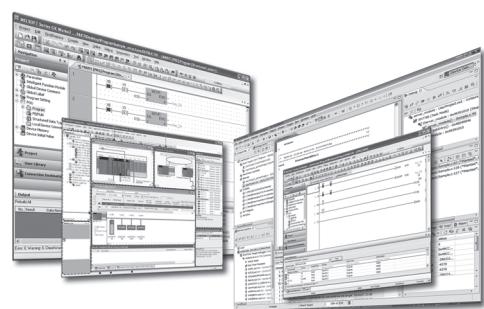


FA集成软件

MX Component Version 4 编程手册

-SW4DNC-ACT



●安全注意事项●

(使用之前请务必阅读)

使用本产品之前，应仔细阅读本手册及本手册中介绍的关联手册，同时在充分注意安全的前提下正确操作。本手册中的注意事项仅记载了与本产品有关的内容。关于可编程控制器系统方面的安全注意事项，请参阅所使用的CPU模块的用户手册。

在“安全注意事项”中，安全注意事项被分为“ 警告”和“ 注意”两个等级。



表示错误操作可能造成危险后果，导致死亡或重伤事故。



表示错误操作可能造成危险后果，导致中度伤害、轻伤及设备损失。

此外，根据情况不同，即使“ 注意”这一级别的事项也有可能引发严重后果。

两级注意事项记载的都是重要内容，请务必遵照执行。

请妥善保管本手册以备需要时阅读，并将本手册交给最终用户。

[设计注意事项]



- 应在可编程控制器系统外部设置一个互锁电路，确保通过计算机对运行中的可编程控制器进行数据更改、状态控制时能保证整个系统的安全运行。
此外，应预先确定通过外围设备对可编程控制器CPU进行在线操作的过程中由于电缆连接不良等导致发生通信异常时系统方面的处理方法。



- 将计算机连接到运行中的CPU模块上进行在线操作（尤其是强制输出、运行状态更改）时，应在仔细阅读手册并充分确认安全的基础上实施操作。
操作错误可能导致机械损坏或引发事故。

●关于产品的应用●

- (1) 在使用三菱可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效安全功能。
- (2) 三菱可编程控制器是以一般工业用途等为对象设计和制造的通用产品。因此，三菱可编程控制器不应用于以下设备・系统等特殊用途。如果用于以下特殊用途，对于三菱可编程控制器的质量、性能、安全等所有相关责任（包括但不限于债务未履行责任、瑕疵担保责任、质量保证责任、违法行为主张、制造物责任），三菱电机将不负责。
 - 面向各电力公司的核电站以及其它发电厂等对公众有较大影响的用途。
 - 用于各铁路公司或公用设施目的等有特殊质量保证体系要求的用途。
 - 航空航天、医疗、铁路、焚烧・燃料装置、载人移动设备、载人运输装置、娱乐设备、安全设备等预计对人身财产有较大影响的用途。

然而，对于上述应用，如果在限定于具体用途，无需特殊质量（超出一般规格的质量等）要求的条件下，经过三菱电机的判断也可以使用三菱可编程控制器，详细情况请与当地三菱电机代表机构协商。

使用注意事项

在本项中以下述顺序说明注意事项。

- 1) 所使用的操作系统、计算机的注意事项
- 2) 安装、卸载时的注意事项
- 3) 可编程控制器 CPU 相关注意事项
- 4) 使用其它 MELSOFT 产品时的注意事项
- 5) 使用以太网模块时的注意事项
- 6) 使用 CC-Link 模块时的注意事项
- 7) 使用串行通信模块时的注意事项
- 8) 调制解调器通信时的注意事项
- 9) 编程时的注意事项
- 10) 使用 Microsoft[®] Excel[®] 时的注意事项
- 11) 使用 Microsoft[®] Access[®] 时的注意事项
- 12) 使用 VBScript 时的注意事项
- 13) 使用机器人控制器时的注意事项

所使用的操作系统、计算机的注意事项

(1) 以无 Administrator 权限的用户运行 MX Component 时的限制

以无 Administrator 权限的用户运行 MX Component 时，有以下限制。

(a) 通信设置实用程序

- 不能进行逻辑站号的创建、更改及删除。
- 不能进行通信设置的导入。
- 以 MX Component Version 3.00A 之前的版本进行了通信设置的情况下，无法启动。^{*1}

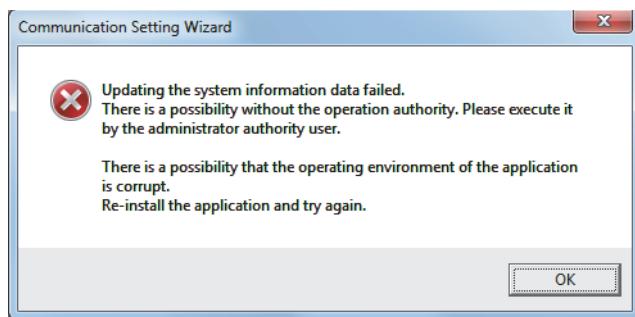
(b) 可编程控制器监视实用程序

- 以 MX Component Version 3.00A 之前的版本进行了通信设置的情况下，无法启动。^{*1}
- 不能通过软元件登录监视进行软元件登录。

(c) 通信板

- 不能通过 CC-Link IE 控制网络、CC-Link IE 现场网络、MELSECNET/H、CCLink 板的各实用程序进行各种设置。

^{*1}: 显示了以下出错信息的情况下，应以具有 Administrator 权限的用户启动 • 结束实用程序。此后即使以无 Administrator 权限的用户身份也可以启动实用程序。



(2) 关于计算机的唤醒功能等

设置了计算机的唤醒功能 • 暂停设置 • 节电功能 • 待机模式后与可编程控制器 CPU 进行通信时，有可能发生通信出错。

因此，与可编程控制器 CPU 通信的情况下，请勿进行上述功能设置。

安装、卸载时的注意事项

(1) 关于安装

进行覆盖安装的情况下，应安装到与已安装的文件夹相同的文件夹中。

(2) 关于开始菜单

已卸载了 MX Component 的情况下，项目有可能会残留在开始菜单中。

在这种情况下，应重新启动计算机。

(3) 下载安装程序以使用时

下载安装程序以使用时，应将程序保存至不含空格的目录中并执行程序。

可编程控制器 CPU 相关注意事项

(1) USB 通信时的注意事项

在与可编程控制器 CPU 的通信过程中如果频繁地进行 USB 电缆的拆装、可编程控制器 CPU 的复位及电源的 OFF/ON，有可能导致发生通信出错且无法恢复。

在这种情况下，应将 USB 电缆完全拔下且经过 5 秒以上时间后重新装上。

此外，本操作后初次通信时有可能出错，但第 2 次以后将恢复正常功能。

(2) 关于可编程控制器 CPU 的时钟数据

- (a) 在 QCPU(Q 模式)、LCPU 及 FXCPU 中，即使在可编程控制器 CPU 处于 RUN 的状态下也可执行。
- (b) 对于 QCPU(Q 模式) 及 LCPU，可以进行设置，与时钟设置用软元件“SM1028”的 ON/OFF 状态无关。
- (c) 对于 FXCPU，只有内置时钟功能的机型或安装了 RTC 盒的 FXU、FX2C、FX2NC 可以设置时钟数据。
- (d) 时钟设置时将产生相当于传送时间的误差，应加以注意。

(3) 使用 FXCPU 时的限制事项

- (a) 使用 FXCPU 时访问 TN 软元件（定时器当前值）及 CN 软元件（计数器当前值）的情况下，不能从软元件编号 199 之前对软元件编号 200 以后进行访问。
- (b) 对于 FXCPU，由于作为可编程控制器 CPU 而不具有 PAUSE 开关，因此通过 SetCpuStatus 指定远程 PAUSE 时将返回出错信息。
- (c) 即使指定不存在模块的起始 I/O 编号执行 WriteBuffer() 方式也不会返送出错信息，因此应加以注意。
- (d) 对于 FXCPU 的变址寄存器 (Z、V)，在 WriteDeviceBlock() 中不能连续写入 2 点以上。（只能写入 1 点。）

(4) Q00UJ/Q00/Q00U/Q01/Q01U/Q02UCPU^{*1} 的串行通信功能

*1：在本项中，记述支持串行通信功能的 Q00UJ/Q00/Q00U/Q01/Q01U/Q02UCPU。

满足以下所有条件时，计算机—支持串行通信功能的 CPU 之间的通信速度为 9600bps。

- 连接 CPU 的串行通信功能处于有效状态。
- 计算机侧传送速度设置与支持串行通信功能 CPU 侧传送速度设置不相同。

此外，希望提高通信速度的情况下，应使计算机侧传送速度与支持串行通信功能 CPU 侧传送速度一致。

(5) 使用以太网内置型 CPU 时的注意事项

使用 MX Component 且 TCP/IP 连接状态下 (Open 中) 如果对可编程控制器 CPU 进行复位, 此后通信时将发生通信出错或接收出错。

在这种情况下, 应在使用 MX Component 的应用程序内执行关闭处理后, 重新执行打开处理。

(6) 使用 QSCPU 时的注意事项

为了保护安全可编程控制器系统, 不可执行对缓冲存储器进行写入、对软元件进行写入・设置、对时钟数据进行写入的函数。

使用其它 MELSOFT 产品使用时的注意事项

(1) GX Simulator 通信时的注意事项

执行可编程控制器监视实用程序、通信设置实用程序及用户程序之前, 应确认 GX Simulator 及 GX Developer 处于已启动状态。此外, 在用户程序执行过程中, 请勿结束 GX Simulator 及 GX Developer。
否则用户程序将无法正常结束。

(2) MT Simulator2 通信时的注意事项

- 应在安装 MX Component 后, 安装 MT Developer2。
- 至 MT Simulator2 的可连接数 *1 最多为 4 根。

*1: 可连接数中包含了 MT Developer2。

例) 启动 MT Developer2 后, 启动了 1 台 MT Simulator2 时
通过 MX Component 最多可连接 3 根。

使用以太网模块时的注意事项

(1) TCP/IP 连接状态下的可编程控制器 CPU 复位

使用 MX Component 且 TCP/IP 连接状态下 (Open 中) 如果对可编程控制器 CPU 进行复位, 此后通信时将发生通信出错或接收出错。

在这种情况下, 应在使用 MX Component 的应用程序内执行关闭处理后, 重新执行打开处理。

(2) 关于以太网模块的目标存在确认开始间隔

有时会发生即使通过计算机执行关闭处理 (Close), 以太网模块也不执行关闭处理 (Close) 的现象。这有可能是电缆断线所致。

在以太网模块未执行关闭处理 (Close) 的状态下即使通过计算机执行打开处理 (Open), 以太网模块也将进行目标存在确认, 在以太网模块的关闭处理 (Close) 执行之前, 通过计算机的打开处理 (Open) 将不会正常结束。

希望尽快结束通过计算机的打开处理 (Open) 时, 应缩短以太网模块的目标存在确认间隔设置。

(以太网模块的目标存在确认开始间隔的默认值为 10 分钟。)

(3) 以太网模块的更换

进行以太网通信时, 由于调试及故障等更换了以太网模块的情况下, 需要进行其它节点 (计算机) 侧的再启动。
(这是由于以太网地址 (MAC 地址) 根据各设备而有所不同。)

(4) 关于使用 Q 系列以太网模块时的同时访问

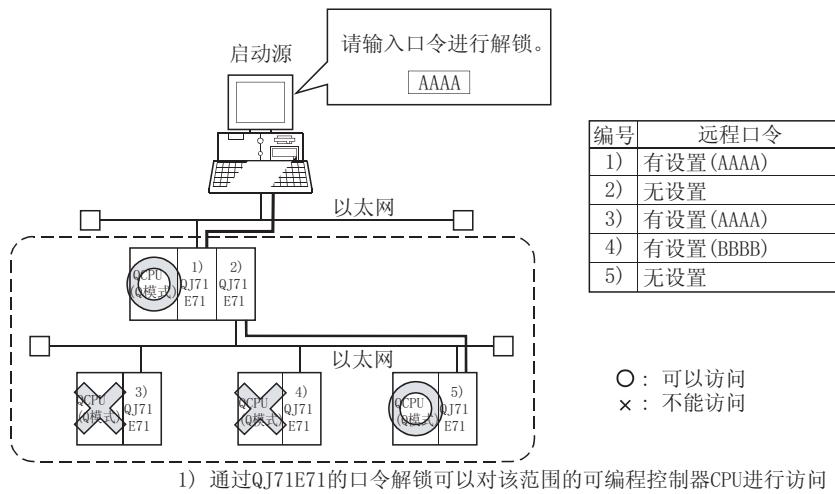
使用 TCP/IP 协议从多个计算机同时对同一模块进行通信的情况下, 应满足以下条件。

- 使用序列号的前 5 位为 “02122” 以后且功能版本 B 以后的 Q 系列 E71 模块 (QJ71E71-100 除外)。
- 使用 GX Developer Version 6.05F 以后版本, 将以太网参数的 [打开方式] 设置为 “MELSOFT 连接”。

(5) 关于使用 QJ71E71 时的口令解锁

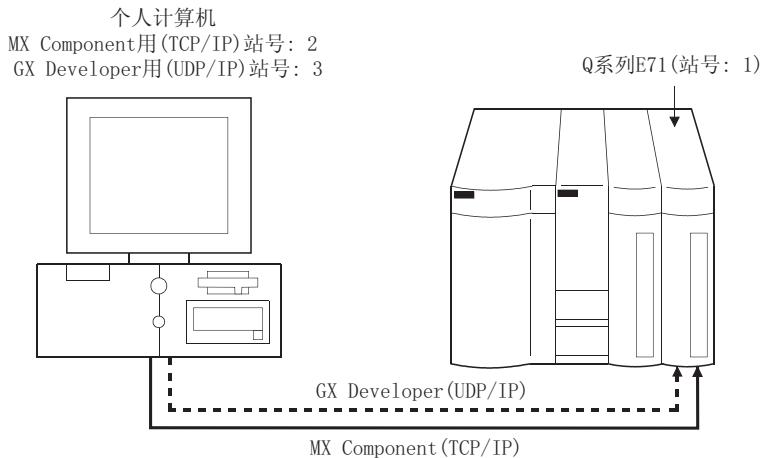
可通过远程操作进行口令解锁的范围为至连接对象站为止。

如果对低位分级也设置了口令，将无法与低位分级的可编程控制器 CPU 进行通信。



(6) 关于以太网通信

- (a) 以太网通信（协议为 TCP/IP 的情况下）中发生了 CPU 死机或以太网模块的复位时，通信线路将被断开。
在这种情况下应执行线路关闭处理 (Close) 后，进行重新打开处理 (Open)。
- (b) 使用 2 种通信方式（协议）从 1 个计算机对 1 个 Q 系列 E71 进行访问的情况下，需要进行 TCP/IP 用站号及 UDP/IP 用站号这 2 种设置。
但是，使用 MX Component Version 3 以后及序列号“05051”以后的 Q 系列 E71 的情况下，不需要分别进行 TCP/IP 用站号及 UDP/IP 用站号设置。
例) MX Component 使用 TCP/IP，GX Developer 使用 UDP/IP 的情况下



MX Component用(TCP/IP)站号及GX Developer用(UDP/IP)站号应设置不同的站号。
如果设置为相同的站号，以太网模块侧将发生出错。

使用 CC-Link 模块时的注意事项

(1) CC-Link 主站 • 本地站模块的软件版本

CC-Link 通信中使用的 CC-Link 主站 • 本地站模块应使用软件版本 “N” 以后的产品。

在 “M” 以前的软件版本的模块中，将无法正常运行。

使用串行通信模块时的注意事项

(1) 关于串行通信

(a) 在串行通信模块中，所有的连接中远程操作 “PAUSE” 将变为出错状态。

(b) 在 FX0N、FX1S、FX1N(C)、FX2N(C)、FX3S、FX3G(C)、FX3U(C)CPU 中进行串行通信的情况下，需要使用 FX 扩展端口。

(2) 连接计算机与串行通信模块时的注意事项

(a) 使用 QJ71C24-R2 的功能版本 A 的情况下

对于 MX Component 应用程序，只能使用 CH1 或 CH2 其中的一个。

此外，GX Developer、GOT 等的 MELSOFT 产品使用了其中一个通道时，另一个通道将不能使用。

但是，使用功能版本 B 的 QJ71C24-R2 的情况下，可以使用两个通道。

调制解调器通信时的注意事项

(1) 调制解调器通信时与其它应用程序的共存

进行调制解调器通信的情况下，MX Component 及 GX Developer 等其它应用程序将不能同时进行通信。

通过 MX Component 进行调制解调器通信的情况下，应避免在其它应用程序中进行调制解调器通信。

使用 MX Component 以及其它应用程序同时进行了调制解调器通信的情况下，将发生通信出错、电话线路断开等现象。

(2) 使用电话线路时的注意事项

(a) 请勿使用呼叫等待线路。

在呼叫等待线路中，中断的读取音有可能会发生数据干扰、电话线路断开等现象。

(b) 请勿进行至总机及分机的线路连接。

在至总机及分机的电话线路连接中拿起了分机等的听筒的情况下，电话线路有可能被断开。

(c) 电话线路应使用模拟 2 线式。

使用数字线路的情况下，应使用终端适配器。

此外，电话线路为 4 线式的情况下，根据模块化插口的配线类型有可能导致无法进行线路连接。

4 线式的情况下应事先进行连接测试，确认可否连接。

(3) 使用移动电话时的注意事项

(a) 关于使用移动电话进行无线通信时的调制解调器

根据各生产厂商调制解调器的名称有所不同，在本手册中将统称为移动电话用通信模块。

应根据使用的移动电话选择移动电话用通信模块的机型。

详细内容请咨询所使用的移动电话公司。

(b) 关于无自动呼叫功能的移动电话

对于无自动呼叫功能的移动电话，应使用具有 ANS/ORG/TEL 切换开关的移动电话用通信模块。

使用了无 ANS/ORG/TEL 切换开关的移动电话用通信模块的情况下，将无法进行线路连接。

此外，根据移动电话公司及移动电话的机型，线路连接的步骤有所不同。

详细内容请咨询所使用的移动电话的生产厂商。

编程时的注意事项

(1) 关于样本程序、测试程序、样本顺控程序

(a) 样本程序、测试程序

作为创建用户程序时的参考，附加了样本程序。

此外，为了进行通信测试，附加了测试程序。

关于这些程序的使用，应由用户自担风险。

(b) 样本顺控程序

对于 MX Component 中附加的样本顺控程序，需要根据系统配置、参数设置对其进行更改。

应修改为最适用于系统的内容。

此外，关于样本顺控程序的使用，应由用户自担风险。

(2) 关于通信中的进程的强制结束

在多个进程中打开相同类型的控制进行通信的情况下，如果通过任务管理器等对进程进行强制结束，其它进程可能通过通信函数执行部分而停止。

(3) 关于通信开始时的出错

在按下通信诊断按钮时、监视开始时、执行各函数时等的通信开始时，有时会在设置的超时值以内发生通信出错。这是由于在超时出错之前检测出了错误。

(例：未连接通信电缆、可编程控制器电源 OFF 时等)

(4) CheckDeviceString

各 ACT 控件中有 CheckDeviceString 方式，但请勿使用。

(5) ActUMsg 控件、ActUWzd 控件

安装 MX Component 时，ActUMsg 控件及 ActUWzd 控件将被登录，但请勿使用。

(6) 使用以太网模块时的注意事项

(a) 在执行 Open 方式后，执行 Close 方式之前，应设置以太网模块安装站的顺控程序扫描时间以上的间隔。

(b) 在执行 Close 方式后，再次执行 Open 方式之前应设置最少 500ms 以上的间隔。

(7) 执行 Disconnect 时的注意事项

由于某种原因导致即使执行 Disconnect 也无法断开电话线路的情况下，应切断电话所连接的调制解调器的电源，强制断开电话线路。

使用 Microsoft® Excel® 时的注意事项

(1) 使用 Excel VBA 时的注意事项

在使用了 Excel VBA 的应用程序中，请勿设置分页预览功能。

否则可能导致发生存储器溢出、操作系统基本操作（文件操作、打印等）异常现象。

(2) 使用 Microsoft® Excel® 时的注意事项

- (a) 粘贴操作时有时会发生无法将控件粘贴到 Excel 中的现象。

这是由于残留了 Excel 的缓存文件（临时文件）。

在这种情况下，应按以下步骤进行操作。

操作步骤

1. 结束 Excel。
2. 删除位于 temp 文件夹^{*1} 的 Excel8.0 文件夹中的 “*.exd”。^{*2}
3. 重新启动 Excel。

*1: temp 文件夹所在位置根据操作系统而有所不同。

*2: 无法显示相应文件夹、文件的情况下，应在文件夹选项设置中设置为显示所有文件及文件夹。

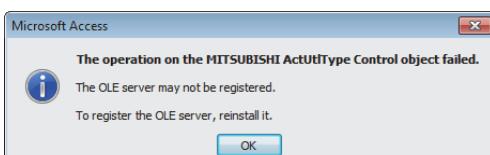
- (b) 在 Excel 中可以更改 ACT 控件的大小，但对 MX Component 的运行无影响。

将大小恢复为原状的情况下，应重新将 ACT 控件的 Height 属性及 Width 属性设置为“24”。

使用 Microsoft® Access® 时的注意事项

(1) 使用 Microsoft® Access® 时的注意事项

- (a) 将 ACT 控件粘贴到 Access 窗体中，双击 ACT 控件或选择属性中的自定义控件时将显示以下出错信息，但对 ACT 控件的运行无影响。
(有时也会显示其它出错信息。)



- (b) 粘贴 ACT 控件后进行了属性显示的情况下，有时会发生属性名显示不完整的现象。

此现象仅为属性显示问题，属性的功能方面不存在问题。

- (c) 在 Access 中可以更改 ACT 控件的大小，但对 MX Component 的运行无影响。

将大小恢复为原状的情况下，应重新将 ACT 控件的 Height 属性及 Width 属性设置为“24”。

使用 VBScript 时的注意事项

(1) 使用 VBScript 时的互联网 / 企业内部网的安全

MX Component 中没有互联网 / 企业内部网的安全功能。

需要使用安全功能的情况下，应在用户端进行设置。

使用机器人控制器时的注意事项

(1) 通过 USB 连接机器人控制器时

关于通过 USB 连接机器人控制器时的注意事项，请参阅以下手册。

 CR750/700/500 series RT ToolBox2 / RT ToolBox2 mini User's Manual

通过 USB 连接机器人控制器时，即使在机器人控制器无法使用的状态下，控制的 Open 方法中也不会发生错误。

在执行 Open 后使用的方法中发生了错误代码 106（连接被切断）时，应在执行 Close 后，重新执行 Open。

(2) 关于多重通信

请勿对 1 台机器人控制器进行多重通信。

前言

在此感谢贵方购买了三菱综合 FA 软件 MELSOFT 系列的产品。
本手册是用于让用户了解 MX Component 的有关内容的手册。

在使用之前应熟读本手册及关联手册，在充分了解 MX Component 的功能 • 性能的基础上正确地使用本产品。

关联手册

与本产品相关的手册如下所示。

请根据需要参考本表订购。

备注

“MX Component Version 4 操作手册”以 PDF 文件被存储在软件包的 CD-ROM 中。

备有用于另售的印刷品，希望单独购买手册的情况下，请根据上表中的手册编号购买。

目录

安全注意事项	1
关于产品的应用	2
使用注意事项	3
前言	11
关联手册	11
手册的阅读方法	17
术语	18
第 1 章 概要	21
1.1 控件的概要	21
1.2 控件及函数一览	23
1.2.1 控件一览	23
1.2.2 函数一览	24
第 2 章 关于控件	25
2.1 用于使用控件的设置	25
2.1.1 使用 VBA 的情况下	25
2.1.2 使用 VBScript 的情况下	29
2.1.3 使用 Visual Studio® .NET 的情况下	30
2.2 编程步骤	41
2.2.1 使用 VBA 的情况下	41
2.2.2 使用 VBScript 的情况下	42
2.2.3 使用 Visual Basic® .NET 的情况下	43
2.2.4 使用 Visual C++® .NET 的情况下	44
2.2.5 使用 Visual C#® .NET 的情况下	45
2.3 软元件类型	46
2.4 可访问范围	49
第 3 章 控件的属性	50
3.1 属性一览	50
3.2 属性的详细内容	52
第 4 章 各通信路径中设置的属性	65
4.1 设置的属性的阅读方法	65
4.2 串行通信	66
4.2.1 连接站为 R 系列 C24 的串行通信	66
4.2.2 连接站为 Q 系列 C24 的串行通信	72
4.2.3 连接站为 L 系列 C24 的串行通信	78
4.2.4 连接站为 FX 扩展端口的串行通信	83
4.3 以太网通信	85
4.3.1 连接站为 R 系列 E71 的以太网通信 (TCP)	85
4.3.2 连接站为 R 系列 E71 的以太网通信 (UDP)	88
4.3.3 连接站为 Q 系列 E71 的以太网通信 (TCP)	91
4.3.4 连接站为 Q 系列 E71 的以太网通信 (UDP)	94
4.3.5 连接站为 RCPU 的以太网通信 (TCP)	97

4.3.6 连接站为 RCPU 的以太网通信 (UDP)	100
4.3.7 连接站为以太网端口内置 QCPU 的以太网通信 (TCP)	103
4.3.8 连接站为以太网端口内置 QCPU 的以太网通信 (UDP)	107
4.3.9 连接站为以太网端口内置 LCPU 的以太网通信 (TCP)	111
4.3.10 连接站为以太网端口内置 LCPU 的以太网通信 (UDP)	114
4.3.11 连接站为以太网适配器模块的以太网通信 (TCP)	117
4.3.12 连接站为以太网适配器模块的以太网通信 (UDP)	121
4.3.13 连接站为以太网适配器的以太网通信 (TCP)	125
4.3.14 连接站为以太网适配器的以太网通信 (UDP)	126
4.3.15 连接站为机器人控制器的以太网通信	127
4.4 COM 通信	128
4.4.1 连接站为 QCPU(Q 模式) 的 CPU COM 通信.	128
4.4.2 连接站为 LCPU 的 CPU COM 通信.	131
4.4.3 连接站为 Q 运动 CPU 的 CPU COM 通信	134
4.4.4 连接站为 FXCPU 的 CPU COM 通信	136
4.4.5 连接站为变频器的 COM 通信	138
4.4.6 连接站为机器人控制器的 COM 通信	139
4.5 USB 通信	140
4.5.1 连接站为 RCPU 的 USB 通信.	140
4.5.2 连接站为 R 运动 CPU 的 USB 通信	143
4.5.3 连接站为 QCPU(Q 模式) 的 USB 通信.	146
4.5.4 连接站为 LCPU 的 USB 通信.	150
4.5.5 连接站为 QSCPU 的 USB 通信	153
4.5.6 连接站为 Q 运动 CPU 的 USB 通信	154
4.5.7 连接站为 FXCPU 的 USB 通信	156
4.5.8 连接站为 变频器的 USB 通信	158
4.5.9 连接站为 机器人控制器的 USB 通信	159
4.6 MELSECNET/H 通信	160
4.7 CC-Link IE 控制网络通信	164
4.8 CC-Link IE 现场网络通信	168
4.9 CC-Link 通信	172
4.10 CC-Link G4 通信	175
4.11 GX Simulator 通信	178
4.12 GX Simulator2 通信	178
4.13 MT Simulator2 通信	178
4.14 调制解调器通信	179
4.14.1 连接站为 Q 系列 C24 的调制解调器通信	179
4.14.2 连接站为 L 系列 C24 的调制解调器通信	186
4.14.3 连接站为 FXCPU 的调制解调器通信	192
4.15 网关功能通信	194
4.16 GOT 透明通信	195
4.16.1 计算机侧端口：串行，GOT2000/1000 侧端口：串行， CPU 侧端口：直接连接	195
4.16.2 计算机侧端口：USB，GOT2000/1000 侧端口：USB， CPU 侧端口：直接连接	203
4.16.3 计算机侧端口：串行，GOT2000/1000 侧端口：串行， CPU 侧端口：总线.	210

4.16.4	计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：总线.	213
4.16.5	计算机侧端口：串行, GOT2000/1000 侧端口：串行, CPU 侧端口：串行通信模块	216
4.16.6	计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：串行通信模块	232
4.16.7	计算机侧端口：串行, GOT2000/1000 侧端口：串行, CPU 侧端口：R 系列 E71	246
4.16.8	计算机侧端口：串行, GOT2000/1000 侧端口：串行, CPU 侧端口：Q 系列 E71	249
4.16.9	计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：R 系列 E71	252
4.16.10	计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：Q 系列 E71	255
4.16.11	计算机侧端口：串行, GOT2000/1000 侧端口：串行, CPU 侧端口：以太网端口.	258
4.16.12	计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网端口.	268
4.16.13	计算机侧端口：串行, GOT2000/1000 侧端口：串行, CPU 侧端口：以太网适配器模块	277
4.16.14	计算机端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网适配器模块	281
4.16.15	计算机侧端口：串行, GOT2000/1000 侧端口：串行, CPU 侧端口：以太网适配器 / 模块	285
4.16.16	计算机端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网适配器 / 模块	287
4.16.17	计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：串行.	289
4.16.18	计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：串行通信模块	296
4.16.19	计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：总线.	305
4.17	Q 系列总线通信	309

第 5 章 函数	310
----------	-----

5.1	编程时的注意事项	311
5.2	函数的详细内容 (ACT 控件用)	314
5.2.1	Open(通信线路的打开).	314
5.2.2	Close(通信线路的关闭)	316
5.2.3	ReadDeviceBlock(软元件的批量读取)	317
5.2.4	WriteDeviceBlock(软元件的批量写入)	321
5.2.5	ReadDeviceRandom(软元件的随机读取)	325
5.2.6	WriteDeviceRandom(软元件的随机写入)	329
5.2.7	SetDevice(软元件数据的设置)	333
5.2.8	GetDevice(软元件数据的获取)	335
5.2.9	ReadBuffer(缓冲存储器读取)	338
5.2.10	WriteBuffer(缓冲存储器写入)	343

5.2.11	GetClockData(时钟数据读取)	348
5.2.12	SetClockData(时钟数据写入)	353
5.2.13	GetCpuType(可编程控制器CPU型号读取)	358
5.2.14	SetCpuStatus(远程控制)	362
5.2.15	EntryDeviceStatus(软元件的状态监视登录)	366
5.2.16	FreeDeviceStatus(软元件的状态监视登录解除)	371
5.2.17	OnDeviceStatus(事件通知)	372
5.2.18	ReadDeviceBlock2(软元件的批量读取)	374
5.2.19	WriteDeviceBlock2(软元件的批量写入)	377
5.2.20	ReadDeviceRandom2(软元件的随机读取)	380
5.2.21	WriteDeviceRandom2(软元件的随机写入)	384
5.2.22	SetDevice2(软元件数据的设置)	388
5.2.23	GetDevice2(软元件数据的获取)	391
5.2.24	Connect(电话线路的连接)	394
5.2.25	Disconnect(电话线路的断开)	396
5.2.26	GetErrorMessage(出错信息的获取)	398
5.3	函数的详细内容(.NET控件用)	400
5.3.1	Open(通信线路的打开)	400
5.3.2	Close(通信线路的关闭)	401
5.3.3	ReadDeviceBlock(软元件的批量读取)	402
5.3.4	WriteDeviceBlock(软元件的批量写入)	405
5.3.5	ReadDeviceRandom(软元件的随机读取)	408
5.3.6	WriteDeviceRandom(软元件的随机写入)	414
5.3.7	SetDevice(软元件数据的设置)	420
5.3.8	GetDevice(软元件数据的获取)	422
5.3.9	ReadBuffer(缓冲存储器读取)	424
5.3.10	WriteBuffer(缓冲存储器写入)	426
5.3.11	GetClockData(时钟数据读取)	428
5.3.12	SetClockData(时钟数据写入)	430
5.3.13	GetCpuType(可编程控制器CPU型号读取)	432
5.3.14	SetCpuStatus(远程控制)	433
5.3.15	EntryDeviceStatus(软元件的状态监视登录)	434
5.3.16	FreeDeviceStatus(软元件的状态监视登录解除)	437
5.3.17	OnDeviceStatus(事件通知)	438
5.3.18	ReadDeviceBlock2(软元件的批量读取)	440
5.3.19	WriteDeviceBlock2(软元件的批量写入)	443
5.3.20	ReadDeviceRandom2(软元件的随机读取)	446
5.3.21	WriteDeviceRandom2(软元件的随机写入)	452
5.3.22	SetDevice2(软元件数据的设置)	458
5.3.23	GetDevice2(软元件数据的获取)	460
5.3.24	Connect(电话线路的连接)	462
5.3.25	Disconnect(电话线路的断开)	464
5.3.26	GetErrorMessage(出错信息的获取)	466

6.1	VBA样本程序	470
6.1.1	Excel样本程序	470

6.1.2 Excel 样本程序 (软元件读取 / 写入)	472
6.1.3 Access 样本程序	474
6.2 VBScript 样本程序	476
6.3 ASP 样本程序	478
6.4 Visual Basic® .NET 样本程序	481
6.4.1 调制解调器通信用样本程序	481
6.4.2 Read/Write 样本程序	483
6.4.3 型转换样本程序	486
6.5 Visual C++® .NET 样本程序	488
6.5.1 Read/Write 样本程序	488
6.5.2 故障排除功能样本程序	489
6.6 Visual C#® .NET 样本程序	491
6.6.1 Read/Write 样本程序	491
6.7 Visual C++® .NET (MFC) 样本程序	492
6.7.1 调度 I/F	492
6.7.2 自定义 I/F	494
6.7.3 故障排除功能样本程序	495

第 7 章 出错代码	496
------------	-----

7.1 各控件返回的出错代码	496
7.2 CPU、模块、网卡返回的出错代码	527
7.3 HRESULT 型的出错代码	529
7.4 事件查看器中显示的出错代码	529

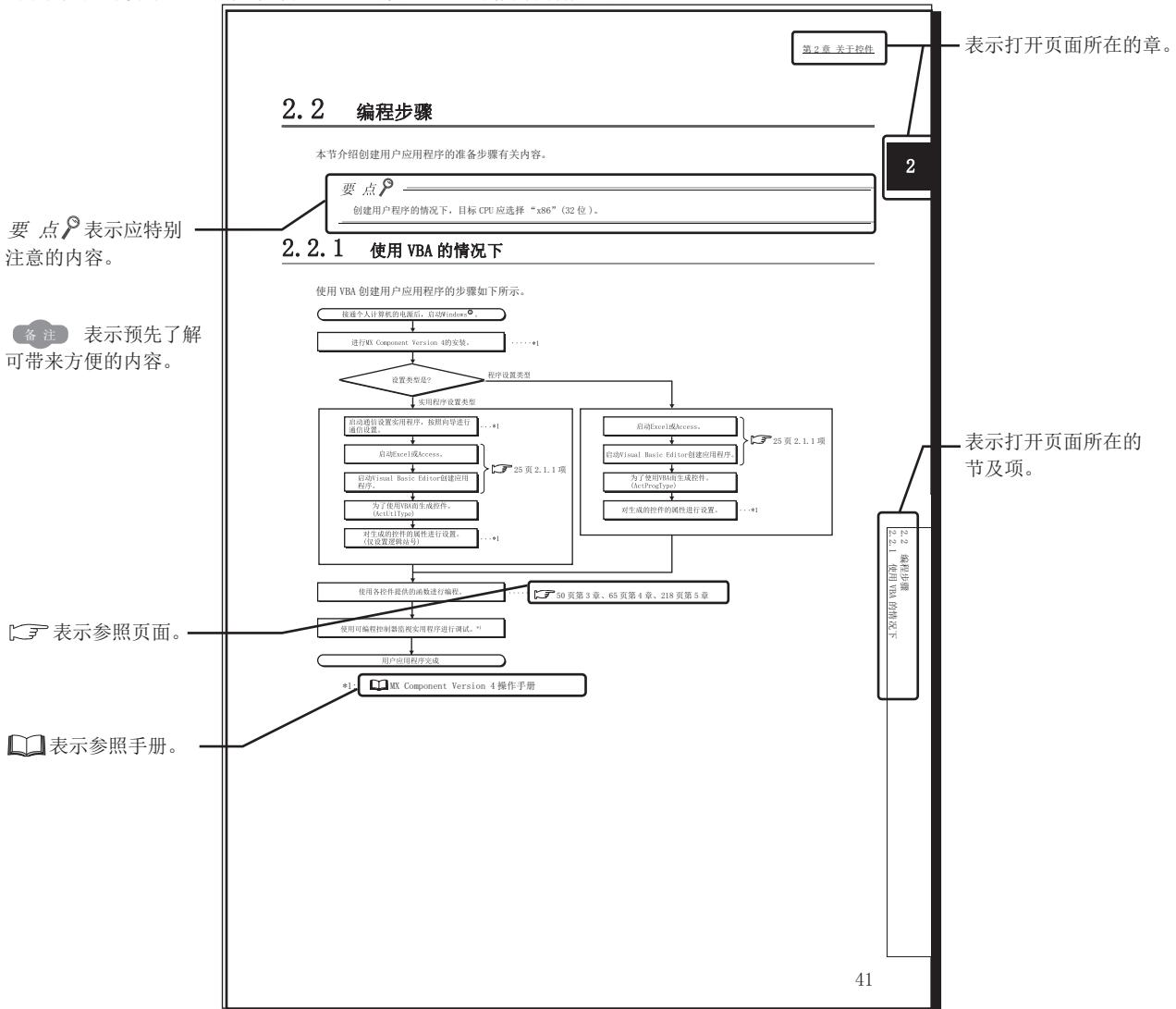
附录	530
----	-----

附录 1 关于回调功能的连接方式	530
附录 2 字软元件状态监视情况下的编程示例	531
附录 3 关于超时时间	534
附录 3.1 由于超时出错而重试的情况下	534
附录 3.2 由于接收数据出错而重试的情况下	549
附录 3.3 由于 ACT 控件内部的固定时间发生超时的情况下	551
修订记录	557
质保	559

手册的阅读方法

以下对本手册页面构成及符号有关内容进行说明。

以下为手册阅读方法的相关说明，因此与实际的记载内容有所不同。



本手册中使用的符号及其内容示例如下所示。

符号	内容	示例
[]	菜单栏的菜单名	[工具] → [属性]
<< >>	画面的选项卡名	<<.NET>> 选项卡
“ ”	画面内的各项目名	“浏览设置”
	画面的按钮	(确定) 按钮

术语

在本手册中，除非特别标明，将使用下述术语进行说明。

术语	内容
MX Component	是产品型号 SWnDNC-ACT(-E)、SWnDNC-ACT-A(-EA) 的产品名总称。 (n = 版本。) -A 及 -EA 表示多个许可产品。
计算机	是基于 Windows [®] 运行的个人计算机的总称。
计算机 CPU 模块	是 CONTEC CO., LTD. 生产的 MELSEC-Q 系列个人计算机 CPU 模块的略称。
GX Developer	是产品型号 SWnD5C-GPPW、SWnD5C-GPPW-A、SWnD5C-GPPW-V、SWnD5C-GPPW-VA 的产品名总称。 (n = 版本。) -A 表示多个许可产品，-V 表示版本升级产品。
GX Works2	是产品型号 SWnDNC-GXW2 的产品名总称。(n = 版本。)
MT Developer2	是产品型号 SWnDNC-MTW2 的产品名总称。(n = 版本。)
GX Simulator	是产品型号 SWnD5C-LLT、SWnD5C-LLT-A、SWnD5C-LLT-V、SWnD5C-LLT-VA 产品名总称。 (n = 版本。) -A 表示多个许可产品，-V 表示版本升级产品。
MELSECNET/H 板	是 Q80BD-J71LP21-25、Q80BD-J71LP21S-25、Q81BD-J71LP21-25、Q80BD-J71LP21G、Q80BD-J71BR11 的总称。 是 MELSECNET/H 接口板的略称。
CC-Link IE 控制网卡	是 Q80BD-J71GP21-SX、Q80BD-J71GP21S-SX 的总称。 是 CC-Link IE 控制网络接口板的略称。
CC-Link IE 现场网卡	是 Q81BD-J71GF11-T2 型 CC-Link IE 现场网络接口板的略称。
CC-Link 板	是 Q80BD-J61BT11N、Q81BD-J61BT11 的总称。 是 CC-Link 系统主站・本地站接口板的略称。
RCPU	是 R04、R08、R16、R32、R120 的总称。
QCUP (Q 模式)	是 Q00J、Q00UJ、Q00、Q00U、Q01、Q01U、Q02、Q02H、Q02PH、Q02U、Q03UD、Q03UDE、Q03UDV、Q04UDH、Q04UDEH、Q04UDV、Q06H、Q06PH、Q06UDH、Q06UDEH、Q06UDV、Q10UDH、Q10UDEH、Q12H、Q12PH、Q12PRH、Q13UDH、Q13UDEH、Q13UDV、Q20UDH、Q20UDEH、Q25H、Q25PH、Q25PRH、Q26UDH、Q26UDEH、Q26UDV、Q50UDEH、Q100UDEH 的总称。
以太网端口内置 QCUP	是 Q03UDE、Q03UDV、Q04UDEH、Q04UDV、Q06UDEH、Q06UDV、Q10UDEH、Q13UDEH、Q13UDV、Q20UDEH、Q26UDEH、Q26UDV、Q50UDEH、Q100UDEH 的总称。
LCPU	是 L02S、L02、L06、L26、L26-BT 的总称。
以太网内置型 CPU	是 RCPU、以太网端口内置 QCUP、LCPU 的总称。
FXCPU	是 FX0、FX0s、FX0n、FX1、FX1n、FX1s、FX2、FX2n、FX2nc、FX3s、FX3g、FX3gc、FX3u、FX3uc 的总称。
R 运动 CPU	是 R16MT、R32MT 的总称。
Q 运动 CPU	是 Q172、Q173、Q172H、Q173H、Q172D、Q173D、Q172DS、Q173DS 的总称。
QSCPU	是 QS001CPU(安全 CPU) 的略称。
C 语言控制器	是 Q12DCCPU-V(基本功能模式)、Q12DCCPU-V(功能扩展模式)、Q24DHCCPU-V、Q24DHCCPU-LS 的总称。
Q12DCCPU-V (基本功能模式)	在基本功能模式下对 Q12DCCPU-V 进行了初始化时的状态。 关于 Q12DCCPU-V(基本功能模式)，请参阅以下手册。  C 语言控制器模块用户手册 (硬件设计 / 功能解说篇)
Q12DCCPU-V (功能扩展模式)	在功能扩展模式下对 Q12DCCPU-V 进行了初始化时的状态。 关于 Q12DCCPU-V(功能扩展模式)，请参阅以下手册。  MELSEC-Q C 语言控制器模块用户手册

术语	内容
可编程控制器 CPU	是 RCPU、QCPU(Q 模式)、LCPU、FXCPU、R 运动 CPU、Q 运动 CPU、QSCPU 及 C 语言控制器的总称。
R 系列 C24	是 RJ71C24、RJ71C24-R2、RJ71C24-R4 的总称。
Q 系列 C24	是 QJ71C24、QJ71C24-R2、QJ71C24N、QJ71C24N-R2、QJ71C24N-R4 的总称。
L 系列 C24	是 LJ71C24、LJ71C24-R2 的总称。
FX 扩展端口	是 FX _{0N} -485ADP、FX _{2NC} -485ADP、FX _{1N} -485-BD、FX _{2N} -485-BD、FX _{3G} -485-BD、FX _{3U} -485-BD、FX _{3U} -485ADP 的总称。
串行通信模块	是 R 系列 C24、Q 系列 C24、L 系列 C24、FX 扩展端口的总称。
R 系列 E71	是 RJ71EN71 的总称。
Q 系列 E71	是 QJ71E71、QJ71E71-B2、QJ71E71-B5、QJ71E71-100 的总称。
以太网适配器模块	是 NZ2GF-ETB 型 CC-Link IE 现场网络以太网适配器模块的略称。
以太网适配器 / 模块	是 FX3U-ENET-ADP、FX3U-ENET(-L) 的总称。
以太网模块	是 R 系列 E71、Q 系列 E71 的总称。
CC-Link G4 模块	是 AJ65BT-G4-S3 型 GPP 功能用外围设备连接模块的略称。
GOT	是图形操作终端的略称。
GOT2000	是图形操作终端 GOT2000 系列的略称。
GOT1000	是图形操作终端 GOT1000 系列的略称。
GOT900	是图形操作终端 GOT900 系列的略称。
变频器	是 FREQROL-A800 系列的略称。
机器人控制器	CR750-D/CRnD-700 系列的略称。
串行通信	是使用串行通信模块与可编程控制器 CPU 进行通信时的略称。
以太网通信	是计算机与以太网模块或以太网内置型 CPU 相连接进行通信时的略称。
CPU COM 通信	是将计算机连接到可编程控制器 CPU 的 RS-232 接口或 RS-422 接口进行通信时的略称。
CPU USB 通信	是将计算机与 QCPU(Q 模式)、LCPU 的 USB 连接器相连接进行通信时的略称。
MELSECNET/H 通信	是使用 MELSECNET/H 板与可编程控制器 CPU 进行通信时的略称。
CC-Link IE 控制网络通信	是使用 CC-Link IE 控制网卡与可编程控制器 CPU 进行通信时的略称。
CC-Link IE 现场网络通信	是使用 CC-Link IE 现场网卡与可编程控制器 CPU 进行通信时的略称。
CC-Link 通信	是使用 CC-Link 板与可编程控制器 CPU 进行通信时的略称。
CC-Link G4 通信	是使用 CC-Link G4 模块与可编程控制器 CPU 进行通信时的略称。
Q 系列总线通信	是使用计算机 CPU 模块与同一基板上的可编程控制器与可编程控制器 CPU 进行通信时的略称。
GX Simulator 通信	是与 GX Simulator 进行通信时的略称。
GX Simulator2 通信	是使用 GX Works2 的模拟功能进行通信时的略称。
MT Simulator2 通信	是使用 MT Developer2 的模拟功能进行通信时的略称。
调制解调器通信	是使用 Q 系列 C24、L 系列 C24、FXCPU 经由调制解调器与可编程控制器 CPU 进行通信时的略称。
网关功能通信	是使用 GOT 的网关功能与可编程控制器 CPU 及各公司可编程控制器进行通信时的略称。
GOT 透明通信	是使用 GOT 的透明功能与可编程控制器 CPU 进行通信时的略称。
变频器 COM 通信	将变频器连接到计算机的 COM 端口上进行通信时的略称。
变频器 USB 通信	将变频器连接到计算机的 USB 端口上进行通信时的略称。
机器人控制器 COM 通信	将机器人控制器连接到计算机的 COM 端口上进行通信时的略称。

术语	内容
机器人控制器 USB 通信	将机器人控制器连接到计算机的 USB 端口上进行通信时的略称。
机器人控制器以太网通信	将机器人控制器与计算机连接到以太网上进行通信时的略称。
实用程序设置类型	是使用通信设置实用程序创建用户程序的开发类型。
程序设置类型	是不使用通信设置实用程序创建用户程序的开发类型。
ACT 控件	是 MX Component 提供的 ActiveX 控件的总称。
.NET 控件	是 MX Component 提供的 .NET 控件的总称。
冗余 CPU	是 Q12PRH、Q25PRH 的总称。
冗余扩展基板	是 Q65WRB 型 CPU・电源冗余系统用扩展基板的略称。
Windows® 8.1	是 Microsoft® Windows® 8.1 Operating System、 Microsoft® Windows® 8.1 Pro Operating System、 Microsoft® Windows® 8.1 Enterprise Operating System 的总称。 但是，仅指 32 位版的情况下记载“Windows® 8.1(32 位版)”，仅指 64 位版的情况下记载“Windows® 8.1(64 位版)”。
Windows® 8	是 Microsoft® Windows® 8 Operating System、 Microsoft® Windows® 8 Pro Operating System、 Microsoft® Windows® 8 Enterprise Operating System 的总称。 但是，仅指 32 位版的情况下记载“Windows® 8(32 位版)”，仅指 64 位版的情况下记载“Windows® 8(64 位版)”。
Windows® 7	是 Microsoft® Windows® 7 Starter Operating System、 Microsoft® Windows® 7 Home Premium Operating System、 Microsoft® Windows® 7 Professional Operating System、 Microsoft® Windows® 7 Ultimate Operating System、 Microsoft® Windows® 7 Enterprise Operating System 的总称。 但是，仅指 32 位版的情况下记载“Windows® 7(32 位版)”，仅指 64 位版的情况下记载“Windows® 7(64 位版)”。
Windows Vista®	是 Microsoft® Windows Vista® Home Basic Operating System、 Microsoft® Windows Vista® Home Premium Operating System、 Microsoft® Windows Vista® Business Operating System、 Microsoft® Windows Vista® Ultimate Operating System、 Microsoft® Windows Vista® Enterprise Operating System 的总称。
Windows XP®	是 Microsoft® Windows XP® Professional Operating System、 Microsoft® Windows XP® Home Edition Operating System 的总称。
Excel	是 Microsoft® Excel® 2003、Microsoft® Excel® 2007、Microsoft® Excel® 2010(32 位版)、 Microsoft® Excel® 2013(32 位版) 的略称。
Access	是 Microsoft® Access® 2003、Microsoft® Access® 2007、Microsoft® Access® 2010(32 位版)、 Microsoft® Access® 2013(32 位版) 的略称。
Visual Basic® .NET	Microsoft® Visual Studio® 2005、Microsoft® Visual Studio® 2008、 Microsoft® Visual Studio® 2010、Microsoft® Visual Studio® 2012、 Microsoft® Visual Studio® 2013 的 Visual Basic 的总称。
Visual C++® .NET	是使用 .net Framework 创建应用程序时的略称。
Visual C++® .NET (MFC)	是使用 MFC/ATL/Win32 创建应用程序时的略称。
Visual C#® .NET	Microsoft® Visual Studio® 2005、Microsoft® Visual Studio® 2008、 Microsoft® Visual Studio® 2010、Microsoft® Visual Studio® 2012、 Microsoft® Visual Studio® 2013 的 Visual C# 的总称。

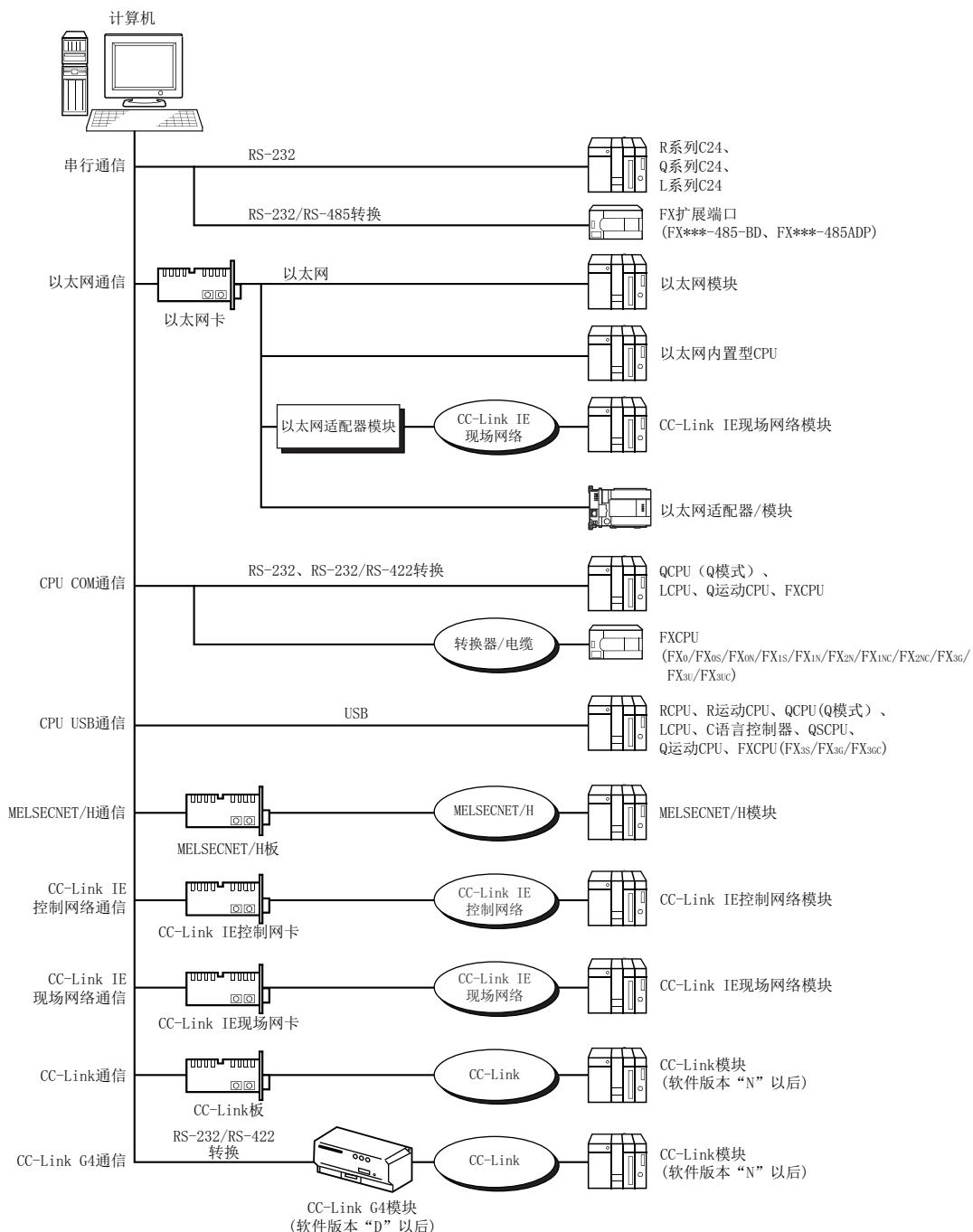
第1章 概要

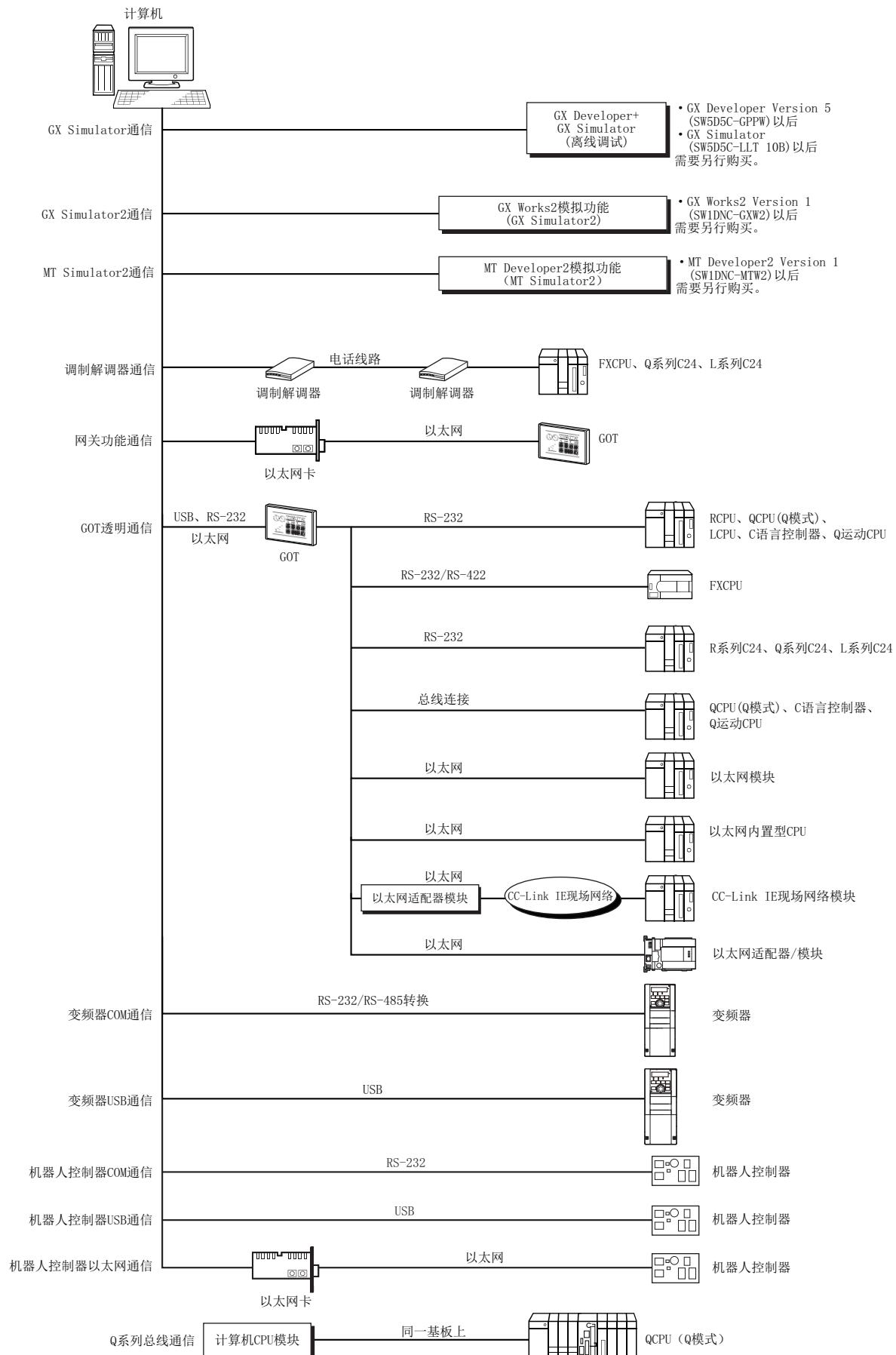
本手册中记载了使用 MX Component 进行编程的步骤及出错代码有关内容。

1.1 控件的概要

本节介绍 MX Component 提供的控件的功能概要有关内容。

本控件用于创建对可编程控制器 CPU 进行通信的用户程序。由此，可以无需考虑对象的硬件、通信协议即可进行通信。





1.2 控件及函数一览

控件及函数的一览如下所示。

1.2.1 控件一览

MX Component 提供的包含在各 DLL 中的控件的一览如下所示。

(1) ACT 控件

MX Component 提供的 ActiveX 控件如下所示。

ACT 控件可以使用软元件对数据进行访问。

DLL 名	包含的控件名		用途
	VB、VC++、 VC#、VBA 用	VBScript 用	
ActUtlType.dll	ActUtlType	ActMLUtlType*1	是实用程序设置类型的控件。 在使用通信设置实用程序创建用户程序的情况下使用此控件。
ActProgType.dll*2	ActProgType	ActMLProgType*1	是程序设置类型的控件。 在不使用通信设置实用程序创建用户程序的情况下使用此控件。
ActSupportMsg.dll	ActSupportMsg	ActMLSupportMsg	用于故障排除功能。

*1：通信路径为调制解调器的情况下不能进行通信。

*2：变频器通信 / 机器人控制器通信时无法使用。

(2) .NET 控件

MX Component 提供的 .NET 控件如下所示。

.NET 控件可以使用标签对数据进行访问。

DLL 名	包含的控件名		用途
	VB、VC++、VC# 用	For VBScript、VBA	
DotUtlType.dll	DotUtlType	-	是实用程序设置类型的控件。 在使用通信设置实用程序创建用户程序的情况下使用此控件。
DotSupportMsg.dll	DotSupportMsg	-	用于故障排除功能。

1.2.2 函数一览

各函数的功能及各控件中可使用的函数的一览如下所示。

函数名	功能	参照
Open	通信线路、电话线路的打开	314页5.2.1项、400页5.3.1项
Close	通信线路的关闭	316页5.2.2项、401页5.3.2项
ReadDeviceBlock	软元件的批量读取(4字节数据)	317页5.2.3项、402页5.3.3项
WriteDeviceBlock	软元件的批量写入(4字节数据)	321页5.2.4项、405页5.3.4项
ReadDeviceRandom	软元件的随机读取(4字节数据)	325页5.2.5项、408页5.3.5项
WriteDeviceRandom	软元件的随机写入(4字节数据)	329页5.2.6项、414页5.3.6项
SetDevice	软元件1点的设置(4字节数据)	333页5.2.7项、420页5.3.7项
GetDevice	软元件1点的数据获取(4字节数据)	335页5.2.8项、422页5.3.8项
ReadBuffer	缓冲存储器的读取	338页5.2.9项、424页5.3.9项
WriteBuffer	缓冲存储器的写入	343页5.2.10项、426页5.3.10项
GetClockData	可编程控制器CPU的时钟数据读取	348页5.2.11项、428页5.3.11项
SetClockData	可编程控制器CPU的时钟数据写入	353页5.2.12项、430页5.3.12项
GetCpuType	可编程控制器CPU型号读取	358页5.2.13项、432页5.3.13项
SetCpuStatus	可编程控制器CPU的远程RUN/STOP/PAUSE	362页5.2.14项、433页5.3.14项
EntryDeviceStatus	软元件的状态监视登录	366页5.2.15项、434页5.3.15项
FreeDeviceStatus	软元件的状态监视登录的解除	371页5.2.16项、437页5.3.16项
OnDeviceStatus	事件通知	372页5.2.17项、438页5.3.17项
ReadDeviceBlock2	软元件的批量读取(2字节数据)	374页5.2.18项、440页5.3.18项
WriteDeviceBlock2	软元件的批量写入(2字节数据)	377页5.2.19项、443页5.3.19项
ReadDeviceRandom2	软元件的随机读取(2字节数据)	380页5.2.20项、446页5.3.20项
WriteDeviceRandom2	软元件的随机写入(2字节数据)	384页5.2.21项、452页5.3.21项
SetDevice2	软元件1点的设置(2字节数据)	388页5.2.22项、458页5.3.22项
GetDevice2	软元件1点的数据获取(2字节数据)	391页5.2.23项、460页5.3.23项
Connect	至电话线路的连接	394页5.2.24项、462页5.3.24项
Disconnect	电话线路的断开	396页5.2.25项、464页5.3.25项
GetErrorMessage	出错内容及处理方法的显示	398页5.2.26项、466页5.3.26项

要 点

- 通信对象可编程控制器CPU为QSCPU的情况下
为了保护安全可编程控制器系统，执行对缓冲存储器的写入、对软元件的写入・设置、时钟数据的写入的函数时，将返回出错代码。

第2章 关于控件

本章介绍用于使用控件的设置、编程步骤、软元件类型及可访问范围有关内容。

2.1 用于使用控件的设置

本节介绍用于使用控件的设置操作有关内容。

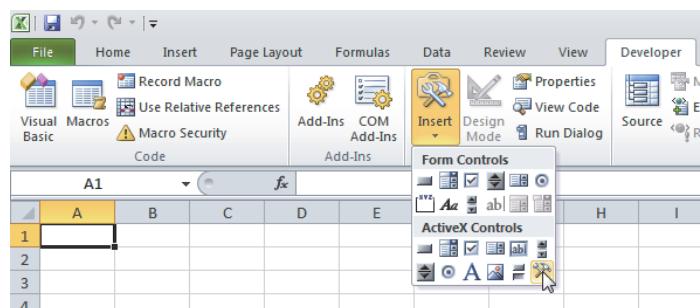
2.1.1 使用 VBA 的情况下

使用 VBA 的情况下的设置操作如下所示。

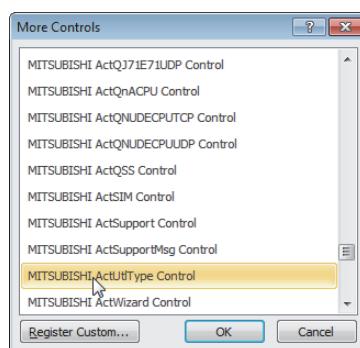
(1) 使用 Microsoft® Excel 2010 的情况下

操作步骤

- 启动 Excel，选择 <<Insert(开发)>> 选项卡的 [Developer(插入)] 后，点击  按钮。



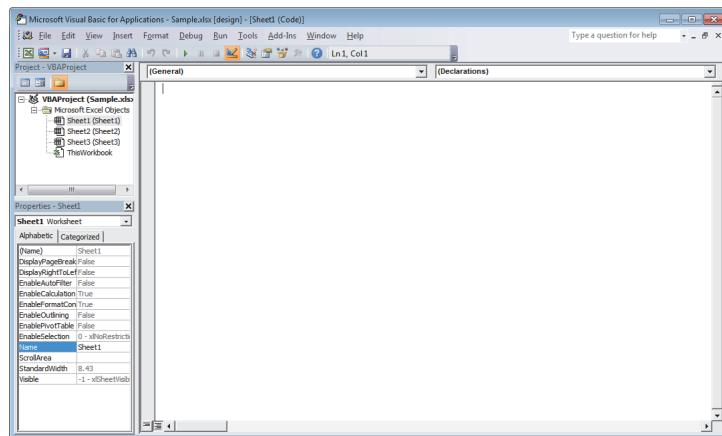
- 选择希望使用的 ACT 控件后，点击  按钮。



3. 将选择的 ACT 控件粘贴到表单中。

	A	B	C
1			
2		URL	
3			
4			
5			
6			
7			

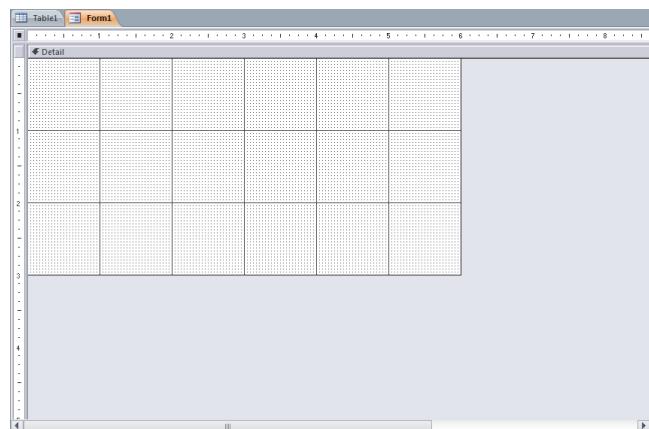
4. 选择<<Developer(开发)>>选项卡的[Visual Basic]，启动 Visual Basic Editor。



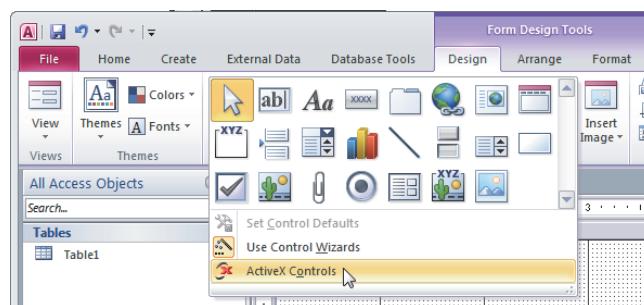
5. 通过 Visual Basic Editor 进行编程。

(2) 使用 Microsoft® Access 2010 的情况下

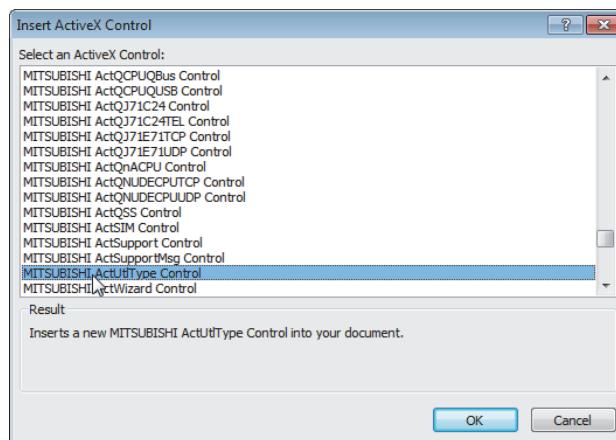
1. 启动 Access 后，激活数据库的窗体。



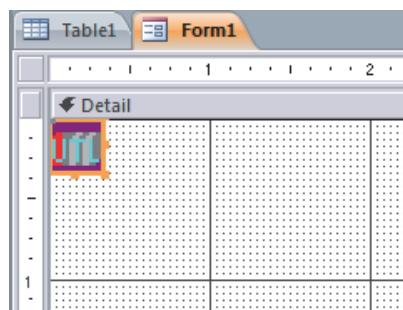
2. 选择 <>Design(设计)>> 选项卡的 [Controls(控件)] → [ActiveX Controls(ActiveX 控件)]。



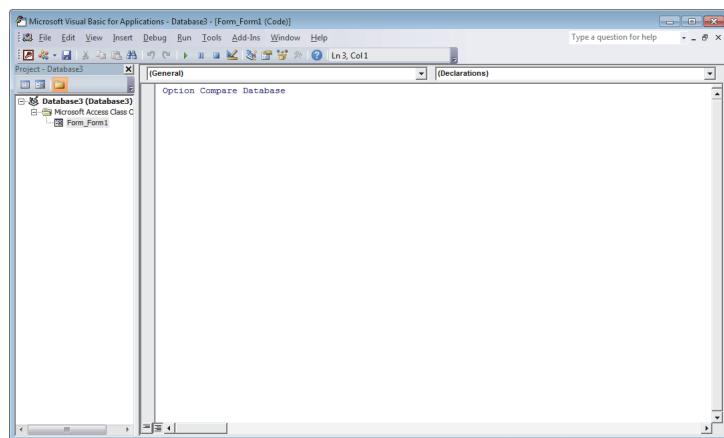
3. 选择希望使用的 ACT 控件后，点击 按钮。



4. 将选择的 ACT 控件粘贴到表单中。



5. 选择<<Design(设计)>>选项卡的[View Code(代码的显示)], 启动 Visual Basic Editor。



6. 通过Visual Basic Editor进行编程。

2.1.2 使用 VBScript 的情况下

使用记事本或市面上销售的文本编辑器、HTML 创建工具等创建 HTML、ASP。

关于 HTML、ASP 语法，请参阅市面上销售的参考书。

此外，由于 MX Component 安装有 HTML 样本程序、ASP 样本程序，因此也请参阅这些样本程序。

2.1 用于使用控件的设置
2.1.2 使用 VBScript 的情况下

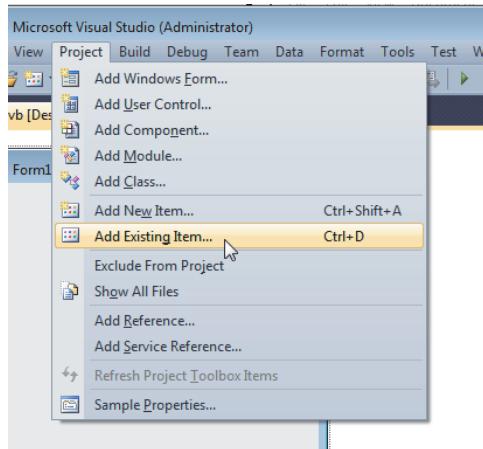
2.1.3 使用 Visual Studio® .NET 的情况下

使用 Visual Studio® .NET 的情况下的设置操作如下所示。

(1) 包含文件的设置 (Visual Basic® .NET 的情况下)

操作步骤

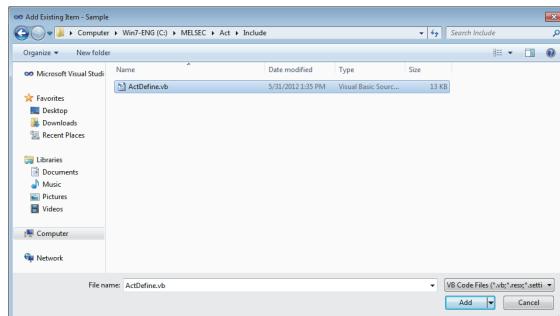
1. 启动工程后，选择 [Project(工程)] → [Add Existing Item(现有项目的添加)]。



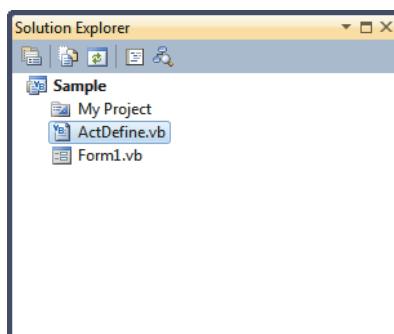
2. 浏览 ActDefine.vb 文件，点击 (添加) 按钮。

安装 ActDefine.vb 文件时，被存储在下述位置。

<用户指定文件夹> - <Act> - <Include>



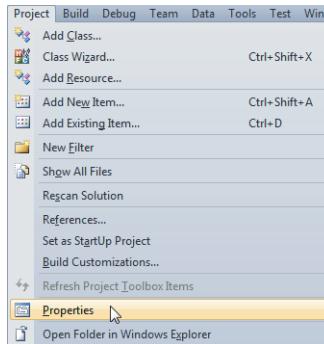
3. 解决方案资源管理器 • 窗口内将显示 ActDefine.vb 文件。



(2) 包含文件的设置 (Visual C++® .NET的情况下)

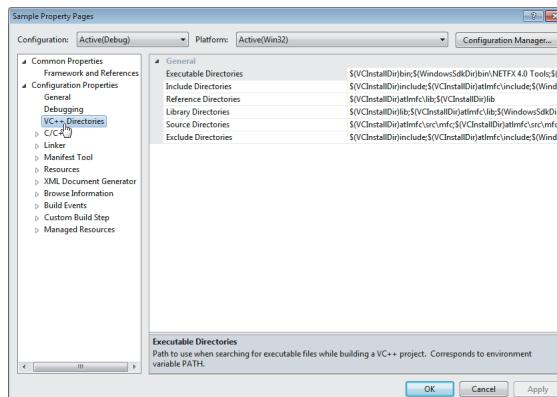
操作步骤

1. 启动 Visual Studio® .NET 后, 选择 [Project(工程)] → [Property(属性)]。*1



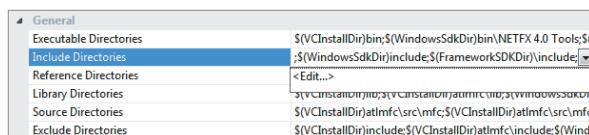
*1: Visual Studio® 2005 及 Visual Studio® 2008 的情况下选择 [Tools(工具)] → [Options(选项)]。

2. 在显示的画面左侧的导航窗格中, 选择 [Configuration Properties(构成属性)] → [VC++ Directories(VC++ 目录)]。*2

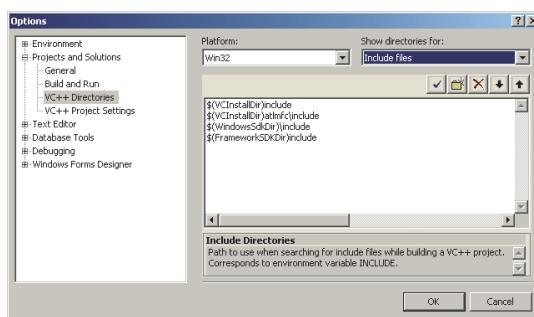


*2: Visual Studio® 2005 及 Visual Studio® 2008 的情况下选择 [Projects and Solutions(工程及解决方案)] → [VC++ Directories(VC++ 目录)]。

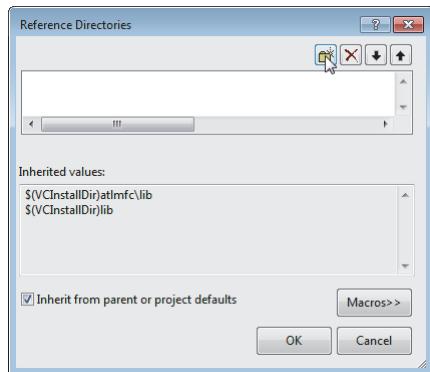
3. 在选中画面右侧的 “Include Directories(包含目录)” 的状态下右击鼠标, 选择 <Edit... (编辑...)>。*3



*3: Visual Studio® 2005 及 Visual Studio® 2008 的情况下在画面右上方的 “Shows directories for:(显示目录的工作)” 中选择 “Include files(包含文件)”。



4. 选择  (新起一行) 按钮。



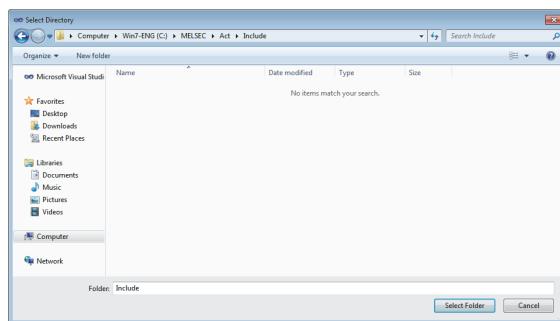
5. 然后选择  按钮。



6. 浏览有包含文件的文件夹。

安装包含文件时，被存储在下述位置。

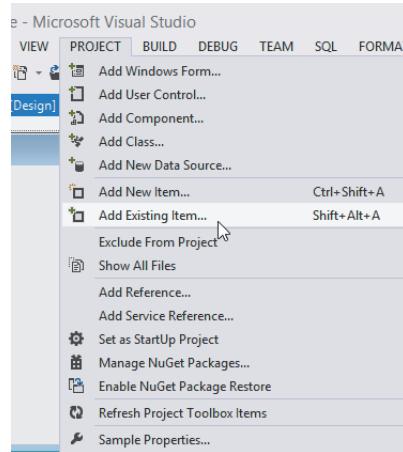
<用户指定文件夹> - <Act> - <Include>



(3) 包含文件的设置 (Visual C#® .NET 的情况下)

操作步骤

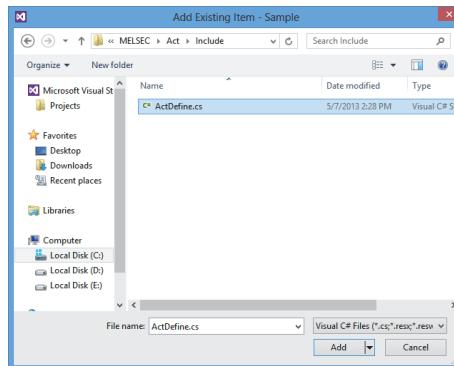
1. 启动 Visual Studio® .NET 后, 选择 [Project(工程)] → [Add Existing Item(现有项目的添加)]。



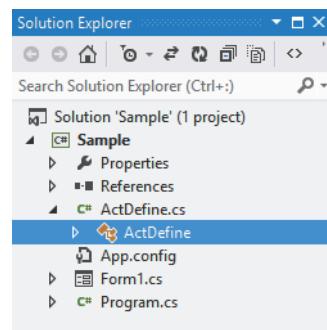
2. 浏览 ActDefine.cs 文件, 点击 (添加) 按钮。

安装 ActDefine.cs 文件时, 被存储在下述位置。

<用户指定文件夹> - <Act> - <Include>



3. 解决方案资源管理器窗口内将显示 ActDefine.cs 文件。



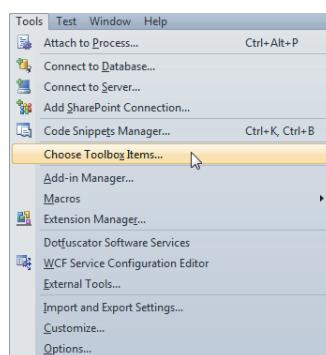
2.1 用于使用控件的设置
2.1.3 使用 Visual Studio® .NET 的情况下

(4) 将控件粘贴到窗体中使用的情况下

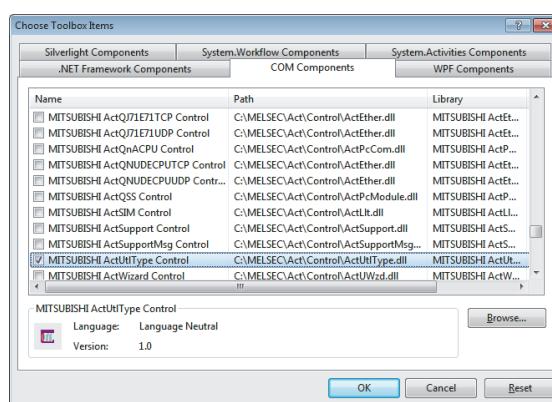
(Visual C++® .NET/Visual Basic® .NET/Visual C#® .NET 工程通用)

操作步骤

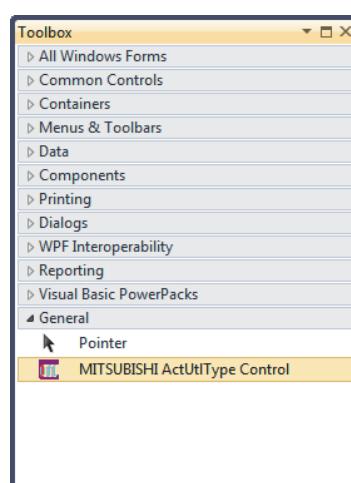
1. 选择 [Tools(工具)] → [Choose Toolbox Items(工具箱项目的选择)]。



2. 选择 <<COM Components(COM 组件)>> 选项卡 (粘贴 ACT 控件时) 或 <<.NET Framework Components(.NET Framework 组件)>> 选项卡 (粘贴 .NET 控件时), 勾选希望使用的控件后, 点击 **OK** 按钮。



3. 控件将被添加到 “Toolbox(工具箱)” 中选择的选项卡的最下部。



要 点

- 创建使用了.NET Framework 4/.NET Framework 4.5 的应用程序的情况下
在包含有应用程序.exe 文件的文件夹中，应添加记述了下述内容的应用程序配置文件 (app.config)。

```
<configuration>
    <startup useLegacyV2RuntimeActivationPolicy="true">
        <supportedRuntime version="v4.0"/>
    </startup>
</configuration>
```

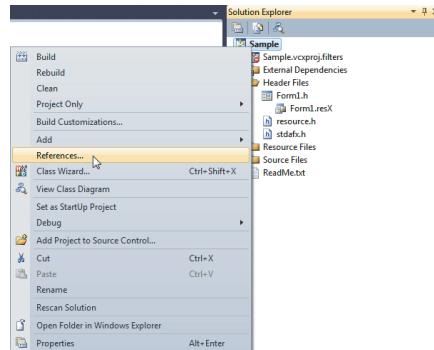
有关详细内容请参阅 Visual Studio[®]的帮助。

(5) 不将控件粘贴到窗体中使用的情况下（浏览设置）

(Visual C++[®] .NET/Visual Basic[®] .NET/Visual C#[®] .NET 工程通用)

操作步骤

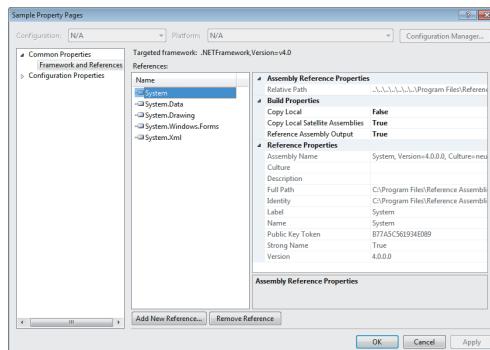
1. 选择 [View(显示)] → [Solution Explorer(解决方案资源管理器)] 显示 “Solution Explorer(解决方案资源管理器)”。
2. 鼠标右击工程选择 “Reference(浏览)”。^{*1}



*1: Visual Studio[®] 2010 及 Visual Studio[®] 2012 的情况下选择 “Add Reference(浏览的添加)”。(至步骤 4)

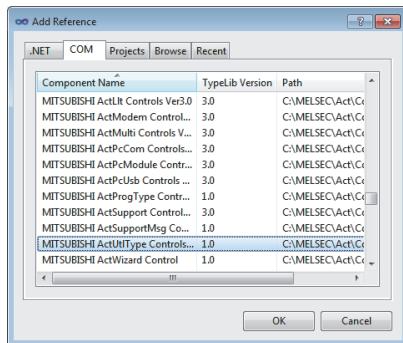
Visual Studio[®] 2013 的情况下选择 “Add (添加)” → “References (参照)”。(至步骤 4)

3. 在 “Property Pages(属性页面)” 中选择 **Add New Reference...** (新浏览的添加) 按钮。

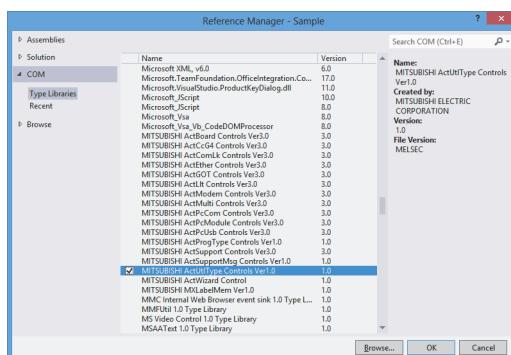


2.1 用于使用控件的设置
2.1.3 使用 Visual Studio[®] .NET 的情况下

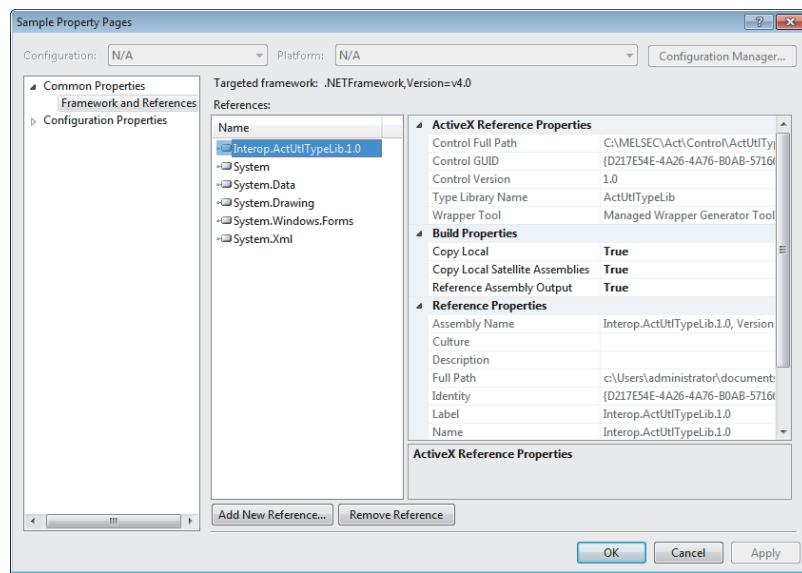
4. 选择<<COM>>选项卡（粘贴 ACT 控件时）或<<.NET>>选项卡（粘贴 .NET 控件时），在选中使用的控件的状态下点击  按钮。^{*1}



*1: (Visual Studio[®] 2012 及 Visual Studio[®] 2013 的情况下，选择“COM”→“Type Libraries（类型库）”（粘贴 ACT 控件时）或“Assemblies（集）”→“Extensions（扩展）”（粘贴 .NET 控件时），勾选要使用的控件，点击  按钮。)

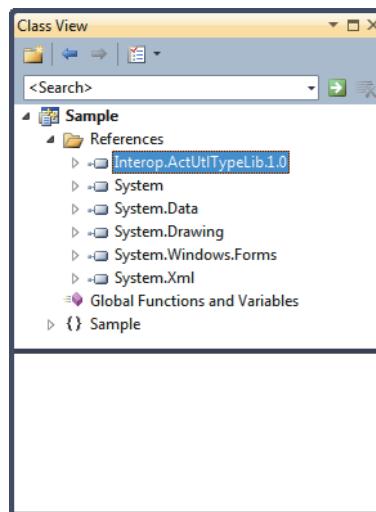


5. “References(浏览)” 中使用的组件的库将被进行浏览设置。



对于使用的组件的库，通过分类视图的浏览也可确认。

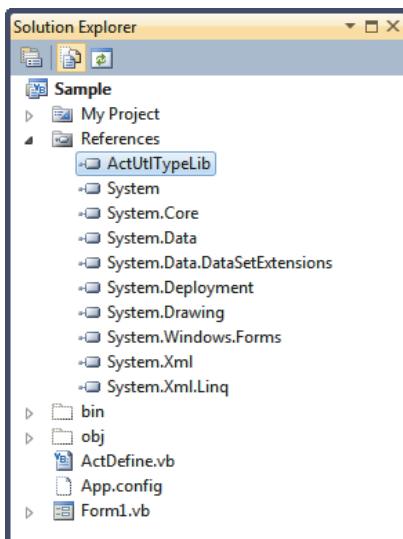
(Visual Studio® 2008 及 Visual Studio® 2010 的情况下，应在“Class View Settings(分类视图设置)”中将“Show Project References(工程浏览的显示)”设置为有效)



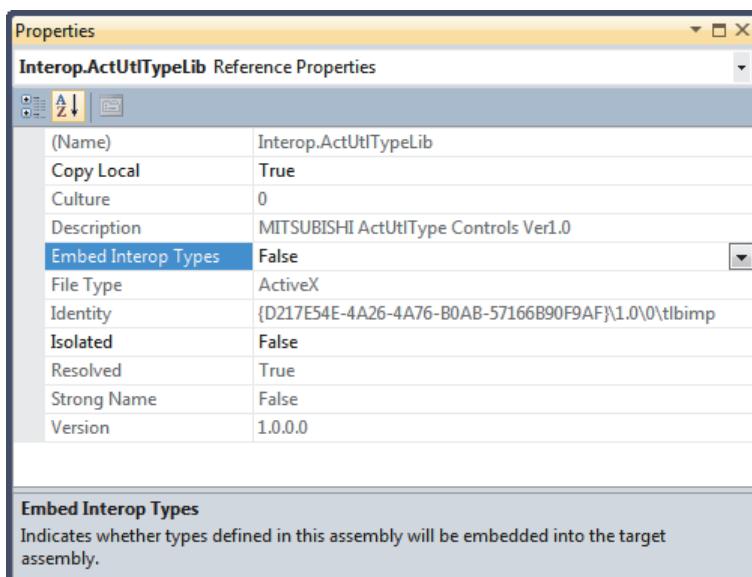
2.1 用于使用控件的设置
2.1.3 使用 Visual Studio® .NET 的情况下

6. Visual Studio® 2010/2012/2013 的 Visual Basic® .NET 的情况下，点击“Solution Explorer(解决方案资源管理器)”的  按钮，显示所有文件。

鼠标右击“References(浏览设置)”内使用的组件的库后，选择“Properties(属性)”。



将“Embed Interop Types(相互应用型的嵌入)”设置为“False”。



要 点

- 使用.NET Control的情况下
在“References(浏览设置)”中将添加下述库。
 - DotUtilType的情况下 : ActTypeLib
 - DotSupportMsg的情况下 : ActSupportMsgLib应在各库的属性中将“Embed Interop Types(相互应用型的嵌入)”设置为“False”。
- 创建使用了.NET Framework 4/.NET Framework 4.5 的应用程序的情况下
在包含有应用程序.exe文件的文件夹中，应添加记述了下述内容的应用程序配置文件(app.config)。

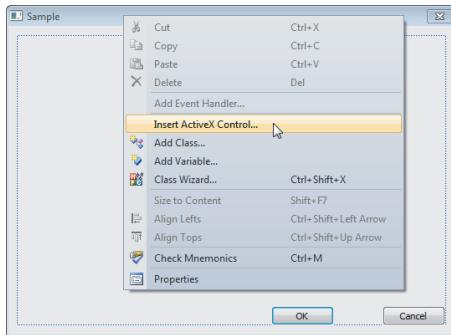
```
<configuration>
    <startup useLegacyV2RuntimeActivationPolicy="true">
        <supportedRuntime version="v4.0"/>
    </startup>
</configuration>
```

有关详细内容请参阅 Visual Studio® 的帮助。

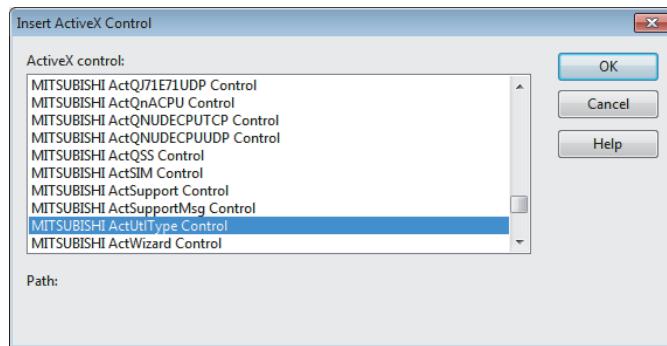
(6) 在 Visual C++®.NET(MFC) 中使用 ACT 控件的情况下

操作步骤

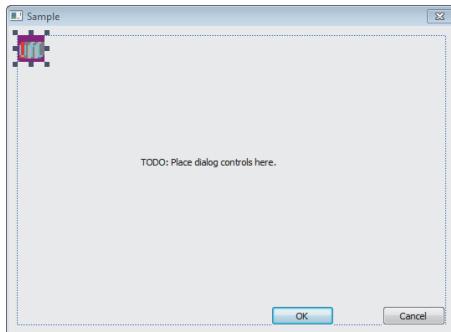
1. 鼠标右击窗体，选择“Insert ActiveX Control(ActiveX 控件的插入)”。



