NET　多功能交换机 C 型 以太网模块

用户手册

版本：V2.01

发布日期：08/2017

大连德嘉工控设备有限公司

版权声明

在线支持

除本手册外，还可以在网上获取相关的产品资料和技术服务。
http://www.dl-winbest.com

目录

1 产品概述

大连德嘉推出的多功能交换机 C 型产品。

- 可以作为ModbusTCP服务器，为带有ModbusTCP协议的上位软件等提供连接西门子PLC的接口。例如，想通过 Modbus TCP 读取西门子 PLC 中的数据，可以先用 C 型读取 PLC 中数据，再用 Modbus 主站读取 C 型中的数据。
- 可以实现西门子 PLC 之间通讯，包括 S7-200、300、1200、200smart。
- C 型内部具有与 200PLC 一样的存储区域，是一个不带 IO 点的 200PLC。

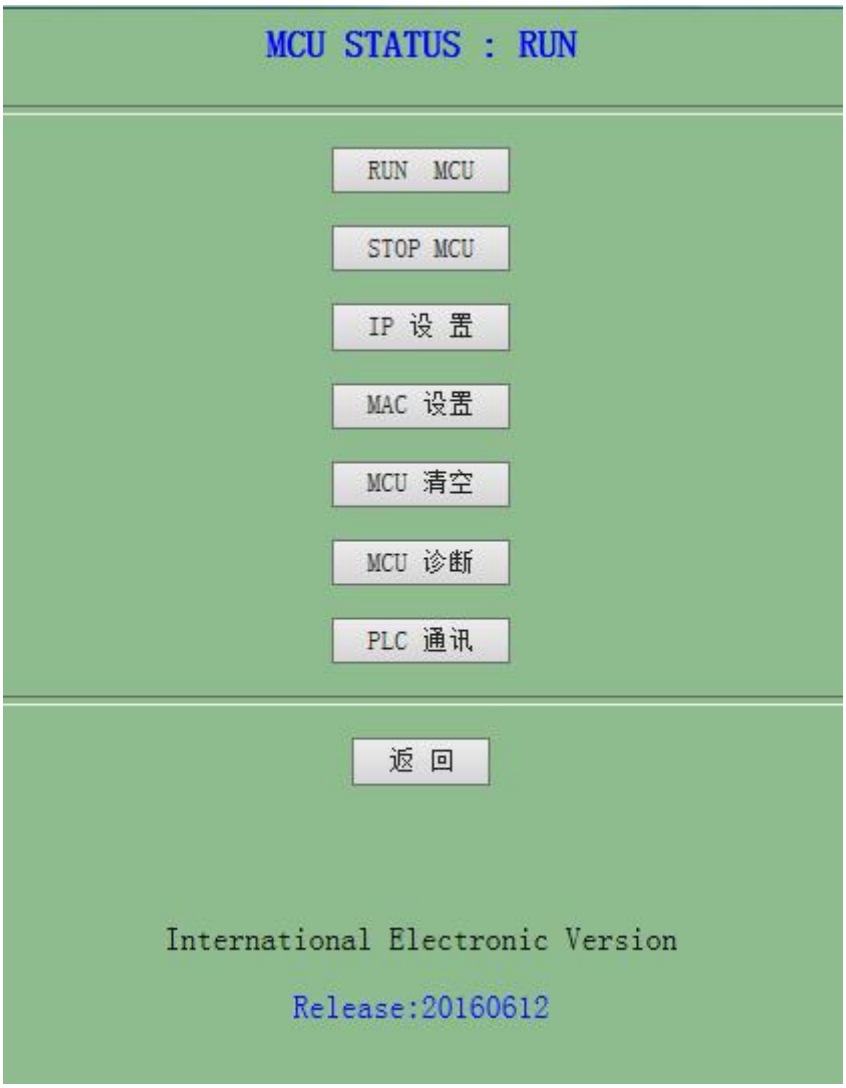
C 型产品可以建立的 ModbusTCP 连接数量和 PLC 通讯通道数量

每台多功能交换机 C 型可以同时建立 12 路 ModbusTCP 连接，具有 12 个 PLC 通讯通道（每个通道最大 200 个字节，支持取数据/送数据）

2 参数设置

1. 多功能交换机设置的后门 IP 地址为 xxx.xxx.xxx.222：（例如 192.168.1.222 、192.168.0.222）；将计算机和多功能交换机通过网线连接
2. 在计算机的 IE 浏览器中键入该后门 IP 地址就可进入多功能交换机主菜单;当然用实际的起始 IP 地址也可直接进入。

