

面向小型全自动生化分析仪的 同步可靠的 UART 通信 *

张洪辉^{1,2}, 黎海文¹, 高春海^{1,2}, 吴一辉¹, 宣明¹

(1.中国科学院长春光学精密机械与物理研究所 应用光学国家重点实验室,吉林 长春 130033;

2. 中国科学院研究生院, 北京 100039)

摘要: 为了在小型全自动生化分析控制系统中实现双单片机之间同步、可靠的数据传输,提出了一种基于握手等待协议的同步可靠的 UART 通信系统方案。采用 3 次等待握手协议实现通信的同步,在数据传输过程中加入同步数据改进传输应用的稳定性。目前该方案已成功应用于小型全自动生化分析仪控制系统的数据传输,运行情况良好,性能稳定可靠。

关键词: C8051F060; UART 通信; 三次握手

中图分类号: TP368.1

文献标识码: A

The synchronous and reliable UART communication applied for the automatic biochemistry analyzer

ZHANG Hong Hui^{1,2}, LI Hai Wen¹, GAO Chun Hai^{1,2}, WU Yi Hui¹, XUAN Ming¹

(1. Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;

2. Graduate Institute of Chinese Academy of Sciences, Beijing 100039, China)

Abstract: In order to realize synchronous and reliable UART communication, a new system was introduced by this paper. The communication was established by three-way handshaking, and flag data was added during the communication in order to improve the reliability and stability. The experiments results showed the communication system worked stably and reliably.

Key words: C8051F060; UART communication; three-way handshaking

为了提高小型全自动生化分析仪的工作效率,其控制系统需采取并行处理的策略,以单个单片机为核心的单处理器因其功能的局限性,已不能完成多任务的并行处理。采用双单片机的控制方案可以提高系统的性能,但必须为双机通信提供稳定可靠的方案。

在传统的 8 位、16 位及 32 位处理器中,一般都带有 UART 串行口。传统基于 UART 的数据通信有 2 种方式:查询方式和中断方式。查询方式下 CPU 的负担较重,浪费了处理器资源,不能够及时处理其他事件^[1];另外,因发送端与接收端不能同步,当数据需要接收而接收端未及时响应时,就导致数据的丢失,造成通信不可靠。中断方式可以在接收到或者发送完数据时产生中断,在中断服务进程中完成数据存储与处理。由于中断方式下 CPU 利用率较高,也能够实现发送与接收的同步^[1],因此在

CPU 任务简单的系统中,一般采用中断方式实现 UART 数据通信。但是小型全自动化生化分析仪控制系统需要实现多电机位置的闭环控制、多传感器的数据采集及处理、病人资料的传输处理等任务,采用中断方式的 UART 通信时,一方面带来中断的优先级分配问题,当系统处于更高优先级的中断处理子程序中时,导致串行通信的中断不能及时被响应,造成数据丢失;另一方面为协调双处理器间各部件的协同工作,有时不允许程序的正常执行流程被打断,禁止中断响应。为了保证串口通信的可靠性与稳定性,在设计小型全自动生化分析仪的控制系统时,采用了基于 3 次等待握手协议的同步、可靠的 UART 通信。本文将分别从物理层、传输层与应用层介绍等待握手协议在同步可靠的 UART 通信中的实现。

1 物理层

物理层实现通信底层的物理连接,为数据端设备提供传送数据的通路,其作用是确保比特流能在物理信道

* 基金项目: 国家 863 高技术研究发展计划资助项目(2006AA04Z367);
国家 863 重点资助项目(2007AA042102)

上传输^[2]。物理层的连接由单片机的异步通信端口(UART 串口)实现。

1.1 异步通信简介^[3-4]

在异步通信中,数据是以字符为单位组成字符帧进行传送的。发送端与接收端分别按字符帧规定的格式和波特率来协调数据的发送和接收。字符帧和波特率可以由用户根据实际情况选择。

字符帧由起始位、数据位、停止位和空闲位 4 部分组成,如图 1 所示。

空闲位	起始位	数据位										停止位	空闲位
11...1	0	D0	D1	D2	D3	D4	D5	D6	D7	I/O	1	11...1	

图 1 异步通信字符帧格式

1.2 C8051F060 的 UART

C8051F060 是 SiLab 公司推出的混合信号系统芯片,是高度集成的片上系统,具有高速指令处理能力^[3]。C8051F060 有 2 个 UART 口,并没有分配在固定的 I/O 端口上,而是需要通过优先权交叉开关译码器进行分配。“交叉开关”按优先权顺序将端口 0~3 的引脚分配给器件上的数字外设。C8051F060 的 UART 串口与端口的连接有多种形式,灵活可靠。

