

基于 ARM 和 W5100 的嵌入式以太网通信接口设计*

王广维, 张浩然

(浙江师范大学 数理与信息工程学院, 浙江 金华 321004)

摘要: 设计了一种基于以太网控制器 W5100 和 32 bit ARM 微处理器 STM32F105V 的嵌入式以太网通信接口。介绍了 W5100 的工作原理、接口硬件设计及软件设计方案。硬件部分采用 SPI 总线接口模式实现 STM32F105V 与 W5100 连接, 软件部分着重对网络控制器的 4 个独立端口编程, 实现以太网数据快速收发功能。本接口设计可直接应用于工业数据采集系统开发。

关键词: W5100; STM32F105V; 以太网; 嵌入式

中图分类号: TP311

文献标识码: B

文章编号: 1674-7720(2011)05-0050-04

Design of embedded Ethernet communication interface based on ARM and W5100

Wang Guangwei, Zhang Haoran

(College of Mathematics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China)

Abstract: The paper describes a design of embedded Ethernet communication interface based on W5100 and 32-bit ARM processor STM32F105V. The paper presents the work principle of W5100, the hardware and software design of the interface in detail. In the part of hardware, SPI bus interface mode is used to achieve the connection between STM32F105V and W5100. In the part of software, it emphasizes four independent socket of network controller programming to achieve fast Ethernet transceiver functions. The interface design can be directly applied to industrial data acquisition system.

Key words: W5100; STM32F105V; Ethernet; embedded

随着以太网技术在工业远程监控和数据采集领域的飞速发展, 基于以太网的应用接口设备也迅速普及到医疗设备、工厂电力系统等应用。然而, 嵌入式以太网接口设计的核心器件是网络控制芯片, 其性能优劣直接影响网络通信的效率。

本文结合 ARM 技术, 采用 32 bit STM32F105V 微处理器和高性能的以太网控制芯片 W5100 实现高性能、高可靠性的嵌入式以太网通信接口设计, 其系统各功能模块容易扩展和升级。

1 总体设计

在嵌入式系统设计高速发展的过程中, 与传统的基于现场总线方式相比, 结合嵌入式系统和以太网技术来实现数据采集和控制功能越来越受到广大嵌入式设计者的青睐, 而通信接口设计实质是能够实现 TCP/IP 网络通信协议。使用本接口模块的核心控制器 W5100, 应

* 基金项目: 浙江省大学生科技创新活动计划(新苗人才计划)项目(2010R404066)

用程序设计者无需深入了解 TCP/IP 协议, 也无需考虑以太网的控制, 只需访问网络控制器的寄存器, 并灵活创建和选择 TCP 及 UDP 套接字(Socket)函数就可以简单地实现网络通信, 且不需要操作系统的支持, 其具有硬件电路简单、编程方便等特点, 解决了一般嵌入式设计的软件设计复杂、网络编程工作量大等问题, 再结合 32 bit 高性能 ARM 处理器以真正实现以太网的高速实时传输。ARM 完成对应用程序的处理, W5100 实现数据传输和通信网络协议的处理。

具体实现过程: 数据信号或模拟信号可以通过 STM32F105V 丰富的外设接口(RS232 总线、CAN 总线等)输入或直接通过外部数据总线输入, 并且可以用 ARM 处理器对传输的数据和信号做一些预处理工作, 然后传输到 W5100 芯片完成网络协议的处理, 再通过网络接口传输到远程终端(PC 机)。相反, 远程终端可以通过以太网发出控制指令, 将传输信号发送至 ARM 或数据输

网络与通信

Network and Communication

出端,从而实现嵌入式系统中网络数据的采集和传输控制。系统框架如图1所示。

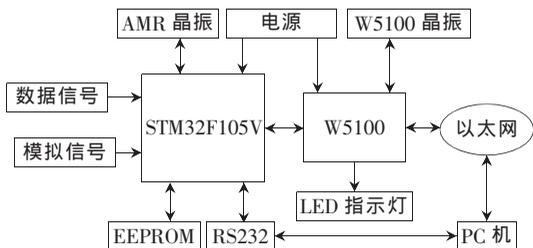


图1 系统框架图

本设计利用JTAG接口烧写程序到ARM内部的Flash,外扩I²C接口的128×8(1 Kbits)EEPROM,基地址为0x40005400,用来存储网络IP地址、端口号、子网掩码等网络信息,网络传输状态指示灯(LED)反映了数据传输的实时状态。电源由外部电源提供,分别经过MC7805和AS1117芯片稳压转换输出5V和3.3V的电压,这样就能够很好地满足内核、外设以及外部电路的供电^[1]。

