模拟外部环境和服务模块，将消息发往 MMI 应用程序。如图 5.3 所示：

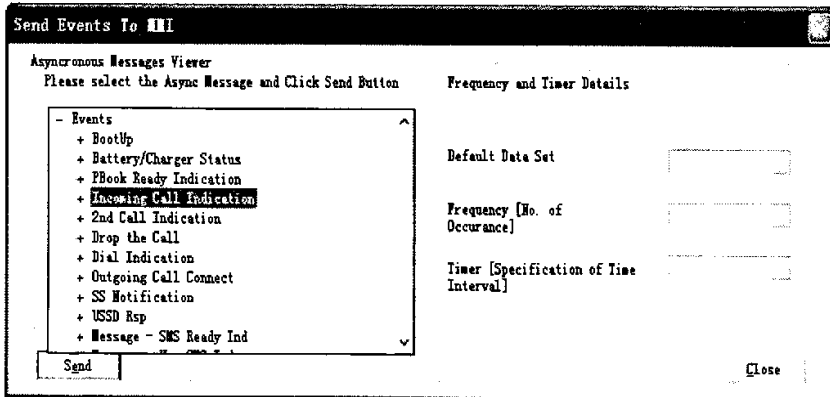


图 5.3 手动向 MMI 发送消息

5.4.5 实时操作系统的模拟

实时操作系统的模拟主要是模拟操作系统任务的运行，在 Windows 平台上，采用线程模拟操作系统任务无疑是比较好的选择。仿真平台采用了线程来模拟各种任务，仿真只需要实现界面运转和界面与通信模块的交互。因此最重要的线程为界面线程、键盘处理线程、通信模块线程，还有一个线程用于关闭模拟器的运转。创建任务时调用了 windows 的 API 函数：CreateThread（）

```
HANDLE                                WINAPI                                CreateThread
( LPSECURITY_ATTRIBUTES,DWORD,LPTHREAD_START_ROUTINE,PVOID,DWORD,PDWORD);
```

这些线程处理函数的实现和手机软件中任务的实现基本上是类似的。

用户点击运行模拟器按钮时，仿真平台启动这些线程，关闭时，通过设定线程事件来通知关闭模拟器线程进行模拟器关闭，关闭模拟器线程通过调用关机操作（即模拟长按 Power 键）来进行关闭模拟器。

5.4.6 定时器的模拟

定时器的模拟采用 Windows 的 Timer 来模拟。实现如下：

创建和删除函数：

```
oslTimerid OslIntCreateTimer (PS8 timer_name) {}
OSLSTATUS OslIntDeleteTimer (oslTimerid timerid) {}
```

设置和取消函数：

```
OSLSTATUS OslIntSetTimer (oslTimerid timerid,
                          oslTimerFuncPtr func_ptr,
                          void * funcArg,
                          U32 timeDurationMsec) {}
OSLSTATUS OslIntSetTimer (oslTimerid timerid,
                          oslTimerFuncPtr func_ptr,
                          void * funcArg,
                          U32 timeDurationMsec) {}
```

开始和停止函数：

```
OSLSTATUS OslIntStartSoftTimer (unsigned short nTimerId,
                                oslTimerFuncPtr func_ptr,
                                void * funcArg,
                                U32 timeDurationMsec) {}
```


OSLSTATUS OslIntStopSoftTimer (unsigned short nTimerId) {}

主要有 Set、Start、Stop 三种操作。Set 是设置一个定时器，也就是申请一个定时器资源，Start 和 Stop 是开始和停止一个定时器。

5.5 仿真平台上层的实现

前面 5.4 节分别实现了运行上层 MMI 线程（界面部分）所需的硬件和软件，下面接着描述上层界面部分的工作原理及设计思想。

5.5.1 仿真平台上层概述

仿真平台的上层工作包括以下两个部分：

- (1) NWS 和 PMS 之间的信息交互，仿真 MMI 之间以及协议栈和 MMI 之间的交互；
- (2) 在一个数据文件中保存数据，仿真数据在 flash 上的存贮。

