Q系列CPU

MITSUBISHI

用户参考手册

(功能解说, 程序基础)



可编程控制器

MELSEC-Q

Q02CPU
Q02HCPU
Q06HCPU
Q12HCPU
Q25HCPU

● 安全注意事项●

（使用设备前请阅读本说明）

使用本产品前，请仔细阅读本手册及本手册提及的相关资料，正确操作并注意安全。
在本手册中，安全守则的等级分为“危险”和“小心”。

⚠ 危险	表示错误操作可能造成灾难性后果，引起死亡或重伤事故。
⚠ 小心	表示错误操作可能造成危险后果，引起人员受轻伤或中度伤害或财产损失。

注意根据情况不同，即使⚠小心这一级也可能引发严重后果。
所以对两级注意事项，都须遵照执行，因为它们对于人员安全是非常重要的。

仔细保管本手册，请放在操作人员易于取阅的地方，并应将本手册交给最终使用者。

[设计安全要求]

⚠ 危险

- 在 PLC 外部安装一条安全回路，即使在外部电源和 PLC 模块发生问题时也能保证整个系统的安全。否则，错误的输出或错误操作会引发事故。

  (1) 在 PLC 外部安装一条保护机械损坏的互锁回路，诸如紧急停车，保护回路，上下限位开关和正反向互锁操作等。

  (2) 当 PLC 检测到以下问题时，在（a）种情况下，将停止运算并切断所有的输出。在（b）种情况下，将停止运算并按参数设定维持或切断所有的输出。

  (a) 电源模块具有过电流保护设备或过电压保护设备。

  (b) PLC 的 CPU 具有自诊断功能，诸如警戒定时器出错，检测各种问题等。

  此外当产生 PLC 的 CPU 不能检测到的问题时，输出将接通，诸如 I/O 控制器。在 PLC 外部建立一条失效安全电路，以保证设备在发生问题时安全操作。参见 QCPU（Q 模式）用户手册中的“装入和安装”（硬件设计/维护和检修）的失效安全电路范例。

  (3) 当输出模块中的继电器或晶体管发生故障时，输出可保持接通或断开。因此要安装一条外部监控回路，以监控任何可能会发生严重故障的单项输出。

A - 1

[设计安全要求]

⚠ 危险

- 当过电流超过额定值或因短路引起负荷电流在输出模块中存在较长时间时，会产生烟雾或起火。为防止此类事故发生，要配置一条外部安全电路，诸如熔断器等。
- 建立一条回路，当 PLC 主模块的电源接通时，该回路也接通外部电源。但如果首先是外部电源接通，则可能造成错误输出或误操作。
- 当数据通讯中存在通信问题时，请参见用于每个操作站与操作状况的相应数据通讯手册。
  否则会产生错误输出或误操作。
- 在将外围设备连接到 CPU 模块或将个人计算机之类连接到智能功能模块以便在运行中的 PLC 上进行控制时（数据改变），要在顺控程序中配置一条互锁回路，以保证整个系统始终安全运行。另外，在运行中的 PLC 上进行其它控制前（程序改变，操作状态改变（状态控制）），要小心地阅读手册并充分保证安全。
  特别是从一外部设备对远程 PLC 进行上述控制时，不能对由于数据通信故障产生的 PLC 的故障立即采取措施。
  在顺控程序中增设一条互锁电路，在外部装置和 PLC 的 CPU 之间要预先确定出现通信故障时应该采取修正等措施。

⚠ 小心

- 不得将控制线或通讯电缆与主回路或电源线捆成一束，或相互靠近。相互之间的安装间距应为 100mm(3.94 英寸)以上。
  否则会产生噪音，并引起误操作。
- 当控制部件如灯负荷，加热器或电磁线圈使用输出模块时，在输出接通和切断时会产生大电流（大约十倍于正常条件下的电流值）。
  要采取措施，诸如更换一个具有足够额定电流值的模块。

[安装注意事项]

⚠ 小心
• 在符合本手册所列出的通用规格的环境中使用 PLC。在超出通用规格要求范围的环境下使用 PLC，可能会产生电击，起火，误操作，损坏或产品劣化。 • 按下模块底部的模块安装杆，可靠地将模块的固定挂钩插入基板的固定孔中。模块不正确的安装会引发失效，故障或脱落。 在有较大振动的环境中使用 PLC 时，要用螺钉固定模块。并以规定范围的扭矩拧紧螺钉。 螺钉松动会导致脱落，短路或失效。 螺钉太紧又会因损坏螺钉或模块而产生脱落，短路或失效。 • 在安装较多电缆时，要保证基板和模块连接器正确安装。 安装后要检查是否松动。 连接不良会产生输入或输出故障。 • 可靠地将存储卡装入存储卡连接口中。 装入后检查是否有浮起。 浮起会产生由接触故障引起的失效。 • 在装入和卸下模块之前应完全切断外部电源。 否则会产生电击和损坏产品。 • 不得直接接触模块的导电量件或电子部件。 接触导电量部件会产生操作故障或损坏模块。

[接线注意事项]

⚠ 危险
• 在安装或更换接线时，应完全切断外部电源。 不完全切断电源会产生电击或损坏产品。 • 在安装和接线工作完成后，接通电源操作模块时，要保证模块的接线端子盖正确盖上。 不盖上接线端子盖会导致电击。

A - 3

[接线注意事项]

⚠ 小心
• 保证 FG 接线端子和 LG 接线端子通过保护接地导线接地。否则会产生电击和误操作。 • 在对 PLC 接线时，要检查产品的额定电压和接线端子的布置，以确保正确无误。 当连接的电源不同于额定值或产品接线错误，将导致起火或损坏。 • 外部联接必须用专用工具压接或压焊，或正确焊接。 接线不良将导致短路，起火，或误操作。 • 用专用的扭力螺钉起子拧紧接线的端子螺钉。 如果接线端子的螺钉松动，会产生短路，起火或误操作。如果接线端子的螺钉拧得太紧，会损坏螺钉及/或模块，导致脱落，短路或故障。 • 保证没有诸如锯屑、残头等异物掉进模块。这些异物将导致起火，损坏或误操作。 • 在模块顶部有一条入口保护标签，是防止焊渣等异物在接线时从入口处掉进模块中的。 接线时不得撕去保护标签。 在启动系统操作之前，确保撕去该标签以有利于散热。

[启动和维护安全要求]

⚠ 危险
• 在通电时不得触摸接线端子。 触摸端子将产生电击和误操作。 • 正确地连接电池。对电池不得充电，拆解，加热，或置于有火处，及短路或焊接电池。 电池使用不当将产生过热开裂，从而引起人身伤害和起火。 • 在清洁模块或重新拧紧接线端子或模块安装螺钉时，应断开外部电源的各个相位。否则会导致电击。 接线端子螺钉松动会引发短路和故障。 螺钉太紧会损坏螺钉及/或模块，导致脱落，短路或故障。

[启动和维护注意事项]

⚠ 小心
• 对运行中的 CPU 模块进行在线操作连接到外围设备上时（特别是当改变量据和操作状态时），必须仔细阅读本手册并对安全措施充分检查之后方可进行。 操作失误将导致模块的损坏和各种问题。 • 不得拆解或修改模块。 否则将产生故障，误操作，人身伤害或起火。 • 必须在距 PLC 25cm（9.85 英寸）之外使用手机或 PHS。 否则将引发故障。 • 在安装或拆下模块时，要切断外部电源的各相位。 不切断外部电源，会造成模块的故障或损坏。

[报废处理注意事项]

⚠ 小心
• 报废处理时，将本产品当成工业废料处理。

A - 5

修订

* 手册编号见背面左下方

印刷日期	*手册编号	修订
12,1999	SH(NA)-080232C–A	第一版

日本手册版号 SH-080020-D

本手册不授予任何工业产权或任何其它类型的产权，也不授予专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业产权的任何问题不承担责任。

© 1999 年三菱电机

A - 6

说明

感谢你选用三菱公司 MELSEC–Q 系列通用可编程控制器。
请仔细阅读本手册，以便使用该设备时获得最佳效果。

目录

安全注意事项 ····· A–1
修订 ····· A–6
目录 ····· A–7
关于手册 ····· A–17
如何使用本手册 ····· A–18
关于通用术语和缩写语 ····· A–19

1 综述 1– 1 至 1–11

1.1 特点 ····· 1–2
1.2 程序 ····· 1–5
1.3 简易的可编程软元件和指令 ····· 1–8

2 对单 CPU 系统的系统配置 2– 1 至 2–6

2.1 系统配置 ····· 2–1
2.2 系统安全注意事项 ····· 2–4
2.3 确认系列编号和功能版 ····· 2–6

3 性能规格 3– 1 至 3–3

4 顺控程序配置和执行条件 4– 1 至 4–51

4.1 顺控程序 ····· 4–1
4.1.1 主程序 ····· 4–3
4.1.2 子程序 ····· 4–4
4.1.3 中断程序 ····· 4–6
4.2 程序执行形式 ····· 4–10
4.2.1 初始执行型程序 ····· 4–15
4.2.2 扫描执行型程序 ····· 4–17
4.2.3 低速执行型程序 ····· 4–19
4.2.4 待机型程序 ····· 4–25
4.2.5 固定扫描执行型程序 ····· 4–31
4.3 操作处理 ····· 4–33
4.3.1 初始处理 ····· 4–33
4.3.2 I/O 刷新(I/O 模块刷新处理) ····· 4–33
4.3.3 智能型功能模块的自动刷新 ····· 4–34
4.3.4 终端处理 ····· 4–34
4.4 运行，停止，暂停三种操作处理 ····· 4–35
4.5 瞬时掉电时的操作处理 ····· 4–36
4.6 数据清除处理 ····· 4–37

4.7 输入/输出处理和响应滞后 4–38
4.7.1 刷新模式 4–38
4.7.2 直接模式 4–41
4.8 在顺控程序中能使用的数值 4–43
4.8.1 BIN（二进制编码） 4–45
4.8.2 HEX（十六进制） 4–46
4.8.3 BCD（二进制编码的十进制） 4–47
4.8.4 实数（十进制浮点数据） 4–48
4.9 字符串数据 4–51

5 分配 I/O 地址 5– 1 至 5–19

5.1 扩展基板的级和插槽编号间的关系 5–1
5.2 安装扩展基板和设定段级数 5–2
5.3 基板分配（基本模式） 5–3
5.4 什么是 I/O 地址? 5–7
5.5 I/O 地址分配的概念 5–8
5.5.1 远程站 I/O 地址 5–10
5.6 用 GX Developer 分配 I/O 地址 5–11
5.6.1 用 GX Developer 分配 I/O 的目的 5–11
5.6.2 用 GX Developer 分配 I/O 的概念 5–12
5.7 I/O 地址分配示例 5–15
5.8 检查 I/O 地址 5–19

6 QCPU 文件 6– 1 至 6–23

6.1 关于 QCPU 的存储器 6–3
6.2 程序存储器 6–6
6.3 关于标准 ROM 6–8
6.4 关于标准 RAM 6–9
6.5 存储卡 6–11
6.6 在标准 ROM 或在闪存卡中写入数据 6–12
6.6.1 用 GX Developer 在标准 ROM 或闪存卡中写入数据 6–12
6.6.2 自动写入标准 ROM
（从存储卡将所有的数据自动写入标准 ROM） 6–14
6.7 执行标准 ROM/存储卡程序（引导运行） 6–17
6.8 程序文件配置 6–19
6.9 GX Developer 文件操作和文件处理的注意事项 6–20
6.9.1 文件操作 6–20
6.9.2 文件处理的注意事项 6–21
6.9.3 文件容量 6–22

A - 8

7 功能 7– 1 至 7–77

7.1 功能表 7–1
7.2 恒定扫描 7–2
7.3 锁存功能 7–5
7.4 从停止状态切换到运行状态时设置输出（Y）状态 7–7
7.5 时钟功能 7–9
7.6 远程操作 7–12
7.6.1 远程运行/停止 7–12
7.6.2 远程暂停 7–15
7.6.3 远程复位 7–17
7.6.4 远程锁存清除 7–19
7.6.5 远程操作和 QCPU 运行/停止开关间的关系 7–20
7.7 改变 QCPU-兼容输入模块和中断模块的响应时间
（I/O 响应时间） 7–21
7.7.1 改变输入模块的响应时间 7–21
7.7.2 改变高速输入模块的响应时间 7–22
7.7.3 改变中断模块的响应时间 7–23
7.8 设定智能-功能模块的切换 7–24
7.9 监控功能 7–25
7.9.1 监控条件设置 7–25
7.9.2 对本地软元件的监控测试 7–29
7.9.3 对外部 I/O 强制 ON/OFF 7–31
7.10 在 CPU 运行期间写入程序 7–35
7.10.1 运行状态下用回路模式写入数据 7–35
7.10.2 运行状态下写入批文件 7–37
7.11 执行时间的测量 7–39
7.11.1 程序监控表 7–39
7.11.2 中断程序监控表 7–41
7.11.3 扫描时间的测量 7–42
7.12 采样跟踪功能 7–44
7.13 多用户的调试功能 7–53
7.13.1 多用户监控功能 7–54
7.13.2 多用户运行写入功能 7–55
7.14 警戒定时器（WDT） 7–57
7.15 自诊断功能 7–59
7.15.1 由于出错而中断 7–62
7.15.2 出错时 LED 显示 7–62
7.15.3 删除错误 7–63
7.16 故障历史记录 7–64
7.17 系统保护 7–65
7.17.1 口令注册 7–65
7.17.2 远程口令 7–67
7.18 用 GX Developer 监控 QCPU 系统状态（系统监控） 7–70
7.19 LED 显示 7–74
7.19.1 LED 显示 7–74
7.19.2 优先级的设置 7–76

8 与智能型功能模块/特殊功能模块的通信 8– 1 至 8–9

8.1 QCPU 与 Q 系列智能型功能模块组之间的通信 8–1
8.1.1 使用 GX CONFIGURATOR 进行初始值设定及自动刷新设定 8–2
8.1.2 使用软元件的初始值通信 8–3
8.1.3 使用 FROM/TO 指令的通信 8–4
8.1.4 使用智能型功能模块的软元件的通信 8–4
8.1.5 使用智能型功能模块的专用指令的通信 8–5
8.2 从智能型功能模块向 QCPU 发出的请求 8–6
8.2.1 从智能型功能模块发出的中断请求 8–6
8.3 QCPU 与 AnS-系列特殊功能模块组之间的通信 8–7
8.3.1 使用软元件初始值的通信 8–7
8.3.2 使用 FROM/TO 的指令通信 8–8
8.3.3 使用智能型功能模块的软元件的通信 8–8
8.3.4 对特殊功能模块进行快速访问的影响和防范措施 8–9

9 参数表 9– 1 至 9–5

10 软元件 10– 1 至 10–71

10.1 软元件表 10–1
10.2 内部用户软元件 10–3
10.2.1 输入（X） 10–5
10.2.2 输出（Y） 10–8
10.2.3 内部继电器（M） 10–10
10.2.4 锁存继电器（L） 10–11
10.2.5 信号报警器 10–12
10.2.6 边沿继电器（V） 10–16
10.2.7 通信继电器（B） 10–17
10.2.8 特殊通信继电器（SB） 10–18
10.2.9 步进继电器 10–18
10.2.10 定时器 10–19
10.2.11 计数器 10–24
10.2.12 数据寄存器 10–28
10.2.13 通信寄存器 10–29
10.2.14 特殊通信寄存器 10–30
10.3 内部系统软元件 10–31
10.3.1 功能软元件（FX，FY，FD） 10–31
10.3.2 特殊继电器（SM） 10–33
10.3.3 特殊寄存器（SD） 10–34
10.4 链接直接软元件（J□\□） 10–35
10.5 智能型功能模块的软元件（U□\G□） 10–38
10.6 变址寄存器(Z) 10–39
10.6.1 扫描执行程序与低速执行程序之间的切换 10–40
10.6.2 扫描/低速执行程序与中断/固定扫描执行程序间的切换 10–41
10.7 文件寄存器（R） 10–43
10.7.1 文件寄存的容量 10–44

10.7.2 不同的存储卡，类型决定访问存储卡方法的差别 10–44
10.7.3 注册文件寄存器 10–45
10.7.4 文件寄存器的指定方法 10–49
10.7.5 使用文件寄存器的注意事项 10–50
10.8 嵌套(N) 10–52
10.9 指针 10–53
10.9.1 本地指针 10–53
10.9.2 公共指针 10–54
10.10 中断指针(I) 10–56
10.11 其它软元件 10–58
10.11.1 SFC 块软元件(BL) 10–58
10.11.2 SFC 转换软元件(TR) 10–58
10.11.3 网络编号指定软元件(J) 10–58
10.11.4 I/O 地址指定软元件(U) 10–59
10.11.5 宏指令自变量软元件(VD) 10–60
10.12 常数 10–61
10.12.1 十进制常数(K) 10–61
10.12.2 十六进制常数（H） 10–61
10.12.3 实数（E） 10–62
10.12.4 字符串（〃） 10–62
10.13 软元件的简便使用 10–63
10.13.1 全局软元件和本地软元件 10–63
10.13.2 软元件的初始值 10–69

11 QCPU 的处理时间 11– 1 至 11–4

11.1 读取 QCPU 的扫描时间 11–1
11.2 延长扫描时间的影响因素 11–2
11.3 缩短扫描时间的影响因素 11–4

12 将程序写入 QCPU 中的步骤 12– 1 至 12–8

12.1 写入 1 个程序的步骤 12–1
12.1.1 创建多程序时的考虑事项 12–1
12.1.2 向 QCPU 写入程序的步骤 12–2
12.2 写入多个程序的步骤 12–5
12.2.1 创建多个程序时的考虑事项 12–5
12.2.2 将程序写入 QCPU 的步骤 12–6

13 多个 PLC 系统的简介 13– 1 至 13–6

13.1 特点 13–1
13.2 多个 PLC 程序简介 13–3
13.3 与单个 CPU 系统的区别 13–5

14 多 CPU 系统的系统配置 14– 1 至 14–20

14.1 系统配置 14–1
14.2 在多 CPU 系统配置期间的注意事项 14–4
14.2.1 可用 QCPU 和动作 CPU 的功能版本及其安装位置 14–4
14.2.2 使用 Q 系列相应的 I/O 模块和智能型功能模块时的注意事项 14–8
14.2.3 安装 AnS 系列对应的 I/O 模块和特殊功能模块时的限制 14–9
14.2.4 有安装限制的模块 14–11
14.2.5 使用 CX–Developer 和 GX–配置器 14–12
14.2.6 使多 CPU 系统生效的参数 14–13
14.2.7 多 CPU 系统的复位 14–17
14.2.8 当发生 QCPU 停止错误时的处理 14–18
14.2.9 减少多 CPU 系统处理所需的时间 14–20

15 多 CPU 系统 I/O 地址的分配 15– 1 至 15–3

15.1 分配 I/O 地址的概念 15–1
15.1.1 I/O 模块和智能型功能模块的 I/O 地址 15–1
15.1.2 QCPU 和动作 CPU 的 I/O 地址 15–2
15.2 用 GX–Developer 进行 PC 参数 I/O 分配的目的 15–3

16 多 CPU 系统的 QCPU 与动作 CPU 之间的通信 16– 1 至 16–15

16.1 公共 CPU 存储器的自动刷新 16–2
16.2 用多 CPU 命令与智能型功能模块软元件进行通信 16–9
16.3 QCPU 和动作 CPU 之间的相互通信 16–11
16.3.1 从 QCPU 到动作 CPU 的控制指令 16–11
16.3.2 读和写软元件数据 16–12
16.4 公共 CPU 存储器 16–13

17 多 CPU 系统的 I/O 模块与智能型功能模块之间的通信 17– 1 至 17–5

17.1 控制 PLC 通信的范围 17–1
17.2 非控制 PLC 通信的范围 17–1

18 多 CPU 系统 QCPU 的处理时间 18– 1 至 18–3

18.1 QCPU 扫描时间的概念 18–1
18.2 延长扫描时间的因素 18–2

19 多 CPU 系统的启动 19– 1 至 19–9

19.1 多 CPU 系统启动的流程图 19–1
19.2 多 CPU 系统参数的设定（多 CPU 设置，控制 PLC 设置） 19–3
19.2.1 系统配置 19–3
19.2.2 创建新系统 19–4
19.2.3 使用现有的预设多 CPU 设置和 I/O 分配 19–7

附件 附件– 1 至附件–52

附件 1 特殊继电器表 ..... 附件– 1
附件 2 特殊寄存器表 ..... 附件– 22
附件 3 中断指针编号和中断因素表 ..... 附件– 52

索引 索引– 1 至索引–3

A - 13

(相关手册).....QCPU（Q 模式）用户手册（硬件设计/维护和检修）

目录

1 综述

1.1 综述
1.2 特性

2 系统配置

2.1 系统配置
2.2 系统配置的安全要求
2.3 确定系列编号

3 通用规格

4 CPU 模块的硬件规格

4.1 性能规格
4.2 部件名称和设定
4.3 写入程序后的切换操作
4.4 锁存清除操作
4.5 在标准 RAM 中自动写入

5 电源模块

5.1 规格
5.1.1 电源模块规格
5.1.2 选择电源模块
5.2 部件名称和设置

6 基板和扩展电缆

6.1 基板规格表
6.2 扩展电缆规格表
6.3 基板部件名称
6.4 设置扩展基板
6.5 I/O 配置

A - 14

7 存储卡和电池

7.1 存储卡规格
7.2 电池规格（用于 CPU 模块和 SRAM 卡）
7.3 操作存储卡
7.4 存储卡部件名称
7.5 存储卡装入/退出程序
7.6 电池安装（用于 CPU 模块和存储卡）

8 EMC 和低电压指令

8.1 与 EMC 指令一致的要求
8.1.1 可适用于 EMC 指令的标准
8.1.2 EMC 指令的安装指令
8.1.3 电缆
8.1.4 电源模块
8.1.5 使用 Q1AS6□B 型基板时
8.1.6 其它
8.2 与低压指令一致的要求
8.2.1 用于 MELSEC–Q 系列的标准
8.2.2 MELSEC–Q 系列的 PLC 选择
8.2.3 电源
8.2.4 控制盒
8.2.5 接地
8.2.6 外部接线

9 装载和安装

9.1 一般安全要求
9.2 PLC 计算产生的热量
9.3 模块安装
9.3.1 安装的注意事项
9.3.2 安装基板的说明
9.3.3 模块的安装和拆卸
9.4 设定扩展基板的扩展号
9.5 扩展电缆的连接和断开
9.6 接线
9.6.1 接线的注意事项
9.6.2 与电源模块连接

10 维护和检查

10.1 每天检查
10.2 定期检查
10.3 更换电池
10.3.1 电池的使用寿命
10.3.2 更换电池的步骤

11 故障排除

11.1 故障排除的基本概念
11.2 故障排除
11.2.1 故障排除流程图
11.2.2 “MODE”灯不亮时的流程图
11.2.3 “MODE”灯闪烁时的流程图
11.2.4 “POWER”灯熄灭时的流程图
11.2.5 “RUN”灯熄灭时的流程图
11.2.6 “RUN”灯闪烁时
11.2.7 “ERROR”灯发光/闪烁时的流程图
11.2.8 “USER”灯发光时
11.2.9 “BAT.ARM”灯发光时
11.2.10 “BOOT”灯闪烁时的流程图
11.2.11 输出模块的灯不亮时的流程图
11.2.12 输出模块的输出负载灯不亮时的流程图
11.2.13 不能读取程序时的流程图
11.2.14 不能写入程序时的流程图
11.2.15 不能通过存储卡执行引导程序时的流程图
11.2.16 “UNIT VERIFY ERR”发生时的流程图
11.2.17 “CONTROL BUS ERR”发生时的流程图
11.3 出错代码表
11.3.1 读取出错代码的步骤
11.3.2 出错代码表
11.4 出错取消
11.5 I/O 模块的故障排除
11.5.1 输入电路的故障排除
11.6 特殊继电器表
11.7 特殊寄存器表

附件

附件 1 在一般数据处理期间，出错代码的复原
附件 1.1 出错代码的概括性解释
附件 1.2 错误及出错代码的描述
附件 2 外部尺寸图
附件 2.1 CPU 模块
附件 2.2 电源模块
附件 2.3 主基板
附件 2.4 扩展基板

索引

关于手册

下列手册都与本产品有关。
必要时可从下表中选出订购。

相关手册

手册名称	手册编号 (型号代码)
QCPU（Q模式）用户手册（硬件设计，维护和检修） 本手册提供CPU模块，电源模块，基板模块，扩展电缆，存储卡的规格及其它内容。 (选用)	SH-080037 (13JL97)
QCPU（Q模式）/QnACPU编程手册（通用指令） 阐述如何使用顺控指令，基本指令，高级指令及微机程序。 (选用)	SH-080039 (13JF58)
QCPU（Q模式）/QnACPU编程手册PID控制指令 本手册阐述关于学习PID控制的特殊指令。 (选用)	SH-080040 (13JF59)
QCPU（Q模式）/QnCPU编程手册（SFC） 本手册说明系统配置，性能规格，功能，编程，调试，出错代码及MELSAP3的其它内容。 (选用)	SH-080041 (13JF60)

A - 17

如何使用本手册

本手册是为使用 MELSEC–Q 系列可编程逻辑控制器的用户准备的。以使你可以了解存储图，功能，程序和 CPU 模块的软元件。

本手册可简略地分为下列三部分。

(1) 第 1 章和第 2 章	阐述 CPU 模块的概况以及系统配置，CPU 模块的特性和 CPU 系统配置的基本概念。
(2) 第 3 章至第 6 章	阐述 CPU 模块的性能规格，可执行程序，I/O 地址和存储器。
(3) 第 7 章	阐述 CPU 模块的功能。
(4) 第 8 章	阐述使用智能功能模块的通信功能。
(5) 第 9 章和第 10 章	阐述在 CPU 模块中使用的参数和装置。
(6) 第 11 章	阐述 CPU 模块的处理时间。
(7) 第 12 章	阐述写入参数的过程以及将 GX Developer 上创建的程序写入 CPU 模块。
(8) 第 13 章至第 19 章	阐述多个 PLC 系统的概况，系统配置，I/O 地址，CPU 模块之间的通信，以及 I/O 模块和智能型功能模块之间的通信。

备注

本手册不对电源模块，基板，扩展电缆，存储卡和 QCPU 电池的功能进行介绍。
这些功能请参见下列手册。
• QCPU（Q 模式）用户手册（硬件设计，维护和检查）。

A - 18

关于通用术语和缩写语

下表列出本手册中使用的关于 Q02CPU，Q02HCPU，Q06HCPU，Q12HCPU，和 Q25HCPU 的缩写语和通用名称。

通用名称	说明
QCPU	Q02CPU，Q02HCPU，Q06HCPU，Q12HCPU 和 Q25HCPU 模块的通用名称。
QnCPU	Q02CPU 的通用名称。
QnHCPU	Q02HCPU，Q06HCPU，Q12HCPU 和 Q25HCPU 的通用名称。
Q 系列	三菱公司 MELSEC–Q 系列通用型可编程逻辑控制器的缩写语。
AnS 系列	三菱公司小型 MELSEC–A 系列通用型可编程逻辑控制器的缩写语。
GX Developer	SWnD5C–GPPW–E，SWnD5C–GPPW–A–E，SWnD5C–GPPW–V–E，SWnD5C–GPPW–VA–E 的通用产品名称。 用于第 4 版及之后的 QCPU
主基板	可装 Q 系列电源模块，I/O 模块，智能型功能模块的 Q33B，Q35B，Q38B，Q312B 型主基板的通用名称。
Q6□B	可装 Q 系列电源模块，I/O 模块，智能型功能模块的 Q63B，Q65B，Q68B，Q612B 型扩展基板的通用名称。
QA1S6□B	可装 AnS 系列电源模块，I/O 模块，专用功能模块的 QA1S65B 和 QA1S68B 型扩展基板的通用名称。
扩展基板	Q6□B 和 QA1S6□B 的通用名称。
基板	基板和扩展基板的通用名称。
SRAM 卡	Q2MEM–1MBS 型 SRAM 卡的缩写语。
闪存卡	Q2MEM–2MBS 和 Q2MEM–4MBS 型闪存卡的通用名称。
ATA 卡	Q2MEM–8MBS，Q2MEM–16MBS 和 Q2MEM–32MBS 型 ATA 卡的通用名称。
存储卡	SRAM 卡，闪存卡和 ATA 卡的通用名称。
电源模块	Q61P–A1，Q62P–A2，A1S61PN，A1S62P 和 A1S63P 型电源模块的通用名称。
电池	用于 Q6BAT 型 CPU 模块和 Q2MEM–BAT 型 SRAM 卡的电池的通用名称。
扩展电缆	QC06B，QC12B，QC30B，QC50B，QC100B 型扩展电缆的通用名称。

A - 19

1 综述

1

本手册阐述了 QCPU 的存储图，功能，程序和软元件。

有关 QCPU 电源模块，基板，扩展安装电缆，存储卡，电池规格和其它资料详见下列手册。
QCPU（Q 模式）用户手册（硬件建立，维护和检查）。

(1) 前五位数字为“02092”的系列号的增加功能

系列编号前五位为“02092（02092**********–A）”的 QCPU 已经增加了下列功能。（每一种增加的功能详见相关说明。）

- 自动写入标准 ROM（见第 6.6.2 节）
- 对外部 I/O 支持增强 ON/OFF（见第 7.9.3 节）
- 远程口令设定（见 7.19 节）
- Q12HCPU 和 Q25HCPU 的 RAM 容量增大(见第 3 章,第 6.4 节)
- 支持 MELSECNET/H 远程 I/O 网络(见第 1.1 节)
- 支持中断模块(Q160)(见 7.7.3 节)

(2) 对 B 版增加的功能

B 版的 QCPU 功能中已经增加了下列功能：
(这一增加功能详见相关说明)

- 支持多个 PLC 系统(见第 13 章至 19 章)

要点
（1）在使用系列编号前五位为“02091”或更早期的 QCPU 时，应跳过说明增加到系统编号前五位为“02092”的 QCPU 上的功能在页面。 （2）当不使用多个 PLC 系统时，跳过第 13 至 19 章。 （3）参见第 2.3 节，确认 QCPU 系列编号和功能版本。

备注

QCPU 是 Q02CPU，Q02HCPU，Q06HCPU，Q12HCPU 和 Q25HCPU 通用型可编程控制器的通用名称。当对每种模块进行单独说明时，也必须提到 Q02CPU，Q02HCPU，Q06HCPU，Q12HCPU 和 Q25HCPU。

1.1 特点

QCPU 具有下列特性：

(1) 可控制多个 I/O 点

所有的 QCPU 支持 4096 个点（X/Y0 到 FFF）。这些实有的 I/O 点能够对装在基板上 I/O 模块的进行存取。
这些 QCPU 最多可支持作为 I/O 软元件数的 8192 个点（X/Y0 到 1FFF）。这些点可用于远程 I/O 站，如 MELSECNET/H 远程 I/O 网，CC–Link 数据通讯和 MELSECNET/MINI–S3 数据通讯。

(2) 按程序容量排列

按使用的程序容量，可选择最佳的 CPU 模块。

Q02CPU，Q02HCPU	：28 K 步
Q06HCPU	：60 K 步
Q12HCPU	：124 K 步
Q25HCPU	：252 K 步

(3) 实现高速处理

某些类型的可编程控制器已经实现了高速处理。（例：当使用 LD 指令时）

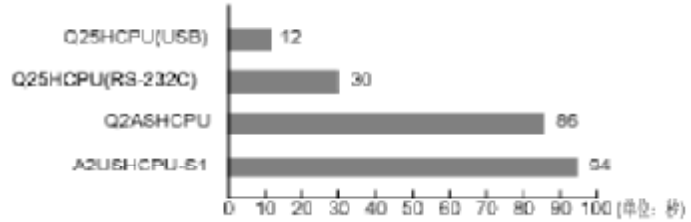
Q02CPU	：0.079μs
Q02HCPU，Q06HCPU，Q12HCPU，Q25HCPU	：0.034μs

此外，通过采用新开发的基板的高速系列总线，可实现对智能型功能模块的存取并提高网络链接刷新的速度。
对智能型功能模块的存取：20us/字（大约 7 倍）*1
MELSECNET/H 链接刷新处理：4.6ms/8k 字（大约 4.3 倍）*1
*1：此处是将 Q02HCPU 和 Q2ASHCPU–S1 相比较。

(4) 通过用 GX Developer 的高速通信来提高调试效率

在 QCPU 中，通过速度最大为 115.2kbps 的高速通信，减少程序的写/读和监控所需时间。采用 RS–232C，在调试时，其通信时间效率得以提高。在 Q02HCPU，Q06HCPU，Q12HCPU，Q25HCPU 中，通过 USB 允许进行速度为 12Mbps 的高速通信。

26k 步程序传输时间



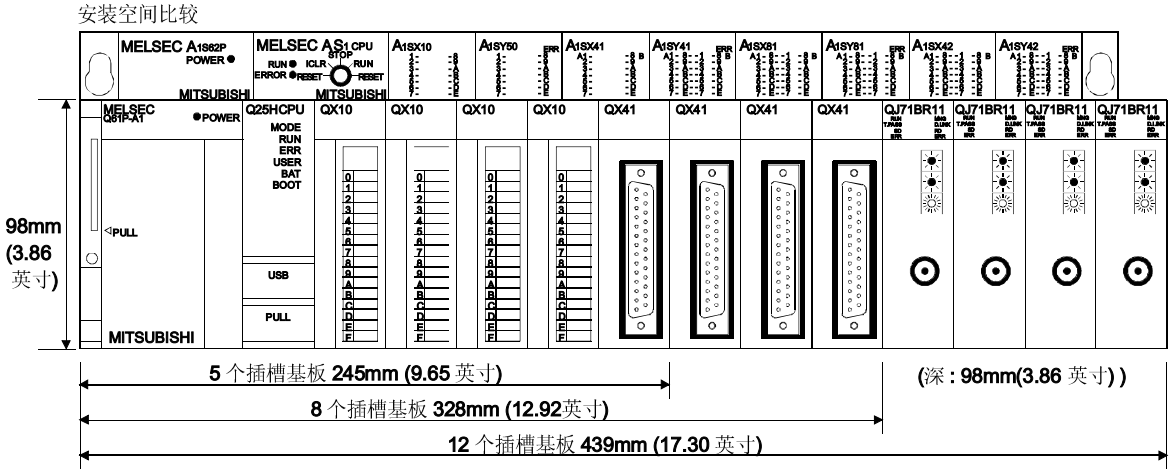
(5) 提供 AnS 系统 I/O 模块或特殊功能模块。

对 Q 系列系统，如果未提供适当的模块，那么通过使用 QA1S65B/QA1S68B 扩展基板，AnS 系列 I/O 模块或特殊功能模块也能用于 QCPU。

(6) 通过减小尺寸可节省空间

Q 系列的安装空间比安装 AnS 系列的空间减少约 60%。

安装空间比较



(7) 扩展基板最多可连接 7 级。

最多可连接 7 级扩展基板到 QCPU（包括主基板共有八段）上，而最多可连接 64 个基板到 QCPU 上。

且扩展电缆的总扩展长度最长为 13.2m（43.28 英尺），因而扩展基板的安装受到的限制不大。

(8) 用存储卡扩展存储

QCPU 随存储卡安装连接器一起提供，因而可连接最大为 32Mb 的存储卡。（当使用 ATA 卡时，可用量 32M 字节）

当装上大容量的存储卡时，就可处理大容量的文件，并建立所有数据软元件的注释。过去的程序可通过修正过的历史记录形式存储在存储器中。

甚至未装上存储卡，程序也可通过组装在 QCPU 中的标准 ROM 而保存在 ROM 中。通过标准 RAM 可操作一个 32k 点/128k 点的文件寄存器。

(9) 能自动写入标准 ROM

不用 GX Developer 也能将存储卡参数和程序写入 QCPU 的标准 ROM 上。
这表示为了改写参数和程序，不必携带 GX Developer（个人计算机），当用标准 ROM 执行 ROM 操作时，装入 QCPU 的存储卡参数和程序可写入标准 ROM 中。

备注

＊ 系列编号前五位以“02092”开始的 Q12HCPU 和 Q25HCPU 可操作 128k 点的文件寄存器。
系列编号前五位以“02091”开始的 Q02CPU 和 Q02HCPU，Q06HCPU 和 Q12HCPU 和 Q25HCPU 或更早一些的系列可操作 32k 点的文件寄存器。

(10) 可执行外部 I/O 强制 ON/OFF

用 GX Developer 操作可执行外部 I/O 强制 ON/OFF，即使当 QCPU 在 RUN 状态而不用管程序的执行状态。即使不将 QCPU 设置在 STOP 状态，也能进行接线和操作试验。

(11) 能设置远程口令

当从外部源访问以太网模块和串行通信模块时，可以选择是否用远程口令访问 QCPU。

(12) 可安装 MELSECNET/H 远程 I/O 网络

能安装 MELSECNET/H 远程主站以创建 MELSECNET/H 远程 I/O 系统。

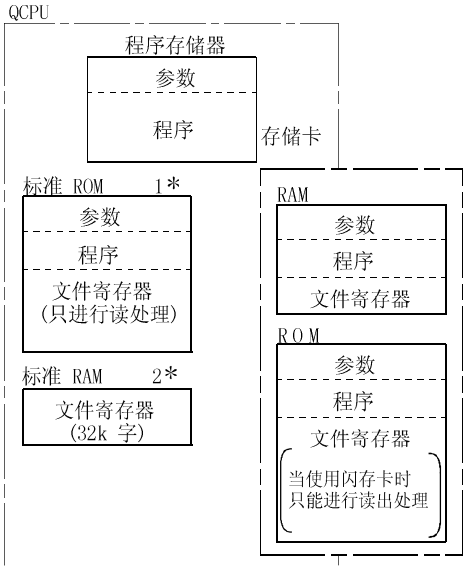
备注

- （9）至（12）的主要特性是系列编号前五位为“02092”的 QCPUs 的增加功能。
- 当使用以太网模块，串行通信模块，和 GX Developer（在 SW6D5C–GPPW–E 之后的产品）的 B 版功能时，可执行远程口令。
- 除了远程口令外，还提供下列 QCPU 的保护功能：
  带有 QCPU 系统设置的完整 CPU 保护
  带有存储卡上写保护开关的存储卡保护
  每个单独的带口令的文件保护
- 在使用 MELSECNET/H 网络模块功能 B 版和 GX Developer（SW6D5C–GPPW–E 之后的产品）时，可操作 MELSECNET/H 远程 I/O 网络。

1.2 程序

(1) 能用存储卡进行程序管理

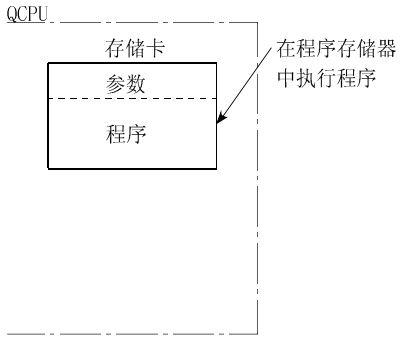
(a) 在 **GX Developer** 上创建的程序可以保存在 **QCPU** 的程序存储器、标准 **ROM** 或存储卡上。



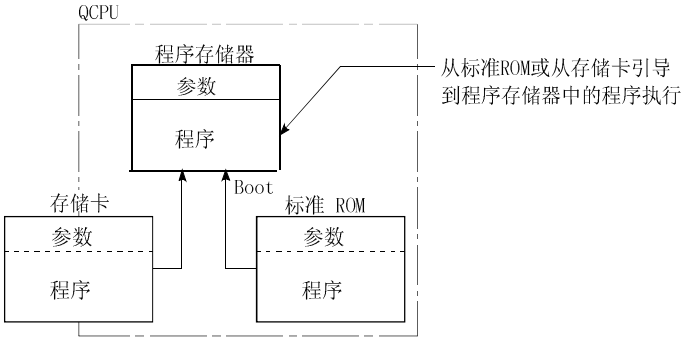
***1.** 当将参数和程序写入 **ROM** 时，使用标准 **ROM**。

***2.** 当对文件寄存器的访问需要加速时，使用标准 **RAM**。

(b) 保存在程序存储器中的 **QCPU** 处理程序。

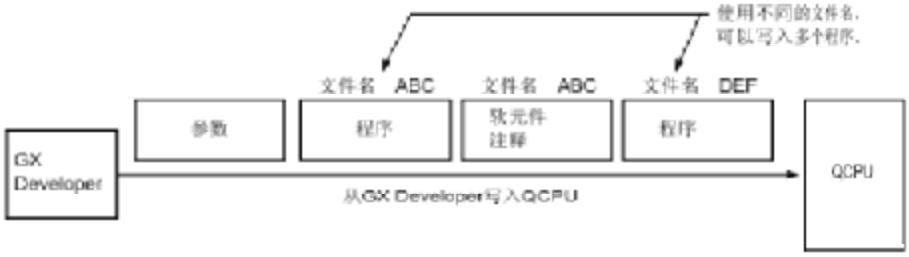


保存在标准 ROM/存储卡里的程序在经引导（读到）QCPU 程序存储器里后，即能执行。（由参数设置来指定被引导至 QCPU 中的程序，而参数驱动器由 QCPU 中的 DIP 开关设置来指定。）



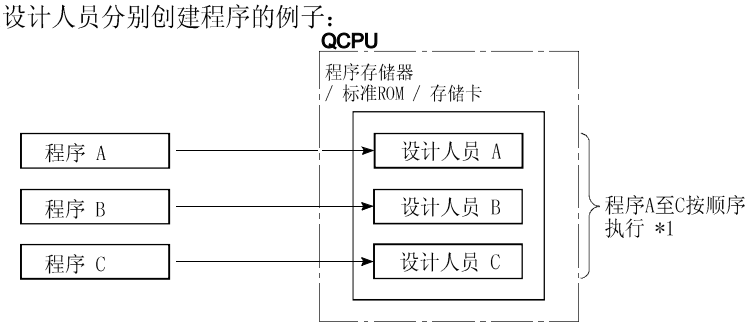
(2) 程序程序

QCPU 程序以文件格式保存在程序存储器、标准 ROM 或存储卡中。
使用不同的文件名，多个文件也能保存在程序存储器、标准 ROM 或存储卡中。



这一格式允许程序的创建操作可由几个程序设计人员进行，并可按创建中的处理过程或功能，执行程序管理和维护。
此外，当规格发生改变时，只需对有关的程序进行修改和调试。

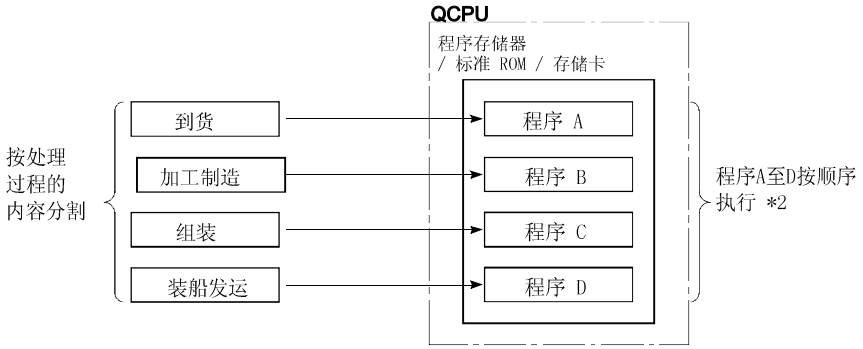
(a) 由 n 位设计人员分别创建程序的例子：



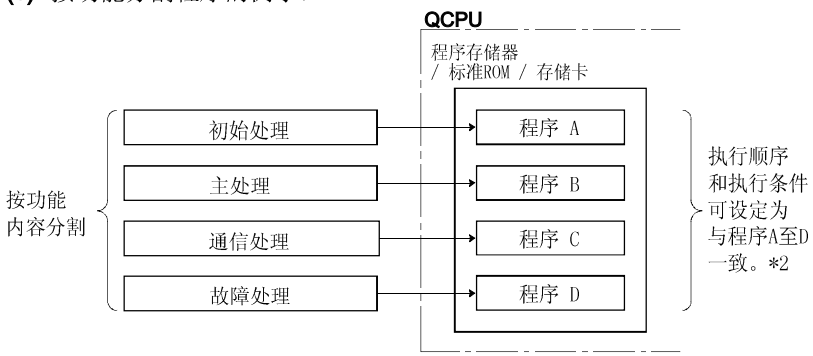
备注

***1.** 关于执行顺序详见 **4.2** 节。

(b) 按处理过程分割程序的例子：*1



(c) 按功能分割程序的例子：



备注

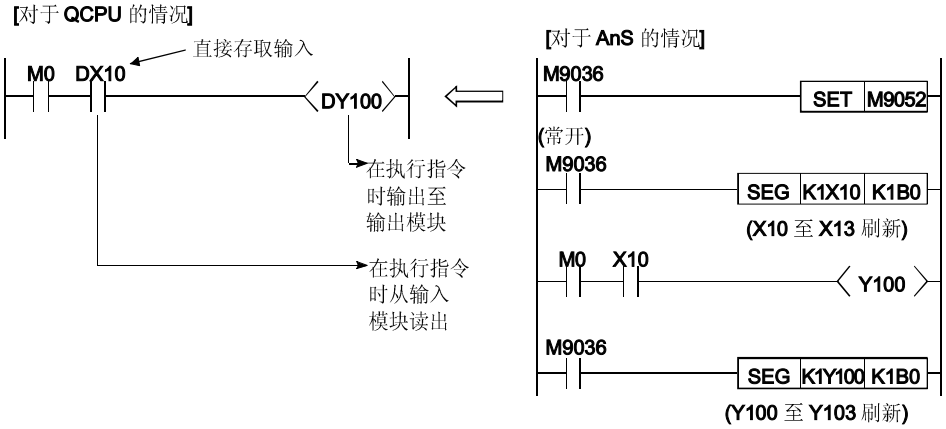
*1. 按处理过程分割的文件可按功能进一步分割。
*2. 执行顺序和执行条件详见 4.2 节。

1.3 简易的可编程软元件及指令

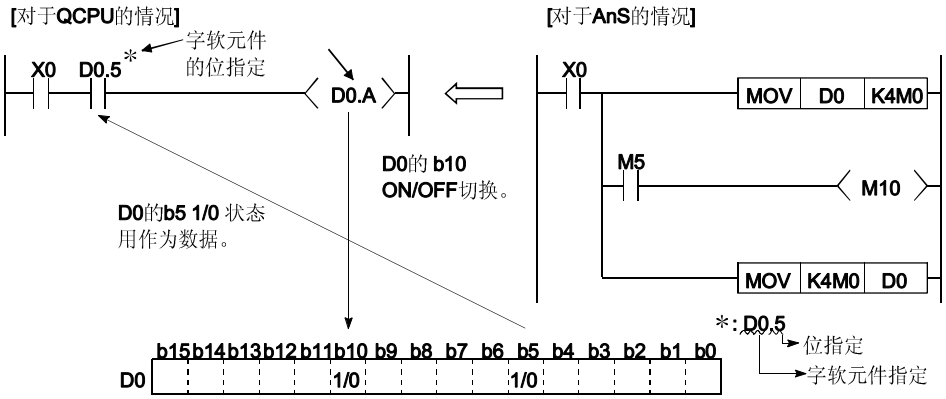
QCPU 具有软元件和指令有助于简化程序的创建的特点。以下阐述几种情况。

(1) 柔性软元件指定

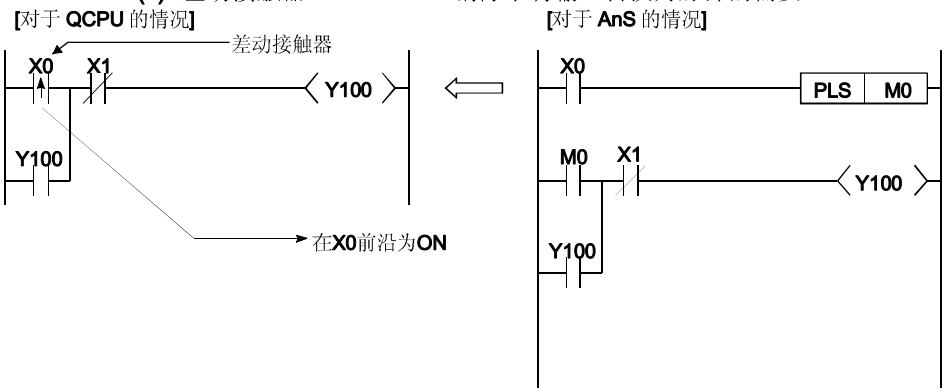
(a) 指定字软元件位用作为接触器或线圈。



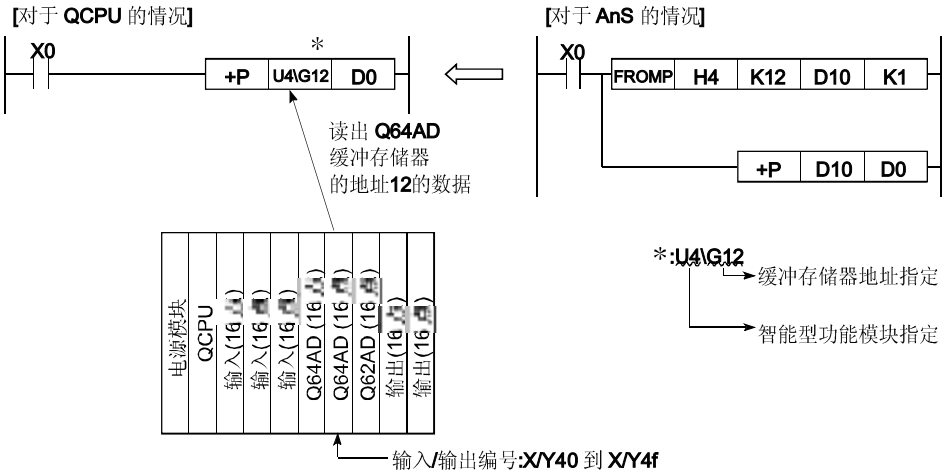
(b) 使用直接存取输入（DX□）和直接访问输出（DY□），就能简易地用程序以 1–点为单元直接处理。



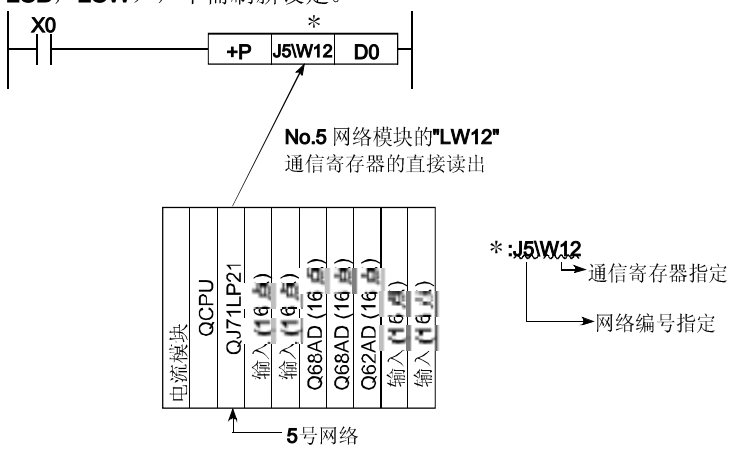
(c) 差动接触器（⊣↑⊢/⊣↓⊢）消除了将输入转换为脉冲的需要。



(d) 在编程时智能型功能模块（如 Q4AD，Q62DA）的缓冲存储器可作为软元件，以同样的方式使用。



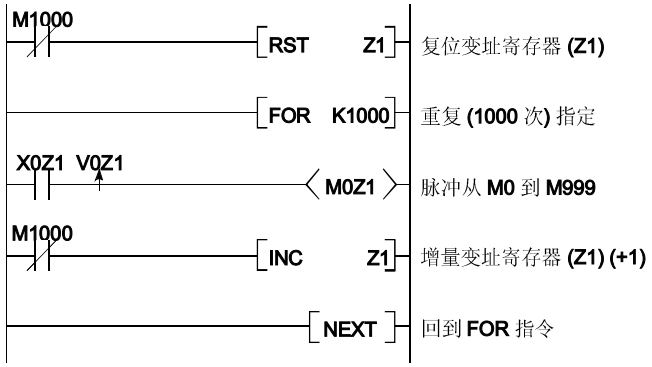
(e) 直接访问 MELSECNET/10H 网络模块的通信软元件（LX，LY，LB，LW，LSB，LSW），不需刷新设定。



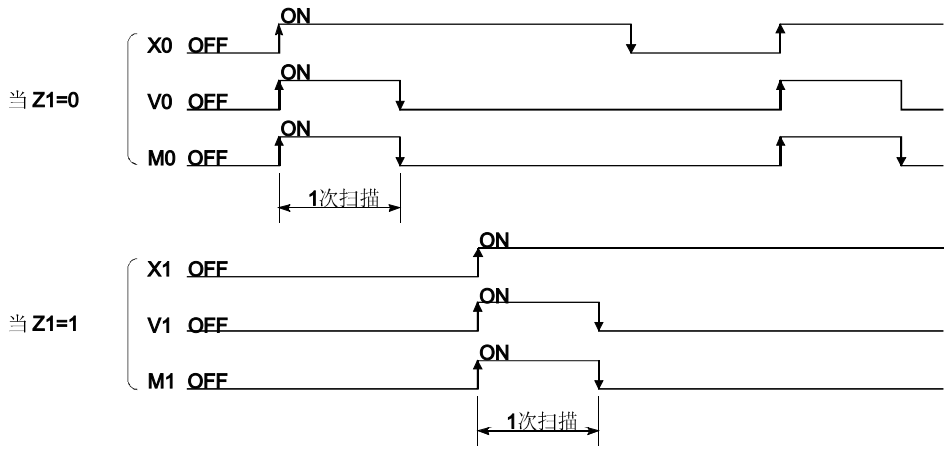
(2) 边沿继电器简化脉冲转换处理

(a) 当接触器变址修饰完成时，在输入条件前沿使用继电器（V），以 ON 表示，可简化脉冲处理。

[回路示例]

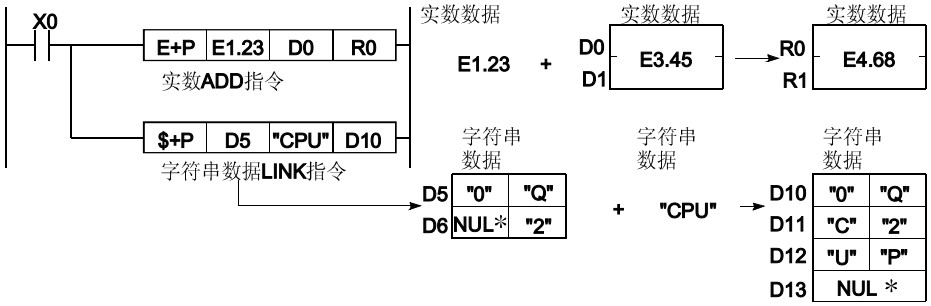


[时序图]



(3) 较简便的数据处理

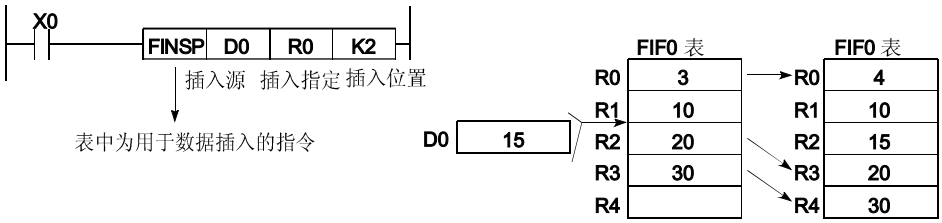
(a) 在编程时可使用实数（十进制浮点数据）和字符串常数。



备注

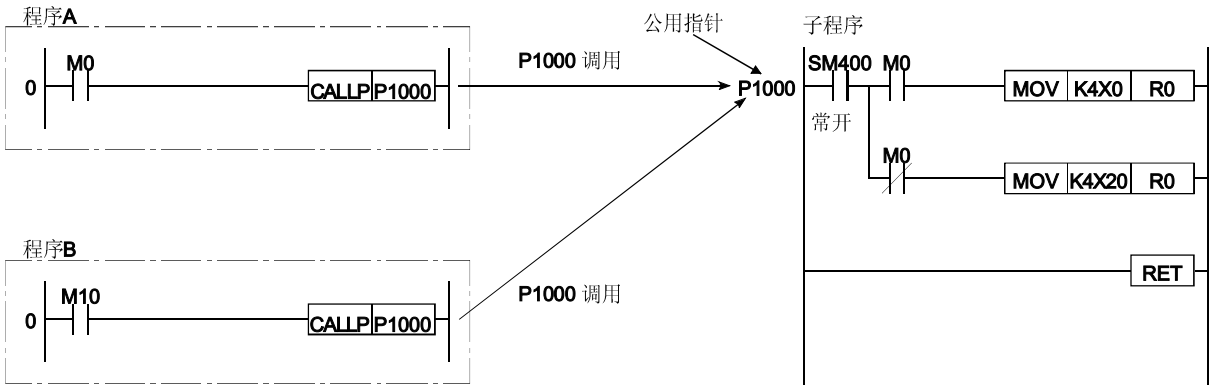
*： NUL 表示“00H（字符串 END）”。

(b) 如表处理指令等的数据处理指令能高速处理大量数据。

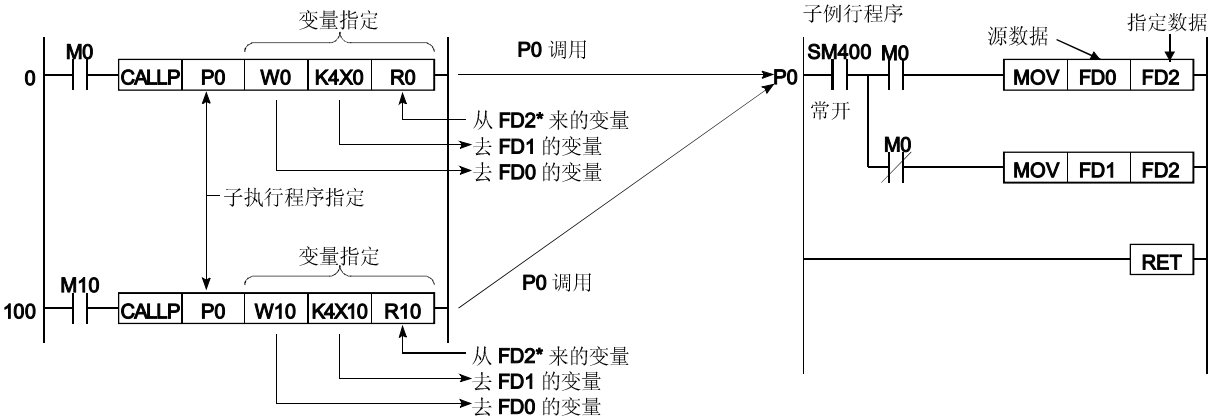


(4)子程序的简易共享使用

(a) 公共指针用来从所有的执行中的顺控程序中调用相同的子程序。



(b) 使用带变量的子程序调用指令时，可简化多次调用子程序的创建。



备注

＊ 有关变量输入/输出条件详见第 **10.3.1** 节。

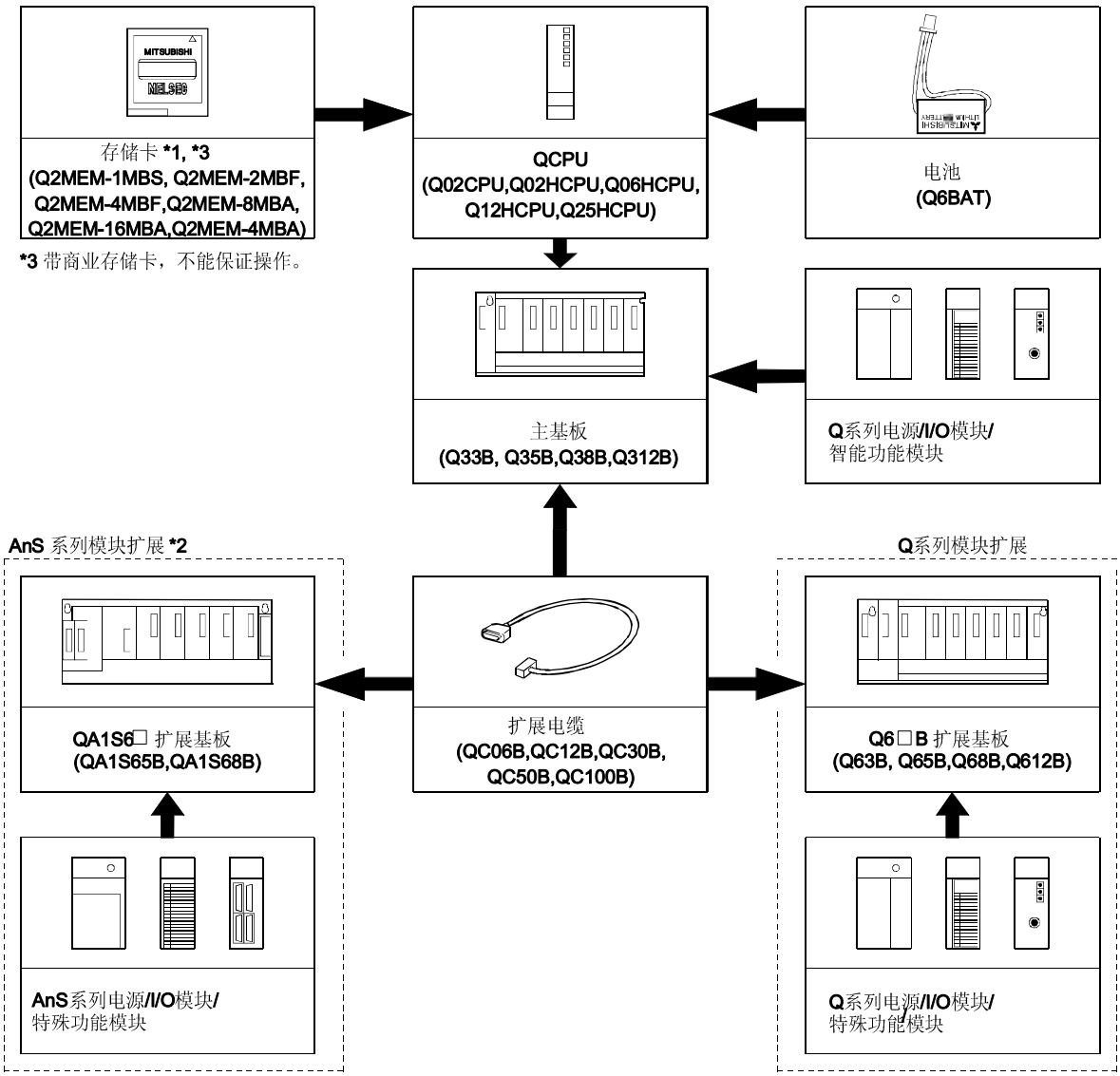
2. 对单 CPU 系统的系统配置

本节阐述 QCPU 系统的配置，系统安全事项和软元件。

2.1 系统配置

阐述与 QCPU 系统有关的系统设备，外围设备配置和系统配置的概况。

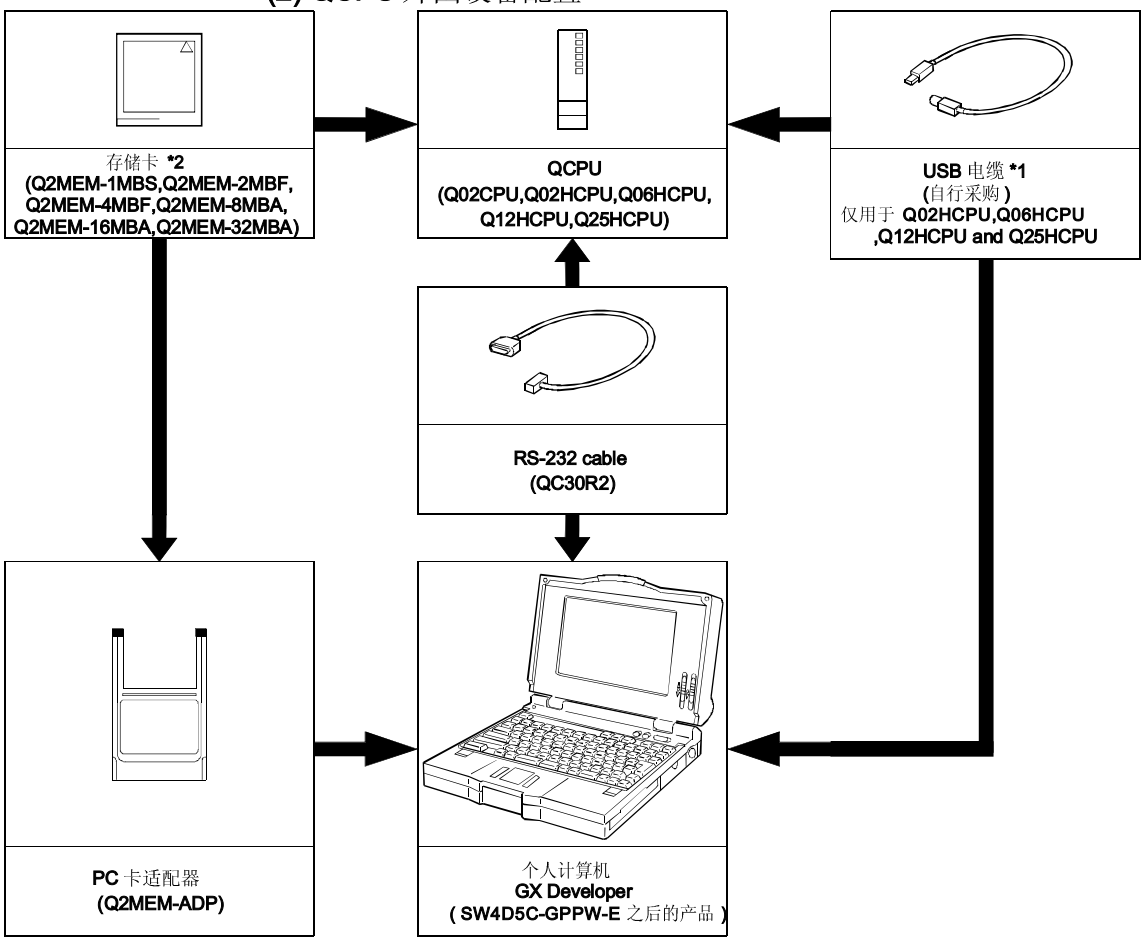
(1) QCPU 系统设备



要点

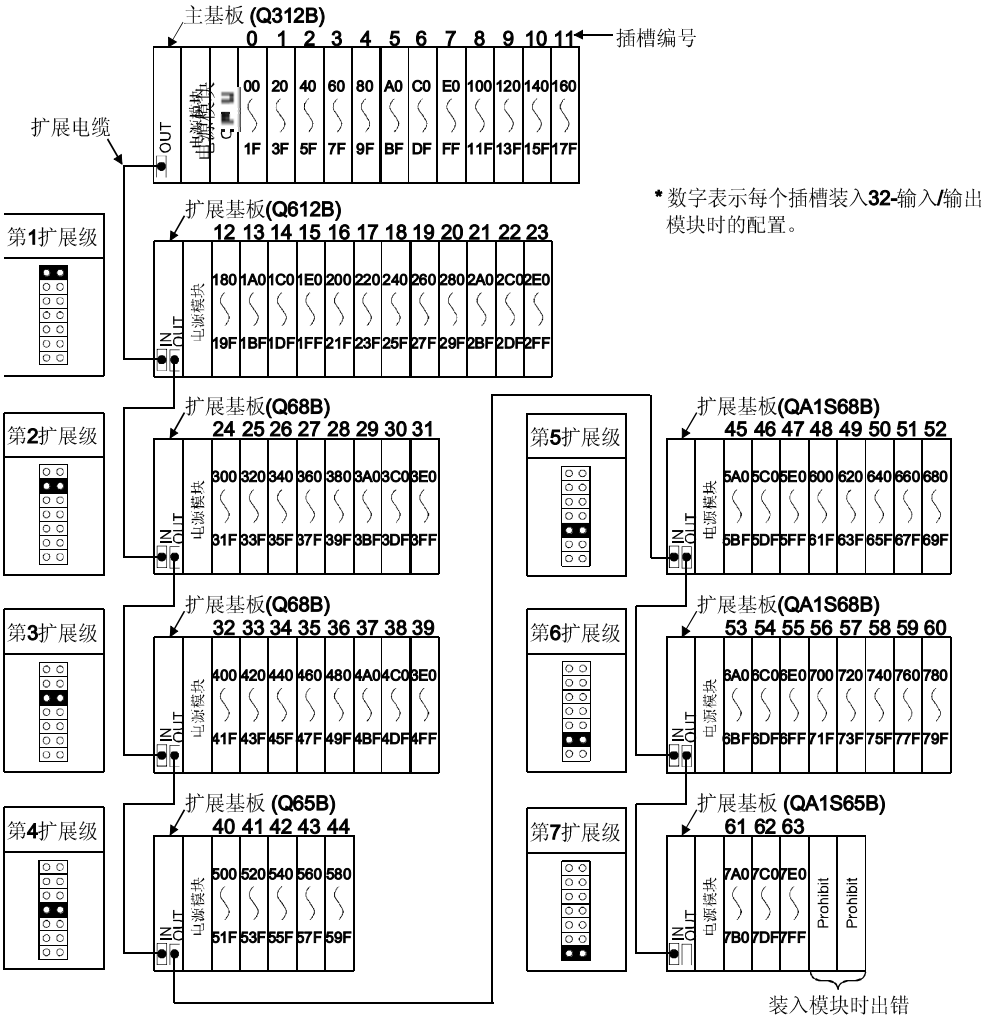
*1: 需要时可装入一块存储卡。
按应用和容量要求，存储卡可从 SRAM，闪速卡和 ATA 卡中选择。
*2: QA1S65B 和 QA1S68B 扩展基板可用于 AnS 系列电源模块，I/O 模块和 特殊功能模块。

(2) QCPU 外围设备配置



*1: 有关写入存储卡的步骤和 USB 电缆见 GX Developer 操作手册。

(3) 系统配置概况

系统配置	
最大扩展级数	7 个扩展级
最多可装入的输入/输出模块	64 个模块
最大的输入/输出点	4096
主基板	Q35B, Q38B, Q312B
扩展基板	Q65B, Q68B, Q612B, QA1S65B, QA1S68B
扩展电缆	QC06B, QC12B, QC30B, QC100B
注意	(1) 最多可装 7 个扩展基板。 (2) 使用扩展电缆长度不得超过 13.2m（43.28 英尺） (3) 使用扩展电缆时不得接入或靠近主回路（高压和大电流） (4) 当设定扩展级数时，要保证以升序进行以避免设定同样的编号。 (5) 如果在同一个扩展基板上装 Q6□B 和 QA1S6□B，则必须保证先连接 Q6□B，然后再连接 QA1S6□B。 (6) 在一个扩展基板的 OUT 连接器与另一个扩展基板的 IN 连接器之间连接扩展电缆。 (7) 如果安装 64 个以上的扩展基板，就会出错。

2.2 系统安全注意事项

本节阐述与 QCPU 兼容的硬件和软元件。

(1) 硬件

(a) 允许的模块数和功能受特殊模块的限制。

名称	型号	模块数量/限定
Q 系列 MELSECNET/H 网络模块	• QJ71LP21 • QJ71BR11 • QJ71LP21–25	内部 PLC 到 PLC 网络和远程 I/O 网络总数可达 4 块
Q 系列以太网接口模块	• QJ71E71 • QJ71E71–B2	4 块
Q 系列 CC–Link 系统主/本地模块	• QJ61BT11	4 块
MELSECNET/MINI–S3 数据通讯模块	• A1SJ71PT32–S3 • A1SJ71T32–S3	没有限制（但是不能设定自动刷新功能）
AnS 系列特殊功能模块如右	• A1SD51S，A1SD21–S1 • A1SJ71J92–S3 （在 GET/PUT 服务中用）	总数 6 块
中断模块	• A1SI61，Q160	只能一块

以太网（Ethernet）为美国施乐公司的注册商标

(b) 下列 AnS 系列特殊功能模块受可访问软元件范围的限制：

• A1SD51S 型智能型通信模块

装置	可访问软元件的范围
输入（X），输出（Y）	X/Y0—7FF
内部继电器（M）/锁存继电器（L）	M0—8191
通信继电器（B）	B0—FFF
定时器（T）	T0—2047
计数器（C）	C0—1023
数据寄存器（D）	D0—6143
通信寄存器（W）	W0—FFF
信号报警器（F）	F0—2047

(c) 只有 GOT900 系列可以用图形操作终端

它要求安装 Q 模式兼容的操作系统（OS）和通讯驱动程序，

不能用 GOT800 系列，A77GOT 和 A64GOT。

(d) QCPU 不能使用下列模块：

名称	型号
MELSECNET/10 网络模块	A1SJ71LP21，A1SJ71BR11，A1SJ71QLP21，A1SJ71QLP21S，A1SJ71QLP21GE，A1SJ71QBR11
MELSECNET（II），/B 数据通讯模块	A1SJ71AP21，A1SJ71AR21，A1SJ71AT21B
以太网接口模块	A1SJ71QE71–B2（–B5），A1SJ71E71–B2–S3（–B5–S3）
串行通信模块，计算机通信模块	A1SJ71QC24（N），A1SJ71UC24–R2（–R4/–PRF）
CCLink 主/本地模块	A1SJ61QBT11，A1SJ61BT11
MOdem 接口模块	A1SJ71CMO–S3
ME–NET 接口模块	A1SJ71ME81

(e) 虽然模块特殊指令包括在 QnA/A 系列的顺控指令中，但 QCPU 不处理下列模块的模块–特殊指令。因此用户必须重写 FROM/TO 指令。

名称	型号
高速计数模块	A1SD61，A1SD62，A1SD62D（–S1），A1SD62E
MELSECNET/MINI–S3	A1SJ71PT32，A1SJ71T32–S3
定位模块	A1SD75P1–S3（P2–S3/P3–S3），A1SD75M1（M2/M3）
ID 模块	A1SJ71ID1–R4，A1SJ71ID2–R4

(2) 软件包

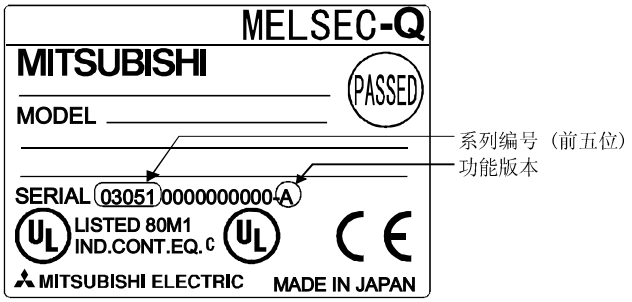
创建 QCPU 程序的软件是 GX Developer（SW4D5C–GPPW–E 或更高）。但在使用编号前五位为“02092”的 QCPU 的增加功能时，必须使用 GX Developer（SW6D5C–GPPW–E 或更高）。

2.3 确认系列编号和功能版

QCPU 系列编号和功能版可用额定铭牌和 GX Developer 系统监控器进行确认。

(1) 用额定铭牌确认

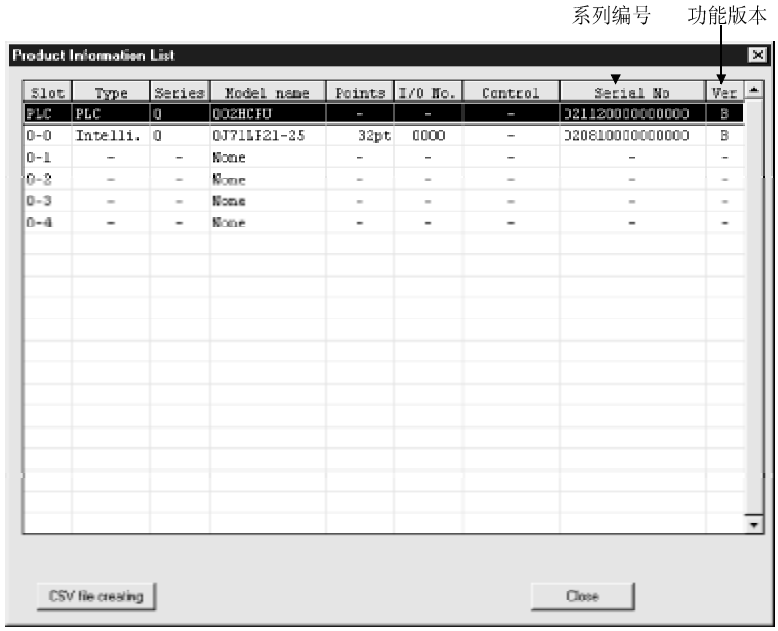
QCPU 系列编号和功能版本可通过额定铭牌上的“SERIAL”一栏确认。



(2) 由系统监控器确认（产品资料表）

QCPU 系列编号和功能版本可通过 GX Developer（SW6D5C–GPPW–E 或更高）系统监控器中的产品资料表进行确认。

QCPU 的系列编号和功能版本也可通过智能型功能模块进行确认。



3. 性能规格

QCPU 模块的性能规格如下：

表 3.1 性能规格

项目			型号					备注
			Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU	
控制方式			重复操作保存的程序					
I/O 控制方式			刷新方式					通过软元件名称进行直接 I/O
程序语言			程序控制的专用语言					
			继电器符号语言，逻辑符号语言 MELSAP–3（SFC）					
处理速度（顺控指令）（μS/步）		LD	0.079	0.034				
		MOV	0.237	0.102				
指令数			360（除智能型功能模块的特殊指令之外）					
恒定扫描（ms） 设置常规扫描时间的功能			0.5——2000（以 0.5 的倍数配置）					用于规定的设定参数值
程序容量（步）*2	程序存储器（驱动器 0）		28k		60k	124k	252k	参见第 6.2 节
存储器容量(字节)	存储卡(RAM)(驱动器 1)		装入存储卡的容量(最大 1M)					参见第 6.5 节
	存储卡(ROM)(驱动器 2)		装入存储卡的容量 (闪存卡最大 4M，ATA 卡最大 32M)					参见第 6.5 节
	标准 RAM(驱动器 3)		64k			256k*3		参见第 6.4 节
	标准 ROM(驱动器 4)		112k		240k	496k	1008k	参见第 6.3 节
	PLC 公共存储器		8k					参见第 14.2.4 节
存储文件的最大数量(文件)	存储器		28		60	124	252*	参见第 6.2 节
	存储卡(RAM)		256					
	存储卡(ROM)	闪存卡	288					参见第 6.5 节
								参见第 6.5 节
		ATA 卡	512					参见第 6.5 节
	标准 RAM		2					每一个用于寄存区和本地软元件的文件，可参见第 6.4 节
	标准 ROM		28		60	124	252*1	参见第 6.3 节
I/O 软元件点(点数)			8192(X/Y0——1FFF)					程序中可用的点数
I/O 点（点数）			4096(X/Y0——FFF)					可对实有 I/O 模块存取的点数

*1. 执行程序的最大数量为 124。超过 124 则不能执行。

*2. 程序步数的最大值可保存在其它软元件的参数中，并能由 QCPU 执行，可通过下列公式求得：（程序容量）-（文件标题容量（缺省值：34 步））程序容量和文件详见第 6 章。

*3. Q12HCPU 和 125HCPU 等系列编号前五位为“02091”或更早的编号，其存储容量为 64k 字节。

*4. 共享 CPU 不能锁存。
当 PLC 的电源切换至 ON 或 QCPU 复位时，共享 CPU 存储器将被清除。

表 3.1 性能规格（续）

项目			型号					备注
			Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU	
软元件点数	内部继电器[M](点数)		缺省 8192(M0~8191)					设定参数值，以指定使用点
	锁存继电器[L](点数)		缺省 8192(L0~8191)					
	通信继电器[B](点数)		缺省 8192(B0~1FFF)					
	定时器[T](点数)		缺省 2048(T0~2047)（对低/高速定时器） 通过指令在低/高速定时器中选择 由参数设定低/高速定时器的测量模块 （低速定时器，1~1000ms，1ms/模块，缺省 100ms） （高速定时器，0.1~100ms，0.1ms/模块，缺省 10ms）					
	积算定时器[ST](点数)		缺省 0(对低/高速定时器) 由指令在低/高速积算定时器中选择 由参数设定低/高速积算定时器中的测量模块 （低速积算定时器，1~1000ms，1ms/模块，缺省 100ms） （高速积算定时器，0.1~100ms，0.1ms/模块，缺省 10ms）					
	计数器[C](点数)		● 正常计数器缺省 1024(C0~1023) ● 间断计数器最大 256（缺省 0 点，由参数设定）					
	数据寄存器[D](点数)		缺省 12288（D0~12287）					
	通信寄存器[W](点数)		缺省 8192（W0~1FFF）					
	信号器[F](点数)		缺省 2048（F0~2047）					
	沿继电器[V](点数)		缺省 2048（V0~2047）					
	文件寄存器	[R]（点数）	● 当使用标准 RAM 时：32768（R0~32768） ● 当使用标准 SRAM 卡时： 以 32768（R0~32767）为单元的块转换最大可用 517120 点 ● 当使用闪存卡（2M 字节）时： 以 32768（R0~32767）为单元的块转换最大可用至为 1040384 点 ● 当使用闪存卡（4M 字节）时： 32768（R0~32767）为单元的块转换最大可用 1042432 点					当使用闪存卡时，只能进行读处理
		[ZR]（点数）	● 当使用标准 RAM 时：32768（ZR0~32767） ● 当使用标准 SRAM 卡时： 517120（ZR0~517119），不需块转换 ● 当使用闪存卡（2M 字节）时： 1040384（ZR0~1040383），不需块转换 ● 当使用闪存卡（4M 字节）时： 1042432（ZE0~1042431），不需块转换					

表 3.1 性能规格（续）

项目		型号					备注
		Q02CPU	Q02HCPU	QO6HCPU	Q12HCPU	Q25HCPU	
软元件点数数	特殊通信继电器[SB](点数)	2048(SB0~7FF)					软元件点数固定
	特殊通信寄存器[SW](点数)	2048(SW0~7FF)					
	步进继电器[S](点数)	8192(S0~8191)					
	变址寄存器[Z](点数)	16（Z0~15）					
	指针[P](点数)	4096（P0~4095），设定参数值，以选择系统文件内指针/公共指针的使用范围					
	中断指针[I](点数)	256（IO~255）设置参数值以选择，中断指针（128~131）（0.5~1000ms，0.5ms 模块）的周期性间隔时间。 缺省 128：100ms 129：40ms 130：20ms 131：10ms					
	特殊继电器[SM](点数)	2048（SM0~2047）					软元件点数固定
	特殊寄存器[SD](点数)	2048（SD0~2047）					
	功能输入[FX](点数)	16（FX0~F）					
	功能输出[FY](点数)	16（FY0~F）					
	功能寄存器[FD]（点数）	5（FD0~4）					
直接通信软元件		该软元件用于直接访问通信软元件。 仅用于 MELSECNET/10H。 规定型式：J□\W□□，J□□\B□□，J□□\SW□□，J□□\SB□□					
智能型功能模块软元件		该软元件用于直接访问智能型功能模块缓冲存储器。 规定型式：U□□\G□□，					
锁存（断电补偿）范围		L0~8191（缺省） （锁存范围可用于规定：B，F，V，T，ST，C，D，W）					通过设定参数值来指定
远程运行/暂停接触器		以 X0~1FFF 可指定每次运行/暂停接触器					
时钟功能		年，月，日，时，分，秒，周日 （自动检测闰年） 精度：-3.18~+5.25 秒(TYP.+2.12 秒)/d，0℃ 精度：-3.93~+5.25 秒(TYP.+1.90 秒)/d，25℃ 精度：-14.69~+3.53 秒(TYP.-3.67 秒)/d，55℃					
允许瞬间的掉电间隔时间		由电源模块决定					
5V 直流内部消耗电流（A）		0.60	0.64	0.64	0.64	0.64	
重量[kg(磅)]		0.20 (0.44)	0.20 (0.44)	0.20 (0.44)	0.20 (0.44)	0.20 (0.44)	
外形尺寸[mm(英寸)]		98(高)×27.4(宽)×89.3(深)(3.86×1.08×3.52)					

备注

QCPU 的通用规格详见 QCPU（Q 模式）用户手册（硬件设计/维护-和检查）。

3 - 3

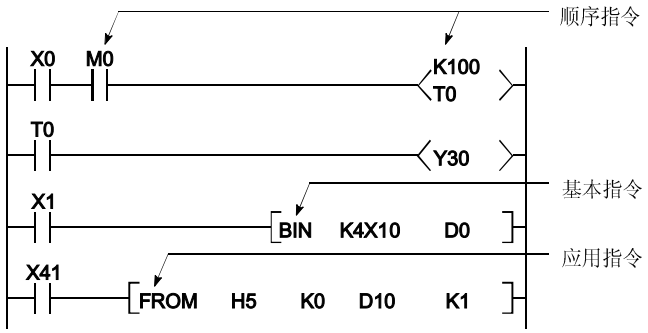
4 顺控程序配置和执行条件

QCPU 可执行顺控程序和 SFC 程序。
本章阐述顺控程序配置和执行条件。
本手册不阐述有关 SFC 程序的问题。
有关 SFC 程序，详见 QCPU（Q 模式）QnACPU 编程手册（SFC）。

4.1 顺控程序

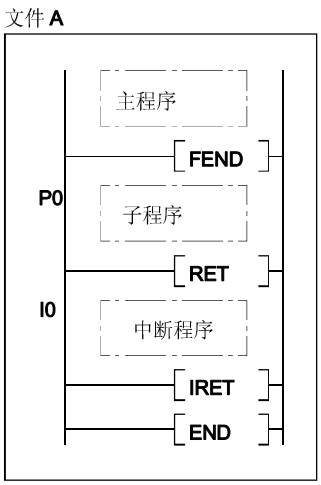
(1) 顺控程序的定义

(a) 顺控程序用 QCPU 顺序指令，基本指令和应用指令等编制。



(b) 共有三种顺控程序类型：主程序，子程序和中断程序。
有关这些程序，请参见本手册下列各章节：

- 主程序 : 第 4.1.1 节
- 子程序 : 第 4.1.2 节
- 中断程序 : 第 4.1.3 节



备注

1) 有关 QCPU 的顺序指令，基本指令和应用指令，详见“QCPU（Q 模式）QnACPU 编程手册（公共指令）”。

(2) 顺控程序的写格式

用梯形图模式或列表模式可对顺控程序编程。

(a) 梯形图模式

- 梯形图模式基于继电器控制顺序的梯形图。编程的表达方式类似于继电器控制顺序的梯形图
- 继电器符号语言编程以梯形块为单位出现。
  一个梯形块是顺控程序处理中的最小单位，该梯形图从左边的总线开始到右边的总线结束。

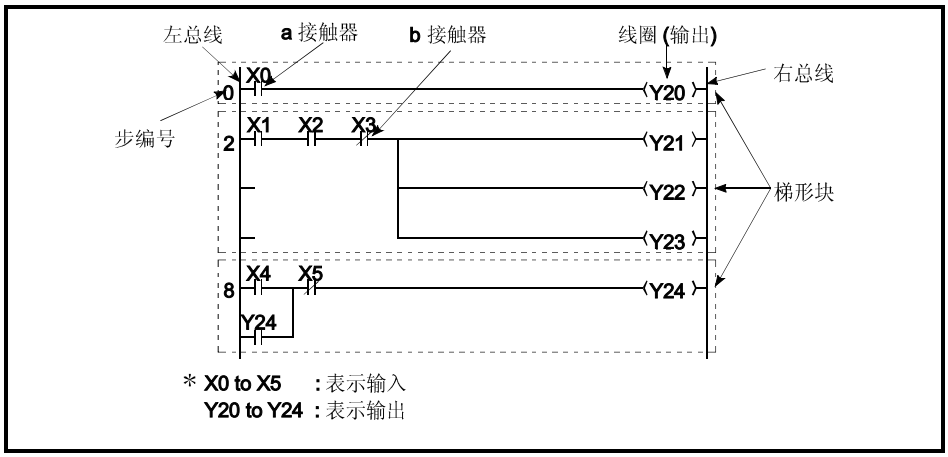


图 4.1 梯形块

(b) 列表模式

列表模式使用专用指令代替那些梯形图模式中用的接触器符号，线图符号等。

接触器 a，接触器 b 和线圈指令如下：

- a 接触器 .....LD, AND, OR
- b 接触器 .....LDI, ANI, ORI
- 线圈.....OUT

(2) 程序处理

顺控程序按顺序处理，从 0 步开始，到 END/FEND 指令结束。

梯形模式梯形块处理从左总线开始向右进行。当一个梯形块完成后，处理就向下进入下一个梯形块。

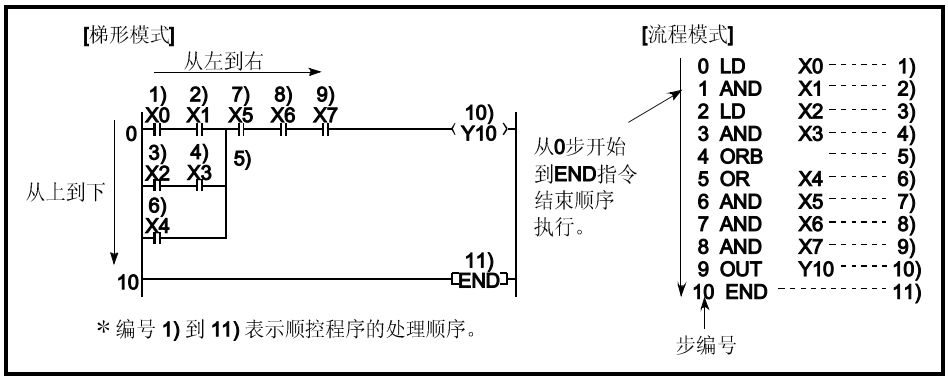


图 4.2 顺控程序处理

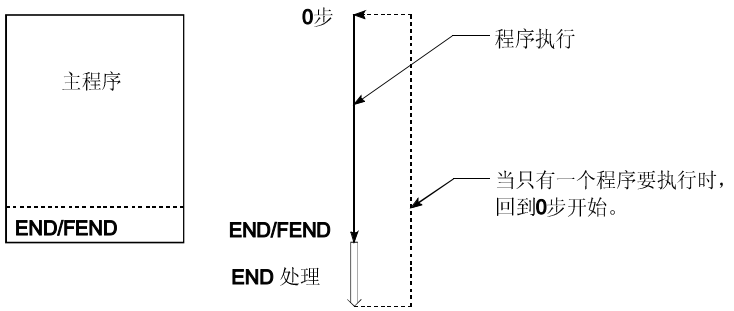
4.1.1 主程序

(1) 主程序的定义

(a) 主程序是从 0 步开始到 END/FEND 指令结束的程序。*1

(b) 主程序的执行是从 0 步开始到 END/FEND 指令结束。

1) 如果只要执行一个程序要，那么在 END/FEND 指令处理后再从 0 步开始进行处理。



2) 如果有多个程序要执行，那就在 END/FEND 指令后按照指定的不同执行条件来进行处理。

(2) 主程序的执行条件*2

如果多个程序要执行，则按照应用情况，用 PLC 参数程序来指定下列五种执行条件。

- 初始执行程序 : 参见第 4.2.1 节
- 扫描执行程序 : 参见第 4.2.2 节
- 低速执行程序 : 参见第 4.2.3 节
- 待机型程序 : 参见第 4.2.4 节
- 固定扫描型程序 : 参见第 4.2.5 节

备注

1) *1: 关于 END/FEND 指令，详见“QCPU（Q 模式）QnACPU 编程手册（公共指令）”。

2) *2: 如果只有一个程序要执行，也可在不用 PLC 参数程序指定执行条件而在“扫描执行型程序”条件下处理。

4.1.2 子程序

(1) 子程序的定义

(a) 子程序是从指针(P□)开始到 RET 指令结束的程序。

(b) 子程序只有当用从主程序来的调用指令（如 CALL（P），FCALL（P））调用时，才能执行。

(c) 子程序的应用

1) 当一次扫描中一个程序执行多次时，使用子程序可以减少总步数。

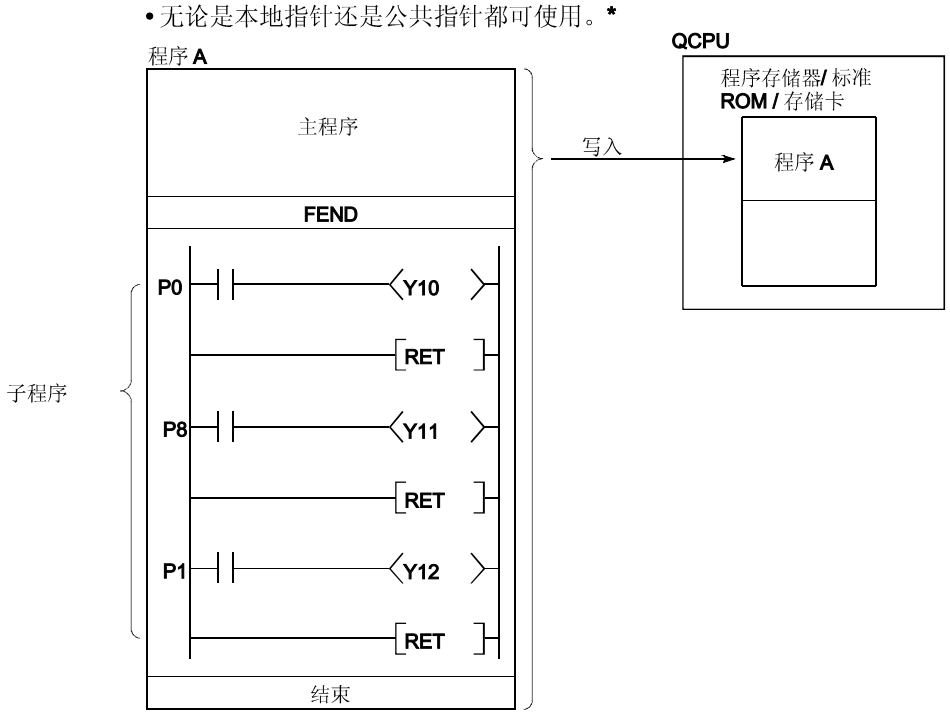
2) 当所给条件得到满足时才可执行程序时，使用子程序可以减少恒定执行程序的步数。

(2) 子程序的管理

子程序可以在主程序（在 FEND 指令后）之后编制，主程序和子程序可当作一个程序进行管理。

(a) 在主程序之后编制时

- 子程序编制在主程序的 FEND 和 END 之间。
- 当编制多个子程序时，因为编制子程序时没有顺序的限制条件，因此没有必要按升序设定指针。
- 无论是本地指针还是公共指针都可使用。*



备注

1) *: 详见第 10.9 节“关于本地和公共指针”。

2) 详见第 10.8 节“子程序的嵌套”。

(b) 将子程序作为一个单独程序使用时

子程序可以作为一个单独、离散的程序（待机型程序）进行管理。（详见第 **4.2.4** 节“有关待机型程序”）。

4.1.3 中断程序

(1) 中断程序的定义

(a) 中断程序是从中断指针（I□）开始，到 IRET 指令结束的程序。*1

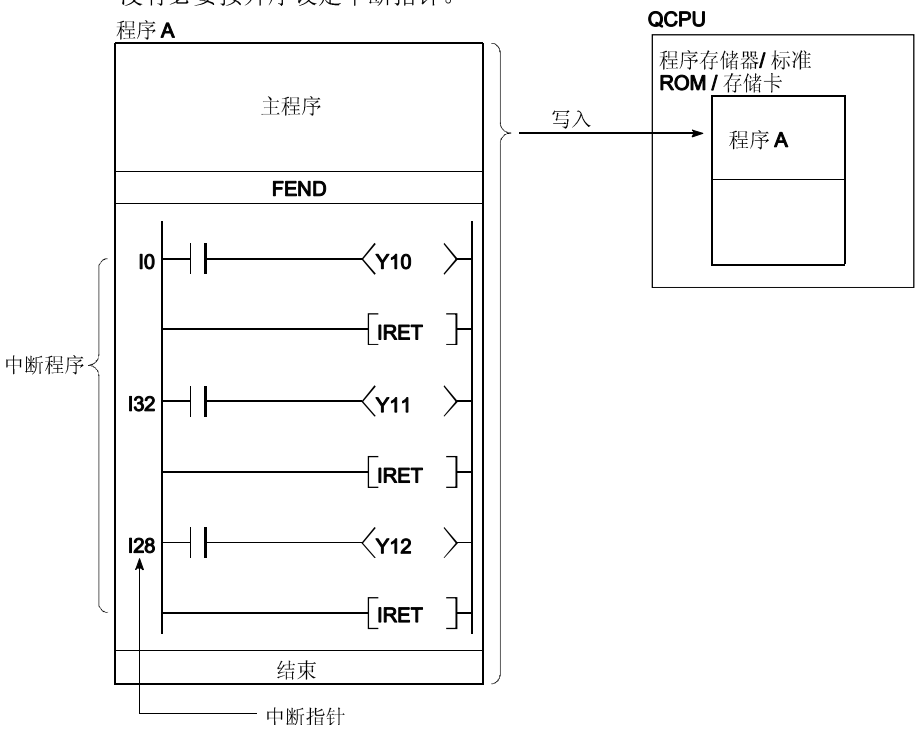
(b) 只有当中断因素发生时，才执行中断程序。*1

(2) 中断程序的管理

中断程序编制在主程序（FEND 指令后）之后，将主程序和子程序结合作为一个程序来管理。

(a) 当编制在主程序之后时

- 中断程序编制在主程序 FEND 和 END 指令之间。
- 当编制多个中断程序时，因为中断程序的编制对顺序没有限制条件，因此没有必要按升序设定中断指针。



(b) 将中断程序作为单独程序使用时

中断程序可作为单独、离散的程序管理（待机型程序）。（详见第 4.2.4 节关于待机型程序）。

但是在程序被 QCPU 执行时，同样的中断程序指针编号只能使用一次。

备注

1) *1: 详见 10.10 节“关于中断因素和中断指针”。

2) *2: 在中断程序的最后一步需要一个 END 指令。缺少 END 指令，可能产生“END NO EXECUTE”错误。

(3) 执行中断程序

(a) 为了用中断指针 132~147 执行一个中断程序，要用 1MASK 和 E1 指令来得到中断许可。*1

1) 如果一个中断因素先于一个中断许可状态发生，当建立起一个中断许可状态时，可以执行对应于该因素的中断程序。

2) 如果一个中断因素在 STOP/PAUSE 时产生，随着回到运行状态而建立一个“中断许可”条件，可以执行对应于该因素的中断程序。

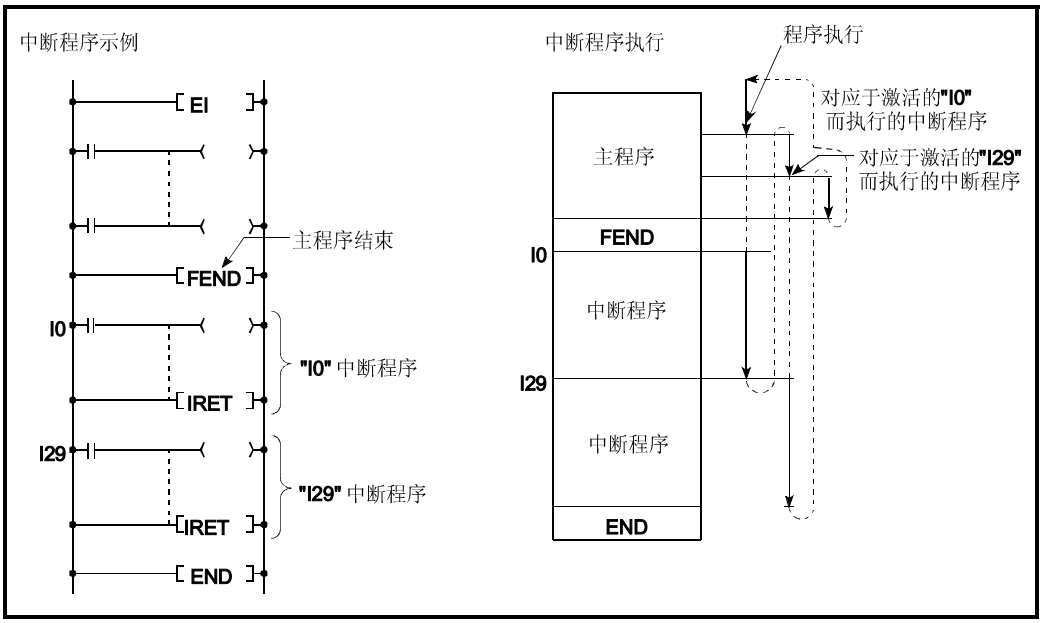


图 4.3 中断程序执行

(b) 当一个中断因素发生时，与这一因素相对应的中断指针编号的中断程序将被执行。

但是按当时的条件，中断程序的执行也不同：

1) 当多个中断程序被同时激活时，
将按顺序执行程序，从最高优先级中断指针编号开始。*2
剩下的中断程序留作待机，直至高一级优先级中断程序完成。

2) 当执行一个指令时：
当执行指令时，禁止中断。
如果在一个指令执行时发生中断因素，将在指令处理完成后再执行中断程序。

备注

1)*1: 关于 1MASK 和 E1 指令，详见“QCPU（Q 模式）QnACPU 编程手册（公共指令）”。
为执行中断指令 10—131 和 148—1255，用 E1 指令将中断程序输入达到允许中断状态。

2) *2: 关于间断程序的优先级排列详见 10.10 节。

3) 在网络刷新时的中断：
如果在网络刷新过程中产生中断因素，该网络刷新操作就被暂停，执行中断程序。
这表明，使用一个指定的软元件，诸如在 MESSECNET/10H 网络系统中指定的链接刷新操作也不能使“每个站循环数据块的保证”安全可靠。*1

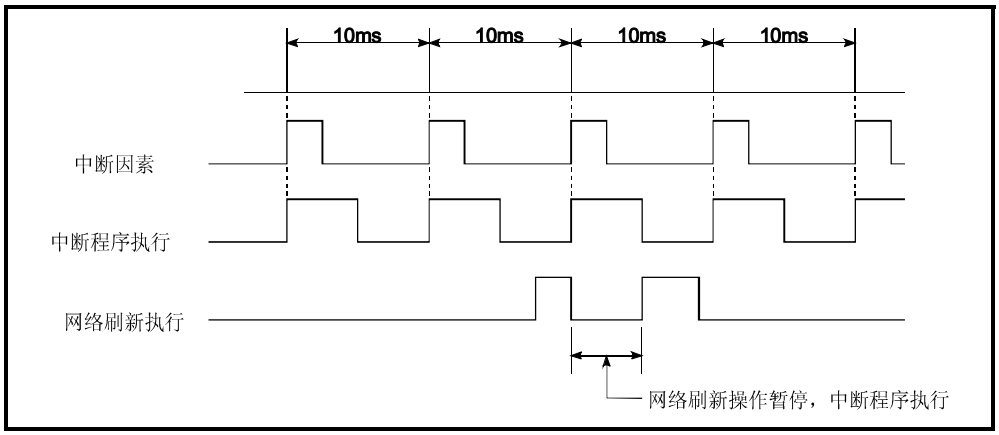


图 4.4　在网络刷新操作中的中断

4) 在 END 处理中的中断：
如果在恒定扫描的 END 处理等待期内产生一个中断因素，将执行对应于这个因素的中断程序。

(c) 当从一个扫描执行型程序或低速执行程序切换到中断程序时，有关变址寄存器处理的内容详见第 10.6.2 节。

(4) 中断程序的高速处理和花费时间

当执行一个中断程序时，通过缺省，QCPU 执行下列处理：
• 隐藏和恢复变址寄存器。
• 隐藏和恢复一个使用中的文件寄存器的文件名。
如果在 PLC 参数对话框中，在 PLC 系统设定表中选择“高速执行”，则上列处理就不执行。
这就可能缩短执行一个中断程序所需的时间。

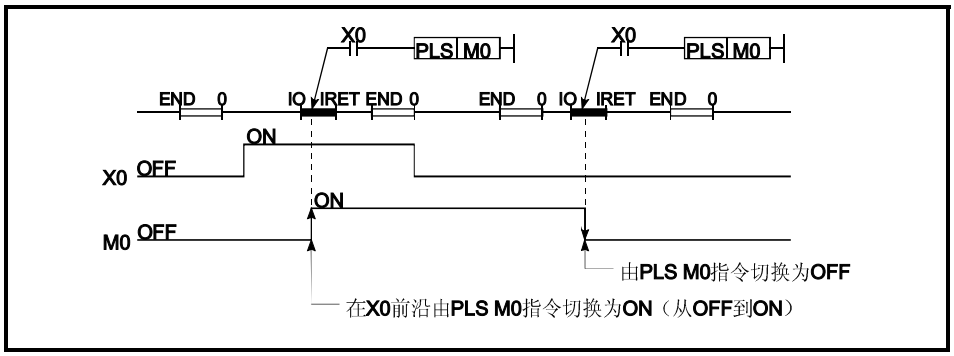
CPU 型号	花费时间 (μs)	
	低速执行	高速执行
Q02CPU	380	230
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	165	100

备注

*1: 关于“每个站循环数据块的保证”参见“MELECNET/H 网络系统参考手册。”

(5) 程序编制的限制条件

(a) 由一个中断程序中的 PLS 指令切换为 ON 的软元件，将保持 ON 的状态，直至中断程序再次执行。



(b) 在执行中断程序过程中，会建立 DI 状态（中断禁止）。
在中断程序中不得执行 EI/DI 指令。

(c) 定时器不得用于中断程序。
由于用于 OUT T□ 指令的定时器更新当前值，并切换接触器的 ON 和 OFF，因此在中断程序中使用定时器会使得正常的时间计数变得不可能。

(d) 执行中断程序和固定扫描执行型程序，在计算扫描时间和执行时间时，该测量的时间值应该加上执行中断程序或恒定循环执行型程序的值。
因此，在执行中断程序/恒定循环执行型程序时，该值就保存在下列特殊寄存器中，而 GX Developer 监控器的值也比不执行中断程序/恒定循环执行型程序的要长。

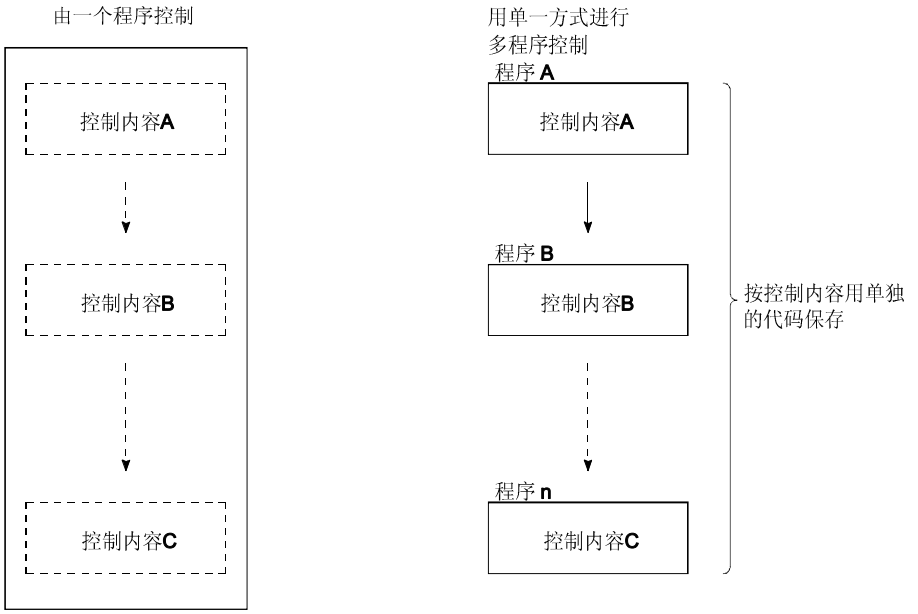
1) 特殊寄存器
- SD520，SD521：当前扫描时间
- SD522，SD523：初始扫描时间
- SD524，SD525：最小扫描时间
- SD526，SD527：最大扫描时间
- SD528，SD529：低速时的当前扫描时间
- SD532，SD533：低速时的最小扫描时间
- SD534，SD535：低速时的最大扫描时间
- SD540，SD541：END 处理时间
- SD542，SD543：恒定扫描等待时间
- SD544，SD545：低速执行程序的累积执行时间
- SD546，SD547：低速执行程序的执行时间
- SD548，SD549：扫描程序执行时间
- SD551，SD552：服务间隔时间

2) GX Developer 监控器值
- 执行时间测量
- 扫描时间测量
- 恒定时间

4.2 程序执行型式

由 QCPU 执行的程序可保存在 QCPU 程序存储器、标准 ROM 或存储卡中。
程序可以作为单一程序保存在标准 ROM 或存储卡中，但也可以看作按控制功能分割成的多程序。
这就允许在编程的过程中，由 n 个程序设计人员将其分割，这样，程序设计人员就可对每种操作设计单独的程序，并保存在标准 ROM 或存储卡中。

当多程序被 QCPU 执行时，必须指定程序所设定的“程序名（文件名）”和“执行类型”。

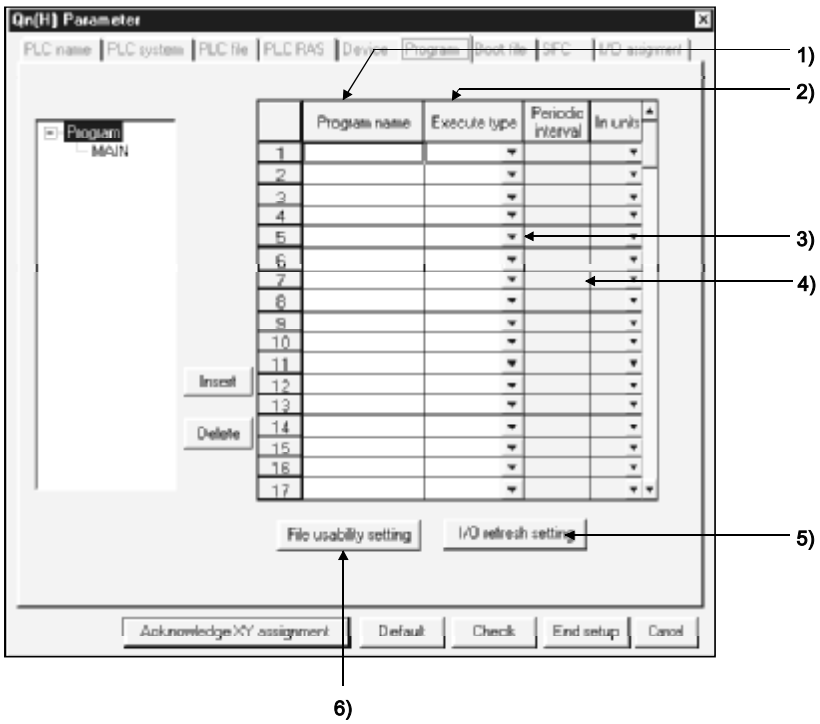


4 - 10

(1) 设置执行类型

(a) 为了执行 n 个程序，在 PLC 参数对话框的程序设置表中，指定每个程序的“程序名”和“执行类型”。

QCPU 将按设置所规定的执行类型次序，执行所选择的程序。



1) 程序名

本栏用来为由 QCPU 执行的程序指定一个程序名（文件名）。

2) 执行类型

本栏用来为已确定的“程序名”指定一个执行类型，见（b）。

3) 固定扫描

本栏用来指定用于执行某种类型文件的时间间隔。

固定扫描设定范围由时间间隔的单位决定。

间隔如下：

- 在“ms”单位时：0.5—999.5
- 在“s“单位时：1—60

4) 单位

本栏用来指定固定扫描时间间隔的单位（ms/s）。

4 - 11

5) 文件使用设置

在 PLC 参数对话框的 PLC 文件设置表中有几种选择：文件寄存器，软元件的初始值，注释和本地软元件。对每个程序都需指定是否要在这几种选择中使用文件。通过缺省，选择“使用 PLC 文件设定”。如果选择“不使用”，那么文件的使用设置如下表所列。

设置项目	当选择“不使用”时的处理
文件寄存器	程序中不能使用文件寄存器。
软元件初始值	当程序文件名和软元件初始值相同时，不设置软元件初始值。
注释	在程序中不得使用注释。
本地软元件	在程序转换时本地软元件不得隐藏和恢复。

6) I/O 刷新设置

QCPU 使用 I/O 刷新设置更新 I/O 模块和智能型功能模块的输出和输入。

I/O 刷新设定按钮用来更新所选程序的范围。如果你想在执行一个固定扫描执行型程序之前接收一个输入（X）或产生一个输出（Y），你可将 I/O 刷新设置用于扫描执行型程序。

(b) 共有下列五种执行类型：

1) 初始执行（初始化）

只有当接通电源或产生 STOP–RUN 切换时，本程序类型执行一次。（参见第 4.2.1 节）

2) 扫描执行（扫描）

从初始执行程序执行后的扫描开始，每次扫描，该程序就执行一次。（参见第 4.2.2 节）

3) 低速执行（低速）

只有当恒定扫描设置或设置低速执行程序的时间时，才能执行该类型程序。

- 当已设置好恒定扫描时，该程序只有在扫描执行型程序的盈余时间内才能执行。
- 当执行一个低速执行程序的时间设定好时，该程序在此时间内执行。（参见第 4.2.3 节）

4) 待机（等待）

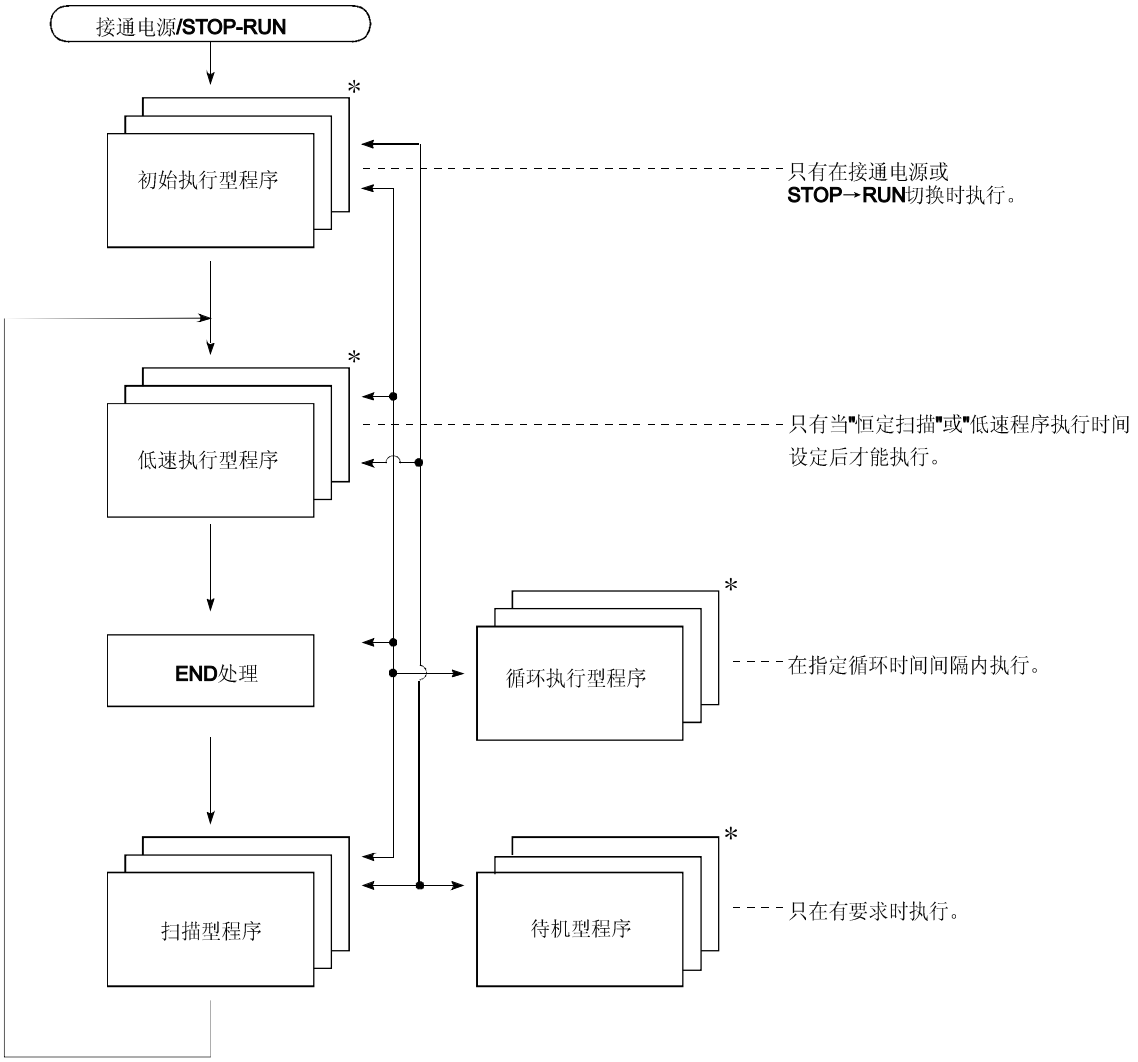
只有当要求执行该程序时才执行。（参见第 4.2.4 节）

5) 固定扫描执行（固定扫描）

只有在 PLC 参数对话框中的程序设置表上“固定扫描”和“单位”栏指定的时间间隔里，才执行该程序。（参见第 4.2.5 节）

(c) 可执行程序的扫描时间可以在程序表的监控器上检查（除固定扫描执行型程序外）。（参见第 7.11.1 节）

(2) 下列为接通电源或 STOP–RUN 切换时产生的程序操作步骤

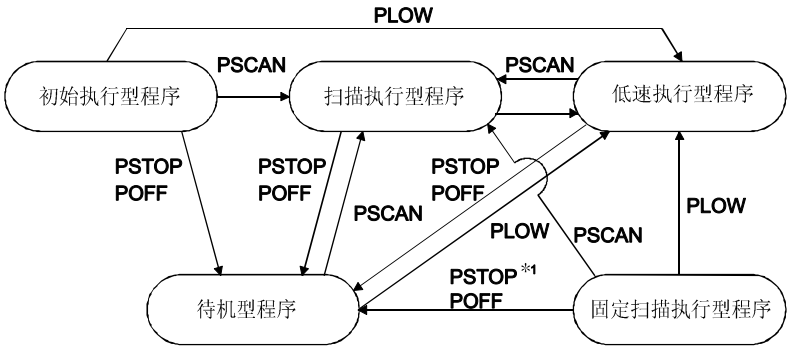


要点
(1) QCPU 并不需要所有的执行类型都作设置。 需要的类型用"*"标记，如初始执行，低速执行，待机执行和固定扫描执行型程序。

4 - 13

(3) 改变执行类型

(a) 在执行顺控程序的任何时间，PLC 参数对话框中的程序设定表中的执行类型的设定都可以改变。用 PSCAN，PLOW，PSTOP 或 POFF 指令来改变程序的执行类型。



(b) 下表为用 PSCAN，PLOW，PSTOP 或 POFF 指令改变程序执行类型的时间。

执行指令 / 改变前执行指令	PSCAN	PSTOP	POFF	PLOW
改变前执行指令	没有变化-保持扫描执行类型。	变成待机型	在下一次扫描时输出为 OFF。此后在下次扫描时变为待机型。	变成低速型。
初始执行类型	变成扫描执行型。			
待机型		没有变化-保持待机型。	没有处理。	
低速执行型	低速执行型执行停止：从下一次扫描变成扫描执行型（从 0 步开始执行）。	低速执行型执行停止：从下一次扫描变成扫描执行型。	低速执行型执行停止，下一次扫描输出转为 OFF。此后从下一次扫描变为待机型。	没有变化–保持低速执行型。
固定扫描执行型	变成扫描执行型	变成待机型	下一次扫描输出要为 OFF。此后下一次扫描变成待机型。	变成低速型

