General Ethernet 驱动程序

1 什么是 General Ethernet? 3
2 系统配置 5
3 选择外接控制器 6
4 通讯设置示例 7
5 设置项目 9
6 支持的寄存器 13
7 寄存器和地址代码 14
8 错误消息 15
9 直接通讯功能 18
10 示例程序 26

简介

本手册介绍如何连接人机界面和外接控制器 (目标 PLC)。

在本手册中，将按以下章节顺序介绍连接步骤：

1	系统配置 本节介绍可连接的外接控制器和串口的类型。	☞ “2 系统配置 ” (第 5 页)
2	选择外接控制器 选择要连接的外接控制器的机型(系列)以及连接方式。	☞ “3 选择外接控制器 ” (第 6 页)
3	通讯设置示例 本节给出连接人机界面和外接控制器的设置示例。	☞ “4 通讯设置示例 ” (第 7 页)
4	设置项目 本节介绍人机界面上的通讯设置项目。 请使用GP-Pro EX或在离线模式下进行人机界面的通讯设置。	☞ “5 设置项目 ” (第 9 页)

操作

1 什么是 General Ethernet?

1.1 概述

General Ethernet 驱动程序并不是一个只针对人机界面连接的特定以太网通讯设备的特殊驱动程序，而是一个对所有以太网通讯设备都适用的驱动程序。
要发送的通讯包内容通过 D 脚本和人机界面的梯形图程序 (以下简称 “ 脚本等 ”) 在人机界面的内存表上生成数据。驱动程序将内存表上生成的数据从人机界面的以太网接口发送出去，再将人机界面以太网接口接收到的数据保存在内存表中。
通讯包由上述脚本等生成，使用该驱动程序，只需要编辑发送和接收函数来使用以太网接口，即可以实现人机界面与所有以太网通讯设备之间的数据发送和接收。
该通讯协议支持 UDP/IP 和 TCP/IP。

无论是采用 UDP/IP 还是 TCP/IP，可连接的以太网通讯设备的最大数量都是 8 台。
但是，目标 IP 地址和端口号是用脚本等定义的。当进行更改时，与当前通讯设备的连接会关闭，且需要打开与之后的通讯设备的连接。当与另一台设备通讯时，可以采用此方法关闭连接，连接的数量没有限制。

重 要

- General Ethernet 驱动程序采用 Memory Link 通讯方式。可以为每台人机界面设置一个 Memory Link 驱动程序。不能同时使用两个或多个 Memory Link 驱动程序。
  有关 Memory Link 方式的更多信息，请参阅 GP-Pro EX 控制器 /PLC 连接手册。在制造商列表部分，请参阅 “ 与外接控制器通讯 ”。

1.2 术语表

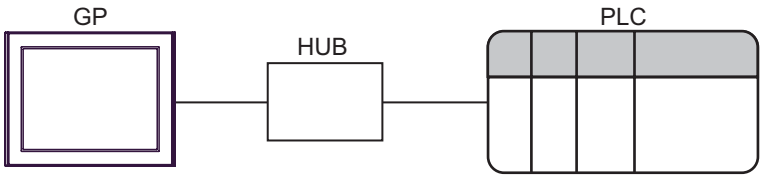
术语	定义和描述
内存表	是人机界面内部的内存表。 该驱动程序读取内存表中的数据或向内存表中写入数据。 系统区占据从 0000 到 9999 的 16 位地址。
脚本等	本手册中把在内存表上读取和写入数据的函数称为 “ 脚本等 ”。它是 “D 脚本 ” 和 “ 梯形图程序 ” 的合称。
环形缓冲区	环形缓冲区是一个以先入先出的形式临时保存数据的存储区。因缓冲区通过指针链接成环形，因此被称为环形缓冲区。在该驱动程序中，环形缓冲区建立在人机界面内存表的基础上，并被作为接收缓冲区。要管理接收缓冲区中的数据位置，需要使用两个指针：GET 指针和 PUT 指针。
GET 指针	GET 指针指出接下来要在环形缓冲区中读取的脚本等数据的位置。本驱动程序保存相对于环形缓冲区数据区首地址的偏移值。在读取了 GET 指针所指区域的数据后，GET 指针将继续指向读取位置。
PUT 指针	PUT 指针指出本驱动程序向环形缓冲区写入后续接收数据的位置。本驱动程序保存相对于环形缓冲区数据区首地址的偏移值。在将接收数据写入 PUT 指针所指的区域后，本驱动程序将指针移动到写入位置。
接收函数控制区	这是人机界面内存表中的一个固定区域，其中包含本驱动程序接收缓冲区的管理信息。
发送函数控制区	这是人机界面内存表中的一个固定区域，其中包含本驱动程序发送缓冲区的管理信息。
系统区	它是反映人机界面内存表中有关人机界面系统信息的一个区域。可读取和写入人机界面数量和各种系统信息，如日历信息等。它占据内存表中的固定地址 (0000 到 0019)。
特殊继电器	它占据内存表中的固定地址 (2032 到 2047)，和系统区一样，也反映人机界面系统信息。
9000 区	保存人机界面上的内部处理信息等，如历史趋势图的数据和通讯扫描时间等。也存在一部分可设置的区域。
用户区	它是除 “ 系统区 ”、“ 特殊继电器 ” 和 “9000 区 ” 以外的反映人机界面内存表中人机界面系统信息的区域。它是用户可自由使用的存储区。

2 系统配置

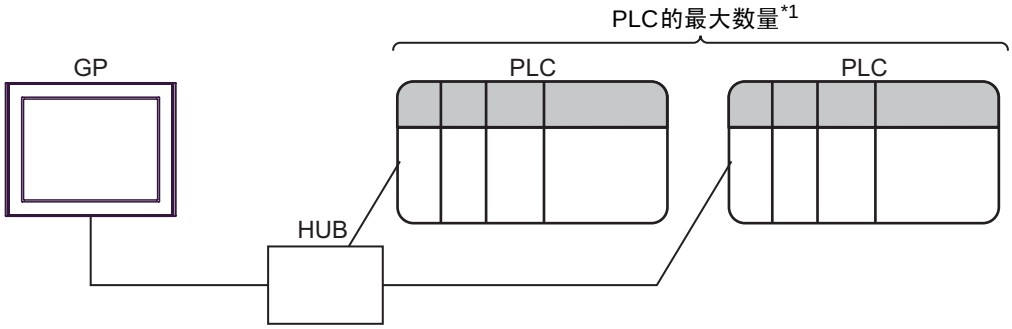
General Ethernet 驱动程序对命令终端等的通讯性能要求不高，并且使设备的通讯更容易。

