**串口自由协议**

自由协议通讯也称无协议通讯，需要根据对方设备的通讯数据格式编写一个临时协议，虽然随着标准协议（Modbus，USS等）的普及，自由协议通讯应用越来越少，但是对于一些小的设备如扫码枪、LCD显示屏等，由于没有集成标准通讯协议，所以只能选用自由协议通讯。

使用串口自由协议指令前须在工程中添加“Serial Communication”库文件。

本文介绍SC2-C使用串口自由协议实现与MC500 PLC之间的串口数据交互。

**1.串口自由协议指令**

串口自由协议指令一览。

|  |  |  |
| --- | --- | --- |
| **指令类别** | **名称** | **功能** |
| 串口通讯 | COM.Close | 关闭串行端口 |
| COM.Open | 打开串行端口 |
| COM.Write | 从串行端口进行自由协议数据发送 |
| COM.Read | 从串行端口读取自由协议的接收数据 |

* + - 1. **打开串口自由协议通讯**

COM.Open(xExecute :=启动信号, xDone=>完成信号, xBusy=>执行中信号,

xError =>错误标志, usiListLength :=参数参数列表条目数,

pParameterList:=参数列表指针, eError=>错误信号, hCom =>连接句柄 );

其中：连接句柄hCom等于0，表示通讯连接失败。

* + - 1. **关闭串口自由协议通讯**

COM.Close (xExecute:= 启动信号, xDone=>完成信号, xBusy=>执行中信号,

xError =>错误标志, hCom:=连接句柄, eError=>错误信号 );

其中：连接句柄hCom与指令COM.Open的相同。

* + - 1. **自由协议通讯数据读取**

COM.Read(xExecute:=启动信号, xAbort:=功能执行中止信号, udiTimeOut:=超时时间 ,

xDone=>完成信号, xBusy=>执行中信号, xError=>错误标志,

xAborted=>操作中止信号, hCom:=连接句柄, pBuffer:=指向接收数据缓存区的指针,

szBuffer:= 接收数据最大字节数, eError=>错误信号,

szSize=>pBuffe中接收到的数据字节数 );

其中：连接句柄hCom与指令COM.Open的相同。

* + - 1. **自由协议通讯数据写入**

COM.Write(xExecute:=启动信号, xAbort:=功能块执行中止信号, udiTimeOut:=超时时间,

xDone=>完成信号, xBusy=>执行中信号, xError=>错误标志,

xAborted=>中止信号, hCom:=连接句柄, pBuffer:=指向发送数据缓存区的指针,

szSize:= 发送数据的字节数, eError=>错误信号);

其中：连接句柄hCom与指令COM.Open的相同。

以上是Serial Communication库中自由协议通讯的指令，下面两个指令是自由协议通讯读写数据时常常会用到的两个指令：SysMemSet和SysMemCpy，这两个指令在SysMem函数库中。

* + - 1. **SysMemCpy指令**

SysMemCpy指令的主要作用是将内容从源(pSrc)复制到目标缓冲区(pDest)，返回值为完成复制的目标缓冲区的指针，其指令格式如下。

SysMemCpy（pDest:=指向要复制到的内存地址指针, pSrc:=指向要从中复制的内存地址指针,

udiCount:=复制字节数）;

* + - 1. **SysMemSet指令**

SysMemSet指令是使用指定值初始化内存空间，返回值为已初始化的内存块的指针。如果操作失败，则为0。其指令格式如下：

SysMemSet（pDest:=指向初始化内存块指针, udivalue:=初始化内存值,

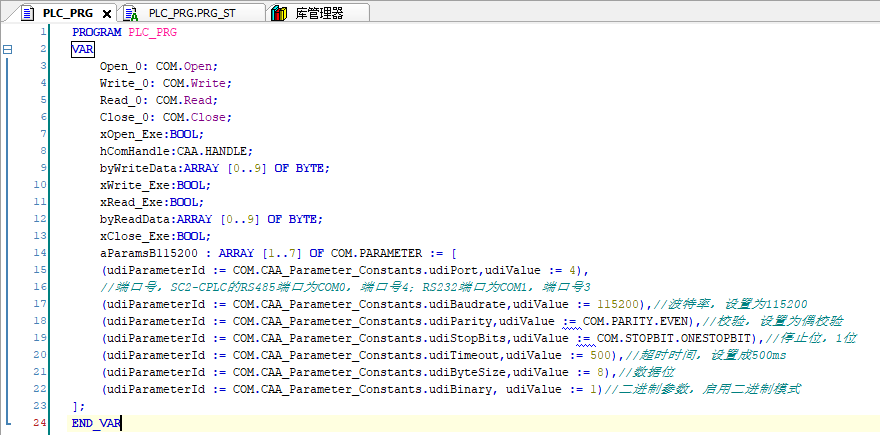
udiCount:= 内存块中初始化的字节数）;

