

基于 ARM 的 Windows CE 移植及应用开发

夏 军,胡景春

(南昌航空大学 信息工程学院,江西 南昌 330063)

摘 要:Windows CE 操作系统定制是 Internet 嵌入式虚拟仪表应用平台开发中的重要环节。结合硬件资源的相关性,分析了 BSP 的框架结构和执行流程。提出了基于 S3C2440 处理器的 Windows CE5.0 平台下 BSP 开发步骤,详细说明了 Boot loader, OAL, 驱动程序和配置文件的设计方法,充分利用代码重用思想,完成了整个操作系统的移植。基于此平台完成设备监控终端软件设计,这对整个系统的后续研究和完善奠定了基础。实验表明操作系统运行稳定,监控终端能持续显示波形并与 PC 机通信,具备良好的可靠性。

关键词:S3C2440; Windows CE 5.0; BSP; 监控终端

中图分类号: TP273+.5

文献标识码: A

文章编号: 1673-629X(2011)04-0129-04

The Migration and Development of Windows CE Based on ARM

XIA Jun, HU Jing-chun

(Dept. of Information Engineering, Nanchang Hangkong University, Nanchang 330063)

Abstract: In the development of embedded Internet virtual instrument application platform, the customization of Windows CE is a critical section. According to the correlation of hardware resources, the frame construction and flow process of BSP was analyzed. BSP development steps of Windows CE5.0 based on S3C2440 was proposed. A design method of Boot loader, OAL, device driver and configuration files was stated in detail. Transplant of the whole OS was accomplished by using the thought of code reuse. A monitor terminal was designed on the platform. The foundation was formed for further research and improvement. The experiment showed that the system ran stably and the monitor terminal displayed smoothly and communicated with PC continuously.

Key words: S3C2440; Windows CE 5.0; BSP; monitor terminal

0 引 言

随着科技进步和后 PC 时代的到来,嵌入式技术的应用范围越来越广泛,嵌入式产品也充斥着各个领域,而嵌入式系统是这些应用的核心,是连接嵌入式设备和应用的重要部件^[1]。Windows CE 是微软公司面向嵌入式领域推出的一个功能强大的实时嵌入式操作系统。Windows CE 提供一系列集成定制和开发工具,能方便地按需进行系统定制。在微处理器方面,S3C2440 是 Samsung 公司推出的一款基于 ARM920T 内核的 16/32 位 RISC 嵌入式 CPU,最高工作频率 533MHz。在常用电信号测量系统中,对实时性和数据处理能力有很高的要求,而以 Windows CE 操作系统为核心的 ARM 板能满足系统需求。根据不同应用场合,它可以作为连接多个分布式测量终端的集散式数据处

理服务器。

文中阐述系统开发过程中的重要环节——Windows CE 5.0 操作系统的移植。

1 Windows CE 概述

1.1 Windows CE 体系结构

Windows CE 5.0 是微软推出的 32 位实时较高的操作系统,主要针对嵌入式领域,它的应用范围涉及移动设备至工业自动化控制领域^[2]。由于 Windows CE 的灵活性,开发者可以选择合适的组件和模块完成个性化操作系统的定制。同时,Windows CE 在设计之初就采取了分层的结构化设计,从底层向上依次为:硬件层、OEM 层、操作系统层和应用层^[3]。

硬件层除了处理器、存储器基本设备外,还包括一些特定的外设,如网络接口、数据采集模块、触摸屏模块和按键等。这也是 Windows CE 操作系统开发的基础。

OEM 层是 Windows CE 操作系统和底层交互的中间层,很多操作系统级别功能代码就是在这里面实现的。根据不同应用的需求,OEM 的实际大小可以变

收稿日期:2010-09-02;修回日期:2010-12-25

基金项目:江西省教育科研基金项目(GJJ08222)

作者简介:夏 军(1987-),男,江西抚州人,硕士,主要研究领域为计算机控制与智能系统;胡景春,教授,主要研究领域为计算机智能控制系统及虚拟仪表研究。

化,其组成模块有:

