高等院校计算机精品教材系列

嵌入式系统实践教程

陈渝 主编
韩超 康烁 李明 编著

机械工业出版社
CHINA MACHINE PRESS

高等院校计算机精品教材系列

嵌入式系统实践教程

陈渝 主编
韩超 康烁 李明 编著

机械工业出版社

嵌入式系统是一个涉及多方面知识的交叉学科，目前 ARM 和 Linux 是嵌入式系统发展中的两个重点方向。本书是关于嵌入式系统的实践教程，主要关注基于 ARM 的 Linux 嵌入式系统开发，同时提供了大量由浅入深、易于扩展的实践环节。配套光盘提供了嵌入式系统的开发工具和源程序，大部分内容使用 SkyEye 仿真环境，避免了对具体开发板的依赖，通用性更强。本书条理清晰、重点突出、实践性强，既能满足在短时间内通过实践进入嵌入式系统领域的需要，又能满足深入学习拓展知识面的要求。

本书适合计算机等相关专业师生教学使用，也可供广大嵌入式系统开发人员学习、参考。

图书在版编目（CIP）数据

嵌入式系统实践教程 / 陈渝主编. —北京：机械工业出版社，2008.5
（高等院校计算机精品教材系列）
ISBN 978-7-111-24340-3

Ⅰ. 嵌… Ⅱ. 陈… Ⅲ. 微型计算机－系统开发－高等学校－教材
Ⅳ. TP360.21

中国版本图书馆 CIP 数据核字（2008）第 088096 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：唐德凯
责任印制：杨 曦

三河市国英印务有限公司印刷

2008 年 8 月第 1 版 • 第 1 次印刷
184mm×260mm • 18.5 印张 • 460 千字
0001－5000 册
标准书号：ISBN 978-7-111-24340-3
ISBN 978-7-89482-760-9（光盘）
定价：37.00 元（含 1DVD）

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739
封面无防伪标均为盗版

前　言

本书是《嵌入式系统原理及应用开发》的姊妹篇，是一本理论与实践结合，并偏重于实践环节的教材。在本书的教学理念中，注重将各个理论环节与实践相结合，避免空洞的理论讲解；而所有的实践环节则以理论思想为指导，注重技术的可扩展性，避免出现以偏概全的情况，实现对嵌入式系统感性认识和理性认识的统一。本书的主要内容以最为流行的嵌入式系统 ARM-Linux 下的开发为纲，重点实现了交叉编译工具、make 工具、GDB 调试、引导加载器、系统构建、驱动程序应用开发等几个方面的实践环节。书中示例力求精简，关注知识和技术要点，配套光盘的程序则具有很强的扩展性。

与以往的嵌入式系统的实践教程相比，本书主要具有以下几个特点：

第一、在具体技术的实践过程中，重视从全局把握，各个实践环节均以理论为指导；

第二、实践环节均具有可扩展性，重点突出了“移植”这个嵌入式开发的核心概念；

第三、很多示例均可以在 SkyEye 环境下进行仿真，这为没有具体开发板的读者提供了很大的方便，同时避免了学习单个开发板产生的局限性；

第四、在配套光盘中，提供了大量的程序和源代码资源，为读者搭建嵌入式系统的开发环境提供了极大的方便；

第五、在嵌入式系统的实践开发中，力求使用最精简、最标准的开发方法和步骤。本书选择的移植程序一般来自各个开源项目的官方发布版，读者通过对本书的学习，能将尽快融入嵌入式系统领域的主流技术之中。

本书由八章组成，具体的内容如下：

第 1 章，嵌入式系统教学与学习概述，知识结构，开发环境的组成；

第 2 章，嵌入式 Linux 开发环境构建，重点为 ARM GCC 交叉开发工具和 Makefile 的使用；

第 3 章，嵌入式系统的调试技术，重点为远程 GDB 的使用方法；

第 4 章，嵌入式系统的仿真环境，重点为 SkyEye 仿真环境的理念及使用；

第 5 章，嵌入式系统的 BootLoader，重点为 U-Boot 的开发；

第 6 章，嵌入式 Linux 操作系统构建，包括基于具有 MMU 的 ARM 处理器内核编译和文件系统构建，以及无 MMU 的 ARM 处理器的 μClinux 的内容；