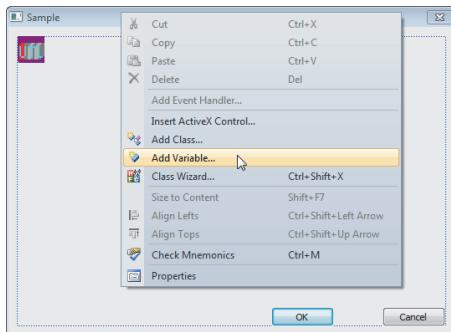
2. 选择使用的 ACT 控件，点击 **OK** 按钮。



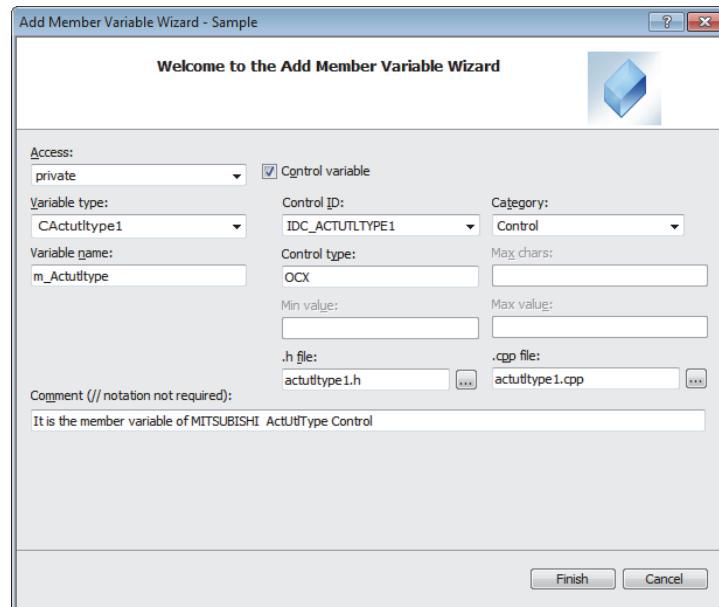
3. 选择的 ACT 控件将被粘贴到窗体中。



4. 鼠标右击窗体，选择“Add Variable(变量的添加)”。

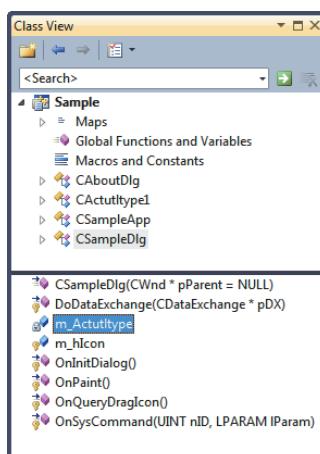


5. 勾选“Control variable(控件变量)”后，在“Control ID(控件 ID)”中选择 ACT 控件的 ID，输入变量名并点击 **Finish** (完成) 按钮。

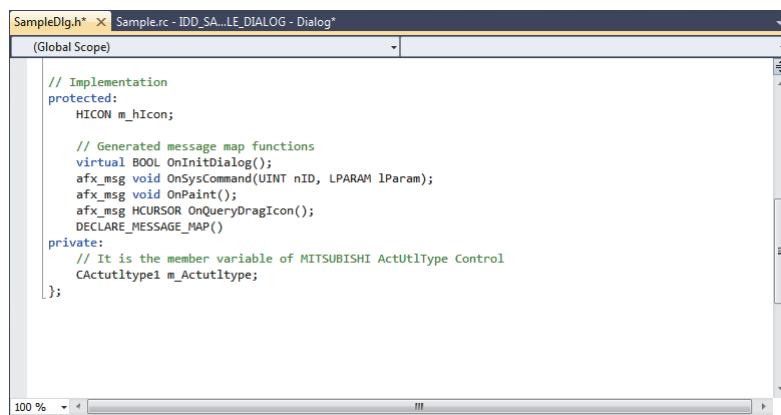


6. 确认 5. 中设置的成员变量已被创建。

● 分类视图的窗体分类内



● 窗体分类的头文件内



```
SampleDlg.h* Sample.rc - IDD_SA...LE_DIALOG - Dialog*
(Global Scope)

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()

private:
    // It is the member variable of MITSUBISHI ActUtType Control
    CActutltype1 m_Actutltype;
};
```

2.2 编程步骤

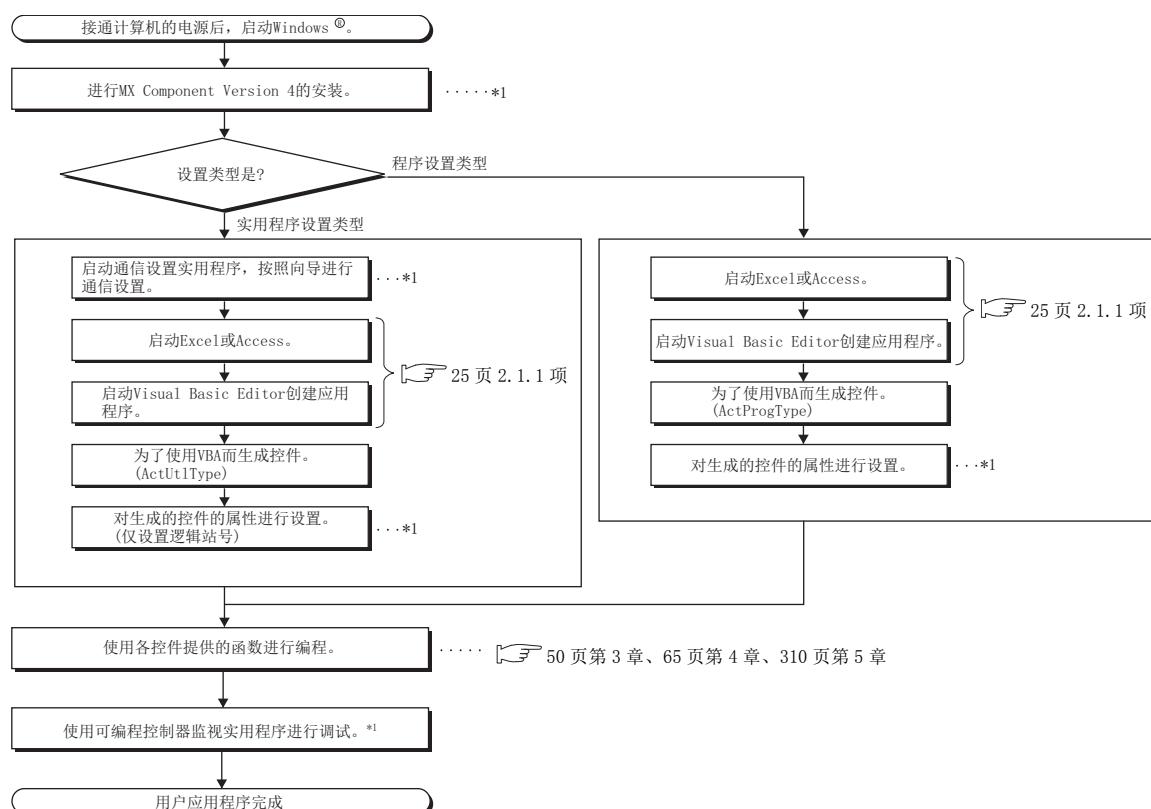
本节介绍创建用户应用程序的准备步骤有关内容。

要 点

创建用户程序的情况下，目标CPU应选择“x86”(32位)。

2.2.1 使用VBA的情况下

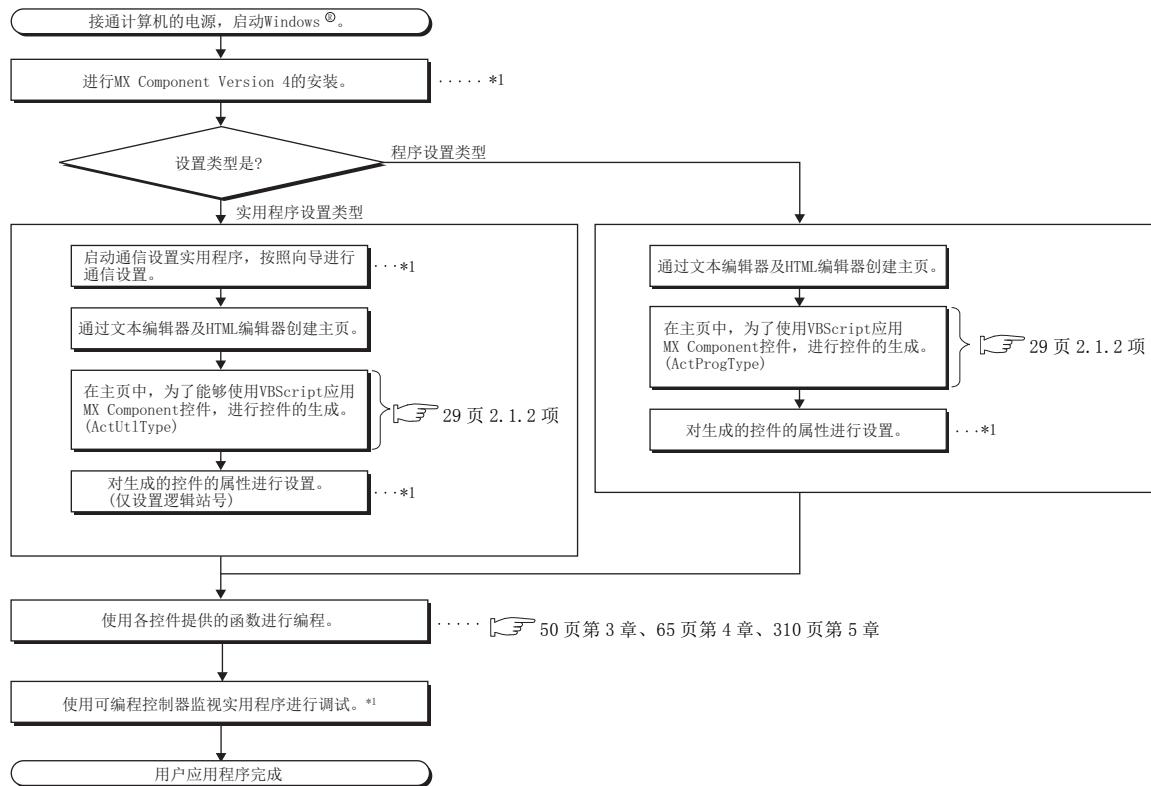
使用VBA创建用户应用程序的步骤如下所示。



*1: MX Component Version 4 操作手册

2.2.2 使用 VBScript 的情况下

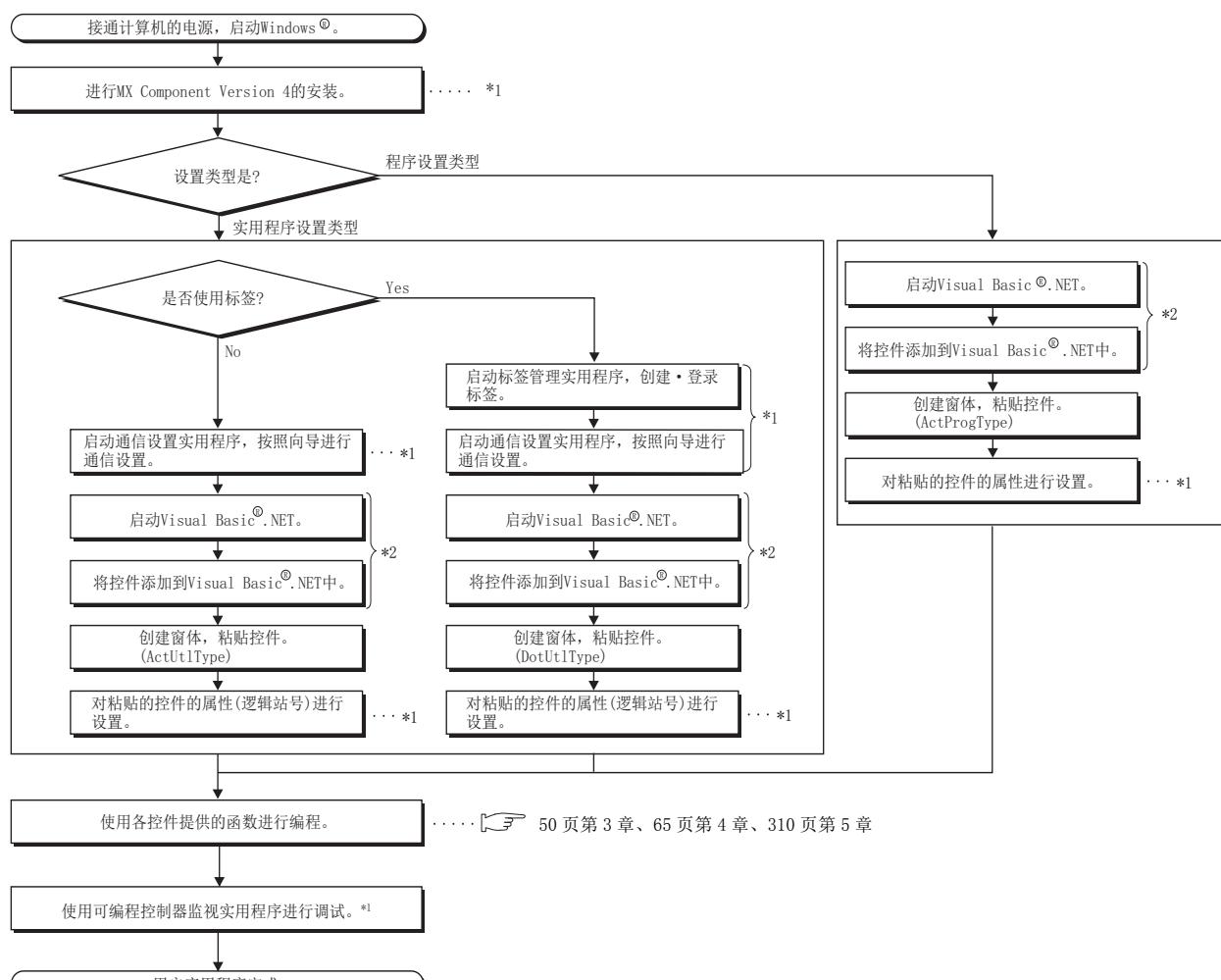
使用 VBScript 创建用户应用程序的步骤如下所示。



*1: MX Component Version 4 操作手册

2.2.3 使用 Visual Basic® .NET 的情况下

使用 Visual Basic® .NET 创建用户应用程序的步骤如下所示。

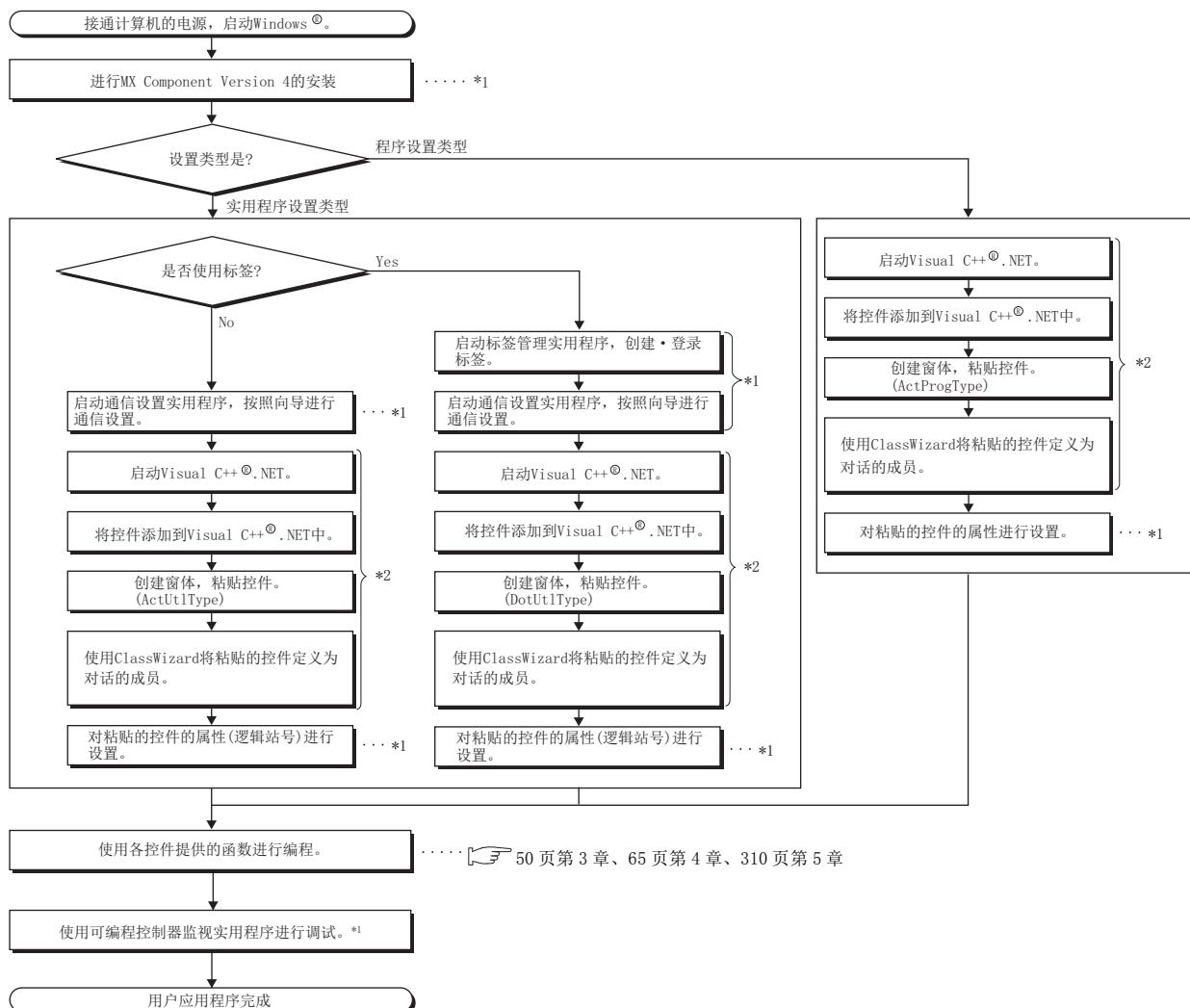


*1: MX Component Version 4 操作手册

*2: 30页 2.1.3项 (1)

2.2.4 使用 Visual C++® .NET 的情况下

使用 Visual C++® .NET 创建用户应用程序的步骤如下所示。



*1: MX Component Version 4 操作手册

*2: 30页 2.1.3项 (2)

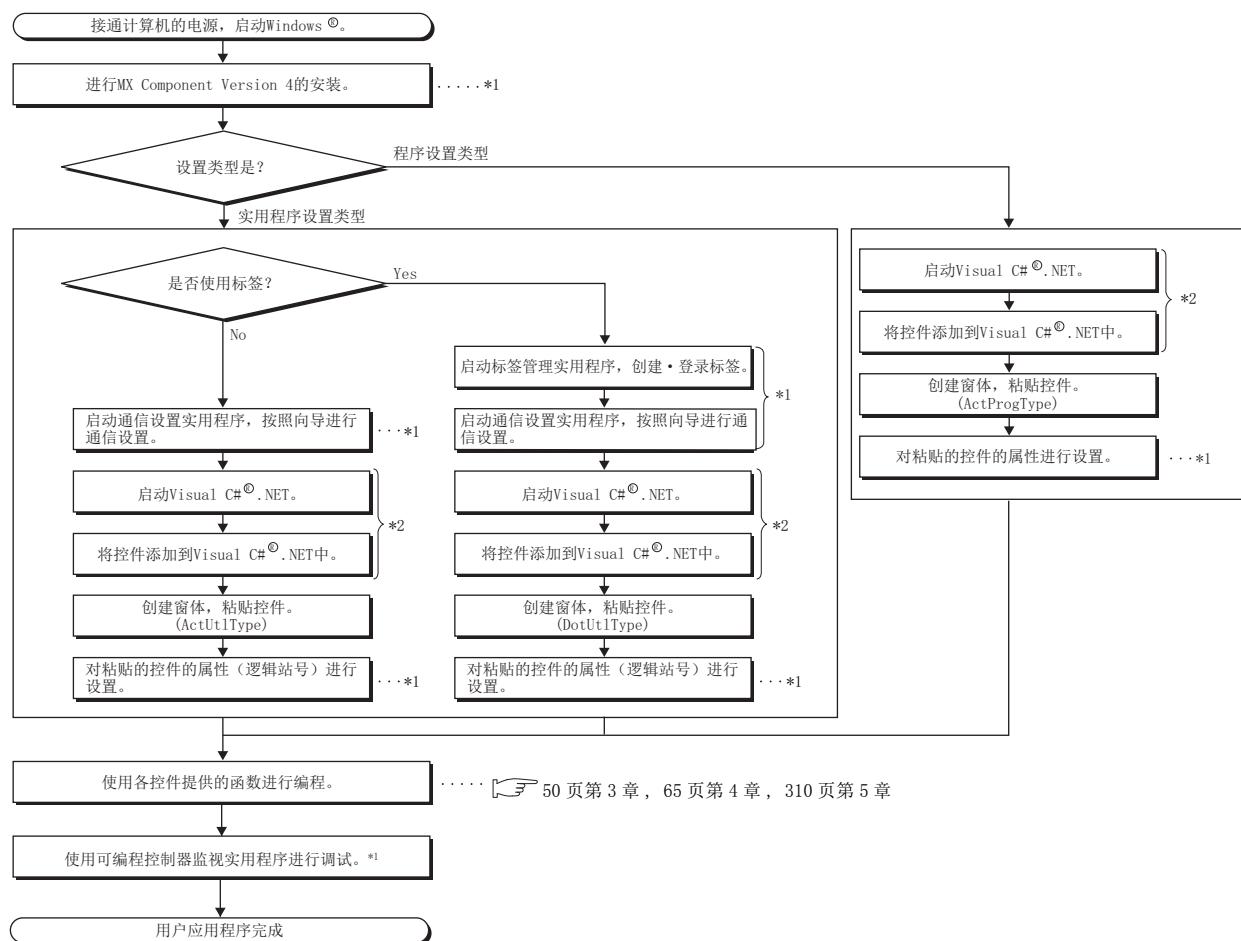
要 点

在一个程序中同时粘贴 .NET 控件 (DotUtlType) 和 Act 控件 (ActUtlType) 以使用的情况下，应通过解决方案资源管理器的 [Project (工程)] → [Property (属性)] 菜单实施以下操作。

- Visual Studio® 2005/2008 的情况下
 - 1) 选择 [Common Properties (通用属性)] → [References (参照设置)]。
 - 2) 从 “References (参照)” 的 “Name (名称)” 中选择 “ActUtlTypeLib”，将 “Build Properties (构建属性)” 的 “Use (在构建中使用)” 的设置更改为 “False”。
 - 3) 从 “References (参照)” 的 “Name (名称)” 中选择 “ActInterop.ActUtlTypeLib.1.0”，将 “Build Properties (构建属性)” 的 “Use (在构建中使用依存关系)” 的设置更改为 “False”。
 - 4) 点击 [OK (确定)]，反映设置的更改。
- Visual Studio® 2010/2012/2013 的情况下
 - 1) 选择 [Common Properties (通用属性)] → [Framework and references (Framework与参照)]。
 - 2) 从 “References (参照)” 的 “Name (名称)” 中选择 “ActUtlTypeLib”，将 “Build Properties (构建属性)” 的 “Reference Assembly Output (参照集的输出)” 的设置更改为 “False”。
 - 3) 从 “References (参照)” 的 “Name (名称)” 中选择 “Interop.ActUtlTypeLib.1.0”，将 “Build Properties (构建属性)” 的 “Reference Assembly Output (参照集的输出)” 的设置更改为 “False”。
 - 4) 点击 [OK (确定)]，反映设置的更改。

2.2.5 使用 Visual C#® .NET 的情况下

使用 Visual C#® .NET 创建用户应用程序的步骤如下所示。



*1: MX Component Version 4 操作手册

*2: 33页 2.1.3项 (3)

要点

在 Visual Studio® 2010/2012/2013 中，在一个程序中同时粘贴 .NET 控件 (DotUtlType) 和 Act 控件 (ActUtlType) 以使用的情况下，应通过解决方案资源管理器的“References (参照设置)”实施以下操作。

- 1) 选择并删除 ActUtlTypeLib。
- 2) 选择并删除 Interop.ActUtlTypeLib。

2.3 软元件类型

本节介绍函数中可指定的软元件有关内容。

要 点

- 对于下述函数中指定的软元件的指定方法，应通过“软元件名 + 软元件编号”进行指定。
关于软元件编号，应注意 8 进制数、10 进制数及 16 进制数的区别。
 - 对象函数：ReadDeviceBlock、ReadDeviceBlock2、WriteDeviceBlock、WriteDeviceBlock2、ReadDeviceRandom、ReadDeviceRandom2、WriteDeviceRandom、WriteDeviceRandom2、SetDevice、SetDevice2、GetDevice、GetDevice2
- 在 ReadDeviceBlock、ReadDeviceBlock2、WriteDeviceBlock 及 WriteDeviceBlock2 中指定位软元件的情况下，软元件编号应以 16 的倍数进行指定。
- 对于可编程控制器 CPU 的局部软元件及各程序的文件寄存器，不能指定程序名进行访问。
- 不支持本节中所示的软元件以外。请勿使用不支持的软元件。

(1) 通用（网管功能通信 / 变频器通信 / 机器人控制器通信时除外）

是除网管功能通信、变频器通信及机器人控制器通信以外的所有通信路径中通用的软元件类型。

软元件	软元件名	软元件类型	软元件编号
功能输入	FX	位软元件	10 进制数
功能输出	FY	位软元件	10 进制数
功能寄存器	FD	字软元件 ^{*1}	10 进制数
特殊继电器	SM	位软元件	10 进制数
特殊寄存器	SD	字软元件	10 进制数
输入继电器	X	位软元件	16 进制数 ^{*2}
输出继电器	Y	位软元件	16 进制数 ^{*2}
内部继电器	M	位软元件	10 进制数
锁存继电器	L	位软元件	10 进制数
报警器	F	位软元件	10 进制数
变址继电器	V	位软元件	10 进制数
链接继电器	B	位软元件	16 进制数
数据寄存器	D	字软元件	10 进制数
链接寄存器	W	字软元件	16 进制数
定时器 ^{*12}	触点	TS	位软元件
	线圈	TC	位软元件
	当前值	TN	字软元件
计数器 ^{*12}	触点	CS	位软元件
	线圈	CC	位软元件
	当前值 ^{*3}	CN	字软元件
累计定时器 ^{*12}	触点	STS/SS	位软元件
	线圈	STC/SC	位软元件
	当前值	STN/SN	字软元件
链接特殊继电器	SB	位软元件	16 进制数
链接特殊寄存器	SW	字软元件	16 进制数
步进继电器	S	位软元件	10 进制数

软元件	软元件名	软元件类型	软元件编号
累加器	A *4	字软元件	10 进制数
变址寄存器	Z *4	字软元件	10 进制数
	V *4	字软元件	10 进制数
文件寄存器	R *5	字软元件	10 进制数
	ZR	字软元件	10 进制数
扩展文件寄存器	ER*\R *7	字软元件	10 进制数
直接链接 *8	链接输入	J*\X *7	位软元件
	链接输出	J*\Y *7	位软元件
	链接继电器	J*\B *7	位软元件
	链接特殊继电器	J*\SB *7	位软元件
	链接寄存器	J*\W *7	字软元件
	链接特殊寄存器	J*\SW *7	字软元件
特殊直接缓冲存储器 *9、*10	U*\G** *7、*11	字软元件	16 进制数 / 10 进制数

*1: 1 点 4 字。批量操作的情况下：以 1 字单位进行连续操作。随机操作的情况下：仅读取起始的 1 字。

*2: FXCPU 的情况下为 8 进制数。

*3: FXCPU 的情况下 200 以上为 32 位数据。

*4: 经由 E71 的情况下，不能使用。

*5: 在除 FX3G(C)CPU、FX3U(C)CPU 以外的 FXCPU 中指定文件寄存器的情况下，应指定数据寄存器 (D)。只有 FX3G(C)CPU、FX3U(C)CPU 才可以指定扩展寄存器 (R)。

*6: 指定扩展文件寄存器时，在块编号部与文件寄存器部之间应记述 “\”。

指定了 R** 的情况下，块 No. 0 的 R 将成为对象。

指定了 ERO\R** 的情况下，将返回出错信息。

指定 ER**\R** 的情况下，不能进行扩展表示（间接指定、位数指定）。

*7: 进行直接指定时，在直接指定部与软元件指定部之间应记述 “\”。

8: 对于 J 指定网络编号。

9: 对于 U 指定特殊模块 I/O 编号 (16 进制数)，对于 G** 指定缓冲存储器地址 (10 进制数)。

(例：特殊模块 I/O 编号指定 200H，缓冲存储器地址指定 100 的情况下，将变为 “U20\G100”。)

*10: QCPU 的多 CPU 构成时，如果指定本机 QCPU 的共享存储器将发生出错。

此外，无论是本机还是其它机号，如果对共享存储器进行写入将发生出错。

*11: FXCPU 的情况下，仅可使用 FX3U(C)CPU。

*12: 使用系统标签 Ver. 2 的标签进行访问的情况下，需要对与触点 / 线圈 / 当前值各自的软元件对应的系统标签 Ver. 2 的标签进行定义。

(2) 访问目标为 RCPU 的情况下

访问目标为 RCPU 的情况下，可以使用下表的软元件。

软元件	软元件名	软元件类型	软元件编号
长定时器 *1	触点	LTS	位软元件
	线圈	LTC	位软元件
	当前值	LTN	双字软元件
长计数器 *1	触点	LCS	位软元件
	线圈	LCC	位软元件
	当前值	LCN	双字软元件
累积长定时器 *1	触点	LSS	位软元件
	线圈	LSC	位软元件
	当前值	LSN	双字软元件
长变址寄存器	LZ	双字软元件	10 进制数
模块用刷新软元件	RD	字软元件	10 进制数

*1: 使用系统标签 Ver. 2 的标签进行访问的情况下，需要对与触点 / 线圈 / 当前值各自的软元件对应的系统标签 Ver. 2 的标签进行定义。

(3) 访问目标为 R 运动 CPU 及 Q 运动 CPU 的情况下

访问目标为 R 运动 CPU 及 Q 运动 CPU 的情况下，可以使用下表的软元件。

软元件	软元件名	软元件类型	软元件编号
运动寄存器	#	字软元件	10 进制数

(4) CC-Link 通信时

CC-Link 通信中，自板访问时可以使用下表的软元件。

在其它通信路径中不能使用。

软元件	软元件名	软元件类型	软元件编号	备注
特殊继电器	SM	位软元件	10 进制数	自板的特殊继电器
特殊寄存器	SD	字软元件	10 进制数	自板的特殊寄存器
链接特殊继电器 (CC-Link 用)	SB	位软元件	16 进制数	自板的链接特殊继电器
链接特殊寄存器 (CC-Link 用)	SW	字软元件	16 进制数	自板的链接特殊寄存器
远程输入	X	位软元件	16 进制数	RX
远程输出	Y	位软元件	16 进制数	RY
链接寄存器	W	字软元件	16 进制数	-
远程寄存器 (CC-Link 用写入区域)	WW	字软元件	16 进制数	RWw
远程寄存器 (CC-Link 用读取区域)	WR	字软元件	16 进制数	RWr
缓冲存储器	ML	字软元件	16 进制数	本站 CC-Link 模块的缓冲存储器
随机访问缓冲	MC	字软元件	16 进制数	位于本站 CC-Link 模块的缓冲存储器中的随机访问缓冲
自动刷新缓冲	MF	位软元件	16 进制数	本站 CC-Link 模块的自动刷新缓冲

(5) 网关功能通信时

在网关功能通信中使用的软元件名的指定方法如下所示。

关于其它通信中使用的软元件指定方法，请参阅以下手册。

 MX Component Version 4 操作手册

软元件	软元件名	软元件类型	软元件编号
网关软元件 *1	EG	字软元件	10 进制数

*1： 对未分配可编程控制器 CPU 的软元件的网关软元件进行读取的情况下，读取的数据将变为 0。

(6) 变频器通信 / 机器人控制器通信时

关于在变频器通信 / 机器人控制器通信时使用的监视类型，请参阅以下手册。

 MX Component Version 4 操作手册

(7) 关于软元件扩展表示

软元件扩展表示的可否表如下所示。

在 ReadDeviceBlock 及 WriteDeviceBlock 中不能使用。

软元件扩展表示	对象 CPU					
	RCPU			R 运动 CPU		
位数指定 (例 : K4M0)	<input type="radio"/> *1				×	
位指定 (例 : D0.1)	<input type="radio"/> *2				○	
变址修饰 (例 : M100Z0)	<input type="radio"/> *3				×	

软元件扩展表示	对象 CPU						GOT	变频器	机器人控制器
	QCPU (Q 模式)	C 语言控 制器	LCPU	QS CPU	FX CPU	Q 运动 CPU			
位数指定 (例 : K4M0)*4	○	○ *6	○	○	○	×	×	×	×
位指定 (例 : D0.1)	○ *7	○	○ *7	○ *7	○ *7	×	○	×	×
变址修饰 (例 : M100Z0)*5	○	×	○	×	×	×	×	×	×

○ : 可以使用 × : 不能使用

*1: FX/FY、FD/SD、V、T/C/ST、LT/LC/LST、W/SW、G、Z、R/ZR、LZ 不能指定。

*2: Z、T/C/ST、LT/LC/LST、LZ 不能指定。

*3: FX/F、Z、LZ 不能指定。

*4: FX/FY、DX/DY、T/C/ST(触点、线圈)不能指定。

*5: FX/FY、DX/DY、T/C/ST(触点、线圈)、Z 及 S 不能指定。

*6: 可进行位数指定的软元件仅限软元件编号是 16 的倍数的位软元件。(直接链接软元件仅支持 K4、K8 的位数指定。)

*7: Z、V、T/C/ST(当前值)不能指定。

2.4 可访问范围

关于各通信时的可访问范围, 请参阅以下手册。

 MX Component Version 4 操作手册

第3章 控件的属性

本章介绍控件具有的属性一览以及各自属性的详细内容。

3.1 属性一览

各控件具有的属性如下所示。

控件名	属性名
ACT Control	ActUtilType
	ActMLUtilType
	ActATCommand ^{*1}
	ActATCommandPasswordCancelRetryTimes ^{*1}
	ActATCommandResponseWaitTime ^{*1}
	ActBaudRate
	ActCallbackCancelWaitTime ^{*1}
	ActCallbackDelayTime ^{*1}
	ActCallbackNumber ^{*1}
	ActCallbackReceptionWaitingTimeOut ^{*1}
	ActConnectionCDWaitTime ^{*1}
	ActConnectionModemReportWaitTime ^{*1}
	ActConnectUnitNumber
	ActConnectWay ^{*1}
	ActControl
	ActCpuTimeOut
	ActCpuType
	ActDataBits
	ActDestinationIONumber
	ActDestinationPortNumber
	ActDialNumber ^{*1}
	ActDidPropertyBit
	ActDisconnectionCDWaitTime ^{*1}
	ActDisconnectionDelayTime ^{*1}
	ActDsidPropertyBit
	ActHostAddress
	ActIntelligentPreferenceBit
	ActIONumber
	ActLineType ^{*1}
	ActMultiDropChannelNumber
	ActMxUnitSeries ^{*2}
	ActNetworkNumber
	ActOutsideLineNumber ^{*1}
	ActPacketType
	ActParity
	ActPassword

控件名	属性名
ACT Control	ActPasswordCancelResponseWaitTime* ¹
	ActPortNumber
	ActProtocolType
	ActSourceNetworkNumber
	ActSourceStationNumber
	ActStationNumber
	ActStopBits
	ActSumCheck
	ActTargetSimulator
	ActThroughNetworkType
	ActTimeOut
	ActTransmissionDelayTime* ¹
	ActUnitNumber
	ActUnitType
.NET Control	ActSupportMsg
	ActMLSupportMsg
	DotUtilType
.NET Control	ActLogicalStationNumber
	ActPassword
.NET Control	DotSupportMsg
	-

*1: 是调制解调器通信用属性。只能在 ActProgType 中使用。

*2: 属性值自动设置，无法更改。

3.2 属性的详细内容

创建用户应用程序时设置的属性的详细内容如下所示。

(1) 实用程序设置类型的控件的属性

Act(ML)UtlType 控件及 DotUtlType 控件的属性如下所示。

属性名(型)	内容	默认值
ActLogicalStationNumber (LONG)	是通过通信设置实用程序设置的逻辑站号。 (可设置范围：0 ~ 1023)	0 (0x00)
ActPassword (BSTR)	为了对可进行口令锁定的下述模块中设置的口令锁定进行解除，指定口令。*1、*2、*3 • R 系列 C24 • Q 系列 C24 • Q 系列 E71 • 以太网端口内置 QCPU 未使用可进行口令锁定的模块的情况下将被忽略。	空余

*1：未设置口令的情况下，无需进行 ActPassword 的设置。

*2：输入的字符超过口令的最多字符数时将被忽略。

*3：指定了除英文数字以外的字符的情况下，Open 时将发生字符代码转换出错 (0xF1000001)。

(2) 程序设置类型的控件的属性

Act(ML)ProgType 控件的属性如下所示。

属性名(型)	内容	默认值								
ActMxUnitSeries (LONG)	指定连接对象模块的系列。 属性值自动设置，无法更改。 <table border="1"><thead><tr><th>属性值</th><th>内容</th></tr></thead><tbody><tr><td>0 (0x00)</td><td>可编程控制器 / 运动控制器 / GOT 指定</td></tr><tr><td>1 (0x01)</td><td>变频器指定</td></tr><tr><td>2 (0x02)</td><td>机器人控制器指定</td></tr></tbody></table>	属性值	内容	0 (0x00)	可编程控制器 / 运动控制器 / GOT 指定	1 (0x01)	变频器指定	2 (0x02)	机器人控制器指定	0 (0x00)
属性值	内容									
0 (0x00)	可编程控制器 / 运动控制器 / GOT 指定									
1 (0x01)	变频器指定									
2 (0x02)	机器人控制器指定									
ActNetworkNumber (LONG)	指定 MELSECNET/H 中的网络编号。 (本站指定时指定“0”(0x00)) 指定通过 GX Simulator 进行的其它站访问中的网络编号。 多点连接(经由 Q 系列 C24、QJ61BT11) 时进行了指定的情况如下所示。 <table border="1"><thead><tr><th>ActIntelligentPreferenceBit 的属性值</th><th>内容</th></tr></thead><tbody><tr><td>0 (0x00)</td><td>本网络指定</td></tr><tr><td>1 (0x01)</td><td>多点目标的其它网络指定</td></tr></tbody></table>	ActIntelligentPreferenceBit 的属性值	内容	0 (0x00)	本网络指定	1 (0x01)	多点目标的其它网络指定	0 (0x00)		
ActIntelligentPreferenceBit 的属性值	内容									
0 (0x00)	本网络指定									
1 (0x01)	多点目标的其它网络指定									

属性名(型)	内容		默认值
ActStationNumber (LONG)	指定 MELSECNET/H、CC-Link 中的站号。 (本站指定时指定“255”(0xFF)) 指定通过 GX Simulator 进行的其它站访问的站号。 多点连接(经由 Q 系列 C24、QJ61BT11) 时指定的情况如下所示。		255 (0xFF)
	ActIntelligentPreferenceBit 的属性值	内容	
	0 (0x00)	本网络指定	
	1 (0x01)	多点目标的其它网络指定	
变频器通信时，指定连接的变频器的站号(0 ~ 31)。			
ActUnitNumber (LONG)	指定串行通信模块的模块编号及 Q 系列智能功能模块对象时的站号。 未将串行通信及 Q 系列智能功能模块作为对象的情况下无效。 指定多点链接时对象的串行通信模块的模块编号。		0 (0x00)
ActConnectUnit Number (LONG)	指定串行通信模块、QE71 及 Q 系列 E71 的模块编号。 多点链接时，指定请求源的串行通信模块的模块编号。但是，经由 CPU COM 通信的多点链接的情况下，无需请求源站的模块编号。(指定“0”(0x00)) 多点链接以外的情况下，指定“0”(0x00)。 QE71 及 Q 系列 E71 的情况下，指定中继目标站号。 (本网络内访问的情况下固定为“0”(0x00)) 经由 MELSECNET/10 访问其它网络的情况下，指定连接以太网模块的参数中设置的站号。		0 (0x00)
ActionNumber (LONG)	指定模块 I/O 编号。 多点链接及智能功能模块访问时，指定对象串行通信模块及智能功能模块的实际输入输出 No. (起始输入输出编号 ÷ 16)。 (多点链接时，指定经由站、请求源站的 I/O 编号) 经由本站 CPU 及网络访问其它站的情况下，指定“1023”(0x3FF)。		1023 (0x3FF)

属性名(型)	内容			默认值																																																																																																																																																									
ActCpuType (LONG)	<p>指定进行通信的对象 CPU。 在参数中指定下述 CPU 类型。</p> <table border="1"> <thead> <tr> <th colspan="3">属性值 (属性窗口输入值)</th><th rowspan="2">对象 CPU</th></tr> <tr> <th>CPU 类型</th><th>10 进制</th><th>16 进制</th></tr> </thead> <tbody> <tr><td>CPU_R04CPU</td><td>4097</td><td>0x1001</td><td>R04CPU</td></tr> <tr><td>CPU_R08CPU</td><td>4098</td><td>0x1002</td><td>R08CPU</td></tr> <tr><td>CPU_R16CPU</td><td>4099</td><td>0x1003</td><td>R16CPU</td></tr> <tr><td>CPU_R32CPU</td><td>4100</td><td>0x1004</td><td>R32CPU</td></tr> <tr><td>CPU_R120CPU</td><td>4101</td><td>0x1005</td><td>R120CPU</td></tr> <tr><td>CPU_R16MTCPU</td><td>4113</td><td>0x1011</td><td>R16MTCPU</td></tr> <tr><td>CPU_R32MTCPU</td><td>4114</td><td>0x1012</td><td>R32MTCPU</td></tr> <tr><td>CPU_Q00JCPU</td><td>48</td><td>0x30</td><td>Q00JCPU</td></tr> <tr><td>CPU_Q00UJCPU</td><td>128</td><td>0x80</td><td>Q00UJCPU</td></tr> <tr><td>CPU_Q00CPU</td><td>49</td><td>0x31</td><td>Q00CPU</td></tr> <tr><td>CPU_Q00UCPU</td><td>129</td><td>0x81</td><td>Q00UCPU</td></tr> <tr><td>CPU_Q01CPU</td><td>50</td><td>0x32</td><td>Q01CPU</td></tr> <tr><td>CPU_Q01UCPU</td><td>130</td><td>0x82</td><td>Q01UCPU</td></tr> <tr><td>CPU_Q02CPU</td><td>34</td><td>0x22</td><td>Q02(H)CPU</td></tr> <tr><td>CPU_Q02PHCPU</td><td>69</td><td>0x45</td><td>Q02PHCPU</td></tr> <tr><td>CPU_Q02UCPU</td><td>131</td><td>0x83</td><td>Q02UCPU</td></tr> <tr><td>CPU_Q03UDCPU</td><td>112</td><td>0x70</td><td>Q03UDCPU</td></tr> <tr><td>CPU_Q03UDECPU</td><td>144</td><td>0x90</td><td>Q03UDECPU</td></tr> <tr><td>CPU_Q03UDVCPU</td><td>209</td><td>0xD1</td><td>Q03UDVCPU</td></tr> <tr><td>CPU_Q04UDHCPU</td><td>113</td><td>0x71</td><td>Q04UDHCPU</td></tr> <tr><td>CPU_Q04UDEHCPU</td><td>145</td><td>0x91</td><td>Q04UDEHCPU</td></tr> <tr><td>CPU_Q04UDVCPU</td><td>210</td><td>0xD2</td><td>Q04UDVCPU</td></tr> <tr><td>CPU_Q06CPU</td><td>35</td><td>0x23</td><td>Q06HCPU</td></tr> <tr><td>CPU_Q06PHCPU</td><td>70</td><td>0x46</td><td>Q06PHCPU</td></tr> <tr><td>CPU_Q06UDHCPU</td><td>114</td><td>0x72</td><td>Q06UDHCPU</td></tr> <tr><td>CPU_Q06UDEHCPU</td><td>146</td><td>0x92</td><td>Q06UDEHCPU</td></tr> <tr><td>CPU_Q06UDVCPU</td><td>211</td><td>0xD3</td><td>Q06UDVCPU</td></tr> <tr><td>CPU_Q10UDHCPU</td><td>117</td><td>0x75</td><td>Q10UDHCPU</td></tr> <tr><td>CPU_Q10UDEHCPU</td><td>149</td><td>0x95</td><td>Q10UDEHCPU</td></tr> <tr><td>CPU_Q12CPU</td><td>36</td><td>0x24</td><td>Q12HCPU</td></tr> <tr><td>CPU_Q12PHCPU</td><td>65</td><td>0x41</td><td>Q12PHCPU</td></tr> <tr><td>CPU_Q12PRHCPU</td><td>67</td><td>0x43</td><td>Q12PRHCPU</td></tr> <tr><td>CPU_Q13UDHCPU</td><td>115</td><td>0x73</td><td>Q13UDHCPU</td></tr> <tr><td>CPU_Q13UDEHCPU</td><td>147</td><td>0x93</td><td>Q13UDEHCPU</td></tr> <tr><td>CPU_Q13UDVCPU</td><td>212</td><td>0xD4</td><td>Q13UDVCPU</td></tr> <tr><td>CPU_Q20UDHCPU</td><td>118</td><td>0x76</td><td>Q20UDHCPU</td></tr> <tr><td>CPU_Q20UDEHCPU</td><td>150</td><td>0x96</td><td>Q20UDEHCPU</td></tr> </tbody> </table>	属性值 (属性窗口输入值)			对象 CPU	CPU 类型	10 进制	16 进制	CPU_R04CPU	4097	0x1001	R04CPU	CPU_R08CPU	4098	0x1002	R08CPU	CPU_R16CPU	4099	0x1003	R16CPU	CPU_R32CPU	4100	0x1004	R32CPU	CPU_R120CPU	4101	0x1005	R120CPU	CPU_R16MTCPU	4113	0x1011	R16MTCPU	CPU_R32MTCPU	4114	0x1012	R32MTCPU	CPU_Q00JCPU	48	0x30	Q00JCPU	CPU_Q00UJCPU	128	0x80	Q00UJCPU	CPU_Q00CPU	49	0x31	Q00CPU	CPU_Q00UCPU	129	0x81	Q00UCPU	CPU_Q01CPU	50	0x32	Q01CPU	CPU_Q01UCPU	130	0x82	Q01UCPU	CPU_Q02CPU	34	0x22	Q02(H)CPU	CPU_Q02PHCPU	69	0x45	Q02PHCPU	CPU_Q02UCPU	131	0x83	Q02UCPU	CPU_Q03UDCPU	112	0x70	Q03UDCPU	CPU_Q03UDECPU	144	0x90	Q03UDECPU	CPU_Q03UDVCPU	209	0xD1	Q03UDVCPU	CPU_Q04UDHCPU	113	0x71	Q04UDHCPU	CPU_Q04UDEHCPU	145	0x91	Q04UDEHCPU	CPU_Q04UDVCPU	210	0xD2	Q04UDVCPU	CPU_Q06CPU	35	0x23	Q06HCPU	CPU_Q06PHCPU	70	0x46	Q06PHCPU	CPU_Q06UDHCPU	114	0x72	Q06UDHCPU	CPU_Q06UDEHCPU	146	0x92	Q06UDEHCPU	CPU_Q06UDVCPU	211	0xD3	Q06UDVCPU	CPU_Q10UDHCPU	117	0x75	Q10UDHCPU	CPU_Q10UDEHCPU	149	0x95	Q10UDEHCPU	CPU_Q12CPU	36	0x24	Q12HCPU	CPU_Q12PHCPU	65	0x41	Q12PHCPU	CPU_Q12PRHCPU	67	0x43	Q12PRHCPU	CPU_Q13UDHCPU	115	0x73	Q13UDHCPU	CPU_Q13UDEHCPU	147	0x93	Q13UDEHCPU	CPU_Q13UDVCPU	212	0xD4	Q13UDVCPU	CPU_Q20UDHCPU	118	0x76	Q20UDHCPU	CPU_Q20UDEHCPU	150	0x96	Q20UDEHCPU	34 (CPU_Q02CPU)
属性值 (属性窗口输入值)			对象 CPU																																																																																																																																																										
CPU 类型	10 进制	16 进制																																																																																																																																																											
CPU_R04CPU	4097	0x1001	R04CPU																																																																																																																																																										
CPU_R08CPU	4098	0x1002	R08CPU																																																																																																																																																										
CPU_R16CPU	4099	0x1003	R16CPU																																																																																																																																																										
CPU_R32CPU	4100	0x1004	R32CPU																																																																																																																																																										
CPU_R120CPU	4101	0x1005	R120CPU																																																																																																																																																										
CPU_R16MTCPU	4113	0x1011	R16MTCPU																																																																																																																																																										
CPU_R32MTCPU	4114	0x1012	R32MTCPU																																																																																																																																																										
CPU_Q00JCPU	48	0x30	Q00JCPU																																																																																																																																																										
CPU_Q00UJCPU	128	0x80	Q00UJCPU																																																																																																																																																										
CPU_Q00CPU	49	0x31	Q00CPU																																																																																																																																																										
CPU_Q00UCPU	129	0x81	Q00UCPU																																																																																																																																																										
CPU_Q01CPU	50	0x32	Q01CPU																																																																																																																																																										
CPU_Q01UCPU	130	0x82	Q01UCPU																																																																																																																																																										
CPU_Q02CPU	34	0x22	Q02(H)CPU																																																																																																																																																										
CPU_Q02PHCPU	69	0x45	Q02PHCPU																																																																																																																																																										
CPU_Q02UCPU	131	0x83	Q02UCPU																																																																																																																																																										
CPU_Q03UDCPU	112	0x70	Q03UDCPU																																																																																																																																																										
CPU_Q03UDECPU	144	0x90	Q03UDECPU																																																																																																																																																										
CPU_Q03UDVCPU	209	0xD1	Q03UDVCPU																																																																																																																																																										
CPU_Q04UDHCPU	113	0x71	Q04UDHCPU																																																																																																																																																										
CPU_Q04UDEHCPU	145	0x91	Q04UDEHCPU																																																																																																																																																										
CPU_Q04UDVCPU	210	0xD2	Q04UDVCPU																																																																																																																																																										
CPU_Q06CPU	35	0x23	Q06HCPU																																																																																																																																																										
CPU_Q06PHCPU	70	0x46	Q06PHCPU																																																																																																																																																										
CPU_Q06UDHCPU	114	0x72	Q06UDHCPU																																																																																																																																																										
CPU_Q06UDEHCPU	146	0x92	Q06UDEHCPU																																																																																																																																																										
CPU_Q06UDVCPU	211	0xD3	Q06UDVCPU																																																																																																																																																										
CPU_Q10UDHCPU	117	0x75	Q10UDHCPU																																																																																																																																																										
CPU_Q10UDEHCPU	149	0x95	Q10UDEHCPU																																																																																																																																																										
CPU_Q12CPU	36	0x24	Q12HCPU																																																																																																																																																										
CPU_Q12PHCPU	65	0x41	Q12PHCPU																																																																																																																																																										
CPU_Q12PRHCPU	67	0x43	Q12PRHCPU																																																																																																																																																										
CPU_Q13UDHCPU	115	0x73	Q13UDHCPU																																																																																																																																																										
CPU_Q13UDEHCPU	147	0x93	Q13UDEHCPU																																																																																																																																																										
CPU_Q13UDVCPU	212	0xD4	Q13UDVCPU																																																																																																																																																										
CPU_Q20UDHCPU	118	0x76	Q20UDHCPU																																																																																																																																																										
CPU_Q20UDEHCPU	150	0x96	Q20UDEHCPU																																																																																																																																																										

属性名(型)	内容			默认值
	属性值 (属性窗口输入值)		对象 CPU	34 (CPU_Q02CPU)
	CPU 类型	10 进制	16 进制	
ActCpuType (LONG)	CPU_Q25CPU	37	0x25	Q25HCPU
	CPU_Q25PHCPU	66	0x42	Q25PHCPU
	CPU_Q25PRHCPU	68	0x44	Q25PRHCPU
	CPU_Q26UDHCPU	116	0x74	Q26UDHCPU
	CPU_Q26UDEHCPU	148	0x94	Q26UDEHCPU
	CPU_Q26UDVCPU	213	0xD5	Q26UDVCPU
	CPU_Q50UDEHCPU	152	0x98	Q50UDEHCPU
	CPU_Q100UDEHCPU	154	0x9A	Q100UDEHCPU
	CPU_Q02CPU_A	321	0x141	Q02(H)CPU-A
	CPU_Q06CPU_A	322	0x142	Q06HCPU-A
	CPU_Q12DC_V	88	0x58	Q12DCCPU-V
	CPU_Q24DHC_V	89	0x59	Q24DHCCPU-V
	CPU_Q24DHC_LS	91	0x5B	Q24DHCCPU-LS
	CPU_QS001CPU	96	0x60	QS001CPU
	CPU_Q172CPU	1569	0x621	Q172CPU
	CPU_Q173CPU	1570	0x622	Q173CPU
	CPU_Q172HCPU	1569	0x621	Q172HCPU
	CPU_Q173HCPU	1570	0x622	Q173HCPU
	CPU_Q172DCPU	1573	0x625	Q172DCPU
	CPU_Q173DCPU	1574	0x626	Q173DCPU
	CPU_Q172DSCPU	1578	0x62A	Q172DSCPU
	CPU_Q173DSCPU	1579	0x62B	Q173DSCPU
	CPU_FX0CPU	513	0x201	FX0(S)CPU
	CPU_FX0NCPU	514	0x202	FX0NCPU
	CPU_FX1CPU	515	0x203	FX1CPU
	CPU_FX1SCPU	518	0x206	FX1sCPU
	CPU_FX1NCPU	519	0x207	FX1N(C)CPU
	CPU_FX2CPU	516	0x204	FXuCPU FX2cCPU
	CPU_FX2NCPU	517	0x205	FX2N(C)CPU
	CPU_FX3SCPU	522	0x20A	FX3sCPU
	CPU_FX3GCPU	521	0x209	FX3G(C)CPU
	CPU_FX3UCCPU	520	0x208	FX3U(C)CPU
	CPU_BOARD	1025	0x401	自板访问时 *1
	CPU_L02SCPU	163	0xA3	L02SCPU
	CPU_L02CPU	161	0xA1	L02CPU
	CPU_L06CPU	165	0xA5	L06CPU
	CPU_L26CPU	164	0xA4	L26CPU
	CPU_L26CPUBT	162	0xA2	L26CPU-BT
	INV_A800	7776	0x1E60	A800
	UCPU_CRD700	77825	0x13001	CRnD-7xx/CR75x-D

属性名(型)	内容	默认值																																															
ActPortNumber (LONG)	<p>指定计算机的连接端口编号。</p> <p>以太网模块连接时，将请求源(计算机)的端口编号设置为任意值。</p> <p>将端口编号指定为“=0”的情况下，站号↔IP关联信息设置方式将变为自动响应方式。 (选择除经由QE71自动响应方式以外时，设置固定值“5001”)</p> <p>此外，使用网卡时，将第1个板设置PORT_1，将第2个以后按PORT_2、PORT_3···的顺序进行指定。</p> <table border="1"> <thead> <tr> <th colspan="3">属性值(属性窗口输入值)</th> <th rowspan="2">内容</th> </tr> <tr> <th>端口编号</th> <th>10进制</th> <th>16进制</th> </tr> </thead> <tbody> <tr> <td>PORT_1</td> <td>1</td> <td>0x01</td> <td>通信端口1</td> </tr> <tr> <td>PORT_2</td> <td>2</td> <td>0x02</td> <td>通信端口2</td> </tr> <tr> <td>PORT_3</td> <td>3</td> <td>0x03</td> <td>通信端口3</td> </tr> <tr> <td>PORT_4</td> <td>4</td> <td>0x04</td> <td>通信端口4</td> </tr> <tr> <td>PORT_5</td> <td>5</td> <td>0x05</td> <td>通信端口5</td> </tr> <tr> <td>PORT_6</td> <td>6</td> <td>0x06</td> <td>通信端口6</td> </tr> <tr> <td>PORT_7</td> <td>7</td> <td>0x07</td> <td>通信端口7</td> </tr> <tr> <td>PORT_8</td> <td>8</td> <td>0x08</td> <td>通信端口8</td> </tr> <tr> <td>PORT_9</td> <td>9</td> <td>0x09</td> <td>通信端口9</td> </tr> <tr> <td>PORT_10</td> <td>10</td> <td>0x0A</td> <td>通信端口10</td> </tr> </tbody> </table>	属性值(属性窗口输入值)			内容	端口编号	10进制	16进制	PORT_1	1	0x01	通信端口1	PORT_2	2	0x02	通信端口2	PORT_3	3	0x03	通信端口3	PORT_4	4	0x04	通信端口4	PORT_5	5	0x05	通信端口5	PORT_6	6	0x06	通信端口6	PORT_7	7	0x07	通信端口7	PORT_8	8	0x08	通信端口8	PORT_9	9	0x09	通信端口9	PORT_10	10	0x0A	通信端口10	1(PORT_1)
属性值(属性窗口输入值)			内容																																														
端口编号	10进制	16进制																																															
PORT_1	1	0x01	通信端口1																																														
PORT_2	2	0x02	通信端口2																																														
PORT_3	3	0x03	通信端口3																																														
PORT_4	4	0x04	通信端口4																																														
PORT_5	5	0x05	通信端口5																																														
PORT_6	6	0x06	通信端口6																																														
PORT_7	7	0x07	通信端口7																																														
PORT_8	8	0x08	通信端口8																																														
PORT_9	9	0x09	通信端口9																																														
PORT_10	10	0x0A	通信端口10																																														
ActBaudRate (LONG)	<p>指定串行通信时的传送速度。</p> <table border="1"> <thead> <tr> <th>属性值(属性窗口输入值)</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>BAUDRATE_300 (300)</td> <td>300bps</td> </tr> <tr> <td>BAUDRATE_600 (600)</td> <td>600bps</td> </tr> <tr> <td>BAUDRATE_1200 (1200)</td> <td>1200bps</td> </tr> <tr> <td>BAUDRATE_2400 (2400)</td> <td>2400bps</td> </tr> <tr> <td>BAUDRATE_4800 (4800)</td> <td>4800bps</td> </tr> <tr> <td>BAUDRATE_9600 (9600)</td> <td>9600bps</td> </tr> <tr> <td>BAUDRATE_19200 (19200)</td> <td>19200bps</td> </tr> <tr> <td>BAUDRATE_38400 (38400)</td> <td>38400bps</td> </tr> <tr> <td>BAUDRATE_57600 (57600)</td> <td>57600bps</td> </tr> <tr> <td>BAUDRATE_115200 (115200)</td> <td>115200bps</td> </tr> </tbody> </table> <p>变频器通信时，指定BAUDRATE_4800(4800)以上。</p>	属性值(属性窗口输入值)	内容	BAUDRATE_300 (300)	300bps	BAUDRATE_600 (600)	600bps	BAUDRATE_1200 (1200)	1200bps	BAUDRATE_2400 (2400)	2400bps	BAUDRATE_4800 (4800)	4800bps	BAUDRATE_9600 (9600)	9600bps	BAUDRATE_19200 (19200)	19200bps	BAUDRATE_38400 (38400)	38400bps	BAUDRATE_57600 (57600)	57600bps	BAUDRATE_115200 (115200)	115200bps	19200 (BAUDRATE_19200)																									
属性值(属性窗口输入值)	内容																																																
BAUDRATE_300 (300)	300bps																																																
BAUDRATE_600 (600)	600bps																																																
BAUDRATE_1200 (1200)	1200bps																																																
BAUDRATE_2400 (2400)	2400bps																																																
BAUDRATE_4800 (4800)	4800bps																																																
BAUDRATE_9600 (9600)	9600bps																																																
BAUDRATE_19200 (19200)	19200bps																																																
BAUDRATE_38400 (38400)	38400bps																																																
BAUDRATE_57600 (57600)	57600bps																																																
BAUDRATE_115200 (115200)	115200bps																																																
ActDataBit (LONG)	<p>指定串行通信时发送及接收的字节数据的位数。</p> <table border="1"> <thead> <tr> <th>属性值(属性窗口输入值)</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>DATABIT_7 (7)</td> <td>7位</td> </tr> <tr> <td>DATABIT_8 (8)</td> <td>8位</td> </tr> </tbody> </table> <p>机器人控制器的情况下，指定字符大小。</p>	属性值(属性窗口输入值)	内容	DATABIT_7 (7)	7位	DATABIT_8 (8)	8位	8(DATABIT_8)																																									
属性值(属性窗口输入值)	内容																																																
DATABIT_7 (7)	7位																																																
DATABIT_8 (8)	8位																																																

属性名(型)	内容	默认值																						
ActParity (LONG)	<p>指定串行通信时使用的奇偶方式。</p> <table border="1"> <thead> <tr> <th>属性值(属性窗口输入值)</th><th>内容</th></tr> </thead> <tbody> <tr> <td>NO_PARITY(0)</td><td>无奇偶性</td></tr> <tr> <td>ODD_PARITY(1)</td><td>奇数</td></tr> <tr> <td>EVEN_PARITY(2)</td><td>偶数</td></tr> </tbody> </table>	属性值(属性窗口输入值)	内容	NO_PARITY(0)	无奇偶性	ODD_PARITY(1)	奇数	EVEN_PARITY(2)	偶数	1 (ODD_PARITY)														
属性值(属性窗口输入值)	内容																							
NO_PARITY(0)	无奇偶性																							
ODD_PARITY(1)	奇数																							
EVEN_PARITY(2)	偶数																							
ActStopBits (LONG)	<p>指定串行通信时使用的停止位的数。</p> <table border="1"> <thead> <tr> <th>属性值(属性窗口输入值)</th><th>内容</th></tr> </thead> <tbody> <tr> <td>STOPBIT_ONE(0)</td><td>1停止位</td></tr> <tr> <td>STOPBITS_TWO(2)</td><td>2停止位</td></tr> </tbody> </table> <p>机器人控制器的情况下，指定下列停止位的数。</p> <table border="1"> <thead> <tr> <th>属性值(属性窗口输入值)</th><th>内容</th></tr> </thead> <tbody> <tr> <td>ONESTOPBIT(0)</td><td>1停止位</td></tr> <tr> <td>ONE5STOPBITS(1)</td><td>1.5停止位</td></tr> <tr> <td>TWOSTOPBITS(2)</td><td>2停止位</td></tr> </tbody> </table>	属性值(属性窗口输入值)	内容	STOPBIT_ONE(0)	1停止位	STOPBITS_TWO(2)	2停止位	属性值(属性窗口输入值)	内容	ONESTOPBIT(0)	1停止位	ONE5STOPBITS(1)	1.5停止位	TWOSTOPBITS(2)	2停止位	3 0 (STOPBIT_ONE)								
属性值(属性窗口输入值)	内容																							
STOPBIT_ONE(0)	1停止位																							
STOPBITS_TWO(2)	2停止位																							
属性值(属性窗口输入值)	内容																							
ONESTOPBIT(0)	1停止位																							
ONE5STOPBITS(1)	1.5停止位																							
TWOSTOPBITS(2)	2停止位																							
ActControl (LONG)	<p>指定信号线的控制设置。</p> <table border="1"> <thead> <tr> <th rowspan="2">控制设置</th><th colspan="2">属性值(属性窗口输入值)</th><th rowspan="2">内容</th></tr> <tr> <th>10进制</th><th>16进制</th></tr> </thead> <tbody> <tr> <td>TRC_DTR</td><td>1</td><td>0x01</td><td>DTR控制</td></tr> <tr> <td>TRC_RTS</td><td>2</td><td>0x02</td><td>RTS控制</td></tr> <tr> <td>TRC_DRT_AND_RTS</td><td>7</td><td>0x07</td><td>DTR控制加上RTS控制</td></tr> <tr> <td>TRC_DTR_OR_RTS</td><td>8</td><td>0x08</td><td>DTR控制或RTS控制</td></tr> </tbody> </table>	控制设置	属性值(属性窗口输入值)		内容	10进制	16进制	TRC_DTR	1	0x01	DTR控制	TRC_RTS	2	0x02	RTS控制	TRC_DRT_AND_RTS	7	0x07	DTR控制加上RTS控制	TRC_DTR_OR_RTS	8	0x08	DTR控制或RTS控制	8 (TRC_DTR_OR_RTS)
控制设置	属性值(属性窗口输入值)		内容																					
	10进制	16进制																						
TRC_DTR	1	0x01	DTR控制																					
TRC_RTS	2	0x02	RTS控制																					
TRC_DRT_AND_RTS	7	0x07	DTR控制加上RTS控制																					
TRC_DTR_OR_RTS	8	0x08	DTR控制或RTS控制																					
ActHostAddress (BSTR)	是表示以太网通信时的连接主机名(IP地址)字符串的指针。	1.1.1.1																						
ActCpuTimeOut (LONG)	<p>指定以太网通信时的CPU监视定时器。(单位为“×250ms”)</p> <p>FXCPU/变频器的情况下，指定串行通信时的发送等待时间。(单位为“×10ms”)</p> <p>机器人控制器的情况下，指定发送时的超时时间。(1000～30000ms)</p>	0(0x00)																						
ActTimeOut (LONG)	<p>设置计算机与可编程控制器/变频器之间通信的超时值。(单位为“ms”)</p> <p>机器人控制器的情况下，指定接收时的超时时间。(5000～120000ms)</p> <p>在MX Component中，根据通信路径有时会通过内部处理进行超时处理。</p> <p>有关详细内容请参阅下述章节。</p> <p>534页附录3 关于超时时间</p>	10000																						
ActSumCheck (LONG)	<p>指定和校验的有无。</p> <p>仅串行通信模块有效。</p> <table border="1"> <thead> <tr> <th>属性值(属性窗口输入值)</th><th>内容</th></tr> </thead> <tbody> <tr> <td>NO_SUM_CHECK(0)</td><td>无和校验</td></tr> <tr> <td>SUM_CHECK(1)</td><td>有和校验</td></tr> </tbody> </table>	属性值(属性窗口输入值)	内容	NO_SUM_CHECK(0)	无和校验	SUM_CHECK(1)	有和校验	0 (NO_SUM_CHECK)																
属性值(属性窗口输入值)	内容																							
NO_SUM_CHECK(0)	无和校验																							
SUM_CHECK(1)	有和校验																							

属性名(型)	内容	默认值																								
ActSourceNetworkNumber (LONG)	指定 QE71 及 Q 系列 E71 指定时的请求源网络编号。 指定与连接 QE71 及 Q 系列 E71 相同的网络 No. (网络参数中指定的网络 No.)。	0 (0x00)																								
ActSourceStationNumber (LONG)	指定 QE71 及 Q 系列 E71 指定时的请求源站号 (计算机侧站号)。 设置时应避免与以太网同一环内设置的 QE71 的站号重复。	0 (0x00)																								
ActDestinationPortNumber (LONG)	<p>指定以太网通信指定时对象的端口编号。 访问其它网络时，指定中继目标端口编号。 除自动响应方式以外则按下表进行设置。</p> <table border="1"> <thead> <tr> <th colspan="2">通信</th> <th>设置</th> </tr> </thead> <tbody> <tr> <td rowspan="2">R 系列 E71 (TCP/IP)</td> <td>MELSOFT 连接</td> <td>固定为 “5002”</td> </tr> <tr> <td>OPS 连接</td> <td>根据网络参数</td> </tr> <tr> <td colspan="2">R 系列 E71 (UDP/IP)</td> <td>固定为 “5001”</td> </tr> <tr> <td colspan="2">QE71 (UDP/IP)</td> <td>固定为 “5001”</td> </tr> <tr> <td rowspan="3">Q 系列 E71 (TCP/IP)</td> <td>冗余 CPU 以外</td> <td>固定为 “5002”</td> </tr> <tr> <td>MELSOFT 连接 *2</td> <td>固定为 “5002”</td> </tr> <tr> <td>OPS 连接 *2</td> <td>根据网络参数</td> </tr> <tr> <td colspan="2">Q 系列 E71 (UDP/IP)</td> <td>固定为 “5001”</td> </tr> </tbody> </table>	通信		设置	R 系列 E71 (TCP/IP)	MELSOFT 连接	固定为 “5002”	OPS 连接	根据网络参数	R 系列 E71 (UDP/IP)		固定为 “5001”	QE71 (UDP/IP)		固定为 “5001”	Q 系列 E71 (TCP/IP)	冗余 CPU 以外	固定为 “5002”	MELSOFT 连接 *2	固定为 “5002”	OPS 连接 *2	根据网络参数	Q 系列 E71 (UDP/IP)		固定为 “5001”	0 (0x00)
通信		设置																								
R 系列 E71 (TCP/IP)	MELSOFT 连接	固定为 “5002”																								
	OPS 连接	根据网络参数																								
R 系列 E71 (UDP/IP)		固定为 “5001”																								
QE71 (UDP/IP)		固定为 “5001”																								
Q 系列 E71 (TCP/IP)	冗余 CPU 以外	固定为 “5002”																								
	MELSOFT 连接 *2	固定为 “5002”																								
	OPS 连接 *2	根据网络参数																								
Q 系列 E71 (UDP/IP)		固定为 “5001”																								
ActDestinationIONumber (LONG)	多点连接 (经由 R 系列 C24 • Q 系列 C24 • L 系列 C24 • CC-Link) 时，指定最终访问对象站的实际输入输出 No. (起始输入输出 ÷ 16)。 (对象为智能功能模块的情况下) 对象为 CPU 的情况下，指定 “1023” (0x3FF) 。	0 (0x00)																								
ActMultiDropChannelNumber (LONG)	多点连接 (经由 R 系列 C24 • Q 系列 C24 • L 系列 C24 • CC-Link) 时，指定多点连接通道 No. (CH1/CH2)。 机器人控制器的情况下，指定通信出错时的重试次数。 (0 ~ 10 次) 其它连接时无效。	0 (0x00)																								
ActThroughNetworkType (LONG)	<p>经由网络访问其它站的情况下，指定经由网络中是否包含 MELSECNET/10。</p> <table border="1"> <thead> <tr> <th>属性值</th> <th>经由网络的类型</th> </tr> </thead> <tbody> <tr> <td>0 (0x00)</td> <td>未包含 MELSECNET/10。</td> </tr> <tr> <td>1 (0x01)</td> <td>包含有 MELSECNET/10。</td> </tr> </tbody> </table> <p>机器人控制器的情况下，指定使用协议 (通信方法) 。 (有序 (0) / 无序 (1))</p>	属性值	经由网络的类型	0 (0x00)	未包含 MELSECNET/10。	1 (0x01)	包含有 MELSECNET/10。	0 (0x00)																		
属性值	经由网络的类型																									
0 (0x00)	未包含 MELSECNET/10。																									
1 (0x01)	包含有 MELSECNET/10。																									
ActIntelligentPreferenceBit (LONG)	<p>多点连接 (经由 R 系列 C24 • Q 系列 C24 • L 系列 C24 • CC-Link) 时，指定是否经由多点链接目标网络。 (由于区分本网络模块)</p> <table border="1"> <thead> <tr> <th>属性值</th> <th>经由网络的类型</th> </tr> </thead> <tbody> <tr> <td>0 (0x00)</td> <td>不访问多点目标其它网络。</td> </tr> <tr> <td>1 (0x01)</td> <td>访问多点目标其它网络。</td> </tr> </tbody> </table>	属性值	经由网络的类型	0 (0x00)	不访问多点目标其它网络。	1 (0x01)	访问多点目标其它网络。	0 (0x00)																		
属性值	经由网络的类型																									
0 (0x00)	不访问多点目标其它网络。																									
1 (0x01)	访问多点目标其它网络。																									

属性名(型)	内容	默认值																																							
ActDidPropertyBit (LONG)	<p>访问 Q 系列的本站智能功能模块(本站 CPU 中安装的智能功能模块)时, 通过将下述设置设置为无效可以无需指定“ActUnitNumber”。</p> <p>(仅在模块 I/O 编号“ActIONumber”中指定)</p> <table border="1"> <thead> <tr> <th>属性值</th><th>经由网络的类型</th></tr> </thead> <tbody> <tr> <td>0 (0x00)</td><td>将模块编号设置为有效。</td></tr> <tr> <td>1 (0x01)</td><td>将模块编号设置为无效。</td></tr> </tbody> </table>	属性值	经由网络的类型	0 (0x00)	将模块编号设置为有效。	1 (0x01)	将模块编号设置为无效。	1 (0x01)																																	
属性值	经由网络的类型																																								
0 (0x00)	将模块编号设置为有效。																																								
1 (0x01)	将模块编号设置为无效。																																								
ActDsdPropertyBit (LONG)	<p>多点连接(经由 R 系列 C24・Q 系列 C24・L 系列 C24・CC-Link)时, 通过将下述设置设置为无效可以无需指定“ActDestinationIONumber”。</p> <p>但是, 将下述设置设置为无效的情况下必须将“ActDidPropertyBit”设置为有效。</p> <p>(在“ActUnitNumber”中指定)</p> <table border="1"> <thead> <tr> <th>属性值</th><th>经由网络的类型</th></tr> </thead> <tbody> <tr> <td>0 (0x00)</td><td>将最终访问对象站的 I/O 编号设置为有效。</td></tr> <tr> <td>1 (0x01)</td><td>将最终访问对象站的 I/O 编号设置为无效。</td></tr> </tbody> </table>	属性值	经由网络的类型	0 (0x00)	将最终访问对象站的 I/O 编号设置为有效。	1 (0x01)	将最终访问对象站的 I/O 编号设置为无效。	1 (0x01)																																	
属性值	经由网络的类型																																								
0 (0x00)	将最终访问对象站的 I/O 编号设置为有效。																																								
1 (0x01)	将最终访问对象站的 I/O 编号设置为无效。																																								
ActPacketType (LONG)*3	<p>设置 CR/LF 的有无。</p> <table border="1"> <thead> <tr> <th>属性值 (属性窗口输入值)</th><th>内容</th></tr> </thead> <tbody> <tr> <td>CRLF_NONE (0)</td><td>CR/LF 无</td></tr> <tr> <td>CRLF_CR (1)</td><td>仅 CR 有</td></tr> <tr> <td>CRLF_CRLF (2)</td><td>CR/LF 有</td></tr> </tbody> </table>	属性值 (属性窗口输入值)	内容	CRLF_NONE (0)	CR/LF 无	CRLF_CR (1)	仅 CR 有	CRLF_CRLF (2)	CR/LF 有	1 (CRLF_CR)																															
属性值 (属性窗口输入值)	内容																																								
CRLF_NONE (0)	CR/LF 无																																								
CRLF_CR (1)	仅 CR 有																																								
CRLF_CRLF (2)	CR/LF 有																																								
ActPassword (BSTR)*4	<p>为了对可进行口令锁定的下述模块中设置的口令锁定进行解除, 指定口令。*5、*6、*7</p> <ul style="list-style-type: none"> • R 系列 C24 • Q 系列 C24 • Q 系列 E71 • 以太网端口内置 QCPU <p>未使用可进行口令锁定的模块的情况下将被忽略。</p>	空余																																							
ActConnectWay (LONG)*8	<p>设置连接方式。</p> <table border="1"> <thead> <tr> <th colspan="3">属性值(属性窗口输入值)</th><th rowspan="2">连接方式</th></tr> <tr> <th>连接方式</th><th>10 进制</th><th>16 进制</th></tr> </thead> <tbody> <tr> <td>TEL_AUTO_CONNECT</td><td>0</td><td>0x00</td><td>自动</td></tr> <tr> <td>TEL_AUTO_CALLBACK</td><td>1</td><td>0x01</td><td>自动(固定为回调时)</td></tr> <tr> <td>TEL_AUTO_CALLBACK_NUMBER</td><td>2</td><td>0x02</td><td>自动(指定回调编号时)</td></tr> <tr> <td>TEL_CALLBACK</td><td>3</td><td>0x03</td><td>回调连接(固定时)</td></tr> <tr> <td>TEL_CALLBACK_NUMBER</td><td>4</td><td>0x04</td><td>回调连接(指定编号时)</td></tr> <tr> <td>TEL_CALLBACK_REQUEST</td><td>5</td><td>0x05</td><td>回调请求(固定时)</td></tr> <tr> <td>TEL_CALLBACK_REQUEST_NUMBER</td><td>6</td><td>0x06</td><td>回调请求(指定编号时)</td></tr> <tr> <td>TEL_CALLBACK_WAIT</td><td>7</td><td>0x07</td><td>回调接收等待</td></tr> </tbody> </table>	属性值(属性窗口输入值)			连接方式	连接方式	10 进制	16 进制	TEL_AUTO_CONNECT	0	0x00	自动	TEL_AUTO_CALLBACK	1	0x01	自动(固定为回调时)	TEL_AUTO_CALLBACK_NUMBER	2	0x02	自动(指定回调编号时)	TEL_CALLBACK	3	0x03	回调连接(固定时)	TEL_CALLBACK_NUMBER	4	0x04	回调连接(指定编号时)	TEL_CALLBACK_REQUEST	5	0x05	回调请求(固定时)	TEL_CALLBACK_REQUEST_NUMBER	6	0x06	回调请求(指定编号时)	TEL_CALLBACK_WAIT	7	0x07	回调接收等待	0 (TEL_AUTO_CONNECT)
属性值(属性窗口输入值)			连接方式																																						
连接方式	10 进制	16 进制																																							
TEL_AUTO_CONNECT	0	0x00	自动																																						
TEL_AUTO_CALLBACK	1	0x01	自动(固定为回调时)																																						
TEL_AUTO_CALLBACK_NUMBER	2	0x02	自动(指定回调编号时)																																						
TEL_CALLBACK	3	0x03	回调连接(固定时)																																						
TEL_CALLBACK_NUMBER	4	0x04	回调连接(指定编号时)																																						
TEL_CALLBACK_REQUEST	5	0x05	回调请求(固定时)																																						
TEL_CALLBACK_REQUEST_NUMBER	6	0x06	回调请求(指定编号时)																																						
TEL_CALLBACK_WAIT	7	0x07	回调接收等待																																						
ActATCommand (BSTR)*8	<p>指定进行调制解调器初始化的 AT 指令。</p> <p>未进行任何设置(未输入属性值)的情况下, 使用调制解调器标准的 AT 指令。最多可设置 70 个字符。*9</p>	空余																																							

属性名(型)	内容	默认值																			
ActDialNumber (BSTR)*8	指定电话号码。最多可设置 50 个字符。 ^{*9} (可设置字符：0、1、2、3、4、5、6、7、8、9、-、*、#)	空余																			
ActOutsideLine Number (BSTR)*8	指定外线发送编号。最多可设置半角 10 个字符。 ^{*9} (可设置字符：0、1、2、3、4、5、6、7、8、9、-、*、#)	空余																			
ActCallbackNumber (BSTR)*8	指定回调用的电话号码。最多可设置 62 个字符。 ^{*9} (可设置字符：0、1、2、3、4、5、6、7、8、9、-、*、#) 对于连接方式 (ActConnectWay) 为自动 (指定回调编号时)、回调连接 (指定编号时) 及回调请求 (指定编号时) 以外的连接方式，无需进行设置。	空余																			
ActLineType (LONG)*8	识别线路类型。 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">属性值(属性窗口输入值)</th> <th rowspan="2">内容</th> </tr> <tr> <th>线路类型</th> <th>10 进制</th> <th>16 进制</th> </tr> </thead> <tbody> <tr> <td>LINETYPE_PULSE</td> <td>0</td> <td>0x00</td> <td>脉冲(拨号线路)</td> </tr> <tr> <td>LINETYPE_TONE</td> <td>1</td> <td>0x01</td> <td>音频(按钮拨号线路)</td> </tr> <tr> <td>LINETYPE_ISDN</td> <td>2</td> <td>0x02</td> <td>ISDN(ISDN 线路)</td> </tr> </tbody> </table>	属性值(属性窗口输入值)			内容	线路类型	10 进制	16 进制	LINETYPE_PULSE	0	0x00	脉冲(拨号线路)	LINETYPE_TONE	1	0x01	音频(按钮拨号线路)	LINETYPE_ISDN	2	0x02	ISDN(ISDN 线路)	1(LINETYPE_TONE)
属性值(属性窗口输入值)			内容																		
线路类型	10 进制	16 进制																			
LINETYPE_PULSE	0	0x00	脉冲(拨号线路)																		
LINETYPE_TONE	1	0x01	音频(按钮拨号线路)																		
LINETYPE_ISDN	2	0x02	ISDN(ISDN 线路)																		
ActConnectionCD WaitTime (LONG)*8	是线路连接 CD 信号确认时间。 设置线路连接时对 CD 信号线的 ON/OFF 进行监视的时间。 (设置范围：1 ~ 999，单位：s) 根据线路连接区域(例：海外)在设置时间内 CD 信号未变为 ON 的情况下延长设置时间。	90																			
ActConnectionModem ReportWaitTime (LONG)*8	是线路连接调制解调器等待时间。 设置线路连接后来自于调制解调器的结果代码响应的等待时间。 (设置范围：1 ~ 999，单位：s) 调制解调器的响应速度较慢的情况下延长设置时间。	5																			
ActDisconnectionCD WaitTime (LONG)*8	是线路断开 CD 信号确认时间。 设置线路的断开时对 CD 信号线的 ON/OFF 进行监视的时间。 (设置范围：1 ~ 999，单位：s) 根据线路连接区域(例：海外)在设置时间内 CD 信号未变为 OFF 的情况下，延长设置时间。	5																			
ActDisconnectionDelay Time (LONG)*8	是线路断开延迟时间。 设置对调制解调器发送的跳脱指令的保护时间(无通信时间)。 (设置范围：1 ~ 999，单位：s) 调制解调器的响应速度较慢的情况下延长设置时间。	3																			
ActTransmissionDelay Time (LONG)*8	是数据发送延迟时间。 设置发送 AT 指令之前的准备时间。 (设置范围：0 ~ 999，单位：s) 尽管设置了正确的 AT 指令仍然返回出错代码(0xF2100008)的情况下延长设置时间。 调制解调器的响应速度较慢的情况下延长设置时间。	0																			
ActATCommand ResponseWaitTime (LONG)*8	是 AT 指令发送响应等待时间。(设置范围：1 ~ 999，单位：s) 调制解调器的响应速度较慢的情况下延长设置时间。	1																			
ActPasswordCancel ResponseWaitTime (LONG)*8	是口令解除响应等待时间。(设置范围：1 ~ 999，单位：s) 对象的线路质量较差的情况下延长设置时间。	5																			
ActATCommand PasswordCancelRetry Times (LONG)*8	是 AT 指令 / 口令解除发送重试此数。 (设置范围：1 ~ 999，单位：次)	3																			

属性名(型)	内容	默认值																				
ActCallbackCancelWaitTime (LONG)*8	是回调线路断开等待时间。(设置范围：1 ~ 180，单位：s) 根据线路连接区域(例：海外)，对象(Q系列C24侧)线路在设置时间内未能断开的情况下延长设置时间。 连接方式(ActConnectWay)为回调连接及回调请求以外的情况下无需进行设置。	90																				
ActCallbackDelayTime (LONG)*8	是回调实施延迟时间。(设置范围：1 ~ 999，单位：s) 线路连接中继设备(例：调制解调器等)在线路断开后的再连接时需要隔开一定时间的情况下延长设置时间。 连接方式(ActConnectWay)为回调连接及回调请求以外的情况下，无需进行设置。	20																				
ActCallbackReceptionWaitingTimeOut (LONG)*8	回调接收等待用超时时间。(设置范围：1 ~ 3600，单位：s) 在回调接收等待的状态下，设置来自于Q系列C24的电话线路连接请求的等待时间。 经过设置时间时停止回调接收等待的状态，由于电话线路的连接未完成，因此Connect函数非正常结束。 连接方式(ActConnectWay)为回调接收等待以外的连接方式时，无需进行设置。	120																				
ActTargetSimulator (LONG)	<p>指定启动中的连接目标GX Simulator2。 连接FXCPU的情况下，指定“0”(0x00)。</p> <table border="1"> <thead> <tr> <th>属性值</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>0 (0x00)</td> <td>无指定 (启动中的模拟器为1个的情况下，连接到启动中的模拟器中。多个的情况下，以A、B、C、D的顺序查找启动中的模拟器并连接。)</td> </tr> <tr> <td>1 (0x01)</td> <td>模拟器A</td> </tr> <tr> <td>2 (0x02)</td> <td>模拟器B</td> </tr> <tr> <td>3 (0x03)</td> <td>模拟器C</td> </tr> <tr> <td>4 (0x04)</td> <td>模拟器D</td> </tr> </tbody> </table> <p>指定启动中的连接目标MT Simulator2。</p> <table border="1"> <thead> <tr> <th>属性值</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>2 (0x02)</td> <td>模拟器2号机</td> </tr> <tr> <td>3 (0x03)</td> <td>模拟器3号机</td> </tr> <tr> <td>4 (0x04)</td> <td>模拟器4号机</td> </tr> </tbody> </table>	属性值	内容	0 (0x00)	无指定 (启动中的模拟器为1个的情况下，连接到启动中的模拟器中。多个的情况下，以A、B、C、D的顺序查找启动中的模拟器并连接。)	1 (0x01)	模拟器A	2 (0x02)	模拟器B	3 (0x03)	模拟器C	4 (0x04)	模拟器D	属性值	内容	2 (0x02)	模拟器2号机	3 (0x03)	模拟器3号机	4 (0x04)	模拟器4号机	0 (0x00)
属性值	内容																					
0 (0x00)	无指定 (启动中的模拟器为1个的情况下，连接到启动中的模拟器中。多个的情况下，以A、B、C、D的顺序查找启动中的模拟器并连接。)																					
1 (0x01)	模拟器A																					
2 (0x02)	模拟器B																					
3 (0x03)	模拟器C																					
4 (0x04)	模拟器D																					
属性值	内容																					
2 (0x02)	模拟器2号机																					
3 (0x03)	模拟器3号机																					
4 (0x04)	模拟器4号机																					

属性名(型)	内容			默认值					
ActUnitType (LONG)	指定与物理端口连接的模块类型。								
	<table border="1"> <thead> <tr> <th>模块类型</th><th>连接方法</th><th>值</th></tr> </thead> </table>		模块类型	连接方法	值	R 系列 C24 模块直接连接 经由 GOT2000/1000 系列 R 系列 C24 连接	0x1000		
模块类型	连接方法	值							
<table border="1"> <tr> <td>UNIT_RJ71C24</td><td>R 系列 C24 模块直接连接</td><td rowspan="4">0x1000</td></tr> <tr> <td></td><td>经由 GOT2000/1000 系列</td></tr> <tr> <td></td><td>R 系列 C24 连接</td></tr> </table>			UNIT_RJ71C24	R 系列 C24 模块直接连接	0x1000		经由 GOT2000/1000 系列		R 系列 C24 连接
UNIT_RJ71C24	R 系列 C24 模块直接连接	0x1000							
	经由 GOT2000/1000 系列								
	R 系列 C24 连接								
<table border="1"> <tr> <td>UNIT_QJ71C24</td><td>Q 系列 C24 模块直接连接</td><td rowspan="4">0x19</td></tr> <tr> <td></td><td>经由 调制解调器 Q 系列 C24/CMO 连接</td></tr> <tr> <td></td><td>经由 GOT2000/1000 系列 Q 系列 C24 连接</td></tr> </table>			UNIT_QJ71C24	Q 系列 C24 模块直接连接	0x19		经由 调制解调器 Q 系列 C24/CMO 连接		经由 GOT2000/1000 系列 Q 系列 C24 连接
UNIT_QJ71C24	Q 系列 C24 模块直接连接	0x19							
	经由 调制解调器 Q 系列 C24/CMO 连接								
	经由 GOT2000/1000 系列 Q 系列 C24 连接								
<table border="1"> <tr> <td>UNIT_LJ71C24</td><td>L 系列 C24 模块直接连接</td><td rowspan="3">0x54</td></tr> <tr> <td></td><td>经由 调制解调器 L 系列 C24 连接</td></tr> <tr> <td></td><td>经由 GOT2000/1000 系列 L 系列 C24 连接</td></tr> </table>			UNIT_LJ71C24	L 系列 C24 模块直接连接	0x54		经由 调制解调器 L 系列 C24 连接		经由 GOT2000/1000 系列 L 系列 C24 连接
UNIT_LJ71C24	L 系列 C24 模块直接连接	0x54							
	经由 调制解调器 L 系列 C24 连接								
	经由 GOT2000/1000 系列 L 系列 C24 连接								
UNIT_FX485BD	FX 扩展端口 (RS-485) 直接连接	0x24							
UNIT_RJ71EN71	经由 R 系列 E71 连接 (IP 指定)	0x1001							
UNIT_RJ71EN71_DIRECT	经由 R 系列 E71 连接 (直接连接)	0x1005							
UNIT_QJ71E71	Q 系列 E71 以太网端口连接 / 直接连接	0x1A							
UNIT_RETHER	RCPU 以太网端口连接 (IP 指定)	0x1002	0x13 (UNIT_QNCPU)						
UNIT_RETHER_DIRECT	RCPU 以太网端口直接连接	0x1003							
UNIT_LNETHER	LCPU 以太网端口连接	0x52							
UNIT_LNETHER_DIRECT	LCPU 以太网端口直接连接	0x53							
UNIT_QNETHER	QCPU 以太网端口连接	0x2C							
UNIT_QNETHER_DIRECT	QCPU 以太网端口直接连接	0x2D							
UNIT_NZ2GF_ETB	以太网适配器模块 (NZ2GF-ETB) 连接	0x59							
UNIT_NZ2GF_ETB_DIRECT	以太网适配器模块 (NZ2GF-ETB) 直接连接	0x5A							
UNIT_FXETHER	经由 FXCPU 以太网适配器 (FX3U-ENET-ADP) 总线连接	0x4A							
UNIT_FXETHER_DIRECT	FXCPU 以太网适配器 (FX3U-ENET-ADP) 直接连接	0x4B							
UNIT_QNCPU	QCUP (Q 模式) -RS232C 端口直接连接 GOT2000/1000 系列 QCUP (Q 模式) 直接连接	0x13							
UNIT_LNCPU	LCPU -RS232C 端口直接连接 GOT2000/1000 系列 LCPU 直接连接	0x50							
UNIT_QNMOTION	Q 运动 CPU -RS232C 端口直接连接	0x1C							

属性名(型)	内容		默认值
	模块类型	连接方法	值
ActUnitType (LONG)	UNIT_FXCPU	FXCPU -RS422 端口直接连接	0x0F
		经由调制解调器 FXCPU 直接连接	
		FXCPU USB 直接连接	
		GOT2000/1000 系列 FXCPU 直接连接	
	UNIT_RUSB	RCPU USB 端口直接连接	0x1004
	UNIT_QNUSB	QCPU(Q 模式)USB 端口直接连接	0x16
	UNIT_LNUSB	LCPU USB 端口直接连接	0x51
	UNIT_QSUSB	QSCPU(安全可编程控制器)USB 端口直接连接	0x29
	UNIT_QNMOTIONUSB	Q 运动 CPU USB 端口直接连接	0x1D
	UNIT_MNETHBOARD	MELSECNET/H 板连接	0x1E
ActUnitType (LONG)	UNIT_MNETGBOARD	CC-Link IE 控制网卡连接	0x2B
	UNIT_CCIEFBORD	CC-Link IE 现场网卡连接	0x2F
	UNIT_CCLINKBOARD	CC-Link 板连接	0x0C
	UNIT_G4QNCPU	Q 系列 AJ65BT-G4-S3 模块直接连接	0x1B
	UNIT_SIMULATOR	模拟器 (GX Simulator) 连接	0x0B
	UNIT_SIMULATOR2* ¹⁰	模拟器 (GX Simulator2) 连接	0x30
		模拟器 (MT Simulator2) 连接	
	UNIT_A900GOT	GOT2000/1000/900 系列连接	0x21
	UNIT_GOT_RJ71EN71	经由 GOT2000/1000 系列 R 系列 E71 连接	0x1051
	UNIT_GOT_QJ71E71	经由 GOT2000/1000 系列 Q 系列 E71 连接	0x40
ActUnitType (LONG)	UNIT_GOT_RETHER	经由 GOT2000/1000 系列 RCPU 以太网端口连接	0x1052
	UNIT_GOT_QNETHER	经由 GOT2000/1000 系列 QCPU 以太网端口连接	0x41
	UNIT_GOT_LNETHER	经由 GOT2000/1000 系列 LCPU 以太网端口连接	0x55
	UNIT_GOT_NZ2GF_ETB	经由 GOT2000/1000 系列以太网适配器模块 (NZ2GF-ETB) 连接	0x5B
	UNIT_GOT_FXETHER	经由 GOT2000/1000 系列 FXCPU 以太网适配器 (FX3U-ENET-ADP) 连接	0x61
	UNIT_GOT_FXENET	经由 GOT2000/1000 系列 FXCPU 以太网模块 (FX3U-ENET(-L)) 连接	0x62
	UNIT_GOTETHER_RJ71C24	经由 GOT2000/1000 系列 RCPU 以太网端口 (RS422) 连接	0x1061

属性名(型)	内容		默认值
指定连接模块(板)的通信协议类型。			
	通信协议类型	连接方法	值
ActProtocolType (LONG)	UNIT_GOTETHER_QNCPU	经由 GOT2000/1000 系列 QCPU(Q 模式)以太网端口(RS422)连接	0x56
	UNIT_GOTETHER_LNCPU	经由 GOT2000/1000 系列 LCPU 以太网端口(RS232C)连接	0x57
	UNIT_GOTETHER_FXCPU	经由 GOT2000/1000 系列 FXCPU 以太网端口连接	0x60
	UNIT_GOTETHER_QBUS	经由 GOT2000/1000 系列 Q 系列总线以太网端口连接	0x58
	UNIT_QBF	总线连接	0x1F
	PROTOCOL_SERIAL	经由串行端口	0x04 (4)
	PROTOCOL_USB	经由 USB 端口	0x0D (13)
	PROTOCOL_TCPIP	经由 TCP/IP	0x05 (5)
	PROTOCOL_UDPIP	经由 UDP/IP	0x08 (8)
	PROTOCOL_MNETH	经由 MELSECNET/H 板	0x0F (15)
	PROTOCOL_MNETG	经由 CC-Link IE 控制网卡	0x14 (20)
	PROTOCOL_CCIEF	经由 CC-Link IE 现场网卡	0x15 (21)
	PROTOCOL_CCLINK	经由 CC-Link	0x07 (7)
	PROTOCOL_SERIALMODEM	经由串行端口 + 调制解调器	0x0E (14)
	PROTOCOL_TEL	经由 TEL	0x0A (10)
	PROTOCOL_QBF	经由 Q 系列总线	0x10 (16)
	PROTOCOL_USBGOT	经由 USB 端口 + GOT	0x13 (19)
	PROTOCOL_SHAREDMEMORY	经由共享存储器服务器(模拟器)	0x06 (6)
	COMM_RS232C ^{*11}	变频器 RS-232C 通信	0x00 (16)
	COMM_USB ^{*11}	变频器 USB 通信	0x01 (1)
	RC_PROTOCOL_SERIAL	机器人控制器 RS-232C 通信	0x01 (1)
	RC_PROTOCOL_USB	机器人控制器 USB 通信	0x04 (4)
	RC_PROTOCOL_TCPIP	机器人控制器以太网(TCP/IP)通信	0x02 (2)

*1: CPU 板、AF 板除外。

*2: 关于详细内容, 请参阅以下手册。

 Q 对应以太网接口模块用户手册(基本篇)

*3: 进行变频器通信的情况下可以使用。

*4: 进行以太网通信的情况下可以使用。

*5: 未设置口令的情况下, 无需进行 ActPassword 的设置。

*6: 超过口令的最大字符数后输入的字符将被无视。

*7: 指定了英文字母以外的字符的情况下, Open 时会发生字符代码转换出错(0xF1000001)。

*8: 进行调制解调器通信的情况下可以使用。在 ActMLProgType 控件中不能使用。

*9: 输入了超过设置字符数的字符的情况下, 超过设置范围的字符将被无视。

*10: CPU 类型为 Q 运动 CPU 的情况下, 连接至 MT Simulator2, 其他情况下连接至 GX Simulator2。

*11: ActMxUnitSeries 为(0x01)的情况下可以使用。

第4章 各通信路径中设置的属性

本章介绍可访问的通信路径的详细内容及设置的属性有关内容。

4.1 设置的属性的阅读方法

85页4.3节以后的页面阅读方法如下所示。

4

第4章 各通信路径中设置的属性

4.2 串行通信

4.2.1 连接站为Q系列C24的串行通信

(1) 连接站除Q系列C24以外存在经由模块的情况下

(a) 构成

(b) 属性的模式

连接站CPU (Q模式)	Q运动 CPU ^{*1}	经由网络	经由CPU (Q模式)				
			QCPU CPU-Y	Q12DC CPU-Y	LCPU	QSCPU	Q运动CPU
A	A	CC-IE Control	2)	2) ^{*2}	2) ^{*3}	2) ^{*2}	×
		CC-IE Field	2)	2) ^{*2}	2)	2)	×
		MELSOON/T网	2)	2)	2)	2)	2)
		以太网	2)	2)	2)	2)	2)
串行通信	2) ^{*4}	2)	2)	2)	2)		
CC-Link	4)	4)	4)	4)	4)	4) ^{*4}	

○：可以访问（○内为属性的模式）。×：不能访问

*1：不能经由Q运动CPU访问经由站。
*2：对于Q12DCCPU-Y、QSCPU及Q运动CPU，由于不支持CC-Link IE 现场网络，因此不能访问CC-Link IE 现场网络。
*3：由于LCPU不支持CC-Link IE 控制网络，因此不能访问CC-Link IE 控制网络。
*4：冗余CPU的情况下，不能访问位于主基板上的串行通信模块。
*5：Q12DC、FXCPU 基本上和两部分支持。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载属性无需进行设置。

属性	默认值	属性的模式			
		1)	2) ^{*1}	3)	4)
ActBaseRate	19200 (BAUDRATE_19200)	根据Q系列C24的设置			
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号			
ActControl	8 (TBC_DTR_OR_RTS)	根据使用的电缆			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型			

(1) 构成
是系统构成的示意图。

(2) 属性模式
表示所使用的控件的可访问范围及属性的模式。

22

(3) 属性一览

(a) 属性

记载了属性名。

(b) 默认值

- 记载了属性的默认值。

- 对于在程序内部更改属性时的默认值用“()”括号进行记载。

(c) 属性的模式

记载了进行通信设置时必要的属性的设置内容。属性的模式编号对应于“(2)属性的模式”。



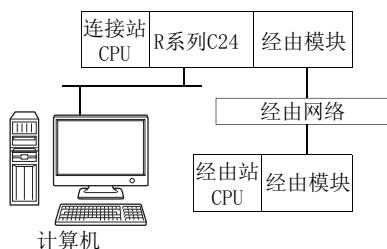
默认值记载的是Visual Basic[®].NET、Visual C++[®].NET或Visual C#[®].NET的属性窗口中显示的属性值。
在程序内部更改属性值时，对于必须以10进制数以外的值进行更改的属性的默认值，用括号进行记载。

4. 2 串行通信

4. 2. 1 连接站为 R 系列 C24 的串行通信

(1) 连接站侧除 R 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1) ^{*1}	CC IE Control	2) ^{*2}	×
		CC IE Field	×	×
		MELSECNET/H	×	×
		以太网	2)	2)
		串行通信	3)	3)
		CC-Link	4)	4)

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1) ^{*1}	CC IE Control	×	×	×	×	×	×
		CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： 多 CPU 构成时，仅 2 号机以后可以访问。

*2： 对于 RCPU，由于不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	与 R 系列 C24 的设置保持一致			
ActConnectUnitNumber	0(0x00)	连接站侧模块编号			
ActControl	8(TRC_DTR_OR_RTS)	根据使用电缆			
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActIntelligentPreferenceBit	0(0x00)	固定为 0(0x00)			
ActIONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber	0(0x00)	固定为 0(0x00)	对象站侧模块网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActParity	1(ODD_PARITY)	固定为 ODD_PARITY			
ActPortNumber	1(PORT_1)	计算机侧 COM 端口 No.			

