

想深入学习嵌入式和手机开发吗，看这本书吧！

嵌入式设备驱动 开发精解

孙智博 著

- 涵盖 ARM CPU 及各种常用外部设备驱动开发的方方面面，包括各种硬件接口、硬件接口协议说明及各种外设的使用和调试方法。
- 以讲解 ARM 系统知识为主线，结合一些实际项目中的例子穿插学习，让读者真正入门和学通嵌入式开发技术。

嵌入式设备驱动

开发精英

• 孙智博 著



人民邮电出版社
北京

图书在版编目 (C I P) 数据

嵌入式设备驱动开发精解 / 孙智博著. -- 北京 :
人民邮电出版社, 2013.4
ISBN 978-7-115-30710-1

I. ①嵌… II. ①孙… III. ①微型计算机—系统设计
IV. ①TP360.21

中国版本图书馆CIP数据核字(2013)第019135号

内 容 提 要

本书的内容主要涵盖 ARM CPU 以及各种常用外部设备驱动开发的方方面面, 包括各种硬件接口, 硬件接口协议说明以及各种外设的使用及调试方法, 特别是对于开发调试过程中可能遇到的各种问题以及如何解决这些问题进行了讨论, 同时还包括软件方面的调试方法与硬件方面的调试方法等。

嵌入式设备驱动开发精解

- ◆ 著 孙智博
责任编辑 张 涛
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 11.5
字数: 280 千字 2013 年 4 月第 1 版
印数: 1—3 500 册 2013 年 4 月北京第 1 次印刷

ISBN 978-7-115-30710-1

定价: 35.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

为什么写本书

本书的编写主要是针对从事嵌入式软件开发人员。本书的内容主要涵盖 ARM CPU 以及各种常用外部设备驱动开发的方方面面，包括各种硬件接口、硬件接口协议说明以及各种外设的使用及调试方法，特别是对于开发调试过程中可能遇到的各种问题以及如何解决这些问题进行了讨论，同时包括但不限于软件方面的调试方法与硬件方面的调试方法等。

从事嵌入式驱动开发的工作主要就是与各种外设、外设接口以及 OS(操作系统)打交道，这是一个会涉及非常多的知识点的、系统级的软件开发与设计过程。

狭义上的嵌入式驱动开发的定义：基于某一确定架构和确定型号的 CPU，为各种外设开发能够正确稳定运行的驱动程序，这是狭义上的一个定义，实际上作为一个驱动开发人员，不但要非常熟悉所使用的 CPU，同时对操作系统也要非常熟悉，这里指的 OS 并不是某一个具体的 OS，而是泛指各种 OS，而我们要做的就是掌握 OS 本身的几大功能特性。而市面上现有的 OS 也都是遵循这几大功能特性设计的，所不同的就是，这些特性的实现方式以及使用方法，所以不可拘泥于某一个 OS。另外，还需要掌握常用的外设及这些外设常用的接口方式和这些接口的具体协议规范。对于硬件接口协议来说，大家看一遍书能大概知道是个什么意思，而要真正去调试一个设备时，那又完全是另外一回事了，就像我们学习完 C 语言，就像让你开发一个功能完备的软件一样，往往无从下手。

现在市面上的嵌入式开发的书主要都是讲 ARM 处理器的，很少涉及嵌入式开发中常用的外设器件，关于这些常用外设器件的使用及调试方法更是非常少。

本书的主要特色

本书内容主要集中在 ARM 系统知识、各种外设的使用及调试上，结合一些实际项目中的例子穿插学习。只有真正让这些外设运行起来，才能说明你对 ARM CPU 已经非常熟悉了，如果仅仅是学习 ARM 相关知识，没有真正地用 ARM 让这些外设工作起来，其实并没有真正地学习到 ARM 的知识。现在嵌入式设备中所使用的 CPU 基本上是 ARM，并且现在也有很多大的半导体公司，例如，ST 和 PHILIPS 等早已推出面向低端市场的 ARM 系列产品。另外，ARM 公司推出的 Cortex-MX 系列的处理器，更是瞄准中低端市场，随着市场的扩大，ARM 处理器性价比将有更多的优势。

在本书每章的开始，都会事先把用到的相关专业术语罗列出来方便大家理解。

本书的读者对象

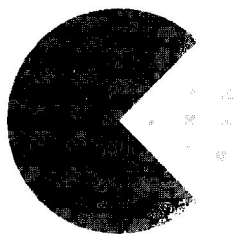
- 想入门嵌入式开发的人员，本书提供了环境搭建、CPU 原理以及整个嵌入式开发的各个环节知识。
- 从事嵌入式设备开发的专业人员，作者实际经验的分享，各个标准的通俗解读。
- 从事手持类设备应用开发的人员。