■ 连接配置

- 1:1 连接

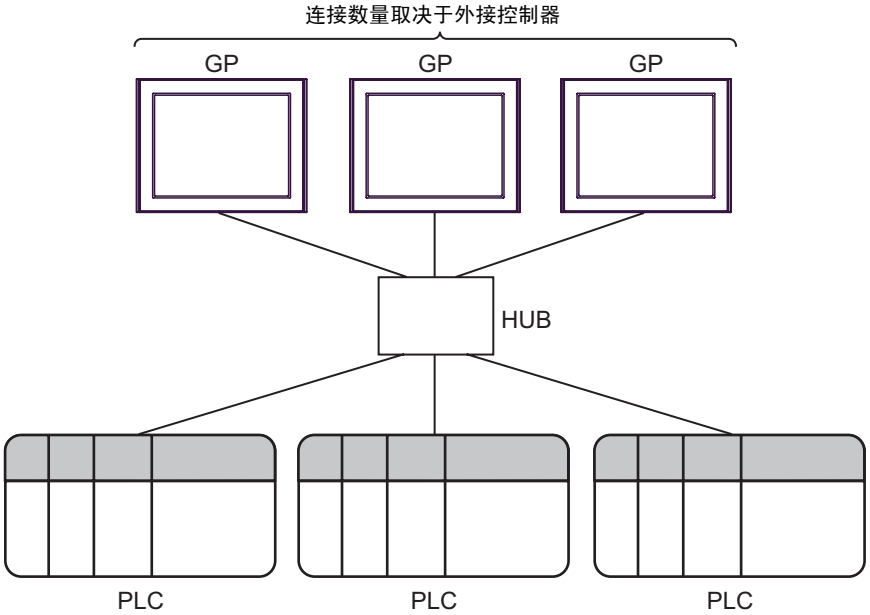


- 1:n 连接



*1 无论是采用 UDP/IP 还是 TCP/IP，可连接的以太网通讯设备的最大数量都是 8 台。但是，由于目标设备的 IP 地址和端口号是用脚本定义的。如果需要更换某个目标设备，则该设备的连接将首先被关闭，然后打开与新设备的连接。用此方法，可连接的设备数量没有限制。

- n:m 连接



3 选择外接控制器

选择要连接到人机界面的外接控制器。

设置项目	设置描述
控制器 /PLC 数量	输入 1 到 4 之间的整数表示连接到人机界面的外接控制器的数量。
制造商	选择要连接的外接控制器的制造商。请选择 “Digital Electronics Corporation”。
系列	选择要连接的外接控制器的机型 (系列) 以及连接方式。请选择 “General Ethernet”。 在系统配置的 “General Ethernet” 中检查可连接的外接控制器。 ☞ “2 系统配置 ” (第 5 页)
端口	选择要连接到外接控制器的人机界面接口。
使用系统区	此驱动程序无此项。

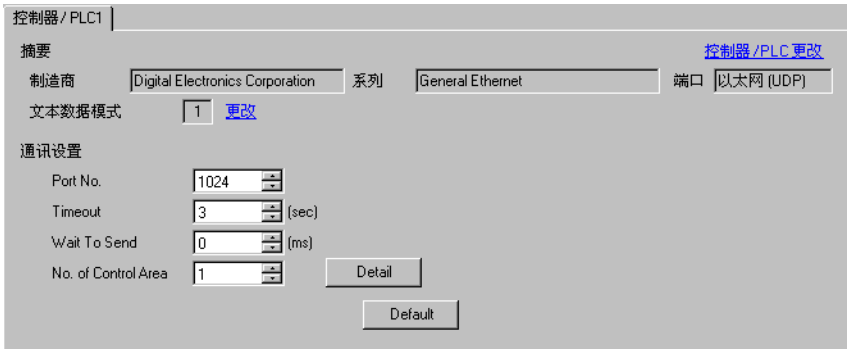
4 通讯设置示例

Pro-face 推荐的人机界面与外接控制器的通讯设置示例如下所示。

■ GP-Pro EX 设置

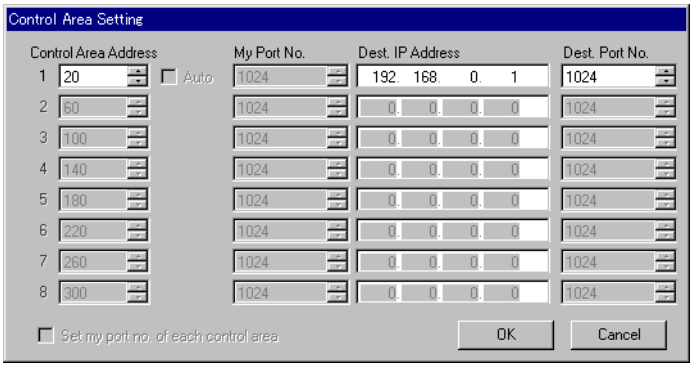
◆ 通讯设置

从 [工程] 菜单中指向 [系统设置]，点击 [控制器 /PLC]，显示设置画面。



◆ 控制区设置

要显示设置画面，请点击 “ 通讯设置 ” 下 “No. of Control Area” 旁边的 [Detail]。



注 释

- 有关 “No. of Control Area” 和 “Control Area Address”，请参阅 “9 直接通讯功能 ”。
  ☞ “9 直接通讯功能 ” (第 18 页)

■ 外接控制器设置

具体的通讯设置因使用的外接控制器而不同。

更多详情，请参阅外接控制器的手册。

◆ 步骤

(1) 请如下所示进行通讯设置。

设置项目	设置描述
IP Address	外接控制器的 IP 地址
Port No.	外接控制器的端口号

5 设置项目

请使用 GP-Pro EX 或在人机界面的离线模式下进行人机界面的通讯设置。

各参数的设置必须与外接控制器的匹配。

☞“4 通讯设置示例 ” (第 7 页)

注 释

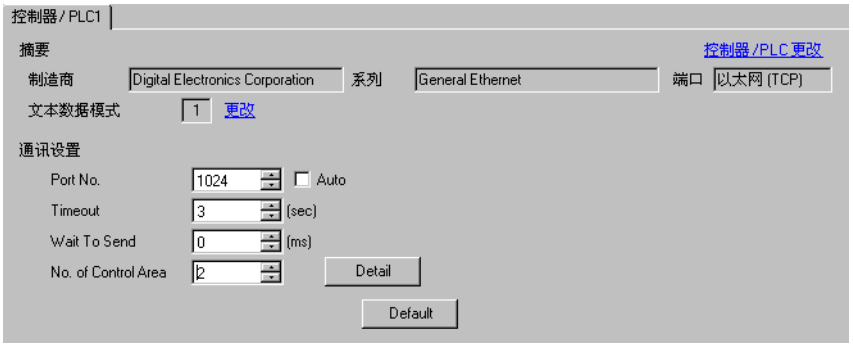
- 在人机界面的离线模式下设置其 IP 地址。

  ☞ 维护 / 故障排除手册 “ 以太网设置 ”