(1)Boot loader。它的主要作用是初始化硬件设备,从而将系统的软/硬件环境带到一个合适的状态,为最终调用操作系统内核做准备^[4]。最常见的 Boot loader 通过以太网从上位机下载操作系统到目标机,称之为 EBoot(Ethernet Boot)。

(2)OAL(OEM 抽象层)。它是操作系统内核抽象出来的与硬件交互的接口,通过调用内核库文件产生内核可执行文件^[5]。OAL 主要负责硬件平台的初始化,包括中断服务例程(Interrupt Service Routines, ISR)、计时器(Timer)、通用 I/O 口控制、内核调试^[5]、开关中断和内核性能监测等工作。

(3)配置文件。它是一些包含配置信息的文本文件。BSP 中的配置文件包括 BIB, DB, REG 和 DAT 4 类平台初始化文件,还包括 Sources, DIRS 和 CEC 文件,这些配置信息告诉编译系统如何配置最终的操作系统映像文件。

(4)设备驱动程序。把物理外设或虚拟设备抽象成软件接口,主要用于操作系统对外设或虚拟设备进行操作,应用程序可以调用设备驱动产生的动态链接库文件达到控制设备的目的。

操作系统层是微软提供的丰富的组件库,是 Windows CE 的核心层,它既为 OEM 层提供注册、调用接口和服务,也为应用程序层提供服务 and 应用程序编程接口^[6]。在 Windows CE5.0 中,操作系统层核心代码主要集中于 NK.exe 模块,另外还包括 Core DLL, 文件系统,图形窗口事件子系统,设备管理和各种服务。

应用层主要包括 Windows CE 应用程序、Internet 客户服务、用户接口、客户应用程序四个模块^[7]。

1.2 Windows CE 开发工具

Platform Builder(下面统称 PB)是微软提供给开发人员进行 Windows CE 系统内核定制的综合开发环境,它提供了所有进行设计、创建、编译、测试和调试 Windows CE 操作系统平台的工具^[8],开发人员可以通过它选择自己所需的组件模块添加到操作系统中。同时,PB 也可以用来开发驱动程序和应用程序,并且可以一起打包到系统中。

一般应用程序的开发可以利用 EVC(Microsoft Embedded Visual C++)以及 Microsoft Visual Studio 完成。常用电信号测量系统中 Windows CE 部分主要采用的开发工具是 PB5.0, Visual Studio 2005 及 PB 自带的测试工具。

2 Windows CE 系统移植

2.1 Windows CE 系统定制

Windows CE 的移植过程主要是修改 BSP(Board Support Package)。BSP 又称板级支持包,它是一个包括启动程序、OEM 适配层程序(OAL)标准开发板(SDB)和相关硬件设备的驱动程序的软件包^[9]。Windows CE 系统定制流程如图 1 所示,主要进行以下工作:

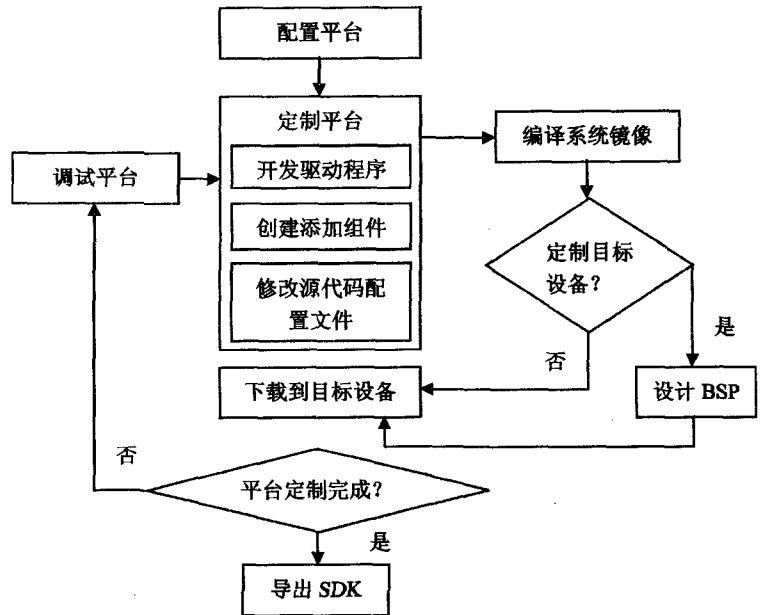


图 1 Windows CE 系统定制流程

- (1)编写基于特定目标设备的 BSP;
- (2)根据需要添加和裁减 Windows CE 组件,并修改相应的配置文件;
- (3)编译内核、组件和 BSP,生成 OS 映像文件;
- (4)将映像文件下载到目标设备上,进行调试;
- (5)重复修改、创建、下载和调试的过程,直到得到所需的 OS 映像;
- (6)为方便应用程序开发,需导出 SDK。