在本书内容中，由于本人的知识范围的局限性，错误之处在所难免，或存在理解上的误差，欢迎读者和嵌入式系统的专家批评指正，本人不胜感激，任何建议与意见可以发送到我的私人邮箱 broadken@eyou.com 或者 sunzbnn@hotmail.com。编辑联系邮箱：zhangtao@ptpress.com.cn。

目 录

第 1 章 关于本教程	1	6.3 认识 ARM CPU 的 MEMORY MAP	32
1.1 本书内容的组织	2	6.4 总线的概念	35
1.2 关于本教程涉及的实例	2	6.5 CPU 是如何操控外设的	37
1.3 计算机配置需求	3	第 7 章 快速 ARM 汇编语言	40
第 2 章 如何开始嵌入式开发	4	7.1 ARM 汇编文件格式	40
2.1 什么是嵌入式系统	4	7.2 认识基本的 ARM 指令	42
2.2 充分利用现有的硬件资源	5	第 8 章 ARM 体系结构	49
2.3 嵌入式开发入门第一步	5	8.1 认识嵌入式系统软件的组成	49
2.4 建立一个具体的嵌入式 开发的小项目	6	8.2 ARM 处理器及 Memory 系统	51
第 3 章 建立 Windows 下的交叉 编译环境	7	8.3 大小端的概念	52
3.1 Cygwin 的安装	7	8.4 ARM 存储系统	53
3.2 ARM 编译工具 RVDS 的安装	13	8.5 THUMB 与 ARM 指令	57
第 4 章 开发环境的介绍与使用	17	8.6 解析 Image 文件	57
4.1 Cygwin 的作用	17	8.7 认识 ARM 的编译工具	60
4.2 RVDS 的作用	17	8.8 汇编编译器	61
4.3 开发环境的使用	17	8.9 C & C++ 编译器 armcc	63
第 5 章 认识 ARM CPU 的工作原理	19	8.10 ARM 链接器 armlink	66
5.1 从新的角度认知 CPU	19	8.11 为什么有 Load region 和 execution region	71
5.2 ARM 系列的 CPU 概览	20	8.12 认识什么是 scatter 文件	73
5.3 ARM CPU 的内部初探	21	8.13 烧写到嵌入式设备里面的 文件到底是什么	81
5.4 ARM 处理器的工作模式	22	第 9 章 认识信号和时序	84
5.5 ARM CPU 的寄存器	24	9.1 硬件之间通信的桥梁——信号	84
5.6 关于寄存器的命名	25	9.2 什么是时序	86
5.7 ARM 寄存器的使用规则	25	9.3 LCD 时序分析	86
5.8 CPSR 和 SPSR 寄存器	27	9.4 如何满足外设的时序要求	89
5.9 本章小结	29	第 10 章 认识并使用 GPIO	91
第 6 章 从整体上认识 ARM CPU	30	10.1 认识 GPIO	91
6.1 实例 CPU 内核特性简介	30	10.2 如何配置并使用 GPIO	92
6.2 ARM CPU 的内部结构初探	31		

第 11 章 认识中断	95	14.9 读懂示波器上的 I2C	133
11.1 中断分为两类：内部中断、 外部中断	95	14.10 I2C 接口设备的调试	135
11.2 中断实例分析	96	第 15 章 SPI 接口驱动的开发	136
11.3 配置中断管脚	96	15.1 SPI 是什么	136
11.4 中断信号触发条件的配置	96	15.2 SPI 总线的特点	136
11.5 中断在哪里被处理	97	15.3 总线构成及信号类型	137
第 12 章 让你的 LCD 工作起来	100	15.4 SPI 的四种工作模式	137
12.1 LCD 的历史与原理	100	15.5 读懂示波器上的 SPI	139
12.2 为 LCD 编写驱动程序	102	15.6 SPI 接口设备的调试	140
12.3 LCD 的硬件信号及驱动 程序示例	103	第 16 章 认识 EBI 接口	142
第 13 章 NAND Flash 驱动的开发	109	16.1 EBI 硬件接口	142
13.1 认识 NAND Flash	109	16.2 访问 EBI 接口的外设	143
13.2 NAND Flash 的存储结构	111	16.3 EBI 接口的配置	145
13.3 坏块的概念	112	16.4 关于 page mode 与 burst mode 的理解	147
13.4 认识 NAND Flash 的特性	113	16.5 BURST 模式	148
13.5 NAND Flash 的硬件接口	114	第 17 章 利用 TRACE32 进行调试	150
13.6 NAND Flash 的操作命令说明	115	17.1 ARM 调试接口简介	150
13.7 NAND Flash 如何寻址	116	17.2 TRACE32 是什么	151
13.8 读取 NAND Flash ID	118	17.3 TRACE32 的安装	152
13.9 读取 NAND Flash 的页数据	119	17.4 TRACE32 调试功能原理	159
13.10 Page program, 给某个页 进行编程操作	120	17.5 TRACE32 能做什么	159
13.11 擦除 (Erase) 一个 block	122	17.6 运行 TRACE32	159
13.12 关于 8 bit 接口的 NAND Flash	124	17.7 如何把 TRACE32 与工程 关联起来	161
13.13 ECC 是什么	126	17.8 TRACE32 调试常用命令	164
13.14 NAND Flash 中一个 特殊的 block	127	17.9 常用窗口选项介绍	166
第 14 章 I2C 接口设备驱动的开发	128	17.10 Trace32 常用调试方法	169
14.1 I2C 是什么	128	第 18 章 认识硬件原理图	173
14.2 I2C 总线的特点	128	18.1 什么是硬件原理图	173
14.3 I2C 接口的模式	129	18.2 用掌握硬件原理图吗	173
14.4 总线的构成	129	18.3 硬件原理图好掌握吗	174
14.5 I2C 设备的寻址	130	18.4 理解一个简单的原理图	174
14.6 I2C 总线信号协议	130	第 19 章 后记	176
11.7 I2C 数据信号传输的规定	131	附录 A Glossary (缩略词汇表)	177
14.8 一个完整的 I2C 数据传递过程	132	附录 B 参考文献与相关网站	179