5.1 GP-Pro EX 中的设置项目

■ 通讯设置

从 [工程] 菜单中指向 [系统设置]，点击 [控制器 /PLC]，显示设置画面。

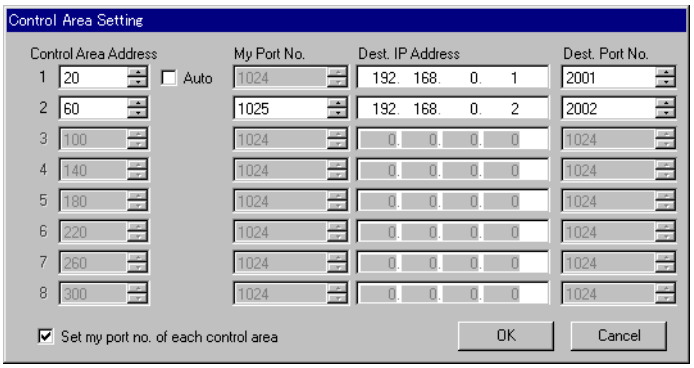


设置项目	设置描述
Port No.	输入 1024 到 65535 之间的整数表示人机界面的端口号。如果勾选 [Auto] 选项，将会自动设置端口号。 注 释 • 仅当在 [连接方式] 中选择了 “ 以太网 (TCP)” 时，才可以设置 [Auto]。
Timeout	用 1 到 127 之间的整数表示人机界面等待外接控制器响应的时间 (s)。
Wait To Send	用 0 到 255 之间的整数表示人机界面从接收包到发送下一命令之间的等待时间 (ms)。
No. of Control Area	输入从 1 到 8 的整数表示控制区数量。

■ 控制区设置

要显示设置画面，请点击 “ 通讯设置 ” 下 “No. of Control Area” 旁边的 [Detail]。

将根据在通讯设置的 [No. of Control Area] 中输入的数量显示设置项目。



设置项目	设置描述
Control Area Address	输入 20 到 1992 或 2096 到 8152 表示控制区的地址。 注 释 • 请勿设置重复的地址。
Auto	如果勾选了 Auto 并输入了第一个控制区地址，则会自动设置第二个及以后的控制区地址。 当控制区为两个或以上时会显示 “Auto”。
My Port No.	如果勾选了 [Set my port no. of each control area]，请输入一个 1024 到 65535 之间的整数表示本机端口号。如果在 TCP 连接的通讯设置中勾选了 [Auto]， [Port No.]、 [My Port No.] 会变成 [Auto]。 注 释 • 在通讯设置中设置的 [Port No.] 将被设置给控制区地址 1 的 [My Port No.]。
Dest. IP Address	设置外接控制器的 IP 地址。 注 释 • 和网络管理员确认 IP 地址。请勿设置重复的 IP 地址。
Dest. Port No.	输入 1 到 65535 之间的整数表示外接控制器的端口号。
Set my port no. of each control area	勾选此项，可设置每个控制区的 [My Port No.]。在 TCP 连接中为每个控制区设置 [My Port No.] 时，请取消勾选通讯设置中 [Auto]。

重 要

• 如果将 “ 内存表指定 ” 设置为发送控制区 “ 属性 ”，则此处设置的 IP 地址和端口号均不使用。

☞ “ ■ 发送函数控制区详情 ” (第 24 页)

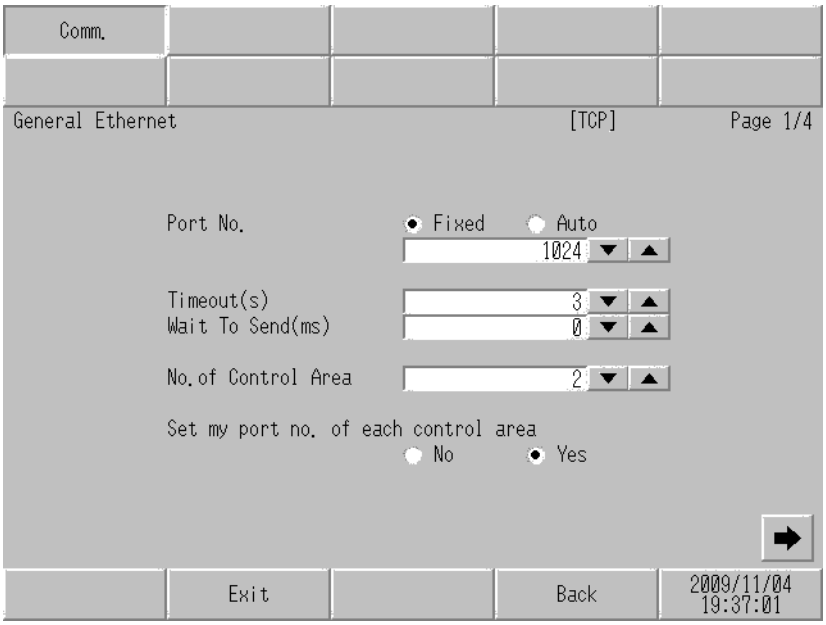
5.2 离线模式下的设置项目

注 释
- 有关如何进入离线模式以及操作方面的更多信息，请参阅 “ 维护 / 故障排除手册 ”。
  ☞ 维护 / 故障排除手册 “ 离线模式 ”

■ 通讯设置

如需显示设置画面，请在离线模式下触摸 [Peripheral Settings] 中的 [Device/PLC Settings]。在显示的列表中触摸您想设置的外接控制器。

(第 1 页)

