



S7-1200 基本以太网通信使用指南

S7-1200 Basic Ethernet Communication User Guide

User Guide

Edition (2010 年 1 月)

摘 要 本文介绍了 S7-1200 以太网通信基本知识及使用注意事项。

关键词 以太网, S7-1200, TCP, ISO-on-TCP, 通信指令

Key Words Ethernet, S7-1200, TCP, ISO-on-TCP, Communication instructions

目 录

S7-1200 基本以太网通信使用指南	1
1. 概述	4
2. 两种协议的特点	4
2.1 传输控制协议: TCP	4
2.2 基于TCP的ISO传输服务的协议: ISO-on-TCP	4
3. 传输数据长度与协议的应用	5
3.1 对于TCP 协议	5
3.2 对于ISO on TCP协议	5
4. 通信连接的参数	7
4.1 连接描述数据块	7
4.2 IP 地址	9
4.3 端口号的分配	10
4.4 TSAP结构	11
5. 回读连接描述参数功能	12
5.1 更改连接描述中的参数值	12
5.2 回读各个连接参数的功能	13
6. 开放式用户通信的指令	13
6.1 连接建立	13
6.2 连接参数分配	13
6.3 启动连接参数分配	15
6.4 创建和分配连接参数	16
6.5 删除连接	17
6.6 TSEND_C	18
6.7 TRCV_C	19
6.8 TCON	20
6.9 TDISCON	20
6.10 TSEND	21
6.11 TRCV	21
7. S7-1200 以太网通信连接数	22
附录一推荐网址	23

1. 概述

S7-1200 CPU 具有一个集成的以太网接口，支持面向连接的以太网传输层通信协议。协议会在数据传输开始之前建立到通信伙伴的逻辑连接。数据传输完成后，这些协议会在必要时终止连接。面向连接的协议尤其适用于注重可靠性的数据传输。一条物理线路上可以存在多个逻辑连接（8 个）。

开放式用户通信支持以下连接类型：

- TCP
- ISO-on-TCP

对于不支持 ISO-on-TCP 连接的通信伙伴，应使用 TCP 连接。对于诸如第三方设备或 PC 这些类型的通信伙伴，请在分配连接参数时，为伙伴端点输入“未指定”。

2. 两种协议的特点

2.1 传输控制协议: TCP

TCP 是由 RFC 793 描述的一种标准协议：传输控制协议。TCP 的主要用途是在过程之间提供可靠、安全的连接服务。该协议有以下特点：

- 由于它与硬件紧密相关，因此它是一种高效的通信协议；
- 它适合用于中等大小或较大的数据量（最多 8K 字节）；
- 它为应用带来了更多的便利，比如**错误恢复**，**流控制**，**可靠性**，这些是由传输的报文头进行确定的；
- 一种**面向连接**的协议；
- 非常灵活地用于只支持 TCP 的第三方系统；
- 有**路由**功能；
- 应用**固定长度**数据的传输；
- 发送的数据报文会被确认；
- 使用端口号对应用程序寻址；
- 大多数用户应用协议（例如 TELNET 和 FTP）都使用 TCP。

2.2 基于 TCP 的 ISO 传输服务的协议: ISO-on-TCP

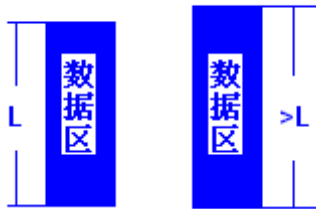
ISO on TCP 是一种能够将 ISO 应用移植到 TCP/IP 网络的机制。该协议有以下特点：

- 与硬件关系紧密的高效通信协议；
- 适合用于中等大小或较大的数据量（最多 8K 字节）；
- 与 TCP 相比，它的消息提供了**数据结束标识符**并且它是面向消息的；
- 具有路由功能；可用于 WAN；
- 可用于实现**动态长度**数据传输；
- 由于使用 SEND/RECEIVE 编程接口的缘故，需要对数据管理进行编程；
- 通过传输服务访问点 (TSAP, Transport Service Access Point)，TCP 协议允许有多个连接访问单个 IP 地址（最多 64K 个连接），借助 RFC 1006，TSAP 可唯一标识与同一个 IP 地址建立通信的端点连接。

3. 传输数据长度与协议的应用

3.1 对于 TCP 协议

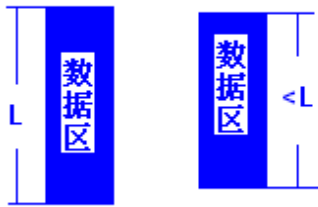
PLC1 发送侧 PLC2 接收侧



始。

如果要接收的数据的长度（参数 LEN，指令 TRCV/TRCV_C）大于要发送的数据的长度（参数 LEN，指令 TSEND/TSEND_C）：仅当达到所分配的长度后，TRCV/TRCV_C 才会将接收到的数据复制到指定的接收区（参数 DATA）。达到所分配的长度时，已经接收了下一个作业的数据。因此，接收区包含的数据来自两个不同的发送作业。如果不知道第一条消息的确切长度，将无法识别第一条消息的结束以及第二条消息的开

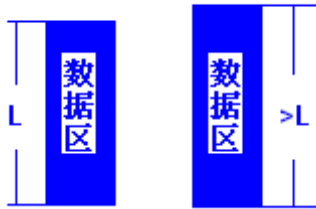
PLC1 发送侧 PLC2 接收侧



如果要接收的数据的长度（参数 LEN，指令 TRCV/TRCV_C）小于要发送的数据的长度（参数 LEN，指令 TSEND/TSEND_C）：TRCV/TRCV_C 将 LEN 参数中指定字节的数据复制到接收数据区（参数 DATA）。然后，将 NDR 状态参数设置为 TRUE（作业成功完成）并将 LEN 的值分配给 RCVD_LEN（实际接收的数据量）。对于每次后续调用，都会接收已发送数据的另一个块。

