



三菱微型可编程控制器

**MELSEC iQ-F**  
series

MELSEC iQ-F

FX5用户手册 (MELSEC通信协议篇)



---






# 安全方面注意事项

(使用之前请务必阅读)

在安装、运行、保养·检查本产品之前，请务必仔细阅读本使用说明书以及其他相关设备的所有附带资料，正确使用。请在熟悉了所有关于设备的指示、安全信息，以及注意事项后使用。

在本使用说明书中，安全注意事项的等级用[警告]、[注意]进行区分。

 <b>警告</b>	错误使用时，有可能会引起危险，导致死亡或是重伤事故的发生。
 <b>注意</b>	错误使用时，有可能会引起危险，导致中度伤害或受到轻伤，也有可能造成物品方面的损害。

此外，即使是[注意]中记载的事项，根据状况的不同也可能导致重大事故的发生。

两者记载的内容都很重要，请务必遵守。

此外，请妥善保管好产品中附带的使用说明，以便需要时可以取阅，并请务必将其交给最终用户的手中。

## 【设计注意事项】

### **警告**

- 请在可编程控制器的外部设置安全回路，以便在出现外部电源异常、可编程控制器故障等情况时，也能确保整个系统在安全状态下运行。误动作、误输出有可能会导致事故发生。  
(1) 当CPU模块通过看门狗定时器出错等的自诊断功能检测出异常时，所有的输出变为OFF。此外，当发生了CPU模块不能检测出的输入输出控制部分等的异常时，输出控制有时候会失效。此时，请设计外部回路以及结构，以确保机械在安全状态下运行。
- 将外部设备连接在支持MC协议的设备上，对运行中的他站可编程控制器进行控制（变更数据）时，为了让整个系统一直在安全状态下运行，请在他站可编程控制器的程序上设置互锁回路。  
另外，对运行中的他站可编程控制器进行其他控制（变更程序、变更运行状态（状态控制））时，请务必先熟读手册，在充分确认安全的情况下方可进行操作。尤其是在对离外部设备较远的他站可编程控制器进行上述控制时，有时会因数据通信异常造成无法立刻处理可编程控制器侧的故障的情况。  
在他站可编程控制器的程序上设置互锁回路的同时，作为系统请在外部设备和他站可编程控制器之间规定发生数据通信异常时的处理方法。
- 在支持MC协议的设备及智能功能模块的缓冲存储区中，请勿在“系统区域”或“不可写区域”中写入数据。另外，在对支持MC协议的设备及智能功能模块输出信号时，请勿输出（ON）“禁止使用”的信号。如果在“系统区域”或“不可写区域”中写入数据，或对“禁止使用”的信号进行输出，有造成可编程控制器系统误动作的危险。

## 【启动·维护注意事项】

### **警告**

- 要在运行过程中更改程序、执行强制输出、RUN，STOP等操作前，请务必先熟读手册，在充分确认安全的情况下方可进行操作。操作错误有可能导致机械破损及事故发生。
- 请勿从多个外围设备（编程工具以及GOT）同时更改可编程控制器中的程序。否则可能会破坏可编程控制器的程序，引起误动作。

## 【启动・维护注意事项】

---

### 注意

---

- 对于将周边设备连接到正在运行的支持MC协议的设备、他站的CPU模块后进行的在线操作（运行状态的变更），请务必先熟读手册，在充分确认安全的情况下方可进行。操作错误有可能导致机械破损及事故发生。
-

# 前言

此次承蒙购入MELSEC iQ-F系列可编程控制器产品，诚表谢意。

本手册中记载了有关MELSEC iQ-F系列的MC协议功能的规格和设定。

在使用之前，请阅读本书以及相关产品的手册，希望在充分理解其规格的前提下正确使用产品。

此外，希望本手册能够送达至最终用户处。

## 使用时的请求

- 产品是以一般的工业为对象制作的通用产品，因此不是以用于关系到人身安全之类的环境下使用的机器或是系统为目的而设计、制造的产品。
- 讨论将该产品用于原子能用、电力用、航空宇宙用、医疗用、搭乘移动物体用的机器或是系统等特殊用途的时候，请与本公司的营业窗口查询。
- 虽然该产品是在严格的质量体系下生产的，但是用于那些因该产品的故障而可能导致的重大故障或是产生损失的设备的时候，请在系统上设置备用机构和安全功能的开关。

## 预先通知

- 设置产品时如有疑问，请向具有电气知识(电气施工人员或是同等以上的知识)的专业电气技术人员咨询。关于该产品的操作和使用方法有疑问时，请向技术咨询窗口咨询。
- 本书、技术资料、样本等中记载的事例是作为参考用的，不是保证动作的。选用的时候，请用户自行对机器・装置的功能和安全性进行确认以后使用。
- 关于本书的内容，有时候为了改良可能会有不事先预告就更改规格的情况，还望见谅。
- 关于本书的内容期望能做到完美，可是万一有疑问或是发现有错误，烦请联系本公司或办事处。

# 目录

安全方面注意事项	1
前言	3
关联手册	6
术语	7
<b>第1章 概要</b>	<b>9</b>
1.1 MC协议的概要	9
1.2 MC协议的特点	9
<b>第2章 关于利用MC协议的数据通信</b>	<b>10</b>
2.1 数据通信用帧的种类和用途	10
2.2 MC协议的控制步骤的想法	10
2.3 CPU模块侧访问时间	11
2.4 通信时间	12
<b>第3章 报文格式</b>	<b>14</b>
3.1 报文类型及用途	14
形式及代码	14
帧	14
3.2 各形式的报文格式	15
形式1	15
形式4	16
形式5	17
3.3 设置数据的详细内容(形式)	18
控制代码	18
数据字节数	19
块号	20
帧识别编号	20
和校验码	20
出错代码	22
3.4 各帧的访问范围及设置的数据	23
4C帧	23
3C帧	24
3.5 设置数据的详细内容(帧)	25
站号	25
网络编号、可编程控制器编号	26
请求目标模块I/O编号、请求目标模块站号	27
本站站号	27
<b>第4章 指令</b>	<b>28</b>
4.1 指令和功能一览	28
指令一览	28
4.2 软元件访问	30
在指令内指定的数据	30
软元件代码一览	33
软元件点数	34
访问点数	35

块数 . . . . .	36
读取数据、写入数据 . . . . .	37
软元件存储器扩展指定（以子指令的位7进行设定）. . . . .	43
设置/复位 . . . . .	44
批量读取 . . . . .	45
批量写入 . . . . .	49
随机读取 . . . . .	52
随机写入 . . . . .	57
批量读取多个块 . . . . .	64
批量写入多个块 . . . . .	68
<b>4.3 远程操作 . . . . .</b>	<b>73</b>
在远程操作之前 . . . . .	73
远程RUN . . . . .	73
远程STOP . . . . .	75
远程PAUSE . . . . .	75
远程锁存清除 . . . . .	76
远程RESET . . . . .	77
CPU型号读取 . . . . .	78
<b>4.4 出错代码的初始化 . . . . .</b>	<b>80</b>
<b>4.5 反复测试 . . . . .</b>	<b>81</b>
<b>附录 . . . . .</b>	<b>83</b>
<b>附1 软元件存储器的扩展指定 . . . . .</b>	<b>83</b>
至模块访问软元件的访问 . . . . .	83
以变址寄存器或长变址寄存器间接指定软元件编号的访问 . . . . .	85
通过字软元件中存储的值间接指定软元件编号的访问 . . . . .	90
<b>附2 MC协议的CPU模块侧的处理时间 . . . . .</b>	<b>92</b>
<b>索引 . . . . .</b>	<b>93</b>
修订记录 . . . . .	94
关于保修 . . . . .	95
商标 . . . . .	96

# 关联手册

## 对象模块的用户手册

手册名称<手册编号>	内容
MELSEC iQ-F FX5用户手册(入门篇) <JY997D59501>	记载FX5 CPU模块的性能规格、运行前的步骤、故障排除相关的内容。
MELSEC iQ-F FX5U用户手册(硬件篇) <JY997D58601>	记载FX5U CPU模块的输入输出规格、配线、安装及维护等的硬件相关的详细事项。
MELSEC iQ-F FX5UC用户手册(硬件篇) <JY997D61501>	记载FX5UC CPU模块的输入输出规格、配线、安装及维护等的硬件相关的详细事项。
MELSEC iQ-F FX5用户手册(应用篇) <JY997D58701>	记载程序设计中必要的基础知识、CPU模块的功能、软元件/标签、参数的说明等内容。
MELSEC iQ-F FX5编程手册(程序设计篇) <JY997D58801>	记载梯形图、ST、FBD/LD等程序的规格以及标签的内容。
MELSEC iQ-F FX5编程手册(指令/通用FUN/FB篇) <JY997D58901>	记载在程序中可使用的命令及函数的规格的内容。
MELSEC iQ-F FX5用户手册(串行通信篇) <JY997D59001>	记载简易PLC间链接、MC协议、变频器通信、无顺序通信、通信协议支持相关的内容。
MELSEC iQ-F FX5用户手册(MELSEC通信协议篇) <JY997D60901>(本手册)	对对方设备采用基于MC协议的通信对CPU模块的数据进行读取、写入等的方法进行说明。
MELSEC iQ-F FX5用户手册(MODBUS通信篇) <JY997D59201>	记载MODBUS串行通信相关的内容。
MELSEC iQ-F FX5用户手册(以太网通信篇) <JY997D59301>	记载内置以太网端口通信功能相关的内容。
MELSEC iQ-F FX5用户手册(SLMP篇) <JY997D59101>	对对方设备采用基于SLMP的通信对CPU模块的数据进行读取、写入等的方法进行说明。
MELSEC iQ-F FX5用户手册(定位篇) <JY997D59401>	记载内置定位功能相关的内容。
MELSEC iQ-F FX5用户手册(模拟量篇) <JY997D60601>	记载模拟量功能相关的内容。
GX Works3操作手册 <SH-081271CHN>	记载GX Works3的系统配置、参数设置、在线功能的操作方法等简单工程及结构化工程通用的功能相关的内容。



# 术语

除特别注明的情况外，本手册中使用下列术语进行说明。

- 表示多个型号及版本等的总称时的可变部分。

(例) FX5U-32MR/ES、FX5U-32MT/ES⇒FX5U-32M□/ES

- 关于能够与FX5连接的FX3的设备，请参照FX5用户手册(硬件篇)。

术语	内容
<b>■设备</b>	
FX5	FX5U、FX5UC可编程控制器的总称
FX3	FX3S、FX3G、FX3GC、FX3U、FX3UC可编程控制器的总称
FX5 CPU模块	FX5U CPU模块、FX5UC CPU模块的总称
FX5U CPU模块	FX5U-32MR/ES、FX5U-32MT/ES、FX5U-32MT/ESS、FX5U-64MR/ES、FX5U-64MT/ES、FX5U-64MT/ESS、FX5U-80MR/ES、FX5U-80MT/ES、FX5U-80MT/ESS的总称
FX5UC CPU模块	FX5UC-32MT/D、FX5UC-32MT/DSS的总称
扩展模块	FX5扩展模块、FX3扩展模块的总称
• FX5扩展模块	I/O模块、FX5扩展电源模块、FX5智能功能模块的总称
• FX3扩展模块	FX3扩展电源模块、FX3智能功能模块的总称
• 扩展模块(扩展电缆型)	输入模块(扩展电缆型)、输出模块(扩展电缆型)、总线转换模块(扩展电缆型)、智能功能模块的总称
• 扩展模块(扩展连接器型)	输入模块(扩展连接器型)、输出模块(扩展连接器型)、输入输出模块、总线转换模块(扩展连接器型)、连接器转换模块(扩展连接器型)的总称
I/O模块	输入模块、输出模块、输入输出模块、电源内置输入输出模块的总称
输入模块	输入模块(扩展电缆型)、输入模块(扩展连接器型)的总称
• 输入模块(扩展电缆型)	FX5-8EX/ES、FX5-16EX/ES的总称
• 输入模块(扩展连接器型)	FX5-C32EX/D、FX5-C32EX/DS的总称
输出模块	输出模块(扩展电缆型)、输出模块(扩展连接器型)的总称
• 输出模块(扩展电缆型)	FX5-8EYR/ES、FX5-8EYT/ES、FX5-8EYT/ESS、FX5-16EYR/ES、FX5-16EYT/ES、FX5-16EYT/ESS的总称
• 输出模块(扩展连接器型)	FX5-C32EYT/D、FX5-C32EYT/DSS的总称
输入输出模块	FX5-C32ET/D、FX5-C32ET/DSS的总称
电源内置输入输出模块	FX5-32ER/ES、FX5-32ET/ES、FX5-32ET/ESS的总称
扩展电源模块	FX5扩展电源模块、FX3扩展电源模块的总称
• FX5扩展电源模块	FX5-1PSU-5V的别称
• FX3扩展电源模块	FX3U-1PSU-5V的别称
智能模块	智能功能模块的简称
智能功能模块	FX5智能功能模块、FX3智能功能模块的总称
• FX5智能功能模块	FX5的智能功能模块的总称
• FX3智能功能模块	FX3的特殊功能模块的别称
简单运动模块	FX5-40SSC-S的别称
扩展板	FX5U CPU模块用板的总称
• 通信板	FX5-232-BD、FX5-485-BD、FX5-422-BD-GOT的总称
扩展适配器	FX5 CPU模块用适配器的总称
• 通信适配器	FX5-232ADP、FX5-485ADP的总称
• 模拟量适配器	FX5-4AD-ADP、FX5-4DA-ADP的总称
总线转换模块	总线转换模块(扩展电缆型)、总线转换模块(扩展连接器型)的总称
• 总线转换模块(扩展电缆型)	FX5-CNV-BUS的别称
• 总线转换模块(扩展连接器型)	FX5-CNV-BUSC的别称
电池	FX3U-32BL的别称
外围设备	工程工具、GOT的总称
GOT	三菱图形操作终端 GOT1000、GOT2000系列的总称
<b>■软件包</b>	
工程工具	MELSEC可编程控制器软件包的产品名
GX Works3	SWnDND-GXW3的总称产品名(n表示版本)

术语	内容
<b>■手册</b>	
用户手册	另附手册的总称
• 用户手册(入门篇)	MELSEC iQ-F FX5用户手册(入门篇)的简称
• FX5用户手册(硬件篇)	MELSEC iQ-F FX5U用户手册(硬件篇)、MELSEC iQ-F FX5UC用户手册(硬件篇)的总称
• FX5U用户手册(硬件篇)	MELSEC iQ-F FX5U用户手册(硬件篇)的简称
• FX5UC用户手册(硬件篇)	MELSEC iQ-F FX5UC用户手册(硬件篇)的简称
• 用户手册(应用篇)	MELSEC iQ-F FX5用户手册(应用篇)的简称
编程手册(程序设计篇)	MELSEC iQ-F FX5编程手册(程序设计篇)的简称
编程手册(指令/通用FUN/FB篇)	MELSEC iQ-F FX5编程手册(指令/通用FUN/FB篇)的简称
通信手册	MELSEC iQ-F FX5用户手册(串行通信篇)、MELSEC iQ-F FX5用户手册(MELSEC通信协议篇)、MELSEC iQ-F FX5用户手册(MODBUS通信篇)、MELSEC iQ-F FX5用户手册(以太网通信篇)、MELSEC iQ-F FX5用户手册(SLMP篇)的总称
• 串行通信手册	MELSEC iQ-F FX5用户手册(串行通信篇)的简称
• MELSEC通信协议手册	MELSEC iQ-F FX5用户手册(MELSEC通信协议篇)的简称
• MODBUS通信手册	MELSEC iQ-F FX5用户手册(MODBUS通信篇)的简称
• 以太网通信手册	MELSEC iQ-F FX5用户手册(以太网通信篇)的简称
• SLMP手册	MELSEC iQ-F FX5用户手册(SLMP篇)的简称
定位手册	MELSEC iQ-F FX5用户手册(定位篇)的简称
模拟量手册	MELSEC iQ-F FX5用户手册(模拟量篇)的简称
<b>■通信相关</b>	
内置RS-485接口	CPU模块内置的RS-485接口
串行接口	FX5 CPU模块的内置RS-485接口 (CH1)、通信接口 (CH2)、通信适配器1 (CH3)、通信适配器2 (CH4) 的4个接口的总称
MC协议	MELSEC通信协议的简称。 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。
MC协议对应设备	是可发送接收MC协议的设备的总称。
SLMP	SeamLess Message Protocol的简称。 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报文的设备的总称。
外部设备	通信对象侧的设备的总称。(个人计算机、显示器等)
连接站(本站)	连接站(本站)表示与外部设备直接连接的站。
其它站	其它站表示与连接站(本站)相连接的网络上的站。
中继站	是将多个网络模块安装到1个可编程控制器中, 对其它网络的瞬时传送进行中继的站。
模块访问软元件	MELSEC iQ-R系列/MELSEC iQ-F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。
缓冲存储器	用于储存设定值、监视值等的智能功能模块的存储器。

# 1 概要

本手册中说明了利用对象设备和MC协议的数据通信功能，对CPU模块的数据进行读取、写入等的方法。利用MC协议进行数据通信时，请务必阅读 10页 关于利用MC协议的数据通信。

## 1.1 MC协议的概要

MC协议(MELSEC通信协议)是用于CPU模块或外部设备(个人计算机或显示器等)对MC协议对应设备进行访问的协议。如果是通过MC协议的控制步骤可进行报文的发送接收的设备，则可以通过MC协议进行通信。FX5的串行口的情况下，可以使用MC协议的QnA兼容3C/4C帧进行通信。能够通过个人计算机、显示器等，使用MC协议读取、写入CPU模块的软元件。通过读取、写入软元件，能够利用个人计算机、显示器等对CPU模块实施动作监视、数据分析、生产管理等。在开始MC协议的通信前的流程如下所示。

### 1. 连接电缆与外部设备

进行用来实施MC协议通信的连接。  
关于详细内容，请参考下列手册。

📖 MELSEC iQ-F FX5用户手册(串行通信篇)

### 2. 设定参数

利用工程工具设定模块参数。

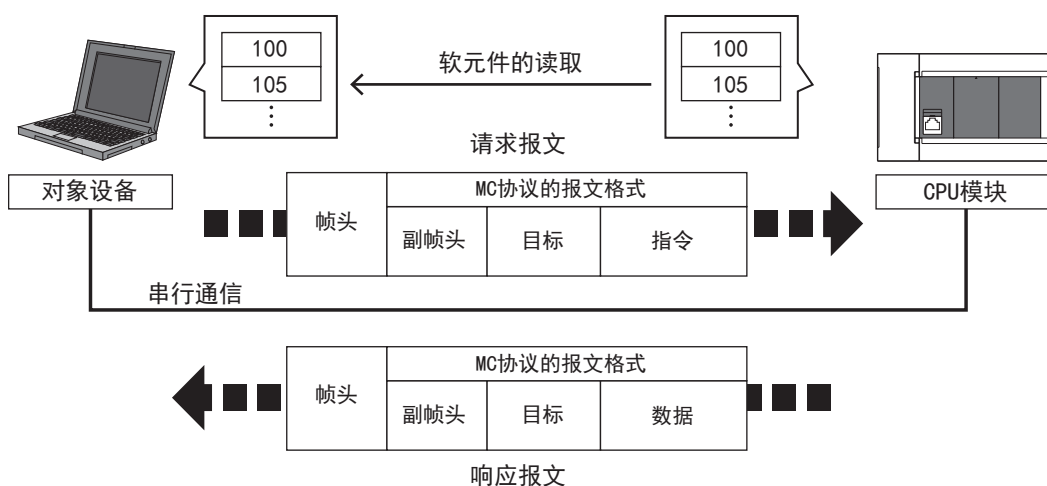
### 3. 写入CPU模块

将设定好的参数写入CPU模块。通过电源的OFF→ON或者系统复位，使参数生效。

## 1.2 MC协议的特点

### 外部设备(个人计算机或显示器等)监视系统

通过从外部设备如下图所示发送MC协议的请求报文，能够读取CPU模块的软元件，因此能够监视系统。此外，不仅是读取软元件，还能够写入软元件或将CPU模块复位等。



## 2 关于利用MC协议的数据通信

以下说明为了让对象设备对CPU模块进行数据的读取、写入等而利用MC协议的数据通信。

### 2.1 数据通信用帧的种类和用途

以下说明用于让对象设备利用MC协议访问CPU模块的帧（数据通信报文）的种类和用途。

对象设备使用串行通信访问CPU模块时，通过收发下述帧的请求报文（访问请求）和响应报文（响应）来进行数据通信。

对象通信方式	能够使用的通信帧	特点及用途	通信数据的代码	控制步骤说明项
串行通信	4C帧	可以从外部设备在最大的访问范围内进行访问。	ASCII代码或二进制代码	14页 报文格式
	3C帧	将通过4C帧访问路径的报文形式进行简略。	ASCII代码	14页 报文格式

#### 要点

FX5 CPU模块支持MC协议的3E帧（以太网通信）。

MC协议的3E帧与SLMP的3E帧为相同的报文格式。

关于MC协议的3E帧详情，请参照以下手册。

📖 MELSEC iQ-F FX5用户手册(SLMP篇)

📖 MELSEC iQ-F FX5用户手册(以太网通信篇)

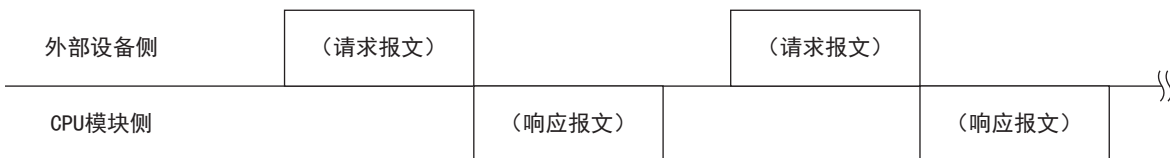
### 2.2 MC协议的控制步骤的想法

以下说明对象设备利用MC协议访问CPU模块时的步骤（控制步骤）的想法。

#### 请求报文的发送

利用MC协议的数据通信采用半双工通信。

访问CPU模块时，请在相对于前一个请求报文的发送，接收到来自CPU模块侧的响应报文后，发送下一个请求报文。  
(在完成响应报文的接收前，不能发送下一个请求报文。)



#### 相对于请求报文，不能接收正常结束的响应报文的情况下

##### ■接收到异常结束的响应报文时

请根据响应报文中的出错代码进行处理。

##### ■无法接收响应报文时或无法全部接收时

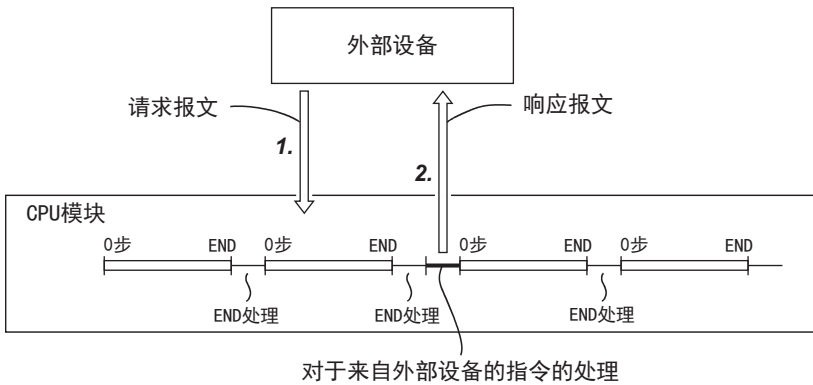
请在经过响应监视定时器值的监视时间后，再次发送请求报文。

请根据需要，变更监视时间的设定值。

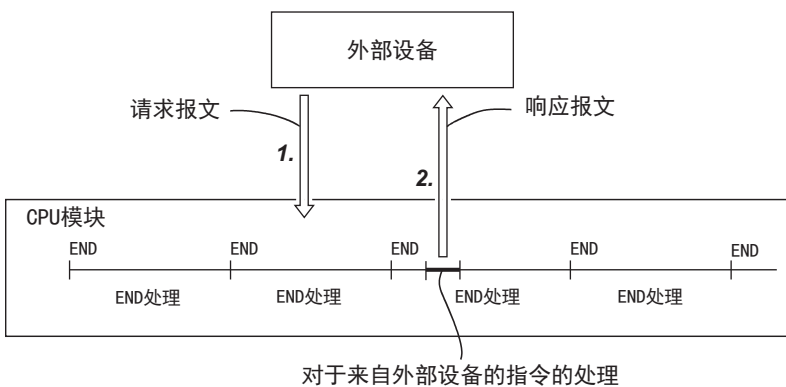
## 2.3 CPU模块侧访问时间

显示从外部设备访问CPU模块（串行通信接口）时的CPU模块侧的访问时间。

• RUN中



• STOP中



1. 为了从外部设备对CPU模块侧进行数据读取/写入请求，发送请求报文。
2. CPU模块根据在执行顺控程序的END命令时从外部设备请求的内容，读取/写入数据，将含有处理结果的响应报文（响应）发送给提出请求的外部设备。

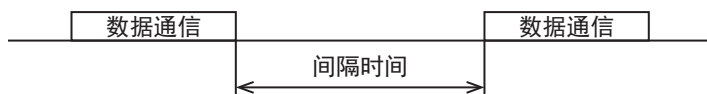
### 要点

- 在相对于指令的请求，CPU模块为RUN中的情况下，以END处理为单位处理外部设备与CPU模块的访问。（扫描时间会延长指令的请求处理时间。）
- 从多个外部设备同时对CPU模块发出访问请求的情况下，根据请求时间，有时请求自外部设备的处理会发送等待，直至实施多次END处理。

## 2.4 通信时间

### 链接时间

#### ■数据通信



#### ■数据通信时间

R: 读取点数、W: 写入点数、T: 每1个字符的发送接收时间、V: 间隔时间、S: 可编程控制器的最大扫描时间、D: 报文等待

##### (1) 3C帧

每个站中连续字软元件(数据寄存器等)的读取时间(ms) =  $(43^{*1} + 4 \times R^{*2}) \times T$  (ms) + V + S (SD524) × 3 + D

每个站中连续字软元件(数据寄存器等)的写入时间(ms) =  $(42^{*1} + 4 \times W^{*2}) \times T$  (ms) + V + S (SD524) × 3 + D

\*1 执行成批读取/写入指令时的协议模式1、无和校验的字符数。

选择协议模式4时, 要在这个数值上“+4”。设置为有和校验时, 还要“+4”。要指定扩展时, 还要“+7”。

\*2 点数是以1个字为单位。

##### (2) 4C帧: ASCII码的情况(使用协议模式1~协议模式4时)

每个站中连续字软元件(数据寄存器等)的读取时间(ms) =  $(49^{*3} + 4 \times R^{*4}) \times T$  (ms) + V + S (SD524) × 3 + D

每个站中连续字软元件(数据寄存器等)的写入时间(ms) =  $(48^{*3} + 4 \times W^{*4}) \times T$  (ms) + V + S (SD524) × 3 + D

\*3 执行成批读取/写入指令时的协议模式1、无和校验的字符数。

选择协议模式4时, 要在这个数值上“+4”。设置为有和校验时, 还要“+4”。要指定扩展时, 还要“+7”。

\*4 点数是以1个字为单位。

##### (3) 4C帧: 二进制码的情况(使用协议模式5时)

每个站中连续字软元件(数据寄存器等)的读取时间(ms) =  $(42^{*5} + 4 \times R^{*6}) \times T$  (ms) + V + S (SD524) × 3 + D

每个站中连续字软元件(数据寄存器等)的写入时间(ms) =  $(40^{*5} + 4 \times W^{*6}) \times T$  (ms) + V + S (SD524) × 3 + D

\*5 执行成批读取/写入指令时的协议模式5、无和校验的字符数。

设置为有和校验时, 还要“+4”。要指定扩展时, 还要“+7”。数据部分存在“10H”时, 为了将DLE(10H)附加到“10H”之前, 要“+ ‘10H的字符数’”。

\*6 点数是以1个字为单位。

## ■每1个字符的发送接收时间

当设定为起始位1位、数据长度7位、奇偶校验1位、停止位1位(合计10位)时，每1个字符的发送接收时间如下所示。

传送速度(波特率)(bps)	1个字符的发送接收时间(ms)
300	33.34
600	16.67
1200	8.34
2400	4.17
4800	2.08
9600	1.04
19200	0.52
38400	0.26
57600	0.17
115200	0.08

当设定报文等待为0ms\*1、最大扫描时间为20ms、间隔时间为100ms、以传送速度9600bps或是19200bps读取或写入连续的软元件时，点数与数据通信的时间如下所示。

- 传送速度为9600bps时(单位:秒)

数据点数	站点数		
	1站	8站	16站
10点	0.3	1.9	3.7
32点	0.4	2.6	5.2
64点	0.5	3.7	7.3

- 传送速度为19200bps时(单位:秒)

数据点数	站点数		
	1站	8站	16站
10点	0.2	1.6	3.2
32点	0.3	2.0	3.9
64点	0.4	2.5	5.0

当读取或写入的软元件的种类增加时，时间为“上表中的数据通信时间×软元件的种类”。

此外，当读取或写入的点数超过64点时，通信次数也会增加这一超出部分。

因此，为了能够更加有效进行数据通信，建议尽量减少要通信的软元件种类，将要通信的软元件编号尽可能集中。

- \*1 在RS-485中使用FX-485PC-IF的1对接线的情况下，报文等待需要(每1次通信)70~150ms。  
在RS-485中使用2对接线或是RS-232C的情况下，报文等待为0ms。

# 3 报文格式

本章中，将说明对串行通信接口进行数据通信时报文的数据格式、数据的指定方法、限制等。

## 3.1 报文类型及用途

对于MC协议的报文，根据对应设备及用途，可以按以下方式进行分类。

### 形式及代码

MC协议的报文有5种形式。

形式	通信数据代码	备注	参照	FX5对应	GX Works3的设置
形式1	ASCII代码	—	15页 形式1	○	传文模式:模式1
形式2	ASCII代码	附加了块号的形式	—	×	—
形式3	ASCII代码	以STX、ETX围绕的形式	—	×	—
形式4	ASCII代码	末尾处附加了CR、LF的形式	16页 形式4	○	传文模式:模式4
形式5	二进制代码	只可以在4C帧中使用。	17页 形式5	○	传文模式:模式5

