

三菱可编程控制器

SLMP参考手册

安全注意事项

(使用之前务必阅读)

使用本产品前，请仔细阅读本手册，同时在充分注意安全的前提下正确地操作。

在·安全注意事项·中，安全注意事项被分为“警告”和“注意”这二个等级。

 警告	表示错误操作可能造成危险后果，导致死亡或重伤事故。
 注意	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。

注意根据情况不同，即使“注意”这一级别的事项也有可能引发严重后果。

对两级注意事项都须遵照执行，因为它们对于操作人员安全是至关重要的。

请妥善保管本手册以备需要时查阅，并应将本手册交给最终用户。

[设计注意事项]

警告

- 将外部设备连接到SLMP对应设备上，对运行中的可编程控制器进行控制(数据更改)时，应在程序中配置互锁电路，确保整个系统始终都会安全运行。此外，在对运行中的可编程控制器执行其它控制(程序更改、参数更改、强制输出、运行状态更改(状态控制))时，应仔细阅读手册并充分确认安全之后再进行操作。如果未认真确认，由于操作错误可能导致机械损坏或引发事故。
- 从外部设备对远程的可编程控制器进行控制时，由于数据通讯异常，可能不能对可编程控制器的故障立即采取措施。应在程序中配置互锁电路的同时，预先在外部设备与可编程控制器之间确定发生数据通信异常时系统方面的处理方法。
- 在SLMP对应设备及智能功能模块的缓冲存储器中，请勿对“系统区域”或“禁止写入区域”进行数据写入。此外，在对SLMP对应设备及智能功能模块的输出信号之中，请勿对“禁止使用”的信号进行输出(ON)操作。若对“系统区域”或“禁止写入区域”进行数据写入，或对“禁止使用”的信号进行输出，有可能导致可编程控制器系统误动作。关于“系统区域”或“禁止写入区域”、“禁止使用”的信号详细内容，请参阅各模块的用户手册。

[启动·维护注意事项]

注意

- 将外部设备连接到运行中的SLMP对应设备、其它站的CPU模式上进行的在线操作(程序更改、参数更改、强制输出、运行状态更改(状态控制))时，应仔细阅读手册并充分确认安全之后再进行操作。如果未认真确认，由于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生产的通用产品。

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、违法行为责任、生产物责任），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。

前言

在此非常感谢贵方购买了三菱可编程控制器产品。

本手册是用于让用户了解使用SLMP(SeamLess Message Protocol)时的必要对应设备、访问范围、通信步骤、报文格式有关内容的手册。

在使用之前应熟读本手册，在充分了解SLMP的基础上正确地使用本产品。

将本手册中介绍的程序示例应用于实际系统的情况下，应充分验证对象系统中不存在控制方面的问题。

目录

安全注意事项	1
关于产品的应用	2
前言	3
术语	6
第1章 SLMP的作用	7
第2章 规格	9
2.1 SLMP规格	9
2.2 SLMP对应设备	9
2.3 访问范围及可其它站访问的模块	10
第3章 SLMP通信步骤	11
3.1 使用TCP/IP时	11
3.2 使用UDP/IP时	12
3.3 注意事项	13
第4章 报文格式	15
4.1 请求报文	15
4.2 响应报文	20
第5章 指令	23
5.1 指令一览	24
5.2 Device(软元件访问)	29
指令内指定的数据	29
Read(指令: 0401)	39
Write(指令: 1401)	43
Read Random(指令: 0403)	46
Write Random(指令: 1402)	50
Entry Monitor Device(指令: 0801)	55
Execute Monitor(指令: 0802)	59
Read Block(指令: 0406)	62
Write Block(指令: 1406)	66
5.3 Label(标签访问)	70
指令内指定的数据	70
Array Label Read(指令: 041A)	78
Array Label Write(指令: 141A)	87
Label Read Random(指令: 041C)	96
Label Write Random(指令: 141B)	102
5.4 Memory(本站缓冲存储器访问)	108
指令内指定的数据	108
Read(指令: 0613)	110
Write(指令: 1613)	112
5.5 Extend Unit(智能功能模块的缓冲存储器访问)	113
指令内指定的数据	114
Read(指令: 0601)	116
Write(指令: 1601)	118

5.6 Remote Control(远程操作)	120
在远程操作之前	120
Remote Run(指令: 1001)	121
Remote Stop(指令: 1002)	123
Remote Pause(指令: 1003)	124
Remote Latch Clear(指令: 1005)	125
Remote Reset(指令: 1006)	126
Read Type Name(指令: 0101)	127
5.7 Remote Password(远程口令)	130
指令内指定的数据	130
Lock(指令: 1631)	131
Unlock(指令: 1630)	133
5.8 File(文件控制)	135
指令内指定的数据	136
执行步骤	140
注意事项	144
Read Directory/File(指令: 1810)	145
Search Directory/File(指令: 1811)	154
New File(指令: 1820)	157
Delete File(指令: 1822)	160
Copy File(指令: 1824)	163
Change File State(指令: 1825)	167
Change File Date(指令: 1826)	170
Open File(指令: 1827)	173
Read File(指令: 1828)	176
Write File(指令: 1829)	179
Close File(指令: 182A)	182
5.9 Self Test(反复测试)(指令: 0619)	184
5.10 Clear Error(出错代码初始化、LED熄灯)(指令: 1617)	186
5.11 Ondemand(指令: 2101)	187
第6章 故障排除	188
附录	190
附1 通过软元件的扩展指定进行读取、写入	190
至链接直接软元件的访问	190
至模块访问软元件的访问	194
至CPU缓冲存储器访问软元件的访问	197
通过变址寄存器间接指定网络No.及起始输入输出编号的访问	200
通过变址寄存器或长变址寄存器间接指定软元件编号的访问	205
通过字软元件中存储的值间接指定软元件编号的访问	211
附2 MC协议与SLMP的指令对应表	215
附3 访问目标为多CPU系统的情况下	217
索引	220
修订记录	222
质保	223
商标	224

术语

本手册中除了特别标明的情况外，将使用下述术语进行说明。

术语	内容
SLMP	是SeamLess Message Protocol的略称。 是用于通过外部设备对SLMP对应设备及SLMP对应设备上连接的可编程控制器进行访问的协议。
SLMP对应设备	是可发送接收SLMP报文的三菱产品设备的总称。(以太网适配器模块、以太网搭载模块)
外部设备	是对于SLMP对应设备，发送SLMP请求报文的设备的总称。(个人计算机、显示器等)
MC协议	是MELSEC通信协议的略称。 是用于通过外部设备对MC协议对应设备及MC协议对应设备上连接的可编程控制器进行访问的协议。
本站	本站表示与外部设备直接连接的站。
其它站	其它站表示与本站相连接的网络上的站。 The diagram illustrates a network topology. On the left, under the label '外部设备' (External Device), there is a computer icon consisting of a monitor, keyboard, and tower unit. A line connects this computer to a circle labeled '本站' (This Station). This circle is then connected to a larger oval labeled '网络' (Network). Inside the '网络' oval, there are two smaller circles, each labeled '其它站' (Other Station), representing stations connected to the network.
请求报文	是通过外部设备对于SLMP对应设备发送的处理请求的报文。
响应报文	是对于请求报文，SLMP对应设备发送到外部设备的处理结果的报文。
工程工具	是MELSEC可编程控制器软件包的别称。
CC-Link IE现场网络	是使用了以太网(1000BASE-T)的高速且大容量的开放现场网络。
CC-Link IE现场网络主站·本地站搭载模块	是RJ71GF11-T2、QJ71GF11-T2、LJ71GF11-T2型CC-Link IE现场网络主站·本地站模块及RJ71EN71(使用CC-Link IE现场网络功能时)的总称。
CC-Link IE现场网络起始模块	是LJ72GF15-T2型CC-Link IE现场网络起始模块的略称。
CC-Link IE控制网络搭载模块	是RJ71GP21-SX、QJ71GP21-SX、QJ71GP21S-SX型CC-Link IE控制网络模块的略称。
以太网搭载模块	是RJ71EN71(使用以太网功能时)、QJ71E71-100、LJ71E71-100型以太网接口模块以及MELSEC iQ-R 系列CPU模块(使用以太网功能时)，以太网端口内置CPU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 端口内置CPU	是Q03UDVCP、Q04UDVCP、Q04UDPVCPU、Q06UDVCP、Q06UDPVCPU、Q13UDVCP、Q13UDPVCPU、Q26UDVCP、Q26UDPVCPU、L02CPU、L02CPU-P、L06CPU、L06CPU-P、L26CPU、L26CPU-P、L26CPU-BT、L26CPU-PBT的总称。
中继站	是将多个网络模块安装到1个可编程控制器中，对至其它网络的数据链接进行中继的站。
软元件	是SLMP对应设备及其它站的CPU模块内部具有的软元件(X/Y/W等)。
链接软元件	是CC-Link IE现场网络模块内部具有的软元件(RX/Ry/RWr/RWw)。
CPU模块	是MELSEC iQ-R系列、MELSEC-Q系列及MELSEC-L系列CPU模块的总称。
RCPU	是MELSEC iQ-R系列CPU模块的总称。
QCPU	是MELSEC-Q系列CPU模块的总称。
模块访问软元件	是MELSEC iQ-R系列的模块访问软元件、MELSEC Q/L系列的智能功能模块软元件的总称。
智能功能模块	是A/D、D/A转换模块等具有输入输出以外功能的MELSEC iQ-R系列、MELSEC-Q/L系列模块的总称。
缓冲存储器	是用于储存设置值、监视值等SLMP对应设备及智能功能模块的存储器。

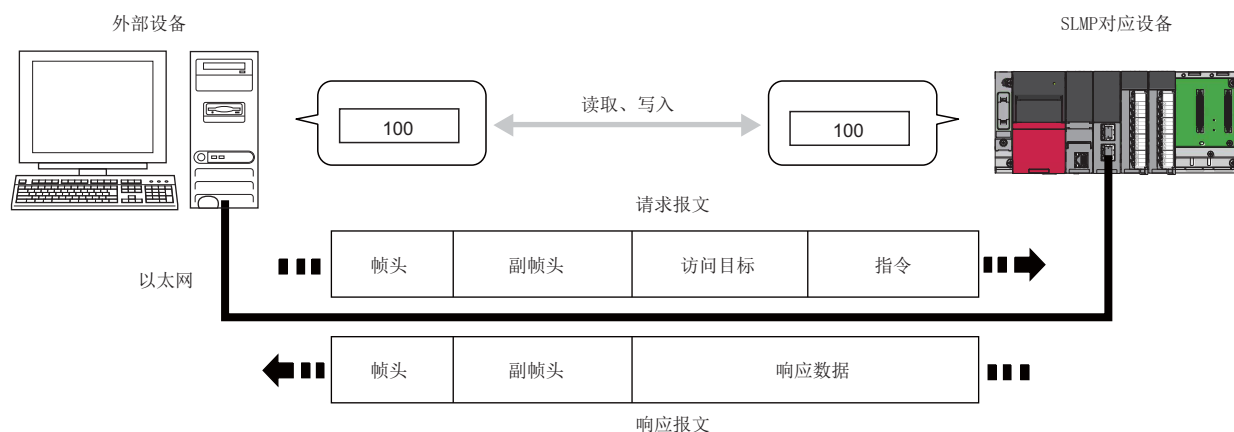
1 SLMP的作用

1

SLMP是用于外部设备使用以太网对SLMP对应设备进行访问的协议。如果是通过SLMP的控制步骤可进行报文的发送接收的设备，则可以通过SLMP进行通信。(个人计算机及显示器等)

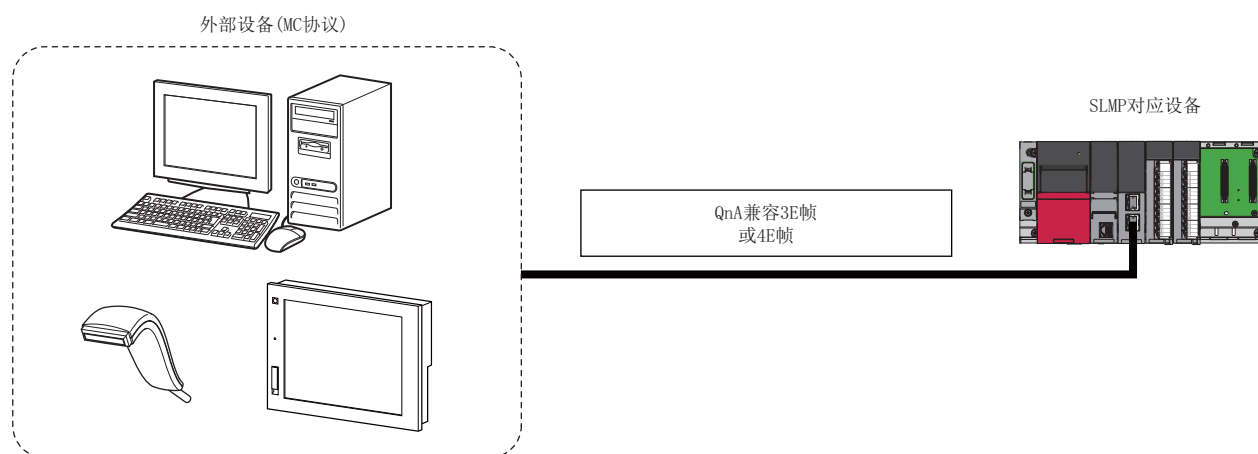
通过外部设备监视系统

通过用以太网连接的外部设备以SLMP的报文格式发送请求报文，由于可以进行软元件读取，因此可以监视系统。此外不仅可以进行软元件的读取，也可以进行软元件的写入及SLMP对应设备的复位等。(23页 指令)



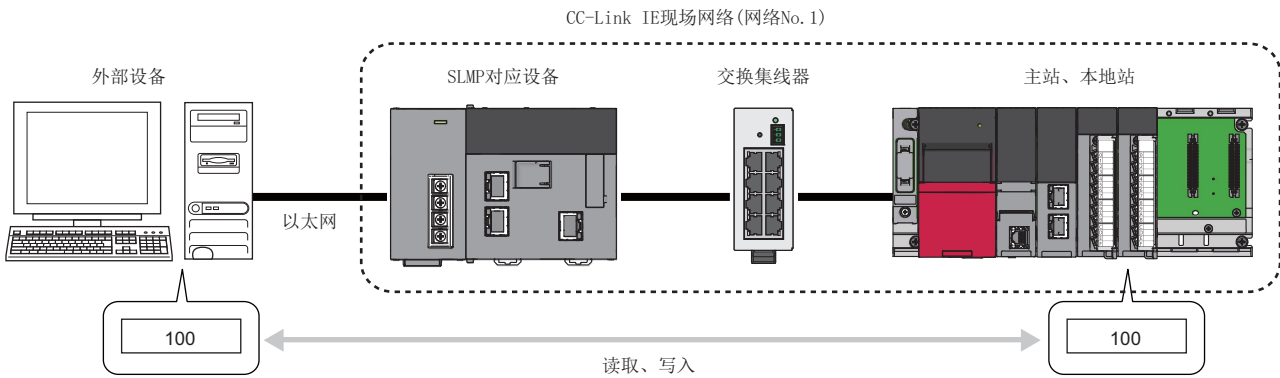
通过MC协议连接使用的外部设备

SLMP是与MC协议的QnA系列3E帧以及4E帧相同的报文格式。可以通过MC协议将使用的外部设备原样不变地连接到SLMP对应设备上。(215页 MC协议与SLMP的指令对应表)



可进行经由了网络的访问

如果使用SLMP，则也可以通过外部设备经由SLMP对应设备，对同一网络与其它网络模块进行无缝访问。(10页 访问范围及可其它站访问的模块)



可通过通信协议支持功能容易地执行SLMP

通过使用工程工具的通信协议支持功能，可以容易地进行SLMP通信。
与通过外部设备进行SLMP通信时相同，可以通过以太网搭载模块控制SLMP对应设备。



2 规格

本章对SLMP对应设备及SLMP的通信规格有关内容进行说明。

2.1 SLMP规格

通过外部设备，或通信协议支持功能发送的SLMP规格如下所示。

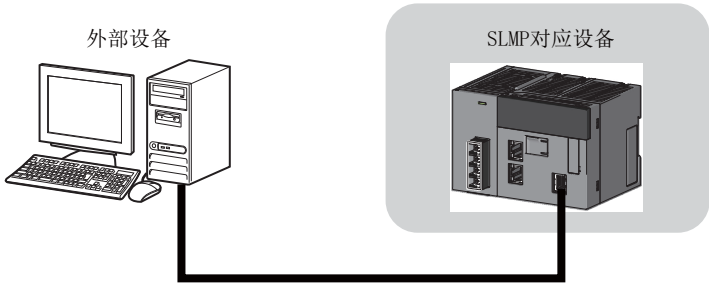
项目	通信数据代码	内容	浏览目标
SLMP	• ASCII代码 • 二进制代码	是与MC协议的QnA系列3E帧以及4E帧相同的报文格式。	15页 请求报文

要点

通过二进制代码进行通信与通过ASCII代码进行通信相比，由于通信数据量变为一半左右，因此通信时间被缩短。

2.2 SLMP对应设备

所使用模块是否对应于SLMP，请参阅所使用模块的手册。



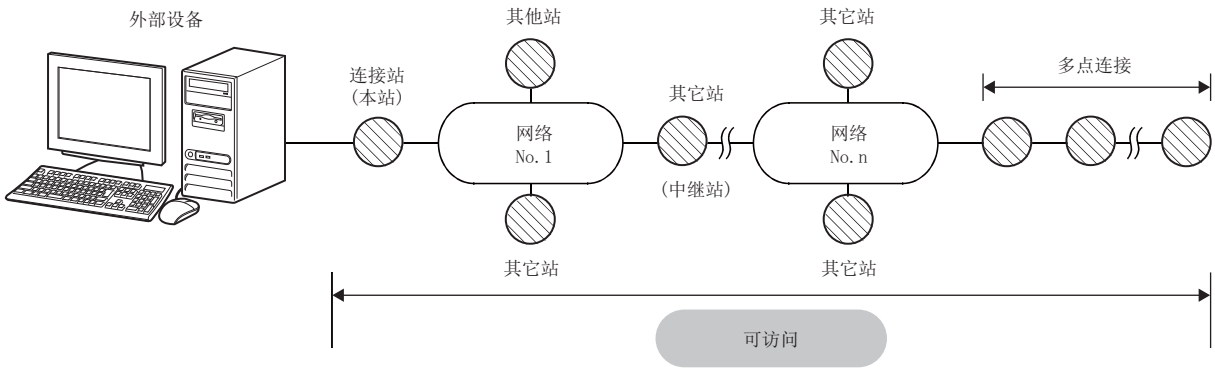
2.3 访问范围及其它站访问的模块

访问范围

可以访问下述设备。

- 与外部设备直接连接的SLMP对应设备(本站)
- 与SLMP对应设备同一网络上的其它站
- 与SLMP对应设备同一网络上的其它站上连接的其它网络的其它站*1

*1 可以访问设置网络No.、站号的其它站，以及多点连接的串行通信模块。



要点

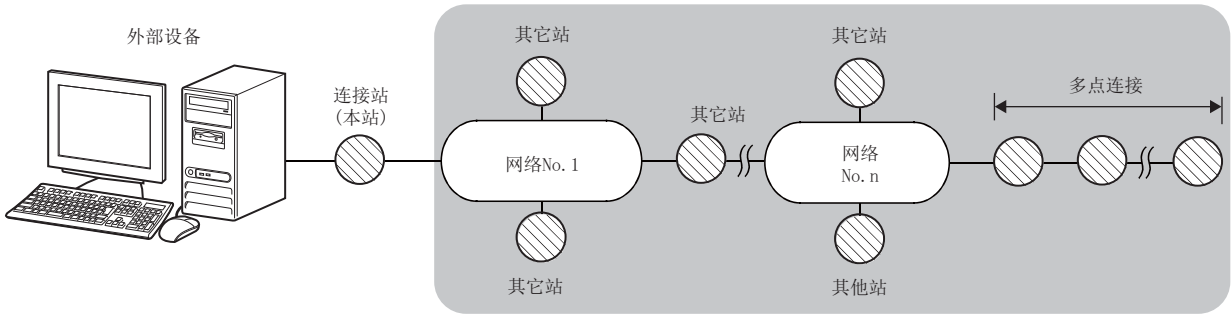
可以访问下述网络。

- 以太网(需要设置网络No.、站号。)
- CC-Link IE 控制网络
- CC-Link IE 现场网络
- MELSECNET/H

可访问范围，最多为8个网络目标(中继站数：7站)。

可其它站访问的模块

访问其它站时的可访问的模块如下所示。



■CPU模块

可以访问SLMP对应设备对应的网络内的CPU模块。(所使用的各网络模块用户手册)

■CPU模块以外

可以访问下述模块。

- SLMP对应设备
- CC-Link IE现场网络主站・本地站搭载模块
- CC-Link IE现场网络起始模块
- CC-Link IE控制网络搭载模块
- 多点连接的串行通信模块

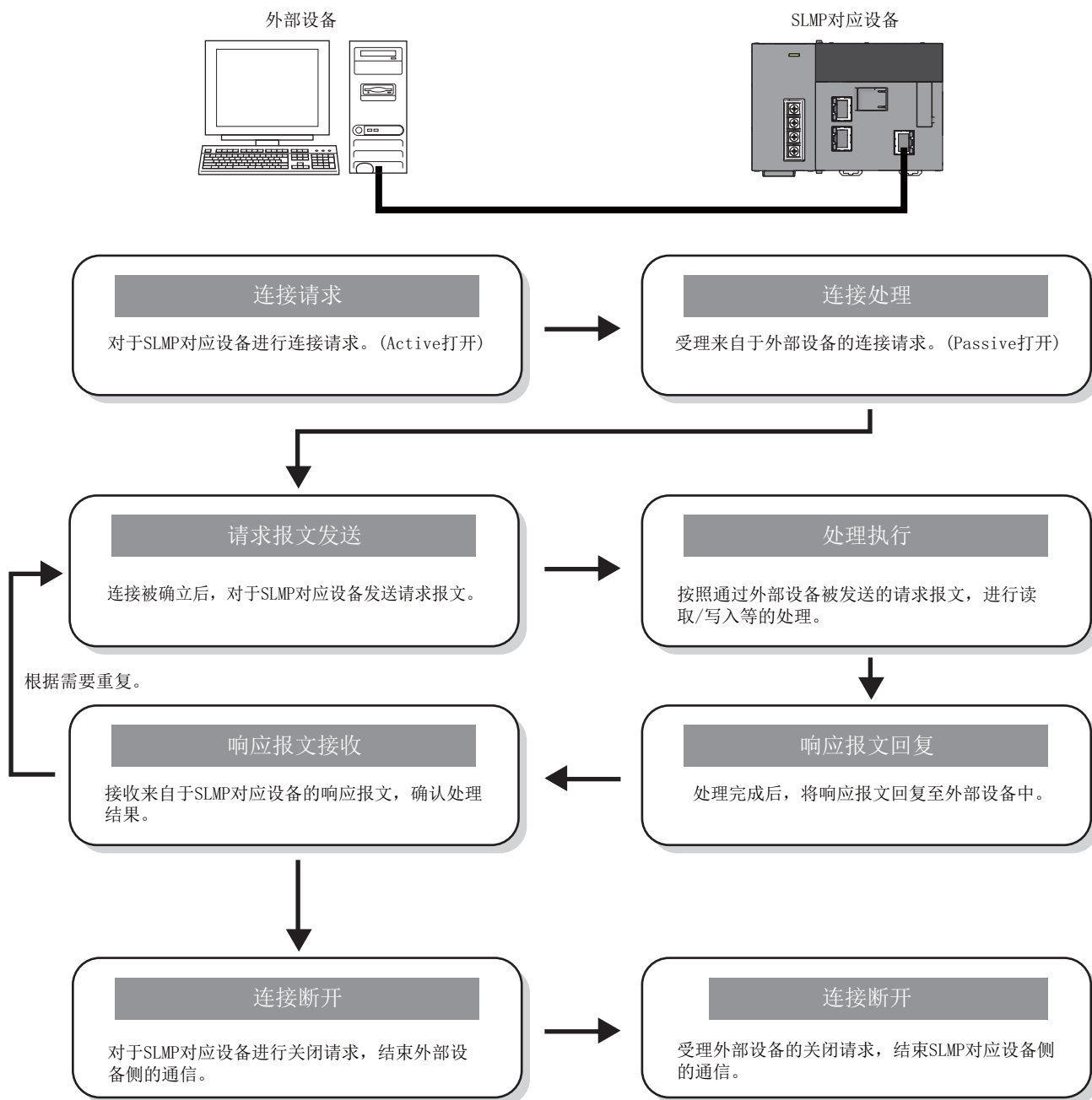
3 SLMP通信步骤

外部设备与SLMP对应设备，按照下述中所示的步骤进行通信。

3.1 使用TCP/IP时

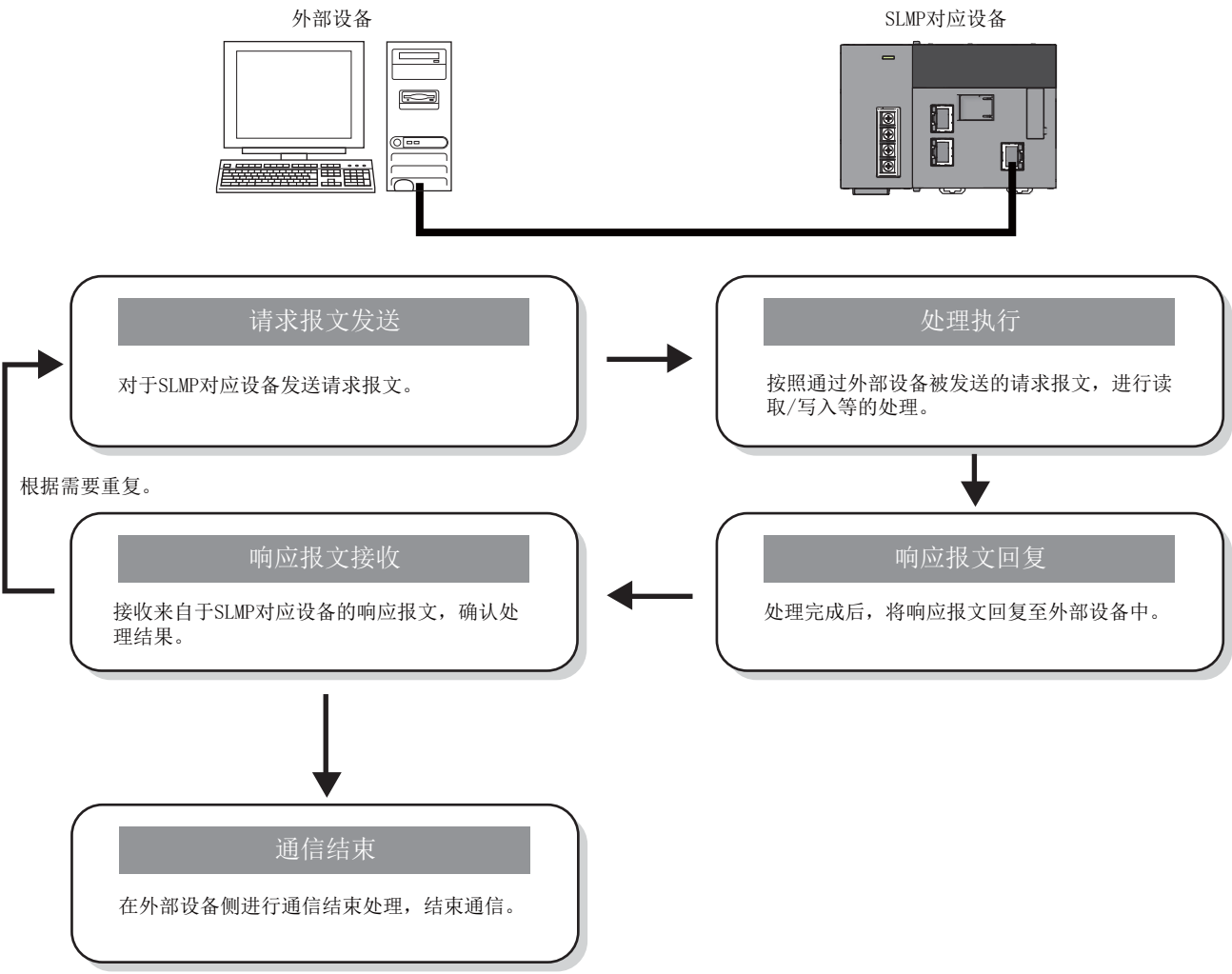
以TCP/IP进行SLMP通信时的通信步骤如下所示。

对于TCP/IP，由于在通信时确立连接后，确认数据正常传送到通信对象的同时进行通信，因此数据的可靠性被确保。但是与UDP/IP相比，线路负荷将变大。



3.2 使用UDP/IP时

以UDP/IP进行SLMP通信时的通信步骤如下所示。
对于UDP/IP由于在通信时未确立连接，也不确认数据正常传送到通信对象，因此线路负荷将变低。但是与TCP/IP相比，数据的可靠性将变低。



3.3 注意事项

请求报文的发送

外部设备应在对SLMP对应设备处于可接收请求报文的状态进行确认之后，再发送请求报文。

发送多个请求报文的情况下

应在外部设备侧中将“序列号”添加到请求报文的副帧头上，再进行发送。通过添加“序列号”，发送了多个请求报文的情况下也可以在外部设备侧判别响应报文的发送源。(15页 请求报文)

连续发送请求报文的情况下

在不等待响应报文接收的状况下连续发送添加了“序列号”的请求报文的情况下，应在外部设备不超出下述指令数的状况下发送请求报文。

SLMP对应设备		TCP/UDP	1个连接中可处理的指令数 ^{*1}
产品名称	型号		
以太网适配器模块	NZ2GF-ETB	TCP	1+(50÷使用的连接数)
		UDP	
以太网搭载模块	RJ71EN71	TCP	(TCP窗口容量(11680字节)÷SLMP报文容量(字节))
		UDP	1+(接收缓冲中可容纳的报文数(190)÷使用的连接数)
	QJ71E71-100、 LJ71E71-100	TCP	(6144÷SLMP报文容量(字节))
		UDP	1+(57÷使用的连接数)
	RCPU	TCP	(TCP窗口容量(11680字节)÷SLMP报文容量(字节))
		UDP	1+接收缓冲中可容纳的报文数(110)

*1 计算结果为小数的情况下，将舍去小数点以后置为整数。

超出指令数时，将变为下述。变为了下述的情况下，应降低发送请求报文的频率。

- TCP/IP的情况下，SLMP对应设备的接收缓冲中将变为无空余。由于窗口容量为0，因此外部设备在SLMP对应设备的接收缓冲中可为空余之前，可暂时停止请求报文的发送。
- UDP/IP的情况下，SLMP对应设备中有可能发生出错，或通过SLMP对应设备无法返回响应报文。

未返回对于请求报文的响应报文的情况下

通过SLMP对应设备未返回响应报文的情况下，应在经过请求报文的“监视定时器”中设置的时间后，通过外部设备再次发送请求报文。

关于SLMP对应设备的更换

故障等导致更换外部设备及SLMP对应设备时，由于连接设备的MAC地址发生变化，因此有可能无法进行通信。(更换为相同IP地址的设备的情况下)

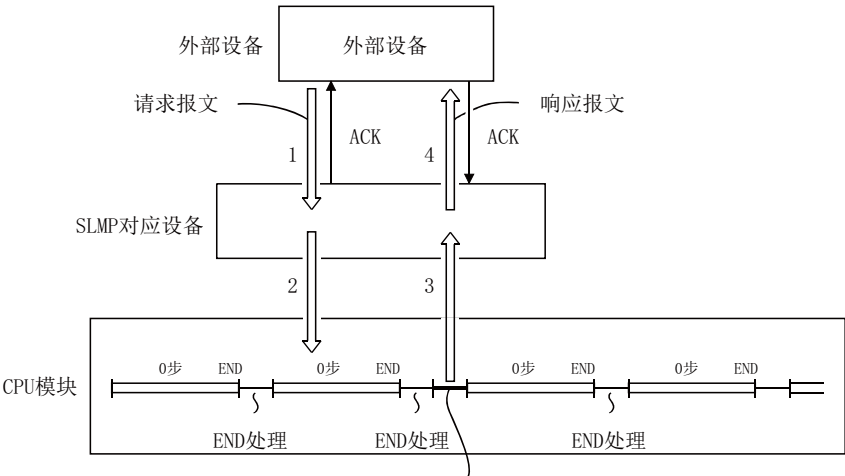
更换了以太网内的设备的情况下，应重新启动网络内的全部设备。

访问CPU模块的情况下

通过外部设备经由SLMP对应设备访问CPU模块时的注意事项如下所示。

■CPU模块侧的处理时机

在CPU模块的END处理时，执行对于请求报文的处理。



对于来自外部设备的请求报文的处理

- 1. 通过外部设备将请求报文发送至SLMP对应设备中。
- 2. 如果SLMP对应设备接收来自于外部设备的请求报文，则按照请求报文的内容，对CPU模块进行数据的读取请求或写入请求。
- 3. CPU模块在程序的END处理时通过外部设备按照请求的内容，进行数据的读取或写入，将处理结果返回SLMP对应设备中。
- 4. SLMP对应设备通过CPU模块收取处理结果时，将包括处理结果的响应报文发送至外部设备中。

■CPU模块为RUN中的读取或写入

- 由于对于来自于外部设备的请求进行处理，CPU模块的扫描时间将会延长，应加以注意。扫描时间的延长影响控制的情况下，应以较少的点数分数次进行访问。
- 进行写入的情况下，应确认在CPU模块侧变为允许RUN中写入。(是否解除系统保护等)

■对访问目标的CPU模块添加系统保护时

在访问目标中将发生出错，异常响应被返回外部设备中。应在解除CPU模块侧的系统保护后，再次发送请求报文。

■多个外部设备对同一站同时发生了访问请求时

根据请求时机，进行多次END处理之前有可能通过外部设备等待被请求的处理。通过下述方法之一，可以在1个扫描以内处理多个请求。

- 通过程序执行COM指令。
- 使用工程工具的“服务处理设置”，将服务处理设置确保在1~100ms。

4 报文格式

本章中对SLMP的报文格式有关内容进行说明。

4.1 请求报文

通过外部设备发送至SLMP对应设备的请求报文的格式如下所示。

帧头	副帧头	请求目标网络编号	请求目标站号	请求目标模块I/O编号	请求目标多点站号	请求数据长	监视定时器	请求数据	页脚
----	-----	----------	--------	-------------	----------	-------	-------	------	----

帧头

是TCP/IP及UDP/IP用的帧头。帧头在外部设备侧进行添加发送。此外，通常根据外部设备自动被添加。

副帧头

在是添加序列号，还是不添加中有所不同。

序列号是指，在外部设备添加的报文认识用的任意编号。如果添加序列号后发送请求报文，则响应报文中也添加相同的序列号。在通过外部设备将多个请求报文发送至SLMP对应设备时使用。

请求报文中添加序列号的情况下(序列号为1234H的情况下)		请求报文中未添加序列号的情况下																																													
ASCII代码时	<table><tr><td colspan="4">(固定值)</td><td colspan="4">(固定值)</td></tr><tr><td>5</td><td>4</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="4">35H, 34H, 30H, 30H</td><td colspan="4">31H, 32H, 33H, 34H</td><td colspan="4">30H, 30H, 30H, 30H</td></tr></table> 序列号	(固定值)				(固定值)				5	4	0	0	1	2	3	4	0	0	0	0	35H, 34H, 30H, 30H				31H, 32H, 33H, 34H				30H, 30H, 30H, 30H				ASCII代码时	<table><tr><td colspan="4">(固定值)</td></tr><tr><td>5</td><td>0</td><td>0</td><td>0</td></tr><tr><td colspan="4">35H, 30H, 30H, 30H</td></tr></table>	(固定值)				5	0	0	0	35H, 30H, 30H, 30H			
(固定值)				(固定值)																																											
5	4	0	0	1	2	3	4	0	0	0	0																																				
35H, 34H, 30H, 30H				31H, 32H, 33H, 34H				30H, 30H, 30H, 30H																																							
(固定值)																																															
5	0	0	0																																												
35H, 30H, 30H, 30H																																															
二进制代码时	<table><tr><td colspan="2">(固定值)</td><td colspan="2">(固定值)</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td colspan="2">54H, 00H</td><td colspan="2">34H, 12H, 00H, 00H</td></tr></table> 序列号	(固定值)		(固定值)						54H, 00H		34H, 12H, 00H, 00H		二进制代码时	<table><tr><td colspan="2">(固定值)</td></tr><tr><td></td><td></td></tr><tr><td colspan="2">50H, 00H</td></tr></table>	(固定值)				50H, 00H																											
(固定值)		(固定值)																																													
54H, 00H		34H, 12H, 00H, 00H																																													
(固定值)																																															
50H, 00H																																															

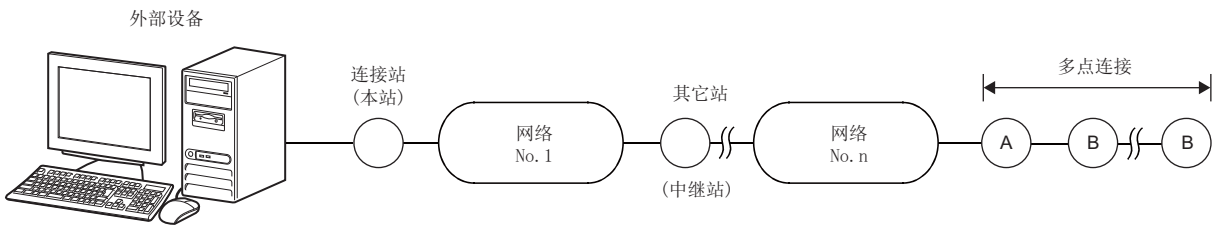
要点

- 序列号应在外部设备侧管理并使用。
- 以ASCII代码发送时，按照从高位字节到低位字节的顺序存储序列号。
- 以二进制代码发送时，按照从低位字节到高位字节的顺序存储序列号。

请求目标网络编号、请求目标站号

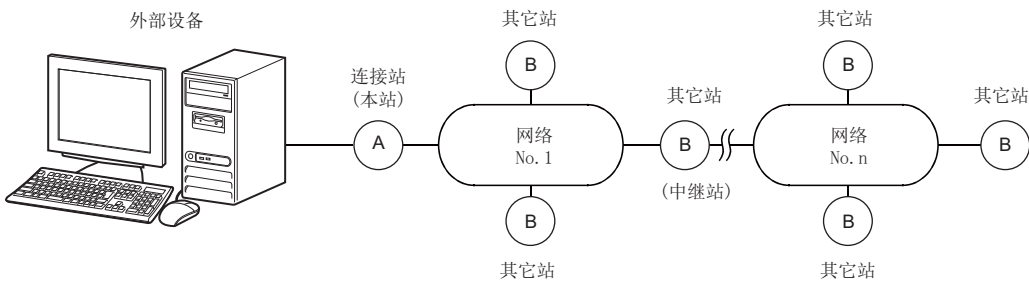
指定访问目标的网络No.、站号。网络No.、站号以16进制数进行指定。
请求目标网络编号、请求目标站号，按照从高位字节到低位字节的顺序进行发送。

■访问目标为多点连接站的情况下



访问目标	请求目标网络编号	请求目标站号
B(多点连接站)	与A(对多点连接与网络进行中继的站)相连接的网络No. n的网络No.	A(对多点连接与网络进行中继的站)的网络模块站号

■访问目标为多点连接站以外的情况下



访问目标	请求目标网络编号	请求目标站号
A(连接站)	00H	FFH
B(其它站)	01H~EFH(1~239) 对于网络No.为240~255的站，不可以访问。	01H~78H(1~120): 站号 7DH(125): 指定管理站/主站*1 7EH(126): 当前管理站/主站*2

*1 7DH(125): 指定管理站/主站访问通过参数设置为管理站/主站的站。
*2 7EH(126): 当前管理站/主站访问实际上作为管理站/主站进行动作的站。

例

请求目标网络编号中指定1AH(26)的情况下

ASCII代码时
1 A
31H, 41H

二进制代码时
1AH

请求目标站号中指定1AH(26)的情况下

ASCII代码时
1 A
31H, 41H

二进制代码时
1AH

请求目标模块I/O编号

指定访问目标的模块。

访问目标	请求目标模块I/O编号
本站	03FFH
管理CPU	03FFH
多CPU1号机	03E0H
多CPU2号机	03E1H
多CPU3号机	03E2H
多CPU4号机	03E3H
经由多点连接的CPU模块的多点连接站	0000H~01FFH

要点

经由多点连接的CPU模块的情况下，应对多点连接源的串行通信模块的I/O 编号通过以16除的值以4 位数(16进制数)进行指定。

例

请求目标模块I/O编号中指定03FFH的情况下

ASCII代码时

0	3	F	F
30H, 33H, 46H, 46H			

二进制代码时

FFH, 03H

■以ASCII代码进行数据通信时

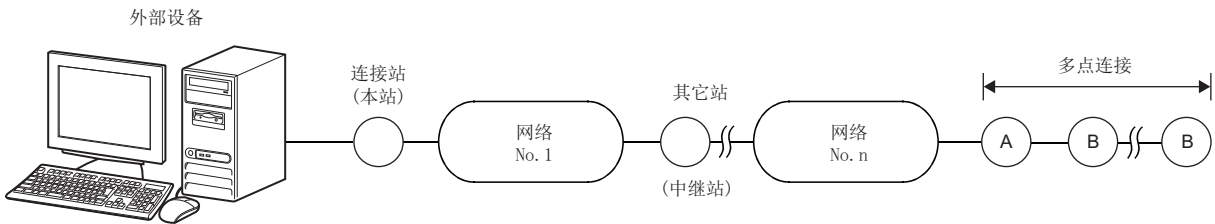
按照从高位字节到低位字节的顺序进行发送。

■以二进制代码进行数据通信时

按照从低位字节到高位字节的顺序进行发送。

请求目标多点站号

访问目标为多点连接站的情况下进行指定。



对象设备的访问目标	请求目标多点站号
B(多点连接站)	00H~1FH(0~31): 局番
A(对多点连接与网络进行中继的站)	00H
不是多点连接站的站	00H

例

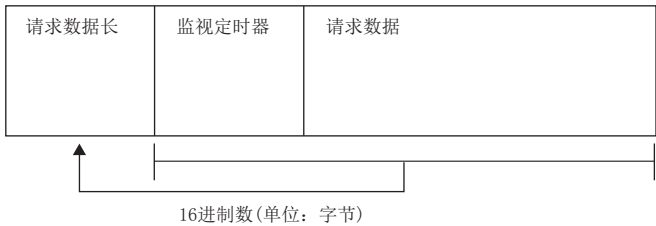
请求目标多点连接站号中指定0的情况下

ASCII代码时
0 0
30H, 30H

二进制代码时
00H

请求数据长

以16进制数指定从监视定时器开始到请求数据为止的数据长。(单位: 字节)



例

请求数据长为24字节的情况下

ASCII代码时
0 0 1 8
30H, 30H, 31H, 38H

二进制代码时
18H, 00H

■以ASCII代码进行数据通信时

按照从高位字节到低位字节的顺序进行发送。

■以二进制代码进行数据通信时

按照从低位字节到高位字节的顺序进行发送。

监视定时器

是对通过外部设备接收了请求报文的SLMP对应设备向访问目标请求处理之后到返回响应为止的等待时间进行设置的定时器。

- 0000H(0)：无限等待(在处理完成之前继续等。)
- 0001H～FFFFH(1～65535)：等待时间(单位：250ms)

为了进行正常的数据通信，建议在通信目标的下表设置范围内被使用。

访问目标	监视定时器
本站	01H～28H(0.25秒～10秒)
其它站	02H～F0H(0.5秒～60秒)

例

监视定时器中指定10H的情况下

ASCII代码时	0010
	30H, 30H, 31H, 30H

二进制代码时	1000
	10H, 00H

■以ASCII代码进行数据通信时

按照从高位字节到低位字节的顺序进行发送。

■以二进制代码进行数据通信时

按照从低位字节到高位字节的顺序进行发送。

请求数据

指定表示请求内容的指令、子指令、数据。(23页 指令)

页脚

是TCP/IP及UDP/IP用的页脚。页脚在外部设备侧进行添加发送。此外，通常根据外部设备自动被添加。

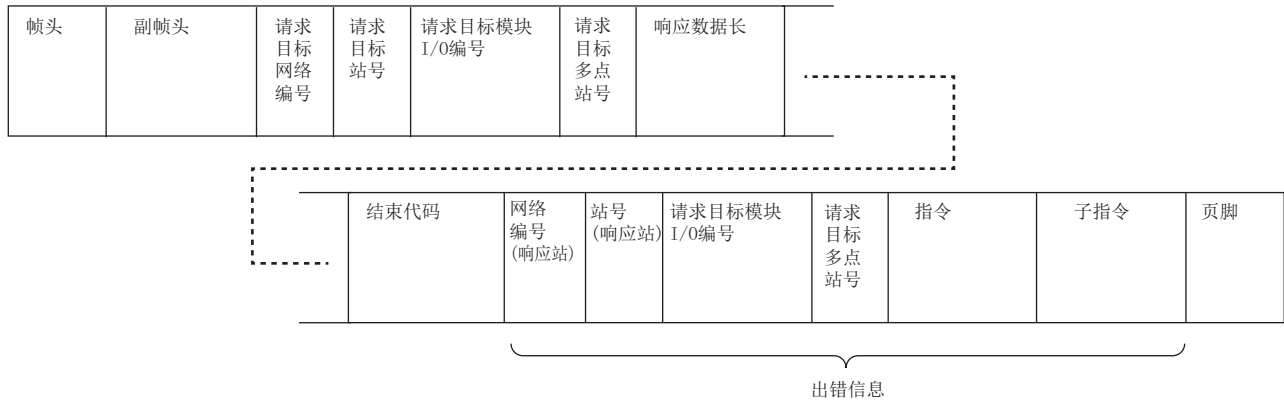
4.2 响应报文

通过SLMP对应设备发送至外部设备的响应报文的格式如下所示。

(正常结束时)

帧头	副帧头	请求目标网络编号	请求目标站号	请求目标模块I/O编号	请求目标多点站号	响应数据长	结束代码	响应数据	页脚
----	-----	----------	--------	-------------	----------	-------	------	------	----

(异常结束时)



下述项目存储与请求报文相同的内容。(15页 请求报文)

- 请求目标网络编号
- 请求目标站号
- 请求目标模块I/O编号
- 请求目标多点站号

帧头

存储以太网的帧头。

副帧头

存储对应于请求报文的副帧头。

请求报文中添加了序列号的情况下(序列号为1234H的情况下)		请求报文中未添加了序列号的情况下																																													
ASCII代码时	<table><tr><td colspan="4">(固定值)</td><td colspan="4">(固定值)</td></tr><tr><td>D</td><td>4</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>44H</td><td>34H</td><td>30H</td><td>30H</td><td>31H</td><td>32H</td><td>33H</td><td>34H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> 序列号	(固定值)				(固定值)				D	4	0	0	1	2	3	4	0	0	0	0	44H	34H	30H	30H	31H	32H	33H	34H	30H	30H	30H	30H	ASCII代码时	<table><tr><td colspan="4">(固定值)</td></tr><tr><td>D</td><td>0</td><td>0</td><td>0</td></tr><tr><td>44H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	(固定值)				D	0	0	0	44H	30H	30H	30H
(固定值)				(固定值)																																											
D	4	0	0	1	2	3	4	0	0	0	0																																				
44H	34H	30H	30H	31H	32H	33H	34H	30H	30H	30H	30H																																				
(固定值)																																															
D	0	0	0																																												
44H	30H	30H	30H																																												
二进制代码时	<table><tr><td colspan="2">(固定值)</td><td colspan="2">(固定值)</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td>D4H, 00H</td><td>34H, 12H</td><td>00H, 00H</td><td></td></tr></table> 序列号	(固定值)		(固定值)						D4H, 00H	34H, 12H	00H, 00H		二进制代码时	<table><tr><td colspan="2">(固定值)</td></tr><tr><td></td><td></td></tr><tr><td>D0H, 00H</td><td></td></tr></table>	(固定值)				D0H, 00H																											
(固定值)		(固定值)																																													
D4H, 00H	34H, 12H	00H, 00H																																													
(固定值)																																															
D0H, 00H																																															

■以ASCII代码进行数据通信时

序列号按照从高位字节到低位字节的顺序被存储。

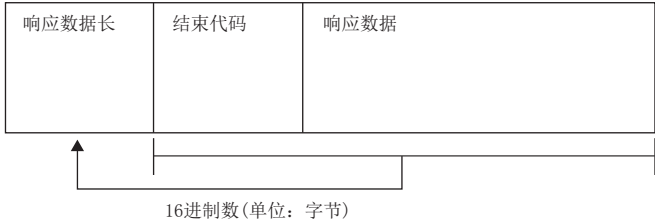
■以二进制代码进行数据通信时

序列号按照从低位字节到高位字节的顺序被存储。

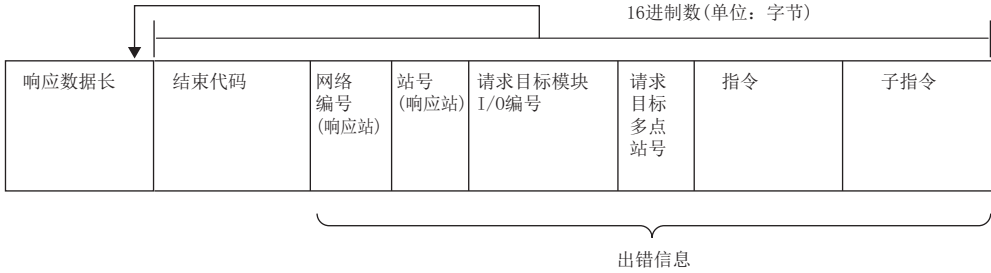
响应数据长

从结束代码开始到响应数据(正常结束时)或出错信息(异常结束时)为止的数据, 以16进制数被存储。(单位: 字节)

(正常结束时)



(异常结束时)



例

响应数据长为22字节的情况下

ASCII代码时

0	0	1	6
30H	30H	31H	36H

二进制代码时

16H	00H
-----	-----

- 以ASCII代码进行数据通信时
按照从高位字节到低位字节的顺序被存储。
- 以二进制代码进行数据通信时
按照从低位字节到高位字节的顺序被存储。

结束代码

存储指令处理结果。
正常结束时存储0。异常结束时存储访问目标的出错代码。(📖SLMP对应设备的手册)

正常结束时	异常结束时 (0400H的情况下)
ASCII代码时 0000 30H,30H,30H,30H 二进制代码时 00H,00H	ASCII代码时 0400 30H,34H,30H,30H 二进制代码时 00H,04H

- 以ASCII代码进行数据通信时
按照从高位字节到低位字节的顺序被存储。
- 以二进制代码进行数据通信时
按照从低位字节到高位字节的顺序被存储。

响应数据

在指令正常结束时，对于指令的读取数据等被存储。关于响应数据有关内容，请参阅指令说明部分的“响应数据”。(📖 39 页 Read(指令: 0401)以后)

出错信息

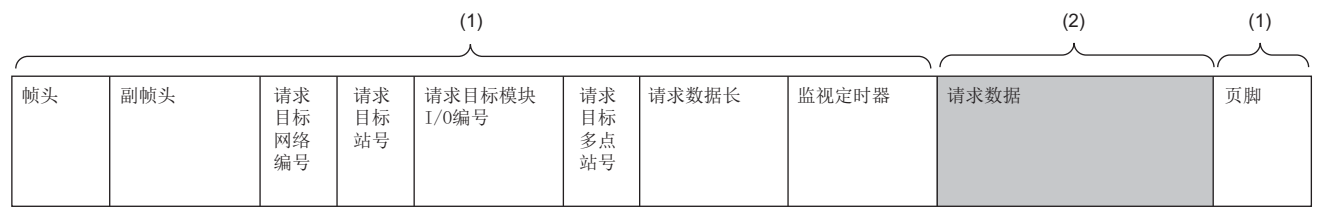
存储进行了出错响应的站的网络编号(响应站)、站号(响应站)、请求目标模块I/O编号、多点站号。由于进行了出错响应的站的信息被存储，因此有可能存储与请求报文的内容不同的编号。此外，发生出错时的指令及子指令被存储。

5 指令

本章对SLMP指令有关内容进行说明。
关于指令部分以外的报文格式，请参阅下述内容。
☞ 15页 报文格式

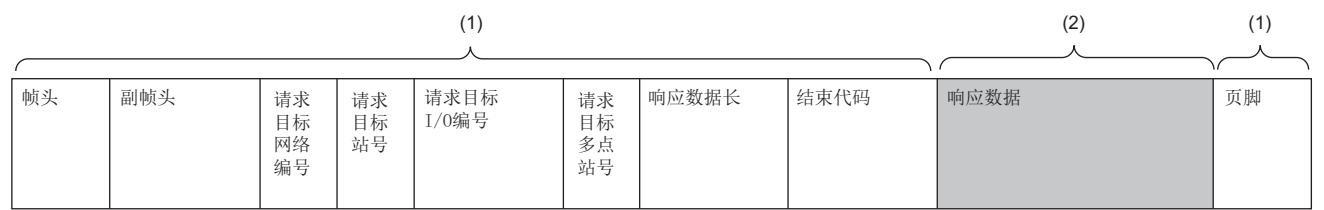
请求报文

- (1) ☞ 15页 请求报文
- (2) 请求数据中包括指令、子指令。详细内容请参阅下述页以后。
☞ 29页 Device(软元件访问)



响应报文

- 正常结束时
- (1) ☞ 20页 响应报文
- (2) 请参阅下述页以后。
☞ 29页 Device(软元件访问)



- 异常结束时
- 请参阅下述页以后。
☞ 15页 报文格式

5.1 指令一览

指令一览

指令一览如下所示。下述“子指令”的□部分根据指定的软元件有所不同。请参阅下述页以后。

☞ 29页 Device(软元件访问)

项目		指令	子指令	内容	参照目标
类型	操作				
Device	Read	0401	00□1 00□3	通过位软元件(连续的软元件编号)以1点为单位读取值。	39页 Read(指令: 0401)
			00□0 00□2	• 通过位软元件(连续的软元件编号)以16点为单位读取值。 • 通过字软元件(连续的软元件编号)以1字为单位读取值。	
	Write	1401	00□1 00□3	以1点为单位将值写入到位软元件(连续的软元件编号)中。	43页 Write(指令: 1401)
			00□0 00□2	• 以16点为单位将值写入到位软元件(连续的软元件编号)中。 • 以1字为单位将值写入到字软元件(连续的软元件编号)中。	
	Read Random	0403	00□0 00□2	指定软元件编号, 读取软元件值。可以指定未连续的软元件编号。 通过字软元件以1字为单位, 或以2字为单位读取。	46页 Read Random(指令: 0403)
	Write Random	1402	00□1 00□3	以1点为单位指定软元件编号, 将值写入到位软元件中。可以以不连续的软元件编号指定。	50页 Write Random(指令: 1402)
			00□0 00□2	• 以16点为单位指定软元件编号, 将值写入到位软元件中。可以以不连续的软元件编号指定。 • 以1字为单位或以2字为单位指定软元件编号, 将值写入到字软元件中。可以以不连续的软元件编号指定。	
	Entry Monitor Device	0801	00□0 00□2	通过Execute Monitor(指令: 0802)登录读取的软元件。	55页 Entry Monitor Device(指令: 0801)
	Execute Monitor	0802	0000	通过Entry Monitor Device(指令: 0801)读取已登录的软元件值。	59页 Execute Monitor(指令: 0802)
	Read Block	0406	00□0 00□2	将字软元件及位软元件(1点为16位)的n点作为1个块, 指定多个块后读取。可以以不连续的软元件编号指定。	62页 Read Block(指令: 0406)
Label	Write Block	1406	00□0 00□2	将字软元件及位软元件(1点为16位)的n点作为1个块, 指定多个块后写入。可以以不连续的软元件编号指定。	66页 Write Block(指令: 1406)
	Array Label Read	041A	0000	排列型标签及结构体的构件通过排列的标签读取数据。	78页 Array Label Read(指令: 041A)
	Array Label Write	141A	0000	排列型标签及结构体的构件将输入写入到排列的标签中。	87页 Array Label Write(指令: 141A)
	Label Read Random	041C	0000	指定标签, 读取数据。	96页 Label Read Random(指令: 041C)
	Label Write Random	141B	0000	指定标签, 写入数据。	102页 Label Write Random(指令: 141B)

项目		指令	子指令	内容	参照目标
类型	操作				
Memory	Read	0613	0000	读取本站(SLMP对应设备)的缓冲存储器数据。	110页 Read(指令: 0613)
	Write	1613	0000	将数据写入到本站(SLMP对应设备)的缓冲存储器中。	112页 Write(指令: 1613)
Extend Unit	Read	0601	0000	读取智能功能模块的缓冲存储器数据。	116页 Read(指令: 0601)
	Write	1601	0000	将数据写入到智能功能模块的缓冲存储器中。	118页 Write(指令: 1601)
Remote Control	Remote Run	1001	0000	对于访问目标模块执行远程RUN。	121页 Remote Run(指令: 1001)
	Remote Stop	1002	0000	对于访问目标模块执行远程STOP。	123页 Remote Stop(指令: 1002)
	Remote Pause	1003	0000	对于访问目标模块执行远程PAUSE。	124页 Remote Pause(指令: 1003)
	Remote Latch Clear	1005	0000	对于访问目标模块执行远程锁存清除。	125页 Remote Latch Clear(指令: 1005)
	Remote Reset	1006	0000	对于访问目标模块执行远程RESET。	126页 Remote Reset(指令: 1006)
	Read Type Name	0101	0000	读取访问目标模块的型号及型号代码。	127页 Read Type Name(指令: 0101)
Remote Password	Lock	1631	0000	指定远程口令后, 置为对于其它设备不可以通信的状态。 (从解锁状态置为锁定状态。)	131页 Lock(指令: 1631)
	Unlock	1630	0000	指定远程口令后, 置为对于其它设备可通信的状态。 (从锁定状态置为解锁状态。)	133页 Unlock(指令: 1630)
File	Read Directory/File	1810	0000 0040	读取文件一览信息。	145页 Read Directory/File(指令: 1810)
	Search Directory/File	1811	0000 0040	读取指定文件的有无、文件No.、文件容量。	154页 Search Directory/File(指令: 1811)
	New File	1820	0000 0040	确保指定文件的存储区。	157页 New File(指令: 1820)
	Delete File	1822	0000 0004 0040	删除文件。	160页 Delete File(指令: 1822)
	Copy File	1824	0000 0004 0040	复制指定文件。	163页 Copy File(指令: 1824)
	Change File State	1825	0000 0004 0040	更改文件属性。	167页 Change File State(指令: 1825)
	Change File Date	1826	0000 0040	更改文件的创建日期。	170页 Change File Date(指令: 1826)
	Open File	1827	0000 0004 0040	为了防止通过其它设备更改文件内容, 对文件进行锁定。	173页 Open File(指令: 1827)
	Read File	1828	0000	读取文件内容。	176页 Read File(指令: 1828)
	Write File	1829	0000	将内容写入到文件中。	179页 Write File(指令: 1829)
	Close File	182A	0000	对通过打开处理的文件锁定进行解除。	182页 Close File(指令: 182A)
Self Test		0619	0000	对与对象设备的通信是否正常动作进行测试。	184页 Self Test(反复测试)(指令: 0619)
Clear Error		1617	0000	将本站的COM. ERR. LED熄灯。	186页 Clear Error(出错代码初始化、LED熄灯)(指令: 1617)

