

基于 MSP430 的 CAN 总线控制系统的设计

佟 宇, 李 伟

(大连理工大学机械工程学院, 大连 116024)

摘 要: 本文介绍了基于 MSP430F149 单片机的一种低功耗 CAN 总线控制系统。解决了 MSP430 系列单片机与 CAN 总线控制芯片之间的电平兼容问题, 着重阐述了控制系统硬件和软件部分的设计。

关键词: MSP430; 单片机; CAN 总线

中图分类号: TP13 **文献标识码:** B **文章编号:** 1002-2279(2004)01-0056-03

The Design of CAN Control System Based on MSP430 Chip Microcomputer

TONG Yu, LI Wei

(The Institute of mechanical engineering, Dalian University of Technology, Dalian 116023, China)

Abstract: The paper presents a low power consumption control system of CAN based on MSP430F149. Both the hardware design and the software design of the system are given emphasized descriptions. An efficient solution to the level compatibility problem between the MSP430 series of MCU and the control chips of CAN is proposed.

Key words: MSP430; chip microcomputer; CAN

1 引 言

在现代工业控制领域,分散式工业控制系统的应用越来越广泛。典型的分散式控制系统由现场设备、接口与计算设备以及通讯设备组成。现场总线能同时满足过程控制和制造业自动化的需要,因而现场总线已成为工业数据总线领域中最为活跃的一个领域。而控制器局域网(CAN)是现场总线中最有前途的一种总线。本文讨论了一种基于美国 TI 公司 MSP430 单片机的 CAN 总线控制系统。

2 MSP430 系列单片机特性

MSP430 是 TI 公司最新出产的一种具有超低功耗的功能强大的 16 位单片机, MSP430F149 是该系列中的一种型号。这种单片机的优点是:①超低功耗, MSP430F149 运行在 1Mhz 时钟条件下时,工作电流视工作模式不同为 0.1~400 μ A,工作电压为 1.8~3.6V。②具有强大的处理能力,采用 RISC 指令集,只需简洁的 27 条指令,高度正交指令结构,处理内核功能强大并且非常灵活。③超低功耗的数控振荡器技术,可以实现频率调节和无晶体运行。外接时钟频率范围为 32768Hz~8MHz,并且可以实现分频。从低功耗模式到运行模式的唤醒时间仅为 6 μ s。④程序代码空间为 60KB+256 字节的 FLASH, 2K 字节的 RAM。采用串行在线编程方式,为用户编译

程序和控制参数提供灵活的空间。⑤MSP430F149 具有强大的中断功能, 48 个 I/O 引脚, 它的 I/O 与传统的 I/O 不同, 每个 I/O 口分别对应输入、输出、功能选择、中断等多个寄存器, 使得功能口和通用 I/O 口复用, 在对同一个 I/O 口进行操作前要选择其要实现的功能, 这样大大的增强了端口的功能和灵活性, 我们就是利用 I/O 口来实现对 CAN 总线控制器的控制的。

由于 MSP430F149 具有以上的特点, 十分适合开发的要求, 所以选用其作为系统的主芯片。

3 CAN 总线简介

CAN(Controllor Area Net, 控制器局域网)总线是德国 Boshch 公司从 80 年代初为解决现代汽车中众多的控制与测试仪器之间的数据交换而开发的一种串行数据通讯协议, 它是一种多主总线, 通信介质可以是双绞线、同轴电缆或光导纤维。CAN 总线通信接口中集成了 CAN 协议的物理层和数据链路层功能, 可完成对通信数据的成帧处理(包括位填充、数据块编码、循环冗余检验等)。CAN 总线有 CAN2.0A 和 CAN2.0B 两种协议。CAN 协议的一个最大的特点是对通信数据块进行编码(取消了对站地址编码)。CAN 协议采用 CRC 校验并提供相应的错误信息处理功能, 保证了数据通信的可靠性。

佟宇(1955-),男(满族),黑龙江通河县,高级工程师,主要研究方向:数字控制技术及数字化仿形。
收稿日期:2003-05-19

CAN 总线通讯最高速度为 1Mbps (采用双绞线通讯距离 40m), 通讯距离 10Km 时仍可达 5Kbps, CAN 总线以其可靠性高、通讯速率快、稳定性好、抗干扰能力强、开发成本低等特点, 而被工控领域所普遍采用, 被公认为最有前途的现场总线之一。

本系统采用 PHILIPS 公司生产的 SJA1000CAN 控制器, 它是一种独立控制器, 用于移动目标和一般工业环境中的区域网控制。而 CAN 总线收发器选用 PCA82C250, 它是 CAN 控制器与 CAN 总线的接口器件, 对 CAN 总线差分式发送。

