

文章编号:1671-6833(2006)03-0089-04

# 基于 AR 和 uCLinux 的 CAN 接口电路的设计

韦宝泉

(华东交通大学交通信息工程及控制研究所,江西 南昌 330013)

**摘要:**介绍了一种新型 IED 上的 ARM 与 CAN 接口电路的设计方法.在开发一种基于 ARM 微处理器和 uCLinux 操作系统的新型 IED 时,考虑到变电站自动化系统的现场级通信要求具有较高的可靠性和实时性,选择了 CAN 现场总线.由于 CAN 收发器 SJA1000 是地址和数据分时复用芯片,而 ARM 的地址线 and 数据线是分开的,所以采用了软件模拟读写时序的方法,利用 ARM 的 I/O 引脚产生 SJA1000 的读写信号,设计了 ARM 与 CAN 的硬件接口电路,编写了 CAN 在 uCLinux 下的驱动程序,并调试通过.该接口的设计为新型智能化 IED 的通信打好了软硬件基础,能更好的适应变电站自动化系统的需求.

**关键词:** CAN; ARM; uCLinux

**中图分类号:** TP 302 **文献标识码:** A

## 0 引言

CAN 总线是控制局域网络(Control Area Network)的简称,属于现场总线的范畴.它基于优先级的多主节点结构,具有较强的纠错能力,克服了 RS-232 和 RS-485 的缺点,具有突出的可靠性、实时性和灵活性.而与 Lon Work 总线相比又有系统结构简单、实用性强等特点.另外,与其它现场总线比较,CAN 总线具有通信速率高、容易实现、性价比高等诸多优点,且有效支持分布式控制系统,是变电站自动化系统的一种性价比最优的选择.

变电站自动化系统的发展,对其智能间隔装置 IED(Intelligent Electronic Device)提出了越来越高的要求.微处理器技术、操作系统及通信技术的发展,则为新型间隔装置的研制提供了软硬件条件.我们在设计一种新型 IED 时,选择了先进的 ARM 微处理器和 uCLinux 操作系统,而对于通信总线的选择,分析如下:

变电站自动化系统的通信包括两种:一种是变电站内部的现场级通信,另外一种是与上级调度的通信.变电站的现场级通信是变电站内部各 IED 之间以及它们与监控主机之间的数据通讯与信息交换,由于工作环境比较恶劣,变电站内部电磁干扰比较强,并且 IED 本身对实时性、可靠

性要求较高,所以选择了 CAN 现场总线.图 1 为采用 CAN 现场总线的分布式变电站自动化系统结构图.

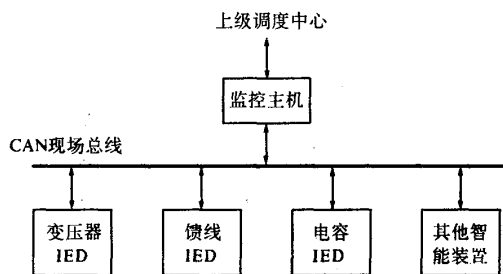


图 1 采用 CAN 现场总线的分布式变电站自动化系统结构图

Fig.1 The distributed SAS with CANbus

图中的各 IED 采用 ARM 处理器,安装 uCLinux 操作系统,根据需要扩展了各接口电路.其中,由于现场级通信采用了 CAN 总线通信方式,因此必须设计好 IED 上 CAN 与 ARM 的接口电路,它对 IED 之间的通信以及整个变电站自动化系统的可靠运转起着至关重要的作用.

## 1 CAN 总线接口硬件电路设计

### 1.1 CAN 总线接口功能

系统通过 CAN 总线与站控层通信,CAN 通信的硬件接口电路包括 CAN 通信控制器与微处理

收稿日期:2006-01-12;修订日期:2006-02-28

基金项目:江西省教育厅科研项目(赣教技字[2005]123号)

作者简介:韦宝泉(1979-),男,山东曲阜人,华东交通大学讲师,硕士,主要从事变电站自动化、微机远动系统、嵌入式系统等方面的研究和开发工作.