用鼠标点击 “Chinese”，进入中文页面

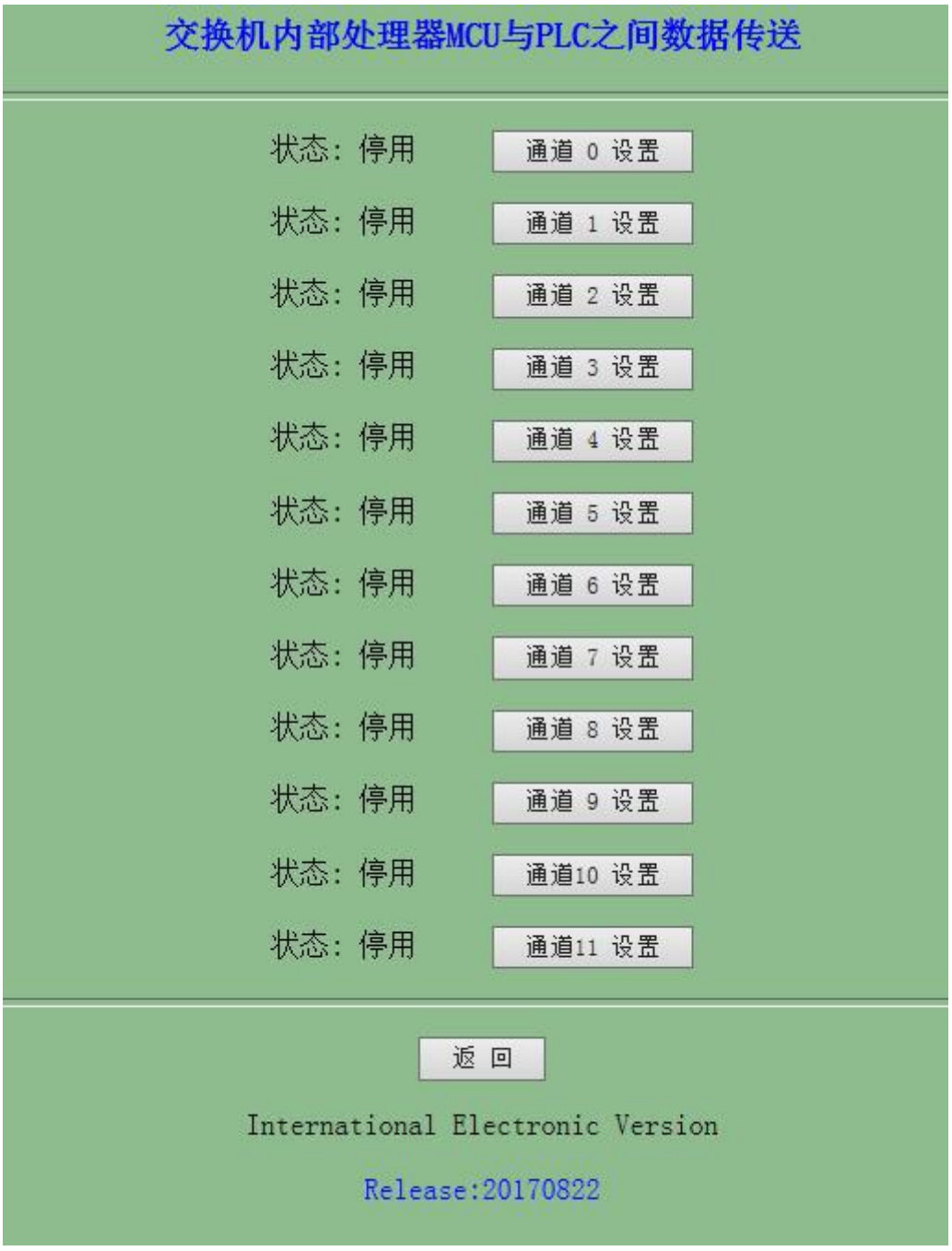


用鼠标点击“IP 设置”即可修改 C 型的 IP 地址（默认是 192.168.1.10）

注：参数设置提交后，如果想再次进入主菜单，最好在 cmd 窗口键入 C:> arp -d （删除计算机中已保留的 IP/MAC 表），以便新改动的 IP 与老地址无冲突。