属性	默认值	属性的模式			
		1)	2) ^{*1}	3)	4)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActHostAddress	1. 1. 1. 1	固定为 NULL			
ActCpuTimeOut	0(0x00)	固定为 0(0x00)			
ActDataBits	8(DATABIT_8)	固定为 8(0x08)			
ActStopBits	0(STOPBIT_ONE)	固定为 0(0x00)			
ActSum Check	0(NO_SUM_CHECK)	固定为 0(0x00)			
ActSourceNetworkNumber	0(0x00)	固定为 0(0x00)			
ActSourceStationNumber	0(0x00)	固定为 0(0x00)			
ActDestinationPortNumber	0(0x00)	固定为 0(0x00)			
ActStationNumber	255(0xFF)	固定为 255(0xFF)	对象站侧模块编号	固定为 255(0xFF)	固定为 255(0xFF)
ActThroughNetworkType	0(0x00)	不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71C24			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

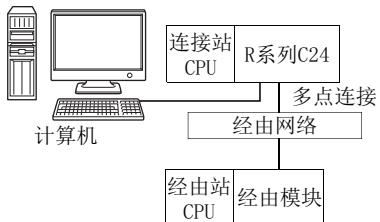
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 R 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



4

(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU				
RCPU			RCPU	R 运动 CPU			
独立模式 *1	1)	串行 通信	2)	×			
联动模式 *2	×		3)	×			
连接站 CPU		经由网络	经由站 CPU				
RCPU			QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU
独立模式 *1	1)	串行 通信	×	×	×	×	×
联动模式 *2	×		×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： 独立模式表示进行了如下参数设置的情况。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

*2： 联动模式表示进行了如下参数设置的情况。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 8
- CH2 侧：传送设置的动作设置 = 联动 (1)，通信协议设置 = 0

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式		
		1)	2)	3)
ActBaudRate	19200 (BAUDRATE_19200)	与 R 系列 C24 的设置保持一致		
ActConnect UnitNumber	0(0x00)	连接站侧模块编号		对象站侧模块编号
ActControl	8(TRC_DTR_OR_RTS)	根据使用电缆		
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型		
ActDestination IONumber *1	0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	固定为 0(0x00)
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 0(0x00)	固定为 1(0x01)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 0(0x00)	固定为 1(0x01)
ActIntelligent PreferenceBit	0(0x00)	固定为 0(0x00)		
ActIONumber *1	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActMultiDrop ChannelNumber *2	0(0x00)	固定为 0(0x00)	多点 通道 No.	固定为 0(0x00)
ActNetworkNumber	0(0x00)	固定为 0(0x00)		
ActParity	1(ODD_PARITY)	固定为 ODD_PARITY		
ActPortNumber	1(PORT_1)	计算机侧 COM 端口 No.		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL		

属性	默认值	属性的模式		
		1)	2)	3)
ActPacketType	0x01 (PACKET_PLCL)	PACKET_PLCL		
ActHostAddress	1. 1. 1. 1	固定为 NULL		
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)		
ActDataBits	8 (DATABIT_8)	固定为 8 (0x08)		
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)		
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)		
ActSourceNetwork Number	0 (0x00)	固定为 0 (0x00)		
ActSourceStation Number	0 (0x00)	固定为 0 (0x00)		
ActDestination PortNumber	0 (0x00)	固定为 0 (0x00)		
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)		
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000	以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块编号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71C24		

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

要 点

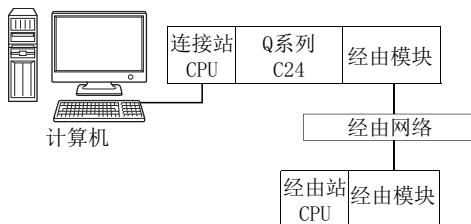
将连接站侧 R 系列 C24 设置为联动模式的情况下，必须将 R 系列 C24 参数的传送规格 S/W 开关设置的“和校验 (SW06)”设置为有 (ON)。

设置为无 (OFF) 的情况下，将发生通信出错而无法正常进行通信。

4.2.2 连接站为 Q 系列 C24 的串行通信

(1) 连接站侧除 Q 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU			经由网络	经由站 CPU	
QCPCU (Q 模式)	C 语言控 制器	Q 运动 CPU		RCPUCPU	R 运动 CPU
1)	1)*6	1)*1	CC IE Control	×	×
			CC IE Field	×	×
			MELSECNET/H	×	×
			以太网	×	×
			串行通信	×	×
			CC-Link	×	×

连接站 CPU			经由网络	经由站 CPU					
QCPCU (Q 模式)	C 语言控 制器	Q 运动 CPU		QCPCU (Q 模式)	C 语言 控制器	LCPUCPU	QSCPCU	Q 运动 CPU	FXCPU
1)	1)*6	1)*1	CC IE Control	2)	2)*2	2)*3	2)*2	2)*2	×
			CC IE Field	2)	2)	×	2)	2)	×
			MELSECNET/H	2)	2)	×	2)	2)	×
			以太网	2)	×	×	2)	2)	×
			串行通信	3)*4	×	3)	×	3)	×
			CC-Link	4)	4)	4)	×	4)	4)*5

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能经由 Q 运动 CPU 访问经由站。

*2：对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、QSCPCU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPUCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*5：仅 FX_{3G(C)} CPU、FX_{3U(C)} CPU 加上本网络才支持。

*6：多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载属性无需进行设置。

属性	默认值	属性的模式				
		1)	2)*1	3)	4)	
ActBaudRate	19200 (BAUDRATE_19200)	根据 Q 系列 C24 的设置				
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号				
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆				
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型				
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	
ActHostAddress	1.1.1.1	固定为 NULL				
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)				
ActIONumber*2	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧模块 I/O 地址		连接站侧模块 I/O 地址
ActMultiDropChannelNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActParity	1 (ODD_PARITY)	根据 Q 系列 C24 的设置			
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

*1: 经由以太网模块 (Q 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 Q 系列 E71 的参数设置中设置的值。
- 应设置 Q 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

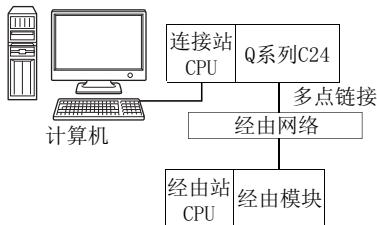
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 Q 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU			
QCPCU(Q 模式)、 C 语言控制器 *3	RCPUCPU		R 运动 CPU			
独立模式 *1 1)	串行通信	×	×			
联动模式 *1 ×		×	×			
连接站 CPU		经由网络	经由站 CPU			
QCPCU (Q 模式)	QCPCU (Q 模式)		LCPU	QSCPU	Q 运动 CPU	FXCPU
独立模式 *1 1)	串行通信	2)*2	×	2)	×	×
联动模式 *1 ×		3)*2	×	3)	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： 表示 CH2 侧的设置 (CH1 侧固定为独立模式)

*2： 冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*3： 多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式		
		1)	2)	3)
ActBaudRate	19200 (BAUDRATE_19200)	根据 Q 系列 C24 的设置		
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号		
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆		
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型		
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)		
ActIONumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)		
ActParity	1 (ODD_PARITY)	根据 Q 系列 C24 的设置		
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)		
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)		
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000	以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)		

*1：对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2：对多点链接的通道 No. 应进行下述指定。

0：使用模块的默认通道

1：通道 1

2：通道 2

要 点

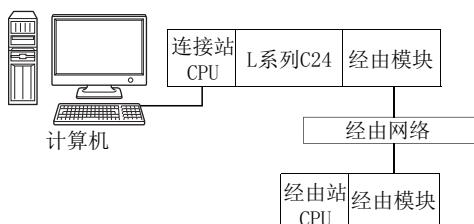
将连接站侧 Q 系列 C24 设置为联动模式的情况下，必须将 Q 系列 C24 参数的传送规格 S/W 开关设置的“和校验 (SW06)”设置为有 (ON)。

设置为无 (OFF) 的情况下，将发生通信出错而无法正常进行通信。

4.2.3 连接站为 L 系列 C24 的串行通信

(1) 连接站侧除 LJ71C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field*1	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*1	4)	4)*2	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)*3	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 LCPU 不支持 CC-Link IE 控制网络因此不能访问 CC-Link IE 控制网络。

*2：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	根据 L 系列 C24 的设置			
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号			
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU:992 (0x3E0) 2 号 CPU:993 (0x3E1) 3 号 CPU:994 (0x3E2) 4 号 CPU:995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU:992 (0x3E0) 2 号 CPU:993 (0x3E1) 3 号 CPU:994 (0x3E2) 4 号 CPU:995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU:992 (0x3E0) 2 号 CPU:993 (0x3E1) 3 号 CPU:994 (0x3E2) 4 号 CPU:995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActParity	1 (ODD_PARITY)	根据 L 系列 C24 的设置			
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetwork Type	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

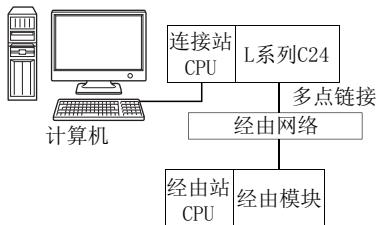
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 LJ71C24 与经由模块进行多点链接的情况下

(a) 构成



4

(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
LCPU			RCPU	R 运动 CPU				
独立模式	1)	串行通信	×	×				
联动模式	×		×	×				
连接站 CPU		经由网络	经由站 CPU					
LCPU			QCUP (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
独立模式	1)	串行通信	2)*1	×	2)	×	×	×
联动模式	×		3)*1	×	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式		
		1)	2)	3)
ActBaudRate	19200 (BAUDRATE_19200)		根据 L 系列 C24 的设置	
ActConnectUnitNumber	0 (0x00)		连接站侧模块站号	
ActControl	8 (TRC_DTR_OR_RTS)		根据使用的电缆	
ActCpuType	34 (CPU_Q02CPU)		对应于对象站的 CPU 类型	
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)

属性	默认值	属性的模式		
		1)	2)	3)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)		固定为 0 (0x00)	
ActIOPumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDropChannel1Number*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)		固定为 0 (0x00)	
ActParity	1 (ODD_PARITY)		根据 L 系列 C24 的设置	
ActPortNumber	1 (PORT_1)		计算机侧 COM 端口 No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)		PROTOCOL_SERIAL (0x04)	
ActStationNumber	255 (0xFF)		固定为 255 (0xFF)	
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)	
ActTimeOut	10000		以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_LJ71C24 (0x54)	

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

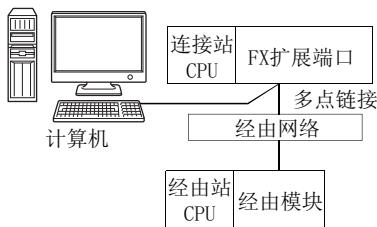
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.2.4 连接站为 FX 扩展端口的串行通信

(1) 构成



4

(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1) FXCPU	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) FXCPU	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	1)*1
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问。

*1： FX0CPU、FXosCPU、FX1CPU、FX2CPU、FX2cCPU 不能访问。

(3) 属性一览

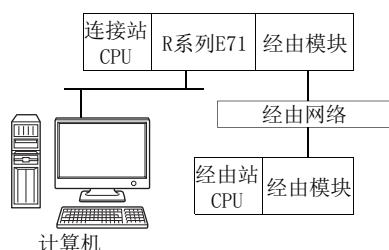
各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActBaudRate	19200 (BAUDRATE_19200)	根据 FX 扩展端口的设置
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆
ActCpuTimeOut	0 (0x00)	以 10ms 为单位由用户任意设置
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型
ActDataBits	8 (DATABIT_8)	根据 FX 扩展端口的设置
ActParity	1 (ODD_PARITY)	根据 FX 扩展端口的设置
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)
ActStopBits	0 (STOPBIT_ONE)	根据 FX 扩展端口的设置
ActSumCheck	0 (NO_SUM_CHECK)	根据 FX 扩展端口的设置
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitNumber	0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FX485BD (0x24)

4.3 以太网通信

4.3.1 连接站为R系列E71的以太网通信(TCP)

(1) 构成



4

(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R运动CPU		RCPU	R运动CPU
1)	1)*1,*3	CC IE Control	2)*2	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	2)
		串行通信	3)	3)
		CC-Link	4)	4)

连接站 CPU		经由网络	经由站 CPU					
RCPU	R运动CPU		QCPU (Q模式)	C语言 控制器	LCPU	QSCPU	Q运动CPU	FXCPU
1)	1)*1,*3	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过R运动CPU访问经由站。

*2：对于RCPU，由于不支持CC-Link IE控制网络，因此不能访问CC-Link IE控制网络。

*3：多CPU构成时，仅2号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2) ^{*1}	3)	4)
ActConnectUnitNumber	0(0x00)	固定为 0(0x00)	连接站侧模块编号	固定为 0(0x00)	固定为 0(0x00)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)
ActDestinationPortNumber	0(0x00)	MELSOFT 连接时 5002 OPS 连接时 任意的端口 No. *2			
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHostAddress	1. 1. 1. 1	连接站侧模块的主机名或 IP 地址			
ActIONumber *3	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber *4	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber *5	0(0x00)	对象站侧模块 网络 No.	对象站侧模块 网络 No.	连接站侧 R 系列 E71 网络 No.	连接站侧 R 系列 E71 网络 No.
ActPassword	空余	连接站侧的 R 系列 E71 中设置的口令			

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP(0x05)			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActPortNumber	1 (PORT_1)	计算机侧端口 No.			
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0(0x00)			
ActCpuTimeOut	0(0x00)	固定为 0(0x00)			
ActControl	8(TRC_DTR_OR_RTS)	固定为 0(0x00)			
ActDataBits	8(DATABIT_8)	固定为 0(0x00)			
ActParity	1(ODD_PARITY)	固定为 0(0x00)			
ActStopBits	0(STOPBIT_ONE)	固定为 0(0x00)			
ActSum Check	0(NO_SUM_CHECK)	固定为 0(0x00)			
ActIntelligentPreferenceBit	0(0x00)	固定为 0(0x00)			
ActSourceNetworkNumber	0(0x00)	计算机侧网络 No.			
ActSourceStationNumber *6	0(0x00)	计算机侧站号			
ActStationNumber *5	255(0xFF)	对象站侧模块编号	对象站侧模块编号	连接站侧 R 系列 E71 站号	连接站侧 R 系列 E71 站号
ActThroughNetworkType	0(0x00)	不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71EN71(0x1001)			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动生成方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 冗余 CPU 连接时使用 OPS 连接功能的情况下, 应指定网络参数中指定的任意端口 No.。

(设置范围为 1025 ≤ 端口 No. ≤ 4999 或 5003 ≤ 端口 No. ≤ 65534)

*3: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*4: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

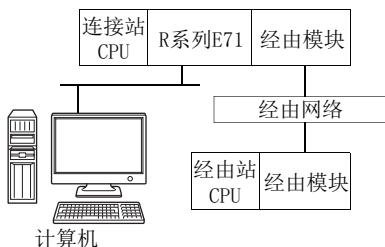
2: 通道 2

*5: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*6: 在指定计算机侧的站号时, 应避免与以太网的同一环内设置的 R 系列 E71 的站号重复。

4.3.2 连接站为 R 系列 E71 的以太网通信 (UDP)

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1)*1,*3	CC IE Control	2)*2	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	2)
		串行通信	3)	3)
		CC-Link	4)	4)

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1)*1,*3	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过 R 运动 CPU 访问经由站。

*2：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActConnectUnitNumber	0(0x00)	固定为 0(0x00)	连接站侧模块编号	固定为 0(0x00)	固定为 0(0x00)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)
ActDestinationPortNumber	0(0x00)	指定 IP 地址通信时：5001 未指定 IP 地址直接通信时：5003			
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHostAddress	1.1.1.1	指定 IP 地址通信时：连接站侧模块的主机名或 IP 地址 未指定 IP 地址直接通信时：255.255.255.255			
ActIONumber *3	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber *4	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber *5	0(0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站侧 R 系列 E71 网络 No.	连接站侧 R 系列 E71 网络 No.
ActPassword	空余	连接站侧的 R 系列 E71 中设置的口令			
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPPIP (0x08)			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0 (0x00)			
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	固定为 0 (0x00)			
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)			
ActParity	1 (ODD_PARITY)	固定为 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)			
ActSum Check	0 (NO_SUM_CHECK)	固定为 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	计算机侧网络 No.			
ActSourceStationNumber *6	0 (0x00)	计算机侧站号			
ActStationNumber *5	255 (0xFF)	对象站侧模块编号	对象站侧模块编号	连接站侧 R 系列 E71 站号	连接站侧 R 系列 E71 站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	指定 IP 地址通信时: UNIT_RJ71EN71 (0x1001) 未指定 IP 地址直接通信时: UNIT_RJ71EN71_DIRECT (0x1005)			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“站号 ↔ IP 关联信息”。此外, 进行设置时, 对“站号 ↔ IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 冗余 CPU 连接时使用 OPS 连接功能的情况下, 应指定网络参数中指定的任意端口 No.。

(设置范围为 1025 ≤ 端口 No. ≤ 4999 或 5003 ≤ 端口 No. ≤ 65534)

*3: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*4: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

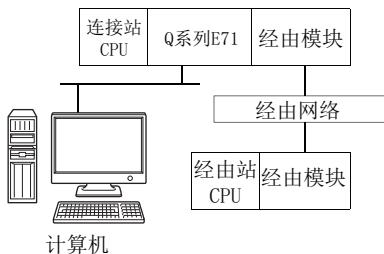
2: 通道 2

*5: 属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*6: 在指定计算机侧的站号时, 应避免与以太网的同一环内设置的 R 系列 E71 的站号重复。

4.3.3 连接站为Q系列E71的以太网通信(TCP)

(1) 构成



(2) 属性的模式

连接站 CPU				经由网络	经由站 CPU					
QCPU (Q 模式)	C 语言控 制器	QSCPU	Q 运动 CPU		RCPU	R 运动 CPU				
1)	1)*6	1)*2	1)*1,*2	CC IE Control	×	×				
				CC IE Field						
				MELSECNET/H	×	×				
				以太网	×	×				
				串行通信	×	×				
连接站 CPU				经由网络	经由站 CPU					
QCPU (Q 模式)	C 语言控 制器	QSCPU	Q 运动 CPU		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1)*6	1)*2	1)*1,*2	CC IE Control	2)	2)*3	2)*4	2)*2、*3	2)*3	×
				CC IE Field						
				MELSECNET/H	2)	2)	×	2)*2	2)	×
				以太网	2)	×	×	2)*2	2)	×
				串行通信	3)*5	×	3)	×	3)	×
连接站 CPU				经由网络	CC-Link	4)	4)	4)	4)	4)
QCPU (Q 模式)	C 语言控 制器	QSCPU	Q 运动 CPU							

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: Q172CPU、Q173CPU、Q172HCPU 及 Q173HCPU 的情况下，不能进行访问。

*2: 不能经由 QSCPU、Q 运动 CPU 访问经由站。

*3: 对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5: 元余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*6: 多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnitNumber	0 (0x00)	固定为 0 (0x00)	连接站侧模块站号	固定为 0 (0x00)	固定为 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	MELSOFT 连接时 5002 OPS 连接时 任意的端口 No.*1			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1. 1. 1. 1	连接站侧模块的主机名或 IP 地址			
ActIOPortNumber*2	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*4	0 (0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站侧 Q 系列 E71 网络 No.	连接站侧 Q 系列 E71 网络 No.
ActPassword	空余	连接站侧的 Q 系列 E71 中设置的口令			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActSource NetworkNumber	0 (0x00)	计算机侧网络 No.			
ActSourceStatio nNumber ^{*5}	0 (0x00)	计算机侧站号			
ActStation Number ^{*4}	255 (0xFF)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站侧 Q 系列 E71 站号	连接站侧 Q 系列 E71 站号
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71E71 (0x1A)			

*1: 冗余 CPU 连接时使用 OPS 连接功能的情况下，应指定网络参数中指定的任意端口 No.。（设置范围为 1025 ≤ 端口 No. ≤ 4999 或 5003 ≤ 端口 No. ≤ 65534）

*2: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

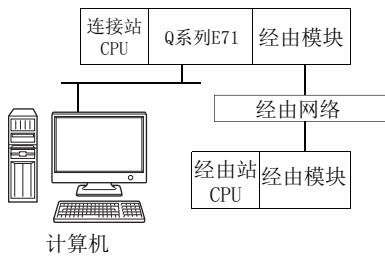
2: 通道 2

*4: 属性的模式为 1) 或 2) 的情况下，对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*5: 在指定计算机侧的站号时，应避免与以太网的同一环内设置的 Q 系列 E71 的站号重复。

4.3.4 连接站为 Q 系列 E71 的以太网通信 (UDP)

(1) 构成



(2) 属性的模式

连接站 CPU				经由网络	经由站 CPU	
QCPU (Q 模式)	C 语言控 制器	QSCPU	Q 运动 CPU		RCPU	R 运动 CPU
1)	1)*6, *7	1)*2	1)*1, *2	CC IE Control	×	×
				CC IE Field		
				MELSECNET/H	×	×
				以太网	×	×
				串行通信	×	×
				CC-Link	×	×

连接站 CPU				经由网络	经由站 CPU					
QCPU (Q 模式)	C 语言控 制器	QSCPU	Q 运动 CPU		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1)*6, *7	1)*2	1)*1, *2	CC IE Control	2)	2)*3	2)*4	2)*2、 *3	2)*3	×
				CC IE Field						
				MELSECNET/H	2)	2)	×	2)*2	2)	×
				以太网	2)	×	×	2)*2	2)	×
				串行通信	3)*5	×	3)	×	3)	×
				CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: Q172CPU、Q173CPU、Q172HCPU 及 Q173HCPU 的情况下无法访问。

*2: 不能经由 QSCPU、Q 运动 CPU 访问经由站。

*3: 对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5: 冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*6: 多 CPU 构成时，仅 2 号机以后可以访问。

*7: C 语言控制器不支持 MELSOFT 直接连接，因此，以太网端口直接连接时，不能访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnitNumber	0 (0x00)	固定为 0 (0x00)	连接站侧模块站号	固定为 0 (0x00)	固定为 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIONumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*3	0 (0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站 Q 系列 E71 网络 No.	连接站 Q 系列 E71 网络 No.
ActPassword	空余	连接站侧的 Q 系列 E71 中设置的口令			
ActPortNumber*4	1 (PORT_1)	计算机侧端口编号			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_UDPPIP (0x08)			
ActSource NetworkNumber ^{*4}	0 (0x00)	计算机侧网络 No.			
ActSourceStation Number ^{*5}	0 (0x00)	计算机侧站号			
ActStation Number ^{*3}	255 (0xFF)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站 Q 系列 E71 网络 No.	连接站 Q 系列 E71 网络 No.
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71E71 (0x1A)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

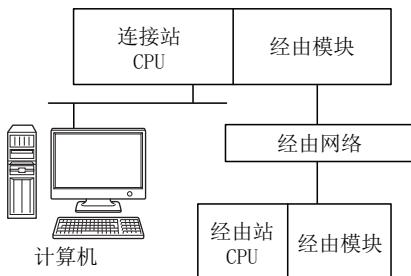
*3: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*4: 请勿使用 ActPortNumber1 ~ 1024。

*5: 在指定计算机侧的站号时, 应避免与以太网的同一环内设置的 Q 系列 E71 的站号重复。

4.3.5 连接站为 RCPU 的以太网通信 (TCP)

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1) ^{*1,*3}	CC IE Control	2) ^{*2}	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	2)
		串行通信	3)	3)
		CC-Link	4)	4)

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1) ^{*1,*3}	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 不能通过 R 运动 CPU 访问经由站。

*2: 对于 RCPU，由于不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3: 多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
Act Destination IONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)
ActDestination PortNumber	0(0x00)	5007			
ActDid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHost Address	1. 1. 1. 1	连接站侧模块的主机名或 IP 地址			
Act Intelligent Preference Bit	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU: 1(0x01) 除上述以外 : 0(0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU: 1(0x01) 除上述以外 : 0(0x00)
Act IONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActMultiDrop					
Channel Number *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetwork Number *4	0(0x00)	固定为 0(0x00)	对象站侧模块 网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActPassword	空余		连接站中设置的远程口令		
ActProtocol Type	0x04 (PROTOCOL_SERIAL)		PROTOCOL_TCPIP(0x05)		
ActStation Number *4	255(0xFF)	固定为 255(0xFF)	对象站侧模块编号	固定为 255(0xFF)	固定为 255(0xFF)
ActThrough NetworkType	0(0x00)		不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnit Number	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_RETHER(0x1002)		

*1: 经由以太网模块(Q系列E71以及R系列E71)访问时,应注意下述事项。

- 对于ActNetworkNumber、ActStationNumber,应指定对象站侧的Q系列E71以及R系列E71的参数设置中设置的值。
- 应设置Q系列E71以及R系列E71参数设置的“站号↔IP关联信息”。此外,进行设置时,对“站号↔IP关联信息设置方式”应指定除自动响应方式以外(IP地址计算方式、表转换方式、并用方式之一)。

*2: 对于I/O地址,应指定将实际I/O起始No.用16相除后的值。

*3: 对多点链接的通道No.应进行下述指定。

0: 使用模块的默认通道

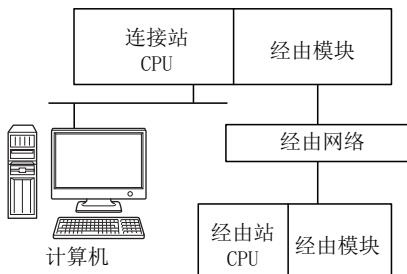
1: 通道1

2: 通道2

*4: 属性的模式为2)的情况下,对ActNetworkNumber及ActStationNumber应指定对象站侧参数中设置的值。

4.3.6 连接站为 RCPU 的以太网通信 (UDP)

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1)*1, *3	CC IE Control	2)*2	×
		CC IE Field	×	×
		MELSECNET/H	×	×
		以太网	2)	2)
		串行通信	3)	3)
		CC-Link	4)	4)

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCUP (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1)*1, *3	CC IE Control	×	×	×	×	×	×
		CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过 R 运动 CPU 访问经由站。

*2：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
Act Destination I0Number *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)
ActDid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
Act Intelligent Preference Bit	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU: 1(0x01) 除上述以外 : 0(0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU: 1(0x01) 除上述以外 : 0(0x00)
ActHost Address	1. 1. 1. 1	指定 IP 地址通信时：连接站侧模块的主机名或 IP 地址 未指定 IP 地址直接通信时：255.255.255.255			
ActDestination PortNumber	0(0x00)	指定 IP 地址通信时：5006 未指定 IP 地址直接通信时：未使用			
Act I0Number *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号： 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976(0x3D0) 无指定 ： 1023(0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActMultiDrop Channel Number *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点 通道 No.	固定为 0(0x00)
ActNetwork Number *4	0(0x00)	固定为 0(0x00)	对象站侧模块网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActPacket Type	0x01 (PACKET_PLC1)			PACKET_PLC1	
ActPassword	空余			连接站中设置的远程口令	
ActProtocol Type	0x04 (PROTOCOL_SERIAL)			PROTOCOL_UDPIP(0x08)	
ActStation Number *4	255(0xFF)	固定为 255(0xFF)	对象站侧模块编号	固定为 255(0xFF)	固定为 255(0xFF)
ActThrough NetworkType	0(0x00)			不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)	
ActTimeOut	10000			以 ms 为单位由用户任意设置	
ActUnit Number	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)			指定 IP 地址通信时：UNIT_RETHER(0x1002) 未指定 IP 地址直接通信时：UNIT_RETHER_DIRECT(0x1003)	

*1: 经由以太网模块 (R 系列 E71) 访问时，应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber，应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“站号↔IP 关联信息”。此外，进行设置时，对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

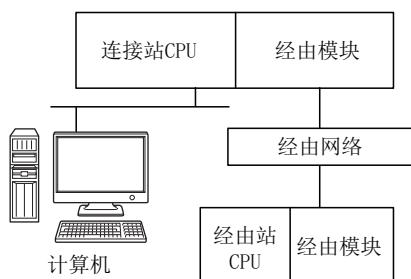
*3: 对多点链接的通道 No. 应进行下述指定。

- 0: 使用模块的默认通道
1: 通道 1
2: 通道 2

*4: 属性的模式为 2) 的情况下，对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.3.7 连接站为以太网端口内置 QCPU 的以太网通信 (TCP)

(1) 构成



(2) 属性的模式

连接站 CPU QnUDE (H) CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU *4
1)	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU QnUDE (H) CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control	2)	2)*1	2)*2	2)*1	2)*1	×
	CC IE Field	2)	2)	×	2)	2)	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	2)	×	×	2)	2)	×
	串行通信	3)*3	×	3)	×	3)	×
	CC-Link	4)	4)	4)	×	4)	×

连接站 CPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU *4
1) *4, *5, *6	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	L CPU	QSCPU	Q 运动 CPU	FXCPU
1) *4, *5, *6	CC IE Control	2)	2)*1	2)*2	2)*1	2)*1	×
	CC IE Field	2)	2)	×	2)	2)	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	4)	4)	4)	×	4)	×

连接站 CPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU
1) *7	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	L CPU	QSCPU	Q 运动 CPU	FXCPU
1) *7	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

- *1：对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。
- *2：对于 L CPU，由于不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。
- *3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。
- *4：多 CPU 构成时，仅 2 号机以后可以访问。
- *5：由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。
- *6：C 语言控制器不支持 MELSOFT 直接连接，因此，以太网端口直接连接时，不能访问。
- *7：Q172D、Q173D、Q172DS、Q173DS 为多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5007			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)
ActIOPortNumber ^{*1}	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber ^{*2}	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActNetworkNumber ^{*3}	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActPassword	空余	连接站中设置的远程口令			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber ^{*3}	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNETHER (0x2C)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

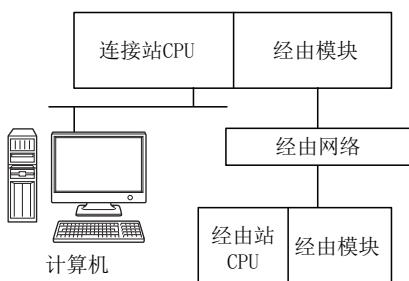
1: 通道 1

2: 通道 2

*3: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.3.8 连接站为以太网端口内置 QCPU 的以太网通信 (UDP)

(1) 构成



(2) 属性的模式

连接站 CPU QnUDE(H) CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU *4
1)	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU QnUDE(H) CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control CC IE Field	2)	2)*1	2)*2	2)*1	2)*1	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	2)	×	×	2)	2)	×
	串行通信	3)*3	×	3)	×	3)	×
	CC-Link	4)	4)	4)	×	4)	×

连接站 CPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU *4
1) *4, *5, *6	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	L CPU	QSCPU	Q 运动 CPU	FXCPU
1) *4, *5, *6	CC IE Control	2)	2)*1	2)*2	2)*1	2)*1	×
	CC IE Field	2)	2)	×	2)	2)	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	4)	4)	4)	×	4)	×

连接站 CPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU
1) *7	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	L CPU	QSCPU	Q 运动 CPU	FXCPU
1) *7	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*2：对于 L CPU，由于不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*4：多 CPU 构成时，仅 2 号机以后可以访问。

*5：由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。

*6：C 语言控制器不支持 MELSOFT 直接连接，因此，以太网端口直接连接时，不能访问。

*7：Q172D、Q173D、Q172DS、Q173DS 为多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5006			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	指定 IP 地址通信时：连接站侧模块的主机名或 IP 地址 不指定 IP 地址直接连接通信时：指定无效			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 RCPU、 R 运动 CPU、 QCUP(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 RCPU、 R 运动 CPU、 QCUP(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)
ActIONumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActNetworkNumber ^{*3}	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActPassword	空余	连接站中设置的远程口令			
\ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPPIP (0x08)			
ActStationNumber ^{*3}	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	指定 IP 地址通信时 : UNIT_QNETH (0x2C) 不指定 IP 地址直接连接通信时 : UNIT_QNETH_DIRECT (0x2D)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

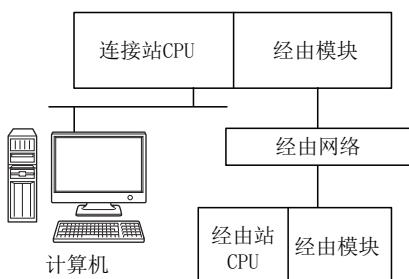
1: 通道 1

2: 通道 2

*3: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.3.9 连接站为以太网端口内置LCPU的以太网通信(TCP)

(1) 构成



(2) 属性的模式

连接站 CPU LCPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field*1	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU LCPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*1	4)	4)*2	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)*3	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于LCPU不支持CC-Link IE控制网络，因此不能访问CC-Link IE控制网络。

*2：由于Q12DCCPU-V(基本功能模式)不支持CC-Link IE现场网络，因此不能访问CC-Link IE现场网络。

*3：冗余CPU的情况下，不能访问位于主基板上的串行通信模块。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 RCPU、 R 运动 CPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActPassword	空余	连接站中设置的远程口令			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber*3	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetwork Type	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNETH (0x52)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

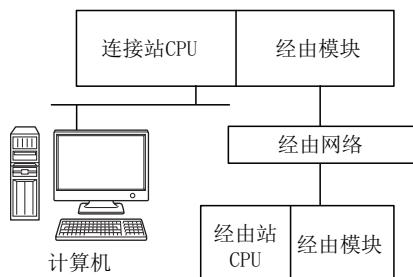
1: 通道 1

2: 通道 2

*3: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.3.10 连接站为以太网端口内置 LCPU 的以太网通信 (UDP)

(1) 构成



(2) 属性的模式

连接站 CPU LCPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU
1)	CC IE Field ^{*1}	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU LCPU	经由网络	经由站 CPU					
		Q CPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field ^{*1}	4)	4) ^{*2}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2) ^{*3}	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*2：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestination IONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDid PropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsid PropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActHostAddress	1. 1. 1. 1	指定 IP 地址通信时：连接站侧模块的主机名或 IP 地址 不指定 IP 地址直接连接通信时：指定无效			
ActIntelligent PreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 RCPU、 R 运动 CPU、 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 RCPU、 R 运动 CPU、 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)
ActIONumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDrop ChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActPassword	空余	连接站中设置的远程口令			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_UDPPIP (0x08)			
ActStationNumber ^{*3}	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetwork Type	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	指定 IP 地址通信时 : UNIT_LNETHETHER(0x52) 不指定 IP 地址直接连接通信时 : UNIT_LNETHETHER_DIRECT(0x53)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

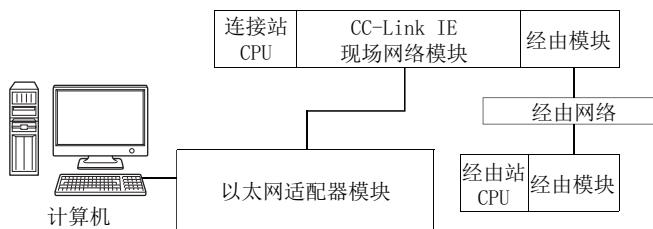
1: 通道 1

2: 通道 2

*3: 属性的模式为 1) 或 2) 的情况下, 应对 ActNetworkNumber 及 ActStationNumber 指定对象站侧参数中设置值。

4.3.11 连接站为以太网适配器模块的以太网通信 (TCP)

(1) 构成



(2) 属性的模式

连接站 CPU QnUDE (H) CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU QnUDE (H) CPU	经由网络	经由站 CPU					
		QCPCU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control CC IE Field	2)	2)*1	2)*2	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field *2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*2	2)	2)*1	2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	以太网适配器模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIONumber ^{*1}	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber ^{*2}	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber ^{*3}	0 (0x00)	连接站侧 CC-Link IE 现场网络模块网络 No.	对象站侧模块网络 No.	连接站侧 CC-Link IE 现场网络模块网络 No.	连接站侧 CC-Link IE 现场网络模块网络 No.

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStation Number ^{*3}	255 (0xFF)	连接站侧 CC-Link IE 现场网络模块站号	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块站号	连接站侧 CC-Link IE 现场网络模块站号
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_NZ2GF_ETB (0x59)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

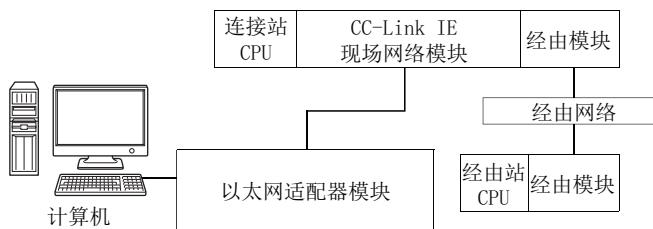
2: 通道 2

*3: 属性的模式为 1)3)4) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定连接站侧 CC-Link IE 现场网络模块参数中设置的值。

属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.3.12 连接站为以太网适配器模块的以太网通信(UDP)

(1) 构成



(2) 属性的模式

连接站 CPU QnUDE(H) CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU QnUDE(H) CPU	经由网络	经由站 CPU					
		QCPCU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control CC IE Field	2)	2)*1	2)*2	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field *2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*2	2)	2)*1	2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	指定 IP 地址通信时：以太网适配器模块的主机名或 IP 地址 不指定 IP 地址直接连接通信时：指定无效			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIOPumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定：1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*3	0 (0x00)	连接站侧 CC-Link IE 现场网络模块网络 No.	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块网络 No.	连接站侧 CC-Link IE 现场网络模块网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPPIP (0x08)			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActStationNumber ^{*3}	255 (0xFF)	连接站侧 CC-Link IE 现场网络模块站号	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块站号	连接站侧 CC-Link IE 现场网络模块站号
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		指定 IP 地址通信时 : UNIT_NZ2GF_ETB (0x59) 不指定 IP 地址直接连接通信时 : UNIT_NZ2GF_ETB_DIRECT (0x5A)		

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

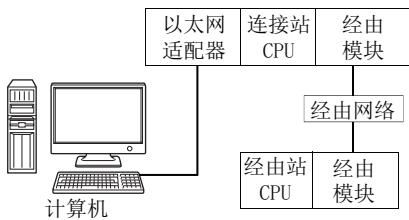
2: 通道 2

*3: 属性的模式为 1) 3) 4) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定连接站侧 CC-Link IE 现场网络模块参数中设置的值。

属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.3.13 连接站为以太网适配器的以太网通信(TCP)

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) ^{*1}	CC IE Control	×	×				
	CC IE Field						
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1}	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3sCPU、FX3g(C)CPU、FX3u(C)CPU 可以访问。

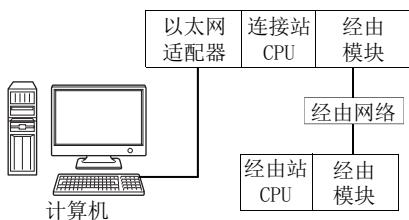
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	
ActCpuType	34(CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitType	0x13(UNIT_QNCPU)	UNIT_FXETHER	

4.3.14 连接站为以太网适配器的以太网通信 (UDP)

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) ^{*1}	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1}	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3sCPU、FX3g(c)CPU、FX3u(c)CPU 可以访问。

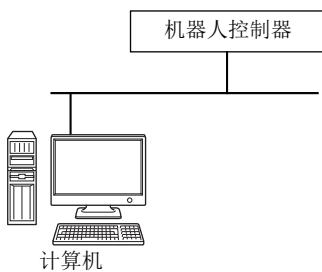
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPPIP	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXETHER_DIRECT	

4.3.15 连接站为机器人控制器的以太网通信

(1) 构成



(2) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	RC_PROTOCOL_TCPIP(0x02)
ActCpuType	34(CPU_Q02CPU)	机器人控制器机型(0x013001)
ActPortNumber	1(PORT_1)	连接端口编号(通常指定为10001)
ActHostAddress	1.1.1.1	机器人控制器的主机名或IP地址
ActTimeOut	10000	以ms为单位由用户任意设置
ActCpuTimeOut	0(0x00)	以10ms为单位由用户任意设置
ActMultiDropChannelNumber	8(TRC_DTR_OR_RTS)	重试次数
ActMxUnitSeries	0(0x00)	2(0x02)

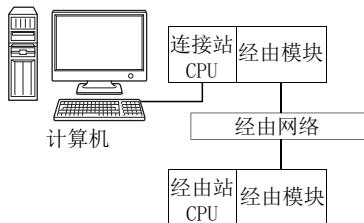
要点

机器人控制器通信时，不能使用程序设置类型的控件。
应使用实用程序设置类型的控件。

4. 4 COM 通信

4. 4. 1 连接站为 QCPU(Q 模式) 的 CPU COM 通信

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
QCPU (Q 模式)	C 语言控制 器		RCPU	R 运动 CPU				
1)	1) ^{*5}	CC IE Control	×	×				
		CC IE Field						
		MELSECNET/H	×	×				
		以太网	×	×				
		串行通信	×	×				
		CC-Link	×	×				
连接站 CPU		经由网络	经由站 CPU					
QCPU (Q 模式)	C 语言控制 器		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1) ^{*5}	CC IE Control	2)	2) ^{*1}	2) ^{*2}	2) ^{*1}	2) ^{*1}	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	2)	2)	×
		以太网	2)	×	×	2)	2)	×
		串行通信	3) ^{*3}	×	3)	×	3)	×
		CC-Link	4)	4)	4)	×	4)	4) ^{*4}

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*4：仅 FX3G(C)CPU、FX3U(C)CPU 且在本网络中支持。

*5：多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)
ActIONumber*2	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

*1: 经由以太网模块 (Q 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧 Q 系列 E71 参数设置中设置的值。
- 应设置 Q 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

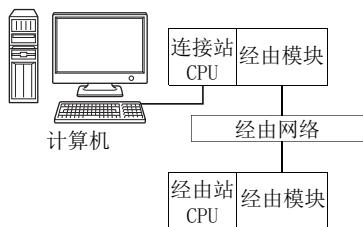
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.4.2 连接站为LCPU的CPU COM通信

(1) 构成



4

(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1) LCPU	CC IE Field*1	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) LCPU	CC IE Field*1	4)	4)*2	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)*3	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于LCPU不支持CC-Link IE控制网络，因此不能访问CC-Link IE控制网络。

*2：由于Q12DCCPU-V(基本功能模式)不支持CC-Link IE现场网络，因此不能访问CC-Link IE现场网络。

*3：冗余CPU的情况下，不能访问位于主基板上的串行通信模块。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)
ActIONumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDropChannelINumber*2	0 (0x00)	固定为 0 (0x00)			
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetwork Type	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPU (0x50)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

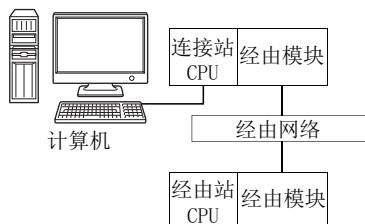
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.4.3 连接站为 Q 运动 CPU 的 CPU COM 通信

(1) 构成



(2) 属性的模式

连接站 CPU Q 运动 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) ^{*1}	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1}	CC-Link	×	×				
连接站 CPU Q 运动 CPU	经由网络	经由站 CPU					
		QCPCU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： Q172CPU、Q173CPU、Q172HCPU、Q173HCPU 的情况下不能访问。

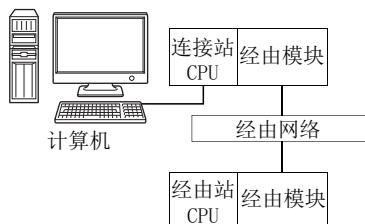
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、 BAUDRATE_57600、BAUDRATE_115200
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型
ActIONumber	1023 (0x3FF)	多CPU时 管理CPU: 1023 (0x3FF) 2号CPU: 993 (0x3E1) 3号CPU: 994 (0x3E2) 4号CPU: 995 (0x3E3)
ActPortNumber	1 (PORT_1)	计算机侧COM端口No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)
ActTimeOut	10000	以ms为单位由用户任意设置
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNMOTION (0x1C)

4.4.4 连接站为 FXCPU 的 CPU COM 通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) FXCPU	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) FXCPU	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	2)*1

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3G(c)CPU、FX3U(c)CPU 可以访问。

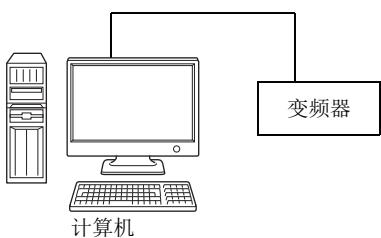
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActBaudRate	19200 (BAUDRATE_19200)	FX _{0(S)} 、FX _{0N} 、FX ₁ 、FX _{1S} 、FX _{2(C)} (固定为 BAUDRATE_9600)	
		FX _{1N(C)} 、FX _{2N(C)} (BAUDRATE_9600、BAUDRATE_19200)	
		FX _{3S} 、FX _{3UC} 、FX _{3G(C)} (BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、 BAUDRATE_57600、BAUDRATE_115200)	
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActDestinationIONumber	0(0x00)	固定为 0(0x00)	固定为 1023(0x3FF)
ActDidPropertyBit	1(0x01)	0(0x00)	
ActDsidPropertyBit	1(0x01)	0(0x00)	
ActIONumber	1023(0x3FF)	固定为 0(0x00)	连接站模块编号
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL(0x04)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0(0x00)	固定为 0(0x00)	对象站模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU(0x0F)	

4.4.5 连接站为变频器的 COM 通信

(1) 构成



(2) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

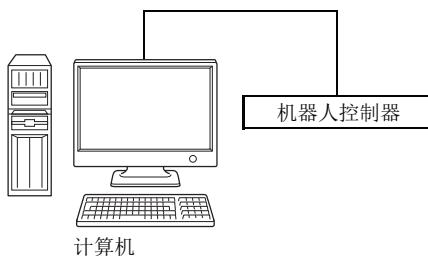
属性	默认值	属性的模式
		1)
ActMxUnitSeries	0(0x00)	1(0x01)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	COMM_RS232C(0x00)
ActStationNumber	255(0xFF)	变频器的站号(0~31)
ActCpuType	34(CPU_Q02CPU)	固定为 INV_A800(0x1E60)
ActPortNumber	1(PORT_1)	计算机侧端口 No.
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_4800、BAUDRATE_9600、BAUDRATE_19200、 BAUDRATE_38400、BAUDRATE_57600、 BAUDRATE_115200
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActCpuTimeOut	0(0x00)	以 10ms 为单位由用户任意设置
ActPacketType	1(CRLF_CR)	CRLF_NONE、CRLF_CR、CRLF_CRLF
ActDataBits	8(DATABIT_8)	DATABIT_7、DATABIT_8
ActParity	1(ODD_PARITY)	NO_PARRITY、ODD_PARITY、EVEN_PARITY
ActStopBits	0(STOPBIT_ONE)	STOPBIT_ONE、STOPBIT_TWO

要 点

变频器通信时，不能使用程序设置类型的控件。
应使用实用程序设置类型的控件。

4.4.6 连接站为机器人控制器的 COM 通信

(1) 构成



4

(2) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	RC_PROTOCOL_SERIAL(0x01)
ActCpuType	34(CPU_Q02CPU)	机器人控制器机型(0x013001)
ActPortNumber	1(PORT_1)	计算机侧端口No.
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_4800、BAUDRATE_9600、 BAUDRATE_19200、BAUDRATE_38400、 BAUDRATE_57600、BAUDRATE_115200
ActTimeOut	10000	以ms为单位由用户任意设置
ActCpuTimeOut	0(0x00)	以10ms为单位由用户任意设置
ActDataBits	8(DATABIT_8)	DATABIT_7、DATABIT_8
ActParity	1(ODD_PARITY)	NO_PARRITY、ODD_PARITY、EVEN_PARITY
ActStopBits	0(ONESTOPBIT)	ONESTOPBIT、ONE5STOPBITS、TWOSTOPBITS
ActMultiDropChannelNumber	0(0x00)	重试次数
ActThroughNetworkType	0(0x00)	有顺序/无顺序(0/1)
ActMxUnitSeries	0(0x00)	2(0x02)

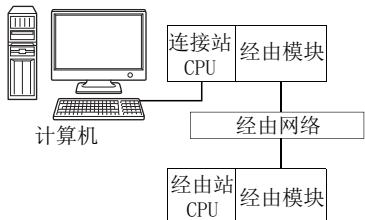
要 点

机器人控制器通信时，不能使用程序设置类型的控件。
应使用实用程序设置类型的控件。

4.5 USB 通信

4.5.1 连接站为 RCPU 的 USB 通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control	2)*1	×
	CC IE Field	×	×
	MELSECNET/H	2)	2)
	以太网	3)	3)
	串行通信	4)	4)

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
Act Destination IONumber *2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定 ：1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定 ：1023 (0x3FF)
ActDid PropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsid PropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
Act Intelligent Preference Bit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU (Q 模式)、 C 语言控制器、 LCPU、RCPU： 1 (0x00) 除上述以外： 0 (0x00)	对象站 QCPU (Q 模式)、 C 语言控制器、 LCPU、RCPU： 1 (0x00) 除上述以外： 0 (0x00)
Act IONumber *2	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定 ：1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU：992 (0x3E0) 2 号 CPU：993 (0x3E1) 3 号 CPU：994 (0x3E2) 4 号 CPU：995 (0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976 (0x3D0) 无指定 ：1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDrop Channel Number *3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点 通道 No.	固定为 0 (0x00)
ActNetwork Number	0 (0x00)	固定为 0 (0x00)	对象站侧模块 网络 No.	固定为 0 (0x00)	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActProtocol Type	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB			
ActPacket Type	0x01 (PACKET_PLCL)	PACKET_PLCL			
ActStation Number	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块编号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActConnect UnitNumber	0 (0x00)	固定为 0 (0x00)			
ActPort Number	1 (PORT_1)	固定为 1 (PORT_1)			
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0 (0x00)			
ActHost Address	1. 1. 1. 1	固定为 NULL			
ActCpu TimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	固定为 0 (0x00)			
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)			
ActParity	1 (ODD_PARITY)	固定为 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)			
ActSum Check	0 (NO_SUM_CHECK)	固定为 0 (0x00)			
ActSource Network Number	0 (0x00)	固定为 0 (0x00)			
ActSource Station Number	0 (0x00)	固定为 0 (0x00)			
Act Destination PortNumber	0 (0x00)	固定为 0 (0x00)			
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnit Number	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RUSB			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

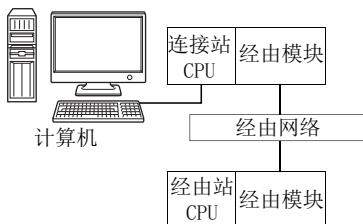
*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

- 0: 使用模块的默认通道
- 1: 通道 1
- 2: 通道 2

4.5.2 连接站为R运动CPU的USB通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R运动 CPU				
1) ^{*1,*2}	CC IE Control	×	×				
	CC IE Field						
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1,*2}	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q模式)	C语言 控制器	LCPU	QSCPU	Q运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过R运动CPU访问经由站。

*2：多CPU构成时，仅2号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2) *1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
Act Destination IONumber *2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定 ： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定 ： 1023 (0x3FF)
ActDid PropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsid PropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
Act Intelligent Preference Bit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 RCPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU: 1 (0x00) 除上述以外 : 0 (0x00)	对象站 RCPU、 QCPU(Q 模式)、 C 语言控制器、 LCPU: 1 (0x00) 除上述以外 : 0 (0x00)
Act IONumber *2	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定 ： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 对象站侧 控制系统： 976 (0x3D0) 无指定 ： 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDrop Channel Number *3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点 通道 No.	固定为 0 (0x00)
ActNetwork Number	0 (0x00)	固定为 0 (0x00)	对象站侧模块 网络 No.	固定为 0 (0x00)	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2) *1	3)	4)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB			
ActStationNumber	255(0xFF)	固定为 255(0xFF)	对象站侧模块编号	固定为 255(0xFF)	固定为 255(0xFF)
ActThroughNetworkType	0(0x00)	不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RUSB			

*1: 经由以太网模块(R系列E71)访问时,应注意下述事项。

- 对于ActNetworkNumber、ActStationNumber,应指定对象站侧的R系列E71的参数设置中设置的值。
- 应设置R系列E71参数设置的“站号↔IP关联信息”。此外,进行设置时,对“站号↔IP关联信息设置方式”应指定除自动响应方式以外(IP地址计算方式、表转换方式、并用方式之一)。

*2: 对于I/O地址,应指定将实际I/O起始No.用16相除后的值。

*3: 对多点链接的通道No.应进行下述指定。

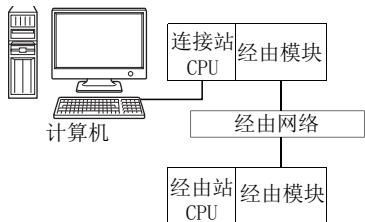
0: 使用模块的默认通道

1: 通道1

2: 通道2

4.5.3 连接站为 QCPU(Q 模式) 的 USB 通信

(1) 构成



(2) 属性的模式

连接站 CPU QCPU (Q 模式)	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1)	CC-Link	×	×				
连接站 CPU QCPU (Q 模式)	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	2)	2)*1	2)*2	2)*1	2)*1	×
	CC IE Field	2)	2)	×	2)	2)	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	2)	×	×	2)	2)	×
	串行通信	3)*3	×	3)	×	3)	×
	CC-Link	4)	4)	4)	×	4)	4)*4

连接站 CPU C 语言控制器	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU C 语言控制器	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control CC IE Field	2)	2) ^{*1}	2) ^{*2}	2) ^{*1}	2) ^{*1}	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：对于 R 运动 CPU、Q12DCCPU-V（基本功能模式）、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*4：仅 FX3G(C)CPU、FX3U(C)CPU 加上本网络才支持。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)
ActIONumber*2	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)			

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCP)		UNIT_QNUSB (0x16)		

*1: 经由以太网模块 (Q 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧 Q 系列 E71 参数设置中设置的值。
- 应设置 Q 系列 E71 参数设置的“站号↔IP 关联信息”。对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

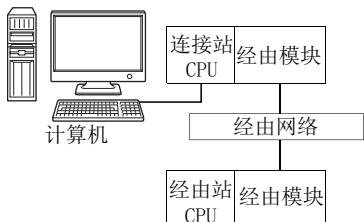
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.5.4 连接站为 LCPU 的 USB 通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field ^{*1}	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field ^{*1}	4)	4) ^{*2}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2) ^{*3}	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*2：由于 Q12DCCPU-V（基本功能模式）不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1(0x01) 除上述以外 : 0(0x00)	固定为 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNUSB (0x51)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

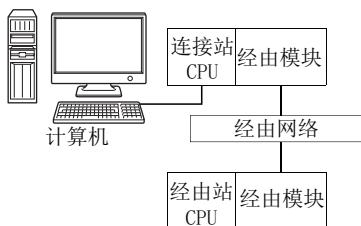
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.5.5 连接站为QSCPU的USB通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPUs	R 运动 CPU				
1) ^{*1}	CC IE Control	×	×				
	CC IE Field						
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1}	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCpus (Q 模式)	C 语言控 制器	LCPUs	QSCPUs	Q 运动 CPU	FXCPUs
1) ^{*1}	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： 不能经由 QSCPU 访问经由站。

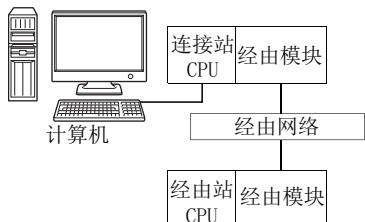
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitType	0x13 (UNIT_QNCPUs)	UNIT_QSUSB (0x29)

4.5.6 连接站为 Q 运动 CPU 的 USB 通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		R CPU	R 运动 CPU				
1) ^{*1}	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1}	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		Q CPU (Q 模式)	C 语言控 制器	L CPU	Q S CPU	Q 运动 CPU	FX CPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: Q172CPU、Q173CPU、Q172HCPU、Q173HCPU 的情况下不能访问。

(3) 属性一览

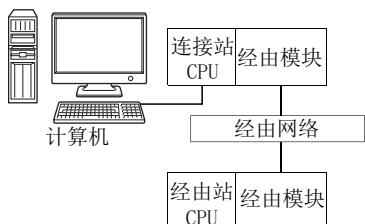
各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型
ActIOPNumber	1023 (0x3FF)	对象站侧 多 CPU 时 管理 CPU: F1023 (0x3FF) 2 号 CPU: 993 (0xE1) 3 号 CPU: 994 (0xE2) 4 号 CPU: 995 (0xE3)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNMOTIONUSB (0x1D)

4.5 USB 通信
4.5.6 连接站为 Q 运动 CPU 的 USB 通信

4.5.7 连接站为 FXCPU 的 USB 通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) FXCPU	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) FXCPU	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	2)*1

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3G(C)CPU、FX3U(C)CPU 可以访问。

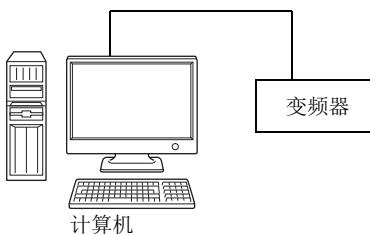
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActDestinationI0Number	0 (0x00)	固定为 0 (0x00)	固定为 1023 (0x3FF)
ActI0Number	1023 (0x3FF)	固定为 0 (0x00)	连接站模块编号
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)	

4.5.8 连接站为变频器的 USB 通信

(1) 构成



(2) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

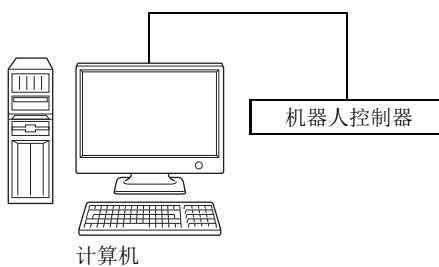
属性	默认值	属性的模式
		1)
ActMxUnitSeries	0(0x00)	1(0x01)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	COMM_USB(0x01)
ActStationNumber	255(0xFF)	变频器的站号(0 ~ 31)
ActCpuType	34(CPU_Q02CPU)	固定为 INV_A800(0x1E60)
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActCpuTimeOut	0(0x00)	以 10ms 为单位由用户任意设置

要 点

变频器通信时，不能使用程序设置类型的控件。
应使用实用程序设置类型的控件。

4.5.9 连接站为机器人控制器的 USB 通信

(1) 构成



4

(2) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

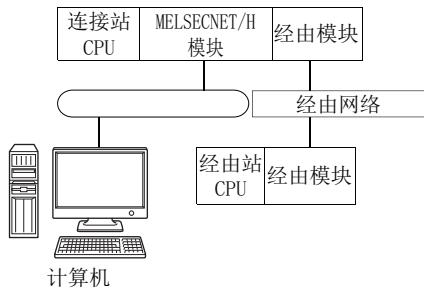
属性	默认值	属性的模式
		1)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	RC_PROTOCOL_USB (0x04)
ActCpuType	34(CPU_Q02CPU)	机器人控制器机型 (0x013001)
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActCpuTimeOut	0(0x00)	以 10ms 为单位由用户任意设置
ActMultiDropChannelNumber	0(0x00)	重试次数
ActMxUnitSeries	0(0x00)	2(0x02)

要 点

机器人控制器通信时，不能使用程序设置类型的控件。
应使用实用程序设置类型的控件。

4. 6 MELSECNET/H 通信

(1) 构成



(2) 属性的模式

自板	连接站 CPU			经由网络	经由站 CPU	
	QCPU (Q 模式)	QSCPU	Q 运动 CPU		RCPU	R 运动 CPU
1)	2)	2)* ¹	2)* ^{1, *5}	CC IE Control	×	×
				CC IE Field	×	×
				MELSECNET/H	×	×
				以太网	×	×
				串行通信	×	×
				CC-Link	×	×

自板	连接站 CPU			经由网络	经由站 CPU					
	QCPU (Q 模式)	QSCPU	Q 运动 CPU		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	2)	2)* ¹	2)* ^{1, *5}	CC IE Control	2)	2)* ²	2)* ³	2)* ^{1, *2}	×	×
				CC IE Field	2)	2)	×	2)* ¹	×	×
				MELSECNET/H	2)	2)	×	2)* ¹	×	×
				以太网	2)	×	×	2)* ¹	×	×
				串行通信	3)* ⁴	×	3)	×	3)	×
				CC-Link	4)	4)	4)	×	4)	×

自板	连接站 CPU C 语言控制器	经由网络	经由站 CPU	
			RCPU	R 运动 CPU
1) 2)*6		CC IE Control CC IE Field	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

自板	连接站 CPU C 语言控制器	经由网络	经由站 CPU					
			QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) 2)*6		CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 不能经由 QSCPU 及 Q 运动 CPU 访问经由站。

*2: 对于 Q12DCCPU-V(基本功能模式)及 QSCPU, 由于不支持 CC-Link IE 现场网络, 因此不能访问 CC-Link IE 现场网络。

*3: 由于 LCPU 不支持 CC-Link IE 控制网络, 因此不能访问 CC-Link IE 控制网络。

*4: 冗余 CPU 的情况下, 不能访问位于主基板上的串行通信模块。

*5: 多 CPU 构成时, 仅 2 号机以后可以访问。

*6: 由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径, 因此不能访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIOPumber ^{*1}	1023 (0x3FF)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber ^{*2}	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	连接站侧模块网络 No.	连接站侧模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 MELSECNET/H 板的端口 No.、PORT1 ~ PORT4 (第 1 个~ 第 4 个)			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_MNETH (0x0F)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	连接站侧模块站号	连接站侧模块站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETHBOARD (0x1E)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

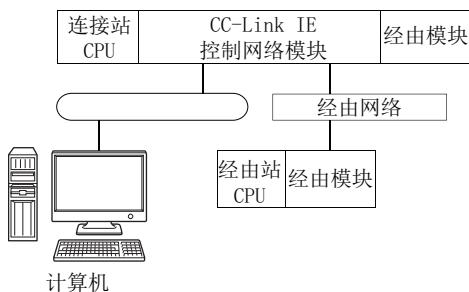
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.7 CC-Link IE 控制网络通信

(1) 构成



(2) 属性的模式

自板	连接站 CPU			经由网络	经由站 CPU					
	QCPU (Q 模式)	QSCPU	Q 运动 CPU		RCPU	R 运动 CPU				
1)	2)	2) ^{*1}	2) ^{*1, *5}	CC IE Control	×	×				
				CC IE Field						
				MELSECNET/H	×	×				
				以太网	×	×				
				串行通信	×	×				
CC-Link										
自板	连接站 CPU			经由网络	经由站 CPU					
	QCPU (Q 模式)	QSCPU	Q 运动 CPU		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	2)	2) ^{*1}	2) ^{*1, *5}	CC IE Control	2)	2) ^{*2}	2) ^{*3}	2) ^{*1, *2}	×	×
				CC IE Field						
				MELSECNET/H	2)	2)	×	2) ^{*1}	×	×
				以太网	2)	×	×	2) ^{*1}	×	×
				串行通信	3) ^{*4}	×	3)	×	3)	×
CC-Link							4)	4)	4)	4)

自板	连接站 CPU C 语言控制器	经由网络	经由站 CPU	
			RCPU	R 运动 CPU
1) 2)*6		CC IE Control CC IE Field	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

自板	连接站 CPU C 语言控制器	经由网络	经由站 CPU					
			QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) 2)*6		CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能经由 QSCPU 及 Q 运动 CPU 访问经由站。

*2：对于 Q12DCCPU-V(基本功能模式)及 QSCPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*5：多 CPU 构成时，仅 2 号机以后可以访问。

*6：由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestination IONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIONumber*1	1023 (0x3FF)	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDrop ChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	连接站侧模块网络 No.	连接站侧模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 CC-Link IE 控制网卡的端口 No.、PORT1 ~ PORT4(第 1 个~ 第 4 个)			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_MNETG (0x14)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	连接站侧模块站号	连接站侧模块站号
ActThroughNetwork Type	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETGBOARD (0x2B)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

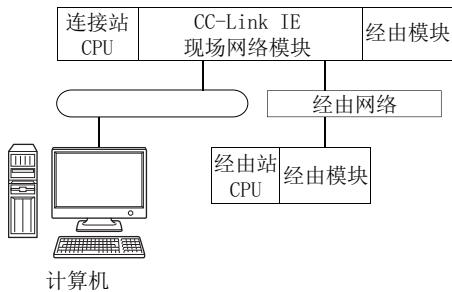
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.8 CC-Link IE 现场网络通信

(1) 构成



(2) 属性的模式

自板	连接站 CPU QCPCU (Q 模式)	经由网络	经由站 CPU	
			RCPU	R 运动 CPU
1)	2)	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

自板	连接站 CPU QCPCU (Q 模式)	经由网络	经由站 CPU					
			QCPCU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	2)	CC IE Control	2)	2)*1	2)*2	×	×	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	×	×
		以太网	2)	×	×	×	×	×
		串行通信	3)	×	3)	×	×	×
		CC-Link	4)	4)	4)	×	×	×

自板	连接站 CPU LCPU	经由网络	经由站 CPU	
			RCPU	R 运动 CPU
1)	2)	CC IE Field *2	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

自板	连接站 CPU LCPU	经由网络	经由站 CPU					
			QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	2)	CC IE Field *2	2)	×	2)	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	3)	×	3)	×	×	×
		CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 Q12DCCPU-V(基本功能模式) 及 QSCPU 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestination IONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIONumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址
ActMultiDrop ChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	连接站侧模块网络 No.	连接站侧模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 CC-Link IE 现场网卡的端口 No.、PORT1 ~ PORT4(第 1 个~ 第 4 个)			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_CCIEF (0x15)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	连接站侧模块站号	连接站侧模块站号
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_CCIEFBOARD (0x2F)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

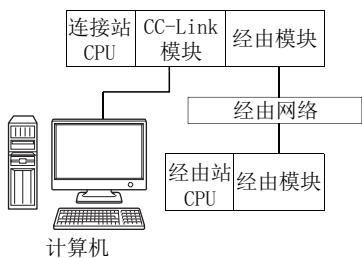
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.9 CC-Link 通信

(1) 构成



(2) 属性的模式

自板	连接站 CPU		经由网络	经由站 CPU	
	QCPU (Q 模式)	Q 运动 CPU		RCPU	R 运动 CPU
1)	2)	2)*1,*4	CC IE Control	×	×
			CC IE Field	×	×
			MELSECNET/H	×	×
			以太网	×	×
			串行通信	×	×
			CC-Link	×	×

自板	连接站 CPU		经由网络	经由站 CPU					
	QCPU (Q 模式)	Q 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	2)	2)*1,*4	CC IE Control	3)	3)*2	3)	3)*2	3)*2	×
			CC IE Field	3)	3)	×	3)	3)	×
			MELSECNET/H	3)	3)	×	3)	3)	×
			以太网	3)	×	×	3)	3)	×
			串行通信	×	×	×	×	×	×
			CC-Link	×	×	×	×	×	×

自板	连接站 CPU	经由网络	经由站 CPU	
	C 语言控制器		RCPU	R 运动 CPU
1) 2) ^{*5}	CC IE Control CC IE Field	×	×	
	MELSECNET/H	×	×	
	以太网	×	×	
	串行通信	×	×	
	CC-Link	×	×	

自板	连接站 CPU	经由网络	经由站 CPU				
	C 语言控制器		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU
1) 2) ^{*5}	CC IE Control CC IE Field	3)	3) ^{*2}	×	3) ^{*2}	3) ^{*2}	×
	MELSECNET/H	3)	3)	×	3)	3)	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

自板	连接站 CPU	经由网络	经由站 CPU	
	LCPU		RCPU	R 运动 CPU
1) 2)	CC IE Control CC IE Field	×	×	
	MELSECNET/H	×	×	
	以太网	×	×	
	串行通信	×	×	
	CC-Link	×	×	

自板	连接站 CPU	经由网络	经由站 CPU				
	LCPU		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU
1) 2)	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能经由 Q 运动 CPU 访问经由站。

*2：对于 Q12DHCCPU-V(基本功能模式)、QSCPU 及 Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4：多 CPU 构成时，仅 2 号机以后可以访问。

*5：由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。

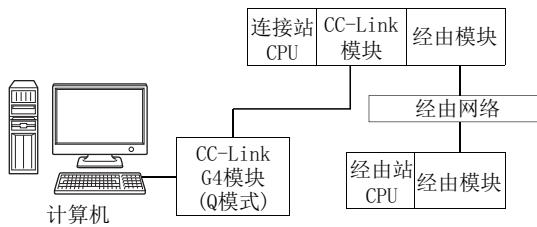
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式		
		1)	2)	3)
ActCpuType	34(CPU_Q02CPU)	对应于对象站的 CPU 类型		
ActDestinationIONumber	0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActIONumber	1023 (0x3FF)	固定为 0 (0x00)	固定为 1023 (0x3FF)	固定为 1023 (0x3FF)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 CC-Link 板的端口 No.、PORT1 ~ PORT4 (第 1 个~ 第 4 个)		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_CCLINK (0x07)		
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧 CC-Link 模块站号	对象站侧模块站号
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	连接站侧 CC-Link 模块 站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_CCLINKBOARD (0x0C)		

4. 10 CC-Link G4 通信

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCPCU (Q 模式)	Q 运动 CPU		RCPUCPU	R 运动 CPU
1) 2)*1,*5	2)*1,*5	CC IE Control	×	×
		CC IE Field	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)	Q 运动 CPU		QCPCU (Q 模式)	C 语言控 制器	LCPUCPU	QSCPCU	Q 运动 CPU	FXCPU
1) 2)*1,*5	2)*1,*5	CC IE Control	2)	2)*2	2)*3	2)*2	2)*2	×
		CC IE Field	2)	2)	×	2)	2)	×
		MELSECNET/H	2)	2)	×	2)	2)	×
		以太网	2)	×	×	2)	2)	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) ^{*4}	CC IE Control CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*4}	CC IE Control CC IE Field	2)	2) ^{*2}	2) ^{*3}	2) ^{*2}	2) ^{*2}	×
	MELSECNET/H	2)	2)	×	2)	2)	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×
连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

- *1: 不能经由 Q 运动 CPU 访问经由站。
- *2: 对于 R 运动 CPU、Q12DHCCPU-V(基本功能模式)、QSCPU、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。
- *3: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。
- *4: 由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。
- *5: 多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式		
		1)	2)	3)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200		
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号		
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆		
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型		
ActIOnumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由 模块 I/O 地址
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)		
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000	以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块 站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_G4QNCPU (0x1B)		

*1：对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

4.11 GX Simulator 通信

(1) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		本站	其它站
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SHAREDMEMORY (0x06)	
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站站号
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_SIMULATOR (0x0B)	

要 点

在属性设置中，包含有下述设置之一时将访问本站。

- ActNetworkNumber：设置超出了 1 ~ 255 的范围时
- ActStationNumber：设置超出了 1 ~ 64 的范围时

4.12 GX Simulator2 通信

(1) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
ActTargetSimulator	0 (0x00)	参阅 3.2 节 的属性 [ActTargetSimulator]
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_SIMULATOR2 (0x30)

4.13 MT Simulator2 通信

(1) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

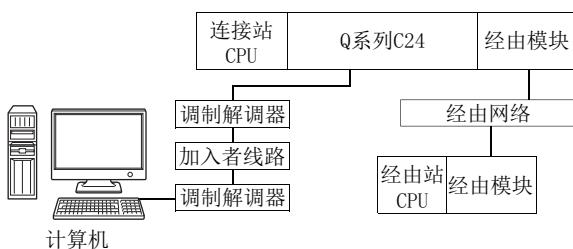
属性	默认值	属性的模式
ActTargetSimulator	0 (0x00)	参阅 3.2 节 的属性 [ActTargetSimulator]
ActCpuType	CPU_Q02CPU	对应于对象站的 CPU 类型 (Q 运动 CPU)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_SIMULATOR2 (0x30)

4.14 调制解调器通信

4.14.1 连接站为Q系列C24的调制解调器通信

(1) 连接站侧有除Q系列C24以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU QCPCU (Q 模式)	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU QCPCU (Q 模式)	经由网络	经由站 CPU					
		QCPCU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control CC IE Field	2)	2) ^{*1}	2) ^{*2}	2) ^{*1}	×	×
	MELSECNET/H	2)	2)	×	2)	×	×
	以太网	2)	×	×	2)	×	×
	串行通信	3) ^{*3}	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：对于Q12DCCPU-V(基本功能模式)及QSCPU，由于不支持CC-Link IE 现场网络，因此不能访问CC-Link IE 现场网络。

*2：由于LCPU不支持CC-Link IE 控制网络，因此不能访问CC-Link IE 控制网络。

*3：冗余CPU的情况下，不能访问位于主基板上的串行通信模块。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActATCommand	空余	用户任意			
ActATCommand PasswordCancel RetryTimes	3	用户任意			
ActATCommand ResponseWait Time	1	用户任意(单位:秒)			
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200 (根据Q系列C24的设置)			
ActCallback CancelWaitTime	90	用户任意(单位:秒)			
ActCallback DelayTime	20	用户任意(单位:秒)			
ActCallback Number	空余	用户任意的电话号码			
ActCallback ReceptionWaitin g TimeOut	120	用户任意(单位:秒)			
ActConnection CDWaitTime	90	用户任意(单位:秒)			
ActConnection ModemReport WaitTime	5	用户任意(单位:秒)			
ActConnectUnit Number	0 (0x00)	连接站侧模块站号			
ActConnectWay	0 (TEL_AUTO_ CONNECT)	TEL_AUTO_CONNECT(0)、TEL_AUTO_CALLBACK(1)、TEL_AUTO_CALLBACK_NUMBER(2)、 TEL_CALLBACK(3)、TEL_CALLBACK_NUMBER(4)、TEL_CALLBACK_REQUEST(5)、 TEL_CALLBACK_REQUEST_NUMBER(6)、TEL_CALLBACK_WAIT(7) (根据回调功能的设置)			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDialNumber	空余	用户任意的电话号码			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDisconnectio nCDWaitTime	5	用户任意 (单位：秒)			
ActDisconnectio nDelayTime	3	用户任意 (单位：秒)			
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	连接站侧经由模块 I/O 地址	
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0)、LINETYPE_TONE(1)、LINETYPE_ISDN(2)			
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActOutsideLineNumber	空余	用户任意的外线发送编号			
ActPassword	空余	Q 系列 C24 中设置的口令			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActPassword					
CancelResponse					
WaitTime		用户任意 (单位：秒)			
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIALMODEM (0x0E)			
ActStation Number	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActTransmission DelayTime	0	用户任意 (单位：秒)			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

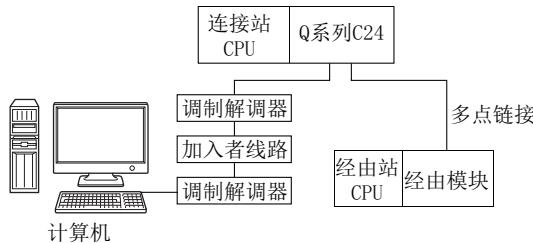
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 Q 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)			RCPUCPU	R 运动 CPU				
独立模式 *1	1)	串行通信	×	×				
连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)			QCPCU (Q 模式)	C 语言控 制器	LCPUCPU	QSCPCPU	Q 运动 CPU	FXCPU
独立模式 *1	1)	串行通信	2)*2	×	2)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： 独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

*2： 冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActATCommand	空余	用户任意	
ActATCommandPasswordCancelRetryTimes	3	用户任意	
ActATCommandResponseWaitTime	1	用户任意(单位：秒)	
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、 BAUDRATE_57600、BAUDRATE_115200 (根据Q系列C24的设置)	
ActCallbackCancelWaitTime	90	用户任意(单位：秒)	
ActCallbackDelayTime	20	用户任意(单位：秒)	
ActCallbackNumber	空余	用户任意的电话号码	
ActCallbackReceptionWaitingTimeOut	120	用户任意(单位：秒)	
ActConnectionCDWaitTime	90	用户任意(单位：秒)	
ActConnectionModemReportWaitTime	5	用户任意(单位：秒)	
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号	
ActConnectWay	0 (TEL_AUTO_CONNECT)	TEL_AUTO_CONNECT(0)、TEL_AUTO_CALLBACK(1)、 TEL_AUTO_CALLBACK_NUMBER(2)、TEL_CALLBACK(3)、 TEL_CALLBACK_NUMBER(4)、TEL_CALLBACK_REQUEST(5)、 TEL_CALLBACK_REQUEST_NUMBER(6)、TEL_CALLBACK_WAIT(7) (根据回调功能的设置)	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型	
ActDestinationIONumber	0 (0x00)	固定为0 (0x00)	对象站侧 单CPU时 固定为1023 (0x3FF) 多CPU时 连接CPU机号：1023 (0x3FF) 1号CPU：992 (0x3E0) 2号CPU：993 (0x3E1) 3号CPU：994 (0x3E2) 4号CPU：995 (0x3E3) 冗余CPU时 控制系统：976 (0xD0) 无指定：1023 (0x3FF)
ActDialNumber	空余	用户任意的电话号码	
ActDidPropertyBit	1 (0x01)	固定为1 (0x01)	固定为0 (0x00)
ActDisconnectionCDWaitTime	5	用户任意(单位：秒)	
ActDisconnectionDelayTime	3	用户任意(单位：秒)	
ActDsidPropertyBit	1 (0x01)	固定为1 (0x01)	固定为1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为0 (0x00)	

属性	默认值	属性的模式	
		1)	2)
ActionNumber*1	1023 (0x3FF)	对象站侧 单CPU时 固定为1023 (0x3FF) 多CPU时 连接CPU机号：1023 (0x3FF) 1号CPU：992 (0x3E0) 2号CPU：993 (0x3E1) 3号CPU：994 (0x3E2) 4号CPU：995 (0x3E3) 冗余CPU时 控制系统：976 (0x3D0) 无指定：1023 (0x3FF)	连接站侧经由模块I/O地址
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0)、LINETYPE_TONE(1)、LINETYPE_ISDN(2)	
ActMultiDropChannelNumber*2	0 (0x00)	固定为0 (0x00)	多点通道No.
ActNetworkNumber	0 (0x00)		固定为0 (0x00)
ActOutsideLineNumber	空余		用户任意的外线发送编号
ActPassword	空余		Q系列C24中设置的口令
ActPasswordCancelResponseWaitTime	5		用户任意(单位：秒)
ActPortNumber	1 (PORT_1)		计算机侧COM端口No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)		PROTOCOL_SERIALMODEM (0x0E)
ActStationNumber	255 (0xFF)		固定为255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含MELSECNET/10 : 0 (0x00) 包含MELSECNET/10 : 1 (0x01)	
ActTimeOut	10000		以ms为单位由用户任意设置
ActTransmissionDelayTime	0		用户任意(单位：秒)
ActUnitNumber	0 (0x00)	固定为0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_QJ71C24 (0x19)

*1: 对于I/O地址，应指定将实际I/O起始No.用16相除后的值。

*2: 对多点链接的通道No.应进行下述指定。

0: 使用模块的默认通道

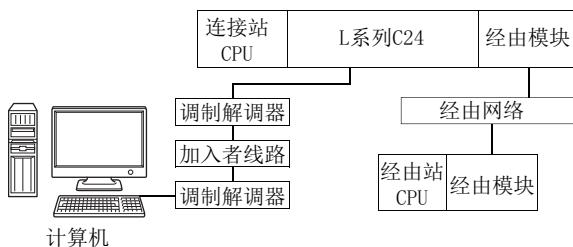
1: 通道1

2: 通道2

4.14.2 连接站为 L 系列 C24 的调制解调器通信

(1) 连接站侧除 LJ71C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1) ^{*1}	CC IE Field *2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	CC IE Field *2	4)	4) ^{*3}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2) ^{*4}	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 表示 CH2 侧的设置。(CH1 侧固定为独立模式)

*2: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3: 由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4: 冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActATCommand	空余	用户任意			
ActATCommand PasswordCancel RetryTimes	3	用户任意			
ActATCommand ResponseWaitTime	1	用户任意(单位:秒)			
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200 (根据 LJ71C24 模块的设置)			
ActCallback CancelWaitTime	90	用户任意(单位:秒)			
ActCallback DelayTime	20	用户任意(单位:秒)			
ActCallbackNumber	空余	用户任意的电话号码			
ActCallback ReceptionWaiting TimeOut	120	用户任意(单位:秒)			
ActConnection CDWaitTime	90	用户任意(单位:秒)			
ActConnection ModemReportWait Time	5	用户任意(单位:秒)			
ActConnectUnit Number	0 (0x00)	连接站侧模块站号			
ActConnectWay	0 (TEL_AUTO_CONNECT)	TEL_AUTO_CONNECT(0)、TEL_AUTO_CALLBACK(1)、TEL_AUTO_CALLBACK_NUMBER(2)、 TEL_CALLBACK(3)、TEL_CALLBACK_NUMBER(4)、TEL_CALLBACK_REQUEST(5)、 TEL_CALLBACK_REQUEST_NUMBER(6)、TEL_CALLBACK_WAIT(7)			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型			
ActDestination IONumber	0 (0x00)	固定为0 (0x00)	对象站侧 单CPU时 固定为 1023 (0x3FF) 多CPU时 连接CPU机号： 1023 (0x3FF) 1号CPU: 992 (0x3E0) 2号CPU: 993 (0x3E1) 3号CPU: 994 (0x3E2) 4号CPU: 995 (0x3E3) 冗余CPU时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	对象站侧 单CPU时 固定为 1023 (0x3FF) 多CPU时 连接CPU机号： 1023 (0x3FF) 1号CPU: 992 (0x3E0) 2号CPU: 993 (0x3E1) 3号CPU: 994 (0x3E2) 4号CPU: 995 (0x3E3) 冗余CPU时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)	固定为0 (0x00)

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActDialNumber	空余	用户任意的电话号码			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDisconnection CDWaitTime	5	用户任意 (单位：秒)			
ActDisconnection DelayTime	3	用户任意 (单位：秒)			
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligent PreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIOnumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧经由模块 I/O 地址	连接站侧经由模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3) 冗余 CPU 时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0)、LINETYPE_TONE(1)、LINETYPE_ISDN(2)			
ActMultiDrop ChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActOutsideLine Number	空余	用户任意的外线发送编号			
ActPassword	空余	模块中设置的口令			
ActPasswordCancel ResponseWaitTime	5	用户任意 (单位：秒)			
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIALMODEM (0x0E)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActTransmission DelayTime	0	用户任意 (单位：秒)			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

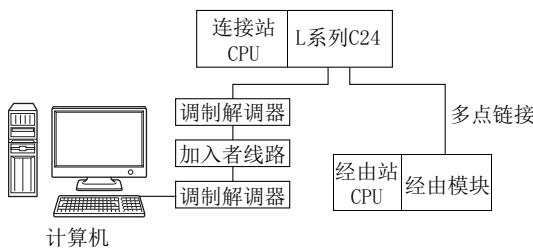
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 LJ71C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
LCPU	RCPU		R 运动 CPU	
独立模式	1)	串行通信	×	×

连接站 CPU		经由网络	经由站 CPU					
LCPU	QCPU (Q 模式)		C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU	
独立模式	1)	串行通信	2)*1	×	2)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActATCommand	空余	用户任意	
ActATCommandPasswordCancelRetryTimes	3	用户任意	
ActATCommandResponseWaitTime	1	用户任意（单位：秒）	
ActBaudRate	19200(BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200 (根据 LJ71C24 模块的设置)	
ActCallbackCancelWaitTime	90	用户任意（单位：秒）	
ActCallbackDelayTime	20	用户任意（单位：秒）	
ActCallbackNumber	空余	用户任意的电话号码	
ActCallbackReceptionWaitingTimeOut	120	用户任意（单位：秒）	
ActConnectionCDWaitTime	90	用户任意（单位：秒）	
ActConnectionModemReportWaitTime	5	用户任意（单位：秒）	
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号	

属性	默认值	属性的模式	
		1)	2)
ActConnectWay	0 (TEL_AUTO_CONNECT)	TEL_AUTO_CONNECT(0)、TEL_AUTO_CALLBACK(1)、 TEL_AUTO_CALLBACK_NUMBER(2)、TEL_CALLBACK(3)、 TEL_CALLBACK_NUMBER(4)、TEL_CALLBACK_REQUEST(5)、 TEL_CALLBACK_REQUEST_NUMBER(6)、TEL_CALLBACK_WAIT(7)	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型	
ActDestinationIONumber	0 (0x00)	固定为0 (0x00)	对象站侧 单CPU时 固定为1023 (0x3FF) 多CPU时 连接CPU机号： 1023 (0x3FF) 1号CPU: 992 (0x3E0) 2号CPU: 993 (0x3E1) 3号CPU: 994 (0x3E2) 4号CPU: 995 (0x3E3) 冗余CPU时 控制系统： 976 (0x3D0) 无指定： 1023 (0x3FF)
ActDialNumber	空余	用户任意的电话号码	
ActDidPropertyBit	1 (0x01)	固定为1 (0x01)	固定为0 (0x00)
ActDisconnectionCDWaitTime	5	用户任意(单位：秒)	
ActDisconnectionDelayTime	3	用户任意(单位：秒)	
ActDsidPropertyBit	1 (0x01)	固定为1 (0x01)	固定为0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为0 (0x00)	
ActIONumber*1	1023 (0x3FF)	固定为1023 (0x3FF)	连接站侧经由模块I/O地址
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0)、LINETYPE_TONE(1)、LINETYPE_ISDN(2)	
ActMultiDropChannelNumber*2	0 (0x00)	固定为0 (0x00)	多点通道No.
ActNetworkNumber	0 (0x00)	固定为0 (0x00)	
ActOutsideLineNumber	空余	用户任意的外线发送编号	
ActPassword	空余	模块中设置的口令	
ActPasswordCancelResponseWaitTime	5	用户任意(单位：秒)	
ActPortNumber	1 (PORT_1)	计算机侧COM端口No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIALMODEM(0x0E)	
ActStationNumber	255 (0xFF)	固定为255 (0xFF)	
ActThroughNetworkType	0 (0x00)	不包含MELSECNET/10 : 0 (0x00) 包含MELSECNET/10 : 1 (0x01)	
ActTimeOut	10000	以ms为单位由用户任意设置	
ActTransmissionDelayTime	0	用户任意(单位：秒)	
ActUnitNumber	0 (0x00)	固定为0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)	

