

摘 要

近几年来，基于嵌入式系统的智能设备技术研发，已经成为二十一世纪微电子技术应用的主要动力。然而，目前国内对于嵌入式智能终端的研究往往侧重于特定的应用领域，乃至专一的某项目的实际成果或应用。究其主要原因，其一，当然是嵌入式系统自身的专用性特点；其二，则是嵌入式产品的生命周期从一定程度上限制了其通用性的研究；其三，则是克服上述两点的适用性高的嵌入式智能终端研究难度颇大。而这种针对每个项目设计不同的主板的嵌入式产品生产方式不仅增加了人力和资源成本，也极大的延长了开发周期。但是，由于嵌入式系统在某一领域之内往往具备功能相近、硬件系统相似的特点，搭建一个基本的、满足一定通用性、并具备高可扩展性的嵌入式平台必然会大幅度减少成本和节约时间。

在此目的下，本论文使用高性能 ARM9 微处理器，引入功能强大的嵌入式 WinCE 操作系统，开发出一种具备良好的通用性和可扩展性并提供丰富的外设接口的嵌入式智能终端核心板系统。为提高通用性和可扩展性，本系统使用主频为 400MHz 的 S3C2442A 微控制器，这在一定程度上满足了功能扩展带来的高速处理性能需求，因此能够延长该系统的生命周期；并且采用 Window CE5.0 操作系统使系统能够更加方便的进行二次开发和上层软件移植，同时给出总线扩展设备(并口微型打印机)的流接口驱动方法，提高系统的可扩展能力；另外，针对实际产品应用的不同方面，本系统提供 LCD 和 VGA 两种显示方式，并通过 USB Hub 扩展 4 个 USB Host，以及部分模块采用可插拔方式等方法，实现一板多用、裁剪扩展容易等优点，这也是本文创新的一个亮点。

本文首先对课题的选题背景作了简单叙述，阐述了本课题的研究目的、意义和国内外研究的现状；然后对嵌入式系统相关技术进行论述，从而为后续的研究奠定理论基础；接着深入探讨了整个系统平台的构建，其中包括硬件平台的搭建以及操作系统的移植、驱动程序的开发，对智能终端核心板系统的主要功能与实现方法进行了重点分析，并简要介绍了智能终端核心板系统的软硬件测试方法、过程以及测试结果；文章最后对课题所作工作进行了总结，并指出本系统尚待完善的地方。

关键词： 智能终端，嵌入式系统，Windows CE，BSP，ARM

Abstract

In the recent years, the development of intelligent equipment based on embedded system has been the main power of the application of microelectronic technology in the 21st century. However, in the home market, most of the researches on the development of embedded intelligent terminal focus on a specific application field or an appointed project. As for the main reasons, first of course is its own specificity of embedded system; second, the life cycle of embedded productions limits the study of its generality to some extent; third, the study of embedded intelligent terminal with higher adaptability that can overcome the above two problems is of high difficulty. While the production mode of embedded equipments that different boards are designed aiming at different projects not only increases the cost of the manpower resources, but also extends the developing periods. As the embedded system in a certain field always has the same functions and the similar hardware system, so establishing an embedded platform which has basic functions, satisfies the multipurpose feature, and has a great characteristic of extension absolutely can greatly reduce the cost and save time.

With this purpose, the thesis aims to develop an embedded intelligent terminal core board system that has a good generality and extensibility and can provide many peripheral interfaces by using high performance ARM9 microprocessor and porting powerful WinCE operating system. In order to strengthen the generality and extensibility, this system uses the S3C2442A microprocessor with the frequency of 400MHz, which satisfies the demand of high-speed processing due to functional extension to a certain degree and accordingly can extend the life cycle of the system. What's more, the system uses Windows CE 5.0 Operating System making the further development and the transplant of upper software more easily, and giving the driver method of BUS extending equipment to enhance the system's extensibility; On the other hand, as to the various purposes of application, the system takes some measures to design the multipurpose board and realize the feature of customizing and extending easily, e.g. providing two alternative display ways :LCD or VGA, giving 4 USB

Hosts by using a USB Hub, and offering some plug-in modules. This is the shining point of my thesis.

The thesis first briefly introduces the project background, the research purpose and significance, and status QUO at home and abroad. Second, it analyzes the relevant technologies in embedded system to lay a strong theoretical foundation for the further study. Third, the thesis deeply explores how to build up the whole platform, including the building of hardware platform, the transplant of operating system and the drivers' development. What's more, it emphasizes the analysis of the system's main functions and their realization. And the testing method as well as its procedure and its result of the system are presented. Finally the thesis summarizes the whole work of the project and points out some aspects the system needs to perfect.

Keywords: Enteligent Terminal, Embedded System, Windows CE, BSP, ARM

缩略表

AMBA: Advanced Microcontroller Bus Architecture
ARM: Advanced RISC Machines
BSP: Board Support Packet
CISC: Complex Instruction Set Computer
CRT: Cathode Ray Tube
DMA: Directly Memory Access
DSP: Digital Signal Processing
GWES: Graphic Window Event System
LCD: Liquid Crystal Display
MIPS: Million Instructions Per Second
MMU: Memory Management Unit
MPU: Micro Processor Unit
OAL: OEM adaptation layer
RAM: Random Access Memory
RISC: Reduced Instruction Set Computer
ROM: Read-Only Memory
RTC: Real-Time Clock
SD 卡: Secure Digital Memory Card
SDK: Software Development Kits
TFT: Thin Film Transistor
UART: Universal Asynchronous Receiver/Transmitter
USB: Universal Serial BUS
VGA: Video Graphic Array
WinCE: Windows CE

独创性声明

本人声明，所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得武汉理工大学或其它教育机构的学位或证书而使用过的材料。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作了明确的说明并表示了谢意。

研究生（签名）：王志伟 日期 2009.6.1

关于论文使用授权的说明

本人完全了解武汉理工大学有关保留、使用学位论文的规定，即：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，允许论文被查阅和借阅。本人授权武汉理工大学可以将本学位论文的全部内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存或汇编本学位论文。同时授权经武汉理工大学认可的国家有关机构或论文数据库使用或收录本学位论文，并向社会公众提供信息服务。

（保密的论文在解密后应遵守此规定）

研究生（签名）：王志伟 导师（签名）：张平 日期 2009.6.1

第1章 绪论

1.1 课题研究的的目的及意义

随着半导体技术的发展,越来越多的设备开始具备“智能”这一特性,而嵌入式系统就是各种设备里“智能”的实现手段。嵌入式系统是以应用为中心,以计算机技术为基础,软硬件开裁剪,适应应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统^[1]。它融合了计算机软硬件技术、通信技术和半导体微电子技术,是信息技术 (Information Technology)的最终产品^[2]。

由于需求的多样化,现实生活中到处都有嵌入式系统的身影。小至电子体温计,大到飞机的自动导航系统,都是嵌入式技术在各个领域里的应用。嵌入式系统由嵌入式处理器、嵌入式系统软件和嵌入式应用软件所组成^[3]。因此,当面向一个具体的应用建立一个专用的嵌入式系统之前,需要选择一款嵌入式处理器作为核心,并以此为基础选择相应的外设和软件方案。由于嵌入式系统的复杂性和设计任务的紧迫性,一般不采用先设计硬件再开发软件的顺序开发方式,而是采用软硬件协同设计的方式^[4]。目前,针对每个项目设计不同的主板的嵌入式产品生产方式不仅增加了人力和资源成本,也极大的延长了开发周期。并且,目前市场上的嵌入式产品普遍存在功能单一、兼容性低、可扩展性差、可维护性不高的缺陷。然而,在某一领域之内,嵌入式系统往往具备功能相近、硬件系统相似的特点。例如,在工业控制器领域,往往选用包括嵌入式处理器和实时操作系统的满足实时性的嵌入式系统。这些系统往往都具备一些如,Memory, Flash, 电源, 时钟, 串口, LCD, AD 转换接口, SD 卡以及 USB 等基本的外设接口。我们将嵌入式系统在内表现出的这种特性称为局部相似性。由于嵌入式系统在局部应用范围之内的相似性,搭建一个基本的、满足一定通用性、并具备高可扩展性的嵌入式平台必然会大幅度减少成本和节约时间。

在这种情况下,研究与开发具有一定通用性的嵌入式主板将由于其减少投资和缩短开发周期的特色而具备相当程度的市场竞争力。本课题源于深圳某公司智能终端核心板研发项目,目的是致力于研究具有兼容性、可扩展性、可维

护性的高性价比的核心板，使其通过软件裁剪可广泛应用于医疗、工控、消费电子等商业领域。本课题研究了嵌入式系统开发的相关技术，并以 WinCE 操作系统为例子，给出嵌入式系统开发的一般手段与方法，并在此基础上详细叙述了常见接口的驱动实现方式。综合来看，本课题的研究将为今后应用开发提供一个很好的起点，并能够为后续开发功能更完善、兼容性更好的系统提供借鉴。

1.2 国内外研究历史与现状

纵观嵌入式技术的发展轨迹，现代嵌入式技术呈现着明显的多元化发展趋势，各种专用的、通用的处理器层出不穷。这些种类繁多的处理器逐渐说明这样一个事实，高性能计算不再是台式机或大中型机的专利，嵌入式系统同样可以拥有强劲的计算能力和处理能力。

由于目前需求朝着多功能方向的发展，以 80C51 为代表的 8 位功能单一的单片机逐渐淡出核心处理器的地位，取而代之的是以 ARM (Advanced RISC Machines) 为代表的高主频支持多外设的 32 位处理器。对于 32 位处理器的研究与应用，国外始于二十世纪九十年代初期，而国内则始于二十世纪九十年代末期。因此，对于嵌入式技术的应用，特别是在某些领域，如车载、医疗、军事等领域，国外积累的经验比我们丰富的多。但是二十一世纪以来，国内电子工业的蓬勃发展带来的电子产品需求扩大与电子技术应用的逐步普及，以及计算机系统更新换代的速度，使得国内对嵌入式系统的研究处于相对较晚的劣势正在改变。国内对于智能终端的研究，从应用层面上来讲，主要偏向两方面：

1. 偏向于家居方面，如“智能家居系统”、“家庭媒体智能终端”、“智能小区监控系统”等；
2. 偏向于车载方面，如“车载智能终端”、“车载导航系统终端”、“GPS/GPRS 导航通讯系统在交通方面的应用”等。

当然，也不乏许多对于手持终端的应用与研究。总之，国内外对于嵌入式智能终端的研究往往侧重于特定的应用领域，乃至专一的某一项目的实际成果或应用。然而，对于系统的多用性甚至通用性方面的研究依然偏少。究其原因，其一，当然是嵌入式系统自身的专用性特点；其二，则是嵌入式产品的生命周期从一定程度上限制了其通用性的研究；其三，则是克服上述两点的适用性高的嵌入式智能终端研究难度颇大。因此，加大对通用性系统的研发力量

投入，解决嵌入式系统专用性与通用性的矛盾，并在此之上延长通用嵌入式系统的生命周期，是目前亟待解决的问题。

1.3 本文主要研究内容

本文主要致力于研究具备可扩展性和一定通用性的智能终端核心板的软硬件设计。为实现通用性和可扩展性目的，本系统使用主频为 400MHz 的 S3C2442A 微控制器能够延长该系统的生命周期；并且采用 Window CE5.0 操作系统使系统能够更加方便的进行二次开发和上层软件移植，提高系统的可扩展能力；另外，针对实际产品应用的不同方面，本系统提供 LCD 和 VGA 两种显示方式，并通过 USB Hub 扩展 4 个 USB Host，以及部分模块采用可插拔方式等方法，实现一板多用、裁剪扩展容易等优点。

本论文的主要工作包括：硬件方面，进行了 LCD 显示与触摸屏模块设计、并行打印机模块电路设计、网口电路设计、可插拔 VGA 模块电路设计和可插拔条形码扫描模块电路设计等；软件方面，进行了 WinCE 5.0 操作系统移植与构建，并详细叙述了部分模块的驱动开发，主要有 LCD、并口打印机、5*5 矩阵键盘和以太网口驱动开发。并在开发过程中综合运用多种测试方法，对整个软硬件平台进行了有效的检测。

本文共分为三个部分，共七章。

第一部分包括第一章和第二章。这部分首先对课题的选题背景做了简单叙述，阐述了开发具有一定通用性核心板的重要性以及国内外研究技术现状；然后对嵌入式系统的处理器和操作系统等关键技术进行论述和分析。

第二部分是本文的主体论述部分，包括第三、四、五、六章。这部分首先深入探讨了整个核心板平台的总体设计和主要硬件模块的设计，在搭建硬件平台之后，分析了嵌入式操作系统 Windows CE 的特性并给出了 BSP 的开发和系统的构建过程。接着，本部分详细介绍了核心板系统平台关键接口的驱动开发方法。作为论文的主体，本部分最后介绍了嵌入式系统测试的方法与流程，并给出本文的测试内容和测试结果。

第七章是本文的第三部分。这一章是论文的结束部分，主要介绍了本论文工作所取得的成果，并指出论文需要完善的地方。

第2章 相关技术分析

2.1 嵌入式系统

随着计算机技术及集成电路技术的发展，嵌入式技术日渐普及，嵌入式系统在通讯、网络、工控、医疗、电子等领域发挥着越来越重要的作用。我国嵌入式系统市场预计每年将直接创造千亿元的效益，所带动的相关工业产值将超过上万亿元，称为电子信息产业新的经济增长点。甚至有专家表示，嵌入式技术是中国厂商从“中国制造”向“中国创造”转变的最佳契机。所以嵌入式系统的软硬件开发，已被人们公认为是一个具有良好发展潜力的行业。

虽然嵌入式系统（Embedded System）已经称为了当今最热门的技术之一，但是嵌入式系统本身还是一个相对比较模糊的定义^[6]。到底什么是嵌入式系统，什么样的技术可以称之为嵌入式技术，目前仍在讨论之中。

嵌入式系统根据不同的标准有多种定义。目前最通用的定义是：以应用为中心，以计算机技术为基础，软硬件可以裁剪，适应于对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统^[1]。按照历史性、本质性及普遍性要求，嵌入式系统应定义为：嵌入到对象体系中的专用计算机系统。嵌入式系统一般由嵌入式处理器，外围硬件设备，嵌入式操作系统以及用户应用软件四个部分组成^[7]，用于实现对其他设备的控制、监视或者管理等功能。

总而言之，嵌入式系统是结合了微处理器或微控制器的系统电路与其专属的软件，用来达到系统操作效率与成本的最高比^[8]。与传统的通用计算机相比，嵌入式系统具有以下几个重要的特征^{[9][10][11][12]}：

（1）嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物。这一特性决定了它的设计与开发要将软件和硬件相结合，特别是要对计算机系统底层技术整合。

（2）嵌入式系统具有较高的可靠性，较低的成本和较低的耗电量，同时，还对实时性有一定的要求。

（3）嵌入式系统的 CPU 等硬件设备必须具备体积小、功耗低、集成度高的特性，只有这样才能把很多功能集成在一个嵌入式芯片的内部。

(4) 嵌入式系统的硬件和软件都必须高效率的设计, 量体裁衣、去除冗余, 力争在同样的硅片或载板面积上实现更好的性能, 这样才能在具体应用中对处理器的选择更具有竞争力。

(5) 为了提高软件执行速度和系统可靠性, 嵌入式系统的软件一般固化在 EPROM、EEPROM 或 Flash 等非易失性存储器中, 而不是像通用计算机系统那样存储于磁盘等载体中。

(6) 嵌入式系统本身不具备自举开发能力, 即使设计完成以后用户通常也是不能对其中的程序功能进行修改的, 必须具备一套开发工具和环境才能进行开发。

同时, 嵌入式系统作为控制、监视或辅助设备、机器甚至操作的装置, 它又具有下列四项特征^{[6][13]}:

- (1) 通常执行特定的功能;
- (2) 与微电脑外围构成核心;
- (3) 严格的时序与稳定性要求;
- (4) 全自动循环操作。

综上所述, 嵌入式系统主要由嵌入式处理器、相关支撑硬件、嵌入式操作系统以及特定的应用软件系统等组成。但总的来讲嵌入式系统包括软件和硬件两个部分, 硬件包括处理器/微处理器、存储器、外设和 I/O 端口、图形控制等; 软件部分包括操作系统软件 and 应用程序软件。

2.2 嵌入式微处理器

嵌入式系统的核心是嵌入式微处理器。嵌入式微处理器一般具备以下 4 个特点^[14]: (1) 对实时多任务有很强的支持功能, 能完成多任务并且有较短的中断响应时间; (2) 具有功能较强的存储区保护功能; (3) 可扩展的处理器结构, 以能最迅速地开发出满足应用的各种性能的嵌入式微处理器; (4) 功耗很低。

嵌入式微处理器就是与通用计算机的微处理器对应的 CPU^[15]。但在工作温度、抗电磁干扰、可靠性等方面一般都作了各种增强。嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点, 芯片中往往包括少量的 ROM 和 RAM 甚至一定容量的 Flash, 一般还包括总线接口、常用设备的控制器、各种外设等器件, 从而极大地降低了构成系统的复杂性。在嵌入式微处理器的发展中,

以微处理器的指令集特性，可以将嵌入式系统微处理器分成两种主要架构，一种是 CISC（复杂指令集 Complex Instruction Set Computer）架构，另一种是 RISC（精简指令集 Reduced Instruction Set Computer）架构。本文主要介绍我们所使用的 RISC 架构的代表者——ARM 微处理器。

2.2.1 ARM 微处理器

ARM(Advanced RISC Machines)微处理器，由 1991 年成立于英国剑桥的 ARM 公司设计。然后由世界各大半导体生产商购买 ARM 微处理器核，根据各自不同的应用需求，加入适当的外围电路，从而形成自己的 ARM 微处理器芯片。

ARM 处理器已经遍布工业控制，消费类电子产品、通信系统和网络系统等各类产品市场，约占据了 32 位 RISC 微处理器 75% 以上的市场份额，ARM 技术正在渗入生活的各个方面^[16]。ARM 微处理器目前包括几个系列，除了具有 ARM 体系结构的共同特点以外，每一个系列的 ARM 微处理器都有各自的特点和应用领域。其中 ARM9 系列微处理器在高性能和低功耗特性方面提供最佳的性能。具有以下特点^{[17][18]}：

1. 体积小、低功耗、低成本、高性能；
2. 大量使用寄存器，指令执行速度更快；
3. 数据操作多数在寄存器中完成，方式灵活简单，执行效率高；
4. 指令长度固定；
5. 5 级整数流水线，指令执行效率更高；
6. 提供 1.1 MIPS/MHz 的哈佛结构；
7. 支持 32 位 ARM 指令集和 16 位 Thumb 指令集；
8. 支持 32 位的高速 AMBA 总线接口；
9. 全性能的 MMU，支持 WinCE, Linux, Palm OS 等多种嵌入式操作系统；
10. MPU 支持实时操作系统；
11. 支持数据 Cache 和指令 Cache，具有更高的指令和数据处理能力。

另外，ARM920T 的片上混合数据和指令 Cache 以及写缓冲器显著地提高了平均运行速度，降低了处理器所需的存储器带宽的平均量。同时，允许外部存储器以最小性能损失支持附加的处理器和直接存储器访问 DMA 通道，以及用不同任务 ID 进行虚拟的分配改善了在 Cache 使能下任务切换操作的性能。这些重

