CAN 总线系统智能节点设计

华东地质学院信息工程系 邹继军 饶运涛

摘要：CAN 总线上的节点是网络上的信息接收和发送站；智能节点能通过编程设置工作方式、ID 地址、波特率等参数。它主要由单片机和可编程的 CAN 通信控制器组成。本文介绍这类节点的硬件设计和软件设计；其中软件设计包括 SJA1000 的初始化、发送和接收等应用中的最基本的模块子程序。

关键词：总线 节点 CAN 控制器

引言

CAN (Controller Area Network) 总线，又称控制器局域网，是 Bosch 公司 在现代汽车技术中领先推出的一种多主机局部网，由于其卓越的性能，极高的可靠性，独特灵活的设计和低廉的价格，现已广泛应用于工业现场控制、智能大厦、小区安防、交通工具、医疗仪器、环境监控等众多领域。CAN 已被公认为几种最有前途的现场总线之一。CAN 总线规范已被 ISO 国际标准组织制订为国际标准，CAN 协议也是建立在国际标准组织的开放系统互连参考模型基础上的，主要工作在数据链路层和物理层。用户可在其基础上开发适合系统实际需要的应用层通信协议，但由于 CAN 总线极高的可靠性，从而使应用层通信协议得以大大简化。

CAN 总线与其它几种现场总线比较而言，是最容易实现、价格最为低廉的的一种，但其性能并不比其它现场总线差。这也是目前 CAN 总线在众多领域被广泛采用的原因。节点是网络上信息的接收和发送站，所谓智能节点是由微处理器和可编程的 CAN 控制芯片组成，它们有两者合二为一的，如芯片 P8XC592；也有如本文介绍的，独立的通信控制芯片与单片机接口，后者的优点是比较灵活。当然，也有不要微处理器的节点。下面以 CAN 通信控制器 SJA1000 为例，对 CAN 总线系统智能节点硬件和软件设计作一个全面的介绍。

一、 CAN 通信控制器 SJA1000 功能简介

CAN的通信协议主要由CAN控制器完成。CAN控制器主要由实现CAN总线协议的部分和实现与微处理器接口部分的电路组成。对于不同型号的CAN总线通信控制器，实现CAN协议部分电路的结构和功能大多相同，而与微处理器接口部分的结构和方式存在一些差异。这里主要以SJA1000为代表对CAN控制器的功能作一个简单介绍。

SJA1000是一种独立CAN控制器，它是PHILIPS公司的PCA82C200 CAN控制器的替代产品。SJA1000具有BasicCAN和PeliCAN两种工作方式 , PeliCAN工作方式支持具有很多新特性的CAN 2.0B协议。

SJA1000在软件和引脚上都是与它的前一款PCA82C200 独立CAN控制器兼容的（SJA1000引脚功能如表1所示），在此基础上增加了很多新的功能。为了实现软件兼容，SJA1000 采用了两种工作方式： BasicCAN方式（PCA82C200兼容方式）， PeliCAN 方式（扩展特性方式）。工作方式通过时钟分频寄存器中的CAN方式位来选择。上电复位默认工作方式是BasicCAN方式。BasicCAN和PeliCAN方式的区别如下。

在PeliCAN 方式下，SJA1000有一个重新设计的含很多新功能的寄存器组。SJA1000包含PCA82C200中的所有位，同时增加了一些新的功能位。PeliCAN 方式支持CAN2.0B协议规定的所有功能（29位的标识符）。

SJA1000 的主要新功能如下：

- 标准结构和扩展结构报文的接收和发送
- 64字节的接收FIFO
- 标准和扩展帧格式都具有单/双接收滤波器（含接收屏蔽和接收码寄存器）
- 可进行读/写访问的错误计数器
- 可编程的错误报警限制
- 最近一次的错误代码寄存器

- 每一个CAN总线错误都可以产生错误中断
- 具有丢失仲裁定位功能的丢失仲裁中断
- 单发方式（当发生错误或丢失仲裁时不重发）
- 只听方式（监听CAN总线，无应答，无错误标志）
- 支持热插拔（无干扰软件驱动位速率检测）
- 硬件禁止CLKOUT输出