要点

*1: 如果用 PSTOP 或 POFF 指令将固定扫描执行型程序定义为“待机型”，则不再执行这一程序。

4 - 14

4.2.1 初始执行型程序

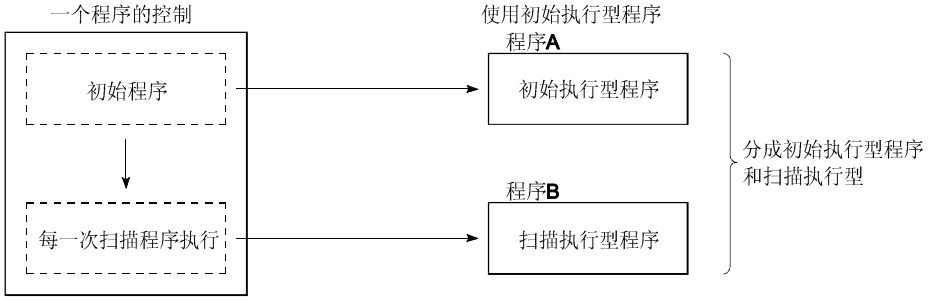
(1) 初始执行型程序的定义

(a) 只有在电源 ON，或从 STOP–RUN 切换开始时，才执行一次初始执行型程序。

(b) 该程序的执行类型在 PLC 参数程序中指定为“初始”。

(c) 对于智能型功能模块，由于初始处理的方式相同，初始执行程序只执行一次，不需要随后扫描。

包括一个完成软元件的指令不能用于初始执行型程序，因为完整的软元件需要 n 次扫描才能完成执行。

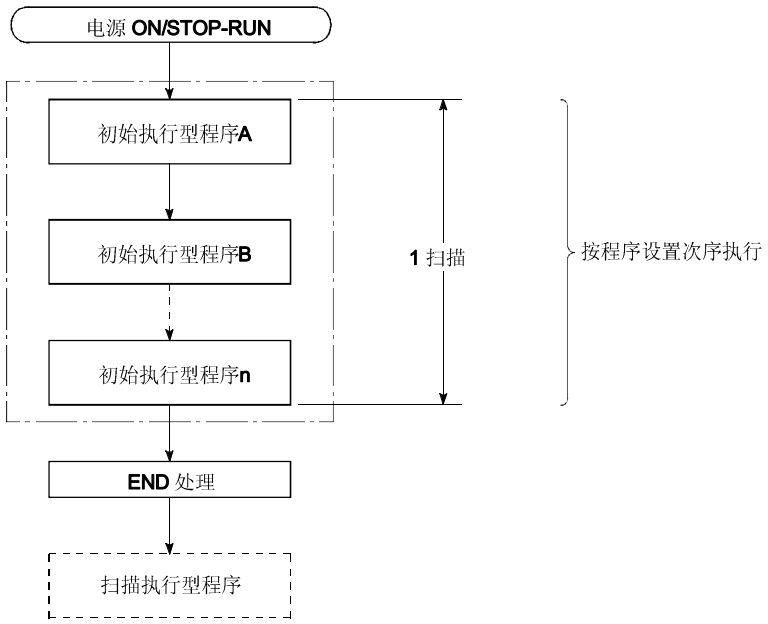


(2) 使用多个初始执行型程序

在使用多个初始执行型程序时，它们按升序的 PLC 参数程序编号一个一个地执行。

(3) END 处理

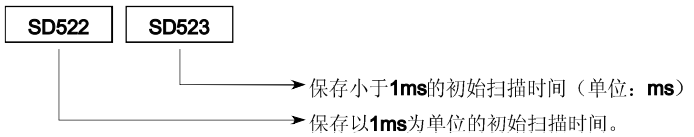
当所有的初始执行类程序完成后，才进行 END 处理，然后从下一次扫描执行“扫描执行型程序”。



(4) 初始扫描时间

(a) 这是初始执行型程序的执行时间。
如果使用多个初始扫描型程序，这是执行所有程序的时间。
当一个初始执行型程序在运行时，执行中断程序/固定循环执行型程序，则中断程序/固定循环执行型程序的执行时间要加上初始执行型程序的时间。

(b) QCPU 测量初始扫描时间，并将结果保存在特殊寄存器里（SD522，SD523）。*1
初始扫描时间可在监控 SD522，SD523 特殊寄存器时检查。



如果 SD522 的值是"3"，SD523 的值是"400"，则初始扫描时间为 3.4 ms。

(5) 初始执行监控时间

(a) 初始执行型程序的执行时间可由定时器监控。不设置缺省值。
当监控初始执行型程序的执行时间时，在 PLC 参数的 PLC RAS 设定表上指定初始执行监控时间为 10ms—2000ms。（设定单位：10ms）

(b) 在初始执行型程序执行完成后执行低速执行型程序。
为了使用低速执行型程序，指定时间要长于初始扫描时间和低速执行型程序执行时间之和。

(c) 当初始扫描时间超过设定的初始执行监控时间，产生“WDT ERROR 出错代码，5000”，QCPU 操作停止。

注意
(1)*1: 保存在特殊寄存器内的初始扫描时间精度为±0.1ms。 即使在顺控程序中执行 WDT 复位指令，初始扫描时间计数仍会继续。 (2) 当初始执行监控时间已指定时，计数值会有 10ms 的误差。 因此，当初始扫描时间范围在 10ms<t<20ms 时，10ms 的监控时间设定(t)也会有“WDT ERROR”的结果。

4.2.2 扫描执行型程序

(1) 扫描执行型程序的定义

(a) 从初始执行型程序执行后的扫描开始，每一次扫描，扫描执行型程序就执行一次。

(b) 在 PLC 参数的程序中设定执行类型为“扫描”。

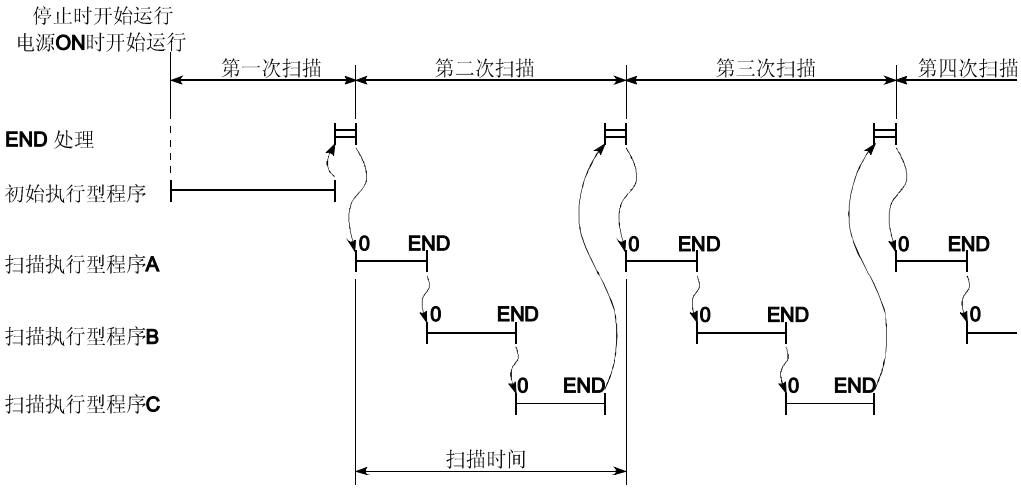
(2) 执行多个扫描执行型程序

在使用多个扫描执行型程序时，按升序的 PLC 参数的程序编号的一个接一个地执行。

(3) END 处理

当所有的初始执行型程序执行完成后，才出现 END 处理，然后从下一次扫描开始执行扫描执行型程序。

当执行 n 个扫描执行型程序时，每个程序都可实施 END 处理。为此在每个扫描执行型程序结束时，都包含一个 COM 指令。



(4) 恒定扫描设置*1

当指定一个恒定扫描时，在每个指定的恒定扫描时间内执行扫描执行型程序。

备注

1)*1: 在规定时间间隔内，“恒定扫描”功能重复执行扫描型程序。
有关恒定扫描详见第 7.2 节

注意

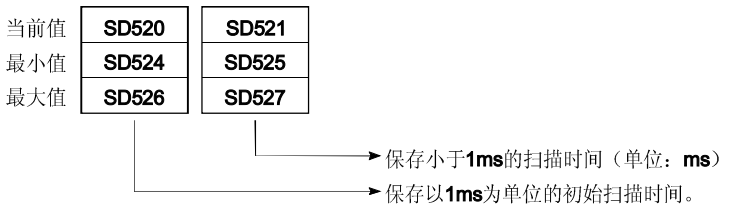
(1) 在扫描执行型程序运行过程中执行中断程序时，有关变址寄存器的处理详见第 10.6.2 节。

4 - 17

(5) 扫描时间

(a) “扫描时间”是扫描执行型程序和 END 处理执行时间的总和。
如果使用多个扫描执行型程序，“扫描时间”就是执行所有程序所需时间的总和。
当执行中断程序/固定循环执行型程序时，要加上中断程序/固定循环执行型程序的执行时间才是扫描时间。

(b) 在 QCPU 中计算扫描时间的当前值，最小值和最大值，其结果保存在特殊寄存器里（SD520，SD521，和 SD524 到 SD527）。*1
因此初始扫描时间可由监控 SD520，SD521，SD524 到 SD527 特殊寄存器来检查。



如果 SD520 之值为 3，SD521 之值为 400，初始扫描时间则为 3.4ms。

(6) WDT（警戒定时器）

该定时器监控扫描时间，其缺省值设定为 200ms。
在 PLC 参数的 PLC RAC 设置中，WDT 设定值指定在 10ms—2000ms 的范围内。
（设定值单位：10ms）
在使用低速执行程序时，WDT 的设定值大于扫描时间加上低速执行程序的执行时间。
如果扫描时间（扫描执行程序的执行时间+低速执行程序的执行时间）超过 WDT 的设定值，将出现“WDT ERROR（出错代码：5000）”，QCPU 操作停止。

注意

(1)*1: 保存在特殊寄存器里的扫描时间精度为±0.1ms。在顺控程序中即使执行警戒时间复位指令，也将继续扫描时间的计数。
(2) WDT 计量误差为 10ms。
因此，当扫描时间范围为 10ms<t<20ms 时, 10ms 的 WDT 设置时间(t)也会有“WDT ERROR”的结果。

备注

1) 用 GX Developer 的程序监控表检查正在执行的程序的执行时间。关于 GX Developer 程序监控表，详见第 7.11.1 节。

4.2.3 低速执行型程序

(1) 低速执行型程序的定义

(a) 只有在“恒定扫描盈余时间”或在“低速执行程序执行时间”的指定时间内才执行低速执行型程序。

1) 为了使恒定扫描时间具有增强的控制精度，在 PLC 参数的 PLC RAS 上设定时可指定一个恒定扫描时间。
（设定值范围：0.5—2000ms，设定值单位：0.5ms）

2) 为了保证在每次扫描时低速执行型程序的执行时间，在 PLC 参数的 PLC RAS 中指定一个低速执行程序的执行时间。
（设定值范围：1—2000ms，设定值单位：1ms）

3) 为了执行低速执行型程序，既可设定恒定扫描时间，也可设定低速执行程序的执行时间。

(b) 在 PLC 参数的程序中，将低速执行程序的执行类型设定为“低速”。

(c) 低速执行型程序用于那些不需要每次执行扫描的程序，例如打印输出程序。

(2) 执行多个低速执行型程序

在使用多个低速执行型程序时，在 PLC 参数的程序中按升序编号一个一个地执行。

(3)低速执行型程序的一次扫描执行时间

(a) 如果所有的低速执行型程序在一次扫描中完成操作，并有盈余时间，此后的执行处理取决于 SM330 寄存器的 ON/OFF 状态和低速执行型程序的执行条件。

1) 异步法（SM330=OFF）
该方法为在盈余时间内继续进行低速执行型程序操作。

2) 同步法（SM330=ON）
该方法为即使有盈余时间，低速执行型程序操作也不继续进行，操作从下一次扫描重新开始。

低速执行型程序操作方法	SM330 设定状态	低速执行型程序的执行条件	
		当设定恒定扫描时间时	当设定低速执行程序执行时间时
异步法	OFF	再次执行低速执行型程序 *1。	再次执行低速执行型程序*2。
同步法	ON	产生恒定扫描等待时间 *3。	扫描执行型程序操作开始*4。

4 - 19

*1 如果已指定恒定扫描时间，则低速执行型程序将在恒定扫描盈余时间内执行。
因此，低速执行型程序的执行时间将随每次扫描而变化。
如果恒定扫描盈余时间为 0.5ms 或更少，则低速执行型程序将完全不执行，因此指定的恒定扫描时间设定值所提供的盈余时间要大于 0.5ms。

*2 如果已指定低速执行型程序的执行时间，按照其时间设定，低速执行型程序将反复执行。
因此其扫描时间将随每次扫描而变化。

*3 如果已指定恒定扫描时间，在完成低速 END 处理后盈余时间即为等待时间，如果恒定扫描时间已过，则扫描执行型程序开始执行。
这就表示每次扫描的扫描时间是一个常量。
但是，当常量扫描后的盈余时间少于 0.5ms，低速执行型程序就不能执行。如果使用低速执行型程序，设定的恒定扫描时间要使盈余时间为 0.5ms 或更长。

*4 如果“低速执行型程序执行时间”已指定，则在低速 END 处理完成后将不管此盈余时间，开始扫描执行型程序的操作。
这就表示每次扫描时间都不同。

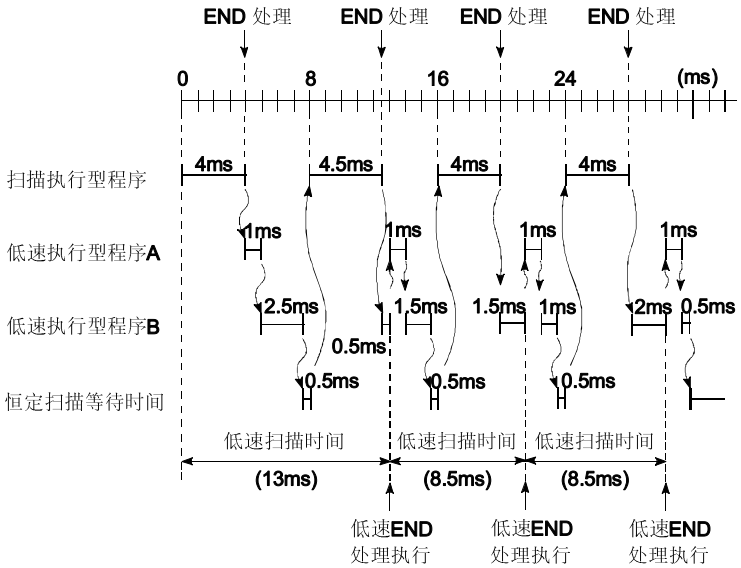
(b) 如果在恒定扫描盈余时间或在低速执行型程序执行时间内，不能处理低速执行型程序，则程序执行暂时停止，剩余的程序在下一次扫描时执行。

1：异步法

(1) 恒定扫描时间的设定

在下列条件时出现的低速执行型程序操作说明如下。

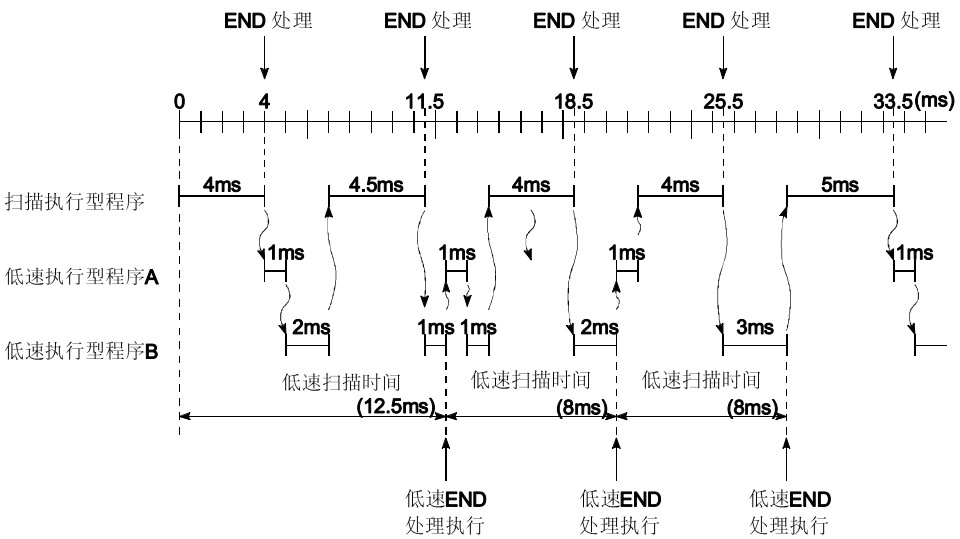
- 恒定扫描时间 : 8ms
- 总的扫描执行型程序时间 : 4ms - 5ms
- 低速执行型程序 A 执行时间 : 1ms
- 低速执行型程序 B 执行时间 : 3ms
- END 处理/低速 END 处理 : 0ms (0 ms 用以简化说明)



(2) 低速执行程序执行时间设定

在下列条件时出现的低速执行型程序操作说明如下。

- 低速程序执行时间 : 3ms
- 总的扫描执行型程序时间 : 4ms - 5ms
- 低速执行程序型 A 执行时间 : 1ms
- 低速执行程序 B 执行时间 : 3ms
- END 处理/低速 END 处理 : 0ms (0 ms 用以简化说明)

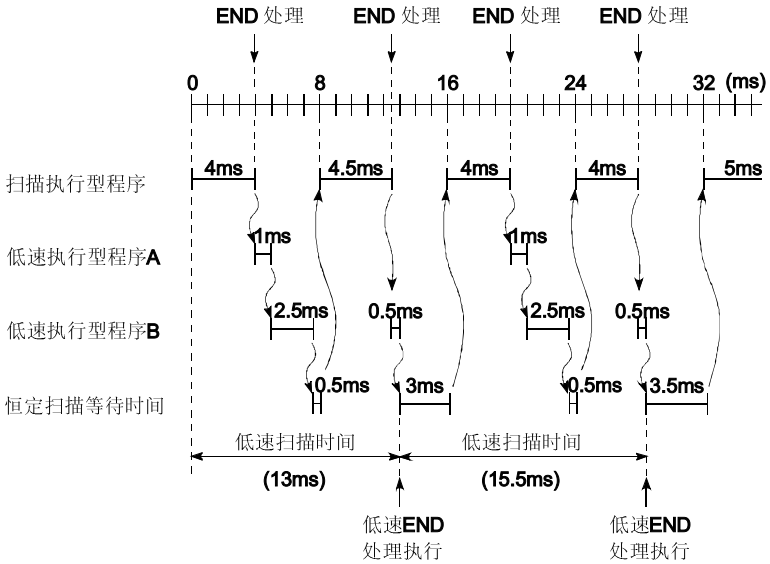


2：同步法

(1) 恒定扫描时间的设定

在下列条件时出现的低速执行型程序操作说明如下。

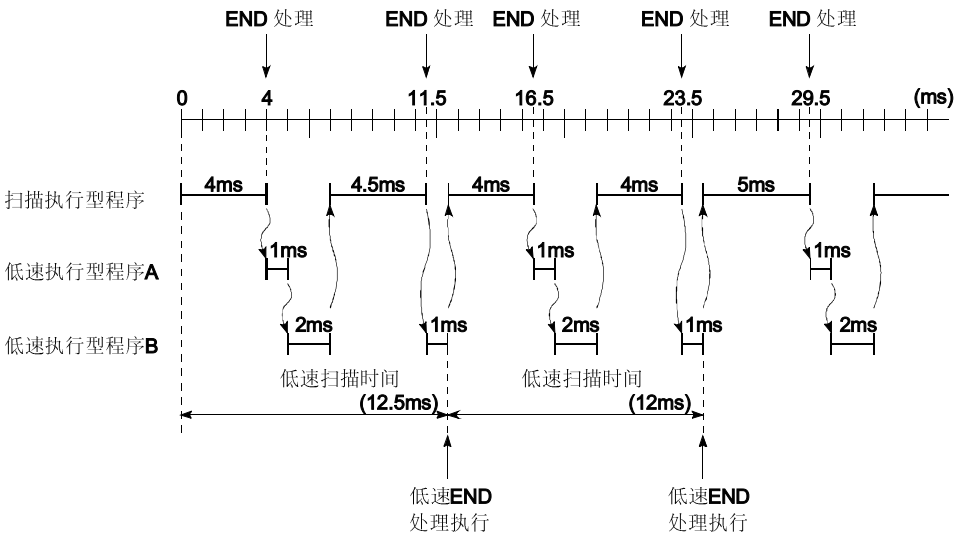
- 恒定扫描时间 : 8ms
- 总的扫描执行型程序时间 : 4ms - 5ms
- 低速执行型程序 A 执行时间 : 1ms
- 低速执行型程序 B 执行时间 : 3ms
- END 处理/低速 END 处理 : 0ms (0 ms 用以简化说明)



(2) 低速执行程序执行时间设定

在下列条件时出现的低速执行型程序操作说明如下。

- 低速程序执行时间 : 3ms
- 总的扫描执行型程序时间 : 4ms - 5ms
- 低速执行型程序 A 执行时间 : 1ms
- 低速执行程序 B 执行时间 : 3ms
- END 处理 : 0ms (0 ms 用以简化说明)



(4) 低速执行型程序编制的注意事项

(a) 当扫描执行型程序切换到低速执行型程序时，所出现的变址寄存器处理情况详见第 10.6.1 节。

(b) 低速执行型程序的运行过程中，执行中断程序/固定扫描执行型程序时所出现的变址寄存器的处理情况，详见第 10.6.2 节。

(c) 低速执行型程序的执行时间设定为：[扫描时间]+[低速执行型程序执行时间]之和小于 WDT 的设定值。

(d) COM 指令不得用于低速执行型程序。

(e) 低速执行型程序也可以用执行初始执行型程序的扫描来执行。
在扫描执行型程序完成后，为使低速执行型程序的运行有效，用 SM402 和 SM403 为电路建立互锁。

(5) 低速 END 处理

所有的低速执行型程序执行时，都执行低速 END 处理。

下列处理用于执行低速 END 处理。

- 低速程序专用继电器/特殊寄存器设置
- 运行时写入低速执行程序
- 低速扫描时间的测量
- 低速执行型程序警戒定时器（WDT）的复位

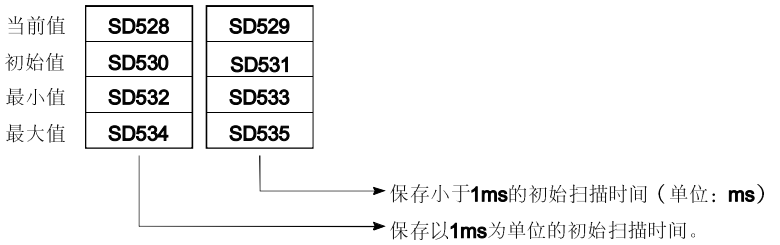
当低速 END 处理完成后，再从头开始执行低速执行型程序。

要点
(1) 在执行低速执行型程序时，恒定扫描时间可能会偏离一个“最大指令处理时间+低速 END 处理时间”的量。

(6) 低速扫描时间

(a) 低速扫描时间为低速执行程序的执行时间与低速 END 处理时间之和。
如果使用多个低速执行型程序，则低速扫描时间为执行所有程序所需时间的总和，加上低速 END 处理时间。
当中断程序/固定循环执行型程序执行时，其值加上中断程序/固定循环执行型程序的执行时间，为低速扫描时间。

(b) 低速扫描时间由 QCPU 测量，其结果保存在特殊寄存器（SD528—SD535 中）。*1
因此通过监控 SD528—SD535 的特殊寄存器，可检查低速扫描时间。



如果 SD528 之值为 50，SD529 之值为 400，则低速扫描时间为 50.4ms。

要点
*1: 保存在特殊寄存器里的扫描时间精度为±0.1ms。 在顺控程序中，即使执行警戒定时器（WDT）的复位指令，扫描时间计数仍继续进行。

(7) 低速执行监控时间

低速执行型程序时间可由定时器监控。不设定缺省值。

在监控低速执行型程序的执行时间时，在 PLC 参数的 PLC RAS 设定值中指定低速执行监控时间范围为 10ms—2000ms。（设定单位 ms）

如果低速执行型程序的执行时间超过定时器的设定值，即出现“PRG TIME OVER”的错误。

要点
在低速 END 处理时出现低速执行时间的测量。 因此如果低速执行监控时间（t）指定为 100ms，而在低速 END 处理时测量的低速扫描时间超过 100ms，将产生“PRG TIME OVER”的错误。

4 - 24

4.2.4 待机型程序

(1) 待机型程序的定义

(a) 待机型是指只有在需要时才执行的程序。

(b) 待机型程序的应用如下。

1) 将程序放于程序库中

子例行程序和中断程序可转换成待机型程序，可独立于主程序进行管理。

2) 改变程序的建立

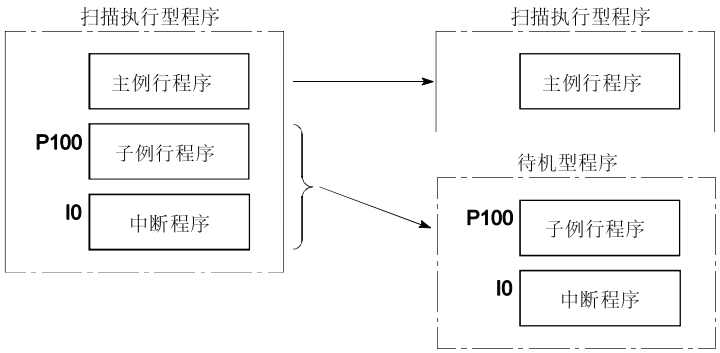
主例行程序也可作为待机程序寄存，然后按需要转换成扫描执行程序执行。不需要的程序则转换成待机型程序。

(2) 待机型程序的应用

(a) 将程序放于程序库中

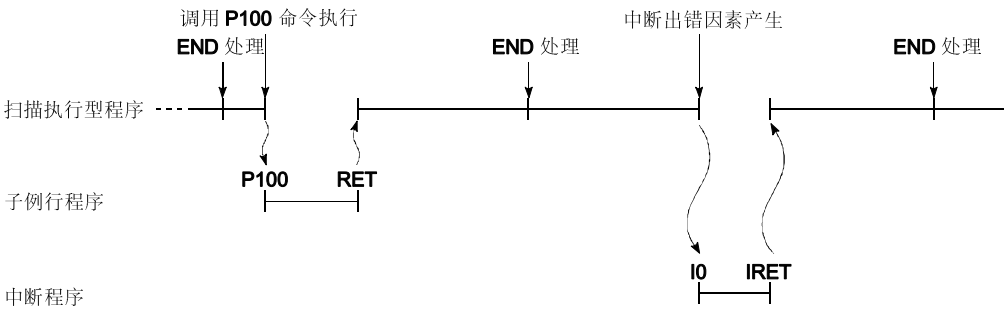
1) 该应用用以管理独立于主例行程序的子例行程序和中断程序。

多个子例行程序和中断程序也可创建为一个单独待机程序。



2) 当待机型程序执行完成后，处理又返回到在待机型程序执行前正在运行的程序中。

下图表示当待机型程序的子程序的子例行程序和中断程序执行时发生的操作。



4 - 25

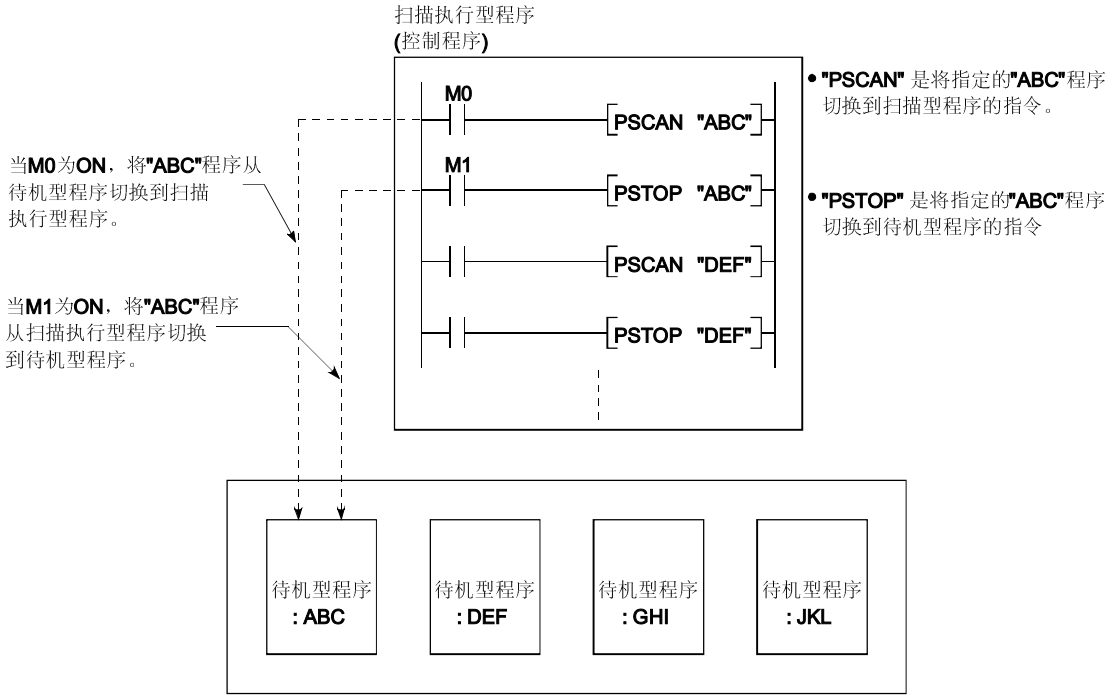
(b) 改变程序的建立

1) 创建一个与所有程序兼容的程序，仅在执行必须的程序时才用此程序。
由 **PLC** 参数设定指定为待机型程序的那些程序可以切换为扫描执行型程序并在顺控程序中执行。
用 **PSCAN**，**PLOW**，**PSTOP**，和 **POFF** 指令可通过 **QCPU** 改变执行类型。（参见 **4.2.3** 节）

(c) 下列方法可以用来切换执行的程序。

1) 从单独管理的程序中选择执行的程序：

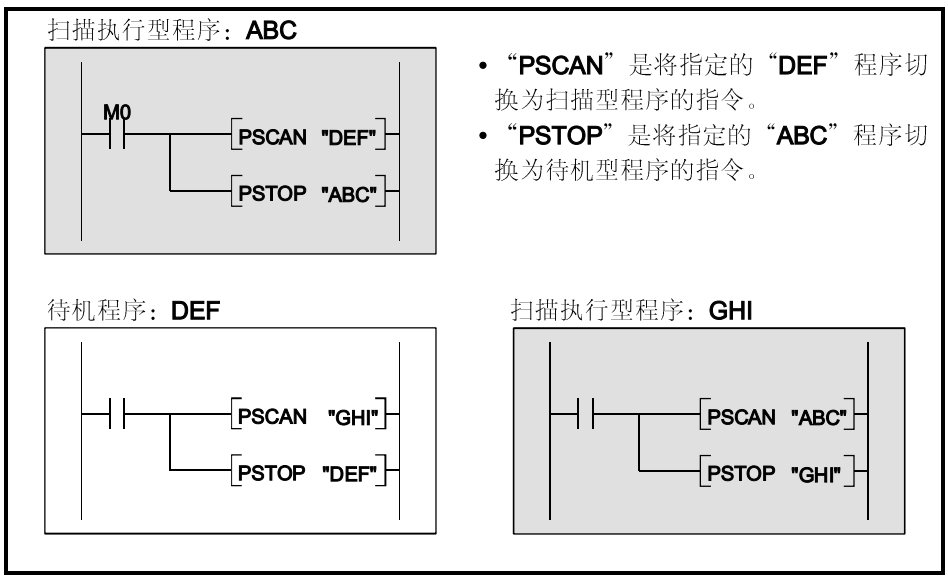
- 使用一个恒定执行的扫描执行型程序作为管理程序，一个与指定条件相一致的待机型程序就能切换成扫描执行型程序并被执行。
  不需要的扫描执行型程序就能切换成待机型程序。
- 下图为在一个单独管理的程序中当“**ABC**”，“**DEF**”，“**GHI**”，和“**JKL**”待机型程序执行类型切换时产生的操作。

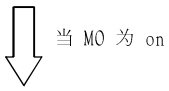


4 - 26

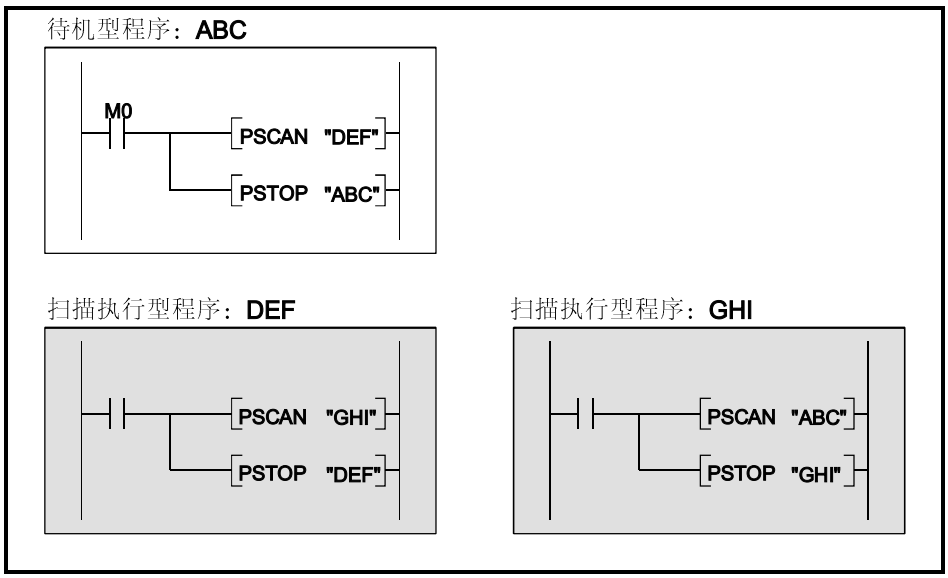
2) 当把一个正在执行的扫描执行型程序改变为另一个程序的执行类型。
- 对正在执行的扫描执行型程序而言，下一个执行的程序是把待机型程序切换成扫描执行型程序并被执行。
- 将“ABC”和“GHI”程序设置为扫描型执行型程序，把“DEF”设置为待机型程序。下图为当条件满足时，“ABC”和“DEF”程序的执行类型切换时产生的操作。

[在执行 PSCAN 和 PSTOP 指令前]



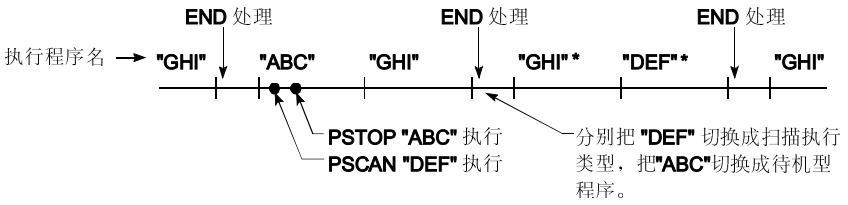


[在执行 PSCAN 和 PSTOP 指令后]



4 - 27

(d) 在 END 处理时进行程序执行类型的切换。
程序正在执行时，程序类型不能切换。
在相同的扫描时如把不同的执行类型指定为相同的程序，那么在执行切换指令后最后指定的执行类型有效。



备注

1)*: GHI 和 DEF 程序执行的顺序由 PLC 参数的程序决定。

4 - 28

(3) 创建待机型程序时的安全要求

(a) 在执行 **OUT T** □ 指令时，当前值更新以及接触器 **ON/OFF** 切换，因此定时器不得用于待机型程序。

(b) 将子例行程序集合为一个单独的程序

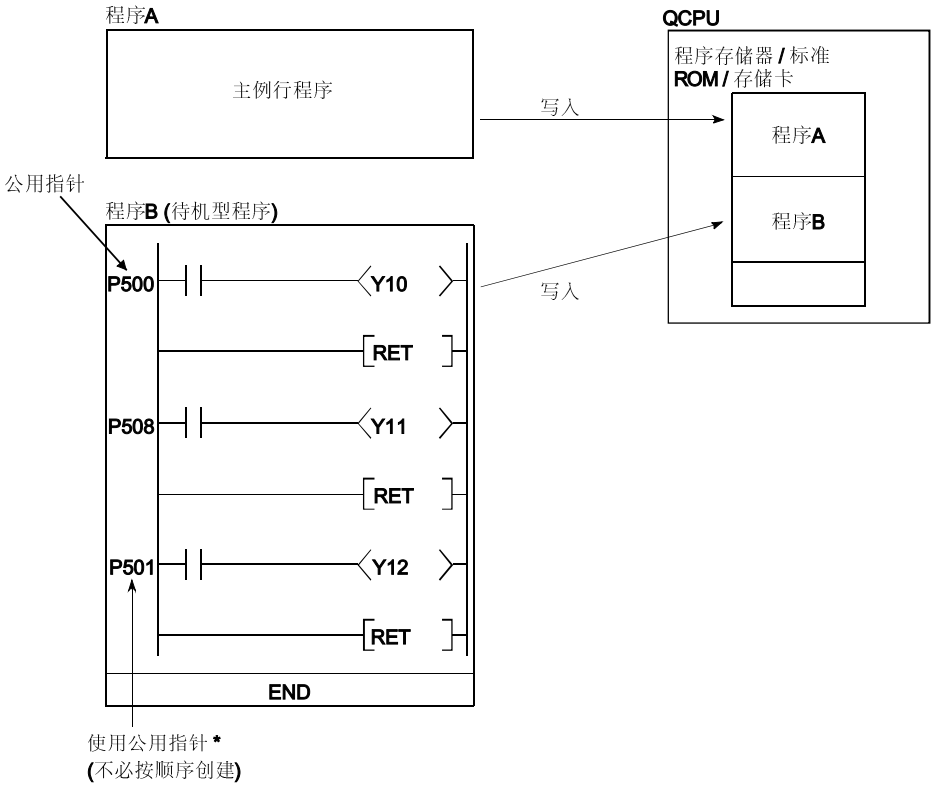
1) 按顺序从待机型程序的 **0** 步开始创建子例行程序。在子例行程序的结束处需要一个 **END** 指令。

2) 因为子例行程序的创建顺序没有限制要求，因此在创建多个子例行程序时指针编号就不需指定升序。

3) 使用公用指针*

用公用指针的子例行程序可以从所有被 **QCPU** 执行的程序中调用。

（如果使用本地指针，将不能执行待机型程序的子例行程序。）



4) 有关包括本地软元件的子例行程序的执行，参见第 **10.13.1** 节。

备注

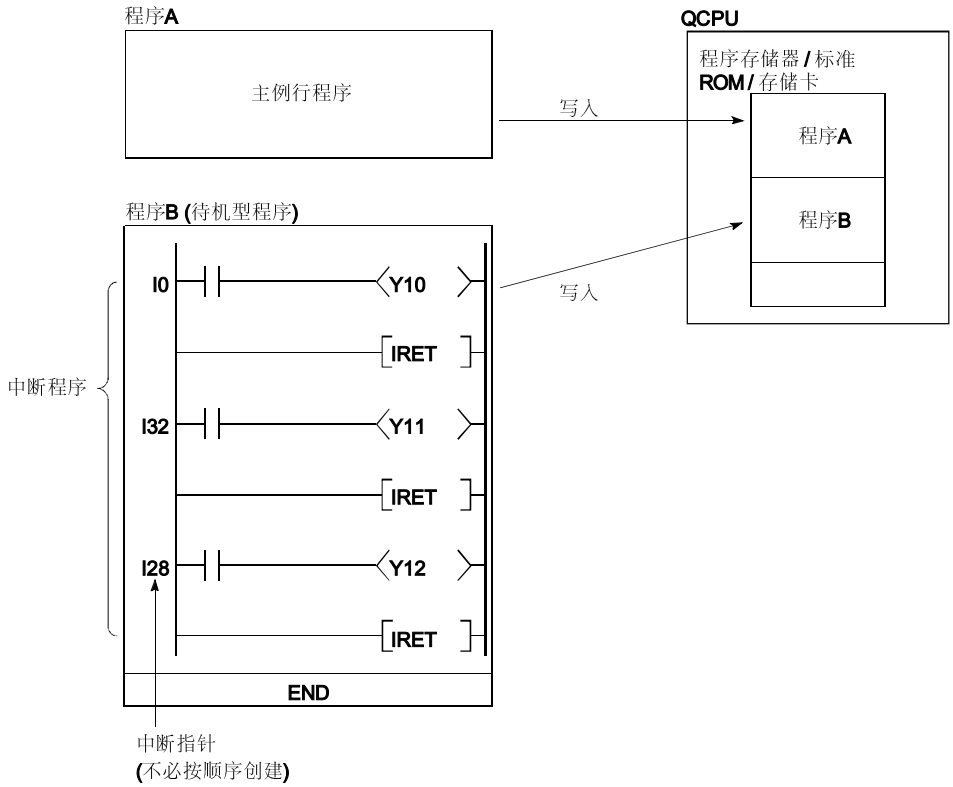
***:** 有关公用指针和本地指针详见第 **10.9** 节。

4 - 29

(c) 将中断程序集合为一个单独的程序

1) 按顺序从待机型程序的 0 步开始创建中断程序。
在中断程序的结束处需要一个 END 指令。

2) 因为中断程序的创建顺序没有限制要求，因此在创建多个中断程序时指针编号就不需指定升序。



备注

有关中断指针详见第 10.10 节。

4 - 30

4.2.5 固定扫描执行型程序

(1) 固定扫描执行型程序的定义

(a) 固定扫描执行型程序是在指定间隔期内执行的程序。
没有说明中断点和 IRET 指令，对每个文件都可进行固定扫描。

(b) 通过 PLC 参数的程序设定将执行类型设定为“固定扫描”。

(2) 固定扫描执行型程序的执行

(a) 当存在多个固定扫描执行型程序时，在一个指定的时间内执行一个固定扫描执行型程序。
如果在指定的时间内已经有多个扫描执行型程序按同一时序执行，那么这些程序是按 PLC 参数的程序设置以编号的升序执行的。

(b) 当一个指定的时间内同时有一个固定扫描执行型程序和中断程序（128-131）要执行，执行的优先权给中断程序。

(c) 在网络刷新期间的执行
在网络刷新期间，建立固定扫描执行型程序的执行条件，则网络刷新被暂停，执行中断程序。
相应地，即使由 MELESECNET/H 网络系统给予“每个站循环数据的块保证”，如果在站内设置的软元件被刷新而且用于中断程序，即不能得到这一保证。*1

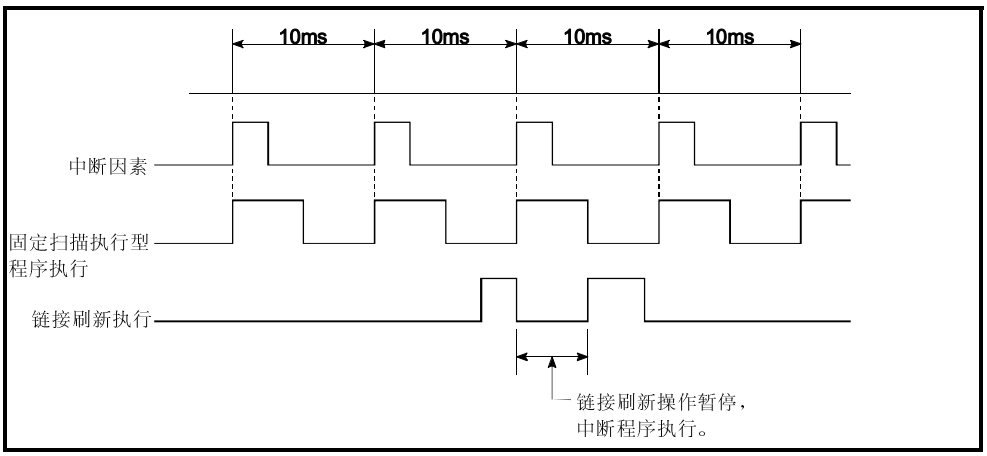


图 4.5 在网络刷新期间固定扫描执行型程序的执行

备注

*1: 有关每个站循环数据的块保证详见下列手册。
• 具 Q 功能的 MELSECNET/H 网络系统参考手册。

4.3 操作处理

4.3.1 初始处理

这是顺序操作执行的预处理，并只执行一次。如下表所列。
预处理完成后，QCPU 进入 RUN/STOP 开关设置状态。（参见第 4.4 节）

初始处理项目	QCPU 状态		
	接通电源	执行复位	从STOP切换至RUN*1
I/O模块初始化	执行	执行	不执行
标准ROM/存储卡引导	执行	执行	执行
锁存范围的软元件初始化（位软元件：OFF，字软元件：ON）	执行	执行	执行
QCPUR的自诊断执行	执行	执行	不执行
安装模块的I/O地址自动分配	执行	执行	执行
MELSECNET/H网络信息设置和网络通信的启动	执行	执行	执行*2
软元件初始化值的设置	不执行	不执行	执行

备注

1)*1: 当程序或参数在 STOP 状态下更改，则用 RESET/L.CLR 开关复位。
当 RUN/STOP 开关从 STOP 切换至 RUN 而不复位时，RUN LED 闪烁。
当 RUN/STOP 开关从 RUN 切换至 STOP 再至 RUN 时，QCPU 进入 RUN 状态，并且“从 STOP 至 RUN”的状态有效。

2)*2: 当程序或参数在 STOP 状态下改变时执行。（当参数改变时，QCPU 使用改变后的参数。）
当程序或参数在其它状态下改变时不执行。

4.3.2 I/O 刷新（I/O 模块刷新处理）

在 I/O 刷新时，从输入模块/智能型功能模块中接收输入（X），而 QCPU 的输出（Y）送至输出模块/智能型功能模块。

在顺控程序开始运行之前执行 I/O 刷新。

4.3.3 智能型功能模块的自动刷新

设置智能型功能模块的自动刷新时，执行与指定数据的智能型功能模块的通信。
有关智能性功能模块的自动刷新设定，详见智能型功能模块使用手册。

4.3.4 END 处理

这是在完成一次整个顺控程序的操作处理后，将顺控程序的执行返回到 0 步的后处理。

(a) 当从网络模块得到刷新处理要求时，执行刷新处理。
(b) 当采样跟踪的跟踪点在每次扫描时（END 指令执行后），设置的软元件状态保存在采样跟踪区域。

要点
(1) 当设置恒定扫描功能时（参见第 7.2 节），在 END 处理之后到下一次扫描的时间间隔内保存 END 处理时间结果 (2) 执行低速执行型程序时，低速 END 处理在全部低速执行型程序完成之后开始。有关低速执行型程序和低速 END 处理，详见第 4.2.3 节。

4.4 RUN，STOP，PAUSE（运行，停止，暂停）三种操作处理

QCPU 有三种操作状态：RUN，STOP 和 PAUSE 状态。
以下说明 QCPU 的操作处理。

(1) RUN（运行）状态操作处理

(a) RUN 状态为顺控程序从 0 步到 END（FEND）指令再重复到 0 步的运行状态。

(b) 当进入 RUN 状态时，输出状态通过在 STOP→RUN 时参数输出模式的设置而在 STOP 时作保存。

(c) 处理时间从 STOP 切换至 RUN，直至随系统配置顺控程序操作开始改变，通常为 1—3 秒。
但是，根据具体条件，此时间可能会稍长。

(2) STOP（停止）状态操作处理

(a) STOP 状态为用 RUN/STOP 键开关，或执行远程 STOP 时顺控程序操作停止的状态。（关于远程 STOP 功能，详见第 7.6.1 节）

(b) 当进入 STOP 状态时，保存所有的输出状态并切断所有的输出。
除了输出（Y）外，软元件存储器保存所有的输出。

(3) PAUSE（暂停）状态操作处理

(a) PAUSE 状态为通过远程 PAUSE 功能顺控程序的操作暂停，保存输出和软元件存储的状态。（关于远程 PAUSE 功能详见第 7.6.2 节）

(4) 在 RUN/STOP 状态下的 QCPU 操作处理

QCPU 操作处理 / RUN/STOP 状态	顺控程序操作处理	外部输出	软元件存储器 (Y, M, L, S, T, C, D)
从 RUN→STOP	执行至 END 指令并停止。	OS 保持输出状态，所有输出切断。	保存 STOP 状态前的瞬间状态。
从 STOP→RUN	从 0 步开始。	在 STOP→RUN 时，由 PLC 参数的输出模式决定。	从 STOP 状态前的瞬间状态开始执行操作。但如果指定软元件的初始值，则该值已经设定。清除本地软元件。

要点

QCPU 在 RUN，STOP 和 PAUSE 状态下都可执行如下处理：
- I/O 模块的刷新处理
- 用 GX Developer 和串行通信模块进行数据通信
- 网络刷新处理

因此在 STOP 和 PAUSE 状态时，用 GX Developer 的 I/O 监控和试验操作，以及用 MELSECNET/H 与其它站的通信和从串行通信进行的读/写等都可执行。

4.5 瞬时掉电时的操作处理

当输出电源电压低于正常范围时，QCPU 就探测到电源模块的瞬时掉电。
当 QCPU 探测到瞬时掉电时，执行下列操作处理：

(1) 当发生掉电时间小于允许掉电时间的瞬时掉电时

(a) 当瞬时掉电发生时，保持输出存取文件的文件名以及出错记录。然后系统中断操作处理。（定时器的时钟继续工作。）

(b) 如果有 SFC 连续指定时，则执行系统保存处理。

(c) 当瞬时掉电结束，操作处理恢复。

(d) 即使由于瞬时掉电操作中断，警戒定时器（WDT）和测量仍继续。例如，如果 GX Developer PLC 参数模式的 WDT 设定值设定为 200ms，在扫描时间为 190ms 时发生 15ms 的瞬时掉电，设置警戒定时器出错。

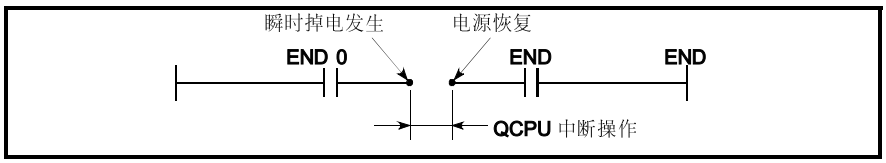


图 4.6　发生瞬时掉电时的操作处理

(2) 瞬时掉电超过允许掉电时间时

QCPU 开始启动。（QCPU 电源接通。）
当接通电源或用 RESET/L.CLR 开关进行复位操作时，执行复位操作，也进行相同的操作处理。

4.6 数据清除处理

(1) 数据清除

当 QCPU 用 RESET/L.CLR 开关进行复位操作时，或电源由 ON→OFF→ON 时，除了下列各项之外，将清除所有的数据。

(a) 程序存储数据（除了在引导规格中设置“清除程序存储”外）
(b) 在存储卡中的数据
(c) 锁存规格（存锁清除有效）中的软元件数据
(d) 锁存规格（存锁清除无效）中的软元件数据
(e) 文件寄存器数据
(f) 故障历史记录数据（当规定用特殊寄存器 SD 存储时）

当使用 RESET/L.CLR 开关锁存清除操作，或用 GX Developer 功能进行远程锁存清除时，清除（c）项的数据。

(2) 软元件的锁存规定

(a) 对每个软元件在 PLC 参数的软元件设置中规定软元件锁存（锁存范围设置）。

有两类锁存范围设置：

1) 有效锁存清除键
设置用 RESET/L.CLR 开关进行锁存清除操作和远程锁存清除的锁存范围。

2) 无效锁存清除键
设置用 RESET/L.CLR 开关进行锁存清除操作和远程锁存清除的锁存范围。

(b) 设置为无效 RESET/L.CLR 开关的软元件只能用指令和 GX Developer 清除操作时被清除。

1) 指令清除法
用 RST 指令复位，或用 MOV/FMOV 传送 K0 指令。

2) GX Developer 清除法
以在线的 PLC 存储清除法（包括锁存）清除所有软元件的存储器。
有关 GX Developer 操作方法，详见 GX Developer 操作手册。

要点

清除文件寄存器或本地软元件，可使用 RST 指令执行复位操作，或使用 MOV/FMOV 指令来传送 K0。

备注

关于 MOV/FMOV 指令，参见下列手册：
• QCPU（Q 模式）/QnACPU 编程手册（公共指令）

4.7 输入/输出处理和响应滞后

QCPU 具有刷新型输入/输出处理格式的特点，在 END 处理时用这种格式进行与输入/输出模块的批量通信。
在顺控程序中使用的直接访问输入/输出法也可成为直接的通信格式，当执行顺控顺控指令时，用输入/输出模块可进行直接通信。
有关直接输入和直接输出，分别详见第 10.2.1 节和 10.2.2 节。

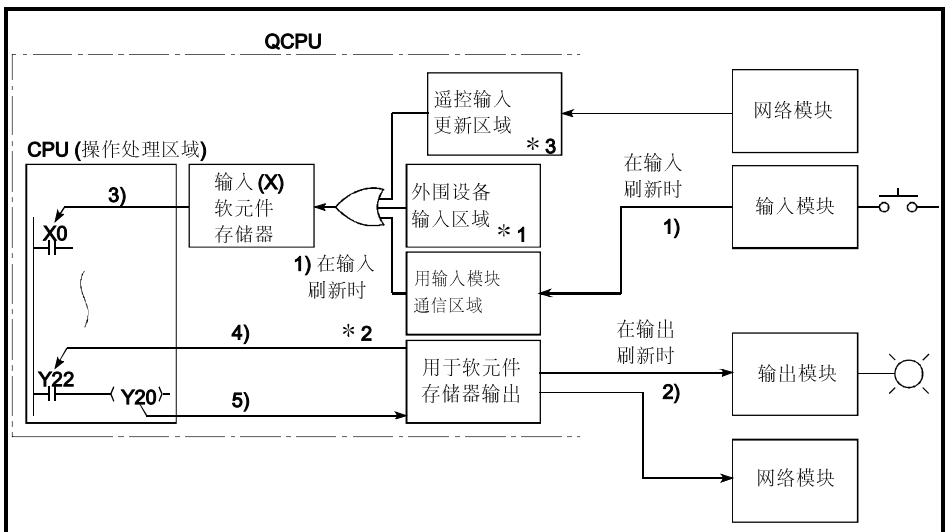
4.7.1 刷新模式

(1) 刷新模式的定义

使用刷新模式，在 END 处理时可用输入/输出模块进行批量通信。

(a) 当进行 END 处理时，在 QCPU 内置输入软元件存储器中成批读入执行输入模块的 ON/OFF 信息。当执行顺控程序时，这些 ON/OFF 数据（在输入软元件存储器里）就能用来进行处理。

(b) 输出（Y）顺控程序的处理结果输出到 QCPU 的内置输出软元件的存储器中，而且将 ON/OFF 的数据（在输出软元件的存储器中）成批输出到输出模块中，在 END 处理时执行。



- 输入刷新：
  输入信息批 1）在 END 处理时，从输出模块读出，并通过外围设备的输入区域的 OR 操作保存在输入(X)软元件存储器中。
- 输出刷新：
  在输出（Y）软元件存储器中的数据在 END 处理时以批 2 的形式输出到输出模块中。
- 当执行输入接触器指令完成时：
  输入信息从输入（X）软元件存储器中读出 3），执行顺控程序。
- 当执行输出接触器指令完成时：
  输出信息从输出（Y）软元件存储器中读出 4），执行顺控程序。
- 当执行输出 OUT 指令完成时：
  顺控程序的操作结果 5）被保存在输出（Y）软元件存储器中。

图 4.7　刷新模式时的输入/输出信息流

备注

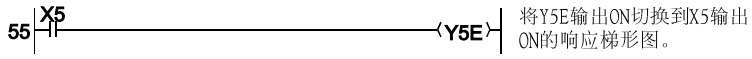
1)*1: 外围设备输入区域可通过下列方法进行 ON/OFF 切换：
- 用 GX Developer 进行试验操作
- 用 MELSECNET/H 网络系统进行网络刷新
- 从串行通信模块上写入
- CC–Link 自动刷新

2)*2: 输出（Y）软元件存储器可通过下列方法进行 ON/OFF 切换：
- 用 GX Developer 进行试验操作
- 用 MELSECNET/H 网络系统进行网络刷新
- 从串行通信模块上写入
- CC–Link 自动刷新

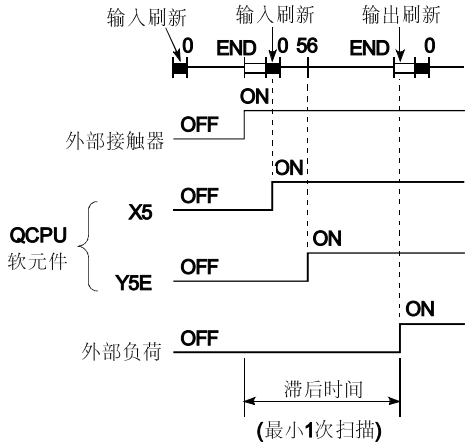
(2) 响应滞后

因为输出模块的变化，输出响应滞后可达 **2** 次扫描。（见图 **4.8**）

梯形图举例

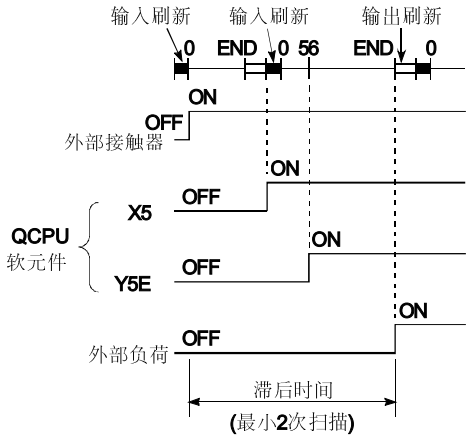


最快 **Y5E ON**



如果外部接触器在刷新操作之前的瞬间切换至 **ON**，就可能产生最快的 **Y5E ON**。然后 **X5** 在输入刷新时切换至 **ON**，随着 **END** 指令的执行，**Y5E** 在 **56** 步时切换至 **ON**，在输出刷新时，外部负荷切换至 **ON**。在这种情况下，外部接触器 **ON** 和外部负荷 **ON** 之间的时间滞后为 **1** 次扫描。

最慢的 **Y5E ON**



如果外部接触器在刷新操作之前的瞬间切换至 **ON** 的刷新操作，就可能产生最慢的 **Y5E ON**。然后 **X5** 在输入刷新时切换至 **ON**，随着 **END** 指令的执行，**Y5E** 在 **56** 步时切换至 **ON**，在输出刷新时，外部负荷切换至 **ON**。在这种情况下，外部接触器 **ON** 和外部负荷 **ON** 之间的时间滞后为 **2** 次扫描。

图 **4.8** 输出变化对输入变化的响应

4 - 40

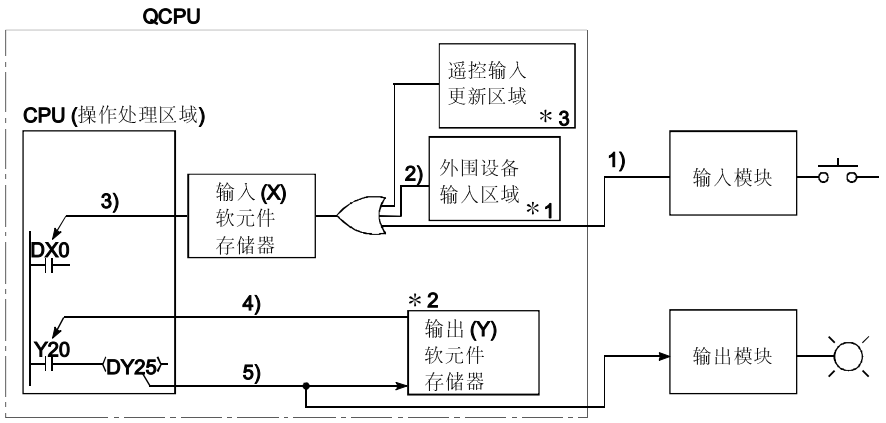
4.7.2 直接模式

(1) 直接模式的定义

当执行顺控顺控指令时，用直接模式执行与输入/输出模块的通信。

对于 QCPU，直接模式 I/O 处理可使用直接访问输入（DX）和直接访问输出（DY）的方法执行。

有关直接访问输入，参见第 10.2.1 节。直接访问输出，参见第 10.2.2 节。



- 当执行输入接触器指令完成时：
  对输入模块的输入信息 1）及外围设备输入区域的输入信息 2）执行 OR 操作，结果保存在输入（X）软元件存储器中。然后这一数据在执行顺控程序时当作输入信息 3）使用。
- 当执行输出接触器指令完成时：
  输出信息 4）从输出软元件存储器中读出，并执行顺控程序。
- 当执行输出 OUT 指令完成时：
  顺控程序的操作结果 5）输出至输出模块，保存在输出（Y）软元件存储器中。

图 4.9　直接模式时的输入/输出信息流

备注

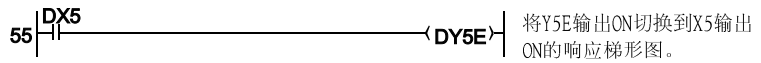
*1: 外围设备输入区域可通过下列方法进行 ON/OFF 切换：
- 用 GX Developer 进行试验操作
- 用 MELSECNET/H 网络系统进行网络刷新
- 从串行通信模块上写入
- CC–Link 自动刷新

*2: 输出（Y）软元件存储器可通过下列方法进行 ON/OFF 切换：
- 用 GX Developer 进行试验操作
- 用 MELSECNET/H 网络系统进行网络刷新
- 从串行通信模块上写入
- CC–Link 自动刷新

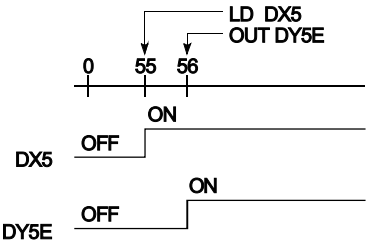
(2) 响应滞后

因为输出模块的变化，输出响应滞后可达 1 次扫描。(见图 4.10)

梯形图举例

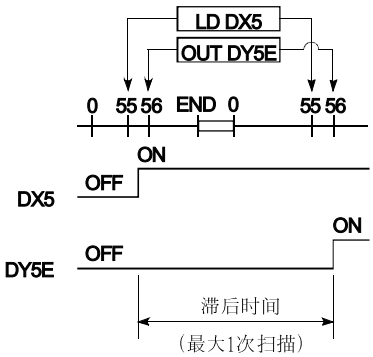


最快 Y5E ON



如果 DX5 输入在 55 步操作之前的瞬间切换至 ON，就可能产生最快的 DY5E 输出 ON。如果当第 55 步时执行 LD DX5，DX5 ON，在此扫描时 DY5E 切换至 ON。
该状态表示在 DX5 输入 ON 和 DY5 输出 ON 之间的最小时间滞后。

最慢的 DY5E ON



如果 DX5 输入在第 55 步之后的瞬间切换至 ON，就可能产生最慢的 DY5E ON 输出 ON。在这种情况下，在下一次扫描中，DY5E 输出将切换至 ON。
该状态表示在 DX5 输入 ON 和 DY5 输出 ON 之间的最大时间滞后（1 次扫描）。

图 4.10 输出“Y”变化对输入“X”变化的响应

4 - 42

4.8 在顺控程序中能使用的数值

在 QCPU 中用“0”（OFF）和“1”（ON）来表示数字和字母数据。
这一方法称为“二进制码”（BIN）。
十六进制（HEX）表示法中二进制数据以 4 位为单位来表达，而二进制编码的十进制（BCD）表示法也能用于 QCPU。也可用实数。（参见第 4.8.4 节）
BIN、HEX、BCD 的数字表达形式和十进制 (DEC)的表达形式如下表 4.1 所示。

表 4.1BIN，HEX，BCD 和十进制数字表达法

DEC（十进制）	HEX（十六进制）	BIN（二进制）	BCD（二进制编码的十进制）
0	0	0	0
1	1	1	1
2	2	10	10
3	3	11	11
•	•	•	•
•	•	•	•
•	•	•	•
9	9	1001	1001
10	A	1010	1 0000
11	B	1011	1 0001
12	C	1100	1 0010
13	D	1101	1 0011
14	E	1110	1 0100
15	F	1111	1 0101
16	10	1 0000	1 0110
17	11	1 0001	1 0111
•	•	•	•
•	•	•	•
•	•	•	•
47	2F	10 1111	100 0111
•	•		
•	•		
•	•		
32766	7FFE	0111 1111 1111 1110	——
32767	7FFF	0111 1111 1111 1111	——
-32768	8000	1000 0000 0000 0000	1000 0000 0000 0000
-32767	8001	1000 0000 0000 0001	1000 0000 0000 0001
•	•		
•	•		
•	•		
-2	FFFE	1111 1111 1111 1110	——
-1	FFFF	1111 1111 1111 1111	——

4 - 43

(1) 外部数字对 QCPU 输入

从外部数字源（如数字开关等）对 **QCPU** 指定数字设置。与十进制设置相同，也可指定 **BCD**（二进制编码的十进制）的设置。

但是，因为 **QCPU** 的操作基础是 **BIN**，如果 **QCPU** 使用的数值用 **BCD** 的方法指定，则 **QCPU** 会把该值当作 **BIN** 的数，在这样的数值基础上操作将与用指定的数值规定的操作不同。

因此给 **QCPU** 提供的 **BIN** 指令可以将 **BCD** 输入数据转换成 **QCPU** 使用的 **BIN** 数据。

为了不考虑外部数字源的数字是否与 **BIN** 值相应，而允许对其指定的数字进行设定，可在顺控程序上编制一个将数字数据变换为二进制数据的程序。

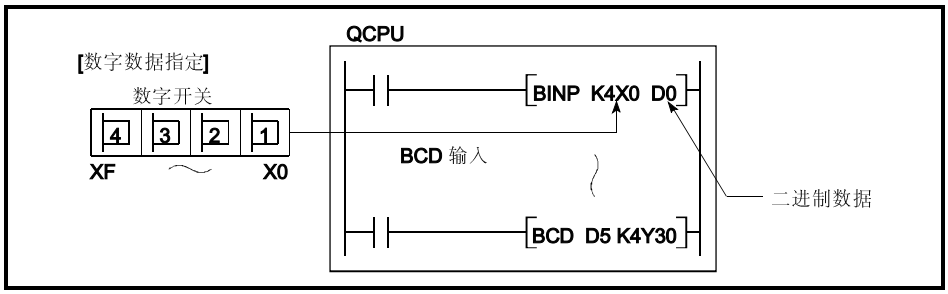


图 **4.11** 数字开关数据对 **QCPU** 的输入

(2) 从 QCPU 向外部的数字输出

可用数字来显示从 **QCPU** 向外部的数字数据输出。

但是，由于 **QCPU** 使用 **BIN** 数据，所以不能像普通数字显示那样显示。

因此，一个提供给 **QCPU** 的 **BCD** 指令将 **BIN** 数据转换成 **BCD** 数据。为了以与十进制数据相一致的方式来显示输出数据，可在顺控程序上编制一个将二进制数据变换为 **BCD** 数据的程序。

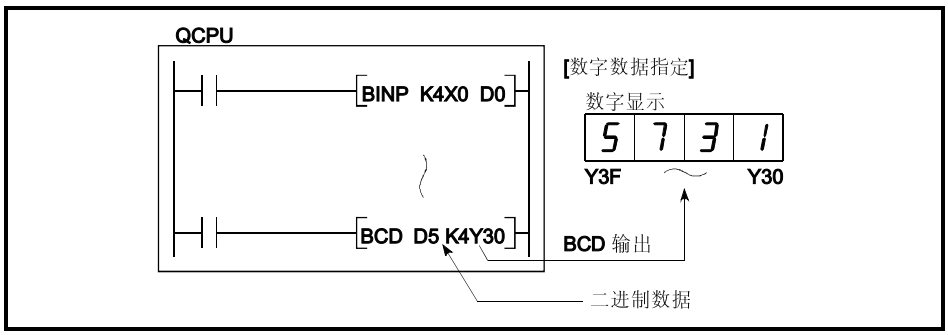


图 **4.12** **QCPU** 数据的数字显示

4 - 44

4.8.1 BIN(二进制编码)

(1) 二进制编码

在二进制编码中，数值用数字“**0**”（**OFF**）和数字“**1**”（**ON**）表示。
当在十进制系统中计数时，从 **9** 之后进位至“**10**”位。（**8** 到 **9** 到 **10**）。
在二进制系统中在 **1** 之后进位。因为二进制的“**10**”即代表十进制的“**2**”。
二进制值和其相应的十进制值如下表 **4.2** 所示。

下表 **4.2** 所列为二进制相对于十进制的数值

DEC（十进制）	BIN（二进制）	
0	0000	
1	0001	进位
2	0010	
3	0011	进位
4	0100	
5	0101	
6	0110	
7	0111	进位
8	1000	
9	1001	
10	1010	
11	1011	

(2) 二进制数字表达形式

(a) QCPU 寄存器（数据寄存器，通信寄存器，等等）由 **16** 位构成，寄存器每位都分配一个“**2**”的值。
最重要的位（最初位）用来识别“正”和“负”。
1) 当最重要的位是“**0**”：正
2) 当最重要的位是“**1**”：负
QCPU 寄存器的数字表达形式如下图 **4.13**。

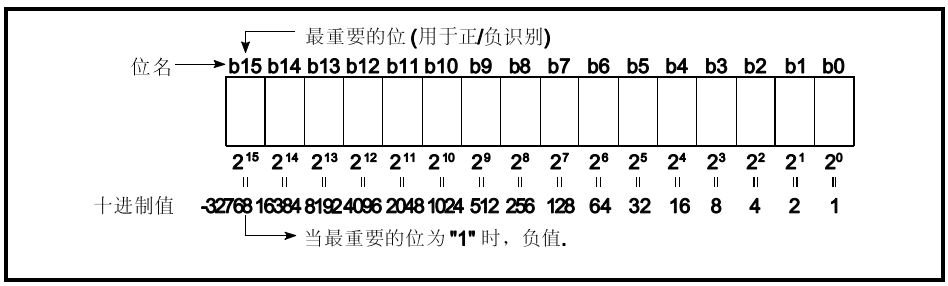


图 **4.13** **QCPU** 寄存器的数字形式

(b) 可用于 **QCPU** 的数字数据
如图 **4.11** 所示，数字表达形式范围从**-32768** 至 **32767**。因此，在该范围内的数字数据可保存在 **QCPU** 寄存器内。

4 - 45

4.8.2 HEX（十六进制）

(1) 十六进制形式

在十六进制形式中，二进制的 **4** 位用 **1** 个数字表示。二进制数据的 **4** 位可表示 **16** 个数值（**0**—**15**）

在十六进制形式中，数值 **0** 至 **15** 用 **1** 个数字表示。

从“**9**”之后用字母表示，“**F**”以后进位。

如下表 **4.3** 为二进制，十六进制和十进制数字表达形式的比较表。

表 **4.3 BIN**，**HEX** 和 **DEC** 数字表达形式比较

DEC（十进制）	HEX（十六进制）	BIN（二进制）
0	0	0
1	1	1
2	2	10
3	3	11
•	•	•
•	•	•
•	•	•
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	1 0000
17	11	1 0001
•	•	•
•	•	•
•	•	•
47	2F	10 1111

进位

(2) 十六进制的数字表达形式

QCPU 寄存器（数据寄存器，通信寄存器等，等等）由 **16** 位构成。

因此如果用十六进制编码表示，可保存的数值范围为 **0**—**FFFF**$_{H}$。

4 - 46

4.8.3 BCD（二进制编码十进制）

(1) BCD 形式

BCD 数字表达形式是进位格式与十进制系统相同的二进制表达形式。
正如在十六进制系统中，BCD 表达形式相当于 4 个二进制位，但 BCD 系统不用 A—F 的字母。
二进制、BCD 和十进制数字表达形式的比较如下图 4.4 所示。

表 4.4 BIN，BCD 和 DEC 数字表达形式比较

DEC（十进制）	HEX（十六进制）	BIN（二进制）
0	0	0
1	1	1
2	10	10
3	11	11
4	100	100
5	101	101
6	110	110
7	111	111
8	1000	1000
9	1001	1001
10	1010	1 0000
11	1011	1 0001
12	1100	1 0010

进位

(2) BCD 数字表达形式

QCPU 寄存器（数据寄存器，通信寄存器等，等等）由 16 位构成。
因此如果要用 BCD 编码表示，可保存的数值范围为 0—9999。

4 - 47

4.8.4 实数（十进制浮点数据）

(1) 实数

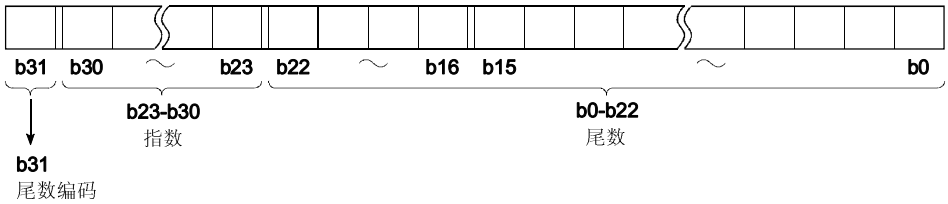
实数为单精度十进制浮点数据。

(2) 十进制浮点数据的内部表达

下面解释接收实数数据 QCPU 的内部表达。
实数数据表示如下，使用 2 个字软元件。

1. [尾数] × 2 (指数)

用于十进制浮点小数内部表达的位配置说明如下：



- 尾数编码： 尾数编码在 b31 上表示。
  0：正
  1：负
- 指数： “2n”中的“n”依 b23–b30 BIN 值的不同，在 b23–b30 中以不同的方式表示。

b23 to b30	FFH	FEH	FDH		81H	80H	7FH	7EH		02H	01H	00H
n	不用	127	126		2	1	0	-1		-125	-126	不用

- 尾数：对 1.××××××...的二进制数值来说，数值中的“××××××”部分用 b0–b22（23 位）来表示。

(3) 计算举例

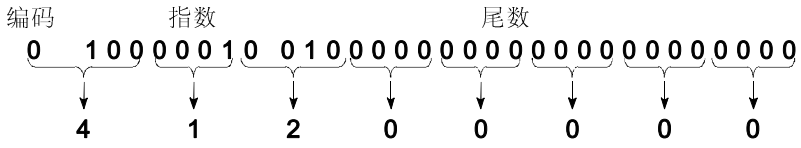
下列为计算举例（nnnnn“X”表示 X–系统数据表达）

(a) 保存“10”

$(10)_{10} \rightarrow (1010)_2 \rightarrow (1.01000....\times 2^3)_2$

尾数编码	正—0
指数	3 — 82H to $(10000010)_2$
尾数	$(010\ 00000\ 00000\ 00000\ 00000)_2$

因此数据表示为 41200000_H，如下所示。

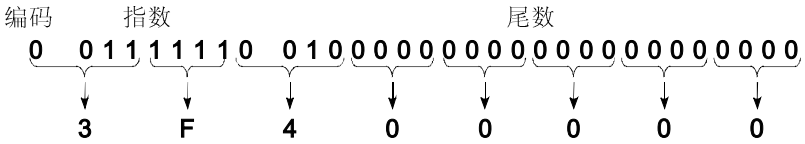


(b) 保存“0.75”

$(0.75)_{10} \rightarrow (0.11)_2 \rightarrow (1.100\ldots\times 2^{-1})_2$

尾数编码　　正—0
指数　　-1 至 $7E_H$ 至 $(01111110)_2$
尾数　　$(100\ 00000\ 00000\ 00000\ 00000)_2$

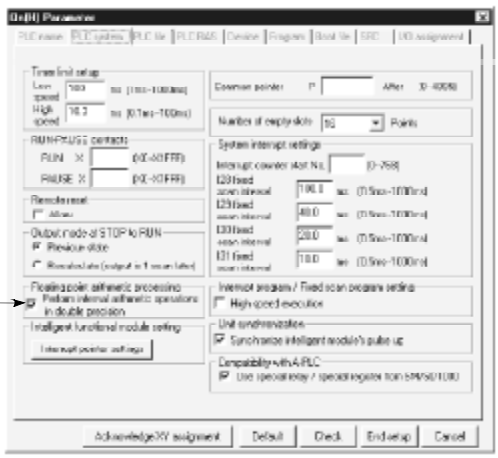
因此数据表示为 $3F400000_H$，如下所示。



要点

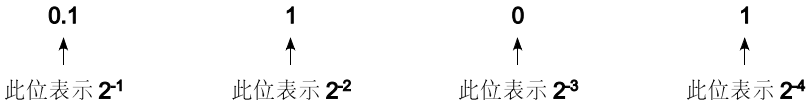
（1） GX Developer 的监控功能允许监控 QCPU 的实数数据。
（2） 对于“0”值，在所有的 b0—b31 位上表示。
（3） 在 PLC 参数的 PLC 系统设置中，既可以选择“以双精度进行内部算术运算”，也可以选择“不以双精度进行内部算术运算”。（其运算结果是短精度，与浮点算术处理设置无关。）
如果需要对实际算术运算提高速度，建议选择“不以双精度进行内部算术运算”。当需与常规设备具有兼容性时，建议选择“以双精度进行内部算术运算。”

- 当选择“以双精度进行内部算术运算时”（缺省设置），只能以双精度进行内部算术运算。（64 位）
  当设置为双精度时，内部算术运算使用实际算术运算，其指令的精度会提高，诸如 SIN 指令和 COS 指令。
- 当“不以双精度进行内部算术运算”设置时，由于内部算术运算以短精度（32 位）进行，故实算术运算进行得较快，但同时也会产生一定程度的精度丢失。



备注

在二进制系统中，小数点后的数值部分计算如下：



$(0.1101)_2 = 2^{-1} + 2^{-2} + 2^{-4} = 0.5 + 0.25 + 0.125 = (0.875)_{10}$

4 - 50

4.9 字符串数据

(1) 字符串数据

QCPU 使用 ASCII 编码数据。

(2) ASCII 编码字符串

ASCII 编码字符串如下表所示。

“00_H”（NUL 编码）用在字符串末尾。

b8	b7	b6	b5	b4	b3	b2	b1	Column / Low	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
									0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
									0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
									0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
									0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
				0	0	0	0	0	NUL		(SP)	0	@	P	`	p								
				0	0	0	1	1			!	1	A	Q	a	q								
				0	0	1	0	2			"	2	B	R	b	r								
				0	0	1	1	3			#	3	C	S	c	s								
				0	1	0	0	4			$	4	D	T	d	t								
				0	1	0	0	5			%	5	E	U	e	u								
				0	1	1	0	6			&	6	F	V	f	v								
				0	1	1	1	7			'	7	G	W	g	w								
				1	0	0	0	8			(	8	H	X	h	x								
				1	0	0	1	9			)	9	I	Y	i	y								
				1	0	1	0	A			*	:	J	Z	j	z								
				1	0	1	1	B			+	;	K	[	k	{								
				1	1	0	0	C			(Comma) ,	<	L	¥	l	|							
				1	1	0	1	D			(Minus) -	=	M	]	m	}								
				1	1	1	0	E			(Period) .	>	N	^	n	¯								
				1	1	1	1	F			/	?	O	Under line _	o									

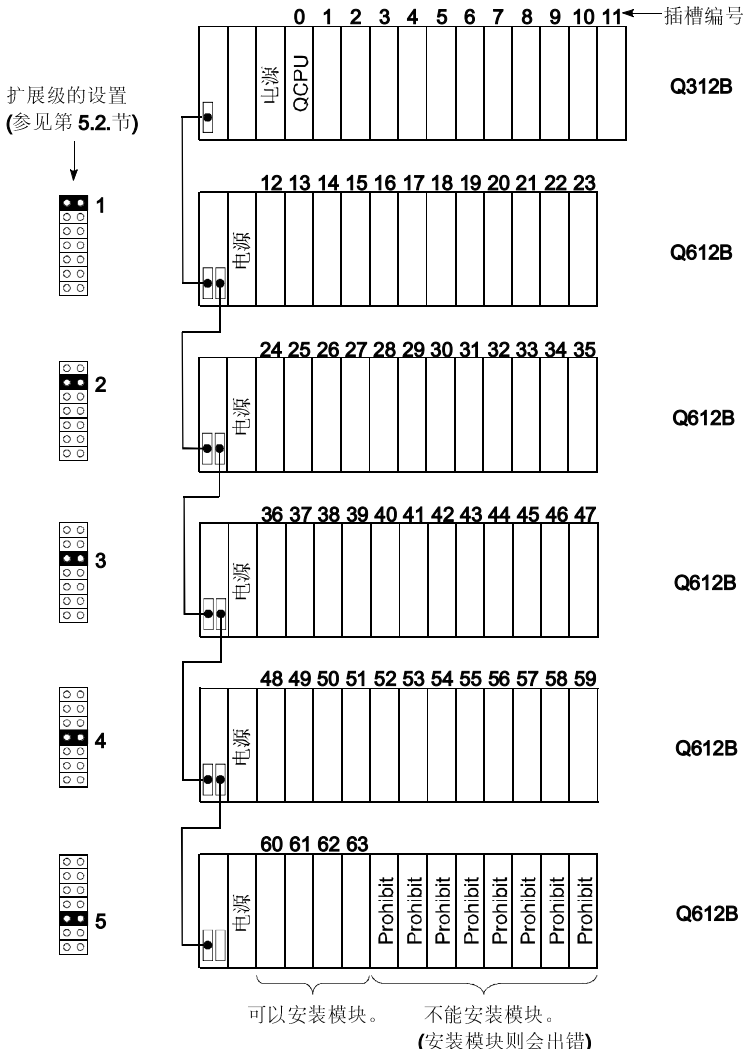
4 - 51

5. 分配 I/O 地址

本章阐述在 QCPU 和 I/O 模块或智能型功能模块之间交换数据时，关于分配 I/O 地址的必要说明。

5.1 扩展基板的级和插槽编号间的关系

QCPU 允许系统配置使用 8 块基板：一块主基板和七块扩展基板。
但是，可配置的插槽（模块）数（包括空闲的）限制为 64 个。
当模块（输入，输出或智能型功能模块）安装到第 65 个或再后面的插槽中时，将会出错。（SP.UNIT LAY ERR。）
模块的安装数不得超过 64 个插槽范围数。（只要安装的总模块数在 64 个插槽的范围内时，则不会出错。即使主基板和扩展基板的插槽总数多于 64 个[例如，当安装上 6 个 12–插槽基板时]）。



5 - 1

5.2 安装扩展基板和设定级数

共有两种扩展基板：Q6□B 用于安装 Q 系列模块，和 QA1S6□B 用于安装 AnS 系列模块。

(1) 扩展基板的连接顺序

当同时使用 Q6□B 和 QA1S6□B 两种扩展基板时，在靠近主基板处连接所有的 Q6□B 模块，然后再连接 QA1S6□B 模块。

(2) 设置扩展基板的扩展级编号

扩展基板要求设定扩展级编号（1—7）。
从与主基板连接的一块扩展基板开始编号，按 1—7 设置扩展段级号。

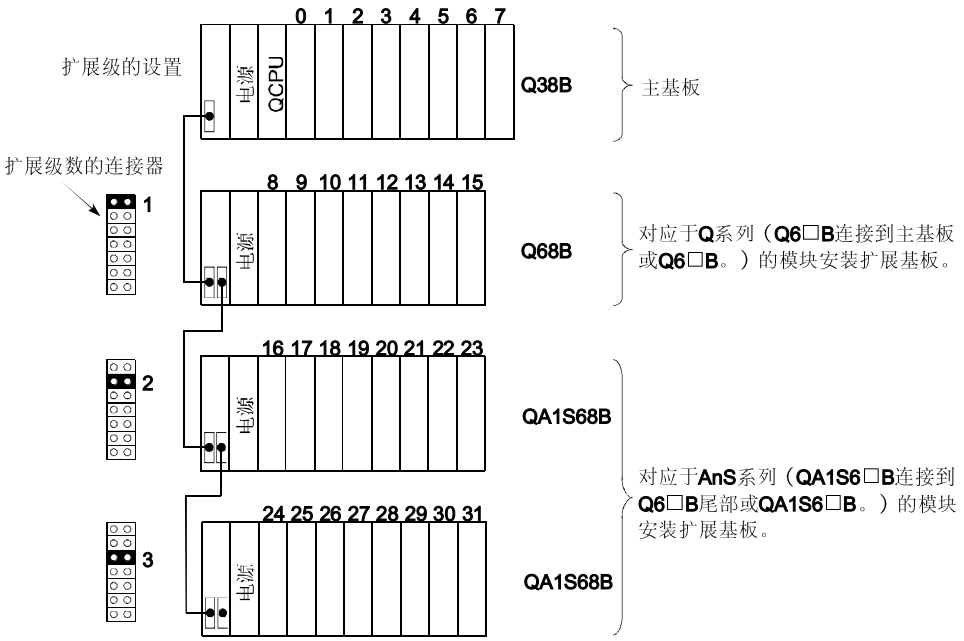
(3) 分配扩展基板的扩展级编号时的注意事项

(a) 对扩展级分配连续编号

如果用“自动”模式指定基板的级编号，则一些级编号分配给了不存在的模块，“0”作为插槽编号中的跳越级。其结果是空闲插槽并未增大。在 I/O 地址分配中，“0”也作为 I/O 点的跳越级。

(b) 不能对一个以上的扩展基板设置和使用相同的扩展级编号。

(c) 如果有一个以上的连接器引脚插入设置级数的连接器中，则不能使用该系统。反之，如果没有将连接器引脚插入到该设置级数的连接器中，则也不能使用该系统。



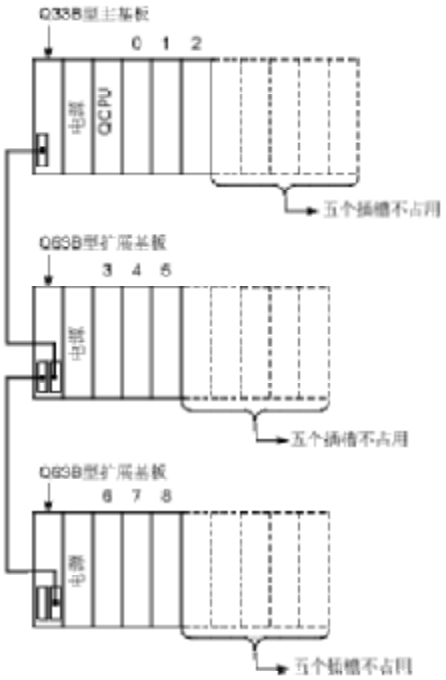
5.3 基板分配（基本模式）

分配 QCPU 主基板和扩展基板的模块编号的模式分为“自动”和“具体”两种模式。

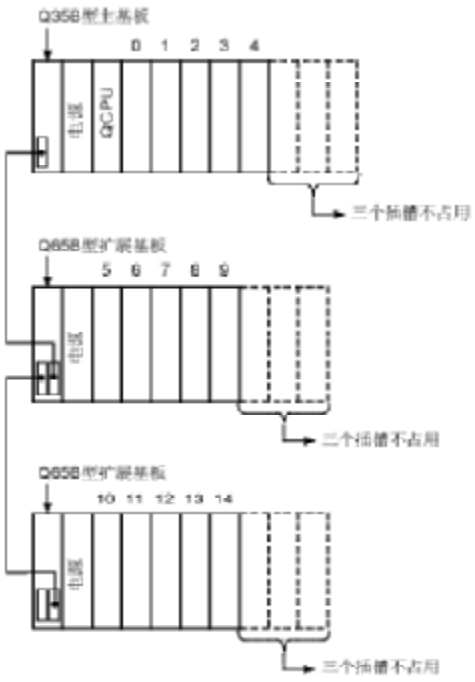
(1) 自动模式

在自动模式中，对应于所安装的主基板和扩展基板来分配基板的插槽编号。
对应于可以装入模块的当前基板来分配 I/O 地址。

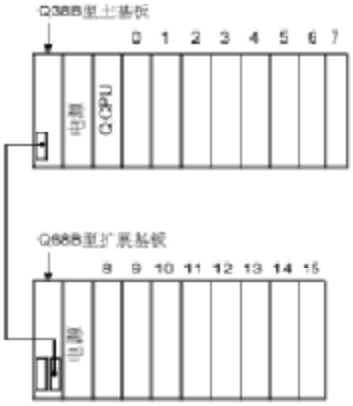
(a) 对于 3 个插槽的基板：占用 3 个插槽。



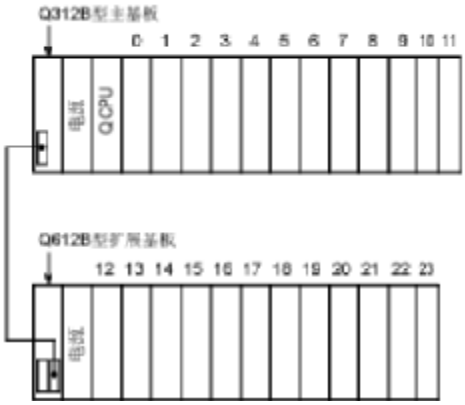
(b) 对于 5 个插槽的基板：占用 5 个插槽。



(c) 对于 8 个插槽的基板：占用 8 个插槽。



(d) 对 12 个插槽的基板：占用 12 个插槽。

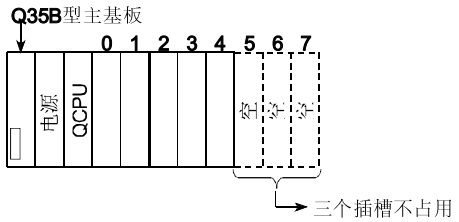


(2) 具体模式

(a) 在具体模式中，通过设置 **PLC** 参数的 **I/O** 分配将插槽编号分配给各个基板。
（主基板和扩展基板）
使用这一模式使插槽编号与 **AnS** 系列的基板的其中一种相符合（**8** 个定位）。

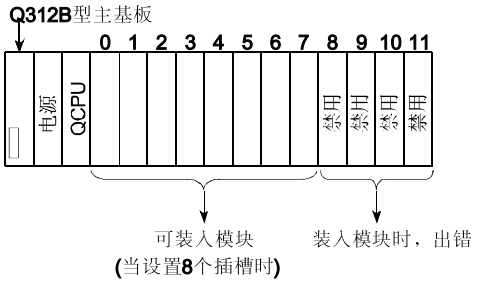
(b) 设置插槽编号的注意事项
设置插槽编号与当前使用模块的插槽编号无关。
但是必须为所有使用中的基板设置。
如果不为所有的基板设置插槽编号，就不能正确进行 **I/O** 分配。
如果当前设置的插槽编号与已安装的基板插槽编号不同，会产生如下结果。

1) 当分配的插槽编号大于已安装的基板插槽编号时：
在分配的插槽中，基板上除了被安装的基板占用的插槽之外，将会有空闲插槽出现。

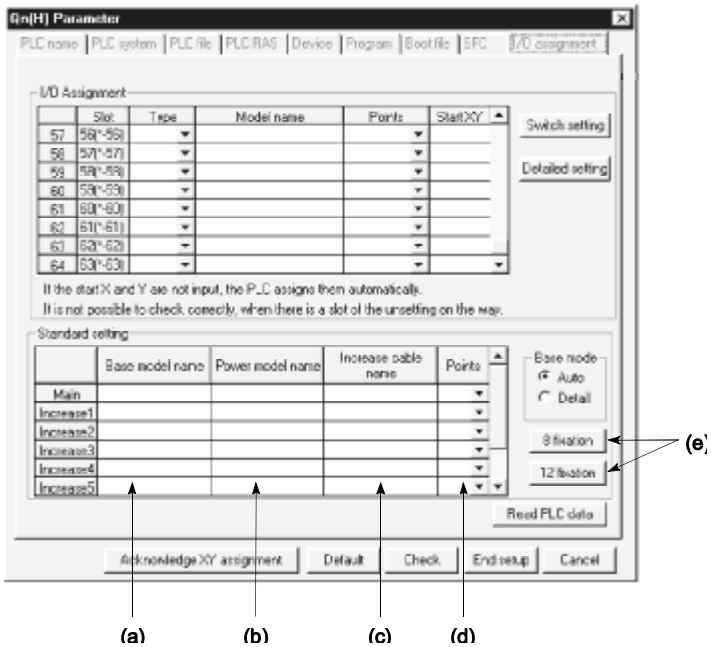


例如，当 **8** 个插槽分配用作 **5** 基板插槽时，有三个将会是空闲插槽。空闲插槽的点数由 **PLC** 参数的 **PLC** 系统指定或由 **I/O** 分配。（缺省值：**16** 点）

2) 当分配的插槽编号小于当前使用的基板插槽编号时：
除了已分配的，其它插槽无效。
例如，当 **12** 个插槽的基板分配了 **5** 个插槽，则基板右方 **4** 个插槽无效。
（如果模块装入无效插槽中，将会出错**[SP，UNIT LAY ERR.]**。）



(3) 设置 GX Developer 屏幕和设置基本模式项目



(a) 基本型号名称

用 **16** 个或更少的字母来指定安装的基板的型号名称。**QCPU** 不使用指定的型号名称。（可用作为用户备忘录）

(b) 电源型号名称

用 **16** 个或更少的字母来指定安装的电源模块的型号名称。**QCPU** 不使用指定的型号模式名称。（可用作为用户备忘录）

(c) 扩展电缆名称

用 **16** 个或更少的字母来指定所使用的扩展电缆的型号名称。**QCPU** 不使用指定的型号名称。（可用作为用户备忘录）

(d) 点（用于 **QCPU**）

从下列数中选择用于基板插槽的点数：

- **2**（**2** 个插槽）
- **3**（**3** 个插槽）
- **5**（**5** 个插槽）
- **8**（**8** 个插槽）
- **10**（**10** 个插槽）
- **12**（**12** 个插槽）

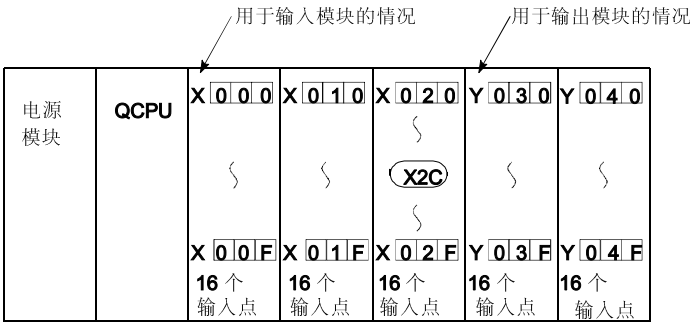
(e) 8 个定位**/12** 个定位（用于 **QCPU**）

对所有的基板选择分配相同的插槽编号。

5.4 什么是 I/O 地址?

I/O 地址是顺控程序中用来接收 QCPU 的 ON/OFF 数据，以及从 QCPU 输出到 ON/OFF 数据的。
输入（X）用于接收 QCPUON/OFF 的数据。输出（Y）用于从 QCPU 输出 ON/OFF 数据。
I/O 地址用十六进制表示。
当使用 16–点 I/O 模块时，I/O 地址是一个连续编号，一个插槽有 16 个点，从□□0 到□□F，如下所示。
安装于基板上的模块分配如下：

- 对于输入模块，I/O 地址的开头是“X”。
- 对于输出模块，I/O 地址的开头是“Y”。



5.5 I/O 地址分配的概念

按下列情况，QCPU 接通或复位时指定 I/O 地址。
为了分配 I/O 地址，应遵循下列规定：

(1) 基板的插槽编号

按照基本模式的设置，设置主基板和扩展基板的插槽编号。（有关基本模式，请参见 5.3 节）

(a) 在自动模式中，插槽编号根据安装到每一块基板上的模块的可用编号来决定。
例如，5 个插槽分配给 5 个插槽基板，12 个插槽分配给 12 个插槽基板。

(b) 在具体模式中，插槽编号以 PLC 参数的 I/O 分配来决定。

(2) I/O 地址分配的顺序

I/O 地址从左依次向右分配给模块，在主基板上从 0_H 开始分配给 QCPU 右边的模块。

(3) 对扩展基板 I/O 地址的分配顺序

对扩展基板 I/O 地址的分配是从主基板 I/O 地址的最后一个编号开始依次进行。
对扩展基板的 I/O 地址的分配应按照扩展基板级设置连接器的顺序，顺次地从左（Y00）至右进行。

(4) 每个插槽的 I/O 地址

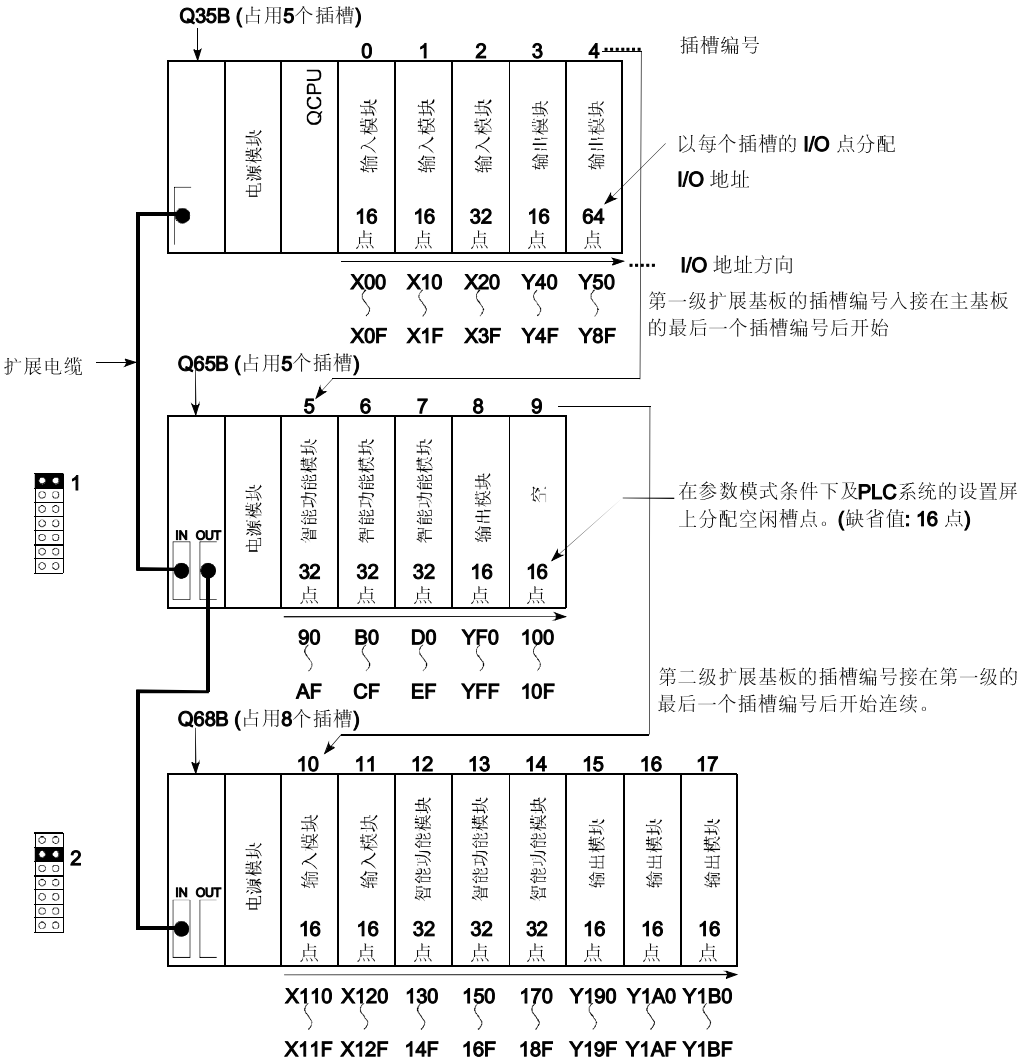
每个基板插槽占用安装 I/O 模块或智能型功能模块（特殊功能模块）I/O 地址的点数。
当在 QCPU 的右边安装 32 点的输入模块时，X0 到 X1F 将作为 I/O 地址来进行分配。

(5) 空闲插槽的 I/O 地址

如果基板具有既不装 I/O 模块又不装智能型功能模块（特殊功能模块）的空闲插槽，将由 PLC 参数的 PLC 系统设置分配的点数分配给空闲插槽。（省缺值：16 点）

要点
当对基板的分配以自动模式进行，而且对扩展基板没有分配级编号时，“0”就分配给插槽编号和扩展基板的 I/O 点数。 结果，即使跳过扩展基板的扩展级，空闲插槽编号也不会增大。

下图为基板设定为自动模式不作 I/O 分配时，I/O 地址分配的示例：

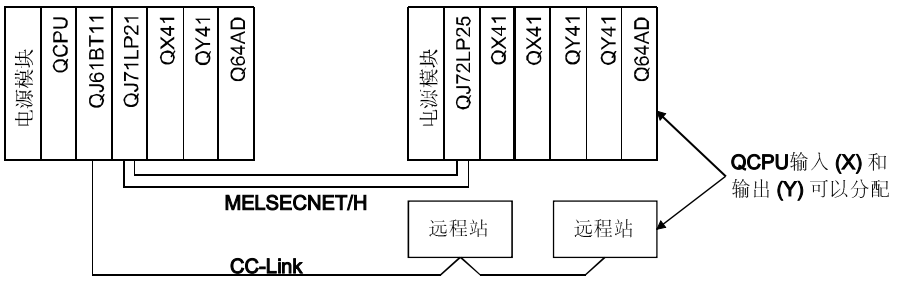


要点
上例所示为智能型功能模块有 32 个 I/O 点。 按智能型功能模块的不同，I/O 点数也不同。 参见所使用的智能型功能模块的手册，且在分配 I/O 地址前检查 I/O 点数。

5 - 9

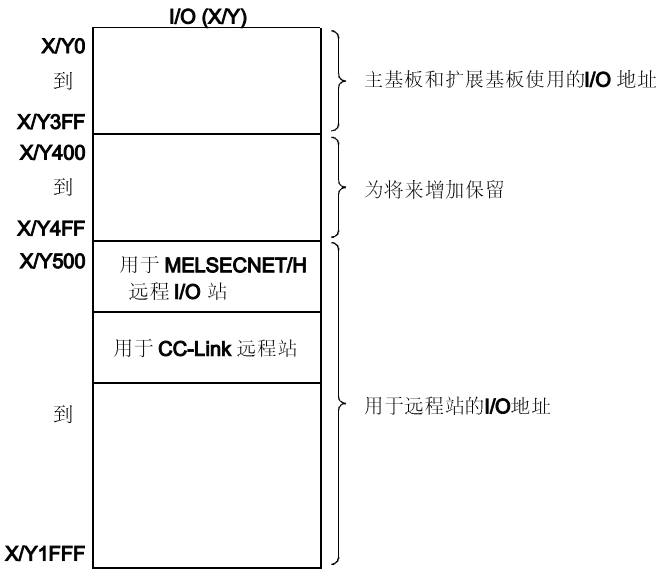
5.5.1 远程站 I/O 地址

能将 QCPU 软元件输入（X）和输出（Y）分配给远程站 I/O 模块和智能型功能模块，并通过 MELSECNET/H 远程网络，CC–Link 和其它远程 I/O 系统进行控制。



当将 QCPU 软元件输入(X)和输出(Y)用于远程站时，可以分配接在主基板和扩展基板的 I/O 模块和智能型功能模块使用的编号后的 I/O 地址。
例如，如果 X/Y0 至 X/Y3FF 是用于主基板和扩展基板的 I/O 模块和智能型功能模块的，则大于 X/Y400 的编号可用于远程站。
但是，用于远程站的 I/O 地址设置时要考虑到主基板和扩展基板的 I/O 模块和智能型功能模块的增加点数。

例如，如果 X/Y0 到 X/Y3FF 的 1024 个点用于主基板和扩展基板，则从 X/Y400 到 X/Y4FF 的 256 个点就必须保留以备将来增加时使用，从下图就能观察到这一状态。



备注

MELESECNET/H 远程 I/O 网络，CC–Link 或其它网络的 I/O 地址分配，并无顺序控制。

5.6 用 GX Developer 分配 I/O 地址

本节阐述用 GX Developer 分配 I/O 地址。

5.6.1 用 GX Developer 分配 I/O 地址的目的

在下列环境下条件用 GX Developer 分配 I/O。

(1) 当将模块更换为非 16 点模块时应保留点数

当前的模块将来改变为 I/O 点数不同的模块时，可以预先保留点数，从而不必再改变 I/O 地址。

例如，你可以将一个 32 点 I/O 模块分配到当前装有 16 点 I/O 模块的插槽上。

(2) 当更换模块时，防止 I/O 地址改变

当非 16 点的 I/O 模块或智能型功能模块由于故障而拆下时，可避免 I/O 地址的改变。

(3) 改变用于程序的 I/O 地址

当分配程序的 I/O 地址不同于实际系统的 I/O 地址时，基板的每个模块的 I/O 地址可设置为程序 I/O 地址。

(4) 设定输入模块和中断模块的输入响应时间（I/O 响应时间）

为了使输入模块和中断模块的输入响应时间和系统相匹配，预先在 I/O 分配中选择“类型”。（详见第 7.7 节）

(5) 设置智能型功能模块的开关

为了设置智能型功能模块的开关，预先在 I/O 分配中选择“类型”。（详见第 7.8 节）

(6) 在 QCPU 出错期间设置输出

当 QCPU 由于停止错误而停止操作时，为了设置输出模块和智能型功能模块的输出状态（保持/清除），预先在 I/O 分配中选择“类型”。

(7) 在智能型功能模块硬件出错时设置 QCPU 操作

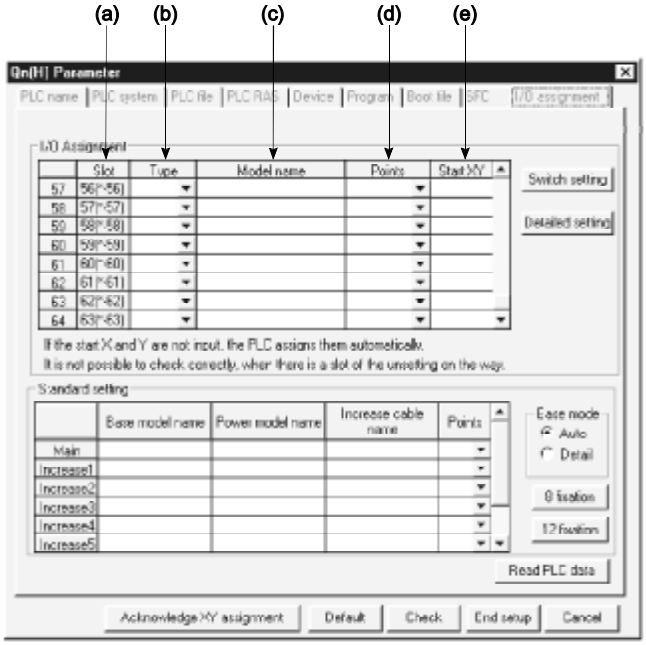
在智能型功能模块硬件出错时，为了设置 QCPU 操作（继续/停止），预先在 I/O 分配中选择“类型”。

要点
为了进行输出模块的响应时间设定和智能型功能模块的开关设置，必须进行 I/O 分配。

5.6.2 用 GX Developer 分配 I/O 地址的概念

(1) 对每个插槽分配 I/O 地址

可以对基板的每个插槽分别指定“Type”（模块型号），“Points”（I/O 点数），和“Start XY”（起首的 I/O 编号）。
例如，要改变分配插槽的 I/O 点数，你只需分配 I/O 的点数。
非指定项目可设置安装基板的状态。
I/O 分配按 PLC 参数的 I/O 分配设置进行。



(a) 插槽

显示基板上的插槽上插槽号和序号。
如果基板在具体模式中没有指定，基板的级号表示为“*”，插槽的序号从主基板的 0 插槽开始计数。

(b) 型号（用于 QCPU）

从下列各项中选择要安装的模块型号：

- 空（空闲插槽）
- 输入（输入模块）
- Hi 输入（对应于高速模块的 Q 系列）*1
- 输出（输出模块）
- I/O Mix（I/O 混合模块）
- 智能（智能型功能模块或 AnS–兼容特殊功能模块）
- 中断（对应于中断模块的 Q 系列）*2

如果型号未予指定，则使用实际安装的模块。

备注

*1. “Hi 输入”可用于 GX Developer 设置（SW5D5C–GPPW–E 之后的产品）。
*2. “中断”可用于 GX Developer 设置（SW6D5C–GPPW–E 之后的产品）。

(c) 型号名称

用不多于 16 个的字母指定安装模块的型号名称。QCPU 不使用指定的型号名称。（可用作为用户备忘录）

(d) 点数（用于 QCPU）

为了改变每个插槽的 I/O 点数，从下列各项中选择：

- 0（0 点）
- 16（16 点）
- 32（32 点）
- 48（48 点）
- 64（64 点）
- 128（128 点）
- 256（256 点）
- 512（512 点）
- 1024（1024 点）

如果未对插槽指定 I/O 点数，则使用实际安装模块的点数。

(e) XY 起始编号（用于 QCPU）

1） 当每个插槽的 I/O 地址改变时，可根据改变情况指定 I/O 的起始地址。
如果未指定插槽的起始 XY 编号，则 I/O 地址就接着当前指定插槽的最后一个编号开始。

2） 要避免在对每个插槽指定 I/O 地址时与由 QCPU 分配的 I/O 地址重叠。
当 I/O 地址重叠时，将会出错（SP，UNIT LAY ERR.）。

(2) 在 I/O 分配后的插槽状态

在 I/O 地址分配给插槽后，分配的 I/O 编号就具有优先权，而与实际安装模块无关。

(a) 如果指定的 I/O 点数少于实际安装的 I/O 模块的点数，则部分安装模块的 I/O 点数就未使用。
例如，当在一个插槽中装入 32 点的输入模块，却指定用作为 16 点的输入模块，则 32 点输入模块的后面 16 点将不能用。

(b) 如果指定的 I/O 点数多于实际安装的 I/O 模块，则超过实际模块的点数成为虚设。

(c) 要确保对安装模块和 I/O 分配设置相同的模块型号。
如果 I/O 分配的模块型号与实际安装的模块不同，将会发生故障。
对于智能型功能模块，要保证 I/O 点数相同。

实际安装模块	I/O 分配	结果
输入模块	输出/空	空
输出模块	输入/空	空
输入模块/输出模块	智能	出错 （SP.UNIT LAY ERR.）
智能型功能模块	输入/输出/空	出错 （SP.UNIT LAY ERR.）
空闲插槽	智能	无出错

(d) 分配 I/O 地址时，一定要使最后一个 I/O 地址在 FFFH的范围之内或少于 FFFH。但如最后一个 I/O 地址超过 FFFH，将会出错（SP.UNIT LAY ERR.）。（GX Developer 系统监视器将以“***”显示 I/O 地址。）

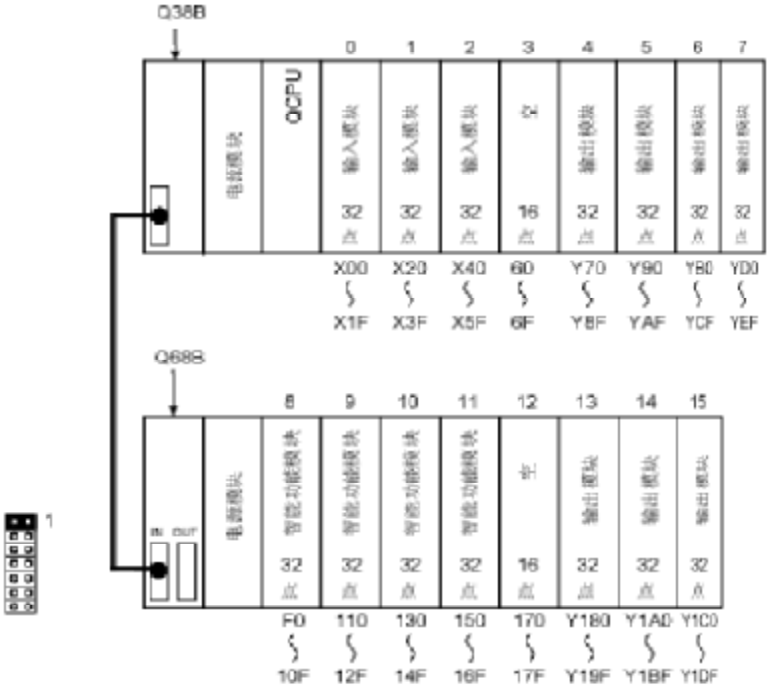
5.7 I/O 地址分配示例

本节为用 GPPW 分配 I/O 地址的示例。

(1) 当在 16—32 点范围内改变空闲插槽点数时：

在当前空闲插槽位置（第 3 插槽）处保留 32 点，这样，当将来安装 32 点输入模块时，I/O 点数就不再改变。（第 12 个空插槽的 16 点不作改变。）*1

(a) 在用 GX Developer 分配 I/O 前的系统配置和 I/O 地址分配

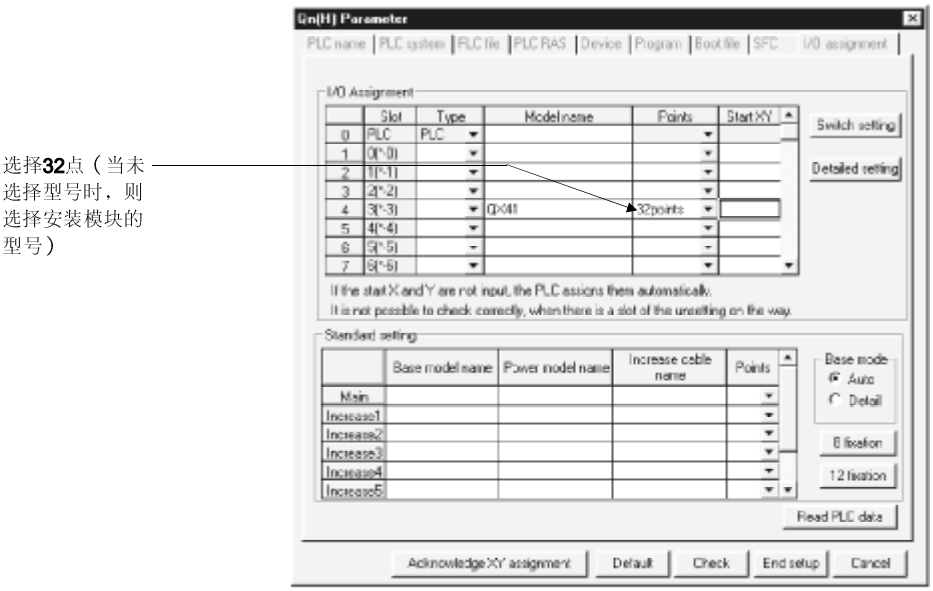


备注

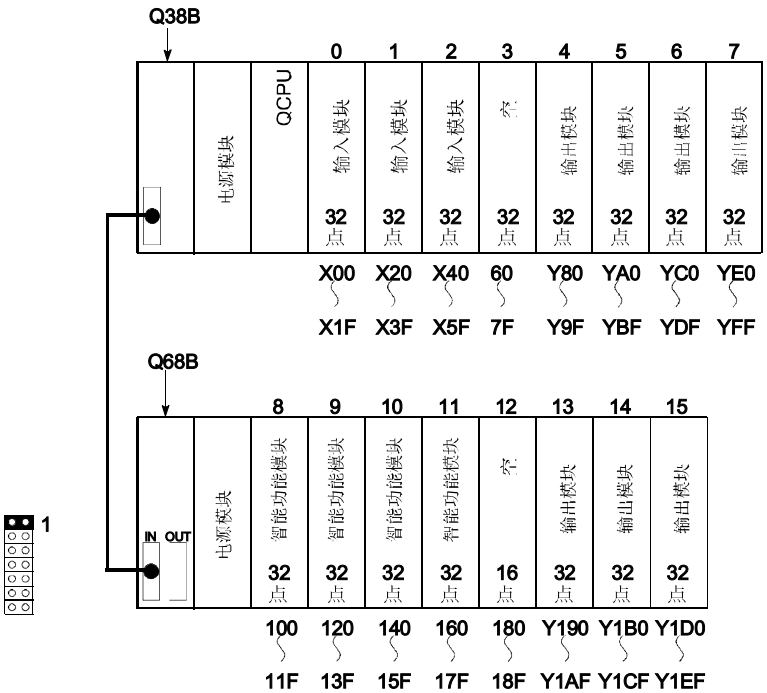
*1: 在这种情况下，PLC 参数的 PLC 系统设置将空闲插槽的点数设定为 16 点。

(b) 用 GX Developer 进行 I/O 分配

在 GX Developer 的 I/O 分配屏幕上指定第 3 槽为“32 点”。



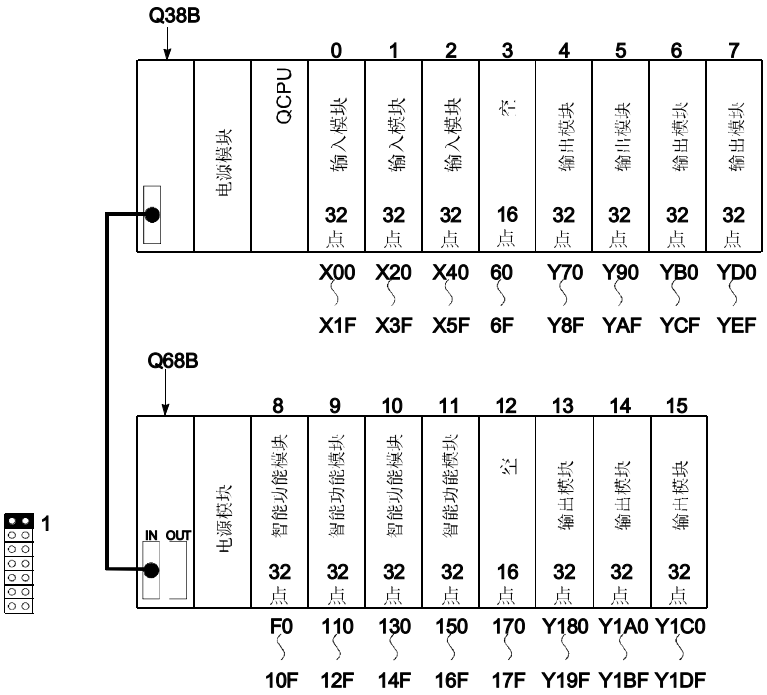
(C) 在用 GX Developer 进行 I/O 分配后的 I/O 地址分配。



(2) 改变插槽的 I/O 地址

将当前空闲插槽（NO.3 插槽）的 I/O 地址改变为 X200 至 X21F。这样，当一个 32 点的输入模块装入当前的空闲插槽（NO.3 插槽）时，NO.4 插槽和后面的插槽的 I/O 地址不再改变。

(a) 在用 GX Developer 进行 I/O 分配前的系统配置和 I/O 地址的分配



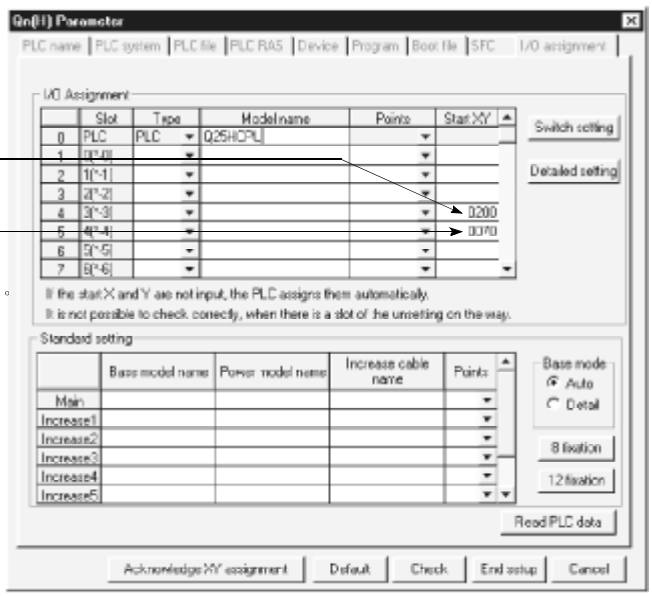
5 - 17

(b) 用 GX Developer 进行 I/O 分配

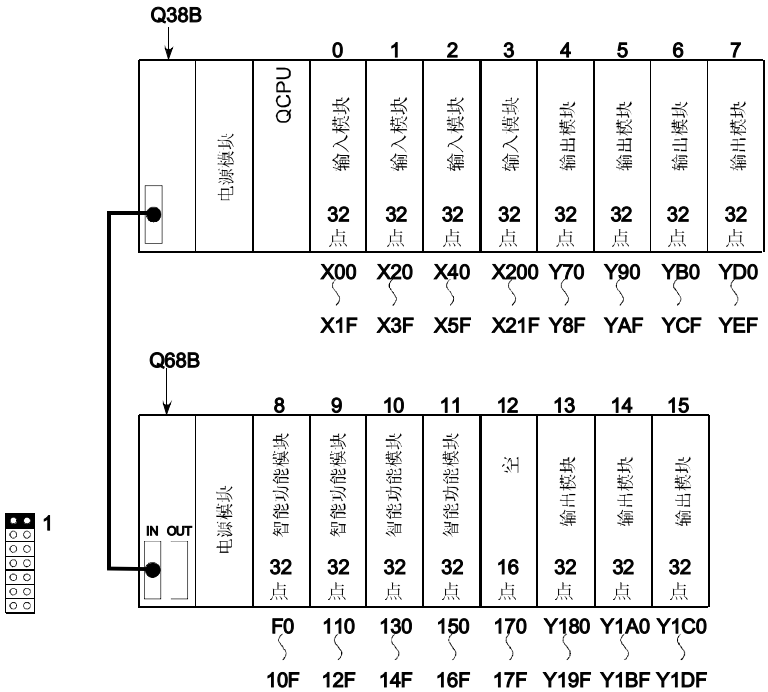
在 GX Developer 的 I/O 地址分配屏幕上将插槽 3 的起始 I/O 地址指定为“200”，插槽 4 的起始 I/O 地址为“70”。

"200" 指定为起始 I/O 地址。

"70" 指定为起始 I/O 地址。（当未指定起始 I/O 编号时，I/O 编号从第3个插槽之后分配）。



(C) 用 GX Developer 进行 I/O 分配后的 I/O 地址分配。



5.8 检查 I/O 地址

GX Developer 的系统监控器能检查 QCPU 的安装模块和它们的 I/O 地址。（有关系统监视参见第 7.18 节）

5 - 19

6. QCPU 文件

(1) QCPU 文件类型

(a) QCPU 参数，程序，注释数据等都可指定“文件名”和“扩展名”，然后保存在下列存储器中。

- 程序存储器
- 标准 ROM
- 存储卡

当从 GX Developer 向 QCPU 读写这些数据时，文件可用它们的类型来指定（参数，程序，注释，等等），而与其扩展无关。（GX Developer 能自动对已指定的文件类型分配适当的扩展名。）

(b) 不能对一个以上的扩展基板模块设置，并使用相同的扩展级编号。

(2) QCPU 文件管理

使用不同的文件名和扩展名称就可使多个文件保存在 QCPU 中。因为 QCPU 也能像一个文件那样处理所提供的程序，通过使用不同的文件名，按其“设计者”，“处理”或“功能”，可分别管理编制的程序。

此外，对保存在 QCPU 中的多个程序也能进行程序执行。

（有关 QCPU 程序执行详见第 4 章。）

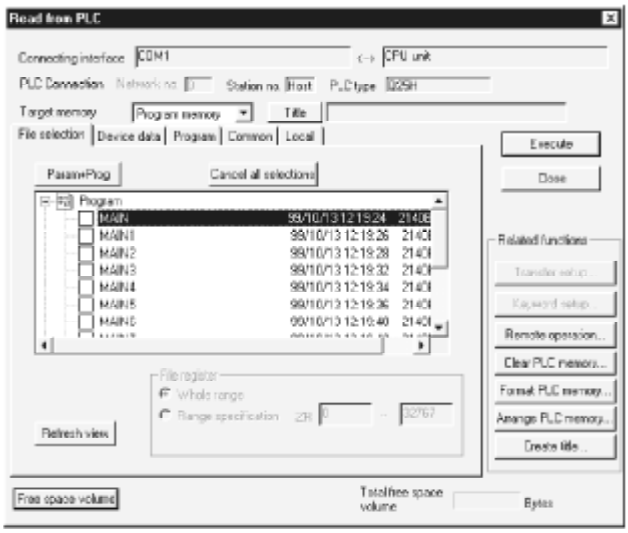
(3) 用 GX Developer 写文件

QCPU 保存用 GX Developer 写入存储器（程序存储器/标准 ROM/存储卡的文件）。

(4) 文件细节

文件名，文件容量和用 GX Developer 创建时设定的写入日期可添加到写入 QCPU 中的每个文件。

当用 GX Developer 监控文件时，文件如下图显示。



(a) 文件名

1) 文件名包括文件名（最多 8 个字符）和扩展名（3 个字符）。

任何由 GX Developer 写入 QCPU 的文件名都以大写字符在屏幕上显示。当用顺控程序命名文件时，文件名用大写字母输入。

对应于指定文件类型的扩展名在文件由 GX Developer 写入 QCPU 时将自动添加到文件名中。

2) 下列 Windows 专用的字母不能用作文件名

- COM1 - COM9
- LPT1 - LPT9
- AUX
- CON
- PRN
- NUL
- CLOCK$

(b) 日期和时间

当文件用 GX Developer 写入 QCPU 时即表示出日期和时间。

屏幕上显示的日期和时间表示 GX Developer 方面的日期和时间。

(c) 容量

用 GX Developer 写入 QCPU 的文件容量用字节单位来表示。（如要查阅最新的 QCPU 数据，可单击更新按钮。）

在 QCPU 中保存的顺控程序及在 ROM 中保存的顺控程序以 4 字节为单位（1 步），而保存在存储卡中的文件以 1 字节为单位。

计算文件容量时，除了文件寄存器，至少 64 个字节（程序为 136 字节）要加入到所有的用户创建文件中。

6.1 关于 QCPU 的存储器

(1) 用户存储器

使用 GPPW/顺控程序，可以在 QCPU 的存储器内创建用户存储器。

QCPU 有下列内部存储器：

- 程序存储器
- 标准 RAM
- 标准 ROM

QCPU 上可安装存储卡以增加用户存储器的容量。

(a) 用于 QCPU 算术运算的程序保存在程序存储器中。保存在标准 ROM 或存储卡中的算术运算用程序引导（读出）至程序存储器中。保存在程序存储器中的一批参数和程序可复制到标准 ROM/存储卡（闪存卡）中。

(b) 参数和程序保存在标准 ROM 中。这些数据用于 QCPU 的 ROM 操作。

(c) 文件寄存器和本地软元件数据保存在标准 RAM 中。在标准 RAM 中使用的文件寄存器当在用作数据寄存器时可执行高速访问。

(d) 存储卡

存储卡可连接到 QCPU 的存储卡接口。

这样就能读/写数据。QCPU 支持三类存储卡：SRAM 卡，闪存卡，和 ATA 卡。

1) SRAM 卡能通过顺控程序在下列情况下读/写程序：
   - 所用文件寄存器超过 32k 点时。
   - 保存采样跟踪数据。
   - 保存 SFC 跟踪数据。
   - 保存故障历史记录数据。

   在顺控程序中文件寄存器可用于 505k 点时的数据读/写。

2) 闪存卡仅能通过顺控程序读出。当用 QCPU 写入的数据通过顺控程序读出，但不改变量据时，闪存卡十分有用。
   使用文件寄存器使得顺控程序能够读出最大为 1018k 点用 QCPU 写入的数据。

3) ATA 卡用于 PLC 用户数据（通用数据）。
   使用在顺控程序中的文件访问指令（如：FWRITE），可以用 CSV 格式/二进制格式存取保存在 ATA 卡上的 PLC 用户数据。

(2) 在 QCPU 存储器或存储卡中保存数据的类型

下表为在标准/RAM/标准 RAM 或在存储卡中保存数据的类型。

数据名称	QCPU内置			存储卡 (RAM)	存储卡(ROM)	
	程序存储器	标准RAM	标准ROM	SRAM卡	闪存卡	ATA 卡
参数	可储存	不可储存	可储存	可储存	可储存	可储存
智能参数	可储存	不可储存	可储存	可储存	可储存	可储存
程序	必需	不可储存	可储存 *1	可储存 *1	可储存 *1	可储存 *1
软元件注释	可储存 *2	不可储存	可储存	可储存	可储存	可储存 *3
软元件初始值	可储存	不可储存	可储存	可储存	可储存	可储存
文件寄存器	不可储存	可储存 *5	不可储存	可储存	可储存 *4	不可储存
本地软元件	不可储存	可储存	不可储存	可储存	不可储存	不可储存
调试数据	不可储存	不可储存	不可储存	可储存	不可储存	不可储存

备注

*1: 为了执行程序，要求启动程序存储器。
*2: 数据能用 GX Developer 操作写入。顺控程序的指令中不能使用软元件注释。
*3: 用顺控程序读要求几次扫描。
*4: 顺控程序只允许读。用顺控程序进行访问，不能写入数据。
*5: 标准 RAM 保存的单独文件最大为 32k 点。
*6: 用下列指令可读/写的数据：

- S.FREND（允许在存储卡规定的文件中成批读出）
- S.FWRITE（允许成批写入到存储卡的规定文件中）

下表为保存在 QCPU 或存储卡中的文件名和数据文件的扩展名。

数据名	文件名
参数	PARAM.QPA
智能参数	IPARAM.QPA
程序	***.QPG
软元件注释	***.QCD
软元件初始值	***.QDI
文件寄存器	***.QDR
本地软元件	***.QDL
调试数据	***.QTD
SFC 跟踪数据	***.QTR
故障历史记录数据	***QFD
PLC 用户数据	***.***

标有*这部分可由用户命名。

(3) 驱动器编号

(a) QCPU 使用驱动器编号来控制标准 RAM，标准 ROM 和存储卡。GPPW 指定了一个选择存储器（标准 RAM，标准 ROM 或存储卡）来执行参数的读/写和从 QCPU 进出的程序文件。当使用 GX Developer 时就不需要规定驱动器的编号。

(b) 下表所示为用于指定选择存储器（程序存储器，标准 RAM， 标准 ROM 或存储卡）的驱动器编号。当通过对串行通信模块的存取进行读/写时，驱动器编号就必须用来指定一个选择的存储器。

存储器		驱动器编号
QCPU 内部	程序存储器	0
	标准 RAM	3
	标准 ROM	4
存储卡（RAM）	SRAM 卡	1
存储卡（ROM）	闪存卡	2
	ATA 卡	2

(4) 存储器容量和格式化

下表所示为 QCPU 存储器的容量和存储器是否要格式化。

		Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU	是否要格式化
标准 RAM		64k 字节			256k 字节*		需要 (用 GX Developer)
程序存储器		28k 步 (112k 字节)	28k 步 (112k 字节)	60k 步 (240k 字节)	124k 步 (496k 字节)	252k 步 (1008k 字节)	需要 (用 GX Developer)
标准 ROM		112k 字节	112k 字节	240k 字节	496k 字节	1008k 字节	不需要
存储卡	SRM 卡	Q2MEM–1MBS:IM 字节					需要 (用 GX Developer 或个人计算机)
	闪存卡	Q2MEM–2MBS:2M 字节 Q2MEM–4MBS:4M 字节					不需要
	ATA 卡	Q2MEM–8MBS:8M 字节 Q2MEM–16MBS:16M 字节 Q2MEM–32MBS:32M 字节					需要 (用 GX Developer 或个人计算机)

*:系列编号前五位数字为“02091”或更早期的 Q12HCPU 和 Q25HCPU，其存储器容量为 64k 字节。

6.2 程序存储器

(1) 什么是程序存储器

(a) QCPU 的程序存储器是一个内部的 RAM，保存由 QCPU 执行的程序。

(b) 在程序存储器中的数据保存靠 QCPU 的内置电池（Q6BAT）支持。

(c) 在 QCPU 第一次使用之前，程序存储器必须用 GX Developer 进行格式化。
有关用 GX Developer 格式化的步骤，详见 GX Developer 手册。

要点
(1) 在 QCPU 第一次使用之前，程序存储器必须用 GX Developer 进行格式化。 有关用 GX Developer 格式化的步骤，详见 GX Developer 手册。 (2) 程序以 1k 步为单位保存在程序存储器中。

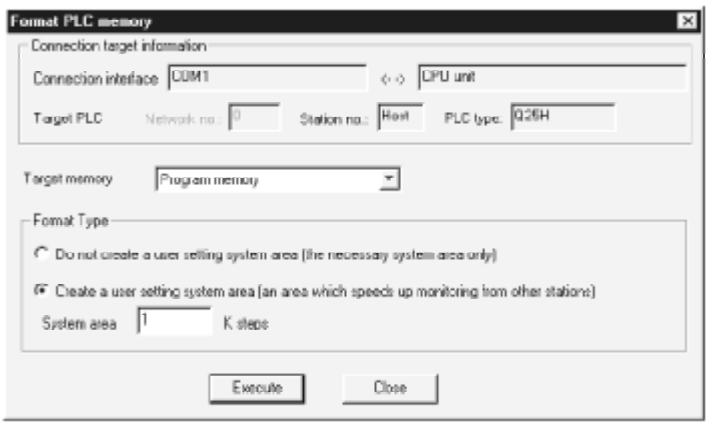
(2) 数据保存

参数或程序数据可保存在程序存储器中。有关数据保存在程序存储器中的类型，参见第 6.1 节。

(3) 格式

(a) 格式化

选择“在线”→“PLC 存储器格式化”打开 PLC 存储器格式化对话框。从目标存储器表上选择“程序存储器”。



(b) 格式化之后的存储器容量

格式化之后的存储器容量如下表所列

表 6.1 格式化之后的存储器容量*1

型号名称	存储器*2	最大保存文件数
Q02CPU	28k 步（114688 字节）	28
Q02HCPU	28k 步（114688 字节）	28
Q06HCPU	60k 步（245760 字节）	60
Q12HCPU	124k 步（507904 字节）	124
Q25HCPU	252k 步（1032192 字节）	252 *3

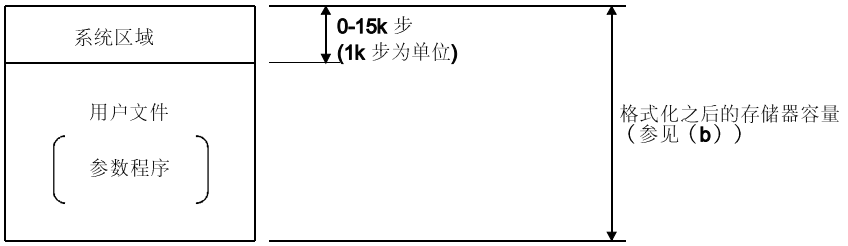
(c) 格式化注意事项

1) 程序存储器格式化

QCPU 程序存储器只有在用 GX Developer 格式化之后才能使用。

在程序存储器格式化时，必须指定是否要给用户设置分配一个系统区域。

（分配给用户设置的系统区域可达 16k 步（以 1k–步为单位）。）



2) 系统区域设置

如果将 RS–232 和 USB 接口与 GX Developer 相连，则系统区域用户设置数据就从 GX Developer 连接到串行通信模块，作为寄存监控数据，

通过 GX Developer 连接到串行通信模块。

为系统指定空间和用户定义区域的设置，通过连接到串行通信模块的 GX Developer 的操作会使监视操作容易得多。

虽然指定用户设置区域加快了从 GX Developer 连接到系统通信模块的监控，但也减少了提供给用户文件的空间量。

备注

1) *1: 本表所示为系统区域分配为 0 k 步时的例子。
2) *2: 在计算存储器容量时，1 步相当于 4 个位。
3) *3: 可执行程序最多为 124 个。超过 124 个不能执行。

6.3 关于标准 ROM

(1) 什么是标准 ROM

(a) 标准 ROM 用于 QCPU 的 ROM 时的操作。

(b) 在 PLC 参数对话框的引导文件表上设置后，保存在标准 ROM 里的程序可使用并引导（读）至程序存储器。

(c) 标准 ROM 不需要格式化。

(d) 用 GX Developer 的在线“PC 写入”（闪存 ROM）和“创建程序存储器 ROM”（参见第 6.6.1 节）写入标准 ROM。不必用 GX Developer，而用“自动写入标准 ROM”也能从存储卡写入标准 ROM 中。

要点

(1) 在将数据写入标准 ROM 之前，所有先前保存在标准 ROM 中的数据都要擦除。因此，要将数据写入标准 ROM，在你向里面写入必须的数据之前，首先要读和复制先前保存在标准 ROM 里的数据。注意当保存在标准 ROM 里的数据用于顺控程序时会出错，因为数据是写入标准 ROM 的。
(2) 有关用 GX Developer 格式化的程序，详见 GX Developer 手册。
(3) 保存在标准 ROM 中的程序单位为 1 k 步。

(2) 数据保存

标准 ROM 保存诸如参数和程序之类的数据。
有关数据在标准 ROM 的保存，参见第 6.1 节。

(3) 存储器容量

表 6.2 所示为标准 ROM 的存储容量。

表 6.2　存储器容量

型号名称	存储容量*2	最大保存的文件量
Q02CPU	28k 步（114688 字节）	28
Q02HCPU	28k 步（114688 字节）	28
Q06HCPU	60k 步（245760 字节）	60
Q12HCPU	124k 步（507904 字节）	124
Q25HCPU	252k 步（1032192 字节）	252

备注

计算存储器的容量，1 步等于 4 位。

6.4 关于标准 RAM

(1) 什么是标准 RAM?

