

西门子销售低压断路器经销商

| | |
|------|---|
| 产品名称 | 西门子销售低压断路器经销商 |
| 公司名称 | 湖南西控自动化设备有限公司 |
| 价格 | .00/件 |
| 规格参数 | |
| 公司地址 | 中国（湖南）自由贸易试验区长沙片区开元东路1306号开阳智能制造产业园（一期）4#栋301 |
| 联系电话 | 15344432716 15386422716 |

产品详情

西门子销售低压断路器经销商

PLC是什么意思？相信很多人处于大概知道是什么，但是又无法准确说出的阶段，作为专注于为企业提供数据采集和设备控制解决方案的众诚工业，今天和大家探讨一下。

而众诚工业还能根据用户需求，设计PLC控制程序，为客户提供PLC编程和上位机软件的定制化开发技术服务，满足用户的多种需求，比如，自主研发的洁净空调智能控制系统和通风排风智能控制系统就配置PLC，不仅具有报警和定时控制功能，还兼具可扩展性和兼容性，系统能被第三方系统集成。

以上PLC的基本介绍，相信大家对PLC也有一个初步的了解。PLC的型号、品牌不同，对应着其结构形式、性能、编程方式等等都有所差异，价格也各不相同，在挑选时候，建议先要明确自己的应用需求，比如具体的应用场景，希望实现的运动和控制功能，已经特殊的控制要求，这些将决定了PLC的选型和搭配组合。

简单地说，PLC就是一种小型的计算机，和我们常用的计算机不同的是，PLC是设备之间通过数字信号进行互动，而我们常用的计算机，是人和计算机的互动。

控制是PLC的核心功能，其控制类型主要分为以下几种1、开关量的开环控制。这是PLC*基本的控制功能，它能凭借其强大的逻辑运算能力，取代传统继电接触器的控制系统；

2、数据采集与监控。这是PLC非常必要的功能，否则它将无法完成现场控制；

3、数字量智能控制。PLC具有实现接收和输出高速脉冲的功能，近年来先进的PLC还开发了数字控制模块和新型运动单元模块，让工程师更加轻松地通过PLC实现数字量控制；

4、PLC能通过模拟量采集和调节温度、压力、速度等参数。

正因为PLC功能强大，且具有设计方便、重量体积小、能耗低、改造工作量小、通用性强、维护方便等易学易用的特点，深受工程师的欢迎，因此应用非常广泛，钢铁、石油、化工、纺织、交通、机械制造等等行业都能看到它的身影。

说明 发送或接收数据时发生通信错误 如果发送或接收数据时发生通信错误，则 CPU

将终止现有连接。错误如下：80C4 -临时通信错误；将暂时终止指定的连接。80C5 -

远程伙伴已主动终止了连接。80A1 - 指定连接已断开或尚未建立。即，可查看连接终止时所有返回的

STATUS 值，以及连接中止时仅输出造成连接终止原因的 STATUS

代码。示例：如果接收数据时发生临时通信错误，则先输出 STATUS 7003 (ERROR=false)，然后再 输出

80C4(ERROR=true)。MB_RED_SERVER (作为 Modbus TCP 服务器通过 PROFINET

进行通信) 使用该指令可在 S7-1200 CPU 与支持 Modbus TCP 协议的设备之间建立连接。表格

13-73MB_RED_SERVER 指令 LAD/FBD SCL 描述 "MB_RED_SERVER_DB"(DISCONNECT:=_bool_in_,

LICENSED=>_bool_out_IDENT_CODE=>_string_out_DR_NDR_0A=>_bool_out,ERROR_0A=>_bool_out,

STATUS_0A=>_word_out_,DR_NDR_1A=>_bool_out, ERROR_1A=>_bool_out,STATUS_1A=>_word_out_,

DR_NDR_0B=>_bool_out,ERROR_0B=>_bool_out, STATUS_0B=>_word_out_,DR_NDR_1B=>_bool_out,