2.2 BSP 开发

开发 BSP 是一项比较复杂的工作,为了提高开发效率,一般都是基于 Windows CE 自带的 BSP 源代码做相应的修改。主要开发步骤为:硬件开发设计和测试,开发 Boot loader,开发 OAL,添加驱动程序,添加电源管理,发布 BSP。

2.2.1 Boot Loader 移植

系统加电后执行的第一条指令就是 Boot loader 的代码^[10,11],所以,Boot loader 通常位于目标设备的非易失性存储设备中,并且在系统加电或重启时自动执行^[10,11]。Boot loader 的主要作用分为 3 类:初始化硬件设备,包括内存、中断控制器和 MMU 等;一般通过串口打印整个 Boot loader 运行过程中的输出信息,并

提供与用户交互的接口,用户可选择启动过程和配置相关参数;下载并执行操作系统镜像。

Boot loader 的执行过程如下:

(1)执行 Startup. s 文件。Startup. s 一般用汇编语言编写,把 CPU 设置为合适的状态。主要工作是清空 TLB,使看门狗、指令和数据 Cache 失效,关闭所有中断和 MMU,设置锁相环频率。设置完栈指针后,打开可读写权限位,打开 MMU 进行物理和虚拟地址映射,打开 Cache,清空 RAM,拷贝 Boot loader 代码到 RAM 中并执行,最后跳转到 C 语言的 Main() 函数。

(2)Main() 函数是 Boot loader 中执行的第一个 C 语言函数,它需要 OEM 来编码实现,主要是调用 BLCommon 中的 BootloaderMain()。BootloaderMain() 控制整个启动代码的执行流程。

(3)全局变量重定位函数 KernelRelocate()。BootloaderMain() 第一个调用的就是 KernelRelocate(), 作用是把 Boot loader 中的全局变量定位在 RAM 中。

(4)调试端口初始化函数 OEMDebugInit()。主要作用是初始化硬件调试端口,一般是初始化第一个 UART 口,这样可以方便地在微机端获取调试信息。

(5)平台初始化函数 OEMPlatformInit()。主要任务是初始化目标平台,包括 BSP 参数,USB,块设备驱动,Bin 文件系统分区,以太网。同时,在 PC 机的终端软件中看到输出信息,用户可以选择菜单进行相关操作。

(6)下载镜像准备函数 OEMPreDownload()。完成镜像下载前的工作,包括获取 IP 地址,初始化 TFTP 等。

(7)下载镜像函数 DownloadImage()。其作用下载系统镜像到目标设备。除此之外,它还须解析下载文件的格式,并把映像文件的长度、大小和运行起始地址返回给 BLCommon 库。

(8)启动镜像函数 OEMLaunch()。首先,OEMLaunch() 函数会判断用户是否需要将镜像写入 Smart Media Card,接着把 Bootloader 参数写入 NAND Flash 当中,然后写入用户配置信息和启动配置信息,完成虚拟地址到物理地址转换后,调用 Launch() 函数进行实际的跳转。

Boot loader 移植的主要过程有:

1)修改相应的 DIR 和 Sources 文件,这里列出部分库路径:

```
TARGETLIBS=\
$( _PLATCOMMONLIB ) \ $( _CPUINDPATH ) \ oal_blc-
mon. lib \
$( _COMMONOAKROOT ) \ lib \ $( _CPUINDPATH ) \ eboot.
lib \
$( _TARGETPLATROOT ) \ lib \ $( _CPUINDPATH ) \ dm9000
```

```
_debug. lib \
$( _TARGETPLATROOT ) \ lib \ $( _CPUINDPATH ) \ smflash
_eboot. lib. lib \
$( _COMMONOAKROOT ) \ lib \ $( _CPUINDPATH ) \ full-
libc. lib \
```