(a) 当在 QCPU 上未装上存储卡而使用文件寄存器和本地软元件时，采用标准 RAM。

(b) 当第一次使用 QCPU 时，在通过 GX Developer 操作而使用 RAM 之前，必须格式化。有关格式化步骤参见 GX Developer 手册。

(c) 使用在线功能“写入 PLC”，可将数据写入标准 RAM。

(2) 保存数据

一个标准 RAM 可保存两类文件：文件寄存器文件和本地软元件文件，任何其它文件都不能写入标准 RAM。

(3) 格式

(a) 格式化

为了标准 RAM 的格式化，选择“在线”→“PLC 存储器格式化”，然后在“目标存储器”列表框中选择“标准 RAM”。有关 PLC 存储器格式对话框，参见第 6.2 节。

(b) 格式化之后的存储器容量

表 6.3 所示为经“格式化”的标准 RAM 存储容量。

表 6.3 存储容量

CPU 型号	系列编号	保存文件数量	保存文件数量	
			文件寄存器文件	本地软元件文件
Q02CPU	—	32k 字（64k 字节）	1	1
Q02HCPU	—	32k 字（64k 字节）	1	1
Q06HCPU	—	32k 字（64k 字节）	1	1
Q12HCPU	“02091”之前	32k 字（64k 字节）	1	1
	“02092”之后	128k 字（256k 字节）		
Q25HCPU	“02091”之前	32k 字（64k 字节）	1	1
	“02092”之后	128k 字（256k 字节）		

(4) 安全要求

当在标准 RAM 中设置文件寄存器和本地软元件时，对系列编号前五位数字是“02092”的 Q12HCPU 和 Q25HCPU 来说，安全的存储容量为每单位 1024 字节。

对系列编号前五位数字是“02091”的 Q12HCPU/Q25HCPU，Q02CPU，Q02HCPU 和 Q06HCPU 来说，安全的存储容量为每单位 512 字节。

要点

当用访问 02092 上的标准 RAM 和其后的 QCPU 的指令，采用系列访问法（ZR□）来规定文件寄存器时，处理从 02091 和先前的 QCPU 来的每个命令所需时间将延长（QnCPU：平均 0.65μs，QnHCPU：平均 1.1μs）。如下所示为使用 MOV 命令时的处理时间。

命令	Q12HCPU		Q02CPU	
	02092P 之后	02091 之后	02092 之前	02091 之前
MOV KO RO	0.11	0.11	0.26	0.26
MOV KO RO	3.55	2.88	7.71	6.64

6.5 存储卡

(1) 存储卡

(a) 存储卡用来扩展 QCPU 内部存储器的容量。

(b) 有三种类型的存储卡用于 QCPU：SRAM 卡，闪存卡和 ATA 卡。

要点

(1) 存储卡在第一次使用之前，必须用 GX Developer 格式化。
有关用 GX Developer 的格式化的过程，详见 GX Developer 手册。

(2) 在将数据写入闪存卡之前，先前保存在闪存卡里的数据要擦除。因此，为了将数据写入闪存卡中，在你向里面写入必须的数据之前，首先要读和复制先前保存在闪存卡里的数据。注意写入闪存卡的数据和保存在闪存卡里面的数据用于顺控程序时会出错。

(3) 保存在存储卡的程序以 512 字节（128 步）步为单位。

(2) 保存数据

存储卡保存参数和程序数据。关于在存储卡中保存的数据的类型，参见第 6.1 节。

(3) 格式

(a) 格式化

为了将存储卡格式化，选择“在线”→“PLC 存储器格式化”，然后在“目标存储器”列表框中选择“存储卡(RAM)”或“存储卡（ROM）”。有关 PLC 存储卡格式对话框，参见第 6.2 节。

(b) 格式化之后的存储容量

表 6.4 所示为存储卡格式化之后的存储容量。

表 6.4　存储容量

存储卡型号	存储容量	保存的文件数量
Q2MEM–1MBS	1011.5k 字节	256
Q2MEM–2MBF	2032k 字节	288
Q2MEM–4MBF	4080k 字节	288
Q2MEM–8MBA	7972k 字节	512
Q2MEM–16MBA	15964k 字节	512
Q2MEM–32MBS	31918k 字节	512

(c) 安全要求

对格式化的存储卡来说，“存储卡资料”区是在存储卡中自动创建的。这就表示因新创建的“存储卡资料区域”占用容量，可用的空间会减少。

6.6 在标准 ROM 或闪速卡中写入数据

6.6.1 用 GX Developer 在标准 ROM 或闪速卡中写入数据

GX Developer Online 菜单上的“写入 PLC”功能不允许用户将文件写入标准 ROM 或闪存卡中。为了通过 GX Developer 的操作将文件写入标准 ROM 或闪存卡中，GX Developer 在线菜单提供两种功能：“将程序存储器写入 ROM”和“写入 PLC（闪存卡）”。

(1) 将程序存储器写入 ROM

(a) “将程序存储写入 ROM”功能能将保存在程序存储器里的批量文件写入标准 ROM 或写入闪存卡中。该功能也可用于调试保存在程序存储器里的程序。

(b) 在执行“将存储写入 ROM”的功能时，在保存在程序存储器里的批量文件写入之前，所有保存在标准 ROM 或闪存卡里的文件要被擦除。没有文件可加入到标准 ROM 或闪存卡里。

(c) 标准 ROM 或闪存卡的存储容量和程序存储器相同。不能使用比程序存储器存储容量大的存储器。

(d) 为了执行“将程序存储写入 ROM”的功能，设定 GX Developer 时间–检查长度为 60 秒或更长。较短的时间–检查可能会在 GX Developer 一侧发生超时。为了在本地站用 GX Developer 操作通过 CC–Link 网络执行“将程序存储写入 ROM”功能，设定 CC–Link CPU 监控时间（SW0A）为 60 秒或更长。缺省值为 90 秒。在设定时使用缺省值。

(2) 写入 PLC（闪存卡）

(a) 当用 GPPW 将批量文件写入标准 ROM 或闪存卡中时，可使用“写入 PLC（闪存卡）”功能。

(b) “写入 PLC（闪存卡）”功能可将数据装满标准 ROM 或闪存卡所有可用的空间。如果写入闪存卡的程序步数较少，将用较长时间写入带有程序的闪存卡。当 RS–232C 接口装上 Q2MEM–4MBF，波特率为 115.2kbps 时需要 14 分钟。为了将数据写入闪存卡，可增大波特率或使用 USB 接口。如果从本地站执行“写入 PLC（闪存卡）”功能，通信时间将更长。

(c) 为了执行“写入 PLC（闪存卡）”功能，设定 GX Developer 时间–检查长度为 60 秒或更长。较短的时间–检查可能会在 GX Developer 一侧发生超时。为了在本地站用 GX Developer 操作，通过 CC–Link 网络执行“将程序存储写入 ROM”功能，设定 CC–Link 的 CPU 监控时间（SW0A）为 60 秒或更长。缺省值为 90 秒。在设定时使用缺省值。

(d) 当执行“写入 PLC（闪存卡）”功能时，在用 GX Developer 规定的成批文件写入之前，所有保存在标准 ROM 或闪存卡里的文件都要擦除。没有文件可以加入到标准 ROM 或闪存卡里。为了将新文件装入旧文件里，所有的旧文件都要从 QCPU 里读出，然后再写入到 QCPU 中。

(e) 当 QCPU 在 RUN 状态时，可执行“写入 PLC（闪存卡）”功能，但是在下列情况下，QCPU 进入 STOP 状态后，执行“写入 PLC（闪存卡）”功能。

1） 闪存卡的文件寄存器用于顺控程序。

2） 用 PLC 参数将文件寄存器设置为“设置不使用”，可将文件寄存器用于顺控程序。

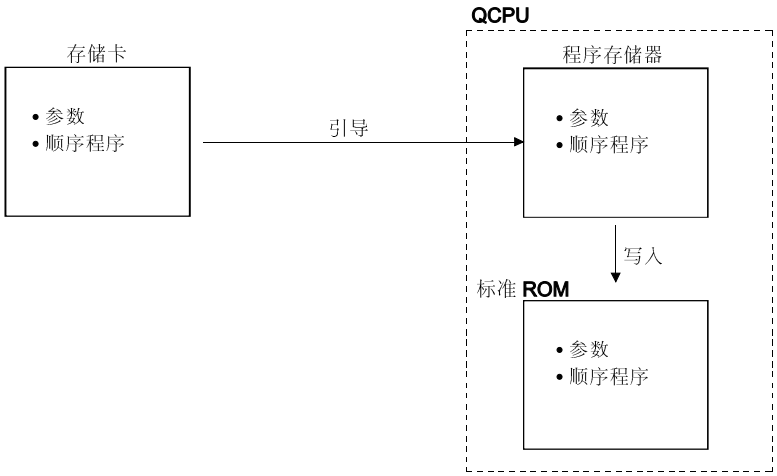
如果 QCPU 在 RUN 状态时，执行“写入 PLC（闪存卡）”功能，可能会出错，QCPU 可能停止运行。

(f) 当执行“写入 PLC（闪存卡）”功能时，不能从其它模块进行读/写。否则其它模块方面会发生超时。

要点
当 QCPU 扩展到 STOP 状态而执行“写入 PLC（闪存卡）”功能时，不要将其设置为 RUN 状态。 当写入闪存卡时一般不执行 RUN。 当写入闪存卡完成后才能执行 RUN。

6.6.2 自动写入标准 ROM（从存储卡将所有的数据自动写入标准 ROM）

自动写入标准 ROM 就表示不用 GX Developer，而将保存在存储卡里的参数和顺控程序写入 QCPU 的标准 ROM。（用 GX Developer（SW6D5C–GPPW–E 或更高）可将参数和顺控程序写入存储卡。）
如下图所示，用标准 ROM 的自动写入，可将参数和顺控程序从存储卡引导至程序存储器，然后再把引导过来的参数和顺控程序从程序存储器写入标准 ROM。



自动写入标准 ROM 可用于修改用标准 ROM 执行 ROM 操作的 QCPU 程序。
用 GX Developer 执行标准 ROM 的改写，但是使用自动写入标准 ROM 时，可以将写上参数和修改后程序的存储卡插入 QCPU，然后从存储卡写入标准 ROM。

自动写入标准 ROM 也可通过系列编号前五位数字为“02092”或更高的 QCPU 和 GX Developer（SW6D5C–GPPW–E 或更高）的结合来执行。
如果设定为自动写入标准 ROM 的存储卡安装在系列编号前五位为比“02092”早的 QCPU 上，则就从标准 ROM 执行引导操作。

自动写入标准 ROM 时以下各项是必须的。
- 用 PLC 参数设置“自动写入标准 ROM 设定”。
- 保存有参数和程序的存储卡。
- 存储卡装入 QCPU，而 QCPU 开关设置好。

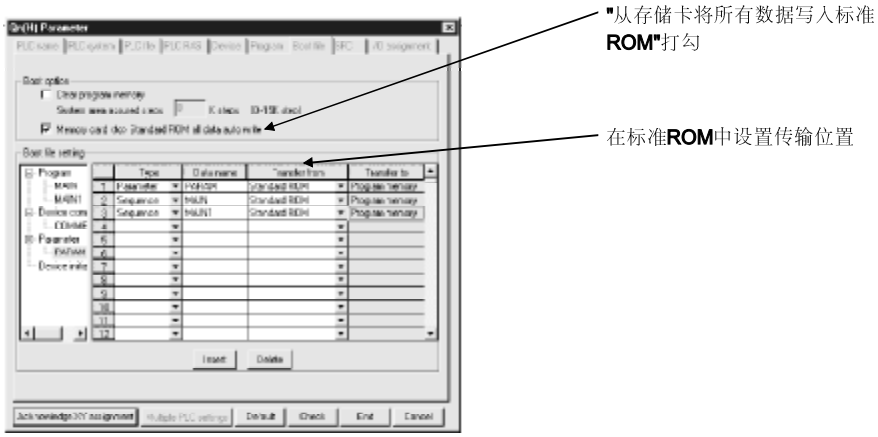
要点
在 QCPU 控制暂停时，可执行自动写入标准 ROM。 当自动写入标准 ROM 完成后，会触发暂停错误（BOOT OK（出错代码：9020））。 当自动写入标准 ROM 完成后，QCPU 必须复位，或重新启动可编程控制器电源。

(1) 自动写入标准 ROM 的执行步骤

观察下列自动写入标准 ROM 的步骤。

(a) 用 GX Developer 操作（设置自动写入标准 ROM）

1） 在 PLC 系数引导文件设置中将“从存储卡将所有的数据写入标准 ROM”打勾。
这样就设置好了用引导文件引导的参数和程序。
（在“标准 ROM”设置中设置传输位置。）



2） 存储引导到存储卡中的参数和程序。

(b) QCPU 操作（自动写入标准 ROM）

1） 将 PLC 电源切断。

2） 安装存有引导到 QCPU 中去的参数和程序的存储卡。

3） 用 CPU 的微型双列开关设置装上的存储卡的参数的有效驱动器。
- 当装入 SRAM 卡时：SW2：ON，SW3：OFF
- 当装入闪存/ATA 卡时：SW2：OFF，SW3：ON

4） 将 PLC 电源接通。

5） 当自动写入标准 ROM 完成时，“BOOT LED”（引导灯）闪烁，QCPU 会采取暂停错误状态。

6） 将 PLC 电源切断。

7） 取下存储卡，然后用 CPU 的微型双列开关设置标准 ROM 中参数的有效驱动器。
- 标准 ROM：SW2：ON，SW3：ON

(c) 当 PLC 切换到 ON 而启动实际操作时，参数和程序将从标准 ROM 引导至程序存储器中。

(2) 注意事项

本节概述当自动写入标准 ROM 时应遵守的注意事项。

(a) 如果从存储卡中引导出来的文件与保存在程序存储器里的文件共享同样的文件名，则存储卡数据就被覆盖。如果从存储卡中引导出来的文件与保存在程序存储器里的文件不共享同样的文件名，则会被加入到程序存储里去。此时如果超过程序存储器的容量，就会触发“FILE SET ERROR（出错代码：2401）”。

(b) 在程序存储器清除之后可选择是否执行引导，或者当从存储卡引导到程序存储器时，不清除程序存储器而执行引导。
当自动写入标准 ROM 时，在清除程序存储器之后执行引导，可以防止引导过程中程序存储卡的溢流。

(c) 当 QCPU 参数有效驱动设置在“存储卡”位置时，则“从存储卡将所有文件写入标准 ROM”的设置只对引导文件有效。
如果参数的有效驱动设置为“程序存储器”或“标准 ROM”，则“从存储卡将所有文件写入标准 ROM”的设置对引导文件无效。

6.7 执行标准 ROM/存储卡程序（引导运行）

(1) 执行 QCPU 程序

(a) QCPU 能处理保存在程序存储器里的程序。
QCPU 不能执行保存在标准 ROM 或存储卡里的程序操作。

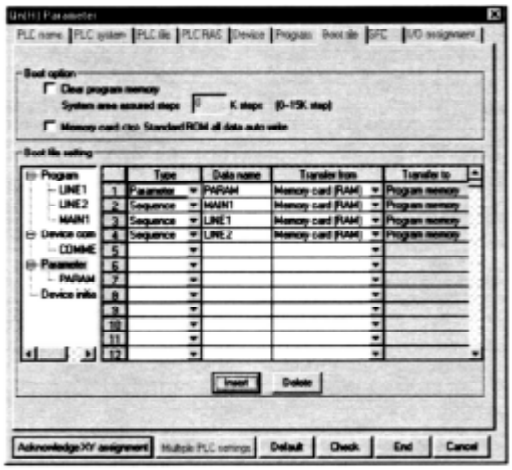
(b) 为了处理保存在标准 ROM 和存储卡里的程序，用 GX Developer 指定引导（读）到程序存储器里的文件名。（用 PLC 参数设置引导文件）
当电源接通或 QCPU 复位时，由引导文件设置指定文件名的程序，从标准 ROM/存储卡中引导到程序存储器中，并进行处理。

(2) 引导运行的准备

在作引导运行的准备时，执行下列步骤：

(a) 用 GX Developer 创建一个程序。
创建一个用于引导运行的文件。

(b) 用 GX Developer 选择一个引导文件。
从 PLC 参数对话框里的“引导文件”表中选择一个引导文件。



(c) 进行 QCPU 硬件设置。
设置 QCPU 微型双列开关，以指定参数驱动的驱动器。

(d) 插入存储卡
如果在引导运行中想将参数或程序储存在存储卡里，则在插槽中插入存储卡。

(e) 用 GX Developer 写入参数和程序
将参数写入参数驱动的驱动器。将程序写入 PLC 参数对话框的“引导文件”表上指定的存储器。

(f) 执行程序
用 RESET/L.CLR 开关将 QCPU 复位。在指定的存储器中完成引导运行后，“BOOT LED”亮起。

(3) QCPU 处于运行状态时改变程序文件

(a) 当 QCPU 处于运行状态时，通过使用顺控程序中的下列指令，可以对从标准 ROM 或存储卡引导到程序存储器的程序文件进行增加/改变/删除。

- PLOAD（从存储卡装入程序）
- PUNLAOD（从程序存储器卸载文件）
- PSWAP（装入+卸载）

有关 PLOAD，PUNLAOD 和 PSWAP 指令，详见 QCPU（Q 模式）/QnACPU 编程手册（公共指令）。

(b) 即使在 QCPU 处于运行状态时改变程序文件，在 PLC 参数对话框的程序表上指定的设置仍保持不变。当 QCPU 处于停止状态，在 PLC 参数对话框的程序表上的设置必须调整到 QCPU 运行状态的任一种改变（程序名称的增加，改变或删除）。

如果 PLC 参数对话框的程序表上不进行调整，则当 QCPU 从停止状态进入运行状态时就会出错。

(4) 在标准 ROM/存储卡上执行程序的注意事项

(a) 对引导运行来说，要将引导文件设置的参数（PLC 参数）保存在标准 ROM 或存储卡里。如果参数保存在程序存储器中，参数驱动驱动器设置为“程序存储器”，在 PLC 参数对话框中的引导文件设置就不起作用。结果，当电源接通或当 PLC 复位时，不执行引导运行。

(b) 当使用存储卡（RAM）执行引导运行时，在 QCPU 处于运行状态时如果程序写入程序存储器，任何变化都会在保存在存储卡（RAM）的程序中反映出来。有关运行状态下写入程序，详见第 7.10 节。

(c) 当使用标准 ROM/存储卡（ROM）执行引导运行时，在 QCPU 处于运行状态时，如果程序写入程序存储器，任何变化都不会在保存于标准 ROM 或存储卡（ROM）的程序中反映出来。

(d) 在 PLC 参数对话框的引导文件表里，把引导文件的最大数量设定为保存在程序存储器的文件数量。凡遇下列的任一种情况引导文件的数量就会减少。

- 指定文件标题。
- 设置引导文件的 PLC 参数被引导。

6.8 程序文件配置

(1) 程序文件配置

(a) 程序文件由文件标题，执行程序，和为“在运行期间写入”分配的存储器组成。

(b) 保存在 QCPU 中的文件容量包括上述的所有部分，如下：

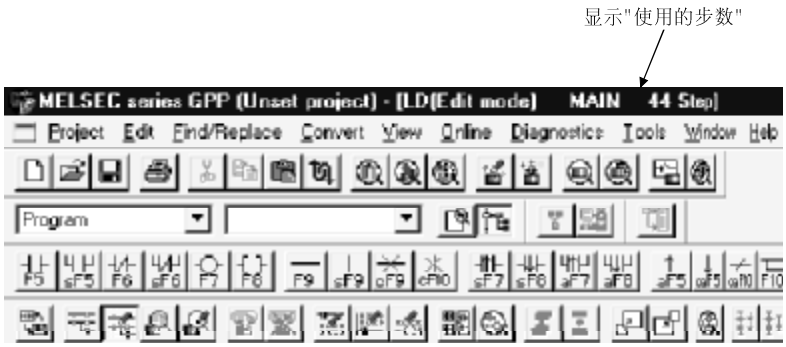
1) 文件标题：文件名，文件容量，文件创建日期等等都保存在这一区域。
文件标题容量固定为 136 字节（34 步）。

2) 执行程序，创建的程序保存在这一区域。
1 步为 4 字节。

3) 为“在运行期间写入”而分配存储器：该区域用于在运行期间 GPPW 执行的步数增加时写入程序。
缺省值设定为 500 步（2000 字节）。
使用 PLC 程序的在线写入，用于“运行期间写入”而分配的存储器数量可以改变。
如果用于运行期间写入而分配的存储器数量不够，其数量可以重新定义。
（参见第 7.10.1 节）

(2) 用 GX Developer 显示程序容量

在用 GX Developer 编程的过程中，程序容量（文件标题容量和创建程序的步数的总和）按步数为单位来显示。如下图。
在编程过程中显示创建程序的容量。



6.9 GX Developer 文件操作和文件处理的注意事项

6.9.1 文件操作

使用 GX Developer 的“在线”功能，可对保存在程序存储器，标准 ROM 和存储卡里的文件进行操作。如下表 6.5 所示。
但是可以进行的文件操作根据是否有相应的口令（由 GPPW 登录），QCPU“写保护”开关设置状态，和 QCPU RUN/STOP 状态而不同。

表 6.5 使用 GX Developer 的文件操作

文件操作	操作有效/无效				操作说明
	A*	B*	C	D	
从PLC读出	○	△	○	○	文件从相关存储器中读出。
写入PLC	△	△	×	○	文件写入程序存储器或SRAM卡。
校验PLC	△	△	○	○	校验目标存储器和GX Developer文件。
将程序存储器写入ROM	△	△	×	○	从程序存储器将成批文件写入标准ROM或闪存卡中。
写入PLC（闪存卡）	△	△	×	○	用GX Developer将成批文件写入标准ROM或闪存卡中。
删除PLC数据	△	△	×	×	删除保存在存储器的文件。
PLC存储器的格式化	○	○	×	×	执行存储器的格式化。
排列PLC存储器	○	○	×	×	不相互邻接的文件重新组合，使其相互邻接。

备注

1) 上表中在“操作有效/无效时使用的代码（A，B，C，D）说明如下。

T 表 6.6 操作有效/失效

代码	说明
A	当在文件中注册“禁止写入”口令时
B	当在文件中注册“禁止读/写”口令时
C	当 QCPU“系统保护”开关接通时
D	当 QCPU 运行状态有效时

2)*: 只有当口令相符时方能执行。

6.9.2 文件处理的注意事项

(1) 在程序操作期间电源切断（或复位）

(a) 如果在文件操作期间电源切断，不会产生文件移位，存储器数据也不会丢失。

(b) 如果 **QCPU** 备用电池（**Q6BAT**）支持有效时，程序存储器数据不会丢失，在下列文件操作期间电源切断，将引起文件移位。

- 文件容量改变
- 排列 **PLC** 存储器
- 新的文件创建
- 在运行状态下写入程序文件
- 在超过为“运行期间写入”而分配的存储器容量时写入程序
- 用 **PLOAD** 指令读文件

在电源切断时，除非把存储卡取出，否则保存在存储卡中的文件不会丢失

要点
(1) 如果进行上述操作，直至电源切断为止的数据会保存在 **QCPU** 的内部存储器中，电源重新接通恢复。因为这个缘故，为了保存内部存储器数据，需要用备用电池

(2) 用多个 **GX Developer** 同时访问一个文件

QCPU 允许用一个 **GX Developer** 访问一个文件。当用多个 **GX Developer** 访问同一个文件时，只有当用 **GX Developer** 的当前处理完成时，在下一个处理之前方可访问文件。

备注

有关 **PLOAD** 指令，详见 **QCPU**（**Q** 模式）**/QnACPU** 编程手册（公共指令）。

6.9.3 文件容量

文件容量因所用文件的类型而不同。当使用程序存储器，标准 RAM，标准 ROM 和存储卡时，参照下表 6.7 计算文件容量。

为文件提供的空间如下：

• 程序存储器，标准 ROM：4096k 字节（1k 步）

• 存储卡：512 字节

在引导运行时，如将一个文件从存储卡传输至程序存储器时，在文件传输后可用的存储器容量会改变。

表 6.7 文件容量表

功能	估算文件容量（字节）
驱动器标题	64
参数	缺省：564（用参数设置增加） 参数： 引导设置至 70+（18×（文件数量）） MELSECNET/H 设置最大每个模块增加 4096 以太网设置最大每个模块增加 922 用 CC–Link 设置最大每个模块增加 251
顺控程序	136+（4×步数））
软元件注释	74+（每个软元件容量注释数据容量的总和） • 一个软元件注释数据容量=10+10250×a+40×b • a:（软元件点数）/256 之商 • b:（软元件点数）/256 之余数
软元件初始值	66+44×n+2×（软元件初始值设定时指定的软元件总点数） • n：指定的软元件初始值数量
文件寄存器	2×（文件寄存器点数）
取样跟踪数据	362+（20+2×（软元件点字数）+（位软元件点数）/8）×（跟踪数）+12×（软元件范围）
故障历史记录数据	72+54×（保存的故障数）
SFC 跟踪数据	最大 48k（以 1k 为单位）
本地软元件	70+6（指定的软元件类型）+2×（M 和 V 总点数）/16+（D 点数）+18×（T，ST 和 C 的总点数）/16）×（程序数） • 符号“M，V，D，T，ST，和 C”代表下列软元件 M：内部继电器 V：边沿继电器 D：数据寄存器 T：定时器 ST：相关定时器 C：计数器

下列为在程序存储器里写入参数和顺控程序时，计算所需存储器容量的示例。

(1) 写入文件

文件名	程序容量*
PARAM.QPA(参数)	—
MAIN.QPG(顺控程序)	5000 步（20000 字节）
SUB.QPG(顺控程序)	11500 步（46000 字节）

* 表示用 GX Developer 显示的程序容量（文件标题和创建的文件步数的总和。）（参见第 6.8 节）

(2) 写入条件

(a) 参数：缺省设置（564 字节）

(b) 在运行期间，安全的写入步数：缺省设置（500 步（2000 字节））

(3) 文件存储器容量计算

文件名	文件容量（单位：字节）		存储器容量*
PARAM.QPG	564		4096 字节/1k 步
MAIN.QPG	顺控程序容量	20000	24576 字节/6k 步
	运行期间安全的写入步数	2000	
	总和	22000	
SUB.QPG	顺控程序容量	46000	49152 字节/12k 步
	运行期间安全的写入步数	2000	
	总和	48000	
文件存储器的容量总计			77824 字节/19k 步

* 一个程序存储器容量以 4096 字节为单位（1k 步）时是安全的。

7. 功能

QCPU 模块功能如下：

7.1 功能表

QCPU 功能如下表：

项目	说明	参考
恒定扫描	在设定的时间间隔内执行程序的功能，与程序扫描时间无关。	参见 7.2 节
锁存功能	在切断电源、复位操作时保存软元件数据的功能。	参见 7.3 节
从 STOP 切换到 RUN 时的输出状态设置	当 QCPU 设置为从 STOP 状态切换至 RUN 状态时设置输出 Y 状态的功能（STOP 前的输出/运算后的输出）。	参见 7.4 节
时钟功能	QCPU 内部时钟功能。	参见 7.5 节
远程操作	远程操作 QCPU 的功能。	参见 7.6 节
远程 RUN/STOP	启动和停止操作 QCPU 的功能。	参见 7.6.1 节
远程 PAUSE	暂停操作 QCPU 的功能。	参见 7.6.2 节
远程 RESET	QCPU 复位功能。	参见 7.6.3 节
远程锁存清除	清除 QCPU 锁存数据的功能。	参见 7.6.4 节
QCPU 按键开关 RUN/STOP 状态和远程操作的关系	说明远程操作时，解释与 QCPU 按键开关设置的 RUN/STOP 状态的关系。	参见 7.6.5 节
改变 QCPU 兼容输入模块的响应时间	改变 QCPU 兼容输入模块响应时间的功能。响应时间为 1ms，5ms，10ms，20ms，70ms。缺省为 10ms。	参见 7.7.1 节
支持 QCPU 的高速输入模块的输入响应时间修改	支持 QCPU 的高速输入模块输入响应时间修改为：0.1ms，0.2ms，0.4ms，0.6ms 和 1ms（缺省：0.2ms）的功能。	参见 7.7.2 节
支持 QCPU 的中断模块输入响应时间修改	支持 QCPU 的中断模块输入响应时间个修改为：0.1ms，0.2ms，0.4ms，0.6ms 和 1ms（缺省：0.2ms）的功能。	参见 7.7.3 节
设置 QCPU 兼容的开关	建立 QCPU 兼容智能型功能模块的功能。 有关建立步骤，参见智能型功能模块手册。	参见 7.8 节
监控功能	用 GX Developer 操作，监控程序状态和 QCPU 的软元件的功能。	参见 7.9 节
设置监控条件	使用 QCPU 精密定时器的监控功能。	参见 7.9.1 节
监控/测试本地软元件	使用 GX Developer 对指定程序的本地软元件进行测试或监控功能。	参见 7.9.2 节
外部 I/O 的 ON/OFF 切换	用 GX Developer 将 QCPU 的 I/O 强制进行 ON/OFF 切换的监控功能。	参见 7.9.3 节
在运行期间写入	当 QCPU 在运行状态时写入程序的功能。	参见 7.10 节
测量执行时间	显示执行程序的处理时间，执行中断程序的次数，和程序执行时间的功能。	参见 7.11 节
程序表监控	显示执行程序处理时间的功能。	参见 7.11.1 节
中断程序监控	显示执行中断程序的次数功能。	参见 7.11.2 节
扫描时间测量	在选择步数之前执行时间测量的功能。	参见 7.11.3 节
采样跟踪功能	在指定的时间从 QCPU 指定的软元件数据中采样的功能。	参见 7.12 节
多用户调试功能	使用多个 GX Developer 能让多个用户调试程序的功能。	参见 7.13 节
警戒定时器	监控因 QCPU 硬件和程序出错产生操作延迟的功能。	参见 7.14 节
自诊断功能	QCPU 检查故障的功能。	参见 7.15 节
故障历史记录	在存储器中保存故障历史和诊断结果的功能。	参见 7.16 节
系统保护	用 GX Developer 通信模块时，防止程序变化的功能。	参见 7.17 节
口令注册	用 GX Developer 访问 QCPU 中保存文件时，提供读/写保护的功能。	参见 7.17.1 节
远程口令	防止用几个通信模块和以太网模块从外部源进行非法访问的功能。	参见 7.17.2 节
系统显示	接通 GX Developer 并监控系统配置的功能。	参见 7.18 节
LED 显示	前面的 LED 指示 QCPU 操作状况的功能。	参见 7.19.节
LED 显示	指示 QCPU 正常或异常操作状况的功能。	参见 7.19.1 节
优先权设置	设置故障优先关闭 LED 显示的功能。	参见 7.19.2 节

7 - 1

7.2 恒定扫描

(1) 什么是恒定扫描?

扫描时间因为处理时间根据是否执行用于顺控程序的不同而相异。
恒定扫描是保持扫描时间为恒定值并重复执行顺控程序的功能。
在执行顺控程序前，执行 I/O 刷新处理。所以即使顺控程序执行时间变化时，I/O 刷新仍保持恒定。

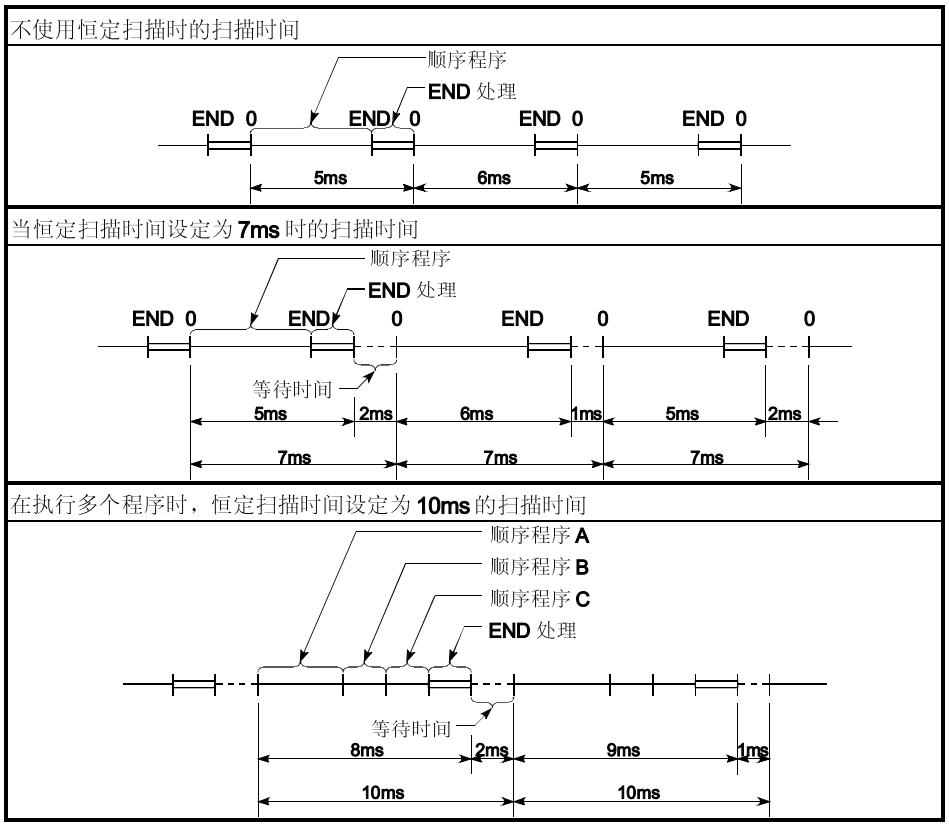


图 7.1 恒定扫描操作

备注

当使用低速执行型程序时，必须设置恒定扫描功能或低速执行型程序执行时间。

(2) 设定恒定扫描时间

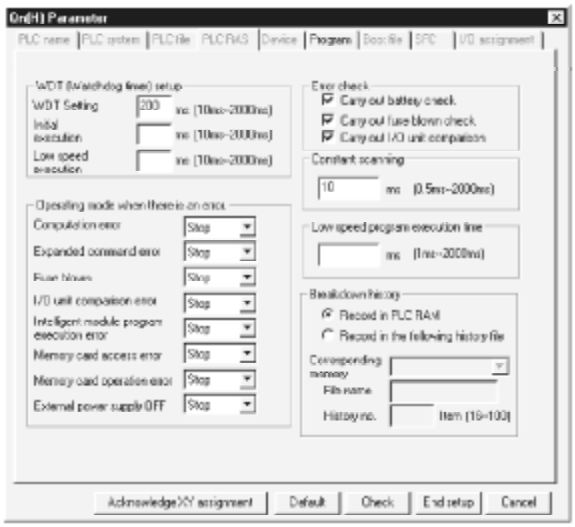
(a) 用参数模式 **PLC RAS** 进行恒定扫描时间的设定。

恒定扫描设定范围为 **0.5ms—2000ms**。

用 **0.5ms** 的模块可进行设定。

- 当执行恒定扫描时，设定恒定扫描时间。
- 当不执行恒定扫描时，恒定扫描时间栏空白。

[例]当恒定扫描设定为 **10ms** 时



(b) 恒定扫描的设定时间要大于顺控程序的最大扫描时间。恒定扫描的设定时间也要小于 **WDT** 的设定时间。

（**WDT** 设定时间）**>**（恒定扫描的设定时间）**>**（顺控程序的最大扫描时间）

如果顺控程序的扫描时间大于恒定扫描设定值，则 **QCPU** 将探测出 **PRG.TIME OVER**（出错代码：**5010**），则恒定扫描设定不起作用，而以扫描时间执行顺控程序。

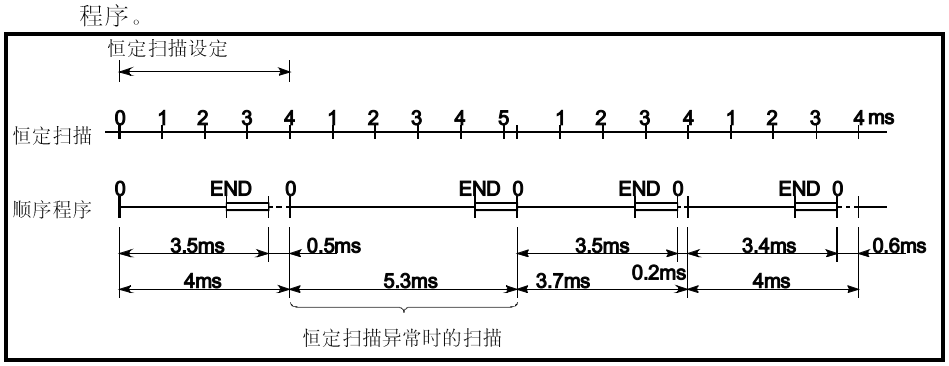


图 **7.2** 当扫描时间大于恒定扫描时的操作

如果恒定扫描值大于 **WDT** 的设定值，**QCPU** 将探测出 **WDT** 错误，停止执行程序。

(c) 从前一次执行 END 处理至下一次扫描开始的等待时间期间，停止处理顺控程序。

1) 如果使用低速执行型程序，程序的执行将中断-0.5ms（恒定扫描时间设定）。

2) 如果在执行 END 处理之后发生中断错误，或存在一个低速执行型程序，将执行中断程序或低速执行型程序。

(d) 恒定扫描时间差
本节阐述恒定扫描时间的设定精度。

1) 当下列程序不执行时，恒定扫描时间设定的盈余部分（等待时间）为：
Q02CPU 0.02ms，Q02HCPU，Q06HCPU，Q12HCPU，Q25HCPU 0.1ms。
- 低速执行型程序
- 中断程序
- 固定扫描执行型程序

2) 当使用低速执行型程序时，等待时间为 0.5ms。如果低速执行型程序的一项指令的最大处理时间为 0.5ms，则恒定扫描的盈余时间与 1）所述相同。如果最大处理时间超过 0.5ms，恒定扫描将延迟到超过的时间。

3) 当执行中断程序/固定扫描执行型程序时，可作中断操作。当执行中断程序/固定扫描执行型程序时，如果恒定扫描超时，恒定扫描无法结束。当使用中断程序/固定扫描执行型程序时，恒定扫描时间可以错位一个中断程序/固定扫描执行型程序的执行时间。

备注

有关命令处理时间，参见“QCPU（Q 模式）/QnACPU 编程手册（公共指令）”。

7.3 锁存功能

(1) 什么是锁存功能

(a) 每个 QCPU 软元件的值都设回到缺省值（位软元件：OFF，字软元件：0）的前提条件为：

- QCPU 电源接通时。
- 执行复位操作时。
- 瞬时掉电超过允许的时间量时。

锁存是当上述状况发生时维持软元件内容的功能。
锁存的有效性并不影响由程序执行的操作。

(b) 即使当瞬时掉电超过允许的时间时，锁存仍可用于继续控制，以维持生产量，缺陷计数和寻址。

(c) 下列软元件能使用锁存功能：
（缺省锁存范围仅为锁存继电器。）

1) 锁存继电器（L）
2) 通信继电器（B）
3) 信号报警器（F）
4) 边沿继电器（V）
5) 定时器（T）
6) 积算定时器（ST）
7) 计算器（C）
8) 数据寄存器（D）
9) 通信寄存器（W）

(2) 锁存范围的设置

用 PLC 参数模式软元件设置来设置锁存范围。
在锁存范围设置中有两种范围类型，锁存清除键（RESET/L.CLR 开关）和远程锁存清除操作，可设置为有效或失效。

(3) 清除锁存范围软元件数据

(a) 清除锁存范围软元件的内容并设定缺省值，执行“锁存清除”。当执行锁存清除时，同时也清除在未锁存范围内软元件的详细内容。

(b) 在锁存清除键变为无效的范围内，用参数设置软元件时，即使执行锁存清除，也不会清除掉软元件。如果选择“锁存（2）：用锁存清除键”禁止清除，用 GX Developer 的远程锁存清除也能够清除影响到的软元件。（有关远程锁存清除，参见第 7.6.4 节。）

注意
用锁存清除不能清除文件寄存器（R）。 （有关清除文件寄存器，参见第 10.7 节。）

(4) 注意事项

(a) 即使软元件指定了锁存功能，如果已指定本地软元件或已指定软元件的初始化，软元件也不能保持锁存。

(b) 电池（Q6BAT）装入 QCPU 模块，可以保存锁存范围软元件的详细内容。

1） 即使顺控程序为使用起见保存在标准 ROM 或存储卡 ROM 中，也必须装上电池。

2） 当 QCPU 模块的电源切断，电池连接器与 CPU 模块连接器断开时，锁存范围内的软元件内容将被破坏。

7.4 从停止状态切换到运行状态时设置输出（Y）状态

(1) 从停止状态切换到运行状态时设置输出（Y）

当从运行状态切换到停止状态时，运行状态的输出（Y）按顺序保存而所有的输出（Y）关闭。

如果从停止状态切换到运行状态，在停止状态之前，QCPU 会提示使用者生成或者清除输出状态。（通过缺省，在停止状态产生之前，输出（Y）状态之后执行程序，同时停止状态切换成运行状态。）

(a) 原先状态

在输出（Y）状态之后，停止状态被输出之前，执行顺控程序计算。

(b) 复算（输出为 1 次扫描之后）

清除所有的输出（Y），在执行顺控程序计算后，输出（Y）被输出。

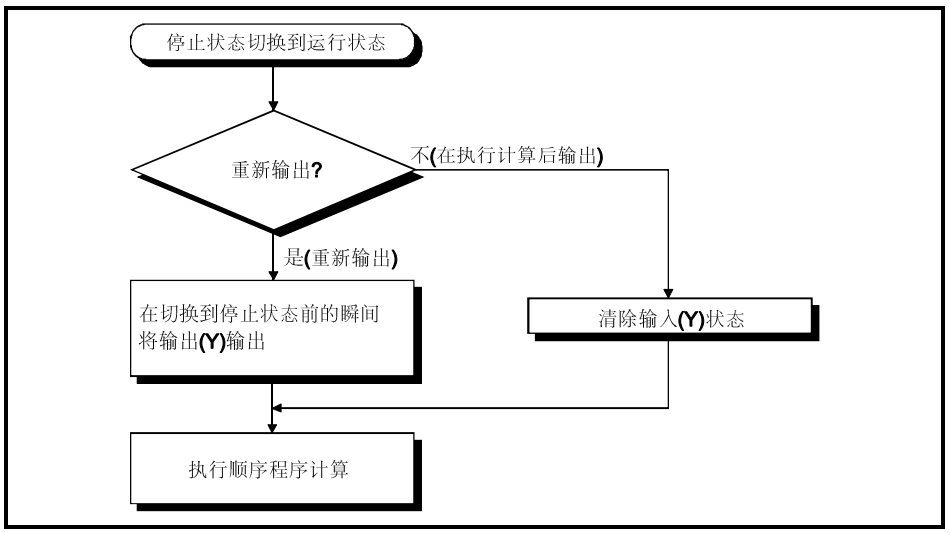
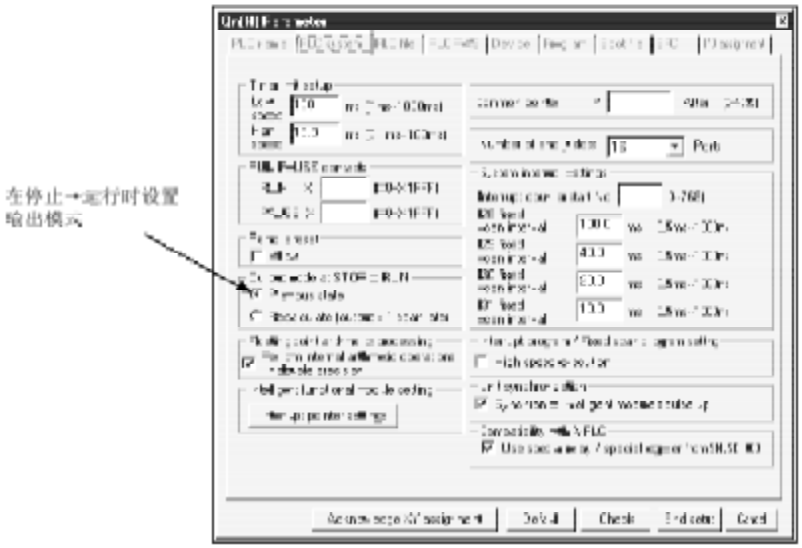


图 7.3　当从停止状态切换到运行状态时的处理

(2) 当从停止状态切换到运行状态时设置输出（Y）状态

当从停止状态切换到运行状态时，在停止状态前，输出（Y）状态可用 **PLC** 参数对话框的 **PLC** 系统表设置。



(3) 注意事项

如果 **QCPU** 处在停止状态时强制接通输出（Y），则即使停止状态切换成运行状态，也不能保持接通状态。

在从停止状态切换到运行状态、在 **PLC** 系统上设置输出模式时，输出状态有效。

7.5 时钟功能

(1) 什么是时钟功能?

(a) 在 CPU 模块中，QCPU 具有时钟功能。

因为由时钟功能而得到的时间数据可以用顺控程序读出，时间数据可用作时间维持。

时间数据也可作为 QCPU 系统功能诸如用于故障历史记录等的时间维持。

即使当 PLC 断电或瞬时掉电时间大于允许时间，使用电池（Q6BAT），即可维持时钟功能的运行。

(b) 时钟数据

时间数据为：年，月，日，小时，分，秒以及星期的数据，用于 QCPU 时钟元件。数据如下：

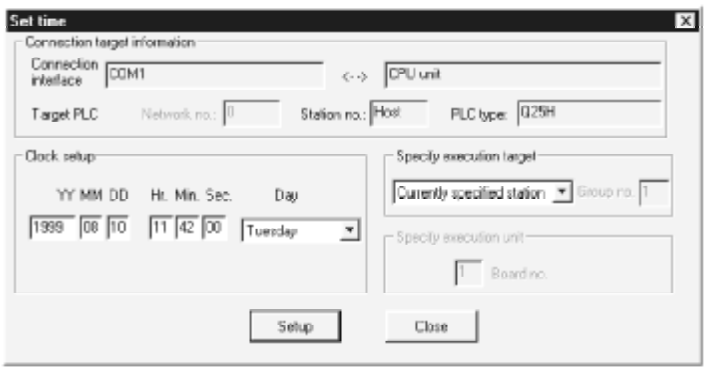
数据名称	内容	
年	公历 4 位数字（从 1980—2079 计数）	
月	1—12	
日	1—31（自动计算闰年）	
时	0—23（24 小时）	
分	0—59	
秒	0—59	
星期	0	星期日
	1	星期一
	2	星期二
	3	星期三
	4	星期四
	5	星期五
	6	星期六

(2) 在时间数据时钟元件中写入和读出

(a) 下列两种方法可用来写入时间数据时钟元件。

1) 用 GX Developer 写入法

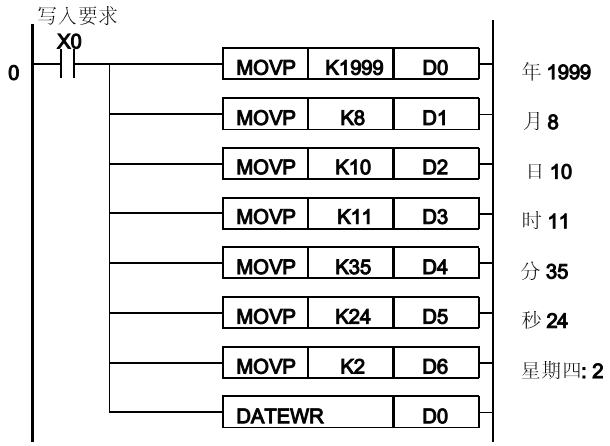
通过显示“在线”→“设置时间”窗口，将时间数据写入时钟元件。



2) 用程序写入法

时间数据也可用时钟指令（DATEWR）写入时钟元件。

下列为用时间数据写入指令（DATEWR）写入时间数据的例子。



有关 DATEWR 指令，详见 QCPU（Q 模式）/QnACPU 编程手册（公共指令）。

(b) 读时间数据

用程序中时间数据读指令（DATERD），将时间数据读至时间寄存器。

下图为用 DATERD 指令读出时钟数据然后保存在 D10 至 D16 中的程序例子。



有关 DATEWR 指令，详见 QCPU（Q 模式）/QnACPU 编程手册（公共指令）。

备注

1) 时间数据的写入和读出可用特殊继电器（SM210—SM213）和特殊寄存器（SD210—SD213）执行。

有关特殊继电器，详见附件 1。有关特殊寄存器，详见附件 2。

2) *: 下列数字为保存在 D10—D16 中的时钟数据。

D10	1999	公历4位数
D11	8	月
D12	10	日
D13	11	小时
D14	35	分
D15	24	秒
D16	2	星期

参见第 7.5.1(1). 节

(3) 注意事项

(a) 开始时时间数据不写入时钟元件。

QCPU 系统将时钟数据用于故障历史记录等。当第一次使用 QCPU 时，为得到精确的时间和日期，必须首先设置时钟。

(b) 即使修正一部分时间数据，也必须将全部数据重新写入时钟。

(c) 如果将失效的时间数据写入时钟，那不能执行正常的时钟操作。

[例]

月 13 日 32

(4) 精度

时间功能的精度随环境温度而变化，如下所示：

环境温度（℃）	精确度（日差，S）
0	-3.18 - +5.25 (TYP.+2.12)
+25	-3.93 - +5.25 (TYP.+1.9)
+55	-14.69 - +3.53 (TYP.-3.67)

(5) 关于 2000 年的闰年问题

(a) 闰年调整

QCPU 时钟功能允许 QCPU 内置时钟元件自动对 2000 年进行跨年度校正。用户不需要对数据和时钟元件的时间设置进行调整。2000 年 2 月 28 日之后显示为 2000 年 2 月 29 日。

(b) 时钟数据比较

为了将 QCPU 的时钟数据和顺控程序相比较，使用 DATERD 指令读时钟数据。年度数据读出为 4 位数字。使用比较指令即可进行比较。

7.6 远程操作

QCPU 在停止状态和运行状态之间提供一个用于切换的 RUN/STOP 开关。
RESET/L.CLR 开关则具有复位和锁存清除功能。
QCPU 允许从外部源(GX Developer 功能，智能型功能模块，和远程接触器控制 QCPU 的操作状态。
下列三种选择可用于远程操作：
- 远程运行/停止
- 远程暂停
- 远程锁存清除

备注

该串行通信模块是用以说明智能型功能模块的例子。

7.6.1 远程运行/停止

(1) 什么是远程运行/停止?

(a) 远程运行/停止是在运行时用 QCPU RUN/STOP 键开关从外部源执行 QCPU 的运行/停止操作。

(b) 使用远程运行/停止功能进行下列远程操作：
1) 当无法到达 QCPU 所在位置时
2) 从外部源对 QCPU 控制板执行运行/停止操作

(c) 在远程运行/停止期间计算
执行远程运行/停止的程序计算如下：
1) 远程停止
将程序执行至 END 指令，并进入停止状态。

2) 远程运行
当使用远程停止处于停止状态时，执行远程运行，则状态改变为运行，并从 0 步开始执行程序。

(2) 远程运行/停止方法

有两种执行远程运行/停止的方法：

(a) 远程运行接触器法

用 PLC 参数模式的 PLC 系统设置来设置远程运行接触器。

可设置的软元件范围为输入 X0—1FFF。

通过切换设置的软元件运行接触器 ON/OFF，执行远程运行/停止。

1) 当远程运行接触器为 OFF，QCPU 进入运行状态。

2) 当远程运行接触器为 ON，QCPU 进入停止状态。

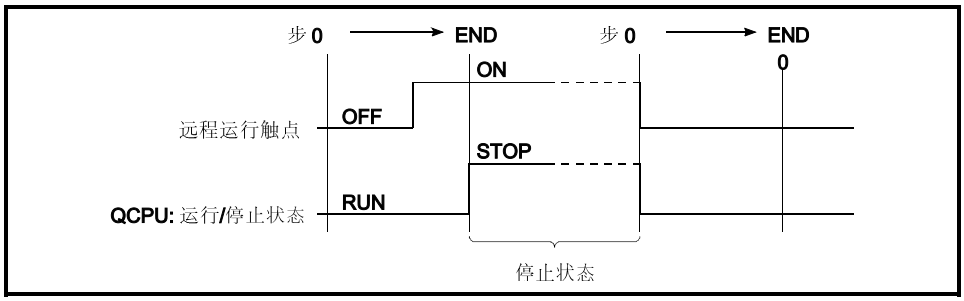


图 7.4　用远程运行接触器执行运行/停止的时序图

(b) 使用 GX Developer 功能，串行通信模块等的方法

可使用 GX Developer 功能、串行通信模块等来执行运行/停止操作，从而执行 QCPU 的运行/停止操作。

GX Developer 操作为在线远程操作。

串行通信模块控制是使用其自己的协议命令来执行的。

有关串行通信模块控制，详见“Q 型 MELSEC 通信协议参考手册”。

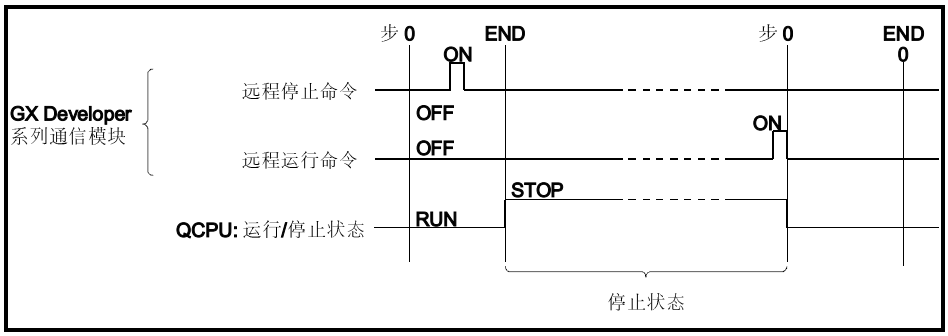


图 7.5　使用 GX Developer，串行通信模块远程运行/停止时间图

(3) 注意事项

(a) 因为在 QCPU 中停止有优先权，因此必须注意下列情况：

1) 当用远程运行接触器，GX Developer 功能，或串行通信模块等执行远程停止时，QCPU 进入停止状态。

2) 当 QCPU 用远程停止设置为停止状态时，所有执行远程停止（远程运行接触器，远程通信模块，等）的外部因素必须设置为运行。

备注

运行/停止状态说明如下：

• 运行状态.....反复执行在顺控程序中 0 步到 END 指令的计算状态。

• 停止状态.....顺控程序计算停止，而输出（Y）全部关闭的状态。

7 - 14

7.6.2 远程暂停

(1) 什么是远程暂停?

(a) 远程暂停是当 CPU 运行/停止键开关在 RUN 位置时从部源执行 QCPU 暂停的功能。

暂停功能停止 QCPU 的计算而同时维持，所有的输出（Y）的 ON/OFF 状态。

(b) 即使在过程控制区域，QCPU 切换到停止状态，也能用来保持输出（Y）接通。

(2) 远程暂停的方法

有两种使用远程暂停的方法：

(a) 远程暂停接触器法

用 GX Developer 功能参数模式的 PLC 系统来设置设置远程暂停接触器。

可设置的软元件范围为输入 X0—1FFF。

1) 使用远程暂停接触器和暂停允许标志（SM206）ON，当对扫描执行 END 处理时，暂停状态接触器（SM204）接通。

2) 当远程暂停接触器断开或 SM206 断开时，暂停状态删除，顺控程序计算从 0 步开始重新执行。

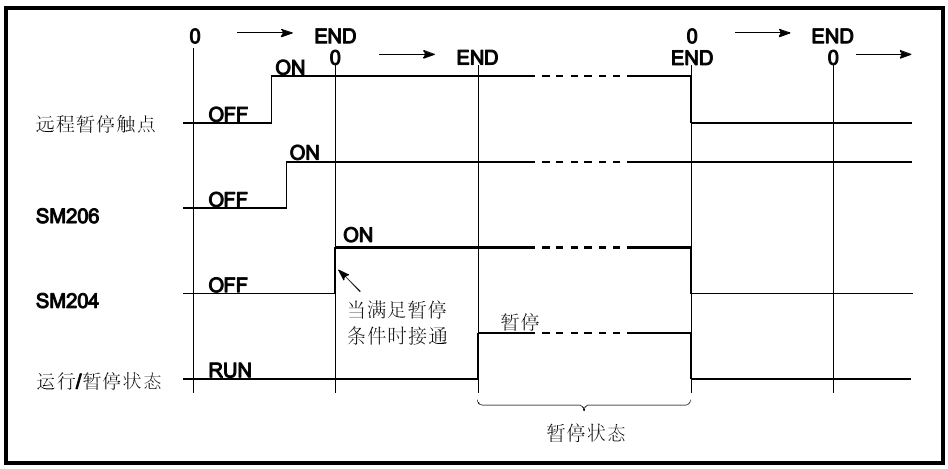


图 7.6　用远程暂停接触器的暂停时序图

(b) GX Developer 功能，串行通信模块法

可用 GX Developer 或串行通信模块执行远程暂停操作。

GX Developer 用在线远程操作执行其功能操作。

串行通信模块使用其本身的协议命令执行控制。

有关串行通信模块控制，详见“Q 型 MELSEC 通信协议参考手册”。

1) 当对接受远程暂停命令的扫描执行 END 处理时，暂停状态接触器（SM204）接通。

在暂停状态接触器接通后对扫描执行 END 处理时，即进入暂停状态，计算停止。

2) 接受远程运行命令后，顺控程序重新从 0 步开始执行计算。

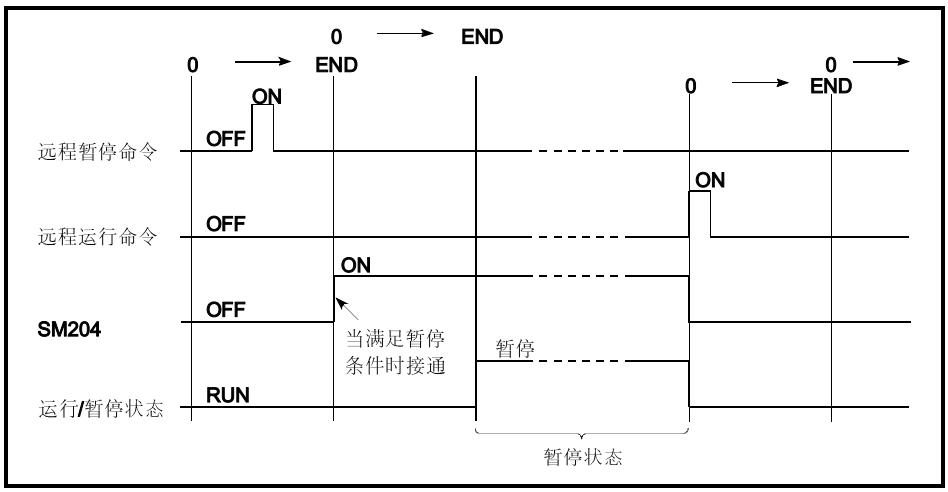
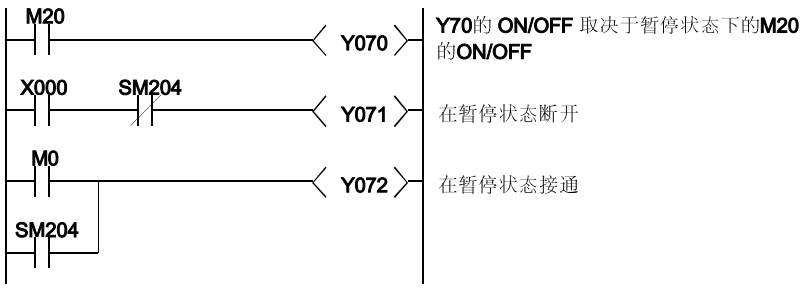


图 7.7 用 GX Developer 功能操作时的暂停时序图

(3) 注意事项

为了在变为暂停状态时设置输出（Y）的 ON/OFF 状态，用暂停状态接触器（SM204）执行互锁。



7.6.3 远程复位

(1) 什么是远程复位?

(a) 远程复位是当 QCPU 处于停止状态时从外部源将 QCPU 复位。
当 QCPU 停止并由自诊断功能探测到出错时，即使 QCPU 运行/停止键开关处于运行状态，也能执行复位操作。

(b) 当发生 QCPU 不能直接操作的错误时，可用远程复位来将 QCPU 复位。
在这种情况下，执行远程 RUN /STOP 操作。

(2) 远程复位的方法

远程复位只能用 GPPW 功能或串行通信模块操作来进行。
按下列步骤执行远程复位：

(a) 在 PLC 参数对话框的 PLC 系统表上，将“远程复位”部分的“允许”复选框接通，然后将参数写入 QCPU。

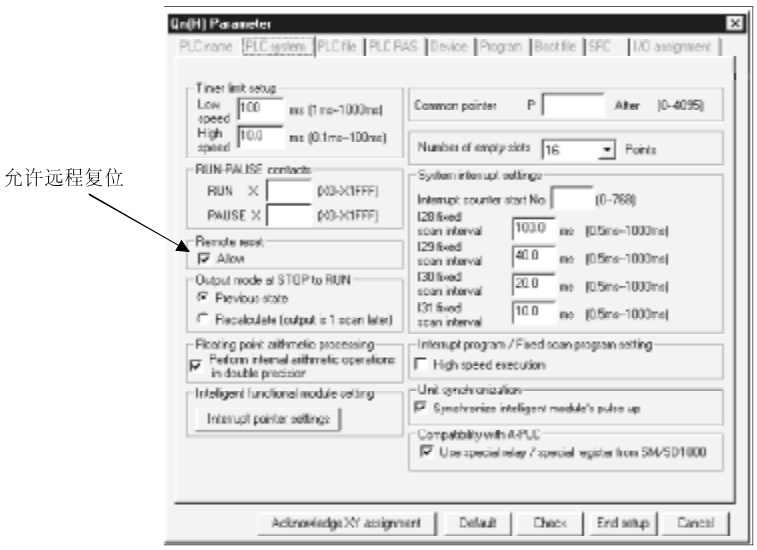
(b) 用远程停止使 QCPU 进入停止状态。

(c) 用远程复位操作将 QCPU 复位。
- 对于 GPPW 功能，执行在线远程操作。
- 对于串行通信模块，是用其本身的协议命令。

有关串行通信模块控制，详见“Q 型 MELSEC 通信协议参考手册”。

(3) 注意事项

(a) 为了执行远程复位，在 PLC 参数对话框的 PLC 系统表上，将“远程复位”部分的“允许”复选框接通，然后将参数写入 QCPU。如果对“允许”复选框未作选择，就不能执行远程复位操作。



(b) 当 QCPU 处于运行状态时，不能执行远程复位。

(c) 在复位操作完成后，QCPU 进入运行/停止键开关状态。
1) 当运行/停止开关在“停止”位置时，QCPU 就进入“停止”状态。
2) 当运行/停止键开关在“运行”位置时，QCPU 就进入“运行”状态。

要点
如果在 QCPU 因为出错而停止时执行远程复位操作，复位完成后 QCPU 即进入运行/停止开关状态。

7 - 18

7.6.4 远程锁存清除

(1) 什么是远程锁存清除

(a) 远程锁存清除为当 QCPU 处于停止状态时，使用 GPPW 功能或其它功能，将锁存在 QCPU 中的软元件数据复位。

(b) 当 QCPU 在下列区域时，使用远程锁存清除。在这种情况下，结合远程运行/停止来执行。

- 当 QCPU 在无法达到的位置时
- 当从外部源进行 QCPU 控制板的锁存清除操作时

(2) 远程锁存清除方法

远程锁存清除只能用 GX Developer 功能或串行通信模块来执行。

按下列步骤执行远程锁存清除：

(a) 用远程停止，使 QCPU 进入停止状态。

(b) 用锁存清除，使 QCPU 进入停止状态。

1) 用在线远程操作执行 GPPW 功能。

2) 串行通信模块控制使用本身的协议命令。

有关串行通信模块控制，详见“Q 型 MELSEC 通信协议参考手册”。

(c) 在远程锁存清除后，为了使 QCPU 返回到运行状态，执行远程运行操作。

(3) 注意事项

(a) 当 QCPU 处于运行状态时，无论是远程锁存清除或是用 RESET/L.CLR 开关的锁存清除，都无法执行。

(b) 用参数模式软元件设置的软元件设置的锁存范围有一个能使锁存清除（RESET/L.CLR 开关）有效或无效的范围。

远程锁存清除操作是与“锁存清除有效/无效”设置无关的复位操作。

(c) 当执行远程锁存清除时，没有锁存的软元件也被清除。

远程锁存清除操作也将保存在 QCPU 存储器里的故障历史记录数据一并清除。

7.6.5 远程操作和 QCPU 运行/停止开关间的关系

(1) 远程操作和 QCPU 运行/停止开关间的关系

远程操作与 RUN/STOP 开关的组合形成的 QCPU 操作状态如下表所示：

远程操作 / 运行/停止开关	运行 *1	停止	暂停 *2	复位 *3	锁存清除
运行	运行	停止	暂停	不能操作*4	不能操作*4
停止	停止	停止	停止	复位*5	锁存清除

*1 当用远程运行接触器执行操作时，必须用参数模式 PLC 系统设置来设置“运行－暂停接触器”。

*2 当用远程暂停接触器执行操作时，必须用参数模式 PLC 系统设置来设置“运行－暂停接触器”。此外，远程暂停启动线圈（SM206）必须设置为 ON。

*3 必须用参数模式 PLC 系统设置来设置“远程复位启动”。

*4 如果用远程操作将 QCPU 切换至停止状态，可执行复位或锁存清除。

*5 包括 QCPU 因出错而停止的情况。

(2) 用同样的 GX Developer 执行远程操作

在用同样的 GX Developer 执行远程操作时，最后执行的远程操作状态有效。

(3) 用多个 GX Developer 执行远程操作

用一个 GX Developer 执行远程操作时，就不能用另一个 GX Developer 执行另一个远程操作。

当一个 GX Developer 执行的远程操作删除后，才能由另一个 GX Developer 执行新的远程操作。

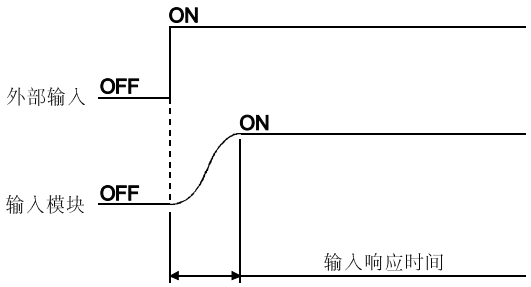
例如，当一个 GX Developer 执行远程暂停操作时，即使另一个 GX Developer 想执行远程停止/远程运行操作，但远程暂停状态仍将继续维持。一个正在执行远程暂停操作的 GX Developer 要执行远程运行操作，则该远程操作即被取消，可由另一个 GX Developer 执行新的远程操作。

7.7 改变 QCPU–兼容的输入模块和中断模块的响应时间（I/O 响应时间）

7.7.1 改变输入模块的响应时间

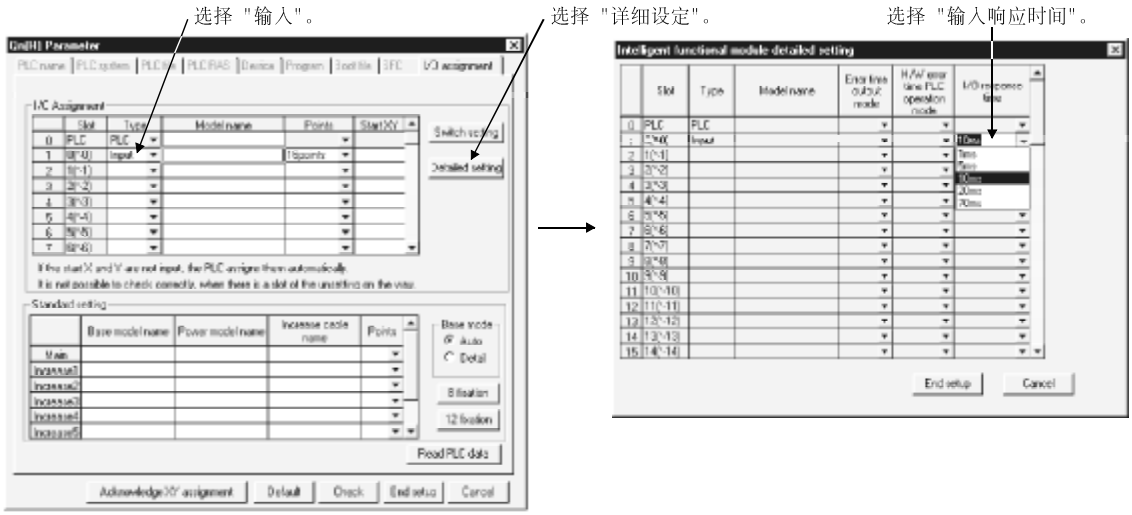
(1) 改变输入响应时间

可将一个 QCPU 兼容的输入模块的响应时间设定为一个期望的响应时间：1ms，5ms，10ms，20ms 或 70ms。在设定的响应时间内输入模块读外部输入。输入响应时间的缺省值为 10ms。



(2) 设定输入响应时间

在 PLC 参数对话框的“I/O 分配”表上，规定期望的输入响应时间。在插槽的“类型”栏里选择输入，以指定期望的输入响应时间。



(3) 反作用

(a) 较高的输入响应时间会产生对受噪音影响的输入作出响应的结果。在设定期望的输入响应时间时要考虑使用输入模块的操作环境。

(b) AnS 系列兼容的输入模块的输入响应时间不能改变。对 AnS 系列兼容的输入模块其插槽不能进行输入响应时间设定。

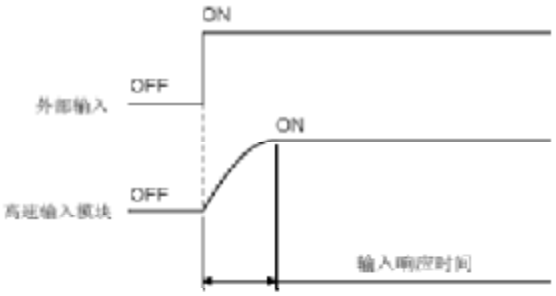
7 - 21

7.7.2 改变高速输入模块的响应时间

(1)“改变高速输入模块的响应时间”含义是什么?

改变高速输入模块的响应时间意为：将支持 **QCPU** 的高速输入模块（**QX40–S1**）的输入响应速度修改为 **0.1ms**，**0.2ms**，**0.4ms**，**0.6ms** 和 **1ms**。

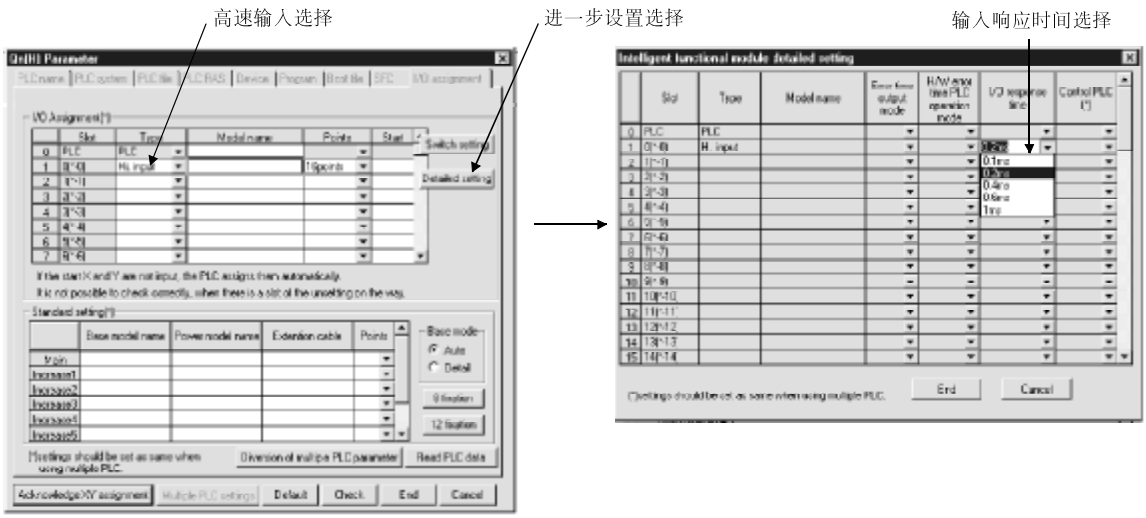
以对高速输入模块设定的输入响应速度接收来自外部源的输入。对输入响应时间的缺省设置为 **0.2ms**。



(2) 设置输入响应时间

用 **PLC** 参数的 **I/O** 分配来设置输入响应时间。

在要设置输入响应时间的插槽类型栏中选择“高速输入”。



(3) 注意事项

(a) 当输入响应时间设置为高速时，系统会反过来受到噪音的影响。在设定输入响应时间时要考虑系统使用的环境。

(b) 当改变高速输入模块的响应时间时，需要使用 **GX Developer**（**SW5D5C–GPPW–E** 或更高）。

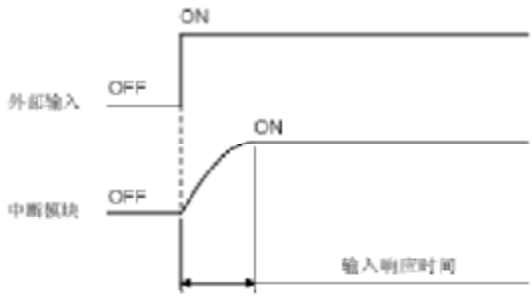
不能用 **GX Developer**（**SW4D5C–GPPW–E**）（固定为 0.2**ms**（缺省设置））修改高速输入模块的输入响应时间。

7.7.3 改变中断模块的响应时间

(1) “改变中断模块的响应时间”含义是什么?

改变高速输入模块的响应时间意为：将支持 **QCPU** 的中断模块（**Q160**）的输入响应速度修改为 **0.1ms**，**0.2ms**，**0.4ms**，**0.6ms** 和 **1ms**。

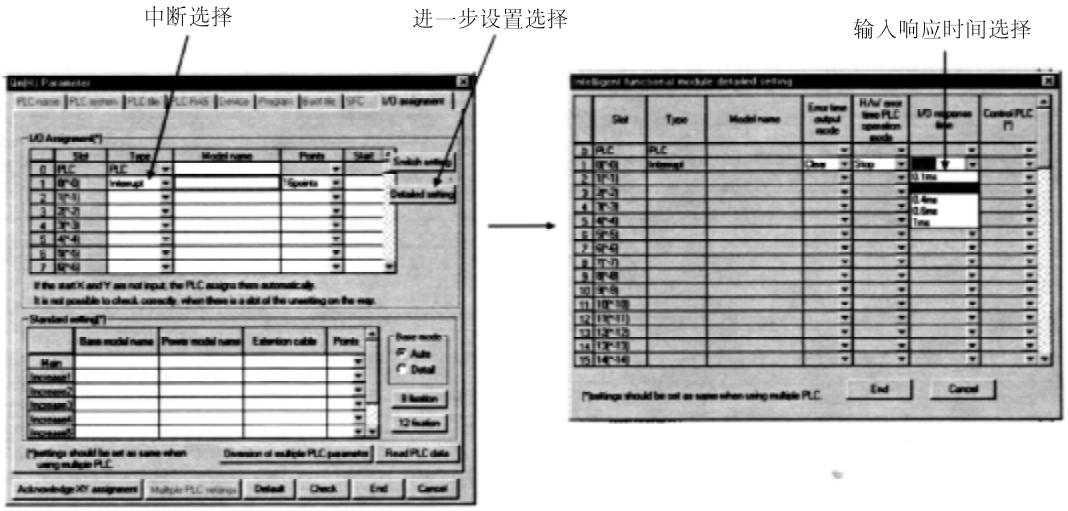
以为中断模块设定输入响应速度接收来自外部源的输入。对输入响应时间的缺省设置为 **0.2ms**。



(2) 设置输入响应时间

用 **PLC** 参数的 **I/O** 分配来设置输入响应时间。

在要设置输入响应时间的插槽类型栏中选择“中断”。



(3) 注意事项

(a) 当输入响应时间设置为高速时，系统会反过来受到噪音的影响。在设定输入响应时间时要考虑系统使用的环境。

(b) 对于 **A1S161**（支持 **AnS** 系列的中断模块），输入响应时间不能修改。即使插槽已设置了中断响应时间，用支持 **AnS** 系列的中断模块也不能执行处理。

(c) 当改变中断模块的响应时间时，需要使用 **GX Developer**（**SW6D5C–GPPW–E** 或更高）。

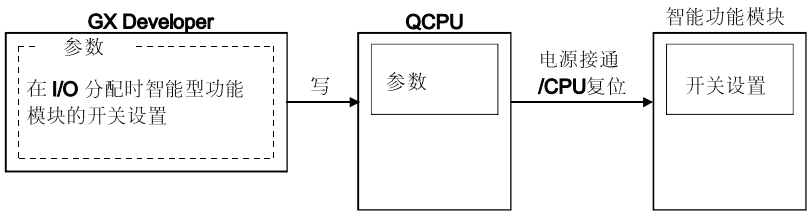
不能用 **SW5D5C–GPPW–E** 或更早期的 **GX Developer**（固定为 **0.2ms[**缺省设置**]**）修改中断模块的输入响应时间。

7.8 智能型功能模块的开关设置

(1) 智能型功能模块的开关设置

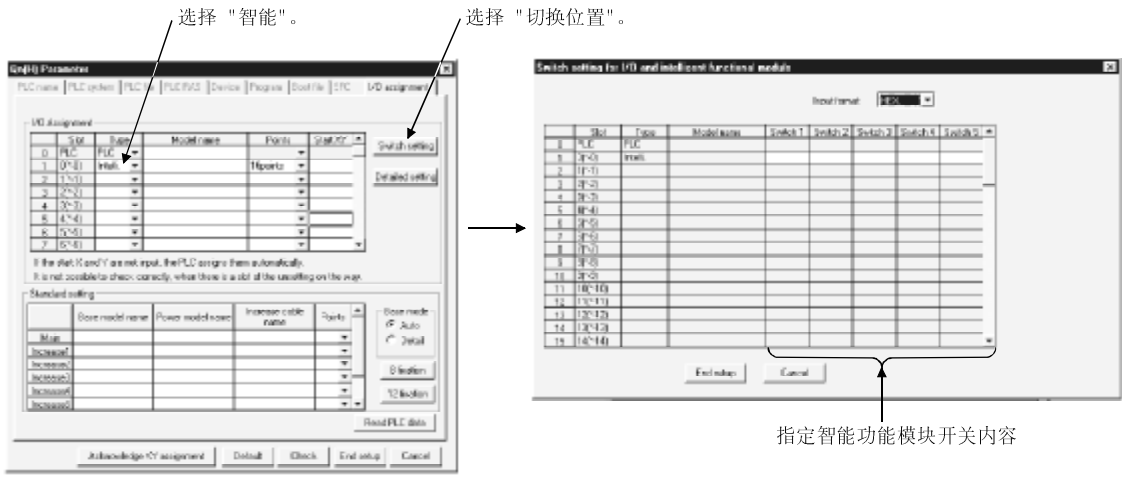
智能型功能模块的开关是用来用 GX Developer 设置 QCPU 兼容的智能型功能模块的开关的。

用 GX Developer 的开关设置在前沿或 QCPU 复位时从 QCPU 写入型每个智能型功能模块。



(2) 智能型功能模块的开关设置

在 PLC 参数对话框“I/O 分配”表上指定期望的开关设置。对要设置智能型功能模块开关的相应插槽的“类型”栏里选择“智能”。



(3) 注意事项

(a) 不得将智能型功能模块的开关设置用于 AnS 系列–兼容的特殊功能模块。
如果把智能型功能模块的开关设置指定给 AnS 系列–兼容的特殊功能模块，将会出错（SP.PARA.ERROR）。

(b) 有关智能型功能模块的开关设置，详见所用智能型功能模块的手册。

(c) 通过设置“中断”类型，用 GX Developer（SW6D5C–GPPW–E 或更高）来设置中断模块的开关。
通过设置“智能”类型，用 SW5D5C–GPPW–E 或更早期的 GX Developer 来设置中断模块开关。
有关中断模块开关设置，详见下列手册。

- 积木式 I/O 模块用户手册。

7.9 监控功能

(1) 什么是监控功能?

(a) 监控功能是一种用 **GX Developer** 读 **QCPU** 程序，软元件和智能型功能模块状态的功能。
QCPU 执行 **END** 处理以完成来自 **GX Developer** 的控制监控请求。**QCPU END** 处理的结果在 **GX Developer** 一侧显示出来。

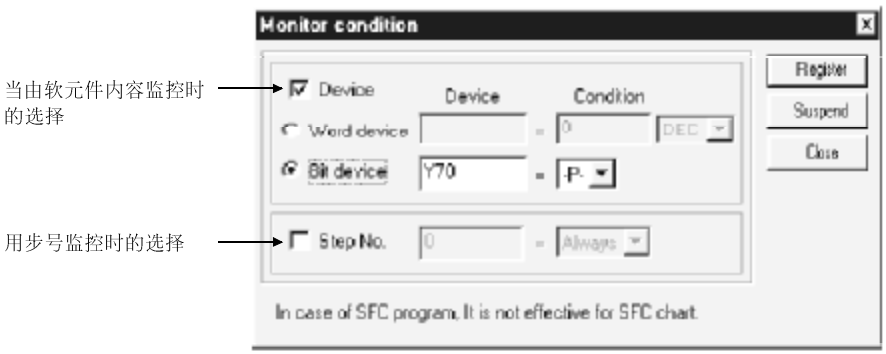
(b) 用 **GX Developer** 设置监控条件，就能在指定的条件下监控 **QCPU** 的操作状态。
也能通过设置监控停止条件，维持指定条件下的监控状态。

(c) 使用本地软元件以执行多个程序就可监控本地软元件数据。

7.9.1 监控条件设置

(1) 监控回路时设置监控执行条件。

选择“在线”→“监控”→“监控条件”，以打开监控条件对话框。下例所示为在 **Y70** 前沿开始监控操作。



(a) 仅指定步号时。

1） 指定的步数执行前瞬间的状态变为“指定状态”时，即为监控数据采样开始时刻。

2） 执行状态的指定方法表示如下:

a）	当从不执行状态改变为执行状态时	: **<–P–>**
b）	当从执行状态改变为不执行状态时	: **<–F–>**
c）	当仅为始终执行时	: **<–ON–>**
d）	当仅为始终不执行时	: **<–OFF–>**
e）	始终与状态无关	: **<–Always–>**

备注

当指定“步数为**[0]**”时，设置条件为“始终”。

7 - 25

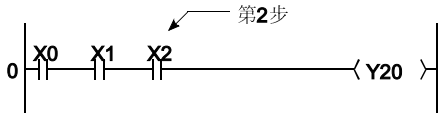
要点

如果在 **AND/OR** 程序块之间，指定步为监控条件，在程序块之间的指定步的原先状态改变为由 **LD** 指令指定的状态后，监控数据开始采样。监控开始时刻取决于指定为监控条件的步。下例所示为第 **2** 步 **ON** 时的监控。

(1) 当第 **2** 步连接到 **AND** 指令时：

如下所示，当“**X0**”和“**X1**”都为 **ON** 时，监控执行条件建立。

回路模式

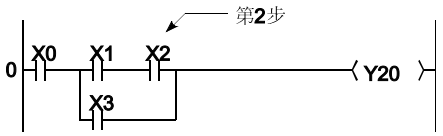


模式表

```
0 LD   X0
1 AND  X1
2 AND  X2
3 OUT  Y20
```

(2) 当第 **2** 步连接到 **AND/ON** 程序块之间时：

如下所示，当“**X1**”为 **ON** 时，监控执行条件建立。“**X**”为 **ON** 或 **OFF**，都不影响监控执行条件。




```
0 LD   X0
1 LD   X1
2 AND  X2
3 OR   X3
4 ANB
5 OUT  Y20
```

(b) 当仅指定“软元件”时：

可指定“字软元件”或“位软元件”。

1） 当选择“字软元件”时：

当指定的字软元件当前值变为指定值时，监控数据采样开始。

打印当前值为十进制或十六进制数。

2） 当选择“位软元件”时：

当指定的位软元件执行值变为指定值时，监控数据采样开始。

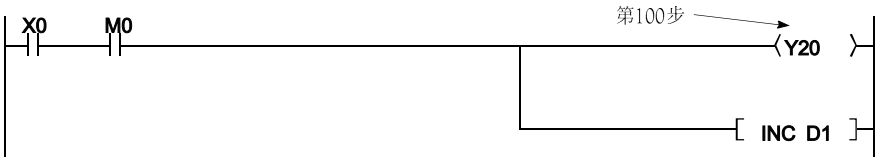
无论是前沿或延迟都可指定为执行条件。

(c) 当选择“步数”和“软元件”时：

在执行指定步之前瞬间的状态和指定的位软元件（字软元件）的执行状态变为指定状态时，监控数据采样开始。

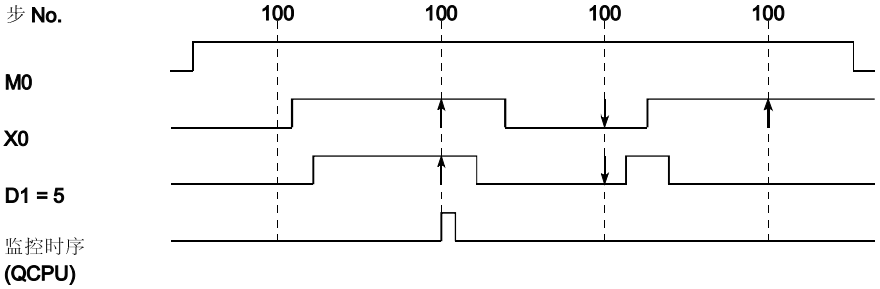
要点

在下列回路中，当“步 No.[100]=<↑>，字软元件[D1]=[K5]”指定为详细条件，在第 100 步，D1=5 的前沿，监控执行条件成立。



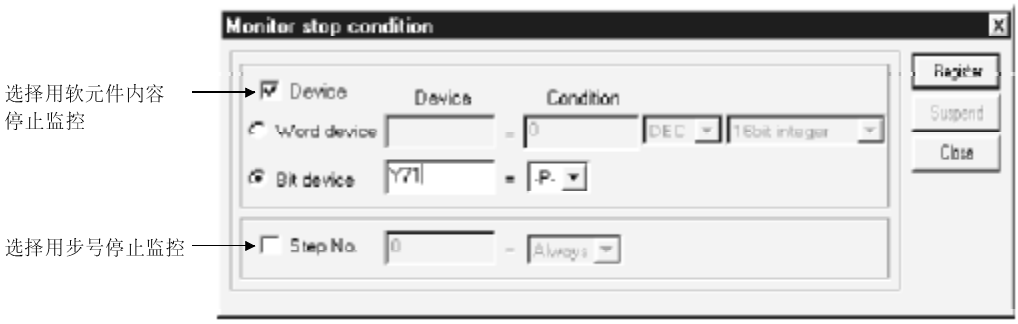
GPPW 的监控时间间隔取决于 GPPW 的处理速度。

当在 GPPW 监控时间间隔内监控条件成立时，即使在比监控时间间隔更短的间隔中监控条件成立，也将执行监控。



(2) 建立监控停止条件

选择“在线”→“监控”→“监控停止条件”，以打开监控停止条件对话框。下例所示为在 Y70 前沿停止监控操作。



(a) 当指定“步号”时：

1） 当在监控条件中指定的步号由执行状态变为指定状态时，监控停止。

2） 下列为执行状态的指定方法：

a） 当从不执行状态改变为执行状态时 ：<–P–>
b） 当从执行状态改变为不执行状态时 ：<–F–>
c） 当仅为始终执行时 ：<–ON–>
d） 当仅为始终不执行时 ：<–OFF–>
e） 始终与状态无关 ：<–Always–>

3） 当不指定“步号”时，在 QCPU END 处理后，监控停止。

7 - 27

(b) 当指定“软元件”时：
“字软元件”或“位软元件”都能指定。
1） 当选择“字软元件”时：
当指定的字软元件当前值变为指定值时，监控停止。
当前值可用十进制数字，十六进制数字，16 位积分数，32 位积分数或实数表示。
2） 当选择“位软元件”时：
当指定的位软元件从执行值变为指定值时，监控停止。
无论是前沿还是后沿都可指定为执行条件。

(3) 注意事项

(a) 在设置监控条件后，用 GX Developer 显示的文件即被监控。通过用 GX Developer 执行“新 PLC 读”和文件名匹配要监控的文件。

(b) 监控文件寄存器时如果未指定文件寄存器，即显示“0”。

(c) QCPU 和 GX Developer 的软元件分配必须匹配方能执行监控。

(d) 当监控智能型功能模块的缓冲存储器时，正如执行 FROM/TO 指令那样扫描时间较长。

(e) 多个用户可同时执行监控。
当多个用户同时执行监控时，注意下列情况：
- 在内置存储器格式化期间，对每个站的监控文件而言，在系统区域增加 1k 步，可执行高速监控。
可有 15 个站设置站监控文件，但程序的空间将减少。
- 用于监控的详细设置条件只能为一个用户设置。

(f) 监控详细条件设定仅能设置为回路监控。

(g) 当同一软元件指定为监控条件和监控停止条件，要指定“ON”或“OFF”。

(h) 当“步数”指定为监控条件时，除非执行下列指定的步指令，否则不能建立监控条件。
1） 当用 CJ 命令，SCJ 命令和 JMP 命令指定跳越步时。
2） 当指定的步是 END 命令时，而程序运行时，存在 FEND 命令，则 END 命令不执行。

(i) 当监控条件已注册时，QCPU 不得复位。

7.9.2 对本地软元件的监控测试

(1) 监控和测试本地软元件

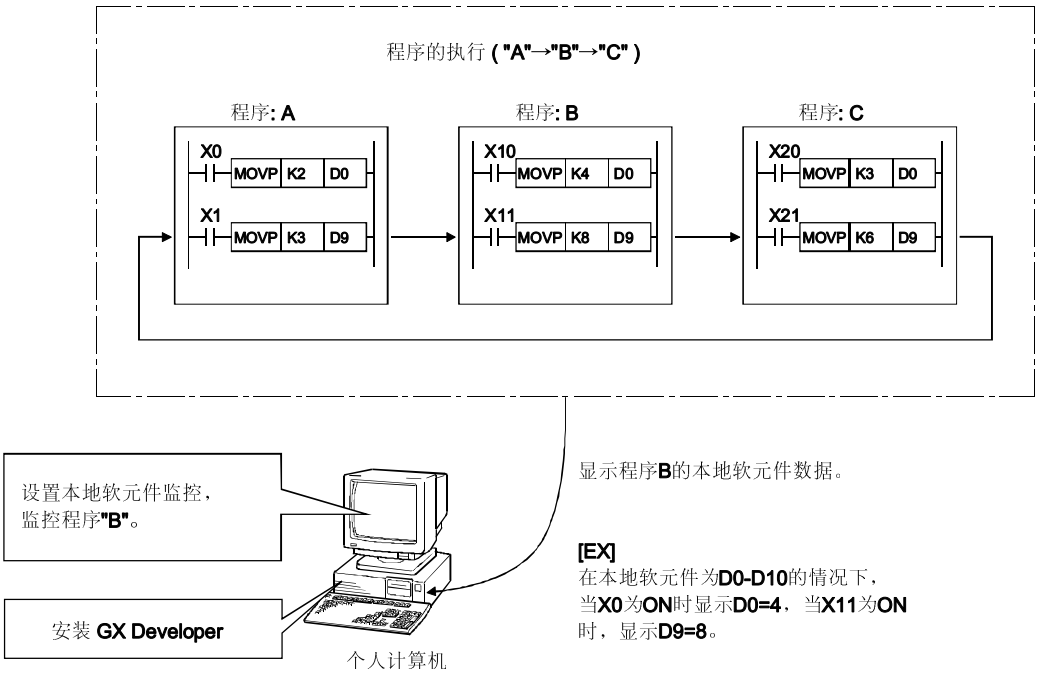
(a) 在 PLC 参数对话框的软元件表中所指定的本地软元件可用 GX Developer 操作进行监控或测试。该功能可用于调试程序和在用 GX Developer 监控程序时监控本地软元件。

(b) 监控本地软元件

下表所示为由 QCPU 执行的三个程序 A，B 和 C 的状态，指定本地软元件为 D0—D99。假定这三个程序按 A→B→C（END 处理）→A→B...的顺序执行。

	监控软元件	
	D0	D100
当指定本地软元件时	在程序 C 中的 D0 被监控。	在程序 C 被执行后，D100 被监控。
当未指定本地软元件时	显示出的程序中的 D0 被监控。	在显示的程序被执行后，D100 被监控。

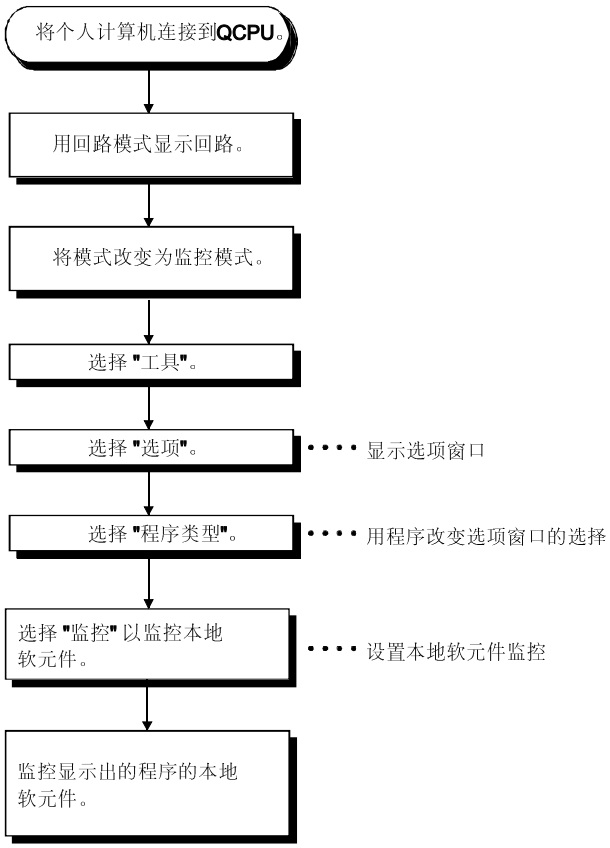
例如，如果已设置本地软元件，而程序 B 显示，就能监控的程序 B 中本地软元件。



7 - 29

(2) 监控本地软元件

监控本地软元件步骤如下：



(3) 注意事项

(a) 用 GX Developer 操作可以进行本地软元件的监控和测试的只是一个单一程序。单一 GX Developer 不能操作多个程序的本地软元件的监控和测试。

(b) 连接到 QCPU 的 RS–232C 串行通信模块的多个 GX Developer 最多可监控或测试 16 个程序的本地软元件。

(c) 如果监控待机型程序中的本地软元件，因为需读和保存本地软元件的数据，因此扫描时间有时会延长。详见第 10.13.1 节。

(d) 对固定扫描执行型程序中的本地软元件不能监控和测试。

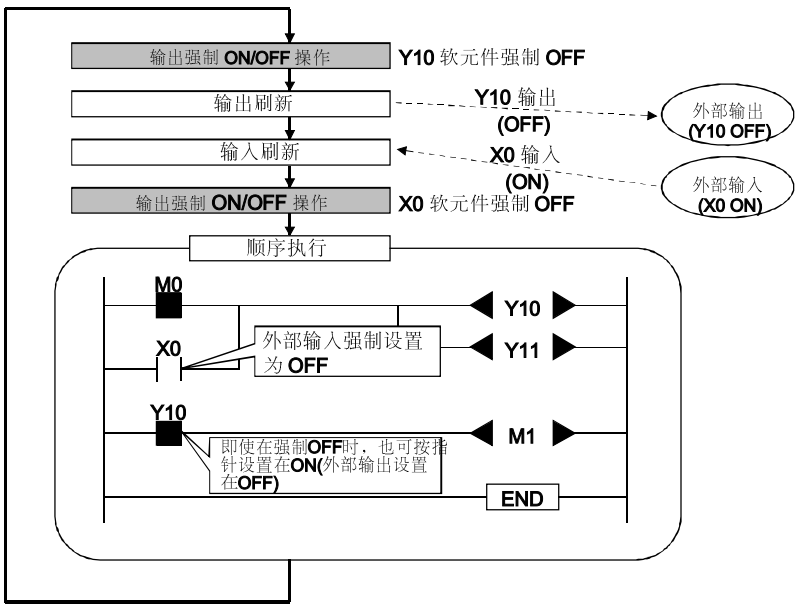
7.9.3 对外部 I/O 强制 ON/OFF

用 GX Developer 执行的强制 ON/OFF 操作将强制性地切换外部 I/O 的接通和断开。
用 GX Developer 的操作将删除 ON/OFF 注册信息。
使用本功能时需要用 GX Developer（SW6D5C–GPPW–E 或更高）。

使用强制 ON/OFF 功能，可执行强制 ON（强制 ON 注册），强制 OFF（强制 OFF 注册），和删除强制 ON/OFF（删除注册）。
下表所示为执行强制 ON，强制 OFF 和强制删除 ON/OFF 的操作。

操作	输入（X）操作	输出（Y）操作
在删除时（无操作）	用外部输入执行顺控程序操作。	向外部输出顺控程序操作结果。
在强制 ON 时	在强制 ON 状态执行顺控程序操作。	向外部输出 ON，与顺控程序操作结果无关。
在强制 OFF 时	在强制 OFF 状态执行顺控程序操作。	向外部输出 OFF，与顺控程序操作结果无关。

当执行强制 ON/OFF 时的操作如下图所示。



(1) 对指定的说明

(a) 执行强制 ON/OFF 与 QCPU 的运行/停止状态无关，但对输入而言，仅在停止出错时方可执行强制 ON/OFF。输出仅对软元件 Y 执行。

(b) 可以注册的软元件为：输入：X0~X1FFF，输出：Y0~Y1FFF。

(c) 适合于强制 ON/OFF 的输入和输出如下所列：

1） 安装于基板上的模块的输入（X）和输出（Y）。

2） MELSECNET/H 模块 LX/LY 被刷新的 QCPU 的输入和输出（X/Y）。

3） CC–link RX/RY 被刷新的 QCPU 的输入和输出（X/Y）。
对未被上述的刷新范围（例如空闲插槽）所覆盖的软元件执行强制 ON/OFF 注册时，仅有 QCPU 软元件存储器被设置为 ON/OFF，而且不是向外输出。

(d) 删除 ON/OFF 注册信息

1） 用 GX Developer 操作可删除 ON/OFF 注册信息。在假定 ON/OFF 注册信息被删除的状态下，执行软元件的强制 ON/OFF。

强制 ON/OFF 的软元件		用顺控程序执行 ON/OFF	不用顺控程序执行 ON/OFF
输入	从装入基板的模块输入	假定从模块接收 ON/OFF 状态。	
	从 MELSECNET/H 模块 LX 要刷新的 QCPU 输入	假定由 MELSECNET/H 刷新的 ON/OFF 状态。	
	从 CC–Link RX 要刷新的 QCPU 输入	假定由 CC–Link 刷新的 ON/OFF 状态。	
输出	从装入基板的模块输出	输出顺控程序操作的结果。	输出为 OFF。
	MELSECNET/H 模块 LX 要刷新的 QCPU 的输出	输出顺控程序操作的结果。	输出为 OFF。
	从 CC–Link RX 要刷新的 QCPU 的输出	输出顺控程序输出的结果。	输出为 OFF。
	输出未覆盖上述的（超出刷新范围）	输出顺控程序输出的结果。	假定 OFF 状态。

2） 用下列操作清除强制 ON/OFF 设置。

- 电源从 OFF→ON
- QCPU RESET/L.CLR 开关复位
- 用远程复位操作复位

(e) 下表所示为外部 I/O 强制 ON/OFF 的时间。

刷新区域	输入	输出
装入基板的 I/O 模块（X，Y）	• 在 END 处理时（输入刷新） • 在执行使用直接访问输入（DX）的命令时（LD，LDI，AND，ANI，OR，ORI，LDP，LDF，ANDP，ANDF，ORP，ORF）	• 在执行使用直接访问输出（DY）的命令时（OUT，SET，DELTA，RST，PLS，PLF，FF，LDF，MC）
MELSECNET/H（LX，LY）	• 在 END 处理时（MELSECNET/H 刷新） • 在执行 COM 命令时 • 在执行 ZCOM 命令时	
CC–Link（RX，RY）	• 在 END 处理时（CC–Link 刷新） • 在执行 COM 命令时 • 在执行 ZCOM 命令时	

(f) 总数为 32 个软元件可用于强制 ON 和 OFF 的注册。

(g) 如使用输入 Y 接触器，顺控程序操作优先执行。

(h) 用 GX Developer 可确认强制 ON，OFF 以及删除状态（包括未建立的）。当至少有一个模块注册时，用模式判断 LED 也可确认。（模式判断 LED 闪烁（ON：200ms，OFF：200ms）

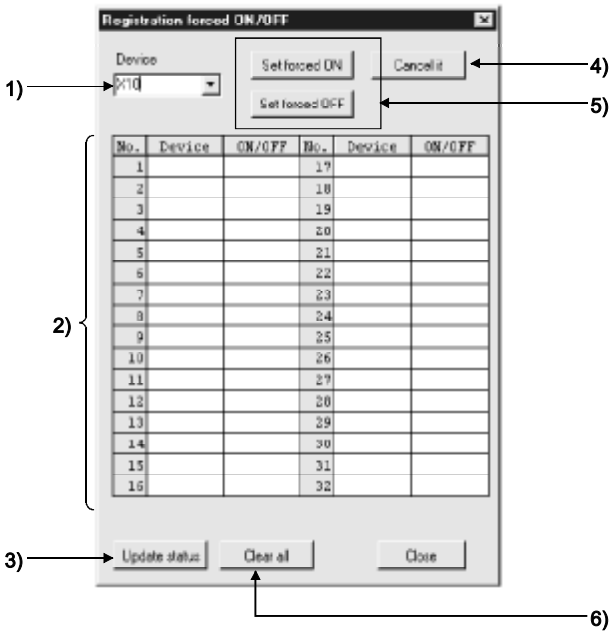
(i) 用多个 GX Developer 连接到网络时，在同一 QCPU 上也可为外部 I/O 注册强制 ON/OFF。
但是用多个 GX Developer 对同一软元件注册强制 ON/OFF 时，要假定最近的注册 ON/OFF 状态。
由于这一原因， GX Developer 第一次执行时显示相对于 QCPU ON/OFF 信息的不同的 ON/OFF 信息。当用多个 GX Developer 执行强制 ON/OFF 时，要确保在执行强制 ON/OFF 步骤之前，必须用“加载注册状态”开关设置最新信息。

(2) 操作步骤

操作步骤说明如下。

(a) 对特殊软元件注册强制 ON/OFF。
[在线]→[调试]→[强制 I/O 注册/删除]

在[强制 I/O 注册/删除]设置屏幕上指定软元件之后，选择[强制 ON 注册]或[强制 OFF 注册]，可以对指定的软元件执行强制 ON 和强制 OFF。



(b) 所提供的设置字段说明如下。

序号	设置字段名称	功能说明
1）	软元件	对要设置的强制 ON/OFF 输入 I/O 地址，或对要删除的强制 ON/OFF 输入 I/O 地址。
2）	注册状态显示区	对注册的强制输入和输出显示注册状态。
3）	加载注册状态	显示从 QCPU 加载的注册状态。
4）	强制 ON/OFF 注册	对特殊软元件执行强制 ON/OFF 注册。
5）	注册删除	对注册的软元件删除强制 ON/OFF。
6）	成批注册删除	删除所有强制 I/O 注册。

7.10 在 QCPU 运行期间写入程序

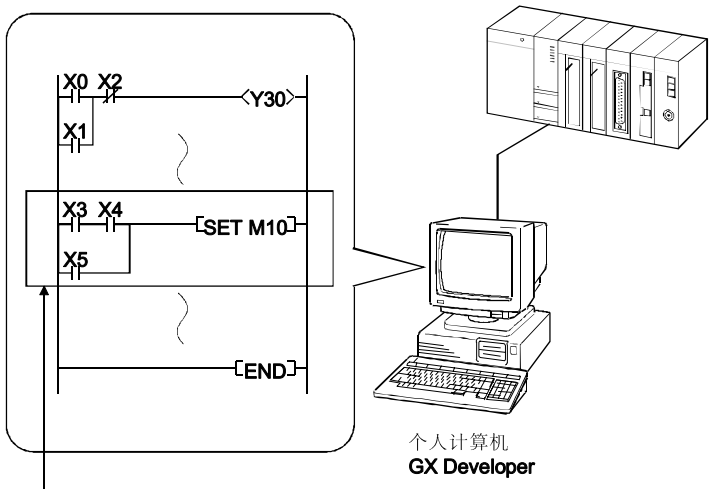
当 QCPU 处于运行状态时，可通过下列任一步骤写入程序或文件：
- 在运行状态时，将数据写入回路模式。
- 在运行状态时，使用指针写入数据（参见 7.13.2 节）
- 在运行状态时，写入成批文件。

7.10.1 运行状态下用回路模式写入数据

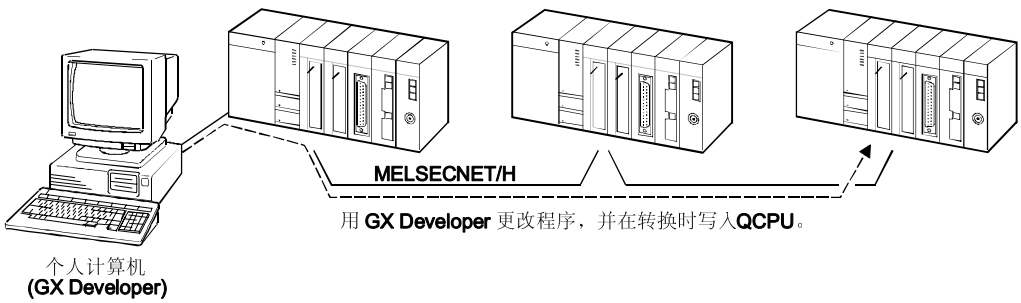
(1)运行状态下用回路模式写入数据

(a) 在运行状态时，用回路模式写数据，用于 QCPU 处于运行状态时写入程序。

(b) 在运行状态时，用回路模式写数据的方法，能够不停止 QCPU 程序的处理而更改程序。



(c) 用 GX Developer 功能连接到网络中的另一站点的外围设备，可以在运行时写入程序。



(2) 注意事项

运行状态下写入时要注意下列几点：

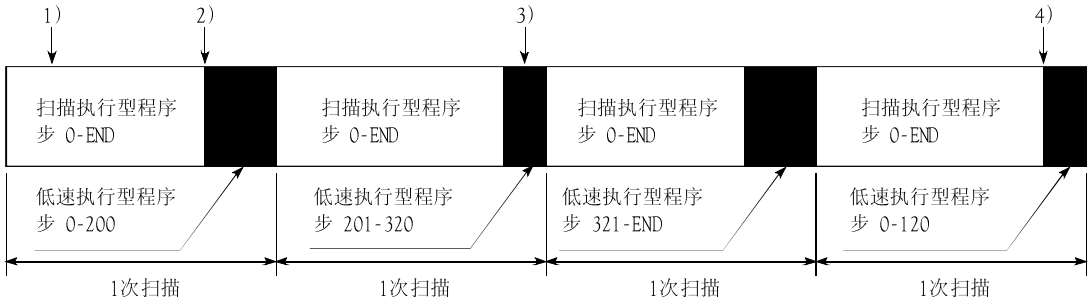
(a)运行状态下可以写入的存储器只有程序存储器。

1) 如果在从存储卡（**RAM**）引导程序时执行运行状态下写入，则被引导的程序将会改变。引导需一定时间，直至运行状态下的写入完成。

2) 如果是在从标准 **RAM** 或存储卡（**ROM**）中引导程序时执行运行状态下写入，被引导的程序不会改变。

(b) 在运行时每次写入最大为 **512** 步。

(c) 在执行低速执行型程序时，一旦低速执行型程序完成，运行写入马上开始。在运行写入时，低速执行也暂时停止。



1): 扫描执行型程序的运行写入命令
2): 扫描执行型程序的运行写入执行
3): 低速执行型程序的运行写入命令
4): 低速执行型程序的运行写入执行

(d) 如果运行时写入是在执行 **PLOAD**，**PUNLOAD** 或 **PSWAP** 指令时执行，为了运行时写入，处理将进入待机状态。如果在执行 **PLOAD**，**PUNLOAD** 或 **PSWAP** 指令时进行运行时写入，指令将延迟到运行时写入完成时。

(e) QCPU 程序文件的容量为所创建的程序容量和用于运行时写入的步数之和。当程序文件的容量增加时，执行运行时写入，如果程序文件的容量大于以前的容量，为了在运行时写入，可以指定步数。这就表示，仅当在用户存储器区域有足够的空间时，才能执行运行时写入。如果在执行运行时写入时步数重新指定，扫描时间可能延长至某个值。如下表所示，控制将中断一个如下表所示的时间值。

CPU 型号	运行时写入步数	
	如果不改变	如果重新指定
QnCPU	最大 0.963ms	最大 0.042ms
QnHCPU	最大 30ms	最大 90ms

7.10.2 运行状态下写入批文件

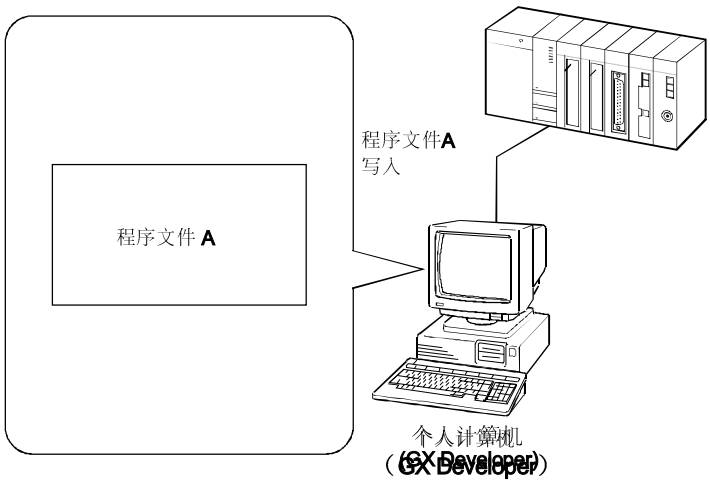
(1) 运行状态下的文件写入

(a) 运行时的文件写入功能用于将批文件写入 QCPU，如下表所示。

存储器名称	QCPU 内置			存储卡（RAM）	存储卡（ROM）	
	程序存储器	标准 RAM	标准 ROM	SRAM 卡	闪存卡	ATA 卡
参数	×	×	×	×	×	×
智能参数	×	×	×	×	×	×
程序	○	×	×	○	×	○
软元件注释	○	×	×	△	×	△
软元件初始值	×	×	×	×	×	×
文件寄存器	×	△	×	△	×	×
本地软元件	×	×	×	×	×	×
调试数据	×	×	×	×	×	×
SFC 跟踪数据	×	×	×	×	×	×
故障历史记录数据	×	×	×	×	×	×
PLC 用户数据	×	×	×	×	×	×

○：可写入数据， ×：不可写入数据

△：用顺控程序访问时可写入数据



注意
运行时允许写入三类文件： • 程序：程序存储器，SRAM 卡，ATA 卡 • 软元件注释：程序存储器，SRAM 卡，ATA 卡 • 文件寄存器：标准 RAM，SRAM 卡 当 QCPU 处于运行状态时，其它任何文件都不能写入。

7 - 37

(2) 注意事项

运行时文件写入的注意事项如下。

(a) 当下列任何条件得到满足时，即可执行运行时的文件写入。在运行时 SFC 程序不允许写入批文件。

1) 程序存储器
- 有可用的连续空间。
- 有可用空间。

2) 存储卡
- 有可用空间。

(b) 在运行时执行文件写入，要注意扫描时间可能会延长，如下表所示。控制会中断一个下表所列的时间值。

事件	QnCPU	QnHCPU
当在程序存储器中有可用的连续空间时（ms）	最大 80ms	最大 300ms
当在程序存储器中有可用空间时（ms）	最大 80ms	最大 300ms
当在存储卡中有可用空间时（除 ATA 卡外）（ms）	最大 120ms	最大 570ms

在使用 ATA 卡时，30k 步的扫描时间为 1.25 秒。

(c) 注意当 QCPU 处于运行状态下的写入批文件时，不能用顺控程序的指令进行的访问。当运行执行文件写入时，不能执行访问文件的指令。

(d) 当 QCPU 处于运行状态时，如果写入执行的文件。下列命令不能正确地执行。写入完成后，只有当执行条件为 OFF 时，才能执行下降沿指令。

- LDF
- ANDF
- ORF
- MEF
- PLF

7 - 38

7.11 执行时间的测量

本功能对执行程序的处理时间进行显示。

本功能用于发现每个程序处理时间对总扫描时间的影响。

共有三种功能用于执行时间的测量。每种功能详见 **7.11.1—7.11.3**。
- 程序监控表
- 中断程序监控表
- 扫描时间测量

7.11.1 程序监控表

(1) 什么是程序监控表?