4 系统硬件的设计

4.1 系统的工作原理

MSP430F149 单片机在逻辑时序的控制下将要发送的数据通过 I/O 口发向 CAN 总线控制器的发

送缓冲区, 然后启动 CAN 总线控制器的发送命令, 此时 CAN 控制器将自动向总线发送数据, 不需 MSP430F149 的干预。若系统中有多多个 CAN 控制器同时向总线发送数据, 则 CAN 控制器通过信息帧中的标识符来进行仲裁 (在仲裁过程期间, 标识符首先被送至总线, 标识符的二进制数值越低, 其优先权越高), 控制执行节点中的 CAN 控制器检测到总线上有数据时会自动接收总线上的数据, 存入其接收缓冲区 (CAN 总线控制器通过激活接收器向网络发送远程帧, 借助标识符并置 RTR 位为高寻址数据源), 并向 MSP430 发送接收中断, 启动 MSP430 的中断接收服务程序, MSP430 通过执行中断接收服务程序, 从 CAN 控制器的接收缓冲区读取数据, 并对其进行处理后通过输出模块输出, 从而控制一些被控对象。电路结构如图 1 所示。

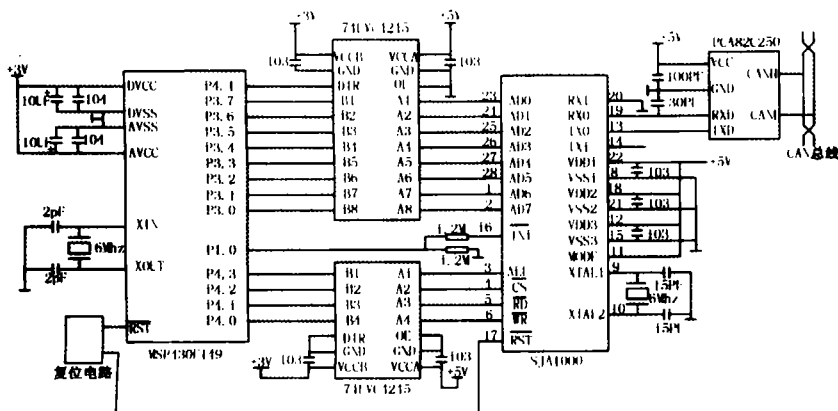


图 1 MSP430CAN 总线控制系统接口电路图

4.2 硬件结构设计

在设计中, 由于 MSP430 的 P1 和 P2 的 I/O 端口有中断功能, 而其余的 I/O 无中断功能, 所以选用 P3 口作为地址和数据总线进行对 SJA1000 控制器操作, 选用 P4 口部分引脚对 SJA1000 控制器的读写逻辑进行控制, 利用 P1.0 引脚的中断功能使得控制器的中断操作被 MSP430 响应。

虽然 MSP430 单片机的设计者们本意是希望他们的用户只需进行适当的选型, 无需扩展, 但事实上, 有些特殊功能的芯片他们并未集成, 扩展在所难免。而且 SJA1000 的工作电压是单 5V, 所以在电路设计中, 逻辑电平的转换是本次试验能否顺利进行的关键。在这个系统中, MSP430F149 和 SJA1000 之间有数据的相互交换, 所以它们之间的逻辑电平转换芯片我们采用 TI 公司提供的 SN74LVCC4245A。它是一种双电源的电平移位器,

采用 5V 和 3V 电源独立供电, 它的功能类似于常用的收发器 74LVC245, 所不同的是用两个电源而不是一个电源。74LVC4245 的电平移位在其内部进行。双电源能保证两边端口的输出摆幅都能达到满电源幅值, 并且有很好的噪声抑制性能。为了简化系统的设计, 对于 SJA1000 控制器的中断, 采用分压电阻直接和 MSP430F149 的 P1.0 引脚相连。

另外, 由于 MSP430 与 CAN 总线控制器的逻辑电压不匹配, 所以我们不能像用 MCS-51 系列单片机一样, 两个芯片只利用一个晶振。我们需要 MSP430 和 CAN 总线控制器单独外接晶振, 在本次试验中选用两个 6MHz 的晶振分别供给 MSP430F149 和 SJA1000 控制器。

5 系统软件的设计

系统软件的设计主要是通讯程序的设计。在程序设计中, 该通讯软件由三部分组成: MSP430 控制

器与 CAN 总线控制器的初始化程序、MSP430 控制器发送程序和 MSP430 控制器接收程序。在编程过程中,有两点值得注意:①由于 MSP430 目前的所有型号总线对外不开放,只能通过 I/O 口与 CAN 总线控制器相连,使用通用 I/O 口线模拟 CAN 总线控制器的读写时序完成与控制器的接口,这样我们必须通过编程来完成控制功能;②应该充分利用 MSP430 的特点,发挥它的超低功耗特性。

