项目一　认识嵌入式系统及平台

项目概述

随着社会信息化的日益加强，计算机和网络已经全面渗透到日常生活的每一个角落。对于我们每个人来说，需要的已经不仅仅是那种放在桌上处理文档、进行工作管理和生产控制的“机器”。任何一个普通人都可能拥有大小不一的、形状各异的、使用嵌入式技术的电子产品。嵌入式系统已成为当前最热门、最有发展前途的 IT 应用之一。据预测，在未来几年，嵌入式系统的发展将为几乎所有的电子设备注入新的活力。

知识目标

（1）了解嵌入式系统的定义、分类与特点；
（2）了解嵌入式系统的组成；
（3）了解嵌入式处理器的分类与特点；
（4）了解嵌入式操作系统的种类与特点；
（5）了解嵌入式系统硬件和软件；
（6）了解嵌入式系统开发流程和开发要点。

技能目标

（1）会在虚拟机中安装操作系统；
（2）学会嵌入式系统开发流程和开发要点。

任务一　了解嵌入式系统

知识 1　嵌入式系统简介

根据美国嵌入式系统专业媒体 RTC 报道，在 21 世纪初的 10 年中，全球嵌入式系统市场具有比 PC 市场大 10 到 100 倍的商机。中国拥有世界上最大的电子产品消费市场，手机、彩电等家用电器的拥有量都居世界第一。随着经济水平的提高和消费结构的改变，人们对消费电子产品的要求越来越高，比如产品的灵活性、可控性、耐用性、性价比等，这些要求我们都可以通过采用合理、有效的嵌入式系统设计和优化来实现。另外，在现代化的医疗、测控仪器和机电产品中对系统的可靠性、实时性要求较高，更需要有专用的嵌入式系统的支持，这些需求都极大地刺激了嵌入式系统的发展和产业化的进程。

一、嵌入式系统的定义

嵌入式系统（Embedded System）是一种“完全嵌入受控器件内部，为特定应用而设计的专用计算机系统”，根据英国电气工程师协会（U.K. Institution of Electrical Engineer）的定

义，嵌入式系统为控制、监视或辅助设备、机器或用于工厂运作的设备。与个人计算机这样的通用计算机系统不同，嵌入式系统通常执行的是带有特定要求的、预先定义的任务。由于嵌入式系统只针对一项特殊的任务，所以设计人员能够对它进行优化，减小尺寸、降低成本。嵌入式系统通常生产规模较大，所以单个产品的成本节约，能够随着产量进行成百上千倍的放大。

嵌入式系统的核心由一个或几个预先编程好以用来执行少数几项任务的微处理器或者单片机组成。与通用计算机能够运行用户选择的软件不同，嵌入式系统上的软件通常是暂时不变的，所以经常称为“固件”。长期以来，学术界对嵌入式系统的定义一直在争论中，目前，国内普遍认同的嵌入式系统定义为：以应用为中心，以计算机技术为基础，软、硬件可裁剪，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。

二、嵌入式系统的特点

嵌入式系统是面向应用的专用计算机系统，与通用型计算机系统相比具有以下特点。

1. 系统内核小

由于嵌入式系统一般是应用于小型电子装置的，系统资源相对有限，所以内核较之传统的操作系统要小得多。目前的嵌入式系统的核心往往容量只有几千到几万比特，需要根据实际的使用进行功能扩展或者裁减，由于微内核的存在，使得这种扩展能够非常顺利地进行。

2. 专用性强

嵌入式系统的专用性很强，其中软件系统和硬件系统的结合非常紧密，一般要针对硬件进行系统的移植，即使在同一品牌、同一系列的产品中也需要根据系统硬件的变化和增减不断进行修改。同时针对不同的任务，往往需要对系统进行较大更改，程序的编译下载要和系统相结合，这种修改和通用软件的“升级”完全是两个概念。

嵌入式处理器与通用微处理器的最大区别在于，嵌入式微处理器大多工作在为特定用户群所专门设计的系统中，它将通用 CPU 许多由板卡完成的任务集成到嵌入式微处理器内部，从而有利于嵌入式系统在设计时趋于小型化，同时还具有很高的效率和可靠性，可以增强移动能力和网络耦合能力。

3. 系统比较精简

嵌入式系统一般没有系统软件和应用软件的明显区分，不要求其功能设计及实现上过于复杂，这样一方面利于控制系统成本，同时也利于保证系统安全。

4. 知识密集

嵌入式系统是将先进的计算机技术、半导体技术、电子技术和各个行业的具体应用相结合的产物，这一特点就决定了它必然是一个技术密集、资金密集、高度离散、不断创新的知识集成系统。所以，进入嵌入式系统行业，对知识和技术的要求较高。例如，Palm 之所以在 PDA 领域占有很大的市场份额，就是因为其立足于个人电子消费品，着重发展图形界面和多任务管理；而风河的 VxWorks 之所以在火星车上得以应用，则是因为其高实时性和高可靠性。

5. 高实时性和高可靠性

嵌入式系统经常用于控制领域，这就要求系统软件实时性要强，因此，常常需要将软件

固化，以提高速度；软件代码要求高质量和高可靠性。

无论用于控制领域还是用于独立设备、仪器仪表，都要求嵌入式系统具有高可靠性，特别是一些在极端环境下工作的嵌入式系统，其可靠性设计尤其重要。大多数嵌入式系统都包含一些硬件和软件机制来保证系统的可靠性。例如，硬件的看门狗电路在软件失去控制后使系统重新启动；软件的自动纠错功能可使系统检测到软件运行偏离正常流程时，通过软“陷阱”将其重新纳入正常轨道。

6. 多任务的操作系统

嵌入式系统的应用程序可以不通过操作系统直接在芯片上运行；但是为了合理地调度多任务、利用系统资源、系统函数以及和专家库函数接口，用户必须自行选配 RTOS（Real Time Operating System）开发平台，这样才能保证程序执行的实时性、可靠性，并减少开发时间，保障软件质量。

7. 开发环境的特殊性

由于嵌入式系统本身资源有限，其本身不具备自主开发能力，设计完成以后用户通常是不能对其中的程序功能进行修改的，必须有一套开发工具和环境才能进行开发，这些工具和环境一般基于通用计算机上的软硬件设备以及各种逻辑分析仪、混合信号示波器等。开发时往往有主机和目标机的概念，主机用于程序的开发，目标机作为最后的执行机，开发时需要交替结合进行。

三、嵌入式系统的应用

随着信息化、智能化、网络化的发展，嵌入式技术获得了广阔的发展空间，几乎渗透到人们生活的每一个角落。下面介绍一些嵌入式技术的典型应用。嵌入式系统的应用前景如图 1-1 所示。

1. 工业控制

基于嵌入式芯片的工业自动化设备正获得长足的发展，目前已经有大量的 8 位、16 位、32 位、64 位嵌入式微控制器在应用中。网络化是提高生产效率和产品质量、减少人力资源的主要途径，可用于工业过程控制、数控机床、电力系统、电网安全、电网设备监测、石油开采等。就传统的工业控制产品而言，低端型采用的往往是 8 位单片机。但是随着技术的发展，32 位、64 位的处理器逐渐成为工业控制设备的核心。

嵌入式系统还广泛应用于 ATM 机、自动售货机、工业控制系统等专用设备，与通信相结合可生产出高质量的 GPS、GPRS 等优质产品。

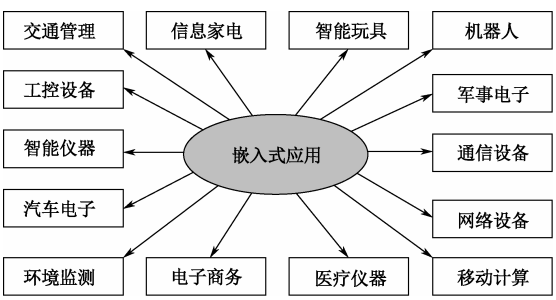

图 1-1　嵌入式系统的应用

2. 交通管理

在车辆导航、流量控制、信息监测与汽车服务方面，嵌入式系统技术已经获得了广泛的应用，内嵌 GPS 模块、GSM 模块的移动定位终端已经在各种运输行业获得了成功的使用。目前 GPS 设备已经从尖端产品领域进入了普通百姓的家庭，通过 GPS 嵌入式系统，可以随时随地跟踪移动目标。

3. 信息家电

这已成为嵌入式技术最大的应用领域，冰箱、彩电、洗衣机、空调等家电产品的智能化将引领人们的生活步入一个崭新的空间。智能家居系统的诞生使得即使主人不在家里，也可以通过电话、手机、网络进行远程控制。在这些设备中，嵌入式技术发挥着重要作用。

4. 家庭智能管理系统

水、电、煤气表的远程自动抄表，智能化安全防火、防盗系统的实现，其中嵌有的专用控制芯片将代替传统的人工检查，并实现更高、更准确和更安全的性能。目前在服务领域，如远程点菜系统等已经体现了嵌入式系统的优势。

5. POS 网络及电子商务

公共交通无接触智能卡（Contactless Smart Card，CSC）发行系统、公共电话卡发行系统、自动售货机以及各种智能 ATM 终端将全面走入人们的生活，手持一卡行遍天下已经成为现实。

6. 环境工程与自然

嵌入式系统可应用于水文资料实时监测，防洪体系及水土质量监测、堤坝安全，地震监测网，实时气象信息网，水源和空气污染监测。在很多环境恶劣、地况复杂的地区，嵌入式系统将实现无人监测、自动报警和应急处理。

7. 机器人

嵌入式芯片的发展将使机器人在微型化、高智能方面优势更加明显，同时会大幅度降低机器人的价格，使其在工业领域和服务领域获得更广泛的应用。

8. 消费电子

嵌入式技术提升了移动数据处理和通信的功效，加之易于实现自然人机交互的多媒体界面，因而在消费电子领域获得广泛应用。机顶盒的诞生使数字电视从单向音频传输变成了双向交互平台。手机手写文字输入、语音拨号上网、网页浏览、收发电子邮件已经成为现实。用于物流管理、条码扫描、移动信息采集的小型手持嵌入式系统在企业管理中也已发挥了巨大的作用。未来的手机和 PDA 将成为个人日常事务处理的综合平台和遥控工具。

这些应用中的重点之一就是控制应用。就远程家电控制而言，除了开发出支持 TCP/IP 的嵌入式系统之外，家电产品控制协议也需要制定和统一，同样的道理，所有基于网络的远程控制器件与嵌入式系统之间都应有接口，然后再由嵌入式系统来通过网络实现控制。所以，开发和探讨嵌入式系统有着十分重要的意义。

四、嵌入式系统的发展

我们目前正处于后 PC 时代中，而嵌入式系统就是与这一时代紧密相关的产物。它拉近人与计算机的距离，形成一个人机和谐的工作与生活环境。嵌入式技术的发展经历了 4

个阶段：第一阶段是以单芯片为核心的可编程控制器形式的系统；第二阶段是以嵌入式CPU为基础、以简单任务调度程序为核心的系统；第三阶段是以嵌入式操作系统为标志的系统；第四阶段是以基于Internet互联为标志的系统。嵌入式系统的发展趋势是：3C融合（多媒体手机、IP视频电话、无线PDA等）；灵活性/功率/性能/成本等方面的平衡考虑（尤其体现在便携式嵌入产品的开发）；灵活支持多种格式/制式/技术（包括音频、视频、显示格式、多种无线技术）。

嵌入式操作系统是一种实时的、支持嵌入式系统应用的操作系统软件，它是嵌入式系统（包括硬、软件系统）极为重要的组成部分，通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。目前，嵌入式操作系统的品种较多，据统计，仅用于信息电器的嵌入式操作系统就有40种左右，其中较为流行的主要有：Windows CE、Palm OS、Real-Time Linux、VxWorks、pSOS、Power TV以及Microware公司的OS-9。与通用操作系统相比较，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。

在软件方面，操作系统是嵌入式系统一个非常重要的组成部分。从上世纪90年代开始，陆续出现了一些非常优秀的实时多任务的操作系统，目前应用最广的几个嵌入式操作系统为嵌入式Linux、VxWorks、Windows CE等。

实际上，嵌入式系统本身是一个外延极广的名词，凡是与产品结合在一起的具有嵌入式特点的控制系统都可以叫嵌入式系统，而且有时很难给它下一个准确的定义。现在人们讲嵌入式系统时，某种程度上指近些年比较热门的、具有操作系统的嵌入式系统。一般而言，嵌入式系统由硬件和软件两部分组成。

知识2　嵌入式系统硬件

嵌入式系统是将计算机硬件和软件结合起来构成的一个专门的装置，这个装置可以完成一些特定的功能和任务，能够在没有人工干预的情况下独立地进行实时监测和控制。由于被嵌入对象的体系结构、应用环境不同，所以各个嵌入式系统可以由各种不同的结构组成。一个典型的嵌入式系统如图1-2所示。

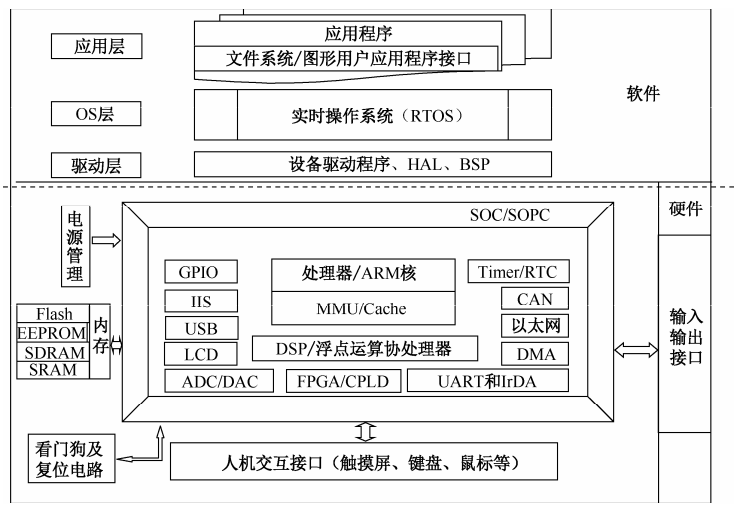

图1-2　嵌入式系统的组成

嵌入式系统硬件通常包含嵌入式微处理器、存储器（SDRAM、ROM、Flash等）、通用

设备接口和 I/O 接口（A/D、D/A、I/O 等）。在一片嵌入式处理器基础上添加电源电路、时钟电路和存储器电路，就构成了一个嵌入式核心控制模块。其中操作系统和应用程序都可以固化在 ROM 中。

知识 3　嵌入式系统软件

嵌入式系统软件通常由系统引导程序、操作系统和应用程序组成。BSP 是一个介于操作系统和底层硬件之间的软件层次，包括了系统中大部分与硬件联系紧密的软件模块。设计一个完整的 BSP 需要完成两部分工作：嵌入式系统的硬件初始化以及 BSP 功能设计，完成硬件相关的设备驱动。下面简单介绍设计 BSP 需要完成的工作。

1. 嵌入式系统硬件初始化

系统初始化过程可以分为 3 个主要环节，按照自底向上、从硬件到软件的次序依次为：片级初始化、板级初始化和系统级初始化。

片级初始化完成嵌入式微处理器的初始化，包括设置嵌入式微处理器的核心寄存器和控制寄存器、嵌入式微处理器核心工作模式和嵌入式微处理器的局部总线模式等。片级初始化把嵌入式微处理器从上电时的默认状态逐步设置成系统所要求的工作状态。这是一个纯硬件的初始化过程。

板级初始化完成嵌入式微处理器以外的其他硬件设备的初始化。另外，还设置某些软件的数据结构和参数，为随后的系统级初始化和应用程序的运行建立硬件和软件环境。这是一个同时包含软硬件两部分在内的初始化过程。

系统初级始化的过程以软件初始化为主，主要进行操作系统的初始化。BSP 将对嵌入式微处理器的控制权转交给嵌入式操作系统，由操作系统完成余下的初始化操作，包含加载和初始化与硬件无关的设备驱动程序，建立系统内存区，加载并初始化其他系统软件模块，如网络系统、文件系统等。最后，操作系统创建应用程序环境，并将控制权交给应用程序的入口。