项目		指令	子指令	内容	参照目标
类型	操作				
Ondemand		2101	0000	通过CPU模块对于SLMP对应设备提出发送请求后，将数据发送到外部设备。	187页 Ondemand(指令：2101)

各指令的可访问模块

SLMP的请求报文中可指定的访问目标模块如下所示。

访问目标模块中指定以太网搭载模块的情况下，请参阅所使用的以太网搭载模块的用户手册。

○：可访问，×：不可访问

项目		指令	子指令	可访问模块		
类型	操作			CPU模块		CC-Link IE现场网络智能设备站
				MELSEC iQ-R系列	MELSEC-Q/L系列	
Device	Read	0401	00□1 00□0	○	○	○
			00□3 00□2		×	×
	Write	1401	00□1 00□0		○	○
			00□3 00□2		×	×
	Read Random	0403	00□0		○	○
			00□2		×	×
	Write Random	1402	00□1 00□0		○	○
			00□3 00□2		×	×
	Entry Monitor Device	0801	00□0		○	○
			00□2		×	×
	Execute Monitor	0802	0000		○	○
	Read Block	0406	00□0		○	○
			00□2		×	×
	Write Block	1406	00□0		○	○
			00□2		×	×
Label	Array Label Read	041A	0000	×	×	
	Array Label Write	141A	0000			
	Label Read Random	041C	0000			
	Label Write Random	141B	0000			
Memory	Read	0613	0000	○	○ ^{*3}	
	Write	1613	0000			
Extend Unit	Read	0601	0000		○ ^{*4}	
	Write	1601	0000			
Remote Control	Remote Run	1001	0000			○
	Remote Stop	1002	0000			×
	Remote Pause	1003	0000			
	Remote Latch Clear	1005	0000			○
	Remote Reset	1006	0000			
	Read Type Name	0101	0000			
Remote Password ^{*1}	Lock	1631	0000	○	×	
	Unlock	1630	0000			

项目		指令	子指令	可访问模块		
类型	操作			CPU模块		CC-Link IE现场网络智能设备站
				MELSEC iQ-R系列	MELSEC-Q/L系列	
File	Read Directory/File	1810	0000	×	○	○ ^{*4}
			0040	○	×	×
	Search Directory/File	1811	0000	×	○	○ ^{*4}
			0040	○	×	×
	New File	1820	0000	×	○	○ ^{*4}
			0040	○	×	×
	Delete File	1822	0000	×	○ ^{*6*7}	○ ^{*4*8}
			0004			
			0040	○	×	×
	Copy File	1824	0000	×	○ ^{*6*7}	○ ^{*4*8}
			0004			
			0040	○	×	×
	Change File State	1825	0000	×	○ ^{*6*7}	○ ^{*4*8}
			0004			
			0040	○	×	×
	Change File Date	1826	0000	×	○	○ ^{*4}
			0040	○	×	×
	Open File	1827	0000	×	○ ^{*6*7}	○ ^{*4*8}
			0004			
			0040	○	×	×
	Read File	1828	0000	○	○	○ ^{*4}
	Write File	1829	0000			
	Close File	182A	0000			
Self Test		0619	0000	— ^{*2}	×	×
Clear Error		1617	0000	×	×	○ ^{*1*3}
Ondemand		2101	0000	— ^{*5}	— ^{*5}	— ^{*5}

- *1 仅对于与外部设备连接的连接站可以使用。
- *2 仅对于与外部设备连接的以太网搭载模块可以使用。
- *3 只可是CC-Link IE现场网络以太网适配器模块。
- *4 只可是CC-Link IE现场网络起始模块。
- *5 是通过SLMP对应设备至外部设备的发送用。
- *6 子指令的0004不可以访问QCPU。
- *7 在访问LCPU时，对象文件中未设置口令的情况下可以使用。
- *8 对象文件中未设置口令的情况下可以使用。

5.2 Device(软元件访问)

以下说明进行软元件读取及写入的指令有关内容。

要点

- 访问目标、连接站为MELSEC-Q/L系列模块以外的情况下，应使用子指令的00□1、00□0。
- 访问目标、连接站为MELSEC iQ-R系列模块的情况下，应使用子指令的00□3、00□2。子指令的00□1、00□0可以作为与MELSEC-Q/L系列模块的互换用使用。

指令内指定的数据

软元件代码

请求数据中，通过下述软元件代码指定访问目标的软元件。
子指令为0001、0000的情况下，应指定()中记载的软元件代码。

软元件	类型	软元件代码		软元件编号范围		备注
		ASCII代码 时 ^{*1}	二进制代码 时			
功能输入(FX)	位	—	—	—	16进制	SIMP中不可以指定。
功能输出(FY)		—	—		16进制	
功能寄存器(FD)	字	—	—		10进制	
特殊继电器(SM)	位	SM** (SM)	0091H (91H)	在访问目标模块具有的软元件编号的范围内进行指定。	10进制	—
特殊寄存器(SD)	字	SD** (SD)	00A9H (A9H)		10进制	
输入(X)	位	X*** (X*)	009CH (9CH)		16进制	—
输出(Y)		Y*** (Y*)	009DH (9DH)		16进制	
内部继电器(M)		M*** (M*)	0090H (90H)		10进制	不可以访问局部软元件。
锁存继电器(L)		L*** (L*)	0092H (92H)		10进制	—
报警器(F)		F*** (F*)	0093H (93H)		10进制	
变址继电器(V)		V*** (V*)	0094H (94H)		10进制	不可以访问局部软元件。
链接继电器(B)		B*** (B*)	00A0H (A0H)		16进制	—
数据寄存器(D)	字	D*** (D*)	00A8H (A8H)		10进制	不可以访问局部软元件。
链接寄存器(W)		W*** (W*)	00B4H (B4H)		16进制	—

软元件		类型	软元件代码		软元件编号范围		备注
			ASCII代码 时*1	二进制代码 时			
定时器(T)	触点(TS)	位	TS** (TS)	00C1H (C1H)	在访问目标模块具有的软元件编号的范围内进行指定。	10进制	不可以访问局部软元件。
	线圈(TC)		TC** (TC)	00C0H (C0H)			
	当前值(TN)	字	TN** (TN)	00C2H (C2H)			
长定时器(LT)	触点(LTS)	位	LTS* (—)	0051H (—)		10进制	• 仅子指令的0003、0002中可以使用。 • 不可以访问局部软元件。
	线圈(LTC)		LTC* (—)	0050H (—)			
	当前值(LTN)	双字	LTN* (—)	0052H (—)			
累计定时器(ST)	触点(STS)	位	STS* (SS)	00C7H (C7H)		10进制	不可以访问局部软元件。
	线圈(STC)		STC* (SC)	00C6H (C6H)			
	当前值(STN)	字	STN* (SN)	00C8H (C8H)			
长累计定时器(LST)	触点(LSTS)	位	LSTS (—)	0059H (—)		10进制	• 仅子指令的0003、0002中可以使用。 • 不可以访问局部软元件。
	线圈(LSTC)		LSTC (—)	0058H (—)			
	当前值(LSTN)	双字	LSTN (—)	005AH (—)			
计数器(C)	触点(CS)	位	CS** (CS)	00C4H (C4H)		10进制	不可以访问局部软元件。
	线圈(CC)		CC** (CC)	00C3H (C3H)			
	当前值(CN)	字	CN** (CN)	00C5H (C5H)			
长计数器(LC)	触点(LCS)	位	LCS* (—)	0055H (—)		10进制	• 仅子指令的0003、0002中可以使用。 • 不可以访问局部软元件。
	线圈(LCC)		LCC* (—)	0054H (—)			
	当前值(LCN)	双字	LCN* (—)	0056H (—)			

软元件	类型	软元件代码		软元件编号范围		备注
		ASCII代码 时 ^{*1}	二进制代码 时			
链接特殊继电器(SB)	位	SB** (SB)	00A1H (A1H)	在访问目标模块具有的软元件编号的范围内进行指定。	16进制	—
链接特殊寄存器(SW)	字	SW** (SW)	00B5H (B5H)		16进制	
步继电器(S)	位	—	—	—	10进制	SLMP中不可以指定。
直接访问输入(DX)	位	DX** (DX)	00A2H (A2H)	在访问目标模块具有的软元件编号的范围内进行指定。	16进制	—
直接访问输出(DY)		DY** (DY)	00A3H (A3H)		16进制	
变址寄存器(Z)	字	Z*** (Z*)	00CCH (CCH)		10进制	不可以访问局部软元件。
长变址寄存器(LZ)	双字	LZ** (—)	0062H (—)			• 仅子指令的0003、0002中可以使用。 • 不可以访问局部软元件。
文件寄存器(R、ZR) ^{*2*3}	字	R*** (R*)	00AFH (AFH)		10进制	块切换方式
		ZR** (ZR)	00BOH (BOH)		16进制	连号访问方式
扩展数据寄存器(D) ^{*4}	字	— (D*)	— (A8H)	二进制代码时：在访问目标模块具有的软元件编号的范围内进行指定。 ASCII代码时： 000000~999999	10进制	—
扩展链接寄存器(W) ^{*4}	字	— (W*)	— (B4H)	在访问目标模块具有的软元件编号的范围内进行指定。	16进制	
模块刷新用寄存器(RD)	字	RD** (—)	002CH (—)		10进制	仅子指令的0003、0002中可以使用。
链接直接软元件	☞ 190页 至链接直接软元件的访问					
模块访问软元件	☞ 194页 至模块访问软元件的访问					
CPU缓冲存储器访问软元件	☞ 197页 至CPU缓冲存储器访问软元件的访问					

1 以ASCII代码进行数据通信时，子指令为00□3、00□2的情况下，以4位指定软元件代码。软元件代码为3位以下时，在软元件代码的后面添加“”（ASCII代码：2AH）或空格（ASCII代码：20H）。

子指令为00□1、00□0的情况下，以2位指定软元件代码。软元件代码为1位的情况下，在软元件代码的后面添加“*”（ASCII代码：2AH）或空格（ASCII代码：20H）。

*2 对于通过CPU模块的“CPU参数”或“可编程控制器参数”，指定为“对每个程序使用文件寄存器”的文件寄存器，不可以通过外部设备访问。

*3 CPU模块的文件寄存器由多个块构成的情况下，应使用连号访问方式的软元件代码（“ZR**、ZR”或“00BOH、BOH”）。

关于通过连号访问方式对由多个块构成的文件寄存器进行指定的方法有关内容，请参阅CPU模块手册。

*4 访问目标CPU模块不支持至D65536以后的扩展数据寄存器，及W10000以后的扩展链接寄存器的访问的情况下，应替换为文件寄存器(ZR)后指定。关于至文件寄存器(ZR)替换方法，请参阅CPU模块手册。

■以ASCII代码进行数据通信时

将软元件代码转换为ASCII代码(4位或2位)使用后，按照从高位字节到低位字节的顺序发送。英文使用大写字母代码。
在子指令为0003、0002与0001、0000中，转换为ASCII代码的位数有所不同。

子指令	位数	例
0003 0002	转换为ASCII代码4位	输入(X)的情况下(4位) ^{*1} X * * * 58H, 2AH, 2AH, 2AH
0001 0000	转换为ASCII代码2位	输入(X)的情况下(2位) ^{*1} X * 58H, 2AH

^{*1} 输入继电器的软元件代码从“X”开始依次发送。此外，第2字符以后的“*”也可通过空格(代码：20H)指定。

■以二进制代码进行数据通信时

使用数值(2字节或1字节)，按照从低位字节到高位字节的顺序发送。
在子指令为0003、0002与0001、0000中，数值的数据容量有所不同。

子指令	数据容量	例
0003 0002	2字节	输入(X)的情况下(2字节) 9CH, 00H
0001 0000	1字节	输入(X)的情况下(1字节) 9CH

要点

对于访问目标模块中可使用的软元件，请参阅访问目标模块的手册。(所使用的模块手册)

起始软元件编号(软元件编号)

指定进行读取或写入的软元件编号。指定连续的软元件的情况下，指定软元件的起始编号。
起始软元件编号根据软元件类型以10进制数或16进制数进行指定。(29页 软元件代码)

■以ASCII代码进行数据通信时

将软元件编号转换为ASCII代码(8位或6位)使用后，按照从高位字节到低位字节的顺序发送。
在子指令为0003、0002与0001、0000中，转换为ASCII代码的位数有所不同。

子指令	位数	例
0003 0002	转换为ASCII代码8位	软元件编号为1234的情况下(8位) ^{*1} 0 0 0 0 1 2 3 4 30H, 30H, 30H, 30H, 31H, 32H, 33H, 34H
0001 0000	转换为ASCII代码6位	软元件编号为1234的情况下(6位) ^{*1} 0 0 1 2 3 4 30H, 30H, 31H, 32H, 33H, 34H

^{*1} 从0开始依次发送。高位数的0，也可通过空格(代码：20H)指定。

■以二进制代码进行数据通信时

使用数值(4字节或3字节)，按照从低位字节到高位字节的顺序发送。将软元件编号为10进制数的软元件转换为16进制数发送。
在子指令为0003、0002与0001、0000中，数值的数据容量有所不同。

子指令	数据容量	例
0003 0002	4字节	内部继电器M1234、链接继电器B1234的情况下(4字节) ^{*1} M1234 D2H, 04H, 00H, 00H B1234 34H, 12H, 00H, 00H
0001 0000	3字节	内部继电器M1234、链接继电器B1234的情况下(3字节) ^{*2} M1234 D2H, 04H, 00H B1234 34H, 12H, 00H

*1 由于软元件编号为10进制数，因此内部继电器M1234将转换为16进制数。将变为000004D2H，按照D2H、04H、00H、00H的顺序发送。链接继电器B1234将变为00001234H，按照34H、12H、00H、00H的顺序发送。
*2 由于软元件编号为10进制数，因此内部继电器M1234将转换为16进制数。将变为0004D2H，按照D2H、04H、00H的顺序发送。链接继电器B1234将变为001234H，按照34H、12H、00H的顺序发送。

软元件点数

指定进行读取或写入的软元件点数。

■以ASCII代码进行数据通信时

将点数转换为ASCII代码4位(16进制数)使用后，按照从高位字节到低位字节的顺序发送。指定英文的情况下，使用大写字母代码。

例

5点，20点的情况下



■以二进制代码进行数据通信时

使用表示处理点数的2字节的数值，按照从低位字节到高位字节的顺序发送。

例

5点，20点的情况下



读取数据、写入数据

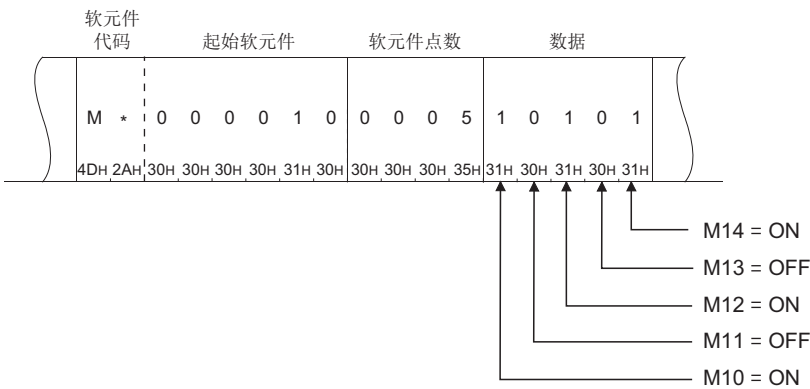
读取的情况下，存储已读取的软元件的值。写入的情况下，存储写入的数据。
根据位单位(子指令：00□1、00□3)或字单位(子指令：00□0，00□2)，数据排列有所不同。

■位单位(子指令：00□1、00□3)的情况下

以ASCII代码进行数据通信时，通过指定的起始软元件将指定软元件点数从高位位开始按顺序发送。如果为ON以“31H”(1)表现，如果为OFF以“30H”(0)表现。指定英文的情况下，使用大写字母代码。

例

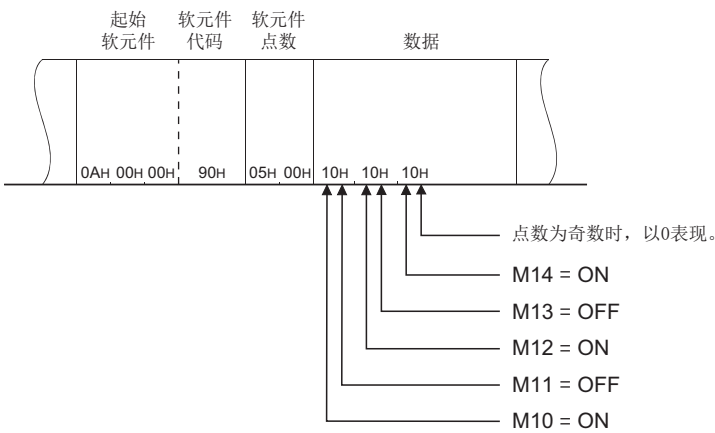
通过M10表示5点的ON/OFF的情况下



以二进制代码进行数据通信时，以4位指定1点后，通过指定的起始软元件将指定软元件点数从高位位开始按顺序发送。如果为ON以“1”表现，如果为OFF以“0”表现。

例

通过M10表示5点的ON/OFF的情况下

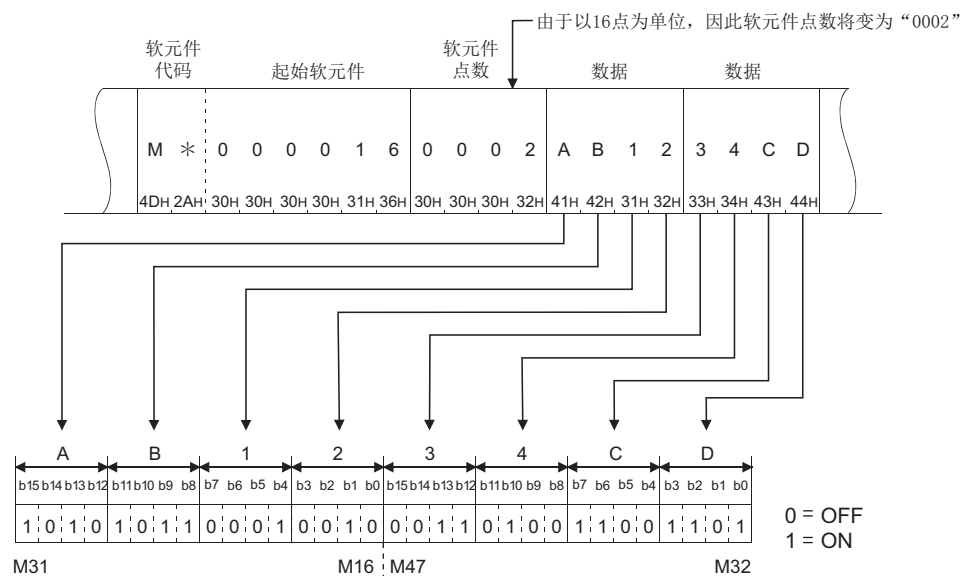


■字单位(子指令: 00□0、00□2)的情况下

以ASCII代码进行数据通信时，将1字以4位为单位从高位位开始按顺序发送。数据将以16进制数表现。指定英文的情况下，使用大写字母代码。

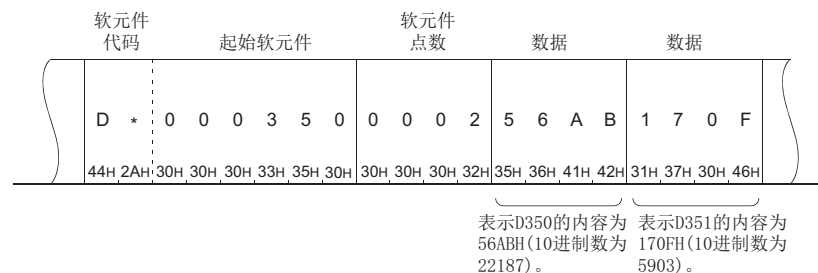
例

通过M16表示32点的ON/OFF的情况下



例

表示D350、D351的存储内容的情况下



要点

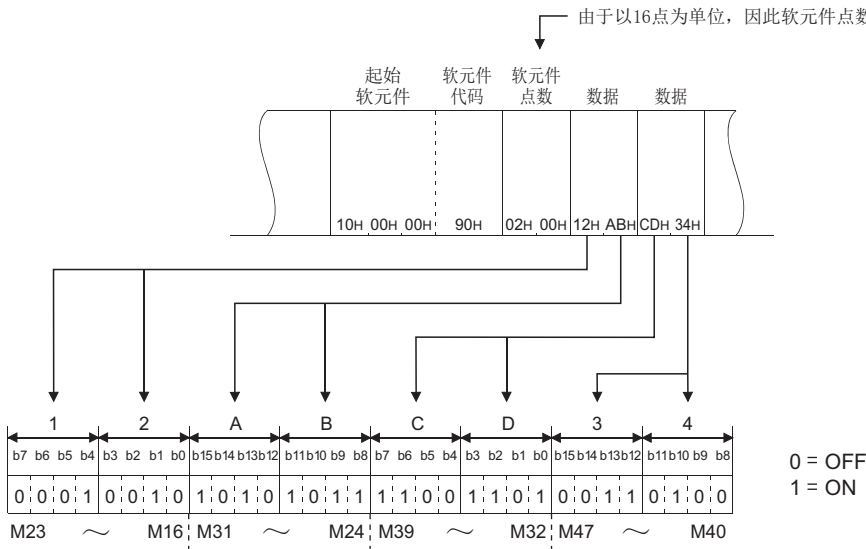
读取数据的字软件件中存储了整数以外(实数、字符串)时,将存储值作为整数值读取。

- D0~D1中实数(0.75)被存储时, D0=0000H、D1=3F40H
- D2~D3中字符串(“12AB”)被存储时, D2=3231H、D3=4241H

以二进制代码进行数据通信时，以字单位处理位软元件的情况下，按照下述示例所示以1位指定1点。存储顺序将变为从低位字节(位0~7)到高位字节(位8~15)。

例

通过M16表示32点的0N/0FF的情况下

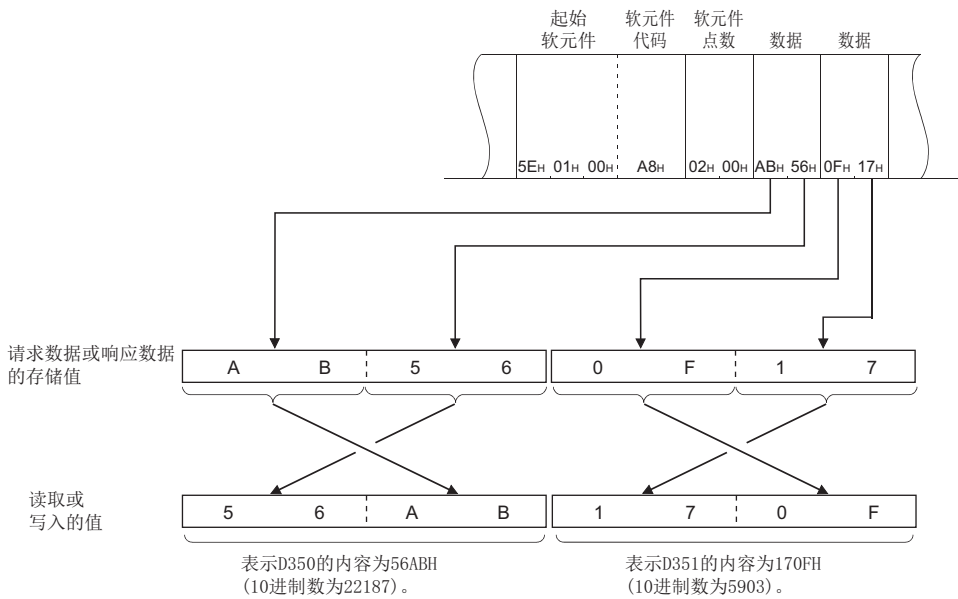


对于字软元件，按照下述示例所示以16位指定1字。存储顺序将变为从低位字节(位0~7)到高位字节(位8~15)。

读取时，应将响应数据中存储的值在用户侧上下字节置换后进行读取。
写入时，应将希望写入的值在用户侧上下字节置换后，存储到请求数据中。

例

表示D350、D351的存储内容的情况下



要点

读取数据的字软元件中存储了整数以外(实数、字符串)时，将存储值作为整数值读取。

- D0~D1中实数(0.75)被存储时，D0=0000H、D1=3F40H
- D2~D3中字符串(“12AB”)被存储时，D2=3231H、D3=4241H

■注意事项

进行ASCII数据的通信的情况下，从外部设备向CPU模块转移字符串时，应按照下述方式进行处理。

以下对SLMP对应设备将通过外部设备接收了的数据转换为二进制代码数据后，写入到指定软元件中的步骤进行说明。

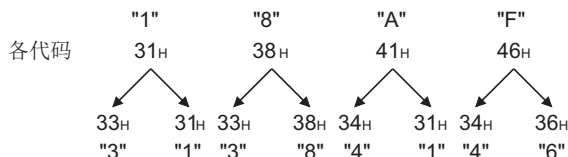
1. 将通过外部设备发送的字符串展开为每1字符2字节的代码。

2. 将展开为2字节的字符串按每2字符排序后发送至SLMP对应设备中。

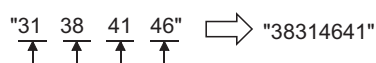
3. 将SLMP对应设备中被发送的数据写入到指定软元件中。

将通过外部设备接收了的字符串(“18AF”)转换为二进制代码数据后写入到D0～D1中的情况下的示例如下所示。

1. 将通过外部设备发送的字符串(“18AF”)展开为每1字符2字节的代码。



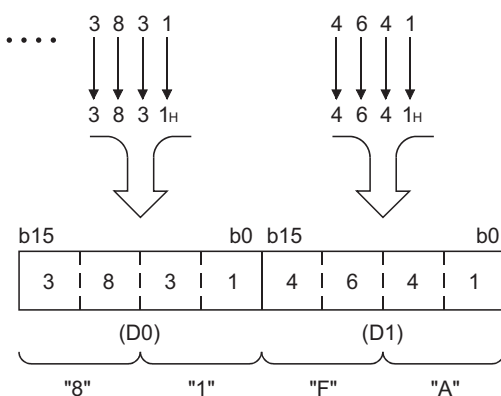
2. 将展开为2字节的字符串按每2字符排序后发送至SLMP对应设备中。



从外部设备将“38314641”发送至CPU模块中。

3. 将SLMP对应设备中被发送的“38314641”数据写入到D0～D1中。

将接收数据转换为
二进制代码的数据
写入



各字节的数值
表示的数字

位访问点数

是用于指定以位单位进行访问的点数的数据。

■以ASCII代码进行数据通信时

将点数转换为ASCII代码2位(16进制数)后，从高位位开始发送。指定英文的情况下，使用大写字母代码。

例

5点，20点的情况下

5点

0	5
30H	35H

20点

1	4
31H	34H

■以二进制代码进行数据通信时

将点数转换为16进制数后发送。

例

5点，20点的情况下

5点

05H

20点

14H

Read(指令：0401)

通过软元件读取值。

请求数据

ASCII

0	4	0	1	子指令	软元件代码	起始软元件编号	软元件点数
30H	34H	30H	31H				

二进制

子指令	起始软元件编号	软元件代码	软元件点数
01H, 04H			

子指令

项目	子指令 ^{*1}																									
	ASCII代码	二进制代码																								
以位单位读取的情况下	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>31H</td></tr></table>	0	0	0	1	30H	30H	30H	31H	0	0	8	1	30H	30H	38H	31H	<table><tr><td></td><td></td></tr><tr><td>01H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>81H</td><td>00H</td></tr></table>			01H	00H			81H	00H
	0	0	0	1																						
30H	30H	30H	31H																							
0	0	8	1																							
30H	30H	38H	31H																							
01H	00H																									
81H	00H																									
	<table><tr><td>0</td><td>0</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>33H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>33H</td></tr></table>	0	0	0	3	30H	30H	30H	33H	0	0	8	3	30H	30H	38H	33H	<table><tr><td></td><td></td></tr><tr><td>03H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>83H</td><td>00H</td></tr></table>			03H	00H			83H	00H
0	0	0	3																							
30H	30H	30H	33H																							
0	0	8	3																							
30H	30H	38H	33H																							
03H	00H																									
83H	00H																									
以字单位读取的情况下	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	0	0	8	0	30H	30H	38H	30H	<table><tr><td></td><td></td></tr><tr><td>00H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>80H</td><td>00H</td></tr></table>			00H	00H			80H	00H
	0	0	0	0																						
30H	30H	30H	30H																							
0	0	8	0																							
30H	30H	38H	30H																							
00H	00H																									
80H	00H																									
	<table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>32H</td></tr></table>	0	0	0	2	30H	30H	30H	32H	0	0	8	2	30H	30H	38H	32H	<table><tr><td></td><td></td></tr><tr><td>02H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>82H</td><td>00H</td></tr></table>			02H	00H			82H	00H
0	0	0	2																							
30H	30H	30H	32H																							
0	0	8	2																							
30H	30H	38H	32H																							
02H	00H																									
82H	00H																									

*1 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。(190页 通过软元件的扩展指定进行读取、写入)

软元件代码

指定读取的软元件类型。(29页 软元件代码)

但是，不可以指定下述软元件的触点、线圈、当前值。

- 长定时器(LTS、LTC、LTN)
- 长累计定时器(LSTS、LSTC、LSTN)
- 长计数器(LCS、LCC、LCN)

起始软元件编号

指定读取的软元件的起始编号。(32页 起始软元件编号(软元件编号))

软元件点数

指定读取的软元件点数。(33页 软元件点数)

项目	点数	
	ASCII代码	二进制代码
以位单位读取的情况下	1~3584点	1~7168点
以字单位读取的情况下	1~960点	

响应数据

以16进制数存储已读取的软元件值。根据ASCII代码及二进制代码，数据的排列有所不同。(34页 读取数据、写入数据)

读取数据

通信示例(以位单位读取的情况下)

读取M100～M107。

■以ASCII代码进行数据通信时
(请求数据)

子指令				软元件 代码	起始软元件编号						软元件点数								
0	4	0	1	0	0	0	1	M	*	0	0	0	1	0	0	0	0	0	8
30H	34H	30H	31H	30H	30H	30H	31H	4DH	2AH	30H	30H	30H	31H	30H	30H	30H	30H	30H	38H

(响应数据)

0	0	0	1	0	0	1	1	0 = OFF
30H	30H	30H	31H	30H	30H	31H	31H	1 = ON
M100				～	M107			

■以二进制代码进行数据通信时
(请求数据)

子指令		软元件 代码		起始软元件 编号		软元件 点数
:	:	:	:	:	:	:
01H	04H	01H	00H	64H	00H	00H
90H	08H	00H				

(响应数据)

00H	01H	00H	11H	0 = OFF
				1 = ON
M101		M107		
M100		M106		

通信示例(以字单位读取的情况下(位软元件))

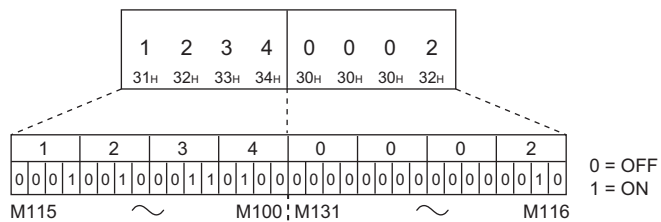
读取M100～M131(2字)。

■以ASCII代码进行数据通信时

(请求数据)

子指令				软元件 代 码	起始软元件编号				软元件点数								
0	4	0	1	0 0 0 0	M	*	0	0	0	1	0	0	0	0	0	2	
30 _H	34 _H	30 _H	31 _H	30 _H	30 _H	30 _H	30 _H	4D _H	2A _H	30 _H	30 _H	30 _H	31 _H	30 _H	30 _H	30 _H	32 _H

(响应数据)

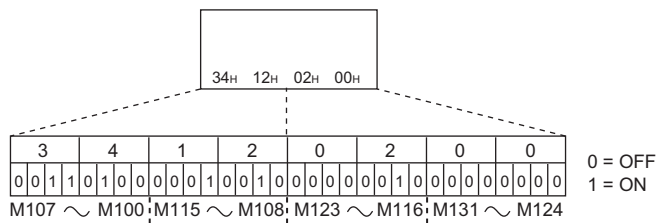


■以二进制代码进行数据通信时

(请求数据)

子指令		软元件 代码	
起始软元件 编号		软元件 点数	
01 _H	04 _H	00 _H	00 _H
64 _H	00 _H	00 _H	90 _H
02 _H	00 _H		

(响应数据)



通信示例(以字单位读取的情况下(字软元件))

读取T100~T102的值。
作为存储T100=4660(1234H)、T101=2(2H)、T102=7663(1DEFH)的软元件。

■以ASCII代码进行数据通信时
(请求数据)

子指令				软元件 代码		起始软元件编号						软元件点数							
0	4	0	1	0	0	0	0	T	N	0	0	0	1	0	0	0	0	0	3
30H	34H	30H	31H	30H	30H	30H	30H	54H	4EH	30H	30H	30H	31H	30H	30H	30H	30H	30H	33H

Write(指令：1401)

将值写入到软元件中。

请求数据

ASCII

1	4	0	1	子指令	软元件代码	起始软元件编号	软元件点数	写入数据
31H	34H	30H	31H					

二进制

	子指令	起始软元件编号	软元件代码	软元件点数	写入数据
01H, 14H					

子指令

项目	子指令 ^{*1}																					
	ASCII代码	二进制代码																				
以位单位写入的情况下	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>31H</td></tr></table>	0	0	0	1	30H	30H	30H	31H	0	0	8	1	30H	30H	38H	31H	<table><tr><td>01H</td><td>00H</td></tr></table> 或 <table><tr><td>81H</td><td>00H</td></tr></table>	01H	00H	81H	00H
	0	0	0	1																		
30H	30H	30H	31H																			
0	0	8	1																			
30H	30H	38H	31H																			
01H	00H																					
81H	00H																					
	<table><tr><td>0</td><td>0</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>33H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>33H</td></tr></table>	0	0	0	3	30H	30H	30H	33H	0	0	8	3	30H	30H	38H	33H	<table><tr><td>03H</td><td>00H</td></tr></table> 或 <table><tr><td>83H</td><td>00H</td></tr></table>	03H	00H	83H	00H
0	0	0	3																			
30H	30H	30H	33H																			
0	0	8	3																			
30H	30H	38H	33H																			
03H	00H																					
83H	00H																					
以字单位写入的情况下	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	0	0	8	0	30H	30H	38H	30H	<table><tr><td>00H</td><td>00H</td></tr></table> 或 <table><tr><td>80H</td><td>00H</td></tr></table>	00H	00H	80H	00H
	0	0	0	0																		
30H	30H	30H	30H																			
0	0	8	0																			
30H	30H	38H	30H																			
00H	00H																					
80H	00H																					
	<table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>32H</td></tr></table>	0	0	0	2	30H	30H	30H	32H	0	0	8	2	30H	30H	38H	32H	<table><tr><td>02H</td><td>00H</td></tr></table> 或 <table><tr><td>82H</td><td>00H</td></tr></table>	02H	00H	82H	00H
0	0	0	2																			
30H	30H	30H	32H																			
0	0	8	2																			
30H	30H	38H	32H																			
02H	00H																					
82H	00H																					

*1 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。(190页 通过软元件的扩展指定进行读取、写入)

软元件代码

指定写入的软元件类型。(29页 软元件代码)

但是，不可以指定下述软元件的触点、线圈、当前值。

- 长定时器(LTS、LTC、LTN)
- 长累计定时器(LSTS、LSTC、LSTN)
- 长计数器(LCS、LCC、LCN)

起始软元件编号

指定写入软元件的起始编号。(32页 起始软元件编号(软元件编号))

软元件点数

指定写入的软元件点数。(33页 软元件点数)

项目	点数	
	ASCII代码	二进制代码
以位单位读取的情况下	1~3584点	1~7168点
以字单位读取的情况下	1~960点	

写入数据

通过“软元件点数”中指定的点数指定软元件中写入的值。(34页 读取数据、写入数据)

响应数据

无Write指令的响应数据。

通信示例(以位单位写入的情况下)

将值写入到M100～M107中。

■以ASCII代码进行数据通信时
(请求数据)

子指令				软元件 代码	起始软元件编号				软元件点数	写入数据																
1	4	0	1	0	0	0	1	M	*	0	0	0	1	0	0	0	0	8	1	1	0	0	1	1	0	0
31H	34H	30H	31H	30H	30H	30H	31H	4DH	2AH	30H	30H	30H	31H	30H	30H	30H	38H	31H	31H	30H	30H	31H	31H	30H	30H	
										M100								~	M107				0 = OFF 1 = ON			

■以二进制代码进行数据通信时
(请求数据)

子指令	起始软元件 编号	软元件 代码	软元件 点数	写入数据
01H, 14H	01H, 00H	64H, 00H, 00H	90H, 08H, 00H	11H, 00H, 11H, 00H
				M107 ~ M100

0 = OFF
1 = ON

通信示例(以字单位写入的情况下(位软元件))

将值写入到M100～M131(2字)中。

■以ASCII代码进行数据通信时
(请求数据)

子指令				软元件 代码		起始软元件编号				软元件点数				写入数据													
1	4	0	1	0	0	0	0	M	*	0	0	0	1	0	0	0	0	0	2	2	3	4	7	A	B	9	6
31 _H	34 _H	30 _H	31 _H	30 _H	30 _H	30 _H	30 _H	4D _H	2A _H	30 _H	30 _H	30 _H	31 _H	30 _H	30 _H	30 _H	30 _H	32 _H	32 _H	33 _H	34 _H	37 _H	41 _H	42 _H	39 _H	36 _H	
										M115										M100;M131		M116		0 = OFF 1 = ON			

■以二进制代码进行数据通信时
(请求数据)

子指令	起始软元件 编号	软元件 代码	软元件 点数	写入数据
01H, 14H	00H, 00H	64H, 00H, 00H	90H, 02H, 00H	47H, 23H, 96H, ABH
				M107 ~ M100; M115 ~ M108; M123 ~ M116; M131 ~ M124

0 = OFF
1 = ON

通信示例(以字单位写入的情况下(字软元件))

将6549(1995H)写入到D100中， 4610(1202H)写入到D101中， 4400(1130H)写入到D102中。

■以ASCII代码进行数据通信时
(请求数据)

子指令				软元件 代码		起始软元件编号				软元件点数				写入数据																	
1	4	0	1	0	0	0	0	D	*	0	0	0	1	0	0	0	0	0	3	1	9	9	5	1	2	0	2	1	1	3	0
31H	34H	30H	31H	30H	30H	30H	30H	44H	2AH	30H	30H	30H	31H	30H	30H	30H	30H	30H	33H	31H	39H	39H	35H	31H	32H	30H	32H	31H	31H	33H	30H
													D100				D101				D102										

■以二进制代码进行数据通信时
(请求数据)

子指令		起始软元件 编号		软元件 代码	软元件 点数	写入数据								
01H	14H	00H	00H	64H	00H	A8H	03H	00H	95H	19H	02H	12H	30H	11H
									D100		D101		D102	

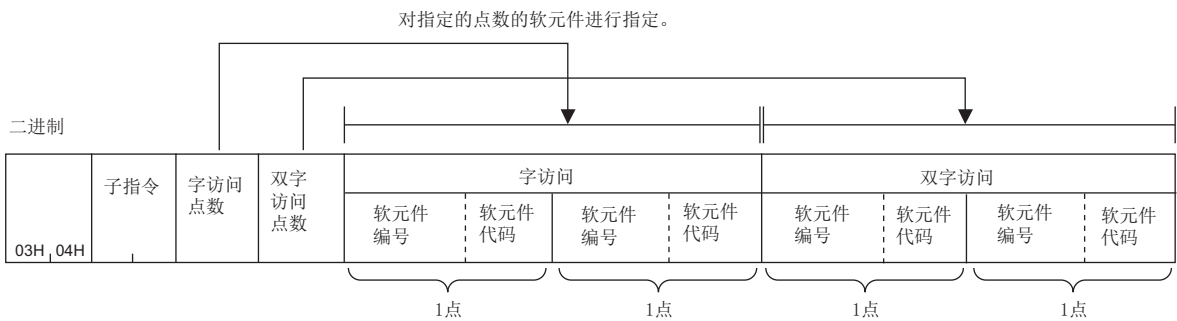
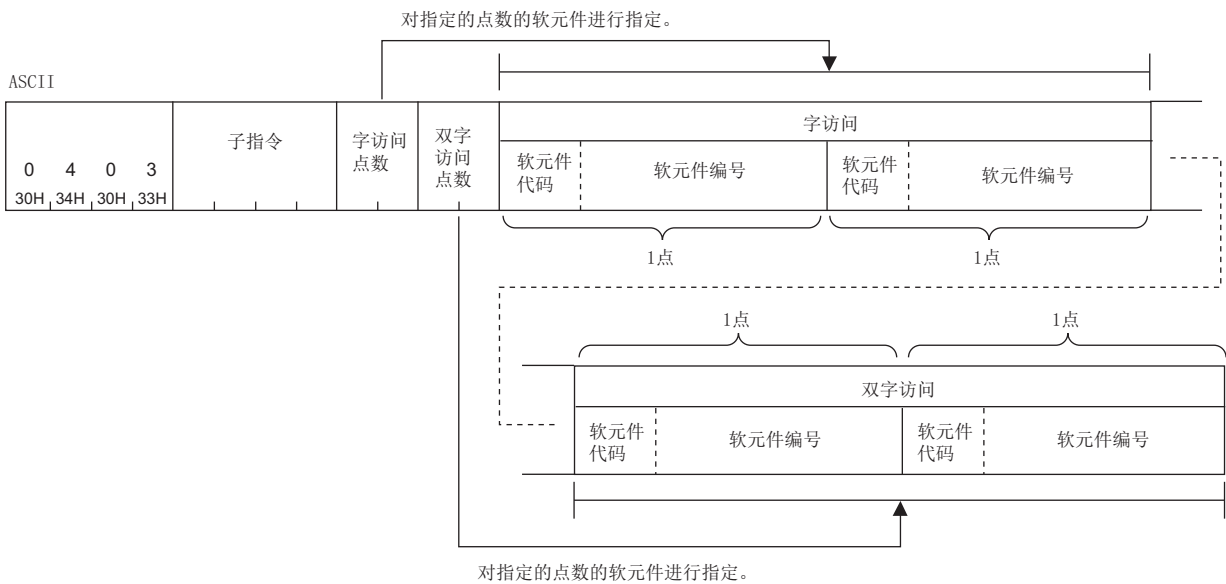
Read Random(指令：0403)

指定软元件编号，读取软元件值。可以以不连续的软元件编号指定。

要点

对于带条件监视中的CPU模块，请勿执行Read Random指令。SLMP指令将异常结束。
此外，无条件监视中可以执行指令。

请求数据



子指令

子指令*1					二进制代码				
ASCII代码									
0	0	0	0	或	0	0	8	0	00H, 00H
30H	30H	30H	30H		30H	30H	38H	30H	80H, 00H
0	0	0	2	或	0	0	8	2	02H, 00H
30H	30H	30H	32H		30H	30H	38H	32H	82H, 00H

*1 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。(190页 通过软元件的扩展指定进行读取、写入)

■字访问点数、双字访问点数

指定读取的软元件点数。

☞ 33页 软元件点数

☞ 48页 通信示例

子指令	项目	内容	点数
0002	字访问点数	指定以1字为单位进行访问时的点数。 位软元件将变为以16点为单位，字软元件将变为以1字为单位。	1≤字访问点数+双字访问点数≤96
	双字访问点数	指定以2字为单位进行访问时的点数。 位软元件将变为以32点为单位，字软元件将变为以2字为单位。	
0000	字访问点数	指定以1字为单位进行访问时的点数。 位软元件将变为以16点为单位，字软元件将变为以1字为单位。	1≤字访问点数+双字访问点数≤192 ^{*1}
	双字访问点数	指定以2字为单位进行访问时的点数。 位软元件将变为以32点为单位，字软元件将变为以2字为单位。	

^{*1} 指定高性能型CPU模块的文件寄存器(ZR)的情况下，将作为访问点数×2进行计算。此外，使用子指令的008□的情况下，将作为访问点数×2进行计算。

■软元件代码、软元件编号

以字访问、双字访问的顺序指定读取的软元件。

☞ 29页 软元件代码

☞ 32页 起始软元件编号(软元件编号)

项目	内容
字访问	指定通过“字访问点数”指定的点数的软元件。将“字访问点数”置为了0点的情况下无需指定。
双字访问	指定通过“双字访问点数”指定的点数的软元件。将“双字访问点数”置为了0点的情况下无需指定。

响应数据

以16进制数存储已读取的软元件值。根据ASCII代码及二进制代码，数据的排列有所不同。

☞ 34页 读取数据、写入数据

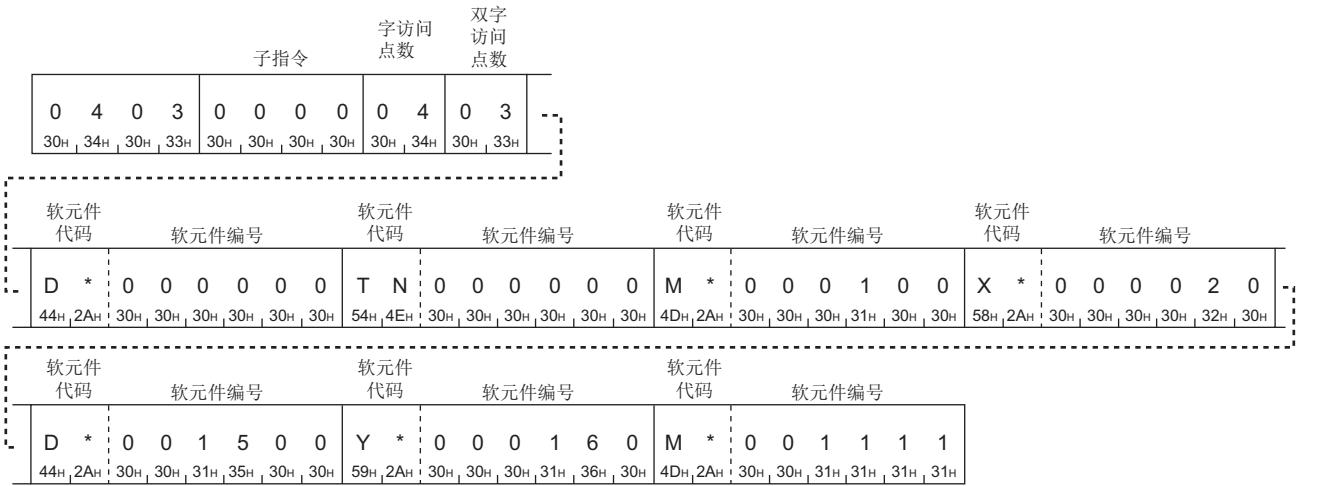
☞ 48页 通信示例

字访问点数的数据		双字访问点数的数据	
字访问		双字访问	
读取数据1	读取数据2	读取数据1	读取数据2

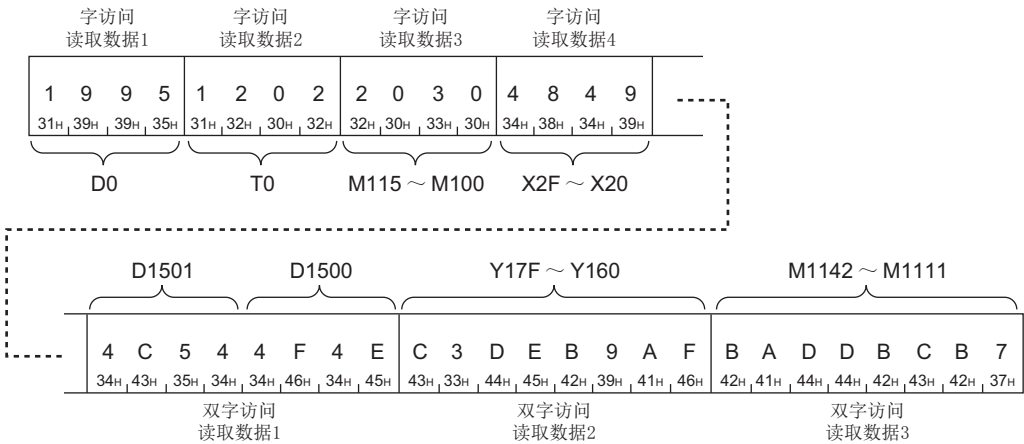
通信示例

通过字访问读取D0、T0、M100~M115、X20~X2F，通过双字访问读取D1500~D1501、Y160~Y17F、M1111~M1142。
作为存储D0=6549(1995H)、T0=4610(1202H)、D1500=20302(4F4EH)、D1501=19540(4C54H)的软元件。

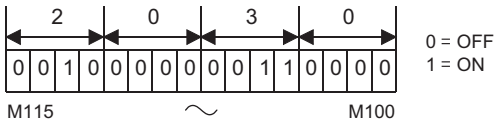
■以ASCII代码进行数据通信时
(请求数据)



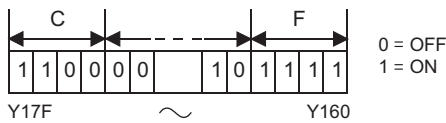
(响应数据)



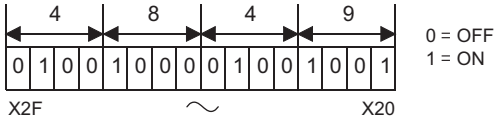
字访问读取数据3



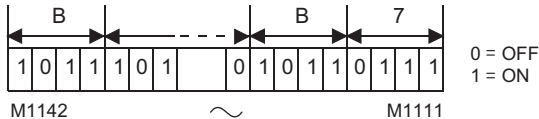
双字访问读取数据2



字访问读取数据4

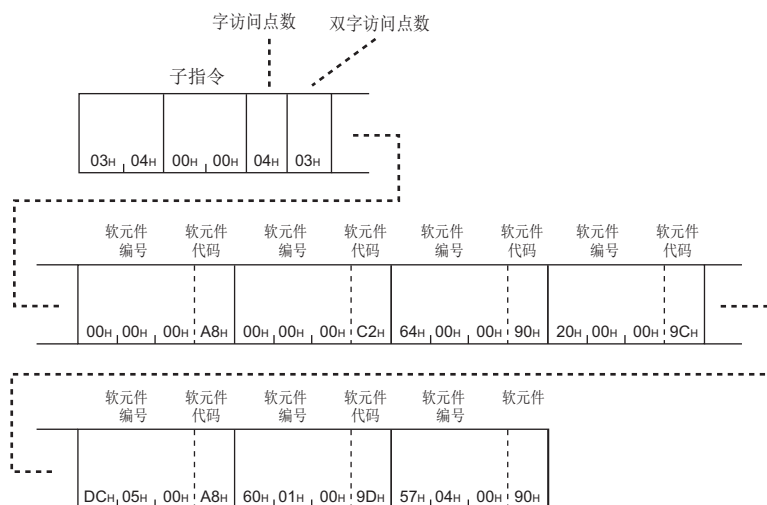


双字访问读取数据3

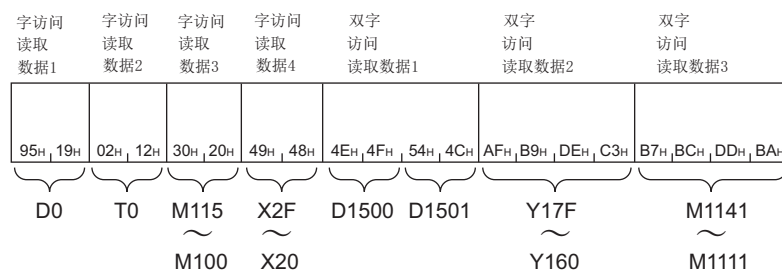


■以二进制代码进行数据通信时

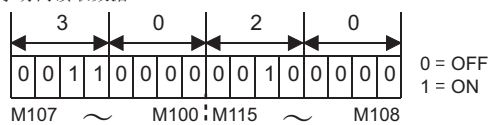
(请求数据)



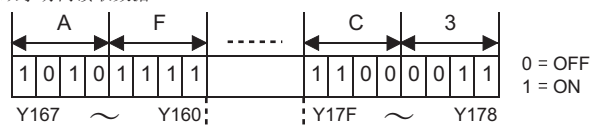
(响应数据)



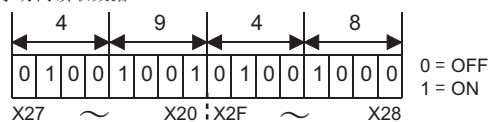
字访问读取数据3



双字访问读取数据2



字访问读取数据4



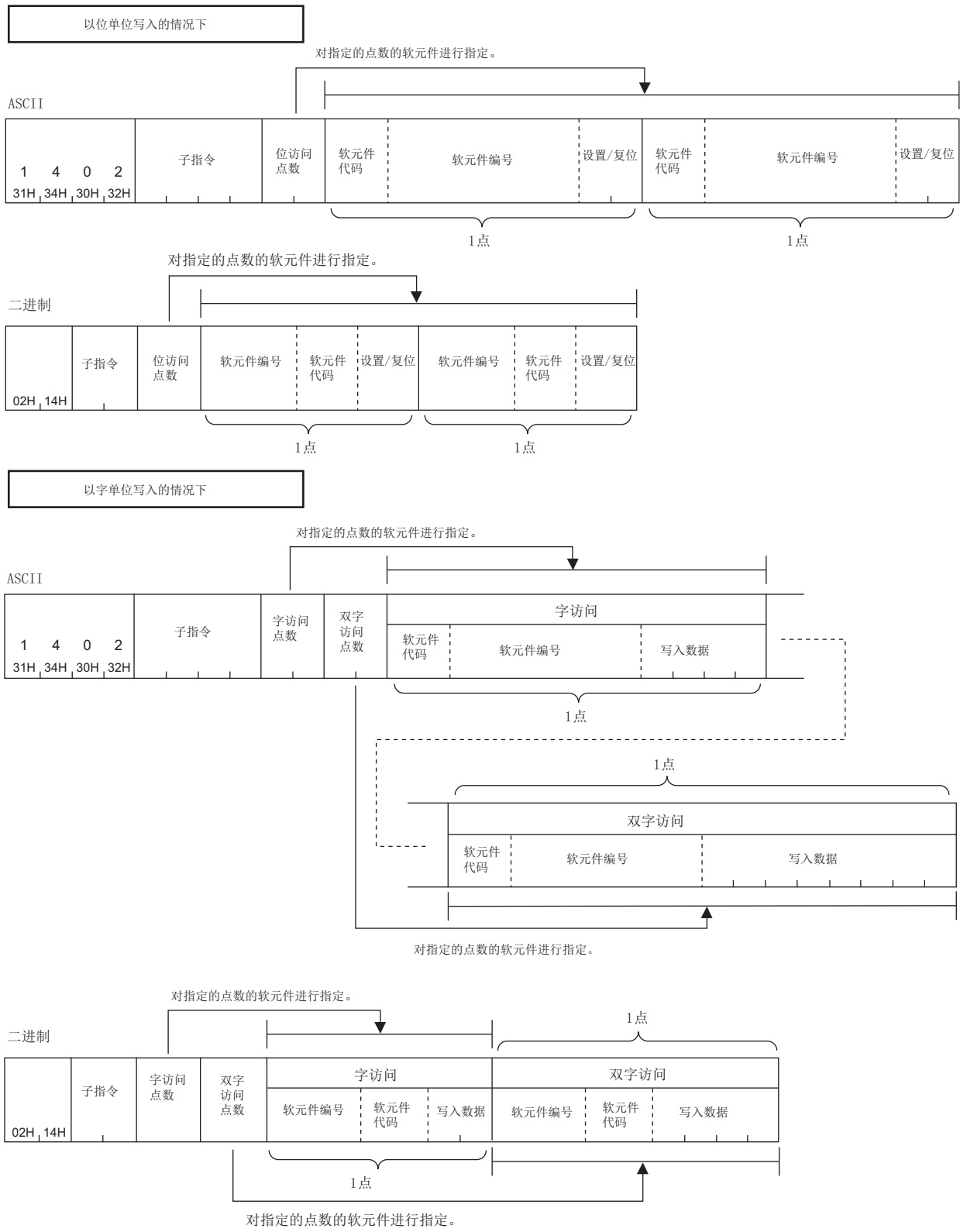
双字访问读取数据3



Write Random(指令: 1402)

指定软元件编号，将值写入到软元件中。可以以不连续的软元件编号指定。

请求数据



■子指令

项目	子指令 ^{*1}	
	ASCII代码	二进制代码
以位单位写入的情况下	0 0 0 1 30H, 30H, 30H, 31H	或
	0 0 8 1 30H, 30H, 38H, 31H	01H, 00H 或 81H, 00H
以字单位写入的情况下	0 0 0 3 30H, 30H, 30H, 33H	或
	0 0 8 3 30H, 30H, 38H, 33H	03H, 00H 或 83H, 00H
	0 0 0 0 30H, 30H, 30H, 30H	或
	0 0 8 0 30H, 30H, 38H, 30H	00H, 00H 或 80H, 00H
	0 0 0 2 30H, 30H, 30H, 32H	或
	0 0 8 2 30H, 30H, 38H, 32H	02H, 00H 或 82H, 00H

^{*1} 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。（☞ 190页 通过软元件的扩展指定进行读取、写入）

■位访问点数、字访问点数、双字访问点数

指定写入的软元件点数。

☞ 33页 软元件点数

☞ 52页 通信示例(以位单位写入的情况下)

