CAN总线数据通信系统的设计

摘　　要

现场总线是当今自动化领域技术发展的热点之一，被誉为自动化领域的计算机局域网。它的出现为分布式控制系统实现各节点之间实时、可靠的数据通信提供了强有力的技术支持。CAN(Controller Area Network)属于现场总线的范畴，是一种多主方式的串行通讯总线，数据通信实时性强。与其它现场总线比较而言，CAN 总线具有通信速率高、容易实现、可靠性高、性价比高等诸多特点。

本系统要在单片机中实现CAN总线的接口，通过CAN总线，实现两个模块之间的数据通讯。系统主要由四部分所构成：PC机、微控制器80C51、独立CAN通信控制器SJA1000和CAN总线收发器PCA82C250。微处理器80C51负责SJA1000的初始化，通过控制SJA1000实现数据的发送和接收等通信任务。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。

本系统通过扩展 CAN 总线控制器 SJA1000，在单片机系统中实现了 CAN 总线的接口，并且编写了 SJA1000 的驱动程序，通过读写其的内部寄存器，完成工作方式的设置、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）的设置等基本操作；利用各基本操作，完成了对 SJA1000 的初始化，并且实现了数据发送和接收。

目　　录

第 1 章　原理与方案

1.1　设计目的与要求

扩展 CAN 总线控制器，在单片机系统中实现 CAN 总线的接口，并编写接口芯片的驱动程序。通过 CAN 总线，实现两个模块之间的数据通讯，CPU 控制第一个模块发送 1 帧数据，第二个模块收到这帧数据并送至另一个 CPU 的内部存储器。

1.2　CAN 总线介绍

CAN 全称为“Controller Area Network”， 即控制器局域网，是国际上应用最广泛的现场总线之一。最初 CAN 被设计作为汽车环境中的微控制器通讯，在车载各电子控制装置 ECU 之间交换信息，形成汽车电子控制网络。比如发动机管理、系统变速箱控制器、仪表装备中，均嵌入 CAN 控制装置。

一个由 CAN 总线构成的单一网络中，理论上可以挂接无数个节点。实际应用中，节点数目受网络硬件的电气特性所限制。例如当使用 **Philips PCA82C250 作为 CAN 收发器时，同一网络中允许挂接 110 个节点**。CAN 可提供高达 1Mbit/s 的数据传输速率，这使实时控制变得非常容易，另外硬件的错误检定特性也增强了 CAN 的抗电磁干扰能力。

CAN 是一种多主方式的串行通讯总线。基本设计规范要求有高的位速率，高抗电磁干扰性，而且能够检测出产生的任何错误。当信号传输距离达到 10Km 时，CAN 仍可提供高达 50Kbit/s 的数据传输速率。由于 CAN 总线具有很高的实时性能，因此 CAN 已经在汽车工业、航空工业、工业控制、安全防护等领域中得到了广泛应用。

1.3 设计方案

在本系统中，采用 80C51 单片机，80C51 与 PC 机串行通信，设置 SJA1000 工作于 Intel 模式，由 PC 机发送的数据写入 SJA1000 并通过 CAN 收发器发送。接收数据是通过中断进行的，CAN 总线传输过来的数据经 CAN 接口芯片 PCA82C250 接收并写入 SJA1000 的 RXFIFO，然后通过中断提请 CPU 读取，读取的数据上传送给 PC 机。

总体设计框图如图 1-2 所示。

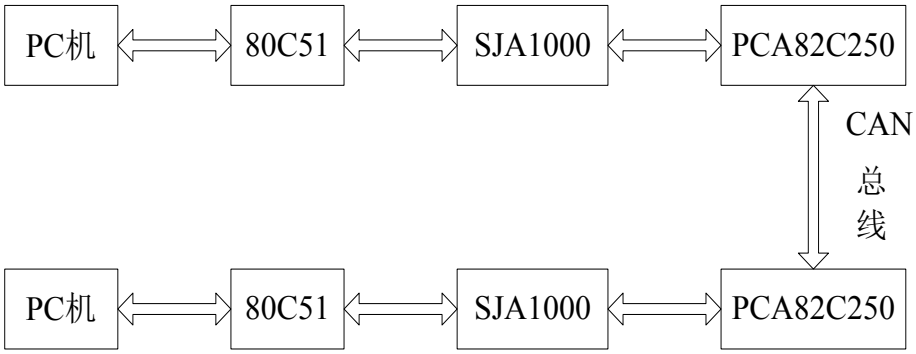

图 1-2　总体设计框图

1.3.1 硬件设计方案

1. 芯片介绍

SJA1000：独立式 CAN 控制器，具有 64 字节的 FIFO 作为接收缓存。

6N137：高速光隔，最高速度 10Mb / s，用于保护 CAN 控制器。

PCA82C250：CAN 总线收发器，是 CAN 控制器与 CAN 总线的接口器件，对 CAN 总线差分式发送。

2. CAN 控制器与 CPU 接口设计

对于 CPU 来说，CAN 控制器是确保双方独立工作的存贮器映象外围设备。CAN 控制器与外部 CPU 的接口是通过控制器接口逻辑（CIL）实现的，

80C51 的 CPU 通过将地址总线（AB）和数据总线（DB）连接到 CIL 上来完成与 CAN 控制器之间的信息交换,不需要专门的控制总线（CB），CPU 与 PCA82C250 之间的状态、控制和命令信号的交换在 CAN 控制器中完成。