2 网络控制器 W5100

W5100是WIZnet公司最新推出的固件网络芯片,内部集成了10/100 Mb/s以太网控制器,支持自动应答(全双工/半双工模式),最高通信速率达25 Mb/s。W5100将TCP/IP协议栈、以太网MAC和PHY三种功能集成于一体,支持硬件化的TCP、UDP、ICMP、IPv4、ARP、ICMP、PPPoE、以太网等协议,并提供了多种总线接口方式(直接总线接口、间接总线接口、SPI总线)便于连接各类单片机,可以满足不同应用场合的需求。

W5100内含公共存储器、端口存储器、发送存储器以及接收存储器。公共寄存器用来设置W5100的工作模式、中断向量、IP地址、网关地址、子网掩码、物理地址、超时值等相关信息;端口寄存器平均分为4个相等的存储器大小,可以单独对4个独立的网络通道(Socket)设置端口的通信模式(TCP客户端模式、TCP服务器模式、UDP模式),实现网络数据的通信;内部16 KB存储器的发送和接收数据缓冲区(8 KB的发送缓存和8 KB接收缓存区)用来存放临时数据。W5100支持ADSL连接(支持PPPoE协议、带PAP/CHAP验证),支持自动极性变换(MDI/MDIX)并附带有多种LED指示灯输出,时钟信号由外部晶振25 MHz提供,分别接入XTLP、XTLN管脚。

W5100支持80脚的LQFP小型封装,0.18 μm CMOS工艺,符合环保要求。3.3 V的单电源工作电压,I/O口可承受5 V电压,可以满足低功耗要求^[2]。

3 通信接口硬件设计

本设计采用嵌入式微处理器和以太网控制器的方式实现接口转换,需要用到的芯片主要包括STM32F105V微处理器,W5100网络控制器及带网络变压器的RJ-45接口13F-60系列,AT24LC01B EEPROM等。采用SPI总线接口模式,实现STM32F105V和网络控制器的硬件连接。在串

行接口模式下,只需要连接几个简单的引脚就可以进行数据通信^[3]。SPI总线接口模式下硬件连接如图2所示。

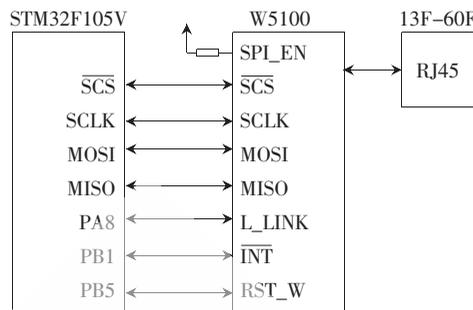


图2 STM32F105V与W5100连接图

本系统微处理器采用ST公司的STM32F105V,并采用专门设计的Cortex-M3内核,时钟频率可达72 MHz,256 KB Flash存放程序或数据,64 KB的RAM存放数据,多达80个I/O接口可以映射到20个外部中断。STM32F105V与W5100供电电压都是3.3 V,所以可以直接连接。

配置W5100的SEN引脚必须通过一个10 kΩ的电阻接高电平,以选择W5100的SPI接口模式,置STM32F105V为SPI主模式,W5100为SPI从模式,在SPI总线模式下的配置复用输入输出接口(AFIO)的PA5、PA6、PA4、PA7分别为SCLK、MISO、/SCS(SPI从模式选择,低电平有效)、MOSI信号线直接相连,并用软件设置管脚的输出最大速度为10 MHz,STM32F105V通过SPI对W5100读写操作。

配置PB5口作为W5100的复位信号/RST_W,低电平有效。为实现准确的硬件复位,复位引脚RST_W上复位信号至少保持2 μs。

配置PB1口为外部中断线1(EXTI1),并作为W5100的中断信号输出端/INT,低电平有效。当W5100在端口产生连接、断开、接收数据、数据发送完成以及通信超时等条件下,该引脚输出低电平信号指示微处理器。