○：支持，×：不支持

在GX Works3的模块参数中设置形式。

#### 要点

对于通过二进制代码进行的通信，与通过ASCII代码的数据进行的通信相比，数据通信量将变为其一半左右，因此可以缩短通信时间。

### 帧

以下对用于外部设备以MC协议访问对应设备的帧(数据通信报文)的类型及各帧的用途有关内容进行说明。

MC协议(串行通信)的帧有如下类型。

帧	特点及用途	有兼容性的报文格式	形式	FX5对应
4C帧	可以从外部设备在最大的访问范围内进行访问。	MELSEC-QnA系列串行通信模块专用协议(QnA扩展帧)	形式1~5	○ 对应与形式1、4、5
3C帧	将通过4C帧访问路径的报文形式进行简略。	MELSEC-QnA系列串行通信模块专用协议(QnA帧)	形式1~4	○ 对应与形式1、4
2C帧	可以有效利用MELSEC-QnA系列可编程控制器用上创建的数据通信用软件。	MELSEC-QnA系列串行通信模块专用协议(QnA简易帧)		×
1C帧	是与使用FX3及MELSEC-A系列的计算机链接模块进行访问时相同的报文构成。 可以有效利用FX3及MELSEC-A系列可编程控制器用上创建的数据通信用软件。	MELSEC-A系列计算机链接模块专用协议		×

○：支持，×：不支持



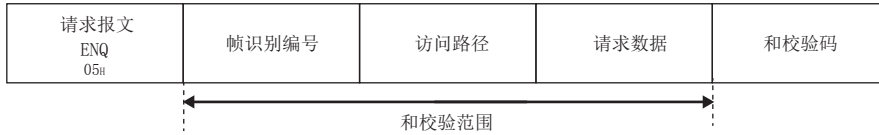
## 3.2 各形式的报文格式

各形式的报文格式及设置的数据如下所示。

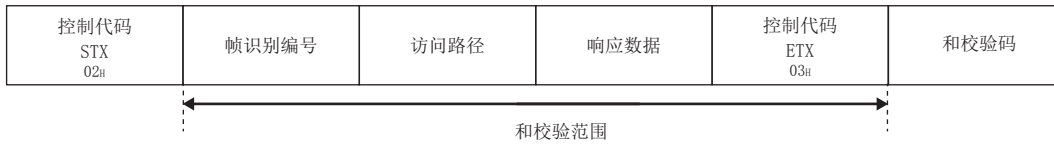
### 形式1

#### 报文格式

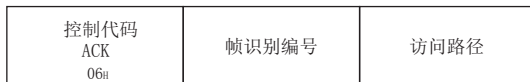
##### ■请求报文



##### ■响应报文(正常结束: 有响应数据)



##### ■响应报文(正常结束: 无响应数据)



##### ■响应报文(异常结束)



#### 设置的数据

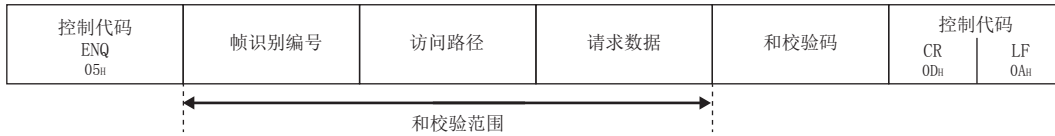
设置下述项目。

项目	内容	参照
控制代码 (ENQ、STX、ACK、NAK、ETX)	代码被定义为控制用。	18页 控制代码
帧识别编号	指定所使用的帧。	20页 帧识别编号
访问路径	指定访问路径。	23页 各帧的访问范围及设置的数据
请求数据	设置表示请求内容的指令等。请参阅各指令说明的“请求数据”。	28页 指令
响应数据	存储对于指令的读取数据等。请参阅各指令说明的“响应数据”。	
和校验码	是将和校验对象范围的数据视作二进制数据进行了加法运算结果的低位1字节(8位)的值。	20页 和校验码
出错代码	出错代码表示发生的出错内容。	22页 出错代码

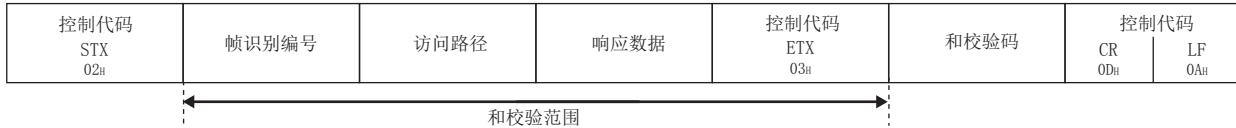
# 形式4

## 报文格式

### ■请求报文



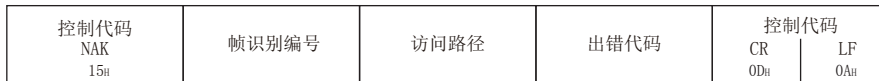
### ■响应报文(正常结束: 有响应数据)



### ■响应报文(正常结束: 无响应数据)



### ■响应报文(异常结束)



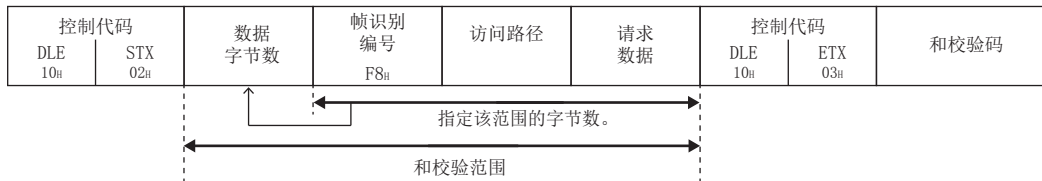
## 设置的数据

设置下述项目。

项目	内容	参照
控制代码 (ENQ、STX、ACK、NAK、ETX、CR、LF)	代码被定义为控制用。	18页 控制代码
帧识别编号	指定所使用的帧。	20页 帧识别编号
访问路径	指定访问路径。	23页 各帧的访问范围及设置的数据
请求数据	设置表示请求内容的指令等。请参阅各指令说明的“请求数据”。	28页 指令
响应数据	存储对于指令的读取数据等。请参阅各指令说明的“响应数据”。	
和校验码	是将和校验对象范围的数据视作二进制数据进行了加法运算结果的低位1字节(8位)的值。	20页 和校验码
出错代码	出错代码表示发生的出错内容。	22页 出错代码

## 报文格式

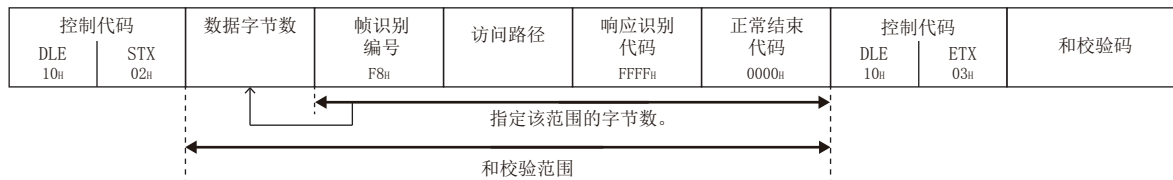
### ■请求报文



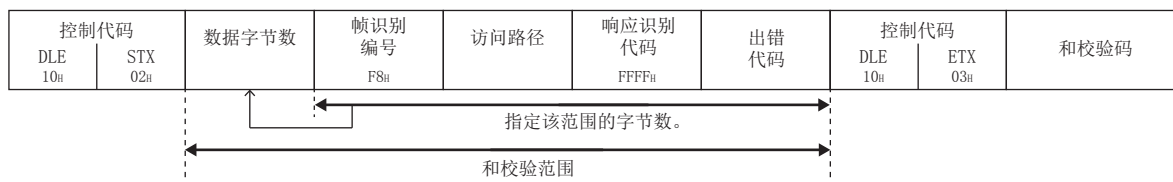
### ■响应报文(正常结束: 有响应数据)



### ■响应报文(正常结束: 无响应数据)



### ■响应报文(异常结束)



## 设置的数据

设置下述项目。

项目	内容	参照
数据字节数 (DLE, STX, ETX)	代码被定义为控制用。	19页 数据字节数
帧识别编号	是帧识别编号~控制代码(DLE、ETX)为止的字节数。	20页 帧识别编号
访问路径	指定访问路径。	23页 各帧的访问范围及设置的数据
请求数据	设置表示请求内容的指令等。请参阅各指令说明的“请求数据”。	28页 指令
响应数据	存储对于指令的读取数据等。请参阅各指令说明的“响应数据”。	
和校验码	是将和校验对象范围的数据视作二进制数据进行了加法运算结果的低位1字节(8位)的值。	20页 和校验码
响应识别代码	表示是响应报文。存储2字节的数值“FFFFH”。	—
正常结束代码	表示处理正常结束。存储2字节的数值“0000H”。	—
出错代码	出错代码表示发生的出错内容。	22页 出错代码

## 3.3 设置数据的详细内容(形式)

关于各报文中的共同数据项目，对内容及指定方法进行说明。

### 控制代码

控制代码是作为传送控制用具有特别意义的数据(报文的起始数据等)。

#### 在ASCII代码的报文(形式1~形式4)中使用的控制代码

在ASCII代码的报文(形式1~形式4)中使用的控制代码如下表所示。

符号名	内容	代码(16进制数)
STX	Start of Text	02H
ETX	End of Text	03H
EOT	End of Transmission	04H
ENQ	Enquiry	05H
ACK	Acknowledge	06H
LF	Line Feed	0AH
CL	Clear	0CH
CR	Carriage Return	0DH
NAK	Negative Acknowledge	15H

#### ■EOT(04H)、CL(0CH)

EOT、CL是用于对通过MC协议的ASCII代码的数据通信的传送顺控程序进行初始化，将CPU模块置为来自于外部设备的指令接收等待状态的代码。

使用二进制代码(形式5)时，通过指令(指令代码:1615)将传送顺控程序进行初始化。

在外部设备侧进行下述事项时，应根据使用的形式将EOT/CL从外部设备发送至CPU模块。

- 取消通过之前发送的指令的读取/写入请求。(发送了写入请求的情况下，已经将数据写入到CPU模块中时，将无法取消写入请求。)
- 在进行指令发送之前，应预先将CPU模块置为指令接收等待状态。
- 不能正常地进行数据通信时，应将CPU模块置为启动时的状态。

发送EOT、CL时的报文构成如下所示。

发送时只可以发送以下数据。不需要站号及可编程控制器编号等。

形式	EOT	CL												
形式1	<table border="1"><tr><td>EOT</td></tr><tr><td>04<sub>H</sub></td></tr></table>	EOT	04 <sub>H</sub>	<table border="1"><tr><td>CL</td></tr><tr><td>0C<sub>H</sub></td></tr></table>	CL	0C <sub>H</sub>								
EOT														
04 <sub>H</sub>														
CL														
0C <sub>H</sub>														
形式4	<table border="1"><tr><td>EOT</td><td>CR</td><td>LF</td></tr><tr><td>04<sub>H</sub></td><td>0D<sub>H</sub></td><td>0A<sub>H</sub></td></tr></table>	EOT	CR	LF	04 <sub>H</sub>	0D <sub>H</sub>	0A <sub>H</sub>	<table border="1"><tr><td>CL</td><td>CR</td><td>LF</td></tr><tr><td>0C<sub>H</sub></td><td>0D<sub>H</sub></td><td>0A<sub>H</sub></td></tr></table>	CL	CR	LF	0C <sub>H</sub>	0D <sub>H</sub>	0A <sub>H</sub>
EOT	CR	LF												
04 <sub>H</sub>	0D <sub>H</sub>	0A <sub>H</sub>												
CL	CR	LF												
0C <sub>H</sub>	0D <sub>H</sub>	0A <sub>H</sub>												

如果接收EOT/CL，CPU模块将变为如下所示。

- 在来自于外部设备的请求中，中止进行的读取/写入处理。在此情况下，不发送对于之前接收的指令的响应报文。
- MC协议传送顺控程序进行初始化后，置为来自于外部设备的指令接收等待状态。
- 没有对于EOT/CL接收的响应报文。(不发送任何信息至外部设备。)

## 在二进制代码的报文(形式5)中使用的控制代码

在二进制代码的报文(形式5)中使用的控制代码如下表所示。

符号名	内容	代码(16进制数)
STX	Start of Text	02H
ETX	End of Text	03H
DLE	Data Link Escape	10H

### ■附加代码(10H)

附加代码是在形式5中当设置的数据与控制代码DLE(10H)相同时,为了区别而附加的代码。

在请求报文的“数据字节数”~“请求数据”中包含有“10H”的情况下,将附加代码“10H”附加到其之前。

在响应报文的“数据字节数”~“响应数据”中有“10H”的情况下,附加代码“10H”被附加。

(以“10H”→“10H”+“10H”进行发送接收。)

#### 要点

计算出下述值的情况下,应除去附加代码后进行计算。

- 数据字节数(形式5的设置项目)
- 和校验码

## 数据字节数

数据字节数表示从帧识别编号起至控制数据之前为止的数据的总字节数。

### 范围

关于从帧识别编号起至DLE(10H)之前为止的范围,将除去附加代码后进行计算。(☞ 19页 附加代码(10H))



### 设置方法

以二进制代码(形式5)进行数据通信时设置。

将2字节的数值从低位字节(L:位0~7)进行发送。

#### 例

响应报文(正常结束:有响应数据)的情况下

- 帧识别编号:1字节
- 访问路径:7字节
- 响应识别代码、正常结束代码:4字节
- 响应数据:2字节+附加代码(10H)1字节

数据字节数=1+7+2=14(0EH)



## 块号

块号是在外部设备侧添加了意义的任意的编号，用于数据整理等。

对于块号，在00H~FFH范围内转换为ASCII代码2位(16进制数)后，从高位进行发送。

CPU模块只对块号是否指定在正确范围内进行检查。不检查块号是否按顺序被发送。

## 帧识别编号

指定所使用的帧。

类型	设置值
4C帧	F8
3C帧	F9

### 设置方法

#### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

#### ■以二进制代码进行数据通信时

发送1字节的数值。

#### 例

4C帧(F8)的情况下

ASCII代码	二进制代码					
<table border="1"><tr><td>F</td><td>8</td></tr><tr><td>46<sub>Hi</sub></td><td>38<sub>Hi</sub></td></tr></table>	F	8	46 <sub>Hi</sub>	38 <sub>Hi</sub>	<table border="1"><tr><td>F8<sub>Hi</sub></td></tr></table>	F8 <sub>Hi</sub>
F	8					
46 <sub>Hi</sub>	38 <sub>Hi</sub>					
F8 <sub>Hi</sub>						

## 和校验码

进行和校验的情况下设置。

对于和校验码，为了检测出出错，设置从和校验范围的数据计算出的值。

### 和校验

和校验是在传送过程中发生了数据变化时，用于检测出出错的功能。

通过工程工具设置和校验的有无。

#### ■以有和校验进行了设置的情况下

应将和校验码附加到请求报文上。

CPU模块对和校验码进行检查。响应报文上附加和校验码。

#### ■以无和校验进行了设置的情况下

请求报文上不需要和校验码。

CPU模块不对和校验码进行检查。响应报文上无法附加和校验码。

## 和校验范围

表示各报文形式的和校验范围。

形式	报文构成	参照
形式1		15页 形式1
形式4		16页 形式4
形式5		17页 形式5

## 和校验码的算出

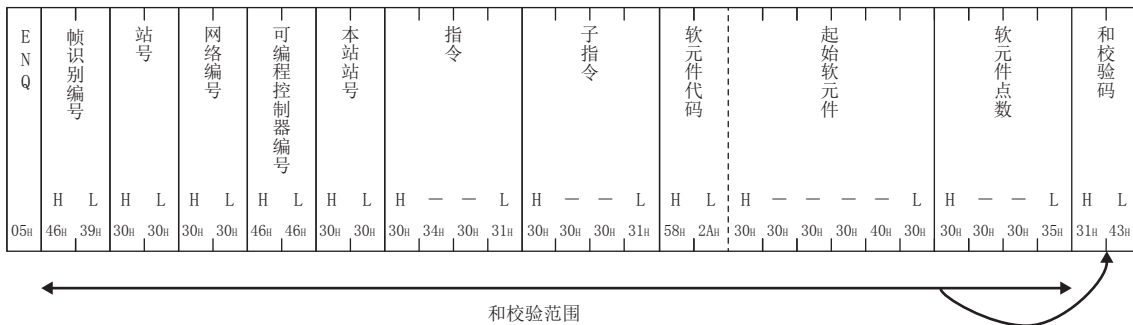
对于和校验码设置将和校验范围的数据作为二进制数据进行了加法运算后的结果(和)的低位1字节(8位)的数值。除去附加代码后进行计算。(☞ 19页 附加代码(10H))

### 例

3C帧形式1的下述情况，和校验码将变为“1C”。

计算公式： $46H+39H+30H+30H+30H+30H+46H+46H+30H+30H+30H+34H+30H+31H+30H+30H+30H+31H+58H+2AH+30H+30H+30H+30H+34H+30H+30H+30H+30H+35H=61CH$

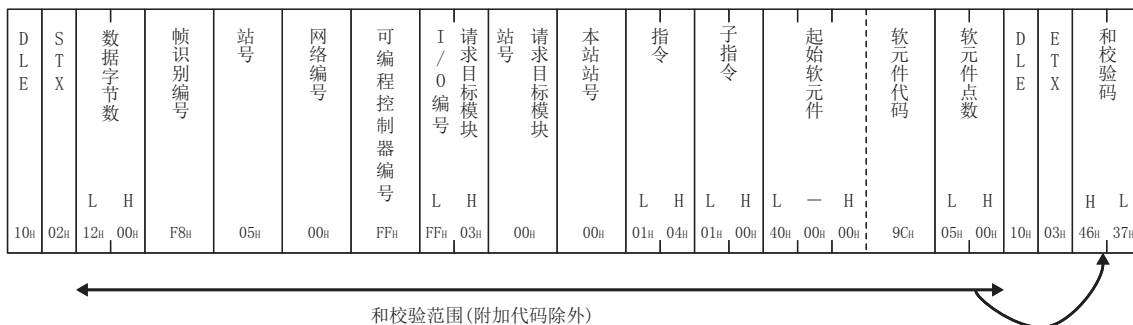
和校验码：“1C” (ASCII代码31H、43H)



4C帧形式5的下述情况，和校验码将变为“F7”。

计算公式： $12H+00H+F8H+05H+00H+FFH+FFH+03H+00H+00H+01H+04H+01H+00H+40H+00H+00H+9CH+05H+00H=3F7H$

和校验码：“F7” (ASCII代码46H、37H)



## 设置方法

### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

### ■以二进制代码进行数据通信时

与通过ASCII代码进行数据通信时相同，使用转换为ASCII代码2位(16进制数)的数值。  
将2字节的数值从高位字节(H:位8~15)进行发送。

#### 例

和校验码：“F7”(ASCII代码46H、37H)的情况下

ASCII代码、二进制代码

F	7
46H	37H

## 出错代码

出错代码表示发生的出错内容。

同时发生了多个出错的情况下，将返回最初检测的出错代码。

关于出错代码的内容及处理方法，请参考下列手册。

📖 MELSEC iQ-F FX5用户手册(串行通信篇)

📖 MELSEC iQ-F FX5用户手册(应用篇)

## 设置方法

### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码4位(16进制数)后从高位进行发送。

### ■以二进制代码进行数据通信时

将2字节的数值从低位字节(L:位0~7)进行发送。

#### 例

出错代码7143H的情况下

ASCII代码	二进制代码										
<table border="1"><tr><td>7</td><td>1</td><td>4</td><td>3</td></tr><tr><td>37H</td><td>31H</td><td>34H</td><td>33H</td></tr></table>	7	1	4	3	37H	31H	34H	33H	<table border="1"><tr><td>43H</td><td>71H</td></tr></table>	43H	71H
7	1	4	3								
37H	31H	34H	33H								
43H	71H										



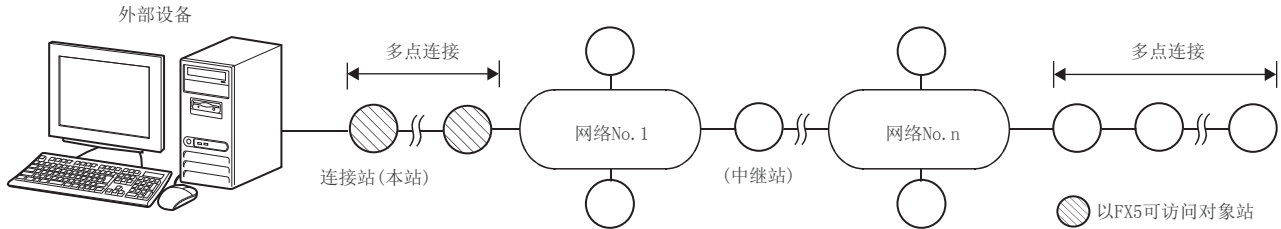
## 3.4 各帧的访问范围及设置的数据

各帧的访问范围及设置访问路径的数据项目如下所示。

### 4C帧

#### 4C帧的访问范围

可以访问下述范围。



#### 报文格式(连接站(本站)访问时的设置示例)

##### ■以ASCII代码进行数据通信时(形式1、形式4)

站号	网络编号	可编程控制器编号	请求目标模块I/O编号	请求目标模块站号	本站站号
0 0	0 0	F F	0 3 F F	0 0	0 0
30h, 30h	30h, 30h	46h, 46h	30h, 33h, 46h, 46h	30h, 30h	30h, 30h

##### ■以二进制代码进行数据通信时(形式5)

网络编号	请求目标模块I/O编号	请求目标模块站号
站号	可编程控制器编号	本站站号
00h	00h FFh FFh	03h 00h 00h

#### 设置的数据

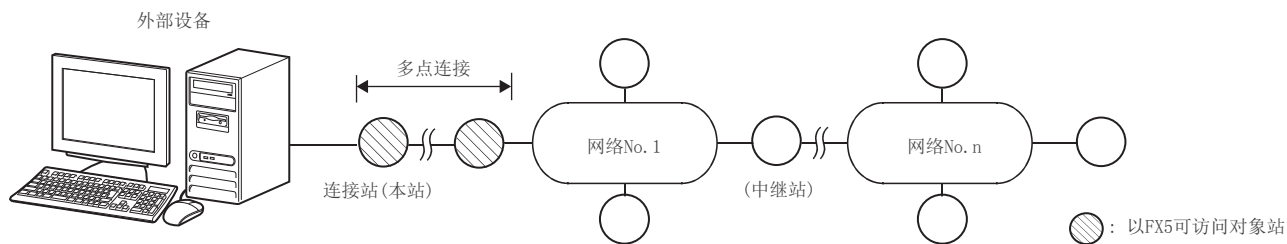
设置下述项目。

项目	内容	参照
站号	指定从外部设备连接的站。	25页 站号
网络编号	指定访问目标的网络编号。	26页 网络编号、可编程控制器编号
可编程控制器编号	指定访问目标的网络站号。	
请求目标模块I/O编号	对经由了网络的多点连接的连接源模块(中继站)的起始I/O No. 进行指定。	27页 请求目标模块I/O编号、请求目标模块站号
请求目标模块站号	对经由了网络的多点连接的访问目标模块的站号进行指定。	
本站站号	在m:n的多点连接时, 指定请求源外部设备的站号。	27页 本站站号

# 3C帧

## 3C帧的访问范围

可以访问下述范围。



## 报文格式(连接站(本站)访问时的设置示例)

站号	网络编号	可编程控制器编号	本站站号
0 0	0 0	F F	0 0
30 <sub>H</sub> , 30 <sub>H</sub>	30 <sub>H</sub> , 30 <sub>H</sub>	46 <sub>H</sub> , 46 <sub>H</sub>	30 <sub>H</sub> , 30 <sub>H</sub>

## 设置的数据

设置下述项目。

项目	内容	参照
站号	指定从外部设备连接的站。	25页 站号
网络编号	指定访问目标的网络编号。	26页 网络编号、可编程控制器编号
可编程控制器编号	指定访问目标的网络站号。	
本站站号	在m: n的多点连接时，指定请求源外部设备的站号。	27页 本站站号

## 3.5 设置数据的详细内容(帧)

关于设置访问路径的数据项目，对内容及指定方法进行说明。

项目	4C帧	3C帧	参照
站号	○	○	25页 站号
网络编号	○	○	26页 网络编号、可编程控制器编号
可编程控制器编号			
请求目标模块I/O编号	○	—	27页 请求目标模块I/O编号、请求目标模块站号
请求目标模块站号			
本站站号	○	○	27页 本站站号

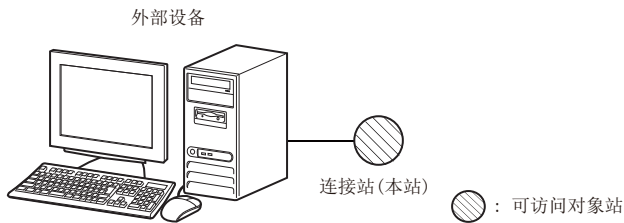
○：需要设置，—：无需设置

### 站号

指定从外部设备进行访问的站。

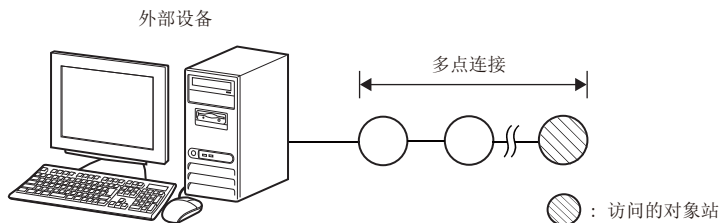
#### 访问连接站(本站)的情况下

访问连接站(本站)的情况下，将指定0。



#### 访问多点连接站的情况下

进行了多点连接的情况下，指定进行访问站的站号0~31 (00H~1FH)。



### 设置方法

站号通过工程工具的下述参数项目指定，将“模块参数”写入到CPU模块中。

- GX Works3：“模块参数”的“站号设置”

#### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

#### ■以二进制代码进行数据通信时

发送1字节的数值。

#### 例

访问的CPU模块站号设置为“5”的情况下

ASCII代码	二进制代码					
<table border="1"> <tr> <td>0</td> <td>5</td> </tr> <tr> <td>30H</td> <td>35H</td> </tr> </table>	0	5	30H	35H	<table border="1"> <tr> <td>05H</td> </tr> </table>	05H
0	5					
30H	35H					
05H						

# 网络编号、可编程控制器编号

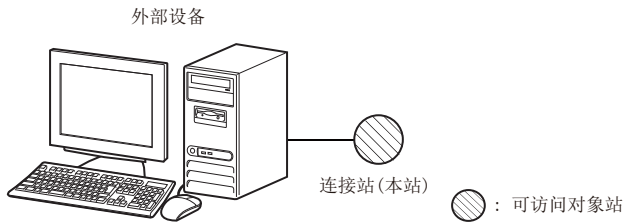
指定在访问目标的网络模块的参数中设置的网络No. 及站号。  
访问连接站的情况下，将指定固定值。

## 要点

网络编号应以如下所示的值进行设置。  
指定了不正确的值时有可能不会返回响应。

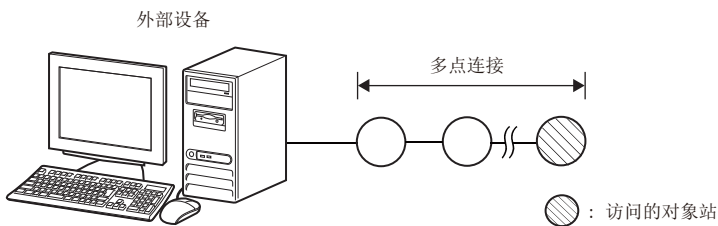
### 访问连接站(本站)的情况下

指定网络编号0、可编程控制器编号FF。



### 访问多点连接站的情况下

指定网络编号0、可编程控制器编号FF。



## 设置方法

### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

### ■以二进制代码进行数据通信时

发送1字节的数值。

### 例

访问连接站(本站)、多点连接站的情况下

ASCII代码		二进制代码	
网络编号	可编程控制器编号	网络编号	可编程控制器编号
0 0	F F	00h	FFh
30h   30h	46h   46h		

## 请求目标模块I/O编号、请求目标模块站号

在FX5 CPU模块指定以下固定值。

请求目标模块I/O编号	请求目标模块站号
03FFH	00H

### 设置方法

#### ■以ASCII代码进行数据通信时

对于请求目标模块I/O编号，将数值转换为ASCII代码4位(16进制数)后从高位进行发送。

对于请求目标模块站号，将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

#### ■以二进制代码进行数据通信时

对于请求目标模块I/O编号，将2字节的数值从低位字节(L:位0~7)进行发送。

对于请求目标模块站号，发送1字节的数值。

#### 例

访问连接站(本站)的情况下

ASCII代码	二进制代码
<p>请求目标模块 I/O编号      请求目标模块站号</p> <p>0 3 F F    0 0</p> <p>30h   33h   46h   46h   30h   30h</p>	<p>请求目标模块 I/O编号      请求目标模块站号</p> <p>FFh   03h    00h</p>

