

基于 Windows CE 的可视电话的研究与实现

尹晓翠, 王 盟

(东营职业学院 山东 东营 257091)

摘 要: 在分析和比较当代比较流行的嵌入式操作系统如嵌入式 Windows CE、嵌入式 Linux, VxWorks, Palms 等的基础上, 结合系统所使用的 StrongARM CPU 性能、资源、功耗的特点, 提出了基于嵌入式微处理器 StrongARM 和嵌入式操作系统 Windows CE 的可视电话的多媒体控制板的总体方案。对嵌入式 Windows CE 进行了平台定制, 设计和开发了系统引导程序、彩色 LCD、标准 8232 串口、串口鼠标、Ethernet boot loader 等驱动程序。

关键词: Windows CE; 嵌入式; 可视电话; 驱动

中图分类号: TN949. 28

文献标识码: B

文章编号: 1004 373X(2006)08 142 03

Research and Implement of Vediohone Based on Windows CE

YIN Xiaocui, WANG Meng

(Dongying Vocation College, Dongying, 257091, China)

Abstract: The paper compares and analyses popular embedded operating system at present, for example, Windows CE, embedded Linux, VxWorks, Palms, and so on. The general scheme of media control board of videophone based on embedded microprocessor StrongARM and emebdeded operating system Windows CE is designed. It performs the customization of embedded operating system, designs and develops the programs of system start up, the driver of color LCD, standard RS232 serial port, serial mouse, Ethernet boot loader.

Keywords: Windows CE; embedded; videophone; driver

可视电话属于多媒体通信范畴, 是一种有着广泛应用领域的视讯会议系统, 使人们在通话时能够看到对方影像, 他不仅适用于家庭生活, 而且还可以广泛应用于各项商务活动、远程教学、保密监控、医院护理、医疗诊断、科学考察等不同行业的多种领域, 因而有着极为广阔的市场前景。

1 总体方案设计

1.1 主板硬件电路设计

系统选择 Intel 的 SA1110 嵌入式处理器作为该多媒体系统控制板的核心, 如图 1 所示。

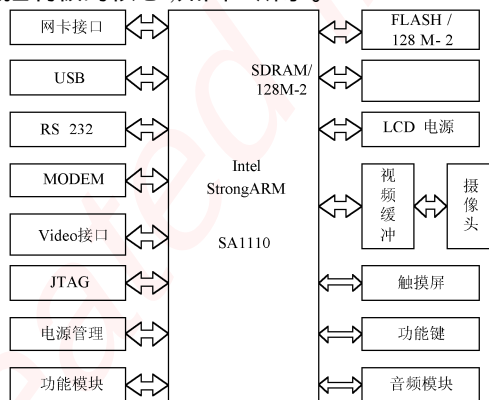


图 1 整体框图

1.2 操作系统及开发工具

由于 Microsoft 的嵌入式操作系统 Windows CE. NET 支持 Intel 的 SA1110 处理器, 且功能强大, 他是快速开发高端嵌入式系统的理想选择, 提供了强大的开发工具的支持, 平台和驱动的开发可以使用最新的 Microsoft 嵌入式开发工具 Microsoft Platform builder 4.1, 基于 Windows CE 特定平台的应用程序的开发采用 Microsoft 的应用开发工具 Microsoft Embedded Visual C++ 4.1 或 Microsoft Embedded Visual basic 4.1。

1.3 多媒体通信终端协议

可视电话的终端软件标准采用 ITU T 的基于 PSTN 的多媒体通信终端协议 H. 324, 可提供语音、图像和其他数据的实时通信。由于电话网带宽的限制, 决定了语音和图像都必须采用高压缩比的算法, 以低数码率进行传输。

2 系统板平台定制及模块驱动开发

Windows CE 作为商业化的嵌入式操作系统, 微软公司提供了强大的开发支持, 出版发行了用于开发基于 Windows CE 操作系统映像的集成开发环境 (IDE), 其目的是实现快速定制嵌入式设备上的基于 Windows CE 操作系统的映象。如 Windows CE 3.0, 还有最近推出的 Microsoft Windows CE Platform Builder 3.0 用于开发 Win-