第 7 章，Linux 驱动程序开发，包括简单字符设备、串口、帧缓冲、网络、内存盘五种驱动，每种驱动介绍驱动的结构原理、具体驱动以及驱动编程的切入点；

第 8 章，Linux 应用程序开发，包括 GUI 应用开发和网络应用开发两方面内容。

由于嵌入式的实践环节涉及的知识点和技术点很多，且各个实验的难度和复杂度相差很大，为了更好地保证条理性和层次性，本书采用了较为独特的章节组织方式。在每章的正文中，基本按照知识结构和技术层次介绍实践的原理和操作，在每章的后面包含了对本章的实验指导。在实验指导中，简单介绍了实验的步骤，具体实验的方法需要参考每章的正文。本书的每个实验都具有可扩展性，读者可以根据本书的配套光盘和自己的知识实践更多的内容。

在本书的光盘中，提供了开发工具和 SkyEye 仿真环境、各种相关开源软件的源代码、测试和示例代码。同时，本书也提供了一些二进制的工具和映像文件。读者使用本书进行实验的时候，需要使用光盘中的内容，同时对照书中的步骤和方法。对于某些实验，既可以完全从源代码进行，也可以使用部分二进制文件，前者的优点在于可以更深入地了解实验的原理和流程，后者的优点在于可以缩短实验环境的构建过程。

为了更好地学习本书内容，我们希望读者熟悉 C 语言，有较强的源代码阅读能力和分析能力；熟悉 ARM 体系结构；具有软件开发的基础知识；能熟练使用 Linux 系统，有解决安装配置问题的能力。本书实验的大部分内容在 Linux 主机环境下完成，推荐使用 OpenSuse、Fedora、Ubuntu 等 Linux 桌面发行版，主机需要具有本机 gcc 工具。

读者在学习的过程中，应该注意区分嵌入式系统运行的程序、PC 主机运行的程序、仿真环境下运行的程序三者的区别和联系。

本书的主要编写工作由陈渝、韩超、康烁和李明完成，全书由韩超、陈渝规划和统稿。本书的配套光盘及其实验内容由康烁规划，康烁、韩超完成调试和制作。此外，参与本书编写工作的人员还有谢劲松、王月龙、刘永安、吴德新、冯学青、邓祺、李鹏飞、魏治宇等。

感谢北京亚嵌教育研究中心的技术专家和学员，由于他们高效的辅助工作，才使得本书能够及时与读者见面。

编　者

目　录

第1章　嵌入式系统教学与学习概述

本章主要从嵌入式系统的知识结构、相关专业和开发方式等方面，来介绍嵌入式系统的教学与学习的方法。

通过本章学习，读者应该掌握以下内容：

- ✧ 嵌入式系统的知识结构
- ✧ 嵌入式系统的学习方法
- ✧ 如何构建嵌入式系统

1.1　嵌入式系统的教学和学习

本节介绍嵌入式系统教学和学习的特点、嵌入式系统的知识结构以及嵌入式系统的开发流程。

1.1.1　嵌入式系统教学和学习的特点

嵌入式系统是一个交叉学科，其核心的知识主要来自计算机学科和电子学科。此外，嵌入式系统的学习还涉及了与具体应用密切相关的通信、控制等学科。

在嵌入式系统的学习中，存在着知识点众多、内容庞杂的问题。因此，理论基础和工程实践相结合的方式在嵌入式系统的学习中是至关重要的。嵌入式系统涉及的知识和技术众多，在理论基础学习和教学中，需要保持清晰的脉络，保持各个部分的知识相对独立。在嵌入式系统的工程实践中，则需要采用循序渐进的方式，保证学习者在学习的过程中，逐步掌握技术的应用。