SJA1000 与单片机的接口电路如图 1-3 所示。

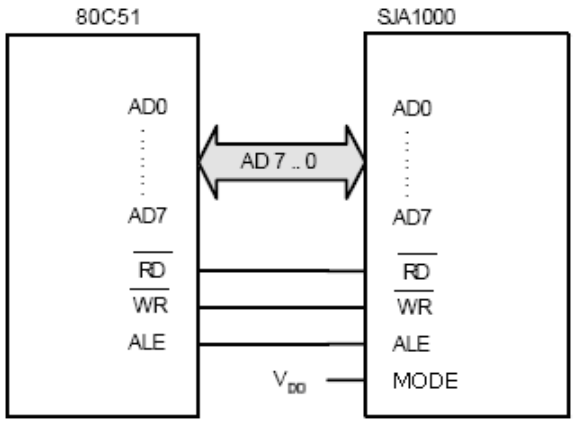

图 1-3　SJA1000 与单片机的接口电路

3. CAN 控制器工作电路的连接

为了增强 CAN 总线节点的抗干扰能力，SJA1000 的 TX0 和 RX0 并不是直接与 PCA82C250 的 TXD 和 RXD 相连，而是通过高速光隔 6N137 后与 PCA82C250 相连，这样就很好的实现了总线上各 CAN 节点间的电气隔离。

若 PCA82C250 处于 CAN 总线的网络终端，总线接口部分需加一个 120 欧姆的匹配电阻。

CAN 控制器工作电路如下图所示:

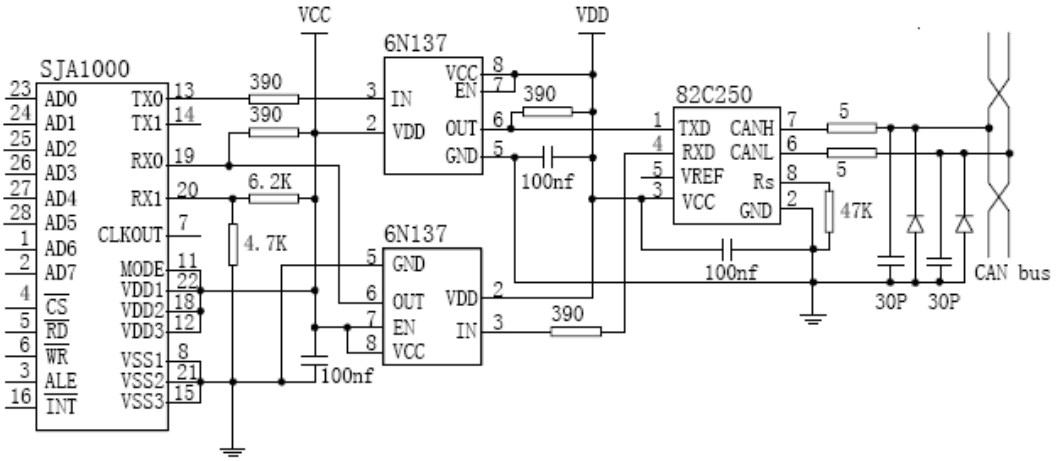

图 1-4　CAN 控制器工作电路

1.3.2　软件设计方案

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。

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。

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。

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，两种接收方式编程的思路基本相同，如果对通信的实时性要求不是很强，一般采用查询接收方式。

