

基于 ARM9 的 SMS 服务平台

SMS Service Platform Based on ARM9

李 程 黎福海 闫 旭 (湖南大学 电气与信息工程学院 湖南 长沙 410082)

摘 要: 介绍了一种将嵌入式平台与 GPRS 短消息数据传输模式相结合构建的无线数据传输系统, 系统以三星公司的 ARM9 系列产品 S3C2440 为核心, 配合 GPRS 模块实现了一个中文短消息的服务平台。文章简要介绍了系统硬件平台的搭建, 重点介绍了系统软件架构及开发环境, 特别详述了一种将纯 ASCII 字符信息与其他信息采用不同编码模式分别处理的方法。实际的测试表明该系统很好地满足了设计要求, 并且具有结构简单、功耗低、可扩展性强等特点。

关键词: ARM GPRS 短消息 Qtopia Core

1 引言

嵌入式系统与 GPRS 模块的结合, 使得非通用计算机系统的设备具有网络化和远程处理能力。随着科技的发展, 人类生活节奏的加快, 信息在生活中的地位日益重要, 如何方便快捷并及时有效地获取信息成为现代信息处理中的关键问题, 以 ARM 嵌入式控制平台和以 GPRS 无线传输网络为核心的系统得到了越来越广泛的应用。基于这些需求, 本文提出一种嵌入式平台上的能根据信息内容自动选择编码方式的短信息处理方案。

2 系统硬件架构

整个数据传输终端的硬件平台主要是由微处理器、存储器、UART 串行口、GPRS 模块、LCD 显示模块构成, 硬件结构框图如图 1 所示^[1]。

由于系统连接外设较多, 且没有大数据量的处理运算, 非常适合通用微处理器的发挥。综合考虑开发难度及处理器流行程度后, 处理核心选用了基于 ARM9 内核的 16 / 32 位嵌入式微处理器 S3C2440。与 DSP 等专用处理器相比, ARM 主要具有接口丰富, 开发方式灵活等特点。GPRS 模块选用的是华为公司的 GTM900B, 模块提供一整套 AT 命令接口, S3C2440 通过 UART

串口向 GPRS 模块发送 AT 指令来与 GPRS 模块进行通讯, 从而完成数据的接收和发送。

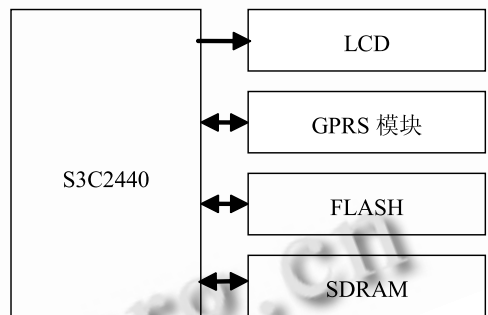


图 1 硬件结构框图

3 系统软件架构

系统整体控制部分比较繁杂, 故引进了操作系统进行管理。嵌入式 linux 系统是较为成熟的嵌入式操作系统, 由于有庞大的开源社区的支持, 基于 linux 的资源非常丰富, 便于上层应用程序的开发。系统应用层移植了 Qtopia Core 4.2.2 进行辅助开发。Qtopia Core 是一款基于嵌入式 Linux 的面向单一应用嵌入式产品的开发平台, 它支持一个不断扩展且超过 400 C++ 类的库, 封装了完整的用于端对端应用程序开发的基础结构。非常适合于嵌入式 GUI 的开发及其他一些处理。