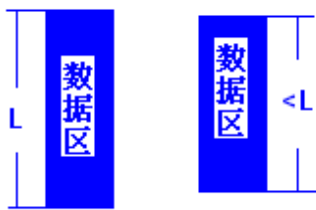
3.2 对于 ISO on TCP 协议

PLC1 发送侧 PLC2 接收侧



如果要接收的数据的长度（参数 LEN，指令 TRCV/TRCV_C）大于要发送的数据的长度（参数 LEN，指令 TSEND/TSEND_C）：TRCV/TRCV_C 将所有发送数据复制到接收数据区（参数 DATA）。然后，将 NDR 状态参数设置为 TRUE（作业成功完成）并将所发送数据的长度分配给 RCVD_LEN（实际接收的数据量）。

PLC1 发送侧 PLC2 接收侧



如果要接收的数据的长度（参数 LEN，指令 TRCV/TRCV_C）小于要发送的数据的长度（参数 LEN，指令 TSEND/TSEND_C）：TRCV/TRCV_C 并不会将任何数据复制到接收数据区（参数 DATA），而是提供以下错误信息：ERROR=1，STATUS=W#16#8088（目标缓冲区太小）。

总结：可以根据传输数据的长度选择使用的通信协议。

数据长度比较 没TSEND_C/TSEND TRCV_C/TRCV	支持协议
>	TCP
<	ISO-on-TCP
=	TCP/ISO-on-TCP

表1：传输数据长度与支持协议关系

4. 通信连接的参数

4.1 连接描述数据块

结构符合 TCON_Param 的连接描述数据块用于分配 TCP 和 ISO-on-TCP 的通信连接参数。TCON_Param 的固定数据结构包含了建立连接所需的全部参数。使用 TSEND_C、TRCV_C 或 TCON 指令时，将根据开放式用户通信的连接参数分配情况自动为新连接创建连接描述数据块。TSEND_C、TRCV_C 或 TCON 相应背景数据块中的 CONNECT 连接参数包含对所用数据块的引用。

字节	参数	数据类型	初始值	描述
0 ... 1	block_length	UINT	64	长度： 64 个字节（固定）
2 ... 3	id	CONN_OUT_C	1	对该连接的引用（取值范围： 1 - 4095）。 对于 TSEND_C、TRCV_C 或 TCON 指令，在 ID 中指定该参数的值。
4	connection_type	USINT	17	连接类型： 17: TCP 18: ISO-on-TCP
5	active_est	BOOL	TRUE	建立连接的方式所对应的 ID： FALSE： 被动连接建立 TRUE： 主动连接建立
6	local_device_id	USINT	2	本地 PN/IE 接口的 ID。
7	local_tsap_id_len	USINT	0	所用参数 local_tsap_id 的长度（字节）；可能值： 0 或 2（连接类型 = 17 (TCP) 时） 主动端只允许使用值 0。 2 到 16（连接类型 = 18 (ISO-on-TCP) 时）
8	rem_subnet_id_len	USINT	0	该参数未使用。
9	rem_staddr_len	USINT	4	伙伴端点的地址长度（字节）： 0: 未指定，即参数 rem_staddr 不相关。 4: 参数 rem_staddr 中有效的 IP 地

				址
10	rem_tsap_id_len	USINT	2	所用参数 rem_tsap_id 的长度（字节）；可能值： 0 或 2（连接类型 = 17 (TCP) 时） 被动端只允许使用值 0。 2 到 16（连接类型 = 18 (ISO-on-TCP) 时）
11	next_staddr_len	USINT	0	该参数未使用。
12 ... 27	local_tsap_id	BYTE 类型的 ARRAY [1..16]	-	连接的本地地址部分： 17 (TCP) : 本地端口号（可能值： 1 到 49151；建议值：2000...5000）； local_tsap_id[1] = 端口号的高位字节 （十六进制表示）； local_tsap_id[2] = 端口号的低位字节 （十六进制表示）； local_tsap_id[3-16] = 不相关 18 (ISO-on-TCP) : 本地 TSAP ID: local_tsap_id[1] = B#16#E0; local_tsap_id[2] = 本地端点的机架和 插槽（位 0 到 4: 插槽号；位 5 到 7: 机架号）； local_tsap_id[3-16] = TSAP 扩展 （可选） 注意：请确保 local_tsap_id 的每个 值在 CPU 中都是唯一的。
28 ... 33	rem_subnet_id	USINT 类型 的 ARRAY [1..6]	-	该参数未使用。
34 ... 39	rem_staddr	USINT 类型 的 ARRAY [1..6]	-	伙伴端点的 IP 地址，例如， 192.168.002.003: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = 不相关
40 ... 55	rem_tsap_id	BYTE 类型 的 ARRAY [1..16]	-	连接的伙伴地址部分 17 (TCP) : 伙伴端口号（可能值： 1 到 49151；建议值：2000...5000）； rem_tsap_id[1] = 端口号的高位字 节（十六进制表示）； rem_tsap_id[2] = 端口号的低位字节 （十六进制表示）； rem_tsap_id[3-16] = 不相关 18 (ISO-on-TCP) : 伙伴 TSAP- ID: rem_tsap_id[1] = B#16#E0;

				rem_tsap_id[2] = 伙伴端点的机架和插槽（位 0 到 4：插槽号；位 5 到 7：机架号）； rem_tsap_id[3-16] = TSAP 扩展（可选）
56 ... 61	next_staddr	BYTE 类型的 ARRAY [1..6]	-	该参数未使用。
62 ... 63	spare	WORD	W#16# 0000	保留。

表 2: 符合 TCON_Param 的连接描述结构

4.2 IP 地址

如果具有通信功能的模块支持 TCP/CP 协议，则 IP 参数可见。通常对于所有以太网模块都是这样。IP 地址由 4 个 0 到 255 之间的十进制数字组成。各十进制数字相互之间用点隔开。例如：140.80.0.2

IP 地址包括

- （子）网的地址
- 节点的地址（通常也称为主机或网络节点）

子网掩码将这两个地址拆分。它确定 IP 地址的哪一部分用于网络定址，哪一部分用于节点定址。子网掩码的设置位确定 IP 地址的网络部，例如：子网掩码：255.255.0.0 = 11111111.11111111.00000000.00000000在针对上述 IP 地址给出的实例中，此处显示的子网掩码具有以下含义：IP 地址的前 2 个字节标识子网 - 即 140.80。最后的两个字节标识节点，例如 0.2。这样用 **AND** 连接 IP 地址和子网掩码就产生网络地址，用 **AND NOT** 连接 IP 地址和子网掩码就产生节点地址。