5.1 MSP430 控制器与 CAN 总线控制器初始化

在系统上电之后,两个控制器会自动复位,为了能够按照设计要求进行通讯,需要对控制器进行初始化。对于 CAN 总线控制器的初始化是通过 MSP430 向 SJA1000 控制段中的寄存器写入控制字来完成的。在复位期间,必须对下列寄存器进行初始化:模式寄存器(仅 PeliCAN 模式)、时分寄存器、接收代码、接收屏蔽寄存器、总线定时寄存器、输出控制寄存器等。初始化完成后,接收码、接收屏蔽、总线定时器以及输出控制寄存器的内容不应改变。而且当控制寄存器中的复位请求位置为高时,这些寄存器才能够被访问。程序流程图如图 2 所示。

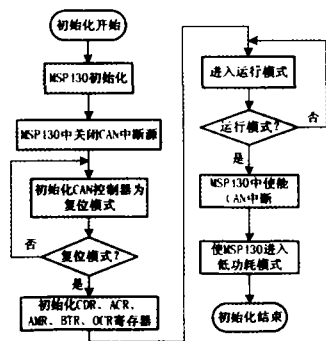


图 2 主程序(初始化)

下面是利用 MSP430C 语言对 CAN 总线寄存器进行初始化的部分程序:

```
void InitCan(void)
{
    P4DIR |= BIT0 + BIT1 + BIT2 + BIT3 + BIT4;
    //430 对 CAN 进行控制的引脚设为输出
    P4OUT&= ~(BIT4 + BIT3); //数据方向由
    430 向 SJA1000 发送,ALE 置低
    P4OUT |= BIT0 + BIT1 + BIT2; //CS,WR,RD
    置高
    P4OUT ^ = BIT4; //复位请求,CAN 进入 reset
    状态
    P3OUT=0X00;
    P4OUT ^ = BIT4;
    P4OUT ^ = BIT1;
    P3OUT=0X01;
```

```
P4OUT ^ =BIT1;
P4OUT ^ =BIT4; //初始化接收码寄存器,同理
P3OUT=0X00; // 进行其余寄存器的设置
P4OUT ^ =BIT4;
P4OUT ^ =BIT1;
P3OUT=0X01;
P4OUT ^ =BIT1;
P4OUT ^ =BIT4; //允许接收中断,清除复位请求
P3OUT=0X00;
P4OUT ^ =BIT4;
P4OUT ^ =BIT1;
P3OUT=0X01;
P4OUT ^ =BIT1; }
```

5.2 发送程序的设计

微处理器将发送数据发送到 CAN 控制器的发送缓存中,并将命令寄存器的发送请求标志位置 1,发送数据根据 CAN 总线协议在 CAN 总线控制器的作用下自动完成。发送程序我们采用中断方式进行处理。中断使能标志位在 BasicCAN 模式下位于控制寄存器中,在 PeliCAN 模式下位于中断使能寄存器中。只要 SJA1000 发送完数据,发送缓存器被锁为写状态,微处理器去检查状态寄存器中的发送缓存器状态标志位,然后将数据发送到发送缓存器中。程序分为主程序和中断处理程序,流程图如图 3、4 所示。

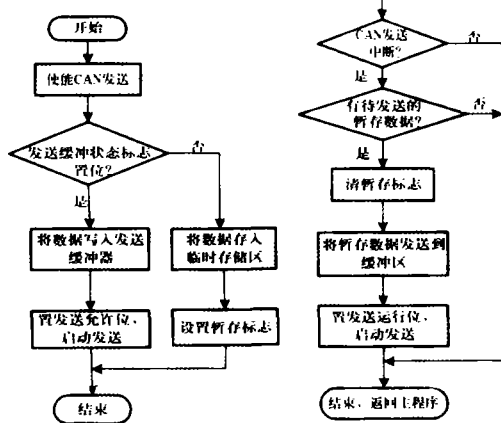


图3 CAN发送主程序

图4 CAN发送中断程序

5.3 接收程序设计

接收数据也是根据 CAN 总线协议在 CAN 总线控制器的作用下自动完成。接收的数据会先存入接收缓存器中。接收程序通过检查接收中断标志位和接收缓存器状态位将待发送的数据发送给微处理器。微处理器会将此数据读到内存中,并释放接收缓存。程序也分为主程序和中断处理程序,主程序的目的就是打开 CAN 的接收中断。(下转第 61 页)