表 1 SJA1000 引脚功能

符 号	引 脚	功 能
AD0～AD7	2,1,28～23	地址/数据复用总线
ALE	3	ALE 信号（INTEL 方式）或 AS 信号（Motorola 方式）
/CS	4	片选输入，低电平允许访问 SJA1000
/RD	5	微控制器的读信号（Intel 方式）或 E 信号（Motorola 式）
/WR	6	微控制器的写信号（Intel 方式）或读写信号（Motorola 方式）
CLKOUT	7	SJA1000 产生的提供给微控制器的时钟输出信号，此信号由内部振荡器经可编程分频器得到。可编程禁止该引脚
VSS1	8	逻辑电路地
XTAL1	9	振荡放大器输入，外部振荡放大器信号经此引脚输入
XTAL2	10	振荡放大器输出，使用外部振荡信号时此引脚必须开路
MODE	11	方式选择输入端：1=Intel 方式，0=Motorola 方式
VDD3	12	输出驱动器 5V 电源
TX0	13	由输出驱动器 0 至物理总线的输出端
TX1	14	由输出驱动器 1 至物理总线的输出端
VSS3	15	输出驱动器地
/INT	16	中断输出端，用于向微控制器提供中断信号
/RST	17	复位输入端，用于重新启动 CAN 接口（低电平有效）
VDD2	18	输入比较器 5V 电源
RX0，RX1	19，20	由物理总线至SJA1000输入比较器的输入端。显性电平将唤醒处于睡眠方式的SJA1000。当RX0高于RX1时，读出为隐性电平，否则为显性电平
VSS2	21	输入比较器地
VDD1	22	逻辑电路 5V 电源

二、CAN 总线系统智能节点硬件电路设计

本文中所设计的 CAN 总线系统智能节点，采用 89C51 作为节点的微处理器，在 CAN 总线通信接口中，采用 PHILIPS 公司的 SJA1000 和 82C250 芯片。SJA1000 是独立 CAN 通信控制器，82C250 为高性能 CAN 总线收发器。

如图 1 所示为 CAN 总线系统智能节点硬件电路原理图。从图中可以看出，电路主要由四部分所构成：微控制器 89C51、独立 CAN 通信控制器 SJA1000、CAN 总线收发器 82C250 和高速光电耦合器 6N137。微处理器 89C51 负责 SJA1000 的初始化，通过控制 SJA1000 实现数据的接收和发送等通信任务。

SJA1000 的 AD0～AD7 连接到 89C51 的 P0 口，$\overline{CS}$ 连接到 89C51 的 P2.0，P2.0 为 0 的 CPU 片外存贮器地址可选中 SJA1000，CPU 通过这些地址可对 SJA1000 执行相应的读写操作。SJA1000 的 $\overline{RD}$、$\overline{WR}$、ALE 分别与 89C51 的对应引脚相连，$\overline{INT}$ 接 89C51 的 $\overline{INT0}$，89C51 也可通过中断方式访问 SJA1000。

为了增强 CAN 总线节点的抗干扰能力，SJA1000 的 TX0 和 RXO 并不是直接与 82C250 的 TXD 和 RXD 相连，而是通过高速光耦 6N137 后与 82C250 相连，这样就很好的实现了总线上各 CAN 节点间的电气隔离。不过，应该特别说明的一点是光耦部分电路所采用的两个电源 VCC 和 VDD 必须完全隔离，否则采用光耦也就失去了意义。电源的完全隔离可采用小功率电源隔离模块或带多 5V 隔离输出的开关电源模块实现。这些部分虽然增加了节点的复杂，但是却提高了节点的稳定性和安全性。

82C250 与 CAN 总线的接口部分也采用了一定的安全和抗干扰措施。82C250 的 CANH 和 CANL 引脚各自通过一个 5Ω的电阻与 CAN 总线相连，电阻可起到一定的限流作用，保护 82C250 免受过流的冲击。CANH 和