ERROR_1B=>_bool_out,STATUS_1B=>_word_out_,

RED_ERR_S7=>_bool_out_,RED_ERR_DEV=>_bool_out_,

TOT_COM_ERR=>_bool_out_,MB_HOLD_REG:=_variant_inout_); MB_RED_SERVER 指令作为 Modbus

TCP 服务器通过PROFINET 连接进行通信。 MB_RED_SERVER 指令将处理 Modbus TCP

客户端的连接请求、 接收并处理Modbus 请求并发描述 REG_KEY 1 IN STRING[17] 授权注册码

必须在每个 CPU 上分别对MB_RED_SERVER 指令进行授权。 DISCONNECT IN Bool 使用

MB_RED_SERVER指令建立与一个伙伴模块的被动连接。 服务器会对连接描述中给出的指定或未指定 IP

地址的连接请求作出 响应。 接受一个连接请求后， 可以使用该参数进行控制： 0： CPU

在无通信连接时建立被动连接。 1： 终止连接初始化。 如果设置了输入， 则CPU 不会处理其它

客户端请求， 并且启动终止连接。 成功终止连接后， STATUS_x 参数将输出值 0003。 MB_HOLD_REG 2

IN_OUT Variant 指向 MB_RED_SERVER 指令中 Modbus 保持性寄存器的指针MB_HOLD_REG

引用的存储区必须大于两个字节。 保持性寄存器中包含 Modbus 客户端通过 Modbus 功能

3 (读取)、 6 (写入)、 16 (多次写入) 和 23 (在一个作业中读写) 可访问的值。 LICENSED 1 OUT

Bool 0： 指令未获授权 1： 指令已获授权 IDENT_CODE 1 OUT STRING[18]

授权标识。 使用此字符串申请REG_KEY 注册码。 . DR_NDR_0A OUT Bool “ Data Read ” 或 “ New Data

Ready ” 至连接0A： 0： 无新数据 1： Modbus 客户端读取或写入的新数据 ERROR_0A OUT Bool

如果在调用MB_RED_SERVER 指令到连接 0A 的过程中出错， 则将 ERROR_0A

参数的输出设置为 “ 1 ”。 有关错误原因的详细信息， 将由 STATUS_0A 参数指定。 STATUS_0A 3 OUT

Word 连接 0A 上指令的详细状态信息。 DR_NDR_1AOUT Bool “ Data Read ” 或 “ New Data

Ready ” 至连接 1A： 0： 无新数据 1： Modbus客户端读取或写入的新数据 ERROR_1A OUT Bool

如果在调用 MB_RED_SERVER 指令到连接 1A的过程中出错， 则将 ERROR_1A

参数的输出设置为 “ 1 ”。 有关错误原因的详细信息， 将由 STATUS_1A 参数指定。 STATUS_1A 3 OUT

Word 连接 1A 上指令的详细状态信息。描述 DR_NDR_0BOUT Bool “Data Read” 或 “New Data Ready” 至连接 0B：0：无新数据 1：Modbus 客户端读取或写入的新数据 ERROR_0B OUT Bool

