



GE Fanuc Automation

**与90-70™系列PLC
CPU 指令集**

参 考 手 册

本手册中的“警告”、“小心”和“说明”

警告

本册中此提示用于对危险电压、电流、温度或其他可能对人身造成伤害的场合。
在疏忽情况下，可能对使用者造成人身伤害或对设备造成损坏。

小心

在疏忽情况下，可能对设备造成损坏。

说明

此提示仅用于提示使用者一些重要信息，尤其是对于要点的理解和对相关设备的操作。

本手册的内容基于该版本出版时的信息。内容力求精确，但是难以涵盖软硬件所有细节和变更信息，也不能提供涉及安装、运行或维护的所有可能的偶然事件，也可能提及所有软硬件系统中均不存在的特性。此手册涉及到的产品信息如有更改恕不提醒，GE Fanuc公司对此不做任何形式的表述，并对包含于此的信息的准确性、完整性、充分性或有效性不承担责任。未经授权不得用于商业和其他应用。

以下均为GE Fanuc自动化系统北美公司注册商标：

Alarm Master	Genius	ProLoop	Series Three
CIMPLICITY	Helpmate	PROMACRO	VersaMax
CIMPLICITY 90-ADS	Logicmaster	Series Five	VersaPro
CIMSTAR	Modelmaster	Series 90	Vumaster
Field Control	Motion Mate	Series One	Workmaster
Genet	PowerTRAC	Series Six	

本手册主要介绍系列 90™-70 可编程序控制器的系统操作，故障处理和 Logicmaster 9070 的编程指令集。系列 9070 PLC 是 GE Fanuc 自动化公司可编程逻辑控制器 90™系列家族的一员。

手册修订说明

以下是本手册所反映的特性变更，修正和现有信息的更新：

- CPX 和 CGR 型处理器的参考地址，应通过 $y^{\prime} \text{ } ^{\prime \prime} \text{ } ^{\prime \prime \prime} \text{ } ^{\prime \prime \prime \prime}$ 表示。
- 定时扫描方式的计时器值修正（第二章，2-46 页）。
- 表 2-18 后增加了关于 CPU 运行方式选择开关的说明和并更新了表中 level 1 优先级的描述（第二章，2-79 页）。
- 系统错误描述更新（第三章，3-2 页）。
- 第四章至第十二章内容在前一版本中包含在一独立章节（第四章）中。这些内容现分为多个独立章节，以便逐步深入对编程指令进行介绍。
- 附录 A，CPU 运行数据中的表格修正（其他未明确的信息将在下一版本中注明）。
- 增加了“每个以太网全局……”一段，A-24 页。
- 附录 A 的最后增加了一部分，标题为：“相对 CPU 性能测试”。
- 其他必须的修正和说明。

本手册内容

第一章 介绍：系列 9070 系统和指令集综述。

第二章 系统操作：介绍了系列 9070 系统的有关操作。包括 PLC 系统的扫描时序，系统上电、掉电顺序，时钟和计时器、安全性、I/O 端口及故障处理。其中包含了理解基本梯形图编程逻辑的一般信息。

第三章 故障说明与更正：提供了系列 9070 PLC 的系统故障诊断和处理方法。这里详细说明了 PLC 故障表中的错误描述和 I/O 故障表中的错误类别。

第四章 至 第十二章：系列 9070 编程指令集：介绍了 9070 的编程指令，这些章节同主程序功能组相对应。

附录 A CPU 运行数据：表中列出了每个编程指令占用的内存字节数及执行时间（ms）。内存字节数指梯形图应用程序中功能要占用的字节数。该附录也包含其他 PLC 任务占用时间的相关信息，可用于预计 CPU 扫描时间。附录 F 列出了用于浮点运算的 IEEE 标准格式。

附录 B 故障表的一般解释：说明利用 Logicmaster 9070 读故障表时如何解释故障信息。

附录 C 指令助记符：在查找或编辑程序时，可用助记符来调用的编程指令清单。

附录 D 内存分配：提供一个工作表单，用于了解用户已使用的全部内存的字节数以及可供用户程序使用的内存量。

附录 E 快捷键功能：提供可用于 Logicmaster 9070 软件的特殊功能键的列表。
附录 F 使用浮点数：说明使用浮点运算时要特别考虑的内容。

相关手册

Logicmaster™ 9070 编程软件使用手册 (GFK0263)
Logicmaster™ 9070 重要产品信息 (GFK0350)
系列 90™- 70 可编程序控制器安装手册 (GFK0262)
系列 90™可编程序协处理器模块及其软件用户手册 (GFK0255)
系列 90™ PCM 开发软件 (PCOP) 使用手册 (GFK0487)
90™- 70 PLC 程序工具包使用手册 (GFK0646)
系列 90™时序流程图编程语言 (SFC) 使用手册 (GFK0854)
MegaBasic™编程语言参考手册 (GFK0256)
CIMPLICITY™ 90 符号显示系统 (ADS) 使用手册 (GFK0499)
CIMPLICITY™ 90 符号显示系统 (ADS) 参考手册 (GFK0641)
符号显示协处理器模板一览表 (GFK0521)
系列 90™- 70 Genius I/O 系统使用手册 (GEK904861)
系列 90™- 70 Genius I/O 模拟量和开关量模块使用手册 (GEK904862)
Workmaster® II PLC 编程单元操作向导 (GFK0401)
系列 90™- 70 Genius 总线控制器使用手册 (GFK0398)
系列 90™- 70 FIP 总线控制器使用手册 (GFK1038)
第三方 VME 模板选择指南 (GFK0448)
系列 90™以太网通讯使用手册 (GFK0868)
系列 90™ MAP 3.0 通讯使用手册 (GFK0869)
系列 90™ PLC TCP/IP 以太网通讯使用手册 (GFK-1541)

第一章 概述.....	1-1
软件体系.....	1-1
使用的术语	1-1
故障处理.....	1-2
硬件配置	1-2
使用本手册	1-2
第二章 操作系统.....	2-1
第一节 基本PLC扫描概述.....	2-2
基本PLC 扫描	2-3
内务处理	2-4
输入扫描	2-4
应用程序任务执行(逻辑窗口)	2-5
输出扫描.....	2-5
编程器通讯窗口	2-6
系统通讯窗口	2-7
后台窗口.....	2-7
窗口模式	2-8
通讯窗口中的数据一致性	2-8
STOP模式下的CPU扫描.....	2-9
PLC 扫描模式	2-10
第二节 用户变量数据	2-11
用户变量.....	2-11
间接变量	2-11
用户变量大小及其默认值.....	2-14
用户变量%G 和CPU内存定位	2-15
Genius全局数据.....	2-15
跳变和强制	2-16
逻辑和程序的保持性	2-16
数据作用域.....	2-18
数据类型	2-19
系统状态变量	2-20
其他变量.....	2-25
第三节 程序的组织	2-27
梯形图逻辑的编写.....	2-28
主程序块.....	2-29
程序块.....	2-30
使用程序块举例	2-30
如何调用程序块.....	2-33
参数子程序块.....	2-35
参数子程序块和局部数据	2-35
如何调用参数子程序块.....	2-36

参数子程序块中的形式参数变量	2-37
参数子程序块中的形式参数的使用限制.....	2-38
PSB中的位型参数.....	2-38
外部程序块	2-39
如何调用外部程序块	2-39
外部程序块和局部数据.....	2-40
局部数据初始化	2-40
独立C程序.....	2-41
数据封装.....	2-41
输入/输出列表.....	2-42
独立C程序和局部数据	2-44
局部数据初始化.....	2-44
在独立C程序中使用I/O表数据.....	2-45
I/O 列表数据一致性.....	2-45
梯形图程序vs. 独立C程序	2-46
操作上的不同点: 梯形图程序和独立C程序.....	2-46
数据的保持性.....	2-46
全局数据	2-47
中断执行	2-47
中断队列	2-47
系统状态变量.....	2-47
第四节 PLC扫描方式和程序时序模式.....	2-48
一般扫描方式	2-48
恒定时间扫描方式.....	2-49
恒定窗口扫描方式.....	2-50
微循环扫描方式.....	2-51
微循环扫描方式下的输出扫描估算.....	2-53
准7.00 CPU输出扫描预算	2-53
7.00或更新版CPU输出扫描预算	2-54
选择PLC扫描方式和程序时序模式.....	2-56
用户程序执行.....	2-56
用户程序优先级.....	2-56
用户程序在一般扫描方式, 恒定时间, 恒定窗口扫描方式下的用户程序执行.....	2-57
微循环扫描方式下的程序执行.....	2-60
中断处理	2-64
中断处理和程序块时序	2-64
I/O中断程序块.....	2-65
定时中断程序块.....	2-66
中断处理和用户程序时序	2-67
I/O触发程序.....	2-67
定时程序	2-68
中断块 vs. 中断程序.....	2-69

