

文章编号: 1005-8451 (2010) 06-0052-03

## 基于双单片机的串行通信设计与应用

詹平红, 李红星

(炮兵学院 炮兵指挥自动化与仿真系, 合肥 230031)

**摘要:** 在安全性要求较高的环境中, 若控制系统只用一个单片机作为核心控制, 不论系统各种防护措施的设计如何完善, 只要单片机出现故障, 其环境安全就会受到威胁。这就是只含一个单片机的控制系统在实际应用中表现出的弊端, 要解决此弊端可以通过增加单片机在控制系统中的数量, 使单片机相互监测以满足高安全性要求。本文以铁路车站联锁控制系统中 ZPW-2000 编码模块为例, 详细介绍基于 AT90CAN128 双单片机串行通信的软、硬件设计方案。

**关键词:** 串行通信; 编码; AT90CAN128; ZPW-2000

中图分类号: U285

文献标识码: A

### Design and application of serial communication based on double MCU

ZHAN Ping-hong, LI Hong-xing

(Department of Artillery Command Automation and Simulation, Artillery Academy, Hefei 230031, China)

**Abstract:** The single MCU was the only kernel control part in conditions of high security requirements. No matter how the various protective measures was designed, environmental security should be threatened if microcontroller failed. This was the drawback of Control System with only one MCU. Multi-MCU Control System which could monitor mutually was designed to avoid this problem. The software and hardware design of serial communication with double MCU AT90CAN128 was introduced by the ZPW-2000 coding module of Interlocking Control System in the railway station.

**Key words:** serial communication; coding; AT90CAN128; ZPW-2000

单片机技术以微处理器技术及超大规模集成电路技术的发展为先导, 广泛应用于各个领域。但是不论在何领域使用, 单片机系统可靠安全运行是最基本的要求, 影响单片机系统可靠安全运行的主要因素除来自系统内部和外部的各种电气干扰外, 还会受系统结构设计、元器件选择、安装、制造工艺影响, 其中, 系统结构设计是最主要的<sup>[1]</sup>。在目前实际应用中, 多数控制系统在结构设计上均采用一个单片机控制, 系统在高可靠性、高安全性的环境中已出现了弊端, 为防止单片机出现故障, 导致控制系统运行失常, 可以通过增加单片机的数量来提高整个系统的安全可靠性。本文以铁路车站联锁控制系统中 ZPW-2000 编码模块的双单片机串行通信设计为例, 介绍基于 AT90CAN128 双单片机的串行通信设计与应用<sup>[2]</sup>。

### 1 微处理器 AT90CAN128 功能

#### AT90CAN128 单片机使用广泛, 成本低, 功

收稿日期: 2009-09-27

作者简介: 詹平红, 助教; 李红星, 在读硕士研究生。

能强大, 整合了一个V2.0A/V2.0B CAN 控制器和一个 128 kbyte 的闪存程序存储器。是一种高性能、低功耗的微处理器。先进的RISC结构, 4 kbyte 的EPROM, 4 kbyte 的内部SRAM, 工作于16 MHz 时性能高达 16 MIPS, 32 个通用工作寄存器, 53 个通用 I / O 口, 4 个带有比较模式灵活的定时器 / 计数器, 2 个可编程的 USART 接口, 可极大满足系统需要。

### 2 硬件设计

采用 AT90CAN128 双单片机实现铁路车站联锁控制系统中 ZPW-2000 编码模块的设计, 其主要功能是将联锁机下发的命令, 通过 CAN 总线通信分别存入单片机 MCU1 和 MCU2 中, MCU1 和 MCU2 将接收到的命令进行存储并相互发送, 两个单片机将接收的数据与对方发送的数据进行比较, 如一致, 单片机输出控制命令, 如不一致, 则等待下次接收的命令。开关检测实时检测编码开关状态, 并上传给单片机。对于编码开关的控制, 采用了双 MCU 同步“与”逻辑控制电路, 这样的

设计方案可以确保如果一个MCU由于干扰等因素产生错误动作时,不至于输出非安全性的信号。编码模块电路包括两个MCU电路、通信电路、动态电路、控制电路及检测电路等。它的外部时钟是16 MHz。硬件结构如图1。

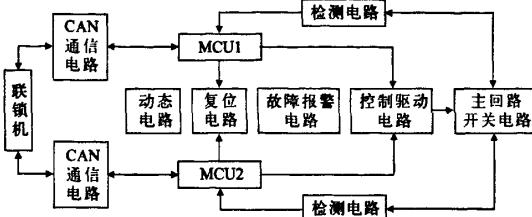


图1 硬件结构框图

### 3 硬件实现

使用AT90CAN128芯片的通用同步和异步串行接收器和转发器(USART),实现同步“与”逻辑控制,从而提高编码开关的准确性。

#### 3.1 USART 结构

在硬件连线中将MCU1的端口PE0和PE1与MCU2的端口PE0和PE1交替连接。USART结构如图2。

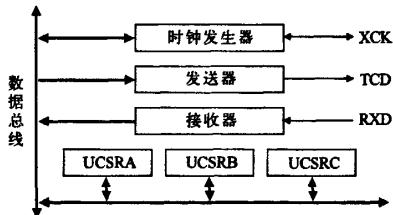


图2 USART结构图

USART主要包括3个部分:时钟发生器,发送器和接收器。时钟发生器包括同步从机操作用来与外部输入时钟进行同步的逻辑,以及波特率发生器。XCK(发送器时钟)引脚用于同步发送模式。发送器包括单个写缓冲器,串行移位寄存器,奇偶发生器以及处理不同的帧格式所需的控制逻辑。接收器具有时钟和数据恢复单元,是USART模块中最复杂的。恢复单元用于异步数据的接收。除了恢复单元,接收器还包括奇偶校验,控制逻辑,移位寄存器和两个接收缓冲器UDR。接收器支持与发送器相同的帧格式,而且可以检测帧错误,数据过速和奇偶校验错误。