器之间和 CAN 总线收发器与物理总线之间的接口电路的设计。其中, CAN 通信控制器是 CAN 总线接口电路的核心, 主要完成 CAN 的通信协议, 而 CAN 总线收发器的优点是通信距离较大, 系统的瞬间抗干扰能力较强, 能保护总线, 降低射频干扰(RFI), 实现热防护等<sup>[1]</sup>。

CAN 总线接口电路的主要功能是: 通过 CAN 总线接收来自上位机的数据进行分析组态然后下传给下位机的控制电路实现控制功能; 当 CAN 总线接口接收到下位机的上传数据, SJA1000 就产生一个中断, 引发微处理器产生中断, 通过中断处理程序处理每一帧信息并通过 CAN 总线上传给上位机进行分析。而 S3C4510B 是 CAN 总线接口电路的核心, 承担 CAN 控制器的初始化、CAN 的收发控制等任务。

### 1.2 CAN 总线器件选择

目前广泛流行的 CAN 总线器件有两大类: 一类是独立的 CAN 控制器, 如 82C200、SJA1000 及 Intel82526/82527 等, 另一类是带有片内 CAN 的微控制器, 如 P8XC582 及 16 位微控制器 87C196CA/CB 等。目前国内很多设计 ARM 上 CAN 接口电路都是采用的内部含有 CAN 控制器的 ARM 芯片, 外围只需要连接收发器, 如参考文献[2]。而本设计所选用的 ARM 处理器 S3C4510B 本身没有片内 CAN 控制器, 所以选择独立的控制器: PHILIPS 公司的 SJA1000 CAN 控制器以及 82C250 总线收发器。

### 1.3 CAN 接口电路设计

因为 SJA1000 是地址和数据分时复用芯片,

而 ARM 的地址线和数据线是分开的, 所以不能将 SJA1000 和 ARM 直接接口。实际设计中有两种方法可以采用。

第一种方法, 通过 ARM 的 nECS[3:0] (访问外部 I/O 片选信号) 和输出使能信号 nOE、写使能信号 nWE 通过逻辑电路组合产生 ALE 信号, 完成对数据/地址分时复用芯片的读写。有不少文献中采用了这种方法, 其缺点是操作较困难, 可靠性也不太高;

第二种方法, 利用 ARM 的 I/O 引脚产生读写信号, 即用软件模拟读写时序, 这种方法实现起来简单可靠, 但要求有足够的 I/O, 占用资源多。

本系统 ARM 的 I/O 口可以满足要求, 所以这种方法实现比较容易。接口电路设计如图 2 所示。

- SJA1000 的 AD0—AD7 接 ARM 的 P0—P7, 通过 ARM 的 I/O 口和 SJA 交换数据;
- SJA1000 的 5 脚: 读允许输入端( $\overline{RD}$ )接 ARM 的 P12;
- SJA1000 的 6 脚: 写允许端( $\overline{WR}$ )接 ARM 的 P13;
- SJA1000 的 3 脚: 地址选通输入脚(ALE)接 ARM 的 P14;
- 第一片 SJA1000 的片选端  $\overline{CS}$  接 ARM 的 P15, 第二片 SJA1000 的片选端  $\overline{CS}$  接 ARM 的 P16;
- 第一片 SJA1000 的中断信号  $\overline{INT}$  脚接 ARM 的 XINTREQ0 (P8); 第二片 SJA1000 的  $\overline{INT}$  脚接 DSP 的 XINTREQ1 (P9)。