定位的虚拟地址由嵌入式 ICE 模块监控。所有这些特性使得 ARM9 系列微处理器非常适用于嵌入式控制和面向对象系统。

2.2.2 ARM 微处理器的选择

ARM 微处理器有十几种内核结构以及为数众多的内部功能配置组合,给选型方案带来一定的复杂性。在一个系统中使用什么样的嵌入式处理器内核主要取决于应用的领域、用户的需求、成本问题、开发的难易程度等因素。从应用的角度出发,本项目在选择 ARM 微处理器时除价格这一重要因素外,主要依据以下原则^{[19][20][21]}:

1. ARM 微处理器的内核选择

ARM 微处理器包含一系列的内核结构,以适应不同的应用领域,用户如果希望使用 WinCE 或标准 Linux 等操作系统以减少软件开发时间,就需要选择 ARM720T 以上带有 MMU(Memory Management Unit)功能的 ARM 芯片。本课题中选择的一款 ARM920T 内核的微处理器,带有 MMU 功能,从而实现对 WinCE 操作系统的支持。

2. 工作频率

系统的工作频率在很大程度上决定了 ARM 微处理器的处理能力。常见的 ARM7 芯片系统主时钟为 20MHz-133MHz,常见的 ARM9 的系统主时钟频率为 100MHz-233MHz,ARM10 最高可以达到 700MHz。另外,不同芯片对时钟的处理不同,有的芯片只需要一个主时钟频率,有的芯片内部时钟控制器可以分别为 ARM 核和 USB、UART、DSP、音频等功能部件提供不同频率的时钟。

3. 芯片内存储器的容量

芯片内部存储器通常包括 RAM、ROM、Flash 等存储单元。在实际项目中,内部存储器的容量通常与项目 PCB 板的尺寸大小是否受限联合考虑。因为大多数的 ARM 微处理器片内存储器的容量都不太大,如果项目对 PCB 板的尺寸大小有限定,通常考虑内含存储器的微处理器。否则,在设计系统的时候可以采用外扩存储器的方式。

4. 片内外围电路的选择

除 ARM 微处理器核以外,几乎所有的 ARM 芯片都针对不同的应用领域,扩展了相关功能模块,并集成在芯片之中,称之为片内外围电路,如 USB 接口、IIC 接口、LCD 控制器、键盘接口、RTC、ADC 和 DAC、DSP 协处理器等,设

计者应分析系统的需求，尽可能采用片内外围电路完成所需的功能，这样既可简化系统的设计，又能够提高系统的可靠性。

另外，芯片可加密性、芯片的 GPIO 数量等也是在处理器选型过程中需要统筹考虑的。

2.3 嵌入式操作系统

2.3.1 嵌入式操作系统简介

嵌入式操作系统是一种支持嵌入式系统应用的操作系统软件，它是嵌入式系统极为重要的组成部分，通常包括系统内核、与硬件相关的底层驱动程序、设备驱动接口、通信协议、图形界面、标准化浏览器等^{[22][23]}。通用操作系统的基本特点如能够有效管理越来越复杂的系统资源；能够把硬件虚拟化，使得开发人员从繁忙的驱动程序移植和维护中解脱出来；能够提供库函数、驱动程序、工具集以及应用程序。嵌入式操作系统除具备以上特点外，还是与通用操作系统有许多不同的地方，如嵌入式操作系统在系统实时高效性、硬件的相关依赖性、软件固态化以及应用的专用性等方面具有较为突出的特点^[7]。

目前市场上主流的嵌入式操作系统包括：VxWorks、Windows CE、嵌入式 Linux、uC/OSII、PlamOS、Symbian 等。按实时性划分，嵌入式操作系统可分为实时操作系统和非实时操作系统^[24]。上述几种嵌入式操作系统除嵌入式 Linux 为非实时操作系统以外，其余都属于嵌入式实时操作系统。

在嵌入式系统中采用嵌入式实时操作系统，主要基于以下几方面考虑^{[19][22][25]}：

1. 嵌入式实时操作系统提高了系统的可靠性。在嵌入式系统中，传统的前后台系统软件设计在遇到强干扰时，容易使得运行的程序产生异常，从而造成系统的崩溃。而实时操作系统管理的系统，这种干扰可能只是引起若干进程中的一个被破坏，可以通过系统运行的系统监控进程对其进行修复。通常情况下，这个系统监视进程用来监视各进程的运行状况，遇到异常情况时采取一些利于系统稳定可靠的措施，比如把有问题的任务清除掉等等。

2. 嵌入式实时操作系统充分发挥了 32 位微处理器的多任务潜力，使得 32 位微处理器比 8 位、16 位微处理器更快。32 位微处理器本来是为运行多用户、

多任务操作系统而设计的，特别适用于运行多任务实时系统。

3. 提高了开发效率，缩短了开发周期。嵌入式实时操作系统都具有可伸缩和易裁剪的特性，易于在不同的 ARM 核的嵌入式处理器上移植。另外，嵌入式实时操作系统可以使用大量高效的 ARM 开发工具，极大的方便了系统开发。

2.3.2 嵌入式操作系统的选择

嵌入式操作系统在目前的嵌入式系统中得到了越来越广泛的应用，尤其在功能复杂、系统庞大的系统中。当我们在设计嵌入式产品时，嵌入式操作系统的选择至关重要。一般而言，在选择嵌入式操作系统时，可以遵循以下原则^[22]：

1. 实时性

不同的操作系统，实时性差别比较大。使用加入实时补丁等技术的嵌入式 Linux，最坏的情况下，中断延时只有 436 微秒，而 99.9%的情况是 195 微秒。考虑到最新的 Linux 在实时性方面的改进。它可以适合于 90~95%的嵌入式系统应用。如果希望更快的实时响应，如高速的 A/D 转换，要求几个微秒以内的中断延时，采用 $\mu\text{C}/\text{OS}$ 是合适的，另外 Vxworks 这样传统的嵌入式操作系统也可以满足这样的强实时性要求。

2. 可移植性

嵌入式开发的专用型，对操作系统的可移植性提出了较高的要求。在进行嵌入式系统开发时，可移植性是要重点考虑的问题。

3. 可利用资源

操作系统的可利用资源对于快速、低成本、高质量的推出适合用户需求的产品是一个非常重要的方面。Linux 和 WinCE 都有大量的资源可以利用，这是他们被看好的重要原因。

4. 系统定制能力

信息产品不同于传统 PC 的 Wintel 结构的单纯性，用户的需求千差万别，硬件平台也不一样，对系统的定制能力提出了较高的要求。Linux 由于其源代码，在定制能力方面具有优势；而 WinCE 则以其良好的技术支持，在系统定制方面也占有一定优势。

5. 成本

操作系统按经营模式分为免费和商业系统两大类。Linux 属于免费操作系统，而 Windows CE 以及 VxWorks 等都属于商业系统，需要支付许可证使用费。

同时，选择某一系统可能会对其他一系列的因素产生影响，如对硬件设备的选型、人员投入、以及公司管理和与其他合作伙伴的共同开发之间的沟通等等。

另外，系统的稳定性、开发难度以及系统对中文内核的支持等等都是在选择操作系统的时候需要考虑的因素。

第3章 系统硬件设计

3.1 智能终端核心板的功能需求

本课题研究的智能终端核心板的主要功能如下：

- 能够支持 TFT-LCD 和 VGA 两种可选界面显示方式；
- 能够支持总线方式扩展低速设备，如微型打印机；
- 支持多串口用于扩展其他功能，如条码识别、GPS 等；
- 支持 4 个 USB Host，用于功能扩展，如 USB 鼠标、键盘等；
- 支持 10/100M 以太网口，用于机器间交互或终端与服务器交互等；
- 支持 5×5 矩阵键盘用于人机交互；
- 包含 4 通道 10 为 ADC 模数转换接口，用于数据采集等；

3.2 S3C2442A 微处理器特性

根据项目需求和微处理器选型的原则，本系统将采用频率 400M 的高性能 Samsung S3C2442A 微处理器。该处理器芯片内部集成 64MB SDRAM 和 128M NAND FLASH，片内外围设备接口丰富，满足本项目需求并能良好的支持功能扩展。以下是该微处理器的特征介绍^[26]：

S3C2442A 处理器是 Samsung 公司基于 ARM920T 处理器核，采用 0.13um 制造工艺生产的 32 位微控制器。该处理器除具备 ARM9 系列处理器特性外(有关 ARM9 系列处理器的特性可参见本文 2.2.1 节)，还包含如下各功能模块：

1. 最高工作频率可达 400MHz，外带 16KB 指令 Cache 和 16KB 数据 Cache；
2. 扩展内存控制器(SDRAM 控制和芯片选择逻辑)；
3. 专用 DMA 通道的 LCD 控制器(支持 4K 彩色 STN 和 256K 彩色 TFT)；
4. 4 路 DMA(带外部请求信号)；
5. 3 路 UART(红外 1.0，64 字节发送/接收 FIFO)；
6. 2 路 SPI；
7. 1 路多主 IIC 总线接口；

8. 1路 IIS 音频总线接口;
9. SD 卡主机接口(version 1.0)和 MMC(Multi-Media Card)接口(兼容 V2.11 协议);
10. 2 个 USB 主机接口/1 个 Device 接口(V1.1);
11. 4 路带 PWM 定时器/一路内置定时器/看门狗定时器;
12. 8 路 10 位的 ADC 触摸屏接口;
13. 数码相机接口(最大支持 4096*4096 输入像素, 2048*2048 缩放输入像素);
14. 130 位通用 I/O 口, 其中 24 路外部中断源;
15. 电源管理方面: 支持 Normal, Slow, Idle 和 Sleep 模式;
16. 带 PLL 的片上始终发生器;

由此可见, 该 CPU 具备优良的运算能力, 具有进行较为复杂事务处理能力, 并且支持多外设扩展。同时, 较大的容量和快速访问的 RAM 是执行复杂处理的基本保证。另外, 较大容量外部存储器能够保证操作系统和用户软件的保存与正确运行, 还能为用户预留较大的存储空间来存储业务信息或私人资料。

3.3 系统硬件总体设计

智能终端核心板主要由微处理器、触摸屏、LCD 显示模块、并口打印机模块、USB Host 模块、USB Device 模块、矩阵键盘模块、网口模块、SD 卡、JTAG 接口等主要部分组成, 系统结构如图 3-1:

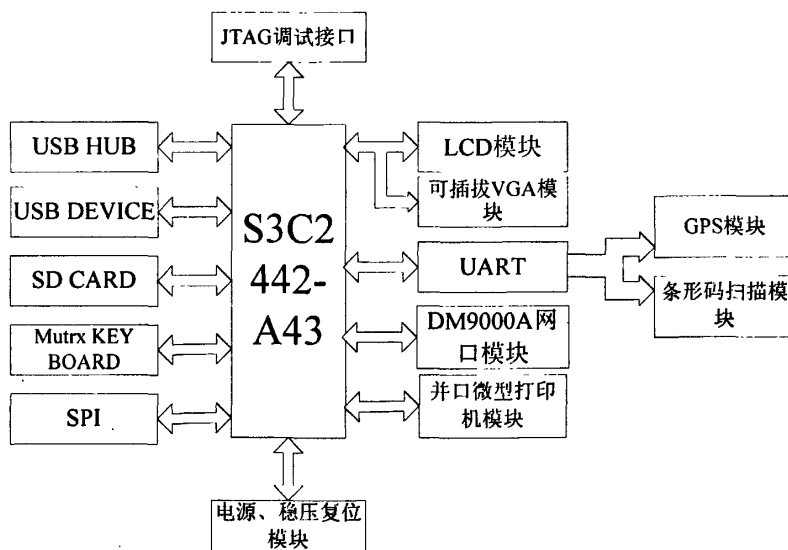


图 3-1 智能终端核心板结构框图

3.4 系统硬件各模块设计

3.4.1 LCD 显示与触摸屏模块电路设计

LCD 模块满足了嵌入式系统日益提高的现实需求，是目前消费、医疗电子类嵌入式产品不可或缺的模块之一。它可以显示汉字，字符和图形，同时还具有低压，低功耗，体积小，重量轻，超薄等很多优点^{[27][28]}。目前使用较多的屏主要有两种：STN 和 TFT 屏。S3C2442A 中的 LCD 控制器可以方便的控制上述两种屏的显示，本系统中采用的是 TFT 屏。由于 S3C2442A 处理器本身自带 LCD 控制器可以驱动本课题选用的液晶屏，所以控制电路的设计可以省去显示控制电路，只需进行电源驱动和偏压驱动的设计即可^[29]。LCD 模块电路图如图 3-2 所示。

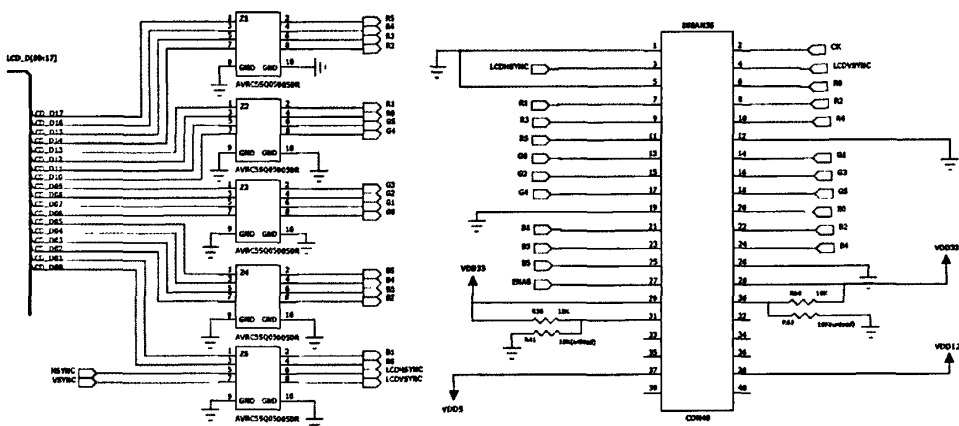


图 3-2 LCD 模块电路图

S3C2442A 内置 1 个 8 信道的 10bit 模数转换器(ADC),该 ADC 能以 500KSPS 的转换速率将外部的模拟信号转换为 10bit 分辨率的数字量。同时 ADC 能与微处理器的触摸屏控制器协同工作，完成对触摸屏绝对地址的测量。

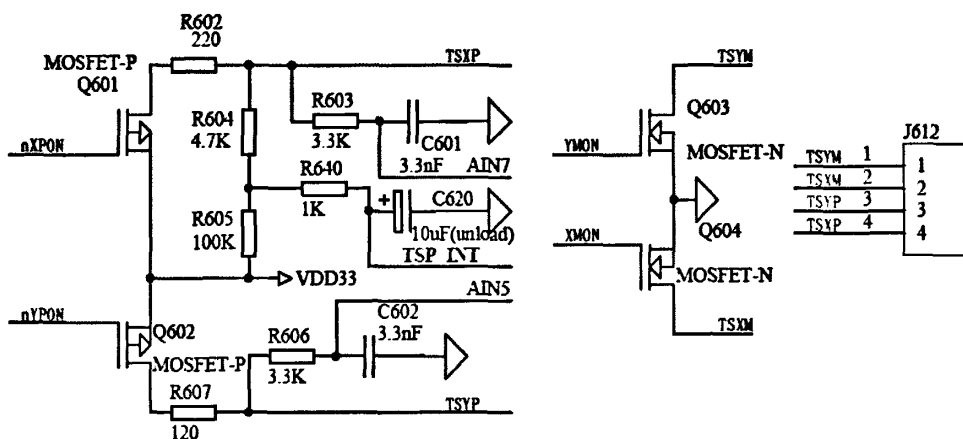


图 3-3 触摸屏坐标转换控制电路

当触摸屏被按下时，平常绝缘的两层导电层在触摸点位置导通，从而触发中断通知处理器进行 A/D 转换。过程如下：首先导通 FET 管组 Q602 和 Q604，X 轴回路加上 +5V 电压，同时将 FET 管组 Q1 和 Q3 关闭；再启动处理器的 A/D 转换，电路电阻与触摸屏按下产生的电阻输出分量电压，并由 A/D 转换器将电压值数字化，计算出 X 轴的坐标。中断处理程序通过导通不同 MOS 管组，使接触部分与控制器电路构成电阻电路，并产生一个电压降作为坐标值输出。

3.4.2 并口打印机模块电路设计

并口打印机采用北京炜煌公司的 WH-A6 型热敏微型打印机。该打印机时序严格，能打印全部 ASCII 码、国家一二级标准汉字、曲线、条形码等。由于该微型打印机接收 5V TTL 电平，而 S3C2442A 输出电压只有 1.8V，因此为了匹配电平需要连接两级 BUFFER。本文采用 PHILIPS 公司的 74ALVC164245 电平转换芯片。在实际数据传输过程中，由于总线速度过高，数据在总线上保存时间太短^[30]（以 50M 的 PCLK 为例，数据保存时间小于 20ns）通过 IO 控制打印机读取总线数据时往往读不到正确数据，因此，在设计中采用 74HC573 作为锁存器用以解决打印机信号与总线速度不匹配的问题。同时，锁存器的加入除了延长数据在总线上保留的时间之外，也给驱动的调试带来很大方便。因为总线上的数据是杂乱的，加入锁存器并通过片选选通打印机后，在锁存器的输出端则是送往打印机的数据。可以通过断点方式，测试锁存器是否工作正常，时序是

否符合，从而快速判断程序问题所在。并口打印机模块与 CPU 的连接如下图 3-4 所示。

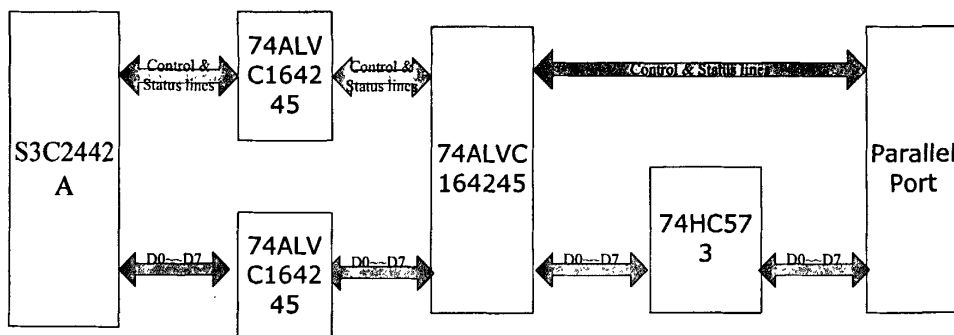


图 3-4 并口打印机模块

3.4.3 网口模块电路设计

为更好的与上位机、服务器或其他终端通讯，本系统通过外接一片 DM9000A 以太网 MAC 芯片扩展了一个 10M/100M 自适应以太网接口，占用资源 nGC S7/EINT1。DM9000A 是中国台湾 DAVICOM 公司推出的一款高速以太网接口芯片，其基本特征是：集成 10/100M 物理层接口；内部带有 16K 字节 SRAM 用作接收发送的 FIFO 缓存；支持 8/16bit 两种主机工作模式；通过 HP 认证的 AUTO-Mdix(支持直接互连自动翻转)功能；支持 TCP/IP 加速(IPV4 check sum offload)减轻 CPU 负担，提高整机效能；10ns I/O 读写时间。DM9000A 以太网控制器遵循 IEEE 颁布的 802.3 以太网传输协议。该芯片还集成 EEPROM 接口，自举时通过 EEPROM 接口输入到芯片中，从而实现自动初始化。考虑到成本原因，本系统没有接 EEPROM，而是通过软件方式初始化芯片。网口模块电路图如下图 3-5 所示。