子指令	项目	内容	点数
0003	位访问点数	以1点为单位指定位软元件点数。	1~94
0002	字访问点数	指定以1字为单位进行访问时的点数。 位软元件将变为以16点为单位，字软元件将变为以1字为单位。	1≤字访问点数×12+双字访问点数×14≤960
	双字访问点数	指定以2字为单位进行访问时的点数。 位软元件将变为以32点为单位，字软元件将变为以2字为单位。	
0001	位访问点数	以1点为单位指定位软元件点数。	1~188
0000	字访问点数	指定以1字为单位进行访问时的点数。 位软元件将变为以16点为单位，字软元件将变为以1字为单位。	1≤字访问点数×12+双字访问点数×14≤1920 ^{*1}
	双字访问点数	指定以2字为单位进行访问时的点数。 位软元件将变为以32点为单位，字软元件将变为以2字为单位。	

^{*1} 访问目标为MELSEC iQ-R系列模块的情况下，且使用子指令的008□时，将作为访问点数×2进行计算。

■软元件代码、软元件编号、写入数据

指定写入的软元件。

☞ 29页 软元件代码

☞ 32页 起始软元件编号(软元件编号)

☞ 34页 读取数据、写入数据

写入数据以16进制数指定。

项目	内容
字访问	指定通过“字访问点数”指定的点数的软元件。将“字访问点数”置为了0点的情况下无需指定。
双字访问	指定通过“双字访问点数”指定的点数的软元件。将“双字访问点数”置为了0点的情况下无需指定。

■安装/复位

指定位软元件的ON/OFF。

项目	子指令	写入数据		备注
		置为ON时	置为OFF时	
ASCII代码	0003 0002	“ 0001 ”	“ 0000 ”	从“ 0 ”开始按顺序4位发送
	0001 0000	“ 01 ”	“ 00 ”	从“ 0 ”开始按顺序2位发送
二进制代码	0003 0002	0100H	0000H	发送左述2字节的数值
	0001 0000	01H	00H	发送左述1字节的数值

响应数据

无Write Random指令的响应数据。

通信示例(以位单位写入的情况下)

将M50置为OFF，将Y2F置为ON。

■以ASCII代码进行数据通信时
(请求数据)

子指令				位访问 点数	软元件 代码	软元件编号				设置/ 复位	软元件 代码	软元件编号				设置/ 复位					
1	4	0	2	0 0 0 1	0 2	M *	0	0	0	0	5	0	0 0	Y *	0	0	0	0	2	F	0 1
31H, 34H, 30H, 32H	30H, 30H, 30H, 31H	30H, 32H	4DH, 2AH	30H, 30H, 30H, 30H, 35H, 30H	30H, 30H	59H, 2AH	30H, 30H, 30H, 30H, 32H, 46H	30H, 31H													

■以二进制代码进行数据通信时
(请求数据)

子指令		位访问点数	软件代码		设置/复位	软件代码		设置/复位						
		软件编号			软件编号									
02H	14H	01H	00H	02H	32H	00H	00H	90H	00H	2FH	00H	00H	9DH	01H

通信示例(以字单位写入的情况下)

按照下述方式，将值写入到软元件中。

项目	写入软元件
字访问	D0、D1、M100～M115、X20～X2F
双字访问	D1500～D1501、Y160～Y17F、M1111～M1142

■以ASCII代码进行数据通信时

(请求数据)

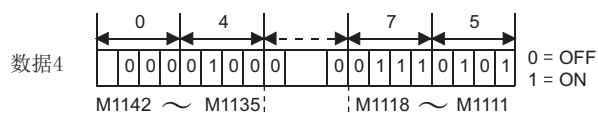
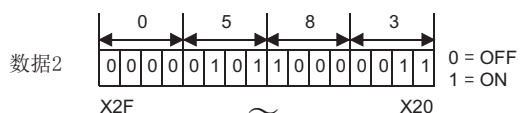
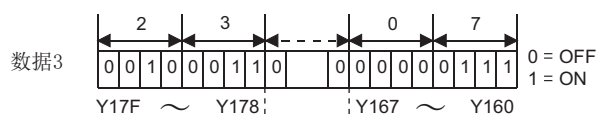
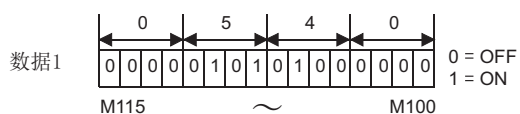
子指令				字访问 点数	双字访问 点数
1	4	0	2	0 0 0 0	0 4 0 3
31 _H	34 _H	30 _H	32 _H	30 _H 30 _H 30 _H 30 _H	30 _H 34 _H 30 _H 33 _H

软元件 代码	软元件编号	写入数据	软元件 代码	软元件编号	写入数据
D *	0 0 0 0 0 0	0 5 5 0	D *	0 0 0 0 0 1	0 5 7 5
44 _H 2A _H	30 _H 30 _H 30 _H 30 _H 30 _H 30 _H	30 _H 35 _H 35 _H 30 _H	44 _H 2A _H	30 _H 30 _H 30 _H 30 _H 30 _H 31 _H	30 _H 35 _H 37 _H 35 _H

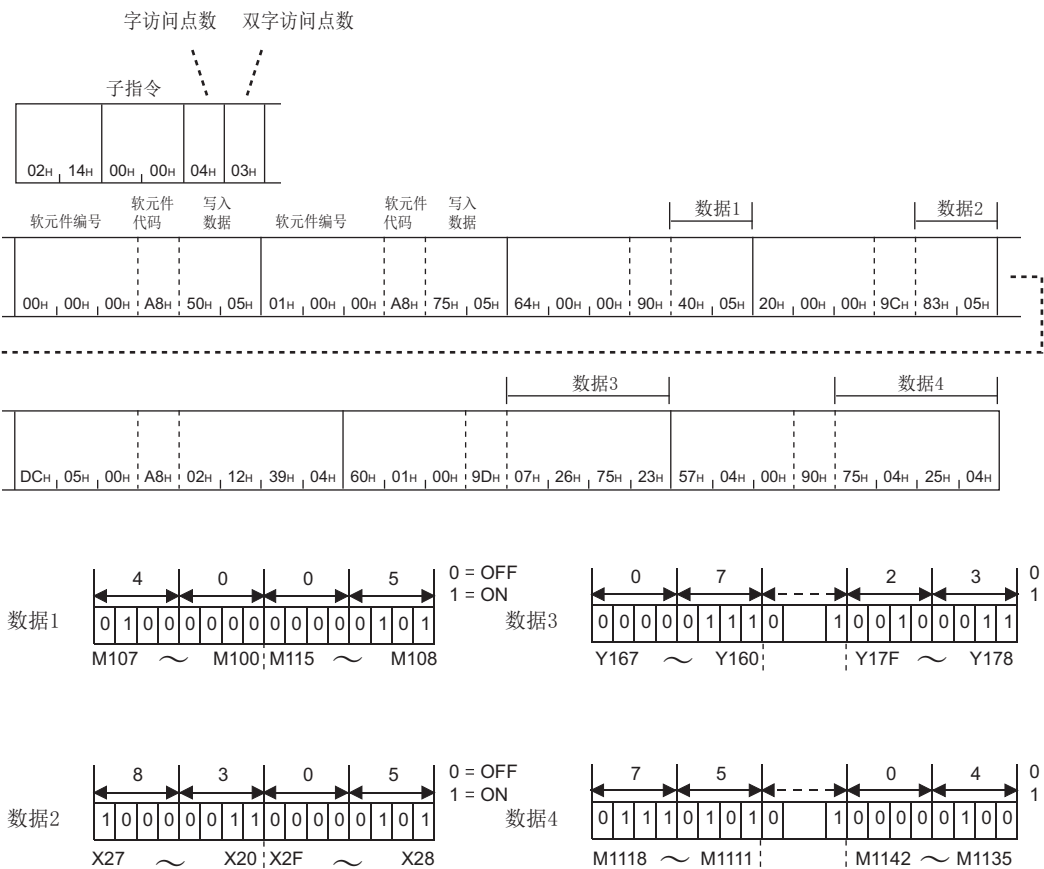
数据1				数据2			
M *	0 0 0 1 0 0	0 5 4 0	X *	0 0 0 0 2 0	0 5 8 3		
4D _H 2A _H	30 _H 30 _H 30 _H 31 _H 30 _H 30 _H	30 _H 35 _H 34 _H 30 _H	58 _H 2A _H	30 _H 30 _H 30 _H 30 _H 32 _H 30 _H	30 _H 35 _H 38 _H 33 _H		

数据3											
D *	0 0 1 5 0 0	0 4 3 9 1 2 0 2	Y *	0 0 0 1 6 0	2 3 7 5 2 6 0 7						
44 _H 2A _H	30 _H 30 _H 31 _H 35 _H 30 _H 30 _H	30 _H 34 _H 33 _H 39 _H 31 _H 32 _H 30 _H 32 _H	59 _H 2A _H	30 _H 30 _H 30 _H 31 _H 36 _H 30 _H	32 _H 33 _H 37 _H 35 _H 32 _H 36 _H 30 _H 37 _H						

数据4											
M *	0 0 1 1 1 1	0 4 2 5 0 4 7 5									
4D _H 2A _H	30 _H 30 _H 31 _H 31 _H 31 _H 31 _H	30 _H 34 _H 32 _H 35 _H 30 _H 34 _H 37 _H 35 _H									



■以二进制代码进行数据通信时 (请求数据)



Entry Monitor Device(指令: 0801)

通过Execute Monitor(指令: 0802)登录读取的软元件。通过预先登录读取的软元件, 由于可以缩短读取时的请求报文, 因此可以降低线路负载。

对于Entry Monitor Device(指令: 0801)及Execute Monitor(指令: 0802), 按下述方式使用。

1. 监视软元件登录

通过Entry Monitor Device(指令: 0801), 登录读取的软元件。

2. 监视执行

如果Execute Monitor(指令: 0802), 将通过Entry Monitor Device(指令: 0801)从登录的软元件读取值。(59页 Execute Monitor(指令: 0802))

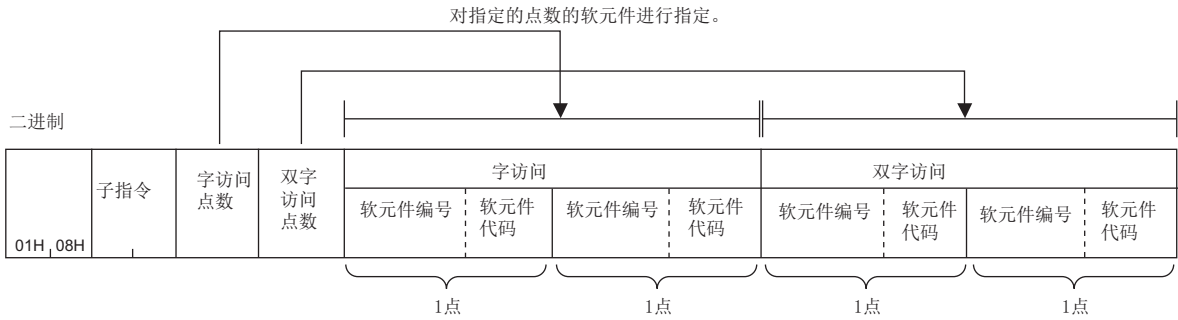
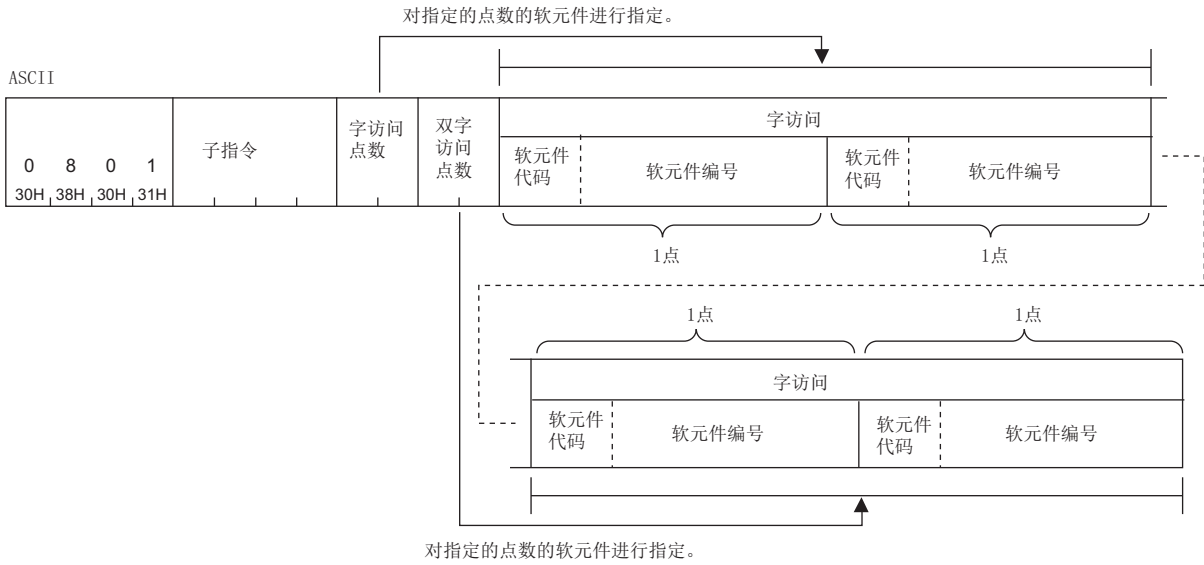
3. 监视软元件更改

更改读取的软元件的情况下, 通过Entry Monitor Device(指令: 0801), 更改读取的软元件。

要点

- 对于带条件监视中的CPU模块, 请勿执行Entry Monitor Device指令。SLMP指令将异常结束。此外, 无条件监视中可以执行指令。
- 进行访问目标再启动的情况下, 登录内容将被消去。应再次执行Entry Monitor Device, 登录读取的软元件。

请求数据



子指令*1

子指令 ^{*1}																					
ASCII代码	二进制代码																				
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	0	0	8	0	30H	30H	38H	30H	<table><tr><td></td></tr><tr><td>00H, 00H</td></tr></table> 或 <table><tr><td></td></tr><tr><td>80H, 00H</td></tr></table>		00H, 00H		80H, 00H
0	0	0	0																		
30H	30H	30H	30H																		
0	0	8	0																		
30H	30H	38H	30H																		
00H, 00H																					
80H, 00H																					
<table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>32H</td></tr></table>	0	0	0	2	30H	30H	30H	32H	0	0	8	2	30H	30H	38H	32H	<table><tr><td></td></tr><tr><td>02H, 00H</td></tr></table> 或 <table><tr><td></td></tr><tr><td>82H, 00H</td></tr></table>		02H, 00H		82H, 00H
0	0	0	2																		
30H	30H	30H	32H																		
0	0	8	2																		
30H	30H	38H	32H																		
02H, 00H																					
82H, 00H																					

*1 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。(190页 通过软元件的扩展指定进行读取、写入)

■字访问点数、双字访问点数

指定读取的软元件点数。

☞ 33页 软元件点数

☞ 58页 通信示例

子指令	项目	内容	点数
0002	字访问点数	指定以1字为单位进行访问时的点数。 位软元件将变为以16点为单位，字软元件将变为以1字为单位。	1≤字访问点数+双字访问点数≤96
	双字访问点数	指定以2字为单位进行访问时的点数。 位软元件将变为以32点为单位，字软元件将变为以2字为单位。	
0000	字访问点数	指定以1字为单位进行访问时的点数。 位软元件将变为以16点为单位，字软元件将变为以1字为单位。	1≤字访问点数+双字访问点数≤192 ^{*1}
	双字访问点数	指定以2字为单位进行访问时的点数。 位软元件将变为以32点为单位，字软元件将变为以2字为单位。	

*1 指定高性能型CPU模块的文件寄存器(ZR)的情况下，将作为访问点数×2进行计算。此外，使用子指令的008□的情况下，将作为访问点数×2进行计算。

■软元件代码、软元件编号

以字访问、双字访问的顺序指定读取的软元件。

☞ 29页 软元件代码

☞ 32页 起始软元件编号(软元件编号)

项目	内容
字访问	指定通过“字访问点数”指定的点数的软元件。将“字访问点数”置为了0点的情况下无需指定。
双字访问	指定通过“双字访问点数”指定的点数的软元件。将“双字访问点数”置为了0点的情况下无需指定。

要点

不可以指定定时器、长定时器、累计定时器、长累计定时器、计数器、长计数器的触点及线圈。

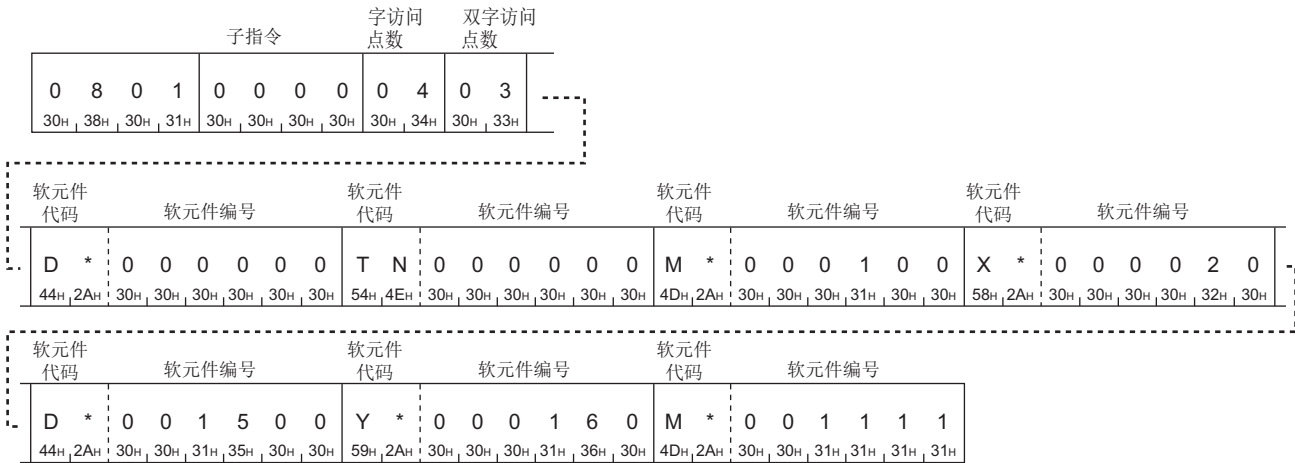
响应数据

无Entry Monitor Device指令的响应数据。

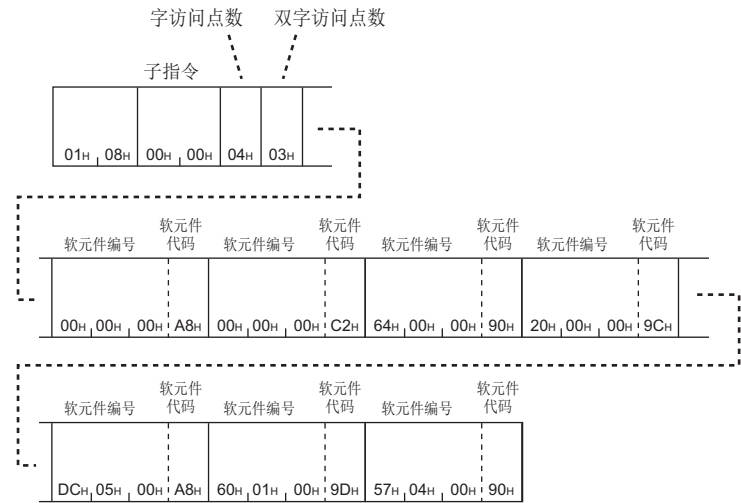
通信示例

按照通过字访问读取D0、T0、M100～M115、X20～X2F，通过双字访问读取D1500～D1501、Y160～Y17F、M1111～M1142的登录示例如下所示。

■以ASCII代码进行数据通信时
(请求数据)



■以二进制代码进行数据通信时
(请求数据)



Execute Monitor(指令：0802)

通过Entry Monitor Device(指令：0801)读取已登录的软元件值。

要点

- Execute Monitor(指令：0802)的执行前，应通过Entry Monitor Device(指令：0801)，登录读取的软元件。对于Entry Monitor Device(指令：0801)的状况下执行指令将变为出错。(参照 55页 Entry Monitor Device(指令：0801))
- 进行访问目标再启动的情况下，登录内容将被消去。应再次执行Entry Monitor Device(指令：0801)，登录读取的软元件。

请求数据

ASCII

0	8	0	2	0	0	0	0
30H	38H	30H	32H	30H	30H	30H	30H

二进制

02H	08H
00H	00H

响应数据

字访问点数的数据		双字访问点数的数据	
字访问		双字访问	
读取数据1	读取数据2	读取数据1	读取数据2

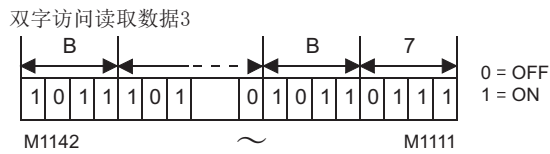
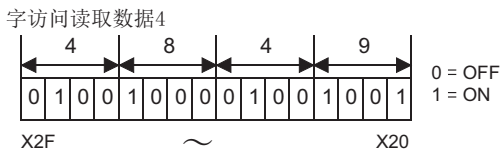
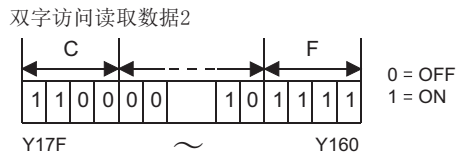
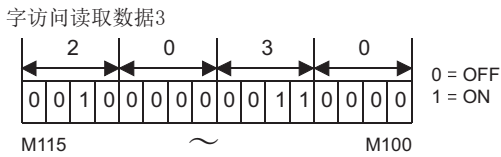
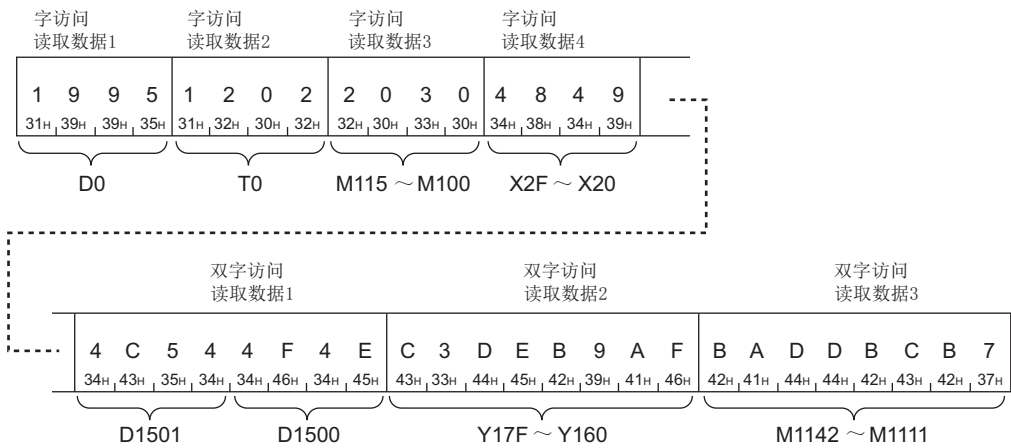
通过58页通信示例

58页 通信示例中登录的软元件，读取值。
作为存储D0=6549(1995H)、T0=4610(1202H)、D1500=20302(4F4EH)、D1501=19540(4C54H)的软元件。

■以ASCII代码进行数据通信时
(请求数据)

0	8	0	2	0	0	0	0
30H	38H	30H	32H	30H	30H	30H	30H

(响应数据)

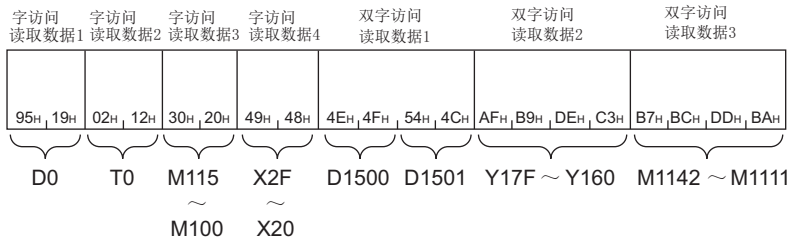


■以二进制代码进行数据通信时

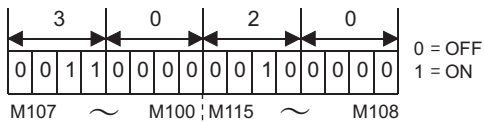
(请求数据)

02H, 08H	00H, 00H
----------	----------

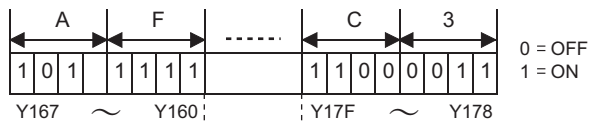
(响应数据)



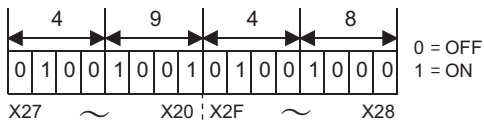
字访问读取数据3



双字访问读取数据2



字访问读取数据4



双字访问读取数据3



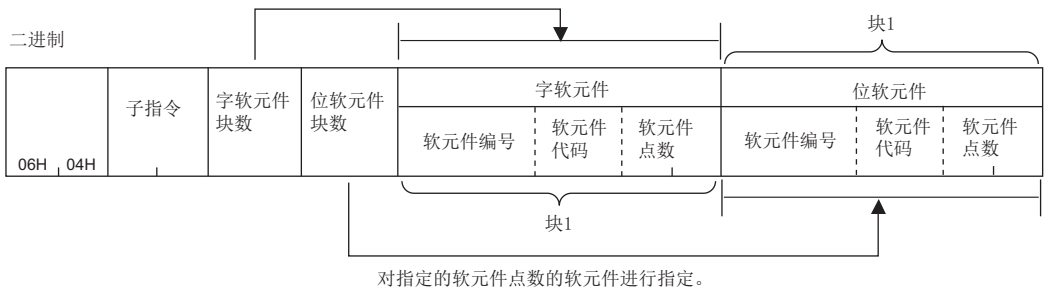
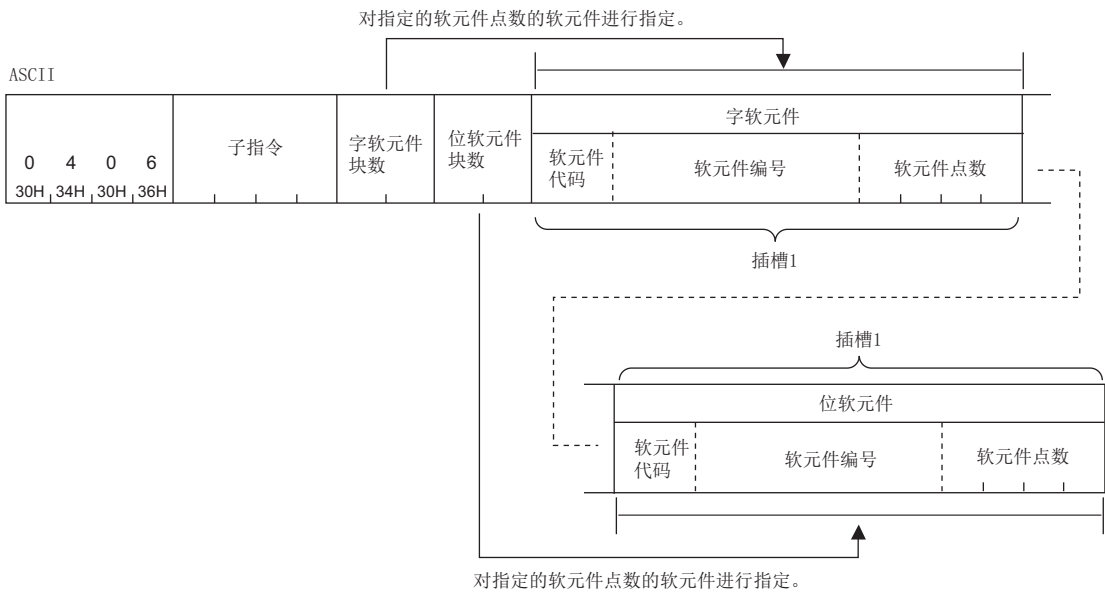
Read Block(指令：0406)

将字软元件及位软元件(1点为16位)的n点作为1个块，指定多个块后读取。可以以不连续的软元件编号指定。

要点

对于通用型QCPU及LCPU进行通信的情况下，如果通过CPU模块的“服务处理设置”置为“指定服务处理次数”以外，有可能发生数据背离。为了防止数据背离，应设置为“指定服务处理次数”。

请求数据



子指令

子指令 ^{*1}																									
ASCII代码	二进制代码																								
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	0	0	8	0	30H	30H	38H	30H	<table><tr><td></td><td></td></tr><tr><td>00H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>80H</td><td>00H</td></tr></table>			00H	00H			80H	00H
0	0	0	0																						
30H	30H	30H	30H																						
0	0	8	0																						
30H	30H	38H	30H																						
00H	00H																								
80H	00H																								
<table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>32H</td></tr></table>	0	0	0	2	30H	30H	30H	32H	0	0	8	2	30H	30H	38H	32H	<table><tr><td></td><td></td></tr><tr><td>02H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>82H</td><td>00H</td></tr></table>			02H	00H			82H	00H
0	0	0	2																						
30H	30H	30H	32H																						
0	0	8	2																						
30H	30H	38H	32H																						
02H	00H																								
82H	00H																								

^{*1} 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。(190页 通过软元件的扩展指定进行读取、写入)

■字软元件块数、位软元件块数

以16进制数指定读取的软元件的块数。

子指令	项目	内容	点数
0002	字软元件块数	指定读取的字软元件的块数。	字软元件块数+位软元件块数≤60
	位软元件块数	指定读取的位软元件的块数。	
0000	字软元件块数	指定读取的字软元件的块数。	字软元件块数+位软元件块数≤120 ^{*1}
	位软元件块数	指定读取的位软元件的块数。	

*1 访问目标为MELSEC iQ-R系列模块的情况下，且也使用子指令的008□的情况下，将作为块数×2进行计算。

■软元件代码、软元件编号、软元件点数

指定读取的软元件。

☞ 29页 软元件代码

☞ 32页 起始软元件编号(软元件编号)

☞ 33页 软元件点数

但是，不可以指定下述软元件的触点、线圈、当前值。

- 长定时器(LTS、LTC、LTN)
- 长累计定时器(LSTS、LSTC、LSTN)
- 长计数器(LCS、LCC、LCN)

软元件点数应按满足下述条件的方式指定。

- 字软元件的各块的合计点数+位软元件的各块的合计点数≤960

此外，应按字软元件、位软元件的顺序指定。

项目	内容
字软元件	指定通过“字软元件块数”指定的点数的软元件。将“字软元件块数”置为了0点的情况下无需指定。
位软元件	指定通过“位软元件块数”指定的点数的软元件。将“位软元件块数”置为了0点的情况下无需指定。

要点

指定定时器、累计定时器、计数器的触点及线圈的情况下，应使用位软元件块。

响应数据

以16进制数存储已读取的软元件值。根据ASCII代码及二进制代码，数据的排列有所不同。(☞ 34页 读取数据、写入数据)

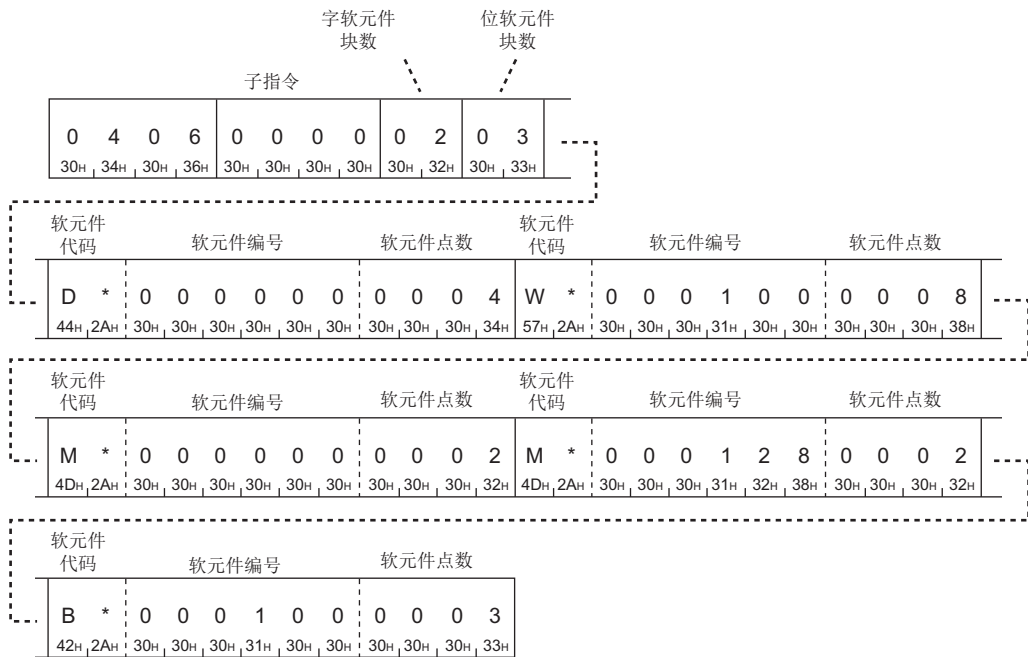
字软元件块数的数据		位软元件块数的数据	
字软元件		位软元件	
第1块的数据	第2块的数据	第1块的数据	第2块的数据

通信示例

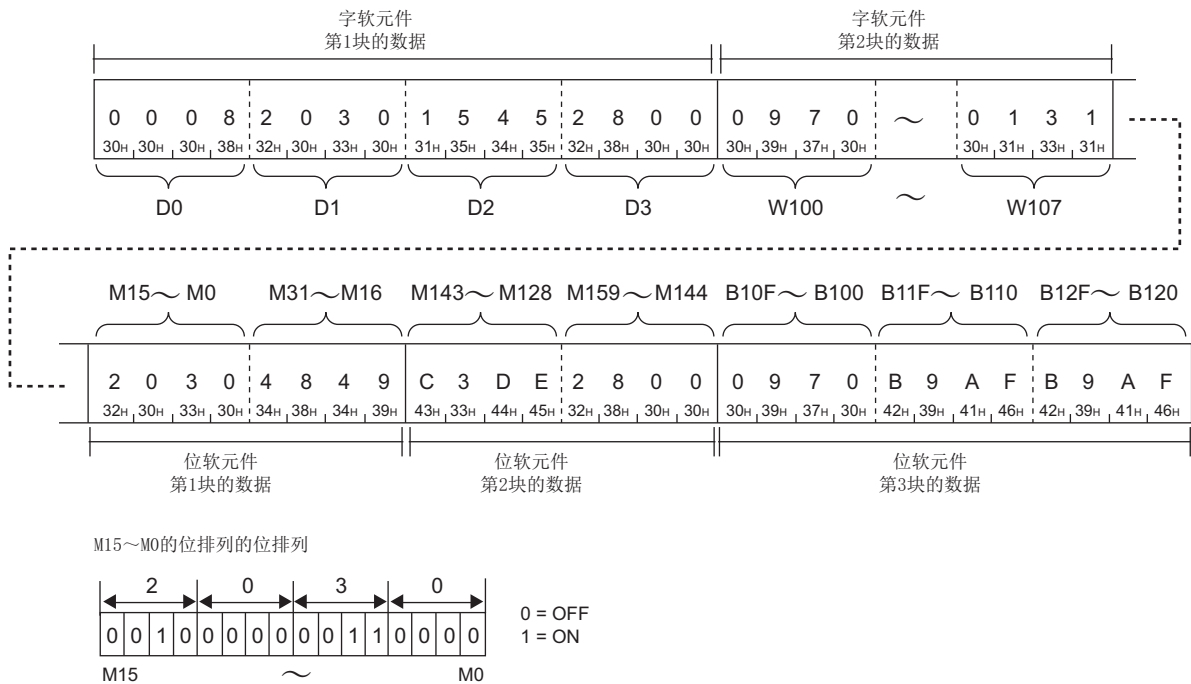
按下述方式通过软元件读取值。

项目	读取内容
字软元件	• 块1: D0~D3(4点) • 块2: W100~W107(8点)
位软元件	• 块1: M0~M31(2点) • 块2: M128~M159(2点) • 块3: B100~B12F(3点)

■以ASCII代码进行数据通信时
(请求数据)

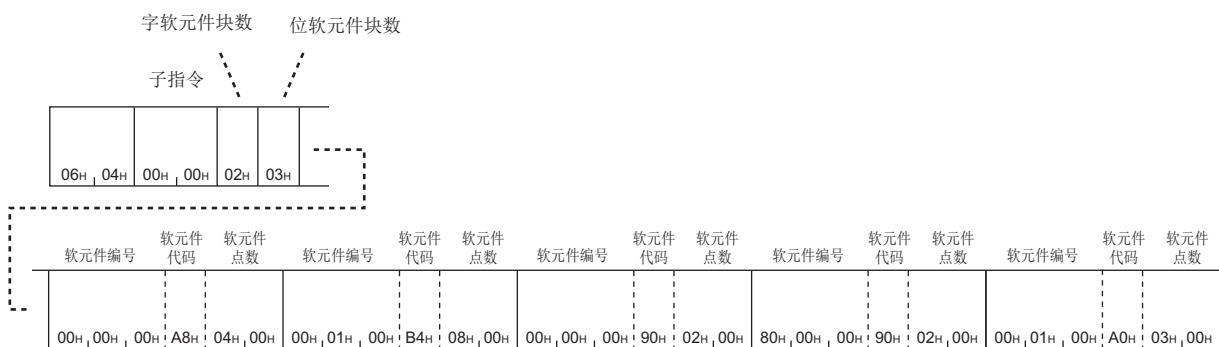


(响应数据)

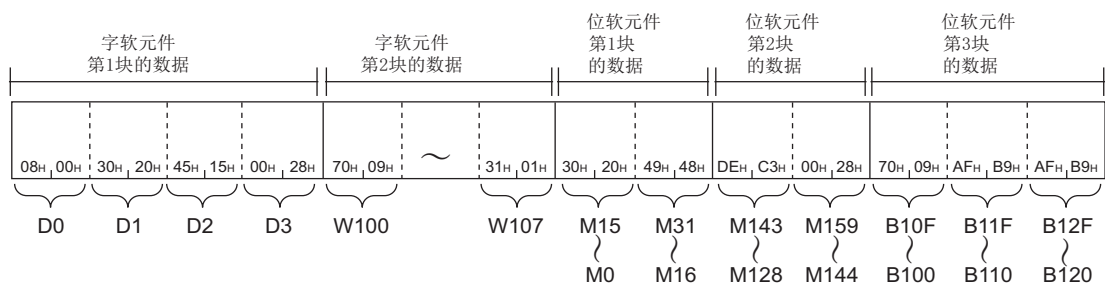


■以二进制代码进行数据通信时

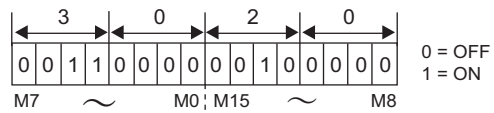
(请求数据)



(响应数据)



M15~M0的位排列



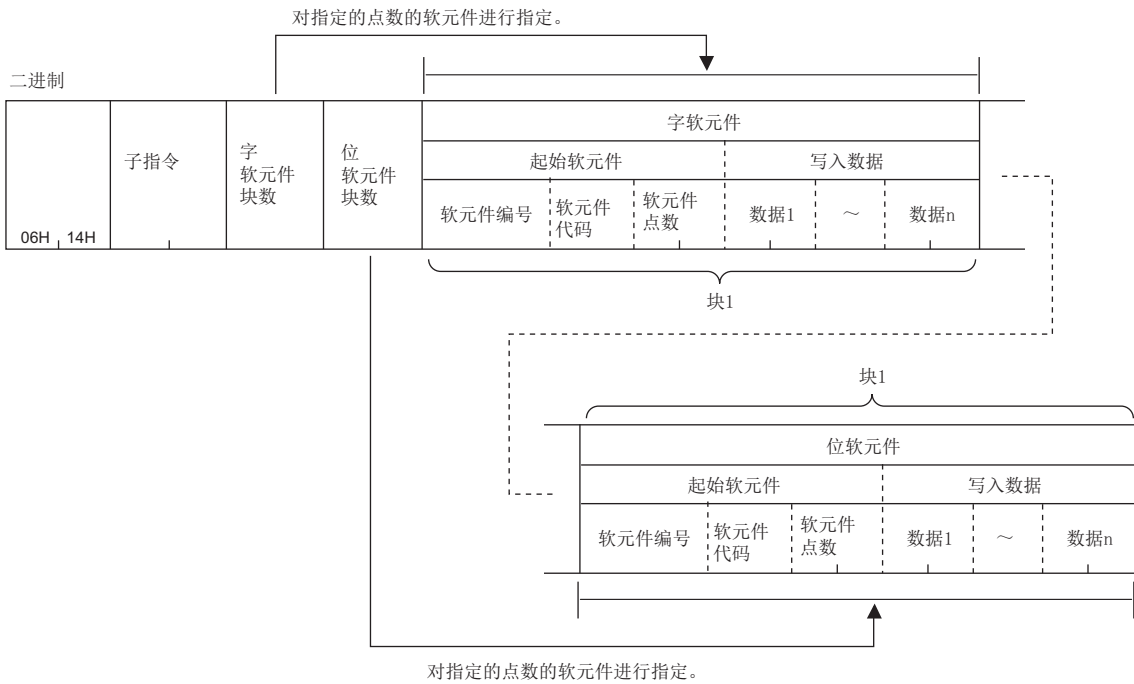
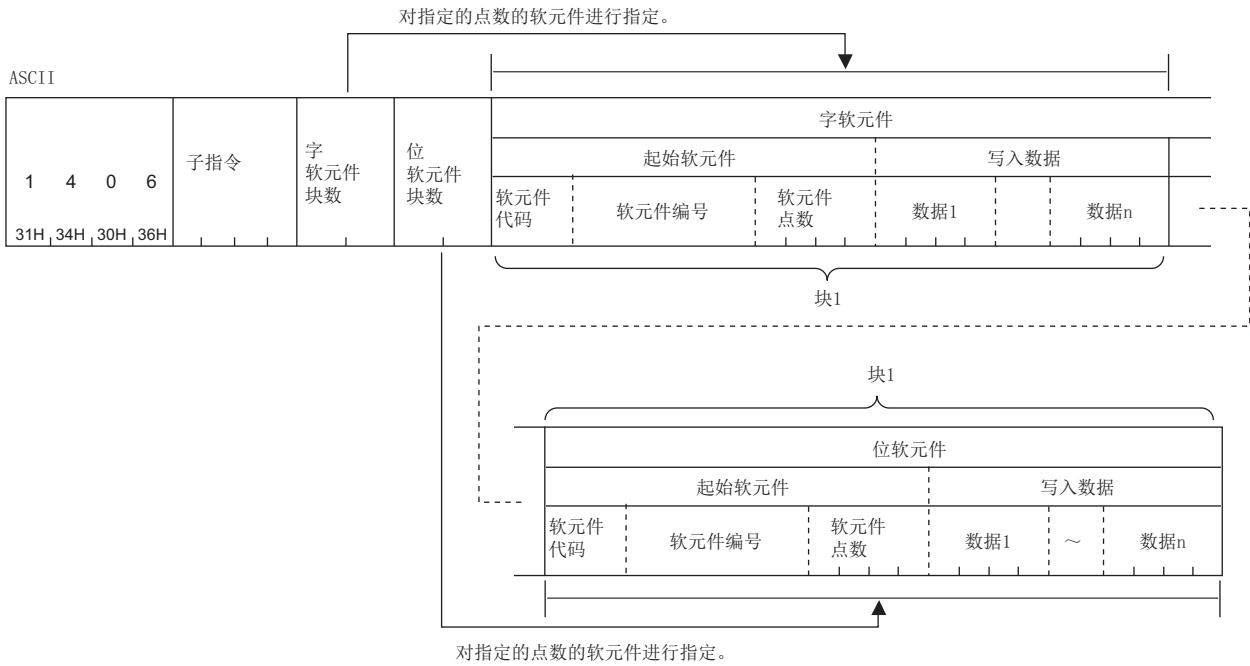
Write Block(指令: 1406)

将字软元件及位软元件(1点为16位)的n点作为1个块，指定多个块后写入。可以以不连续的软元件编号指定。

要点

对于通用型QCPU及LCPU进行通信的情况下，如果通过CPU模块的“服务处理设置”置为“指定服务处理次数”以外，有可能发生数据背离。为了防止数据背离，应设置为“指定服务处理次数”。

请求数据



■子指令

子指令 ^{*1}																									
ASCII代码	二进制代码																								
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	0	0	8	0	30H	30H	38H	30H	<table><tr><td></td><td></td></tr><tr><td>00H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>80H</td><td>00H</td></tr></table>			00H	00H			80H	00H
0	0	0	0																						
30H	30H	30H	30H																						
0	0	8	0																						
30H	30H	38H	30H																						
00H	00H																								
80H	00H																								
<table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table> 或 <table><tr><td>0</td><td>0</td><td>8</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>38H</td><td>32H</td></tr></table>	0	0	0	2	30H	30H	30H	32H	0	0	8	2	30H	30H	38H	32H	<table><tr><td></td><td></td></tr><tr><td>02H</td><td>00H</td></tr></table> 或 <table><tr><td></td><td></td></tr><tr><td>82H</td><td>00H</td></tr></table>			02H	00H			82H	00H
0	0	0	2																						
30H	30H	30H	32H																						
0	0	8	2																						
30H	30H	38H	32H																						
02H	00H																								
82H	00H																								

^{*1} 子指令的008□在访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下使用。将子指令置为了008□时，报文格式有所不同。(190页 通过软元件的扩展指定进行读取、写入)

■字软元件块数、位软元件块数

指定写入的软元件的块数。(33页 软元件点数)

子指令	项目	内容	点数
0002	字软元件块数	指定写入的字软元件的块数。	字软元件块数+位软元件块数≤60
	位软元件块数	指定写入的位软元件的块数。	
0000	字软元件块数	指定写入的字软元件的块数。	字软元件块数+位软元件块数≤120 ^{*1}
	位软元件块数	指定写入的位软元件的块数。	

^{*1} 访问目标为MELSEC iQ-R系列模块的情况下，且使用子指令的008□的情况下，将作为块数×2进行计算。

■软元件代码、软元件编号、软元件点数

指定写入的软元件。

29页 软元件代码

32页 起始软元件编号(软元件编号)

33页 软元件点数

但是，不可以指定下述软元件的触点、线圈、当前值。

- 长定时器(LTS、LTC、LTN)
- 长累计定时器(LSTS、LSTC、LSTN)
- 长计数器(LCS、LCC、LCN)

软元件点数应按满足下述条件的方式指定。

子指令	条件
0002	(字软元件块数+位软元件块数)×9+字软元件的各块合计数+位软元件的各块合计数≤960
0000	(字软元件块数+位软元件块数)×4+字软元件的各块合计数+位软元件的各块合计数≤960

此外，应按字软元件、位软元件的顺序指定。

项目	内容
字软元件	指定通过“字软元件块数”指定的点数的软元件。将“字软元件块数”置为了0点的情况下无需指定。
位软元件	指定通过“位软元件块数”指定的点数的软元件。将“位软元件块数”置为了0点的情况下无需指定。

要点

指定定时器、累计定时器、计数器的触点及线圈的情况下，应使用位软元件块。

响应数据

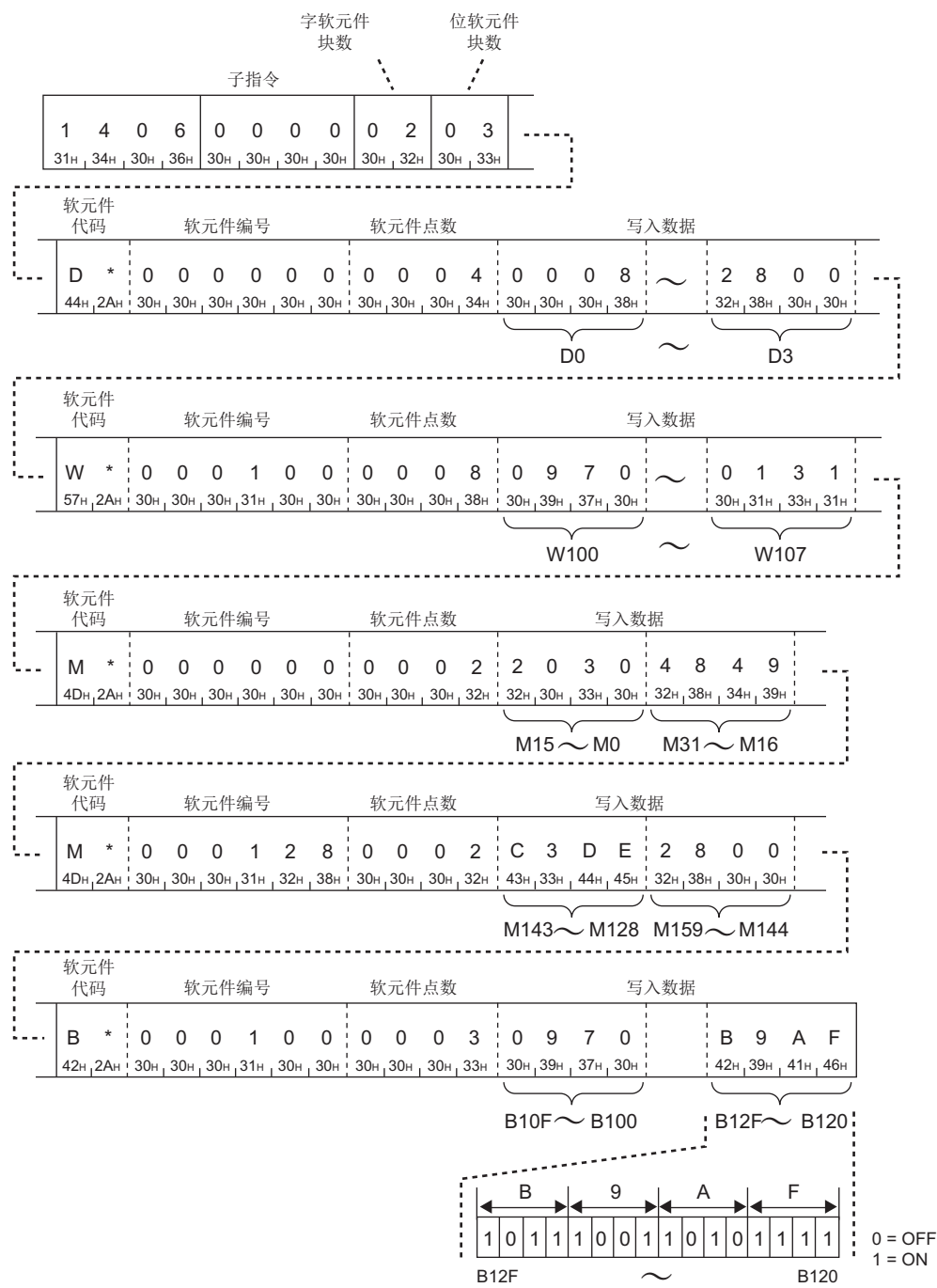
无Write Block指令的响应数据。

通信示例

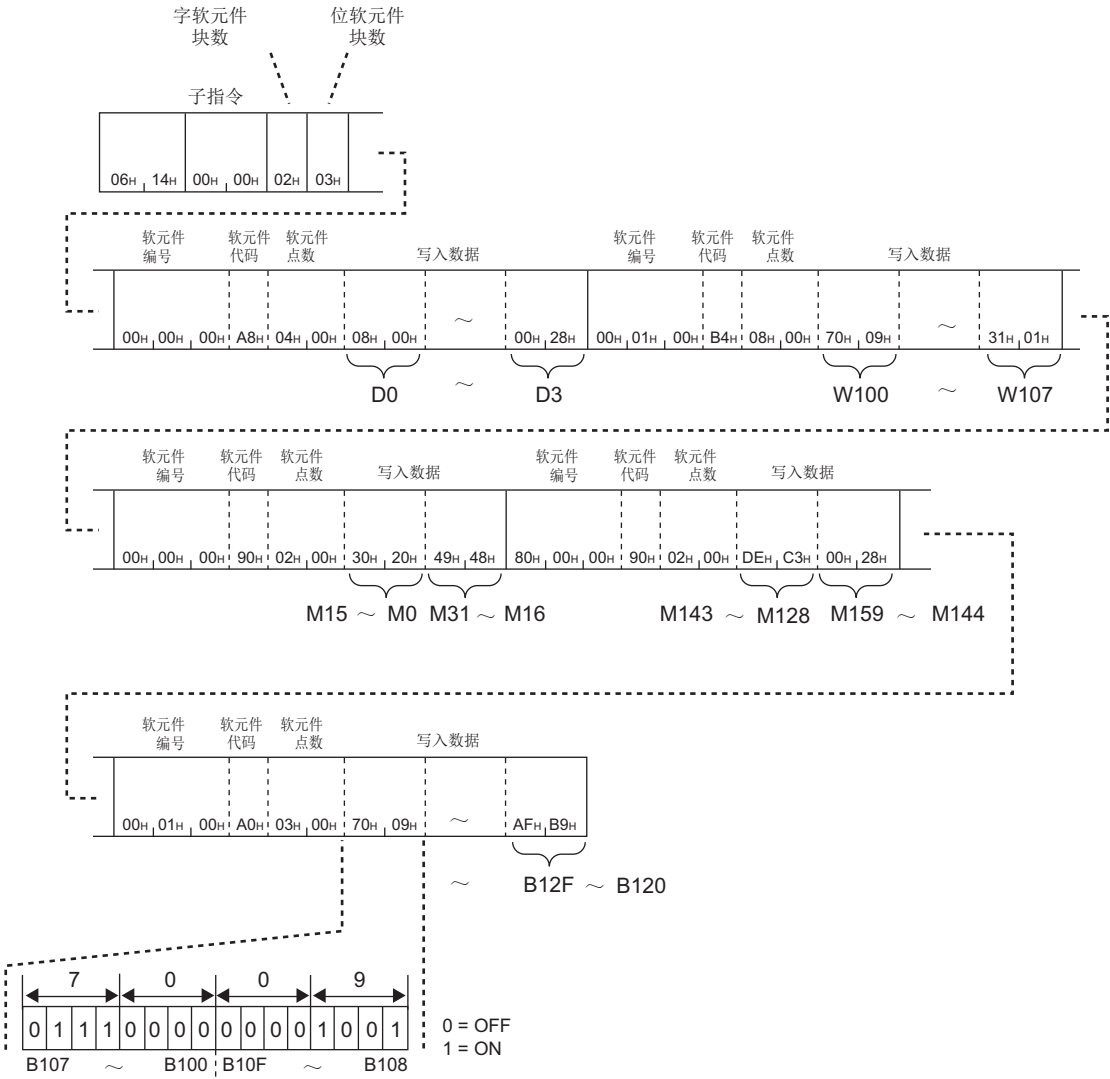
按照下述方式，将值写入到软元件中。

项目	写入内容
字软元件	• 块1: D0~D3(4点) • 块2: W100~W107(8点)
位软元件	• 块1: M0~M31(2点) • 块2: M128~M159(2点) • 块3: B100~B12F(3点)

■以ASCII代码进行数据通信时
(请求数据)



■以二进制代码进行数据通信时
(请求数据)



5.3 Label(标签访问)

以下说明使用全局标签进行数据读取及写入的指令有关内容。

限制事项

- 不可以访问局部标签。
- 不可以访问通过GX Works2设置的全局标签。
- 为了访问全局标签，需要通过GX Works3的全局标签设置编辑器将“来自于外部设备的访问”的设置项目置为有效。(默认为无效。)
- 以ASCII代码进行数据通信时，由于需要从UTF-16将标签名转换为ASCII代码，因此报文容量将增加。

指令内指定的数据

排列点数

指定进行读取或写入的排列的数。

由于发送数据最多为1920字节，因此可指定的最多点数根据数据名长而变化。

■以ASCII代码进行数据通信时

将点数转换为ASCII代码4位使用后，按照从高位字节到低位字节的顺序发送。

例

点数为3时

0	0	0	3
30 _H	30 _H	30 _H	33 _H

■以二进制代码进行数据通信时

使用表示点数的2字节的数值，按照从低位字节到高位字节的顺序发送。

例

点数为3时

03 _H	00 _H
-----------------	-----------------

读取数据点数、写入数据点数

指定进行读取或写入的标签数。

由于发送数据最多为1920字节，因此可指定的最多点数根据数据名长而变化。

■以ASCII代码进行数据通信时

将标签数转换为ASCII代码4位使用后，按照从高位字节到低位字节的顺序发送。

例

标签数为3时

0	0	0	3
30 _H	30 _H	30 _H	33 _H

■以二进制代码进行数据通信时

使用表示标签数的2字节的数值，按照从低位字节到高位字节的顺序发送。

例

标签数为3时

03 _H	00 _H
-----------------	-----------------

省略点数

指定省略定义的标签的点数。未省略定义的情况下指定0。

省略定义是指，将标签名省略为“%1”“%2”...“%n”(n: 省略点数中指定的数)进行标记。(74页 标签名)

■以ASCII代码进行数据通信时

以ASCII代码进行数据通信时，将省略点数转换为4位后，按照从高位字节到低位字节的顺序发送。

例

省略点数为3时

0	0	0	3
30 _H	30 _H	30 _H	33 _H

■以二进制代码进行数据通信时

以二进制代码进行数据通信时，使用表示省略点数的2字节的数值，按照从低位字节到高位字节的顺序发送。

例

省略点数为3时

03 _H	00 _H
-----------------	-----------------

标签名长

指定通过“ 标签名 ”指定的标签名的字符数。

■以ASCII代码进行数据通信时

将字符数转换为ASCII代码4位使用后，按照从高位字节到低位字节的顺序发送。

例

字符数为8字符时

0	0	0	8
30H	30H	30H	38H

■以二进制代码进行数据通信时

使用2字节的数值，按照从低位字节到高位字节的顺序发送字符数。

例

字符数为8字符时

08H, 00H

标签名

指定标签的名称。

- 以ASCII代码进行数据通信时，将表示全局标签名的UTF-16的值转换为ASCII代码后，按照从高位字节到低位字节的顺序发送。
- 以ASCII代码进行数据通信时，对表示全局标签名的UTF-16的值按照从高位字节到低位字节的顺序进行发送。

■基本数据型的标签

指定全局标签名。

全局标签名为“AAA”的情况下，ASCII代码与二进制代码的指定示例如下所示。

标签名(UTF-16(16进制数))	A(0041)	A(0041)	A(0041)
ASCII代码(16进制数)	30303431	30303431	30303431
二进制代码(16进制数)	4100	4100	4100