第五节 Run/Stop操作.....	2-70
运行模式	2-70
模式转换	2-71
Stop -Run转换	2-71
Run -Stop转换	2-71
微循环模式下的掉电周期.....	2-71
第六节 上电和掉电时序.....	2-72
上电时序	2-72
上电自检	2-72
PLC 内存确认	2-72
系统配置	2-73
智能可选模块自检完成.....	2-73
智能可选模块双端口测试	2-73
I/O系统初始化.....	2-74
掉电时序	2-74
掉电后的数据内存的保持.....	2-75
第七节 时钟和计时器.....	2-76
计时时钟.....	2-76
日历时钟.....	2-76
看门狗计时器	2-77
软件看门狗计时器.....	2-77
硬件看门狗计时器.....	2-77
第八节 系统安全性	2-78
密码和优先级	2-78
编程器的保护级请求.....	2-79
禁用密码.....	2-80
OEM保护.....	2-80
写保护开关	2-80
第九节 系列90-70 PLC I/O系统.....	2-81
I/O数据映射	2-82
缺省状态	2-82
Genius I/O.....	2-82
Genius I/O总线配置	2-82
Genius I/O数据映射	2-82
模拟量分组块.....	2-83
低等模拟量块	2-83
缺省状态	2-83
Genius全局数据通讯.....	2-84
FIP I/O	2-84
FIP I/O 总线配置.....	2-84
FIP I/O 数据映射.....	2-85

默认状态	2-85
诊断数据收集	2-85
离散量I/O诊断信息	2-86
模拟量I/O诊断数据	2-86
第三章 故障解释及纠正.....	3-1
第一节 系统故障处理(General)	3-2
系统故障变量	3-3
配置故障作用	3-4
不可配置故障	3-5
故障接点.....	3-6
故障定位变量(机架, 槽, 总线, 模板).....	3-7
故障变量的格式.....	3-7
故障变量的行为.....	3-8
报警接点.....	3-8
点故障	3-9
第二节 故障处理.....	3-10
报警处理器	3-10
故障等级	3-11
系统对故障的响应	3-11
故障表.....	3-11
故障作用	3-12
故障响应	3-12
PLC故障表	3-13
用户定义故障.....	3-13
I/O故障表	3-14
获取附加故障信息.....	3-15
第三节 PLC故障表解释	3-16
可配置故障	3-17
机架未接通或丢失	3-17
可选模块未接通或丢失.....	3-18
新增或多余机架	3-21
重启, 新增或多余可选模块	3-21
系统配置不匹配	3-22
系统总线错误.....	3-25
PLC CPU硬件故障.....	3-26
模板硬件故障	3-26
可选模块软件故障	3-27
程序或程序块检测故障.....	3-28
电池电量低信号	3-28
恒定时间或微循环扫描超时	3-29

PLC 系统故障表满	3-29
I/O 故障表满	3-29
应用程序故障	3-30
不可配置故障	3-31
系统总线故障.....	3-32
上电时无用户程序.....	3-32
上电时用户程序冲突.....	3-33
窗口完成故障.....	3-34
口令错误.....	3-34
运行时无系统配置	3-34
PLC CPU 系统软件故障.....	3-35
总线控制器过多	3-36
存储期间通讯故障	3-36
运行模式存储失败.....	3-37
第四节 I/O故障表解释	3-38
回路故障	3-41
离散量故障	3-42
模拟量故障.....	3-43
低等模拟量故障.....	3-44
GENA故障	3-45
IOC (I/O控制器)未接通.....	3-45
新增IOC (I/O控制器)	3-45
I/O模板未接通.....	3-46
新增I/O模板	3-46
多余I/O模板.....	3-46
块未接通	3-47
新增块.....	3-47
多余块.....	3-47
I/O总线故障	3-48
模板故障	3-49
IOC (I/O控制器)软件故障	3-49
IOC (I/O控制器)硬件故障	3-50
强制和解除强制回路.....	3-50
块开关	3-50
第四章 继电器功能块	4-1
使用接点	4-2
使用线圈.....	4-3
常开接点 $\text{—} \text{—}$	4-4
常闭接点 $\text{—} \text{—}$	4-4
正跳变接点 $\text{—} \uparrow \text{—}$	4-4
负跳变接点 $\text{—} \downarrow \text{—}$	4-4

故障接点 -[FAULT]-.....	4-7
无故障接点 -[NOFLT]-.....	4-7
高报警接点 -[HIALR]-.....	4-7
低报警接点 -[LOALR]-.....	4-7
线圈 -()-.....	4-8
负向线圈 -(/)-.....	4-8
保持线圈 -(M)-.....	4-8
负向保持线圈 -(/M)-.....	4-8
正跳变线圈 -(↑)-	4-9
负跳变线圈 -(↓)-	4-9
置位线圈 -(S)-	4-10
复位线圈 -(R)-.....	4-10
保持型置位线圈 -(SM)-.....	4-11
保持型复位线圈 -(RM)-.....	4-11
连接线.....	4-11
续行线圈 (- - <+>)和续行接点(<+>- - -)	4-12
第五章 计时器和计数器.....	5-1
计时器和计数器功能块所需数据.....	5-1
ONDTR.....	5-3
OFDT.....	5-6
TMR	5-9
UPCTR	5-12
DNCTR.....	5-14
第六章 数学运算功能块	6-1
MATH (ADD, SUB, MUL, DIV).....	6-2
MOD (INT, UINT, DINT).....	6-4
SQRT (INT, DINT, REAL)	6-6
ABS (INT, DINT, REAL).....	6-8
三角运算功能块(SIN, COS, TAN, ASIN, ACOS, ATAN)	6-10
对数/指数功能块(LOG, LN, EXP, EXPT)	6-12
弧度转换功能块(RAD, DEG)	6-14
第七章 关系运算功能块.....	7-1
EQ, NE, GT, GE, LT和LE (INT, UINT, DINT, REAL)	7-2
CMP (INT, UINT, DINT, REAL)	7-4
RANGE (INT, UINT, DINT, WORD, DWORD).....	7-6
第八章 位操作功能块.....	8-1
AND和OR (WORD, DWORD)	8-3
XOR (WORD, DWORD)	8-5

NOT(WORD, DWORD).....	8-7
SHL和SHR (WORD, DWORD)	8-9
ROL和ROR (WORD, DWORD).....	8-12
BTST (WORD, DWORD)	8-14
BSET和BCLR (WORD, DWORD).....	8-16
BPOS(WORD, DWORD).....	8-18
MCMP(WORD, DWORD).....	8-20
第九章 数据移动功能块	9-1
MOVE (INT, UINT, DINT, BIT, WORD, DWORD, REAL)	9-2
BLKMOV (INT, UINT, DINT, WORD, DWORD, REAL).....	9-4
BLKCLR(WORD).....	9-6
SHFR(BIT, WORD, DWORD).....	9-8
BITSEQ (BIT).....	9-11
SWAP (WORD, DWORD).....	9-15
COMMREQ	9-17
VMERD (BYTE, WORD)	9-25
VMEWRT (BYTE, WORD).....	9-27
VMERMW (BYTE, WORD).....	9-29
VMETST (BYTE, WORD).....	9-31
VME_CFG_RD.....	9-34
VME_CFG_WRITE.....	9-37
DATA_INIT (INT, UINT, DINT, WORD, DWORD, REAL)	9-40
进入DATA_INIT_type功能块.....	9-42
DATA_INIT_COMM.....	9-43
进入DATA_INIT_COMM功能块.....	9-45
DATA_INIT_ASCII.....	9-46
进入DATA_INIT_ASCII功能块.....	9-47
DATA_INIT_DLAN	9-48
第十章 数据表功能块	10-1
移入和移出表.....	10-1
TBLRD (INT, UINT, DINT, WORD, DWORD).....	10-3
TBLWRT (INT, UINT, DINT, WORD, DWORD)	10-5
LIFORD (INT, UINT, DINT, WORD, DWORD)	10-7
LIFOWRT (INT, UINT, DINT, WORD, DWORD)	10-9
FIFORD (INT, UINT, DINT, WORD, DWORD).....	10-11
FIFOWRT (INT, UINT, DINT, WORD, DWORD)	10-13
SORT (INT, UINT, WORD).....	10-15
ARRAY_MOVE (INT, UINT, DINT, BIT, BYTE, WORD, DWORD).....	10-17
SRCH_EQ和SRCH_NE (INT, UINT, DINT, BYTE, WORD, DWORD)	
SRCH_GT和SRCH_LT SRCH_GE和SRCH_LE.....	10-21

ARRAY RANGE (INT, DINT, WORD, DWORD).....	10-24
第十一章 转换功能块.....	11-1
BCD-4 (INT, UINT).....	11-2
BCD-8 (DINT)	11-4
UINT (INT, DINT, BCD-4, REAL)	11-6
INT (UINT, DINT, BCD-4, REAL)	11-8
DINT (INT, UINT, BCD-8, REAL)	11-10
REAL (INT, UINT, DINT, BCD-4, BCD-8)	11-12
TRUN (INT, DINT).....	11-14
第十二章 控制功能块.....	12-1
CALL	12-3
CALL EXTERNAL.....	12-4
CALL SUBROUTINE.....	12-6
DOIO	12-10
SUSIO.....	12-14
MCR	12-16
ENDMCR	12-17
JUMP	12-18
LABEL.....	12-19
COMMENT	12-20
FOR, END_FOR 和 EXIT	12-21
SVCREQ.....	12-25
SVCREQ #1: 更改/读恒定时间扫描定时器	12-28
SVCREQ #2: 读窗口值.....	12-31
SVCREQ #3: 更改编程器通讯窗口模式和定时器值.....	12-32
SVCREQ #4: 更改编程器通讯窗口模式和定时器值.....	12-33
SVCREQ #5: 更改后台任务窗口模式和定时器值	12-34
SVCREQ #6: 更改/读检测任务状态和检测字数.....	12-36
SVCREQ #7: 更改/读日历时钟状态和值.....	12-38
SVCREQ #8: 复位看门狗定时器	12-42
SVCREQ #9: 从开始扫描读扫描时间	12-43
SVCREQ #10: 读文件夹名.....	12-44
SVCREQ #11: 读PLC ID.....	12-45
SVCREQ #12: 读PLC 运行状态	12-46
SVCREQ #13: 停止(Stop) PLC	12-47
SVCREQ #14: 清故障表.....	12-48
SVCREQ #15: 读最新的故障表条目.....	12-49
SVCREQ #16: 读消逝的时钟.....	12-53