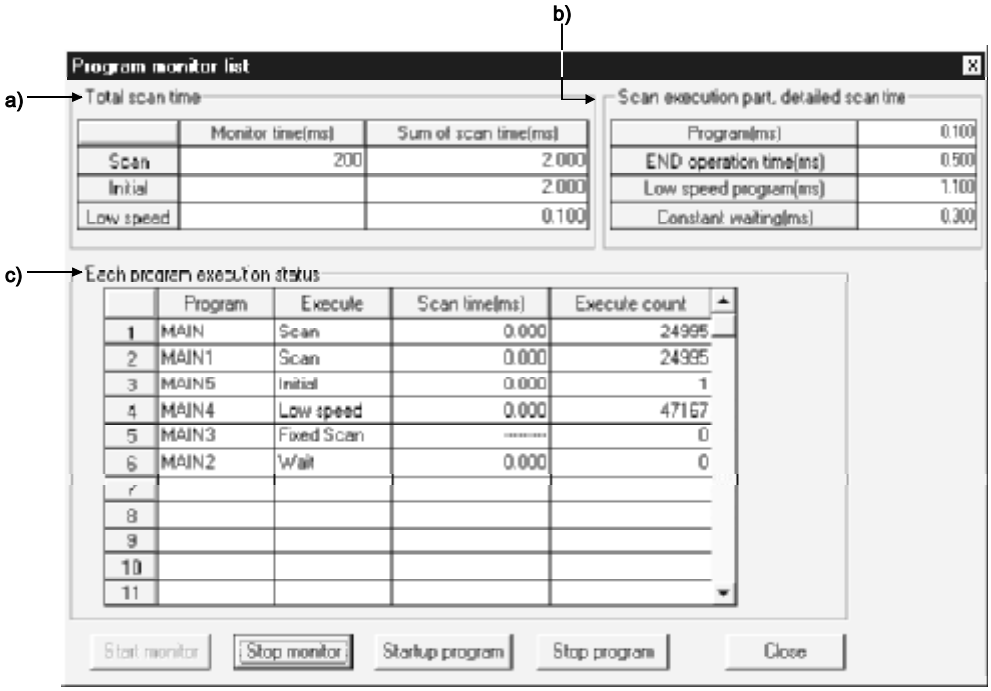
(a) 这是对执行程序的处理时间进行显示的一种功能。

(b) 每个程序的扫描时间，执行次数和处理时间可逐项显示。

(2) 使用程序监控表

(a) 选择“在线”→“监控”→“程序监控表”。屏幕上出现程序监控表对话框。

(b) 每个程序的扫描时间，执行次数和处理时间可逐项显示。



a) “总扫描时间”
用参数模式在“PLC RAS 设置”的警戒定时器（WDT）上设置的监控时间，显示每种程序的总扫描时间。
1) “监控时间”
显示扫描执行型程序、初始化程序、和低速执行型程序的监控时间。
如果扫描时间超过这一时间，QCPU 显示警戒定时器出错。
2) “总扫描时间”
显示“扫描执行的扫描时间的详细内容”所表述的各项的时间之和。
“恒定”表示当设置恒定扫描时的恒定扫描等待时间。

b) “扫描执行的扫描时间的详细内容”
显示扫描时间的详细内容。
1) “程序”
显示扫描执行型程序总执行时间。
2) “END 操作时间”
显示 END 操作时间。
3) “低速执行型程序”
表示对低速执行型程序的执行时间或恒定扫描时间进行设置时，低速执行型程序的总执行时间。
4) “恒定等待”
设定恒定扫描时间时显示恒定扫描待机。但是，当设置低速执行型程序的执行时间时，该值为 0.000ms。

c) “每个程序的执行状态”
显示在参数模式“辅助设置”中指定的程序执行状态。
1) “程序”
按用参数设定的顺序显示程序名。
2) “执行”
显示用参数设置的程序类型。
3) “扫描时间”
显示实际扫描时间（当前值）。在程序停止（等待）时，扫描时间显示为 0.000ms。
4) “恒定等待”
显示程序执行的次数。当测量从“0”开始时，设置为开始点（当次数达 65536 次时，复位至 0）。即使程序停止，但次数保存。

(3) 注意事项
执行恒定扫描执行型程序的扫描时间不在屏幕上显示，但在扫描时间栏里显示–短划符号。

7.11.2 中断程序监控表

(1) 什么是中断程序监控表?

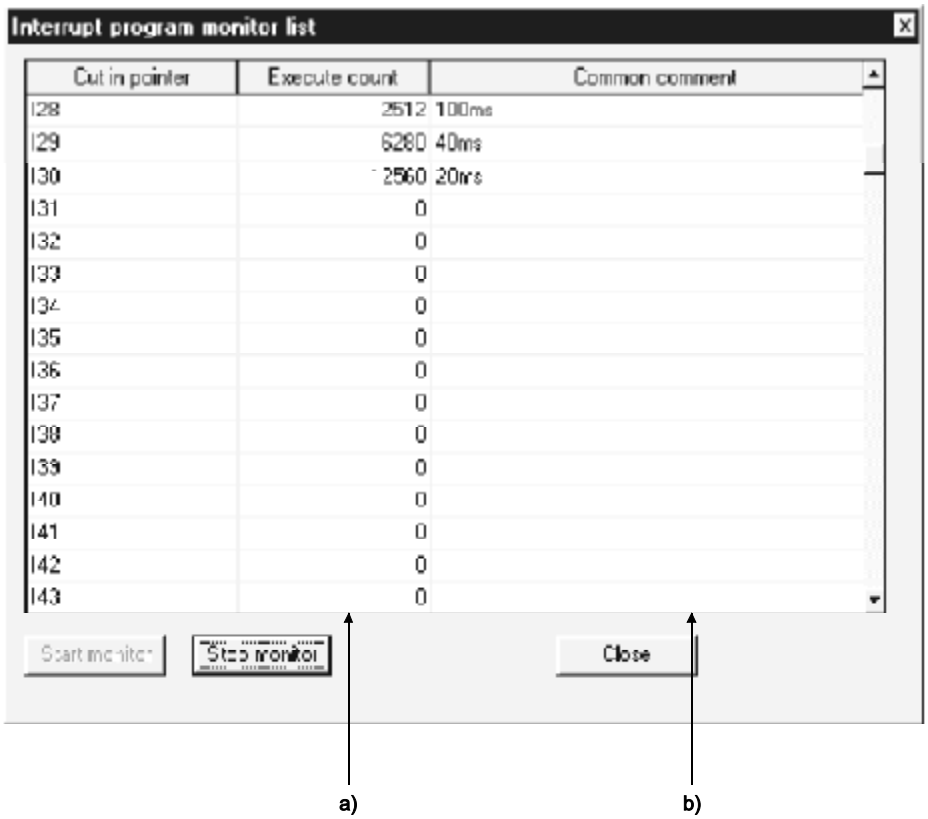
(a) 这是一种显示中断程序执行次数（**10—1255**）的功能。

(b) 用于确认中断程序的执行状态。

(2) 使用中断程序监控表

(a) 选择“在线”→“监控”→“中断程序监控表”。屏幕上出现中断程序监控表对话框。

(b) 下表所示为中断程序监控表的执行例子：



a) “执行次数”
显示执行中断程序的次数。设置在运行时操作状态的开始点（当次数达到65536 次时，复位至 0。）。

b) “公共注释”
表示要在中断点（10—1255）创建的软元件的注释。

7 - 41

7.11.3 扫描时间测量

(1) 什么是扫描时间测量?

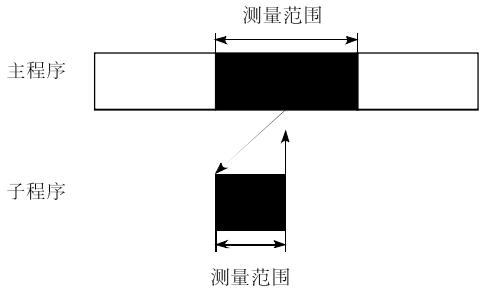
(a) 这是一种显示设置的程序处理时间间隔的功能。

(b) 用下列二种步骤的任何一种来指定扫描时间测量范围:

- 使用回路监控屏幕进行设置。
- 使用扫描时间测量对话框进行设置。

(c) 子程序和中断程序的扫描时间也可设置。

(d) 当存在子程序调用命令时，扫描时间测量的范围要包括子程序的时间。
该时间还要加上执行中断程序和固定循环执行型程序的时间。



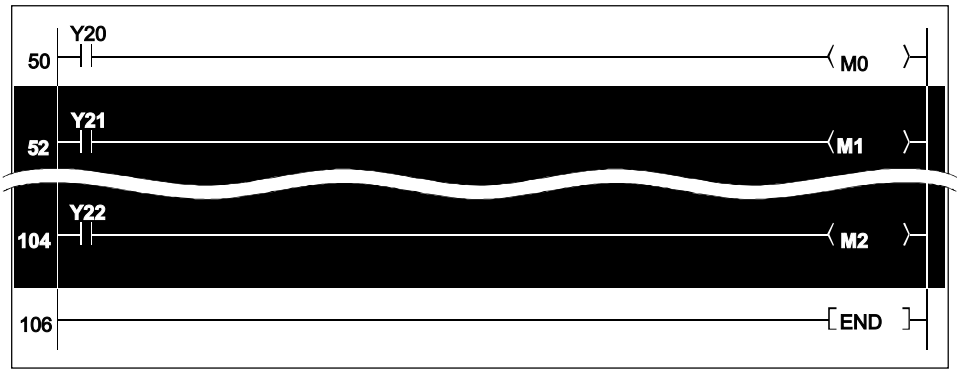
(2)扫描时间的测量

按下列步骤测量扫描时间。

(a) 显示要测量扫描时间的回路前沿，并设置监控模式。



(b) 指定扫描时间测量范围。
（高亮部分为指定区域。）



(c) 选择“在线”→“监控”→“扫描时间测量”，以打开扫描时间测量对话框。



(d) 点击“开始”（**Start**）按钮。



(3) 注意事项

(a) 设置开始步的值<完成步的值。

(b) 跳越至另一个程序文件的时间不能测量。

(c) 如果测量时间少于 **0.100ms**，显示为 **0.0000ms**。

(d) 如果在 **FOR** 指令和 **NEXT** 指令之间指定测量范围，扫描时间即显示为指定步之间的测量范围内进行测量的执行时间。

7.12 采样跟踪功能

(1) 什么是采样跟踪功能?

(a) 本功能为按指定的时间连续对 **QCPU** 软元件采样的功能。

(b) 在调试时，程序使用的软元件的详细变化能按指定的时间进行检查。
当触发条件满足时，采样跟踪功能能读出软元件的详细内容。

(c) 在设置的时间间隔（采样周期）里，采样跟踪能对指定的软元件采集详细数据，并将跟踪结果保存在存储卡的采样跟踪文件里。

(d) 采样跟踪文件保存跟踪条件数据和执行采样跟踪所必须的跟踪执行数据。当用 **GX Developer** 开始跟踪时，跟踪按设置的次数进行。
采样跟踪区域为 **60k** 字节。
用为一个软元件规定的字节数去除 **60k** 字节即可得到跟踪次数。其公式为：
（位软元件数）**/8+2**×（字软元件数）

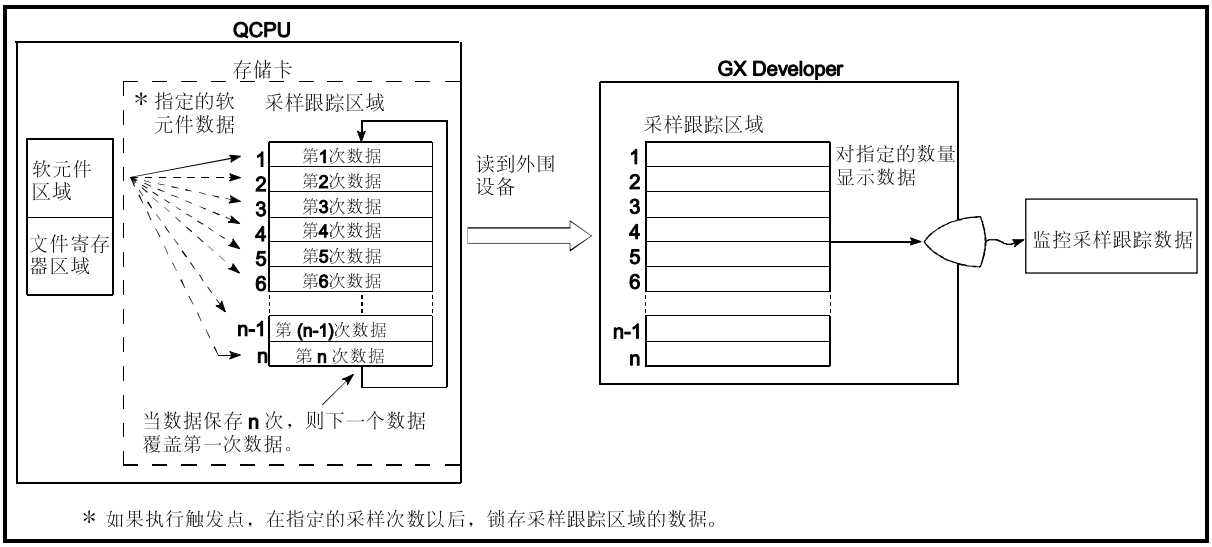
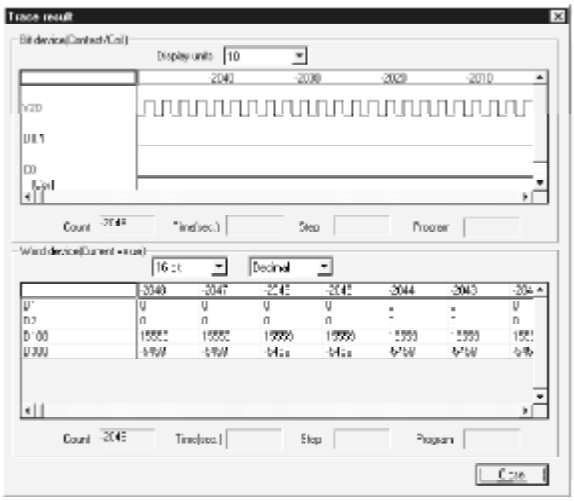


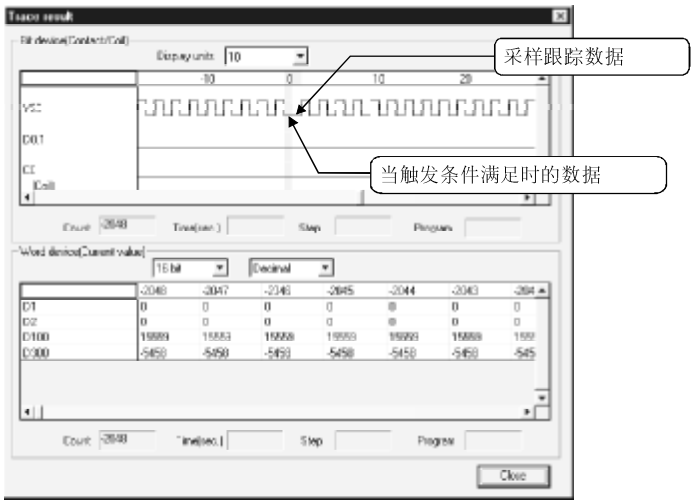
图 **7.8** 采样跟踪操作

(e) 在采样周期，跟踪结果显示为位软元件的 **ON/OFF** 状态，以及字软元件的当前值。



要点

在触发点设置所指定的触发条件下，可读出软元件的详细数据。每次扫描都执行采样，在由外围设备的触发操作采样结束之前，因为采样时间和触发条件中的相同，因此数据被两次采集。



7 - 45

(f) 采样跟踪功能的执行状态被保存在特殊继电器里（**SM**800，**SM**802，**SM**804 和 **SM**805）。

在使用采样跟踪功能时如果出错，**SM**826 接通。在顺控程序中使用特殊继电器，允许检查采样跟踪功能的执行状态。

1) 当 **GX Developer** 设置的“跟踪数据”和“跟踪条件”写入完成后，**SM**800（采样跟踪准备）接通。**SM**800 表示是否可以执行采样跟踪。

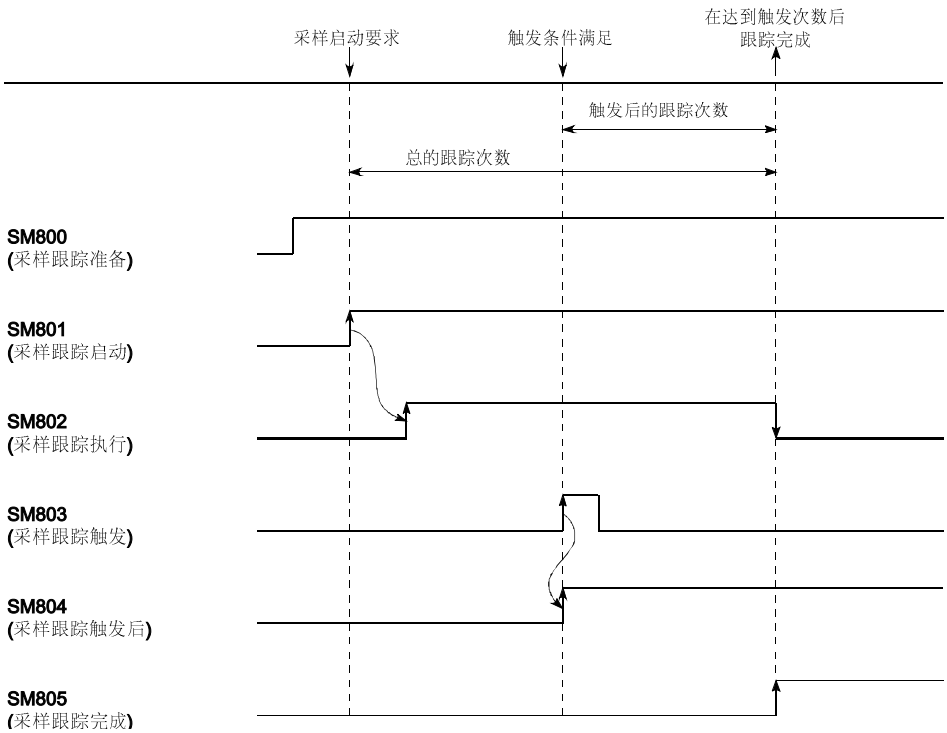
2) 当接受采样跟踪开始的要求后，采样跟踪启动，而 **SM**802（采样跟踪执行）接通。**SM**802 表示，是否可以执行采样跟踪：

- **GX Developer** 发出跟踪启动请求。
- **SM**801 接通。

3) 当下一个触发条件满足时，**SM**804（在采样跟踪触发之后）接通。**SM**804 表示是否满足触发条件：

- **GX Developer** 发出跟踪启动请求。
- 执行跟踪指令。
- **SM803** 接通

4) 采样跟踪完成后，**SM**805（采样跟踪完成）接通。

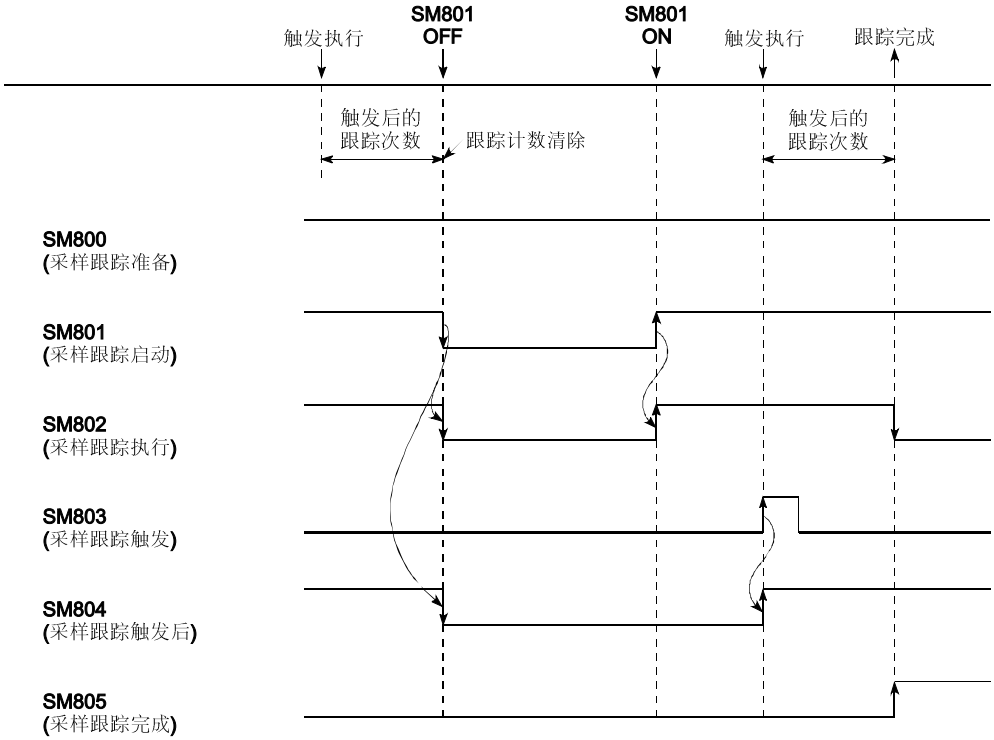


7 - 46

(d) 跟踪中断

1) 在采样跟踪期间，当 **SM**801（采样跟踪启动）断开，则采样跟踪中断。同时，跟踪次数清除。

2) 当 **SM**801 再次接通，跟踪重新启动。



＊ 当用不着 **GX Developer** 中断跟踪时, **SM800** 也断开。

7 - 47

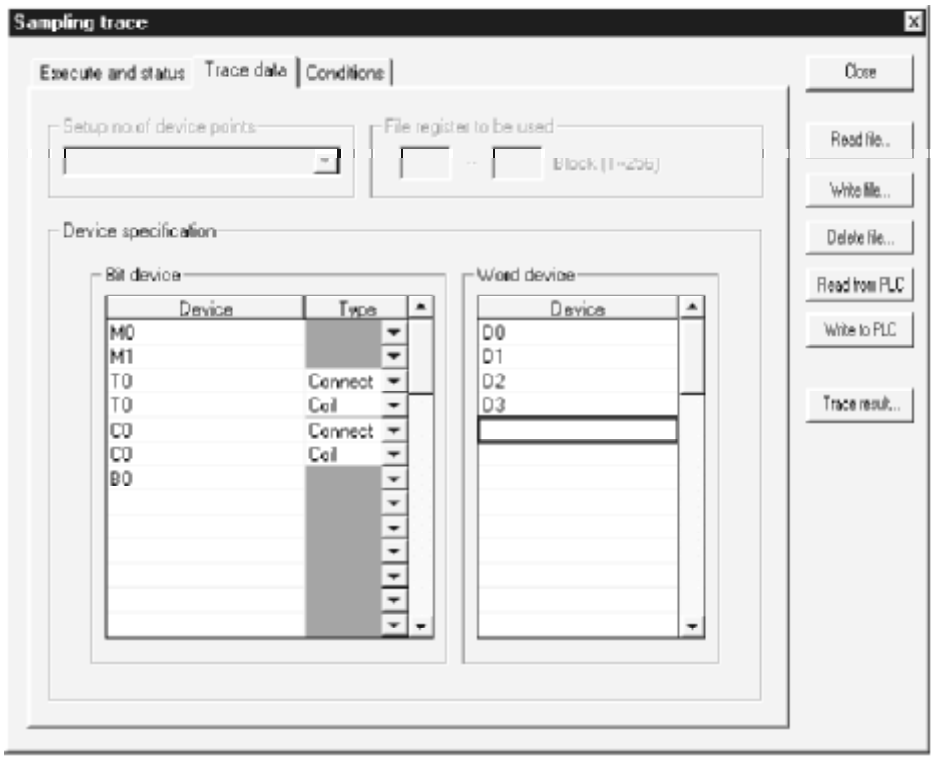
(2) 操作程序

采样跟踪操作以下列方式进行：

每个操作都在“采样跟踪”屏幕在线模式跟踪菜单上的执行。

(a) 跟踪软元件设置

在采样跟踪屏幕的“跟踪数据设置”上设置软元件，以执行采样跟踪。



1) 位软元件

最大为 50 位的软元件可按如下方式设置。

- X，DX，Y，DY，M，L，F，SM，V，B，SB
- T（接触器），T（线圈），ST（接触器），ST（线圈）
- C（接触器），C（线圈）
- J□\X，J□\Y，J□\B，J□\SB，BL□\S

2) 字软元件

最大为 50 字的软元件可按如下方式设置。

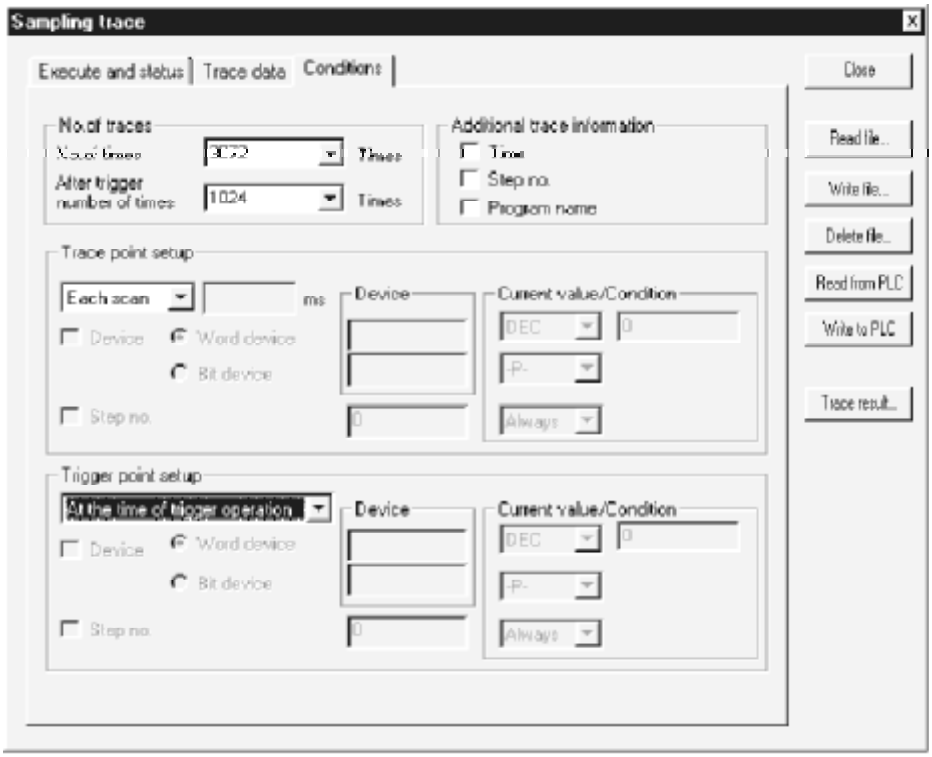
- T（当前值），ST（当前值），C（当前值），D，SD，W，SW，R，Z，ZR，
- U□\G，J□\W，J□\SW

7 - 48

(b) 设置跟踪条件

用“采样跟踪”屏幕的“跟踪条件设置”设置跟踪条件。

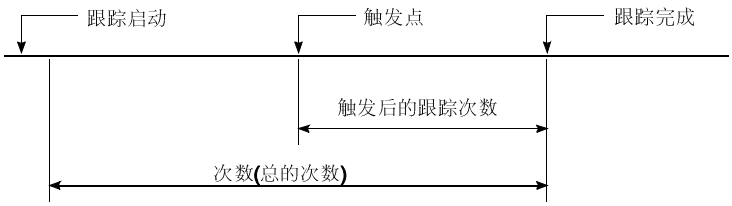
跟踪条件可设置为：“跟踪次数”，“跟踪点设置”，“触发点设置”，和“附加跟踪信息”。



1) 跟踪次数

a) 总的跟踪次数设置为从跟踪执行到跟踪完成，执行采样跟踪总的次数设置。

b) 触发后的次数设置为从触发执行到跟踪完成，执行采样跟踪的次数设置。



c) 每个次数的设置范围如下：

（触发后的次数）≤（总次数）≤（8192）

2) 跟踪点设置

设置采样跟踪数据的计时时间，可从下列几项中选择：

a) 每次扫描

每次扫描对 END 处理执行。

b) 间隔

在每个指定时间执行。

c) 详细设置

设置软元件和步数。设置方法和跟踪数据采样时间与第 7.9.1 节设置监控条件相同。

在详细条件下可设置的软元件如下：

- 位软元件 ：X，Y，M，L，F，SM，V，B，SB，T（接触器），ST（接触器），C（接触器），J□\X，J□\Y，J□\B，J□\SB，BL□\S
- 字软元件 ：T(当前值)，ST(当前值)，C(当前值)，D，SD，W，SW，R，Z，ZR，U□\G， J□\W，J□\SW

可为上述软元件设置以下属性：

- 数字规格的位软元件号
- 字软元件的位数规格

3) 触发点设置

用于设置执行触发的点，可从下列几项中选择：

a) 在“跟踪”命令的时间：

执行跟踪指令的时间即为触发点。

b) 触发操作的时间：

从 GPPW 功能的外围设备而来的触发操作即为触发点。

c) 详细设置

设置软元件和步数。设置方法和触发执行计时与第 7.9.1 节监控条件设置相同。

4) 附加跟踪信息

对每次跟踪设置附加信息。下列多个项目可供选择（没有一项是必须选择的）

a) 时间

执行跟踪后保存时间。

b) 步数

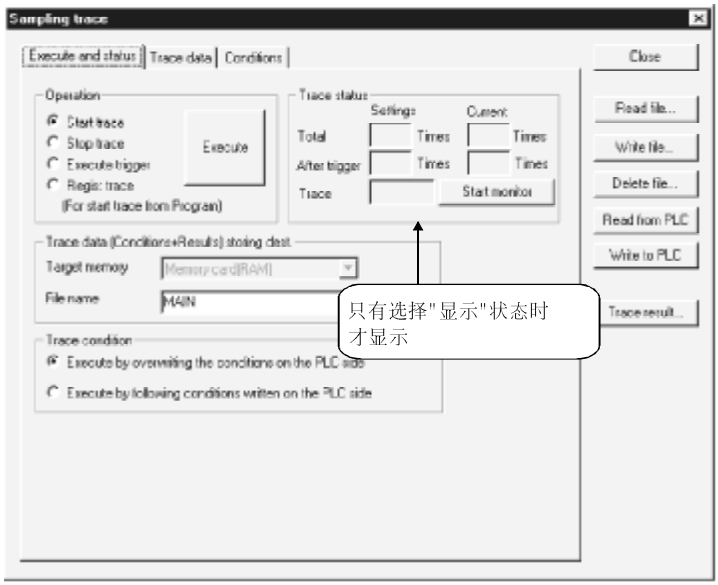
执行跟踪后保存步数。

c) 程序名

执行跟踪后保存程序名。

(c) 创建的跟踪数据和跟踪条件写入存储卡（**RAM**）。
跟踪文件写入存储卡（**RAM**）。
用“采样跟踪”屏幕的“写入 PLC”将跟踪文件写入存储卡。
将带有文件名的文件写入存储卡，这样就能保存多个跟踪文件。

(d) 执行采样跟踪。
用“采样跟踪”屏幕的“执行和状态”执行采样跟踪。



当显示“执行和状态”时，方能设置“操作”，“跟踪数据（条件+结果）保存目的地）”，和“跟踪条件”。

1) 用“选项”选择下列项目之一：

- 启动跟踪
  跟踪启动，开始跟踪计数。
- 停止跟踪
  跟踪中断。触发后，跟踪计数和次数清除。（当重新启动跟踪时，再选择“启动跟踪”。）
- 执行触发
  触发后开始对次数计数。当触发后达到跟踪执行的次数时，跟踪完成。
- 注册跟踪
  当执行程序时，注册跟踪数据。

2) 在“跟踪数据（条件+结果）保存目的地”部分，给保存跟踪数据和跟踪条件的文件指定一个文件名。
跟踪条件也可保存在指定文件名的文件中。

3) 用“跟踪条件”选择下列项目之一：
- 在 PLC 侧，通过条件的覆盖来执行。
  在已有的跟踪文件覆盖跟踪条件。
- 把下列条件写入 PLC 侧来执行。
  在“跟踪数据（条件+结果）保存目的地”上指定跟踪文件的条件，执行程序。

(e) 从 QCPU 读出跟踪结果并显示数据。
1) 从“由 PLC 读出”读出跟踪数据。
2) 在“跟踪结果”的显示屏上显示读出跟踪结果。

要点
当采样跟踪执行一次后，不会执行第二次跟踪。 要重新执行时，要执行“跟踪”指令，并复位采样跟踪。

(3) 注意事项

(a) 在存储卡（SRAM）上设置采样跟踪文件。

(b) 采样跟踪能从网络的另一个站或串行通信模块执行。但是，一次不能从多个区域执行跟踪。跟踪只能用 QCPU 在一个区域执行。

(c) 在 QCPU 上注册的跟踪资料也注册在 SRAM 卡上并锁存。当条件数据保存在跟踪文件里时，即使电源切断或 QCPU 复位，在注册跟踪条件下，采样跟踪也能执行。
当电源切断→接通/QCPU 复位时，在下列情况时，锁存的跟踪资料会被清除：
- 在跟踪文件上注册的 SRAM 卡未插入
- 跟踪文件破坏

这时需要用 GPPW 操作要求再次注册跟踪信息。
为清除数据，用 RESET/L. CLR 开关执行锁存清除操作。

(d) 将 QCPU 连接到 GX Developer 功能的外围设备进行操作。

(e) 在停止状态，QCPU 不能读出采样跟踪结果。为了能使 QCPU 读出采样跟踪结果，使 QCPU 进入运行状态。

(f) 在执行采样跟踪时，要保证在触发点的触发条件不满足。如果当执行采样跟踪时，触发条件满足，它仍将不能识别出触发条件。

7.13 多用户的调试功能

(1) 什么是多用户的调试功能?

(a) 本功能为同时从多个连接到 **QCPU** 的 **GX Developer** 功能外围设备或串行通信模块执行调试的功能。

(b) 如果调试任务根据处理或功能分类，当多个 **GX Developer** 功能外围设备用不同文件调试时，本功能可用于同时执行调试。

(2) 功能说明

多用户的调试功能如下：

后面执行的功能 / 正在执行的功能	监控	运行时写入	执行时间测量	采样跟踪
监控	○	×	○	○
运行时写入	×	○	×	×
执行时间测量	○	×	×	○
采样跟踪	○	×	○	×

○：可同时执行。（但是详细条件只能用一个 **GX Developer** 功能外围设备来设置。在这种情况下，详细条件不能用另一个 **GX Developer** 功能的外围设备来设置。）

×：仅能用一个 **GX Developer** 功能的外围设备来执行。（当另一个 **GXDeveloper** 正在执行本功能时，该功能不能用 **GX Developer** 执行。）

7.13.1 多用户监控功能

(1) 什么是多用户监控功能?

(a) 用 GX Developer 连接到 QCPU 或串行通信模块上可执行多用户监控操作。

(b) 多个用户可同时监控。通过设置站监控文件，可执行高速监控。（站监控文件并非必须设置。）

(2) 操作步骤

(a) 为进行多用户监控操作，按下列步骤创建用户系统文件。

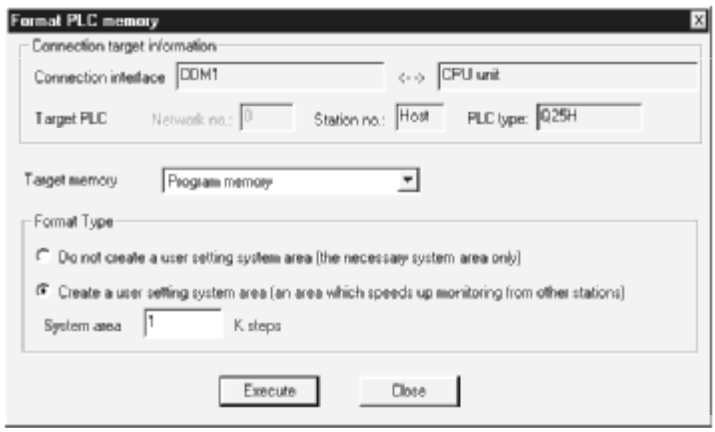
1) 选择“在线”→“PLC 存储器格式化”，以打开 PLC 存储器格式化对话框。

2) 从目标存储器列表框选择“程序存储器”。

3) 在格式化部分，选择“创建用户设置系统区域”，以便可检查单选按钮。

4) 在系统区域文字框里指定期望的 k 步。

(b) 下图举例说明在系统区域文本框里指定“1k 步”。



1) 在系统区域 1k 步模块中，最大可设置 15k 步。1k 步只可对应于一个站的监控文件。因此，最多可设置 15 个站监控文件。

(3) 注意事项

(a) 监控的详细条件设置只能从一个区域设置。

(b) 即使未设置站的监控文件，也能执行监控。但不能执行高速监控。
系统区域是在程序存储器的同一区域内。所以当设置系统区域后，保存程序的区域会减小。

(c) 一旦分配了用户定义系统区域，单个 PLC 能被 16 个站访问。

7 - 54

7.13.2 多用户运行写入功能

(1) 什么是多用户运行写入功能?

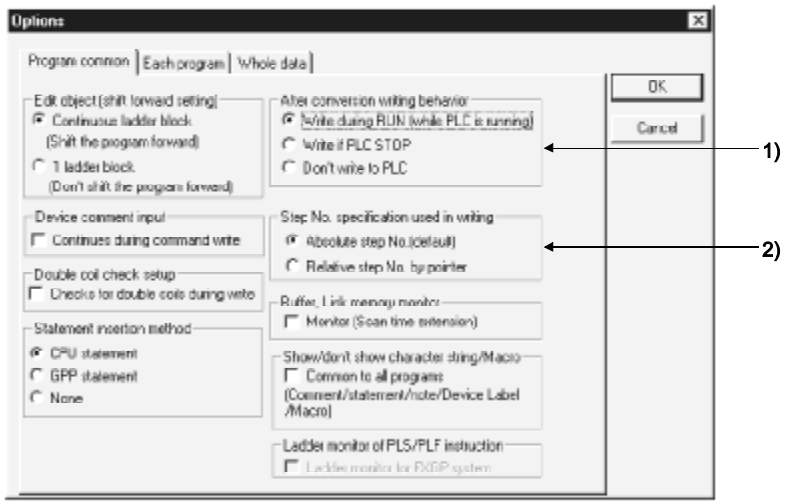
(a) 运行中多个用户可以写入一个文件或多个不同的文件。

(b) 为了在运行操作中，多个用户能同时写入一个文件，预先为“运行时写入”指定期望指针。然后选择“用指针定相关步号”，以便选择单选按钮。

(2) 操作步骤

多用户运行写入操作以下列方式进行：

(a) 从选择菜单上选择工具，并设置“转换后写入性能”和“用于写入的步号指定。”



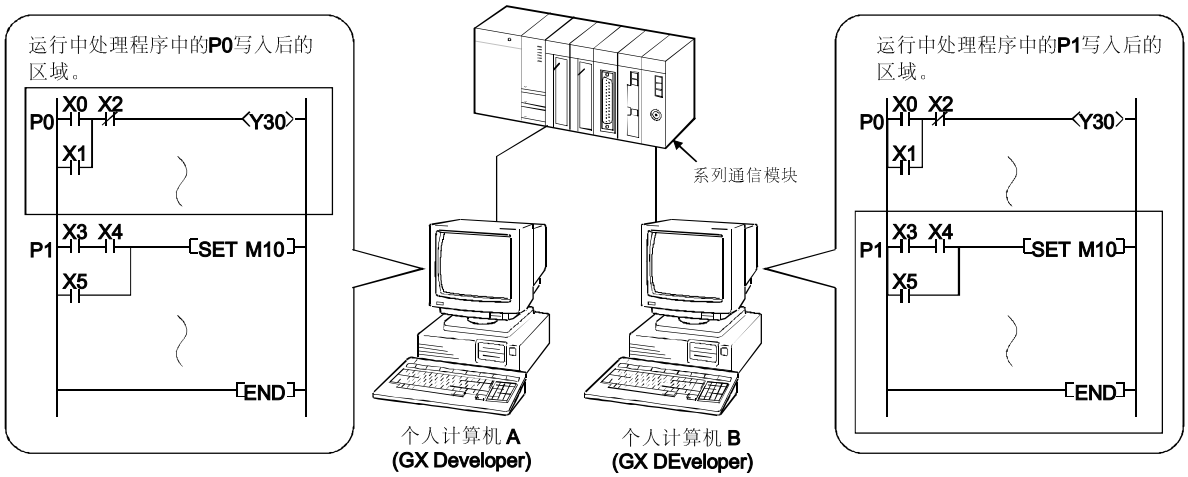
1) 在“转换中写入设定后”中设置“运行时写入（当 **PLC** 在运行时）”。

2) 在“用于写入的步号指定”中选择“正常”或“使用相关指针”。

7 - 55

(b) 显示指定指针回路以便在运行中转换后写回路。

下例为运行中 GX Developer A 从 P0 写入，而运行中 GX Developer B 从 P1 写入。周围有 [ ] 的程序区域即运行中写入的区域。



(3) 注意事项

“运行时写入”的注意事项与“运行中写入回路模式”相同。参见第 7.10.1 节。

进一步的资料，参见第 7.10.1 节。

7 - 56

7.14 警戒定时器（WDT）

(1) 什么是警戒定时器?

(a) 警戒定时器是一种内置顺序定时器，用以探测 QCPU 硬件和/或顺控程序的出错。

(b) 当警戒定时器达到定时值，则会出错。
QCPU 以如下方式响应警戒定时器出错。
1) QCPU 切断所有的输出。
2) 前部的 RUN LED 熄灭，而 ERROR LED 开始闪烁。
3) SM1 接通，出错代码在 SD0 中。

(c) 警戒定时器的缺省值为 200ms。
设置范围为 10—2000ms(10ms 模块)。

(2) 警戒定时器的设置和复位

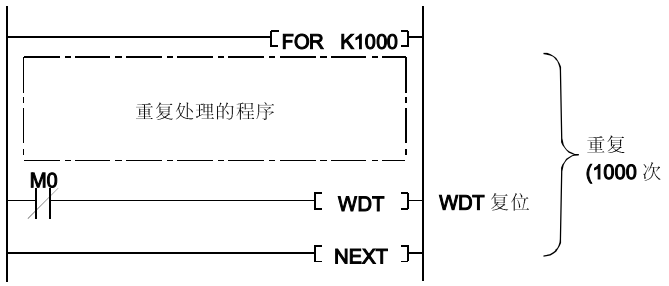
(a) PLC 参数的 PLC RAS 设置可将警戒定时器复位。

(b) 当执行 END 指令时，QCPU 将警戒定时器复位。
1) 当 END/FEND 指令在顺控程序中警戒定时器设定值之内执行，而 QCPU 操作正确，警戒定时器不会超时。
2) 当 QCPU 硬件有故障或顺控程序扫描时间太长，在警戒定时器设置值之内就不能执行 END/FEND 指令，警戒定时器超时。

(3) 注意事项

(a) 在警戒定时器的测量时间以内会产生 0—10ms 的误差。设置警戒定时器的期望值时已把这一误差考虑在内。

(b) 在顺控程序中用 WDT 指令来使警戒定时器复位。如果在 FOR 和 NEXT 指令重复执行时警戒定时器达到定时值，则用 WDT 指令将警戒定时器复位。



(c) 即使在顺控程序中警戒定时器复位，扫描时间也不会复位。
扫描时间值根据 END 指令来测量。

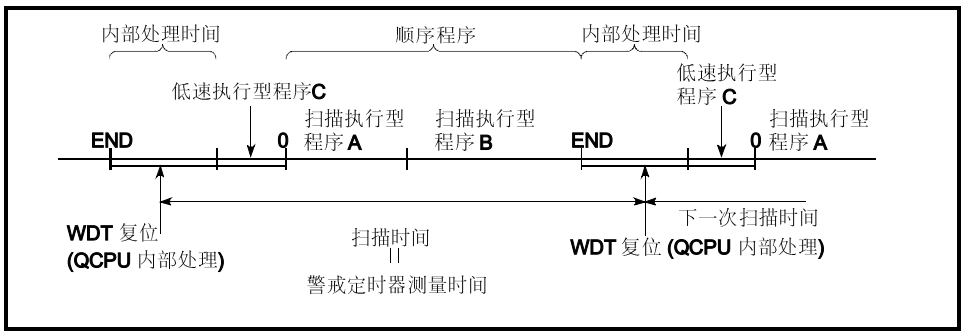


图 7.9 警戒定时器复位

备注

扫描时间为 QCPU 从 0 步开始处理一个顺控程序到 QCPU 重新从 0 步处理另一个同样文件名的顺控程序所经过的时间。
并非每次扫描的扫描时间相同或不同，要取决于
• 是否执行所使用的命令。
• 是否执行中断程序和固定扫描执行型程序。

7.15 自诊断功能

(1) 什么是自诊断功能?

(a) 自诊断功能为 QCPU 本身对 QCPU 内是否出错进行诊断的功能。

(b) 自诊断功能的目的在于防止 QCPU 误操作并作为一种预防性的维护措施。
当 QCPU 通电或在 QCPU 处于运行模式时，自诊断处理可以探测并显示出错。该功能也能停止 QCPU 的计算。

(2) 出错探测的处理

(a) 当 QCPU 探测出错误时，ERROR LED 亮。探测到错误时特殊继电器（SM0，SM1）接通，该错误的出错代码保存到特殊寄存器 SD0 中。当探测到多个错误时，最后一个错误的出错代码保存到特殊寄存器 SD0 中。为了探测错误，在程序中使用特殊继电器和特殊寄存器，从而使这些软元件与可编程控制器和机械系统互锁。

(b) QCPU 保存 16 个最新的出错代码。（参见第 7.16 章）
用 GX Developer 功能 PLC 诊断模式也能检查故障历史记录。
当电源切断使用备用电池时，故障历史记录也能保存。

(3) 在出错探测时的 QCPU 操作

(a) 当由自诊断探测出错误时，QCPU 的操作可以变为两种模式。

1) QCPU 计算停止模式
在探测到故障的点停止计算，并切断所有的输出（Y）。

2) QCPU 计算继续模式
当探测到错误时，发生故障的程序（指令）区域被跳越，执行其余的程序。

(b) 用参数模式 PLC RAS，下列错误可将计算设置为“继续/停止”。
（所有的参数缺省设置为“Stop”。）

1) 计算错误
2) 扩展命令错误
3) 保险丝熔断
4) I/O 模块比较
5) 智能型模块程序执行错误
6) 存储卡存取错误
7) 存储卡操作错误
8) 外部电源切断（为将来使用）

例如，如果 I/O 模块验证出错被设定为“继续”，则在出错前，I/O 地址的计算继续。

(4) 出错检查选择

在参数模式 PLC RAS 设置中下列出错检查可设置为“是/否”。
（所有的参数缺省设置为“是”。）

(a) 电池检查

(b) 保险丝熔断检查

(c) I/O 模块比较

自诊断一览表

诊断说明		出错信息	诊断时间
硬件故障	CPU 错误	MAIN CPU DOWN	• 始终
	END 指令不执行	END NOT EXECUSE	• 当执行 END 指令时
	RAM 检查	RAM ERROR	• 当电源接通时/当复位时
	计算回路检查	OPE.CIRCUIT ERR.	• 当电源接通时/当复位时
	保险丝断路 （缺省：停止）*1	FUSE BREAK OFF	• 当执行 END 指令时 （缺省：是）*2
	外部电源切断	EX.POWR OFF	• 为将来使用
	I/O 中断错误	I/O INT ERROR	• 当发生中断时
	智能型功能模块错误	SP.UNIT DOWN	• 当电源接通时/当复位时 • 当执行 FROM/TO 指令时
	控制总线错误	CONTROL–BUS ERROR.	• 当电源接通时/当复位时 • 当执行 END 指令时 • 当执行 FROM/TO 指令时
	发生瞬间停止	AC/DC DOWN	• 始终
	电池低电压	BATTERY ERROR	• 始终（缺省：是）*3
处理错误	I/O 模块验证 （缺省：停止）*1	UNIT VERIFY ERROR	• 当执行 END 指令时 （缺省：是）*2
	智能型功能模块分配错误	SP.UNIT LAY ERR.	• 当电源接通时/当复位时 • 当从 STOP 切换到 RUN 时
	智能型程序执行错误 （缺省：停止）*1	SP.UNIT ERROR	• 当执行 FROM/TO 指令时
	无参数	MISSING PARA.	• 当电源接通时/当复位时
	引导错误	BOOT ERROR	• 当电源接通时/当复位时
	存储卡操作错误 （缺省：停止）*1	ICM.OPE.EROR	• 当装入存储卡时/拿出时
	文件设置错误	FILE SET ERROR	• 当电源接通时/当复位时
	文件存取错误 （缺省：停止）*1	FILE OPE.ERROR	• 当执行指令时
	不能执行命令	CAN'T EXE.PRG.	• 当电源接通时/当复位时
参数错误	参数设置检查	PARAMETER ERROR	• 当电源接通时/当复位时 • 当从 STOP 切换到 RUN 时
	链接参数错误	LINK PARA.ERROR	• 当电源接通时/当复位时 • 当从 STOP 切换到 RUN 时
	SFC 参数错误	SFC PARA.ERROR	• 当从 STOP 切换到 RUN 时
	智能参数错误	SP.PARA.ERROR	• 当电源接通时/当复位时

*1: 在 GX Developer 功能参数设置中，可以改变为“继续”。

*2: 在 GX Developer 功能参数设置中，可以设定为“否”。当 SM251 接通时，不能进行检查。

*3: 在 GX Developer 功能参数设置中，可以设定为“否”。

自诊断一览表（续前页）

诊断说明		出错信息	诊断计时
程序错误	指令代码检查	INSTRUCT CODE.ERR.	• 当电源接通时/当复位时 • 从 STOP 切换到 RUN 时
	无 END 指令	MISSING END INS.	• 当电源接通时/当复位时 • 从 STOP 切换到 RUN 时
	指针设置错误	CAN'T SET(P)	• 当电源接通时/当复位时 • 从 STOP 切换到 RUN 时
	指针设置错误	CAN'T SET(I)	• 当电源接通时/当复位时 • 从 STOP 切换到 RUN 时
	操作检查错误（缺省：停止）*1	OPERATION ERROR	当执行指令时
	FOR 至 NEXT 指令结构错误	FOR NEXT ERROR	当执行指令时
	CALL 至 RET 指令结构错误	CAN'T EXECUTE(P)	当执行指令时
	中断程序错误	CAN'T EXECUTE(I)	当执行指令时
	不能执行指令	INST.FORMAT ERR.	当从 STOP 切换到 RUN 时
	SFC 程序结构错误	SFCP.CODE ERROR	当从 STOP 切换到 RUN 时
	SFC 块结构错误	CAN'T SET(BL)	当从 STOP 切换到 RUN 时
	SFC 步结构错误	CAN'T SET(S)	当从 STOP 切换到 RUN 时
	SFC 语法错误	SFCP.FORMAT ERR.	当从 STOP 切换到 RUN 时
	SFC 操作检查错误（缺省：停止）*1	SFCP.OPE.ERROR	当执行指令时
	SFC 程序执行错误	SFCP.EXE.ERROR	当从 STOP 切换到 RUN 时
	SFC 块执行错误	BLOCK EXE.ERROR	当执行指令时
	SFC 步执行错误	STEP EXE.ERROR	当执行指令时
PLC 错误	警戒定时器错误管理	WDT ERROR	始终
	程序超时	PRG.TIME OVER	始终
信号报警器检查		F * * * *	当执行指令时
CHK 指令检查		<CHK>ERR* * * _ * * *	当执行指令时

*1: 用 GX Developer 功能参数设置可以变为“继续”。

7.15.1 由于出错而中断

出错时，QCPU 可执行中断指针中断程序。

用 GX Developer 功能参数模式的 PLC RAS，可将计算设置为“继续/停止”的出错，对设置为“继续”的出错可执行计算。对设置为“停止”的出错，则执行停止错误的中断程序（132）。

相应于中断指针的错误如下表列。

中断指针	相应的出错信息
I32	停止所有错误
I33	空
I34	UNIT VERIFY ERR. FUSE BREAK OFF EX.POWER OFF SP.UNIT ERROR
I35	OPERATION ERROR SFCP OPE.ERROR SFCP EXE.ERROR
I36	ICM.OPE.ERROR FILE OPE.ERROR
I38	PRG.TIME OVER
I39	CHK 信号 信号器探测
I40 to I47	空

当出错时，系统可继续驱动模式。
当可选择"继续/停止"的区域出错，
可设置 "继续"。

要点

当电源接通或 QCPU 复位时，中断指针 I32—I39 处于禁止执行模式。当使用 I32—I39 指针时，要用 IMASK 指令启动执行。

7.15.2 出错时 LED 显示

出错时，装在 QCPU 前面的 LED 亮起。
有关 LED 的操作，详见第 7.19 节。

7 - 62

7.15.3 删除错误

只有当出错时能继续操作 QCPU 时才能进行 QCPU 出错取消操作。

(1) 删除错误

(a) 删除错误的步骤

以下列方式删除错误：

1) 分辨产生错误的原因。

2) 在特殊寄存器 SD50 里保存要删除错误的出错代码。

3) 将特殊继电器 SM5 从 OFF 切换到 ON。

4) 删除错误

(b) 删除错误后的状态

当 QCPU 的错误删除后恢复时，特殊继电器，特殊寄存器和 LED 都因出错前错误所设置的状态而受到影响。

当删除错误后又发生同样的错误，它再次被记录进故障历史。

(c) 信号报警器的删除

由于多次探测到信号报警器的删除，只删除第一次探测到的“F”。

要点
当删除错误通过将要删除的出错代码存储到 SD50 中的方法来执行时，不考虑代码的低端的 2 位数。 [例] 当出错代码中 2100 和 2111 出错时，2100 出错代码被删除，2111 出错代码也被删除。

7.16 故障历史记录

QCPU 能在存储器里保存故障历史记录（用自诊断功能探测到的结果和时间）。

要点
探测时间使用 QCPU 互锁，因此当第一次使用 QCPU 时一定要设置准确的时间。

(1) 保存区域

(a) 最新的 16 项故障保存在锁存 QCPU 故障历史记录的保存存储器里。

(b) 当保存内容超过 16 项时，可将历史记录保存在使用 PLC 参数的 PLC RAS 存储卡文件里。

(c) 如果故障历史记录的参数计数和存储卡计数不同，在执行下列操作时，清除存储卡历史记录文件的内容，然后将在 QCPU 故障历史记录保存存储器里的 16 项故障数据输入文件。

1) 当在参数历史记录文件中的历史计数变化时。

2) 当存储卡历史记录的计数与设置的参数不同时。

(d) 在故障历史纪录文件中的保存区域如下：

保存区域	设置存储卡中的文件
可保存的数量	最大 100（可变化*1）

*1: 当要保存量超过可保存量时，最新的历史记录覆盖最早的历史记录。

要点
即使存储卡中并不存在用参数设置的故障历史记录文件，QCPU 也不会出错。 QCPU 只能执行把故障保存在故障历史记录文件中的处理。

(2) 故障历史记录的清除方法

用 GX Developer PLC 诊断模式的故障历史纪录清除功能来清除故障历史纪录存储器和故障历史纪录文件。

用故障历史纪录清除功能可清除保存在 QCPU 故障历史记录的存储器里的数据文件和故障历史记录文件。

7.17 系统保护

QCPU 还有几项保护功能（系统保护）用于改变程序，以处理来自除设计人员之外的第三方的数据。（用 GX Developer 功能或串行通信模块的存取处理。）
有以下几种系统保护方法：

保护项目	保护有效文件	保护说明	方法	有效时间	备注
全部 CPU	全部文件	禁止对 QCPU 的全部写入/控制指定	QCPU 系统设置开关 SW1 设置在“ON”。	始终	也对软元件有效
存储卡模块	全部文件	对存储卡执行驱动保护和写入保护。	存储卡设置写保护开关设置在“ON”。	始终	
文件模块	程序 软元件注释 软元件初始值	文件属性变化为如下： 1） 禁止读/写显示 2） 禁止写	改变口令注册中的文件属性。	始终	

*对上述的控制指令，读/写显示，和写，说明如下：

项目	说明
控制指定	用远程操作进行 QCPU 操作指定。 （远程 RUN，远程 STOP，等等）
读/写显示	程序读/写操作。
写	写入程序和测试的操作。

要点

在 QCPU 系统的 SW1 设定为 ON，即使系统保护功能在起作用，用 PLC 参数和 QCPU 微动双列开关设置的下列功能也能执行：
- 从标准 ROM 和存储卡的引导
- 在标准 ROM 上自动写

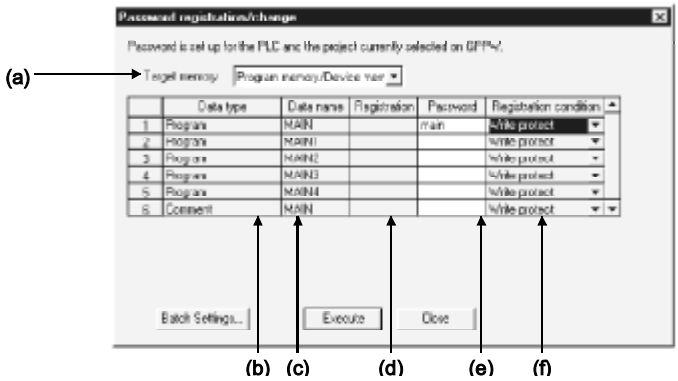
7.17.1 口令注册

口令用于禁止从 GPPW 的外围设备对 QCPU 中程序和注释进行读和写。
对特殊存储器（程序存储器，标准存储器/存储卡）的程序文件，软元件注释文件和软元件初始文件设置口令注册。
对注册项目有两点说明。
- 文件名不能显示，也不能进行读/写。
- 不能写入文件。（只能读）

如果口令被注册，除非输入相同的口令，否则 GX Developer 的文件操作不能进行。

(1) 口令注册

为执行口令注册，选择 **GX Developer** 在线→设置口令/建立写入 **PLC** 的关键字→注册口令。



每个项目说明如下：

(a) 目标存储器 .....设置保存文件的存储器，文件的口令被注册或更改。

(b) 数据类型 .....规定保存在目标存储器里的文件类型。

(c) 数据名 .....显示保存在目标存储器里的文件名。

(d) 注册.....显示表示口令保护文件的星号“*”。

(e) 口令.....定义或改变口令。

(f) 注册条件

1) 写入保护 .....由口令限制写操作。
（能读）

2) 读/写保护 .....由口令限制读/写操作。

3) 清除 .....口令清除。
（设置用该口令当前注册的口令。）

要点
(1) 口令保护仅限于程序文件，软元件注释文件，和软元件初始值文件。其它文件不能用口令保护。 **(2)** 文件注册口令不能从文件中读出。 如果没有记住口令，除以下情况外不能执行文件操作。 • 程序存储器/存储卡：**PLC** 格式 • 标准 **ROM**：成批写 要记录下注册的口令而且随时放在手边。

7 - 66

7.17.2 远程口令

用远程口令功能可防止远程用户对 **QCPU** 的非法访问。
在 **QCPU** 中建立远程口令功能后即使用该功能。
远程口令功能设置后，用户在远地通过串行通信模块或具有调制解调功能的以太网模块访问 **QCPU** 时，就能用远程口令核查。

(1) 建立，修改和删除远程口令

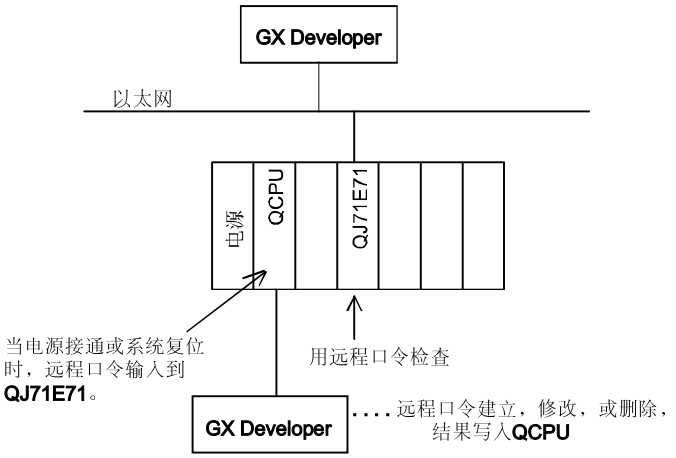
(a) 远程口令建立

在 **GX Developer** 远程口令设置屏幕上即可建立远程口令。然后 **GX Developer** 就可连接到已设置远程口令的 **QCPU** 上，远程口令上载。当电源接通或 **QCPU** 复位时，**QCPU** 就将远程口令输入特殊串行通信模块和以太网模块。

(b) 修改和删除远程口令

将 **GX Developer** 连接到相关的 **QCPU** 上就能修改和删除远程口令。
用 **GX Developer** 设置一个修改口令或删除一个远程口令，就能修改或删除 **QCPU** 中设置的远程口令。
从远地不能修改或删除远程口令。

例如，下图为用以太网模块建立、修改或删除一个远程口令的情况概要。

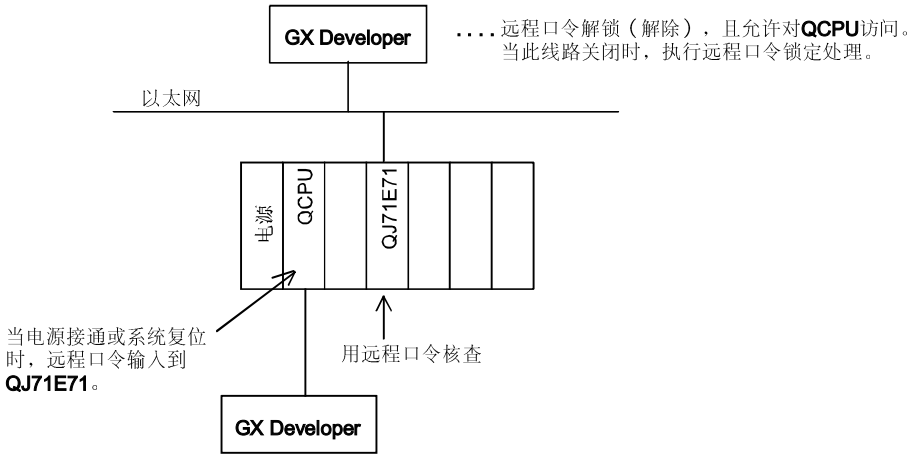


(2) 远程口令的锁定/解锁处理

通过调制解调器，串行通信模块和以太网，以太网模块远程口令可对访问源解锁。

如果远程口令相符，即可对 **QCPU** 访问。

例如，用以太网模块进行远程口令锁定/解锁处理时的情况概要如下。

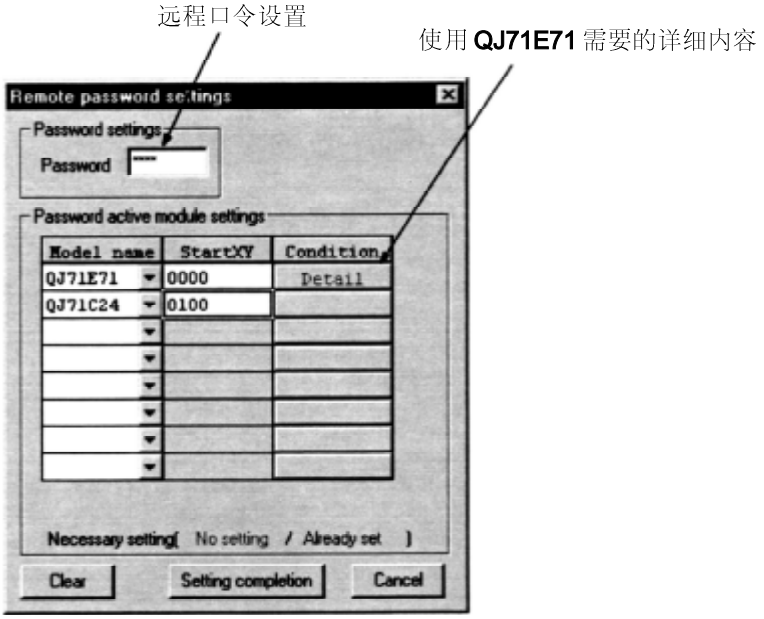


7 - 68

(3) 建立远程口令的步骤

[GX Developer]→[远程口令**]→[**远程口令设置**]**屏幕**→[**高级远程口令设置**]**屏幕。

(a) 设置屏幕



(b) 设置字段

字段			说明	设置范围/选择范围
口令设置			远程口令输入	**4** 字节，字母数字，特殊字符
口令启用，模块设置	型号		型号选择	**QJ71E71/QJ71C24**
	起始 **I/O**		设置模块的起始地址	000_H→$0FE0_H$
详细内容			—	设置/不设置
	用户连接端 **No.**		设置用户连接端 **No.**	连接端 **No.1**—连接端 **No.16**
	系统连接	自动打开 **UDP** 端口	对有效远程口令端口进行检查	—
		FTP 通信端口（**TCP/IP**）		
		GX 通信收发端口（**TCP/IP**）		
		GX Developer 通信端口（**UDP/IP**）		

要点

有关远程口令功能的详细内容，请参见下列手册。

- 使用串行通信模块
  Q 对应串行通信模块用户手册（应用）
- 使用以太网模块
  Q 对应以太网接口模块用户手册（基本）

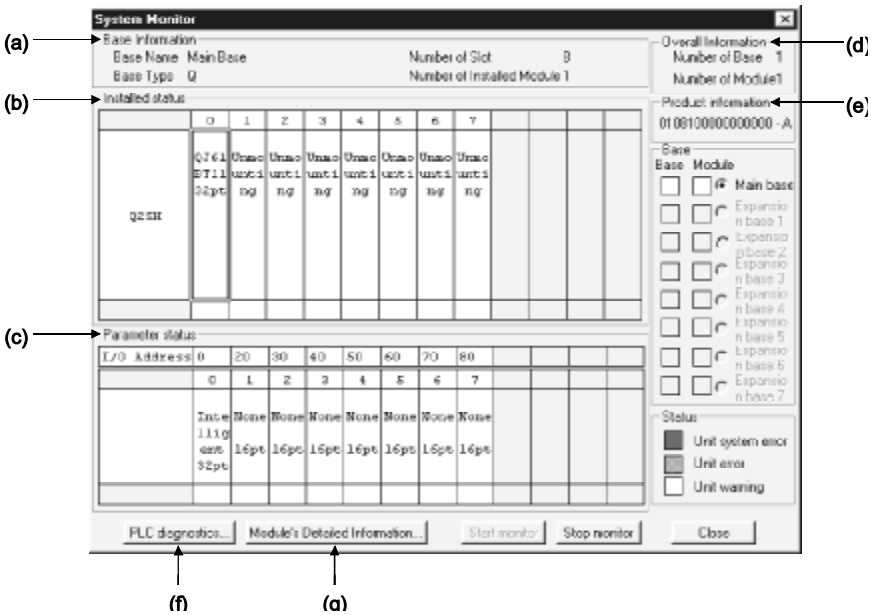
7 - 69

7.18 用 GX Developer 监控 QCPU 系统状态（系统监控）

(1) 用 GX Developer（SW4D5C–GPPW–E，SW5D5C–GPPW–E）监控 QCPU 系统状态

系统监控窗口提供有关连接到个人计算机的 QCPU 的如下信息：

- 基本信息
- 安装信息
- 参数状态
- 整体信息



- 产品信息

(a) 基本信息

基本信息部分显示的信息为：选择的基板，基板名称（主基板或扩展基板 1—7），插槽数量，基本类型，装入基板的模块数量。

1） “基板名称”字段表示主基板或扩展基板（带电源），即使在扩展基板上未装入电源模块，仍表示扩展基板。

2） “基本类型”字段显示下列缩写：

- Q 型代表　　Q33B，Q35B，Q38B，Q312B
- Q 型代表　　Q63B，Q65B，Q68B，Q612B
- QA 型代表　　QA1S65B，QA1S68B

3） “装入模块数量”字段表示安装在基板上的模块数量。

(b) 安装状态

“安装状态”部分表示已选择基板的型号名称和点数。插槽栏表示，如果基板未安装入特殊插槽，即为“不安装”。

7 - 70

(c) 参数状态

“参数状态”部分显示基板每个插槽的输入/输出数，模块类型，以及点数。如果可用的点数和安装状态在参数状态部分的任何一栏里显示，则作设定使 PLC 参数的 I/O 分配能和安装状态相符合。

(d) 整体信息

整体信息部分显示使用的基板数和安装在基板上的模块数。

(e) 基板

基板部分表示使用的基板状态和装入的模块。模块栏表示当模块处于异常状况下时的模块状态。

(f) PLC 诊断

此按钮用于监控 QCPU 状态和错误。

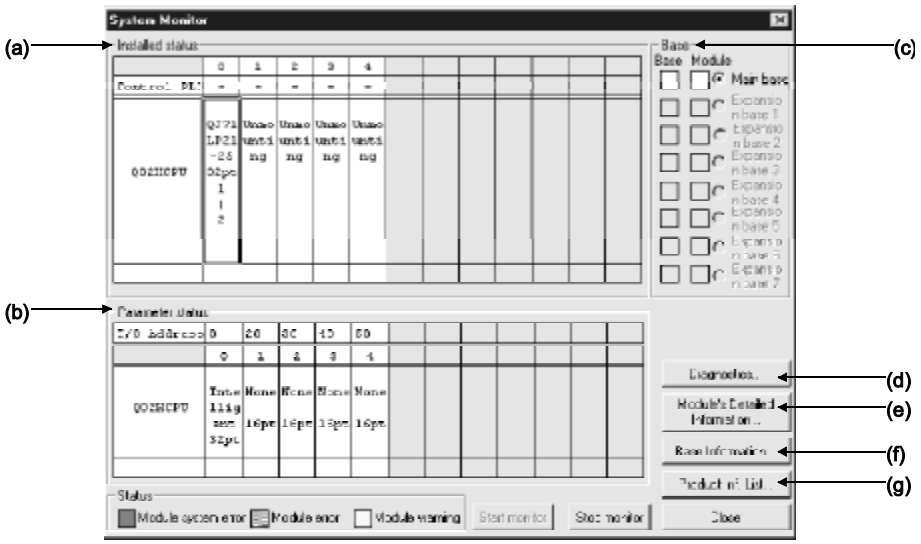
(g) 模块的详细资料

此按钮用于查阅选用模块的详细资料。有关智能型功能模块，详见智能型功能模块手册。

(2) 用 GX Developer（SW6D5C–GPPW–E）显示 QCPU 系统

用 GX Developer 系统监控器可确认连接到个人计算机上的 QCPU 的下列信息。（见以下说明。）

- 安装状态
- 操作状态
- 模块详细资料
- 产品信息



(a) 安装状态

启动控制 QCPU，确认装入基板的模块类型和数量。

不装模块的插槽就显示为“不安装”。

用 PLC 参数的 I/O 分配设置将插槽设置为“空”时，在装入模块后不显示模块型号。

(b) 操作状态

启用 I/O 地址，确认在选用的基板上装入各个插槽的模块类型和模块数量。

如果操作状态显示 0 空点并显示分配出错，则表示 PLC 参数的 I/O 分配和实际状态不符。

这种情况下，要采用分配 I/O 的办法来使 PLC 参数的 I/O 分配和实际状态一致。

(c) 基板

可以确认装入使用中的基板的模块状态。

即使有一个模块出错，其状态也会在模块栏里显示。

(d) PC 诊断

该功能用于确认 QCPU 的状态和错误。

7 - 72

(e) 模块的详细资料

本功能用于确认所选模块的详细资料。

有关智能型功能模块和智能型功能模块的详细资料。参见有关的说明手册。

(f) 基板信息

可以确认“整体信息”和“基板信息”。

1) 整体信息

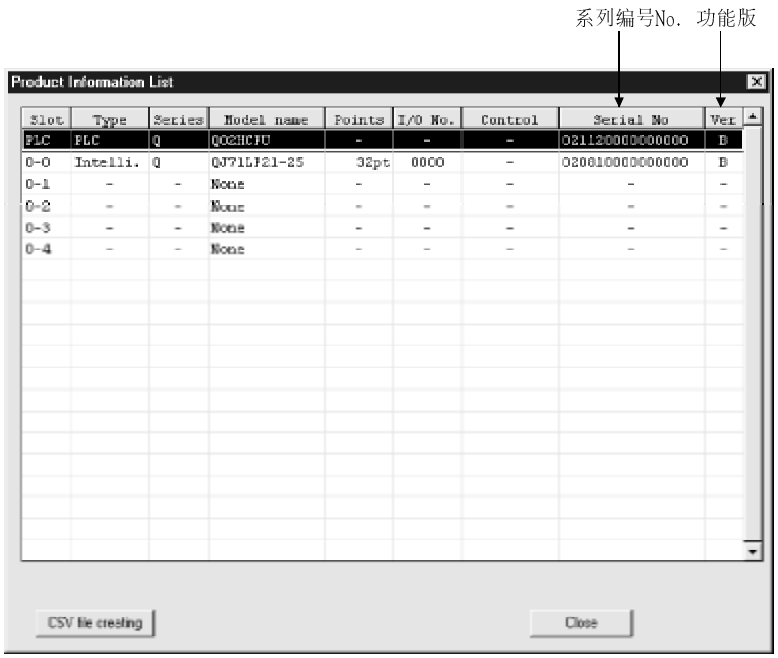
可以确认使用的基板数量和装入基板的模块数量。

2) 基板信息

可以确认基板名称，插槽数量，基板类型和装入所选基板上的模块数量。（主基板，扩展基板 **1—7**）

(g) 产品信息表

可以确认安装的 **QCPU** 模块，**I/O** 模块和智能型功能模块的个别信息。（类型，系列，模式，数量，起始 **I/O**，控制 **PLC**，系列编号 **No.**，功能版。）



7 - 73

7.19 LED（发光二极管）显示

在 QCPU 的前端有发光二极管显示 QCPU 的操作状态。每个发光二极管显示内容的详细说明如下。

7.19.1 发光二极管显示

(1) 发光二极管显示内容的详细说明如下：

LED 名称	显示说明
模式	显示 QCPU 的模式（Q 模式/A 模式） ON（绿色）：Q 模式（对外部 I/O 的强制 ON/OFF 未注册） ON（桔黄色）：A 模式 闪烁（绿色）200ms 亮□200ms 灭，对外部 I/O 的强制 ON/OFF 已注册
运行	显示 QCPU 的操作状态 接通　：RUN/STOP 开关置于“RUN”时的操作。 断开　：RUN/STOP 开关置于“STOP”时停止。 或探测到使操作停止的出错。 闪烁　：在停止时，写入参数程序，以及当设置 RUN/STOP 开关从[STOP]切换至[RUN]时。在程序写入后，执行如下操作点亮 RUN LED。 • 设置 RUN/STOP 开关从[RUN]→[STOP]→[RUN]。 • 用 RESET/L.CLR 开关使系统复位。 • 再次对 PLC 接通电源。 在参数写入后，执行如下操作点亮 RUN LED。 • 用 RESET/L.CLR 开关使系统复位。 • 再次对 PLC 接通电源。 （当参数修改后，RUN/STOP 开关设置为[RUN]→[STOP]→[RUN]时，有关智能型功能模块的参数和其它的网络参数无法反映出来。）
错误	表示 QCPU 出错探测状态。 接通　：当探测到不停止操作的自诊断错误（除了电池错误）时。（用参数模式 PLC RAS 设置将操作错误设置为“继续”。） 断开　：正常 闪烁　：当探测到停止操作的错误时。 当自动写入到标准 ROM 正常完成时。（引导 LED 也闪烁）
用户	显示 CHK 指令探测状态和信号报警器 F 状态。 接通　：当用 CHK 指令探测到错误时，或当接通信号报警器时。 断开　：正常 闪烁　：执行锁存清除时。
电池	显示 CPU 和存储器电池状态。 接通　：当探测到由于低电压引起的电池故障时。 断开　：正常
引导	表示引导操作的执行状态。 接通　：当执行完成后。 断开　：当未执行时。 闪烁　：当自动写入标准 ROM 正常完成时。（出错 LED 也闪烁）

7 - 74

(2) 熄灭 LED 的方法

用下列操作可将 LED 从接通切换到断开。（除复位操作之外。）

切断 LED 的方法	可用的 LED			
	错误	用户	电池	引导
在分辨出错误原因后，执行 LEDR 指令。	○	○	○	×
在分辨出错误原因后，操作特殊继电器 SM50 和特殊寄存器 SD50 以删除错误。（仅用于操作连续错误。）	○	○	○	×
操作特殊继电器 SM202 和特殊寄存器 SD202 来熄灭 LED。	×	○	×	○

○：有效，　　×：无效

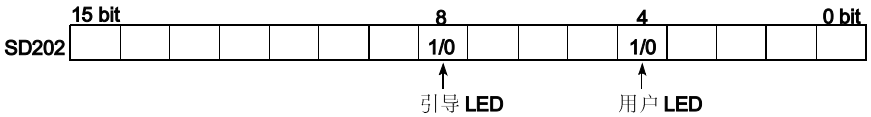
*1：特殊继电器和特殊寄存器

SM50 ·····当开关从 OFF 切换到 ON 时，与保存在 SD50 中的出错代码相对应的错误被删除。

SD50·····保存要删除的错误的出错代码。
（有关出错代码进一步的资料，参见 QCPU（Q 模式）用户手册–硬件设计和维护。）

SM202 ·····当从 OFF 切换到 ON 时，相应于 SD202 中每个位的 LED 熄灭。

SD202·····指定 LED 熄灭。（仅用户 LED 和引导 LED 可熄灭。）



在设置中 1 表示“熄灭”，而“0”表示“仍亮”。
每个 LED 的熄灭设置如下。（十六进制）

- 当二个 LED 都熄灭时：$SD202=110_H$
- 当引导 LED 熄灭时：　$SD202=100_H$
- 当用户 LED 熄灭时：　$SD202=10_H$

(3) 不显示出错 LED，用户 LED，电池和报警 LED 的方法

在第 7.19.2 节说明出错 LED，用户 LED，电池和报警 LED 有同样的优先级。
当一个 LED 的出错编号从优先级中删除后，即使发生该出错编号的错误，LED 也不会亮。
（有关设置方法，参见第 7.19.2 节的要点。）

7.19.2 优先级的设置

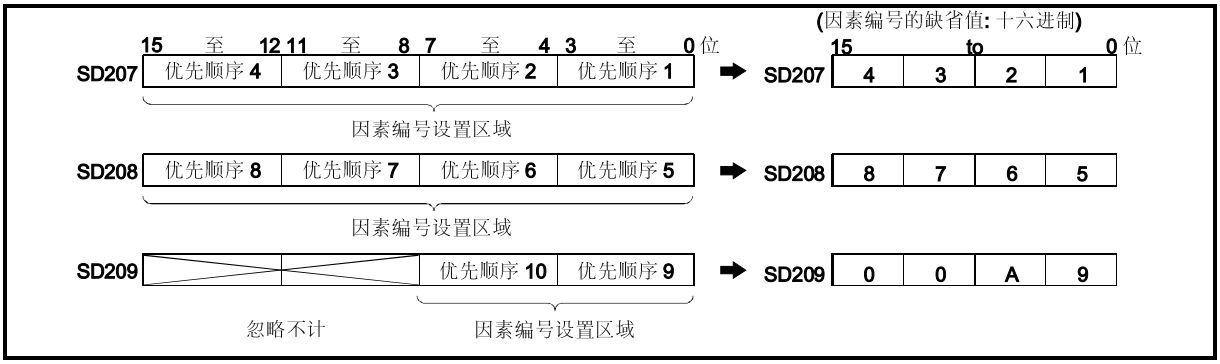
当发生可显示的多种因素时，则以下列条件进行显示：

1） 无条件显示停止错误。

2） 操作继续错误按缺省设置的优先级因素编号显示。

3） 发生同样优先级别的错误时，显示首先探测到的错误。

以下列方式用特殊寄存器 SD207 至 SD209 设置优先级：



在特殊寄存器 SD207 至 SD209 中设置因素编号的说明和缺省优先级如下：

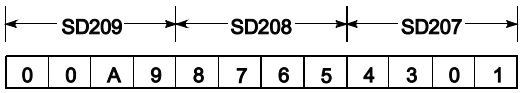
优先级	因素编号（十六进制）	说明	备注
1	1	AC DOWN	断电
2	2	UNIT VERIFY ERR. FUSE BREAK OFF SP.UNIT ERROR EX.POWER OFF	I/O 模块验证错误 保险丝熔断 特殊功能模块验证错误 外部电源切断（将来使用）
3	3	OPERATIN ERROR LINK PARA.ERROR SFCP OPE.ERROR SFCP EXE.ERROR	计算错误 链接参数错误 SFC 指令计算错误 SFC 程序执行错误
4	4	ICM.OPE.ERROR FILE OPE.ERROR	存储卡操作错误 文件访问错误
5	5	PRG.TIME OVER	恒定扫描设置时间超时 低速执行监控时间超时
6	6	CHK 指令	
7	7	信号报警器	
8	8	—	
9	9	BATTERY ERR.	
10	A	时钟数据	

要点

(1) 当在上述错误时保持 LED 熄灭，在因素编号设置区进行设置，（每 4 位），因素编号保存在相应的 SD207—SD209 中为 “0” 。

[例]

当探测到保险丝熔断错误时，要保持出错 LED 切断，在设置区域设置因素编号为 “0”，而该区域的出错编号为 “2” 。



因为因素编号 “2” 未设定，即使探测到保险丝熔断，出错 LED 仍断开。此时即将探测到因素编号为 “2” 的错误（I/O 模块验证错误或智能型功能模块验证错误。），错误 LED 仍断开。

(2) 即使 LED 设置为断开，仍然执行 SM0（诊断错误标志）“ON”，对 SM1（自诊断标志）“ON”，和 SD0（CPU 诊断错误寄存器）的出错代码存储。

7 - 77

8. 与智能型功能模块/特殊功能模块的通信

(1) 智能型功能模块/特殊功能模块的说明

QCPU 允许使用 QCPU—兼容智能型功能模块和 AnS—兼容特殊功能模块。
智能型功能模块/特殊功能模块是一种允许 QCPU 处理模拟值或不能用 I/O 模块来处理的高速脉冲。
例如，可以用智能型功能模块中的模/数转换模块将一个模拟量在使用之前先转换成一个数字量。

(2) 与智能型功能模块/特殊功能模块的通信

智能型功能模块/特殊功能模块装备有存贮器（缓冲存贮器），用来存储从外部设备接收来的数据或者输出到外部设备的数据。
QCPU 从缓冲存储器中读取数据且把数据写入到缓冲存储器中。

8.1 在 QCPU 与 Q 系列智能型功能模块之间的通信

使用以下方法使 QCPU 与智能型功能模块之间能进行信息交换：

- 用 GX—Configurator 进行初始设定和自动刷新设定
- 软元件的初始值
- FROM/TO 指令
- 智能型功能模块软元件
- 智能型功能模块的专用指令

下列表格表示与上述智能型功能模块的通信的通信时序：

用智能型功能模块的通信方式		通信时序					存贮位置*1	
		电源 ON	QCPU 复位	停止→运行	执行指令	结束处理	*2 QCPU	*3 智能
GX Configurator	初始设定	○	○	○	—	—	○	—
	自动刷新设定	—	—	—	—	○	○	—
软元件的初始值		○	○	○	—	—	○	—
FROM/TO 指令*4		—	—	—	○	—	○	—
智能型功能模块软元件*4		—	—	—	○	—	○	—
智能型功能模块的专用指令*4		—	—	—	○	—	○	—

○: 可以存储 —: 不能存储

备注

*1: 表示不管数据（由 GX Configurator 所指定的，或软元件的初始值等）是存储在 QCPU 还是在一个智能型功能模块中。
*2: 代表 QCPU 内部存储器或存储卡。
*3: “智能”代表一个智能型功能模块。
*4: 代表了使用智能型功能模块软元件的程序，FROM/TO 指令，或智能型功能模块的专用指令。

8.1.1 使用 GX Configurator 进行初始设定和自动刷新设定。

(1) 智能型功能模块的初始值设定和自动刷新设定。

安装与智能型功能模块兼容的 **GX Configurator**，使初始值设定和自动刷新设定能用 **GX Developer** 进行。

当用 **GX Developer** 指定初始值设定与自动刷新设定时，你能在没有创建与智能型功能模块通讯对程序的情况下写或读数据。而且你不必指定智能型功能模块的缓冲存储器地址，就能进行初始值设定和自动刷新设定。

(2) 用 GX Configurator 进行设定

本节介绍了设置 **A/D** 转换模块 **Q64AD** 的初始值设定和自动刷新设定的示例。

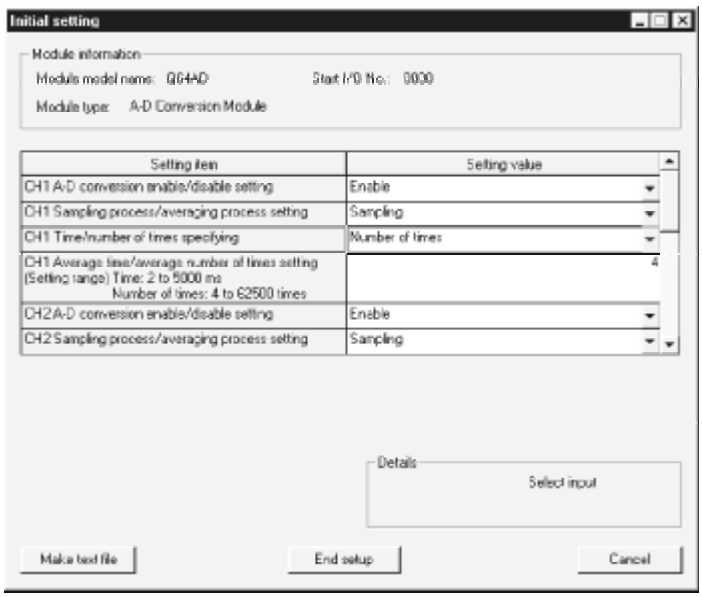
(a) 初始值设定

Q64AD 的初始值设定有以下四种设定：

- 指定 **A/D** 转换可用/禁用
- 指定采样/平均值处理
- 指定取时间平均值/取执行平均值
- 指定平均时间/平均执行

可在下列 **GX Configurator** 的初始值设定屏幕上指定 **Q64AD** 的初始值设定。

[初始值设定屏幕]



指定的初始值设定数据存储在智能型功能模块中。

8

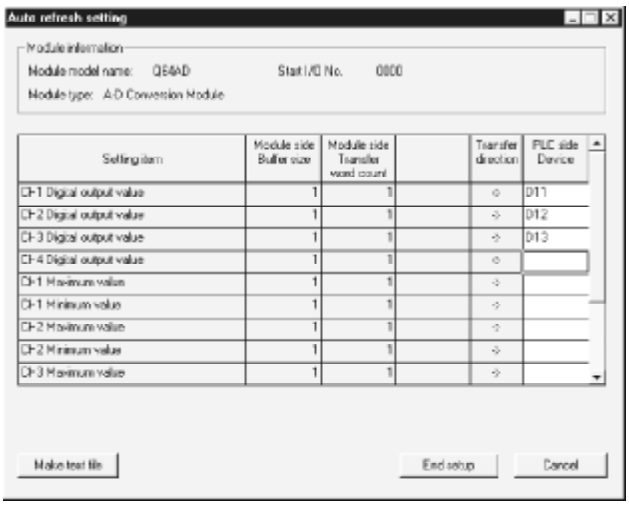
(b) 自动刷新的设定

须指定 QCPU 的软元件来存储下列用于自动刷新设定的数据。

- Q64AD 的数字输出
- Q64AD 的最大/最小值
- 出错代码

在下列 GX Configurator 的自动刷新设定屏幕上指定 Q64AD 的自动刷新设定。

[自动刷新设定屏幕]



用 QCPU 的智能型功能参数存储所指定的自动刷新设定数据。

备注

有关 GX Configurator 的详细说明，请参考所采用的智能型功能模块的手册。

8.1.2 使用软元件的初始值通信

(1) 软元件的初始值

软元件的初始值是用来指定智能型功能模块在不使用程序时的初始值设定。
当 QCPU 在接通电源、复位、或者从停止状态（STOP）切换到运行状态（RUN）时，所指定的软元件初始值将从 QCPU 写入到智能型功能模块中。

(2) 软元件初始值的指定

使用 GX Developer 的软元件存储器来指定用作软元件初始值的智能型功能模块的数据。
在 GX Developer 的软元件初始值设定中，指定作为软元件初始值的软元件的智能型功能模块软元件的使用范围。

备注

1) 有关软元件初始值，参阅 10.13.2 节。
2) 有关智能型功能模块软元件，参阅 10.5 节。

8.1.3 使用 FROM/TO 指令的通信

(1) FROM/TO 指令

在执行 **FROM/TO** 指令时，能读出存储在智能型功能模块缓冲存储器中的数据或者能把数据写入到智能型功能模块的缓冲存储器中去。

FROM/TO 指令把从智能型功能模块的缓冲存储器中读出的数据存储到指定的软元件中。

TO 指令把指定软元件的数据写入到智能型功能模块的缓冲存储器中。

备注

1) 有关 **FROM/TO** 指令的细节，请参阅以下手册

• **QCPU**（**Q** 模式）**/QnACPU** 编程手册（公共指令）

2) 有关智能型功能模块的缓冲存储器的细节，请参阅所采用的智能型功能模块的手册。

8.1.4 使用智能型功能模块的软元件的通信

(1) 智能型功能模块的软元件

智能型功能模块软元件在 **QCPU** 的程序中是代表 **QCPU** 软元件的智能型功能模块的缓冲存储器的。

它使得读出存储在智能型功能模块缓冲存储器中的数据，或者把数据写入到智能型功能模块的缓冲存储器中去成为可能。

(2) 与 FROM/TO 指令的区别

智能型功能模块的软元件可用作 **QCPU** 的软元件，且能够通过一条指令来处理从智能型功能模块中读出的数据。

这样就使整个程序节省许多步骤。

处理速度是指令执行时间与智能型功能模块存取数据时间和总和。

要点

当在程序中频繁地读取和处理智能型功能模块数据时，使用 **FROM** 指令来读取程序中某一点的数据且存贮在数据寄存器中，而不是每次都使用智能型功能模块的软元件。

否则每次执行指令时，智能型功能模块的软元件都要访问智能型功能模块，这样使程序需要更长的扫描时间。

备注

有关智能型功能模块的软元件,请参阅 **10.5** 节。

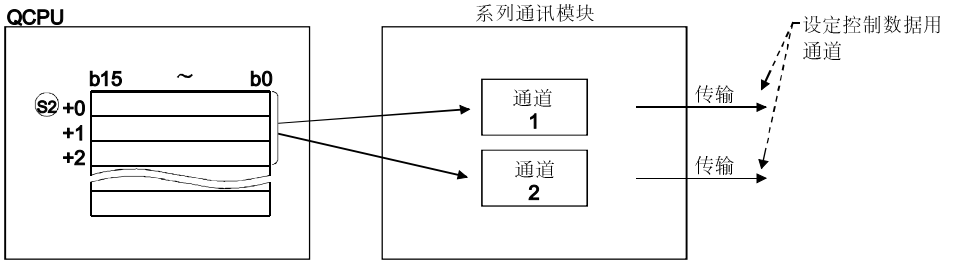
8.1.5 使用智能型功能模块的专用指令的通信

(1) 智能型功能模块组专用指令的说明

(a) 智能型功能模块的专用指令是一种使用智能型功能模块的功能来简化编程的指令。

例如：**OUTPUT** 指令，是专用于串行通信模块组的指令，可允许在没有信号交换协议的情况下，使用用户指定的信息格式来进行数据传输。

在这种情况下，不必考虑目标串接通信模块的缓冲存储器地址便可以进行通信。



(b) 完成软元件应被指定用作智能型功能模块专用指令。

当专用于智能型功能模块的指令执行完成时，所指定的完成软元件在一次扫描期内“**ON**”。当完成的软元件“**ON**”时，则对同一个智能型功能模块执行另一条智能型功能模块专用指令。

要对一个智能型功能模块使用一条以上的智能型功能模块专用指令，必须确保在完成软元件“**ON**”以后才执行下一条智能型功能模块专用指令。

(2) 注意

(a) 如果在完成软元件“**ON**”之前，执行了智能型功能模块的专用指令，且 **QCPU** 从 **RUN** 切换到了 **STOP** 状态，则当下一次 **QCPU** 切换到 **RUN** 状态时，完成软元件会在一次扫描后“**ON**”。

备注

有关智能型功能模块组专用指令和完成软元件，请参阅所采的智能型功能模块的手册。

8.2 从智能型功能模块向 QCPU 发出的请求

8.2.1 从智能型功能模块发出的中断请求

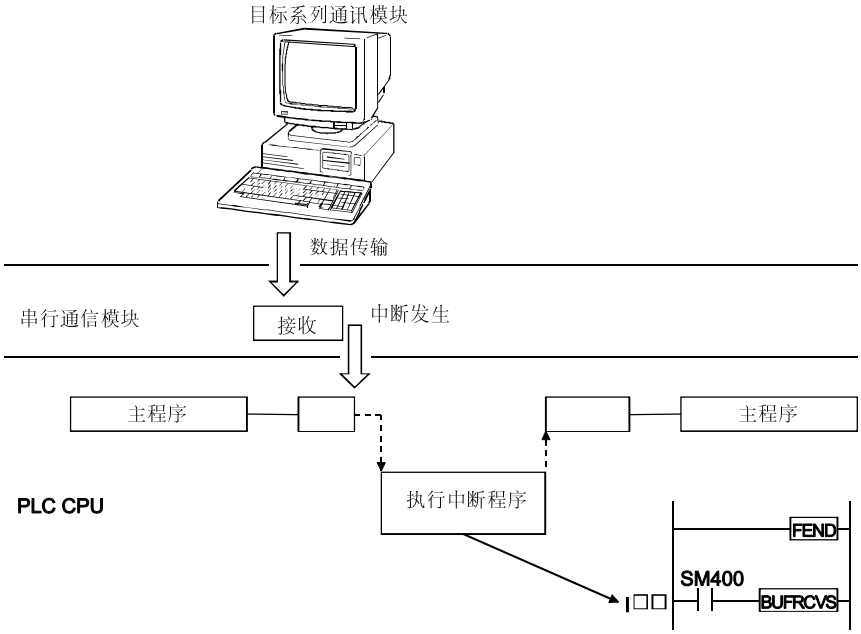
(1) 从智能型功能模块发出的中断请求

从智能型功能模块发出的中断请求使 QCPU 执行中断程序(I50 到 I255)

例如:当执行下列数据通讯功能时，串接通信模块执行用中断程序进行数据接收的处理。

- 在没有数据交换协议的通信期间的数据接收。
- 在有双方协议的通信期间的数据接收。

用中断程序处理数据接收，改进 QCPU 数据接收的速度。



(2) 设定从智能型功能模块发出的中断请求

要通过智能型功能模块的中断请求来执行中断程序，需要用 PLC 参数的 PLC 系统设定指定“智能型功能模块设定（中断指针设定）”。

你也应该指定智能型功能模块的“系统设定”。要通过智能型功能模块的中断（请求）来执行中断程序，可参阅所采用的智能型功能模块的手册。

8.3 QCPU 与 AnS 系列特殊功能模块之间的通信

下列方法使 QCPU 与 AnS 系列兼容的特殊功能模块之间实现通信：

- 软元件的初始值
- 智能型功能模块的软元件
- FROM/TO 指令

下列表格表示与上述特殊功能模块的通信方式的通信时序。

与特殊功能模块的通信方式	通信时间					存储器位置	
	电源接通	QCPU 复位	停止→运行	指令执行	结束处理	*2 QCPU	
软元件的初始值	○	○	○	—	—	○	
FROM/TO指令 *4	—	—	—	○	—	○	
智能型功能模块软元件*4	—	—	—	○	—	○	

○：可执行　　—：不可执行

备注

*1: 表示软元件初始值的数据是存贮在 QCPU 中还是存储在特殊功能模块中。
*2: 代表是 QCPU 的内部存储器或是存储器卡。
*3: “特殊”代表特殊功能模块。
*4: 代表使用智能型功能模块软元件程序或 FROM/TO 指令。

8.3.1 使用软元件初始值的通信

(1) 软元件初始值

软元件初始值是当不使用程序时，用来指定特殊功能模块的初始设定。
当 QCPU 切换至“ON”时，复位时或从 STOP 切换到 RUN 状态时，指定的软元件初始值从 QCPU 写入到特殊功能模块中去。

(2) 软元件初始值的指定

用 GX Developer 的软元件初始值设定，指定智能型功能模块软元件作为软元件初始值的软元件。

备注

1) 有关软元件初始值，参阅 10.13.2。
2) 有关智能型功能模块软元件，参阅 10.5 节。

8.3.2 使用 FROM/TO 的指令通信

(1) FROM/TO 指令

在执行 FROM/TO 指令时，能读出存储在特殊功能模块缓冲存储器中的数据，或者把数据写入到特殊功能模块的缓冲存储器中去。

FROM 指令把从特殊功能模块缓冲存储器中读出的数据存储到指定的软元件中。

TO 指令把所指定软元件的数据写入到特殊功能模块的缓冲存储器中。

备注

1) 有关 FROM/TO 指令的细节参阅下列手册。
   • QCPU（Q 模式）/QnACPU 编程手册（公共指令）。
2) 有关特殊功能模块缓冲存储器的细节，参阅正在使用的特殊功能模块手册。

8.3.3 使用智能功能模块软元件的通信

(1) 智能型功能模块的软元件

智能型功能模块软元件在 QCPU 程序中是一个代表 QCPU 软元件的特殊功能模块缓冲存储器。

它使我们能读出存储在特殊功能模块缓冲存储器中的数据。或写入数据到特殊功能模块缓冲存储器中。

(2) 与 FROM/TO 指令的区别

智能型功能模块软元件可以当作 QCPU 的软元件来处理，使之只用一条指令就能处理从特殊功能模块读出的数据。

这在整个程序中就节省了许多步骤。

处理的速度是指令执行时间和向/从智能型功能模块作存取的时间的总和。

要点

在程序中，当频繁地读出和处理特殊功能模块的数据时，使用 FROM 指令来读取在程序中某一点的数据并用数据寄存器进行存储和处理，而不是每次都使用智能型功能模块的软元件。

否则，每次执行指令时，智能型功能模块软元件访问特殊功能模块都会导致程序需要较的扫描时间。

备注

有关智能型功能模块软元件，参阅 10.5 节。

8.3.4 对特殊功能模块进行快速访问的影响和防范措施

(1) 对特殊功能模块进行快速访问的影响

由于 QCPU 扫描时间的增加，对 AnS-兼容特殊功能模块，执行 FROM/TO 指令有一些限制。

从特殊功能模块读出/写入数据的程序中，每次扫描时，执行 FROM/TO 指令可能会产生下列影响。

(a) 为 FROM/TO 指令指定优先级的特殊功能模块

1) 在执行 FROM/TO 指令时，特殊功能模块停止其处理而首先处理 FROM/TO 指令，结果使特殊功能模块的处理时间延长。
造成特殊功能模块的警戒定时器出错。

2) 下面列出的是为 FROM/TO 指令指定优先级的特殊功能模块。
- A1S64AD, A1S68AD
- A1S62RD3, A1S62RD4
- A1S68DAV, A1S68DAI
- A1S68TD
- A1SD75P1(-S3), A1SD75P2(-S3), A1SD75P3(-S3)
- A1SD75M1, A1SD75M2, A1SD75M3

(b) 暂停处理 FROM/TO 指令的特殊功能模块

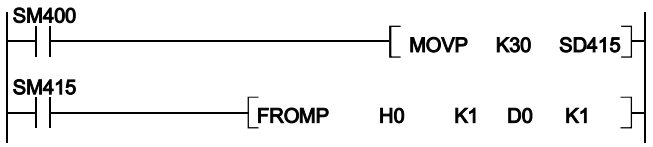
1) 甚至在执行 FROM/TO 指令时，特殊功能模块会继续其处理，而暂不处理 FROM/TO 指令，直到处理完成。
结果由于等待特殊功能模块处理完成的时间使扫描时间变得更长。

2) 下面列出的是暂停 FROM/TO 指令的特殊功能模块。
- A1S63ADA, A1S66ADA
- A1SD61, A1SD62, A1SD62D, A1SD62E
- A1SD70, A1SD71-S2, A1SD71-S7
- A1SJ71PT32-S3, A1SJ71T32-S3
- A1SD51S
- A1SJ71ID1-R4, A1SJ71ID2-R4

(2) 对进行快速访问特殊功能模块影响的防范措施

与 QCPU 配合使用一块特殊功能模块，来调整 SM415（2n-ms 时钟）和 SD415（2n-ms 时钟设置）的执行时间。

SD415 的初始化值是“30”，当 SD415 用于 FROM/TO 指令的互锁时，每隔 120ms 执行 FROM/TO 指令。



备注

1) 要改变 SM415 的时钟值，将新的值存储在 SD415 中。
2) 有关 SM415 的细节参阅附录 1，有关 SD415 的细节参阅附录 2。

9. 参数表

在 QCPU 的步骤中使用的参数有两种类型：操作 PLC 时使用的"PLC 参数"和连接到 MELSECNET/H 或 CC-Link 系统时使用的“网络参数”。

在配置多个 PLC 系统到功能版 B QCPUs 的 PC 参数中时，已经加入了下列所要求的项目。

- 多个 PLC 的设置
- 控制 PLC 的设置（I/O 分配）

本章列出了 GX-Developer 使用的 PLC 参数表和网络参数表。
有关每一个项目设定的细节，参阅有关章节或手册。
有关 GX- Developer 的设定步骤，参见 GX- Developer 操作手册。

用 GX-Developer 写入的参数，在下列情况下在 QCPU 内有效：

- 当 PLC 接通电源时
- 当 QCPU 复位时
- 当 QCPU 从 STOP 切换到 RUN 状态时

然而，在下列情况下，PC 参数的 I/O 配置开关设定和网络参数从 QCPU 传输到一个指定的智能型功能模块，在智能功能模块中是有效的：

- 当 PLC 接通电源时
- 当 QCPU 复位时

当 QCPU 从 STOP 切换到 RUN 状态时，不会从 QCPU 向指定的智能型功能模块传输 PC 参数的 I/O 配置开关设定和网络参数。
当 PC 参数的 I/O 配置开关设置和网络参数已更改时，要再次接通 PLC 的供电开关（ON-OFF-ON）或将 QCPU 复位。

9 - 1

表 9.1 参数表

项目			说明	
PLC 名称设定			指定所使用的 QCPU 的标贴和注解。 这些设定不影响 CPU 的运行。	
	标贴		指定标贴设定(名称和使用)。	
	注解		指定注解的设定。	
PLC 系统设定			这些是使用 QCPU 所需要的设定。 PLC 控制可使用缺省值。	
	定时器设定	低速定时器	指定低速/高速定时器的设定。	
		高速定时器		
	运行—暂停接触器		指定控制 QCPU 的运行/暂停运行的接触器。	
	远程复位		GPP 功能的远程复位操作起作用或不起作用。	
	停止—运行输出模式		指定在停止/运行切换时的输出(Y)模式。	
	浮点算术运算		指定是否执行双倍精度的浮点算术运算。	
	智能型功能设定 (中断指针设定)		指定中断指针的分配(I50 到 I255)，起始 I/O 地址和一个智能型功能模块的起始 SI 号。	
	公共指针编号		指定公共指针“第一个编号”。	
	空闲槽点的编号		指定在基本模块/扩展基本模块上空槽点的编号。	
	系统中断设置	中断计数器	指定中断计数器的“第一编号”。	
		固定循环周期	指定执行中断指针（128-131）的时间间隔。	
	中断程序/固定扫描程序设定		指定是否高速执行中断程序。	
	模块同步化		指定是否使 QCPU 的启动与智能型功能模块的启动同步。	
	与 A 系列 CPU 的兼容性		指定是否使用 MELSEC-A 系列特殊继电器/特殊寄存器 （SM/OOO/SD/OOO 到 SM1299/SD1299）。	
PLC 文件设定			指定 QCPU 中使用的各种文件。	
	文件寄存器		指定程序中使用的文件寄存器的文件。	
	指令注解文件		指定程序中使用的注解文件。	
	软元件初始化值		指定 QCPU 中使用的软元件初始值的文件。	
	本地软元件文件		指定程序中使用的本地软元件的文件。	

	缺省值	设定范围	参考章节
	—	—	—
	未设定	最多 10 个字符	—
	未设定	最多 64 个字符	—
	—	—	—
	100ms	1ms 到 1000ms（1 毫秒为单位）	10.2.10 节
	10.0ms	0.1ms 到 100ms	10.2.10 节
	未设定	×0 到×1FFF	7.6.1 节
	禁	禁用/可用	7.6.3 节
	先前状态(在停止之前产生一个(Y)输出的状态)	在停止之前产生一个(×)输出的状态/清除输出(1 次扫描后输出)。	7.4 节
	执行双倍精度的内部算术运算	选择/不选择执行双倍精度的内部算术运算。	4.8.4 节
	未设定	150-1255，起始 I/O 地址，起始 SI 号	10.10 节
	未设定	P0 到 P4095	10.9.2 节
	16 点	16/32/64/128/256/512/1024 点	5.6.1 节
	未设定	C0 到 C22722(计数器设定点可设定到 256)	10.2.11 节
	I28: 100.0 ms I29: 40.0 ms I30: 20.0 ms I31: 10.0 ms	0.5 到 1000ms(0.5ms 为单位)	10.10 节
	不能高速执行	可用/禁用高速执行	4.1.3 节 4.2.5 节
	一个智能型功能模块的启动同步化	是/否使智能型功能模块的启动同步	—
	使用 SM1000/SD1000 以后的特殊继电器/特殊寄存器	是/否在 SM1000/SD1000 以后使用特殊继电器/特殊寄存器。	10.3.2 节 10.3.3 节
	—	—	—
	不用	• 不用 • 将相同的文件作为程序使用 • 用指定文件	10．7 节
	不用	• 不用 • 将相同的文件作为程序使用 • 用指定文件	—
	不用	• 不用 • 将相同的文件作为程序使用 • 用指定文件	10.13.2 节
	不用	• 不用 • 用指定文件	10.13.1 节

表 9.1 参数表(续)

项目			说明	
PLC RAS 设定			这些设定用于 RAS 功能。	
	WDT 设定	WDT 设定	设定 QCPU 警戒定时器。	
		初始化执行监视时间	设定用于初始执行型程序使用的警戒定时器。	
		低速执行监视时间	设定用于低速执行型程序使用的警戒定时器。	
	出错时的运行模式		指定检测到错误时建立的 QCPU 运行模式。	
	错误检查		指定是否检测一个特定的错误。	
	恒量扫描		指定恒定扫描时间。	
	低速执行型程序的执行时间		指定在执行每一次扫描时，低速执行程序的时间设定。	
	故障注册		指定 QCPU 故障记录的存储指定。	
软元件设定			这些设定指定每一个软元件，锁存范围和本地软元范围的点的编号。	
	软元件点的编号		指定所使用的软元件点的数目。	
	锁存(1)范围 (锁存清除键可用)		指定能使用锁存清除键的锁存范围。	
	锁存(2)范围 (锁存清除键禁用)		指定不能使用锁存清除键的锁存范围。	
	本地软元件的设定		指定用作本地软元件的软元件范围。	
程序设定			指定程序的名称且把多个程序写入到 QCPU 上的执行条件。	
引导文件设定			指定引导运行程序所须要的文件设定。	
SFC 设定			指定 SFC 程序的启动模式，启动条件和在程序块停止时的输出模式，以便使用 SFC 程序	
I/O 配置			指定每一个系统模块的安装状态。	
	I/O 配置		指定模块型号，型号名称，点数，起始 I/O 地址/起始 XY)。	
	基板设定		指定基板模块的型号名称，电源模块扩展电缆和槽的编号。	
X/Y 分配检查			使用户能检查I/O的分配，MELSECNET/ETHERNET 设定和 CC-Link 设定。	
MELSECNET/ETHERNET 设定			指定 MELSECNET/ETHERNET 的网络参数。	
CC-Link 设定			定义 CC-Link 的参数。	

	缺省值	设定范围	参考章节
	—	—	—
	200ms	10ms-2000ms(10ms 为单位)	4.2.2 节
	未设定	10ms-2000ms(10ms 为单位)	4.2.1 节
	未设定	10ms-2000ms(10ms 为单位)	4.2.3 节
	停止	停止/继续	7.1.5 节
	检查	检查/不检查	7.15 节
	未设定	0.5ms-2000ms(0.5ms 为单位)	7.2 节
	未设定	1ms-2000ms	4.2.3 节
	存储在程序存储器中	存储在程序存储器/存储在指定的历史文件中。	7.16 节
	—	—	—
	X: 8 k 点 Y: 8 k 点 M: 8 k 点 L: 8 k 点 B: 8 k 点 F: 2 k 点 SB: 2 k 点 V: 2 k 点 S: 8 k 点 T: 2 k 点 ST: 0 k 点 C: 1 k 点 D: 12 k 点 W: 8 k 点 SW: 2 k 点	X(8K 点)，Y(8K 点)，S(8K 点)，SB(2K 点)和 SW(2K 点)是固定的。 包括以上的点(3,7K 字),可使用全范围 29K 字。 • 一个软元件最大 32K 点。 • 位软元件的总数：最大 64K 点。	10.1 节 10.2 节
	未设定	对 B,F,V,T,ST,C,D,W 每个软元件只指定 1 个范围。	7.3 节
	未设定	对 L,F,V,T,ST,C,D,W 每个软元件只指定 1 个范围。	7.3 节
	未设定	对:M,V,T,ST,C,D 每个软元件只指定一个范围。	10.13.1 节
	未设定	程序名称，执行类型(用于固定扫描执行型的固定扫描)，文件使用 I/O 刷新设定。	4.2 节
	未设定	类型，数据名称，目标驱动器。	6.6 节
	—	参见 QnACPU/QCPU(Q 模式)编程手册（SFC 版）。	—
	—	—	—
	未设定	参见：SW4D5C-GPPW 用于 Windows 软件包的 GPP 功能操作手册。	5.6 节
	未设定	参见：SW4D5C-GPPW 用于 Windows 软件包的 GPP 功能操作手册。	5.3 节
	—	—	—
	未设定	参见 Q 兼容 MELSECNET/10H 和 ETHERNET 手册。	—
	未设定	参考 CC Link 手册。	—

10 软元件

本章叙述了可以在 QCPU 中使用的所有软元件。

10.1 软元件列表

能使用于 QCPU 的软元件的名称和数据范围均列出于以下的表 10.1 中。

表 10.1 软元件列表

分类	类型	软元件名称	缺省值		指定设定范围的参数	参考章节
			点的数目	使用范围		
内部用户软元件	位软元件	输入*3	8192 点	X0-X1FFF	可以在小于等于 29K 字的范围内改变. *3	10.2.1
		输出*3	8192 点	Y0-Y1FFF		10.2.2
		内部继电器	8192 点	M0-M8191		10.2.3
		锁存继电器	8192 点	L0-L8191		10.2.4
		信号报警器	2048 点	F0-F2047		10.2.5
		边沿继电器	2048 点	V0-V2047		10.2.6
		步进继电器*3	8192 点	S0-S511/block		10.2.9
		通信特殊继电器*3	2048 点	SB0-SB7FF		10.2.8
		通信继电器	8192 点	B0-B1FFF		10.2.7
	字软元件	定时器*1	2048 点	T0-T2047		10.2.10
		积算定时器*1	0 点	(ST0-ST2047)		10.2.10
		计数器*1	1024 点	C0-C1023		10.2.11
		数据寄存器	12288 点	D0-D12287		10.2.12
		通信寄存器	8192 点	W0-W1FFF		10.2.13
		通信特殊寄存器*3	2048 点	SW0-SW7FF		10.2.14
内部系统软元件	位软元件	功能输入	5 点	FX0-FX4	不可	10.3.1
		功能输出	5 点	FY0-FY4		10.3.1
		特殊继电器	2048 点	SM0-SM2047		10.3.2
	字软元件	功能寄存器	5 点	FD0-FD4		10.3.1
		特殊寄存器	2048 点	SD0-SD2047		10.3.3
通信直接软元件	位软元件	通信输入	8192 点	Jn\X0-Jn\X1FFF	不可	10.4 节
		通信输出	8192 点	Jn\Y0-Jn\Y1FFF		
		通信继电器	16384 点	Jn\B0-Jn\B3FFF		
		通信特殊继电器	512 点	Jn\B0-Jn\B1FF		
	字软元件	通信寄存器	16384 点	Jn\W0-Jn\W3FFF		
		通信特殊寄存器	512 点	Jn\SW0-Jn\SW1FF		

分类	类型	软元件名称	缺省值		指定设定范围的参数	参考章节
			点的数目	使用范围		
智能功能模块软元件	字软元件	缓冲寄存器	65536 点	Un/G0-Un/G65535*2	不可能	10.5
变址寄存器	字软元件	变址寄存器	16 点	Z0-Z15	不可能	10.6
文件寄存器	字软元件	文件寄存器	0 点	—	0-1018K 点 (1K 为单位)	10.7
嵌套	—	嵌套结构	15 点	N0-N14	不可能	10.8
指针	—	指针	4096 点	P0-P4095	不可能	10.9
		中断指针	256 点	I0-I1255		10.10
其它	位软元件	SFC 程序块	320 点	BL0-BL319	不可能	10.11.1
		SFC 传送软元件	512 点	TR0-TR511		10.11.2
	—	网络编号	256 点	J1-J255		10.11.3
		I/O 地址	—	U0-UFF		10.11.4
常数	—	十进制常数	K-2147483648-K2147483647			10.12.1
		十六进制常数	H0-HFFFFFFFF			10.12.2
		常数	E+(-)1.17549—38 到 E+(-)3.40282+38			10.12.3
		字符串常数	“ABC”和“123”			10.12.4