其中 fulllibc. lib 文件是专供 Boot loader 和 OAL 使用的 C 语言函数库,库中的函数可在不依赖操作系统的情况下运行,因此在操作系统启动前,代码中使用的 C 语言函数,必须通过 fulllibc. lib 实现。

2)BIB 文件包含了目标设备上的内存分配信息,ROM 配置信息和需要打包的文件列表。为了让 Platform Builder 知道目标设备上哪些地址是可读写的,需要修改 Boot. bib 文件。下面是修改后的部分内容:

```
MEMORY
; Name      Start      Size      Type
RAM         80021000  00006000  RAM
STACK      80027000  00009000  RESERVED
EBOOT      80038000  00019000  RAMIMAGE

CONFIG
ROMSTART=80038000
ROMSIZE=19000
MODULES
;NamePath Memory Type
nk. exe      $( _TARGETPLATROOT ) \ target \ $( _
TGTCPU ) \ $( WINCEDEBUG ) \ eboot. exe      EBOOT
```

由于 Windows CE 操作系统会按 1:1 的比例把物理内存映射到虚拟地址的 0x80000000 和 0xA0000000 处。因此,如果 EBOOT 要写入的物理地址是 0x00038000,在配置文件中要写成 0x80038000。

2.2.2 OAL 移植

OAL 是操作系统内核的一部分,运行在内核态下,可直接对硬件资源进行访问^[12]。在系统构建构成中,OAL 代码被编译成 OAL. lib 库文件,与其他一些库文件进行统一的链接^[13]。因此 OAL 的启动过程也是整个操作系统的启动过程,OAL 执行大致经历了启动、内核启动、ARM 芯片初始化、内核初始化和第一个任务调度的过程。

(1)Startup() 主要完成硬件的初始化工作,如果 OAL 是通过 Boot loader 引导执行的,那么很多硬件设备无须重复初始化,仅需跳转到 OAL 的主控函数 KernelStart() 开始执行。Startup. s 核心部分代码如下:

```
.....
INCLUDE oemaddrtab_cfg. inc
IEAF_ENTRY StartUp
.....
add    r0, pc, #g_oalAddressTable - (. + 8)
bl     KernelStart
```

KernelStart() 需要一个参数:g_oalAddressTable 表的物理地址。在 ARM 代码中,此值被放在寄存器 r0

中,由于 ARM 采用流水线结构,允许指令预取,所以载入 r0 中的值是预取指令的前一条指令,这也是-(. + 8)的原因。

(2)KernelStart()主要完成内核的最小初始化并且通过调用 OEMInit()函数完成板级硬件初始化^[14]。主要过程是:初始化页表、代开 MMU 和 Cache,设置异常向量跳转表,初始化栈。这样可以安全地进入 C 语言代码执行。

(3)ARMInit()函数的主要功能是初始化基于 ARM 的硬件平台(Cache、中断、时钟、内核传输层)。ARMInit()调用的最重要的函数是 OEMInit(),它几乎完成了所有的硬件初始化工作。

(4)KernelInit()函数完成初始化内核工作,大致工作过程是初始化系统 API 函数调用表,然后依次初始化系统堆、内存池、第 1 个进程和线程。

(5)FirstSchedule()调度实际上不是一个函数,而是 armtrap.s 文件中的一个标签。系统会调用 HandleException()函数并传入 ID_RESCCHEDULE 参数,作用是让第 1 个处于就绪态的线程上台执行。

3 应用程序开发

常用电信号测量系统要求服务器端能对采集的电压、电流、频率等信号进行处理和显示,不仅能反应各个采集终端的实时状态,而且能对特定的采集终端发送命令。这样不论在世界的任何角落,只要服务器和采集终端连接上 Internet,服务器就能实时地反应各个终端的状态并且控制这些设备。