CANL 与地之间并联了两个 30P 的小电容，可以起到滤除总线上的高频干扰和一定的防电磁辐射的能力。另外在两根 CAN 总线接入端与地之间分别反接了一个保护二极管，当 CAN 总线有较高的负电压时，通过二极管的短路可起到一定的过压保护作用。82C250 的 Rs 脚上接有一个斜率电阻，电阻大小可根据总线通讯速度适当调整，一般在 16K～140K 之间。

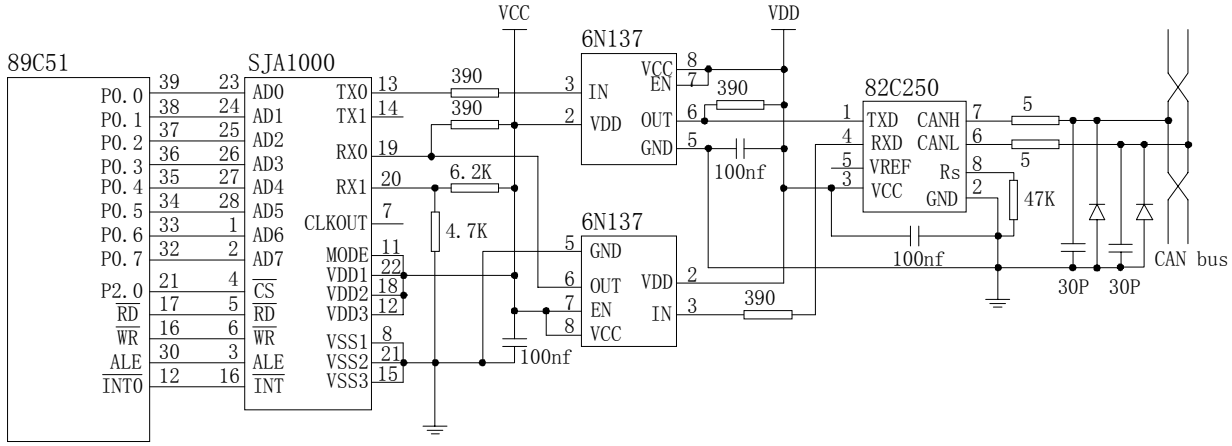

图 1　CAN 总线系统智能节点硬件电路原理图

三、　CAN 总线系统智能节点软件设计

CAN 总线节点的软件设计主要包括三大部分：CAN 节点初始化、报文发送和报文接收。熟悉这三部分程序的设计，就能编写出利用 CAN 总线进行通信的一般应用程序。当然要将 CAN 总线应用于通信任务比较复杂的系统中，还需详细了解有关 CAN 总线错误处理、总线脱离处理、接收滤波处理、波特率参数设置和自动检测以及 CAN 总线通信距离和节点数的计算等方面的内容。下面仅就前面提到的三部分程序的设计作一个描述，以供大家在实际应用中参考。

1、初始化子程序

SJA1000 的初始化只有在复位模式下才可以进行，初始化主要包括工作方式的设置、接收滤波方式的设置、接收屏蔽寄存器（AMR）和接收代码寄存器（ACR）的设置、波特率参数设置和中断允许寄存器（IER）的设置等。在完成 SJA1000 的初始化设置以后，SJA1000 就可以回到工作状态，进行正常的通信任务。下面提供了 SJA1000 初始化的 51 汇编源程序，程序中寄存器符号表示的是 SJA1000 相应寄存器占用的片外存贮器地址，这些符号可在程序的头部用伪指令 EQU 进行定义。后文对这一点不再作特别说明。