如果在调用 MB_RED_SERVER 指令到连接 0B 的过程中出错，则将 ERROR_0B

参数的输出设置为 “1”。有关错误原因的详细信息，将由 STATUS_0B 参数指定。STATUS_0B 3 OUT

Word 连接 0B 上指令的详细状态信息。DR_NDR_1B OUT Bool “DataRead” 或 “New Data

Ready” 至连接 1B：0：无新数据 1：Modbus 客户端读取或写入的新数据 ERROR_1B OUT Bool

如果在调用 MB_RED_SERVER 指令到连接 1B 的过程中出错，则将

ERROR_1B 参数的输出设置为 “1”。有关错误原因的详细信息，将由 STATUS_1B 参数指定。

STATUS_1B 3 OUT Word 连接 1B 上指令的详细状态信息。RED_ERR_S7 3 OUT Bool 0：SIMATIC

中无冗余错误 1：SIMATIC 中存在冗余错误 RED_ERR_S7 3 OUT Bool 0：链接伙伴侧无冗余错误

1：链接伙伴侧存在冗余错误 RED_ERR_S7 3 OUT Bool 0：至少已建立 1 个组态连接

1：完全丢失通信，所有组态连接均终止 1 有关详细信息，参见下文中“授权”部分。2

有关详细信息，参见下文中“MB_HOLD_REG 输入参数”部分。

3 有关详细信息，参见下文中“输出参数：ERROR_x、RED_ERR_S7、RED_ERR_DEV 和

TOT_COM_ERR”部分。说明 安全信息 请注意，网络中的每个客户端对过程映像输入和输出以及

Modbus 保持寄存器定义的数据块或位存储区域都具有读写访问权限。可以选择限制对某个 IP

地址的访问，从而阻止未经授权的读写操作。但请注意，共享地址也可用于未经授权的访问 CPU

可以用于：处理多个服务器连接 在同一个服务器端口同时接受多个来自不同的客户端的多个连接

连接的*大数目取决于所使用的 CPU，具体请参见相关 CPU 的技术规范。一个 CPU 的总连接数，包括

Modbus TCP

客户端和服务器的连接数，不能超过所支持的*大连接数。连接服务器时，请记住以下规则：每个

MB_RED_SERVER 连接都必须使用唯一的背景数据块。

要连接到服务器端口的每个客户端都需要一个唯一的连接/连接 ID。连接 ID 在整个 CPU

范围内必须唯一。Modbus地址到过程映像的映射 MB_RED_SERVER 指令允许到达的 Modbus

功能（1、2、4、5 和 15）直接读取和写入访问CPU的过程映像输入和输出（使用数据类型 BOOL 和

WORD）。对于 S7-1200-CPU，输入和输出过程映像的地址空间为1 KB。下表列出了上述 Modbus

功能的地址空间：Modbus 功能 功能代码 功能 数据区 地址空间 01 读取：位 Output 0 到 65.535 02

读取：位 Input 0 到 65.535 04 读取：WORD Input 0 到65.535 05 写入：位 Output 0 到 65.535 15 写入：位

Output 0 到 65.535 通过功能代码3、6、16 和 23 将到达的 Modbus 请求写入 Modbus 保持性寄存器或从寄存

器中读取（可通过 MB_HOLD_REG参数或 Data_Area_Array 指定保持性寄存器）写入输出位 06

写入保持性寄存器 08 诊断功能：回送测试（子功能0x0000）：MB_RED_SERVER 指令接收数据字并按

原样返回 Modbus 客户端。复位事件计数器（子功能0x000A）：使用指令 MB_RED_SERVER，可复位以

下事件计数器：“Success_Count”、“Xmt_Rcv_Count”、“Exception_Count”、“Server_Message_Cou

nt”和“Re”。11诊断功能：获取通信的事件计数器 MB_RED_SERVER

指令使用一个通信的内部事件计数器，记录发送到 Modbus服务器上成功执行的读写请求数。执行功能 8

或 11 时，事件计数器不会递增。这种情况同样适用于会导致通信错误的请求。例如，发生协议错误（如

，不支持所接收 Modbus 请求 中的功能代码）。15 写入多个输出位 16写入保持性寄存器 23 通过请求写

入和读取保持性寄存通信节点可设计为独立式或者冗余式。如果其中一个伙伴采用独立式设计，则将其

称为单侧冗余。如果两个伙伴均为冗余式设计，可将其称为双侧冗余。单侧冗余：-

必须为通信伙伴之间的每个连接组态一个连接。SIMATIC S7 的连接点称为 0 和 1；通

信伙伴的连接点称为 A 和 B。R-CPU 或 H-CPU 1 是指连接点 0，R-CPU 或 H-CPU 2 为连接点 1。-

组态：如果 S7 采用冗余设计，则为 S7 连接点 0 到链接伙伴的节点 A 创建一个连接（从 S7 连接点 0 到伙伴/节点 A 的连接 => 连接 0A），一个从 S7 连接点 1 到链接伙伴的节点 A 的连接（从 S7 连接点 1 到伙伴/节点 A 的连接 => 连接 1A）。下图给客户端和服务器的端口号 Modbus 客户端使用的端口号从