第1章 关于本教程

本书主要是针对从事嵌入式软件开发人员编写的,而嵌入式开发主要有两个方向,一个是从事嵌入式设备上应用程序的开发;另一个是驱动程序的开发,本书的方向更加侧重于驱动程序的开发。当然,本书对于从事应用程序的开发人员也是不错的参考,通过它可以很好地了解一个嵌入式设备是如何协调各种硬件一起工作的,对更好地进行应用层的开发是很有意义的。在一个具体的嵌入式产品中,上层软件主要是指具体的应用类程序,如可以在某一特定的硬件平台上开发一个游戏、开发一台照相机、开发一台录像机、开发一款 E-mail 收发程序等,这些具体的应用都非常直观地展现给了用户。而驱动程序的开发主要是针对硬件层面的设备进行软件开发,具体而言就是,为某一具体的硬件编写驱动程序,让这些硬件外设可以在特定的硬件平台上正确、稳定地工作,如使用手机上的照相机、LCD 显示屏、键盘以及 MP3、MP4 多媒体功能,要使这些设备工作,就需要先给照相机 (Camera Sensor)、LCD、多媒体等这些硬件编写驱动程序,让它们工作起来,然后结合上层的相关应用程序才组成了一个完整的应用程序。

本教程的潜在读者对象是谁?本教程主要是针对想从事嵌入式设备的驱动开发人员的。而具体到什么是驱动开发,简单意义上就是给某个设备编写驱动程序,并使之融合到整个系统中来。希望读完本书能让您对于嵌入式开发有一个全新的认识。



下面我们以一部手机为例来看一下驱动开发的工作内容是什么。我们知道一部手机包括很多的元器件,如屏幕 (LCD)、按键、触摸屏、耳机、收音机、照相机、录像机、录音机、T 卡、麦克风 (microphone) 等。这些都是可以直观看到和触摸到的,而在手机的内部还有 ARM CPU、Memory (内存系统)、Flash (NAND Flash 或者 Nor Flash 存储系统),电源管理单元,处理射频、音频的 DSP、USB、串口等。所有这些元件依靠印刷电路板 (PCB) 连接起来,然后通过一款巨大的软件工程把这些设备组织并管理起来,最后就变成了使用的手机了。所以说,软件是灵魂,硬件是躯体,两者缺一不可。

设备驱动开发的工作就是为这些设备编写程序,让这些设备准确地工作,并融入到手机系统中来,例如 LCD,就要为不同厂商的 LCD 编写适合某个手机开发平台的驱动程序,这样不同的 LCD 才能工作在不同的平台上,其他设备类似,但是驱动的工作并不局限于此。对于整个系统架构理解最深的也是驱动开发人员,只有对整个系统有透彻的、完整的理解才能开发出一个健全、健壮的系统。

本教程主要内容包括手机驱动开发设计中用到的常用外设的驱动程序的开发,包括如何写这些

驱动程序，如何去调试这些设备，出了问题如何进行跟踪定位。我们知道驱动程序的调试不仅仅是软件方面的，同时也会涉及硬件方面的内容，这就要求我们具有一定的硬件知识，最少也能读懂硬件原理图，能够根据原理图编写程序，能够拿起万用表、示波器这些常用硬件信号电性能测量设备进行硬件的调试。往往一个设备的成功运行都是建立在反复的测试和测量的基础上完成的，特别是设备间联调的时候，例如，手机系统要外加一个蓝牙、一个 WIFI、一个 CMMB 数字电视等。这个时候就需要我们和供应商的 FAE（Field Application Engineer 现场支持工程师）进行联调，这种联调工作是对我们嵌入式知识体系的一个综合考量，而往往这个时候板子还没有做好，只能通过飞线进行调试，所以调试环境恶劣，那么如何开始进行调试呢？这个疑问我们会在后面针对每一类设备进行详细讲解。