第 2 章　硬件连接与说明

2.1　硬件连接

单片机与 CAN 模块等外围器件的连接如图 2-1 所示。

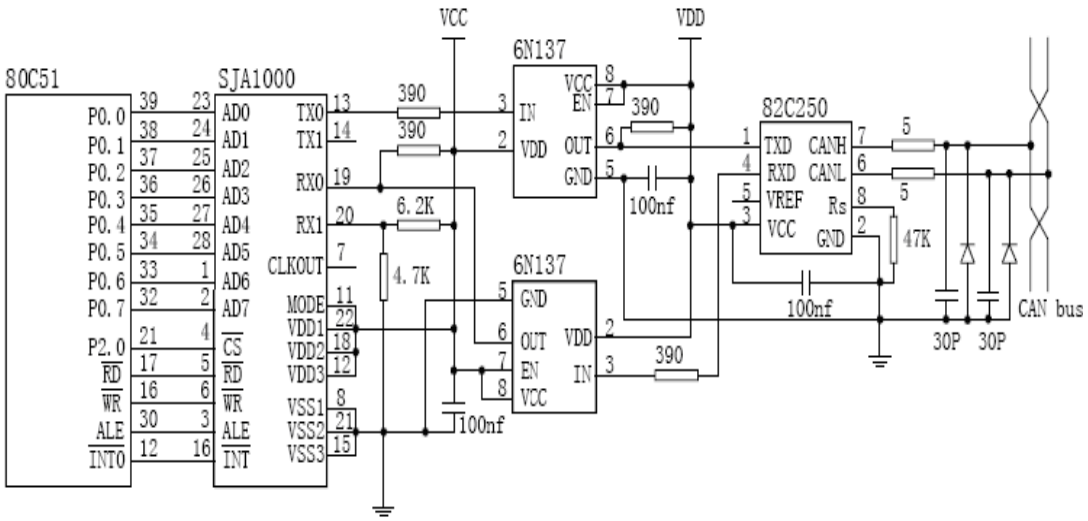

图 2-1　系统原理图

2.1.1　模块使用说明

CAN 总线模块由一个 CAN 总线控制器 SJA1000 和一个 CAN 收发器 PCA82C

250 组成,它们共同构成一个 CAN 节点。模块的电源由接口挂箱上的接口插座提供。

模块上的 RESET、INT、TX0、RX0 插孔分别对应于 SJA1000 芯片上的相应引脚。模块上带有上电复位电路，也可通过 RESET 插孔进行手动复位，只需在 RESET 上加上负脉冲。

模块上提供两个 RJ45 接口和一组“CANH、CANL”插孔接口，这三组接口是完全一致的。对于近距离 CAN 模块之间的通讯，可将各模块的“CANH、CANL”插孔用导线连接；对于远距离 CAN 模块之间的通讯，则可用双绞线连接各 RJ45 接口。

每个 CAN 模块上都有一组终端电阻接口，即“A、B”插孔。当总线上只有两个 CAN 节点时，终端电阻可不接。如总线上的 CAN 节点数为 3 个或 3 个以上时，必须有一个而且只能有一个 CAN 模块接上终端电阻。具体接法为：将 A 插孔和 CANL 插孔、B 插孔和 CANH 插孔分别用导线连接。

2.1.2 实验箱连线

两个 CAN 模块分别接在两个实验台上，第一个模块（发送）跳线接 LCS2,第二个模块（接收）跳线接 LCS3，用双绞线连接两个模块的 RJ45 接口，将第一个 CAN 模块接上终端电阻。

2.2 CAN 总线控制器 SJA1000

SJA1000 是一种独立的 CAN 控制器，主要用于移动目标和一般工业环境中的区域网络控制。它是 Philips 半导体公司 PCA82C200 CAN 控制器（BasicCAN）的替代产品，增加了一种新的操作模式——PeliCAN，这种模式支持具有很多新特性的 CAN2.0B 协议。

2.3 CAN 控制器接口 PCA82C250

PCA82C250 是 CAN 协议控制器和物理总线间的接口，它主要是为汽车中高速通讯（高达 1Mbps）应用而设计。此器件对总线提供差动发送能力，对 CAN 控制器提供差动接收能力，与 ISO11898 标准完全兼容。

PCA82C250 芯片由接收器、驱动器、基准电压产生电路、工作模式选择电路及保护电路等组成。PCA82C250 内部的限流电路可以防止发送输出级对电池电压的正端和负端短路。虽然在这种故障条件出现时，功耗将增加，但这种特性可以阻止发送器输出级的破坏。

在节点温度大约超过 160℃时，两个发送器输出端的极限电流将减少。由于发送器是功耗的主要部分，因此芯片温度会迅速降低。PCA82C50 芯片的其他部分将继续工作。当总线短路时，热保护十分重要。

CANH 和 CANL 两条线也可以防止在汽车环境下可能发生的电气瞬变现象。

第 3 章　软件流程图及说明

3.1　软件流程图

3.1.1　主程序流程图

程序开始运行后，先调用初始化子程序，分别对两个 CAN 模块中的 SJA1000 进行初始化，然后把要发送的数据写入 CPU 的存储器中，然后循环调用发送数据子程序和接收数据子程序。具体流程如图 3-1 所示。

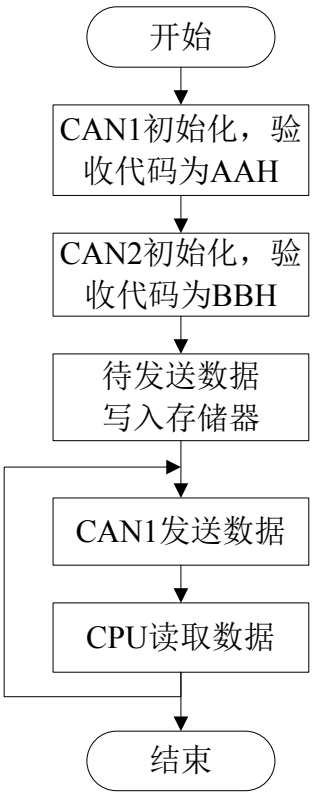

图 3-1　主程序流程图

3.1.2　初始化子程序流程图

初始化子程序先设置 MOD 选择复位模式，然后分别设置 CDR 选择工作模式；设置 IER 选择中断类型；设置 BTR0、BTR1 设定传输速率；设置 OCR 选择输出模式；设置 ACR、AMR 设定接收数据类型；RBSA、TXERR、ECC 均清零，最后设置 MOD 进入工作模式。具体流程如图 3-2 所示。