配置PA8口为W5100输出的以太网物理层信号灯(LINKLED)指示W5100的网络连接状态,通过上拉电阻输入到微处理器,低电平有效。

此外,为缩小接口设计的面积,本设计采用10/100 Mb/s的13F-60系列带网络变压器的RJ45接口,W5100的RXIP接RJ45的RD+,RXIN接RD-,并带有左右两个状态灯,连接状态灯(LINKLED)和动态指示灯显示接收与发送状态(ACT_LED),通过一个高速开关二极管共阳极(开关速度最大值4 ns,重复反向峰值耐压最大值75 V)与W5100的RXLED/TXLED管脚相连,供电电压为3.3 V,直接与W5100相连。差分接收和发送引脚分别通过两个75 Ω的电阻和0.001 μF的电容接地。RJ45接口的内部结构图如图3所示。

4 通信接口的软件设计

4.1 初始化程序设计

本系统初始化通过定义结构体的方式完成STM32F105V

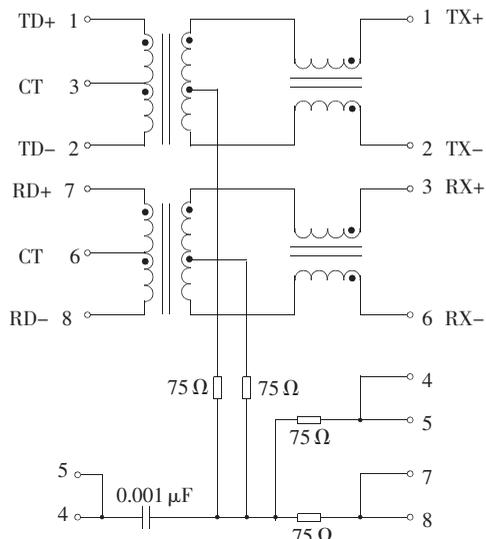


图3 13F-60F系列内部接口图

微处理器和W5100的初始化^[4-5]。

微处理器完成系统时钟、外设时钟、系统启动模式、嵌入式向量中断控制寄存器、I²C、通用输入输出接口、通用异步接收发送器、通用定时器以及SPI等的初始化。

W5100的初始化主要设置一些关键的寄存器：

(1) 设置模式寄存器(MR)bit[7](软件复位位)为1,初始化芯片内部寄存器,复位后自动清0。

(2) 设置中断屏蔽寄存器(IMR)为0xFF(屏蔽中断源),启动IP地址冲突异常中断和端口n寄存器(Sn_INT)中断等,通过向相应的中断屏蔽位写1,任何时候只要中断寄存器(IR)对应的位也置1,则中断将产生,CPU通过访问IR获得中断来源。

(3) 设置重发时间寄存器(RTR)为200ms(0x07D0),当发出连接、断开等命令而没有收到远端对端的响应或响应延迟时,产生重发过程。

(4) 配置重发计数寄存器(RCR)为8,设定重发的次数。

(5) 设置接收缓冲区的大小寄存器(RMSR)和发送存储空间大小寄存器(TMSR)都为0x55,每个端口接口和发送存储空间分别分配2KB的存储空间。

(6) 设置端口n模式寄存器(Sn_MR)为0xA1,启动广播功能,设置端口n协议类型为TCP模式。

(7) 设置端口n命令寄存器(Sn_CR),端口的初始化、建立/断开连接以及数据传输等。

(8) 设置端口n的最大分片长度寄存器(Sn_MSS)为1460。

初始化完成后,根据SPI协议编写发送字节函数SPI_SendByte(),配合Read_W5100和Write_W5100完成字节的读取和发送,这里需要定义读操作码(0xF0)和写操作码(0x0F),实现微处理器与W5100数据通信。

4.2 Socket 初始化程序设计

W5100与终端之间的数据交换有多种通信方式,本

文采用基于TCP模式的通信方式。TCP是以连接为基础的通信方式,端口n在进行数据通信时,必须先建立连接。TCP有两种建立连接方式,一种是服务器模式(被动模式),需要等待连接请求;另一种是客户端模式(主动打开),需要发送连接请求给服务器。本设计配置W5100为服务器模式,只需对W5100的Socket进行配置就可以完成网络数据的收发和启动功能。

