

基于 MSP430 的多处理器通信技术研究

廖明燕, 崔凤新, 郑金吾

(中国石油大学信息与控制工程学院, 东营 257061)

摘 要: MSP430 系列单片机处理能力强大, 超低功耗设计, 功能高度集成, 适合众多的场合使用。但是在大型复杂的场合或者实时性要求较高的场合, 使用一个处理器处理所有的任务, 总是显得有些不足。提出用多片 MSP430 组成多处理器系统, 利用 MSP430 固有的 UART 硬件模块, 采用 MODBUS 通讯协议实现不同处理器之间的数据交换。

关键词: MSP430 单片机; 多处理器; UART 模块; MODBUS 协议

中图分类号: TP399 **文献标识码:** B **文章编号:** 1002-2279(2006)01-0086-03

The Communication Technology of Multiprocessor Based on MSP430

LIAO Ming-yan, CUI Feng-xin, ZHENG Jin-wu

(College of Information and Control Engineering, China University of Petroleum, Dongying 257061, China)

Abstract: In this paper, a multiprocessing system based on MSP430 is designed to enhance processing rate and control more external events. Considering the properties of MSP430 and the multiprocessor architecture, a UART communication method using ModBus protocol has been proposed.

Key words: MSP430; Multiprocessor; UART; ModBus

1 引 言

1.1 多处理器简介

在单片机多机系统中, 多处理器是指多个相同或者不同类型的单片机处理器协作处理一个系统中的任务, 他们之间具备一定的数据交换和协作处理能力, 共同完成一个系统化的工作。不同处理器之间可以采用数据交换方式、并行总线方式、串行总线方式进行通讯。其中, 数据交换方式又可以称为共享内存交换方式; 串行总线方式又可以分为单总线方式、集成电路之间的通讯方式和 UART 方式。

1.2 MSP430 简介

MSP430 系列单片机是 TI 公司推出的一种 16 位超低功耗的混合信号控制器。低电压供电、宽工作电压范围, 有正常工作模式和 4 种低功耗工作模式, 单片机可以方便地在各种工作模式之间切换, 用中断请求将 CPU 唤醒只要 6 μ s。MSP430 也具有非常高的集成度, 单片集成了多通道 12bit 的 A/D 转换、片内精密比较器、多个具有 PWM 功能的定时器、斜边 A/D 转换、片内 USART、看门狗定时器、片内数控振荡器 (DCO)、大量的 I/O 端口以及大容量的片内存储器, 单片可以满足绝大多数的应用需要。

MSP430 的超低功耗特性使其在电池供电、便携式设计的应用中表现出非常优良的特性, 在国内已经有了非常广泛的应用。

但是在大型复杂的场合或者实时性要求较高的场合, 使用一个处理器处理所有的业务, 总是显得有些不足。为此引入多个 MSP430 处理器协作工作的模式, 把复杂的功能和计算分解为若干子任务后分配给多个单片机去完成, 从而可以提高系统的实时性、可靠性和适用性。

2 技术要点

2.1 MSP430 多处理器组成

MSP430 系列中 F14X 和 F44X 系列集成有 64K 的 Flash ROM 和 2K 的 RAM, 在多数场合, 无须为每个处理器扩展 Flash ROM 也无须扩展 RAM, 采用共享内存的数据交换方式组成多处理器系统并非最佳选择。此外, 片内有两个串行通信接口 (USART)。可直接进行双机通信。可以把两组串行端口中的一组串行端口供内部通讯使用, 另一组串行端口供外部通讯使用, 采用串行总线中的串行通讯方式组成多处理器系统是比较理想的选择。

采用 UART 串行通讯模式组成主从结构多处理

器系统,建立一个主处理器和若干从处理器。在硬件结构中,涉及两组信号引脚串行发送信号 TXD 和串行接收信号 RXD,主处理器的 TXD 端与所有从处理器的 RXD 端相连,所有从处理器的 TXD 端与主处理器的 RXD 端相连,接收和发送操作采用双缓冲结构。主处理器通过 TXD 端发送出指令报文,传输到从处理器的 RXD 接收端,从处理器对指令报文进行解包并且对这个指令报文进行响应,同时,从处理