仿真平台上层框架如图 5.4 所示：

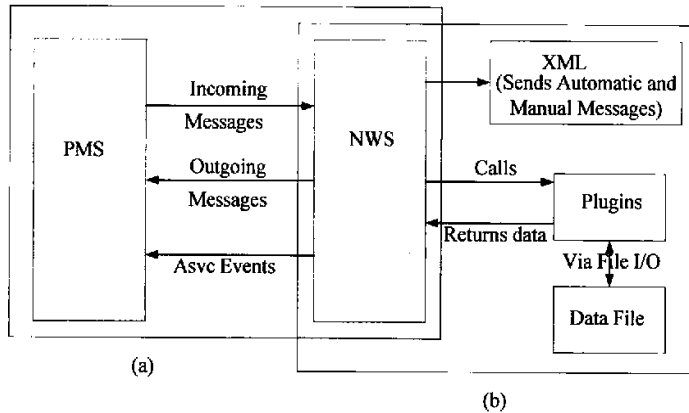


图 5.4 仿真平台上层框架图

注：PMS: Pixtel MMI Simulator

NWS: Network Simulator

XML: Extensible Markup Language

NWS 有下面 3 种方式和 PWS 进行信息交互

- 接受从 PMS 到 NWS 的 incoming messages，例如：一个呼叫请求。
- 发送从 NWS 到 PMS 的 outgoing message，此消息作为一个 incoming message 的响应消息。例如：响应一个获得电话本记录的请求消息。
- 发送从 NWS 到 PMS 的异步消息，像 Boot Up（开机），连接充电器和耳机，USSD 消息等等。

在文件中读取和存储一个应用的数据是 NWS 在收到一个请求消息后调用 plugins 实

现的，例如：PMS 需要显示电话本记录的过程如下：

- ① PMS 向 NWS 发送显示电话本的请求消息
- ② NWS 调用 plugin 读取文件获得需要显示的数据
- ③ NWS 携带数据信息向 MMI 发送响应消息
- ④ MMI 显示电话本记录给用户

仿真 plugins 也允许在电话本中添加、删除和编辑电话本记录。其它的应用像 SMS, Call history 等，以相同的方式调用 NWS 的 plugins 向 MMI 提供数据。

对 Incoming messages 的回复是自动还是手动是依据消息元素的一个标志位的设置。Plugin 是一个函数，它被 NWS 调用在 XML 中有一个 entry 去执行动态序列的消息，以仿真 NVRAM 文件的 I/O 操作。

NWS 向 PMS 发送的异步消息如下：

- Boot Up
- Battery Charger
- Hardware Charger Connected
- Phone Book ready indication
- Drop the call indication
- Dial call indication
- Outgoing call connect
- Messages (Class 0, CB message etc)
- SS notification
- USSD Reponse
- SAT Messages
- PLMN status
- AT commands
- Line1/Line2 indication
- Call Forward
- Call Barring response
- Sim detection failure indication
- NITZ messages

5.5.2 仿真平台上层的设计与实现

NWS 是一个多层的 windows 应用。NWS 如此设计是为了在添加新消息时，代码的修改量以达到最小。一个新消息的增加或改变消息中的数据结构可能不需要改变任何代码，而只需改变 XML 即可。NWS 包括以下五层结构：

- GUI Layer
- Communication Handler (Queue handler)
- Network Simulator Engine
- Message Handling Manager
- XML Manager

下面将对各层做详细介绍:

(1) GUI Layer: 该层是 NWS 提供的用户接口, 它使用户和仿真能进行信息交互, 并允许对 incoming message 的响应, 发送异步信息序列等。该层能和消息处理管理者交互以获得需要显示的消息, 并且也能传输任何用户输入的信息。

