

基于 ARM 的远程监控数据采集系统的设计与应用

李腾, 闫菲, 于志强, 张啸, 朱立鹏

(天津电气科学研究院有限公司, 天津 300301)

摘要: 设计并实现了一套远程监控数据采集系统, 通过 ARM+FPGA+ENC28J60 的系统架构, 完成数据采集、存储和网络通信功能, 并对 LWIP 协议栈进行了移植与改进, 实现了服务器对设备的实时监控。实践表明, 所提出的设计具有良好的可行性与实用性。

关键词: 远程监控; ARM 微处理器; LWIP 协议栈

中图分类号: TM461

文献标识码: A

DOI: 10.19457/j.1001-2095.dqcd19586

The Design and Application of Remote Monitor Data Acquisition System Based on ARM

LI Teng, YAN Fei, YU Zhiqiang, ZHANG Xiao, ZHU Lipeng

(Tianjin Research Institute of Electric Science Co., Ltd., Tianjin 300301, China)

Abstract: A remote monitoring data acquisition system was designed and implemented. Through the system architecture of "ARM+FPGA+ENC28J60", the functions of data acquisition, storage and network communication, and transplants were completed and the LWIP protocol stack was improved to realize the real time monitoring of the device by server. Practice shows that the design has good feasibility and practicability.

Key words: remote monitor; advanced RISC machines (ARM) microprocessor; LWIP protocol stack

在通信技术的引领下, 电控设备的网络化控制水平不断提高。对于传统电控设备行业来说, 因为设备的工作环境复杂且负载较大, 一旦发生故障将造成较大损失, 所以为了提高电控设备的工作性能, 实现电控设备的远程监控、远程维护, 为用户提供更优质的服务, 减少售后服务费用, 有必要引入通信与控制技术, 实现电控设备的嵌入式智能监控^[1]。

目前, 很多企业都建立了嵌入式智能监控系统。当前监控系统终端应用最广泛的方案一般有 2 种: 1) 采用 RS232, RS485 与上位机进行通信, 但是这种方案最高通信速率只有 120 kbps, 难以传输较大的数据量; 2) 采用短距离无线传输, 虽然能接入互联网, 也能实现大数据量传输, 但是无法承载长距离传输^[2]。这些方案显然已经无法满足现在的要求, 因此需要开发一种新的高效远程监控系统以满足长距离大数据量传输和接入网络的需求。本文基于 ARM 设计了一种嵌入

式远程监控系统, 不仅能实现设备工作状态的快速采集与传输, 也有助于平台对设备实现管理和维护。

1 系统硬件平台

远程监控数据采集系统硬件平台的主体由 ARM, FPGA 和以太网控制器组成, 它们可以划分为数据采集和以太网通信 2 大模块。FPGA 的 I/O 口丰富, 数据采集和处理能力强, 负责采集和存储设备运行过程的实时与故障数据; ARM 负责向 FPGA 获取数据, 进行封装打包校验, 并通过以太网协议发送给以太网控制器; 以太网控制器采用 ENC28J60 芯片, 负责将数据通过网络发送至上位机服务器中。

系统整体结构图如图 1 所示。可看出硬件平台的架构, 其主控单元由 STM32F407 系列 ARM 负责, 通过串行外设接口 (serial peripheral interface, SPI) 与以太网控制器 ENC28J60 进行通信,