当Socket作为服务器模式时,初始化端口需要设置运行模式(Sn_MR)和本机端口号(Sn_Port),并在端口命令寄存器打开(OPEN)端口。引用Socket_Listen(SOCKET s)程序,只调用一次该程序就可使W5100设置为服务器模式。主要程序如下所示。

```
Write_W5100((W5100_S0_MR+s*0x100),S_MR_TCP);
//设置Socket为TCP模式
Write_W5100((W5100_S0_CR+s*0x100),S_CR_OPEN);
//打开Socket
Write_W5100((W5100_S0_CR+s*0x100),S_CR_LISTEN);
//设置Socket为侦听模式
Write_W5100((W5100_S0_CR+s*0x100),S_CR_CLOSE);
//关闭Socket
```

完成Socket的打开和设置侦听工作后,至于远程客户端是否与其连接,则需要等待Socket中断,在服务器侦听模式下,不需要设置目的IP和目的端口号。

W5100在TCP服务器模式下的处理流程如图4所示。

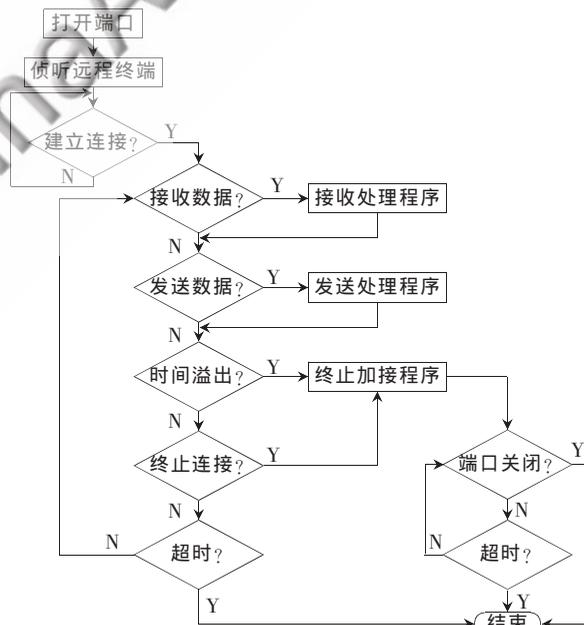


图4 W5100在TCP服务器模式下的处理流程图

4.3 中断处理程序设计

本设计采用中断方式来处理数据包的接收和发送,以提高效率。在W5100处理中断的过程中,微处理器首先通过访问W5100的中断寄存器(IR)获得产生中断的来源。中断寄存器与中断屏蔽寄存器配合使用,且位

网络与通信 Network and Communication

是一一对应的,中断屏蔽寄存器(IMR)的相应位可屏蔽中断寄存器中任何中断源。因此,若要使用某个中断源,先要将该中断源在中断屏蔽寄存器中的相应位置位,以打开所需的中断源,这样,中断才会产生。当中断产生后,即进入中断处理程序。本设计中的中断源主要包括端口 n 中断事件。一般设置有 Socket 成功连接(S_IR_CON)、断开连接(S_IR_DISCON)、数据发送完成(S_IR_SENDOK)、接收到数据(S_IR_RECV)或传输超时(S_IR_TIMEOUT)等事件中中断。

本文以端口 0 接收到数据包后的产生一个接收数据中断为例说明中断处理过程。首先,在使用端口 0 中断之前,应在初始化程序中将中断屏蔽寄存器(IMR)中的端口 0 置位 (IMR_S0_INT),当 Socket0 发生中断时,IMR_S0_INT=1、IR_S0_INT=1、动态 LED 灯亮显示接收状态,此时,W5100 的中断输出管脚(/INT)输出低电平,以通知微处理器有中断产生,微处理器访问中断寄存器获取中断源为接收数据中断;然后进入中断处理函数,启动发送函数(S_rx_process)。