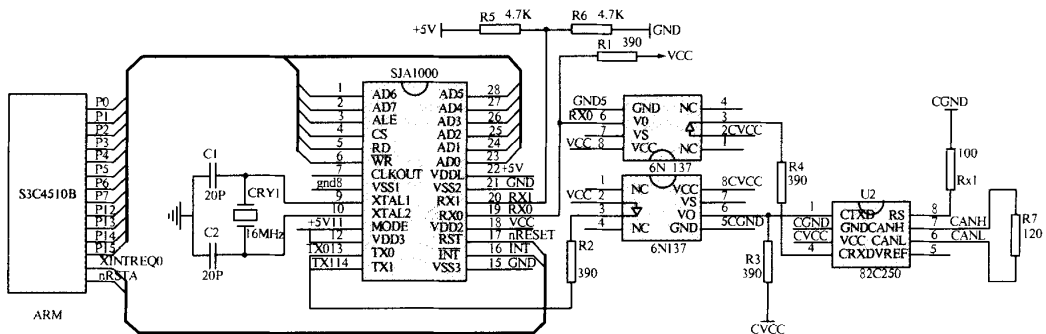


图 2 CAN 总线接口电路原理图

Fig.2 CANBUS interface circuit

这里利用 ARM 的 I/O 引脚产生读写信号, 软件中产生读写时序。SJA1000 的  $\overline{WR}$ 、 $\overline{RD}$ 、 $\overline{CS}$  和 ALE 脚都是接 ARM 的 I/O 脚, 通过编程控制 ARM 的 I/O 脚来送出对 SJA1000 的读、写时序。可以编制读写 SJA1000 的函数进行封装, 以后只需调用这两个读、写函数就可以了。

另外 SJA1000 的中断信号输出连到 S3C4510B 的外部中断信号输入管脚; 复位管脚与 S3C4510B 的复位信号相接;

为了增强 CAN 总线模块抗干扰能力, SJA1000 的 TX0 和 RX0 与 82C250 的 TXD 和 RXD 之间加了高速光耦 6N137, 实现了总线上各 CAN

模块之间的电气隔离.光耦两侧采用的电源 VCC 和 CVCC 必须隔离,因此采用了定电压输入隔离非稳压单输出型 DC-DC 隔离电源模块 B0505S,提高了模块的稳定性和安全性.

## 2 CAN 驱动程序的设计

### 2.1 工作流程

本系统中 CAN 工作过程设计思想如下:

发送数据时:用户进程通过系统调用向驱动程序传送一帧任意长度的数据;驱动程序中发送数据的程序先判断直接发送还是放入发送队列:正在发送报文或发送队列有未发的数据,则将数据放入发送队列,否则直接写入发送缓冲区发送.

接收数据时:CAN 接口接收到数据,产生接收中断,启动接收中断处理程序,将 CAN 控制器中接收缓冲区中的内容复制到接收帧队列,由用户通过系统调用从接收帧队列中读取完整的一帧.

对于数据的组织和处理,考虑到有很多种类型的数据,通信协议由驱动程序实现过于烦琐,并且不灵活,所以采用由用户程序实现的方式:数据的格式、组包都由用户程序完成,组成每帧 10 个字节的数据,与驱动之间每次传输一帧数据,驱动程序只需要对一帧的数据进行收、发,而不用理解其具体含义,实现起来非常简洁.

驱动程序的发送数据流程设计如图 3 所示.

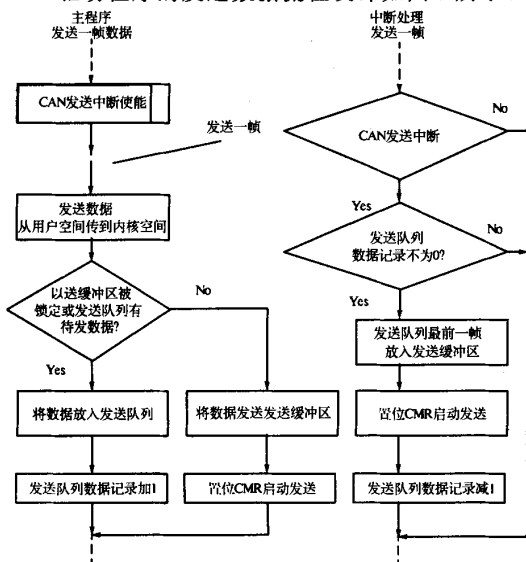


图3 发送一个报文流程图(中断控制)

Fig.3 Flow Chat of Send a Message (Interrupt Control)

图3建立了一个FIFO形式的发送队列(发送数据暂存区),有多个发送数据时,等待发送的数万方数据

据先进入该队列等候.当一个发送结束,产生发送中断后,先判断该队列有无数据,如果有数据继续发送,最先进来的先发,每发送完一帧该队列的数据记录值减一.

接收数据流程也建立了一个FIFO形式的接收队列(接收数据暂存区),接收中断接收的数据依次放入该队列.用户读取一帧时,先读最先接收的.每读一次,接收队列数据记录减一.流程图略.