3.4.4 可插拔 VGA 模块电路设计

大多数嵌入式产品的显示终端都选择 LCD，但在某些需要大屏幕显示的应用中，工业级 LCD 的价格比较昂贵，并且现有的大屏幕显示器(包括 CRT 显示器和 LCD 显示器)一般都采用统一的 15 针 VGA 显示接口。因此，与 LCD 相比，VGA 显示接口标准统一，市场覆盖广，技术成熟，使用非常广泛。VGA 接口是一个模拟信号接口。为适用于不同的应用场合，本文在设计中除了给出 LCD 显示方案外，还给出 VGA 显示方案，其中 VGA 采用可插拔方式。VGA 模块使用外扩 CHROTEL 的 CH7005C 芯片实现，利用该芯片将 S3C2442A 自带的 LCD 扫描式接口转换为 VGA 接口，然后用带有 VGA 接口的显示器显示。

CH7005C 芯片可实现数字 PC 信号到标准的 VGA 编码输出。该芯片提供了通用的数字像素输入，可以通过高精度的锁相环(PLL)提供稳定时钟，进而提供高质量的图形信息。CH7005C 支持垂直和水平同步非全画面功能，同时支持 5 中不同的分辨率，包括 640×480 和 800×600。图 3-6 为 CH7005C 芯片的原理框图，主要包括了数字输入接口、RGB-YUV 转换器、YUV-RGB 转换器、NTSC/PAL 编码器和滤波器、一个三路的 DAC、锁相环、总线控制模块以及时钟与同步发生器等。

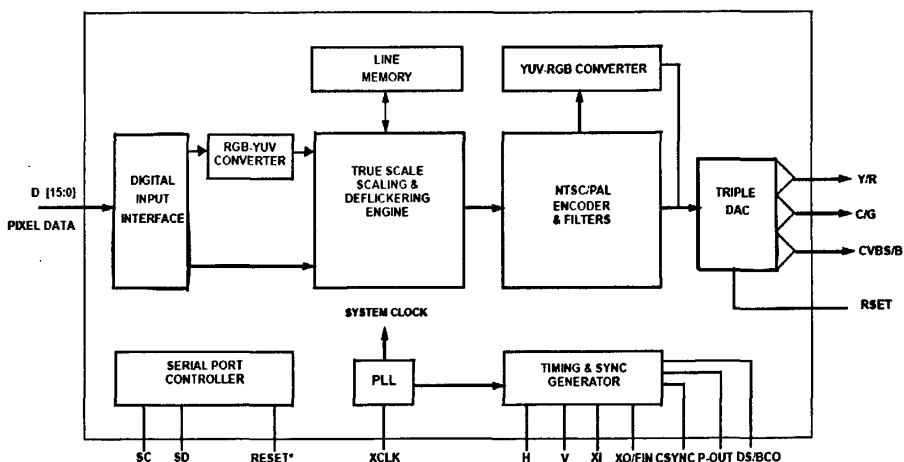


图 3-6 CH7005C 芯片内部结构框图

采用CH7005C是目前性价比最好的VGA实现方式,该芯片内部自带行缓存,但缓存较少,不支持大数据量的视频播放。由于对CH7005C芯片的初始化需要IIC时序,而S3C2442A芯片内部集成IIC总线控制器,因此本文直接采用S3C2442A的IIC总线控制器接口GPE14和GPE15。而对于内部未集成IIC总线控制器的处理器,也可采用IO口模拟IIC时序。CH7005C模块接口电路如下图3-7所示。

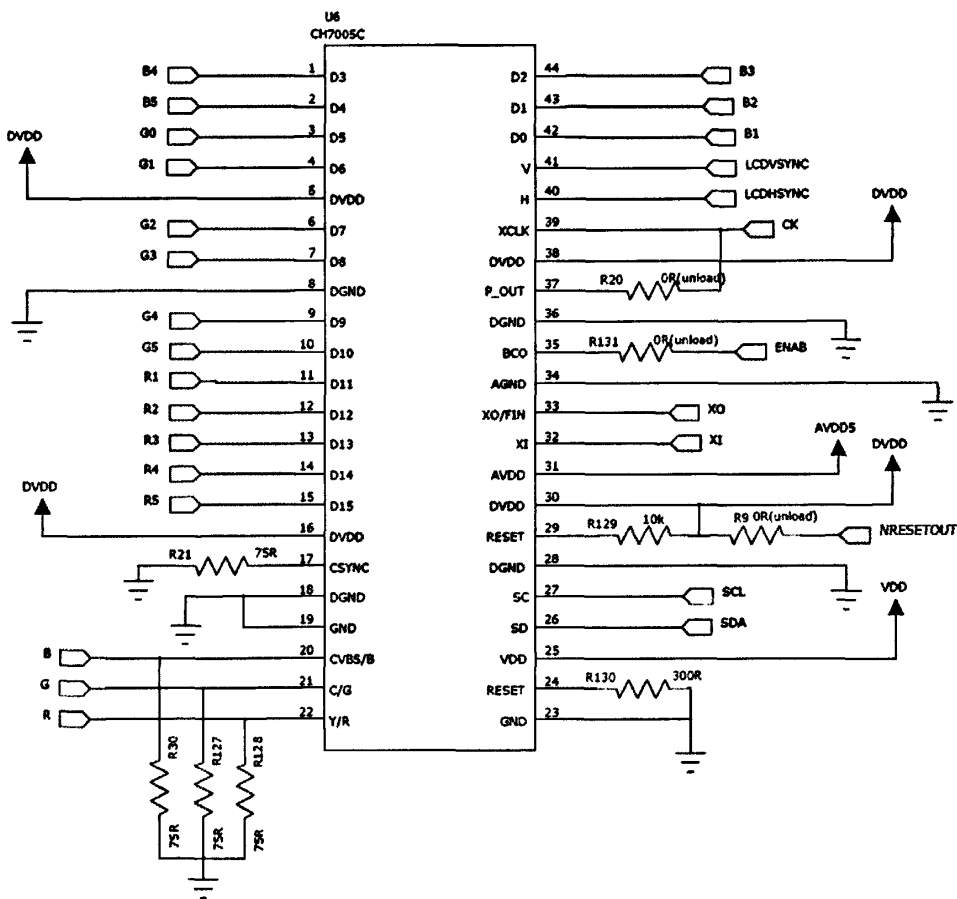


图 3-7 VGA 模块电路图

3.4.5 可插拔条形码扫描模块电路设计

为扩展核心板的应用领域提升其商业价值,可通过串口扩展条形码扫描模

块、GPS 模块等。本系统采用可插拔的方式扩展条形码扫描模块。条码扫描模块采用 Intermec Technologies 公司的 EV15 扫描芯片，该芯片集扫描、解码、传输数据于一体，并使用串口与 S3C2442A 连接通信。占用资源：1 个 IO 口，1 个中断、1 个 5 线串口。其电路连接图如下图 3-8 所示。

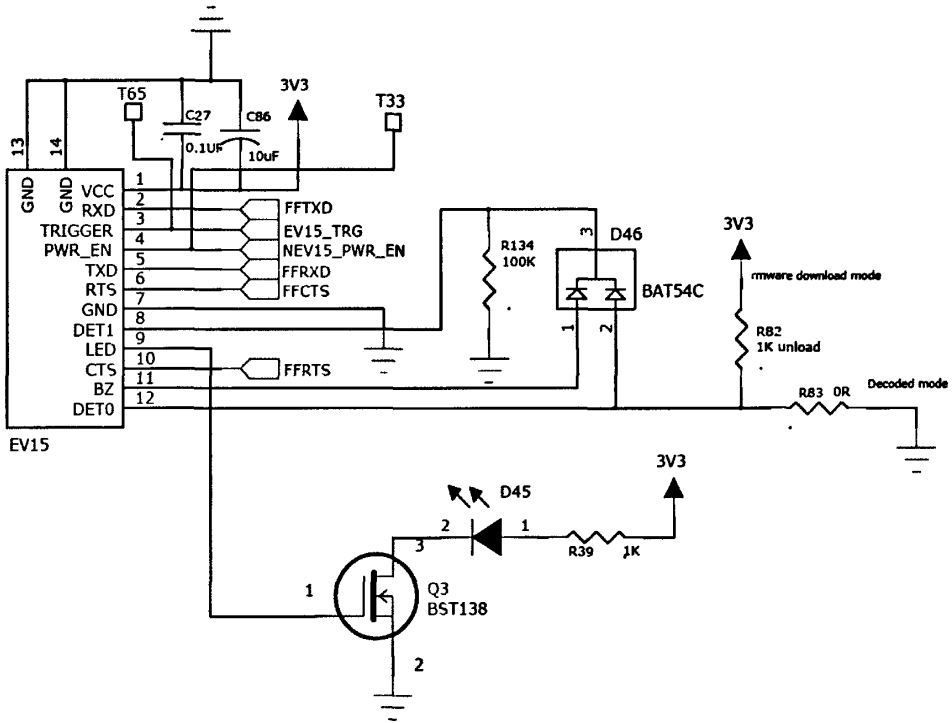


图 3-8 条形码扫描模块

第4章 WinCE 操作系统平台的构建

4.1 WinCE 操作系统分析

Microsoft Windows CE（以下简称 WinCE）是一个开放的、可裁剪的、32 位的实时嵌入式窗口操作系统^[31]。和其他桌面版窗口操作系统相比，它具有可靠性好、实时性高、内核体积小等特点，所以被广泛应用于嵌入式智能设备的开发，被广泛应用于工业控制、信息家电、移动通信、汽车电子、个人电子消费品等各个领域，是当今应用最多、增长最快的嵌入式操作系统之一。

WinCE 诞生于 1996 年，发行的第一个版本是 Windows CE 1.0，现在已经发展到 WinCE 6.0^[32]。由于本文所做的智能终端核心板系统研究起始于一年以前，考虑到 WinCE 操作系统应用的成熟性，所以选择了 WinCE 5.0。WinCE 5.0 是微软公司在嵌入式操作系统市场上的一个重要的产品，较之应用非常广泛的前一个版本 WinCE 4.2，它具有更加丰富的功能、更好的易用性和更稳定的性能。

4.1.1 WinCE 的设计目标与特点

为了满足市场和嵌入式环境的要求，WinCE 的设计要求主要有以下几点^[32]^[33]：

1. 高度模块化和小内存占用；
2. 支持多种处理器和计算机结构，并支持多种设备接口；
3. 支持高要求的实时性，满足不通设备对于时间关键性任务的时间和性能要求；
4. 为用户提供丰富的多语言和多媒体支持；
5. 提供强大的开发工具支持。

选择 WinCE 5.0 作为本核心板平台的操作系统，主要基于以下特点^[32]：

1. 多硬件平台支持

WinCE 支持包括 x86、ARM、MIPS、SuperH 等嵌入式主流的 CPU 结构。WinCE 提供了相应的高质量 BSP 和驱动程序支持，并为每种不同类型的硬件设备、总线或端口都提供了例驱动程序源代码，便于用户直接应用这些驱动程

序或快速为自己的硬件设备开发驱动程序。

2. 稳健的实时性支持

实时系统需要软件和硬件的协作支持，当硬件确定后，控制系统的实时行则由操作系统来决定。WinCE 支持硬实时操作系统，满足了基本商业应用领域产品的实时行要求。

3. 良好的可定制性，精简的模块化

WinCE 是高度模块化的嵌入式操作系统，在用户定制过程中，可以按照开发要求，只把需要的模块包含进来，对于硬件资源宝贵的嵌入式设备而言，尤其重要。

4. 强大的开发工具

与其他嵌入式操作系统相比，WinCE 提供了更为强大的开发工具支持。对于操作系统定制，可使用 Platform Builder，对于应用程序，可选择 Embedded Visual C++或 Visual Studio.NET。

另外，丰富的多媒体和多语言支持、有线和无线网络连接支持，类 Windows 的友好的人机界面，也有助于 WinCE 产品的推广应用。

4.1.2 WinCE 的体系结构

在嵌入式领域里，层次化的设计方法在软件体系结构中非常普遍。在这种设计方法中，软件系统被划分成彼此相关的几个层次。不同的层次都有不同的关注要点和要实现的功能。层次与层次之间构成单项依赖，原则上每一层次都只与它的邻层打交道，即利用下层提供服务构建自身，同时向它的上层提供服务。这样的系统有良好的扩展性和可维护性。

WinCE 正是这样一种分层结构，如下图图 4-1 所示，从底层向上分别为硬件层、OEM 层、操作系统层和应用层。每一层分别由不同的模块组成，每个模块又由不同的组件组成。这种层次性的结构试图尽量将硬件和软件、操作系统与应用程序隔离开，以便于实现系统的移植，便于进行硬件、驱动程序、操作系统和应用程序等开发的人员分工合作、并行开发^[31]。

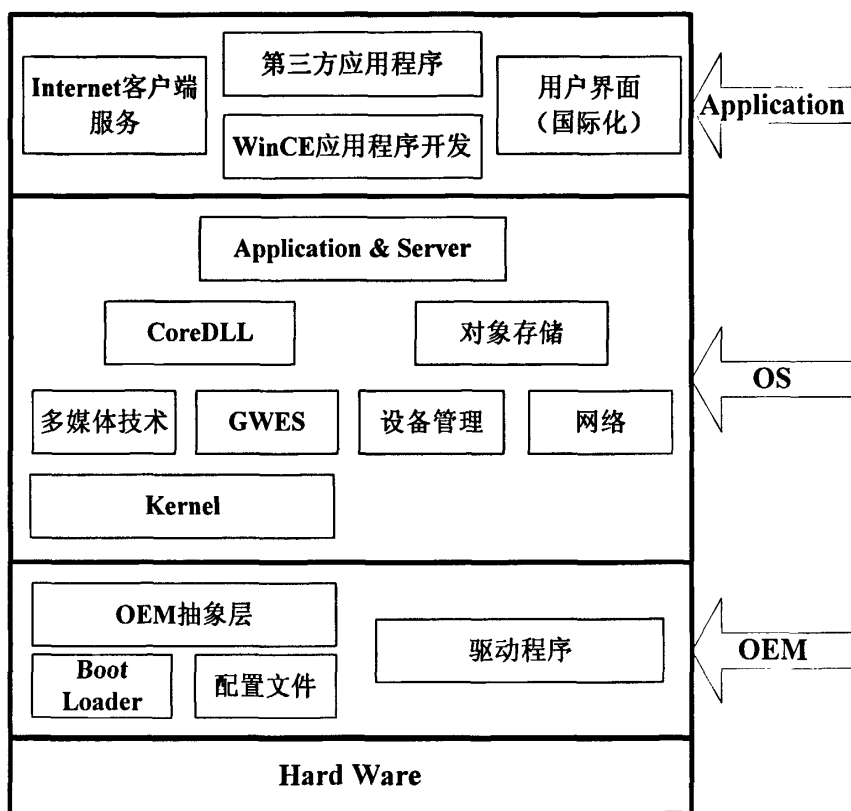


图 4-1 WinCE 体系结构

1. 硬件层

硬件层是系统的最底层，主要由硬件部分，包括微处理器、存储器、电源和周边的各种设备。WinCE 系统所需的最小硬件配置包括处理器、实时时钟、存储器等。硬件平台还可以支持其他的外围设备，例如串口、网卡、键盘等。

2. OEM 层

OEM 层逻辑上位于操作系统层与硬件层之间，是一个硬件抽象层，它提供了硬件和操作系统之间的接口。WinCE 操作系统要能在特定的硬件上运行，硬件必须能够加载 WinCE 操作系统，同时，操作系统也要能够实现对硬件系统的初始化并驱动这些硬件，这些工作都是由 OEM 来完成的。OEM 层包括以下模块：OAL(OEM Adaptation Layer, OEM 抽象层)、引导程序(Bootloader)、配置文件、驱动程序。

3. 操作系统层

操作系统层是 WinCE 操作系统的核心层，它既要为下层 OEM 层提供接口和

服务,也要为上层的应用程序提供服务与应用程序编程接口。WinCE 操作系统层主要由以下几个模块组成:内核模块、CoreDLL、设备管理模块、文件管理模块、GWES 模块、网络通信模块、对象存储模块、多媒体模块、Win32 系统服务模块等^{[7][31]}。

4. 应用层

应用层就是应用程序的集合,它们通过 Win32 API 来获得操作系统服务。它包含 WinCE 为用户提供的一些应用程序(如 Internet 客户服务,用户接口等),也可以是用户为特定的嵌入式系统开发的应用程序。

4.1.3 WinCE 的开发环境

在 WinCE 操作系统定制、开发过程中,使用最为广泛的工具就是 Platform Builder。Platform Builder 是微软公司提供开发人员进行设计、创建、编译、测试和调试 WinCE 操作系统平台的工具。它运行在桌面 Windows 下,开发人员可以通过交互式的环境来设计和定制内核、选择系统特性,然后进行编译和调试。同时,开发人员还可以利用 Platform Builder 来进行驱动程序开发和应用程序项目的开发等。Platform Builder 的功能强大,已成为 WinCE 平台下嵌入式操作系统开发和定制的必备工具。Platform Builder 提供的主要开发特性包括^{[34][35]}:

1. 平台开发向导和 BSP 开发向导。开发向导使得开发人员可以很容易的去创建系统平台或 BSP,然后再根据最近的需要来定制。
2. 基础配置。它为自定义操作系统的创建提供一个起点。各种流行的设备都有预置的操作系统基础平台。
3. 特性目录。操作系统可选特性均在特性目录(Catalog)中列出,开发人员可以方便的选择相应的特性来定制操作系统。
4. 内核调试器。内核调试器使得开发人员可以对自定义的操作系统映像进行调试,并能向用户提供有关映像性能的信息。
5. 远程工具。远程工具主要执行同基于 WinCE 的目标设备有关的各种调试任务和收集任务。
6. 仿真器。通过硬件仿真加速和简化了系统的开发,使得用户可以在开发工作站对平台和应用程序进行测试,大大简化了系统开发流程。
7. 自动化的依赖性检查。特性之间的依赖关系是系统自动维护的,这样

- 的好处是开发人员不会因为遗漏某一特性而使得其它特性无法表现。
8. 应用程序调试器。通过应用程序调试器，开发人员可以在自定义的操作系统映像上对应用程序进行调试。
 9. 导出向导。导出向导可以向其他 Platform Builder 用户导出自定义的目录特性。
 10. 导出 SDK 向导。用户可以通过它导出一个自定义的软件开发工具包 (SDK)到特定的开发环境中(如 EVC)去，使得应用程序的开发可以与 BSP 开发几乎同步进行，大大缩短产品开发周期。

本文开发 WinCE 5.0 操作系统，即是采用微软提供的 Platform Builder 5.0 开发工具。Platform Builder 5.0 的界面和 Microsoft Visual Studio 的界面非常相似，熟悉 Windows 程序开发的人员可以很快就能掌握 Platform Builder 的使用。下图图 4-2 是 Platform Builder 5.0 开发工具的界面。