器的响应报文通过 TXD 发送到主处理器的 RXD 接收端,主处理器获取响应报文确认指令是否被正确执行。从这个结构上看,主处理器可以与任何从处理器进行通讯,任何从处理器也可以和主处理器进行通讯,但是从处理器与从处理器之间却不可以进行直接通讯。MSP430 多处理器拓扑结构图如图 1 所示。

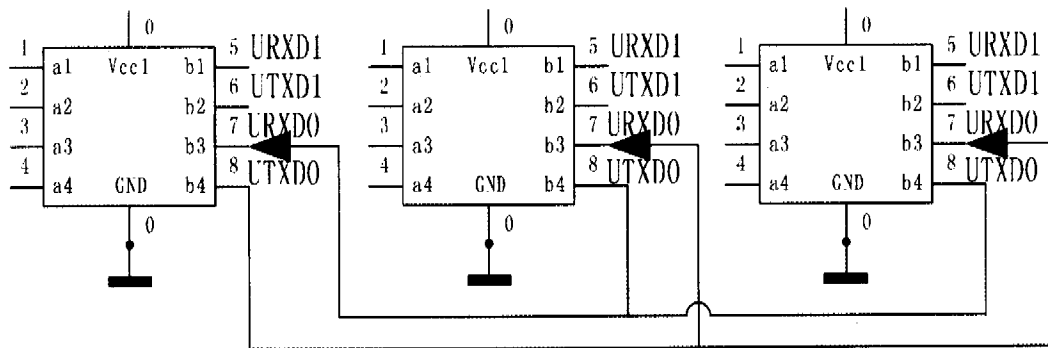


图 1 MSP430 多处理器拓扑结构图

在主从式多处理器系统中,要保证主处理器与所选择的从处理器实现可靠地通信,必须保证通信接口具有从处理器识别功能,因此采用地址位多机通讯格式,主处理器选择从处理器时发送一个地址识别编码,使每个从处理器能自动判别,完成相应的通讯和数据传输。

发送和接收由于有独立的中断向量,故都可以采用中断方式处理。发送当然可以采用数据查询方式,但在多任务系统中为保证系统实时性,采用中断方式更为合理。接收过程应和定时器配合,在接收开始后在定时器中断里面计算延时时间,在超出最长的接收等待延时时间后就可判定一次完整的传输过程结束,转而进行报文处理。发送是在发送中断处理程序中进行发送计数,一旦完成,立即关中断。

2.2 串行通讯协议

在串行通讯方式中,可以有若干种通讯协议进行选择,例如 ModBus、Brooks、工业总线协议等等。在此,我们可以选择业界通用的 ModBus 通讯协议作为处理器与处理器之间的通讯协议。

2.3 ModBus 通讯协议

ModBus 协议支持传统的 RS - 232、RS - 422、RS - 485 和以太网设备。ModBus 协议包括 ASCII、RTU、TCP 等报文格式,并没有规定物理层。此协议定义了控制器能够认识和使用的消息结构,而不管它们是经过何种网络进行通信的。ModBus 的 ASCII、RTU 协议规定了消息、数据的结构、命令和应答的方式,数据通讯采用 Master/Slave 方式,Master

端发出数据请求消息,Slave 端接收到正确消息后就可以发送数据到 Master 端以响应请求;Master 端也可以直接发消息修改 Slave 端的数据,实现双向读写。

2.4 ModBus 在多处理器系统中的报文格式

由于是一个主处理器对应多个从处理器的系统,处理器与处理器之间的报文传输必须明确标注目标地址和源地址,以免不相干的处理器之间进行误导操作。除此以外,为了提高主处理器的处理能力,同时避免不同处理器存在报文响应速度差异以及不同指令任务之间存在处理量差异的问题,必须采用异步通讯模式进行通讯。要满足异步通讯模式,必须在报文中对每次的通讯操作加注请求报文的 16 位标示,可以称之为句柄(Handle),同时对这些句柄进行记录。

2.4.1 数据读取请求报文格式

数据读取请求报文格式如表 1 所示。

表 1 数据读取请求报文格式

源地址	目标地址	句柄	功能代码
起始地址	字节数	校验码高字节	校验码低字节

源地址:主处理器地址;目标地址:指令目的地地址(从处理器);句柄:指令请求标示号;功能代码:操作指令代码;数据数量:操作所涉及的寄存器数量;数据 1... 数据 n:数据;校验码: CRC 校验码或者 LRC 校验码。