(2) Communication Handler: 它的职责是处理所有的 incoming 和 outgoing 消息, 包括两个子模块:

① Incoming message handler: 读取所有来自 PMS 的 incoming messages, 并传送给 Message Handler 处理;

② Outgoing message handler: 把 Message Handler 传过来的消息写到 PMS 队列中, PMS 中的 task 读取 PMS 队列中的信息并执行所要求的动作。

(3) Network Simulator Engine: 初始化 NWS 线程, 创建 Communication handler, 主要的 GUI 被一个独立的线程创建, 并有自己的信息队列, simulator engine 在完成各个组件的创建后退出并将控制权转向 GUI 层。

(4) Message Handling Manager: 它主要是一个控制管理, 一方面接受所有 incoming messages, 并在 XML manager 的帮助下解释这些消息, 然后在需要的情况下委派控制到 plugin handler; 另一方面解释 XML handler 的信息并为它们创建 buffer, 这些消息是发往 PMS 的, 然后将这些消息传送到 Communications handler。

(5) XML Manager: XML Manager 是网络仿真的神经中枢, 它提供所有关于 incoming 消息、outgoing 消息、异步消息、数据值和数据格式的信息。所有被仿真器执行的动作都是基于 XML 文件, XML Manager 是网络仿真的信息管理者。仿真平台序列如图 5.5 所示。

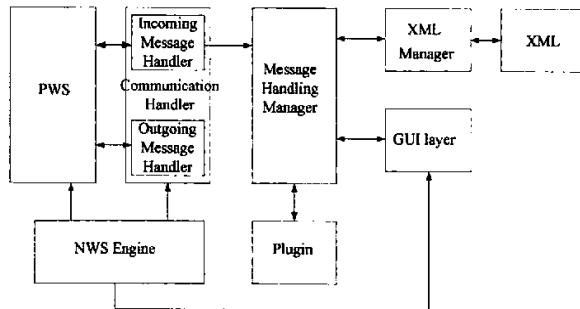


图 5.5 仿真平台序列图

(6) Incoming 消息的消息序列: Incoming 消息序列如图 5.6 所示, 这里用的是面向对象的设计思想, 各个模块分工合作, 并各自完成自己独立的工作, 这种思想体现在消息的处理过程中。所有的 Incoming 消息首先到 Incoming Message Handler 模块中, 然后再由 Message Handler Manager 处理, 处理完的消息再交到 Outgoing Message Handler 模块中发送出去。