备注

*1: 定时器,积算定时和计数器，位软元件是用于“点数”和“线圈”，而字软元件是用于“当前值”。

*2: 可使用的实际点数是依照智能/特殊模块改变的。
有关缓冲存储器的点数请参考智能/特殊功能模块的手册。

*3: 输入；输出，步进继电器，通信特殊寄存器，保持它们的缺省值，不能改变。

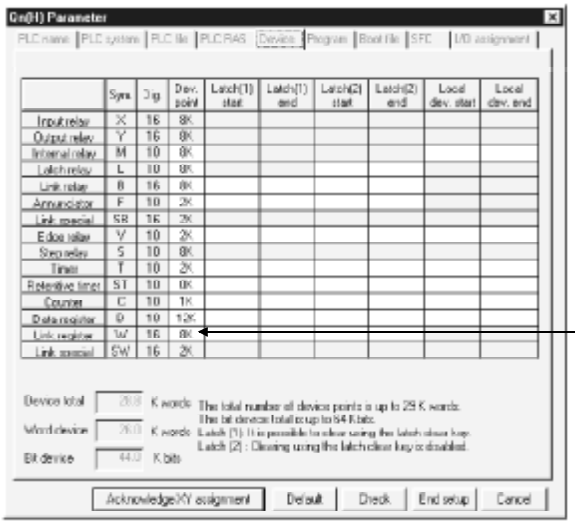
10 - 2

10.2 内部用户软元件

内部用户软元件可用于各种用户应用。

内部用户软元件的可使用点数设定是事先（缺省值）指定的。
然而这个设定可以通过 **PLC** 参数软元件的设定来改变。

[软元件设定屏幕**]**



缺省值
"Dev. point" 可以在括号中显示
"Dev. point" 值的软元件处改变。

(1) 内部用户软元件的设定范围

所有的 **QCPU** 内部用户软元件除了输入（**X**），输出（**Y**），步进继电器（**S**），特殊通信继电器和特殊通信寄存器软元件以外，所使用的点数目是可以在 **29K** 字（包括用于内部用户软元件）范围内通过 **PLC** 参数软元件的设定来改变的 **3.7K** 字。

以下讨论当作这样的改动时，要考虑的项目：

(a) 设定范围

1) 指定软元件的点数是以 **16** 点为单位的。

2) 对一种软元件最多只能指定 **32K** 点。

内部继电器，锁存继电器，信号报警器，边沿继电器，通信继电器，特殊通信继电器，步进继电器，定时器，计算定时器和计数器的最大的总点数是 **64K** 点。

对于定时器，积算定时器和计数器，一个点是作为二个点来计算的（**1** 用于线圈，**1** 用于接触器）。

(2) 存储器的容量

使用以下公式来计算用户内部软元件的存储器的容量。

3.7+（位软元件容量）**+**（字软元件容量）**+**（定时器，积算定时器和计数器容量）**≤29K**

(a) 位软元件：

对于位软元件，**16** 点作为一个字来计算。

$$[\text{位软元件容量}] = \frac{(\text{M+L+F+V+B 总点数})}{16} \text{(字)}$$

10 - 3

(b) 定时器**(T),**积算定时器**(ST),**和计数器**(C):**

对于定时器，积算定时器和计数器，**16** 点作为 **18** 字来计算。

$$(\text{定时器, 积算定时器, 计数器容量}) = \frac{(\text{T, ST, C 的总点数})}{16} \times 18\ (\text{字})$$

(c) 字软元件：

对于数据寄存器（**D**）和通信寄存器（**W**），**16** 点作为 **16** 字来计算。

$$(\text{字软元件容量}) = \frac{(\text{D, W 的总点数})}{16} \times 16\ (\text{字})$$

注意

(1) 当一个内部用户软元件的“可使用的点的数”设定改变时，就不能使用在以前的设定条件所创建的下列文件。

- 顺控程序
- **SFC** 程序

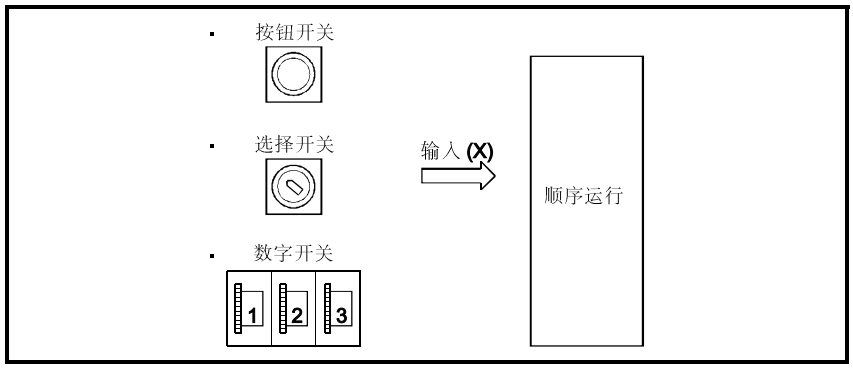
在改变设定以后，必须从 **QCPU** 将顺控程序和 **SFC** 程序读入 **GX Developer**，然后再重新写回到 **QCPU** 中去。

10 - 4

10.2.1 输入（X）

(1) 定义

(a) 输入是从外围设备通过按钮开关，选择开关，限位开关，数字开关等来向 QCPU 发送的数据或命令。



(b) 输入点是 QCPU 内部的 Xn 虚拟继电器，并且使用 Xn'S 常开接触器或常闭接触器的程序。

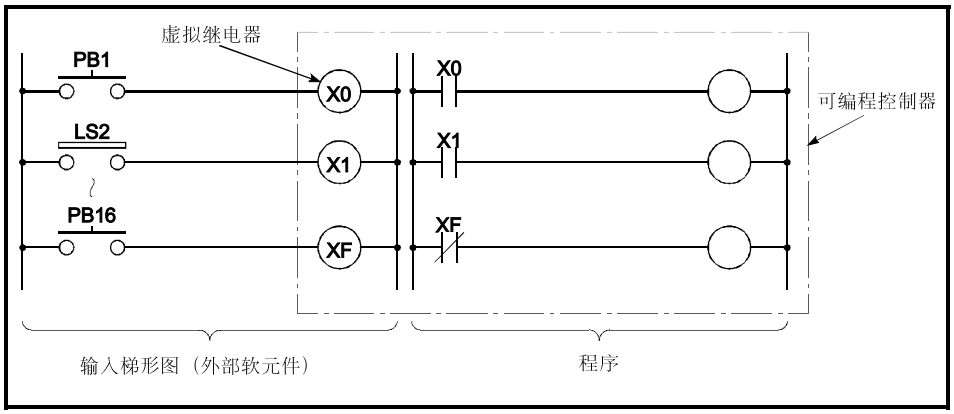


图 10.1 输入(X)

(c) 程序中所使用的 Xn 常开接触器和常闭接触器的数目是没有限制的。

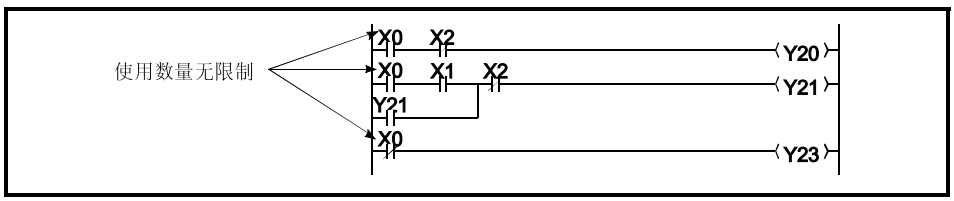


图 10.2 程序中使用的输入(X)

(2) 读出输入

(a) 有二种类型的输入“刷新输入”和“直接访问输入”

1) 刷新输入是使用刷新模式从输入模块读出 ON/OFF 数据。*1
在顺控程序中将这些输入表示为“X□”。
例如：一个“100”输入表示为“X100”。

2) 直接访问输入是使用直接模式从输入模块读出 ON/OFF 数据。*2
在顺控程序中将这些输入表示为“DX□”。
例如：一个“100”输入表示为“DX100”。
可以用以 1 点为单位的输入的 LD/AND/OR 指令直接访问输入。

(b) 刷新输入和直接访问输入之间的区别

对直接访问输入，输入模块是由执行的指令进行直接访问。因而处理的速度比刷新输入慢。

此外，直接访问输入只能用作输入模块和智能/特殊功能模块（安装在基板单元和扩展基板单元智能/特殊功能模块上）使用的输入。

刷新输入与直接访问输入的差别列出于以下的表 10.2 中。

表 10.2 刷新输入与直接访问的区别

项目		刷新输入	直接访问输入
处理速度(μs) (LD X/DX)	Q02CPU	0.079	基板:8.0 扩展基板:8.0
	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.034	基板:4.0 扩展基板:4.8
安装在基板/扩展基板上的输入模块		可使用	可使用
安装在基板/扩展基板的智能功能模块的输入			
安装在基板/扩展基板 I/O 通信模块上的输入			
用于 MELSECNET/H 网络系统的输入		可使用	不可使用

备注

*1: 参见 4.3.1 节有关刷新模式的细节。

10 - 6

(c) 刷新输入和直接访问输入可以指定为相同的输入号。
如果在用作直接访问输入以后用作刷新输入。系统将按照在直接访问输入时读出的开/关数据来运行。

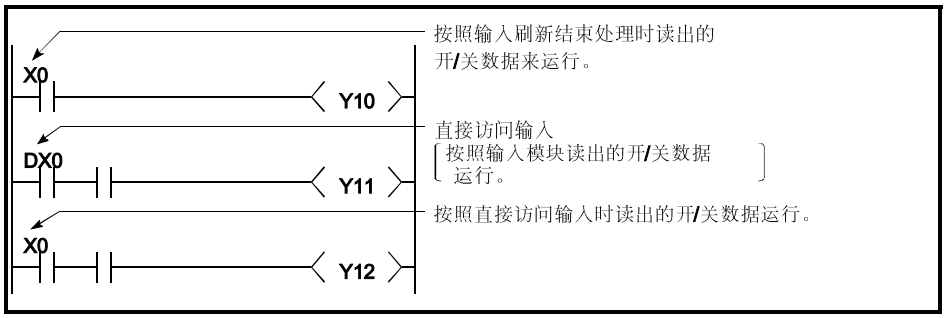
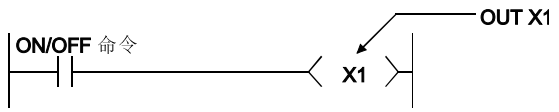


图 10.3 刷新输入和直接访问输入

POINT

(1) 当调试一个程序时,可以按如下的叙述将输入(X)设置为 ON/OFF。

• OUT Xn 指令



• GX Configurator 试运行

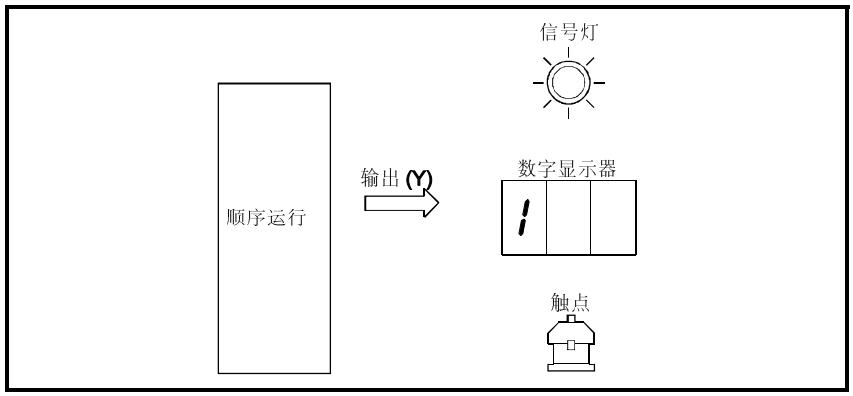
(2) 通过设置 CC-LINK 自动刷新对于 CC-Link 可以指定输入（X）作为 RX 刷新（在 QCPU 一侧）的目的软元件。

10 - 7

10.2.2 输出（Y）

(1) 定义

(a) 输出是程序控制的结果，输出到外部的目标（电磁线圈，电磁开关，信号灯，数字显示器等）。



(b) 输出发生在常开接触器或相当的器件上。

(c) 程序中所使用的输出常开/常闭接触器的数目是没有限制的。

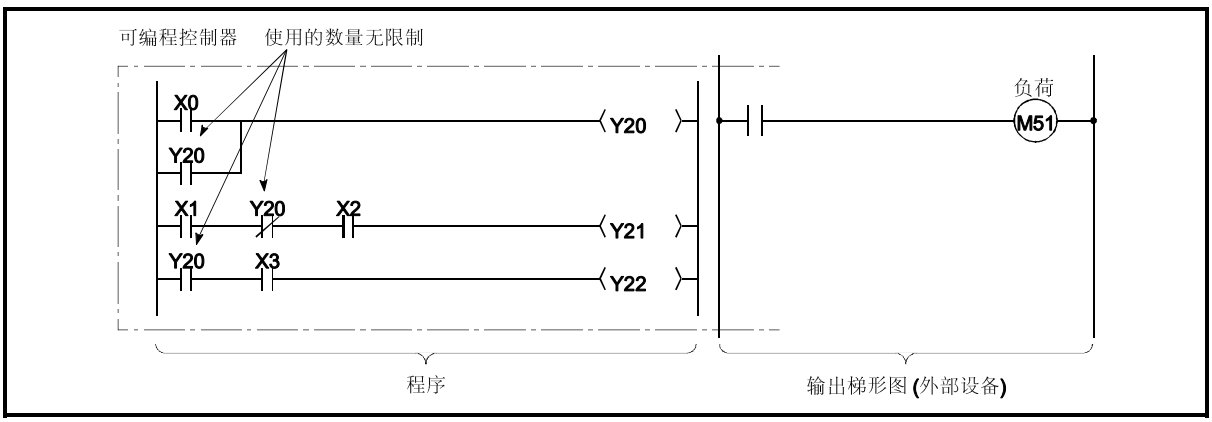
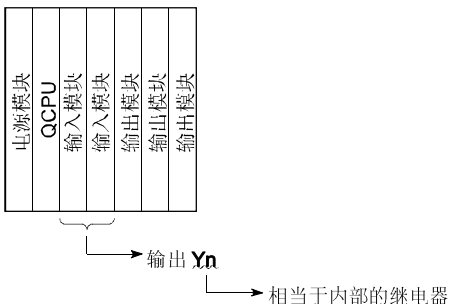


图 10.4 输出（Y）运行

(2) 使用输出作为内部继电器（M）

“Y”输出相应于空的插槽和安装有输入模块能起内部继电器作用的插槽。



(3) 输出方法

(a) 有二种输出类型：“刷新输出”和“直接访问输出”。

1) 刷新输出是输出到正使用刷新模式的输出模块的 ON/OFF 数据。*1
在顺控程序中，表示这些输出为“Y□”。
例如：一个“100”输出表示为“Y100”。

2) 直接访问输出是输出到使用直接模式的输出模块的 ON/OFF 数据。*2
在顺控程序中将这些输出指定为“DY□”。
例如：一个“100”输出表示为“DY100”。

(b) 刷新输出和直接访问输出之间的区别

直接访问输出，通过执行指令来直接访问输出模块。因而处理的速度比刷新输出慢。

刷新输出处理指令须要比直接访问输出更长的时间。

而且，直接访问输出只能用作安装在输出模块和智能/特殊功能模块（基板单元和扩展基板单元上的输出模块和智能/特殊功能模块）所使用的输出。

下列的表 10.3 表明了刷新输出与直接访问输出的差别

表 10.3 刷新输出和直接访问输出的差别

项目		刷新输入	直接访问输入
处理速度(μs) (输出 Y/DY)	Q02CPU	0.158	基板:8.0 扩展基板:8.8
	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.068	基板:4.0 扩展基板:4.8
安装在基板/扩展基板的输出模块		可使用	可使用
安装在基板/扩展基板的智能功能模块的输出			
安装在基板/扩展基板的 I/O link 模块的输出			
用于 MELSECNET/H 网络的输出		可使用	不可使用

备注

*1: 有关刷新模式的细节,参见 4.3.1 节。

10 - 9

10.2.3 内部继电器(M)

(1) 定义

(a) 内部继电器是不能由可编程控制器的内部锁存(后备存储器)来锁存的辅助继电器。

在下列时刻，所有的内部继电器都被切换到 OFF(关)

- 当电源从 OFF 到 ON 时
- 当发生 QCPU 复位时
- 当执行 QCPU 锁存清除操作时

(b) 在程序中使用的接触器的数目(常开接触器，常闭接触器)是没有限制的。

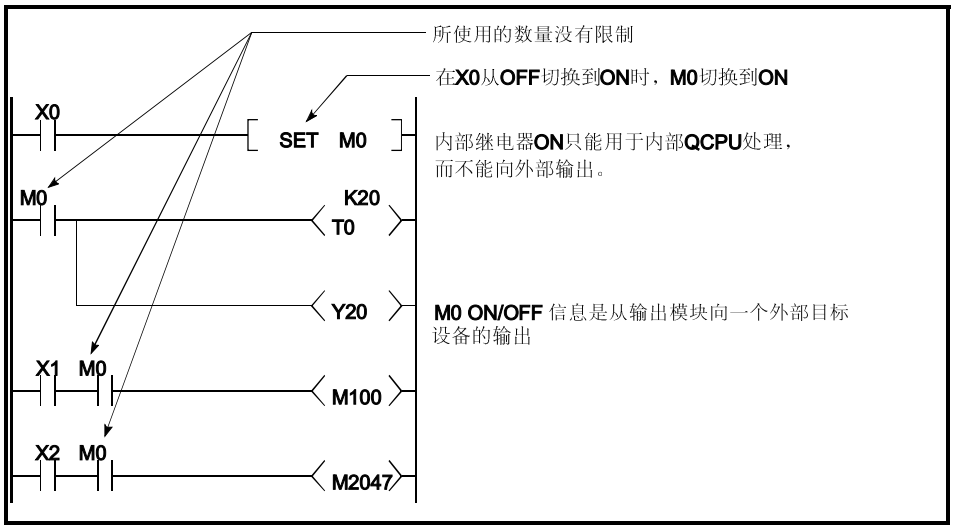


图 10.5 内部继电器

(2) 向外部输出的步骤

输出(Y)是用于输出顺控程序操作结果到一台外部目标设备的。

备注

1) 锁存继电器(L)应该用于需要锁存时(后备存贮器)。
有关锁存继电器的细节，参见 10.2.4 节。

10.2.4 锁存继电器(L)

(1) 定义

(a) 锁存继电器是可由可编程控制器的内部锁存(后备存储器)来锁住的辅助继电器。即使在下列情况下，锁存继电器的操作结果（ON/OFF 讯号）也被保存起来。

- 当电源从 OFF 到 ON 切换时
- 当发生 QCPU 复位时
- 锁存是由 QCPU 的电池支持供电的

(b) 锁存继电器可由 QCPU 的锁存清除来切换到关（OFF）。然而如果已经用软元件设定参数将该锁存继电器定为锁存清除无效，则不能通过 RESET/L.CLR 开关来使该锁存继电器切换到 OFF(关)。

(c) 程序中使用的接触器的数目(常开接触器，常接触)是没有限制的。

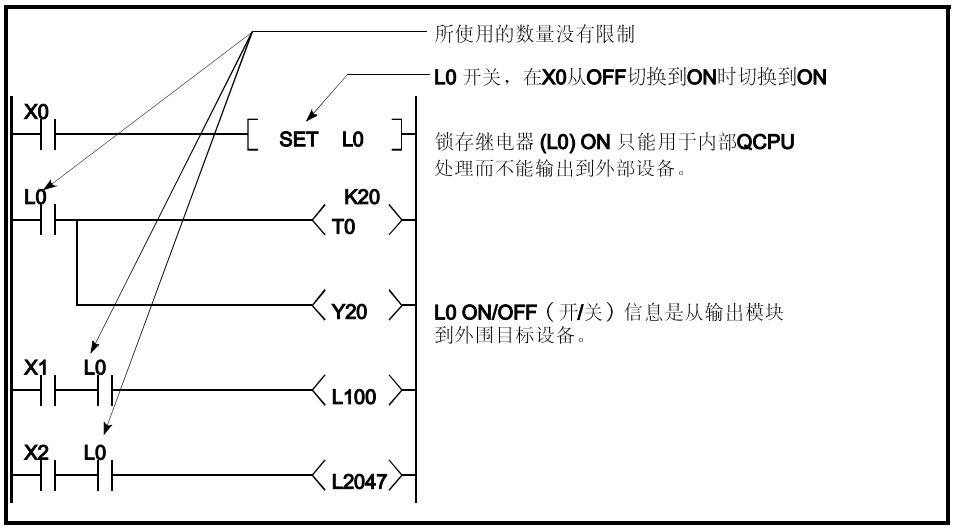


图 10.6 锁存继电器

(2) 对外部输出的步骤

输出(Y)是用于输出顺控程序操作结果到外围目标设备的。

备注

当不需要锁存(后备存储器)时，应使用内部继电器(M)。
有关内部继电器的细节参见 10.2.3 节。

10.2.5 信号报警器(F)

(1) 定义

(a) 信号报警器是用户在错误检测程序中使用的内部继电器。

(b) 当信号报警器切换到 ON 时，特殊继电器(SM62)切换到 OFF。接通的信号报警器编号的数量都存储在特殊的寄存器中（SD62-SD79）。

- 特殊继电器 :SM62..... :只要有一个信号报警器接通,它就接通。
- 特殊寄存器 :SD62 ..... :SD62 首先接通的第一个信号报警器的编号存储在这里。
  SD63 ..... 接通的信号报警器的数目存储在这里。
  SD64 至 SD79 ..... 信号报警器的编号按照它们接通的顺序存储在这里。
  （同一信号报警器存储在 SD62 和 SD64）。

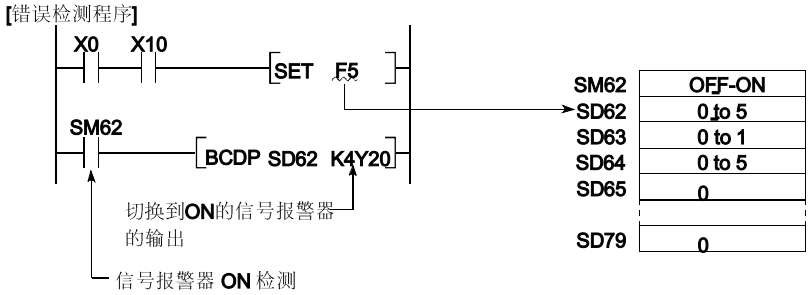
存储在 SD62 中的信号报警器编号也寄存在“错误历史区”。

(c) 当特殊继电器（SM62）接通时，通过监视特殊寄存器（SD62-SD79），使用出错检测程序中的，信号报警器使用户能检查是否存在错误和检查错误内容（信号报警器编号）。

举例

以下是输出"ON"的信号报警器 (F5) 的编号的程序。

[错误检测程序]



10 - 12

(2) 信号报警器 ON 的步骤

(a) 信号报警器 ON 的步骤

信号报警器的运行可由 SET F□和 OUT F□指令来控制

1) SET F□指令是只在输入条件脉冲的前沿（OFF→ON）使信号报警器接通到 ON 的。并且当输入条件切换到 OFF 时仍保持信号报警器 ON。
在使用许多信号报警器的地方，OUT F□指令可用来加快扫描的速度。

2) OUT F□指令可以使信号报警器 ON/OFF,但它的 SET F□指令须要更长的时间，如果信号报警器是用 OUT F□指令来切换到 OFF，这就须要执行一条 RSTF□指令或 LEDR 指令。用 SETF□指令来使信号报警器切换到 ON。

注意
(1) 如果用除了 SETF□和 OUTF□指令以外的任何其它方法来使信号报警器切换到 ON，信号报警器的功能就如同内部继电器一样。(不要接通 SM62，信号报警器的编号不存储在 SD62，SD64-SD79 中)

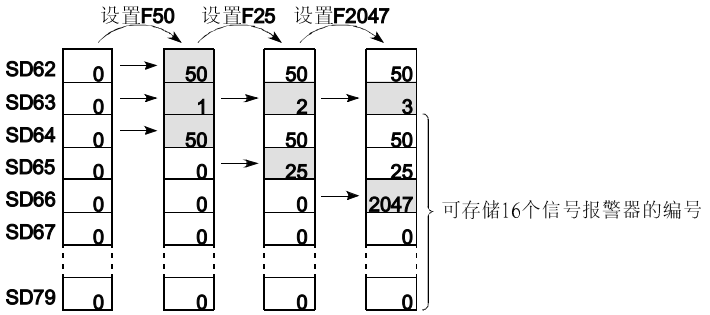
(b) 信号报警器 ON 时的处理

1) 存储在特殊寄存器(SD62-SD79)中的数据。

a) 切换到 ON 的讯号报警器的编号是依次存储在 SD62-SD79 中。

b) 存储在 SD64 的信号报警器的编号存储在 SD62。

c) “1”加到 SD63 的数值上。



2) QCPU 的处理

在 QCPU 面板上的“USER”用户指示灯发光。

(3) 信号报警器切换到 OFF 的步骤和处理内容

(a) 信号报警器切换到 OFF 的步骤

信号报警器可以用 RSTF□，LEOR，BKRST 和 OUT□指令来切换到 OFF。

1) 由 SETF□指令切换到 ON 的信号报警器的编号可由 RSTF□指令来切换到 OFF。

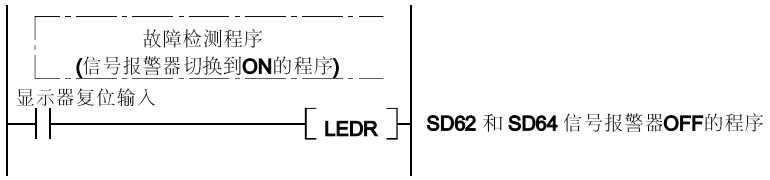
2) LEDR 指令是用来将存储在 SD62 和 SD64 中的信号报警器编号切换到 OFF。

3) 如果你想要切换一个特定范围内的所有的信号报警器编号切换，你可用 BKRST 指令。

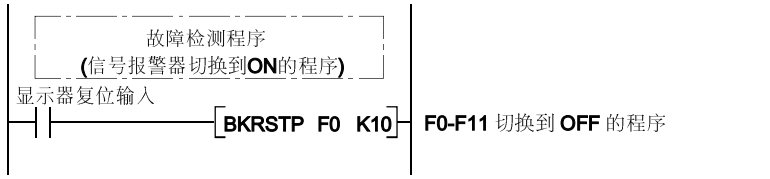
10 - 13

4) OUT F□指令可用同样的指令来执行信号报警器编号的 ON/OFF。
然而，如果用 OUT F□指令将某一信号报警器切换到 OFF，“处理信号报警器 OFF”（下述项目（b））就不再发生。
在信号报警器已由 OUT F□指令切换到 OFF 以后，执行 RST F□，LEDR 或 BKRST 指令。

1) 只切换编号存储在 SD62 和 SD64 中的信号报警器到 OFF



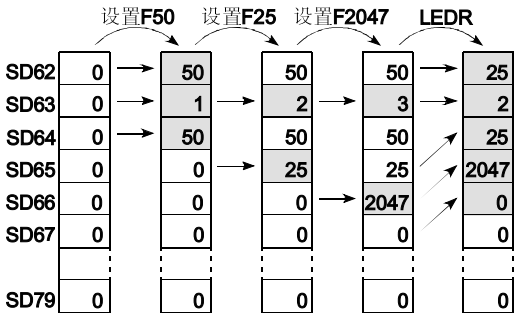
2) 关断所有处于 ON 状态的信号报警器



备注

有关 LEDR 和 BKRST 指令的细节,参见 SX QCPU(Q 模式)/QnA 编程手册(共用指令)

(b) 信号报警器 OFF 时的处理
1) 特殊寄存器(SD62-SD79)在 LEDR 指令下的数据操作。
a) 存储在 SD64 中的信号报警器的编号被删除。存贮在其后的寄存器（SD65-SD79）中的信号报警器编号被移动填补空闲的空间。
b) 存贮在 SD64 中的信号报警器编号存放到 SD62 中。
c)从 SD63 的数值中被减“-1”。
d) 如果 SD63 的值是“0”，SM62 就切换到 OFF。



10 - 14

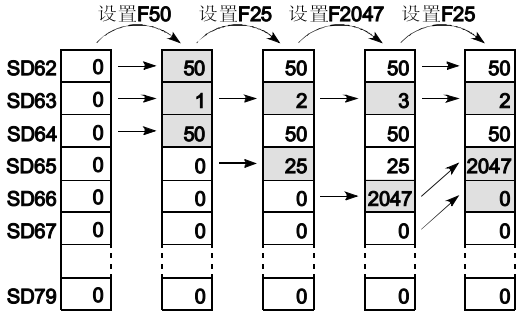
2) 当一个信号报警器由 RSTF□ 指令切换到 OFF 时的特殊寄存器（SD62-SD79）的数据操作。

a) 被切换到 OFF 的信号报警器的编号被删除，所有其后的信号报警器的编号被移动去填补空闲的空间。

b) 如果存贮在 SD64 中的信号报警器的编号被切换到 OFF，存贮在 SD64 中的新的信号报警器编号被存贮到 SD62。

c) 存贮 SD63 中的数值减去“-1”。

d) 如果 SD63 的值为“0”SM62 就被切换到 OFF。



3) QCPU 的处理

如果所有 SD64-SD79 的信号报警器的编号全部被切换到 OFF。QCPU 面板显示器上的“USER”用户 LED 就熄灭。

注意

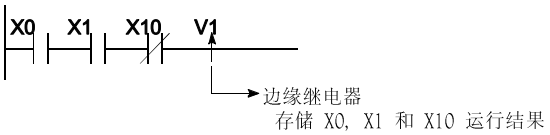
当信号报警器切换到 ON 时，如果出现以比信号报警器更高的优先级继续运行的错误，可以通过执行一条 LEDR 指令来消除出错。其优先级顺序请参见 7.20。在这种情况下，执行一条 LED 指令不能把信号报警器切换到 OFF。要将信号报警器切换到 OFF，你必须在执行 LEDR 指令之前先消除这个错误，因为出错会取比信号报警器为先的优先级。

10 - 15

10.2.6 边沿继电器(V)

(1) 定义：

(a) 一个边沿继电器是用来存储。从梯形图的（程序）块开始的运行结果（ON/OFF 信息）的软元件。



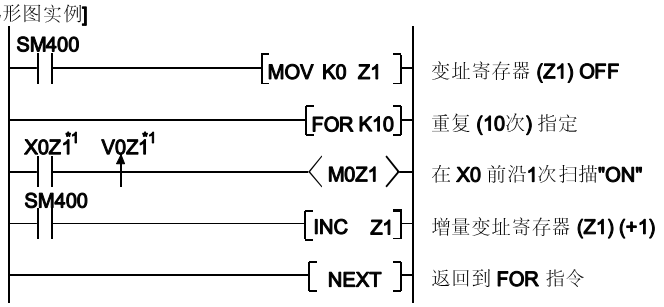
边沿继电器只能用作接触器，而不能作为线圈使用。

(b) 在 QCPU 执行的程序中，相同的边沿继电器编号，不能使用二次。

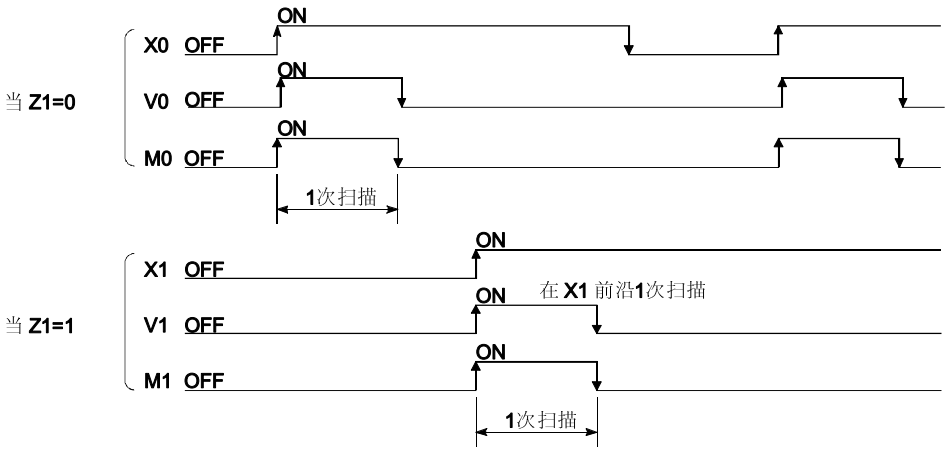
(2) 边沿继电器的应用

边沿继电器是用来在使用变址修饰配置的程序中检测前沿（从 OFF 到 ON）。

[梯形图实例]



[时序表]



备注

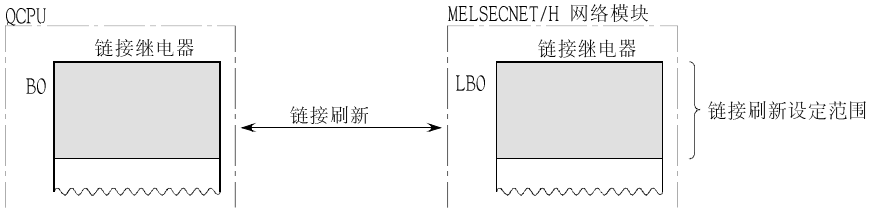
1) *1: X0Z1 的 ON/OFF 信息存储在 V0Z1 边沿继电器中。
例如：X0 的 ON/OFF（开/关）信息存储在 V0，X1 的 ON/OFF（开/关）信息存储在 V1。

10 - 16

10.2.7 通信继电器（B）

(1) 定义

(a) 一个通信继电器是用于用 MELSECNET/H 网络模块的通信继电器（LB）刷新 QCPU 的 QCPU 继电器，和用 QCPU 的数据刷新 MELSECNET/H 网络模块的通信继电器。



内部继电器或锁存继电器可用于 MELSECNET/H 网络系统不使用的数据范围。

- 不发生通信继电器锁存的范围....内部继电器
- 发生通信继电器锁存的范围.....锁存继电器

(b) 程序中使用的接触器(常开接触器,常闭接触器)的数目无限制。

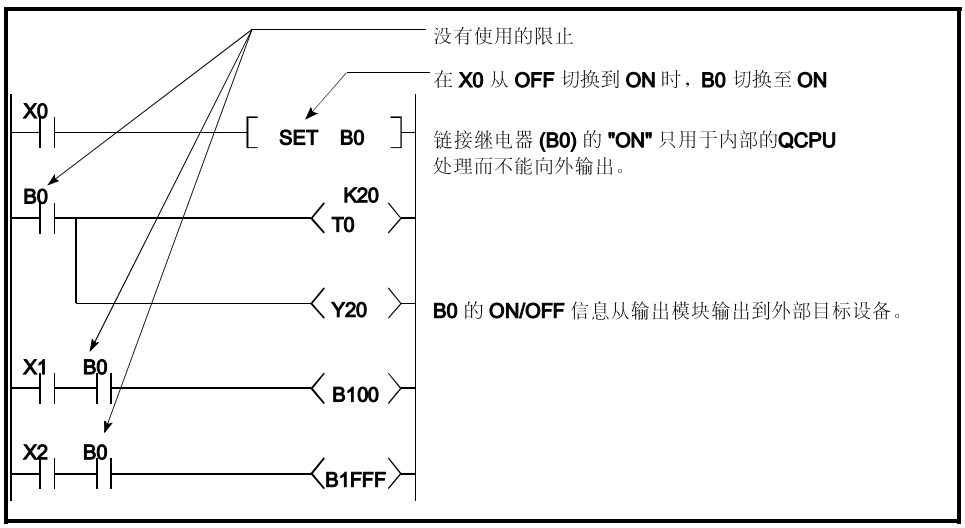


图 10.7 通信继电器

(2) 在网络系统中使用通信继电器

为了在网络系统中使用通信继电器，需要进行网络参数设定。设有指定网络参数设定的通信继电器可以作为内部继电器或锁存继电器使用。

备注

1) 有关网络参数的细节，参考 Qs MELSECNET/H 网络系统参考手册。
2) MELSECNET/H 网络模块分配有 16384 个通信继电器点。QCPU 配有 8192 个通信继电器点，当使用点 8192 以后的点时，用 PLC 参数对话框的软元件设定页来改变通信继电器点的编号

10 - 17

10.2.8 特殊通信继电器（SB）

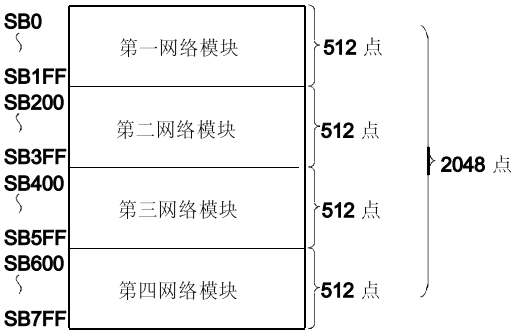
(1) 定义

(a) 特殊通信继电器表示通讯的状态和 MELSECNET/H 网络模块之类的智能功能模块的出错检测。

(b) 因为特殊通信继电器是按照在数据通讯期间可能发生的各种问题来 ON/OFF 的，它们作为验证数据通讯问题的工具来使用。

(2) 特殊通信继电器的点的数目

从 SB0 到 SB7FF 一共有 2048 个特殊通信继电器的点，特殊通信继电器按照每个智能功能模块如 MELSECNET/10H 网络模块 512 点的比率来分配。特殊通信继电器的分配。如下图所示。



备注

1) 用于 QCPU 的特殊通信继电器的细节，参考 QCPU（Q 模式）/QnACPU 编程手册。（公用指令）

10.2.9 步进继电器(S)

步进继电器是一种 SFC 程序软元件。
有关使用步进继电器的顺序的细节，参考 QCPU（Q 模式）/QnACPU 编程手册（SFC）。

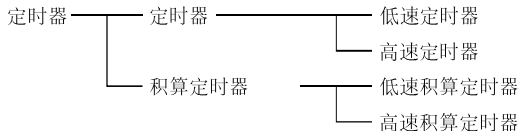
注意

因为步进继电器是专用于 SFC 程序的软元件，它不能在顺控程序中作为内部继电器来使用。
如果用这种方式，将使 SFC 发生出错。系统将停止运行（系统宕机）。

10 - 18

10.2.10 定时器(T)

定时器是向前定时器型的，当线圈 ON 时，时间测量开始。
当当前值超过设定值时，测量结束（时间用完）。
当出现“时间用完”时，当前值与设定值相符。有二种类型的定时器：一种是当定时器线圈 OFF 时，允许当前值归“0”的低/高速定时器，另一种是即使定时器线圈 OFF 时，仍保持当前值的积算定时器。



通过定时器的设定（指令格式），一个软元件被指定用作低速定时器或高速定时器。
OUT TO 指令用来指定一软元件为低速定时器。而 OUTH TO 指令则用于指定一软元件为高速定时器。
通过定时器设定（指令格式）指定一软元件为低速积算定时器或高速积算定时器，
OUT TO 指令用来指定一软元件为低速积算定时器，而 OUTH TO 指令则用来指定一软元件为高速积算定时器。

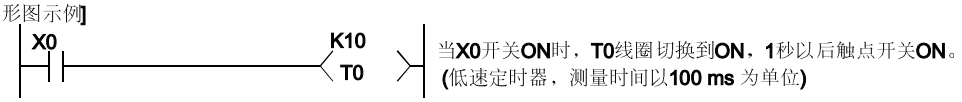
低速定时器

(1) 定义：

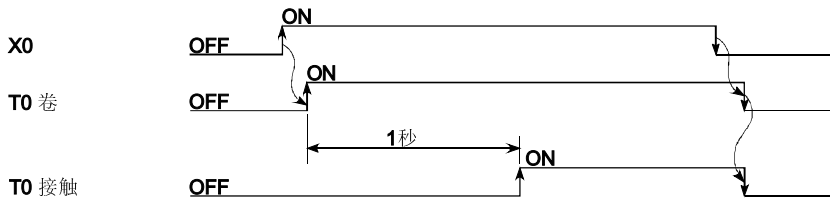
(a) 低速定时器是只有当线圈 ON 时才运行的定时器。

(b) 当定时器线圈 ON 时，计时开始，当“时间用完”出现时接触器切换到 ON。当定时器线圈 OFF 时，当前值归“0”而接触器 OFF。

[梯形图示例]



[时序表]



(2) 测量单位：

(a) 低速定时器的测量时间单位缺省值为 100ms。

(b) 可以指定时间测量单位设定值以 1ms 为单位在 1-1000ms 的范围内。
可以用 PLC 参数设定的“PLC 系统设定”指定这些设定。

10 - 19

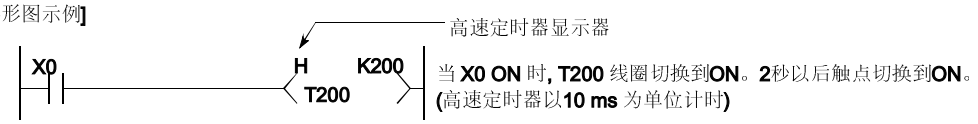
高速定时器

(1) 定义

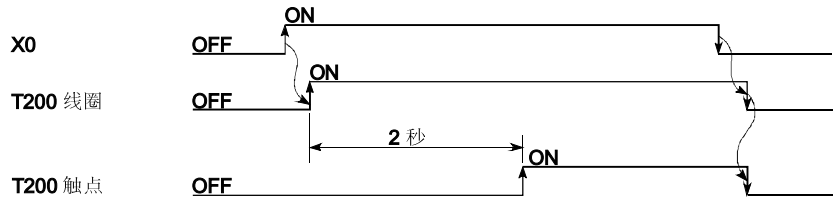
(a) 高速定时器是只有在线圈 ON 时才运行的定时器。高速定时器用符号“H”来表示。

(b) 当定时器线圈 ON 时，计时开始，当时间用完时，接触器切换到 ON。当定时器线圈 OFF 时，当前值成为“0”，接触器切换到 OFF。

[梯形图示例]



[时序表]



(2) 测量单位

(a) 高速定时器的计时单位设定的缺省值为 10ms。

(b) 可以指定计时单位设定为 0.1ms，在 0.1ms-100ms 之间。
用 PLC 参数设定的 PLC 系统设定中可以指定这些设定。

10 - 20

积算定时器 *

(1) 定义

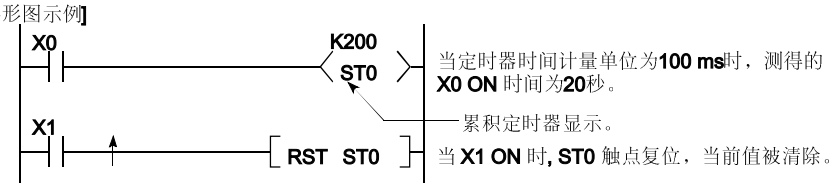
(a) 积算定时器测量“线圈 ON”的时间。

(b) 当定时器线圈 ON 时，计时开始，当“时间用完”发生时，接触器切换到 ON。即使定时器线圈 OFF，当前值和接触器的 ON/OFF 状态亦被保存起来。当线圈再一次切换到 ON 时，从被保存的当前值重新开始计时。

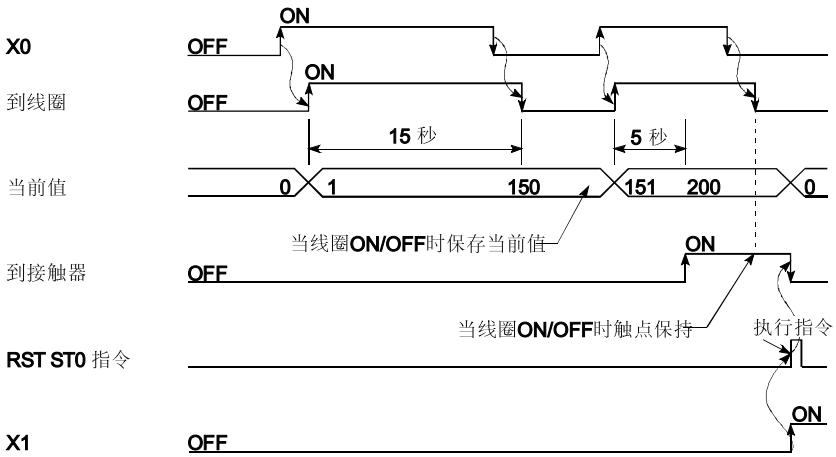
(c) 有二种类型的积算定时器：低速积算定时器和高速积算定时器。

(d) RSTT□指令是用来清除（复位）当前值和把接触器切换到 OFF 的。

[梯形图示例]



[时序表]



(2) 测量单位

(a) 积算定时器计时单位的设定与低速定时器和高速定时器相同。

- 低速积算定时器与低速定时器相同。
- 高速积算定时器与高速定时器相同。

备注

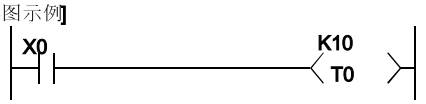
1) *: 为了使用积算定时器，积算定时器“已使用的点数”的设定值必须用 PLC 参数软元件设定来指定。

10 - 21

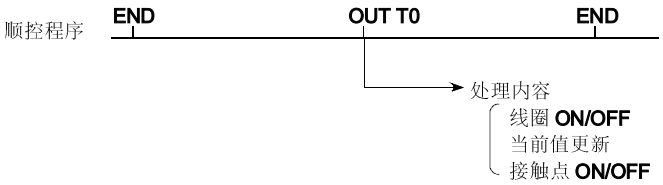
定时器处理及精度

(a) 在执行 OUT T□指令时，发生下列的处理过程：定时器线圈 ON/OFF，当前值更新和接触器 ON/OFF 处理。在作 END（结束）处理时，定时器当前值更新和接接触器 ON/OFF 处理不再发生。

[梯形图示例]

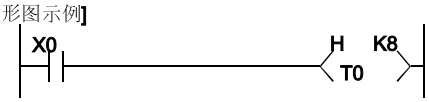


[OUT TO指令的处理]

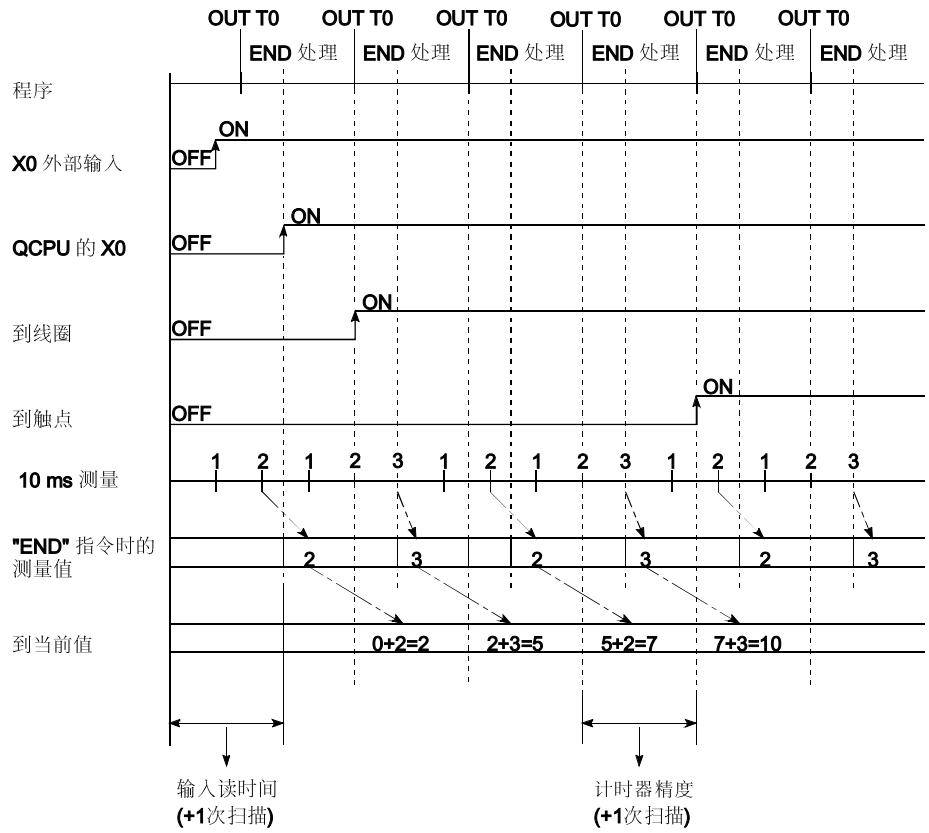


(b) 执行 OUT T□指令时，在 END 指令时所测量的扫描时间加入到当前值中去。
执行 OUT T□指令时，定时器线圈处于 OFF 状态，当前值就不会被更新。

[梯形图示例]



[当前值更新时序]



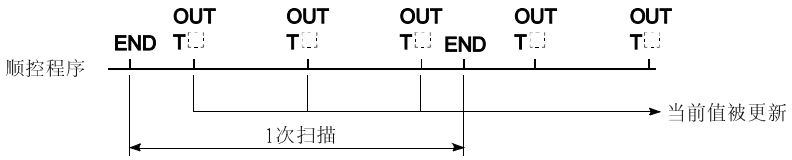
10 - 22

(c) 定时器的响应精度从输入（**X**）读发生的点到输出发生的点是**+2**次扫描。

使用定时器时注意事项

下面列出的是有关使用定时器的注意事项

(a) 在一次扫描中，一个给定的定时器不能被 **OUT T□** 指令指定一次以上。
如果不止一次，每一个 **OUT T□** 指令都将使定时器的当前值更新。结果导致测量无意义。



(b) 当一个定时器（例如：**T1**）线圈 **ON** 时。
使用一个 **CJ** 指令，就不能跳过 **OUT T1** 指令。如果跳过 **OUT T□** 指令，定时器的当前值就不会被更新。

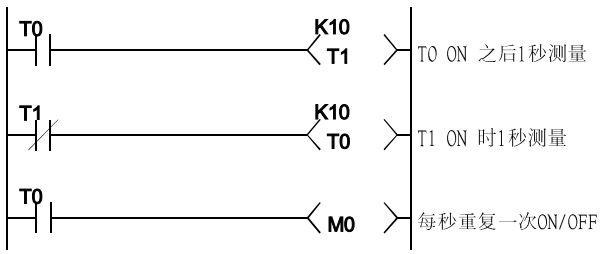
(c) 定时器不能用在中断程序中和固定循环执行的程序中。

(d) 如果定时器设定值为“**0**”，当执行 **OUT T□** 指令时，接触器变为 **ON**。

(e) 如果设定值改变到一个高于当前值的数值，随后定时器“时间用完”，这“时间用完”的状态会保持有效。不会发生定时器运行。

(f) 如果定时器用于一个低速执行型的程序，执行 **OUT T□** 指令时，当前值将被加到低速扫描时间上。
有关低速扫描时间的细节。参见 **4.3.2** 节。

(g) 如使用 **2** 个定时器，**ON/OFF** 梯形图应该如下表所示编制：



10.2.11 计数器(c)

计数器是“增计数器”型的，当计数值等于设定值（计数结束条件）时接触器切换到**ON**。
有二种类型的计数器：在顺控程序中，对输入条件启动（前边沿）计数的计数器，和对中断因素发生的次数计数的计数器。

计数器(c)

(1) 定义

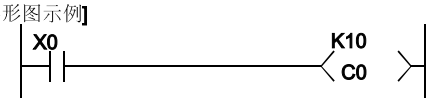
计数器是一种在顺控程序中对输入条件脉冲前边沿数计数的软元件。

(2) 计数处理

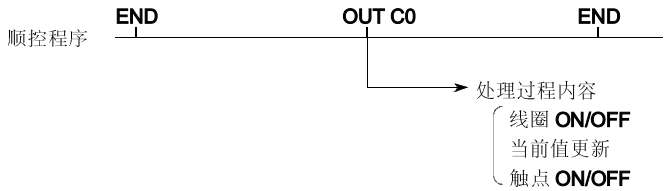
计数器是一种在顺控程序中对输入条件前沿数计数的软元件。

(a) 当执行 **OUT C□** 指令时，发生下列计数处理过程：
线圈 **ON/OFF**，当前值更新（计数值**+1**）和接触器 **ON/OFF**：在“**END**”处理过程，计数器当前值更新和接触器 **ON/OFF** 处理不会发生。

[梯形图示例]

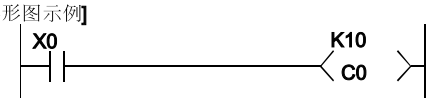


[OUT C0 指令时的处理过程 (X0: OFF→ON)]

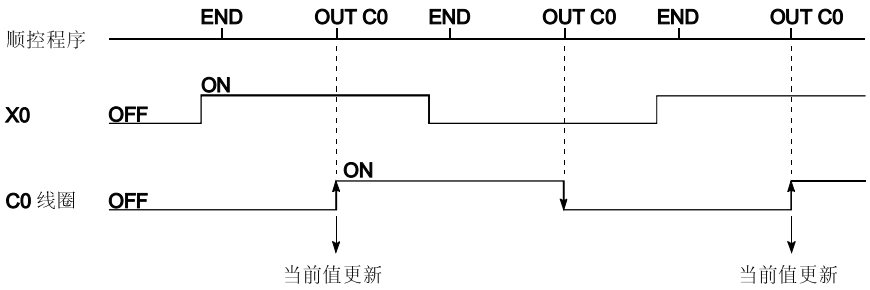


(b) 当前值的更新（计数值**+1**）发生在 **OUT C□** 指令的前沿（**OFF→ON**）。
在下列的 **OUT C□** 指令状态下，当前值不会被更新：**OFF**，**ON** 到 **ON**，**ON** 到 **OFF**。

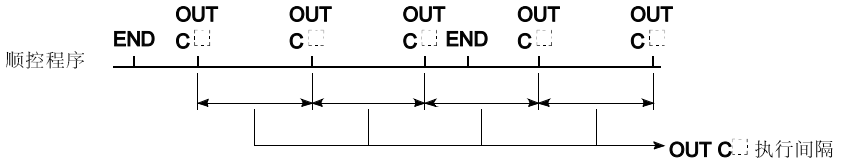
[梯形图示例]



[当前值更新时序]



(c) 可以在一次扫描之内使用多个计数器以达到最大的计数速度。
在这种情况下，应该使用直接访问输入（DX□）方式的号计数器输入信号。*1

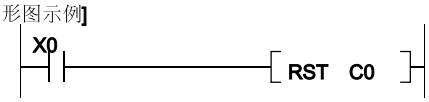


(3) 使计数器复位

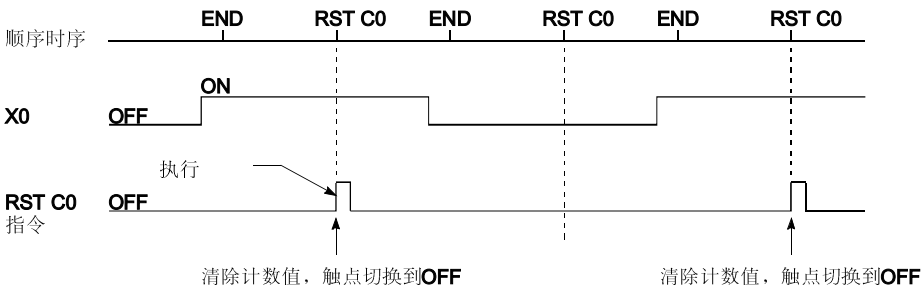
(a) 即使当 OUT C□ 指令切换到 OFF 时，计数器的当前值不会被清除。要用 RST C□ 指令来清除计数器的当前值。并将接触器切换到 OFF。

(b) 在执行 RST C□ 指令时的点上，计数值被清除并且将接触器切换到 OFF。

[梯形图示例]



[计数复位时序]



(4) 最大计数速度

只有当输入条件的 ON/OFF 时间（间隔）比相应的 OUT C□ 指令的执行（时间）间隔长的时候，计数器才计数。
最大的计数速度由下列公式来计算：

□

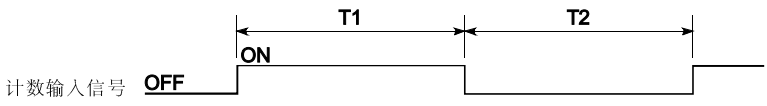
$$\text{最大计数速度(Cmax)} = \frac{n}{100} \times \frac{1}{T} \text{[次/秒]}$$

n: 占空率（%）*2
T: OUT C 指令的执行间隔时间

备注

1) *1: 有关直接访问输入的细节 参见 10.2.1 节。
2) *2: “占空率”是用百分比来表示的计数输入信号的开/关时间比率。

- 当 $T1 \geqq T2$ $n = \frac{T1}{T1+T2} \times 100$ %
- 当 $T1 < T2$ $n = \frac{T2}{T1+T2} \times 100$ %



10 - 25

中断计数器

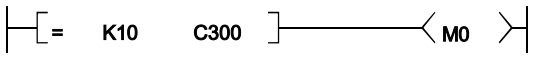
(1) 定义

中断计数器是中断因素发生数的计数软元件。

(2) 计数处理过程

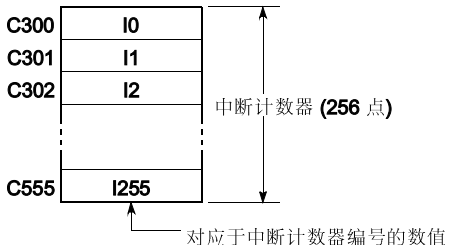
(a) 当中断发生时，中断计数器的当前值被更新。因而不必去创建一个包含中断计数功能的程序。

(b) 中断计数器的操作要求不仅是简单地指定一个设定值。
中断计数器用于控制的用途时，也必须使用比较指令（**=**，**<=**等）来进行与设定值的比较,依照比较结果使内部继电器**(M)**等 **ON/OFF**。
下图是一个采样程序，其中，**M0** 在 **10** 次中断输入发生以后切换到 **ON**。（在本实例中，“**C300**”是对应于 **10** 的中断计数器的编号。



(3) 设定中断计数器

(a) 为了使用中断计数器，首先必须用 **PLC** 参数设定在 **PLC** 系统设定中指定中断计数器编号的设定。
256 点被分配作中断计数器。从这些指定的“第一个计数器编号”开始。
如果 **C300** 被指定为第一个中断计数器的编号，编号 **C300** 到 **C555** 将分配给中断计数器。



(b) 为了使用中断计数器，必须通过主程序的 **E1** 指令来建立一个“允许中断”的状态。

(4) 注意事项

(a) 执行中断计数器和中断程序操作时，一个中断指针是不够的。
而且在用 **PLC** 参数设定的 **PLC** 系统设定中只指定一个中断计数器设定，是不能执行一个中断程序的。

(b) 如果当中断发生时，正在进行下列项目的处理，计数的操作将被延迟到完成对这些项目的处理。
在完成程序的执行以后，计数的处理过程才开始。
即使在处理这些项目的过程中再次发生同样的中断，只有一次中断被计数到。
- 在顺控顺控指令的执行期间
- 在中断程序的执行期间
- 在固定扫描执行型的程序的执行期间

(c) 中断定时器的最大计数速度是由下列项目的最长处理时间来决定的。
- 在程序所使用的指令中，处理时间最长的指令。
- 中断程序处理时间
- 固定扫描执行型的程序的处理时间

(d) 太多中断计数器的使用将增加顺控程序的处理时间并可能引起“**WDT ERROR**”出错。
在这种情况发生时，要减少中断计数器的数目或减少对输入脉冲讯号的计数速度。

(e) 在顺控程序中，在 **FEND** 指令之前，使用 **RSTC** 指令可以使中断计数器的计数值复位。

(f) 使用顺控程序的 **MOV** 指令可以读出中断计数器的计数值。

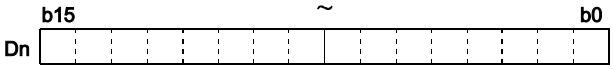
10 - 27

10.2.12 数据寄存器（D）

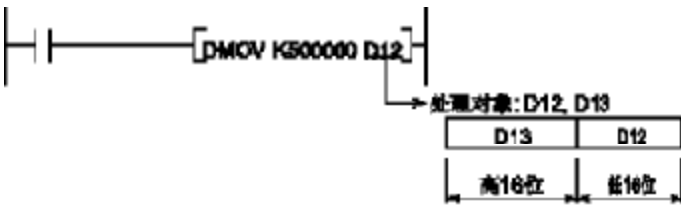
(1) 定义

(a) 数据寄存器是在 QCPU 中存储数字数据（-32768-32767 或 0000H 到 FFFFH）的存储软元件。

(b) 数据寄存器每个点由 16 位组成，以 16 位为单位执行读和写的操作。



(c) 如果数据寄存器用于 32 位指令，数据将被存储在寄存器 Dn 和 Dn+1 中，低 16 位的数据存储在顺控程序中指定编号（D0）的数据寄存器中，而高 16 位的数据则存储在指定编号+1（Dn+1）的寄存器中。举例：如果在 DMOV 指令中指定寄存器 D12，则低 16 位部份存储在 D12。而高 16 位的部分则存储在 D13。



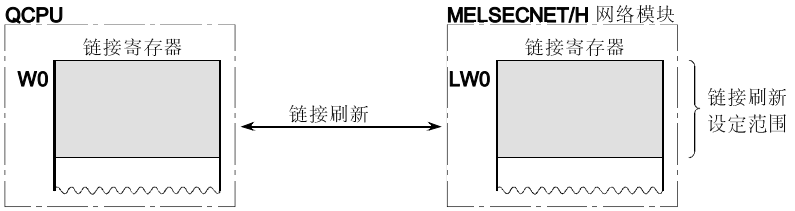
二个数据寄存器可以存贮一定范围的数字数据，从-2147483648 到 2147483647 或从 0H 到 FFFFFFFFH。

(d) 由顺控程序存储的数据一直保持到发生存储另一个数据的操作时。

10.2.13 通信寄存器 (W)

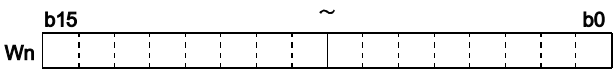
(1) 定义

(a) 通信寄存器是用于用从包含 MELSECNET/H 网络模块的智能功能模块的通信寄存器（LW）的数据来刷新 QCPU。
通信寄存器是用来将数字数据（-32768 到 32767，或从 OOOOH 到 FFFFH）储存到 QCPU 的。

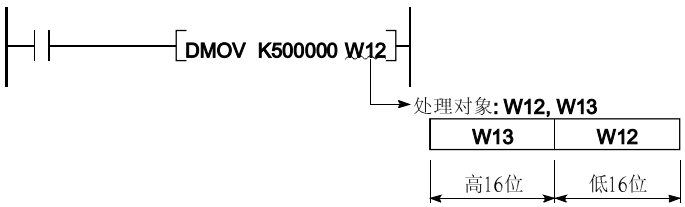


在 MELSECNET/10H 网络系统范围以外使用时，通信寄存器可以用作数据寄存器。

(b) 通信寄存器每个点由 16 位组成，在 16 位模块中执行读和写的操作。



(c) 如果通信寄存器是用于 32 位指令，数据存储在寄存器 Wn 和 Wn+1 中，低 16 位数据存储在顺控程序中指定编号为 Wn 的通信寄存器中，而高 16 位数据则存储在指定的编号为（Wn+1）的寄存器中。
例如：在 DMOV 指令中指定 W12 通信寄存器，则低 16 位存储在 W12，而高 16 位存储在 W13。



在二个通信寄存器点中，可存储，-2147483648 到 2147483647 或 OH 到 FFFFFFFFH 个数据。

(d) 由顺控程序存入的数据将保持到发生另一次保存数据操作。

备注

MELSECNET/H 网络模块有 16384 个通信寄存器点。QCPU 有 8192 通信寄存器点。当在点 8192 以后的顺序点被用作通信寄存器时，要改变在 PLC 参数对话框的软元件设置表中的通信寄存器“点的数量”的设定。

10 - 29

(2) 在网络系统中使用通信寄存器

为了在网络系统中使用通信寄存器，必须进行网络参数设定。
没有在网络参数设定中设置的通信寄存器可用作数据寄存器。

备注

1) 有关网络参数，详见：QS MELSECNET/H 网络系统参考手册。

10.2.14 特殊通信寄存器 (SW)

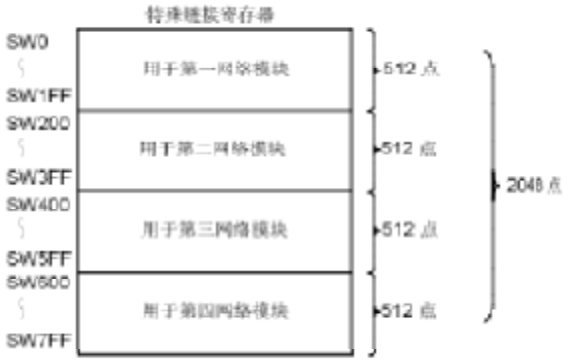
(1) 定义

(a) 特殊通信寄存器是用来储存关于通讯状态和智能功能出错的数据的。

(b) 因为数据通讯讯息是作为数字数据来存储的。特殊通信寄存器是作为确认位置和故障原因的工具来使用的。

(2) 特殊通信寄存器点的数目

从 SW0 到 SW7FF，共有 2048 个特殊通信寄存器的点。特殊通信寄存器点以每个智能功能模块 512 个点的比率分配（如 MELSECNET/H 网络模块）。
根据缺省值，如下所示：下列点被分配给通信寄存器：



备注

有关 QCPU 中使用的特殊通信寄存器，详见 QCPU（Q 模式）/QnACPU 编程手册（公共指令）。

10 - 30

10.3 内部系统软元件

内部系统软元件是用于内部系统运行的软元件。
内部系统软元件的分配和大小是固定的，用户不能改动。

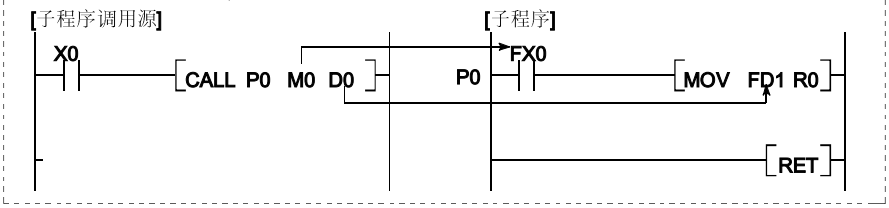
10.3.1 功能软元件(FX, FY, FD)

(1) 定义

(a) 功能软元件是用于带变量的子程序的软元件，允许在带变量的子程序和该子程序的调用源之间传送数据。

示例

如果 FX0 和 FD1 用于子程序，而子程序中的CALL指令指定 M0 和 D0，M0的 ON/OFF 数据传送到 FX0, D0 的数据传送到 FD1。



(b) 因为可以设置用于每一个子程序调用源的功能软元件。因而可以使用相同的子程序而不必考虑其它的子程序调用源。

(2) 功能软元件的类型

有三种类型的功能软元件：功能输入软元件（FX），功能输出软元件（FY）和功能寄存软元件（FD）。

(a) 功能输入软元件 (FX)

- 这些软元件用来指定输入到子程序的 ON/OFF 数据。
- 在子程序中，这些软元件用于读出和处理由带变量的子程序的调用指令指定的位数据。
- 可以使用所有的 QCPU 位数据指定软元件。

(b) 功能输出软元件 (FY)

- 这些软元件是用来指定子程序运行结果（ON/OFF 数据）向子程序程序调用源的输出的。
- 带变量的子程序的运行结果存储在指定的软元件中。
- 可以使用除了 QCPU 输入（X，DX）以外的所有位数据指定软元件。

10 - 31

(c) 功能寄存器

- 功能寄存器是用来指定在子程序调用源和子程序之间传送的数据的。
- QCPU 自动地决定功能寄存器输入/输出的条件。如果子程序数据就是源数据，就指定数据为子程序输入数据。
  如果子程序数据是目标数据，就指定数据为子程序输出数据。
- 1 点占有 4 字
  所使用的字数取决于子程序中的指令。

一条一个字的指令需要一个字。



数据存储在一个点(D0)。

一条二工作指令需要二个字。



数据存储在二个点 (D0 和 D1).

32 位乘/除操作的目标址需要四个字。



数据存储在四个点 (D0 到 D3).

- 在含有变量的子程序中不能使用使用功能寄存器的软元件。如果使用分配给功能寄存器的软元件，功能寄存器的值就不能正常地返回到调用程序。



因为点 (D0-D3) 已用于FD0，D3就不能用于子程序。

- 可以使用 QCPU 的字数据软元件。

备注

1) 关于使用功能寄存器的步骤，参见 QCPU（Q 模式）/QnACPU 编程手册（公用指令）。

10.3.2 特殊继电器 (SM)

(1) 定义

(a) 特殊继电器是用来存储 QCPU 状态数据的。

(2) 特殊继电器分类

特殊继电器是按照它们的应用来分类的，如下所示：

(a) 用于故障诊断	: SM0 - SM199
(b) 系统信息	: SM200 - SM399
(c) 系统时钟/系统计数器	: SM400 - SM499
(d) 扫描信息	: SM500 - SM599
(e) 存储卡信息	: SM600 - SM699
(f) 相关指令	: SM700 - SM799
(g) 用于调试	: SM800 - SM899
(h) 锁存区	: SM900 - SM999
(i) 用于 A-PLC	: SM1000 - SM1299*

备注

1) 有关 QCPU 能使用的特殊继电器，详见附录 1。

2) * 当你打开了 PLC 参数对话框的 PLC 系统表中“与 A-PLC 兼容”部分中的“从 SM1000/SD1000 开始使用特殊继电器/特殊寄存器”复选框后才生效。

10 - 33

10.3.3 特殊寄存器 (SD)

(1) 定义

(a) 特殊寄存器是用于存储 QCPU 状态数据（诊断和系统信息）的。

(2) 特殊寄存器分类

特殊寄存器按照它们的应用来分类，如下所示

(a) 用于故障诊断	: SD0 - SD199
(b) 系统信息	: SD200 - SD399
(c) 系统时钟/系统计数器	: SD400 - SD499
(d) 扫描信息	: SD500 - SD599
(e) 存储卡信息	: SD600 - SD699
(f) 相关指令	: SD700 - SD799
(g) 用于调试	: SD800 - SD899
(h) 锁存区	: SD900 - SD999
(i) 用于 A-PLC	: SD1000 - SD1299*
(j) 熔丝烧断模块	: SD1300 to SD1399
(k) 检查输入/输出模块	: SD1400 to SD1499

备注

1) 有关 QCPU 能使用的特殊寄存器的细节参考附录 2。
2) *: 当你打开了 PLC 参数对话框的 PLC 系统表中“与 A-PLC 兼容”部分中的“从 SM1000/SD1000 开始使用特殊继电器/特殊寄存器”复选框后才生效。

10 - 34

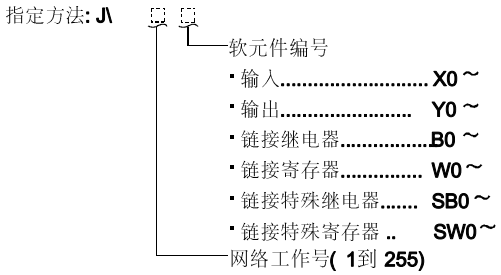
10.4 链接直接软元件 (J□\□)

(1) 定义

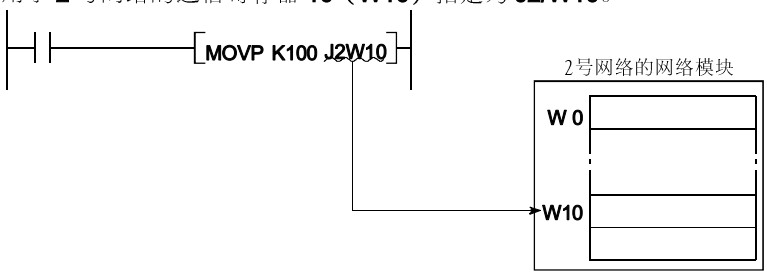
(a) 在 **END** 处理过程中，在 **QCPU** 和 **MELSECNET/H** 网络系统模块之间发生数据刷新（数据传送）操作。链接直接软元件这时用来直接访问在 **MELSECNET/H** 网络模块中的通信软元件。

(b) 指定方法

• 由网络编号和软元件编号来指定链接直接软元件。



• 用于 **2** 号网络的通信寄存器 **10**（**W10**）指定为 **J2/W10**。



• 对于位软元件（**X**，**Y**，**B**，**SB**）必须指定数字。
指定示例：**J1\K1X0**，**J10\K4B0**

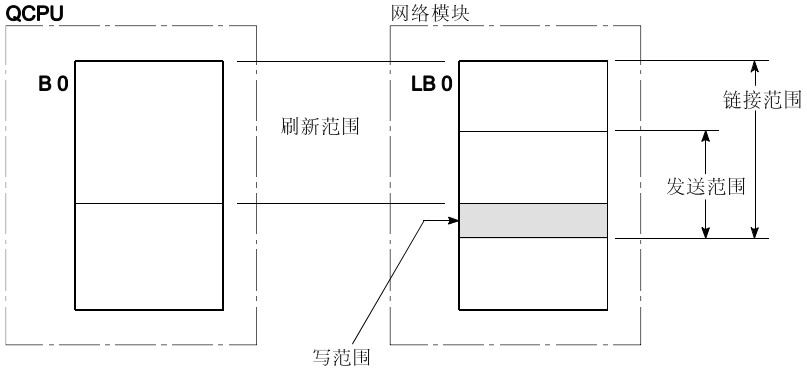
(2) 指定范围

可以指定链接直接软元件用于所有的网络模块中的通信软元件。

〔 也可指定在网络刷新参数规定范围以外的软元件。 〕

(a) 写

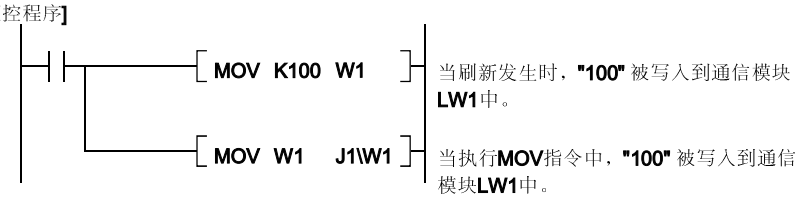
1) 写操作是在用网络参数的公用参数设置为发送范围的通信软元件范围部分之内进行的。这部分是在指定的网络刷新参数的“刷新范围”以外的。



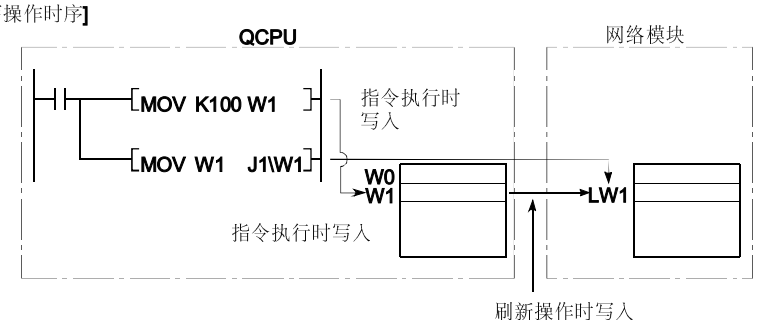
2) 虽然，在通信软元件范围(由刷新参数指定)中的刷新范围部分,也可以写入，当刷新操作发生时, 通信模块的通信软元件数据将被重写。
因而，当用链接直接软元件写入时，同样的数据也应当被写入到由刷新参数指定的 QCPU 相关软元件中去。

[刷新参数设定]
·网络编号 : 1
·QCPU(W0 到 W3F)←→网络模块 (LW0 到 LW3F)

[顺控程序]



[写操作时序]

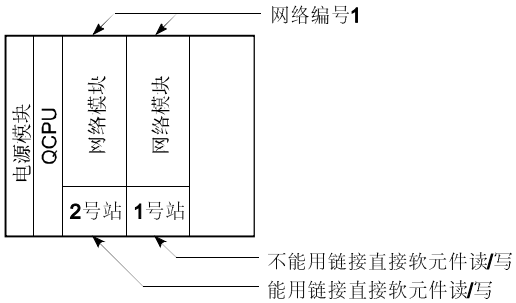


3) 当使用链接直接软元件把数据写入到另一个站的写范围时，从该站接收到的数据将代替写入的数据。

(b) 读
在网络模块的整个通信软元件范围，可以用链接直接软元件来读数据。

注意

(1) 对每一个网络编号，只有一个读/写链接直接软元件的网络模块可以使用。
如果安装有二个或更多的模块在同一网络编号上，则有最低的起始 I/O 地址的网络模块将是使用链接直接软元件执行读/写操作的模块。
举例说：编号为 1 的网络装有 1 号和 2 号网络模块，如下表所示，那么 2 号站网络模块将执行链接直接软元件的运行。



(3) 链接直接软元件与链接刷新之间的区别。

链接直接软元件与链接刷新之间的区别如下表 **10.4** 所示。

表 **10.4** 链接直接软元件与链接刷新之间的区别

项目		链接直接软元件	链接刷新
程序表示法	通信继电器	**J□\K4B0** 或以后	**B0** 或以后
	通信寄存器	**J□\W0** 或以后	**W0** 或以后
	链接特殊继电器	**J□\K4SB0** 或以后	**SB0** 或以后
	链接特殊寄存器	**J□\SW0** 或以后	**SW0** 或以后
步数		二步	一步
网络模块访问范围		所有网络模块通信软元件	刷新参数指定的范围
访问数据保证范围		字单位**(16** 位**)**	

REMARK

1) 有关 **MELSECNET/H** 网络系统，详见 **Q MELSECNET/H** 网络系统参考手册。

2) 有关网络参数，公用参数，网络刷新参数参考下列手册。

- 详细信息 ： **Q MELSECNET/H** 网络系统参考手册。
- 设置步骤 ： **GX DEVELOPER** 操作手册，视窗版。

10 - 37

10.5 智能功能模块软元件 (U□\G□)

(1) 定义

(a) 智能功能模块软元件允许 QCPU 直接访问安装在主基板和扩展基板上的智能功能模块/特殊功能模块的缓冲存储器。

(b) 智能功能模块软元件用智能/特殊功能模块输入/输出地址和缓冲存储器地址来指定的。

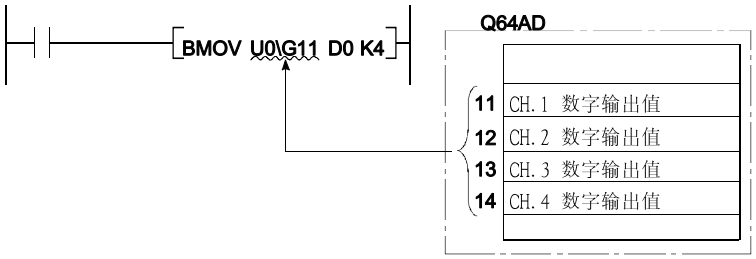
指定方法: U□\G□

- □ → 缓冲存储器地址(设定范围: 0-16383 (十进制)) 1*
- □ → 智能功能模块/特殊功能模块I/O地址
  - 设定: 如果输入/输出地址是一个3位数值，指定前2位数字

    For X/YF0.....X/Y1F0

    指定 "1F"
  - 设定范围: 00 H到 FE H

当安装在主基板的 0 槽的 Q64AD 型模数转换模块(X\Y0 到 X\YF)的通道(CH.1 到 CH.4)的数字输出值存储在 D0 到 D3，输入/输出地址和缓冲存储器的地址，指定如下:



(2) 处理速度

智能功能模块软元件处理速度是:

(a) 从智能功能模块/特殊功能模块的缓冲存储器读入和读出数据的处理速度与在“MOV U0/G11 D0”情况下执行 FROM/TO 指令时相同。

(b) 从智能功能模块/特殊功能模块的缓冲存储中读出数据和执行其它程序的处理速度是执行“FROM/TO 指令的处理速度”与在“+U0\G11 D0 D10”情况下的指令处理速度之和。

如果同样的智能功能模块/特殊功能模块的同样的缓冲存储器在顺控程序中被使用了两次或两次以上，则用 FROM/TO 指令把缓冲存储的数据读到 QCPU 软元件中去的处理速度会上升。

备注

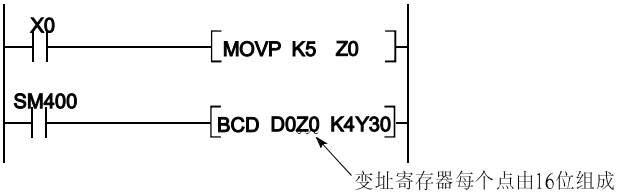
1) *1: 有关缓冲存储器的地址和应用，详见相应的智能功能模块/特殊功能模块手册。

10 - 38

10.6 变址寄存器 (Z)

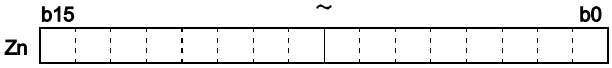
(1) 定义

(a) 变址寄存器是在顺控程序中用于指定间接设定（变址限制）的。
变址寄存器的点是用于变址修改的。

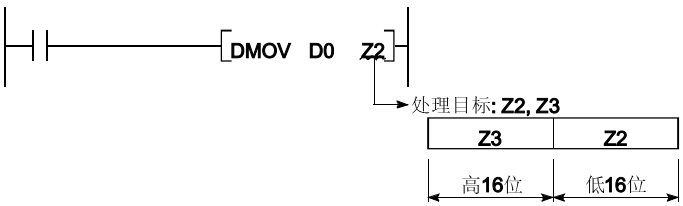


(b) 有 16 个变址寄存器(Z0-Z15)

(c) 变址寄存器每个点由 16 位组成，读和写发生在 16 位的模块中。



(d) 如果变址寄存器用于 32 位指令，数据就存储在寄存器 Zn 和 Zn+1 中。数据的低 16 位存储在变址寄存器地址（Zn），而数据的高 16 位则存储在指定的地址为（Zn+1）的变址寄存器中。举例：在 DMOV 指令中，指定寄存器 Z2，那末低 16 位存放在 Z2，而高 16 位存放在 Z3。



10 - 39

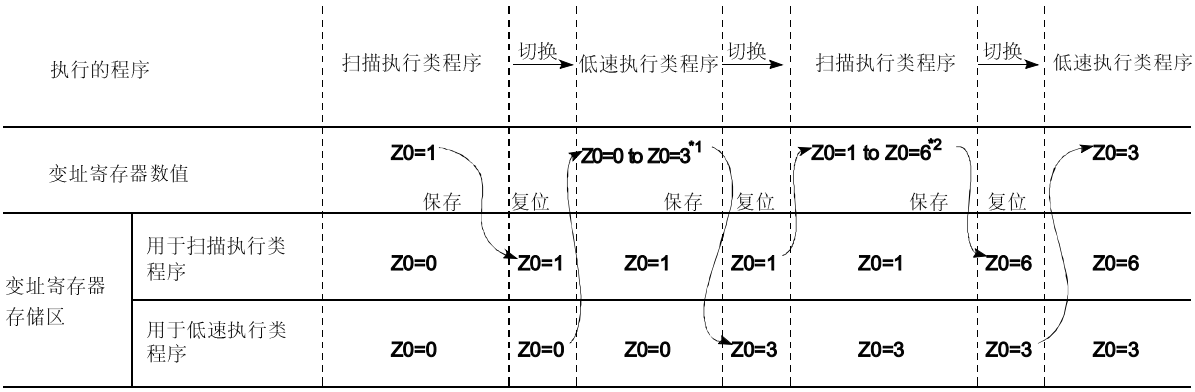
10.6.1 在扫描执行型程序和低速执行型程序之间的切换。

当从扫描执行型程序或低速执行型程序切换到其它型程序时，变址寄存器（Z0 至 Z15）的数据会被保存（保护）起来并复位。

(1) 在扫描执行型程序和低速执行型程序之间切换时的变址寄存器处理。

(a) 当从扫描执行型程序切换到低速执行型程序发生时，扫描执行型程序的变址寄存器数据被保存起来，而低速执行型程序的变址寄存器数据被复位。

(b) 当从低速执行型程序切换到扫描执行型程序发生时，低速执行型程序的变址寄存器数据被保存起来，而扫描执行型程序的变址寄存器数据被复位。



*1: 对于低速执行类程序，将Z0转为3。
*2: 对于扫描执行类程序，将Z0转为6。

(2) 变址寄存器数据的交换

在扫描执行类程序和低速执行类程序之间进行变址寄存器数据交换，应该使用字软元件。

备注

1) 有关使用变址寄存器的变址限制的细节，参考 QCPU（Q 模式）/QnACPU 编程手册（公用指令）。

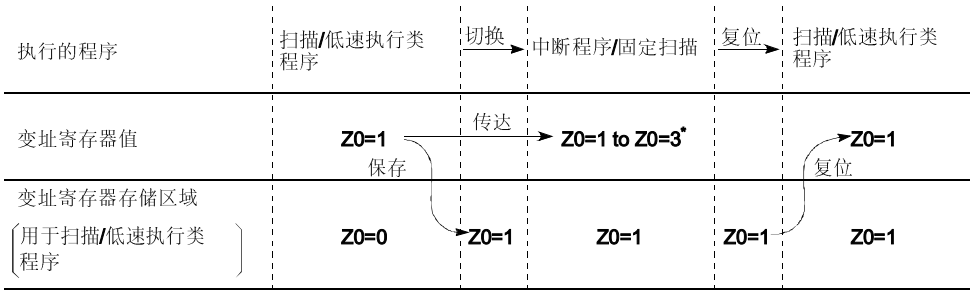
10.6.2 在扫描/低速执行类程序和中断/固定扫描执行类程序之间的切换。

PLC 参数对话框的 PLC 系统表提供了当在扫描执行型程序和低速执行型程序之间以及在中断程序和固定扫描执行型程序之间进行切换时，选择保存（保护）或恢复变址寄存器数据。（Z0-Z15）。
如果：在使用中断程序/固定扫描执行型程序时，你不想写入数据到变址寄存器，只要在 PLC 参数对话框的 PLC 的系统表的“中断程序/固定扫描执行型程序”部分打开“高速执行”复选框，这将使你能在程序之间进行快速切换。

(1) 当“高速执行”复选框处于关闭状态时

(a) 当扫描/低速执行类程序被切换到中断/固定循环执行类程序时，扫描/低速执行类程序的变址寄存器数值首先被保存起来，然后现传送到中断/固定扫描执行类程序。

(b) 当中断/固定扫描执行类程序切换到扫描/低速执行类程序时，被保存的变址寄存器数值复位。



*: 对于中断程序, Z0 变为 3。

将变址寄存器数据从一个中断或固定扫描执行类程序传送到一个扫描或低速执行类程序时应使用字软元件。

10 - 41

(2) 打开“高速执行”复选框时

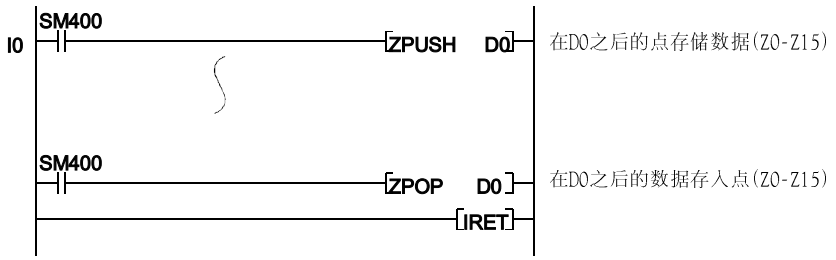
(a) 如果扫描执行型程序/低速执行型程序切换到中断程序/固定扫描执行型程序，变址寄存器数据不会被保存/恢复。

(b) 如果使用一个中断程序/固定扫描执行型程序来写入数据到变址寄存器，用于一个扫描/低速执行型程序的变址寄存器的数值将是不可靠的。

执行的程序	扫描/低速执行类程序	切换	中断程序/固定扫描执行型程序	复位	扫描/低速执行类程序
变址寄存器值	Z0=1	转换	Z0=1 to Z0=3*	传送	Z0=3
变址寄存器存储区域（用于扫描/低速执行类程序）	Z0=0	Z0=0	Z0=0	Z0=0	Z0=0

*: 对于中断程序, Z0 变为 3。

(c) 在使用一个中断程序/固定扫描执行程序写入数据到变址寄存器之前，使用 ZPUSH/SPOP 指令来保存/恢复数据。



10.7 文件寄存器 (R)

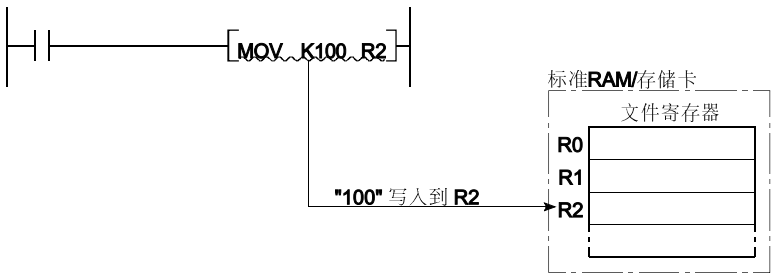
(1) 定义

(a) 文件寄存器是数据寄存器的扩展软元件。

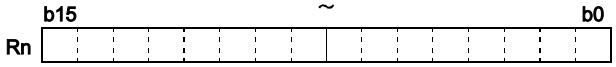
(b) 文件寄存器数据存储在 QCPU 标准 RAM（即 QCPU 存储卡）的中的文件中。

1) 标准 RAM 有 32K 点分配给文件寄存器，可以用与数据寄存器相同的处理速度来使用文件寄存器。

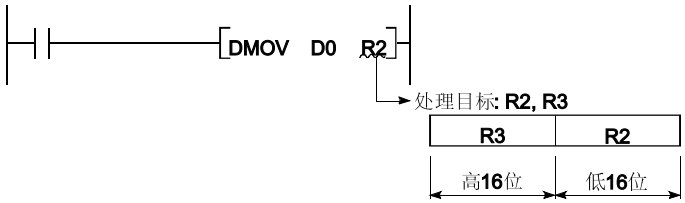
2) 如果 32K 或 32K 以上的点分配给文件寄存器，则可使用存储卡。



(c) 文件寄存器每个点由 16 位组成，在 16 位模块中进行读和写的操作。



(d) 如果文件寄存器用于 32 位指令，则数据将存入在寄存器 Rn 和 Rn+1 中。
数据的低 16 位存入到指定的编号为 Rn 的文件寄存器，而数据的高 16 位存入指定的编号+1 的文件寄存器。
例如：如果在 DMOV 指令中，指定文件寄存器为 R2，则低 16 位存入 R2，而高的 16 位存入 R3。



二个文件寄存器可用来存储数字数据，从-214748368 到 2147483647 或 OH 到 FFFFFFFF$_H$。

(e) 即使电源接通或复位以后，存储在文件寄存器中的数据将会保留。不能用锁存清除来使文件寄存器初始化。当电源关断或复位时，如果你想初始化文件寄存器，可使用顺控程序。

10 - 43

10.7.1 文件寄存器的容量

(1) 使用标准 RAM

最多 32K 点的文件寄存器可存储在标准 RAM 中，标准 RAM 保持有文件寄存器和本地软元件。当不使用本地软元件时，所有的 32K 点可分配给文件寄存器。

(2) 使用 SRAM 卡

文件的大小可以以每块 32K 的比率扩展到 16 块 505K 字。可扩展的块的数目由存储在存储卡上的程序或软元件注解的容量来决定。

(3) 使用闪存卡

文件的大小可以以每存储块 32K 字节的比率来扩展到 32 块 1018K 字。可扩展的块数由存放在存储卡上的程序或软元件注解的容量来决定。

备注

1) 有关 QCPU 存储卡的细节，参见 6.1 节。

10.7.2 不同的存储卡类型决定访问存储卡方法的差别

文件寄存器存储在三种类型的存储器中：标准 RAM，SRAM 卡和闪存卡。
注意：不同的存储器类型决定了不同的访问文件寄存器方法。

怎样访问		标准 RAM	SRAM 卡	闪存卡
由用户程序读		○	○	○
由用户程序写		○	○	×
由软元件设定 PLC 读		○	○	○
由软元件设定 PLC 写		○	○	×
如何修改存储数据	GX DEVELOPER 在线测试操作	○	○	×
	用 GX DEVELOPER 写入 PLC	○	○	×
	用 GX DEVELOPER（闪存 ROM）写入 PLC	×	×	○
	从串行通讯模块批量写入	○	○	×
	从 GOT9000 系列写入软元件	○	○	×
	从 GOT9000 系列随机写入命令	○	○	×

10 - 44

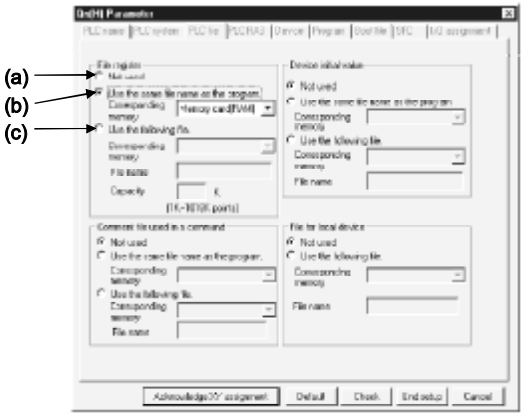
10.7.3 注册文件寄存器

使用文件寄存器，要按照下列步骤用 **QCPU** 注册文件寄存器：



(1) 指定使用的文件寄存器

在顺控程序中所使用的标准 **RAM** 或存储卡文件寄存器是用 **PLC** 参数设置的 **PLC** 文件设定所决定的。



(a) 选择“不使用”时

在下列情况下应选择这个设定。

- 不使用文件寄存器时
- 指定在顺控程序中所使用的文件寄存器时 **QDRSET** 指令是用来指定使用哪一个文件寄存器。

(b) 选择“使用与程序相同的文件名称”时

1) 当使用与顺控程序有相同文件名称的文件寄存器时。

2) 如果程序改变，文件寄存器自动改变以适应新的程序名称。

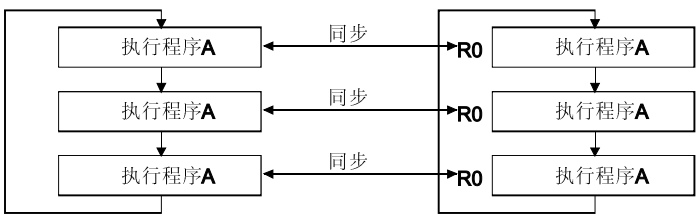
在许多情况下，使用文件寄存器作为只与当前所执行的程序一起使用的本地软元件是十分方便的。

3) 通过在线写入 **PLC**，可以设置所使用的文件寄存器的编号。

举例

当使用与 **(A-C)** 具有相同名称的文件寄存器 **(A-C)** 时,操作如下图所示：

- 执行程序**A**时 — 访问文件寄存器**A**
- 执行程序**B**时 — 访问文件寄存器**B**
- 执行程序**C**时 — 访问文件寄存器**C**



10 - 46

(c) 选择“使用以下文件”时

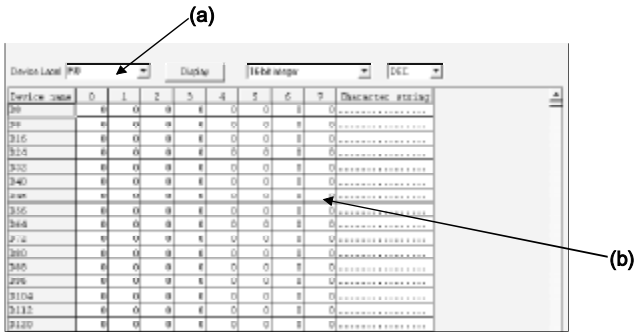
1) 当给定的文件寄存器是被所执行的所有程序共享时，应选择这个设定。

2) 在“相对应的存储器”、“文件名称”和“容量”的文本框中指定所希望的参数，QCPU 就会用指定的参数，创立一个“文件寄存器”文件。如果在“容量”文本框中没有指定参数，可能会产生下列结果。

- 如果具有指定文件名的文件寄存器文件存储在指定的驱动器中，可使用这个文件寄存器文件。
- 在指定的驱动器上，如果没有找到有指定文件名称的文件寄存器文件，就会发生参数错误（3002）。
- 当使用 ATA 卡时，存储器卡（ROM）不能用目标存储器注册。如果用目标存储器注册存储器卡，当文件寄存文件被写到 QCPU 上时，将发生参数错误（3000）。

(2) 文件寄存器设定

使用软元件存储器屏幕来指定一个文件寄存器文件的文件名称。



(a) 设定文件寄存器

在表的框内键入 Rn 来观察文件寄存器的列表。

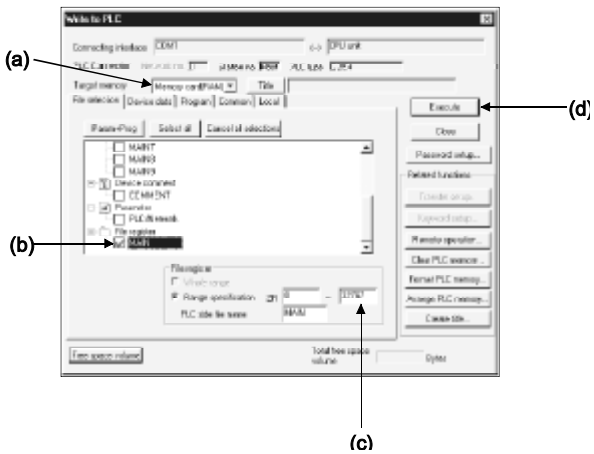
(b) 设定参数

在竖列中输入希望的数据来指定文件寄存器。当你仅指定文件寄存器的容量时，就不要做这一步。

(3) 用 QCPU 注册文件寄存器文件

如果你在 PLC 参数对话框的 PLC 文件表中选择下列的复选框，你必须注册一个 QCPU 文件寄存器文件。

- 不用
- 使用与程序相同的文件名



要注册一个文件寄存器文件，使用“写入到 PLC”对话框。

(a) 选择一个存储器来存储文件寄存器。

选择，标准 RAM，存储卡（RAM），或存储卡（ROM），可从这个表的复选框中指定一个存储器来存储文件寄存器。如果你想要使用与程序名称相同的文件名，在 PLC 参数对话框的 PLC 文件表中指定一个存储器来存储文件寄存器文件。

(b) 选择文件寄存器文件

如果一个存储器被选作文件寄存器，文件寄存器文件的文件名就显示出来，选择所希望的文件寄存器文件的名称。

(c) 指定容量和文件名称

该部分是用来指定写入到 QCPU（QCPU 一边名称）中去的文件寄存器文件的文件名和文件寄存器的容量。

1) 文件寄存器的容量可以被指定从以 256 点为单位的 ZR0 开始。如果文件寄存器不能从 ZR0 开始分配，这将会导致产生一个含有点数从 ZR0 到最后一个点的文件寄存器文件。举例说：如果指定的文件寄存器的存储范围从 ZR1000 到 ZR1791，文件寄存器文件将含有从 ZR0 到 ZR1791 的点数。指定文件寄存从 ZR0 开始是因为未定义的数据是从 ZR0 到 ZR999 的。对文件寄存器容量的检查是以 1K 点为单位的。文件寄存器的容量应该被指定以 1K 点为单位从 R0 开始。

2) 该部分是用来指定一个与所指定的文件寄存器文件名称不同的文件名称。

(d) 把文件寄存器文件存储到 QCPU 的存储器中。

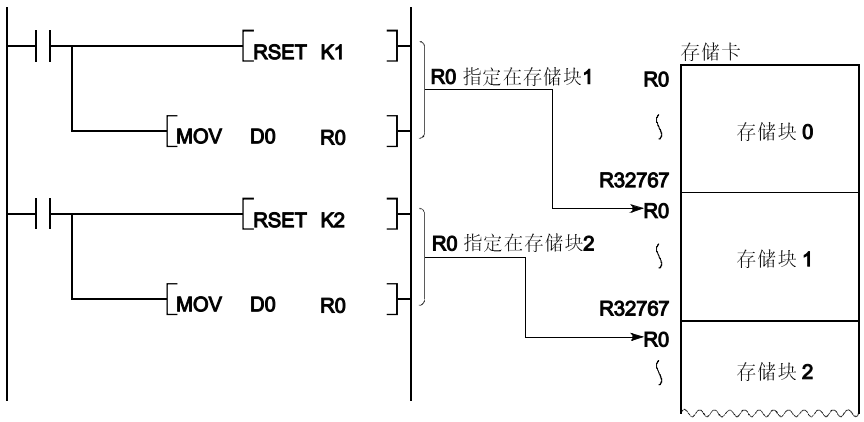
该按钮是用来存储一个规定点数的文件寄存器文件到指定的 QCPU 的存储器中。

10 - 48

10.7.4 文件寄存器的指定方法

(1) 存储块切换格式

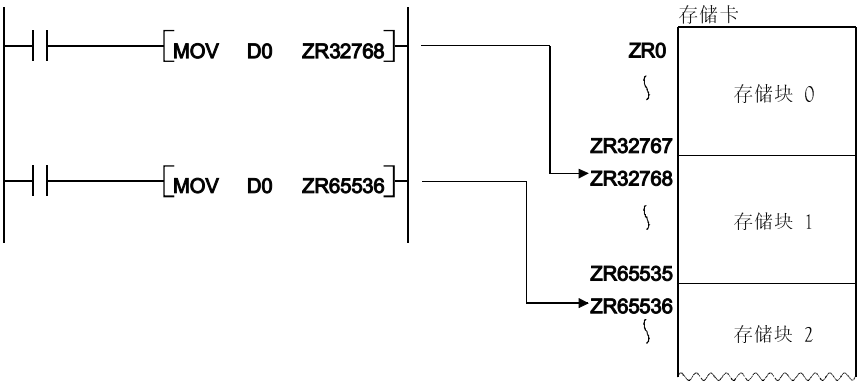
存储块切换格式指定以 **32K** 点（**R0** 到 **R32767**）为单位的文件寄存器点的数目。
如果：使用多个存储块。**RSET** 指令可用来切换到另一个编号的存储块作进一步的文件寄存器设定。



在每一块中，指定在 **R0-R32767** 的范围内设定。

(2) 系列编号存取格式

该格式用于不切换存储块编号情况下超过 **32K** 点的文件寄存器的设定。
多个块的文件寄存器可以用作为一个连续的文件寄存器。



10 - 49

10.7.5 使用文件寄存器的注意事项

(1) 使用未注册的或在注册范围之外的文件寄存器地址。

(a) 当文件寄存器文件未在 QCPU 中注册，即使对文件寄存器进行读/写也不会出错。
从文件寄存器读出数据会造成下列结果：
- 未定义的数据存储到标准 RAM 中。
- “OH”被存储到存储卡中。

(b) 读/写到注册范围（点）之外编号的文件寄存器（点）
即使读/写到这些文件寄存器，也不会出错。
从文件寄存器读出数据，会造成下列后果：
- 未定义的数据存储到标准 RAM 中
- “OH”被存储到存储卡中。

(2) 文件寄存器容量检查

(a) 对文件寄存器进行容量检查以使为 QCPU 而作出的文件寄存器的大小（点）的设定能允许对文件寄存器进行数据的写入和读出。
- 对文件寄存器的容量检查应该在所使用的文件寄存器的程序的“0”步执行。
- 在切换到另一个使用 QDRSET 指令文件寄存器文件以后，执行文件寄存器容量检查。
- 当使用 RSET 指令来切换存储块时，在执行 RSET 指令之前检查切换目标存储块是否有 1K 点或更大的容量。

（文件寄存器容量）>[32K 点×（切换存储块编号）+1K 点]

(b) 可使用的文件寄存器的容量可以在文件寄存器容量存储寄存器中（SD647）检测到。*1
文件寄存器的容量以 1K 点的单位存储在 SD647 中。

文件寄存器中“小于 1K 点”的剩余部分，不被存储。
为了保证精确地检查“使用的范围”。一定要以 1K 点（1024 点）为单位指定文件寄存器设定。

备注

*1 : 如果一个文件寄存器文件被切换到另一个，当前所选择的文件寄存器的容量存储在 SD647。

10 - 50

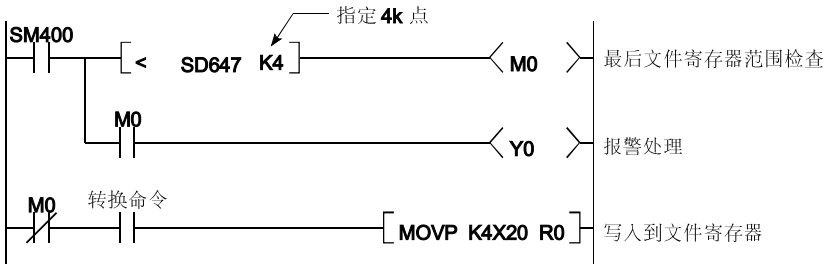
(c) 检查文件寄存器容量

1) 检查用于每一个顺控程序文件寄存器的容量。

2) 根据在顺控程序中 SD647 中设置的文件寄存器的总容量来决定文件寄存器的容量是否大于所使用的点数。

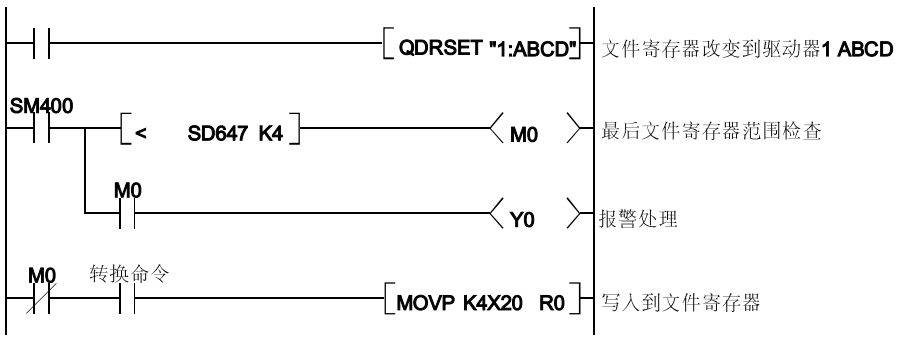
[程序举例 1]

在每一程序开始时，检查文件寄存器的“使用范围”。



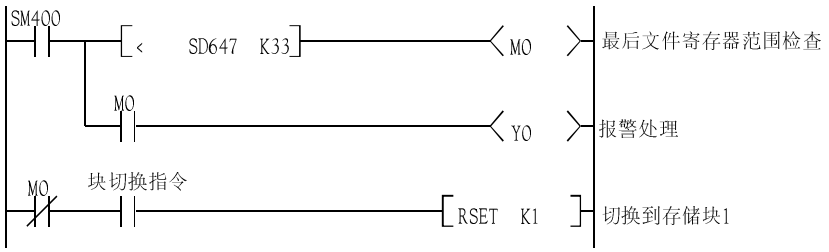
[程序举例 2]

在执行 QDRSET 指令以后，检查文件寄存器的“使用范围”。



[程序举例 3]

用于块切换。



(3) 删除文件寄存器

要擦除不想要的文件寄存器文件，在线执行 PLC 数据删除。

10.8 嵌套(N)

(1) 定义

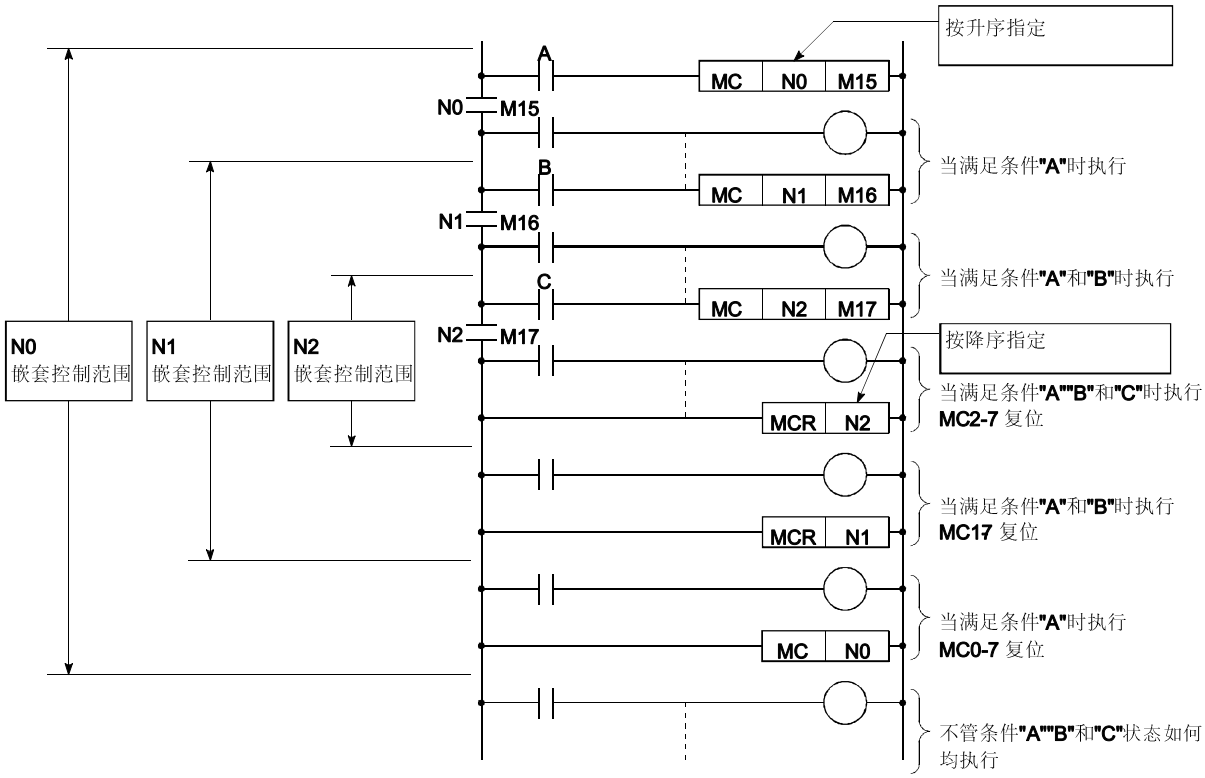
在编程运行条件下，嵌套软元件是用来嵌套 **MC** 或 **MCR** 主控指令的。

(2) 主控制指定法

主控指令是用来打开和关闭梯形图的公用总线的，从而可以用顺控程序有效地来执行梯形图的切换。

嵌套软元件必须按照嵌套关系的升序（从 **N0** 到 **N7**）来编号。

如何使用主控的细节，参考 **QCPU**（**Q** 模式）**/QnACPU** 编程手册（公用指令）。



10 - 52

10.9 指针

(1) 定义

指针软元件是用于转移指令（CJ SCJ JUMP）或子程序调用指令（ALL ECALL）的。可以使用总数为 4096 指针（用于所有正被执行的程序）。
可以使用的指针总数为 4096（用于所有正在执行的程序的总数）。

(2) 指针应用

(a) 在转移指令中（CJ SCJ JUMP），指针是用来指定转移的目的地和标贴（转移目的地起始址）。

(b) 在子程序调用指令（CALL，CALLP）中，指针用来指定调用的目的地和标贴（子程序起始址）。

(3) 指针类型

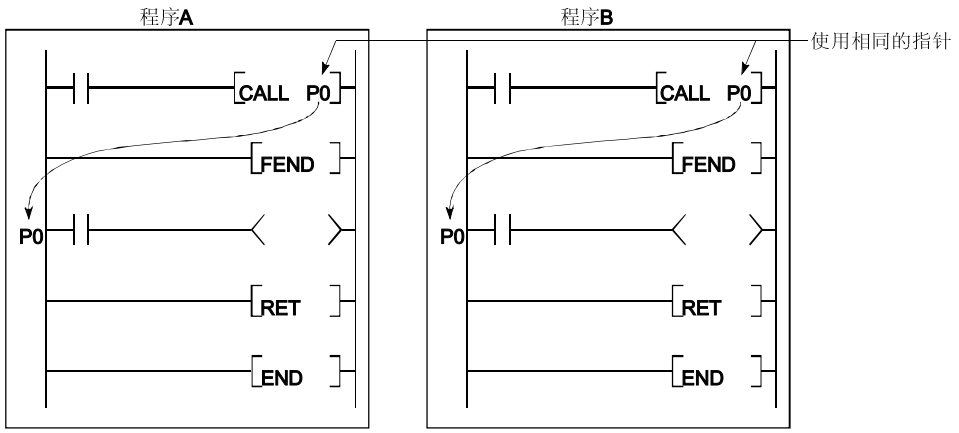
有两种指针类型：在 QCPU 程序中独立地使用的“本地指针”（10.9.1）和从 QCPU 执行的所有程序中调用子程序时使用的“公用指针”（10.9.1）。

10.9.1 本地指针

(1) 定义

(a) 本地指针是在 QCPU 程序转移指令和子程序调用指令中独立使用的指针。
其它程序的转移指令和子程序调用指令不能使用本地指针。
用“ECALL”指令可以调用在含有本地指针的程序文件中的子程序。

(b) 相同编号的指针可用于不同程序中



备注

有关转移指令和子程序调用指令的详细信息，参阅 QCPU（Q 模式）/QnACPU 编程手册（公用指令）。

(2) 本地指针的点数

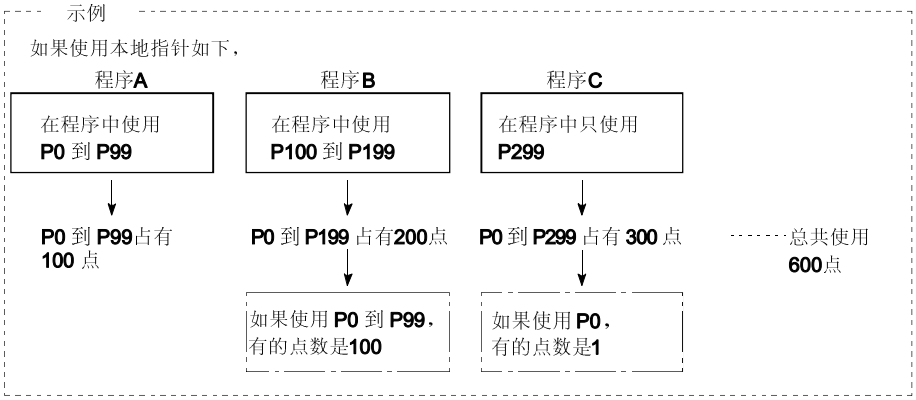
在存储在程序存储器中的所有程序中，可以分派本地指针。

本地指针的编号范围从 P0 到使用中的本地指针的最高编号。（QCPU 的操作系统计算使用的点的数目）。

即使在程序中只用到 P99，例如，使用的点数将被计数为 100（在 P0-P99 之间）。

因此当在几个程序中使用本地指针时，指针设定应从 P0 开始。

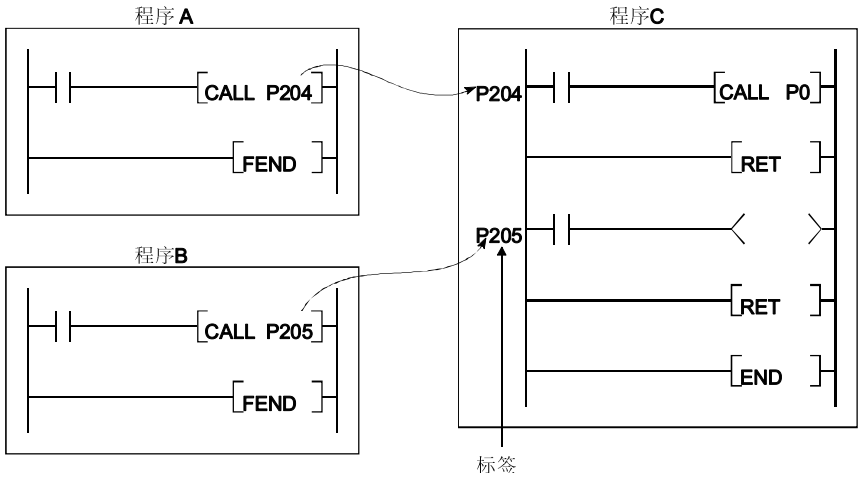
如果指针的总数（用于所有程序）超过 4096 点，将会发生指针配置出错（出错码：4020）。



10.9.2 公共指针

(1) 定义

(a) 公用指针是用于从正在 QCPU 中执行的所有程序中调用子程序。



(b) 同一指针编号不能作为标贴再次使用。

这样做会发生指针配置错误（错误码：4021）。

(2) 公用指针使用范围

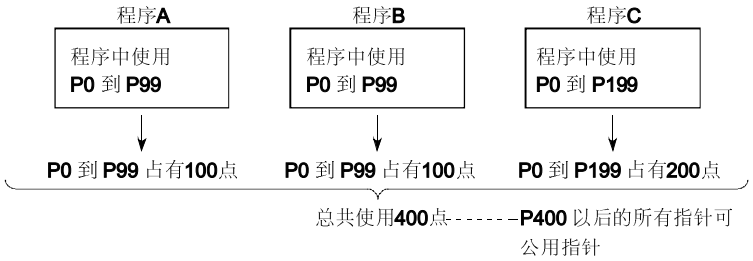
为了使用公用指针，必须在用 **PLC** 参数设定的 **PLC** 系统设定中指定第一个公用指针的编号。

公用指针的范围从指定编号的指针开始到 **P4095**。

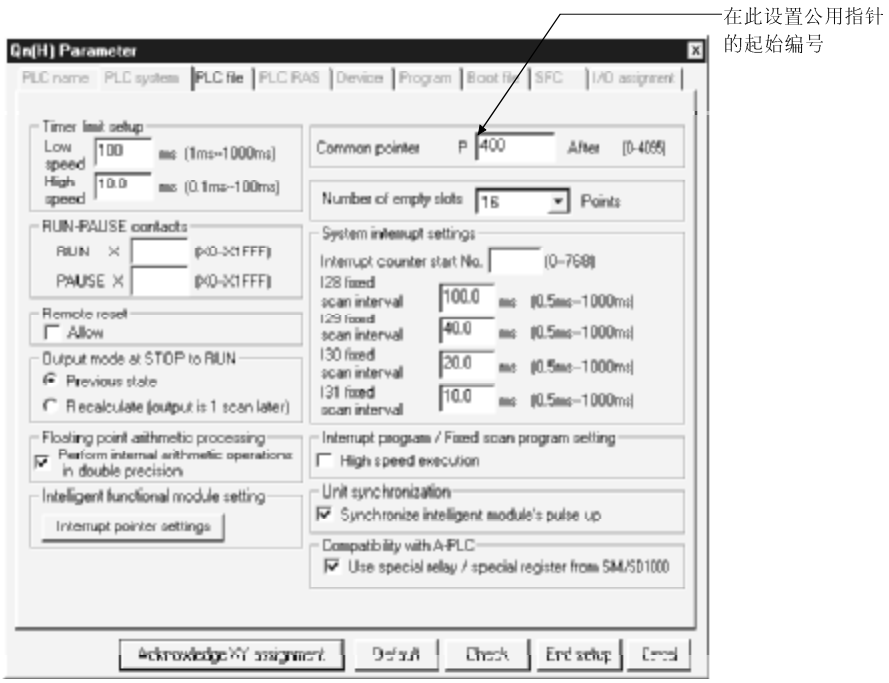
然而，只有在本地指针范围之后的指针编号可由参数设定来指定为公用指针。

如果在三个程序中使用总量为 **400** 点（**100** 点用于程序 **A**，**100** 点用于程序 **B** 和 **200** 点用于程序 **C**），举例来说：在 **400** 点以后的所有本地指针可用作公用指针。

如果，在几个程序中使用的本地指针的最后一个编号与公用指针的第一个编号重叠，将发生“指针配置出错”（出错码：**4020**）



[公共指针设置屏幕]

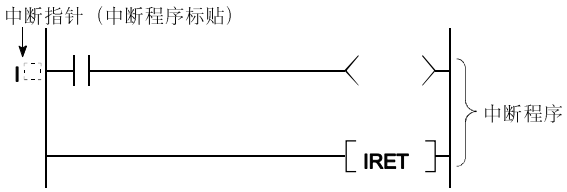


注意
(1) 在转移指令中，不允许转移到其它程序的公用指针。 公用指针只可与子程序调用指令一起使用。

10.10 中断指针（I）

(1) 定义

(a) 中断指针是在中断程序开始时用作标贴的中断指针（中断程序标贴）



(b) 总共可使用 256 个中断点（10-1255）（用于正在执行的所有程序的总数）。

(2) 中断指针编号和中断因素

(a) 如下所示，有四种中断因素。

- Q160/A1SI61 因素 .....从 QI60/A1SI61 中断模块来的中断输入
- 顺序启动发生器 .....从能对 QCPU 发出启动中断命令的特
  模块因素　　殊功能模块来的中断因素。
- 内部时间因素 .....由 QCPU 内部定时器发生的固定周期中断因素。
- 出错中断.....不停止顺控程序运行的出错产生的中断因素。
- 智能功能模块 .....由智能功能模块产生的中断因素。
  中断

备注

*：要使用智能功能模块中断，对 PLC 参数的 PLC 系统设定需要智能功能模块设定（中断点设定）。
（对来自智能功能模块的中断因素，参见 8.2.1 节）

10 - 56

(b) 中断指针的编号和中断因素见以下的表 10.5:

表 10.5 中断指针编号和中断因素表

I No.	中断因素		优先级	I No.	中断因素		优先级
I0	QI60/A1SI61 中断模块因素	1st 点	237	I32*2	出错因素*3 *4	停止运行的出错	1
I1		2nd 点	238	I33		空闲	——
I2		3rd 点	239	I34		UNIT VERIFY ERR. FUSE BREAK OFF SP. UNIT ERROR	2
I3		4th 点	240				
I4		5th 点	241				
I5		6th 点	242	I35		OPERATION ERROR SFCP OPE. ERROR SFCP ECE. ERROR EX. POWER OFF	3
I6		7th 点	243				
I7		8th 点	244				
I8		9th 点	245				
I9		10th 点	246	I36		ICM. OPE ERROR FILE OPE. ERROR	4
I10		11th 点	247				
I11		12th 点	248	I37		空闲	——
I12		13th 点	249	I38		PRG. TIME OVER	5
I13		14th 点	250	I39		CHK 指令执行信号报警器检测	6
I14		15th 点	251				
I15		16th 点	252	I40 - I49	——	空闲	——
I16	顺序启动发生器模块因素*1	1st 点	224				
I17		2nd 点	225				
I18		3rd 点	226				
I19		4th 点	227	I50- I255	智能功能模块因素*5	用参数 指定使用哪一块智能功能模块	18- 223
I20		5th 点	228				
I21		6th 点	229				
I22		7th 点	230				
I23		8th 点	231				
I24		9th 点	232				
I25		10th 点	234				
I26		11th 点	235				
I27		12th 点	236				
I28	内部定时器因素*2	100ms	256				
I29		40ms	255				
I30		20ms	254				
I31		10ms	253				

备注

1) *1 : 从第一点到第 12 点按顺序分配，从安装在最靠近 QCPU 的顺序启动发生模块开始。
2) *2 : 所表示的内部时间是缺省的设定时间。
可由 PLC 参数设定的 PLC 系统设定以 0.5ms 为单位从 0.5ms 到 1000ms 的范围指定这些时间。
3) *3 : 当出错中断“132”（停止运行出错）发生时，直到“132”处理完成，QCPU 不会停止运行。
4) *4 : 当电源接通时和在 QCPU 复位期间，对“132”到“139”的中断指针编号，禁止执行出错中断。当使用编号“132”到“139”的中断指针时，使用 IMASK 指令来设置允许中断状态。

10 - 57

10.11 其它软元件

10.11.1 SFC 块软元件 (BL)

本软元件是用来检查由 SFC 程序指定的模块是否是活动的。
有关 SFC 块软元件使用的细节，参考 QCPU（Q 模式）/QnACPU 编程手册（SFC）。

10.11.2 SFC 传送软元件(TR)

本软元件是用来检查在指定的 SFC 程序块中是否对指定的传送条件指定了强制传送。
有关使用 SFC 传送软元件的细节，参考 QCPU（Q 模式）/QnACPU 编程手册（SFC）。

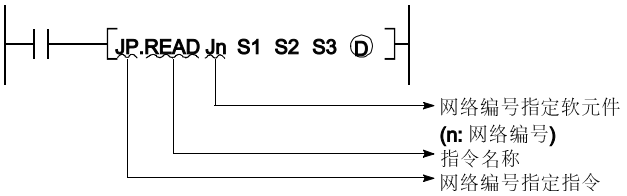
10.11.3 网络编号指定软元件(J)

(1) 定义

网络编号指定软元件是用来指定数据通讯指令中的网络编号。

(2) 指定网络编号指定软元件

用数据通讯指令指定网络编号指定软元件如下图所示：



备注

有关数据通讯指令的细节，参考 Q 模式 MELSECNET/H 网络系统的参考手册。

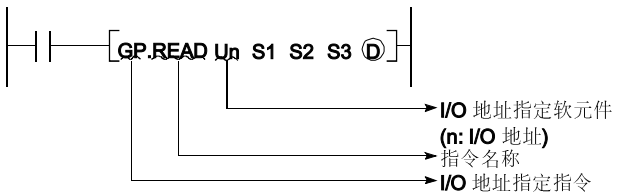
10.11.4 I/O 地址指定软元件(U)

(1) 定义

I/O 地址指定软元件是使用智能功能模块指令来指定 I/O 地址的。

(2) 指定 I/O 地址指定软元件

用智能功能模块指令来指定 I/O 地址指定软元件，如下图所示。



备注

有关智能功能模块指令细节，参考所使用的智能功能模块的相应手册。

10 - 59

10.11.5 宏指令变量软元件(VD)

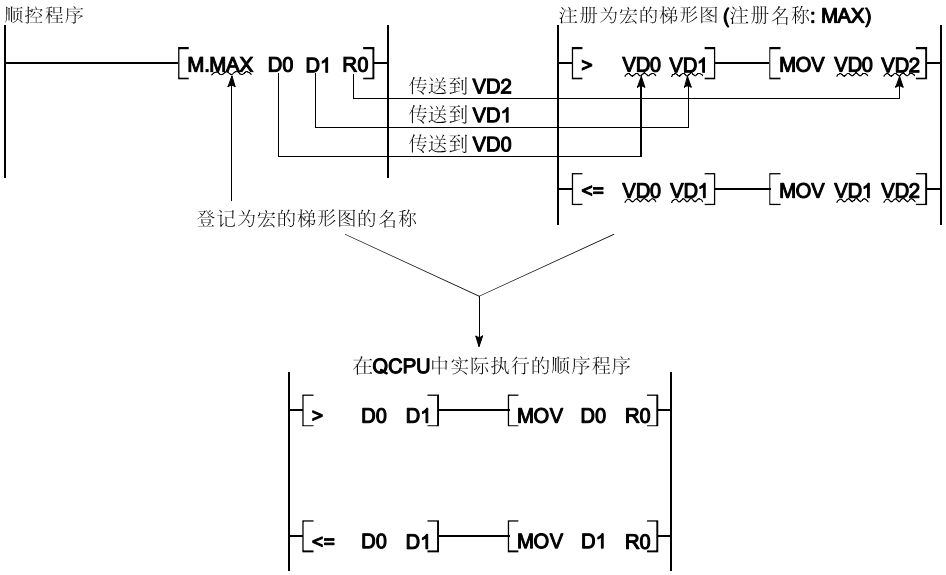
(1) 定义

宏指令变量软元件是与注册为宏的梯形图一起使用的。当指定 VD□设定为注册为宏的梯形图，则在执行宏指令时，转换成指定的软元件。

(2) 指定宏指令变量软元件

指定宏指令变量软元件用于在外围设备的宏注册时注册为宏指令的梯形图中，设置为 VD 的软元件。*

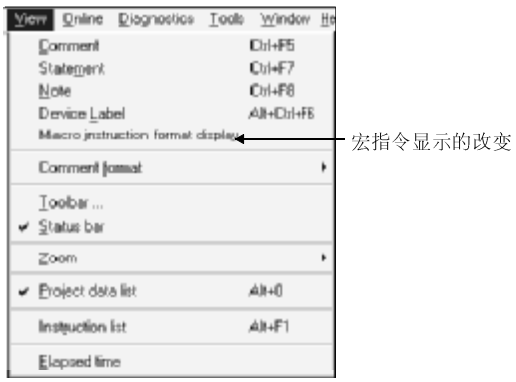
在顺控程序中使用宏指令时，要以升序来指定响应软元件，它对与注册为宏的梯形图一起使用的指令变量软元件作响应。



备注

1) * : 使用宏指令变量软元件时，VD0 到 VD9 可以用在一个注册为宏指令的梯形图中。

2) GX-Developer 的读模式提供一种选择，即：可以用宏指令格式来观察程序（选择“视图”—“宏指令格式显示”来观察宏指令。



10 - 60

10.12 常数

10.12.1 十进制常数(K)

(1) 定义

十进制常数是在顺控程序中指定十进制数据的软元件，
指定为“K□”的设定形式（例：K1234），并以二进制码的形式存储在 QCPU 中。
有关十进制码的细节参见 4.8.1 节。

(2) 指定范围

十进制常数的设定范围如下：

- 字数据（16 位）.....K-32768 到 K32767
- 2 字数据（32 位）.....K-2147483648 到 K2147483647

10.12.2 十六进制常数(H)

(1) 定义

十六进制常数是在顺控程序中指定十六进制或 BCD 数据的软元件。
(BCD 数据指定是使用 0 到 9 数字。)
十六进制常数指定为“H □”的设定形式（例：H1234）。
有关十六进制码的细节参见 4.8.3 节。

(2) 指定范围

十六进制常数的设定范围如下

- 字数据（16 位..... H0 到 HFFFF(BCD 码是 H0 到 H9999)
- 二字数据(32 位) ..... H0 到 HFFFFFFFF(BCD 码是 H0 到 H99999999)

10 - 61

10.12.3 实数(E)

(1) 定义

实数是在顺控程序中指定实数的软元件。

实数指定为“E□”设定形式（例 E1234）。



有关实数的细节参见 4.8.4 节。

(2) 指定范围

实数的设定范围是-1.0×2^{128}到-1.0×2^{-126}, 0 和 1.0×2^{-126}到 1.0×2^{128}.