4.4 Socket 数据接收程序设计

当端口接收数据时,产生接收中断。首先调用端口接收数据包函数 Process_Socket_Data(),并对接收到的数据类型进行判断和加工。本过程先调用接收函数 S_rx_process () 从 W5100 端口的接收数据缓冲区读取数据,然后将读取的数据加上接收存储器读指针寄存器(SO_RX_RD)的值再写入 SO_RX_RD,最后将 RECV 命令重新写入端口 0 的命令寄存器(SO_CR),以等待下次数据的接收。或者将处理完的数据拷贝到发送缓冲区,再调用 S_tx_process 函数发送数据包给 CPU。主要程序如下:

```
rx_size=Read_W5100(W5100_S0_RX_RSR+s*0x100)
//读取接收数据的字节数,
//接收的字节为 2 个字节数 32 bit,分两次读
rx_size*=256
rx_size+=Read_W5100(W5100_S0_RX_RSR+s*0x100+1)
rx_offset=Read_W5100(W5100_S0_RX_RR+s*0x100)
//读取接收缓冲区的偏移量
rx_offset*=256
rx_offset+=Read_W5100(W5100_S0_RX_RR+s*0x100+1)
Rx_Buffer[i]=Read_W5100(j) //将数据缓存到接收
//缓冲区,j 为实际物理地址为 W5100_RX+rx_offset
Write_W5100((W5100_S0_CR+s*0x100),S_CR_RECV)
//设置接收命令,等下一次接收
其中通信数据包格式为:
```

0xaa	0x55	长度	命令	目标代码	数据
------	------	----	----	------	----

其中 0xaa 和 0x55 为接收数据包的标志头;长度位为数据包字节长度,不包括数据包头和本身字符,命令位为对对象数据的操作,为 0 时读取数据,为 1 时设置

对象数据,目标代码位用来显示对象代码,如 0x00 为网关 IP、0x01 为子网掩码、0x02 为物理地址、LED 为状态显示等;数据位为接收到的数据,数据是以 16 进制形式接收,最后再加上 2 B 的数据报头和 1 B 的数据本身。

4.5 Socket 数据发送程序设计

通过 Socket 发送数据时,调用发送数据函数 S_tx_process。首先把要发送的数据缓存在发送缓冲区(Tx_buffer)中。此外,在发送数据时,需先检查发送缓存区的剩余空间的大小(Sn_TX_FSR),控制发送数据的字节数,如用以太网协议发送的数据最大传送单元(MTU)不超过 1 500 B。在 TCP 服务器模式下,在数据发送处理过程中,可不设置目标主机的 IP 和端口号。剩余空间的大小因写入数据的增加而减少,数据发送后又自动增加。当发送缓冲区的数据完全写入端口的发送数据缓存区后,则将数据本身长度加上端口传输写指针寄存器(Sn_TX_WR)中的值再写入 Sn_Tx_WR,再计算发送缓冲区的偏移量(tx_offset),用于指示发送数据的长度,最后启动发送(Sn_CR_SEND)。相关程序如下:

```
i=tx_offset/S_TX_SIZE //计算实际物理偏移值,
//S_TX_SIZE 由 TMSR 定义为 2 K
tx_offset=tx_offset-i*S_TX_SIZE //计算实际物理地址
//j= W5100_TX+s*S_TX_SIZE+tx_offset
Write_W5100(j,Tx_Buffer[i]) //将发送缓冲区中的
//数据写入到端口的发送缓冲区
Write_W5100((W5100_S0_CR+s*0x100),S_CR_SEND)
//启动发送的指令
```

本文以基于 ARM Cortex-M3 的微处理器 STM32F105V 为核心,结合 W5100 网络控制芯片实现了嵌入式以太网的连接,该接口设计具有硬件设计简单、成本低、集成高度以及软件开发周期短等优点,在自动化控制和数据传输领域有着广泛的市场和应用前景。

参考文献

- [1] ST.STM32F105xx datasheet.2010.
- [2] WIZnet.W5100 datasheet version1.1.8. 2009.
- [3] 姚冰,康世英,谢佳.嵌入式以太网接口硬件部分的设计与实现[J].微处理机,2008,29(2):155-157.
- [4] 成都加联科技有限公司.W5100 在嵌入式系统中实现 TCP/IP 网络通信的应用[R].2009.
- [5] 段海龙,彭辉俊,程健.基于 ARM 的嵌入式以太网通信的实现[J].机械与电子,2006(2):9-12.

(收稿日期:2010-11-05)

作者简介:

王广维,男,1986 年生,硕士研究生,主要研究方向:嵌入式系统与应用。

张浩然,男,1972 年生,副教授,硕士生导师,主要研究方向:现代电子技术。