CANINI：

```
MOV     DPTR，#MOD          ；方式寄存器
MOV     A，#09H             ；进入复位模式，对 SJA1000 进行初始化.
MOVX    @DPTR，A
MOV     DPTR，#CDR          ；时钟分频寄存器
MOV     A，#88H             ；选择 PeliCAN 模式，关闭时钟输出（CLKOUT）
MOVX    @DPTR，A
MOV     DPTR，#IER          ；中断允许寄存器
MOV     A，#0DH             ；开放发送中断、超载中断和错误警告中断
MOVX    @DPTR，A
MOV     DPTR，#AMR          ；接收屏蔽寄存器
MOV     R6，#4
MOV     R0，#DAMR           ；接收屏蔽寄存器内容在片内 RAM 中的首址
```

```
AMR：   MOV     A，@R0
        MOVX    @DPTR，A        ；接收屏蔽寄存器赋初值
        INC     DPTR
        DJNZ    R6，AMR
        MOV     DPTR，#ACR      ；接收代码寄存器
        MOV     R6，#4
        MOV     R0，#DACR       ；接收代码寄存器内容在片内 RAM 中的首址
ACR：   MOV     A，@R0
        MOVX    @DPTR，A        ；接收代码寄存器赋初值
        INC     DPTR
        DJNZ    R6，ACR
        MOV     DPTR，#BTR0     ；总线定时寄存器 0
        MOV     A，#03H
        MOVX    @DPTR，A
        MOV     DPTR，#BTR1     ；总线定时寄存器 1
        MOV     A，#0FFH        ；16MHz 晶振情况下，设置波特率为 80kbps.
        MOVX    @DPTR，A
        MOV     DPTR，#OCR      ；输出控制寄存器
        MOV     A，#0AAH
        MOVX    @DPTR，A
        MOV     DPTR，#RBSA     ；接收缓存器起始地址寄存器
        MOV     A，#0           ；设置接收缓存器 FIFO 起始地址为 0
        MOVX    @DPTR，A
        MOV     DPTR，#TXERR    ；发送错误计数寄存器.
        MOV     A，#0           ；清除发送错误计数寄存器
        MOVX    @DPTR，A
        MOV     DPTR，#ECC      ；错误代码捕捉寄存器
        MOVX    A，@DPTR        ；清除错误代码捕捉寄存器
        MOV     DPTR，#MODE     ；方式寄存器
        MOV     A，#08H         ；设置单滤波接收方式，并返回工作状态
        MOVX    @DPTR，A
        RET
```

2、发送子程序

发送子程序负责节点报文的发送。发送时用户只需将待发送的数据按特定格式组合成一帧报文，送入 SJA1000 发送缓存区中，然后启动 SJA1000 发送即可。当然在往 SJA1000 发送缓存区送报文之前，必须先作一些判断（如下文程序所示）。发送程序分发送远程帧和数据帧两种，远程帧无数据场。下面以发送数据帧为例对发送子程序作一个说明。

```
TDATA： MOV     DPTR，#SR       ；状态寄存器
        MOVX    A，@DPTR        ；从 SJA1000 读入状态寄存器值
        JB      ACC.4，TDATA    ；判断是否正在接收，正在接收则等待
TS0：   MOVX    A，@DPTR
```

```
        JNB     ACC.3，TS0          ；判断上次发送是否完成，未完成则等待发送完成
TS1：   MOVX    A，@DPTR
        JNB     ACC.2，TS1          ；判断发送缓冲区是否锁定，锁定则等待
TS2：   MOV     DPTR，#CANTXB       ；SJA1000 发送缓存区首址
        MOV     A，#88H             ；发送数据长度为 8 个字节的扩展帧格式报文
        MOVX    @DPTR，A
        INC     DPTR
        MOV     A，#ID0             ；4 个字节的标识符（ID0-ID3），依据实际情况赋值
        MOVX    @DPTR，A
        INC     DPTR
        MOV     A，#ID1
        MOVX    @DPTR，A
        INC     DPTR
        MOV     A，#ID2
        MOVX    @DPTR，A
        INC     DPTR
        MOV     A,#ID3
        MOVX    @DPTR，A
        MOV     R0，#TRDATA         ；CPU 发送数据区首址，数据内容由用户定义
MTBF：  MOV     A，@R0
        INC     DPTR
        MOVX    @DPTR，A
        INC     R0
        CJNE    R0，#TRDATA+8，MTBF；向发送缓冲区写 8 个字节
        MOV     DPTR，#CMR          ；命令寄存器地址
        MOV     A，#01H
        MOVX    @DPTR，A            ；启动 SJA1000 发送
        RET
```