通过可变静态存储控制器(flexible static memory controller, FSMC)与FPGA进行通信。

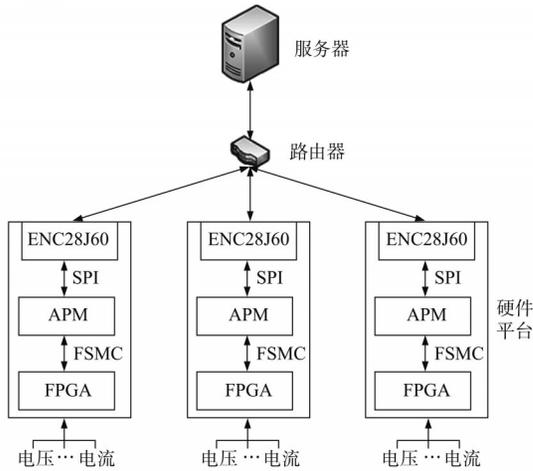


图1 远程监控网络结构图

Fig.1 Remote monitoring network structure

本研究中的ARM芯片采用的是STM32F407系列微处理器,它由意法半导体公司(ST)推出,采用Cortex-M4内核架构,是一款高性能的32位微控制器。该芯片最高主频可达168 MHz,集成单精度浮点运算单元,支持增强的DSP指令集。它不仅外设资源丰富、集成度高、功耗低,而且程序执行效率高,可以实现复杂的计算与控制。

FPGA采用的是Altera公司推出的Cyclone®IV

系列FPGA,具有低成本、低功耗的特点,高达532个用户I/O,支持高达200 MHz的DDR2 SDRAM接口。

并口通信:本设计使用了STM32F407的FSMC模块,它是一种新型存储器扩展技术,可以根据不同的外部存储器类型发出相应的数据/地址/控制信号类型以匹配信号的速度,使得微控制器可以应用各类外部静态存储器。STM32的FSMC支持8/16/32位的数据宽度,可以实现并口通信。

SPI通信:本设计中ARM与以太网控制器的通信使用SPI总线实现,它是一种高速、全双工的通信总线,只需要占用4根连线,SPI这种简单易用的特点使得它的应用越来越广泛。ENC28J60是一个28引脚的完整以太网控制器,具有可编程8 KB双端口SRAM缓冲器,通信速率为10 MB/s,它的优点是以SPI为接口,内部集成了MAC与PHY模块,几乎可以适用于任何MCU上,并且可以实现对数据包的过滤、校验、滤波等功能。

2 系统软件设计

ARM的软件程序主要由2部分组成:1)通过并口在外部中断中对数据进行采集存储;2)基于轻型网际协议(light weight IP, LWIP)协议栈实现TCP/IP协议并与上位机通信。程序流程如图2所示。

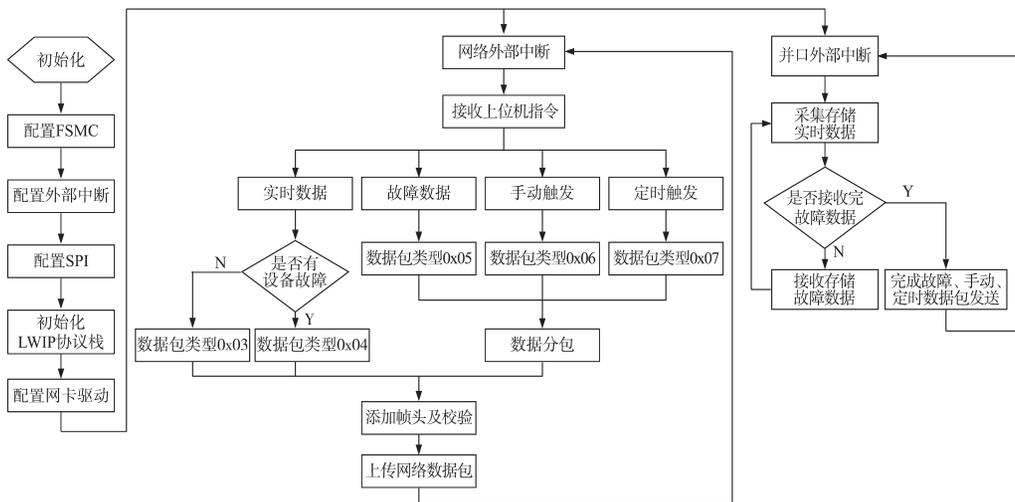


图2 远程监控软件流程图

Fig.2 Remote monitoring software flow chart

从图2可以看出,ARM在上电完成时钟等设置的初始化之后,就对FSMC,SPI和外部中断等硬件设备进行配置,初始化LWIP协议栈以及网卡驱动设备。在完成外设的配置后,程序通过不