1.1 本书内容的组织

本书从嵌入式开发入门的角度进行阐述，首先假设读者具有一定的 C 语言基础，一些嵌入式方面的知识。在第 1 章对什么是驱动开发进行了阐述，读者可以对驱动开发有一个基本的概念上的认识。第 2 章讲述了什么是嵌入式开发以及学习嵌入式开发需要掌握的基本知识及要求。第 3 章建立自己的 Windows 系统下的交叉编译环境，让读者可以在 Windows 下进行交叉编译。第 4 章阐述该交叉编译环境的工作原理及使用的方法。第 5 章和第 6 章主要阐述 ARM CPU 的工作机制以及从开发者的角度是如何看待 CPU 的。第 7 章使用很小的篇幅阐述了 ARM 的汇编知识，达到认识基本 ARM 指令即可的要求。第 8 章阐述 ARM 的体系结构及各种关于 ARM CPU 使用的相关知识。第 9 章阐述硬件通信信号和时序的概念，让读者了解硬件是如何通过信号进行通信的。第 10 章阐述 GPIO 的配置及使用。第 11 章阐述 ARM CPU 中断配置及其使用。第 12 章阐述如何让 LCD 工作起来。第 13 章阐述 NAND Flash 的使用，其中包括了 16bit 和 8bit Flash 的使用。第 14 章结合时序图阐述了 IIC 接口的原理及使用。第 15 章也是结合时序图阐述了 SPI 接口的原理及使用。第 16 章阐述了 EBI 总线的使用。第 17 章阐述了 LAUTERBACH 公司的 TRACE32 的使用。第 18 章通过举例阐述了如何读懂硬件原理图。通过这些章节可以让读者了解一个完整的嵌入式系统开发需要具备的各个知识点，可以沿着这些知识点进行更深入的学习相关开发技术。

1.2 关于本教程涉及的实例

在后续的章节中会经常提及**实例**这个词，此处的**实例**指的是我们进行开发实验的一整套包括软件、硬件的整体开发和调试环境，软件方面的开发环境可以依据后续章节中的关于如何建立自己的开发环境进行搭建，有了软件开发环境，同样需要有一套硬件开发板来支撑后续程序的开发调试与功能的验证，这里选用的是一套手机开发的硬件平台，当然也可以使用一些 ARM 开发板作为硬件环境，因为我们建立的开发环境只要稍做修改就可以适配各种编译器、各种 CPU，是不依赖于某一类 CPU 的。

那么利用哪种硬件平台来进行开发实验与调试呢？这里可能会存在一个误区，就是本书的内容是不是只能针对某一具体的硬件平台呢？答案当然是否定的，因为本书主要内容是针对 ARM 系列

的 CPU 和各种常用的外设如何使用来展开的,通过本书掌握 ARM 系列的 CPU 的使用以及各种常用外设的使用及其调试方法,所有内容并不拘泥于特定硬件平台的,所以,可以充分利用现有的硬件环境资源。

下面举一个例子进行说明,比如一块具体的 LCD 显示屏,它是可以连接到不同的硬件平台上的,它之所以能够运行于不同的硬件平台上,就是因为该 LCD 的硬件接口是统一的,不同的硬件平台要支持该类型接口的 LCD 就必须在 CPU 上提供相应的硬件接口,我们只要掌握了该接口的原理,对于不同的硬件平台某一特定类型的接口基本都是相同的,唯一不同的就是对于特定的硬件在接口配置方面有些不同,但是 LCD 这个硬件的接口是固定的。所以,只要掌握了关键的几个知识点并在不同的硬件平台上进行这几个关键点的配置,就可以让 LCD 运行在自己的硬件平台上。

所以嵌入式系统的开发学习的关键是找到正确的方法与方向,这样思想就不会拘泥于某一特定硬件环境,而是可以从一个产品的层面进行思考、设计。

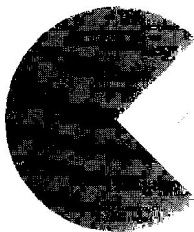
另外,学习过程中非常重要的一点就是借鉴他人的开发成果往上是最高效的学习方法,在以后的讨论学习中慢慢体会吧。

正如牛顿所说的,如果说我比别人看得更远些,那是因为我站在了巨人的肩上。

1.3 计算机配置需求

能够运行实例及运行编译开发环境的计算机配置要求如下。

- 操作系统: Windows XP/Windows 7 (版本不限)。
- 硬盘空间: 大于等于 4 GB, 主要用于安装开发环境所需软件。
- 内存: 大于等于 512 MB。