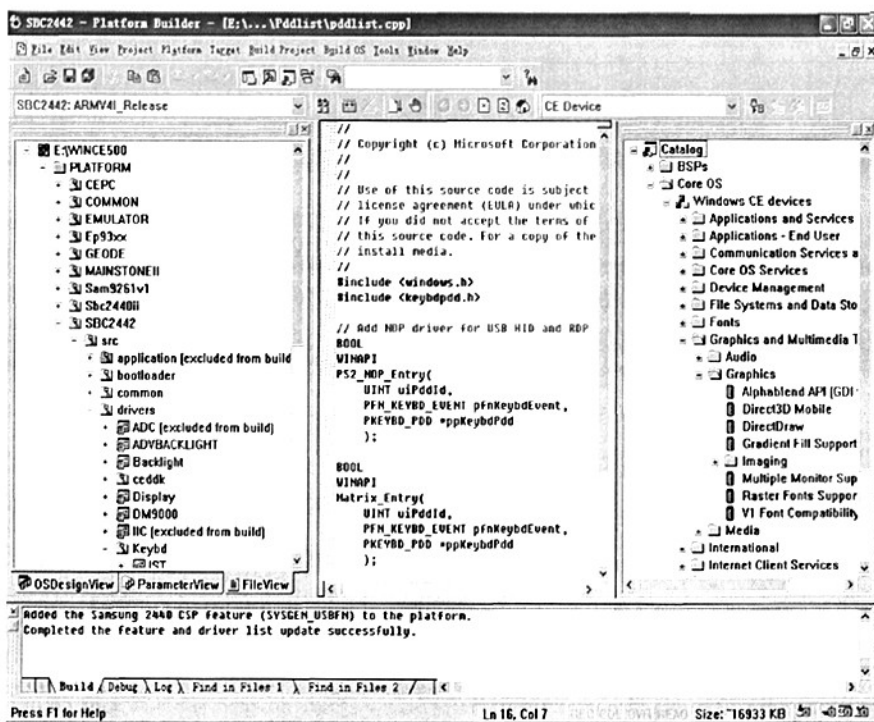


图 4-2 Platform Builder 5.0 开发界面

4.2 WinCE 操作系统 BSP 开发

BSP (Board Support Package) 即板级支持包, 是介于主板硬件和操作系统之间的一层, 主要目的是为了支持操作系统, 使其能够正常运行于硬件平台上, 并支持所有的外围设备。BSP 是针对某个单板而设计的, 它对于用户 (开发者) 是开放的, 用户可以根据不同的硬件需求对其作改动或二次开发, 而操作系统本身仅仅提供了 CPU 内核的无关性。BSP 在系统中的角色, 与 BIOS 在 PC 系统中的地位类似。WinCE 5.0 BSP 通常包含以下几部分: BootLoader、OAL(OEM Adaptation Layer)、设备驱动程序、配置文件等^{[7][36]}。

4.2.1 BootLoader 开发

4.2.1.1 WinCE 下的 BootLoader 特点概述

BootLoader 的功能, 简单来说就是系统加电或复位后, 初始化必要设备, 建立系统的内存空间映射, 然后下载内核镜像到内存, 再跳转到内核镜像的入口地址, 执行内核镜像。WinCE 的 BootLoader 开发应该尽量使用微软所提供的 BLCOMMON、eboot、networkdrivers 等支持库, 这样可以减少开发工作量, 并且能够较容易的享受到微软对支持库升级所带来的好处。作为 Windows CE 操作系统的 BootLoader, 微软推荐这样一个 Boot Loader 应该具备以下的一些功能^[32]

(1) BootLoader 应该存放在掉电非易失性的存储介质中, 如 Flash 中, 这样系统开发者就可以有选择的升级系统的 BootLoader;

(2) BootLoader 需要提供一个错误信息输出和下载状态、进程状态输出的功能, 典型的解决方式是采用一个串口输出;

(3) BootLoader 应该对下载的镜像文件提供校验和计算的功能, 在确保镜像校验和正确之前, 应避免直接更新 Flash 内的镜像, 这样可以保证镜像的完整性, 及之后运行的正确性;

(4) BootLoader 的硬件平台初始化代码应该是能被 BootLoader 和 OAL 共同复用的, 因为两者都需要对硬件平台进行初始化;

(5) BootLoader 应该提供一个选择菜单, 以允许用户进行相应的选择设置, 并能保存这些配置;

(6) BootLoader 应该提供至少一种方式下载操作系统镜像, 并能将其烧写

至 Flash，如串口、USB 或网口等。

为了调试方便通常在函数 OEMPlatformInit() 中加入一些自定义命令的处理，这样便可以通过串口接收命令完成相应的功能。还可以通过 OEMPlatformInit() 实现在 WinCE 启动的时候，显示 logo 图像。

4.2.1.2 BootLoader 在平台上的移植

创建功能完善的 BootLoader，是 WinCE 移植的第一步，也是极为关键的一步。智能终端核心板系统在硬件平台上实现了基于 WinCE 5.0 的 BootLoader，主要工作包括：

1. Startup.s 文件修改

Startup 的正确加载是 WinCE 启动的重中之重，该文件位于 %_WINCEROOT%\Platform\SBC2442\Src\BootLoader\Eboot 文件夹下。主要功能如下：

- (1) 设置 ARM 处理器为监控模式；
- (2) 禁止 IRQ 和 FIQ 中断；
- (3) 禁止 MMU 和指令、数据 Cache；
- (4) 配置 GPIO，通常用来点亮 LED，以方便调试；
- (5) 配置内存管理器，设置刷新频率，使能时钟；
- (6) 设置电源管理寄存器；
- (7) 取得 OEMAddressTable 的物理基地址并把它存在 r0 寄存器中；
- (8) 跳转到 main()。

Startup.s 修改后的部分代码如下：

```

.....
ldr    r0, = GPFCON                ; 初始化 GPIO
ldr    r1, = 0x55aa
str    r1, [r0]
.....
ldr    r0, = INTMSK                ; INTMSK 寄存器
ldr    r1, = 0xffffffff            ; 禁止所有中断
str    r1, [r0]
ldr    r0, = INTSUBMSK             ; INTSUBMSK 寄存器
ldr    r1, = 0x7ff                ; 禁止所有子中断
str    r1, [r0]
ldr    r0, = CLKDIVN               ; 设置设置时钟寄存器
ldr    r1, = 0x3

```

```

.....
add    r0, pc, #g_oalAddressTable - (. + 8)
bl     main                                ; 跳转到 main

```

注意：S3C2442A 有两个 PLL：MPLL 和 UPLL。UPLL 专用于 USB 设备；MPLL 用于 CPU 及其他外围器件。通过 MPPLL 会产生三个部分的时钟频率：FCLK、HCLK、PCLK。FCLK 用于 CPU 核，HCLK 用于 AHB 总线的设备(比如 SDRAM)，PCLK 用于 APB 总线的设备(比如 UART)。因此，上述的 CLKDIVN 时钟寄存器的设置与下面提到的 OEMInitDebugSerial()函数中对 UART 波特率的设置，是有密切联系的。

2. 系统初始化相关函数修改

从 Startup 跳转到 main()后，bootloader 进入 C 语言运行环境。通常需要由 OEM 来编写实现。在 BSP 的 bootloader 代码中，main()的实际控制函数是 BootloaderMain()，它位于

%_WINCEROOT%\PUBLIC\COMMON\OAK\DRIVERS\ETHDBG\BLCOMMON\Blcommon.c 中，其功能主要包括：重新定位 BootLoader 在 RAM 中的位置、解压 .bin 文件、验证校验及跟踪加载过程等。BootLoader 函数中调用的主要函数包括 OEMDebugInit、OEMPlatformInit、OEMPreDownload、DownloadImage 和 OEMLunch，它们被定义在

%_WINCEROOT%\Platform\SBC2442\Src\BootLoader\Eboot\main.c 文件中。

下载函数由 DownloadImage 函数调用，下载函数包括 OEMReadData、OEMShowProcess 和 OEMMapMemAddr，Flash 函数也是有 DownloadImage 函数调用的，当下载的映像需要被写入 Flash 时，DownloadFlash 调用 Flash 函数。Flash 函数包括 OEMIsFlashAddr、OEMWriteFlash、OEMStartEraseFlash、OEMFinishEraseFlash 和 OEMContinueEraseFlash。Flash 函数在

%_WINCEROOT%\Platform\SBC2442\Src\BootLoader\Eboot\flash.c 文件中定义。

表 4-1 列举出了需要修改或填充的函数。

表 4-1

函数名称	功能
OEMDebugInit()	用来初始化调试信息的 I/O 设备。通常调用 OEMInitDebugSerial 函数来初始化串口。
OEMPlatformInit()	执行如时钟设置、驱动和传输设备接口的初始化。提供菜单选择，提示用户选择下一步所要执行的操作，并配置以太网控制器。
KernerRelocate()	将全局变量拷贝到 RAM 中。
OEMPreDownload()	在下载操作系统前执行这个函数，用来设置从网络下载或者跳过下载直接加载 Flash 中的镜像文件。
DownloadImage()	下载操作系统镜像到 RAM 或者 Flash。
OEMLaunch()	负责启动镜像。
OEMReadData()	从远程计算机读取数据。
OEMWriteFlash()	写镜像到 Flash。
OEMStartEraseFlash()	开始擦除 Flash。
OEMContinueEraseFlash()	继续擦除 Flash 工作。

下面以 OEMInitDebugSerial()函数为例具体说明修改内容。

LED 或数码管显示、串口输出、以太网口输出是底层程序开发常用的调试方式。在 Startup.s 的调试过程中，通常只能利用 LED 的开关对程序进行跟踪。尽快实现 OEMInitDebugSerial()函数，利用串口输出进行程序调试，可以极大的提高开发效率。

主要参数设置如下：

```

.....
CLRREG32(&pioPortReg->GPHCON, (3 << 8)|(3 << 10)); // 设置 URAT 引脚，选择 UART0
SETREG32(&pioPortReg->GPHCON, (2 << 8)|(2 << 10));
SETREG32(&pioPortReg->GPHUP, (1 << 4)|(1 << 5)); //禁用 TXD0 和 RXD0 的 pull-up 功能
.....

```

见 WINCE500\PLATFORM\SBC2442\SRCKERNEL\OAL\debug.c 文件。

3. 配置文件修改

WinCE 的内存（包括 SDRAM 及 FLASH）配置包含两个方面：源代码中的定义及 bib 系统配置文件中的定义。config.bib 中一般会将内存定义成用途不同的若干段。应修改 boot.bib，使其与 config.bib 中的内存配置相匹配^[37]。

4.2.2 OAL 开发

4.2.2.1 OAL 简介

OAL 的全称是 OEM Adaption Layer, 即原始设备制造商适配层。它位于操作系统的内核与硬件之间, 是连接系统与硬件的枢纽, 具有初始化设备、引导操作系统以及抽象硬件功能等作用。

OAL 层的首要任务是加载内核, WinCE 内核的启动顺序如下:

1. CPU 执行引导向量, 跳转到硬件初始化代码, 即 Startup;

2. 在 Startup 代码部分完成最小硬件环境初始化后跳转到 KernelStart()函数, 来对内核进行初始化;

3. Kernelstart()函数调用 OEMInitDebugSerial()完成对调试串口的初始化, 调用 OEMInit()函数来完成硬件初始化工作以及设置时钟、中断, 调用 OEMGetExtensionDRAM()函数来判断是否还有另外一块 DRAM。

至此, 内核加载完毕。

4.2.2.2 OAL 在平台上的移植

OAL 主要移植工作如下:

1. startup.s、OEMInitDebugSerial()

这部分可以与 BootLoader 共享使用, 移植方法参考本文 4.2.1.2 节。需要注意的是, OAL 的 startup.s 文件位于

WINCE500\PLATFORM\SBC2442\SRC\KERNEL\OAL 目录下, 并且由于此前 Bootloader 已将硬件初始化完毕, OAL 的 startup.s 文件仅仅将 OEMAddressTable 传递给内核, 并直接跳转到内核初始化函数 KernelStart()。

2. OALIntrInit()

OALIntrInit()函数用来初始化中断映射表, 在物理中断号和系统中断号之间建立映射关系, 然后清除外部中断、内部中断等。WinCE 操作系统用一个 ISR(interrupt service routine)来处理所有的 IRQ 请求。当一个中断发生时, 系统先调用内核 ISR, 禁用所有具有相同优先级和较低优先级的中断, 然后调用已经注册的 OAL ISR 程序。WinCE 的中断实际上是由 ISR 和 IST 两部分组成的。其中 IST 包含在驱动程序中, 而 ISR 包含在 OAL 层中。所以, 要支持某个硬件的中断, 首先必须从 OAL 层为其作好准备, 这一点在本文 5.4.2 节有详细的描述。

3. OALTimerInit()

此函数用来完成 S3C2442A 系统时钟的初始化工作，通常是在 OEMInit()函数中调用。具体包括初始化 OAL 层变量 g_oalTimer，初始化内核变量 curridlehigh、curridlelow、idleconv，初始化内核函数指针，初始化硬件时钟定时器。

4.2.3 驱动程序开发简介

WinCE 驱动程序是操作系统与硬件交互的方式，是连接硬件与操作系统之间的桥梁，操作系统通过它对硬件设备实现操作和管理。驱动程序的开发是 BSP 开发过程中的一个重要环节，根据不同的系统要求，需要为目标板开发各种驱动程序^[35]。

在 WinCE 5.0 中，驱动程序模型分为两种：本机驱动和流接口驱动。

(1) 本机设备驱动适于集成到基于 Windows CE.net 平台的设备。这些设备驱动程序是一些硬件所必需的，是由原始设备制造商创建的，用以驱动如键盘、触摸屏、音频设备等，往往在设备售出后就不会再更换，如通用 LED 驱动、电源驱动、键盘驱动和显示驱动等都是本机设备驱动。对于本机设备驱动程序，Platform Builder 提供了一些驱动程序样本，目的是为了更方便开发人员快速开发出自己的驱动程序。当 Win CE 系统启动时，本地设备驱动程序将被加载到系统的内存中。本地驱动程序的开发分为分层驱动和单片驱动程序^{[31][38]}。分层驱动要利用微软提供的与应用程序通信的上层，称为模块驱动程序层 MDD (Model Device Driver)。MDD 层通过设备驱动程序接口 DDI (Device Driver Interface) 与应用程序通信，开发驱动程序通常不修改 MDD 层，主要关心与具体硬件相关的下层，依赖平台的设备驱动程序层 PDD (Platform Dependent Driver)，PDD 层通过设备驱动服务接口 (Device Driver Service Provider Interface) 直接管理硬件^[31]。

(2) 流接口设备驱动程序 (指可安装的启动程序) 可以由第三方生产商提供，以支持添加到系统中的设备。Windows CE 下的设备驱动程序在与应用程序相同的保护级上工作。当系统启动时，大多数驱动程序是由设备管理进程

(DEVICE.EXE) 加载的，所有这些驱动程序将共享同一个进程地址空间。流接口驱动程序是动态连接库，是由一个叫做设备管理程序的特殊应用程序加载、管理和卸载，与具有单独目的接口的内部驱动程序相比，所有流接口驱动程序

使用同一个接口并调用同一个函数集流接口函数^{[31][39]}。对于每个流接口驱动程序来说，其所要求的入口点用来实现标准文件 I/O 函数和电源管理函数，这些函数由 Windows CE 操作系统的内核使用。

本文将在第 5 章对主要驱动的开发过程进行介绍。

4.3 系统平台构建过程

对于同一个嵌入式操作系统，不同的硬件设备，它的操作系统镜像是不同的，但建立该镜像的操作过程及环境是相同的。下面通过使用 Platform Builder 来描述本项目的 WinCE 平台构建过程。具体如下：

1. 运行 Platform Builder 5.0 开发软件。选择 Platform Builder 5.0 的菜单栏上的 File->New Platform 打开新建平台向导，如图 4-3 所示。

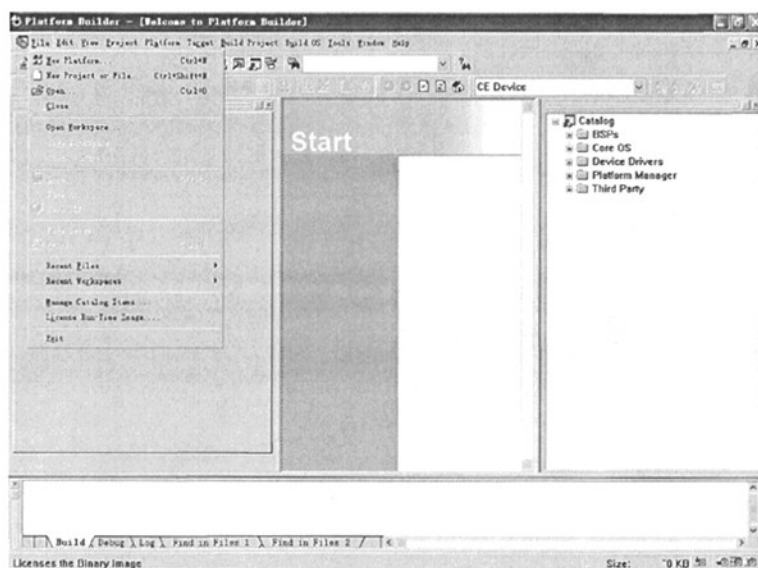


图 4-3 运行操作系统镜像向导

2. 向导启动后先显示如图 4-4 所示的欢迎屏幕，直接点击 Next 按钮，进入下一步。

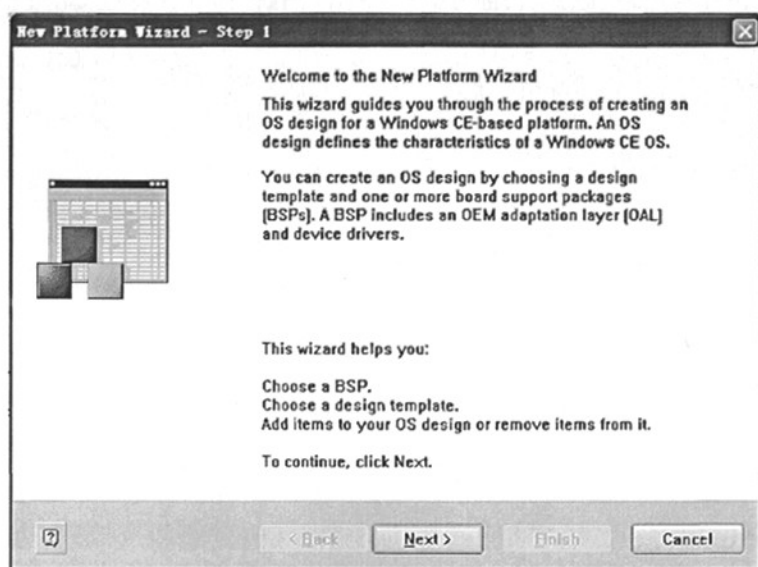


图 4-4 定制内核欢迎屏幕

3. 在出现的向导中填入定制平台的名称和选择相应的板级支持包(BSP)。在本项目中，选用针对 S3C2442A 处理器的 BSP，如图 4-5 所示。

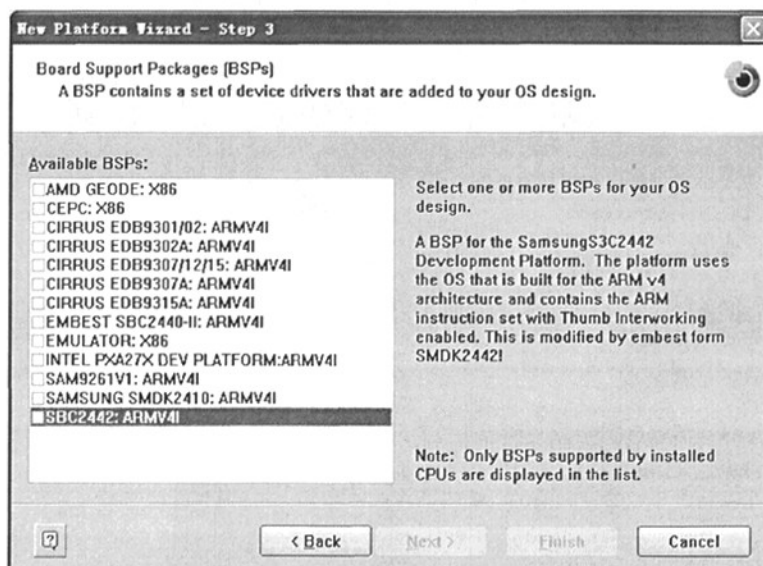


图 4-5 选择相应的 BSP

4. 完成上述步骤之后, 点 Next 按钮。接下来会出现一个如图 4-6 的列表框。列表框中, 有多个 WinCE 已经设定的基本结构供选择。如果设计人员设定的产品与列表中保存的基本配置一样, 那么直接选择相应项就可以完成基本的设定。