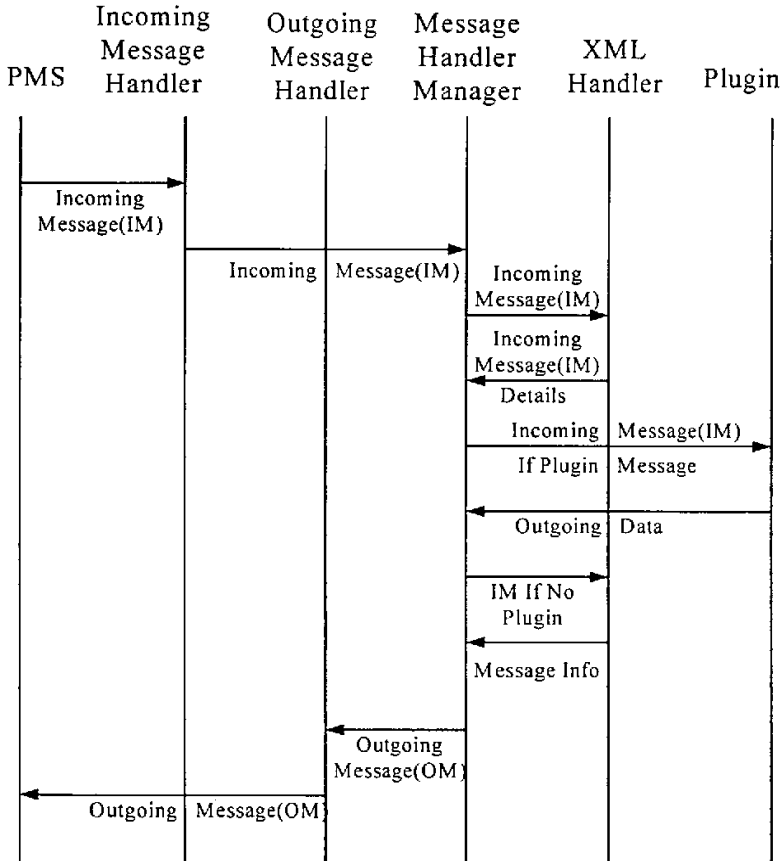


图 5.6 Incoming 消息序列图

(7) Outgoing 消息的消息序列: Outgoing 消息序列如图 5.7 所示, 这里举例说明了发送 Async Message (异步消息) 的过程 (响应消息的发送过程类似, 区别在于发起者的不同)。

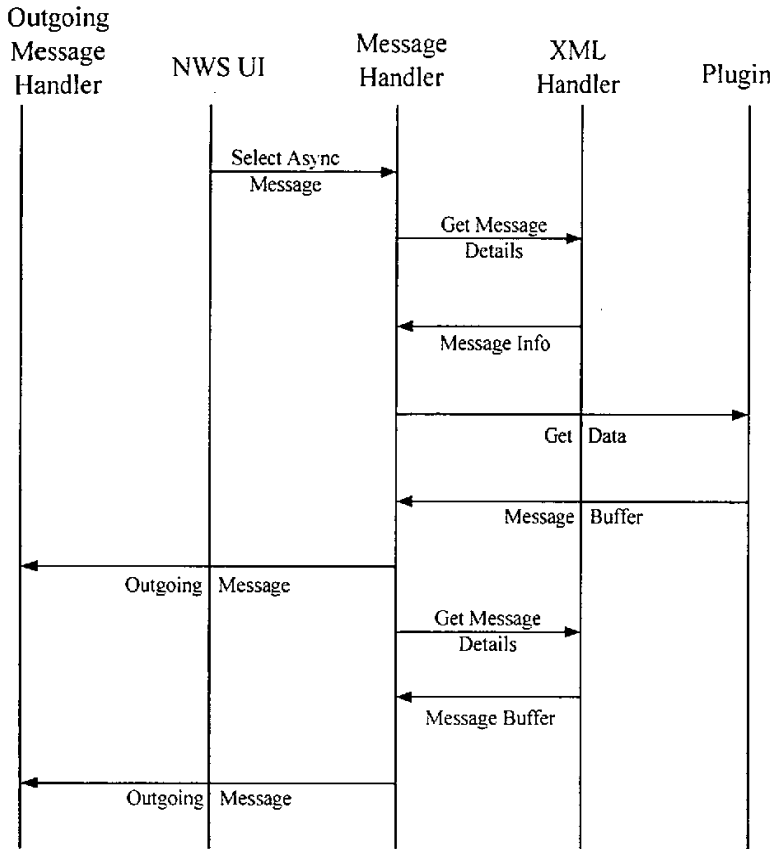


图 5.7 Outgoing 消息序列图

(8) XML 的工作机制: XML 文件“GCML.txt”包含了所有的 incoming 消息, outgoing 和异步消息, 应用, 数据类型、格式、数据列表和数据。Network simulator engine 用 XML 文件获得每个 incoming 消息的信息, 包括该消息的数据和响应消息, XML 有所有消息和消息序列的记录, 还有将被执行的动作。

① Incoming Message: Incoming 消息是从 MMI 到 NWS 的消息, 每个 incoming 消息有一个对应的响应消息 ID, 此响应消息能被自动或手动发送。这些消息 ID 有一定的数据格式 (就像 C 语言中的数据结构), 包括下面 5 个元素:

- name
- Id
- Format
- Application Id
- Outgoing message id

② Outgoing Message: Outgoing 消息是从 NWS 发往 PMS 的消息, 每一个 outgoing 消息都有一个消息 ID, 包括以下内容:

- name
- Id
- Format
- Application

③ Async Message Sequence (异步消息序列): 异步消息序列是从 NWS 发往 PMS 的消息。它包含几个 outgoing messages, 当用户选中一个异步消息序列时, 那么在这个异步消息序列里的所有 outgoing messages 将会从 NWS 发送到 PMS。

5.6 仿真平台的运行和结果

仿真平台的运行步骤如图 5.8 所示。模拟器线程启动后, 首先启动 L4 线程, 初始化各服务模块。接着, 用户可以通过菜单打开手机配置文件, 模拟器线程将读入手机配置信息。根据配置信息, 模拟器线程调入手机背景图片和该手机的 MMI 应用程序的动态链接库文件。最后启动 MMI 线程, 该线程调用 MMI 应用中的初始化函数, MMI 应用程序开始运行, 准备处理模拟的用户输入和网络信息。

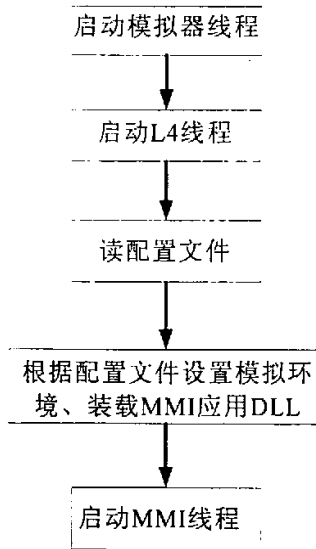


图 5.8 仿真平台运行流程

至此, 仿真平台的设计和实现已全部完成。本论文所实现的仿真平台已达到了项目开发的需求, 开发人员可以在模拟环境上对 MMI 应用程序作充分的调试和测试, 而不需要每次调式都烧录程序至手机平台。模拟器启动后如图 5.9 所示:



图 5.9 手机在 PC 上的仿真图

在图 5.9 所示的手机待机界面上，按下数字键后的仿真界面如图 5.10 的 (a) 所示，在按键界面按下发射键后的呼叫连接界面如图 5.10 中的 (b) 所示。这些仿真图和手机中实际运行的界面完全一致。



图 5.10 按键和拨号连接仿真图

5.7 本章小结

本章重点完成了对手机软件仿真平台的设计与实现。首先对仿真平台做了简单的介绍,接着根据其特点和需求提出了设计方案,依据设计方案对整体架构作了分析和设计,然后对仿真平台做了模块划分,并对各个模块分别进行了详细设计和实现,最后给出了仿真平台的运行流程和仿真结果图。

6 结论

6.1 结论

截至 2005 年底，中国手机用户近 4 亿，且每年需求量仍高达 6 千万台以上。然而，这其中国外品牌占了绝大多数。要想改变这种局面、在竞争激烈的手机市场上取得主动权，国产手机厂商就要首先掌握与市场关系最直接的 MMI 软件技术，才能把握市场的先机，并逐步从 OEM 方式过渡到拥有协议栈软件设计能力、硬件设计能力，直到拥有行业协议标准的制定能力。可见，研发自己知识产权的手机现已成为当务之急。本论文正是为实现这一长远目标所进行的一些有益探索。

论文重点对手机软件的仿真平台进行了设计实现。首先对手机系统的硬件、软件总体结构进行了分析研究，接着对 MMI 软件的设计实现进行了全面的剖析，最后成功的在 PC 机上搭建了手机软件的仿真平台。此仿真平台已达到在 PC 机上开发 MMI 应用软件的要求，并且此 MMI 源代码和手机中运行的源代码完全一致，在仿真平台上调试、测试好的程序可直接下载到手机中运行。