Windows CE.NET。他们都包含最新版本的 Windows CE 操作系统、嵌入式开发工具和 IDE,支持 Microsoft 运行时库 (run time libraries),并提供了一些实例代码,集成开发环境能选择 Windows CE 组件,可以将定制的操作系统 (OS) 包括在内。

2.1 系统引导程序

由于 Windows CE 支持的 CPU 很多,大概有 30 多种,对于任何的嵌入式操作系统都有一个针对不同 CPU 和不同硬件的启动代码,这段程序是用汇编语言编写的,称为引导程序或启动代码,他主要负责将 CPU 初始化为系统内核已知的状态,由于大多数的 Windows CE 支持的 CPU 是嵌入式时,这些 CPU 都有一定数量的寄存器,必须设置这些寄存器以配置系统的速度,配置系统内存的基址地址等,启动程序将负责初始化任意高速缓存并确保系统处于未高速缓存的平面地址模式。

基于嵌入式 StrongARM 芯片在 Windows CE 下的引导程序主要完成 CPU 的时钟频率设置、SDRAM 初始化、MMU 初始化等,最后导入操作系统的内核。

2.2 串口驱动程序设计

该控制板提供 2 个标准的 RS 232 串口,分别使用 StrongARM SA1110 微处理器内置的 UART1 和 UART3 来实现。外围标准的 RS 232 电平转换采用 3 发 5 收的 MAX3244 芯片来实现。

Windows CE 下的串口驱动属于流驱动模型,是分层的驱动程序,分为 MDD 层和 PDD 层,他与操作系统接口为 DDI,(是通过 Device.exe 之间的接口),串口驱动的 DDI 函数为 COM Init,COM Open,COM Close,COM Read,COM Write,COM PowerDown,COM PowerUp,COM Deinit,COM IOControl。根据不同的硬件电路在板级支持包中实现串口驱动的 DDSI 接口函数,即为 PDD 层的函数,他和具体的硬件有关。

2.3 LCD 驱动程序设计

Windows CE 中显示驱动程序应归于内部驱动程序,因为驱动程序与图形、窗口和事件系统 (GWES) 之间存在定制接口。GWES 模块或设备管理器可以被其父进程直接加载和调用。Windows CE 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,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。Windows CE 显示器驱动程序使用 Microsoft 提供的一套称作图形原始引擎 (GPE) 的 C++ 类来实现,这些 GPE 类提供了基本的代码,可以通过这些代码导出自己硬件的显示器驱动程序。GPE 类处理显示 DDI 层的所有通信。GPE 类要求显示器硬件采用平铺框架的

缓冲区,也就是说,显示器的内存必须存在于连续的内存区域,使用非连续的框架缓冲区来修改 GPE 类将要付出巨大的努力。

如图 2 所示,驱动程序被 GEWS 直接调用,Windows CE 的显示器驱动程序采用分层结构,微软提供的 GPE 库处理所有缺省的画图和在显示驱动程序的模型设备驱动层 (MDD) 的动作,OEM 开发商根据不同的显示设备在 PDD 层编写特定硬件的代码。