同的中断接收上位机和FPGA发来的数据,并进行处理。当上位机向远程监控设备发出指令时,ENC28J60在接收到数据后产生一个外部中断给ARM,ARM读取网络数据包,并依据数据类型的

不同进行不同的操作。

2.1 并口通信及协议

当FPGA采集到数据时,将会触发外部中断,ARM在中断中接收实时数据,之后根据FPGA发来的状态字RBC进行判断,若是刚收到故障数据,则接收故障数据并置位标志位,表明设备存在故障;若是故障数据接收完成,则结束故障接收过程。最后根据状态字RSW判断是否上位机主动读取故障数据,若是,则将命令字CMD发送给FPGA,并于下一周期接收故障数据。在处理完FPGA发来的数据后,程序退出并口外部中断。

图3所示为ARM与FPGA通信中的命令字与状态字,数据通讯有2种模式。

1)模式0:触发数据传输,触发流程如下:①故障触发。当故障触发时,RSW=0x0001,表明故障触发的数据已经在铁电中存储完成。此时ARM发送命令字CMD=0x0040,FPGA接收到后将在下一个中断周期内发送故障数据。当RBC发送完成标志位置1后,故障数据发送完成,ARM将命令字CMD置为0x0000,此时RSW反馈为0x0000。②命令/定时触发。ARM发送命令字CMD=0x4000,FPGA收到后会从设备中读取故障数据并存储。当RSW=0x0002时,表明存储过程已经完成,此时ARM可以按照流程①进行故障数据读取。

2)模式1:实时数据传输,数据直接通过并口

传输。

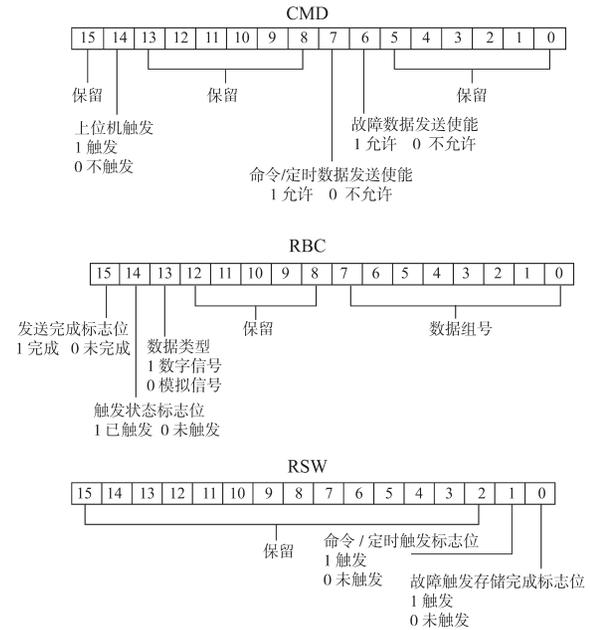


图3 ARM与FPGA通信命令字及状态字

Fig.3 ARM and FPGA communication command words and status words

2.2 网络通信及协议

在网络通信中,上位机作为服务器,通过TCP/IP协议发出读取数据的指令,ENC28J60在接收到上位机的数据后,触发ARM外部中断进行处理,ARM根据指令将从FPGA处采集到的数据组包上传给上位机。网络通信协议如图4所示,数据传输过程按类型也可以分为2大类。



图4 ARM与上位机通信协议

Fig.4 ARM and host computer communication protocol

1)模式0:触发数据传输,可以细分为2类:①故障触发。当FPGA检测到设备故障时,通过状态字RSW告知ARM,ARM随即完成故障数据

接收,在接收完成之后,上位机通过命令字0x05读取故障数据,ARM回传的故障数据一共14组,每组2 048个字节,故需要分为28包数据

先后传输,每包包含故障数据为1 024字节。②命令/定时触发。当上位机需要在没有故障触发的情况下主动读取相关数据时,通过发送命令字0x06/0x07给ARM,ARM随即发送命令字CMD=0x4000给FPGA,从FPGA中接收完成全部触发数据后,再由ARM分包上传给上位机。