#### 3.2 控制和状态寄存器

控制和状态寄存器由3个单元共享,分别是UCSRA、UCSRB、UCSRC。倍速模式(只适用于异步模式)受控于UCSRA寄存器的U2X,UCSRB根据需要使能接收器或发送器;UCSRC用于选择异步模式和同步模式;数据帧的结构由UCSRB和UCSRC寄存器中的UCSZ2:0、UPM1:0与USBS设定。

#### 3.3 控制和状态寄存器主要特征

(1)全双工操作(独立的串行接收和发送寄存器)。(2)同步或异步操作。(3)主机或从机提供时钟的同步操作。(4)高精度的比特率发生器。(5)支持5 bit、6 bit、7 bit、8 bit和9 bit数据位,1 bit或2 bit停止位。(6)硬件支持的奇偶校验操作。(7)数据过速检测及帧错误检测。(8)噪声滤波,包括错误的起始位检测和数字低通滤波器。(9)3个完全独立的中断(发送结束中断、发送数据寄存器中断及接收结束中断)。(10)多处理器通信模式和倍速异步通信模式。

### 4 软件设计

编码的软件设计采用结构化、模块化设计方案,兼顾实时性和可靠性,这种设计思路具有良好的可修改性、可移植性及可维护性,为将来增强软件功能做准备。软件采用ICCAVR编译器,它综合了编译器和工程管理器的集成工作环境(IED),是一个纯32 bit的程序,源文件全部被组织到工程之中。文件的编译和工程的构筑也在IDE的环境中完成;INTEL HEX格式文件的烧写文件和符合AVR Studio的调试文件(COFF格式)也在IDE中直接产生<sup>[3]</sup>。

模块的软件设计主要包括系统初始化、CAN通信、同步通信、命令控制及反馈检测5部分。

进行同步通信之前首先要对USART初始化。初始化过程通常包括比特率的设定,帧结构的设定,以及根据需要使能接收器或发送器。对于中断驱动的USART操作,在初始化时首先要清零全局中断标志位。重新改变USART的设置应该在没有数据传输的情况下进行。

同步通信过程采用查询发送和中断接收,使用的通信串口为USART0。发送数据缓冲寄存器和接收数据缓冲寄存器共享相同的I/O地址,称

为USART数据寄存器或UDR0。程序包括两次比较，即命令比较和状态比较。命令比较是两个MCU互发命令，将接收到的命令存入UDR0数据寄存器与保存的联锁机命令进行比较，如果命令一致，则命令同步成功，然后控制输出，否则退出控制输出模块，维持原始控制输出。状态比较则是返回编码开关的状态信息，进行同步通信。同步通信的发送程序和接受程序如下：

```
void Uart_Send(void)
{
    char i;
    if(Uart_Flag != BIT_flag)           / // 判断串口通信标志位
        return;
    Uart_Flag = 0
    while(!(UCSR0A & (1 << UDRE0)));
    // 判断缓冲器是否已准备好接收数据。
    UDR0 = 0xfb;
    // 发送帧头
    for(i=0;i<3;i++)
    {
        while(!(UCSR0A & (1 << UDRE0)));
        UDR0 = TB_Order;
        // 将数据存入缓冲区
    }
    while(!(UCSR0A & (1 << UDRE0)));
    UDR0 = 0xfc;
    // 发送帧尾
}
#pragma interrupt_handler RXC0_isr:22
// 中断向量 22, UART0 接收中断 22
void RXC0_isr(void)
{
    unsigned char data;
    if((UCSR0A & 0x80) != 0x80)           / // 判断接受缓冲区是否有数据
        return;
    data = UDR0;
    // 读取 USART 的数据寄存器
    if(data == 0xfb)
        Frame_count = 0;
    else

```

```

    {
        Frame_count++;
        Uart_data[UFrame_count] = data;
    }
    if((Frame_count >= 2) && (data == 0xfc))
    {
        if(Uart_data[1] == art_data[2])
            // 两次数据比较
        {
            if(Uart_data[1] == TB_Order)
                // 接受到的数据与自身存储的数据比较
            {
                Ls_OrderA = TB_Order;
                // 命令一致则打开编码开关
            }
        }
        Frame_count = 0;
        // 记数清零
    }
}

```

## 5 结束语

单片机技术是计算机技术、通信技术及电子技术有机结合的产物。目前，单片机控制系统已处于实用阶段，随着实践经验的积累，系统的性能也在不断提高。系统安全性的控制更是重中之重。本设计中采用双单片机结构提高了系统的安全性和可靠性，内嵌CAN控制器的AT90CAN128在硬件上更加简单，实用且降低成本，便于维护。软件上使用模块化设计，简单易懂的C语言编写，增强了可读性和可移植性。该系统的双单片机结构设计能满足实际应用中高安全性能环境的要求。

### 参考文献：

- [1] 李平均, 张江. 双单片机的应用[J]. 微电子学与计算机, 2004, 21 (4): 78-80.
- [2] ZPW-2000A 无线移频自动闭塞系统技术培训教材[R]. 北京: 北京铁路通信信号研究设计院, 2003.
- [3] 沈文, Eagle lee, 唐卫前. AVR单片机C语言开发入门指导[M]. 北京: 清华大学出版社, 2004.