2 传输层

传输层实现数据流的传输,并负责传输控制,具备流量控制的功能。其目的是为应用层提供可靠无误的数据传输。传输层的服务要经历传输连接建立阶段和数据传送阶段^[2]。该层由软硬件结合实现。硬件实现传输线路与控制线路的连接,软件对其进行设置和控制。

2.1 传输层的硬件电路设置

传输层的数据传输线路为物理层的 UART 串口,其传输连接的建立、传输流量的控制则通过单片机的另外 2 个独立 I/O 口,用于实现 3 次等待握手协议。电路设计的原理为:发送端单片机 UART 口的 TX0 和 RX0 分别连接到接收端单片机的 UART 口的 RX0 和 TX0。在每 1 个单片机上分别使用 2 个 I/O 端口用于握手信号的发送和接收。其电气连接如图 2 所示。

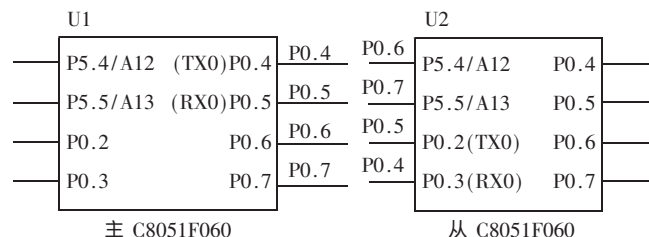


图 2 UART 实时通信的电路连接

2.2 传输层的软件设计原理

为实现数据通信的同步和提高通信的可靠性,传输

层采用 3 次等待握手协议实现数据的实时同步通信。协议的原理如图 3 所示。

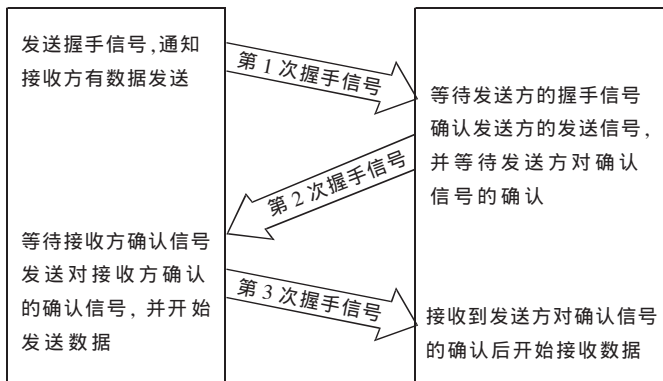


图 3 传输层的 3 次握手协议

协议的原理如下:

(1)当发送端要发送数据时,发出第 1 次握手信号,通知接收端,第 1 次握手信号始终有效,直到收到接收端对第 1 次握手信号的应答信号,即第 2 次握手信号。当发送端接收到第 2 次握手信号后,使第 1 次握手信号无效,发出对第 2 次握手信号的应答信号,即第 3 次握手信号,并随即发送数据。至此,3 次握手协议完成,发送端与接收端实现了数据的同步传输。

(2)当接收端要接收数据时,首先等待发送端发出的第 1 次握手信号,当发送端发送第 1 次握手信号后,表明发送端已经准备好要发送数据,接收端将第 2 次握手信号持续一段时间后清除,等待第 3 次握手信号。检测到第 3 次握手信号后,随即开始接收数据。至此,3 次握手协议完成,发送端与接收端实现了数据的同步传输。

2.3 传输层的软件实现

传输层的软件与硬件电路密切相关,传输层的软件质量的好坏决定了最终应用层软件的质量。在传输层的软件实现中,与硬件相关的系统初始化尤为重要。系统上电后,首先要进行的就是初始化,初始化包括系统时钟初始化、EMIF 初始化、定时器初始化、端口初始化以及交叉开关的配置等。可以通过设置相应的特殊功能寄存器(SFR)来进行。由于篇幅所限,这里主要介绍关键初始化模块:端口初始化与 UART 初始化。系统的初始化分为主机的初始化与从机的初始化部分。

2.3.1 端口初始化

端口初始化的关键是合理地分配 C8051F060 的数字外设,使其依据硬件电路的设计,对交叉开关进行配置,并设置相应的端口输入输出类型。主机端口初始化中交叉开关配置部分的程序如下:

```
void MasterPORTInit (void)
{
    OldSFRPAGE = SFRPAGE;
    SFRPAGE = CONFIG_PAGE;           //切换页面地址
    XBRO = 0x02;                      //SPI0 连接到端口
```

```

//脚 P0.0~P0.3
XBR1 = 0x00;
XBR2 = 0x44;           //弱上拉全局禁止,交叉
                        //开关使能, UART 连接到 P0.4~P0.5
SFRPAGE = OldSFRPAGE;   //恢复页面地址
}