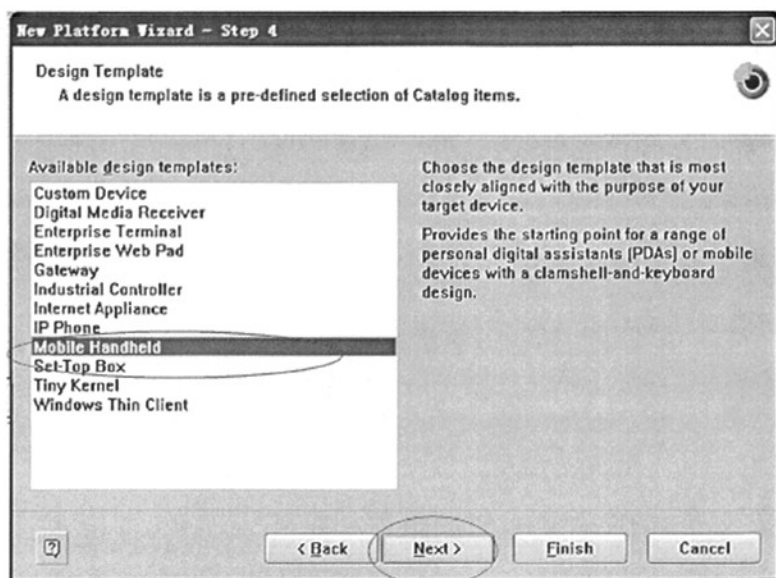


图 4-6 选择操作系统镜像列表框

当然, 也可以自定义配置。在向导建立完成后的 Catalog 窗口中可以右键添加的方式将一些特征组件添加, 使得定义的操作系统镜像包含某种功能和特性。这些特征组件包括:

- 开发应用程序和服务程序的基础类库(Application and Service Development)
- 供最终用户使用的应用程序(Applications-End User)
- 操作系统核心服务(Core OS Services)
- 网络通信服务和网络选项(Communication Services and Networking)
- 设备管理(Device Management Client)
- 文件系统与数据存储(File System and Data Store)
- 字体(Fonts)
- 图形和多媒体技术(Graphics and Multimedia Technologies)
- 国家支持(International)

- Internet 服务相关的程序(Internet Client Services)
 - 安全设置(Security)
 - 操作系统外壳与用户接口(Shell and User Interface)
 - IP 电话语音服务(Voice over IP Phone Services)
 - WinCE 错误报告(Windows CE Error Reporting)
5. 对以上配置选择后, 就完成操作系统定制向导的操作如图 4-7 所示。

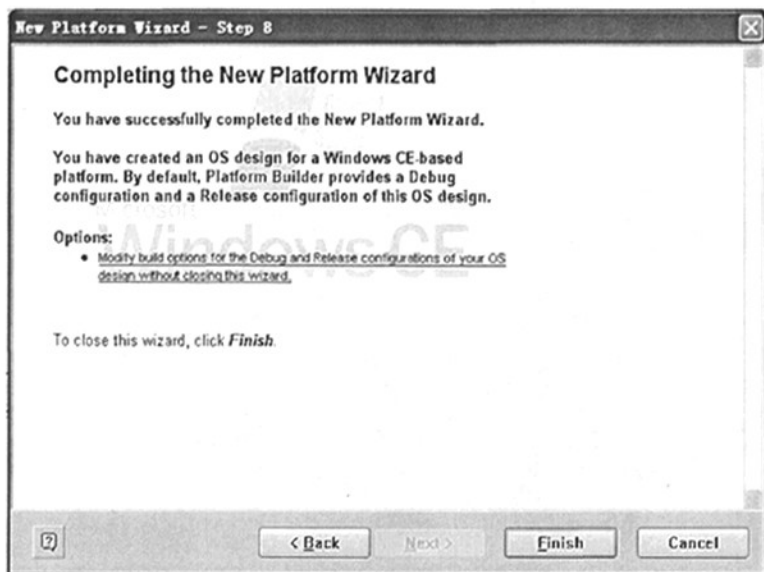


图 4-7 完成定制

6. 按向导完成定制后就可以根据对应的目标设备要求对这个操作系统镜像进行修改, 包括添加或删除特征等。以上工作完成后就可以编译了, 操作系统编译成功后界面如图 4-8 所示。

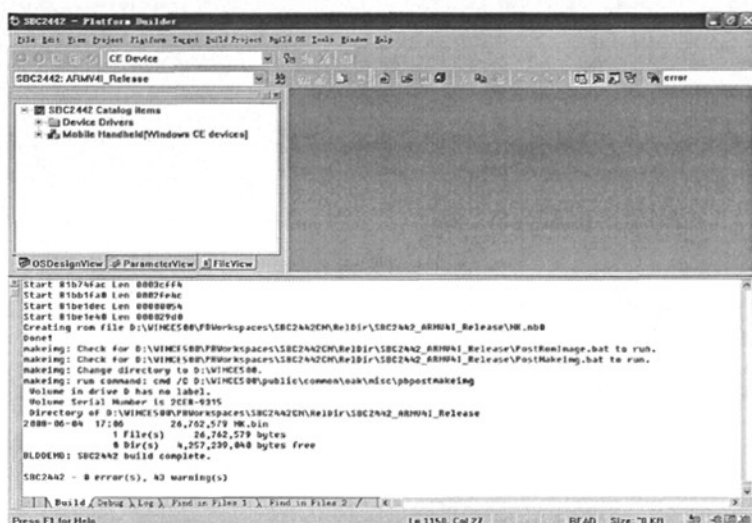


图 4-8 编译成功后的界面

7. 编译成功后, Platform Builder 会生成二进制启动文件 Eboot.nb0 以及系统镜像 NK。使用仿真器将 Eboot.nb0 下载到 RAM, 再用 Samsung 公司提供的工具 DNW.exe 通过 USB 线或网线将 NK 下载到开发板里面。镜像 NK 下载界面如图 4-9 所示。

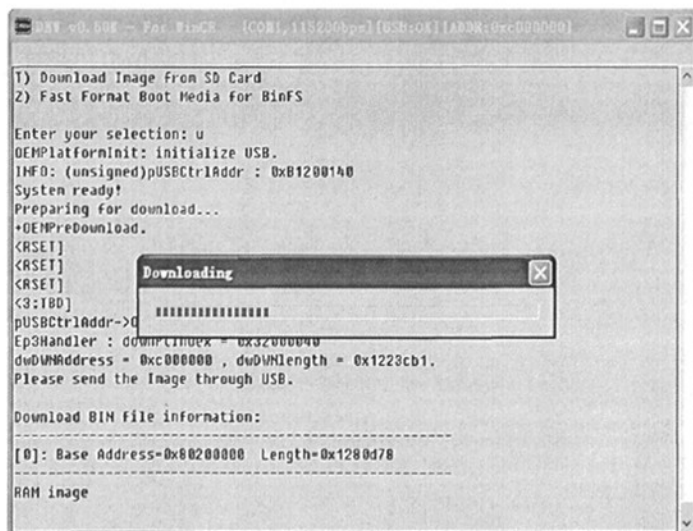


图 4-9 镜像 NK 下载界面

8. 下载完成后, WinCE 系统被加载后启动, 界面如图 4-10 所示。

```

DHW v0.50K - For WinCE [COM1,115200bps][USB:x][ADDR:0xc000000]
Serial Port USB Port Configuration Help
DEM: Not cleaning user profiles
OEMIoControl: Unsupported Code 0x10100d0 - device 0x0101 Func 52
OEMIoControl: Unsupported Code 0x10100f8 - device 0x0101 Func 62
INFO: CPdd2440Serial10 (37240).
INFO: CPdd2440Serial12 (38730).
parallel port process attach
[LPT_Init]: entry Drivers\Active\06.
+LptPortInit:Initialize*****successful*****
GPFCOH = 180000 GPCCOH = 180050 GPJCOH = ffff GPCCOH = 2f2195h GPCDAT = 2f214f0
<Davicom 9000h driver v3.04 for WinCE 5.0>
S3C2440usbFn!MapRegisterSet: GPIO VirtualCopy Succeeded, pUMen:5e0000
S3C2440ISP::S3C2440ISP
OEMIoControl: Unsupported Code 0x10100fc - device 0x0101 Func 63
H3DIF_Matrix_Entry: Initialize Matrix ID 1
Ps2Keybd::Initialize
Ps2Keybd::Initialize Done
ReadRegDWORD(): HARDWARE\DEVICEMAP\KEYBD -> IObase is 0xb1600000
ReadRegDWORD(): HARDWARE\DEVICEMAP\KEYBD -> IntrBase is 0xb0a00000
**Ps2Keybd::KeybdPowerOn
--Ps2Keybd::KeybdPowerOn
H3DIF_Matrix_Entry: Initialization complete
ReadRegDWORD: RegQueryValueEx(Priority256) returned 2
ReadRegDWORD(): HARDWARE\DEVICEMAP\KEYBD -> Priority256 is 0x0
    
```

图 4-10 运行信息输出图

串口显示以上信息后, 就表明 WinCE 系统已经在目标板上启动了。

第5章 智能终端模块的驱动实现

系统各模块的功能得以体现是与其驱动程序切切相关的。设备驱动程序就是将操作系统和设备连接起来，使得操作系统能够识别设备，并为应用程序提供服务。

目前，WinCE 提供了 4 种设备模型，其中本地设备驱动程序模式和流接口驱动程序模式是唯一提供给 WinCE 驱动程序的开发方法。另外两个外部驱动模式也可应用于其它操作系统，分别是 USB 设备驱动和 NDIS(Networking Driver Interface Specification)网口设备驱动。

下面部分对主要驱动的开发过程进行介绍。

5.1 显示驱动开发

5.1.1 LCD 驱动开发

WinCE 中显示驱动程序应归于内部驱动程序，因为驱动程序和图形、窗口和事件系统(GWES) 间存在定制接口。GWES 模块或设备管理器可以被它们的父进程直接加载和调用。WinCE 的显示设备驱动程序接口(DDI) 是Windows NT 显示设备驱动程序接口(DDI) 的子集^[40]，在Windows CE 2.0 及其以上版本中，GDI 调用显示其驱动程序并将显示器驱动程序写到物理的显示设备上。所有基于Windows CE 的显示器驱动程序必须实现一套显示的DDI 函数，这些函数用来初始化显示器驱动程序并画到显示器上

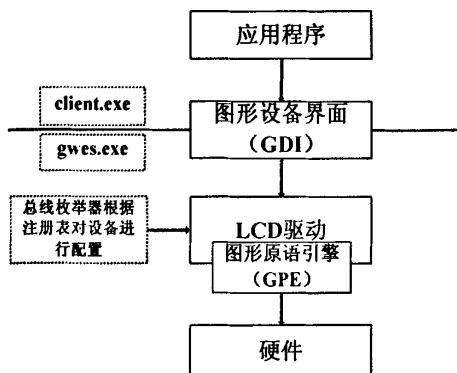


图 5-1 WinCE5.0 LCD 驱动程序结构

如图 5-1, 当应用程序需要向 LCD 控制器发送数据时, 它调用由 `coredll.dll` (WinCE 5.0 专门用来提供操作系统核心函数的动态链接库文件)提供的 GDI(图形设备接口)函数^[32]。所有的图形绘制与窗口调用操作均被发往 `gwes.exe`。

LCD 驱动程序需要实现 DDI (显示设备驱动程序) 函数。DDI 导出 `DrvEnableDriver()` 函数, 而 `DrvEnableDrive()` 通过返回指向 27 个函数的指针来实现对这些函数的调用。这些函数可以随意命名, 但必须符合头文件 `Winddi.h` 中的原型说明。

除了 DDI 函数, 大部分 WinCE 的 LCD 驱动程序使用了 GPE 的 C++ 类, 这些类提供了一些最基本的代码, 帮助开发者实现自己的驱动程序。它负责处理 LCD 显示驱动程序接口(DDI)层与图形绘制之间的所有通信。GPE 类的定义在 `%_WINCEROOT%\WINCE500\PUBLIC\COMMON\OAK\INC\` 目录下的 `Gpe.h` 文件中。本文讨论的 S3C2442A 的 LCD 驱动正是利用 GPE 实现的。

Samsung 提供了 S3C2442A 的 LCD 驱动程序源代码, 但是这些源代码需要根据相应的硬件平台(与使用的 LCD 屏相关)进行修改。

本课题中, 根据设备的具体配置, 作了以下修改:

1. 驱动程序文件: `s3c2442disp.h` 和 `s3c2442disp.cpp`

`s3c2442disp.h` 中, 需要定义分辨率、`CLKVAL_CSTN`、`FRAMEBUF_DMA_BASE`、`MVAL_USED`、`MVAL` 等重要参数, 还可以根据需要定义基本 IO 口, 比如用哪个口作 `PWR_EN`, 哪个口作背光控制等。

```
#define LCD_VBPD          ((33-1)&0xff)
#define LCD_VFPD          ((10-1)&0xff)
#define LCD_VSPW          ((2-1)&0x3f)
#define LCD_HBPD          ((48-1)&0x7f)
#define LCD_HFPD          ((16-1)&0xff)
#define LCD_HSPW          ((96-1)&0xff)
#define INVCLK             (0)
#define PWREN              (0)
#define ENLEND             (0)
#define VCLK               (2)
```

`s3c2442disp.cpp` 中相关的寄存器设置应符合实际的硬件配置:

```
static BOOL InitializeLCDRegisters(DWORD dwPhysicalFrameBase)
{
    ..... // LCD 相关控制寄存器设置
    s2442LCD->LCDCON1 = (CLKVAL_CSTN<<8)|(MVAL_USED<<7)|(2<<5)|(3<<1)|0;
    s2442LCD->LCDCON2 = (0<<24)|(LINEVAL_CSTN<<14)|(0<<6)|(0<<0);
    s2442LCD->LCDCON3 = (WDLY_CSTN<<19)|(HOZVAL_CSTN<<8)|(LINEBLANK_CSTN<<0);
```

```

s2442LCD->LCDCON4 = (MVAL<<8)|(WLH_CSTN<<0);
s2442LCD->LCDCON5 = (1<<1);
s2442LCD->LCDSADDR1=((FRAMEBUF_DMA_BASE>>22)<<21)|M5D(FRAMEBUF_DMA_BA
SE>>1);
s2442LCD->LCDSADDR2=M5D((FRAMEBUF_DMA_BASE+(LCD_XSIZE_CSTN*LCD_YSIZE_
CSTN))>>1);
s2442LCD->LCDSADDR3=((LCD_XSIZE_CSTN*1-LCD_XSIZE_CSTN)/1)<<11)|(LCD_XSIZE_
CSTN/2);
.....
}
.....
WORD TempBuffer[641][480]; // 与分辨率相关
S3C2442DISP::S3C2442DISP (void)
{
.....
m_nScreenWidth = 640; // 水平尺寸
m_nScreenHeight = 480; // 垂直尺寸
m_colorDepth = 16;
m_cbScanLineLength = m_nScreenWidth * 2;
m_FrameBufferSize = m_nScreenHeight * m_cbScanLineLength;
.....
}

```

2. 配置文件：platform.bib、platform.reg、source

WinCE 中，驱动程序都是以 dll 即动态链接库的形式存在的。配置文件则用来确定 LCD 驱动程序最后生成的动态链接库的文件名、存放目录等，并把这些重要信息放在注册表文件中提供给操作系统，以便操作系统对驱动程序进行管理。

(1) platform.reg 中 LCD 驱动相关部分：

```

[HKEY_LOCAL_MACHINE\Drivers\Display\S3C2410\CONFIG]
"DisplayDll"="s3c2442disp.dll" ; 驱动程序名字,要与 source 文件
"LCDVirtualFrameBase"=dword:ac100000 ; 中的一致
"LCDPhysicalFrameBase"=dword:30100000
[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
"Display"="s3c2442disp.dll"
[HKEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]
"Candidate10"="Drivers\Display\S3C2442\CONFIG"

```

(2) platform.bib 文件中相关部分：

```

s3c2442disp.dll $(FLATRELEASEDIR)s3c2442disp.dll NK SH

```

(3) source 文件：

```

TARGETNAME=S3C2442DISP ; 要生成的驱动程序名字
RELEASETYPE=PLATFORM

```

TARGETTYPE=DYNLINK
 DEFFILE=S3C2442DISP.DEF

; 驱动程序文件类型

修改完成后，在 Platform Builder 5.0 中重新编译驱动程序，然后将新生成的内核镜像下载到开发设备中，WinCE 5.0 的桌面及 LOGO（图标）正常显示，说明 LCD 驱动运行正常。

说明：当用户选择或更换不同的 LCD 屏时，只需按照新的 LCD 时序完成步骤 1 即可。

5.1.2 LCD 转 VGA 的驱动开发技术

VGA 接口是与显示器进行通信的接口，其主要参数的工业标准是：像素输出频率为 25.175MHz；行频(线频率)为 31.469kHz；场频(刷新率)为 59.94Hz；分辨率为 640×480。S3C2442A 的 LCD 控制器输出的 RGB 格式的视频信号数据必须通过 CH7005C 转换后，将数据信号转换成模拟信号，配合行、场同步信号即能使其在 VGA 显示器上显示。

VGA 的标准参考显示时序如下图 5-2 所示^[41]。行时序和帧时序都需要产生同步脉冲、显示后沿、显示时序段和显示前沿 4 个部分。由图可知，VGA 接口的行同步信号、场同步信号和 LCD 扫描式接口的同步信号是一致的。所以只需要通过设置 S3C2442A 的 LCD 控制器中的寄存器就能够产生出 16.67ms 的场同步信号的 31.77us 的行同步信号。