第2章 如何开始嵌入式开发

2.1 什么是嵌入式系统

在进行嵌入式开发的讨论之前，先从概念这一层面来认识什么是嵌入式系统。从宏观上来说，一个嵌入式系统一般是一个被特意设计来做某一个或某几个特定功能的并且可能需要具备实时计算要求的系统。简单的嵌入式系统比如常见的空调、洗衣机等家用产品，这些产品就是为特定目的而设计的，它们并没有实时计算的需求，只要能把事情做完目的也就达到了。而一些涉及通信、安全、交通的嵌入式设备就必须有实时计算的要求，如手机和网络之间的通信、现代汽车的电子辅助系统都要求具有实时计算的能力，因为这些关系到了实时通信，以及关系到人身安全的问题。对于空调、洗衣机、手机之类的产品都是可直接感觉到的，是作为一个独立的产品。而另外一些嵌入式系统会作为一个具有硬件以及机械机构的完整设备的一部分，典型的例子就是交通、工程类的各种机械车辆。嵌入式系统相对于通用的计算机，就像个人 PC，是被设计来做更广的应用满足更多的终端客户。

今天，嵌入式系统控制着越来越多的设备，而一个嵌入式系统一般是一个主控制器或者一个 DSP 来作为主控制器，并执行着某些特定的任务。由于嵌入式系统专注于特定任务，那么设计者就可以在体积大小上、成本上来优化，同时可以提高它的稳定性和性能。

从应用层面讲，嵌入式系统从手持设备，例如从电子表到 MP3，再到大型的嵌入式设备，如交通信号灯、工厂控制系统，更有核能工厂，复杂度从单个控制器，到多个控制单元，其应用涵盖了各行各业的方方面面。

一般而言，嵌入式系统并没有一个严格意义上的定义，因为现在很多嵌入式系统都是具有可扩展和编程的，如具有像操作系统和微处理器的一些手持的嵌入式设备，它们是可以加载不同的应用程序以及通过硬件接口与其他设备进行连接共享，此外即使系统不具有可编程的特性，那么一般也是可以支持软件升级的。

目前为止，嵌入式设备在我们生活的方方面面占据了非常重要的地位，嵌入式系统再不是以前仅仅作为一个孤立的设备，随着物联网的推进，许多的嵌入式产品将被连接到一起，在任何可以联网的地方就可以让远端的设备为我们服务。

当然这些都是未来的嵌入式产品的发展方向，而我们现在要做的就是掌握好嵌入式开发的知识，为将来物联网的蓬勃发展做好准备工作，俗话说，成功都是留给有准备的人的。

2.2 充分利用现有的硬件资源

拥有一个开发板对于学习嵌入式开发是必不可少的，如何选择一个开发板来进行开发实验，这个会让很多人头疼的问题。因为大家会在开发板的价格上犹豫是选择 ARM7、还是 ARM9，甚至是选择 Cortex A8、Cortex A9 之间纠结半天。另外就是开发板的利用率也是个很大的问题，我想很多人基本上是把实例跑一遍板子也就束之高阁了，利用率确实不高。其实，对于嵌入式开发的学习，买一块 ARM7 或者 ARM9 的、接口稍微丰富一点的、经济型的开发板就可以了，不需要考虑太多功能的开发板，我们要学习的是嵌入式开发的最基础的知识，而所谓的丰富功能的开发板都是建立在这些基础知识之上的，充分利用好手头现有的开发板资源即可，本书的内容没有针对任何具体的开发板。

2.3 嵌入式开发入门第一步

当我们谈论嵌入式开发的时候，最常听到的就是嵌入式开发是如何地难学，比如既需要一定的开发语言基础，又需要相关的数、模、电方面的知识，难道嵌入式真的那么难吗？其实难与不难最关键是一个入门的问题，等真正入了门就会发现先前的努力与汗水都是值得的，你会迈入一个全新的世界。入门嵌入式开发确实有一定的难度，因为它不像学习 Java，只要有一台 PC，然后按照教科书的例子一个个抄下来，通过一段时间的学习就能编写小的程序了，并且在 PC 上能立刻看到执行的效果，也可以一步一步地跟踪调试编写的程序，直到你的 Java 程序完美实现了原来的设计。

而嵌入式开发就没有这么简单了，首先在 PC 上不能立即看到运行的效果，需要建立一个交叉编译环境，用这个环境编译、链接写的程序，然后把链接后的程序烧写到板子上的 Flash 里，然后给板子加电，让程序在板子上去执行，程序的执行结果并不像 PC 程序中 Java 那样，立刻就能直观地看到结果，如果程序执行不正确也不能像 PC 上那样简单的一步一步地跟踪调试，需要借助第三方的工具才能一步一步地跟踪调试，所以也相应地增加了难度。