在 IP 地址范围的分配和所谓的“默认子网掩码”方面，存在一个共识。IP 地址中的第一个十进制数字（从左边起）决定默认子网掩码的结构。如下所示，它决定数值“1”（二进制）的个数：

IP 地址（十进制）	IP 地址（二进制）	地址类别	默认子网掩码
0 到 126	0xxxxxxx.xxxxxxxx....	A	255.0.0.0
128 到 191	10xxxxxx.xxxxxxxx....	B	255.255.0.0
192 到 223	110xxxxx.xxxxxxxx...	C	255.255.255.0

表3: IP地址及默认子网掩码

注意：IP 地址的第一个十进制数字也可以是 224 和 255 之间的值（地址类别 D 等）。但由于对这些值不进行地址检查，因此不建议使用该方法。

屏蔽其它子网可使用子网掩码添加更多结构并为被指定了地址类别 A、B 或 C 之一的子网形成“专用”子网。这通过将子网掩码的其它低位部分设置为“1”实现。每将一个位设置为“1”，“专用”网络的数目就会加倍，而它们包含的节点数将减半。在外部，该网络像以前那样，以单个网络的方式运行。

例如：有一个地址类别为 B 的子网（例如 IP 地址 129.80.xxx.xxx）将默认子网掩码为由 255.255.0.0改为255.255.128.0，则产生结果是地址在 129.80.001.xxx 和129.80.127.xxx 之间的所有节点都位于一个子网上，地址在 129.80.128.xxx 和 129.80.255.xxx 之间的所有节点都位于另一个子网上。

4.3 端口号的分配

创建开放式用户通信时，系统会自动分配值 **2000** 作为端口号。

端口号的允许值为 **1** 到 **49151**。可以分配该范围内的任何端口号。但是，由于某些端口已被使用（取决于系统），因而建议使用 **2000** 到 **5000** 范围内的端口号。

端口号	描述	系统响应
2000 ... 5000	建议范围	不出现警告或错误消息 允许使用并且接受端口号
1 ... 1999, 5001 ... 49151	可以使用，但不在建议范围内	会出现警告消息 允许使用并且接受端口号
20, 21, 25, 80, 102, 135, 161, 34962 ... 34964	在一定条件下可以使用 采用 TCP 连接类型时，这些端口由 TSEND_C 和 TRCV_C 使用。	
53, 80, 102, 135, 161, 162, 443, 520, 9001, 34962 ... 34964	在一定条件下可以使用 这些端口是否封锁取决于所用 S7-1200 CPU 的功能范围。 相应 CPU 的文档中提供了这些端口的分配信息。	

表4：系统对各个端口号的响应。

4.4 TSAP 结构

对于 ISO-on-TCP 连接，必须同时为两个通信伙伴分配传输服务访问点 (TSAP, Transport Service Access Point)。创建了 ISO-on-TCP 连接后，系统会自动分配 TSAP ID。要确保 TSAP ID 在设备中唯一，可以在连接参数分配中更改预分配的 TSAP。

分配 TSAP 时，必须遵守某些规则。TSAP 必须包含一定数量的字节，这些字节可以十六进制值 (TSAP-ID) 或 ASCII 字符 (ASCII-TSAP) 的形式显示和输入：

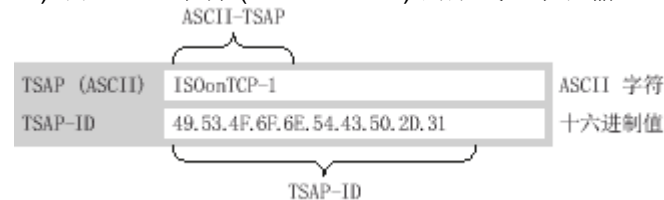


图 1：TSAP 结构

如果 TSAP 包含无效的 ASCII 字符，则 TSAP 将只显示为 TSAP-ID，而不显示为 ASCII-TSAP。创建连接后就是如此。前两个十六进制字符作为 TSAP-ID，用来标识通信类型和机架/插槽。因为这些字符对于 CPU 是无效 ASCII 字符，所以在这种情况下不显示 ASCII-TSAP。

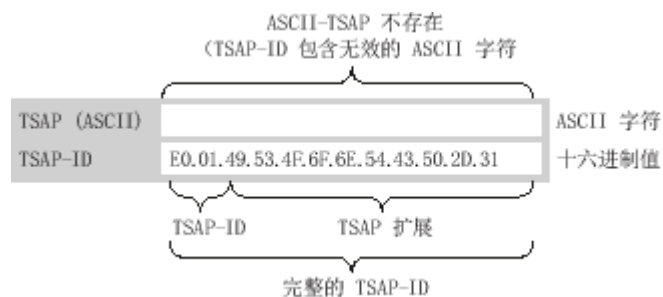


图 2：含无效的 ASCII 字符 TSAP 结构

除了遵守 TSAP 在长度和结构方面的规则外，还必须确保 TSAP-ID 是唯一的。所分配的 TSAP 不会自动具有唯一性。

TSAP 的长度和内容

- 具有 TSAP 扩展的 TSAP-ID 长度 = 2 到 16 个字节
 - $x_tsap_id[0] = 0xE0$ (开放式用户通信)
 - $x_tsap_id[1]$ (位 0 到 4) = CPU 的插槽号
 - $x_tsap_id[1]$ (位 5 到 7) = CPU 的机架号
 - $x_tsap_id[2...15]$ = 任何字符 (TSAP 扩展, 可选)
 - ($x = loc$ (本地) 或 $x = rem$ (伙伴))
- ASCII-TSAP 形式的 TSAP-ID 长度 = 3 到 16 个字节
 - $x_tsap_id[0 \text{ 到 } 2] = 3$ 个 ASCII 字符 (0x20 到 0x7E)
 - $x_tsap_id[3...15]$ = 任何字符 (可选)
 - ($x = loc$ (本地) 或 $x = rem$ (伙伴))