系统软件紧密承接硬件，整个系统的软件架构如图 2：

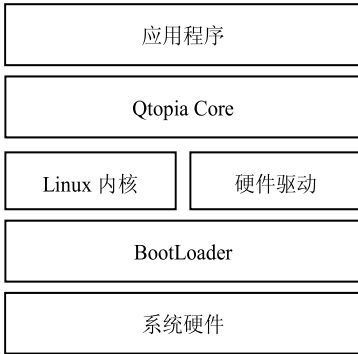


图 2 系统软件架构图

3.1 嵌入式系统的引导代码 Bootloader

Bootloader(引导加载程序)是嵌入式系统加电后运行的第一段代码。Bootloader 启动大多数分为两个阶段。第一阶段主要包含依赖于 CPU 的体系结构硬件初始化的代码，通常都用汇编语言来实现。这个阶段任务有：基本的硬件设备初始化(屏蔽所有的中断、关闭处理器内部指令/数据 Cache 等)、为第二阶段准备 RAM 空间、完成必要的硬件驱动、如果是从某个固态存储介质中启动，则复制 Bootloader 的第二阶段代码到 RAM 中、设置堆栈、转到第二阶段的 C 程序入口点；第二阶段通常用 C 语言完成。以便实现更复杂的功能。这个阶段任务有：初始化本阶段要使用到的硬件设备、检测系统内存映射、将内核映像和根文件系统映像从 Flash 读到 RAM、为内核设置启动参数、解压及调用内核。

3.2 置和编译 Linux 内核

该短消息平台系统使用 linux 操作系统。在装有 Linux 操作系统的宿主机上建立交叉编译工具 arm-linux-gcc 3.4.1，然后根据处理器架构、NAND Flash 分区信息、文件系统以及数据传输系统所使用的各个外部模块的信息对内核进行配置。系统移植的 Linux 系统版本为 2.6.22.9，修改必要的驱动后就可以完成交叉编译生成 zImage 内核映像了。这个映像最终会被下载到系统的 NAND Flash 存储器中^[2]。

3.3 载文件系统

文件系统是操作系统用于明确存储器上的文件的

方法和数据结构，即在存储器上组织文件的方法。该系统使用了两个文件系统。分别是 cramfs 和 yaffs2。cramfs 文件系统是一个只读的文件系统。一经烧入该文件系统内部的文件内容是不允许修改和保存。该文件系统的压缩率非常高，将一些不需更改的内容固化在 cramfs 文件系统中能有效的节省紧张的 Flash 空间。在确定根文件系统的内容后，输入

```
mkcramfs ./root
```

即可得到一个 root.cramfs 的文件。

在 cramfs 文件系统之后另外加载一个可读写 yaffs2 文件系统，方便用户进行应用程序的开发。后面编写的短消息收发程序以及图像化界面程序等应用程序都可以存放在这个文件系统中。

将各文件系统烧写入 Flash 后，设定 Linux 内核启动参数为：

```
noinitrd init=/rc root=/dev/mtdblock2
console=ttySAC0
```

其中 noinitrd 表示不使用 RAMDISK，init 指定启动脚本，root 指定根文件系统位置，console 指定终端控制台。其中最关键的是启动脚本 rc，该脚本需要执行 linux 系统交由用户控制前的最后工作，一般包括挂载文件系统、加载 udevd、启动 shell 等功能。本系统的启动脚本 rc 部分关键内容如下：

```
#!/bin/sh
#加载各伪文件系统
/bin/mount -n -t tmpfs tmpfs /dev/shm
.....
#启动 udevd
/sbin/udev -daemon
/sbin/udevstart
#运行 shell
```

```
exec /sbin/init
```

执行之后 linux 就正式交由用户控制了。

4 短消息的发送与接收

4.1 SMS 短信息简介

发送短消息常用 Text 和 PDU(Protocol Data Unit，协议数据单元)模式。使用 Text 模式收发短信

代码简单,实现起来十分容易,但最大的缺点是不能收发中文短信;而 PDU 模式不仅支持中文短信,也能发送英文短信。在 PDU Mode 中,可以采用三种编码方式来对发送的内容进行编码,它们是 7-bit、8-bit 和 UCS2 编码。7-bit 编码用于发送普通的 ASCII 字符,它将一串 7-bit 的字符(最高位为 0)编码成 8-bit 的数据,每 8 个字符可“压缩”成 7 个;8-bit 编码通常用于发送数据消息,比如图片和铃声等;而 UCS2 编码用于发送 Unicode 字符。PDU 串的用户信息(TP-UD)段最大容量是 140 字节,所以在这三种编码方式下,可以发送的短消息的最大字符数分别是 160、140 和 70^[3]。