2000 开始。通常通过端口号 502 对 Modbus 服务器进行寻

址。可以为多个连接组态端口 502（多端口），具体取决于 CPU。如果为本地端口 502

组态至少两个连接，则在未指定连接的情况下，将发出请求的客户端随机分配给现有的服务器

连接。不会自动将想要连接到"MB_RED_SERVER"指令的第一个客户端分配给连接

0A。将客户端请求分配给服务器连接后，在连接终止前，分配在交换帧期间保持不变。参数分配对于

S7-1200，可以使用 MB_RED_SERVER 指令 V1.0 和 V1.1。CPU 通过 CPU 或 CM/CP 的

本地接口建立连接。CPU 使用 TCON_IP_V4 结构组态和建立连接。MB_RED_SERVER 的组态：使用

MB_RED_SERVER 指令的组态对话框进行以下设置：连接 0A、1A、0B 和 1B

的连接参数（有关冗余组态的更多信息，请参见上文的“运行和冗余”。）内部参数（可选）通过

MB_RED_SERVER

指令或通过工艺对象，可以打开组态对话框 Modbus 主站可写入输出的寻址范围的起始地址（0 到 65535

个字节）UINT 0 Modbus 主站可写入的输出字节数。示例：=0 和 =10：Modbus 主

站可写入的输出字节数为 0 到 9。=0：Modbus 主站无法写入任何输出字节。UINT 0

Modbus主站可读取输出的寻址范围的起始地址（0到65535个字节）UINT 0

Modbus主站可读取的输出字节数。示例：`=0`和`=10`：Modbus主站可读取的输出字节数为0 to 9。

`=0`：Modbus主站无法读取任何输出字节。IB_Read_Start UINT 0 Modbus

主站可读取输入的寻址范围的起始地址（0到65535个字节）a_type参数指定 Modbus

主站在该数据块中映射的 MODBUS 数据类型。如果在 data_type 中输入值“0”，则

Modbus主站不使用相应的数据区域。如果 Modbus 主站要使用多个

Data_Area，则必须依次定义这些区域。Modbus 主站将不会处理 data_type = 0 之后的任何条目。标识符

数据类型 描述 0 未使用区域 1 输出位（线圈）位 2 输入位（输入）位 3 保持性寄存器 字 4

输入字（输入寄存器）字 db db 参数指定映射 MODBUS

寄存器或下面定义的位值的数据块。数据块编号0为系统保留，CPU不允许使用。start, length start 指定

Modbus 主站在数据块的数据字 0 中映射的第一个 Modbus 地址。length 参数定义了 Modbus

主站在数据块中映射的 MODBUS 地址的数量和长度。定义的数据区不得重叠。length 参数不得等于 0。

示例：使用 Data_Area_Array 映射地址。数据区 1 data_type 3：保持性寄存器 db 11 起始 0 长度 500 数据区

2 data_type 3：保持性寄存器 必须在每个 CPU 上分别对 MB_RED_SERVER

指令进行授权，且该指令需要付费。授权分为两个步骤：显示许可证 IDENT_CODE 输入 REG_KEY

注册密钥：必须在每个 MB_RED_SERVER 指令中分配 REG_KEY 注册密钥。将

REG_KEY 保存在共享数据块中，所有 MB_RED_SERVER 指令均可通过该数据块接收所需的注册密钥。

显示许可证 IDENT_CODE 的程序：1. 根据循环 OB 中的要求，为 MB_RED_SERVER

指令分配参数。将程序下载到 CPU，并将 CPU 设置为 RUN。2. 打开 Modbus

指令的背景数据块，然后单击“监视全部”(Monitor all)按钮。3. 背景数据块将在IDENT_CODE

输出中显示一个18位的字符串。输入注册密钥REG_KEY的程序：1. 通过“添加新块...”(Addnew

block...)插入具有唯一符号名的全新共享数据块，例如“License_DB”。2.

在该块中创建数据类型为STRING[17]的REG_KEY参数。图13-25 REG_KEY 3.