**2.串口自由协议通讯例程**

本例程实现SC2-C COM0（RS485）与MC500 COM1（RS485）之间的串口自由协议通讯，实现两个PLC之间的数据交互。其中，两个PLC串口自由协议通讯程序除端口号设置不同外，其余部分完全相同。下面以SC2-C程序介绍此例程的实现方法。

2.1 创建变量

创建下图所示变量。



2.2 打开串口连接

①调用COM.Open指令打开串口。该指令的变量usiListLength为变量pParameterList的数据长度（根据后续介绍可知，数据长度为 7）；变量pParameterList为指针变量，指针地址指向结构体COM.PARAMETER的数组aParamsB115200，数组包含了 udiPort,udiValue(波特率)、 udiBaudrate,udiValue (端口号)、udiParity,udiValue (校验位)、udiStopBits,udiValue (停止位)udiTimeout,udiValue (超时时间)、udiByteSize,udiValue（数据位）及udiBinary,udiValue（二进制参数，启用二进制模式）7 个关于串口的参数，相关的变量创建直接参照下列库文件中的例程，再对参数进行设置；变量hCom为结构体 CAA.HANDLE数据类型，获取通讯句柄；

②本例程SC2-C作以下串口参数设置。

|  |  |
| --- | --- |
| 端口号 | COM0 |
| 波特率 | 115200 |
| 校验位 | 偶校验 |
| 停止位 | 1 |
| 超时时间 | 500 |
| 数据位 | 8 |
| 二进制参数 | 1 |

MC500作以下串口参数设置。

|  |  |
| --- | --- |
| 端口号 | COM1 |
| 波特率 | 115200 |
| 校验位 | 偶校验 |
| 停止位 | 1 |
| 超时时间 | 500 |
| 数据位 | 8 |
| 二进制参数 | 1 |

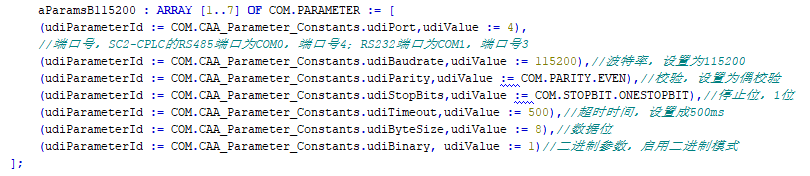
SC2-C串口自由协议通讯COM端口号对应udiPort值如下。

|  |  |  |
| --- | --- | --- |
| 端口号 | COM0(RS485) | COM1(RS485) |
| udiPort值 | 4 | 3 |

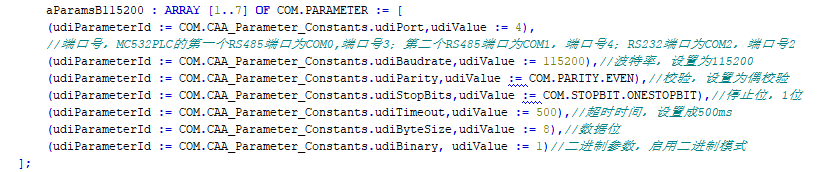
MC500串口自由协议通讯COM端口号对应udiPort值如下。

|  |  |  |  |
| --- | --- | --- | --- |
| 端口号 | COM0(RS485) | COM1(RS485) | COM2(RS232) |
| udiPort值 | 3 | 4 | 2 |