### 2.2 驱动程序

整个CAN驱动程序结构如下(CAN接口属于字符型设备):

#### 2.2.1 初始化函数

```

int CAN_init(void) /* 设备初始化函数 */
{
    /* CAN 寄存器配置(略) */

    int k, result;

    result = register_chrdev(254, "CAN",
    &CANfops); /* 注册设备 */

    k = request_irq(CAN_irq, CAN_interrupt, SA_INTERRUPT, "CAN", NULL);

    if (k < 0) {printk("CAN-irq can't get assigned\n", CAN_irq);}

    return 0;
}
  
```

#### 2.2.2 中断处理程序

CAN口数据的接收、发送采用中断方式.中断处理分为上半部分和下半部分.上半部分由硬件触发,工作于关中断状态下,要求处理尽可能快速;下半部分是一段单独的处理程序,一般挂入立即队列以便快速执行,它处于开中断的安全状态下,适合处理耗时的任务<sup>[3]</sup>.对于我们的CAN接口,由于采用了协议由用户程序实现的方式,驱动程序的功能只是简单的收发数据,并不耗时,所以没有必要再编写下半部.中断程序编写结构略.

#### 2.2.3 入口函数

CAN口驱动的入口操作集包括打开、关闭文件操作函数和读、写操作函数,如下:

```

struct file_operations CANfops = {
    read: CAN_read,
    write: CAN_write,
    open: CAN_open,
    release: CAN_release
};
  
```

### 2.3 驱动程序编译进内核

由于uClinux不支持动态加载,所以须将CAN

驱动静态编译进内核,步骤如下:

(1) 首先将驱动程序源代码 CAN.c 放到宿主机的 uClinux/linux/driver/char 目录下;

(2) 然后在 uClinux 的字符设备初始化函数 chr-dev-init()(linux/drive/char/mem.c)中加入编写的 CAN 口初始化函数 CAN-init();

(3) 在 uClinux/linux/driver/char 目录下 Makefile 中增加: L-OBJS += CAN. 这样就将 CAN 的设备驱动编译进内核.

(4) 重新编译内核并启动,在 ARM 板的/dev 目录下即可看到 CAN 设备.

### 3 结束语

本设计基于 ARM 和 uClinux 操作系统,设计了 ARM 和 CAN 的接口电路及其驱动程序.由于该接口是变电站自动化系统 IED 装置的接口,是

现场级通信的关键所在,所以必须保证其可靠性和实时性.采用软件模拟读写时序的方法,使得连接简单可靠、工作速度快,满足了各项要求.目前该装置已经进入实验阶段,本接口电路已经能够正常可靠的工作,达到了现场级通信要求,对这种新型 IED 在变电站自动化系统中的应用起到了促进作用.

### 参考文献:

- [1] 罗雪梅. 基于 SJA1000 的 CAN 总线接口电路的设计与实现[J]. 贵州工业大学学报, 2003, 32(4): 25 ~ 26.
- [2] 高红玉, 徐建城, 曾成奇. 基于 arm 的 can 总线智能节点的设计[J]. 电子技术应用. 2005, (4): 24 ~ 26.
- [3] ALESSANDRO R, LISOLEG. LINUX 设备驱动程序[M]. 北京: 中国电力出版社, 2001. 102 ~ 103.
- [4] 黄俊杰, 黄云峰. AVR 单片机实现光电隔离 RS-422/485 智能接口研究[J]. 郑州大学学报(工学版), 2004, 25(1): 85 ~ 87.

## A CAN Interface Design Based on ARM and uClinux

WEI Bao - quan

(Institute of Transportation Info - engineering and Control, East China Jiaotong University, Nanchang 330013, China)

**Abstract:** In this paper, the interface circuit of ARM and CAN is introduced in developing a new IED. Considering that the local communication of SAS requires high reliability and real-time performance, CAN bus is adopted in developing a new IED based on ARM and uClinux. The data bus and address bus of SJA1000 are time-sharing, while those of ARM are parted. So a new method which generates temporal logic by software is adopted. The read-write signal is generated by I/O of ARM. Besides the interface circuit of ARM and CAN, the driver of CANbus in uClinux is also designed and debugged successfully. This interface paves way for the new IED's communication in software and hardware. Then the IED will be more applicable to substation automation systems.

**Key words:** CAN; ARM; uClinux