因此理论教学和工程实践在嵌入式系统的教学和学习中应相辅相成。如果偏重于理论教学的方式，需要保持清晰的知识脉络和模块化的特点。这时产生的问题是：学习者在学习每个部分的知识的时候，将由于缺乏其他知识和感性认识，难以理解相关的内容。如果偏重于工程实践，则需要采用逐步引导的方式，按照工程的思路将技术细节逐步教授。这种方式产生的问题是：学习的过程交互性在教程的编写上有一定的困难，容易产生知识点凌乱和以偏概全的问题；对于嵌入式系统的初学者，在经验上和工程人员存在差距，有时也不易理解这种工程思路。因此，嵌入式系统的教学必须同时兼顾知识点的条理性和技术的易学性。

嵌入式教学的根本目的，是让学习者尽快地学习到更多、更完整的知识体系结构，并掌握嵌入式工程中分析和工作的方法、增长工程技术经验。理论知识的教学和工程实践的教学需要相互结合。一般来说，理论知识的教学应保持清晰的结构和知识点，而工程实践的教学则包含了针对单独或若干知识点的实验环节，以及实际的工程实践。

在嵌入式系统的学习中，需要根据自身掌握的知识和技术情况，选择适合自己的学习方法。在理论知识的学习中，应该从宏观的角度掌握完整的知识体系，在微观的角度根据需要，

有选择地学习相关知识、技术的重点。在工程实践方面，需要将工程实践中的技术和理论知识点结合，同时增强自己的基本功。

结合理论学习和工程实践的嵌入式系统学习方法如图 1-1 所示。