2. 完成硬件相关的设备驱动程序

BSP 的另一个主要功能是硬件相关的设备驱动。硬件相关的设备驱动程序的初始化通常是一个从高到低的过程。尽管 BSP 中包含硬件相关的设备驱动程序，但是这些设备驱动程序通常不直接由 BSP 使用，而是在系统初始化过程中由 BSP 将其与操作系统中通用的设备驱动程序关联起来，并在随后的应用中由通用的设备驱动程序调用，实现对硬件设备的操作。与硬件相关的驱动程序是 BSP 设计与开发中另一个非常关键的环节。

系统软件层由实时多任务操作系统（RTOS）、文件系统、图形用户接口（Graphic User Interface，GUI）、网络系统及通用组件模块组成。RTOS 是嵌入式应用软件的基础和开发平台。

嵌入式操作系统（Embedded Operation System，EOS）是一种用途广泛的系统软件，过去它主要应用于工业控制和国防系统领域。EOS 负责嵌入式系统的全部软、硬件资源的分配、任务调度、控制、协调并发起活动。它必须体现其所在系统的特征，能够通过装卸某些模块来达到系统所要求的功能。目前，已推出一些应用比较成功的 EOS 产品系列。随着 Internet 技术的发展、信息家电的普及应用及 EOS 的微型化和专业化，EOS 开始从单一的弱功能向高专业化的强功能方向发展。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

化以及应用的专用性等方面具有较为突出的特点。EOS 是相对于一般操作系统而言的，它除具备了一般操作系统最基本的功能，如任务调度、同步机制、中断处理、文件功能等外，还具有很多特点：可装卸性（开放性、可伸缩性的体系结构）；强实时性（EOS 实时性一般较强，可用于各种设备控制当中）；统一的接口（提供各种设备驱动接口）；操作方便、简单、提供友好的图形界面，追求易学易用；提供强大的网络功能，支持 TCP/IP 协议及其他协议，提供 TCP/UDP/IP/PPP 协议支持及统一的 MAC 访问层接口，为各种移动计算设备预留接口；强稳定性，弱交互性（嵌入式系统一旦开始运行就不需要用户过多的干预，这就要负责系统管理的 EOS 具有较强的稳定性。嵌入式操作系统的用户接口一般不提供操作命令，它通过系统调用命令向用户程序提供服务）；固化代码（在嵌入式系统中，操作系统和应用软件被固化在嵌入式系统计算机的 ROM 中，辅助存储器在嵌入式系统中很少使用，因此，嵌入式操作系统的文件管理功能应该能够很容易地使用各种内存文件系统）；更好的硬件适应性（也就是良好的移植性）。

任务二　嵌入式系统硬件

嵌入式系统硬件由嵌入式处理器、总线、存储器和外设组成。

知识 1　硬件介绍

嵌入式微处理器将通用 CPU 许多由板卡完成的任务集成在芯片内部，从而有利于嵌入式系统在设计时趋于小型化，同时还具有很高的效率和可靠性。

嵌入式微处理器的体系结构可以采用冯·诺依曼或哈佛体系结构，早期采用冯·诺依曼体系结构，现在多数采用哈佛体系结构。指令系统可以选用精简指令系统（Reduced Instruction Set Computer，RISC）和复杂指令系统（Complex Instruction Set Computer，CISC），一般选用精简指令系统。RISC 计算机在通道中只包含最有用的指令，确保数据通道快速执行每一条指令，从而提高了执行效率并使 CPU 硬件结构设计变得更为简单。

嵌入式微处理器有各种不同的体系，即使在同一体系中也可能具有不同的时钟频率和数据总线宽度，或集成了不同的外设和接口。据不完全统计，目前全世界嵌入式微处理器已经超过 1000 多种，体系结构有 30 多个系列，其中主流的体系有 ARM、MIPS、Power PC、X86 和 SH 等。不同的是，没有一种嵌入式微处理器可以主导市场，仅以 32 位的产品而言，就有 100 种以上的嵌入式微处理器。嵌入式微处理器的选择是根据具体的应用而决定的。

一、嵌入式处理器

目前，常用的嵌入式处理器一般分为嵌入式微处理器、嵌入式微控制器、嵌入式 DSP 处理器和嵌入式片上系统。

1. 嵌入式微处理器

嵌入式微处理器（Embedded MicroProcessor Unit，EMPU）是由通用计算机中的 CPU 演变而来的。它的特征是具有 32 位以上的处理器，具有较高的性能，当然其价格也相应较高。但与计算机处理器不同的是，在实际嵌入式应用中，只保留和嵌入式应用紧密相关的功能硬件，去除其他的冗余功能部分，这样就以最低的功耗和资源实现嵌入式应用的特殊要求。和

工业控制计算机相比，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。主要的嵌入式处理器类型有 Am186/88、386EX、SC-400、Power PC、68000、MIPS、ARM/StrongARM 系列等。

CISC 和 RISC 是目前设计、制造微处理器的两种典型技术，虽然它们都试图在体系结构、操作系统、运行硬件、编译时间和运行时间等诸多因素中实现某种平衡，以求达到高效的目的，但采用的方法不同，因此在很多方面差异很大。

2. 嵌入式微控制器

嵌入式微控制器（MicroController Unit，MCU）又称单片机，顾名思义是将整个计算机系统集成到一块芯片中。从上世纪 70 年代末单片机出现到今天，虽然已经经过了几十年的时间，但这种 8 位的电子器件在嵌入式设备中仍然有着极其广泛的应用。单片机芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串行口、脉宽调制输出、A/D、D/A、Flash RAM、EEPROM 等各种必要功能和外设。和嵌入式微处理器相比，微控制器的最大特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高。微控制器的片上外设资源一般比较丰富，适合于控制，因此称微控制器。为适应不同的应用需求，一般一个系列的单片机具有多种衍生产品，每种衍生产品的处理器和内核都是一样的，不同之处在于存储器和外设的配置及封装，这样可以使单片机最大限度地同应用需求相匹配，从而降低功耗和成本。

由于 MCU 低廉的价格，优良的功能，所以拥有的品种和数量最多，比较有代表性的包括 8051、MCS-251、MCS-96/196/296、P51XA、C166/167、68K 系列等，MCU 占嵌入式系统约 70%的市场份额，Atmel 出产的 AVR 单片机由于其集成了 FPGA 等器件，所以具有很高的性价比，势必将推动单片机获得更好的发展。

3. 嵌入式 DSP 处理器

嵌入式 DSP 处理器（Embedded Digital Signal Processor，EDSP）对系统结构和指令进行了特殊设计，使其适合于 DSP 算法，编译效率较高，指令执行速度较快。它是专门用于信号处理方面的处理器，在数字滤波、FFT、谱分析等各种仪器上获得了大规模的应用。

DSP 的理论算法在上世纪 70 年代就已经出现，但是由于专门的 DSP 处理器还未出现，所以这种理论算法只能通过 MPU 等分立元件实现。MPU 较低的处理速度无法满足 DSP 的算法要求，其应用领域仅仅局限于一些尖端的高科技领域。随着大规模集成电路技术发展，1982 年世界上诞生了首枚 DSP 芯片。其运算速度比 MPU 快了几十倍，在语音合成和编码解码器中得到了广泛应用。至上世纪 80 年代中期，随着 CMOS 技术的进步与发展，第二代基于 CMOS 工艺的 DSP 芯片应运而生，其存储容量和运算速度都得到成倍提高，成为语音处理、图像硬件处理技术的基础。到上世纪 80 年代后期，DSP 的运算速度进一步提高，应用领域也从上述范围扩大到了通信和计算机方面。上世纪 90 年代后，DSP 发展到了第五代产品，集成度更高，使用范围也更加广阔。

应用最为广泛的是 TI 的 TMS320C2000/C5000 系列，另外如 Intel 的 MCS-296 和 Siemens 的 TriCore 也有各自的应用范围。

DSP 的设计者们把重点放在了处理连续的数据流上。在嵌入式应用中，如果强调对连续数据流的处理及高精度复杂运算，则应该选用 DSP 器件。

4. 嵌入式片上系统

嵌入式片上系统（System On Chip，SoC）是追求产品系统最大包容的集成器件，是嵌入式应用领域的热门话题之一。它指的是在单个芯片上集成一个完整的系统，对所有或部分必要的电子电路进行包分组。所谓完整的系统一般包括中央处理器、存储器以及外围电路等。SoC 是与其他技术并行发展的，如绝缘硅（SOI），它可以提供增强的时钟频率，从而降低微芯片的功耗。

SoC 最大的特点是成功实现了软硬件无缝结合，直接在处理器片内嵌入操作系统的代码模块。而且 SoC 具有极高的综合性，在一个硅片内部运用 VHDL 等硬件描述语言，实现一个复杂的系统。用户不需要再像传统的系统设计一样，绘制庞大复杂的电路板，一点点地连接焊制，只需要使用精确的语言，综合时序设计直接在器件库中调用各种通用处理器的标准，然后通过仿真，就可以直接交付芯片厂商进行生产。由于绝大部分系统构件都在系统内部，整个系统特别简洁，不仅减小了系统的体积和功耗，而且提高了系统的可靠性，提高了设计生产效率。

由于 SoC 往往是专用的，所以大部分都不为用户所知，比较典型的 SoC 产品是 Philips 的 Smart XA。少数通用系列有 Siemens 的 TriCore，Motorola 的 M-Core，另有某些 ARM 系列器件，如 Echelon 和 Motorola 联合研制的 Neuron 芯片等。

系统芯片技术通常应用于小型的，日益复杂的客户电子设备。例如，声音检测设备的系统芯片是在单个芯片上为所有用户提供包括音频接收端、模数转换器（ADC）、微处理器、必要的存储器以及输入输出逻辑控制等设备。此外系统芯片还应用于单芯片无线产品，例如蓝牙设备，支持单芯片 WLAN 和蜂窝电话解决方案。由于系统芯片的高效集成性能，使 SoC 成为了替代集成电路的主要解决方案。SoC 已经成为当前微电子芯片发展的必然趋势。

二、存储器

嵌入式系统需要存储器来存放和执行代码。嵌入式系统的存储器包含 Cache、主存储器和辅助存储器。

1. Cache

Cache 是一种容量小、速度快的存储器阵列，它位于主存和嵌入式微处理器内核之间，存放的是最近一段时间微处理器使用最多的程序代码和数据。在需要进行数据读取操作时，微处理器尽可能地从 Cache 中读取数据，而不是从主存储器中读取，这样就大大改善了系统的性能，提高了微处理器和主存之间的数据传输速率。Cache 的主要目标：减小存储器（如主存器和辅助存储器）给微处理器内核造成的存储器访问瓶颈，使处理速度更快，实时性更强。

在嵌入式系统中 Cache 全部集成在嵌入式微处理器内，可分为数据 Cache、指令 Cache 和混合 Cache，Cache 的大小依不同处理器而定。一般中高档的嵌入式微处理器才会把 Cache 集成进去。

2. 主存储器

主存储器是嵌入式微处理器能直接访问的寄存器，用来存放系统和用户的程序及数据。它可以位于微处理器的内部或外部，其容量为 256KB~1GB，根据具体的应用而定，一般片内存储器容量小，速度快，片外存储器容量大。常用作主存储器的有：ROM 和 RAM。ROM 中的信息一次写入后只能被读出，而不能被操作者修改或删除，一般由芯片制造商进行掩膜写

入，价格便宜，适合于大量应用。ROM 一般用于存放固定的程序，如监控程序，汇编程序等，以及存放各种表格。ROM 可用 Flash、EPROM 和 PROM 等。EPROM（Erasable Programmable ROM）和一般的 ROM 不同点在于它可以用特殊的装置擦除和重写内容，一般用于软件的开发过程。RAM 就是我们平常所说的内存，主要用来存放各种现场的输入、输出数据，中间计算结果，以及与外部存储器交换信息和作为堆栈用。它的存储单元根据具体需要可以读出，也可以写入或改写。RAM 只能用于暂时存放程序和数据、一旦关闭电源或发生断电，其中的数据就会丢失。RAM 可用 SRAM、DRAM 和 SDRAM 等。

3. 辅助存储器

辅助存储器用来存放大数据量的程序代码或信息，它的容量大，但读取速度与主存相比就慢很多，用来长期保存用户的信息。

嵌入式系统中常用的辅助存储器有硬盘、NAND Flash、CF 卡、MMC 和 SD 卡等。

三、通用设备接口和 I/O 接口

嵌入式系统和外界交互需要一定形式的通用设备接口，如 A/D、D/A、I/O 等，外设通过和片外其他设备（或传感器）的连接来实现微处理器的输入/输出功能。每个外设通常都只有单一的功能，它可以在芯片外也可以内置于芯片中。外设的种类很多，可从一个简单的串行通信设备到非常复杂的 802.11 无线设备。I/O 接口有 RS-232 接口（串行通信接口）、Ethernet（以太网接口）、USB（通用串行总线接口）、音频接口、VGA 视频输出接口、I^2C（现场总线）、SPI（串行外围设备接口）和 IrDA（红外线接口）等。

四、总线

总线是指一组互连和传输信息的信号线，是连接系统各个部件之间的桥梁。采用总线结构便于部件和设备的扩充。

嵌入式系统的总线一般分为片内总线和片外总线。片内总线是 CPU 内部的寄存器、算术逻辑部件、控制部件以及总线接口部件之间的公共信息通道。片外总线则泛指 CPU 与外部器件之间的公共信息通道。我们通常所说的总线大多是指片外总线。有的资料上也把片内总线叫内部总线或内总线（Internal Bus），把片外总线叫外部总线或外总线（External Bus）。

总线一般划分为两个结构层：第 1 层是物理层，定义电气参数和总线宽度，如 12 位、32 位或 64 位。第 2 层是协议层，定义处理器与外围设备之间进行数据通信的逻辑规则。

总线的性能指标主要有：总线带宽、总线宽度和总线工作频率。总线的带宽指的是单位时间内总线上可传送的数据量，即我们常说的每秒钟传送多少字节。单位是字节/秒（B/s）或兆字节/秒（MB/s）。总线的宽度指的是总线能同时传送的数据位数，即我们常说的 16 位、32 位、64 位等。在工作频率固定的条件下，总线的带宽与总线的宽度成正比。总线的工作频率即总线的时钟频率，以 MHz 为单位。它是指用于协调总线上的各种操作的时钟信号的频率。工作频率越高则总线工作速度越快。

从微机体系结构来看，有两种总线结构，即单总线结构和多总线结构。在多总线结构中，又以双总线结构为主。单总线结构如图 1-3 所示。计算机的各个部件都与系统总线相连，所以它又称为面向系统的单总线结构。在单总线结构中，各种设备之间都通过系统总线交换信息。单总线结构的优点是控制简单方便，扩充方便。但由于所有设备部件均挂在单一总线上，

使这种结构只能分时工作，即同一时刻只能在两个设备之间传送数据，这就使系统总体数据传输的效率和速度受到限制，这是单总线结构的主要缺点。双总线结构又分为面向 CPU 的双总线结构和面向存储器的双总线结构。

面向 CPU 的双总线结构如图 1-4 所示。其中一组总线是 CPU 与主存储器之间进行信息交换的公共通路，称为存储总线。另一组是 CPU 与 I/O 设备之间进行信息交换的公共通路，称为输入/输出总线(I/O 总线)。外部设备通过连接在 I/O 总线上的接口电路与 CPU 交换信息。由于在 CPU 与主存储器之间、CPU 与 I/O 设备之间分别设置了总线，从而提高了微机系统信息传送的速率和效率。但是由于外部设备与主存储器之间没有直接的通路，它们之间的信息交换必须通过 CPU 才能进行中转，从而降低了 CPU 的工作效率，这是面向 CPU 的双总线结构的主要缺点。