2.4.2 数据读取应答报文格式

数据读取应答报文格式如表 2 所示。

表 2 数据读取应答报文格式

源地址	目标地址	句柄	功能代码	数据字节数
数据 1	...	数据 n	校验码高字节	校验码低字节

源地址:从处理器地址;目标地址:主处理器地址;句柄:指令请求标示号;功能代码:操作指令代码;数据字节数:操作所涉及的寄存器字节数;数据 1...数据 n:数据;校验码:CRC 校验码或者 LRC 校验码。

其中,应答报文中的目标地址等价于请求报文中的源地址,应答报文中的源地址等价于请求报文中的目标地址。

2.4.3 数据操作请求报文格式

数据操作请求报文格式如表 3 所示。

表 3 数据操作请求报文格式

源地址	目标地址	句柄	功能代码	数据字节数
数据 1	...	数据 n	校验码高字节	校验码低字节

源地址:从处理器地址;目标地址:主处理器地址;句柄:指令请求标示号;功能代码:操作指令代码;数据字节数:操作所涉及的寄存器字节数;数据 1...数据 n:数据;校验码:CRC 校验码或者 LRC 校验码。

其中,应答报文中的目标地址等价于请求报文中的源地址,应答报文中的源地址等价于请求报文中的目标地址。

2.5 校验码

在 ModBus 中,通用的校验方式是 ASCII 协议方式采用 LRC 校验方式,RTU 协议方式采用 CRC 校验方式。

2.5.1 LRC 校验

LRC 校验比较简单,它在 ASCII 协议中使用,检测了消息域中除开始的冒号及结束的回车换行号外的内容。它仅仅是把每一个需要传输的数据按字节叠加后取反加 1 即可。下面是它的 VC 代码:

```

BYTE GetCheckCode (const char * pSendBuf, int nEnd)//获得校验码
{
    BYTE byLrc = 0;
    char pBuf[4];
    int nData = 0;
    for(i=1; i<end; i+=2) //i 初始为 1,避开“开始标记”冒号
    { //每两个需要发送的 ASCII 码转化为一个十六进制数
        pBuf[0] = pSendBuf[i];
        pBuf[1] = pSendBuf[i+1];
        pBuf[2] = '\0';
        sscanf(pBuf, "%x", &nData);
    }
}

```

```

byLrc += nData; }
byLrc = ~ byLrc;
byLrc ++;
return byLrc;
}

```

2.5.2 CRC 校验

CRC 是先调入一值是全“1”的 16 位寄存器,然后调用一过程将消息中连续的 8 位字节给当前寄存器中的值进行处理。仅每个字符中的 8Bit 数据对 CRC 有效,起始位和停止位以及奇偶校验位均无效。

CRC 产生过程中,每个 8 位字符都单独和寄存器内容相或(OR),结果向最低有效位方向移动,最高有效位以 0 填充。LSB 被提取出来检测,如果 LSB 为 1,寄存器单独和预置的值或一下,如果 LSB 为 0,则不进行。整个过程要重复 8 次。在最后一位(第 8 位)完成后,下一个 8 位字节又单独和寄存器的当前值相或。最终寄存器中的值,是消息中所有的字节都执行之后的 CRC 值。

CRC 添加到消息中时,低字节先加入,然后是高字节。下面是它的 VC 代码:

```

WORD GetCheckCode (const char * pSendBuf, int nEnd)//获得校验码
{
    WORD wCrc = WORD(0xFFFF);
    for(int i=0; i<nEnd; i++) {
        wCrc ^= WORD(BYTE(pSendBuf[i]));
        for(int j=0; j<8; j++) {
            if(wCrc & 1) {
                wCrc >>= 1;
                wCrc ^= 0xA001; } else {
                wCrc >>= 1;
            }
        }
    }
    return wCrc;
}

```

3 报文示范

ModBus 包含 ASCII 和 RTU 两种报文格式,RTU 报文较短,但是没有边界定义,ASCII 报文较长,但是边界明了。在多处处理器通讯之中,由于通讯距离很短,干扰较小,因此可以选择较高的通讯速率。通讯速率提高了,报文长度较长对通讯的影响不大,因此可以选择 ASCII 报文格式进行通讯。ModBus 的 ASCII 读取请求报文格式如表 4 所示。ModBus 的 ASCII 读取应答报文格式如表 5 所示。