3 PLC 通讯功能

用鼠标点击 “PLC 通讯”进入通讯功能设置页面



交换机 C 型共有 12 路通道，每个通道 200 字节

C 型可以将西门子各型号 PLC 中的数据通过以太网取到 C 型的存储区中（V 区），也可以把这些数据转发到其它 PLC 中。

例如：
您需要把 IP 地址为 192.168.1.20 的 S7-1200 PLC 中 DB1.DBB0 开始的 200 个字节取到 C 型中 VB0 起始的 200 个字节中，可以如下图填表：

注：对方 PLC IP 地址后的起始地址可以先选择数据区后再填写，如图中先选择 DB 块然后填入 DB 块号 00001，则起始地址 00000 表示从 DB1.DBB0 开始。

4 C# Modbus TCP 通讯实例

这里我只是简单的理解一下 Modbus TCP/IP 协议的内容，就是去掉了 modbus 协议本身的 CRC 校验，增加了 MBAP 报文头。

这里只是简单的理解，深入之后可能会有更多的东西需要学习，但为了可以快速入门，我们先按照这个思路往下走。

我们首先来看一下，MBAP 报文头都包括了哪些信息和内容

MBAP 报文头包括下列域：

域	长度	描述	客户机	服务器
事务元标识符	2 个字节	MODBUS 请求/响应事务处理的识别码	客户机启动	服务器从接收的请求中重新复制
协议标识符	2 个字节	0=MODBUS 协议	客户机启动	服务器从接收的请求中重新复制
长度	2 个字节	以下字节的数量	客户机启动（请求）	服务器（响应）启动
单元标识符	1 个字节	串行链路或其它总线上连接的远程从站的识别码	客户机启动	服务器从接收的请求中重新复制

下面我们再来介绍一下针对 C 型产品的功能码

1、0x01 功能码： 按位读取 Q 区（线圈 ）

例：我们来读取从 Q0.0 到 Q0.5 这 6 个线圈

发送码分析：

请求 PDU

功能码	1 个字节	**0x01**
起始地址	2 个字节	0x0000 至 0xFFFF
线圈数量	2 个字节	1 至 2000（0x7D0）

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x01, 0x00, 0x00, 0x00, 0x06

接收码分析：

响应 PDU

功能码	1 个字节	0x01
字节数	1 个字节	N*
线圈状态	N 个字节	n=N 或 N+1

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x04, 0x01, 0x01, 0x01, 0x2A

modbus 数据中从左数，0x01 表示功能码，0x01 表示 1 个字节数据，0x2A 表示数据值

把 0x2A 转换为 2 进制为 0010 1010 ， 从左数起，前 2 位是补充数据 00，剩下的 101010 表示我们读取的 Q0.5 到 Q0.0 的状态。

Q0.5------ ON,

Q0.4 ------ OFF,

Q0.3-------ON,

Q0.2--------OFF,

Q0.1-------ON,

Q0.0----------OFF。

注意数据的顺序，左侧是高位，右侧是低位。

注意：上述发送及接收数据中，红色数码是 MBAP 报文头，黑色码是 modbus 数据， 下同

2、0x02 功能码：按位读取 I 区 （离散输入）

例：我们来读取从 I0.0 到 I0.5 这 6 个离散输入点

发送码分析：

请求 PDU

功能码	1 个字节	**0x02**
起始地址	2 个字节	0x0000 至 0xFFFF
输入数量	2 个字节	1 至 2000（0x7D0）

根据上面的分析，我们需要发送 0x00，0x01，0x00，0x00，0x00，0x06，0x01， 0x02，0x00，0x00，0x00， 0x06

接收码分析：

响应 PDU

功能码	1 个字节	0x82
字节数	1 个字节	N*
输入状态	N*×1 个字节	

*N＝输出数量/8，如果余数不等于 0，那么N＝N+1

我们收到的数据为 0x00， 0x01， 0x00， 0x00， 0x00， 0x04， 0x01, 0x02，0x01，0x00

modbus 数据中从左数，0x02 表示功能码，0x01 表示 1 个字节数据，0x00 表示数据值

把 0x0 转换为 2 进制为 0000 0000 ， 从左数起，前 2 位是补充数据 00，剩下的 000000 表示我们读取的 I0.5 到 I0.0 的状态。