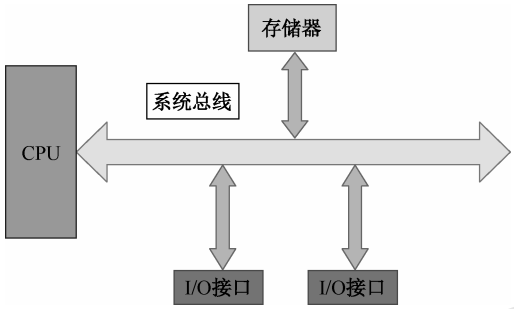

图 1-3　单总线结构

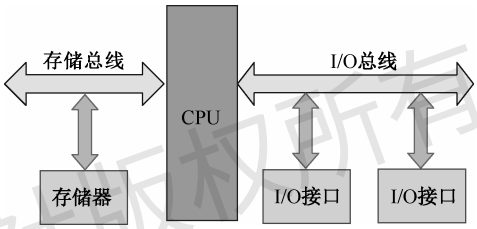

图 1-4　双总线结构

随着对微机性能要求越来越高，现代微机的体系结构已不再采用单总线或双总线的结构，而是采用更复杂的多总线结构。图 1-5 是多总线结构的示意图。

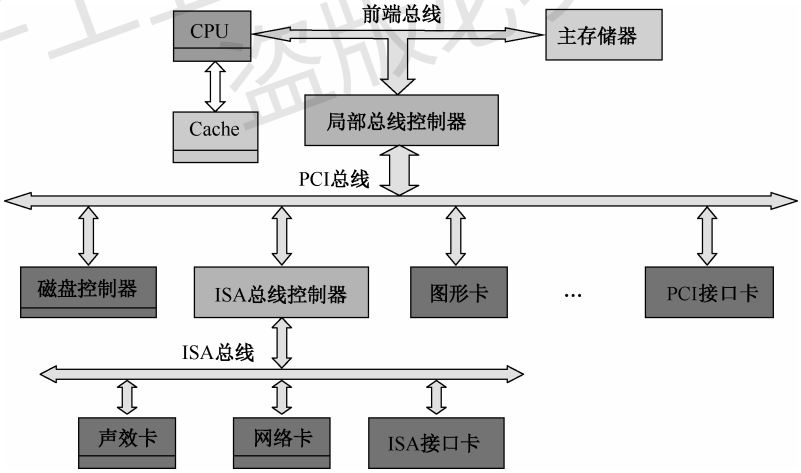

图 1-5　多总线结构

ARM 系统中常用的总线标准有：

1. ISA

ISA（Industry Standard Architecture：工业标准体系结构）是 IBM 公司为 PC/ATJ 计算机而制定的总线标准，为 16 位体系结构，只能支持 16 位的 I/O 设备，数据传输率大约是 16MB/s。也称为 AT 标准。ISA 总线定义如下：IBM-P、RESET、BCLK：复位及总线基本时钟，

BLCK=8MHz；SA19～SA0：存储器及 I/O 空间 20 位地址，带锁存；LA23～LA17：存储器及 I/O 空间 20 位地址，不带锁存；BALE：总线地址锁存，外部锁存器的选通；AEN：地址允许，表明 CPU 让出总线，DMA 开始；SMEMR#、SMEMW#：8 位 ISA 存储器读写控制。ISA 总线引线定义主要信号说明：MEMR#、MEMW#：16 位 ISA 存储器读写控制；SD15～SD0：数据总线，访问 8 位 ISA 卡时高 8 位自动传送到 SD7～SD0；SBHE#：高字节允许，打开 SD15～SD8 数据通路；ISA 总线接口与控制电路；MEMCS16#、IOCS16#：ISA 卡发出此信号确认可以进行 16 位传送；I/OCHRDY：ISA 卡准备好，可控制插入等待周期；NOWS#：不需等待状态，快速 ISA 发出不同插入等待；I/OCHCK#：ISA 卡奇偶校验错；IRQ15、IRQ14、IRQ12～IRQ9、IRQ7～IRQ3：中断请求；DRQ7～DRQ5、DRQ3～DRQ0：ISA 卡 DMA 请求；DACK7#～DACK5#、DACK3#～DACK0#：DMA 请求响应；MASTER#：ISA 主模块确立信号，ISA 发出此信号，与主机内 DMAC 配合使 ISA 卡成为主模块，完全控制总线。C/AT 系统，ISA 从 8 位扩充到 16 位，地址线从 20 条扩充到 24 条。

2. PCI

外设互联标准（或称个人计算机接口，Personal Computer Interface），实际应用中简称为 PCI，是一种连接电子计算机主板和外部设备的总线标准。一般 PCI 设备可分为两种形式：直接布放在主板上的集成电路，在 PCI 规范中称作“平面设备”（Planar Device）；或者安装在插槽上的扩展卡。最早设计出的 PCI 总线工作在 33MHz 频率之下，传输带宽达到 132MB/s，基本上满足了当时处理器的发展需要。随着对更高性能的要求，后来又提出把 PCI 总线的频率提升到 66MHz，传输带宽能达到 264MB/s。1993 年又提出了 64bit 的 PCI 总线，称为 PCI-X，目前广泛采用的是 32bit、33MHz 或者 32bit、66MHz 的 PCI 总线，64bit 的 PCI-X 插槽更多应用于服务器产品。从结构上看，PCI 是在 CPU 和原来的系统总线之间插入的一级总线，具体由一个桥接电路实现对这一层的管理，并实现上下之间的接口以协调数据的传送。管理器提供信号缓冲，能在高时钟频率下保持高性能，适合为显卡、声卡、网卡、MODEM 等设备提供连接接口，工作频率为 33MHz/66MHz。PCI 总线系统要求有一个 PCI 控制卡，它必须安装在一个 PCI 插槽内。这种插槽是目前主板带有最多数量的插槽类型，在当前流行的台式机主板上，ATX 结构的主板一般带有 5～6 个 PCI 插槽，而小一点的 MATX 主板也都带有 2～3 个 PCI 插槽。根据实现方式不同，PCI 控制器可以与 CPU 一次交换 32 位或 64 位数据。

PCI 总线是一种不依附于某个具体处理器的局部总线。管理器提供了信号缓冲，使之能支持 10 种外设，并能在高时钟频率下保持高性能。PCI 总线也支持总线主控技术，允许智能设备在需要时取得总线控制权，以加速数据传送。

3. I^2C 总线

I^2C（Inter Integrated Circuit）总线是由 Philips 公司开发的两线式串行总线，用于连接微控制器及其外围设备。是微电子通信控制领域广泛采用的一种总线标准。它是同步通信的一种特殊形式，具有接口线少，控制方式简单，器件封装形式小，通信速率较高等优点。

I^2C 总线支持任何 IC 生产过程（CMOS、双极性）。通过串行数据（SDA）线和串行时钟（SCL）线在连接到总线的器件间传递信息。每个器件都有一个唯一的地址识别（无论是微控制器、LCD 驱动器、存储器或键盘接口），而且都可以作为一个发送器或接收器（由器件的功能决定）。LCD 驱动器只能作为接收器，而存储器则既可以接收又可以发送数据。除了发

送器和接收器外，器件在执行数据传输时也可以被看作是主机或从机。主机是初始化总线的数据传输并产生允许传输的时钟信号的器件。此时，任何被寻址的器件都被认为是从机。

4. SPI 总线

SPI 是串行外设接口（Serial Peripheral Interface）的缩写。它是一种高速的、全双工的、同步的通信总线，并且在芯片的管脚上只占用四根线，节约了芯片的管脚，同时为 PCB 的布局节省空间，提供方便，正是出于这种简单易用的特性，如今越来越多的芯片集成了这种通信协议，比如 AT91RM9200。

在点对点的通信中，SPI 接口不需要进行寻址操作，且为全双工通信，显得简单高效。在多个从设备的系统中，每个从设备需要独立的使能信号，硬件上比 I^2C 系统要稍微复杂一些。SPI 接口的不足是：没有指定的流控制，没有应答机制确认是否接收到数据。

5. PC/104 总线

PC/104 是一种工业计算机总线标准。PC/104 有两个版本，8 位和 16 位，分别与 PC 和 PC/AT 相对应。PC/104PLUS 则与 PCI 总线相对应，在 PC104 总线的两个版本中，8 位 PC/104 共有 64 个总线管脚，为单列双排插针和插孔，P1：64 针，P2：40 针，合计 104 个总线信号，PC/104 因此得名。当 8 位模块和 16 位模块连接时，16 位模块必须在 8 位模块的下面。P2 总线连接在 8 位元模块中是可选的。

PC/104PLUS 是专为 PCI 总线设计的，可以连接高速外接设备。PC/104PLUS 在硬件上通过一个 120 孔插座实现，PC/104PLUS 包括了 PCI 规范 2.1 版要求的所有信号。为了向下兼容，PC/104PLUS 保持了 PC/104 的所有特性。PC/104PLUS 与 PC/104 相比有 3 个特点：相对 PC/104 连接，增加了第三个连接接口支持 PCI Bus；改变了组件高度的需求，增加模块的柔韧性；加入了控制逻辑单元，以满足高速度 Bus 的需求；由于 PC/104 的管脚定义与 ISA、PCI 的规范完全兼容，所以公司在产品内部用 PC/104 模块时，也可以应自己的需要设计生产更多的专业应用 PC/104 模块种类。

6．CAN 总线

CAN 是控制器局域网络（Controller Area Network）的简称，由研发和生产汽车电子产品著称的德国 BOSCH 公司开发，并最终成为国际标准（ISO 11898）。是国际上应用最广泛的现场总线之一。 在北美和西欧，CAN 总线协议已经成为汽车计算机控制系统和嵌入式工业控制局域网的标准总线，并且拥有以 CAN 为底层协议，专为大型货车和重工机械车辆设计的 J1939 协议。

CAN 是符合 ISO 的串行通信协议。在汽车产业中，出于对安全性、舒适性、方便性、低公害、低成本的要求，各种各样的电子控制系统被开发了出来。由于这些系统之间通信所用的数据类型及对可靠性的要求不尽相同，由多条总线构成的情况很多，线束的数量也随之增加。为适应“减少线束的数量”、“通过多个 LAN，进行大量数据的高速通信”的需要，1986 年德国电气商博世公司开发出面向汽车的 CAN 通信协议。此后，CAN 通过 ISO 11898 及 ISO 11519 进行了标准化，在欧洲已是汽车网络的标准协议。CAN 的高性能和可靠性已被认同，并被广泛地应用于工业自动化、船舶、医疗设备、工业设备等方面。现场总线是当今自动化领域技术发展的热点之一，被誉为自动化领域的计算机局域网。它的出现为分布式控制系统

实现各节点之间实时、可靠的数据通信提供了强有力的技术支持。

CAN 属于现场总线的范畴，它是一种有效支持分布式控制或实时控制的串行通信网络。较之许多 RS-485 基于 R 线构建的分布式控制系统而言，基于 CAN 总线的分布式控制系统在以下方面具有明显的优越性。首先，CAN 控制器工作于多种方式，网络中的各节点都可根据总线访问优先权（取决于报文标识符）采用无损结构的逐位仲裁方式竞争，向总线发送数据，且 CAN 协议废除了站地址编码，而代之以对通信数据进行编码，这可使不同的节点同时接收到相同的数据，这些特点使得 CAN 总线构成的网络各节点之间的数据通信实时性强，并且容易构成冗余结构，提高系统的可靠性和灵活性。其次，CAN 总线通过 CAN 收发器接口芯片 82C250 的两个输出端 CANH 和 CANL 与物理总线相连，而 CANH 端的状态只能是高电平或悬浮状态，CANL 端只能是低电平或悬浮状态，这就不会出现多节点同时向总线发送数据导致总线呈现短路，从而损坏某些节点的现象。而且 CAN 节点在错误严重的情况下具有自动关闭输出功能，以使总线上其他节点的操作不受影响，从而保证不会出现因个别节点出现问题，使得总线“死锁”的现象。而且，CAN 具有的完善的通信协议，可由 CAN 控制器芯片及其接口芯片来实现，从而大大降低系统开发难度，缩短了开发周期。另外，与其他现场总线比较而言，CAN 总线是具有通信速率高、容易实现、且性价比高等诸多特点的一种已形成国际标准的现场总线。上述这些也是 CAN 总线应用于众多领域，具有强劲的市场竞争力的重要原因。与一般的通信总线相比，CAN 总线的数据通信具有突出的可靠性、实时性和灵活性。由于其良好的性能及独特的设计，CAN 总线越来越受到人们的重视。它在汽车领域上的应用是最广泛的，世界上一些著名的汽车制造厂商，如奔驰、宝马、保时捷、劳斯莱斯和美洲豹等都采用了 CAN 总线来实现汽车内部控制系统与各检测和执行机构间的数据通信。同时，由于 CAN 总线本身的特点，其应用范围已不再局限于汽车行业，而向自动控制、航空航天、航海、过程工业、机械工业、纺织机械、农用机械、机器人、数控机床、医疗器械及传感器等领域发展。CAN 已经形成国际标准，并已被公认为几种最有前途的现场总线之一。其典型的应用协议有：SAE J1939/ISO11783、CANOpen、CANaerospace、DeviceNet、NMEA 2000 等。

由于 CAN 为越来越多不同领域所采用和推广，导致要求各种应用领域通信报文的标准化。为此，1991 年 9 月 Philips Semiconductors 制定并发布了 CAN 技术规范（VERSION 2.0）。该技术规范包括 A 和 B 两部分。2.0A 给出了曾在 CAN 技术规范版本 1.2 中定义的 CAN 报文格式，能提供 11 位地址；而 2.0B 给出了标准的和扩展的两种报文格式，提供 29 位地址。此后，1993 年 11 月 ISO 正式颁布了道路交通运载工具 —— 数字信息交换 —— 高速通信控制器局域网（CAN）国际标准（ISO 11898），为控制器局域网标准化、规范化推广铺平了道路。

对于 CAN 总线来讲，数据通信没有主从之分，任意一个节点可以向任何其他（一个或多个）节点发起数据通信，靠各个节点信息优先级先后顺序来决定通信次序，高优先级节点信息在 134μs 以内通信；多个节点同时发起通信时，优先级低的避让优先级高的，不会对通信线路造成拥塞；通信距离最远可达 10km（速率低于 5Kbps），速率可达到 1Mbps（通信距离小于 40m）；CAN 总线传输介质可以是双绞线，同轴电缆；CAN 总线适用于大数据量短距离通信或者长距离小数据量通信，实时性要求比较高，多主多从或者各个节点平等的现场中使用。

知识 2　PXA255 最小系统

一、PXA255 系统框架示意图

PXA255 系统框架示意如图 1-6 所示。

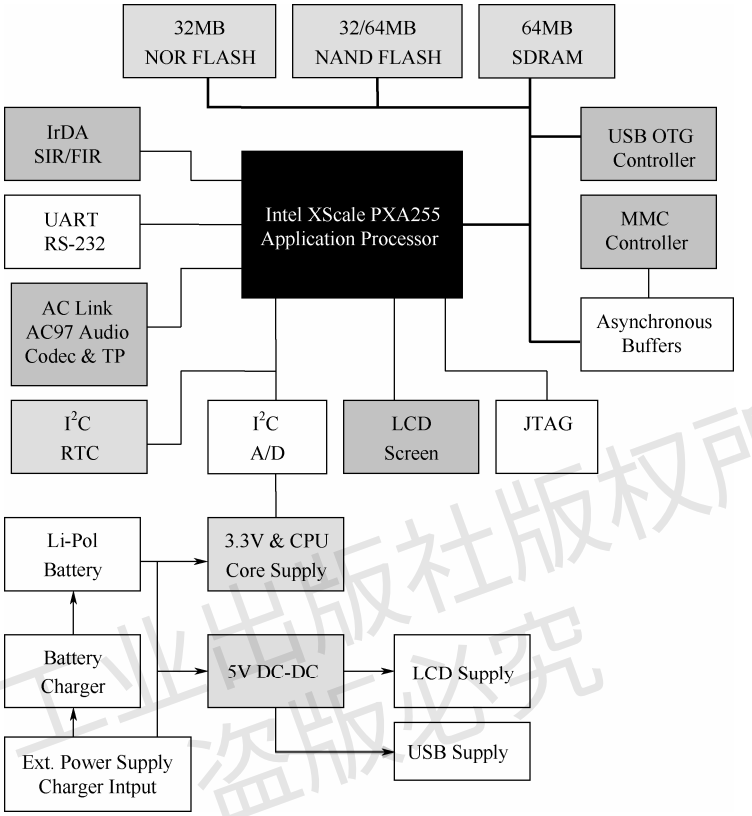

图 1-6　PXA255 系统框架示意图

二、Intel Xscale PXA255 处理器介绍

英特尔公司在 2000 年 9 月推出了基于 StrongARM 处理器的面向无线互联网的嵌入式系统架构 —— Intel 个人互联网用户架构 PCA，该架构可以分为应用、通信、内存等三个子系统，各个子系统之间可以以模块方式集成、扩充。PCA 应用子系统是基于处理器的可编程计算环境，在嵌入式操作系统的支持下，能够进行用户输入输出设备、扩充设备内存、电源等的管理以及与通信子系统进行交互通信。PCA 通信子系统由一个或多个处理器构成，它完成通信协议的处理任务。PCA 的内存子系统提供具有 Intel 特色的低电压、低功耗和高度集成的 FLASH、SRAM 和 DRAM，可以支持分级存储、高速缓存、片上内存、系统内存和拆卸内存等。