(下转第 91 页)

证数据的时间有效性,结构中记录了变量的上传时间,用户在使用实时数据时,通过比较上传时间和当前时间来判断实时数据的有效性。另外为实时数据添加标签属性来控制数据的读写权限许可,从而有效的避免了数据的冲突。这样的实时数据库已经初步具备专用的实时数据库的基本特征和属性:①结构化的,可靠的数据集合;②基本的数据访问功能;③及时有效的数据更新。所以监控系统实时数据库属于实时数据库的一种,但和专用的实时数据库相比,它还没有严格的具备实时数据库在理论上的所有模型特征。从某种意义上讲,监控系统实时数据库只是以适合监控系统要求,最大限度地保证数据的实时性更新为基本目标。

实时数据库服务器选型时应考虑如下因素:

(1)数据存储效率。不同的系统采用不同的压缩算法,因此存储效率也不相同。

(2)实时数据库系统支持的最大点数和采集数据的精度。

(3)系统访问结构。传统数据库一般采用 C/S 客户/服务器)结构,考虑到网络技术的发展,未来的数据库访问模式必然向 B/S 三层结构发展。

(4)操作系统。实时/历史数据库一般基于 Windows NT 平台或 UNIX 平台,从操作维护管理角度出发,应考虑与全厂其他服务器操作系统的一致性。

(5)数据安全性。为防止潜在的数据丢失或被非法访问和操作的危险,实时数据库应具有良好的安全性策略。

4 结束语

基于目前国内火电厂烟气排放的实际状况,国家对火电厂大气污染物排放及监测要求将日益严格,烟气排放连续监测系统(CEMS)在火电厂的应用发展也必将是大势所趋,实际的需要将会使更多更好的 CEMS 出现。

参考文献:

[1] 王源泉. 烟气排放在线监测系统结构设计的改进[J]. 中国电力,2002;45-48.
 [2] 刘钊. 电力监控组态系统的一种设计[J]. 计算机应用,2002;72-76.
 [3] Kristin. Department of Environment Implementation Plan for the Continuous Emission Monitoring System[J]. 2004; 154-156.

(上接第 88 页)

表 4 ModBus 的 ASCII 读取请求报文格式

源地址	目标地址	句柄	功能代码
起始地址	字节数	校验码高字节	校验码低字节

表 5 ModBus 的 ASCII 读取应答报文格式

报文头	源地址	目标地址	句柄	功能代码	数据字节数
数据 1	...	数据 n	校验码高字节	校验码低字节	结束标志

假设主机地址 01,要对从机地址 02 进行读取 247 和 248 地址的两个寄存器值的通讯,并且本次通讯为第一次通讯,设定流水号为 0001。请求报文格式范本如表 6 所示。

:010200010300F6000201/r/n

表 6 请求报文格式范本事例

;	报文头
01	源地址
02	目标地址
0001	句柄
03	功能代码
00F6	地址
0002	数据字节数
01	LRC 校验
/r/n	结束标志

假设反馈回来的 247 数据为 5D,反馈回来 248 的数据为 3C。应答报文格式范本如表 7 所示。

:020100010300025D3C5E/r/n

表 7 应答报文格式范本事例

;	报文头
02	源地址
01	目标地址
0001	句柄
03	功能代码
0002	数据字节数
5D3C	数据
5E	LRC 校验
/r/n	结束标志

4 结束语

试验证明,利用 UART 模块,采用 MODBUS 的通讯协议可以实现多片处理器之间的通信和数据传输。用多片 430 单片机组成多处理器系统可以增强系统的实时处理能力和应用范围。结构灵活性强,便于扩展,同时减轻了主 CPU 负担,提高了系统的可靠性。

参考文献:

[1] 魏小龙. MSP430 系列单片机接口技术及系统设计实例[M]. 北京:北京航空航天大学出版社,2002-11.
 [2] 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,2003-01.
 [3] 李正军. 现场总线及其应用技术[M]. 北京:机械工业出版社,2005-01.