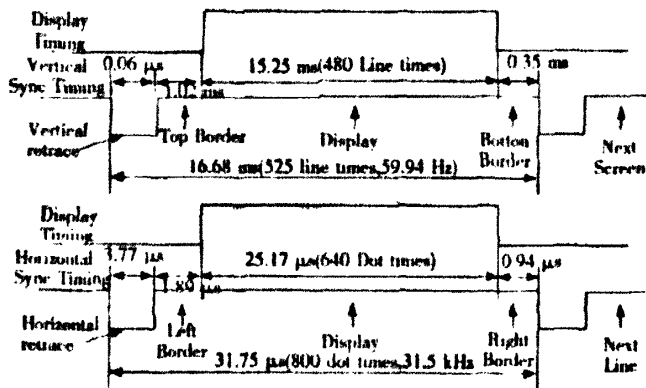


图 5-2 VGA 的显示时序图

因此，VGA 接口的驱动程序，主要是初始化 VGA 接口的相关电路。这些初始化工作包括：设置 LCD 寄存器、初始化 IIC 接口、初始化高速 VGA 转换芯

片。

1. 设置 LCD 寄存器

LCD 寄存器设置与上小节(5.1.1 节)中所述 LCD 驱动开发完全相同, 只是定义的部分参数不同。现列出如下:

```
#define LCD_VBPD          ((15-1)&0xff)
#define LCD_VFPD          ((4-1)&0xff)
#define LCD_VSPW          ((1-1)&0x3f)
#define LCD_HBPD          ((27-1)&0x7f)
#define LCD_HFPD          ((15-1)&0xff)
#define LCD_HSPW          ((55-1)&0xff)
#define INVCLK            (1)
#define PWREN             (1)
#define ENLEND            (1)
#define VCLK              (2)
```

2. 初始化 IIC 接口

由于对 CH7005C 芯片的初始化需要通过 IIC 接口, 因此, 配置 VGA 转换芯片之前, 需对 IIC 进行初始化。以下是初始化 IIC 接口的代码:

```
s2442IOP->GPEDN |= 0xc000;           //IO 电阻上拉使能
s2442IOP->GPECON &= ~(0xa<<28);
s2442IOP->GPECON |= (0xa<<28);       //GPE15:IICSDA , GPE14:IIC_SCL
s2442IIC->IICCON = 0x00;             //IIC 控制寄存器设置
//使能 ACK, Prescaler IICCLK=PCLK/512, 使能中断, 中断时钟 Tx clock=IICCLK/16
s2442IIC->IICCON |= (1<<7)|(1<<6)|(1<<5)|(0x0f);
s2442IIC->IICADD = 0x10;              //2442 slave address = [7:1]
s2442IIC->IICSTAT = 0x10;            //IIC 总线数据输出使能(Rx/Tx)
```

3. 初始化高速 VGA 转换芯片

配置完 IIC 总线后, 先对 CH7005C 芯片进行设定, 给内部的寄存器分配地址并对一些寄存器赋值。以下给出 VGA 显示的测试程序, 即本文的显示器上电显示程序。

```
CH7005C_RegWrite(PMR,1<<6)| CH7005C_RESET| NORMAL); //设置电源管理模式
CH7005C_RegWrite(DMR, 0x62); //设置显示模式
CH7005C_RegWrite(CM, CFRB | M_S | MCP); //设置主从及时钟模式
CH7005C_RegWrite(BLR,0); //设置亮度
CH7005C_RegWrite( SPR,DES|SYO|VSP|HSP); //设置同步极性
CH7005C_RegWrite( PLLC,0); //设置 PLL
DisplayImageDemo();
```

其中 CH7005C_RegWrite 函数是对 CH7005C 芯片的内部寄存器进行调用和

赋值。通过对上述寄存器的赋值，实现对 CH7005C 芯片的设置。本系统中 VGA 显示输入数据定为 16bit 的 RGB 格式，显示分辨率为 640×480。DisplayImageDemo 函数则是显示一幅 LOGO 图片。

5.2 并口打印机驱动开发

在流接口驱动程序^[42]中，驱动程序负责把外设抽象成一个文件，而应用程序则使用操作系统提供的文件 API 对外设进行访问。通过流接口驱动程序对硬件设备的访问涉及如图 5-3 所示五个模块^[7]。其中应用程序与流接口驱动由用户编写，File System 与 Device Manager 是 WinCE 操作系统提供的组件，用户无法修改。

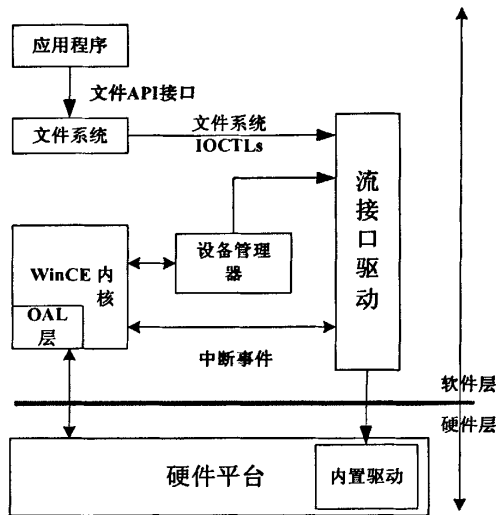


图 5-3 流接口驱动的结构

首先，应用程序利用文件 API 对设备进行访问，文件 API 被操作系统转发到 File System 进程；然后 File System 发现是对设备的操作，就把执行交给 Device Manager 处理；接着 Device Manager 根据具体的要求，调用不同的流接口程序中提供的接口；最终，驱动程序负责与硬件交互。

并口打印机驱动即采用流接口驱动实现。驱动完成以下功能：根据并口微型打印机时序向并行口发送数据，采用查询方式检测打印机状态和进行错误处理。该驱动在微软提供的打印机驱动基础上修改完成，开发并口打印机流接口驱动

包含如下几个方面的工作：

1. 主要配置文件的编写

(1) source 文件设置要生成的文件类型、目标库、源文件等，设置如下：

```
SYNCHRONIZE_DRAIN=1
TARGETNAME=parallel // 要与注册表中的加载名字一致
RELEASETYPE=PLATFORM // 生成文件要存放的位置
TARGETTYPE=DYNLINK // 生成 DLL 动态链接库文件
DEFFILE=parallel.def // DLL 导出文件名
DLENTY=DllMain // DLL 入口函数
TARGETLIBS= \
    $_COMMONSDKROOT)\lib\$_CPUINDPATH)\coredll.lib \
    $_SYSGENOAOKROOT)\lib\$_CPUINDPATH)\ceddk.lib \
SOURCES= \
    mdd.c \ // 要编译的驱动程序文件
    Lptlib.c \
```

(2) parallel.def 文件提供流驱动的接口，设置如下：

```
LIBRARY parallel
EXPORTS
    LPT_Init // 以下函数具体内容参考下文
    LPT_Deinit
    LPT_Open
    LPT_Close
    LPT_Read
    LPT_Write
    LPT_Seek
    LPT_PowerDown
    LPT_PowerUp
    LPT_IOControl
```

(3) platform.bib 文件，设置将驱动程序打包到操作系统镜像中：

```
IF BSP_NOPRINTER !
    parallel.dll          $_FLATRELEASEDIR)\parallel.dll          NK SH
ENDIF BSP_NOPRINTER !
```

(4) platform.reg 文件，用来加载驱动：

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Parallel] // WinCE 启动时，设备管理器搜寻
// 此键值下的每个驱动进行加载

"DeviceArrayIndex"=dword:0
"Prefix"="LPT" // 流驱动前缀名
"Index"=dword:1 // 应用程序 API 调用索引
"Dll"="parallel.Dll" // 要加载的驱动程序
"Order"=dword:0 // 同目录下驱动加载顺序
"DevConfig"=hex: 10,00, 00,00, 05,00,00,00, 10,01,00,00, 00,4B,00,00, 00,00, 08, 00, 00,
00,00,00,00 // 驱动类型
```


2. 主要接口函数的实现

LPT_Init()函数主要完成为寄存器分配内存空间，并调用并口初始化函数。

LPT_Write()函数完成该驱动的核心功能——打印数据。该函数基于微型打印机的操作时序实现，如下图 5-4 所示(其中 STB、BUSY、ACK 分别与 GPIO 相连)。

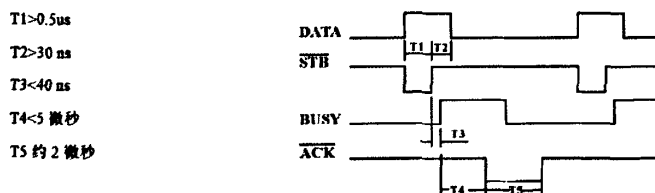


图 5-4 微型打印机时序图

```

ULONG
LPT_Write(PPortInformation pPortObj, /*@parm [IN] HANDLE returned by LPT_Open.*/
          PCHAR pSourceBytes, /*@parm [IN] Pointer to bytes to be written.*/
          ULONG NumberOfBytes /*@parm [IN] Number of bytes to be written.*/
        )
{
    ..... //初始化结构体
    EnterCriticalSection( &(pPortObj->lptCrit)); //进入临界区
    .....
    while (pPortObj->pPortData.QOutCount && CheckPrinterStatus (pPortObj)) //检查打印机状态
    {
        OutByte (uDataPort, *pSourceBytes++); //调用写函数
        pPortObj->pPortData.QOutCount --;
    }
    LeaveCriticalSection(&(pPortObj->lptCrit)); //离开临界区
    .....
}
    
```

其中，OutByte 函数实现 LPT_Write 的主要功能，包括在打印数据之前先检测打印机状态(CheckPrinterStatus 函数)对错误状态进行处理，并将结果以消息方式通知操作系统，如果打印机状态正常则将 1 字节数据写入指定地址(函数的第一个参数)，写入完成后检测打印机返回状态信息(CheckPrinterStatus 函数)。其程序结构流程如下图 5-5 所示。

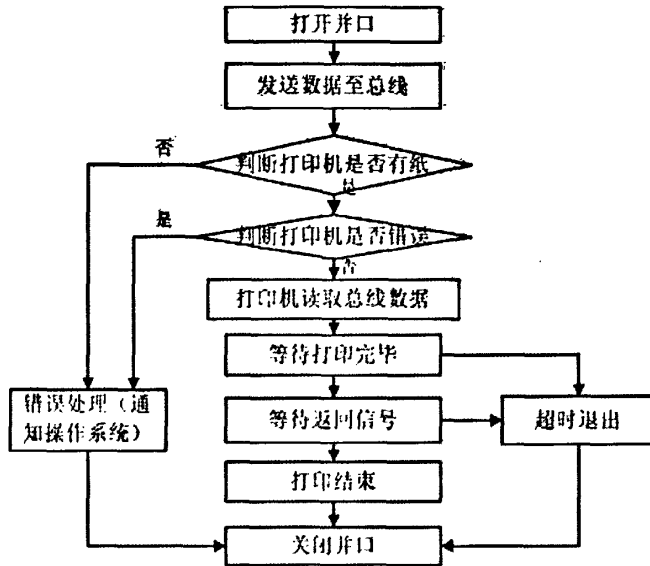


图 5-5 写数据函数结构流程图

5.3 矩阵键盘驱动开发

对于许多基于 WinCE 的设备来说，键盘是一个很重要的输入设备。WinCE 给出了一个设备独立的键盘模型，可以支持各种键盘。如下图 5-6 所示，驱动结构分为多层：键盘布局管理器和当前平台输入语言属于 MDD 层，由 WinCE 操作系统实现；矩阵键盘平台相关驱动和矩阵键盘设备布局属于 PDD 层，与硬件设备密切相关，由用户实现；图形事件窗口子系统(GWES)负责对键盘驱动进行调用和管理。

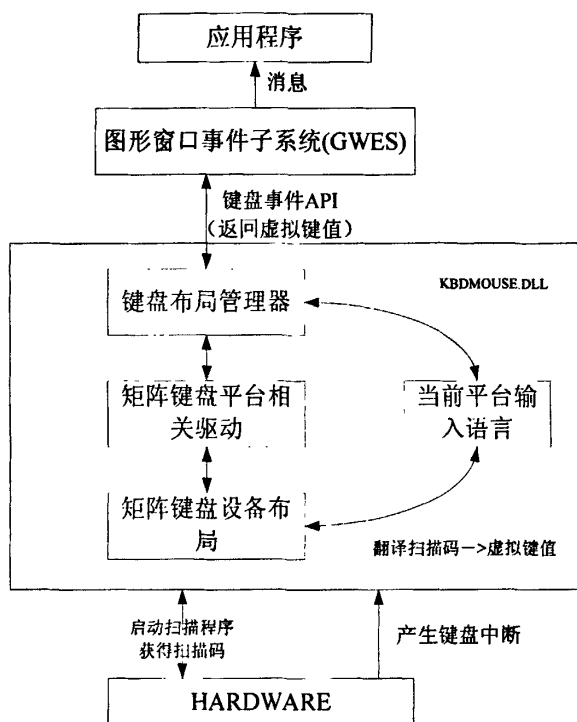


图 5-6 键盘驱动结构图

WinCE系统启动后，首先启动GWES，GWES调用键盘驱动MDD层的初始化接口函数KeybdDriverInitializeEx(),同时创建事件处理线程Event Thread(对应函数入口为: KeybdEventThreadproc(), 位于WINCE500\PUBLIC\COMMON\oak\drivers\keybd\laymgr\Laymgr.cpp文件中); 然后进入键盘驱动PDD层的入口函数: Matrix_entry(), 对硬件平台进行相关初始化。其工作包括: 首先对IO端口初始化, 然后启动中断服务线程IST(对应的函数入口为:MatrixKeybd.IsrThreadProc(), 位于SBC2442\Src\Drivers\keybd\kbdcommom\s3c2442kbd.cpp文件中), 该IST用于处理键盘事件, 主要工作是向系统申请软硬件资源, 并调用ISR扫描矩阵返回扫描码然后再将扫描码转化为虚拟键值。初始化完毕后, 无键盘事件时, IST线程被阻塞。一旦有按键被按下, 某列GPIO口则会产生下降沿触发的中断, 进入中断服务例程: OEMInterruptHandler。中断服务例程返回逻辑中断号, 由内核进行逻辑中断号与相关中断事件的关联, 从而唤醒阻塞的IST线程。键盘矩阵扫描和读取键值等对键盘按下事件的处理均通过IST线程来实现。

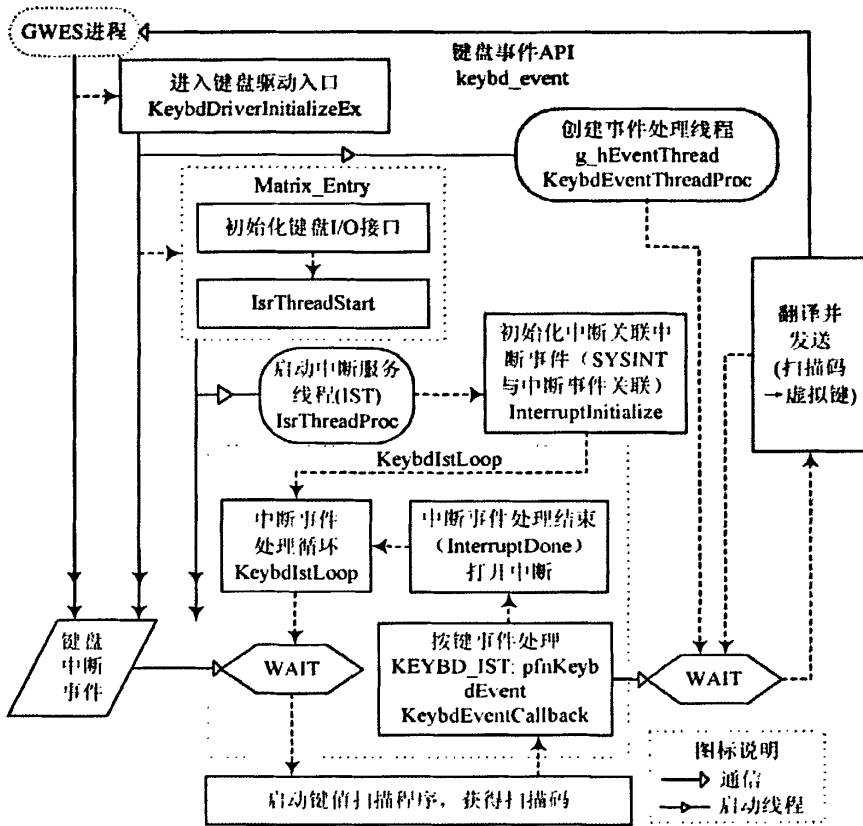


图5-7 键盘驱动流程图

矩阵键盘驱动开发，在Samsung提供的标准BSP包下进行修改，包括以下几个方面工作：

1. 扫描码表与虚拟码表数组定义

首先需要按照自己的要求和物理键盘的布局确定SC2VKMap[i][j]二维数组，该数组为包涵两个元素的结构体，分别为扫描码和虚拟码。其中第i行第j列元素存放键盘第i行第j列按键对应的扫描码和虚拟码。本文所设计的5×5矩阵键盘中，i和j的值均为5。通过这个二维数组的设定，就可以在程序运行的过程中，当检测有按键被按下时，通过KeydpDd_GetEventEx2函数得到的扫描码，然后通过查表方式得到扫描码所对应的虚拟按键值。

2. IO口初始化工作

配置相关IO口功能，以及中断触发方式。填充s3c2442kbd.cpp文件中的MatrixKeybd.Initialize函数。

3. 中断处理过程

(1) 按键信号中断处理

按键信号中断处理是键盘中断处理过程的第一步，属于按键后对中断的响应过程。该过程的实现包括如下几个步骤：首先在Interrupt Initialize 中初始化中断(关于中断的初始化与映射，可参见在本文5.4.2节),然后在KeybdIstLoop中调用 WaitForSingleObject 监听中断，之后在中断处理函数KeybdPdd_Get Event Ex2 对中断进行处理。关键代码如下：

```

// 创建键盘事件、初始化中断
m_hevInterrupt = CreateEvent ( NULL , FALSE , FALSE ,NULL);
Interrupt Initialize ( g_dwSysInt r_ Keybd , m_hevInterrupt ,NULL ,0)
// 关联中断事件、系统中断号、中断处理函数、键码虚拟码
// 转换函数
keybdIst . hevInterrupt = m_hevInterrupt ;
keybdIst . dwSysInt r_ Keybd = g_dwSysInt r_ Keybd ;
// 系统中断号
keybdIst . uiPddId = v_uiPddId ; // PDD 标识
keybdIst . pfnGet KeybdEvent = KeybdPdd_ Get Event Ex2 ;
// 键盘中断处理函数
keybdIst . pfnKeybdEvent = v_pfnKeybdEvent ;
// 调用 WaitForSingleObject 监听中断
KeybdIstLoop ( &keybdIst );
    
