固高科技（深圳）有限公司

地　　址：深圳市高新技术产业园南区深港产学研基地西座二层 W211 室

电　　话：0755-26970823　26970819　26970824

传　　真：0755-26970821

电子邮件：support@googoltech.com

网　　址：http://www.googoltech.com.cn

Googol Technology (HK) Ltd

Addr: Room 3639,Annex Building Hong Kong University of Science and Technology, Hong Kong

Tel: (852) 2358-1033

Fax: (852) 2358-4931

E-mail: info@googoltech.com

Web: http://www.googoltech.com

CPAC_TCP 库的函数使用说明

（2010-7-8）

务必将此手册交给用户

- 非常感谢您选购 CPAC 控制器
- 在您使用之前，请仔细阅读此手册，确保正确使用。
- 请将此手册妥善保存，以备随时查阅。

版权声明

 注意	***运动中的机器有危险！使用者有责任在机器中设计有效的出错处理和安全保护机制，固高科技没有义务或责任对由此造成的附带的或相应产生的损失负责。***

目 录

1 前言

CPAC_TCP.lib 是基于CPAC-OtoStudio软件基础库SysLibSocket.lib开发的，SysLibSocket.lib是OtoStudio自带的标准系统库，用于实现TCP,UDP Socket数据交互的基本功能。

该库采用非常简单的配置与参数，包含TCP server和TCP client的标准通讯函数

该文档描述了该库提供的模型以及其函数的功能和使用方法。

2 使用方法

TCP协议库的使用方法非常简单。所需的步骤在OtoStudio工程中的描述如下。

2.1 创建一个 OtoStudio 工程

- 打开OtoStudio软件
- 创建一个新的工程通过“文件/新建”
- 选择对应的Otobox控制器，如：CPAC-X00-TPV控制器
- 创建新的POU “PLC_PRG“ (选择编程语言，如：FBD).

2.2 添加库

- 打开资源->库文件管理器
- 右键添加库：CPAC_TCP.LIB
- 库STANDARD.LIB 以及 SYSLIBSOCKET.LIB，SYSLIBMEM.LIB 将会自动被添加. 如果没有被自动添加，请重复上一步操作，将这两个库也添加进来.

2.3 功能函数列表

函数名	功能
TcpServerOpenSocket	打开 TCP/IP 服务器
TcpClientOpenSocket	打开 TCP/IP 客户端
TcpServerWaitForConnect	诊断客户端是否连接服务器成功
TcpReceiveData	TCP/IP 客户端或者服务器接受数据
TcpSendData	TCP/IP 客户端或者服务器发送数据

2.4 功能函数的参数介绍

2.4.1 TcpServerOpenSocket

Opens a TCP server socket. Return: Socket-ID

输入参数	类型	意义
uiPort	UINT	Port number
diMaxConnections	DINT	Max possible client connections of the server

2.4.2 TcpClientOpenSocket

Opens a client socket to connect to a server. Return: Socket-ID for connection session

输入参数	类型	意义
iPort	INT	Port number of TCP socket to open
stIpAddress	STRING	IP-Address of server to connect to

2.4.3 TcpServerWaitForConnect

Wait for a client to connect.Return: Socket and IP-Address of connected client

输入参数	类型	意义
diSocket	DINT	Socket-ID
diTimeout	DINT	Timeout in ms

2.4.4 TcpReceiveData

Receive data via TCP socket.Return: Number of bytes sent or a value < 0 for an error (-1:

gracefully closed; -2: error)

输入参数	类型	意义
diSocket	DINT	Socket-ID
pbyData	DWORD	Address of data buffer

diDataSize	DINT	Size of data to send
diTimeout	DINT	Timeout in ms

2.4.5 TcpSendData

Send data via TCP socket.Return: Number of bytes sent or a value < 0 for an error (-1:

gracefully closed; -2: error)

输入参数	**类型**	**意义**
diSocket	DINT	Socket-ID
pbyData	DWORD	Address of data buffer
diDataSize	DINT	Size of data to send
diTimeout	DINT	Timeout in ms