表 5：TSAP-ID 的示意性结构

TSAP-ID	tsap_id_len	tsap_id[0]	tsap_id[1]	tsap_id[2..15]
...具有扩展	2 到 16 个字节	0xE0	0x01 (0x00)*	扩展 (可选)
...ASCII-TSAP 形式	3 到 16 个字节	0x20...0x7E	0x20...0x7E	0x20...0x7E

* 已识别的 CPU 通常插在机架 0 的插槽 1 中。因此，十六进制值 01 对具有扩展 TSAP-ID 的第二个位置有效。如果连接伙伴是未指定的 CPU（例如第三方设备），则还允许对插槽地址使用十六进制值 00。

对于未指定的通信伙伴，本地 TSAP-ID 和伙伴 TSAP-ID 的长度可以为 0 到 16 个字节，其中允许使用 00 到 FF 之间的所有十六进制值。

在连接参数分配中输入 ASCII-TSAP 时，只允许使用 20 到 7E 之间的十六进制值：

代码	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
2..		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3..	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4..	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5..	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6..	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7..	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

表 6：用于输入 ASCII TSAP 的 ASCII 代码表

5. 回读连接描述参数功能

5.1 更改连接描述中的参数值

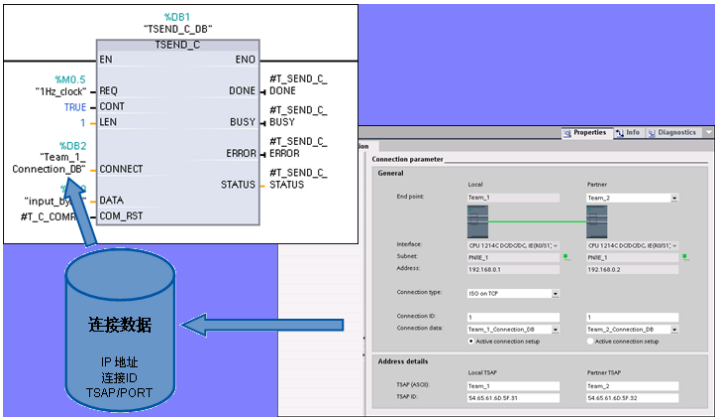


图 3：连接参数组态

与开放式用户通信的连接一一对应的连接描述通过连接参数分配输入到连接描述数据块中。可以不通过连接参数分配在用户程序中更改连接描述数据块的参数值。不能更改连接描述的结构。随后可通过连接参数分配回读包含更改值的连接描述数据块。在“属性 > 组态 > 连接参数”下，巡视窗口仅显示连接描述数据块中所存储的连接参数。连接参数分配不支持（只能通过偏移引用来进行查找的）数据块类型（例如，全局数据块）的连接描述的嵌套输入。

5.2 回读各个连接参数的功能

对于通信伙伴的“地址”(Address) 参数, 所显示的 IP 地址来自连接描述的“rem_staddr”参数。

以下值也可以从连接描述重新装载:

- 连接类型
- 本地连接 ID
- 连接建立 (主动/被动)
- 本地 TSAP (仅限 ISO-on-TCP)
- 伙伴 TSAP (仅限 ISO-on-TCP)
- 本地端口 (仅限 TCP)
- 伙伴端口 (仅限 TCP)

通信伙伴的连接 ID 参数值、连接数据以及连接建立, 都不包含在本地连接描述数据块的连接描述中。因此, 在重新打开连接参数分配时无法显示这些参数。伙伴的连接建立是由本地连接建立而产生的, 所以也会显示出来。可随时在“伙伴”下拉列表框中选择新的通信伙伴。将项目中识别的 CPU 选为指定的通信伙伴时, 连接 ID 和连接数据相应的输入选项将再次显示。

6. 开放式用户通信的指令

要创建连接, 在打开程序编辑器后, 可使用“指令 > 扩展指令 > 通信”任务卡中提供的各种指令:

- 用于发送和接收数据并集成了连接建立/终止功能的简化指令:
 - TSEND_C (连接建立/终止, 发送)
 - TRCV_C (连接建立/终止, 接收)
- 单独用于发送或接收数据或者用于建立或终止连接的指令:
 - TCON (连接建立)
 - TDISCON (连接终止)
 - TSEND (发送)
 - TRCV (接收)

6.1 连接建立

对于开放式用户通信, 两个通信伙伴都必须具有用来建立和终止连接的指令。其中一个通信伙伴通过 TSEND 或 TSEND_C 发送数据, 而另一个通信伙伴通过 TRCV 或 TRCV_C 接收数据。

其中一个通信伙伴作为主动方启动连接建立过程。另一个通信伙伴通过作为被动方启动连接建立过程来进行响应。如果两个通信伙伴都触发了连接建立过程, 操作系统便完全能够建立通信连接。

6.2 连接参数分配

可按如下方式使用具有 TCON_Param 结构的连接描述 DB 来分配参数以建立连接:

- 手动创建、分配参数并直接写入指令。
- 使用属性窗口分配连接参数。

在检查窗口属性栏中配置连接参数如下:

- 连接伙伴
- 连接类型
- 连接 ID
- 连接描述 DB
- 与所选连接类型相应的地址详细信息

注意：连接参数分配不会检查连接 ID 和端口号 (TCP) 或 TSAP (ISO-on-TCP) 是否唯一。因此，组态开放式用户通信时，应确保参数设置在设备内是唯一的。