SVCREQ #17: 屏蔽/解除屏蔽 I/O 中断	12-54
SVCREQ #18: 读I/O 强制状态.....	12-56
SVCREQ #19: 设置运行的启用/禁用	12-57
SVCREQ #20: 读故障表.....	12-58
SVCREQ #21: 用户定义故障信息.....	12-62
SVCREQ #22: 屏蔽/解除屏蔽定时中断.....	12-64
SVCREQ #23: 读主检测.....	12-65
SVCREQ #25: 禁用/启用EXE块和独立C程序检测.....	12-67
SVCREQ #26: 任务开关	12-68
SVCREQ #27 和 #28: 写/读 反传送区域.....	12-69
SVCREQ #32: 延迟/恢复I/O中断.....	12-70
SVCREQ #39: ESCM 端口状态.....	12-72
返回值.....	12-73
SVCREQ #44: 逻辑驱动同步以太网全局数据.....	12-74
服务请求功能块.....	12-74
返回状态值.....	12-75
服务请求命令描述.....	12-75
命令2 – 找回本地发送者ID	12-76
命令3 和4 – 建立发送者交换/接收者交换.....	12-77
建立一个发送者交换命令的格式.....	12-78
建立一个接收者交换命令的格式.....	12-80
命令5 和 6 – 终止发送者交换/接收者交换.....	12-82
命令7 – 每次扫描更新发送数据.....	12-83
逻辑驱动以太网全局数据的其他注意事项.....	12-84
交换状态字.....	12-84
PID	12-88

附录 A CPU 运行数据 A-1

指令定	
时.....	A-1
超时扫描影响时间.....	A-10
表中的内容.....	A-10
基本扫描时	
间.....	A-11
编程器扫描影响时间.....	A-12
I/O 扫描和I/O 故障扫描影响.....	A-14
系列90-70 I/O 模板影响.....	A-14
Genius I/O和GBC扫描影响.....	A-17
FIP I/O 和FBC扫描影响.....	A-21
以太网全局数据扫描影响.....	A-24
智能可选模块扫描影响.....	A-26
I/O 中断运行和扫描影响.....	A-27
定时中断影响.....	A-30
预算扫描时间举例.....	A-31
小型系统.....	A-31
大型系统.....	A-33

相关CPU运行比较.....	A-38
测试程序.....	A-38
本章解释.....	A-38
附录 B 使用Logicmaster 90-70 软件解释故障.....	B-1
附录 C 指令助记符.....	C-1
附录 D 内存分配.....	D-1
故障表.....	D-2
以太网全局数据.....	D-2
C 测试连接.....	D-2
I/O 扫描设置文件	D-2
模板配置文件.....	D-2
名称解析文件.....	D-2
用户协议文件.....	D-2
用户程序.....	D-2
用户程序内存的使用.....	D-3
附录 E 功能键.....	E-1
附录 F 使用浮点数.....	F-1

图 2-1. 典型PLC扫描的阶段.....	2-3
图 2-2. 在Stop/NoIO和Stop/IOScan模式下的CPU扫描.....	2-9
图 2-3. 一般扫描模式下的典型扫描.....	2-48
图 2-4. 恒定时间扫描模式下的典型扫描.....	2-50
图 2-5. 恒定窗口扫描模式下的典型扫描.....	2-51
图 2-6. 微循环扫描模式下的典型扫描.....	2-53
图 2-7. 顺序程序的执行时序.....	2-58
图 2-8. 顺序, 定时, I/O触发和中断块执行时序.....	2-59
图 2-9. 周期程序执行时序.....	2-60
图 2-10. 周期和I/O触发执行时序.....	2-62
图 2-11. I/O中断块的声明.....	2-65
图 2-12. 定时中断块的声明.....	2-66
图 2-13. 系列90-70 PLC I/O结构.....	2-81
图 12-1. 独立条件运算(PIDIND)	12-97
图 A-1. 相关CPU 运行图图.....	A-39

表 2-2.	寄存器变量.....	2-11
表 2-3.	离散变量.....	2-12
表 2-3.	离散变量- 续.....	2-13
表 2-4.	用户变量大小.....	2-14
表 2-5.	默认内存大小.....	2-14
表 2-6.	%G 变量和内存定位.....	2-15
表 2-7.	用户变量数据的数值范围.....	2-18
表 2-8.	数据类型.....	2-19
表 2-9.	系统状态变量.....	2-21
表 2-9.	系统状态变量 - 续.....	2-23
表 2-9.	系统状态变量 - 续.....	2-24
表 2-10.	系统故障变量.....	2-25
表 2-11.	可配置故障.....	2-25
表 2-12.	不可配置故障.....	2-26
表 2-13.	块类型.....	2-28
表 2-14.	I/O数据的一致性.....	2-45
表 2-15.	LD vs. 独立C 程序比较.....	2-46
表 2-16.	每种PLC扫描模式的可用程序时序模式.....	2-56
表 2-17.	定时和I/O触发程序的优先级值.....	2-57
表 2-18.	优先级.....	2-79
表 3-1.	系统故障变量.....	3-3
表 3-2.	可配置故障的故障变量.....	3-4
表 3-3.	不可配置故障.....	3-5
表 3-3.	不可配置故障 - 续.....	3-6
表 3-4.	故障变量名.....	3-7
表 3-5.	故障等级.....	3-11
表 3-6.	故障属性.....	3-11
表 3-7.	故障作用.....	3-12
表 3-8.	故障信息说明.....	3-39
表 3-8.	故障信息说明 - 续.....	3-40
表 3-9.	回路故障信息说明.....	3-41
表 4-1.	接点的类型.....	4-2
表 4-2.	线圈类型.....	4-3
表 12-1.	服务请求功能块.....	12-25
表 12-2.	SVCREQ #44功能块的一般格式.....	12-74
表 12-3.	命令的可能的命令状态.....	12-75

表 12-4. 设定本地发送者ID命令的格式.....	12-75
表 12-5. 设定本地发送者ID命令的命令状态.....	12-76
表 12-6. 恢复本地发送者ID命令的格式.....	12-76
表 12-7. 恢复本地发送者ID命令的命令状态.....	12-76
表 12-8. 建立交换命令的交换状态字.....	12-77
表 12-9. 建立发送者交换命令的格式.....	12-78
表 12-10. 建立发送者交换命令的命令状态.....	12-79
表 12-11. 建立接收者交换命令的格式.....	12-80
表 12-12. 建立接收者交换命令的命令状态.....	12-81
表 12-13. 终止发送者交换命令格式.....	12-82
表 12-14. 终止接收者交换命令格式.....	12-82
表 12-15. 终止交换命令的命令状态.....	12-83
表 12-16. 每次扫描刷新发送数据的格式.....	12-83
表 12-17. 每次扫描刷新发送数据的命令状态.....	12-84
表 12-18. 交换状态字地址的格式.....	12-85
表 12-19. 交换状态字值.....	12-85
表 12-20. 变量的格式.....	12-86
表 12-21. 时间标记地址的格式.....	12-86
表 12-22. POSIX 时钟时间标记格式.....	12-86
表 12-23. PLC 内存类型格式.....	12-87
表 12-4. PID 参数一览.....	12-90
表 12-4. PID 参数一览 - 续.....	12-91
表 12-5. PID 参数说明.....	12-93
表 12-5. PID 参数说明- 续.....	12-94
表 12-5. PID 参数说明- 续.....	12-95
表 A-1. 指令时间.....	A-2
表 A-1. 指令时间 - 续.....	A-3
表 A-1. 指令时间 - 续.....	A-4
表 A-1. 指令时间 - 续.....	A-5
表 A-1. 指令时间 - 续.....	A-6
表 A-1. 指令时间 - 续.....	A-7
表 A-1. 指令时间 - 续.....	A-8
表 A-1. 指令时间 - 续.....	A-9
表 A-2. 基本扫描 vs. 完全扫描状态.....	A-11
表 A-3. 基本扫描时间.....	A-12
表 A-4. 编程器扫描影响时间.....	A-12
表 A-5. I/O 扫描所占时间	A-14