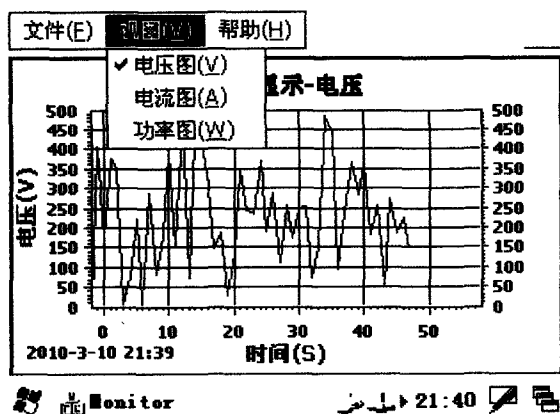


图 2 常用电信号测量与控制模拟软件运行效果图

在 Windows CE 设备上开发应用程序时,首先要利用 Platform Builder 导出定制平台的 SDK,这一步是在编译完操作系统后进行的。安装好 SDK 后,可以使用 Visual Studio 2005 开发应用程序。图 2 为在一个所定制的操作系统的常用电信号测量与控制的模拟软件运行结果图,通过内部函数随机产生电压值模拟显示效果。

4 结束语

嵌入式硬件设备千变万化,移植操作系统时需要根据不同的硬件环境进行相应地配置,从而满足不同的应用要求。Windows CE 的 BSP 是一个复杂的函数集,需要详细了解 BSP 的目录组织结构和各类配置文件的含义,而且很多函数与硬件相关性很大。文中没有详细讲解每个 Boot loader 和 OAL 函数,而是就一些遇到的重要函数和细节问题进行层次划分,同时归纳了整个开发流程,提出了开发移植 Windows CE 操作系统的一种方法和思路。这不仅适合于基于 S3C2440 处理器的移植,而且对其他平台的嵌入式开发也有参考价值。

参考文献:

- [1] 何宗键. Windows CE 嵌入式系统[M]. 北京:北京航空航天大学出版社,2006.
- [2] Dedicated Systems Experts NV. RTOS Evaluation Project [EB/OL]. 2004-08-18. <http://download.microsoft.com/download/7/2/f/72fef3b0-9545-46a4-8886-a94f265df9c4/EVA-2.9-OS-CE-01-101.pdf>.
- [3] 吕秀平,李小民. Windows CE. NET 的定制和裁剪[J]. 科学技术与工程,2006(21):3483-3486.
- [4] 辛雁峰,夏海宝. 基于 S3C2440 移动终端的 Windows CE 开发[J]. 微计算机信息,2009(3-2):71-73.
- [5] Developing an OEM Adaptation Layer[EB/OL]. 2006-01. <http://msdn.microsoft.com/en-us/library/aa447042.aspx>.
- [6] 司浩乐. 基于 Windows CE 的显示控制手柄设计[D]. 西安:西安电子科技大学,2008.
- [7] 李瑛达. 车载导航系统的硬件设计与 WinCE 系统内核定制[D]. 大连:大连海事大学,2007.
- [8] 方鑫,党宏社. 发射机控制器显示控制手柄的 Windows CE 操作系统移植[J]. 陕西科技大学学报,2009(2):76-80.
- [9] 丁玉芳,李允俊. 嵌入式操作系统 Windows CE. NET 在 S3C2410 平台上的移植[J]. 延边大学学报,2007(1):56-59.
- [10] 彭红五. 基于 WinCE 的车载监控终端的研究与设计[D]. 哈尔滨:哈尔滨工程大学,2008.
- [11] 张立为. 基于 ARM9 的加油站液位仪控制台设计[D]. 上海:华东师范大学,2009.
- [12] Plagge M. Microsoft Windows CE 5.0 Board Support Package, Boot Loader, and Kernel Startup Sequence[EB/OL]. 2004-05. <http://msdn.microsoft.com/en-us/library/aa446905.aspx>.
- [13] 何剑锋,李祥. 基于 XScale PXA270 处理器平台 Windows CE5.0 系统的 BSP 二次开发[J]. 化工自动化及仪表,2009(4):72-75.
- [14] 齐晓静,王卫东. Windows CE OAL 层的结构与开发[J]. 单片机与嵌入式系统应用,2005(2):30-33.