另外如下的知识点是需要重点掌握的。

- 掌握 ARM 系列的 CPU，要求对 ARM CPU 的体系结构有深入的理解，并且能够做到熟悉一款特定 ARM CPU 的总线、时钟、外设接口，以及通过 Memory map 迅速了解它的大概配置。
- 掌握常用的外设，如 LCD、各种存储设备（NAND flash 或者 Nor Flash）的使用方法。
- 掌握常用的外设接口，如 IIC、SPI、MIPI、EBI 等。
- 熟悉硬件原理图，能够看懂基本的硬件原理。可以说硬件原理图是硬件和软件的接口的表达，要做到通过读硬件原理图能够编写相应的软件。
- 掌握常用的调试手段，如串口 log 跟踪，JTAG 工具的熟练使用，这些工具可以协助我们跟踪程序内部流程，能够直观看到程序流。
- 扎实的 C 语言基础，我们可以看到嵌入式开发充分利用了 C 语言的灵活，嵌入式开发是 C 语言灵活展示的舞台，另外，通过读别人的代码可以了解设计者的思想，同时也可以去学习并发扬有价值的设计思想。

当你掌握了一款 ARM CPU 之后,那么对于其他使用 ARM CPU 的嵌入式产品来说,你也会非常容易上手,并能很快对整个系统有一个很好的认识。另外驱动程序的研发工作不光是写好一个设备的驱动程序,同时也是要架构一个驱动程序框架,可以让同类的设备方便地加入该框架之下,让框架来管理这些相似设备的驱动,一个好的框架设计对于一个项目来说是至关重要的。

2.4 建立一个具体的嵌入式开发的小项目

为了照顾到学习的效率性,我们先设计一个嵌入式开发的任务,这样就可以带着疑问进行学习,所以本教程会带领初学者从建立一个简单但功能完备的开发环境开始。从编译开发环境的使用,再到 ARM CPU 的体系结构,再到各种外设、外设接口的使用,一步步地认识什么是嵌入式设备驱动开发。最后把这些知识综合起来开发一个完备的世嘉游戏模拟器,让原来运行在世嘉游戏机上的游戏运行在 ARM CPU 上,不过本教程主要是嵌入式系统知识,所以并不提供该游戏模拟器的软件代码,需要者可以发邮件给作者索取源代码,邮件地址 sunzbnn@hotmail.com 或者 broadken@eyou.com。

本教程使用的硬件环境。

- CPU: ARM926EJ CPU (500MHZ)。
- Memory: NAND Flash (128MB) +DDR RAM (128MB)。
- LCD: 16Bit 色 TFT 彩色 LCD。

完成该项目步骤可分为。

- 熟悉编译环境。
- 掌握 ARM 系列的 CPU 体系结构。
- 熟悉各种硬件外设。
- 熟悉各种硬件接口。
- 编写 Boot 程序引导 CPU 的启动,可以在 Boot 中初始化关键硬件。
- 加载主程序到 Memory 中运行。
- 在主程序中完成所有硬件初始化工作。
- 编写项目具体软件,并使之运行在 ARM CPU 上。

当然以上内容是每个嵌入式项目都不可或缺的,是一个项目需求实现的基础部分,在此基础上加上一个嵌入式 OS 和简单的 GUI 界面,就可以很完备地展现这个项目了,那么从后面的章节开始一步步阐述这些内容。



第3章 建立 Windows 下的交叉编译环境

在本章我们将建立自己的编译开发环境,尽管 ARM 公司已经提供了完善的 IDE 集成开发环境,但是在实际应用当中,这些 IDE 是不足以提供系统开发的各种需求的,还是需要建立自己的编译开发环境,这样可以对整个工程有更好的配置和管理,为以后的开发工作提供灵活方便的开发调试环境,而不需要依赖于某一个特定版本的 IDE 环境。ARM 公司会根据新推出的 CPU 而发布新版本的 IDE 环境,以适配新推出的 CPU 类型。如果我们拥有自己的开发环境,那么就可以不受新版本 IDE 更新带来的影响,同时也方便于对不同版本的 CPU 进行配置,往往在实际项目当中一款软件工程需要适配不同的硬件配置以满足不同产品的需求,所以一个完备的开发环境是不可或缺的。

实例中使用的开发环境是完全建立在 Windows 系统上的,该开发环境使用的也是 make file 的机制,之所以把它建立在 Windows 系统上,是因为更多的人对 Windows 系统会比 UNIX 系统更熟悉一些,如果把它建立在 UNIX 或者 Linux 系统上,这样的话就会引入偏离了我们学习方向的因素,因为在学习嵌入式开发之前要先掌握一些 UNIX 的知识,否则没法开展后续的工作,这就是在实际的学习过程当中遇到知识交叉的情况。简单的例子就是本来我们要学习 A 的,结果 A 需要 B 知识我们转而又去学习 B,而 B 又需要 C 的知识,而 C 可能又牵涉其他的知识点,太多的知识交叉点就会妨碍我们的学习目标了,弄不好还会打消学习的积极性。最好的办法就是摒弃其他,专攻一个,熟悉了一个然后再去学习另外一个,尽量别把众多知识掺和到一起学,否则效果往往不是很理想。