表 A-6.	工作表 A: I/O 模板扫描时间.....	A-15
表 A-7.	70型I/O模板和机架扫描的影响时间 *	A-16
表 A-7.	70型I/O模板和机架扫描的影响时间- 续.....	A-17
表 A-8.	Genius I/O和GBC的扫描影响时间	A-19
表 A-9.	工作表B: Genius I/O 扫描时间.....	A-20
表 A-10.	FIP I/O和FBC扫描影响时间	A-22
表 A-11.	工作表B: FIP I/O 扫描时间.....	A-23
表 A-12.	工作表: 以太网全局数据扫描时间.....	A-25
表 A-13.	智能可选模块的固定扫描时间	A-26
表 A-14.	I/O 中断块运行和扫描影响时间.....	A-28
表 A-15.	I/O触发中断运行和扫描影响时间	A-28
表 A-16.	工作表C:编程器, IOM, I/O 中断扫描时间.....	A-29
表 A-17.	定时中断运行和扫描影响时间.....	A-30
表 A-18.	I/O触发中断运行和扫描影响时间.....	A-31
表 A-19.	工作表A.....	A-35
表 A-20.	工作表B.....	A-36
表 A-21.	工作表A示例.....	A-37
表 A-22.	工作表B示例.....	A-37
表 B-1.	PLC 故障组.....	B-4
表 B-2.	PLC 故障作用.....	B-5
表 B-3.	PLC CPU软件故障警报错误代码.....	B-6
表 B-4.	PLC CPU故障警报错误代码.....	B-7
表 B-4.	PLC CPU故障警报错误代码- 续.....	B-8
表 B-4.	PLC CPU故障警报错误代码- 续.....	B-9
表 B-5.	PLC故障附加数据 – 系统配置不符	B-10
表 B-6.	Genius块模板号.....	B-11
表 B-7.	GENA 应用ID 号.....	B-12
表 B-8.	Genius 安装模板I/O 类型.....	B-12
表 B-9.	Genius 配置模板I/O 类型	B-12
表 B-10.	故障特征数据 – 无效的Genius总线请求.....	B-13
表 B-11.	PLC 故障时间.....	B-14
表 B-12.	I/O 故障表格式表示字节.....	B-18
表 B-13.	I/O 参考地址.....	B-19
表 B-14.	I/O 变量地址内存类型.....	B-19
表 B-15.	I/O 故障组.....	B-20
表 B-16.	PLC 故障作用.....	B-21
表 B-17.	I/O 故障信息.....	B-22

表 B-18. I/O 故障类型..... B-23

表 B-18. I/O 故障类型 – 续..... B-24

表 B-19. I/O 故障说明..... B-24

表 B-19. I/O 故障说明 – 续..... B-25

表 B-20. I/O 故障特性数据..... B-27

表 B-20. I/O 故障特性数据 – 续..... B-28

表 B-21. I/O 故障时间..... B-28

系列 90™-70 PLC 是 GE Fanuc 系列 90™ PLC 家族可编程控制器中的一员。它易于安装和配置，提供了先进的编程功能，具有与其他系列 90 家族 PLC 完全兼容的特点。通过使用最新的设计和制造技术，开放式的 VME 总线结构以及 Genius 和 FIP I/O 的方便的连接能力，系列 90-70 PLC 为小规模到极大规模的系统应用提供了一个功能强大，价格合理的控制系统平台。本手册讨论的特性是最新系列 90-70 PLC，版本号为 7.80。

软件体系

编程软件体系提供了建立结构化控制程序的平台。程序可由许多程序块构成，每程序块完成相应的控制功能。结构化的程序允许由不同的开发人员和 OEM 商独立的开发程序块，然后组合到一起，并行开发成为一个完整的程序。结构化的程序也易于理解和调试。一个控制程序可以由多个小程序块组成，更易于实现控制功能程序隔离或连锁的控制逻辑。

从版本 6.0 的 PLC CPU 发布开始，合并多个程序到一个文件夹就成为可能。所有的这些程序可以用 C 程序编写，或用 RLD 或 SFC 语言编写的程序仍可调用 C 程序编写的程序块。另外，版本 6.0 或更新的 PLC 都内建了 C 程序和外部程序块的测试功能。如果需要了解此方面的更多信息，请参阅《C 程序工具包》手册(GFK-0646)。需要注意的是 Logicmaster 90-70 不支持 7.0 和更新 CPU 的许多新特性，如以太网全局数据通讯，I/O 扫描设定，第三方 VME 中断和大内存访问(BMA)等功能。

使用的术语

本手册中所使用的术语及所表示的含义表述如下：

用户程序：任何用户生成的代码，例如，一个梯形图程序，一个顺序流程图程序，或一个独立的 C 程序。

程序块：任何梯形图程序块，参数化子程序块，或外部程序块，外部程序块既可以是 C 程序块也可以是 C 功能块。

故障处理

由软件报警处理功能来处理故障，它可将系统故障和 I/O 故障按时间标明并分别记录于两种故障表（PLC 故障表和 I/O 故障表）中。这两个表可以在编程软件窗口中显示，也可上传至主机或者其他协同处理机中。

应用程序也可获取这些故障信息。

硬件配置

硬件配置就是给系统中的硬件模板分配逻辑地址的过程。它可以在编程之前也可以在编程之后进行，然而我们推荐首先进行硬件配置。

使用本手册

本手册分部分介绍了 Logicmaster 90 编程软件，详细阐述了目前 PLC 的 CPU 硬件和软件特性。对于 IPI 分类在本版手册中未予讨论。

本手册中的附加参考信息如下所列：

附录 A 的表中列出了每个编程指令占用的内存字节数及执行时间（ms）。内存字节数指梯形图应用程序中功能要占用的字节数。该附录也包含其他 PLC 任务占用时间的相关信息，可用于预计 CPU 扫描时间。

附录 B 描述了利用 Logicmaster 9070 读 PLC 和 I/O 故障表时如何解释故障信息。

使用附录 D 中工作表，可以了解用户已使用的全部内存的字节数以及可供用户程序使用的内存量。

附录 F 中当处理浮点运算时 IEEE 标准格式。

本章主要介绍了系列 90-70 PLC 系统的一些系统操作，各节的内容如下表所示：

节	标题	内容	页码
1	PLC 扫描概述	介绍了典型 PLC 扫描方式下的扫描过程的几个主要步骤，即应用程序执行，编程器通讯窗口，系统通讯窗口和后台窗口。	2-2
2	用户参考数据	介绍用户参考数据，系统状态和系统错误以及数据类型。	2-11
3	程序的组织	介绍梯形图程序块的结构和用法，参数化子程序块，外部程序块，以及独立的 C 程序。	2-27
4	PLC 扫描方式和程序时序方式	说明正常扫描方式、恒定时间扫描方式、恒定窗口扫描方式、多周期扫描方式和停止方式，并介绍了触发中断块/程序和定时中断。	2-48
5	运行/停止操作	介绍 9070 PLC 的四种运行方式，即 Run/Outputs Enabled, Run/Outputs Disabled, Stop/IOScan 和 Stop/No IO Scan。	2-70
6	系统上电/掉电时序	介绍系统上电过程的三个组成部分，包括上电自检，PLC 初始化和系统配置；同时介绍掉电时序和内存数据的保持。	2-72
7	时钟和计时器	介绍计时钟，日历钟和看门狗定时器。	2-76
8	系统安全性	介绍编程者的保护级请求，包括密码配置和程序块锁，OEM 保护和密码配置，写保护开关。	2-78
9	系列 9070 PLC 的 I/O 系统	介绍了 I/O 数据映射和诊断数据。	2-81

第一节 PLC 扫描概述

逻辑程序在系列 90-70 PLC 中将以重复方式执行，除非被编程器或来自于其他设备或 CPU 模板上的“运行/停止”强制开关的命令中止。另外 PLC 在执行用户程序的过程中，PLC 从输入设备获取数据，发送数据到输出设备，运行内务处理程序，服务于编程器，服务于其他通讯，运行自检。执行一系列指令一次所必须的时间称为一个扫描。本节将概要介绍扫描的各个阶段；若需了解详细信息，参见本章第四节。

基本 PLC 扫描

一个典型的扫描周期主要包括七个主要步骤，如下图所示：

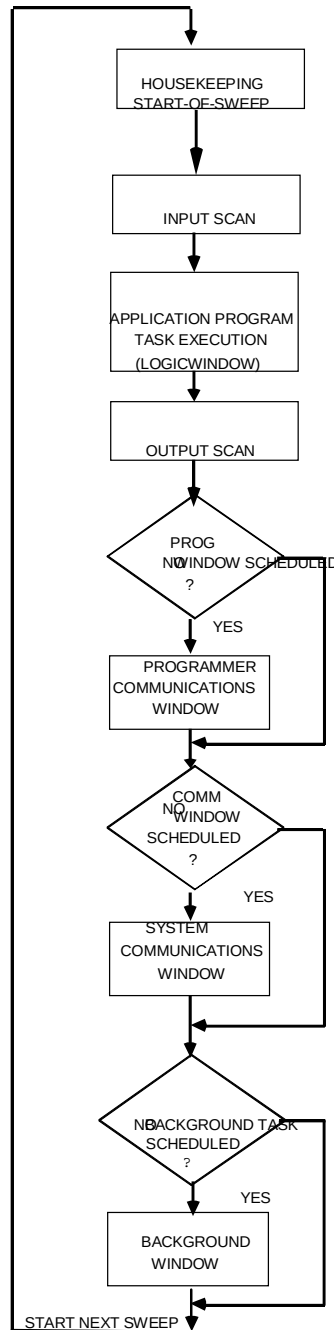


图 2-1. 典型的PLC扫描步骤

表 2-1. 一般扫描的主要步骤

步骤	说明
内务处理	更新系统寄存器，确定计时器值，确定在这一阶段的扫描模式。
输入扫描	CPU从总线控制器和输入模板读取输入数据。
应用程序执行 (逻辑窗口)	CPU处理逻辑程序，使用从输入设备获得的数据，送相应的输出至输出设备。.
输出扫描	在这个阶段，CPU写输出数据至总线控制器或输出模板。用户程序在此时进行总检（除多周期扫描方式），故障板的查询也在这个阶段进行。
编程器通讯窗口	使用串口或WSI设备与编程器进行数据或状态信息交互通讯。另外，重新配置模板或机架也在这个阶段进行。
系统通讯窗口	同所有智能设备进行通讯 (除了串口或 WSI编程器进行连接时的通讯)。例如，提供数据至 PCM* 以驱动进程的显示。以太网编程器的通讯也在系统通讯窗口中进行。
后台任务窗口	进行CPU 自检。

*For information about the PCM, refer to the *Series 90™ Programmable Coprocessor Module and Software Support* (GFK-0255).
**In Microcycle sweep mode the user program checksum is computed during the Input Scan.