2)模式1:实时数据传输,上位机每隔1 s向ARM请求1次实时数据传输,命令字为0x03。ARM向上位机回复数据时,数据包内的实时数据为8字节,若未检测到设备故障则命令字置为0x03,有故障则命令字置为0x04。上位机在接收到ARM上传的数据包后进行分析,若回传的命令字为0x04,则在下一周期向ARM请求故障数据传输,否则继续请求实时数据传输。

3 LWIP协议栈的移植与改进

LWIP是一种轻量级的开源嵌入式TCP/IP协议栈,由瑞士计算机科学院的Adam Dunkels等利用C语言开发完成,不仅能移植到多种操作系统上,也能在无操作系统的环境下使用。LWIP只需要几十KB的RAM与ROM即可运行,很适合应用在资源有限的嵌入式芯片上。它不仅保留了TCP协议的主要功能,同时集成了IP,UDP,ICMP,ARP及HTTP服务器等多种常见协议。LWIP被广泛应用于嵌入式芯片的网络通信中。

3.1 协议栈移植

LWIP采用分层方式实现TCP/IP协议,可以分为物理层、数据链路层、网络层、传输层和应用层。LWIP并没有针对具体的网络芯片编写网卡驱动程序,而是在`ethernetif.c`中提供了驱动接口给用户,用于实现网卡驱动设备的配置、发送和接收^[3]。用户需要根据使用的网卡芯片完成网卡初始化、数据接收、数据发送、中断处理,如图5所示。

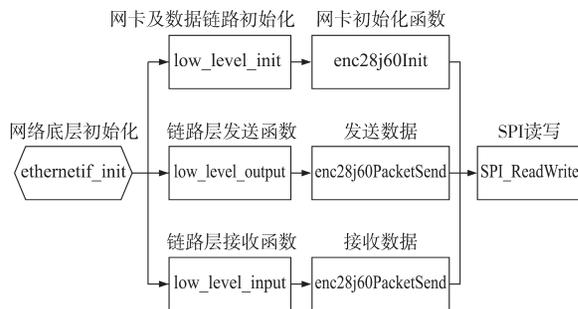


图5 LWIP的底层驱动接口流程图

Fig.5 Underlying driver interface flow chart of LWIP

1)网卡初始化。`ethernetif_init()`是网络接口

初始化函数,完成物理层和数据链路层的初始化,在其中调用`low_level_init(struct netif *netif)`函数中实现物理层的设置,将IP地址与MAC地址发送给ENC28J60,配置发送和接收缓冲区、帧长度、指示灯等一系列参数,完成ENC28J60的初始化。

2)网卡发送。在网络接口注册完毕后,应用层发送数据是通过发送函数`ethernetif_output`调用链路层发送函数`low_level_output`完成网络数据的发送。`low_level_output`中通过SPI将数据发送至ENC28J60的发送缓冲区中,并设置ENC28J60的ETXST和ETXND指针,EIE,EIR及ECON寄存器等信息,由ENC28J60将数据包发送至以太网。

3)网卡接收。在ENC28J60接收到以太网数据后,触发ARM的外部中断,在中断中调用接收函数`low_level_input`,通过SPI复位EIE,EIR和ECON寄存器,读取接收缓冲区内的全部数据,并计算接收缓冲器剩余空间,最后将得到的数据报文传递至网络层中。

3.2 协议栈改进

TCP/IP协议为了确保数据的可靠传输,需要具备快速重传与快速恢复算法,LWIP采用的是RFC5681文档中定义的算法,这一算法被称为Reno算法^[4]。该算法能够在网络丢包的情况下,快速进行报文重传而不必等待超时,同时在收到报文确认后启动拥塞避免算法,节约时间,提高通信效率。但是该算法只针对1个数据包进行重传,所以当网络中出现多个数据包丢失时,发送方就会发生多次快速重传,而无法选择重传哪一包,不利于减少网络拥塞。RFC2582和RFC3782文档定义了NewReno算法,加入了选择确认功能,能有效避免多次快速重传的缺陷,为了改进LWIP的TCP拥塞控制,本研究将NewReno算法引入LWIP中^[5]。