2002 年 2 月 25 日，英特尔公司正式推出了基于 Xscale 技术为新一代无线手持应用产品开发的嵌入式处理器 PXA255。它的内核和 ARM 架构 V5TE 结构兼容，集成了多种微结构的特点，内置 JTAG 调试接口、存储器控制器、实时时钟及系统时钟、通用及红外串行接口、蓝牙接口、AC97 接口、扩展卡接口、LCD 控制器、电源管理模块等等。它主要针对高性能的 PDA 市场，为支持视频流、MP3、无线互联网存取以及其他前沿技术而设计。XScale PXA255

芯片结构如图 1-7 所示。

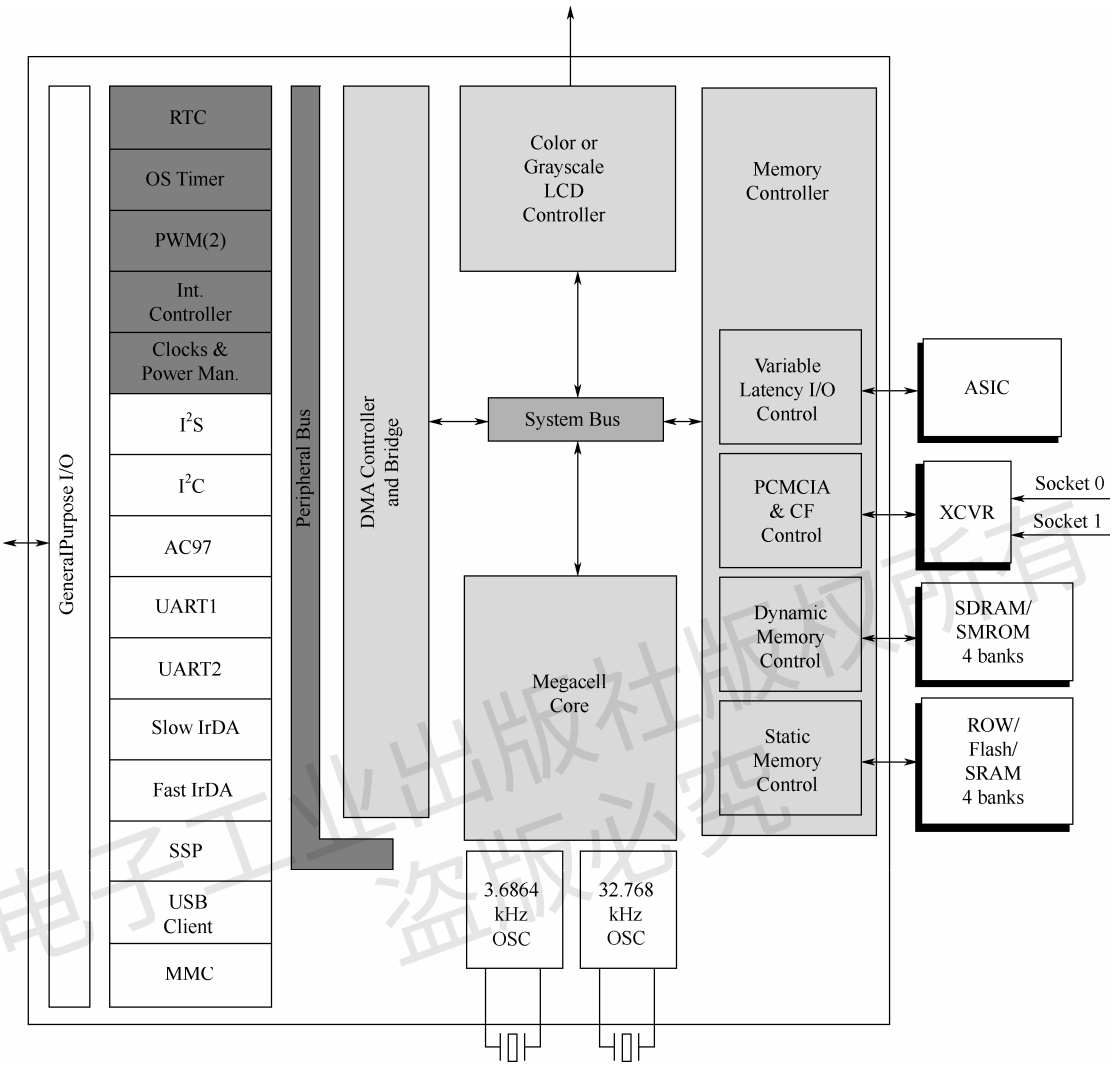

图 1-7　Xscale PXA255 芯片结构图

从图可以看到 PXA255 除了采用了 Xscale 的核外，还集成了众多的外设，比如 DMA 控制器、Memory 控制器、LCD 控制器、UART、AC97、I^2C 等。

三、Xscale 微架构系统结构

Xscale 核是采用 ARM V5TE 架构的处理器，是 Intel 公司的 StongARM 的升级换代产品。具有高性能、低功耗的特点。但它以核的形式作为 ASSP（Application Specific Standard Productor）的构件（Building Block）。PXA255 应用处理机就是为手持式设备设计的 ASSP。Xscale 微架构的系统结构特性如图 1-8 所示。

从图中可以看出，XScale 微架构采用了 ARM V5TE 架构，具有以下显著的特性：7 级超级流水线；乘/累加器 MAC（Multiply/Accumulate）：DSP 功能的 40 位乘累加器；单周期的 16×32 位操作；单指令多数据流 SIMD 的 16 位操作；存储器管理部件 MMU：识别可快存和不可快存（Cacheable 和 Non-cacheable）编码；写回和写直通；允许存储外部存储器的

写缓冲器合并操作；允许数据写分配策略；允许 Xscale 扩展的页面属性操作；指令 Cache：32KB，32 路组相联映像，32 字节/行；循环代替算法；支持锁操作，以提高指令 Cache 的效率；2KB 微小型指令 Cache，2 路组相关映像，32 字节/行，只用于常驻在核内的软件调试；分支目标缓冲器 BTB：128 入口的直接映像 Cache；数据 Cache：32KB，32 路组相关联映像，32 字节/行；循环替代算法；支持锁操作，提高数据 Cache 效率；2KB 微小型数据 Cache，2 路组相联映像，32 字节/行（用于大型流媒体数据）；填入缓冲器：4～8 入口；提高外部存储器的数据读取；相关的暂挂缓冲器；写缓冲器：8 入口；支持合并操作；性能监视：2 个性能监视计数器（监视 Xscale 核的各种事件）；允许用软件测量 Cache 效率，监测系统瓶颈以及程序总的时延；电源管理；时钟管理；调试：测试访问端口 TAP 控制器；支持 JTAG 的标准测试访问端口及边界扫描。

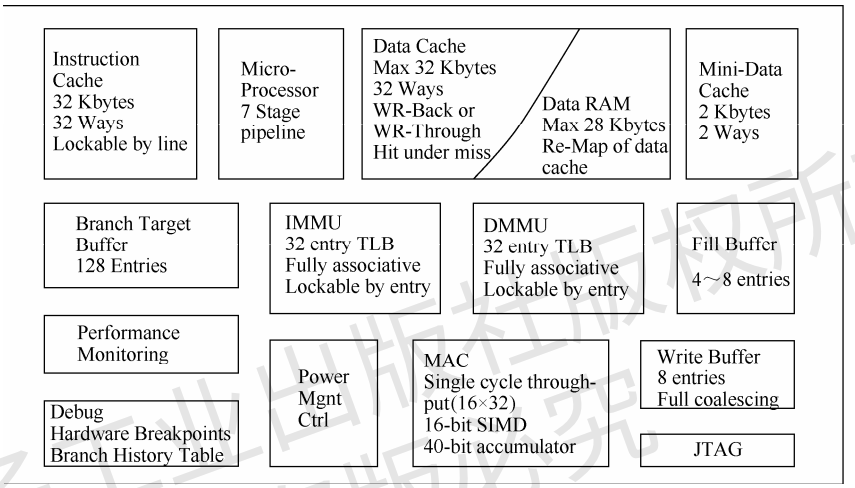

图 1-8　Xscale 微架构的系统结构特性图

任务三　嵌入式系统软件

嵌入式软件就是嵌入在硬件中的操作系统和开发工具软件，它广泛应用于国防、工控、家用、商用、办公、医疗等领域，如我们常见的移动电话、掌上机、数码相机、机顶盒、MP3 播放器等都是用嵌入式软件技术对传统产品进行智能化改造的结果。

知识 1　引导程序

在嵌入式操作系统中，BootLoader 在操作系统内核运行之前运行。它可以初始化硬件设备、建立内存空间映射图，从而将系统的软硬件环境带到一个合适状态，以便为最终调用操作系统内核准备好正确的环境。在嵌入式系统中，通常并没有像 BIOS 那样的固件程序（有的嵌入式 CPU 也会内嵌一段短小的启动程序），因此整个系统的加载启动任务就完全由 BootLoader 来完成。在一个基于 ARM7TDMI 核心的嵌入式系统中，系统在上电或复位时通常都从地址 0x00000000 处开始执行，而在这个地址处安排的通常就是系统的 BootLoader 程序。

一、BootLoader 概述

1. BootLoader 所支持的 CPU 和嵌入式开发板

每种不同的 CPU 体系结构都有不同的 BootLoader。有些 BootLoader 也支持多种体系结构的 CPU，如后面要介绍的 U-Boot 就同时支持 ARM 体系结构和 MIPS 体系结构。除了依赖于 CPU 的体系结构外，BootLoader 实际上也依赖于具体的嵌入式板级设备的配置。

2. BootLoader 的安装媒介

系统加电或复位后，所有的 CPU 通常都从某个由 CPU 制造商预先安排的地址上取指令。而基于 CPU 构建的嵌入式系统通常都有某种类型的固态存储设备（如 ROM、EEPROM 或 FLASH 等）被映射到这个预先安排的地址上。因此在系统加电后，CPU 将首先执行 BootLoader 程序。

3. BootLoader 的启动过程

BootLoader 的启动过程分为单阶段和多阶段两种。通常多阶段的 BootLoader 能提供更为复杂的功能以及更好的可移植性。从固态存储设备上启动的 BootLoader 大多都是 2 阶段的启动过程，启动过程分为 stage1 和 stage2 两部分。

4. BootLoader 的操作模式

大多数 BootLoader 都包含两种不同的操作模式：“ 启动加载 ” 模式和 “ 远程下载 ” 模式，这两种操作模式的区别仅对于开发人员才有意义。但从最终用户的角度看，BootLoader 的作用就是用来加载操作系统的，而并不存在所谓的 “ 启动加载 ” 模式和 “ 远程下载 ” 模式的区别。

启动加载模式：这种模式也称为 “ 自主 ” 模式。也就是 BootLoader 从目标机上的某个固态存储设备上将操作系统加载到 RAM 中运行，整个过程并没有用户的介入。这种模式是嵌入式产品发布时的通用模式。下载模式：在这种模式下，目标机上的 BootLoader 将通过串口或网络等通行手段从开发主机（Host）上下载内核映像等到 RAM 中。可以被 BootLoader 写到目标机上的固态存储媒质中，或者直接进入系统的引导。也可以通过串口接收用户的命令。比如：下载内核映像和根文件系统映像等。从主机下载的文件通常首先被 BootLoader 保存到目标机的 RAM 中，然后再被 BootLoader 写到目标机上的 FLASH 类固态存储设备中。BootLoader 的这种模系统是在更新时使用的。工作于这种模式下的 BootLoader 通常都会向它的终端用户提供一个简单的命令行接口。

5. BootLoader 与主机之间进行文件传输所用的通信设备及协议

常见的情况是目标机上的 BootLoader 通过串口与主机之间进行文件传输，传输协议通常是 xmodem/ ymodem/zmodem 协议中的一种。但是，串口传输的速度是有限的，因此通过以太网连接并借助 TFTP 协议来下载文件是个更好的选择。

二、BootLoader 层次介绍

一个嵌入式 Linux 系统从软件的角度看通常可以分为四个层次：

1．引导加载程序。包括固化在固件（Firmware）中的 Boot 代码（可选）和 BootLoader 两大部分。

2．Linux 内核。特定于嵌入式板子的定制内核以及内核的启动参数。

3．文件系统。包括根文件系统和建立于 Flash 内存设备之上的文件系统。通常用 RamDisk 来作为 Rootfs。

4．用户应用程序。特定于用户的应用程序。有时在用户应用程序和内核层之间可能还会包括一个嵌入式图形用户界面。常用的嵌入式 GUI 有：MicroWindows 和 MiniGUI 等。

引导加载程序是系统加电后运行的第一段软件代码。PC 机中的引导加载程序由 BIOS 和位于硬盘 MBR 中的 OS BootLoader 一起组成。BIOS 在完成硬件检测和资源分配后，将硬盘 MBR 中的 BootLoader 读到系统的 RAM 中，然后将控制权交给 OS BootLoader。BootLoader 的主要运行任务就是将内核映像从硬盘上读到 RAM 中，然后跳转到内核的入口点去运行，也即开始启动操作系统。

对于嵌入式系统，BootLoader 是基于特定硬件平台来实现的。因此，几乎不可能为所有的嵌入式系统建立一个通用的 BootLoader，不同的处理器架构都有不同的 BootLoader。BootLoader 不但依赖于 CPU 的体系结构，而且依赖于嵌入式系统板级设备的配置。对于两块不同的嵌入式开发板而言，即使它们是基于同一种 CPU 构建的，要想让运行在一块板子上的 BootLoader 程序也能运行在另一块板子上，通常也需要修改 BootLoader 的源程序。尽管如此，仍然可以对 BootLoader 归纳出一些通用的概念来，用以指导用户特定的 BootLoader 设计与实现。

三、BootLoader 启动流程

BootLoader 的启动流程一般分为两个阶段：Stage1 和 Stage2，下面分别对这两个阶段进行讲解。

1．BootLoader 的 Stage1

在 Stage1 中 BootLoader 主要完成以下工作：基本的硬件初始化，包括屏蔽所有的中断、设置 CPU 的速度和时钟频率、RAM 初始化、初始化 LED、关闭 CPU 内部指令和数据 Cache 灯。

为加载 Stage2 准备 RAM 空间，通常为了获得更快的执行速度，通常把 Stage2 加载到 RAM 空间中来执行，因此必须为加载 BootLoader 的 Stage2 准备好一段可用的 RAM 空间范围。设置堆栈指针 sp，这是为执行 Stage2 的 C 语言代码做好准备。

2．BootLoader 的 Stage2

在 Stage2 中 BootLoader 主要完成用汇编语言跳转到“ main ”入口函数。由于 Stage2 的代码通常用 C 语言来实现，目的是实现更复杂的功能和取得更好的代码可读性和可移植性。但是与普通 C 语言应用程序不同的是，在编译和链接 BootLoader 这样的程序时，不能使用‘ glibc ”库中的任何支持函数。

初始化本阶段要使用到的硬件设备，包括初始化串口、初始化计时器等。在初始化这些设备之前可以输出一些打印信息。检测系统的内存映射，所谓内存映射就是指在整个 4GB 物理地址空间中指出哪些地址范围被分配用来寻址系统的 RAM 单元。加载内核映像和根文件系统映像，这里包括规划内存占用的布局和从 Flash 上复制数据。设置内核的启动参数。