内务处理

内务处理完成开始扫描前的准备工作，包括刷新所有%S标志位，确定定时器更新值，确定PLC的扫描模式（STOP或RUN）。

输入扫描

在逻辑处理之前作输入扫描，在此期间，CPU从总线控制器上读取Genius、FIP总线输入数据和系列9070输入模板数据。同时，EGD通讯数据的数据接收者从以太网板读取数据至PLC内存中。有关详细信息，参见《系列90 PLC TCP/IP以太网通讯使用手册（GFK-1541）》。
在扫描时进行FIP总线读取，仅周期性的VCOM (MPS)服务受影响。在系统通讯窗口会收到这些信息。

系列90-70 I/O模板是从I/O参考地址的最低端到最高端进行扫描的。对总线控制器没有确定的扫描顺序。

说明

如果程序在上一个扫描周期设置了主动中断I/O的功能，则输入扫描将不在执行。

应用程序执行(逻辑窗口)

逻辑窗口是扫描周期中执行用户程序的阶段。输入扫描一完成，PLC确定执行哪个用户程序。随后必要的程序被恢复或调用。通过解析逻辑程序来获得新的输出信息。

中断程序和中断程序块可在扫描的任何阶段被执行。详细信息参见第四节相关内容。

有多种控制程序执行的方法来满足系统定时的需要，这些方法包括：

- ☐ JUMP功能用于跳过逻辑程序段。
- ☐ 中止I/O功能可以用来中止一个扫描周期的输入和输出扫描。必要时，可在逻辑执行时使用DO I/O指令来刷新I/O。
- ☐ 服务请求功能可用于暂停或改变分配给扫描窗口的部分时间。
- ☐ 程序逻辑可设计为多次调用程序块的结构，调用次数取决于它们的重要程度和定时约束。
- ☐ 多周期扫描模式可用于各部分程序要求执行频度不同的情况下来限制各逻辑窗口的执行时间。

各指令的执行时间由附录A表中给出。

说明

多周期扫描模式下，在PLC执行过程中逻辑窗口可被跳过或优先占用。

输出扫描

逻辑处理完成之后立即进行输出扫描。此时，CPU向Genius和FIP总线控制器以及90-70输出模板发送输出数据。PLC EGD配置中的数据发送者将通过以太网设备定时的产生新的采样数据。（详细内容参见《系列90 PLC TCP/IP以太网通讯使用手册（GFK-1541）》）。系列90-70输出模板按照I/O参考地址从低到高依次扫描。总线控制器则按照从0到7机架的顺序，在各机架内部按低位槽到高位槽的顺序进行扫描。

说明

如果程序中当前扫描周期有一个有效中止I/O功能块，则下扫描不进行输出扫描。

在扫描时进行FIP总线读取，仅周期性的VCOM (MPS)服务受影响。在系统通讯窗口会收到这些信息。

故障板查询也在输出扫描时进行。故障板查询提示产生错误需要置换的板子并在系统中对其进行重新配置。如果一块原先在系统中的或被用户配置在系统中的板发生故障，它必须周期性地记录以确认是否已被一块新板所替代。

一旦原来的一块故障板而现在检测已无故障，则需在编程器通讯窗口重新进行配置直到该板被重新配置到系统中为止。

后台检验计算也在输出扫描时进行。检验计算包括了用户程序配置的总字数，这可以确保CPU运行时用户逻辑的完整性。如果CPU被配置成执行一个后台检验和计算（默认为16）则执行这部分的扫描，否则它将被忽略。

还有其他测试在输出扫描阶段进行，包括：处理器检测—检测微处理器基本的操作，和BCP运行代码检测—检测所有BCP指令的基本操作。

说明

从7.0版的CPU开始，仅在多周期扫描模式下，后台检测计算在输入扫描阶段进行。

编程器通讯窗口

编程器通讯窗口用来实现通过串口或并口与编程器之间的连接和执行故障板重新配置。在同C语言测试工具的通讯也要用到它。如果有编程器连接，测试任务将被激活，或者如果在系统中有模板需要重新配置（在扫描的故障板查询阶段检测到的）的两种情况，CPU都将执行编程器窗口；否则此窗口不执行。在编程器窗口中，对模板进行配置的优先级最高。板子按其需要进行配置，直到为编程器窗口分配的时间到，才结束此过程。

CPU通过编程器窗口来完成与并口（两个专用编程器端口）或者是与内置的SNP接口之间的通讯。对于CPX型的CPU板，均配备了3个内置的SNP接口。CPU首先完成原先未完成的请求，接着完成在队列中等待处理的请求，最后处理新的请求。分配给编程器窗口的时间终止时该过程结束。

编程器窗口的时间默认为10ms，这个值可以通过编程软件进行配置和保存或在线更改。

编程器窗口的时间及执行可通过使用服务请求功能#3动态控制。编程器通讯窗口的时间值范围为0到255ms。需要说明的是，如果编程器窗口的时间被设为0，有3种方式可以打开该窗口：内存掉电重新上电，改变PLC状态至停止方式，或使用以下说明中的编程并口。

说明

即使编程器窗口被跳过，PLC仍可通过指令改变其模式或状态，或者通过BTM上的并口与编程器连接并重新定义编程窗口，或者手动将PLC转入停止方式。

系统通讯窗口

此窗口是用于CPU与其他智能模块如PCM，Genius总线控制器，FIP总线控制器和TCP/IP以太网板等设备之间的通讯。以太网的编程器通讯也在系统通讯窗口中进行。
在此窗口开始时，CPU在执行队列中等待的请求之前，要首先完成原先未完成的请求。当为此窗口分配的时间结束，此过程也就此停止。

系统通讯窗口默认模式为“Run to Completion”，这意味着每次扫描中所有智能模块上的当前等待的请求都要被处理。使用编程软件可对此窗口的限定时间进行配置和保存或者是在线修改。

系统通讯窗口的时间及执行可通过使用服务请求功能#4动态控制。在某个临界时间的扫描内，运行跳过该窗口的执行。系统通讯窗口的时间值范围为0到255ms。

后台窗口

在此窗口进行CPU自检。自检包括对90-70 CPU操作系统软件的校验。
后台窗口的时间默认为0。该值可以被配置和保存至PLC或者是使用编程软件进行在线的更改。

窗口的时间及执行可通过使用服务请求功能#5动态控制。在某个临界时间的扫描内，运行跳过该窗口的执行。

Window Modes

在前面的章节里对一个典型的扫描进行了说明。编程器窗口，系统通讯窗口和后台窗口这些PLC扫描的阶段有基于PLC扫描方式的多种运行方式。（PLC扫描方式在本章第四节中进行说明。）以下是PLC的窗口的三种运行方式：

Run-to-Completion	在 Run-to-Completion 方式下,当窗口开始时刻所产生的所有请求都要被处理。当所有在给定窗口中等待的请求完成后, PLC才转入下一阶段的扫描。
Constant	在恒定窗口方式下, 编程器窗口, 系统通讯窗口和后台窗口的总扫描时间是固定的。当一个请求正在进行当中时, 如果时间结束, 则这些窗口关闭, 通讯将在下一个扫描周期继续。如果在规定的窗口时间内不再有等待的请求, 则PLC将在剩余时间循环执行这些窗口并查询进一步的请求。如果有任何一个窗口被配置为恒定窗口方式, 则所有的窗口都将按恒定窗口的方式执行。
Limited	在限制模式下, 窗口运行的最大时间是固定的。当一个请求正在进行当中时, 如果时间结束, 则这些窗口关闭, 通讯将在下次给定窗口运行时继续。如果当前窗口没有等待的请求, 则PLC将转入下一阶段的扫描。

在通讯窗口中的数据相关性

当运行在恒定时间窗口或有限时间窗口模式时, 编程器和系统通讯窗口可能被中止。如果一个外部设备, 例如GBC (Genius总线控制器), 正在传送数据包时, 在完成该请求前被中止, 则数据包的相关就有可能被打乱。该请求将在下一个扫描中继续完成, 然而, 部分数据将在一个扫描中输出结果, 而剩下的数据将在下一个扫描中继续。当PLC在一般扫描方式, 通讯窗口在Run-to-Completion方式时, 则不存在上面所述的数据相关性的问题。

说明

当与PLC通讯的外部设备停止时, 将按照最后的状态读信息。这可能误导操作者, 他们通过HMI系统看到的并不代表PLC的RUN/STOP状态。过程图表将总是显示当前仍运行正常。

