

文章编号:1000-582X(2005)03-0065-04

# “三层传输模型”在嵌入式系统与PC间通信中的实现

陈华鸿, 庞涛, 周莎

(中山大学 计算机应用研究所, 广东 广州 510275)

**摘要:**针对嵌入式系统与PC间通信的特点,基于tcp/ip通讯模型的思想,提出适合嵌入式系统与PC间通信的“三层传输模型”,把嵌入式系统的通信过程设计为数据传输层、数据控制层、数据应用层,每层利用下层提供的功能为上层服务,解决嵌入式系统与PC间数据传输的高效性、可靠性的问题,并通过实验得出相关数据进行分析说明。本模型的思想具备一定的通用性,适用串口,蓝牙,红外等各种接口或同类嵌入式系统与PC进行通信的问题。

**关键词:**串口通信;嵌入式系统;三层传输模型;TCP/IP分层模型;数据传输

**中图分类号:**TP393.04

**文献标识码:**A

近年来,嵌入式系统设备特别是PDA(Personal Digital Assistant)得到很快普及,其功能也越来越强大。其中,PDA与PC的通信功能日益显出其重要性,它大大地扩展了PDA的应用,如PDA与PC之间数据的导入、导出及同步,包括名片、记事、行程、理财和E-mail等;另外,它为实现PDA的系统升级及动态程序加载提供了底层支持。

文中的PC与PDA之间利用串口进行通信的研究,PC系统为Windows,开发平台为Delphi;PDA处理器为东芝TMP91C815<sup>[1]</sup>,RAM 4M,开发平台是用VC的IDE。实际研究中是在有线串口上实现通信的。考虑到在实际应用过程中,有可能采用红外、蓝牙等其它接口方式或者不同类型的嵌入式系统设备来实现互连以及数据通信,因此在项目的开发过程中,针对程序的可扩展性问题,尝试利用类似TCP/IP分层模型<sup>[2]</sup>对该类通信机制进行规划,以提高可扩展性和灵活性。希望为该类型的程序开发定义一个具体的模型架构。

## 1 串口通信的原理和机制

不论何种通信,背后都需要一个通信协议的支持。串口通信大多采用了美国电子工业协会(EIA)于1969年制定的RS-232标准。该标准规定了数据终端设备和数据通信设备之间的连接和通信规则。该协议运用RTS(Request to Send)和CTS(Clear to Send)信号来实现串口和外围设备的硬件“握手”,从而建立通信双方的连接和应答。在通信的连接和应答完成以后,双方就可以在误差允许范围内进行串行通信。

串行端口的本质功能作为CPU和串行设备间的编码转换器。当数据从CPU经过串行端口发送出去时,字节数据转换为串行的位;在接收数据时,串行的位被转换为字节数据。应用程序要使用串口进行通信,必须在使用之前向操作系统提出资源申请要求(打开串口),通信完成后必须释放资源(关闭串口)。串口通信的优点是开发简单,在传输数据量不大、要求速度不高而传输距离较大的通信场合得到广泛应用。

## 2 通信的流程

通信的流程采用传统的“C/S”通信模式<sup>[3]</sup>,以PC作为一台“服务器”必须一直不停地监视串口,一旦有收到PDA的请求则相应进行应答。而PDA作为一台“客户机”,则可以随时发送请求,从而简化了PDA端程序开发的难度。图1为PDA与PC数据通信流程图。

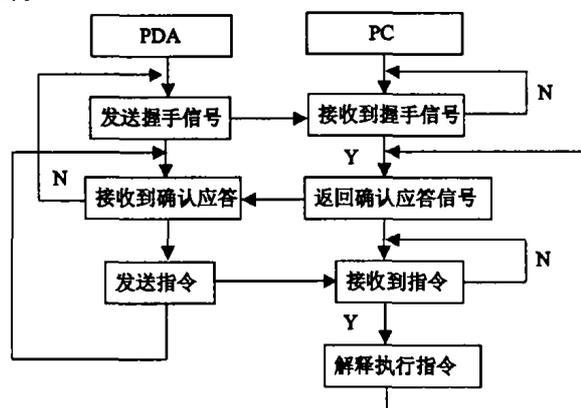


图1 PDA与PC数据通信流程

收稿日期:2004-10-28

基金项目:广东省教育厅《软件技术》重点实验室研究基金资助项目

作者简介:陈华鸿(1977-),男,广东广州人,助理工程师,中山大学博士研究生,研究方向:无线计算机技术。

### 3 三层传输模型在嵌入式系统通信中的探讨

考虑到程序的可扩展性,尝试借用 TCP/IP 的分层思想来规范嵌入式系统设备通信行为,在原有模型上进行必要增删及具体定义,形成“三层传输模型”,此分层模型在嵌入式系统通信中具有一定普遍意义。