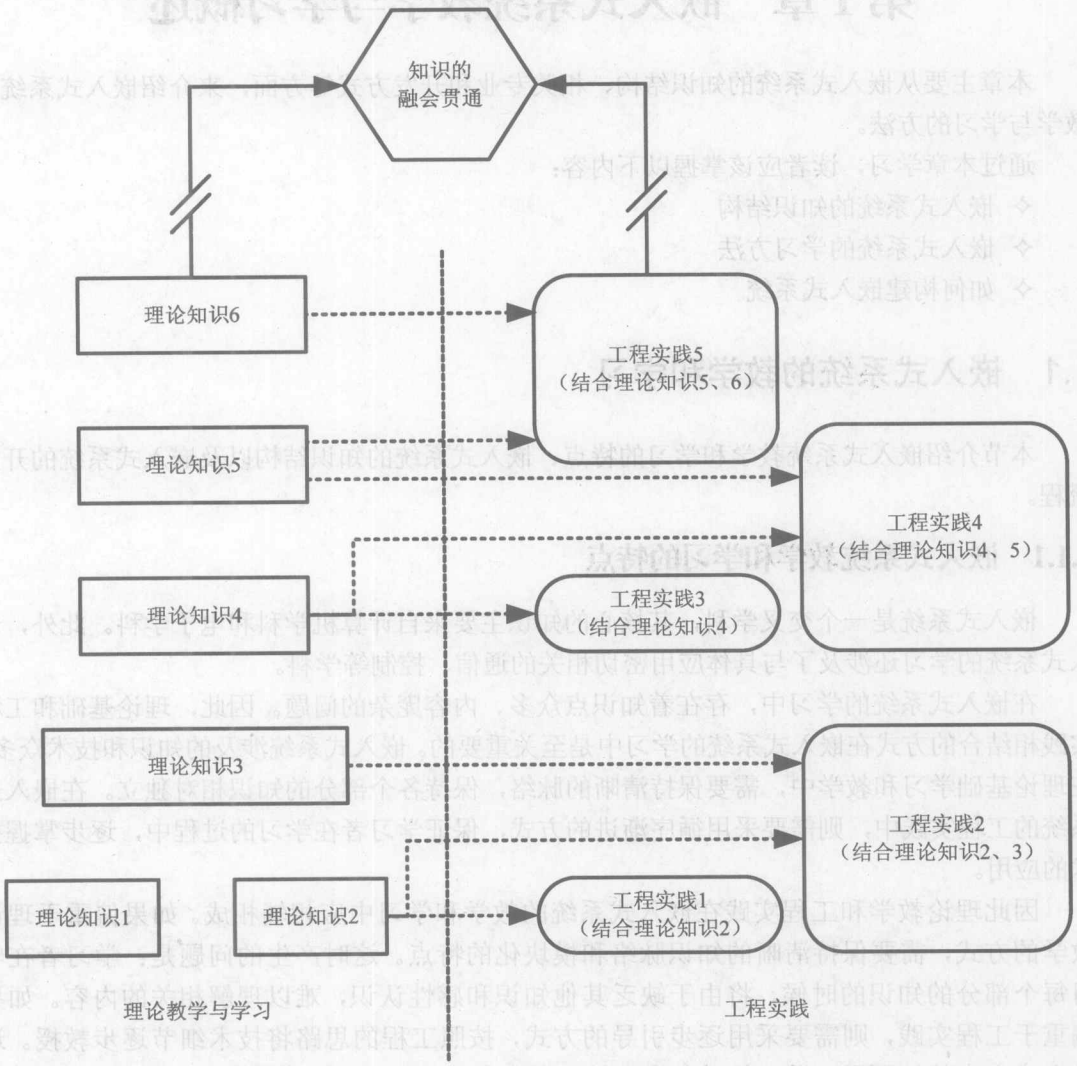

图 1-1　嵌入式系统的理论学习和工程实践

可以将嵌入式系统的教学与学习分成理论学习和工程实践两个部分。

在理论学习方面，知识结构层次清晰，知识体系具有模块化的特点，各部分相对独立，又可形成体系。

在工程实践方面，每个工程实践会对应几方面的理论知识，随着理论知识的增长，工程实践将越来越复杂。

理论知识给了工程实践以指导，工程实践给理论知识的学习提供感性上的认识。最终力争实现理论知识和实践技术融会贯通的目的。

注意

在嵌入式系统的学习中，理论和实践是一个交替上升的过程，最终应达到融会贯通的目的。

1.1.2　嵌入式系统的知识结构

嵌入式系统是一个交叉学科，其核心部分涉及到微电子、电子、计算机等几个学科。嵌入式系统本身也是一个计算机系统，具有计算机系统一般的特点。同时，嵌入式系统没有通用计算机系统标准化的特点。可以从一个嵌入式系统构建的角度划分嵌入式系统的知识结构。

1．嵌入式系统的核心知识

嵌入式系统一般是一个以处理器为核心，增加了一定外围硬件作为物理基础的系统，较为复杂的嵌入式系统也具有嵌入式的操作系统，同时可能运行较为复杂的应用程序。从系统划分的角度，嵌入式系统可以分为处理器、外围硬件、操作系统、应用程序、嵌入式开发环境等部分。与之类似，嵌入式系统的知识结构同样可以按照这种方式划分。