6.2 展望

由于移动终端的技术研发难度较高，各个公司对自己的方案都有极为严格的技术保密制度，可以借鉴和参考的资料极为缺少，开发时间有限以及本人经验不足，还存在一些不足之处：如模拟器的可配置性。模拟环境应该可以灵活的配置以根据不同需要来模拟具有相同结构、不同配置的手机系统。每个厂家的手机产品往往采用相同的软件体系结构，但各款手机的配置不尽相同，比如显示屏尺寸，键盘布局和功能键的设置都可能发生变化。

针对此问题本人所提的改良方案：为了模拟器能够透明地适应这些变化，可以为每种型号的手机指定不同的设备配置文件。配置文件中包含了特定型号手机的信息，模拟器根据这些信息自动设置不同的显示屏尺寸，用不同的键盘映射表隔离不同的键盘布局和功能键的设置，并调用特定手机的 MMI 应用程序对应的动态链接库文件。

这样，同样的模拟环境就可以适用与多款手机的开发和测试，这将进一步缩短 MMI 软件开发周期，降低开发成本。

致 谢

感谢我的导师吴延海教授！本文是在我的导师吴延海教授的悉心指导下完成的。两年来吴老师渊博的知识、敏锐的洞察力、严谨的治学态度、勤奋的敬业精神以及兢兢业业的工作作风，时刻激励和鞭策着我，给我留下了深刻的印象，并将是我以后学习和工作中的榜样。在此，谨向我的导师吴延海教授表示我最真诚的敬意和衷心地感谢。

感谢石伟琨、韦玉平、刁应君和陈宏浪。在本文的理论和实践研究中，他们给予了很大的支持和帮助。近一年的相处、一次次的技术和心灵的交流使我受益匪浅。

同时，还要感谢在学习生活中互相鼓励互相帮助共同奋斗的其它同学们。他们是：唐杰、李磊、张瑞先、王蓓、和煦、常淑娟、韩冰等同学。感谢他们在我课题和论文工作期间给予的关心和帮助，感谢他们营造了浓厚的学习气氛和轻松的实验室环境。

最后，对在百忙中抽出宝贵时间来阅读本文的各位评委老师表示衷心的感谢。

参考文献

- [1] 陈德荣. 数字移动通信系统. 北京邮电大学出版社, 1996.155~156
- [2] 孙孺石, 丁怀元. GSM 数字移动通信工程. 人民邮电出版社, 1996.168~179
- [3] 卢尔瑞, 孙孺石, 丁怀元. 移动通信工程. 人民邮电出版社, 1996.505~508
- [4] (法) Michel Mouly. GSM数字移动通信系统. 电子工业出版社, 1996.102~107
- [5] MTK. GSM Handset Application Layer. version w05.01, 2004
- [6] MTK. MT6219 GSM GPRS Baseband Processor Data Sheet V1.02_Malata. w05.01, 2004
- [7] MTK. Base band application. w05.01, 2004
- [8] MTK. KAL_ProgrammingGuide. version w05.01, 2004
- [9] MTK. Pixtel MMI Platform Source Code Training. version w05.01, 2004
- [10] 候俊杰. 深入浅出 MFC. 华中科技大学出版社, 1998. 202~208
- [11] 赵昌兴. A simple Introduction of the MMI Software Development of GSM Mobile Telephone.移动通信.1999年第6期
- [12] 张世琨, 王立福, 杨芙清. 基于体系结构开发模式.世界科技研究与发展. 2000年第3期
- [13] 左爱群, 黄水松. 软件体系结构的发展与研究.计算机工程与应用. 1999年第3期
- [14] 邵维忠, 杨芙清. 面向对象的系统分析. 清华大学出版社, 2000.99~103
- [15] 林华, 夏耘. 面向对象与结构化.计算机工程.2000年第7期
- [16] 郑人杰. 软件工程. 清华大学出版社, 1999.136~138
- [17] 赵永峰, 何方白. GSM 双频手机 MMI 的软件开发和硬件设计.重庆邮电学院学报.2001年6月, 第二期
- [18] 赵昌兴. 浅谈 GSM 手机的 MMI 软件开发.移动通信.1999年, 第六期
- [19] 夏庆德. 手机实时操作系统 RTXC 在 Windows 环境中的模拟.计算机应用.2004年1月, 第一期
- [20] 顾增辉, 王淑仙, 童晓慧, 刘锦高. 一种 GSM、GPRS 手机 MMI 的体系结构及其实现.华东师范大学学报.2003年12月, 第四期
- [21] 宋茂强. 通信软件设计基础. 北京邮电大学出版社, 2000
- [22] 马忠梅, 马广云, 许英慧, 田泽.ARM 嵌入式处理器结构与应用基础.北京航空航天大学出版社, 2002
- [23] David J. Kruglinski, Scot Wingo, George Shepherd. 希望图书创作室译. Visual C++ 6.0 技术内幕.2000年2月