同样需要注意, 不保持型的输出也是只有当PLC CPU从STOP转入RUN方式时才清除数据。

停止模式下的CPU扫描

90-70 PLC 在停止模式时有两种方式： Stop/NoIO 和 Stop/IOScan。

当 PLC在 Stop/NoIO 方式时， 输入扫描， 逻辑窗口和输出扫描在PLC扫描时被跳过。

当 PLC在Stop/IOScan 方式时， 逻辑窗口在PLC扫描时被跳过。输入扫描和输出扫描仍在每个扫描中进行。

在Stop/NoIO和Stop/IOScan两种方式下， 两个通讯窗口都按照Run-to- Completion方式运行， 后台窗口按照10ms的限制时间方式进行。

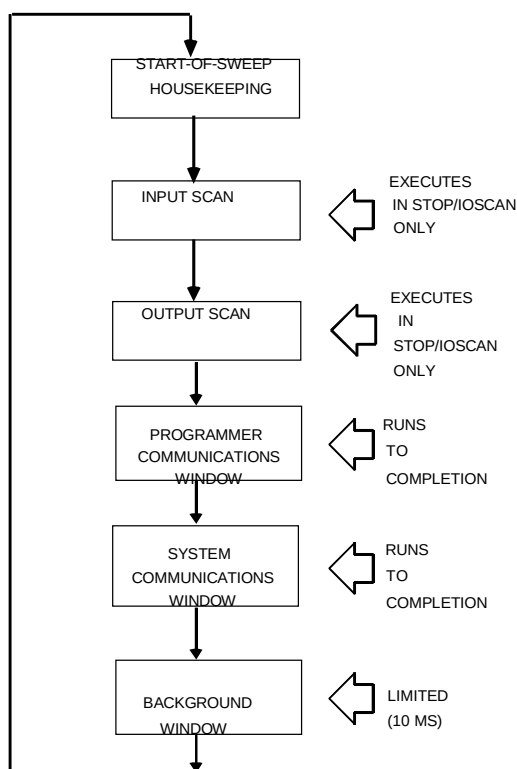


图 2-2. Stop/NoIO 和Stop/IOScan 方式下的CPU扫描

说明

多周期扫描模式下，不支持Stop/IOScan 方式。

PLC 扫描模式

90-70 PLC 共支持四种扫描模式：

一般扫描模式	在一般扫描模式下，PLC扫描时间是可变的。逻辑窗口在每个扫描中都被完全执行。通讯窗口和后台窗口可以被配置为限制时间方式或 Run-to-Completion 方式。
恒定时间扫描模式	在恒定时间扫描模式下，PLC按照用户定义的扫描时间进行扫描。逻辑窗口在每个扫描中都被完全执行。如果有足够的时间结束所有窗口的扫描，则PLC将交替于通讯窗口和后台窗口，知道下一个扫描周期开始。
恒定窗口扫描模式	在恒定窗口扫描模式下，PLC扫描时间是可变的。逻辑窗口在每个扫描中都被完全执行。PLC将交替执行通讯窗口和后台窗口，直到用户定义的恒定窗口时间结束。
多循环扫描模式	同恒定时间扫描方式相同，在多循环扫描方式下，每个扫描周期按照固定的时间进行。总的扫描时间和总的通讯和后台窗口时间由用户定义。为了保持总的扫描时间和通讯窗口、后台窗口的时间一定，逻辑窗口被优先执行。为了满足定义的窗口时间，PLC交替执行编程器通讯窗口，系统通讯窗口和后台窗口，直到下一个扫描开始。

说明

以上介绍了PLC四种不同的扫描模式，有关PLC扫描模式的详细内容见本章第四节“PLC扫描模式和程序时序方式”。

第二节: 用户寄存器数据

用户寄存器

应用程序中使用的数据以离散变量或寄存器变量的形式存储于PLC。

表 2-2. 寄存器变量

类型	
%R	前缀 %R 表示系统寄存器变量，用来存储计算结果等程序数据。
%AI	前缀 %AI 表示模拟量输入寄存器。其后面跟的是寄存器地址（如%AI0015），模拟量输入寄存器用来存储模拟量输入的值或其他非离散量值。
%AQ	前缀表示模拟量输出寄存器。其后面跟的是寄存器地址（如%AQ0056），模拟量输出寄存器用来存储模拟量输出的值或其他非离散量值。
%P*	前缀 %P表示该种寄存器是分配给程序寄存器的，它存储主程序块中的程序数据，这些数据可供所有程序块存取，%P数据块的大小由所有程序块中最大使用地址决定。（有关更多信息请参考附录D“寄存器分配”部分内容）。%P通常不能被外部主机访问。
%L*	前缀%L表示该种寄存器是分配给局部寄存器的，它存储对应于特定程序块中的程序数据。%L数据块的大小由所有程序块中最大使用地址决定。（有关更多信息请参考附录D“寄存器分配”部分内容）。%L仅用于定义它的程序块内。

* These reference types are scoped at a program level and are therefore only visible to LD programs.

说明

所有寄存器都通过CPU的电源回路来保持数据。

间接变量访问

间接变量访问适用于所有寄存器变量 (%R, %AI, %AQ, %P, %L)。它可通过包含了偏移量的同类型数据来识别寄存器内存单元。间接变量的输入方法与直接变量相同，只是需要在%前使用@，例如，若%R00101内存储的值是1000，则@R00101将指令PLC定位使用%R01000的存储数据。

在使用多个寄存器进行相同的逻辑或运算时使用间接变量访问是相当方便的。使用间接变量也可避免在一个应用程序中重复使用相同的梯形图逻辑。它可用在循环状态下，即寄存器累加常数或某一值，直到循环结束。

表 2-3. 离散变量

类型	说明	状态保持
%I	前缀%I 表示开关量输入，其后跟变量地址（例如，%I00121）。%I定位于输入状态表中，用于存储上次输入扫描中由输入模块获得的所有输入状态。用编程软件来对输入模板进行输入变量地址分配。只有在分配好变量地址后，才可从输入模板获取数据。%I总是保持型的。	保持型
%Q	前缀%Q表示开关量输出，线圈检查功能可用于对重复使用%Q继电器输出或功能块输出的检查。从Logicmaster第4.0版开始，允许用户选择线圈检查的等级（如仅允许单个输出，多个输出警告和允许多处输出）。 其后跟变量地址（例如，%Q0016）。%Q定位于输出状态表中，用于存储上次扫描中由应用程序获得的所有输出状态。在程序扫描的最后阶段把输出状态表的值送输出模板。用编程软件来对输出模板进行输出变量地址分配。只有在分配好变量地址后，输出数据才可送输出模板。%Q可以是保持型的，也可以是非保持型的*。	决定于所使用的线圈类型
%M	前缀%M 表示中间变量。线圈检查功能可用于对重复使用%Q继电器输出或功能块输出的检查。%Q可以是保持型的，也可以是非保持型的*。	决定于所使用的线圈类型
%T	前缀 %T表示临时变量。这类变量不进行线圈的重复使用检查因此可以在同一程序中多次作为线圈输出使用。然而并不推荐这样使用，因为那样会增加查错工作的难度。%T可用来避免在使用剪切/粘贴功能时出现的线圈使用冲突的报错。由于这类变量都是临时使用的，故其存储的数据掉电或在Run-to-Stop-to-Run切换中不保持，也不能用于保持型的线圈。	不保持型

表 2-3. 离散变量—续

类型	说明	状态保持
%S %SA %SB %SC	前缀%S, %SA, %SB和 %SC 表示系统状态变量。用来存取专用的PLC数据, 如计时器, 扫描信息, 故障信息等。举例来说, %S C0012 可用来检查PLC故障表的状态, 一旦该位被一个故障置位, 直到该故障消除才会被复位。 ☐ %S, %SA, %SB, %SC 可用于任何接点。 ☐ %SA, %SB, %SC 可用于保持型线圈。—(M)— 说明 虽然编程软件逻辑在使用%SA, %SB和 %SC变量时使用保持型线圈, 但大部分在系统掉电后并不能保持。 ☐ %S可在功能或功能块中用作字或位串输入。 ☐ %SA, %SB和 %SC可在功能或功能块中用作字或位串输入或输出。	保持, 但在系统上电时要初始化。 (详见表2-9中每个变量的使用规范)
%G %GA %GB %GC %GD %GE	前缀 %G, %GA, %GB, %GC, %GD和 %GE 表示全局数据变量。这些变量用于几个PLC之间的共享数据。%G, %GA, %GB, %GC, %GD和 %GE能用于接点和保持型线圈, 不能用于非保持的情况。	保持

* Retentiveness is based on the type of coil. For more information, refer to “Retentiveness of Logic and Data” later in this section.