3 示例程序

下面是用ST语言编写的使用CPAC_TCP开发通讯的示例程序。

3.1 TcpServer example

```
PROGRAM TCPServer
(*
    Example for a TCP server.
*)
VAR
    diSocket: DINT:= SOCKET_INVALID;
    uiPort: UINT:= 4444;
    Client: CLIENT_ACCEPT;
    abySend: ARRAY [1..10] OF BYTE;
    abyRecv: ARRAY [1..10] OF BYTE;
    abyRecvRaw: ARRAY [1..10] OF BYTE;
    bActive: BOOL;
    bInit: BOOL:= TRUE;
    bSend: BOOL;
    bEcho: BOOL:= TRUE;
```

```
    bSimulate: BOOL;
    bSimulationMode: BOOL:= TRUE; (*TRUE: Counter; FALSE: Random*)
    SimulateTimer: TON:= (PT:= t#250ms);
    byCounter: BYTE;
    bReset: BOOL;
    iIndex: INT;
    fbRandom: Random;
    diReceived: DINT;
    diSent: DINT;
    bSent: BOOL;
    bReceived: BOOL;
    bSendActivity: BOOL;
    bReceiveActivity: BOOL;
    LifeSignTimer: TON;
    tLifeSign: TIME:= t#500ms;
    LifeSignTimeoutTimer: TON;
    bLifeSignTimeout: BOOL;
    tLifeSignTimeout: TIME:= t#1s;
    byLifeSign: BYTE;
    bCheckConnection: BOOL:= TRUE;
END_VAR


IF bInit THEN
    (*Register Callbacks*)
    RegisterCallbacks();
    (*Initialising random number generator*)
    fbRandom(Seed:=TIME_TO_DINT(TIME()));
    bInit:= FALSE;
END_IF
IF bActive THEN
    (*Open listening socket*)
    IF diSocket = SOCKET_INVALID THEN
        diSocket := TcpServerOpenSocket(uiPort, 1);
    ELSE
        (*Wait for a client connection*)
        IF Client.diSocket = SOCKET_INVALID THEN
            Client := TcpServerWaitForConnect(diSocket, 250);
        ELSE

            (*Lifesign*)
            IF bCheckConnection THEN
                LifeSignTimer(IN:= TRUE, PT:= tLifeSign);
                LifeSignTimeoutTimer(IN:= TRUE, PT:= tLifeSignTimeout);
```

```
            IF LifeSignTimeoutTimer.Q THEN
                bLifeSignTimeout:= TRUE;
            ELSE
                bLifeSignTimeout:= FALSE;
            END_IF
        ELSE
            bLifeSignTimeout:= FALSE;
        END_IF

        (*Simulate data*)
        IF bSimulate THEN
            SimulateTimer(IN:= TRUE);
            IF SimulateTimer.Q THEN
                IF bSimulationMode THEN (*Count bytes*)
                    byCounter:= byCounter + 1;
                    FOR iIndex:= 1 TO 10 DO
                        abySend[iIndex]:= byCounter;
                    END_FOR
                ELSE (*Random bytes*)
                    FOR iIndex:= 1 TO 10 DO
                        fbRandom();
                        abySend[iIndex]:= DINT_TO_BYTE(fbRandom.Value);
                    END_FOR
                END_IF
                bSend:= TRUE;
                SimulateTimer(IN:= FALSE);
            END_IF
        END_IF

        (*Transmit*)
        IF bSend THEN
            diSent:= TcpSendData(Client.diSocket, ADR(abySend), SIZEOF(abySend),
10);
            IF diSent <= 0 THEN (*Problem occured*)
                CloseSocket(ADR(Client.diSocket));
                Client.stIPAddress:= '';
                LifeSignTimeoutTimer(IN:= FALSE);
            ELSE
                bSent:= TRUE;
                LifeSignTimer(IN:= FALSE);
            END_IF
            bSend:= FALSE;
        ELSIF LifeSignTimer.Q THEN
            diSent:= TcpSendData(Client.diSocket, ADR(byLifeSign),
```

```
SIZEOF(byLifeSign), 10);
                    IF diSent <= 0 THEN (*Problem occured*)
                        CloseSocket(ADR(Client.diSocket));
                        Client.stIPAddress:= '';
                        LifeSignTimeoutTimer(IN:= FALSE);
                    ELSE
                        bSent:= TRUE;
                        LifeSignTimer(IN:= FALSE);
                    END_IF
                END_IF

                (*Receive*)
                REPEAT
                    diReceived:= TcpReceiveData(Client.diSocket, ADR(abyRecvRaw),
SIZEOF(abyRecvRaw), 10);
                    IF diReceived < 0 THEN (*Problem occured*)
                        CloseSocket(ADR(Client.diSocket));
                        Client.stIPAddress:= '';
                        LifeSignTimeoutTimer(IN:= FALSE);
                    ELSIF diReceived > 0 THEN (*Something received*)
                        LifeSignTimeoutTimer(IN:= FALSE);
                        bReceived:= TRUE;
                        IF NOT (diReceived = 1 AND abyRecvRaw[1] = 0) THEN
                            abyRecv:= abyRecvRaw;
                            IF bEcho THEN (*Echo received data*)
                                abySend:= abyRecv;
                                diSent:= TcpSendData(Client.diSocket, ADR(abySend),
SIZEOF(abySend), 10);
                                IF diSent <= 0 THEN (*Problem occured*)
                                    CloseSocket(ADR(Client.diSocket));
                                    Client.stIPAddress:= '';
                                ELSE
                                    bSent:= TRUE;
                                    LifeSignTimer(IN:= FALSE);
                                END_IF
                            END_IF
                        END_IF
                    END_IF
                UNTIL
                    diReceived <= 0
                END_REPEAT;

                IF bSent THEN
                    bSendActivity:= NOT bSendActivity;
```

```
                bSent:= FALSE;
            END_IF
            IF bReceived THEN
                bReceiveActivity:= NOT bReceiveActivity;
                bReceived:= FALSE;
            END_IF

        END_IF
    END_IF
ELSE
    (*Shutdown server*)
    IF diSocket <> SOCKET_INVALID THEN
        CloseSocket(ADR(diSocket));
        IF Client.diSocket <> SOCKET_INVALID THEN
            CloseSocket(ADR(Client.diSocket));
            Client.stIPAddress:= '';
        END_IF
    END_IF
    IF LifeSignTimer.IN THEN
        LifeSignTimer(IN:= FALSE);
    END_IF
    IF LifeSignTimeoutTimer.IN = TRUE THEN
        LifeSignTimeoutTimer(IN:= FALSE);
    END_IF
    IF bLifeSignTimeout THEN
        bLifeSignTimeout:= FALSE;
    END_IF
END_IF
IF bReset THEN
    (*Reset data*)
    SysMemSet(dwDest:= ADR(abySend), bCharacter:= 0, dwCount:= SIZEOF(abySend));
    SysMemSet(dwDest:= ADR(abyRecv), bCharacter:= 0, dwCount:= SIZEOF(abyRecv));
    bReset:= FALSE;
END_IF
```