## 本站站号

在FX5 CPU模块指定以下固定值。

本站站号
00H

### 设置方法

#### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

#### ■以二进制代码进行数据通信时

发送1字节的数值。

#### 例

00H指定的情况下

ASCII代码	二进制代码
<p>0 0</p> <p>30h   30h</p>	<p>00h</p>

# 4 指令

本章对MC协议指令有关内容进行说明。

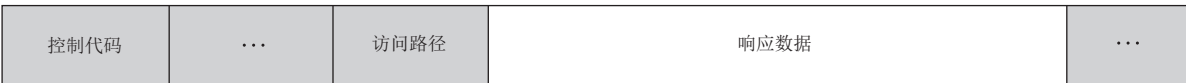
## 4.1 指令和功能一览

报文的功能在每个指令被定义。请求数据与响应数据的报文格式根据指令而有所不同。对于指令，根据所使用帧的类型分配固有的值。指令的值以请求数据的起始进行指定。

请求报文



响应报文



各指令的说明中，说明请求数据与响应数据部分的报文格式。

关于请求数据与响应数据部分以外的报文格式，请参阅下述内容。

☞ 14页 报文格式

## 指令一览

### 3C/4C帧

名称	指令	子指令	处理内容	1次通信可以处理的点数
批量读取	0401H	0001H	以1位为单位，从位软件或字软件读取数据。	ASCII:3584点 BIN:3584点
		0000H	<ul style="list-style-type: none"> <li>以16位为单位，从位软件读取数据。</li> <li>以1个字为单位，从字软件读取数据。</li> </ul>	960字(15360点)
		0081H	<ul style="list-style-type: none"> <li>以1位为单位，从智能模块的缓冲存储区读取数据。</li> <li>以1位为单位，从使用变址寄存器间接指定的软件读取数据。</li> </ul>	ASCII:3584点 BIN:3584点
		0080H	<ul style="list-style-type: none"> <li>以1个字为单位，从智能模块的缓冲存储区读取数据。</li> <li>以1个字为单位，从使用变址寄存器间接指定的软件读取数据。</li> </ul>	960字(15360点)
		0083H	<ul style="list-style-type: none"> <li>以1位为单位，从智能模块的缓冲存储区读取数据。</li> <li>以1位为单位，从使用变址寄存器间接指定的软件读取数据。</li> </ul>	ASCII:3584点 BIN:3584点
		0082H	<ul style="list-style-type: none"> <li>以1个字为单位，从智能模块的缓冲存储区读取数据。</li> <li>以1个字为单位，从使用变址寄存器间接指定的软件读取数据。</li> </ul>	960字(15360点)
批量写入	1401H	0001H	以1位为单位，向位软件写入数据。	ASCII:3584点 BIN:3584点
		0000H	<ul style="list-style-type: none"> <li>以16位为单位，向位软件写入数据。</li> <li>以1个字为单位，向字软件写入数据。</li> </ul>	960字(15360点)
		0081H	<ul style="list-style-type: none"> <li>以1位为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。</li> <li>使用变址寄存器间接指定位软件、字软件、缓冲存储区。</li> </ul>	ASCII:3584点 BIN:3584点
		0080H	以1个字(16位)为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。	960字(15360点)
		0083H	以1位为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。	ASCII:3584点 BIN:3584点
		0082H	以1个字(16位)为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。	960字(15360点)
随机读取	0403H	0000H	随机指定软件编号，以1个字为单位或是以2个字为单位读取字软件。	192点
		0080H	以1个字(16位)为单位，从支持MC协议的设备或智能模块的缓冲存储区读取数据。	192点
		0082H	以1个字(16位)为单位，从支持MC协议的设备或智能模块的缓冲存储区读取数据。	192点

名称	指令	子指令	处理内容	1次通信可以处理的点数
随机写入	1402H	0001H	随机指定软元件编号，以1位为单位向位软元件写入数据。	188点
		0000H	<ul style="list-style-type: none"> <li>随机指定软元件编号，以16位为单位向位软元件写入数据。</li> <li>随机指定软元件编号，以1个字或是2个字为单位，向字软元件写入数据。</li> </ul>	(字访问点数)×12+ (双字访问点数)×14≤1920
		0081H	<ul style="list-style-type: none"> <li>以1位为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。</li> <li>使用变址寄存器间接指定缓冲存储区。</li> </ul>	188点
		0080H	以1个字(16位)或是2个字为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。	(字访问点数)×12+ (双字访问点数)×14≤1920*1
		0083H	以1位为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。	188点
		0082H	以1个字(16位)为单位或是以2个字为单位，向支持MC协议的设备或智能模块的缓冲存储区写入数据。	(字访问点数)×12+ (双字访问点数)×14≤1920*1
多个块批量读取	0406H	0000H	以n点位软元件或字软元件为1模块，随机指定多个模块，读取数据。(指定位软元件时，1点中以16位为对象。)	960点
		0080H	以n点支持MC协议的设备或智能模块的缓冲存储区为1模块，随机指定多个模块，读取数据。(指定位软元件时，1点中以16位为对象。)	960点
		0082H	以n点支持MC协议的设备或智能模块的缓冲存储区为1模块，随机指定多个模块，读取数据。	960点
多个块批量写入	1406H	0000H	以n点位软元件或字软元件为1模块，随机指定多个模块，写入数据。(指定位软元件时，1点中以16位为对象。)	770点
		0080H	以n点支持MC协议的设备或智能模块的缓冲存储区为1模块，随机指定多个模块，写入数据。(指定位软元件时，1点中以16位为对象。)	770点*1
		0082H	以n点支持MC协议的设备或智能模块的缓冲存储区为1模块，随机指定多个模块，写入数据。	770点*1
远程RUN	1001H	0000H	对设备执行远程RUN请求。	—
远程STOP	1002H	0000H	对设备执行远程STOP请求。	—
远程PAUSE	1003H	0000H	对设备执行远程PAUSE请求。	—
远程锁存清除	1005H	0000H	设备处于STOP状态时，执行远程锁存清除请求。	—
远程RESET	1006H	0000H	为了解除设备的错误停止状态，执行远程RESET请求。	—
CPU型号读取	0101H	0000H	读取设备的处理器模块名称代码(处理器类型)。	—
反复测试	0619H	0000H	确认能否正常通信。	—
显示LED的炮灯、出错代码的初始化	1617H	0000H	成批解除所有错误，使LED灯灭。	—

\*1 由于软元件的扩展指定，可以设定的点数变小。有软元件内存的扩展指定时，访问点数按2倍计算。

## 4.2 软元件访问

以下说明实施软元件存储器的读取、写入的情况下的控制步骤的指定内容以及指定示例。

### 在指令内指定的数据

#### 子指令

用于指定读取/写入的单位、指定的软元件的种类、读取数据的条件等的的数据。

设定项目的详情如下所述。

设定项目	内容	
数据大小指定	字单位	<ul style="list-style-type: none"> <li>将作为读取/写入对象的数据规定为字单位。</li> <li>没有读取/写入数据的情况下，指令的参数中也选择“0”。</li> </ul>
	位单位	将作为读取/写入对象的数据规定为位单位。
软元件指定形式	指定代码2字符/编号6位	以下述大小表现与数据、地址指定有关的项目。 <ul style="list-style-type: none"> <li>软元件代码在ASCII代码时为2位，在二进制时为1字节</li> <li>软元件编号在ASCII代码时为6位，在二进制时为3字节</li> </ul>
	指定代码4字符/编号8位	放大至下述大小，表现与数据、地址指定有关的项目。 <ul style="list-style-type: none"> <li>软元件代码在ASCII代码时为4位，在二进制时为2字节</li> <li>软元件编号在ASCII代码时为8位，在二进制时为4字节</li> </ul>
软元件存储器扩展指定	无指定	CPU模块的软元件指定时进行设置。 *不使用软元件存储器扩展指定的情况下，无指定。
	有指定	<ul style="list-style-type: none"> <li>智能功能模块的缓冲存储器指定时进行设置。</li> <li>对应利用变址寄存器的软元件的间接指定。</li> </ul>

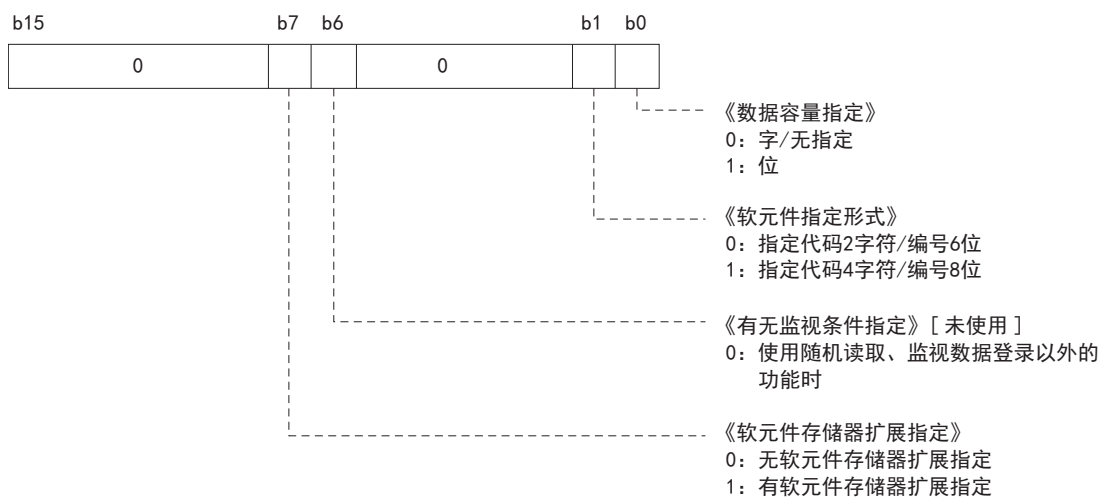
①以ASCII代码进行数据通信时

0000H(0)或下述③数值转换为ASCII代码4位（16进制数）后使用，并从高位（“0”）开始发送。

②以二进制代码进行数据通信时

使用0000H或下述③2字节的数值并发送。

③子指令的指定内容如下所示。



④以下情况下，子指令为0000H/0001H。

- 选择监视条件无指定以及软元件存储器无扩展指定时。
- 使用无法选择监视条件指定和软元件存储器扩展指定的指令时。



## 软元件

以软元件代码及软元件编号指定访问的软元件。

- 根据ASCII代码及二进制代码，数据排列有所不同。
- 根据子指令的软元件指定形式的设置，设置的数据容量有所不同。

子指令类型	ASCII代码	二进制代码
指定代码2字符/编号6位	软元件 代码                  软元件编号  (2位)                  (6位)	软元件 编号    软元件 代码  (3字节)    (1字节)
指定代码4字符/编号8位	软元件代码                  软元件编号  (4位)                  (8位)	软元件编号    软元件代码  (4字节)    (2字节)

### 要点

访问下述软元件的情况下，使用软元件扩展指定(子指令：008□)。

- 模块访问软元件

软元件扩展指定时的报文格式请参阅下述内容。

☞ 83页 软元件存储器的扩展指定

## 软元件代码

指定访问的软元件名。

应指定具有访问目标模块的软元件。

各软元件的软元件代码值，请参阅下述内容。

☞ 33页 软元件代码一览

### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位或4位(16进制数)后从高位进行发送。

- 指定代码2字符/编号6位：ASCII代码2位
- 指定代码4字符/编号8位：ASCII代码4位

软元件代码中的“\*”也可以通过空格(代码：20H)指定。

### ■以二进制代码进行数据通信时

将1字节或2字节的数值从低位字节(L:位0~7)进行发送。

- 指定代码2字符/编号6位：1字节
- 指定代码4字符/编号8位：2字节

### 例

输入(X)的情况下

子指令类型	ASCII代码	二进制代码
指定代码2字符/编号6位	 X * 58h 2Ah	 9Ch
指定代码4字符/编号8位	 X * * * 58h 2Ah 2Ah 2Ah	 9Ch 00h

## 软元件编号

指定访问的软元件编号。

软元件编号应在具有访问目标模块的软元件编号的范围内进行指定。

### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码6位或8位后从高位进行发送。

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。(☞ 33页 软元件代码一览)

- 指定代码2字符/编号6位: ASCII代码6位
- 指定代码4字符/编号8位: ASCII代码8位(软元件扩展指定时为10位)

高位位的“0”也可以通过空格(代码: 20H)指定。

### ■以二进制代码进行数据通信时

将3字节或4字节的数值从低位字节(L:位0~7)进行发送。

对于软元件编号为10进制数的软元件,转换为16进制数进行指定。

- 指定代码2字符/编号6位: 3字节\*1
- 指定代码4字符/编号8位: 4字节\*1

\*1 有可能会添加附加代码。(☞ 19页 附加代码(10H))

#### 例

链接继电器(B)1234的情况下(软元件编号为16进制数的软元件的情况下)

子指令类型	ASCII代码	二进制代码																				
指定代码2字符/编号6位	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>30h</td><td>30h</td><td>31h</td><td>32h</td><td>33h</td><td>34h</td></tr> </table>	0	0	1	2	3	4	30h	30h	31h	32h	33h	34h	<table border="1"> <tr><td>34h</td><td>12h</td><td>00h</td></tr> </table>	34h	12h	00h					
0	0	1	2	3	4																	
30h	30h	31h	32h	33h	34h																	
34h	12h	00h																				
指定代码4字符/编号8位	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>30h</td><td>30h</td><td>30h</td><td>30h</td><td>31h</td><td>32h</td><td>33h</td><td>34h</td></tr> </table>	0	0	0	0	1	2	3	4	30h	30h	30h	30h	31h	32h	33h	34h	<table border="1"> <tr><td>34h</td><td>12h</td><td>00h</td><td>00h</td></tr> </table>	34h	12h	00h	00h
0	0	0	0	1	2	3	4															
30h	30h	30h	30h	31h	32h	33h	34h															
34h	12h	00h	00h																			

内部继电器(M)1234的情况下(软元件编号为10进制数的软元件的情况下)

二进制代码时,将软元件编号转换为16进制数。“1234”(10进制)→“4D2”(16进制)

子指令类型	ASCII代码	二进制代码																				
指定代码2字符/编号6位	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>30h</td><td>30h</td><td>31h</td><td>32h</td><td>33h</td><td>34h</td></tr> </table>	0	0	1	2	3	4	30h	30h	31h	32h	33h	34h	<table border="1"> <tr><td>D2h</td><td>04h</td><td>00h</td></tr> </table>	D2h	04h	00h					
0	0	1	2	3	4																	
30h	30h	31h	32h	33h	34h																	
D2h	04h	00h																				
指定代码4字符/编号8位	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>30h</td><td>30h</td><td>30h</td><td>30h</td><td>31h</td><td>32h</td><td>33h</td><td>34h</td></tr> </table>	0	0	0	0	1	2	3	4	30h	30h	30h	30h	31h	32h	33h	34h	<table border="1"> <tr><td>D2h</td><td>04h</td><td>00h</td><td>00h</td></tr> </table>	D2h	04h	00h	00h
0	0	0	0	1	2	3	4															
30h	30h	30h	30h	31h	32h	33h	34h															
D2h	04h	00h	00h																			

内部继电器(M)16的情况下(附加代码进入的情况下)

二进制代码时,10H以10H+10H进行指定。(☞ 19页 附加代码(10H))

子指令类型	二进制代码				
指定代码2字符/编号6位	<table border="1"> <tr><td>DLE</td><td></td></tr> <tr><td>10h</td><td>10h   00h   00h</td></tr> </table>	DLE		10h	10h   00h   00h
DLE					
10h	10h   00h   00h				
指定代码4字符/编号8位	<table border="1"> <tr><td>DLE</td><td></td></tr> <tr><td>10h</td><td>10h   00h   00h   00h</td></tr> </table>	DLE		10h	10h   00h   00h   00h
DLE					
10h	10h   00h   00h   00h				

# 软元件代码一览

关于MC协议的通信功能所使用的指令中，可以处理的软元件以及软元件编号范围如下所示。

请对执行数据的读出、写入等的对象模块指定所存在的软元件、软元件编号范围。

3C/4C帧通过下列“软元件代码”指定要访问的软元件。

分类	软元件		种类	软元件代码*1 (软元件指定格式:长)		软元件编号	FX5有无软元件*2	
				ASCII码	二进制码			
内部用户软元件	输入		位	X*(X***)	9CH (9C00H)	在要访问的模块所具有的软元件编号范围中指定。	8进制	○
	输出			Y*(Y***)	9DH (9D00H)		8进制	○
	内部继电器			M*(M***)	90H (9000H)		10进制	○
	锁存继电器			L*(L***)	92H (9200H)		10进制	○
	报警继电器			F*(F***)	93H (9300H)		10进制	○
	边沿继电器			V*(V***)	94H (9400H)		10进制	—
	链接继电器			B*(B***)	A0H (A000H)		16进制	○
	步进继电器			S*(S***)	98H (9800H)		10进制	○
	数据寄存器			字	D*(D***)		A8H (A800H)	10进制
	链接寄存器		W*(W***)		B4H (B400H)	16进制	○	
	定时器	触点	位	TS (TS**)	C1H (C100H)	10进制	○	
				TC (TC**)	C0H (C000H)		○	
		当前值	字	TN (TN**)	C2H (C200H)		○	
	长时间定时器	触点	位	— (LTS*)	51H (5100H)	10进制	—	
				— (LTC*)	50H (5000H)		—	
		当前值	双字	— (LTN*)	52H (5200H)		—	
	累计定时器	触点	位	SS (STS*)	C7H (C700H)	10进制	○	
				SC (STC*)	C6H (C600H)		○	
		当前值	字	SN (STN*)	C8H (C800H)		○	
	长时间累计定时器	触点	位	— (LSTS)	59H (5900H)	10进制	—	
				— (LSTC)	58H (5800H)		—	
		当前值	双字	— (LSTN)	5AH (5A00H)		—	
	计数器	触点	位	CS (CS**)	C4H (C400H)	10进制	○	
				CC (CC**)	C3H (C300H)		○	
		当前值	字	CN (CN**)	C5H (C500H)		○	
	长计数器	触点	位	— (LCS*)	55H (5500H)	10进制	○	
				— (LCC*)	54H (5400H)		○	
当前值		双字	— (LCN*)	56H (5600H)	○			
链接特殊继电器		位	SB (SB**)	A1H (A100H)	16进制	○		
链接特殊寄存器		字	SW (SW**)	B5H (B500H)	16进制	○		
系统软元件	特殊继电器		位	SM (SM**)	91H (9100H)	在要访问的模块所具有的软元件编号范围中指定。	10进制	○
	特殊寄存器		字	SD (SD**)	A9H (A900H)		10进制	○
	功能输入		位	—	—		16进制	—
	功能输出		位	—	—		16进制	—
	功能寄存器		字	—	—		10进制	—
变址寄存器			16位	Z*(Z***)	CCH (CC00H)	在要访问的模块所具有的软元件编号范围中指定。	10进制	○
			32位	LZ (LZ**)	62H (6200H)		10进制	○
文件寄存器			字	R*(R***)	AFH (AF00H)	10进制	○	
				ZR (ZR**)	B0H (B000H)		10进制	—
模块访问软元件*3	链接寄存器		字	W*(W***)	B4H (B400H)	16进制	—	
	链接特殊寄存器			SW (SW**)	B5H (B500H)		16进制	—
	模块访问软元件			G*(G***)	ABH (AB00H)		10进制	○

\*1 【ASCII码】

“软元件代码”不满指定字符数时，在软元件代码后附加“\*”（ASCII码:2AH）或者<空格>（ASCII码:20H）。

【二进制码】

“软元件代码”不满指定大小时，在软元件代码后附加“00H”。

\*2 ○:FX5中有软元件

—:FX5中没有软元件

\*3 需要将子指令的“软元件内存的扩展指定”置ON(1)。

## 软元件点数

指定进行读取或写入的软元件点数。

### 设置方法

#### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码4位(16进制数)后从高位进行发送。

指定英文字母的情况下，使用大写字母代码。

#### ■以二进制代码进行数据通信时

将2字节\*1的数值从低位字节(L:位0~7)进行发送。

\*1 有可能会添加附加代码。(☞ 19页 附加代码(10H))

#### 例

5点、20点的情况下

软元件点数	ASCII代码	二进制代码										
5点	<table border="1"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>5</td> </tr> <tr> <td>30H</td> <td>30H</td> <td>30H</td> <td>35H</td> </tr> </table>	0	0	0	5	30H	30H	30H	35H	<table border="1"> <tr> <td>05H</td> <td>00H</td> </tr> </table>	05H	00H
0	0	0	5									
30H	30H	30H	35H									
05H	00H											
20点	<table border="1"> <tr> <td>0</td> <td>0</td> <td>1</td> <td>4</td> </tr> <tr> <td>30H</td> <td>30H</td> <td>31H</td> <td>34H</td> </tr> </table>	0	0	1	4	30H	30H	31H	34H	<table border="1"> <tr> <td>14H</td> <td>00H</td> </tr> </table>	14H	00H
0	0	1	4									
30H	30H	31H	34H									
14H	00H											

## 访问点数

指定以字单位、双字单位或位单位进行访问的软件元件点数。

在指令一览表（☞ 28页）所示的1次通信可处理的点数以内进行指定。

### 设置方法

#### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

指定英文字母的情况下，使用大写字母代码。

#### ■以二进制代码进行数据通信时

发送1字节\*1的数值(16进制数)。

\*1 有可能会添加附加代码。（☞ 19页 附加代码(10H)）

#### 例

5点、20点的情况下

软件元件点数	ASCII代码	二进制代码					
5点	<table border="1"> <tr> <td>0</td> <td>5</td> </tr> <tr> <td>30<sub>H</sub></td> <td>35<sub>H</sub></td> </tr> </table>	0	5	30 <sub>H</sub>	35 <sub>H</sub>	<table border="1"> <tr> <td>05<sub>H</sub></td> </tr> </table>	05 <sub>H</sub>
0	5						
30 <sub>H</sub>	35 <sub>H</sub>						
05 <sub>H</sub>							
20点	<table border="1"> <tr> <td>1</td> <td>4</td> </tr> <tr> <td>31<sub>H</sub></td> <td>34<sub>H</sub></td> </tr> </table>	1	4	31 <sub>H</sub>	34 <sub>H</sub>	<table border="1"> <tr> <td>14<sub>H</sub></td> </tr> </table>	14 <sub>H</sub>
1	4						
31 <sub>H</sub>	34 <sub>H</sub>						
14 <sub>H</sub>							

### 位访问点数

指定以位单位进行访问的软件元件点数。

### 字访问点数、双字访问点数

指定以字单位或双字单位进行访问的软件元件点数。

## 块数

以16进制数指定访问的软元件块数。

在以下范围内设置各块数。

- 字软元件块数+位软元件块数≤120

### 要点

以下情况下，作为块数×2计算。

- 通过软元件扩展指定(子指令：008□)访问

## 设置方法

### ■以ASCII代码进行数据通信时

将数值转换为ASCII代码2位(16进制数)后从高位进行发送。

指定英文字母的情况下，使用大写字母代码。

### ■以二进制代码进行数据通信时

发送1字节\*1的数值(16进制数)。

\*1 有可能会添加附加代码。(☞ 19页 附加代码(10H))

### 例

5点、20点的情况下

软元件点数	ASCII代码	二进制代码					
5点	<table border="1"><tr><td>0</td><td>5</td></tr><tr><td>30<sub>H</sub></td><td>35<sub>H</sub></td></tr></table>	0	5	30 <sub>H</sub>	35 <sub>H</sub>	<table border="1"><tr><td>05<sub>H</sub></td></tr></table>	05 <sub>H</sub>
0	5						
30 <sub>H</sub>	35 <sub>H</sub>						
05 <sub>H</sub>							
20点	<table border="1"><tr><td>1</td><td>4</td></tr><tr><td>31<sub>H</sub></td><td>34<sub>H</sub></td></tr></table>	1	4	31 <sub>H</sub>	34 <sub>H</sub>	<table border="1"><tr><td>14<sub>H</sub></td></tr></table>	14 <sub>H</sub>
1	4						
31 <sub>H</sub>	34 <sub>H</sub>						
14 <sub>H</sub>							

## 字软元件块数

指定字软元件的块数。

## 位软元件块数

指定位软元件的块数。

## 读取数据、写入数据

进行读取的情况下，存储读取的软元件的值。进行写入的情况下，存储写入的数据。  
根据是位单位还是字单位，数据的排列有所不同。

### 位单位的情况下

以位单位读取、写入数据时的数据有关内容如下所示。

#### ■以ASCII代码进行数据通信时

以ASCII代码1位表示各软元件的ON/OFF。

- ON的情况下：“1” (31H)
- OFF的情况下：“0” (30H)

#### ■以二进制代码进行数据通信时

将1点以4位表示，且以4位表示各软元件的ON/OFF。

- ON的情况下：“1”
- OFF的情况下：“0”

点数为奇数的情况下，最低位的4位置为“0”。

#### 例

显示从M10起5点ON/OFF的情况下

M10	M11	M12	M13	M14																						
ON	OFF	ON	OFF	ON																						
ASCII代码		二进制代码*1																								
<table border="1"> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>31h</td> <td>30h</td> <td>31h</td> <td>30h</td> <td>31h</td> </tr> </table>		1	0	1	0	1	31h	30h	31h	30h	31h	<table border="1"> <tr> <td>DLE</td> <td></td> <td>DLE</td> <td></td> <td>DLE</td> <td></td> </tr> <tr> <td>10h</td> <td>10h</td> <td>10h</td> <td>10h</td> <td>10h</td> <td>10h</td> </tr> </table>			DLE		DLE		DLE		10h	10h	10h	10h	10h	10h
1	0	1	0	1																						
31h	30h	31h	30h	31h																						
DLE		DLE		DLE																						
10h	10h	10h	10h	10h	10h																					

\*1 有可能会添加附加代码。(☞ 19页 附加代码(10H))

## 字单位(位软元件16点单位)的情况下

以字单位读取、写入数据时的数据有关内容如下所示。

处理位数据以外的情况下，请参阅下述内容。

☞ 42页 处理实数数据、字符串数据时的注意事项

### ■以ASCII代码进行数据通信时

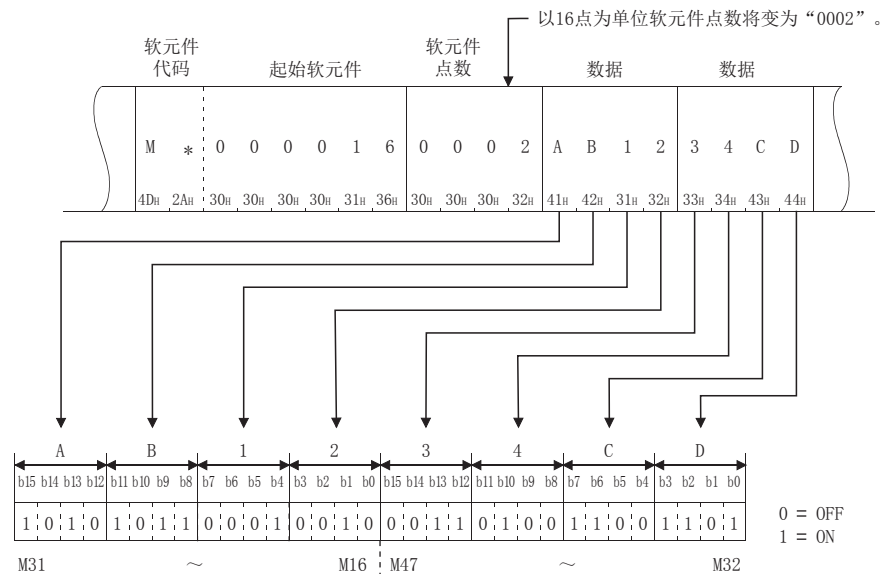
将1字(位软元件16点)的数值转换为ASCII代码4位(16进制数)后从高位进行发送。

指定英文字母的情况下，使用大写字母代码。

位软元件的ON/OFF将以4点为单位，置为16进制数1位的值。

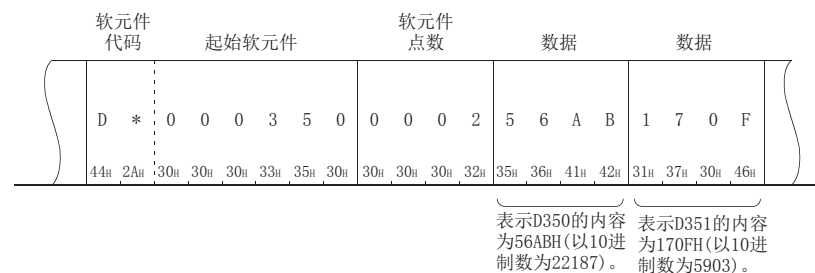
#### 例

通过M16显示32点的ON/OFF的情况下



#### 例

显示D350、D351的存储内容的情况下



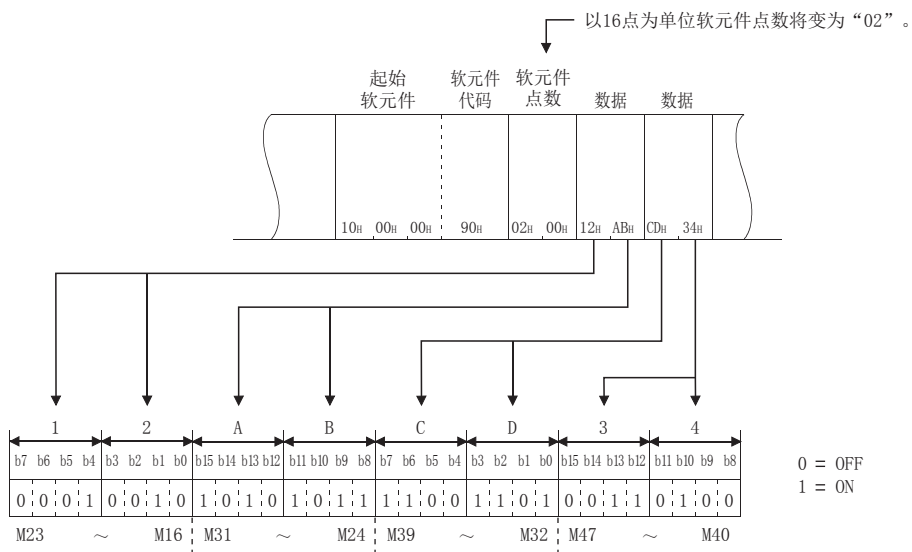


### ■以二进制代码进行数据通信时

以16点为单位，置为2字节的数值从低位字节(L:位0~7)进行发送。

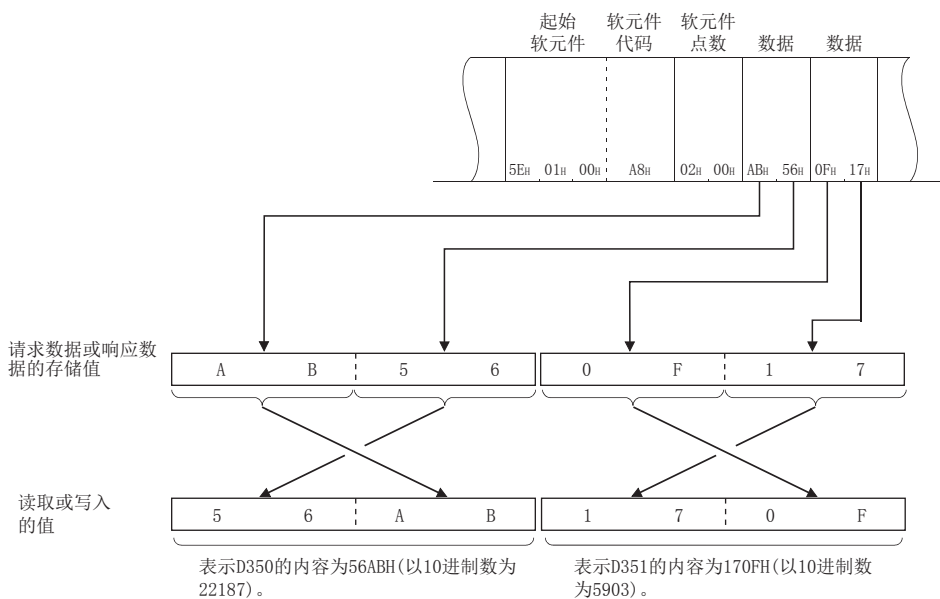
#### 例

通过M16显示32点的ON/OFF的情况下



#### 例

显示D350、D351的存储内容的情况下



## 双字单位(位软元件32点单位)的情况下

以双字单位读取、写入数据时的数据有关内容如下所示。

### ■以ASCII代码进行数据通信时

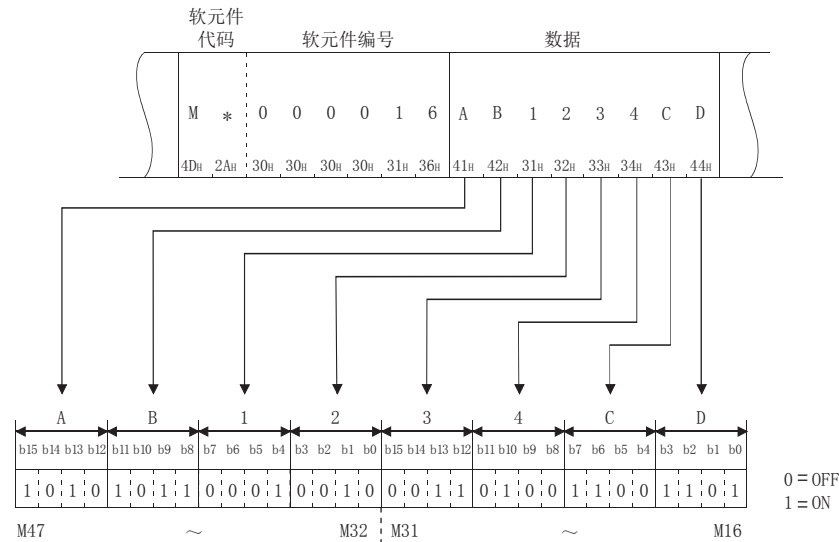
将2字(位软元件32点)的数值转换为ASCII代码8位(16进制数)后从高位进行发送。