实例中使用的交叉编译环境同样是建立在前人积累的丰富软件之上的了,所以使用 make 文件来组织编译环境。要解析 make 文件就需要 make 程序,另外还有一些 shell 脚本需要解析,所以我们安装了 Windows 系统下的一个 UNIX 类操作系统的模拟软件 Cygwin, Cygwin 提供了 make 程序,能够很好地解析 make 文件,并准确地执行,当然也提供了相应的 Shell 程序,而编译环境需要也就是 make 程序和 shell 脚本解析程序。

Cygwin 是个开源软件,让我们可以在 Windows 系统上使用 UNIX 类操作系统上的功能,当然可以在这个软件下执行 UNIX 下的常用命令,是一个非常不错的 UNIX 的学习环境,等熟悉了一些后就可以去真正的 UNIX 下体验一下。

编译环境使用的 Cygwin 的版本是: 1.7.9-1, 关于 Cygwin 的安装程序以及更多的信息可以参考 Cygwin 的网站 <http://cygwin.com>。

3.1 Cygwin 的安装

Cygwin 是一款开源软件,我想正是这些开源软件推动了软件业的极大发展,让人们使用并获

得比商业软件更好的软件与思想，如果只有商业软件，那么某些技术与思想将会被禁锢，所以要感谢这些无私的开源软件作者，是他们打破了这些禁锢。下面开始依次安装这些软件。

在后续的安装过程中，都是假设 Windows 系统是安装在 C 盘根目录下，如果 Windows 系统安装在其他盘下，那么安装目录请做相应修改。我的电脑光驱盘符为 H 盘，请注意你自己的光驱所在盘符，也请做相应的修改。Cygwin 的安装光盘是从网上下载并刻录的，是为了方便安装过程的说明，为节约成本请从网上直接下载到本地进行安装。