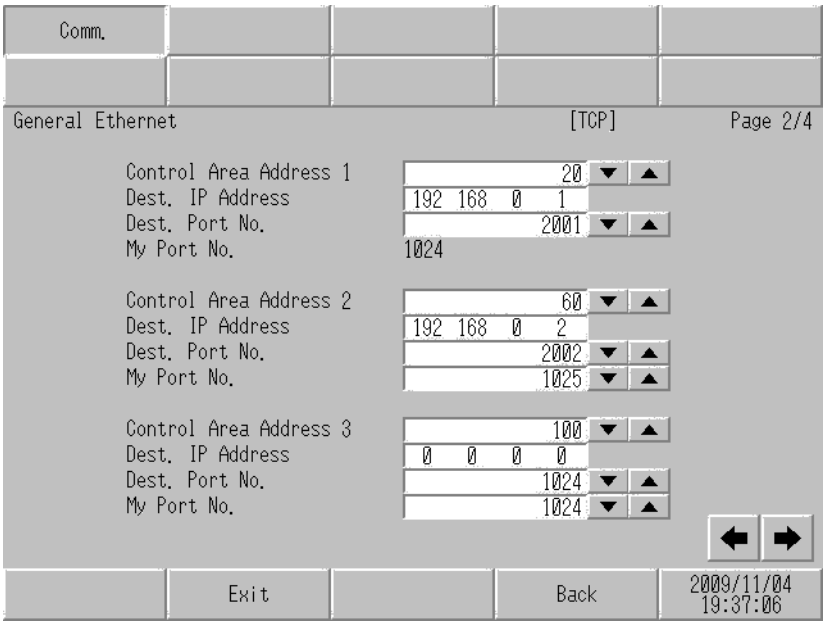


设置项目	设置描述
Port No.	设置人机界面的端口号。 如果是 UDP 连接，无论选择的是 [Fixed] 还是 [Auto]，都会分配输入的端口号。 如果是 TCP 连接，请选择 [Fixed] 或 [Auto]。如果选择 [Fixed]，请输入 1024 到 65535 之间的整数表示人机界面的端口号。如果选择 [Auto]，则无论输入何值，都将自动分配端口号。
Timeout(s)	用 1 到 127 之间的整数表示人机界面等待外接控制器响应的时间 (s)。
Wait To Send(ms)	用 0 到 255 之间的整数表示人机界面从接收包到发送下一命令之间的等待时间 (ms)。
No. of Control Area	输入从 1 到 8 的整数表示控制区数量。
Set my port no. of each control area	勾选此项，可设置每个控制区的 [My Port No.]。采用 TCP 连接时，请将 [Port No.] 选为 [Fixed]，以便为每个控制区设置 [My Port No.]。如果选择了 [Auto]，则无论是否勾选了 [Set my port no. of each control area]，都会自动为每个控制区设置相同的 [My Port No.]。

(第 2 页)

只对 “No. of Control Area” 所定义数量的控制区设置首地址、 IP 地址和端口号。

由于最初已分配了控制区数量，因此不使用大于该数量以外的控制区的设置内容。



设置项目	设置描述
Control Area Address	输入 20 到 1992 或 2096 到 8152 表示控制区的地址。 注 释 • 请勿设置重复的地址。
Dest. IP Address	设置外接控制器的 IP 地址。 注 释 • 和网络管理员确认 IP 地址。请勿设置重复的 IP 地址。
Dest. Port No.	输入 1 到 65535 之间的整数表示外接控制器的端口号。
My Port No.	如果勾选了 [Set my port no. of each control area]，请输入一个 1024 到 65535 之间的整数表示本机端口号。 注 释 • 在通讯设置中设置的 [Port No.] 将被设置给控制区地址 1 的 [My Port No.]。

重 要

• 如果将 “ 内存表指定 ” 设置为发送控制区 “ 属性 ”，则此处设置的 IP 地址和端口号均不使用。

☞ “ ■ 发送函数控制区详情 ” (第 24 页)

6 支持的寄存器

支持的寄存器地址范围如下表所示。请注意，实际支持的寄存器范围取决于所使用的外接控制器。请在您所使用的外接控制器的手册中确认实际范围。

寄存器	位地址	字地址	32 位	注释
内部寄存器	000000 - 999915	0000 - 9999	H/L	

重 要

- 本驱动程序仅对应内部寄存器。
- 系统区变为 Memory Link 方式。
- 控制区的可用范围是 20 - 2031 和 2096 - 8191。

注 释

- 请参阅手册前言部分的符号说明表。

☞“ 手册符号和术语 ”

7 寄存器和地址代码

在数据显示器中选择“控制器类型和地址”时，请使用寄存器代码和地址代码。

寄存器	寄存器名称	寄存器代码 (HEX)	地址代码
内部寄存器	-	0000	字地址

8 错误消息

错误消息在人机界面上显示如下：“ 代码：控制器名称：错误消息 (错误发生位置)”。各描述如下所示。

项目	描述
代码	错误代码
控制器名称	发生错误的外接控制器的名称。控制器名称是用 GP-Pro EX 设置的外接控制器的名称。 (初始设置为 [PLC1])
错误消息	显示与错误相关的消息。
错误发生位置	显示发生错误的外接控制器的 IP 地址或寄存器地址，或者从外接控制器接收到的错误代码。 注 释 • IP 地址显示为：“IP 地址 (十进制)：MAC 地址 (十六进制)”。 • 寄存器地址显示为：“ 地址：寄存器地址 ”。 • 收到的错误代码显示为：“ 十进制数 [十六进制数]”。

错误消息显示示例

“RHAA035：PLC1：Error has been responded for device write command (Error Code: 2[02H])”

注 释
- 有关错误代码的更多详情，请参阅您的外接控制器手册。
- 有关驱动程序错误消息的更多详情，请参阅 “ 维护 / 故障排除手册 ” 中的 “ 与人机界面相关的错误 ”。

■ 驱动程序的特定错误消息

与本驱动程序相关的特定错误代码如下所示。

消息代码	错误消息	描述
RHxx128	(节点名称):UDP/IP:It is not completed to open the socket	如果在套接字打开之前传输数据，则将显示此错误。
RHxx129	(节点名称):TCP/IP:It is not completed to open the connection	如果在连接打开之前传输数据，则将显示此错误。
RHxx130	(节点名称):TCP/IP: Cannot open passive connection with PortAuto setting	用 TCP 自动设置交换数据标题时打开被动连接。
RHxx131	(节点名称): An error occurred in protocol stack(Code:%02XH)	协议栈发生错误时显示此消息。
RHxx014	(驱动程序名称):Memory link type drivers cannot be used at the same time	同时使用 General Ethernet 驱动程序和 Memory Link 驱动程序时显示此消息。
RHxx132	(节点名称):Cannot set due to setting My Port No. range limitation (Control Area No.%d)	打开连接时分配的本机端口号如果不在 1024 到 65536 之间，则会显示此消息。
RHxx133	(节点名称):TCP/IP: It was requested transmission before connection establishment	传输数据时，未从处于被动打开状态的外接控制器执行打开过程。
RHxx134	(节点名称): It was required to open twice with different attributes	尝试重新打开时，此驱动程序的打开设置已发生改变。 如需更改打开设置，请将它关闭后再重新打开。