■排列指定型的标签

对标签名与3次元为止的索引(要素编号)进行指定。

全局标签名为1次元排列“BBB[20]”的情况下，ASCII代码与二进制代码的指定示例如下所示。

标签名(UTF-16(16进制数))	B(0042)	B(0042)	B(0042)	[(005B)	2(0032)	0(0030)	](005D)
ASCII代码(16进制数)	30303432	30303432	30303432	30303542	30303332	30303330	30303544
二进制代码(16进制数)	4200	4200	4200	5B00	3200	3000	5D00

全局标签名为2次元排列“BBB[20,10]”的情况下，ASCII代码与二进制代码的指定示例如下所示。

标签名(UTF-16(16进制数))	B(0042)	B(0042)	B(0042)	[(005B)	2(0032)
ASCII代码(16进制数)	30303432	30303432	30303432	30303542	30303332
二进制代码(16进制数)	4200	4200	4200	5B00	3200

标签名(UTF-16(16进制数))	0(0030)	(002C)	1(0031)	0(0030)	](005D)
ASCII代码(16进制数)	30303330	30303243	30303331	30303330	30303544
二进制代码(16进制数)	3000	2C00	3100	3000	5D00

全局标签名为3次元排列“BBB[20,10,30]”的情况下，ASCII代码与二进制代码的指定示例如下所示。

标签名(UTF-16(16进制数))	B(0042)	B(0042)	B(0042)	[(005B)	2(0032)	0(0030)	,(002C)
ASCII代码(16进制数)	30303432	30303432	30303432	30303542	30303332	30303330	30303243
二进制代码(16进制数)	4200	4200	4200	5B00	3200	3000	2C00

标签名(UTF-16(16进制数))	1(0031)	0(0030)	,(002C)	3(0033)	0(0030)	](005D)
ASCII代码(16进制数)	30303331	30303330	30303243	30303333	30303330	30303544
二进制代码(16进制数)	3100	3000	2C00	3300	3000	5D00

■结构体型的标签

以半角句号结合结构体的要素名，对到最后要素指定的字符串进行指定。

全局标签名为“XXX.YYY.ZZZ”的情况下，ASCII代码与二进制代码的指定示例如下所示。

标签名(UTF-16(16进制数))	X(0058)	X(0058)	X(0058)	.(002E)	Y(0059)	Y(0059)
ASCII代码(16进制数)	30303538	30303538	30303538	30303245	30303539	30303539
二进制代码(16进制数)	5800	5800	5800	2E00	5900	5900

标签名(UTF-16(16进制数))	Y(0059)	.(002E)	Z(005A)	Z(005A)	Z(005A)
ASCII代码(16进制数)	30303539	30303245	30303541	30303541	30303541
二进制代码(16进制数)	5900	2E00	5A00	5A00	5A00

■结构体型的标签(构件排列的情况下)

应组合排列体型的标签与排列指定型的标签的指定方法进行指定。
全局标签名为“XXX.YYY[20,10,30]”的情况下，ASCII代码与二进制代码的指定示例如下所示。

标签名(UTF-16(16进制数))	X(0058)	X(0058)	X(0058)	.(002E)	Y(0059)	Y(0059)
ASCII代码(16进制数)	30303538	30303538	30303538	30303245	30303539	30303539
二进制代码(16进制数)	5800	5800	5800	2E00	5900	5900
标签名(UTF-16(16进制数))	Y(0059)	[(005B)	2(0032)	0(0030)	,(002C)	1(0031)
ASCII代码(16进制数)	30303539	30303542	30303332	30303330	30303243	30303331
二进制代码(16进制数)	5900	5B00	3200	3000	2C00	3100
标签名(UTF-16(16进制数))	0(0030)	(002C)	3(0033)	0(0030)	](005D)	
ASCII代码(16进制数)	30303330	30303243	30303333	30303330	30303544	
二进制代码(16进制数)	3000	2C00	3300	3000	5D00	

■结构体型的标签的数据类型

数据类型为下述的情况下，将变为结构体型的标签。

- 定时器
- 计数器
- 长计数器
- 累计定时器
- 长累计定时器
- 长定时器

此外，结构体具有触点、线圈、当前值的构件名及数据类型。

构件名	数据类型	说明
S	位	触点
C	位	线圈
N	定时器、计数器、累计定时器的情况下：字[无符号]/位列[16 位]	当前值
	长定时器、长计数器、长累计定时器的情况下：双字[无符号]/位列[32 位]	

■不可指定标签名

标签类型	内容	例
标签的位指定	不可以通过半角句号结合标签名与位指定作为字符串指定。	AAA.3
标签的位数指定	不可以将标签名与位数指定作为字符串指定。	K4AAA
排列指定型的标签	不可以将要素编号作为字符串指定。	BBB[XXX] BBB[XXX, YYY] BBB[XXX, YYY, ZZZ]
结构体型的标签	可以指定末端构件的结构体型的标签名。	XXX
	不可以通过构件中具有结构体型标签的的结构体型的标签指定以手动分配了软元件的标签。	Label1.Member1.Member2(仅以手动分配了软元件的情况下)

■标签名省略定义

结构体型标签的情况下，通过省略定义可以通过省略表记指定标签名。
省略定义时，指定省略点数中省略的标签名的点数，依次指定省略的标签的标签名长与标签名后登录。
但是，指定的标签名应通过以“.”隔开的单位指定。不可以指定标签名的字符单位。
例如，“LabelA.memberA3.memberB1”的结构体型标签的情况下，“LabelA”、“LabelA.memberA3”可以作为省略的标签名指定，但是不可以通过如“Label”、“LabelA.member”所示的标签名的字符单位指定。
登录的标签的字符串可以按照“%1”“%2”...“%n”(n: 省略点数中指定的数)方式通过“%”与偏置值(从上依次1、2、3...n)进行省略指定。
通过下述结构体型标签，按照“%1.memberA1”、“%1.memberA2”、“%1.%2.memberB1”、“%1.%2.memberB2”方式，以省略表记指定“LabelA”与“memberA3”时的登录步骤如下所示。

- LabelA.memberA1
- LabelA.memberA2
- LabelA.memberA3.memberB1
- LabelA.memberA3.memberB2

1. 指定省略点数中省略标记的标签名数

由于省略“LabelA”与“memberA3”的2种标签名，因此省略点数中指定“2”。

2. 指定标签名长中省略标记的标签名字符数

标签名	字符数	标签名长	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
LabelA	6字符	30303036	0600
memberA3	8字符	30303038	0800

3. 指定省略标记的标签名

指定省略的标签名。通过步骤1指定的省略点数重复步骤2与步骤3。

数据	指定的值	内容	
省略点数	2	指定省略定义的标签名的点数。	
标签名长	6	每省略定义的标签名1点的指定	指定通过省略点数指定的点数的标签
标签名	LabelA		
标签名长	8	每省略定义的标签名1点的指定	
标签名	memberA3		

限制事项

结构体型标签的构件中有排列指定型的标签的情况下，排列指定型的标签名不可以省略标记。

数据类型ID

数据类型ID被存储到响应数据中。

以ASCII代码进行数据通信时以2位的ASCII代码表示数据类型ID。

以ASCII代码进行数据通信时以1字节的二进制代码表示数据类型ID。

响应数据中存储的数据类型ID如下所示。

分类	数据类型名称	数据类型ID
基本数据型的标签	位	1
	字[无符号]/位列[16位]	2
	双字[双字]/位列[32位]	3
	字[带符号]	4
	双字[带符号]	5
	单精度实数	6
	双精度实数	7
	时间	8
	字符串	9
	字符串[Unicode]	10
	下述数据类型的触点/线圈 • 定时器 • 计数器 • 长计数器 • 累计定时器 • 长累计定时器 • 长定时器	1
	下述数据类型的当前值 • 定时器 • 计数器 • 累计定时器	2
	下述数据类型的当前值 • 长计数器 • 长累计定时器 • 长定时器	3
排列指定型的标签	排列指定型的标签将变为排列要素数据类型(基本数据类型)。	
结构体型的标签	结构体型的标签将变为末端要素数据类型(基本数据类型)。	

读取单位指定、写入单位指定

指定读取数据长或写入数据长的单位。

值	内容
0	标签数据类型为位的情况下指定。(位指定)
1	标签数据类型为位以外的情况下指定。(字节指定)

各数据类型中指定的读取单位指定、写入单位指定如下所示。

分类	数据类型名称	读取单位指定、写入单位指定
基本数据型的标签	位	0
	字[无符号]/位列[16位]	1
	双字[双字]/位列[32位]	1
	字[带符号]	1
	双字[带符号]	1
	单精度实数	1
	双精度实数	1
	时间	1
	字符串	1
	字符串[Unicode]	1
	下述数据类型的触点/线圈 • 定时器 • 计数器 • 长计数器 • 累计定时器 • 长累计定时器 • 长定时器	0
	下述数据类型的当前值 • 定时器 • 计数器 • 累计定时器	1
	下述数据类型的当前值 • 长计数器 • 长累计定时器 • 长定时器	1
排列指定型的标签	排列指定型的标签将变为排列要素数据类型(基本数据类型)的值。	
结构体型的标签	结构体型的标签将变为末端要素数据类型(基本数据类型)的值。	

■以ASCII代码进行数据通信时

将值转换为ASCII代码2位使用后，按照从高位字节到低位字节的顺序发送。

例

将值指定为0的情况下



■以二进制代码进行数据通信时

使用表示值的1字节的数值发送。

例

将值指定为0的情况下



固定值

指定0。

■以ASCII代码进行数据通信时

转换为ASCII代码2位使用后，按照从高位字节到低位字节的顺序发送。

例

0	0
30H	30H

■以二进制代码进行数据通信时

使用表示值的1字节的数值发送。

例

00H

读取数据长、写入数据长

以2字节为单位表示每标签1点的读取数据、写入数据的容量。

标签数据类型为位的情况下，应指定“2”。(☞ 75页 数据类型ID)

■以ASCII代码进行数据通信时

将容量转换为ASCII代码4位使用后，按照从高位字节到低位字节的顺序发送。

例

将容量指定为4的情况下

0	0	0	4
30H	30H	30H	34H

■以二进制代码进行数据通信时

使用表示容量的2字节的数值，按照从低位字节到高位字节的顺序发送。

例

将容量指定为4的情况下

04H, 00H

读取排列数据长、写入排列数据长

指定排列标签的读取或写入的数据容量。

通过读取单位指定、写入单位指定中指定的单位(位单位、字节单位)指定。(☞ 76页 读取单位指定、写入单位指定)

以位单位指定的情况下，以16位(2字节)单位指定。

数据的发送顺序与“读取数据长、写入数据长”相同。(☞ 77页 读取数据长、写入数据长)

Array Label Read(指令: 041A)

请求数据

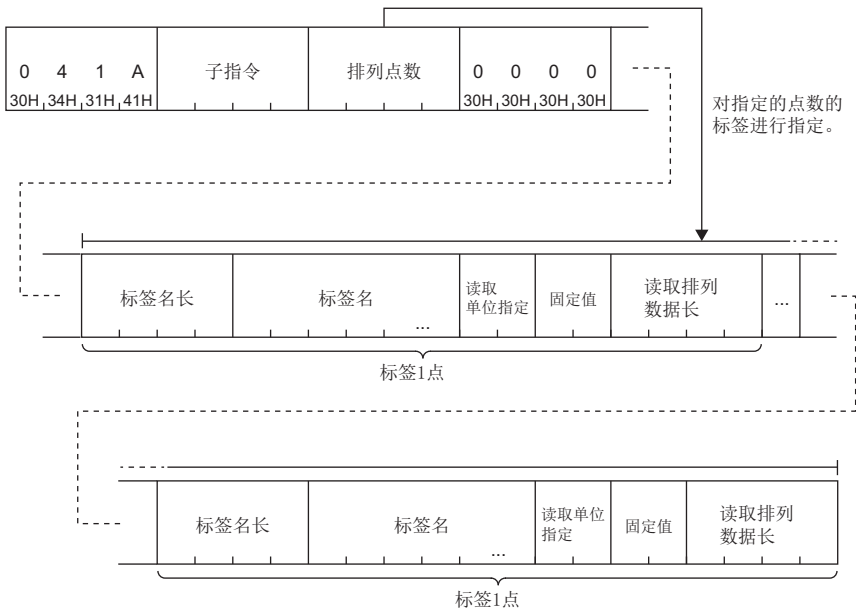
在排列指定型标签及结构体型标签的构件排列的情况下，通过标签读取数据。
即使不是排列指定型标签的情况下，也可作为排列要素数为1的标签读取。

限制事项

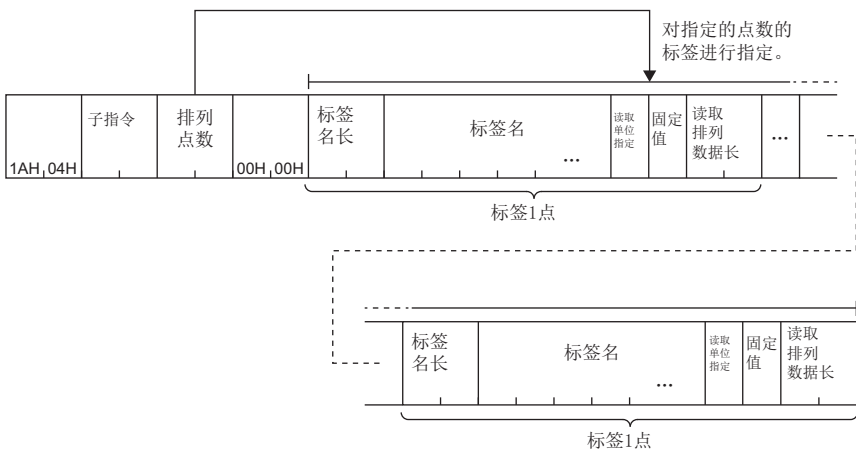
- 不可以指定下述数据类型的标签。
- 定时器
 - 计数器
 - 长计数器
 - 累计定时器
 - 长累计定时器
 - 长定时器
- 此外，也不可以指定数据类型为位的2次元排列及3次元排列。

■未省略定义的情况下

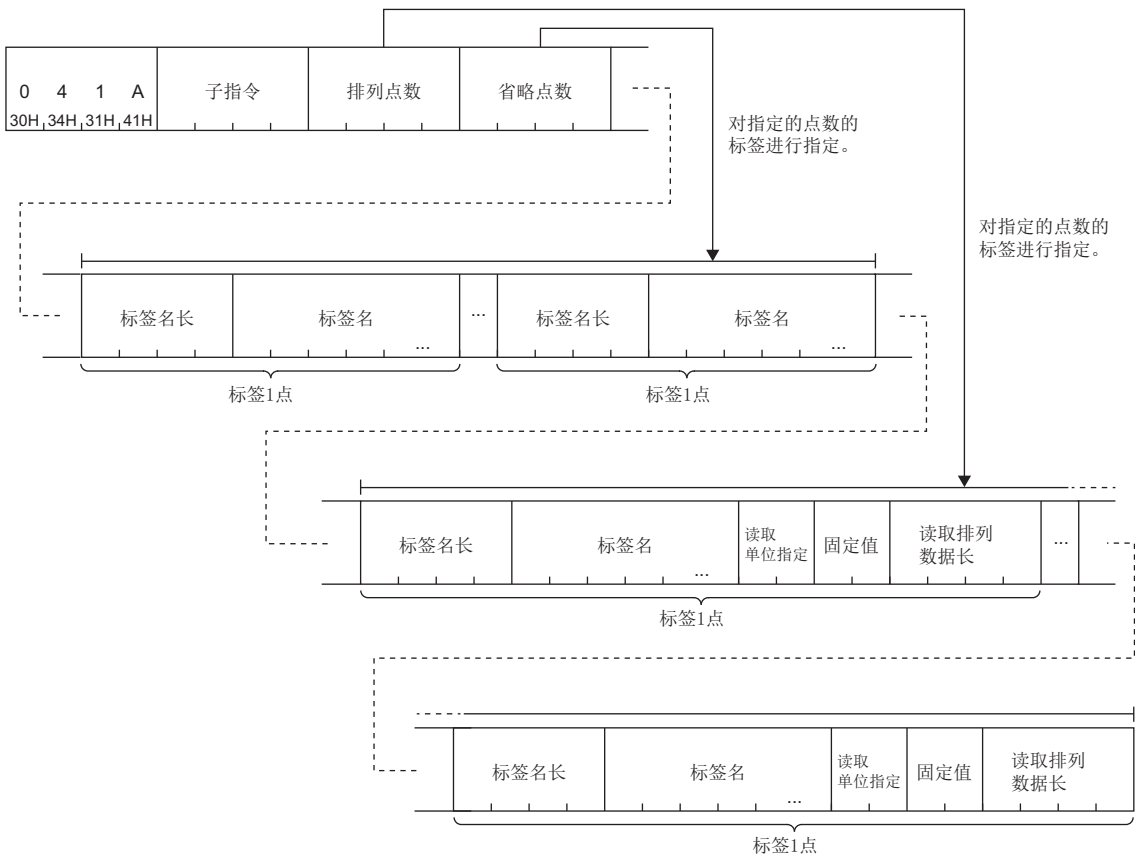
ASCII



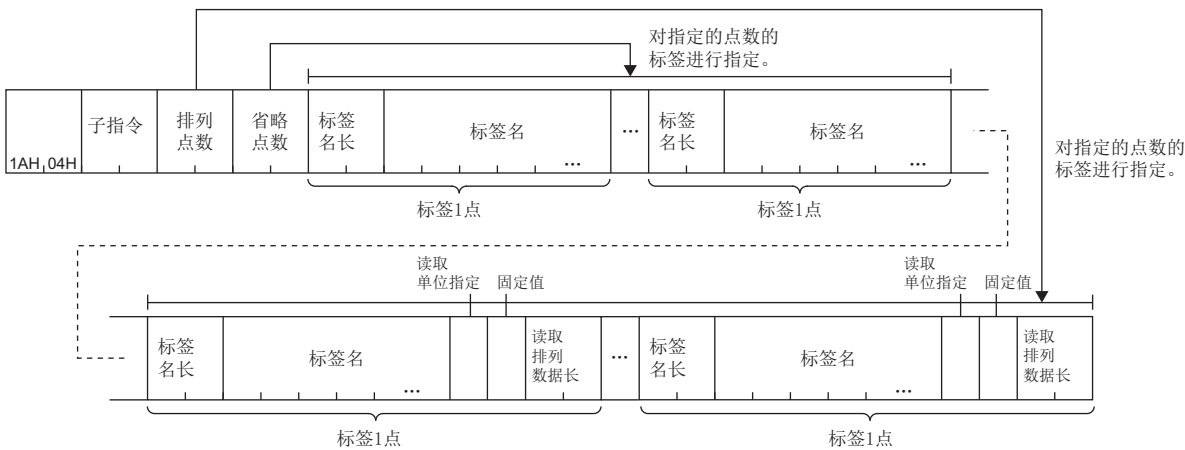
二进制



■省略定义的情况下
ASCII



二进制



■子指令

子指令	
ASCII代码	二进制代码
0 0 0 0 30H, 30H, 30H, 30H	00H, 00H

■排列点数

指定进行读取的排列的数。(70页 排列点数)

■省略点数

指定进行标签名省略定义的点数。(71页 省略点数)

■省略点数的标签名长、标签名

以省略点数指定省略定义的标签的标签名长与标签名。(71页 省略点数)

■排列点数的标签名长、标签名、读取单位指定、固定值、读取排列数据长

指定排列点数中指定数的标签。

- 72页 标签名长
- 73页 标签名
- 76页 读取单位指定、写入单位指定
- 77页 固定值
- 77页 读取排列数据长、写入排列数据长

响应数据

以16进制数存储已读取的标签值。根据ASCII代码及二进制代码，数据的排列有所不同。

ASCII

排列点数	数据类型ID	读取单位指定	读取排列数据长	读取数据	...	数据类型ID	读取单位指定	读取排列数据长	读取数据
------	--------	--------	---------	------	-----	--------	--------	---------	------

二进制



■排列点数

存储与请求数据相同的内容。

读取通过排列点数指定的点数。

因读取单位指定为位单位指定或字节单位指定，读取数据有所不同。

N与NULL终端的值如下所示。

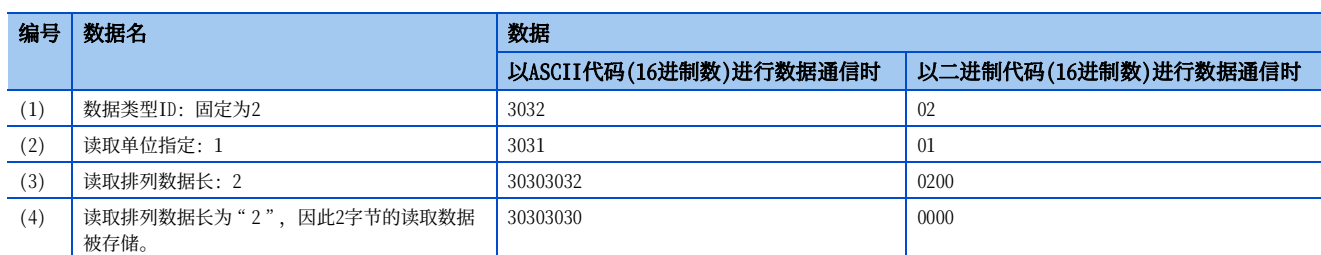
5

读取数据与数据类型无关, 以2字节(字)为单位被存储。

- 读取单位指定: 0
- 读取排列数据长: 6
- 读取数据: 0



- 读取单位指定: 1
- 读取排列数据长: 2
- 读取数据: 0

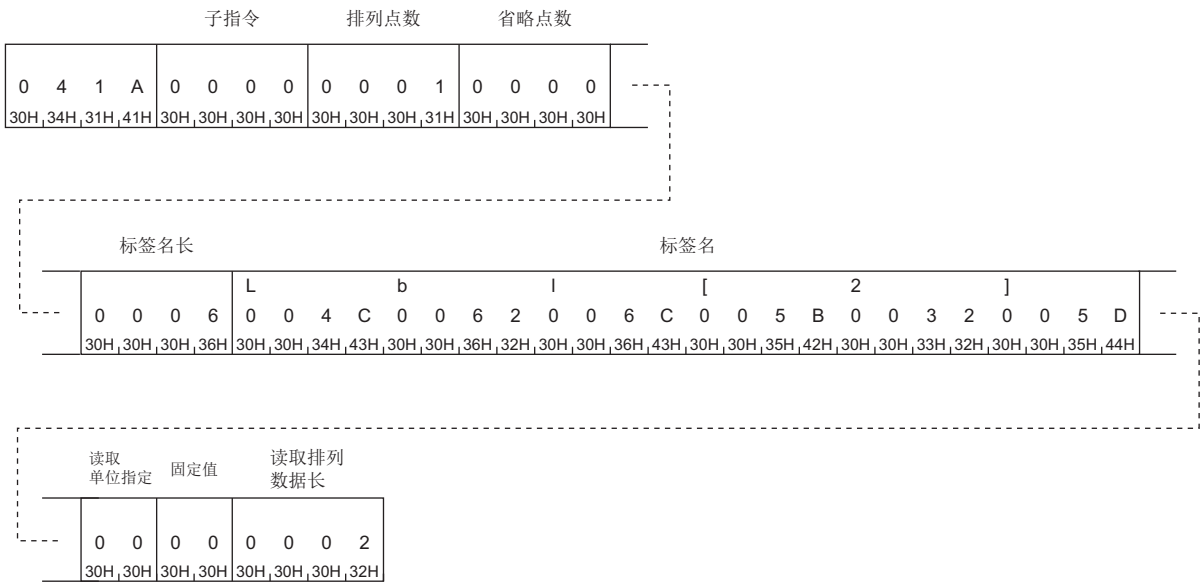


通信示例(排列指定型标签(位指定))

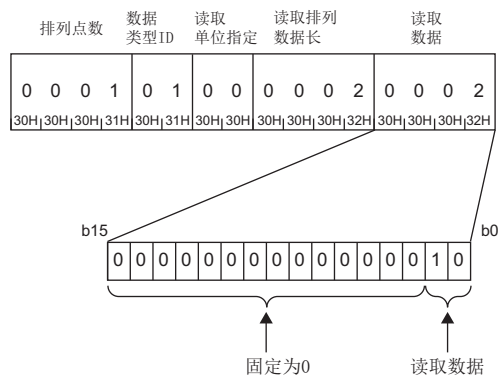
通过数据类型为位的排列指定型的标签“Lb1[2]”进行2位读取。
标签是存储下述值的标签。

- Lb1[2]:0(0FF)
- Lb1[3]:1(0N)

■以ASCII代码进行数据通信时
(请求数据)



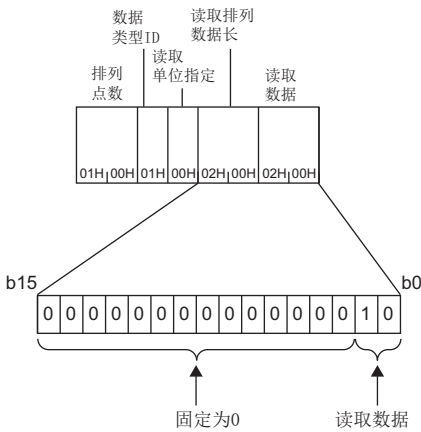
(响应数据)



■以二进制代码进行数据通信时
(请求数据)

										固定值			
子指令	排列 点数	省略 点数	标签 名长	标签名						读取 单位指定	读取排列 数据长		
					L	b	l	[	2	]			
1AH,04H	00H,00H	01H,00H	00H,00H	06H,00H	4CH,00H	62H,00H	6CH,00H	5BH,00H	32H,00H	5DH,00H	00H	00H	02H,00H

(响应数据)



通信示例(排列指定型标签(字节指定))

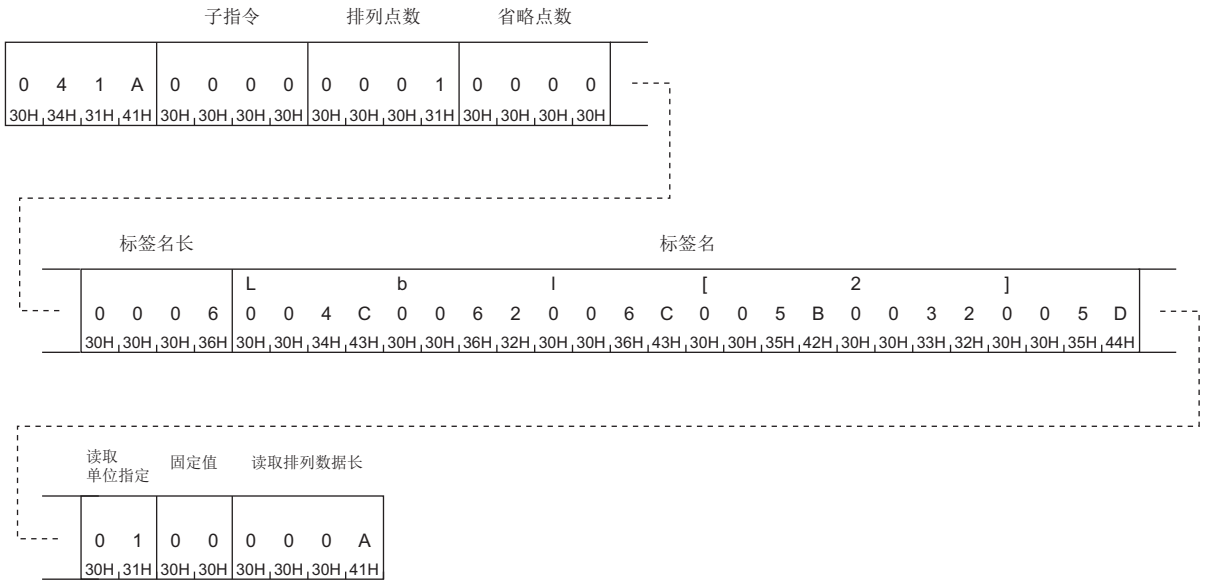
通过数据类型为字的排列指定型的标签“Lb1[2]”进行5字读取。

标签是存储下述值的标签。

- Lb1[2]:0044H
- Lb1[3]:0061H
- Lb1[4]:0074H
- Lb1[5]:0061H
- Lb1[6]:0031H

■以ASCII代码进行数据通信时

(请求数据)



(响应数据)

排列点数	数据类型ID	读取单位指定	读取排列数据长	读取数据
0 0 0 1 30H,30H,30H,31H	0 2 30H,32H	0 1 30H,31H	0 0 0 A 30H,30H,30H,41H	0 0 4 4 0 0 6 1 0 0 7 4 0 0 6 1 0 0 3 1 30H,30H,34H,34H,30H,30H,36H,31H,30H,30H,37H,34H,30H,30H,36H,31H,30H,30H,33H,31H

■以二进制代码进行数据通信时

(请求数据)

子指令	排列点数	省略点数	标签名长	标签名	固定值	读取单位指定	读取排列数据长
1AH,04H	00H,00H	01H,00H	00H,00H	06H,00H	L b l [2]	01H	00H
00H,00H	00H,00H	00H,00H	00H,00H	00H,00H	00H,00H	00H,00H	0AH,00H

(响应数据)

排列点数	数据类型ID	读取单位指定	读取排列数据长	读取数据
01H,00H	02H,01H	0AH,00H	44H,00H	61H,00H
74H,00H	61H,00H	31H,00H		

通信示例(结构体型标签)

通过数据类型为字的结构体型的标签“Typ1.led[2]”进行4字读取，通过数据类型为字的结构体型的标签“Typ1.No[1]”进行2字读取。

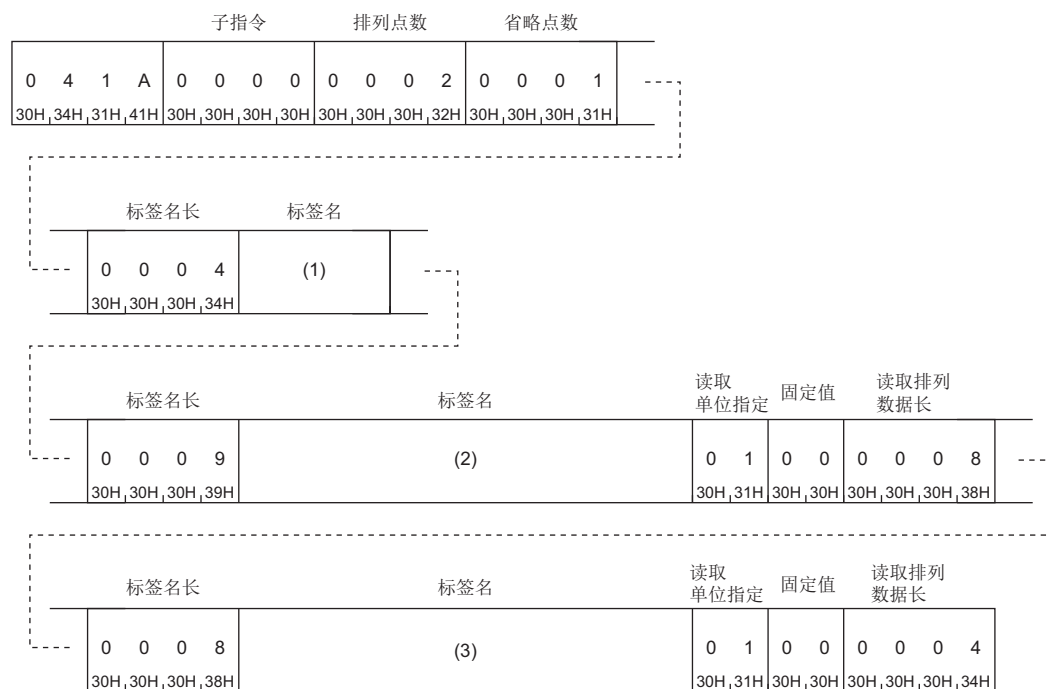
标签是存储下述值的标签。

- Typ1.led[2]:0031H
- Typ1.led[3]:0032H
- Typ1.led[4]:0033H
- Typ1.led[5]:0034H
- Typ1.No[1]:0030H
- Typ1.No[2]:0031H

进行将标签名的“Typ1”可省略标记为“%1”的省略定义。

■以ASCII代码进行数据通信时

(请求数据)

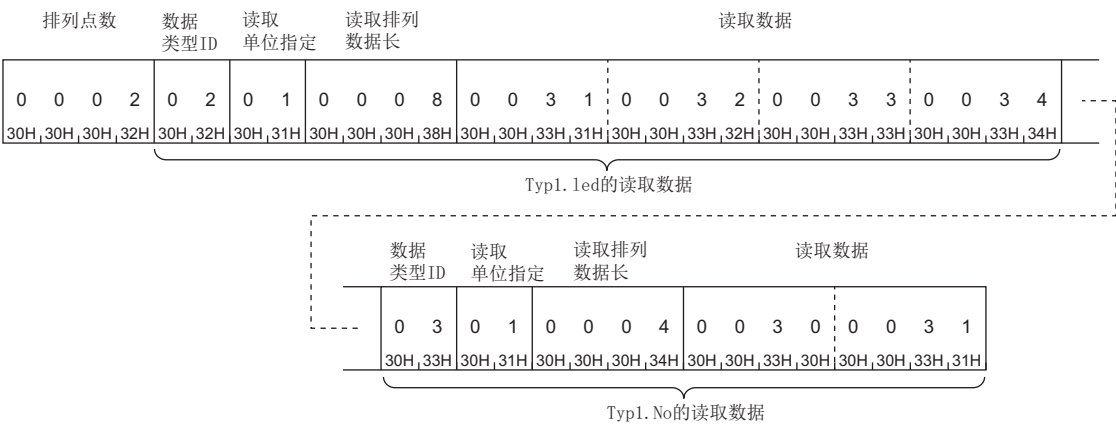


编号	项目	值
—	标签名	Typ1
	UTF-16(16进制数)	0054007900700031
(1)	ASCII代码(16进制数)	30303534303037393030373030303331

编号	项目	值
—	标签名	%1.led[2]
	UTF-16(16进制数)	00250031002E006C00650064005B0032005D
(2)	ASCII代码(16进制数)	303032353030333130303245303036433030363530303634303035423030333230303544

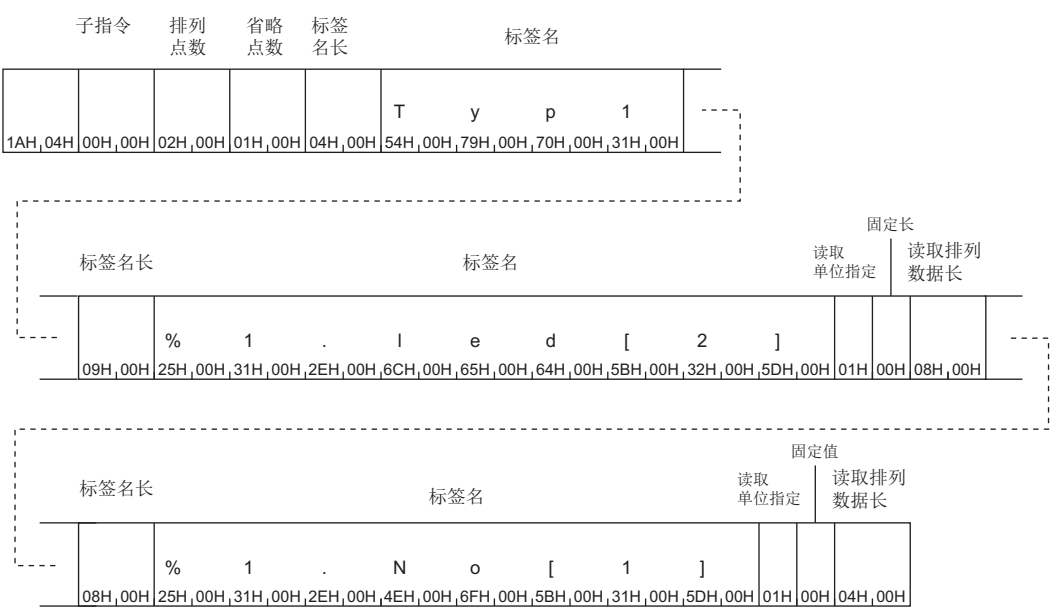
编号	项目	值
—	标签名	%1.No[1]
	UTF-16(16进制数)	00250031002E004E006F005B0031005D
(3)	ASCII代码(16进制数)	3030323530303331303032453030344530303646303035423030333130303544

(响应数据)

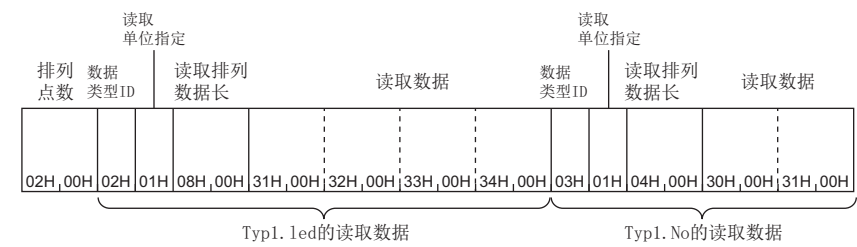


■以二进制代码进行数据通信时

(请求数据)



(响应数据)



Array Label Write(指令: 141A)

在排列指定型标签及结构体型标签的构件排列的情况下，将数据写入至标签中。
即使不是排列指定型标签的情况下，也可作为排列要素数为1的标签写入。

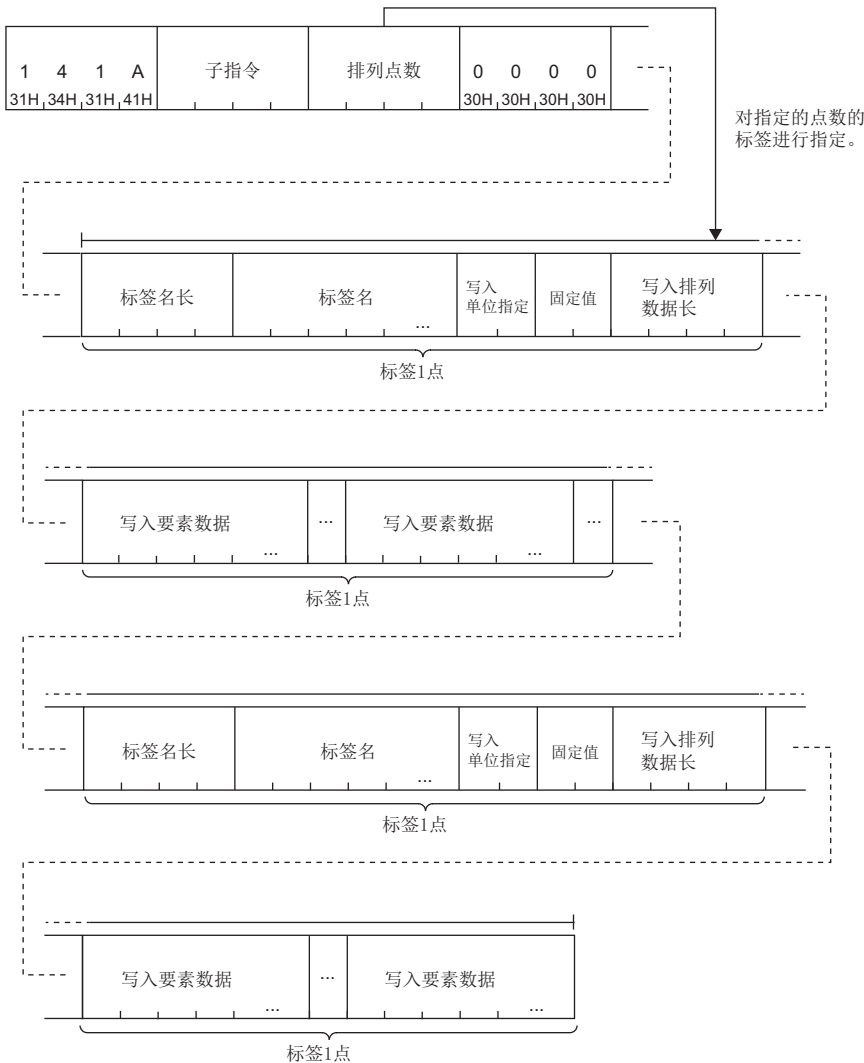
限制事项

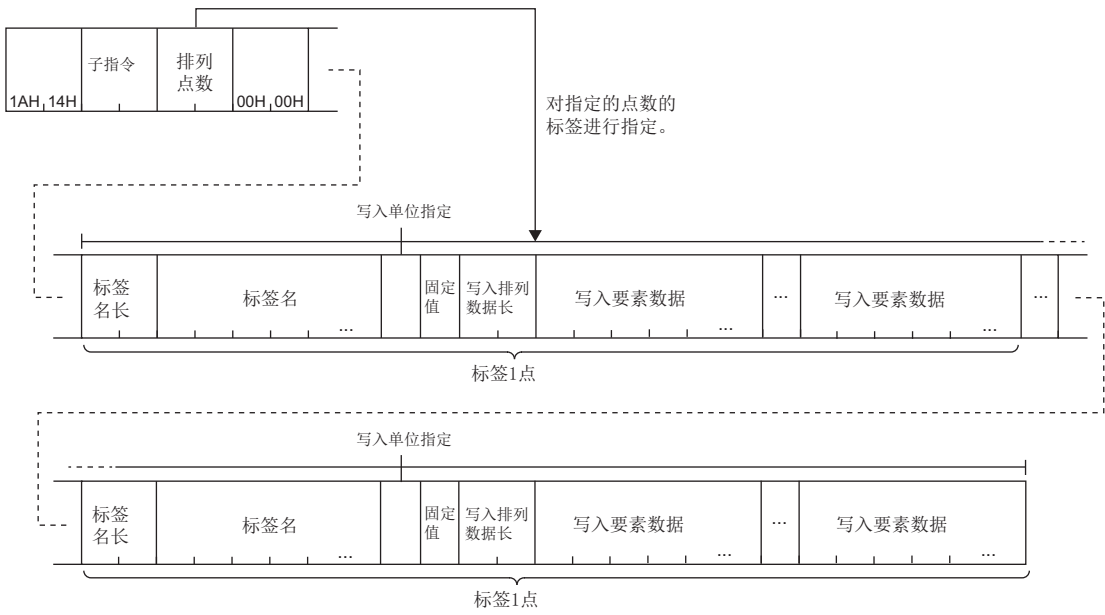
- 不可以指定下述数据类型的标签。
- 定时器
 - 计数器
 - 长计数器
 - 累计定时器
 - 长累计定时器
 - 长定时器
- 此外，也不可以指定数据类型为位的2次元排列及3次元排列。

请求数据

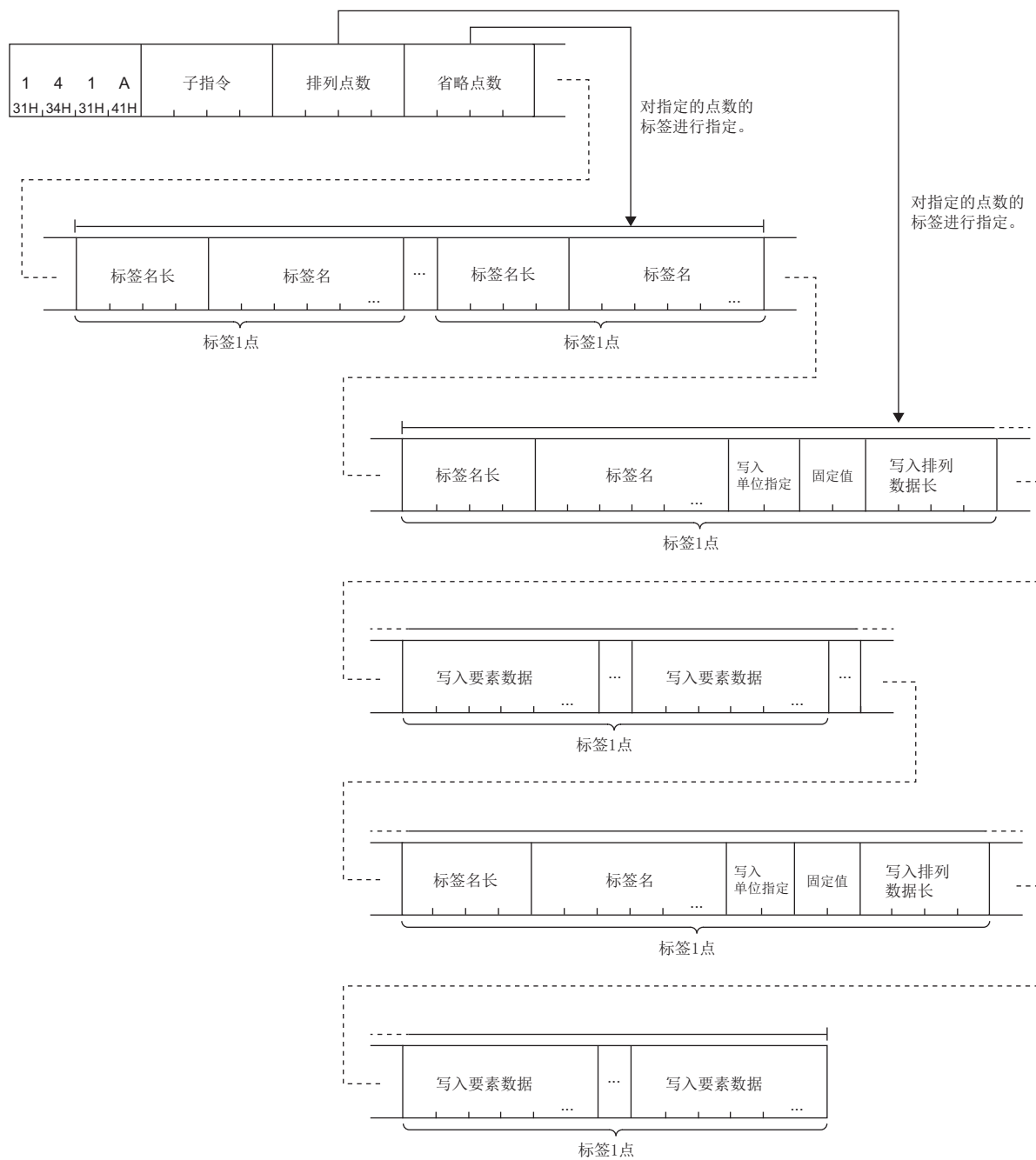
■未省略定义的情况下

ASCII





ASCII



子指令	
ASCII代码	二进制代码
0 0 0 0 30_H 30_H 30_H 30_H	00_H 00_H

写入数据由下述构成。

5 指令

5.3 Label(标签访问)

因写入单位指定为位指定或字节指定，写入要素数据有所不同。

写入单位指定为位指定的情况下，写入要素数据应以2字节为单位进位的大数指定。

写入单位指定与标签数据类型不匹配的情况下将变为通信出错，响应报文的结束代码中出错代码被存储。关于出错代码有关内容，请参阅CPU模块手册。（[MELSEC iQ-R CPU模块用户手册\(应用篇\)](#)）

数据类型为字符串或字符串(Unicode)排列的情况下，写入要素数据将在每排列一点中以包括NULL终端的形式指定，且以全要素、标签定义字符数+N的大数指定。

N与NULL终端的值如下所示。

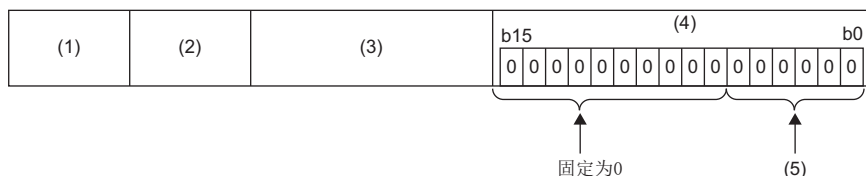
数据类型	N的值	NULL终端的值
字符串	- 定义字符数为奇数的情况下：1 - 定义字符数为偶数的情况下：2	00H
字符串(Unicode)	2	0000H

要点

写入要素数据应与数据类型无关，以2字节(字)为单位被存储。

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 写入单位指定：0
- 写入排列数据长：6
- 写入数据：0



编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	写入单位指定：0	3030	00
(2)	固定值	3030	00
(3)	写入排列数据长：6	30303036	0600
(4)	写入数据以16位(2字节)为单位存储。	30303030	0000
(5)	写入排列数据长为“6”，因此存储6位的写入要素数据。	—	

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 写入单位指定：1
- 写入排列数据长：2
- 写入数据：0



编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	写入单位指定：1	3031	01
(2)	固定值	3030	00
(3)	写入排列数据长：2	30303032	02
(4)	写入排列数据长为“2”，因此存储2字节的写入要素数据。	30303030	0000

响应数据

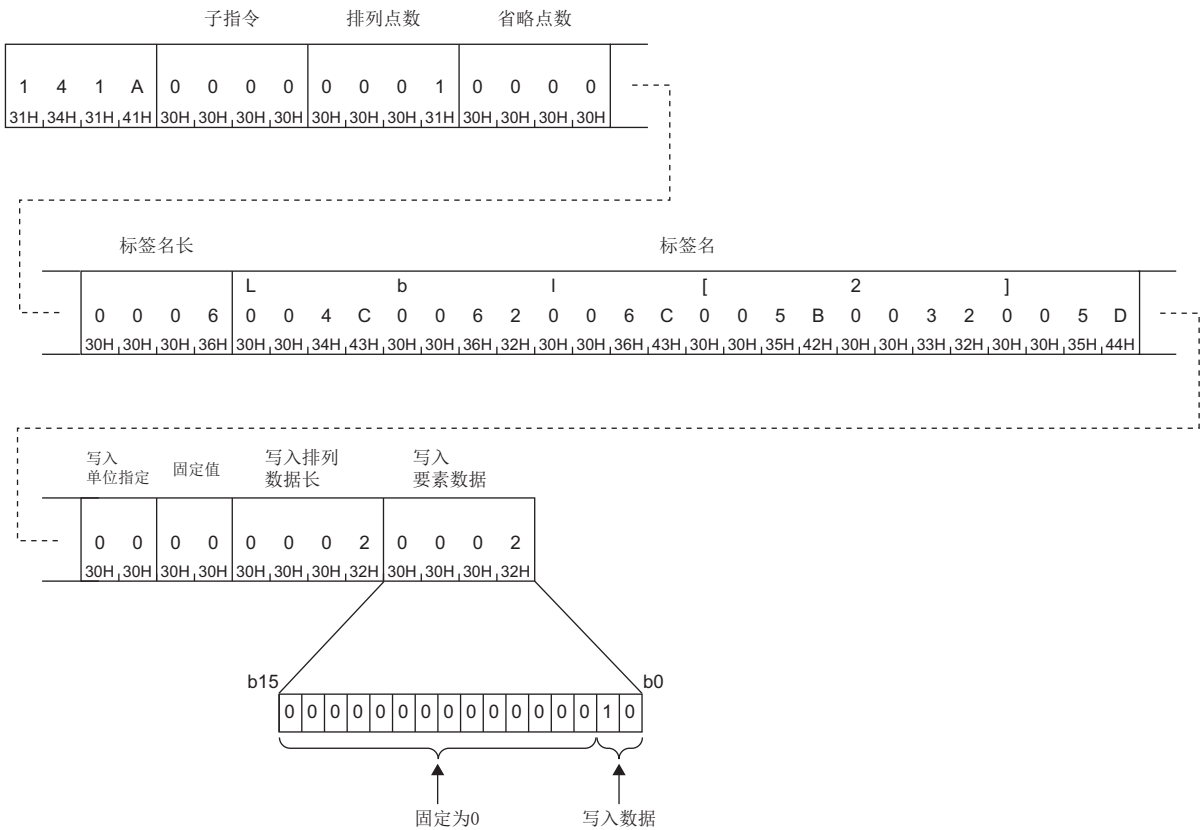
无Array Label Write指令的响应数据。

通信示例(排列指定型标签(位指定))

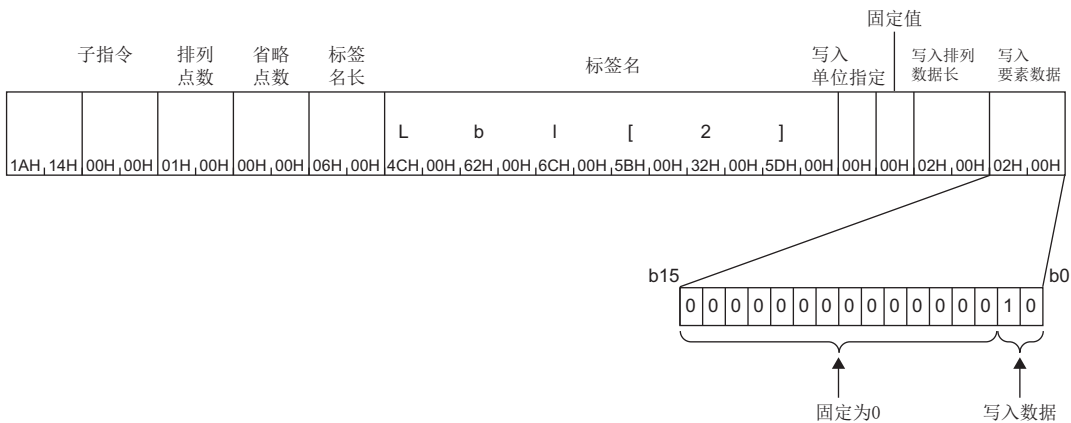
通过数据类型为位的排列指定型的标签“Lb1[2]”进行2位写入。
标签是写入下述值的标签。

- Lb1[2]:0(OFF)
- Lb1[3]:1(ON)

■以ASCII代码进行数据通信时
(请求数据)



■以二进制代码进行数据通信时
(请求数据)



通信示例(排列指定型标签(字节指定))

通过数据类型为字的排列指定型的标签“Lb1[2]”进行5字写入。

标签是写入下述值的标签。

- Lb1[2]:4400H
- Lb1[3]:6100H
- Lb1[4]:7400H
- Lb1[5]:6100H
- Lb1[6]:3100H

■以ASCII代码进行数据通信时 (请求数据)

子指令				排列点数				省略点数			
1	4	1	A	0	0	0	0	0	0	0	1
31H,34H,31H,41H	30H,30H,30H,30H	30H,30H,30H,31H	30H,30H,30H,30H	30H,30H,30H,31H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H
标签名长				标签名							
0	0	0	6	L	b	1	[	2	]	0	0
30H,30H,30H,36H	30H,30H,34H,43H	30H,30H,36H,32H	30H,30H,36H,43H	30H,30H,35H,42H	30H,30H,33H,32H	30H,30H,35H,44H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,30H
写入单位指定		固定值		写入排列数据长		写入数据					
0	1	0	0	0	0	0	A	0	0	4	4
30H,31H	30H,30H	30H,30H,30H,41H	30H,30H,34H,34H	30H,30H,36H,31H	30H,30H,37H,34H	30H,30H,36H,31H	30H,30H,33H,31H	30H,30H,33H,31H	30H,30H,33H,31H	30H,30H,33H,31H	30H,30H,33H,31H

■以二进制代码进行数据通信时 (请求数据)

子指令	排列 点数	省略 点数	标签名长	标签名
				L b l [2] - - -
1AH,14H	00H,00H	01H,00H	00H,00H	06H,00H
4CH,00H	62H,00H	6CH,00H	5BH,00H	32H,00H
5DH,00H				

固定值							
写入 单位指定	写入排列 数据长	写入数据					
01H	00H	0AH,00H	44H,00H	61H,00H	74H,00H	61H,00H	31H,00H

通信示例(结构体型标签)

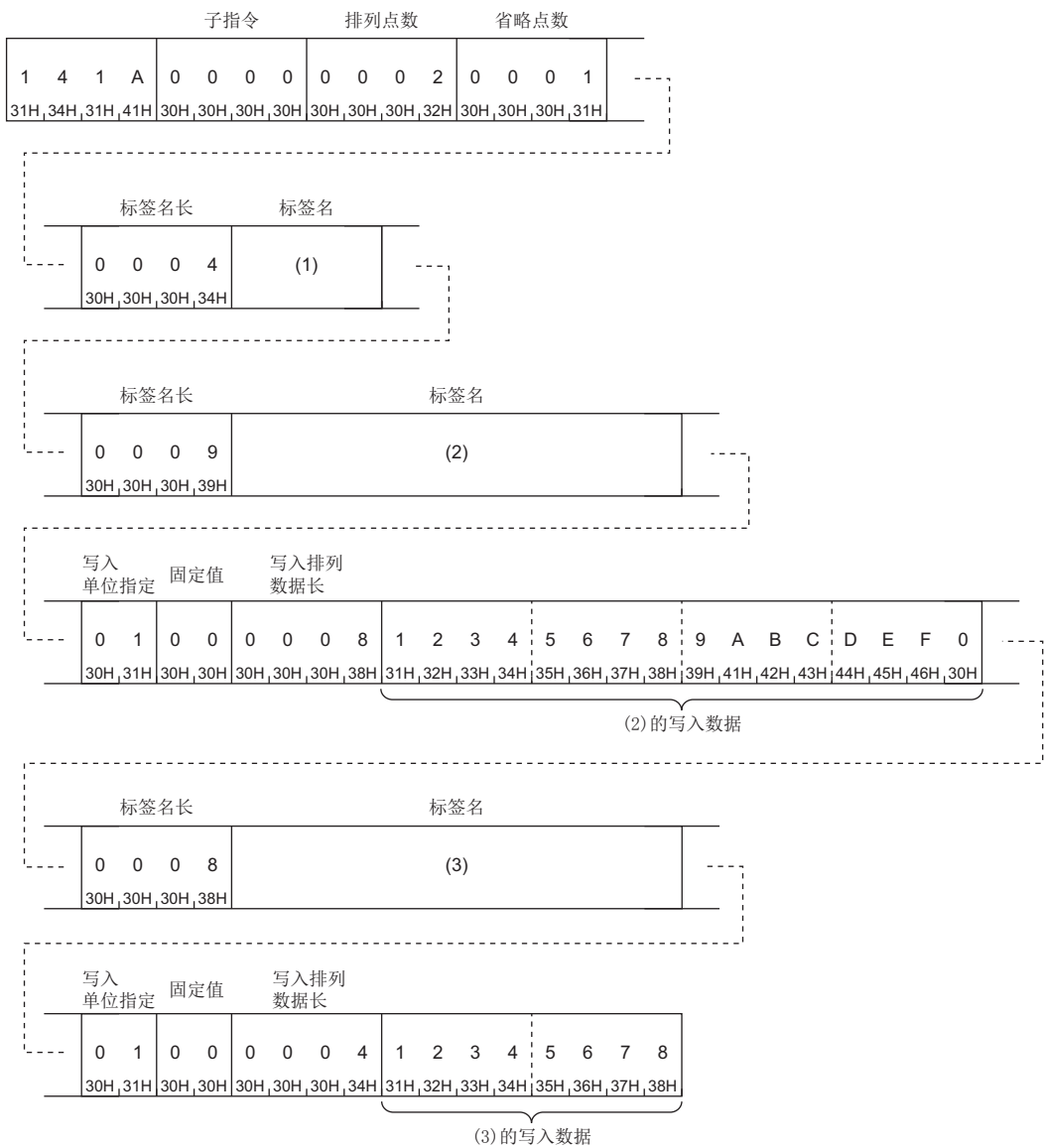
通过数据类型为字的结构体型的标签“Typ1.led[5]”进行4字写入，通过数据类型为字的结构体型的标签“Typ1.No[7]”进行2字写入。