4.2 AT 命令接收和发送

GPRS 模块与 S3C2440 的 UART1 进行通讯,通过该串口可以发送 AT 命令到 GPRS 模块并获取 GPRS 模块的输入。通过如下函数发送 cmdstring 中的 AT 命令到 GPRS 模块中:

```
int gsm_send_data (const string &cmd,
char end_char);
```

其中 cmd 为所要发送的 AT 指令, end_char 为指令终止字符,一般为回车,但有一些特殊指令需要以特殊的字符来终止。

系统开辟一个线程用于监视 GSM 模块的返回信息,从中提取所需要的信息进行处理。

4.3 短信息相关 AT 指令

系统启动之后首先需要进行一些初始化的工作,具体包括获取 SIM 卡状态,查询注册信息,获取运营商信息等。等待系统启动完成之后就可以进行短消息的收发工作了。

1) 选择短消息模式

发送 AT+CMGF= n 。 $n=0$ 时,选择 PDU 模式; $n=1$ 时,选择文本模式。执行命令后,模块返回 OK。本文选取的是 PDU 模式,该模式支持各种类型的短消息。

2) 设置短消息中心号码

发送 AT+CSCA=<string>, string 为当地短信息中心号码。执行命令后,模块返回 OK。

3) 发送短消息

在 PDU 模式下,发送 AT+CMGS= n 。 n 为短消

息 PDU 数据包的字符数。7-bit 编码时,指原始短消息的字符个数,而不是编码后的字节数。8-bit 编码时,就是字节数。UCS2 编码时,也是字节数,等于原始短消息的字符数的两倍。等模块返回>符号后,发送短消息的内容以 Ctrl+Z 结束。模块开始发送短消息,如果发送成功,则模块返回:

```
+CMGS: <mr>[, ackpdu]OK;
```

否则,模块返回:

```
ERROR
```

4) 接收短消息

发送 AT+CMGR=<index>。在 PDU 模式下模块会返回收到的短消息资料^[4]。

4.4 PDU 编码

PDU 模式发送短消息的关键在于编码。一般的 PDU 编码由 13 项组成,具体包括:

- A. 短信息中心地址长度, 2 位十六进制数(1 字节)。
- B. 短信息中心号码类型, 2 位十六进制数。
- C. 短信息中心号码, B+C 的长度将由 A 中的数据决定。
- D. 文件头字节, 2 位十六进制数。
- E. 信息类型, 2 位十六进制数。
- F. 被叫号码长度, 2 位十六进制数。
- G. 被叫号码类型, 2 位十六进制数, 取值同 B。
- H. 被叫号码, 长度由 F 中的数据决定。
- I. 协议标识, 2 位十六进制数。
- J. 数据编码方案, 2 位十六进制数。
- K. 有效期, 2 位十六进制数。
- L. 用户数据长度, 2 位十六进制数。
- M. 用户数据。

其中, A、B、C、D、E、I、K 在发送前都可以固定下来, F、G、H 由被叫号码决定, J 取决于编码方式, L、M 取决于具体信息。故 PDU 编码程序主要需要完成的是 M 部分。

系统通过调用如下函数发送短消息:

```
int gsm_send_msg (const string &d_id,
const string &msg);
```

其中 d_no 为目标号码, msg 为具体消息内容。

根据 d_no 可以获得 F、G、H 部分,其中目标号码类型取决于目标号码是手机还是小灵通。虽然文字信息都可以采用 UCS2 的方式进行编码,但是根据 PDU 编码方式的介绍,采用 7-bit 模式编码能发送更多的英文信息。故为了尽可能的压缩信息,该函数对 msg 的内容进行了检查判断。

由于 ASCII 编码的值为 0-127,而中文采用 2 个字节才能代表 1 个字符,且其中第一个字节的值大于 127。根据这一点程序可以判断字符串中是否包含中文。如果 msg 的所有字节的值均在 0-127 之间,则为纯 ASCII 字符,可采用 7-bit 编码,发送 160 个字符;若之中至少有 1 个字节的值大于 127,则 msg 中包含其他字符,必须采用 UCS2 编码,此时只能发送 70 个字符。部分判断程序如下:

```
while(i!=msg.size()) {
    if((unsigned short)msg[i]>127) {
        //字符为中文,直接退出循环
        break;
    } else {
        //字符为英文,继续检测
        i++;
    }
}
```