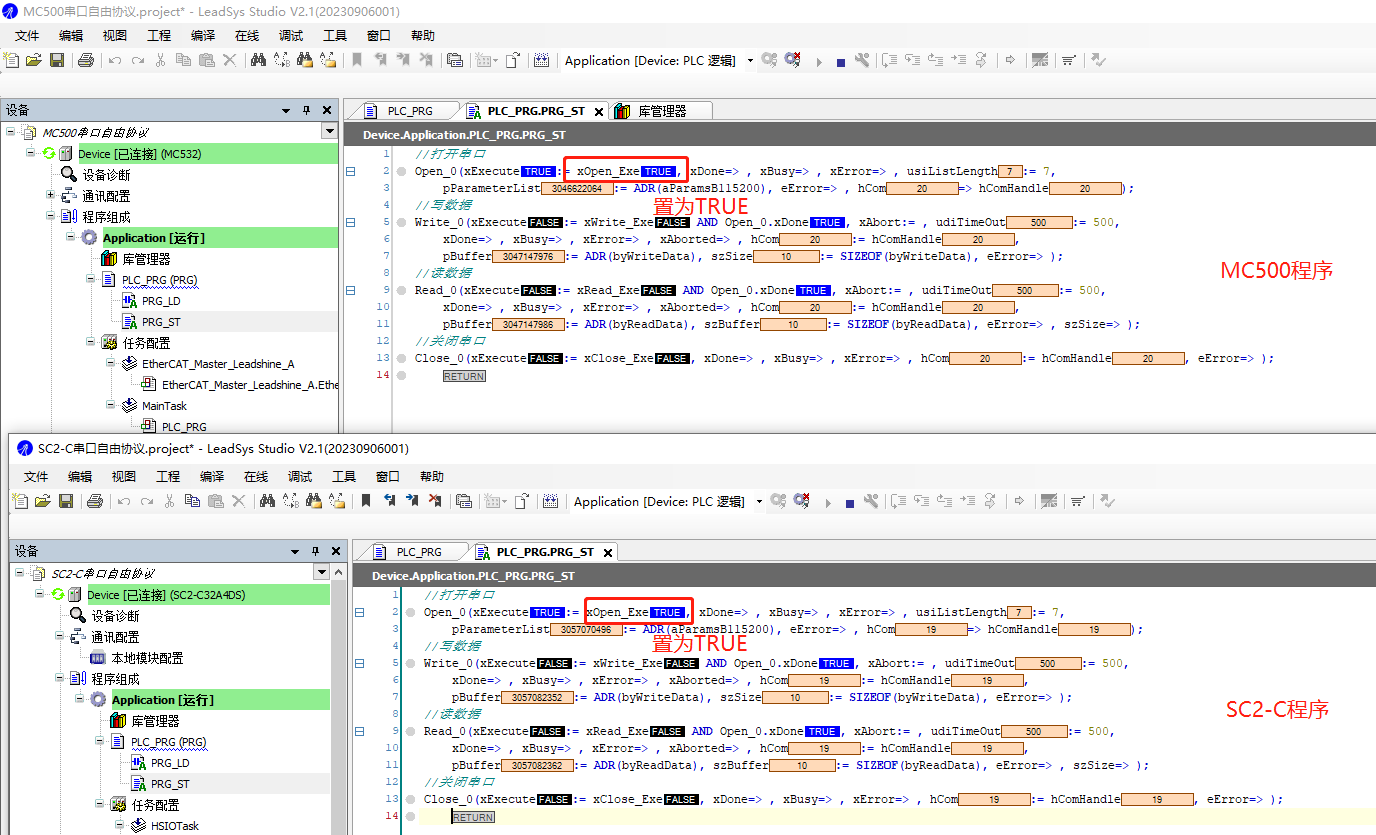
故SC2-C pParameterList指针变量指向的结构体数组COM.PARAMETER按以下参数设置。