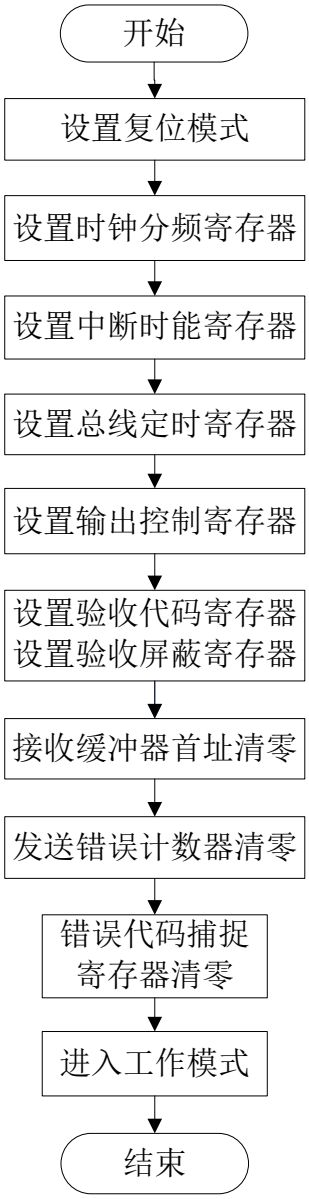

图 3-2　初始化子程序流程图

3.1.3　发送数据子程序流程图

发送数据子程序先把三个控制字节写入发送缓冲区，然后把等待发送的数据也写入发送缓冲区，最后设置 CMR，发出发送请求、启动 SJA1000 发送数据。具体流程如图 3-3 所示。

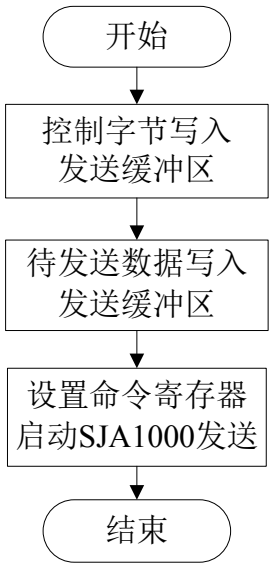

图 3-3　发送数据子程序流程图

3.1.4　接收数据子程序流程图

接收数据子程序首先要读 SR 和 IR，判断工作状态及中断类型并做相应处理，若 RXFIFO 有数据，应判断帧类型并做相应处理，若数据正确则送至 CPU 的内部存储器。具体流程如图 3-4 所示。

3.2　软件实现过程

两个实验台运行程序 CAN.ASM（程序见附录），发送实验台全速运行程序，接收实验台要在主程序中调用接收数据子程序后设置断点，等待接收到数据后送至 CPU 的存储器，然后查看 CPU 的内部存储器 30H~37H 中的数据与程序中发送的数据是否一致。