指定英文字母的情况下,使用大写字母代码。

位软元件的ON/OFF将以4点为单位,置为16进制数1位的值。

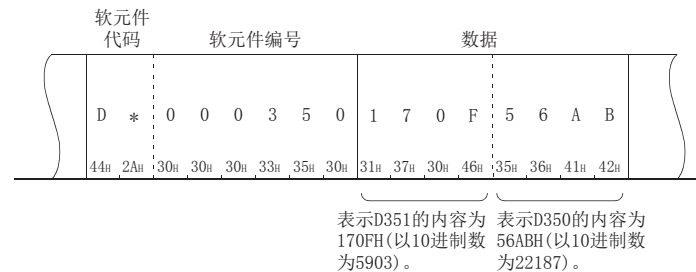
#### 例

通过M16显示32点的ON/OFF的情况下



#### 例

显示D350(D351)的存储内容的情况下

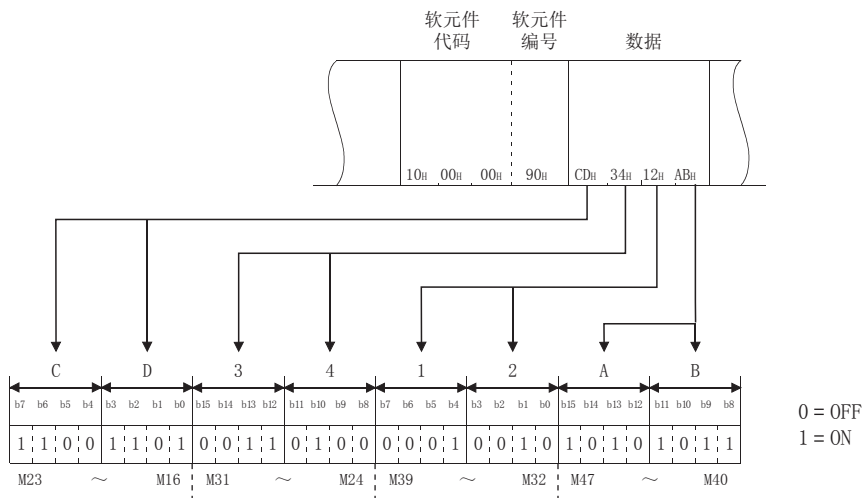


### ■以二进制代码进行数据通信时

以32点为单位，置为4字节的数据从低位字节(L:位0~7)进行发送。

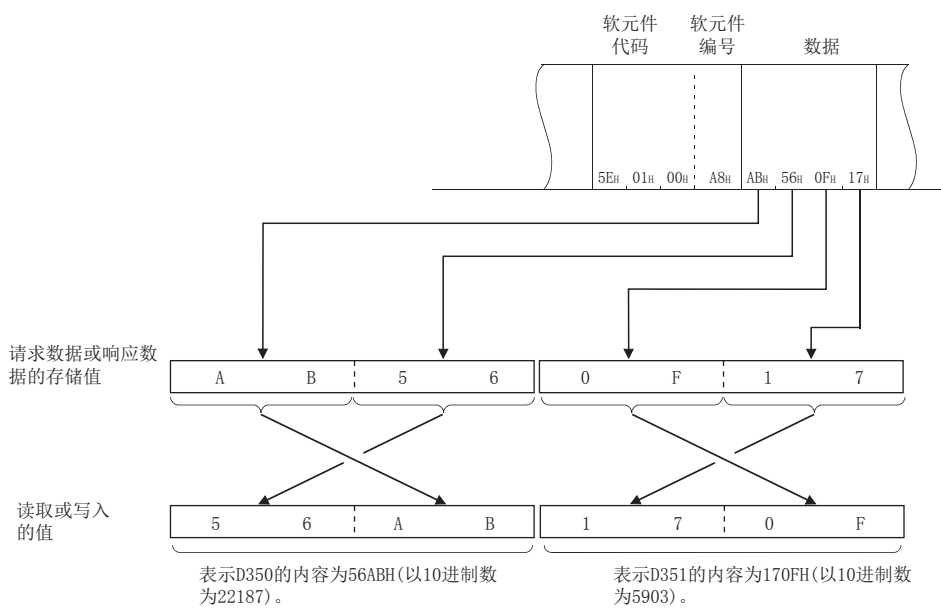
#### 例

通过M16显示32点的ON/OFF的情况下



#### 例

显示D350 (D351) 的存储内容的情况下



## 处理实数数据、字符串数据时的注意事项

将字数据、双字数据作为整数值(16位数据或32位数据)处理。

软件中存储整数以外(实数、字符串)的数据的情况下，将存储值作为整数值读取。

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

应在可编程控制器指令中作为实数或字符串使用的数据，以规定的指定方法，写入到软件/标签中。关于指令中使用的数据的指定方法，请参照《MELSEC iQ-F FX5编程手册(指令/通用FUN/FB篇)》。

### ■字符串数据的情况下

字符串数据的存储映像如下所示。

项目	ASCII代码字符串的情况下																
存储的字符串	“ABC”	“ABCD”															
字符代码	“41H”、“42H”、“43H”	“41H”、“42H”、“43H”、“44H”															
从D0开始存储了字符串数据时的映像	NULL表示00H。 <table border="1" style="margin-left: 20px;"> <tr> <td>D0</td> <td>B</td> <td>A</td> </tr> <tr> <td>D1</td> <td>NULL</td> <td>C</td> </tr> </table>	D0	B	A	D1	NULL	C	NULL表示00H。 <table border="1" style="margin-left: 20px;"> <tr> <td>D0</td> <td>B</td> <td>A</td> </tr> <tr> <td>D1</td> <td>D</td> <td>C</td> </tr> <tr> <td>D2</td> <td>NULL</td> <td>NULL</td> </tr> </table>	D0	B	A	D1	D	C	D2	NULL	NULL
D0	B	A															
D1	NULL	C															
D0	B	A															
D1	D	C															
D2	NULL	NULL															

### 例

通过处理字符串指令将使用的ASCII代码字符串数据写入到字软件的情况下

D0~D1中存储字符串(“ABCD”)：D0=4241H(“BA”)、D1=4443H(“DC”)

写入数据指定下述内容。

ASCII代码	二进制代码																																																																																				
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2">B</td> <td colspan="2">A</td> <td colspan="2">D</td> <td colspan="2">C</td> <td colspan="2">NULL</td> <td colspan="2">NULL</td> </tr> <tr> <td>4</td><td>2</td><td>4</td><td>1</td><td>4</td><td>4</td><td>4</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>34h</td><td>32h</td><td>34h</td><td>32h</td><td>34h</td><td>34h</td><td>34h</td><td>33h</td><td>30h</td><td>30h</td><td>30h</td><td>30h</td> </tr> <tr> <td colspan="4">D0</td> <td colspan="4">D1</td> <td colspan="4">D2</td> </tr> </table>	B		A		D		C		NULL		NULL		4	2	4	1	4	4	4	3	0	0	0	0	34h	32h	34h	32h	34h	34h	34h	33h	30h	30h	30h	30h	D0				D1				D2				<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2">A</td> <td colspan="2">B</td> <td colspan="2">C</td> <td colspan="2">D</td> <td colspan="2">NULL</td> <td colspan="2">NULL</td> </tr> <tr> <td>41h</td><td>42h</td><td>43h</td><td>44h</td><td>00h</td><td>00h</td><td>00h</td><td>00h</td><td>00h</td><td>00h</td><td>00h</td><td>00h</td> </tr> <tr> <td colspan="4">D0</td> <td colspan="4">D1</td> <td colspan="4">D2</td> </tr> </table>	A		B		C		D		NULL		NULL		41h	42h	43h	44h	00h	00h	00h	00h	00h	00h	00h	00h	D0				D1				D2			
B		A		D		C		NULL		NULL																																																																											
4	2	4	1	4	4	4	3	0	0	0	0																																																																										
34h	32h	34h	32h	34h	34h	34h	33h	30h	30h	30h	30h																																																																										
D0				D1				D2																																																																													
A		B		C		D		NULL		NULL																																																																											
41h	42h	43h	44h	00h	00h	00h	00h	00h	00h	00h	00h																																																																										
D0				D1				D2																																																																													

### 要点

将ASCII代码字符串数据通过ASCII代码进行数据通信的情况下，对各2字符排列替换存储。

## 软元件存储器扩展指定（以子指令的位7进行设定）

详情请参照 83页 软元件存储器的扩展指定。

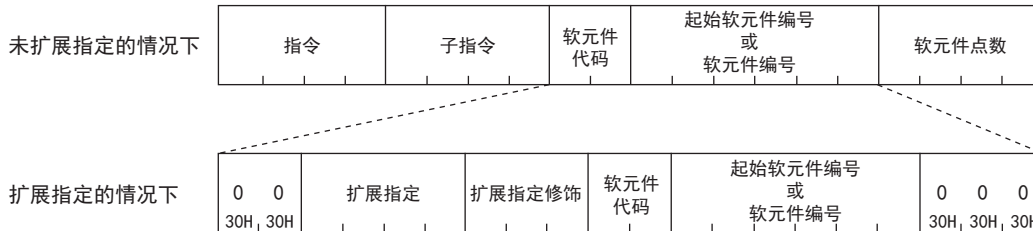
显示以模块访问软元件为对象的软元件的读取/写入以及使用变址寄存器的软元件的间接指定方法。

### ① 报文格式

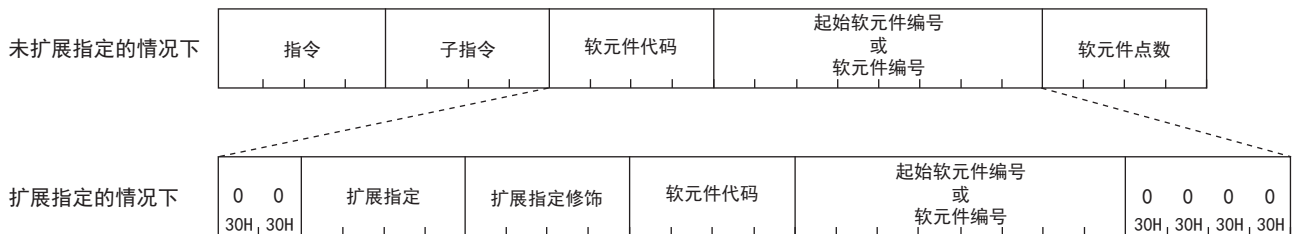
响应报文也同样扩展。

#### · 以ASCII代码进行数据通信时

指定代码2字符/编号6位时

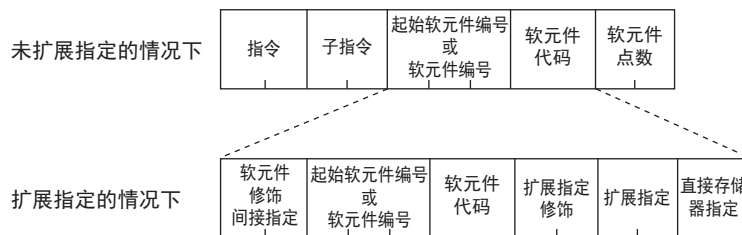


指定代码4字符/编号8位时

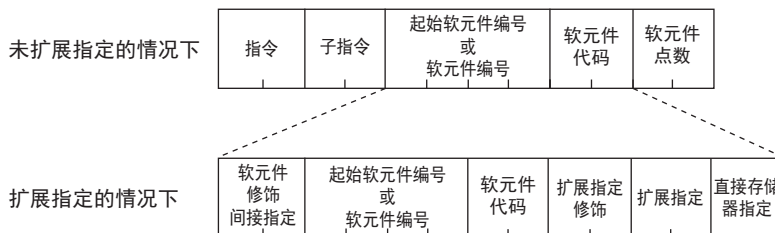


#### · 以二进制代码进行数据通信时

指定代码2字符/编号6位时

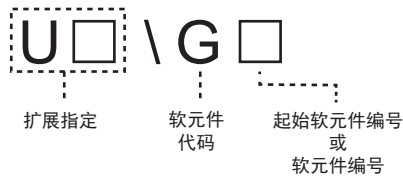


指定代码4字符/编号8位时



## ② 模块访问软元件的指定

模块访问软元件的编程时的指定形式和请求数据的对应如下所示。



### • 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。	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。
<p>例 001的情况下</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">             U [ ] [ ] [ ]              55H                     </div> <div style="border: 1px solid black; padding: 5px;">             U 0 0 1              55H   30H   30H   31H           </div> </div>	<p>例 001的情况下</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">             [ ] [ ] [ ] [ ]              [ ]H   [ ]H   [ ]H   [ ]H           </div> <div style="border: 1px solid black; padding: 5px;">             01H   00H           </div> </div>

### • 软元件代码

指定软元件范围的表（[P.33](#)）中的模块访问软元件。

#### • 起始软元件编号或软元件编号

格式与未扩展指定的情况下的报文相同。

#### • 直接存储器指定(仅以二进制代码通信时)

指定访问软元件的类型(智能功能模块软元件)。

模块访问软元件:指定F8H

## 设置/复位

指定位软元件的ON/OFF。

#### • ON的情况下：“1”

子指令类型	ASCII代码	二进制代码
指定代码2字符/编号6位	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             0 1              30H   31H           </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             01H           </div>
指定代码4字符/编号8位	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             0 0 0 1              30H   30H   30H   31H           </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             01H   00H           </div>

#### • OFF的情况下：“0”

子指令类型	ASCII代码	二进制代码
指定代码2字符/编号6位	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             0 0              30H   30H           </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             00H           </div>
指定代码4字符/编号8位	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             0 0 0 0              30H   30H   30H   30H           </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             00H   00H           </div>

# 批量读取

批量读取软元件的数据。

## 请求数据

### ■以ASCII代码进行数据通信时

指定代码2字符/编号6位时

	4字节	4字节	2字节	6字节	4字节	
未扩展指定的情况下	0 4 0 1 30H, 34H, 30H, 31H	子指令	软元件 代码	起始软元件编号	软元件点数	
扩展指定的情况下	0 0 30H, 30H	扩展指定	扩展设定修饰	软元件 代码	起始软元件编号	0 0 0 30H, 30H, 30H
	2字节	4字节	3字节	2字节	6字节	3字节

指定代码4字符/编号8位时

	4字节	4字节	4字节	8字节	4字节	
未扩展指定的情况下	0 4 0 1 30H, 34H, 30H, 31H	子指令	软元件 代码	起始软元件编号	软元件点数	
扩展指定的情况下	0 0 30H, 30H	扩展指定	扩展设定修饰	软元件 代码	起始软元件编号	0 0 0 0 30H, 30H, 30H, 30H
	2字节	4字节	4字节	4字节	10字节	4字节

### ■以二进制代码进行数据通信时

指定代码2字符/编号6位时

	2字节	2字节	3字节	1字节	2字节	
未扩展指定的情况下	01H, 04H	子指令	起始软元件 编号	软元件 代码	软元件 点数	
扩展指定的情况下	软元件 修饰 间接指定	起始软元件 编号	软元件 代码	扩展指定 修饰	扩展指定	直接存储 器指定
	2字节	3字节	1字节	2字节	2字节	1字节

指定代码4字符/编号8位时

	2字节	2字节	4字节	2字节	2字节	
未扩展指定的情况下	01H, 04H	子指令	起始软元件 编号	软元件 代码	软元件 点数	
扩展指定的情况下	软元件 修饰 间接指定	起始软元件 编号	软元件 代码	扩展指定 修饰	扩展指定	直接存储 器指定
	2字节	4字节	2字节	2字节	2字节	1字节

## ■子指令

指定通过项目选择的子指令。

项目			子指令					
数据大小指定	软元件指定形式	软元件存储器扩展指定	ASCII代码 (上段: 字符, 下段: 字符代码)				二进制代码	
位单位	指定代码2字符/编号 6位	无指定	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		有指定	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	指定代码4字符/编号 8位	有指定	0	0	8	3	83H	00H
			30H	30H	38H	33H		
字单位	指定代码2字符/编号 6位	无指定	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		有指定	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	指定代码4字符/编号 8位	有指定	0	0	8	2	82H	00H
			30H	30H	38H	32H		

## ■软元件代码

指定对应要读取的软元件的种类的“软元件代码”。请参照软元件代码一览(☞ 33页)

未对应双字软元件、长变址寄存器(LZ)。

## ■软元件编号

指定要读取的软元件的起始编号。

## ■软元件点数

指定要读取的软元件的点数。

项目	软元件点数	
	ASCII代码	二进制代码
以位单位进行读取的情况下	1~3584点	1~3584点
以字单位进行读取的情况下	1~960点	1~960点

## 响应数据

以16进制数存储所读取的软元件的值。根据ASCII代码和二进制代码，数据的排列会有所不同。

读取数据

## 通信示例

### ■以位单位进行读取的情况下

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 30H, 34H, 30H, 31H	0 0 0 1 30H, 30H, 30H, 31H	M * 4DH, 2AH	0 0 0 1 0 0 30H, 30H, 30H, 31H, 30H, 30H
			0 0 0 8 30H, 30H, 30H, 38H

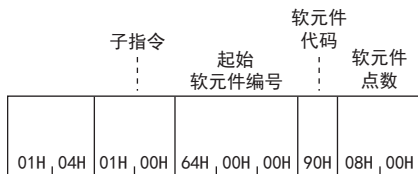
(响应数据)

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



- 以二进制代码进行数据通信时

(请求数据)



(响应数据)



■以字单位读取的情况下 (位软元件)

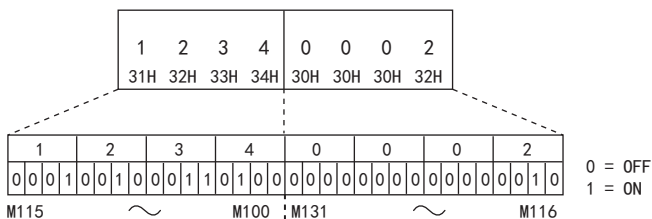
读取M100~M131 (2字)。

- 以ASCII代码进行数据通信时

(请求数据)

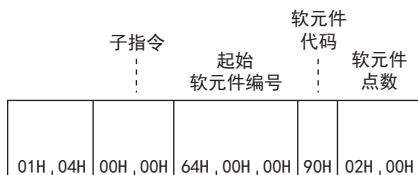


(响应数据)

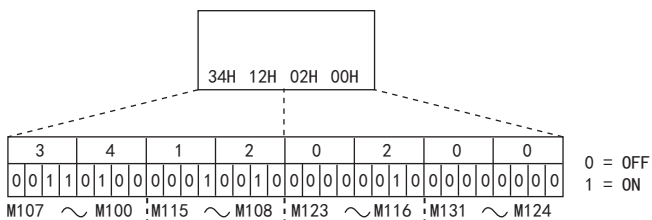


- 以二进制代码进行数据通信时

(请求数据)



(响应数据)



## ■以字单位读取的情况下（字软元件）

读取T100~T102的值。

将其作为存储了T100=4660(1234H)、T101=2(2H)、T102=7663(1DEFH)。

- 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	T	N	0	0	0	1	0	0	0	0	0	3
30H	34H	30H	31H	30H	30H	54H	4EH	30H	30H	30H	31H	30H	30H	30H	30H	30H	33H

(响应数据)

1	2	3	4	0	0	0	2	1	D	E	F
31H	32H	33H	34H	30H	30H	30H	32H	31H	44H	45H	46H
T100				T101				T102			

- 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	
01H	04H	00H	00H	64H	00H	00H	C2H
03H	00H						

(响应数据)

34H	12H	02H	00H	EFH	1DH
T100		T101		T102	

# 批量写入

批量写入软元件的数据。

## 请求数据

### ■以ASCII代码进行数据通信时

指定代码2字符/编号6位时

	4字节	4字节	2字节	6字节	4字节	
未扩展指定的情况下	1 4 0 1 31H, 34H, 30H, 31H	子指令	软元件 代码	起始软元件编号	软元件点数	点数的写入数据

扩展指定的情况下	0 0 30H, 30H	扩展指定	扩展设定修饰	软元件 代码	起始软元件编号	0 0 0 30H, 30H, 30H
	2字节	4字节	3字节	2字节	6字节	3字节

指定代码4字符/编号8位时

	4字节	4字节	4字节	8字节	4字节	
未扩展指定的情况下	1 4 0 1 31H, 34H, 30H, 31H	子指令	软元件 代码	起始软元件编号	软元件点数	点数的写入数据

扩展指定的情况下	0 0 30H, 30H	扩展指定	扩展设定修饰	软元件 代码	起始软元件编号	0 0 0 0 30H, 30H, 30H, 30H
	2字节	4字节	4字节	4字节	10字节	4字节

### ■以二进制代码进行数据通信时

指定代码2字符/编号6位时

	2字节	2字节	3字节	1字节	2字节	
未扩展指定的情况下	01H, 14H	子指令	起始软元件 编号	软元件 代码	软元件 点数	点数的写入数据

扩展指定的情况下	软元件 修饰 间接指定	起始软元件 编号	软元件 代码	扩展设定 修饰	扩展指定	直接 存储器 指定
	2字节	3字节	1字节	2字节	2字节	1字节

指定代码4字符/编号8位时

	2字节	2字节	4字节	2字节	2字节	
未扩展指定的情况下	01H, 14H	子指令	起始软元件 编号	软元件 代码	软元件 点数	点数的写入数据

扩展指定的情况下	软元件 修饰 间接指定	起始软元件 编号	软元件 代码	扩展设定 修饰	扩展指定	直接 存储器 指定
	2字节	4字节	2字节	2字节	2字节	1字节

## ■子指令

指定通过项目选择的子指令。

项目			子指令					
数据大小指定	软元件指定形式	软元件存储器扩展指定	ASCII代码 (上段: 字符, 下段: 字符代码)				二进制代码	
位单位	指定代码2字符/编号 6位	无指定	0	0	0	1	01H	00H
			30H	30H	30H	31H		
		有指定	0	0	8	1	81H	00H
			30H	30H	38H	31H		
	指定代码4字符/编号 8位	有指定	0	0	8	3	83H	00H
			30H	30H	38H	33H		
字单位	指定代码2字符/编号 6位	无指定	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		有指定	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	指定代码4字符/编号 8位	有指定	0	0	8	2	82H	00H
			30H	30H	38H	32H		

## ■软元件代码

指定对应要写入的软元件的种类的“软元件代码”。请参照软元件代码一览(☞ 33页)

未对应双字软元件、长变址寄存器(LZ)。

## ■软元件编号

指定要写入的软元件的起始编号。

## ■软元件点数

指定要写入的软元件的点数。

项目	软元件点数	
	ASCII代码	二进制代码
以位单位写入的情况下	1~3584点	1~3584点
以字单位写入的情况下	1~960点	1~960点

## ■写入数据

将写入软元件的值指定“软元件点数”中指定的点数。

## 响应数据

没有批量写入指令的响应数据。

## 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	写入数据																						
1	4	0	1	0	0	0	1	M	*	0	0	0	1	0	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	30H	38H	31H	31H	30H	30H	31H	31H	30H	30H	

M100

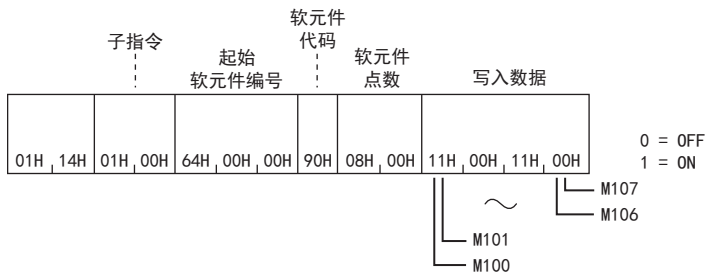
~

M107

0 = OFF

1 = ON

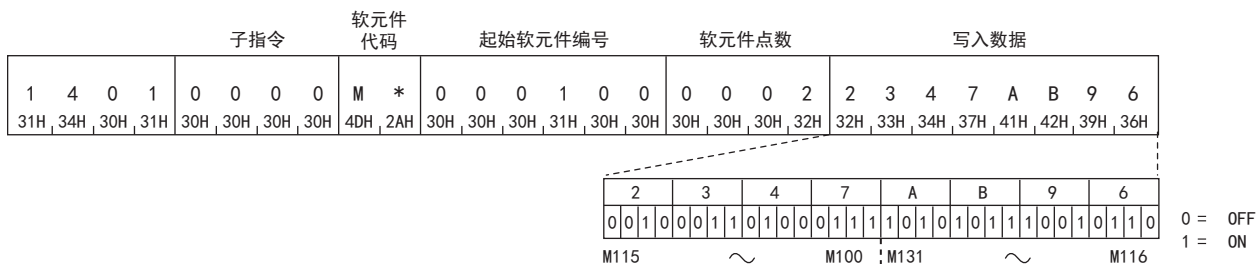
- 以二进制代码进行数据通信时  
(请求数据)



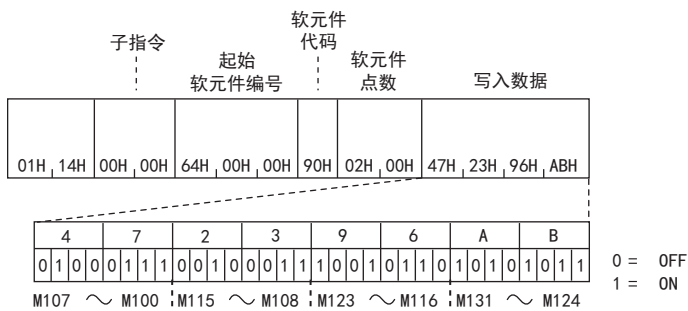
■以字单位写入的情况下 (位软元件)

将值写入M100~M131 (2字)。

- 以ASCII代码进行数据通信时  
(请求数据)



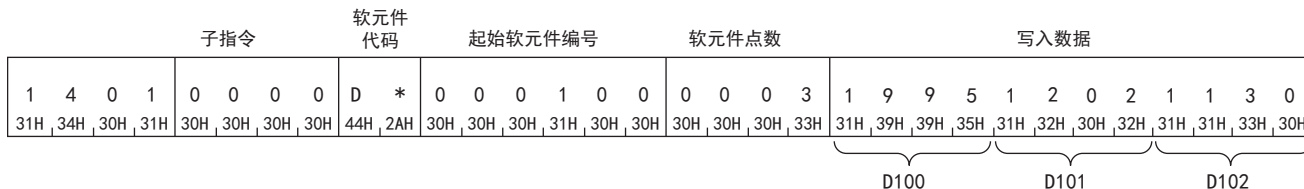
- 以二进制代码进行数据通信时  
(请求数据)



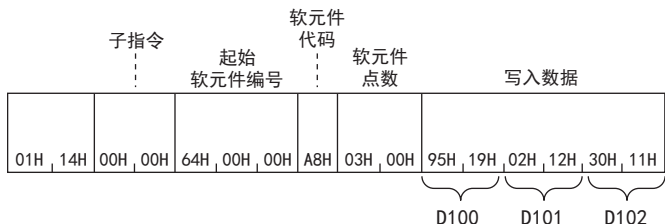
■以字单位写入的情况 (字软元件)

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

- 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)