参数	描述
端点	<p>显示本地端点和伙伴端点的名称。</p> <p>本地端点就是为其设置 TCON、TSEND_C 或 TRCV_C 的 CPU。因此，本地端点始终是已知的。</p> <p>伙伴端点则需要从下拉列表中选择。下拉列表将显示所有可用的连接伙伴，包括那些项目中还未知其数据的设备对应的未指定的连接伙伴。</p> <p>只要未设置任何连接伙伴，就会禁用其它所有被屏蔽的参数。</p>
接口	显示本地端点的接口。只有指定伙伴端点后，才会显示伙伴接口。
子网	<p>显示本地端点的子网。只有选择伙伴端点后，才会显示伙伴子网。</p> <p>如果所选伙伴端点未通过子网连接到本地端点，则会自动将两个连接伙伴联网。为此，必须指定伙伴端点。</p> <p>不同子网中的伙伴之间只能通过 IP 路由建立连接。可在相关的接口属性中编辑路由设置。</p>
地址	<p>显示本地端点的 IP 地址。只有选择伙伴端点后，才会显示伙伴的 IP 地址。</p> <p>如果选择了未指定的连接伙伴，输入框将为空并且背景为红色。在这种情况下，需要指定有效的 IP 地址。</p>
连接类型	<p>从“连接类型”(Connection type) 下拉列表选择要使用的连接类型：</p> <ul style="list-style-type: none"> • TCP • ISO-on-TCP <p>所需连接数据的参数会因所选连接类型的不同而变化。</p>
连接 ID	<p>在输入框中输入连接 ID。创建新连接时，会分配默认值 1。可以在输入框中更改连接 ID，也可以在 TCON 中直接输入连接 ID。</p> <p>请确保所分配的连接 ID 在设备内是唯一的。</p>
连接数据	<p>下拉列表中将显示其连接描述结构符合 TCON_Param 的连接描述 DB 的名称。</p> <p>创建连接时，将为指定的每个连接伙伴生成一个数据块，并会用连接参数分配</p>

	<p>的值自动填充该数据块。对于本地连接伙伴，所选数据块的名称将自动输入所选 TSEND_C、TRCV_C 或 TCON 指令的块参数 CONNECT 中。</p> <p>对于另一个连接伙伴，也可以在 TSEND_C、TRCV_C 或 TCON 指令的 CONNECT 输入中直接使用第一个连接伙伴所生成的连接描述 DB。对于本步骤，可在选择第一个连接伙伴后使用现有的连接描述 DB，或创建新的连接描述 DB。</p> <p>也可以从下拉列表中引用另一个有效数据块。如果使用 TSEND_C、TRCV_C 或 TCON 扩展指令的 CONNECT 输入参数引用数据块，而该数据块的结构与 TCON_Param 的结构不符，则下拉列表将不显示任何内容，且背景为红色。</p>
主动连接建立	启用“主动连接建立”(Active connection establishment) 复选框，可指定开放式用户通信的主动方。
端口	TCP 连接的地址部分。创建新的 TCP 连接之后的默认值为 2000。
(仅限 TCP)	<p>可以更改端口号。</p> <p>端口号必须在设备中唯一！</p>
TSAP	ISO-on-TCP 连接的地址部分。创建新的 ISO-on-TCP 连接之后的默认值为 E0.01.49.53.4F.6F.6E.54.43.50.2D.31。
(仅限 ISO-on-TCP)	<p>可输入具有扩展的 TSAP-ID 或者输入 ASCII TSAP。</p> <p>TSAP 必须在设备中唯一！</p>

表 7：常规的连接参数

6.3 启动连接参数分配

只要在程序块中选择了用于通信的 **TCON**、**TSEND_C** 或 **TRCV_C** 指令，便会启用开放式用户通信的连接参数分配。

具体步骤如下：

- 打开任务卡、窗格和文件夹“指令 > 扩展指令 > 通信”；
- 将指令（**TSEND_C**、**TRCV_C** 和在“其它”子文件夹中的 **TCON**）之一拖到程序段中，将打开“调用选项”对话框；
- 在“调用选项”对话框中，编辑背景数据块的属性，可更改默认名称或选中手动复选框分配编号；
- 单击“确定”。

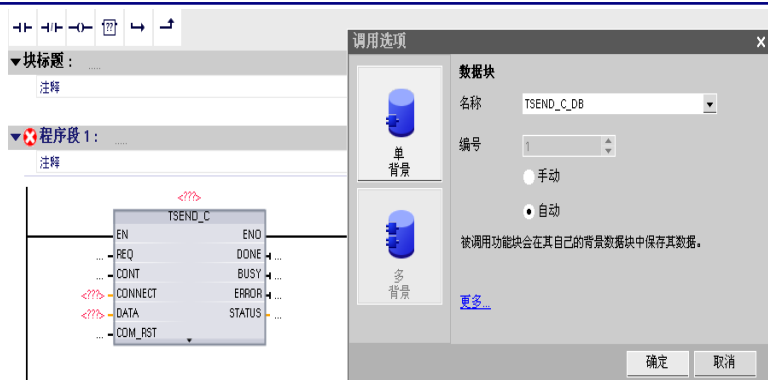


图 4：调用选项

这样就创建一个根据 TCON_Param 构造的连接描述 DB，且它是所插入指令的背景数据块。选中 TSEND_C、TRCV_C 或 TCON 时，可在巡视窗口的“属性”下看到“组态”选项卡。之后，用户可以在区域导航的“连接参数”组中进行连接参数分配。

6.4 创建和分配连接参数

在开放式用户通信的连接参数分配中，可创建 TCP 或 ISO-on-TCP 类型的连接并设置参数。具体步骤如下：

- 在程序编辑器中，选择开放式用户通信的 TCON、TSEND_C 或 TRCV_C 块。
- 在巡视窗口中，打开“属性 > 组态”选项卡。
- 选择“连接参数”组。在选择连接伙伴之前，只启用了伙伴端点的空下拉列表。其它所有输入选项均被禁用。同时显示一些已知的连接参数：本地端点的名称、本地端点的接口、本地端点的 IP 地址、连接 ID、包含连接数据的数据块的唯一名称和作为主动连接伙伴的本地端点。
- 在伙伴端点的下拉列表框中选择连接伙伴。可以选择项目中未指定的设备或 CPU 作为通信伙伴。随后会自动输入一些特定的连接参数。现有伙伴将自动与本地端点组网，同时会为伙伴 CPU 创建一个数据块，该数据块是根据 TCON_Param 为连接数据构造的。用户需要设置以下参数：伙伴端点的接口、本地子网和伙伴子网的名称、伙伴端点的 IP 地址、连接类型、连接 ID 和包含连接数据的数据块的唯一名称。