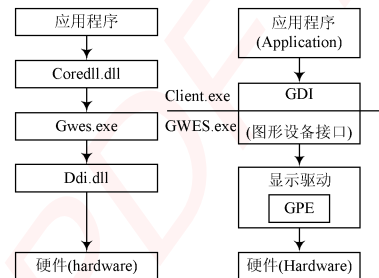


图 2 显示驱动程序的调用过程与体系结构

笔者使用最新的 RTOS 开发工具 Microsoft Platform builder 4.1,开发的 LCD 显示驱动程序是基于 StrongARM 处理器的,该处理器内置了 1 个 LCD 接口控制器,所采用的液晶显示器为 NL6448BC20 08,他是一款 Active Matrix 型 640 x480 TFT 彩色液晶显示器。根据具体 LCD 的电器特性,在板级支持包 BSP 驱动中通过设置 LCD 的控制寄存器和 DMA 通道的地址等实现对这些参数设置(如 LCD 屏大小、单屏还是双屏等),以及完成显示驱动接口 GDI 的各种功能函数,具体程序可参看板级支持包 BSP。下面做简要说明,注册表文件 Platform.reg 配置与显示驱动有关的信息:

```
[ HKEY_LOCAL_MACHINE\ Drivers\ Display\ SALCD2
\ CONFIG]
"DisplayDll" = "sa_lcd2.d11"
"LCDBaseVirtual" = dword:ab100000
"GPIOBaseVirtual" = dword:a9040000
"LCDVirtualFrameBase" = dword:ac008900
"LCDPhysicalFrameBase" = dword:c0008900
"bClearAlternateVideoBCR" = dword:1
[ HKEY_LOCAL_MACHINE\ System\ GDI\ Drivers]
"Display" = "sa_lcd2.d11"
```

在 sa_lcd2.cpp 文件中具体实现显示 PDD 层 DDSI 的有关接口函数。其中 SA_LCD2 主要实现屏、鼠标参数的初始化;InitializeHardware 函数读注册表配置信息、分配显示缓存、设置调色板、配置 LCD 控制寄存器等。以下以构造函数 SA_LCD2 和 InitializeHardware 函数简要说明。

```
staticlong gBitMasks[] = {0xF800,0x07E0,0x001F};
SA_LCD2::SA_LCD2(void)
{ ...
  m_nScreenWidth = 640;
  m_nScreenHeight = 480;
  ...}
void SA_LCD2::InitializeHardware(void)
{ ...
```

```
WRITE_BITFIELD(struct lccrOBits, &m LCDRegs >  
lccr0 ,len ,0) ;  
WRITE_BITFIELD(struct lccrOBits, &m LCDRegs >  
lccr0 ,cms ,0) ;  
...}
```

2.4 串口鼠标驱动设计

串口鼠标驱动程序是串口之上的一个应用级驱动程序,他主要在系统启动时监测鼠标,如果鼠标存在加载鼠标驱动,得到鼠标的相对偏移量和相应的事件,然后传递给操作系统。串口鼠标是直接接在标准串口上的传输设备,依靠串口的硬件流控信号的 RSR232 电平供电。

串口物理连线分别为 TD PC 发送、RD PC 接收、RTS、DTR 鼠标电源;设置波特率:1 200 b/s,字节位数:7 b,停止位:1 b。串口数据包由 3 个字节组成,第 1 字节的 D6 位为同步位。LB(D5)鼠标左键状态、RB(D4)鼠标右键状态(1 表示按下,0 表示抬起);第 2 字节为 y 轴偏移量;第 3 字节为 x 轴偏移量。

(1) 添加设备组件文件 Serialmouse.ccc

```
Component Type(  
Name(sermouse)  
GUID({72BA7897 E608 4026 9C8D 9492B399A40F})  
...)
```

(2) 配置注册表文件 Platform.reg

```
[HKEY_LOCAL_MACHINE\ Drivers \ BuiltIn \ Ser-  
Mouse]  
"Index" = dword:1  
"D11" = "SerMouse.dll"  
"Prefix" = "SRM"  
"Order" = dword:1  
"Port" = dword:2
```

(3) 配置板级支持文件 SAllXOBD.BSP

```
# ADD COMPONENT BY GUID TO ROOT  
"{72BA7897 E608 4026 9C8D 9492B399A40F}"
```

(4) 配置驱动加载文件 PLATFORM.BIB

```
IF PLAT NOSERIAL !  
serial.dll $( LFLATRELEASEDIR)\serial.dll  
Sermouse.dll $( FLATRELEASEDIR)\Sermouse.dll  
...  
ENDIF
```