嵌入式系统的知识结构在宏观上可以划分成以下几个部分，如图 1-2 所示。

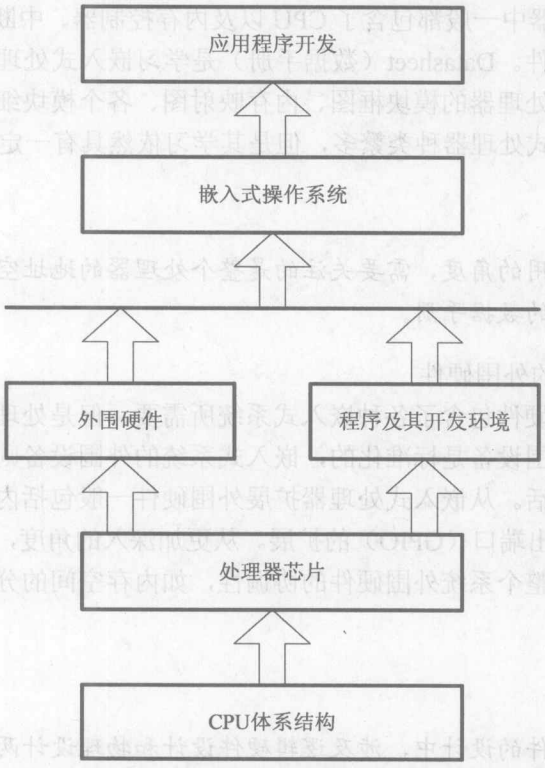

图 1-2　嵌入式系统的知识结构

（1）CPU 的体系结构

本部分的知识涉及到嵌入式系统的核心，在通用计算机领域，一般使用 x86 体系结构的

CPU，从 Intel 的 386 到 Pentium 一代至四代，乃至 AMD 的 K5、K6、毒龙、速龙都具有很强的兼容性，甚至 AMD 的 64 位 x86 也可以兼容 32 位的指令集。在嵌入式领域中，由于功耗的原因，x86 体系处理器使用不是很多，常用的嵌入式处理器为 ARM、MIPS 等体系，此为还有众多的 8 位、16 位单片机和各种 DSP（数字信号处理器）。在 CPU 体系结构知识方面，一般的嵌入式开发者需要关注某体系结构处理器的指令系统和内部架构，也可能涉及到 CPU 的设计，如基于 FPGA 的软核设计。

提示

在 CPU 体系架构中，涉及 CPU 内核的设计实现以及 CPU 的使用两方面的内容。CPU 的设计和实现需要涉及微电子技术，在一般的嵌入式系统开发中，只需要关心某体系结构 CPU 的使用。

（2）嵌入式处理器芯片

在通用计算机领域，CPU 一般需要与芯片组（如南北桥的架构）结合才能形成完整系统，CPU 提供控制和逻辑运算功能，芯片组提供内存管理和 I/O 功能。在嵌入式领域，嵌入式处理器的特点是具有较高的集成度，包括各种 32 位嵌入式 MPU（微处理单元）、DSP、8 位和 16 位的单片机。各类嵌入式处理器都具有一个特点，即都包含了 CPU 核心和众多集成在芯片内的部件。嵌入式处理器中一般都包含了 CPU 以及内存控制器、中断控制器、定时器、通用输入输出引脚等片内部件。Datasheet（数据手册）是学习嵌入式处理器的基础，其相关的重点内容包括封装形式、处理器的模块框图、内存映射图、各个模块细节、电气和机械特性、引脚信号等。虽然嵌入式处理器种类繁多，但是其学习依然具有一定的规律性。

提示

从嵌入式处理器使用的角度，需要关注的是整个处理器的地址空间和片内各个模块的使用，这些均来自处理器的数据手册。

（3）嵌入式系统的外围硬件

嵌入式系统的外围硬件包含了各种嵌入式系统所需要、但是处理器内部没有集成的模块和功能。通用计算机外围设备是标准化的，嵌入式系统的外围设备则不具有这一特点，其设计方式也更加复杂和灵活。从嵌入式处理器扩展外围硬件一般包括内部部件的扩展、总线部件的扩展、通用输入输出端口（GPIO）的扩展。从更加深入的角度，嵌入式系统外围硬件扩展的时候，还需要考虑整个系统外围硬件的协调性，如内存空间的分配、中断资源的分配、各部件复用的资源等。