(3) 指定方法

实数在顺控程序中可用“常规表达”或“指数表达”方式指定。

•常规表达方式.....指定值指定为其本身

例如: 10, 2345 表达为 E10.2345

•指数表达方式.....指定值被乘以一个“10”

例如：1234 表达为 **E1.234 + 3.***

备注

1) *: "+3"在上述例子中代表 10^n 值（10^3）。

10.12.4 字符串(")

(1) 定义

字符串常数是用来在顺控程序中指定字符串的软元件, 字符串和引号来指定（例：“ABCD1234”）

(2) 使用的字符

所有的 ASCⅡ码字符可以以字符串的形式使用。

QCPU 能区分大写和小写字符。

(3) 指定字符的数目

字符串从指定的字符扩展到 NUL 代码（00_H）。

在如 $MOV 的指令中，可以最多使用 32 个字符作为字符串。

10.13 软元件的简便用法

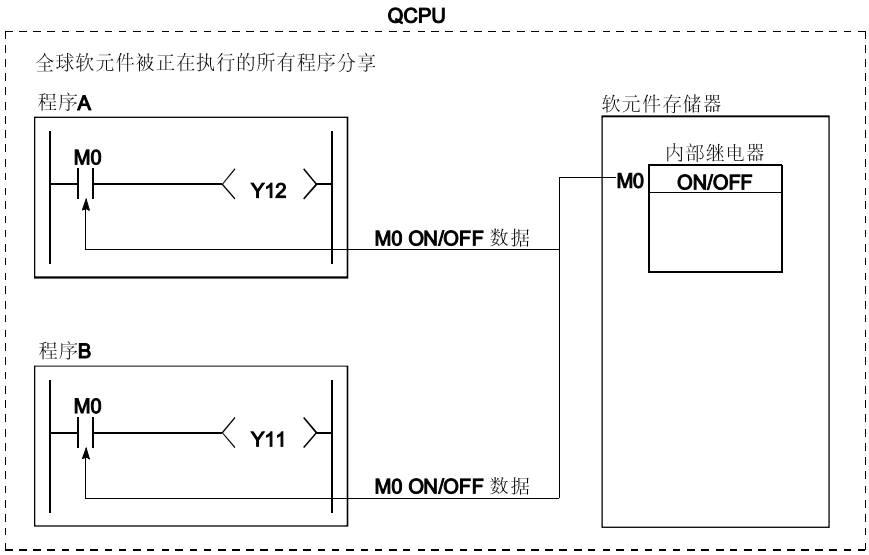
在 QCPU 中执行多个程序时，在内部用户软元件之中的本地软元件可以被指定来以独立方式执行多程序中的一个程序。
而且，软元件的初始值设定可以用来指定软元件和智能功能模块/特殊功能模块数据设定而不必使用程序。

10.13.1 全局软元件和本地软元件

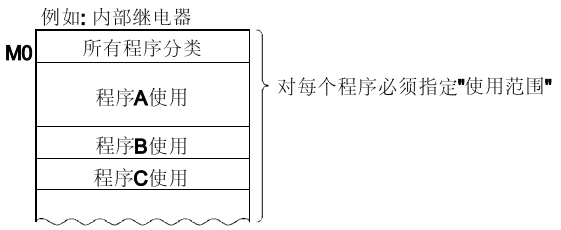
在 QCPU 中可以存储和执行大量的程序。
可以被所有程序分享的 QCPU 软元件是“全局软元件”，而被程序中的一个程序独立使用的软元件称为“本地软元件”。

(1) 全局软元件

(a) 全局软元件是可以被所有的在 QCPU 中正在执行的程序分享的软元件。全局软元件存储在 QCPU 的软元件存储器中，所有的程序使用相同的软元件。



(b) 在执行多个程序时，必须事先指定所有程序“分享的范围”和每个程序的“独立的范围”。

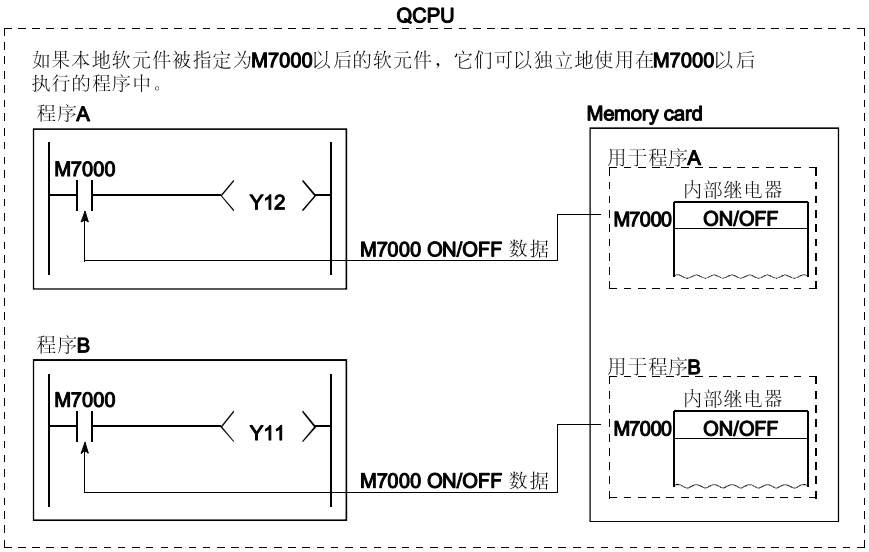


10 - 63

(2) 本地软元件

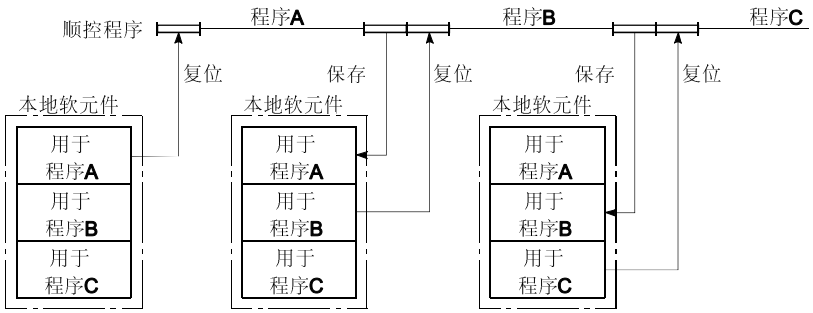
(a) 本地软元件是由程序独立地使用的软元件

使用本地软元件允许多个“独立执行”的程序的编程而不必考虑其他程序。
然而，本地软元件数据可以存储在标准 RAM 和存储卡中。
然而，本地软元件数据可以存储在标准 RAM 和存储卡中。



(b) 五种类型的软元件可以用作本地软元件：内部继电器（M），边沿继电器（V），定时器（T.ST），计数器（C），和数据寄存器（D）。

(c) 使用本地软元件时，存储在标准 RAM 或存储卡中的本地软元件文件数据与 QCPU 软元件区中的数据发生交换。
因此，这种数据交换时间延长了扫描时间。



备注

1) * : 有关本地软元件“字的数目”细节，参见：10.1.2(第 2 页)。

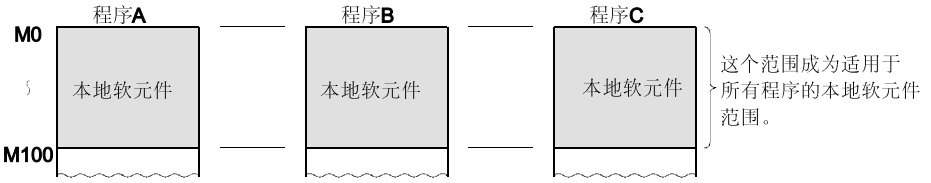
10 - 64

(d) 本地软元件指定

1) 为了将上述软元件作为本地软元件使用，必需在用 PLC 参数设定的软元件设定中，指定一个本地软元件使用设定的范围。

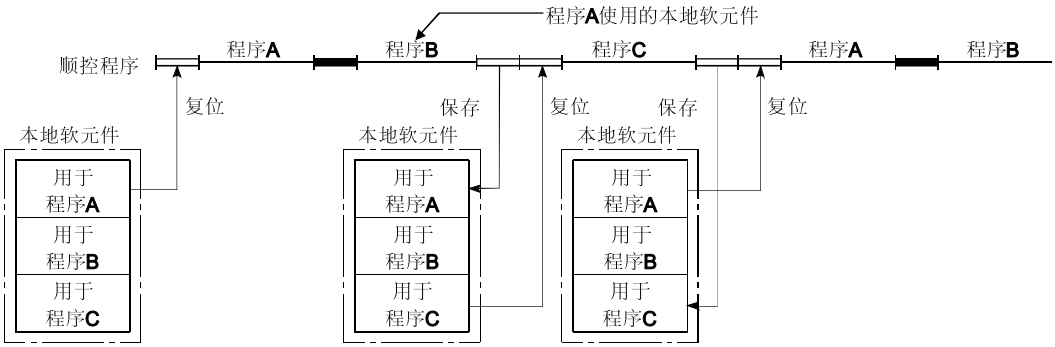
注意：指定给本地软元件的范围适用于所有的程序，并不能改变为用于个别程序。

例如：如果指定本地软元件的范围是 M0 到 M100，这个范围适用于所有的程序中的本地软元件。



2) 在指定本地软元件设定时，必须在用 PLC 参数设定的 PLC 文件设定中指定存储本地软元件数据的驱动器和文件名称。

3) 要用 GPPW 写数据到 QCPU，可用 PLC 参数对话框的 PLC 文件表来指定是否使用本地软元件。如果不指定本地软元件，它就会选择先前执行程序所使用的本地软元件，这不需要用 QCPU 的软元件存储器来替换存储卡中的本地软元件。如果在执行程序 A，B 和 C 时，程序 B 不使用本地软元件，则本地软元件的使用如下图所示：



注意
(1) 除非特别指定为“本地软元件”，所有的软元件都是“全局软元件”。

(e) 使用存储子程序程序的文件所使用的本地软元件

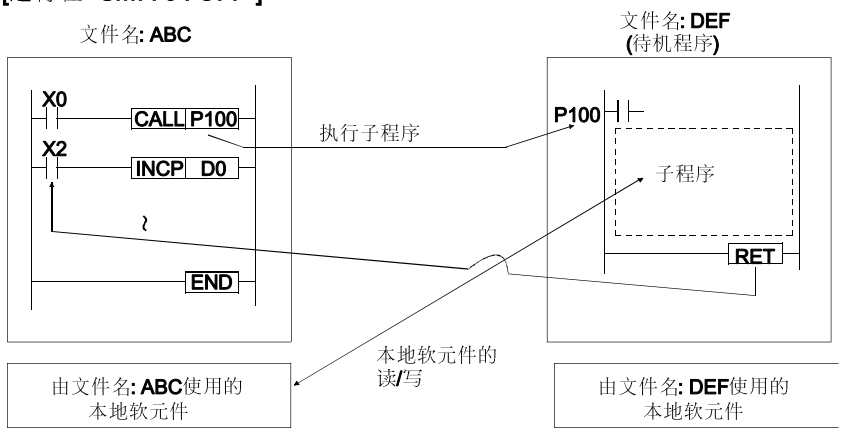
在执行子程序程序时，就可以使用存储子程序程序的文件所使用的本地软元件。

是否使用这个本地软元件是用特殊继电器“SM776”ON/OFF 设定来设置的。

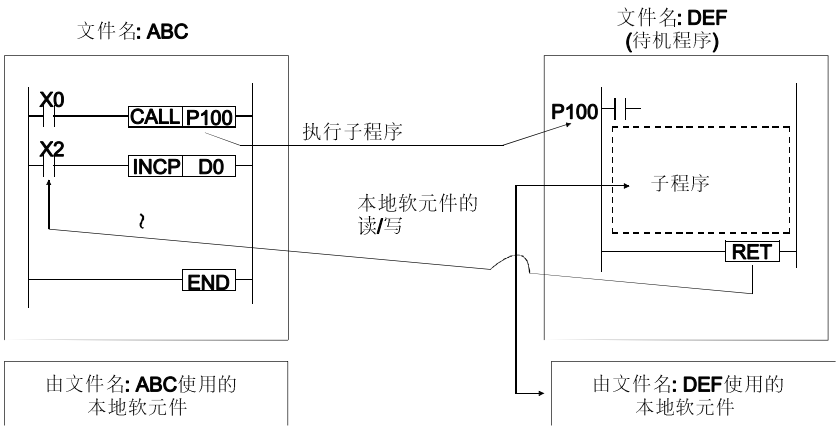
1) 通过为特殊继电器（SM776）设定 ON/OFF 来切换本地软元件。

	SM776
OFF	由调用子程序的文件所使用的本地软元件来执行计算
ON	由存储子程序的文件所使用的本地软元件来执行计算

[运行在 "SM776 : OFF"]



[在 "SM776 : ON"时的运行]



2) 注意事项

- 如果 SM776 在 ON 的状态，在调用子程序程序时，从本地软元件中读出数据，在执行了 RET 指令以后再将本地软元件数据保存起来。因此，一旦“SM776：ON”的设定完成后，执行子程序程序时，因如下所示的时间的原因，扫描时间被延长。 (参见 10.13.1)
- SM776 的 ON/OFF 设定可以在 CPU 模块中进行。
  但不能在文件单元中进行。
- 如果在执行顺控程序时 SM776 的 ON/OFF 设定改变，就将依照改变后的信息进行控制。

备注

SM776 的细节请参阅本手册的附录 1。

10 - 66

(f) 在执行中断/固定扫描执行型程序时,使用本地软元件.

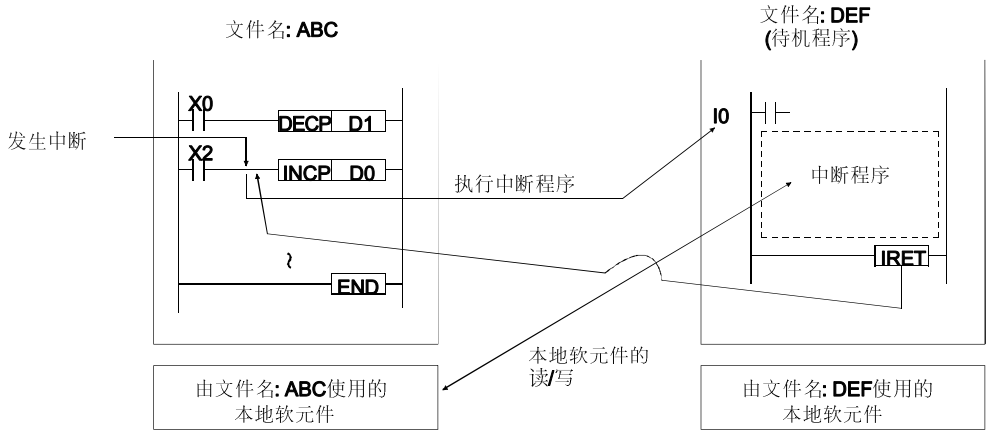
在执行中断/固定扫描执行型程序时,使用存储中断/固定扫描执行型程序的文件所使用的本地软元件是可能的.

是否使用这个本地软元件是由特殊继电器"SM777"的 ON/OFF 设定来设置的。

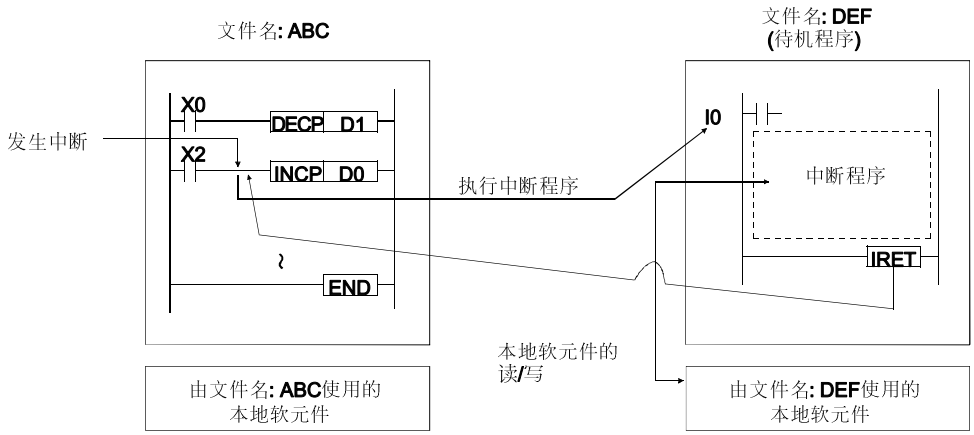
1) 通过特殊继电器的 ON/OFF 设定（SM777）来切换本地软元件。

	SM777
OFF	在执行中断/固定扫描执行型程序之前执行的文件所使用的本地软元件执行计算
ON	由存储中断/固定扫描执行型程序的文件使用的本地软元件来执行计算

[运行在 "SM777 : OFF"]



[运行在 "SM777 : ON"]



备注

SM776 的细节请参阅本手册的附录 1。

10 - 67

2) 警告

•如果 SM777 处于 ON 的状态，在中断/固定扫描执行型程序执行之前，读出本地软元件数据。在 IRET 指令执行以后，本地软元件数据被保存起来。因此一旦“SM777：ON”设定完成，执行中断/固定扫描执行型程序时，扫描时间增加。(参见 10.13.1)

• SM777 的 ON/OFF 设定可能在 CPU 模块中进行。
但不能以文件为单位中设置。

•如果在执行顺控程序时改变 SM777 的 ON/OFF 设定，就将依照改变后的信息进行控制。

(g) 清除本地软元件数据

在下列情况下，本地软元件数据被清除

1) 在 QCPU 接通电源或复位时

2) QCPU 从 STOP 状态进入 RUN 的状态时。

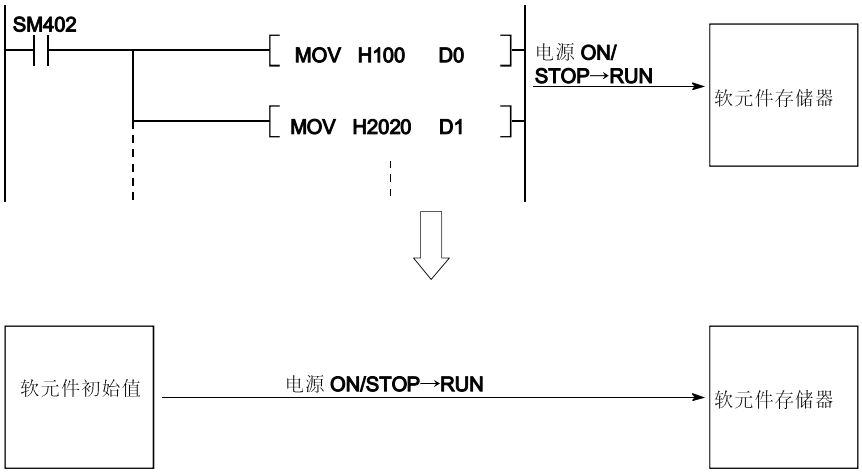
本地软元件数据不会因 GX—Developer 的操作被清除。要清除本地软元件数据可按照以上列出的步骤 1）和 2）进行。

10.13.2 软元件初始值

(1) 定义

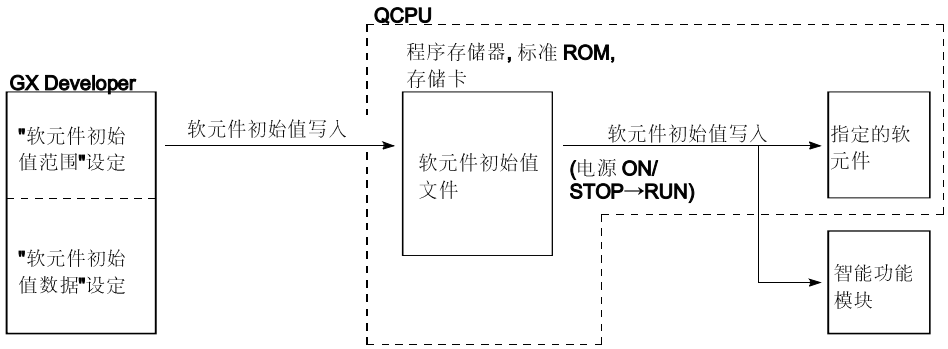
(a) 用软元件初始设定值，用于一个程序的数据可存储在软元件或智能功能模块缓冲存储器中而不必使用数据设定程序。

软元件初始值的使用提供了一个快捷方法来指定程序中的软元件数据，而不必使用一个软元件数据设定程序（初始化程序）。



[使用初始化处理程序的数据设定]

(b) 为了使用软元件初始值，必须在 GX Developer 软元件中事先创建软元件初始值，这个数据必须作为一个软元件初始值文件存储在 QCPU 的程序存储器、标准 RAM 或存储卡中。
在电源接通时，或者从 STOP 切换到 RUN 时，QCPU 把数据从软元件初始值文件写入到指定的软元件或智能功能模块的缓冲存储器中。



10 - 69

(c) 软元件初始值可以使用于下列软元件

1) 定时器当前值(T)
2) 积算定时器当前值(ST)
3) 计数器当前值(C)
4) 数据寄存器(D)
5) 特殊寄存器(SD)
6) 通信寄存器(W)
7) 特殊通信寄存器(SW)
8) 文件寄存器(R0 至 R32767)
9) 智能功能模块软元件
(U□\G□)
10) 链接直接软元件(J□\W□,
J□\SW□)

(2) 使用软元件初始值的步骤

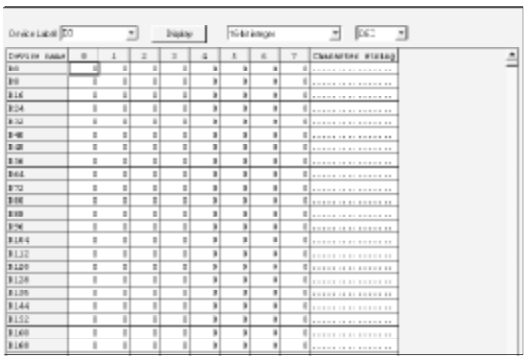
(a) 用软元件模式，在软元件初始值设定屏幕上指定软元件初始值范围设定。

(b) 在软元件模式屏幕上指定软元件初始值数据设定。

[软元件初始化范围设定屏幕]

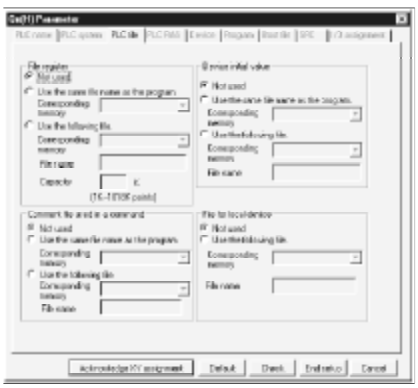


[软元件模式屏幕]



(c) 在用 PLC 参数设定的 PLC 文件设定中，指定存储软元件初始值数据文件的名称。

[PLC 文件屏幕]



(d) 将软元件初始值数据和参数设定写入到 QCPU 中去。

10 - 70

(3) 使用软元件初始值的注意事项

(a) 在软元件初始值数据和锁存范围数据都存在的情况下，软元件初始值数据有优先权。
因此，在电源接通时，锁存范围数据被软元件初始值数据所改写。

(b) 软元件初始值数据不能用在对从 **STOP** 切换到 **RUN** 时不做任何设定的区域，（对在电源接通时，由程序来改变的数据）。在主程序中使用 **MOV** 指令，创建一个程序来指定软元件。使用 **TO** 指令来写入数据到智能功能模块的缓冲存储器。

备注

有关“软元件初始值范围”“软元件初始值数据”项目和将软元件初始值写入到 **QCPU** 的设定步骤的细节，参考 **GX Deleloper** 操作手册。

10 - 71

11. QCPU 的处理时间

本章介绍如何来估算 QCPU 处理时间的长度。

11.1 读取 QCPU 的扫描时间

扫描时间的长度是下列时间的总和

- I/O 刷新时间
- 指令执行时间
- “END”处理时间

(1) I/O 刷新时间

(a) I/O 刷新时间是用来刷新下列模块的输入/输出数据所需时间的总和。

- 输入模块
- 输出模块
- 智能型功能模块（特殊功能模块）

(b) I/O 刷新时间可按下列公式计算：

（I/O 刷新时间）=（输入数/16）×N1+（输入数/16）×N2

(c) 下表给出了 N1 和 N2 的值

CPU 类型	N1(μs)			N2(μs)		
	Q3□B	Q6□B	QA1S□B	Q3□B	Q6□B	QA1S□B
Q02CPU	2.2	2.9	4.3	1.3	2.1	3.5
Q02HCPU Q06HCPU Q12HCPU Q25HCPU	1.7	2.4	3.7	1.3	2.1	3.5

(2) 指令执行时间

(a) 指令执行时间是在 QCPU 的程序中执行一条指令所需时间的总和。
有关每一条指令执行时间的细节参见 QCPU（Q 模式）/QnACPU 编程手册（公共指令）。

(b) 中断程序/固定扫描执行型程序需要额外的时间。应将额外时间加到执行时间中。

(3) END 处理时间

(a) “END”处理时间是对上述（1）和（2）项的一般的 QCPU 处理时间。

(b) 下表列出了“END”处理时间的长度。

CPU 类型	END 处理时间（ms）
Q02CPU	0.38
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.15

11 - 1

11.2 延长扫描时间的影响因素

下列功能增加了扫描时间的长度。在使用下列任何一项功能时，要在 11.1 节列出的时间值上加上延长的时间值。

- MELSECNET/H 刷新
- CC-LINK 自动刷新
- 采样跟踪
- GPPW 监视
- 本地软元件
- 执行多程序
- 存储卡的安装和拆卸
- 有与程序相同文件名称的文件寄存器

(1) MELSECNET/H 刷新

MESECNET/H 刷新需要额外的处理时间，以便在 QCPU 和 MELSECNET/H 网络模块之间刷新数据。有关 MELSECNET/H 刷新时间的细节，参见 MELSECNET/H 网络系统参考手册。

(2) CC-LINK 自动刷新

CC-LINK 自动刷新需要额外的处理时间，以便在 QCPU 与 CC-LINK 主控模块之间刷新数据，当装有 CC-LINK 本地主模块时，通过将 CC-LINK 设定调整到系统设定来缩短延长的扫描时间。有关 CC-LINK 自动刷新时间的细节，参见 QJ618T11 型 CC-LINK 系统本地主模块用户手册。

(3) 采样跟踪时间

采样跟踪需要额外的处理时间，当指定采样数据来执行采样跟踪功能时，总的处理时间要加上采样跟踪时间。

下表列出了将指定采样跟踪数据赋值于 50 个用于位数软元件的内部继电器点和 50 个用于字软元件的数据寄存器点时所需处理时间的长度。

CPU 类型	处理时间 (ms)
Q02CPU	0.24
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.12

(4) GX Developer 监视

GX Developer 监视需要额外的处理时间，GX Developer 监视时间要加到总的处理时间中去。

(a) 下表列出当 64 个数据寄存器点由注册的监视器分配时所需的处理时间。

CPU 类型	处理时间（ms）
Q02CPU	0.10
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.06

(b) 下表列出在指定监视条件时所需的处理时间

CPU 类型	处理时间（ms）	
	步匹配时	软元件匹配时
Q02CPU	0.05	0.01
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.03	0.01

(5) 本地软元件

本地软元件需要额外的处理时间，本地软元件的处理时间要加到总的处理时间中去。

CPU 类型		处理时间（ms）
标准 RAM	Q02CPU	0.94+0.40×n
	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.39+0.17×n
SRAM 卡	Q02CPU	0.94+1.38×n
	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.39+0.95×n

条件：本地软元件：1K 点　n：程序文件的数目

(6) 多程序的执行

多程序的执行需要每个执行程序附加的辅助时间。附加的辅助时间要加到总的处理时间中去。

CPU 类型	处理时间（ms）
Q02CPU	0.08×n
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.03×n

条件：n：程序文件的数目

(7) 存储卡的安装/拆除

安装/拆除一个存储卡需要额外的处理时间，如果一个存储卡被安装或拆除，要增加一次扫描时间到总的处理时间中去。

CPU 类型	处理时间（ms）	
	存储卡插入	存储卡拆下
Q02CPU	0.16	0.10
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.08	0.04

(8) 文件寄存器

文件寄存器需要额外的处理时间，应把文件寄存器的处理时间加到总的处理时间中去。

CPU 类型		处理时间（ms）
标准 RAM	Q02CPU	1.03
	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.41
SRAM 卡	Q02CPU	0.94+0.2×n
	Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.40+0.1×n

条件：n：程序文件的数目

11.3 缩短扫描时间的影响因素

通过将 PLC 参数设定作如下的改变，可以缩短扫描时间的长度。

- A-PLC（与 A-系列 CPU 兼容）
- 浮点算术运算

(1) A-PLC（与 A 系列 CPU 兼容）

在 PLC 参数的 PLC 系统设定中，设置了“在 SM1000/SD1000 以后使用特殊继电器/特殊寄存器”时，通过设定“在 SM1000/SD1000 以后不使用特殊继电器/特殊寄存器”，可以减少如下表所列的扫描时间数值。

有这种情况下，必须用 Q 系列的专用特殊继电器/特殊寄存器 SM0/SD0 到 SM999/SD999 来替换 A 系列兼容的特殊继电器/特殊寄存器 SM1000/SD1000 到 SM1299/SD1299。

CPU 类型	处理时间（ms）
Q02CPU	0.07
Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU	0.03

(2) 浮点算术运算

根据缺省值，在 PLC 参数对话框的 PLC 系统表中，“以双倍精度执行内部算术运算”的复选框打开。如果关闭此选择框置，会加快包含浮点的指令的执行处理速度。

有关浮点算术运算的细节，参阅 QCPU（Q 模式）/QnACPU 编程手册（公共指令）。

11 - 4

12. 将程序写入 QCPU 的步骤

本章介绍将 GX Developer 创建的程序写入到 QCPU 的步骤。

12.1 写入 1 个程序的步骤

本节介绍写入一个程序到 QCPU 并加以执行的步骤。

12.1.1 在创建一个程序时要考虑的事项

为了创建程序，必须事先设置程序的大小、所使用的软元件点的数目和程序文件的名称等。

(1) 考虑程序的大小

检查 CPU 的程序容量是否足够用来存储程序和参数数据。
CPU 的程序容量如下所示：

- Q02CPU : 28 k 步
- Q02HCPU : 28 k 步
- Q06HCPU : 60 k 步
- Q12HCPU : 124 k 步
- Q25HCPU : 252 k 步

如果 CPU 的容量只够程序使用，参数数据应存储在标准 ROM/存储卡中。

(2) 指定程序文件的名称

必须指定存储在 QCUP 中的程序的名称。
当用 GX Developer 写程序及参数到 QCPU 中、及指定在 QCPU 中执行的程序时，使用这个文件名称。
参见第 6 章有关文件名的细节。

(3) 指定软元件

必须决定程序要使用的软元件数目。
参见第 10 章有关在 QCPU 中可使用软元件的细节。

(4) 软元件的初始值设定

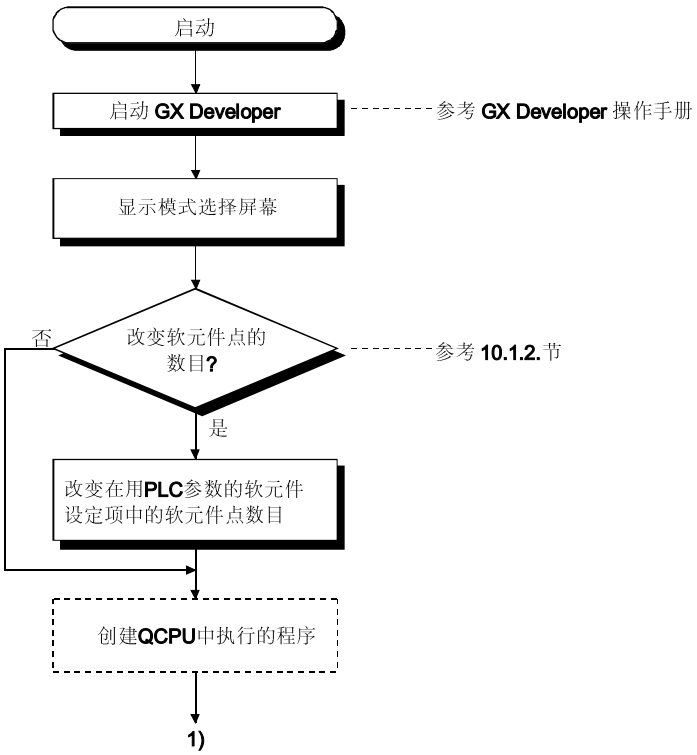
指定在 QCPU 软元件和智能型功能模块中是否使用软元件初始值设定。
参见 10.13.2 节有关软元件初始值的细节。

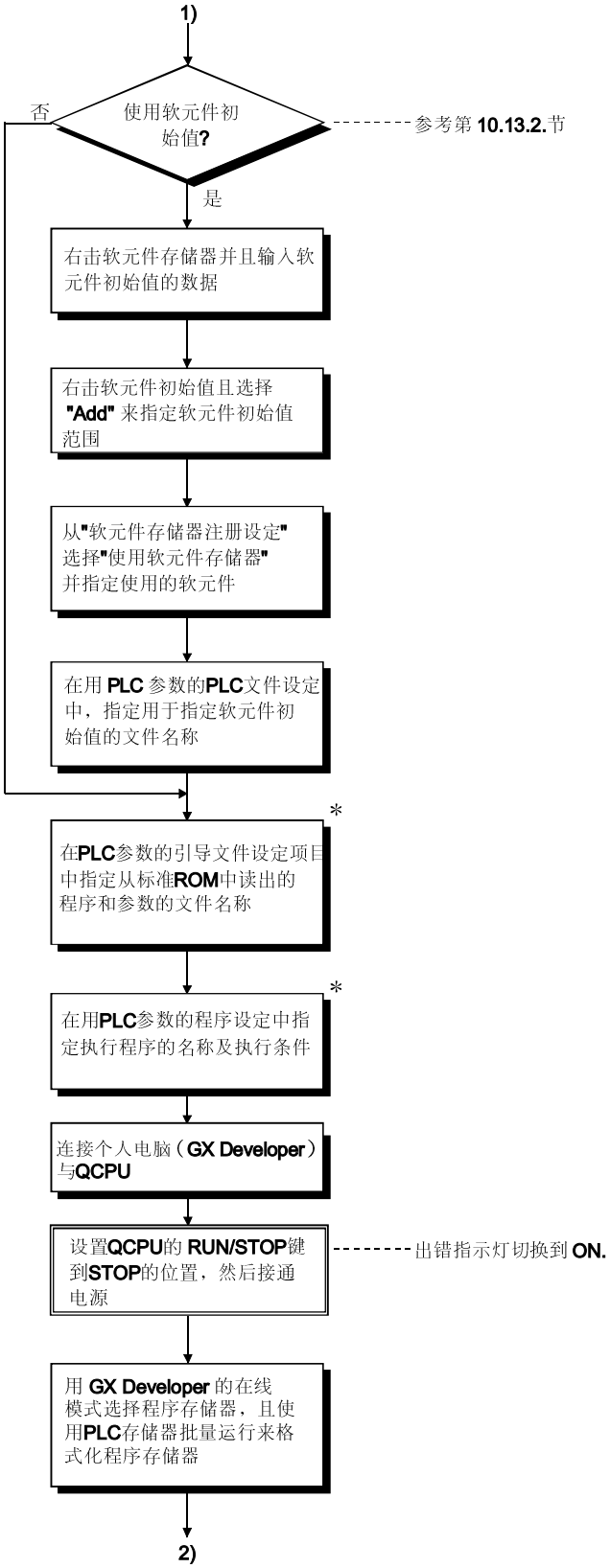
12.1.2 将程序写入 QCPU 的步骤

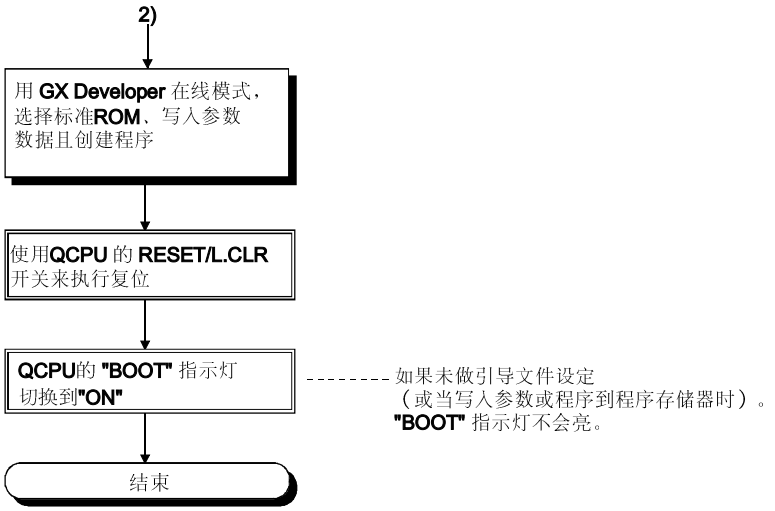
把用 GX Developer 中创建的程序和参数写入到 QCPU 标准 ROM 的步骤如下所示：
为了将程序和参数写入到 QCUP 标准 ROM 中去，必须用 QCPU DIP 开关（SW2，SW3）来指定有效的参数设定，并且用在 PLC 参数模式指定引导设定。
有关 QCPU DIP 开关的细节参考 QCUP（Q 模式）用户手册（硬件设计及维护检查）。

在将程序和参数写入到 QCPU 的程序存储器中时，以下加注(*)号的步骤是不需要的。

在□框中的步骤是用 GX Developer 执行的，而在□框中的则是在 QCPU 中执行的。




1)
否
使用软元件初始值?
参考第 10.13.2.节
是
右击软元件存储器并且输入软元件初始值的数据
右击软元件初始值且选择 "Add" 来指定软元件初始值范围
从"软元件存储器注册设定"选择"使用软元件存储器"并指定使用的软元件
在用 PLC 参数的PLC文件设定中，指定用于指定软元件初始值的文件名称
*
在PLC参数的引导文件设定项目中指定从标准ROM中读出的程序和参数的文件名称
*
在用PLC参数的程序设定中指定执行程序的名称及执行条件
连接个人电脑（GX Developer）与QCPU
设置QCPU的 RUN/STOP键到STOP的位置，然后接通电源
出错指示灯切换到 ON.
用 GX Developer 的在线模式选择程序存储器，且使用PLC存储器批量运行来格式化程序存储器
2)


2)
用 GX Developer 在线模式，
选择标准ROM、写入参数
数据且创建程序
使用QCPU 的 RESET/L.CLR
开关来执行复位
QCPU的 "BOOT" 指示灯
切换到"ON"
结束
如果未做引导文件设定
（或当写入参数或程序到程序存储器时）。
"BOOT" 指示灯不会亮。

12.2 写入多个程序的步骤

本节介绍将按照功能、处理、设计人分类的多个程序写入 QCPU 并加以执行的步骤。

12.2.1 创建多程序时的考虑事项

若要创建多个程序，必需预先设置每个程序的容量，使用的软元件和程序文件名称等。

(1) 考虑程序的容量

在 QCPU 程序容量的范围内设置程序的容量。
QCPU 程序的容量如下所示：
- Q02CPU : 28 k 步
- Q02HCPU : 28 k 步
- Q06HCPU : 60 k 步
- Q12HCPU : 124 k 步
- Q25HCPU : 252 k 步

决定将参数存储在程序存储器、标准 ROM 或是存储卡中。
如果存储在程序存储器/标准 ROM 中，程序可使用的区域是以上所示的容量减去参数数据的容量。

(2) 指定程序的文件名称

指定要存储在 QCPU 中程序的文件名称。
当用 GX-Developer 写入程序、参数到 QCPU 时及指定要在 QCPU 中执行的程序时，可使用这个文件名称。
参见第六章有关文件名称的细节。

(3) 指定执行程序的条件

为了在 QCPU 中执行多个程序，应为每个程序指定执行条件。
没有设定文件名和执行条件的程序是不可执行的。
参见 4.2 节有关执行条件的细节。

(4) 指定软元件

(a) 指定每个程序所使用的软元件点的数目和所有程序分享的软元件点的数目。参见第 10 章有关 QCPU 中可用软元件的细节。
(b) 指定每个程序的内部继电器，边沿继电器，定时器，计数器，和数据寄存器是否要指定为本地软元件。
参见 10.13.1 节有关本地软元件的细节。
(c) 当创建子程序时，指定是否使用公共指针。
参见 10.9.2 节有关公共指针的细节。

(5) 软元件初始值设定

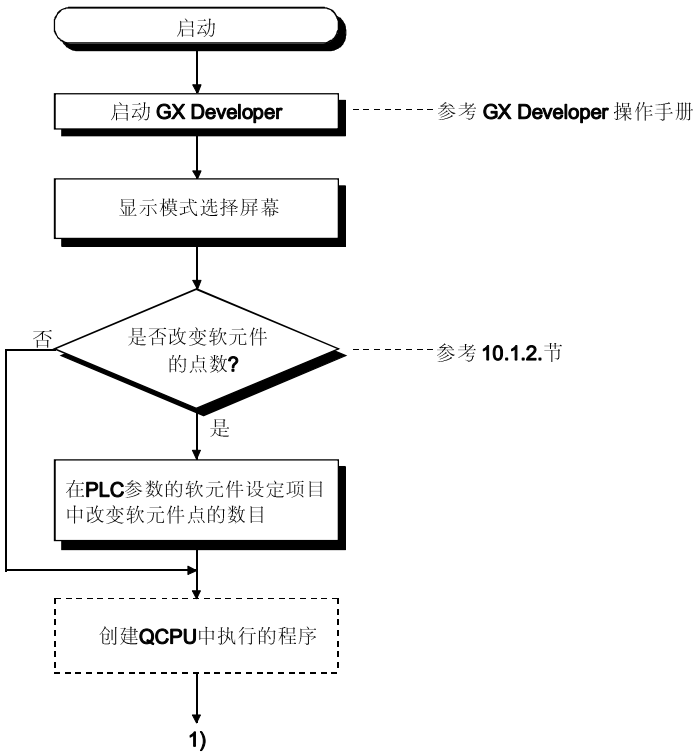
指定软元件初始值设定是否用于 QCPU 软元件和智能型功能模块。
参见 10.13.2 节有关软元件初始值的细节。

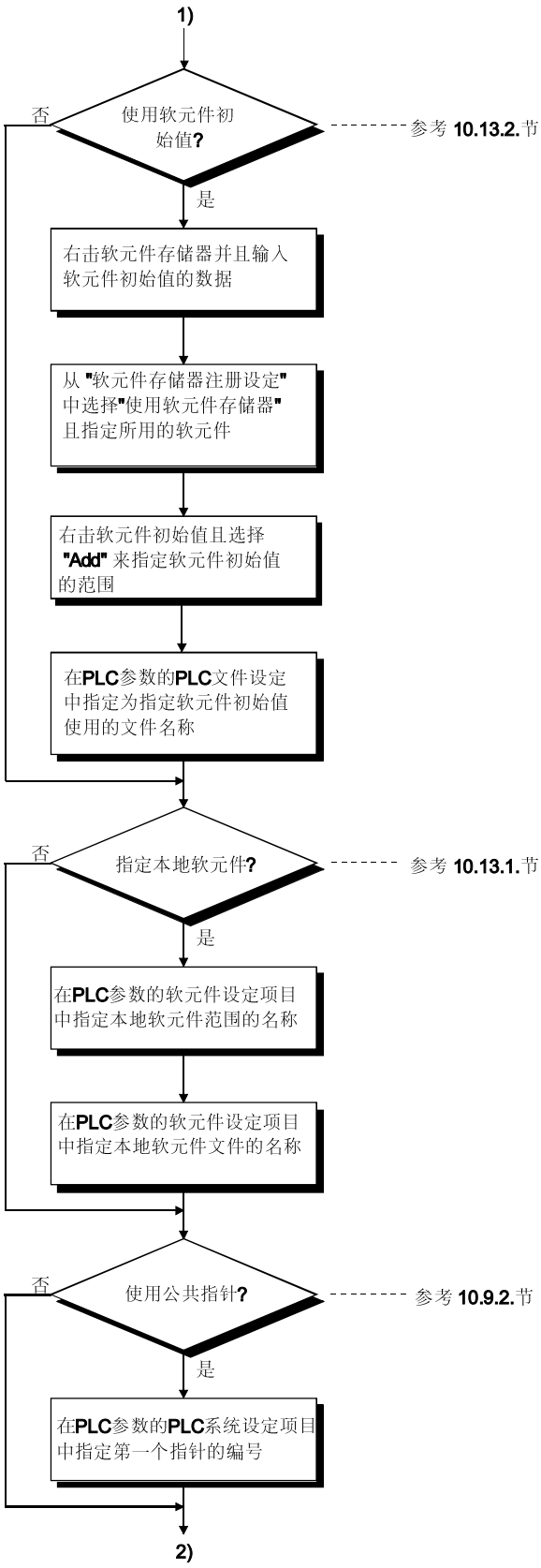
12.2.2 将程序写入 QCPU 的步骤

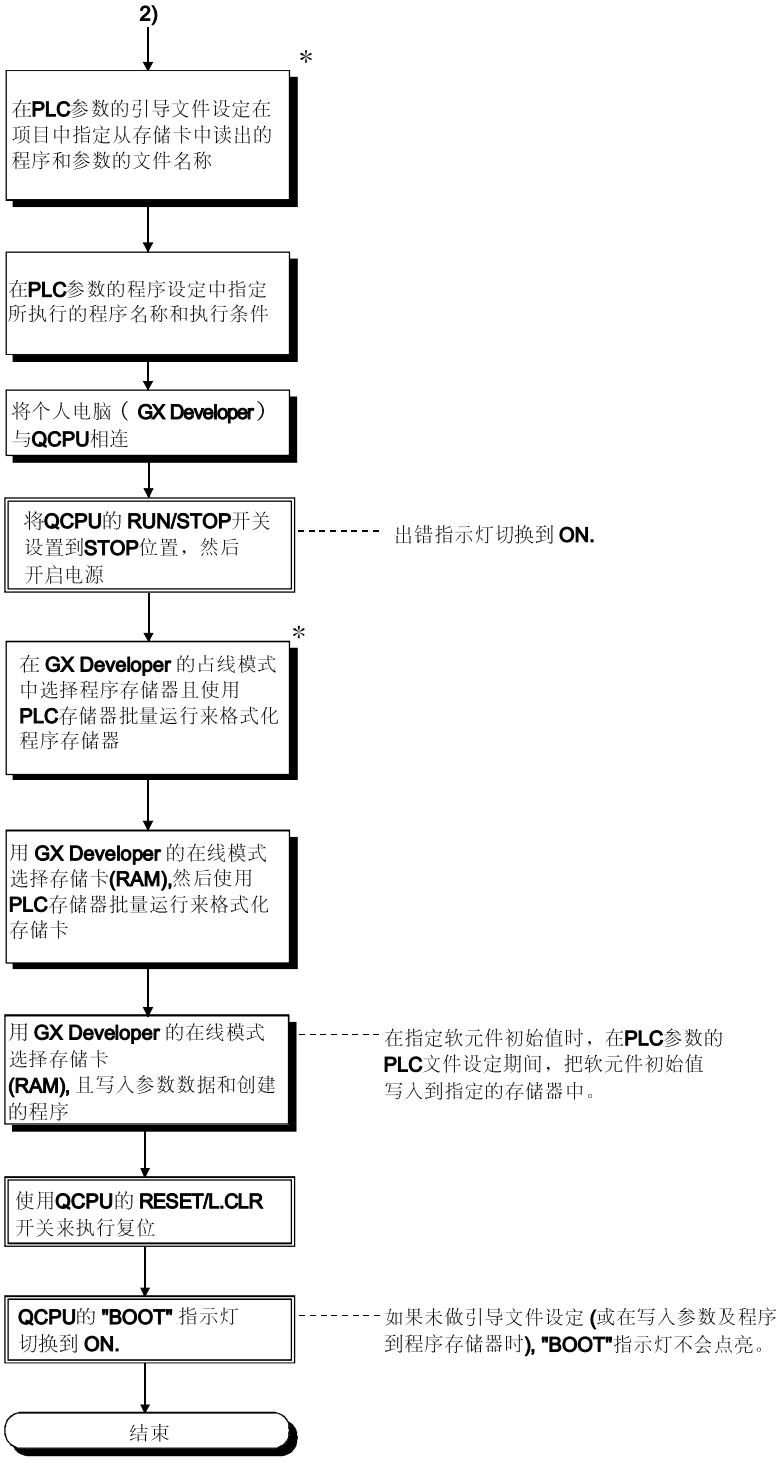
将用 GX Developer 创建的程序和参数写入到安装在 QCUP 存储卡接口中的存储卡中的步骤如下所示。
为了将程序和参数写入到 QCUP 的存储卡中，必须安装存储卡，必须用 QCPU DIP 开关（SW2，SW3）来指定有效的参数驱动设定，并且用 GX-Developer 指定 PLC 参数的引导设定。
有关 QCPU DIP 开关的细节，参考 QCPU（Q 模式）用户手册（硬件设计及维护和检验）

当写入程序和参数到 QCPU 的程序存储器中时，有(*)号的步骤不需要执行。

在□框中的步骤是用 GX Developer 执行的，而在▭框中的步骤则是在 QCPU 中执行的。




1)
使用软元件初始值?
否
是
参考 10.13.2.节
右击软元件存储器并且输入软元件初始值的数据
从 "软元件存储器注册设定" 中选择"使用软元件存储器" 且指定所用的软元件
右击软元件初始值且选择 "Add" 来指定软元件初始值的范围
在PLC参数的PLC文件设定中指定为指定软元件初始值使用的文件名称
指定本地软元件?
否
是
参考 10.13.1.节
在PLC参数的软元件设定项目中指定本地软元件范围的名称
在PLC参数的软元件设定项目中指定本地软元件文件的名称
使用公共指针?
否
是
参考 10.9.2.节
在PLC参数的PLC系统设定项目中指定第一个指针的编号
2)


2)
*
在PLC参数的引导文件设定在项目中指定从存储卡中读出的程序和参数的文件名称
在PLC参数的程序设定中指定所执行的程序名称和执行条件
将个人电脑（ GX Developer）与QCPU相连
将QCPU的 RUN/STOP开关设置到STOP位置，然后开启电源
出错指示灯切换到 ON.
*
在 GX Developer 的占线模式中选择程序存储器且使用PLC存储器批量运行来格式化程序存储器
用 GX Developer 的在线模式选择存储卡(RAM),然后使用PLC存储器批量运行来格式化存储卡
用 GX Developer 的在线模式选择存储卡(RAM), 且写入参数数据和创建的程序
在指定软元件初始值时，在PLC参数的PLC文件设定期间，把软元件初始值写入到指定的存储器中。
使用QCPU的 RESET/L.CLR 开关来执行复位
QCPU的 "BOOT" 指示灯切换到 ON.
如果未做引导文件设定 (或在写入参数及程序到程序存储器时), "BOOT"指示灯不会点亮。
结束

13 多个 PLC 系统的简介

13.1 特点

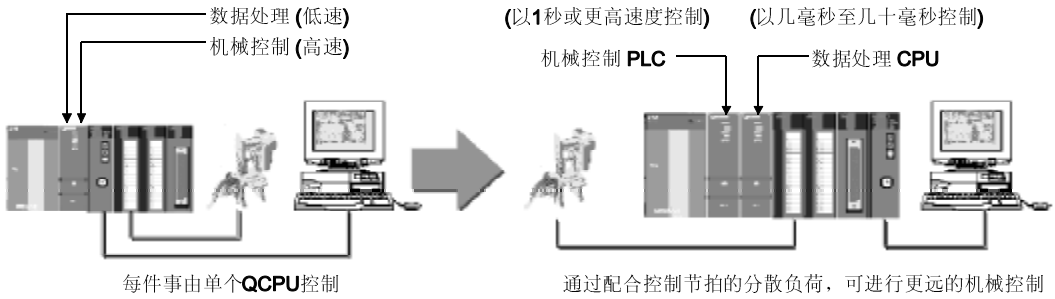
(1) 多元控制

(a) 代替在单个 QCPU 上配置的完整系统，为了在合适的扩展环境中使用每一个 QCPU，可以用多 CPU 系统来增加系统的扩展效率和可维护性。

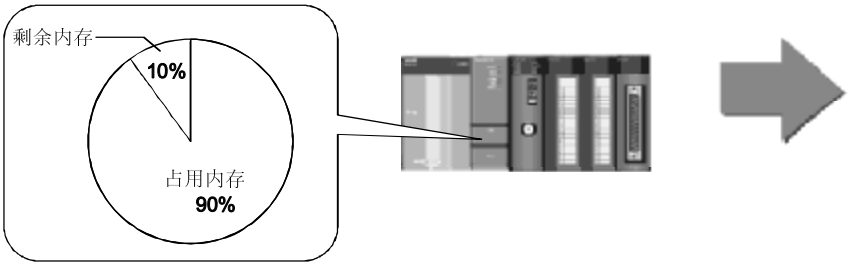
(b) 可以通过每块 QCPU 插入各自的插槽中的办法来控制主基板和附加基板的 I/O 模块和智能型功能模块。
GX-Developer 将把每块 QCPU 所控制的 I/O 模块和智能型功能模块组合起来。

(2) 通过分散负荷使之能进行系统配置

(a) 通过分散在单个 QCPU 上执行的高负荷的处理到几个 QCPU 上，就可能减少总的系统扫描时间。



(b) 通过将所用的存储器分散到几个 QCPU 上，可以增加整个系统所使用的存储器数量。



(3) 通过分散功能，使之能进行系统配置

通过分散功能，使生产线 A 的控制和生产线 B 的控制可在不同的 QCPU 中进行，且可以单独调整每一个功能。

13 - 1

(4) 使每个 QCPU 与多 CPU 系统的动作 CPU 之间能进行二路传输。
在每个 QCPU 与多 CPU 系统的动作 CPU 之间，可以二路传输下列数据。

(a) 用 GX-Developer 执行自动刷新的设置，即可容易地实现在各个 QCPU 之间的二路数据传输。

(b) 需要时，可以使用 FROM/S.TO 命令从其它设备中读取数据。

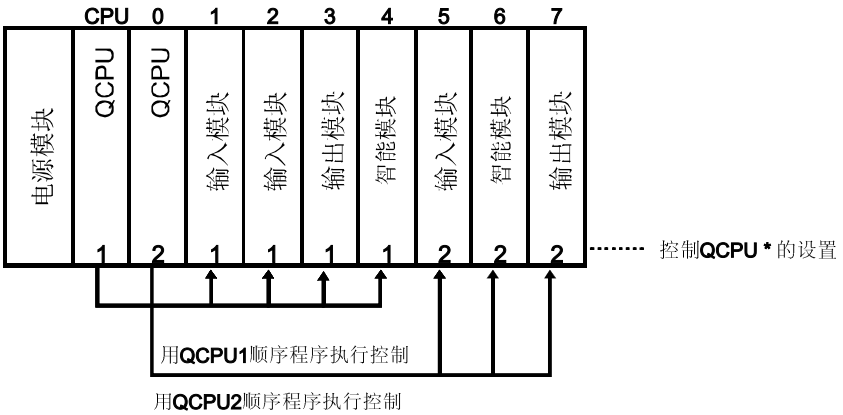
(c) 可以从 QCPU 发出控制指令到动作 CPU，以及在多 CPU 系统内通过专用的动作命令和通信线命令，从 QCPU 中读取动作 CPU 的软元件数据。

13 - 2

13.2 多 CPU 系统简介

(1) 什么是多 CPU 系统

(a)多 CPU 系统是为了控制 I/O 模块和智能型功能模块，在主基板上安装有多块（最多 4 块）QCPU 和动作 CPU 的系统。在多 CPU 系统中，必须设置（控制 PLC 设置）哪一个 QCPU 和动作 CPU 是用来控制哪一些 I/O 模块和智能型功能



模块。

(b) 控制 I/O 模块和智能型功能模块的 QCPU 称为“控制 PLC”。由“控制 PLC”控制的 I/O 模块和智能型功能模块称为“控制模块”。其它不由“控制 PLC”控制的模块称为“非控制模块”。

(2) 多 CPU 系统的建立

为了控制多 CPU 系统（见第 9 章），用 PC 参数在所有安装在主基板上的 QCPU 中设置“安装的 QCPU 数目”和“控制 PLC”是必要的。

(3) 多 CPU 系统的存取范围

(a) 多 CPU 系统的控制 CPU 可以与一个独立的 CPU 系统一样在控制模块上执行 I/O 刷新步骤，写入到智能型功能模块的缓冲存储器中去。

备注

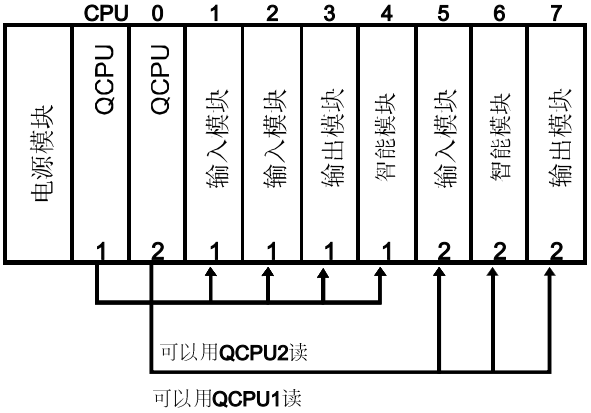
*: 显示用 GX-Developer 编组的配置。
QCPU1 在 I/O 模块上显示“PLC 编号”和“1”，而智能型功能模块显示出它们的“控制 PLC”是 PLC 1 号。

(b) 用以下方法，可以访问非控制模块。

- 刷新对于 I/O 模块和智能型功能模块的输入（PC 参数的多 CPU 设置是必要的）。
- 读智能型功能模块的缓冲存储器。
- 下载输出模块，I/O 组合模块，和智能型功能模块的输出数据。

然而，用以下方法不能访问非控制模块。

- 输出数据到输出模块和智能型功能模块中。
- 写入数据到智能型功能模块的缓冲存储器中。



(4) DX-Developer 的访问范围

(a) 可以在连接到个人电脑的 QCPU 中写参数程序，执行监控和测试。
要访问未连接到个人电脑的 QCPU 时，用 GX-Developer 来指定要访问的 QCPU（连接目标指定）。

(b) GX-Developer 可以访问 QCPU，而不必考虑控制模块和非控制模块。
通过将单个 QCPU 连接到个人电脑，可以用与独立的 CPU 相同的方法来执行对多 CPU 系统的 QCPU 所控制的所有模块的监控和测试。
可以访问同一 MELSECNET/H、以太网或其它网络工作站的其它站的 QCPU。

(c) 通过与同一网络中的其它工作站相连的 GX-Developer，可以访问多 CPU 系统中所有的 QCPU。

13.3 与单个 CPU 系统的区别

单个 CPU 系统与多 CPU 系统的区别如下：

(1) 功能版本（参见 14.2.1 到 14.2.5 节）

(a) 多 CPU 系统支持功能版本 B 的 QCPU。
功能版本 A 的 QCPU 不能用于多 CPU 系统。

(b) 所有 I/O 模块都能用于多 CPU 系统。

(c) 在多 CPU 系统中，使用功能版本 B 的智能型功能模块。
如果 CPU N0 1 设置为控制 CPU，就可以使用功能版 A 的智能型功能模块。

(2) QCPU/动作 CPU 的安装位置（见 14.2.1 节）

最多可以安装 4 个 QCPU，依次从 CPU 插槽(电源模块的右边)到 2 号插槽。几个动作 CPU 一起安装在 QCPU 右面插槽中。

(3) 多 CPU 系统的参数（见 14.2.6 节）

与独立 CPU 系统相比，多 CPU 系统有更多的 PLC 参数项目。加入到多 CPU 系统的 PC 参数中，必须设置的参数如下：

- CPU 数目 ：设置使用的 QCPU 和动作 CPU 的数目。
- 控制 PLC 设定 ：设置哪一个 QCPU 和哪些动作 CPU 控制哪些模块。

(4) 一致性检查

一项设定的存在即表示：所使用的 QCPU 和动作 CPU，在 CPU 的数目、控制 CPU 设定和多 CPU 系统参数上都是相同的。
QCPU 和动作 CPU 要进行一项检查（一致性检查）来确定，当可编程控制器电源设置为 ON，QCPU 复位，和从 STOP 状态切换到 RUN 状态时，多 CPU 系统的参数是相同的。
如果在一致性检查期间发生错误，多 CPU 系统即不能启动。

(5) 有关 I/O 地址的概念（见 15.1 节）

“OH”I/O 地址在多 CPU 系统中是在所装的 QCPU 和动作 CPU 的左面。由于这个原因，“OH”编号根据所使用的 QCPU 和动作 CPU 的编号而不同。

(6) 与非控制 CPU 交互传输（见 17 章）

(a) 可以用与独立 CPU 系统中采用的相同的方法控制受主机控制的 I/O 模块和智能型功能模块。

(b) 不能输出 ON/OFF 数据到不受主机控制的模块和写入到智能型功能模块的缓冲存储器中。
可以通过 PLC 参数设定，从非控制模块中读出 I/O 数据。
可以确认其受它设备控制的模块状态和其它设备的控制状态并控制主机。

(7) 在多 CPU 系统中，每个 QCPU 和动作 CPU 之间的交互传输（见 16 章）

在多 CPU 系统中,每个 QCPU 与动作 CPU 之间可以进行下列的交互传输。

- 用多 CPU 系统的参数设定，在每个 QCPU 与动作 CPU 之间自动刷新软元件数据。
- 在 QCPU 与动作 CPU 之间，用特殊的多 CPU 命令进行数据交换。
- 用特殊动作命令从 QCPU 向动作 CPU 发控制指令。
- 用特殊的多 CPU 交互通信命令，从 QCPU 中读出软元件数据和写入到动作 CPU 中去。

(8) 在复位和出错期间的处理（见 14.2.7 和 14.2.8 节）

对多 CPU 系统的 1 号机和 2 号到 4 号机，发生复位和出错时所执行的处理是不相同的。

(a) 只有多 CPU 系统的 CPU 1 号可以复位。CPU 2 号到 CPU 4 号以及动作 CPU 不能复位。

(b) 当 CPU 1 号发生 STOP 错误时，多 CPU 系统的操作将暂停。
当 CPU 2 号到 PLC 4 号以及动作 CPU 发生 STOP 错误时，可以选择暂停或继续运行多 CPU 系统。

(9) 时钟功能

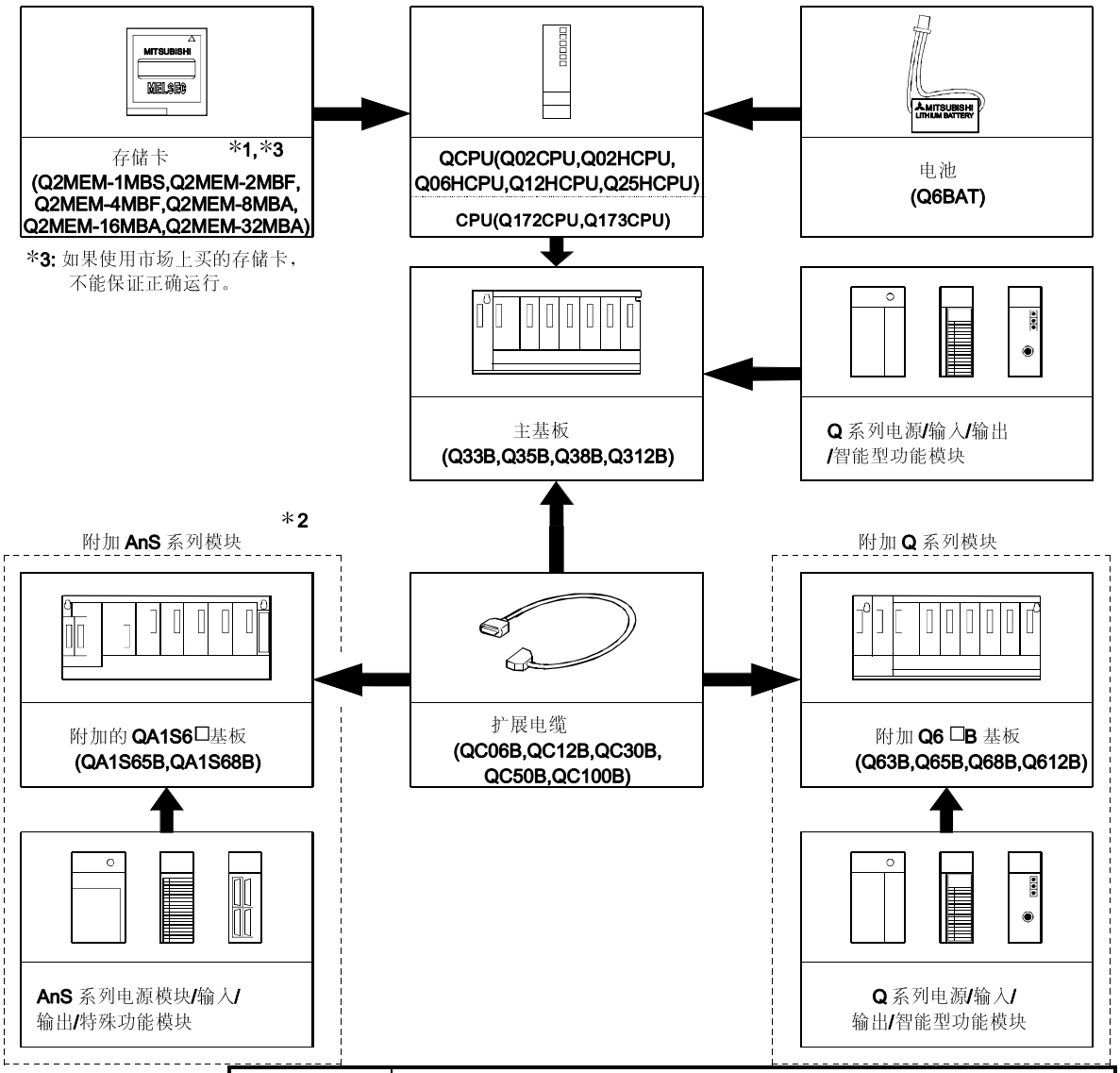
配备有出错时，出错代码和发生时间储存到缓冲存储器中的智能型功能模块（时间数据从 QCPU 中读出）。
不管涉及的模块是控制 CPU 还是非控制 CPU，CPU 1 号的时间数据将作为出错时间而被存储起来。

14 多 CPU 系统的系统配置

本章叙述多 CPU 系统的系统配置及在多 CPU 系统配置期间要遵守的注意事项。

14.1 系统配置

本节对多 CPU 系统的设备配置、与外围设备的连接、及系统配置的输出作简描述。



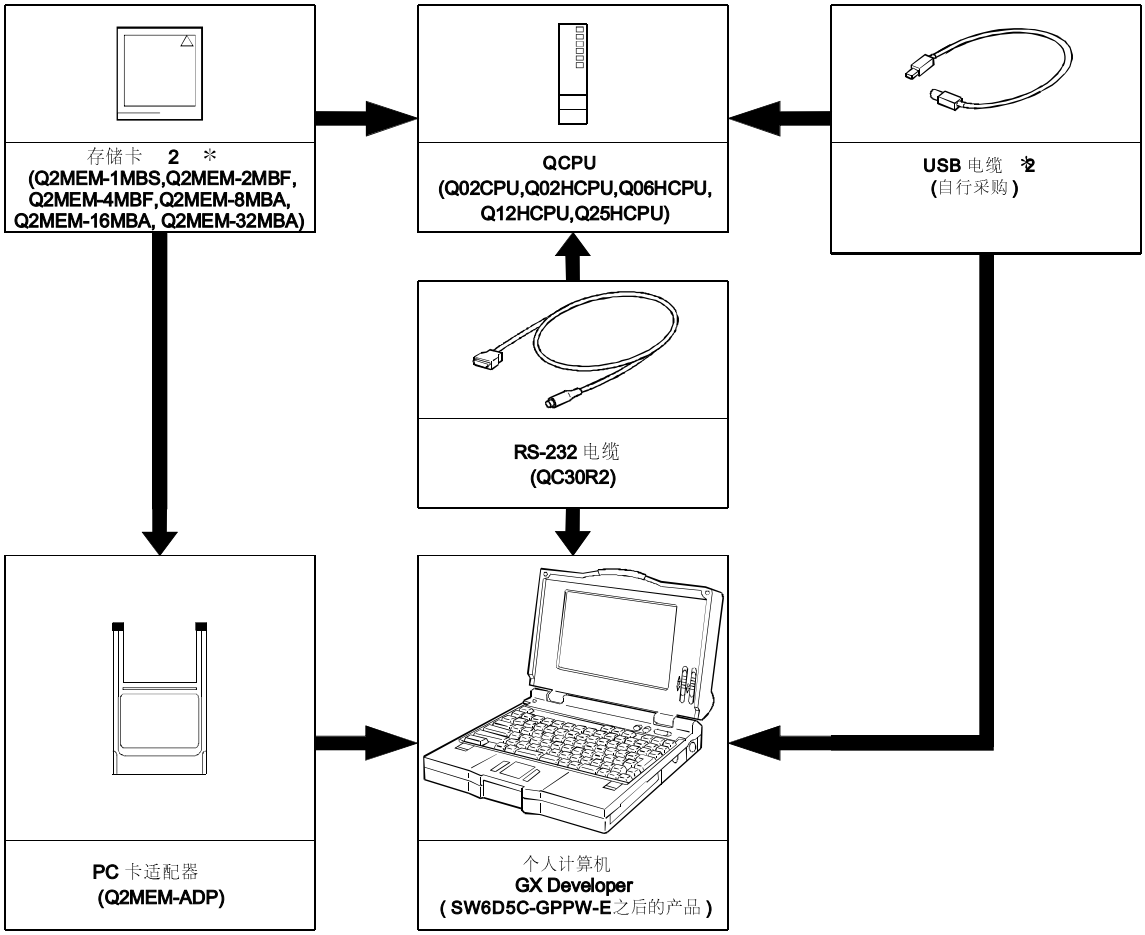
要点

*1. 安装存储卡的数量是 1。
按照用途和容量从 SRAM，内存或 ATA 中选择存储卡。

*2. 附加的 QA1S65B 和 QA1S68B 基板是用于 AnS 系列电源模块、I/O 模块、和特殊功能模块的。

14

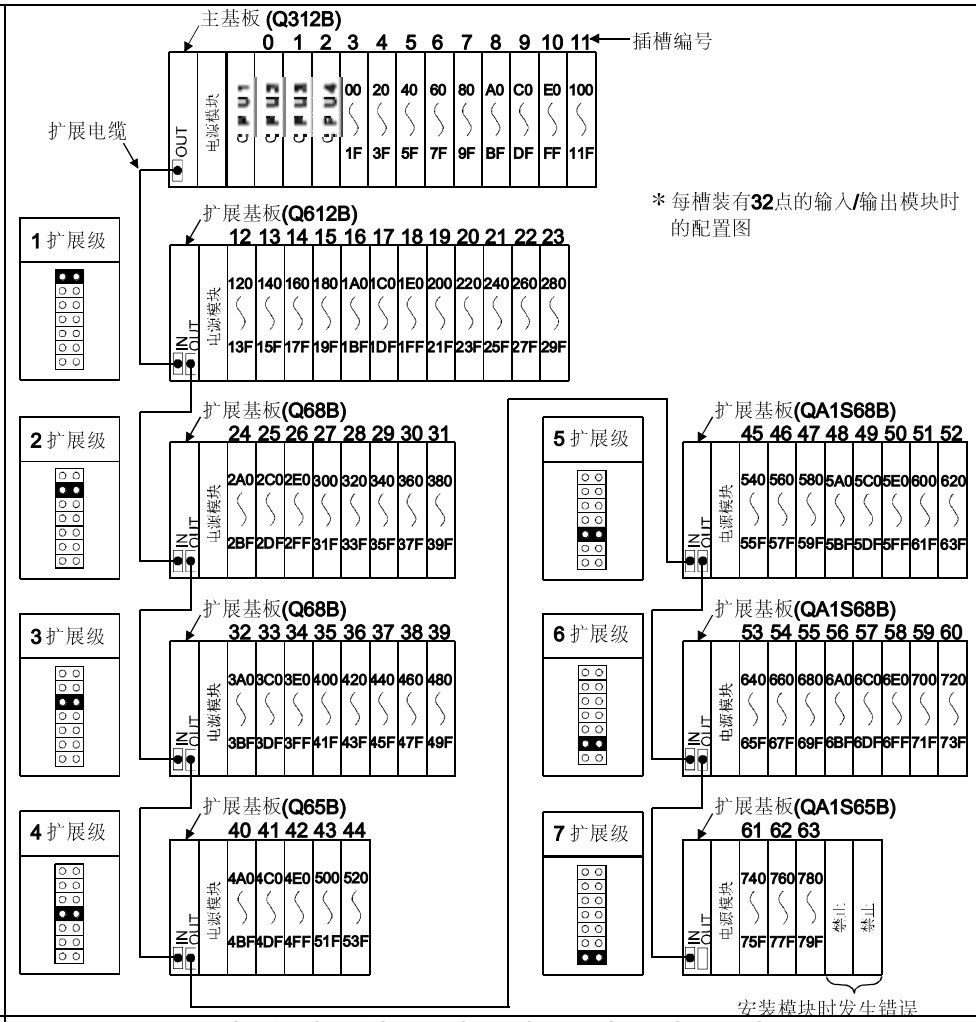
(2) QCPU 使用的外围设备的配置



*1. 有关写入存储卡的方法和 USB 电缆的细节参考 GX Developer 操作手册。

要点
有关连接动作 CPU 与外围设备的细节，参考动作控制器手册。

14 - 2

<table>
<tr><td>系统配置</td><td colspan="5">
</td></tr>
<tr><td>CPU 数目</td><td colspan="5">CPU1:PLC No 1,CPU2:PLC No 2,CPU3:PLC No 3,CPU4:PLC No 4</td></tr>
<tr><td>扩展级的
的最大数目</td><td colspan="5">7 级</td></tr>
<tr><td rowspan="2">安装输入/输出模块的最大数目</td><td>多 CPU 设定的模块数设定</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>安装模块的数目</td><td>64 模块</td><td>63 模块</td><td>62 模块</td><td>61 模块</td></tr>
<tr><td>输入/输出点的最大数目</td><td colspan="5">4096</td></tr>
<tr><td>主基板</td><td colspan="5">Q33B，Q35B，Q38B，Q312B</td></tr>
<tr><td>扩展基板</td><td colspan="5">Q63B，Q65B，Q68B，Q612B，QA1S65B，QA1S68B</td></tr>
<tr><td>扩展电缆</td><td colspan="5">QC06B，QC12B，QC30B，QC508，QC100B</td></tr>
<tr><td>注意</td><td colspan="5">（1） 最多可安装 7 块扩展基板。
（2） 扩展电缆不要长于 13.2m(43.28 英尺)。
（3） 使用扩展电缆时，不应连接到主电路或靠近主电路(高压，大电流)。
（4） 设定延扩展级数时，按升序编号，以避免重复设定相同的编号。
（5） 如果 Q6□B 和 QA1S6□B 安装在同一块扩展基板上，必须先接 Q6□B 然后再接 QA1S6□B，进行扩展基板的等级数设定时，从 Q6□B 开始，按降序排列。
（6） 在一块扩展基板的 OUT 接头与另一块扩展基板的 IN 接头之间连接一条扩展电缆。
（7） 安装超过 66 块扩展基板，可能发生出错。
（8） 安装动作 CPU 时，参考 14.2.1 节。
（9） QCPU 或动作 CPU 有二或三个 CPU 时的 I/O 地址参考 15.1.1 节的 I/O 地址。</td></tr>
</table>

14.2 在多 CPU 系统配置期间的注意事项

14.2.1 可使用的 QCPU 和动作 CPU 的功能版本和它们的安装位置

(1) 功能版本

(a) 可使用的功能版本和确认的方法

在配置多 CPU 系统时，使功能版本“B”的 QCPU 和动作 CPU。QCPU 和动作 CPU 的功能版本可用下列方法确认。

- QCPU 和动作 CPU 上的额定值铭牌。
- GX-Developer 系统监视产品信息表
  （详见 2.3 节）

(b) 不同功能版本的运行

功能版本 A QCPU 不能在多 CPU 系统中使用。

如果功能版本 A 的 QCPU 使用在多 CPU 系统中，如表 14.1 中所示，就会出错，多 CPU 系统将不能启动。

如果 GX-Developer 的（SW6D5C-GPPW-E 或更高版本）PLC 诊断功能显示表 14.1 中的任何一种出错，可用功能版本 B QCPU 替换功能版本 A 的 QCPU。

表 14.1 不同功能版本的运行表

PLC No.1	PLC No.2 到 4	PLC No.1 的状态	PLC No.2 到 4 的状态
功能版本 A	功能版本 A	UNIT VERIFY ERROR （出错代码：2000）	SP. UNIT LAY ERROR （出错代码：2125）
功能版本 A	功能版本 B	UNIT VERIFY ERROR （出错代码：2000）	MULTI EXE。ERROR （出错代码：7010）
功能版本 B	功能版本 A	MULTI EXE。ERROR （出错代码：7010）	SP.UNIT LAY ERROR （出错代码：2125）
功能版本 B	功能版本 B	NO ERROR	NO ERROR

*：在多 CPU 系统中安装了功能版本 A QCPU，不能保证系统的正常运行。

某些情况下，在已经安装了功能版本 B QCPU 的设备上，当 PLC 电源接通时，将发生下列的错误而不是 MULT1 EXE ERROR。这取决于 QCPU 复位时间。

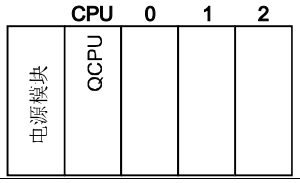
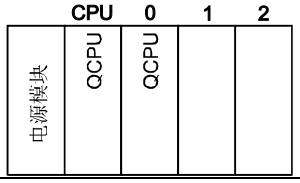
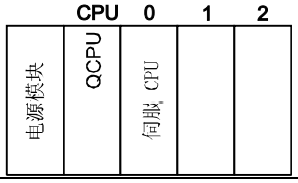
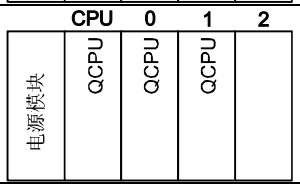
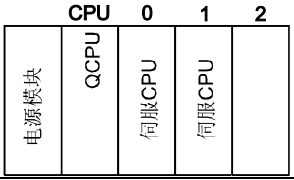
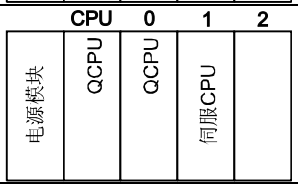
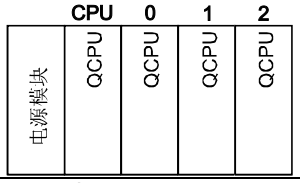
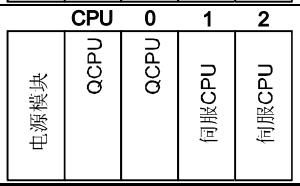
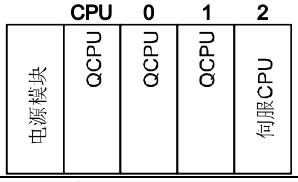
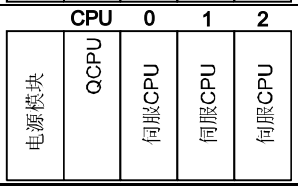
- 控制总线错误（出错代码：1413，1414）
- 多 PLC 宕机(出错代码：7000，7002)

14 - 4

(2) QCPU 和动作 CPU 安装位置

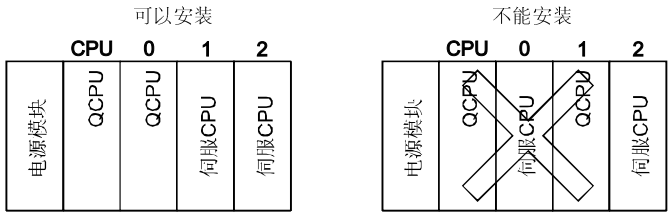
(a) 主基板最多可安装四个 QCPU 依次从电源模块右边的插槽到 2 号槽。在 QCPU 与 QCPU 之间，QCPU 与动作 CPU 之间，动作 CPU 与动作 CPU 之间不能留有空档。

表 14.2 QCPU 和动作 CPU 安装位置

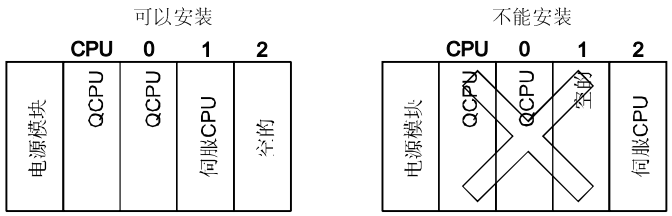
CPU 的数目	QCPU 和动作 CPU 的安装位置	
1		—
2		
3	 	 —
4	 	 

14 - 5

(b) 几个动作 CPU 一起安装在 QCPU 右边的插槽中。而 QCPU 则不可以安装在动作 CPU 的右边。

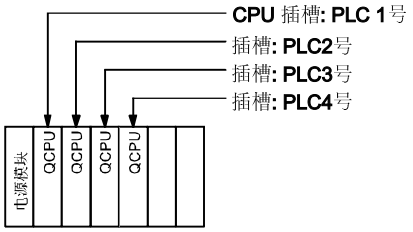


(c) 在进行惯例扩展时，不能与原安装的 QCPU 和动作 CPU 之间留下空槽。
CPU（包括空槽）的数目在 CPU 计数设定时进行设置，类型则是在“CPU（空）”设定时从用 PC 参数的 I/O 分配设置的 CPU 的数的右边紧挨着的位置开始设置。
例如：在多 CPU 设定中已经设置了 4 块 CPU 并安装了 2 块 QCPU 和一块动作 CPU，则 QCPU 安装在 CPU 插槽和 0 号插槽中，动作 CPU 安装在 1 号插槽中而插槽 2 让其空着。



(2) QCPU 和动作 CPU 安装位置

(a) 设备编号是用来确认在多 CPU 系统中，安装在基本基板上的 QCPU 和动作 CPU。PLC1 号是分配给 CPU 插槽，而 PLC2 号，3 号，4 号是分配给 PLC1 号右边的插槽。



这些设备编号在多 CPU 系统中用于以下的用途：

- 在未连接 GX-Developer（个人电脑）时，访问 QCPU。
- 用 I/O 分配来设置控制 PLC。

14 - 6

(b) 主机的编号是储存在多 CPU 系统的特殊寄存器（SD395）中的。建议在多 CPU 系统中创建一个主机编号确认程序。创建主机编号确认程序，如果 QCPU 安装不正确或者用 GX-Developer 将程序写入到其它设备中时，就容易验证。
在下列程序中，当 QCPU 写程序是编号“1”（SD395=1）的 PLC 以外的设备时，信号报警器（F1）将设置为 ON。
当信号报警器设置为 ON 时，在 QCPU 面板上的“USER”指示灯将点亮。已经被设置为“ON”的信号报警器的编号将存储在特殊寄存器（SD62）中。

14 - 7

14.2.2 使用 Q 系列相应的 I/O 模块和智能型功能模块时的注意事项

(1) 可使用的 I/O 模块

所有的 I/O 模块（QX□，QY□）都与多 CPU 系统相对应。可以将 PLC1 到 4 设置为控制 PLC。

(2) 使用智能型功能模块

（a） 在多 CPU 系统中使用功能版本 B 的智能型功能模块。可以用功能版本 B 的智能型功能模块来设置 PLC1 到 PLC4 为控制 PLC。

（b） 多 CPU 系统可支持“A”和“A”以后的功能版本而 Q 系列对应的高速计数模块。（QD62、QP62D、QD62E）支持这些 Q 系列模块，CPU1 到 4 可设置为“控制“PLC”。

（c） Q 系列对应的中断模块（Q160）是没有功能版本的，但是它受到多 PLC 系统的支持。PLC1 到 PLC 4 可设置为“控制 PLC”。

（d） 如果 PLC1 号设置为控制 PLC，功能版本 A 的智能型功能模块不同于上面（b）和（c）提到的那些模块，可以在多 CPU 系统中使用。然而，只有控制 PLC 可从串行通信模块和其它外部模块来进行访问（MELSECNET/H，串行通信模块和其它外部模块不能访问非控制模块）。
如果 PLC2 到 4 设置为控制 PLC，将会发生“SP.UNIT.VER.ERR”（出错代码：2150）出错。多 CPU 系统将不能启动。

(3) 控制 PLC 和非控制 PLC 的访问范围

多 CPU 系统和 DII 非控制 PLC 对控制 PLC 的访问范围如下所示：

访问目标		控制 PLC	组合以外的 I/O 设定	
			不可能	可能
输入（X）		○	×	○
输出（Y）		○	×	○
缓冲存储器	读	○	○	○
	写	○	×	×

备注

- 可以通过智能型功能模块上的铭牌和 GX-Developer 的系统监视产品信息表来确认智能型功能模块的功能版本。（见 2.3 节）
- 有关智能型功能模块可使用的数量限制细节，参考 14.2.4 节。

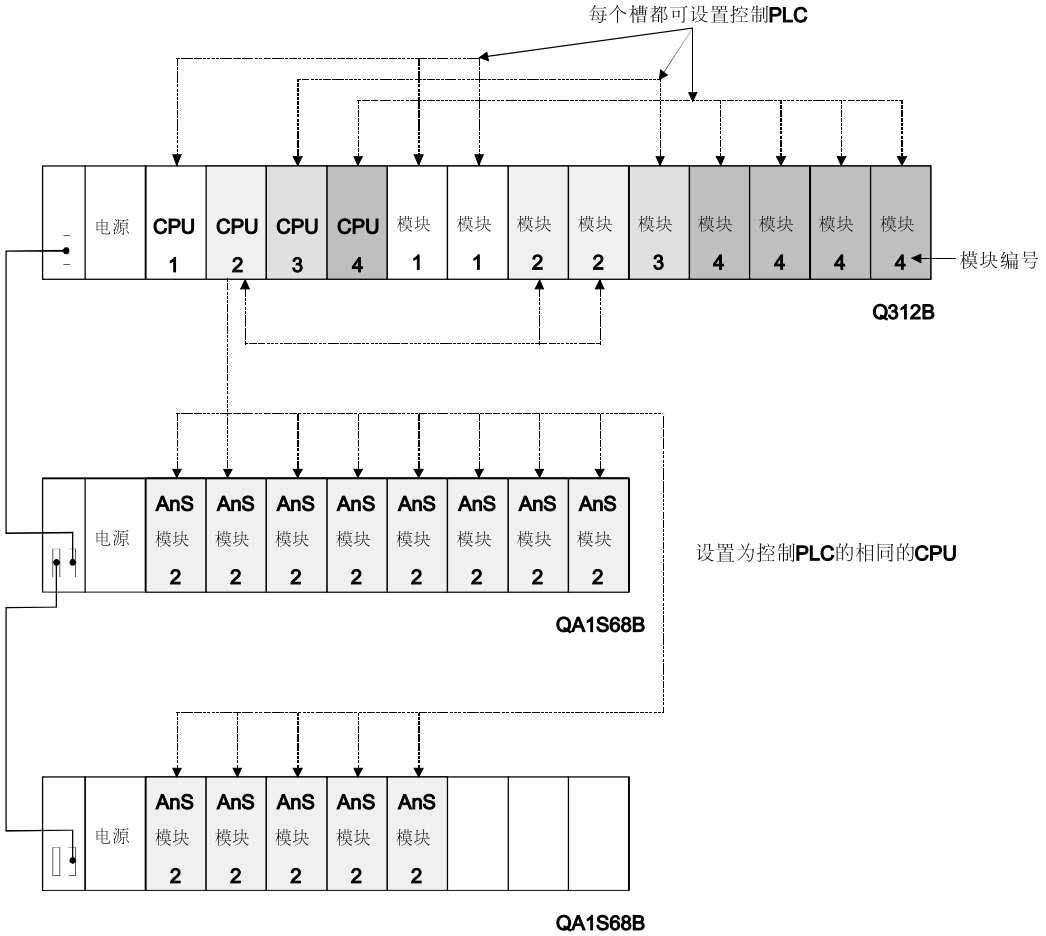
14.2.3 安装 AnS 系列对应的 I/O 模块和特殊功能模块时的限制

(1) 使用 I/O 模块和特殊功能模块

AnS 系列对应的 I/O 模块和特殊功能模块（小型）可与 QCPU 一起使用。

(2) 控制 PLC

在配置多 CPU 系统时，在 PLC1 号和 PLC 4 号之间只能控制（控制 PLC）一个 AnS 系列对应的 I/O 模块和特殊功能模块。例如：如果控制 PLC 设置为 PLC2，如下图所示，然后每个安装有支持 AnS 系列的 I/O 模块和特殊功能模块的插槽中的控制 PLC 会被设置为 PLC2。如果一个以上的 AnS 系列对应的 I/O 模块和特殊功能模块的被设置，则将会发生 “参数错误”（出错代码 3009），多 CPU 系统不会启动。



14 - 9

(3) 控制 PLC 和非控制 PLC 的访问范围

多 CPU 系统的控制 PLC 和非控制 PLC 的访问范围如下图所示：

访问目标		控制 PLC	组合以外的 I/O 设定	
			不可能	可能
输入（X）		○	×	×
输出（Y）		○	×	×
缓冲存储器	读	○	×	×
	写	○	×	×

14 - 10

14.2.4 有安装限制的模块

多 CPU 系统中可安装的模块数目的限制如下表所示。必须保证所安装的模块数目在这个范围之内。

产品	型号	每个系统可安装的模块数目限制	每个 CPU 可安装的模块数目限制
Q 系列 MELSECNET/H 网络模块	• QJ71LP21 • QJ71BR11 • QJ71LP21-25	最多可将四个 PLC 连到 PLC 网络和远程 I/O 网络	最多可将四个 PLC 连到 PLC 网络和远程 I/O 网络
Q 系列以太网接口模块	• QJ71E71 • QJ71E71-B2	最多四个	最多四个
Q 系列 CC-Link 系统主站/本地模块	• QJ61BT11	最多十六个	最多四个
MELSECNET/MINI-S3 数据通讯模块	• A1SJ71PT32-S3 • A1SJ71T32-S3	没有限制（然而，不能建立自动刷新功能）	没有限制（然而，不能建立自动刷新功能）
AnS 系列对应的特殊功能模块	• A1SD51S • A1SD21-S2 • A1SJ71J92-S3 （当使用 GET 和 PUT 服务时）	最多六个	最多六个
中断模块	• AIS161	只能一个	只能一个
	• Q160	最多四个（当使用 AIS161 时最多三个）	

* 即使基板上安装了多于上述限制的模块时（CPU1：4，系统 1：16）QJ61BT11 也不会发生出错。
然而，GX-Developer 不能为超过了限制模块数目的 QJ61BT11 建立 CC-Link，也不能执行与远程站的交互通信等的功能。

14.2.5 可使用的 GX Developer 和 GX-配置器

(1) 可使用的 GX Developer

GX-Developer（SW6D5C-GPPW-E 或更高版本）可以在多 CPU 系统中使用。
SW5D5C-GPPW-E 和更早期的 GX-Developer 不能使用。

(2) 可使用的 GX-Configurators

下表所列的 GX- Configurators 可在多 CPU 系统中使用而不必修改。然而，必须保证与 GX-Developer（SW6D5C-GPPW-E 或更高版本）结合使用。

产器型号	应用
SW0D5C—QADU-E	模拟输入
SW0D5C—QDAU-E	模拟输出
SW0D5C—QSCU-E	串行通信
SW0D5C—QCTU-E	高速计数

14 - 12

14.2.6 使多 CPU 系统生效的参数

(1) 使多 CPU 系统生效的参数

与独立 CPU 系统相比较，多 CPU 系统必须有“CPU 计数”“控制 CPU”和“刷新设定”（自动刷新设定）参数设置。

设置在多 CPU 系统中使用的所有参数“CPU 计数”“控制 PLC”和“刷新设定”（自动刷新设定）必须写入到 QCPU 和动作 CPU 中去。

(2) 在多 CPU 系统中使用的 PC 参数设定

PC 参数，设置的必要性和使用多 CPU 系统所要求的说明列于表 14.3 中。

表 14.3 多 CPU 和 I/O 分配的设置表

PC 参数			设置的必要性 *1	说明*2
I/O 分配	I/O 分配			
		类型	—	○
		型号	—	—
		点数	—	○
		第一 X/Y	—	○
	基板设定			
		基板型号	—	—
		电源模块型号	—	—
		扩展电缆型号	—	—
		插槽数目	—	○
	开关设定		—	—
	细节设定			
		出错时的输出模式	—	—
		H/W 出错时 CPU 运行模式	—	—
		I/O 响应时间	—	—
		控制 PLC	○	○
PLC 系统设定	空插槽的数目		—	○
多 CPU 设定	CPU 计数		○	○
	运行模式		△	○
	组合以外的输入设定		△	△
	组合以外的输出设定		△	△
	刷新设定			
		每一个 CPU 的传输范围	△	○
		CPU 软元件	△	—

*1：设置栏的必要性

○：多 CPU 系统中必须设置的项目（如果不设置则不能运行）

△：当多 CPU 系统需要时可以设置的项目（不设置时，依照缺省值运行）

—：与独立 CPU 系统相同的项目

*2：说明

○：多 CPU 系统中对所有的 QCPU 和动作 CPU 都具有相同设定的项目。

△：多 CPU 系统中对所有的 QCPU 都具有相同的设定项目（对动作 CPU 没有设定的项目）。

—：多 CPU 系统中可以对每个 QCPU 和动作 CPU 独立设置的项目。

14 - 13

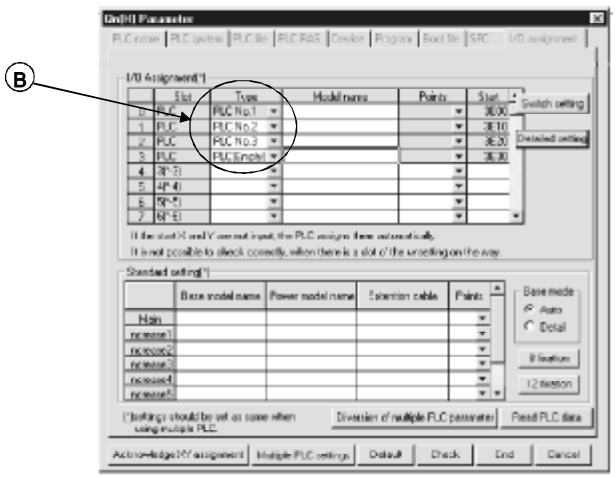
如果在表 14.3 说明列中的“O”需要修正，必须保证多 CPU 系统中的所有 QCPU 和动作 CPU 都是同样的设定。
可以进行跨越传送和使用 GX-Developer 为其它项目设置的 CPU 设定和 I/O 分配。（有关传送和使用多 CPU 设定和 I/O 分配的细节，参考 19.2.3 节）

a） CPU 计数设定（必须设置）

1）多 CPU 系统中所使用的 QCPU 数目是根据 PC 参数的多 CPU 设定（箭头“A”指出的）来选择的。



2）必须保证多 CPU 系统的 CPU 计数设置与实际安装的 CPU 数目相同。
当为了将来安装附加 QCPU 的目的而留出一个空闲插槽时，应该用参数的 I/O 分配设定设置“CPU（空闲）”。
例如：如果在多 CPU 设定中设置了四个 CPU 和一个为将来使用而留出的 CPU 空槽，那么应该在插槽 2 号中（图中箭头 B 所指）设置“CPU（空闲）”。



3）在下列情况下，所有安装的 CPU 将发生出错“DARAMETER ERROR”（出错代码 3010）。

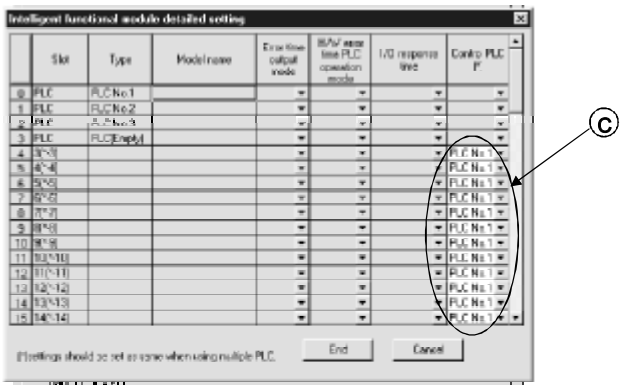
- 所安装的 QCPU 数目超过了 CPU 计数设定中的数目。
- 设置为 CPU1 到 CPU4 的槽中没有安装 QCPU 或动作 CPU。

b） 运行模式的设定(选项)
设置系统使系统在 PLC 2 到 PLC 4 发生 STOP 的错误时,其它设备的运行不会暂停。
PLC 1 的运行模式是不能修正的(当 PLC 1 发生 STOP 错误时，所有的设备都将暂停运行。) 有关细节，参考 14.2.8 节。

c） 组合以外的 I/O 设定(选项)
这个设置是在其它设备控制的 I/O 模块和智能型功能模块的输入(X)和输出(Y)被下载到主机时的设置。
有关细节，参考 17.2 节。

d） 刷新设定(选项)
这个设置是用来在多 CPU 系统中自动刷新软元件数据的。
有关细节，参考 16.1 节。

e） 控制 PLC 设置(必要的设置)
将安装在主基板和附加基板的 I/O 模块和智能型功能模块设置为多 PCL 系统中的控制 PLC(箭头 C 所指)时，所有的默值设定为 PLC 1。



(3) 多 CPU 设置和 I/O 分配检查

如表 14.4 所列出，当表 14.3 的说明这一列都设置为"O"符号时，可编程控制器的电源接通，QCPU 复位或工作状态从停止切换到运行时，将执行检查，确定所有的 QCPU 和动作 CPU 都有相同的设定（一致性检查）。

（a）如果所有的设备都具有相同的设定，多 CPU 系统将启动。

（b）当所有的设备不具有相同设定时，将按照表 14.4 中所说明的方式运行。
在这种情况下，检查多 CPU 设定和 I/O 分配，并设置所有的设备具有相同的设定，

表 14.4 一致性检查内容列表

<table>
<tr><th colspan="2">项目</th><th>CPU 1</th><th>CPU 1-CPU 4</th></tr>
<tr><td colspan="2">当电源接通时
当 PLC 1 复位时</td><td>不进行一致性检查</td><td>• 进行比较检查，检查 PLC 1 的多个 PLC 设定和 I/O 分配。
• 如果结果不相符，主机将发生一个“参数错误”出错(代码 3012)</td></tr>
<tr><td rowspan="3">• 当 STOP /RUN 开关从 STOP 切换到 RUN 时
• 用 Gx-Developer 写入参数时</td><td>当存在处于运行模式的设备时</td><td colspan="2">• 对于处于运行模式的最低编号的设备进行多 CPU 设置和 I/O 分配的比较检查。
• 如果不相符，主机将发生"参数错误"出错（出错代码 3012）。</td></tr>
<tr><td>当不存在处于运行模式的设备时</td><td>• 对 PLC 2 号的多 CPU 设置和 I/O 分配运行比较检查。
• 如果不相符，主机将发生“一个参数错误”出错（出错代码 3012）。</td><td>• 对 PLC 1 号的多 CPU 设定和 I/O 分配运行比较检查。
• 如果不相符，主机将发生“一个参数错误”出错（出错代码 3012）。</td></tr>
<tr><td>PLC 1 发生停止出错时</td><td>–</td><td>因为主机发生“多 CPU 宕机”出错（出错代码 7000）。
不能从 STOP 切换到 RUN。</td></tr>
</table>

要点

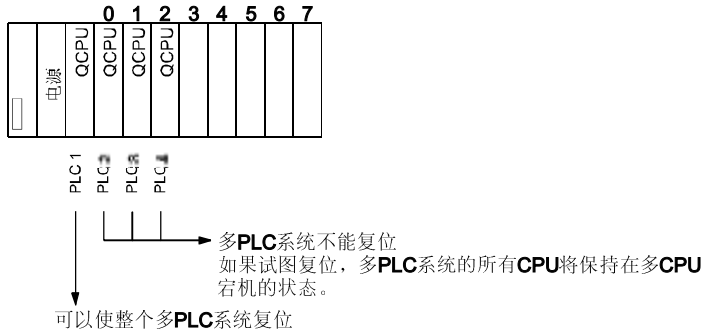
当 QCPU 用于改变多 CPU 参数（包括动作 CPU 设定为没有动作 CPU 的多 CPU 参数）时，QCPU 将对动作 CPU 的多 CPU 系统参数进行一致性检查。在这种情况下，QCPU 可能会发生“参数错误”出错(出错代码:3012)
如果当 QCPU 的多 CPU 系统参数改变时，QCPU 发生 “参数错误”出错；检查每个 QCPU 的多 CPU 系统参数是否相同，然后或将 PLC 1 复位，或者将可编程控制器电源切断，再切换到 ON。

14 - 16

14.2.7 多 CPU 系统的复位

通过将 PLC 1 复位，可以使整个多 CPU 系统复位。
当 PLC 1 复位时，所有的 QCPU，I/O 模块，和智能型功能模块都将复位。

如果多 CPU 系统中的任何一 CPU 发生了停止错误。在问题得到解决以后，或者将 CPU 1 复位，或者重新启动可编程控制器（ON→OFF→ON）。
（不能通过对已经出错的 CPU 2 到 CPU 4 或动作 CPU 进行复位使系统复位）。



要点
（1） 多 CPU 系统中，使 CPU 2 到 CPU 4 或动作 CPU 分别复位是不可能的。如果在多 CPU 系统运行期间试图将 CPU 2 到 CPU 4 或动作 CPU 中的任何一个复位，其它的设备将发生“多 CPU 宕机”出错（出错代码 7000），整个多 CPU 系统将停止运行。然而，根据 QCPU 或动作 CPU 的复位时间，则有可能发生“多 CPU 宕机”以外的错误会使其它设备停止运行。
（2） 不管 PC 参数的多 CPU 设定中所设置的运行模式（停止/继续 CPU 2 到 CPU 4 的所有其它设备），当将 CPU 2 到 CPU 4 复位时，将发生“多 CPU 宕机”出错（出错代码 7000）（参考 14.2.8 节有关多 CPU 设定运行模式的细节）。

14 - 17

14.2.8 当发生 QCPU 停止错误时的处理

当在多 CPU 系统中发生 CPU 1 停机错误时，及当发生 CPU 2 到 CPU 4 停机错误时，整个系统的运行是不同的。

(1) 当在 CPU 1 发生停机错误时

（a） 所有 PLC 2 到 PLC 4 和动作 CPU。当 PLC 1 发生停机出错时，所有的 PLC 2 到 PLC 4 和动作 CPU 将发生出错“多 CPU 宕机”(出错代码 7000)，多 CPU 系统将停止运行。

（b） 遵守下面的步骤来恢复系统

1）用 PC 诊断功能来确认 PLC 1 出错的原因。

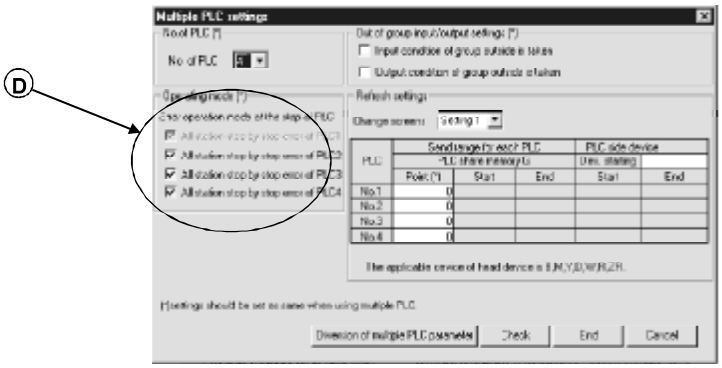
2）消除出错的原因。

3）使 PLC 1 复位或重新启动 PLC 的电源。
当 PLC 1 复位或接通电源重新启动时，整个 PLC 系统的所有设备将复位，系统恢复。

(2) 当 PLC 2 到 PLC 4 中发生停机错误时

当 PLC 2 到 PLC 4 中发生停止出错时，整个系统是否停机将取决于多 CPU 设定的“运行模式”设定。

缺省的设定是：随着停止出错的发生，所有的设备将停止运行。当 QCPU 中发生停机错误而不需要暂停所有设备时，可点击相应字段，使所有的设备不停机。（箭头 D 所指的）。

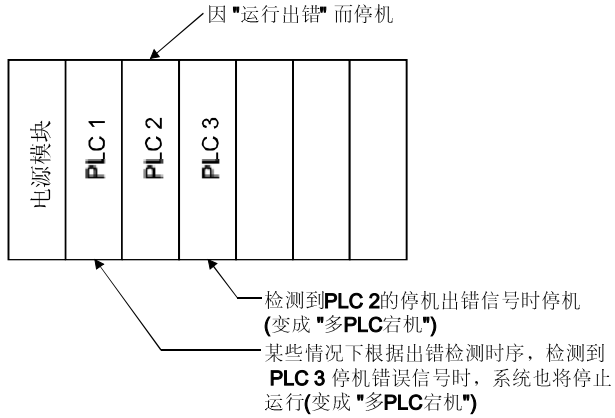


（a） 如果设置“‘n’设备出错时停止所有设备”，那么在 QCPU 中当停机错误发生时，QCPU 将发生“多 CPU 宕机”(出错代码 7000)出错信号，多 CPU 系统将停止运行。

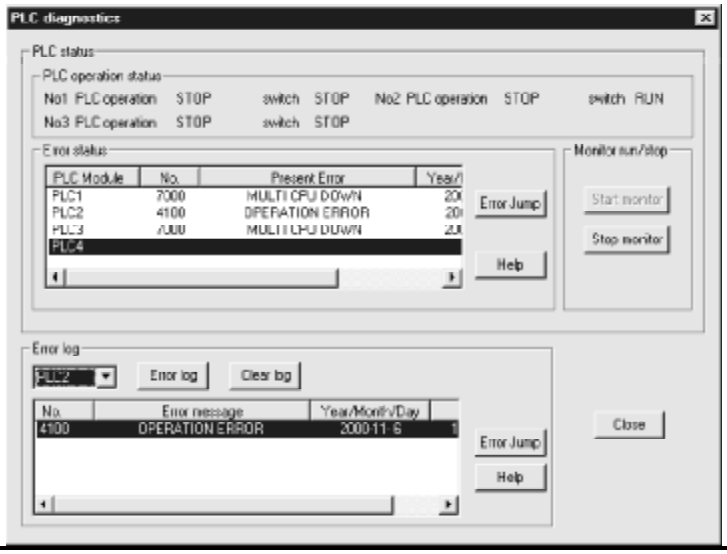
（b） 如果设置“‘n’设备出错时不停止所有设备”，那么当 QCPU 中发生停机错误时，对所有其它设备将发出“多 CPU 出错”出错(出错代码 7010)，但系统运行仍将继续。

要点

* 当停机错误发生时，被检测到出错的设备也发出“多 CPU 宕机”的停机出错信号。
在某些情况下，出错检测的时序将搜寻引起“多 CPU 宕机”出错的故障发生设备。不只是发生停机错误的第一台设备，而是整个系统将保持在“多 CPU 宕机”状态。
例如：如果停机错误发生在 PLC 2，作为直接结果，PLC 3 停机。某些情况下根据出错检测的时序，因为 PLC 3 的停机出错，PLC 1 也将停机。



由于这个原因，某些情况下：对最初引起停机出错的设备，一个不同的设备编号将被存放到出错数据的公共信息类别中，这样，当恢复系统时，除了“多 CPU 宕机”出错以外还消除了引起停机错误的设备的出错原因。
在下图中：引起 PLC 2 出错的原因被消除，这个原因未引起“多 CPU 宕机”出错。



(c) 遵循以下的步骤来恢复系统。
1）用 PC 诊断功能来确认 PLC 1 出错的原因。
2）消除出错原因。
3）将 PLC 1 复位，或重新接通 PLC 电源。
当 PLC 1 复位或重新接通到 PLC 的电源时，整个多 PLC 系统的所有设备将被复位，系统恢复。

14.2.9 减少多 CPU 系统处理所需的时间

(1) 多 CPU 系统处理

当访问 I/O 模块和智能型功能模块时，QCPU 使用总线（基板电路，扩展电缆）。
但这个总线不能同时被其它设备的 QCPU 使用。
当多 CPU 系统的 QCPU 同时使用总线时，准备随后进行总线访问的设备的 QCPU 将停留在“等待”状态，直到执行步骤的 QCPU 首先处理完成。
这个“等待”状态（QCPU 必须等待的时间量）将使多 CPU 系统中的输入输出延迟，造成扫描时间延长。
参考第 18 章有关扫描时间延长的细节。

(2) 最大的等待时间

在下列情况下，多 CPU 系统中，主机将达到最大等待时间。

- 当系统中使用四个 QCPU 时。
- 当使用附加的基板时。
- 当占有大量数据的智能型功能模块安装在附加基板上时。
- 当四个 QCPU 同时访问安装在附加基板上的模块时。

(3) 减少多 CPU 系统处理所需的时间

下列方法可用来减少多 CPU 系统处理所需的时间。

- 将模块与许多访问点结合，如：MELSECNET/H 和 CC-LINK 刷新等一起装在主基板上。
- 设置模块有许多访问点，如：MELSECNET/H 和 CC-LINK 刷新等作为在单独 QCPU 上的控制模块，并确保不会发生同时访问。
- 减少 MELSECNET/H 和 CC-LINK 刷新访问点的数目。
  减少 QCPU 之间自动刷新访问点的数目。

要点
通过修改以下的 PC 参数设定，可以减少扫描时间： • A 系列 CPU 兼容性设定 • 浮点算术运算 详见 18.3 节

15 多 CPU 系统 I/O 地址的分配

15.1 分配 I/O 地址的概念

多 CPU 系统占有的 I/O 地址使 QCPU 与 I/O 模块和智能型功能模块之间能互相传输信息，也使 QCPU 与动作 CPU 之间能相互传输信息。

15.1.1 I/O 模块和智能型功能模块的 I/O 地址

与独立 CPU 系统的区别是多 CPU 系统 I/O 地址的 OH 位置（槽）。然而，根据分配附加基板 I/O 地址的顺序的概念，每一个槽的 I/O 数目和空槽的 I/O 数目在两类系统中是相同的。
参考第 5 章（分配 I/O 地址），了解分配附加基板 I/O 地址的概念，每个槽的 I/O 数目和空槽 I/O 数目的细节。

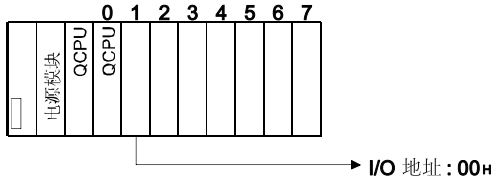
(1) I/O 地址的“OH”位置

(a) PC 参数的多 CPU 设定中设置的槽数目是由多 CPU 系统中 QCPU 和动作 CPU 所占据的。

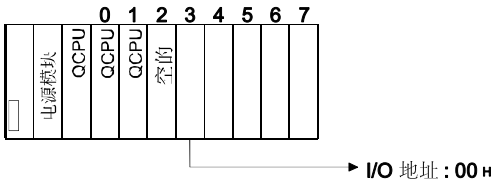
(b) I/O 模块和智能型功能模块从 QCPU 和动作 CPU 所占槽的右边开始安装。