3.2 TcpClient example

```
PROGRAM TCPClient
(*
    Example for a TCP client.
*)
VAR
    diSocket: DINT:= SOCKET_INVALID;
```

```
        uiPort: UINT:= 4444;
        stDestIPAddress: STRING:= '127.0.0.1';
        abySend: ARRAY [1..10] OF BYTE;
        abyRecv: ARRAY [1..10] OF BYTE;
        abyRecvRaw: ARRAY [1..10] OF BYTE;
        bActive: BOOL;
        bInit: BOOL:= TRUE;
        bEcho: BOOL;
        bSimulate: BOOL:= TRUE;
        bSimulationMode: BOOL:= TRUE; (*TRUE: Counter; FALSE: Random*)
        SimulateTimer: TON:= (PT:= t#250ms);
        byCounter: BYTE;
        bReset: BOOL;
        iIndex: INT;
        fbRandom: Random;
        bSend: BOOL;
        diReceived: DINT;
        diSent: DINT;
        bSent: BOOL;
        bReceived: BOOL;
        bSendActivity: BOOL;
        bReceiveActivity: BOOL;
        LifeSignTimer: TON;
        tLifeSign: TIME:= t#500ms;
        LifeSignTimeoutTimer: TON;
        bLifeSignTimeout: BOOL;
        tLifeSignTimeout: TIME:= t#1s;
        byLifeSign: BYTE;
        bCheckConnection: BOOL:= TRUE;
    END_VAR

    IF bInit THEN
        (*Register Callbacks*)
        RegisterCallbacks();
        (*Initialising random number generator*)
        fbRandom(Seed:=TIME_TO_DINT(TIME()));
        bInit:= FALSE;
    END_IF
    IF bActive THEN
        IF diSocket = SOCKET_INVALID THEN
            diSocket := TcpClientOpenSocket(uiPort, stDestIPAddress);
        ELSE

            (*Lifesign*)
```