*1: 对于I/O地址，应指定将实际I/O起始No.用16相除后的值。

*2: 对多点链接的通道No.应进行下述指定。

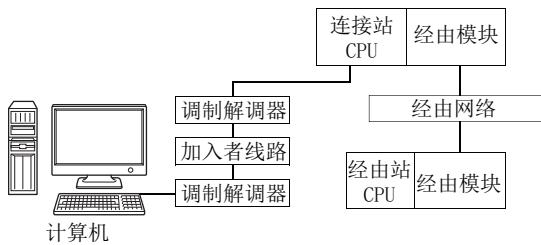
0: 使用模块的默认通道

1: 通道1

2: 通道2

4.14.3 连接站为 FXCPU 的调制解调器通信

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field						
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1)	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

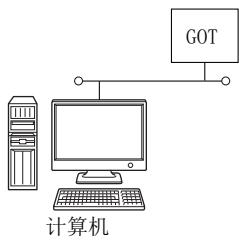
(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActATCommand	空余	用户任意
ActATCommandPasswordCancelRetryTimes	3	用户任意
ActATCommandResponseWaitTime	1	用户任意(单位:秒)
ActCallbackCancelWaitTime	90	用户任意(单位:秒)
ActCallbackDelayTime	20	用户任意(单位:秒)
ActCallbackNumber	空余	用户任意的电话号码
ActCallbackReceptionWaitingTimeOut	120	用户任意(单位:秒)
ActConnectionCDWaitTime	90	用户任意(单位:秒)
ActConnectionModemReportWaitTime	5	用户任意(单位:秒)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型
ActDialNumber	空余	用户任意的电话号码
ActDisconnectionCDWaitTime	5	用户任意(单位:秒)
ActDisconnectionDelayTime	3	用户任意(单位:秒)
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0)、LINETYPE_TONE(1)、 LINETYPE_ISDN(2)
ActOutsideLineNumber	空余	用户任意的外线发送编号
ActPasswordCancelResponseWaitTime	5	用户任意(单位:秒)
ActPortNumber	1 (PORT_1)	计算机侧COM端口No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TEL (0x0A)
ActTimeOut	10000	以ms为单位由用户任意设置
ActTransmissionDelayTime	0	用户任意(单位:秒)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)

4.15 网关功能通信

(1) 构成



(2) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActHostAddress	1.1.1.1	连接的 GOT 的主机名或 IP 地址
ActPortNumber	1 (PORT_1)	计算机侧端口 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPPIP (0x08)
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_A900GOT (0x21)

4.16 GOT 透明通信

要点

关于可使用的系统配置，请参阅以下手册。

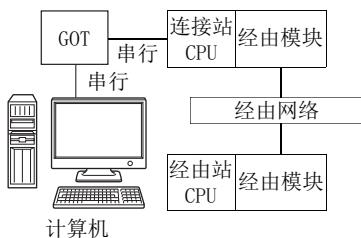
- GOT2000 系列的各连接手册
(三菱电机机器连接篇、其他公司机器连接篇 1、其他公司机器连接篇 2、微型计算机 /MODBUS/ 周边机器连接篇)
- GOT1000 系列的各连接手册
(三菱电机机器连接篇、其他公司机器连接篇 1、其他公司机器连接篇 2、微型计算机 /MODBUS/ 周边机器连接篇)

4

4.16.1 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：直接连接

(1) 连接站为 QCPU(Q 模式) 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU			经由网络	经由站 CPU	
QCPU (Q 模式)	Q 运动 CPU	C 语言控制 器		RCPU	R 运动 CPU
1)*1	1)*2	1)*5	CC IE Control	×	×
			CC IE Field	×	×
			MELSECNET/H	×	×
			以太网	×	×
			串行通信	×	×
			CC-Link	×	×

连接站 CPU			经由网络	经由站 CPU					
QCPU (Q 模式)	Q 运动 CPU	C 语言控制 器		QCPU (Q 模式)*1	C 语 言 控 制 器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)*1	1)*2	1)*5	CC IE Control	2)	2)*3	2)*4	×	2)*3	×
			CC IE Field	2)	2)	×	×	2)	×
			MELSECNET/H	2)	2)	×	×	2)	×
			以太网	2)	×	×	×	2)	×
			串行通信	3)	×	3)	×	3)	×
			CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

- *1: 冗余 CPU 的情况下不能进行访问。
- *2: 不能经由 Q 运动 CPU 访问经由站。
- *3: 对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、Q 运动 CPU, 由于不支持 CC-Link IE 现场网络, 因此不能访问 CC-Link IE 现场网络。
- *4: 由于 LCPU 不支持 CC-Link IE 控制网络, 因此不能访问 CC-Link IE 控制网络。
- *5: 多 CPU 构成时, 仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIOPortNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPortNumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

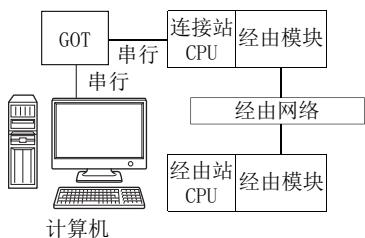
*2: 对多点链接的通道 No. 应进行下述指定。

- 0: 使用模块的默认通道
- 1: 通道 1
- 2: 通道 2

4.16 GOT 透明通信
4.16.1 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：直接连接

(2) 连接站为 LCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field *2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field *2	4)	4) ^{*3}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：由于 Q12DCCPU-V(基本功能模式)不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式				
		1)	2)	3)	4)	
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200				
ActControl	8 (TRC_DTR_OR_RTS)	8 (TRC_DTR_OR_RTS)				
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型				
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)	
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)	
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)	
ActIOPumber ^{*1}	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	
ActMultiDropChannelNumber ^{*2}	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)	
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.				
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)				
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)				

4.16 GOT 透明通信
4.16.1 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：直接连接

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPU (0x50)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

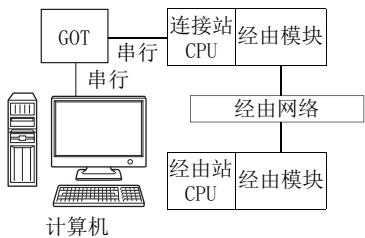
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(3) 连接站为 FXCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) FXCPU	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) FXCPU	CC-Link	×	×				
	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	2)*1

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3G(C)CPU、FX3U(C)CPU 可以访问。

4.16 GOT 透明通信
4.16.1 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：直接连接

(c) 属性一览

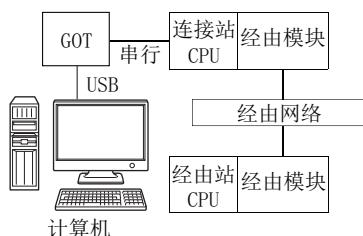
各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、 BAUDRATE_38400、BAUDRATE_57600、 BAUDRATE_115200	
ActControl	8 (TRC_DTR_OR_RTS)	8 (TRC_DTR_OR_RTS)	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActDestinationI0Number	0 (0x00)	固定为 0 (0x00)	固定为 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 0 (0x00)	固定为 1 (0x01)
ActI0Number	1023 (0x3FF)	固定为 0 (0x00)	连接站模块编号
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)	

4.16.2 计算机侧端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：直接连接

(1) 连接站为 QCPU(Q 模式)的情况下

(a) 构成



4

(b) 属性的模式

连接站 CPU			经由网络	经由站 CPU	
QCPU (Q 模式)	Q 运动 CPU	C 语言控制 器		RCPU	R 运动 CPU
1) ^{*1}	1) ^{*2}	1) ^{*5}	CC IE Control	×	×
			CC IE Field	×	×
			MELSECNET/H	×	×
			以太网	×	×
			串行通信	×	×
			CC-Link	×	×

连接站 CPU			经由网络	经由站 CPU					
QCPU (Q 模式)	Q 运动 CPU	C 语言控制 器		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	1) ^{*2}	1) ^{*5}	CC IE Control	2)	2) ^{*3}	2) ^{*4}	×	2) ^{*3}	×
			CC IE Field	2)	2)	×	×	2)	×
			MELSECNET/H	2)	2)	×	×	2)	×
			以太网	2)	×	×	×	2)	×
			串行通信	3)	×	3)	×	3)	×
			CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：不能经由 Q 运动 CPU 访问经由站。

*3：对于 R 运动 CPU、Q12DCCPU-V(基本功能模式)、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5：多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCUP(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

*1：对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2：对多点链接的通道 No. 应进行下述指定。

0：使用模块的默认通道

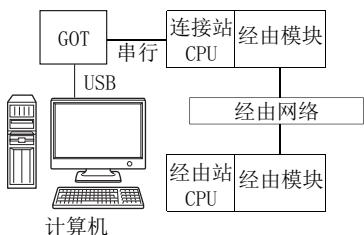
1：通道 1

2：通道 2

4.16 GOT 透明通信
4.16.2 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：直接连接

(2) 连接站为 LCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field *2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field *2	4)	4) ^{*3}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	2)	×
	CC-Link	3)	3)	3)	×	3)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 冗余 CPU 的情况下不能进行访问。

*2: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3: 由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)
ActIOnumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPU (0x50)			

*1：对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2：对多点链接的通道 No. 应进行下述指定。

0：使用模块的默认通道

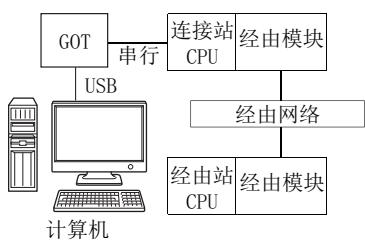
1：通道 1

2：通道 2

4.16 GOT 透明通信
4.16.2 计算机侧端口：USB, GOT2000/I000 侧端口：USB, CPU 侧端口：直接连接

(3) 连接站为 FXCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1)	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	2) ^{*1}

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3G(C)CPU、FX3U(C)CPU 可以访问。

(c) 属性一览

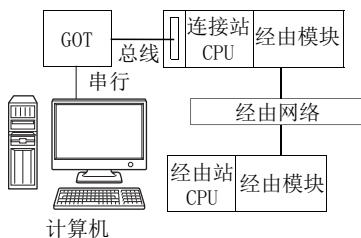
各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActDestinationIONumber	0 (0x00)	0 (0x00)	固定为 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	0 (0x00)	固定为 1 (0x01)
ActIONumber	1023 (0x3FF)	0 (0x00)	连接站模块编号
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	0 (0x00)	对象站模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)	

4.16 GOT 透明通信
4.16.2 计算机侧端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：直接连接

4.16.3 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：总线

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCPCU (Q 模式)	Q 运动 CPU		RCPUCPU	R 运动 CPU
1)*1	1)*2,*5	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)	Q 运动 CPU		QCPCU (Q 模式)*1	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)*1	1)*2,*5	CC IE Control	2)	2)*3	2)*4	×	2)*3	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	2)	×
		以太网	2)	×	×	×	2)	×
		串行通信	3)	×	3)	×	3)	×
		CC-Link	4)	4)	4)	×	4)	×

连接站 CPU C 语言控制器	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1) *6	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU C 语言控制器	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) *6	CC IE Control CC IE Field	2)	2) ^{*3}	2) ^{*4}	×	2) ^{*3}	×
	MELSECNET/H	2)	2)	×	×	2)	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：不能经由 Q 运动 CPU 访问经由站。

*3：对于 Q12DHCCPU-V（基本功能模式）、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5：多 CPU 构成时，仅 2 号机以后可以访问。

*6：由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestination IONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActTimeOut*3	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_A900GOT (0x21)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

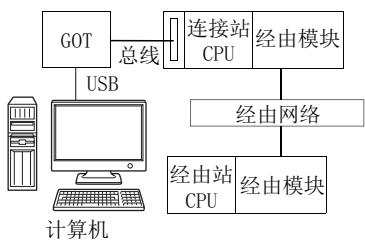
1: 通道 1

2: 通道 2

*3: 指定了 0 ~ 5000[ms] 的情况下，将被化整为 5000ms。此外，设置了大于 255000ms 的值的情况下，将被化整为 255000ms，应加以注意。

4.16.4 计算机侧端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：总线

(1) 构成



(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCPCU (Q 模式) ^{*1}	Q 运动 CPU		RCPUCPU	R 运动 CPU
1) ^{*1}	1) ^{*2,*5}	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式) ^{*1}	Q 运动 CPU		QCPCU (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	1) ^{*2,*5}	CC IE Control	2)	2) ^{*3}	2) ^{*4}	×	2) ^{*3}	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	2)	×
		以太网	2)	×	×	×	2)	×
		串行通信	3)	×	3)	×	3)	×
		CC-Link	4)	4)	4)	×	4)	×

4.16 GOT 透明通信
4.16.4 计算机侧端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：总线

连接站 CPU		经由网络	经由站 CPU					
C 语言控制器			RCPU	R 运动 CPU				
1)	CC IE Control	×	×					
	CC IE Field							
	MELSECNET/H	×	×					
	以太网	×	×					
	串行通信	×	×					
1)	CC-Link	×	×					
连接站 CPU		经由网络	经由站 CPU					
C 语言控制器			QCPU (Q 模式) ^{*1}	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control	2)	2) ^{*3}	2) ^{*4}	×	2) ^{*3}	×	
	CC IE Field							
	MELSECNET/H	2)	2)	×	×	×	×	
	以太网	×	×	×	×	×	×	
	串行通信	×	×	×	×	×	×	
1)	CC-Link	4)	4)	4)	×	×	×	

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：不能经由 Q 运动 CPU 访问经由站。

*3：由于 Q12DCCPU-V(基本功能模式) 及 Q 运动 CPU 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5：多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIOPNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCUP(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCUP(Q 模式), C 语言控制器, LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActTimeOut*3	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_A900GOT (0x21)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

*3: 指定了 0 ~ 5000[ms] 的情况下, 将被化整为 5000ms。此外, 设置了大于 255000ms 的值的情况下, 将被化整为 255000ms, 应加以注意。

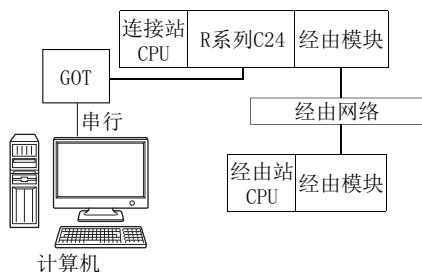
4.16 GOT 透明通信
4.16.4 计算机侧端口 : USB, GOT2000/1000 侧端口 : USB, CPU 侧端口 : 总线

4

4.16.5 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：串行通信模块

(1) 连接站侧除 R 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1) ^{*1}	1) ^{*1, *3}	CC IE Control	2) ^{*2}	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	×
		串行通信	3)	×
		CC-Link	4)	×

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	1) ^{*1, *3}	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1： 不能通过 R 运动 CPU 访问经由站。

*2： 由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3： 多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	与 R 系列 C24 的设置保持一致			
ActConnectUnitNumber	0(0x00)	连接站侧模块编号			
ActControl	8(TRC_DTR_OR_RTS)	根据使用电缆			
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIOnumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActIntelligentPreferenceBit	0(0x00)	固定为 0(0x00)			
ActIOnumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber	0(0x00)	固定为 0(0x00)	对象站侧模块 网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActPortNumber	1(PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL			

4.16 GOT 透明通信
4.16.5 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：串行通信模块

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActPacketType	0x01 (PACKET_PLCL)	PACKET_PLCL			
ActHostAddress	1. 1. 1. 1	固定为 NULL			
ActCpuTimeOut	0(0x00)	固定为 0(0x00)			
ActDataBits	8(DATABIT_8)	固定为 8(0x08)			
ActParity	1(ODD_PARITY)	固定为 1(0x01)			
ActStopBits	0(STOPBIT_ONE)	固定为 0(0x00)			
ActSumCheck	0(NO_SUM_CHECK)	固定为 0(0x00)			
ActSourceNetworkNumber	0(0x00)	固定为 0(0x00)			
ActSourceStationNumber	0(0x00)	固定为 0(0x00)			
ActDestinationPortNumber	0(0x00)	固定为 0(0x00)			
ActIntelligentPreferenceBit	0(0x00)	固定为 0(0x00)			
ActStationNumber	255(0xFF)	固定为 255(0xFF)	对象站侧模块编号	固定为 255(0xFF)	固定为 255(0xFF)
ActThroughNetworkType	0(0x00)	不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71C24			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

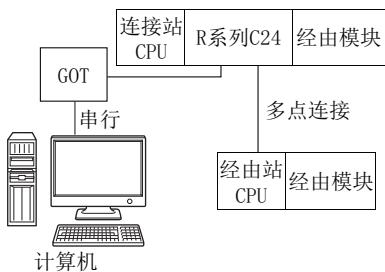
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 R 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
RCPU,	R 运动 CPU *1, *4		RCPU	R 运动 CPU				
独立模式 *2	1)	串行通信	2)	×				
连接站 CPU		经由网络	经由站 CPU					
RCPU,	R 运动 CPU *1, *4		QCPU (Q 模式) ^{*3}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
独立模式 *2	1)	串行通信	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 不能通过 R 运动 CPU 访问经由站。

*2: 独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

*3: 冗余 CPU 的情况下不能进行访问。

*4: 多 CPU 构成时，仅 2 号机以后可以访问。

4.16 GOT 透明通信
4.16.5 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：串行通信模块

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActBaudRate	19200 (BAUDRATE_19200)	与 R 系列 C24 的设置保持一致	
ActConnectUnitNumber	0 (0x00)	连接站侧模块编号	
ActControl	8 (TRC_DTR_OR_RTS)	根据使用电缆	
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型	
ActDestinationIOPNumber *1	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号 :1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	
ActIOPNumber *1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号 :1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber *2	0 (0x00)	固定为 0 (0x00)	多点通道 No.
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	
ActParity	1 (ODD_PARITY)	固定为 ODD_PARITY	
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL	
ActHostAddress	1. 1. 1. 1	固定为 NULL	
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)	
ActDataBits	8 (DATABIT_8)	固定为 8 (0x08)	
ActParity	1 (ODD_PARITY)	固定为 1 (0x01)	
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)	
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)	
ActSourceNetworkNumber	0 (0x00)	固定为 0 (0x00)	
ActSourceStationNumber	0 (0x00)	固定为 0 (0x00)	
ActDestinationPortNumber	0 (0x00)	固定为 0 (0x00)	
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	

属性	默认值	属性的模式	
		1)	2)
ActThroughNetworkType	0(0x00)	不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)	
ActTime0u	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0(0x00)	固定为 0(0x00)	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_RJ71C24

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

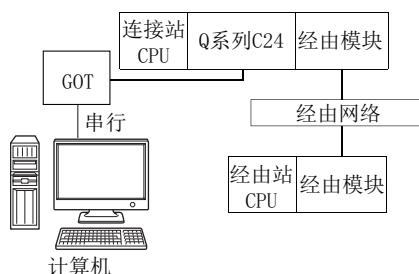
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(3) 连接站侧除 Q 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCPCU (Q 模式)	Q 运动 CPU		RCPU	R 运动 CPU
1)* ¹	1)* ^{2,*5}	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)	Q 运动 CPU		QCPCU (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)* ¹	1)* ^{2,*5}	CC IE Control	2)	2)* ³	2)* ⁴	×	2)* ³	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	2)	×
		以太网	2)	×	×	×	2)	×
		串行通信	3)	×	3)	×	3)	×
		CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 允余 CPU 的情况下不能进行访问。

*2: 不能经由 Q 运动 CPU 访问经由站。

*3: 对于 R 运动 CPU、Q12DCCPU-V (基本功能模式)、Q 运动 CPU，由于不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5: 多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationI0Number	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActI0Number*1	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			

4.16 GOT 透明通信
4.16.5 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：串行通信模块

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)			UNIT_QJ71C24 (0x19)	

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

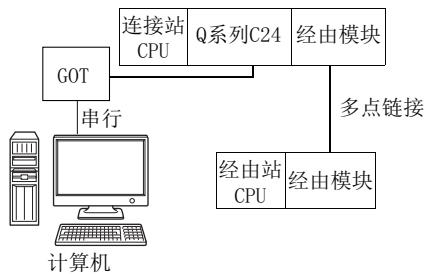
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(4) 通过连接站侧 Q 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式) ^{*1}			RCPCU	R 运动 CPU				
独立模式 ^{*2}	1)	串行通信	×	×				
连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式) ^{*1}			QCPCU (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPCU	Q 运动 CPU	FXCPU
独立模式 ^{*2}	1)	串行通信	2)	×	2)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下，不能访问位于主基板上的串行通信模块。

*2：独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActBaudRate	19200 (BAUDRATE_19200)	Q series BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200	
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号	
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	
ActIONumber*2	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*1	0 (0x00)	固定为 0 (0x00)	多点通道 No.
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	
ActParity	1 (ODD_PARITY)	根据 Q 系列 C24 的设置	
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)	
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)	

*1：对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2：对多点链接的通道 No. 应进行下述指定。

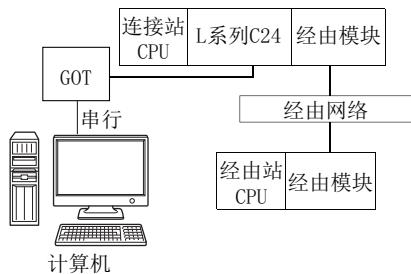
0：使用模块的默认通道

1：通道 1

2：通道 2

(5) 连接站侧除 L 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field*2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)*1	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*2	4)	4)*3	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	2)	×
	CC-Link	3)	3)	3)	×	3)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：由于 Q12DCCPU-V(基本功能模式)不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

4.16 GOT 透明通信
 4.16.5 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：串行通信模块

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号			
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActParity	1 (ODD_PARITY)	固定为 ODD_PARITY			
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)			UNIT_LJ71C24 (0x54)	

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

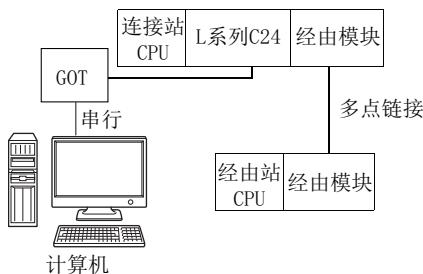
1: 通道 1

2: 通道 2

4.16 GOT 透明通信
4.16.5 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：串行通信模块

(6) 通过连接站侧 L 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
LCPU	RCPU		R 运动 CPU	
独立模式 *2 1)	串行通信		×	×

连接站 CPU		经由网络	经由站 CPU					
LCPU	QCPU (Q 模式) ^{*1}		C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU	
独立模式 *2 1)	串行通信		2)	×	2)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、 BAUDRATE_38400、BAUDRATE_57600、 BAUDRATE_115200	
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号	
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型	
ActDestinationIOPNumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	
ActIOPNumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	
ActParity	1 (ODD_PARITY)	根据 L 系列 C24 的设置	
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)	
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)	

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

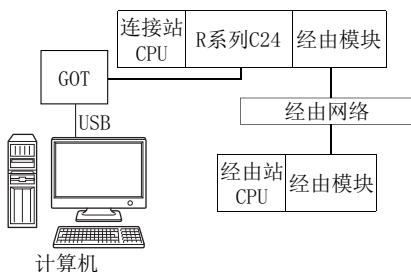
1: 通道 1

2: 通道 2

4.16.6 计算机侧端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：串行通信模块

(1) 连接站侧除 R 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1) ^{*1}	1) ^{*1, *3}	CC IE Control	2) ^{*2}	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	2)
		串行通信	3)	3)
		CC-Link	4)	4)

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	1) ^{*1, *3}	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过 R 运动 CPU 访问经由站。

*2：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActConnectUnitNumber	0(0x00)	连接站侧模块编号			
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActIntelligentPreferenceBit	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	固定为 0(0x00)	固定为 0(0x00)
ActIONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber *4	0(0x00)	固定为 0(0x00)	对象站侧模块网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT			
ActPacket Type	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActPortNumber	1 (PORT_1)	固定为 0(0x00)			
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0(0x00)			

4.16 GOT 透明通信
4.16.6 计算机侧端口 : USB, GOT2000/1000 侧端口 : USB, CPU 侧端口 : 串行通信模块

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActHostAddress	1. 1. 1. 1	固定为 NULL			
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_ OR_RTS)	固定为 0 (0x00)			
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)			
ActParity	1 (ODD_ PARITY)	固定为 0 (0x00)			
ActStopBits	0 (STOPBIT_ ONE)	固定为 0 (0x00)			
ActSumCheck	0 (NO_SUM_ CHECK)	固定为 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	固定为 0 (0x00)			
ActSourceStationNumber	0 (0x00)	固定为 0 (0x00)			
ActDestinationPortNumber	0 (0x00)	固定为 0 (0x00)			
ActStationNumber *4	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块编号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71C24			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“MNET/10 路由信息”。此外, 进行设置时, 对“MNET/10 路由方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

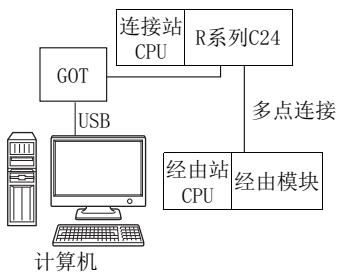
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(2) 通过连接站侧 R 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
RCPU,	R 运动 CPU *1, *4		RCPU	R 运动 CPU				
独立模式 *3	1)	串行通信	2)	×				
连接站 CPU		经由网络	经由站 CPU					
RCPU,	R 运动 CPU *1, *4		QCPU (Q 模式)*2	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
独立模式 *3	1)	串行通信	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过 R 运动 CPU 访问经由站。

*2：冗余 CPU 的情况下不能进行访问。

*3：独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

*4：多 CPU 构成时，仅 2 号机以后可以访问。

4.16 GOT 透明通信
4.16.6 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：串行通信模块

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActConnectUnitNumber	0 (0x00)	连接站侧模块编号	
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型	
ActDestinationIONumber *1	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号 :1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x00)	固定为 1 (0x01)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	
ActIONumber *1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号 :1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧经由模块 I/O 地址
ActMultiDropChannel Number *2	0 (0x00)	固定为 0 (0x00)	多点通道 No.
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT	
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1	
ActPortNumber	1 (PORT_1)	固定为 0 (0x00)	
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0 (0x00)	
ActHostAddress	1. 1. 1. 1	固定为 NULL	
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)	
ActControl	8 (TRC_DTR_OR_RTS)	固定为 0 (0x00)	
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)	
ActParity	1 (ODD_PARITY)	固定为 0 (0x00)	
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)	
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)	
ActSourceNetworkNumber	0 (0x00)	固定为 0 (0x00)	
ActSourceStationNumber	0 (0x00)	固定为 0 (0x00)	
ActDestinationPortNumber	0 (0x00)	固定为 0 (0x00)	
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	

属性	默认值	属性的模式	
		1)	2)
ActThroughNetworkType	0(0x00)	不包含 MELSECNET/10 : 0(0x00) 包含 MELSECNET/10 : 1(0x01)	
ActTimeOut	10000	以 ms 为单位由用户任意设置	
ActUnitNumber	0(0x00)	固定为 0(0x00)	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_RJ71C24

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

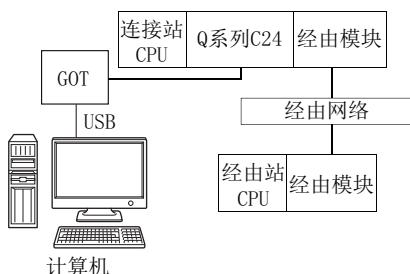
1: 通道 1

2: 通道 2

4.16 GOT 透明通信
4.16.6 计算机侧端口 : USB, GOT2000/1000 侧端口 : USB, CPU 侧端口 : 串行通信模块

(3) 连接站侧除 Q 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCUP (Q 模式)	Q 运动 CPU		RCPU	R 运动 CPU
1)*1	1)*2, *5	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCUP (Q 模式)	Q 运动 CPU		QCUP (Q 模式)*1	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)*1	1)*2, *5	CC IE Control	2)	2)*3	2)*4	×	2)*3	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	2)	×
		以太网	2)	×	×	×	2)	×
		串行通信	3)	×	3)	×	3)	×
		CC-Link	4)	4)	4)	×	4)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 冗余CPU的情况下不能进行访问。

*2: 不能经由Q运动CPU访问经由站。

*3: 对于R运动CPU、Q12DCCPU-V(基本功能模式)、Q运动CPU,由于不支持CC-LinkIE现场网络,因此不能访问CC-LinkIE现场网络。

*4: 由于LCPU不支持CC-LinkIE控制网络,因此不能访问CC-LinkIE控制网络。

*5: 多CPU构成时,仅2号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnit Number	0 (0x00)			连接站侧模块站号	

4.16 GOT透明通信
4.16.6 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：串行通信模块

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

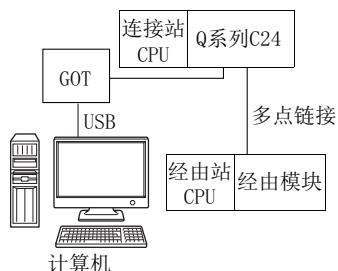
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(4) 通过连接站侧 Q 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
QCPU (Q 模式) ^{*1}			RCPU	R 运动 CPU				
独立模式 ^{*2}	1)	串行通信	×	×				
连接站 CPU		经由网络	经由站 CPU					
QCPU (Q 模式) ^{*1}			QCPU (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
独立模式 ^{*2}	1)	串行通信	2)	×	2)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActConnectUnitNumber	0 (0x00)		连接站侧模块站号
ActCpuType	34 (CPU_Q02CPU)		对应于对象站的 CPU 类型
ActDestinationI0Number	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)		固定为 0 (0x00)
ActI0Number*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.
ActNetworkNumber	0 (0x00)		固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)		PROTOCOL_USBGOT (0x13)
ActStationNumber	255 (0xFF)		固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)
ActTimeOut	10000		以 ms 为单位由用户任意设置
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_QJ71C24 (0x19)

*1：对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2：对多点链接的通道 No. 应进行下述指定。

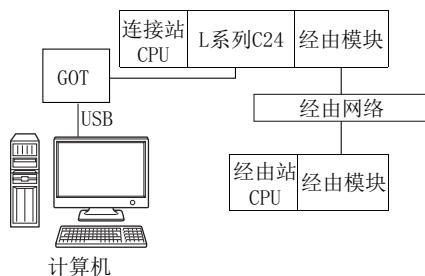
0：使用模块的默认通道

1：通道 1

2：通道 2

(5) 连接站侧除 L 系列 C24 以外存在经由模块的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU
1)	CC IE Field ^{*2}	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	L CPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field ^{*2}	4)	4) ^{*3}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	2)	×
	CC-Link	3)	3)	3)	×	3)	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnitNumber	0 (0x00)	连接站侧模块站号			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActDestinationIOPNumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsdPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)		固定为 0 (0x00)		
ActIOPNumber ^{*1}	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActMultiDropChannelNumber ^{*2}	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)		PROTOCOL_USBGOT (0x13)		
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_LJ71C24 (0x54)		

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

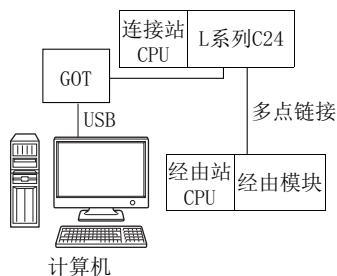
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(6) 通过连接站侧 L 系列 C24 与经由模块进行多点链接的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
LCPU	RCPU		R 运动 CPU	
独立模式 *2	1)	串行通信	×	×

连接站 CPU		经由网络	经由站 CPU					
LCPU	QCPU (Q 模式) ^{*1}		C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU	
独立模式 *2	1)	串行通信	2)	×	2)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：独立模式表示进行了如下参数设置的情况下。

- CH1 侧：传送设置的动作设置 = 独立 (0)，通信协议设置 = 0
- CH2 侧：传送设置的动作设置 = 独立 (0)

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式	
		1)	2)
ActConnectUnitNumber	0 (0x00)		连接站侧模块站号
ActCpuType	34 (CPU_Q02CPU)		对应于对象站的 CPU 类型
ActDestinationI0Number	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)		固定为 0 (0x00)
ActI0Number*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧经由模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.
ActNetworkNumber	0 (0x00)		固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)		PROTOCOL_USBGOT (0x13)
ActStationNumber	255 (0xFF)		固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)
ActTimeOut	10000		以 ms 为单位由用户任意设置
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_LJ71C24 (0x54)

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

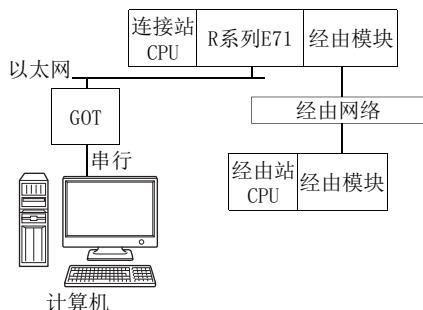
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.16.7 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：R 系列 E71

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control	2)*1	×
	CC IE Field		
	MELSECNET/H	×	×
	以太网	2)	×
	串行通信	3)	×
	CC-Link	4)	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、 BAUDRATE_57600、BAUDRATE_115200			
ActConnectUnitNumber *1	0(0x00)	固定为 0(0x00)	连接站侧模块编号	固定为 0(0x00)	固定为 0(0x00)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActDestinationPortNumber	0(0x00)	5001 OPS 连接时 本站端口 No. *3			
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber *4	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber *5	0(0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站侧 R 系列 E71 网络 No.	连接站侧 R 系列 E71 网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_ OR_RTS)	根据使用电缆			
ActDataBits	8 (DATABIT_8)	固定为 8 (0x08)			
ActParity	1 (ODD_ PARITY)	固定为 1 (0x01)			
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActSourceNetworkNumber *6	0 (0x00)	GOT 侧网络 No.			
ActSourceStationNumber *7	0 (0x00)	GOT 侧站号			
ActStationNumber *5	255 (0xFF)	对象站侧模块编号	对象站侧模块编号	连接站侧 R 系列 E71 站号	连接站侧 R 系列 E71 站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_RJ71EN71			

*1: 属性的模式为 2) 的情况下, 应指定连接站侧 R 系列 E71 的以太网参数中设置的连接站侧 R 系列 E71 的站号。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 冗余 CPU 连接时使用 OPS 连接功能的情况下, 应指定网络参数的“本站端口编号”中设置的端口 No.。

*4: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

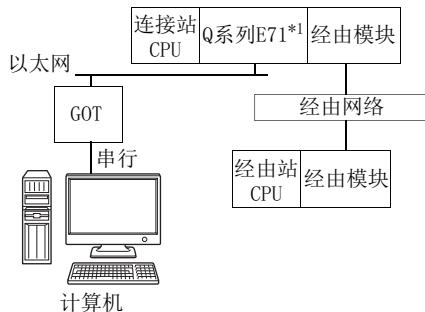
*5: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*6: 在对象站侧 R 系列 E71 的以太网参数设置中, 应指定与 R 系列 E71 中设置的 MELSECNET/10 用网络 No. 相同的网络 No.。

*7: 在指定 GOT 侧的站号时, 应避免与以太网的同一环内设置的 R 系列 E71 的站号重复。

4.16.8 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：Q 系列 E71

(1) 构成



*1: 连接站的 Q 系列 E71 中设置了口令的情况下不能进行通信。

(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)*1	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)*1	CC IE Control	2)	2) ^{*2}	2) ^{*3}	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 冗余 CPU 的情况下不能进行访问。

*2: 由于 Q12DCCPU-V(基本功能模式)不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActConnectUnitNumber	0 (0x00)	固定为 0 (0x00)	连接站侧模块站号	固定为 0 (0x00)	固定为 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	5001			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIONumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*3	0 (0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站 Q 系列 E71 网络 No.	连接站 Q 系列 E71 网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActSourceNetworkNumber	0 (0x00)	GOT 侧网络 No.			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActSource StationNumber ^{*4}	0 (0x00)	GOT 侧站号			
ActStation Number ^{*3}	255 (0xFF)	对象站侧模块站号	对象站侧模块站号	连接站侧 Q 系列 E71 站号	连接站侧 Q 系列 E71 站号
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QJ71E71 (0x40)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

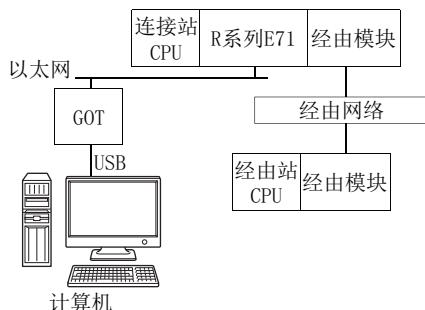
2: 通道 2

*3: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*4: 指定 GOT 侧的站号时, 应避免与以太网的同一环内设置的 Q 系列 E71 的站号重复。

4.16.9 计算机侧端口：USB，GOT2000/1000 侧端口：USB， CPU 侧端口：R 系列 E71

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Control	2)*1	×
	CC IE Field		
	MELSECNET/H	×	×
	以太网	2)	×
	串行通信	3)	×
	CC-Link	4)	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnitNumber *1	0(0x00)	固定为 0(0x00)	连接站侧模块编号	固定为 0(0x00)	固定为 0(0x00)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActDestinationPortNumber	0(0x00)	5001 OPS 连接时 本站端口 No. *3			
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber *4	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber *5	0(0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站侧 R 系列 E71 网络 No.	连接站侧 R 系列 E71 网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT			
ActPacketType	0x01 (PACKET_PLCL)	PACKET_PLCL			
ActPortNumber	1 (PORT_1)	固定为 0(0x00)			

4.16 GOT 透明通信
4.16.9 计算机侧端口 : USB, GOT2000/1000 侧端口 : USB, CPU 侧端口 : R 系列 E71

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0 (0x00)			
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	固定为 0 (0x00)			
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)			
ActParity	1 (ODD_PARITY)	固定为 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActSourceNetworkNumber *6	0 (0x00)	GOT 侧网络 No.			
ActSourceStationNumber *7	0 (0x00)	GOT 侧站号			
ActStationNumber *5	255 (0xFF)	对象站侧模块编号	对象站侧模块编号	连接站侧 R 系列 E71 站号	连接站侧 R 系列 E71 站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0x00	固定为 0x00	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_RJ71EN71			

*1: 属性的模式为 2) 的情况下, 应指定连接站侧 R 系列 E71 的以太网参数中设置的连接站侧 R 系列 E71 的站号。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 冗余 CPU 连接时使用 OPS 连接功能的情况下, 应指定网络参数的“本站端口编号”中设置的端口 No.。

*4: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

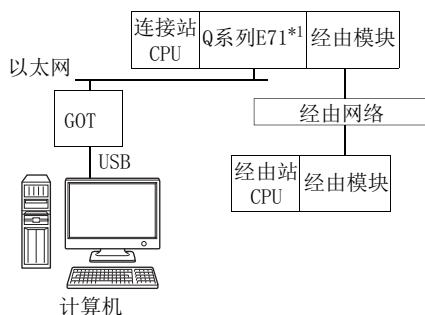
*5: 属性的模式为 1) 或 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*6: 在对象站侧 R 系列 E71 的以太网参数设置中, 应指定与 R 系列 E71 中设置的 MELSECNET/10 用网络 No. 相同的网络 No.。

*7: 在指定 GOT 侧的站号时, 应避免与以太网的同一环内设置的 R 系列 E71 的站号重复。

4.16.10 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：Q 系列 E71

(1) 构成



*1: 连接站的 Q 系列 E71 中设置了远程口令的情况下，不能进行通信。

(2) 属性的模式

连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)	C 语言控制器		RCPU	R 运动 CPU				
1)*1	1)*4	CC IE Control	×	×				
		CC IE Field						
		MELSECNET/H	×	×				
		以太网	×	×				
		串行通信	×	×				
连接站 CPU		经由网络	经由站 CPU					
QCPCU (Q 模式)	C 语言控制器		QCPCU (Q 模式)*1	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)*1	1)*4	CC IE Control	2)	2)*2	2)*3	×	×	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	×	×
		以太网	2)	×	×	×	×	×
		串行通信	3)	×	3)	×	×	×
连接站 CPU		CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 冗余 CPU 的情况下不能进行访问。

*2: 由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4: 多 CPU 构成时，仅 2 号机以后可以访问。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnitNumber	0 (0x00)	固定为 0 (0x00)	连接站侧模块站号	固定为 0 (0x00)	固定为 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	5001			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIONumber ^{*1}	1023 (0x3FF)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber ^{*2}	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber ^{*3}	0 (0x00)	对象站侧模块网络 No.	对象站侧模块网络 No.	连接站 Q 系列 E71 网络 No.	连接站 Q 系列 E71 网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActSourceNetworkNumber	0 (0x00)	GOT 侧网络 No.			
ActSourceStationNumber ^{*4}	0 (0x00)	GOT 侧站号			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActStationNumber ^{*3}	255 (0xFF)	对象站侧模块站号	对象站侧模块站号	连接站侧 Q 系列 E71 站号	连接站侧 Q 系列 E71 站号
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_GOT_QJ71E71 (0x40)		

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

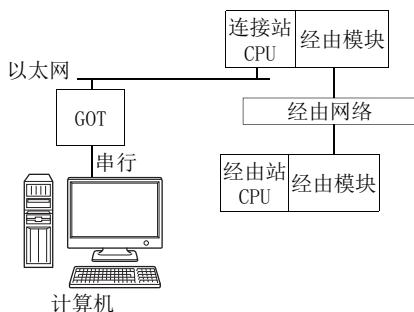
*3: 属性的模式为 1) 或 2) 的情况下，对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

*4: 指定 GOT 侧的站号时，应避免与以太网的同一环内设置的 Q 系列 E71 的站号重复。

4.16.11 计算机侧端口：串行，GOT2000/1000 侧端口：串行， CPU 侧端口：以太网端口

(1) 连接站为 RCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1)*1,*3	CC IE Control	2)*2	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	×
		串行通信	3)	×
		CC-Link	4)	×

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1)*1,*3	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过 R 运动 CPU 访问经由站。

*2：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
ActDestinationIONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActDidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsidPropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站 RCPU、 QCUP(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 RCPU、 QCUP(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetworkNumber *4	0(0x00)	固定为 0(0x00)	对象站侧模块网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActPortNumber	1(PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL			

4.16 GOT 透明通信
4.16.11 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网端口

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActPacketType	0x01 (PACKET_PLCL)			PACKET_PLCL	
ActConnectUnitNumber	0 (0x00)			固定为 0 (0x00)	
ActCpuTimeOut	0 (0x00)			固定为 0 (0x00)	
ActControl	8 (TRC_DTR_OR_RTS)			根据使用电缆	
ActDataBits	8 (DATABIT_8)			固定为 8 (0x08)	
ActParity	1 (ODD_PARITY)			固定为 1 (0x01)	
ActStopBits	0 (STOPBIT_ONE)			固定为 0 (0x00)	
ActSumCheck	0 (NO_SUM_CHECK)			固定为 0 (0x00)	
ActSourceNetworkNumber	0 (0x00)			固定为 0 (0x00)	
ActSourceStationNumber	0 (0x00)			固定为 0 (0x00)	
ActDestinationPortNumber	0 (0x00)			5006	
ActStationNumber *4	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块编号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)			不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)	
ActTimeOut	10000			以 ms 为单位由用户任意设置	
ActUnitNumber	0 (0x00)	固定为 0x00	固定为 0x00	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)			UNIT_GOT_RETHER	

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“MNET/10 路由信息”。此外, 进行设置时, 对“MNET/10 路由方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

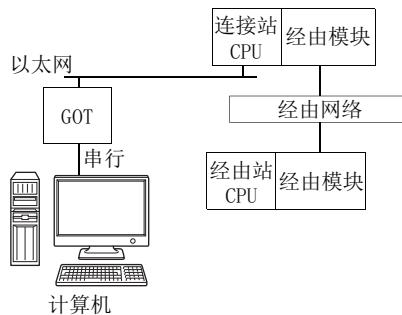
*3: 对多点链接的通道 No. 应进行下述指定。

- 0: 使用模块的默认通道
- 1: 通道 1
- 2: 通道 2

*4: 属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

(2) 连接站为 QnUDE(H) CPU 或 C 语言控制器的情况下

(a) 构成



(b) 属性的模式

连接站 CPU*1	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) QnUDE(H) CPU	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) QnUDE(H) CPU	CC-Link	×	×				
连接站 CPU*1	经由网络	经由站 CPU					
		QCUP (Q 模式)*2	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	2)	2)*3	2)*4	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

4.16 GOT 透明通信
4.16.11 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网端口

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
C 语言控制器			
1) ^{*5}	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*2}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
C 语言控制器							
1) ^{*5}	CC IE Control CC IE Field	2)	2) ^{*3}	2) ^{*4}	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：连接站 CPU 中设置了远程口令的情况下，不能进行通信。

*2：冗余 CPU 的情况下不能进行访问。

*3：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*5：由于 Q24DHCCPU-V、Q24DHCCPU-LS 不支持通信路径，因此不能访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1. 1. 1. 1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIONumber*2	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*4	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber*4	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			

4.16 GOT 透明通信
4.16.11 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网端口

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QNETHER (0x41)			

*1: 经由以太网模块 (Q 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 Q 系列 E71 的参数设置中设置的值。
- 应设置 Q 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

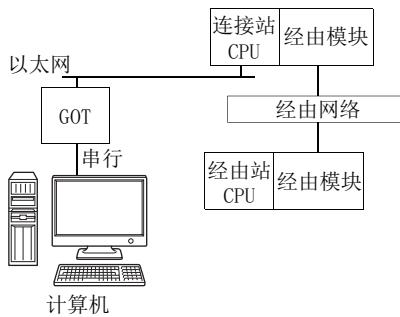
1: 通道 1

2: 通道 2

*4: 属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

(3) 连接站为 LCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU*1	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field*3	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU*1	经由网络	经由站 CPU					
		QCPU (Q 模式)*2	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*3	4)	4)*4	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：连接站 CPU 中设置了远程口令的情况下，不能进行通信。

*2：冗余 CPU 的情况下不能进行访问。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

4.16 GOT 透明通信
4.16.11 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网端口

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActControl	8 (TRC_DTR_OR_RTS)	根据使用的电缆			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_LNETHER (0x55)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

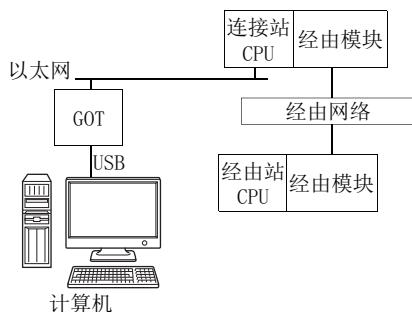
2: 通道 2

4.16 GOT 透明通信
4.16.11 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网端口

4.16.12 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网端口

(1) 连接站为 RCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1) ^{*1, *3}	CC IE Control	2) ^{*2}	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	2)	×
		串行通信	3)	×
		CC-Link	4)	×

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1) ^{*1, *3}	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：不能通过 R 运动 CPU 访问经由站。

*2：由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
Act Destination IONumber *2	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
ActDid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHost Address	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligent PreferenceBit	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU、 RCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU、 RCPU : 1(0x01) 除上述以外 : 0(0x00)
ActIONumber *2	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDrop Channel Number *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetwork Number *4	0(0x00)	固定为 0(0x00)	对象站侧模块网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActProtocol Type	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT			
ActPacket Type	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActConnect UnitNumber	0(0x00)	固定为 0(0x00)			
ActPortNumber	1(PORT_1)	固定为 0(0x00)			

4.16 GOT 透明通信
4.16.12 计算机侧端口 : USB, GOT2000/1000 侧端口 : USB, CPU 侧端口 : 以太网端口

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0 (0x00)			
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	固定为 0 (0x00)			
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)			
ActParity	1 (ODD_PARITY)	固定为 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	固定为 0 (0x00)			
ActSourceStationNumber	0 (0x00)	固定为 0 (0x00)			
ActDestinationPortNumber	0 (0x00)	5006			
ActStationNumber *4	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块编号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0x00	固定为 0x00	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_RETHER			

*1: 经由以太网模块 (R 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 R 系列 E71 的参数设置中设置的值。
- 应设置 R 系列 E71 参数设置的“MNET/10 路由信息”。此外, 进行设置时, 对“MNET/10 路由方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*3: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

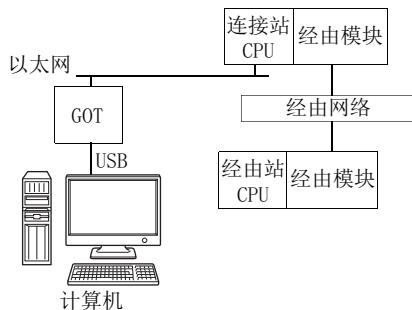
1: 通道 1

2: 通道 2

*4: 属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

(2) 连接站为 QCPU(Q 模式) 或 C 语言控制器的情况下

(a) 构成



(b) 属性的模式

连接站 CPU*1	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1)	CC-Link	×	×				
连接站 CPU*1	经由网络	经由站 CPU					
		QCPU (Q 模式)*2	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	2)	2)*3	2)*4	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

4.16 GOT 透明通信
4.16.12 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网端口

连接站 CPU*1	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
2)	CC-Link	×	×				
连接站 CPU*1	经由网络	经由站 CPU					
		QCPU (Q 模式)*2	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Control	2)	2)*3	2)*4	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
4)	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：连接站 CPU 中设置了远程口令的情况下，不能进行通信。

*2：冗余 CPU 的情况下不能进行访问。

*3：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*4：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)*1	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPumber*2	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*3	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*4	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber*4	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号

4.16 GOT 透明通信
4.16.12 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网端口

属性	默认值	属性的模式			
		1)	2) ^{*1}	3)	4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QNETHER (0x41)			

*1: 经由以太网模块 (Q 系列 E71) 访问时, 应注意下述事项。

- 对于 ActNetworkNumber、ActStationNumber, 应指定对象站侧的 Q 系列 E71 的参数设置中设置的值。
- 应设置 Q 系列 E71 参数设置的“站号↔IP 关联信息”。此外, 进行设置时, 对“站号↔IP 关联信息设置方式”应指定除自动响应方式以外 (IP 地址计算方式、表转换方式、并用方式之一)。

*2: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

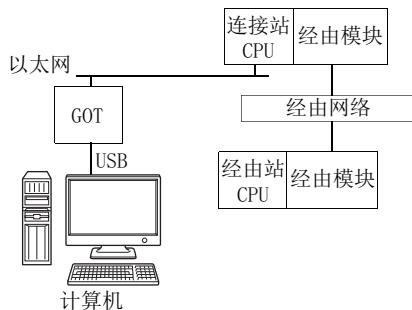
*3: 对多点链接的通道 No. 应进行下述指定。

- 0: 使用模块的默认通道
1: 通道 1
2: 通道 2

*4: 属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

(3) 连接站为 LCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU ^{*1} LCPU	经由网络	经由站 CPU	
		R CPU	R 运动 CPU
1)	CC IE Field ^{*3}	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU ^{*1} LCPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*2}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field ^{*3}	4)	4) ^{*4}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：连接站 CPU 中设置了远程口令的情况下，不能进行通信。

*2：冗余 CPU 的情况下不能进行访问。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4：由于 Q12DCCPU-V（基本功能模式）不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)			对应于对象站的 CPU 类型	

4.16 GOT 透明通信
4.16.12 计算机侧端口：USB, GOT2000/1000 侧端口：USB, CPU 侧端口：以太网端口

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActHostAddress	1.1.1.1	连接站侧模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_LNETHER (0x55)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

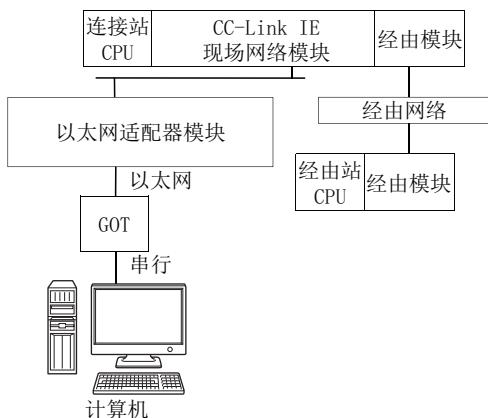
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.16.13 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网适配器模块

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1)	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	2)	2) ^{*2}	2) ^{*3}	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

4.16 GOT 透明通信
4.16.13 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网适配器模块

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field ^{*3}	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field ^{*3}	2)	2) ^{*2}	2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、BAUDRATE_38400、BAUDRATE_57600、BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIOPNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	以太网适配器模块的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIOPNumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber*3	0 (0x00)	连接站侧 CC-Link IE 现场网络模块网络 No.	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块网络 No.	连接站侧 CC-Link IE 现场网络模块网络 No.
ActPortNumber	1 (PORT_1)	计算机侧 COM 端口 No.			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActStationNumber ^{*3}	255 (0xFF)	连接站侧 CC-Link IE 现场网络模块站号	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块站号	连接站侧 CC-Link IE 现场网络模块站号
ActThroughNetworkType	0 (0x00)		不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)		
ActTimeOut	10000		以 ms 为单位由用户任意设置		
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_GOT_NZ2GF_ETB (0x5B)		

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

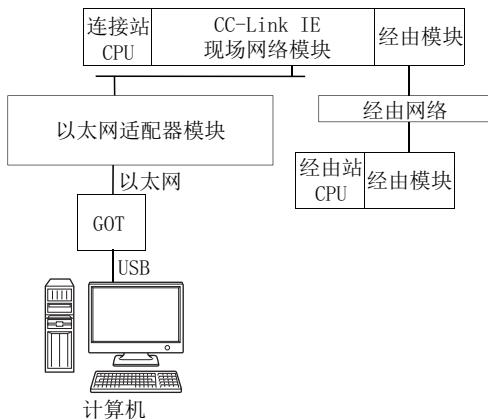
2: 通道 2

*3: 属性的模式为 1) 3) 4) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定连接站侧的 CC-Link IE 现场网络模块参数中设置的值。

属性的模式为 2) 的情况下, 对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.16.14 计算机端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：以太网适配器模块

(1) 构成



(2) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) QnUDE (H) CPU	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) QnUDE (H) CPU	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCUP (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	2)	2) ^{*2}	2) ^{*3}	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	2)	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

4.16 GOT 透明通信
4.16.14 计算机端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：以太网适配器模块

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field ^{*3}	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field ^{*3}	2)	×	2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestination IONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidProperty Bit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1. 1. 1. 1	以太网适配器模块的主机名或 IP 地址			
ActIntelligent PreferenceBit	0 (0x00)	固定为 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDrop ChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetwork Number*3	0 (0x00)	连接站侧 CC-Link IE 现场网络模块网络 No.	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块网络 No.	连接站侧 CC-Link IE 现场网络模块网络 No.
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStation Number*3	255 (0xFF)	连接站侧 CC-Link IE 现场网络模块站号	对象站侧模块站号	连接站侧 CC-Link IE 现场网络模块站号	连接站侧 CC-Link IE 现场网络模块站号
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_NZ2GF_ETB (0x5B)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

1: 通道 1

2: 通道 2

*3: 属性的模式为 1)3)4) 的情况下，对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧 CC-Link IE 现场网络模块参数中设置的值。

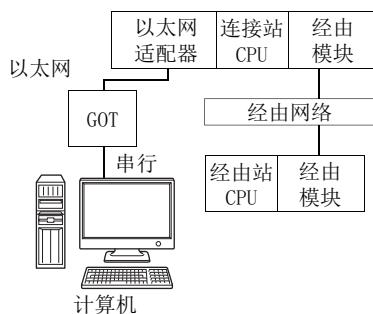
属性的模式为 2) 的情况下，对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

4.16.15 计算机侧端口：串行，GOT2000/1000 侧端口：串行，CPU 侧端口：以太网适配器 / 模块

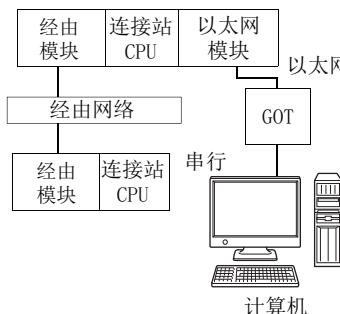
(1) 连接站为 FXCPU 的情况下

(a) 构成

以太网适配器：FX3U-ENET-ADP 的情况下



以太网模块：FX3U-ENET(-L) 的情况下



4

(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1) ^{*1}	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1) ^{*1}	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3sCPU(FX3U-ENET-ADP)、FX3G(C)CPU、FX3U(C)CPU 可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

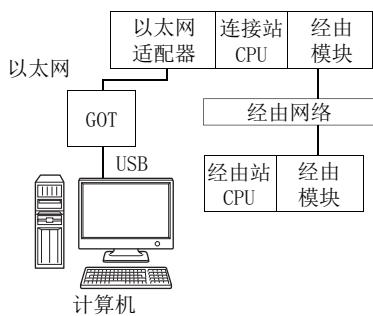
属性	默认值	属性的模式
		1)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600、BAUDRATE_19200、 BAUDRATE_38400、BAUDRATE_57600、 BAUDRATE_115200
ActControl	8 (TRC_DTR_OR_RTS)	根据使用电缆
ActCpuType	34(CPU_Q02CPU)	对应于对象站的 CPU 类型
ActHostAddress	1.1.1.1	FX3U-ENET-ADP 模块的主机名或 IP 地址 FX3U-ENET(-L) 模块的主机名或 IP 地址
ActPortNumber	1(PORT_1)	计算机侧 COM 端口 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitType	0x13 (UNIT_QNCPU)	FX3U-ENET-ADP: UNIT_GOT_FXETHER FX3U-ENET(-L): UNIT_GOT_FXENET

4.16.16 计算机端口：USB，GOT2000/1000 侧端口：USB，CPU 侧端口：以太网适配器 / 模块

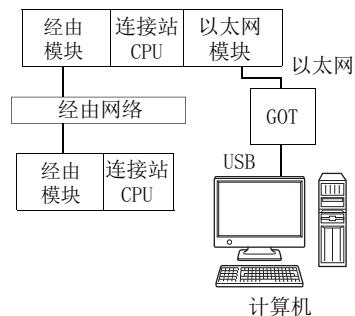
(1) 连接站为 FXCPU 的情况下

(a) 构成

以太网适配器：FX3U-ENET-ADP 的情况下



以太网模块：FX3U-ENET(-L) 的情况下



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1) ^{*1}	CC IE Control	×	×
	CC IE Field		
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	CC IE Control	×	×	×	×	×	×
	CC IE Field						
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：仅 FX3sCPU(FX3U-ENET-ADP)、FX3G(C)CPU、FX3U(C)CPU 可以访问。

(c) 属性一览

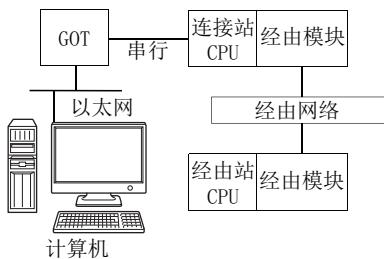
各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型
ActHostAddress	1.1.1.1	FX3U-ENET-ADP 模块的主机名或 IP 地址
		FX3U-ENET(-L) 模块的主机名或 IP 地址
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitType	0x13 (UNIT_QNCPU)	FX3U-ENET-ADP: UNIT_GOT_FXETHER
		FX3U-ENET(-L): UNIT_GOT_FXENET

4.16.17 计算机侧端口：以太网卡，GOT2000/1000 侧端口：以太网端口，CPU 侧端口：串行

(1) 连接站为 QCPU(Q 模式) 或 C 语言控制器的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCPU(Q 模式)	C 语言控制器		RCPU	R 运动 CPU
1)*1	1)*4	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCPU(Q 模式)	C 语言控制器		QCPU (Q 模式)*1	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)*1	1)*4	CC IE Control	2)	2)*2	2)*3	×	×	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	3)	3)	3)	×	×	×
		CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 Q12DHCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*4：Q24DHCCPU-V、Q24DHCCPU-LS 多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	GOT 的端口编号			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	GOT 的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIONumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QNCPU (0x56)			

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

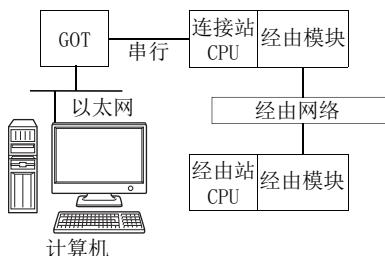
1: 通道 1

2: 通道 2

4.16 GOT 透明通信
4.17 计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：串行

(2) 连接站为 LCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field*2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCUP (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*2	4)	4) ^{*3}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：由于 Q12DCCPU-V（基本功能模式）不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDestinationPortNumber	0 (0x00)	GOT 的端口编号			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)
ActHostAddress	1.1.1.1	GOT 的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)
ActIONumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			

4

4.16 GOT 透明通信
4.16.17 计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：串行

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_LNCPU (0x57)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

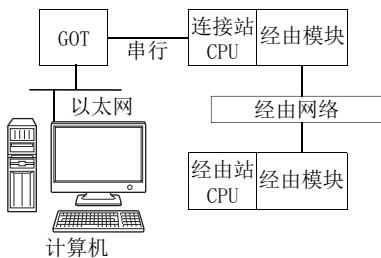
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

(3) 连接站为 FXCPU 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU					
		RCPU	R 运动 CPU				
1)	CC IE Control	×	×				
	CC IE Field	×	×				
	MELSECNET/H	×	×				
	以太网	×	×				
	串行通信	×	×				
1)	CC-Link	×	×				
连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
	CC IE Control	×	×	×	×	×	×
	CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

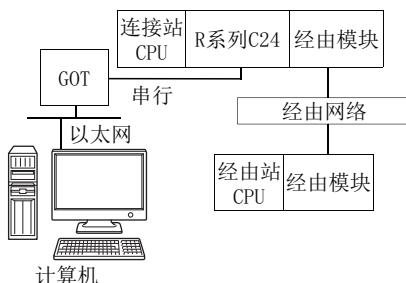
属性	默认值	属性的模式
		1)
ActCpuType	34(CPU_Q02CPU)	对应于对象站的 CPU 类型
ActDestinationPortNumber ^{*1}	0(0x00)	GOT 的端口编号
ActHostAddress	1.1.1.1	GOT 的主机名或 IP 地址
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP
ActTimeOut	10000	以 ms 为单位由用户任意设置
ActUnitType	0x13 (UNIT_GOTETHER_FXCPU)	UNIT_GOTETHER_FXCPU

*1：对于 ActPortNumber 应指定 GOT 侧设置的值。

4.16.18 计算机侧端口：以太网卡，GOT2000/1000 侧端口：以太网端口，CPU 侧端口：串行通信模块

(1) 连接站侧为 R 系列 C24 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
RCPU	R 运动 CPU		RCPU	R 运动 CPU
1)	1) ^{*1,*3}	CC IE Control	4) ^{*2}	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	2)	×
		CC-Link	3)	×

连接站 CPU		经由网络	经由站 CPU					
RCPU	R 运动 CPU		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	1) ^{*1,*3}	CC IE Control	×	×	×	×	×	×
		CC IE Field						
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 不能通过 R 运动 CPU 访问经由站。

*2: 由于 RCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3: 多 CPU 构成时，仅 2 号机以后可以访问。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	与对象站对应的 CPU 类型			
Act Destination IONumber *1	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)
Act Destination PortNumber *2	0(0x00)	GOT 的端口编号			
ActDid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActDsid PropertyBit	1(0x00)	固定为 1(0x01)	固定为 1(0x01)	固定为 0(0x00)	固定为 0(0x00)
ActHost Address	1. 1. 1. 1	GOT 的主机名或 IP 地址			
ActIntelligent PreferenceBit	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU、 RCPU : 1(0x01) 除上述以外 : 0(0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU、 RCPU : 1(0x01) 除上述以外 : 0(0x00)
ActINumber *1	1023(0x3FF)	单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 : 1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	对象站侧 单 CPU 时 固定为 1023(0x3FF) 多 CPU 时 连接 CPU 机号 :1023(0x3FF) 1 号 CPU: 992(0x3E0) 2 号 CPU: 993(0x3E1) 3 号 CPU: 994(0x3E2) 4 号 CPU: 995(0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDrop Channel Number *3	0(0x00)	固定为 0(0x00)	固定为 0(0x00)	多点通道 No.	固定为 0(0x00)
ActNetwork Number *4	0(0x00)	固定为 0(0x00)	对象站侧模块网络 No.	固定为 0(0x00)	固定为 0(0x00)
ActProtocol Type	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP			
ActPacket Type	0x01 (PACKET_ PLC1)	PACKET_PLC1			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActConnectUnitNumber	0 (0x00)	固定为 0 (0x00)			
ActPortNumber	1 (PORT_1)	固定为 0 (0x00)			
ActBaudRate	19200 (BAUDRATE_19200)	固定为 0 (0x00)			
ActCpuTimeOut	0 (0x00)	固定为 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	固定为 0 (0x00)			
ActDataBits	8 (DATABIT_8)	固定为 0 (0x00)			
ActParity	1 (ODD_PARITY)	固定为 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	固定为 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	固定为 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	固定为 0 (0x00)			
ActSourceStationNumber	0 (0x00)	固定为 0 (0x00)			
ActStationNumber *4	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块编号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			
ActUnitNumber	0 (0x00)	固定为 0x00	固定为 0x00	对象站侧模块编号	对象站侧模块编号
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_RJ71C24			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: ActDestinationPortNumber 请指定为在 GOT 侧设置的值。

*3: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

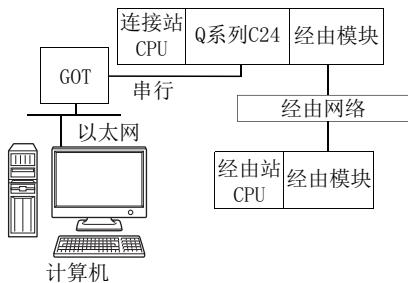
1: 通道 1

2: 通道 2

*4: 属性的模式为 2) 的情况下，对 ActNetworkNumber 及 ActStationNumber 应指定对象站侧参数中设置的值。

(2) 连接站为 Q 系列 C24 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU		经由网络	经由站 CPU	
QCPCU(Q 模式)	C 语言控制器		RCPU	R 运动 CPU
1) ^{*1}	1)	CC IE Control	×	×
		CC IE Field		
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站 CPU		经由网络	经由站 CPU					
QCPCU(Q 模式)	C 语言控制器		QCPCU (Q 模式) ^{*1}	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	1)	CC IE Control	2)	2) ^{*2}	2) ^{*3}	×	×	×
		CC IE Field						
		MELSECNET/H	2)	2)	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	3)	3)	3)	×	×	×
		CC-Link	4)	4	4	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1: 允余 CPU 的情况下不能进行访问。

*2: 由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3: 由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

4, 16 GOT 透明通信
4, 16, 18 计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：串行通信模块

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	GOT 的端口编号			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	GOT 的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIONumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_GOTETHER_QNCPU (0x56)		

*1: 对于 I/O 地址, 应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

0: 使用模块的默认通道

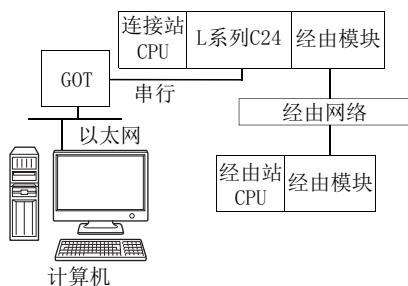
1: 通道 1

2: 通道 2

4.16 GOT 透明通信
4.16.18 计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU 侧端口：串行通信模块

(3) 连接站为 L 系列 C24 的情况下

(a) 构成



(b) 属性的模式

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1)	CC IE Field*2	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1)	CC IE Field*2	4)	4) ^{*3}	4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	2)	×	2)	×	×	×
	CC-Link	3)	3)	3)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

*3：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

(c) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式				
		1)	2)	3)	4)	
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型				
ActDestinationIO Number	0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	固定为 0 (0x00)
ActDestinationPort Number	0 (0x00)	GOT 的端口编号				
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)	
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 1 (0x01)	
ActHostAddress	1.1.1.1	GOT 的主机名或 IP 地址				
ActIntelligent PreferenceBit	0 (0x00)	固定为 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPCU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	固定为 0 (0x00)	
ActIONumber*1	1023 (0x3FF)	固定为 1023 (0x3FF)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	
ActMultiDrop ChannelNumber*2	0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)	固定为 0 (0x00)	
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)				
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	
ActThrough NetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)				
ActTimeOut	10000	以 ms 为单位由用户任意设置				

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号	固定为 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_LNCPU (0x57)			

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

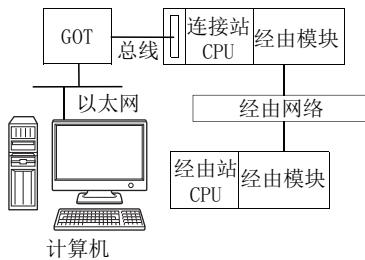
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.16.19 计算机侧端口：以太网卡，GOT2000/1000 侧端口：以太网端口，CPU 侧端口：总线

(1) 构成



4

(2) 属性的模式

连接站 CPU QCPU (Q 模式)	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
1) ^{*1}	CC IE Control	×	×
	CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU QCUP (Q 模式)	经由网络	经由站 CPU					
		QCUP (Q 模式) ^{*1}	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
1) ^{*1}	CC IE Control	2)	2) ^{*2}	2) ^{*3}	×	×	×
	CC IE Field	2)	2)	×	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	3)	×	3)	×	×	×
	CC-Link	4)	4)	4)	×	×	×

连接站 CPU	经由网络	经由站 CPU	
		RCPU	R 运动 CPU
C 语言控制器			
1)	CC IE Control CC IE Field	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

连接站 CPU	经由网络	经由站 CPU					
		QCPU (Q 模式) ^{*1}	C 语言控 制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
C 语言控制器							
1)	CC IE Control CC IE Field	2)	2) ^{*2}	2) ^{*3}	×	×	×
	MELSECNET/H	2)	2)	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	4)	4)	4)	×	×	×

带括号的数字：可以访问（数字为属性的模式），×：不能访问

*1：冗余 CPU 的情况下不能进行访问。

*2：由于 Q12DCCPU-V(基本功能模式) 不支持 CC-Link IE 现场网络，因此不能访问 CC-Link IE 现场网络。

*3：由于 LCPU 不支持 CC-Link IE 控制网络，因此不能访问 CC-Link IE 控制网络。

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的 CPU 类型			
ActDestinationIONumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	GOT 的端口编号			
ActDidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActDsidPropertyBit	1 (0x01)	固定为 1 (0x01)	固定为 1 (0x01)	固定为 0 (0x00)	固定为 0 (0x00)
ActHostAddress	1.1.1.1	GOT 的主机名或 IP 地址			
ActIntelligentPreferenceBit	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)	对象站 QCPU(Q 模式)、 C 语言控制器、 LCPU : 1 (0x01) 除上述以外 : 0 (0x00)
ActIOPortNumber*1	1023 (0x3FF)	单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	对象站侧 单 CPU 时 固定为 1023 (0x3FF) 多 CPU 时 连接 CPU 机号： 1023 (0x3FF) 1 号 CPU: 992 (0x3E0) 2 号 CPU: 993 (0x3E1) 3 号 CPU: 994 (0x3E2) 4 号 CPU: 995 (0x3E3)	连接站侧模块 I/O 地址	连接站侧模块 I/O 地址
ActMultiDropChannelNumber*2	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	多点通道 No.	固定为 0 (0x00)
ActNetworkNumber	0 (0x00)	固定为 0 (0x00)	对象站侧模块网络 No.	固定为 0 (0x00)	固定为 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	固定为 255 (0xFF)	对象站侧模块站号	固定为 255 (0xFF)	固定为 255 (0xFF)
ActThroughNetworkType	0 (0x00)	不包含 MELSECNET/10 : 0 (0x00) 包含 MELSECNET/10 : 1 (0x01)			
ActTimeOut	10000	以 ms 为单位由用户任意设置			

4.16 GOT 透明通信
4.16.19 计算机侧端口：以太网卡，GOT2000/1000 侧端口：以太网端口，CPU 侧端口：总线

属性	默认值	属性的模式			
		1)	2)	3)	4)
ActUnitNumber	0 (0x00)	固定为 0 (0x00)	固定为 0 (0x00)	对象站侧模块站号	对象站侧模块站号
ActUnitType	0x13 (UNIT_QNCPU)		UNIT_GOTETHER_QBUS (0x58)		

*1: 对于 I/O 地址，应指定将实际 I/O 起始 No. 用 16 相除后的值。

*2: 对多点链接的通道 No. 应进行下述指定。

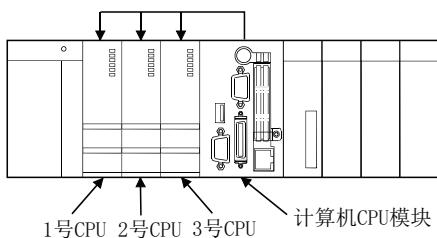
0: 使用模块的默认通道

1: 通道 1

2: 通道 2

4.17 Q系列总线通信

(1) 构成



(2) 属性的模式

本机基板	
RCPU	R运动CPU
×	×

本机基板		
QCPU(Q模式)	C语言控制器	Q运动CPU
1)	×	1)

带括号的数字：可以访问（数字为属性的模式），×：不能访问

(3) 属性一览

各通信路径中设置的属性如下所示。未记载的属性无需进行设置。

属性	默认值	属性的模式
		1)
ActCpuType	34 (CPU_Q02CPU)	对应于对象站的CPU类型
ActIOnumber*1	1023 (0x3FF)	1号CPU 992 (0x3E0) 2号CPU 993 (0x3E1) 3号CPU 994 (0x3E2)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_QBF (0x10)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QBF (0x1F)

4.17 Q系列总线通信
4.16, 19 计算机侧端口：以太网卡, GOT2000/1000 侧端口：以太网端口, CPU侧端口：总线

第 5 章 函数

本章介绍编程时的注意事项及函数的有关详细内容。

在开始创建程序之前必须阅读下节所示的编程时的注意事项。

(1) 函数的分类

函数中有 .NET 控件用及 ACT 控件用这 2 种类型。可使用的接口及编程语言有所不同。

对应控件	I/F 类型	编程语言	参照
ACT 控件	调度 I/F (推荐)	<ul style="list-style-type: none">• VBA• Visual C++® .NET (MFC)• VBScript• Visual Basic® .NET• Visual C++® .NET• Visual C#® .NET	314 页 5.2 节
	自定义 I/F	<ul style="list-style-type: none">• Visual C++® .NET (MFC)	
.NET 控件	调度 I/F	<ul style="list-style-type: none">• Visual Basic® .NET• Visual C++® .NET• Visual C#® .NET	400 页 5.3 节

(2) I/F 类型

在 ACT 控件用函数中，可以使用调度 I/F 及自定义 I/F 这 2 种类型。

I/F 类型	说明	特点
调度 I/F	是将控件粘贴到 Visual Basic® .NET、Visual C++® .NET、Visual C#® .NET、VBA(Excel, Access) 上的窗体中使用控件的方法。	无需进行用于 I/F 获取及生成对象的编程。(也可在不将控件粘贴到窗体的状况下，创建生成对象的程序) 与自定义 I/F 相比，可以轻松地创建程序。
自定义 I/F	在使用控件时使用 I/F 获取函数获取 I/F。 此外，生成对象时也使用对象生成函数。	与调度 I/F 相比，其程序较为复杂，但可对对象的生成 / 消失进行细致的管理。

5.1 编程时的注意事项

本节介绍创建程序时的注意事项有关内容。

(1) 多线程编程时的注意事项

进行多线程编程时，应遵守 COM 及 ActiveX 控件的规则。

有关详细内容请参阅 COM 及 ActiveX 控件的规则、参考书。

要 点

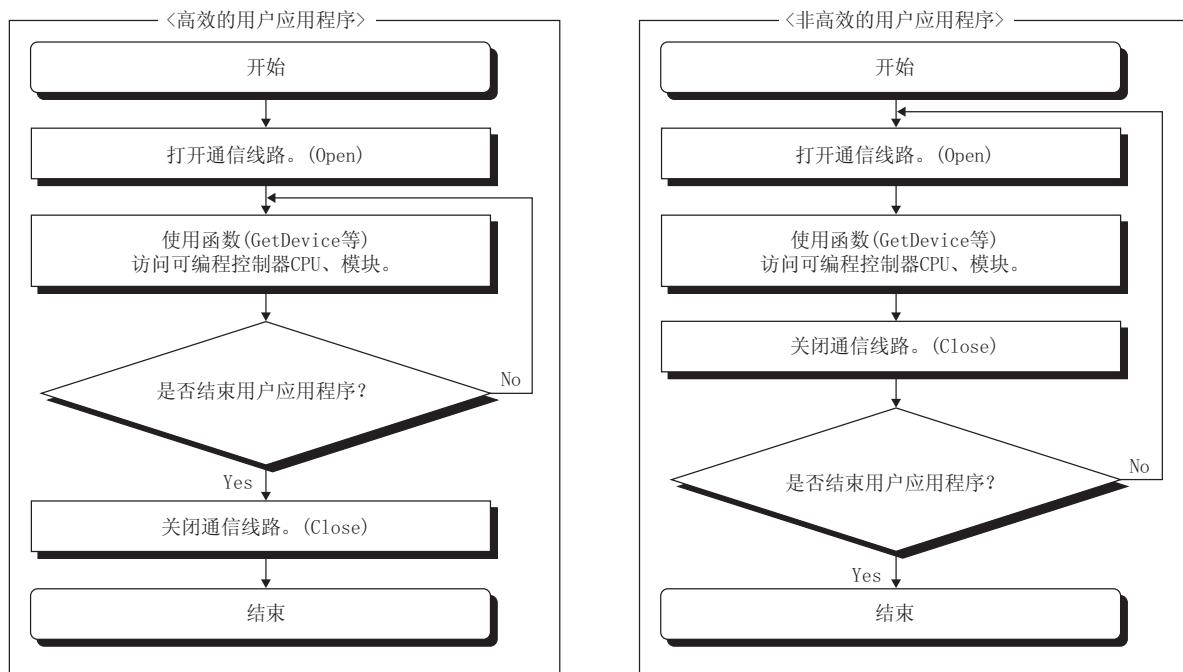
- 在 MX Component 中使用的 ActiveX 控件属于 STA 模式。
- 将接口指针传递到其它单元的情况下，需要对接口指针进行编组。
应使用 COM 函数的 CoMarshalInterThreadInterfaceInStream、CoGetInterfaceAndReleaseStream 进行同步。

(2) 关于 Open(通信线路的打开)

由于 Open 函数进行通信路径的建立、可编程控制器内部信息的获取等处理，因此处理时间有可能较长。

为了提高用户应用程序的速度性能，需要创建高效的程序。

高效的用户应用程序及非高效的用户应用程序的创建流程如下所示。



(3) 使用 QSCPU 时的注意事项

在 QSCPU 中不能使用下述函数。

函数名	功能
WriteDeviceBlock	软元件的批量写入
WriteDeviceRandom	软元件的随机写入
SetDevice	软元件数据的设置
WriteBuffer	缓冲存储器写入
SetClockData	时钟数据写入
SetCpuStatus	远程控制
WriteDeviceBlock2	软元件的批量写入
WriteDeviceRandom2	软元件的随机写入
SetDevice2	软元件数据的设置

要 点 

使用了不能使用的函数的情况下将返回出错代码“0x010A42A0”(CPU 访问口令不符)。

(4) 使用变频器 / 机器人控制器时的注意事项

.NET 控件的函数，不能在变频器通信 / 机器人控制器通信时使用。

要 点 

使用了不能使用的函数的情况下将返回出错代码“0xF0000007”(功能不支持出错)。

(5) LONG型的函数与SHORT型/INT型函数的不同点

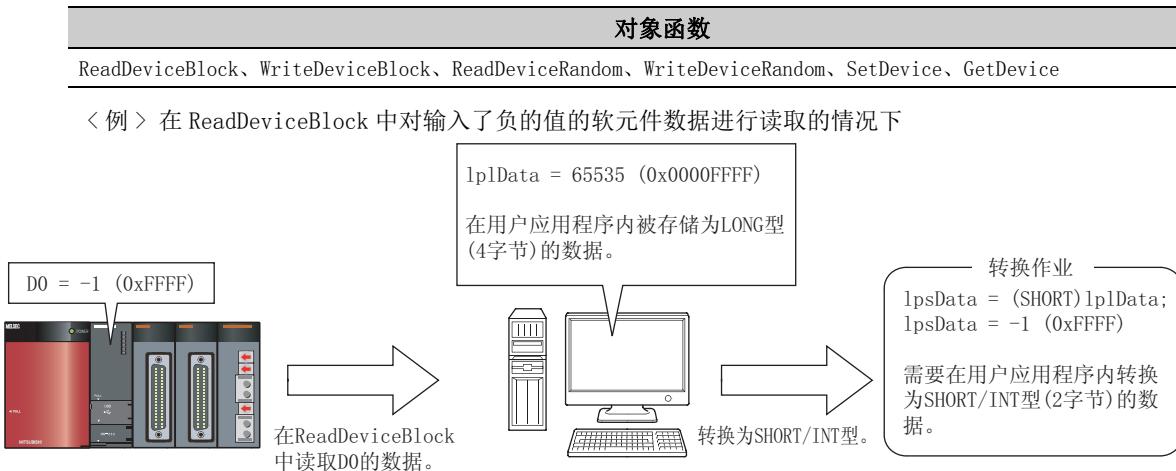
在MX Component中作为用于进行“软元件的批量读取”、“软元件的批量写入”、“软元件的随机读取”、“软元件的随机写入”、“软元件数据的设置”及“软元件数据的获取”的函数配备了LONG型函数及SHORT型/INT型函数这2种类型。

LONG型的函数与SHORT型/INT型的函数的不同点如下所示。

(a) LONG型函数 (317页5.2.3项～335页5.2.8项)

对LONG型函数以负的软元件值进行写入/读取的情况下，需要在用户应用程序内进行至SHORT型/INT型的转换作业。

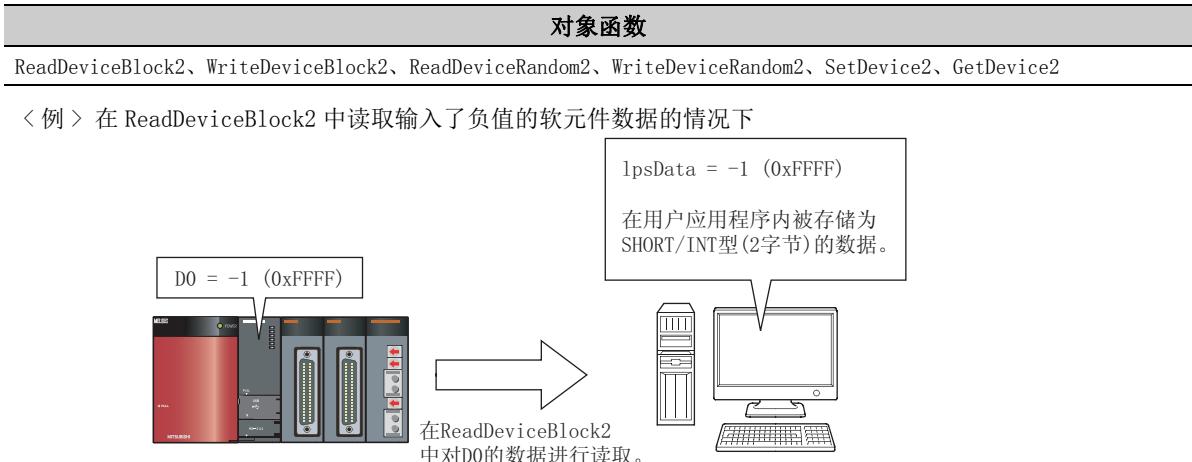
对负的软元件值进行写入/读取的情况下应使用(b)中所示的SHORT型/INT型函数。



(b) SHORT型/INT型函数 (374页5.2.18项～391页5.2.23项)

对SHORT型/INT型函数以负的软元件值进行写入/读取的情况下，可以原样不变地使用进行写入/读取的软元件值。(无需进行转换作业。)

处理双字软元件等的情况下，应使用(a)中所示的LONG型函数。



5.2 函数的详细内容 (ACT 控件用)

5.2.1 Open(通信线路的打开)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) Ut1Type 控件中可以使用。

(2) 功能

打开通信线路。

(3) 格式 (调度 I/F)

(a) Visual C++® .NET (MFC)、VBA

```
lRet = object.Open()  
Long           lRet          返回值          Output
```

(b) VBScript

```
varRet = object.Open()  
VARIANT       varRet        返回值 (内容为 LONG 型)    Output
```

(c) Visual Basic® .NET

```
iRet = object.Open()  
Integer        iRet          返回值          Output
```

(d) Visual C++® .NET

```
iRet = object.Open()  
int            iRet          返回值          Output
```

(e) Visual C#® .NET

```
iRet = object.Open()  
int            iRet          返回值          Output
```

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.Open( *lplRetCode )  
HRESULT      hResult        COM 的返回值      Output  
LONG         *lplRetCode   通信函数的返回值  Output
```

(5) 说明

基于 Open 函数用属性的设置值进行线路连接。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 使用调制解调器通信时在 Connect 执行完毕之前不能执行 Open。
- Open 完毕后即使更改 Open 函数用属性，通信对象也会被更改。进行通信设置的更改的情况下，应关闭 1 次通信线路后，进行通信对象的设置后再次打开通信线路。
- Open 时，即使在 ActCpuType 属性中输入了与进行通信的 CPU 不同 CPU 类型的情况下，Open 也可能会正常结束。
在这种情况下，有可能发生连接范围、可以使用的方法及软元件范围变窄等现象。
Open 时，应在 ActCpuType 属性中设置正确的 CPU 类型。

5.2.2 Close(通信线路的关闭)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。

(2) 功能

关闭通信线路。

(3) 格式 (调度 I/F)

(a) Visual C++® .NET (MFC)、VBA

lRet = object.Close()	Long	IRet	返回值	Output
-----------------------	------	------	-----	--------

(b) VBScript

varRet = object.Close()	VARIANT	varRet	返回值 (内容为 LONG 型)	Output
-------------------------	---------	--------	------------------	--------

(c) Visual Basic® .NET

IRet = object.Close()	Integer	IRet	返回值	Output
-----------------------	---------	------	-----	--------

(d) Visual C++® .NET

iRet = object.Close()	int	iRet	返回值	Output
-----------------------	-----	------	-----	--------

(e) Visual C#® .NET

iRet = object.Close()	int	iRet	返回值	Output
-----------------------	-----	------	-----	--------

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

hResult = object.Close(*lpIRetValCode)	HRESULT	hResult	COM 的返回值	Output
	LONG	*lpIRetValCode	通信函数的返回值	Output

(5) 说明

通过 Open 函数关闭连接的线路。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

5.2.3 ReadDeviceBlock(软元件的批量读取)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

进行软元件的批量读取。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.ReadDeviceBlock(szDevice, lSize, lData(0))
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Long	lSize	读取点数	Input
Long	lData(n)	读取的软元件值	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.ReadDeviceBlock(szDevice, lSize, *iplData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Long	lSize	读取点数	Input
Long	*iplData	读取的软元件值	Output

(c) VBScript

```
varRet = object.ReadDeviceBlock(varDevice, varSize, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	varSize	读取点数 (内容为 LONG 型)	Input
VARIANT	lpvarData	读取的软元件值 (内容为 LONG 数组型)	Output

(d) Visual Basic® .NET

```
IRet = object.ReadDeviceBlock(szDevice, iSize, iData(0))
```

Integer	IRet	返回值	Output
String	szDevice	软元件名	Input
Integer	iSize	读取点数	Input
Integer	iData(n)	读取的软元件值	Output

(e) Visual C++® .NET

```
iRet = object.ReadDeviceBlock(*szDevice, iSize, *iplData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
int	iSize	读取点数	Input
int	*iplData	读取的软元件值	Output

(f) Visual C#® .NET

```
iRet = object.ReadDeviceBlock(szDevice, iSize, out iData[0])
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
int	iSize	读取点数	Input
int[n]	iData	读取的软元件值	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.ReadDeviceBlock( szDevice, lSize, *lplData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名	Input
LONG	lSize	读取点数	Input
LONG	*lplData	读取的软元件值	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 从通过 szDevice(varDevice) 指定的软元件开始，对 lSize(varSize) 容量的软元件值进行批量读取。
- 读取的软元件值被存储到 lData(lplData 或 lpvarData) 中。
- lData(lplData 或 lpvarData) 应确保 lSize(varSize) 以上的数组。

(6) 软元件指定方法

软元件的指定方法如下所示。

〈指定位软元件时〉

(例) 从 M0 开始以 16 点单位读取 3 点(3字)。

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	M0 ~ M15*1
	M16 ~ M31*1
	M32 ~ M47*1

〈指定字软元件时〉

(例) 从 D0 开始读取 3 点。

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0
	D1
	D2

〈指定双字软元件时〉

(例) 从 LTN0 开始读取 3 点。

高位 2 字节	低位 2 字节
LTN0 的 H	LTN0 的 L
LTN1 的 H	LTN1 的 L
LTN2 的 H	LTN2 的 L

〈指定 FXCPU 的 CN200 以后时〉

(例) 从 CN200 开始读取 6 点。*2

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	CN200 的 L
	CN200 的 H
	CN201 的 L
	CN201 的 H
	CN202 的 L
	CN202 的 H

〈指定 FD 软元件时(4字软元件)〉

(例) 从 FD0 开始读取 6 点。

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	FD0 的 LL
	FD0 的 LH
	FD0 的 HL
	FD0 的 HH
	FD1 的 LL
	FD1 的 LH

〈网关软元件中分配的 8 位软元件〉

(例) 从 EG0 开始读取 4 点。

(EG0 ~ EG3 中分配了 8 位软元件(SHARP 生产的可编程控制器的 E0000 ~ E0007)的情况下)

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	EG0
	(E0001) (E0000)
	EG1
	(E0003) (E0002)
	EG2
	(E0005) (E0004)
	EG3
	(E0007) (E0006)

*1: 按照软元件的编号顺序从低位开始存储。

*2: 对于 FXCPU 的 CN200 以后, 以 2 点进行 2 字读取。如果以 1 点进行读取将发生出错。

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- lSize(varSize) 中可指定的最大读取点数为满足下述条件的范围。
 读取开始软元件编号 + 读取点数 ≤ 最终软元件编号
 - 指定位软元件时, 对软元件编号可以以 16 的倍数进行指定。
 - 对于 lData(lplData 或 lpvarData), 应预留出相当于 lSize(varSize) 中指定点数的存储器区域。
 没有存储器区域的情况下, 有可能会发生应用程序出错等严重现象。
-

5.2.4 WriteDeviceBlock(软元件的批量写入)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1: 变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

进行软元件的批量写入。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.WriteDeviceBlock(szDevice, lSize, lData(0))
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Long	lSize	写入点数	Input
Long	lData(n)	写入软元件值	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.WriteDeviceBlock(szDevice, lSize, *iplData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Long	lSize	写入点数	Input
Long	*iplData	写入软元件值	Input

(c) VBScript

```
varRet = object.WriteDeviceBlock(varDevice, varSize, varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	varSize	写入点数 (内容为 LONG 型)	Input
VARIANT	varData	写入的软元件值 (内容为 LONG 数组型)	Input

(d) Visual Basic® .NET

```
IRet = object.WriteDeviceBlock(szDevice, iSize, iData(0))
```

Integer	IRet	返回值	Output
String	szDevice	软元件名	Input
Integer	iSize	写入点数	Input
Integer	iData(n)	写入的软元件值	Input

(e) Visual C++® .NET

```
iRet = object.WriteDeviceBlock(*szDevice, iSize, *iplData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
int	iSize	写入点数	Input
int	*iplData	写入的软元件值	Input