若是未指定的伙伴，请设置以下参数：TCP 连接类型和端口号 2000。

- 从相关下拉列表中选择所需的连接类型 TCP 或 ISO-on-TCP，地址详细信息将根据连接类型在端口号 (TCP) 和 TSAP (ISO-on-TCP) 之间进行切换。
- 在连接伙伴的相应输入框中，输入连接 ID。不能为未指定的伙伴分配任何连接 ID。
- 可在相应的“连接数据”下拉列表中选择其它连接描述 DB，也可以更改连接描述 DB 的名称以创建新的数据块。
- 使用“建立主动连接”复选框设置连接建立行为。用户可以决定由哪个通信伙伴主动建立连接。
- 可以编辑地址详细信息中的输入框。根据所选的协议，可以编辑端口 (TCP) 或 TSAP (ISO-on-TCP)。



图 5：分配连接参数

连接参数分配将立即检查更改后的值是否存在输入错误，然后将值输入连接描述数据块中。

注意：

可以在所选 TCON、TSEND_C 或 TRCV_C 指令的 CONNECT 输入参数互连中查看所选的数据块。如果已使用 TCON、TSEND_C 或 TRCV_C 指令的 CONNECT 参数为连接伙伴指定了连接描述 DB，则可使用此 DB 或创建一个新 DB。如果连编辑接数据下拉列表中所显示的数据块的名称，则会生成一个新数据块用于该连接，新数据块使用更改的名称，但结构和内容不变。更改的数据块名称必须在通信伙伴环境中唯一。连接描述 DB 必须具有 TCON_Param 结构。无法为未指定的伙伴选择数据块。

只有在将伙伴端点的程序段下载到硬件后，两个通信伙伴之间的开放式用户通信才能进行工作。要实现功能完整的通信，应确保在设备上不仅下载了本地 CPU 的连接描述，而且还要下载伙伴 CPU 的连接描述。

6.5 删除连接

为开放式用户通信所创建的连接数据存储在连接描述 DB 中。通过删除包含连接描述的数据块，便可删除连接。具体步骤如下：

- 在项目树中，选择开放式用户通信的通信伙伴。
- 打开所选通信伙伴下方的“程序块”文件夹。
- 从包含连接参数分配的数据块的右键快捷菜单中选择“删除”命令。

注意：如果不确定要删除哪个块，请打开扩展指令 TCON、TSEND_C 或 TRCV_C。找到 CONNECT 输入参数使用的数据块名称，或者连接参数分配中“连接数据”参数对应的数据块名称。如果仅删除扩展指令 TCON、TSEND_C 或 TRCV_C 的背景数据块，并不能一同删除所分配的连接。

6.6 TSEND_C

图 4: 分配连接参数

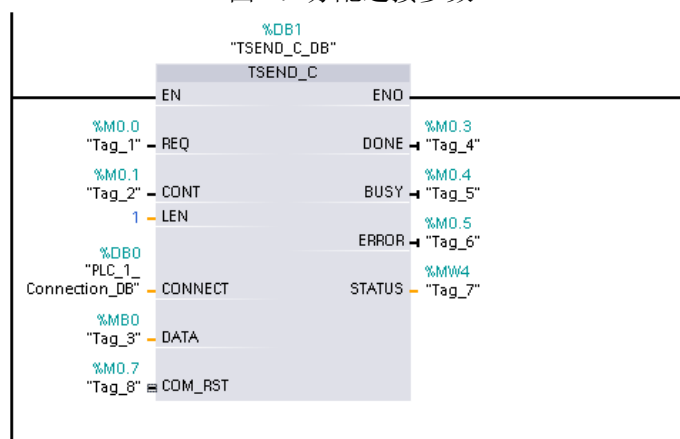


图 6: TSEND_C 函数块

TSEND_C 是异步指令且具有以下功能:

- 设置并建立通信连接:
TSEND_C 可设置并建立 TCP 或 ISO-on-TCP 通信连接。设置并建立连接后, CPU 会自动保持和监视该连接。在参数 **CONNECT** 中指定的连接描述用于设置通信连接。要建立连接, 参数 **CONT** 的值必须设置为“1”。连接成功建立后, 参数 **DONE** 在一个周期内设置为“1”。若 CPU 转到 **STOP** 模式, 将终止现有连接并删除所设置的相应连接。必须再次执行 TSEND_C, 才能重新设置并建立该连接。
- 通过现有通信连接发送数据:
通过参数 **DATA** 可指定要发送的区域。这包括要发送数据的地址和长度。在参数 **REQ** 中检测到上升沿时执行发送作业。使用参数 **LEN** 指定通过一个发送作业可发送的最大字节数。在发送作业完成前不允许编辑要发送的数据。如果发送作业成功执行, 则参数 **DONE** 将设置为“1”。参数 **DONE** 的信号状态为“1”并不表示是确认通信伙伴已读取发送数据。
- 终止通信连接:
参数 **CONT** 设置为“0”时, 将终止通信连接。
- 通信复位:
参数 **COM_RST** 设置为“1”时, 将再次执行 TSEND_C。这会终止现有通信连接并建立新连接。如果再次执行该指令时正在传送数据, 可能会导致数据丢失。

参数描述及状态代码描述参见系统手册。

注意:

由于 TSEND_C 采用异步处理, 所以在 **DONE** 参数值或 **ERROR** 参数值为 **TRUE** 前, 必须保持发送方区域中的数据一致。对于 TSEND_C, **DONE** 参数状态为 **TRUE** 表示数据成功发送。但并不表示连接伙伴 CPU 实际读取了接收缓冲区。