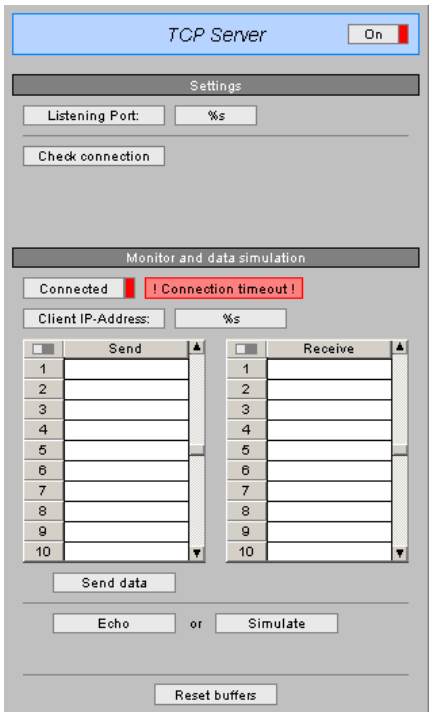
```
IF bCheckConnection THEN
    LifeSignTimer(IN:= TRUE, PT:= tLifeSign);
    LifeSignTimeoutTimer(IN:= TRUE, PT:= tLifeSignTimeout);
    IF LifeSignTimeoutTimer.Q THEN
        bLifeSignTimeout:= TRUE;
    ELSE
        bLifeSignTimeout:= FALSE;
    END_IF
ELSE
    bLifeSignTimeout:= FALSE;
END_IF

(*Simulate data*)
IF bSimulate THEN
    SimulateTimer(IN:= TRUE);
    IF SimulateTimer.Q THEN
        IF bSimulationMode THEN (*Count bytes*)
            byCounter:= byCounter + 1;
            FOR iIndex:= 1 TO 10 DO
                abySend[iIndex]:= byCounter;
            END_FOR
        ELSE (*Random bytes*)
            FOR iIndex:= 1 TO 10 DO
                fbRandom();
                abySend[iIndex]:= DINT_TO_BYTE(fbRandom.Value);
            END_FOR
        END_IF
        bSend:= TRUE;
        SimulateTimer(IN:= FALSE);
    END_IF
END_IF

(*Transmit*)
IF bSend THEN
    diSent:= TcpSendData(diSocket, ADR(abySend), SIZEOF(abySend), 10);
    IF diSent <= 0 THEN (*Problem occured*)
        CloseSocket(ADR(diSocket));
        LifeSignTimeoutTimer(IN:= FALSE);
    ELSE
        bSent:= TRUE;
        LifeSignTimer(IN:= FALSE);
    END_IF
    bSend:= FALSE;
ELSIF LifeSignTimer.Q THEN
```

```
            diSent:= TcpSendData(diSocket, ADR(byLifeSign), SIZEOF(byLifeSign), 10);
            IF diSent <= 0 THEN (*Problem occured*)
                CloseSocket(ADR(diSocket));
                LifeSignTimeoutTimer(IN:= FALSE);
            ELSE
                bSent:= TRUE;
                LifeSignTimer(IN:= FALSE);
            END_IF
        END_IF

        (*Receive*)
        REPEAT
            diReceived:= TcpReceiveData(diSocket, ADR(abyRecvRaw),
SIZEOF(abyRecvRaw), 10);
            IF diReceived < 0 THEN (*Problem occured*)
                CloseSocket(ADR(diSocket));
                LifeSignTimeoutTimer(IN:= FALSE);
            ELSIF diReceived > 0 THEN (*Something received*)
                LifeSignTimeoutTimer(IN:= FALSE);
                bReceived:= TRUE;
                IF NOT (diReceived = 1 AND abyRecvRaw[1] = 0) THEN
                    abyRecv:= abyRecvRaw;
                    IF bEcho THEN (*Echo received data*)
                        abySend:= abyRecv;
                        diSent:= TcpSendData(diSocket, ADR(abySend), SIZEOF(abySend),
10);
                        IF diSent <= 0 THEN (*Problem occured*)
                            CloseSocket(ADR(diSocket));
                        ELSE
                            bSent:= TRUE;
                            LifeSignTimer(IN:= FALSE);
                        END_IF
                    END_IF
                END_IF
            END_IF
        UNTIL
            diReceived <= 0
        END_REPEAT;

        IF bSent THEN
            bSendActivity:= NOT bSendActivity;
            bSent:= FALSE;
        END_IF
        IF bReceived THEN
```

```
                bReceiveActivity:= NOT bReceiveActivity;
                bReceived:= FALSE;
            END_IF

        END_IF
    ELSE
        (*Shutdown client*)
        IF diSocket <> SOCKET_INVALID THEN
            CloseSocket(ADR(diSocket));
        END_IF
        IF LifeSignTimer.IN THEN
            LifeSignTimer(IN:= FALSE);
        END_IF
        IF LifeSignTimeoutTimer.IN THEN
            LifeSignTimeoutTimer(IN:= FALSE);
        END_IF
        IF bLifeSignTimeout THEN
            bLifeSignTimeout:= FALSE;
        END_IF
    END_IF
    IF bReset THEN
        (*Reset data*)
        SysMemSet(dwDest:= ADR(abySend), bCharacter:= 0, dwCount:= SIZEOF(abySend));
        SysMemSet(dwDest:= ADR(abyRecv), bCharacter:= 0, dwCount:= SIZEOF(abyRecv));
        bReset:= FALSE;
    END_IF
```

4 人机界面

4.1 TcpServer example



4.2 TcpClient example