四、常见 BootLoader 介绍

1．Redboot 介绍

Redboot 是 RedHat 公司随 eCos 发布的一个 Boot 方案，是嵌入式操作系统 eCos 的一个最小版本，它是一个开源项目。Redboot 是标准的嵌入式调试和引导解决方案，是一个专门为嵌

入式系统定制的引导工具。

Redboot 支持的处理器构架有 ARM、MIPS、MN10300、PowerPC、Renesas SHx、v850、x86 等，是一个完善的嵌入式系统 Boot Loader。

Redboot 是在 eCos 的基础上剥离出来的，继承了 eCos 的简洁、轻巧、可灵活配置、稳定可靠等优点。它可以使用 X-modem 或 Y-modem 协议经由串口下载，也可以经由以太网口通过 BOOTP/DHCP 服务获得 IP 参数，使用 TFTP 方式下载程序映像文件，常用于调试支持和系统初始化（Flash 下载更新和网络启动）。Redboot 可以通过串口和以太网口与 GDB 进行通信，调试应用程序，甚至能中断被 GDB 运行的应用程序。Redboot 为管理 FLASH 映像、映像下载、Redboot 配置以及其他如串口、以太网口提供了一个交互式命令行接口，自动启动后，Redboot 用来从 TFTP 服务器或者从 Flash 下载映像文件，加载系统的引导脚本文件保存在 Flash 上。Redboot 是标准的嵌入式调试和引导解决方案，支持几乎所有的处理器构架以及大量的外围硬件接口，并且还在不断地完善过程中。

2. ARMboot 介绍

ARMboot 是一个 ARM 平台的开源固件项目，它严重依赖于 PPCBoot，一个为 PowerPC 平台上的系统提供类似功能的姊妹项目。

ARMboot 支持的处理器构架有 StrongARM、ARM720T、PXA250 等，是为基于 ARM 或者 StrongARM CPU 的嵌入式系统所设计的。ARMboot 的目标是成为通用的、容易使用和移植的引导程序，非常轻便地运用于新的平台上。ARMboot 特性为：支持多种类型的 FLASH；允许映像文件经由 BOOTP、DHCP、TFTP 从网络传输；支持串行口下载 S-record 或者 binary 文件；允许内存的显示及修改；支持 jffs2 文件系统等。

Armboot 对 S3C44B0 板的移植相对简单，在经过删减后，仅仅需要完成初始化、串口收发数据、启动计数器和 FLASH 操作等步骤，就可以下载引导 uCLinux 内核完成板上系统的加载。总体来说，ARMboot 介于大、小型 BootLoader 之间，相对轻便，基本功能完备，缺点是缺乏后续支持。

3. U-Boot 介绍

U-Boot（Universal BootLoader）是遵循 GPL 条款的开放源代码项目，它是由开源项目 PPCBoot 发展起来的，ARMboot 并入了 PPCBoot，和其他一些架构的 Loader 合称 U-Boot。U-Boot 涵盖了绝大部分处理器构架，提供了大量外设驱动，支持多个文件系统，附带调试、脚本、引导等工具，特别支持 Linux，为板级移植做了大量的工作。U-Boot1.1.1 版本特别包含了对 SA1100 和 44B0 芯片的移植，所以 44B0 移植主要是针对 Board 的移植，包括 FLASH、内存配置以及串口波特率等。U-Boot 的完整功能性和后续不断的支持，使系统的升级维护变得十分方便。

U-Boot 支持的处理器构架包括 PowerPC（MPC5xx，MPC8xx，MPC82xx，MPC7xx，MPC74xx，4xx）、ARM（ARM7，ARM9，StrongARM，Xscale）、MIPS、x86 等，它是在 GPL 下资源代码最完整的一个通用 BootLoader。

U-Boot 主要特性为：SCC/FEC 以太网支持；BOOTP/TFTP 引导；IP、MAC 预置功能；在线读写 FLASH、DOC、IDE、IIC、EEROM、RTC；支持串行口 kermit、S-record 下载代码；识别二进制、ELF32、pImage 格式的 Image，对 Linux 引导有特别的支持；监控（minitor）命令集：读写 I/O、内存、寄存器、内存、外设测试功能等；脚本语言支持（类似 BASH 脚本）；

支持 WatchDog，LCD logo，状态指示功能等。

4. Blob 介绍

Blob（Boot Loader Object）是由 Jan-Derk Bakker and Erik Mouw 发布的，是专门为 StrongARM 构架下的 LART 设计的 Boot Loader。

Blob 支持 SA1100 的 LART 主板，但用户也可以自行修改移植。Blob 提供两种工作模式，在启动时处于正常的启动加载模式，但是它会延时 10 秒等待终端用户按下任意键而将 Blob 切换到下载模式。如果在 10 秒内没有用户按键，则 Blob 继续启动 Linux 内核。其基本功能为：初始化硬件（CPU 速度，存储器，中断，RS 232 串口）；引导 Linux 内核并提供 Ramdisk；给 LART 下载一个内核或者 Ramdisk；给 FLASH 片更新内核或者 Ramdisk；测定存储配置并通知内核；给内核提供一个命令行；Blob 功能比较齐全，代码较少，比较适合做修改移植，用来引导 Liunx，目前大部分 S3C44B0 板都将 Blob 修改移植后来加载 uCLinux。

5. Bios-lt 介绍

Bios-lt 专门支持三星（Samsung）公司 ARM 构架处理器 S3C4510B，可以设置 CPU/ROM/SDRAM/EXTIO，管理并烧写 FLASH，装载引导 uCLinux 内核。Bios-lt 还提供了 S3C4510B 的一些外围驱动。

6. Bootldr 介绍

Bootldr 是康柏（Compaq）公司发布的，类似于 Compaq iPAQ Pocket PC，支持 SA1100 芯片。它被推荐用来引导 Linux，支持串口 Y-modem 协议以及 jffs 文件系统。

7. Vivi 介绍

Vivi 是韩国 Mizi 公司开发的，适用于 ARM9 处理器。和所有的 BootLoader 一样，Vivi 有两种工作模式：启动加载模式和下载模式。启动加载模式可以在一段时间后（这个时间可更改）自行启动 Linux 内核，这是 Vivi 的默认模式。在下载模式下，Vivi 为用户提供一个命令行接口，通过接口可以使用 Vivi 提供的一些命令。

Vivi 作为一种 BootLoader，其运行过程分成两个阶段。第一阶段的代码在 vivi/arch/s3c2410/head.s 中定义，大小不超过 10 KB，它包括从系统上电后在 0x00000000 地址开始执行的部分。这部分代码运行在 Flash 中，它包括对 S3C2410 的一些寄存器、时钟等的初始化并跳转到第二阶段执行。第二阶段的代码在 vivi\init\main.c 中，主要进行开发板初始化、内存映射和内存管理单元初始化等工作，最后会跳转到 boot_or_vivi()函数中接收命令并进行处理。需要注意的是在 Flash 中执行完内存映射后，会将 Vivi 代码复制到 SDRAM 中执行。

知识 2　操作系统

一、操作系统概述

1. 操作系统

操作系统（Operating System，OS）是计算机用户和计算机硬件之间的媒介，它是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行。

操作系统是用户和计算机的接口，同时也是计算机硬件和其他软件的接口。操作系统的

功能包括管理计算机系统的硬件、软件及数据资源、控制程序运行、改善人机界面、为其他应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，提供了各种形式的用户界面，使用户有一个好的工作环境，为其他软件的开发提供必要的服务和相应的接口。实际上，用户是不用接触操作系统的，操作系统管理着计算机硬件资源，同时根据应用程序的资源请求，为其分配资源。

操作系统的种类相当多，各种设备安装的操作系统根据从简单到复杂，可分为智能卡操作系统、实时操作系统、传感器节点操作系统、嵌入式操作系统、个人计算机操作系统、多处理器操作系统、网络操作系统和大型机操作系统。按应用领域划分主要有三种：桌面操作系统、服务器操作系统和嵌入式操作系统。

操作系统一般提供以下服务：

（1）程序运行

一个程序的运行离不开操作系统的配合，其中包括指令和数据载入内存、I/O 设备和文件系统的初始化等。

（2）I/O 设备访问

每种 I/O 设备的管理和使用都有其自身的特点，而操作系统接管了这些工作，从而使得用户在使用这些 I/O 设备的过程中会感觉更方便。

（3）文件访问

文件访问不仅需要熟悉相关 I/O 设备（磁盘驱动器等）的特点，而且还要熟悉相关的文件格式。另外，对于多用户操作系统或者网络操作系统，从计算机安全角度考虑，需要对文件的访问权限做出相应的规定和处理。这些都是操作系统所要完成的工作。

（4）系统访问

对于一个多用户或者网络操作系统而言，操作系统需要对用户系统访问权限做出相应的规定和处理。

（5）错误检测和反馈

当操作系统运行时，会出现这样那样的问题。操作系统应当提供相应的机制来检测这些信息，并且能对某些问题给出合理的处理方法，或者向用户提供相应的报告信息。

（6）系统使用纪录

在一些现代操作系统中，出于系统性能优化或者系统安全角度考虑，操作系统会对用户使用过程记录相关信息。

（7）程序开发

一般操作系统都会提供丰富的 API 供程序员开发应用程序；很多程序编辑工具，集成开发环境等也都是通过操作系统提供的。而计算机有很多资源，它们分别用于数据的传输、处理或存储以及这些操作的控制。这些资源的管理工作就交给了操作系统。

2. 操作系统发展史

（1）串行处理系统

在二十世纪四五十年代，电子计算机发展初期，没有操作系统的概念，人们通过显示灯、跳线、某些输入输出设备同计算机打交道。当需要执行某个计算机程序时，人们通过输入设备将程序灌入计算机中，然后等待运行结果。如果中间出现错误，程序员就得检查计算机寄存器，内存甚至是一些元器件以找出原因所在。如果顺利完成，结果就从打印机上打印出来。

人们称这种工作方式为串行处理方式。

（2）简单批处理系统

由于早期的计算机系统十分昂贵，人们希望通过某种方式提高计算机的利用率。于是批处理的概念就被引入了。

在早期的批处理系统中，功能相对比较简单，其核心思想就是借助某个称为监视器的软件，用户不需要直接和计算机硬件打交道，而只需要将自己所要完成的计算任务提交给计算机操作员。在操作员那里，所有计算任务按照一定的顺序被成批输入计算机中。当某个计算任务结束之后，监视器会自动开始执行下一个计算任务。

（3）多道程序设计批处理系统

即便是采用了批处理技术，也不能对计算机资源进行有效利用。一个很头疼的问题就是I/O 设备的操作速度往往比处理器慢很多。当某个批处理任务需要访问 I/O 设备的时候，处理器往往处于空闲状态。基于这方面的考虑，多道程序设计思想被引入了批处理系统中。通常，多道程序设计也可被称为多任务，即多道程序设计批处理系统也可称为多任务批处理系统。

多道程序设计思想的引入允许某个计算任务在等待 I/O 操作的时候，计算机可以转而执行其他计算任务。从而提高处理器的利用率。

（4）分时系统

在多任务批处理系统中，计算机资源的利用率得到了很大提高。问题是如果用户希望能够干预计算任务的执行该怎么办？此时我们需要引入一种交互模式来实现这一功能，分时的概念就由此产生。在分时系统中，处理器时间按照一定的分配策略在多个用户中间共享。在实际的单处理器系统中，多个任务交替获取处理器控制权，交替执行，从而提供更好的交互性能。

（5）现代操作系统

现代操作系统技术是在综合了以上四种典型的操作系统技术的基础上提出的操作系统实现方式，它适应了现代计算机系统管理和使用的要求。其主要特征是多任务、分时，而且很多系统都开始陆续加入多用户功能。现代操作系统一般包括：进程及进程管理；内存及虚拟管理；信息保护和安全；调度和资源管理；模块化系统化设计。

3. 嵌入式操作系统

嵌入式操作系统（Embedded Operating System，EOS）是指用于嵌入式系统的操作系统。嵌入式操作系统是一种用途广泛的系统软件，通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。嵌入式操作系统负责嵌入式系统的全部软、硬件资源的分配、任务调度，控制、协调并发活动。它必须体现其所在系统的特征，能够通过装卸某些模块来达到系统所要求的功能。目前在嵌入式领域广泛使用的操作系统有：Linux、Windows CE、μC/OS-II、VxWorks 等，以及应用在智能手机和平板电脑的 Android、iOS 等。

Linux 是一套免费使用和自由传播的类 UNIX 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。它能运行主要的 UNIX 工具软件、应用程序和网络协议；支持 32 位和 64 位硬件。Linux 继承了 UNIX 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。Linux 可安装在各种计算机硬件设备中，比如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机。

在嵌入式系统应用方面，Linux 小得可以放在一张软盘上运行。为实时系统而开发的各种

RTLinux（Real-Time Linux），可以让 Linux 支持硬件实时任务。Linux 的开放式原则使得 Linux 下的驱动和升级越来越多，越来越快。

Windows CE 是微软公司推出的面向移动智能连接设备的模块化实时嵌入式操作系统。凭借其广泛的适应性，丰富的功能，强大的多媒体能力和友好的开发环境，Windows CE 已经被广泛地应用于掌上机，智能手机，汽车电子，信息终端等领域。

简单地说，Windows CE 就是基于掌上型设备的电子设备操作系统。其中 CE 中的 C 代表袖珍（Compact）、消费（Consumer）、通信能力（Connectivity）和伴侣（Companion）；E 代表电子产品（Electronics）。它是一个抢占式多任务式的、具有强大通信能力的 Win32 嵌入式操作系统，是微软专门为信息设备、移动应用、消费电子产品、嵌入式等非 PC 领域而从头设计的战略性操作系统产品。Windows CE 的设计目标是：模块化及可伸缩性、实时性能好，通信能力强大，支持多种 CPU。

从操作系统的内核角度看，Windows CE 具有灵活的电源管理功能，包括睡眠/唤醒模式。在 Windows CE 中，还使用了对象存储技术，包括文件系统、注册表及数据库。它还具有很多高性能、高效率的操作系统特性，包括按需换页、共享存储、交叉处理同步、支持大容量对等。

Windows CE 具有良好的通信能力。它广泛支持各种通信硬件，亦支持直接的局域网连接以及拨号连接，并提供与 PC、内部网以及 Internet 的连接，包括用于应用级数据传输的设备至设备间的连接。在提供各种基本的通信基础结构的同时，Windows CE 还提供与 Windows 9x／NT 的最佳集成和通信。

Windows CE 的图形用户界面相当出色。它拥有基于 Microsoft Internet Explorer 的 Internet 浏览器，此外，还支持 TrueType 字体。开发人员可以利用丰富灵活的控件库在 Windows CE 环境下为嵌入式应用建立各种专门的图形用户界面。Windows CE 甚至还能支持诸如手写体和声音识别、动态影像、3D 图形等特殊应用。

μC/OS-II 是 Jean J.Labrosse 开发的一种小型嵌入式操作系统，是一种基于优先级的抢占式多任务实时操作系统，包含了实时内核、任务管理、时间管理、任务间通信同步（信号量、邮箱、消息队列）和内存管理等功能。它主要面向中小型嵌入式系统，具有执行效率高、占用空间小、可移植性强、实时性能优良和可扩展性强等特点。μC/OS-II 结构小巧，最小内核可编译至 2KB，即使包含全部功能如信号量、消息邮箱、消息队列及相关函数等，编译后的内核也仅有 6～10KB；扩展性能好，如果需要，可自行加入文件系统等。

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统（RTOS），是嵌入式开发环境的关键组成部分，具有良好的持续发展能力、高性能的内核以及友好的用户开发环境，在嵌入式实时操作系统领域占据一席之地。VxWorks 操作系统以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通信、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18 战斗机、B-2 隐形轰炸机和爱国者导弹上，甚至火星探测器上也都使用到了 VxWorks。

嵌入式 Vxworks 系统的主要应用领域有以下几方面：数据网络、远程通信、医疗设备、消费电子、交通运输、工业、航空航天和多媒体等。VxWorks 的系统结构是一个相当小的微内核的层次结构。内核仅提供多任务环境、进程间通信和同步功能。这些功能模块足够支持 VxWorks 在较高层次所提供的丰富的性能要求。