使用 TSEND_C 指令可以传送的最小数据单位是字节。LEN 参数的默认设置 (LEN = 0) 使用 DATA 参数来确定要传送的数据的长度。

6.7 TRCV_C

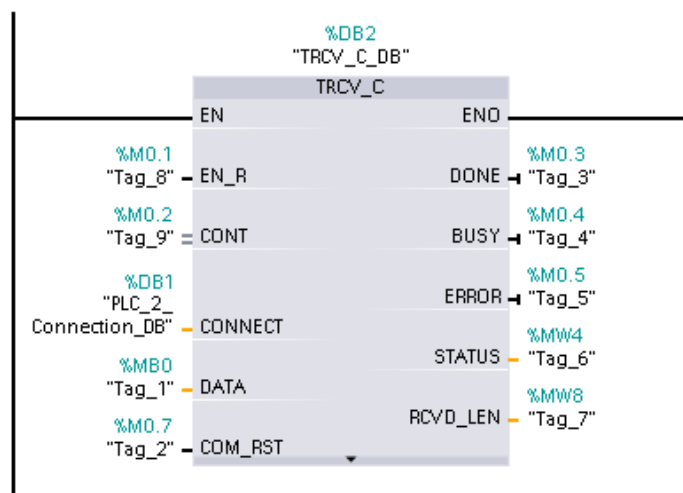


图 7: TRCV_C 函数块

TRCV_C 是异步指令且具有以下功能:

- 设置并建立通信连接:
TRCV_C 可设置并建立 TCP 或 ISO-on-TCP 通信连接。设置并建立连接后, CPU 会自动保持和监视该连接。使用参数 **CONNECT** 指定的连接描述, 可以设置通信连接。要建立连接, 参数 **CONT** 的值必须设置为“1”。连接成功建立后, 参数 **DONE** 在一个周期内设置为“1”。若 CPU 转到 **STOP** 模式, 将终止现有连接并删除所设置的相应连接。必须再次执行 **TRCV_C**, 才能重新设置并建立该连接。
 - 通过现有通信连接接收数据:
如果参数 **EN_R** 的值设置为“1”, 则启用数据接收。接收到的数据将输入到接收区中。根据所用的协议选项, 通过参数 **LEN** 指定接收区长度 (如果 **LEN** \neq 0), 或者通过参数 **DATA** 的长度信息来指定 (如果 **LEN** = 0)。成功接收数据后, 参数 **DONE** 的信号状态为“1”。如果数据传送过程中出错, 参数 **DONE** 将设置为“0”。
 - 终止通信连接:
参数 **CONT** 设置为“0”时, 将终止通信连接。
 - 通信复位:
参数 **COM_RST** 设置为“1”时, 将再次执行 **TRCV_C**。这会终止现有通信连接并建立新连接。如果再次执行该指令时正在传送数据, 可能会导致数据丢失。
- 参数描述及状态代码描述参见系统手册。

注意:

使用 **TRCV_C** 指令可以接收的最小数据单位是字节。TRCV_C 指令不支持传送布尔数据或布尔数组。由于 TRCV_C 采用异步处理, 因此仅当参数 **DONE** = 1 时, 接收器区域中的数据才一致。

处理 **TSend_C** 和 **TRCV_C** 指令花费的时间量无法确定。要确保这些指令在每次扫描循环中都被处理, 务必从主程序循环扫描中对其调用, 例如, 从程序循环 **OB** 中或从程序循环扫描中调用的代码块中对其调用。不要从硬件中断 **OB**、延时中断 **OB**、循环中断 **OB**、错误中断 **OB** 或启动 **OB** 调用这些指令。

6.8 TCON

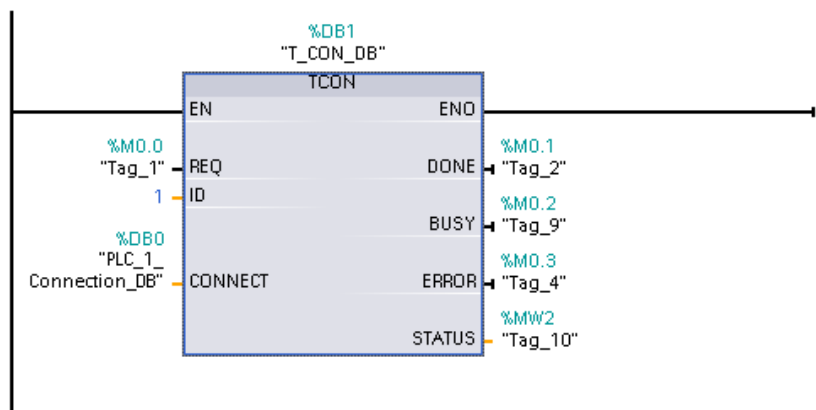


图 8: TCON 函数块

使用 TCON 可设置并建立通信连接。设置并建立连接后，CPU 会自动保持和监视该连接。TCON 是异步指令。将使用为参数 CONNECT 和 ID 指定的连接数据来设置通信连接。要建立该连接，必须在参数 REQ 中检测到上升沿。如果成功建立连接，参数 DONE 将设置为“1”。使用 TCP 和 ISO-on-TCP 的连接。

两个通信伙伴都调用“TCON”指令来设置并建立通信连接。用户需要在参数中指定哪个伙伴是主动通信节点以及哪个是被动通信节点。如果连接由于断线或远程通信伙伴而中止，主动伙伴会尝试重新建立组态的连接。用户不必再次调用 TCON。

执行“TDISCON”指令时或 CPU 切换到 STOP 模式后，会终止现有连接并删除所设置的相应连接。要再次设置并建立连接，需重新执行 TCON。
参数描述及状态代码描述参见系统手册。

6.9 TDISCON

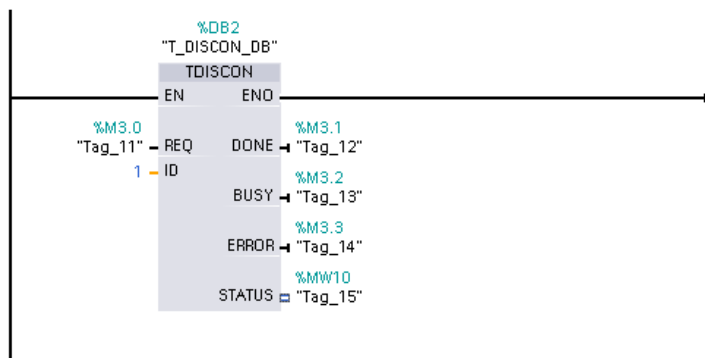


图 9: TDISCON 函数块

使用 TDISCON 可终止通信连接。在参数 REQ 中检测到上升沿时，即会启动终止通信连接的作业。在参数 ID 中，输入要终止的连接的引用。TDISCON 是异步指令。执行 TDISCON 后，为 TCON 指定的 ID 不再有效，因此不能再用于发送或接收。
参数描述及状态代码描述参见系统手册。