1) 最底层可以称为数据传输层,类似于 TCP/IP 的网络接口层,它的任务就是利用传输介质为上层透明的传送和接收“数据流”。注意,在这里的传输介质是指通过串口以某一种具体的方式进行传输,实际上可以是其他的传输方式,比如红外接口、蓝牙技术等。这样当传输方式改变时,则不需要改变接口,只需要改变实现方式就可以完成了,使得程序的可扩展性大大加强。

2) 第 2 层可称为数据控制层,其完成的功能类似 TCP/IP 的网际层和传输层所完成的功能,也就是对上一层传下来的数据根据事先定的协议进行分组打包,并采取某种校验方式对包数据加入校验码和解开校验码,当数据包经校验发现有错时,负责进行重传,其最终目的就是向上一层提供一个可靠的无差错的运输连接。在数据控制层以上的层就不再关心信息传输的问题了。这样分层的好处是当双方的数据控制协议有变时(例如包长度,校验的方式有变动),只需要改变这一层的实现方式,而不会影响到其他层。在设计时,最好根据实际试验数据,在限定范围内设置包大小,以获取高传输效率以及稳定性能,因为如果包容量设置过小,虽然波特率比较高,但发包次数增多;如果过大,则波特率相对降低,也可能不能符合 PDA 的硬件限制,容易导致不稳定现象,具体试验数据在实验部分详细叙述。例如笔者做这个项目的时候,针对 V 系列一款 PDA,包的长度定为 64 个字节,并且采用了某种事先定好的类似于 CRC 的校验。当以后有变化时,就只需要改变这一层而不会影响到其他层了,从而增强了程序的可扩展性。

3) 第 3 层可以称为数据应用层。其完成的功能其实类似于 TCP/IP 的应用层,确定通信的性质以满足用户的需要。它主要是用来完成一些为进行语义上有意义的信息交换所必须的功能,如图 2 所示。

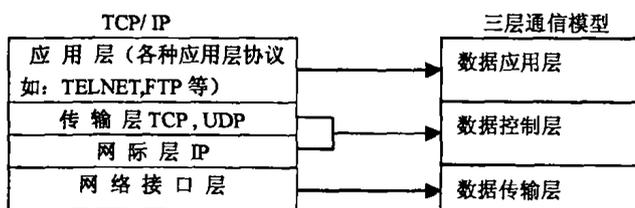


图 2 三层结构所对应的 TCP/IP 模型的结构

由于串口通信没有网络包在网络路由、网络阻塞控制以及最佳的利用网络资源等问题,所以设计中也并没有考虑传输层和网际层,取而代之的是进行数据分

包,数据校验,数据重传的数据控制层。

### 4 基于串口的通信的实现

在 Win32 下串行通信口是作为文件的方式来进行操作的<sup>[4]</sup>。一般 PC 机有 2 个串口,使用时文件名分别定义为“COM1”和“COM2”;操作时一般步骤见下表,程序主要分为 2 个部分:PC 机和 PDA,采用客户/服务端(C/S)模式。PC 机作为服务端执行以下 2 个功能:1)作为监听程序,负责接收 PDA 的指令;2)将数据通过串口传输给 PDA。以下是分层的具体实现<sup>[5-6]</sup>。

表 1 各操作函数定义

步骤	内容	函数	备注
1	打开串口	Createfile()	设置参数可以改变同步和异步模式
		BuildCommDCB()	填写设备控制块 DCB
2	设置串口	SetCommState()	改变各个参数
		SetCommTimeouts()	设置超时时间
3	对串口进行读写操作	WriteFile() ReadFile()	串口写数据 串口读数据
4	关闭串口	CloseFile()	

1) 数据传输层。在这一层分别封装了接收和发送两个函数,分别完成一个字节的发送和接收。以下是接收函数的主要部分:

```
eCode := CE_RXOVER or CE_BREAK or CE_O-
VERRUN;
ClearCommError(hCom, eCode, @ ComStat);
ReadNum := 0;
if not ReadFile(hCom, ReadBuf, 1, ReadNum,
Nil) Then exit;
```

注意:ReadFile 的参数 hCom 是通过 CreateFile 打开的指向串口的句柄。发送函数主要部分与之大体相同,只不过 Readfile 改成 WriteFile。

2) 数据控制层。这一层也是分别封装了接收和发送两个函数,完成打包和校验功能。以下是发送函数的主要部分:

```
while(Size > 0) Do
begin
Count := FillPackage(Buffer, PkgBuf, size,
Pos);
if not SendPackage(PkgBuf, Count + 4) then
begin
Result := ERRTRANSMIT;
Exit;
end;
Pos := Pos + Count;
Size := Size - count;
end;
```