提示

嵌入式系统外围硬件的设计中，涉及逻辑硬件设计和物理设计两方面的技术，在逻辑硬件设计中，主要关注外围部件和电路的功能和逻辑；在物理设计中，将涉及 PCB 布线等方面的技术。这两方面的技术是可以分开的，嵌入式系统设计的初期，首先需要考虑系统的逻辑功能，硬件 PCB 可以使用不同的方式实现。

（4）嵌入式系统的程序开发基础及环境

在嵌入式系统中，程序开发基础及环境是嵌入式系统软件开发的基础。目前，嵌入式系统主要使用 C 语言开发，在某些与硬件密切相关的地方也使用相应的嵌入式处理器汇编语言。嵌入式的程序开发基础包含了具体体系结构的汇编语言的设计原则、C 语言开发的要点、面向某种体系结构的算法、与具体处理器相关的启动代码。嵌入式程序开发环境包含了代码生成工具（编译器、汇编器、链接器）、调试工具、仿真工具等。

提示

嵌入式系统程序开发基础涉及了 C 语言程序可移植性的问题。虽然 C 语言可以屏蔽很多硬件细节，但是在不同体系的 CPU 中依然存在着大小端、内存对齐以及程序优化等问题。

（5）嵌入式操作系统

嵌入式操作系统与通用计算机常用的 Windows 或者 Linux 操作系统不同。一般嵌入式操作系统只提供了系统运行的基础环境，一般嵌入式操作系统的核心包含了任务调度、任务间通信、内存管理等模块。学习嵌入式操作系统需要了解的相关知识包含了嵌入式系统的基本概念、模块划分、源程序、扩展部分（如文件系统、网络协议、图形用户界面等）、驱动程序、应用程序开发等。目前常见的通用嵌入式操作系统包括μC/OS- Ⅱ、嵌入式 Linux、Windows CE 等。

提示

对于嵌入式操作系统来说，完成操作系统的移植是嵌入式系统设计最需要关注的问题。

（6）嵌入式系统的应用程序

嵌入式系统的应用程序属于软件系统的上层，从这个角度看，嵌入式系统的应用程序设计与通用计算机的程序设计具有很大的类似性。嵌入式系统应用程序设计也具有自身的特点：首先，嵌入式系统的软件模块的层次比通用计算机简单，嵌入式系统程序设计具有一定的底层性的特点；其次，由于嵌入式系统的资源有限（处理器速度、内存容量、内存带宽），嵌入式系统的程序设计更需要注重效率。

提示

在上层应用程序的开发中，需要关注的涉及嵌入式系统的知识和技术较少，只需要注重应用程序在嵌入式系统中的优化。

在嵌入式系统的知识体系学习中可以注意到，它们各个部分存在着联系，形成一个有机的整体，同时各部分又具有着一定的独立性。由此可见，在从事某一个部分的开发工作的时候，并不需要掌握所有嵌入式系统各方面的知识。

各部分之间的知识联系主要体现在 CPU 是整个系统的核心；处理器由 CPU 和片内外设组成；嵌入式系统外围硬件扩展与具体处理器相关；在程序基础方面，与处理器的指令系统

相关，启动代码则与具体的处理器相关；而操作系统则需要建立在以上四点的基础上，涉及具体 CPU 的指令体系、处理器的必需外设、外部扩展的内存等部件、编译系统等；应用程序的设计又是建立在操作系统之上的。