```

(2) 键盘中断处理函数

在中断处理函数Keybd2Pdd_ Get Event Ex2 中，主要完成键盘扫描码的读取，数据流程如图5-8 所示。

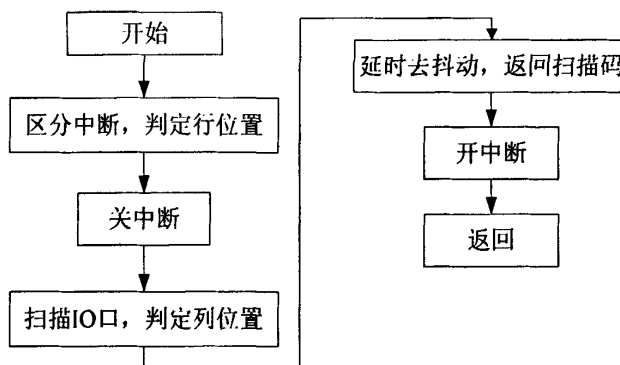


图5-8 读取扫描码函数流程

5.4 DM9000A 网口驱动开发

5.4.1 NDIS 驱动模型

以太网卡驱动属于 Windows CE 提供的网络驱动接口规范(Network Driver Interface Specification,NDIS)中的一部分。NDIS 的主要目的就是为 NIC(网络接口卡, Network Interface Cards)制定出标准的 API 接口。

在 WinCE 中, 整个网络架构体系如下图 5-9:

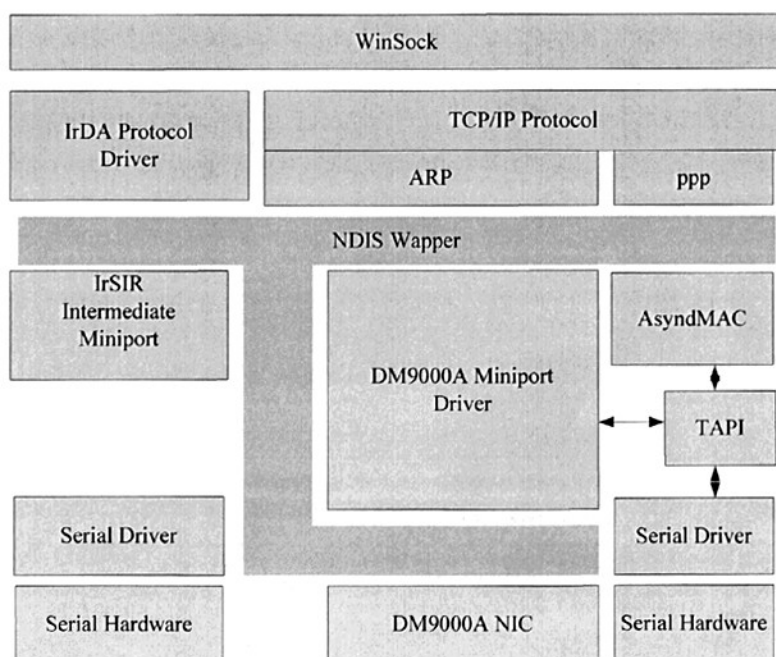


图 5-9 WinCE 网络架构体系图

在 WinCE 中, 开发网卡驱动就是写一个 Miniport Driver, 导出相应的 Miniport 接口函数, 调用 NDIS 的接口访问底层硬件。驱动架构如下图 5-10:

在 WinCE 中支持三种类型的网络驱动: Miniport driver, Intermediate driver 和 Protocol driver。本文介绍针对硬件网卡的驱动, 也就是 Miniport driver。Miniport driver 直接管理硬件网卡, 它使用 NDIS Library 中的接口函数读写硬件网卡, 同时对上层导出 Miniport 接口函数, 这样上层通过该接口可以配置网卡,

发送和接收网络数据包。

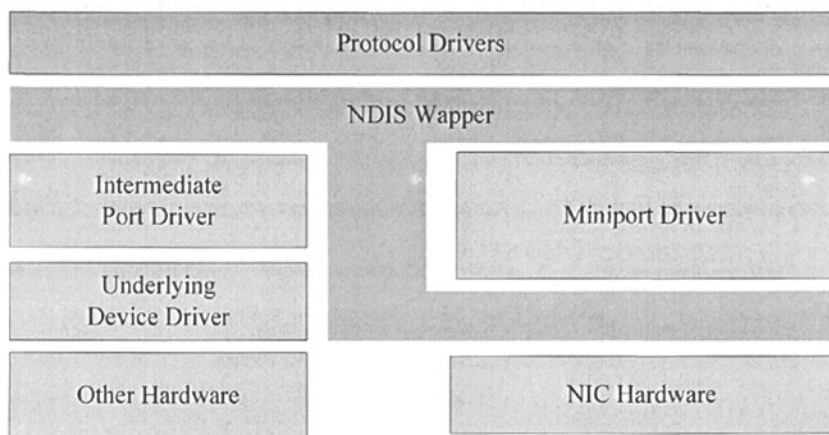


图 5-10 WinCE 网口驱动架构图

5.4.2 DM9000A 网口驱动开发

开发 DM9000A 网口驱动工作主要是填充 NDIS 定义的函数接口，以便对上层协议函数提供支持。如 `MiniportInitialize()`, `MiniportReset()`, `MiniportISRHandler()`, `MiniportSend()`等。这些函数位于 `Driver.cpp` 文件中提供 NDIS 所需的协议层服务，而实际上则调用 `dm9000a.cpp` 文件中的对应函数，最终 `dm9000a.cpp` 中的函数调用 `ndis` 提供的接口函数读写硬件网卡，实现具体的读写等操作。下文主要说明网口驱动开发过程中的重点工作——中断。

关于中断：如果一个驱动程序要处理一个中断，那么驱动程序需要首先使用 `CreateEvent()`函数建立一个事件，之后调用 `InterruptInitialize()`将该事件与中断号绑定，这一步就会使能该中断，OAL 中的 `OEMInerruptEnable()`就会被调用。然后驱动程序中的 IST 使用 `WaitForSingleObject()`函数来等待中断的发生。当一个硬件中断发生之后，异常处理程序调用 OAL 的 `OEMInterruptHandler()`函数，该函数检测硬件之后，将硬件中断转换为软件的中断号，返回给系统。该中断号就是 `InterruptInitialize()`中使用的那个中断号。系统得到该中断号之后，就会找到该中断号对应的事件，并唤醒等待相应事件的线程(IST)进行中断处理。处理完成之后，IST 调用 `InterruptDone()`通知操作系统中断处理结束，操作系统再次

调用 OAL 中的 OEMInterruptDone() 函数，完成中断的处理。

SysIntr 在文件 Nkintr.h 中预定义，用于唯一标识中断设备。开发者可以在文件 Oalintr.h 中定义自己的 SysIntr。常见的预定义 SysIntr 有 SYSINTR_NOP(中断只由 ISR 处理，IST 不再处理)、SYSINTR_RESCHEDED(重新调度线程)和 SYSINTR_DEVICES(由 WinCE 预定义的设备中断 ID 的基值)。其它如 SYSINTR_PROFILE、SYSINTR_TIMING、SYSINTR_FIRMWARE 等都是基于 SYSINTR_DEVICES 定义的。

网口的中断处理函数实现代码如下：