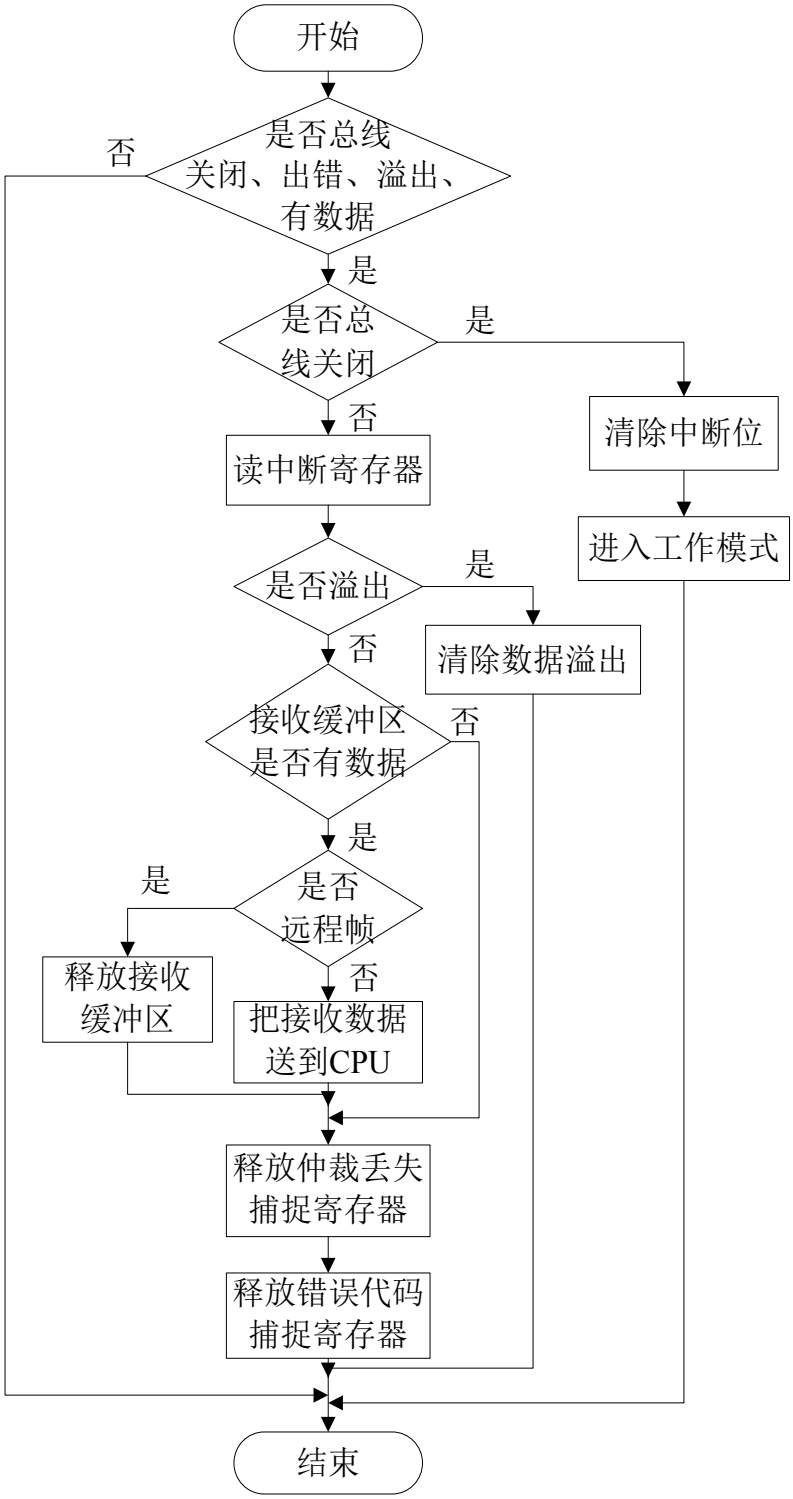
开始
是否总线
关闭、出错、溢出、
有数据
否
是
是否总
线关闭
是
否
清除中断位
读中断寄存器
进入工作模式
是否溢出
是
否
清除数据溢出
接收缓冲区
是否有数据
否
是
是否
远程帧
是
否
释放接收
缓冲区
把接收数据
送到CPU
释放仲裁丢失
捕捉寄存器
释放错误代码
捕捉寄存器
结束

图 3-4　接收数据子程序流程图

第 4 章　结果分析及心得体会

4.1　结果分析

本试验通过扩展 CAN 总线控制器，在单片机系统中实现了 CAN 总线的接口，通过 CAN 总线，实现了两个模块之间的数据通讯，在第一个模块中发送 1 帧数据，在第二个模块中收到这帧数据并送至了 CPU 的内部存储器 30H~37H。