各部分知识的独立性则主要体现在以下几点：

1）从外围硬件扩展的角度来说，对于一般的外围硬件，只需要熟悉具体芯片的 I/O 即可，不需要熟悉处理器中 CPU 核心使用何种指令系统。

2）从程序开发基础及环境的角度来说，编译、汇编、链接的过程对于各种系统都是类似的，在这个层次也不需要涉及 CPU 体系结构方面的知识。

3）从嵌入式操作系统的角度来说，一般都具有移植层，移植层可以屏蔽一些底层的细节，纯粹嵌入式操作系统的知识可以不涉及底层。

4）某些嵌入式的应用程序可以是和操作系统、硬件都没有关系的，能够移植到各种系统上。

2. 嵌入式系统的相关知识

以上介绍了有关嵌入式系统各方面的知识，主要为电子学科和计算机学科的交叉内容，它们共同构成了嵌入式系统知识的核心部分。

从知识的体系结构上，嵌入式系统的知识体系的发展来自上下两个层次。在发展过程中，嵌入式系统的技术主要来源于基于单片机的电子设计和基于通用计算机的设计。二者在发展中也产生了融合：一方面，基于通用计算机的嵌入式系统设计向下层的具有针对性、专业化的方向发展；另一方面，基于单片机的电子设计向上层的复杂化应用的方向发展。也就是说，电子和计算机两大学科在各自的发展中交融成嵌入式系统这一交叉学科。

高性能嵌入式处理器的出现，让二者完成统一。新型的嵌入式系统既具有通用计算机系统人机交互、网络、多媒体等功能，也具有单片机系统低功耗、高集成度、针对对象设计的特点。嵌入式系统相对核心的知识包括：不同体系 CPU 的指令集、嵌入式处理器的使用、外围逻辑电路设计、嵌入式开发调试系统、操作系统和应用程序移植等。

对于整个嵌入式系统，还涉及到电子学科和计算机学科其他的内容，以及其他学科的知识。例如：

1）电子学科中各类电子模块的知识、PCB 相关技术。

2）计算机学科中软件工程的思想在嵌入式程序开发中的使用。

3）微电子学科中的芯片制造技术。

4）编解码等算法与具体硬件的结合。

5）与应用结合紧密的控制和通信领域的各种技术也是嵌入式系统相关技术的一部分。

1.1.3 嵌入式系统的开发流程

在嵌入式系统的开发中，对应嵌入式系统整体的产业链的各个环节，分成若干个阶段。例如：系统构建阶段、嵌入式开发平台阶段、解决方案阶段、产品化阶段等。嵌入式系统产品的开发过程如图 1-3 所示。

图 1-3　嵌入式系统产品的开发过程

嵌入式系统开发包括从处理器、外围硬件、开发环境构建、操作系统到嵌入式的应用程序等各方面的内容，这是一个完整的开发流程。然而，嵌入式系统开发中一般不使用这种全面的开发方式，而是复用一些成熟的系统和技术。嵌入式工程的开发可以是嵌入式产品形成过程中的一个或者几个环节。

基于平台的开发可以大大加快嵌入式产品的开发速度。在嵌入式系统的开发中，开发形式有很多种，例如：基于现有系统的改造，根据现有的系统作出硬件及软件方面的更改，形成新的系统；构建自己的解决方案；基于解决方案做出商业化的产品。

提示

嵌入式工程师的技术不一定需要构建整个的嵌入式系统，很可能只需要完成嵌入式系统整个产业链中的一部分环节和工作。

1.2　系统的学习与使用

嵌入式系统的学习和开发需要在嵌入式系统的开发环境中进行。如上所述，嵌入式系统的开发中包含很多环节的内容。嵌入式系统的构建是一个系统化的工程。在嵌入式硬件开发中，由于没有通用计算机系统标准化的特性，需要嵌入式工程师具有更加全面的知识和相对丰富的经验。随着嵌入式处理器功能的丰富，可以让系统拥有更强大的功能，甚至接近 PC 系统的功能，例如 PDA、智能手机等。嵌入式操作系统和应用软件的开发在嵌入式系统中的地位越来越重要。

由于系统性能和开发工具方面的因素，程序的调试在嵌入式系统中的复杂程度远远大于通用计算机中的程序调试。同时，由于嵌入式系统的多样性，调试的手段也更加多种多样。调试手段在嵌入式系统中的作用日益重要，例如：基于主机环境的嵌入式程序开发、基于开发板的嵌入式系统开发、基于主机仿真环境的嵌入式系统开发。