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android 操作系统最初由 Andy Rubin 开发，主要支持手机。第一部 Android 智能手机发布于 2008 年 10 月。Android 逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。

iOS 是由苹果公司开发的移动操作系统。苹果公司最早于 2007 年 1 月 9 日的 Macworld 大会上公布这个系统，最初是设计给 iPhone 使用的，后来陆续套用到 iPod、iPad 以及 Apple TV 等产品上。iOS 与苹果的 Mac OS X 操作系统一样，它也是以 Darwin 为基础的，因此同样属于类 UNIX 的商业操作系统。原本这个系统名为 iPhone OS，因为 iPad，iPhone，iPod 都使用 iPhone OS，所以 2010WWDC 大会上宣布改名为 iOS。

从软件上，主要可以依据操作系统的类型来划分。嵌入式系统的软件主要有两大类：实时系统和分时系统。其中实时系统又分为两类：硬实时系统和软实时系统。

实时嵌入系统是为执行特定功能而设计的，可以严格按时序执行功能。其最大的特征就是程序的执行具有确定性。在实时系统中，如果系统在指定的时间内未能实现某个确定的任务，会导致系统的全面崩溃，则系统被称为硬实时系统。而在软实时系统中，虽然响应时间同样重要，但是超时却不会导致致命错误。一个硬实时系统往往在硬件上需要添加专门用于时间和优先级管理的控制芯片，而软实时系统则主要在软件方面通过编程实现时限的管理。比如 Windows CE 就是一个多任务分时系统，而 μC/OS-II 则是典型的实时操作系统。

知识 3　应用软件

嵌入式应用软件是针对特定应用领域，基于某一固定的硬件平台，用来达到用户预期目标的计算机软件。由于用户任务可能有时间和精度上的要求，因此有些嵌入式应用软件需要特定嵌入式操作系统的支持。嵌入式应用软件和普通应用软件有一定的区别，它不仅要求准确性、安全性和稳定性等方面能够满足实际应用的需要，而且还要尽可能地进行优化，以减少对系统资源的消耗，降低硬件成本。目前我国市场上已经出现了各式各样的嵌入式应用软件，包括浏览器、E-mail 软件、文字处理软件、通信软件、多媒体软件、个人信息处理软件、智能人机交互软件、各种行业应用软件等。嵌入式系统中的应用软件是最活跃的力量，每种应用软件均有特定的应用背景，尽管规模较少，但专业性较强，所以嵌入式应用软件不像操作系统和支撑软件那样受制于国外产品垄断，是我国嵌入式软件的优势领域。

知识 4　嵌入式系统开发流程

一、嵌入式系统开发模式

嵌入式系统开发分为软件开发部分和硬件开发部分。嵌入式系统在开发过程一般都采用如图 1-9 所示的“宿主机/目标板”开发模式，开发时使用宿主机上的交叉编译工具链（包括编译、汇编及链接工具）来生成目标板上运行的二进制的代码，然后把可执行文件下载到目标机上运行。当内核编译成功后，通过串口或 USB 将其下载到开发板上运行。交叉编译调试环境建立在宿主机（即一台 PC 机）上，基于“宿主机—目标机”的交叉编译模式。图 1-10 是一种典型的交叉开发的方法。

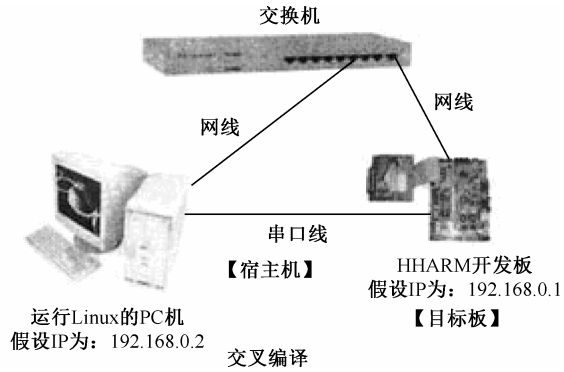

图 1-9 “宿主机/目标板”开发模式

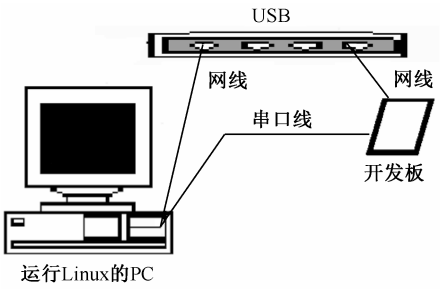

图 1-10 交叉开发

在软件设计上，如图 1-11 所示为结合 ARM 硬件环境及 ADS 软件开发环境所设计的嵌入式系统开发流程图。整个开发过程基本包括以下几个步骤。

（1）源代码编写：编写源 C/C++及汇编程序；

（2）程序编译：通过专用编译器编译程序；

（3）仿真调试：在 SDK 中仿真软件运行情况；

（4）下载：通过 JTAG、USB、UART 方式下载到目标板上；

（5）测试、调试：通过 JTAG 等方式联合调试程序；

（6）固化：程序无误，下载到产品上生产。

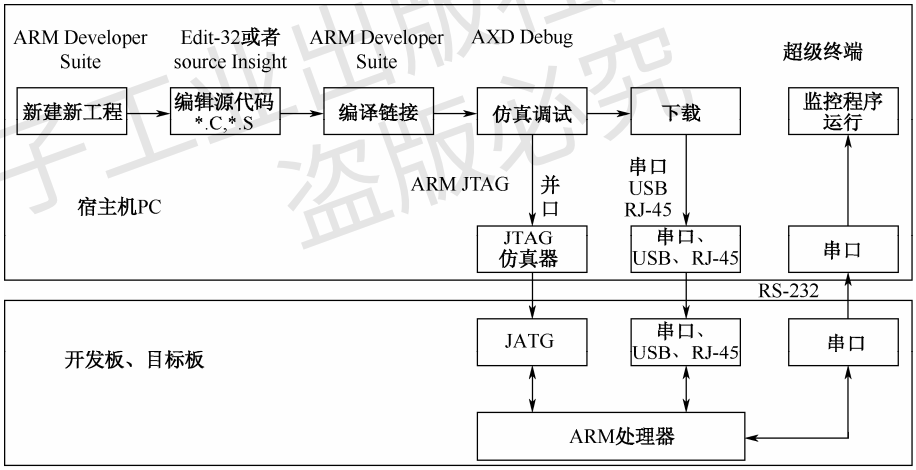

图 1- 11 嵌入式系统软件开发流程

二、嵌入式系统开发流程

嵌入式开发已经逐步规范化，在遵循一般工程开发流程的基础上，嵌入式开发有其自身的一些特点，如图 1-12 所示为嵌入式系统开发的一般流程。主要包括系统需求分析（要求有严格规范的技术要求）、体系结构设计、软硬件及机械系统设计、系统集成、系统测试，最终完成产品的开发。

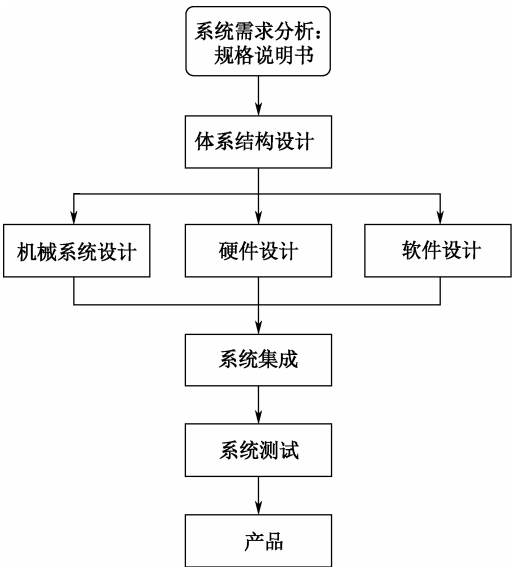

图 1-12　嵌入式开发流程图

① 需求分析。确定设计任务和设计目标，并提炼出设计规格说明书，作为正式设计指导和验收的标准。系统的需求一般分功能性需求和非功能性需求两方面。功能性需求是系统的基本功能，如输入输出信号、操作方式等；非功能需求包括系统性能、成本、功耗、体积、重量等因素。

② 结构设计。描述系统如何实现所述的功能和非功能需求，包括对硬件、软件和执行装置的功能划分，以及系统的软件、硬件选型等。一个好的体系结构是设计成功与否的关键。

③ 硬软件协同设计。基于体系结构，对系统的软件、硬件进行详细设计。为了缩短产品开发周期，设计往往是并行的。嵌入式系统设计的工作大部分都集中在软件设计上，采用面向对象技术、软件组件技术、模块化设计是现代软件工程经常采用的方法。

④ 系统集成。把系统的软件、硬件和执行装置集成在一起，进行调试，发现并改进单元设计过程中的错误。

⑤ 系统测试。对设计好的系统进行测试，看其是否满足规格说明书中给定的功能要求。

嵌入式系统开发模式最大特点是软件、硬件综合开发。这是因为嵌入式产品是软硬件的结合体，软件针对硬件开发、固化，不可修改。

如果在一个嵌入式系统中使用 Linux 技术开发，根据应用需求的不同有不同的配置开发方法，但是，一般情况下都需要经过如下的过程。

① 开发环境，操作系统一般使用 RedHat Linux，选择定制安装或全部安装，通过网络下载相应的 GCC 交叉编译器进行安装（比如，arm-1inux-gcc、arnl-uclibc-gcc），或者安装产品厂家提供的相关交叉编译器。

② 开发主机，配置 MINICOM，一般的参数为波特率 115200 Baud/s，数据位 8 位，停止位为 1，无奇偶校验，软件硬件流控设为无。在 Windows 下的超级终端的配置也是这样。MINICOM 软件的作用是作为调试嵌入式开发板的信息输出的监视器和键盘输入的工具。配置网络主要是配置 NFS 网络文件系统，需要关闭防火墙，简化嵌入式网络调试环境设置过程。

③ 引导装载程序 BootLoader，可从网络上下载一些公开源代码的 BootLoader，如 U-BOOT、BLOB、VIVI、LILO、ARM-BOOT、RED-BOOT 等，根据具体芯片进行移植修改。有些芯片没有内置引导装载程序，比如，三星的 ARV17、ARM9 系列芯片，这就需要编写开发板上 FLASH 的烧写程序，读者可以在网上下载相应的烧写程序，也有 Linux 下的公开源代码的 J-FLASH 程序。如果不能烧写自己的开发板，就需要根据自己的具体电路进行源代码修改。这是让系统可以正常运行的第一步。如果用户购买了厂家的仿真器，则比较容易烧写 FLASH，虽然无法了解其中的核心技术，但对于需要迅速开发自己的应用的人来说可以极大提高开发速度。

④ 已经移植好的 Linux 操作系统，如 MCLinux、ARM-Linux、PPC-Linux 等，如果有专门针对所使用的 CPU 移植好的 Linux 操作系统那是再好不过，下载后再添加特定硬件的驱动程序，然后进行调试修改，对于带 MMU 的 CPU 可以使用模块方式调试驱动，而对于 MCLinux 这样的系统只能编译内核进行调试。

⑤ 根文件系统，可以从网上下载使用 BUSYBOX 软件进行功能裁减，产生一个最基本的根文件系统，再根据自己的应用需要添加其他的程序。由于默认的启动脚本一般都不会符合应用的需要，所以就要修改根文件系统中的启动脚本，它的存放位置位于/etc 目录下，包括 /etc/init.d/rc.S、/etc/profile、/etc/.profile 等，自动挂装文件系统的配置文件/etc/fstab，具体情况会随系统不同而不同。根文件系统在嵌入式系统中一般设为只读，需要使用 mkcramfs、genromfs 等工具产生烧写映像文件。

⑥ 应用程序的 FLASH 磁盘分区，一般使用 JFFS2 或 YAFFS 文件系统，这需要在内核中提供这些文件系统的驱动，有的系统使用一个线性 FLASH（NOR 型，512KB～32MB），有的系统使用非线性 FLASH（NAND 型，8MB～512MB），有的两个同时使用，需要根据应用规划 FLASH 的分区方案。

⑦ 应用程序，可以放入根文件系统中，也可以放入 YAFFS、JFFS2 文件系统中，有的应用不使用根文件系统，直接将应用程序和内核设计在一起，这有点类似于 μC/OS-II 的方式。

⑧ 内核、根文件系统和应用程序，发布产品。

实验一　嵌入式实验平台的搭建

【实训目的】

（1）掌握嵌入式 Linux 开发的流程；
（2）熟悉嵌入式 Linux 的环境搭建。

【实训设备】

（1）装有 Linux 系统或装有 Linux 虚拟机的 PC 机一台；
（2）物联网多网技术综合教学开发设计平台一套；
（3）miniUSB 线一条；
（4）JTAG 线一条；
（5）串口线一条或 USB 线（A-B）。

【实训要求】

（1）能够熟练掌握嵌入式 Linux 的开发流程；
（2）熟练掌握嵌入式 Linux 开发的环境搭建。

【实训准备】

一、预备知识

绝大多数 Linux 软件开发都是以本地方式进行的，即本机开发、调试，本机运行的方式。这种方式通常不适合于嵌入式系统的软件开发，因为对于嵌入式系统的开发，没有足够的资源在本机（即板子上系统）运行开发工具和调试工具。通常的嵌入式系统的软件开发采用一种交叉编译调试的方式。交叉编译调试环境建立在宿主机（即一台 PC 机）上，对应的开发板叫作目标板。 运行 Linux 的 PC（宿主机）开发时使用宿主机上的交叉编译、汇编及链接工具形成可执行的二进制代码（这种可执行代码并不能在宿主机上执行，而只能在目标板上执行），然后把可执行文件下载到目标机上运行。调试的方法很多，可以使用串口，以太网口等，具体使用哪种调试方法可以根据目标机处理器提供的支持选择。宿主机和目标板的处理器一般不相同，比如宿主机为 Intel 处理器，而目标板为凌阳 Cortex-A8 实验仪，核心芯片为三星 S5PV210。GNU 编译器提供这样的功能，在编译器编译时可以选择开发所需的宿主机和目标机从而建立开发环境。所以在进行嵌入式开发前第一步的工作就是要配备一台装有指定操作系统的 PC 机作为宿主开发机，对于嵌入式 Linux，宿主机上的操作系统一般要求为 Ubuntu。嵌入式开发通常要求宿主机配置有网络，支持 NFS（为交叉开发时 Mount 所用）。然后要在宿主机上建立交叉编译调试的开发环境。

二、嵌入式 Linux 开发流程

嵌入式Linux开发，根据应用需求的不同有不同的配置开发方法。

三、对宿主 PC 机的性能要求

由于 Ubuntu 安装后占用空间约为 2.4GB～5GB 之间，还要安装 ARM-LINUX 开发软件，因此对开发计算机的硬盘空间要求较大。硬件要求：CPU：高于奔腾 500MB，推荐高于奔腾 1.0GB；内存：大于 128MB，推荐高于 256MB；硬盘：大于 10GB，推荐高于 40GB。

四、实验流程

系统搭建流程如图1-13所示，其中包括PC平台Linux虚拟机环境建立、QT 环境安装、ARM平台Linux系统搭建。

图 1-13　系统搭建流程图

【实训步骤】

1. Ubuntu 10.10 的安装

（1）安装 VMware 虚拟机软件

双击 VMware 图标，本书所使用的图标名称为 VMware-player-3.1.0-261024.exe，开始安装虚拟机软件。等待系统自动弹出如图 1-14 所示的窗口，显示安装 VMware Player 对话框，单击“ Next ”进入下一步。