如果要修改发送数据，只需修改程序中“TXDATA”后 8 个字节的数据即可。

发送实验台内部存储器如图 4-1 所示。

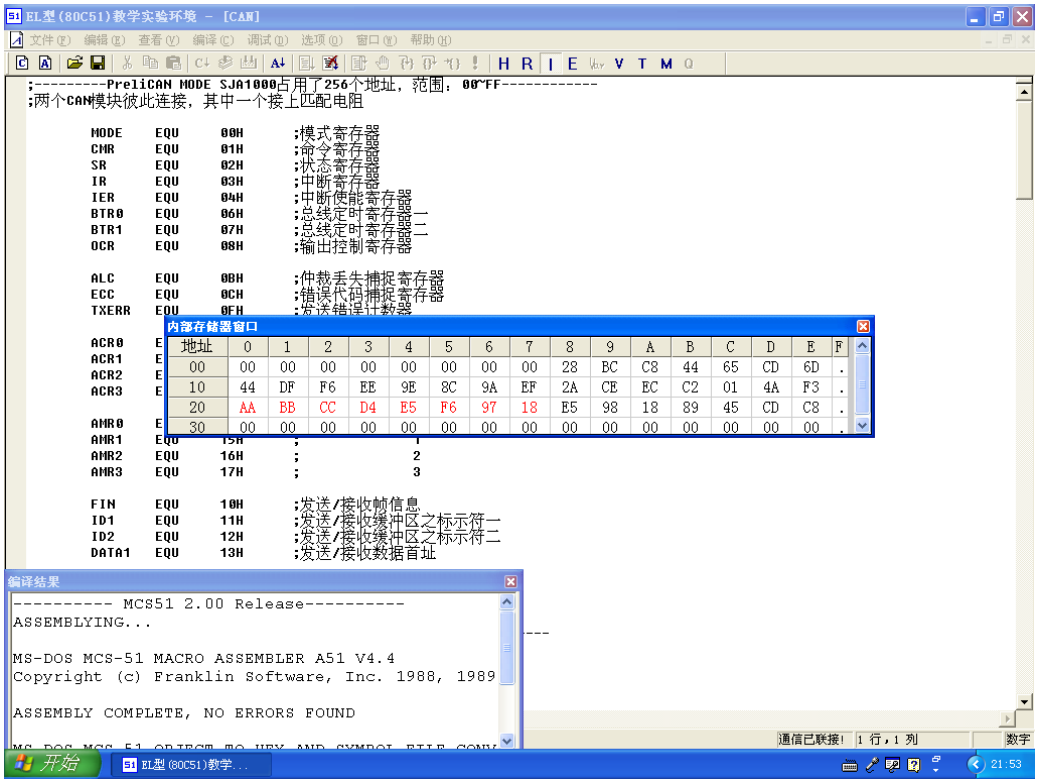

图 4-1　发送实验台

接收实验台内部存储器如图 4-2 所示。

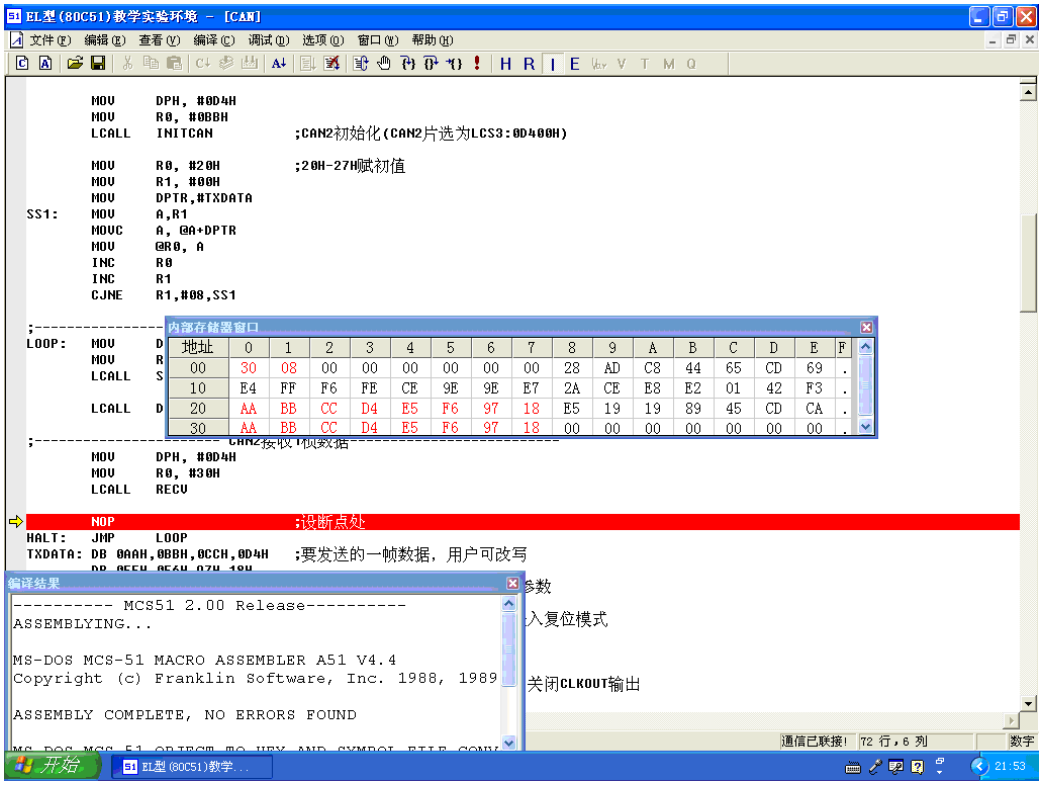

图 4-2　接收实验台

4.2　心得体会

通过本次设计对 CAN 总线的工作原理及其与 CPU 的接口方式有了一定程度的理解。基于 CAN 总线可实现多种数据的传输，例如：可将文字、图像等进行编码后传输，并在接收端进行解码后恢复，由于时间有限未做深入研究。

开发 CAN 总线控制节点时，还可以使用集成了 CAN 控制器的 CPU 80C592，但支持 80C592 的开发工具少，给开发工作带来一定的难度。所以一般使用独立的 CAN 控制器 SJA1000 配合单片机进行开发。

4.2.1 CAN 应用中的问题

SJA1000 有 BasicCAN 和 PeliCAN 两种工作模式，SJAl000 上电复位后自动进入 BasicCAN，因此选用 PeliCAN 模式必须在程序中设置时钟分频寄存器（CDR）选择工作模式。

总线定时寄存器的设置极大影响了 CAN 性能的发挥。一般来说，若硬件连接无误，通信失败的主要原因在于总线定时寄存器设置不当。在实验中，曾出现过因 SJA1000 的时钟电路中晶振严重漂移导致通信失败的情况。