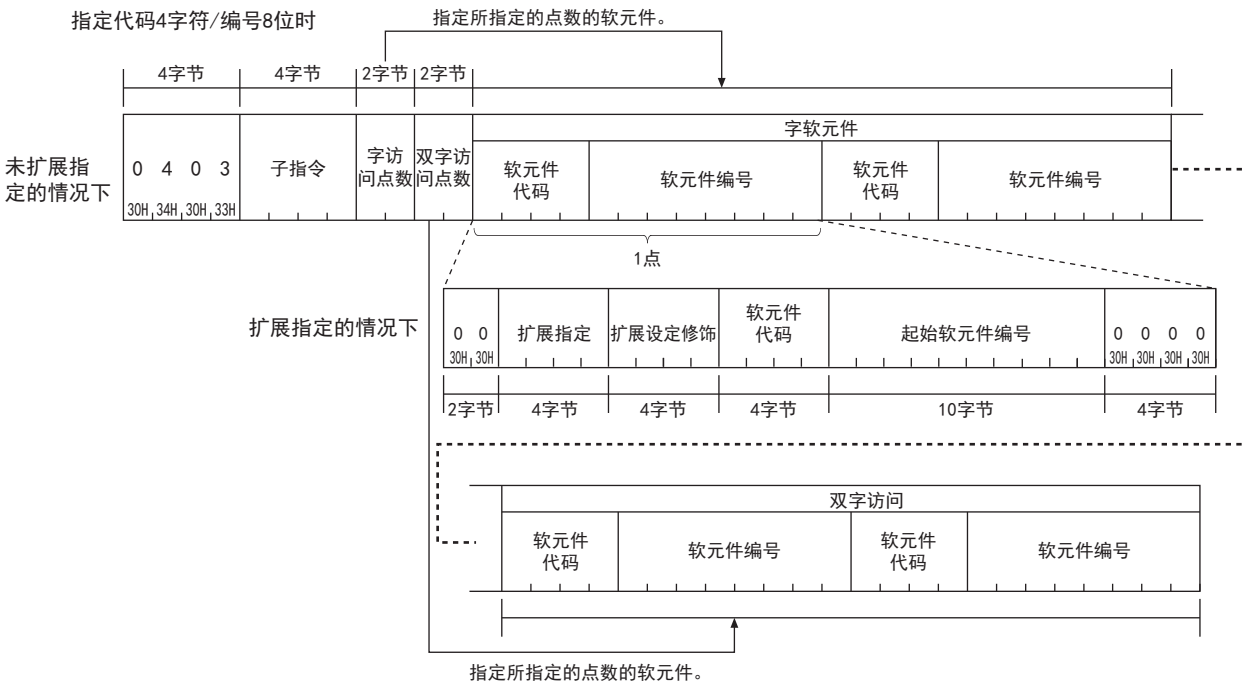
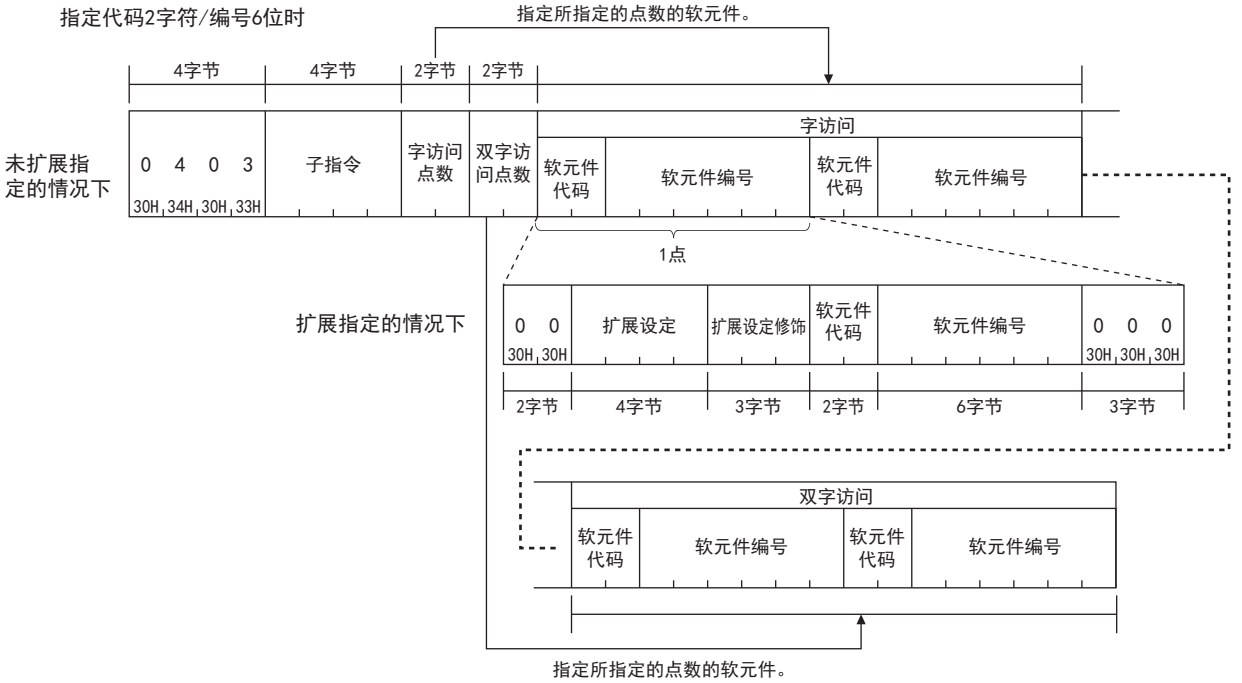


# 随机读取

随机指定软元件编号，读取软元件的值。

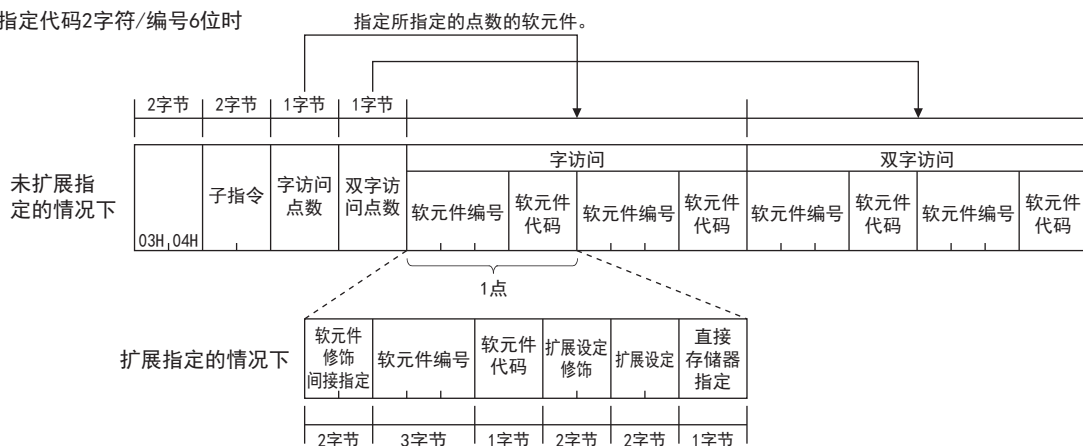
## 请求数据

### ■以ASCII代码进行数据通信时

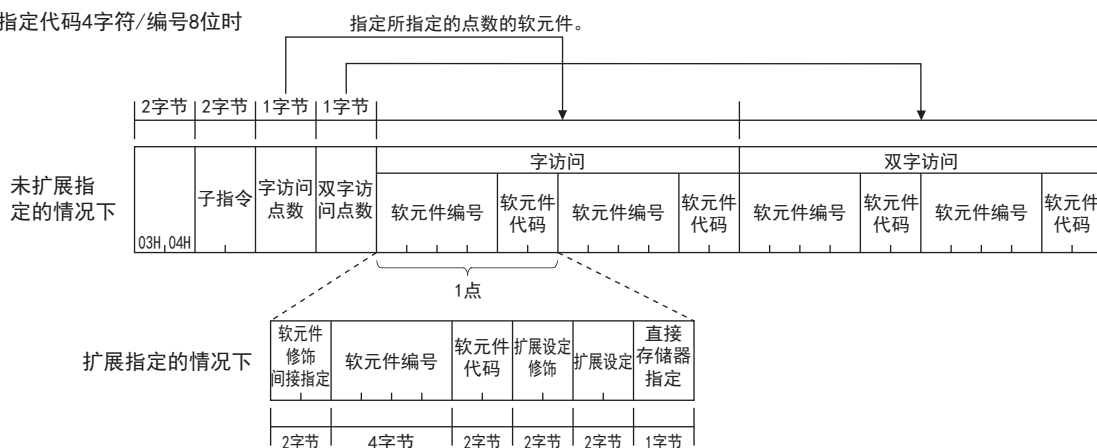


## ■以二进制代码进行数据通信时

指定代码2字符/编号6位时



指定代码4字符/编号8位时



## ■子指令

指定通过项目选择的子指令。

项目			子指令					
数据大小指定	软元件指定形式	软元件存储器扩展指定	ASCII代码 (上段: 字符, 下段: 字符代码)				二进制代码	
字单位	指定代码2字符/编号6位	无指定	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		有指定	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	指定代码4字符/编号8位	有指定	0	0	8	2	82H	00H
			30H	30H	38H	32H		

## ■字访问点数、双字访问点数

指定要读取的软元件的点数。

项目	内容	点数	
		ASCII代码	二进制代码
字访问点数	指定以1字为单位访问的情况下的点数。 位软元件以16点为单位, 字软元件以1字为单位。	1≤字访问点数+双字访问点数≤192 软元件存储器有扩展指定的情况下, 设为访问点数×2进行计算。	
双字访问点数	指定以2字为单位访问的情况下的点数。 位软元件以32点为单位, 字软元件以2字为单位。		

## ■软元件代码、软元件编号

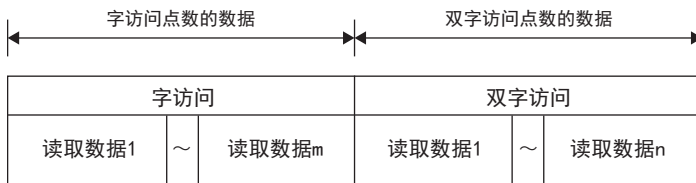
指定要读取的软元件。

项目	内容
字访问	指定以“字访问点数”指定的点数的软元件。将“字访问点数”设为0点的情况下，无需指定。
双字访问	指定以“双字访问点数”指定的点数的软元件。将“双字访问点数”设为0点的情况下，无需指定。

字访问的软元件→按照双字访问的软元件的顺序进行设定。

## 响应数据

以16进制数存储所读取的软元件的值。根据ASCII代码和二进制代码，数据的排列会有所不同。

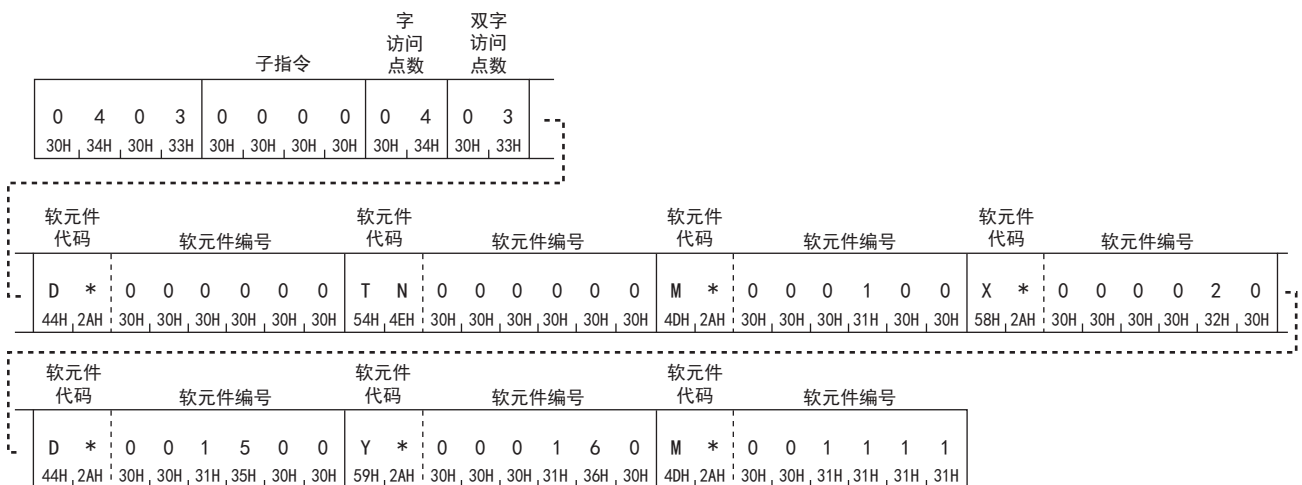


## 通信示例

字访问时，读取D0、T0、M100~M115、X20~X37，双字访问时，读取D1500~D1501、Y160~Y217、M1111~M1142。将其作为存储了D0=6549（1995H）、T0=4610（1202H）、D1500=20302（4F4EH）、D1501=19540（4C54H）。

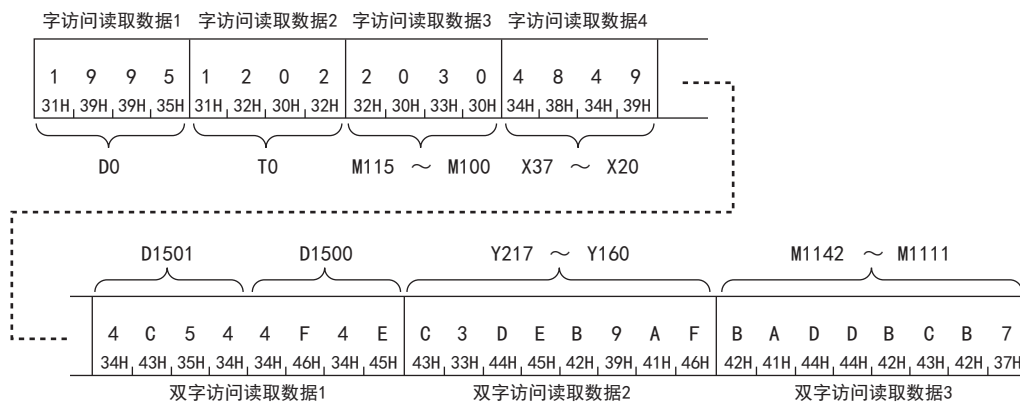
### ■以ASCII代码进行数据通信时

- 请求数据

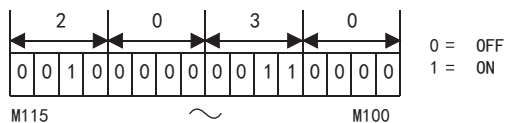




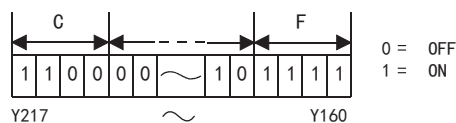
• 响应数据



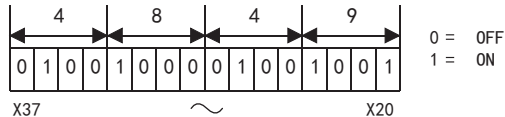
字访问读取数据3



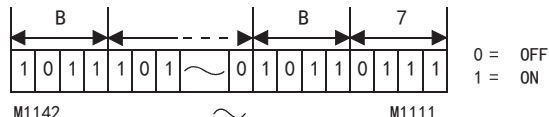
双字访问读取数据2



字访问读取数据4

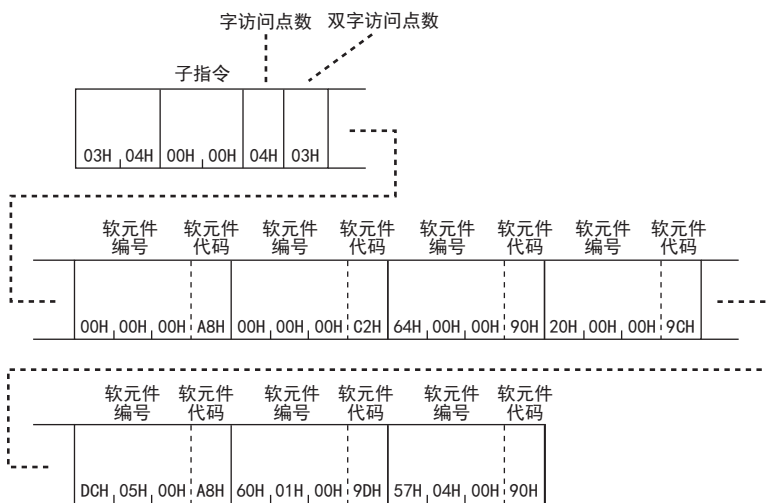


双字访问读取数据3

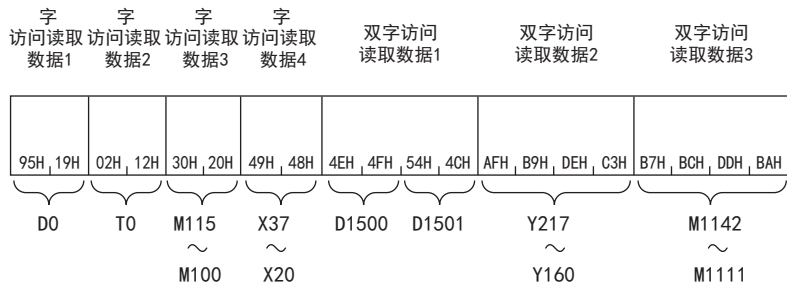


■以二进制代码进行数据通信时

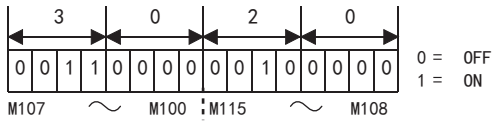
• 请求数据



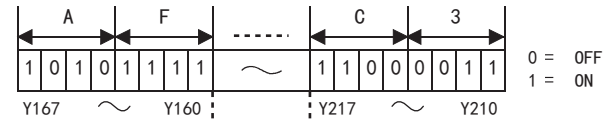
• 响应数据



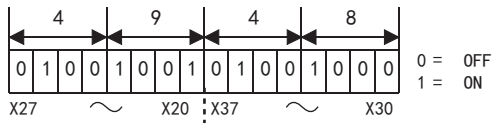
字访问读取数据3



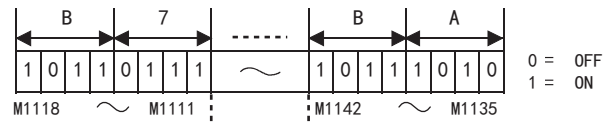
双字访问读取数据2



字访问读取数据4



双字访问读取数据3



# 随机写入

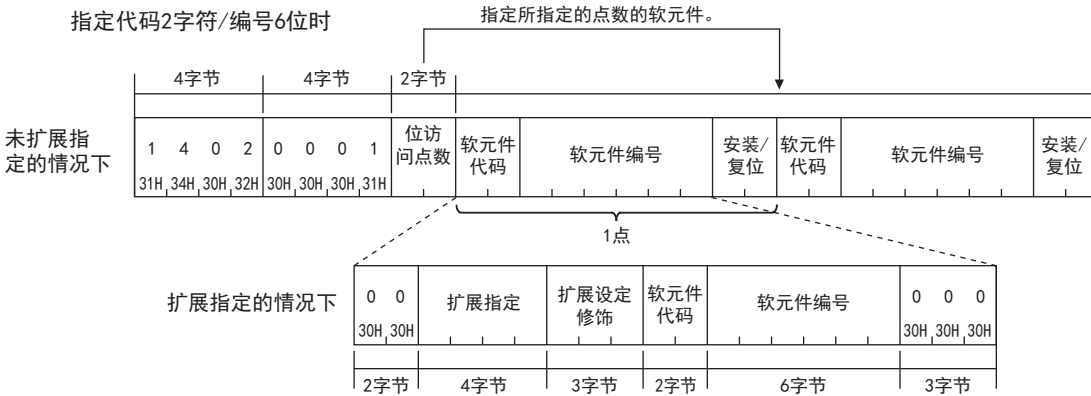
随机指定元件的元件编号，写入数据。

## 请求数据

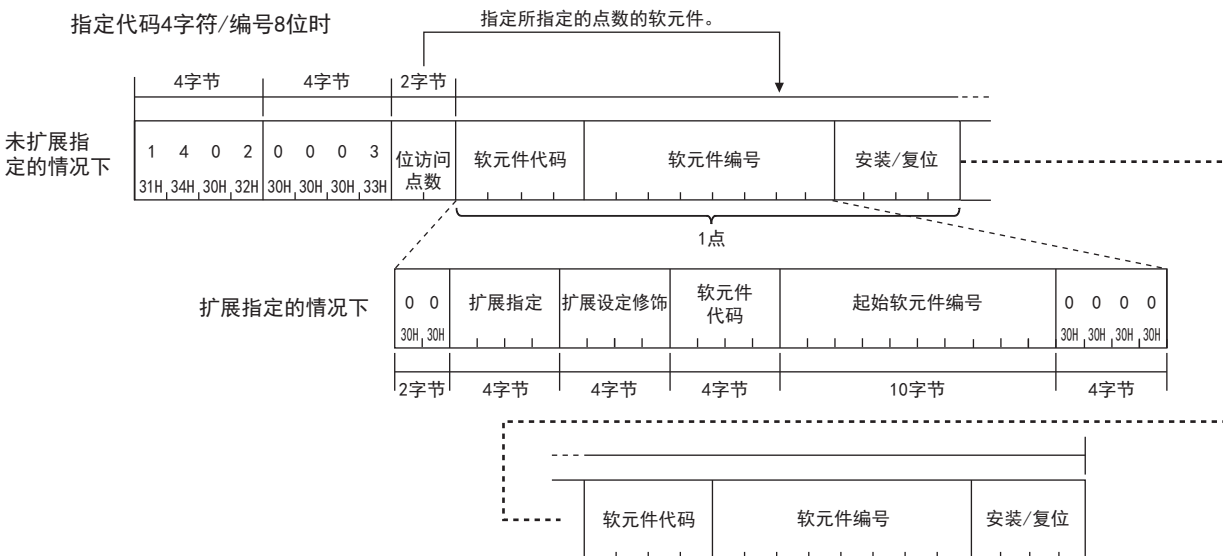
### ■以位单位写入的情况下

- 以ASCII代码进行数据通信时

指定代码2字符/编号6位时

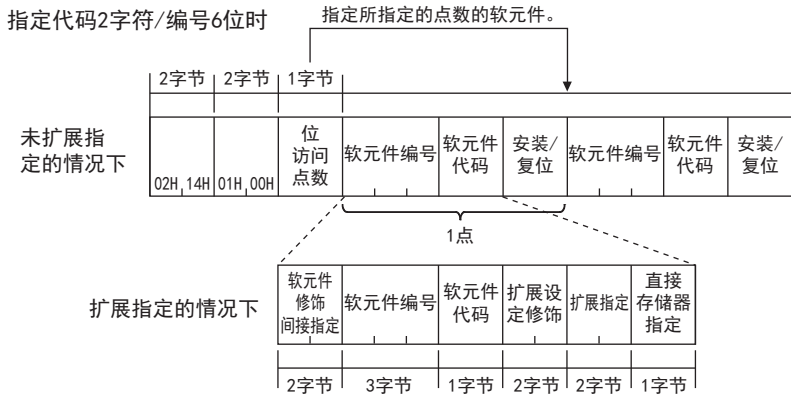


指定代码4字符/编号8位时

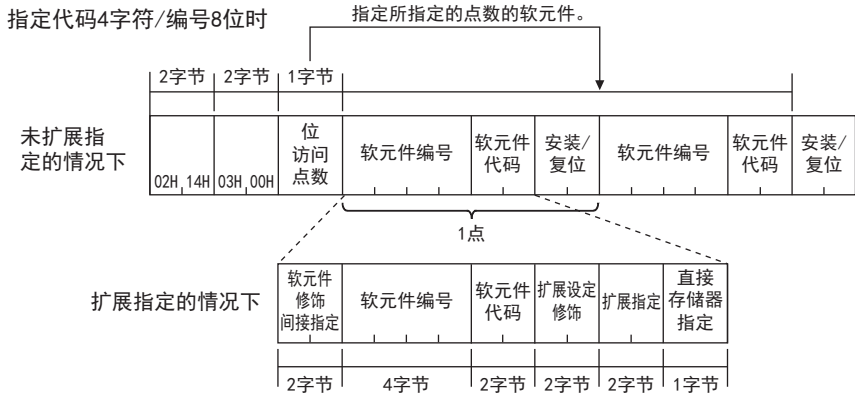


- 以二进制代码进行数据通信时

指定代码2字符/编号6位时



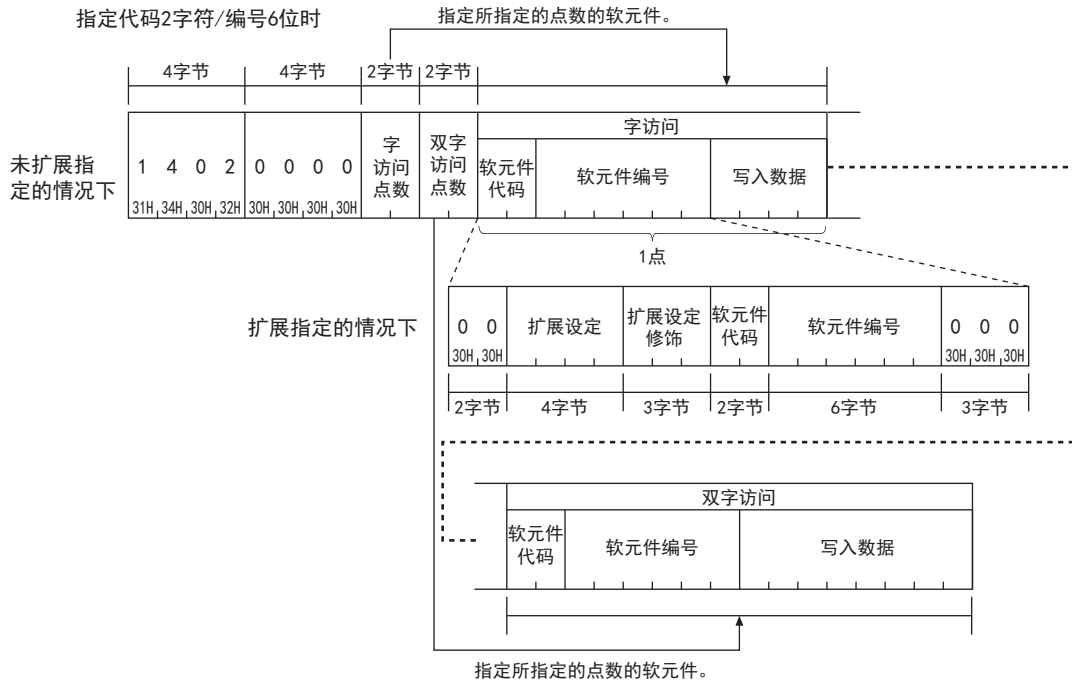
指定代码4字符/编号8位时



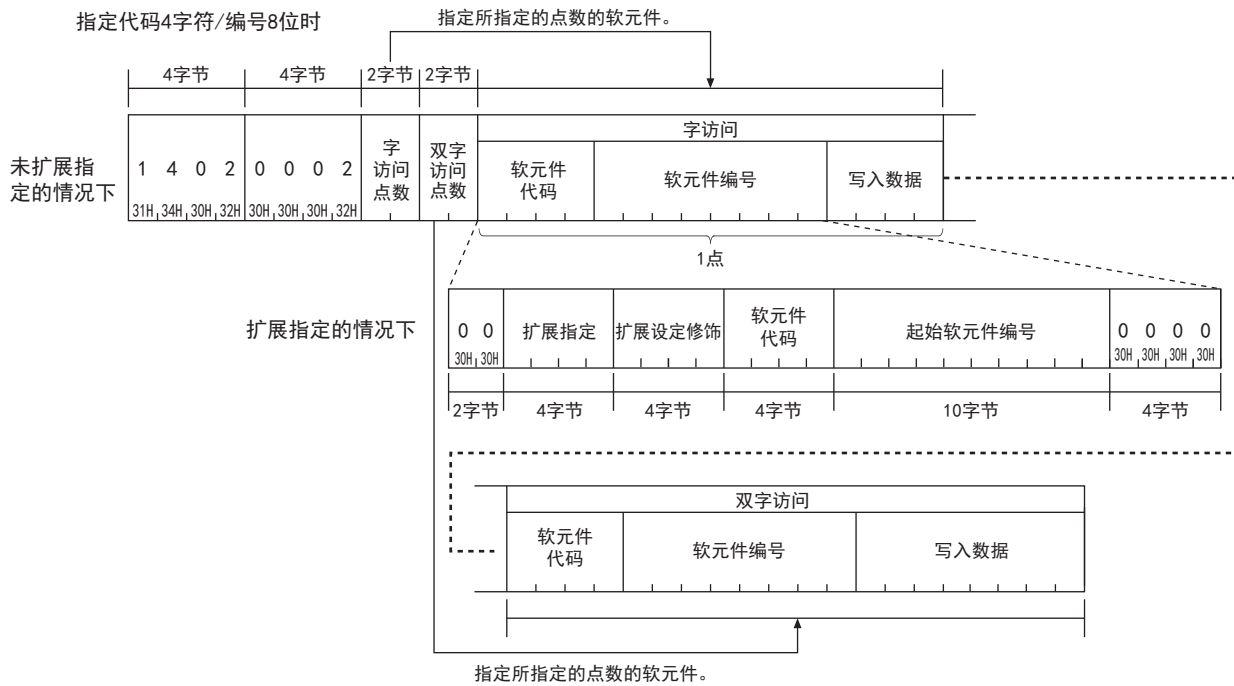
## ■以字单位写入的情况下

- 以ASCII代码进行数据通信时

指定代码2字符/编号6位时

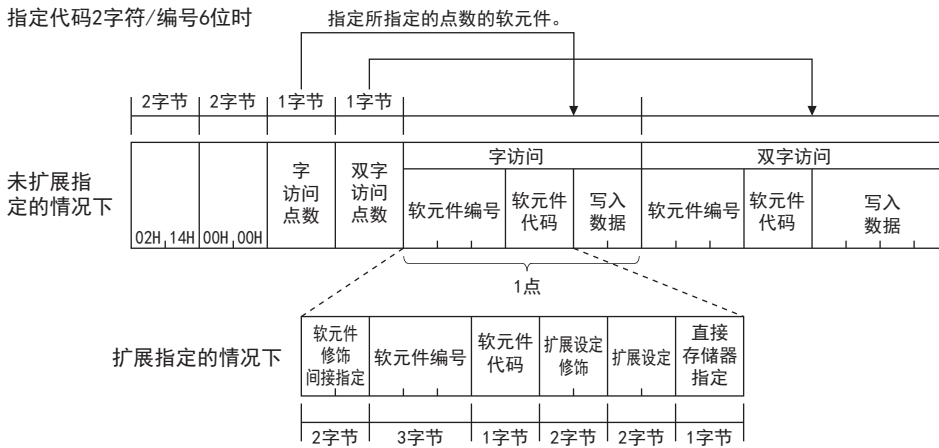


指定代码4字符/编号8位时

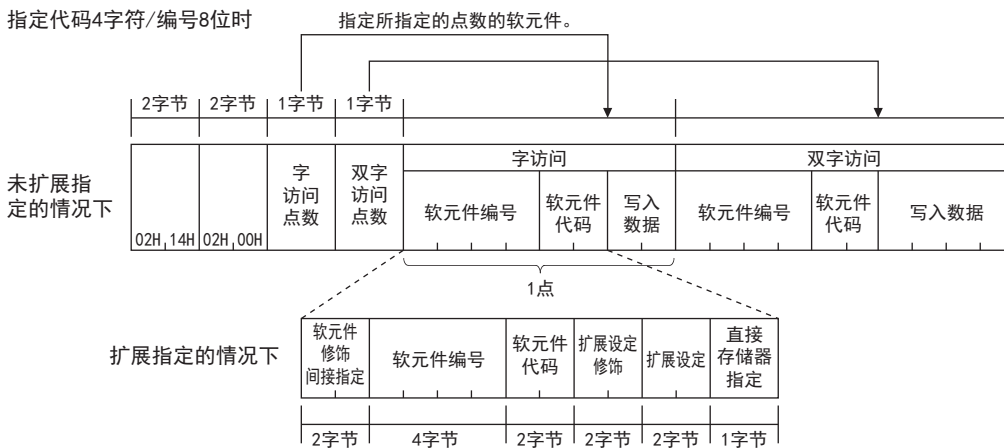


- 以二进制代码进行数据通信时

指定代码2字符/编号6位时



指定代码4字符/编号8位时



## ■子指令

指定通过项目选择的子指令。

项目	子指令							
数据大小指定	软元件指定形式	软元件存储器扩展指定	ASCII代码 (上段: 字符, 下段: 字符代码)				二进制代码	
位单位	指定代码2字符/编号6位	无指定	0	0	0	1	01H	00H
			30H	30H	30H	31H		
	有指定	0	0	8	1	81H	00H	
		30H	30H	38H	31H			
字单位	指定代码2字符/编号6位	无指定	0	0	0	0	00H	00H
			30H	30H	30H	30H		
	有指定	0	0	8	0	80H	00H	
		30H	30H	38H	30H			
指定代码4字符/编号8位	有指定	0	0	8	2	82H	00H	
		30H	30H	38H	32H			