■ 协议栈错误代码

协议栈错误代码如下所示。保存在各控制区中的内部设置的错误代码是显示的错误代码 0x 和 1000 进行 OR 运算得到的结果。

显示的错误代码	内部设置错误代码	描述
< 无显示 >	0000	正常 (正常时不显示任何代码)
00	1000	初始化中的 IP 地址设置错误
05	1005	初始化失败
06	1006	通讯终止过程失败。
07	1007	试图在初始化未正常完成的情况下建立连接
08	1008	源端口号错误
09	1009	目标端口号错误
0A	100A	目标 IP 地址错误
0B	100B	在 UDP/IP 中已使用了相同的端口号
0C	100C	已由作为同一目标的同一号码在 TCP/IP 中建立了连接
0D	100D	协议栈拒绝打开处理
0E	100E	协议栈打开处理失败
0F	100F	连接中断
10	1010	所有连接均被使用，没有空闲连接
13	1013	被目标取消
30	1030	协议栈无应答
32	1032	目标无应答
F0	10F0	未知错误

■ 非协议栈错误代码

下表为非协议栈错误代码，是准备保存在各控制区中的内部设置错误代码。

☞“ ■ 驱动程序的特定错误消息 ” (第 15 页)

消息代码	内部设置错误代码
RHxx128	1080
RHxx129	1081
RHxx130	1082
RHxx132	1084
RHxx133	1085
RHxx134	1086

9 直接通讯功能

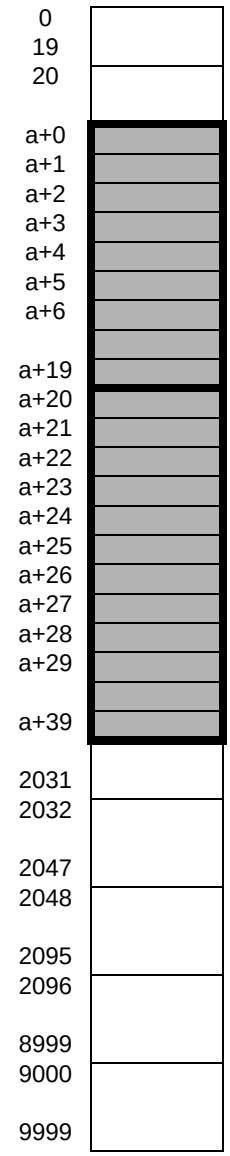
该驱动程序通过采用人机界面的脚本等生成人机界面的内存表，从而支持以太网通讯功能。这叫做 “ 直接通讯功能 ”。

9.1 内存表

人机界面的内存表示意图如下。

：这是本驱动程序使用的区域。若要同时支持多条连接，则需要多个这样的区域 (最多 8 个)。

a: 控制区首地址



: 系统区
☞ GP-Pro EX 参考手册 “ 附录 A.1.5 系统区 (Memory Link 方式)”

: 用户区

接收函数控制区
☞ “9.2 接收函数控制区 ” (第 19 页)

发送函数控制区
☞ “9.3 发送函数控制区 ” (第 22 页)

: 用户区

: 特殊继电器
☞ GP-Pro EX 参考手册 “ 附录 A.1.5 系统区 (Memory Link 方式)”

: 保留区

: 用户区

: 9000 区
☞ GP-Pro EX 参考手册 “ 附录 A.1.5 系统区 (Memory Link 方式)”

9.2 接收函数控制区

该驱动程序从以太网通讯设备接收数据，并将数据保存到接收缓冲区。
可以在人机界面的内存表中的任意地址处以任意大小建立这一接收缓冲区。
这些缓冲区中用于处理接收数据的区域被称为 “ 接收函数控制区 ”。该接收函数控制区占据人机界面内存表中的固定地址 (a+9 到 a+19)，如下图所示。

a: 控制区首地址

☞“ ■ 接收函数控制区详情 ” (第 21 页)

a+0		: 接收函数控制 (0：函数停止， 1：执行处理)
a+1		: 接收结果 (0：正常， 1：错误)
a+2		: 接收环形缓冲区起始内存表地址
a+3		: 接收环形缓冲区字数
a+4		: GET 指针 (相对于缓冲区头的偏移值)
a+5		: PUT 指针 (相对于缓冲区头的偏移值)
a+6		
a+19		

保存接收数据的接收缓冲区是用该接收函数控制区来定义的。通常，人面界面启动后即立刻对此进行定义。
GET 指针 (a+4) 指示的是用于保存脚本所读取的接收数据的内存表首地址。PUT 指针 (a+5) 指示的是人机界面向接收缓冲区写入接收数据所用到的内存表首地址。如果定义了相对于上述接收缓冲区首地址的偏移值，并且它超过了定义的接收缓冲区大小， GET 指针和 PUT 指针会被再次置 0。
接收函数控制字 (a+0) 用来控制将数据从人机界面的接收缓冲区转移到接收环形缓冲区。如果接收函数控制字是 a+0，则接收处理停止，并且不会将数据从人机界面的接收缓冲区转移到接收环形缓冲区。如果接收函数控制字是 a+1，则会将人机界面收到的数据内容转移到接收环形缓冲区。

■ 接收缓冲区

PUT 指针指示后续接收数据的写入位置，指针由人机界面侧控制。

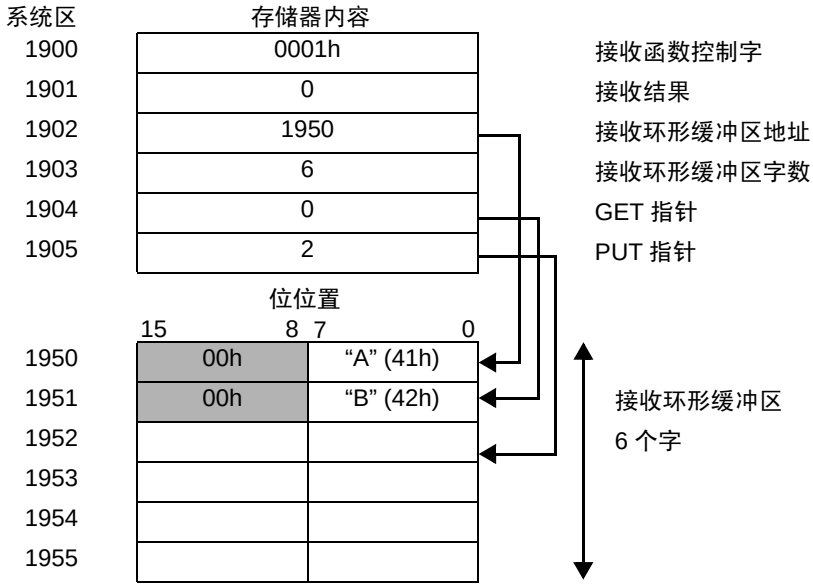
接收缓冲器格式如下，其中显示已接收数据的状态。环形缓冲区从内存表地址的 a+50 开始，占据 6 个字。下图显示出已接收了两个字节 (“A”、 “B”) 的状态。