ACR 和 AMR 两个寄存器构成硬件过滤，CAN 节点通过它来决定是否接收总线上的数据，是否置 CAN 的接收中断，这极大地提高了系统的灵活性。

可以通过中断寄存器（IR）、状态寄存器（SR）查询 CAN 总线的工作状态，了解数据传输状况。为了保证数据的正常收发，需要对状态寄存器和中断寄存器各状态位的变化做出相应处理。

4.2.2 CAN 总线的其他应用

CAN（Controller Area Network）总线最早由德国 BOSCH 公司提出，主要用于汽车内部测量与控制中心之间的数据通信。由于其良好的性能，在世界范围内广泛应用于其他领域当中，如工业自动化、汽车电子、楼宇建筑、电梯网络、电力通讯和安防消防等诸多领域，并逐渐成为这些行业的主要通讯手段。

一个由CAN总线构成的单一网络中，可以挂接多个节点，实际应用中只需要设置几个节点为上位节点与PC机进行通信，其他节点则可以做其他用途。如用于数据的采集，则与A/D转换芯片相接即可；如与控制相关，则与控制口相接即可，这样一来可以灵活地构成各种系统。

CAN总线具有多方面的优势，可以组建一个具有高可靠性、远距离、多节点、多主方式的设备通讯网络，例如：我国许多煤矿中都采用基于

CAN-BUS网络的煤矿通讯网络。

附录　程序清单

CAN.ASM

```
MOD      EQU   00H      ;模式寄存器
CMR      EQU   01H      ;命令寄存器
SR       EQU   02H      ;状态寄存器
IR       EQU   03H      ;中断寄存器
IER      EQU   04H      ;中断使能寄存器
BTR0     EQU   06H      ;总线定时寄存器 0
BTR1     EQU   07H      ;总线定时寄存器 1
OCR      EQU   08H      ;输出控制寄存器
ALC      EQU   0BH      ;仲裁丢失捕捉寄存器
ECC      EQU   0CH      ;错误代码捕捉寄存器
TXERR    EQU   0FH      ;发送错误计数器
ACR0     EQU   10H      ;验收代码寄存器 0
ACR1     EQU   11H      ;验收代码寄存器 1
ACR2     EQU   12H      ;验收代码寄存器 2
ACR3     EQU   13H      ;验收代码寄存器 3
AMR0     EQU   14H      ;验收屏蔽寄存器 0
AMR1     EQU   15H      ;验收屏蔽寄存器 1
AMR2     EQU   16H      ;验收屏蔽寄存器 2
AMR3     EQU   17H      ;验收屏蔽寄存器 3
FIN      EQU   10H      ;发送/接收帧信息
ID1      EQU   11H      ;发送/接收缓冲区之标示符一
```

```
ID2             EQU     12H             ;发送/接收缓冲区之标示符二
```

```
        DATA1       EQU     13H         ;发送/接收数据首址
        RBSA        EQU     1EH         ;接收缓冲器起始地址
        CDR         EQU     1FH         ;时钟分频寄存器
;--------------- READER COS 1.0 -------------------------
         ORG        4000H
         JMP        START
         ORG        4080H
;-------------------------------主程序---------------------------------
START: MOV      DPH, #0D3H          ;CAN1 初始化(CAN1 片选为
                                    ;LCS2:0D300H)
         MOV    R0, #0AAH           ;验收代码为 AAH
         LCALL  INITCAN
         MOV    DPH, #0D4H          ;CAN2 初始化(CAN2 片选为
                                    ;LCS3:0D400H)
         MOV    R0, #0BBH           ;验收代码为 BBH
         LCALL  INITCAN
;---------------------------------------------------------------
         MOV    R0, #20H             ;20H-27H 赋初值
         MOV    R1, #00H
         MOV    DPTR,#TXDATA
SS1:     MOV    A,R1
         MOVC   A, @A+DPTR
         MOV    @R0, A
         INC    R0
         INC    R1
         CJNE   R1,#08,SS1
```

```
;------------------------ CAN1 发送 1 帧数据------------------
LOOP:   MOV     DPH, #0D3H
        MOV     R0, #20H
        LCALL   SEND            ;发送 20H 为首址的 1 帧数据(前三
                                ;控制字节为:08H、BBH、FFH，由程序给出）
        LCALL   DELAY           ;调用延时子程序
;-------------------------CAN2 接收 1 帧数据-----------------------
        MOV     DPH, #0D4H
        MOV     R0, #30H
        LCALL   RECV            ;调用接收数据子程序
        NOP                     ;设断点处
HALT:   JMP     LOOP

TXDATA:DB 0AAH,0BBH,0CCH,0D4H ;要发送的一帧数据，用户可改写
        DB 0E5H,0F6H,97H,18H
;----------------------------初始化子程序-------------------------------
INITCAN:                        ;DPH、R0 为入口参数
        MOV     DPL,#MOD        ;模式寄存器，选择单验收滤波器模
                                ;式，进入复位模式
        MOV     A,#09H
        MOVX    @DPTR,A
        MOV     DPL,#CDR        ;时钟分频器，选择 PeliCAN 模式，
        MOV     A,#88H          ;关闭 CLKOUT 输出
        MOVX    @DPTR, A