标签是写入下述值的标签。

- Typ1.led[5]:1234H
- Typ1.led[6]:5678H
- Typ1.led[7]:9ABCH
- Typ1.led[8]:DEF0H
- Typ1.No[7]:1234H
- Typ1.No[8]:5678H

进行将标签名的“Typ1”可省略标记为“%1”的省略定义。

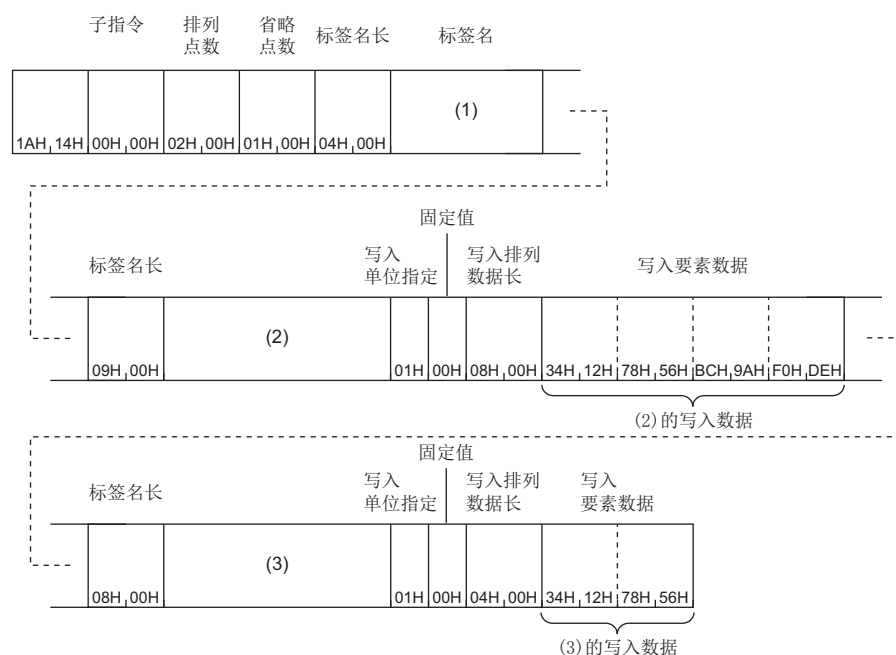
■以ASCII代码进行数据通信时
(请求数据)



编号	项目	值
—	标签名	Typ1
	UTF-16(16进制数)	0054007900700031
(1)	ASCII代码(16进制数)	30303534303037393030373030303331
编号	项目	值
—	标签名	%1.led[5]
	UTF-16(16进制数)	00250031002E006C00650064005B0035005D
(2)	ASCII代码(16进制数)	303032353030333130303245303036433030363530303634303035423030333530303544
编号	项目	值
—	标签名	%1.No[7]
	UTF-16(16进制数)	00250031002E004E006F005B0037005D
(3)	ASCII代码(16进制数)	3030323530303331303032453030344530303646303035423030333730303544

■以二进制代码进行数据通信时

(请求数据)



编号	项目	值
—	标签名	Typ1
	UTF-16(16进制数)	0054007900700031
(1)	二进制代码(16进制数)	5400790070003100
编号	项目	值
—	标签名	%1.led[5]
	UTF-16(16进制数)	00250031002E006C00650064005B0035005D
(2)	二进制代码(16进制数)	250031002E006C00650064005B0035005D00
编号	项目	值
—	标签名	%1.No[7]
	UTF-16(16进制数)	00250031002E004E006F005B0037005D
(3)	二进制代码(16进制数)	250031002E004E006F005B0037005D00

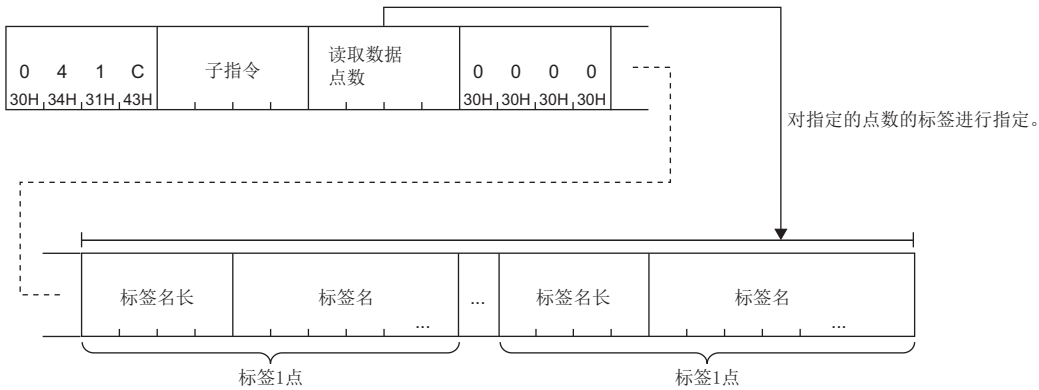
Label Read Random(指令: 041C)

指定标签，读取数据。
排列的情况下，也可指定各要素数据进行读取。
Label Read Random指令中的响应数据将变为标签1点单位的读取。连续读取排列数据的情况下，应使用Array Label Read指令。(☞ 78页 Array Label Read(指令: 041A))

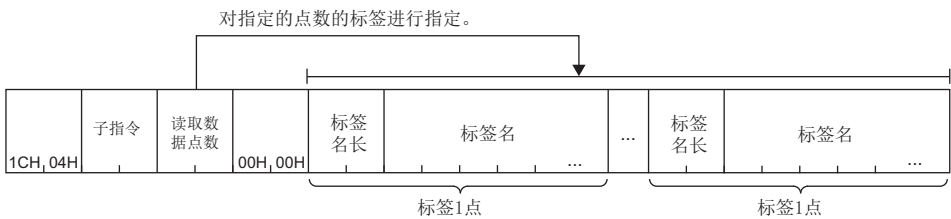
请求数据

■未省略定义的情况下

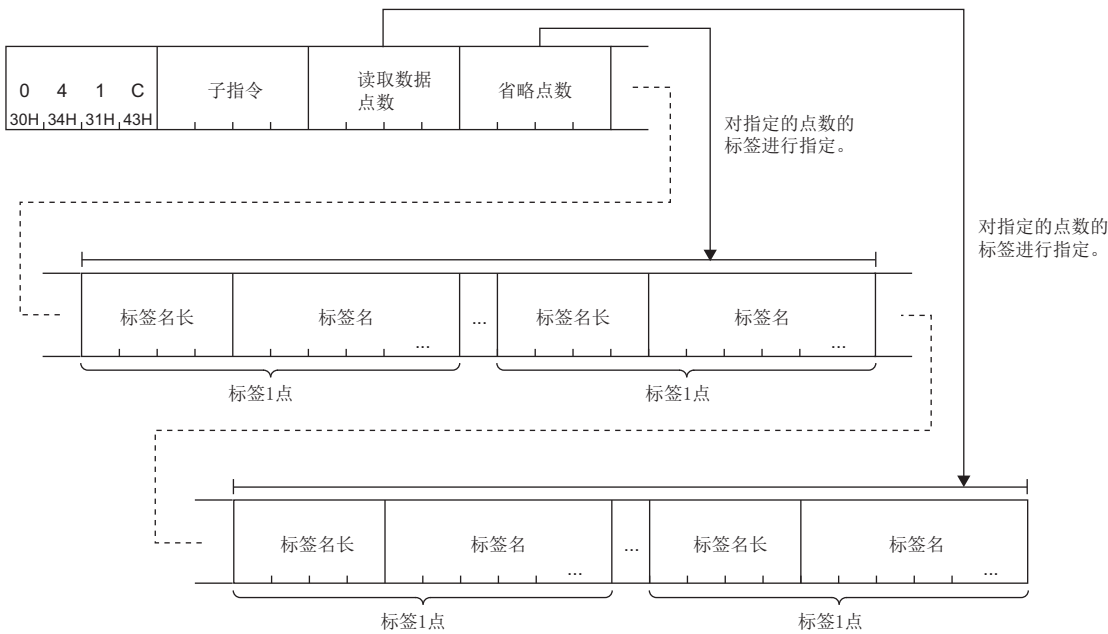
ASCII



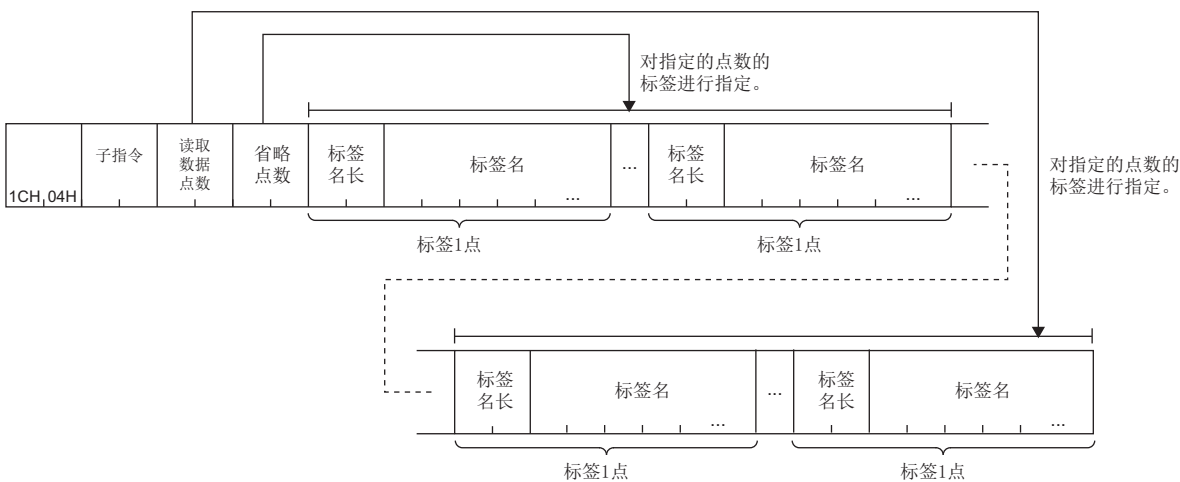
二进制



■省略定义的情况下
ASCII



二进制



■子指令

子指令	
ASCII代码	二进制代码
0 0 0 0 30H 30H 30H 30H	00H 00H

■读取数据点数

指定进行读取的标签数。(71页 读取数据点数、写入数据点数)

■省略点数

指定进行标签名省略定义的点数。(71页 省略点数)

■省略点数的标签名长、标签名

以省略点数指定省略定义的标签的标签名长与标签名。(71页 省略点数)

■读取数据点数的标签名长、标签名

指定读取数据点数中指定数的标签。

☞ 72页 标签名长

☞ 73页 标签名

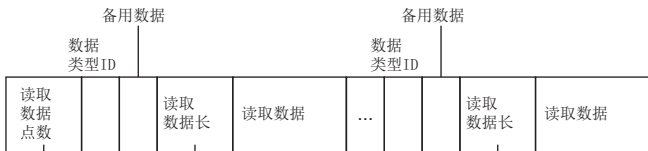
响应数据

以16进制数存储已读取的标签值。根据ASCII代码及二进制代码，数据的排列有所不同。

ASCII

读取数据点数	数据类型ID	备用数据	读取数据长	读取数据	...	数据类型ID	备用数据	读取数据长	读取数据
--------	--------	------	-------	------	-----	--------	------	-------	------

二进制



■读取数据点数

存储与请求数据相同的内容。

■数据类型ID、读取数据长、备用数据、读取数据

读取通过读取数据点数指定的点数。

编号	数据名	数据构成																																				
(1)	数据类型ID	<table><tr><td rowspan="2">(1)</td><td rowspan="2">(2)</td><td rowspan="2">(3)</td><td colspan="16">(4)</td></tr><tr><td>b15</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>b0</td></tr></table>	(1)	(2)	(3)	(4)																b15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b0
(1)	(2)					(3)	(4)																															
			b15	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	b0																		
(2)	备用数据																																					
(3)	读取数据长																																					
(4)	读取数据																																					

根据读取的标签的数据类型ID读取数据有所不同。(☞ 75页 数据类型ID)

数据类型为字符串或字符串(Unicode)的情况下，读取数据将变为标签的定义字符数+N。有效的字符串将变为NULL终端为止，以后将变为不定值。

N与NULL终端的值如下所示。

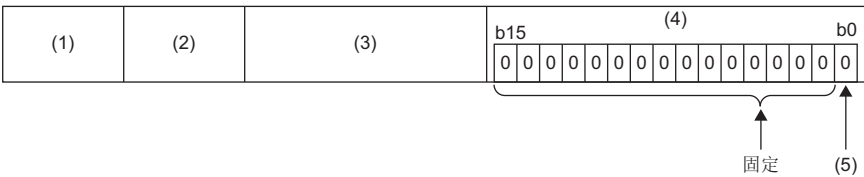
数据类型	N的值	NULL终端的值
字符串	• 定义字符数为奇数的情况下：1 • 定义字符数为偶数的情况下：2	00H
字符串(Unicode)	2	0000H

要点

- 读取数据与数据类型无关，应以2字节(字)为单位指定。
- 由于备用数据存储不定的值，因此请勿使用。

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 数据类型ID：1
- 读取数据长：2
- 读取数据：0



编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	数据类型ID：固定为1	3031	01
(2)	备用数据	—	—

编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(3)	读取数据长: 固定为2	30303032	0200
(4)	读取数据以16位(2字节)为单位存储。	30303030	0000
(5)	数据类型为“1”, 因此1位的读取数据被存储。	—	

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 数据类型ID: 2
- 读取数据长: 2
- 读取数据: 2

(1)	(2)	(3)	(4)
-----	-----	-----	-----

编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	数据类型ID:2	3032	02
(2)	备用数据	—	—
(3)	读取数据长: 2	30303032	0200
(4)	读取数据以读取数据长中指定的数据容量被存储。	30303130	1000

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 数据类型ID: 10
- 读取数据长: 8
- 读取数据: AAAA

(1)	(2)	(3)	(4)
-----	-----	-----	-----

编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	数据类型ID: 10	3130	10
(2)	备用数据	—	—
(3)	读取数据长: 8	30303038	0800
(4)	读取数据: AAAA	30303431303034313030343130303431	4100410041004100

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 数据类型ID: 8
- 读取数据长: 4
- 读取数据: 24日20时31分23秒647毫秒

(1)	(2)	(3)	(4)
-----	-----	-----	-----

编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	数据类型ID: 8	3038	08
(2)	备用数据	—	—
(3)	读取数据长: 4	30303034	0400
(4)	读取数据*1: 24日20时31分23秒647毫秒	3746464646464646	7FFFFFFF

*1 在80000000H(-24日20时31分23秒648毫秒)~7FFFFFFFH(24日20时31分23秒647毫秒)范围内以1毫秒为单位通过16进制数被存储。

通信示例

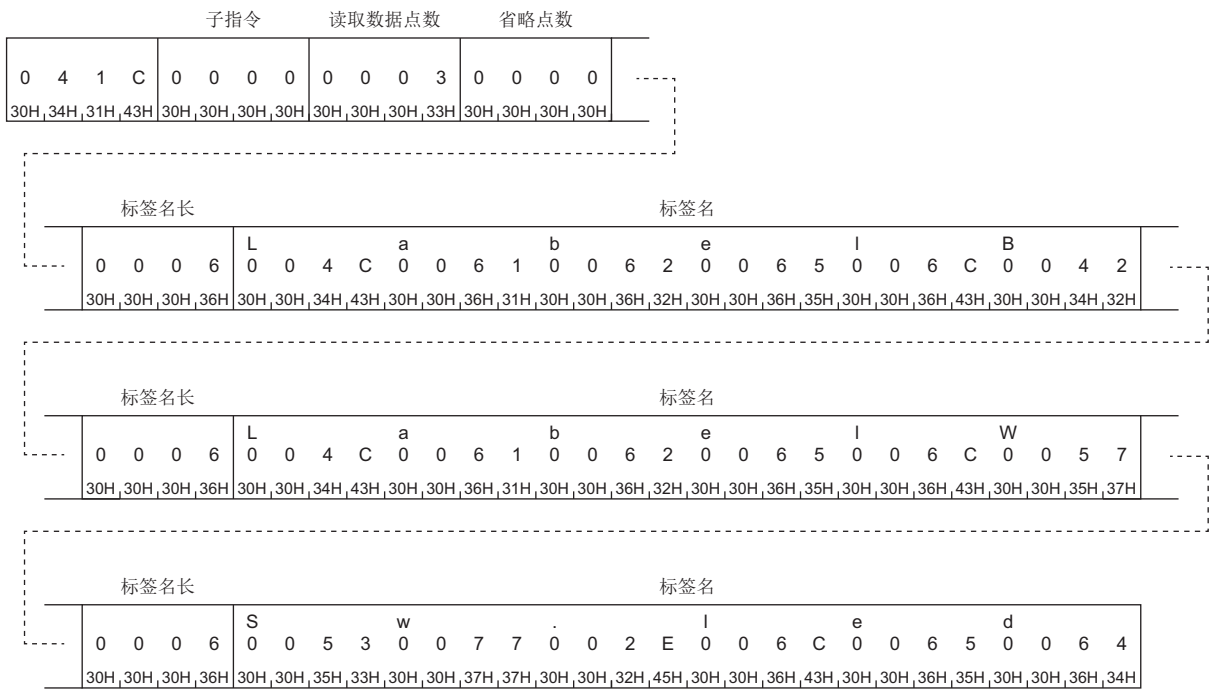
通过下述3种标签读取数据。

- 数据类型为位的基本数据类型标签 “LabelB”
- 数据类型为字的基本数据类型标签 “LabelW”
- 数据类型为字的结构体类型标签 “Sw.lcd”

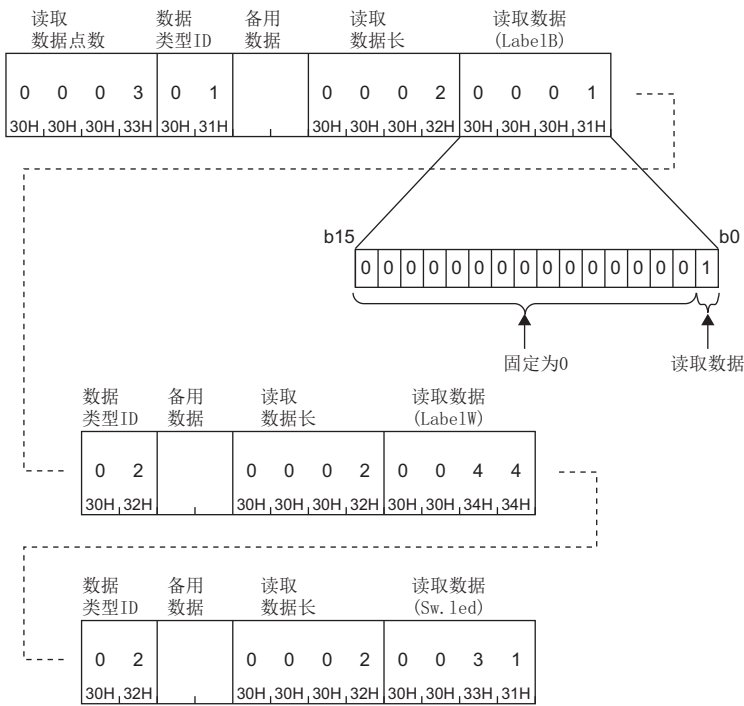
标签是存储下述值的标签。

- LabelB:1(ON)
- LabelW:0044H
- Sw.lcd:0031H

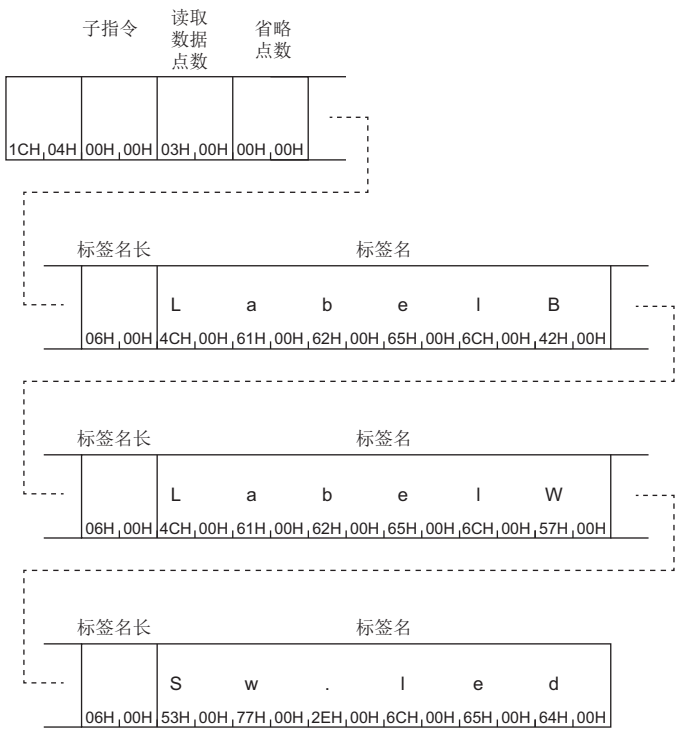
■以ASCII代码进行数据通信时
(请求数据)



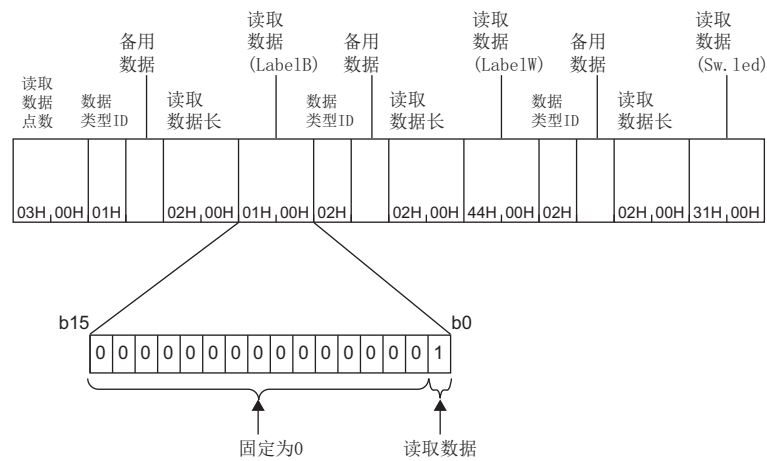
(响应数据)



■以二进制代码进行数据通信时
(请求数据)



(响应数据)



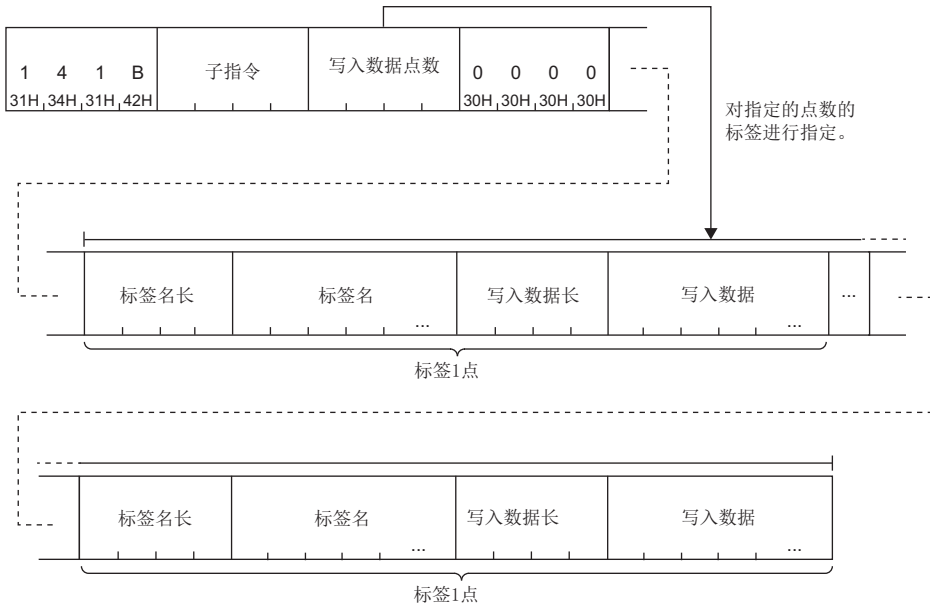
Label Write Random(指令: 141B)

Label Write Random指令中将变为标签1点单位的写入。
指定标签，写入数据。
排列的情况下，也可指定各要素数据进行写入。
Label Write Random指令中将变为标签1点单位的写入。连续写入排列数据的情况下，应使用ArrayLabel Write指令。(87 页 Array Label Write(指令: 141A))

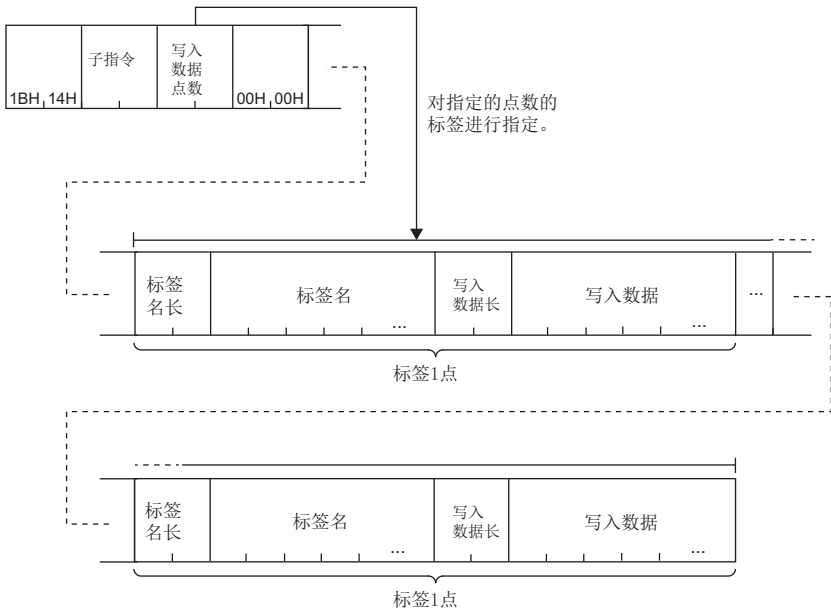
请求数据

■未省略定义的情况下

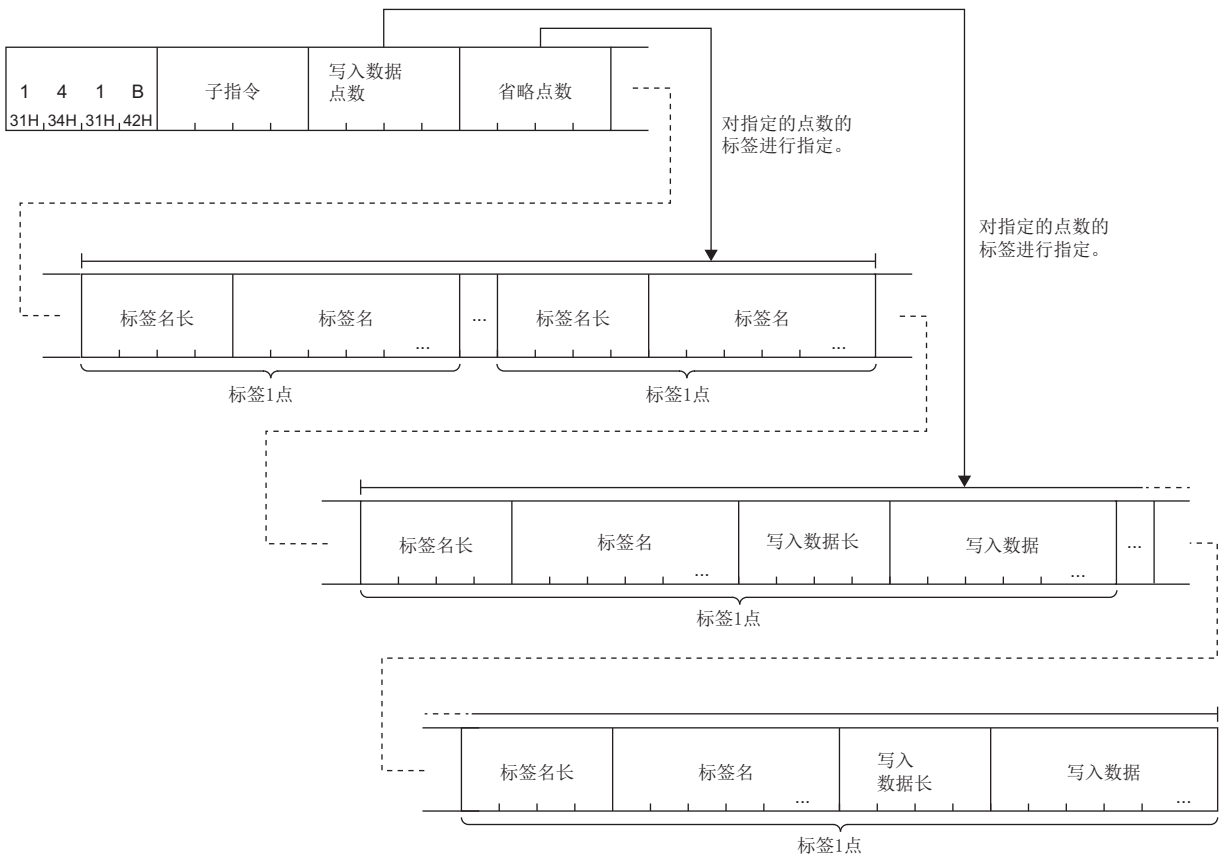
ASCII



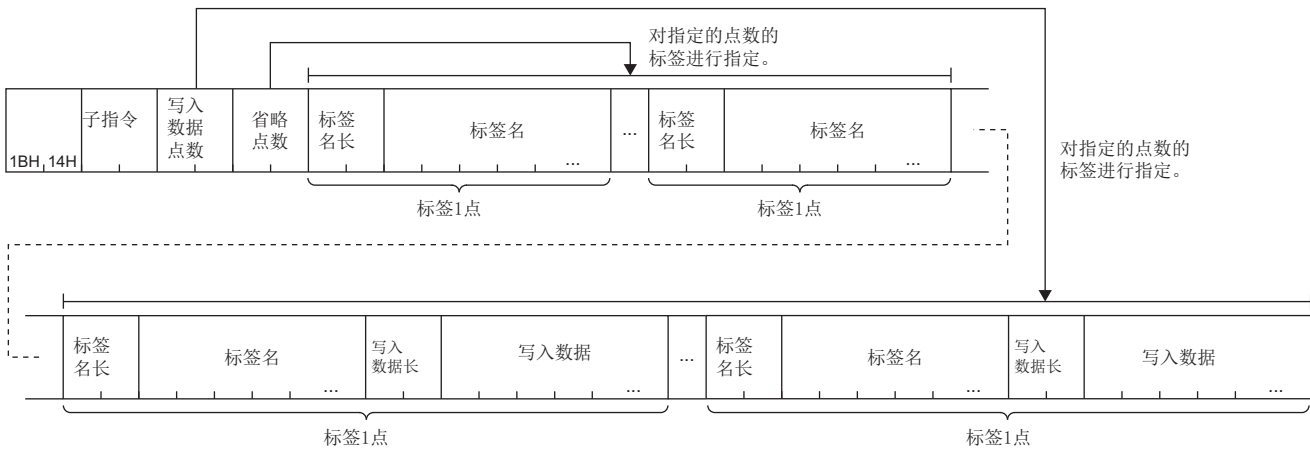
二进制



■省略定义的情况下
ASCII



二进制



■子指令

子指令	
ASCII代码	二进制代码
0 0 0 0 30H, 30H, 30H, 30H	00H, 00H

■写入数据点数

指定进行写入的标签数。(71页 读取数据点数、写入数据点数)

■省略点数

指定进行标签名省略定义的点数。(71页 省略点数)

■省略点数的标签名长、标签名

以省略点数指定省略定义的标签的标签名长与标签名。(71页 省略点数)

■写入数据点数标签名长、标签名、写入数据长、写入数据

指定写入数据点数中指定的点数。

72页 标签名长

73页 标签名

77页 读取数据长、写入数据长

写入数据由下述构成。

编号	数据名	数据构成			
(1)	写入数据长	<table><tr><td>(1)</td><td>(2)</td></tr></table>		(1)	(2)
(1)	(2)				
(2)	写入数据				

Label Write Random指令的写入数据长需要符合标签的数据类型。

各数据类型中指定的写入数据长如下所示。

分类	数据类型名称	写入数据长
基本数据型的标签	位	2
	字[无符号]/位列[16位]	2
	双字[双字]/位列[32位]	4
	字[带符号]	2
	双字[带符号]	4
	单精度实数	4
	双精度实数	8
	时间	4
	字符串	标签定义字符数+N*1
	字符串[Unicode]	标签定义字符数×2+2
	下述数据类型的触点/线圈 • 定时器 • 计数器 • 长计数器 • 累计定时器 • 长累计定时器 • 长定时器	2
	下述数据类型的当前值 • 定时器 • 计数器 • 累计定时器	2
	下述数据类型的当前值 • 长计数器 • 长累计定时器 • 长定时器	4
排列指定型的标签	排列指定型的标签将变为排列要素数据类型(基本数据类型)的值。	
结构体型的标签	结构体型的标签将变为末端要素数据类型(基本数据类型)的值。	

*1 N的值在标签定义字符数为奇数的情况下将变为1，在偶数的情况下将变为2。

写入数据长与标签数据类型不匹配的情况下将变为通信出错，响应报文的结束代码中出错代码被存储。关于出错代码有关内容，请参阅CPU模块手册。(MELSEC iQ-R CPU模块用户手册(应用篇))

数据类型为字符串或字符串(Unicode)的情况下，写入数据长将通过标签的定义字符数+N进行指定。此外写入数据应以包括NULL 终端的形式指定。

N与NULL终端的值如下所示。

数据类型	N的值	NULL终端的值
字符串	- 定义字符数为奇数的情况下: 1 - 定义字符数为偶数的情况下: 2	00H
字符串(Unicode)	2	0000H

要点 

- 写入数据与数据类型无关，应以2字节(字)为单位指定。

下述条件中以ASCII代码与二进制代码进行数据通信时的示例如下所示。

- 写入数据长: 2
- 写入数据: 1

(1)	(2)
-----	-----

编号	数据名	数据	
		以ASCII代码(16进制数)进行数据通信时	以二进制代码(16进制数)进行数据通信时
(1)	写入数据长: 2	30303032	0200
(2)	写入数据: 1	30303031	0100

响应数据

无Label Write Random指令的响应数据。

通信示例

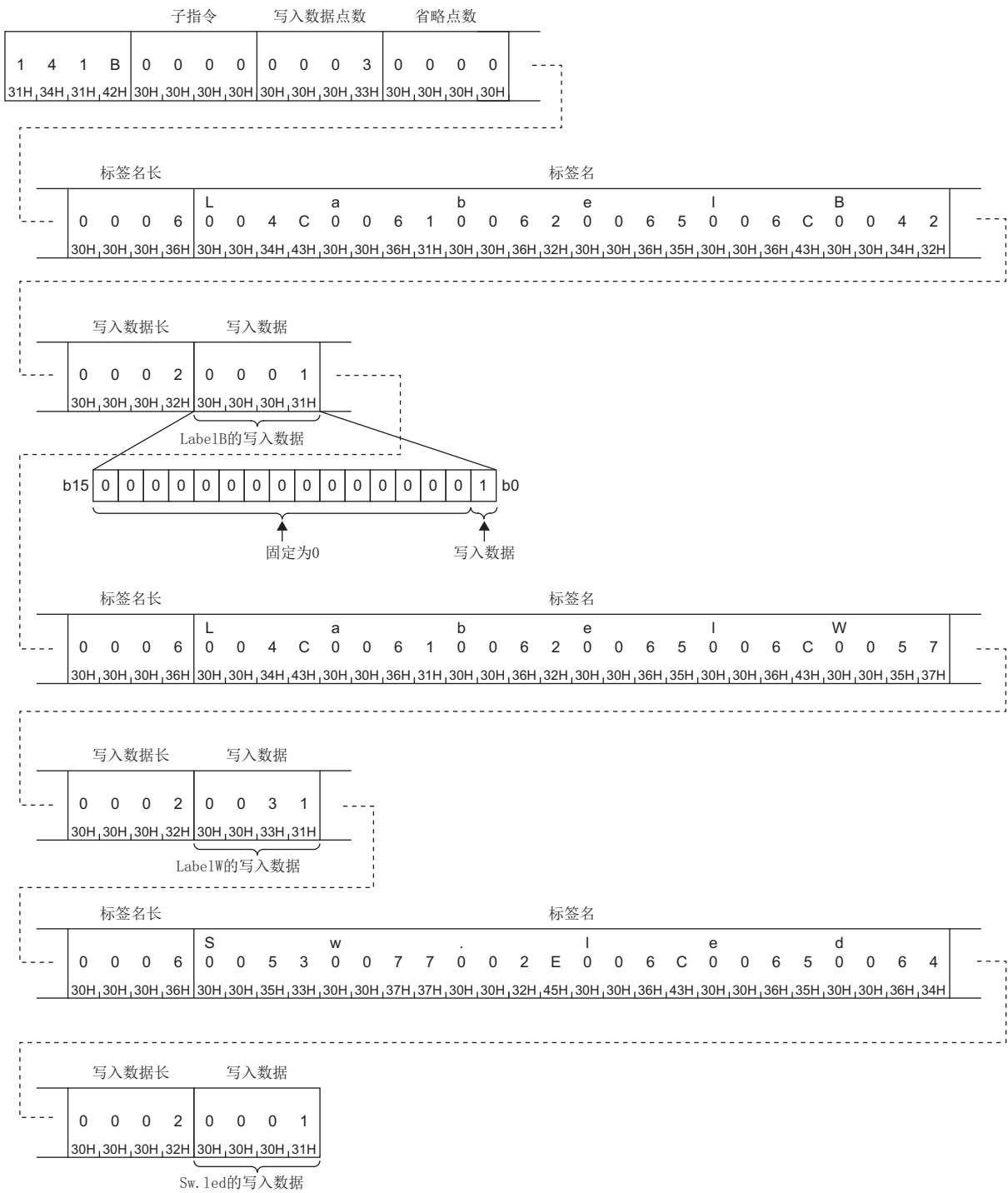
将数据写入下述3种标签中。

- 数据类型为位的基本数据类型标签 “LabelB”
- 数据类型为字的基本数据类型标签 “LabelW”
- 数据类型为字的结构体类型标签 “Sw.lcd”

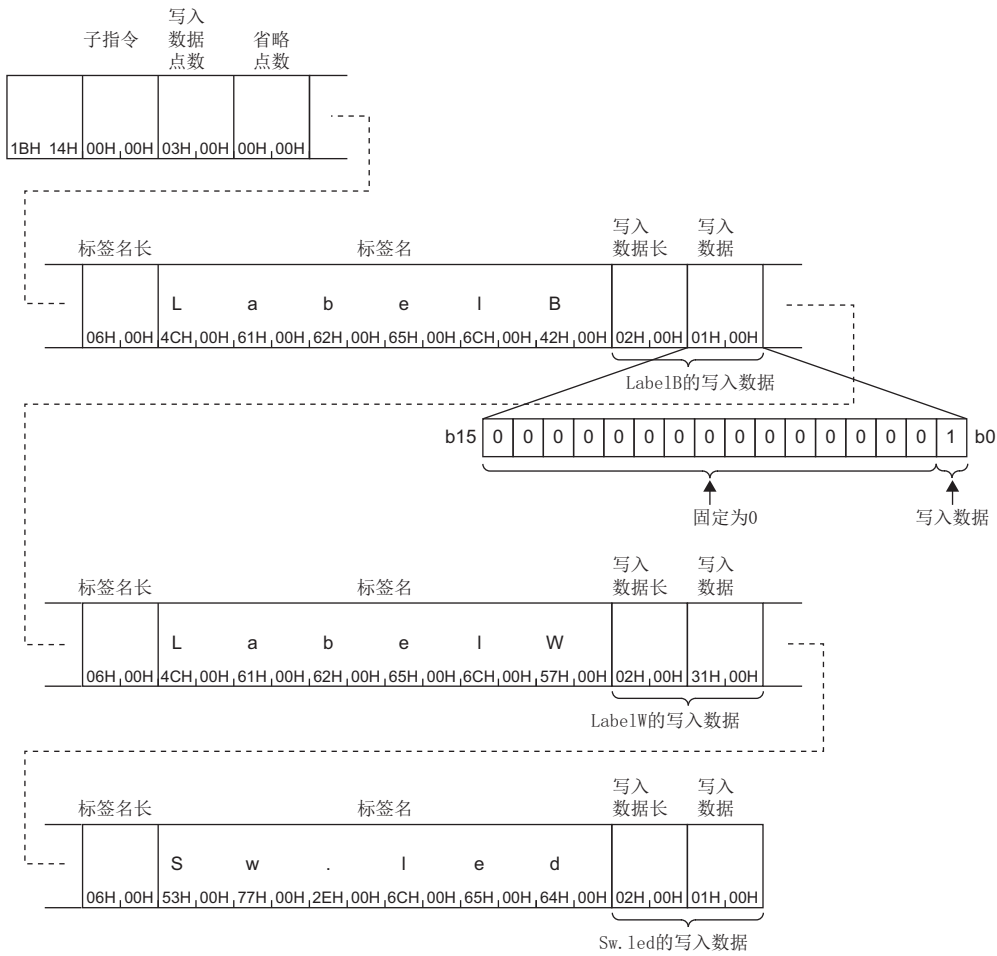
标签是写入下述值的标签。

- LabelB:1(0N)
- LabelW:0031H
- Sw.lcd:0001H

■以ASCII代码进行数据通信时
(请求数据)



■以二进制代码进行数据通信时 (请求数据)



5.4 Memory(本站缓冲存储器访问)

以下说明进行本站的SLMP对应设备的缓冲存储器读取及写入的指令有关内容。

指令内指定的数据

请求目标网络编号、请求目标站号

将访问目标指定为本站。(不可以进行其它站指定。)

- 请求目标网络编号：00H
- 请求目标站号：FFH

起始地址

指定进行读取或写入的缓冲存储器的起始地址。

■以ASCII代码进行数据通信时

将地址转换为ASCII代码8位(16进制数)后，按照从高位字节到低位字节的顺序发送。指定英文的情况下，使用大写字母代码。

例

地址为1E1H的情况下

0	0	0	0	0	1	E	1
30H	30H	30H	30H	30H	31H	45H	31H

■以二进制代码进行数据通信时

使用4字节的数值，按照从低位字节到高位字节的顺序发送。

例

地址为1E1H的情况下

E1H	01H	00H	00H
-----	-----	-----	-----

字长

指定进行读取或写入的缓冲存储器的字长。

■以ASCII代码进行数据通信时

将字长转换为ASCII代码4位(16进制数)使用后，按照从高位字节到低位字节的顺序发送。指定英文的情况下，使用大写字母代码。

例

5字、20字的情况下

5字

0	0	0	5
30H	30H	30H	35H

20字

0	0	1	4
30H	30H	31H	34H

■以二进制代码进行数据通信时

使用2字节的数值，按照从低位字节到高位字节的顺序发送。

例

5字、20字的情况下

5字



20字



读取数据、写入数据

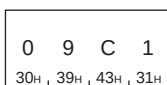
读取的情况下，存储已读取的缓冲存储器的值。写入的情况下，存储写入的数据。

■以ASCII代码进行数据通信时

以ASCII代码4位(16进制数)被存储。

例

09C1H的情况下



■以二进制代码进行数据通信时

使用2字节的数值，按照从低位字节到高位字节的顺序被存储。

例

09C1H的情况下



Read(指令： 0613)

读取本站 (SLMP对应设备) 的缓冲存储器数据。

要点

- 本指令不可以访问下述缓冲存储器。
- 本站 (SLMP对应设备) 上安装的智能功能模块
 - 其它站缓冲存储器
- 访问上述缓冲存储器的情况下，应使用Device(软元件访问)指令，访问缓冲存储器。(190页 通过软元件的扩展指定进行读取、写入)

请求数据

ASCII

0	6	1	3	0	0	0	0	起始地址	字长
30H	36H	31H	33H	30H	30H	30H	30H		

二进制

		起始地址	字长
13H	06H	00H	00H

■起始地址

指定读取的缓冲存储器的起始地址。(108页 起始地址)

■字长

指定读取的缓冲存储器的字长。(108页 字长)

- 指定范围：1H～1E0H(480)

响应数据

已读取的缓冲存储器的值按照从高位字节到低位字节的顺序以16进制数被存储。(109页 读取数据、写入数据)

读取数据1	～	读取数据n
-------	---	-------

通信示例

读取缓冲存储器地址78H~81H(120~129)的内容。

■以ASCII代码进行数据通信时
(请求数据)

起始地址												字长				
0	6	1	3	0	0	0	0	0	0	0	7	8	0	0	0	A
30H	36H	31H	33H	30H	30H	30H	30H	30H	30H	30H	37H	38H	30H	30H	30H	41H

(响应数据)

读取数据1				读取数据2				读取数据10			
0	5	0	0	0	9	C	1	0	0	C	8
30H	35H	30H	30H	30H	39H	43H	31H	30H	30H	43H	38H
地址78H的值 =0500H				地址79H的值 =09C1H				地址81H的值 =00C8H			

■以二进制代码进行数据通信时
(请求数据)

起始地址				字长
13H	06H	00H	00H	78H, 00H, 00H, 00H
0AH	00H			

(响应数据)

读取数据1	读取数据2	读取数据10
00H, 05H	C1H, 09H	C8H, 00H
地址78H的值 =0500H	地址79H的值 =09C1H	地址81H的值 =00C8H

Write(指令：1613)

将数据写入到本站(SLMP对应设备)的缓冲存储器中。

要点

- 本指令不可以访问下述缓冲存储器。
- 本站(SLMP对应设备)上安装的智能功能模块
 - 其它站缓冲存储器
- 访问上述缓冲存储器的情况下，应使用Device(软元件访问)指令，访问缓冲存储器。(P190页 通过软元件的扩展指定进行读取、写入)
- 请勿将数据写入到缓冲存储器的“系统区域”。如果将数据写入到“系统区域”，有可能导致可编程控制器系统误动作。

请求数据

ASCII

1	6	1	3	0	0	0	0	起始地址	字长	写入数据	写入数据
31H	36H	31H	33H	30H	30H	30H	30H				

二进制

		起始地址	字长	写入数据1	~	写入数据n
13H	16H	00H	00H			

■起始地址

指定写入的缓冲存储器的起始地址。(P108页 起始地址)

■字长

指定写入的缓冲存储器的字长。(P108页 字长)

- 指定范围：1H~1E0H(480)

响应数据

无Write指令的响应数据。

通信示例

将值写入到缓冲存储器地址2680H~2683H(9856~9859)。

■以ASCII代码进行数据通信时

(请求数据)

起始地址																字长				写入数据1				写入数据4				
1	6	1	3	0	0	0	0	0	0	0	0	2	6	8	0	0	0	0	4	2	0	0	0	0	0	0	0	0
31H	36H	31H	33H	30H	30H	30H	30H	30H	30H	30H	30H	32H	36H	38H	30H	30H	30H	30H	34H	32H	30H	30H	30H	30H	30H	30H	30H	30H
																地址2680H的值 =2000H				地址2683H的值 =0H								

■以二进制代码进行数据通信时

(请求数据)

起始地址				字长	写入数据1	写入数据4
13H, 16H	00H, 00H	80H, 26H, 00H, 00H	04H, 00H	00H, 20H	00H, 00H	
				地址2680H的值 =2000H		地址2683H的值 =0H

5.5 Extend Unit(智能功能模块的缓冲存储器访问)

以下说明进行智能功能模块的缓冲存储器读取及写入的指令有关内容。

Extend Unit中，可以访问下述所示的MELSEC-Q系列智能功能模块。对下述模块以外的缓冲存储器进行访问的情况下，应通过Read(指令：0401，子指令：008□)，或Write(指令：1401，子指令：008□)，指定模块访问软元件后进行访问。

☞ 194页 至模块访问软元件的访问

模块型号	起始地址 ^{*1}	0插槽上安装了时的模块No. ^{*1}
QD35ID1/ID2型ID接口模块	4000H	0000H
Q62AD-DGH、Q64AD(-GH)、Q66AD-DG、Q68AD-G、Q68ADV/ADI型模—数转换模块	1008H	
Q62DA(-FG)、Q62DAN、Q64DA、Q64DAN、Q66DA-G、Q68DAV/Q68DAI、Q68DAVN/Q68DAIN型数—模转换模块	1008H	
Q64AD2DA模拟输入输出模块	2000H	
Q62HLC型环路控制模块	10000H	
Q64TCIT/Q64TCRT型温度调节模块	1000H	
Q61LD称重传感器输入模块	2000H	
Q64TCITBW/Q64TCRTBW型温度调节模块	1000H	
Q64TD、Q64RD型温度输入模块(功能版本B)	2000H	
Q64TD、Q64TDV-GH、Q64RD(-G)型温度输入模块(功能版本C)	8000H	
Q68TD-G-H01、Q68TD-G-H02型通道间绝缘热电偶输入模块	1008H	0000H
Q68RD3-G型通道间绝缘测温电阻输入模块	1008H	
QD51(-R24)型智能通信模块	10000H	
QD60P8-G型通道间绝缘脉冲输入模块	2000H	
QD62、QD62E、QD62D型高速计数器模块	3CH	
QD63P6型多通道高速计数器模块	2000H	
QD64D2型4Mbps对应高速计数器模块	2000H	
QD70P4/P8型定位模块	5000H	
QD70D4/D8型定位模块	5000H	
QD72P3C3型计数器功能内置定位模块	5000H	
QD75P1/P2/P4、QD75D1/D2/D4、QD75M1/M2/M4、QD75MH1/MH2/MH4型定位模块	10000H	0000H
QD81DL96高速数据记录模块	10000H	
QJ61BT11(N)型CC-Link系统主站·本地站模块	10000H	
QJ61CL12型CC-Link/LT主站模块	01B4H	
QJ71C24N(-R2/R4)、QJ71C24(-R2)型串行通信模块	10000H	
QJ71AS92型AS-i主站模块	10000H	
QJ71CMO(N)型调制解调器接口模块	10000H	
QJ71E71-100/-B5/-B2型以太网接口模块	10000H	
QJ71FL71-T/-B5/-B2-F01型FL-net(OPCN-2)接口模块	10000H	
QJ71MES96型MES接口模块	10000H	
QJ71WS96 型Web 服务器模块	10000H	

*1 “起始地址”及“0插槽上安装了时的模块No.”在请求数据中使用。

☞ 116页 Read(指令：0601)

☞ 118页 Write(指令：1601)

指令内指定的数据

起始地址

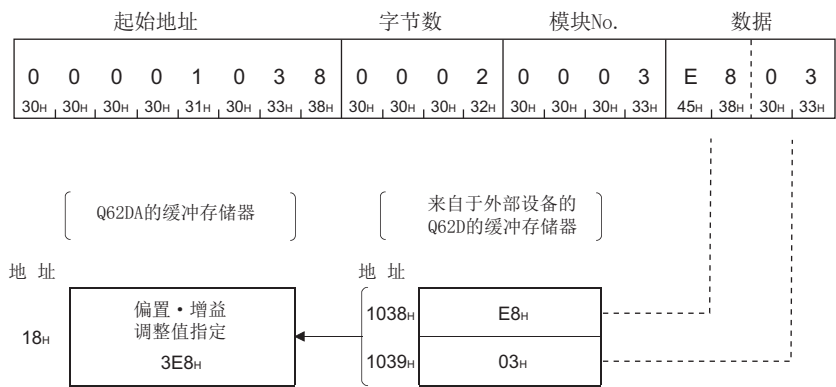
指定进行读取或写入的缓冲存储器的起始地址。数据发送顺序与Memory(缓冲存储器访问)相同。(108页 起始地址)
起始地址按下述方式计算。
起始地址= {(将模块缓冲存储器地址×2)以16进制数化为的数} +(Extend Unit(智能功能模块的缓冲存储器访问)的表中所示的“起始地址”)*1

*1 计算公式中使用的起始地址应使用下述表中所示的“起始地址”。
113页 Extend Unit(智能功能模块的缓冲存储器访问)

例

指定输入输出信号为30H~4FH(模块No.: 03H)的Q62DA的缓冲存储器地址18H的情况下
(18H×2)+1008H=30H+1008H=1038H

Q62DA的缓冲存储器内容	起始地址	模块的缓冲存储器地址
D/A转换允许/禁止	1008H	0H
	1009H	
CH.1数字值	100AH	1H
	100BH	
CH.2数字值	100CH	2H
	100DH	
系统区域	—	3H~10H
偏置·增益调节值指定	1038H	18H



字节数

指定进行读取或写入的缓冲存储器的字节数。数据发送顺序与Memory(缓冲存储器访问)相同。(108页 字长)
对于智能功能模块的缓冲存储器，由于1个区域以2字节(1字)被构成，因此通过地址数×2(字节数)指定。

模块No.

指定进行读取或写入的智能功能模块。

模块No.按下述方式计算。

模块No.=(以4位表现了智能功能模块的起始输入输出编号时的前3位)+(Extend Unit(智能功能模块的缓冲存储器访问)的表中所示的“0插槽上安装了时的模块No.”)*¹

*1 计算公式中使用的0插槽上安装了时的模块No.应使用下述表的“0插槽上安装了时的模块No.”。

☞ 113页 Extend Unit(智能功能模块的缓冲存储器访问)

■以ASCII代码进行数据通信时

将模块No.转换为ASCII代码4位(16进制数)使用后,按照从高位字节到低位字节的顺序发送。

例

起始输入输出编号为0080H的情况下

模块No.将变为“0008”,从“0”开始按顺序发送。

0	0	0	8
30H	30H	30H	38H

■以二进制代码进行数据通信时

将模块No.按照从低位字节到高位字节的顺序发送。

例

起始输入输出编号为0080H的情况下

模块No.将变为“0008H”,按照08H、00H开始按顺序发送。

08H, 00H

读取数据、写入数据

读取的情况下,存储已读取的缓冲存储器的值。写入的情况下,存储写入的数据。

■以ASCII代码进行数据通信时

以ASCII代码2位(16进制数)被存储。

例

09C1H的情况下

C	1	0	9
43H	31H	30H	39H

缓冲存储器地址

1个的数据

■以二进制代码进行数据通信时

以1字节为单位,按照从低位字节到高位字节的顺序被存储。

例

09C1H的情况下

C1H, 09H

Read(指令： 0601)

读取智能功能模块的缓冲存储器数据。

请求数据

0	6	0	1	0	0	0	0	起始地址	字节数	模块No.
30H	36H	30H	31H	30H	30H	30H	30H			

二进制

		起始地址	字节数	模块No.
01H	06H	00H	00H	

■起始地址

指定读取的缓冲存储器的起始地址。(114页 起始地址)

■字节数

指定读取的缓冲存储器的字节数。(114页 字节数)

- 指定范围：2H～780H(1920)

■模块No.

指定进行读取的智能功能模块。(115页 模块No.)

响应数据

已读取的缓冲存储器的值以16进制数被存储。(115页 读取数据、写入数据)

读取数据1	～	读取数据n
-------	---	-------

通信示例

读取输入输出信号为30H~4FH(模块No.: 03H)的Q62DA的缓冲存储器地址1H~2H的内容。

■以ASCII代码进行数据通信时

(请求数据)

起始地址				字节数	模块No.
0	6	0	1	0 0 0 0 1 0 0 A	0 0 0 4 0 0 0 3
30 _H	36 _H	30 _H	31 _H	30 _H 30 _H 30 _H 30 _H 31 _H 30 _H 30 _H 41 _H	30 _H 30 _H 30 _H 34 _H 30 _H 30 _H 30 _H 33 _H

(响应数据)

0	1	0	0	1	2	0	0
30 _H	31 _H	30 _H	30 _H	31 _H	32 _H	30 _H	30 _H
地址1 _H 的值 =0001 _H				地址2 _H 的值 =0012 _H			

■以二进制代码进行数据通信时

(请求数据)

起始地址		字节数	模块No.
01 _H	06 _H	00 _H 00 _H	0A _H 10 _H 00 _H 00 _H
04 _H	00 _H	03 _H	00 _H

(响应数据)

01 _H	00 _H	12 _H	00 _H
地址1 _H 的值 =0001 _H		地址2 _H 的值 =0012 _H	

Write(指令：1601)

将数据写入到智能功能模块的缓冲存储器中。

请求数据

ASCII

1	6	0	1	0	0	0	0	起始地址	字节数	模块No.	写入数据
31H	36H	30H	31H	30H	30H	30H	30H				~

二进制

		起始地址	字节数	模块No.	写入数据
01H	16H	00H	00H		~

■起始地址

指定写入的缓冲存储器的起始地址。(114页 起始地址)

■字节数

指定写入的缓冲存储器的字节数。(114页 字节数)

- 指定范围：2H～780H(1920)

■模块No.

指定进行写入的智能功能模块。(115页 模块No.)