每次收到一个字节时，人机界面的接收函数将收到的数据保存到 PUT 指针所指的位置。脚本等从 GET 指针所指的位置读取数据，并将指针推移到下一个读取位置。接收到的数据将被逐字节地保存在内存表中各地址 (字：16 位) 的 后续字节中。

PUT 指针指示后续接收数据的写入位置，指针由人机界面侧控制。

当它达到接收缓冲区的最后地址 (a+55) 时，会再次从首地址 (a+50) 开始保存。但是， GET 指针的位置不会被超越 (在读取未完成的地方不存在数据重写)。因此，接收到的数据是脚本等。当用脚本等读取数据之后，收到的数据应正确地推移 GET 指针。如果 “ 在不更新 GET 指针的情况下不能将收到的数据写入接收缓冲区 ” 的状态持续的话，人机界面的接收缓冲区可能发生溢出。

例) 假设控制区中的首地址是 1900



* 接收到的数据将被逐字节地保存在内存表中各地址 (字) 的低字节中。

■ 接收函数控制区详情

a: 控制区首地址

系统区地址	名称	更新责任方*1 (触发方)	描述
a+00	接收函数控制字	脚本等	0: 函数停止 接收函数停止。 1: 允许接收 将接收自人机界面接收缓冲区的数据保存到接收环形缓冲区。它由脚本来更新。
a+01	接收结果	人机界面	反映由打开、关闭和接收等处理产生的协议栈错误代码。 0: 正常结束 非 0: 错误 确认错误后，脚本等在此区域写入 0 并接收后续数据。 ☞ “ ■ 协议栈错误代码 ” (第 16 页)
a+02	接收环形缓冲区起始内存表地址	脚本等	设置接收环形缓冲区的起始内存表地址。 它被设置到系统区、特殊继电器区和 9000 区以外的用户区，尽管可以将它设置到任意一个内存表地址。
a+03	接收环形缓冲区字数	脚本等	设置接收缓冲区的字数。 (保存一个比接收字节数大的数值) 会指定从上述起始地址开始的数个连续字。
a+04	GET 指针	脚本等	指出要读取的下一个接收数据的地址，它保存的是相对于接收环形缓冲区起始地址的偏移值。 脚本等从该指针所指的位置获取数据，然后更新指针位置。
a+05	PUT 指针	人机界面	指出人机界面接收到接收缓冲区的数据被写入的位置，它保存相对于接收环形缓冲区起始地址的偏移值。 只要人机界面接收到数据，就会自动更新该指针。
a+06	接收源 IP 地址	脚本等	接收源 IP 地址 (高 16 位)。 用于指定被动打开及连接目标。
a+07	接收源 IP 地址	脚本等	接收源 IP 地址 (低 16 位)。 用于指定被动打开及连接目标。
a+08	接收源端口号	脚本等	接收源端口号 用于指定被动打开及连接目标。
a+09	保留		
:	保留		
a+19	保留		

*1 显示由何方来负责数据更新，以确保函数的正确运行。
人机界面：由人机界面负责。
脚本等：由使用此函数的脚本等负责。

重 要

- 注意，请勿使发送缓冲区、其他人机界面系统区等和定义区紧随接收缓冲区之后。否则会造成运行错误。
- 如果在 GP-Pro EX 中使用 [备份内部寄存器]，设置地址时请勿包含接收函数控制字。如果地址包含接收函数控制字，在恢复备份数据时人机界面可能无法正常进行数据接收。

9.3 发送函数控制区

为了向以太网通讯设备发送数据，该驱动程序含有临时存储发送数据包的发送缓冲区。可以在人机界面的内存表中的任意地址处以任意大小建立这一发送缓冲区。这些缓冲区中用于处理发送数据的区域被称为 “ 发送函数控制区 ”。该发送函数控制区占据人机界面内存表中的特定地址 (a+20 到 a+39)，如下图所示。

a: 控制区首地址

☞“ ■ 发送函数控制区详情 ” (第 24 页)

a+20		: 发送函数控制字 (0：函数停止， 1：执行处理)
a+21		: 发送结果 (0：正常， 1：错误)
a+22		: 发送缓冲区起始内存表地址
a+23		: 发送数据位数
a+24		: 属性
a+25		: 保留
a+26		: 目标 IP 地址 (高 16 位)
a+27		: 目标 IP 地址 (低 16 位)
a+28		: 目标端口号
a+29		: 保留
a+30		: 保留
a+31		: 我的端口号
a+32		: 保留
:		: 保留
a+39		: 保留

准备发送到以太网的数据首先会被保存在发送缓冲区中。之后，通过执行传输操作，缓冲区中的数据被发送到以太网。

在将发送缓冲区的首地址设置为发送缓冲区起始内存表地址 (a+22)，并将数据保存到设置的缓冲区之后，会设置传输数据的字节数 (a+23)。生成传输数据包之后，如果在发送函数控制字 (a+20) 中写入 1，则将通过以太网传输数据包。

执行完各项处理之后，发送函数控制字自动变为 0。

重 要

- 随后即使在发送函数控制字中写入 1，也不会发送数据包。
- 控制区地址超过 8192 时
- 如果 “ 接收环形缓冲区首地址字数 + 接收环形缓冲区 ” 超过地址 8192。
- 如果 “ 发送缓冲区首地址 + 传输数据字节数 ” 超过 8192。
- 如果 GET 指针或 PUT 指针超过接收环形缓冲区末端。