3.2.1 NewReno算法

当发送方收到某个数据包的3个重复ACK包,且没有处于快速重传算法中时,则记录下当前已发送最大报文序号`recover`,并修改慢启动门限`ssthresh`为

$$ssthresh = \max (FlightSize / 2 , 2 \times SMSS) \quad (1)$$

式中:`FlightSize`为已发出但未收到确认的字节数;`SMSS`为一帧数据最大可发送的字节数。

此时重传该数据包,并将拥塞窗口`cwnd`设置如

下式:

$$cwnd = ssthresh + 3 \times SMSS \quad (2)$$

当收到新的ACK时,根据其确认序号的大小,可以分为如下2种情形:

1)ACK确认序号大于等于recover,表明recover之前的数据已经得到接收方的确认,此时可以将cwnd设置为 $\min(ssthresh, FlightSize + SMSS)$ 或ssthresh,退出快速恢复算法。

2)ACK确认序号小于recover,表明接收方只确认了一部分数据帧,此时重传第1个未确认的数据帧,将cwnd设置为

$$cwnd += SMSS - ackno + lastack \quad (3)$$

式中:ackno为收到的ACK确认序号;lastack为上一个ACK确认序号。

3.2.2 NewReno算法的实现

本研究采用的LWIP版本为1.4.1,要将NewReno算法取代LWIP中使用的Reno算法,主要需要修改3部分程序。

1)在tcp_in.c文件中的tcp_receive函数中,将原代码:

```
if(pcb->flags & TF_INFR)
{
    pcb->flags &= ~TF_INFR;
    pcb->cwnd = pcb->ssthresh;
}
```

修改为如下代码:

```
If(pcb->flags & TF_INFR)
{
    If(ackno >= recover)
    {
        pcb->flags &= ~TF_INFR;
        pcb->cwnd = pcb->ssthresh;
    }
    else
    {
        pcb->cwnd += (pcb->mss - ackno + pcb->lastack);
    }
}
```

2)在tcp_in.c文件中的tcp_receive函数中,在原代码:

```
pcb->polltmr = 0;
```

后面添加如下代码:

```
if(((pcb->flags & TF_INFR) != 0) && (ackno < recover))
{
    tcp_rexmit(pcb);
}
```

3)在tcp_out.c文件中的tcp_rexmit_fast函数最后,添加如下代码:

```
recover=LWIP_MAX (pcb->lastack+1,  pcb->snd_nxt);
```

4 结论

本文在分析了当前远程监控需求的基础上,以STM32F407系列的ARM芯片为主控单元,完成并口数据采集、以太网数据传输以及LWIP协议栈的移植与改进,设计并实现了一套嵌入式远程监控数据采集系统。经过产品的实践表明:

1)将系统的数据采集模块和网络通信模块划分开,分别由FPGA和ARM承担的方案是可行的,这样不仅减少了ARM运行所需的资源,提高系统实时性,有效缓解了通信与I/O控制的矛盾,并有利于设备进一步扩展。

2)通信模块移植了LWIP,并以NewReno算法进行改进,弥补了LWIP原算法的不足,提高了设备通信的安全性和可靠性。

参考文献

- [1] 唐文俊,李维波,贺洪,等.一种基于ARM的远程监控系统的设计与实现[J].船电技术,2011,31(11):1-5.
- [2] 杨国均.制造装备的嵌入式监控与维护单元设计与开发[D].南京:南京理工大学,2011.
- [3] 孙棣华,陆旭,廖孝勇,等.面向设备监控的LwIP协议栈扩展设计[J].电子技术应用,2013,39(6):10-13.
- [4] 马敏锐,白瑞林,邹骏宇.嵌入式视觉系统的以太网通信拥塞控制改进[J].计算机工程与应用,2016,52(9):116-121.
- [5] 孟开元,陈晨,曹庆年.LwIP协议在工业远程控制网络中的应用[J].西安石油大学学报(自然科学版),2010,25(6):85-87.

收稿日期:2018-10-11

修改稿日期:2019-01-10