图 1-14　安装虚拟机环境

系统弹出路径选择对话框，可选择任意磁盘路径，如图 1-15 所示；单击“ Next ”，进入下一步。

图 1-15　选择安装路径

系统弹出如图 1-16 所示对话框，选择默认选项即可，单击“ Next ”，进入下一步。

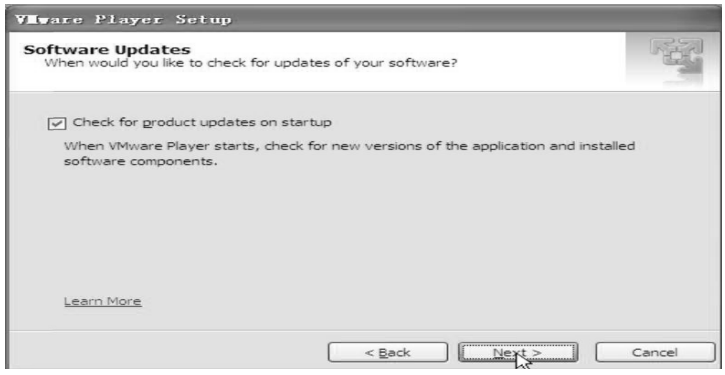

图 1-16　检测软件更新

系统弹出如图 1-17 所示的界面，选择默认选项即可，单击“ Next ”，进入下一步。

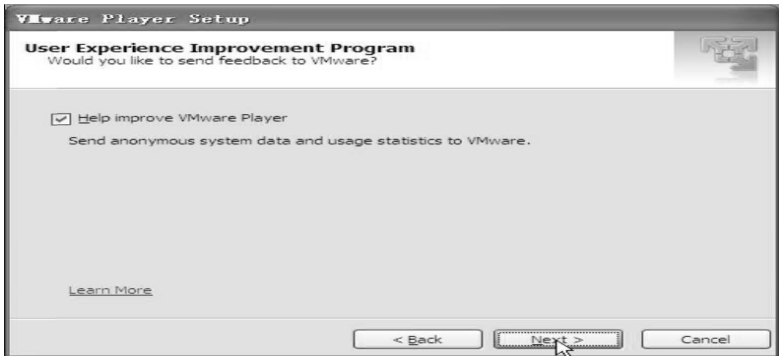

图 1-17　帮助改善 VMware Player

系统弹出如图 1-18 所示的界面，选择默认选项即可，单击“ Next ”，进入下一步。

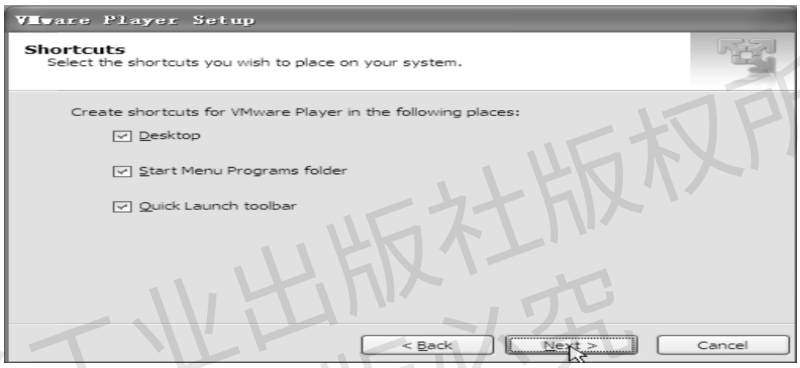

图 1-18　启动方式选择

系统弹出如图 1-19 所示的界面，单击“ Continue ”，进入下一步。

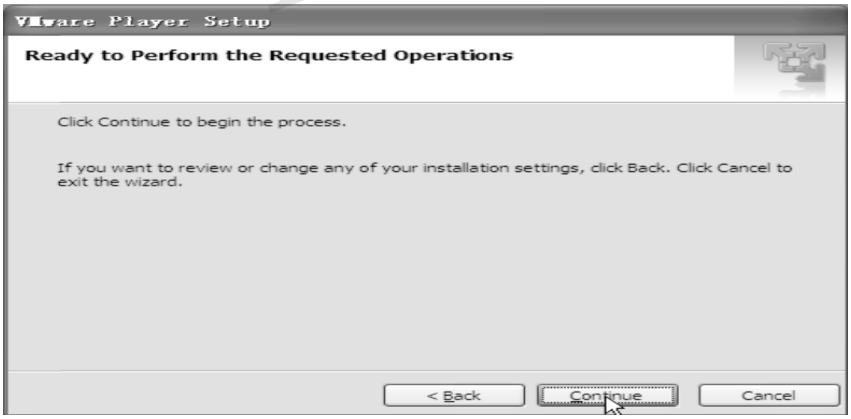

图 1-19　按照要求安装

接下来等待安装结束后，系统弹出安装完毕对话框，选择“ Restart Now ”；如图 1-20 所示。至此虚拟机安装完毕。

图 1-20　安装完毕

（2）在虚拟机中安装 Ubuntu10.10

打开虚拟机，双击桌面图标，出现如图 1-21 所示界面。

图 1-21　虚拟机开启界面

单击“ Open a Virtual Machine ”，弹出如图 1-22 所示对话框，选择已经配置过的 Ubuntu 系统，将其解压至 PC 相应磁盘中，选择.vmx 文件打开，返回到虚拟机主界面；单击“ Play virtual machine ”，即可打开 PC 机 Ubuntu 操作系统，进行程序开发，如图 1-23 所示。

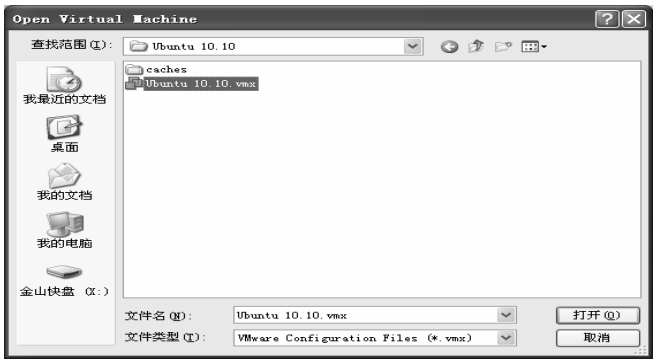

图 1-22　虚拟机系统路径

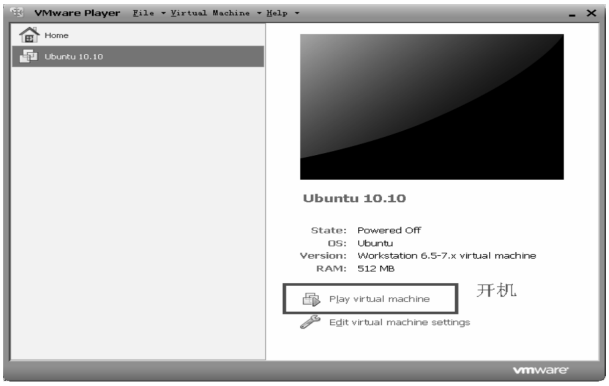

图 1-23　开机

等待片刻，开机后出现登录界面，单击选择“ UNSP ”用户，并输入密码“ 111111 ”，登录到系统，如图 1-24 所示。

如果认为默认的 Ubuntu 系统的显示界面不符合屏幕要求，可在“ 系统 ”→“ 首选项 ”→“ 显示器 ”菜单中更改系统的分辨率，如图 1-25 所示。

图 1-24　输入系统密码

图 1-25　更改系统分辨率

2. Ubuntu 系统和 Windows 系统之间相互复制文件

（1）从 Window 系统复制文件到 Ubuntu 系统

将文件或文件夹复制到 Ubuntu 虚拟机系统内的方法非常简单，直接将 Windows 系统上的文件拖拽到 Ubuntu 的桌面即可完成复制工作，类似于图 1-26 所示。

复制完成之后，可以看到在 Ubuntu 的桌面中出现拖拽过来的文件，如图 1-27 所示。

图 1-26　拖动文件到 Ubuntu 系统

图 1-27　文件被复制到 Ubuntu 内

（2）从 Ubuntu 系统复制文件到 Windows 系统

将文件从 Ubuntu 系统复制到 Windows 系统的方法类似，只需要从 Ubuntu 中拖动文件到 Windows 的文件夹内即可，如图 1-28 所示。

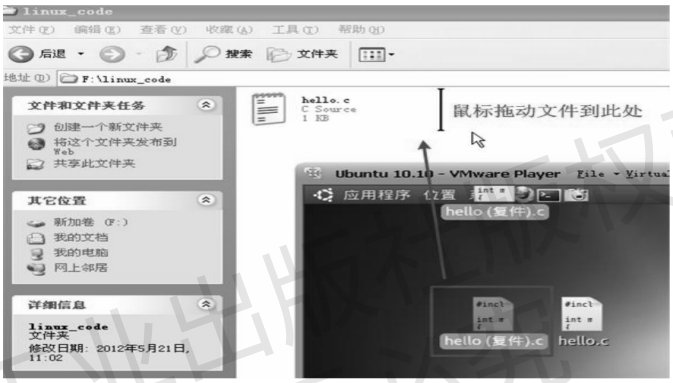

图 1-28　拖动文件到 Windows 系统中

复制完成后，在 Windows 的文件夹内即可看到拖拽过来的文件，如图 1-29 所示。

图 1-29　文件被复制到 Windows 系统中

3. *为实验箱的开发准备 PC 端的环境*

实验箱是一个完整的计算机系统，其内部运行了一个与 PC 上类似的 Linux 系统。在一般的开发过程中，我们需要首先在 PC 端做一些准备工作，这些设置包括：实验箱与 PC 的硬件连接、串口通信软件设置、网络环境设置。一般情况下，实验箱同时需要两种方式与 PC 建立连接：串口和以太网。首先使用标准 9 针串口线，将实验仪的 UART0 与 PC 的串口相连；然后，使用实验箱附带的网线，将实验箱的以太网接口与 PC 的网卡直接相连，或者将实验箱与路由器相连。这样就完成了硬件连接，如图 1-30 所示。

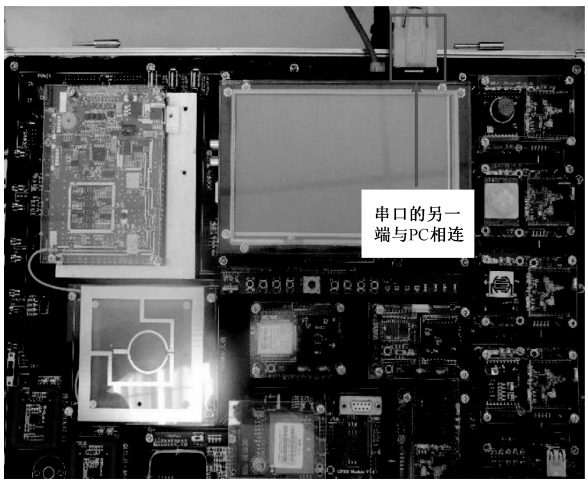

图 1-30　实验箱与 PC 的基本硬件连接

（1）串口通信软件设置

在 PC 端需要使用串口通信软件来对实验箱进行控制。通常情况下，使用 Windows 系统自带的“超级终端”工具即可（或者用户也可以使用其他同类型的软件，这里仅针对“超级终端”做详细设置说明）。首先在“开始”菜单中，找到“程序”→“附件”→“通讯”→“超级终端”，如图 1-31 所示。

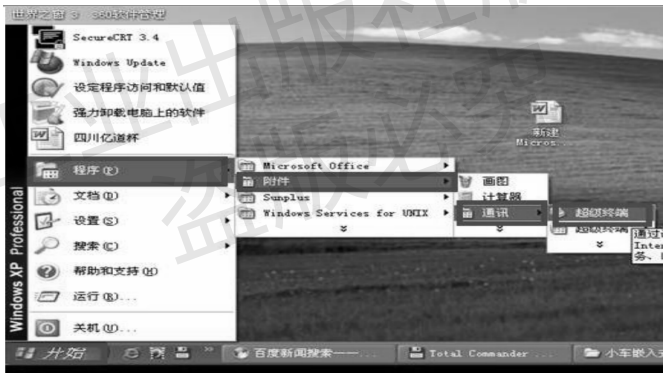

图 1-31　打开超级终端

设置超级终端名称，任意名称即可，如图 1-32 所示。

选择串口，例如，如果串口线接在串口 1 上就选择 COM1，如图 1-33 所示。

图 1-32　输入连接的名称

图 1-33　选择连接的串口

设置串口属性，每秒位数设置为 115200，数据流控制选择无，如图 1-34 所示。

图 1-34　选择串口的设置属性

此时，将物联网多网设计平台的电源打开，A8 实验仪的拨动开关拨至“ ON ”，并按下实验仪上的“ Power ”键，可以在超级终端中看到图 1-35 所示的启动提示信息。

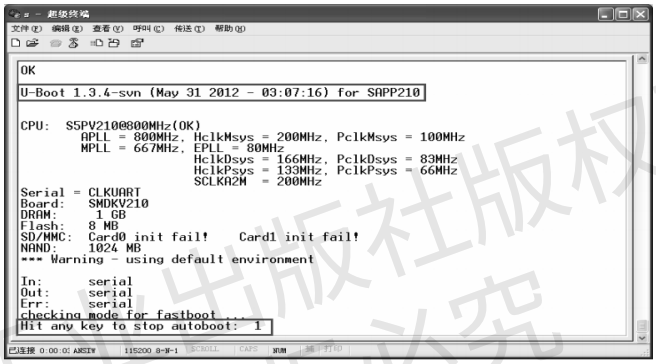

图 1-35　U-boot 启动界面

看到“ Hit any key to stop autoboot ”的提示，表示实验仪正在准备启动 Linux 系统。此时，如果不做任何操作，则在倒计时结束后将会启动 Linux。如果在倒计时的过程中按下键盘的空格键，即可进入到 U-Boot 的命令行，可以对系统启动参数进行调整，或者可以重新安装操作系统等。待系统正常启动之后，可以看到“ SAPP210。×××× login: ”的提示，如图 1-36 所示。其中，××××的内容根据不同的实验箱可能会有所不同。此时，表示 Linux 系统已经正常启动，等待用户登录。

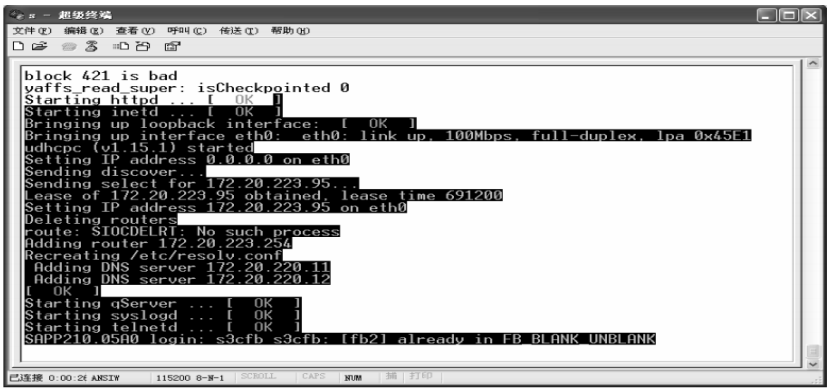

图 1-36　Linux 启动完成

按下“ Enter ”键开始登录，输入用户名“ root ”，密码“ 111111 ”，即可登录到系统，如图 1-37 所示。注意，密码输入时超级终端中不会有任何显示。登录成功之后，可以看到类似于“ [root@SAPP210 /root]# ”的提示。