(f) Visual C#® .NET

```
iRet = object.WriteDeviceBlock(szDevice, iSize, ref iData[0])
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
int	iSize	写入点数	Input
int[n]	iData	写入的软元件值	Input

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.WriteDeviceBlock( szDevice, lSize, *lplData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名	Input
LONG	lSize	写入点数	Input
LONG	*lplData	写入的软元件值	Input
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 从通过 szDevice(varDevice) 指定的软元件开始，批量写入 lSize(varSize) 容量的软元件值。
- 将写入的软元件值存储到 lData(lplData 或 varData) 中。
- 对于 lData(lplData 或 varData)，应预留出 lSize(varSize) 以上的数组。

(6) 软元件指定方法

软元件的指定方法如下所示。

〈指定位软元件时〉

(例) 从 M0 开始以 16 点单位写入 3 点(3字)。

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	M0 ~ M15*1
	M16 ~ M31*1
	M32 ~ M47*1

〈指定字软元件时〉

(例) 从 D0 开始写入 3 点。

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0
	D1
	D2

〈指定双字软元件时〉

(例) 从 LTN0 开始写入 3 点。

高位 2 字节	低位 2 字节
LTN0 的 H	LTN0 的 L
LTN1 的 H	LTN1 的 L
LTN2 的 H	LTN2 的 L

〈指定 FXCPU 的 CN200 以后时〉

(例) 从 CN200 开始写入 6 点。*2

高位 2 字节	低位 2 字节
	CN200 的 L
	CN200 的 H
不使用。 (存储 0。)	CN201 的 L
	CN201 的 H
	CN202 的 L
	CN202 的 H

〈指定 FD 软元件时(4字软元件)〉

(例) 从 FD0 开始写入 6 点。

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	FD0 的 LL
	FD0 的 LH
	FD0 的 HL
	FD0 的 HH
	FD1 的 LL
	FD1 的 LH

〈网关软元件中分配的 8 位软元件〉

(例) 从 EG0 开始写入 4 点。

(EG0 ~ EG3 中分配了 8 位软元件(SHARP 生产的可编程控制器的 E0000 ~ E0007)的情况下)

高位 2 字节	低位 2 字节
	EG0
	(E0001) (E0000)
	EG1
不使用。 (存储 0。)	(E0003) (E0002)
	EG2
	(E0005) (E0004)
	EG3
	(E0007) (E0006)

*1: 按照软元件的编号顺序从低位开始存储。

*2: 对于 FXCPU 的 CN200 以后, 以 2 点写入 2 字。如果以 1 点进行写入将发生出错。

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- lSize(varSize) 中可指定的最大读取点数为满足下述条件的范围。
写入开始软元件编号 + 写入点数 ≤ 最终软元件编号
 - 指定位软元件时，对软元件编号可以以 16 的倍数进行指定。
 - lData(lplData 或 varData)，应预留出相当于 lSize(varSize) 中指定点数的存储器区域。
没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
 - 向长定时器软元件 (LT) 及累积长定时器软元件 (LST) 写入时，不能使用 WriteDeviceBlock。
请使用 WriteDeviceRandom 或 SetDevice。
-

5.2.5 ReadDeviceRandom(软元件的随机读取)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。

(2) 功能

进行软元件的随机读取。

进行变频器 / 机器人控制器的监视类型的随机读取。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.ReadDeviceRandom(szDeviceList, lSize, lData(0))
```

Long	lRet	返回值	Output
String	szDeviceList	软元件名 / 监视类型	Input
Long	lSize	读取点数	Input
Long	lData(n)	读取的软元件值	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.ReadDeviceRandom(szDeviceList, lSize, *iplData)
```

Long	lRet	返回值	Output
CString	szDeviceList	软元件名 / 监视类型	Input
Long	lSize	读取点数	Input
Long	*iplData	读取的软元件值	Output

(c) VBScript

```
varRet = object.ReadDeviceRandom(varDeviceList, varSize, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDeviceList	软元件名 / 监视类型 (内容为字符串型)	Input
VARIANT	varSize	读取点数 (内容为 LONG 型)	Input
VARIANT	lpvarData	读取的软元件值 (内容为 LONG 数组型)	Output

(d) Visual Basic® .NET

```
IRet = object.ReadDeviceRandom(szDeviceList, iSize, iData(0))
```

Integer	IRet	返回值	Output
String	szDeviceList	软元件名 / 监视类型	Input
Integer	iSize	读取点数	Input
Integer	iData(n)	读取的软元件值	Output

(e) Visual C++® .NET

```
iRet = object.ReadDeviceRandom(*szDeviceList, iSize, *iplData)
```

int	iRet	返回值	Output
String	*szDeviceList	软元件名 / 监视类型	Input
int	iSize	读取点数	Input
int	*iplData	读取的软元件值	Output

(f) Visual C#® .NET

```
iRet = object.ReadDeviceRandom(szDevice, iSize, out iData[0])  
int iRet 返回值 Output  
String szDevice 软元件名 / 监视类型 Input  
int iSize 读取点数 Input  
int[n] iData 读取的软元件值 Output
```

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.ReadDeviceRandom( szDevice, lSize, *lplData, *lplRetCode )  
HRESULT hResult COM 的返回值 Output  
BSTR szDevice 软元件名 / 监视类型 Input  
LONG lSize 读取点数 Input  
LONG *lplData 读取的软元件值 Output  
LONG *lplRetCode 通信函数的返回值 Output
```

(5) 说明

- 对于通过 szDeviceList(varDeviceList) 指定的软元件群 / 监视类型群，仅按 lSize(varSize) 量的软元件值 / 监视类型量进行读取。
- 读取的软元件值被存储到 lData(lplData 或 lpvarData) 中。
- 对于软元件列表中指定的字符串应通过换行符分开。最终软元件的后面无需换行符。

(例)

使用 Visual Basic®.NET ,VBA, VBScript 时 : “D0” & vbLf & “D1” & vbLf & “D2”

使用 Visual C++®.NET 时 : D0\nD1\nD2

使用 Visual C#®.NET 时 : D0\nD1\nD2

- 对于 lData(lplData 或 lpvarData)，应预留出 lSize(varSize) 以上的数组。
- 关于可监视的项目（监视类型），请参阅以下手册。

 MX Component Version 4 操作手册

(6) 软元件指定方法

软元件的指定方法如下所示。

(例 1) 按以下方式进行指定的情况下(点数为3点)

使用 Visual Basic[®].NET, VBA, VBScript 时 : “M0” & vbLf & “D0” & vbLf & “K8M0”

使用 Visual C++[®].NET 时 : M0\nD0\nK8M0

使用 Visual C#[®].NET 时 : M0\nD0\nK8M0

高位2字节	低位2字节
不使用。 (存储0。)	M0 D0
M16 ~ M31*1	M0 ~ M15*1

(例 2) 指定了双字软元件时

使用 Visual Basic[®].NET, VBA, VBScript 时 : “LTN0” & vbLf & “LTN1” & vbLf & “LTN2”

使用 Visual C++[®].NET 时 : LTN0\nLTN1\nLTN2

使用 Visual C#[®].NET 时 : LTN0\nLTN1\nLTN2

高位2字节	低位2字节
LTN0 的 H	LTN0 的 L
LTN1 的 H	LTN1 的 L
LTN2 的 H	LTN2 的 L

(例 3) 指定包含 FXCPU 的 CN200 以后的软元件的情况下(合计点数为3点)*2

使用 Visual Basic[®].NET, VBA, VBScript 时 : “D0” & vbLf & “CN200” & vbLf & “D1”

When using Visual C++[®].NET 时 : D0\nCN200\nD1

使用 Visual C#[®].NET 时 : D0\nCN200\nD1

高位2字节	低位2字节
不使用。 (存储0。)	D0
CN200 的 H	CN200 的 L
不使用。 (存储0。)	D1

(例 4) 指定包含 FD 的软元件的情况下(合计点数为3点)

使用 Visual Basic[®].NET, VBA, VBScript 时 : “D0” & vbLf & “FD0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nFD0\nD1

使用 Visual C#[®].NET 时 : D0\nFD0\nD1

高位2字节	低位2字节
不使用。 (存储0。)	D0
	FD0 的 LL
	D1

*1：按照软元件的编号顺序从低位开始存储。

*2：对于 FXCPU 的 CN200 以后，随机读取以1点读取2字。

(例 5) 指定包含 EG 的软元件的情况下(合计点数为 3 点)

EG0 中分配了 8 位软元件(SHARP 生产的可编程控制器的 E0000、E0001) 的示例如下所示。

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “EG0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nEG0\nD1

使用 Visual C#[®].NET 时 : D0\nEG0\nD1

高位 2 字节	低位 2 字节
	D0
不使用。 (存储 0。)	EG0
	(E0001) (E0000)
	D1

(例 6) 指定变频器的监视类型(1、2、5)的情况下(合计点数为 3 点)

使用 Visual Basic[®].NET 、VBA、VBScript 时 : “1” & vbLf & “2” & vbLf & “5”

使用 Visual C++[®].NET 时 : 1\n2\n5

使用 Visual C#[®].NET 时 : 1\n2\n5

高位 2 字节	低位 2 字节
1 的 H	1 的 L
2 的 H	2 的 L
5 的 H	5 的 L

(例 7) 指定机器人控制器的监视类型(223.102.A、223.103.B)的情况下(合计点数为 2 点)

类型: (请求 ID). (数据类型). (参数)

使用 Visual Basic[®].NET 、VBA、VBScript 时 : “223.102.A” & vbLf & “223.103.B”

使用 Visual C++[®].NET 时 : 223.102.A\n223.103.B

使用 Visual C#[®].NET 时 : 223.102.A\n223.103.B

高位 2 字节	低位 2 字节
223.102.A 的 H	223.102.A 的 L
223.103.B 的 H	223.103.B 的 L

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。(☞ 496 页第 7 章 出错代码)

要 点

- lSize(varSize) 可指定的最大读取点数为 0x7FFFFFFF 点。
- 对于 lData(lpData 或 lpvarData), 应预留出相当于 lSize(varSize) 中指定点数的存储器区域。没有存储器区域的情况下, 有可能会发生应用程序出错等严重现象。

5.2.6 WriteDeviceRandom(软元件的随机写入)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1: 变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

进行软元件的随机写入。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.WriteDeviceRandom(szDeviceList, lSize, lData(0))
```

Long	lRet	返回值	Output
String	szDeviceList	软元件名	Input
Long	lSize	写入点数	Input
Long	lData(n)	写入的软元件值	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.WriteDeviceRandom(szDeviceList, lSize, *lplData)
```

Long	lRet	返回值	Output
CString	szDeviceList	软元件名	Input
Long	lSize	写入点数	Input
Long	*lplData	写入的软元件值	Input

(c) VBScript

```
varRet = object.WriteDeviceRandom(varDeviceList, varSize, varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDeviceList	软元件名 (内容为字符串型)	Input
VARIANT	varSize	写入点数 (内容为 LONG 型)	Input
VARIANT	varData	写入的软元件值 (内容为 LONG 数组型)	Input

(d) Visual Basic® .NET

```
iRet = object.WriteDeviceRandom(szDeviceList, iSize, iData(0))
```

Integer	iRet	返回值	Output
String	szDeviceList	软元件名	Input
Integer	iSize	写入点数	Input
Integer	iData(n)	写入的软元件值	Input

(e) Visual C++® .NET

```
iRet = object.WriteDeviceRandom(*szDeviceList, iSize, *iplData)
```

int	iRet	返回值	Output
String	*szDeviceList	软元件名	Input
int	iSize	写入点数	Input
int	*iplData	写入的软元件值	Input

(f) Visual C#® .NET

```
iRet = object.WriteDeviceRandom(szDevice、 iSize、 ref iData[0])  
int iRet 返回值 Output  
String szDevice 软元件名 Input  
int iSize 写入点数 Input  
int[n] iData 写入软元件 Input
```

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.WriteDeviceRandom( szDeviceList、 lSize、 *lplData、 *lplRetCode )  
HRESULT hResult COM 的返回值 Output  
BSTR szDeviceList 软元件名 Input  
LONG lSize 写入点数 Input  
LONG *lplData 写入的软元件值 Input  
LONG *lplRetCode 通信函数的返回值 Output
```

(5) 说明

- 将通过 szDeviceList(varDeviceList) 指定的软元件群，只写入相当于 lSize(varSize) 容量的软元件值。
- 将写入的软元件值存储到 lData(lpData 或 varData) 中。
- 对于软元件列表中指定的字符串应通过换行符分开。

最终软元件的后面无需换行符。

(例)

使用 Visual Basic®.NET ,VBA,VBScript 时 :	“D0” & vbLf & “D1” & vbLf & “D2”
使用 Visual C++®.NET 时 :	D0\nD1\nD2
使用 Visual C#®.NET 时 :	D0\nD1\nD2

- 对于 lData(lpData 或 varData)，应预留出 lSize(varSize) 以上的数组。

(6) 软元件指定方法

软元件的指定方法如下所示。

(例 1) 按下述方式进行指定的情况下 (点数为 3 点)

使用 Visual Basic®.NET ,VBA,VBScript 时 :	“M0” & vbLf & “D0” & vbLf & “K8M0”
使用 Visual C++®.NET 时 :	M0\nD0\nK8M0
使用 Visual C#®.NET 时 :	M0\nD0\nK8M0

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	M0
	D0
M16 ~ M31*1	M0 ~ M15*1

*1：按照软元件的编号顺序从低位开始存储。

(例2) 指定了双字软元件时

使用 Visual Basic[®].NET , VBA, VBScript 时 :

“LTN0” & vbLf & “LTN1” & vbLf & “LTN2”

使用 Visual C++[®].NET 时 :

LTN0\nLTN1\nLTN2

使用 Visual C#[®].NET 时 :

LTN0\nLTN1\nLTN2

高位 2 字节	低位 2 字节
LTN0 的 H	LTN0 的 L
LTN1 的 H	LTN1 的 L
LTN2 的 H	LTN2 的 L

(例3) 指定包含 FXCPU 的 CN200 以后的软元件的情况下(合计点数为3点)*2

使用 Visual Basic[®].NET , VBA, VBScript 时 :

“D0” & vbLf & “CN200” & vbLf & “D1”

使用 Visual C++[®].NET 时 :

D0\nCN200\nD1

使用 Visual C#[®].NET 时 :

D0\nCN200\nD1

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0
CN200 的 H	CN200 的 L
不使用。 (存储 0。)	D1

(例4) 指定包含 FD 的软元件的情况下(合计点数为3点)

使用 Visual Basic[®].NET , VBA, VBScript 时 :

“D0” & vbLf & “FD0” & vbLf & “D1”

使用 Visual C++[®].NET 时 :

D0\nFD0\nD1

使用 Visual C#[®].NET 时 :

D0\nFD0\nD1

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0
	FD0 的 LL
	D1

(例5) 指定包含 EG 的软元件的情况下(合计点数为3点)

在 EGO 中分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000、E0001) 的示例如下所示。

使用 Visual Basic[®].NET , VBA, VBScript 时 :

“D0” & vbLf & “EG0” & vbLf & “D1”

使用 Visual C++[®].NET 时 :

D0\nEG0\nD1

使用 Visual C#[®].NET 时 :

D0\nEG0\nD1

高位 2 字节	低位 2 字节
	D0
不使用。 (存储 0。)	EG0
	(E0001) (E0000)
	D1

*2: 对于 FXCPU 的 CN200 以后，随机读取以 1 点读取 2 字。

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 在 lSize(varSize) 中可指定的最大写入点数为 0x7FFFFFFF 点。
 - 对于 lData(lplData 或 varData)，应预留出相当于 lSize(varSize) 中指定点数的存储器区域。
没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
 - 如果对 Q 运动 CPU 进行访问，将返回出错信息。
-

5.2.7 SetDevice(软元件数据的设置)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1: 变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

进行 1 点软元件的设置。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.SetDevice(szDevice, lData)
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Long	lData	设置数据	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.SetDevice(szDevice, lData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Long	lData	设置数据	Input

(c) VBScript

```
varRet = object.SetDevice(varDevice, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	varData	设置数据 (内容为 LONG 型)	Input

(d) Visual Basic® .NET

```
IRet = object.SetDevice(szDevice, iData)
```

Integer	IRet	返回值	Output
String	szDevice	软元件名	Input
Integer	iData	设置数据	Input

(e) Visual C++® .NET

```
iRet = object.SetDevice(*szDevice, iData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
int	iData	设置数据	Input

(f) Visual C#® .NET

```
iRet = object.SetDevice(szDevice, iData)
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
int	iData	设置数据	Input

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.SetDevice( szDevice, lData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名	Input
LONG	lData	设置数据	Input
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 对通过 szDevice (varDevice) 指定的 1 点软元件使用 lData (varData) 进行指定的操作。
- 位软元件设置时，lData 值 (varData 值) 的最低位的位将有效。

(6) 软元件指定方法

软元件的指定方法如下所示。

〈指定位软元件时〉

(例) M0

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	M0

〈指定字软元件时〉

(例) D0

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0

〈指定双字软元件时〉

(例) LTN0

高位 2 字节	低位 2 字节
LTN0 的 H	LTN0 的 L

〈指定双字软元件时 (位数指定)〉

(例) K8M0

高位 2 字节	低位 2 字节
M16 ~ M31 ^{*1}	M0 ~ M15 ^{*1}

〈指定 FXCPU 的 CN200 以后时〉

(例) CN200

高位 2 字节	低位 2 字节
CN200 的 H	CN200 的 L

〈指定网关软元件时〉

(例) 在 EGO 分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000、E0001) 的情况下

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	EG0 (E0001) (E0000)

*1：按照软元件的编号顺序从低位开始存储。

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。([ 496 页第 7 章 出错代码])

5.2.8 GetDevice(软元件数据的获取)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。

(2) 功能

获取 1 点软元件的数据。

获取变频器 / 机器人控制器的监视类型 1 点的数据。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.GetDevice(szDevice, lData)
```

Long	lRet	返回值	Output
String	szDevice	软元件名 / 监视类型	Input
Long	lData	获取数据	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.GetDevice(szDevice, *lpIData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名 / 监视类型	Input
Long	*lpIData	获取数据	Output

(c) VBScript

```
varRet = object.GetDevice(varDevice, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 / 监视类型 (内容为字符串型)	Input
VARIANT	lpvarData	获取数据 (内容为 LONG 型)	Output

(d) Visual Basic® .NET

```
iRet = object.GetDevice(szDevice, iData)
```

Integer	iRet	返回值	Output
String	szDevice	软元件名 / 监视类型	Input
Integer	iData	获取数据	Output

(e) Visual C++® .NET

```
iRet = object.GetDevice(*szDevice, *ipIData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名 / 监视类型	Input
int	*ipIData	获取数据	Output

(f) Visual C#® .NET

```
iRet = object.GetDevice(*szDevice, *ipIData)
```

int	iRet	返回值	Output
String	szDevice	软元件名 / 监视类型	Input
int	iData	获取数据	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.GetDevice( szDevice, *lplData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名 / 监视类型	Input
LONG	*lplData	获取数据	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

将通过 szDevice(varDevice) 指定的 1 点软元件的数据存储到 lData(lpData 或 lpvarData) 中。
关于可监视的项目（监视类型），请参阅以下手册。

 MX Component Version 4 操作手册

(6) 软元件指定方法

软元件 / 监视类型的指定方法如下所示。

＜指定位软元件时＞

(例) M0

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	M0

＜指定字软元件时＞

(例) D0

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0

＜指定双字软元件时＞

(例) LTN0

高位 2 字节	低位 2 字节
LTNO 的 H	LTNO 的 L

＜指定双字软元件时（位数指定）＞

(例) K8M0

高位 2 字节	低位 2 字节
M16 ~ M31*1	M0 ~ M15*1

*1：按照软元件的编号顺序从低位开始存储。

＜指定 FXCPU 的 CN200 以后时＞

(例) CN200

高位 2 字节	低位 2 字节
CN200 的 H	CN200 的 L

＜指定网关软元件时＞

(例) 在 E00 中分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000、E0001) 的情况下

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	E00
	(E0001) (E0000)

<指定变频器的监视类型时>

(例) 2

高位 2 字节	低位 2 字节
1 的 H	1 的 L

<指定机器人控制器的监视类型时>

类型：(请求 ID). (数据类型). (参数)

(例) 223. 102. A

高位 2 字节	低位 2 字节
223. 102. A 的 H	223. 102. A 的 L

(7) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。(☞ 496 页第 7 章 出错代码)

5.2.9 ReadBuffer(缓冲存储器读取)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 通信路径的使用可否

通信路径的使用可否如下表所示。

通信路径		属性的设置		使用可否
		ActUnitType	ActProtocolType	
串行通信	连接模块为 R 系列 C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○
	连接模块为 Q 系列 C24	UNIT_QJ71C24		○
	连接模块为 L 系列 C24	UNIT_LJ71C24		○
	连接模块为 FX 扩展端口	UNIT_FX485BD		×
以太网通信	UNIT_RJ71EN71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	○	○
	UNIT_RJ71EN71_DIRECT	ROTOCOL_UDPIP		
	UNIT_QJ71E71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP		
	UNIT_RETHER	PROTOCOL_TCPIP、 PROTOCOL_UDPIP		
	UNIT_RETHER_DIRECT	ROTOCOL_UDPIP		
	UNIT_LNETHETHER	PROTOCOL_TCPIP		
	UNIT_LNETHETHER_DIRECT	PROTOCOL_UDPIP		
	UNIT_QNETHETHER	PROTOCOL_TCPIP		
	UNIT_QNETHETHER_DIRECT	PROTOCOL_UDPIP		
	UNIT_NZ2GF_ETB	PROTOCOL_TCPIP		
	UNIT_NZ2GF_ETB、 UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP		
	UNIT_FXETHER	PROTOCOL_TCPIP		
	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP		
CPU COM 通信	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	连接目标 CPU 为 LCPU	UNIT_LNCPU		○
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTION		×
	连接目标 CPU 为 FXCPU	UNIT_FXCPU		○ *1
USB 通信	连接目标 CPU 为 RCPU	UNIT_RUSB	PROTOCOL_USB	○
	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNUSB		○
	连接目标 CPU 为 LCPU	UNIT_LNUUSB		○
	连接目标 CPU 为 QSCPU	UNIT_QSUSB		○
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTIONUSB		×
	连接目标 CPU 为 FXCPU	UNIT_FXCPU		○

○：可以使用，×：不能使用

*1：FXIN、FXINC、FX2N、FX2NC、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

通信路径	属性的设置		使用可否
	ActUnitType	ActProtocolType	
MELSECNET/H 通信	UNIT_MNETHBOARD	PROTOCOL_MNETH	<input type="radio"/> *2
CC-Link IE 控制网络通信	UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE 现场网络通信	UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link 通信	UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 通信	UNIT_G4QCPU	PROTOCOL_SERIAL	<input type="radio"/>
GX Simulator 通信	UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	<input type="radio"/> *3
GX Simulator2 通信	UNIT_SIMULATOR2	-	<input type="radio"/>
MT Simulator2 通信	UNIT_SIMULATOR2	-	<input checked="" type="checkbox"/>
调制解调器 通信	连接模块为 Q 系列 C24	UNIT_QJ71C24	<input type="radio"/>
	连接模块为 L 系列 C24	UNIT_LJ71C24	
	连接目标 CPU 为 FXCPU	UNIT_FXCPU	<input type="radio"/> *1
网关功能通信	UNIT_A900GOT	PROTOCOL_UDPPIP	<input checked="" type="checkbox"/>
GOT 透明功能通信	UNIT_RJ71C24、 UNIT_QNCPU、 UNIT_LNCPU	UNIT_FXCPU	<input type="radio"/>
	UNIT_FXCPU		<input type="radio"/> *1
	UNIT_QJ71C24、 UNIT_LJ71C24、 UNIT_GOT_RJ71EN71、 UNIT_GOT_QJ71E71、 UNIT_GOT_RETHER、 UNIT_GOT_QNETHER、 UNIT_GOT_LNETHER、 UNIT_A900GOT、 UNIT_GOT_NZ2GF_ETB、 UNIT_GOT_FXETHER、 UNIT_GOT_FXENET	PROTOCOL_SERIAL, PROTOCOL_USBGOT	<input type="radio"/>
	UNIT_GOTETHER_RJ71C24、 UNIT_GOTETHER_QNCPU、 UNIT_GOTETHER_LNCPU		<input type="radio"/>
	UNIT_GOTETHER_FXCPU		<input type="radio"/> *1
	UNIT_GOTETHER_QBUS		<input type="radio"/>
Q 系列总线通信	UNIT_QBF	PROTOCOL_QBF	<input type="radio"/>
变频器通信	-	COMM_RS232C、 COMM_USB	<input checked="" type="checkbox"/>

：可以使用，：不能使用

*1：FX1N、FX1NC、FX2N、FX2NC、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

*2：自板访问时将返回出错信息。

*3：FX0N、FX1N、FX1NC、FXU、FX2C、FX2N、FX2NC、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

5.2 函数的详细内容 (ACT 挂件用)
5.2.9 ReadBuffer(缓冲存储器读取)

(3) 功能

读取特殊功能模块的缓冲存储器的值。

(4) 格式 (调度 I/F)

(a) VBA

```
iRet = object.ReadBuffer(lStartIO, lAddress, lReadSize, iData(0))
```

Long	iRet	返回值	Output
Long	lStartIO	读取值的模块的起始 I/O 编号	Input
Long	lAddress	缓冲存储器的地址	Input
Long	lReadSize	读取容量	Input
Integer	iData(n)	从缓冲存储器中读取的值	Output

(b) Visual C++® .NET (MFC)

```
iRet = object.ReadBuffer(lStartIO, lAddress, lReadSize, *lpsData)
```

Long	iRet	返回值	Output
Long	lStartIO	读取值的模块的起始 I/O 编号	Input
Long	lAddress	缓冲存储器的地址	Input
Long	lReadSize	读取容量	Input
Short	*lpsData	从缓冲存储器中读取的值	Output

(c)

```
varRet = object.ReadBuffer(varStartIO, varAddress, varReadSize, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varStartIO	读取值的模块的起始 I/O 编号 (内容为 LONG 型)	Input
VARIANT	varAddress	缓冲存储器的地址 (内容为 LONG 型)	Input
VARIANT	varReadSize	读取容量 (内容为 LONG 型)	Input
VARIANT	lpvarData	从缓冲存储器中读取的值 (内容为 SHORT 数组型)	Output

(d) Visual Basic® .NET

```
iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, iData(0))
```

Integer	iRet	返回值	Output
Integer	iStartIO	读取值的模块的 I/O 编号	Input
Integer	iAddress	缓冲存储器的地址	Input
Integer	iReadSize	读取容量	Input
short	iData(n)	从缓冲存储器中读取的值	Output

(e) Visual C++® .NET

```
iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, *ipsData)
```

int	iRet	返回值	Output
int	iStartIO	读取值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iReadSize	读取容量	Input
short	*ipsData	从缓冲存储器中读取的值	Output

(f) Visual C#® .NET

```
iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, out sData[0])
```

int	iRet	返回值	Output
int	iStartIO	读取值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iReadSize	读取容量	Input
short[n]	sData	从缓冲存储器中读取的值	Output

(5) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.ReadBuffer( lStartIO, lAddress, lReadSize, *lpsData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
LONG	lStartIO	读取值的模块的起始 I/O 编号	Input
LONG	lAddress	缓冲存储器的地址	Input
LONG	lReadSize	读取容量	Input
SHORT	*lpsData	从缓冲存储器中读取的值	Output
LONG	*lplRetCode	通信函数的返回值	Output

(6) 说明

- 对于 lStartIO(varStartIO) 中指定的模块的 I/O 编号，应指定将实际的 I/O 编号用 16 相除后的值。
- 将位于 lStartIO(varStartIO) 中指定的起始 I/O 编号处的特殊功能模块的 lAddress(varAddress) 中指定的缓冲存储器地址的缓冲值进行相当于 lReadSize(varReadSize) 容量的读取。
- 进行连接站为 FXCPU 的 CPU COM 通信及 GX Simulator 通信的情况下，在模块起始 I/O 编号中，指定特殊扩展设备的块 No. (0 ~ 7)，在缓冲存储器地址中指定 0 ~ 32767。
- 对于 iData(lpsData 或 lpvarData)，应预留出 lReadSize(varReadSize) 以上的数组。

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

(8) 多 CPU 系统配置时的缓冲存储器读取 / 写入时的注意事项

如果在实际配置的多 CPU 系统与 GX Developer 中设置的 I/O 分配不相同的情况下执行函数，将发生下述现象。

应在对 GX Developer 中的 I/O 分配进行确认，对进行值的读取 / 写入的模块的 I/O 编号进行确认之后，再执行函数。

- 即使指定正确的 I/O 编号执行函数也发生出错。
- 如果指定了特定的 I/O 编号（实际 I/O 分配错误的 I/O 编号），虽然可以正常执行缓冲存储器读取，但如果进行缓冲存储器写入则发生出错（出错代码：0x010A4030、0x010A4042 等）。
- 虽然在多 CPU 系统中未发生可编程控制器 CPU 的出错（参数出错、SP.UNIT LAY ERR 等），但如果执行函数则用户应用程序中将发生出错。

要 点

- 如果对 RCPU、R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
- 对于 iData(lpsData 或 lpvarData)，应预留出相当于 lReadSize(varReadSize) 中指定点数的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 对 QCPU(Q 模式) 进行缓冲存储器读取 (ReadBuffer) 的情况下，只能对 Q 系列专用模块进行读取操作。此外，也不能从 QCPU(Q 模式) 的共享存储器中进行读取操作。

5.2.10 WriteBuffer(缓冲存储器写入)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) Ut1Type 控件中可以使用。^{*1}

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 通信路径的使用可否

通信路径的使用可否如下表所示。

通信路径		属性的设置		使用可否
		ActUnitType	ActProtocolType	
串行通信	连接模块为 R 系列 C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○
	连接模块为 Q 系列 C24	UNIT_QJ71C24		○
	连接模块为 L 系列 C24	UNIT_LJ71C24		○
	连接模块为 FX 扩展端口	UNIT_FX485BD		×
以太网通信	UNIT_RJ71EN71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	○	
	UNIT_RJ71EN71_DIRECT	ROTTOCOL_UDPIP		
	UNIT_QJ71E71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP		
	UNIT_RETHER	PROTOCOL_TCPIP、 PROTOCOL_UDPIP		
	UNIT_RETHER_DIRECT	ROTTOCOL_UDPIP		
	UNIT_LNETHER	PROTOCOL_TCPIP		
	UNIT_LNETHER、 UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP		
	UNIT_QNETHER	PROTOCOL_TCPIP		
	UNIT_QNETHER、 UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP		
	UNIT_NZ2GF_ETB	PROTOCOL_TCPIP		
	UNIT_NZ2GF_ETB、 UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP		
	UNIT_FXETHER	PROTOCOL_TCPIP		
	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP		
CPU COM 通信	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	连接目标 CPU 为 LCPU	UNIT_LNCPU		○
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTION		×
	连接目标 CPU 为 FXCPU	UNIT_FXCPU		○ *1
USB 通信	连接目标 CPU 为 RCPU	UNIT_RUSB	PROTOCOL_USB	○
	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNUSB		○
	连接目标 CPU 为 LCPU	UNIT_LNUUSB		○
	连接目标 CPU 为 QSCPU	UNIT_QSUSB		× *2
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTIONUSB		×
	连接目标 CPU 为 FXCPU	UNIT_FXCPU		○

○：可以使用，×：不能使用

*1：FX1N、FX1NC、FX2N、FX2NC、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

*2：访问 QSCPU 时将发生出错 0x010A42A0(访问口令不符)。

通信路径	属性的设置		使用可否
	ActUnitType	ActProtocolType	
MELSECNET/H 通信	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ *3
CC-Link IE 控制网络通信	UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE 现场网络通信	UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link 通信	UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 通信	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator 通信	UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○ *4
GX Simulator2 通信	UNIT_SIMULATOR2	-	○
MT Simulator2 通信	UNIT_SIMULATOR2	-	×
调制解调器通信	连接模块为 Q 系列 C24	UNIT_QJ71C24	○
	连接模块为 L 系列 C24	UNIT_LJ71C24	○
	连接目标 CPU 为 FXCPU	UNIT_FXCPU	○ *1
网关功能通信	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT 透明功能通信	UNIT_RJ71C24、 UNIT_QNCPU、 UNIT_LNCPU	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
	UNIT_FXCPU		○ *1
	UNIT_QJ71C24、 UNIT_LJ71C24、 UNIT_GOT_RJ71EN71、 UNIT_GOT_QJ71E71、 UNIT_GOT_RETHER、 UNIT_GOT_QNETHER、 UNIT_GOT_LNETHER、 UNIT_A900GOT、 UNIT_GOT_NZ2GF_ETB、 UNIT_GOT_FXETHER、 UNIT_GOT_FXENET		○
	UNIT_GOTETHER_RJ71C24、 UNIT_GOTETHER_QNCPU、 UNIT_GOTETHER_LNCPU	PROTOCOL_TCPIP	○
	UNIT_GOTETHER_FXCPU		○ *1
	UNIT_GOTETHER_QBUS		○
Q 系列总线通信	UNIT_QBF	PROTOCOL_QBF	○
变频器通信	-	COMM_RS232C、 COMM_USB	×

○：可以使用，×：不能使用

*1: FXIN、FXINC、FX2N、FX2NC、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

*2: 访问 QSCPU 时将发生出错 0x010A42A0(访问口令不符)。

*3: 自板访问时将返回出错信息。

*4: FXON、FXIN、FXINC、FX2N、FX2NC、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

(3) 功能

写入特殊功能模块的缓冲存储器的值。

(4) 格式 (调度 I/F)

(a) VBA

```
lRet = object.WriteBuffer(lStartIO, lAddress, lWriteSize, iData(0))
```

Long	lRet	返回值	Output
Long	lStartIO	写入值的模块的起始 I/O 编号	Input
Long	lAddress	缓冲存储器的地址	Input
Long	lWriteSize	写入的容量	Input
Integer	iData(n)	写入到缓冲存储器中的值	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.WriteBuffer(lStartIO, lAddress, lWriteSize, *lpsData)
```

Long	lRet	返回值	Output
Long	lStartIO	写入值的模块的起始 I/O 编号	Input
Long	lAddress	缓冲存储器的地址	Input
Long	lWriteSize	写入的容量	Input
Short	*lpsData	写入到缓冲存储器中的值	Input

(c) VBScript

```
varRet = object.WriteBuffer(varStartIO, varAddress, varWriteSize, varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varStartIO	写入值的模块的起始 I/O 编号 (内容为 LONG 型)	Input
VARIANT	varAddress	缓冲存储器的地址 (内容为 LONG 型)	Input
VARIANT	varWriteSize	写入的容量 (内容为 LONG 型)	Input
VARIANT	varData	写入到缓冲存储器中的值 (内容为 SHORT 数组型)	Input

(d) Visual Basic® .NET

```
lRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, iData(0))
```

Integer	lRet	返回值	Output
Integer	iStartIO	写入值的模块的 I/O 编号	Input
Integer	iAddress	缓冲存储器的地址	Input
Integer	iWriteSize	写入的容量	Input
Short	iData(n)	写入到缓冲存储器中的值	Output

(e) Visual C++® .NET

```
iRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, *ipsData)
```

int	iRet	返回值	Output
int	iStartIO	写入值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iWriteSize	写入的容量	Input
Short	*ipsData	写入到缓冲存储器中的值	Output

(f) Visual C#® .NET

```
iRet = object. WriteBuffer ( iStartIO、 iAddress、 iWriteSize、 ref sData[0])  
int           iRet          返回值          Output  
int           iStartIO      写入值的模块的 I/O 编号    Input  
int           iAddress      缓冲存储器的地址        Input  
int           iWriteSize    写入的容量        Input  
short[n]     sData         写入到缓冲存储器中的值    Input
```

(5) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.WriteBuffer( lStartIO、 lAddress、 lWriteSize、 *lpsData、 *lplRetCode )  
HRESULT      hResult       COM 的返回值      Output  
LONG         lStartIO      写入值的模块的起始 I/O 编号    Input  
LONG         lAddress      缓冲存储器的地址        Input  
LONG         lWriteSize    写入的容量        Input  
SHORT        *lpsData     写入到缓冲存储器中的值    Input  
LONG         *lplRetCode   通信函数的返回值      Output
```

(6) 说明

- 对于 lStartIO(varStartIO) 中指定的模块的 I/O 编号，应指定将实际的 I/O 编号用 16 相除后的值。
- 将位于 lStartIO(varStartIO) 中指定的起始 I/O 编号处的特殊功能模块的 lAddress(varAddress) 中指定的缓冲存储器地址的缓冲值进行相当于 lWriteSize(varWriteSize) 容量的写入。
- 进行连接站为 FXCPU 的 CPU COM 通信及 GX Simulator 通信的情况下，在模块起始 I/O 编号中指定特殊扩展设备的块 No. (0 ~ 7)，在缓冲存储器地址中指定 0 ~ 32767。
- 对于 iData(lpsData 或 varData)，应预留出 lWriteSize(varWriteSize) 以上的数组。

(7) 返回值

正常结束 : 返回 0。
非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

(8) 多 CPU 系统配置时的缓冲存储器读取 / 写入时的注意事项

如果在实际配置的多 CPU 系统与 GX Developer 中设置的 I/O 分配不相同的情况下执行函数，将发生下述现象。

应在对 GX Developer 中的 I/O 分配进行确认，对进行值的读取 / 写入的模块的 I/O 编号进行确认之后，再执行函数。

- 即使指定正确的 I/O 编号执行函数也发生出错。
- 如果指定了特定的 I/O 编号（实际 I/O 分配错误的 I/O 编号），虽然可以正常执行缓冲存储器读取，但如果进行缓冲存储器写入则发生出错（出错代码：0x010A4030、0x010A4042 等）。
- 虽然在多 CPU 系统中未发生可编程控制器 CPU 的出错（参数出错、SP.UNIT LAY ERR 等），但如果执行函数则用户应用程序中将发生出错。

要 点

- 如果对 RCPU、R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
- 对于 iData(lpsData 或 varData)，应预留出相当于 lWriteSize(varWriteSize) 中指定点数的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 对 QCPU(Q 模式) 进行缓冲存储器写入 (WriteBuffer) 的情况下，只能对 Q 系列专用模块进行写入操作。此外，也不能对 QCPU(Q 模式) 的共享存储器进行写入操作。

5.2.11 GetClockData(时钟数据读取)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 通信路径的使用可否

通信路径的使用可否如下表所示。

通信路径		属性的设置		使用可否
		ActUnitType	ActProtocolType	
串行通信	连接模块为 R 系列 C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○
	连接模块为 Q 系列 C24	UNIT_QJ71C24		○
	连接模块为 L 系列 C24	UNIT_LJ71C24		○
	连接模块为 FX 扩展端口	UNIT_FX485BD		○
以太网通信	UNIT_RJ71EN71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	○	○
	UNIT_RJ71EN71_DIRECT	ROTOCOL_UDPIP		
	UNIT_QJ71E71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP		
	UNIT_RETHER	PROTOCOL_TCPIP、 PROTOCOL_UDPIP		
	UNIT_RETHER_DIRECT	ROTOCOL_UDPIP		
	UNIT_LNETHETHER	PROTOCOL_TCPIP		
	UNIT_LNETHETHER、 UNIT_LNETHETHER_DIRECT	PROTOCOL_UDPIP		
	UNIT_QNETHETHER	PROTOCOL_TCPIP		
	UNIT_QNETHETHER、 UNIT_QNETHETHER_DIRECT	PROTOCOL_UDPIP		
	UNIT_NZ2GF_ETB	PROTOCOL_TCPIP		
	UNIT_NZ2GF_ETB、 UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP		
	UNIT_FXETHER	PROTOCOL_TCPIP		
CPU COM 通信	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP	○	○
	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNCPU		
	连接目标 CPU 为 LCPU	UNIT_LNCPU		
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTION		
USB 通信	连接目标 CPU 为 FXCPU	UNIT_FXCPU	PROTOCOL_USB	○
	连接目标 CPU 为 RCPU	UNIT_RUSB		○
	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNUSB		○
	连接目标 CPU 为 LCPU	UNIT_LNUUSB		○
	连接目标 CPU 为 QSCPU	UNIT_QSUSB		○
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTIONUSB		×

○：可以使用，×：不能使用

*1：自板访问时将返回出错信息。

通信路径	属性的设置		使用可否
	ActUnitType	ActProtocolType	
MELSECNET/H 通信	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ *1
CC-Link IE 控制网络通信	UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE 现场网络通信	UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link 通信	UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 通信	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator 通信	UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○
GX Simulator2 通信	UNIT_SIMULATOR2	-	○
MT Simulator2 通信	UNIT_SIMULATOR2	-	×
调制解调器通信	连接模块为 Q 系列 C24	UNIT_QJ71C24	○
	连接模块为 L 系列 C24	UNIT_LJ71C24	
	连接目标 CPU 为 FXCPU	UNIT_FXCPU	
网关功能通信	UNIT_A900GOT	PROTOCOL_UDPPIP	×
GOT 透明功能通信	UNIT_RJ71C24、 UNIT_QNCPU、 UNIT_LNCPU、 UNIT_FXCPU、 UNIT_QJ71C24、 UNIT_LJ71C24、 UNIT_GOT_RJ71EN71、 UNIT_GOT_QJ71E71、 UNIT_GOT_RETHER、 UNIT_GOT_QNETHER、 UNIT_GOT_LNETHER、 UNIT_A900GOT、 UNIT_GOT_NZ2GF_ETB、 UNIT_GOT_FXETHER、 UNIT_GOT_FXENET	PROTOCOL_SERIAL、 PROTOCOL_USBGOT	○
	UNIT_GOTETHER_RJ71C24、 UNIT_GOTETHER_QNCPU、 UNIT_GOTETHER_LNCPU、 UNIT_GOTETHER_FXCPU、 UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q 系列总线通信	UNIT_QBF	PROTOCOL_QBF	○
变频器通信	-	COMM_RS232C、 COMM_USB	×

○：可以使用，×：不能使用

*1：自板访问时将返回出错信息。

5.2 函数的详细内容 (ACT 控件用)
 5.2.11 GetClockData(时钟数据读取)

(3) 功能

从可编程控制器 CPU 的时钟数据中读取时间。

(4) 格式 (调度 I/F)

(a) VBA

```
lRet = object.GetClockData(iYear、iMonth、iDay、iDayOfWeek、iHour、iMinute、iSecond)

    Long          lRet      返回值          Output
    Integer       iYear     读取的年的值      Output
    Integer       iMonth    读取的月的值      Output
    Integer       iDay      读取的日的值      Output
    Integer       iDayOfWeek(n) 读取的星期的值  Output
    Integer       iHour     读取的小时的值    Output
    Integer       iMinute   读取的分钟的值    Output
    Integer       iSecond   读取的秒的值      Output
```

(b) Visual C++® .NET (MFC)

```
lRet = object.GetClockData(*lpsYear、*lpsMonth、*lpsDay,*lpsDayOfWeek、*lpsHour、*lpsMinute、
*lpsSecond)

    Long          lRet      返回值          Output
    Short         *lpsYear  读取的年的值      Output
    Short         *lpsMonth 读取的月的值      Output
    Short         *lpsDay   读取的日的值      Output
    Short         *lpsDayOfWeek 读取的星期的值  Output
    Short         *lpsHour  读取的小时的值    Output
    Short         *lpsMinute 读取的分钟的值    Output
    Short         *lpsSecond 读取的秒的值      Output
```

(c) VBScript

```
varRet = object.GetClockData(lpvarYear、lpvarMonth、lpvarDay、lpvarDayOfWeek、lpvarHour、
lpvarMinute、lpvarSecond)

    VARIANT      varRet    返回值 (内容为 LONG 型)      Output
    VARIANT      lpvarYear  读取的年的值 (内容为 SHORT 型)  Output
    VARIANT      lpvarMonth 读取的月的值 (内容为 SHORT 型)  Output
    VARIANT      lpvarDay   读取的日的值 (内容为 SHORT 型)  Output
    VARIANT      lpvarDayOfWeek 读取的星期的值 (内容为 SHORT 型)  Output
    VARIANT      lpvarHour  读取的小时的值 (内容为 SHORT 型)  Output
    VARIANT      lpvarMinute 读取的分钟的值 (内容为 SHORT 型)  Output
    VARIANT      lpvarSecond 读取的秒的值 (内容为 SHORT 型)  Output
```

(d) Visual Basic® .NET

```
iRet = object.GetClockData(iYear、iMonth、iDay、iDayOfWeek、iHour、iMinute、iSecond)
```

Integer	iRet	返回值	Output
short	iYear	读取的年的值	Output
short	iMonth	读取的月的值	Output
short	iDay	读取的日的值	Output
short	iDayOfWeek	读取的星期的值	Output
short	iHour	读取的小时的值	Output
short	iMinute	读取的分钟的值	Output
short	iSecond	读取的秒的值	Output

(e) Visual C++® .NET

```
iRet = object.GetClockData(*lpsYear、*lpsMonth、*lpsDay、*lpsDayOfWeek、*lpsHour、*lpsMinute、*lpsSecond)
```

int	iRet	返回值	Output
short	*lpsYear	读取的年的值	Output
short	*lpsMonth	读取的月的值	Output
short	*lpsDay	读取的日的值	Output
short	*lpsDayOfWeek	读取的星期的值	Output
short	*lpsHour	读取的小时的值	Output
short	*lpsMinute	读取的分钟的值	Output
short	*lpsSecond	读取的秒的值	Output

(f) Visual C#® .NET

```
hResult = object.GetClockData(out sYear、out sMonth、out sDay、out sDayOfWeek、out sHour、out sMinute、out sSecond)
```

int	iRet	返回值	Output
short	sYear	读取的年的值	Output
short	sMonth	读取的月的值	Output
short	sDay	读取的日的值	Output
short	sDayOfWeek	读取的星期的值	Output
short	sHour	读取的小时的值	Output
short	sMinute	读取的分钟的值	Output
short	sSecond	读取的秒的值	Output

5.2 函数的详细内容 (ACT 控件用)
5.2.11 GetClockData(时钟数据读取)

(5) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.GetClockData( *lpsYear、*lpsMonth、*lpsDay、*lpsDayOfWeek、*lpsHour、*lpsMinute、  
*lpsSecond、*lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
SHORT	*lpsYear	读取的年的值	Output
SHORT	*lpsMonth	读取的月的值	Output
SHORT	*lpsDay	读取的日的值	Output
SHORT	*lpsDayOfWeek	读取的星期的值	Output
SHORT	*lpsHour	读取的小时的值	Output
SHORT	*lpsMinute	读取的分钟的值	Output
SHORT	*lpsSecond	读取的秒的值	Output
LONG	*lplRetCode	通信函数的返回值	Output

(6) 说明

- 可编程控制器 CPU 中未设置正确的时钟数据的情况下，将返回出错信息。
- 对于 iYear(lpsYear 或 lpvarYear) 中存储的值，RCPU 及 QCPU(Q 模式) 返回公历 4 位，除此以外的 CPU 时将返回公历低 2 位。
但是，RCPU 及 QCPU(Q 模式) 的范围为从 1980 年开始至 2079 年为止。
- 在 iDayOfWeek(lpsDayOfWeek 或 lpvarDayOfWeek) 中存储的值如下所示。

值	星期
0	星期日
1	星期一
2	星期二
3	星期三
4	星期四
5	星期五
6	星期六

(7) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)

要 点

-
- 如果对 R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
 - 对于 FXCPU，只有时钟功能内置的机型或安装了 RTC 盒的 FXU、FX2C、FX2NC 可以读取时钟数据。上述以外的 FXCPU 将返回出错信息。
 - 进行时钟设置时将产生相当于传送时间的误差，应加以注意。
-

5.2.12 SetClockData(时钟数据写入)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) Ut1Type 控件中可以使用。^{*1}

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 通信路径的使用可否

通信路径的使用可否如下表所示。

通信路径		属性的设置		使用可否	
		ActUnitType	ActProtocolType		
串行通信	连接模块为 R 系列 C24	UNIT_RJ71C24	PROTOCOL_SERIAL	<input type="radio"/>	
	连接模块为 Q 系列 C24	UNIT_QJ71C24		<input type="radio"/>	
	连接模块为 L 系列 C24	UNIT_LJ71C24		<input type="radio"/>	
	连接模块为 FX 扩展端口	UNIT_FX485BD		<input type="radio"/>	
以太网通信	UNIT_RJ71EN71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	○		
	UNIT_RJ71EN71_DIRECT	ROTTOCOL_UDPIP			
	UNIT_QJ71E71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP			
	UNIT_RETHER	PROTOCOL_TCPIP、 PROTOCOL_UDPIP			
	UNIT_RETHER_DIRECT	ROTTOCOL_UDPIP			
	UNIT_LNETHER	PROTOCOL_TCPIP			
	UNIT_LNETHER、 UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP			
	UNIT_QNETHER	PROTOCOL_TCPIP			
	UNIT_QNETHER、 UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP			
	UNIT_NZ2GF_ETB	PROTOCOL_TCPIP			
	UNIT_NZ2GF_ETB、 UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP			
	UNIT_FXETHER	PROTOCOL_TCPIP			
CPU COM 通信	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP	○		
	UNIT_QNCPU	PROTOCOL_SERIAL		<input type="radio"/>	
	UNIT_LNCPU			<input type="radio"/>	
	UNIT_QNMOTION			×	
USB 通信	UNIT_FXCPU	○	<input type="radio"/>		
	UNIT_RUSB		PROTOCOL_USB		<input type="radio"/>
	UNIT_QNUSB				<input type="radio"/>
	UNIT_LNUUSB				<input type="radio"/>
	UNIT_QSUSB				× ^{*1}
	UNIT_QNMOTIONUSB				×

○：可以使用，×：不能使用

*1：访问QSCPU时将发生出错0x010A42A0(访问口令不符)。

5.2 函数的详细内容 (ACT 控件用)
5.2.12 SetClockData(时钟数据写入)

通信路径	属性的设置		使用可否
	ActUnitType	ActProtocolType	
MELSECNET/H 通信	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ *2
CC-Link IE 控制网络通信	UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE 现场网络通信	UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link 通信	UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 通信	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator 通信	UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○
GX Simulator2 通信	UNIT_SIMULATOR2	-	×
MT Simulator2 通信	UNIT_SIMULATOR2	-	×
调制解调器 通信	连接模块为 Q 系列 C24	UNIT_QJ71C24	○
	连接模块为 L 系列 C24	UNIT_LJ71C24	
	连接目标 CPU 为 FXCPU	UNIT_FXCPU	
网关功能通信	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT 透明功能通信	UNIT_RJ71C24、 UNIT_QNCPU、 UNIT_LNCPU、 UNIT_FXCPU、 UNIT_QJ71C24、 UNIT_LJ71C24、 UNIT_GOT_RJ71EN71、 UNIT_GOT_QJ71E71、 UNIT_GOT_RETHER、 UNIT_GOT_QNETHER、 UNIT_GOT_LNETHER、 UNIT_A900GOT、 UNIT_GOT_NZ2GF_ETB、 UNIT_GOT_FXETHER、 UNIT_GOT_FXENET	PROTOCOL_SERIAL、 PROTOCOL_USBGOT	○
	UNIT_GOTETHER_RJ71C24、 UNIT_GOTETHER_QNCPU、 UNIT_GOTETHER_LNCPU、 UNIT_GOTETHER_FXCPU、 UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q 系列总线通信	UNIT_QBF	PROTOCOL_QBF	○
变频器通信	-	COMM_RS232C、 COMM_USB	×

○：可以使用，×：不能使用

*1：访问 QSCPU 时将发生出错 0x010A42A0(访问口令不符)。

*2：自板访问时将返回出错信息。

(3) 功能

向可编程控制器 CPU 的时钟数据写入时间。

(4) 格式 (调度 I/F)

(a) VBA

```
lRet = object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)
```

Long	lRet	返回值	Output
Integer	iYear	写入年的值	Input
Integer	iMonth	写入月的值	Input
Integer	iDay	写入日的值	Input
Integer	iDayOfWeek(n)	写入星期的值	Input
Integer	iHour	写入小时的值	Input
Integer	iMinute	写入分钟的值	Input
Integer	iSecond	写入秒的值	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)
```

Long	lRet	返回值	Output
Short	sYear	写入年的值	Input
Short	sMonth	写入月的值	Input
Short	sDay	写入日的值	Input
Short	sDayOfWeek	写入星期的值	Input
Short	sHour	写入小时的值	Input
Short	sMinute	写入分钟的值	Input
Short	sSecond	写入秒的值	Input

(c) VBScript

```
varRet = object.SetClockData(varYear, varMonth, varDay, varDayOfWeek, varHour, varMinute, varSecond)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varYear	写入年的值 (内容为 SHORT 型)	Input
VARIANT	varMonth	写入月的值 (内容为 SHORT 型)	Input
VARIANT	varDay	写入日的值 (内容为 SHORT 型)	Input
VARIANT	varDayOfWeek	写入星期的值 (内容为 SHORT 型)	Input
VARIANT	varHour	写入小时的值 (内容为 SHORT 型)	Input
VARIANT	varMinute	写入分钟的值 (内容为 SHORT 型)	Input
VARIANT	varSecond	写入秒的值 (内容为 SHORT 型)	Input

5.2 函数的详细内容 (ACT 控件用)
5.2.12 SetClockData(时钟数据写入)

(d) Visual Basic® .NET

```
iRet = object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)
```

	iRet	返回值	Output
Integer			
short	iYear	写入年的值	Input
short	iMonth	写入月的值	Input
short	iDay	写入日的值	Input
short	iDayOfWeek	写入星期的值	Input
short	iHour	写入小时的值	Input
short	iMinute	写入分钟的值	Input
short	iSecond	写入秒的值	Input

(e) Visual C++® .NET

```
iRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)
```

	iRet	返回值	Output
int			
short	sYear	写入年的值	Input
short	sMonth	写入月的值	Input
short	sDay	写入日的值	Input
short	sDayOfWeek	写入星期的值	Input
short	sHour	写入小时的值	Input
short	sMinute	写入分钟的值	Input
short	sSecond	写入秒的值	Input

(f) Visual C++® .NET

```
iRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)
```

	iRet	返回值	Output
int			
short	sYear	写入年的值	Input
short	sMonth	写入月的值	Input
short	sDay	写入日的值	Input
short	sDayOfWeek	写入星期的值	Input
short	sHour	写入小时的值	Input
short	sMinute	写入分钟的值	Input
short	sSecond	写入秒的值	Input

(5) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.SetClockData(sYear, sMonth、sDay、sDayOfWeek、sHour、sMinute、sSecond、
*lpRetCode )
```

HRESULT	hResult	COM 的返回值	Output
SHORT	sYear	写入年的值	Input
SHORT	sMonth	写入月的值	Input
SHORT	sDay	写入日的值	Input
SHORT	sDayOfWeek	写入星期的值	Input
SHORT	sHour	写入小时的值	Input
SHORT	sMinute	写入分钟的值	Input
SHORT	sSecond	写入秒的值	Input
LONG	*lpRetCode	通信函数的返回值	Output

(6) 说明

- 设置的时钟数据不是正确的值的情况下，将返回出错信息。
- 对于 iYear(sYear 或 varYear) 中指定的值，RCPU 及 QCPU(Q 模式) 时公历 4 位有效，除此以外的 CPU 时为公历 2 位有效。
但是，RCPU 及 QCPU(Q 模式) 时的范围为从 1980 年开始至 2079 年为止。
在 RCPU 及 QCPU(Q 模式) 以外的 CPU 中设置 4 位的情况下将发生出错。
- 在 iDayOfWeek(sDayOfWeek 或 varDayOfWeek) 中指定的值如下所示。

值	星期
0	星期日
1	星期一
2	星期二
3	星期三
4	星期四
5	星期五
6	星期六

(7) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)

要 点

-
- 如果对 R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
 - 对于 FXCPU，只有时钟功能内置的机型或安装了 RTC 盒的 FXU、FX2C、FX2NC 可以写入时钟数据。上述以外的 FXCPU 将返回出错信息。
 - 进行时钟设置时将产生相当于传送时间的误差，应加以注意。
-

5.2.13 GetCpuType(可编程控制器 CPU 型号读取)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) Ut1Type 控件中可以使用。

(2) 功能

读取可编程控制器 CPU、网卡、GOT 的型号字符串及型号代码。

(3) 格式 (调度 I/F)

(a) VBA

```
iRet = object.GetCpuType(szCpuName, lCpuType)
```

Long	iRet	返回值	Output
String	szCpuName	可编程控制器 CPU 型号字符串	Output
Long	lCpuType	可编程控制器 CPU 型号代码	Output

(b) Visual C++® .NET (MFC)

```
iRet = object.GetCpuType(*szCpuName, *lplCpuType)
```

Long	iRet	返回值	Output
BSTR	*szCpuName	可编程控制器 CPU 型号字符串	Output
Long	*lplCpuType	可编程控制器 CPU 型号代码	Output

(c) VBScript

```
varRet = object.GetCpuType(varCpuName, lpvarCpuCode)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varCpuName	可编程控制器 CPU 型号字符串 (内容为字符串型)	Output
VARIANT	lpvarCpuCode	可编程控制器 CPU 型号代码 (内容为 LONG 型)	Output

(d) Visual Basic® .NET

```
iRet = object.GetCpuType(szCpuName, ICpuType)
```

Integer	iRet	返回值	Output
String	szCpuName	可编程控制器 CPU 型号字符串	Output
Integer	ICpuType	可编程控制器 CPU 型号代码	Output

(e) Visual C++® .NET

```
iRet = object.GetCpuType (**szCpuName, *iplCpuType)
```

int	iRet	返回值	Output
String	**szCpuName	可编程控制器 CPU 型号字符串	Output
int	*iplCpuType	可编程控制器 CPU 型号代码	Output

(f) Visual C#® .NET

```
iRet = object.GetCpuType (out szCpuName, out iCpuType)
```

int	iRet	返回值	Output
String	szCpuName	可编程控制器 CPU 型号字符串	Output
int	iCpuType	可编程控制器 CPU 型号代码	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.GetCpuType( *szCpuName、*lplCpuType、*lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	*szCpuName	可编程控制器 CPU 型号字符串	Output
LONG	*lplCpuType	可编程控制器 CPU 型号代码	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 将正在进行通信的可编程控制器 CPU 的型号存储到 szCpuName (lpvarCpuName) 中，将型号代码存储到 lCpuType (lplCpuType 或 lpvarCpuCode) 中。
- 可编程控制器 CPU 型号字符串将通过 UNICODE 返回。

(6) CPU 型号字符串及型号代码

通过 GetCpuType 读取的可编程控制器 CPU、网卡、GOT 的型号字符串及型号代码一览如下表所示。

CPU/ 网卡 /GOT 类型	型号			
	CPU/ 自板连接时		GX Simulator 连接时	
	字符串	代码	字符串	代码
R04CPU	R04CPU	4800H		
R08CPU	R08CPU	4801H		
R16CPU	R16CPU	4802H		
R32CPU	R32CPU	4803H		
R120CPU	R120CPU	4804H		
R16MTCPU	R16MTCPU	4C00H		
R32MTCPU	R32MTCPU	4C01H		
Q00JCPU	Q00JCPU	250H	Q00JCPU	250H
Q00UJCPU	Q00UJCPU	260H	Q00UJCPU	260H
Q00CPU	Q00CPU	251H	Q00CPU	251H
Q00UCPU	Q00UCPU	261H	Q00UCPU	261H
Q01CPU	Q01CPU	252H	Q01CPU	252H
Q01UCPU	Q01UCPU	262H	Q01UCPU	262H
Q02CPU	Q02CPU		Q02CPU	
Q02HCPU	Q02HCPU		Q02HCPU	
Q02PHCPU	Q02PHCPU		Q02PHCPU	
Q02UCPU	Q02UCPU	263H	Q02UCPU	263H
Q03UDCPU	Q03UDCPU		Q03UDCPU	
Q03UDECPU	Q03UDECPU		Q03UDECPU	
Q03UDVCPU	Q03UDVCPU	366H		
Q04UDHCPU	Q04UDHCPU		Q04UDHCPU	
Q04UDEHCPU	Q04UDEHCPU		Q04UDEHCPU	
Q04UDVCPU	Q04UDVCPU	367H		
Q06HCPU	Q06HCPU		Q06HCPU	
Q06PHCPU	Q06PHCPU		Q06PHCPU	

5.2 函数的详细内容 (ACT 控件用)
5.2.13 GetCpuType (可编程控制器 CPU 型号读取)

CPU/ 网卡 /GOT 类型	型号				
	CPU/ 自板连接时		GX Simulator 连接时		
	字符串	代码	字符串	代码	
Q06UDHCPU	Q06UDHCPU	26AH	Q06UDHCPU	26AH	
Q06UDEHCPU	Q06UDEHCPU		Q06UDEHCPU		
Q06UDVCPU	Q06UDVCPU	368H	-		
Q10UDHCPU	Q10UDHCPU	266H	Q10UDHCPU	266H	
Q10UDEHCPU	Q10UDEHCPU		Q10UDEHCPU		
Q12HCPU	Q12HCPU	43H	Q12HCPU	43H	
Q12PHCPU					
Q12PRHCPU	Q12PRHCPU	4BH	Q12PRHCPU	4BH	
Q13UDHCPU	Q13UDHCPU	26BH	Q13UDHCPU	26BH	
Q13UDEHCPU	Q13UDEHCPU		Q13UDEHCPU		
Q13UDVCPU	Q13UDVCPU	36AH	-		
Q20UDHCPU	Q20UDHCPU	267H	Q20UDHCPU	267H	
Q20UDEHCPU	Q20UDEHCPU		Q20UDEHCPU		
Q25HCPU	Q25HCPU	44H	Q25HCPU	44H	
Q25PHCPU					
Q25PRHCPU	Q25PRHCPU	4CH	Q25PRHCPU	4CH	
Q26UDHCPU	Q26UDHCPU	26CH	Q26UDHCPU	26CH	
Q26UDEHCPU	Q26UDEHCPU		Q26UDEHCPU		
Q26UDVCPU	Q26UDVCPU	36CH	-		
Q50UDEHCPU	Q50UDEHCPU	26DH			
Q100UDEHCPU	Q100UDEHCPU	26EH			
Q02CPU-A	Q02CPU	141H	Q02CPU-A	141H	
Q02HCPU-A	Q02HCPU				
Q06HCPU-A	Q06HCPU	142H	Q06HCPU-A	142H	
L02SCPU	L02SCPU	543H	-		
L02CPU	L02CPU	541H			
L06CPU	L06CPU	544H			
L26CPU	L26CPU	545H			
L26CPU-BT	L26CPU-BT	542H			
Q12DCCPU-V	Q12DCCPU-V	2043H			
Q24DHCCPU-V	Q24DHCCPU-V	2044H			
Q24DHCCPU-LS	Q24DHCCPU-LS	2045H			
QS001CPU	QS001CPU	230H			
Q172CPU	Q172CPU	2010H			
Q172HCPU	Q172HCPU	2012H	-		
Q173CPU	Q173CPU	2011H			
Q173HCPU	Q173HCPU	2013H			
Q172DCPU	Q172DCPU	2014H			
Q173DCPU	Q173DCPU	2015H			
Q172DSCPU	Q172DSCPU	2018H			
Q173DSCPU	Q173DSCPU	2019H			
FX0	FX0/FX0s	F0H	FX0/FX0s	F0H	
FX0s					
FXON	FXON	8EH	FXON	8EH	
FX1	FX1	F1H	FX1	F1H	

CPU/ 网卡 /GOT 类型	型号				
	CPU/ 自板连接时		GX Simulator 连接时		
	字符串	代码	字符串	代码	
FX1S	FX1S	F2H	FX1S	F2H	
FX1N	FX1N	9EH	FX1N	9EH	
FX1NC					
FXU	FXU/FX2C	8DH	FXU/FX2C	8DH	
FX2C					
FX2N	FX2N/FX2NC	9DH	FX2N/FX2NC	9DH	
FX2NC					
FX3S	FX3S	F5H	-		
FX3G	FX3G	F4H	FX3G	F4H	
FX3GC					
FX3U	FX3UC	F3H	FX3UC	F3H	
FX3UC					
Q80BD-J71LP21-25	Q80BD-J71LP21-25	90H	-		
Q80BD-J71LP21G	Q80BD-J71LP21G				
Q80BD-J71BR11	Q80BD-J71BR11				
Q81BD-J71GF11-T2	Q81BD-J71GF11-T2				
GOT	*1				
A800	0xA8	A8H			
CRnD-7xx/CR75x-D	CRnD-7xx	03H			

*1: 表示 GOT 的产品型号。

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

5.2 函数的详细内容 (ACT 控件用)
5.2.13 GetCpuType(可编程控制器 CPU 型号读取)

5.2.14 SetCpuStatus(远程控制)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 通信路径的使用可否

通信路径的使用可否如下表所示。

通信路径		属性的设置		使用可否
		ActUnitType	ActProtocolType	
串行通信	连接模块为 R 系列 C24	UNIT_RJ71C24	PROTOCOL_SERIAL	<input type="radio"/>
	连接模块为 Q 系列 C24	UNIT_QJ71C24		<input type="radio"/>
	连接模块为 L 系列 C24	UNIT_LJ71C24		<input type="radio"/>
	连接模块为 FX 扩展端口	UNIT_FX485BD		<input type="radio"/> *1 *2
以太网通信		UNIT_RJ71EN71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	<input type="radio"/>
		UNIT_RJ71EN71_DIRECT	ROTOCOL_UDPIP	
		UNIT_QJ71E71	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	
		UNIT_RETHER	PROTOCOL_TCPIP、 PROTOCOL_UDPIP	
		UNIT_RETHER_DIRECT	ROTOCOL_UDPIP	
		UNIT_LNETHETHER	PROTOCOL_TCPIP	
		UNIT_LNETHETHER、 UNIT_LNETHETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER、 UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB、 UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
		UNIT_FXETHER	PROTOCOL_TCPIP	<input type="radio"/> *1
		UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP	
CPU COM 通信	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNCPU	PROTOCOL_SERIAL	<input type="radio"/>
	连接目标 CPU 为 LCPU	UNIT_LNCPU		<input type="radio"/>
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTION		<input type="radio"/>
	连接目标 CPU 为 FXCPU	UNIT_FXCPU		<input type="radio"/> *1
USB 通信	连接目标 CPU 为 RCPU	UNIT_RUSB	PROTOCOL_USB	<input type="radio"/>
	连接目标 CPU 为 QCPU(Q 模式)	UNIT_QNUSB		<input type="radio"/>
	连接目标 CPU 为 LCPU	UNIT_LNUUSB		<input type="radio"/>
	连接目标 CPU 为 QSCPU	UNIT_QSUSB		X *3
	连接目标 CPU 为 Q 运动 CPU	UNIT_QNMOTIONUSB		<input type="radio"/>
	连接目标 CPU 为 FXCPU	UNIT_FXCPU		<input type="radio"/> *1

○：可以使用，×：不能使用

*1：如果进行 PAUSE 指定将返回出错信息。

*2：FX1N、FX1NC、FX2N、FX2NC、FX3S、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

*3：访问 QSCPU 时将发生出错 0x010A42A0 (访问口令不符)。

通信路径	属性的设置		使用可否
	ActUnitType	ActProtocolType	
MELSECNET/H 通信	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ *4
CC-Link IE 控制网络通信	UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE 现场网络通信	UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link 通信	UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 通信	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator 通信	UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○
GX Simulator2 通信	UNIT_SIMULATOR2	-	×
MT Simulator2 通信	UNIT_SIMULATOR2	-	○
调制解调器 通信	连接模块为 Q 系列 C24	PROTOCOL_SERIALMODEM	○
	连接模块为 L 系列 C24	PROTOCOL_SERIALMODEM	
	连接目标 CPU 为 FXCPU	PROTOCOL_TEL	○ *1
网关功能通信	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT 透明功能通信	UNIT_RJ71C24、 UNIT_QNCPU、 UNIT_LNCPU	PROTOCOL_SERIAL、 PROTOCOL_USBGOT	○
	UNIT_FXCPU		○ *1
	UNIT_QJ71C24、 UNIT_LJ71C24、 UNIT_GOT_RJ71EN71、 UNIT_GOT_QJ71E71、 UNIT_GOT_RETHER、 UNIT_GOT_QNETHER、 UNIT_GOT_LNETHER、 UNIT_A900GOT、 UNIT_GOT_NZ2GF_ETB、 UNIT_GOT_FXETHER、 UNIT_GOT_FXENET		○
	UNIT_GOTETHER_RJ71C24、 UNIT_GOTETHER_QNCPU、 UNIT_GOTETHER_LNCPU	PROTOCOL_TCPIP	○
	UNIT_GOTETHER_FXCPU		○ *1
	UNIT_GOTETHER_QBUS		○
Q 系列总线通信	UNIT_QBF	PROTOCOL_QBF	○
变频器通信	-	COMM_RS232C、 COMM_USB	×

○：可以使用，×：不能使用

*1：如果进行 PAUSE 指定将返回出错信息。

*2：FX1N、FX1NC、FX2N、FX2NC、FX3S、FX3U、FX3UC、FX3G、FX3GC 以外的 CPU 将返回出错信息。

*3：访问 QSCPU 时将发生出错 0x010A42A0(访问口令不符)。

*4：自板访问时将返回出错信息。

5.2 函数的详细内容 (ACT 控件用)
5.2.14 SetCpuStatus(远程控制)

(3) 功能

进行可编程控制器 CPU 的远程操作。

(4) 格式 (调度 I/F)

(a) VBA

```
lRet = object.SetCpuStatus(l0peration)
```

Long	lRet	返回值	Output
Long	l0peration	远程 RUN/STOP/PAUSE	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.SetCpuStatus(l0peration)
```

Long	lRet	返回值	Output
Long	l0peration	远程 RUN/STOP/PAUSE	Input

(c) VBScript

```
varRet = object.SetCpuStatus(var0peration)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	var0peration	远程 RUN/STOP/PAUSE(内容为 LONG 型)	Input

(d) Visual Basic® .NET

```
lRet = object.SetCpuStatus(l0peration)
```

Integer	lRet	返回值	Output
Integer	l0peration	远程 RUN/STOP/PAUSE	Input

(e) Visual C++® .NET

```
iRet = object.SetCpuStatus (i0peration)
```

int	iRet	返回值	Output
int	i0peration	远程 RUN/STOP/PAUSE	Input

(f) Visual C#® .NET

```
iRet = object.SetCpuStatus (i0peration)
```

int	iRet	返回值	Output
int	i0peration	远程 RUN/STOP/PAUSE	Input

(5) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

hResult = object.SetCpuStatus(10peration, *lplRetCode)			
HRESULT	hResult	COM 的返回值	Output
LONG	10peration	远程 RUN/STOP/PAUSE	Input
LONG	*lplRetCode	通信函数的返回值	Output

(6) 说明

进行通过 10peration(varOperation) 指定的操作。

指定了下述以外的值时将发生出错。

值	操作
0	远程 RUN
1	远程 STOP
2	远程 PAUSE

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 由于 FXCPU 作为可编程控制器 CPU 未配备 PAUSE 开关，因此通过 SetCpuStatus 指定远程 PAUSE 时将返回出错信息。
- 对 Q 运动 CPU 进行访问，指定 PAUSE 时将返回出错信息。
- 如果对 R 运动 CPU 进行访问，将返回出错信息。

5.2.15 EntryDeviceStatus(软元件的状态监视登录)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

登录进行状态监视的软元件。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.EntryDeviceStatus(szDeviceList, lSize, lMonitorCycle, lData(0))
```

Long	lRet	返回值	Output
String	szDeviceList	登录软元件名列表	Input
Long	lSize	登录软元件点数	Input
Long	lMonitorCycle	状态监视间隔时间	Input
Long	lData(n)	登录软元件值列表	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.EntryDeviceStatus(szDeviceList, lSize, lMonitorCycle, *lplData)
```

Long	lRet	返回值	Output
CString	szDeviceList	登录软元件名列表	Input
Long	lSize	登录软元件点数	Input
Long	lMonitorCycle	状态监视间隔时间	Input
Long	*lplData	登录软元件值列表	Input

(c) VBScript

```
varRet = object.EntryDeviceStatus(varDeviceList, varSize, varMonitorCycle, varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDeviceList	登录软元件名列表 (内容为 BSTR 型)	Input
VARIANT	varSize	登录软元件点数 (内容为 LONG 型)	Input
VARIANT	varMonitorCycle	状态监视间隔时间 (内容为 LONG 型)	Input
VARIANT	varData	登录软元件值列表 (内容为 LONG 型)	Input

(d) Visual Basic® .NET

```
lRet = object.EntryDeviceStatus(szDeviceList, ISize, IMonitorCycle, IData(0))
```

Integer	lRet	返回值	Output
String	szDeviceList	登录软元件名列表	Input
Integer	lSize	登录软元件点数	Input
Integer	lMonitorCycle	状态监视间隔时间	Input
Integer	IData(n)	登录软元件值列表	Input

(e) Visual C++® .NET

```
iRet = object.EntryDeviceStatus(szDeviceList、iSize、iMonitorCycle、*iplData)
```

int	iRet	返回值	Output
String	*szDeviceList	登录软元件名列表	Input
int	iSize	登录软元件点数	Input
int	iMonitorCycle	状态监视间隔时间	Input
int	*iplData	登录软元件值列表	Input

(f) Visual C#® .NET

```
iRet = object.EntryDeviceStatus(szDeviceList、iSize、iMonitorCycle、ref iData[0])
```

int	iRet	返回值	Output
String	szDeviceList	登录软元件名列表	Input
int	iSize	登录软元件点数	Input
int	iMonitorCycle	状态监视间隔时间	Input
int[n]	iData	登录软元件值列表	Input

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.EntryDeviceStatus(szDeviceList、lSize、lMonitorCycle、*lplData、*lplRetCode)
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDeviceList	登录软元件名列表	Input
LONG	lSize	登录软元件点数	Input
LONG	lMonitorCycle	状态监视间隔时间	Input
LONG	*lplData	登录软元件值列表	Input
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 对通过 szDeviceList(varDeviceList) 指定的 lSize(varSize) 容量的软元件群是否处于 lData(lpData 或 varData) 中指定的状态进行确认。
确认时间是通过 lMonitorCycle(varMonitorCycle) 进行指定。
根据状态成立执行用户应用程序的 OnDeviceStatus 函数。
- 对于软元件列表中指定的字符串应通过换行符分开。
最终软元件的后面无需换行符。
(例)

使用 Visual Basic®.NET , VBA, VBScript 时 : “D0” & vbCrLf & “D1” & vbCrLf & “D2”

使用 Visual C++®.NET 时 : D0\nD1\nD2

使用 Visual C#®.NET 时 : D0\nD1\nD2

- 在 lSize(varSize) 中可指定的最大软元件点数为 20 点。
- 对于 lMonitorCycle(varMonitorCycle), 应以 1 秒~1 小时的范围 (以 1 ~ 3600 的秒单位进行设置) 进行指定。
指定超出此范围的值时将发生出错。
- 将登录软元件值列表存储到 lData(lpData 或 varData) 中。

(6) 软元件指定方法

软元件的指定方法如下所示。

(例 1) 按下述方式进行指定的情况下(点数为3点)

使用 Visual Basic[®].NET, VBA, VBScript 时 : “M0” & vbLf & “D0” & vbLf & “K8M0”

使用 Visual C++[®].NET 时 : M0\nD0\nK8M0

使用 Visual C#[®].NET 时 : M0\nD0\nK8M0

高位2字节	低位2字节
不使用。 (存储0。)	M0
D0	
M16 ~ M31*1	M0 ~ M15*1

(例 2) 指定了双字软元件时

使用 Visual Basic[®].NET, VBA, VBScript 时 : “LTN0” & vbLf & “LTN1” & vbLf & “LTN2”

使用 Visual C++[®].NET 时 : LTN0\nLTN1\nLTN2

使用 Visual C#[®].NET 时 : LTN0\nLTN1\nLTN2

高位2字节	低位2字节
LTN0 的 H	LTN0 的 L
LTN1 的 H	LTN1 的 L
LTN2 的 H	LTN2 的 L

(例 3) 指定包含 FXCPU 的 CN200 以后的软元件的情况下(合计点数为3点)*2

使用 Visual Basic[®].NET, VBA, VBScript 时 : “D0” & vbLf & “CN200” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nCN200\nD1

使用 Visual C#[®].NET 时 : D0\nCN200\nD1

高位2字节	低位2字节
不使用。 (存储0。)	D0
CN200 的 H	CN200 的 L
不使用。 (存储0。)	D1

(例 4) 指定包含 FD 的软元件的情况下(合计点数为3点)

使用 Visual Basic[®].NET, VBA, VBScript 时 : “D0” & vbLf & “FD0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nFD0\nD1

使用 Visual C#[®].NET 时 : D0\nFD0\nD1

高位2字节	低位2字节
不使用。 (存储0。)	D0
	FD0 的 LL
	D1

*1: 按照软元件的编号顺序从低位开始存储。

*2: 对于 FXCPU 的 CN200 以后, 随机读取以1点读取2字。

(例5) 指定包含EG的软元件的情况下(合计点数为3点)

在EG0中分配了8位软元件(SHARP生产的可编程控制器的E0000、E0001)的示例如下所示。

使用Visual Basic[®].NET,VBA,VBScript时：“D0”&vbLf&“EG0”&vbLf&“D1”

使用Visual C++[®].NET时：D0\nEG0\nD1

使用Visual C#[®].NET时：D0\nEG0\nD1

高位2字节	低位2字节
	D0
不使用。 (存储0。)	EG0
	(E0001) (E0000)
	D1

(7) 返回值

正常结束：返回0。

非正常结束：返回0以外。(☞ 496页第7章 出错代码)

(8) 进行字软元件状态监视情况下的注意事项

对字软元件为负的值-1～-32768 (FFFFH～8000H)的状态进行监视的情况下，应设置EntryDeviceStatus的监视软元件值的高位2字节中存储了“0”的65535～32768 (0000FFFFH～00008000H)。

(例) 对D0变为“-10”进行状态监视的情况下

应将“-10(FFFFFF6H)”的高位2字节中存储了“0”的值“65526(0000FFF6H)”设置为监视软元件值。

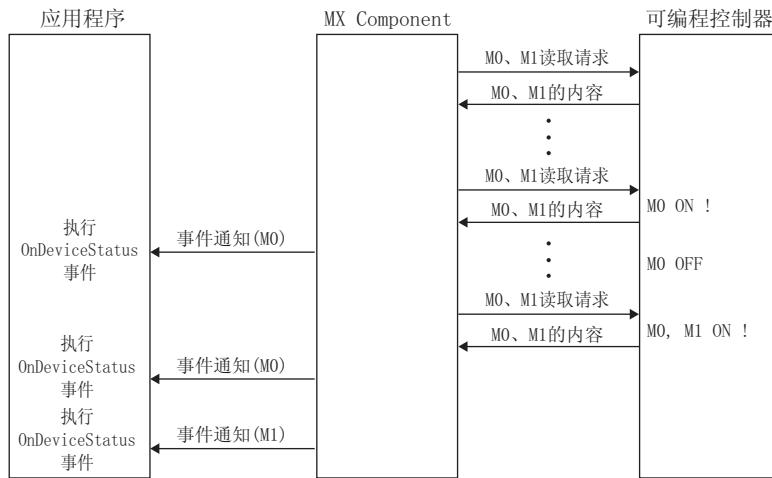
由于可编程控制器CPU的字软元件为WORD型，而EntryDeviceStatus的监视软元件值的类型为LONG型，因此将可编程控制器CPU的当前值与EntryDeviceStatus的监视软元件值进行了比较的情况下，值不匹配，因此需要进行上述设置。(使用位软元件及双字软元件时，本注意事项不适用。)

关于本注意事项的编程示例，请参阅下述内容。

☞ 534页附录3 关于超时时间

要 点

- 根据计算机性能、执行中的应用程序负载、与可编程控制器通信所需时间等的条件，有可能在指定的状态监视间隔时间内无法进行软元件的状态监视。此外，在同时使用控件的其它函数的情况下，也可能导致无法在指定的状态监视间隔时间内进行软元件的状态监视。
- 对于 lData(lplData 或 varData)，应预留出相当于 lSize(varSize) 中指定点数的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 在状态监视中执行了 EntryDeviceStatus 的情况下，将发生出错。更改状态监视条件的情况下，应执行 FreeDeviceStatus 之后，再次执行 EntryDeviceStatus。
- 多个软元件同时发生了状态变化的情况下，每次状态变化时，均将执行 OnDeviceStatus 事件。
(例：监视 M0 的情况下)



- 该功能是控件定期执行软元件随机读取，确认状态成立的功能。因此，不是将可编程控制器CPU的软元件的状态成立通知到MX Component中的功能。所以，根据指定的状态监视间隔时间，控件有可能无法确认可编程控制器CPU的软元件的状态成立。

5.2.16 FreeDeviceStatus(软元件的状态监视登录解除)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

对通过 EntryDeviceStatus 登录的执行状态监视的软元件进行解除。

(3) 格式 (调度 I/F)

(a) Visual C++® .NET (MFC)、VBA

```
lRet = object.FreeDeviceStatus()
```

Long	lRet	返回值	Output
------	------	-----	--------

(b) VBScript

```
varRet = object.FreeDeviceStatus()
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
---------	--------	--------------------	--------

(c) Visual Basic® .NET

```
IRet = object.FreeDeviceStatus()
```

Integer	IRet	返回值	Output
---------	------	-----	--------

(d) Visual C++® .NET

```
iRet = object.FreeDeviceStatus()
```

int	iRet	返回值	Output
-----	------	-----	--------

(e) Visual C#® .NET

```
iRet = object.FreeDeviceStatus()
```

int	iRet	返回值	Output
-----	------	-----	--------

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.FreeDeviceStatus( *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

对通过 EntryDeviceStatus 函数设置的执行状态监视的软元件进行解除。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。 ( 496 页第 7 章 出错代码)

5.2.17 OnDeviceStatus(事件通知)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

在通过 EntryDeviceStatus 函数登录的软元件条件成立时执行此功能。

(3) 格式 (调度 I/F)

(a) VBA

```
object.OnDeviceStatus(szDevice, lData, lReturnCode)
    String          szDevice      条件成立的软元件名      Input
    Long           lData        条件成立的软元件名      Input
    Long           lReturnCode  条件检查处理的返回值     Input
```

(b) Visual C++® .NET (MFC)

```
object.OnDeviceStatus(*szDevice, lData, lReturnCode)
    LPCTSTR         *szDevice    条件成立的软元件名      Input
    Long            lData       条件成立的软元件名      Input
    Long            lReturnCode 条件检查处理的返回值     Input
```

(c) VBScript

```
object.OnDeviceStatus(varDevice, varData, varReturnCode)
    VARIANT         varDevice    条件成立的软元件名 (内容为 BSTR 型) Input
    VARIANT         varData     条件成立的软元件名 (内容为 LONG 型) Input
    VARIANT         varReturnCode 条件检查处理的返回值 (内容为 LONG 型) Input
```

(d) Visual Basic® .NET

```
Private Sub object_OnDeviceStatus(ByVal sender As System.Object, ByVal e As
objectLib._objectIEvents_OnDeviceStatusEvent)
    sender        事件的发生源          Output
    e            事件的数据          Output
    e 的成员如下所示。
        e.szDevice    条件成立的软元件名
        e.lReturnCode 条件成立的软元件值
        e.lData       条件检查处理的返回值
```

(e) Visual C++® .NET

```
private: System::Void object_OnDeviceStatus (System::Object * sender,
objectLib._objectIEvents_OnDeviceStatusEvent * e)
    sender        事件的发生源          Output
    e            事件的数据          Output
    e 的成员如下所示。
        e->szDevice    条件成立的软元件名
        e->lReturnCode 条件成立的软元件值
        e->lData       条件检查处理的返回值
```

(f) Visual C#® .NET

```
private void object_OnDeviceStatus
(object sender, object._IActUtlTypeEvents_OnDeviceStatusEvent e)
{
    sender          事件的发生源
    e              事件的数据
    e 的成员如下所示。
    e->szDevice    条件成立的软元件名
    e->lReturnCode   条件检查处理的返回值
    e->lData        条件成立的软元件值
}
```

Output
Output

(4) 格式(自定义 I/F)

(a) Visual C++® .NET (MFC)

object.OnDeviceStatus(szDevice、lData、lReturnCode、*lplRetCode)

LPCTSTR	szDevice	条件成立的软元件名	Input
LONG	lData	条件成立的软元件值	Input
LONG	lReturnCode	条件检查处理的返回值	Input
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 在通过 EntryDeviceStatus 函数登录的软元件条件成立时对应用程序进行事件通知。
通过在用户应用程序侧安装本函数，当登录的软元件条件成立时，可以接受事件。
- 在 lData(varData) 中，输入通过 EntryDeviceStatus 登录的软元件值。
(例) 通过字软元件对“-1”的值进行监视的情况下
在 EntryDeviceStatus 中将 (0000FFFFH) 设置为登录软元件值。
可编程控制器 CPU 的对象字软元件变为“-1”(FFFFH) 的状态时，将执行 OnDeviceStatus，在 lData(varData) 中输入 (0000FFFFH)。

(6) 返回值

无



-
- 在用户应用程序内进行下述设置的情况下，即使 EntryDeviceStatus 函数中登录的软元件的条件成立也不发生 OnDeviceStatus 的事件。变为事件发生等待的情况下，在下述设置结束之前控制将不返回到控件中，软元件管理处理将停止，因此应加以注意。
 - 通过 Visual Basic® .NET、VBA(Excel) 创建的用户应用程序
在用户应用程序内显示信息框。
在用户应用程序内显示 InputBox/OutputBox。
 - 通过 Visual Basic® .NET、Visual C++® .NET、Visual C#® .NET、VBA(Excel, Access) 及 VBScript 创建的用户程序
在用户应用程序内使用 Sleep 处理、WaitForSingleObject 函数等的待机函数。
 - 关于浏览设置的安装方法，请参阅 VB.NET/VC.NET 的浏览设置用的样本程序 (481 页 6.4 节 ~ 488 页 6.5 节)
-

5.2.18 ReadDeviceBlock2(软元件的批量读取)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

以 2 字节数据进行软元件的批量读取。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.ReadDeviceBlock2(szDevice、lSize、aData(0))
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Long	lSize	读取点数	Input
Integer	iData(n)	读取的软元件值	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.ReadDeviceBlock2(szDevice、lSize、*lpsData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Long	lSize	读取点数	Input
Short	*lpsData	读取的软元件值	Output

(c) VBScript

```
varRet = object.ReadDeviceBlock2(varDevice、varSize、lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	varSize	读取点数 (内容为 LONG 型)	Input
VARIANT	lpvarData	读取的软元件值 (内容为 SHORT 型)	Output

(d) Visual Basic® .NET

```
lRet = object.ReadDeviceBlock2(szDevice、lSize、sData(0))
```

Integer	lRet	返回值	Output
String	szDevice	软元件名	Input
Integer	lSize	读取点数	Input
short	sData(n)	读取的软元件值	Output

(e) Visual C++® .NET

```
iRet = object.ReadDeviceBlock2(*szDevice、iSize、*lpsData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
int	iSize	读取点数	Input
short	*lpsData	读取的软元件值	Output

(f) Visual C#® .NET

```
iRet = object.ReadDeviceBlock2(szDevice、iSize、out sData[0])
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
int	iSize	读取点数	Input
short[n]	sData	读取的软元件值	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.ReadDeviceBlock2( szDevice、lSize、*lpsData、*lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名	Input
LONG	lSize	读取点数	Input
SHORT	*lpsData	读取的软元件值	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 从通过 szDevice(varDevice) 指定的软元件开始，将 lSize(varSize) 容量的软元件值作为 SHORT 型的数据进行批量读取。
- 读取的软元件值将被存储到 iData(lpsData 或 lpvarData) 中。
- 对于 iData(lpsData 或 lpvarData)，应预留出 lSize(varSize) 容量以上的数组。

(6) 软元件指定方法

软元件的指定方法如下所示。

〈指定位软元件时〉

(例) 从 M0 开始以 16 点单位读取 3 点 (3 字节)。

2 字节
M0 ~ M15*1
M16 ~ M31*1
M32 ~ M47*1

〈指定字软元件时〉

(例) 从 D0 开始读取 3 点。

2 字节
D0
D1
D2

〈指定双字软元件时〉

(例) 从 LTN0 开始读取 3 点。

2 字节
LTN0 的 L
LTN1 的 L
LTN2 的 L

〈指定 FXCPU 的 CN200 以后时〉

(例) 从 CN200 开始读取 6 点。*2

2 字节
CN200 的 L
CN200 的 H
CN201 的 L
CN201 的 H
CN202 的 L
CN202 的 H

*1：按照软元件的编号顺序从低位开始存储。

*2：对于 FXCPU 的 CN200 以后，以 2 点进行 2 字读取。以 1 点进行时将发生出错。

< 指定 FD 软元件时 (4 字软元件) >

(例) 从 FD0 开始读取 6 点。

2 字节
FD0 的 LL
FD0 的 LH
FD0 的 HL
FD0 的 HH
FD1 的 LL
FD1 的 LH

< 网关软元件中分配的 8 位软元件 >

(例) 从 EG0 开始读取 4 点。

(EG0 ~ EG3 中分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000 ~ E0007) 的情况下)

2 字节	
EG0	
(E0001)	(E0000)
EG1	
(E0003)	(E0002)
EG2	
(E0005)	(E0004)
EG3	
(E0007)	(E0006)

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 在 lSize(varSize) 中可指定的最大读取点数为满足下述条件的范围。
读取开始软元件编号 + 读取点数 ≤ 最终软元件编号
- 指定位软元件时, 对软元件编号可以以 16 的倍数进行指定。
- 对于 iData(lpsData 或 lpvarData), 应预留出相当于 lSize(varSize) 中指定点数的存储器区域。
没有存储器区域的情况下, 有可能会发生应用程序出错等严重现象。

5.2.19 WriteDeviceBlock2(软元件的批量写入)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1: 变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

以 2 字节数据进行软元件的批量写入。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.WriteDeviceBlock2(szDevice, lSize, iData(0))
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Long	lSize	写入点数	Input
Integer	iData(n)	写入的软元件值	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.WriteDeviceBlock2(szDevice, lSize, *lpsData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Long	lSize	写入点数	Input
Short	*lpsData	写入的软元件值	Input

(c) VBScript

```
varRet = object.WriteDeviceBlock2(varDevice, varSize, varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	varSize	写入点数 (内容为 LONG 型)	Input
VARIANT	varData	写入的软元件值 (内容为 SHORT 型)	Input

(d) Visual Basic® .NET

```
IRet = object.WriteDeviceBlock2(szDevice, ISize, sData(0))
```

Integer	IRet	返回值	Output
String	szDevice	软元件名	Input
Integer	ISize	写入点数	Input
short	sData(n)	写入的软元件值	Input

(e) Visual C++® .NET

```
iRet = object.WriteDeviceBlock2(*szDevice, iSize, *lpsData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
int	iSize	写入点数	Input
short	*lpsData	写入的软元件值	Input

(f) Visual C#® .NET

```
iRet = object.ReadDeviceBlock2(szDevice, iSize, out sData[0])  
int iRet 返回值 Output  
String szDevice 软元件名 Input  
int iSize 写入点数 Input  
short[n] sData 写入的软元件值 Input
```

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.WriteDeviceBlock2(szDevice, lSize, *lpsData, *lplRetCode )  
HRESULT hResult COM 的返回值 Output  
BSTR szDevice 软元件名 Input  
LONG lSize 写入点数 Input  
SHORT *lpsData 写入的软元件值 Input  
LONG *lplRetCode 通信函数的返回值 Output
```

(5) 说明

- 从通过 szDevice(varDevice) 指定的软元件开始，将 lSize(varSize) 容量的软元件值进行批量写入。
- 写入的软元件值将被存储到 iData(lpsData 或 varData) 中。
- 对于 iData(lpsData 或 varData)，应预留出 lSize(varSize) 以上的数组。

(6) 软元件指定方法

软元件的指定方法如下所示。按照软元件的编号顺序，从低位开始存储。

〈指定位软元件时〉

(例) 从 M0 开始以 16 点单位写入 3 点 (3 字)。

2 字节
M0 ~ M15*1
M16 ~ M31*1
M32 ~ M47*1

〈指定字软元件时〉

(例) 从 D0 开始写入 3 点。

2 字节
D0
D1
D2

〈指定双字软元件时〉

(例) 从 LTN0 开始写入 3 点。

2 字节
LTN0 的 L
LTN1 的 L
LTN2 的 L

〈指定 FXCPU 的 CN200 以后时〉

(例) 从 CN200 开始写入 6 点。*2

2 字节
CN200 的 L
CN200 的 H
CN201 的 L
CN201 的 H
CN202 的 L
CN202 的 H

*1: 按照软元件的编号顺序从低位开始存储。

*2: 对于 FXCPU 的 CN200 以后，以 2 点写入 2 字。以 1 点写入时将发生出错。

<指定 FD 软元件时 (4 字软元件)>

(例) 从 FD0 开始写入 6 点。

2 字节
FD0 的 LL
FD0 的 LH
FD0 的 HL
FD0 的 HH
FD1 的 LL
FD1 的 LH

<网关软元件中分配的 8 位软元件>

(例) 从 EG0 开始写入 4 点。

(EG0 ~ EG3 中分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000 ~ E0007) 的情况下)

2 字节	
EG0	
(E0001)	(E0000)
EG1	
(E0003)	(E0002)
EG2	
(E0005)	(E0004)
EG3	
(E0007)	(E0006)

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 在 lSize(varSize) 中可指定的最大读取点数为满足下述条件的范围。
写入开始软元件编号 + 写入点数 ≤ 最终软元件编号
- 指定位软元件时, 对软元件编号可以以 16 的倍数进行指定。
- 对于 iData(lpsData 或 varData), 应预留出相当于 lSize(varSize) 中指定点数的存储器区域。
没有存储器区域的情况下, 有可能会发生应用程序出错等严重现象。
- 向长定时器软元件 (LT) 及累积长定时器软元件 (LST) 写入时, 不能使用 WriteDeviceBlock2。
请使用 WriteDeviceRandom2 或 SetDevice2。

5.2.20 ReadDeviceRandom2(软元件的随机读取)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

以 2 字节数据进行软元件的随机读取。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.ReadDeviceRandom2(szDeviceList, lSize, iData(0))
```