Cygwin 版本：1.7.x，可到 <http://cygwin.com> 网站下载安装最新版本。

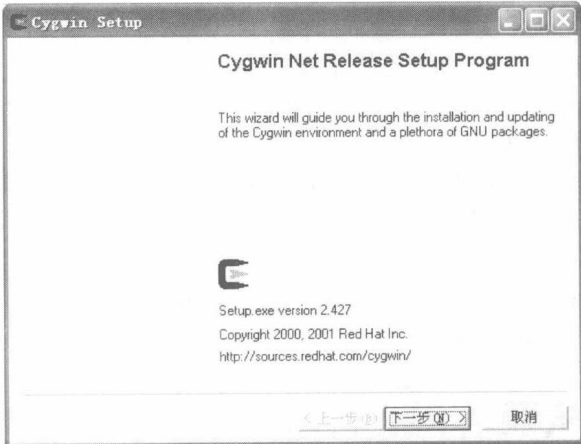
安装环境：Windows XP / Windows 7

第一步：找到 Cygwin 的安装程序 setup.exe，双击该安装程序开始安装，如图 3-1 所示。



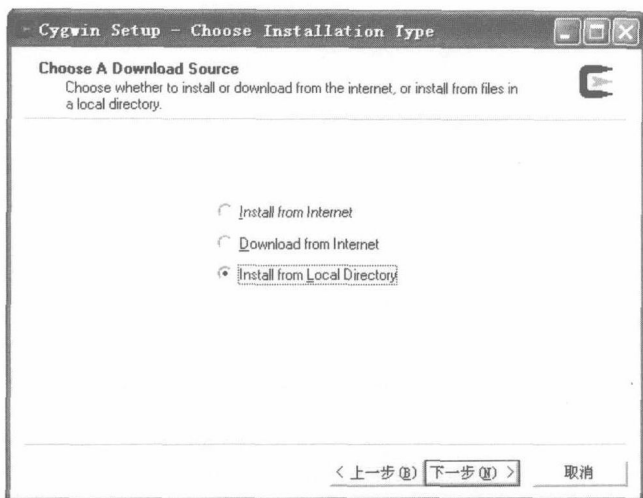
▲图 3-1 Cygwin 安装程序

第二步：出现图 3-2 所示的界面后，直接单击『下一步』按钮开始安装。



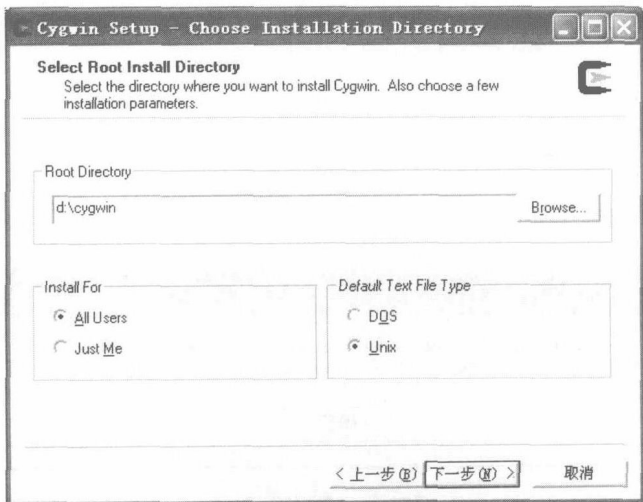
▲图 3-2

第三步：出现图 3-3 所示的界面后，选择 *Install from Local Directory*（从本地安装），如果没有安装光盘，可以选择 *install from internet*（从网络直接安装），或者选择 *Download from Internet* 先下载到本地，下载完成后再进行安装，然后单击【下一步】按钮继续安装。



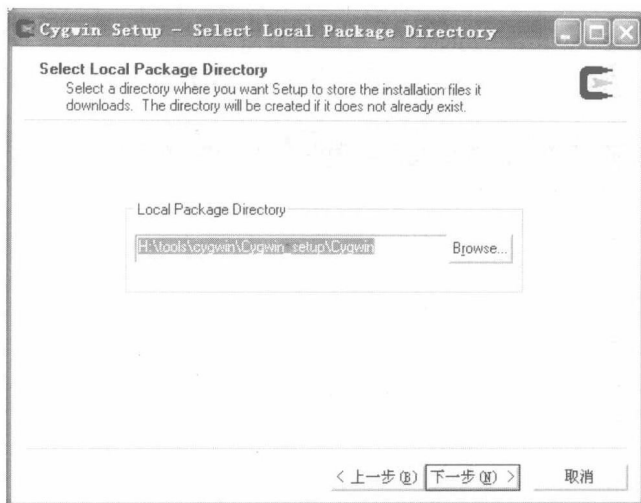
▲图 3-3 选择 Install from Local Directory

第四步：出现图 3-4 所示的界面后，输入 *C:\cygwin* 作为安装目录，当然也可以使用其他目录，就如图 3-4 所示的 *d:\cygwin* 目录，建议安装在 *C:\cygwin* 目录下，然后单击【下一步】按钮。



▲图 3-4 安装目录选择框

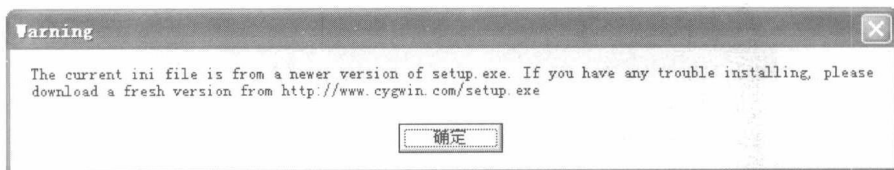
第五步：由于第三步中选择的是从本地安装，所以会出现图 3-5 所示的选择 Cygwin 安装程序所在目录的选择框，一般安装程序会自动填写目录，所以直接单击【下一步】按钮继续安装。



▲图 3-5 安装程序所在路径选择

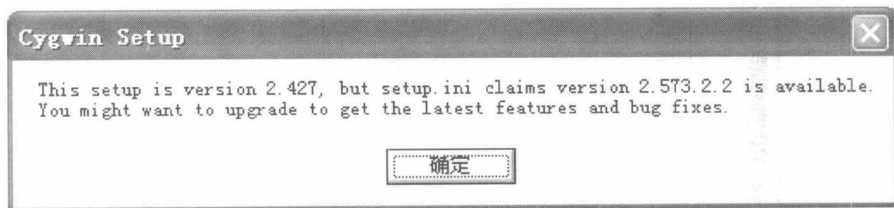
第六步：第五步之后，可能会弹出如图 3-6 所示的警告提示框，直接点击『确定』按钮，忽略该警告即可。

稍后如果多次弹出如图 3-6 所示的警告窗口，直接点击『确定』按钮忽略此类警告即可，该警告是因为 ini 文件与 setup 程序版本不匹配造成的。



▲图 3-6 安装警告提示

第七步：稍后可能会弹出如图 3-7 所示的提示窗口，直接点击『确定』按钮继续。



▲图 3-7 提示窗口

第八步：安装前的准备，验证 Cygwin 包含的各个软件的完整性（MD5 验证），如图 3-8 所示。

第九步：出现如图 3-9 所示的安装界面后，点击列表中的第一项 +All 中的 **Default**，直到列表中的 Default 都变为 Install，然后点击『下一步』按钮继续安装。