-
- [24] (美) Roger S. Pressman. 软件工程—实践者的研究方法. 机械工业出版社, 1999
- [25] Ivar Jacobson, James Rumbaugh, Grady Booch. 姚淑兰, 唐发根译. UML 参考手册. 机械工业出版社, 2001
- [26] Van Schyndel R, Tirkel A, Osborne C. A digital watermarking. IEEE Proceeding on International conference on Image Processing, Austin, Tex., IEEE Press, 1994:86-90.
- [27] Wolfgang R B, Delp E J. A watermark for digital images. Proc. IEEE Int. conf. on Image Processing, Piscataway: IEEE Press, 1996, 3:219-222.
- [28] Pitas I. A method for signature casting on digital images. Proc. IEEE Int. conf. on Image Processing 1996, 3:215-218
- [29] Bender W, Gruhl D, Morimoto N. Techniques for data hiding. Proceedings of the SPIE, San Jose, CA, Feb. 1995, 2420:11-19
- [30] E. Walters, T. Richter. GSM Turnkey System Requirements Specification. 1999
- [31] Muller, F., Sorelius, J., Turina, D. Further evolution of the GSM/EDGE radio access network, Ericsson Review (English Edition) [Ericsson Rev (Engl Ed)], vol. 78, 2001
- [32] OMG. UML 1.3. June 1999
- [33] ETSI. GSM 02.07 Digital cellular telecommunication system (Phase 2), Mobile Station (MS) features. version 6.0.1 Release 1997
- [34] ETSI. GSM 02.30 Digital cellular telecommunication system (Phase 2), Man-Machine Interface (MMI) of the Mobile Station (MS). version 6.0.0 Release 1997
- [35] ETSI. GSM 02.40 Digital cellular telecommunication system (Phase 2), Procedures for call progress indications, version 6.0.0 Release 1997
- [36] ETSI. GSM 04.08 Digital cellular telecommunication system (Phase 2), Mobile radio interface layer 3 specification, version 6.0.0 Release 1997
- [37] TTP. GSM Handset Application Layer. version 2.2 1999
- [38] K. Verschaeve, A. Ek. Three scenarios for combining UML and SDL. R. Dssouli, Gbochmann, Y. Lahav (Eds). Proceedings of the Ninth SDL Forum. Montreal, Quebec, Canada, Elsevier, Amsterdam. 1999
- [39] B. Regnell, M. Andersson, J. Bergstrand. A hierarchical use case model with graphical representation, IEEE International Symposium and Workshop on Engineering of Computer-Based Systems. Germany. March 1996

附 录

硕士期间发表论文

- [1] 苏世鹏, 吴延海, 黄健. 用 C51 实现单片机 I/O 口模拟串口通信的方法, 西安科技大学学报增刊, 2005 年 6 月