使用复制/粘贴将已发送的17位注册密钥复制到“起始值”(Start value)列。4. 在循环OB中，在

MB_RED_SERVER指令的REG_KEY参数中，输入许可证数据块的名称和字符串名称(例如

License_DB.REG_KEY)。5. 将修改后的块下载到CPU。可以在运行时间输入注册密钥；不需要从STOP

更改RUN。6. 现在，该CPU已获得使用MB_RED_SERVER指令进行Modbus/TCP

通信的授权；LICENSED输出位为TRUE。授权缺失或不正确的纠正程序：

如果输入的注册密钥错误或未输入注册密钥，则CPU的ERROR LED指示灯闪烁。此外，S7-1200

CPU会在诊断缓冲区中生成有关许可证缺失的循环条目。REG输入参数MB_HOLD_REG

参数是一个指向数据缓冲区的指针，该缓冲区用于存储Modbus

客户端对其具有读写访问权限的数据。可使用全局数据块(D)或位存储器(M)作为存储区：数据块(D)

中地址的*大数量取决于CPU的*大数据块大小。位存储器(M)中地址的*大数量取决于CPU

的*大位存储区。下图说明了针对Modbus功能3(读取多个WORD)、6(写入一个

WORD)、16(写入多个WORD)和23(读写多个WORD)将

Modbus地址映射到保持性寄存器的情况。数ERROR_x、STATUS_x、RED_ERR_S7、RED_ERR_DEV

和TOT_COM_ERR CPU将错误消息显示在MB_RED_SERVER指令的状态输出中：

说明在程序编辑器中，错误状态代码可显示为整数或十六进制值：1. 在程序编辑器中打开所需的块。

2.单击“启用/禁用监视”(Monitor on/off)切换程序状态的显示格式。(如果尚未建立在线连接,则打开“转至在线”(Goonline)对话框。在此对话框中,可以建立在线连接。)3.

选择待监视的变量,然后在快捷菜单的“显示格式”(Displayformat)中,选择所需的显示格式。

STATUS_x 参数(常规状态信息) STATUS(W#16#) 描述 0000指令已执行,且无任何错误。 0001 连接已建立。 0003 连接已终止。 0A90

MB_RED_SERVER指令未授权。有关详细信息,参见上文中“授权”部分。 0AFF

连接未组态且未使用。必须组态 0A 连接。 7000未激活任何调用且未建立任何连接 (REQ=0,

DISCONNECT=1)。 7001 首次调用。已触发连接建立操作。 7002中间调用。正在建立连接。 7003

正在终止连接。 7005 正在发送数据。 7006 正在接收数据。 STATUS_x参数(协议错误) STATUS

(W#16#) 来自 MB_RED_SERVER 的错误消息中的错误代码 (B#16#) 描述8380 - 接收到的 Modbus

帧格式错误或字节数过少。 8381 01 功能代码不受支持。数据长度错误: 接收到的Modbus 帧长度无效

Modbus 帧头中输入的帧长度与实际接收到的字节数不一致。 Modbus帧头中输入的字节数与实际接收到

的字节数不一致(功能 15 和 16)。 8383 02 数据地址错误或访问了保持性寄存器 (MB_HOLD_REG

参数) 地址以外的区域。有关详细信息,请参见上文“MB_HOLD_REG”部分。 8384

03数据值错误(功能 05)。 8385 03 诊断代码不受支持 (**功能 08)。 STATUS_x 参数(参数错误)

STATUS(W#16#) 描述 80BB 参数 ActiveEstablished 的值无效

只允许对服务器建立被动连接(active_established = FALSE)。 8187 参数 MB_HOLD_REG

的指针无效。数据区过小。 8389数据区域定义无效: data_type 值无效 数据块编号无效或不存在: - db

值无效 - 数据块编号不存在 - 数据块编号已由另一个数据区域使用 - 数据块具有优化访问权限 -

数据块不在工作存储器中 length 值无效 属于同一MODBUS 数据类型的 MODBUS 地址范围重叠