3、查询方式接收子程序

接收子程序负责节点报文的接收以及其它情况处理。接收子程序比发送子程序要复杂一些，因为在处理接收报文的过程中，同时要对诸如总线脱离、错误报警、接收溢出等情况进行处理。SJA1000 报文的接收主要有两种方式：中断接收方式和查询接收方式，如果对通信的实时性要求不是很强，建议采用查询接收方式。两种接收方式编程的思路基本相同。下面仅以查询方式接收报文为例对接收子程序作一个说明。

```
SEARCH：
        MOV     DPTR，#SR           ；状态寄存器地址
        MOVX    A，@DPTR
        ANL     A，#0C3H            ；读取总线脱离、错误状态、接收溢出、有数据等位
        JNZ     PROC
        RET                         ；无上述状态，结束
PROC：  JNB     ACC.7，PROCI
BUSERR：
```

```
        MOV     DPTR，#IR           ；IR 中断寄存器；出现总线脱离
        MOVX    A，@DPTR            ；读中断寄存器，清除中断位.
        MOV     DPTR，#MODE         ；方式寄存器地址
        MOV     A，#08H             ；
        MOVX    @DPTR，A            ；将方式寄存器复位请求位清 0
        LCALL   ALARM.              ；调用报警子程序
        RET
        NOP
PROCI：MOV      DPTR，#IR           ；总线正常
        MOVX    A，@DPTR            ；读取中断位
        JNB     ACC.3，OTHER
OVER:   MOV      DPTR，#CMR         ；数据溢出中断置位.
        MOV     A，#0CH
        MOVX    @DPTR，A            ；在命令寄存器中清除数据溢出和释放接收缓冲区
        RET
        NOP
OTHER：JB       ACC.0，RECE         ；IR.0=1，接收 FIFO 未满或接收中断使能
       LJMP     RECOUT              ；IR.0=0，接收缓冲区无数据，退出接收
       NOP
RECE：  MOV     DPTR，#CANRXB       ；接收缓冲区首地址（16）；准备读取数据
        MOVX    A，@DPTR            ；首字节是接收帧格式字
        JNB     ACC.6，RDATA        ；RTR=1 是远程请求帧，无数据
        MOV     DPTR，#CMR
        MOV     A，#04H             ；CMR.2=1 释放接收缓冲区
        MOVX    @DPTR，A            ；只有接收了数据才能释放接收缓冲区
        LCALL   TDATA               ；发送对方请求的数据
        LJMP    RECOUT              ；退出接收
        NOP
RDATA：
        MOV     DPTR，#CANRXB       ；读取并保存接收缓冲区的数据
        MOV     R1，#CPURBF         ；CPU 片内接收缓冲区首址
        MOVX    A，@DPTR            ；读取读取 CAN 缓冲区的 2 号字节
        MOV     @R1，A              ；保存
        ANL     A，#0FH             ；截取低 4 位是数据长度（0~8）
        ADD     A，#4               ；加 4 个字节的标识符（ID）
        MOV     R6，A
RDATA：INC      DPTR
        INC     R1
        MOVX    A，@DPTR
        MOV     @R1，A
        DJNZ    R6，RDATA0          ；循环读取与保存
        MOV     DPTR，#CMR
```

```
        MOV     A，#04H                 ；释放 CAN 接收缓冲区
        MOVX    @DPTR，A
RECOUT：MOV     DPTR，#ALC              ；释放仲裁丢失捕捉寄存器和错误捕捉寄存器
        MOVX    A，@DPTR
        MOV     DPTR，#ECC
        MOVX    A，@DPTR
        NOP
        RET
```

结束语

上述介绍的是 SJA1000 工作在 PeliCAN 模式下的三种最基本的操作子程序。 由于篇幅的关系，这里没有列出和解释 SJA1000 内部寄存器的地址和位定义。有关接收屏蔽寄存器（AMR）和接收编码寄存器（ACR）的使用在以往的文章中已有介绍。至于其它一些寄存器或位功能的使用，没有在上述例程中体现出来，需要视实际情况而定。

参考文献

1 PHILIPS SJA1000 stand-alone CAN controller product specification 2000 Jan 04

2 邬宽明 CAN 总线原理和应用系统设计 北京航空航天大学出版社 1996