Long	lRet	返回值	Output
String	szDeviceList	软元件名	Input
Long	lSize	读取点数	Input
Integer	iData(n)	读取的软元件值	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.ReadDeviceRandom2(szDeviceList, lSize, *lpsData)
```

Long	lRet	返回值	Output
CString	szDeviceList	软元件名	Input
Long	lSize	读取点数	Input
Short	*lpsData	读取的软元件值	Output

(c) VBScript

```
varRet = object.ReadDeviceRandom2(varDeviceList, varSize, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDeviceList	软元件名 (内容为字符串型)	Input
VARIANT	varSize	读取点数 (内容为 LONG 型)	Input
VARIANT	lpvarData	读取的软元件值 (内容为 SHORT 型)	Output

(d) Visual Basic® .NET

```
lRet = object.ReadDeviceRandom2(szDeviceList, ISize, sData(0))
```

Integer	lRet	返回值	Output
String	szDeviceList	软元件名	Input
Integer	ISize	读取点数	Input
short	sData(n)	读取的软元件值	Output

(e) Visual C++® .NET

```
iRet = object.ReadDeviceRandom(*szDeviceList, iSize, *lpsData)
```

int	iRet	返回值	Output
String	*szDeviceList	软元件名	Input
int	iSize	读取点数	Input
short	*lpsData	读取的软元件值	Output

(f) Visual C#® .NET

```
iRet = object.ReadDeviceBlock2(szDevice、iSize、out sData[0])
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
int	iSize	读取点数	Input
short[n]	sData	读取的软元件值	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.ReadDeviceRandom2( szDeviceList、lSize、*lpsData、*lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDeviceList	软元件名	Input
LONG	lSize	读取点数	Input
SHORT	*lpsData	读取的软元件值	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 将通过 szDeviceList(varDeviceList) 指定的软元件群，只读取 lSize(varSize) 容量的软元件值。
- 读取的软元件值将被存储到 iData(lpsData 或 lpvarData) 中。
- 对于软元件列表中指定的字符串应通过换行符分开。

最终软元件的后面无需换行符。

(例)

使用 Visual Basic® .NET , VBA, VBScript 时 : “D0” & vbLf & “D1” & vbLf & “D2”

使用 Visual C++® .NET 时 : D0\nD1\nD2

使用 Visual C#® .NET 时 : D0\nD1\nD2

- 对于 iData(lpsData 或 lpvarData)，应预留出 lSize(varSize) 以上的数组。

(6) 软元件指定方法

软元件的指定方法如下所示。

(例 1) 按下述方式进行指定的情况下 (点数为 3 点)

使用 Visual Basic® .NET , VBA, VBScript 时 : “M0” & vbLf & “D0” & vbLf & “K8M0”

使用 Visual C++® .NET 时 : M0\nD0\nK8M0

使用 Visual C#® .NET 时 : M0\nD0\nK8M0



*1: 仅将 “M0” 1 点作为读取软元件对象，软元件值将存储 “0” 或 “1”。

*2: 按照软元件的编号顺序从低位开始存储。

高位 2 字节 M16 ~ M31 不被读取。

(例2) 指定了双字软元件时

使用 Visual Basic[®].NET , VBA, VBScript 时 : “LTN0” & vbLf & “LTN1” & vbLf & “LTN2”

使用 Visual C++[®].NET 时 : LTN0\nLTN1\nLTN2

使用 Visual C#[®].NET 时 : LTN0\nLTN1\nLTN2

2字节
LTN0 的 L
LTN1 的 L
LTN2 的 L

(例3) 指定包含 FXCPU 的 CN200 以后的软元件的情况下(合计点数为3点)^{*3}

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “CN200” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nCN200\nD1

使用 Visual C#[®].NET 时 : D0\nCN200\nD1

2字节
D0
CN200 的 L *3
D1

*3: 对于 FXCPU 的 CN200 以后，在 ReadDeviceRandom2 中以 1 点读取指定软元件的 L(低位2字节)。
指定软元件的 H(高位2字节)不被读取。

(例4) 指定包含 FD 的软元件的情况下(合计点数为3点)

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “FD0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nFD0\nD1

使用 Visual C#[®].NET 时 : D0\nFD0\nD1

2字节
D0
FD0 的 LL
D1

(例5) 指定包含 EG 的软元件的情况下(合计点数为3点)

在 EG0 中分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000, E0001) 的示例如下所示。

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “EG0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nEG0\nD1

使用 Visual C#[®].NET 时 : D0\nEG0\nD1

2字节
D0
EG0
(E0001) (E0000)
D1

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- lSize(varSize) 中可指定的最大读取点数为 0x7FFFFFFF 点。
- 对于 iData(lpsData 或 lpvarData)，应预留出相当于 lSize(varSize) 中指定点数的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 指定了双字软元件的情况下，在 ReadDeviceRandom2 中仅存储低位 1 字（2 字节）容量的数据。（不发生出错。）进行双字软元件的读取的情况下，应使用 ReadDeviceRandom 或 GetDevice。

5

5.2 函数的详细内容 (ACT 控件用)
5.2.20 ReadDeviceRandom2(软元件的随机读取)

5.2.21 WriteDeviceRandom2(软元件的随机写入)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

以 2 字节数据进行软元件的随机写入。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.WriteDeviceRandom2(szDeviceList, lSize, iData(0))
```

Long	lRet	返回值	Output
String	szDeviceList	软元件名	Input
Long	lSize	写入点数	Input
Integer	iData(n)	写入的软元件值	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.WriteDeviceRandom2(szDeviceList, lSize, *lpsData)
```

Long	lRet	返回值	Output
CString	szDeviceList	软元件名	Input
Long	lSize	写入点数	Input
Short	*lpsData	写入的软元件值	Input

(c) VBScript

```
varRet = object.WriteDeviceRandom2(varDeviceList, varSize, varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDeviceList	软元件名 (内容为字符串型)	Input
VARIANT	varSize	写入点数 (内容为 LONG 型)	Input
VARIANT	varData	写入的软元件值 (内容为 SHORT 型)	Input

(d) Visual Basic® .NET

```
lRet = object.WriteDeviceRandom2(szDeviceList, ISize, sData(0))
```

Integer	lRet	返回值	Output
String	szDeviceList	软元件名	Input
Integer	ISize	写入点数	Input
short	sData(n)	写入的软元件值	Input

(e) Visual C++® .NET

```
iRet = object.WriteDeviceRandom2(*szDeviceList, iSize, *ipsData)
```

int	iRet	返回值	Output
String	*szDeviceList	软元件名	Input
int	iSize	写入点数	Input
short	*ipsData	写入的软元件值	Input

(f) Visual C#® .NET

```
iRet = object.WriteDeviceRandom2(szDevice, iSize, ref sData[0])
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
int	iSize	写入点数	Input
short[n]	sData	写入的软元件值	Input

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.WriteDeviceRandom2( szDeviceList, lSize, *lpsData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDeviceList	软元件名	Input
LONG	lSize	写入点数	Input
SHORT	*lpsData	写入的软元件值	Input
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

- 将通过 szDeviceList(varDeviceList) 指定的软元件群，只写入 lSize(varSize) 容量的软元件值。
- 将写入的软元件值存储到 iData(lpsData 或 varData) 中。
- 对于软元件列表中指定的字符串应通过换行符分开。最终软元件的后面无需换行符。
(例)

使用 Visual Basic® .NET , VBA, VBScript 时 : “D0” & vbLf & “D1” & vbLf & “D2”

使用 Visual C++® .NET 时 : D0\nD1\nD2

使用 Visual C#® .NET 时 : D0\nD1\nD2

- 对于 iData(lpsData 或 varData)，应预留出 lSize(varSize) 以上的数组。

(6) 软元件指定方法

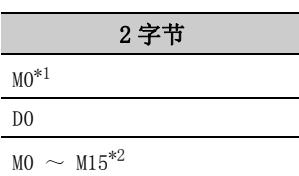
软元件的指定方法如下所示。

(例 1) 按下述方式进行指定的情况下 (点数为 3 点)

使用 Visual Basic® .NET , VBA, VBScript 时 : “M0” & vbLf & “D0” & vbLf & “K8M0”

使用 Visual C++® .NET 时 : M0\nD0\nK8M0

使用 Visual C#® .NET 时 : M0\nD0\nK8M0



*1: 仅将 “M0” 1 点作为写入软元件对象，设置的 2 字节数据的最低位将成为软元件值的写入值。

*2: 按照软元件的编号顺序从低位开始存储。

高位 2 字节 M16 ~ M31 中被写入 “0”。

(例2) 指定了双字软元件时

使用 Visual Basic[®].NET , VBA, VBScript 时 : “LTN0” & vbLf & “LTN1” & vbLf & “LTN2”

使用 Visual C++[®].NET 时 : LTN0\nLTN1\nLTN2

使用 Visual C#[®].NET 时 : LTN0\nLTN1\nLTN2

2字节
LTN0 的 L
LTN1 的 L
LTN2 的 L

(例3) 指定包含 FXCPU 的 CN200 以后的软元件的情况下 (合计点数为 3 点)^{*3}

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “CN200” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nCN200\nD1

使用 Visual C#[®].NET 时 : D0\nCN200\nD1

2字节
D0
CN200 的 L *3
D1

*3: 对于 FXCPU 的 CN200 以后, 在 WriteDeviceRandom2 中以 1 点写入到指定软元件的 L(低位 2 字节)中。指定软元件的 H(高位 2 字节)中将被写入“0”。

(例4) 指定包含 FD 的软元件的情况下 (合计点数为 3 点)

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “FD0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nFD0\nD1

使用 Visual C#[®].NET 时 : D0\nFD0\nD1

2字节
D0
FD0 的 LL
D1

(例5) 指定包含 EG 的软元件的情况下 (合计点数为 3 点)

在 EG0 中被分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000、E0001) 的示例如下所示。

使用 Visual Basic[®].NET , VBA, VBScript 时 : “D0” & vbLf & “EG0” & vbLf & “D1”

使用 Visual C++[®].NET 时 : D0\nEG0\nD1

使用 Visual C#[®].NET 时 : D0\nEG0\nD1

2字节
D0
EG0
(E0001) (E0000)
D1

(7) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。 (参见 496 页第 7 章 出错代码)

要 点

- lSize(varSize) 中可指定的最大写入点数为 0x7FFFFFFF 点。
- 对于 iData(lpsData 或 varData)，应预留出相当于 lSize(varSize) 中指定点数的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 指定了双字软元件的情况下，在 WriteDeviceRandom2 中对低位 1 字（2 字节）容量的数据进行写入，在高位 1 字（2 字节）容量的数据中写入“0”。
进行双字软元件的写入的情况下，应使用 WriteDeviceRandom 或 SetDevice。
- 如果对 Q 运动 CPU 进行访问，将返回出错信息。

5.2 函数的详细内容 (ACT 控件用)
5.2.21 WriteDeviceRandom2(软元件的随机写入)

5.2.22 SetDevice2(软元件数据的设置)

(1) 可以使用的控件

在 Act (ML)ProgType 及 Act (ML)UtlType 控件中可以使用。*1

*1：变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

以 2 字节数据进行 1 点软元件的设置。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.SetDevice2(szDevice、iData)
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Integer	iData	设置数据	Input

(b) Visual C++® .NET (MFC)

```
lRet = object.SetDevice2(szDevice、sData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Short	sData	设置数据	Input

(c) VBScript

```
varRet = object.SetDevice2(varDevice、varData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	varData	设置数据 (内容为 SHORT 型)	Input

(d) Visual Basic® .NET

```
IRet = object.SetDevice2(szDevice、sData)
```

Integer	IRet	返回值	Output
String	szDevice	软元件名	Input
short	sData	设置数据	Input

(e) Visual C++® .NET

```
iRet = object.SetDevice2(*szDevice、sData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
short	sData	设置数据	Input

(f) Visual C#® .NET

```
iRet = object.SetDevice2(szDevice、sData)
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
short	sData	设置数据	Input

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.SetDevice2( szDevice、sData、*lpIRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名	Input
SHORT	sData	设置数据	Input
LONG	*lpIRetCode	通信函数的返回值	Output

(5) 说明

- 对通过 szDevice(varDevice) 指定的 1 点软元件进行 iData(sData 或 varData) 中指定的操作。
- 位软元件设置时，iData 值 (sData 值或 varData 值) 的最低位的位将有效。

(6) 软元件指定方法

软元件的指定方法如下所示。

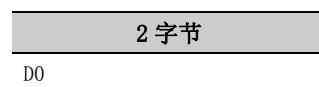
〈指定位软元件时〉

(例) M0



〈指定字软元件时〉

(例) D0



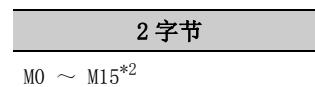
〈指定双字软元件时〉

(例) LTN0



〈指定双字软元件时 (位数指定)〉

(例) K8M0



〈指定 FXCPU 的 CN200 以后时〉

(例) CN200



〈指定网关软元件时〉

(例) 在 EGO 中被分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000、E0001) 的情况下



*1：仅将“M0”1点作为写入软元件的对象，设置的2字节数据的最低位将成为软元件值的写入值。

*2：按照软元件的编号顺序从低位开始存储。

高位2字节 M16 ~ M31 中将被写入“0”。

*3：对于 FXCPU 的 CN200 以后，在 SetDevice2 中以 1 点对指定软元件的 L(低位 2 字节) 进行写入。指定软元件的 H(高位 2 字节) 中将被写入“0”。

(7) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 指定了双字软元件的情况下，在 SetDevice2 中对低位 1 字（2 字节）容量的数据进行写入，在高位 1 字（2 字节）容量的数据中写入“0”。
进行双字软元件的写入的情况下，应使用 WriteDeviceRandom 或 SetDevice。

5.2.23 GetDevice2(软元件数据的获取)

(1) 可以使用的控件

在 Act (ML) ProgType 及 Act (ML) UtlType 控件中可以使用。^{*1}

*1: 变频器通信 / 机器人控制器通信时无法使用。

(2) 功能

以 2 字节数据获取 1 点软元件的数据。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.GetDevice2(szDevice, iData)
```

Long	lRet	返回值	Output
String	szDevice	软元件名	Input
Integer	iData	获取数据	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.GetDevice2(szDevice, lSize, *lpsData)
```

Long	lRet	返回值	Output
CString	szDevice	软元件名	Input
Short	*lpsData	获取数据	Output

(c) VBScript

```
varRet = object.GetDevice2(varDevice, lpvarData)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varDevice	软元件名 (内容为字符串型)	Input
VARIANT	lpvarData	获取数据 (内容为 SHORT 型)	Output

(d) Visual Basic® .NET

```
IRet = object.GetDevice2(szDevice, sData)
```

Integer	IRet	返回值	Output
String	szDevice	软元件名	Input
short	sData	获取数据	Output

(e) Visual C++® .NET

```
iRet = object.GetDevice(*szDevice, *ipsData)
```

int	iRet	返回值	Output
String	*szDevice	软元件名	Input
short	*spsData	获取数据	Output

(f) Visual C#® .NET

```
iRet = object.GetDevice2(szDevice, out sData)
```

int	iRet	返回值	Output
String	szDevice	软元件名	Input
short	sData	获取数据	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.GetDevice2( szDevice, *lpsData, *lplRetCode )
```

HRESULT	hResult	COM 的返回值	Output
BSTR	szDevice	软元件名	Input
SHORT	*lpsData	获取数据	Output
LONG	*lplRetCode	通信函数的返回值	Output

(5) 说明

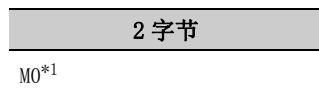
将通过 szDevice(varDevice) 指定的 1 点软元件数据存储到 iData(lpsData 或 lpvarData) 中。

(6) 软元件指定方法

软元件的指定方法如下所示。

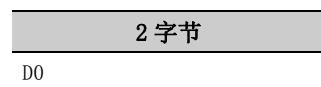
＜指定位软元件时＞

(例) M0



＜指定字软元件时＞

(例) D0



＜指定双字软元件时＞

(例) LTN0



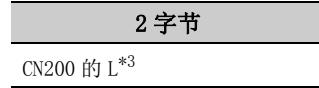
＜指定双字软元件时（位数指定）＞

(例) K8M0



＜指定 FXCPU 的 CN200 以后时＞

(例) CN200



＜指定网关软元件时＞

(例) 在 EGO 中被分配了 8 位软元件 (SHARP 生产的可编程控制器的 E0000、E0001) 的情况下



*1: 仅将“M0”1点作为读取软元件对象，软元件值将被存储“0”或“1”。

*2: 按照软元件的编号顺序从低位开始存储。

高位 2 字节 M16 ~ M31 的数据不被读取。

*3: 对于 FXCPU 的 CN200 以后，在 GetDevice2 中以 1 点读取指定软元件的 L(低位 2 字节)。

指定软元件的 H(高位 2 字节) 不被读取。

(7) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。（ 496 页第 7 章 出错代码）

要 点

- 指定了双字软元件的情况下，在 GetDevice2 中仅存储低位 1 字（2 字节）容量的数据。
(不发生出错。)
- 进行双字软元件的读取的情况下，应使用 ReadDeviceRandom 或 GetDevice。

5.2.24 Connect(电话线路的连接)

(1) 可以使用的控件

在 ActProgType 及 ActUtlType 控件中可以使用。^{*1}

*1：仅在调制解调器通信时可以使用。

(2) 功能

进行电话线路的连接。

(3) 格式 (调度 I/F)

(a) Visual C++® .NET (MFC)、VBA

```
iRet = object.Connect()  
Long          iRet          返回值          Output
```

(b) Visual Basic® .NET

```
iRet = object.Connect()  
Integer        iRet          返回值          Output
```

(c) Visual C++® .NET

```
iRet = object.Connect()  
int           iRet          返回值          Output
```

(d) Visual C#® .NET

```
iRet = object.Connect()  
int           iRet          返回值          Output
```

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.Connect( *lplRetCode )  
HRESULT      hResult         COM 的返回值          Output  
LONG         *lplRetCode     通信函数的返回值          Output
```

(5) 说明

- 根据调制解调器通信用控件的属性的设置值，进行电话线路的连接。
- 经由串行通信模块的情况下，根据 ActConnectWay 属性中设置的连接方式进行电话线路的连接。
通过 ActConnectWay 属性设置了自动（指定回调编号时）、回调连接（指定编号时）及回调请求（指定编号时）的情况下，如果未在 ActCallbackNumber 属性中设置编号将发生出错。

(6) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。（ 496 页第 7 章 出错代码）

(7) 执行 Connect 时的注意事项

- 电话线路的连接必须在 Open 之前执行。
 - 断开电话线路的情况下，应执行 Disconnect。
- 在 Connect 中即使多次重复进行 Open 及 Close，电话线路也将保持为连接状态不变。
- 由于某种原因导致电话线路被断开的情况下，电话线路断开之前的状态为 Open 中时，必须执行 Close 之后再进行电话线路的重新连接。

(8) 使用了多个电话线路连接对象情况下的注意事项

- 控件类型、端口编号、电话号码不相同的情况下
控件类型、端口编号、电话号码不相同的情况下，如果执行 Connect 的控件与最先执行了 Connect 的控件的端口编号、电话号码不相同，将发生出错（出错代码：0xF1000016）。
- 控件的端口编号、电话号码相同的情况下
控件类型、端口编号、电话号码相同的情况下，根据回调功能的连接方式其结束状态有所不同。

回调功能的连接方式与结束状态的关系如下表所示。

最先执行了 Connect 的控件的连接方式	第 2 次以后执行了 Connect 时的连接方式			
	自动 自动（固定为回调时） 自动（指定回调编号时）	回调连接 (固定时) 回调连接 (指定编号时)	回调请求 (固定时) 回调请求 (指定编号时)	回调接收等待
自动 自动（固定为回调时） 自动（指定回调编号时）	○	○	×	×
回调连接（固定时） 回调连接（指定编号时）	○	○	×	×
回调请求（固定时） 回调请求（指定编号时）	○	○	○	○
回调接收等待	×	×	×	○

○：正常结束 ×：非正常结束（发生出错）

5.2.25 Disconnect(电话线路的断开)

(1) 可以使用的控件

在 ActProgType 及 ActUtlType 控件中可以使用。^{*1}

*1：只有在调制解调器通信中才可以使用。

(2) 功能

进行电话线路的断开。

(3) 格式 (调度 I/F)

(a) Visual C++® .NET (MFC)、VBA

lRet = object.Disconnect()			
Long	lRet	返回值	Output

(b) Visual Basic® .NET

IRet = object.Disconnect()			
Integer	IRet	返回值	Output

(c) Visual C++® .NET

iRet = object.Disconnect()			
int	iRet	返回值	Output

(d) Visual C#® .NET

iRet = object.Disconnect()			
int	iRet	返回值	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

hResult = object.Disconnect(*lp1RetCode)			
HRESULT	hResult	COM 的返回值	Output
LONG	*lp1RetCode	通信函数的返回值	Output

(5) 说明

通过 Connect 函数断开连接的电话线路。

(6) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)

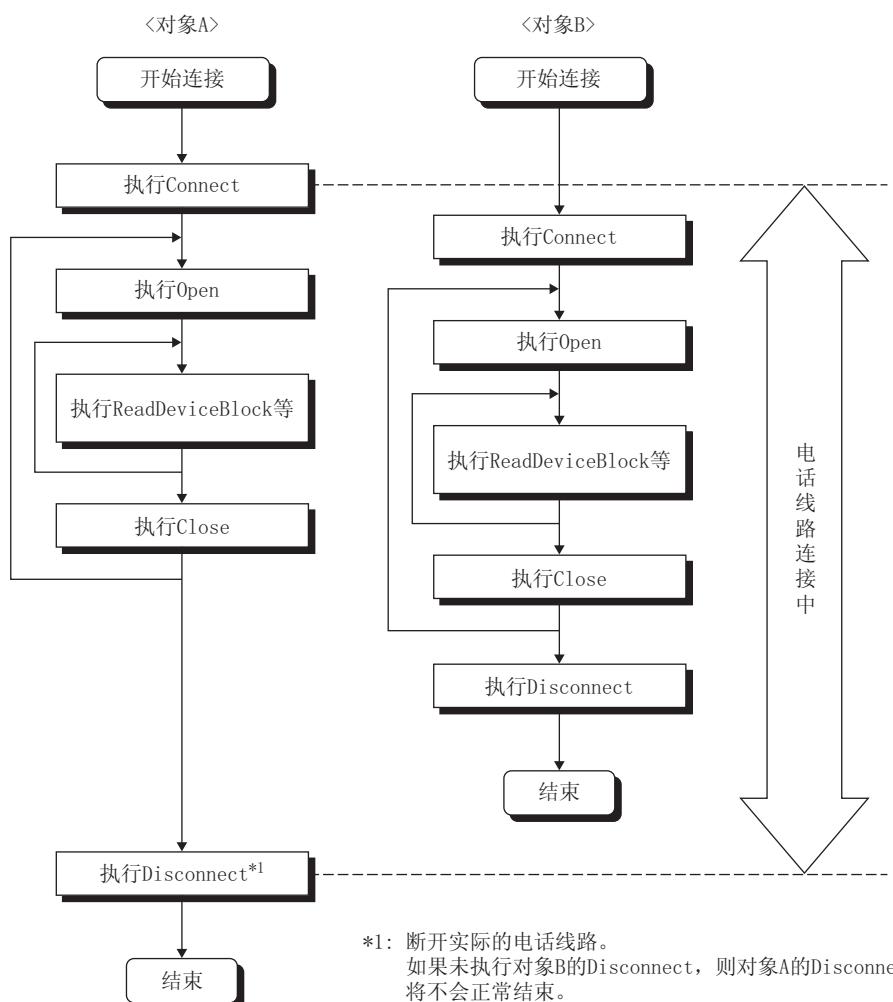
(7) 执行 Disconnect 时的注意事项

- 执行 Disconnect 的情况下

Open 中的情况下，应执行 Close 之后再执行 Disconnect。
 - 由于某种原因导致电话线路被断开的情况下，电话线路断开之前的状态为 Open 中时，必须执行 Close 之后再进行电话线路的重新连接。
 - 使用了多个电话线路连接对象的情况下，最先执行了 Connect 的对象应在其它对象执行 Disconnect 后，再执行 Disconnect。
- 此外，使用了多个电话线路连接对象的情况下，如果最先执行了 Connect 的对象未执行 Disconnect，则电话线路不被断开。

同时使用了多个对象情况下的示例如下所示。

〈例〉同时使用2个控件的情况下
(也包括通过对象A、对象B不同的应用程序执行动作的情况下)



5.2.26 GetErrorMessage(出错信息的获取)

(1) 可以使用的控件

在 Act (ML) SupportMsg 控件中可以使用。^{*1}

*1：变频器通信时不能使用。

(2) 功能

获取对应于出错代码的出错内容及处理方法。

(3) 格式 (调度 I/F)

(a) VBA

```
lRet = object.GetErrorMessage(lErrorCode, szErrorMessage)
```

Long	lRet	返回值	Output
String	lErrorCode	出错代码	Input
String	szErrorMessage	出错信息	Output

(b) Visual C++® .NET (MFC)

```
lRet = object.GetErrorMessage(lErrorCode, *lpszErrorMessage)
```

Long	lRet	返回值	Output
Long	lErrorCode	出错代码	Input
BSTR	*lpszErrorMessage	出错信息	Output

(c) VBScript

```
varRet = object.GetErrorMessage(varErrorCode, lpvarErrorMessage)
```

VARIANT	varRet	返回值 (内容为 LONG 型)	Output
VARIANT	varErrorCode	出错代码 (内容为 LONG 型)	Input
VARIANT	lpvarErrorMessage	出错信息 (内容为字符串型)	Output

(d) Visual Basic® .NET

```
iRet = object.GetErrorMessage(IErrorCode, szErrorMessage)
```

Integer	iRet	返回值	Output
Integer	IErrorCode	出错代码	Input
String	szErrorMessage	出错信息	Output

(e) Visual C++® .NET

```
iRet = object.GetErrorMessage(iErrorCode, **lpsErrorMessage)
```

int	iRet	返回值	Output
int	iErrorCode	出错代码	Input
String	**lpsErrorMessage	出错信息	Output

(f) Visual C#® .NET

```
iRet = object.GetErrorMessage(iErrorCode, out szErrorMessage)
```

int	iRet	返回值	Output
int	iErrorCode	出错代码	Input
String	szErrorMessage	出错信息	Output

(4) 格式 (自定义 I/F)

(a) Visual C++® .NET (MFC)

```
hResult = object.GetErrorMessage( lErrorCode、*lpszErrorMessage、*lpIRetCode )
```

HRESULT	hResult	COM 的返回值	Output
LONG	lErrorCode	出错代码	Input
BSTR	*lpszErrorMessage	出错信息	Output
LONG	*lpIRetCode	通信函数的返回值	Output

(5) 说明

- 对通过 lErrorCode (varErrorCode) 指定的出错代码的出错内容及处理方法进行读取。
- 读取的出错内容及处理方法将被存储到 szErrorMessage (lpszErrorMessage 或 lpvarErrorMessage) 中。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

5.3 函数的详细内容 (.NET 控件用)

5.3.1 Open(通信线路的打开)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

打开通信线路。

(3) 格式

(a) Visual Basic® .NET

IRet = object.Open()			
Integer	iRet	返回值	Output

(b) Visual C++® .NET

iRet = object.Open()			
int	iRet	返回值	Output

(c) Visual C#® .NET

iRet = object.Open()			
int	iRet	返回值	Output

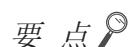
(4) 说明

根据 Open 函数用属性的设置值进行线路的连接。

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)



- 使用调制解调器通信时如果未执行 Connect 则不能执行 Open。
- Open 完毕后即使更改 Open 函数用属性，也不会执行通信对象的更改。进行通信设置更改的情况下，应关闭 1 次通信线路后，进行通信对象的设置并重新打开通信线路。
- 进行 Open 时，即使在 ActCpuType 属性中输入了与进行通信的 CPU 不同 CPU 类型的情况下，Open 也有可能会正常结束。在这种情况下，将发生连接范围、可以使用的方法及软元件范围变窄等现象。
进行 Open 时，应在 ActCpuType 属性中设置正确的 CPU 类型。

5.3.2 Close(通信线路的关闭)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

关闭通信线路。

(3) 格式

(a) Visual Basic® .NET

IRet = object.Close()			
Integer	IRet	返回值	Output

(b) Visual C++® .NET

iRet = object.Close()			
int	iRet	返回值	Output

(c) Visual C#® .NET

iRet = object.Close()			
int	iRet	返回值	Output

(4) 说明

通过 Open 函数关闭连接的线路。

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。(☞ 496 页第 7 章 出错代码)

5.3.3 ReadDeviceBlock(软元件的批量读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行软元件的批量读取。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.ReadDeviceBlock(szLabel, iSize, iData(0))
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Integer	iSize	读取点数	Input
Integer	iData(n)	读取的软元件值	Output

(b) Visual C++® .NET

```
iRet = object.ReadDeviceBlock(*szLabel, iSize, *ipiData)
```

Int	iRet	返回值	Output
String*	szLabel	标签名	Input
int	iSize	读取点数	Input
int*	ipiData	读取的软元件值	Output

(c) Visual C#® .NET

```
iRet = object.ReadDeviceBlock(ref szLabel, iSize, ref iData)
```

Int	iRet	返回值	Output
String	szLabel	标签名	Input
int	iSize	读取点数	Input
int[n]	iData	读取的软元件值	Output

(4) 说明

- 从标签名 szLabel 中指定的软元件开始，对读取点数 iSize 容量的软元件进行批量读取。
- 读取的软元件值将被存储到 iData(ipiData) 中。
- 在读取的软元件值 iData(ipiData) 中，应预留出读取点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及读取软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型		标签名的格式
			标签名
数组	成员	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	标签名 [要素数] 标签名 [n1] [n2] [n3]

*1： 最多可指定半角 32 个字符 +NULL。

- 在读取点数中应根据标签的数据类型设置以下值。

标签(数组)的数据类型	对应的字数	指定的读取点数
位	1	标签的数组要素数 ÷ 16(进位)
字	1	标签的数组要素数
双字、单精度实数	2	标签的数组要素数 × 2
双精度实数	4	标签的数组要素数 × 4
字符串	17	标签的数组要素数 × 17
字符串[Unicode]	33	标签的数组元素数 × 33
时间	2	标签的数组要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的数组要素数

- 读取的软元件值按以下方式被存储。

< 指定位软元件时 >

(例) 从 M0 开始读取 3 点 (3 字 = 48 位)

- 标签设置(数据类型：位；数组要素数：48)

数据类型	软元件
位(0..47)	M0

- 读取点数：3

- 读取的软元件值

高位2字节	低位2字节
不使用。 (存储0。)	M0 ~ M15*3
	M16 ~ M31*3
	M32 ~ M47*3

*3：按照软元件的编号顺序从低位开始存储。

< 指定 FXCPU 的 CN200 以后时 >

(例) 从 CN200 开始读取 6 点 *4

- 标签设置(数据类型：双字；数组要素数：3)

数据类型	软元件
双字(0..2)	CN200

- 读取点数：6

- 读取的软元件值

高位2字节	低位2字节
不使用。 (存储0。)	CN200 的 L
	CN200 的 H
	CN201 的 L
	CN201 的 H
	CN202 的 L
	CN202 的 H

*4：对于 FXCPU 的 CN200 以后，按照每个软元件读取 2 点的方式分别读取高位(H) 及低位(L) 的数据。
如果读取 1 点数据将发生出错。

< 指定字软元件时 >

(例) 从 D0 开始读取 3 点

- 标签设置(数据类型：字；数组要素数：3)

数据类型	软元件
字(0..2)	D0

- 读取点数：3

- 读取的软元件值

高位2字节	低位2字节
不使用。 (存储0。)	D0
	D1
	D2

< 指定 FD 软元件时(4字软元件) >

(例) 从 FD0 开始读取 8 点

- 标签设置(数据类型：双字；数组要素数：2)

数据类型	软元件
双字(0..1)	FD0

- 读取点数：8

- 读取的软元件值

高位2字节	低位2字节
不使用。 (存储0。)	FDO 的 LL
	FDO 的 LH
	FDO 的 HL
	FDO 的 HH
	FD1 的 LL
	FD1 的 LH
	FD1 的 HL
	FD1 的 HH

< 标签为双字数组指定字软元件时 >

(例) 从 D100 开始读取 6 点

- 标签设置 (数据类型：双字；数组要素数：3)

数据类型	软元件
双字 (0..2)	D100

- 读取点数：6

- 读取的软元件值

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D100
	D101
	D102
	D103
	D104
	D105

< 指定长定时器、长计数器、长累计定时器型时 >

(例) 从 LT0 开始读取 3 点。

- 标签设置 (数据类型：长定时器，数组元素数：3)

数据类型	软元件
长定时器 (0..2)	LT0

- 读取点数：3

- 读取的软元件值

高位 2 字节	低位 2 字节
LTO 的 H	LTO 的 L
LT1 的 H	LT1 的 L
LT2 的 H	LT2 的 L

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 读取点数中可指定的最大读取点数为满足下述条件的范围。
读取开始软元件编号 + 读取点数 ≤ 最终软元件编号
- 指定位型数组的标签的情况下，软元件编号只能指定为 0 或 16 的倍数。
- 应将标签名中指定的数据类型所对应的字数指定为读取点数。读取的软元件值中，应预留出相当于读取点数中指定的点数容量的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法读取。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法读取。
- 位软元件的位数指定及变址修饰不能使用。
- 使用系统标签 Ver. 2 的情况下，通过 MX Component 的标签管理实用程序定义的数据类型必须与通过 MELSOFT Navigator 管理的数据类型相一致。
数据类型不一致的情况下，有时会出现实际读取的数据长度不正确、指定了多个标签的情况下读取的软元件值的数组与标签名不对应的情况。

5.3.4 WriteDeviceBlock(软元件的批量写入)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行软元件的批量写入。

(3) 格式

(a) Visual Basic® .NET

```
Ret = object.WriteDeviceBlock(szLabel、iSize、iData(0))
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Integer	iSize	写入点数	Input
Integer	iData(n)	写入的软元件值	Input

(b) Visual C++® .NET

```
iRet = object.WriteDeviceBlock(*szLabel、iSize、*ipiData)
```

int	iRet	返回值	Output
String*	szLabel	标签名	Input
int	iSize	写入点数	Input
int*	ipiData	写入的软元件值	Input

(c) Visual C#® .NET

```
iRet = object.WriteDeviceBlock(ref szLabel、iSize、iData)
```

int	iRet	返回值	Output
String	szLabel	标签名	Input
int	iSize	写入点数	Input
int[n]	iData	写入的软元件值	Input

(4) 说明

- 从标签名 szLabel 中指定的软元件开始，对写入点数 iSize 容量的软元件进行批量写入。
- 写入的软元件值被存储到 iData(ipiData) 中。
- 在写入的软元件值 iData(ipiData) 中，应预留出写入点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及写入的软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型		标签名的格式
数组	标签名		
数组 成员	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、 字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数器、 长计数器、累计定时器、长累计定时器	标签名 [要素数] 标签名 [n1] [n2] [n3]	

*1： 最多可指定半角 32 个字符 +NULL。

- 对写入点数应根据标签的数据类型设置以下值。

标签(数组)的数据类型	对应的字数	指定的读取点数
位	1	标签的数组要素数 ÷ 16(进位)
字	1	标签的数组要素数
双字、单精度实数	2	标签的数组要素数 × 2
双精度实数	4	标签的数组要素数 × 4
字符串	17	标签的数组要素数 × 17
字符串[Unicode]	33	标签的数组元素数 × 33
时间	2	标签的数组要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的数组要素数

- 对写入的软元件值应按以下方式进行设置。

< 指定位软元件时 >

(例) 从 M0 开始写入 3 点 (3 字 = 48 位)

- 标签设置 (数据类型 : 位; 数组要素数 : 48)

数据类型	软元件
位 (0..47)	M0

- 写入点数 : 3

- 写入的软元件值

高位 2 字节	低位 2 字节
	M0 ~ M15*3
不使用。	M16 ~ M31*3
	M32 ~ M47*3

*3: 按软元件的编号顺序从低位开始存储。

< 指定 FXCPU 的 CN200 以后时 >

(例) 从 CN200 开始写入 6 点 *4

- 标签设置 (数据类型 : 双字; 数组要素数 : 3)

数据类型	软元件
双字 (0..2)	CN200

- 写入点数 : 6

- 写入的软元件值

高位 2 字节	低位 2 字节
	CN200 的 L
不使用。	CN200 的 H
	CN201 的 L
	CN201 的 H
	CN202 的 L
	CN202 的 H

*4: 对于 FXCPU 的 CN200 以后, 按照每个软元件写入 2

点的方式分别写入高位 (H) 及低位 (L) 的数据。

如果写入 1 点将发生出错。

< 指定字软元件时 >

(例) 从 D0 开始写入 3 点

- 标签设置 (数据类型 : 字; 数组要素数 : 3)

数据类型	软元件
字 (0..2)	D0

- 写入点数 : 3

- 写入的软元件值

高位 2 字节	低位 2 字节
	D0
不使用。	D1
	D2

< 指定 FD 软元件时 (4 字软元件) >

(例) 从 FDO 开始写入 8 点

- 标签设置 (数据类型 : 双字; 数组要素数 : 2)

数据类型	软元件
双字 (0..1)	FDO

- 写入点数 : 8

- 写入的软元件值

高位 2 字节	低位 2 字节
	FDO 的 LL
不使用。	FDO 的 LH
	FDO 的 HL
	FDO 的 HH
	FD1 的 LL
	FD1 的 LH
	FD1 的 HL
	FD1 的 HH

<标签为双字数组指定字软元件时>

(例) 从 D100 开始写入 6 点

- 标签设置 (数据类型: 双字; 数组要素数: 3)

数据类型	软元件
双字 (0..2)	D100

- 写入点数: 6

- 写入的软元件值

高位 2 字节	低位 2 字节
不使用。	D100
	D101
	D102
	D103
	D104
	D105

<指定长定时器、长计数器、长累计定时器型时>

(例) 从 LT0 开始写入 3 点。

- 标签设置 (数据类型: 长定时器, 数组元素数: 3)

数据类型	软元件
长定时器 (0..2)	LT0

- 写入点数: 3

- 写入的软元件值

高位 2 字节	低位 2 字节
LT0 的 H	LT0 的 L
LT1 的 H	LT1 的 L
LT2 的 H	LT2 的 L

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 写入点数中可指定的最大写入点数为满足下述条件的范围。
写入开始软元件编号 + 写入点数 ≤ 最终软元件编号
- 指定位型数组的标签的情况下, 对软元件编号只能指定 0 或 16 的倍数。
- 应将标签名中指定的数据类型所对应的字数指定为读取点数。在写入的软元件值中, 应预留出相当于写入点数中指定的点数容量的存储器区域。没有存储器区域的情况下, 有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法写入。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法写入。
- 位软元件的位数指定及变址修饰不能使用。
- 向长定时器软元件 (LT) 及累积长定时器软元件 (LST) 写入时, 不能使用 WriteDeviceBlock。
请使用 WriteDeviceRandom 或 SetDevice。
- 使用系统标签 Ver. 2 的情况下, 通过 MX Component 的标签管理实用程序定义的数据类型必须与通过 MELSOFT Navigator 管理的数据类型相一致。
数据类型不一致的情况下, 有时会出现实际读取的数据长度不正确、指定了多个标签的情况下读取的软元件值的数组与标签名不对应的情况。

5.3.5 ReadDeviceRandom(软元件的随机读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行软元件的随机读取。

(3) 格式

(a) Visual Basic® .NET

```
IRet = object.ReadDeviceRandom(szLabel、iSize、iData(0))

    Integer          IRet          返回值          Output
    String           szLabel       标签名          Input
    Integer          iSize         读取点数        Input
    Integer          iData(n)     读取的软元件值  Output

IRet = object.ReadDeviceRandom(szLabelList、iSize、iData(0))

    Integer          IRet          返回值          Output
    String           szLabelList(n)  标签列表      Input
    Integer          iSize         读取点数        Input
    Integer          iData(n)     读取的软元件值  Output
```

(b) Visual C++® .NET

```
iRet = object.ReadDeviceRandom(*szLabel、iSize、*ipiData)

    Int            iRet          返回值          Output
    String*        szLabel       标签名          Input
    int            iSize         读取点数        Input
    int*           ipiData      读取的软元件值  Output

iRet = object.ReadDeviceRandom(**szLabelList、iSize、*arriData)

    Int            iRet          返回值          Output
    String**       szLabelList   标签列表      Input
    int            iSize         读取点数        Input
    int*           ipiData      读取的软元件值  Output
```

(c) Visual C#® .NET

```
iRet = object.ReadDeviceRandom(ref szLabel、iSize、ref iData)

    Int            iRet          返回值          Output
    String          szLabel       标签名          Input
    int            iSize         读取点数        Input
    int[n]          iData        读取的软元件值  Output

iRet = object.ReadDeviceRandom(ref szLabelList、iSize、ref iData)

    Int            iRet          返回值          Output
    System.String[] szLabelList  标签列表      Input
    int            iSize         读取点数        Input
    int[n]          iData        读取的软元件值  Output
```

(4) 说明

- 将标签名 szLabel (szLabelList) 中指定的软元件群只读取相当于读取点数 iSize 容量的软元件值。
- 读取的软元件值将被存储到 iData (ipiData) 中。
- 对于读取的软元件值 iData (ipiData)，应预留出读取点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及读取软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型		标签名的格式
基本型	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器		标签名
数组	成员	(可以进行与基本型相同的指定)	
结构体	成员	(可以进行与基本型相同的指定)	
结构体数组	成员	(可以进行与基本型相同的指定)	

*1： 最多可指定半角 32 个字符 +NULL。

- 在读取点数中应根据标签的数据类型，设置各要素相关的以下值的和。

标签的数据类型	对应的字数	指定的读取点数
位、字	1	标签的要素数
双字、单精度实数	2	标签的要素数 × 2
双精度实数	4	标签的要素数 × 4
字符串	17	标签的要素数 × 17
字符串 [Unicode]	33	标签的数组元素数 × 33
时间	2	标签的要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的要素数

- 读取的软元件值按以下方式被存储。

< 指定位软元件及字软元件时 >

(例) 将 M0 及 D0 逐点进行读取

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	位
[1]	LABEL2	字

- 读取点数：2

- 读取的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。 (存储 0。)	M0 ^{*1}	LABEL1
	D0	LABEL2

*1: 仅将 “M0” 1 点作为读取软元件对象，软元件值将被存储 “0” 或 “1”。

< 指定 FXCPU 的 CN200 以后时 >

(例) 包括 CN200 读取 3 点^{*2}

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	字
[1]	LABEL2	字
[2]	LABEL3	字

- 读取点数：3

- 读取的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。 (存储 0。)	D0	LABEL1
CN200 的 H	CN200 的 L	LABEL2
不使用。 (存储 0。)	D1	LABEL3

*2: 对于 FXCPU 的 CN200 以后，以读取点数 1 点读取 4 字节。

< 指定 FD 软元件时 (4 字软元件) >

(例) 包括 FD0 读取 3 点

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	字
[1]	LABEL2	字
[2]	LABEL3	字

- 读取点数：3

- 读取的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。 (存储 0。)	D0	LABEL1
	FD0 的 LL ^{*3}	LABEL2
	D1	LABEL3

*3: 仅读取低位 2 字节。指定软元件的 HH、HL、LH(高位 6 字节) 不被读取。

〈标签中指定相当于2字以上的数据类型时〉

(例) 指定双字、单精度实数、双精度实数、字符串^{*5}及时间型标签进行读取

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	双字	D0
[1] LABEL2	单精度实数	D100
[2] LABEL3	双精度实数	D200
[3] LABEL4	字符串	D300
[4] LABEL5	时间	D400

- 读取点数：27

- 读取的软元件值

高位2字节	低位2字节	对应标签
不使用。 (存储0。)	D0	LABEL1
	D1	
	D100	LABEL2
	D101	
	D200	LABEL3
	D201	
	D202	
	D203	
	D300	LABEL4
	: D316 ^{*4}	
D400	D401	LABEL5

*4: 读取字符串17点(半角32字符+NULL)。由于不能进行字符串型的字符转换,因此需要在用户程序内进行转换。

〈指定长定时器、长计数器、长累计定时器型时〉

(例) 从LT0开始读取3点。

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL	长定时器	LT0
[1] LABEL	长定时器	LT1
[2] LABEL	长定时器	LT2

- 读取点数：3

- 读取的软元件值

高位2字节	低位2字节	对应标签
LT0的H	LT0的L	LABEL[1]
LT1的H	LT1的L	LABEL[2]
LT2的H	LT2的L	LABEL[3]

〈指定数组型标签时〉

(例) 指定数组型标签进行读取

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	位(0..1)	M0
[1] LABEL2	双字(0..1)	CN200
[2] LABEL3	双字(0..1)	D0

- 读取点数：8

- 读取的软元件值

高位2字节	低位2字节	对应标签
不使用。 (存储0。)	M0	LABEL1[0]
	M1	LABEL1[1]
CN200的H	CN200的L	LABEL2[0]
	CN201的L	LABEL2[1]
不使用。 (存储0。)	D100	LABEL3[0]
	D101	
	D102	LABEL3[1]
	D103	

5.3 函数的详细内容 (.NET控件用)
5.3.5 ReadDeviceRandom(软元件的随机读取)

< 指定结构体型标签时 >

(例) 指定结构体型标签进行读取

- 结构体设置

结构体名	标签名	数据类型
STRUCT	L1	位
	L2	双字

- 标签设置

	szLabelList	数据类型	标签名	软元件
[0]	LABEL1	STRUCT	L1	D0.0
			L2	D0
[1]	LABEL2	STRUCT	L1	M10
			L2	CN200

- 读取点数 : 6

- 读取的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。 (存储 0。)	D0.0	LABEL1.L1
	D0	LABEL1.L2
	D1	
	M0	LABEL2.L1
CN200 的 H	CN200 的 L	LABEL2.L2
CN201 的 H	CN201 的 L ^{*5}	

*5: 双字型标签中设置了 CN200 以后的软元件的情况下，相当于 2 软元件的容量将被读取。

< 指定组合了结构体与数组的标签时 >

(例) 指定并读取结构体型的数组及结构体数组型标签

- 结构体设置

结构体名	标签名	数据类型
STRUCT1	L1	位
	L2	字
STRUCT2	L1	位 (0..2)
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0]	LABEL1	STRUCT1 (0..1)	L1 X0
			L2 D0
[1]	LABEL2	STRUCT2	L1 M0
			L2 D100

- 读取点数 : 9

- 读取的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。 (存储 0。)	X0	LABEL1[0].L1
	D0	LABEL1[0].L2
	X1	LABEL1[1].L1
	D1	LABEL1[1].L2
	M0	LABEL2.L1[0]
	M1	LABEL2.L1[1]
	M2	LABEL2.L1[2]
	D100	LABEL2.L2
	D101	

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 读取点数中可指定的最大读取点数为 0x7FFFFFFF 点。
- 应将标签名中指定的数据类型所对应的字节数指定为读取点数。在读取的软元件值中，应预留出相当于读取点数中指定的点数容量的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法读取。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法读取。
- 位软元件的位数指定及变址修饰不能使用。
- 使用系统标签 Ver. 2 的情况下，通过 MX Component 的标签管理实用程序定义的数据类型必须与通过 MELSOFT Navigator 管理的数据类型相一致。
数据类型不一致的情况下，有时会出现实际读取的数据长度不正确、指定了多个标签的情况下读取的软元件值的数组与标签名不对应的情况。

5.3.6 WriteDeviceRandom(软元件的随机写入)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行软元件的随机写入。

(3) 格式

(a) Visual Basic® .NET

```
Ret = object.WriteDeviceRandom(szLabel、iSize、iData(0))

    Integer          IRet      返回值      Output
    String           szLabel   标签名      Input
    Integer          iSize     写入点数    Input
    Integer          iData(n)  写入的软元件值  Input

Ret = object.WriteDeviceRandom(szLabelList、iSize、iData(0))

    Integer          IRet      返回值      Output
    String           szLabelList(n)  标签列表  Input
    Integer          iSize     写入点数    Input
    Integer          iData(n)  写入的软元件值  Input
```

(b) Visual C++® .NET

```
iRet = object.WriteDeviceRandom(*szLabel、iSize、*ipiData)

    int            iRet      返回值      Output
    String*        szLabel   标签名      Input
    int            iSize     写入点数    Input
    int*           ipiData   写入的软元件值  Input

iRet = object.WriteDeviceRandom(**szLabelList、iSize、*ipiData)

    int            iRet      返回值      Output
    String**       szLabelList  标签列表  Input
    int            iSize     写入点数    Input
    int*           ipiData   写入的软元件值  Input
```

(c) Visual C#® .NET

```
iRet = object.WriteDeviceRandom(ref szLabel、iSize、iData)

    int            iRet      返回值      Output
    String         szLabel   标签名      Input
    int            iSize     写入点数    Input
    int[n]         iData     写入的软元件值  Input

iRet = object. WriteDeviceRandom (ref szLabelList、iSize、iData)

    int            iRet      返回值      Output
    System.String[] szLabelList  标签列表  Input
    int            iSize     写入点数    Input
    int[n]         iData     写入的软元件值  Input
```

(4) 说明

- 将标签名 szLabel (szLabelList) 中指定的软元件群只写入相当于写入点数 iSize 容量的软元件值。
- 将写入的软元件值存储到 iData (ipiData) 中。
- 对于写入的软元件值 iData (ipiData)，应预留出写入点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及写入的软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型		标签名的格式
基本型	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器		标签名
数组	成员	(可以进行与基本型相同的指定)	
结构体	成员	(可以进行与基本型相同的指定)	
结构体数组	成员	(可以进行与基本型相同的指定)	

*1：最多可指定半角 32 个字符 +NULL。

- 在写入点数中应根据标签的数据类型，设置各要素相关的以下值的和。

标签的数据类型	对应的字数	指定的写入点数
位，字	1	标签的要素数
双字，单精度实数	2	标签的要素数 × 2
双精度实数	4	标签的要素数 × 4
字符串	17	标签的要素数 × 17
字符串 [Unicode]	33	标签的数组元素数 × 33
时间	2	标签的要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的要素数

- 对写入的软元件值应按以下方式进行设置。

< 指定位软元件及字软元件时 >

(例) 将 M0 及 D0 逐点进行写入

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	位	M0
[1] LABEL2	字	D0

- 写入点数：2

- 写入的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。	M0	LABEL1
	D0	LABEL2

< 指定 FXCPU 的 CN200 以后时 >

(例) 包括 CN200 写入 3 点 *1

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	字	D0
[1] LABEL2	字	CN200
[2] LABEL3	字	D1

- 写入点数：3

- 写入的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。	D0	LABEL1
CN200 的 H	CN200 的 L	LABEL2
不使用。	D1	LABEL3

*1: 对于 FXCPU 的 CN200 以后，以写入点数 1 点写入 4 字节。

< 指定 FD 软元件时 (4 字软元件) >

(例) 包括 FD0 写入 3 点

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	字	D0
[1] LABEL2	字	FD0
[2] LABEL3	字	D1

- 写入点数：3

- 写入的软元件值

高位 2 字节	低位 2 字节	对应标签
	D0	LABEL1
不使用。	FD0 的 LL*2	LABEL2
	D1	LABEL3

*2: 只能设置低位 2 字节的值。指定软元件的 HH、HL、LH(高位 6 字节) 中将被写入 “0”。

〈标签中指定相当于2字以上的数据类型时〉

(例) 指定及写入双字, 单精度实数、双精度实数、字符串^{*3}及时间型标签

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	双字	D0
[1] LABEL2	单精度实数	D100
[2] LABEL3	双精度实数	D200
[3] LABEL4	字符串	D300
[4] LABEL5	时间	D400

- 写入点数: 27

- 写入的软元件值

高位2字节	低位2字节	对应标签
不使用。	D0	LABEL1
	D1	
	D100	LABEL2
	D101	
	D200	LABEL3
	D201	
	D202	
	D203	
	D300	LABEL4
	:	
	D316 ^{*3}	
	D400	LABEL5
	D401	

*3: 写入字符串17点(半角32字符+NULL)。由于不能进行字符串型的字符转换, 因此需要在用户程序内进行转换。

〈指定长定时器、长计数器、长累计定时器型时〉

(例) 从LT0开始写入3点。

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL	长定时器	LT0
[1] LABEL	长定时器	LT1
[2] LABEL	长定时器	LT2

- 写入点数: 3

- 写入的软元件值

高位2字节	低位2字节	对应标签
LT0的H	LT0的L	LABEL[1]
LT1的H	LT1的L	LABEL[2]
LT2的H	LT2的L	LABEL[3]

〈指定数组型标签时〉

(例) 指定数组型标签进行写入

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	位(0..1)	M0
[1] LABEL2	字(0..1)	CN200
[2] LABEL3	双字(0..1)	D0

- 写入点数: 8

- 写入的软元件值

高位2字节	低位2字节	对应标签
不使用。 (存储0。)	M0	LABEL1[0]
	M1	LABEL1[1]
CN200的H	CN200的L	LABEL2[0]
CN201的H	CN201的L	LABEL2[1]
不使用。	D100	LABEL3[0]
	D101	
	D102	LABEL3[1]
	D103	

< 指定结构体型标签时 >

(例) 指定结构体型标签进行写入

- 结构体设置

结构体名	标签名	数据类型
STRUCT	L1	位
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0] LABEL1	STRUCT	L1	D0.0
		L2	D0
[1] LABEL2	STRUCT	L1	M10
		L2	CN200

• 写入点数 : 5

• 写入的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。	D0.0	LABEL1.L1
	D0	LABEL1.L2
	D1	
	M0	LABEL2.L1
CN200 的 H	CN200 的 L	LABEL2.L2

< 指定组合了结构体及数组的标签时 >

(例) 指定结构体型的数组及结构体数组型标签进行写入

- 结构体设置

结构体名	标签名	数据类型
STRUCT1	L1	位
	L2	字
STRUCT2	L1	位 (0..2)
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0] LABEL1	STRUCT1 (0..1)	L1	X0
		L2	D0
[1] LABEL2	STRUCT2	L1	M0
		L2	D100

• 写入点数 : 9

• 写入的软元件值

高位 2 字节	低位 2 字节	对应标签
不使用。	X0	LABEL1[0].L1
	D0	LABEL1[0].L2
	X1	LABEL1[1].L1
	D1	LABEL1[1].L2
	M0	LABEL2.L1[0]
	M1	LABEL2.L1[1]
	M2	LABEL2.L1[2]
	D100	LABEL2.L2
	D101	

(6) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。（ 496 页第 7 章 出错代码）

要 点

- 写入点数中可指定的最大写入点数为 0x7FFFFFFF 点。
- 应将标签名中指定的数据类型所对应的字数指定为写入点数。在写入的软元件值中，应预留出相当于写入点数中指定的点数容量的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法写入。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法写入。
- 位软元件的位数指定及变址修饰不能使用。
- 如果对 Q 运动 CPU 进行访问，将返回出错信息。
- 使用系统标签 Ver. 2 的情况下，通过 MX Component 的标签管理实用程序定义的数据类型必须与通过 MELSOFT Navigator 管理的数据类型相一致。
数据类型不一致的情况下，有时会出现实际读取的数据长度不正确、指定了多个标签的情况下读取的软元件值的数组与标签名不对应的情况。

5.3.7 SetDevice(软元件数据的设置)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行 1 点软元件的设置。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.SetDevice(szLabel、 iData)
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Integer	iData	写入的软元件值	Input

(b) Visual C++® .NET

```
iRet = object.SetDevice(*szLabel、 iData)
```

int	iRet	返回值	Output
String*	szLabel	标签名	Input
int	iData	写入的软元件值	Input

(c) Visual C++® .NET

```
iRet = object.SetDevice(ref szLabel、 iData)
```

int	iRet	返回值	Output
String	szLabel	标签名	Input
int	iData	写入的软元件值	Input

(4) 说明

- 对标签名 szLabel 中指定的 1 点软元件写入所写入的软元件值 iData 的值。
- 位软元件设置时，写入的软元件值 iData 的最低位的位将有效。

(5) 软元件指定方法

标签名及写入的软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
基本型	位、字、双字 ^{*1} 、单精度实数 ^{*1} 、双精度实数 ^{*1} 、字符串 ^{*1} 、字符串[Unicode]、时间 ^{*1} 、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	标签名

^{*1}: 仅写入起始软元件的低位 2 字节。

希望写入 2 字以上的软元件的情况下应使用 WriteDeviceBlock 函数。

但是指定了以下软元件的情况下，也可以以双字型写入高位 2 字节。

- FXCPU 的 CN200 以后的软元件

- 对写入的软元件值应按以下方式进行设置。

<指定位软元件时>

(例) 写入 M0

- 标签设置

数据类型	软元件
位	M0

- 写入的软元件值

高位 2 字节	低位 2 字节
不使用。	M0

<指定 FXCPU 的 CN200 以后时>

(例) 写入 CN200*1

- 标签设置

数据类型	软元件
双字	CN200

- 写入的软元件值

高位 2 字节	低位 2 字节
CN200 的 H	CN200 的 L

*1: 对于 FXCPU 的 CN200 以后，写入 4 字节。

<在标签中指定相当于 2 字以上的数据类型时>

(例) 指定字符串型标签进行写入

- 标签设置

数据类型	软元件
字符串	D0

- 写入的软元件值

高位 2 字节	低位 2 字节
不使用。	D0*3

*3: 起始软元件只写入低位 2 字节。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要点

- 标签名所对应的软元件不存在的情况下将发生出错而无法写入。
- 指定 FXCPU 的 CN200 以前时指定了双字软元件的情况下，在本函数中将对相当于低位 1 字（2 字节）容量的数据进行写入，在相当于高位 1 字（2 字节）容量的数据中写入“0”。
写入双字软元件的情况下，应使用 WriteDeviceRandom 或 WriteDeviceBlock2。
- 位软元件的位数指定及变址修饰不能使用。

<指定字软元件时>

(例) 写入 D0

- 标签设置

数据类型	软元件
字	D0

- 写入的软元件值

高位 2 字节	低位 2 字节
不使用。	D0

<指定 FD 软元件时(4 字软元件)>

(例) 指定 FD0 进行写入

- 标签设置

数据类型	软元件
字	FD0

- 写入的软元件值

高位 2 字节	低位 2 字节
不使用。	FD0 的 LL*2

*2: 只能设置低位 2 字节的值。指定软元件的 HH、HL、LH(高位 6 字节)将被写入“0”。

<指定长定时器、长计数器、长累计定时器型时>

(例) 写入 LT0

- 标签设置

数据类型	软元件
长定时器	LT0

- 写入的软元件值

高位 2 字节	低位 2 字节
LT0 的 H	LT0 的 L

5.3.8 GetDevice(软元件数据的获取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

获取 1 点软元件的数据。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.GetDevice(szLabel、 iData)
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Integer	iData	读取的软元件值	Output

(b) Visual C++® .NET

```
iRet = object.GetDevice(*szLabel、 *ipiData)
```

int	iRet	返回值	Output
String*	szLabel	标签名	Input
int*	ipiData	读取的软元件值	Output

(c) Visual C#® .NET

```
iRet = object.GetDevice(ref szLabel、 ref iData)
```

int	iRet	返回值	Output
String	szLabel	标签名	Input
int	iData	读取的软元件值	Output

(4) 说明

- 将标签名 szLabel 中指定的 1 点软元件的数据存储到读取的软元件值 iData(ipiData) 中。

(5) 软元件指定方法

标签名及读取软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
基本型	位、字、双字 ^{*1} 、单精度实数 ^{*1} 、双精度实数 ^{*1} 、字符串 ^{*1} 、字符串 [Unicode]、时间 ^{*1} 、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	标签名

*1：只能读取起始软元件的低位 2 字节。

希望读取 2 字节以上的软元件的情况下应使用 ReadDeviceBlock 函数。

但是指定了以下软元件的情况下，也可以以双字型读取高位 2 字节。

- FXCPU 的 CN200 以后的软元件

- 读取的软元件值按以下方式被存储。

<指定位软元件时>

(例) 读取 M0

- 标签设置

数据类型	软元件
位	M0

- 读取的软元件值

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	M0 ^{*1}

*1: 仅将“M0”1点作为读取软元件对象，软元件值中将被存储“0”或“1”。

<指定 FXCPU 的 CN200 以后时>

(例) 读取 CN200^{*2}

- 标签设置

数据类型	软元件
双字	CN200

- 读取的软元件值

高位 2 字节	低位 2 字节
CN200 的 H	CN200 的 L

*2: 对于 FXCPU 的 CN200 以后，读取 4 字节。

<在标签中指定相当于 2 字以上的数据类型时>

(例) 指定字符串型标签进行读取

- 标签设置

数据类型	软元件
字符串	D0

- 读取的软元件值

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0 ^{*4}

*4: 仅读取起始软元件低位 2 字节。

<指定字软元件时>

(例) 读取 D0

- 标签设置

数据类型	软元件
字	D0

- 读取的软元件值

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	D0

<指定 FD 软元件时(4 字软元件)>

(例) 指定 FD0 进行读取

- 标签设置

数据类型	软元件
字	FD0

- 读取的软元件值

高位 2 字节	低位 2 字节
不使用。 (存储 0。)	FD0 的 LL ^{*3}

*3: 只读取低位 2 字节。不读取指定软元件的 HH、HL、LH(高位 6 字节)。

<指定长定时器、长计数器、长累计定时器型时>

(例) 读取 LT0

- 标签设置

数据类型	软元件
长定时器	LT0

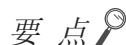
- 读取的软元件值

高位 2 字节	低位 2 字节
LT0 的 H	LT0 的 L

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。(参见 496 页第 7 章 出错代码)



- 标签名所对应的软元件不存在的情况下将发生出错而无法读取。

- 在读取的软元件值中，应预留 4 字节的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。

- 位软元件的位数指定及变址修饰不能使用。

5.3.9 ReadBuffer(缓冲存储器读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

读取特殊功能模块的缓冲存储器的值。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.ReadBuffer(iStartIO、iAddress、iReadSize、sData(0))
```

Integer	iRet	返回值	Output
Integer	iStartIO	读取值的模块的 I/O 编号	Input
Integer	iAddress	缓冲存储器的地址	Input
Integer	iReadSize	读取容量	Input
Short	sData(n)	从缓冲存储器中读取的值	Output

(b) Visual C++® .NET

```
iRet = object.ReadBuffer(iStartIO、iAddress、iReadSize、*ipsData)
```

int	iRet	返回值	Output
int	iStartIO	读取值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iReadSize	读取容量	Input
short*	ipsData	从缓冲存储器中读取的值	Output

(c) Visual C#® .NET

```
iRet = object.ReadBuffer(iStartIO、iAddress、iReadSize、ref sData)
```

int	iRet	返回值	Output
int	iStartIO	读取值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iReadSize	读取容量	Input
short[n]	sData	从缓冲存储器中读取的值	Output

(4) 说明

- 对于 iStartIO 中指定的模块的 I/O 编号，应指定将实际的 I/O 编号用 16 相除后的值。
- 将位于 iStartIO 中指定的起始 I/O 编号处的特殊功能模块的 iAddress 中指定的缓冲存储器地址的缓冲值进行相当于 iReadSize 容量的读取。
- 进行连接站为 FXCPU 的 CPU COM 通信及 GX Simulator 通信的情况下，在模块起始 I/O 编号中，指定特殊扩展设备的块 No. (0 ~ 7)，在缓冲存储器地址中指定 0 ~ 32767。
- 对于 sData(ipsData)，应预留出 iReadSize 以上的数组。

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

(6) 多 CPU 系统配置时的缓冲存储器读取 / 写入时的注意事项

如果在实际配置的多 CPU 系统与 GX Developer 中设置的 I/O 分配不相同的情况下执行函数，将发生下述现象。

应在对 GX Developer 中的 I/O 分配进行确认，对进行值的读取 / 写入的模块的 I/O 编号进行确认之后，再执行函数。

- 即使指定正确的 I/O 编号执行函数也发生出错。
- 如果指定了特定的 I/O 编号（实际 I/O 分配错误的 I/O 编号），虽然可以正常执行缓冲存储器读取，但如果进行缓冲存储器写入则发生出错（出错代码：0x010A4030、0x010A4042 等）。
- 虽然在多 CPU 系统中未发生可编程控制器 CPU 的出错（参数出错、SP.UNIT LAY ERR 等），但如果执行函数则用户应用程序中将发生出错。

要 点

- 如果对 RCPU、R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
- 对于 sData(ipsData)，应预留出相当于 iReadSize 中指定点数的存储器区域。
没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 对 QCPU(Q 模式) 进行缓冲存储器读取 (ReadBuffer) 的情况下，只能对 Q 系列专用模块进行读取操作。
此外，也不能从 QCPU(Q 模式) 的共享存储器中进行读取操作。
- 关于通信路径的使用可否，请参阅  338 页 5.2.9 项中记载的一览表。

5.3.10 WriteBuffer(缓冲存储器写入)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

写入特殊功能模块的缓冲存储器的值。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.WriteBuffer(iStartIO、iAddress、iWriteSize、sData(0))
```

Integer	iRet	返回值	Output
Integer	iStartIO	写入值的模块的 I/O 编号	Input
Integer	iAddress	缓冲存储器的地址	Input
Integer	iWriteSize	写入的容量	Input
Short	sData(n)	写入到缓冲存储器中的值	Output

(b) Visual C++® .NET

```
iRet = object.WriteBuffer(iStartIO、iAddress、iWriteSize、*ipsData)
```

int	iRet	返回值	Output
int	iStartIO	写入值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iWriteSize	写入的容量	Input
Short*	ipsData	写入到缓冲存储器中的值	Output

(c) Visual C#® .NET

```
iRet = object. WriteBuffer (iStartIO、iAddress、iWriteSize、ref sData)
```

int	iRet	返回值	Output
int	iStartIO	写入值的模块的 I/O 编号	Input
int	iAddress	缓冲存储器的地址	Input
int	iWriteSize	写入的容量	Input
short[n]	sData	写入到缓冲存储器中的值	Input

(4) 说明

- 对于 iStartIO 中指定的模块的 I/O 编号，应指定将实际的 I/O 编号用 16 相除后的值。
- 将位于 iStartIO 中指定的起始 I/O 编号处的特殊功能模块的 iAddress 中指定的缓冲存储器地址的缓冲值进行相当于 iWriteSize 容量的写入。
- 进行连接站为 FXCPU 的 CPU COM 通信及 GX Simulator 通信的情况下，在模块起始 I/O 编号中指定特殊扩展设备的块 No. (0 ~ 7)，在缓冲存储器地址中指定 0 ~ 32767。
- 对于 sData(ipsData)，应预留出 iWriteSize 以上的数组。

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

(6) 多 CPU 系统配置时的缓冲存储器读取 / 写入时的注意事项

如果在实际配置的多 CPU 系统与 GX Developer 中设置的 I/O 分配不相同的情况下执行函数，将发生下述现象。

应在对 GX Developer 中的 I/O 分配进行确认，对进行值的读取 / 写入的模块的 I/O 编号进行确认之后，再执行函数。

- 即使指定正确的 I/O 编号执行函数也发生出错。
- 如果指定了特定的 I/O 编号（实际 I/O 分配错误的 I/O 编号），虽然可以正常执行缓冲存储器读取，但如果进行缓冲存储器写入则发生出错（出错代码：0x010A4030、0x010A4042 等）。
- 虽然在多 CPU 系统中未发生可编程控制器 CPU 的出错（参数出错、SP.UNIT LAY ERR 等），但如果执行函数则用户应用程序中将发生出错。

要 点

- 如果对 RCPU、R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
- 对于 sData(lpsData)，应预留出相当于 iWriteSize 中指定点数的存储器区域。
没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 对 QCPU(Q 模式) 进行缓冲存储器写入 (WriteBuffer) 的情况下，只能对 Q 系列专用模块进行写入操作。
此外，也不能对 QCPU(Q 模式) 的共享存储器进行写入操作。
- 关于通信路径的使用可否，请参阅  343 页 5.2.10 项中记载的一览表。

5.3.11 GetClockData(时钟数据读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

从可编程控制器 CPU 的时钟数据中读取时间。

(3) 格式

(a) Visual Basic® .NET

```
IRet = object.GetClockData(sYear、sMonth、sDay、sDayOfWeek、sHour、sMinute、sSecond)
```

Integer	lRet	返回值	Output
Short	sYear	读取的年的值	Output
Short	sMonth	读取的月的值	Output
Short	sDay	读取的日期的值	Output
Short	sDayOfWeek	读取的星期的值	Output
Short	sHour	读取的小时的值	Output
Short	sMinute	读取的分钟的值	Output
Short	sSecond	读取的秒的值	Output

(b) Visual C++® .NET

```
iRet = object.GetClockData(*ipsYear、*ipsMonth、*ipsDay、*ipsDayOfWeek、*ipsHour、*ipsMinute、*ipsSecond)
```

int	iRet	返回值	Output
short*	ipsYear	读取的年的值	Output
short*	ipsMonth	读取的月的值	Output
short*	ipsDay	读取的日期的值	Output
short*	ipsDayOfWeek	读取的星期的值	Output
short*	ipsHour	读取的小时的值	Output
short*	ipsMinute	读取的分钟的值	Output
short*	ipsSecond	读取的秒的值	Output

(c) Visual C#® .NET

```
hResult = object.GetClockData(ref sYear、ref sMonth、ref sDay、ref sDayOfWeek、ref sHour、ref sMinute、ref sSecond)
```

int	iRet	返回值	Output
short	sYear	读取的年的值	Output
short	sMonth	读取的月的值	Output
short	sDay	读取的日期的值	Output
short	sDayOfWeek	读取的星期的值	Output
short	sHour	读取的小时的值	Output
short	sMinute	读取的分钟的值	Output
short	sSecond	读取的秒的值	Output

(4) 说明

- 可编程控制器 CPU 中未设置正确的时钟数据的情况下，将返回出错信息。
- 对于 sYear(ipsYear) 中存储的值，RCPU 及 QCPU(Q 模式) 返回公历 4 位，除此以外的 CPU 时将返回公历低 2 位。
但是，RCPU 及 QCPU(Q 模式) 的范围为从 1980 年开始至 2079 年为止。
- 在 sDayOfWeek(ipsDayOfWeek) 中存储的值如下所示。

值	星期
0	星期日
1	星期一
2	星期二
3	星期三
4	星期四
5	星期五
6	星期六

(5) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)



-
- 如果对 R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
 - 对于 FXCPU，只有时钟功能内置的机型或安装了 RTC 盒的 FXU、FX2C、FX2NC 可以读取时钟数据。上述以外的 FXCPU 将返回出错信息。
 - 进行时钟设置时将产生相当于传送时间的误差，应加以注意。
 - 关于通信路径的使用可否，请参阅  348 页 5.2.11 项中记载的一览表。
-

5.3.12 SetClockData(时钟数据写入)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

向可编程控制器 CPU 的时钟数据写入时间。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.SetClockData(sYear、sMonth、sDay、sDayOfWeek、sHour、sMinute、sSecond)
```

Integer	lRet	返回值	Output
Short	sYear	写入年的值	Input
Short	sMonth	写入月的值	Input
Short	sDay	写入日的值	Input
Short	sDayOfWeek	写入星期的值	Input
Short	sHour	写入小时的值	Input
Short	sMinute	写入分钟的值	Input
Short	sSecond	写入秒的值	Input

(b) Visual C++® .NET

```
iRet = object.SetClockData(sYear、sMonth、sDay、sDayOfWeek、sHour、sMinute、sSecond)
```

int	iRet	返回值	Output
short	sYear	写入年的值	Input
short	sMonth	写入月的值	Input
short	sDay	写入日的值	Input
short	sDayOfWeek	写入星期的值	Input
short	sHour	写入小时的值	Input
short	sMinute	写入分钟的值	Input
short	sSecond	写入秒的值	Input

(c) Visual C#® .NET

```
lRet = object.SetClockData(sYear、sMonth、sDay、sDayOfWeek、sHour、sMinute、sSecond)
```

int	iRet	返回值	Output
short	sYear	写入年的值	Input
short	sMonth	写入月的值	Input
short	sDay	写入日的值	Input
short	sDayOfWeek	写入星期的值	Input
short	sHour	写入小时的值	Input
short	sMinute	写入分钟的值	Input
short	sSecond	写入秒的值	Input

(4) 说明

- 设置的时钟数据不是正确的值的情况下，将返回出错信息。
- 对于 sYear 中指定的值，RCPU 及 QCPU(Q 模式) 时公历 4 位有效，除此以外的 CPU 时为公历 2 位有效。但是，RCPU 及 QCPU(Q 模式) 时的范围为从 1980 年开始至 2079 年为止。
在 RCPU 及 QCPU(Q 模式) 以外的 CPU 中设置 4 位的情况下将发生出错。
- sDayOfWeek 中指定的值如下所示。

值	星期
0	星期日
1	星期一
2	星期二
3	星期三
4	星期四
5	星期五
6	星期六

(5) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。（ 496 页第 7 章 出错代码）



-
- 如果对 R 运动 CPU 及 Q 运动 CPU 进行访问，将返回出错信息。
 - 对于 FXCPU，只有时钟功能内置的机型或安装了 RTC 盒的 FXU、FX2C、FX2NC 可以写入时钟数据。上述以外的 FXCPU 将返回出错信息。
 - 进行时钟设置时将产生相当于传送时间的误差，应加以注意。
 - 关于通信路径的使用可否，请参阅 353 页 5.2.12 项 SetClockData(时钟数据写入)。
-

5.3.13 GetCpuType(可编程控制器 CPU 型号读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

读取可编程控制器 CPU, 网卡, GOT 的型号字符串及型号代码。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.GetCpuType(szCpuName, ICpuType)
```

Integer	iRet	返回值	Output
String	szCpuName	可编程控制器 CPU 型号字符串	Output
Integer	ICpuType	可编程控制器 CPU 型号代码	Output

(b) Visual C++® .NET

```
iRet = object.GetCpuType(**szCpuName, *ipiCpuType)
```

int	iRet	返回值	Output
String**	szCpuName	可编程控制器 CPU 型号字符串	Output
int*	ipiCpuType	可编程控制器 CPU 型号代码	Output

(c) Visual C#® .NET

```
iRet = object.GetCpuType (ref szCpuName, ref iCpuType)
```

int	iRet	返回值	Output
String	szCpuName	可编程控制器 CPU 型号字符串	Output
int	iCpuType	可编程控制器 CPU 型号代码	Output

(4) 说明

- 将正在进行通信的可编程控制器 CPU 的型号存储到 szCpuName 中, 将型号代码存储到 iCpuType(ipiCpuType) 中。
- 可编程控制器 CPU 型号字符串将通过 UNICODE 返回。

(5) CPU 型号字符串及型号代码

 请参阅 358 页 5.2.13 项中记载的一览表。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

5.3.14 SetCpuStatus(远程控制)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行可编程控制器 CPU 的远程操作。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.SetCpuStatus(I0peration)
```

Integer	iRet	返回值	Output
Integer	I0peration	远程 RUN/STOP/PAUSE	Input

(b) Visual C++® .NET

```
iRet = object.SetCpuStatus (i0peration)
```

int	iRet	返回值	Output
int	i0peration	远程 RUN/STOP/PAUSE	Input

(c) Visual C#® .NET

```
iRet = object.SetCpuStatus (i0peration)
```

int	iRet	返回值	Output
int	i0peration	远程 RUN/STOP/PAUSE	Input

(4) 说明

进行通过 i0peration(i0peration) 指定的操作。

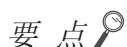
指定了下述以外的值时将发生出错。

值	操作
0	远程 RUN
1	远程 STOP
2	远程 PAUSE

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)



- 由于 FXCPU 作为可编程控制器 CPU 未配备 PAUSE 开关，因此通过 SetCpuStatus 指定远程 PAUSE 时将返回出错信息。
- 对 Q 运动 CPU 进行访问，指定 PAUSE 时将返回出错信息。
- 关于通信路径的使用可否，请参阅  362 页 5.2.14 项中记载的一览表。
- 如果对 R 运动 CPU 进行访问，将返回出错信息。

5.3.15 EntryDeviceStatus(软元件的状态监视登录)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

登录进行状态监视的软元件。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.EntryDeviceStatus(szLabelList, iSize, iMonitorCycle, iData(0))
```

Integer	iRet	返回值	Output
String	szLabelList(n)	登录标签名列表	Input
Integer	iSize	登录软元件点数	Input
Integer	iMonitorCycle	状态监视间隔时间	Input
Integer	iData(n)	登录软元件值列表	Input

(b) Visual C++® .NET

```
iRet = object.EntryDeviceStatus(**szLabelList, iSize, iMonitorCycle, *arriData)
```

int	iRet	返回值	Output
String**	szLabelList	登录标签名列表	Input
int	iSize	登录软元件点数	Input
int	iMonitorCycle	状态监视间隔时间	Input
int*	arriData	登录软元件值列表	Input

(c) Visual C#® .NET

```
iRet = object.EntryDeviceStatus(szLabelList, iSize, iMonitorCycle, ref iData)
```

int	iRet	返回值	Output
System.String[]	szLabelList	登录标签名列表	Input
int	iSize	登录软元件点数	Input
int	iMonitorCycle	状态监视间隔时间	Input
int[n]	iData	登录软元件值列表	Input

(4) 说明

- 对通过 szLabelList 指定的 iSize 容量的软元件群是否处于 iData(arriData) 中指定的状态进行确认。确认时间是通过 iMonitorCycle 进行指定。
根据状态成立执行用户应用程序的 OnDeviceStatus 函数。
- 在 iSize 中可指定的最大软元件点数为 20 点。
- 对于 iMonitorCycle 应以 1 秒～ 1 小时的范围（以 1 ～ 3600 的秒单位进行设置）进行指定。
指定超出此范围的值时将发生出错。
- 将登录软元件值列表存储到 iData(arriData) 中。

(5) 软元件指定方法

标签名及登录软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
基本型	位、字	标签名

- 登录的软元件值应按以下方式进行设置。

〈指定位软元件及字软元件时〉

(例) 将 M0 及 D0 逐点进行登录

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	位	M0
[1] LABEL2	字	D0

- 登录软元件点数：2

- 登录软元件值

高位 2 字节	低位 2 字节
不使用。	M0
	D0

〈指定 FXCPU 的 CN200 以后时〉

(例) 包括 CN200 登录 3 点

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	字	D0
[1] LABEL2	字	CN200
[2] LABEL3	字	D1

- 登录软元件点数：3

- 登录软元件值

高位 2 字节	低位 2 字节
不使用。	D0
CN200 的 H	CN200 的 L ^{*1}
不使用。	D1

*1: 对于 FXCPU 的 CN200 以后，可以登录 4 字节。

〈指定 FD 软元件时 (4 字软元件)〉

(例) 包括 FD0 登录 3 点

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	字	D0
[1] LABEL2	字	FD0
[2] LABEL3	字	D1

- 登录软元件点数：3

- 登录软元件值

高位 2 字节	低位 2 字节
不使用。	D0
	FD0 的 LL ^{*2}
	D1

*2: 只能登录低位 2 字节。指定软元件的 HH、HL、LH
(高位 6 字节) 不能登录。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。(参见 496 页第 7 章 出错代码)

(7) 字软元件状态监视情况下的注意事项

对字软元件为负的值 $-1 \sim -32768$ ($FFFFH \sim 8000H$) 的状态进行监视的情况下，应设置 EntryDeviceStatus 的监视软元件值的高位 2 字节中存储了“0”的 $65535 \sim 32768$ ($0000FFFFH \sim 00008000H$)。

(例) 对 D0 变为“-10”进行状态监视的情况下

应将“ $-10(FFFFFFFFFF6H)$ ”的高位 2 字节中存储了“0”的值“ $65526(0000FFF6H)$ ”设置为监视软元件值。

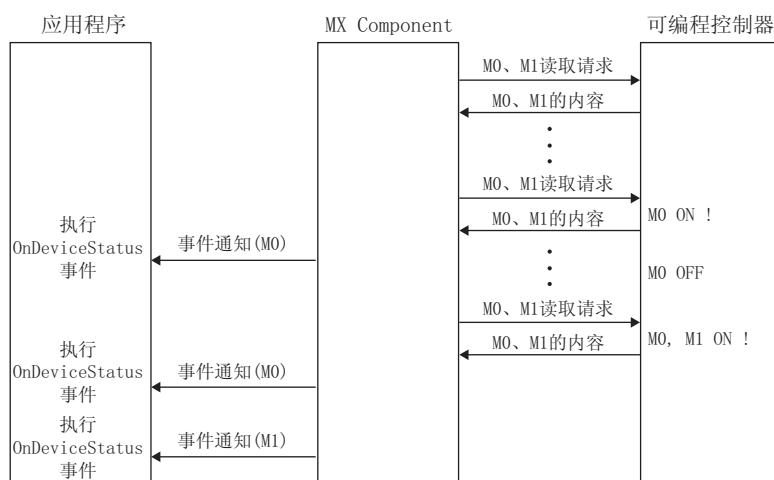
由于可编程控制器 CPU 的字软元件为 WORD 型，而 EntryDeviceStatus 的监视软元件值的类型为 LONG 型，因此将可编程控制器 CPU 的当前值与 EntryDeviceStatus 的监视软元件值进行了比较的情况下，值不匹配，因此需要进行上述设置。(使用位软元件及双字软元件时，本注意事项不适用。)

关于本注意事项的编程示例，请参阅下述内容。

☞ 534 页附录 3 关于超时时间

要 点

- 根据计算机性能、执行中的应用程序负载、与可编程控制器通信所需时间等的条件，有可能在指定的状态监视间隔时间内无法进行软元件的状态监视。
此外，在同时使用控件的其它函数的情况下，也可能导致无法在指定的状态监视间隔时间内进行软元件的状态监视。
- 对于 iData(arriData)，应预留出相当于 iSize 中指定点数的存储器区域。
没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 在状态监视中执行了 EntryDeviceStatus 的情况下，将发生出错。更改状态监视条件的情况下，应执行 FreeDeviceStatus 之后，再次执行 EntryDeviceStatus。
- 多个软元件同时发生了状态变化的情况下，每次状态变化时，均将执行 OnDeviceStatus 事件。
(例：监视 M0 的情况下)



- 该功能是控件定期执行软元件随机读取，确认状态成立的功能。因此，不是将可编程控制器 CPU 的软元件的状态成立通知到 MX Component 中的功能。所以，根据指定的状态监视间隔时间，控件有可能无法确认可编程控制器 CPU 的软元件的状态成立。
- 位软元件的位数指定及变址修饰不能使用。

5.3.16 FreeDeviceStatus(软元件的状态监视登录解除)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

对通过 EntryDeviceStatus 登录的执行状态监视的软元件进行解除。

(3) 格式

(a) Visual Basic® .NET

IRet = object.FreeDeviceStatus()			
Integer	IRet	返回值	Output

(b) Visual C++® .NET

iRet = object.FreeDeviceStatus()			
int	iRet	返回值	Output

(c) Visual C#® .NET

iRet = object.FreeDeviceStatus()			
int	iRet	返回值	Output

(4) 说明

对通过 EntryDeviceStatus 函数设置的执行状态监视的软元件进行解除。

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

5.3.17 OnDeviceStatus(事件通知)

(1) 可以使用的控件

在 DotUtilType 控件中可以使用。

(2) 功能

在通过 EntryDeviceStatus 函数登录的软元件条件成立时执行此功能。

(3) 格式

(a) Visual Basic® .NET

```
Private Sub object_OnDeviceStatus(sender、e)
    ByVal sender As System.Object
    ByVal e As objectLib_IobjectIFEvents_OnDeviceStatusEvent
        e 的成员如下所示。
        e.szLabel      条件成立的标签名
        e.lData        条件成立的软元件值
        e.lReturnCode   条件检查处理的返回值
```

事件的发生源	Input
事件的数据	Input

(b) Visual C++® .NET

```
private: System::Void objectIF_OnDeviceStatus (*sender、e)
    System::Object *sender
    objectLib::objectIF_OnDeviceStatusEvent* e
        e 的成员如下所示。
        e->szLabel      条件成立的标签名
        e->lData        条件成立的软元件值
        e->lReturnCode   条件检查处理的返回值
```

事件的发生源	Input
事件的数据	Input

(c) Visual C#® .NET

```
private void object_OnDeviceStatus(object sender、object.DeviceStatusEventArgs e)
    sender
    e
        e 的成员如下所示。
        e->szLabel      条件成立的标签名
        e->lData        条件成立的软元件值
        e->lReturnCode   条件检查处理的返回值
```

事件的发生源	output
事件的数据	output

(4) 说明

- 在通过 EntryDeviceStatus 函数登录的软元件条件成立时对应用程序进行事件通知。
通过在用户应用程序侧安装本函数，当登录的软元件条件成立时，可以接受事件。
- 在 lData 中，输入通过 EntryDeviceStatus 登录的软元件值。
(例) 在通过字软元件对“-1”的值进行监视的情况下
在 EntryDeviceStatus 中将 65535(0000FFFFH) 设置为登录软元件值。
可编程控制器 CPU 的对象字软元件变为“-1”(FFFFH) 的状态时，将执行 OnDeviceStatus，在 iData(varData) 中输入 (0000FFFFH)。

(5) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。(☞ 496 页第 7 章 出错代码)

要 点

- 在用户应用程序内进行下述设置的情况下，即使 EntryDeviceStatus 函数中登录的软元件的条件成立也不发生 OnDeviceStatus 的事件。变为事件发生等待的情况下，在下述设置结束之前控制将不返回到 .NET 控件，软元件管理处理将停止，因此应加以注意。

- 通过 Visual Basic® .NET 创建的用户应用程序
在用户应用程序内显示信息框的情况下
在用户应用程序内显示 InputBox/OutputBox 的情况下
- 通过 Visual Basic® .NET、Visual C++® .NET 及 Visual C#® .NET 创建的用户应用程序在用户应用程序内使用 Sleep 处理、WaitForSingleObject 函数等的待机函数的情况下

5.3.18 ReadDeviceBlock2(软元件的批量读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

以 2 字节数据进行软元件的批量读取。

(3) 格式

(a) Visual Basic® .NET

```
IRet = object.ReadDeviceBlock2(szLabel, iSize, sData)
```

Integer	IRet	返回值	Output
String	szLabel	标签名	Input
Integer	iSize	读取点数	Input
Short	sData(n)	读取的软元件值	Output

(b) Visual C++® .NET

```
IRet = object.ReadDeviceBlock2(*szLabel, iSize, *ipsData)
```

int	IRet	返回值	Output
String*	szLabel	标签名	Input
int	iSize	读取点数	Input
short*	lpsData	读取的软元件值	Output

(c) Visual C#® .NET

```
IRet = object.ReadDeviceBlock2(ref szLabel, iSize, ref sData)
```

int	IRet	返回值	Output
String	szLabel	标签名	Input
int	iSize	读取点数	Input
short[n]	sData	读取的软元件值	Output

(4) 说明

- 从通过标签名 szLabel 指定的软元件开始，对读取点数 iSize 容量的软元件进行批量读取。
- 读取的软元件值将被存储到 sData(ipsData) 中。
- 对于读取的软元件值 sData(ipsData)，应预留出读取点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及读取软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
		标签名
数组 成员	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、 字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数 器、长计数器、累计定时器、长累计定时器	标签名 [要素数] 标签名 [n1] [n2] [n3]

*1： 最多可指定半角 32 个字符 +NULL。

- 在读取点数中应根据标签的数据类型设置以下值。

标签(数组)的数据类型	对应的字数	指定的读取点数
位	1	标签的数组要素数 ÷ 16(进位)
字	1	标签的数组要素数
双字、单精度实数	2	标签的数组要素数 × 2
双精度实数	4	标签的数组要素数 × 4
字符串	17	标签的数组要素数 × 17
字符串[Unicode]	33	标签的数组元素数 × 33
时间	2	标签的数组要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的数组要素数

- 读取的软元件值按以下方式被存储。

<指定位软元件时>

(例) 从 M0 开始读取 3 点(3 字 = 48 位)

- 标签设置(数据类型：位；数组要素数：48)

数据类型	软元件
位(0..47)	M0

- 读取点数：3

- 读取的软元件值

2字节
M0 ~ M15*3
M16 ~ M31*3
M32 ~ M47*3

*3：按照软元件的编号顺序从低位开始存储。

<指定 FXCPU 的 CN200 以后时>

(例) 从 CN200 开始读取 6 点 *4

- 标签设置(数据类型：双字；数组要素数：3)

数据类型	软元件
双字(0..2)	CN200

- 读取点数：6

- 读取的软元件值

2字节
CN200 的 L
CN200 的 H
CN201 的 L
CN201 的 H
CN202 的 L
CN202 的 H

*4：对于 FXCPU 的 CN200 以后，按照每个软元件读取 2 点的方式分别读取高位(H) 及低位(L) 的数据。
如果读取 1 点数据将发生出错。

<指定字软元件时>

(例) 从 D0 开始读取 3 点

- 标签设置(数据类型：字；数组要素数：3)

数据类型	软元件
字(0..2)	D0

- 读取点数：3

- 读取的软元件值

2字节
D0
D1
D2

<指定 FD 软元件时(4 字软元件)>

(例) 从 FD0 开始读取 8 点

- 标签设置(数据类型：双字；数组要素数：2)

数据类型	软元件
双字(0..1)	FD0

- 读取点数：8

- 读取的软元件值

2字节
FD0 的 LL
FD0 的 LH
FD0 的 HL
FD0 的 HH
FD1 的 LL
FD1 的 LH
FD1 的 HL
FD1 的 HH

5.3 函数的详细内容(.NET 控件用)
5.3.18 ReadDeviceBlock2(软元件的批量读取)

< 标签为双字数组指定字软元件时 >

(例) 从 D100 中读取 6 点

- 标签设置 (数据类型：双字；数组要素数：3)

数据类型	软元件
双字 (0..2)	D100

- 读取点数：6

- 读取的软元件值

2 字节
D100
D101
D102
D103
D104
D105

(6) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 读取点数中可指定的最大读取点数为满足下述条件的范围。
读取开始软元件编号 + 读取点数 ≤ 最终软元件编号
- 指定位型数组的标签的情况下，软元件编号只能指定为 0 或 16 的倍数。
- 应将标签名中指定的数据类型所对应的字数指定为读取点数。读取的软元件值中，应预留出相当于读取点数中指定的点数容量的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法读取。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法读取。
- 位软元件的位数指定及变址修饰不能使用。

5.3.19 WriteDeviceBlock2(软元件的批量写入)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

以 2 字节数据进行软元件的批量写入。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.WriteDeviceBlock2(szLabel, iSize, sData)
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Integer	iSize	写入点数	Input
Short	sData(n)	写入的软元件值	Input

(b) Visual C++® .NET

```
iRet = object.WriteDeviceBlock2(*szLabel, iSize, *ipsData)
```

int	iRet	返回值	Output
String*	szLabel	标签名	Input
int	iSize	写入点数	Input
short*	ipsData	写入的软元件值	Input

(c) Visual C#® .NET

```
iRet = object.WriteDeviceBlock2(ref szLabel, iSize, sData)
```

int	iRet	返回值	Output
String	szLabel	标签名	Input
int	iSize	写入点数	Input
short[n]	sData	写入的软元件值	Input

(4) 说明

- 从标签名 szLabel 中指定的软元件开始，批量读取写入点数 iSize 容量的软元件。
- 写入的软元件值被存储到 sData(ipsData) 中。
- 对于写入的软元件值 sData(ipsData)，应预留出写入点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及写入的软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
数组 成员	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、字符串[Unicode] ^{*1} 、时间、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	标签名 标签名 [要素数] 标签名 [n1] [n2] [n3]

*1：最多可指定半角 32 个字符 +NULL。

- 在写入点数中应根据标签的数据类型设置以下值。

标签(数组)的数据类型	对应的字数	指定的写入点数
位	1	标签的数组要素数 ÷ 16(进位)
字	1	标签的数组要素数
双字、单精度实数	2	标签的数组要素数 × 2
双精度实数	4	标签的数组要素数 × 4
字符串	17	标签的数组要素数 × 17
字符串[Unicode]	33	标签的数组元素数 × 33
时间	2	标签的数组要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的数组要素数

- 写入的软元件值应按以下方式进行设置。

< 指定位软元件时 >

(例) 从 M0 开始写入 3 点(3 字 = 48 位)

- 标签设置(数据类型：位，数组要素数：48)

数据类型	软元件
位(0..47)	M0

- 写入点数：3

- 写入的软元件值

2 字节
M0 ~ M15*3
M16 ~ M31*3
M32 ~ M47*3

*3：按软元件的编号顺序从低位开始存储。

< 指定 FXCPU 的 CN200 以后时 >

(例) 从 CN200 开始写入 6 点*4

- 标签设置(数据类型：双字，数组要素数：3)

数据类型	软元件
双字(0..2)	CN200

- 写入点数：6

- 写入的软元件值

2 字节
CN200 的 L
CN200 的 H
CN201 的 L
CN201 的 H
CN202 的 L
CN202 的 H

*4：对于 FXCPU 的 CN200 以后，按照每个软元件读取 2 点的方式分别写入高位(H)及低位(L)的数据。
如果写入 1 点将发生出错。

< 指定字软元件时 >

(例) 从 D0 开始写入 3 点

- 标签设置(数据类型：字，数组要素数：3)

数据类型	软元件
字(0..2)	D0

- 写入点数：3

- 写入的软元件值

2 字节
D0
D1
D2

< 指定 FD 软元件时(4 字软元件) >

(例) 从 FD0 开始写入 8 点

- 标签设置(数据类型：双字，数组要素数：2)

数据类型	软元件
双字(0..1)	FD0

- 写入点数：8

- 写入的软元件值

2 字节
FD0 的 LL
FD0 的 LH
FD0 的 HL
FD0 的 HH
FD1 的 LL
FD1 的 LH
FD1 的 HL
FD1 的 HH

<标签为双字数组指定字软元件时>

(例) 从 D100 开始写入 6 点

- 标签设置 (数据类型: 双字; 数组要素数: 3)

数据类型	软元件
双字 (0..2)	D100

- 写入点数: 6

- 写入的软元件值

2 字节
D100
D101
D102
D103
D104
D105

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

要 点

- 写入点数中可指定的最大写入点数为满足下述条件的范围。
写入开始软元件编号 + 写入点数 ≤ 最终软元件编号
- 指定位型数组的标签的情况下, 对软元件编号只能指定 0 或 16 的倍数。
- 应将标签名中指定的数据类型所对应的字数指定为读取点数。在写入的软元件值中, 应预留出相当于写入点数中指定的点数容量的存储器区域。没有存储器区域的情况下, 有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法写入。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法写入。
- 位软元件的位数指定及变址修饰不能使用。
- 向长定时器软元件 (LT) 及累积长定时器软元件 (LST) 写入时, 不能使用 WriteDeviceBlock2。
请使用 WriteDeviceRandom2 或 SetDevice2。

5.3.20 ReadDeviceRandom2(软元件的随机读取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

以 2 字节数据进行软元件的随机读取。

(3) 格式

(a) Visual Basic® .NET