9x 下编写虚拟设备驱动程序有两个不错的解决方案: Windows DDK 和 Vireo Software 出品的 VToolsD^[3]。VToolsD 比 DDK 更容易使用,因为它有一套 C++ 类库。本文基于 Windows 9x 平台的虚拟设备驱动程序(VxD)的基本原理和设计方法,并结合开发工具 VToolsD 设计了高精度时钟板的虚拟设备驱动程序和使用范例,部分设计代码如下。

(1)高精度时钟板的虚拟设备驱动程序

```
// 文件名: TRIGGER.cpp - 外部电平触发中断虚拟设备驱动程序
// 可任意使用 PC 机保留的中断, 但该中断在 CMOS 中应设置未被占用
#define DEVICE_MAIN
#include <vtoolscp.h>
Declare_Virtual_Device(TRIGGER)
#undef DEVICE_MAIN
#define IRQ_NUM 7 // 中断号, 高精度时钟中断
#define ADDRESSPASS CTL_CODE(FILE_DEVICE_UNKNOWN, // 异步过程调用
1, METHOD_NEITHER, FILE_ANY_ACCESS)
PVOID CallbackApc=0; // 异步过程回调函数
THREADHANDLE TheThread=0; // 回调的应用程序的线程句柄
DWORD TriggerDevice:: OnW32DeviceIoControl(PIOCTLPARAMS pDIOCPParams)
{
    switch(pDIOCPParams->dioc_IoctlCode) {
        case DIOC_OPEN;
        case DIOC_CLOSEHANDLE; return 0;
        case ADDRESSPASS;
            CallbackApc=pDIOCPParams->dioc_InBuf;
            TheThread=Get_Cur_Thread_Handle();
            return 0;
        default; return -1;
    }
}
```

```
}
}
TriggerInt:: TriggerInt ( ): VHardwareInt ( IRQ_NUM,
VPICD_OPT_CAN-SHARE,0,0){};
void TriggerInt:: OnHardwareInt (VMHANDLE hVM) // 高精度时钟中断触发中断服务程序
{
    VWIN32_QueueUserApc ( CallbackApc, 0,
TheThread);
    sendPhysicalEOI();
}
```

(2)主程序使用高精度时钟板的虚拟设备驱动程序

```
m_hDevice=CreateFile ( "\\.\Trigger.vxd" ,0,0,0, // 动态加载时钟 VxD 驱动程序
OPEN_EXISTING, FILE_FLAG_DELETE_ON_CLOSE,0);
DeviceIoControl(m_hDevice, ADDRESSPASS, NULL, 0, 0, 0, 0, 0); // 异步过程回调函数为空函数
while (! m_bEndThread) // 线程未结束一直运行
{
    SleepEx(INFINITE, TRUE); // 等待高精度时钟中断触发
    /*{ // TODO: Add extra code here begin (在这里加入用户处理代码)
    } // TODO: Add extra code here end (在这里加入用户处理代码)
}
CloseHandle(m_hDevice);
```

参考文献:

- [1] Microsoft Corporation. MSDN Library. 2000-07.
- [2] 上海无线电十九厂翻译组译. 美国德克萨斯仪器公司 TTL 集成电路特性应用手册[Z]. 上海: 上海市半导体器件工业公司上海市半导体器件研究所, 1976.
- [3] 侯俊杰译. Windows 95 系统程序设计—虚拟机器与 VxD 程序设计[M]. 台北: 松岗计算机图书数据股份有限公司, 2000.

(上接第 58 页)

程序流程图如图 5、6 所示。

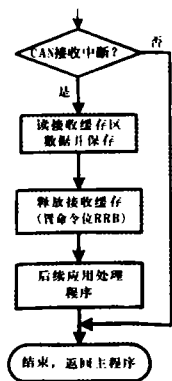


图5 CAN接收中断程序

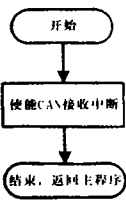


图6 CAN接收主程序

6 结束语

基于 MSP430 单片机的 CAN 总线控制系统, 由于具有通讯效率高、抗干扰能力强、传送距离远、功耗低的优点, 它将在自动化控制领域中应用越来越广泛。

参考文献:

- [1] 邬宽明. CAN 总线原理和应用系统的设计[M]. 北京: 北京航空航天大学出版社, 2001.
- [2] 胡大可. MSP430 系列 FLASH 型超低功耗 16 位单片机[M]. 北京: 北京航空航天大学出版社, 2001.
- [3] 饶运涛. 现场总线 CAN 原理与应用技术[M]. 北京: 北京航空航天大学出版社, 2003.
- [4] 周宝龙. 基于 CAN 控制器的单片机农业室温控制系统的设计[J]. 测控技术, 2000, (12): 19-21.
- [5] 胡大可. MSP430 系列单片机 C 语言程序设计与开发[M]. 北京: 北京航空航天大学出版社, 2003.