■写入数据

将写入的数据指定到缓冲存储器中。(115页 读取数据、写入数据)

响应数据

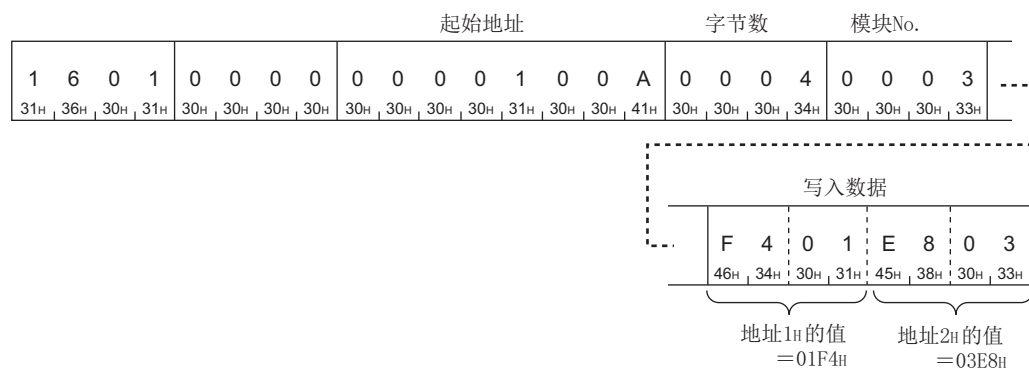
无Write指令的响应数据。

通信示例

将数据写入到输入输出信号为30H~4FH(模块No.: 03H)的Q62DA的缓冲存储器地址1H~2H中。

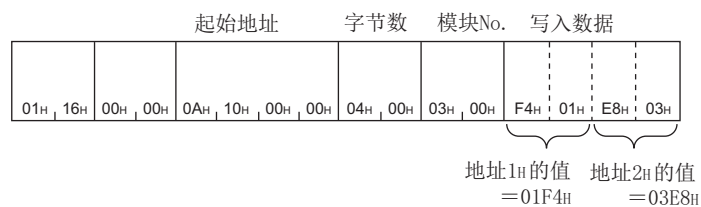
■以ASCII代码进行数据通信时

(请求数据)



■以二进制代码进行数据通信时

(请求数据)



5.6 Remote Control(远程操作)

以下对通过来自于外部设备的报文，将SLMP对应设备及CPU模块置为RUN状态及STOP状态等的指令有关内容进行说明。

在远程操作之前

在远程操作后进行了访问目标电源OFF→ON或复位的情况下

远程操作信息将被删除。

例

CPU模块开关为RUN状态下进行远程STOP，复位CPU模块时将变为RUN状态。

对访问目标CPU模块施加系统保护时

不可以通过外部设备进行远程操作。在访问目标中将发生出错，异常响应被回复到外部设备中。应在解除CPU模块侧的系统保护之后，再次发送请求报文。

对于SLMP对应设备执行远程操作的情况下

建议协议使用UDP，进行远程操作。使用TCP的情况下，由于在复位时连接被切断，因此需要再次确立连接。

通过1次指令可操作的站

可通过1次指令对1站进行远程操作。

Remote Run(指令：1001)

对于访问目标模块执行远程RUN。

要点

远程RUN可以在访问目标模块开关为RUN时使用。访问目标模块开关为STOP的情况下，Remote Run(指令：1001)虽然正常完成，但是访问目标不变为RUN状态。

请求数据

ASCII									
1	0	0	1	0	0	0	0	模式	清除模式
31H	30H	30H	31H	30H	30H	30H	30H		0 0
									30H 30H

二进制				
		模式	清除模式	
01H	10H	00H	00H	00H

■模式

对是否通过进行了远程STOP或远程PAUSE的外部设备以外强制执行远程RUN进行指定。不强制执行的情况下，仅通过进行了远程STOP或远程PAUSE的外部设备可以进行远程RUN。

强制执行是在外部设备故障导致进行了远程操作的设备无法进行远程RUN时使用。

项目	模式											
	ASCII代码	二进制代码										
不强制执行。(通过其它外部设备远程STOP或远程PAUSE过程中，不进行远程RUN。)	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table>	0	0	0	1	30H	30H	30H	31H	<table><tr><td>01H</td><td>00H</td></tr></table>	01H	00H
0	0	0	1									
30H	30H	30H	31H									
01H	00H											
强制执行。(即使通过其它外部设备远程STOP或远程PAUSE过程中，也进行远程RUN。)	<table><tr><td>0</td><td>0</td><td>0</td><td>3</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>33H</td></tr></table>	0	0	0	3	30H	30H	30H	33H	<table><tr><td>03H</td><td>00H</td></tr></table>	03H	00H
0	0	0	3									
30H	30H	30H	33H									
03H	00H											

■清除模式

在远程RUN的运算开始时，对是否进行软元件清除(初始化)处理进行指定。接收了远程RUN请求的设备在软元件清除(初始化)后将变为RUN状态。

CPU模块的情况下，应按照可编程控制器参数的“可编程控制器文件设置”的“软元件初始值”，进行软元件清除(初始化)处理。

项目	模式	
	ASCII代码	二进制代码
不清除软元件。	00 30H, 30H	00H
清除锁存范围外的软元件。	01 30H, 31H	01H
清除包括锁存范围在内的全部软元件。	02 30H, 32H	02H

响应数据

无Remote Run指令的响应数据。

通信示例

模块通过“不强制执行”、清除模式通过“清除包括锁存范围在内的全部软元件”进行远程RUN。

■以ASCII代码进行数据通信时

(请求数据)

模式				清除模式	
1	0	0	1	0	2
31H, 30H, 30H, 31H	0	0	0	0	0
30H, 30H, 30H, 30H	0	0	0	1	0
30H, 30H, 30H, 31H	30H, 32H	30H, 30H			

■以二进制代码进行数据通信时

(请求数据)

模式		清除模式	
01H, 10H	00H, 00H	01H, 00H	02H, 00H

Remote Stop(指令：1002)

对于访问目标模块执行远程STOP。

请求数据

ASCII

1	0	0	2	0	0	0	0	0	0	0	1
31H	30H	30H	32H	30H	30H	30H	30H	30H	30H	30H	31H

二进制

02H, 10H	00H, 00H	01H, 00H

响应数据

无Remote Stop指令的响应数据。

通信示例

以上述“请求数据”中所示的报文格式，通过外部设备发送请求报文。

Remote Pause(指令：1003)

对于访问目标模块执行远程PAUSE。

要点

远程PAUSE可以在访问目标模块开关为RUN时使用。访问目标模块开关为STOP的情况下，Remote Pause(指令：1003)虽然正常完成，但是访问目标不变为PAUSE状态。

请求数据

ASCII

1	0	0	3	0	0	0	0	模式
31H	30H	30H	33H	30H	30H	30H	30H	

二进制

		模式
03H	10H	00H
00H	00H	

模式

对是否通过进行了远程STOP或远程PAUSE的外部设备以外强制执行远程PAUSE进行指定。不强制执行的情况下，仅通过进行了远程STOP或远程PAUSE的外部设备可以进行远程PAUSE。
强制执行是在外部设备故障导致进行了远程操作的设备无法进行远程PAUSE时使用。

项目	模式	
	ASCII代码	二进制代码
不强制执行。(通过其它外部设备远程STOP或远程PAUSE过程中，不进行远程PAUSE。)	0001 30H, 30H, 30H, 31H	01H, 00H
强制执行。(即使通过其它外部设备远程STOP或远程PAUSE过程中，进行远程PAUSE。)	0003 30H, 30H, 30H, 33H	03H, 00H

响应数据

无Remote Pause指令的响应数据。

通信示例

模式通过“不强制执行”进行远程PAUSE。

以ASCII代码进行数据通信时 (请求数据)

模式								
1	0	0	3	0	0	0	0	
31H	30H	30H	33H	30H	30H	30H	30H	

以二进制代码进行数据通信时 (请求数据)

模式		
03H	10H	00H
00H	00H	01H

Remote Latch Clear(指令：1005)

对于访问目标模块执行远程锁存清除。

要点

- 应在进行远程锁存清除之前将访问目标模块置为STOP状态。
- 通过来自其它外部设备等的请求，访问目标为远程STOP或远程PAUSE中的情况下，不可以进行远程锁存清除。指令将异常完成。应在解除远程STOP或远程PAUSE之后，再执行指令。

请求数据

ASCII

1	0	0	5	0	0	0	0	0	0	0	1
31H	30H	30H	35H	30H	30H	30H	30H	30H	30H	30H	31H

二进制

05H, 10H	00H, 00H	01H, 00H

响应数据

无Remote Latch Clear指令的响应数据。

通信示例

以上述“请求数据”中所示的报文格式，通过外部设备发送请求报文。

Remote Reset(指令：1006)

对于访问目标模块执行远程RESET。远程RESET在模块中发生出错时进行出错修复时使用。

要点

- 在进行远程RESET之前访问目标参数中有远程RESET的允许/禁止设置的情况下，应将远程RESET设置为允许。此外，应在进行远程RESET之前将访问目标模块置为STOP状态。
- 由于访问目标硬件异常等，有可能无法远程RESET。
- 远程RESET时，由于访问目标被复位，有可能外部设备中无法回复响应报文。

请求数据

ASCII

1	0	0	6	0	0	0	0	0	0	0	1
31H	30H	30H	36H	30H	30H	30H	30H	30H	30H	30H	31H

二进制

06H, 10H	00H, 00H	01H, 00H

响应数据

无Remote Reset指令的响应数据。

通信示例

以上述“请求数据”中所示的报文格式，通过外部设备发送请求报文。

Read Type Name(指令：0101)

读取访问目标模块的型号及型号代码。

请求数据

ASCII

0	1	0	1	0	0	0	0
30H	31H	30H	31H	30H	30H	30H	30H

二进制

01H, 01H	00H, 00H

响应数据

5

ASCII

型号	型号代码

二进制

型号	型号代码

■型号

模块型号从高位字节16字符被存储。
读取的型号未满足16字符的情况下，剩余字符将存储空格(20H)。在以二进制代码进行通信时，模块型号也以ASCII代码被存储。

■型号代码

存储下述型号代码。

以ASCII代码进行通信时，按照从高位字节到低位字节的顺序被存储。

以二进制代码进行通信时，按照从低位字节到高位字节的顺序被存储。(129页 通信示例)

型号	型号代码
Q00JCPU	250H
Q00CPU	251H
Q01CPU	252H
Q02CPU、Q02HCPU、Q02PHCPU	41H
Q06HCPU、Q06PHCPU	42H
Q12HCPU、Q12PHCPU	43H
Q25HCPU、Q25PHCPU	44H
Q12PRHCPU	4BH
Q25PRHCPU	4CH
Q00UJCPU	260H
Q00UCPU	261H
Q01UCPU	262H
Q02UCPU	263H
Q03UDCPU、Q03UDECPU	268H
Q03UDVCPU	366H
Q04UDHCPU、Q04UDEHCPU	269H
Q04UDVCPU、Q04UDPVCPU	367H
Q06UDHCPU、Q06UDEHCPU	26AH
Q06UDVCPU、Q06UDPVCPU	368H
Q10UDHCPU、Q10UDEHCPU	266H
Q13UDHCPU、Q13UDEHCPU	26BH
Q13UDVCPU、Q13UDPVCPU	36AH
Q20UDHCPU、Q20UDEHCPU	267H
Q26UDHCPU、Q26UDEHCPU	26CH
Q26UDVCPU、Q26UDPVCPU	36CH
Q50UDEHCPU	26DH
Q100UDEHCPU	26EH
L02SCPU、L02SCPU-P	543H
L02CPU、L02CPU-P	541H
L06CPU、L06CPU-P	544H
L26CPU、L26CPU-P	545H
L26CPU-BT、L26CPU-PBT	542H
QS001CPU	230H
LJ72GF15-T2	0641H
NZ2GF-ETB	0642H
R04CPU	4800H
R08CPU	4801H
R16CPU	4802H
R32CPU	4803H
R120CPU	4804H

要点

- CPU模块型号应以型号代码判别。
- 连接站为MELSEC iQ-R系列以外的模块中，RCPU中执行了指令的情况下，型号中“RCPU”、型号代码中“0360H”被存储。

通信示例

对于Q02UCPU执行指令，读取型号及型号代码。

■以ASCII代码进行数据通信时

(请求数据)

0	1	0	1	0	0	0	0
30 _H	31 _H	30 _H	31 _H	30 _H	30 _H	30 _H	30 _H

(响应数据)

型号														型号代码			
Q	0	2	U	C	P	U								0	2	6	3
51 _H	30 _H	32 _H	55 _H	43 _H	50 _H	55 _H	20 _H	20 _H	20 _H	20 _H	20 _H	20 _H	20 _H	30 _H	32 _H	36 _H	33 _H

■以二进制代码进行数据通信时

(请求数据)

01 _H	01 _H	00 _H	00 _H

(响应数据)

型号														型号代码			
Q	0	2	U	C	P	U											
51 _H	30 _H	32 _H	55 _H	43 _H	50 _H	55 _H	20 _H	20 _H	20 _H	20 _H	20 _H	20 _H	20 _H	63 _H	02 _H		

5.7 Remote Password(远程口令)

以下对进行远程口令的解锁处理及锁定处理的指令有关内容进行说明。

注意事项

- MELSEC iQ-R系列模块的口令字符数与，MELSEC-Q/L系列模块的口令字符数有所不同。(130页 远程口令长)

指令内指定的数据

远程口令长

- MELSEC-Q/L系列模块固定为4字符。
- MELSEC iQ-R系列模块对指定的远程口令的字符数(6字符～32字符)进行指定。

项目	远程口令长	
	ASCII代码	二进制代码
MELSEC-Q/L系列(固定为4字符)	0 0 0 4 30H, 30H, 30H, 34H	04H, 00H
MELSEC iQ-R系列模块(远程口令的字符数为32字符的情况下)	0 0 2 0 30H, 30H, 32H, 30H	20H, 00H

远程口令

是通过工程工具CPU模块、MELSEC iQ-R系列对应的智能功能模块中设置的远程口令。

远程口令在以二进制代码进行通信时也以ASCII代码指定。

■以ASCII/二进制代码进行数据通信时

将被设置的远程口令直接通过起始字符发送。

Lock(指令：1631)

指定远程口令，从解锁状态置为锁定状态。(置为对于SLMP对应设备不可通信状态。)

- 要点
- 对于已经变为锁定状态的对象设备发送了Lock指令的情况下，对象设备将保持为锁定状态不变。(也不实施口令校验。)
 - 可以仅对于与外部设备连接的连接站执行本指令。对于经由网络的其它站模块不可以使用。

请求数据

ASCII															
1 6 3 1				子指令				远程口令长				远程口令			
31H, 36H, 33H, 31H															

二进制

				子指令				远程 口令长				远程口令			
31H, 16H															

子指令

子指令					
ASCII代码					
0 0 0 0					
30H, 30H, 30H, 30H					
二进制代码					
00H, 00H					

远程口令长

指定远程口令字符数。(130页 远程口令长)

远程口令

指定设置的远程口令。(130页 远程口令)

响应数据

无Lock指令的响应数据。

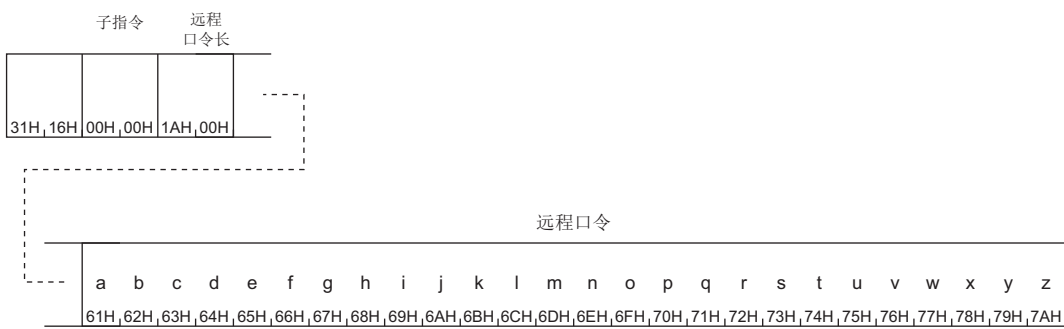
通信示例

MELSEC iQ-R系列模块中被设置的远程口令为“abcdefghijklmnopqrstuvwxyz”的情况下，将置为锁定状态。

以ASCII代码进行数据通信时
(请求数据)

子指令				远程口令长							
1	6	3	1	0	0	0	0	0	0	1	A
31H	36H	33H	31H	30H	30H	30H	30H	30H	30H	31H	41H
远程口令											
a	b	c	d	e	f	g	h	i	j	k	l
61H	62H	63H	64H	65H	66H	67H	68H	69H	6AH	6BH	6CH
m	n	o	p	q	r	s	t	u	v	w	x
6DH	6EH	6FH	70H	71H	72H	73H	74H	75H	76H	77H	78H
y	z										
79H	7AH										

■以二进制代码进行数据通信时
(请求数据)



Unlock(指令: 1630)

指定远程口令，从锁定状态置为解锁状态。（置为对于SLMP对应设备可通信状态。）

要点

- 如果连续一定次数弄错口令，将闭锁，且有可能一定时间无法解除口令。
- 对于已经变为解锁状态的对象设备发送了Unlock指令的情况下，对象设备将保持为解锁状态不变。（也不实施口令校验。）
- 可以仅对于与外部设备连接的连接站执行本指令。对于经由网络的其它站模块不可以使用。

请求数据

ASCTT

1 6 3 0 31H, 36H, 33H, 30H	子指令	远程口令长	远程口令
-------------------------------	-----	-------	------

二进制

30H, 16H	子指令	远程 口令长	远程口令
----------	-----	-----------	------

■子指令

子指令

ASCII代码	二进制代码
0000 30H30H30H30H	00H00H

■远程口令长

指定远程口令字符数。(👉 130页 远程口令长)

■远程口令

指定设置的远程口令。(👉 130页 远程口令)

响应数据

无Unlock指令的响应数据。

通信示例

MELSEC iQ-R系列模块中，被设置的远程口令为“abcdefghijklmnopqrstuvwxyz”的情况下，将置为解锁状态。

■以ASCII代码进行数据通信时

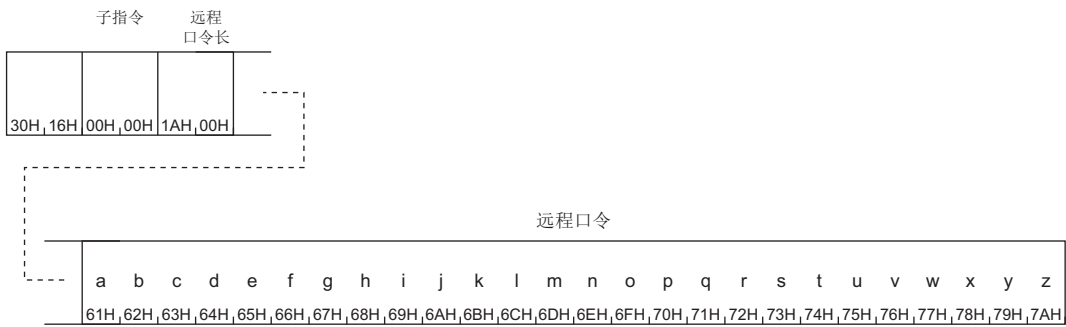
(请求数据)

子指令				远程口令长								
1	6	3	0	0	0	0	0	0	0	1	A	
31H,36H,33H,30H				30H,30H,30H,30H				30H,30H,31H,41H				

远程口令

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
61H	62H	63H	64H	65H	66H	67H	68H	69H	6AH	6BH	6CH	6DH	6EH	6FH	70H	71H	72H	73H	74H	75H	76H	77H	78H	79H	7AH

■以二进制代码进行数据通信时
(请求数据)



5.8 File(文件控制)

以下对操作SLMP对应设备或CPU模块内的文件的指令有关内容进行说明。

File指令在外部设备通过CPU模块读取参数及程序等保存，且根据控制内容通过外部设备将参数及程序等写入至CPU模块的情况下使用。

关于CPU模块中存储的文件的文件名、扩展名、存储位置有关内容，请参阅CPU模块手册。

要点

- File(文件控制)中，虽然有可能也会看到所使用的CPU模块用户手册中记载的文件以外，但是由于为系统用文件，因此请勿访问。
-

指令内指定的数据

口令

■子指令为0000、0004的情况下

指定访问的文件口令。口令在子指令为0000时为4字符，子指令为0004时为32字符的固定长。口令短于固定长的情况下，对未达到固定长的部分指定空格(20H)。
口令在以二进制代码进行通信时也以ASCII代码指定。

要点

- 允许或禁止至程序文件、软元件注释文件、软元件初始值文件的访问。可以对各文件设置读取专用或读取/写入禁止。
- 文件中未设置口令的情况下，将添加空格(代码：20H)。

以ASCII/二进制代码进行数据通信时																	
设置口令的情况下	未设置口令的情况下																
口令为“ABCDEF”时的示例如下所示。 (ASCII代码及二进制代码相同)	对空格(代码: 20H)进行口令字符数指定。																
<table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>...</td></tr><tr><td>41H, 42H, 43H, 44H, 45H, 46H, 20H, 20H, ...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	A	B	C	D	E	F	...	41H, 42H, 43H, 44H, 45H, 46H, 20H, 20H, ...							<table><tr><td>...</td></tr><tr><td>20H, 20H, ..., 20H</td></tr></table>	...	20H, 20H, ..., 20H
A	B	C	D	E	F	...											
41H, 42H, 43H, 44H, 45H, 46H, 20H, 20H, ...																	
...																	
20H, 20H, ..., 20H																	

■子指令为0040的情况下

指定访问的文件的口令字符数与口令。
口令的字符数在口令之前以6～32的范围通过16进制数指定。
口令以6～32字符指定。
口令在以二进制代码进行通信时也以ASCII代码指定。
未设置口令的情况下对口令字符数指定“0”，口令在未指定的状况下将置为左端。

要点

如果连续一定次数弄错口令，将闭锁，且有可能一定时间无法解除口令。

以ASCII代码进行数据通信时									
设置口令的情况下	未设置口令的情况下								
口令为“ ABCDEFGHIJKLMNOPQRSTUVWXYZ ”(口令字符数为26字符)时的示例如下所示。 口令字符数转换为ASCII代码4位使用后，按照从高位字节到低位字节的顺序发送。 <table><tr><td>口令字符数</td><td>口令</td></tr><tr><td>0 0 1 A</td><td>A B C D … Z</td></tr><tr><td>30H,30H,31H,41H</td><td>41H,42H,43H,44H, ,5AH</td></tr></table>	口令字符数	口令	0 0 1 A	A B C D … Z	30H,30H,31H,41H	41H,42H,43H,44H, ,5AH	对口令字符数指定“ 30H ”(0)。 不指定口令。 <table><tr><td>0 0 0 0</td></tr><tr><td>30H, 30H, 30H, 30H</td></tr></table>	0 0 0 0	30H, 30H, 30H, 30H
口令字符数	口令								
0 0 1 A	A B C D … Z								
30H,30H,31H,41H	41H,42H,43H,44H, ,5AH								
0 0 0 0									
30H, 30H, 30H, 30H									

以二进制代码进行数据通信时																
设置口令的情况下	未设置口令的情况下															
口令为“ ABCDEFGHIJKLMNOPQRSTUVWXYZ ”(口令字符数为26字符)时的示例如下所示。 口令字符数使用2字节的数值，按照从低位字节到高位字节的顺序发送。 口令 字符数 口令 <table><tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td><td>...</td><td>Z</td></tr><tr><td>1AH, 00H</td><td>41H</td><td>42H</td><td>43H</td><td>44H</td><td></td><td>5AH</td></tr></table>		A	B	C	D	...	Z	1AH, 00H	41H	42H	43H	44H		5AH	对口令字符数指定“ 0 ”。 不指定口令。 <table><tr><td>00H, 00H</td></tr></table>	00H, 00H
	A	B	C	D	...	Z										
1AH, 00H	41H	42H	43H	44H		5AH										
00H, 00H																

驱动器No.

访问目标为MELSEC iQ-R系列模块的情况下，在下述表的范围内指定进行文件控制的驱动器。

驱动器No.	目标驱动器	驱动器No.	目标驱动器
0001H	软元件/标签存储器 ^{*1}	0004H	数据存储器
0002H	SD存储卡	—	
0003H	软元件/标签存储器 ^{*2}		

*1 是软元件/标签存储器的文件存储区。驱动器No. 中0001H被指定了的情况下，将变为至0003H 的访问。

*2 程序文件及参数文件等与模块控制相关的文件被存储到\$MELPRJ\$中。访问这些文件时请阅读下述中记载的注意事项内容。

☞ 144页 注意事项

访问目标为MELSEC-Q/L系列模块的情况下，在下述表的范围内指定进行文件控制的驱动器。

驱动器No.	目标驱动器	驱动器No.	目标驱动器
0000H	程序存储器	0003H	标准RAM
0001H	SRAM卡	0004H	标准ROM
0002H	Flash卡、ATA卡、SD存储卡	—	—

■以ASCII代码进行数据通信时

将驱动器No.按照从高位字节到低位字节的顺序发送。

例

驱动器No.为0003H的情况下

0	0	0	3
30 _H	30 _H	30 _H	33 _H

■以二进制代码进行数据通信时

将驱动器No.按照从低位字节到高位字节的顺序发送。

例

驱动器No.为0003H的情况下

03 _H	00 _H
-----------------	-----------------

文件名字符数、文件名

■子指令为0000、0004的情况下

文件名字符数将以16进制数指定通过“文件名”指定的文件名的字符数。字符数中也包括扩展名。

文件名在12字符(最大半角8字符*1+句号+扩展名3字符)以内进行指定。

文件名不仅使用半角字符(ASCII代码)，也可以使用全角字符(移位JIS汉字代码)。

*1 全角字符的情况下将变为最大4字符。

“文件名”在以二进制代码进行通信时也以ASCII代码指定。

以ASCII代码进行数据通信时	以二进制代码进行数据通信时												
“文件名”为“ABC.QPG”(文件名字符数为7字符)时的示例如下所示。 将文件名字符数按照从高位字节到低位字节的顺序发送。	“文件名”为“ABC.QPG”(文件名字符数为7字符)时的示例如下所示。 将文件名字符数按照从低位字节到高位字节的顺序发送。												
<table><tr><td>文件名字符数</td><td>文件名</td></tr><tr><td>0 0 0 7</td><td>A B C . Q P G</td></tr><tr><td>30H, 30H, 30H, 37H</td><td>41H, 42H, 43H, 2EH, 51H, 50H, 47H</td></tr></table>	文件名字符数	文件名	0 0 0 7	A B C . Q P G	30H, 30H, 30H, 37H	41H, 42H, 43H, 2EH, 51H, 50H, 47H	<table><tr><td>文件名字符数</td><td>文件名</td></tr><tr><td></td><td>A B C . Q P G</td></tr><tr><td>07H, 00H</td><td>41H, 42H, 43H, 2EH, 51H, 50H, 47H</td></tr></table>	文件名字符数	文件名		A B C . Q P G	07H, 00H	41H, 42H, 43H, 2EH, 51H, 50H, 47H
文件名字符数	文件名												
0 0 0 7	A B C . Q P G												
30H, 30H, 30H, 37H	41H, 42H, 43H, 2EH, 51H, 50H, 47H												
文件名字符数	文件名												
	A B C . Q P G												
07H, 00H	41H, 42H, 43H, 2EH, 51H, 50H, 47H												

■子指令为0040的情况下

文件名字符数将以16进制数指定从通过“文件名”指定的根文件夹开始到文件名为止的字符数。字符数中也包括扩展名。

对于文件名，将除去了起始的“驱动器名：\”的路径从根文件夹开始到文件名为止以UTF-16指定。

即使是添加了起始的“\”时的路径也可以指定。

文件名应在64字符(最大半角或全角60字符+句号+扩展名3字符)以内进行指定。

文件名与来自根文件夹的路径的字符数应在合计252字符以内进行指定。

要点

文件名中有不可以指定的字符。关于文件名中不可以指定的字符有关内容，请参阅所使用的模块手册。(所使用的模块手册)

以ASCII代码进行数据通信时		
“文件名”为“LINE.CSV”(文件名字符数为8字符)时的示例如下所示。 文件名将用UTF-16表示了“文件名”的值通过ASCII代码进行指定。		
文件名字符数	文件名	
0 0 0 8	(1)	
30H, 30H, 30H, 38H		
—	文件名	LINE.CSV
	UTF-16 (16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码 (16进制数)	3030344330303439303034453030343530303245303034333030353330303536

以二进制代码进行数据通信时		
“文件名”为“LINE.CSV”(文件名字符数为8字符)时的示例如下所示。 文件名以UTF-16表示了“文件名”的值，数据按照从低位字节到高位字节的顺序指定。		
文件名字符数	文件名	
	(1)	
08H, 00H		
—	文件名	LINE.CSV
	UTF-16 (16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码 (16进制数)	4C0049004E0045002E00430053005600

属性

指定文件属性。

文件属性有“读取专用”及“可读取、写入”这2种。

要点

- 现有文件的属性可以通过Read Directory/File(指令: 1810)确认。(145页 Read Directory/File(指令: 1810))
- 现有文件的属性可以通过Change File State(指令: 1825)更改。(167页 Change File State(指令: 1825))

文件指针No.

指定用于CPU模块管理文件的编号。

文件指针No.是在文件打开时获取的值，被存储到Open File(指令: 1827)的响应数据中。通过请求数据指定文件指针No.时，将Open File(指令: 1827)的响应数据中存储的值直接使用。

■以ASCII代码进行数据通信时

以ASCII代码4位，按照从高位字节到低位字节的顺序发送。

例

文件指针No.为AH的情况下

0	0	0	A
30 _H	30 _H	30 _H	41 _H

■以二进制代码进行数据通信时

以2字节的数值，按照从低位字节到高位字节的顺序发送。

例

文件指针No.为AH的情况下

0A _H	00 _H
-----------------	-----------------

注意事项

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。

通过SLMP可访问的文件类型，请参阅所使用的模块手册。

执行步骤

文件控制步骤如下所示。

读取文件内容的步骤

1. 文件有无的确认

使用Read Directory/File(指令: 1810)或Search Directory/File(指令: 1811), 确认文件的有无。

☞ 145页 Read Directory/File(指令: 1810)

☞ 154页 Search Directory/File(指令: 1811)

2. 文件打开

使用Open File(指令: 1827)后, 为了防止通过其它设备更改文件内容, 对文件加以锁定。(☞ 173页 Open File(指令: 1827))

3. 文件内容读取

使用Read File(指令: 1828), 读取文件内容。(☞ 176页 Read File(指令: 1828))

4. 文件关闭

使用Close File(指令: 182A), 解除文件的锁定。(☞ 182页 Close File(指令: 182A))

要点

外部设备侧已读取的文件的下述信息应预先记录。文件写入时等将变为需要。(仅访问目标为MELSEC-Q/L系列模块的情况下)

- 文件No.(通过Search Directory/File(指令: 1811)读取)
 - 文件名、属性、文件容量(通过Read Directory/File(指令: 1810)读取)
-

新建文件，写入数据的步骤

要点

在文件新建前，应确保对象存储器的空余区。对象存储器空余容量的确认与确保，通过工程工具进行。

■创建的文件为标题文文件(*.DAT)、顺控程序(*.PRG)、程序文件(*.QPG)、FB文件(*.PFB)、软元件注释文件(*.QCD、*.DCM)、软元件初始值文件(*.QDI、*.DID)的情况下

1. 文件有无的确认

使用Read Directory/File(指令: 1810)或Search Directory/File(指令: 1811)，确认文件的有无。

☞ 145页 Read Directory/File(指令: 1810)

☞ 154页 Search Directory/File(指令: 1811)

2. 文件名登录与容量确保

使用New File(指令: 1820)，新建文件。扩展名应置为DAT、PRG、QPG、PFB、QCD、DCM、QDI、DID以外。(☞ 157页 New File(指令: 1820))

3. 文件打开

使用Open File(指令: 1827)后，为了防止通过其它设备更改文件内容，对文件加以锁定。(☞ 173页 Open File(指令: 1827))

4. 文件内容写入

使用Write File(指令: 1829)，写入文件内容。(☞ 179页 Write File(指令: 1829))

5. 文件关闭

使用Close File(指令: 182A)，解除文件的锁定。(☞ 182页 Close File(指令: 182A))

6. 文件复制

使用Copy File(指令: 1824)，复制DAT、PRG、QPG、PFB、QCD、DCM、QDI、DID文件。在复制后，根据需要删除复制源的文件。(☞ 163页 Copy File(指令: 1824))

■创建的文件为标题文文件(*.DAT)、顺控程序(*.PRG)、程序文件(*.QPG)、FB文件(*.PFB)、软元件注释文件(*.QCD、*.DCM)、软元件初始值文件(*.QDI、*.DID)以外的情况下

1. 文件有无的确认

使用Read Directory/File(指令: 1810)或Search Directory/File(指令: 1811)，确认文件的有无。

☞ 145页 Read Directory/File(指令: 1810)

☞ 154页 Search Directory/File(指令: 1811)

2. 文件名登录与容量确保

使用New File(指令: 1820)，新建文件。(☞ 157页 New File(指令: 1820))

3. 文件打开

使用Open File(指令: 1827)后，为了防止通过其它设备更改文件内容，对文件加以锁定。(☞ 173页 Open File(指令: 1827))

4. 文件内容写入

使用Write File(指令: 1829)，写入文件内容。(☞ 179页 Write File(指令: 1829))

5. 文件关闭

使用Close File(指令: 182A)，解除文件的锁定。(☞ 182页 Close File(指令: 182A))

6. 文件No.确认^{*1}

使用Search Directory/File(指令: 1811)，记录文件No.。Read Directory/File(指令: 1810)使用时将变为需要。(☞ 154页 Search Directory/File(指令: 1811))

*1 仅访问目标为MELSEC-Q/L系列模块的情况下需要。

复制文件步骤

要点

在文件复制前，应确保对象存储器的空余区。对象存储器空余容量的确认与确保，通过工程工具进行。

1. 文件有无的确认

使用Read Directory/File(指令: 1810)或Search Directory/File(指令: 1811)，确认文件的有无。

☞ 145页 Read Directory/File(指令: 1810)

☞ 154页 Search Directory/File(指令: 1811)

2. 文件复制

使用Copy File(指令: 1824)，复制文件。(☞ 163页 Copy File(指令: 1824))

3. 文件No.确认^{*1}

因复制新建了文件的情况下，使用Search Directory/File(指令: 1811)，记录文件No.。使用Read Directory/File(指令: 1810)时将变为需要。(☞ 154页 Search Directory/File(指令: 1811))

^{*1} 仅访问目标为MELSEC-Q/L系列模块的情况下需要。

将数据覆盖到现有文件的步骤

要点

- 在文件覆盖前，应确保对象存储器的空余区。对象存储器空余容量的确认与确保，通过工程工具进行。
- 现有文件与进行覆盖的文件的数据容量不同的情况下及，PRG、PFB文件的情况下应通过Delete File(指令: 1822)删除覆盖对象的文件后，通过下述页中记载的“新建文件，写入数据的步骤”进行覆盖。

☞ 141页 新建文件，写入数据的步骤

1. 文件有无的确认

使用Read Directory/File(指令: 1810)或Search Directory/File(指令: 1811)，确认文件的有无。

☞ 145页 Read Directory/File(指令: 1810)

☞ 154页 Search Directory/File(指令: 1811)

2. 文件打开

使用Open File(指令: 1827)后，为了防止通过其它设备更改文件内容，对文件加以锁定。(☞ 173页 Open File(指令: 1827))

3. 文件内容写入

使用Write File(指令: 1829)，写入文件内容。(☞ 179页 Write File(指令: 1829))

4. 文件关闭

使用Close File(指令: 182A)，解除文件的锁定。(☞ 182页 Close File(指令: 182A))

更改文件创建日期的步骤

执行Change File Date(指令: 1826), 更改文件的创建日期。无需通过Open File(指令: 1827)对文件加以锁定。

删除文件步骤

1. 文件有无的确认

使用Read Directory/File(指令: 1810)或Search Directory/File(指令: 1811), 确认文件的有无。

☞ 145页 Read Directory/File(指令: 1810)

☞ 154页 Search Directory/File(指令: 1811)

2. 文件删除

使用Delete File(指令: 1822), 删除文件。(☞ 160页 Delete File(指令: 1822))

要点

如果在可编程控制器系统动作中删除文件, 系统有可能停止。删除文件的时机应取决于可编程控制器系统整体。

注意事项

进行文件控制时的注意事项如下所示。

关于已读取的文件

读取了程序文件及参数文件等与模块控制相关的文件的情况下，文件是外部设备侧中的备份用。已读取的文件内容请勿在外部设备侧进行编辑。
此外，应将“\$MELPRJ\$”文件夹的数据备份，在恢复时对于“\$MELPRJ\$”内的全部文件执行。
在已读取的文件中，仅将一部分文件恢复到\$MELPRJ\$文件夹内的情况下，有可能无法正常动作。

添加保护的情况下

执行下述指令的情况下，应预先解除访问目标的保护(CPU模块的系统保护、SD存储卡的保护开关的LOCK)。如果在添加保护的
状态下执行指令，指令将异常完成。

指令	浏览目标
New File	157页 New File(指令: 1820)
Delete File	160页 Delete File(指令: 1822)
Copy File	163页 Copy File(指令: 1824)
Change File State	167页 Change File State(指令: 1825)
Change File Date	170页 Change File Date(指令: 1826)
Write File	179页 Write File(指令: 1829)

Read Directory/File(指令: 1810)

读取文件一览信息。

请求数据

■子指令为0000的情况下

ASCII

1 8 1 0	子指令	0 0 0 0	驱动器No.	起始文件No.	文件请求数	0 0 0 0
31H, 38H, 31H, 30H		30H, 30H, 30H, 30H				30H, 30H, 30H, 30H

二进制

子指令	驱动器No.	起始文件No.	文件请求数
10H, 18H	30H, 30H, 30H, 30H		00H, 00H

■子指令为0040的情况下

ASCII

1 8 1 0	子指令	0 0 0 0	驱动器No.	起始文件No.	文件请求数	目录的路径名字符数	目录的路径名
31H, 38H, 31H, 30H		30H, 30H, 30H, 30H					

二进制

子指令	驱动器No.	起始文件No.	文件请求数	目录的路径名字符数	目录的路径名
10H, 18H	00H, 00H, 00H, 00H				

■子指令

子指令											
ASCII代码	二进制代码										
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td></td></tr><tr><td>00H, 00H</td></tr></table>		00H, 00H
0	0	0	0								
30H	30H	30H	30H								
00H, 00H											
<table><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table>	0	0	4	0	30H	30H	34H	30H	<table><tr><td></td></tr><tr><td>40H, 00H</td></tr></table>		40H, 00H
0	0	4	0								
30H	30H	34H	30H								
40H, 00H											

■驱动器No.

指定读取文件一览信息的驱动器。(137页 驱动器No.)

■起始文件No.

指定模块中写入的文件的登录No.。(指定范围: 1H~)

以ASCII代码进行数据通信时, 将文件No.转换为ASCII代码(8位或4位)使用后, 按照从高位字节到低位字节的顺序被存储。

子指令为0040与0000中, 转换为ASCII代码的位数有所不同。

子指令	位数	例																
0040	转换为ASCII代码8位	1FH的情况下(8位) <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>F</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>30H</td><td>31H</td><td>46H</td></tr></table>	0	0	0	0	0	0	1	F	30H	30H	30H	30H	30H	30H	31H	46H
0	0	0	0	0	0	1	F											
30H	30H	30H	30H	30H	30H	31H	46H											
0000	转换为ASCII代码4位	1FH的情况下(4位) <table><tr><td>0</td><td>0</td><td>1</td><td>F</td></tr><tr><td>30H</td><td>30H</td><td>31H</td><td>46H</td></tr></table>	0	0	1	F	30H	30H	31H	46H								
0	0	1	F															
30H	30H	31H	46H															

以二进制代码进行数据通信时, 对文件No.使用数值(4字节或2字节)按照从低位字节到高位字节的顺序发送。

子指令为0040、0000中，数值的数据容量有所不同。

子指令	数据容量	例
0040	4字节	输入(X)的情况下(4字节) 1FH, 00H, 00H, 00H
0000	2字节	输入(X)的情况下(2字节) 1FH, 00H

要点

模块中存储的文件的文件No.可以通过Search Directory/File(指令: 1811)确认。(154页 Search Directory/File(指令: 1811))

■文件请求数

指定读取文件信息时的读取文件数。

子指令	指定范围
0000	1~36
0040	1~36

数据的发送顺序与“起始文件No.”相同。

■目录路径名字符数

以16进制数指定目录路径名的字符数。将字符数指定为“0”的情况下，表示根目录。

以ASCII代码进行数据通信时，将目录路径名字符数转换为ASCII代码4位使用后，按照从高位字节到低位字节的顺序发送。

例

目录路径名字符数为86字符(56H)的情况下

0 0 5 6
30H, 30H, 35H, 36H

以二进制代码进行数据通信时，对目录路径名字符数使用2字节的数值，按照从低位字节到高位字节的顺序发送。

例

目录路径名字符数为86字符(56H)的情况下

56H, 00H

■目录路径名

以UTF-16指定来自根目录的路径名。

- 以ASCII代码进行数据通信时，以UTF-16表示了根目录的路径名的数值以ASCII代码指定。按照从高位字节到低位字节的顺序发送。
- 以二进制代码进行数据通信时，通过以UTF-16表示了根目录的路径名的数值指定。按照从低位字节到高位字节的顺序发送。

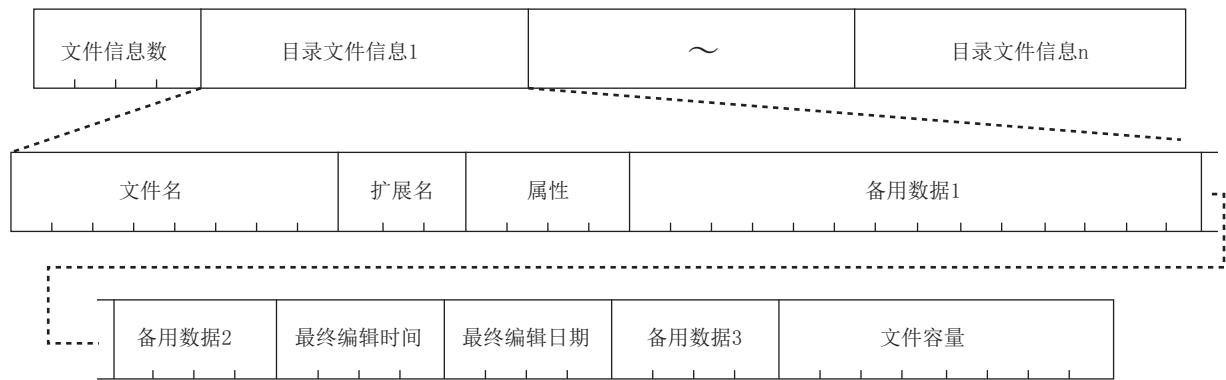
目录路径名为“SUBDIR”时的示例如下所示。

路径名(UTF-16 (16进制数))	S(0053)	U(0055)	B(0042)	D(0044)	I(0049)	R(0052)
ASCII代码(16进制数)	30303533	30303535	30303432	30303434	30303439	30303532
二进制代码(16进制数)	5300	5500	4200	4400	4900	5200

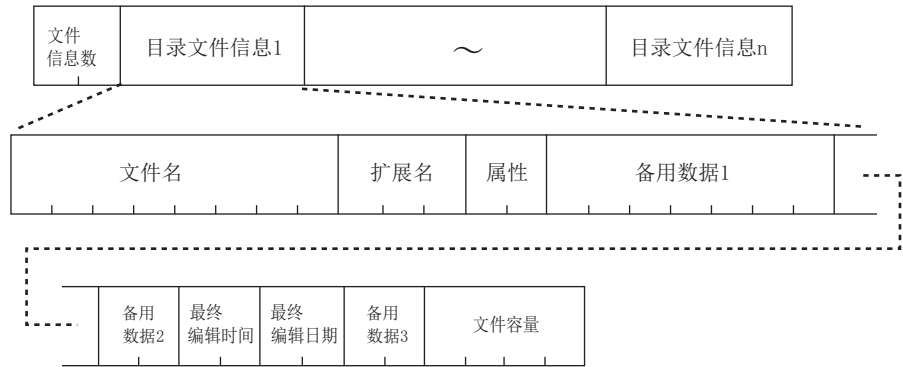
响应数据

■子指令为0000的情况下

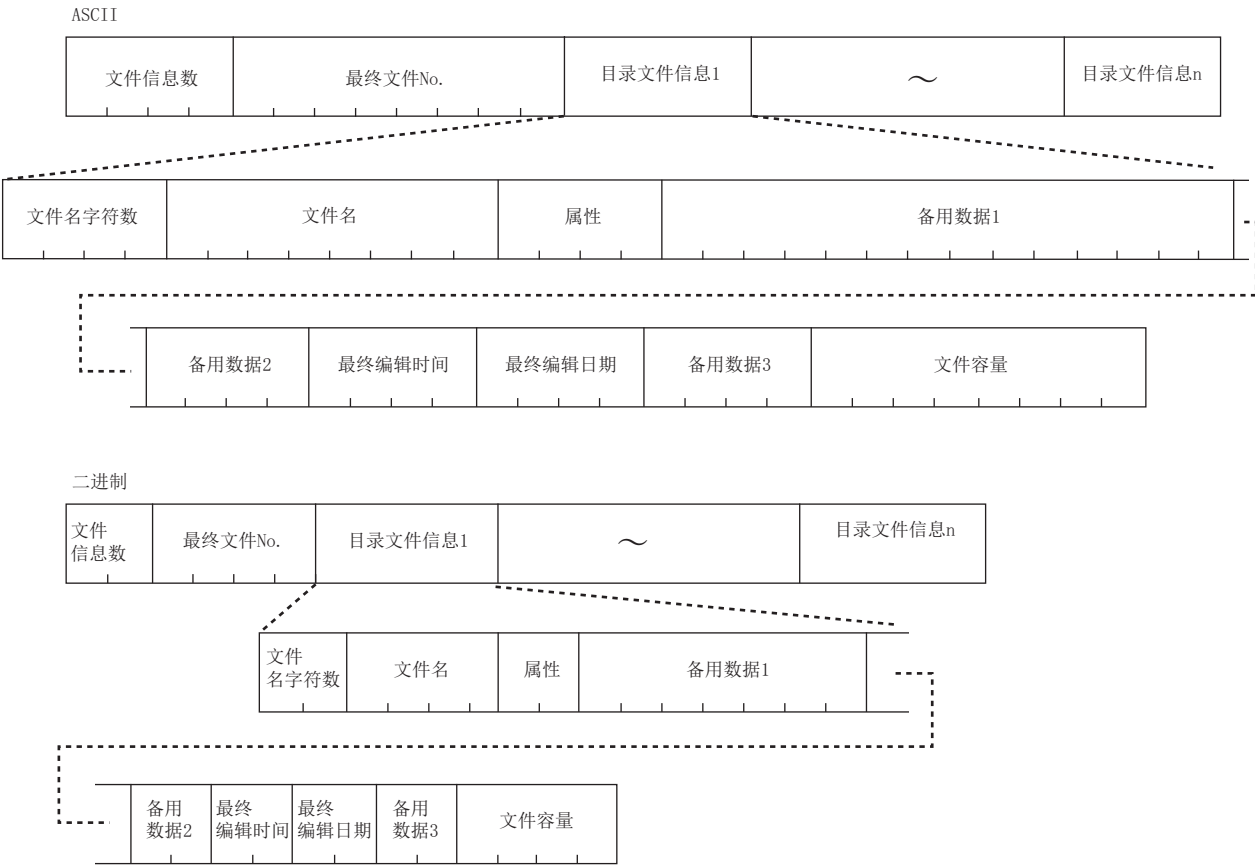
ASCII



二进制



■子指令为0040的情况下



■文件信息数

存储响应数据中存储的文件信息的数。数据的存储顺序与“起始文件No.”相同。
请求数据中指定的“起始文件No.”以后无文件的情况下，存储0。
由于文件名的长度，有可能变为请求的文件的数以下。

■最终文件No.

文件信息的读取已完成的文件(响应数据中也包括无法删除的文件)的文件No.被存储。
在通过1次请求未完成文件信息读取的情况下使用。(☞ 150页 子指令0040 中的目录文件信息的读取步骤)

■文件名、扩展名

存储目录、文件名及扩展名。文件名与扩展名在以二进制代码进行通信时也以ASCII代码被存储。
文件名未满8字符的情况下，在文件后面空格(代码：20H)被添加。
存储目录名的情况下，扩展名存储空格。
本删除的目录名/文件名不可以存储。

■文件名字符数、文件名

存储文件名字符数及文件名。
但是，本删除的目录名/文件名不可以存储。

■属性

- 存储文件属性。
- 读取专用目录：31H、11H
 - 可读取、写入目录：30H、10H
 - 读取专用文件：01H、21H
 - 可读取、写入文件：00H、20H

对属性存储了上述以外值的目录/文件，由于为系统用，因此请勿访问。

■备用数据

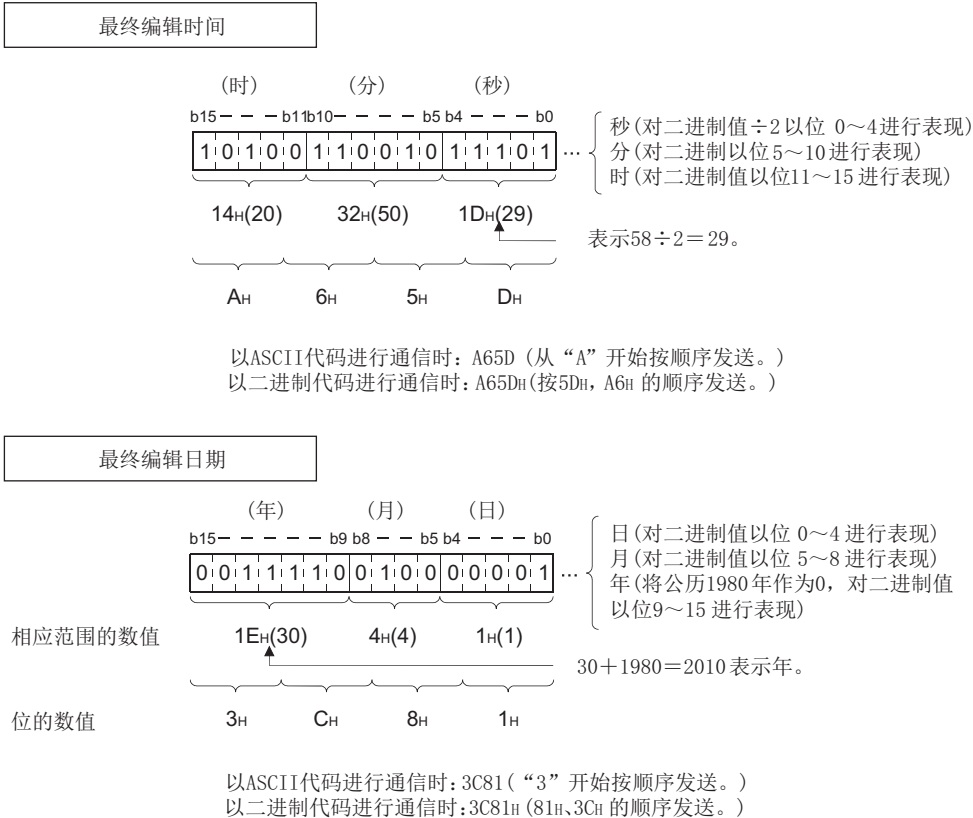
由于存储任意值，因此请勿使用。
备用数据1: ASCII代码时14位。二进制代码时7字节。
备用数据2及备用数据3: ASCII代码时4位。二进制代码时2字节。

■最终编辑时间、最终编辑日期

存储文件的最终编辑时间与日期。
目录的情况下，存储创建了时的时间与日期。

例

“最终编辑时间”为20时50分58秒，“最终编辑日期”为2010年4月1日的情况下



■文件容量

文件容量以字节单位被存储。
目录的情况下，数据容量将变为0。

例

文件容量为7168字节的情况下

ASCII代码	二进制代码
文件容量以ASCII代码8位，按照从高位字节到低位字节的顺序被存储。(16进制数)	文件容量按照从低位字节到高位字节的顺序被存储。(16进制数)
0 0 0 0 1 C 0 0 30H, 30H, 30H, 30H, 31H, 43H, 30H, 30H	00H, 1CH, 00H, 00H

■子指令0000 中的目录文件信息的读取步骤

获取子指令0000 中的目录内的全部文件信息的情况下，多次重复执行本指令。

- 1. 对起始文件No.指定1后，执行本指令。
- 2. 将上次指定的起始文件No.中加上文件请求数计算的值指定到起始文件No.后，执行本指令。
- 3. 文件信息数比文件请求数少之前，重复步骤2。

要点

获取目录内的全部文件信息的过程中，进行了来自其它文件操作的情况下，有可能无法获取正确信息。获取文件信息的过程中，请勿进行来自其它文件操作。

■子指令0040 中的目录文件信息的读取步骤

获取子指令0040 中的目录内的全部文件信息的情况下，多次重复执行本指令。

- 1. 对起始文件No.指定1后，执行本指令。
- 2. 将响应数据的最终文件No.中加上1计算的值指定到起始文件No.后，执行本指令。
- 3. 读取信息数变为-1(0FFFFH)为止，重复步骤2。

要点

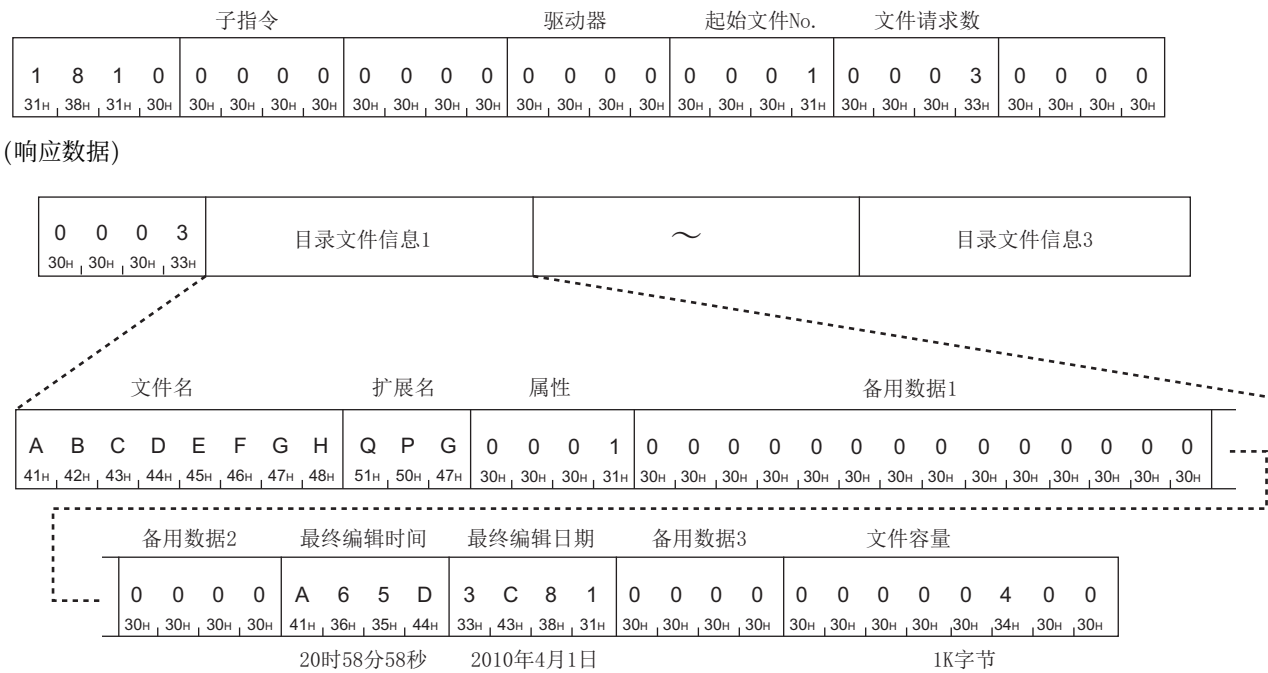
获取目录内的全部文件信息的过程中，进行了来自其它文件操作的情况下，有可能无法获取正确信息。获取文件信息的过程中，请勿进行来自其它文件操作。

通信示例(子指令为0000的情况下)

在下述条件下，读取QCPU的目录·文件信息。

- 驱动器No.: 0
- 起始驱动器No.: 1
- 文件请求数: 3

■以ASCII代码进行数据通信时
(请求数据)



■以二进制代码进行数据通信时

(请求数据)

子指令				驱动器 No.	起始 文件No.	文件 请求数
10 _H , 18 _H	00 _H , 00 _H	30 _H , 30 _H , 30 _H , 30 _H	00 _H , 00 _H	01 _H , 00 _H	03 _H , 00 _H	00 _H , 00 _H

(响应数据)

文件 信息数	目录文件信息1								目录文件信息3											
03H, 00H									~											
文件名								扩展名			属性		备用数据1							
A	B	C	D	E	F	G	H	Q	P	G										
41H	42H	43H	44H	45H	46H	47H	48H	51H	50H	47H	01H	00H	00H	00H	00H	00H	00H	00H	00H	00H
备用数据2				最终 编辑时间				最终 编辑日期				备用数据3				文件容量				
00H, 00H				5DH, A6H				81H, 3CH				00H, 00H				00H, 04H, 00H, 00H				

通信示例(子指令为0040的情况下)

在下述条件下，读取RCPU的目录・文件信息。

- 驱动器No.: 4
- 起始驱动器No.: 1
- 文件请求数: 3
- 目录路径名: SUBDIR

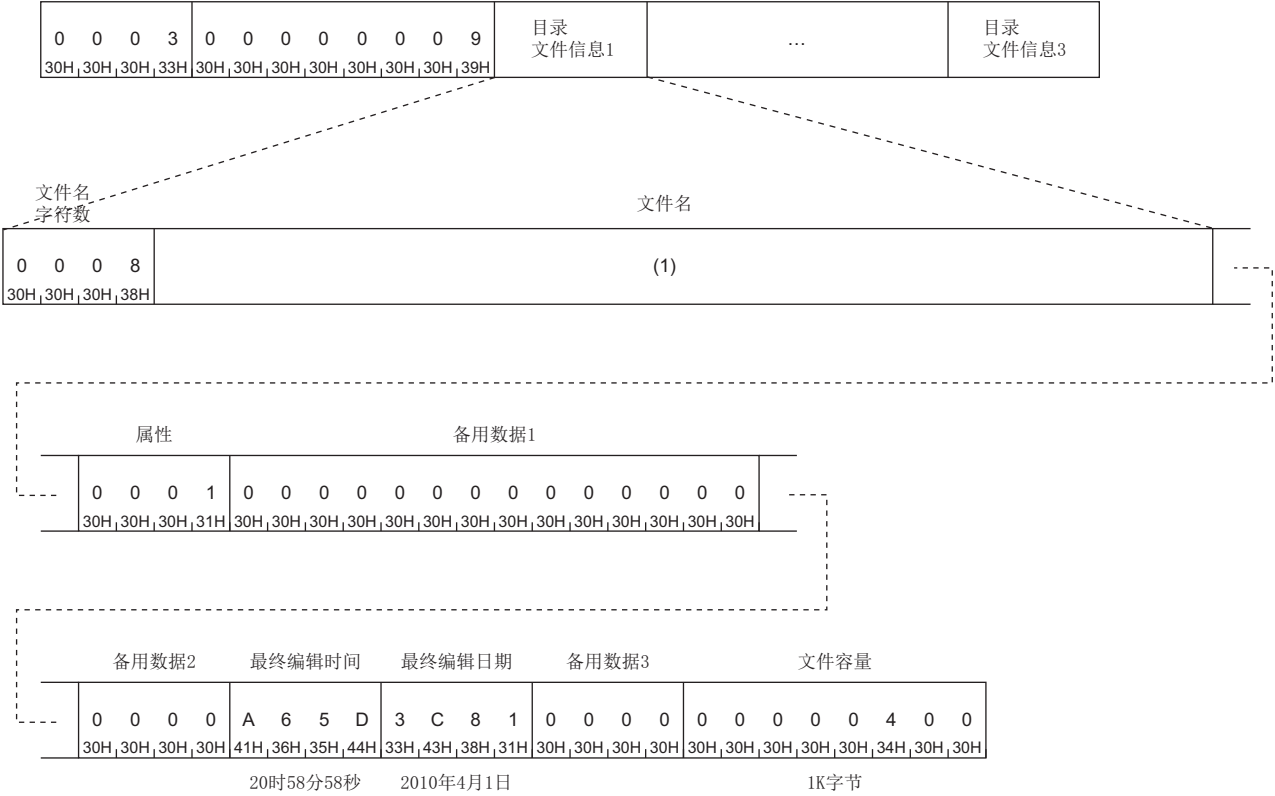
■以ASCII代码进行数据通信时

(请求数据)

子指令				驱动器No.				起始文件No.								文件请求数					
1	8	1	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	3	---
31H,38H,31H,30H	30H,30H,34H,30H	30H,30H,30H,30H	30H,30H,30H,34H	30H,30H,30H,30H	30H,30H,30H,30H	30H,30H,30H,31H	30H,30H,30H,33H														
目录的路径名字符数				目录的路径名																	
0 0 0 6				(1)																	
30H,30H,30H,36H																					

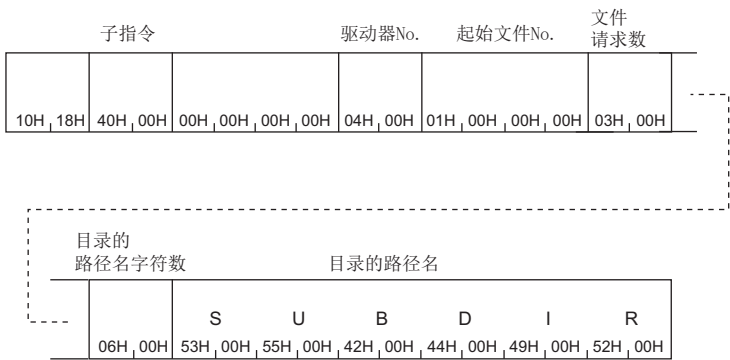
编号	项目	值
—	目录路径名	SUBDIR
	UTF-16(16进制数)	005300550042004400490052
(1)	ASCII代码(16进制数)	3030353330303535303030343230303434303030343930303532

(响应数据)

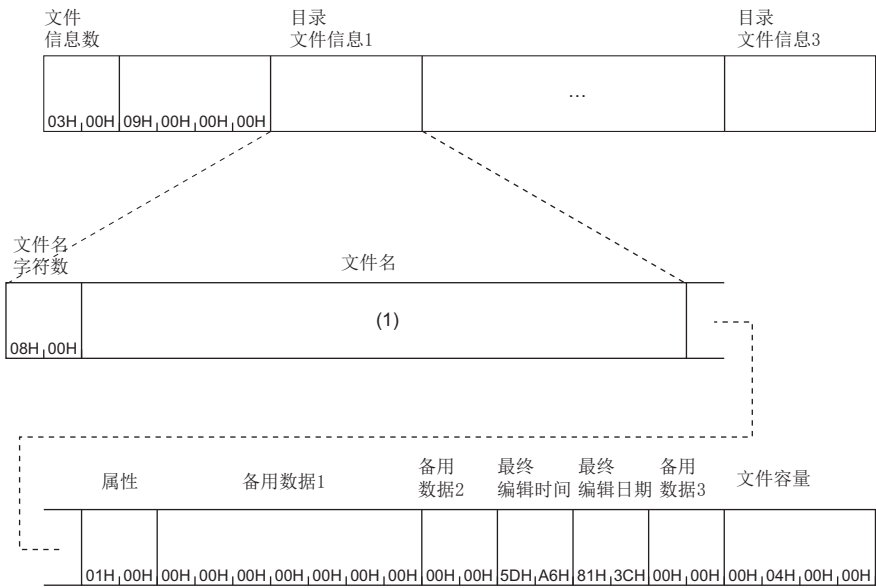


编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	3030344330303439303034453030343530303245303034333030353330303536

■以二进制代码进行数据通信时
(请求数据)



(响应数据)



编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

Search Directory/File(指令: 1811)

读取指定文件的文件No.。文件No.是指模块中写入文件时的文件登录编号。

请求数据

ASCII

1 8 1 1	子指令	口令	驱动器No.	0 0 0 0	文件名字符数	文件名
31H, 38H, 31H, 31H				30H, 30H, 30H, 30H		

二进制

	子指令	口令	驱动器No.		文件名字符数	文件名
11H, 18H				00H, 00H		

■子指令

子指令	
ASCII代码	二进制代码
0000 30H, 30H, 30H, 30H	00H, 00H
0040 30H, 30H, 34H, 30H	40H, 00H

■口令

- 访问目标为MELSEC-Q/L系列模块的情况下，指定文件口令。(136页 口令)
- 访问目标为MELSEC iQ-R系列模块的情况下固定为“0”。ASCII代码的情况下，通过“30H”(0)指定。

■驱动器No.