```
IRet = object.ReadDeviceRandom2(szLabel, iSize, sData(0))  
      Integer          IRet          返回值          Output  
      String           szLabel       标签名          Input  
      Integer          iSize         读取点数        Input  
      Short            sData(n)     读取的软元件值  Output  
  
IRet = object.ReadDeviceRandom2(szLabelList, iSize, sData(0))  
      Integer          IRet          返回值          Output  
      String           szLabelList(n)  标签列表      Input  
      Integer          iSize         读取点数        Input  
      Short            sData(n)     读取的软元件值  Output
```

(b) Visual C++® .NET

```
iRet = object.ReadDeviceRandom2(*szLabel, iSize, *arrsData)  
      Int             iRet          返回值          Output  
      String*         szLabel       标签名          Input  
      int             iSize         读取点数        Input  
      short*          arrsData     读取的软元件值  Output  
  
iRet = object.ReadDeviceRandom2(**szLabelList, iSize, *arrsData)  
      Int             iRet          返回值          Output  
      String**        szLabelList   标签列表      Input  
      int             iSize         读取点数        Input  
      short*          arrsData     读取的软元件值  Output
```

(c) Visual C#® .NET

```
iRet = object.ReadDeviceRandom2(ref szLabel, iSize, ref sData)  
      Int             iRet          返回值          Output  
      String          szLabel       标签名          Input  
      int             iSize         读取点数        Input  
      short[n]        sData        读取的软元件值  Output  
  
iRet = object.ReadDeviceRandom2(ref szLabelList, iSize, ref sData)  
      Int             iRet          返回值          Output  
      System.String[] szLabelList  标签列表      Input  
      int             iSize         读取点数        Input  
      short[n]        sData        读取的软元件值  Output
```

(4) 说明

- 将通过标签名 szLabel (szLabelList) 指定的软元件群只读取相当于读取点数 iSize 容量的软元件值。
- 读取的软元件值将被存储到 sData (arrsData) 中。
- 对于读取的软元件值 sData (arrsData)，应预留出读取点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及读取软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型		标签名的格式
基本型	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器		标签名
数组	成员	(可以进行与基本型相同的指定)	
结构体	成员	(可以进行与基本型相同的指定)	
结构体数组	成员	(可以进行与基本型相同的指定)	

*1：最多可指定半角 32 个字符 +NULL。

- 在读取点数中应根据标签的数据类型，设置各要素相关的以下值的和。

标签的数据类型	对应的字数	指定的读取点数
位、字	1	标签的要素数
双字、单精度实数	2	标签的要素数 × 2
双精度实数	4	标签的要素数 × 4
字符串	17	标签的要素数 × 17
字符串 [Unicode]	33	标签的数组元素数 × 33
时间	2	标签的要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的要素数

- 读取的软元件值按以下方式被存储。

< 指定位软元件及字软元件时 >

(例) 将 M0 及 D0 逐点进行读取

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	位
[1]	LABEL2	字

- 读取点数：2

- 读取的软元件值

2 字节	对应标签
M0*1	LABEL1
D0	LABEL2

*1: 仅将 “M0” 1 点作为读取软元件对象，软元件值将被存储 “0” 或 “1”。

< 指定 FXCPU 的 CN200 以后时 >

(例) 包括 CN200 读取 3 点

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	字
[1]	LABEL2	字
[2]	LABEL3	字

- 读取点数：3

- 读取的软元件值

2 字节	对应标签
D0	LABEL1
CN200 的 L*2	LABEL2
D1	LABEL3

*2: 仅读取低位 2 字节。指定软元件的 H(高位 2 字节)不被读取。

< 指定 FD 软元件时 (4 字软元件) >

(例) 包括 FDO 读取 3 点

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	字
[1]	LABEL2	字
[2]	LABEL3	字

- 读取点数：3

- 读取的软元件值

2 字节	对应标签
D0	LABEL1
FDO 的 LL*3	LABEL2
D1	LABEL3

*3: 仅读取低位 2 字节。指定软元件的 HH、HL、LH(高位 6 字节)不被读取。

〈在标签中指定相当于2字以上的数据类型时〉

(例) 指定双字、单精度实数、双精度实数、字符串^{*5}及时间型标签进行读取

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	双字	D0
[1] LABEL2	单精度实数	D100
[2] LABEL3	双精度实数	D200
[3] LABEL4	字符串	D300
[4] LABEL5	时间	D400

- 读取点数：27

- 读取的软元件值

2字节	对应标签
D0	LABEL1
D1	
D100	LABEL2
D101	
D200	
D201	LABEL3
D202	
D203	
D300	
:	LABEL4
D316 ^{*4}	
D400	LABEL5
D401	

*4: 读取字符串17点(半角32字符+NULL)。由于不能进行字符串型的字符转换，因此需要在用户程序内进行转换。

〈指定数组型标签时〉

(例) 指定数组型标签进行读取

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	位(0..1)	M0
[1] LABEL2	字(0..1)	CN200
[2] LABEL3	双字(0..1)	D0

- 读取点数：8

- 读取的软元件值

2字节	对应标签
M0	LABEL1[0]
M1	LABEL1[1]
CN200的L	LABEL2[0]
CN201的L	LABEL2[1]
D100	LABEL3[0]
D101	
D102	LABEL3[1]
D103	

〈指定结构体型标签时〉

(例) 指定结构体型标签进行读取

- 结构体设置

结构体名	标签名	数据类型
STRUCT	L1	位
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0]	LABEL1	STRUCT	L1 D0.0
			L2 D0
[1]	LABEL2	STRUCT	M10
			CN200

- 读取点数：6

- 读取的软元件值

2字节	对应标签
D0.0	LABEL1.L1
D0	LABEL1.L2
D1	
M0	LABEL2.L1
CN200 的 L	LABEL2.L2
CN201 的 L ⁵	

*5: 双字型标签中设置了CN200以后的软元件的情况下，相当于2软元件的容量将被读取。

〈指定组合了结构体及数组的标签时〉

(例) 指定并读取结构体型的数组及结构体数组型标签

- 结构体设置

结构体名	标签名	数据类型
STRUCT1	L1	位
	L2	字
STRUCT2	L1	位(0..2)
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0]	LABEL1	STRUCT1 (0..1)	L1 X0
			L2 D0
[1]	LABEL2	STRUCT2	M1 M0
			D100 D100

- 读取点数：9

- 读取的软元件值

2字节	对应标签
X0	LABEL1[0].L1
D0	LABEL1[0].L2
X1	LABEL1[1].L1
D1	LABEL1[1].L2
M0	LABEL2.L1[0]
M1	LABEL2.L1[1]
M2	LABEL2.L1[2]
D100	LABEL2.L2
D101	

(6) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。（ 496 页第 7 章 出错代码）

要 点

- 读取点数中可指定的最大读取点数为 0x7FFFFFFF。
- 应将标签名中指定的数据类型所对应的字数指定为读取点数。在读取的软元件值中，应预留出相当于读取点数中指定的点数容量的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法读取。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法读取。
- 指定了双字软元件的情况下，在本函数中仅存储低位 1 字（2 字节）容量的数据。（不发生出错。）进行双字软元件的读取的情况下，应使用 ReadDeviceRandom 或 ReadDeviceBlock2。
- 位软元件的位数指定及变址修饰不能使用。

5.3.21 WriteDeviceRandom2(软元件的随机写入)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

以 2 字节数据进行软元件的随机写入。

(3) 格式

(a) Visual Basic® .NET

```
Ret = object.WriteDeviceRandom2(szLabel, iSize, sData(0))

    Integer          IRet      返回值      Output
    String           szLabel   标签名      Input
    Integer          iSize     写入点数    Input
    Short            sData(n)  写入的软元件值  Input

Ret = object.WriteDeviceRandom2(szLabelList, iSize, sData(0))

    Integer          IRet      返回值      Output
    String           szLabelList(n)  标签列表  Input
    Integer          iSize     写入点数    Input
    Short            sData(n)  写入的软元件值  Input
```

(b) Visual C++® .NET

```
iRet = object.WriteDeviceRandom2(*szLabel, iSize, *arrsData)

    int             iRet      返回值      Output
    String*         szLabel   标签名      Input
    int             iSize     写入点数    Input
    short*          arrsData  写入的软元件值  Input

iRet = object.WriteDeviceRandom2(**szLabelList, iSize, *arrsData)

    int             iRet      返回值      Output
    String**        szLabelList  标签列表  Input
    int             iSize     写入点数    Input
    short*          arrsData  写入的软元件值  Input
```

(c) Visual C++® .NET

```
iRet = object.WriteDeviceRandom2(ref szLabel, iSize, sData)

    int             iRet      返回值      Output
    String          szLabel   标签名      Input
    int             iSize     写入点数    Input
    short[n]        sData     写入的软元件值  Input

iRet = object. WriteDeviceRandom2 (ref szLabelList, iSize, sData)

    int             iRet      返回值      Output
    System.String[] szLabelList  标签列表  Input
    int             iSize     写入点数    Input
    short[n]        sData     写入的软元件值  Input
```

(4) 说明

- 将标签名 szLabel 中指定的软元件群只写入相当于写入点数 iSize 容量的软元件值。
- 将写入的软元件值存储到 sData(arrsData) 中。
- 对于写入的软元件值 sData(arrsData)，应预留出写入点数 iSize 以上的数组。

(5) 软元件指定方法

标签名及写入的软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型		标签名的格式
基本型	位、字、双字、单精度实数、双精度实数、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器		标签名
数组	成员	(可以进行与基本型相同的指定)	
结构体	成员	(可以进行与基本型相同的指定)	
结构体数组	成员	(可以进行与基本型相同的指定)	

*1： 最多可指定半角 32 个字符 +NULL。

- 在写入点数中应根据标签的数据类型，设置各要素相关的以下值的和。

标签的数据类型	对应的字数	指定的写入点数
位、字	1	标签的要素数
双字、单精度实数	2	标签的要素数 × 2
双精度实数	4	标签的要素数 × 4
字符串	17	标签的要素数 × 17
字符串 [Unicode]	33	标签的数组元素数 × 33
时间	2	标签的要素数 × 2
定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	1	标签的要素数

- 写入的软元件值应按以下方式进行设置。

< 指定位软元件及字软元件时 >

(例) 将 M0 及 D0 逐点进行写入

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	位
[1]	LABEL2	字

- 写入点数：2

- 写入的软元件值

2 字节
MO*1
DO

*1: 仅将“M0”1点作为写入软元件对象，设置的2字节数据的最低位将成为软元件值的写入值。

< 指定 FXCPU 的 CN200 以后时 >

(例) 包括 CN200 写入 3 点 *2

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	字
[1]	LABEL2	CN200
[2]	LABEL3	D1

- 写入点数：3

- 写入的软元件值

2 字节	对应标签
D0	LABEL1
CN200 的 L	LABEL2
D1	LABEL3

*2: 对于 FXCPU 的 CN200 以后，以 1 点写入 4 字节。只能设置低位 2 字节的值。

指定软元件的 H(高位 2 字节) 中将被写入“0”。

< 指定 FD 软元件时 (4 字软元件) >

(例) 包括 FD0 写入 3 点

- 标签设置

szLabelList	数据类型	软元件
[0]	LABEL1	字
[1]	LABEL2	字
[2]	LABEL3	D1

- 写入点数：3

- 写入的软元件值

2 字节	对应标签
D0	LABEL1
FD0 的 LL*3	LABEL2
D1	LABEL3

*3: 只能设置低位 2 字节的值。指定软元件的 HH、HL、LH(高位 6 字节) 中将被写入“0”。

〈在标签中指定相当于2字以上的数据类型时〉

(例) 指定及写入双字、单精度实数、双精度实数、字符串^{*4} 及时间型标签

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	双字	D0
[1] LABEL2	单精度实数	D100
[2] LABEL3	双精度实数	D200
[3] LABEL4	字符串	D300
[4] LABEL5	时间	D400

- 写入点数：27

- 写入的软元件值

2字节	对应标签
D0	LABEL1
D1	
D100	LABEL2
D101	
D200	
D201	
D202	LABEL3
D203	
D300	
:	LABEL4
D316 ^{*4}	
D400	LABEL5
D401	

*4: 写入字符串17点(半角32字符+NULL)。由于不能进行字符串型的字符转换，因此需要在用户程序内进行转换。

〈指定数组型标签时〉

(例) 指定数组型标签进行写入

- 标签设置

szLabelList	数据类型	软元件
[0] LABEL1	位(0..1)	M0
[1] LABEL2	字(0..1)	CN200
[2] LABEL3	双字(0..1)	D0

- 写入点数：8

- 写入的软元件值

2字节	对应标签
M0	LABEL1[0]
M1	LABEL1[1]
CN200的L	LABEL2[0]
CN201的L	LABEL2[1]
D100	LABEL3[0]
D101	
D102	LABEL3[1]
D103	

〈指定结构体型标签时〉

(例) 指定结构体型标签进行写入

- 结构体设置

结构体名	标签名	数据类型
STRUCT	L1	位
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0]	LABEL1	STRUCT	L1
			D0.0
[1]	LABEL2	STRUCT	L1
			M10
			L2
			CN200

• 写入点数：5

• 写入的软元件值

2字节	对应标签
D0.0	LABEL1.L1
D0	LABEL1.L2
D1	
M0	LABEL2.L1
CN200 的 L	LABEL2.L2

〈指定组合了结构体及数组的标签时〉

(例) 指定并写入结构体型的数组及结构体数组型标签

- 结构体设置

结构体名	标签名	数据类型
STRUCT1	L1	位
	L2	字
STRUCT2	L1	位(0..2)
	L2	双字

- 标签设置

szLabelList	数据类型	标签名	软元件
[0]	LABEL1	STRUCT1 (0..1)	L1
			D0
[1]	LABEL2	STRUCT2	L1
			M0
			D100
			D100

• 写入点数：9

• 写入的软元件值

2字节	对应标签
X0	LABEL1[0].L1
D0	LABEL1[0].L2
X1	LABEL1[1].L1
D1	LABEL1[1].L2
M0	LABEL2.L1[0]
M1	LABEL2.L1[1]
M2	LABEL2.L1[2]
D100	LABEL2.L2
D101	

(6) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。（ 496 页第 7 章 出错代码）

要 点

- 写入点数中可指定的最大写入点数为 0x7FFFFFFF 点。
- 应将标签名中指定的数据类型所对应的字数指定为写入点数。在写入的软元件值中，应预留出相当于写入点数中指定的点数容量的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 标签名所对应的软元件不存在的情况下将发生出错而无法写入。
多个标签名所对应的软元件中有部分软元件不存在的情况下也将发生出错而无法写入。
- 指定了双字软元件的情况下，在本函数中将对相当于低位 1 字（2 字节）容量的数据进行写入，在相当于高位 1 字（2 字节）容量的数据中写入“0”。
写入双字软元件的情况下，应使用 WriteDeviceRandom 或 WriteDeviceBlock2。
- 位软元件的位数指定及变址修饰不能使用。
- 如果对 Q 运动 CPU 进行访问，将返回出错信息。

5.3.22 SetDevice2(软元件数据的设置)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

以 2 字节数据进行 1 点软元件的设置。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.SetDevice2(szLabel, sData)
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Short	sData	写入的软元件值	Input

(b) Visual C++® .NET

```
iRet = object.SetDevice2(*szLabel, sData)
```

int	iRet	返回值	Output
String*	szLabel	标签名	Input
short	sData	写入的软元件值	Input

(c) Visual C#® .NET

```
iRet = object.SetDevice2(ref szLabel, sData)
```

int	iRet	返回值	Output
String	szLabel	标签名	Input
short	sData	写入的软元件值	Input

(4) 说明

- 对标签名 szLabel 中指定的 1 点软元件写入所写入的软元件值 sData 的值。
- 位软元件设置时，写入的软元件值 sData 的最低位的位将有效。

(5) 软元件指定方法

标签名及写入的软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
基本型	位、字、双字 ^{*1} 、单精度实数 ^{*1} 、双精度实数 ^{*1} 、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间 ^{*1} 、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	标签名

*1：仅写入起始软元件的低位 2 字节。

希望写入 2 字以上的软元件的情况下应使用 WriteDeviceBlock2 函数。

- 写入的软元件值应按以下方式进行设置。

〈指定位软元件时〉

(例) 写入 M0

- 标签设置

数据类型	软元件
位	M0

- 写入的软元件值

2字节
M0*2

*2: 仅将“M0”1点作为写入软元件对象，设置的2字节数据的最低位将成为软元件值的写入值。

〈指定 FXCPU 的 CN200 以后时〉

(例) 写入 CN200*3

- 标签设置

数据类型	软元件
双字	CN200

- 写入的软元件值

2字节
CN200 的 L

*3: 对于 FXCPU 的 CN200 以后，写入 4 字节。只能设置低位 2 字节的值。指定软元件的 H(高位 2 字节)中将被写入“0”。

〈在标签中指定相当于 2 字以上的数据类型时〉

(例) 指定字符串型标签进行写入

- 标签设置

数据类型	软元件
字符串	D0

- 写入的软元件值

2字节
D0*5

*5: 在起始软元件中只写入低位 2 字节。

〈指定字软元件时〉

(例) 写入 D0

- 标签设置

数据类型	软元件
字	D0

- 写入的软元件值

2字节

D0

〈指定 FD 软元件时 (4 字软元件)〉

(例) 指定 FDO 进行写入

- 标签设置

数据类型	软元件
字	FDO

- 写入的软元件值

2字节

FDO 的 LL*4

*4: 只写入低位 2 字节。指定软元件的 HH、HL、LH(高位 6 字节)不被写入。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。(☞ 496 页第 7 章 出错代码)

要点

- 标签名所对应的软元件不存在的情况下将发生出错而无法写入。
- 指定了双字软元件的情况下，在本函数中对相当于低位 1 字(2字节)容量的数据进行写入，在相当于高位 1 字(2字节)容量的数据中写入“0”。
写入双字软元件的情况下，应使用 WriteDeviceRandom 或 WriteDeviceBlock2。
- 位软元件的位数指定及变址修饰不能使用。

5.3.23 GetDevice2(软元件数据的获取)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

以 2 字节数据获取 1 点软元件的数据。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.GetDevice2(szLabel, sData)
```

Integer	iRet	返回值	Output
String	szLabel	标签名	Input
Short	sData	读取的软元件值	Output

(b) Visual C++® .NET

```
iRet = object.GetDevice2(*szLabel, *lpsData)
```

int	iRet	返回值	Output
String*	szLabel	标签名	Input
short*	lpsData	读取的软元件值	Output

(c) Visual C#® .NET

```
iRet = object.GetDevice2(ref szLabel, ref sData)
```

int	iRet	返回值	Output
String	szLabel	标签名	Input
short	sData	读取的软元件值	Output

(4) 说明

- 将标签名 szLabel 中指定的 1 点软元件的数据存储到读取的软元件值 sData(lpsData) 中。

(5) 软元件指定方法

标签名及读取软元件值的指定方法如下所示。

- 标签名中可以指定下述数据类型。

类型分类	标签的数据类型	标签名的格式
基本型	位、字、双字 ^{*1} 、单精度实数 ^{*1} 、双精度实数 ^{*1} 、字符串 ^{*1} 、字符串 [Unicode] ^{*1} 、时间 ^{*1} 、定时器、长定时器、计数器、长计数器、累计定时器、长累计定时器	标签名

*1：仅读取起始软元件的低位 2 字节。

希望读取 2 字以上的软元件的情况下应使用 ReadDeviceBlock2 函数。

- 读取的软元件值按以下方式被存储。

<指定位软元件时>

(例) 读取 M0

- 标签设置

数据类型	软元件
位	M0

- 读取的软元件值

2字节
M0*2

*2: 仅将“M0”1点作为读取软元件对象，软元件值将被存储“0”或“1”。

<指定 FXCPU 的 CN200 以后时>

(例) 读取 CN200*3

- 标签设置

数据类型	软元件
双字	CN200

- 读取的软元件值

2字节
CN200 的 L

*3: 对于 FXCPU 的 CN200 以后，以 1 点读取指定软元件的 L(低位 2 字节)。

指定软元件的 H(高位 2 字节)不被读取。

<在标签中指定相当于 2 字以上的数据类型时>

(例) 指定字符串型标签进行读取

- 标签设置

数据类型	软元件
字符串	D0

- 读取的软元件值

2字节
D0*5

*5: 在起始软元件中仅读取低位 2 字节。

<指定字软元件时>

(例) 读取 D0

- 标签设置

数据类型	软元件
字	D0

- 读取的软元件值

2字节
D0

<指定 FD 软元件时(4 字软元件)>

(例) 指定 FD0 进行读取

- 标签设置

数据类型	软元件
字	FD0

- 读取的软元件值

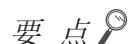
2字节

*4: 仅读取低位 2 字节。指定软元件的 HH、HL、LH(高位 6 字节)不被读取。

(6) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。(参见 496 页第 7 章 出错代码)



- 标签名所对应的软元件不存在的情况下将发生出错而无法读取。
- 对于读取的软元件值，应预留出 2 字节的存储器区域。没有存储器区域的情况下，有可能会发生应用程序出错等严重现象。
- 指定了双字软元件的情况下，在本函数中仅存储低位 1 字(2 字节)容量的数据。(不发生出错。)进行双字软元件的读取的情况下，应使用 ReadDeviceRandom 或 ReadDeviceBlock2。
- 位软元件的位数指定及变址修饰不能使用。

5.3.24 Connect(电话线路的连接)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行电话线路的连接。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.Connect()  
      Integer           IRet          返回值          Output
```

(b) Visual C++® .NET

```
iRet = object.Connect()  
      int            iRet          返回值          Output
```

(c) Visual C#® .NET

```
iRet = object.Connect()  
      int            iRet          返回值          Output
```

(4) 说明

- 根据调制解调器通信用控件的属性的设置值，进行电话线路的连接。
- 经由串行通信模块的情况下，根据 ActConnectWay 属性中设置的连接方式进行电话线路的连接。
通过 ActConnectWay 属性设置了自动（指定回调编号时）、回调连接（指定编号时）及回调请求（指定编号时）的情况下，如果未在 ActCallbackNumber 属性中设置编号将发生出错。

(5) 返回值

正常结束：返回 0。

非正常结束：返回 0 以外。( 496 页第 7 章 出错代码)

(6) 执行 Connect 时的注意事项

- 电话线路的连接必须在 Open 之前执行。
- 断开电话线路的情况下，应执行 Disconnect。在 Connect 中即使多次重复进行 Open 及 Close，电话线路也将保持为连接状态不变。
- 由于某种原因导致电话线路被断开的情况下，电话线路断开之前的状态为 Open 中时，必须执行 Close 之后再进行电话线路的重新连接。

(7) 使用了多个电话线路连接对象的情况下的注意事项

- 控件类型、端口编号、电话号码不相同的情况下

控件类型、端口编号、电话号码不相同的情况下，如果执行 Connect 的控件与最先执行了 Connect 的控件的端口编号、电话号码不相同，将发生出错（出错代码：0xF1000016）。

- 控件的端口编号、电话号码相同的情况下

控件类型、端口编号、电话号码相同的情况下，根据回调功能的连接方式其结束状态有所不同。

回调功能的连接方式与结束状态的关系如下表所示。

最先执行了 Connect 的控件 的连接方式	第 2 次以后执行了 Connect 时的连接方式			
	自动 自动（固定为回调时） 自动（指定回调编号时）	回调连接（固定时） 回调连接 (指定编号时)	回调请求（固定时） 回调请求 (指定编号时)	回调接收等待
自动 自动（固定为回调时） 自动（指定回调编号时）	○	○	×	×
回调连接（固定时） 回调连接（指定编号时）	○	○	×	×
回调请求（固定时） 回调请求（指定编号时）	○	○	○	○
回调接收等待	×	×	×	○

○：正常结束 ×：非正常结束（发生出错）

5.3.25 Disconnect(电话线路的断开)

(1) 可以使用的控件

在 DotUtlType 控件中可以使用。

(2) 功能

进行电话线路的断开。

(3) 格式

(a) Visual Basic® .NET

IRet = object.	Disconnect()		
Integer	IRet	返回值	Output

(b) Visual C++® .NET

iRet = object.	Disconnect()		
int	iRet	返回值	Output

(c) Visual C#® .NET

iRet = object.	Disconnect()		
int	iRet	返回值	Output

(4) 说明

通过 Connect 函数断开连接的电话线路。

(5) 返回值

正常结束 : 返回 0。

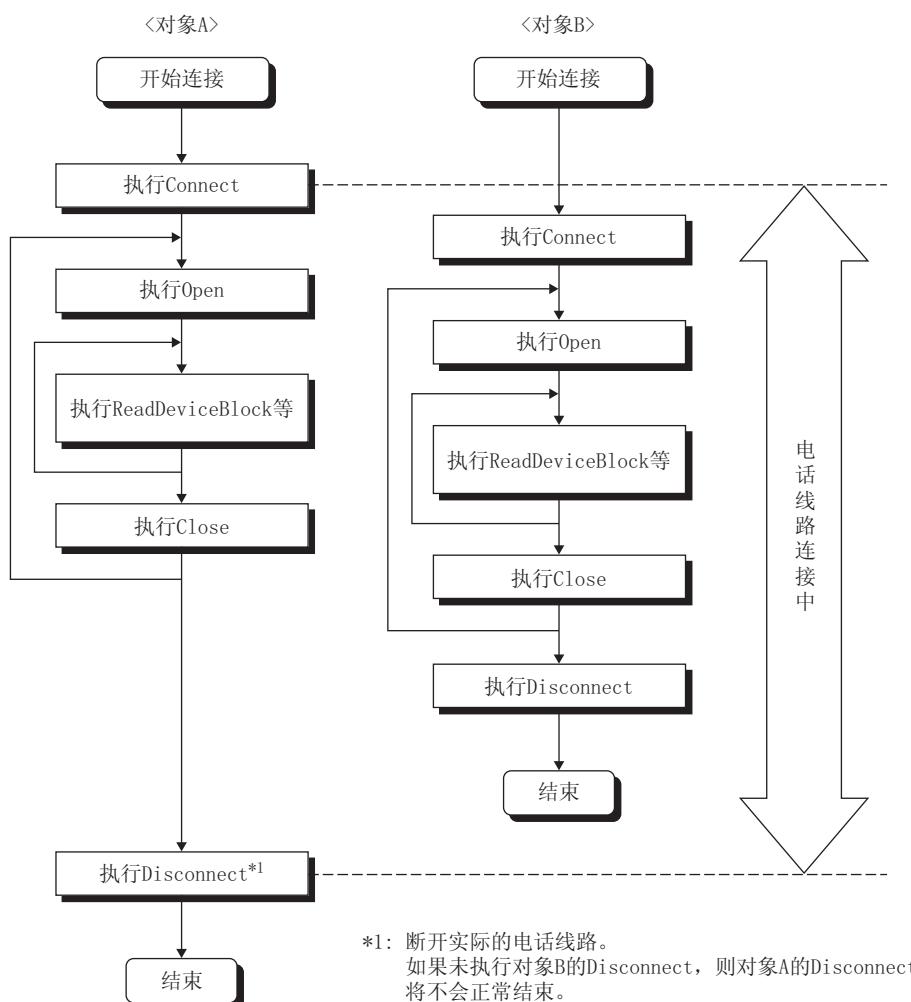
非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

(6) 执行 Disconnect 时的注意事项

- 指定 Disconnect 的情况下
Open 中的情况下，应执行 Close 之后再执行 Disconnect。
- 由于某种原因导致电话线路被断开的情况下，电话线路断开之前的状态为 Open 中时，必须执行 Close 之后再进行电话线路的重新连接。
- 使用了多个电话线路连接对象的情况下，最先执行了 Connect 的对象应在其它对象执行 Disconnect 后，再执行 Disconnect。
此外，使用了多个电话线路连接对象的情况下，如果最先执行了 Connect 的对象未执行 Disconnect，则电话线路不被断开。

同时使用了多个对象情况下的示例如下所示。

〈例〉同时使用2个控件的情况下
(也包括通过对象A、对象B不相同的应用程序执行动作的情况下)



*1: 断开实际的电话线路。
如果未执行对象B的Disconnect，则对象A的Disconnect将不会正常结束。

5.3.26 GetErrorMessage(出错信息的获取)

(1) 可以使用的控件

在 DotSupportMsg 控件中可以使用。

(2) 功能

获取对应于出错代码的出错内容及处理方法。

(3) 格式

(a) Visual Basic® .NET

```
iRet = object.GetErrorMessage(iErrorCode, szErrorMessage)
```

Integer	iRet	返回值	Output
Integer	iErrorCode	出错代码	Input
String	szErrorMessage	出错信息	Output

(b) Visual C++® .NET

```
iRet = object.GetErrorMessage(iErrorCode, **lpszErrorMessage)
```

int	iRet	返回值	Output
int	iErrorCode	出错代码	Input
String**	lpszErrorMessage	出错信息	Output

(c) Visual C#® .NET