## ■位访问点数、字访问点数、双字访问点数

项目	内容	点数	
		ASCII代码	二进制代码
位访问点数	以1点为单位指定位软元件的点数。	1~188 软元件存储器有扩展指定的情况下 1~94	
字访问点数	指定以1字为单位访问的情况下的点数。 位软元件以16点为单位, 字软元件以1字为单位。	1≤字访问点数×12+双字访问点数×14 ≤1920 软元件存储器有扩展指定的情况下, 设为访问点数×2进行计算。	
双字访问点数	指定以2字为单位访问的情况下的点数。 位软元件以32点为单位, 字软元件以2字为单位。		

## ■软元件代码、软元件编号、写入数据

指定要写入的软元件。

以16进制数指定写入数据。

项目	内容
字访问	指定以“字访问点数”指定的点数的软元件。将“字访问点数”设为0点的情况下，无需指定。
双字访问	指定以“双字访问点数”指定的点数的软元件。将“双字访问点数”设为0点的情况下，无需指定。

## ■安装/复位

指定位软元件的ON/OFF。

- 指定代码2字符/编号6位

项目	要写入的数据		备注
	ON时	OFF时	
ASCII代码	“01”	“00”	从“0”开始按顺序发送2字符
二进制代码	01H	00H	发送左侧1字节的数值

- 指定代码4字符/编号8位

项目	要写入的数据		备注
	ON时	OFF时	
ASCII代码	“0001”	“0000”	从“0”开始按顺序发送4字符
二进制代码	0001H	0000H	发送左侧2字节的数值

## 响应数据

没有随机写入指令的响应数据。

## 通信示例

### ■以位单位写入的情况下

将M50设为OFF，将Y27设为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	安装/ 复位	软元件 代码	软元件编号	安装/ 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	7	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	37H	30H	31H

- 以二进制代码进行数据通信时  
(请求数据)

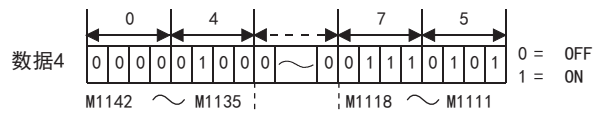
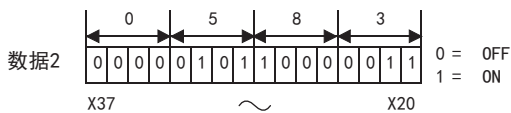
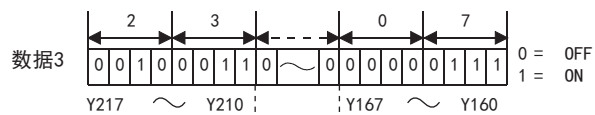
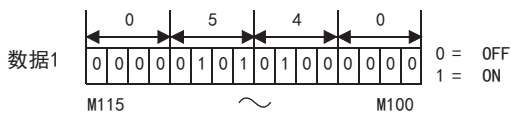
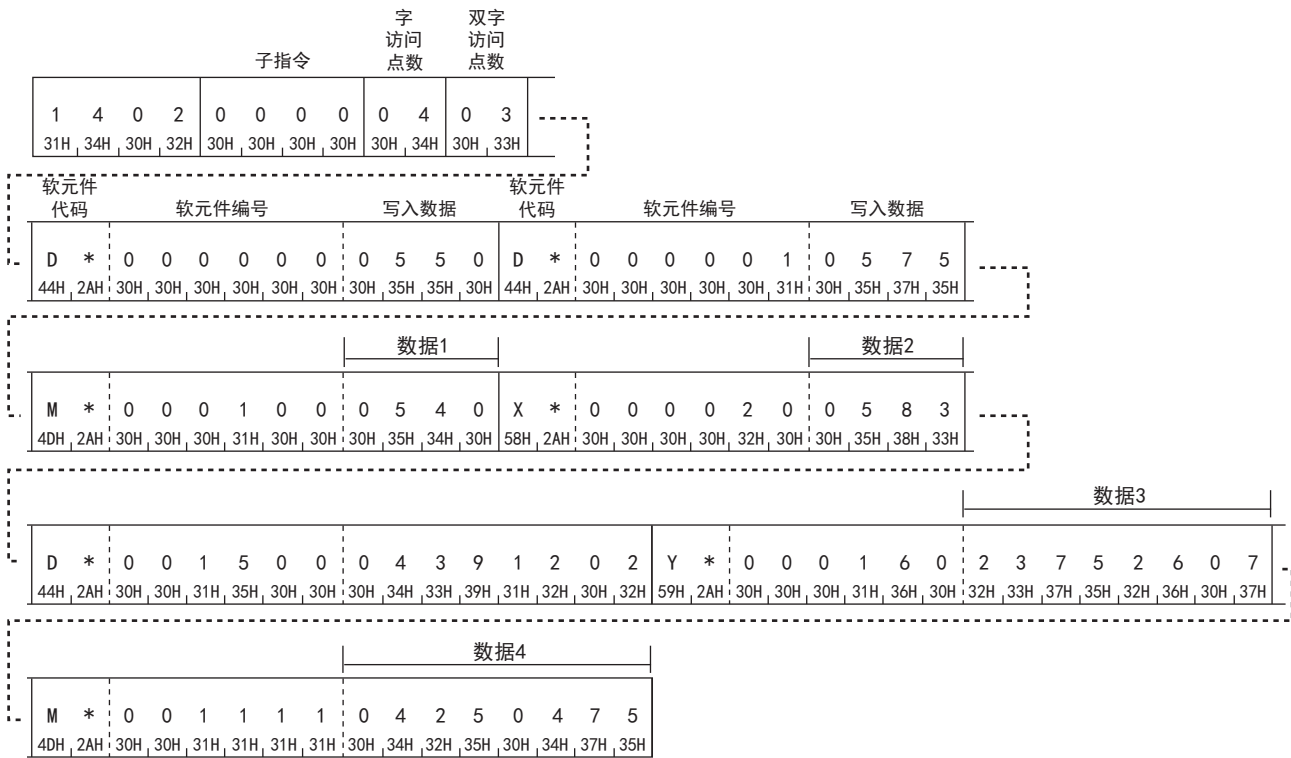
子指令		位访问 点数	软元件 代码	安装/ 复位	软元件 代码	安装/ 复位								
02H	14H	01H	00H	02H	32H	00H	00H	90H	00H	27H	00H	00H	9DH	01H

## ■以字单位写入的情况下

如下将值写入软元件。

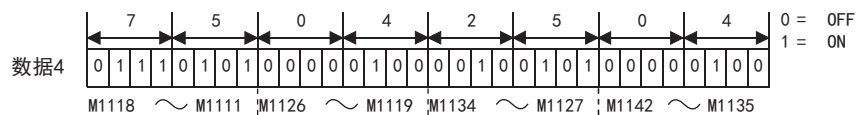
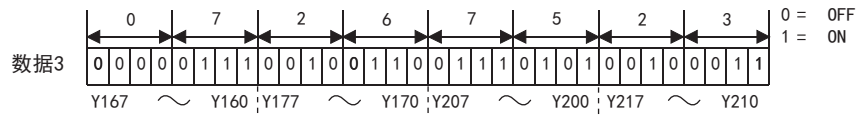
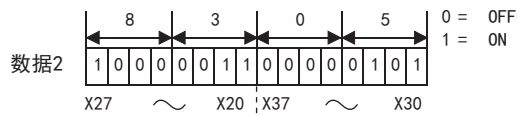
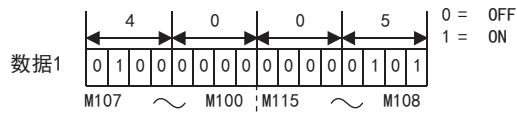
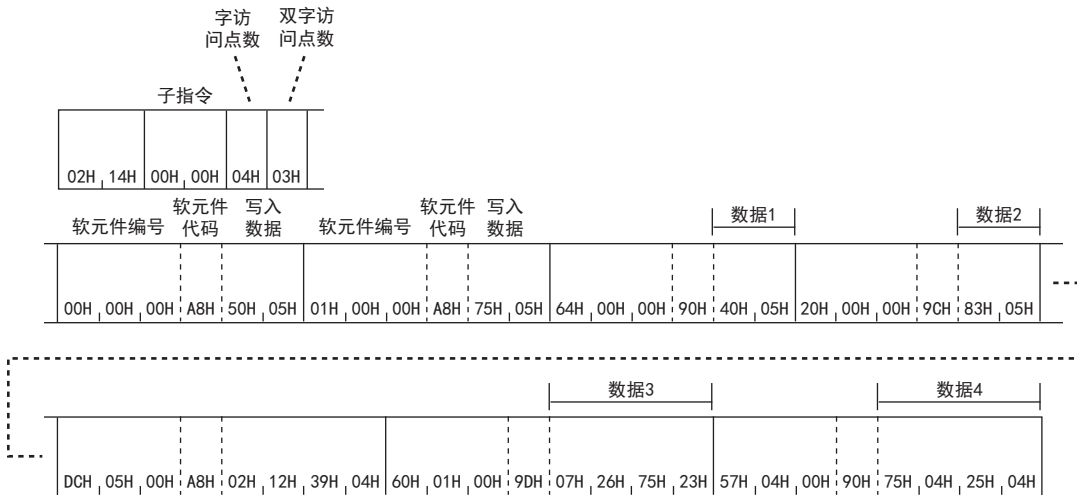
项目	要写入的软元件
字访问	D0, D1, M100~M115, X20~X37
双字访问	D1500~D1501, Y160~Y217, M1111~M1142

- 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)



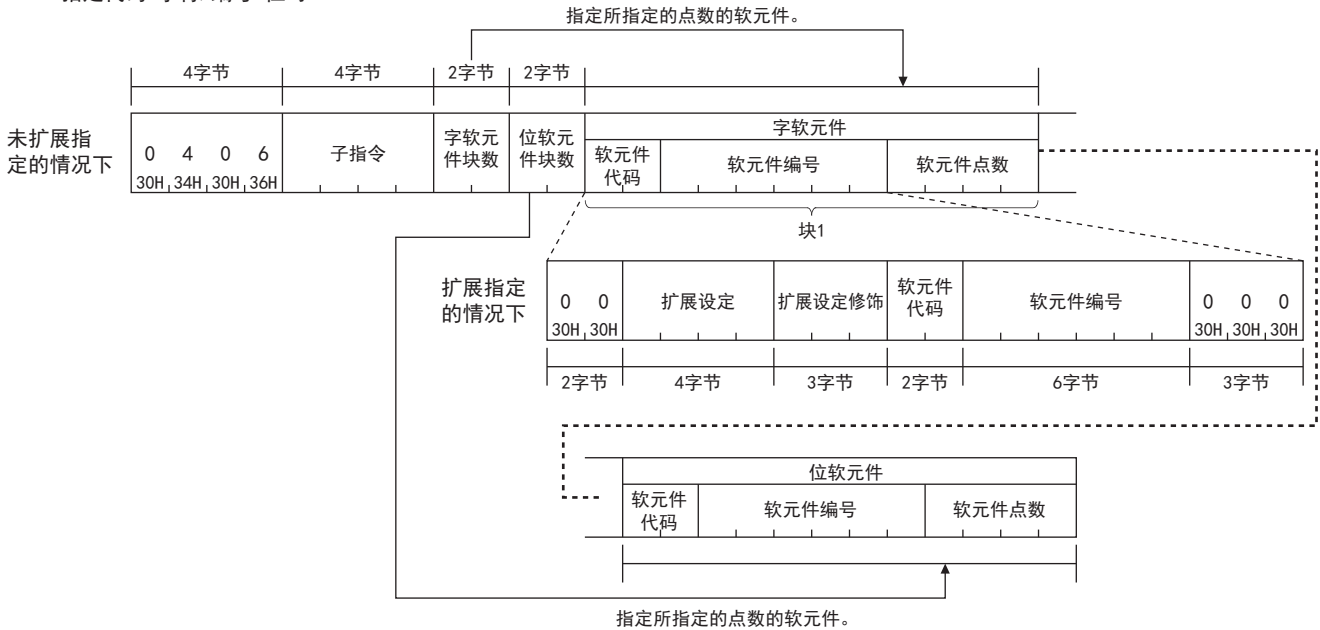
# 批量读取多个块

使用示例，说明将位软元件（1点=16位）和字软元件（1点=1字）的n点作为1块，随机指定多个块进行读取的控制步骤。

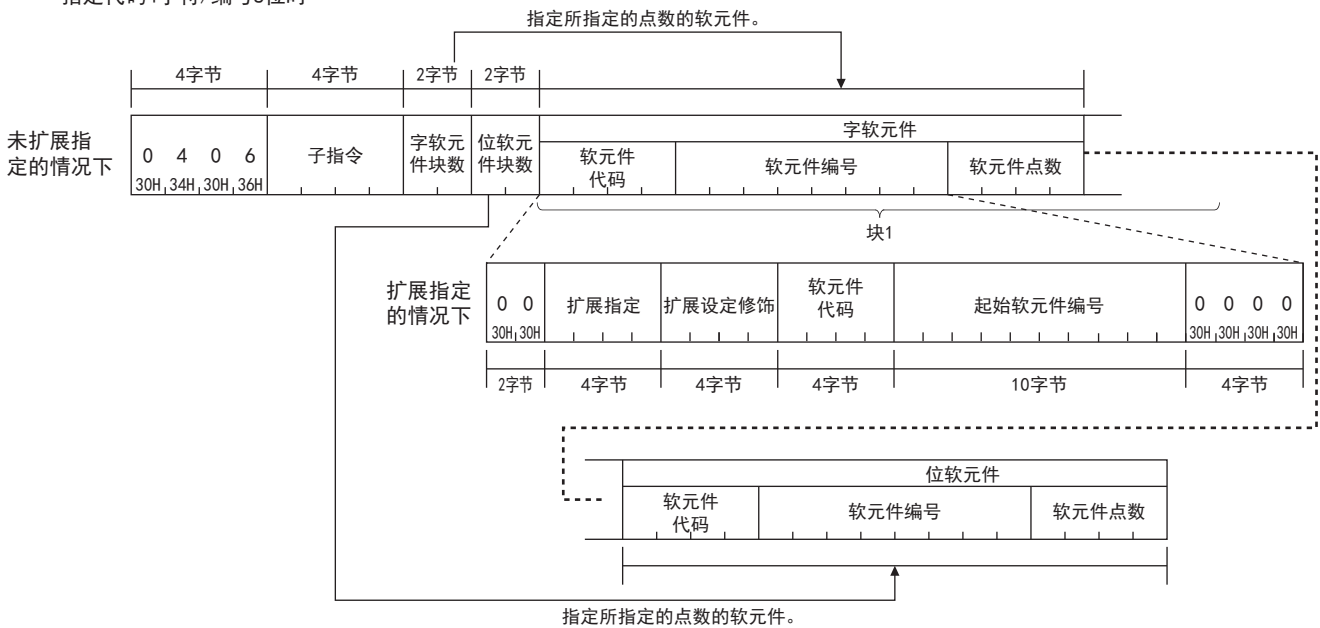
## 请求数据

### ■以ASCII代码进行数据通信时

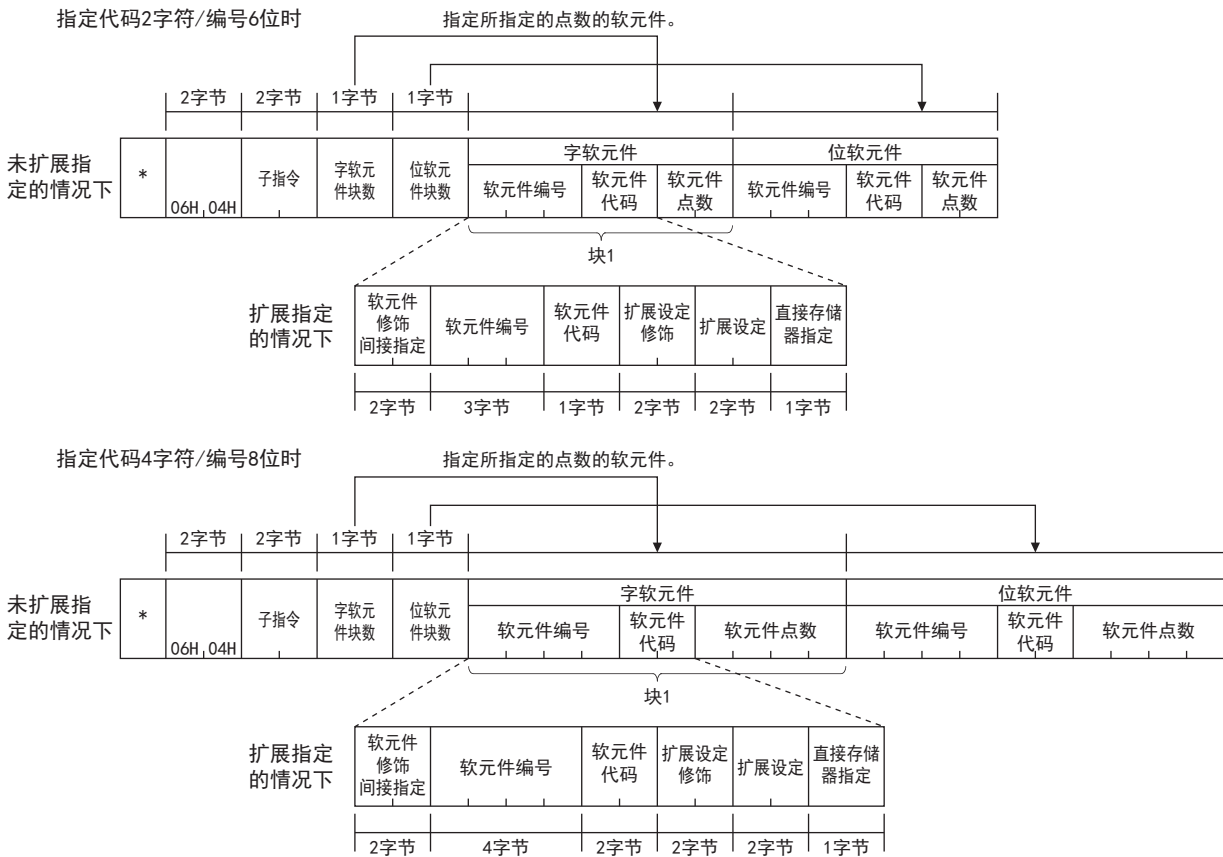
指定代码2字符/编号6位时



指定代码4字符/编号8位时



## ■以二进制代码进行数据通信时



### ①子指令

指定通过项目选择的子指令。

项目			子指令					
数据大小指定	软元件指定形式	软元件存储器扩展指定	ASCII代码 (上段: 字符, 下段: 字符代码)				二进制代码	
字单位	指定代码2字符/编号6位	无指定	0	0	0	0	00H	00H
			30H	30H	30H	30H		
	指定代码4字符/编号8位	有指定	0	0	8	0	80H	00H
			30H	30H	38H	30H		

### ②字软元件块数、位软元件块数

以16进制数指定要读取的软元件的块数。

项目	内容	点数	
		ASCII代码	二进制代码
字软元件块数	指定要读取的字软元件的块数。	字软元件块数+位软元件块数≤120 软元件存储器有扩展指定的情况下, 设为块数×2进行计算。	
位软元件块数	指定要读取的位软元件的块数。		

### ③ 软元件代码、软元件编号、软元件点数

请指定软元件点数，以满足下述条件。

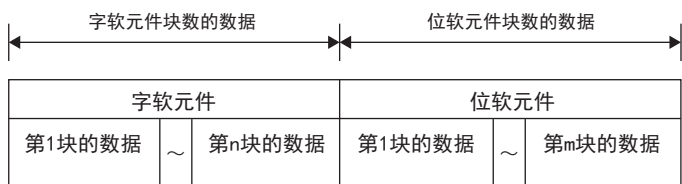
字软元件的各块的合计点数+位软元件的各块的合计点数≤960

项目	内容
字软元件	指定“字软元件块数”中指定的点数的软元件。将“字软元件块数”设为0点的情况下，无需指定。
位软元件	指定“位软元件块数”中指定的点数的软元件。将“位软元件块数”设为0点的情况下，无需指定。

### 要点

指定定时器、累计定时器、计数器的触点以及线圈的情况下，请使用位软元件块。  
按照字软元件→位软元件的顺序进行设定。

## 响应数据



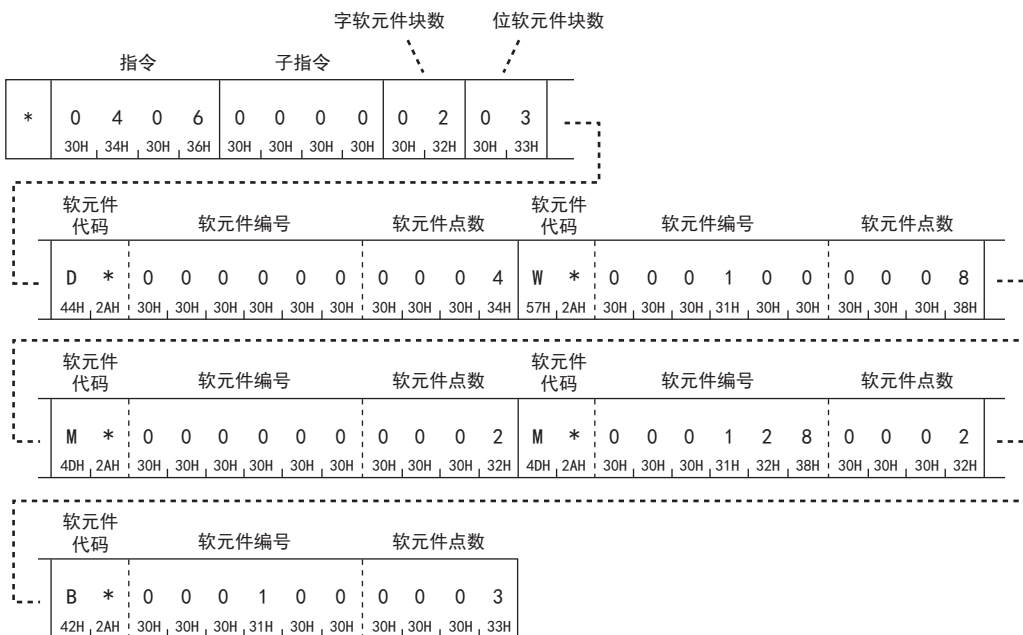
## 通信示例

如下所述，通过软元件读取值。

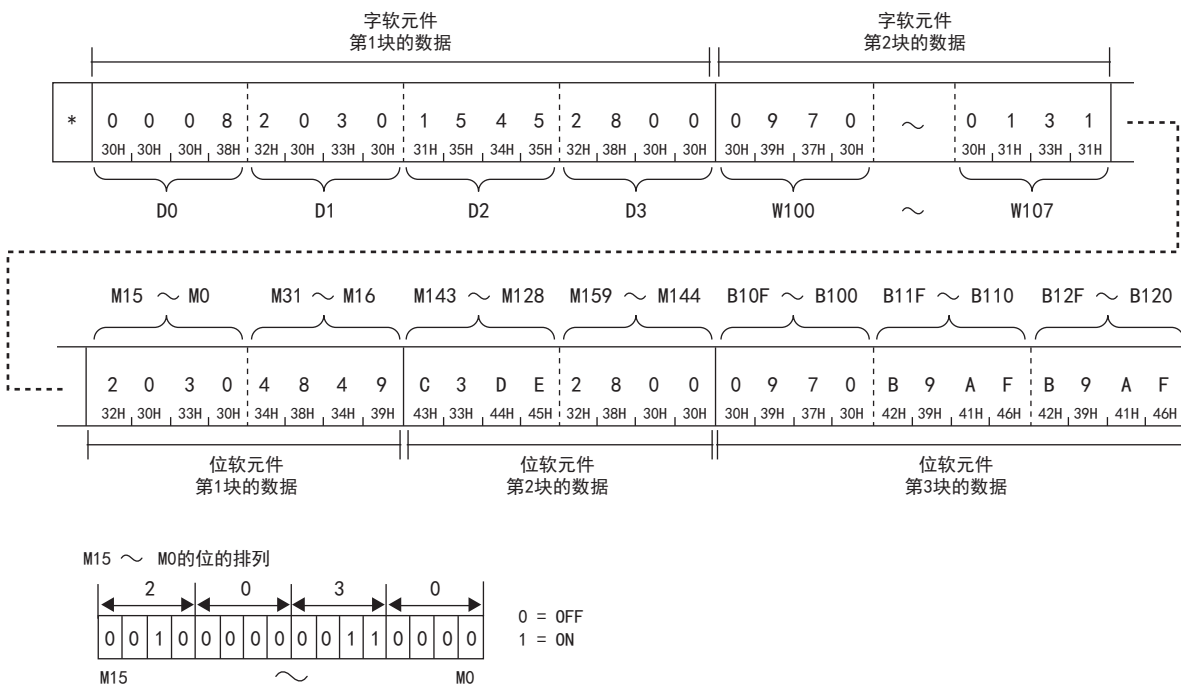
项目	读取内容
字软元件	<ul style="list-style-type: none"> <li>块1: D0~D3 (4点)</li> <li>块2: W100~W107 (8点)</li> </ul>
位软元件	<ul style="list-style-type: none"> <li>块1: M0~M31 (2点)</li> <li>块2: M128~M159 (2点)</li> <li>块3: B100~B12F (3点)</li> </ul>

### ■以ASCII代码进行数据通信时

(请求数据)

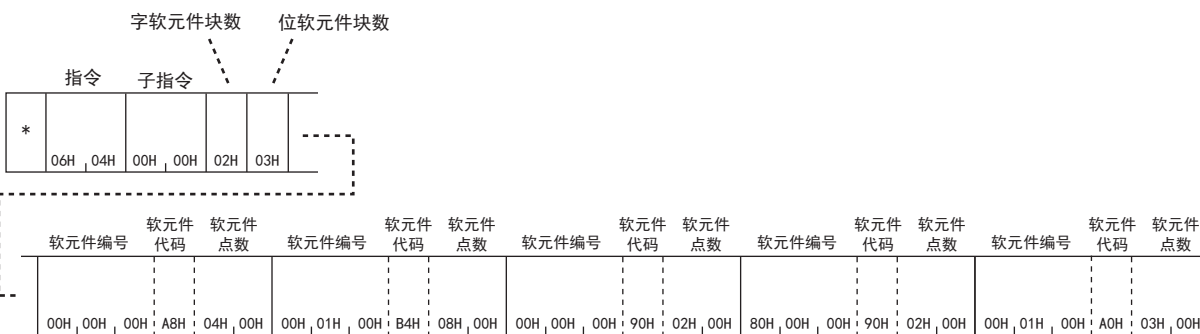


(响应数据)

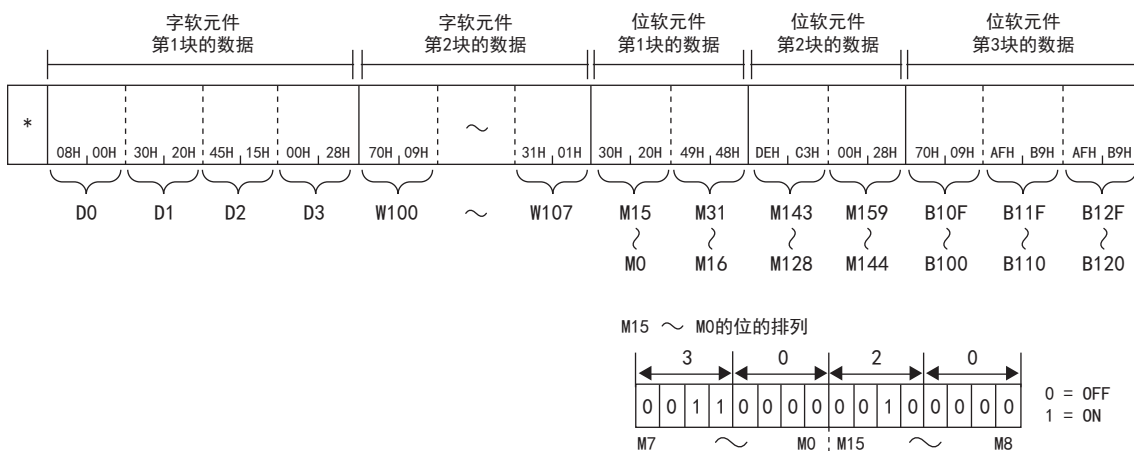


■以二进制代码进行数据通信时

(请求数据)