指定读取文件No.的驱动器。(137页 驱动器No.)

■文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

■文件名

指定读取文件No.的文件名。文件名以带扩展名指定。(138页 文件名字符数、文件名)

响应数据

文件No. 被存储。

以ASCII代码进行数据通信时，对文件No. 以ASCII代码(8位或4位)按照从高位字节到低位字节的顺序被存储。(16进制数)
子指令为0040与0000中，被存储的ASCII代码的位数有所不同。

子指令	位数	例
0040	ASCII代码8位	文件No. 为AH的情况下(8位) 0 0 0 0 0 0 0 A 30H, 30H, 30H, 30H, 30H, 30H, 30H, 41H
0000	ASCII代码4位	文件No. 为AH的情况下(4位) 0 0 0 A 30H, 30H, 30H, 41H

以二进制代码进行数据通信时，对文件No. 以数值(4字节或2字节)按照从低位字节到高位字节的顺序被存储。
子指令为0040、0000中，被存储的数值的数据容量有所不同。

子指令	数据容量	例
0040	4字节	文件No. 为AH的情况下(4字节) 0AH, 00H, 00H, 00H
0000	2字节	文件No. 为AH的情况下(2字节) 0AH, 00H

通信示例(子指令为0000的情况下)

下述条件中的QCPU的目录·文件信息查找示例如下所示。

- 口令：空格4字符(代码：20H)
- 驱动器No.：0
- 文件名：ABC.QPG(文件No.6)

■以ASCII代码进行数据通信时 (请求数据)

子指令	口令	驱动器No.	文件名字符数	文件名
1 8 1 1 31H, 38H, 31H, 31H	0 0 0 0 30H, 30H, 30H, 30H	20H, 20H, 20H, 20H	0 0 0 0 30H, 30H, 30H, 30H	0 0 0 7 30H, 30H, 30H, 37H
				A B C . Q P G 41H, 42H, 43H, 2EH, 51H, 50H, 47H

(响应数据)

文件No.
0 0 0 6 30H, 30H, 30H, 36H

■以二进制代码进行数据通信时
(请求数据)

子指令		指令		驱动器No.		文件名字符数	文件名						
11 _H , 18 _H	00 _H , 00 _H	20 _H , 20 _H , 20 _H , 20 _H	00 _H , 00 _H	00 _H , 00 _H	07 _H , 00 _H		A	B	C	.	Q	P	G
							41 _H	42 _H	43 _H	2E _H	51 _H	50 _H	47 _H

(响应数据)

文件No.
06H, 00H

通信示例(子指令为0040的情况下)

下述条件中的RCPU的目录·文件信息查找示例如下所示。

- 驱动器No.: 4
- 文件名: LINE.CSV(8字符)的情况下(文件No.6)

■以ASCII代码进行数据通信时
(请求数据)

子指令		驱动器No.		文件名字符数		文件名
1 8 1 1	0 0 4 0	0 0 0 0	0 0 0 4	0 0 0 0	0 0 0 8	(1)
31H, 38H, 31H, 31H	30H, 30H, 34H, 30H	30H, 30H, 30H, 30H	30H, 30H, 30H, 34H	30H, 30H, 30H, 30H	30H, 30H, 30H, 38H	

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	3030344330303439303034453030343530303245303034333030353330303536

(响应数据)

文件No.
0 0 0 0 0 0 0 6
30H, 30H, 30H, 30H, 30H, 30H, 30H, 36H

■以二进制代码进行数据通信时
(请求数据)

子指令		驱动器No.		文件 名字符数	文件名
11H, 18H	40H, 00H	00H, 00H, 00H, 00H	04H, 00H	00H, 00H	08H, 00H
11H, 18H	40H, 00H	00H, 00H, 00H, 00H	04H, 00H	00H, 00H	08H, 00H

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

(响应数据)

文件No.
06H, 00H, 00H, 00H

New File(指令：1820)

指定容量，新建文件。

请求数据

ASCII

1 8 2 0 31H, 38H, 32H, 30H	子指令	口令	驱动器No.	文件容量	文件名字符数	文件名
-------------------------------	-----	----	--------	------	--------	-----

二进制

20H, 18H	子指令	口令	驱动器No.	文件容量	文件名字符数	文件名
----------	-----	----	--------	------	--------	-----

子指令

子指令	
ASCII代码	二进制代码
0000 30H, 30H, 30H, 30H	00H, 00H
0040 30H, 30H, 34H, 30H	40H, 00H

口令

- 访问目标为MELSEC-Q/L系列模块的情况下指定文件口令。(136页 口令)
- 访问目标为MELSEC iQ-R系列模块的情况下固定为“0”。ASCII代码的情况下，通过“30H”(0)指定。

驱动器No.

指定新建除文件的驱动器。(137页 驱动器No.)

文件容量

以字节单位指定文件容量。

例

文件容量为7168字节的情况下

ASCII代码	二进制代码																				
转换为ASCII代码8位后，按照从高位字节到低位字节的顺序指定。(16进制数)	按照从低位字节到高位字节的顺序指定。(16进制数)																				
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>C</td><td>0</td><td>0</td></tr><tr><td>30_H</td><td>30_H</td><td>30_H</td><td>30_H</td><td>31_H</td><td>43_H</td><td>30_H</td><td>30_H</td></tr></table>	0	0	0	0	1	C	0	0	30 _H	30 _H	30 _H	30 _H	31 _H	43 _H	30 _H	30 _H	<table><tr><td>00_H</td><td>1C_H</td><td>00_H</td><td>00_H</td></tr></table>	00 _H	1C _H	00 _H	00 _H
0	0	0	0	1	C	0	0														
30 _H	30 _H	30 _H	30 _H	31 _H	43 _H	30 _H	30 _H														
00 _H	1C _H	00 _H	00 _H																		

文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

文件名

指定新建文件名。(138页 文件名字符数、文件名)

注意事项

文件新建应按照下述页中记载的步骤进行。

- 141页 新建文件，写入数据的步骤

通过本指令新建的文件将作为最终编辑日期时间登录模块时间。

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。

通过SLMP可访问的文件类型，请参阅所使用的模块手册。

响应数据

无New File指令的响应数据。

通信示例(子指令为0000的情况下)

在QCPU中新建下述文件。

- 口令：空格4字符(代码：20H)
- 驱动器No.：0
- 文件名：ABC.CSV
- 文件容量：1K字节

■以ASCII代码进行数据通信时
(请求数据)

子指令				口令				驱动器No.				文件容量								文件名字符数			
1	8	2	0	0	0	0	0					0	0	0	0	4	0	0	0	0	0	0	7
31H	38H	32H	30H	30H	30H	30H	30H	20H	20H	20H	20H	30H	30H	30H	30H	34H	30H	30H	30H	30H	30H	30H	37H

文件名

A	B	C	.	C	S	V
41H	42H	43H	2EH	43H	53H	56H

■以二进制代码进行数据通信时
(请求数据)

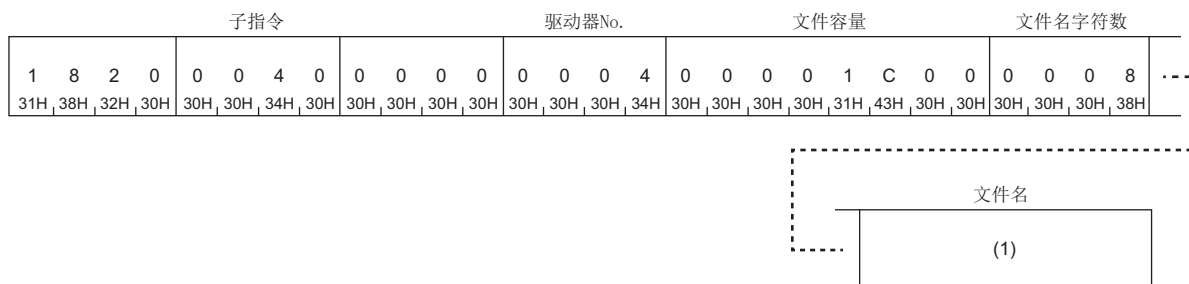
子指令		口令		驱动器No.		文件容量		文件名字符数		文件名					
20 _H , 18 _H	00 _H , 00 _H	20 _H , 20 _H , 20 _H , 20 _H	00 _H , 00 _H	00 _H , 40 _H , 00 _H , 00 _H	07 _H , 00 _H	A	B	C	.	C	S	V			
						41 _H , 42 _H , 43 _H	2E _H	43 _H , 53 _H , 56 _H							

通信示例(子指令为0040的情况下)

在RCPU中新建下述文件。

- 驱动器No.: 4
- 文件名: LINE.CSV(8字符的情况下)
- 文件容量: 7168字节

■以ASCII代码进行数据通信时 (请求数据)



编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	3030344330303439303034453030343530303245303034333030353330303536

■以二进制代码进行数据通信时 (请求数据)

子指令	驱动器No.	文件容量	文件名字符数	文件名
20H, 18H	40H, 00H	00H, 00H, 00H, 00H	04H, 00H	00H, 1CH, 00H, 00H
(1)				

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

Delete File(指令：1822)

删除文件。

请求数据

■子指令为0000、0004的情况下

ASCII

1	8	2	2	子指令	口令	驱动器No.	文件名字符数	文件名
31H	38H	32H	32H					

二进制

	子指令	口令	驱动器No.	文件名字符数	文件名
22H, 18H					

■子指令为0040的情况下

ASCII

1	8	2	2	子指令	口令字符数	口令	驱动器No.	文件名字符数	文件名
31H	38H	32H	32H						

二进制

	子指令	口令字符数	口令	驱动器No.	文件名字符数	文件名
22H, 18H						

■子指令

子指令											
ASCII代码	二进制代码										
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td></td></tr><tr><td>00H, 00H</td></tr></table>		00H, 00H
0	0	0	0								
30H	30H	30H	30H								
00H, 00H											
<table><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>34H</td></tr></table>	0	0	0	4	30H	30H	30H	34H	<table><tr><td></td></tr><tr><td>04H, 00H</td></tr></table>		04H, 00H
0	0	0	4								
30H	30H	30H	34H								
04H, 00H											
<table><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table>	0	0	4	0	30H	30H	34H	30H	<table><tr><td></td></tr><tr><td>40H, 00H</td></tr></table>		40H, 00H
0	0	4	0								
30H	30H	34H	30H								
40H, 00H											

■口令

指定复制目标的文件口令。(136页 口令)

■驱动器No.

指定删除文件的驱动器。(137页 驱动器No.)

■文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

■文件名

指定进行删除的文件名。(138页 文件名字符数、文件名)

响应数据

无Delete File指令的响应数据。

注意事项

- 如果在可编程控制器系统动作中删除文件，系统有可能停止。删除文件的时机应取决于可编程控制器系统整体。
- 通过Open File(指令：1827)被文件锁定的文件不可以删除。应通过Close File(指令：182A)解除文件锁定之后，再执行本指令。
- MELSEC-Q/L系列CPU模块为RUN状态时，不可以删除程序文件、参数文件及引导文件。应将CPU模块置为STOP状态之后，再进行文件删除。
- 在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。通过SLMP可访问的文件类型，请参阅所使用的模块手册。

通信示例(子指令为0000的情况下)

删除QCPU文件。

删除对象的文件将作为下述条件。

- 口令：1234
- 驱动器No.：0
- 删除文件名：ABC.QPG

■以ASCII代码进行数据通信时

(请求数据)

子指令				口令				驱动器No.				文件名字符数				文件名										
1	8	2	2	0	0	0	0	1	2	3	4	0	0	0	0	0	0	0	7	A	B	C	.	Q	P	G
31 _H	38 _H	32 _H	32 _H	30 _H	30 _H	30 _H	30 _H	31 _H	32 _H	33 _H	34 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	37 _H	41 _H	42 _H	43 _H	2E _H	51 _H	50 _H	47 _H

■以二进制代码进行数据通信时

(请求数据)

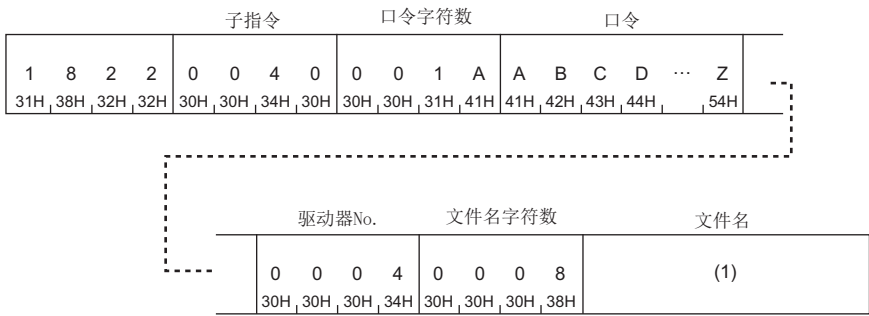
文件																													
子指令		口令				驱动器No.				名字符数								文件名											
		1 2 3 4								A B C . Q P G																			
22 _H , 18 _H		00 _H , 00 _H				31 _H , 32 _H , 33 _H , 34 _H				00 _H , 00 _H				07 _H , 00 _H								41 _H , 42 _H , 43 _H , 2E _H , 51 _H , 50 _H , 47 _H							

通信示例(子指令为0040的情况下)

删除RCPU 文件。
删除对象的文件将作为下述条件。

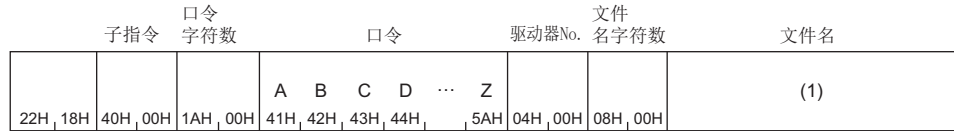
- 口令：A~Z(26字符)
- 驱动器No.：4
- 删除文件名：LINE.CSV(8字符)的情况下

■以ASCII代码进行数据通信时
(请求数据)



编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	303034433030343930303034453030343530303245303034333030353330303536

■以二进制代码进行数据通信时
(请求数据)



编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

Copy File(指令: 1824)

复制指定文件。

请求数据

■子指令为0000、0004的情况下

ASCII

1	8	2	4	子指令	固定数据(16字符)				复制目标口令	复制目标驱动器	复制目标文件名字符数	复制目标文件名
31H	38H	32H	34H		30H	30H	30H	~	30H	30H	30H	

■子指令

子指令											
ASCII代码	二进制代码										
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td>00H</td><td>00H</td></tr></table>	00H	00H
0	0	0	0								
30H	30H	30H	30H								
00H	00H										
<table><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>34H</td></tr></table>	0	0	0	4	30H	30H	30H	34H	<table><tr><td>04H</td><td>00H</td></tr></table>	04H	00H
0	0	0	4								
30H	30H	30H	34H								
04H	00H										
<table><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table>	0	0	4	0	30H	30H	34H	30H	<table><tr><td>40H</td><td>00H</td></tr></table>	40H	00H
0	0	4	0								
30H	30H	34H	30H								
40H	00H										

■固定数据(16字符)

指定“0”。ASCII代码的情况下，通过“30H”(0)指定。

■复制目标口令、复制源口令

指定复制目标的文件口令。(136页 口令)

■复制目标驱动器No.、复制源驱动器No.

指定进行复制的驱动器。(137页 驱动器No.)

要点

RCPU为复制源或复制目标的情况下，驱动器No0(程序存储器)不可以指定。

■复制目标文件名字符数、复制源文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

■复制目标文件名、复制源文件名

指定进行复制的文件名。(138页 文件名字符数、文件名)

响应数据

无Copy File指令的响应数据。

注意事项

应将MELSEC-Q/L系列CPU模块置为STOP状态之后再复制下述文件。在RUN状态下进行复制时将变为出错。

- 参数文件
- 程序存储器(驱动器No.: 0000H)的执行中文件

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。通过SLMP可访问的文件类型，请参阅所使用的模块手册。

通信示例(子指令为0000的情况下)

复制QCPU文件。

下述条件中的文件复制示例如下所示。

- 复制目标口令、复制源口令：1234
- 复制源驱动器No.：0
- 复制目标驱动器No.：1
- 复制源文件名：ABC.QPG
- 复制目标文件名：CBA.QPG

■以ASCII代码进行数据通信时

(请求数据)

子指令								固定数据(16字符)																复制目标口令							
1	8	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	..		
31 _H	38 _H	32 _H	34 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	31 _H	32 _H	33 _H	34 _H	..		

■以二进制代码进行数据通信时

(请求数据)

子指令		固定数据(8字节)								复制目标口令				复制目标驱动器No.	复制目标文件名字数	复制目标文件名							
										1	2	3	4			C	B	A	.	Q	P	G	.
24 _H , 18 _H		00 _H , 00 _H		00 _H , 00 _H		00 _H , 00 _H		00 _H , 00 _H		00 _H , 00 _H		00 _H , 00 _H		31 _H , 32 _H	33 _H , 34 _H	00 _H , 00 _H		07 _H , 00 _H		43 _H , 42 _H	41 _H , 2E _H	51 _H , 50 _H	47 _H

复制源口令				复制源驱动器No.	复制源文件名字数	复制源文件名							
						A	B	C	.	Q	P	G	
31 _H , 32 _H , 33 _H , 34 _H				00 _H , 00 _H	07 _H , 00 _H	41 _H , 42 _H , 43 _H	2E _H , 51 _H		50 _H , 47 _H				

通信示例(子指令为0040的情况下)

复制RCPU文件。

下述条件中的文件复制示例如下所示。

- 复制目标口令、复制源口令：A～Z(26字符)
- 复制源驱动器No.：2
- 复制目标驱动器No.：4
- 复制源文件名：LINE.CSV(8字符)的情况下
- 复制目标文件名：LINE.CSV(8字符)的情况下

■以ASCII代码进行数据通信时

(请求数据)

子指令								固定数据(16字符)															
1	8	2	4	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0				
31H	38H	32H	34H	30H	30H	34H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H				
复制目标 口令字符数				复制目标口令				复制目标 驱动器No.				复制目标 文件名字符数				复制目标文件名							
0	0	1	A	A	B	C	D	...	Z	0	0	0	4	0	0	0	8	(1)					
30H	30H	31H	41H	41H	42H	43H	44H		54H	30H	30H	30H	34H	30H	30H	30H	38H						
复制源 口令字符数				复制源口令				复制源 驱动器No.				复制源文件名字符数				复制源文件名							
0	0	1	A	A	B	C	D	...	Z	0	0	0	2	0	0	0	8	(1)					
30H	30H	31H	41H	41H	42H	43H	44H		54H	30H	30H	30H	32H	30H	30H	30H	38H						

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	3030344330303439303034453030343530303245303034333030353330303536

■以二进制代码进行数据通信时

(请求数据)

子指令		固定数据(8字节)								复制目标 口令 字符数	复制目标口令						复制目标 驱动器 No.	复制目标 文件 名字符数	复制目标文件名	
											A	B	C	D	...	Z			(1)	
24H, 18H	04H, 00H	00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H	1AH, 00H	41H, 42H, 43H, 44H,			54H	04H, 00H	08H, 00H											

复制源 口令 字符数	复制源口令						复制源 驱动器 No.	复制源 文件 名字符数	复制源文件名	
	A	B	C	D	...	Z			(1)	
1AH, 00H	41H, 42H, 43H, 44H,			5AH	02H, 00H	08H, 00H				

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

Change File State(指令: 1825)

更改文件属性。

请求数据

■子指令为0000、0004的情况下

ASCII

1	8	2	5	子指令	口令	驱动器No.	更改属性	文件名字符数	文件名
31H	38H	32H	35H						

二进制

子指令	口令	驱动器No.	更改属性	文件名字符数	文件名
25H, 18H					

■子指令为0040的情况下

ASCII

1	8	2	5	子指令	口令字符数	口令	驱动器No.	更改属性	文件名字符数	文件名
31H	38H	32H	35H							

二进制

子指令	口令字符数	口令	驱动器No.	更改属性	文件名字符数	文件名
25H, 18H						

■子指令

子指令											
ASCII代码	二进制代码										
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td>00H</td><td>00H</td></tr></table>	00H	00H
0	0	0	0								
30H	30H	30H	30H								
00H	00H										
<table><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>34H</td></tr></table>	0	0	0	4	30H	30H	30H	34H	<table><tr><td>04H</td><td>00H</td></tr></table>	04H	00H
0	0	0	4								
30H	30H	30H	34H								
04H	00H										
<table><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table>	0	0	4	0	30H	30H	34H	30H	<table><tr><td>40H</td><td>00H</td></tr></table>	40H	00H
0	0	4	0								
30H	30H	34H	30H								
40H	00H										

■口令

指定复制目标的文件口令。(136页 口令)

■驱动器No.

指定更改属性的驱动器。(137页 驱动器No.)

■更改属性

指定文件属性。

- 读取专用: 01H
- 可读取、写入: 20H

要点

虽然有指定上述以外的值但未变为出错的情况，但是由于为系统用的值，因此请勿使用。

■文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

■文件名

指定更改属性的文件名。(138页 文件名字符数、文件名)

响应数据

无Change File State指令的响应数据。

注意事项

应将MELSEC-Q/L系列CPU模块置为STOP状态之后再更改下述文件属性。在RUN状态下进行属性更改时将变为出错。

- 参数文件
- 程序存储器(驱动器No.: 0000H)的执行中文件

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。通过SLMP可访问的文件类型，请参阅所使用的模块手册。

通信示例(子指令为0000的情况下)

更改QCPU中存储的文件属性。

下述条件中的更改示例如下所示。

- 口令: 1234
- 驱动器No.: 0
- 属性更改文件名: ABC.QPG
- 更改属性: 读取专用

■以ASCII代码进行数据通信时

(请求数据)

子指令				口令				驱动器No.				更改属性				文件名字符数				文件名										
1	8	2	5	0	0	0	0	1	2	3	4	0	0	0	0	0	0	0	1	0	0	0	7	A	B	C	.	Q	P	G
31 _H	38 _H	32 _H	35 _H	30 _H	30 _H	30 _H	30 _H	31 _H	32 _H	33 _H	34 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	30 _H	31 _H	30 _H	30 _H	30 _H	37 _H	41 _H	42 _H	43 _H	2E _H	51 _H	50 _H	47 _H

■以二进制代码进行数据通信时

(请求数据)

子指令		口令		驱动器No.		更改 属性	文件 名字符数	文件名												
		1	2	3	4				A	B	C	.	Q	P	G					
25 _H	18 _H	00 _H	00 _H	31 _H	32 _H	33 _H	34 _H	00 _H	00 _H	01 _H	00 _H	07 _H	00 _H	41 _H	42 _H	43 _H	2E _H	51 _H	50 _H	47 _H

5

下述条件中的更改示例如下所示。

- 口令: A~Z(26字符)
- 驱动器No.: 4
- 属性更改文件名: LINE.CSV(8字符)的情况下
- 更改属性: 读取专用

(请求数据)

子指令				口令字符数				口令									
1	8	2	5	0	0	4	0	0	0	1	A	A	B	C	D	...	Z
31H, 38H,	32H, 35H	30H, 30H,	34H, 30H	30H, 30H,	31H, 41H	41H, 42H,	43H, 44H,	5AH									

驱动器No.	更改属性	文件名字符数	文件名									
0	0	0	4	0	0	0	1	0	0	0	8	(1)
30H, 30H, 30H,	34H	30H, 30H, 30H,	31H	30H, 30H, 30H,	38H							

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	3030344330303439303034453030343530303245303034333030353330303536

(请求数据)

子指令		口令 字符数	口令						驱动器No.	更改 属性	文件 名字符数	文件名
			A	B	C	D	...	Z				(1)
25H, 18H	40H, 00H	1AH, 00H	41H, 42H, 43H, 44H, ...					5AH	04H, 00H	01H, 00H	08H, 00H	

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

Change File Date(指令：1826)

更改文件的创建日期。

请求数据

ASCII

1	8	2	6	子指令	0	0	0	0	驱动器No.	更改日期	更改时间	文件名字符数	文件名
31H	38H	32H	36H		30H	30H	30H	30H					

二进制

	子指令		驱动器No.	更改日期	更改时间	文件名字符数	文件名
26H	18H	00H	00H	00H	00H		

子指令

子指令											
ASCII代码	二进制代码										
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td></td></tr><tr><td>00H, 00H</td></tr></table>		00H, 00H
0	0	0	0								
30H	30H	30H	30H								
00H, 00H											
<table><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table>	0	0	4	0	30H	30H	34H	30H	<table><tr><td></td></tr><tr><td>40H, 00H</td></tr></table>		40H, 00H
0	0	4	0								
30H	30H	34H	30H								
40H, 00H											

驱动器No.

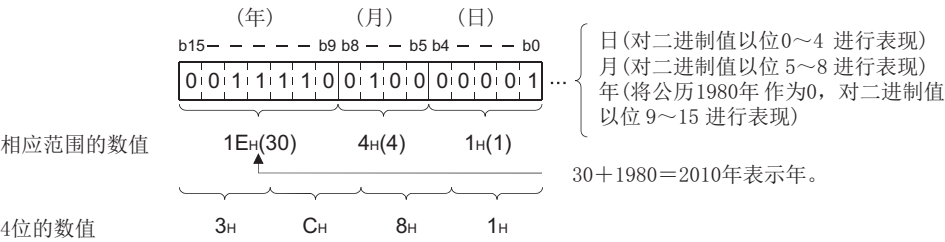
指定更改创建日期的驱动器。(137页 驱动器No.)

更改日期

指定更改的日期。

例

“更改日期”为2010年4月1日的情况下

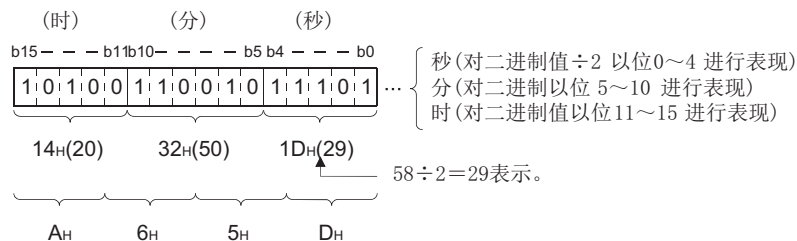


以ASCII代码进行通信时：3C81 (“3”开始按顺序发送。)
以二进制代码进行通信时：3C81H (81H, 3CH的顺序发送。)

■更改时间

例

“更改时间”为20时50分58秒的情况下



以ASCII代码进行通信时：A65D(从“A”开始按顺序发送。)

以二进制代码进行通信时：A65DH(按5DH，A6H的顺序发送。)

■文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

■文件名

指定更改日期的文件名。(138页 文件名字符数、文件名)

响应数据

无Change File Date指令的响应数据。

注意事项

应将MELSEC-Q/L系列CPU模块置为STOP状态之后再更改下述文件日期。在RUN状态下进行日期更改时将变为出错。

- 参数文件
- 程序存储器(驱动器No.: 0000H)的执行中文件

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。通过SLMP可访问的文件类型，请参阅所使用的模块手册。

通信示例(子指令为0000的情况下)

按下述方式更改QCPU文件创建日期。

- 驱动器No.: 0
- 更改的日期: 2010/04/01
- 更改的时间: 20时50分58秒
- 文件名: ABC.QPG

■以ASCII代码进行数据通信时

(请求数据)

子指令								驱动器No.				更改日期				更改时间							
1	8	2	6	0	0	0	0	0	0	0	0	3	C	8	1	A	6	5	D	-			
31H, 38H, 32H, 36H				30H, 30H, 30H, 30H				30H, 30H, 30H, 30H				30H, 30H, 30H, 30H				33H, 43H, 38H, 31H				41H, 36H, 35H, 44H			
文件名字符数										文件名													
0 0 0 7										A B C . Q P G													
30H, 30H, 30H, 37H										41H, 42H, 43H, 2EH, 51H, 50H, 47H													

■以二进制代码进行数据通信时
(请求数据)

子指令		驱动器No.		更改日期	更改时间	文件名 名字符数	文件名
26H, 18H	00H, 00H	00H, 00H, 00H, 00H	00H, 00H	81H, 3CH	5DH, A6H	07H, 00H	A B C . Q P G 41H, 42H, 43H, 2EH, 51H, 50H, 47H

通信示例(子指令为0040的情况下)

按下述方式更改RCPU文件创建日期。

- 驱动器No.: 4
- 更改的日期: 2010/04/01
- 更改的时间: 20时50分58秒
- 文件名: LINE.CSV(8字符)的情况下

■以ASCII代码进行数据通信时
(请求数据)

子指令		驱动器No.		更改日期	更改时间	
1 8 2 6	0 0 4 0	0 0 0 0	0 0 0 4	3 C 8 1	A 6 5 D	
31H, 38H, 32H, 36H	30H, 30H, 34H, 30H	30H, 30H, 30H, 30H	30H, 30H, 30H, 34H	33H, 43H, 38H, 31H	41H, 36H, 35H, 44H	

文件名 字符数	文件名
0 0 0 8 30H, 30H, 30H, 38H	(1)

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	303034433030343930303034453030343530303245303034333030353330303536

■以二进制代码进行数据通信时
(请求数据)

子指令		驱动器No.		更改日期	更改时间	文件名 名字符数	文件名
26H, 18H	40H, 00H	00H, 00H, 00H, 00H	04H, 00H	81H, 3CH	5DH, A6H	08H, 00H	(1)

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

Open File(指令: 1827)

为了防止通过其它设备更改文件内容, 对文件进行锁定。

要点

- 文件锁定, 通过下述之一解除。
- Close File(指令: 182A)的执行 (☞ 182页 Close File(指令: 182A))
 - 模块再启动(CPU模块的复位等)

请求数据

■子指令为0000、0004的情况下

ASCII

1	8	2	7	子指令	口令	打开模式	驱动器No.	文件名字符数	文件名
31H	38H	32H	37H						

二进制

	子指令	口令	打开模式	驱动器No.	文件名字符数	文件名
27H, 18H						

■子指令为0040的情况下

ASCII

1	8	2	7	子指令	口令字符数	口令	打开模式	驱动器No.	文件名字符数	文件名
31H	38H	32H	37H							

二进制

	子指令	口令字符数	口令	打开模式	驱动器No.	文件名字符数	文件名
27H, 18H							

■子指令

子指令													
ASCII代码	二进制代码												
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td></td><td></td></tr><tr><td>00H</td><td>00H</td></tr></table>			00H	00H
0	0	0	0										
30H	30H	30H	30H										
00H	00H												
<table><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>34H</td></tr></table>	0	0	0	4	30H	30H	30H	34H	<table><tr><td></td><td></td></tr><tr><td>04H</td><td>00H</td></tr></table>			04H	00H
0	0	0	4										
30H	30H	30H	34H										
04H	00H												
<table><tr><td>0</td><td>0</td><td>4</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>34H</td><td>30H</td></tr></table>	0	0	4	0	30H	30H	34H	30H	<table><tr><td></td><td></td></tr><tr><td>40H</td><td>00H</td></tr></table>			40H	00H
0	0	4	0										
30H	30H	34H	30H										
40H	00H												

■口令

指定复制目标的文件口令。(☞ 136页 口令)

■打开模式

对将指定文件作为读取用锁定，还是作为写入用锁定进行指定。

项目	打开模式	
	ASCII代码	二进制代码
作为数据的读取用锁定文件	0000 30H, 30H, 30H, 30H	00H, 00H
作为数据的写入用锁定文件	0100 30H, 31H, 30H, 30H	00H, 01H

■驱动器No.

指定锁定的文件驱动器。(137页 驱动器No.)

■文件名字符数

指定通过“文件名”指定的标签名的字符数。(138页 文件名字符数、文件名)

■文件名

指定锁定的文件名。(138页 文件名字符数、文件名)

响应数据

文件指针No.被存储。(139页 文件指针No.)

通信示例(子指令为0000的情况下)

锁定QCPU文件。

锁定下述条件中的文件。

- 口令：1234
- 驱动器No.：0
- 文件名：ABC.QPG
- 打开模式：写入打开

■以ASCII代码进行数据通信时

(请求数据)

子指令				口令				打开模式				驱动器No.				文件名字符数				文件名							
1	8	2	7	0	0	0	0	1	2	3	4	0	1	0	0	0	0	0	0	7	A	B	C	.	Q	P	G
31H	38H	32H	37H	30H	30H	30H	30H	31H	32H	33H	34H	30H	31H	30H	30H	30H	30H	30H	30H	37H	41H	42H	43H	2EH	51H	50H	47H

(响应数据)

文件指针No.

0	0	0	0
30H	30H	30H	30H

■以二进制代码进行数据通信时

(请求数据)

子指令		口令		打开模式		驱动器No.	文件名字符数	文件名
		1	2	3	4			A B C . Q P G
27H, 18H	00H, 00H	31H, 32H, 33H, 34H		00H, 01H	00H, 00H	07H, 00H		41H, 42H, 43H, 2EH, 51H, 50H, 47H

(响应数据)

文件指针No.

00H, 00H

通信示例(子指令为0040的情况下)

锁定RCPU文件。

锁定下述条件中的文件。

- 口令: A~Z(26字符)
- 驱动器No.: 4
- 文件名: LINE.CSV(8字符)的情况下
- 打开模式: 写入打开

■以ASCII代码进行数据通信时

(请求数据)

子指令				口令字符数				口令										
1	8	2	7	0	0	4	0	0	0	1	A	A	B	C	D	...	Z	-
31H, 38H, 32H, 37H	30H, 30H, 34H, 30H	30H, 30H, 31H, 41H	41H, 42H, 43H, 44H, , 5AH															

打开模式				驱动器No.				文件名字符				文件名			
0	1	0	0	0	0	0	4	0	0	0	8	(1)			
30H, 31H, 30H, 30H	30H, 30H, 30H, 34H	30H, 30H, 30H, 38H													

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	ASCII代码(16进制数)	303034433030343930303445303034353030303245303034333030353330303536

(响应数据)

文件指针No.

0	0	0	0
30H	30H	30H	30H

■以二进制代码进行数据通信时

(请求数据)

子指令		口令 字符数	口令								打开模式	驱动器No.	文件名字符数	文件名
			A	B	C	D	...	Z					(1)	
27H, 18H	40H, 00H	1AH, 00H	41H, 42H,	43H, 44H,				34H	00H, 01H	04H, 00H	08H, 00H			

编号	项目	值
—	文件名	LINE.CSV
	UTF-16(16进制数)	004C0049004E0045002E004300530056
(1)	二进制代码(16进制数)	4C0049004E0045002E00430053005600

(响应数据)

文件指针No.

00H	00H
-----	-----

Read File(指令：1828)

读取文件内容。

请求数据

ASCII

1	8	2	8	子指令	文件指针No.	偏置地址	读取字节数
31H	38H	32H	38H				

二进制

子指令	文件指针No.	偏置地址	读取字节数
28H, 18H			

子指令

子指令

ASCII代码	二进制代码
0 0 0 0 30H, 30H, 30H, 30H	00H, 00H

文件指针No.

指定文件指针No.。(139页 文件指针No.)

偏置地址

指定文件读取开始位置。偏置地址在分割文件读取的情况下使用。

以1次读取文件的情况下指定“0”后，在读取字节数中指定文件容量。

偏置地址对来自于文件的起始(偏置地址：0H)的地址(1地址/1字节)以偶数进行指定。



以ASCII代码进行数据通信时，对偏置地址以ASCII代码8位按照从高位字节到低位字节的顺序进行指定。(16进制数)

例

偏置地址为781H(1921)的情况下

0	0	0	0	0	7	8	1
30H	30H	30H	30H	30H	37H	38H	31H

以二进制代码进行数据通信时，按照从低位字节到高位字节的顺序指定偏置地址。(16进制数)

例

偏置地址为781H(1921)的情况下

81H	07H	00H	00H
-----	-----	-----	-----

要点

文件容量为1921字节以上的情况下，应使用偏置地址，分多次进行读取。文件容量可以通过下述指令确认。

- Read Directory/File(指令: 1810)(145页 Read Directory/File(指令: 1810))
- Search Directory/File(指令: 1811)(154页 Search Directory/File(指令: 1811))

已读取的数据应原样不变地保存到外部设备中。已读取的数据不可以在外部设备侧进行编辑。

■读取字节数

指定读取的文件容量(字节数)。作为1地址/1字节指定。(指定范围: 0~1920)

例

读取字节数为780H(1920)的情况下

ASCII代码	二进制代码										
转换为ASCII代码4位后，按照从高位字节到低位字节的顺序指定。(16进制数)	按照从低位字节到高位字节的顺序指定。(16进制数)										
<table><tr><td>0</td><td>7</td><td>8</td><td>0</td></tr><tr><td>30_H</td><td>37_H</td><td>38_H</td><td>30_H</td></tr></table>	0	7	8	0	30 _H	37 _H	38 _H	30 _H	<table><tr><td>80_H</td><td>07_H</td></tr></table>	80 _H	07 _H
0	7	8	0								
30 _H	37 _H	38 _H	30 _H								
80 _H	07 _H										

响应数据

读取字节数及读取的数据被存储。

ASCII

读取字节数	读取数据
-------	------

二进制

读取字节数	读取数据
-------	------

■读取字节数

以与请求数据的“读取字节数”同样的格式，存储已读取文件的字节数。

■读取数据

存储已读取文件的内容。

注意事项

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。通过SLMP可访问的文件类型，请参阅所使用的模块手册。

通过58页通信示例

- 读取下述文件。
- 文件指针No.：0
 - 读取字节数：1K字节

■以ASCII代码进行数据通信时
(请求数据)

子指令				文件 指针No.				偏置地址								读取字节数			
1	8	2	8	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
31H	38H	32H	38H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	34H	30H

(响应数据)

0	4	0	0	读取数据															
30H	34H	30H	30H																

■以二进制代码进行数据通信时
(请求数据)

子指令	文件指针No.	偏置地址	读取 字节数
28H, 18H	00H, 00H	00H, 00H, 00H, 00H	00H, 04H

(响应数据)

00H, 04H	读取数据
----------	------

Write File(指令: 1829)

将内容写入到文件中。

请求数据

ASCII

1	8	2	9	子指令	文件指针No.	偏置地址	写入字节数	写入数据
31H	38H	32H	39H					

二进制

	子指令	文件指针No.	偏置地址	写入字节数	写入数据
29H	18H				

子指令

子指令	
ASCII代码	二进制代码
0 0 0 0 30H 30H 30H 30H	00H 00H

文件指针No.

指定文件指针No.。(139页 文件指针No.)

偏置地址

指定文件写入开始位置。偏置地址在分割文件写入的情况下使用。以1次写入文件的情况下指定“0”。
偏置地址对来自于文件的起始(偏置地址: 0H)的地址(1地址/1字节)以偶数或4的倍数进行指定。

- 驱动器No.0000(程序存储器、参数存储器)中写入的情况下: 以4的倍数指定
- 驱动器No.0000以外中写入的情况下: 以偶数指定

偏置地址

0

数据

(文件容量) - 1

以ASCII代码进行数据通信时, 对偏置地址以ASCII代码8位按照从高位字节到低位字节的顺序进行指定。(16进制数)

例

偏置地址为781H(1921)的情况下

0	0	0	0	0	7	8	1
30H	30H	30H	30H	30H	37H	38H	31H

以二进制代码进行数据通信时, 按照从低位字节到高位字节的顺序指定偏置地址。(16进制数)

例

偏置地址为781H(1921)的情况下

81H	07H	00H	00H
-----	-----	-----	-----

要点 🔍

文件容量为1921字节以上的情况下，应使用偏置地址，分多次进行写入。文件容量可以通过下述指令确认。

- Read Directory/File(指令: 1810)(📖 145页 Read Directory/File(指令: 1810))
- Search Directory/File(指令: 1811)(📖 154页 Search Directory/File(指令: 1811))

下述文件应在将CPU模块置为STOP状态之后再进行写入。在RUN状态下进行写入时将变为出错。

- 参数文件
- 程序存储器(驱动器No.: 0000H)的当前执行中文件

■写入字节数

指定写入文件的容量(字节数)。作为1地址/1字节指定。
(指定范围: 0~1920或0~New File(指令: 1820)中指定的文件容量)

例

写入字节数为780H(1920)的情况下

ASCII代码	二进制代码
转换为ASCII代码4位后，按照从高位字节到低位字节的顺序指定。(16进制数)	按照从低位字节到高位字节的顺序指定。(16进制数)
0780 30H, 37H, 38H, 30H	80H, 07H

■写入数据

通过Read File(指令: 1828)指定已读取的数据。

响应数据

以与请求数据的“写入字节数”同样的格式，存储已写入文件的字节数。

注意事项

在MELSEC iQ-R系列模块中，有通过SLMP不可访问的文件类型。通过SLMP可访问的文件类型，请参阅所使用的模块手册。

通信示例

写入下述文件中。

- 文件指针No.: 0
- 偏置地址: 0
- 写入字节数: 1K字节

■以ASCII代码进行数据通信时
(请求数据)

子指令				文件指针No.				偏置地址								写入字节数				写入数据	
1	8	2	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	
31H	38H	32H	39H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	30H	34H	30H	30H

(响应数据)

写入字节数			
0	4	0	0
30H	34H	30H	30H

■以二进制代码进行数据通信时
(请求数据)

子指令	文件指针No.	偏置地址	写入字节数	写入数据
29H, 18H	00H, 00H	00H, 00H	00H, 00H, 00H, 00H	00H, 04H

(响应数据)

写入字节数
00H, 04H

Close File(指令：182A)

对通过打开处理的文件锁定进行解除。

请求数据

ASCII

1	8	2	A	子指令	文件指针No.	关闭类型
31H	38H	32H	41H			

二进制

	子指令	文件指针No.	关闭类型
2AH	18H		

子指令

子指令	
ASCII代码	二进制代码
0000 30H, 30H, 30H, 30H	00H, 00H

文件指针No.

指定文件指针No.。(139页 文件指针No.)

关闭类型

对仅解除对象文件的文件块还是也解除其它被锁定的文件块进行指定。

解除对象	关闭类型											
	ASCII代码	二进制代码										
仅执行指令的外部设备锁定的文件*1	<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>30H</td></tr></table>	0	0	0	0	30H	30H	30H	30H	<table><tr><td>00H</td><td>00H</td></tr></table>	00H	00H
0	0	0	0									
30H	30H	30H	30H									
00H	00H											
执行指令的外部设备锁定的全部文件*2	<table><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>31H</td></tr></table>	0	0	0	1	30H	30H	30H	31H	<table><tr><td>01H</td><td>00H</td></tr></table>	01H	00H
	0	0	0	1								
30H	30H	30H	31H									
01H	00H											
	<table><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr><tr><td>30H</td><td>30H</td><td>30H</td><td>32H</td></tr></table>	0	0	0	2	30H	30H	30H	32H	<table><tr><td>02H</td><td>00H</td></tr></table>	02H	00H
0	0	0	2									
30H	30H	30H	32H									
02H	00H											

*1 如果对于其它外部设备锁定的文件执行指令，指令将异常完成。
*2 在由外部设备故障等导致锁定了文件的外部设备无法解除锁定时使用。

要点

通过模块再启动(CPU模块复位等)，文件锁定将被解除。

响应数据

无Close File指令的响应数据。

通信示例

下述条件中的文件关闭示例如下所示。

- 文件指针No.：0
- 关闭类型：2(被锁定的全部文件)

■以ASCII代码进行数据通信时
(请求数据)

子指令				文件指针No.				关闭类型			
1	8	2	A	0	0	0	0	0	0	0	2
31H	38H	32H	41H	30H	30H	30H	30H	30H	30H	30H	32H

■以二进制代码进行数据通信时
(请求数据)

子指令		文件指针No.		关闭类型
2AH	18H	00H	00H	02H

5.9 Self Test(反复测试)(指令：0619)

对外部设备与以太网搭载模块的通信是否正常动作进行测试。通过进行反复测试，可以确认与外部设备的连接是否正确、数据通信是否正常动作。

要点

仅对于与外部设备连接的以太网搭载模块可以进行反复测试。对于经由网络的其它站模块不可以使用反复测试。

请求数据

ASCII			
0 6 1 9	子指令	反复数据数	反复数据
30H, 36H, 31H, 39H			

二进制			
	子指令	反复数据数	反复数据
19H, 06H			

子指令

子指令											
ASCII代码	二进制代码										
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>30_H</td><td>30_H</td><td>30_H</td><td>30_H</td></tr></table>	0	0	0	0	30 _H	30 _H	30 _H	30 _H	<table><tr><td></td></tr><tr><td>00_H , 00_H</td></tr></table>		00 _H , 00 _H
0	0	0	0								
30 _H	30 _H	30 _H	30 _H								
00 _H , 00 _H											

反复数据数

以字节数指定“反复数据”的数据数。可指定范围为1～960字节。

例

反复数据数为5字节的情况下

使用ASCII代码时，将字节数转换为ASCII代码4位(16进制数)后，按照从高位字节到低位字节的顺序指定。

0 0 0 5
30H, 30H, 30H, 35H

使用二进制代码时，对表示字节数的2字节的数字按照从低位字节到高位字节的顺序指定。

05H, 00H

反复数据

通过反复测试指定进行发送接收的数据。

以ASCII代码进行数据通信时，指定最大960字符的半角字符串(“0”～“9”、“A”～“F”)后，从起始进行发送。

以二进制代码进行数据通信时，将半角字符(“0”～“9”、“A”～“F”)代码作为1字节数值，从起始字符代码发送最大960字节。

响应数据

与请求报文中指定的“反复数据数”、“反复数据”相同的内容被存储。

ASCII

反复数据数	反复数据
-------	------

二进制

反复数据数	反复数据
-------	------

通信示例

反复数据为“ABCDE”时，进行反复测试。

■以ASCII代码进行数据通信时
(请求数据)

子指令	反复数据数	反复数据
0 6 1 9 30H, 36H, 31H, 39H	0 0 0 0 30H, 30H, 30H, 30H	0 0 0 5 30H, 30H, 30H, 35H
		A B C D E 41H, 42H, 43H, 44H, 45H

(响应数据)

反复数据数	反复数据
0 0 0 5 30H, 30H, 30H, 35H	A B C D E 41H, 42H, 43H, 44H, 45H

■以二进制代码进行数据通信时
(请求数据)

子指令	反复数据数	反复数据
19H, 06H, 00H, 00H	05H, 00H	A B C D E 41H, 42H, 43H, 44H, 45H

(响应数据)

反复数据数	反复数据
05H, 00H	A B C D E 41H, 42H, 43H, 44H, 45H

5.10 Clear Error(出错代码初始化、LED熄灯)(指令：1617)

将本站的COM. ERR. LED熄灯。

请求数据

ASCII

1	6	1	7	0	0	0	0
31H	36H	31H	37H	30H	30H	30H	30H

二进制

17H	16H	00H	00H

响应数据

无Clear Error指令的响应数据。

通信示例

以上述“请求数据”中所示的报文格式，通过外部设备发送请求报文。

5.11 Ondemand(指令: 2101)

通过CPU模块对于SLMP对应设备提出发送请求后，将数据发送到外部设备。

来自于SLMP对应设备的数据

ASCII

2	1	0	1	0	0	0	0	发送数据
32H	31H	30H	31H	30H	30H	30H	30H	

二进制

		发送数据
01H	21H	
00H	00H	

■发送数据

存储来自于SLMP对应设备的发送数据(最大1920字节(最大960字))

通信示例

以上述“来自于SLMP对应设备的数据”中所示的报文格式，通过SLMP对应设备接收数据。
关于来自于SLMP对应设备的数据发送方法，请参阅所使用的SLMP对应设备手册。

6 故障排除

本章对通过外部设备无法与SLMP对应设备进行通信时的外部设备侧的原因确定及处理方法有关内容进行说明。
关于SLMP对应设备侧的故障排除有关内容，请参阅所使用的SLMP对应设备手册。

检查项目	处理方法
在使用TCP/IP时，是否确立与SLMP对应设备的连接。	应通过外部设备对于SLMP对应设备，进行连接请求。(Active打开)
是否通过外部设备发送请求报文。	- 应通过外部设备对于SLMP对应设备，发送请求报文。 - 应确认请求报文的发送目标是否为对应于SLMP的设备。(9页 SLMP对应设备)
网络的线路负载是否过高。	- 应降低通过外部设备发送请求报文的频率。 - 应减少网络的线路负载。
IP地址是否正确。	- 外部设备的IP地址的网络部分应与SLMP对应设备一致。 - 请勿使其它以太网设备与IP地址重复。 - 应将请求报文的目標IP地址置为SLMP对应设备的IP地址。
协议(TCP/IP或UDP/IP)是否正确。	外部设备的协议应与SLMP对应设备中设置的协议一致。
端口编号是否正确。	应将请求报文的目標端口编号与SLMP对应设备中设置的本站端口编号一致。
通信数据代码(ASCII代码或二进制代码)在外部设备与SLMP对应设备中是否一致。	应将请求报文的通信数据代码(ASCII代码或二进制代码)与SLMP对应设备中设置的通信数据代码一致。
请求报文的格式是否正确。	应以本手册中记载的报文格式发送请求报文。(15页 请求报文)
请求报文内指定的数据的存储顺序及值的范围是否正确。	应对请求报文内指定的数据在本手册中记载的存储顺序及范围内进行指定。 15页 报文格式 23页 指令
响应报文的“结束代码”是否变为0。	“结束代码”为0以外的情况下，在SLMP对应设备将发生出错。应通过所使用的SLMP对应设备的手册确认结束代码后进行处理。
在使用TCP/IP时，外部设备中预定接收的响应报文长与实际接收的响应报文长是否一致。	- 实际的响应报文长短于预定接收的响应报文长的情况下，应进行接收残余数据的处理。 - 实际的响应报文长长于预定接收的响应报文长的情况下，应确认请求报文。 - 应复位SLMP对应设备。
是否设置防火墙。	应重新审核防火墙的设置。
使用标签访问的情况下，是否通过GX Works3的全局标签设置编辑器将“来自于外部设备的访问”的设置项目设置为有效。	应通过GX Works3的全局标签设置编辑器将“来自于外部设备的访问”的设置项目设置为有效。

附录

附1 通过软元件的扩展指定进行读取、写入

通过将请求数据内的子指令置为008□，可以进行如下所示访问。

- 至链接直接软元件的访问
- 至模块访问软元件的访问
- 至CPU缓冲存储器访问软元件的访问
- 通过变址寄存器间接指定网络No.及起始输入输出编号的访问
- 通过变址寄存器或长变址寄存器间接指定软元件编号的访问
- 通过字软元件中存储的值间接指定软元件编号的访问

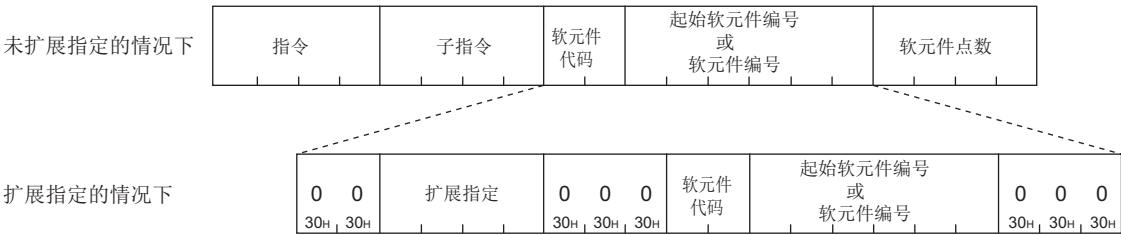
至链接直接软元件的访问

访问远程输入(RX)、远程输出(RY)、链接特殊继电器(SB)等的网络模块的链接软元件。

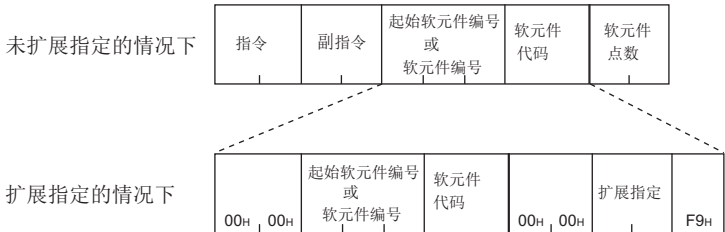
请求数据

■子指令为0081、0080的情况下

ASCII

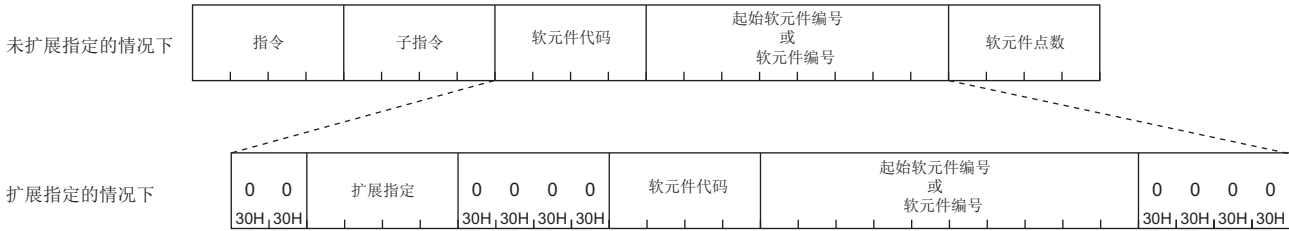


二进制

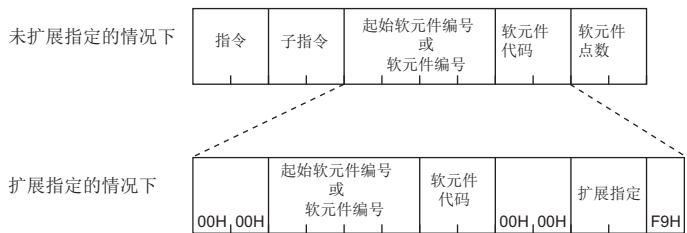


■子指令为0083、0082的情况下

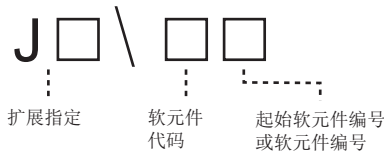
ASCII



二进制



链接直接软元件与请求数据的对应如下所示。



要点 🔍

通过在可指定的指令中，将多个软元件在“扩展指定”中指定0，也可以对下述页中记载的软元件进行访问。

- 📖 29页 软元件代码

但是，在“子指令”中指定了008□时，应通过上述报文格式指定软元件。在1个报文中，未进行扩展指定时的报文格式与进行扩展指定时的报文格式不可以混在一起。

■指令

可以通过下述指令进行访问。

项目		指令
类型	操作	
Device	Read	0401
	Write	1401
	Read Random	0403
	Write Random	1402
	Entry Monitor Device	0801
	Read Block	0406
	Write Block	1406