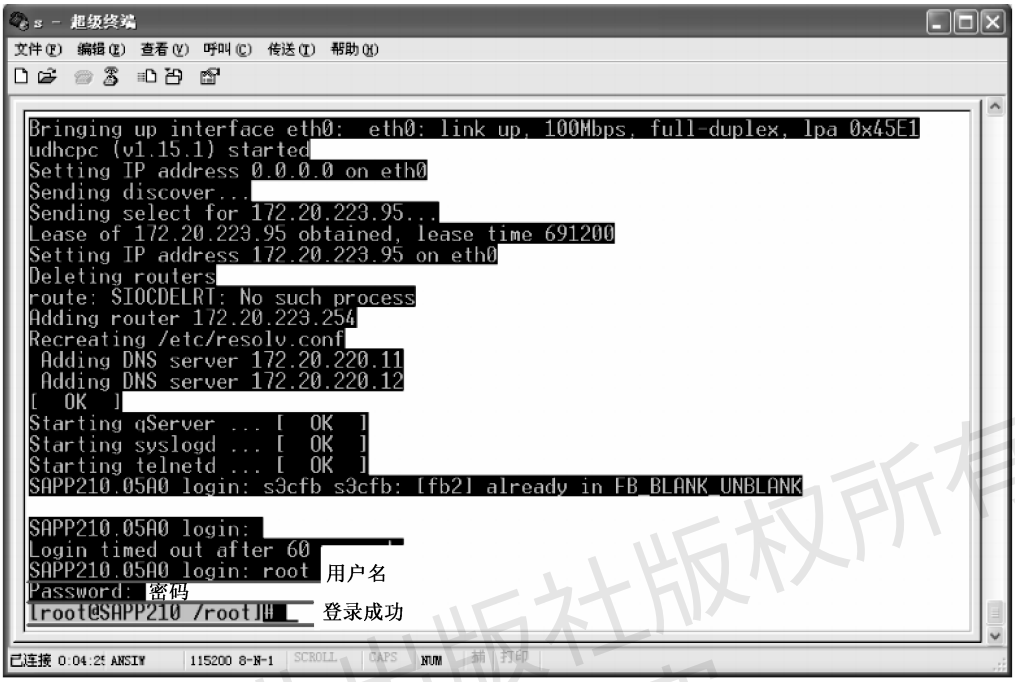

图 1-37　登录到实验箱的 Linux 系统

可以看到，实验箱的 Linux 系统启动过程中，会输出一些带有颜色的符号，导致超级终端软件的屏幕出现黑白相间的花屏。可以执行“ clear ”命令来清除屏幕，如图 1-38 所示。

网络环境设置：如果实验箱使用网线连入局域网，而局域网中存在 DHCP 服务器，则实验箱启动过程中，将会自动获取到 IP 地址，正如上面的图 1-36 中看到的这些提示一样。

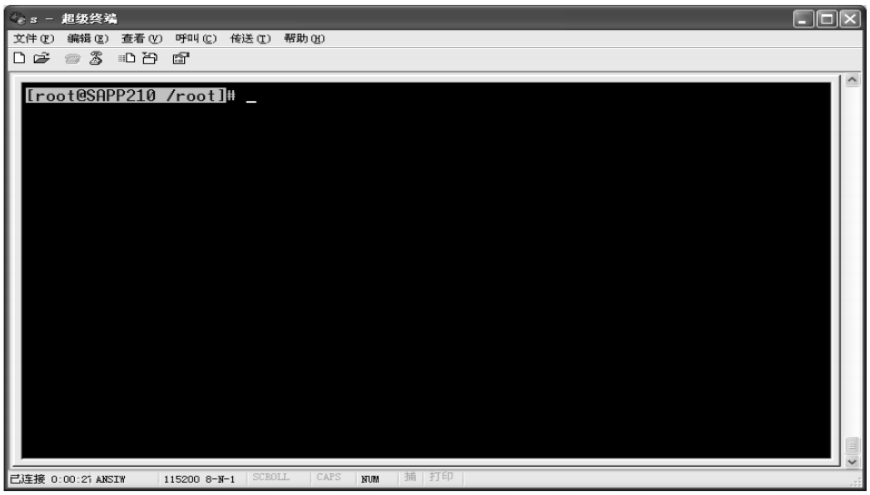

图 1-38　使用 clear 命令清屏

其中，“ 172.20.223.95 ”即为实验箱的 IP 地址。将实验箱连入局域网，用 DHCP 服务器

为其分配 IP 地址，是我们推荐的做法。然而，如果没有局域网的条件，或者局域网不具备 DHCP 服务器，则也可以通过手动配置的方式，来为实验仪分配 IP 地址。使用手动配置实验仪 IP 地址时必须设置计算机为静态 IP，方法如下：

在桌面上“网上邻居”图标上单击鼠标右键，选择“属性”，如图 1-39 所示。

在打开的窗口中，找到“本地连接”，单击鼠标右键，选择“属性”，如图 1-40 所示。

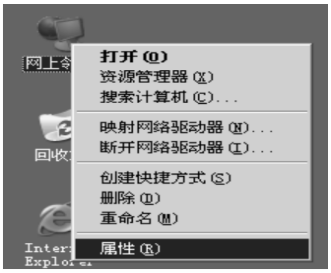

图 1-39　查看网上邻居的属性

图 1-40　查看本地连接的属性

在打开的窗口中，选择“Internet 协议（TCP/IP）”，并单击“属性”按钮，如图 1-41 所示。

在弹出的“Internet 协议（TCP/IP）属性”对话框中，按照如图 1-42 所示设置 IP，单击“确定”按钮，就设置好静态 IP 了。注意：在本例中，将 IP 地址设置为“192.168.87.1”，如用户对计算机网络熟悉，也可以按照自己的需要进行设置。

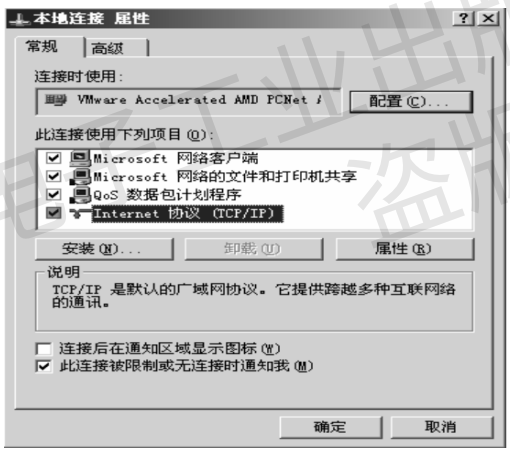

图 1-41　查看 Internet 协议属性

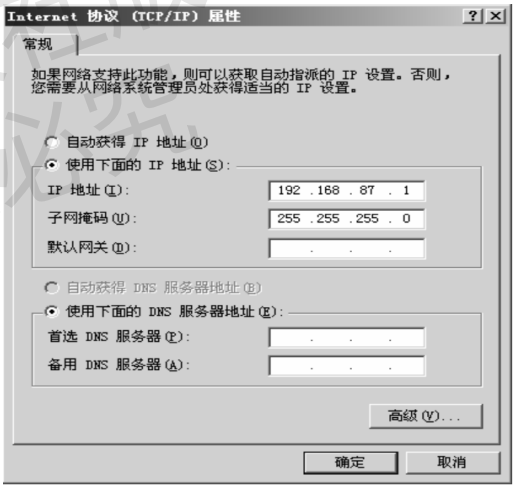

图 1-42　设置静态 IP

（2）配置实验仪 IP 地址

在超级终端中，执行命令“ipconfig eth0-i192.168.87.130 -m 255.255.255.0 -g 192.168.87.1”，即可为实验箱手动配置 IP 地址，如图 1-43 所示。

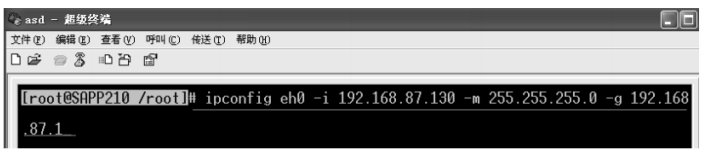

图 1-43　手动配置实验箱的 IP 地址

其中，-i 后面的参数是实验箱的 IP 地址；-m 后面的参数是子网掩码；-g 后面的参数是

网关地址。如果不需要网关，可以将-g 和其后面的参数省略。设置完成之后，需要执行“ service network restart ”命令重启网络服务，使设置生效，如图 1-44 所示。需要注意的是，实验箱的 IP 地址需要设置为与计算机同一个网段，例如，在本例中，计算机的 IP 地址为“ 192.168.87.1/255.255.255.0 ”，而实验箱的 IP 地址为“ 192.168.87.130/255.255.255.0 ”。

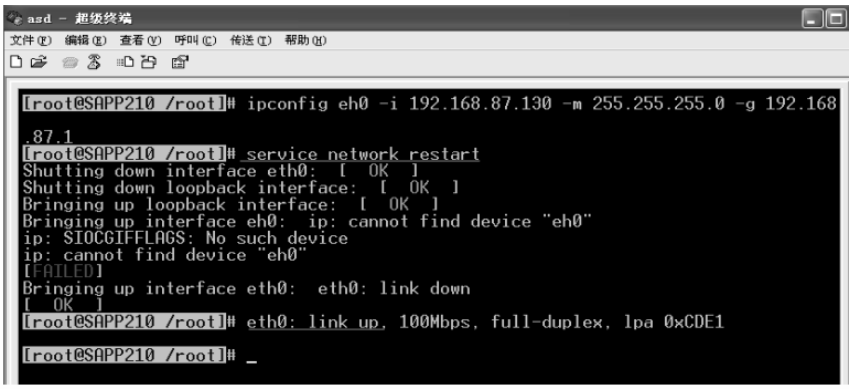

图 1-44　重启网络服务

当看到“ eth0: link up ”的提示，表示配置已经生效。

如需查看实验箱当前的 IP 地址，可以执行命令“ ifconfig eth0 ”，如图 1-45 所示。

图 1-45　查看当前的 IP 地址

注：如需将实验箱重新配置成自动获取 IP 地址，只需执行命令“ ipconfig eth0 -a ”，并重启网络服务即可。

（3）在 Ubuntu 下编译嵌入式 C 程序

此过程是嵌入式 Linux 开发非常重要的一个过程，后续绝大部分的实验都会重复使用本节的方法，以便在 Ubuntu 下编译可以在实验箱上运行的程序。用户无论在 Windows 下，或者在 Ubuntu 下编写完 C 程序之后，都必须首先保证该源程序文件存储到 Ubuntu 系统内。在本例中，我们首先在 Ubuntu 系统中的“ 主文件夹 ”中保存一个名为“ hello.c ”的文件。打开“ 主文件夹 ”的方法如图 1-46 所示。

在主文件夹下，单击右键选件“ 创建文档 ”→“ 空文件 ”，保存文档名为“ hello.c ”，如图 1-47 所示。

图 1-46　打开 Ubuntu 系统的主文件夹

图 1-47　在主文件夹下创建 hello.c 文档

在 hello.c 文档中输入以下内容，如图 1-48 所示。

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    printf("Hello, world!\n");
    return 0;
}
```

图 1-48　在 hello.c 文档中输入的内容

为了编译这个 C 程序，我们需要打开 Ubuntu 系统中的“终端”程序，该程序是一个命令行工具，可以运行标准的 Linux 命令，并可以用来编译程序“终端”，单击即可打开，如图 1-49 所示。或者可以直接单击 Ubuntu 系统中“系统”菜单右侧的“终端”快捷图标，如图 1-50 所示。

图 1-49　在“应用程序”中打开“终端”

“终端”程序打开之后，可以看到图 1-51 所示的界面。

图 1-50　使用快捷图标打开“终端”

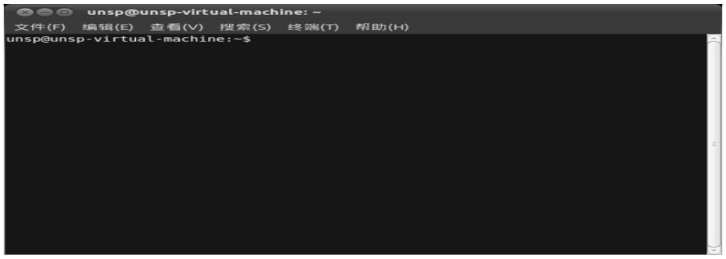

图 1-51　“终端”程序界面

终端程序打开之后，默认的工作目录即为“ 主文件夹 ”，在终端中输入“ ls ”命令，可以看到 hello.c 文件，如图 1-52 所示。

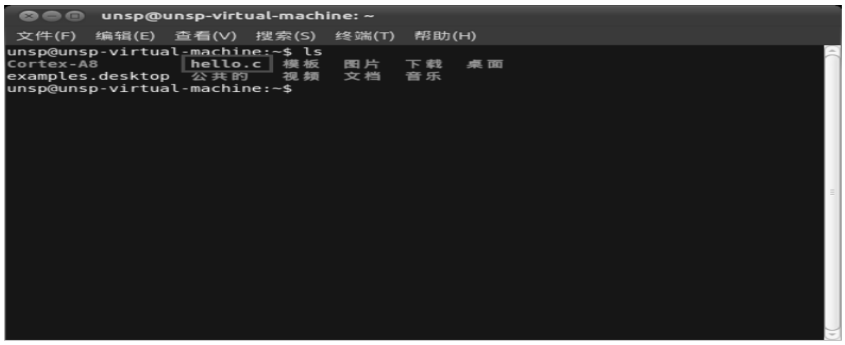

图 1-52　使用 ls 命令确认文件是否存在于当前目录

输入命令“ arm-Linux-gcc -o hello hello.c ”，即可将 hello.c 程序编译成为实验箱可以运行的可执行文件，如图 1-53 所示。其中，“ arm-Linux-gcc ”表示用于 ARM 系列芯片的编译器，“ -o hello ”表示编译之后生成的可执行文件的名字为“ hello ”，最后的“ hello.c ”即为待编译的 C 源程序文件。

图 1-53　编译 hello.c 程序

使用 ls 命令查看，可以看到目录下多了一个“ hello ”文件，该文件即为用于实验箱的可执行程序文件，如图 1-54 所示。

图 1-54　确认编译的文件是否生成

（4）将编译生成的文件复制到实验箱上并运行

首先将编译好的可执行程序文件从 Ubuntu 中复制到 Windows 系统，然后打开“ 我的电

脑”，在地址栏中输入“ ftp://192.168.87.130 ”并按回车键（上述 IP 为开发板的 IP 地址），如图 1-55 所示，其中，开发板的 IP 地址可以参考前面示意图的“ ifconfig eth0 ”命令来查看。

接下来，跟操作本地文件一样，我们可以使用复制/粘贴的方式将编译好的“ hello ”文件放入实验箱内，如图 1-56 所示。

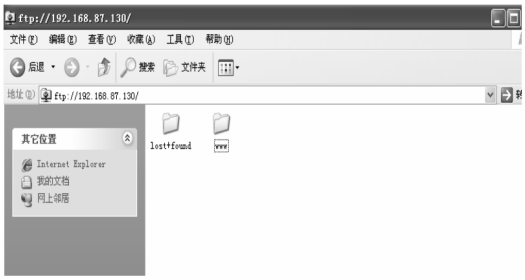

图 1-55　通过 FTP 访问实验箱的文件

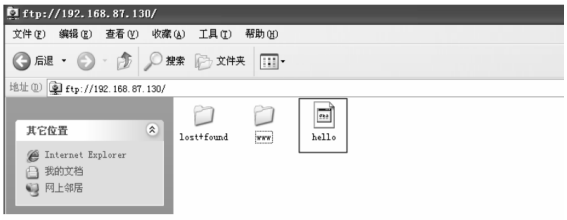

图 1-56　通过 FTP 将文件直接粘贴到实验箱内

在超级终端中，输入“ ls ”命令，可以看到“ hello ”文件已经被复制到了实验箱的系统内，如图 1-57 所示。

图 1-57　确认文件已经复制到实验箱

在超级终端中执行命令“ chmod +x hello ”，为“ hello ”文件增加可执行权限，如图 1-58 所示。

图 1-58　为文件增加可执行权限

最后执行“ ./hello ”命令，即可运行“ hello ”程序，如图 1-59 所示。

图 1-59　运行程序

思考与练习

一、填空

1．嵌入式系统一般由_____和_____两部分组成，硬件通常包含_____、_____和_____，软件通常由_____、_____和_____组成。

2．存储器用于存放_____和_____，嵌入式系统的存储器由_____、_____和_____组成。

3．嵌入式系统硬件的核心是嵌入式处理器，现在多数采用哈佛体系结构，指令系统一般为精简指令集（RISC），常用的嵌入式处理器分为_____、_____以及_____和_____。

4．嵌入式系统定义为：以_____为中心，以_____为基础，软、硬件可裁剪，适应应用系统对_____、_____、_____、_____、_____等严格要求的专用计算机系统。

5．一个嵌入式 Linux 系统从软件的角度看通常可以分为四个层次：_____、_____以及_____和_____。

二、思考与简单题

1．嵌入式系统开发流程主要包括哪些？

2．请列举一些常见的 BootLoader。

3．什么是操作系统？什么是嵌入式操作系统？

4．什么是存储器？嵌入式系统的存储器主要由哪几部分组成？

5．简述操作系统提供的服务。

6．嵌入式开发一般需要哪些流程？

7．请你列举几个嵌入式操作系统。