(响应数据)

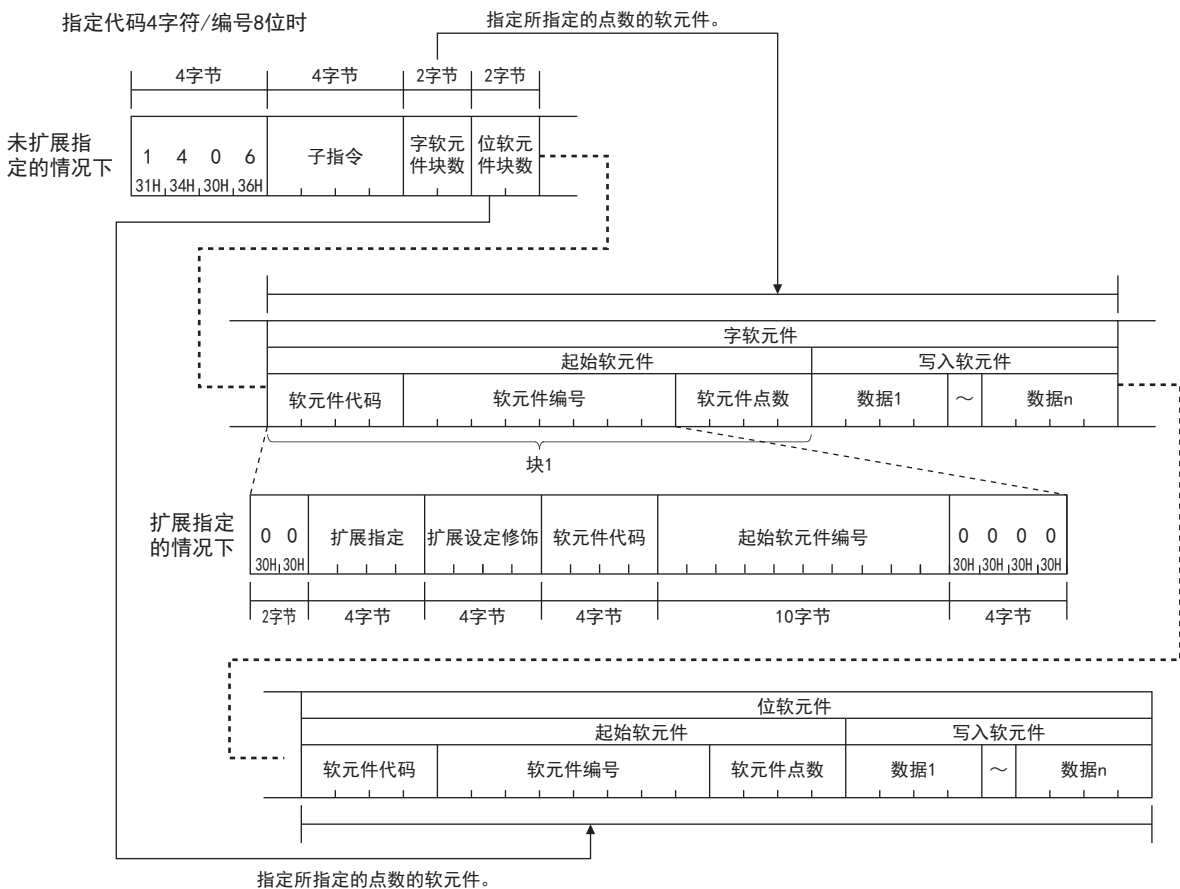
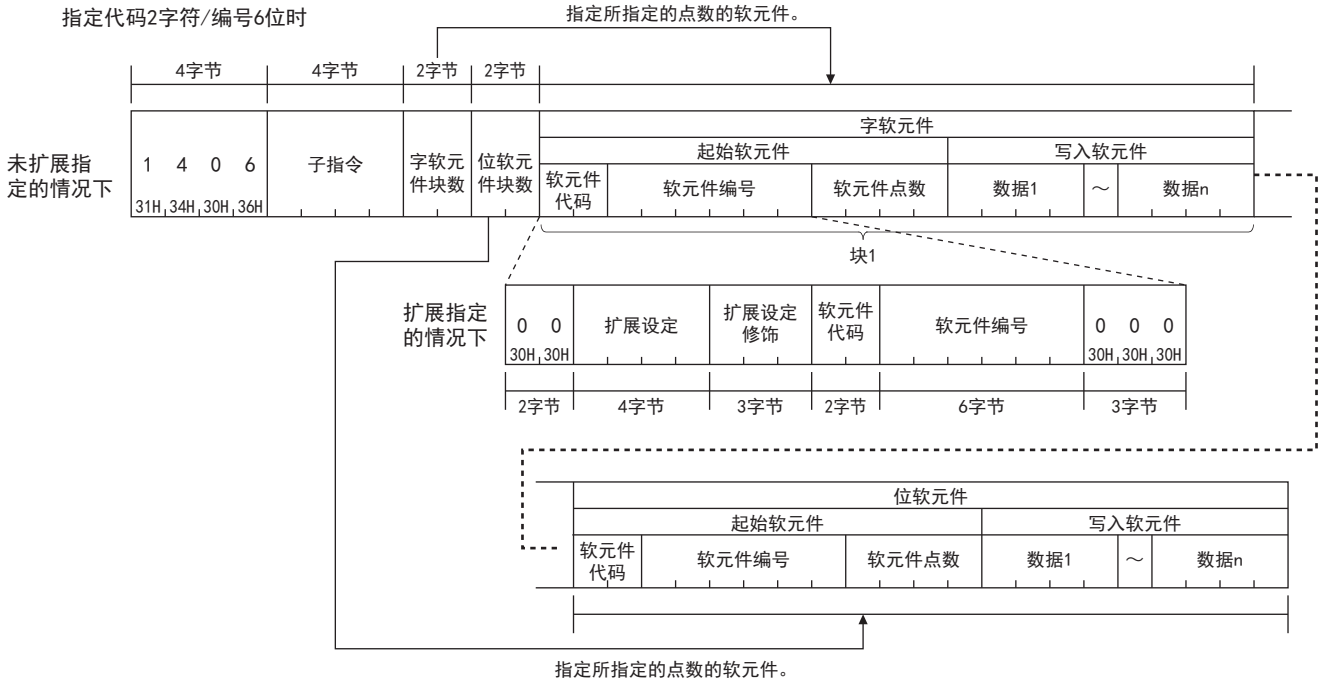


# 批量写入多个块

使用示例，说明将位软元件（1点=16位）和字软元件（1点=1字）的n点作为1块，随机指定多个块进行写入的控制步骤。

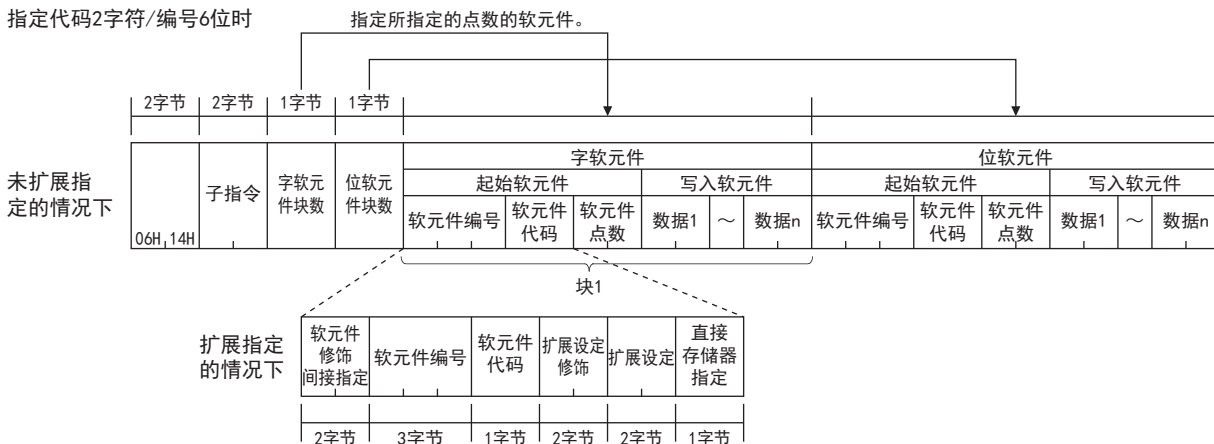
## 请求数据

### ■以ASCII代码进行数据通信时

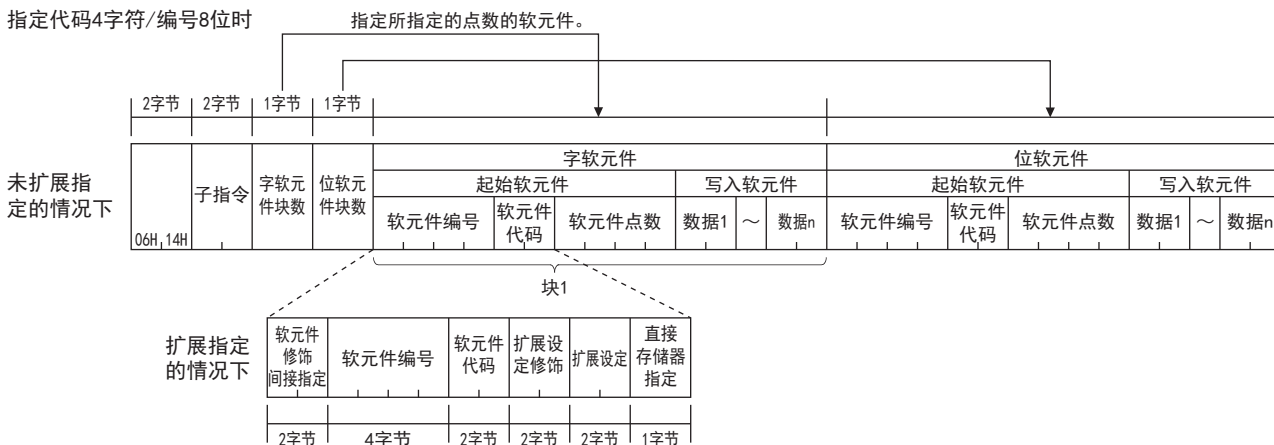


## ■以二进制代码进行数据通信时

指定代码2字符/编号6位时



指定代码4字符/编号8位时



### ①子指令

指定通过项目选择的子指令。

项目			子指令					
数据大小指定	软元件指定形式	软元件存储器扩展指定	ASCII代码 (上段: 字符, 下段: 字符代码)				二进制代码	
字单位	指定代码2字符/编号6位	无指定	0	0	0	0	00H	00H
			30H	30H	30H	30H		
		有指定	0	0	8	0	80H	00H
			30H	30H	38H	30H		
	指定代码4字符/编号8位	有指定	0	0	8	2	82H	00H
			30H	30H	38H	32H		

### ②字软元件块数、位软元件块数

以16进制数指定要写入的软元件的块数。

项目	内容	点数	
		ASCII代码	二进制代码
字软元件块数	指定要写入的字软元件的块数。	字软元件块数+位软元件块数≤120	
位软元件块数	指定要写入的位软元件的块数。	软元件存储器有扩展指定的情况下, 设为块数×2进行计算。	

### ③ 软元件代码、软元件编号、软元件点数

请指定软元件点数，以满足下述条件。

(字软元件块数+位软元件块数) ×4+字软元件的各块的合计点数+位软元件的各块的合计点数≤770

项目	内容
字软元件	指定“字软元件块数”中指定的点数的软元件。将“字软元件块数”设为0点的情况下，无需指定。
位软元件	指定“位软元件块数”中指定的点数的软元件。将“位软元件块数”设为0点的情况下，无需指定。

#### 要点

指定定时器、累计定时器、计数器的触点以及线圈的情况下，请使用位软元件块。  
按照字软元件→位软元件的顺序进行设定。

## 响应数据

没有批量写入多个块指令的响应数据。



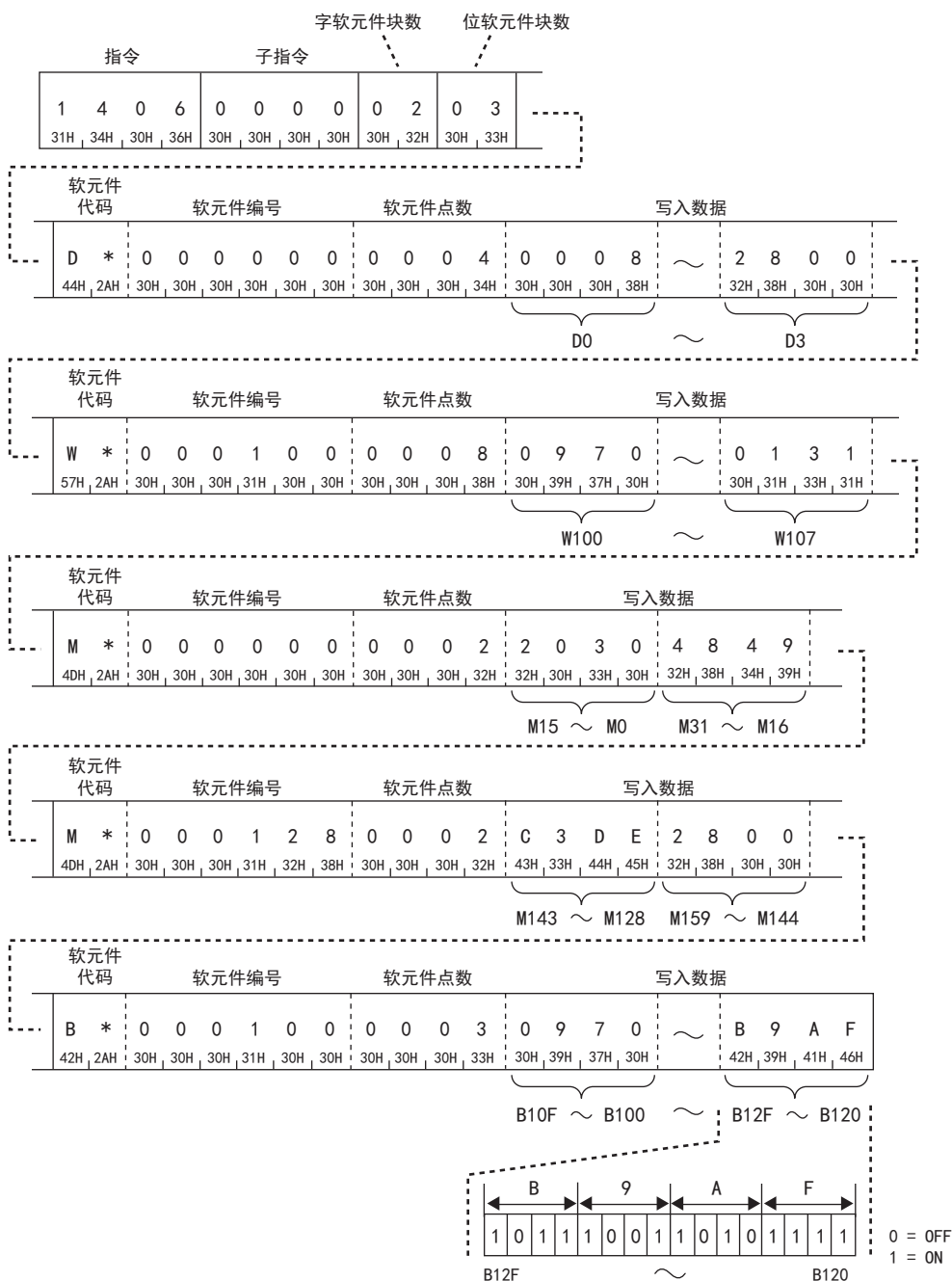
## 通信示例

如下所述，通过软元件写入值。

项目	写入内容
字软元件	<ul style="list-style-type: none"> <li>• 块1: D0~D3 (4点)</li> <li>• 块2: W100~W107 (8点)</li> </ul>
位软元件	<ul style="list-style-type: none"> <li>• 块1: M0~M31 (2点)</li> <li>• 块2: M128~M159 (2点)</li> <li>• 块3: B100~B12F (3点)</li> </ul>

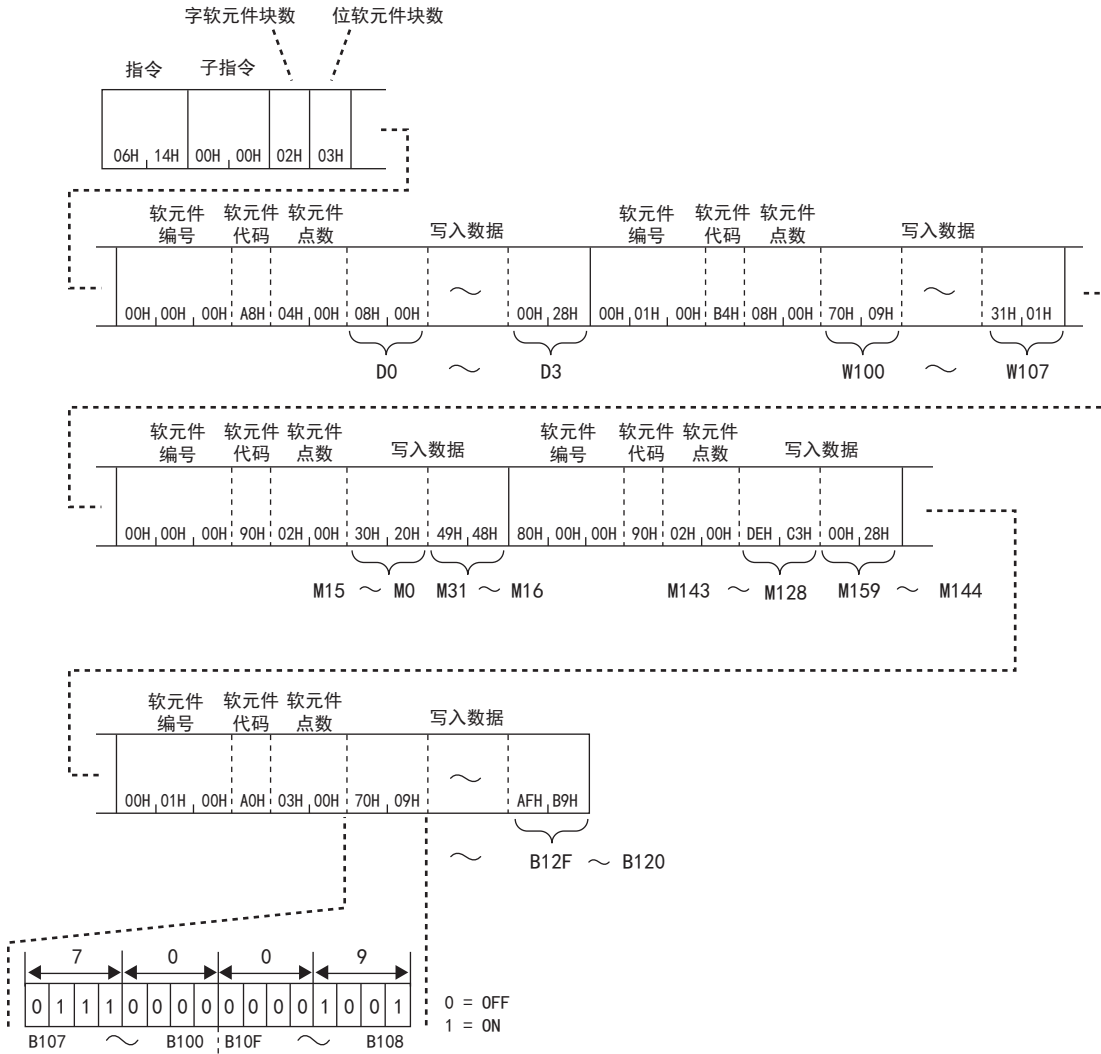
### ■以ASCII代码进行数据通信时

(请求数据)



## ■以二进制代码进行数据通信时

(请求数据)



## 4.3 远程操作

以下对通过来自于对象设备的报文，将MC协议对应设备及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或系统复位的情况下

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模块施加远程口令时

不可以通过对象设备进行远程操作。在访问目标中将发生出错，异常响应被回复到对象设备中。应在解除CPU模块侧的远程口令之后，再次发送请求报文。

#### 通过1次指令可操作的站

可通过1次指令仅对1站进行远程操作。

### 远程RUN

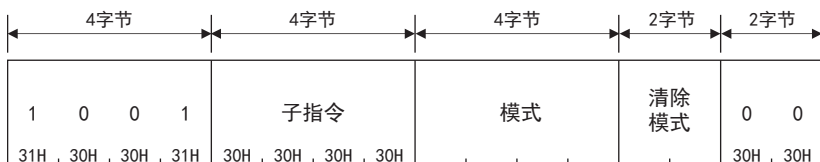
对于访问目标模块执行远程RUN。

#### 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，远程RUN（指令：1001H）虽然正常完成，但是访问目标不变为RUN状态。

### 请求数据

#### ■以ASCII代码进行数据通信时



#### ■以二进制代码进行数据通信时



## ■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 border="1"> <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 border="1"> <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 border="1"> <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 border="1"> <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。

仅00H为有效。

项目	模式						
	ASCII代码	二进制代码					
不清除软元件	<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>30H</td><td>30H</td></tr> </table>	0	0	30H	30H	<table border="1"> <tr><td>00H</td></tr> </table>	00H
0	0						
30H	30H						
00H							

## 响应数据

没有远程RUN指令的响应数据。

## 通信示例

模式通过“不强制执行”进行远程RUN，清除模式通过“不清除软元件”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	0
31H	30H	30H	31H	30H	30H

- 以二进制代码进行数据通信时  
(请求数据)

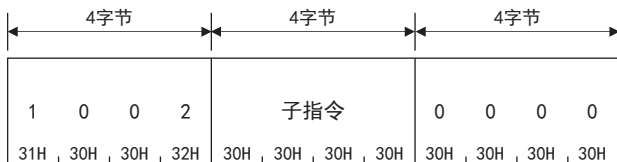
模式		清除模式	
01H	10H	00H	00H
01H	00H	00H	00H

## 远程STOP

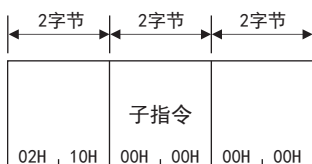
对于访问目标模块执行远程STOP。

### 请求数据

#### ■以ASCII代码进行数据通信时



#### ■以二进制代码进行数据通信时



### 响应数据

没有远程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，通过对象设备发送请求报文。

## 远程PAUSE

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，远程PAUSE（指令：1003H）虽然正常完成，但是访问目标不变为PAUSE状态。

### 请求数据

#### ■以ASCII代码进行数据通信时



#### ■以二进制代码进行数据通信时



## ■ 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。)	<table border="1"> <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 border="1"> <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PAUSE。)	<table border="1"> <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 border="1"> <tr><td>03H</td><td>00H</td></tr> </table>	03H	00H
0	0	0	3									
30H	30H	30H	33H									
03H	00H											

## 响应数据

没有远程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 31H, 30H, 30H, 33H	0 0 0 0 30H, 30H, 30H, 30H	0 0 0 1 30H, 30H, 30H, 31H

### ■ 以二进制代码进行数据通信时

(请求数据)

模式		
03H, 10H	00H, 00H	01H, 00H

## 远程锁存清除

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：

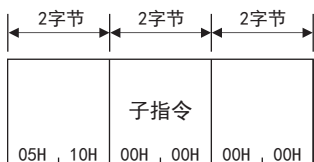
- 不可以进行远程锁存清除。指令将异常完成。
- 应在解除远程STOP或远程PAUSE之后，再执行指令。

## 请求数据

### ■ 以ASCII代码进行数据通信时

4字节	4字节	4字节
1 0 0 5 31H, 30H, 30H, 35H	子指令 30H, 30H, 30H, 30H	0 0 0 0 30H, 30H, 30H, 30H

### ■以二进制代码进行数据通信时



### 响应数据

没有远程锁存清除指令的响应数据。

### 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。

## 远程RESET

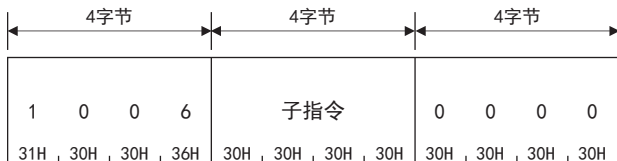
对于访问目标模块执行远程RESET。远程RESET在MC协议对应设备中发生出错时进行出错修复时使用。

### 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。
- 请在进行远程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

#### ■以ASCII代码进行数据通信时



#### ■以二进制代码进行数据通信时



### 响应数据

没有远程RESET指令的响应数据。

### 通信示例

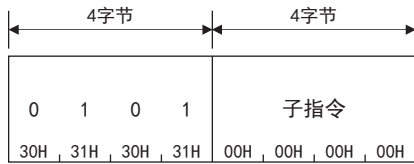
以上述“请求数据”中所示的报文格式，通过外部设备发送请求报文。

# CPU型号读取

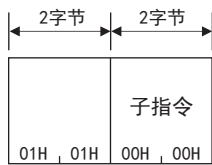
读取访问目标的模块的处理器模块名代码（处理器类型）。

## 请求数据

### ■以ASCII代码进行数据通信时

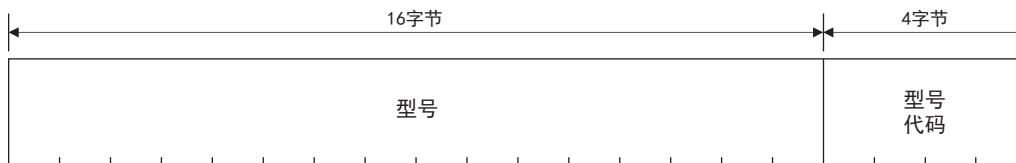


### ■以二进制代码进行数据通信时

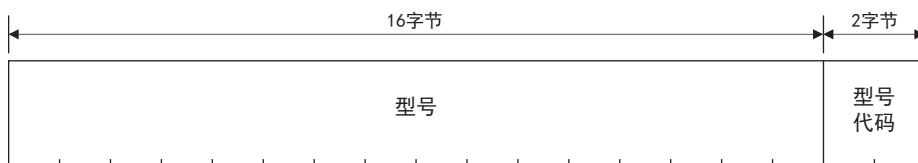


## 响应数据

### ■以ASCII代码进行数据通信时



### ■以二进制代码进行数据通信时



### ■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)。模块型号在以二进制代码进行通信时，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。

型号	型号代码 (16进制)
FX5U-32MR/ES	4A21H
FX5U-64MR/ES	4A23H
FX5U-80MR/ES	4A24H
FX5U-32MT/ES	4A29H
FX5U-64MT/ES	4A2BH
FX5U-80MT/ES	4A2CH
FX5U-32MT/ESS	4A31H
FX5U-64MT/ESS	4A33H
FX5U-80MT/ESS	4A34H
FX5UC-32MT/D	4A91H
FX5UC-32MT/DSS	4A99H

### 要点

CPU模块型号应以型号代码判别。

## 通信示例

### ■以ASCII代码进行数据通信时

(请求数据)

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

(响应数据)

F X 5 U - 3 2 M R / E S	4 A 2 1
46H, 58H, 35H, 55H, 2DH, 33H, 32H, 4DH, 52H, 2FH, 45H, 53H, 20H, 20H, 20H, 20H	34H, 41H, 32H, 31H

### ■以二进制代码进行数据通信时

(请求数据)

01H, 01H	00H, 00H
----------	----------

(响应数据)

F X 5 U - 3 2 M R / E S	
46H, 58H, 35H, 55H, 2DH, 33H, 32H, 4DH, 52H, 2FH, 45H, 53H, 20H, 20H, 20H, 20H	21H, 4AH

## 4.4 出错代码的初始化

对象设备使CPU模块的ERR LED熄灭或使缓冲存储器中存储的出错信息、出错代码初始化的功能。

在要将因对于指令报文的异常响应的响应等而出现的当前出错信息进行初始化，使其恢复为正常时的信息时，或者要将缓冲存储器的出错代码存储区域进行初始化时，使用本功能。

控制步骤图中所示的标志部分的数据项目的排列和内容会根据通信时的帧和形式而有所不同。

### 要点

本功能仅可用于与对象设备连接的CPU模块。

本功能不能用于经由网络系统的其它站的CPU模块。

以下说明通过对象设备使CPU模块的显示LED熄灭或对通信出错信息进行初始化的情况下的指令以及控制步骤内的字符部（以二进制代码进行通信时为数据部）。

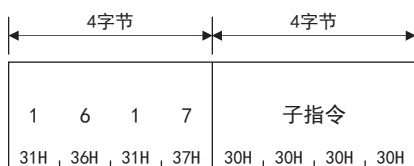
### 指令

功能	指令（子指令）	处理内容	CPU模块的状态		
			STOP中	RUN中	
				可写入设定	不可写入设定
显示LED的熄灭、出错代码的初始化	1617(0000)	进行显示LED的熄灭、出错代码的初始化等。	○	○	○

○：可执行

### 请求数据

#### ■以ASCII代码进行数据通信时



#### ■以二进制代码进行数据通信时



### 响应数据

没有出错代码初始化指令的响应数据。

### 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。

## 4.5 反复测试

反复测试是指测试对象设备与CPU模块的通信功能是否正常运行的功能。以下使用示例，说明使用该功能的情况下的控制步骤。

### 要点

- CPU模块启动时或发生故障时，通过实施本反复测试，能够确认对象设备与CPU模块的连接是否正常、数据通信功能是否正常运行。
- 本功能仅可用于与对象设备连接的CPU模块（包括多点连接站）。本功能不能用于经由网络系统的其它站的CPU模块。

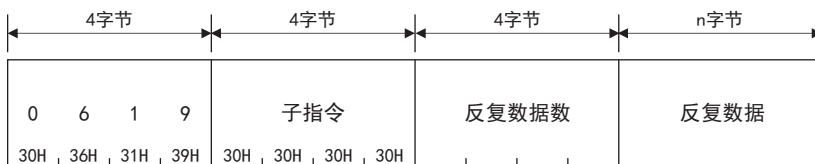
### 指令

功能	指令（子指令）	处理内容	CPU模块的状态		
			STOP中	RUN中	
				可写入设定	不可写入设定
反复测试	0619 (0000)	确认数据通信是否正常运行。	○	○	○