        MOV     DPL,#IER        ;中断使能寄存器，开溢出、错误、
        MOV     A,#0DH          ;接收中断
```

```
MOVX    @DPTR,A
MOV     DPL,#BTR0          ;总线定时寄存器 0
MOV     A,#03H
MOVX    @DPTR,A
MOV     DPL,#BTR1          ;总线定时寄存器 1，6MHz 晶振，
MOV     A,#0FFH            ;波特率 30Kbps
MOVX    @DPTR,A
MOVX    A, @DPTR
MOV     DPL,#OCR           ;输出控制寄存器，
                           ;选择正常输出模式
MOV     A,#0AAH
MOVX    @DPTR,A
MOV     DPL,#ACR0          ;验收代码寄存器 ACR0
MOV     A, R0
MOVX    @DPTR,A
INC      DPTR              ;验收代码寄存器 ACR1
MOV     A,#0FFH
MOVX    @DPTR,A
INC      DPTR              ;验收代码寄存器 ACR2
MOVX    @DPTR,A
INC      DPTR              ;验收代码寄存器 ACR3
MOVX    @DPTR,A
MOV     DPL,#AMR0          ;验收屏蔽寄存器 AMR0
MOV     A,#00H
MOVX    @DPTR,A
INC      DPTR              ;验收屏蔽寄存器 AMR1
MOV     A,#0FFH
```

```
        MOVX    @DPTR,A
        INC     DPTR                ;验收屏蔽寄存器 AMR2
        MOVX    @DPTR,A
        INC     DPTR                ;验收屏蔽寄存器 AMR3
        MOVX    @DPTR,A
        MOV     DPL, #RBSA          ;接收缓冲器起始地址为 0
        MOV     A, #00H
        MOVX    @DPTR, A
        MOV     DPL, #TXERR         ;清除发送错误计数器
        MOVX    @DPTR, A
        MOV     DPL, #ECC           ;清除错误代码捕捉寄存器
        MOVX    @DPTR, A
        MOV     DPL,#MOD            ;单验收滤波器模式，返回工作模式
        MOV     A,#08H
        MOVX    @DPTR,A
        RET
;-------------------------发送数据子程序-----------------------
SEND:                               ;DPH、R0 为入口参数
        MOV     DPL,#FIN            ;SJA1000  发送缓存区首址
        MOV     A, #08H
        MOVX    @DPTR, A
        INC     DPL
        MOV     A, #0BBH
        MOVX    @DPTR, A
        INC     DPL
        MOV     A, #0FFH
        MOVX    @DPTR, A
```

```
        INC     DPL
        MOV     R2, #08H
SEND1:  MOV     A, @R0          ;R0 为发送数据首址
        MOVX    @DPTR, A
        INC     R0
        INC     DPL
        DJNZ    R2, SEND1
        MOV     DPL,#CMR        ;命令寄存器发出发送请求，
        MOV     A,#01H          ;启动 SJA1000  发送
        MOVX    @DPTR,A
        RET
;----------------------接收数据子程序--------------------
RECV:                           ;DPH、R0 为入口参数
        MOV     DPL,#SR         ;状态寄存器地址
        MOVX    A,@DPTR
        ANL     A, #0C3H        ;读取总线关闭、出错、接收溢出、
                                ;有数据等位
        JNZ     PROC
        RET                     ;无上述状态，结束
PROC:   JNB     ACC.7, PROC1
BUSERR:MOV      DPL, #IR        ;IR 中断寄存器，出现总线关闭
        MOVX    A, @DPTR        ;读中断寄存器，清除中断位
        MOV     DPL, #MOD
        MOV     A, #08H
        MOVX    @DPTR, A        ;将方式寄存器复位请求位清 0
        RET
        NOP
```

```
PROC1: MOV      DPL, #IR          ;总线正常
       MOVX     A, @DPTR          ;读取中断位
       JNB      ACC.3, OTHER
OVER:  MOV      DPL, #CMR         ;数据溢出处理
       MOV      A, #0CH
       MOVX     @DPTR, A          ;清除数据溢出位，释放接收缓冲区
       RET
       NOP
OTHER: JB       ACC.0, RECE
       LJMP     RECOUT            ;接收缓冲区无数据
       NOP
RECE:  MOV      DPL, #FIN         ;接收缓冲区有数据
       MOVX  A,@DPTR
       JNB      ACC.6, RDATA
       MOV      DPL, #CMR         ;远程帧处理
       MOV      A, #04H
       MOVX     @DPTR, A
       LJMP     RECOUT
       NOP
RDATA: MOV      DPL, #FIN         ;发送/接收数据首址
       MOVX     A, @DPTR
       INC      DPL
       MOVX     A, @DPTR
       INC      DPL
       MOVX     A, @DPTR
       INC      DPL
       MOV      R2, #08H
```