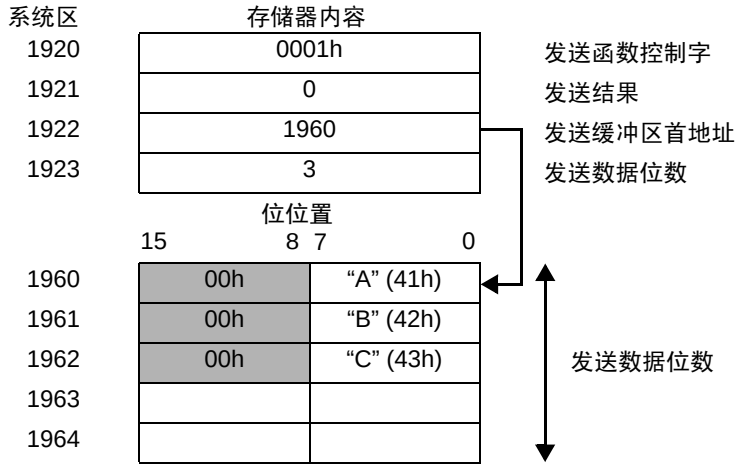
■ 发送缓冲区

发送缓冲区格式如下，其中显示发送数据的状态。

在缓冲区 + 60 所构成的内存表地址处，显示发送三个字节 (“A”， “B”， “C”) 的状态，如下图所示。

发送缓冲区从指定的首地址开始保存数据，并指定保存数据的字节数。如果将数据逐字节地保存在各个字的低字节中，则字节数即是占据内存表的字数。

例) 假设控制区中的首地址是 1900



* 请将传输数据逐字节地保存在内存表中各地址 (字) 的低字节中。

■ 发送函数控制区详情

a: 控制区首地址

系统区地址	名称	更新责任方[*1] (触发方)	描述
a+20	发送函数控制字	脚本等	0: 函数停止 (处理完成) 发送函数停止。 完成下面的处理后，人机界面将该区域的值复位为 0。 0x0001: 通过以太网发送发送缓冲区中的内容。 0x0002: 根据属性 (a+24) 设置，端口对指定的目标 IP 地址打开。 TCP: 打开 TCP 连接。 UDP: 打开 UDP 套接字。 0x0004: 关闭连接 TCP: 关闭 TCP 连接。 UDP: 关闭 UDP 套接字。 * 传输步骤 1. 设置 0x0002，打开连接 / 套接字。 2. 设置 0x0001，发送发送缓冲区内容。 3. 设置 0x0004，关闭连接 / 套接字。
a+21	发送结果	人机界面	反映由传输处理生成的协议栈错误代码。 0: 正常结束 非 0: 错误 确认错误后会在此区域写入 0，脚本继续传输后续数据。 ☞ “ ■ 协议栈错误代码 ” (第 16 页)
a+22	发送缓冲区起始内存表地址	脚本等	设置发送缓冲区的起始内存表地址。 请将它设置到用户区，而不是系统区主特殊继电器区等。尽管可以将它设置到人机界面内存表的任意位置。
a+23	发送数据位数	脚本等	设置发送缓冲区的字数。 (请在此处设置发送缓冲区中保存的字节数。)
a+24	属性	脚本等	设置通讯格式。 • 位 0：目标 IP 地址指定 0: U/I 指定 1: 内存表指定 • 位 1：目标端口号指定 0: U/I 指定 1: 内存表指定 • 位 2：连接方式指定 0: U/I 指定 1: 内存表指定 (参考位 3、位 4) • 位 3：连接方式 0: TCP/IP 1: UDP/IP • 位 4：打开方式 0: 主动打开 1: 被动打开 • 位 5：自动打开 0: 不自动打开 1: 自动打开 • 位 6：My Port No. 0: U/I 指定 1: 内存表指定 • 位 7 至 15：未定义
a+25	保留		保留

系统区地址	名称	更新责任方*1 (触发方)	描述
a+26	目标 IP 地址	脚本等	目标 IP 地址 (高 16 位)。
a+27	目标 IP 地址	脚本等	目标 IP 地址 (低 16 位)。
a+28	目标端口号	脚本等	目标端口号
a+29	保留		保留
a+30	保留		保留
a+31	My Port No.	脚本等	指定给此控制区的本机端口号 (1024 ~ 65535)。 指定 a+24 的属性位 6。
a+32	保留		保留
:	保留		保留
a+39	保留		保留

*1 显示由何方来负责数据更新，以确保函数的正确运行。
人机界面：由人机界面负责。
脚本等：由使用此函数的脚本等负责。

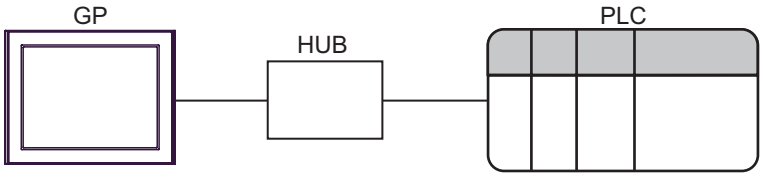
重 要

- 注意，请勿使接收缓冲区、其他人机界面系统区等和定义区紧随发送缓冲区之后。否则会造成运行错误。
- 如果勾选端口号的 “ 自动分配 ”，则不能使用 TCP 被动打开。
- 如果在 GP-Pro EX 中使用 [备份内部寄存器]，设置地址时请勿包含发送函数控制字。如果地址包含发送函数控制字，在恢复备份数据时人机界面可能无法正常进行数据发送。

10 示例程序

发送和接收的过程以及示例脚本如下所示。

< 系统配置 >



< 程序概要 >

示例程序中的通讯详情如下。

1. 向外接设备发送三个字节的数据 (ABC)。
2. 接收两个字节的传输数据。

< 发送和接收过程示例 >

下面举例说明从 AGP 到外接控制器发送命令并从外接设备接收响应的过程。

1. 接收函数控制区设置
   (1) 接收结果清除
   (2) 接收环形缓冲区首地址设置
   (3) 接收环形缓冲区字数设置
   (4) 修正 GET 指针和 PUT 指针之间的差异。 (因为它已成为无用数据。)
   (5) 接收函数控制字设置 (0x0001: 允许接收)
2. 发送函数控制区设置
   (1) 发送结果清除
   (2) 发送缓冲区首地址设置
   (3) 属性设置 (IP 地址、端口号和传输系统设置)
   (4) 发送函数控制字设置 (0x0002: TCP 连接打开或 UDP 套接字打开)*1
3. 发送数据的创建和发送
   (1) 创建发送数据
   (2) 发送数据字节数设置
   (3) 发送函数控制字设置 (0x0001: 发送命令)
4. 数据的接收处理
   (1) 从接收环形缓冲区中清除接收数据。 (移动 GET 指针)
5. 发送函数控制区设置
   (1) 发送函数控制字设置 (0x0004: TCP 连接关闭或 UDP 套接字关闭)

*1 如果在属性 (a+24) 中设置了自动打开，则不需要此步骤。

< 示例脚本 >

以下 “ ■ 接收缓冲区 ” (第 20 页)“ ■ 发送缓冲区 ” (第 23 页) 是根据上述列出的内容编写的脚本。
其中将控制区首地址指定为 1900，并列出了已使用的存储器。

地址	存储器内容	
1900	0001h	接收函数控制字
1901	0	接收结果
1902	1950	接收环形缓冲区首地址
1903	6	接收环形缓冲区字数
1904	0	GET 指针
1905	2	PUT 指针
:	:	
1920	0000h	发送函数控制字
1921	0	发送结果
1922	1960	发送缓冲区首地址
1923	3	发送数据字节数
1924	0x07	属性 (主动打开 / TCP / 其他脚本等设置)
1925		
1926	0xC0A8	目标 IP 地址 (高 16 位：192.168)
1927	0x0101	目标 IP 地址 (低 16 位：1.1)
1928	1025	目标端口号 (1025)
:	:	