MC500 pParameterList指针变量指向的结构体数组COM.PARAMETER按以下参数设置。



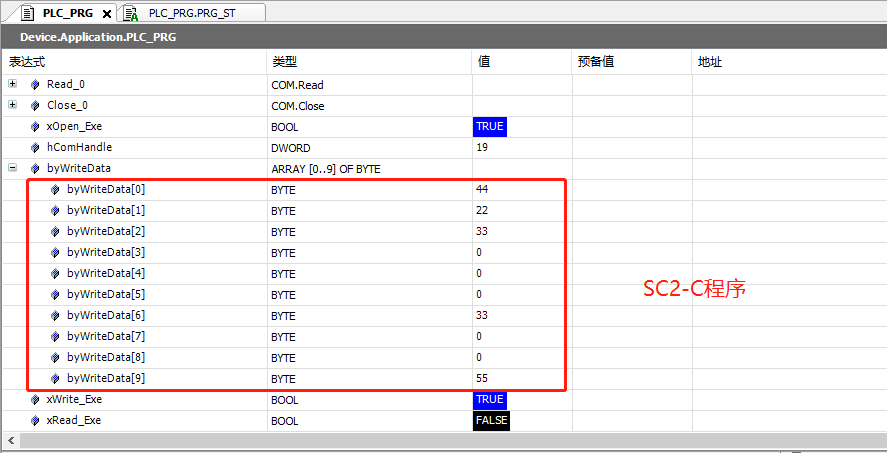
③打开串口连接。将SC2-C和MC500程序的变量xOpen\_Exe置为TRUE，当COM.Open指令的xDone输出TRUE，说明对应的串口已经成功打开。



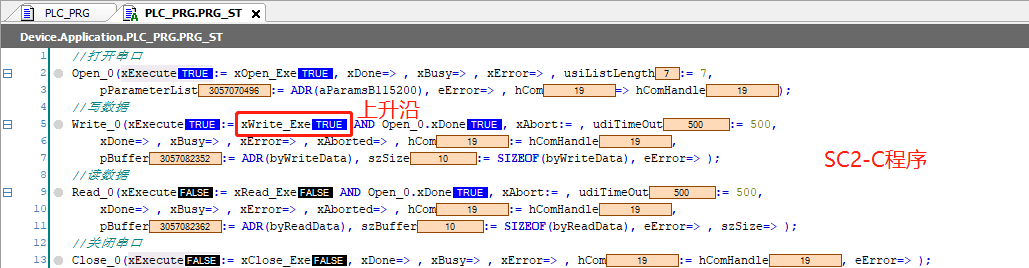
2.3 SC2-C发送数据

① SC2-C调用COM.Write指令发送数据。变量udiTimeout超时时间，此次例程设置为 500ms；变量hCom为结构体CAA.HANDLE数据类型，为通讯句柄，可从Com.Open中的输出变量 hCom 获取；pBuffer为指针类型，数据指向串口发送数据缓存的地址，本例程设置写入BYTE类型数组变量的byWriteData，因此pBuffer填入ADR（byWriteData）；szSize为写入数据长度，通过SIZEOF(byWriteData)获取byWriteData的数据长度；

② SC2-C变量byWriteData按如下图赋值；



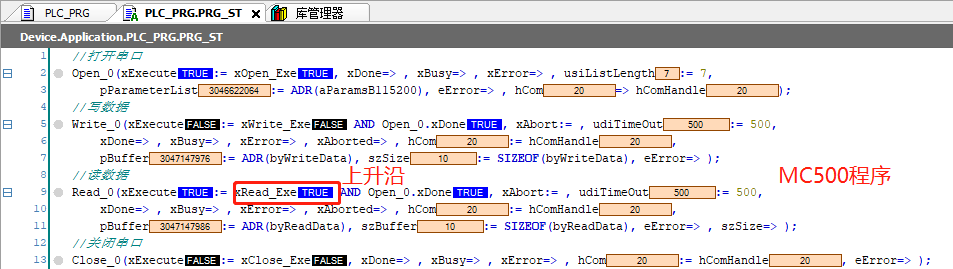
③ COM.Write指令xExecute触发上升沿，此时变量byWriteData的数据被发出；



2.4 MC500接收数据

① MC500调用COM.Read指令接收数据。变量udiTimeout超时时间，此次例程设置为 500ms；变量hCom为结构体CAA.HANDLE数据类型，为通讯句柄，可从 Com.Open 中的输出变量hCom获取；pBuffer为指针类型，数据指向串口读取数据缓存的地址，本例程设置BYTE类型数组变量byReadData来存放接收数据，因此pBuffer填入ADR（byReadData）；szBuffer为从串口缓存读取数据的大小，通过SIZEOF(byReadData)获取byReadData的数据长度；

② COM.Read指令xExecute触发上升沿，此时可接收到SC2-C发送过来的数据，数据存放在数组变量byReadData中。



③ 查看MC500数据接收结果。查看byReadData数组的值与SC2-C发送的数值相同，说明SC2-C数据发送成功、MC500数据接收成功。