○：可执行

### 请求数据

#### ■以ASCII代码进行数据通信时



- 反复数据数（字节数）  
将字节数转换为ASCII代码4位（16进制数）后使用，并从高位（“0”）开始发送。
- 反复数据  
从起始开始发送最大960字符的半角字符串（“0”～“9”、“A”～“F”）的排列。

#### ■以二进制代码进行数据通信时



- 反复数据数（字节数）  
使用表示字节数的2字节的数值，从低位字节（L：位0～7）开始发送。
- 反复数据  
将半角字符（“0”～“9”、“A”～“F”）的排列的各字符代码设为1字节的数值，从起始开始发送最大960字节。

### 响应数据

对象设备发送的反复数据数和反复数据原样不变地以同样内容回复对象设备。

## 通信示例

以“请求数据”（☞ 81页）中所示的报文格式，通过外部设备发送请求报文。  
将反复数据设为“abcdefghijkl”的示例。

### ■以ASCII代码的通信进行反复测试的情况下

（请求数据）

指令	子指令	反复数据数	反复数据
0 6 1 9	0 0 0 0	0 0 1 2	a b c d e f g h i j k l
30H, 36H, 31H, 39H	30H, 30H, 30H, 30H	30H, 30H, 31H, 32H	61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH

（响应数据）

反复数据数	反复数据
0 0 1 2	a b c d e f g h i j k l
30H, 30H, 31H, 32H	61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH

### ■以二进制代码的通信进行反复测试的情况下

（请求数据）

指令	子指令	反复数据数	反复数据
			a b c d e f g h i j k l
19H, 06H	00H, 00H	12H, 00H	61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH

（响应数据）

反复数据数	反复数据
	a b c d e f g h i j k l
12H, 00H	61H, 62H, 63H, 64H, 65H, 66H, 67H, 68H, 69H, 6AH, 6BH, 6CH

# 附录

## 附1 软元件存储器的扩展指定

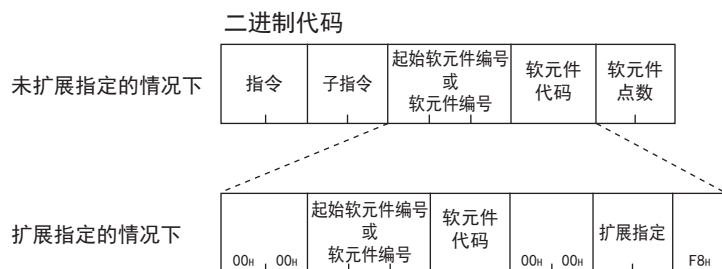
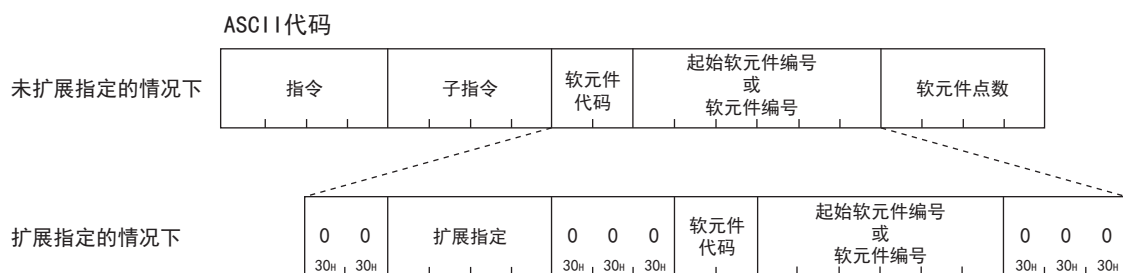
通过将请求数据内的子指令置为008□，可以进行如下所示访问。

- 至模块访问软元件的访问
- 通过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

### 至模块访问软元件的访问

访问MC协议对应设备及智能功能模块的缓冲存储器。

#### 请求数据



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”中指定为0，也可以对软元件代码一览(☞ 33页)中显示的软元件进行访问。但是，“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。

功能	指令
批量读取	0401
批量写入	1401
随机读取	0403
随机写入	1402
多个块批量读取	0406
多个块批量写入	1406

## ■子指令

子指令	ASCII代码	二进制代码										
	<table border="1"> <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	8	0	30H	30H	38H	30H	<table border="1"> <tr><td>80H</td><td>00H</td></tr> </table>	80H	00H
0	0	8	0									
30H	30H	38H	30H									
80H	00H											
	<table border="1"> <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	8	2	30H	30H	38H	32H	<table border="1"> <tr><td>82H</td><td>00H</td></tr> </table>	82H	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。 <table border="1"> <tr><td>U</td><td>□</td><td>□</td><td>□</td></tr> <tr><td>55H</td><td></td><td></td><td></td></tr> </table> <table border="1"> <tr><td>例</td><td>001的情况下</td></tr> <tr><td>U</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>55H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table>	U	□	□	□	55H				例	001的情况下	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。 <table border="1"> <tr><td>□□H</td><td>□□H</td></tr> </table> <table border="1"> <tr><td>例</td><td>001的情况下</td></tr> <tr><td>01H</td><td>00H</td></tr> </table>	□□H	□□H	例	001的情况下	01H	00H
U	□	□	□																						
55H																									
例	001的情况下																								
U	0	0	1																						
55H	30H	30H	31H																						
□□H	□□H																								
例	001的情况下																								
01H	00H																								

## ■软元件代码

指定下述的软元件代码。

类型	软元件代码				软元件编号范围	
	ASCII代码*1		二进制代码			
	指定代码2字符/编号6位时	指定代码4字符/编号8位时	指定代码2字符/编号6位时	指定代码4字符/编号8位时		
字	G*	G***	ABH	AB00H	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

\*1 在软元件字符之后附加“\*”（ASCII代码：2AH）或空格（ASCII代码：20H）。

## ■起始软元件或软元件编号

与未进行扩展指定时的报文相同的格式，以10进制数指定起始软元件或软元件编号。

### 要点

通过使用CPU模块的变址寄存器(Z)或长变址寄存器(LZ)，能够间接指定访问对象的软元件编号。（☞ 85页 以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

与未进行扩展指定的情况下相同。

## 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			软元件 代码	起始软元件编号 或 软元件编号
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)

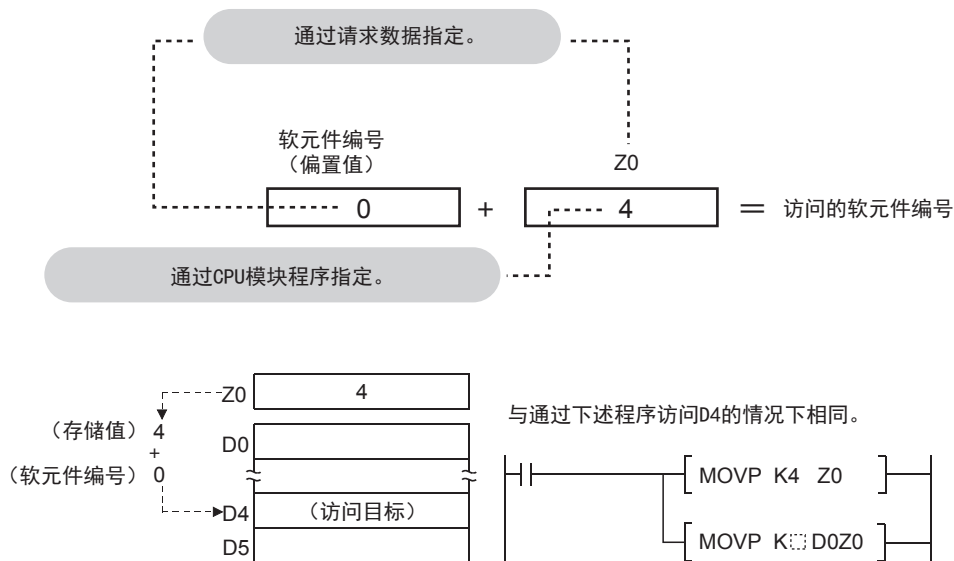
子指令	起始软元件编号 或 软元件编号		软元件 代码	扩展指定		
80H, 00H	00H, 00H	01H, 00H, 00H	ABH	00H, 00H	03H, 00H	F8H

## 以变址寄存器或长变址寄存器间接指定软元件编号的访问

访问软元件时，能够通过变址寄存器，或长变址寄存器间接指定软元件编号。  
通过利用CPU模块的程序变更变址寄存器或长变址寄存器的值，以1个报文切换访问目标。

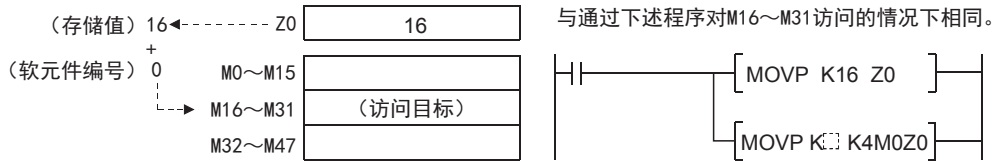
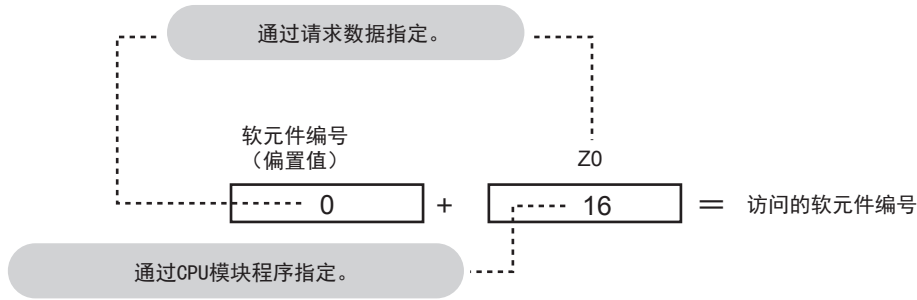
### 例

以D0和Z0的指定访问D4的情况下



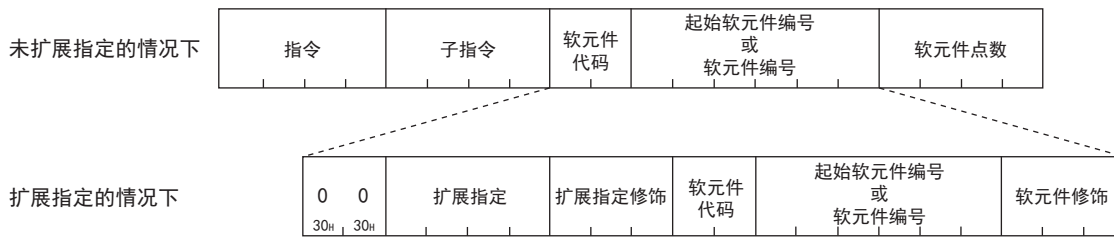
**例**

以M0和Z0的指定访问M16~M31的情况（字单位）

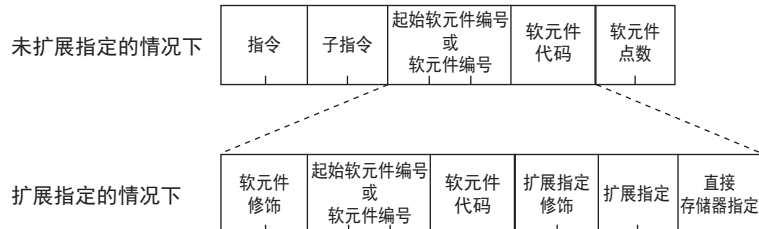


## 请求数据

### ASCII代码

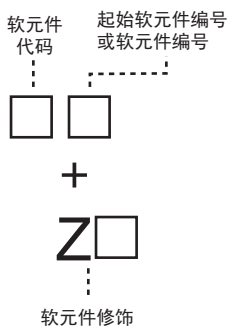


### 二进制代码



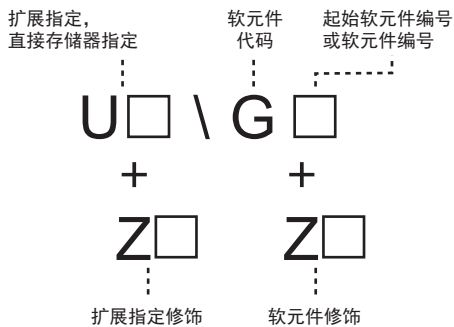
软元件、变址寄存器、长变址寄存器、请求数据的对应如下所示。

- 模块访问软元件以外





• 模块访问软元件



**要点**

“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。

功能	指令
随机读取	0403
随机写入	1402

**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	<table border="1"> <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	8	1	30H	30H	38H	31H	<table border="1"> <tr><td>81H</td><td>00H</td></tr> </table>	81H	00H
	0	0	8	1								
30H	30H	38H	31H									
81H	00H											
<table border="1"> <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	8	3	30H	30H	38H	33H	<table border="1"> <tr><td>83H</td><td>00H</td></tr> </table>	83H	00H	
0	0	8	3									
30H	30H	38H	33H									
83H	00H											
以字单位进行访问的情况下	<table border="1"> <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	8	0	30H	30H	38H	30H	<table border="1"> <tr><td>80H</td><td>00H</td></tr> </table>	80H	00H
	0	0	8	0								
30H	30H	38H	30H									
80H	00H											
<table border="1"> <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	8	2	30H	30H	38H	32H	<table border="1"> <tr><td>82H</td><td>00H</td></tr> </table>	82H	00H	
0	0	8	2									
30H	30H	38H	32H									
82H	00H											

**扩展指定**

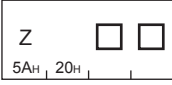



指定访问对象的模块编号。

通过“扩展指定修饰”，对模块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。



项目	ASCII代码	二进制代码																				
模块访问软元件	<p>以16进制数（ASCII代码3位）指定模块编号。</p> <p><b>例</b> 001的情况下</p> <table border="1"> <tr><td>U</td><td>□</td><td>□</td><td>□</td></tr> <tr><td>55H</td><td></td><td></td><td></td></tr> </table> <table border="1"> <tr><td>U</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>55H</td><td>30H</td><td>30H</td><td>31H</td></tr> </table>	U	□	□	□	55H				U	0	0	1	55H	30H	30H	31H	<p>以16进制数（2字节）指定模块编号。</p> <p><b>例</b> 001的情况下</p> <table border="1"> <tr><td>□□H</td><td>□□H</td></tr> </table> <table border="1"> <tr><td>01H</td><td>00H</td></tr> </table>	□□H	□□H	01H	00H
U	□	□	□																			
55H																						
U	0	0	1																			
55H	30H	30H	31H																			
□□H	□□H																					
01H	00H																					
上述以外的软元件	<p>指定0。</p> <table border="1"> <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	<p>指定0。</p> <table border="1"> <tr><td>00H</td><td>00H</td></tr> </table>	00H	00H										
0	0	0	0																			
30H	30H	30H	30H																			
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/iQ-F系列的模块的情况下，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)的编号。（指定范围：0~24） 	以16进制数指定变址寄存器(Z)的编号。（指定范围：00H~18H） 
0081 0080	以10进制数（ASCII代码2位）指定变址寄存器(Z)的编号。（指定范围：0~24） 	以16进制数指定变址寄存器(Z)的编号。（指定范围：00H~18H） 

访问目标为MELSEC-Q/L系列的模块的情况下，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） 	以16进制数指定变址寄存器的编号。（指定范围：00H~0FH） 

### 要点

对扩展指定修饰长变址寄存器(LZ)不能使用。

## ■软元件代码

指定要访问的软元件的代码。（☞ 33页 软元件代码一览）

对模块访问软元件进行访问的情况下，指定下述软元件代码。

类型	软元件代码				软元件编号范围	
	ASCII代码*1		二进制代码			
	指定代码2字符 /编号6位时	指定代码4字符 /编号8位时	指定代码2字符 /编号6位时	指定代码4字符 /编号8位时		
字	G*	G***	ABH	AB00H	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

\*1 在软元件字符之后附加“\*”（ASCII代码：2AH）或空格（ASCII代码：20H）。

## ■起始软元件或软元件编号

以与未进行扩展指定时的报文相同的格式，指定起始软元件或软元件编号。

通过“软元件修饰”，对软元件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。

### ■软元件修饰

将“起始软元件或软元件编号”中指定的值置为偏置值，通过变址寄存器间接指定软元件编号的情况下，指定变址寄存器编号。

访问目标为MELSEC iQ-R/iQ-F系列的模块的情况下，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)的编号。（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)的编号。（指定范围：0~12）  <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">Z □ □ 5Ah, 20h, ,</div> <div style="border: 1px solid black; padding: 2px;">L Z □ □ 4Ch, 5Ah, ,</div> </div>	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）  <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">□□H, 40h</div> <div style="border: 1px solid black; padding: 2px;">□□H, 80h</div> </div>
0081 0080	以10进制数（ASCII代码2位）指定变址寄存器(Z)的编号。（指定范围：0~24）  <div style="border: 1px solid black; padding: 2px;">Z □ □ 5Ah, ,</div>	以16进制数指定变址寄存器(Z)的编号。（指定范围：00H~18H）  <div style="border: 1px solid black; padding: 2px;">□□H, 40h</div>

\*1 变址寄存器(Z)的软元件修饰范围为-32768~32767。软元件修饰范围超过-32768~32767的情况下，请使用长变址寄存器(LZ)。访问目标为MELSEC-Q/L系列的模块的情况下，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）  <div style="border: 1px solid black; padding: 2px;">Z □ □ 5Ah, ,</div>	以16进制数指定变址寄存器的编号。（指定范围：00H~0FH）  <div style="border: 1px solid black; padding: 2px;">□□H, 40h</div>

### ■直接存储器指定（仅在以二进制代码进行通信时）

访问模块访问软元件的情况下，指定软元件的种类。

项目	二进制代码
模块访问软元件	指定F8H。
上述以外	指定00H。

### 响应数据

与未进行扩展指定的情况下相同。

### 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 30h, 30h, 38h, 30h	0 0 30h, 30h	0 0 0 0 30h, 30h, 30h, 30h	0 0 0 30h, 30h, 30h	D * 0 0 0 1 0 0 44h, 2Ah, 30h, 30h, 30h, 31h, 30h, 30h	Z 0 4 5Ah, 30h, 34h

- 以二进制代码进行数据通信时  
(请求数据)

子指令	软元件修饰	起始软元件编号或软元件编号	软元件代码	扩展指定	直接存储器指定
80h, 00h	04h, 40h	64h, 00h, 00h	A8h	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

能够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，从对象设备访问“@D0”且访问D100的情况下

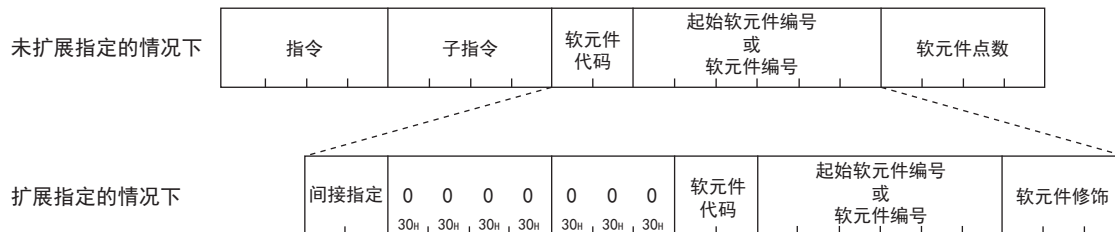
在CPU模块侧使用ADRSET指令，将D100的地址存储到D0中。



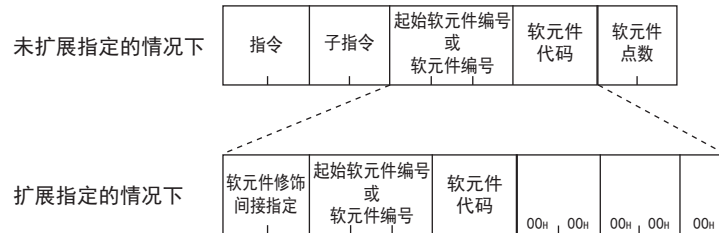
通过在请求数据中指定“@D0”，可以间接访问D100。

## 请求数据

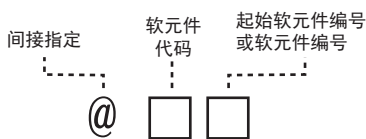
### ASCII 代码



### 二进制代码



间接指定软元件、请求数据的对应如下所示。



## 要点

- “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。
- 与间接指定由变址寄存器的软元件修饰不能并用。

## 指令

可以通过下述指令进行访问。

功能	指令
随机读取	0403
随机写入	1402

### ■子指令

ASCII代码	二进制代码										
<table border="1"> <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	8	0	30H	30H	38H	30H	<table border="1"> <tr> <td>80H</td> <td>00H</td> </tr> </table>	80H	00H
0	0	8	0								
30H	30H	38H	30H								
80H	00H										

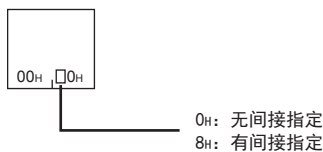
### ■间接指定

指定间接指定软元件的“@”的部分。间接指定仅在字软元件中可以指定。

以ASCII代码进行数据通信时

0	@
30H	40H

以二进制代码进行数据通信时



### ■软元件代码（间接指定仅可指定字软元件代码）

指定要访问的软元件的代码。（☞ 33页 软元件代码一览）

### ■起始软元件或软元件编号

以与未进行扩展指定时的报文相同的格式，指定起始软元件或软元件编号。

## 响应数据

与未进行扩展指定的情况下相同。

## 通信示例

访问@D0。（@D0为D100的间接指定。）

在执行指令前，预先通过下述程序将D100的地址存储至D0。

```
M300 ─────────────────────────────────── [ADRSET D100 D0 ]
```

- 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 0	0 0 0
30H 30H 38H 30H	30H 40H	30H 30H 30H 30H	30H 30H 30H 30H	30H 30H 30H
		D *	0 0 0 0 0 0	0 0 0
		44H 2AH	30H 30H 30H 30H 30H 30H	30H 30H 30H

- 以二进制代码进行数据通信时  
(请求数据)

子指令	间接指定	起始软元件编号 或 软元件编号	软元件代码
80H 00H	00H 80H	00H 00H 00H	A8H
		00H 00H	00H 00H

## 附2 MC协议的CPU模块侧的处理时间

显示通过MC协议的通信从对象设备访问CPU模块的情况下，对CPU模块侧的扫描时间的介入时间以及处理所需的扫描次数。通过MC协议的通信，在CPU模块对于来自对象设备的处理请求执行RUN中的情况下，处理以END为单位1次能够处理的点数。

项目	指令	子指令	访问点数1) / 2)	介入时间[ms]*2 (扫描时间的延长)		处理所需的扫描次数
				访问点数1)时	访问点数2)时	
批量读取	0401	0001	1/3584	0.04	0.87	1
		0000	1/960	0.04	0.73	1
批量写入	1401	0001	1/3584	0.04	0.95	1
		0000	1/960	0.05	0.83	1
随机读取	0403	0000	1/192	0.06	2.40	1
随机写入	1402	0001	1/188	0.02	1.66	1
		0000	1/160*1	0.02	1.85	1
多个块批量读取	0406	0000	1/960	0.05	0.74	1
多个块批量写入	1406	0000	1/960	0.04	0.67	1
CPU型号读取	0101	0000	(1站)	0.03	—	1

\*1 仅指定字访问点数进行访问的情况下的处理时间。

\*2 GX Works3的[CPU参数]的[服务处理设置]的[软元件·标签访问服务处理设置]中，将[设置处理次数]设定为1次的情况下的处理时间。

### 要点

- 处理所需的扫描次数

CPU模块在END处理中仅处理任意1个指令。GX Works3、各模块等同时访问CPU模块的情况下，会待机至其它处理结束，因此处理所需的扫描次数会进一步增加。

- 减少对扫描时间的介入时间的方法

想要减少对扫描时间的介入时间的情况下，请在[CPU参数]的[服务处理设置]的[软元件·标签访问服务处理设置]中，调整CPU模块的服务处理次数。

( MELSEC iQ-F FX5用户手册(应用篇))

- 扫描时间的延长影响控制的情况下

请以较少的点数分成多次进行访问。

# 索引

## [A]

ASCII代码 . . . . . 14

## [B]

报文格式 . . . . . 15  
本站站号 . . . . . 27

## [C]

出错代码 . . . . . 22

## [E]

二进制代码 . . . . . 14

## [F]

附加代码 . . . . . 19

## [H]

和校验码 . . . . . 20  
缓冲存储器 . . . . . 8

## [K]

可编程控制器编号 . . . . . 26  
控制代码 . . . . . 18

## [L]

链接时间 . . . . . 12  
连接站(本站) . . . . . 8

## [M]

MC协议 . . . . . 8  
模块访问软元件 . . . . . 8

## [Q]

其它站 . . . . . 8  
请求目标模块I/O编号 . . . . . 27  
请求目标模块站号 . . . . . 27

## [R]

软元件代码 . . . . . 31

## [S]

SLMP . . . . . 8  
SLMP对应设备 . . . . . 8

## [W]

外部设备 . . . . . 8  
网络编号 . . . . . 26

## [X]

形式 . . . . . 14  
形式1 . . . . . 15  
形式4 . . . . . 16  
形式5 . . . . . 17

## [Z]

站号 . . . . . 25  
帧 . . . . . 14  
帧识别编号 . . . . . 20  
中继站 . . . . . 8

## [ 数字 ]

1C帧 . . . . . 14  
2C帧 . . . . . 14  
3C帧 . . . . . 14, 24  
4C帧 . . . . . 14, 23

# 修订记录

制作日期	版本号	内容
2015年2月	A	制作初版

在本书中，并没有对工业知识产权及其它权利的执行进行保证，也没有对执行权进行承诺。对于因使用本书中所记载的内容而引起的工业知识产权上的各种问题，本公司将不负任何责任。

© 2015 MITSUBISHI ELECTRIC CORPORATION



# 关于保修

在使用时，请务必确认一下以下的有关产品保证方面的内容。

## 1. 免费保修期和免费保修范围

在产品的免费保修期内，如是由于本公司的原因导致产品发生故障和不良（以下统称为故障）时，用户可以通过当初购买的代理店或是本公司的服务网络，提出要求免费维修。

但是、如果要求去海外出差进行维修时，会收取派遣技术人员所需的实际费用。

此外，由于更换故障模块而产生的现场的重新调试、试运行等情况皆不属于本公司责任范围。

### 【免费保修期】

产品的免费保修期为用户买入后或是投入到指定的场所后的12个月以内。但是，由于本公司的产品出厂后一般的流通时间最长为6个月，所以从制造日期开始算起的18个月为免费保修期的上限。

此外，维修品的免费保修期不得超过维修前的保证时间而变得更长。

### 【免费保修范围】

- (1) 只限于使用状态、使用方法以及使用环境等都遵照使用说明书、用户手册、产品上的注意事项等中记载的条件、注意事项等，在正常的状态下使用的情况。
- (2) 即使是在免费保修期内，但是如果属于下列的情况的话就变成收费的维修。
  - ① 由于用户的保管和使用不当、不注意、过失等等引起的故障以及用户的硬件或是软件设计不当引起的故障。
  - ② 由于用户擅自改动产品而引起的故障。
  - ③ 将本公司产品装入用户的设备中使用时，如果根据用户设备所受的法规规定设置了安全装置或是行业公认应该配备的功能构造等情况下，视为应该可以避免的故障。
  - ④ 通过正常维护·更换使用说明书等中记载的易耗品（电池、背光灯、保险丝等）可以预防的故障。
  - ⑤ 即使按照正常的使用方法，但是继电器触点或是触点寿命的情况。
  - ⑥ 由于火灾、电压不正常等不可抗力导致的外部原因，以及地震、雷电、洪水灾害等天灾引起的故障。
  - ⑦ 在本公司产品出厂时的科学技术水平下不能预见的原因引起的故障。
  - ⑧ 其他、认为非公司责任而引起的故障。

## 2. 停产后的收费保修期

(1) 本公司接受的收费维修品为产品停产后的7年内。有关停产的信息，都公布在本公司的技术新闻等中。

(2) 不提供停产后的产品（包括附属品）。

## 3. 在海外的服务

对于海外的用户，本公司的各个地域的海外FA中心都接收维修。但是，各地的FA中心所具备的维修条件有所不同，望用户谅解。

## 4. 对于机会损失、二次损失等保证责任的免除

无论是否在保修期内，对于不是由于本公司的责任而导致的损害；以及由于本公司产品的故障导致用户或第三方的机会损失、利益损失，无论本公司是否可以预见，由于特别的原因导致出现的损害、二次损害、事故赔偿，损坏到本公司以外产品，以及对于用户的更换产品工作，现场机械设备的重新调试、启动试运行等其他业务的补偿，本公司都不承担责任。

## 5. 产品规格的变更

产品样本、手册或技术资料中所记载的规格有时会未经通知就变更，还望用户能够预先询问了解。

## 6. 关于产品的适用范围

(1) 使用本公司MELSEC iQ-F/FX/F微型可编程控制器时，要考虑到万一可编程控制器出现故障·不良等情况时也不会导致重大事故的使用用途，以及在出现故障·不良时起到作用。将以上这些作为条件加以考虑。在设备外部系统地做好后备或是安全功能。

(2) 本公司的可编程控制器是针对普通的工业用途而设计和制造的产品。因此，在各电力公司的原子能发电站以及用于其他发电站等对公众有很大影响的用途中，以及用于各铁路公司以及政府部门等要求特别的质量保证体系的用途中时，不适合使用可编程控制器。

此外，对于航空、医疗、燃烧、燃料装置、人工搬运装置、娱乐设备、安全机械等预计会对人身生命和财产产生重大影响的用途，也不适用可编程控制器。

但是，即使是上述的用途，用户只要事先与本公司的营业窗口联系，并认可在其特定的用途下可以不要求特别的质量时，还是可以通过交换必须的资料后，选用可编程控制器的。

# 商标

---

Microsoft<sup>®</sup>、Windows<sup>®</sup>是美国Microsoft Corporation的美国以及其他国家中的注册商标或者商标。

Ethernet是美国Xerox Corporation的注册商标。

MODBUS<sup>®</sup>是Schneider Electric SA的注册商标。

其他的公司名称、产品名称都是各个公司的商标和注册商标。



Manual number: JY997D60901A

# **MITSUBISHI ELECTRIC CORPORATION**

HEAD OFFICE: TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN

记载的规格可能发生变更，恕不另行通知。