```

从机的端口初始化与主机的端口初始化原理相同,这里不再累述。

2.3.2 UART 初始化

UART 初始化主要包括时钟源的选择、工作方式的选择,逻辑电平的选择及处理中断的方式等,其中主机的 UART 初始化程序如下:

```

void MasterUARTInit(void)
{
    OldSFRPAGE = SFRPAGE;
    SFRPAGE = UART_PAGE;           //切换页面地址
    SCON1 = 0x40;                  //波特率可编程的 8 位 UART,
                                    //停止位的逻辑电平被忽略,
                                    //禁止接收,清空发送及接收中断标志
    SFRPAGE = OldSFRPAGE;         //恢复页面地址
}

```

传输层数据传输软件实现元数据传输的功能,在这里实现传输层的 3 次握手协议。依据 3 次握手协议的设计原理,主机发送软件的实现如下:

```

{
    while(P0_6==1)                //等待第 2 次握手信号
        P0_7=0;                  //发出第 1 次握手信号
    Delay1us(1);                  //延时 1 μs
    P0_7=1;                      //发送第 3 次握手信号
}

```

工作时序为:当主机需要发送数据时,首先将 P0_7 设置为低,即为第 1 次握手信号,从机检测到该信号后,知道主机将发送数据,于是发出第 2 次握手信号,将与主机 P0_6 相连的引脚置低。主机检测到 P0_6 为低电平,即第 2 次握手信号后,将发出第 3 次握手信号,即把 P0_7 设置为高电平,并随即发送数据。

从机接收数据时的软件实现如下:

```

{
    while(P5_5==1)                //等待的第 1 次握手信号
        P5_4=1;
    P5_4=0;                      //发出第 2 次握手信号
    Delay1us(1);
    P5_4=1;
}

```

从机的 P5_4、P5_5 分别连接主机的 P0_6、P0_7。当从机接收数据时,首先等待主机发送的第 1 次握手信号,即主机的 P0_7 端口(从机的 P5_5 端口)为低电平。若检测到第 1 次握手信号,则发送第 2 次握手信号(即将 P5_4 置低),随即将 P5_4 拉高,检测到第 3 次握手信

号后开始接收数据。为实现软件的简便性,减少编程出错的概率,本文定义了 2 个宏来完成上述握手的过程。宏定义如下:

```

#define Send while(P0_6==1)P0_7=0;
\Delay1us(1); P0_7=1;           //主机发送的宏定义
#define Receive OldSFRPAGE=SFRPAGE;
\SFRPAGE=CONFIG_PAGE;while(P5_5==1)P5_4=1;
\P5_4=0;Delay1us(1);P5_4=1;
\SFRPAGE=OldSFRPAGE;           //从机接收的宏定义
主机发送与从机接收元数据的函数分别如下:

void MasterSendUART (Uchar sdata)
{
    Send;                        //完成握手协议
    OldSFRPAGE = SFRPAGE;       //寄存器页面切换
    SFRPAGE = UART_PAGE;
    TI1 = 0;                    //清除发送完成中断标志
    SBUF1 = sdata;              //发送数据
    while(TI1==0);              //等待发送完成
    TI1 = 0;                    //清空发送完成标志
    SFRPAGE = OldSFRPAGE; }
    Uchar SlaverReceiveUART(void)
{

```

```

    Receive;                    //完成握手协议
    OldSFRPAGE = SFRPAGE;
    SFRPAGE = UART_PAGE;       //寄存器页面切换
    SCON1 = 0x50;              //允许接收,清空接收完标志
    while((SCON1&0x01)==0);    //等待接收完成
    ReData = SBUF1;            //接收数据
    SCON1 = 0x40;              //清空接收及发送完成标志位
    SFRPAGE = OldSFRPAGE;
    return ReData;             //返回接收到的数据
}

```

3 应用层

应用层软件主要为实际应用服务,实现系统的应用需求。本文所设计的通信方案是为小型全自动化的生化分析仪控制系统的数据通信服务的,应用程序包括生化检测结束后病人检测结果的交互,以及双机对各功能器件的控制指令等,下面简要介绍应用层软件的流程。

应用层软件完成数据的发送和接收,应用层数据接收软件将实现数据的完整接收与存储,发送软件完成完整的数据发送。小型全自动生化分析仪控制系统的通信内容有结果数据和控制指令。为进一步保障数据传输的可靠性与同步性,在应用层的数据传输中加入了同步数据。应用层函数完成有关数据接收的处理,首先判断接收到的数据类型,并依次调用相应的软件处理模块来对接收后数据的存储。图 4 为应用层的接收端软件的执行流程。