位位置

	15 8	7 0	
1950	00h	“A” (41h)	接收环形缓冲区
1951	00h	“B”(42h)	6 字
1952			
1953			
1954			
1955			
:	:		

	15 8	7 0	
1960	00h		
1961	00h		发送缓冲区
1962	00h		
1963			
1964			
:	:		

2000	0	处理状态保存
2001	0	接收字节数保存
2002		
2003		
2004		

1. 打开处理 (接收函数控制区设置)

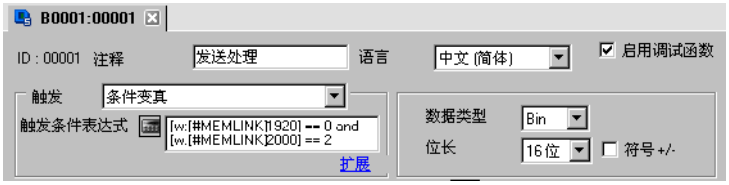
- 触发条件

- 执行语句

```
// 初始化控制区，连接打开处理

// 接收函数控制区设置 -----
[w:[#MEMLINK]1901] = 0          // 接收结果清 0
[w:[#MEMLINK]1902] = 1950       // 接收缓冲区首地址
[w:[#MEMLINK]1903] = 6          // 接收缓冲区字数
[w:[#MEMLINK]1904] = 0          // GET 指针清 0
[w:[#MEMLINK]1905] = 0          // PUT 指针清 0
[w:[#MEMLINK]1931] = 1024       // 我的端口号

// 允许接收
[w:[#MEMLINK]1900] = 1          // 接收控制字接收许可

// 发送函数控制区设置 -----
[w:[#MEMLINK]1921] = 0          // 发送结果清 0
[w:[#MEMLINK]1922] = 1960       // 发送缓冲区首地址
// 属性设置
[w:[#MEMLINK]1924] = 0x47       // b6:1( 本机端口号内存表指定 ) b5:0 无自动打开
                                // b4:0( 主动打开 ) b3:0(TCP 通讯 ) b2:1( 发送系统 )
                                // b1:1( 目标端口号 ) b0:1( 目标 IP 地址 )
// 目标 IP 地址和端口号设置
[w:[#MEMLINK]1926] = 0xC0A8     // 192.168
[w:[#MEMLINK]1927] = 0x0101     // 1.1
[w:[#MEMLINK]1928] = 1025       // 端口号

// 连接打开 -----
[w:[#MEMLINK]1920] = 2          // 发送控制字连接打开

// 在存储器中保存打开完成状态 -----
[w:[#MEMLINK]2000] = 2          // 打开完成
[w:[#MEMLINK]2001] = 0          // 接收数据存储器清 0
```

2. 发送处理 (发送数据的创建和发送)

- 触发条件

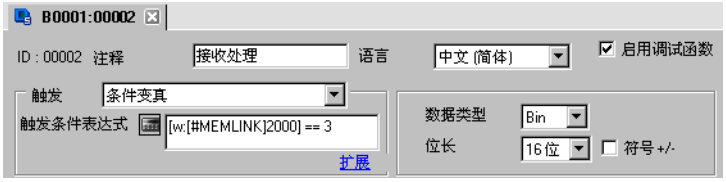


- 执行语句

```
// 包编译和发送处理

// 包编译 -----
[w:[#MEMLINK]1960] = 0x41        // 'A'
[w:[#MEMLINK]1961] = 0x42        // 'B'
[w:[#MEMLINK]1962] = 0x43        // 'C'

// 设置发送字节数
[w:[#MEMLINK]1923] = 3           // 发送 3 字节


// 包发送 -----
[w:[#MEMLINK]1920] = 1           // 发送控制字数据发送


// 在存储器中保存发送完成状态 -----
[w:[#MEMLINK]2000] = 3           // 发送完成
```

3. 接收处理 (数据的接收处理)

- 触发条件

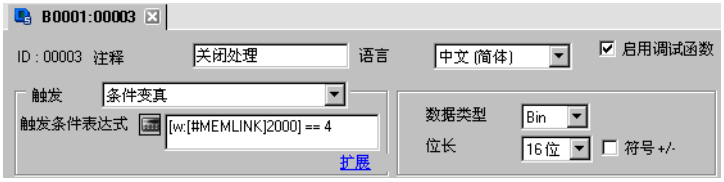


- 执行语句

```
// 接收处理

if( [w:[#MEMLINK]1904] <> [w:[#MEMLINK]1905] )
{ // 检查 GET 指针和 PUT 指针。如果不同，则有一个接收数据。
  [t:0000] = [w:[#MEMLINK]1903]    // 接收缓冲区大小循环
  [t:0001] = [w:[#MEMLINK]2001]    // 接收数据存储器清 0

 loop ([t:0000])
  {
    // 检查 GET 指针和 PUT 指针。如果相同，则获取完成。
    if( [w:[#MEMLINK]1904] == [w:[#MEMLINK]1905] ) {
        break
    }endif
    [t:0002] = [w:[#MEMLINK]1904]  // 获取 GET 指针

   // 获取数据
   [w:[#MEMLINK]3000]#[t:0001] = [w:[#MEMLINK]1950]#[t:0002]

   // 保存 / 获取数据的位置加 1
   [t:0001] = [t:0001] + 1
   [w:[#MEMLINK]1904] = [w:[#MEMLINK]1904] + 1
   if( [w:[#MEMLINK]1904] >= [w:[#MEMLINK]1903]) {
       [w:[#MEMLINK]1904] = 0
   }endif
  }
  endloop

  // 检查获取的数据 (2 字节 )
  if( [t:0001] >= 2 ) {
    // 在存储器中保存接收完成状态 -----
    [w:[#MEMLINK]2000] = 4    // 接收完成
    [w:[#MEMLINK]2001] = 0    // 接收数据存储器清 0
  } else {
    // 表示数据在接收缓冲器中，转向下一处理 -----
    [w:[#MEMLINK]2001] = [t:0001]
  }endif
}
endif
```

4. 关闭处理

- 触发条件



- 执行语句

```
// 连接关闭处理

// 关闭连接 -----
[w:[#MEMLINK]1920] = 4    // 发送控制字关闭


// 在存储器中保存关闭处理完成状态 -----
[w:[#MEMLINK]2000] = 0    // 关闭完成
```