文章编号：2095-2163(2019)01-0264-03 中图分类号：TP393.11 文献标志码：A

Delphi 开发西门子 S7-1200/1500 PLC 以太网心跳通信组件

张 毅
（东风汽车有限公司刃量具厂 刀量具制造管理部，湖北 十堰 442002）

摘 要：采用 Delphi 编译器开发西门子 S7-1200/1500 系列 PLC 客户端 DLL，组件通过 S7 系列 TCP 通信协议与 PLC 建立连接并保持心跳通信，上位机 EXE 通过接口调用组件，定时读取 PLC 寄存器数据，并支持实时写寄存器值，实现 PC 与 PLC 实时通信。

关键词：Delphi7；S7-1200/1500；TCP 协议；DLL 窗口封装

Delphi develops Siemens S7-1200/1500 PLC Etnernet heart rate communication module

ZHANG Yi
（Cutting & Measuring Tool Plant of DongFeng Motor Go., Ltd Shiyan 422002, China）

【Abstract】 In this paper, using Delphi compiler to develop Siemens S7-1200/1500 series PLC client DLL, components connect with PLC through S7 series TCP communication protocol and maintain heartbeat communication, The upper computer EXE calls components through interface, reads PLC register value regularly, and supports real-time write register values. The research realizes real-time communication between PC and PLC.

【Key words】 Delphi7; S7-1200/1500; TCP protocol; DLL window encapsulation

0 引 言

西门子 S7-1200/1500 系列 PLC 以太网通信，支持 TCP、UDP、ISO_on_TCP 通信协议。此时 PLC 作为服务器端，支持 8 个主动和 8 个被动连接。采用 Delphi 开发 PLC 客户端组件，根据西门子 TCP 握手通信协议发送握手指令与 PLC 建立连接。当连接成功后，遵循 TCP 读-写协议，定时器间隔约 0.5 秒自动读取 PLC 寄存器，并支持实时写入 PLC 寄存器，实现完整的心跳通讯。

为实现上述功能，本研究采用 Delphi 7 开发了 Windows 规则 DLL 模块，并在 DLL 内部封装了 1 个 Form 窗口、2 个 Timer 定时器和 1 个 TCPClient 控件。当 DLL 被 EXE 加载时，由 DLL 启动连接、发送套接字完成握手，通过内部窗口函数定时读 PLC 寄存器，实现写 PLC 寄存器函数，并完成接口封装。上位机 EXE 通过调用 DLL 接口，完成实时读写 PLC 寄存器功能，程序功能设计原理如图 1 所示。

1 硬件配置

本通信方式基于以太网卡 TCP/IP 通信协议。首先采用网线连接 PC 和 PLC 网口，使用 TIA 博途软件为 S7-1200/1500 系列 PLC 配置 IP 地址，例如将 PLC 端 IP 设为 192.168.0.2，端口号 0，并将 PLC 配置为 TCP Server，不主动发送数据模式。在 PC 端，将本机 IP 地址设为与 PLC 位于同一网段，使用 ping 命令，确保能 ping 通 PLC 站。

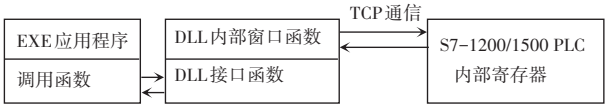

图 1 EXE 应用程序调用 DLL 实现 PLC 心跳通信原理图

Fig. 1 EXE application calls DLL to realize PLC heartbeat communication schematic diagram

2 实现过程

（1）发送与接收 TCP 握手报文，完成与 PLC 连接

在与 S7-1200/1500 系列 PLC 建立连接之前，需完成 2 次握手报文收发，均为固定格式，代码示例如下：

```
Var Buf_Rec:array[0..99] of Byte;  //定义数据接收数组
发送第一次握手报文示例：
Const Buf_Send1:array[0..21] of Byte=
(3,0,0,22,17,224,0,0,0,1,0,193,2,16,0,194,2,3,1,192,1,10);
```

接收第一次握手报文返回数据，判断 Buf_Rec[5]=208 表示第一次握手成功，接收报文示例：

```
Buf_Rec[ ]=(3,0,0,22,17,208,0,1,0,17,0,
```