```
iRet = object.GetErrorMessage(iErrorCode, ref szErrorMessage)
```

int	iRet	返回值	Output
int	iErrorCode	出错代码	Input
String	szErrorMessage	出错信息	Output

(4) 说明

- 对通过 iErrorCode (iErrorCode) 指定的出错代码的出错内容及处理方法进行读取。
- 读取的出错内容及处理方法将被存储到 szErrorMessage (lpszErrorMessage) 中。

(5) 返回值

正常结束 : 返回 0。

非正常结束 : 返回 0 以外。( 496 页第 7 章 出错代码)

第6章 样本程序

本章介绍安装 MX Component 时登录的样本程序有关内容。

(1) 关于样本程序、测试程序、样本顺控程序

(a) 样本程序、测试程序

作为创建用户程序时的参考，附加了样本程序。

此外，为了进行通信测试，附加了测试程序。

关于这些程序的使用，应由用户自担风险。

(b) 样本顺控程序

对于样本顺控程序，需要根据系统配置、参数设对其内容进行更改。

应修改为最适用于系统的内容。

此外，关于样本顺控程序的使用，应由用户自担风险。

(2) 样本程序、测试程序、样本顺控程序一览

安装 MX Component 时在 [用户指定文件夹] - [Act] - [Samples] 内登录的样本程序一览如下所示。

文件夹名		样本程序的详细内容	对应语言	参照
AccessVBA	Sample	ActUtilType 用样本程序	VBA (Access)	474 页 6.1.3 项
	TestPro	ActUtilType 用测试程序 *1		-
ExcelVBA	Sample	ActUtilType 用样本程序	VBA (Excel)	470 页 6.1.1 项
		ActUtilType、ActSupportMsg 用样本程序（软元件读取 / 写入）		472 页 6.1.2 项
	TestPro	ActUtilType、ActSupportMsg 用测试程序 *1		-
VBScript ^{*2}	SampleASP	ActMLUtilType 用样本程序	HTML (ASP 功能)	478 页 6.3 节
	SampleHTML			476 页 6.2 节
	TestPro	ActUtilType、ActSupportMsg 用测试程序 *1	HTML	-

*1：是动作确认用测试程序。应在进行动作确认的情况下使用。

*2：测试程序在以下的 Internet Explorer[®] 中运行。

- Internet Explorer[®] 8
- Internet Explorer[®] 9
- Internet Explorer[®] 10
- Internet Explorer[®] 11

文件夹名	样本程序的详细内容	对应语言	参照	
Vb. NET*3	ModemSample	Visual Basic®.NET	481 页 6. 4. 1 项	
	Sample			
	SampleDot		483 页 6. 4. 2 项	
	Sample_References			
	Sample_ReferencesDot			
	Sample_TypeConv		486 页 6. 4. 3 项	
Vc. NET	Sample2005	Visual C++®.NET (Visual Studio® 2005)	488 页 6. 5. 1 项	
	SampleDot2005			
	Sample2010	Visual C++®.NET (Visual Studio® 2010)		
	SampleDot2010			
	Sample_References2005	Visual C++®.NET (Visual Studio® 2005)		
	Sample_ReferencesDot2005			
	Sample_References2010	Visual C++®.NET (Visual Studio® 2010)		
	Sample_ReferencesDot2010			
	Sample_Support	Visual C++®.NET (Visual Studio® 2005)	489 页 6. 5. 2 项	
Vcs. NET*3	Sample	Visual C#®.NET (Visual Studio® 2005)	491 页 6. 6. 1 项	
	SampleDot			
	Sample_References			
	Sample_ReferencesDot			
Vc	SampleENG	Visual C++®.NET (MFC) (Visual Studio® 2005)	492 页 6. 7. 1 项	
	CustomSampleENG		494 页 6. 7. 2 项	
	Sample_SupportENG		495 页 6. 7. 3 项	

*3: 样本的运行环境为 Visual Studio® 2013 的情况下, 需要追加安装多字节字符编码 (MBCS) 用的 MFC 库 (DLL)。

文件夹名	样本程序的详细内容	对应语言	参照
GppW	CCG4A	顺控程序 (GX Developer)	MX Component Version 4 操作手册
	FXCPUtel		
	QJ71C24Callback		
	QJ71C24Callback_Number		
	QJ71C24TEL		

6.1 VBA 样本程序

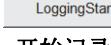
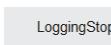
本节介绍 Excel、Access 用的 VBA 样本程序有关内容。

6.1.1 Excel 样本程序

本样本程序是使用 ActUtlType 控件进行可编程控制器 CPU 的软元件值的记录及图表显示的样本程序。

本样本程序通过 Excel 2003 创建。

(1) 使用方法

1. 打开样本表单。
2. 将通过通信设置实用程序进行了通信设置的逻辑站号输入到“LogicalStationNumber”的输入单元格内。
3. 将希望读取的软元件的起始软元件输入到“DeviceName”的输入单元格内。
4. 将进行记录的间隔输入到“LoggingTiming”的输入单元格内。
5. 点击  按钮时将开始进行记录。
此外，开始记录时在图表上将显示 10 个以前的软元件值及曲线图。
6. 点击  按钮时，记录将停止。
此外，画面上的记录数据不被清除。
7. 执行时发生了出错的情况下，在“Message”的输出单元格内将显示出错信息，在“Return Code”的输出单元格内将显示出错代码。发生了出错的情况下应参阅出错代码排除出错原因。
 496 页第 7 章 出错代码)

(2) 使用样本程序时的注意事项

- 在执行本样本程序之前应通过通信设置实用程序进行通信设置。
- 更改输入值的情况下，点击  按钮，停止一下记录并更改输入值，然后点击  按钮开始进行记录。
- 本样本程序中将进行记录的软元件点数设置为 10 点，将记录数设置为 10 个。

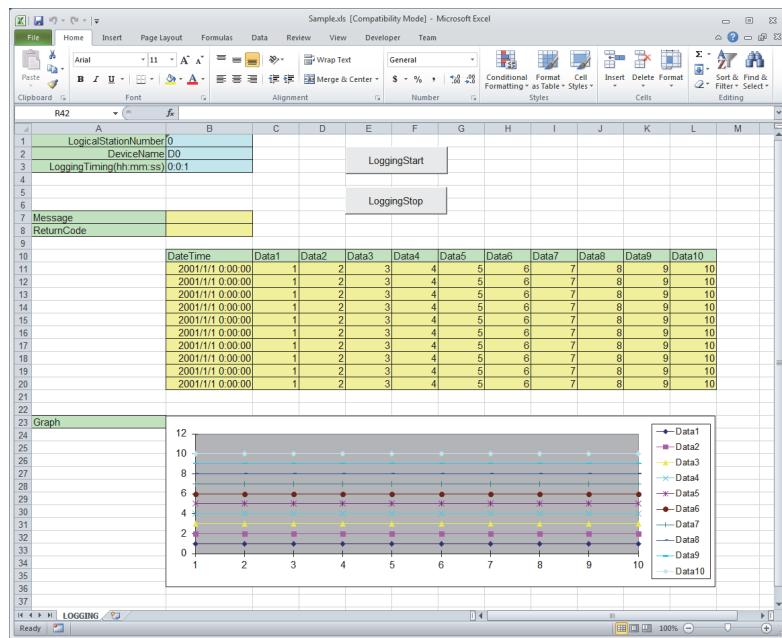
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [ExcelVBA] - [Sample]	Sample.xls

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容
LogicalStationNumber	输入通过通信设置实用程序进行了通信设置的逻辑站号。
DeviceName	输入进行读取的软元件的起始。
LoggingTiming (hh:mm:ss)	输入进行记录的间隔。 例：以 1 秒间隔进行记录的情况下 0:0:1 以 1 小时 30 分间隔进行记录的情况下 1:30:0
Message	显示函数的执行结果。(字符串)
ReturnCode	显示函数的执行结果。(数值 16 进制)
Time	显示记录的系统时间。
Data01 ~ 10	显示记录的软元件值。
Graph	显示 10 个以前记录的 10 点软元件值、曲线图。
LoggingStart	按钮 开始进行记录。
LoggingStop	按钮 停止记录。

6.1.2 Excel 样本程序（软元件读取 / 写入）

本样本程序是使用 ActUtilType 控件进行可编程控制器 CPU 的软元件 (D0 ~ D9) 的读取 / 写入的样本程序。

此外，发生了出错的情况下，使用 ActSupportMsg 控件在对话框中显示出错代码及出错信息。

本样本程序是通过 Excel 2003 创建的。

(1) 使用方法

1. 打开样本表单。
2. 将通过通信设置实用程序进行了通信设置的逻辑站号输入到“LogicalStationNumber”的输入单元格内。
3. 点击  按钮时将读取可编程控制器 CPU 的 D0 ~ D9 的软元件值并将其显示到 DeviceRead 区域中。
4. 此外，将值输入到 DeviceWrite 区域的 D0 ~ D9 中，点击  按钮时将值写入到可编程控制器 CPU 的 D0 ~ D9 中。
5. D0 ~ D9 的软元件读取 / 写入失败的情况下，在对话框中显示对应于出错代码的出错信息。

(2) 使用样本程序时的注意事项

在执行本样本程序之前应通过通信设置实用程序进行通信设置。

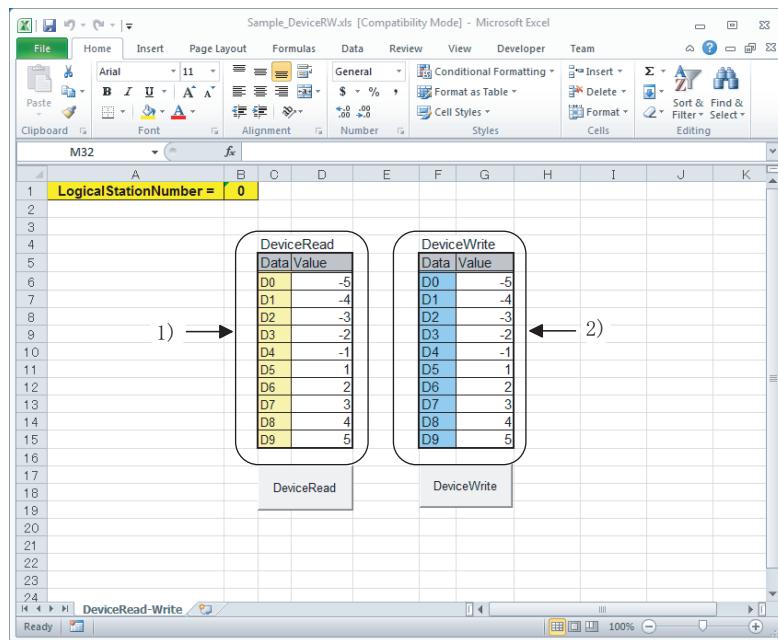
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [ExcelVBA] - [Sample]	Sample_DeviceRW.xls

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容
LogicalStationNumber	输入通过通信设置实用程序进行了通信设置的逻辑站号。
1) (DeviceRead 区域)	显示可编程控制器 CPU 的 D0 ~ D9 的软元件值。
2) (DeviceWrite 区域)	将写入值输入到可编程控制器 CPU 的 D0 ~ D9 中。
DeviceRead 按钮	读取可编程控制器 CPU 的 D0 ~ D9 的软元件值并将其显示到 DeviceRead 区域中。
DeviceWrite 按钮	将输入到 DeviceWrite 区域的 D0 ~ D9 中的软元件值写入到可编程控制器 CPU 的 D0 ~ D9 中。

6.1.3 Access 样本程序

本样本程序是使用 ActUtlType 控件进行可编程控制器 CPU 的软元件值的记录及监视的样本程序。
本样本程序是通过 Access 2003 创建的。

(1) 使用方法

1. 打开数据库。
2. 将通过通信设置实用程序进行了通信设置的逻辑站号输入到“LogicalStationNumber”的输入单元格内。
3. 将进行记录的间隔输入到“Logging Timing”的文本框内。
4. 点击 **LoggingStart** 按钮时将开始进行记录。
5. 点击 **LoggingStop** 按钮时，记录将停止。
此外，画面上的记录数据不被清除。
6. 执行时发生了出错的情况下，在信息框中将显示出错信息、出错代码。发生了出错的情况下应参阅出错代码排除出错原因。
( 496 页第 7 章 出错代码)

(2) 使用样本程序时的注意事项

- 在执行本样本程序之前应通过通信设置实用程序进行通信设置。
- 更改输入值的情况下，点击 **LoggingStop** 停止一下记录并更改输入值，然后点击 **LoggingStart** 按钮开始进行记录。
- 创建本样本程序时，将软元件“D0”～“D4”作为监视对象软元件，将“D10”～“D17”作为记录对象软元件。
此外，监视时间的规格为以 1 秒为间隔进行监视。
- 本样本程序最多可记录 100 个软元件值。
超过 100 个的情况下，删除最旧的记录数据，登录最新的记录数据。

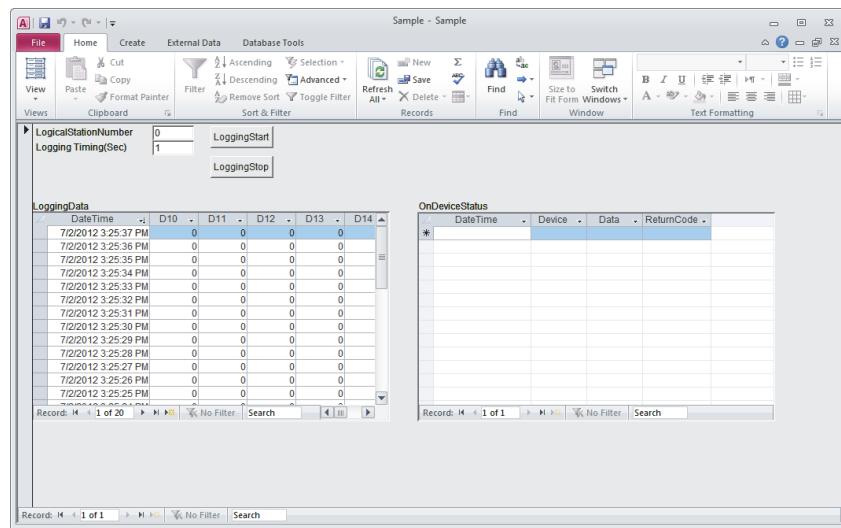
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [AccessVBA] - [Sample]	Sample.mdb

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容
LogicalStationNumber	输入通过通信设置实用程序进行了通信设置的逻辑站号。
LoggingTiming	输入进行记录的间隔。(单位为秒)
LoggingData	显示记录的数据。
OnDeviceStatus	显示监视对象软元件中条件成立的软元件。
LoggingStart 按钮	开始进行记录。
LoggingStop 按钮	停止记录。

6.2 VBScript 样本程序

本节介绍 VBScript 的样本程序有关内容。

本样本程序是使用 ActUtlType 控件，将槽罐的容量及状态设置为可编程控制器 CPU 的软元件值并对该值进行监视的样本程序。

本样本程序是通过 Microsoft® FrontPage® 2000 创建的。

(1) 使用方法

1. 打开样本文件时将打开至所使用的可编程控制器 CPU 的通信线路。
2. 此后，以 1 秒为间隔获取可编程控制器 CPU 的软元件值，使用该软元件值显示槽罐的容量及状态。
3. 执行时发生了出错的情况下，在信息框中将显示出错信息、出错代码。发生了出错的情况下应参阅出错代码排除出错原因。

( 496 页第 7 章 出错代码)

(2) 使用样本程序时的注意事项

- 执行本样本程序之前应通过通信设置实用程序进行逻辑站号“0”的通信设置。
- 在本样本程序中将软元件“D100”设置为槽罐的容量，将“D101”设置为槽罐的状态。

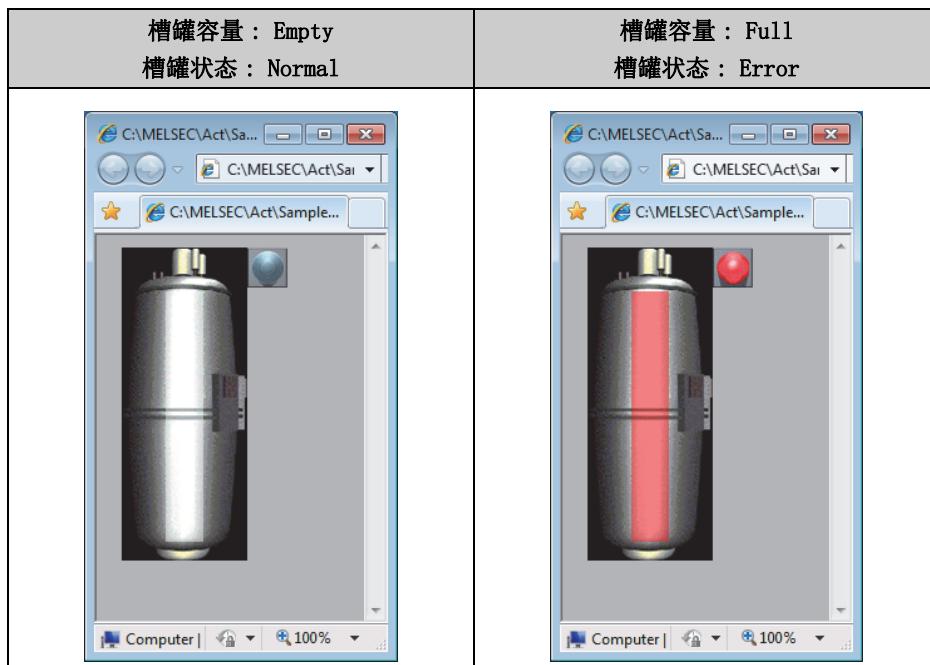
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [VBScript] - [SampleHTML]	Sample.html

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容	备注
Tank capacity (槽罐容量)	显示槽罐的容量。 ($0 \leq$ 软元件值 ≤ 200 的范围内)	与软元件 “D100” 链接。
Tank status (槽罐状态)	显示槽罐的状态。 Normal (软元件值 = 0): 亮蓝色灯 Error (软元件值 $\neq 0$): 亮红色灯	与软元件 “D101” 链接。

6.3 ASP 样本程序

本样本程序是使用 ActUtlType 控件，进行可编程控制器 CPU 的 CPU 读取及软元件值的读取的样本程序。

(1) 文件构成

- Sample.asp (数据输入画面)
是初始显示画面，是用于用户对监视数据进行设置的画面。
- SampleControl.asp (数据获取画面)
获取 Sample.asp 的输入数据，对存储及输入至全局变量中的数据进行出错检查。
- SampleMon.asp (数据显示画面)
根据 Sample.asp 的输入数据进行数据显示。此外，发生了出错的情况下，显示出错内容。

(2) 使用方法

1. 将 Sample.asp、SampleControl.asp、SampleMon.asp 存储到 WWW 服务器上的同一文件夹中。
2. 在 Microsoft Internet Explorer[®] 中，浏览 Sample.asp 的 URL，显示初始画面。
3. 输入初始画面 “MonitorTiming”、“LogicalStationNumber”、“DeviceName”、“DeviceSize”，按压 **MonitorStart** 按钮时，将进行 Open 处理、ReadDeviceBlock 处理、GetCpuType 处理、Close 处理。
4. 在数据显示画面 “MonitorTiming:”、“LogicalStationNumber:”、“DeviceName:” “DeviceSize:” 中，将显示输入的数据。
5. “Message:” 在各处理中发生了出错时显示发生了出错的处理。
6. “Return Code:” 显示各处理的结果。
7. “CpuType:” 显示连接的可编程控制器 CPU 的型名。
8. “ReadData(Hex)” 显示连接的可编程控制器 CPU 的软元件数据。
9. 发生了出错的情况下应参阅出错代码排除出错原因。
( 496 页第 7 章 出错代码)
10. 此外，数据显示画面将以 “MonitorTiming:” 的间隔反复更新。
11. 按压 **Back** 按钮时，将返回至初始画面。

(3) 使用样本程序时的注意事项

- 在执行本样本程序之前，应通过通信设置实用程序进行通信设置。

(4) 样本文件一览

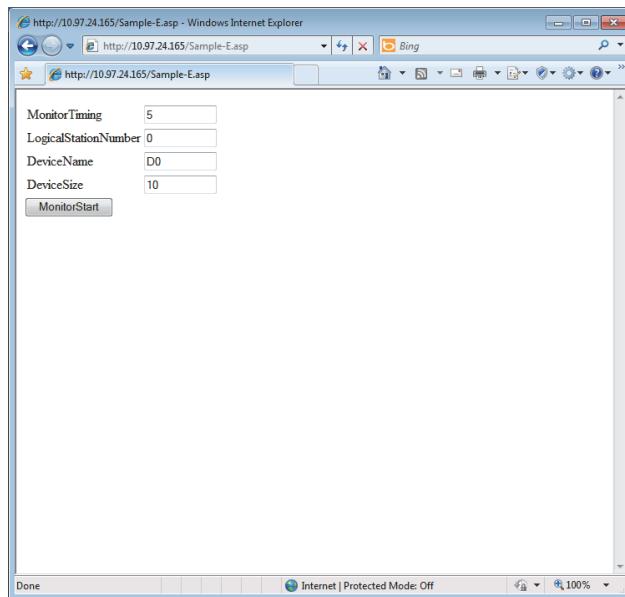
通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [VBScript] - [SampleASP]	Sample-E.asp
	SampleControl-E.asp
	SampleMon-E.asp

(5) 画面

以下介绍样本程序的画面有关内容。

(a) 数据输入画面 (Sample-E.asp)

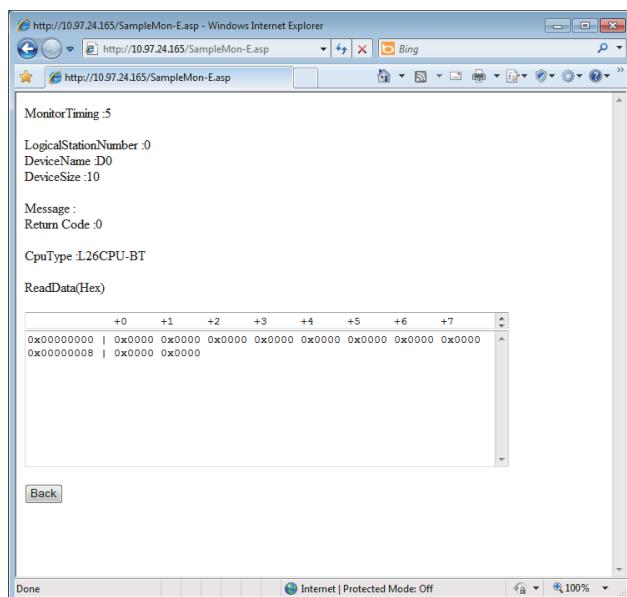


项目	内容
MonitorTiming	输入监视间隔。
LogicalStationNumber	输入逻辑站号。
DeviceName	输入进行读取的软元件名。
DeviceSize	输入进行读取的软元件的点数。
<input type="button" value="MonitorStart"/>	按钮 开始监视处理。

(b) 数据获取画面 (SampleControl-E.asp)

数据获取画面是用于将数据输入画面中输入的各数据存储至全局变量并进行出错检查的画面，数据获取画面不被显示到 Internet Explorer[®] 的画面上。

(c) 数据显示画面 (SampleMon-E.asp)



项目	内容
MonitorTiming:	显示监视间隔。
LogicalStationNumber:	显示逻辑站号。
DeviceName:	显示进行读取的软元件名。
DeviceSize:	显示进行读取的软元件的点数。
Message:	发生出错时显示其内容。
Return Code:	显示各方法的执行结果。
CpuType:	显示 CPU 型号。
ReadData (Hex)	显示读取的软元件值。
<input type="button" value="Back"/> 按钮	返回至初始画面。

6.4 Visual Basic® .NET 样本程序

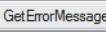
6.4.1 调制解调器通信用样本程序

本样本程序是使用 ActUtilType 控件，对指定的逻辑站号进行可编程控制器 CPU 的软元件监视的样本程序。

此外，发生了出错的情况下，使用 ActSupportMsg 控件，在对话框中显示出错代码及出错信息。

本样本程序是通过 Visual Basic® .NET (Visual Studio® 2005) 创建的。

(1) 使用方法

1. 将通过通信设置应用程序进行了调制解调器通信设置的逻辑站号的值输入到“LogicalStationNumber”文本框内。
2. 连接的模块中设置了口令的情况下，在“Password”文本框内输入口令。
3. 将进行监视的间隔输入到“MonitorInterval”文本框内。
4. 将进行监视的软元件名输入到“DeviceName”文本框中，将软元件点数输入到“Size”文本框内。
5. 点击  按钮时，在连接电话线路后将进行通信线路的打开，以 ReadDeviceBlock 函数中指定的间隔进行软元件值读取。（在无需连接电话线路的逻辑站号的情况下，不进行电话线路的连接，只进行通信线路的打开。）
6. 点击  按钮时，关闭通信线路后断开电话线路。
7. 执行本样本程序时发生了出错的情况下，在“ReturnValue”文本框内将显示出错代码。“ReturnValue”文本框内显示出错代码的情况下，如果点击  按钮，将显示出错代码所对应的出错内容及处理方法。
发生了出错的情况下应参阅出错代码排除出错原因。
 496 页第 7 章 出错代码)

(2) 使用样本程序时的注意事项

- 为了使用 ActUtilType 控件，在执行本样本程序之前应通过通信设置应用程序进行逻辑站号的设置。
- 对逻辑站号、监视间隔、软元件名及读取点数进行更改的情况下，应点击  按钮关闭通信线路，然后点击  按钮重新启动通信。

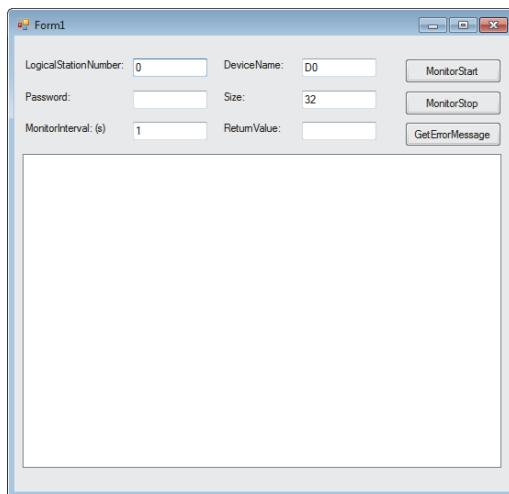
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vb.NET] - [ModemSample]	ModemSample.sln

(4) 画面

以下介绍样本程序的画面有关内容。



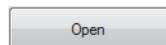
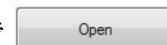
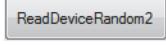
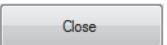
项目	内容
LogicalStationNumber	输入通过通信设置实用程序设置的逻辑站号。
Password	需要口令的情况下，输入口令。
MonitorInterval	设置监视间隔。（单位：s）
DeviceName	输入进行监视的软元件名。
Size	输入读取点数。
ReturnValue	显示执行的方法的返回值。
MonitorStart 按钮	连接电话线路后，打开通信线路开始进行监视。
MonitorStop 按钮	关闭通信线路后，断开电话线路结束监视。
GetErrorMessage 按钮	获取并显示“ReturnValue”的出错代码相关的出错内容及处理方法。

6.4.2 Read/Write 样本程序

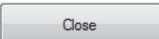
本样本程序是使用 ActUtlType 控件、DotUtlType 控件、ActProgType 控件的其中一个，进行可编程控制器 CPU 的软元件值的读取 / 写入及软元件的状态监视的样本程序。

本样本程序是通过 Visual Basic® .NET (Visual Studio® 2005) 创建的。

(1) 使用方法

1. 加载窗体，选择使用的控件。
2. ActUtlType 控件的情况下，在点击  按钮之前，将通过通信设置实用程序进行了通信设置的逻辑编号输入到“LogicalStationNumber”文本框中。
3. 点击  按钮，打开通信线路。
4. 进行随机读取时，将希望读取的软元件名及点数输入到“Random Read/Write”框内的“DeviceName”、“DeviceSize”文本框中后，点击  按钮时，“Output”框内的“Data”文本框中将显示软元件数据。
5. 进行随机写入时，将希望写入的软元件名及点数输入到“Random Read/Write”框内的“DeviceName”、“DeviceSize”文本框中，将写入的软元件值输入到同一框内的“DeviceData”文本框中后，点击  按钮时，软元件值将被写入到可编程控制器 CPU 中。
6. 进行批量读取时，将希望读取的软元件名及点数输入到“Block Read/Write”框内的“DeviceName”、“DeviceSize”文本框中后，如果点击  按钮，“Output”框内的“Data”文本框中将显示软元件数据。
7. 进行批量写入时，将希望写入的软元件名及点数输入到“Block Read/Write”框内的“DeviceName”、“DeviceSize”文本框中，将写入的软元件值输入到同一框内的“DeviceData”文本框中后，如果点击  按钮，则软元件值将被写入到可编程控制器 CPU 中。
8. 进行软元件的状态监视登录时，将作为事件发生条件的软元件名、点数、软元件值及状态监视间隔输入到“Status Entry/Free”框内的“DeviceName”、“DeviceSize”、“DeviceData”、“MonitorCycle”文本框中后，如果点击  按钮将执行登录。
登录的事件的发生条件成立时，“Output”框内的“Data”文本框中将显示事件数据。
9. 对已登录的软元件的状态监视进行解除时，点击  按钮时将执行解除。
10. 执行时发生了出错的情况下，在“Output”框内“ReturnCode”文本框内将显示出错代码。
发生了出错的情况下，应参照出错代码排除出错原因。
( 496 页第 7 章 出错代码)
11. 点击  按钮时，通信线路将关闭。

(2) 使用样本程序时的注意事项

- 使用 ActUtlType 控件的情况下，在执行本样本程序之前应通过通信设置实用程序进行通信设置。
- 使用 ActProgType 控件的情况下，在本样本程序的规格中将可编程控制器 CPU 设置为“Q02(H)”，将 COM 端口设置为“COM1”，将传送速度设置为固定使用 19200bps。进行更改的情况下，需要对相应属性值进行更改。
- 更改使用的控件的情况下，点击  按钮关闭通信线路之后更改控件，然后点击  按钮重新打开线路。

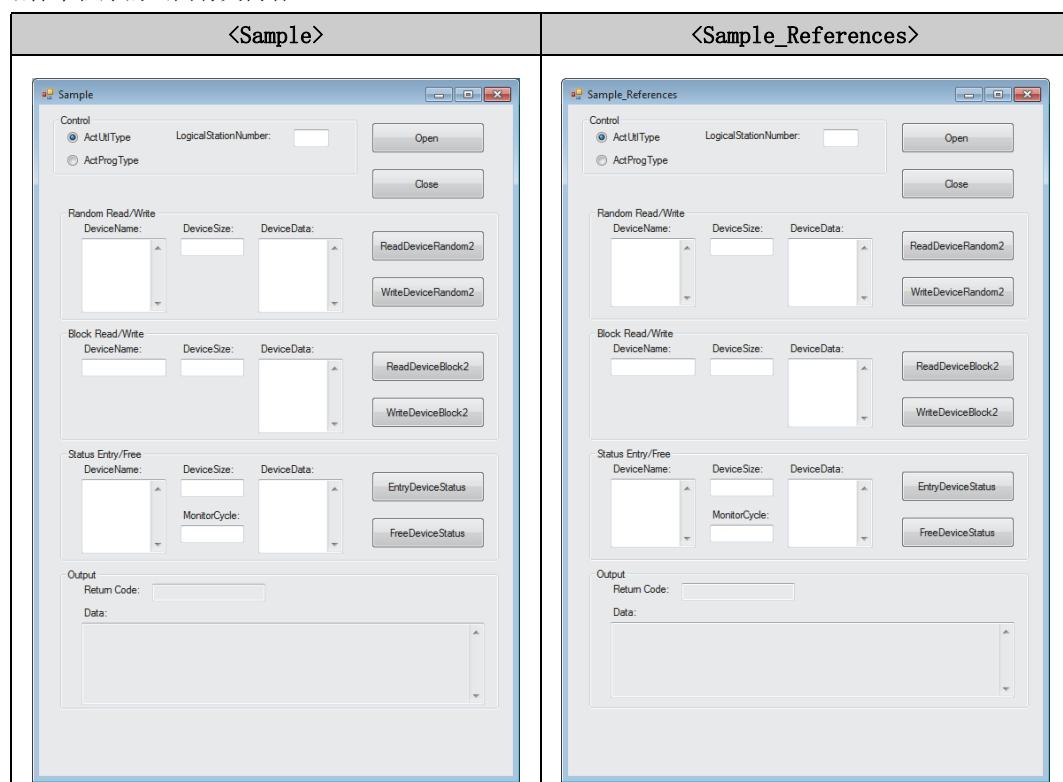
(3) 样本文件一览

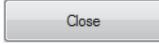
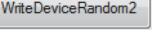
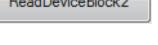
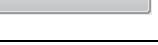
通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vb.NET] - [Sample]	Sample.sln
[用户指定文件夹] - [Act] - [Samples] - [Vb.NET] - [Sample_References]	Sample_References.sln
[用户指定文件夹] - [Act] - [Samples] - [Vb.NET] - [SampleDot]	Sample.sln
[用户指定文件夹] - [Act] - [Samples] - [Vb.NET] - [Sample_ReferencesDot]	Sample_References.sln

(4) 画面

以下介绍样本程序的画面有关内容。



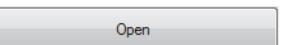
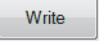
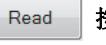
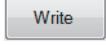
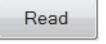
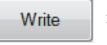
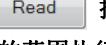
项目	内容
“Control” 框	ActUtlType、ActProgType、DotUtlType 中之一
	LogicalStationNumber
 按钮	打开通信线路。
	 按钮 关闭通信线路。
“Random Read/Write” 框	DeviceName
	DeviceSize
	DeviceData
	 按钮 以同一框内的“DeviceName”、“DeviceSize”文本框中输入的数据为基础，从可编程控制器 CPU 中对软元件数据进行随机读取并显示到“Output”框的“Data”文本框中。
	 按钮 以同一框内的“DeviceName”、“DeviceSize”、“DeviceData”文本框中输入的数据为基础，将软元件数据随机写入到可编程控制器 CPU 中。
	DeviceName
“Block Read/Write” 框	DeviceSize
	DeviceData
	 按钮 以同一框内的“DeviceName”、“DeviceSize”文本框中输入的数据为基础，从可编程控制器 CPU 中对软元件数据进行块读取并显示到“Output”框的“Data”文本框中。
	 按钮 以同一框内的“DeviceName”、“DeviceSize”、“DeviceData”文本框中输入的数据为基础，将软元件数据进行块写入到可编程控制器 CPU 中。
	DeviceName
	DeviceSize
“Status Entry/Free” 框	MonitorCycle
	DeviceData
	 按钮 以同一框内的“DeviceName”、“DeviceSize”、“MonitorCycle”、“DeviceData”文本框中输入的数据为基础，执行 ON 软元件状态 • 事件的登录。登录的事件的发生条件成立时，在“Output”框的“Data”文本框中显示事件数据。
	 按钮 将已登录的 ON 软元件状态 • 事件删除。
	Return Code
	Data
“Output” 框	显示各方法的执行结果。 显示读取的软元件值。

6.4.3 型转换样本程序

本样本程序的作用是，使用 ActUtilType 控件，将 ASCII 字符串、32bit 整数、实数的各数据以各自的数据形式读取 / 写入到可编程控制器 CPU 的软元件中，执行时发生了出错的情况下，使用 ActSupportMsg 控件显示出错信息。

本样本程序是通过 Visual Basic® .NET (Visual Studio® 2005) 创建的。

(1) 使用方法

1. 加载窗体，将通过通信设置实用程序进行了通信设置的逻辑编号输入到“逻辑站号”文本框内后，点击  按钮打开通信线路。
2. 进行 ASCII 字符的写入时，在位于“ASCII 字符”框内的上段的文本框中输入 ASCII 字符数据后，如果点击位于同一框内的  按钮，将对可编程控制器 CPU 执行写入。（进行写入时，与输入的字符数的不足与否无关，将以 D0 ~ D9 的范围执行写入。字符数不足时，用 0(Hex) 填充。）
3. 进行 ASCII 字符的读取时，点击“ASCII 字符”框内的  按钮。在位于同一框内的下段的文本框中将显示来自于可编程控制器 CPU 的的数据。（以 D0 ~ D9 的范围执行读取。）
4. 进行 32bit 整数的写入时，在位于“32bit 整数”框内上段的文本框中输入 32bit 整数数据后，如果点击位于同一框内的  按钮，将对可编程控制器 CPU 执行写入。（以 D10 ~ D11 的范围执行写入。）
5. 进行 32bit 整数的读取时，点击“32bit 整数”框内的  按钮。位于同一框内下段的文本框中将显示来自于可编程控制器 CPU 的数据。（以 D10 ~ D11 的范围执行读取。）
6. 进行实数的写入时，在位于“实数”框内上段的文本框中输入实数数据后，如果点击位于同一框内的  按钮，将对可编程控制器 CPU 执行写入。（以 D12 ~ D13 的范围执行写入。）
7. 进行实数的读取时，点击“实数”框内的  按钮。位于同一框内的下段的文本框中将显示来自于可编程控制器 CPU 的数据。（以 D12 ~ D13 的范围执行读取。）
8. 发生了出错的情况下应参阅出错代码排除出错原因。
 496 页第 7 章 出错代码
9. 点击  按钮时，通信线路将关闭。

(2) 使用样本程序时的注意事项

- 使用 ActUtilType 控件的情况下，在执行本样本程序之前应通过通信设置实用程序进行通信设置。
- 由于非 ActUtilType 控件原因发生了出错的情况下（32bit 整数 / 实数中输入了字符串或超出允许范围的数值等），将在信息框内显示相应出错后结束程序。

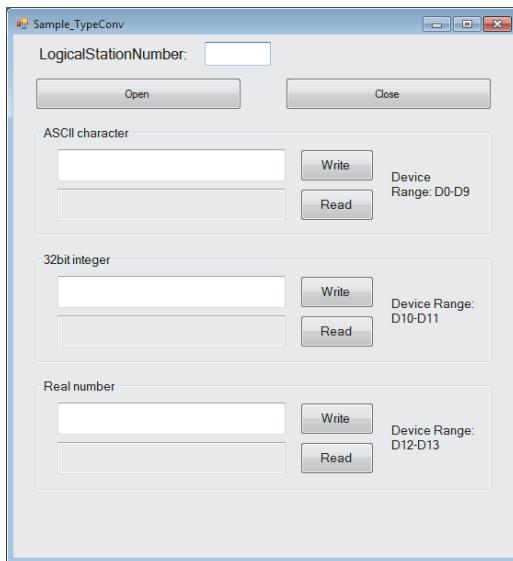
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vb. NET] - [Sample_TypeConv]	Sample_TypeConv.sln

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容	
LogicalStationNumber (逻辑站号)	输入通过通信设置实用程序进行了通信设置的逻辑站号。	
<input type="button" value="Open"/>	按钮	打开通信线路。
<input type="button" value="Close"/>	按钮	关闭通信线路。
“ASII character (ASCII 字符)” 框	<input type="button" value="Write"/>	将框内上段的文本框中输入的 ASCII 字符数据写入到可编程控制器 CPU 中。
	<input type="button" value="Read"/>	将从可编程控制器 CPU 中读取的 ASCII 字符数据显示到框内下段的文本框中。
“32bit integer (32bit 整数)” 框	<input type="button" value="Write"/>	将框内上段的文本框中输入的 32bit 整数数据写入到可编程控制器 CPU 中。
	<input type="button" value="Read"/>	将从可编程控制器 CPU 中读取的 32bit 整数数据显示到框内下段的文本框中。
“Real number (实数)” 框	<input type="button" value="Write"/>	将框内上段的文本框中输入的实数数据写入到可编程控制器 CPU 中。
	<input type="button" value="Read"/>	将从可编程控制器 CPU 中读取的实数数据显示到框内下段的文本框中。

6.5 Visual C++® .NET 样本程序

6.5.1 Read/Write 样本程序

本样本程序是使用 ActUtlType 控件、DotUtlType 控件、ActProgType 控件之一，进行可编程控制器 CPU 的软元件值的读取 / 写入及软元件的状态监视的样本程序。

本样本程序是通过 Visual C++® .NET(Visual Studio® 2005) 及 Visual C++® .NET(Visual Studio® 2010) 创建的。

(1) 使用方法

与使用 Visual Basic® .NET(Visual Studio® 2005) 时相同。

(☞ 483 页 6.4.2 项 (1) 使用方法)

(2) 使用样本程序时的注意事项

与使用 Visual Basic® .NET(Visual Studio® 2005) 时相同。

(☞ 484 页 6.4.2 项 (2) 使用样本程序时的注意事项)

(3) 样本文件一览

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

(a) Visual C++® .NET(Visual Studio® 2005) 用

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [Sample2005]	Sample2005.sln
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [Sample_References2005]	Sample_References2005.sln
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [SampleDot2005]	Sample2005.sln
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [Sample_ReferencesDot2005]	Sample_References2005.sln

(b) Visual C++® .NET (Visual Studio® 2010) 用

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [Sample2010]	Sample2010.sln
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [Sample_References2010]	Sample_References2010.sln
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [SampleDot2010]	Sample2010.sln
[用户指定文件夹] - [Act] - [Samples] - [Vc.NET] - [Sample_ReferencesDot2010]	Sample_References2010.sln

(4) 画面

与使用 Visual Basic® .NET(Visual Studio® 2005) 时相同。

(☞ 484 页 6.4.2 项 (4) 画面)

6.5.2 故障排除功能样本程序

本样本程序是使用 DotSupportMsg 控件执行故障排除功能的样本程序。

(1) 使用方法

1. 加载窗体。
2. 在“ErrorCode”文本框中输入希望了解其出错内容的出错代码。
3. 点击 **GetErrorMessage** 按钮时，出错内容显示用文本框中将显示出错内容及处理方法。
显示的出错内容及处理方法与 496 页 7.1 节 各控件返回的出错代码中记载的内容相同。
4. 点击 **Exit** 按钮时样本程序将结束。

(2) 使用样本程序时的注意事项

对于“ErrorCode”文本框中输入的出错代码，应输入 496 页 7.1 节 各控件返回的出错代码 (0x*****)
或以 10 进制数进行输入。

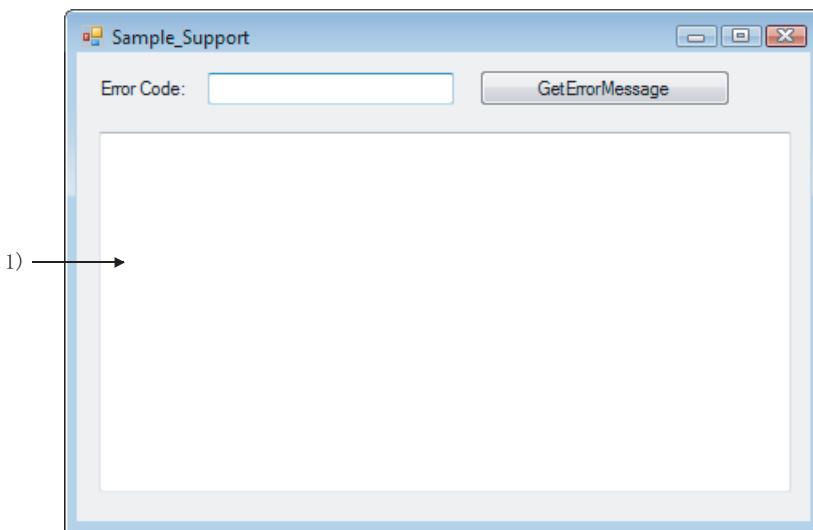
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vc. Net] - [Sample_Support]	Sample_Support.sln

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容
ErrorCode	输入出错代码。
GetErrorMessage 按钮	对“ErrorCode”文本框内输入的出错代码的出错内容及处理方法进行读取。
Exit 按钮	结束样本程序。
1) (出错内容显示用文本框)	显示“ErrorCode”文本框内输入的出错代码的出错内容及处理方法。

6.6 Visual C#® .NET 样本程序

6.6.1 Read/Write 样本程序

本样本程序是使用 ActUtilType 控件、DotUtilType 控件、ActProgType 控件之一，进行可编程控制器 CPU 的软元件值的读取 / 写入及软元件的状态监视的样本程序。

本样本程序是通过 Visual C#® .NET (Visual Studio® 2005) 创建的。

(1) 使用方法

与使用 Visual Basic® .NET (Visual Studio® 2005) 时相同。( 483 页 6.4.2 项 (1) 使用方法)

(2) 使用样本程序时的注意事项

与使用 Visual Basic® .NET (Visual Studio® 2005) 时相同。( 484 页 6.4.2 项 (2) 使用样本程序时的注意事项)

(3) 样本文件一览

通过默认路径进行安装时，样本程序被安装在下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vcs.NET] - [Sample]	Sample.sln
[用户指定文件夹] - [Act] - [Samples] - [Vcs.NET] - [Sample_References]	Sample_References.sln
[用户指定文件夹] - [Act] - [Samples] - [Vcs.NET] - [SampleDot]	Sample.sln
[用户指定文件夹] - [Act] - [Samples] - [Vcs.NET] - [Sample_ReferencesDot]	Sample_References.sln

(4) 画面

与使用 Visual Basic® .NET (Visual Studio® 2005) 时相同。( 484 页 6.4.2 项 (4) 画面)

6.7 Visual C++® .NET (MFC) 样本程序

在本节中，介绍使用调度 I/F 及自定义 I/F 创建的 Visual C++® .NET (MFC) 用的样本程序有关内容。

6.7.1 调度 I/F

本样本程序是在调度 I/F 中使用 ActUtlType 控件或 ActProgType 控件进行连接目标 CPU 的型号读取及软元件值的读取 / 写入的样本程序。

(1) 使用方法

1. 加载窗体，选择使用的控件。
2. 点击 **Open Communication** 按钮时，将通过以太网通信执行通信线路 Open。
3. 点击 **GetCpuType** 按钮时，在“Output Data”文本框（上段）中将显示当前线路连接的可编程控制器 CPU 的型号代码，在“Output Data”文本框（下段）中将显示 CPU 型号。
4. 将希望读取的软元件输入到“软元件”文本框内后，点击 **GetDevice** 按钮时在“Output Data”文本框（上段）中将显示软元件数据。
5. 将希望写入的软元件输入到“软元件”文本框内，将写入的软元件值输入到“软元件值”文本框内后，点击 **SetDevice** 按钮时将进行软元件值的写入。
6. 点击 **Close Communication** 按钮时将进行通信线路的 Close。
7. 执行函数时发生了出错的情况下，“Return Value”文本框内将显示出错代码。发生了出错的情况下应参阅出错代码排除出错原因。
 496 页第 7 章 出错代码)

(2) 使用样本程序时的注意事项

- 使用 ActUtlType 控件的情况下，在执行样本程序之前应通过通信设置实用程序将以太网通信的信息设置到逻辑站号“1”中。
- 更改使用的控件的情况下，点击 **Close Communication** 按钮对通信线路进行 Close 之后，更改控件然后重新对线路进行 Open。

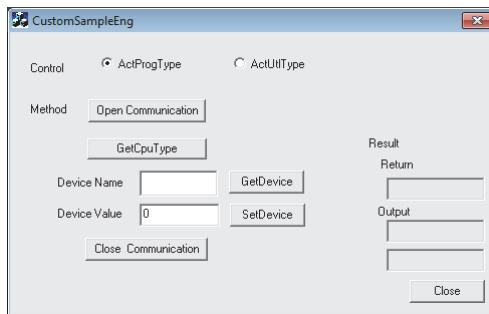
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vc] - [SampleENG]	sampleENG.sln

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容	
Control(控件)	选择使用的控件。	
Open Communication 按钮	对通信线路进行 Open。	
GetCpuType 按钮	执行可编程控制器 CPU 型号读取。	
Device Name(软元件)	输入进行读取 / 写入的软元件。	
Device Value(软元件值)	输入进行写入的软元件的软元件值。	
Close Communication 按钮	对通信线路进行 Close。	
GetDevice 按钮	执行 “软元件” 文本框内输入的软元件的数据读取。	
SetDevice 按钮	执行 “软元件” 文本框内输入的软元件的数据写入。	
Return Value	显示函数的执行结果。	
Output Data	Upper (上段)	显示 CPU 型号代码及读取的软元件值。
	Lower (下段)	显示 CPU 型号。

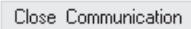
6.7.2 自定义 I/F

本样本程序是通过自定义 I/F 使用 ActUtlType 控件或 ActProgType 控件进行连接目标 CPU 的型号读取及软元件值的读取 / 写入的样本程序。

(1) 使用方法

使用方法与调度 I/F 用样本程序相同。( 492 页 6.7.1 项 (1) 使用方法)

(2) 使用样本程序时的注意事项

- 使用 ActUtlType 控件的情况下，在执行本样本程序之前应通过通信设置实用程序将以太网通信的信息设置到逻辑站号“2”中。
- 更改使用的控件的情况下，点击  按钮对通信线路进行 Close 之后，更改控件并重新对线路进行 Open。

(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vc] - [CustomSampleENG]	CustomSampleENG.sln

(4) 画面

画面与调度 I/F 用样本程序的相同。( 493 页 6.7.1 项 (4) 画面)

6.7.3 故障排除功能样本程序

本样本程序是使用 ActSupportMsg 控件执行故障排除功能的样本程序。

(1) 使用方法

1. 加载窗体。
2. 在 “ErrorCode” 文本框中输入希望了解其出错内容的出错代码。
3. 点击 **GetErrorMessage** 按钮时在出错内容显示用文本框中将显示出错内容及处理方法。显示的出错内容及处理方法与 496 页 7.1 节 各控件返回的出错代码记载的内容相同。
4. 点击 **Exit** 按钮时样本程序将结束。

(2) 使用样本程序时的注意事项

对于 “ErrorCode” 文本框中输入的出错代码，应输入 496 页 7.1 节 各控件返回的出错代码中记载的出错代码 (0x******) 或以 10 进制数进行输入。

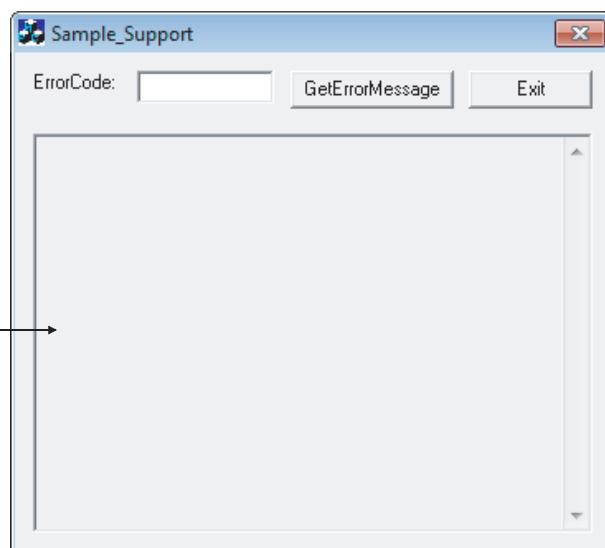
(3) 样本文件

通过默认路径进行了安装时，样本程序被安装到下述文件夹中。

文件夹名称	文件名称
[用户指定文件夹] - [Act] - [Samples] - [Vc] - [Sample_SupportENG]	Sample_SupportENG.sln

(4) 画面

以下介绍样本程序的画面有关内容。



项目	内容
ErrorCode	输入出错代码。
GetErrorMessage 按钮	对 “ErrorCode” 文本框内输入的出错代码的出错内容及处理方法进行读取。
Exit 按钮	结束样本程序。
1) (出错内容显示用文本框)	显示 “ErrorCode” 文本框内输入的出错代码的出错内容及处理方法。

第7章 出错代码

本章介绍各控件返回的出错代码及 CPU、模块、网卡返回的出错代码有关内容。

7.1 各控件返回的出错代码

各控件返回的出错代码如下所示。

出错代码	出错内容	处理方法
0x00000000	正常结束	-
100	ENQ 重试次数超过	
101		• 采取防干扰措施。
102	NACK 重试次数超过	
103	长度异常的报文	
104	接收超时	
105	未检测到 DSR 信号	• 确认电缆的连接。
106	连接被切断	• 确认电缆的连接。 • 执行 Open。
107	发送超时	• 确认电缆的连接。
108	顺控编号错误	• 采取防干扰措施。
200	未发现附属 DLL	• 重新安装。
0x01010002	超时出错	• 对属性的超时值重新进行审核。 • 通过通信设置实用程序重新进行设置。 • 对可编程控制器 CPU、模块的设置及电缆状态等系统重新进行审核。 • 进行 Close 后，重新进行 Open。 • 应结束程序后，重新启动计算机。
0x01010005	报文出错	• 确认系统的噪声。 • 对属性的超时值重新进行审核。 • 通过通信设置实用程序重新进行设置。 • 对可编程控制器 CPU、模块的设置及电缆状态等系统重新进行审核。 • 应进行 Close 后，重新进行 Open。 • 应结束程序后，重新启动计算机。
0x01010010	PC No. 出错 在指定站号中无法进行通信。	• 对通过通信设置实用程序设置的站号进行检查。 • 对 ActStationNumber 中设置的站号进行确认。
0x01010011	模式出错 是不支持的指令。	• 确认是否设置了正确的 CPU 类型。 • 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x01010012	特殊模块指定出错	• 对特殊模块的指定地址进行检查。
0x01010013	其它数据出错 由于某种原因无法通信。	• 确认系统配置是否是不支持的配置。 • 确认是否设置了正确的 CPU 类型。 • 结束程序，重新启动计算机。 • 请向当地三菱电机代理商咨询。

出错代码	出错内容	处理方法
0x01010018	远程请求出错 正在以与通信中的路径不同的路径执行远程操作。	<ul style="list-style-type: none"> 将正在通过其它路径执行的远程操作解除。
0x01010020	链接出错 未能进行链接通信。	<ul style="list-style-type: none"> 确认是否对通信对象或站（主站）、通信路径中的站进行了复位操作。 确认网络参数的设置是否正确进行。
0x01010021	特殊模块总线出错 对象特殊模块无响应。	<ul style="list-style-type: none"> 对对象特殊模块进行修理或更换。 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01800001	无指令出错 是不支持的方法。	<ul style="list-style-type: none"> 不支持相应方法，因此请勿使用。
0x01800002	存储器锁定出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 使用计算机用接口板的情况下，应将计算机的最小工作集区域扩大。^{*1}
0x01800003	存储器预留出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 结束其它程序，预留出空余的存储器。 使用计算机用接口板的情况下，应将计算机的最小工作集区域扩大。^{*1}
0x01800004	DLL 加载出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 结束其它程序，预留出空余的存储器。 重新安装 MX Component。
0x01800005	资源预留出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 结束其它程序，预留出空余的存储器。
0x01801001	资源超时出错 未能在允许时间内获取资源。	<ul style="list-style-type: none"> 待其它对象通信完毕后，重新执行操作。 延长超时时间，重新执行操作。 结束程序，重新启动计算机。 重新安装 MX Component。
0x01801002	多重线路打开出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x01801003	未执行打开	
0x01801004	打开类型出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01801005	指定端口出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。
0x01801006	指定模块出错	<ul style="list-style-type: none"> 确认使用的系统配置与通信设置实用程序的设置内容或设置的属性值是否一致。 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01801007	指定 CPU 出错	<ul style="list-style-type: none"> 确认 ActCpuType 中设置的 CPU 类型。 确认是否是系统配置不支持的配置。 结束程序，重新启动计算机。 重新安装 MX Component。 确认 ActPacketType 中设置的数据包类型。

*1：关于增大计算机的最小工作集区域的方法，请参阅计算机接口板的以下手册。

 MELSEC 数据链接库参考手册

出错代码	出错内容	处理方法
0x01801008	对象站访问出错	<ul style="list-style-type: none"> 重新审核对象站。
0x01801009	注册表打开失败 注册表的数据键的打开处理失败。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x0180100A	数据包类型出错 指定的数据包类型有错误。	<ul style="list-style-type: none"> 重新审核 ActPacketType。 结束程序，重新启动计算机。 重新安装 MX Component。
0x0180100B	协议类型出错 指定的协议有错误。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x0180100C	注册表查找失败	
0x0180100D	GetProcAddress 失败	
0x0180100E	DLL 未加载出错	
0x0180100F	正在执行其它 Object 在排他控制中，无法执行方法。	<ul style="list-style-type: none"> 请在一定时间后重新执行。
0x01802001	软元件出错 方法中指定的软元件字符串是不正确的软元件字符串。	<ul style="list-style-type: none"> 重新审核软元件名。
0x01802002	软元件编号出错 方法中指定的软元件字符串的编号是不正确的软元件编号。	<ul style="list-style-type: none"> 重新审核软元件编号。
0x01802003	程序类型出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01802004	和校验出错 接收数据的和校验值有异常。	<ul style="list-style-type: none"> 确认模块侧的和校验设置。 确认控件的和校验用属性。 确认电缆。 结束程序，重新启动计算机。 重新安装 MX Component。
0x01802005	容量出错 方法中指定的点数不正确。	<ul style="list-style-type: none"> 确认方法中指定的点数。 重新审核可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 结束程序，重新启动计算机。 重新安装 MX Component。
0x01802006	块 No. 出错 方法中指定的软元件字符串的块指定编号不正确。	<ul style="list-style-type: none"> 对方法中指定的软元件字符串的块指定编号进行重新审核。
0x01802007	接收数据出错 接收的数据是异常的数据。	<ul style="list-style-type: none"> 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 确认电缆。 结束程序，重新启动计算机。
0x01802008	写保护出错	
0x01802009	参数读取出错	
0x0180200A	参数写入出错	
0x0180200B	可编程控制器类型不一致 属性中设置的 CPU 类型及通信设置实用程序中设置的 CPU 类型与通信对象 CPU 的类型不一致。	<ul style="list-style-type: none"> 在属性的 CPU 类型中设置正确的 CPU 类型。 通过通信设置实用程序设置正确的 CPU 类型。 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。

出错代码	出错内容	处理方法
0x0180200C	请求取消出错 请求在途中被取消。	
0x0180200D	驱动器名出错 指定的驱动器名有错误。	
0x0180200E	起始步出错 指定的起始步编号有错误。	
0x0180200F	参数类型出错 参数类型有错误。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01802010	文件名出错 文件名有错误。	
0x01802011	状态出错 登录 / 解除 / 设置状态有错误。	
0x01802012	详细条件区分出错	
0x01802013	步条件出错	
0x01802014	位软元件条件出错	
0x01802015	参数设置出错	
0x01802016	站号指定出错 方法不支持对指定站号的操作。	<ul style="list-style-type: none"> 重新审核站号。 确认是否执行了不支持的方法。 重新审核 CPU、模块等系统配置。
0x01802017	关键字出错	
0x01802018	读写标志出错	
0x01802019	刷新方式出错	
0x0180201A	缓冲访问方式出错	
0x0180201B	启动模式 / 停止模式出错	
0x0180201C	写入时钟数据出错 写入指定的时钟数据中有错误，因此无法正常写入。	<ul style="list-style-type: none"> 重新审核写入的时钟数据。
0x0180201D	时钟数据运行中写入出错 时钟数据的写入失败。 可编程控制器 CPU 为运行中状态，因此无法写入时钟数据。	<ul style="list-style-type: none"> 将可编程控制器 CPU 置为 STOP 状态。
0x0180201E	ROM 运行出错	
0x0180201F	跟踪中出错 执行了跟踪中无法执行的操作。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01802020	起始 I/O 编号出错 方法中指定的起始 I/O 编号是不正确的值。	<ul style="list-style-type: none"> 确认方法中指定的起始 I/O 编号的值。 通过 GPP 功能确认可编程控制器的参数 (I/O 分配)。 结束程序，重新启动计算机。
0x01802021	起始地址出错 方法中指定的缓冲地址是不正确的值。	<ul style="list-style-type: none"> 确认方法中指定的缓冲地址的值。 结束程序，重新启动计算机。
0x01802022	模式出错	
0x01802023	SFC 块 No. 出错	
0x01802024	SFC 步 No. 出错	
0x01802025	步 No. 出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x01802026	数据出错	<ul style="list-style-type: none"> 重新安装 MX Component。
0x01802027	系统数据出错	<ul style="list-style-type: none"> 请向当地三菱电机代理商咨询。
0x01802028	TC 设置值个数出错	
0x01802029	清除模式出错	
0x0180202A	信号留出错	

出错代码	出错内容	处理方法
0x0180202B	版本管理出错	
0x0180202C	监视未登录出错	
0x0180202D	PI 类型出错	
0x0180202E	PI No. 出错	
0x0180202F	PI 个数出错	
0x01802030	移位出错	
0x01802031	文件类型出错	
0x01802032	指定模块出错	
0x01802033	出错检查标志出错	
0x01802034	步 RUN 运行出错	
0x01802035	步 RUN 数据出错	
0x01802036	步运行中出错	
0x01802037	对 E ² ROM 的程序运行中写入出错	
0x01802038	时钟数据读取 / 写入出错 对不具有时钟单元的可编程控制器 CPU 执行了时钟数据读取 / 写入方法。	• 请勿执行时钟数据读取 / 写入。
0x01802039	跟踪未完成出错	
0x0180203A	登录清除标志出错	
0x0180203B	操作出错	
0x0180203C	站数出错	
0x0180203D	循环次数指定出错	
0x0180203E	获取数据选择出错	
0x0180203F	SFC 循环数出错	
0x01802040	运动 CPU 出错	
0x01802041	运动 CPU 通信出错	
0x01802042	定时执行时间设置出错	
0x01802043	功能数出错	
0x01802044	系统信息指定出错	
0x01802045	登录条件未成立出错	
0x01802046	功能 No. 出错	
0x01802047	RAM 运行出错	
0x01802048	引导源 ROM 运行出错	
0x01802049	引导源传送模式指定出错	
0x0180204A	存储器不足出错	
0x0180204B	备份驱动器 ROM 出错	
0x0180204C	块容量出错	
0x0180204D	运行中拆装出错	
0x0180204E	模块已登录出错	
0x0180204F	口令登录数据已满出错	
0x01802050	口令未登录出错	
0x01802051	远程口令出错	
0x01802052	IP 地址出错	
0x01802053	超出超时值范围出错	
0x01802054	指令搜索未检测出错误	
0x01802055	跟踪执行类型出错	
0x01802056	版本出错	

出错代码	出错内容	处理方法
0x01802057	热备电缆出错 热备电缆异常。 可编程控制器 CPU 的状态异常。	• 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。
0x0180205C	关键字保护出错 可编程控制器 CPU 由于关键字保护处于禁止访问状态。	• 解除关键字之后重新执行访问。
0x0180205D	关键字解除失败 指定的关键字有错误。	• 指定正确的关键字。
0x0180205E	关键字保护失败 可编程控制器 CPU 未受理指令。	• 重新执行操作，或重新接通可编程控制器 CPU 的电源。
0x0180205F	关键字登录失败 指定的关键字中包含有不能使用的字符。	• 指定正确的关键字。
0x01802060	关键字删除失败 指定的关键字有错误。	
0x01802062	接收数据包 CRC 检查出错 对接收数据包的数据进行 CRC 检查中发生了出错。	• 重新执行通信处理。
0x01802063	接收数据包 CRC 检查出错 对接收数据包的整个数据文件进行的 CRC 检查中发生了出错。	
0x01802064	FX 可编程控制器连接出错	• 请向当地三菱电机代理商咨询。
0x01802070	运行中写入程序出错 运行中写入对象程序在可编程控制器 CPU 中不存在。	• 将可编程控制器 CPU 置为 STOP 状态之后再执行写入。
0x01802071	Ether 直接连接通信多个响应接收出错 Ether 直接连接通信时接收了多个响应。	
0x01802072	Ether 直接连接通信时通信出错 Ether 直接连接通信时从其它计算机对可编程控制器 CPU 进行了通信，因此无法通信。	• 确认计算机与可编程控制器 CPU 是否为 1 对 1 连接。
0x01802073	可编程控制器 CPU 查找响应出错 可编程控制器 CPU 查找中存在有超出可查找最大数的响应。	• 将网络上存在的可编程控制器 CPU 设置为 1024 个以下。
0x01802074	冗余系统其它系统连接时诊断出错	• 将电缆改换插入到当前未连接的可编程控制器 CPU 上。 • 将冗余 CPU 指定设置为本系统。
0x01808001	重复打开出错 在 Open 状态下执行了 Open 方法。	• 结束程序，重新启动计算机。 • 执行除 Open 以外的方法。
0x01808002	通道编号指定出错 设置了属性的端口编号及通信设置实用程序中设置的端口编号是不正确的值。	• 在属性的端口编号中设置正确的值。 • 通过通信设置实用程序重新进行通信设置。
0x01808003	驱动程序未启动 网卡的驱动程序未启动。	• 启动驱动程序。
0x01808004	重叠事件生成出错	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x01808005	MUTEX 生成出错 用于进行排他控制的 MUTEX 的创建失败。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。

出错代码	出错内容	处理方法
0x01808006	Socket 对象生成出错 Socket 对象未能创建。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x01808007	Socket 生成出错 Socket 的生成失败。	<ul style="list-style-type: none"> • 确认是否启动了使用同一个端口编号的应用程序。 • 更改属性的端口编号值后进行重试。 • 通过通信设置实用程序更改端口编号值后进行重试。 • 应通过操作系统的控制面板进行以太网卡及协议设置。 • 结束程序，重新启动计算机。
0x01808008	端口连接出错 连接的建立失败。 对象无响应。	<ul style="list-style-type: none"> • 对属性的 IP 地址及端口编号的值进行重新审核。 • 通过通信设置实用程序对端口编号的值进行重新审核。 • 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 • 结束程序，重新启动计算机。
0x01808009	COM 端口句柄出错 无法获取 COM 端口的句柄。 无法复制 COM 端口对象。 无法复制 SOCKET 对象。	<ul style="list-style-type: none"> • 确认有无正在使用 COM 端口的应用程序。 • 结束程序，重新启动计算机。
0x0180800A	缓冲容量设置出错 COM 端口的缓冲容量的设置失败。	<ul style="list-style-type: none"> • 确认有无正在使用 COM 端口的应用程序。 • 通过操作系统的控制面板进行 COM 端口设置。 • 结束程序，重新启动计算机。
0x0180800B	DCB 值获取出错 COM 端口的 DCB 值的获取失败。	
0x0180800C	DCB 设置出错 COM 端口的 DCB 值的设置失败。	
0x0180800D	超时值设置出错 COM 端口的超时值的设置失败。	<ul style="list-style-type: none"> • 对属性的超时值进行重新审核。 • 通过通信设置实用程序对超时的值进行重新审核。 • 确认有无正在使用 COM 端口的应用程序。 • 通过操作系统的控制面板进行 COM 端口设置。 • 结束程序，重新启动计算机。
0x0180800E	共享存储器打开出错 共享存储器的打开处理失败。	<ul style="list-style-type: none"> • 确认是否启动了 GX Simulator。 • 结束程序，重新启动计算机。
0x01808101	重复关闭出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。
0x01808102	句柄关闭出错 COM 端口的句柄的关闭失败。	
0x01808103	驱动程序关闭出错 驱动程序的句柄的关闭失败	
0x01808104	重叠事件关闭出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x01808105	Mutex 句柄关闭出错	
0x01808106	COM 端口句柄关闭出错	

出错代码	出错内容	处理方法
0x01808201	发送出错 数据发送失败。	<ul style="list-style-type: none"> 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 通过操作系统的控制面板进行 COM 端口设置。 通过控制面板进行以太网卡及协议的设置。 对方法进行重试。 结束程序，重新启动计算机。
0x01808202	发送数据容量出错 数据发送失败。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x01808203	队列清除出错 COM 端口队列的清除失败。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 进行 Close 后，重新进行 Open。
0x01808301	接收出错 数据接收失败。	<ul style="list-style-type: none"> 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 对属性的超时值进行重新审核。 通过通信设置实用程序对超时值进行重新审核。 对方法进行重试。 结束程序，重新启动计算机。
0x01808302	未发送出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01808303	重叠事件获取出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01808304	接收缓冲容量不足 接收数据是大于系统中准备的接收缓冲容量的数据。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x01808401	控件出错 COM 端口的通信控制的更改失败。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x01808402	信号线控制出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01808403	信号线指定出错 COM 端口的通信控制的更改失败。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x01808404	未执行打开	<ul style="list-style-type: none"> 执行打开。 结束程序，重新启动计算机。
0x01808405	通信参数出错 属性的数据位及停止位的组合不正确。	<ul style="list-style-type: none"> 对属性的数据位及停止位的值进行重新审核。 通过通信设置实用程序重新进行设置。
0x01808406	传送速度值指定出错 属性的传送速度不正确。	<ul style="list-style-type: none"> 对属性的传送速度的值进行重新审核。 通过通信设置实用程序重新进行设置。
0x01808407	数据长度出错 属性的数据位的值不正确。	<ul style="list-style-type: none"> 对属性的数据位的值进行重新审核。 通过通信设置实用程序重新进行设置。
0x01808408	奇偶指定出错 属性的奇偶值不正确。	<ul style="list-style-type: none"> 对属性的奇偶值进行重新审核。 通过通信设置实用程序重新进行设置。
0x01808409	停止位指定出错 属性的停止位的值不正确。	<ul style="list-style-type: none"> 对属性的停止位的值进行重新审核。 通过通信设置实用程序重新进行设置。
0x0180840A	通信控制设置出错 属性的控件的值不正确。	<ul style="list-style-type: none"> 对属性的控件的值进行重新审核。 通过通信设置实用程序重新进行设置。

出错代码	出错内容	处理方法
0x0180840B	超时出错 经过了超时时间但却未能接收数据。	<ul style="list-style-type: none"> 对属性的超时值进行重新审核。 通过通信设置实用程序重新进行设置。 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 对方法进行重试。 进行 Close 后，重新进行 Open。 结束程序，重新启动计算机。
0x0180840C	连接出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x0180840D	重复连接出错	
0x0180840E	连接失败 套接字对象的连接失败。	
0x0180840F	信号线状态获取失败 COM 端口的信号线状态的获取失败。	
0x01808410	CD 信号线 OFF 通信对象的 CD 信号处于 OFF 状态。	<ul style="list-style-type: none"> 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 结束程序，重新启动计算机。
0x01808411	口令不一致出错	<ul style="list-style-type: none"> 确认属性的远程口令。
0x01808412	TEL 通信出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01808501	USB 驱动程序加载出错 USB 驱动程序的加载失败。	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 确认 USB 驱动程序的安装。
0x01808502	USB 驱动程序连接出错 USB 驱动程序的连接失败。	
0x01808503	USB 驱动程序发送出错 数据的发送失败。	<ul style="list-style-type: none"> 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 通过操作系统的控制面板（设备管理器）进行 USB 设置。 对方法进行重试。 结束程序，重新启动计算机。
0x01808504	USB 驱动程序接收出错 数据的接收失败。	
0x01808505	USB 驱动程序超时出错	<ul style="list-style-type: none"> 对超时时间进行重新审核。 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0x01808506	USB 驱动程序初始化出错 USB 驱动程序的初始化失败。	<ul style="list-style-type: none"> 通过操作系统的控制面板（设备管理器）进行 USB 设置。 结束程序，重新启动计算机。
0x01808507	USB 其它出错 发生了数据发送接收相关出错。	<ul style="list-style-type: none"> 拔下电缆后，重新进行连接。 结束程序，重新启动计算机。 重新安装 MX Component。
0x01809000	GX Works2 未安装出错 GX Simulator2 的安装路径的查找中发生了出错。	<ul style="list-style-type: none"> 安装 GX Works2。
0x01809001	GX Simulator2 未启动出错 GX Simulator2 未启动。	<ul style="list-style-type: none"> 开始 GX Simulator2。

出错代码	出错内容	处理方法
0x01809002	GX Simulator2 启动出错	
0x01809003	GX Simulator2 启动超时出错	
0x01809004	GX Simulator2 停止出错	
0x01809005	GX Simulator2 启动出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x01809007	GX Simulator2 停止出错	
0x01809008	GX Simulator2 启动出错 已达到可同时启动的模拟数的上限，因此未能启动。	
0x01809009	GX Simulator2 启动出错 只能启动 1 个的工程的模拟处于已启动状态。	
0x01809010	GX Simulator2 启动信息不正确出错 未能预留出用于获取 GX Simulator2 启动信息的存储器区域，因此发生了出错。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x01809021	GX Simulator2 启动出错 已达到可同时启动的模拟数的上限，因此未能启动。	
0x01809022	GX Simulator2 启动出错 FXCPU 工程的模拟已开始，因此其它 CPU 的模拟无法开始。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x02000001	点数溢出出错 监视服务器中登录的点数过多。	<ul style="list-style-type: none"> • 减少监视登录的点数。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x02000002	共享存储器创建出错 共享存储器的创建失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x02000003	共享存储器访问出错	
0x02000004	存储器预留出错 监视服务器的存储器预留失败。	<ul style="list-style-type: none"> • 结束其它应用程序。 • 增加系统的存储器。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x02000005	软元件未登录出错 未进行监视登录。	<ul style="list-style-type: none"> • 对监视服务器进行监视登录。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x02000006	监视服务器启动出错 监视服务器未启动。	<ul style="list-style-type: none"> • 启动监视服务器。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x02000010	软元件值未获取出错 监视尚未完成。	<ul style="list-style-type: none"> • 经过一定时间后，再次执行获取。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。

出错代码	出错内容	处理方法
0x03000001	指令不支持 不支持指令。	
0x03000002	存储器锁定出错 存储器的锁定失败。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03000003	存储器预留出错 存储器的预留失败。	
0x03000004	DLL 读取出错 DLL 的读取失败。	
0x03000005	资源预留出错 资源的预留失败。	
0x03010001	文件创建出错 文件的创建失败。	• 确认 HD 的可用空间。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03010002	文件打开出错 文件打开失败。	
0x03010003	缓冲容量出错 指定的缓冲的容量不正确 / 不充分。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03010004	SIL 语法出错 SIL 语法有错误。	
0x03010005	文件名出错 指定的文件名过长。	• 缩短文件名。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03010006	无文件出错 指定的文件不存在。	• 确认文件名。 • 确认有无文件。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03010007	文件结构出错 指定的文件内的数据结构有错误。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03010008	有文件出错 指定的文件已存在。	• 确认文件名。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03010009	无文件出错 指定的文件不存在。	• 确认文件名。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x0301000A	文件删除出错 指定的文件无法删除。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x0301000B	重复打开出错 对指定工程进行了重复打开。	• 确认文件名。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x0301000C	文件名出错 指定的文件名有错误。	

出错代码	出错内容	处理方法
0x0301000D	文件读取出错 文件的读取失败。	
0x0301000E	文件写入出错 文件的写入失败。	
0x0301000F	文件搜索出错 文件搜索失败。	
0x03010010	文件关闭出错 文件关闭失败。	
0x03010011	文件夹生成出错 文件夹的创建失败。	
0x03010012	文件复制出错 文件复制失败。	
0x03010013	工程路径出错 工程的路径长度有错误。	
0x03010014	工程类型出错 工程类型有错误。	
0x03010015	文件类型出错 文件类型有错误。	
0x03010016	子文件类型出错 子文件类型有错误。	
0x03010017	磁盘容量不足出错 磁盘的容量不足。	
0x03020002	重复打开出错 对 DBProduct 进行了重复打开。	
0x03020003	未打开出错 DBProduct 未能打开。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03020004	Extract 出错 DBProduct 未能 Extract。	
0x03020010	参数出错 DBProduct 的参数有错误。	
0x03020011	语言出错 语言参数有错误。	
0x03020012	生产厂商指定出错 生产厂商参数有错误。	
0x03020013	模块指定出错 模块参数有错误。	
0x03020014	SQL 参数出错 DBProduct 的 SIL、SQL 参数有错误。	
0x03020015	SIL 语法出错 DBProduct 的 SIL 语法有错误。	
0x03020016	字段密钥输入出错 输入的字段密钥有错误。	
0x03020050	记录数据语法出错 DBProduct 的记录数据重构失败。	
0x03020060	记录数据获取出错 DBProduct 的记录数据获取失败。	
0x03020061	最终记录出错 由于是最终记录，因此无法获取下一个记录。	
0x03FF0000	初始化出错	

出错代码	出错内容	处理方法
0x03FF0001	未初始化出错	
0x03FF0002	重复初始化出错	
0x03FF0003	工作区初始化出错	
0x03FF0004	数据库初始化出错	
0x03FF0005	记录集初始化出错	
0x03FF0006	数据库关闭出错	
0x03FF0007	记录集关闭出错	
0x03FF0008	数据库未打开出错 数据库未打开。	
0x03FF0009	记录集未打开出错 记录集未打开。	
0x03FF000A	表格初始化出错 TtableInformation 表格的初始化失败。	
0x03FF000B	表格初始化出错 TfieldInformation 表格的初始化失败。	
0x03FF000C	表格初始化出错 TrelationInformation 表格的初始化失败。	
0x03FF000D	表格初始化出错 Tlanguage 表格的初始化失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x03FF000E	表格初始化出错 Tmaker 表格的初始化失败。	
0x03FF000F	表格初始化出错 TopenDatabase 表格的初始化失败。	
0x03FF0010	现场值出错	
0x03FF0011	现场值出错	
0x03FF0012	结束出错 DB 的结束处理失败。	
0x03FF0100	记录移动出错 记录的移动失败。	
0x03FF0101	记录计数获取出错 记录计数的获取失败。	
0x03FF0110	现场值获取出错 现场值的获取失败。	
0x03FF0111	现场值设置出错 现场值的设置失败。	
0x03FFFFFF	其它出错	
0x04000001	无指令出错 在指定的 CPU 类型中无法处理。	<ul style="list-style-type: none"> • 确认 ActCpuType 中设置的 CPU 类型。 • 确认是否为系统配置不支持的配置。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。
0x04000002	存储器锁定出错 存储器的锁定失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x04000003	存储器预留出错 存储器的预留失败。	
0x04000004	内部服务器 DLL 加载出错 内部服务器的启动失败。	<ul style="list-style-type: none"> • 确认 MX Component 的安装文件是否被删除或移动。 • 结束程序，重新启动计算机。 • 重新安装 MX Component。

出错代码	出错内容	处理方法
0x04000005	资源预留出错 未能预留必要的资源。	
0x04000006	主对象加载出错 文件的读取失败。	
0x04000007	转换表格加载出错 表格数据的读取失败。	
0x04000100	中间代码容量不正确出错	
0x04010001	中间代码无转换出错 转换的1个指令的机器代码超过了256个字节。	
0x04010002	中间代码结束出错 转换源的中间代码区域中途结束。	
0x04010003	中间代码不足出错 转换源的中间代码不足。	
0x04010004	中间代码数据出错 转换源的中间代码不正确。存在有无法转换的代码。	
0x04010005	中间代码构成出错 中间代码的步数不正确。	
0x04010006	步数出错 注释中间代码的步数不正确。	
0x04010007	机器代码存储区域不足出错 机器代码存储区域不足。	
0x04010008	其它出错 (中间代码→机器代码转换中发生的其它出错)	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装MX Component。 • 请向当地三菱电机代理商咨询。
0x04011001	机器代码无转换出错 转换的1个指令的中间代码超过了256个字节。	
0x04011002	机器代码结束出错 转换源的机器代码区域中途结束。	
0x04011003	机器代码异常转换源的机器代码异常而无法转换。	
0x04011004	中间代码存储区域不足出错 中间代码存储区域不足。	
0x04011005	其它出错 (机器代码→中间代码转换中发生的其它出错)	
0x04020001	文本代码无转换出错 转换的1个指令的中间代码超过了256个字节。	
0x04020002	无输入出错 输入的列表代码不足。	
0x04020003	指令出错 转换源的列表代码的指令名不正确。	
0x04020004	软元件出错 转换源的列表代码的软元件名不正确。	
0x04020005	软元件No.出错 转换源的列表代码的软元件No.超出了允许范围。	
0x04020006	转换出错 转换源的列表代码是无法识别的代码。	

出错代码	出错内容	处理方法
0x04020007	文本数据出错 转换源的列表代码不正确。	
0x04020008	SFC 动作输出出错 SFC 的动作输出指令不正确。	
0x04020009	SFC 转移条件出错 SFC 的转移条件指令不正确。	
0x0402000A	行间声明出错 行间声明输入不正确。	
0x0402000B	P. I 声明出错 P. I 声明的输入不正确。	
0x0402000C	注解出错 注解的输入不正确。	
0x0402000D	注释出错 注释的输入不正确。	
0x0402000E	其它出错 (列表→中间代码转换中发生的其它出错)	
0x04021001	中间代码无转换出错 转换的 1 个指令的列表代码超过了 256 个字节。	
0x04021002	中间代码区域溢出出错 转换源的中间存储区域已结束。	
0x04021003	指令出错 转换源的中间代码的指令不正确。	
0x04021004	软元件出错 转换源中间代码的软元件不正确。	
0x04021005	中间代码出错 转换源的中间代码构成不正确。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x04021006	列表存储区域不足出错 存储转换的列表代码的区域不足。	
0x04021007	其它出错 (中间代码→列表转换中发生的其它出错)	
0x04030001	无转换出错 转换的中间代码存储区域不足。	
0x04030002	电路创建不良出错 字符存储器的电路作为顺控程序未完成。	
0x04030003	指定电路容量溢出 指定电路过大。	
0x04030004	回送电路不正确出错 回送电路的源与目标的匹配性不符。 回送电路的设置过多。	
0x04030005	其它出错 (字符存储器→中间代码转换中发生的其它出错)	
0x04031001	无转换出错 指定的字符存储器的纵・横容量不正确。	
0x04031002	指令代码异常出错 转换源的指令中间代码不正确。	
0x04031003	电路创建不良出错 无法转换为顺控程序电路。无 END 指令。	

出错代码	出错内容	处理方法
0x04031004	指定电路容量溢出出错 指定电路过大。	
0x04031005	严重出错 发生了严重出错。	
0x04031006	存储块数不足出错 没有相当于转换的字符存储器电路的块数容量的存储区域。	
0x04031007	电路块搜索出错 电路块的途中数据被断开。	
0x04031008	其它出错 (中间代码→字符存储器转换中发生的其它出错)	
0x04040001	没有 CAD 数据出错转换源 CAD 数据。 CAD 数据的格式有错误。	
0x04040002	输出数据出错 输入 CAD 数据类型与输出数据类型不匹配。	
0x04040003	库加载出错 库的加载失败。	
0x04040004	存储区域预留出错 转换后数据的存储区域不足。	
0x04040005	无 END 指令出错 转换源 CAD 数据中没有 END 指令。	
0x04040006	指令代码异常 转换源 CAD 数据中存在有指令代码异常。	
0x04040007	软元件 No. 出错 软元件 No. 超出了允许范围。	• 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x04040008	步 No. 出错 步 No. 超出了允许范围。	
0x04040009	指定电路容量溢出出错 单个电路块过大。	
0x0404000A	回送电路出错 回送电路不正确。	
0x0404000B	电路创建不良出错 电路数据不正确。	
0x0404000C	SFC 数据出错 转换源 SFC 数据不正确。	
0x0404000D	列表数据出错 转换源列表数据不正确。	
0x0404000E	注释数据出错 转换源注释数据不正确。	
0x0404000F	声明出错 转换源声明数据不正确。	
0x04040010	其它出错 (CAD 代码→中间代码转换中发生的其它出错)	
0x04041001	中间代码数据出错 没有转换源中间代码数据。中间代码的格式有错误。	
0x04041002	CAD 数据类型出错 输入数据类型与输出 CAD 数据类型不匹配。	

出错代码	出错内容	处理方法
0x04041003	库出错 库的加载失败。	
0x04041004	输入数据不足出错 转换源数据不足。	
0x04041005	存储区域不足出错 转换后 CAD 数据存储区域不足。	
0x04041006	无 END 指令出错 转换源数据中没有 END 指令。	
0x04041007	指令代码异常出错 转换源数据中存在有指令代码异常。	
0x04041008	软元件 No. 出错 软元件 No. 超出了允许范围。	
0x04041009	步 No. 出错 步 No. 超出了允许范围。	
0x0404100A	指定电路容量溢出出错 单个电路块过大。	
0x0404100B	回送电路出错 回送电路不正确。	
0x0404100C	电路创建不良出错 电路数据不正确。	
0x0404100D	SFC 数据出错 转换源 SFC 数据不正确。	
0x0404100E	列表数据出错 转换源列表数据不正确。	
0x0404100F	注释数据出错 转换源注释数据不正确。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x04041010	声明出错 转换源声明数据不正确。	
0x04041011	其它出错 (中间代码→ CAD 代码转换中发生的其它出错)	
0x04050001	指定字符串异常出错 软元件字符串不正确。	
0x04050002	软元件点数出错 软元件点数超出了允许范围。	
0x04050003	其它出错 (软元件字符串→软元件中间代码转换中发生的其它出错)	
0x04051001	软元件名出错 软元件中间代码的区分指定不正确。	
0x04051002	软元件名出错 扩展指定软元件中间代码的区分指定不正确。	
0x04051003	其它出错 (软元件中间代码→软元件字符串转换中发生的其它出错)	
0x04052001	指定字符串异常出错 软元件字符串不正确。	
0x04052002	软元件点数出错 软元件点数超出了允许范围。	
0x04052003	其它出错 (软元件字符串→软元件表示代码转换中发生的其它出错)	
0x04053001	软元件显示出错 软元件中间代码的指定区分不正确。	

出错代码	出错内容	处理方法
0x04053002	软元件表示出错 扩展指定软元件中间代码的区分指定不正确。	
0x04053003	软元件显示出错 软元件修饰部分指定不正确。	
0x04053004	软元件显示出错 扩展软元件修飾部指定不正确。	
0x04053005	其它出错 (软元件显示代码→软元件字符串转换中发生的其它出错)	
0x04064001	软元件中间代码异常出错 软元件中间代码不正确。	
0x04064002	其它出错 (软元件中间代码→软元件名转换中发生的其它出错)	
0x04065001	软元件名异常出错 软元件中间代码的区分指定不正确。	
0x04065002	软元件名异常出错 扩展指定软元件中间代码的代码区分指定不正确。	
0x04065003	其它出错 (软元件名→中间代码转换中发生的其它出错)	
0x04066001	软元件中间代码出错 软元件中间代码不正确。	
0x04066002	其它出错 (软元件中间代码→软元件显示转换中发生的其它出错)	
0x04067001	软元件显示出错 软元件中间代码的区分指定不正确。	
0x04067002	软元件显示出错 扩展指定软元件中间代码区分指定不正确。	
0x04067003	软元件显示出错 软元件修饰部指定不正确。	
0x04067004	软元件显示出错 扩展软元件修饰部分指定不正确。	
0x04067005	其它出错 (软元件显示→软元件中间代码转换中发生的其它出错)	
0x04070001	通用数据转换出错 软元件注释转换的输入数据不正确。	
0x04070002	通用数据不足 转换源的数据不足。	
0x04070003	存储区域不足 存储转换数据的区域不足。	
0x04071001	CPU 数据转换出错 软元件注释转换的输入数据不正确。	
0x04071002	CPU 数据不足出错 转换源的数据不足。	
0x04071003	存储区域不足出错 存储转换数据的区域不足。	
0x04072001	打开出错 转换对象的生成失败。	
0x04072002	CPU 类型出错 指定的 CPU 类型不存在。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。

出错代码	出错内容	处理方法
0x04072003	无转换出错 转换对象不存在。	
0x04072004	输入数据出错 输入数据不正确。	
0x04073001	程序通用数据转换出错	
0x04073002	程序通用数据转换出错	
0x04073101	程序 CPU 数据转换出错	
0x04074001	参数通用数据出错	
0x04074002	网络参数通用数据出错 虽然存在参数块，但其中的数据未设置。	
0x04074101	参数 CPU 数据出错	
0x04074102	网络参数 CPU 数据出错 虽然存在参数块，但其中的数据未设置。	
0x04074103	偏置出错	
0x04074201	网络类型指定出错 是在指定 CPU 中不支持的网络类型。	
0x04074202	参数块 No. 出错 指定参数块 No. 的块不存在。	
0x04074203	参数块内容出错 与指定 CPU 中支持的内容不相同。	
0x04074204	参数块信息出错 指定了不存在的块 No.。	
0x04074205	默认参数块异常 指定了不存在的块 No.。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x04074301	通用参数块转换出错	
0x04074302	通用参数块 No. 1001 出错 RUN-PAUSE 设置有无标志的值不正确。	
0x04074303	通用参数块 No. 1003 出错	
0x04074304	通用参数块 No. 1008 出错	
0x04074305	通用参数块 No. 1100 出错	
0x04074306	通用参数块 No. 2001 出错 指定了不存在的软元件中间代码。	
0x04074307	通用参数块 No. 3000 出错	
0x04074308	通用参数块 No. 3002 出错	
0x04074309	通用参数块 No. 3004 出错 报警器显示模式的设置不正确。	
0x0407430A	通用参数块 No. 4000 出错 未创建 I/O 分配数据。	
0x0407430B	通用参数块 No. 5000 出错 指定网络不支持。	
0x0407430C	通用参数块 No. 5001 出错 其它站访问时未设置有效模块 No.。	
0x0407430D	通用参数块 No. 5002 出错	
0x0407430E	通用参数块 No. 5003 出错	
0x0407430F	通用参数块 No. 5NM0 出错	
0x04074310	通用参数块 No. 5NM1 出错	

出错代码	出错内容	处理方法
0x04074311	通用参数块 No. 5NM2 出错	
0x04074312	通用参数块 No. 5NM3 出错	
0x04074313	通用参数块 No. 6000 出错	
0x04074314	通用参数块 No. FF18 出错 未设置链接参数容量。	
0x04074315	通用参数块 No. FF25 出错 未设置运算电路检查。	
0x04074316	通用参数块 No. FF30 出错 未创建采样跟踪数据。	
0x04074317	通用参数块 No. FF31 出错 未创建状态锁存数据。	
0x04074318	通用参数块 No. FF42 出错 未设置定时器处理点数。	
0x04074319	通用参数块 No. FF30 出错 指定扩展定时器用设置值软元件不存在。	
0x0407431A	通用参数块 No. FF44 出错	
0x0407431B	通用参数块 No. FF45 出错	
0x0407431C	通用参数块 No. FF60 出错 未进行终端设置。	
0x0407431D	通用参数块 No. FF70 出错 未设置用户开放区域。	
0x04074401	CPU 参数块转换出错	
0x04074402	CPU 参数块 No. 1001 出错	
0x04074403	CPU 参数块 No. 1003 出错	
0x04074404	CPU 参数块 No. 1008 出错	
0x04074405	CPU 参数块 No. 1100 出错	
0x04074406	CPU 参数块 No. 2001 出错	
0x04074407	CPU 参数块 No. 3000 出错	
0x04074408	CPU 参数块 No. 3002 出错	
0x04074409	CPU 参数块 No. 3004 出错	
0x0407440A	CPU 参数块 No. 4000 出错	
0x0407440B	CPU 参数块 No. 5000 出错 指定网络类型不支持。	
0x0407440C	CPU 参数块 No. 5001 出错	
0x0407440D	CPU 参数块 No. 5002 出错	
0x0407440E	CPU 参数块 No. 5003 出错	
0x0407440F	CPU 参数块 No. 5NM0 出错 指定网络类型不支持。	
0x04074410	CPU 参数块 No. 5NM1 出错	
0x04074411	CPU 参数块 No. 5NM2 出错 指定网络类型不支持。	
0x04074412	CPU 参数块 No. 5NM3 出错	
0x04074413	CPU 参数块 No. 6000 出错	
0x04074414	CPU 参数块 No. FF18 出错	
0x04074415	CPU 参数块 No. FF25 出错	
0x04074416	CPU 参数块 No. FF30 出错	
0x04074417	CPU 参数块 No. FF31 出错	
0x04074418	CPU 参数块 No. FF42 出错	
0x04074419	CPU 参数块 No. FF43 出错	

出错代码	出错内容	处理方法
0x0407441A	CPU 参数块 No. FF44 出错	
0x0407441B	CPU 参数块 No. FF45 出错	
0x0407441C	CPU 参数块 No. FF60 出错	
0x0407441D	CPU 参数块 No. FF70 出错	
0x04075001	通用数据转换出错 软元件存储器设置部分转换失败。	
0x04075002	通用数据转换出错 软元件存储器数据部分转换失败。	
0x04075003	通用数据转换出错 软元件存储器数据部分不存在。	
0x04075101	CPU 数据转换出错 软元件存储器设置部分转换失败。	
0x04075102	CPU 数据转换出错 软元件存储器数据部分转换失败。	
0x04076001	通用数据转换出错 软元件注释设置部分转换失败。	
0x04076002	通用数据转换出错 软元件注释数据部分转换失败。	
0x04076101	CPU 数据转换出错 软元件注释设置部分转换失败。	
0x04076102	CPU 数据转换出错 软元件注释数据部分转换失败。	
0x04077001	通用数据转换出错 采样跟踪设置部分转换失败。	
0x04077002	通用数据转换出错 采样跟踪数据部分转换失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x04077101	CPU 数据转换出错 采样跟踪设置部分转换失败。	
0x04077102	CPU 数据转换出错 采样跟踪数据部分转换失败。	
0x04078001	通用数据转换出错 状态锁存设置部分转换失败。	
0x04078002	通用数据转换出错 状态锁存数据部分转换失败。	
0x04078101	CPU 数据转换出错 状态锁存设置部分转换失败。	
0x04078102	CPU 数据转换出错 状态锁存数据部分转换失败。	
0x04079101	故障履历 CPU 数据转换出错	
0x0407A101	文件一览 CPU 数据转换出错	
0x0407B101	出错信息 CPU 数据转换出错	
0x0407C001	间接地址→软元件名转换出错 未预留软元件名存储区域。	
0x0407C002	软元件名→间接地址转换出错 未预留间接地址存储区域。	
0x0407C003	间接地址→软元件显示转换出错 未预留软元件显示存储区域。	
0x0407C004	软元件显示→间接地址转换出错 未预留间接地址存储区域。	
0x0407C005	间接地址→软元件字符串转换出错 未预留软元件字符串存储区域。	

出错代码	出错内容	处理方法
0x0407C006	软元件字符串→间接地址转换出错 未预留间接地址存储区域。	
0x0407C007	中间代码→软元件名转换出错 未预留软元件名存储区域。	
0x0407C008	软元件名→中间代码转换出错 未预留中间代码存储区域。	
0x0407C009	中间代码→软元件显示转换出错 未预留软元件显示存储区域。	
0x0407C00A	软元件显示→中间代码转换出错 未预留中间代码存储区域。	
0x0407C00B	中间代码→间接地址转换出错 未预留间接地址存储区域。	
0x0407C00C	间接地址→中间代码转换出错 未预留中间代码存储区域。	
0x0407C00D	CPU 类型出错 指定的 CPU 类型不支持。	
0x0407C00E	软元件字符串出错 指定的软元件不支持。	
0x0407C00F	软元件字符串出错 指定的软元件字符串、类型不正确。	
0x0407C010	软元件出错 指定软元件在指定 CPU 中不支持。	
0x0407C011	CPU 类型出错 指定的 CPU 不支持。	
0x0407C012	超出软元件范围出错	
0x0407D001	通用数据转换出错 SFC 跟踪条件设置部分转换出错	
0x0407D002	通用数据转换出错 SFC 跟踪数据部分转换出错	
0x0407D101	CPU 数据转换出错 SFC 跟踪条件设置部分转换出错	
0x0407D102	CPU 数据转换出错 SFC 跟踪数据部分转换出错	
0x04080001	超出中间代码区分范围出错 指定的中间代码区分超出了允许范围。	
0x04080002	扩展指定超出中间代码区分范围出错 指定的扩展指定中间代码区分超出了允许范围。	
0x04080003	无软元件点数检查出错 是不进行软元件点数检查的软元件。	
0x04090001	GPP 工程出错 指定的 CPU 类型与 GPP 工程类型不匹配。	
0x04090002	文件类型出错 指定的 GPP 工程类型与文件类型不匹配。	
0x04090010	转换源 GPP 数据不足 转换源数据不存在。 转换源数据容量的指定不正确。	
0x04090011	转换数据存储区域不足 出错转换数据存储区域不足。	

出错代码	出错内容	处理方法
0x04090012	转换源 GPP 数据出错 转换源 GPP 数据不正确。	
0x04090110	转换源数据不足出错 转换源数据不存在。 转换源数据容量的指定不足。	
0x04090111	转换数据存储区域不足出错 转换数据存储区域不足。	
0x04090112	转换源数据出错 转换源数据不正确。	
0x040A0001	中间代码存储区域不足 转换后数据的存储区域不足。	
0x040A0002	SFC 附加信息存储区域不足	
0x040A0003	转换出错	
0x040A0004	非 SFC 程序出错	
0x040A1001	步未使用 • 无动作输出出错	
0x040A1002	步 No. 超出允许范围出错	
0x040A1003	步未使用 • 无动作输出出错	
0x040A1004	转移 No. 超出允许范围出错	
0x040A1005	最大数溢出出错	
0x040A1006	微机程序容量出错	
0x040A1007	非 SFC 程序出错	
0x040B0001	中间代码存储区域不足 转换后数据的存储区域不足。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x040B0002	转换出错	
0x040B1001	步开始位置表格创建失败	
0x040B1002	步信息读取失败	
0x040B1003	步 No. 出错	
0x040B1004	动作输出 • 转移条件中间代码读取失败出错	
0x040B1005	内部工作区域预留失败出错	
0x040B1006	字符存储器 X 轴方向最大值设置出错	
0x040B1007	内部工作区域不足出错	
0x040B1008	堆栈溢出、字符存储器异常	
0x040B1009	存储块数不足出错	
0x040B100A	非 SFC 程序出错	
0x04FFFFFF	其它出错	
0x10000001	无指令出错	
0x10000002	MX Component 的通信用 DLL 的启动失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。
0x10000003	Open 失败。(DiskDrive)	
0x10000004	重复 Open 出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。
0x10000005	文件访问出错	
0x10000006	文件夹名不正确出错	
0x10000007	文件访问禁止出错	
0x10000008	Disk 已满出错	
0x10000009	文件删除出错	
0x1000000A	文件名不正确出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。

出错代码	出错内容	处理方法
0x1000000C	处于其它应用程序或其它线程请求执行中因此执行失败。	<ul style="list-style-type: none"> • 请在一定时间后重新执行。 • 按照 COM 及 ActiveX 的多线程规则进行编程。 • 结束程序，重新启动计算机。
0x1000000D	文件夹创建出错	
0x1000000E	文件夹 / 文件类型出错	
0x1000000F	偏置地址出错	
0x10000010	请求取消 发生了取消处理。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x10000011	存储器预留出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。
0x10000012	未执行打开	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。
0x10000013	连接未执行出错	
0x10000014	对象无效出错	
0x10000015	请求取消失败出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x10000016	状态读取失败出错	
0x10000017	指定容量（软元件数）不正确。	<ul style="list-style-type: none"> • 确认方法中指定的点数。 • 结束程序，重新启动计算机。
0x10000018	没有登录软元件。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。
0x10000019	未执行数据设置	
0x1000001A	读取未执行出错	
0x1000001B	创建标志不正确出错	
0x1000001C	操作溢出访问	
0x1000001D	软元件重复出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x1000001E	注册表查找失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 结束其它程序，预留出空余的存储器。 • 重新安装 MX Component。
0x1000001F	文件类型出错	
0x10000020	软元件存储器类型出错	
0x10000021	程序范围出错	
0x10000022	TEL 类型出错	
0x10000023	TEL 访问出错	
0x10000024	取消标志类型出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。 • 请向当地三菱电机代理商咨询。
0x10000030	重复软元件登录出错	
0x10000031	未登录软元件出错	
0x10000032	指定软元件出错	<ul style="list-style-type: none"> • 对指定软元件的内容进行重新审核。 • 结束程序，重新启动计算机。
0x10000033	指定软元件范围出错	<ul style="list-style-type: none"> • 结束其它程序，预留出空余的存储器。

出错代码	出错内容	处理方法
0x10000034	文件写入出错	
0x10000040	服务器启动失败。	
0x10000041	服务器停止出错 服务器停止失败。	
0x10000042	服务器重复启动出错	
0x10000043	服务器未启动出错	
0x10000044	资源超时出错	
0x10000045	服务器类型出错	
0x10000046	服务器访问失败出错	
0x10000047	服务器访问已实施出错	
0x10000048	模拟器启动失败	
0x10000049	模拟器结束失败	
0x1000004A	模拟器未启动出错	
0x1000004B	模拟器类型出错	
0x1000004C	模拟器不支持出错	
0x1000004D	模拟器重复启动出错	
0x1000004E	共享存储器未启动出错	
0x8001000C	指定了超出可设置范围的数据。	• 对属性的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80010016		• 对属性的值重新进行审核。 • 通过通信设置实用程序重新进行设置。 • 对模块的设置及电缆状态等的系统重新进行审核。 • 进行 Close 后，重新进行 Open。 • 应结束程序后，重新启动计算机。
0x80010101	超时出错	• 对属性的值重新进行审核。 • 通过通信设置实用程序重新进行设置。 • 对属性的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80020001	数据的类型不正确。	• 对属性的值重新进行审核。
0x80020002	数据的范围不正确。	• 通过通信设置实用程序重新进行设置。
0x80030001	通信端口的指定不正确。	• 对属性的通信端口的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030002	传送速度的指定不正确。	• 对属性的传送速度的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030003	数据位长度的指定不正确。	• 对属性的数据位的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030004	奇偶的指定不正确。	• 对属性的奇偶的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030005	停止位长度的指定不正确。	• 对属性的停止位的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030006	等待时间的指定不正确。	• 对属性的等待时间的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030007	CR/LF 有无的指定不正确。	• 对属性的 CR/LF 的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030008	超时的指定不正确。	• 对属性的超时值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x80030009	站号的指定不正确。	• 对属性的站号的值重新进行审核。 • 通过通信设置实用程序重新进行设置。
0x8003000A	USB 通信设置不正确。	• 对属性的值重新进行审核。
0x8003000B	USB 机型代码不正确。	• 通过通信设置实用程序重新进行设置。

出错代码	出错内容	处理方法
0x8003000C	USB 通信中检测到重复站号。	<ul style="list-style-type: none"> 对属性的站号的值重新进行审核。 通过通信设置实用程序重新进行设置。
0x80200107	通信出错	<ul style="list-style-type: none"> 以相同方法再次执行。 结束程序，重新启动计算机。
0x80200203	存储器预留出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 结束其它程序，预留出空余的存储器。
0x80201001	指定 CPU 出错	<ul style="list-style-type: none"> 连接支持的运动 CPU。
0x80201101	已打开出错 在打开状态下执行了 Open 方法。	<ul style="list-style-type: none"> 更改通信对象 CPU 的情况下，应执行 Close 后，再执行 Open 方法。
0x80201104	DLL 加载出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 结束其它程序，预留出空余的存储器。 重新安装 MX Component。
0x80201106	通信对象生成出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。
0x80201201	未打开出错 未执行 Open 方法。	<ul style="list-style-type: none"> 执行 Open 方法后，再执行相应方法。
0x80201203	存储器预留出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 结束其它程序，预留出空余的存储器。
0x80204203	存储器预留出错	
0x80205203	存储器预留出错	<ul style="list-style-type: none"> 确认 ActCpuType 中设置的 CPU 类型。 确认是否为系统配置不支持的配置。
0x80205001	指定 CPU 出错	
0x80209501	MT Simulator2 启动出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x80209502	MT Simulator2 启动出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x80209503	MT Simulator2 通信出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x80209504	MT Simulator2 启动出错 已达到可同时启动的模拟数的上限，因此未能启动。	<ul style="list-style-type: none"> 结束 MT Simulator2 后重试。
0x80209505	MT Simulator2 停止出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x80209506	MT Simulator2 停止出错	<ul style="list-style-type: none"> 结束程序，重新启动计算机。
0x8020950F	MT Simulator2 停止出错 MT Simulator2 未启动。	<ul style="list-style-type: none"> 请向当地三菱电机代理商咨询。
0x80209510	MT Simulator2 启动出错 MT Simulator2 未启动。	<ul style="list-style-type: none"> 开始 MT Simulator2 后重试。
0x80209516	MT Simulator2 启动出错 MT Developer2 未安装。	<ul style="list-style-type: none"> 安装 MT Developer2。
0x80209518	MT Simulator2 启动出错 指定号机编号已在使用。	<ul style="list-style-type: none"> 结束指定号机编号的 MT Simulator2 后执行。
0x80209519	发送数据不正确出错	<ul style="list-style-type: none"> 对自变量重新进行审核。
0x8020951C	MT Simulator2 启动出错 MT Simulator2 为结束处理中。	<ul style="list-style-type: none"> 结束 MT Simulator2 后重试。
0x8020951D	MT Simulator2 启动出错 已 Open。	<ul style="list-style-type: none"> Close 后重试。
0x8020951E	MT Simulator2 未 Open 出错	<ul style="list-style-type: none"> 请向当地三菱电机代理商咨询。
0x8020951F	MT Simulator2 启动出错 已 Open。	<ul style="list-style-type: none"> Close 后重试。
0x80206004	远程请求出错	<ul style="list-style-type: none"> 将设备本体开关置为 Run 后，结束测试模式重新执行。

出错代码	出错内容	处理方法
0x8020A104	DLL 加载出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 结束其它程序，预留出空余的存储器。 • 重新安装 MX Component。
0x8020A203	存储器预留出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 结束其它程序，预留出空余的存储器。
0x80A00101	未打开出错 未执行 Open 方法。	<ul style="list-style-type: none"> • 执行 Open 方法后，再执行相应方法。
0x80A00104	已打开出错 在打开状态下执行了 Open 方法。	<ul style="list-style-type: none"> • 更改通信对象 CPU 的情况下，应执行 Close 后，再执行 Open 方法。
0x80A00105	自变量的数据类型不正确。	<ul style="list-style-type: none"> • 确认自变量中使用的数据类型。
0x80A00106	自变量的数据范围不正确	<ul style="list-style-type: none"> • 确认自变量中使用的值。
0x80A00107	无指令出错	<ul style="list-style-type: none"> • 不支持该方法。
0x80A00109	数据源无法打开。	<ul style="list-style-type: none"> • 确认电缆的连接。
0x80A0010C	机器人控制器的答复格式错误。	<ul style="list-style-type: none"> • 采取防干扰措施。
0xF0000001	无许可证出错 计算机未被授予许可证。	<ul style="list-style-type: none"> • 通过许可证密钥 FD 对计算机授予许可证。
0xF0000002	设置数据读取出错 逻辑站号的设置数据读取失败。	<ul style="list-style-type: none"> • 指定正确的逻辑站号。 • 通过通信设置实用程序设置逻辑站号。
0xF0000003	已打开出错 在打开状态下执行了 Open 方法。	<ul style="list-style-type: none"> • 更改通信对象 CPU 的情况下，应执行 Close 后，再执行 Open 方法。
0xF0000004	未打开出错 未执行 Open 方法。	<ul style="list-style-type: none"> • 执行 Open 方法后，再执行相应方法。
0xF0000005	初始化出错 MX Component 内部保持的对象的初始化失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 重新安装 MX Component。
0xF0000006	存储器预留出错 MX Component 内部存储器的预留失败。	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。 • 结束其它程序，预留出空余的存储器。
0xF0000007	功能不支持出错 不支持该方法。	<ul style="list-style-type: none"> • 由于不支持相应方法，因此不使用。
0xF1000001	字符代码转换出错 字符代码的转换 (UNICODE → ASCII 代码或 ASCII 代码→ UNICODE) 失败。	<ul style="list-style-type: none"> • 确认方法中指定的字符串。 • 从可编程控制器 CPU 中获取的 ASCII 字符串异常。 • 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 • 结束程序，重新启动计算机。 • 对 GetCpuType 方法进行重试。
0xF1000002	起始 I/O 编号出错 指定的起始 I/O 编号是不正确的值。 符合的起始 I/O 编号不存在。	<ul style="list-style-type: none"> • 确认方法中指定的起始 I/O 编号的值。 • 通过 GPP 功能确认可编程控制器的参数 (IO 分配)。
0xF1000003	缓冲地址出错 指定的缓冲地址是不正确的值。 缓冲地址超出了允许范围。	<ul style="list-style-type: none"> • 确认方法中指定的缓冲地址的值。
0xF1000004	缓冲读取容量出错 进行缓冲读取的结果为无法获取指定容量。	<ul style="list-style-type: none"> • 重新进行再 Open 处理。 • 对可编程控制器 CPU、模块的设置及电缆状态等的系统进行重新审核。 • 进行重试。 • 结束程序。

出错代码	出错内容	处理方法
0xF1000005	容量出错 读取 / 写入方法中指定的容量异常。 读取 / 写入起始 + 容量超出了软元件或缓冲的区域。	• 确认方法中指定的容量。
0xF1000006	操作出错 远程操作中指定的操作是异常值。	• 确认方法中指定的操作指定值。
0xF1000007	时钟数据出错 时钟数据异常。	• 确认方法中指定的时钟数据。 • 在可编程控制器 CPU 的时钟数据中设置正确的时钟数据。
0xF1000008	监视软元件登录数溢出 EntryDeviceStatus 方法中登录的软元件点数为 0 点以下。 EntryDeviceStatus 方法中登录的软元件点数多于 20 点。	• 将 EntryDeviceStatus 方法中登录的软元件点数设置为 1 点以上 20 点以下。
0xF1000009	监视软元件数据已登录出错	• 通过 FreeDeviceStatus 方法进行登录解除后，再次执行 EntryDeviceStatus 方法。
0xF1000010	软元件状态监视处理的启动失败。 软元件状态监视处理的结束失败	• 通过 EntryDeviceStatus 方法重新执行软元件状态监视处理的启动 / 结束。
0xF1000011	VARIANT 自变量的数据类型有错误。	• 对 VARIANT 自变量中指定的数据类型进行重新审核。 • 确认数组变量的容量是否足够。 • 确认是否对各方法中指定的数据类型进行了设置。
0xF1000012	软元件状态监视间隔时间的值超出了 1 秒至 1 小时 (1 ~ 3600) 的范围。	• 将软元件状态监视时间在 1 ~ 3600 之间进行指定。
0xF1000013	已 Connect 出错 对同一对象执行了 Connect 后再次执行了 Connect。	• 应在执行 Disconnect 方法后，执行 Connect。
0xF1000014	电话号码不正确出错 电话号码中包含有超出允许设置字符 “0123456789-*#” 范围的字符。	• 重新设置电话号码后，再次执行 Connect。
0xF1000015	排他控制失败出错 执行 Connect 时以及执行 Disconnect 时排他控制处理失败。	• 其它对象正在执行 Connect 或 Disconnect 的情况下，应在 Connect/Disconnect 方法完成后，重新执行此前失败的方法 (Connect/Disconnect)。 • 仅对象本身正在进行 Connect/Disconnect 处理的情况下，应执行以下处理。 <ul style="list-style-type: none">• 结束程序。• 重新启动计算机。• 重新安装 MX Component。
0xF1000016	电话线路连接中出错 使用了 MX Component 的应用程序以外的其它应用程序已处于电话线路连接中。	• 在电话线路连接中的应用程序中断开电话线路后，再次执行 Connect。
0xF1000017	电话线路未连接中出错 电话线路处于未连接状态。 执行 Connect 后，电话线路处于连接中状态，但由于某种原因导致电话线路断开。	• Connect 方法失败的情况下，应执行 Disconnect 后，再次执行 Connect。 • Connect 以外的方法失败的情况下，应执行 Disconnect 后，再次执行 Connect 连接电话线路。电话线路连接后，执行此前失败的方法。

出错代码	出错内容	处理方法
0xF1000018	无电话号码出错 未设置电话号码。 连接方式为自动（指定回调编号时）、回调连接（指定编号时）、回调请求（指定编号时）时，未设置电话号码或回调编号。	<ul style="list-style-type: none"> 程序设置类型的情况下，应在属性 ActDialNumber 中设置电话号码。 (连接方式为自动（指定回调编号时）、回调连接（指定编号时）、回调请求（指定编号时）时，应在属性 ActDialNumber、属性 ActCallbackNumber 中设置电话号码。) 实用程序设置类型的情况下，应通过向导设置电话号码。 (连接方式为自动（指定回调编号时）、回调连接（指定编号时）、回调请求（指定编号时）时，应设置电话号码、回调编号。)
0xF1000019	未 Close 出错 在 Open 中执行了 Disconnect。	<ul style="list-style-type: none"> 应在执行了 Close 之后，再次执行 Disconnect。
0xF100001A	电话线路连接目标不一致出错 使用已处于电话线路连接中的端口，对与当前连接中的电话号码不同的其它电话号码执行了 Connect。(连接方式为回调接收的情况下，非回调接收的其它连接方式将被视为电话号码不相同。)	<ul style="list-style-type: none"> 对已处于电话线路连接中的电话号码以外的其它电话号码进行连接的情况下，应对连接中的电话线路执行 Disconnect，断开电话线路后，再执行 Connect。 通过回调接收进行线路连接的情况下，应将同一端口中先执行 Connect 的连接方式设置为回调接收。
0xF100001B	控件类型不一致出错 对与已处于电话线路 Tel 连接中的对象不同控件类型的对象执行了 Connect。	<ul style="list-style-type: none"> 通过当前处于电话线路连接中的对象执行 Disconnect，断开电话线路后，再次执行 Connect。
0xF100001C	未 Disconnect 出错 进行了实际电话线路连接的对象执行了 Disconnect 时，其它对象中存在有 Connect 中的对象。	<ul style="list-style-type: none"> 在 Connect 中的对象内，全部执行 Disconnect 后，应再次通过进行了实际电话线路连接的对象执行 Disconnect。
0xF100001D	未 Connect 出错 在未执行 Connect 的状况下执行了 Open。或者执行了 Disconnect。	<ul style="list-style-type: none"> 执行 Connect 之后再次执行 Open。 执行 Connect 之后再次执行 Disconnect。
0xF100001E	严重出错	<ul style="list-style-type: none"> 结束程序。 重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
0xF100001F	Open 时设置出错 Connect 时使用的电话号码、端口编号的设置值与 Open 时设置的值不相同。 连接方式的指定有错误。	<ul style="list-style-type: none"> 确认电话号码、端口编号。 确认连接方式。
0xF2000002	连接目标 TEL 进行了出错响应。 可能是以下原因所致。 • 发生了通信出错。	<ul style="list-style-type: none"> 程序设置类型的情况下，确认属性的设置值。 实用程序设置类型的情况下，确认向导中的设置内容。
0xF2000003	接收了不正确的数据。 可能是以下原因所致。 • 由于噪声导致接收了错误报文。	<ul style="list-style-type: none"> 进行重试。 确认通信对象的设备。
0xF2000004	没有来自于调制解调器的响应。 可能是以下原因所致。 • 调制解调器异常。 • 电话号码设置错误。	<ul style="list-style-type: none"> 确认调制解调器的状态。 确认电话号码。 即使确认上述 2 点后仍然有问题的情况下，在程序设置类型中，更改属性（设置 ActConnectionCDWaitTime 等的时间的属性）的值，在实用程序设置类型中，更改通过向导设置的详细设置的内容。

出错代码	出错内容	处理方法
0xF2000005	线路有可能未断开。	<ul style="list-style-type: none"> 确认电话线路。
0xF2000006	计算机侧调制解调器未受理 AT 指令。 可能是以下原因所致。 <ul style="list-style-type: none"> 指定了不支持的 AT 指令。 调制解调器异常。 	<ul style="list-style-type: none"> 确认设置的 AT 指令的内容。 确认调制解调器的状态。
0xF2000007	调制解调器未正常响应标准的跳脱指令。	<ul style="list-style-type: none"> 确认调制解调器。 确认超时值是否过短。 (建议 5000ms 以上。)
0xF2000009	调制解调器未正常响应线路断开指令。	<ul style="list-style-type: none"> 确认调制解调器。
0xF200000A	对象未接收信号。 <ul style="list-style-type: none"> 对象调制解调器的信号接收设置有错误。 正在通话中。 电话号码有错误。 	<ul style="list-style-type: none"> 确认对象调制解调器的信号接收设置。 确认对象是否正在通话中。 确认电话号码。
0xF200000B	回调接收等待超时。	<ul style="list-style-type: none"> 延长回调接收等待超时 (ActCallbackReceptionWaitingTimeOut) 的时间后，再次执行 Connect。
0xF200000C	QJ71C24 模块的口令未能解除。	<ul style="list-style-type: none"> 在属性 ActPassword 中进行口令设置后，再次执行此前失败的方法。
0xF2010001	回调线路断开等待时间超出了 0 ~ 180 秒的范围。 回调实施延迟时间超出了 0 ~ 1800 秒的范围。 电话号码大于 62 个字符。	<ul style="list-style-type: none"> 确认回调线路断开等待时间是否在 0 ~ 180 秒的范围内。 确认回调实施延迟时间是否在 0 ~ 1800 秒以内。 确认电话号码是否在 62 个字符以内。 结束程序，重新启动计算机。 重新安装 Component。 请向当地三菱电机代理商咨询。
0xF2010002	QJ71C24 未受理指定的连接方式。 可能是以下原因所致。 <ul style="list-style-type: none"> 连接方式错误。 回调用电话号码错误。 	<ul style="list-style-type: none"> 确认 MX Component 中设置的内容与 QJ71C24 侧的设置是否一致。
0xF2010003	QJ71C24 在“自动（固定为回调时 / 指定回调编号时）”中的连接未被许可。	<ul style="list-style-type: none"> 确认 QJ71C24 的设置。
0xF2100005	有可能是线路未断开。	<ul style="list-style-type: none"> 调制解调器及线路中无异常的情况下，在程序设置类型中应更改属性（设置 ActConnection CDWaitTime 等时间的属性）的值，在实用程序设置类型中应更改通过向导设置的详细设置的内容。
0xF2100006	调制解调器未受理初始化 AT 指令。	<ul style="list-style-type: none"> 在程序设置类型中，更改属性 ActATCommand 的设置。 在实用程序设置类型中，更改通过向导设置的 AT 指令。
0xF2100007	计算机侧调制解调器未响应跳脱指令。	<ul style="list-style-type: none"> 计算机侧调制解调器中无异常的情况下，在程序设置类型中应更改属性（设置 ActConnection CDWaitTime 等时间的属性）的值，在实用程序设置类型中应更改通过向导设置的详细设置的内容。
0xF2100008	调制解调器对计算机侧的发送数据无响应。	<ul style="list-style-type: none"> 在程序设置类型中应更改属性（设置 ActConnection CDWaitTime 等时间的属性）的值，在实用程序设置类型中应更改通过向导设置的详细设置的内容。

出错代码	出错内容	处理方法
0xF21000**	<p>调制解调器未响应。 可能是以下原因所致。</p> <ul style="list-style-type: none"> • 调制解调器异常。 • 电话号码设置错误。 	<ul style="list-style-type: none"> • 确认调制解调器的状态。 • 确认电话号码。 • 确认了上述 2 点后仍然有问题的情况下，在程序设置类型中应更改属性（设置 ActConnection CDWaitTime 等时间的属性）的值，在实用程序设置类型中应更改通过向导设置的详细设置的内容。
0xF202****	<p>通信失败。 根据状态有可能是以下原因所致。</p> <ul style="list-style-type: none"> • 通信超时（电缆断线、指定端口不支持、COM 端口指定错误） • 调制解调器的电源处于 OFF 状态。 	<ul style="list-style-type: none"> • 确认电缆是否断线。 • 确认是否指定端口不支持。 • 确认 COM 端口设置是否正确。 • 确认调制解调器的电源是否处于 OFF 状态。 • 关于更加详细的故障排除有关内容，请参阅将前 4 位数替换为“0x0180”的出错代码所对应的内容。 例：“0xF202480B”的情况下 请参阅“0x0180480B”的出错代码。
0xF3000001	标签出错 方法中指定的标签字符串是不正确的标签字符串。	<ul style="list-style-type: none"> • 对标签名进行重新审核。
0xF3000002	标签管理服务未启动出错	<ul style="list-style-type: none"> • 结束程序，重新启动计算机。
0xF3000003	MMS 服务未启动出错	<ul style="list-style-type: none"> • 重新安装 MX Component。
0xF3000004	正在更新系统标签数据	<ul style="list-style-type: none"> • 进行重试。
0xFF000001	MX Component 试用版出错 MX Component 试用版已过期。	<ul style="list-style-type: none"> • 卸载 MX Component 试用版后安装正式版，或重新安装试用版。

7.2 CPU、模块、网卡返回的出错代码

本节介绍CPU、模块及网卡返回的出错代码有关内容。

要 点

有时会发生未按下述(1)所示返回出错代码的现象。
应首先确认(2)的内容后，再参阅(1)。

(1) 出错代码一览

CPU、模块及网卡检测出错误的情况下，将返回如下表所示的出错代码。

高位2字节表示检测出错误的模块，低位2字节表示检测出错误的模块返回的出错代码。

关于出错的详细内容，请参阅出错代码所对应的CPU、模块或网卡的手册。

出错代码	检测出错误的模块
0x01010000 ~ 0x0101FFFF*1,*2	运动控制CPU
0x01070000 ~ 0x0107FFFF*1	CC-Link IE控制网卡、MELSECNET/H板、CC-Link板
0x01090000 ~ 0x0109FFFF*1	FXCPU
0x010A0000 ~ 0x010AFFFF*1	QCPU(Q模式)、QSCPU、RCPU
0x010B0000 ~ 0x010BFFFF*1	Q系列C24
0x010C0000 ~ 0x010CFFFF*1	Q系列E71
0x010D0000 ~ 0x010DFFFF*1	计算机CPU模块
0x010F0000 ~ 0x010FFFFFF*1	GOT

*1：请参阅要点。

*2：496页 7.1节 各控件返回的出错代码中也记载有出错代码。

(2) 确认出错代码时的注意事项

确认CPU、模块及网卡返回的出错代码时的注意事项如下所示。

(a) 属性设置错误

所使用的系统配置与设置的属性值不匹配的情况下，高位2字节将无法正确显示检测出错误的模块。

例如，对于QCPU(Q模式)，在ActCpuType中设置了FXCPU的属性值的情况下，高位2字节将变为FXCPU为检测出错误的模块。

在这种情况下，应确认系统配置及设置的全部属性值后，再次进行通信。

此外，使用ActUtilType控件的情况下，应确认通信设置实用程序的设置内容。

(b) 访问其它站时

访问其它站时，使用的经由模块(CC-Link IE控制网络、MELSECNET/H、CC-Link、串行通信、以太网模块)的出错代码有可能会被放入低位2字节中。

在这种情况下，表示检测出错误的模块的高位2字节有可能与发生了出错的模块不相符，应在确认系统配置的基础上，根据所使用的CPU、经由网络模块及网卡的手册进行确认。

(3) 标签名指定错误

由于 CPU 侧的问题引发了标签相关出错的情况下，例如，系统标签 Ver. 2 不存在、或在对支持系统标签 Ver. 2 的软元件进行读取 / 写入的过程中通过其他方式更改了系统标签 Ver. 2 信息等时，会返回下表所示的出错代码。

出错代码	出错内容	处置方法
0x01802001	标签信息出错	<ul style="list-style-type: none">• 指定系统标签的情况下，请通过标签管理实用程序修改分配给系统标签名的全局标签名。• 请确认 CPU 中是否存在全局标签。
0x010A4000 ~ 0x010A4FFF	关于出错内容和处置方法，请参阅以下手册。  MELSEC iQ-R CPU 模块用户手册（应用篇）	

7.3 HRESLT 型的出错代码

通常，ActiveX 控件将返回 HRESLT 型的返回值。ACT 控件也同样如此。使用自定义 I/F 的情况下，与 API 方法的返回值相当。

使用调度 I/F 的情况下，通过特定处理可以获取 HRESLT 型的返回值。

ACT 控件的 HRESLT 型的返回值如下所示。

返回值	结束状态	内容
S_OK	正常结束	函数处理正常结束。
S_FALSE	正常结束	函数处理（作为 ActiveX 控件的）正常结束，但操作（至可编程控制器的访问）失败。
E_POINTER	非正常结束	传递至函数的指针异常。
E_OUTOFMEMORY	非正常结束	存储器预留或对象创建失败。
E_FAIL	非正常结束	发生了不确定的出错。

要 点

对于调度 I/F，在未执行用于获取 HRESLT 型的返回值的特定处理的情况下，在从 ACT 控件返回 E_POINTER(E_XXXX 定义的返回值)等时在操作系统等级中将显示出错对话。

7.4 事件查看器中显示的出错代码

本节介绍 MX Component 在 Windows[®]的事件查看器中显示的出错代码有关内容。

在 MX Component 中，为了管理标签信息，使用 MMS(系统标签数据库)服务及标签管理服务。

标签管理服务中发生了出错的情况下，事件查看器的系统日志中将显示出错内容。

关于 MX Component 的标签管理服务的出错，在事件查看器的源栏中将显示“MXLabelService”。

出错及处理方法如下所示。

事件 ID	出错信息	处理方法
3	MMS Service not started.	<ul style="list-style-type: none"> 在 MMS(系统标签数据库)服务启动完毕之前请略等片刻。启动未成功的情况下，应结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。
5	MXLabelService error (%1,%2) Please perform one of the following operations. - Restart the personal computer. - Reinstall the application and try again.	<ul style="list-style-type: none"> 结束程序，重新启动计算机。 重新安装 MX Component。 请向当地三菱电机代理商咨询。

附录

附录 1 关于回调功能的连接方式

本节介绍使用了 Q 系列 C24 的调制解调器通信时的回调功能的连接方式有关内容。

回调功能可实现以下目的：通过 MX Component 进行线路连接后，通过 Q 系列 C24 的线路再连接（回调）从 MX Component 访问可编程控制器 CPU。

- (1) ~ (3) 由计算机侧承担电话费。
- (4) ~ (8) 由 Q 系列 C24 侧承担电话费。

关于回调功能的详细内容请参阅下述手册。

 Q 系列串行通信模块用户手册（应用篇）

(1) 自动

Q 系列 C24 中未设置回调功能的情况下选择此项。

(2) 自动（固定为回调时）

Q 系列 C24 中设置了回调功能的情况下在不使用回调功能的状况下进行线路连接。只有 Q 系列 C24 的缓冲存储器 (2101H) 中登录的电话号码的计算机才能连接。

(3) 自动（指定回调编号时）

Q 系列 C24 中设置了回调功能的情况下在不使用回调功能的状况下进行线路连接。只有 MX Component 中指定的电话号码的计算机才能连接。

(4) 回调连接（固定时）

只有 Q 系列 C24 的缓冲存储器 (2101H) 中登录的电话号码的计算机才进行回调。

(5) 回调连接（指定编号时）

只有 MX Component 中指定的电话号码的计算机才进行回调。

(6) 回调请求（固定时）

从任意的计算机发送回调请求，对 Q 系列 C24 的缓冲存储器 (2101H) 中登录的电话号码的计算机进行回调。（初次连接的计算机不进行回调。）

(7) 回调请求（指定编号时）

从任意的计算机发送回调请求，只对 MX Component 中指定的电话号码的计算机进行回调。（初次连接的计算机不进行回调。）

(8) 回调接收等待

通过回调请求（固定时、指定编号时）进行线路连接的情况下，将回调目标计算机选择为“回调接收等待”后进行线路连接。

附录 2 字软元件状态监视情况下的编程示例

本节介绍通过 EntryDeviceStatus 对字软元件变为负值的现象进行监视时的编程示例有关内容。

(1) 使用 Visual Basic® .NET 的情况下

使用 Visual Basic® .NET，对 D0 变为 -10、D1 变为 0、D2 变为 10 的现象进行监视时的编程示例如下所示。

```

Dim szDevice As String          ' 监视的软元件列表
Dim lInputData(2) As Long       ' 设置的值
Dim lEntryData(2) As Long       ' EntryDeviceStatus 的自变量中设置的值
Dim lReturnCode As Long         ' EntryDeviceStatus 返回值
Dim lCount As Long              ' 循环用计数器
' 在监视的软元件列表中设置 D0、D1、D2。
szDevice = "D0" + vbLf + "D1" + vbLf + "D2"
' 对 D0 设置进行监视的软元件值 "-10"。
lInputData(0) = -10
' 对 D1 设置进行监视的软元件值 "0"。
lInputData(1) = 0
' 对 D2 设置进行监视的软元件值 "10"。
lInputData(2) = 10
' 设置的值为负值的情况下，将转换为 EntryDeviceStatus 中设置的值，因此在高位 2 字节中存储 "0"。
' 进行相当于软元件点数的循环。
For lCount = 0 To 2
    ' 设置的值为负值的情况下
    If lInputData(lCount) < 0 Then
        ' 在高位 2 字节中存储 "0"，因此通过 65535(0000FFFF[hex]) 进行屏蔽。
        lEntryData(lCount) = lInputData(lCount) And 65535
    Else
        ' 设置的值为正值的情况下，原样不变地将值代入到 lEntryData 中。
        lEntryData(lCount) = lInputData(lCount)
    End If
Next
' 执行 EntryDeviceStatus。
lReturnCode = AxActUtlType1.EntryDeviceStatus(szDevice, 3, 5, lEntryData(0))

```

<ldata = -1 的情况下 >

```

Private Sub AxActUtlType1_OnDeviceStatus(ByVal szDevice As String, ByVal lData As Long, ByVal lReturnCode As Long)
Dim lCheckData As Long          ' EntryDeviceStatus 中设置的值（高位 2 字节中存储 0 之前的数据）
' 条件成立的软元件值为 WORD 型的负值（大于 32767(7FFF[Hex])）的情况下
If lData > 32767 Then
    ' 在高位 2 字节中存储 "0"，因此通过与 FFFF0000[Hex] 进行逻辑和，转换为 LONG 型的负值。
    lCheckData = lData Or &HFFFF0000
Else
    ' 条件成立的软元件值为正值的情况下，原样不变地将值代入到 lCheckData 中。
    lCheckData = lData
End If
End Sub

```

(2) 使用 Visual C++® .NET 的情况下

使用 Visual C++® .NET，对 D0 变为 -10、D1 变为 0、D2 变为 10 的现象进行监视时的编程示例如下所示。

```
CString szDevice;           // 监视的软元件列表
LONG lInputData[3];         // 设置的值
LONG lEntryData[3];         // EntryDeviceStatus 的自变量中设置的值
LONG lReturnCode;           // EntryDeviceStatus 用返回值
LONG lCount;                // 循环用计数器

// 在监视的软元件列表中设置 D0、D1、D2。
szDevice = "D0\nD1\nD2";
// 对 D0 设置进行监视的软元件值 “-10”。
lInputData[0] = -10;
// 对 D1 设置进行监视的软元件值 “0”。
lInputData[1] = 0;
// 对 D2 设置进行监视的软元件值 “10”。
lInputData[2] = 10;

// 设置的值为负值的情况下，将转换为 EntryDeviceStatus 中
// 设置的值，因此在高位 2 字节中存储 “0”。
// 进行相当于软元件点数的循环。
for(lCount = 0;lCount<=2; lCount++) {
    // 设置的值为负值的情况下
    if (lInputData[lCount] < 0 ) {
        // 在高位 2 字节中存储 “0”，因此通过 0x0000FFFF 进行屏蔽。
        lEntryData[lCount] = lInputData[lCount] & 0x0000FFFF;
        // 设置的值为正值的情况下
    }else{
        // 原样不变地将值代入到 lEntryData 中。
        lEntryData[lCount] = lInputData[lCount];
    }
}

// 执行 EntryDeviceStatus。
lReturnCode = m_Actutltype.EntryDeviceStatus(szDevice, 3, 5, lEntryData);
```

<lData = -1 的情况下 >

```
void CSampleDlg::OnDeviceStatusActutltype1(LPCTSTR szDevice, long lData, long lReturnCode)
{
    LONG lCheckData;           // EntryDeviceStatus 中设置的值（高位 2 字节中存储 0 之前的数据）

    // 条件成立的软元件值为 WORD 型的
    // 负值（大于 7FFF[Hex]）的情况下
    if(lData > 0x7FFF) {
        // 在高位 2 字节中存储 “0”，因此通过与 FFFF0000[Hex] 的
        // 逻辑和，转换为 LONG 型的负值。
        lCheckData = lData | 0xFFFF0000;
    }else{
        // 条件成立的软元件值为正值的情况下
        // 原样不变地将值代入到 lCheckData 中。
        lCheckData = lData;
    }
}
```

(3) 使用 Visual C#® .NET 的情况下

使用 Visual C#® .NET，对 D0 变为 10、D1 变为 0、D2 变为 10 的现象进行监视时的编程示例如下所示。

```

String szDevice;           // 监视的软元件列表
int[] iInputData = new int[3]; // 设置的值
int[] iEntryData = new int[3]; // EntryDeviceStatus 的自变量中设置的值
int iReturnCode;           // EntryDeviceStatus 用返回值
int iCount;                // 循环用计数器

// 在监视的软元件列表中设置 D0、D1、D2。
szDevice = "D0\nD1\nD2";
// 对 D0 设置进行监视的软元件值 “-10”。
iInputData[0] = -10;
// 对 D1 设置进行监视的软元件值 “0”。
iInputData[1] = 0;
// 对 D2 设置进行监视的软元件值 “10”。
iInputData[2] = 10;

// 设置的值为负值的情况下，将转换为 EntryDeviceStatus 中
// 设置的值，因此在高位 2 字节中存储 “0”。
// 进行相当于软元件点数的循环。
for(iCount = 0;iCount<=2; iCount++) {
    // 设置的值为负值的情况下
    if (iInputData[iCount] < 0 ){
        // 在高位 2 字节中存储 “0”，因此通过 0x0000FFFF 进行屏蔽。
        iEntryData[iCount] = iInputData[iCount] & 0x0000FFFF;
    } // 设置的值为正值的情况下
    else{
        // 原样不变地将值代入到 iEntryData 中。
        iEntryData[iCount] = iInputData[iCount];
    }
}
axActUtlType1.ActLogicalStationNumber = 0;
iReturnCode = axActUtlType1.Open();
// 执行 EntryDeviceStatus。
iReturnCode = axActUtlType1.EntryDeviceStatus(szDevice, 3, 5, ref iEntryData[0]);

```

附录 3 关于超时时间

在 MX Component 中有时会发生由于与 ACT 控件内的 ActTimeOut 属性中设置的值不同的其它时间而发生超时的现象。本节介绍各状态下的超时时间有关内容。

附录 3.1 由于超时出错而重试的情况下

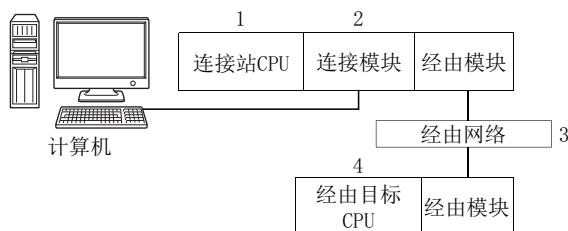
进行通信时发生了超时出错的情况下，有时会在 ACT 控件内部重复进行最多 3 次的超时处理。

此时至发生超时为止将耗费最大相当于设置的超时值 3 倍的时间。

由于超时出错而进行重试的通信路径如下所示。

(1) 串行通信

(a) 构成



(b) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

连接站		3. 经由网络	4. 经由目标 CPU		
1. CPU	2. 连接模块		RCPU	R 运动 CPU	
RCPU	R 系列 C24	CC IE Field	○	×	
		CC IE Control	×	×	
		MELSECNET/H	×	×	
		以太网	○	○	
		串行通信	○	○	
		CC-Link	○	○	
		多点	○	×	

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
RCPU	R 系列 C24	CC IE Field	×	×	×	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×
		多点	×	×	×	×	×	×

连接站		3. 经由网络	4. 经由目标 CPU	
1. CPU	2. 连接模块		RCPU	R 运动 CPU
QCPU (Q 模式)	Q 系列 C24	CC IE Field	×	×
		CC IE Control	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×
		多点	×	×

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU (Q 模式)	Q 系列 C24	CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		以太网	○	×	×	○	×	×
		串行通信	○	×	○	×	×	×
		CC-Link	○	○	○	×	○	○
		多点	○	×	○	×	×	×

连接站		3. 经由网络	4. 经由目标 CPU				
1. CPU	2. 连接模块		RCPU	R 运动 CPU			
LCPU	L 系列 C24	CC IE Field	×	×			
		CC IE Control	×	×			
		MELSECNET/H	×	×			
		以太网	×	×			
		串行通信	×	×			
		CC-Link	×	×			
		多点	×	×			

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	L 系列 C24	CC IE Field	○	×	○	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×
		多点	○	×	○	×	×	×

连接站		3. 经由网络	4. 经由目标 CPU		
1. CPU	2. 连接模块		RCPU	R 运动 CPU	
FXCPU	FX 扩展端口	CC IE Field	×	×	
		CC IE Control	×	×	
		MELSECNET/H	×	×	
		以太网	×	×	
		串行通信	×	×	
		CC-Link	×	×	
		多点	×	×	

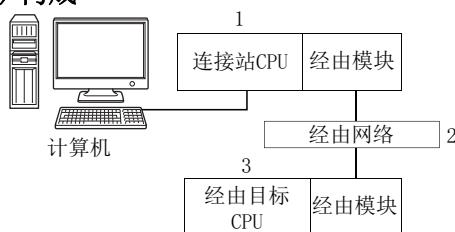
连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
FXCPU	FX 扩展端口	CC IE Field	×	×	×	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	○
		CC-Link	×	×	×	×	×	×
		多点	×	×	×	×	×	○

附

附录 3 关于超时时间
附录 3.1 由于超时出错而重试的情况下

(2) CPU COM 通信

(a) 构成



(b) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
QCPU(Q 模式)	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU(Q 模式)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	○	×	×	○	×	×
	串行通信	○	×	○	×	×	×
	CC-Link	○	○	○	×	○	○

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
LCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	CC IE Field	○	○	○	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	○	×	○	×	×	×
	CC-Link	○	○	○	×	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
Q 运动 CPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
Q 运动 CPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

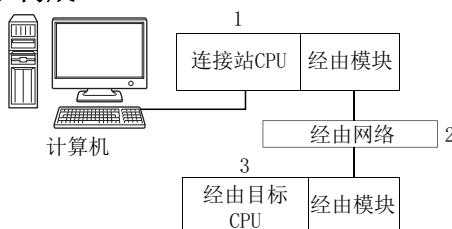
1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU			
		RCPU	R 运动 CPU	LCPU	QSCPU
FXCPU	CC IE Field	×	×		
	CC IE Control	×	×		
	MELSECNET/H	×	×		
	以太网	×	×		
	串行通信	×	×		
	CC-Link	×	×		

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	○ *1

*1：仅 FX3G(C)CPU、FX3U(C)CPU 可以访问。

(3) CPU USB 通信

(a) 构成



(b) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
RCPU	CC IE Field	○	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	○	○
	串行通信	○	○
	CC-Link	○	○

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
RCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
QCPU(Q 模式)	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU(Q 模式)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	○	×	×	○	×	×
	串行通信	○	×	○	×	×	×
	CC-Link	○	○	○	×	○	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
C 语言控制器	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
C 语言控制器	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	○	○	○	×	○	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
LCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	CC IE Field	○	○	○	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	○	×	○	×	×	×
	CC-Link	○	○	○	×	×	×

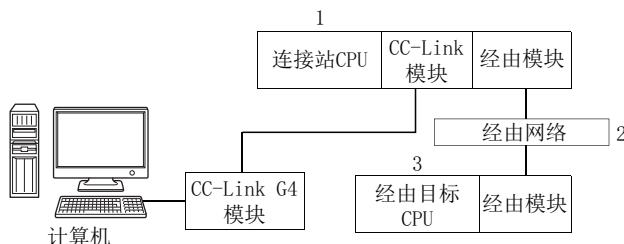
1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
FXCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	○ *1

*1：仅 FX_{3G(C)}CPU、FX_{3U(C)}CPU 可以访问。

(4) CC-Link G4 通信

(a) 构成



(b) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

1) CC-Link G4-S3 模块为 Q 模式, 连接站为 QCPU(Q 模式)、C 语言控制器的情况下

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
QCPU(Q 模式)	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语 言 控 制 器	LCPU	QSCPU	Q 运 动 CPU	FXCPU
QCPU(Q 模式)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	○	×	×	○	×	×
	串行通信	○	×	×	×	×	×
	CC-Link	○	×	×	×	○	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
C 语言控制器	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
C 语言控制器	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

2) CC-Link G4-S3 模块为 Q 模式, 连接站为 LCPU 的情况下

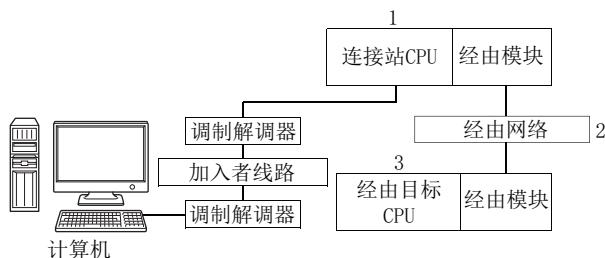
1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
LCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(5) 调制解调器通信

(a) 使用 FXCPU 时

1) 构成



2) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

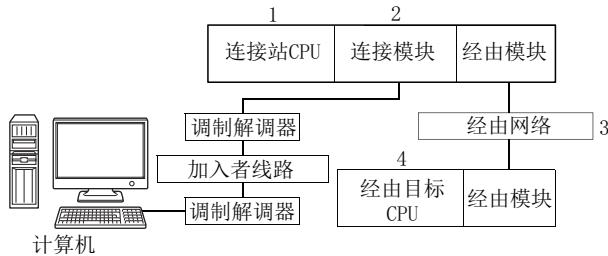
将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
FXCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(b) 使用 Q 系列 C24、QC24N 时

1) 构成



2) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

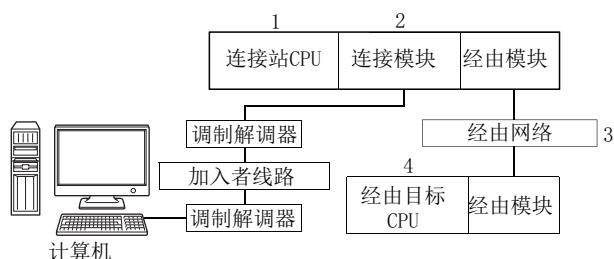
将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

连接站		3. 经由网络	4. 经由目标 CPU	
1. CPU	2. 连接模块		RCPU	R 运动 CPU
QCPU (Q 模式)	Q 系列 C24	CC IE Field	×	×
		CC IE Control	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×
		多点 (独立模式)	×	×

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU (Q 模式)	Q 系列 C24	CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		以太网	○	×	×	○	×	×
		串行通信	○	×	○	×	×	×
		CC-Link	○	○	○	×	○	×
		多点 (独立模式)	○	×	○	×	×	×

(c) 使用 L 系列 C24 时

1) 构成



2) 对象、非对象表

由于超时出错而进行重试的通信路径如下所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

连接站		3. 经由网络	4. 经由目标 CPU						
1. CPU	2. 连接模块		RCPU	R 运动 CPU					
LCPU	L 系列 C24	CC IE Field	×	×					
		CC IE Control	×	×					
		MELSECNET/H	×	×					
		以太网	×	×					
		串行通信	×	×					
		CC-Link	×	×					
		多点（独立模式）	×	×					
连接站		3. 经由网络	4. 经由目标 CPU						
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU	
LCPU	L 系列 C24	CC IE Field	○	○	○	×	×	×	
		CC IE Control	×	×	×	×	×	×	
		MELSECNET/H	×	×	×	×	×	×	
		以太网	×	×	×	×	×	×	
		串行通信	○	×	○	×	×	×	
		CC-Link	○	○	○	×	×	×	
		多点（独立模式）	○	×	○	×	×	×	

附录 3.2 由于接收数据出错而重试的情况下

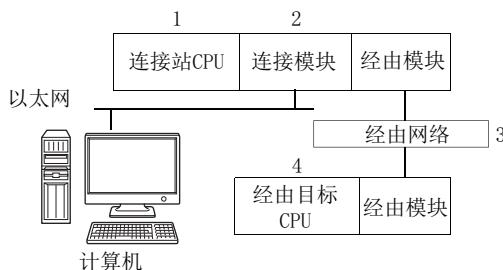
进行通信时发生了接收数据出错的情况下，有时会在 ACT 控件内部重复进行最多 3 次的发送接收重试处理。

此时至函数正常结束或非正常结束为止将耗费最大相当于设置的超时值的 3 倍的时间。

由于接收出错而进行重试的通信路径如下所示。

(1) 以太网通信

(a) 构成



(b) 对象、非对象表

由于接收数据出错而进行重试的通信路径如下表所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○（对象）、×（非对象）表示。

附

连接站		3. 经由网络	4. 经由目标 CPU	
1. CPU	2. 连接模块		RCPU	R 运动 CPU
RCPU	R 系列 E71、RCPU	CC IE Field	○	×
		CC IE Control	×	×
		MELSECNET/H	×	×
		以太网	○	○
		串行通信	○	○
		CC-Link	○	○

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
RCPU	R 系列 E71、RCPU	CC IE Field	×	×	×	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

连接站		3. 经由网络	4. 经由目标 CPU	
1. CPU	2. 连接模块		RCPU	R 运动 CPU
QCPU (Q 模式)	Q 系列 E71、以太网端口内置 QCPU	CC IE Field	×	×
		CC IE Control	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU (Q 模式)	Q 系列 E71、以太网端口内置 QCPU	CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		以太网	○	×	×	○	×	×
		串行通信	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×

连接站		3. 经由网络	4. 经由目标 CPU	
1. CPU	2. 连接模块		RCPU	R 运动 CPU
LCPU	以太网端口内置 LCPU	CC IE Field	×	×
		CC IE Control	×	×
		MELSECNET/H	×	×
		以太网	×	×
		串行通信	×	×
		CC-Link	×	×

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	以太网端口内置 LCPU	CC IE Field	○	○	○	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×

附录 3.3 由于 ACT 控件内部的固定时间发生超时的情况下

在 MX Component 中，执行 Open 函数时进行用于确认计算机与可编程控制器系统之间是否正常连接的通信。

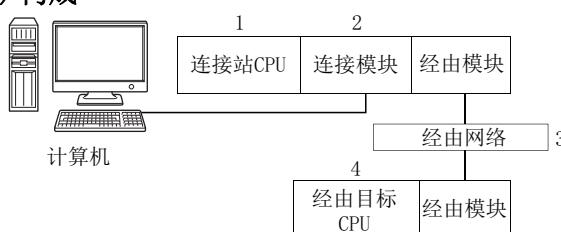
进行上述通信的情况下，使用 ACT 控件内部的固定超时时间 (1000ms ~ 4500ms)。

但是，在进行上述通信期间发生了出错的情况下，有可能发生除超时出错以外的出错。

由于 ACT 控件内部的固定时间而发生超时的通信路径如下所示。

(1) 串行通信

(a) 构成



(b) 对象、非对象表

由于 ACT 控件内部的固定时间而发生超时的通信路径如下表所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

附

连接站		3. 经由网络	4. 经由目标 CPU	
1. CPU	2. 连接模块		RCPU	R 运动 CPU
RCPU	R 系列 C24	CC IE Field	○	×
		CC IE Control	×	×
		MELSECNET/H	×	×
		以太网	○	○
		串行通信	○	○
		CC-Link	○	○
		多点	○	×

连接站		3. 经由网络	4. 经由目标 CPU					
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
RCPU	R 系列 C24	CC IE Field	×	×	×	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		以太网	×	×	×	×	×	×
		串行通信	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×
		多点	×	×	×	×	×	×

连接站		3. 经由网络	4. 经由目标 CPU		
1. CPU	2. 连接模块		RCPU	R 运动 CPU	
QCPU (Q 模式)	Q 系列 C24	CC IE Field	×	×	
		CC IE Control	×	×	
		MELSECNET/H	×	×	
		以太网	×	×	
		串行通信	×	×	
		CC-Link	×	×	
		多点	×	×	

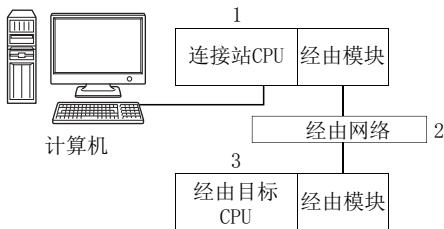
连接站		3. 经由网络	4. 经由目标 CPU				
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU
QCPU (Q 模式)	Q 系列 C24	CC IE Field	○	○	○	○	×
		CC IE Control	○	○	×	○	×
		MELSECNET/H	○	○	×	○	×
		以太网	○	×	×	○	×
		串行通信	○	×	○	×	×
		CC-Link	○	○	○	×	○
		多点	○	×	○	×	×

连接站		3. 经由网络	4. 经由目标 CPU				
1. CPU	2. 连接模块		RCPU	R 运动 CPU			
LCPU	L 系列 C24	CC IE Field	×	×			
		CC IE Control	×	×			
		MELSECNET/H	×	×			
		以太网	×	×			
		串行通信	×	×			
		CC-Link	×	×			
		多点	×	×			

连接站		3. 经由网络	4. 经由目标 CPU				
1. CPU	2. 连接模块		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU
LCPU	L 系列 C24	CC IE Field	○	○	○	×	×
		CC IE Control	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		以太网	×	×	×	×	×
		串行通信	○	×	○	×	×
		CC-Link	○	○	○	×	×
		多点	○	×	○	×	×

(2) CPU COM 通信

(a) 构成



(b) 对象、非对象表

由于 ACT 控件内部的固定时间发生超时的通信路径如下表所示。

连接站 CPU 将全部成为对象。将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
QCPU(Q 模式)、 C 语言控制器	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语 言 控 制 器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU(Q 模式)、 C 语言控制器	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	○	×	×	○	×	×
	串行通信	○	×	○	×	×	×
	CC-Link	○	○	○	×	○	○

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
LCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	CC IE Field	○	○	○	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	○	×	○	×	×	×
	CC-Link	○	○	○	×	×	×

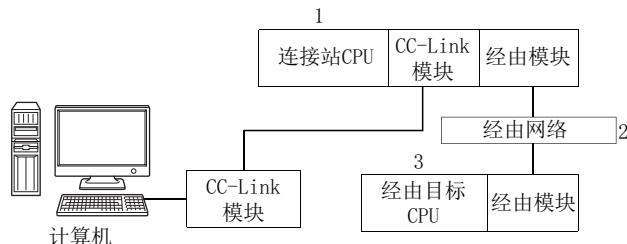
1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
FXCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	○ *1

*1：仅 FX3G(C)CPU、FX3U(C)CPU 可以访问。

(3) CC-Link G4 通信

(a) 构成



(b) 对象、非对象表

由于 ACT 控件内部的固定时间发生超时的通信路径如下表所示。

连接站 CPU 将全部成为对象。

将成为对象的经由目标 CPU 用○(对象)、×(非对象)表示。

通过 Q 模式连接 CC-Link G4-S3。

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
QCPU(Q 模式)	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
QCPU(Q 模式)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	以太网	○	×	×	○	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
C 语言控制器	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
C 语言控制器	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	×	○	×	○	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU	
		RCPU	R 运动 CPU
LCPU	CC IE Field	×	×
	CC IE Control	×	×
	MELSECNET/H	×	×
	以太网	×	×
	串行通信	×	×
	CC-Link	×	×

1. 连接站 CPU	2. 经由网络	3. 经由目标 CPU					
		QCPU (Q 模式)	C 语言 控制器	LCPU	QSCPU	Q 运动 CPU	FXCPU
LCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	以太网	×	×	×	×	×	×
	串行通信	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

修订记录

* 本手册号在封底的左下角。

印刷日期	* 手册编号	修改内容
2013 年 03 月	SH(NA)-081138CHN-A	第一版
2014 年 05 月	SH(NA)-081138CHN-B	第二版
2015 年 02 月	SH(NA)-081138CHN-C	第三版

日文手册原稿 : SH-081083-F

本手册不授予工业产权或任何其它类型的权利，也不授予任何专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业产权的任何问题不承担责任。

备忘录

质保

使用之前请确认以下产品质保的详细说明。

1. 免费质保期限和免费质保范围

在免费质保期内使用本产品时如果出现任何属于三菱电机责任的故障或缺陷（以下称“故障”），则经销商或三菱电机服务公司将负责免费维修。

但是如果需要在国内现场或海外维修时，则要收取派遣工程师的费用。对于涉及到更换故障模块后的任何再试运转、维护或现场测试，三菱电机将不负任何责任。

[免费质保期限]

免费质保期限为自购买日或交货的一年内。

注意产品从三菱电机生产并出货之后，最长分销时间为6个月，生产后最长的免费质保期为18个月。维修零部件的免费质保期不得超过修理前的免费质保期。

[免费质保范围]

(1) 范围局限于按照使用手册、用户手册及产品上的警示标签规定的使用状态、使用方法和使用环境正常使用的情况下。

(2) 以下情况下，即使在免费质保期内，也要收取维修费用。

1. 因不适当存储或搬运、用户过失或疏忽而引起的故障。因用户的硬件或软件设计而导致的故障。

2. 因用户未经批准对产品进行改造而导致的故障等。

3. 对于装有三菱电机产品的用户设备，如果根据现有的法定安全措施或工业标准要求配备必需的功能或结构后本可以避免的故障。

4. 如果正确维护或更换了使用手册中指定的耗材（电池、背光灯、保险丝等）后本可以避免的故障。

5. 因火灾或异常电压等外部因素以及因地震、雷电、大风和水灾等不可抗力而导致的故障。

6. 根据从三菱出货时的科技标准还无法预知的原因而导致的故障。

7. 任何非三菱电机或用户责任而导致的故障。

2. 产品停产后有偿维修期限

(1) 三菱电机在本产品停售后的7年内受理该产品的有偿维修。

停产的消息将以三菱电机技术公告等方式予以通告。

(2) 产品停售后，将不再提供产品（包括维修零件）。

3. 海外服务

在海外，维修由三菱电机在当地的海外FA中心受理。注意各个FA中心的维修条件可能会不同。

4. 意外损失和间接损失不在质保责任范围内

无论是否在免费质保期内，对于任何非三菱电机责任的原因而导致的损失、机会损失、因三菱电机产品故障而引起的用户利润损失、无论能否预测的特殊损失和间接损失、事故赔偿、除三菱电机以外产品的损失赔偿、用户更换设备、现场机械设备的再调试、运行测试及其它作业等，三菱电机将不承担责任。

5. 产品规格的改变

目录、手册或技术文档中的规格如有改变，恕不另行通知。

Microsoft、Windows、Windows Vista、Windows XP、Access、Excel、Visual Basic、Visual C++、Visual C#、Visual Studio 是美国 Microsoft Corporation 在美国及其它国家的商标或注册商标。

Ethernet 是美国 Xerox Corporation 的商标或注册商标。

本手册中使用的其它公司名和产品名是相应公司的商标或注册商标。

SPREAD

Copyright (C) 2004 FarPoint Technologies, Inc.

SH(NA)-081138CHN-C(1502)MEACH

MODEL: SW4DNC-ACT-P-C

三菱电机自动化(中国)有限公司

地址：上海市虹桥路1386号三菱电机自动化中心

邮编：200336

电话：021-23223030 传真：021-23223000

网址：<http://cn.MitsubishiElectric.com/fa/zh/>

技术支持热线 **400-821-3030**



扫描二维码,关注官方微博



扫描二维码,关注官方微信

内容如有更改 恕不另行通知