函数调用了 FillPackage 对 Buffer 里面的数据逐一完成打包功能,然后再用 SendPackage 函数把打好包的数据进行发送。以下是 FillPackage 函数的主要部分:

```
for i := 0 to PkgLen - 1 do
    PkgBuf[ i + 2 ] := SrcBuf[ Pos + i ];
crc := Crc16CalcBlock( PkgBuf, PkgLen + 2, 0 );
PkgBuf[ PkgLen + 2 ] := char( Crc shr 8 );
PkgBuf[ PkgLen + 3 ] := char( Crc and $00ff );
Result := PkgLen;
```

这里函数又调用了 Crc16CalcBlock 函数算出包的校验码,然后把校验码放在包的后面。这里由于篇幅关系就不列出 Crc16CalcBlock 的代码。下面是发送包函数 SendPackage 的主要部分:

```
for RetryCnt := 1 to MaxTimes do
begin
    Result := False;
    eCode := CE_BREAK or CE_TXFULL or CE_
OVERRUN;
    ClearCommError( hCom, eCode, nil );
    if not PurgeComm( hCom, PURGE_TXCLEAR
or PURGE_RXCLEAR ) Then
        exit;
    for i := 0 to Size do
        SerialOut( Buffer[ i ] );
    if( WriteCnt < > Size ) Then
        Exit;
    for i := 0 to MaxTimes do
        begin
            Ret := SerialIn( Ack );
            if( CancelFg = 1 ) Then
                Exit;
            if Ret and ( Ack < > ACK_WAIT ) Then
                Break;
        end;
    if( Ack = ACK_OK ) Then
        begin
            Result := True;
            Exit;
        end
    else
        begin
            Ack := #0;
            Delays( DelayTimes );
        end;
end;
```

注意:由于篇幅关系,所以这里省略了一些数据重发,超时控制的源码。而在真实的项目中,这些控制是很

重要的。没有这些控制,也就无所谓说数据控制层可以向上层提供一个可靠的无差错的数据连接了。

3) 数据应用层。应用层完成的功能就是根据用户的数据交换需要,通过数据控制层再加上一些控制完成相应的功能就可以了。在这里应用层已经不关心数据传输的具体细节了,而把下一层看成是一条无差错的链路。笔者目前的程序在应用层完成的功能就是利用数据控制层传送和接收数据给 PDA,以后还可以进行扩展,如变成数据同步,这只需要改动应用层加入某些控制就可以了,其他层基本可以保持不变。

整个通信模型的分层结构框架以及数据在各层之间的传递过程如图 3 所示。

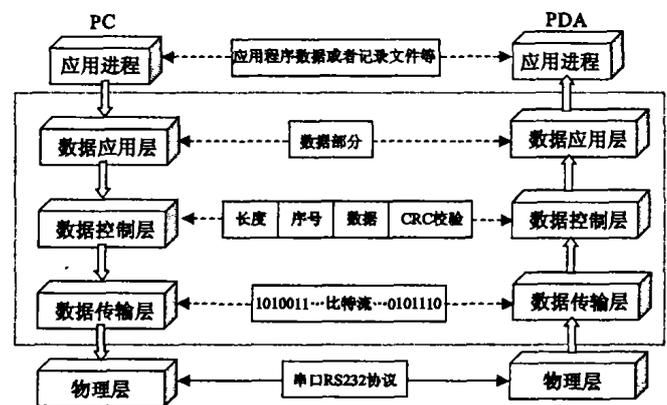


图 3 通信模型的分层结构框架

## 5 实验

采用快译通 V 系列 PDA (处理器类型:东芝 TMP91C815,处理器主频 33 MHz:,内存 RAM 容量:2 M,内存 ROM 容量:8 M) 进行实验测试。该系列 PDA 有串口、红外、USB 及蓝牙多种接口。通过 RS-232 串口进行测试,在测试过程中发现,串口的传输率可以为 67 k baud rate,程序能够非常稳定地在这种速度下进行通讯时,没有发现数据通信错误地情况,这跟笔者采用了 CRC 校验有很大地关系,可以很大程度纠正误码,且所设计的通信前的握手机制,通信错误重传机制,通信异常处理机制都保证了通信的高效可靠的进行。

将三层传输模型分别应用于该系列的 RS232、Bluetooth、IrDA 3 种接口进行速率以及稳定性测试,对多种数据源分别进行 10 次传输测试,结果全部在理论时间内完成并经验证无误差。

在 V68 上进行分包实验,对同一数据分别作 64 Byte 及 32 Byte 的分包测试,如图 4 所示,32 Byte 的包,可以在 115.2 kbps 下通信,64 Byte 的包数据只能在 57.6 kbps 下通信,但是 64 Byte 的数据的通信时间要少于 32 Byte 的包的数据,这是由于虽然 64 Byte 包