用户变量的范围及默认值

下表列出了各种型号CPU的用户变量最大使用范围及其默认的配置值。

表 2-4. 用户变量的范围

项目	CPU 型号					
	935/928 925/915/790	924/914	788	780/781 782/789	771/772	731/732
最大 %I地址	12288 点	12288 点	352 点 ¹	12288 点 ²	2048 点 ²	512 点 ²
最大 %Q地址	12288 点	12288 点	352 点 ¹	12288 点 ²	2048 点 ²	512 点 ²
最大%M地址	12288 点	12288 点	12288 点	12288 点	4096 点	2048 点
最大%T地址	256 点	256 点	256 点	256 点	256 点	256 点
%S (S, SA, SB, SC)	512 点	512 点	512 点	512 点	512 点	512 点
%G (GA, GB, GC, GD, GE)	7680 点	7680 点	7680 点	7680 点	7680 点	1280 点
用户 RAM 1024K 字节	(6MB对于928)	512K 字节	512K 字节	512K 字节 (CPX 782 为 1024 KB)	CPU771/2: 64, 128, 256, 512 KB 取决于订购的 控制内存板。 CPX772: 512 KB	32K 字节
最大 %AI地址	8K 字	8K 字	8K 字	8K 字	8K 字	8K 字
最大 %AQ地址	8K 字	8K 字	8K 字	8K 字	8K 字	8K 字
最大 %R, 1K 增加字	16K 字	16K 字	16K 字	16K 字	16K 字	16K 字
最大%L (每程序块)	8K 字	8K 字	8K 字	8K 字	8K 字	8K 字
最大%P	8K 字	8K 字	8K 字	8K 字	8K 字	8K 字

1 物理输入和输出点数总计不能超过352点。这相当于约100个冗余点。详细信息见GFK-1277第一章。

2 6版之前的Logicmaster编程软件限制总的%I 和 %Q的数量限制见下表。例如，当使用以前版本的编程软件来对782 CPU进行编程的话，总的%I 和 %Q的数量最大为12288点。这一限制对新版的Logicmaster不再适用。

表 2-5. 默认内存大小

内存 类型	CPU 型号					
	935/928 925/915/790	924/914	780/781/782 788/789	781/782	771/772	731/732
%AI	64 字	64 字	64 字	64 字	64 字	64 字
%AQ	64 字	64 字	64 字	64 字	64 字	64 字
%R	1024 字	1024 字	1024 字	1024 字	1024 字	1024 字
%P	0 字	0 字	0 字	0 字	0 字	0 字
%L	0 字	0 字	0 字	0 字	0 字	0 字

%G 用户变量和CPU内存定位

系列 90-70 CPU 为所有的全局数据(%G,%GA, %GB, %GC, %GD, %GE)变量设置了一个数据区。在CPU内存中这些数据长度为7680位。编程软件通过使用前缀%G,%GA, %GB, %GC, %GD, %GE为用户提供了更细的种类划分，每种前缀的使用范围为1-1280。编程软件编译被请求的全局变量的类型(%G,%GA, %GB, %GC, %GD, %GE)在映射到相应的%G内存中去。实际的映射关系如下表所示。

表 2-6. %G用户变量和CPU内存定位

全局数据类型	%G	%GA	%GB	%GC	%GD	%GE
编程软件使用变量地址	%G1-1280	%GA1-1280	%GB1-1280	%GC1-1280	%GD1-1280	%GE1-1280
CPU分配内存空间	%G1-1280	%G1281-2560	%G2561-3840	%G3841-5120	%G5121-6400	%G6401-7680

当使用C程序工具包来编写90-70 CPU的应用程序时，可能会用到以上的信息。关于使用C工具包的详细内容，见《系列90™ PLC C程序工具包使用手册(GFK-0646)》。

说明

731型CPU 仅支持 %G ， 它只有 1280 点的全局变量数据区。

Genius 全局数据

系列90-70 PLC支持连接到一个Genius I/O总线的多PLC系统的共享数据。这一设置为自动并重复传输%G, %I, %Q, %AI, %AQ和%R提供了一种手段。由于它已经集成到I/O扫描中，在使用全局数据时不再需要为其提供特殊的应用程序。所有GE Fanuc的PLC都支持Genius I/O，可直接与系列90-70 PLC进行通讯。

状态转换和强制

%I, %Q, %M和%G类型的用户变量可以有状态转换和强制的状态。 %T, %S, %SA, %SB和%SC类型变量只有状态转换状态。CPU使用状态转换位来控制计数器，切换接点和线圈的状态。计数器不使用类似接点和线圈的状态转换位。计数器的状态转换位是在本地变量中存储。

小心

不要强制状态改变的线圈。如果一个线圈被强制，然后强制被取消，线圈将继续至下一扫描周期。这可能会对PLC梯形图逻辑，或对PLC相关的现场设备产生不可预料的影响。

当强制位置位时，相关变量的状态不再受程序或输入设备的指令而改变。它们只能通过编程者的命令而改变。

逻辑和数据的保持

当PLC由停止切换到运行状态时，若数据被PLC保存，则称此数据具有保持性。由停止转入运行方式时，90-70 PLC保存程序逻辑，故障表和诊断信息，检测程序逻辑，强制量，字数据(%R, %L, %P, %AI, %AQ)和位数据(%I, %G,故障位和保存位)，%Q和%M数据(不包括非保持线圈)。%T数据不具保持性。

除了故障状态位和大部分的%S, %SA, %SB, 和 %SC变量在系统上电时初始化为零，其他保持型数据在PLC掉电时，由后备电池保持其状态。（更多信息见表2-9每个系统状态变量特性）。

当%Q和%M用于非保持型线圈时，它们是非保持型的。（即当PLC在运行状态重新上电或从停止切换到运行方式时，其值会清除。非保持型线圈包括：常开线圈-(Q)-，常闭线圈 -(I)-，置位线圈 -(S)-和复位线圈 -(R)-。

当%Q和%M用于保持型线圈或用作功能块的输出时，数据在系统掉电和在停止/运行切换中是保持的。保持型线圈包括：常开保持线圈-(M)-，常开保持线圈-(/M)-，置位保持线圈-(SM)-，复位保持线圈 -(RM)-。

程序中最后一次使用%Q 或 %M的线圈指令决定了该%Q 或 %M变量是保持型的还是非保持型的。例如，如果%Q00001最后一次作为保持型线圈使用，则%Q00001数据就是保持型的。反之，则%Q00001数据就是非保持型的。

注意

当使用独立的C程序块时，由于C中不存在线圈的指令，数据的保持性仅取决于使用的内存变量的类型。这种情况下 %Q 和 %M变量类型是保持型的。

数据作用域

每种用户变量都有相应的“作用域”；也就是说，它可能对整个系统通用，也可能对所有程序可用或仅在该程序块中可用。

表 2-7. 用户变量的数据作用域

用户变量	作用域	范围
%I, %Q, %M, %T, %S, %SA, %SB, %SC, %G, %R, %AI, %AQ, 设备变量, 故障定位变量	系统	所有的程序块和主机中
%P	程序	所有程序块中 (仅对应一个主机)
%L	局部	一个程序块中 (仅对应一个主机)

在梯形图程序块中：

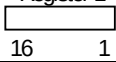
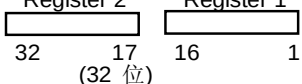
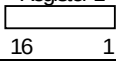
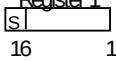
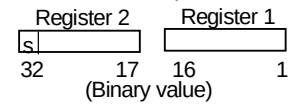
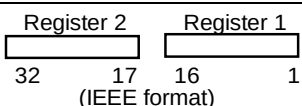
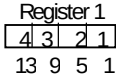
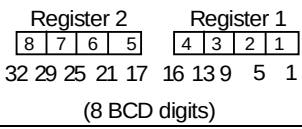
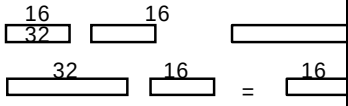
- ☐ %P 可用于与其他程序块共享的程序变量。
- ☐ %L 为局部变量，仅限用于该程序块的局部变量，这些局部变量在程序的其他地方是无效的。

%I, %Q, %M, %T, %S, %SA, %SB, %SC, %G, %R, %AI, and %AQ 可用于整个系统。

附录 D 中的内存分配表用来确定用户数据已使用的总字节数，以及还有多少可供用户逻辑使用。

数据类型

表 2-8. 数据类型

类型	名称	说明	数据格式
BOOL	布尔型	布尔型数据为最小的内存单元，它有1或0两种状态。一个布尔数列可有N个长度。	
BYTE	字节型	一个字节有8位，最大为255 (0–255)，一个字节数列的长度可以为N。	
WORD	字型	一个字型数据包括连续16位内存数据。字的有效范围为0000 hex 到 FFFF hex。	Register 1  (16 位寄存器)
DWORD	双字型	一个双字于单个字具有相同的特定，只不过它使用连续32位内存地址而不是16位。	Register 2 Register 1  (32 位)
UINT	无符号整型	无符号整型数使用16位内存数据地址。它的有效范围为0 到 +65535 (FFFF hex)。	Register 1  (Binary value)
INT	整型	带符号整型数使用16位内存数据地址并由2的补码来代表。它的有效范围为–32768到+32767。	Register 1 (Two's Complement value) 
DINT	双精度整型	双精度整型存储于32位内存地址（两个连续的16位内存地址）并总是带符号值（32位为符号位）。它的有效值范围为–2147483648 到 +2147483647。	Register 2 Register 1  (Binary value)
REAL	浮点型	实型数使用32连续位（两个连续的16位内存地址）并总是带符号值（32位为符号位）。存储于其中的数值范围为± 1.401298E–45 到 ± 3.402823E+38 对于IEEE格式，见附录D“使用浮点运算”。	Register 2 Register 1  (IEEE format)
BCD-4	4位二进制编码数	四位BCD数使用16位内存地址。每个BCD数字使用4位来表示0到9之间的数字9。此16位的BCD 代码的有效范围为 0 到 9999。	Register 1  (4 BCD digits)
BCD-8	8位二进制编码数	八位 BCD数据类型使用两个连续16位数据地址（32连续位）。每个BCD 数字使用4位来表示0到9之间的数字9。此32位的BCD 代码的有效范围为 0 到 99999999。	Register 2 Register 1  (8 BCD digits)
MIXED	混合型	混合型数据类型仅用于MUL和 DIV功能块。MUL功能块使用两个整型输入，得到一个双整型的数据结果。DIV功能块使用一个双整型数去除一个整型数，得到一个整型数的结果。	
ASCII	ASCII	八位编码字符，一个变量由两个打包到一起的两个ASCII 字符组成。第一个字符对应该变量字的低字节。该字节第一位用于奇偶校验，另七位由转换方法转换。命令代码以及非显示字符在屏幕上以非文字字符显示。（例如，@）。	

S = Sign bit (0 = positive, 1 = negative).