3、0x03 功能码：按双字节（VW）读取 V 区或者读 MW

Modbus 寄存器 0--------19999 是读取 VW

Modbus 寄存器 20000-------20031 是读取 MW

例：我们来读取从 VW0 到 VW2 这个数据

发送码分析：

请求

功能码	1 个字节	**0x03**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 125（0x7D）

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x03, 0x00, 0x00, 0x00, 0x03

接收码分析：

响应

功能码	1 个字节	0x03
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	

*N=寄存器的数量

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x09, 0x01, 0x03，0x06，0x04，0x00，0x03，0x01，0x02，0x05

modbus 数据中从左数，0x03 表示功能码，0x06 表示 6 个字节数据，0x04，0x00，0x03，0x01，0x02，0x05 表示数据值

VW0 为 0x0400，VW2 为 0x0301，VW4 为 0x0205

4、0x05 功能码：按位写 Q 区

例：我们来把 Q0.0 置 1，请注意，置位数据为 0xFF00，清零数据为 0x0000

发送码分析：

请求

功能码	1 个字节	0x05
输出地址	2 个字节	0x0000 至 0xFFFF
输出值	2 个字节	0x0000 至 0x00

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x05, 0x00, 0x00, 0xFF, 0x00

接收码分析：

响应

功能码	1 个字节	**0x05**
输出地址	2 个字节	0x0000 至 0xFFFF
输出值	2 个字节	0x0000 至 0xFF00

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x05，0x00，0x00，0xFF，0x00，

5、0x06 功能码： 按双字节（VW）写 V 区或者写 MW

Modbus 寄存器 0--------19999 是写 VW

Modbus 寄存器 20000-------20031 是写 MW

例：我们将数据 0x2636 写入 VW0

发送码分析：

请求

功能码	1 个字节	**0x06**
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x06, 0x00, 0x00, 0x26, 0x36

接收码分析：

响应

功能码	1 个字节	**0x06**
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x06，0x00，0x00，0x26，0x36

6、0x0F 功能码：按多个位写 Q 区

例：我们将 Q0.0 到 Q0.5 共 6 个线圈全部置位 1

发送码分析：

请求 PDU

功能码	1 个字节	**0x0F**
起始地址	2 个字节	0x0000 至 0xFFFF
输出数量	2 个字节	0x0001 至 0x07B0
字节数	1 个字节	**N***
输出值	**N***×1 个字节	

***N**=输出数量/8，如果余数不等于 0，那么N＝N+1

我们要将 Q0.0 到 Q0.5 输出 1，要发送的值应该为二进制 0011 1111，转换为 16 进制为 0x3F

根据上面的分析，我们需要发送 0x00，0x01，0x00，0x00，0x00，0x08，0x01， 0x0F，0x00，0x00，0x00， 0x06，0x01，0x3F

接收码分析：

响应 PDU

功能码	1 个字节	**0x0F**
起始地址	2 个字节	0x0000 至 0xFFFF
输出数量	2 个字节	0x0001 至 0x07B0

我们收到的数据为 0x00，0x01，0x00，0x00，0x00，0x06，0x01，0x0F，0x00，0x00，0x00，0x06

7、0x10 功能码： 写 2N 个 VW 或者 MW

Modbus 寄存器 0-------19999 是写 VW

Modbus 寄存器 20000------20031 是写 MW

例：我们将数据 0x01，0x05，0x0A，0x09 写入 VW0 和 VW2

发送码分析：

请求 PDU

功能码	1 个字节	**0x10**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	0x0001 至 0x0078
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	值

*N＝寄存器数量

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x0B, 0x01, 0x10, 0x00, 0x00, 0x00, 0x02, 0x04, 0x01, 0x05, 0x0A, 0x09

接收码分析：

响应 PDU

功能码	1 个字节	**0x10**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 123（0x7B）

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x10，0x00，0x00，0x00，0x02

好的，至此，我们关于 Modbus TCP 命令连接我们 PLC 的分析就结束了，后面我上传了我做好的 C#程序供大家参考，

这里要注意一个问题，此程序中缺少断线重连机制，请大家自己添加一下吧

大连德嘉工控设备有限公司
Dalian Winbest Industrial Control Co. Ltd.

辽宁省大连市沙河口区高尔基路 751 号和平现代城 A 座 2 单元 4 层 1 号
销售热线：0411-82810696
技术支持：13322207824　15712391325
网址：http://www.dl-winbest.com