1) 7-bit 模式编码

PDU 模式的纯英文短信息编码使用 GSM 字符集的 7 位编码,首先将各个字符转换为 7 位的标准二进制 ASCII 码,然后将后面字符的低位逐位调整到前面,补齐前面的差别。

7-bit 编码采用的是 GSM 技术规范推荐的算法,程序调用如下函数进行编码:

```
int gsmEncode7bit(const char* pSrc,
unsigned char* pDst, int nSrcLength)
```

其中 pSrc 为源字符串指针, pDst 为目标编码串指针, nSrcLength 为源字符串长度,返回的是目标编码串长度。部分编码程序代码如下:

```
if(nChar == 0) {
    // 组内第一个字节,只是保存起来
    nLeft = *pSrc;
```

```
} else {
    // 组内其它字节,将其右边部分与残余数据
    相加,得到一个目标编码字节
    *pDst = (*pSrc << (8-nChar)) | nLeft;
    // 将该字节剩下的左边部分,作为残余数据
    nLeft = *pSrc >> nChar;
    // 修改目标串的指针和计数值 pDst++;
    nDst++;
}
```

2) UCS2 编码

发送中文信息时,必须采用 UCS2 编码模式。此时需要发送的是字符的 Unicode 编码。

由于移植了 Qtopia core,获得字符的 Unicode 编码只需调用一个非常简单的接口 unicode(),故这部分的编码只需将每个字符的 Unicode 表示连接起来即可。具体编码程序代码如下(使用了 Qtopia 类库)^[5]:

```
for (i=0; i<length; i++) {
    //str 为需编码的字符串
    hex = str.at(i).unicode();
    cStr += QString::number (hex, 16);
}
```

短消息发送流程图如图 3 所示:

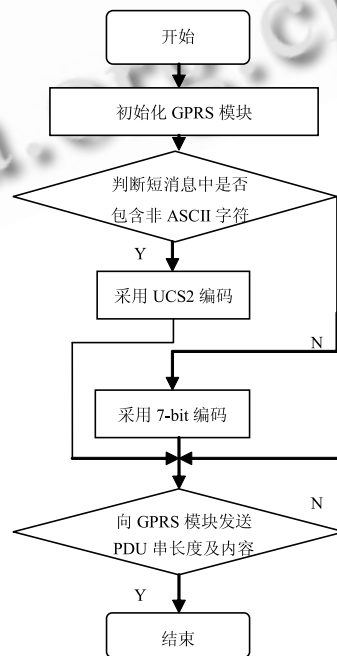


图 3 短消息发送流程图

(下转第 204 页)

(上接第 208 页)

接收到短消息的时候,将收到的 PDU 串经过与编码类似的解码过程,则可以得到发送者信息及具体消息内容。接收流程图如图 4 所示:

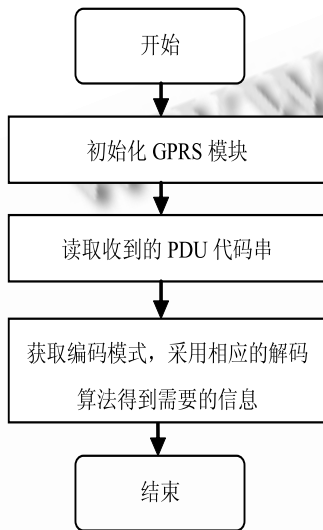


图 4 短消息接收流程图

5 结语

本文设计了一种基于 S3C2440 芯片和 GPRS 模块的短消息数据传输系统。该系统能根据发送内容的不同,选择不同的编码模式,以实现数据压缩的最大化。该系统的扩展性,移植性好,已用于一个基于 ARM 的烟草销售系统,并且可以很方便的移植到其他系统以适应不同领域的需要。

参考文献

- 1 S3C2440A 32-bit CMOS Microcontroller user's manual Revision 1.
- 2 雅墨.构建嵌入式 Linux 系统.北京:中国电力出版社.
- 3 GSM Technical Specification Version 5.3.0.
- 4 华为 GTM900 无线模块 AT 命令手册.
- 5 C++GUI Programming with Qt 4. Prentice Hall PTR.