```
void C_DM9000::DeviceInterruptEventHandler( U32 uValue )
{
    if(uValue & 0x01) Dm9LookupRxBuffers();           // check RX activities
    if(!(uValue & 0x02)) return;                     // return if not TX latch
    U32 nsr;
    nsr = DeviceReadPort(DM9_NSR);
    if(nsr & 0x08)                                   // check TX-END2
    {
        m_nTxPendings--;
        DeviceSendCompleted(m_TQWaiting.Dequeue());
    }
    if(nsr & 0x04)                                   // check TX-END1
    {
        m_nTxPendings--;
        DeviceSendCompleted(m_TQWaiting.Dequeue());
    }
    if( nsr & 0x0C )                                 // report tx available now
    {
        NdisMSendResourcesAvailable(m_pUpper->GetNdisHandle());
    }
    DeviceWritePort(DM9_NSR,nsr);                   //clear NSR by set 1
}

```

笔者最初调试的时候因为 OEMInterruptHandler 未关闭中断，导致中断不断被触发而无法执行中断处理函数。

第6章 系统测试

6.1 嵌入式系统测试概要

由于嵌入式系统较高的实时性要求，与硬件紧密相关，微处理器种类繁多等特点，造成嵌入式软件的测试与一般商用软件的测试有了很大的不同^[43]：

1. 软硬件协同测试

嵌入式系统的硬件平台的依赖性，决定了软件测试时必须验证嵌入式软件与其所控制的硬件设备能否正确地交互，这就需要使用更好的测试方法和工具进行嵌入式系统测试—软硬件协同测试。

2. 测试的专用性

嵌入式软件设计是以一定的目标硬件平台为基础的、面向固定的任务进行的，而且由于应用层界面和功能难以统一，因此不存在通用的测试平台。正是因为这样，黑盒测试成为针对嵌入式软件的主要测试手段

3. 对于实时性测试的要求

嵌入式系统中，应用软件自身算法的复杂度和操作系统任务调度，决定了系统资源的分配和消耗，从而有必要借助一定的测试工具对应用程序算法复杂度和操作系统任务调度进行分析测试。

6.2 测试流程和方法概述

6.2.1 测试流程

本课题嵌入式系统测试可分为 5 个阶段：硬件平台测试、模块测试、集成测试、系统测试、特殊测试。

1 硬件平台测试

这部分主要利用专门的测试工具对各个硬件模块，PCB 线路板等硬件部分进行测试。

2. 模块测试

模块测试又称单元测试，是指编码完成后，是针对软件设计的最小单位——

程序模块进行正确性检验的测试工作。其目的在于检查每个程序单元能否正确实现详细设计说明书中的模块功能、性能、接口和设计约束等要求，发现各模块内部可能存在的各种错误。由于没有其它数据模块进行数据传递的支持，该阶段测试可以在宿主机上仿真硬件环境进行测试，也可以通过编写应用程序对各模块功能调用或函数实现进行测试和校验。模块测试通常要求开发人员从程序内部结构出发设计测试用例(Test Case)，良好的测试用例是测试成功的关键，同时，模块测试也在很大程度上决定了测试的覆盖率。

3. 集成测试

单个软件模块测试正确之后，将所有模块集成起来进行测试。本阶段主要是找出各模块之间数据传递和系统组成后的逻辑结构的错误，以及各模块集成后由于模块同时使用产生的干扰性错误。通常在单元测试的基础上，将所有的程序进行有序的、递增的测试。集成测试是检验程序单元或部件的接口关系，逐步集成为符合概要设计要求的程序部件或整个系统。

4. 系统测试

根据 Goodenough 和 Gerhart 提出的软件测试充分性准则可知，软件测试具有非复合性的特点，也就是说，即使以软件所有成分都进行了充分的测试，也并不意味着整个软件的测试已经充分。所以，即使通过了硬件平台测试、模块测试、集成测试，并不意味着已经完全了充分的测试，还要进行软硬件全面测试，即系统测试。这部分属于软硬件协同测试，需要在目标机上进行黑盒测试，验证每一项具体的功能。对产品进行压力测试（也称测试强度），意外测试和破坏性测试，可以进一步验证系统性能。

5. 其他测试

其他测试的具体内容通常根据产品的使用环境的不同而不同。本文所设计的智能终端核心板目的是应用于医疗以及消费电子领域的产品，因此，除上述测试内容外还应包括：老化测试、高低温测试等。

6.2.2 测试方法

嵌入式软件测试与其他软件测试采用相同的方法，即白盒测试与黑盒测试^[45]。

1. 白盒测试

根据源代码的组织结构查找软件缺陷，要求测试人员对软件的结构和作用有详细的了解。由于严格的安全性和可靠性的要求，嵌入式软件测试同非嵌入

式软件测试相比，通常要求有更高的代码覆盖率。

2. 黑盒测试

根据软件的用途和外部特征查找软件缺陷，不需要了解程序的内部结构。黑盒测试最大的优势在于不依赖代码，而是从实际使用的角度进行测试。在进行嵌入式软件黑盒测试时，要把系统的预期用途作为重要依据，根据需求中对负载、定时、性能的要求，判断软件是否满足这些需求规范。为了保证正确地测试，还须要检验软硬件之间的接口。嵌入式软件黑盒测试的一个重要方面是极限测试。

6.3 测试内容与结果

6.3.1 测试内容

本课题综合采用多种方法，在不同的开发阶段对智能终端核心板系统进行了全面测试，测试内容包括：

1. 裸机测试。裸机测试是指在系统样片完成之后，通过集成开发环境(本文使用 ADS1.2)开发裸机程序(无操作系统下的程序)，并将程序下载至 RAM 运行，以此查看各硬件模块功能。如，LCD 是否点亮、内存能否读写、串口是否正常、以及网口是否正常等。裸机测试通常能够较快发现问题，缩短开发周期。裸机测试主要为发现硬件问题，因此常常配合逻辑分析仪、频谱分析仪、网络分析仪、专业的示波器等仪器进行调试、测试。

2. 尽早实现平台的串口输出。串口的驱动相对来说比较简单，使用串口测试的方式主要目的是追踪代码的执行流程，测试其功能的正确性，也可以追踪变量传输过程的正确性。这是嵌入式系统开发中最常用的调试、测试手段之一。

3. 利用 Platform Builder 进行 Debug 版本的程序调试，方便跟踪各个程序模块的执行，开发调试完成后发布 Release 版本的开发包。

4. 与定时问题有关的白盒测试、中断测试、硬件接口测试等在目标环境中进行；而与逻辑或界面相关的测试、及其他与硬件无关的测试，则采用用仿真器和调试工具完成，加快了测试进度。

5. 充分利用 WinCE 开发工具 Platform Builder 提供的远程工具：

- (1) Remote Call Profiler：图形方式显示代码中算法的效率或资源占用情况；
- (2) RemoteKemelTracker：查看系统从硬件中断产生到处理的延时，跟踪内

存的申请和释放;

(3) RemotePerformanceMonitor: 查看内存、电池、系统进程、线程。

6. 利用 Embedded Visual C++ 开发上层应用程序实现模块测试和系统测试。在 Platform Builder 导出 SDK(Software Development Kits)的基础上, 使用 Embedded Visual C++ 调用操作系统服务或驱动接口函数, 通过完成制定的功能来测试各模块的功能是否满足需求, 以及各模块交互作用时候的干扰或影响情况是否在允许范围。

6.3.2 测试结果

模块测试通过 Embedded Visual C++ 开发上层应用程序(SBC2442.exe)完成, 主要原理即为调用 Win32 API 实现对各驱动模块的调用, 限于篇幅, 这里不再重点介绍该测试程序的实现代码。测试程序在宿主机上编译通过后, 使用 USB Device 线将宿主机与目标板(智能终端核心板)相连, 通过微软同步软件 Microsoft ActiveSync 将测试程序拷贝至目标板上。通过触摸屏或 USB 鼠标点击测试程序 (SBC2442.exe), 测试程序运行界面如下图 6-1 所示:

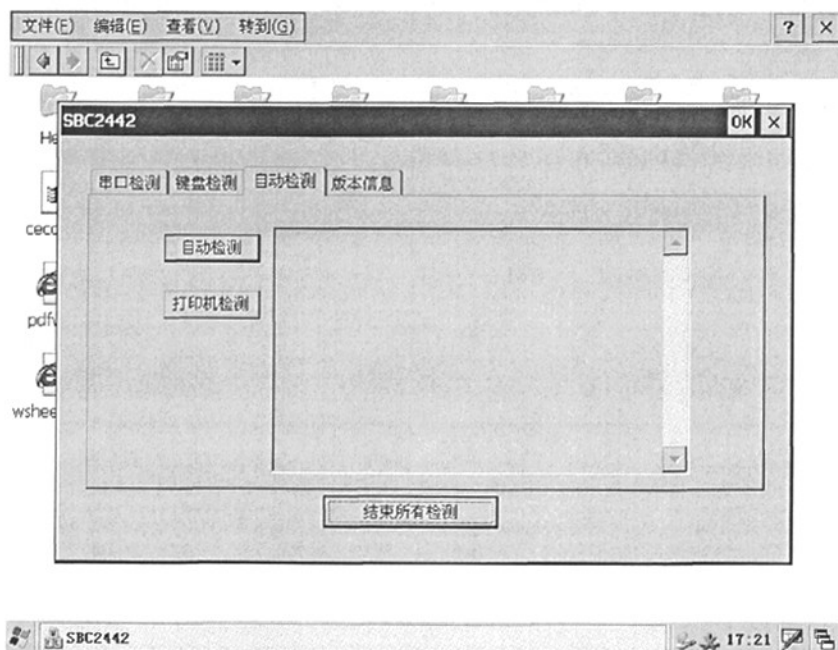


图 6-1 测试程序运行界面

该测试程序完成串口(三个串口)、键盘、网口、RTC、并口微型打印机等模块的测试。其它模块,如 LCD、VGA、USB Host/Device 等的测试可直接通过使用测试其效果,这里不再赘述。

经过测试,WinCE 5.0 操作系统能够正常启动;操作系统串口打印信息正常输出;BootLoader 能正确下载操作系统镜像;LCD/VGA 等设备驱动正常运行。

部分模块测试结果如下:

(1) USB Host 测试

通过 Hub 扩展后的 4 个 USB host 均能正常使用,并能够外接 USB 鼠标和 USB 键盘等;

(2) USB Device 测试

Bootloader 通过 ICE 仿真器下载至 RAM 并运行后,可通过 USB Device 下载操作系统镜像 NK.bin 文件;目标板系统运行后可通过 Device 数据线及微软同步软件 Microsoft ActiveSync 与宿主机同步;

(3) 打印机测试

经测试并行口打印机能打印字符、汉字、图形以及条形码等,打印 10 行汉字平均时间约为 1.1 秒,并且打印时主板功耗仅增加 2W;

(4) 串口测试

在 SBC2442.exe 的“串口检测”界面下可以对智能终端核心板的串口 3 和串口 4 进行检测。其中核心板的串口 2 被 WinCE 5.0OS 占用做调试口。在用串口线连接好 SBC2442 主板和 PC 后,在 PC 端运行 AccessPort.exe(该软件能够用来配置 PC 机串口参数,并能通过串口线发送或接受数据)。

将 PC 串口参数与目标板对应串口参数配置相同,如

“COM4-115200-NO-8-1”,通过目标板发送数据可以在 AccessPort 的“Terminal”端显示,通过 PC 发送的数据可以在测试软件 SBC2442.exe 的“接受区”中显示。测试结果如下图 6-2 所示:

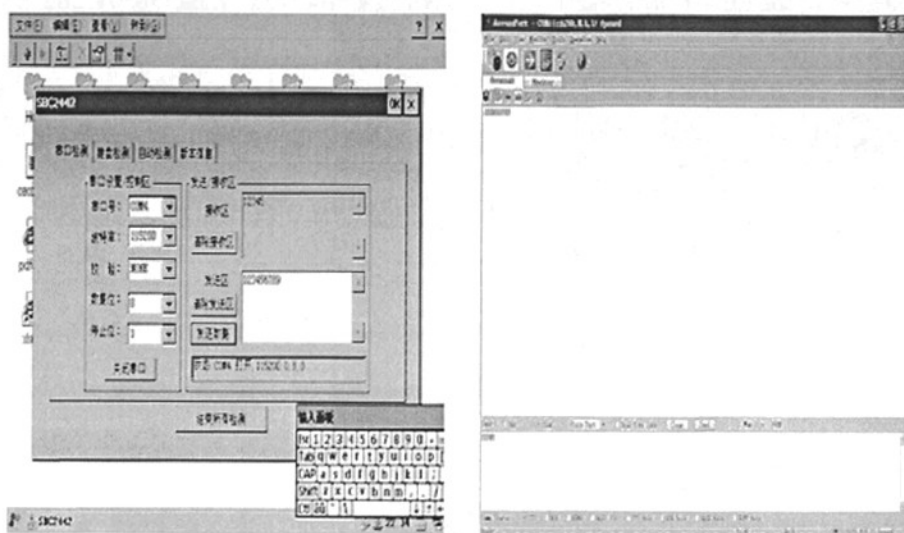


图 6-2 串口测试结果图

(4) SD 卡测试

上述测试程序能通过向 SD 卡写入一个文本文件来完成对 SD 卡的测试，插入 SD 卡后，点击“自动检测”，测试结果如下图 6-3 所示：

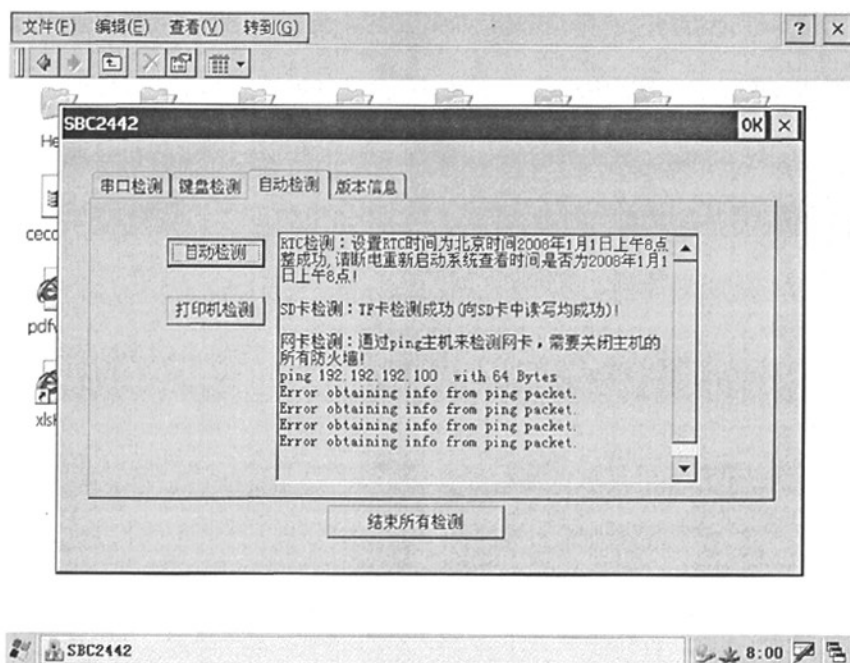


图 6-3 SD 卡测试结果图

(5) 网口测试

设置好 PC 机的 IP 为 192.192.192.100，子网掩码为 255.255.255.0，默认网关为：192.192.192.103，如图（14）所示。用网线连接好目标板和 PC 机的网口（可以用直连的网线也可以用交叉的网线，网卡芯片 DM9000A 自适应直连网线和交叉网线）。将目标板的 IP 地址设置为 192.192.192.115(或其它 IP 地址，只要在同一网段即可)，测试程序通过 ping 命令测试网口是否能正常工作，点击“自动测试”，测试结果如下图 6-4 所示：



图 6-4 网口测试效果图

(6) 键盘测试

在 SBC2442.exe 的“键盘检测”界面可以完成对智能终端核心板键盘模块的检测，在接上键盘之后，按下键盘上的按键即可以在“键盘检测”界面读到按下的按键相应的在 WinCE OS 下的虚拟键值。在分别按下键盘上的 1, 2, 3 按键后在“键盘检测”界面显示如图 6-5 所示：

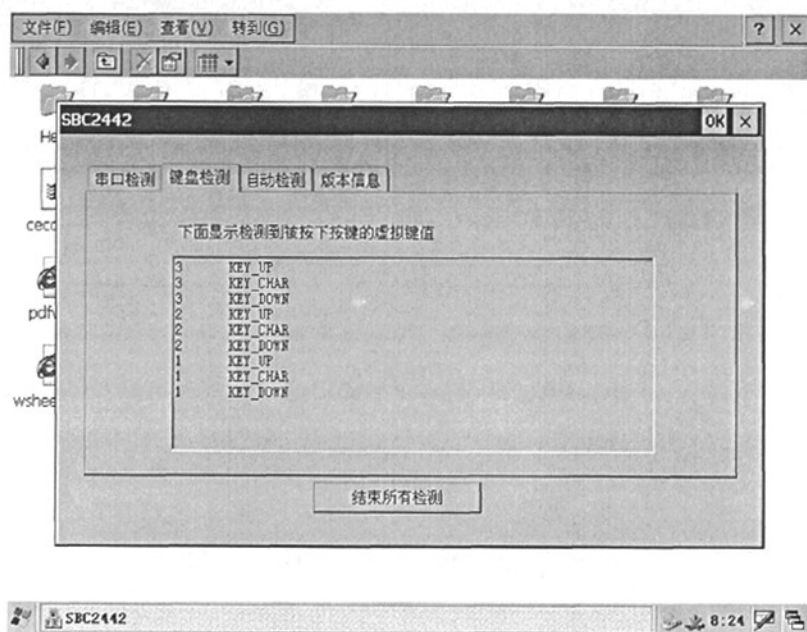


图 6-5 键盘测试结果图

其它测试结果:

- (1) 老化测试: 平均无故障时间为 314 小时;
- (2) 高低温测试: 系统在 -20°C 和 70°C 时运行平均无故障时间分别为 23 小时和 25 小时;

第7章 结束语

经过近一年的项目研发，本课题已经完成，且指定型号的产品已经面向市场，并为南昌某医疗仪器设备公司首批量产 1000 套，另一深圳某消费电子生产商(POS 机)的需求也正在洽谈中。

回首整个开发过程，笔者体会颇多。

本论文使用高性能 Samsung S3C2442A 微控制器，引入功能强大的嵌入式 WinCE 操作系统，开发出一种具备良好的通用性和可扩展性并提供丰富的外设接口的嵌入式智能终端核心板系统。文中主要从嵌入式系统软硬件开发技术、智能终端软硬件实现技术等相关方面进行了研究，主要工作成果如下：

1. 开发了一个可扩展的嵌入式智能终端核心板系统，给出了嵌入式开发的原理、技术和一般性开发的方法，并为二次开发提供了一个良好的平台支持；

2. 在高性能的 S3C2442A 处理器上，成功地移植了 WinCE 5.0 操作系统，并完成 BSP、LCD、触摸屏、LCD 转 VGA、矩阵键盘、DM9000A 网口以及总线外扩并口打印机的驱动开发，对同类嵌入式产品应用研究有一定借鉴价值；

3. 本文致力于具备一定通用性的嵌入式系统平台的研究与开发，因此，在此目的下，本文研究开发的智能终端核心板在如下三个方面具有较强兼容性优势：其一、本系统支持两种可选显示方式：LCD 和 VGA，并支持 6.4 寸和 10.5 寸两种 LCD 尺寸；其二、本系统支持 4 个 USB Host 和 3 个串口(其中一个为全功能串口)，并支持 USB 扩展鼠标、键盘等应用，以及串口扩展条形码扫描、GPS 接口等应用；其三、本系统 Eboot 支持 4 种 WinCE 系统下载方式：串口、USB Device、以太网口和 SD 卡启动，方便用户自更新系统。

4. 对智能终端核心板系统进行了软硬件协同设计与测试，保证了设备的可靠性与稳定性。

经过测试和实际应用的证明，本文所阐述的设计和实现方法，完全实现了设计目标，本文研究和开发的成果已经在相关领域中得到了较好的应用。

但是，由于时间紧迫，开发人员少，本文设计的智能终端核心板软硬件开发平台还有待进一步完善，主要有以下几个方面的工作：

1. 为充分满足消费电子领域要求，加入音频模块；

2. 由于 LCD 转 VGA 芯片内存仅 2 帧缓存(600K), 在读写 FLASH 时播放视屏文件, VGA 显示效果不是非常理想, 可考虑更换 CHROTEL 的 CH7025C 芯片(该芯片自带 2M 缓存, 本课题开始时还没面市);

3. 完善 USB Host 驱动, 使其支持 USB 摄像头功能, 增加产品竞争力;

4. 升级系统至 WinCE 6.0, 改善系统性能;

5. 增加电源管理程序, 进一步降低功耗。

至此, 论文的撰写工作也已完成。这次开发经历使我对嵌入式开发有了深入的了解, 给我日后的学习和工作打下了良好的基础。论文的顺利完成也提高了我的学术水平, 然而, 限于作者水平, 本文难免存在种种缺陷与不足, 恳请各位专家批评指正。

参考文献

- [1] 杨宗德. 嵌入式ARM系统原理与实例开发. 北京: 北京大学出版社, 2006: 1~5
- [2] 陈章龙, 唐志强, 涂时亮. 嵌入式技术与系统: Intel XScale 结构与开发. 北京: 北京航空航天大学出版社, 2004: 1~5
- [3] 孙纪坤, 张小全. 嵌入式 Linux 系统开发技术详解——基于 ARM. 人民邮电出版社, 2006
- [4] 张崙. 32位嵌入式系统硬件设计与调试. 北京: 机械工业出版社, 2005: 1~3
- [5] 周立功等. ARM微控制器基础与实战. 北京: 北京航空航天大学出版社, 2003.11
- [6] 张晓林, 崔迎炜等. 嵌入式系统设计与实践[M]. 北京: 北京航空航天大学出版社, 2006: 1~18
- [7] 何宗健. Windows CE嵌入式系统. 北京: 北京大学出版社, 2006.9: 1~6, 290~297
- [8] 陈文智等. 嵌入式系统开发原理与实践[M]. 北京: 清华大学出版社, 2005: 1~4
- [9] 姜波. Windows CE.Net程序设计. 北京: 机械工业出版社, 2007: 2~3
- [10] Gallas, B; Verma, V. Embedded Pentium(R) processor system design for Windows CE. WESCON/98 15-17 Sept. 1998:114~123
- [11] Wolf, W. Panel: Embedded Systems and Hardware-Software Co-Design: Panacea or Pandora's Box? Design Automation, 1993.30th Conference On 14-18 June 1993:308
- [12] 探砂工作室. 嵌入式开发圣经[M]. 北京: 中国青年出版社, 2002: 1~27
- [13] 于明, 范书瑞, 曾祥焯. ARM9嵌入式同设计与开发教程[M]. 北京: 电子工业出版社, 2006
- [14] 蒋亚群, 张春元. ARM微处理器体系结构及其嵌入式SOC[J]. 计算机工程, 2002,11(28) 4~6
- [15] 田泽等. 嵌入式系统开发与应用[M]. 北京: 北京航空航天大学出版社, 2005: 15~25
- [16] 陈曠. ARM嵌入式技术实践教程. 北京: 北京航空航天大学出版社, 2005:2~8
- [17] 周立功. ARM微控制器基础与实战(第二版). 北京: 北京航空航天大学出版社, 2005.8:1~2
- [18] ARM9 user's manual.<http://www.arm.com/ARM9>.
- [19] 邵向鑫. 基于ARM9的GPS地理信息采集系统. [吉林大学硕士研究生学位论文]:10~12.22~27

- [20] 李建忠. 基于ARM的家庭自动化系统的研究. [南京理工大学硕士研究生学位论文]: 2008.6: 13~14
- [21] 李旭光等. ARM应用系统开发详解: 基于S3C4510B的系统设计[M]. 北京: 清华大学出版社, 2003.12: 15~21
- [22] 宋延昭. 嵌入式操作系统介绍及选型原则[J]. 工业控制计算机, 2005,7(18): 41~42
- [23] 金亮,张学杰. 3种嵌入式操作系统内核的关键技术分析[J]. 云南大学学报, 2007,28(S2):1~4
- [24] 屈鹏. 基于ARM的嵌入式MINIX 3操作系统的移植. [兰州大学硕士研究生学位论文]. 2008.5: 3~4
- [25] 邵贝贝译. 嵌入式实时操作系统[M].北京航空航天大学出版社, 2003.7
- [26] Samsung. S3C2442A45 User's Manual Rev1.1. Samsung Corp. 2004
- [27] 杨显强, 田远富. 嵌入式系统中LCD驱动的实现原理[J], 今日电子,2005, 8: 65
- [28] Yu-Jung Huang, Chi-Feng Liu, Shao-pin Chang etc., Design of LCD Driver IP for OSC Applications, IEEE AP-ASIC2004 Aug 4-5 2004, P62-P65
- [29] 骆飞. ARM嵌入式系统实验开发平台研究与实现. [西南交通大学硕士研究生学位论文]:28~29
- [30] Sangik Chai, Shinwood Kang, Implement of an On-Chip Bus Bridge between Heterogeneous Buses with Different Clock Frequencies. Proceedings of the 9th IDEAS' 05 IEEE 2005
- [31] 张冬泉等. Windows CE实用开发技术[M]. 北京: 电子工业出版社, 2006,09:3~5,14~16
- [32] Microsoft Corporation Microsoft Windows CE 5.0 Document [Z].
- [33] 周毓林等. Windows CE.net内核定制及应用开发[M]. 北京: 电子工业出版社,2006.15
- [34] 何伟光. 基于Windows CE的电子记录仪的开发. [湖南大学硕士研究生学位论文]. 2006.33-34
- [35] 赵彦斌.基于IntelXScale架构的Windows CE操作系统的移植.[西安电子科技大学硕士研究生学位论文]. 2006.13~14
- [36] George, M. R; Wong, W. F. Windows CE for a reconfigurable system-on-a-chip processor. Field-Programmable Technology, 2004. Proceedings. 2004 IEEE International Conference on 2004:201~207
- [37] Ji Wankang, Yang Jia, Hong Yongqiang. BSP Development of WinCE System for Vehicle Navigation Device Based on S3C2440. Electronic Measurement and Instruments,2007. ICEMI '07. 8th International Conference on Aug. 16 2007-July 18 2007:2389~2391

- [38] 胡军辉,王友钊. Windows CE设备驱动程序开发[J]. 计算机工程.2006,8(16):41~43
- [39] 罗家兵. Wince.net下流接口驱动研究与实现[J]. 微计算机信息.2007,9-3:229~230
- [40] Microsoft. Microsoft Windows CE Device Driver Kit [M]. 希望译. 北京:北京希望电子出版社, 2000.
- [41] 吕国亮,赵曙光,赵俊. 嵌入式逻辑分析技术及其在FPGA 系统开发中的应用[J]. 液晶与显示, 2007, 22 (2): 22~23
- [42] 孙记明. Windows CE环境下无线网卡设备及驱动的研究与实现. [北京邮电大学硕士研究生学位论文]. 17-22
- [43] Bart Broekman, Edwin Notenboom. Testing Embedded Software. An imprint of PEARSON EDUCATION.2003:3~6,255~264

致 谢

三年的硕士研究生生涯转瞬即逝，有很多的老师、同学、朋友和家人在学习上给予我指导、关心和帮助，在我即将毕业之际，向他们表示衷心的感谢。

首先感谢我的导师方安平老师。在论文的选题时候，得到导师富有灵感的启发和诱导；在论文的调研、开题和撰写过程中，得到导师严肃、认真、耐心而又负责任的检查和指导。方老师严谨的治学态度，富有感染力的教学风范，以及在所从事的科研工作和工程项目中，实事求是、一丝不苟、精益求精的严谨作风，无论在做人还是做事上都使我受益终生。方老师身体力行的表率，让我更加坚定的相信：成功源于一点一滴的付出；持之以恒，学业终有所成。

我还要感谢我的同学和朋友们，正是在他们的鼓励和帮助之下，我才能顺利完成我的毕业设计，与他们的讨论使我受到很多启发。我还要感谢我的师兄们，你们的敬业精神和好学态度，使我深深受益，你们的帮助和无私的给予，让我无比感动。

论文的撰写过程中还得到了实习单位各位同事的热情帮助，感谢你们为本文的研究工作所付出的辛勤劳动。

感谢我的父母，感谢你们的养育之恩，是你们给了我坦然面对生活的勇气、热情，我的一点一滴的成绩都离不开你们的关心、鼓励。

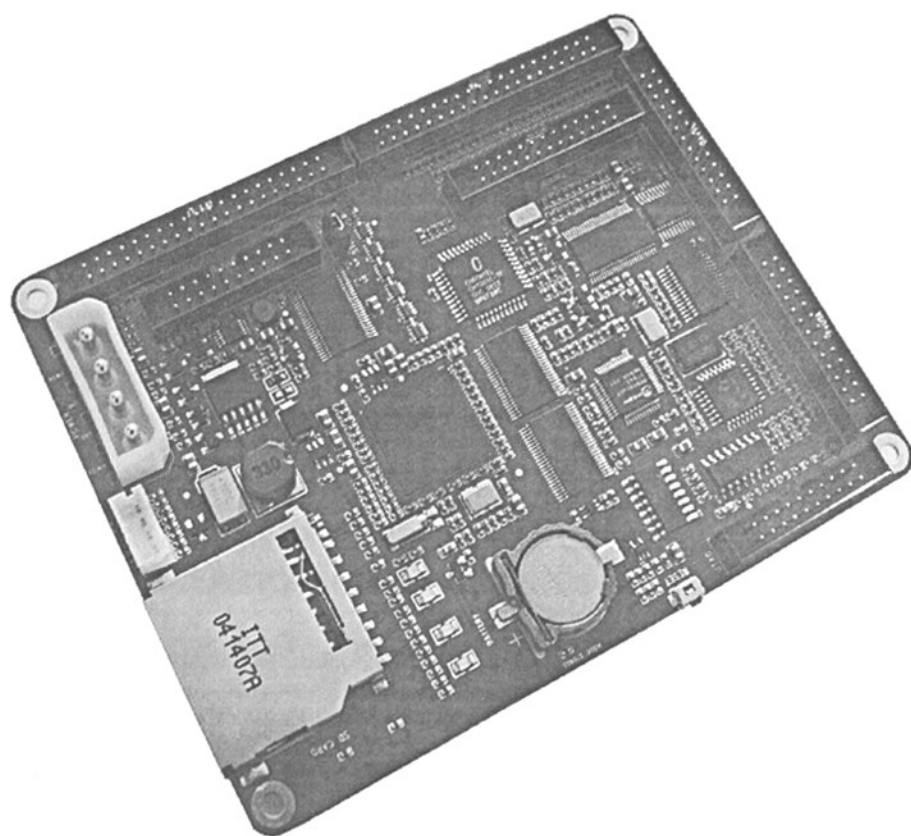
感谢我的女友宋文静，在我写论文的过程中，你的宽容、理解和善良给予我巨大的动力。

最后，衷心感谢在百忙之中评阅论文和参加答辩的各位专家、教授，感谢你们为我的学业付出的心血！

附录 A 智能终端核心板在某医疗仪器中的应用实物图



附录 B 智能终端核心板实物图



附录 D 攻读硕士学位期间公开发表的论文

- 1、方安平 王志伟《基于 S3C2442 的并行口打印机及驱动设计》《武汉理工大学学报信息与管理工程版》2009.2