■子指令

项目	子指令	
	ASCII代码	二进制代码
以位单位进行访问的情况下	0 0 8 1 30H, 30H, 38H, 31H	81H, 00H
	0 0 8 3 30H, 30H, 38H, 33H	83H, 00H
以字单位进行访问的情况下	0 0 8 0 30H, 30H, 38H, 30H	80H, 00H
	0 0 8 2 30H, 30H, 38H, 32H	82H, 00H

■扩展指定

指定访问对象的网络No.。

ASCII代码	二进制代码
以16进制数(ASCII代码3位)指定网络No.。 例 网络No. 8 的情况下 J 4AH, J 0 0 8 4AH, 30H, 30H, 38H	以16进制数(2字节)指定网络No.。 例 网络No. 8 的情况下 H, H 08H, 00H

要点

使用CPU模块的变址寄存器，也可以进行访问对象的网络No.的间接指定。(200页 通过变址寄存器间接指定网络No.及起始输入输出编号的访问)

■软元件代码

指定下述软元件代码。

软元件	类型	软元件代码				软元件编号范围	
		ASCII代码		二进制代码			
		MELSEC iQ-R系列 ^{*1}	MELSEC-Q/L系列 ^{*2}	MELSEC iQ-R系列	MELSEC-Q/L系列		
链接输入(X)	位	X***	X*	009CH	9CH	在访问目标模块具有的软元件编号的范围内进行指定。	16进制
链接输出(Y)		Y***	Y*	009DH	9DH		16进制
链接继电器(B)		B***	B*	00A0H	A0H		16进制
链接特殊继电器(SB)		SB**	SB	00A1H	A1H		16进制
链接寄存器(W)	字	W***	W*	00B4H	B4H		16进制
链接特殊寄存器(SW)		SW**	SW	00B5H	B5H		16进制

1 ASCII代码的情况下，以4位指定软元件代码。软元件代码为3位以下的情况下，将在软元件代码的后面添加“”（ASCII代码：2AH）或空格（ASCII代码：20H）。

2 ASCII代码的情况下，以2位指定软元件代码。软元件代码为1位的情况下，将在软元件代码的后面添加“”（ASCII代码：2AH）或空格（ASCII代码：20H）。

■起始软元件或软元件编号

以16进制数指定起始软元件或软元件编号。（ 32页 起始软元件编号(软元件编号)）

要点	通过使用CPU模块的变址寄存器或长变址寄存器，可以间接指定访问对象的软元件编号。（  205页 通过变址寄存器或长变址寄存器间接指定软元件编号的访问）
-----------	--

响应数据

与未进行扩展指定时相同。

通信示例

访问网络No.1的W100(J1\W100)。

■以ASCII代码进行数据通信时

(请求数据)

子指令	扩展指定	软元件代码	起始软元件编号或软元件编号
0 0 8 0 30H, 30H, 38H, 30H	0 0 30H, 30H	J 0 0 1 4AH, 30H, 30H, 31H	0 0 0 30H, 30H, 30H
0 0 0 30H, 30H, 30H	W * 57H, 2AH	0 0 0 1 0 0 30H, 30H, 31H, 30H, 30H	0 0 0 30H, 30H, 30H

■以二进制代码进行数据通信时

(请求数据)

副指令	起始软元件编号或软元件编号	软元件代码	扩展指定
80H, 00H 00H, 00H	00H, 01H 00H	B4H	00H, 00H 01H, 00H
			F9H

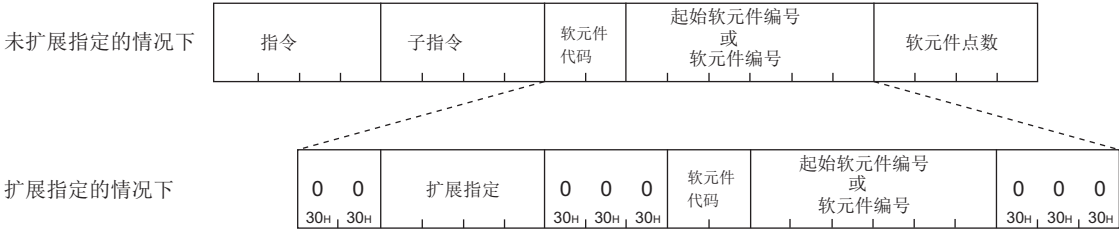
至模块访问软元件的访问

访问SLMP对应设备及智能功能模块的缓冲存储器。

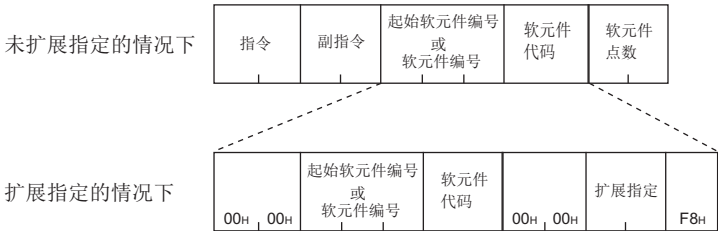
请求数据

■子指令为0080的情况下

ASCII

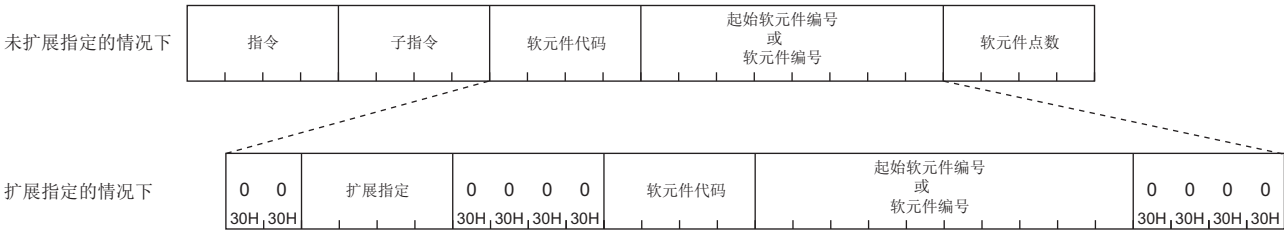


二进制

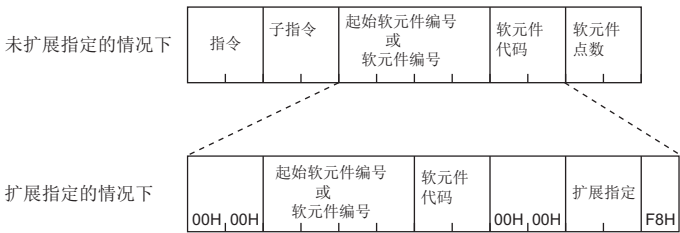


■子指令为0082的情况下

ASCII



二进制代码



模块访问软元件与请求数据的对应如下所示。



要点

通过在可指定的指令中，将多个软元件在“扩展指定”中指定0，也可以对下述页中记载的软元件进行访问。

- ☞ 29页 软元件代码

但是，在“子指令”中指定了008□时，应通过上述报文格式指定软元件。在1个报文中，未进行扩展指定时的报文格式与进行扩展指定时的报文格式不可以混在一起。

■指令

可以通过下述指令进行访问。

项目	指令
类型	操作
Device	Read
	Write
	Read Random
	Write Random
	Entry Monitor Device
	Read Block
	Write Block

■子指令

ASCII代码	二进制代码
0 0 8 0 30H, 30H, 38H, 30H	80H, 00H
0 0 8 2 30H, 30H, 38H, 32H	82H, 00H

■扩展指定

指定智能功能模块的起始输入输出编号。

ASCII代码	二进制代码
以16进制数(ASCII代码3位)指定起始输入输出编号。起始输入输出编号以4位表现了时的前3位进行指定。 例 的情况下 U □ □ □ 55H, , , U 0 0 1 55H, 30H, 30H, 31H	以16进制数(2字节)指定起始输入输出编号。起始输入输出编号以4位表现了时的前3位进行指定。 例 的情况下 □ □ H, □ □ H 01H, 00H

要点

- 在对CC-Link IE现场网络以太网适配器模块等智能功能模块以外的缓冲存储器进行访问的情况下指定0。
- 使用CPU模块的变址寄存器，也可以进行起始输入输出编号的间接指定。(☞ 200页 通过变址寄存器间接指定网络No.及起始输入输出编号的访问)

■软件元件代码

指定下述软件元件代码。

类型	软件元件代码				软件元件编号范围	
	ASCII代码		二进制代码			
	MELSEC iQ-R系列 ^{*1}	MELSEC-Q/L系列 ^{*2}	MELSEC iQ-R系列	MELSEC-Q/L系列		
字	G***	G*	00ABH	ABH	在访问目标模块具有的软件元件编号的范围内进行指定。	10进制

- *1 ASCII代码的情况下，以4位指定软件元件代码。软件元件代码为3位以下的情况下，将在软件元件代码的后面添加“*”（ASCII代码：2AH）或空格（ASCII代码：20H）。
- *2 ASCII代码的情况下，以2位指定软件元件代码。软件元件代码为1位的情况下，将在软件元件代码的后面添加“*”（ASCII代码：2AH）或空格（ASCII代码：20H）。

■起始软件元件或软件元件编号

以10进制数指定起始软件元件或软件元件编号。（☞ 32页 起始软件元件编号（软件元件编号））

要点

通过使用CPU模块的变址寄存器或长变址寄存器，可以间接指定访问对象的软件元件编号。（☞ 205页 通过变址寄存器或长变址寄存器间接指定软件元件编号的访问）

响应数据

与未进行扩展指定时相同。

通信示例

对起始输入输出编号为0030H的智能功能模块的缓冲存储器（地址：1）进行访问。

■以ASCII代码进行数据通信时

（请求数据）

子指令		扩展指定		软件元件代码	起始软件元件编号或软件元件编号
0 0 8 0	0 0	U 0 0 3	0 0 0	G *	0 0 0 0 0 1 0 0 0
30H, 30H, 38H, 30H	30H, 30H	55H, 30H, 30H, 33H	30H, 30H, 30H	47H, 2AH	30H, 30H, 30H, 30H, 30H, 31H, 30H, 30H, 30H

■以二进制代码进行数据通信时

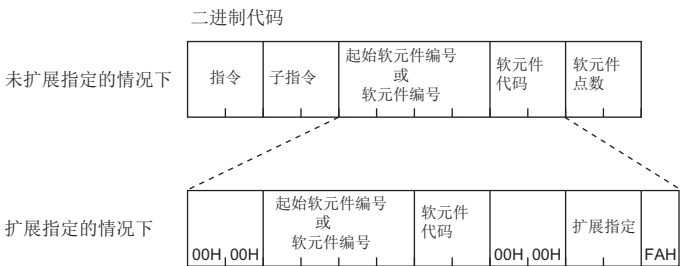
（请求数据）

副指令		起始软件元件编号或软件元件编号	软件元件代码	扩展指定	
80H, 00H	00H, 00H	01H, 00H, 00H	ABH	00H, 00H	03H, 00H, F8H

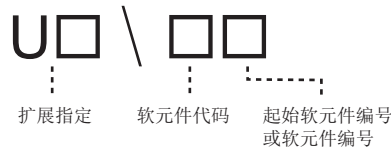
至CPU缓冲存储器访问软元件的访问

访问RCPU的缓冲存储器。

请求数据



CPU模块访问软元件与请求数据的对应如下所示。



■指令

可以通过下述指令进行访问。

项目		指令
类型	操作	
Device	Read	0401
	Write	1401
	Read Random	0403
	Write Random	1402
	Entry Monitor Device	0801
	Read Block	0406
	Write Block	1406

■子指令

ASCII代码	二进制代码
0 0 8 2 30H, 30H, 38H, 32H	82H, 00H

■扩展指定

指定CPU模块的起始输入输出编号。

ASCII代码	二进制代码
以16进制数(ASCII代码3位)指定起始输入输出编号。起始输入输出编号以4位表现了时的前3位进行指定。 U 3 E □ 55H, 33H, 45H,	以16进制数(2字节)指定起始输入输出编号。起始输入输出编号以4位表现了时的前3位进行指定。 □□H, 03H

指定的CPU模块的起始输入输出编号如下所示。

CPU模块的号机编号	起始输入输出编号
1号機	03E0H
2号機	03E1H
3号機	03E2H
4号機	03E3H

要点

使用CPU模块的变址寄存器，也可以间接指定CPU模块的起始输入输出编号。(205页 通过变址寄存器或长变址寄存器间接指定软元件编号的访问)

■软元件代码

指定下述软元件代码。

软元件	类型	软元件代码		软元件编号范围	
		ASCII代码	二进制代码		
		MELSEC iQ-R系列*1	MELSEC iQ-R系列		
CPU缓冲存储器	字	G***	00ABH	在访问目标模块具有的软元件编号的范围内进行指定。	10进制
CPU缓冲存储器的恒定周期区		HQ**	002EH		

1 ASCII代码的情况下，以4位指定软元件代码。软元件代码为3位以下的情况下，将在软元件代码的后面添加“” (ASCII代码：2AH)或空格(ASCII代码：20H)。

■起始软元件或软元件编号

以10进制数指定起始软元件或软元件编号。(32页 起始软元件编号(软元件编号))

要点

通过使用CPU模块的变址寄存器或长变址寄存器，可以间接指定访问对象的软元件编号。(205页 通过变址寄存器或长变址寄存器间接指定软元件编号的访问)

响应数据

与未进行扩展指定时相同。

通信示例

对起始输入输出编号为03E0H的CPU模块的缓冲存储器(地址：1)进行访问。
以ASCII代码进行数据通信时的请求数据如下所示。

■以ASCII代码进行数据通信时
(请求数据)

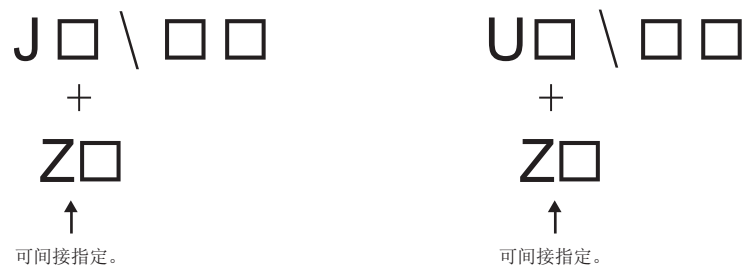
子指令				扩展指定				软元件代码				起始软元件编号或软元件编号																			
0	0	8	2	0	0	U	3	E	0	0	0	0	G	*	*	*	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
30H,30H,38H,32H	30H,30H	55H,33H,45E,30H	30H,30H,30H,30H	47H,2AH,2AH,2AH	30H,30H,30H,30H,30H,30H,30H,30H,30H,30H,30H,30H,31H	30H,30H,30H,30H																									

■以二进制代码进行数据通信时
(请求数据)

子指令		起始软元件编号 或 软元件编号		软元件 代码	扩展指定		
82H, 00H	00H, 00H	01H, 00H, 00H, 00H	ABH, 00H	00H, 00H	E0H, 03H	FAH	

通过变址寄存器间接指定网络No.及起始输入输出编号的访问

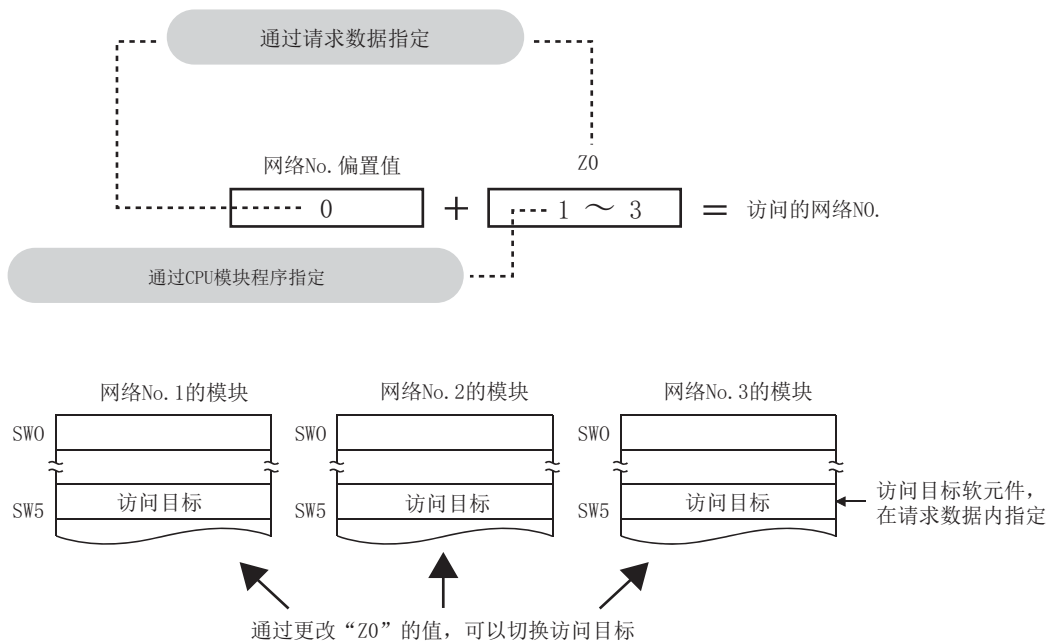
在访问链接直接软元件时，可以通过变址寄存器间接指定访问对象的网络No.。此外，在访问模块访问软元件及CPU缓冲存储器访问软元件时，可以通过变址寄存器间接指定访问对象的起始输入输出编号。



通过在CPU模块程序中更改变址寄存器的值，可以通过1个报文切换访问目标。

例

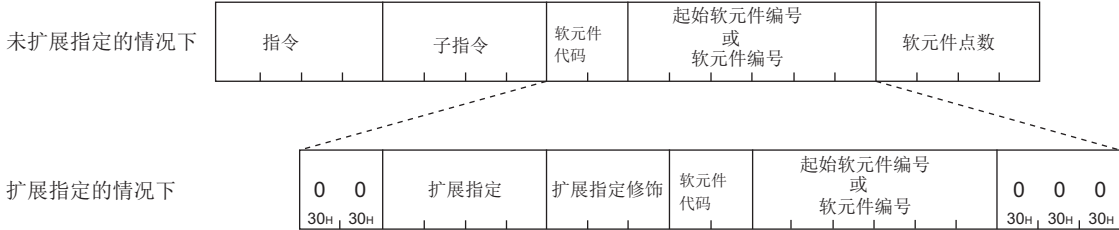
访问目标中安装多个网络模块时，可以在更改“Z0”的值之后切换访问目标。



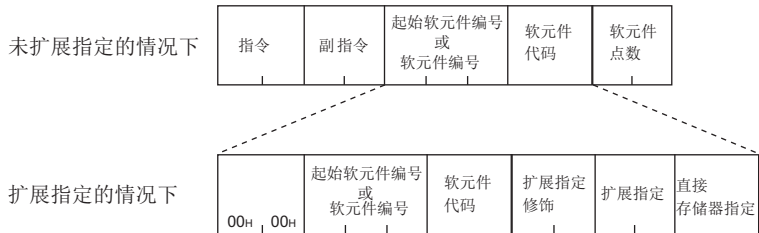
请求数据

■子指令为0081、0080的情况下

ASCII

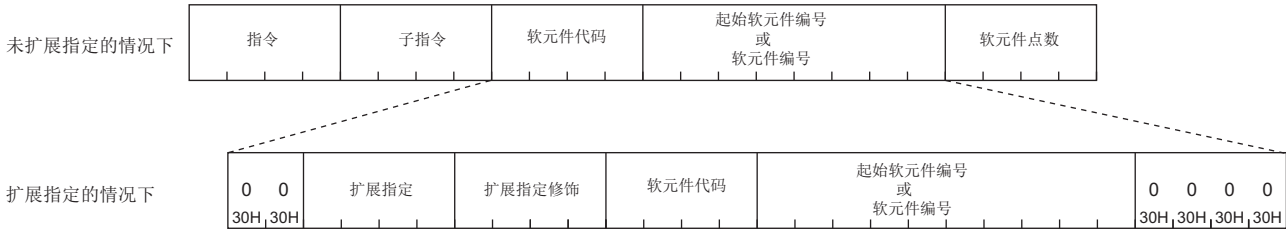


二进制

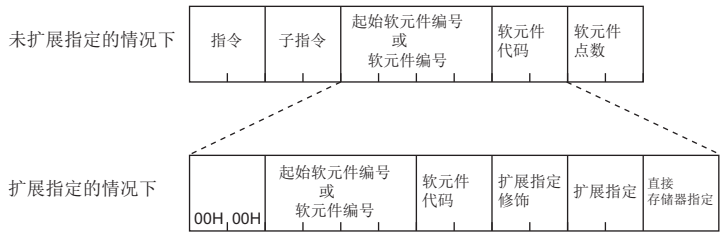


■子指令为0083、0082的情况下

ASCII



二进制



至链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的访问、变址寄存器、请求数据的对应如下所示。



要点

通过在“扩展指定”、“扩展指定修饰”、“直接软元件指定”中指定0，也可以对下述页中记载的软元件进行访问。

- ☞ 29页 软元件代码

但是，在“子指令”中指定了008□时，应通过上述报文格式指定软元件。在1个报文中，未进行扩展指定时的报文格式与进行扩展指定时的报文格式不可以混在一起。

指令

可以通过下述指令进行访问。

项目	指令
类型	操作
Device	Read Random
	Write Random
	Entry Monitor Device
	0403
	1402
	0801

子指令

项目	子指令	
	ASCII代码	二进制代码
以位单位进行访问的情况下	0081 30H, 30H, 38H, 31H	81H, 00H
	0083 30H, 30H, 38H, 33H	83H, 00H
以字单位进行访问的情况下	0080 30H, 30H, 38H, 30H	80H, 00H
	0082 30H, 30H, 38H, 32H	82H, 00H

扩展指定

指定访问对象的网络No.及起始输入输出编号的偏置值。

关于各访问软元件的扩展指定，请参阅下述内容。

项目	参照目标
链接直接软元件	☞ 192页 扩展指定
模块访问软元件	☞ 195页 扩展指定
CPU缓冲存储器访问软元件	☞ 198页 扩展指定

■扩展指定修饰

将"扩展指定"中指定的值置为偏置值，通过变址寄存器间接指定网络No.及起始输入输出编号的情况下，指定变址寄存器编号。

访问目标为MELSEC iQ-R系列模块的情况下指定下述值。

子指令	ASCII代码	二进制代码
0083 0082	以10进制数(ASCII代码2位)指定变址寄存器(Z)编号。(指定范围: 0~24) Z □ □ 5AH, 20H, L Z □ □ 4CH, 5AH,	以16进制数指定变址寄存器(Z)的编号。(指定范围: 00H~18H) □□H, 40H □□H, 80H
0081 0080	以10进制数(ASCII代码2位)指定变址寄存器(Z)编号。(指定范围: 0~24) Z □ □ 5AH,	以16进制数指定变址寄存器(Z)的编号。(指定范围: 00H~18H) □□H, 40H

访问目标为MELSEC-Q/L系列模块的情况下指定下述值。

ASCII代码	二进制代码
以10进制数(ASCII代码2位)指定变址寄存器的编号。 (指定范围: 0~15) Z □ □ 5AH,	以16进制数指定变址寄存器的编号。 (指定范围: 00H~0FH) □□H, 40H

要点

- 以变址寄存器的值间接指定模块访问软元件的起始输入输出编号的情况下，变址寄存器应对“以4位表现了起始输入输出编号时的前3位的值”进行存储。
- 以变址寄存器的值间接指定CPU缓冲存储器访问软元件的起始输入输出编号的情况下，变址寄存器应存储“3E0H~3E3H”。

■软元件代码

指定软元件代码。关于各访问软元件的软元件代码，请参阅下述内容。

项目	浏览目标
链接直接软元件	☞ 193页 软元件代码
模块访问软元件	☞ 196页 软元件代码
CPU缓冲存储器访问软元件	☞ 198页 软元件代码

■起始软元件或软元件编号

以10进制数或16进制数指定起始软元件或软元件编号。

☞ 32页 起始软元件编号(软元件编号)

■直接存储器指定(仅以二进制代码通信时)

指定访问软元件种类。

项目	二进制代码
链接直接软元件	指定F9H。
模块访问软元件	指定F8H。
CPU缓冲存储器访问软元件	指定FAH。

响应数据

与未进行扩展指定时相同。

通信示例

对网络No.1+Z0的W100(J1+Z0\W100)进行访问。

■以ASCII代码进行数据通信时
(请求数据)

子指令				扩展指定				扩展指定修饰		软元件 代码	起始软元件编号 或 软元件编号												
0	0	8	0	0	0	J	0	0	1	Z	0	0	W	*	0	0	0	1	0	0	0	0	0
30 _H , 30 _H , 38 _H , 30 _H				30 _H , 30 _H				4A _H , 30 _H , 30 _H , 31 _H		5A _H , 30 _H , 30 _H		57 _H , 2A _H		30 _H , 30 _H , 30 _H , 31 _H , 30 _H , 30 _H , 30 _H , 30 _H									

■以二进制代码进行数据通信时
(请求数据)

副指令		起始软元件编号 或 软元件编号		软元件 代码	扩展指定 修饰	扩展指定	直接 存储器指定
80H, 00H	00H, 00H	00H, 01H, 00H	B4H	00H, 40H	01H, 00H	F9H	

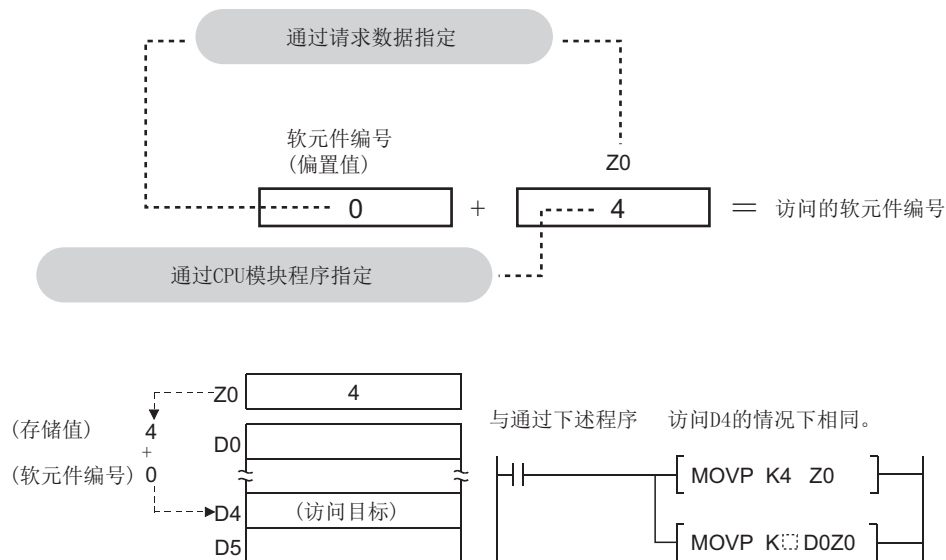
通过变址寄存器或长变址寄存器间接指定软元件编号的访问

在访问软元件时，可以通过变址寄存器或长变址寄存器间接指定软元件编号。

通过在CPU模块程序中更改变址寄存器或长变址寄存器的值，可以通过1个报文切换访问目标。

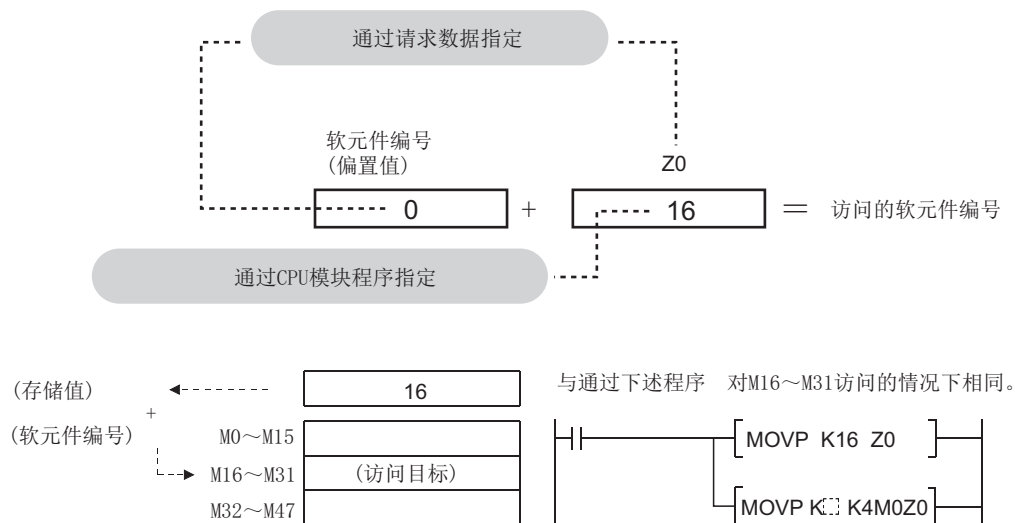
例

通过D0与Z0的指定访问D4的情况下



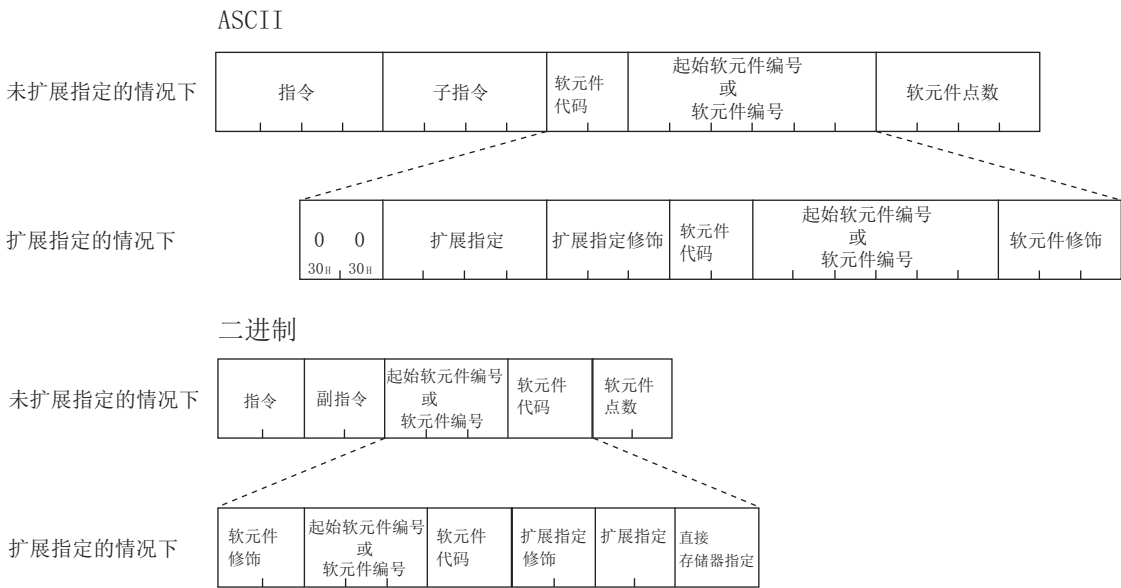
例

通过M0与Z0的指定访问M16~M31的情况下(字单位)

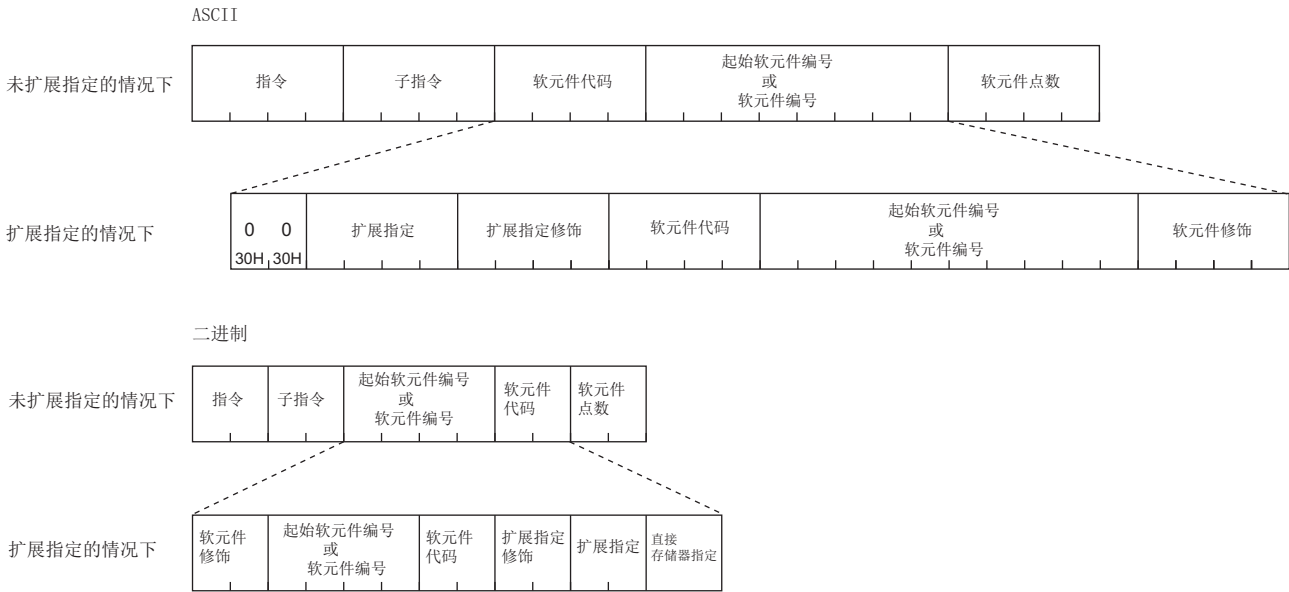


请求数据

■子指令为0081、0080的情况下

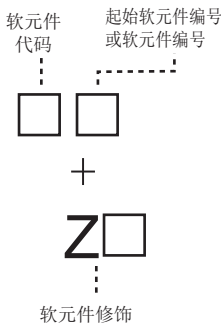


■子指令为0083、0082的情况下



软元件、变址寄存器、长变址寄存、请求数据的对应如下所示。

- 链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件以外



- 链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件



要点 在“子指令”中指定了008□时，应通过上述报文格式指定软元件。在1个报文中，未进行扩展指定时的报文格式与进行扩展指定时的报文格式不可以混在一起。

■指令

可以通过下述指令进行访问。

项目	指令
类型	操作
Device	Read Random
	Write Random
	Entry Monitor Device

■子指令

项目	子指令
	ASCII代码
以位单位进行访问的情况下	0 0 8 1 30H, 30H, 38H, 31H
	0 0 8 3 30H, 30H, 38H, 33H
以字单位进行访问的情况下	0 0 8 0 30H, 30H, 38H, 30H
	0 0 8 2 30H, 30H, 38H, 32H

■扩展指定

指定访问对象的网络No.及起始输入输出编号。
通过“扩展指定修饰”，对网络No.及起始输入输出编号进行间接指定的情况下，通过本项目指定的值将变为偏置值。
关于各访问软元件的扩展指定，请参阅下述内容。

项目	浏览目标
链接直接软元件	☞ 193页 软元件代码
模块访问软元件	☞ 196页 软元件代码
CPU缓冲存储器访问软元件	☞ 198页 软元件代码

访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件以外的软元件时指定“0”。

ASCII代码	二进制代码
指定0。 0 0 0 0 30H, 30H, 30H, 30H	指定0。 00H, 00H

■扩展指定修饰

将“扩展指定”中指定的值置为偏置值，通过变址寄存器间接指定网络No.及起始输入输出编号的情况下，指定变址寄存器编号。(☞ 203页 扩展指定修饰)

要点

以变址寄存器的值间接指定起始输入输出编号的情况下，变址寄存器应对“以4位表现了起始输入输出编号时的前3位的值”进行存储。

■软元件代码

指定访问的软元件代码。(☞ 29页 软元件代码)
请参阅下述软元件代码。

项目	浏览目标
链接直接软元件	☞ 193页 软元件代码
模块访问软元件	☞ 196页 软元件代码
CPU缓冲存储器访问软元件	☞ 198页 软元件代码

■起始软元件或软元件编号

以10进制数或16进制数指定起始软元件或软元件编号。(☞ 32页 起始软元件编号(软元件编号))
通过“软元件修饰”间接指定软元件编号的情况下，通过本项目指定的值将变为偏置值。

■软元件修饰

将"起始软元件或软元件编号"中指定的值作为偏置值，通过变址寄存器或长变址寄存器间接指定软元件编号的情况下，指定变址寄存器或长变址寄存器的编号。

访问目标为MELSEC iQ-R系列模块的情况下指定下述值。

子指令	ASCII代码	二进制代码
0083 0082	以10进制数(ASCII代码2位)指定变址寄存器(Z)编号。(指定范围: 0~24) ^{*1} 以10进制数(ASCII代码2位)指定长变址寄存器(LZ)编号。(指定范围: 0~12) Z □ □ 5AH, 20H, , , L Z □ □ 4CH, 5AH, , ,	以16进制数指定变址寄存器(Z)的编号。(指定范围: 00H~18H) ^{*1} 以16进制数指定长变址寄存器(LZ)的编号。(指定范围: 00H~0CH) □□H, 40H □□H, 80H
0081 0080	以10进制数(ASCII代码2位)指定变址寄存器的编号。(指定范围: 0~24) Z □ □ 5AH, , ,	以16进制数指定变址寄存器的编号。(指定范围: 00H~18H) □□H, 40H

*1 变址寄存器(Z)的软元件修饰范围为-32768~32767。软元件修饰范围超出-32768~32767的情况下，应使用长变址寄存器(LZ)。

(MELSEC iQ-R CPU模块用户手册(应用篇))

访问目标为MELSEC-Q/L系列模块的情况下指定下述值。

ASCII代码	二进制代码
以10进制数(ASCII代码2位)指定变址寄存器的编号。 (指定范围: 0~15) Z □ □ 5AH, , ,	以16进制数指定变址寄存器的编号。 (指定范围: 00H~0FH) □□H, 40H

■直接存储器指定(仅以二进制代码通信时)

访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件的情况下，指定软元件种类。(203页 直接存储器指定(仅以二进制代码通信时))

访问链接直接软元件、模块访问软元件、CPU缓冲存储器访问软元件以外的软元件时指定0。

二进制代码
指定0。 00H, 00H

响应数据

与未进行扩展指定时相同。

通信示例

访问D100+Z4的软元件。

■以ASCII代码进行数据通信时

(请求数据)

子指令	扩展指定		扩展指定修饰	软件 代码	起始软件编号 或 软件编号	软件修饰
0 0 8 0 30 _H , 30 _H , 38 _H , 30 _H	0 0 30 _H , 30 _H	0 0 0 0 30 _H , 30 _H , 30 _H , 30 _H	0 0 0 30 _H , 30 _H , 30 _H	D * 44 _H , 2A _H	0 0 0 1 0 0 30 _H , 30 _H , 30 _H , 31 _H , 30 _H , 30 _H	Z 0 4 5A _H , 30 _H , 34 _H

■以二进制代码进行数据通信时

(请求数据)

副指令	软件件 修饰	起始软件件编号 或 软件件编号	软件件 代码	扩展指定 修饰	扩展指定	直接 存储器指定
80H, 00H	04H, 40H	64H, 00H, 00H	A8H	00H, 00H	00H, 00H	00H

通过字软元件中存储的值间接指定软元件编号的访问

访问对应于字软元件(2点)中存储的地址的软元件。

例

将D100的地址存储到D0中，从外部设备访问“@D0”且访问D100的情况下

在CPU模块侧使用ADRSET指令，将D100的地址存储到D0中。

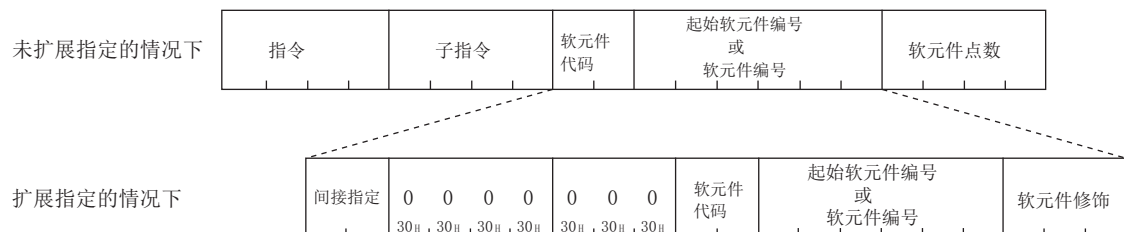


通过在请求数据中指定“@D0”，可以间接访问D100。

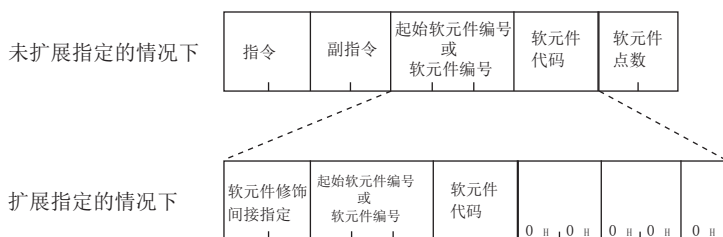
请求数据

■子指令为0080的情况下

ASCII

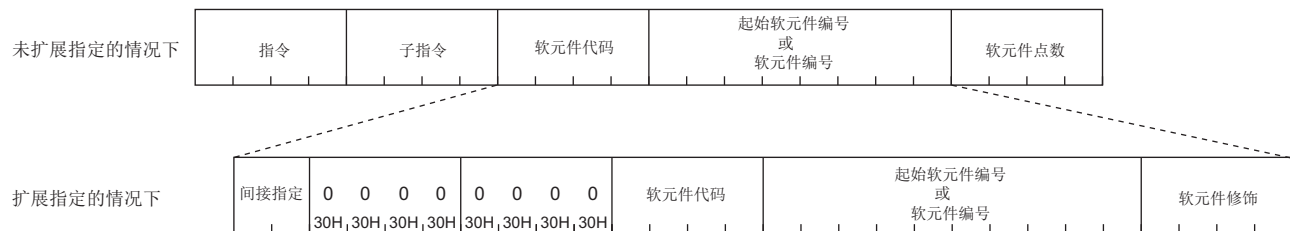


二进制

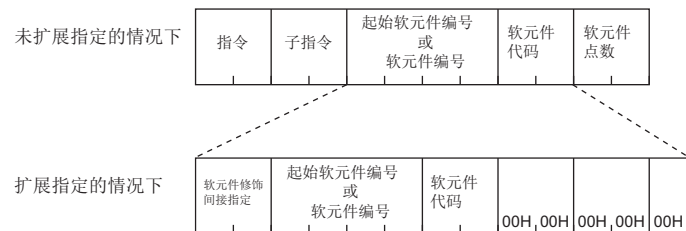


■子指令为0082的情况下

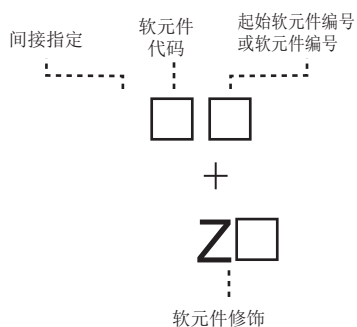
ASCII



二进制



间接指定软元件、变址寄存器、长变址寄存、请求数据的对应如下所示。



要点

在“子指令”中指定了008□时，应通过上述报文格式指定软元件。在1个报文中，未进行扩展指定时的报文格式与进行扩展指定时的报文格式不可以混在一起。

指令

可以通过下述指令进行访问。

项目		指令
类型	操作	
Device	Read Random	0403
	Write Random	1402
	Entry Monitor Device	0801

子指令

ASCII代码	二进制代码
0 0 8 0 30H, 30H, 38H, 30H	80H, 00H
0 0 8 2 30H, 30H, 38H, 32H	82H, 00H

■间接指定、软元件修饰

指定下述2个。

- 间接指定的情况下：指定间接指定软元件的“@”部分。间接指定仅在字软元件中可以指定。
- 软元件修饰的情况下：通过变址寄存器或长变址寄存器对间接指定的软元件进行间接指定的情况下，指定变址寄存器编号。

以ASCII代码进行通信时			
项目	子指令	内容	
间接指定	0082 0080	0 @ 30H, 40H	
软元件修饰	0082	通过变址寄存器进行软元件修饰的情况下	未通过变址寄存器进行软元件修饰的情况下
		Z □ □ 5AH, 20H, ,	0 0 0 0 30H, 30H, 30H, 30H
	0080	通过长变址寄存器进行软元件修饰的情况下	未通过长变址寄存器进行软元件修饰的情况下
		L Z □ □ 4CH, 5AH, ,	0 0 0 0 30H, 30H, 30H, 30H
0080	通过变址寄存器进行软元件修饰的情况下	未通过变址寄存器进行软元件修饰的情况下	
	Z □ □ 5AH, ,	0 0 0 30H, 30H, 30H	

以二进制代码通信时	
子指令	内容
0082	□□H, □□H 无软元件修饰的情况下：00H 有软元件修饰的情况下：设置变址寄存器编号 变址寄存器(Z) (指定范围：00H~18H) 长变址寄存器(LZ) (指定范围：00H~0CH) 0H: 无间接指定 8H: 有间接指定 0H: 无软元件修饰 4H: 有软元件修饰(Z) 8H: 有软元件修饰(LZ)
0080	□□H, □□H 无软元件修饰的情况下： 00H 有软元件修饰的情况下： 变址寄存器编号 (指定范围：00H~0FH) 0H: 无间接指定 8H: 有间接指定 0H: 无软元件修饰 4H: 有软元件修饰

■软元件代码(间接指定仅可指定字软元件代码)

指定访问的软元件代码。(29页 软元件代码)

访问链接直接软元件的情况下，指定下述软元件代码。

软元件	类型	软元件代码				软元件编号范围	
		ASCII代码		二进制代码			
		MELSEC iQ-R系列*1	MELSEC-Q/L系列*2	MELSEC iQ-R系列	MELSEC-Q/L系列		
链接寄存器	字	Ⅱ***	Ⅱ*	00B4H	B4H	在访问目标模块具有的软元件编号的范围内进行指定。	16进制
链接特殊寄存器		SⅡ**	SW	00B5H	B5H		16进制

1 ASCII代码的情况下，以4位指定软元件代码。软元件代码为3位以下的情况下，将在软元件代码的后面添加“” (ASCII代码：2AH)或空格(ASCII代码：20H)。

2 ASCII代码的情况下，以2位指定软元件代码。软元件代码为1位的情况下，将在软元件代码的后面添加“” (ASCII代码：2AH)或空格(ASCII代码：20H)。

访问模块访问软元件的情况下，请参阅下述软元件代码。

196页 软元件代码

访问CPU缓冲存储器访问软元件的情况下，请参阅下述软元件代码。

198页 软元件代码

■起始软元件或软元件编号

以10进制数或16进制数指定起始软元件或软元件编号。(32页 起始软元件编号(软元件编号))

响应数据

与未进行扩展指定时相同。

通信示例

访问@D0+Z4。

在执行指令前，预先放入通过下述程序访问D10的软元件(D100)地址。

此外，置为Z4中K10被存储。



■以ASCII代码进行数据通信时

(请求数据)

子指令	间接指定	软元件代码		起始软元件编号或软元件编号	软元件修饰
0 0 8 0	0 @	0 0 0 0	0 0 0	D *	0 0 0 0 0 0 Z 0 4
30H, 30H, 38H, 30H	30H, 40H	30H, 30H, 30H, 30H	30H, 30H, 30H	44H, 2AH	30H, 30H, 30H, 30H, 30H, 30H, 5AH, 30H, 34H

■以二进制代码进行数据通信时

(请求数据)

副指令	软元件修饰间接指定	起始软元件编号或软元件编号	软元件代码			
80H, 00H	04H, 48H	00H, 00H, 00H	A8H	00H, 00H	00H, 00H	00H

附2 MC协议与SLMP的指令对应表

SLMP是与MC协议的QnA系列3E帧以及4E帧相同的报文格式。MC协议与SLMP的指令对应表如下所示。将使用MC协议的外部设备连接至SLMP对应设备的情况下，应在参考下述内容的基础上确认替换要否。

MC协议			SLMP	
项目	指令	子指令	类型	操作
位单位批量读取	0401	00□1	Device	Read
字单位批量读取		00□0		
位单位批量写入	1401	00□1		Write
字单位批量写入		00□0		
字单位随机读取	0403	00□0		Read Random
位单位随机写入(测试)	1402	00□1		Write Random
字单位随机写入(测试)		00□0		
监视数据登录	0801	00□0		Entry Monitor Device
登录软元件存储器监视	0802	0000	Memory	Execute Monitor
多个块批量读取	0406	00□0		Read Block
多个块批量写入	1406	00□0	Memory	Write Block
缓冲存储器读取	0613	0000		Read
至缓冲存储器写入	1613	0000	Extend Unit	Write
智能功能模块的缓冲存储器读取	0601	0000		Read
智能功能模块的缓冲存储器写入	1601	0000	Remote Control	Write
远程RUN	1001	0000		Remote Run
远程STOP	1002	0000		Remote Stop
远程PAUSE	1003	0000		Remote Pause
远程锁存清除	1005	0000		Remote Latch Clear
远程RESET	1006	0000		Remote Reset
CPU型号读取	0101	0000		Read Type Name
驱动器存储器使用状态读取	0205	0000		在外部设备中使用的情况下，应通过外部设备的程序进行删除。
驱动器存储器的整理整顿	1207	0000		
文件信息一览读取	0201	0000		
		0202		
		0204		
新建文件(文件名登录)	1202	0000		应替换为New File(指令: 1820)。
文件信息更改	1204	0000		应替换为Change File Date(指令: 1826)。
		0001		在外部设备中使用的情况下，应通过外部设备的程序进行删除。
		0002		
文件有无读取(文件查找)	0203	0000		在外部设备中使用的情况下，应通过外部设备的程序进行删除。
文件内容读取	0206	0000		应替换为Read File(指令: 1828)及Write File(指令: 1829)。
至文件的写入	1203	0000		
		0001		
文件锁定的登录、解除	0808	000□		应替换为Open File(指令: 1827)及Close File(指令: 182A)。
文件复制	1206	0000		应替换为Copy File(指令: 1824)。
文件删除	1205	0000		应替换为Delete File指令: 1822)。

MC协议			SLMP	
项目	指令	子指令	类型	操作
目录·文件信息读取	1810	0000	File	Read Directory/File
目录·文件信息查找	1811	0000		Search Directory/File
新建文件	1820	0000		New File
文件删除	1822	0000		Delete File
文件复制	1824	0000		Copy File
文件属性更改	1825	0000		Change File State
文件创建日期更改	1826	0000		Change File Date
文件打开	1827	0000		Open File
文件读取	1828	0000		Read File
至文件的写入	1829	0000		Write File
文件关闭	182A	0000		Close File
反复测试	0619	0000	Self Test	
COM.ERR.LED的熄灯	1617	000□	Clear Error	
远程口令解锁	1630	0000	Remote Password	Unlock
远程口令锁定	1631	0000		Lock

附3 访问目标为多CPU系统的情况下

本章对访问目标为多CPU系统时的SLMP的通信有关内容进行说明。

要点

本内容应在访问目标为多CPU系统的情况下浏览。关于多CPU系统有关内容，请参阅所使用的CPU模块手册。
(所使用的CPU模块用户手册(多CPU系统篇))

访问范围

可以对访问目标的管理CPU及非管理CPU进行访问。
可访问指令如下所示。

项目		浏览目标
类型	操作	
Device	Read	39页 Read(指令: 0401)
	Write	43页 Write(指令: 1401)
	Read Random	46页 Read Random(指令: 0403)
	Write Random	50页 Write Random(指令: 1402)
	Entry Monitor Device ^{*1}	55页 Entry Monitor Device(指令: 0801)
	Execute Monitor ^{*1}	59页 Execute Monitor(指令: 0802)
	Read Block	62页 Read Block(指令: 0406)
	Write Block	66页 Write Block(指令: 1406)
Label	Array Label Read	78页 Array Label Read(指令: 041A)
	Array Label Write	87页 Array Label Write(指令: 141A)
	Label Read Random	96页 Label Read Random(指令: 041C)
	Label Write Random	102页 Label Write Random(指令: 141B)
Extend Unit	Read	116页 Read(指令: 0601)
	Write	118页 Write(指令: 1601)
Remote Control	Remote Run	121页 Remote Run(指令: 1001)
	Remote Stop	123页 Remote Stop(指令: 1002)
	Remote Pause	124页 Remote Pause(指令: 1003)
	Remote Latch Clear	125页 Remote Latch Clear(指令: 1005)
	Remote Reset	126页 Remote Reset(指令: 1006)
	Read Type Name	127页 Read Type Name(指令: 0101)
Remote Password	Lock	131页 Lock(指令: 1631)
	Unlock	133页 Unlock(指令: 1630)
File	Read Directory/File	145页 Read Directory/File(指令: 1810)
	Search Directory/File	154页 Search Directory/File(指令: 1811)
	New File	157页 New File(指令: 1820)
	Delete File	160页 Delete File(指令: 1822)
	Copy File	163页 Copy File(指令: 1824)
	Change File State	167页 Change File State(指令: 1825)
	Change File Date	170页 Change File Date(指令: 1826)
	Open File	173页 Open File(指令: 1827)
	Read File	176页 Read File(指令: 1828)
	Write File	179页 Write File(指令: 1829)
	Close File	182页 Close File(指令: 182A)

*1 不可以对非管理CPU进行访问。

多CPU系统的访问目标CPU的指定

以请求报文内的请求目标模块I/O编号进行指定。(17页 请求目标模块I/O编号)

索引

[Numerics]

4E帧 215

[A]

Array Label Read 78
Array Label Write 87
安全注意事项 1

[B]

报文格式 15
本站 6
本站缓冲存储器访问 108
标签访问 70
标签名 73
标签名长 72

[C]

CC-Link IE现场网络 6
Change File Date 170
Change File State 167
Clear Error(出错代码初始化、LED熄灯) . . . 186
Close File 182
Copy File 163
CPU模块 6,10
出错信息 22

[D]

Delete File 160
读取单位指定、写入单位指定 76
读取数据长、写入数据长 77
读取数据点数、写入数据点数 71
读取数据、写入数据 34
读取文件内容的步骤 140
多CPU系统的情况下 217

[E]

Execute Monitor 59

[F]

访问CPU模块的情况下 14
访问范围 10
副帧头 15
复制文件步骤 142

[G]

各指令的可访问模块 27
更改文件创建日期的步骤 143

[H]

缓冲存储器 6

[J]

监视定时器 19
将数据覆盖到现有文件的步骤 142
结束代码 22

[K]

可其它站访问的模块 10
口令 136

[L]

Label Read Random 96
Label Write Random 102
Lock 131,133
链接软元件 6

[M]

MC协议 6
MC协议与SLMP的指令对应表 215

[N]

New File 157

[O]

Ondemand 187
Open File 173

[P]

排列点数 70

[Q]

QnA系列3E帧 215
起始软元件编号 32
其它站 6
请求报文 6,15
请求目标多点站号 18
请求目标模块I/O编号 17
请求目标网络编号、请求目标站号 16
请求数据长 18
驱动器No. 137

[R]

Read 110,116
Read Block 62
Read Directory/File 145
Read File 176
Read Random 46
Read Type Name 127
Read(指令0401) 39
Remote Latch Clear 125
Remote Pause 124
Remote Reset 126
Remote Run 121
Remote Stop 123

软元件	6
软元件代码	29
软元件点数	33
软元件访问	29

中继站	6
术语	6

[S]

Search Directory/File	154
Self Test(反复测试)	184
SLMP	6
SLMP的作用	7
SLMP对应设备	6,9
SLMP规格	9
SLMP通信步骤	11
删除文件步骤	143
省略点数	71
使用TCP/IP时	11
使用UDP/IP时	12
数据类型ID	75
属性	139

[T]

通过变址寄存器或长变址寄存器间接指定软元件 编号的访问	205
通过变址寄存器间接指定网络No.及起始输入输出 编号的访问	200
通过Entry Monitor Device	55
通过字软元件中存储的值间接指定软元件编号 的访问	211

[W]

Write	112,118
Write Block	66
Write File	179
Write Random	50
Write(指令1401)	43
外部设备	6
位访问点数	38
文件控制	135
文件名	138
文件名字符数	138
文件指针No.	139

[X]

响应报文	6,20
响应数据长	21
新建文件, 写入数据的步骤	141
序列号	15

[Y]

远程操作	120
----------------	-----

[Z]

帧头	15
至CPU缓冲存储器访问软元件的访问	197
至链接直接软元件的访问	190
指令	23
指令一览	24
至模块访问软元件的访问	194
智能功能模块	6
智能功能模块的缓冲存储器访问	113

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
2014年8月	SH(NA) -081292CHN-A	第一版

日文原稿手册：SH-080931-E

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。

© 2014 MITSUBISHI ELECTRIC CORPORATION

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. 根据从三菱电机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. 产品停产后的有偿维修期限

(1) 三菱电机在本产品停产后的 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) 产品停产后，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。

商标

Microsoft、Windows、Windows Vista、Windows NT、Windows XP、Windows Server、Visio、Excel、PowerPoint、Visual Basic、Visual C++、Access是美国Microsoft Corporation在美国、日本及其它国家的注册商标或商标。

Intel、Pentium、Celeron是Intel Corporation在美国及其它国家的商标。

以太网、Ethernet是富士施乐公司的注册商标。

SD标志、SDHC标志是SD-3C、LLC的注册商标或商标。

本手册中使用的其它产品名称和公司名称是各自公司的商标或注册商标。



SH(NA)-081292CHN-A(1408)MEACH

MODEL: SLMP-R-C



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