6.10 TSEND

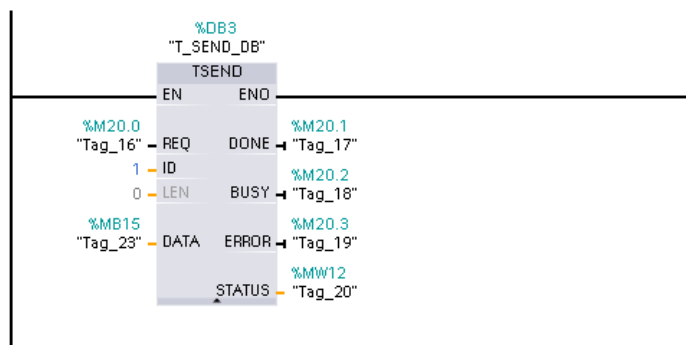


图 10: TSEND 函数块

使用 **TSEND** 可通过已有的通信连接发送数据。TSEND 是异步指令。使用参数 **DATA** 指定发送区。这包括要发送数据的地址和长度。在参数 **REQ** 中检测到上升沿时执行发送作业。使用参数 **LEN** 指定通过一个发送作业可发送的最大字节数。在发送作业完成前不允许编辑要发送的数据。如果发送作业成功执行，则参数 **DONE** 将设置为“1”。参数 **DONE** 的信号状态为“1”并不表示是确认通信伙伴已读出了发送数据。参数描述及状态代码描述参见系统手册。

注意:

由于 **TSEND** 是异步指令，所以需要在参数 **DONE** 或参数 **ERROR** 的值变为“1”前，保持发送区中的数据一致。

6.11 TRCV

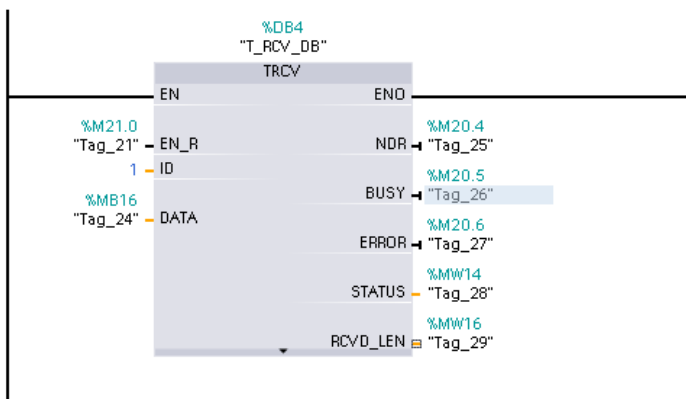


图 11: TRCV 函数块

使用 **TRCV** 可通过已有的通信连接接收数据。TRCV 是异步指令。

当参数 **EN_R** 的值设置为“1”时，启用数据接收。接收到的数据将输入到接收区中。根据所用的协议选项，通过参数 **LEN** 指定接收区长度（如果 **LEN <> 0**），或者通过参数 **DATA** 的长度信息来指定（如果 **LEN = 0**）。成功接收数据后，参数 **NDR** 的值设置为“1”。可在参数 **RCVD_LEN** 中查询实际接收的数据量。

注意: 由于 **TRCV** 是异步指令，因此仅当参数 **NDR** 的值设置为“1”时，接收区中的数据才一致。

7. S7-1200 以太网通信连接数

- 3 个用于 HMI
- 1 个用于编程设备
- 8 个用于用户程序中的以太网通信指令
- 3 个用于 S7-1200 与 S7-200/300/400 的 S7 通信

与本文档相关的知识文档连接：

SIMATIC S7 S7-1200 可编程控制器：

<http://support.automation.siemens.com/CN/view/zh/36932465>

如果您对该文档有任何建议，请将您的宝贵建议提交至[下载中心留言板](#)。

该文档文档编号：**A0414**

附录一推荐网址

自动化系统

西门子（中国）有限公司

工业自动化与驱动技术集团 客户服务与支持中心

网站首页: www.4008104288.com.cn

自动化系统 下载中心:

<http://www.ad.siemens.com.cn/download/DocList.aspx?Typeld=0&CatFirst=1>

自动化系统 全球技术资源:

<http://support.automation.siemens.com/CN/view/zh/10805045/130000>

“找答案”自动化系统版区:

<http://www.ad.siemens.com.cn/service/answer/category.asp?cid=1027>

注意事项

应用示例与所示电路、设备及任何可能结果没有必然联系，并不完全相关。应用示例不表示客户的具体解决方案。它们仅对典型应用提供支持。用户负责确保所述产品的正确使用。这些应用示例不能免除用户在确保安全、专业使用、安装、操作和维护设备方面的责任。当使用这些应用示例时，应意识到西门子不对在所述责任条款范围之外的任何损坏/索赔承担责任。我们保留随时修改这些应用示例的权利，恕不另行通知。如果这些应用示例与其它西门子出版物(例如，目录)给出的建议不同，则以其它文档的内容为准。

声明

我们已核对过本手册的内容与所描述的硬件和软件相符。由于差错难以完全避免，我们不能保证完全一致。我们会经常对手册中的数据进行检查，并在后续的版本中进行必要的更正。欢迎您提出宝贵意见。

版权© 西门子（中国）有限公司 2001-2008 版权保留

复制、传播或者使用该文件或文件内容必须经过权利人书面明确同意。侵权者将承担权利人的全部损失。权利人保留一切权利，包括复制、发行，以及改编、汇编的权利。

西门子（中国）有限公司