作者简介：张 毅(1970-)，男，硕士，高级工程师，主要研究方向：电子技术、工业软件开发、项目风险管理。

收稿日期：2018-09-13

192,1,10,193,2,16,0,194,2,3,1,…)

发送第二次握手报文示例：

Const Buf_Send2:array[0..24] of Byte =

(3,0,0,25,2,240,128,50,1,0,0,204,193,0,8,0,0,240,0,0,1,0,1,3,192);

接收第二次握手报文返回数据,判断 Buf_Rec[0] <> 0 表示第二次握手成功,握手返回报文示例：

Buf_Rec[ ] = (3,0,0,27,2,240,128,50,3,0,0,204,193,0,8,0,0,0,0,240,0,0,1,0,1,0,240…)

发送代码示例：

Form1.TcpClient1.SendBuf(Buf_Send1,SizeOf(Buf_Send1)); //发送第一次握手指令

Form1.TcpClient1.SendBuf(Buf_Send1,SizeOf(Buf_Send2)); //发送第二次握手指令

接收数据代码示例：

Form1.TcpClient1.ReceiveBuf(Buf_Rec[0],80); //接收数据

(2)根据 S7-1200/1500 系列 TCP 读数据协议,实现读 PLC 寄存器值,读指令代码示例如下：

Const ar_bRead:array[0..66] of Byte = ($03, $00, //[0-1]报头

$00,67, //[2-3]整条数据长度

$02, $F0, $80, $32, //[4-7]固定长度:4(协议类型)

$01, //[8]命令类型:发

$00, $00, $00, $01, //[9-12]标识序列号:1(可自定义,与返回数据一致)

$00,50, $00, $00, //[13-16]命令数据总长度:50

$04, //[17]命令起始符:4

$04, //[18]读取数据块个数:4

$12, $0A, $10, $02, //[19-22]固定长度:4(读取地址前缀)

$00, $01, //[23-24]读取数据 byte 个数:1(8 位)

$00,200, //[25-26]读取数据块编号:200

$84, //[27]数据块类型: DB

$00, $00,0, //[28-30]地址偏移量:0 (DB200.0) = 1 Byte

$12, $0A, $10, $02, //[31-34]固定长度:4(读取地址前缀)

$00, $01, //[35-36]读取数据 byte 个数:1(8 位)

$00,200, //[37-38]读取数据块编号:200

$84, //[39]数据块类型: DB

$00, $00,17, //[40-42]地址偏移量:17 (DB200.2) = 17

$12, $0A, $10, $02, //[43-46]固定长度:4(读取地址前缀)

$00, $01, //[47-48]读取数据 byte 个数:1(8 位)

$00,200, //[49-50]读取数据块编号:200

$84, //[51]数据块类型: DB

$00, $00,81, //[52-54]地址偏移量:81 (DB200.10) = 1 Byte

$12, $0A, $10, $02, //[43-46]固定长度:4(读取地址前缀)

$00, $01, //[47-48]读取数据 byte 个数:1(8 位)

$00,200, //[49-50]读取数据块编号:200

$84, //[51]数据块类型: DB

$00, $00,89); //[52-54]地址偏移量:89 (DB200.11) = 1 Byte

Var Buf_Read:array[0..66] of Byte; //声明动态读数据指令数组

Move(ar_bRead, Buf_Read, SizeOf(Buf_Read)); //赋值

发送读指令代码示例：TcpClient1.SendBuf(Buf_Read,SizeOf(Buf_Read));

TCP 读指令只需通过 Timer 控件间隔 0.5-0.75 秒定时发送即可,接收数据与(1)所示相同,修改读指令可对 Buf_Read 数组动态赋值。

(3)根据 TCP 写数据协议,实现写 PLC 寄存器操作,代码示例如下：

Const ar_bWrite:array[0..35] of Byte = ($03, $00,//[0-1]固定报头

$00,36, //[2-3]数据总长

$02, $F0, $80, $32, //[4-7]固定长度:4

$01, //[8]命令类型:发

$00, $00, $00, $09, //[9-12]标记序列号:9

$00, $0E, //[13-14]固定长度:2

$00, $05, //[15-16]有效数据长度:5(地址偏移量后面第一位开始计算)

$05, //[17]命令起始符

$01, //[18]写数据块个数:1

$12, $0A, $10, //[19-21]固定长度:3(返回数据前缀)