(c) 将从 QCPU 和动作 CPU 所占槽的右边安装的 I/O 模块和智能型功能模块的 I/O 地址设置为“OH”。然后按从左到右的顺序分配连续的编号。

1） QCPU：在安装有二块模块时



2） QCPU：在装有三块模块和留有一个空槽时



备注

- 当主基板上装有比多 CPU 设定中所设置的数目少的 QCPU 和动作 CPU 时，QCPU 安装在 CPU 槽中，且 QCPU 和动作 CPU 从“0”槽开始顺序安装。在 QCPU 和动作 CPU 右边的槽空着。
- 多 CPU 系统的 I/O 地址可以用系统监视器来确认。

15.1.2 QCPU 和动作 CPU 的 I/O 地址

为了在 QCPU 和动作 CPU 之间用以下的命令进行相互通讯，要将 I/O 地址分配给多 CPU 系统的 QCPU 和动作 CPU。

- 多 CPU 的专用命令
- 动作专用命令
- 在多 CPU 之间的专用通讯命令

分配给 QCPU 和动作 CPU 的 I/O 地址是按照它们所安装的槽来固定的，并且不能修改。

多 CPU 系统中安装在主基板上的 QCPU 的 I/O 地址如以下所示：

QCPU 安装位置	CPU 槽	“0”槽	“1”槽	“2”槽
I/O 起始地址	$3E00_H$	$3E10_H$	$3E20_H$	$3E30_H$

在下列情况下要使用 QCPU 和动作 CPU 的 I/O 地址。

- 用 S，TO 命令写入数据到主工作站的公共 CPU 存储器时。*1
- 用 FROM 命令从其它设备的公共 CPU 存储器读取数据时。*1
- 用智能型功能模块软元件（U□□G□），从其它设备的公共 CPU 存储器中读取数据时。*1
- 用以太网模块指定被访问的 QCPU 时。*1*2
- 用串行通讯模块指定被访问的 QCPU 时。*3

15

备注

*1. 参考 16 章有关 QCPU 与动作 CPU 之间通讯的细节。
*2. 参考以太网模块手册有关用以太网模块访问 QCPU 的细节。
*3. 参考系列通讯模块手册有关用串行通讯模块访问 QCPU 的细节。

15.2 用 GX-Developer 进行 PC 参数 I/O 分配的目的

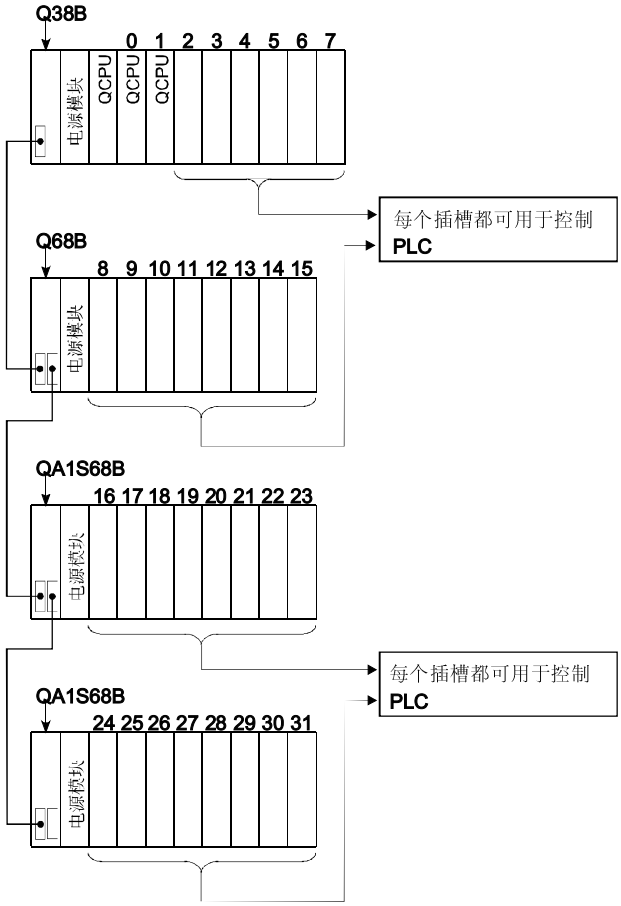
下列情况下：用 GX-Developer 执行 I/O 分配。

(1) 设置控制 PLC

设置控制多 CPU 系统的 I/O 模块和智能型功能模块的 QCPU 和动作 CPU。

(a) 可以选择 Q 系列的 I/O 模块和智能型功能模块作为控制 PLC，安装在各个槽中。

(b) 对相同的 QCPU，可以设置 AnS 系列的 I/O 模块和智能型功能模块作为控制 PLC。



15 - 3

16 多 CPU 系统的 QCPU 与动作 CPU 之间的通信

用多 CPU 专用命令，可在具有自动软元件数据刷新功能的多 CPU 系统的下列模块之间交换数据。

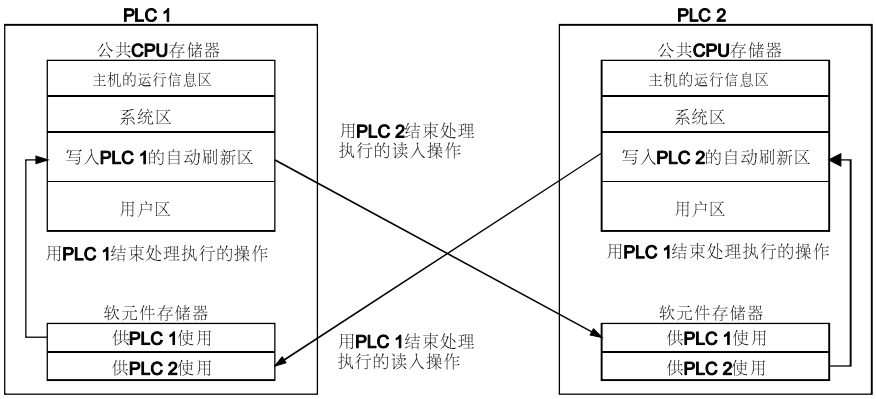
- QCPU 和 QCPU 之间
- QCPU 和动作 CPU 之间
- 动作 CPU 和动作 CPU 之间

用动作专用命令也可以从 QCPU 向动作 CPU 发布控制指令，以及用通信专用指令从 QCPU 向动作 CPU 读和写软元件数据。

(1) 软元件数据的自动刷新

在多 CPU 系统中所有的 CPU 之间已交换了数据以后，QCPU 和动作 CPU 的操作系统运行“END”处理时，会自动执行软元件数据的自动刷新。

因为在使用自动刷新功能时，会自动读取其它设备的软元件存储器数据，所以主机能够使用其它设备的软元件数据。

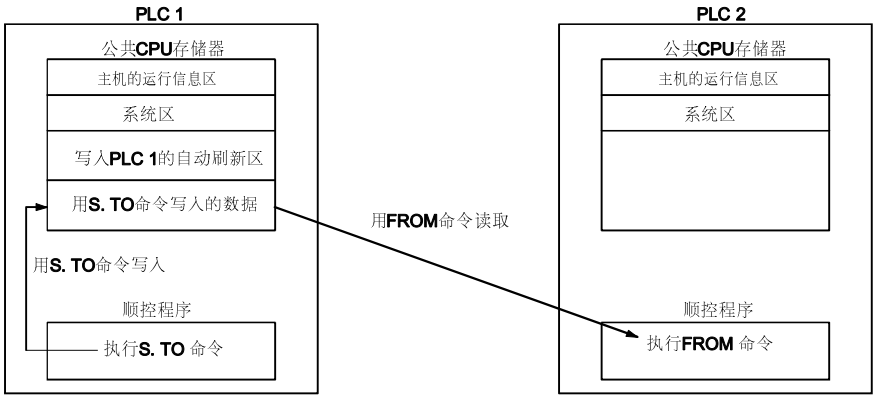


(2) 用多 CPU 命令和使用（U□\G□）的命令交换数据

使用 S TO 命令/FROM，多 CPU 系统的 CPU 写入数据到主机的公共 CPU 存储器中。

用 S TO 命令写入主机的公共 CPU 存储器数据是由其它设备的 QCPU 用 FROM 命令和 U□\G□ 读取的。

执行命令时，也可直接读取非通信软元件数据。



16

16 - 1

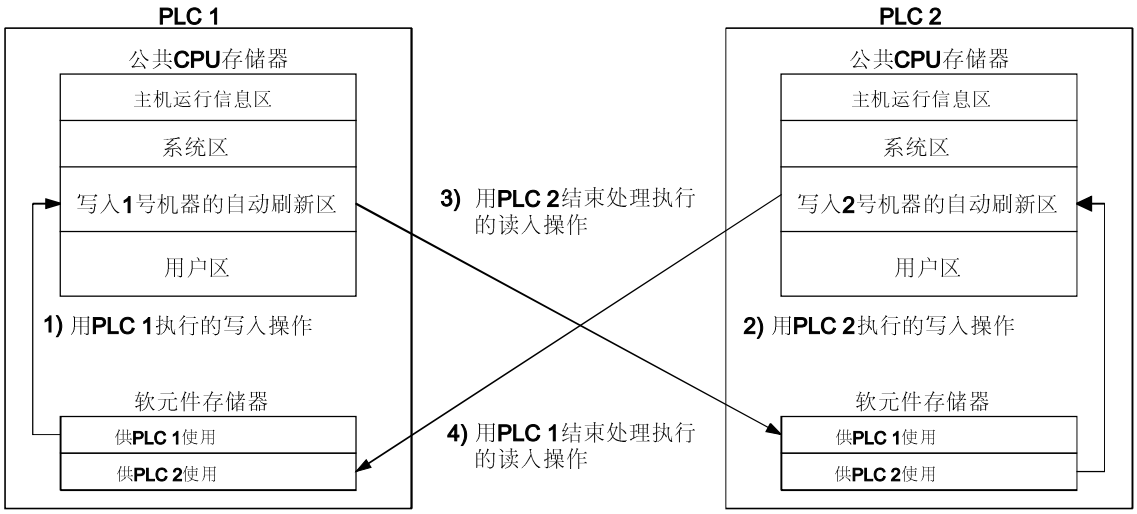
16.1 公共 CPU 存储器的自动刷新

(1) 公共 CPU 存储器的自动刷新

(a) 在多 CPU 系统的所有 CPU 之间交换了数据以后，QCPU 和动作 CPU 操作系统运行 END 处理时，会自动执行公共 CPU 存储器的自动刷新。在执行自动刷新功能时，会自动读取其它设备的软元件存储器数据。可以使主机使用其它设备的软元件数据。

1 号设备在 B0 到 B1F 之间执行 32 点的自动刷新。

2 号设备在 B20 到 B3F 之间执行 32 点的自动刷新。



运行的概况如下：

在 PLC 1 结束处理期间执行的处理过程

1）：PLC 1 的 B0 到 B1F 传输软元件数据被传送到主机公共存储器的自动刷新区。

4）：PLC 2 公共存储器的自动刷新区中的数据被传送到主机的 B20 到 B3F。

在 PLC 2 结束处理期间执行的处理过程

2）：PLC 2 的 B20 到 B3F 传输的软元件数据被传送到主机公共存储器的自动刷新区。

3）：PLC 1 公共存储器的自动刷新区中的数据被传送到主机的 B0 到 B1F。

(b) 执行自动刷新

当 QCPU 或动作 CPU 处在 RUN 状态，STOP 状态或 PAUSE 状态时执行自动刷新。当 QCPU 或动作 CPU 发生 STOP（停止）出错时，不能执行自动刷新。

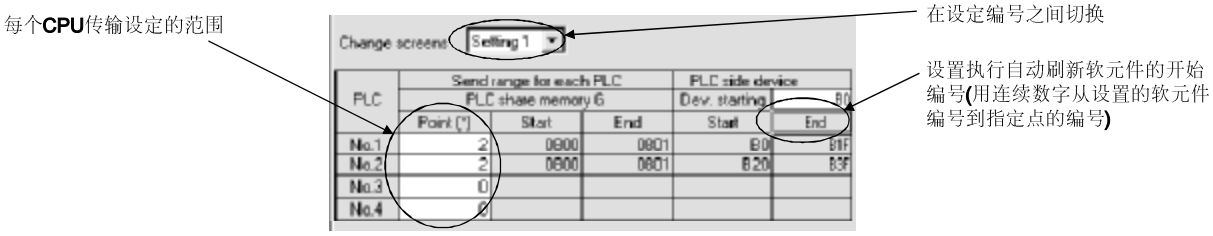
如果一个模块发生了 STOP（停止）出错，尚未发生出错的其它模块，在发出（停止）出错信号之前，将保存数据。

例如：当 B20 为 ON 时，2 号设备发生 STOP（停止）出错，1 号设备的 B0 将保持 ON，运行情况如图（a）中所示。

(c) 当执行自动刷新时，必须在 PLC 参数的多 CPU 设定中设置每一个 CPU 和存储数据的软元件传输的点（将执行自动刷新的软元件）。

(2) 自动刷新的设定

用 PC 参数的多 CPU 设定来设置执行自动刷新时每一个 CPU 和储存数据的软元件传输的点。



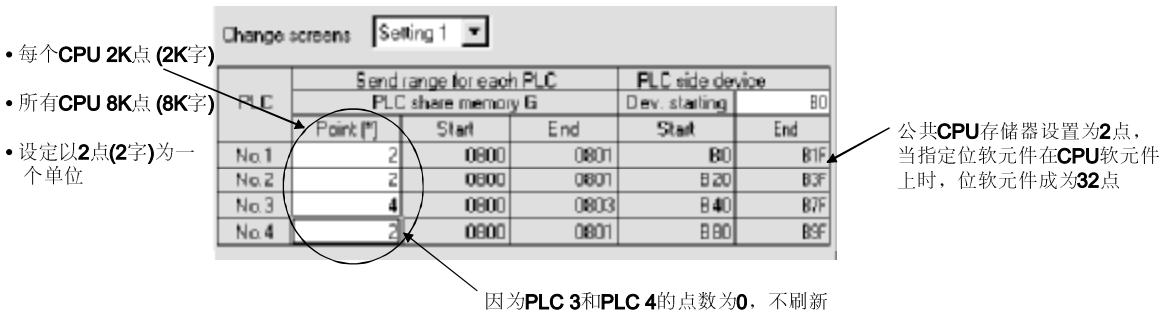
(a) 设置开关和每个 **CPU** 的传输范围（刷新范围）

1）：用设置开关，可以为刷新设置 **4** 个范围（从设置 **1** 到设置 **4**）。例如：可以将刷新功能设置为将 **ON/OFF** 数据存入位数软元件的设置 **1** 和将其它数据存入字软元件的设置 **2**。

2）：每个 **CPU** 的传输范围设置以二个公共 **CPU** 存储器点（两个字）为单位。（当指定 **CPU** 软元件的字软元件时成为 **2** 个点，指定位数软元件时成为 **32** 个点）。

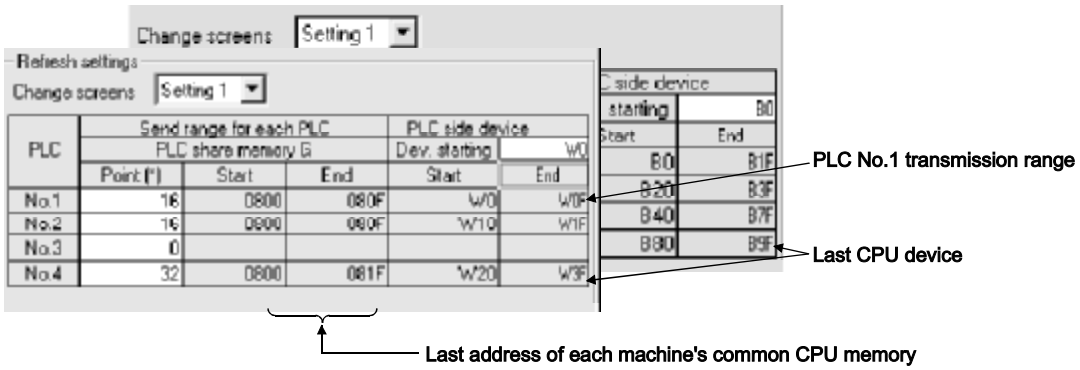
每个 **CPU** 的传输范围中，设定为点“**O**”的设备数据不会被刷新。在 **PLC 1** 的 **B0** 和 **B1F** 之间 **32** 点和 **PLC 2** 的 **B20** 和 **B3F** 之间 **32** 点执行刷新时，在公共 **CPU** 存储器的一个点上，一个位软元件成为 **16** 点，传输点的数目是 **PLC 1** 为 **2** 点，**PLC 2** 为 **2** 点。

3）：传输的点数在总的四个范围内，每个 **QCPU** 和动作 **CPU** 最大是 **2K** 点（**2K** 字），对所有的 **CPU** 总数是 **8K** 点。



16 - 3

4）：自动刷新占据的公共 CPU 存储器刷新为设置 1 到设置 4 的总和。
所使用的公共 CPU 存储器的第一个和最后一个地址，在设置传输点的数目时，会以十六进制的形式显示出来。
在设置 1 和设置 2 中已经设置传输点的设备将成为设置 2 公共 CPU 存储器最后的地址。（用于 PLC 1 和 PLC 2 的向上直到 811H 和用于 PLC 4 的向上直到 821H，如下图所示）
仅传输设置 1 的设备将成为设置 1 公共 CPU 存储器的最后地址。（在下图中，PLC 3 是上升到设定 1 的地址）



5）：多 CPU 系统中，所有的设备都应设置成相同的传输点数。
如果某一台设备的传输点数不相同，将发生“参数错误”出错。

(b) CPU 软元件
下列软元件可用于自动刷新目的（其它软元件不能用 GX-Developer 来设置）。

可建立的软元件	注意
数据寄存器（D） 通信寄存器（W） 文件寄存器（R，ZR）	• 左列软元件对每个传输点，只占据一点
通信继电器（B） 内部继电器（M） 输出（Y）	• 指定第一个数为 0 或 16 的倍数 • 左列软元件对于每个传输点，只占据一个点

1） CPU 软元件使用全部传输点软元件，这些软元件包括从指定的软元件编号到中第一次设定范围中的 PLC1 至 PLC4 的各连续编号。
设置软元件的编号，以便固定传输点软元件的数量。
如果 CPU 软元件中指定了一个位软元件，就要设置传输点数的十六倍的点。
例如：PLC 1 到 PLC 4 的总传输点数为 10。
当指定 B0 通信继电器时，在 B0 到 B9F 之间将设置 160 个点。

2） CPU 软元件按如下设置

- 改变软元件及设定设置 1 到 4 是可能的，只要设置 1 到设置 4 软元件的范围不重迭，就可以指定相同的软元件。

设置 1: 在链接继电器时

Change screens: Setting 1

PLC	Send range for each PLC			PLC side device	
	PLC share memory G			Dev. starting	B0
	Point [*]	Start	End	Start	End
No.1	2	0800	0801	B0	B1F
No.2	2	0800	0801	B20	B3F
No.3	4	0800	0803	B40	B7F
No.4	2	0800	0801	B80	B9F

设置 2: 在链接寄存器时

Change screens: Setting 2

PLC	Send range for each PLC			PLC side device	
	PLC share memory G			Dev. starting	W0
	Point [*]	Start	End	Start	End
No.1	16	0802	0811	W0	W0F
No.2	16	0802	0811	W10	W1F
No.3	0				
No.4	0				

设置 3: 在链接继电器时

Change screens: Setting 3

PLC	Send range for each PLC			PLC side device	
	PLC share memory G			Dev. starting	B100
	Point [*]	Start	End	Start	End
No.1	2	0812	0813	B100	B11F
No.2	2	0812	0813	B120	B13F
No.3	4	0804	0807	B140	B17F
No.4	4	0802	0805	B180	B18F

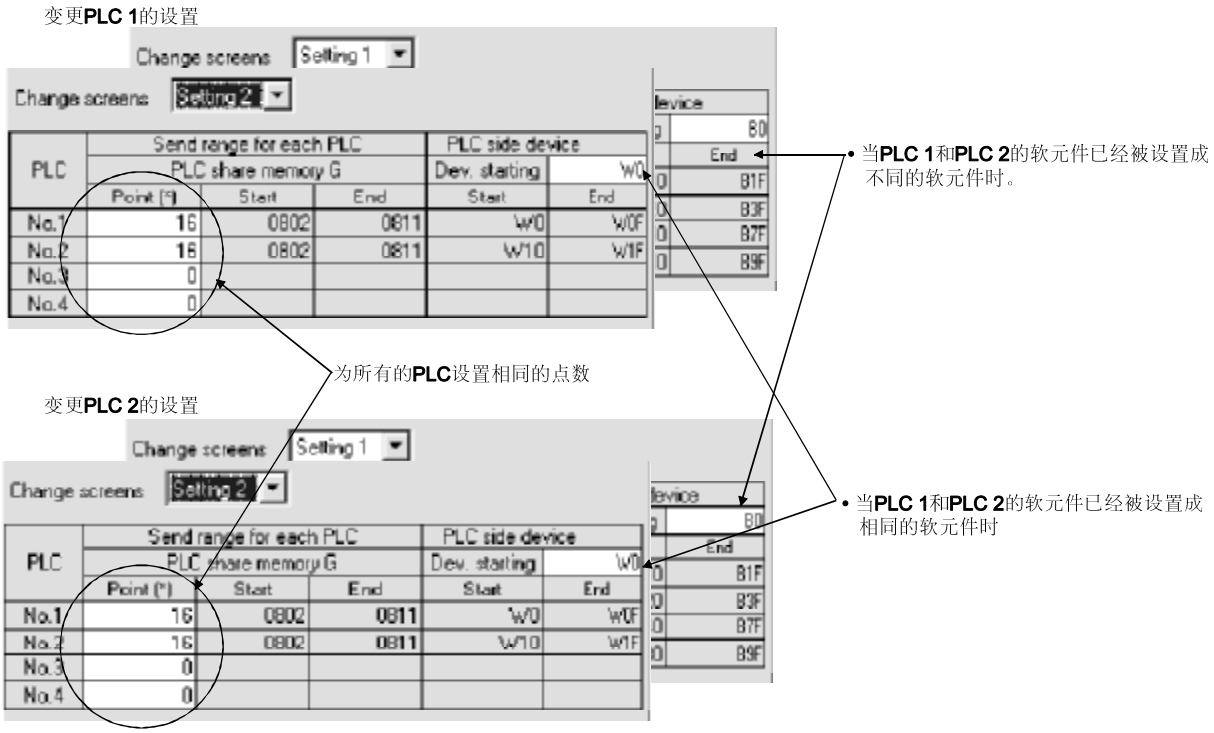
第一个和最后一个将由 GX Developer 自动进行计算

- 可以改变软元件及设定设置1到设置4

- 对设置1到设置4，可以指定相同的软元件，然而，如图左所表明的，设置1在B0和B9F之间使用可用于设置3的3160点BA0或更高的点。软元件编号的一部分不可以加倍。B0到B9F用于设置1，B90到B10F用于设置3。

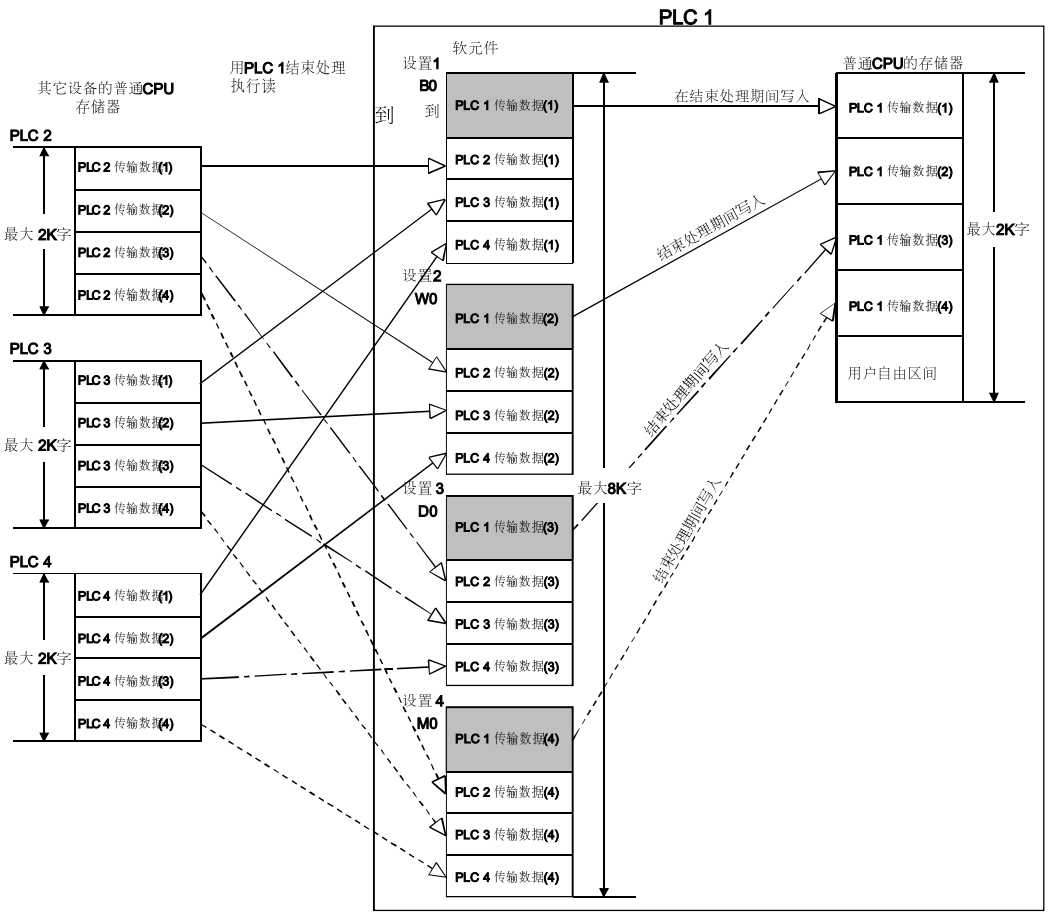
16 - 5

- 设置 **1** 到设置 **4** 软元件中的每一个都能独立地进行设置。
  例如：**PLC 1** 可以设置成一个通信继电器，而 **PLC 2** 也可以设置成一个内部继电器。



16 - 6

3） 当自动刷新功能分成四个范围（设置 1：通信继电器（B）；设置 2：通信寄存器（W）；设定 3：数据寄存器（D）；设定 4：内部继电器（N））随后如下图所示，执行运行：



(3) 注意事项

(a) 自动刷新功能使用的软元件范围设置，不能设置在本地软元件中。
如果：自动刷新功能使用的软元件范围设置，设置在本地软元件中，则该设置刷新数据不起作用。

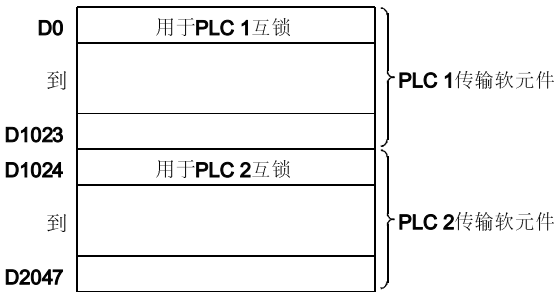
(b) 不要将自动刷新功能使用的软元件设置在所有程序的文件寄存器中。如果将自动刷新功能使用的软元件在所有程序的文件寄存器中。自动刷新将在与上次执行的扫描类型程序相对应的文件寄存器上执行。

(c) 某些情况下，根据刷新主机的定时和从其它设备读取数据的时间，旧的数据和新的数据会混合起来。

当执行自动刷新功能时，创建一个类似于以下所示的使用每台设备第一个刷新软元件的互锁程序，并且在旧的数据未与新数据混合时，不使用从其它设备来的数据。

用下列多 CPU 设定来设置一个程序的实例，刷新设置如下：

- CPU 软元件：D0
- PLC 1 传输点数：1024 点（D0 到 D1023）
- PLC 2 传输点数：1024 点（D1024 到 D2047）



传输端的程序例子

与PLC 2的第一软元件的B0互锁

写入命令

D1024.0

设置在D0至D1023中的传输数据

D0.0

当传输数据设置已完成时，用于互锁的PLC 1的第一软元件（b0）设置为ON。

接收端的程序例子

与PLC 1第一软元件的B0互锁

D0.0

使用传输数据 (D0 至 D1023)的操作

D1024.0

当使用接收到的数据的运行已经完成，用于互锁的PLC 2的第一软元件（D1024）的b0设置为ON。

16.2 用多 CPU 命令和智能型功能模块软元件进行通信

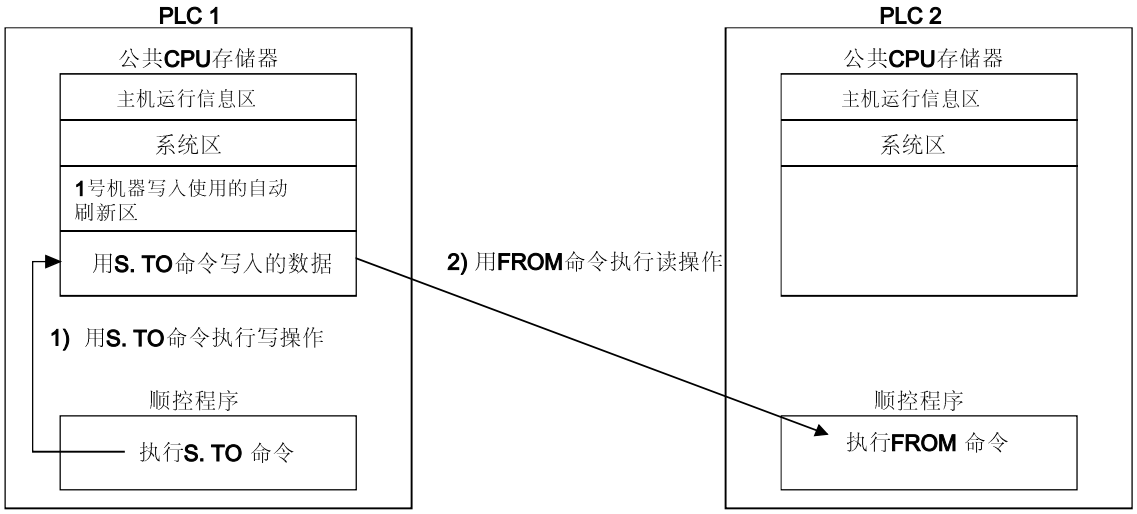
(1) 用多 CPU 命令通信

通过使用 S TO 和 FROM 命令，可以从多 CPU 系统的 QCPU 访问 QCPU/动作 CPU 的公共 CPU 存储器。

用 S TO 命令写入到主机公共 CPU 存储器的数据，可以通过其它设备用 FROM 命令来读取。

公共 CPU 存储器与自动刷新功能相反，在执行命令时，可以直接读取数据。

当用 S TO 命令写入到 PLC 1 的公共 CPU 存储器中的数据，被 PLC 2 用 FROM 命令读取时，执行程序的运行情况如下图所示。



PLC 1 处理

1） 用 S TO 命令写入数据到 PLC 1 的用户区。

PLC 2 处理

2） 用 FROM 命令从 PLC 1 的用户区读出指定的软元件。

参考下列手册，以进一步了解 S TO 命令和 FROM 命令的细节。

QCPU（Q 模式）/QnACPU 编程手册（公共命令）

要点

对于动作 CPU，没有 S TO 命令，FROM 命令，或智能型功能模块软元件，因此，S TO 命令，FROM 命令或智能型功能模块软元件都不能用于 QCPU 与动作 CPU 之间的通信。

要在 QCPU 与动作 CPU 之间通信，可使用公共 CPU 存储器自动刷新或多 CPU 之间通信的专用命令。

(2) 注意事项

（a） 下列数值是用 FROM 命令，S TO 命令和使用 U□\G□的命令设置在 QCPU 的 I/O 起始地址中的。

PLC 号	PLC 1	PLC 2	PLC 3	PLC 4
设置在 I/O 起始地址的数值	$3E0_H$	$3E1_H$	$3E2_H$	$3E3_H$

（b） 可以在公共 CPU 存储器的 OH 到 FFFH 的范围中存取数据。然而，不能在系统区或自动刷新区执行写入操作，如同读入操作一样。（见 16.4 节）

（c） 当用 FROM 命令，S. TO 命令和使用 U□\G□的命令访问的 CPU 复位时，将发生出错。然而，当命令执行完成时，访问执行的标志(SM390)将保持在 OFF 状态。

（d） 建立互锁来防止用 FROM 命令，S. TO 命令和使用 U□\G□的命令进行数据通信期间发生同时访问。
某些情况下，如果发生同时访问，旧的数据和新的数据将混合在一起。

（e） 如果用 S. TO 命令和使用 U□\□的命令将数据写入到其它设备的公共 CPU 存储器中，将发生“SP UNIT ERROR”出错（出错代码 2115）
如果使用 U□\G□的命令把数据写入到主机的公共 CPU 存储器中，会发生“SP UNIT ERROR”出错（出错代码 2114）。

（f） 如果用 FROM 命令和使用 U□\G□的命令从主机的公共 CPU 存储器中读出数据，将发生“SP UNIT ERROR”出错（出错代码 2114）。

（g） 如果使用 U□\G□命令，试图对未安装的设备进行访问，也会发生“SP UNIT ERROR”出错（出错代码 2110）。

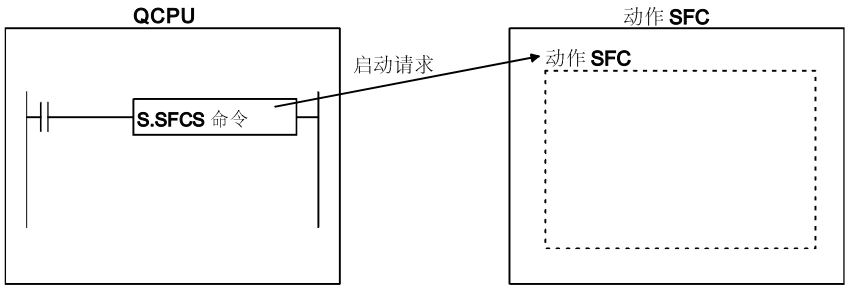
16.3 QCPU 与动作 CPU 之间的相互通信

16.3.1 从 QCPU 到动作 CPU 的控制指令

可以从 QCPU 向动作 CPU 发布控制指令，以及用下表所列的特殊动作命令读和写软元件数据。
（不能使用从动作 CPU 到动作 CPU 的控制指令。）

命令名称	说明
S.SFCS SP.SFCS	请求启动动作 SFC 程序
S.SVST SP.SVST	请求伺服程序运行开始
S.CHGV SP.CHGV	在定位和点动操作期间修改轴的速度
S.CHGT SP.CHGV	在实数模式运行和暂停期间，修改转矩控制值
S.CHGA SP.CHGA	修改暂停轴、同步编码器和凸轮轴的当前值

例如：从 QCPU 使用 S(P).SFCS 命令，可以启动动作 CPU 的动作 SFC。



要点

一个 QCPU 可以最多一次运行 32 条“特殊动作命令”和多 CPU 之间的“通信专用命令”（省去了 S(P).GINT 命令）。然而，如果“特殊动作命令”和多 CPU 之间的“通信专用命令”（省去了 S(P).GINT 命令）是同时产生的，则从接收到的第一条命令开始依次执行。如果有 33 条或更多未执行的命令，将会发生“操作错误”出错（出错代码 4107）。

备注

参考动作 CPU 的编程手册，了解“动作专用命令使用的必要性”细节。

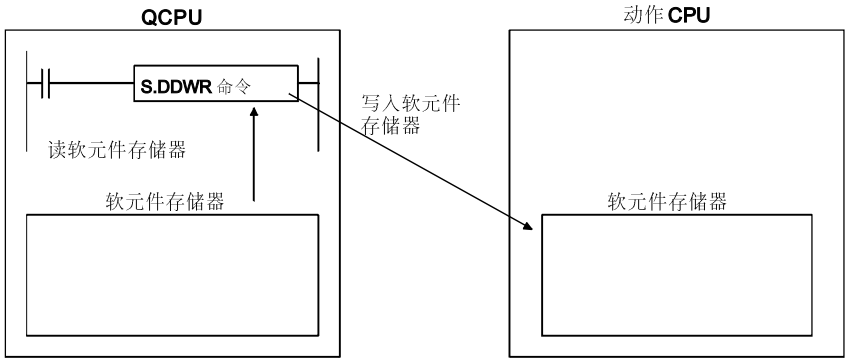
16 - 11

16.3.2 读和写软元件数据

可以用下表列出的在多 CPU 之间使用的通信专用命令，从 QCPU 读和写软元件数据到动作 CPU 中去。
（不能在 QCPU 到 QCPU，动作 CPU 到 QCPU，动作 CPU 到动作 CPU 之间进行读和写的操作。）

命令名称	说明
S.DDWR SP.DDWR	把主 CPU 软元件数据写入到其它 CPU 软元件中。
S.DDRD SP.DDRD	读其它 CPU 软元件数据到主 CPU 中。
S.GINT SP.GINT	请求启动其它 CPU 的中断程序。

例如：可以用多 CPU 之间通信专用命令中的 S.DDWR 命令把 QCPU 软元件数据写入到动作 CPU 的软元件数据中。



要点

一个 QCPU 可以最多一次运行 32 条“特殊动作命令”和多 CPU 之间的“通信专用命令”（省去了 S(P).GINT 命令）。然而，如果“特殊动作命令”和多 CPU 之间的“通信专用命令”（省去了 S(P).GINT 命令）是同时产生的，则从接收到的第一条命令开始依次执行.如果有 33 条或更多未执行的命令,将会发生“操作错误”出错（出错代码 4107）。

备注

参考动作 CPU 的编程手册，了解使用多 CPU 之间的“特殊通信命令”必要性的细节。

16 - 12

16.4 公共 CPU 存储器

公共 CPU 存储器是用来在 QCPU 之间进行数据交换的，它由在 0H 到 FFFFH 之间的 4096 字组成。

公共 CPU 存储器由四个区域组成：主机运行信息区，系统区，自动刷新区和用户区。当设置了软元件数据的自动刷新时，一个从 800H 开始、含有一定数量的自动刷新点数的区域就作为自动刷新区。用户区是直接接在自动刷新区末端的地址后面开始的。如果自动刷新点数是 18（11H 点），那末，800H 到 811H 就成为自动刷新区，而 812H 以后的区则成为用户区。公共 CPU 存储器的配置和访问顺控程序的必要性如下表所列：

公共CPU存储器		主机		其他机器	
		写 *1	读	写	读 *2
0H 到 1FFH	主机运行信息区	不可能	不可能	不可能	可能
200H 到 7FFH	系统区	不可能	不可能	不可能	不可能
800H	自动刷新区	不可能	不可能	不可能	不可能
到 FFFH	用户区	可能	不可能	不可能	可能

备注

*1：在执行写入主机用户区操作时使用 S. T0 命令。
*2：执行从其它设备的读操作时使用 FROM 命令和智能功能模块软元件 (U □ /G)

16 - 13

(1) 主机运行信息区(O_H到 1FFH)

（a） 下列信息存储在多 CPU 系统的主机中，在独立 CPU 系统中，这些参数将保持为 0，而且不会改变。*1

表 16.1 主机运行信息区一览表

公共存储器地址	名称	细节	说明*2	相应的特殊寄存器
O_H	信息的有效性	信息有效性标志	确认信息是否存储在主机的运行信息区(1_H-$1F_H$)中的区 • 0：信息不存储在主机的运行信息区中 • 1：信息存储在主机的运行信息区中	—
1_H	诊断出错	诊断出错编号	在用二进制存储诊断信息期间，发生出错的编号	SD0
2_H	诊断出错发生时间	诊断出错发生时间	将存储在公共 CPU 存储器 1H 地址中的出错编号发生的年和月用二位数字 BCD 码存储起来	SD1
3_H			将存储在公共 CPU 存储器 1H 地址的出错编号发生的日期和时间用二位数字 BCD 码存储起来	SD2
4_H			将存储在公共 CPU 存储器 1H 地址的出错编号发生的分和秒用二位数字 BCD 码存储起来	SD3
5_H	出错信息识别代码	出错信息识别代码	存储识别代码来决定已经存储的出错信息是公共错误信息还是个别错误信息	SD4
6_H-10_H	公共出错信息	公共出错信息	存储与诊断期间发生的出错编号相应的公共信息	SD5-SD15
11_H-$1B_H$	个别出错信息	个别出错信息	存储与诊断期间发生的出错编号相应的个别信息	SD16-SD26
$1C_H$	空	—	不能使用	—
$1D_H$	开关状态	CPU 开关状态	存储 CPU 开关状态	SD200
$1E_H$	指示灯状态	CPU 指示灯状态	存储 CPU 的 LED 位图	SD201
$1F_H$	CPU 运行状态	CPU 运行状态	存储 CPU 的运行状态	SD202

*1. 对于动作 CPU，5H4 到 1CH 的主机运行信息区是不用的，如果主机运行信息区的 5H4 到 1CH 从动作 CPU 来读取，它将作为“0”读出。

*2. 参考有关的特殊寄存器的细节。

（b） 有时，当相应寄存器的内容改变时，主机运行信息区会更新。然而，当 CPU 的扫描时间小于或等于 200ms 时，有时相应寄存器中变化将延迟不超过 200ms 的时间。
如果 CPU 的扫描时间超过 200ms，在相应的寄存器中的变化将延迟 200ms 或更长。

（c） 可以用 FROM 命令从其它设备来读取主机运行信息区的数据。然而，因为数据更新会延迟，所以将读出的数据用于监控用途。

(2) 系统区（200H 到 7FFH）

QCPU 和动作 CPU 系统（操作系统）所使用的区域。当多 CPU 之间执行特殊通讯命令时，操作系统使用这个区。

(3) 自动刷新区

自动刷新多 CPU 系统时使用的区域。不能用 S TO 命令写，也不能用 FROM 命令读数据。

(4) 用户区

在 CPU 之间，用多 CPU 系统的 S.TO(U□\G□)命令和 FROM 命令进行通信的区域。设置用于自动刷新点被以后使用的区域。（当不执行自动刷新时，在 800H 和 FFH 之间的区域可用作用户区）。

17 多 CPU 系统的 I/O 模块与智能型功能模块之间的通信

17.1 控制 PLC 通信的范围

控制 PLC 和控制模块(I/O 模块，智能型功能模块,特殊功能模块)之间的关系是与独立 CPU 系统中的关系相同的。
用控制 PLC 控制控制模块无限制条件。

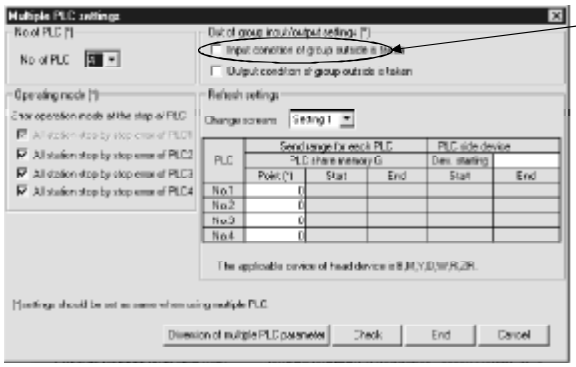
17.2 非控制 PLC 通信的范围

非控制 PLC 可以读取智能型功能模块缓冲存储器的内容。
也可以用 PLC 参数来装入非控制模块的输入（X）ON/OFF 数据和另一设备模块的输出（Y）ON/OFF 数据。
由其它设备控制的输入模块可以当做主机的互锁来使用，且可确认输出到由其它设备控制的外围设备的输出状态。
然而，非控制 PLC 不能输出 ON/OFF 数据到输出模块或智能型功能模块，或写入到智能型功能模块的缓冲存储器中。

17 - 1

(1) 从输入模块和智能型功能模块加载输入(X)

在 PC 参数的多 CPU 设定中，“组合以外的输入”设定决定了是否能从由其它设备控制的输入模块和智能型功能模块来加载输入。



"组合外的输入"设定
☐ 加载"组合外的输入"
不加载"组合外的输入"设定
☑ 加载"组合外的输入"
"加载组合外的输入"设定

(a) 设置“加载组合外的输入”

1） 用“END”处理的输入刷新功能，来从由其它设备控制的输入模块和智能型功能模块加载 ON/OFF 数据。

2） 对安装在下列附加基板槽中的模块执行加载输入（X）。

I/O 分配类型	安装的模块	备注
无	输入模块	—
	智能型功能模块	—
输入	输入模块	—
	输出模块	加载 OFF 数据
智能	智能型功能模块	—

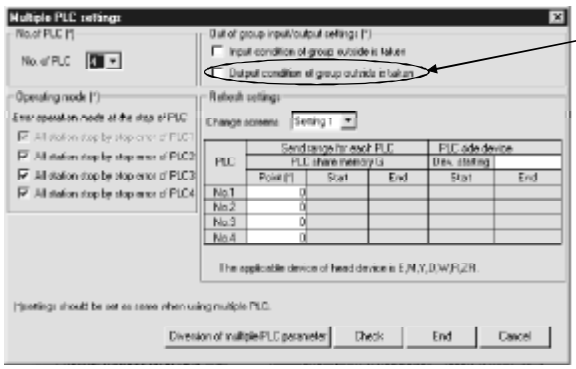
3） 用直接访问输入可以从输入模块和智能型功能模块加载 ON/OFF 数据。

4） 远程站输入，如：空槽，MELSECNET/H 和 CC-Link 不能加载。
使用软元件数据自动刷新功能将 ON/OFF 数据用于 MELSECNET/H，CC-Link 和主站中的其它远程站。

(b) 当设置了“不加载组合外的输入”时，就不能从由其它设备控制的输入模块和智能型功能模块加载 ON/OFF 数据。（保持在 OFF）。

(2) 加载输出（Y）

在 PC 参数的多 CPU 设定中的“组合外输入”设定决定了能否从正由其它设备控制的输出模块和智能型功能模块加载输出。



"组合外输入"设定
☐ 加载"组合外的输出"
"不加载组合外输入"设定
☑ 加载"组合外的输出" :
"加载组合外输入"设定

(a) 当设置了“加载组合外输出”时

1） 用“END”处理的输入刷新功能，把 ON/OFF 数据输出加载到输出由其它设备控制的智能型功能模块加入到主机的输出（Y）。

2） 对安装在下列附加基板槽中的模块执行加载输出（Y）。

I/O 分配类型	安装的模块	备注
无	输出模块	—
	智能型功能模块	—
输入	输入模块	—
	输出模块	—
智能	智能型功能模块	—

3） 可以用直接访问输出来加载正由其它设备控制的输出 ON/OFF 数据。

4） 远程站的输入，如：空槽，MELSECNET/H 和 CC-Link 不能加载。使用软元件数据自动刷新功能，把输入 ON/OFF 数据用于 MELSECNET/H，CC-Link 和主站中的其它远程站。

(b) 当设置了“不加载组合外的输出”时，不能从由其它设备控制的输出模块和智能型功能模块加载“ON/OFF 数据输出”到主机的输出（Y）（保持在 OFF）。

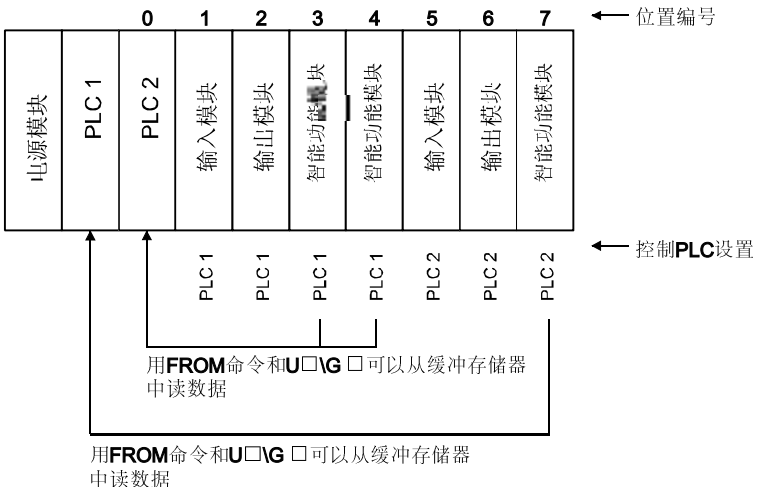
(3) 输出到输出模块和智能型功能模块

不能输出 ON/OFF 数据到非控制模块。
当从其它设备（如顺控程序）控制的输出模块和智能型功能模块的输出，已设置在 ON/OFF 时，将在 QCPU 内执行 ON/OFF，但不可能输出到输出模块或智能型功能模块。

(4) 访问智能型功能模块的缓冲存储器

(a) 用下列命令可以从其它设备控制的智能型功能模块的缓冲存储中读数据。

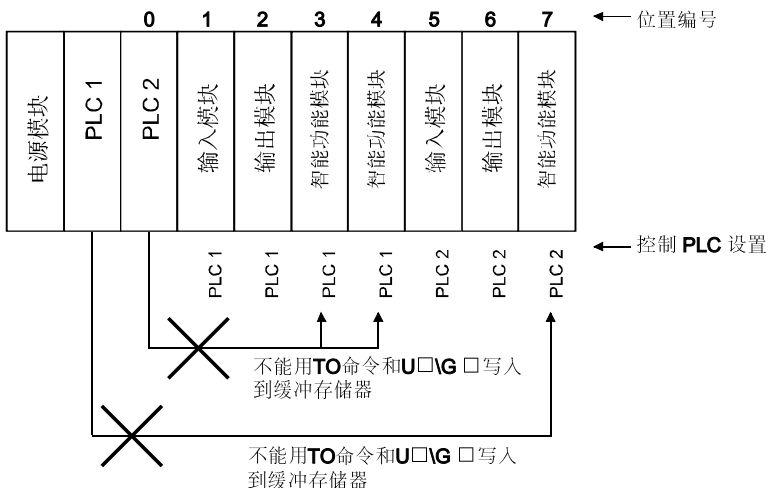
- FROM 命令
- 使用智能型功能模块软元件（U□□G□）的命令



(b) 不能写入数据到正由其它设备控制的智能型功能模块的缓冲存储器中。

- TO 命令
- 智能型功能模块的软元件（U□□G□）
- 智能型功能模块的专用命令

如果一旦发生试图写入到其它设备控制的智能型功能模块的情况，就将发生“SP 模块出错”出错（出错代码 2116）。



(4) 访问 MELSECNET/H 模块

只有控制 PLC 才能访问 MELSECNET/H 模块。

在由其它设备控制的 MELSECNET/H 模块中，不能使用链接直接软元件。

如果由其它设备控制的 MELSECNET/H 模块中使用了使用链接直接软元件的程序，将发生“操作错误”出错（出错代码 4102）

18 多 CPU 系统 QCPU 的处理时间

18.1 QCPU 扫描时间的概念

多 CPU 系统的扫描时间概念与用单个 CPU 进行控制扫描的时间概念相同。
参考 11.1 节，了解扫描时间概念的细节。
本章提供配置多 PLC 时，关于附加到按 11.1 节所说明的方法计算出的扫描时间上的附加因素的说明。

(1) I/O 刷新时间

按照 11.1 节所说明的方程式来计算输入刷新时间。
下列数值的 I/O 刷新时间只有当总线访问与其它设备重叠时才延长。

延长时间=（输入点数+输出点数）/16×N3×（其它设备的数目）（μs）

N3 使用下列数值

CPU 类型	N3	
	系统只有一块主基板	系统有附加基板
Q02CPU Q02HCPU,Q06HCPU Q12HCPU,Q25HCPU	8.7	21

(2) 命令执行时间的总和

参考下列手册，了解特殊多 CPU 命令的处理时间和多 CPU 系统中有不同处理时间的命令的处理时间。

- QCPU（Q 模式）和 QnACPU 编程手册（公共命令）

(3) 结束处理

下列时间值是常用的结束处理时间。

CPU 类型	END 处理时间（ms）
Q02CPU	0.42
Q02HCPU，Q06HCPU，Q12HCPU，Q25HCPU	0.19

18 - 1

18.2 延长扫描时间的因素

在使用下列功能时，与单 CPU 系统相比，多 CPU 系统的处理时间被延长。
把下列数值加到按 11.1 节和 18.a 所计算的数值上去, 就获得使用这些功能的时间量。

- 多 CPU 系统的自动刷新
- MELSECNET/H 刷新
- CC-Link 自动刷新

(1) 多 CPU 系统的自动刷新

(a) 执行用多 CPU 设置刷新功能所需的时间值。
这个时间值是：写入到主机的公共 CPU 存储器所需的时间和从其它设备的公共 CPU 存储器读出数据所需的时间总和。在用 PC 参数的多 CPU 设定来设置刷新设定时，要加入这些时间值。

(b) 执行多 CPU 系统自动刷新功能所需的时间值可按以下公式计算。

自动刷新时间=（N1+（接收到的字点数）×N2）×（其它设备的数目）+（N3+（传输字点数）×N4）（μs）

- 接收字点数必须等于其它设备传输的字点数。
  例如：如果主机是 PLC 1，这个值必须等于 PLC 2 到 PLC 4 传输的点数。
- N1 到 N4 使用下列数值。

CPU 类型	N1	N2	N3	N4
Q02CPU	82	0.52	106	0.17
Q02HCPU，Q06HCPU，Q12HCPU，Q25HCPU	27	0.44	27	0.08

(c) 在其它设备上使用自动刷新功能，处理重复时，自动刷新处理所需的时间值要延长以下的时间值。

延长时间=（传输/接收字点数）×N5×（其它设备的数目）（μs）

N5 使用下列数值

CPU 类型	N5	
	系统只有一块主基板	系统有附加基板
Q02CPU Q02HCPU,Q06HCPU Q12HCPU,Q25HCPU	0.54	1.3

18

18 - 2

(2) MELSECNET/H 刷新

(a) 在 QCPU 与 MELSECNET/H 网络模块之间执行刷新处理所需的时间。
参考下列手册了解 MELSECNET/H 刷新时间的细节。
- Q 系列对应的 MELSECNET/H 网络模块刷新手册

(b) 当多 CPU 系统中其它 MELSECNET/H 网络模块同时发出刷新请求时，自动刷新处理所需的时间将只延长下列的时间值。

延长时间=（传输/接收字点数）×N4×（其它设备的数目）（μs）
传输/接收的字点数是下列传送数据的总和。
- 链接刷新数据 ：（LB+LX+LY+SB）/16+LW
- 传送到存储卡的文件寄存器中的数据 ：（LB+LX+LY+SB）/16+LW
- 在数据通讯之间传送的数据 ：（LB/16+LW）×2

CPU 类型	N4	
	系统只有一块主基板	系统有附加基板
Q02CPU Q02HCPU,Q06HCPU, Q12HCPU,Q25HCPU	0.54	1.30

(3) CC-Link 的自动刷新

(a) 在 QCPU 与 CC-Link 本地主站模块之间执行刷新处理所需的时间。
参考下列手册，了解 CC-Link 自动刷新时间的细节。
- QJ61BT11CC-Link 系统本地主站模块用户手册

(c) 当多 CPU 系统中的其它 CC-Link 模块同时发生刷新请求时，自动刷新处理所需的时间将只延长下列的时间值。

延长时间=（传输/接收的字点数）×N4×（其它设备的数目）（μs）
传输/接收的数据量是下列传送数据
- 链接刷新数据：（RX+RY+SB）/16+SW

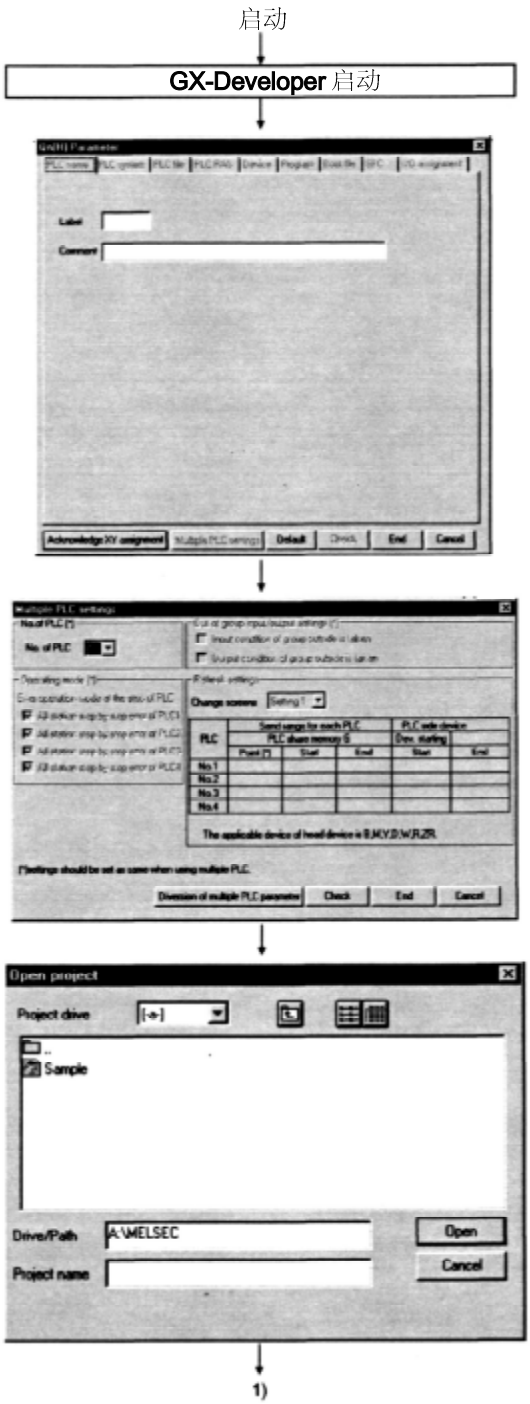
N4 参考下表

CPU 类型	N4	
	系统只有一块主基板	系统有附加基板
Q02CPU Q02HCPU,Q06HCPU, Q12HCPU,Q25HCPU	0.54	1.30

19 多 CPU 系统的启动

本章介绍启动多 CPU 系统的标准步骤。

19.1 启动多 CPU 系统的流程图



参考 GX-Developer 操作手册

打开 GX-Developer 的 PC 参数设置窗口。选择“多 CPU 设置”来显示多 CPU 设置窗口。

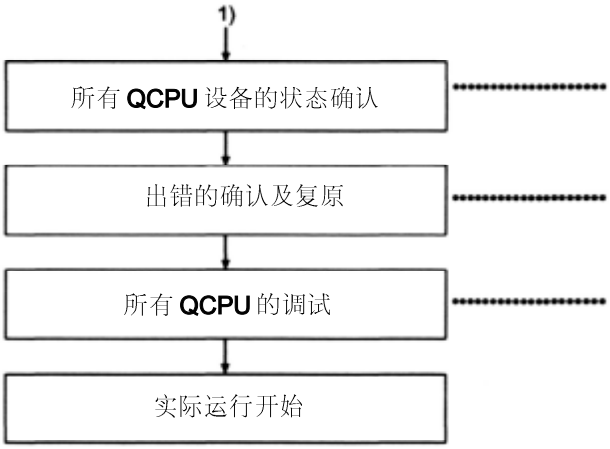
传送多 CPU 设置，点击“传送多 PLC 参数”按钮。

设置传送项目

- 选择接受传送来的现成的多 CPU 设置和 I/O 的项目。
- 点击“开”。

19

19 - 1



检查当 PLC No1 的复位状态取消时，多 CPU 系统中所有的设备是否发生 RUN 状态出错。

如果存在错误，确认其详细情况，并用 GX-Developer 的系统监视器恢复状态。

分别单独调试多 CPU 系统中的 PLC 1 到 PLC 4。

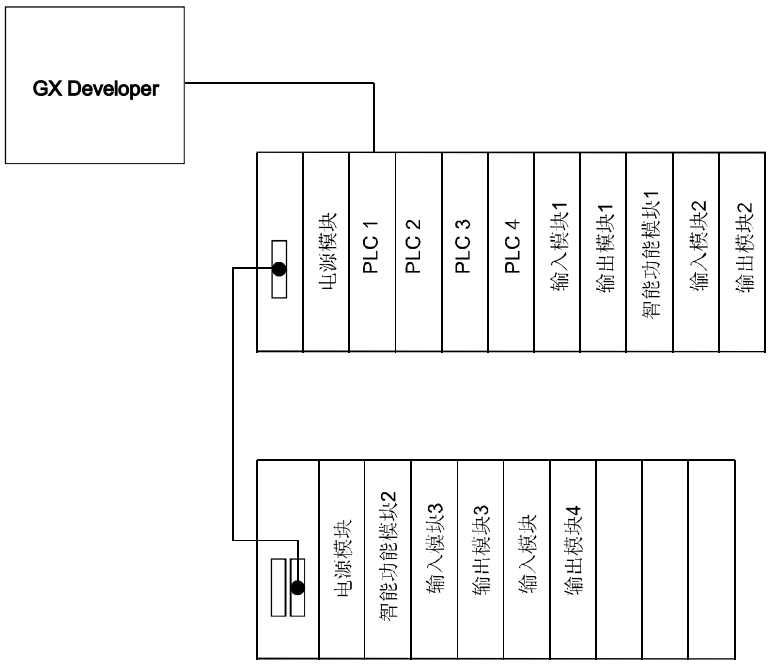
19 - 2

19.2 设置多 CPU 系统参数（多 CPU 设置，控制 PLC 设置）

本节介绍了用 GX-Developer 设置多 CPU 系统参数的步骤。
参考 GX-Developer 的操作手册，了解设置所有其它参数的细节。

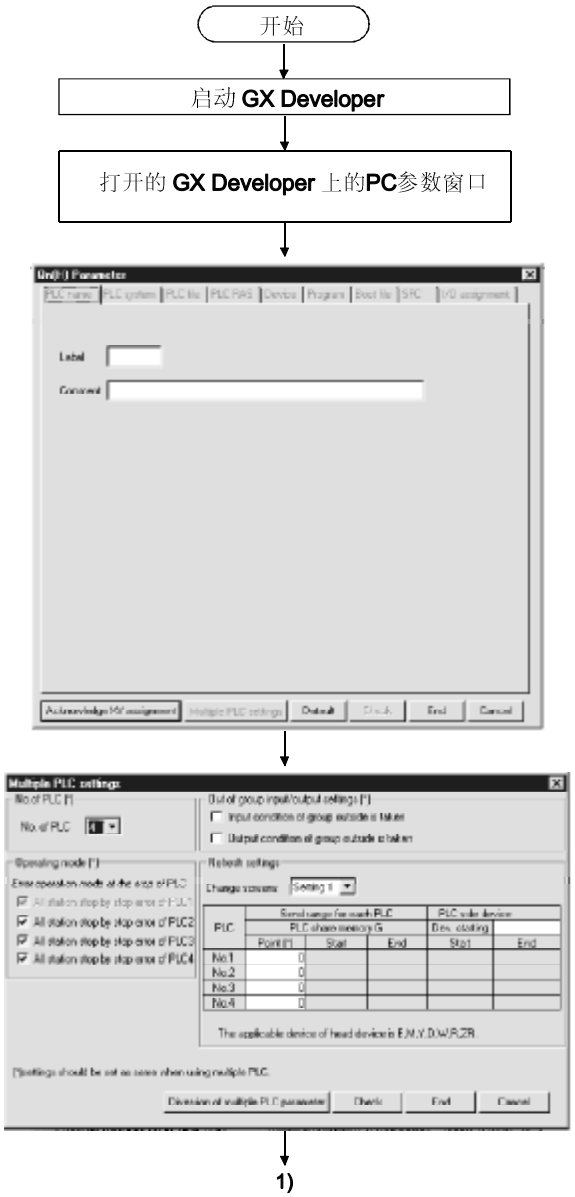
19.2.1 系统配置

以下介绍在类似下图所示的一个系统中，设置多 CPU 系统参数的步骤。



19 - 3

19.2.2 创建新系统



参考 **GX Developer** 手册

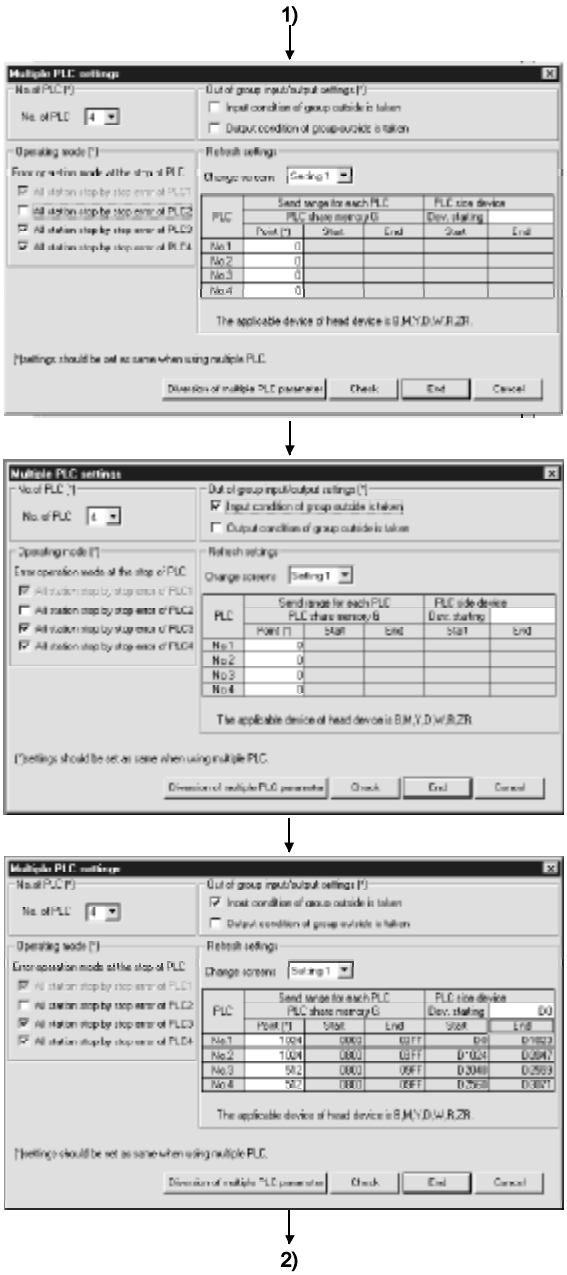
参考 **GX Developer** 手册

选择 **"多PLC设定"** 来显示多**PLC**的设置窗口

设置**CPU**的数目 **(**要求项目**)**

- 设置安装在多**PLC**系统主基板上的**QPCU**数目。

19 - 4



设定运行模式 (选择)

- 选择当发生停止出错时，全部机器停止运行还是继续运行

默认: **PLC**出错时，停运全部机器

- 例如, 如果除了 **"PLC 2**出错，则停止所有机器以外的 **V**全部取消**"** 。那么即使**PLC 1**出错，所有其它的机器将继续运行。

- 不能修改 **PLC 1** 的运行模式。

组合外的**I/O**设定 (选项)

- 设置可加载或不可加载非控制机器的输入数据和输出数据。

缺省: 不可。

多**PLC**系统(选项)

- 设置软元件和执行在**QCPU**之间与自动刷新处理进行数据通信的公共**CPU**存储器**G**的点数。

- 链接和使用从设置的第一个软元件的软元件编号到公共**CPU**存储器**G**点的编号。

第一公共**CPU**存储器**G**点占据的点数如下表所示。

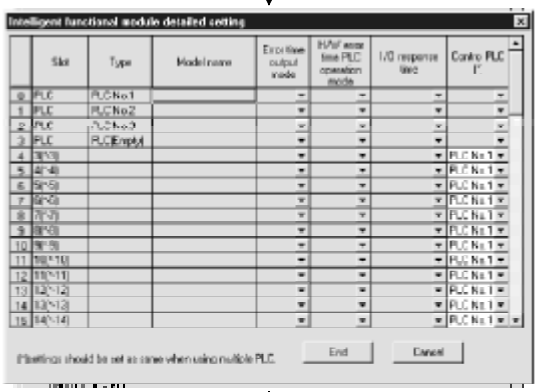
软元件	占据的点数
B, M, Y	**16** 点
D, R, ZR	**1** 点

19 - 5

2)



选择**"CPU (空)"** 作为不安装**QCPU**的槽。
选择 **I/O** 分配窗口上的**"**细节设定**"**来显示细节设置窗口。



控制 **PLC** 设置 **(**所需项目**)**

- 为每个槽选择控制**PLC (PLC 1** 到 **PLC 4)**。
- 功能版本**"A"**智能型功能模块设置控制**PLC 1**
- 输出模块和支持**AnS**系列的特殊功能模块在所有的槽中设置一单台机器。

多**PLC**系统设置以外的参数设置。

写入到硬盘或软盘中的设置参数。

结束

19 - 6

19.2.3 使用现有的预设多 CPU 设置和 I/O 分配

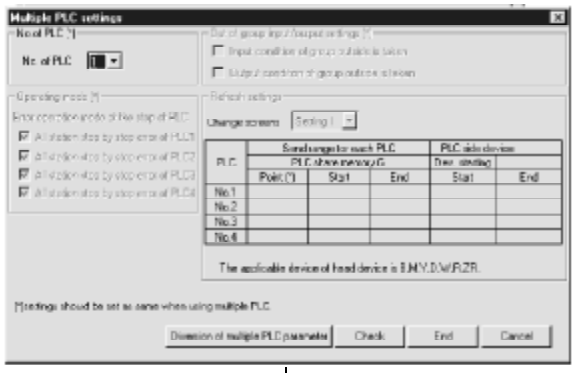
开始

GX Developer 启动

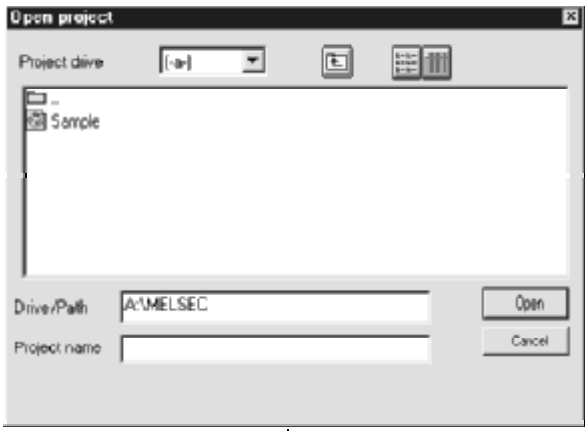
参考 GX Developer 操作手册



打开 GX Developer 的PC参数设置窗口。选择 "多PLC设置" 来显示多PLC设置窗口。



传输多PLC设置，点击 "传输多PLC参数"。



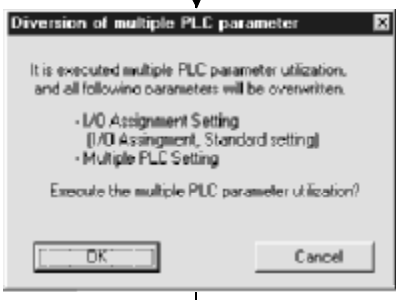
设置传输项目

- 选择接受传输来的现存多PLC设置和I/O 分配的项目。
- 点击 "开"。

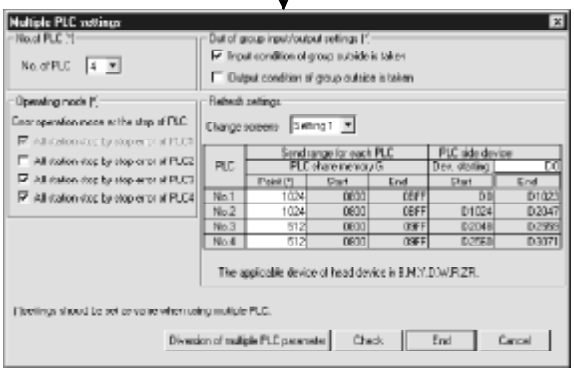
1)

19 - 7

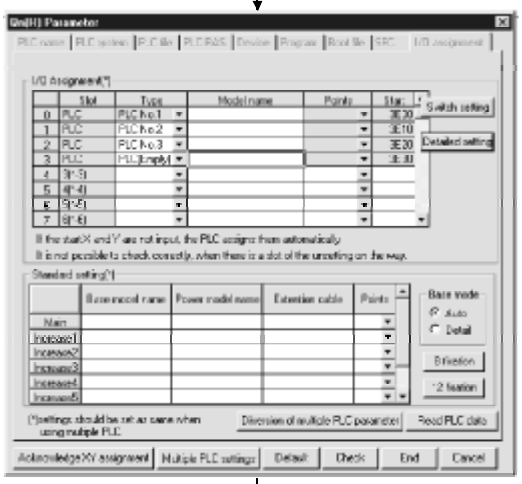
1)



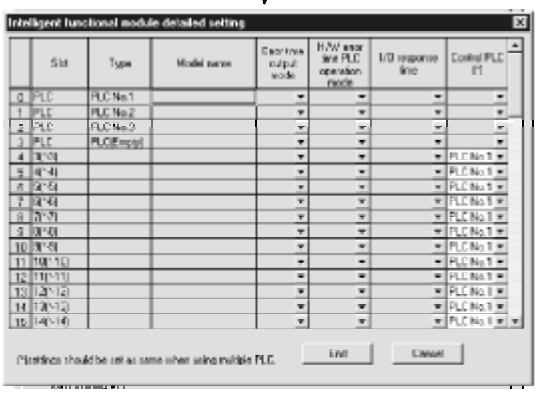
当选择**"OK"**时，多**PLC**设置和 **I/O** 分配数据被读和写到指定的项目中去。



确认多**PLC**设置
要修改**CPU**软元件时，先更改软元件的数目再输入该数目。
(没有注**"*"**的项不能修改**)**

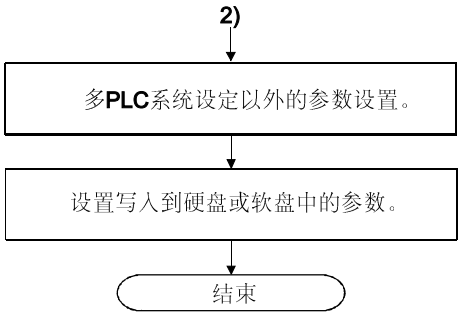


在 **"I/O** 分配窗口**"** 确认**I/O**分配和基本设置。
选择**"**细节设置**"**来显示详细设置窗口。



确认控制**PLC**设置。

2)


2)
多PLC系统设定以外的参数设置。
设置写入到硬盘或软盘中的参数。
结束

附录

附录 1 特殊继电器表

特殊继电器 SM 是在程控器中固定用途的内部继电器。
由于这个理由，顺控程序不能以使用常规内部继电器的方式来使用它。
然而，为了控制 QnACPU 的需要，它们可以被切换到 ON 或 OFF。

App.

表中的标题有下列含义

项目	项目的功能
数目	•表示特殊继电器的数目。
名称	•表示特殊继电器的名称。
含义	•表示特殊继电器的性质。
解释	•含有关于特殊继电器、性质的详细信息。
由…（设置时）来设置	•表示在执行设置时，继电器是由系统还是用户来进行设置。 <由…设置> S : 由系统设置 U : 由用户设置（在顺控程序中或在外围装置的测试运行时） S/U : 由系统和用户二者设置 <何时设置> →只有在系统设定时显示 每次 END : 在每次 END 处理期间设置 初始化 : 只有在初始化处理期间设置 (当电源接通，或从 STOP 切换到 RUN 时) 状态改变 : 只有状态改变时设置 出错 : 出错时设置 指令执行 : 执行指令时设置 请求 : 只有在用户请求时设置 (通过 SM 等.)
对应的 ACPU M9□□□	•表示对应于 ACPU 的特殊继电器 M9□□□ (当内容已经改变时，更变并作记录) •作为新加入 Q/QnACPU 的项目，表示“新”。
对应的 CPU	•表示相应 CPU 型号的名称。 O+Rem : 可适用于所有型号的 CPU 和 MELSECNET/H 远程 I/O 模块。 O : 可适用于所有型号的 CPU。 QCPU : 可适用于 Q 系列 CPU。 QnA : 可适用于 QnA 系列和 QzASCPU 系列。 远程 : 可适用于 MECSECNET/H 远程 I/O 模块。 每个 CPU 型号名称：只能适用于特殊 CPU。（例如 Q4ARCPU，Q3ACPU）

下列项目的细节参见这些手册:

•网络 → • Q MELSECNET/H 网络系统参考手册（PLC—PLC 网络）
• Q MELSECNET/H 网络系统参考手册（远程 I/O 网络）
•MELSECNET/10 网络系统参考手册

• SFC → QCPU/QnACPU 编程手册（SFC）

指示
(1) SM1200 到 SM1255 是用于 QnACPU 的。 对 QCPU，这些继电器是空的。 (2) 特殊继电器 SM1500 及以后版本专用于 Q4ARCPU。

(1) 诊断信息

编号	名称	含义	说明		由谁设置（何时设置）	对应于将 ACPU M9□□□	可应用的 CPU
SM0	诊断错误	OFF ：无错误 ON ：出错	•当诊断结果表明发生错误(包括外部诊断时)：，ON。 • 即使恢复正常运行，仍停留在 ON。		S (错误)	新	
SM1	自诊断错误	OFF ：无自诊断错误 ON ：自诊断	• 当自诊断的结果发生错误时变为 ON。 • 即使恢复正常运行，仍停留在 ON。		S (错误)	M9008	
SM5	出错公用信息	OFF ：无出错公用信息 ON ：出错公用信息	• 当 SM0 ON 时，如果有出错公用信息，则 ON。		S (错误)	新	○+Rem
SM16	出错个别信息	OFF ：无出错公用信息 ON ：出错公用信息	• 当 SM0 ON 时，如果有出错个别信息，则 ON。		S (错误)	新	
SM50	出错复位	OFF ON: 出错复位	•进行出错复位操作 •详情参见 11.3 节		U	新	
SM51	电池低电压锁存	OFF ：正常 ON ：电池低电压	• 当 CPU 或存储卡的电池电压降到设定值以下时为 ON 即使恢复正常操作以后仍然保持 ON 状态 •与 BAT、ALARM LED 指示灯同步		S (错误)	M9007	
SM52	电池低电压	OFF ：正常 ON ：电池低电压	•与 SM51 相同但电池电压恢复正常后即变为 OFF		S (错误)	M9006	○
SM53	交流断电检测	OFF ：未检测到交流断电 ON ：检测到交流断电	• 当有不超过 20ms 的电源瞬间中断时，ON。通过切换电源开关到 OFF 再接通（ON）来复位。		S (错误)	M9005	
			• 当使用 DC 电流模块期间发生小于 10ms 的瞬时电源中断时 ON，切换电源到 OFF 再接通（ON）来复位				
			• 当使用 DC 电流模块期间发生小于 1ms 的瞬间电源中断时 ON，切换电源到 OFF 再接通（ON）来复位				QCPU
SM54	MINI 链接错误	OFF ：正常 ON ：错误	• 当安装的 AJ71PT32（S3）模块中的一个被检测到 MINI*（S3）链接错误时 ON。即使恢复正常运行，仍停留在 ON。		S (错误)	M9004	QnA
SM56	运行错误	OFF ：正常 ON ：运行错误	• 发生运行错误时 ON •即使恢复正常运行，仍停留在 ON		S (错误)	M9011	○
SM60	检测熔丝熔断	OFF ：正常 ON ：模块熔丝熔断	• 即使只有一块带输出模块熔丝熔断，变为 ON，在恢复正常以后仍停留在 ON。 • 即使是远程 I/O 工作站输出模块，也要检查熔丝熔断的站。.		S (错误)	M9000	○+Rem
SM61	模块 I/O 核对错误	OFF ：正常 ON ：错误	• 当电源接通 ON 时实际的 I/O 模块与注册的信息之间有不一致时为 ON • I/O 模块核对可用于远程 I/O 工作站模块。		S (错误)	M9002	
SM62	信号报警器检测	OFF ：未检测到 ON ：检测到	• 一个信号报警器 F 接通时为 ON。.		S (指令执行)	M9009	
SM80	CHK 检测	OFF ：未检测到 ON ：检测到	• 当用 CHK 指令检测到错误时变为 ON。 • 即使恢复正常运行，仍停留在 ON。		S (指令执行)	新	
SM90	步传输警戒定时器启动（只有当 SFC 程序存在时才可能）	OFF ：未启动（警戒定时器复位） ON ：启动 (警戒定时器启动)	对应于 SD90	• 当步传输警戒定时器的测量开始时变为 ON •转为 OFF 时将警戒定时器复位.	U	M9108	○
SM91			对应于 SD91			M9109	
SM92			对应于 SD92			M9110	
SM93			对应于 SD93			M9111	
SM94			对应于 SD94			M9112	
SM95			对应于 SD95			M9113	
SM96			对应于 SD96			M9114	
SM97			对应于 SD97			新	
SM98			对应于 SD98			新	
SM99			对应于 SD99			新	

App - 2

特殊继电器表(续)

编号	名称	含义	说明	由谁设置(何时设置)	对应的ACPU M9□□□	可应用的CPU
SM120	外部电源断开OFF检测	OFF ：正常 ON ：有一模块的外部电源断开	・至少一个模块的外部电源OFF时变为ON，恢复正常运行后仍停留在ON * 仅可应用于Q系列模块（将来使用）	S（错误）	新	QCPU

特殊继电器表

(2) 系统信息

编号	名称	含义	说明	由谁设置(何时设置)	对应的ACPU M9□□□	可应用的CPU
SM202	指示灯OFF命令	OFF→ON：LED OFF	• 从OFF改变到ON，对应于SD202上各别位数的LED指示灯熄灭	U	新	○
SM203	停止触点	STOP状态	• 在停止状态时变为ON	S(状态改变)	M9042	
SM204	暂停触点	PAUSE状态	• 暂停状态时变为ON	S(状态改变)	M9041	
SM205	单步运行触点	STEP-RUN状态	• 在单步运行状态变为ON	S(状态改变)	M9054	
SM206	暂停作用线圈	OFF ：PAUSE不起作用 ON ：PAUSE起作用	• 当远程暂停触点为ON时，继电器为“ON”，进入暂入暂停状态	U	M9040	
	接受软元件测试请求状态	OFF ：尚未执行软元件测试 ON ：执行软元件测试	• 当GX-Develop执行软元件测试模式时为ON	S（请求）	新	远程
SM210	时钟数据设置请求	OFF ：忽略 ON ：设置请求	•这个继电器从OFF变为ON时，把执行更改扫描的END指令写入时钟软元件之后，存储从SD210到SD213的时钟数据。	U	M9025	○
SM211	时钟数据错误	OFF ：无错误 ON ：错误	• 当时钟数据发生错误时为ON（SD210-SD213）未检则到错误时为OFF。	S(请求)	M9026	
SM212	时钟数据显示	OFF ：忽略 ON ：显示	•在CPU面板上LED显示器上显示时钟数据：如月、日、时、分、秒. (只适用于Q3ACPU和Q4ACPU)	U	M9027	Q3A Q4A Q4AR
SM213	时钟数据读请求	OFF ：忽略 ON ：读请求	• 当这继电器ON时读数据到SD210~SD213，作为BCD码	U	M9028	○
SM240	CPU1复位标志	OFF ： PLC 1复位取消 ON ： PLC 1复位	・当PLC 1复位取消时为OFF ・当PLC 1复位时为ON（包括PLC从基板上取下时）其它的PLC仍处于复位状态。	S(状态改变)	新	QCPU“B”功能版本
SM241	CPU2复位标志	OFF ： PLC 2复位取消 ON ： PLC 2复位	・当PLC 2复位取消时为OFF ・当PLC 2复位时为ON（包括PLC从基板上取下时）其它的PLC的结果处于“多CPU宕机”（QAJ YKGDWA DCG 7000）。			
SM242	CPU3复位标志	OFF ： PLC 3复位取消 ON ： PLC 3复位	・当PLC 3复位取消时为OFF ・当PLC 3复位时为ON（包括PLC从基板上取下时）其它的PLC的结果处于“多CPU宕机”（QAJ YKGDWA DCG 7000）。			
SM243	CPU4复位标志	OFF ： PLC 4复位取消 ON ： PLC 4复位	・当PLC 4复位取消时为OFF ・当PLC 4复位时为ON（包括PLC从基板上取下时）其它的PLC的结果处于“多CPU宕机”（QAJ YKGDWA DCG 7000）。			
SM244	CPU1复位标志	OFF ： PLC 1复位取消 ON ： PLC 1复位	・OFF当PLC 1正常时（包括一个连续错误） ・ON当PLC 1在“STOP”错误期间			
SM245	CPU2复位标志	OFF ： PLC 2复位取消 ON ： PLC 2复位	・OFF当PLC 2正常时（包括一个连续错误） ・ON当PLC 2在“STOP”错误期间	S(状态改变)	新	QCPU“B”功能版本
SM246	CPU3复位标志	OFF ： PLC 3复位取消 ON ： PLC 3复位	・OFF当PLC 3正常时（包括一个连续错误） ・ON当PLC 3在“STOP”错误期间			
SM247	CPU4复位标志	OFF ： PLC 4复位取消 ON ： PLC 4复位	・OFF当PLC 4正常时（包括一个连续错误） ・ON当PLC 4在“STOP”错误期间			

特殊继电器表(续)

编号	名称	含义	说明	由谁设置(何时设置)	对应的ACPU M9□□□	可应用的CPU
SM250	最大装入I/O读取	OFF：忽略 ON：读	• 当该继电器从OFF切换到ON时，最大加载的I/O地址读出到SD250。	U	新	○+Rem
SM251	I/O改变标志	OFF：未更换 ON：更换	• 在正被更换的I/O模块的I/O起始地址设置在SD251，当这继电器在ON时可以对在线I/O模块更换(每次设定只能更换一个模块) • 在RUN状态更换I/O模块，使用程序或外围设备来使这继电器切换到ON，在STOP状态更换I/O模块，在外围设备的测试模式使这继电器切换到ON。 • 在I/O模块更换完成之前不要在RUN和STOP状态之间切换。	U (结束)	M9094	Q2A (S1) Q3A Q4A Q4AR
SM252	I/O改变OK	OFF：禁止更换 ON：可以更换	•当I/O更换是OK时为ON。	S (结束)	新	
SM254	全部工作站刷新命令	OFF：刷新到达的工作站 ON：刷新所有工作站	•批量刷新有效(对低速循环也有效) •指定只接收到达的工作站还是接收全部附属工作站。	U (每次结束)	新	QCPU
SM255	MELSECNET/10模块1信息	OFF：运行网络 ON：待机网络	•对待机网络(如果没有指定是活动的还是待机，则假定是活动的)为ON	S (初始)	新	○
SM256		OFF：读 ON：不读	• 对于从链接到CPU的刷新，指定是否要从通信模块读取。	U	新	
SM257		OFF：写 ON：不写	• 对于从CPU到链接（B，W等）的刷新指定是否要写入到通信模块。	U	新	
SM260	MELSECNET/10模块2信息	OFF：运行网络 ON：待机网络	•对待机网络(如果没有指定是活动的还是待机，则假定是活动的)为ON	S (初始)	新	
SM261		OFF：读 ON：不读	• 对于从链接到CPU的刷新（B，W等），指定是否要从通信模块读取。	U	新	
SM262		OFF：写 ON：不写	• 对于从CPU到链接（B，W等）的刷新指定是否要写入到通信模块。	U	新	
SM265	MELSECNET/10模块3信息	OFF：运行网络 ON：待机网络	•对待机网络（如果没有指定是活动的还是待机，则假定是活动的）为ON	S (初始)	新	
SM266		OFF：读 ON：不读	• 对于从链接到CPU的刷新，指定是否要从通信模块读取。	U	新	
SM267		OFF：写 ON：不写	• 对于从CPU到链接（B，W等）的刷新指定是否要写入到通信模块。	U	新	
SM270	MELSECNET/10模块4信息	OFF：运行网络 ON：待机网络	•对待机网络(如果没有指定是活动的还是待机，则假定是活动的)为ON。	S (初始)	新	
SM271		OFF：读取 ON：不读取	• 对于从链接到CPU的刷新（B，W等），指定是否要从通信模块读取。	U	新	
SM272		OFF：写 ON：不写	• 对于从CPU到链接（B，W等）的刷新指定是否要写入到通信模块。	U	新	
SM280	CC-Link错误	OFF：正常 ON：错误	• 当在安装的任何一个QJ61QBT11检测到CC-Link错误时OFF，当恢复正常运行时为ON。	S (状态改变)	新	QCPU
			•当在安装的任何一个A（1S）J61QBT11检测到CC-Link错误时为ON，恢复正常运行后仍停留在ON	S (Error)	新	QnA
SM320	SFC程序存在/不存在	OFF：SFC程序不存在 ON：SFC程序存在	• 如果SFC程序注册正确为ON。未注册为OFF •如果SFC专用指令不正确为OFF。	S (初始)	M9100	○
SM321	SFC程序开始/停止	OFF：SFC SFC程序停止 ON：SFC程序启动	•设置与SM320相同的初始值。(如果SFC程序存在，自动转到ON) •如果在SFC程序处理之前转为OFF，则不会执行SFC程序。 •当从OFF转到ON时，启动SFC程序。 •随后当从ON转到OFF时，停止SFC程序。	S (初始值) U	M9101 format change	

特殊继电器表(续)

编号	名称	含义	说明	由谁设置 (何时设置)	对应的 ACPU M9□□□	可应用的 CPU
SM322	SFC 程序开始状态	OFF ：初始化开始 ON ：重新起动	•根据参数初始值设置为 ON 或 OFF。 •当 OFF 时从 SFC 程序停止时开始，清除所有的程序的执行状态，从作出启动请求的程序块初始步开始启动 •当 ON 时，从执行程序块和 SFC 程序停止的时间开始启动。 （只有用参数指定可从新开始启动时 ON 才能起作用。） •不自动指定 SM902 为锁存。	S (初始) U	M9102 格式改变	○
SM323	整个数据块连续传输存在/不存在	OFF ：连续传输无效 ON ：连续传输有效	•当 OFF 时，所有的数据块发生一次扫描/步的传输。 •当 ON 交换发生连续地到所的块一次扫描。 •在指定个别块时，给块的连续交换位以优先权。 （当块程序开始时，检查指定）	U	M9103	
SM324	连续交换防止标志	OFF ：执行传输时 ON ：无传输时	•当连续交换有效时，当不执行连续交换时为 ON，当执行连续交换时 OFF。 •常规 ON，当连续交换无效时。	S (执行指令)	M9104	
SM325	程序块停止时的输出模式	OFF ：OFF ON ：保留	当块停止时，选择活动步骤运行输出 •当 OFF 时所有的线圈输出 OFF。 •当 ON 时线圈输出被保存。	S (初始) U	M9196	
SM326	SFC 软元件清除模式	OFF ：清除软元件 ON ：保留软元件	当停止的 CPU 在顺序程序之后运行时选择软元件的状态或当存在 SFC 程序时 SFC 程序已被修改。	U	新	
SM327	在执行 END 步骤期间输出	OFF ：OFF ON ：保留	当执行结束步来结束块程序时，选择正被处理这一步的输出动作。 •OFF 时所有的输出线圈 OFF。 •ON 时线圈输出被保存。	S (初始) U	新	
SM330	低速执行型程序的运行模式	OFF ：异步模式 ON ：同步模式	•异步模式 在过剩的时间期内，继续运行低速执行型程序的模式 •同步模式 即使有过剩的时间，低速执行型程序的运行将从下一次扫描开始的模式。	U (END)	新	
SM390	执行访问标志	ON 指出对智能功能模块访问的完成	•存储最后的智能功能模块访问指令的状态。 （这个信息被重写在智能功能模块访问指令下一次执行上） •用户使用这个标志在程序中作为一个完成的位数。	S (状态改变)	新	QCPU

特殊继电器表

(3) 系统时钟/计数器

编号	名称	含义	说明	由谁设置（何时设置）	对应的 ACPU M9□□□	可应用的 CPU
SM400	总是 ON	ON OFF	• 开	S (每次 END 处理)	M9036	
SM401	总是 OFF	ON OFF	• 闭	S (每次 END 处理)	M9037	
SM402	仅 RUN 以后的一次扫描时 ON	ON OFF 1 扫描	•仅运行以后一次扫描时 ON •这个连接仅可用于扫描执行型程序	S (每次 END 处理)	M9038	○
SM403	仅 RUN 以后一次扫描时 OFF	ON OFF 1 扫描	•仅运行以后仅一次扫描时 OFF •这个连接仅可用于扫描执行型程序	S (每次 END 处理)	M9039	
SM404	仅 RUN 以后一次扫描时 ON，低速执行型程序	ON OFF 1 扫描	•仅运行以后一次扫描时 ON •这个连接仅可用于低速执行型程序	S (每次 END 处理)	新	
SM405	仅 RUN 以后一次扫描时 OFF，低速执行型程序	ON OFF 1 扫描	•仅运行以后一次扫描 OFF •这个连接仅可用于低速执行型程序	S (每次 END 处理)	新	
SM409	0.01 秒时钟	0.005 sec. 0.005 sec.	•在 ON-OFF 之间以 5ms 间隔，重复地改变 •在电源切换到 OFF 或执行复位时，从 OFF 变为开始	S (状态改变)	新	QCPU
SM410	0.1 秒时钟	0.05 sec. 0.05 sec.	•以每一次指定的时间间隔，在 ON-OFF 之间重复改变。 •当电源切到 OFF，或执行复位时，从 OFF 转到开始 •注意：在程序执行期间，当指定的时间已过去时 ON-OFF 状态改变。	S (状态改变)	M9030	○
SM411	0.2 秒时钟	0.1sec. 0.1sec.			M9031	
SM412	1 秒时钟	0.5 sec. 0.5 sec.			M9032	
SM413	2 秒时钟	1 sec. 1 sec.			M9033	
SM414	2n 秒时钟	n sec. n sec.	•根据 SD415 所指定的秒数，在 ON、OFF 之间切换。	S (状态改变)	M9034 格式改变	
SM415	2n(ms)时钟	n(ms) n(ms)	•根据 SD415 所指定的毫秒数，在 ON、OFF 之间切换。	S (状态改变)	新	QCPU
SM420	用户定时时钟 0	n2 扫描 n2 扫描 n1 扫描	•以固定扫描间隔，继电器重复地进行 ON/OFF 切换。 •当电源切到 ON 或执行复位时，从 OFF 切换到开始。 • ON/OFF 的间隔用 DUTY 指令来设定。 DUTY n1 n2 SM420	S (每次 END 处理)	M9020	○
SM421	用户定时时钟 1				M9021	
SM422	用户定时时钟 2				M9022	
SM423	用户定时时钟 3				M9023	
SM424	用户定时时钟 4				M9024	
SM430	用户定时时钟 5		•供 SM420 到 SM424 的低速执行型程序使用。	S (每次 END 处理)	新	
SM431	用户定时时钟 6					
SM432	用户定时时钟 7					
SM433	用户定时时钟 8					
SM434	用户定时时钟 9					

App - 6

特殊继电器表

(4) 扫描信息

编号	名称	含义	说明	由谁设置（何时设置）	对应的ACPU M9□□□	可用的CPU
SM510	低速程序执行标志	OFF：完成或未执行 ON：正在执行中	•在执行低速执行型程序时切换到ON。	S (每次END处理)	新	○
SM551	读模块服务间隔	OFF：忽略 ON：读	•当此继电器从OFF切换到ON时由SD550指定的模块服务间隔被读到SD551-SD552。	U	新	○+Rem

(5) 存储卡

编号	名称	含义	说明	由谁设置（何时设置）	相对的ACPU M9□□□	可用的CPU
SM600	存储卡A可使用标志	OFF：不可使用 ON：可使用	• 当存储卡A作好使用准备时为ON	S (初始)	新	○
SM601	存储卡A保护标志	OFF：不保护 ON：保护	•当存储A的保护开关ON时变为ON	S (初始)	新	
SM602	驱动器1标志	OFF：无驱动器1 ON：无驱动器1存在	•当驱动器1（卡1RAM区）存在时变为ON	S (初始)	新	
SM603	驱动器2标志	OFF：无驱动器2 ON：无驱动器2存在	•当驱动器2（卡1RAM区）存在时变为ON	S (初始)	新	
SM604	存储卡A在使用中标志	OFF：不在使用 ON：在使用	•当存储卡A使用时变为ON	S (初始)	新	
SM605	存储卡A取下/插入禁止标志	OFF：可以取下/插入 ON：禁止取下/插入	•当存储卡A不能取出或插入时变为ON	U	新	
SM609	存储卡取下/插入标志	OFF：禁止取下/插入 ON：可以取下/插入	•由用户切换到ON，使可以取下/插入存储卡 • T在卡取下后由系统切换到OFF	U/S	新	
SM620	存储卡B可使用标志	OFF：不可使用 ON：可使用	• 始终ON	S (初始)	新	QCPU
			• 当用户从存储卡B作好使用准备为ON	S (初始)	新	Q2A (S1) Q3A Q4A Q4AR
SM621	存储卡B保护标志	OFF：不保护 ON：保护	•始终ON	S (初始)	新	QCPU
			•当存储B的保护开关ON时变为ON	S (初始)	新	Q2A (S1) Q3A Q4A Q4AR
SM622	驱动器3标志	OFF：无驱动器3 ON：无驱动器3存在	•始终ON	S (初始)	新	QCPU
			•当驱动器3（卡2RAM区）存在时变为ON	S (初始)	新	Q2A (S1) Q3A Q4A Q4AR
SM623	驱动器4标志	OFF：无驱动器4 ON：无驱动器4存在	•始终ON	S (初始)	新	QCPU
			•当驱动器4（卡2ROM区）存在时变为ON	S (初始)	新	Q2A (S1) Q3A Q4A Q4AR
SM624	存储卡B在使用中标志	OFF：不在使用 ON：在使用	•当存储卡B使用时变为ON	S (初始)	新	
SM625	存储卡B取下/插入禁止标志	OFF：可以取下/插入 ON：禁止取下/插入	•当存储卡A不能取下插入时变为ON	U	新	
SM640	文件寄存器使用	OFF：文件寄存器不在使用 ON：文件寄存器在使用	•当文件寄存器使用时变为ON	S (状态改变)	新	○
SM650	注释使用	OFF：文件寄存器不在使用 ON：文件寄存器在使用	•当注释文件使用时变为ON	S (状态改变)	新	
SM660	引导操作	OFF：内部存储器执行 ON：引导操作	•当引导操作在进行中时变为ON •引导指定开关在OFF时切换到OFF	S (状态改变)	新	
SM672	存储卡A文件寄存器访问范围标志	OFF：在访问范围之内 ON：在访问范围之外	•当访问存储卡A文件寄存器范围以外的区域时，变为ON（在END处理中设置） •用户程序时复位。	S/U	新	

App - 7

特殊继电器（续）

编号	名称	含义	说明	向谁设置（何时设置）	对应的ACPU M9□□□	可用的CPU
SM673	存储卡B文件寄存器访问范围标志	OFF ：在访问范围之内 ON ：在访问范围之外	•当访问存储卡A文件寄存器范围以外的区域时，变为ON（在END处理中设置） •用户程序时复位。	S/U	新	Q2A (S1) Q3A Q4A Q4AR

(6) 指令—相关特殊继电器

编号	名称	含义	说明	向谁设置（何时设置）	对应的ACPU M9□□□	可用的CPU
SM700	进位标志	OFF ：进位OFF ON ：进位ON	•在应用指令中使用的进位标志	S (执行指令)	M9012	○
SM701	输出字符数的选择	OFF ：16字符输出 ON ：输出直到NUL	•当SM701“OFF”时，输出ASCII码16位字符。 •当SM701“ON”时，进行输出直到遇到NUL（00H）码。	U	M9049	
SM702	搜索方法	OFF ：搜索下一个 ON ：2部分搜索	•指定搜索指令使用的方法。 •安排数据作二部分搜索。	U	新	
SM703	排序次序	OFF ：升序 ON ：降序	•使用排序指令来指定数据应当按降序分类还是按升序分类。	U	新	
SM704	块比较	OFF ：发现不相符 ON ：全部相符	•当所有数据条件符合BKCMP指令时，变换为ON	S (执行指令)	新	
SM707	选择实数指令处理类型	OFF ：定向速度 ON ：定向精度	•当SM707为OFF时，以高速处理实数指令。 •当SM707为ON时，以高精度处理实数指令。	U	新	Q4AR
SM710	CHK指令优先级排序标志	OFF ：条件优先级 ON ：图形优先级	•当OFF时，保持原始设定。 •当ON时，CHK优先更新。	S (执行指令)	新	○
SM711	分割传输状态	OFF ：在分割处理期以外的 ON ：在分割处理期内的	•在AD57（G1）处理中，当ON时，屏幕被分隔传送。OFF时，分割处理已完成。	S (执行指令)	M9065	QnA
SM712	选择传输处理	OFF ：批量传输 ON ：分割传输	•在AD57（G1）处理中，当底版屏被分割传送时。变为ON	S (执行指令)	M9066	
SM714	通信请求注册区BUSY讯号	OFF ：可以向远程终端模块请求通讯 ON ：不可向远程终端模块请求通讯	•用来决定是否执行与连接到AJ71PT32-S3的远程终端模块的通信请求。	S (执行指令)	M9081	
SM715	EI标志	0 ：在DI期间 1 ：在EI期间	•正在执行E1指令时为ON。	S (执行指令)	新	○
SM720	读注释完成标志	OFF ：未完成注释读出 ON ：已完成注释读出	•当COMRD或PRC指令完成时仅对一次扫描切换到ON	S (状态改变)	新	QCPU
SM721	正在访问的文件	OFF ：未访问文件 ON ：文件正被访问	•当文件正被S.FWRITE，S.FREAD COMRD或LEDC指令访问时切换到ON	S (状态改变)	新	
SM722	BIN/DBIN指令错误不起作用标志	OFF ：错误可使用 ON ：错误不可使用	•用BIN或DBIN指令制止“OPERATION ERROR”时切换到ON	U	新	
SM730	CC-Link通信请求注册区BUSY信号	OFF ：可以请求与智能软元件工作站通信 ON ：不可请求与智能软元件工作站通信	•用来决定与连接到A（1S）J61QBT11的智能软元件工作站的通信请求可使用或不可使用。	S (执行指令)	新	QnA

App - 8

特殊继电器（续）

编号	名称	含义	说明	向谁设置（何时设置）	对应的ACPU M9□□□	可用的CPU
SM736	PKEY 指令在执行中标志	OFF :不执行指令 ON :执行指令	• 当正在执行 PKEY 指令时为 ON，当输入 CR 或输入字符串达到 32 个字符时为 OFF	S（执行指令）	新	○
SM737	PKEY 指令接受键盘输入标志	OFF :可接收键盘输入 ON :不可接收键盘输入	• ON：当进行键盘输入时，OFF：当键盘输入存入 CPU 时	S（执行指令）	新	
SM738	MSG 指令接收标志	OFF :不执行指令 ON :执行指令	•执行 MSG 指令时，变为 ON	S（执行指令）	新	
SM774	PD 柔性处理	OFF :强制匹配 ON :不强制匹配	•在手动模式中，指定是否强制 SV 值与 PV 值匹配	U	新	
SM775	选择在执行 COM 指令期间链接刷新	OFF :执行链接刷新 ON :无链接刷新被执行	•在执行 COM 指令期间，只进行普通数据处理时，选择是否要执行链接刷新处理	U	新	
SM776	本地软元件可/不可调用	OFF :不可用本地软元件 ON :可用本地软元件	•决定在 CALLED 程序的 CALL 中本地软元件是否可使用	U（状态改变）	新	
SM777	在中断程序中可/不可使用本地软元件	OFF :不可用本地软元件 ON :可用本地软元件	•决定在执行中断程序时，是否可使用本地软元件	U（状态改变）	新	
SM780	可执行的 CC-Link 专用指令	OFF :可执行 CC-Link 专用指令 ON :不可执行 CC- Link 专用指令	•当可同时执行的 CC-Link 专用指令的数目达到 32 时，切换到 ON，当数目在 32 以下时，切换到 OFF	U（状态改变）	新	QnA

(7) 调试

编号	名称	含义	说明	向谁设置（何时设置）	对应的ACPU M9□□□	可用的CPU
SM800	跟踪准备	OFF :未准备 ON :准备好	•当完成跟踪准备时，切换到 ON	S（状态改变）	新	QCPU
	采样跟踪准备		•当采样跟踪已准备好，切换到 ON	S（状态改变）	新	QnA
SM801	跟踪开始	OFF :暂停 ON :开始	•当这个继电器切换到 ON 时，跟踪开始 •当这个继电器切换到 OFF，跟踪暂停（所有相关的特殊 M 切换到 OFF）	U	M9047	QCPU
	采样跟踪开始		•当这个继电器切到 ON 时，采样跟踪开始 •当这个继电器切到 OFF 时，暂停（相关的特殊 M 切换到 OFF）	U	M9047	QnA
SM802	跟踪在执行中	OFF :暂停 ON :开始	•在执行跟踪期间，切换到 ON	S（状态改变）	M9046	QCPU
	采样跟踪在执行中		•在执行采样跟踪期间，切换到 ON	S（状态改变）	M9046	QnA
SM803	跟踪触发	OFF →ON: 开始	•当这个继电器从 OFF 切换到 ON 时，触发跟踪（与跟踪指令执行状态一致）	U	M9044	QCPU
	采样跟踪触发		•当这个继电器从 OFF 切换到 ON 时，采样跟踪触发器变为 ON（与 STRA 指令执行状态一致）	U	M9044	QnA
SM804	跟踪触发以后	OFF :未在触发后 ON :在触发后	•跟踪触发后，切换到 ON	S（状态改变）	新	QCPU
	采样跟踪触发以后		•采样跟踪触发后，切换到 ON	S（状态改变）	新	QnA
SM805	跟踪完成	OFF :未完成 ON :结束	•跟踪完成时切换到 ON	S（状态改变）	9043	QCPU
	采样跟踪完成		•采样跟踪完成时切换到 ON	S（状态改变）	9043	QnA
SM806	状态锁存准备	OFF :未准备 ON :准备好	•状态锁存已准备好时变为 ON	S（状态改变）	新	QnA
SM807	状态锁存命令	OFF →ON: 开始	•运行状态锁存命令	U	新	
SM808	状态锁存完成	OFF :锁存未完成 ON :锁存完成	•当状态锁存完成切换到 ON	S（状态改变）	9055	
SM809	状态锁存清除	OFF →ON: 清除	•可以进行下一次状态锁存	U	新	

特殊继电器边缘（续）

编号	名称	含义	说明	向谁设置（何时设置）	相应的ACPU M9□□□	可用的CPU
SM810	程序跟踪准备	OFF :未准备好 ON :准备好	•当程序跟踪准备好时，变为ON	S (状态改变)	新	QnA
SM811	开始程序跟踪	OFF :暂停 ON :开始	•当切换到ON时，程序跟踪开始 •当切换到OFF时，暂停（相关特殊M，全切换到OFF）	S (状态改变)	新	
SM812	程序跟踪执行中	OFF :暂停 ON :开始	• 当程序跟踪在执行中时为ON	U	新	
SM813	程序跟踪触发	OFF →ON: 开始	•当这个继电器从OFF切换到ON，程序跟踪触发ON（与PTRA指令执行状态一致）	S (状态改变)	新	
SM814	程序跟踪触发以后	OFF :未在触发后 ON :触发后	•在程序跟踪触发后，变为ON	S (状态改变)	新	
SM815	程序跟踪完成	OFF :未完成 ON :结束	•程序跟踪完成时，变为ON	S (状态改变)	新	
SM820	步跟踪准备	OFF :未准备 ON :准备好	•程序跟踪注册以后，准备好时变为ON	U	新	○
SM821	步跟踪开始	OFF :暂停 ON :开始	• 当变为ON时步跟踪开始 •OFF时暂停（相关特殊M，全部OFF）	S (状态改变)	M9182 格式改变	
SM822	步跟踪执行中	OFF :暂停 ON :开始	• 步跟踪执行中时为ON • 完成或暂停时为ON	S (状态改变)	M9181	
SM823	步跟踪触发后	OFF :未在触发后 ON :第一次触发后	•在执行步跟踪期内，只要1块被触发变为ON •当步跟踪开始时，变为OFF	S (状态改变)	新	
SM824	触发后的步跟踪	OFF :未在所有触发后 ON :全部触发后	•在执行步跟踪期间，所有的块被触发时，变为ON •步跟踪开始时切换到OFF	S (状态改变)	新	
SM825	步跟踪完成	OFF :未完成 ON :结束	•步跟踪完成时变为ON •步跟踪开始时变为OFF	S (状态改变)	M9180	
SM826	跟踪错误	OFF :正常 ON :错误	•在跟踪执行期间发生错误：切换到ON	S (状态改变)	新	QCPU
	采样跟踪错误		•在采样跟踪执行期间发生错误：切换到ON	S (状态改变)	新	QnA
SM827	状态锁存错误	OFF :正常 ON :错误	•在状态锁存执行期间发生错误：切换到ON	S (状态改变)	新	
SM828	程序跟踪错误	OFF :正常 ON :错误	•在程序跟踪执行期间发生错误：切换到ON	S (状态改变)	新	

(8) 锁存区

编号	名称	含义	说明	向谁设置（何时设置）	对应的ACPU M9□□□	可用的CPU
SM900	断电文件	OFF :无断电文件 ON :有断电文件	•在访问期间电源中断如果有文件，切换到ON	S/U (状态改变)	新	QnA
SM910	RKEY注册标志	OFF :未注册键盘输入 ON :注册键盘输入	• 键盘输入注册时变为ON，键盘输入未注册时为OFF	S (执行指令)	新	○

(9) A 至 Q/QnA 转换的一致性

特殊继电器 SM1000 到 SM1255 是在 A 转换到 Q/Qn 以后相当于 ACPC 特殊继电器的继电器。

所有的这些特殊继电器都由系统控制，用户不能在程序中将它们进行 ON/OFF 切换。

如果用户想要将这些继电器进行 ON/OFF 切换，就应该修改程序以便使用 QCPU/QnACPU 特殊继电器。

但是，对 SM1084 和 SM1200 到 SM1255，如果用户能在转换之前进行特殊继电器 M9084 和 M9200 到 M9255 的 ON/OFF 切换，用户也能在转换后，在 SM1084 和 SM1200 到 SM1255 中对相应的继电器进行 ON/OFF 切换。

要了解 ACPU 特殊继电器的细节，参见用户手册中个别 CPU 和 MELSECNET 或 MELSECNET/B 数据通讯系统参考手册。

注意

当转换的特殊继电器与 QCPU 一起使用时，处理时间会延长。当不使用转换的特殊继电器时，在 GPPW 参数的 PC 系统设定内，不选择“A 系列 CPU 兼容性设置”。

备注

关于用于修改的特殊继电器这一列，下面进行补充说明：

①当提供用于修改的特殊继电器时，软元件编号应该改变为与所提供的 Q/QnACPU 特殊继电器相一致。

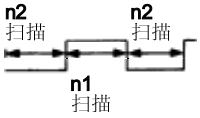
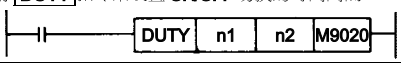
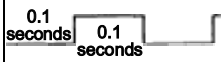
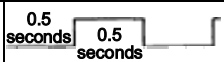
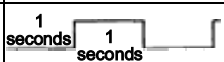
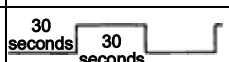
②当⊟时，转换的特殊继电器可以用作软元件的编号。

③当⊠时，软元件编号不能与 Q/QnACPU 一起使用。

特殊继电器表

ACPU 专用寄电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
M9000	SM1000	–	熔丝烧断	OFF : 正常 ON : 存在烧断了了熔丝的熔丝断模块	•当存在一个或几个熔丝已烧断的输出模块时，切换到 ON，如果正常状态恢复，仍保留在 ON。 同样检查远程 I/O 工作站的输出模块的熔丝状况。	○
M9002	SM1002	–	I/O 模块核查错误	OFF : 正常 ON : 错误	•如果当电源切换到 ON 时 I/O 模块的状态不同于输入的状态，将切换到 ON， 如果恢复正常状态，仍保留在 ON，对远程 I/O 站模块也要进行 I/O 模块的核查。 (只有当特殊继电器 SD1116 到 SD1123 复位时，才能复位)	
M9004	SM1004	–	MINI 链接错误	OFF : 正常 ON : 错误	•只要正在加载的 AJ71PT32（S3）模块中的一个被检测到 MINI（S3）链接错误，即切换到 ON，如果恢复正常状态，仍保留在 ON。	QnA
M9005	SM1005	–	交流电源断电检测	OFF : 未检测到断电 ON : 检测到断电	•在使用 AC 电源模块期间，如果发生少于 20ms 的瞬时电源中断即切换到 ON，要先开断电源再打开来使其复位。	○
					•在使用 DC 电源模块期间，如果发生少于 10ms 的瞬时电源中断即切换到 ON，要先开断电源到 OFF 再打开来使其复位。	QCPU
					•在使用 DC 电源模块期间，如果发生少于 1ms 的瞬时电源中断即切换到 ON，要先开断电源再打开来使其复位。	QnA

特殊继电器（续）

ACPU特殊寄电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的CPU
M9006	SM1006	–	电池低电压	OFF :正常 ON :电池低电压	•当电池电压降到指定值以下时，变为ON，电池电压正常后变为OFF。	○
M9007	SM1007	–	电池低电压锁存	OFF :正常 ON :电池低电压	•当电池电压降到指定值以下时，变为ON，电池电压正常后仍保持ON。	
M9008	SM1008	SM1	自诊断错误	OFF :无错误 ON :错误	•当自诊断结果发现错误时，变为ON。	
M9009	SM1009	SM62	信号报警器检测	OFF :未检测到F编号 ON :检测到F编号	• 执行 SET F 指令的 OUT F 时变为ON，当SD1124的数值复0时，变为OFF。	
M9011	SM1011	SM56	运行错误标志	OFF :无错误 ON :错误	•执行应用指令时发生运行错误，变为ON，恢复正常状态后，仍保持在ON。	
M9012	SM1012	SM700	进位标志	OFF :进位 OFF ON :进位 ON	•在应用指令中使用的进位标志。	
M9016	SM1016		数据存储器清除标志	OFF :忽略 ON :清除	•当SM1016为ON时，在远程运行模式中，从计算机来清除包括锁存范围（特殊继电器和特殊寄存器以外的）的数据存储器。	
M9017	SM1017		数据存储器清除标志	OFF :忽略 ON :输出清除	•当SM1017在ON时，远程运行模式中，从计算机来清除未锁存的数据存储器（特殊继电顺和特殊寄存器以外的）。	
M9020	SM1020	–	用户定时时钟1		•继电器以预先决定的扫描间隔，重复地在ON/OFF之间切换。 •当电源接通或执行复位时，时钟从OFF起动。 • 用 DUTY 指令来设置ON/OFF切换的时间间隔 	
M9021	SM1021	–	用户定时时钟2			
M9022	SM1022	–	用户定时时钟3			
M9023	SM1023	–	用户定时时钟4			
M9024	SM1024	–	时钟数据设置请求			
M9025	SM1025	–	时钟数据设置请求	OFF :忽略 ON :设置当前使用的请求	•在扫描期间SM1025W从OFF变成ON时执行 END 指令以后，时钟数据从SD1025到SD1028写入时钟元件。	
M9026	SM1026	–	时钟数据错误	OFF :无错误 ON :错误	•时钟数据（SD1025到SD1028）错误时变为ON。	○
M9027	SM1027	–	时钟数据显示	OFF :忽略 ON :显示	•时钟数据从SD1025读入SD128，同时，月、日、时、分、秒在CPU的面板显示器上显示出来。	
M9028	SM1028	–	请求读时钟数据	OFF :忽略 ON :读请求	•当SD1028ON时，读时钟数据到SD1025到SD1028（以BCD码）。	
M9029	SM1029		数据通信请求批处理	OFF :不进行批处理 ON :进行批处理	•在END扫描处理的一次扫描期间，使用顺控程序来处理所接受的所有数据通信请求。SM1029继电器变为ON。 •在运行期间，数据通信批处理的请求可以被变为ON或OFF。 •默认值是OFF（按接受数据通信请求的次序进行处理，一次只处理一个）。	
M9030	SM1030	–	0.1秒时钟		•产生0.1秒、0.2秒、1秒、2秒和1分的时钟。 •不是每次扫描而是在扫描期间如果相应的时间用完后变为ON/OFF。 •当电源变为ON或复位时，从OFF起动。	
M9031	SM1031	–	0.2秒时钟			
M9032	SM1032	–	1秒时钟			
M9033	SM1033	–	2秒时钟			
M9034	SM1034	–	总是ON			

特殊继电器表（续）

ACPU 特殊寄电器	转换后的特殊继电器	修改的特殊继电器	名称	含义	详细说明	可用的 CPU
M9036	SM1036	–	始终 ON	ON OFF	•用作在顺控程序中的初始化和应用指令的虚拟接触器 •不管 CPU 面板上的键开关的位置，SM1038 和 SM1037 在 ON/OFF 之间切换。除了当键开关在 STOP 位置时，SM1038 和 AM1039 处在与 RUN 状态相同的条件下，在 ON/OFF 之间切换。如果键开关在 STOP 位置则切换到 OFF。如果键开关不在 STOP 位置，SM1038 仅在一次扫描时间内“ON”，SM1039 则仅在一次扫描时间内“OFF”。	○
M9037	SM1037	–	始终 OFF	ON OFF		
M9038	SM1038	–	仅在运行后一次扫描期间“ON”	ON OFF 1 扫描		
M9039	SM1039	–	RUN 标志（在 RUN 以后，仅一次扫描期内 OFF）	ON OFF 1 扫描		
M9040	SM1040	SM206	PAUSE 作用线圈	OFF :暂停无效 ON :暂停有效	•当 RUN 键开关在 PAUSE 位置或远程 PAUSE 接触器已切换到 ON，如果 SM204 为 ON，即设置成 PAUSE 模式，SM206 变为 ON。-	
M9041	SM1041	SM204	USE 状态接触器	OFF :暂停无效 ON :暂停有效		
M9042	SM1042	SM203	STOP 状态接触器	OFF :停止无效 ON :停止有效	•当 RUN 键开关处于 STOP 位置时，切换到 ON。	
M9043	SM1043	SM805	采样跟踪完成	OFF :采样跟踪进行中 ON :采样跟踪完成	•在执行 STRA 指令后，按参数预设次数的采样跟踪完成时，切换到 ON。 当执行 STRAR 指令时复位。	
M9044	SM1044	SM803	采样跟踪	OFF →ON STRA 与执行相同 ON →OFF STRAR 与执行相同	•将 SM803 切换 ON/OFF 可执行 STRA / STRAR 指令。 （SM803 可由外围设备强制地切换 ON/OFF） 当从 OFF 切换到 ON： STRA 指令 当从 ON 切换到 OFF： STRAR 指令 存储在 SD1044 中的数值用作采样跟踪的条件 扫描时→Time（10ms 为单位）	
M9045	SM1045		警戒定时器（WDT）复位	OFF :不复位 ON :复位	•当执行 ZCOM 指令和数据通信请求批量处理时（当扫描时间超过 200ms 时使用），SM1015 继电器切换到 ON 来使 WDT 复位	
M9046	SM1046	SM802	采样跟踪	OFF :跟踪未进行 ON :跟踪进行中	• 在采样跟踪期间切换到 ON	
M9047	SM1047	SM801	采样跟踪准备	OFF :采样跟踪暂停 ON :采样跟踪开始	• 除非 SM801 换到 ON，不会执行采样跟踪。 • 当 SM801 “OFF” 时，采样跟踪停止。	
M9049	SM1049	SM701	选择字符输出数目	OFF :输出直到遇到 NULL 码 ON : 16 字符输出	• 当 SM701 在 OFF 时，直到 NUL 的字符（OOH）被输出 • 当 SM701 在 ON 时，输出 16 字符的 ASCⅡ码。	
M9051	SM1051		不可执行 CGH 指令	OFF :可执行 ON :不可执行	• 切换到 ON 不能执行 CHG 指令。 • 当请求程序转换时切换到 ON，转换完成时自动切换到 OFF。	
M9052	SM1052		SEG 指令开关	OFF : 7SEG 部分显示 ON : I/O 部分刷新	•当 SM1052 是 ON 时，SEG 指令作为 I/O 部分刷新指令来执行。 •当 SM1052 是 OFF 时 SEG 指令作为 7-SEG 显示指令来执行。	
M9054	SM1054	SM205	STEP GUN 标志	OFF :单步运行无效 ON :单步运行有效	•当 RUN 键开关处于 STEP RUN 位置时，切换到 ON。	QnA

M9055	SM1055	SM808	状态锁存 完成标志	OFF :未完成 ON :完成	•当状态锁存完成时切换到 ON，复位指令切换到 OFF。	
M9056	SM1056		主边 P，I 设置请求	OFF :请求设置 P，I 以外 ON :请求设置 P，I	•在 RUN 期间，其它程序（例如：在主程序运行时的子程序）完成以后，提供 P，I 设置请求。当 P，I 设定完成时，自动切换到 OFF。	○
M9057	SM1057		副边 P，I 设置请求	OFF: 请求设置 P，I 以外 ON :请求设置 P，I		
M9058	SM1058		主程序 P，I 设置完成	P，I 设置完成 瞬时 ON	•当 P，I 设置完成时“ON”，然后再次切回到 OFF。	
M9059	SM1059		子程序 P，I 设置完成	P，I 设置完成 瞬时 ON		
M9060	SM1060		子程序 2P，I 设置请求	OFF :请求设置 P,I 以外 ON :请求设置 P,I	•在 RUN 期间其它程序传输（例如：在主程序运行时的子程序）完成以后，提供 P，I 设置请求。当 P，I 设定完成时，自动切换到 OFF。	