1.2.1　基于主机环境下的嵌入式开发

由于开发环境的成熟和性能的原因，在主机上开发程序比在嵌入式系统上开发更加便利，调试的手段也更加成熟。因此，很多程序都是在 PC 中调试完成后，再移植到具体嵌入式系统中的。

在以下几种情况下，适合在主机环境开发嵌入式系统的程序：第一，很多程序是与硬件无关的，例如数据库程序等；第二，某些较高层的软件虽然与硬件相关，但是软件层级结构提供了统一的接口，例如网络编程、图形用户界面编程甚至操作系统的移植等；第三，硬件抽象层可以隔离程序和硬件，因此，即使需要硬件也可以在 PC 系统上调试完成后，再到嵌入式系统中调试。

基于主机的嵌入式系统程序开发如图 1-4 所示。

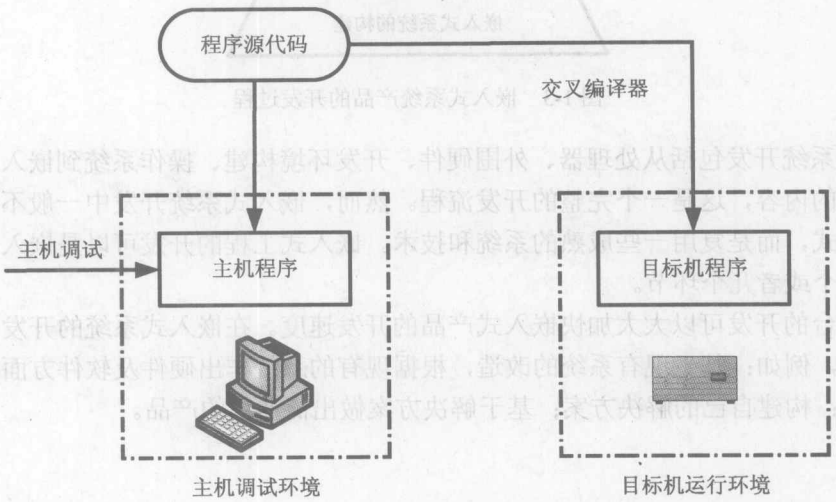

图 1-4　基于主机环境的嵌入式程序开发

在主机（PC）环境的程序调试，需要将程序的源代码使用主机的编译工具（x86 编译器、汇编器、链接器）生成主机的可执行程序，调试完成后，再将系统使用目标机的编译工具生成目标机的可执行程序，在目标机（嵌入式系统）中运行。因此，可以在主机环境中，方便地使用各种调试工具，尤其在程序中设置断点（breakpoint）、单步执行（step）等。PC 程序的开发可以使用 Windows 的 Visual C++、Linux 下 GDB 的等调试环境，同时 PC 具有的较高性能也更适合大规模的程序开发及调试。

在主机环境中开发程序需要注意程序在相关嵌入式系统的可移植性和性能问题。

在移植性方面，以 C 语言为例，虽然它可以屏蔽大部分与 CPU 体系结构相关的内容，但是依然需要注意对齐和大小端（字节序）等与 CPU 体系结构相关的问题。此外，需要注意某些嵌入式系统的 C 语言编译器功能的限制。

性能问题，是嵌入式系统需要注意的核心问题。嵌入式程序对性能具有更高的要求。C 语言固然有较好的移植性，但是在某些与性能密切相关的地方，依然需要使用相应 CPU 的汇编语言实现。在程序的性能比较方面，不同程序在主机（PC）运行效率的高低，并不一定和它们在目标机（嵌入式系统）上运行效率的高低相一致。例如：目前的 PC 一般都支持浮点运算，因此在小数运算方面，使用 C 语言中的浮点库性能较高，使用定点（整数）的算法性能较低；而某些嵌入式处理器没有浮点运算单元，这样浮点算法将完全依靠编译器的转换，性能将很低，这时避免浮点运算就是必要的。