$02, //[22]写入方式:$01 按 bit 写入,$02 按 byte(8 位)写入

$00, $01, //[23-24]写入数据个数:1(byte 方式可多写,bit 方式只能单个写)

$00,200, //[25-26]写入数据块编号:200

$84, //[27]数据类型:DB 块

$00, $00, $09, //[28-30]地址偏移量(bit),此处按 bit 计算偏移量

$00, $04, //[31-32]写入方式:$03 按 bit 写入, $04 按 byte(8 位)写入

$00,8, //[33-34]写入 bit 的个数:8

9); //[35]写入的值

Var Buf_Write:array[0..35] of Byte; //声明动态写数据指令数组

Move(ar_bWrite, Buf_Write, SizeOf(Buf_Write)); //赋值

修改地址偏移量示例(按位计算地址):Buf_Write[30] := Byte(1);

修改写入值示例:Buf_Write[35] := Byte(i_value);

发送写数据指令代码: TcpClient1.SendBuf(Buf_Write,SizeOf(Buf_Write));

接收数据与(2)所示相同。写数据需通过 DLL 接口函数带形参(a:寄存器地址,b:待修改的值),由 EXE 调用实现。

(4)在 DLL 中使用窗体

Delphi 支持在 DLL 工程内部使用窗体。当 DLL 被调用时,DLL 中的窗体被加入调用方的 EXE 进程,并在 DLL 被释放时销毁资源。在本例中,首先在 DLL 工程中添加一个窗体 Form1,在窗体上加入一个 TcpClient 控件和 PLC 通信,一个1 000 MS定时器处理 PLC 握手信号,一个 650MS 定时器用于向 PLC 定时接收和发送指令。然后在 Form 的 Public 部分申明被调用函数接口,提供给 DLL 接口函数使用。示例如下:

```
public
    { Public declarations }
    function fun_ReceiveData():Boolean; //接收数据函数
    procedure fun_SendRead; //发送读指令
    function fun_ReadHeartTime():Integer; //读心跳次数
    function fun_WriteParmData(i_addr, i_value:Integer):Integer; //写操作
    end;
```

在 Form 的 implementation 部分写函数实现代码,DLL 接口函数可直接调用上述内部窗口函数,示例如下:

```
function m23s71200_readio(a:integer):integer; export;stdcall; //读心跳次数
begin
  Result := Form1.fun_ReadHeartTime(); //获取心跳次数
end;
exports m23s71200_writeio; //声明导出函数
```

由于 DLL 内部封装了窗体和定时器控件,在 DLL 被 EXE 加载时,内部定时器可通过接口函数控制开启或关闭,也可自动开启定时器维持与 PLC 心跳通信。EXE 定时从 DLL 模块读数,并通过 DLL 接口实时向 PLC 写值,此调用方式可简化 EXE 程序设计,实现松耦合结构。

传统上位机软件大多采用 IO 模块、继电器加采集卡等硬接线方式与 PLC 实现信号交互,处理的信息量较小,且占用大量 PLC IO 端口资源。本研究采用上位机软件通过网线直接读写 PLC 内部寄存器,可实时监控 PLC 各 IO 口和 DB 块信号状态,并通过实时写寄存器值与 S7-1200/1500 系列 PLC 双向交互信号。该方法不仅能简化通讯设计,节约 PLC IO 资源,降低通信硬件成本,且能极大提高通信信息量,具有 IO 硬接线无法比拟的优势。

3 结束语

在工控领域,本研究采用的方法具有通讯设计简单、实施成本低廉、方案灵活性好、通信信息量大等优点,目前已在多个工程项目中得到成功应用。

参考文献

[1] 龙启明,刘斌,程捷,等. Delphi 7 高级编程范例[M]. 北京:清华大学出版社,2004.

[2] 刘艺. Delphi 模式编程[M]. 北京:机械工业出版社,2004.

[3] 刘华波. 西门子 S7-1200 PLC 编程与应用[M]. 北京:机械工业出版社,2011.

[4] SIEMENS AG 2009. S7-1200 & STEP7 Basic V10.5[K]. Berlin & Munich:SIEMENS,2009.

[5] SIEMENS. S7-1500 PLC 手册[Z]. 北京:西门子(中国)有限公司,2018.

[6] SIEMENS. S7-1500 做服务器端与第三方设备的 TCP 通信[K]. Berlin & Munich:SIEMENS,2015.