App - 14

特殊继电器表（续）

ACPU 特殊寄电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
M9061	SM1061		子程序 3 P，I 设置请求	OFF : 请求 P，I 设置以外 ON : 请求 P，I 设置	•在 RUN 期间，其它程序（例如：主程序运行时的子程序）的传输完成以后，提供 P，I 设置请求。当 P，I 设定完成时，会自动地切换到 OFF。	○
M9065	SM1065	SM711	分割处理执行检测	OFF: 不进行分割处理时 ON : 分割处理时	•当由分割处理将调查屏幕传输到 AD57（S1）/AD58 时，切换到 ON，分割处理完成后切换到 OFF。	QnA
M9066	SM1066	SM712	分割处理请求标志	OFF : 批处理 ON : 分割处理	•当由分割处理将底版屏幕转换到 AD57（S1）/AD58 时，切换到 ON。	
M9070	SM1070		A8UPU/A8PUT 要求的搜索时间	OFF : 读时间不缩短 ON : 读时间不缩短	• 切换到 ON 来缩短在 A8UPU/A8PUJ 中的搜索时间。 (在这情况下，扫描时间大约延长 10%) ＊在 QCPU/QnCPU 特殊继电器中，不能使用 A8UPU/A8PUJ。	○
M9081	SM1081	SM714	通信请求注册区 BUSY 讯号	OFF : 通信请求注册区是空的 ON : 通信请求注册区无空闲	•对边接到 AJ71PT32-S3，A2C 或 A52G 的远程终端模块，可使用/不可使用通信的显示。	QnA
M9084	SM1084		错误检查	OFF : 执行错误检查 ON : 无错误检查	•这是设置：在处理 END 指令时（设置 END 指令处理时间）执行还是不执行以下的错误检查。 •检查熔丝的断路 • I/O 模块的核对检查 •电池检查	○
M9091	SM1091		指令错误标志	OFF : 无错误 ON : 错误	•当运行错误详细因素存储在 SD1091 时设置，恢复正常状态以后保持设置。 •在微机程序包执行中发生错误时设置，在恢复正常状态以后保持设置。	
M9094	SM1094	SM251	I/O 改变标志	OFF : 更换 ON : 无更换	•在需要的 I/O 模块的起始地址被设置在 SD251，将 SD251 切换到 ON，以便用在线模式更改 I/O 模块（只允许通过一个设定来改变一个模块）。 •在 CPU 运行期间。用程序或外围设备测试模式切换为 ON 以便更改模块。在 CPU 停止期间用外围设备测试模式中来更改模块。 •不得改变 RUN/STOP 模式直到 I/O 模块改变完成。	QnA
M9100	SM1100	SM320	SFC 程序存在/不存在	OFF : 不使用 SFC 程序 ON : 使用 SFC 程序	•如果 SFC 程序被注册，变为 ON，如果 SFC 程序未注册，变为 OFF。	○
M9101	SM1101	SM321	SFC 程序启动/停止	OFF : SFC 程序停止 ON : SFC 程序启动	•如果 SFC 程序启动，应该通过程序将该继电器切换到 ON。如果切换到 OFF，执行步的操作输出切换到 OFF，SFC 程序被停止。	
M9102	SM1102	SM322	SFC 程序启动状态	OFF : 初始启动 ON : 继续	•当使用 SM322 重新启动 SFC 程序时，选择起始步。 ON :当 SFC 程序停止时，所有执行条件被清除。 程序将从块“0”的初始步开始。 OFF :当程序停止时，从正在执行块的步开始。 •一旦切换到 ON，程序锁存在系统中，即使电源关断也保持 ON。 当接通电源以后，或从程序块 0 的初始开始时，应由顺控程序将此继电器切换到 OFF。	
M9103	SM1103	SM323	连续传输的存在/不存在	OFF : 连续传输无效 ON : 连续传输无效	•当连续步的所有转换条件都已建立，选择连续或一步一步的传输。 ON: 执行连续传输 OFF: 依次扫描一步地传输	

特殊继电器表（续）

ACPU 特殊寄电器	传输后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
M9104	SM1104	SM324	连续传输暂停标志	OFF : 传输完成 ON : 无传输	•可执行连续传输而未执行连续传输时设置，完成一步传输后复位。 将与条件写入相应的 M90104 来防止的连续传输。	○
M9108	SM1108	SM90	步传输、警戒定时器启动（与 D9108 等效）	OFF : 警戒定时器复位 ON : 警戒定时器复位启动	•当步传输警戒定时器启动时切换到 ON。当警戒定时器复位切换到 OFF。	
M9109	SM1109	SM91	步传输、警戒定时器启动（与 D9109 等效）			
M9110	SM1110	SM92	步传输、警戒定时器启动（与 D9110 等效）			
M9111	SM1111	SM93	步传输、警戒定时器启动（与 D9111 等效）			
M9112	SM1112	SM94	步传输、警戒定时器启动（与 D9112 等效）			
M9113	SM1113	SM95	步传输、警戒定时器启动（与 D9113 等效）			
M9114	SM1114	SM96	步传输、警戒定时器启动（与 D9114 等效）			
M9180	SM1180	SM825	活动步采样跟踪完成标志	OFF : 跟踪启动 ON : 跟踪完成	•当完成所有指定块的采样跟踪时设置。当采样跟踪开始时复位。	
M9181	SM1181	SM822	执行活动步采样跟踪标志	OFF : 未执行跟踪 ON : 跟踪执行中	•当采样跟踪正在执行时设置。 当采样跟踪完成或暂停时复位。	
M9182	SM1182	SM821	允许活动步采样跟踪	OFF : 不可跟踪/停止 ON : 可跟踪	•选择可/不可执行采样跟踪 ON :可以执行采样跟踪 OFF :不可以执行采样跟踪 如果在采样跟踪期间，将切换到 OFF，跟踪暂停。	
M9196	SM1196	SM325	块停止时的运行输出	OFF : 线圈输出 OFF ON : 线圈输出 ON	•当执行块停止时，选择运行输出。 ON :使用块停止时执行步的运行输出来保留正在使用的线圈的 ON/OFF 状态。 OFF:所有线圈输出切换到 OFF。（不管 M9196 的 ON/OFF 状态，SET 指令的运行输出被保留）。	
M9197	SM1197		在烧断熔丝与 I/O 核查出错是间切换显示	SM9197 / SM1198 / 显示的 I/O 地址 OFF / OFF / X/Y 0 - 7F0 ON / OFF / X/Y 800 - FF0 OFF / ON / X/Y 1000- 17F0 ON / ON / X/Y 1800 - 1FF0	按照 S1197 和 SM1198 的 ON/OFF 组合，切换熔丝烧断模块存储寄存器（SD1100 至 SD1107）中的 I/O 地址和 I/O 地址核查错误存储寄存器（SD1116 至 SD1123）。	
M9198	SM1198					
M9199	SM1199		在线采样跟踪/状态锁存的数据恢复	OFF : 不能恢复数据 ON : 能恢复数据	•当执行采样跟踪状态锁存时，在重新启动时恢复存储在 CPU 中的设定数据。 •再次执行时，SM1199 应该是 ON（当从外围设备再次写入数据时，不需要）。	

App - 15

特殊继电器表（续）

ACPU 特殊寄电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
M9200	SM1200	–	ZNRD 指令接收（主站）（LRDP 指令用于 ACPU）	OFF：不接受 ON ：接受	•取决于是否接收到 ZNRD 指令（字软元件读）。 •用在程序中作为用于 ZNRD 指令的互锁。 •用 RST 指令来复位。	QnA
M9201	SM1201	–	ZNRD 指令完成（主站）（LRDP 指令用于 ACPU）	OFF：未完成 ON ：结束	• 取决于执行 ZNRD 指令是否完成（字软元件读。 •作为一个条件接触器用于复位 M9200 和 M9201，在 ZNRD 指令完成以后。 •用 RST 指令来复位。	
M9202	SM1202	–	ZNWR 指令完成（主站）（LWTP 指令用于 ACPU）	OFF：不接受 ON ：接受	•取决于是否接收到 ZNWR 指令（字软元件读出）。 •用在程序中作为用于 ZNWR 指令的互锁 •用 RST 指令来复位。	
M9203	SM1203	–	ZNWR 指令完成（主站）（LWTP 指令用于 ACPU）	OFF：未完成 ON：结束	• 取决于执行 ZNRD 指令是否完成（字软元件读出。 •作为一个条件接触器用于复位 M9200 和 M9201，在 ZNRD 指令完成以后。 •用 RST 指令来复位。	
M9204	SM1204	–	ZNRD 指令完成（主站）（LWTP 指令用于 ACPU）	OFF：未完成 ON ：结束	ON：表明本地站已完成 ZNRD 指令。	
M9205	SM1205	–	ZNWR 指令完成（主站）（LRDP 指令用于 ACPU）	OFF：未完成 ON ：结束	ON：表明本地站已完成 ZNWR 指令。	
M9206	SM1206	–	主站链接参数错误	OFF：正常 ON ：不正常	取决于主站的链接参数设定是否有效。	
M9207	SM1207	–	链接参数检查结果	OFF：是 ON ：否	取决于在一个三层的系统中，主站的第二层的链接参数设定是否与第三层的链接参数设定相匹配。（只有当主站是一个三层系统时有效）。	
M9208	SM1208	–	设置主站 B 和 W 传输范围（只用于更低的链接主站）	OFF：传输到第二层和第三层 ON ：只传输到第二层	•取决于由更高的链接主站（主站）控制的 B 和 W 数据是否送到更低的链接本地站（第三个站）。 •当 SM1208 是 OFF..... 主站的 B 和 W 已送到第三站。 •当 SM1208 是 ON..... 主站的 B 和 W 未送到第三站。	
M9209	SM1209	–	链接参数检查命令（只用语低层链接主站）	OFF：执行检查功能 ON ：不执行检查	•设置在 ON：不匹配高层和低层链接的 B 和 W（当 SM1209 处于 ON 时，不检查高层和低层链接的链接参数。）。 •当 SM1209 OFF 时，检查高层和低层链接的链接参数	
M9210	SM1210	–	链接卡错误（供主站使用）	OFF：正常 ON ：不正常	取决于是否存在链接卡硬件错误（由 CPU 定）。	
M9211	SM1211	–	通信模块错误（供本地站使用）	OFF：正常 ON ：不正常	取决于是否存在链接卡硬件错误（由 CPU 定）。	
M9224	SM1224	–	链接状态	OFF：在线 ON ：离线，工作站-工作站测试或自回路测试	取决于主站在线还是离线，或是处于工作站-到工作站测试或自回测试。	
M9225	SM1225	–	正向环路错误	OFF：正常 ON ：不正常	取决于正向环路线的出错条件。	
M9226	SM1226	–	反向环路错误	OFF：正常 ON ：不正常	取决于反向环路线的出错条件。	
M9227	SM1227	–	环路测试状态	OFF：不在执行 ON ：正在执行正向和反向环路测试	取决于主站是否正在执行正向或反向环路测方式。	
M9232	SM1232	–	本地站运行状态	OFF：RUN 或 STEP RUN 状态 ON ：STOP 或 PAUSE 状态	取决于本地站是否处于 STOP 或 PAUSE 模式。	

App - 16

特殊继电器表（续）

ACPU 特殊寄电器	转换后的特殊继电器	用于修改的特殊继电器	名称	含义	详细说明	可用的 CPU
M9233	SM1233	–	本地站错误检测状态	OFF：无错误 ON ：出错检测	取决于本地站是否在另一个工作站检测到错误。	QnA
M9235	SM1235	–	本地站，远程 I/O 站参数出错检测状态	OFF：无错误 ON ：错误检测	取决于本地或远程 I/O 工作站是否检测到主站的任何链接参数错误。	
M9236	SM1236	–	本地站，远程 I/O 站初始化通讯状态	OFF：无通信 ON ：正进行通信	取决于本地或远程 I/O 工作站是否正在与主站通信初始化数据（如参数）。	
M9237	SM1237	–	本地站，远程 I/O 站出错	OFF：正常 ON ：不正常	取决于本地或远程 I/O 工作站的出错条件	
M9238	SM1238	–	本地站，远程 I/O 站正向或反向环路出错	OFF：正常 ON ：不正常	取决于本地或远程 I/O 工作站的正向和反向环路线的出错条件。	
M9240	SM1240	–	通讯状态	OFF：在线 ON ：离线，站间测 试或自回送测试	取决于本地站是在线还是离线。或处在站至站测试或回路反馈测试模式。	
M9241	SM1241	–	正向环路线路出错	OFF：正常 ON ：不正常	取决于正向环路线的出错条件。	
M9242	SM1242	–	反向环路线路出错	OFF：正常 ON ：不正常	取决于反向环路线的出错条件	
M9243	SM1243	–	回送安装	OFF：不进行环路反馈 ON ：环路实现	取决于在本地站是否发生环路反馈。	
M9246	SM1246	–	未收到的数据	OFF：接受 ON ：未接受	取决于就是否从主站接收到数据	
M9247	SM1247	–	未收到的数据	OFF：接受 ON ：未接受	取决于一个第三层工作站是否接收到从三层系统的主站来的数据。	
M9250	SM1250	–	未收到的参数	OFF：接受 ON ：未接受	取决于是否接收到从主站来的链接参数。	
M9251	SM1251	–	通讯继电器	OFF：正常 ON ：不正常	取决于本地站的数据通讯条件。	
M9252	SM1252	–	环路测试状态	OFF：不在执行 ON ：正常执行向前或反向环路测试	取决于本地站是否正在执行正向或反向环路测试。	
M9253	SM1253	–	主站操作状态	OFF：RUN 或 STEP RUN 状态 ON ：STOP 或 PAUSE 状态	取决于主站是否处于 STOP 或 PAUSE 模式。	
M9254	SM1254	–	本地站的而非上位站的操作状态	OFF：RUN 或 STEP RUN 状态 ON ：STOP 或 PAUSE 状态	取决于主站以外的本地站是否处于 STOP 或 PAUSE 模式。	
M9255	SM1255	–	本地站的而非上位站出错	OFF：正常 ON ：不正常	取决于主站以外的本地站是否存在错误。	

特殊继电器表

(10) 用于冗余系统（主系统 CPU 信息）仅用于 Q4AR

SM1510 至 SM1599 仅对冗余系统有效，对独立应用系统全部为 OFF

编号	名称	含义	说明	向谁设置（何时设置）	对应的 ACPU M9□□□	可用的 CPU
SM1500	保持模式	OFF: 不保持 ON ：保持	•当 S.IN 指令范围检查发生范围超越时，指出是否保持输出值。	U	新	Q4AR
SM1501	保持模式	OFF: 不保持 ON ：保持	•当 S.OUT 指令范围检查发生范围超越时，指出是否保持输出值。	U	新	
SM1510	运行模式	OFF: 冗余系统后备模式，独立系统 ON ：冗余系统分离模式	•当运行模式是冗余系统分离模式时切换到 ON。	S (每次 END)	新	
SM1511	电源接通时启动模式	OFF: 系统固定模式 ON ：先前控制系统锁存模式	•当电源开启，冗余系统的启动模式是先前控制系统锁存模式时，切换到 ON。	S (初始)	新	
SM1512	CPU 启动时的启动模式	OFF: 初始启动 ON ：热启动	•在启动冗余系统时，如果 CPU 的运行模式是热启动，切换到 ON。	S (初始)	新	
SM1513	CPU 启动时运行状态	OFF: 初始启动 ON ：热启动	•在冗余系统实际启动时，如果 CPU 的运行模式是热启动，切换到 ON。	S (初始)	新	
SM1514	CPU 切换时运行模式	OFF: 初始启动 ON ：热启动	• CPU 的运行模式切换到冗余系统时，如果 CPU 的运行模式是热启动，切换到 ON。	S (初始)	新	
SM1515	输出保持模式	OFF: 输出复位 ON :输出保持	•在停止错误期间输出模式是保持输出时，切换到 ON	S (每次 END)	新	
SM1516	操作系统状态	OFF: 控制系统 ON ：待机系统	•当 CPU 操作系统状态是待机系统时，切换到 ON。	S (状态改变)	新	
SM1517	CPU 启动状态	OFF: 电源 ON 启动 ON ：操作系统开关启动	•当 CPU 由操作系统开关启动时，切换到 ON。 • 用用户程序复位	S (状态改变) /U	新	
SM1518	跟踪执行模式	OFF: 批量运行模式 ON ：进位模式	•在 END 期间，如果跟踪存储器在使用中，切换到 OFF，执行待机状态直到可以执行时。 •在 END 期间，如果跟踪存储器正被使用。切换到 ON，这一步将被重复地执行直到下一次 END。	U	新	
SM1520	数据跟踪传输链接规格	OFF: 不触发 ON ：触发	SM1520 块 1 SM1521 块 2 SM1522 块 3 SM1523 块 4 SM1524 块 5 SM1525 块 6 SM1526 块 7 SM1527 块 8 SM1528 块 9 SM1529 块 10 SM1530 块 11 SM1531 块 12 SM1532 块 13 SM1533 块 14 SM1534 块 15 SM1535 块 16 SM1536 块 17 SM1537 块 18 SM1538 块 19 SM1539 块 20 SM1540 块 21 SM1541 块 22 SM1542 块 23 SM1543 块 24 SM1544 块 25 SM1545 块 26 •在使用数据跟踪指令 S.TRUCK 传输数据时，对作为对象的块指定具触发条件。	U	新	
SM1521						
SM1522						
SM1523						
SM1524						
SM1525						
SM1526						
SM1527						
SM1528						
SM1529						
SM1530						
SM1531						
SM1532						
SM1533						
SM1534						
SM1535						
SM1536						
SM1537						
SM1538						
SM1539						
SM1540						
SM1541						
SM1542						
SM1543						
SM1544						
SM1545						

App - 18

特殊继电器表（续）

编号	名称	含义	说明	向谁设置（何时设置）	ACPU M9□□□	可用的 CPU
SM1546	数据跟踪传输链接规格	OFF: 不触发 ON ：触发器	SM1546 块 27 SM1547 块 28 SM1548 块 29 SM1549 块 30 SM1550 块 31 SM1551 块 32 SM1552 块 33 SM1553 块 34 SM1554 块 35 SM1555 块 36 SM1556 块 37 SM1557 块 38 SM1558 块 39 SM1559 块 40 SM1560 块 41 SM1561 块 42 SM1562 块 43 SM1563 块 44 SM1564 块 45 SM1565 块 46 SM1566 块 47 SM1567 块 48 SM1568 块 49 SM1569 块 50 SM1570 块 51 SM1571 块 52 SM1572 块 53 SM1573 块 54 SM1574 块 55 SM1575 块 56 SM1576 块 57 SM1577 块 58 SM1578 块 59 SM1579 块 60 SM1580 块 61 SM1581 块 62 SM1582 块 63 SM1583 块 64 •当由数据跟踪指令 S.TRUCK 传输数据时，指定对应触发器的数据块。	U	新	Q4AR
SM1590	用网络模块切换状态	OFF: 正常 ON ：切换不成功	•当网络模块检测到网络错误和向主系统 CPU 发出切换请求时，如果不能正常执行时，切换到 ON。	S (错误发生)		

App - 19

特殊继电器表

(11) 对于冗余系统（其他系统 CPU 信息*1），仅对于 Q4AR，仅 SM1600 至 SM1650

对 CPU 冗余系统后备模式有效，因此在分离模式期间，不能被刷新。后备模式和分离模式对 SM4651 至 SM1699 有效。SM1600 至 SM1699 在独立应用系统中都切换到 OFF。

编号	名称	含义	说明	由谁设置（由谁设置）	ACPU M9□□□	可用的 CPU
SM1600	诊断错误	OFF: 无错误 ON ：错误	•如果在诊断结果中发生错误（包括外部诊断）切换到 ON。 •即使返回正常状态，仍保持为 ON。	S (每次 END)	SM0	Q4AR
SM1601	自诊断错误	OFF: 无自诊断错误 ON ：自诊断错误	•自诊断结果发生错误时，切换到 ON。 •即使返回正常状态，仍保持为 ON。	S (每次 END)	SM1	
SM1605	出错公用信息	OFF: 无出错公用信息 ON ：出错公用信息	•有出错公用信息时，切换到 ON，同时 SM1600 为 ON。	S (每次 END)	SM5	
SM1616	出错个别信息	OFF: 无出错个别信息 ON ：出错个别信息	•有出错个别信息时，切换到 ON，同时 SM1600 为 ON。	S (每次 END)	SM16	
SM1653	停止接触器	停止状态	・在停止状态时，切换到 ON。	S (每次 END)	SM203	
SM1654	暂停接触器	暂停状态	・在暂停状态时，切换到 ON。	S (每次 END)	SM204	
SM1655	单步运行接触器	步运行状态	・在步运行状态时，切换到 ON。	S (每次 END)	SM205	

＊1 存储其它系统 CPU 诊断信息和系统信息
＊2 这表示主机系统 CPU 的特殊继电器（SM□□）

(12) 对于冗余系统（跟踪），仅对于 Q4AR，

备用模式和秒模式都对 SM1700 至 SM1799 有效，在独立应用系统中全部切换为 OFF。

编号	名称	含义	说明	由谁设置（由谁设置）	ACPU M9□□□	可用的 CPU
SM1700	轨道执行标志	OFF: 不可执行 ON ：可能执行	•正常执行跟踪时切换到 ON。	S (状态改变)	新	Q4AR
SM1712	传输触发结束标志	OFF: 传输未完成 ON ：传输结束	SM1712 块 1 SM1713 块 2 SM1714 块 3 SM1715 块 4 SM1716 块 5 SM1717 块 6 SM1718 块 7 SM1719 块 8 SM1720 块 9 SM1721 块 10 SM1722 块 11 SM1723 块 12 SM1724 块 13 SM1725 块 14 SM1726 块 15 SM1727 块 16 SM1728 块 17 SM1729 块 18 SM1730 块 19 SM1731 块 20 SM1732 块 21 SM1733 块 22 • 当对应的数据传送已经完成时，一次扫描切换到 ON。	S (状态改变)	新	
SM1713						
SM1714						
SM1715						
SM1716						
SM1717						
SM1718						
SM1719						
SM1720						
SM1721						
SM1722						
SM1723						
SM1724						
SM1725						
SM1726						
SM1727						
SM1728						
SM1729						
SM1730						
SM1731						
SM1732						
SM1733						

App - 20

特殊继电器表（续）

编号	名称	含义	说明	由谁设置（由谁设置）	ACPU M9□□□	可用的 CPU
SM1734	传输触发结束标志	OFF: 传输未完成 ON ：传输结束	SM1734 块 23 SM1735 块 24 SM1736 块 25 SM1737 块 26 SM1738 块 27 SM1739 块 28 SM1740 块 29 SM1741 块 30 SM1742 块 31 SM1743 块 32 SM1744 块 33 SM1745 块 34 SM1746 块 35 SM1747 块 36 SM1748 块 37 SM1749 块 38 SM1750 块 39 SM1751 块 40 SM1752 块 41 SM1753 块 42 SM1754 块 43 SM1755 块 44 SM1756 块 45 SM1757 块 46 SM1758 块 47 SM1759 块 48 SM1760 块 49 SM1761 块 50 SM1762 块 51 SM1763 块 52 SM1764 块 53 SM1765 块 54 SM1766 块 55 SM1767 块 56 SM1768 块 57 SM1769 块 58 SM1770 块 59 SM1771 块 60 SM1772 块 61 SM1773 块 62 SM1774 块 63 SM1775 块 64 • 当对应的数据传送已经完成时，一次扫描切换到 ON。	S (状态改变)	新	Q4AR
SM1735						
SM1736						
SM1737						
SM1738						
SM1739						
SM1740						
SM1741						
SM1742						
SM1743						
SM1744						
SM1745						
SM1746						
SM1747						
SM1748						
SM1749						
SM1750						
SM1751						
SM1752						
SM1753						
SM1754						
SM1755						
SM1756						
SM1757						
SM1758						
SM1759						
SM1760						
SM1761						
SM1762						
SM1763						
SM1764						
SM1765						
SM1766						
SM1767						
SM1768						
SM1769						
SM1770						
SM1771						
SM1772						
SM1773						
SM1774						
SM1775						

App - 21

附录 2 特殊寄存器表

特殊寄存器 SD，是在可编程控制器中有固定用途的内部寄存器。
根据这点，在顺控程序中不能按使用通常寄存器的方法来使用这些寄存器。
然而，可以为了控制 CPU 和远程 I/O 模块的需要而写入数据。如果没有做出相反的特别指定，存储在特殊寄存器中的数据二进制值。

下表中的标题有下列含义。

项目	项目的功能
编号	· 表示特殊寄存器的编号
名称	· 表示特殊寄存器的名称
含义	· 表示特殊寄存器的内容
说明	更详细地讨论特殊积存器的内容
由谁设置 （何时设置）	· 指出继电器是由系统还是由用户来设置，和如果由系统设置，在执行设定时 <由谁设置> S ：由系统设置 U ：由用户设置（顺控程序或由外围设备测试操作） S/U ：由系统和用户设置 <何时设置>→仅表示由系统对寄存器设置 每次 END ：在每一次 END 处理期间设置 初始化 ：在初始化处理期间设置（当电源开启或者从 STOP 切换到 RUN 时） 状态改变 ：当状态发生改变时设置 错误 ：错误发生时设置 执行指令 ：执行指令时设置 请求 ：仅当用户请求时设置（通过 SM，等）
对应的 ACPU M9□□□	· 表示 ACPUD9□□□中对应的特殊寄存器（当内容改变时更改并记录） · 作为“新”项目已经被新加入到 QnACPU 中
相应的 CPU	· 指出相应的 CPU 的型号名称 O+Rem：可应用于所有 CPU 类型和 MELSECNET/H 远程 I/O 模块 O：可应用于所有类型的 CPU QCPU：可应用于 Q 系列 CPU QnA：可应用于 QnA 系列和 Q2ASCPU 系列 远程：可应用于 MELSECNET/H 远程 I/O 模块 每一个 CPU 类型名称：只能应用于特殊的 CPU（如 Q4ARCPU、Q3ACPU 等）

有关下列项目的细节参见这些手册：

•网络 → • Q MELSECNET/H 网络系统参考手册（PLC 到 PLC 网络）
• Q MELSECNET/H 网络系统参考手册（远程 I/O 网络）
• MELSECNET/10 网络系统参考手册 QnA

• SFC → QCPU（Q 模式）/QnACPU 编程手册（SFC）

要点

(1) SD1200 到 SD1255 用于 QnACPU。
对 QCPU，这些寄存器是空的。
(2) 特殊寄存器 SD1500 和其后的编号是专用于 Q4ARCPU。

特殊寄存器表

(1) 诊断信息

编号	名称	含义	解释	由谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD0	诊断错误	诊断出错代码	•出错代码用于诊断所发现的错误。用二进制数据存储 •内容与最近出错的记录信息一致。	S（错误）	D9008 格式改变	+Rem
SD1	发生诊断错误的时钟时间	发生诊断错误的时钟时间	•年（后二位数字）和月存储更新的 SD0 数据，用 BCD 二位数字代码。 B15 到 B8 B7 到 B0 年(0-99) | 月(1-12) (示例) ：十月，1995 H9510	S（错误）	新	
SD2			•用 BCD 二位数字代码存储的更新 SD0 的日期和小时。 B15 到 B8 B7 到 B0 天(1-31) | 小时(0-23) (示例) ：10 25日 H2510			
SD3			•用 BCD 二位数字代码存储更新的 SD0 分和秒 B15 到 B8 B7 到 B0 分(0-59) | 秒(0-59) (示例) ：35 分 48 秒 (点钟后) H3548			
SD4	错误信息类别	错误信息类别	•类别代码帮助显示在公用信息（SD5—SD15）和个别信息区（SD15—SD26）存储了哪类信息，存储在哪里。 B15 到 B8 B7 到 B0 特殊信息分类码 | 通用信息分类码 •存储的通用信息分类码有以下内容： 0：无错误 1：模块/模块号/PLC 号* 2：文件名/驱动器名 3：时间（设定值） 4：程序错误位置 5：切换原因（仅适用于 Q4AR） *：对多 CPU 系统来说，存储的模块号或 PLC 号取决于所发生的错误（要知道存储号，参照相应的出错代码） PLC 号：1：1，PLC 号：2：2， PLC 号：3：3，PLC 号：4：4 •存储的个别信息分类码有以下内容： 0：无错误 1：（Open） 2：文件名/驱动器名 3：时间（设置值） 4：程序出错位置 5：参数编号 6：信号报警器编号 7：检查指令故障编号	S（错误）	新	

App - 23

特殊寄存器表（续）

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>由谁设置（何时设置）</th><th>对应的 ACPU D9□□□</th><th>对应的 CPU</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>出错公共信息</td><td>出错公共信息</td><td>•这里存储对应于出错代码（SD0）的公用信息
•这里存储下列四种类型的信息

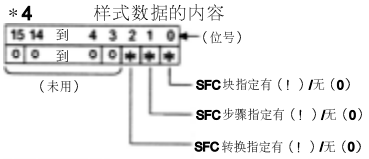
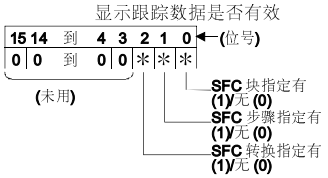
①插槽号
<table>
<tr><th>编号</th><th>含义</th></tr>
<tr><td>SD5</td><td>插槽序号
/PLC*1*2</td></tr>
<tr><td>SD6</td><td>I/O 号</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>（空）</td></tr>
</table>
*1：多 CPU 系统，插槽编号或 PLC 编号是依照发生的错误来存储的。
插槽 0 是多 CPU 系统中最右边 CPU 的右面插槽。
（参考相应的已存入编号的出错代码）
PLC：1：1，PLC：2：2，
PLC：3：3，PLC：4：4
*2：如果 MELSECNET/H 远程 I/O 站加载的模块发生熔丝烧断或 I/O 核对错误，那末网络编号存储在上 8 位中，而站编号则存储在低 8 位中。
用 I/O 地址来检查熔丝烧断或发生 I/O 核对错误的模块。
②文件名/驱动器名
（示例）
文件名=
ABCDEFGH.IJK
B15-B8 B7-B0
<table>
<tr><td>42H(B)</td><td>41H(A)</td></tr>
<tr><td>44H(D)</td><td>43H(C)</td></tr>
<tr><td>46H(F)</td><td>45H(E)</td></tr>
<tr><td>48H(H)</td><td>47H(G)</td></tr>
<tr><td>49H(I)</td><td>2DH(.)</td></tr>
<tr><td>4BH(K)</td><td>4AH(B)</td></tr>
</table>

<table>
<tr><th>编号</th><th>含义</th></tr>
<tr><td>SD5</td><td>驱动器</td></tr>
<tr><td>SD6
SD7
SD8
SD9</td><td>文件名
（ASCII 码：8 个字符）</td></tr>
<tr><td>SD10</td><td>扩展 *3 2EH（.）</td></tr>
<tr><td>SD11</td><td>（ASCII 码：3 个字符）</td></tr>
<tr><td>SD12
SD13
SD14
SD15</td><td>（空）</td></tr>
</table></td><td>S
(错误)</td><td>新</td><td>+Rem</td></tr>
</table>

＊3：参见备注

App - 24

特殊寄存器表（续）

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>由谁设置
（何时设置）</th><th>对应的
ACPU
D9□□□</th><th>对应的
CPU</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>出错公共信息</td><td>出错公共信息</td><td>③ 时间（设定值）
<table><tr><th>编号</th><th>含义</th></tr><tr><td>SD5</td><td>时间 1μs（0-999μs）</td></tr><tr><td>SD6</td><td>时间 1ms（0-65535ms）</td></tr><tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD12
SD13
SD14
SD15</td><td>（空）</td></tr></table>④ 程序错误位置
<table><tr><th>编号</th><th>含义</th></tr><tr><td>SD5
SD6
SD7
SD8</td><td>文件名
（ASCII 码：8 个字符）</td></tr><tr><td>SD9</td><td>扩展名＊3 2EH（.）</td></tr><tr><td>SD10</td><td>（ASCII 码：3 个字符）</td></tr><tr><td>SD11</td><td>样式＊4</td></tr><tr><td>SD12</td><td>块号</td></tr><tr><td>SD13</td><td>步骤号/转换号</td></tr><tr><td>SD14</td><td>序列步骤号（L）</td></tr><tr><td>SD15</td><td>序列步骤号（H）</td></tr></table>＊4 样式数据的内容
</td><td>S（出错）</td><td>新</td><td>+Rem</td></tr>
<tr><td>SD15</td><td></td><td></td><td>⑤ 切换原因
<table><tr><th>编号</th><th>含义</th></tr><tr><td>SD5</td><td>切换原因（0：自动切换
/1：手动切换）</td></tr><tr><td>SD6</td><td>切换方向（0：备用系统到
控制系统/1：控制系统到
备用系统）</td></tr><tr><td>SD7</td><td>跟踪标志＊5</td></tr><tr><td>SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>（空）</td></tr></table>＊5 跟踪标志内容
显示跟踪数据是否有效
</td><td>S（出错）</td><td>新</td><td>+Rem</td></tr>
</table>

*3：参见备注

特殊寄存器表（续）

编号	名称	含义	解释	向谁设置（何时设置）	对应的ACPU D9□□□	对应的CPU
SD16 SD17 SD18 SD19 SD20 SD21 SD22 SD23 SD24 SD25 SD26	出错个别信息	出错个别信息	•该寄存器存储与出错代码（SD0）相对应的出错个别信息 ① 文件名/驱动器名 ② 时间（实际测定值） ③ * ④ 参数号 ⑤ 信号报警器号/CHK指令失效号 ⑥ 智能功能模块参数出错（只适于QCPU） *3 参数号的详情请见所用CPU的用户手册。	S	新	○+Rem

App - 26

特殊寄存器表（续）

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD50	出错复位	执行出错复位的出错编号	•存储执行出错复位的出错编号。	U	新	
SD51	电池低电压锁存	表示发生电池低电压地点的位样图	•当电池电压低落时，所有相应的位切换到 ON。 •其后，即使电压回复正常仍保留在 ON。 15 14 到 4 3 2 1 0 ←(位号) 0 0 到 0 0 * * * (未用) 无效的工作数据 无效 (0)/有效(1) SFC 活动步信息□ 无效 (0)/有效(1) 切换原因 无效 (0)/有效(1) •当使用 QCPU 时，这个标志总是 OFF，因为存储卡 B 用作标准存储卡。	S（错误）	新	
SD52	电池低电压	表示发生电池低电压地点的位样图	•与以上的 SD51 相同配置。 •当电池电压恢复正常后，切换到 OFF。 •当使用 QCPU 时，这个标志总是 OFF，因为存储卡 B 用作标准存储卡。	S（错误）	新	
SD53	交流断电检测	交流断电次数	在使用交流电源模块时，发生 20ms 以内的瞬时失电加“1”。（以二进制代码存储这个数值） 当电源从 OFF 切换到 ON 时，复位。	S（错误）	D9005	
			在使用交流电源模块时，发生 10ms 以内的瞬时失电加“1”。以二进制代码存储这个数值 当电源从 OFF 切换到 ON 时，复位。			QCPU
			在使用交流电源模块时，发生 1ms 以内的瞬时失电加“1”。（以二进制代码存储这个数值） 当电源从 OFF 切换到 ON 时，复位。			QnA
SD54	MINI 链接错误	出错检测状况	①当安装的任何一个 MINI(-S3) X(n+0)/X(n+20), X(n+6)/X(n+26) ,X(n+7)/X(n+27)或 X(n+8)/X(n+28)状态是 ON 时，相关站的对应“位”被置为 1。 ②当安装的 MINI(-S3)与 CPU 不能进行通讯时，变为 ON。 B15 B9 B8 B0 8号模块 ..... 1号模块 8号模块 1号模块 有关 ②的信息 有关 ①的信息	S（错误）	D9004 格式改变	QnA
SD60	烧断熔丝模块的编号	烧断熔丝模块的编号	•存储在这里的是熔丝烧断模块的最低工作站的 I/O 地址	S（错误）	D9000	
SD61	I/O 模块核查出错编号	I/O 块核查出错编号	•发生 I/O 模块核查编号的模块的最低 I/O 地址。	S（错误）	D9002	
SD62	信号报警器编号	信号报警器编号	•检测到的第一个信号报警器的编号存储在这里。	S（指令执行）	D9009	
SD63	信号报警器的数目	信号报警器的数目	•存储搜索到的信号报警器的数目。	S（指令执行）	D9124	

备注

1）扩展如下表：

SD10	SD11		扩展名	文件类型
高 8 位	低 8 位	高 8 位		
51H	50H	41H	QPA	参数
51H	50H	47H	QPG	顺控程序/SFC 程序
51H	43H	44H	QCD	软元件注解
51H	44H	49H	QDI	软元件初始值
51H	44H	52H	QDR	文件寄存器
51H	44H	53H	QDS	仿真数据
51H	44H	4CH	QDL	本地软元件
51H	54H	53H	QTS	采样跟踪数据（用于 QnA）
51H	54H	4CH	QTL	状态锁存数据（用于 QnA）
51H	54H	50H	QTP	程序跟踪数据（用于 QnA）
51H	54H	52H	QTR	SFC 跟踪文件
51H	46H	44H	QFD	故障历史记录数据

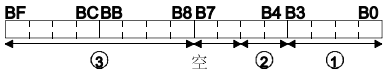
特殊寄存器表（续）

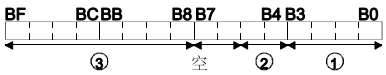
<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>向谁设置（何时设置）</th><th>对应的 ACPU D9□□□</th><th>对应的 CPU</th></tr>
<tr><td>SD64</td><td rowspan="16">信号报警器检测编号表</td><td rowspan="16">信号报警器检测编号</td><td rowspan="16">在因为 OUT F 或 SET F 使 F 切换到 ON 时，从 SD64 至 SD79 逐个变为“ON”的 F 号被注册。
因为 RST F 切换到 OFF 的 F 编号被从 SD64 到 SD79 删去，并被移位到存储删除的 F 编号的数据寄存器后面的数据寄存器。
LED R 指令的执行将 SD64 到 SD79 的内容向上移一位。
（这也可以使用 Q3A/Q4ACP 的 CPU 面板上的 INDICATOR RESET 开关来做到）
在检测到 16 个信号报警器以后，第 17 个的检测，将不存放在 SD64 到 SD79。
设置 设置 设置 设置 设置 设置 设置 设置 设置 设置 设置 设置 设置 设置 设置 设置
F50 F25 F99 F25 F15 F70 F65 F38 F110 F151 F210 LEDR
SD62 0 50 50 50 50 50 50 50 50 50 50 50 99 ...(检测号)
SD63 0 1 2 3 2 3 4 5 6 7 8 9 8 ...(检测到的错误记号)
SD64 0 50 50 50 50 50 50 50 50 50 50 50 99
SD65 0 0 25 25 99 99 99 99 99 99 99 99 15
SD66 0 0 0 99 0 15 15 15 15 15 15 15 70
SD67 0 0 0 0 0 0 70 70 70 70 70 70 65
SD68 0 0 0 0 0 0 0 65 65 65 65 65 38
SD69 0 0 0 0 0 0 0 0 38 38 38 38 110
SD70 0 0 0 0 0 0 0 0 0 110 110 110 151
SD71 0 0 0 0 0 0 0 0 0 0 151 151 210
SD72 0 0 0 0 0 0 0 0 0 0 0 210 0
SD73 0 0 0 0 0 0 0 0 0 0 0 0 0
SD74 0 0 0 0 0 0 0 0 0 0 0 0 0
SD75 0 0 0 0 0 0 0 0 0 0 0 0 0
SD76 0 0 0 0 0 0 0 0 0 0 0 0 0
SD77 0 0 0 0 0 0 0 0 0 0 0 0 0
SD78 0 0 0 0 0 0 0 0 0 0 0 0 0
SD79 0 0 0 0 0 0 0 0 0 0 0 0 0
(检测号)</td><td rowspan="16">S（指令执行）</td><td>D9125</td><td rowspan="16"></td></tr>
<tr><td>SD65</td><td>D9126</td></tr>
<tr><td>SD66</td><td>D9127</td></tr>
<tr><td>SD67</td><td>D9128</td></tr>
<tr><td>SD68</td><td>D9129</td></tr>
<tr><td>SD69</td><td>D9130</td></tr>
<tr><td>SD70</td><td>D9131</td></tr>
<tr><td>SD71</td><td>D9132</td></tr>
<tr><td>SD72</td><td>新</td></tr>
<tr><td>SD73</td><td>新</td></tr>
<tr><td>SD74</td><td>新</td></tr>
<tr><td>SD75</td><td>新</td></tr>
<tr><td>SD76</td><td>新</td></tr>
<tr><td>SD77</td><td>新</td></tr>
<tr><td>SD78</td><td>新</td></tr>
<tr><td>SD79</td><td>新</td></tr>
<tr><td>SD80</td><td>CHK 指令编号</td><td>CHK 指令编号</td><td>•由 CHK 指令检测到的出错代码用 BCD 代码存储。</td><td>S（指令执行）</td><td>新</td><td></td></tr>
<tr><td>SD90</td><td rowspan="10">步传输警戒定时器设定值（仅在 SFC 程序存在时起作用）</td><td rowspan="10">定时器设定值的 F 编号和时间用完错误</td><td>对应于 SM90</td><td rowspan="10">•在步传输监视设定时器设定值和警戒定时器超时错误时切换到 ON 的 F 编号。
B15 B8 B7 B0
F 编号设置 (0-255)
定时器时间极限值 (1-255秒: (1秒单位))
•在活动步期间，通过将 SM90 至 SM99 切换到 ON 来启动定时器。如果在定时器时限内未满足相关步的传输条件，指定的信号报警器（F）将切换到 ON。</td><td rowspan="10">U</td><td>D9108</td><td rowspan="10"></td></tr>
<tr><td>SD91</td><td>对应于 SM91</td><td>D9109</td></tr>
<tr><td>SD92</td><td>对应于 SM92</td><td>D9110</td></tr>
<tr><td>SD93</td><td>对应于 SM93</td><td>D9111</td></tr>
<tr><td>SD94</td><td>对应于 SM94</td><td>D9112</td></tr>
<tr><td>SD95</td><td>对应于 SM95</td><td>D9113</td></tr>
<tr><td>SD96</td><td>对应于 SM96</td><td>D9114</td></tr>
<tr><td>SD97</td><td>对应于 SM97</td><td>新</td></tr>
<tr><td>SD98</td><td>对应于 SM98</td><td>新</td></tr>
<tr><td>SD99</td><td>对应于 SM99</td><td>新</td></tr>
<tr><td>SD105</td><td>CH1 传输速度设定(RS-232)</td><td>在使用 GX Developer 时，存储预先设定的传输速度值。</td><td colspan="2">K3：300bps，K6：600bps，K24：2400bps，K48：4800bps，K96：9600bps，K192：19.2kbps，K384：38.4kbps，K576：57.6kbps，K1152：115.2kbps</td><td>S</td><td>新</td><td>远程 QCPU</td></tr>
</table>

特殊寄存器表

（2）系统信息

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>向谁设置
（何时设置）</th><th>对应的
ACPU
D9□□□</th><th>对应的
CPU</th></tr>
<tr><td>SD120</td><td>外接电源 OFF 时的出错编号</td><td>有外接电源错误的模块编号</td><td>•存储发生外部电源断电错误模块的最小起始地址号。
*仅适用于 Q 系列模块（作为将来使用）。</td><td>S（错误）</td><td>新</td><td>QCPU</td></tr>
<tr><td rowspan="3">SD200</td><td rowspan="3">开关的状态</td><td rowspan="3">CPU 开关的状态</td><td>•远程 I/O 模块的 ON/OFF 状态是以下列格式存储的。

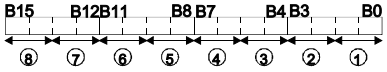
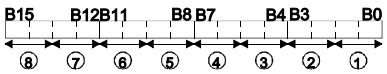
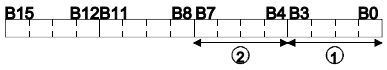

①远程 I/O 模块开关状态　总是 1：STOP</td><td>S（始终）</td><td>新</td><td>远程</td></tr>
<tr><td>• CPUON/OFF 状态以下列格式存储。


①：CPU 开关状态　0：RUN 1: STOP 2: L.CLR
②：存储卡开关　始终“关”
③：DIP 开关　B8-BC 与系统设置开关 1 的 SW1-SW5 相对应。0：OFF，1：ON BD-BF 空闲。</td><td>S
(每次 END 处理)</td><td>新</td><td>QCPU</td></tr>
<tr><td>• CPU 开关状态以下列格式存储。


①：CPU 开关状态　1: STOP 2: L.CLR 0：RUN
②：存储卡开　B4 对应于存储卡 A，B5 对应于存储卡 B 在 0 时断开；在 1 时接通
③：DIP 开关　B8-B12 与系统设置开关 1 的 SW1-SW5 相对应。B14 和 B15 分别对应于系统设置开关 2 的 SW1 和 SW2。在 0 时断开；在 1 时接通</td><td>S
(每次 END 处理)</td><td>新</td><td>QnA</td></tr>
</table>

App - 29

特殊寄存器表

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD201	LED 状态	CPU LED 状态	•下列位置是用来存储 CPU 上的 LED 的状态  ①：RUN ⑤:BOOT ②：ERROR ⑥：空 ③：USER ⑦：空 ④：BAT.ALARM ⑧：模式 模式的位样图 0：熄灭 1：绿色 2：桔黄色	S（状态改变）	新	QCPU
			•与 CPU 上的 LED 的下列状态相关的信息存储在下列位置中。 •1 表示熄灭；0 表示没有熄灭。  ①：RUN ⑤:BOOT ②：ERROR ⑥：CARD A (存储卡) ③：USER ⑦：CARD B (存储卡) ④：BAT.ALARM ⑧：空	S（状态改变）	新	QnA
SD202	LED 熄灭	熄灭的 LED 的位样图	•存储转到 OFF 的 LED 的位样图（仅 USER 和 BOOT 可使用）。 •转到 OFF 为 1：不转到 OFF 为 0。	U	新	QnA
SD203	CPU 运行状态	CPU 运行状态	•远程 I/O 模块的运行状态以下列格式存储  ①远程 I/O 模块运行状态 始终 2：STOP	S（始终）	新	远程
			• CPU 运行状态以下置所指示的方式存储。：  ①：CPU 的运行状态 0：运行 1：单步运行 2：停止 3：暂停 ②：STOP / PAUSE 原因 0：键开关 1：:远程接触器 2：外围设备、计算机数据通讯设备或从其它远程操作源 3：内部顺控指令 4：错误 注意：优先级是最早的一个。	S (每次 END 处理)	D9015 格式改变	
SD206	执行软元件测试的型式	0：未执行测试 1：在 X 期间测试软元件 2：在 Y 期间测试软元件 3：在 X/Y 期间测试软元件	•当在 GX Developer 上执行软元件测试模式时设置。	S（请求）	新	远程

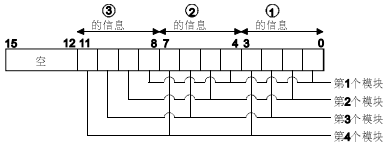
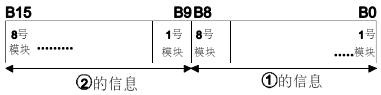
特殊寄存器列表(续)

编号	名称	含义	解释	向谁设置（何时设置）	相应的 ACPU D9□□□	相应的 CPU
SD207	LED 显示优先级	优先级 1 到 4	•发生错误时，按照出错编号设定的优先级使 LED 显示（闪烁） •优先级的设定区如下： B15 B12 B11 B8 B7 B4 B3 B0 SD207 优先级4 优先级3 优先级2 优先级1 SD208 优先级8 优先级7 优先级6 优先级5 SD209 优先级10 优先级9 •如果设置为 0，则不显示； 然而，即使设置为“0”，有关 CPU 运行停止（包括参数设置）错误仍将无条件地被 LED 显示。 要了解相应的优先级顺序，参见 7.9.5 节备注。	U	D9038	
SD208		优先级 5 到 8			D9039 格式改变	
SD209		优先级 9 到 10			新	
SD210	时钟数据	时钟数据 (年 月)	•年（后二位数字）和月用 BCD 码存储在 SD210 如下所示： B15 ..... B12 B11 ..... B8 B7 ..... B4 B3 ..... B0 示例：7，1993 H9307 年 月	S/U（请求）	D9025	+Rem
SD211	时钟数据	时钟数据（日时）	•日和时用 BCD 码存储在 SD211 如下所示： B15 ..... B12 B11 ..... B8 B7 ..... B4 B3 ..... B0 示例：31日, 10 a.m. H3110 天 小时		D9026	
SD212	时钟数据	时钟数据（分秒）	•分和秒(在时以后)用BCD码存储在SD212如下所示： B15 ..... B12 B11 ..... B8 B7 ..... B4 B3 ..... B0 示例：35 分, 48 秒 (点钟后) H3548 分 秒		D9027	
SD213	时钟数据	时钟数据（星期几）	•星期几用 BCD 码存储在 SD213 如下所示： B15 ..... B12 B11 ..... B8 B7 ..... B4 B3 ..... B0 示例：星期五 H0005 年份(0-99)的高位 星期数：0 星期日；1 星期一；2 星期二；3 星期三；4 星期四；5 星期五；6 星期六	S/U（请求）	D9028	远程 QCPU
			•星期几用 BCD 码存储在 SD213 如下所示： B15 ..... B12 B11 ..... B8 B7 ..... B4 B3 ..... B0 示例：星期五 H0005 总是设置为0 星期数：0 星期日；1 星期一；2 星期二；3 星期三；4 星期四；5 星期五；6 星期六	S/U（请求）	D9028	QnA

特殊寄存器列表(续)

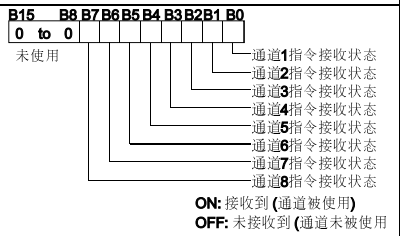
编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD220 SD221 SD222 SD223 SD224 SD225 SD226 SD227	LED 显示数据	显示器数据	•这里存储 LED 显示的 ASCⅡ数据（16 字符） B15 到 B8 B7 到 B0 SD220 从右边数第15个字符 从右边数第16个字符 SD221 从右边数第13个字符 从右边数第14个字符 SD222 从右边数第11个字符 从右边数第12个字符 SD223 从右边数第9个字符 从右边数第10个字符 SD224 从右边数第7个字符 从右边数第8个字符 SD225 从右边数第5个字符 从右边数第6个字符 SD226 从右边数第3个字符 从右边数第4个字符 SD226 从右边数第1个字符 从右边数第2个字符	S（当改变时）	新	
SD240	基本模式	0：自动模式 1：细节模式	•存储基本模式	S（初始化）	新	远程 QCPU
SD241	扩展基板编号	0：仅用于主基板 1-7：扩展基板的编号	•存储安装的扩展基板的最大编号。	S（初始化）	新	
SD242	A/Q 基本区别	基板区别： 0： QA**安装 B（A 模式） 1：Q**安装 B（Q 模式）	7 2 1 0 bit No. 固定为0 to 基板 第1级扩展基板 第2级扩展基板 到 当没有安装扩展基板时，值固定为0。 第7级扩展基板	S（初始化）	新	
SD243 SD244	基板插槽编号	基板插槽号	F CB 87 43 0 bit No. SM243 第三级扩展基板 第二级扩展基板 第一级扩展基板 基板 SM244 第七级扩展基板 第六级扩展基板 第五级扩展基板 第四级扩展基板 •如上所示：每个区存储安装的插槽的编号	S（初始化）	新	
SD250	装入的最大的 I/O	安装的最大 I/O 号	•当 SM250 从 OFF 切换到 ON 时，装入的模块的最后 I/O 地址的高二位数字加 1，作为二进制值存储起来。	S（请求/END 指令）	新	+Rem
SD251	用于更换的 I/O 起始地址	用于模块更换的 I/O 起始地址	•存储在线状态下 取下/更换的 I/O 模块的起始地址的高二位数字（缺省值：100H）	U	D9094	Q2A（S1） Q3A Q4A Q4AR
SD253	RS422 波特率	RS-422 波特率	•存储 RS-422 波特率 0：9600bps 1：19.2kbps 2：38.4kbps	S（当改变时）	新	QCPU
SD254	NET/10 信息	安装模块	•显示安装在 MECSECNET/10 网上的模块数目。	S（初始化）	新	
SD255		来自第一个模块的信息 I/O 号	•安装的第一个模块的 MELSECNET/10 I/O 地址号。			
SD256		网络号	•安装的第一个模块的 MELSECNET/10 网络号			
SD257		组号	•安装的第一个模块的 MELSECNET/10 组号			
SD258		站号	•安装的第一个模块的 MELSECNET/10 站号			
SD259		待机信息	•对备用站情况，备用站的模块号被存储（1-4）。			
SD260-SD264		来自第二个模块的信息	•配置与第一个模块相同。			
SD265-SD269		来自第三个模块的信息	•配置与第一个模块相同。			
SD270-SD274		来自第四个模块的信息	•配置与第一个模块相同。			

特殊寄存器列表(续)

编号	名称	含义	解释	向谁设置 （何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD280	CC-Link (错误)	错误检测状态	（1） 当所安装 CC-Link 的 XnO 切换到 ON 时，相应于工作站的位切换到 ON。 （2） 当所安装 CC-Link 的 Xn1 或 XnF 切换到到 OFF，相应于工作站的位切换到 ON。 （3） 当 CPU 不能与所安装的 CC-Link 通信时切换到 ON。 	S（错误）	新	QCPU 远程
			① 所安装 CC-Link 的 XnO 切换到 ON 时，相应于工作站的位切换到 ON。 ② 当所安装 CC-Link 的 Xn1 或 XnF 切换到 OFF，相应于工作站的位切换到 ON。 ③ 当 CPU 不能与安装的 CC-Link 通信时，切换到 ON。 	S（错误）	新	QnA
SD290	软元件分配 （与参数内容相同）	分配给 X 的点数	・存储当前设置给 X 软元件的点数	S（初始）	新	O+Rem
SD291		分配给 Y 的点数	・存储当前设置给 Y 软元件的点数			
SD292		分配给 M 的点数	・存储当前设置给 M 软元件的点数			
SD293		分配给 L 的点数	・存储当前设置给 L 软元件的点数			O
SD294		分配给 B 的点数	・存储当前设置给 B 软元件的点数			O+Rem
SD295		分配给 F 的点数	・存储当前设置给 F 软元件的点数			○
SD296		分配给 SB 的点数	・存储当前设置给 SB 软元件的点数			O+Rem
SD297		分配给 V 的点数	・存储当前设置给 V 软元件的点数			O
SD298		分配给 S 的点数	・存储当前设置给 S 软元件的点数			
SD299	软元件分配 （与参数内容相同）	分配给 T 的点数	・存储当前设置给 T 软元件的点数	S（初始）	新	O
SD300		分配给 ST 的点数	・存储当前设置给 ST 软元件的点数			
SD301		分配给 C 的点数	・存储当前设置给 C 软元件的点数			

App - 33

特殊寄存器列表(续)

编号	名称	含义		解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD302	软元件分配(与参数内容相同)	分配给 D 的点数		存储当前设置给 D 软元件的点数	S（初始）	新	O+Rem
SD303		分配给 W 的点数		存储当前设置给 W 软元件的点数			
SD304		分配给 SW 的点数		存储当前设置给 SW 软元件的点数			
SD315	通信处理的预定时间	通信处理的预定时间		预定与 GX Developer 或其它模块通信处理的指定时间。 指定的值愈大，与其它软元件(GX Develop，串行通信模块）通信的响应时间就愈短。 设定范围：1－100ms 如果指定值超出上述范围，就被视作未设定。 扫描时间因指定时间而变得更长。	END 处理	新	QCPU
SD340	以太网信息	安装的模块编号		•显示安装在以太网模块的编号	S（初始）	新	
SD341		第一模块的信息	I/O 地址	•安装的第一模块以太网编号			
SD342			网络编号	•安装的第一模块以太网网络编号			
SD343			组合编号	•安装的第一模块的组合编号			
SD344			工作站编号	•安装的第一模块的工作站编号			
SD345－356			空	•空（对于 QCPU，第一模块的以太网的 IP 地址存储在缓冲存储器中）			
SD347			空	•空（对于 QCPU，系统第一模块的以太网出错代码可用 ERRORRD 指令读）			
SD348－354		第二模块的信息		•配置与第一模块相同	S（初始）	新	
SD355－361		第三模块的信息		•配置与第一模块相同			
SD362－368		第四模块的信息		•配置与第一模块相同	S（初始）	新	
SD340	以太网信息	安装的模块的编号		•表示安装在以太网的模块的编号	S（初始）	新	QnA
SD341		第一模块信息	I/O 地址	•安装的第一模块的以太网 I/O 地址			
SD342			网络编号	•安装的第一模块的以太网网络编号			
SD343			组编号	•安装的第一模块的以太网组合编号			
SD344			工作站编号	•安装的第一模块的以太网工作站编号			
SD345－346			IP 地址	•安装的第一模块的以太网 IP 地址			
SD347			出错代码	•安装的第一模块的以太网出错代码			
SD348－354		第二模块的信息		•配置与第一模块相同			
SD355－361		第三模块的信息		•配置与第一模块相同			
SD362－368		第四模块的信息		•配置与第一模块相同	S（初始）	新	
SD380	以太网指令的接收状态	第一模块的指令接收状态			S（初始）	新	

App - 34

特殊寄存器列表(续)

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD381	以太网指令接收状态	第二模块信息	配置与第一模块相同	S（初始）	新	QnA
SD382		第三模块信息	配置与第一模块相同			
SD383		第四模块信息	配置与第一模块相同			
SD392	软件版本	内部系统软件版本	● SCII 码存储内部系统软件版本。 高字节 | 低字节（低字节位置的数据不定，版本号存储在高字节位置。） 例如，对于版本 "A"，存储 "41H"。 注意：内部系统软件版本号可以不同于印在盒上的版本号。	S（初始）	D9060	
SD395	多 CPU 编号	多 CPU 编号	● CPU 系统配置中，存储了主 CPU 的 PLC 编号 PLC 1：1，PLC 2：2，PLC 3：3，PLC 4：4	S（初始）	新	QCPU 功能版本 B

（3） 系统时钟/计数器

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD412	1 秒计数器	以一秒为单位的计数数目	● 程序控制器 CPURUN，每秒加 1 ● 数从 0 到 32767 到-32768 到 0 重复	S（状态）	D9022	O
SD414	2n 秒时钟设定	2n 秒时钟设定	● 储 2n 秒时钟的数值 n（缺省为 30） ● 以在 1 到 32767 之间设定	U	新	
SD415	2nms 时钟设定	2nms 为单位	● 储 2nms 时钟的 n 值（缺省为 30） ● 以在 1 到 32767 之间设定	U	新	QCPU
SD420	扫描计数器	以每次扫描计数数目	● PLC 设置为 RUN 以后每执行一次扫描增加 1。 ● 数从 0 到 32767 到-32768 到 0 重复。	S（每次 END 处理）	新	O
SD430	低速扫描计数器	以每次扫描计数的数目	● PLC 设置为 RUN 以后每执行一次扫描增加 1。 ● 数从 0 到 32767 到-32768 到 0 重复。 ● 用于低速执行	S（每次 END 处理）	新	

*：在初始化执行型程序中不由扫描计数。

App - 35

特殊寄存器列表

（4） 扫描信息

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD500	执行程序号	被执行程序的执行类别	● 前被执行程序的程序编号以二进制形式进行存储。	S（状态改变）	新	
SD510	低速程序号	执行中的低速程序文件名	● 前被执行的低速程序的程序编号以二进制形式进行存储。 ● 在 SM510 处于 ON 状态时才有效。	S (每次 END 处理)	新	
SD520	当前扫描时间	当前扫描时间（以毫秒为单位）	● 存当前扫描时间（以毫秒为单位）。 范围为 0-65535。	S (每次 END 处理)	D9017 格式改变	
SD521		当前扫描时间（以 100 微秒为单位）	储存当前扫描时间（以 100 微秒为单位）。范围为 000-900。 （示例） 当前扫描时间 23.6 毫秒按以下方式进行存储： SD520=23 SD521=600	S (每次 END 处理)	新	
SD522	初始化扫描时间	初始化扫描时间（以毫秒为单位）	● 存起始执行类型程序的扫描时间（以毫秒为单位）。 ● 围为 0-65535。	S（处理第一条 END 指令时）	新	
SD523		初始化扫描时间（以 100 微秒为单位）	● 存起始执行类型程序的扫描时间（以 100 微秒为单位）。 ● 围为 000-900。			
SD524	最小扫描时间	最小扫描时间（以毫秒为单位）	● 存扫描时间的最小值（以毫秒为单位） 范围为 0-65535。	S (每次 END 处理)	D9018 格式改变	
SD525		最小扫描时间（以 100 微秒为单位）	● 存扫描时间的最小值（以 100 微秒为单位）。 ● 围为 000-900。	S（每次 END 处理）	新	
SD526	最大扫描时间	最大扫描时间（以毫秒为单位）	● 存扫描时间的最大值，除了第一次扫描外（以 100 微秒为单位）。 ● 围为 0-65535。	S (每次 END 处理)	D9019 格式改变	
SD527		最大扫描时间（以 100 微秒为单位）	● 存扫描时间的最大值，除了第一次扫描外（以 100 微秒为单位）。 ● 围为 000-900。		新	
SD528	低速执行类型程序的当前扫描时间	当前扫描时间（以毫秒为单位）	● 存低速执行类型程序的当前扫描时间（以毫秒为单位）。	S (每次 END 处理)	新	
SD529		当前扫描时间（以 100 微秒为单位	● 存低速执行类型程序的当前扫描时间（以 100 微秒为单位）。 ● 围为 000-900。			
SD532	低速执行类型程序最小扫描时间	最小扫描时间（以毫秒为单位）	● 存低速执行类型程序扫描时间的最小值（以毫秒为单位）。 ● 围为 0-65535。	S (每次 END 处理)	新	
SD533		最小扫描时间（以 100 微秒为单位）	● 存低速执行类型程序扫描时间的最小值（以 100 微秒为单位）。 ● 围为 000-900。			
SD534	低速执行类型程序最大扫描时间	最大扫描时间（以毫秒为单位）	● 存扫描时间的最大值，除了低速执行类型程序第一次扫描外（以毫秒为单位）。 ● 围为 0-65535。	S (每次 END 处理)	新	
SD535		最大扫描时间（以 100 微秒为单位）	● 存扫描时间的最大值，除了低速执行类型程序第一次扫描外（以 100 微秒为单位）。 ● 围为 000-900。			
SD540	END 指令处理时间	END 指令处理时间（以毫秒为单位）	● 存从扫描程序完成到下一个扫描开始之间的整个时间段（以毫秒为单位）。 ● 围为 0-65535。	S (每次 END 处理)	新	
SD541		END 指令处理时间（以 100 微秒为单位）	● 存从扫描程序完成到下一个扫描开始之间的整个时间段（以 100 微秒为单位）。 ● 围为 000-900。			

App - 36

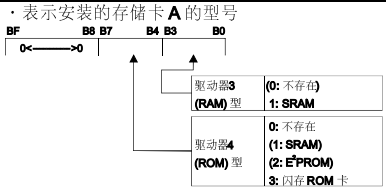
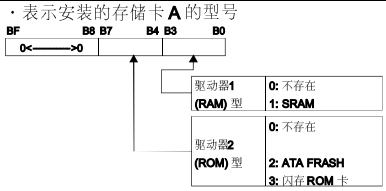
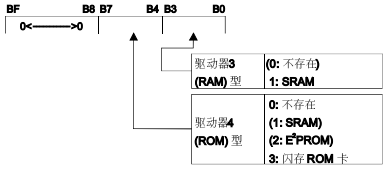
特殊寄存器列表(续)

编号	名称	含义	解释	向谁设置（何时设置）	相应的 ACPU D9□□□	相应的 CPU
SD542	恒定扫描等待时间	恒定扫描等待时间（以 1ms 为单位）	・存储器设置了恒定扫描时间时的等待时间。（以 1ms 为单位） ・范围从 0 到 65535	S（首次 END 处理）	新	O
SD543		恒定扫描等待时间（以 100ms 为单位）	・存储器设置了恒定扫描时间时的等待时间（以 1ms 为单位） ・范围从 000 到 900			
SD544	低速执行型程序的累积执行时间	低速执行型程序的累积执行时间（以 1ms 为单位）	・存储低速执行型程序的累积执行时间（以 1ms 为单位） ・范围从 0 到 65535 ・1 次低速扫描后清除到 0	S（每次 END 处理）	新	
SD545		低速执行型程序的累积执行时间（以 100ms 为单位）	・存储低速执行型程序的累积执行时间（以 1ms 为单位） ・范围从 000 到 900 ・1 次低速扫描后清除到 0			
SD546	低速执行型程序的执行时间	低速执行型程序的执行时间（以 1ms 为单位）	・在一次扫描期间存储低速执行型程序的执行时间。（以 1ms 为单位）・范围从 0 到 65535 ・每一次扫描存储	S（每次 END 处理）	新	
SD547		低速执行型程序的执行时间（以 100ms 为单位）	・在一次扫描期间存储低速执行型程序的执行时间。（以 100ms 为单位） ・范围从 000 到 900 ・每次扫描存储			
SD548	扫描程序执行时间	扫描程序执行时间（以 1ms 为单位）	・存储扫描执行型程序的执行时间在一次扫描期间（以 100ms 为单位） ・范围从 0 到 65535。 ・再次扫描存储	S（每次 END 处理）	新	
SD559		扫描程序执行时间（以 100ms 为单位）	・在一次扫描期间存储扫描执行型程序的执行时间。（以 100ms 为单位） ・范围从 000 到 900 ・每次扫描存储			
SD550	服务间隔测量模块	模块编 号	・设置测量服务间隔的模块的 I/O 地址	U	新	
SD551	服务间隔时间	模块服务间隔（以 1ms 为单位）	・当 SM551 为 ON 时，存储由 SD550 指定的模块的服务间隔。（以 1ms 为单位） ・范围从 0 到 65535	S（请求）	新	
SD552		模块服务间隔（以 100ms 为单位）	・当 SM551 为 ON 时，存储由 SD550 指定的模块的服务间隔。（以 100ms 为单位） ・范围从 000 到 900			

App - 37

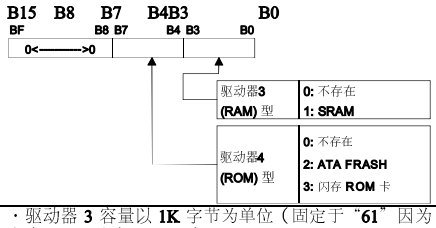
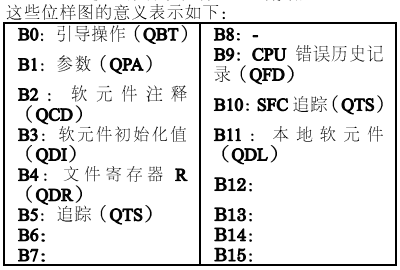
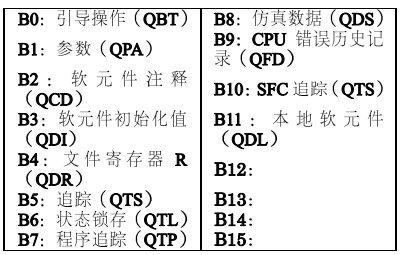
特殊寄存器列表(续)

（5） 存储卡

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
D600	A 型存储卡	A 型存储卡	・表示安装的存储卡 A 的型号 	S（初始化和取出卡）	新	QCPU
			・表示安装的存储卡 A 的型号 	S（初始化和取出卡）	新	QnA
SD602	驱动器 1（RAM）容量	驱动器 1 容量	・驱动器容量（以 1K 字节为单位）	S（初始化和取出卡）	新	QCPU
				S（初始化和取出卡）	新	QnA
SD603	驱动器 2（ROM）容量	驱动器 2 容量	・驱动器容量（以 1K 字节为单位）	S（初始化和取出卡）	新	QCPU
				S（初始化和取出卡）	新	QnA
SD604	存储卡 A 使用条件	存储卡 A 使用条件	・存储卡 A 的使用条件以位样图形式存储（ON 时在使用中） ・这些位样图的意义如下： B0：引导操作（QBT） B1：参数（QPA） B2：软元件注释（QCD） B3：软元件初始化值（QDI） B4：文件寄存器 R（QDR） B5：追踪（QTS） B6： B7： B8：- B9：CPU 错误历史记录（QFD） BA：SFC 追踪（QTS） BB：本地软元件（QDL） BC： BD： BE： BF：	S（状态改变）	新	QCPU
			・存储卡 A 的使用条件以位方式形式存储（ON 时在使用中） ・这些位方式的意义如下： B0：引导操作（QBT） B1：参数（QPA） B2：软元件注释（QCD） B3：软元件初始化值（QDI） B4：文件寄存器 R（QDR） B5：采样追踪（QTS） B6：状态锁存（QTL） B7：程序追踪（QTP） B8：仿真数据（QDS） B9：CPU 错误历史记录（QFD） BA：SFC 追踪（QTS） BB：本地软元件（QDL） B12： B13： B14： B15：	S（状态改变）	新	QnA
SD620	B 型存储卡	B 型存储卡	・表示安装的存储卡 B 的型号  驱动器 4 固定于“3”，因为它有内置式闪存 ROM	S（初始化）	新	QCPU

App - 38

特殊寄存器列表(续)

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>向谁设置
（何时设置）</th><th>对应的
ACPU
D9□□□</th><th>对应的
CPU</th></tr>
<tr><td>SD620</td><td>B 型存储卡</td><td>B 型存储卡</td><td>・表示安装的存储卡 B 的型号

</td><td>S（初始化）</td><td>新</td><td>Q2A（S1）
Q3A
Q4A
Q4AR</td></tr>
<tr><td rowspan="2">SD622</td><td rowspan="2">驱动器 3
（RAM）容量</td><td rowspan="2">驱动器 3 容量</td><td>・驱动器 3 容量以 1K 字节为单位（固定于“61”因为它有 61KB 的内置 RAM）</td><td>S（初始化）</td><td>新</td><td>QCPU</td></tr>
<tr><td>・驱动器 3 容量（以 1K 字节为单位）</td><td>S（初始化）</td><td>新</td><td>Q2A（S1）
Q3A
Q4A
Q4AR</td></tr>
<tr><td rowspan="2">SD623</td><td rowspan="2">驱动器 4
（ROM）容量</td><td rowspan="2">驱动器 4 容量</td><td>・驱动器 4 容量（以 1K 字节为单位）</td><td>S（初始化）</td><td>新</td><td>QCPU</td></tr>
<tr><td>・驱动器 4 容量（以 1K 字节为单位）</td><td>S（初始化）</td><td>新</td><td>Q2A（S1）
Q3A
Q4A
Q4AR</td></tr>
<tr><td rowspan="2">SD624</td><td>驱动器 3/4 使用条件</td><td>驱动器 3/4 使用条件</td><td>・驱动器 3/4 使用条件以位样图形式存储
・这些位样图的意义表示如下：

</td><td>S（状态改变）</td><td>新</td><td>QCPU</td></tr>
<tr><td>存储卡 B
使用条件</td><td>存储卡 B
使用条件</td><td>・存储卡 B 使用条件以位样图形式存储
・这些位样图的意义表示如下：

</td><td>S（状态改变）</td><td>新</td><td>Q2A（S1）
Q3A
Q4A
Q4AR</td></tr>
<tr><td>SD640</td><td>文件寄存器
驱动器</td><td>驱动器号</td><td>・存储正由文件寄存器使用的驱动器编号</td><td>S（初始化）</td><td>新</td><td rowspan="6"></td></tr>
<tr><td>SD641</td><td rowspan="5">文件寄存器
文件名</td><td rowspan="5">文件寄存器
文件名</td><td>・存储用参数选择的或由 QDRSET 指令作为 ASCII 码来使用的文件寄存器文件名称（扩展名）
B15 到 B8 B7 到 B0

</td><td>S（初始化）</td><td rowspan="5">新</td></tr>
<tr><td>SD642</td><td rowspan="4">S（初始化）</td></tr>
<tr><td>SD643</td></tr>
<tr><td>SD644</td></tr>
<tr><td>SD645</td></tr>
<tr><td>SD646</td><td></td><td></td><td></td></tr>
<tr><td>SD647</td><td>文件寄存器容量</td><td>文件寄存器容量</td><td>・存储当前选择的文件寄存器的数据，（以 1K 字为单位）</td><td>S（状态改变）</td><td>新</td><td></td></tr>
<tr><td>SD648</td><td>文件寄存器块号</td><td>文件寄存器块号</td><td>・存储当前选择的文件寄存器块的编号</td><td>S（状态改变）</td><td>D9035</td><td></td></tr>
<tr><td>SD650</td><td>注释驱动器</td><td>注释驱动器号</td><td>・存储用参数选择的或由 QCDSET 指令选择的注释驱动器编号</td><td>S（状态改变）</td><td>新</td><td></td></tr>
</table>

App - 39

特殊寄存器表(续)

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD651 SD652 SD653 SD654 SD655 SD656	注释文件名	注释文件名	·存储用参数中选择的或由 QCDSET 指令以 ASCII 代码型式选择的注释文件名称(扩展名) B15 到 B8 B7 到 B0 SD651 第 2 个字符 第 1 个字符 SD652 第 4 个字符 第 3 个字符 SD653 第 6 个字符 第 5 个字符 SD654 第 8 个字符 第 7 个字符 SD655 扩展名第 1 个字符 2EH（.） SD656 扩展名第 3 个字符 扩展名第 2 个字符	S（状态改变）	新	
SD660	引导操作指定文件	引导指定文件驱动器号	·存储引导指定文件〔*QBT〕的驱动器的编号	S（初始化）	新	
SD641 SD642 SD643 SD644 SD645 SD646		引导指定文件的文件名	·存储引导指定文件〔*QBT〕的文件名称 B15 到 B8 B7 到 B0 SD661 第 2 个字符 第 1 个字符 SD662 第 4 个字符 第 3 个字符 SD663 第 6 个字符 第 5 个字符 SD664 第 8 个字符 第 7 个字符 SD665 扩展名第 1 个字符 2EH（.） SD666 扩展名第 3 个字符 扩展名第 2 个字符	S（初始化）	新	

（6） 与指令相关的寄存器

编号	名称	含义	解释	向谁设置（何时设置）	对应的 ACPU D9□□□	对应的 CPU
SD705 SD706	屏蔽形式	屏蔽形式	·在程序块运行期间，将 SM705 切换到 ON，使可能使用存储在SM705中的屏蔽组合(如果使用双字则在SM705和SM706)用屏蔽数值来运行程序块中的所有数据	U	新	O
SD714	空的通信请求注册区的数目	0 到 32	·存储用连接到 AJ71PT32-S3 的远程终端模块的通信请求区中的空块的数目	S（执行期间）	M9081	QnA
SD715 SD716 SD717	IMASK 指令屏蔽组合	屏蔽形式	·使用以下列方法存储的 IMASK 指令屏蔽的组合 B15 B11 B0 SD715 I15 到 I1 I0 SD716 I31 到 I17 I16 SD717 I47 到 I33 I32	S（执行期间）	新	O
SD718 SD719	累加器	累加器	·用于替代 A 系列程序使用的累加器	S/U	新	
SD720	指定用于 PLOAD 指令的程序编号	指定用于 PLOAD 指令的程序编号	·存储指定的要用 PLOAD 指令装入的程序的程序编号 ·指定范围：1-124。	U	新	QCPU
SD730	用于CC-Link通信请求的空注册区的编号	0 到 132	·存储与连接到 A(1S)J61QBT61 的智能软元件工作站的通信请求的空注册区的编号	S（执行期间）	新	QnA
SD736	PKEY 输入	PKEY 输入	·通过 PKEY 指令暂时存储键盘数据输入的 SD	S（执行期间）	新	O

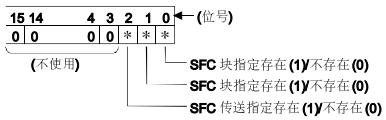
特殊寄存器表(续)

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>向谁设置
（何时设置）</th><th>对应的
ACPU
D9□□□</th><th>对应的 CPU</th></tr>
<tr><td>SD738
SD739
SD740
SD741
SD742
SD743
SD744
SD745
SD746
SD747
SD748
SD749
SD750
SD751
SD752
SD753
SD754
SD755
SD756
SD757
SD758
SD759
SD760
SD761
SD762
SD763
SD764
SD765
SD766
SD767
SD768
SD769</td><td>信息存储</td><td>信息存储</td><td>・存储由 MSG 指令指定的信息
<table><tr><th></th><th>B15 到 B8</th><th>B7 到 B0</th></tr><tr><td>SD738</td><td>第 2 个字符</td><td>第 1 个字符</td></tr><tr><td>SD739</td><td>第 4 个字符</td><td>第 3 个字符</td></tr><tr><td>SD740</td><td>第 6 个字符</td><td>第 5 个字符</td></tr><tr><td>SD741</td><td>第 8 个字符</td><td>第 7 个字符</td></tr><tr><td>SD742</td><td>第 10 个字符</td><td>第 9 个字符</td></tr><tr><td>SD743</td><td>第 12 个字符</td><td>第 11 个字符</td></tr><tr><td>SD744</td><td>第 14 个字符</td><td>第 13 个字符</td></tr><tr><td>SD745</td><td>第 16 个字符</td><td>第 15 个字符</td></tr><tr><td>SD746</td><td>第 18 个字符</td><td>第 17 个字符</td></tr><tr><td>SD747</td><td>第 20 个字符</td><td>第 19 个字符</td></tr><tr><td>SD748</td><td>第 22 个字符</td><td>第 21 个字符</td></tr><tr><td>SD749</td><td>第 24 个字符</td><td>第 23 个字符</td></tr><tr><td>SD750</td><td>第 26 个字符</td><td>第 25 个字符</td></tr><tr><td>SD751</td><td>第 28 个字符</td><td>第 27 个字符</td></tr><tr><td>SD752</td><td>第 30 个字符</td><td>第 29 个字符</td></tr><tr><td>SD753</td><td>第 32 个字符</td><td>第 31 个字符</td></tr><tr><td>SD754</td><td>第 34 个字符</td><td>第 33 个字符</td></tr><tr><td>SD755</td><td>第 36 个字符</td><td>第 35 个字符</td></tr><tr><td>SD756</td><td>第 38 个字符</td><td>第 37 个字符</td></tr><tr><td>SD757</td><td>第 40 个字符</td><td>第 39 个字符</td></tr><tr><td>SD758</td><td>第 42 个字符</td><td>第 41 个字符</td></tr><tr><td>SD759</td><td>第 44 个字符</td><td>第 43 个字符</td></tr><tr><td>SD760</td><td>第 46 个字符</td><td>第 45 个字符</td></tr><tr><td>SD761</td><td>第 48 个字符</td><td>第 47 个字符</td></tr><tr><td>SD762</td><td>第 50 个字符</td><td>第 49 个字符</td></tr><tr><td>SD763</td><td>第 52 个字符</td><td>第 51 个字符</td></tr><tr><td>SD764</td><td>第 54 个字符</td><td>第 53 个字符</td></tr><tr><td>SD765</td><td>第 56 个字符</td><td>第 55 个字符</td></tr><tr><td>SD766</td><td>第 58 个字符</td><td>第 57 个字符</td></tr><tr><td>SD767</td><td>第 60 个字符</td><td>第 59 个字符</td></tr><tr><td>SD768</td><td>第 62 个字符</td><td>第 61 个字符</td></tr><tr><td>SD769</td><td>第 64 个字符</td><td>第 63 个字符</td></tr></table></td><td>S（执行期间）</td><td>新</td><td>O</td></tr>
<tr><td>SD774-SD775</td><td>PID 限制设定</td><td>0:设置限制
1:不设置限制</td><td>・按如下所示指定每个 PID 回路的限制：
<table><tr><th>B15</th><th></th><th></th><th>B1</th><th>B0</th></tr><tr><td>SD774</td><td>回路 16</td><td>到</td><td>回路 2</td><td>回路 1</td></tr><tr><td>SD775</td><td>回路 32</td><td>到</td><td>回路 18</td><td>回路 17</td></tr></table></td><td>U</td><td>新</td><td>QCPU</td></tr>
<tr><td>SD780</td><td>保留同步执行 CC-Link 专用指令的编号</td><td>0 到 32</td><td>・储存 CC-Link 同时执行专用指令的保留数目。</td><td>U</td><td>新</td><td>QnA</td></tr>
<tr><td>SD781-SD793</td><td>IMASK 指令的屏蔽形式</td><td>屏蔽形式</td><td>・按如下所示储存由 IMASK 指令屏蔽的
屏蔽方式：
<table><tr><td>SD781</td><td>I63</td><td>到</td><td>I59</td><td>I48</td></tr><tr><td>SD782</td><td>I79</td><td>到</td><td>I65</td><td>I64</td></tr><tr><td colspan="5">到</td></tr><tr><td>SD793</td><td>I255</td><td>到</td><td>I241</td><td>I240</td></tr></table></td><td>S（执行期间）</td><td>新</td><td>QCPU</td></tr>
</table>

App - 41

特殊寄存器列表

（7） 调制

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>向谁设置
（何时设置）</th><th>对应的
ACPU
D9□□□</th><th>对应的
CPU</th></tr>
<tr><td>SD806
SD807
SD808
SD809
SD810
SD811</td><td>状态锁存文件名称</td><td>状态锁存文件名称</td><td>·从用 ASCII 码进行状态锁存时开始存储文件名称(带扩展名)
B15 到 B8 B7 到 B0
<table><tr><td>SD806</td><td>第 2 个字符</td><td>第 1 个字符</td></tr><tr><td>SD807</td><td>第 4 个字符</td><td>第 3 个字符</td></tr><tr><td>SD808</td><td>第 6 个字符</td><td>第 5 个字符</td></tr><tr><td>SD809</td><td>第 8 个字符</td><td>第 7 个字符</td></tr><tr><td>SD810</td><td>扩展的第 1 个字符</td><td>2EH (.)</td></tr><tr><td>SD811</td><td>扩展的第 3 个字符</td><td>扩展的第 2 个字符</td></tr></table></td><td>S（执行期间）</td><td>新</td><td rowspan="2">QnA</td></tr>
<tr><td>SD812
SD813
SD814
SD815
SD816</td><td>状态锁存步</td><td>状态锁存步</td><td>从进行状态锁存时,开始存储步号
<table><tr><td>SD812</td><td>型式*</td></tr><tr><td>SD813</td><td>块号</td></tr><tr><td>SD814</td><td>步号、传送号</td></tr><tr><td>SD815</td><td>顺序步号（L）</td></tr><tr><td>SD816</td><td>顺序步号（H）</td></tr></table>*型式数据的内容
</td><td>S（执行期间）</td><td>D9055 格式改变</td></tr>
</table>

App - 42

特殊寄存器表

（8） 锁存区

<table>
<tr><th>编号</th><th>名称</th><th>含义</th><th>解释</th><th>向谁设置
（何时设置）</th><th>对应的
ACPU
D9□□□</th><th>对应的 CPU</th></tr>
<tr><td>SD900</td><td>电源曾中断的驱动器</td><td>在断电期间访问文件驱动器编号</td><td>・如果在断电期间正在访问文件,存储驱动器的编号</td><td>S（状态改变）</td><td>新</td><td rowspan="24">QnA</td></tr>
<tr><td>SD901</td><td rowspan="6">断电期间活动的文件</td><td rowspan="6">断电期间访问文件的名称</td><td rowspan="6">・如果在断电期间正在访问文件,用 ASCII 码存储该文件的名称（带扩展名）
B15 到 B8 B7 到 B0
SD901 第 2 个字符 第 1 个字符
SD902 第 4 个字符 第 3 个字符
SD903 第 6 个字符 第 5 个字符
SD904 第 8 个字符 第 7 个字符
SD905 扩展的第 1 个字符 2EH (.)
SD906 扩展的第 3 个字符 扩展的第 2 个字符</td><td rowspan="6">S（状态改变）</td><td rowspan="6">新</td></tr>
<tr><td>SD902</td></tr>
<tr><td>SD903</td></tr>
<tr><td>SD904</td></tr>
<tr><td>SD905</td></tr>
<tr><td>SD906</td></tr>
<tr><td>SD910</td><td rowspan="17">RKEY 输入</td><td rowspan="17">RKEY 输入</td><td rowspan="17">・按输入 PU 键代码的顺序存储
B15 到 B8 B7 到 B0
SD910 第 2 个字符 第 1 个字符
SD911 第 4 个字符 第 3 个字符
SD912 第 6 个字符 第 5 个字符
SD913 第 8 个字符 第 7 个字符
SD914 第 10 个字符 第 9 个字符
SD915 第 12 个字符 第 11 个字符
SD916 第 14 个字符 第 13 个字符
SD917 第 16 个字符 第 15 个字符
SD918 第 18 个字符 第 17 个字符
SD919 第 20 个字符 第 19 个字符
SD920 第 22 个字符 第 21 个字符
SD921 第 24 个字符 第 23 个字符
SD922 第 26 个字符 第 25 个字符
SD923 第 28 个字符 第 27 个字符
SD924 第 30 个字符 第 29 个字符
SD925 第 32 个字符 第 31 个字符</td><td rowspan="17">S（执行时）</td><td rowspan="17">新</td></tr>
<tr><td>SD8911</td></tr>
<tr><td>SD912</td></tr>
<tr><td>SD913</td></tr>
<tr><td>SD914</td></tr>
<tr><td>SD915</td></tr>
<tr><td>SD916</td></tr>
<tr><td>SD917</td></tr>
<tr><td>SD918</td></tr>
<tr><td>SD919</td></tr>
<tr><td>SD920</td></tr>
<tr><td>SD921</td></tr>
<tr><td>SD922</td></tr>
<tr><td>SD923</td></tr>
<tr><td>SD924</td></tr>
<tr><td>SD925</td></tr>
</table>

（9） A 到 Q/ QnA 转换一致性

在 A 系列转换到 Q/ QnA 系列以后，ACPU 特殊寄存器 D900 到 D9255，相当于特殊寄存器 DS1000 到 SD1255。
这些特殊寄存器全部由系统设置，用户不能用它们来设置程序数据。
需要用这些特殊寄存器来设置数据的用户应该编辑用于 Q/ QnA 的特殊寄存器。
然而，在转换之前，用户仅能在特殊存器 D9200 至 D9255 中设置数据，转换以后，用户也可以在寄存器 1200 至 1255 中设置数据。
有关 ACPU 特殊寄存器内容的细节，参阅单个 CPU 的用户手册和 MELSECNET 和 MELSECNET/B 数据通讯系统参考手册。

备注

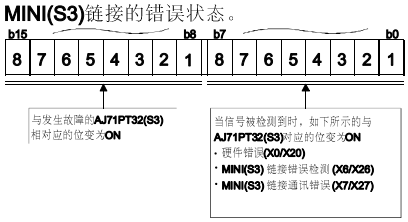
有关用于“修改的特殊寄存器”这一列的补充说明
①为“用于修改的特殊寄存器”指定的软元件编号修改为用于 Q/ QnACPU 特殊寄存器。
②与“□”相应的软元件编号可以使用转换后的特殊寄存器。
③与“□”相应的软元件编号对 Q/ QnA 不起作用。

特殊寄存器列表

<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于修改的特殊寄存器</th><th>名称</th><th>含义</th><th>详细说明</th><th>对应的 CPU</th></tr>
<tr><td>D9000</td><td>SD1000</td><td>—</td><td>熔丝烧断</td><td>熔丝烧断的模块的编号</td><td>· 当检测到熔丝烧断的模块时，用十六进制数来存储检测到模块的最低编号（例如:当 Y50 到 6F 的输出模块的熔丝烧断时存储十六进制数"50"）。用外围设备来监视编号,执行十六进制的监视运行，用外围设备来监视编号,执行十六进制的监视运行,。
（当SD1000到SD1107的内容被复位时,清除为0）
· 也对这程 I/O 工作站的输出模块执行熔丝烧断检查。</td><td rowspan="2">O</td></tr>
<tr><td>D9001</td><td>SD1001</td><td>—</td><td>熔丝烧断</td><td>熔丝烧断的模块的编号</td><td>·当熔丝烧断时,存储相应于设定开关编号或基板插槽编号的模块编号。
<table>
<tr><th colspan="2">A0J2 的 I/O 模块</th><th colspan="2">扩展基板</th></tr>
<tr><th>设置开关</th><th>存储数据</th><th>基板插槽号</th><th>存储数据</th></tr>
<tr><td>0</td><td>1</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>2</td><td>1</td><td>6</td></tr>
<tr><td>2</td><td>3</td><td>2</td><td>7</td></tr>
<tr><td>3</td><td>4</td><td>3</td><td>8</td></tr>
<tr><td>4</td><td>5</td><td></td><td></td></tr>
<tr><td>5</td><td>6</td><td></td><td></td></tr>
<tr><td>6</td><td>7</td><td></td><td></td></tr>
<tr><td>7</td><td>8</td><td></td><td></td></tr>
</table></td></tr>
</table>

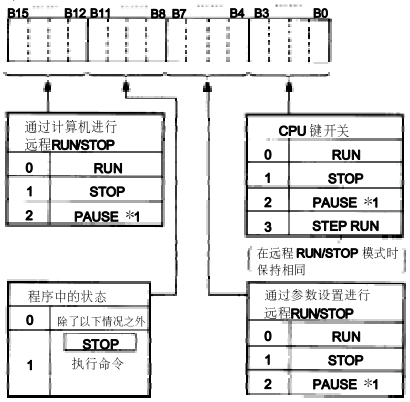
App - 44

特殊寄存器列表（续）

ACPU 特殊转换	转换后的特殊寄储器	用于修改的特殊寄存器	名称	含义	详细说明	对应的 CPU
D9002	SD1002	-	模块 I/0 核对错误	I/0 模块核对错误模块编号	·如果开启电源时检测到其数据不同于输入的数据的 I/O 模块，则被检测到的模块中的最低编号模块的第一个 I/0 编号，以十六进制形式存储起来.(存储方法与 SD1000 的相同). 要通过外围设备来监视编号，执行十六进制的监视运行 (当 SD1116 至 SD1123 的所有内容复位到 0 时,内容被清除)。 ·对远程 I/0 终端的模块,也执行 I/0 模块的核对检查.	O
D9004	SD1004	-	MINI 链接错误	存储用参数制定的设定状态(模块 1 到 8)	·存储在安装的 AJ71DT32(S3)上检测到的 MINI(S3)链接的错误状态。 	QnA
D9005	SD1005	-	交流断电计数器	交流断电次数	在使用 AC 电源模块，发生 20ms 以内的瞬时断电故障时，寄存器数值加 1 (数值以二进制的形式存储)在电源从 OFF 切换到 ON 时复位	O
					在使用 DC 电源模块，发生 10ms 以内的瞬时断电故障时，寄存器数值加 1 (数值以二进制的形式存储)在电源从 OFF 切换到 ON 时复位	QCPU
					在使用 DC 电源模块，发生 1ms 以内的瞬时断电故障时，寄存器数值加 1 (数值以二进制的形式存储)在电源从 OFF 切换到 ON 时复位	QnA
D9008	SD1008	SD0	自诊断错误	自诊断出错编号	·当自诊断的结果 出现错误时,以二进制代码的形式存储起来	O
D9009	SD1009	SD62	信号报警器检测	发生外部故障的信号报警器(F)的编号	·当通过 OUT F 或 SETF 指令将 FO 到 F255 中的一个切换到 ON，在切换到 ON 的 F 编号中,最早检测到的 F 编号被以二进制代码存储起来 ·SD62 可以通过执行 RSTE 或 LEDR 指令来清除，如果：已经检测到任何 F 编号，SD62 的清除将使下一个编号存储到 SD62	
					·当通过 OUT F 或 SETF 指令将 FO 到 F255 中的一个切换到 ON，在切换到 ON 的 F 编号中，最早检测到的 F 编号被以二进制代码存储起来 ·SD62 可以通过执行 RSTE 或 LEDR 指令或将 CPU 面板上的 INDICATOR RESET 开关切换到 ON 的位置来清除，如果:已经检测到任何 F 编号，SD62 的清除将使下一个编号存储到 SD62	
D9010	SD1010		出错步	发生运行故障的步的编号	·当在执行应用指令期间发生了运行错误时，则发生错误的步编号被以二进制代码存储起来，因此：每次发生运行错误，SD1010 的内容就被更新	

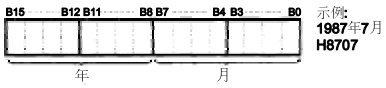
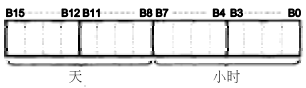
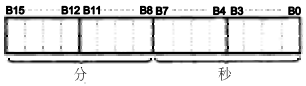
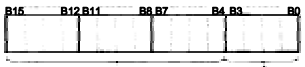
App - 45

特殊寄存器列表（续）

ACPU 特殊转换	转换后的特殊寄存器	用于修改特殊寄存器	名称	含义	细节	对应的 CPU
D9011	SD1011		出错步	发生运行错误的步编号	·在执行应用指令期间,当运行错误发生时,发生错误的步编号,以二进制码形式存储. 当 M1011 从 OFF 改变到 ON 时，存储到 SM1011，除非 SM1011 被用户程序清除，SD1011 的内容不能 更新.	O
D9014	SD1014		I/0 控制模式	I/0 控制模式编号	·I/0 控制模式设置返回到下列任何一种编号中： 0：直接模式的输入和输出 1：刷新模式的输入和直接模式的输出 2：刷新模式的输入和输出	
D9015	SD1015	SD203	CPU 的运行状态	CPU 的运行状态	·CPU 的运行状态如下图所示,存储在 SD203 中。  *1 当 CPU 是在 RUN 模式,SM1040 处于 OFF 位置时，如果改变到 PAUSE 模式则 CPU 保留在 RUN 模式。	
D9016	SD1016		程序号	0:主程序(ROM) 1:主程序(RAM) 2:子程序 1(RAM) 3:子程序 2(RAM) 4:子程序 3(RAM) 5:子程序 3(ROM) 6:子程序 2(ROM) 7:子程序 3(ROM) 8:主程序(EEPROM) 9:子程序 1(EEPROM) A:子程序 2(EEPROM) B:子程序 3(EEPROM)	·表示当前运行的是顺控程序，数值从“0”至 B 中的一个值以二进制代码的形式存储。	
D9017	SD1017	SD520	扫描时间	最小扫描时间 (以 10ms 为单位)	·如果扫描时间小于 SD520 的内容，在每次 END 处理时，将重新存储，即：扫描时间的最小值以二进制的形式存储在 SD520 中	
D9018	SD1018	SD524	扫描时间	扫描时间 (以 10ms 为单位)	·在每次 END 处理时，扫描时间以二进制形式存储起来并重写。	

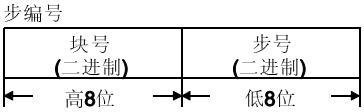
App - 46

特殊寄存器列表（续）

ACPU 特殊转换	转换后的特殊寄存器	用于修改的特殊寄存器	名称	含义	细节	对应的 CPU
D9019	SD1019	SD526	扫描时间	最大扫描时间(以 10ms 为单位)	·如果扫描时间长于 SD526 的内容,每一次 END 处理时重新存储。亦即扫描时间的最大值以二进制码的形式存储在 SD526	O
D9020	SD1020		恒定扫描	恒定扫描时间(用户设置以 10ms 为单位)	·以 10ms 的倍数设置连续程序起点之间的间隔 0:无设定 0：没有设置 1-200：设置程序以（设置值）×10ms 的时间间隔执行。	
D9021	SD1021	—	扫描时间	扫描时间(以 1ms 为单位)	·在每一次 END 以后,以二进制码的形式存储和更新扫描时间	
D9022	SD1022	SD412	1 秒计数器	以 1ms 为单位	·当 PCCPU 开始运行时,开始计数，每秒计数 1 ·从 0 到 32767 计数,然后再到-32767 再回到 0 计数沿这路径重复	
D9025	SD1025	SD210	时钟数据	时钟数据(年,月)	·用 BCD 码存储年（低 2 位数字）和月  示例: 1987年7月 H8707	
D9026	SD1026	SD211	时钟数据	时钟数据(日,时)	·用 BCD 码存储日和时  示例 31日10点 H3110	
D9027	SD1027	SD212	时钟数据	时钟数据(分,秒)	·用 BCD 码存储分和秒  示例: 35 分, 48 秒 H3548	
D9028	SD1028	SD213	时钟数据	时钟数据(星期几)	·用 BCD 码存储星期几  示例: 星期五 H0005 星期几: 0 星期日; 1 星期一; 2 星期二; 3 星期三; 4 星期四; 5 星期五; 6 星期六	
D9035	D1035	SD648	扩展文件寄存器	使用块编号	·用 BCD 码的形式存储正在使用的扩展文件寄存器的块编号	
D9036	SD1036		用于指定软元件编号的扩展文件寄存器	当从扩展文件寄存直接访问个别的软元件时的软元件编号	·以二进制码的形式指定用于在 SD1036 和 SD1037 以二字直接读和写的扩展文件寄存器的软元件编号从 1 号块的 RO 开始,使用连续编号来指定软元件编号 	
D9037	SD1037					
D9038	SD1038	SD207	CED 显示器优先排队	优先级 1 到 4	·设置用出错代码编号发光来表示出错的 ERROR,LEDS 优先级 ·优先级设定区的配置如下表所示: B15 到 B12 B11 到 B8 B7 到 B4 B3 到 B0 SD207: 优先级4	
D9039	SD1039	SD208		优先级 5 到 7	SD208: 优先级7 优先级6 优先级5 ·详细说明：参照应用 CPU 用户手册和 ACPU 编程手册(基础篇)	

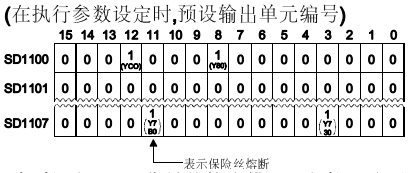
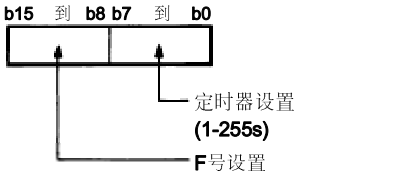
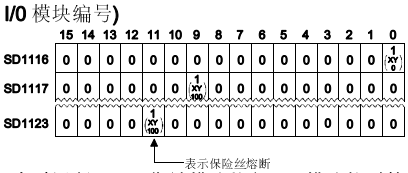
App - 47

特殊寄存器列表（续）

ACPU 特殊转换	转换后的特殊寄存器	修改的特殊寄存器	名称	含义	细节	对应的 CPU
D9044	SD1044		采样跟踪	采样跟踪期间的步或时间	·用外围设备来切换 ON/OFF 在扫描时一————0 时间－－－－－时间(以 10ms 为单位) 以二进制码存储数值	O
D9049	SD1049		SFC 工作区	扩展文件寄存器的块编号	·以二进制数值存储作为执行 SFC 程序的工作区使用的扩展文件寄存器的块编号 ·如果使用不能作为 1 号扩展文件寄存器的 16K 字节或更小的空区，或 SM320 在 OFF 状态，存储“0”。	
D9050	SD1050		SFC 程序出错编号	SFC 程序发生的错误的出错编码	·在二进制码来存储在 SFC 程序中发生的出错的代码编号。 0:无错误 80:SFC 程序参数错误 81:SFC 代码错误 82:超过同步执行的步数 83:块启动错误 84:SFC 程序运行错误	
D9051	SD1051		出错块	发生错误的块编号	·以二进制码存储在 SFC 程序中发生错误的块的编号 在发生 83 号错误时,将启动的块编号存储起来	
D9052	SD1052			发生错误的步编号	·以二进制码存储在 SFC 程序中发生 84 号错误的步编号. 在发生 80 号,81 号,82 号错误时存储"0" 在发生 83 号错误时将块启动步的编号存储起来	
D9053	SD1053		出错传输	发生错误传输条件的编号	·以二进制码存储在 SFC 程序中发生 84 号错误的传输条件的编号 在发生 80,81,82,83 号错误时存储"0"	
D9054	SD1054		错误顺序步	发生错误的顺序步的编号	·以二进制码存储在 SFC 程序中发生 84 号错误的传输条件和运行输出的顺序编号	
D9055	SD1055	SD812	状态锁存	状态锁存步	·存储执行状态锁时的步编号 ·如果在主顺控程序中执行锁存,以二进制数值存储步的编号 ·如果在 SFC 程序中执行锁存,存储块编号和步编号 	
D9060	SD1060	SD392	软件版本	内部软件的软件版本	·用 ASCII 码存储内部系统的软件版本。 低字节位置的数据不定，版本号存储在高字节位置。 例如，对于版本"A",存储"41H"。 注意:初始系统的软件版本可能与打印在包装盒背面的版本不同	QnA
D9072	SD1072		PLC 通信检查	计算机链接数据检查	·在串行通信模块的自回路测试中,串行通信模块自动地写/读数据来进行通信检查	O
D9081	SD1081	SD714	通信请求注册区的空块编号	通信请求注册区的空块编号	·存储连接到 MELSECNET/MINI-S3 主模块，A2C 或 A52G 的远程终端模块的通信请求注册区中空块的编号	QnA

App - 48

特殊寄存器列表（续）

ACPU 特殊转换	转换后的特殊寄存器	修改的特殊寄存器	名称	含义	详细说明	对应的 CPU
D9085	SD1085		设定时间检查值的注册	缺省 10S	•设置 MELSECNET/CO 的数据通讯指令（ZNRD，ZNWR）的检查时间 •设定范围:1 秒到 65535 秒(1-65535) •设定单位:1 秒 •缺省值:10S(如果设置 0，缺省值为 10S)	O
D9090	SD1090		微机子程序输入数据包的起始地址	微机子程序输入数据包的起始地址	•详细说明参考每个微机程序包的手册	
D9091	SD1091		详细出错代码	自诊断详细出错代码	•存储指令错误原因的详细代码	
D9094	SD1094	SD251	更换的起始 I/0 编号	更换的起始 I/0 编号	•用二进制码存储在线模式期间,装入或卸下的 I/0 模块的起始 I/0 地址的高二位数字 例:输入模块 X2FO→H2F	
D9100	SD1100	-	熔丝烧断模块	表示熔丝已烧断的模块,以 16 点为单位的位形式	• (以 16 点为单位的)熔丝烧断的输出模块的编号被输入到位形式中 (在执行参数设定时,预设输出单元编号)  •也对远程 I/0 工作站的输出模块执行熔丝烧断检查 (如果恢复正常状态,不执行清除,因此,要由用户程序来执行清除)	
D9101	SD1101					
D9102	SD1102					
D9103	SD1103					
D9104	SD1104					
D9105	SD1105					
D9106	SD1106					
D9107	SD1107					
D9108	SD1108	-	步传输监视定时器设定	时间用完时的定时器设定值和 F 编号	•设置步传输监视定时器的数值和监视定时器时间用完时,切换到 ON 的 F 的编号  (通过将 MSM708 到 SM1114 中的任何一个切换到 ON)监视定时器启动,如果在设定时间内,在对应于定时器的步后的传输条件未建立,设置信号报警器(F)切换到 ON	
D9109	SD1109					
D9110	SD1110					
D9111	SD1111					
D9112	SD1112					
D9113	SD1113					
D9114	SD1114					
D9116	SD1116	-	I/0 模块核对错误	以 16 点为单位的,指出核对模块的位形式	•当检测到 I/0 模块(其数据与电源接通时输入的数据不同)时,I/0 模块的编号(以 16 点为单位)输入到位形式中去,(当已执行参数设定时,预设 I/0 模块编号)  •也对远程 I/0 工作站模块执行 I/0 模块核对检查,(如果恢复正常状态,不执行清除,因此,要由用户程序来执行清除)	
D9117	SD1117					
D9118	SD1118					
D9119	SD1119					
D9120	SD1120					
D9121	SD1121					
D9122	SD1122					
D9123	SD1123					

App - 49

特殊寄存器列表（续）

<table>
<tr><th>ACPU
特殊
转换</th><th>转换后
的特殊
寄存器</th><th>用于修
改的特
殊寄
存器</th><th>名称</th><th>含义</th><th>详细说明</th><th>相应的
CPU</th></tr>
<tr><td>D9124</td><td>SD1124</td><td>SD63</td><td>信号报警器
检测数量</td><td>信号报警器检测数量</td><td>•当由 SETF 指令将 F0 到 F255(对 AuA 和 AnU 是 F0 到 2047)中的一个切换到 ON，SD63 的内容加 1
当执行 RSTF 或 LEDR 指令时,SD63 的内容减去 1
(如果 CPU 有 INDICATOR RESET 开关,可以按这个开关来执行同样的处理)
•已经由 SETF 指令切换到 ON 的数量，以二进制码的形式存储在 SD63 中,SD63 的最大数值是 8</td><td rowspan="11">O</td></tr>
<tr><td>D9125</td><td>SD1125</td><td>SD64</td><td rowspan="10">信号报警器
检测编号</td><td rowspan="10">信号报警器检测编号</td><td rowspan="10">•当由 SETF 指令将 F0 到 F255(对 AuA 和 AnU 是 F0 到 2047)中的一个切换到 ON，已经切换到 ON 的 F 的编号以二进制码的形式依次输入到 SD64 到 SD71 中去
•由 RSTF 指令切换到 OFF 的 F 编号,被从 SD64 到 SD71 中删除,排在存储了删除的 F 编号的数据寄存器后面的数据寄存器的内容被移位到前一个数据寄存器
•通过执行 LEDR 指令,SD64 到 SD71 中的内容被向上移动一档，(对 A3N,A3HCPU,可以使用 CPU 面板上的 INDICATOR RESET 开关来执行)
•当检测到 8 个信号报警器时,即使再检测到第九个,也不会存储到 SD64 到 SD71 中
设置 F50 设置 F25 设置 F99 设置 F25 设置 F15 设置 F70 设置 F65 设置 F38 设置 F110 设置 F151 设置 F210 LEDR
SD62 0 50 50 50 50 50 50 50 50 50 50 50 99
SD63 0 1 2 3 2 3 4 5 6 7 8 8 8
SD64 0 50 50 50 50 50 50 50 50 50 50 50 99
SD65 0 0 25 25 99 99 99 99 99 99 99 99 15
SD66 0 0 0 99 0 15 15 15 15 15 15 15 70
SD67 0 0 0 0 0 0 70 70 70 70 70 70 65
SD68 0 0 0 0 0 0 0 65 65 65 65 65 38
SD69 0 0 0 0 0 0 0 0 38 38 38 38 110
SD70 0 0 0 0 0 0 0 0 0 110 110 110 151
SD71 0 0 0 0 0 0 0 0 0 0 151 151 210</td></tr>
<tr><td>D9126</td><td>SD1126</td><td>SD65</td></tr>
<tr><td>D9127</td><td>SD1127</td><td>SD66</td></tr>
<tr><td>D9128</td><td>SD1128</td><td>SD67</td></tr>
<tr><td>D9129</td><td>SD1129</td><td>SD68</td></tr>
<tr><td>D9130</td><td>SD1130</td><td>SD69</td></tr>
<tr><td>D9131</td><td>SD1131</td><td>SD70</td></tr>
<tr><td>D9132</td><td>SD1132</td><td>SD71</td></tr>
</table>

App - 50

附录 3 中断指针编号及中断因素表

I No.	中断因素		优先级	I No.	中断因素		优先级
I0	QI60/A1SI61 中断模块因素	第 1 点	237	I32*2	出错因素*3*4	停止运行的错误	1
I1		第 2 点	238	I33		空	——
I2		第 3 点	239	I34		模块核对出错. 保险丝熔断 特殊模块出错	2
I3		第 4 点	240				
I4		第 5 点	241				
I5		第 6 点	242	I35		操作出错 SFCP 操作出错 SFCP ECE 出错 外部电源关闭	3
I6		第 7 点	243				
I7		第 8 点	244				
I8		第 9 点	245				
I9		第 10 点	246	I36		ICM 操作出错 文件操作出错	4
I10		第 11 点	247				
I11		第 12 点	248	I37		空	——
I12		第 13 点	249	I38		程序超时	5
I13		第 14 点	250	I39		CHK 指令执行信号报警器检测	6
I14		第 15 点	251				
I15		第 16 点	252	I40 - I49	——	空	——
I16	顺序启动发生器模块因素*1	第 1 点	224				
I17		第 2 点	225				
I18		第 3 点	226				
I19		第 4 点	227	I50 - I255	智能型功能模块*5	用参数指定使用的智能型功能模块	18 - 223
I20		第 5 点	228				
I21		第 6 点	229				
I22		第 7 点	230				
I23		第 8 点	231				
I24		第 9 点	232				
I25		第 10 点	234				
I26		第 11 点	235				
I27		第 12 点	236				
I28	内部定时器因素*2	100ms	256				
I29		40ms	255				
I30		20ms	254				
I31		10ms	253				

备注

1) *1 : 第一点至第 12 点依次分配从靠近 QCPU 安装的顺序启动发生器模块开始。
2) *2 : 表示的内部时间是缺省的设定时间。
这些时间可以由 GDPW 指定(PLC 参数设定中的 PLC 系统设定)，从 0.5ms 到 1000ms，以 0.5 ms 为单位。
3) *3 : 当 132(停止运行出错)出错中断发生时，直到 132 处理完成，QCPU 不会停止运行。
4) *4 : 当电源接通和 QCPU 复位期间，从中断指针 132 到 139，禁止执行出错中断。当使用中断指针 132 到 139 时，使用 IMASK 指令来设置中断允许状态。

备忘录

App - 52

索引

[A]
初始扫描时间精度 4-16
扫描时间精度 4-18
信号报警器 10-12
ASCII 代码 4-51
ATA 卡 6-11
自动模式 5- 3
自动写入标准 ROM 6-14

[B]
B (通信继电器) 10-17
基本模式 5- 3
BCD (两进制编码十进制数) 4-47
BIN (两进制代码) 4-45
BL (SFC 块软元件) 10-58
引导运行 6-17

[C]
C (计数器) 10-24
PLC 目录存储器 6-18
字符串 4-51
时钟功能 7- 9
精度 7-11
2000 年问题 7-11
公共指针 10-54
I/O 分配的概念 5- 8
使用 GX Developer 分配
I/O 的概念 5-12
恒定扫描 7-2
常数 10-61
计数器度(C) 10-24
计数操作 10-24
最大计数速度 10-25

[D]
D (数据寄存器) 10-28
数据寄存器(D) 10-28
存储在存储卡上的数据 6- 4
十进制常数(K) 10-61
软元件初始值 10- 69
软元件表 10- 1
直接访问输入 10- 6
直接访问输出 10- 9
直接模式 4-41
驱动器编号. 6- 5
责任 10-25
DX (直接访问输入) 10- 6
DY (直接访问输出) 10- 9

[E]
E (初数) 10-62
边沿继电器(V) 10-16
END 处理 4-34
强制 ON/OFF 7-31
用于外部 I/O 的强制 ON/OFF 1- 4
执行类型 4-10
执行时间测量 7-39
低速执行型程序的执行时间 4-19
扩展 6-2, 6-4

[F]
F 信号(报警器) 10-12
错误历史 7-64
FD (功能寄存器) 10-31
文件寄存器 10-43
访问方法 10-44
指定方法 10-49
注册 10-45
文件大小 6-18
固定扫描执行型程序 4-31
闪存卡 6-12
十进制浮点数据 4-48
功能软元件(FX, FY, FD) 10-31
功能版本 13-5, 14-4
FX (功能输入) 10-31
FY (功能输出) 10-31

[G]
全局软元件 10-63
GX 配制器 8-2, 14-12
GX Developer A-18

[H]
H (十六进制常数) 10-61
HEX (十六进制) 4-46
十六进制常数(H) 10-61
高速累积定时器(ST) 10-21
高速定时器(T) 10-20

Index - 1

[I]
I (中断指针)..... 10-56
I/O No. 指定软元件(Un) 10-59
变址寄存器(Z)..... 10-39
初始执行监视时间 4-16
初始执行型程序 4-15
初始执行时间..... 4-16
输入响应时间..... 7-21
智能型功能模块软件(U □ \G □) 10-38
内部继电器(M)..... 10-10
内部系统软元件..... 10-31
内部用户软元件..... 10- 3
中断模块 5-11
中断指针(I)..... 10-56
中断程序 4- 6

[J]
J (网路指定软元件)..... 10-58
J □ \B □ (通信继电器) 10-35
J □ \SB □ (链接特殊继电器) 10-35
J □ \SW □ (链接特殊寄存器) 10-35
J □ \W □ (通信寄存器) 10-35
J □ \X □ (链接输入) 10-35
J □ \Y □ (链接输出) 10-35

[K]
K (十进制常数) 10-61

[L]
L (锁存继电器)..... 10-11
锁存功能 7- 5
锁存继电器(L)..... 10-11
LED 显示器 7-74
链接直接软元件..... 10-35
通信寄存器(W) 10-29
通信继电器(B) 10-17
中断因素列表..... 10-57
App-55
本地软元件 10-63
低速 END 处理..... 4-23
低速执行监视时间 4-24
低速执行型程序..... 4-19
低速累积定时器(ST)..... 10-21
低速扫描定时器 4-23
低速定时器(T)..... 10-19

[M]
M (内部继电器) 10-10
宏指令变量软元件(VD) 10-60
主程序.....4- 3
存储卡.....6-11
监视条件设定 7-25
监视本地软元件.....7-30

[N]
N (嵌套) 10-52

[O]
输出(Y) 10- 8

[P]
P (指针) 10-53
口令 7-65
PLOW 指令 4-14
POFF 指令 4-14
指针（P） 10-53
使用软元件初始值的注意事项 10-71
使用定时器时的注意事项.....10-23
LED 的优先级.....7-76
使用软元件初始值的步骤.....10-70
信号报警器 OFF 时的处理.....10-14
信号报警器 ON 时 10-12
程序结构.....1- 6
程序执行类型 4-10
程序存储器 6- 6
程序监视器表 7-39
PSCAN 指令 4-14
PSTOP 指令.....4-14
I/O 分配的用途 5-11
分配 I/O 的用途
使用 GX Developer.....5-11

[Q]
QCPU A-18
QI60..... 2-4,7-23
QnCPU A-18
QnHCPU A-18

[R]
R (文件寄存器).....10-43
从时间数据读取.....7-9
实数(E) 4-48
10-62
刷新输出.....10- 6

Index - 2

刷新模式 4-38
刷新输出 10- 9
远程锁存清除 7-19
远程运行 7-12
远程 PAUSE 7-15
远程口令 ..... 7 –1, 7-67
远程 RESET 7-17
远程 RUN/STOP 7-12
远程工作站 I/O 地址 5-10
累计定时器(OUT ST □) 10-21
运行状态 4-35

[S]
S (步进继电器) 10-18
SB (特殊通信继电器) 10-18
扫描执行型程序 4-17
扫描时间 4-18
SD (特殊寄存器) 10-34
SD415 (2n ms 时钟设定) 8- 9
SD520, SD521 (扫描时间：当前值) 4-18
SD522, SD523 (初始扫描时间) 4-16
SD524, SD525 (扫描时间：最大值) 4-18
SD526, SD527 (扫描时间：最小值) 4-18
SD528, SD529 (低速扫描时间：当前值) 4-24
SD530, SD531 (L 低速扫描时间：初始值) 4-24
SD532, SD533 (低速扫描时间：最小值) 4-24
SD534, SD535 (低速扫描时间：最大值) 4-24
自诊断功能 7-59
顺控程序 4- 1
系列 NO 1- 1
内部用户软元件设定范围 10- 3
阶段编号的设定 5- 2
SFC 块软元件(BL) 10-58
SFC 交换软元件(TR) 10-58
单一精度十进制浮点数据 4-48
大小（文件容量） 6- 2
SM（特殊继电器） 10-33
SM415 (2n ms 时钟) 8- 9
特殊通信寄存器(SW) 10-30
特殊通信继电器(SB) 10-18
特殊寄存器(SD) 10-34
特殊继电器(SM) 10-33
SRAM 卡 6-11
ST (累积定时器: OUT ST □) 10-21
待机型程序 4-25
标准 RAM 6- 9
标准 RAM 存储器容量 ..... 6 -9
标准 ROM 6- 8
步进继电器(S) 10-18
子程序 4- 4
SW (特殊通信寄存器) 10-30
智能型功能模块的开关设定 7-21
系统保护 7-65

[T]
T (定时器) 10-19
定时器(T) 10-19
精度 10-22
处理 10-22
TR (SFC 交换软元件) 10-58

[U]
U (I/O No. 指定软元件) 10-59
U □ \G □ (智能型功能模块软元件) 10-38
用户存储器 6- 3

[V]
V (边沿继电器) 10-16
VD (宏指令变量软元件) 10-60

[W]
W (通信寄存器) 10-29
警戒定时器 7-57
WDT (警戒定时器) 7-57
RUN 时写入 ..... 7-35，7-37，7-55
写入时间数据 7- 9

[X]
X (输入) 10- 5

[Y]
Y (输出) 10- 8
2000 年问题 7-11

[Z]
Z (变址寄存器) 10-39
ZR (文件寄存器的串行号码访问格式) 10-49

Index - 3

质保

使用之前请确认下述产品质保的细节：

1. 免费质保期限和免费质保范围

如果在质保期内使用本产品时发现因[三菱电机]的责任而导致的异常或缺陷（下文简称为“故障”），则该产品应该由经销商或[三菱电机]维修公司免费维修。注意如果需要在海外、孤立的岛屿或者偏远地方，则要收取派遣工程师的费用。

[免费质保期]

本产品的免费质保期为一年，自购买或货到目的地的日期起算。

注意从制造并运出[三菱电机]开始，最长分销时间不得超过 6 个月，从制造之日开始的最长免费质保期不得超过 18 个月。修理零件的免费质保期不得超过修理以前的免费质保期。

[免费质保范围]

(1) 范围被限制在按照使用手册、用户手册和产品上的警示标贴上规定的使用状态、使用方法和使用环境正常使用的条件下。

(2) 即使在免费质保期内，下列情况下修理要收费。

1. 因不合理存储或搬运、用户的大意或疏忽而导致的故障。因用户的硬件或软件设计而导致的故障。
2. 因用户未经批准对该产品进行改造而引起的故障。
3. 如果把[三菱电机]产品装配在用户设备中，如果本公司供了用户设备根据法律安全条款或工业标准要求必需的功能和结构，故障本来可以避免时。
4. 如果正确采用或更换了用户手册中指定的耗材（电池、背光灯、保险丝等）故障本来可以避免时。
5. 因火灾、不正常电压和因地震、雷电、大风和水灾等不可抗力引发的故障。
6. 按照科学技术标准在产品从[三菱电机]运出时不能预测的原因而导致的故障。
7. 任何不是因[三菱电机]或用户的责任而导致的故障。

2. 停止产品生产以后的有偿修理条款

(1) ［三菱电机］在本产品停止生产后的 7 年内受理对该产品的有偿修理。停止生产的消息将以［三菱电机］技术公告等方式予以通知。

(2) 生产停止以后，不再提供产品（包括修理用零部件）。

3. 海外服务

在海外，修理由［三菱电机］在当地的海外 FA 中心受理。请注意各个 FA 中心的修理条件可能会有所不同。

4. 意外损失和间接损失不在质保责任范围内

不论是否在免费质保期内，［三菱电机］对任何不是［三菱电机］的责任的原因而引起的损失、意外损失、因［三菱电机］产品故障而导致的利润损失、违反［三菱电机］要求的特殊原因而引起的损失或间接损失、事故赔偿、及非［三菱电机］的其它产品的损坏和赔偿等不承担责任。

5. 产品规格的改变

目录、手册或技术文档中的规格的改变不事先通知。

6. 产品应用

(1) 在使用［三菱电机］MELSEC 可编程逻辑控制器时，应该符合下列条件：即使可编程逻辑控制器出现问题或故障也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设施和失效保险功能。

(2) 三菱通用可编程序控制器是应一般工业用的。因此，可编程序控制器的应用不包括那些会影响公众的应用如核电厂和其他由独立供电公司经营的电厂以及需要特殊质量控制系统的应用如铁路公司或用于国防目的的应用。

请注意即使是这些应用，假如用户同意该应用受限制并且不需要特别质量的话，仍然可以作这类应用。

在用于航空、医学、铁路、焚烧和燃料设备，传送人的设备，娱乐和休闲设施和安全设施等与人的生命财产密切相关以及在安全和控制系统方面需要特别高的可靠性时，请与三菱公司联系并讨论所需规格。

Q系列CPU

用户参考手册

型号	QCPU(Q)-U(KI)-CH
	SH(NA)-080232C-A

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : 1-8-12, OFFICE TOWER Z 14F HARUMI CHUO-KU 104-6212,TELEX : J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.
Printed in Japan on recycled paper.