容量是 32 Byte 的 2 倍,但同样包的数目只有其一半,结合校验码、错误重发等因素,实际需要通信的次数要比 32 Byte 的少,实际需时也相对较短,因此各式 PDA 包大小的设置需要根据实验数据选取最佳方案。

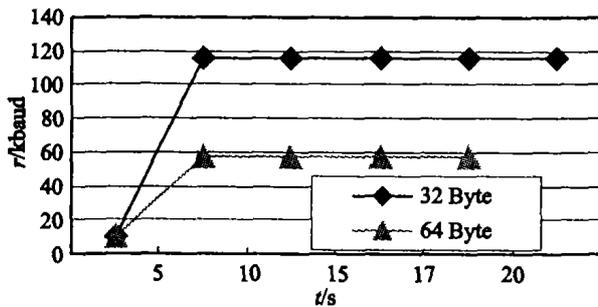


图 4 不同分包设置的波特率表现

图 5 是对快译通 V 系列多个型号的 PDA 进行测试,得出比较理想的分包方案以及其相应的波特率,可见在较早期的型号基本维持在 64 Byte 分包,57.6 kbps 的平衡位置,到后期的型号,由于硬件性能的提升,可以进一步增大包容量并且维持在 115.2 波特率下,可见硬件指标高的系统,通信速度固然比低的快,在同等硬件环境下,数据包的大小对通信速度的影响显得尤为重要,需要作具体试验分析,才能使 PDA 与 PC 通信达到最佳性能。

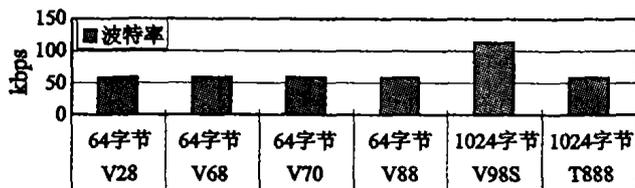


图 5 快译通 V 系列 PDA 分包设置及波特率视图

## 6 结束语

把嵌入式系统的通信过程设计为 3 层,分别封装为数据传输层,数据控制层,数据应用层,各层分别完成各自的功能,每一层利用下层提供的功能,并为上层提供服务,文中提出的分层模型,借鉴了原有 TCP/IP 分层模型的分层原理、纠错等方面的思想,同时针对嵌入式系统通信的特点作了适当增删,使用这样的分层模型进行设计,能有效的避免通信过程数据的丢失,高效可靠,也有利于错误的查找定位等问题,可以运用于蓝牙、红外等其他接口通讯中去,也可以引用至类似的嵌入式系统设备当中,从而使该模型具备一定普遍意义。

### 参考文献:

- [1] TOSHIBA. TMP91C815F DATA BOOK REV4.2 [EB/OL]. [http://www.toshiba.com/taec/components/Datasheet/TMP91C815\\_ETD\\_rev42.pdf](http://www.toshiba.com/taec/components/Datasheet/TMP91C815_ETD_rev42.pdf). 2001.
- [2] W RICHARD STEVENS. TCP/IP Illustrated Volume 1: The Protocols [EB/OL]. <http://home.student.uu.se/j/jolo4453/projekt/tcpip1/>, 1994.
- [3] 韩洪生,胡晨,陆生礼. PDA 与 PC 数据通信技术及其实现[J]. 电子器件, 2002, 25(1): 88-92.
- [4] 张黎军,史浩山. 利用串口实现 PC 与 PDA 的同步通信[J]. 计算机应用研究, 2002, (8): 104-121.
- [5] Allen Denver. MSSerialComminWin32 [M/CD]. Microsoft-Research(MSDN), 2001.
- [6] 尚建华,黄山. 用 VB 实现 PC 与 PDA 的串行通信[J]. 计算机工程, 2002, 28(3): 274-275.

## Realizing 3 Layers Transport Model in the Communication Between Embedded System and PC

CHEN Hua-hong, PANG Tao, ZHOU Sha

(Computer Application Institute, ZhongShan University, Guangzhou Guangdong 510275, China)

**Abstract:** This article puts forward a 3 Layers Model for the communication between embedded system and PC. It is based on the speciality of them and the TCP/IP model. The model defines the communicate process of embedded system to be three layers: Transport Layer, Control Layer and Application Layer. Each layer serves for its higher-up. It can keep the efficiency and credibility of the communicate process and the authors prove and analyse that in the article. the concept of the model is also suitable for Bluetooth, IrDA and other kinds of communication about embedded system.

**Key words:** embedded system; 3 layer transport model; TCP/IP model; data transport