2.5 Ehternet boot loader 设计

Eboot 是(Ethernet boot loader)开发包是专门为 SA 1110 开发板和 SA 1111 开发模块而设计的。他的主要目的是为软件开发人员提供一个快速和高效的传输 Windows CE 操作系统映像、驱动和相联系的应用到目标平台,通过使用 Microsoft Windows CE Platform (IDE) 或 Microsoft CShell。可使生成 debug 板的 CE 的操作系统

映像文件下载到开发板的 RAM 中,实现开发过程中的对 Windows CE 的内核级调。

3 系统的调试与测试

主要包括系统板电源模块、CPU 的边界扫描模块、存储器模块、CPLD 模块锁存器模块、串口通信模块扩展子卡缓冲驱动模块、CF 卡缓冲驱动模块、WIN CE 的下载和整个系统板功耗的测试。系统的硬件测试流程与一般嵌入式系统的测试流程是一样的,这里不做具体阐述。驱动的调试采用开发的 Eboot 和 Microsoft Platform builder 4.1 来进行,其步骤如下:

(1) 将 Eboot.nb0 烧写到 FLASH(从起始地址开始);

(2) 进行 PB 的通信设置,选择 PB 下的 configure remote connection。打开 services 页将 Download 和 Kernel 下拉框中分别设为 Internet,然后点击 configure..弹出对话框;

(3) 系统连线连接好,即将目标机 CF 卡和 PC 机端通过网线连接好;

(4) 复位目标板,当 PB 监测到目标设备名时,即表示通信设置完成;

(5) 单击 PB 菜单 Target 下的 Download/initialize 菜单即可,依照系统提示复位目标系统开始下载,并且提示下载的进度;

(6) 驱动程序中加入断点,如 LCD 或串口驱动程序中的适当位置;

(7) 利用 PB 编写 debug 版的系统影像;

(8) 进行调试测试。

4 结 语

电话作为人们日常生活、工作中不可或缺的通讯工具,以其方便、快捷等特点被广泛应用,但普通电话机只能是“只闻其声,不见其人”。随着人们对快捷、先进、实用通信方式的不断追求,可视电话无疑有着极大的发展潜力。在未来几年内,可视电话将会迈入寻常百姓家。

参 考 文 献

- [1] 欧建平,楼生强,皇甫堪.基于 DSP 的 H.324 可视电话研究与实现[J].电子技术应用,2002,28(2):58-60.
- [2] Goggin T A. Windows CE 高级开发指南[M]. 尤滔,张平,周晓权译.北京:电子工业出版社,2001.
- [3] 陈章龙,涂时亮.嵌入式系统 Intel StrongARM 结构与开发.北京:北京航空航天大学出版社,2001.

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)
30. [总线协议中的 CRC 及其在 SATA 通信技术中的应用](#)
31. [基于 FPGA 的 SATA 硬盘加解密控制器设计](#)
32. [Modbus 协议在串口通讯中的研究及应用](#)
33. [高可用的磁盘阵列 Cache 的设计和实现](#)
34. [RAID 阵列中高速 Cache 管理的优化](#)

35. [一种新的基于 RAID 的 CACHE 技术研究与实现](#)
36. [基于 PCIE-104 总线的高速数据接口设计](#)
37. [基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)
22. [基于 VxBus 的高速数据采集卡驱动程序开发](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)

7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)
24. [掌握 Android 7.0 新增特性 Quick Settings](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)
17. [Windows CE 环境下无线网卡的自动安装](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)
11. [基于 PowerPC 处理器 SMP 系统的 UBoot 移植](#)
12. [基于 PowerPC 双核处理器嵌入式系统 UBoot 移植](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)
15. [CortexA8 平台的 \$\mu\$ C-OS II 及 LwIP 协议栈的移植与实现](#)
16. [基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计](#)
17. [ARM S3C2440 Linux ADC 驱动](#)
18. [ARM S3C2440 Linux 触摸屏驱动](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
13. [基于 UEFI 固件的恶意代码防范技术研究](#)
14. [MIPS 架构计算机平台的支持固件研究](#)

Programming:

1. [计算机软件基础数据结构 - 算法](#)