4 结果

本文在小型全自动生化分析仪控制系统的通信中分
《电子技术应用》2010 年第 1 期

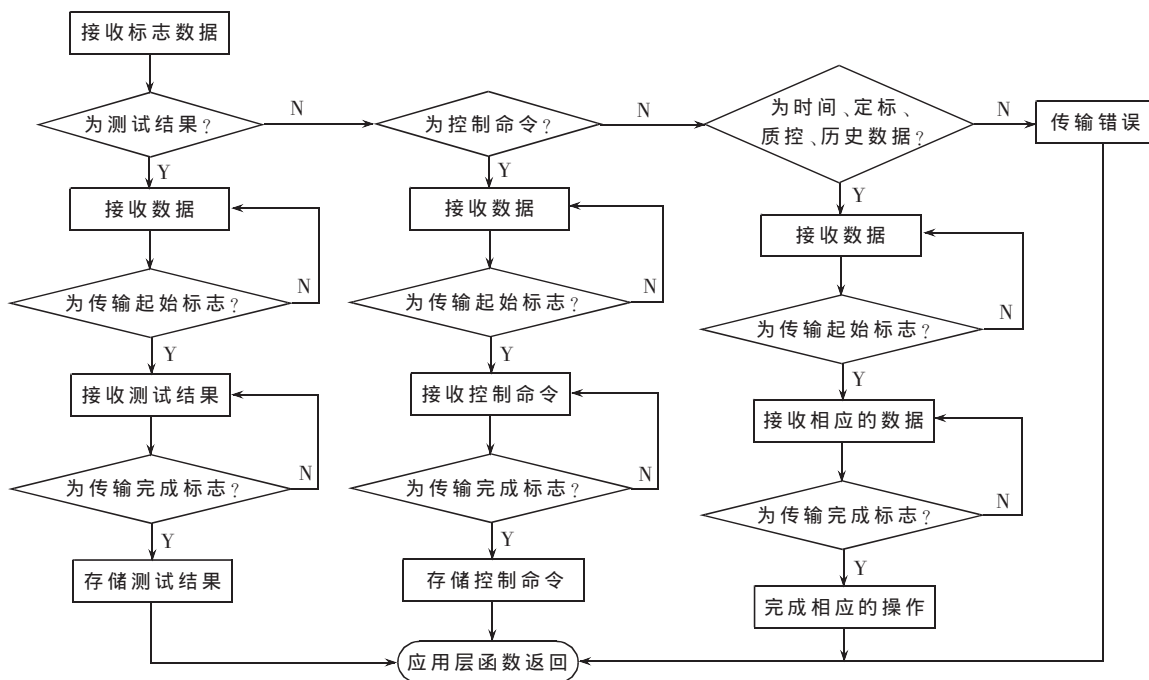


图4 应用层接收端的软件流程

别使用基于3次等待握手协议的UART通信与基于查询式的UART通信方案并进行了对比,其结果如表1所示。当系统使用基于查询式的UART通信时,因双处理器未能同步工作,导致了接收端未能及时查询及存储数据,造成了数据的丢失。而通信建立时的等待握手协议则提高了通信的稳定性与可靠性。

表1 查询方式与等待握手UART传输应用于生化分析仪控制系统通信结果对比

实验	第1组	第2组
分析仪控制系统通信采用查询式UART导致的数据帧丢失	3	5
分析仪控制系统通信采用握手等待UART导致的数据丢失	0	0

本文提出的基于3次等待握手协议的UART传输通信同步可靠,应用于小型全自动生化分析仪的控制系统通信时,解决了查询方式下UART因接收与发送端未同步造成数据帧丢失及中断方式打断正常程序流程执行的问题,取得了令人满意的效果。

参考文献

- [1] 杨福广,李贻斌,尹占芳,等.ARM中基于DMA的高效UART通信及其应用[J].微计算机信息,2008(1-2).
- [2] TANENBAUM A S. AMSTERDAM U V. The netherlands computer networks,forth editI/On[M].北京:清华大学出版社,2004.
- [3] 新华龙电子有限公司.C8051F060混合信号ISP FLASH微控制器数据手册.潘琢金,译.2004.

- [4] 王晓剑,潘顺良,沈为群,等.TMS320DM642中利用McBSP与EDMA实现UART[J].电子测量技术,2008(2).
- [5] Cygnal Integrated Products, Inc. C8051F单片机应用解析[M].潘琢金,译.北京:北京航空航天大学出版社,2002.
- [6] 何建标,王宏远,林静宜,等.基于最佳接收的UART的设计与实现[J].电子技术应用,2006,34(8).

(收稿日期:2009-07-08)

作者简介:

张洪辉,男,1985年生,硕士生,主要研究方向:嵌入式系统应用。

黎海文,男,1976年生,博士、副研究员,主要研究方向:微光机电系统。

高春海,男,1985年生,博士,主要研究方向:精密机械设计。

欢迎订阅

微型机与应用

半月刊

订阅代号:82-417