注意

使用非明确为位类型操作的功能块的状态转换选择，将会影响到所写入的整个字节、字或双字的所有位。

关于使用浮点数的相关内容，见附录D“使用浮点数”部分。

系统状态变量

系列90-70 PLC的系统状态变量（原来称为“便利”变量）包括%S, %SA, %SB, %SC的变量内存。每一个都有系统提供的助记符别名，方便编程者使用。时间接点变量包括T_10MS, T_100MS, T_SEC和 T_MIN等；其他系统状态变量包括FST_SCN, ALW_ON和 ALW_OFF等。

注意

%S 型状态位为只读，编程者不可写入数据。然而%SA、%SB和 %SC型状态位是允许写入数据的。

下面列出的系统状态变量可直接在应用程序中使用。在输入逻辑量时，可使用变量地址或变量别名。如需更详细的故障说明与修正信息，请参阅第三章“故障的说明与排除”。在其他文本中使用这些变量的别名是受限的，如它们不能用于程序块名和文件夹名。

注意

很多不适用于系列90-70 PLC的变量未在下表中列出，如%S0002。带有Genius模板冗余的产品(CPU788, CPU789和 CPU790)有附加的系统状态变量，如780CPU和CGR935（带CPU冗余）。相关Genius模板冗余的产品的状态变量见GFK-1277，CPU冗余产品的见GFK-1527。

表 2-9. 系统状态变量

变量地址	名称	说明
%S0001	FST_SCN	当前扫描为第一次扫描或从停止转换到运行时置位。 注意: 在独立的C程序块中, 使用C语言的相关宏FST_SCN, 而不是%S0001。
%S0003	T_10MS	0.01秒时间接点。.
%S0004	T_100MS	0.1秒时间接点。.
%S0005	T_SEC	1.0 s秒时间接点。.
%S0006	T_MIN	1.0分时间接点。.
%S0007	ALW_ON	常开接点。.
%S0008	ALW_OFF	常闭接点。.
%S0009	SY_FULL	当PLC故障表满(默认为16条, 可单独配置)时置位, 故障清除后复位。
%S0010	IO_FULL	当PLC故障表满(默认为32条, 可单独配置)时置位, 故障清除后复位。f
%S0011	OVR_PRE	当 %I, %Q, %M, or %G内存中存在强制量时置位。.
%S0012	FRC_PRE	当一个 Genius 点被强制时置位。
%S0013	PRG_CHK	当后台程序检查激活时置位。
%S0014	PLC_BAT	在第4版或更新版CPU检测到损坏的电池时置位, 当电池状态发生变化, 该接点的状态随之更新。
%S0121	FST_EXE	当前扫描是该程序块首次调用。当从STOP转入RUN状态时该变量置位。
%SA0001	PB_SUM	当应用程序中计算的和检验于变量的和检验不匹配时置位, 若错误来自一临时故障, 该位将通过再一次CPU存储将之清零, 如果错误来自RAM故障, 则需要更换CPU。 清除此故障位, 需要清除PLC故障表或将CPU重新上电。
%SA0002	OV_SWP	当PLC检测到上次扫描比用户规定的时间长时, 此变量置位, 相反, 若短时复位。清除此故障位, 需要清除PLC故障表或将CPU重新上电。只有当PLC处于恒定时间或微循环扫描模式时有效。
%SA0003	APL_FLT	当发生应用程序故障时置位, 清除此故障位, 需要清除PLC故障表或将CPU重新上电。
%SA0009	CFG_MM	当配置不匹配故障报于故障表时置位, 清除此故障位, 需要清除PLC故障表或将CPU重新上电。
%SA0010	HRD_CPU	当诊断程序检测到CPU硬件故障时置位。清除此故障位, 需要清除PLC故障表或将CPU重新上电。
%SA0011	LOW_BAT	当发生电池故障时置位, 清除此故障位, 需要清除PLC故障表或将CPU重新上电。
%SA0012	LOS_RCK	当扩展机架与PLC CPU通讯中断时置位, 清除此故障位, 需要清除PLC故障表或将CPU重新上电。

注意

%SA, %SB和%SC接点在故障或清除故障表之后的扫描周期中的输入扫描阶段才会被置位或复位。

%SA, %SB和 %SC接点也可以由用户逻辑或PLC监控设备来置位和复位。

请注意若无适当的调节触故障信息，则在PLC重新上电时才会返回故障信息。

表 2-9. 系统状态变量 - 续

变量	名称	定义
%SA0013	LOS_IOC	当总线控制器与PLC的通讯中断时置位，清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0014	LOS_IOM	当I/O模板与PLC的通讯中断时置位，清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0015	LOS_SIO	当可选智能模板与PLC的通讯中断时置位，清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SA0017	ADD_RCK	当扩展机架添加到系统中时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SA0018	ADD_IOC	当某总线控制器添加到系统中时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SA0019	ADD_IOM	当某I/O模板添加到系统中时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0020	ADD_SIO	当某可选智能模板添加到系统中时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0022	IOC_FLT	当总线控制器报总线故障，全局内存故障，和IOC硬件故障时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0023	IOM_FLT	当I/O模板报告一个回路或模板故障时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0027	HRD_SIO	当一个可选模块检测到硬件故障时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0029	SFT_IOC	当在I/O控制器中出现软件故障时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0031	SFT_SIO	当可选模块检测到内部软件错误时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0032	SBUS_ER	当在VME背板总线出现总线错误时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SA0081 – %SA0112		当用户自定义故障信息出现在PLC故障表中时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。更多内容参见第四章关于服务请求21的相关讨论。
%SB0001	WIND_ER	在恒定时间扫描，或微循环扫描模式下没有足够的时间开始编程器窗口或当在微循环模式下没有足够时间来开始逻辑窗口时，此状态变量置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SB0009	NO_PROG	当PLC CPU在内存保护启动，而无用户程序时置位。当PLC有程序上电或清除PLC故障表时，此故障位复位。

表 2-9. 系统状态变量 - 续

变量地址	名称	说明
%SB0010	BAD_RAM	当CPU上电检测到RAM内存冲突时置位。当CPU检测到RAM内存有效或清除PLC故障表时复位。
%SB0011	BAD_PWD	当密码输入有误时置位，清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SB0012	NUL_CFG	当无配置数据情况下试图将PLC转入运行状态时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SB0013	SFT_CPU	当CPU检测到操作系统软件错误时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SB0014	STOR_ER	在编程器存储操作期间发生错误时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SB0016	MAX_IOC	当系统中配置了超过32个IOC时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SB0017	SBUS_FL	当PLC不能访问总线时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SC0009	ANY_FLT	当出现任何登记于PLC或I/O故障表的故障发生时置位。清除此故障位，需要清除PLC和I/O故障表或将CPU重新上电。
%SC0010	SY_FLT	当出现任何登记于PLC故障表的故障发生时置位。清除此故障位，需要清除PLC故障表或将CPU重新上电。
%SC0011	IO_FLT	当出现任何登记于I/O故障表的故障发生时置位。清除此故障位，需要清除I/O故障表或将CPU重新上电。
%SC0012	SY_PRES	当PLC故障表中存在至少一条故障时置位。清除PLC故障表时复位。
%SC0013	IO_PRES	当I/O故障表中存在至少一条故障时置位。清除I/O故障表时复位。
%SC0014	HRD_FLT	当有硬件故障时置位。清除PLC和I/O故障表或PLC重新上电时复位。
%SC0015	SFT_FLT	当有软件故障时置位。清除PLC和I/O故障表或PLC重新上电时复位。

其他变量

本手册第三章对故障变量进行了详细介绍，这里列出以便阅读者查找。

表 2-10. 系统故障变量

系统故障变量	说明
ANY_FLT	系统中的任何一种故障
SY_FLT	PLC故障表中任何一种新产生的故障
IO_FLT	PLC故障表中任何一种新产生的故障
SY_PRES	表示在PLC故障表中至少有一故障
IO_PRES	表示在I/O故障表中至少有一故障
HRD_FLT	任何硬件故障
SFT_FLT	任何软件故障

表 2-11. 可配置故障变量

可配置故障 (默认故障作用)	说明
SBUS_ER (诊断)	系统总线错误。(VME系统总线产生BSERR信号)。
SFT_IOC (诊断)*	Genius总线控制器出现不可恢复软件错误。
LOS_RCK (诊断)	机架未接通或丢失已配置的机架 (BRM错误, 掉电)
LOS_IOC (诊断)*	总线控制器未接通或丢失已配置的总线控制器。
LOS_IOM (诊断)	I/O模块未接通 (无响应) 或丢失已配置的I/O 模块。
LOS_SIO (诊断)	智能可选模块未接通(无响应) 或丢失已配置的可选智能模块。
IOC_FLT (诊断)	非致命总线或总线控制器错误—10秒内大于10次总线错误。
CFG_MM (致命)	上电期间, 存储配置或运行模式检测到错误的模板类型。PLC不检测单个模板的配置参数, 如Genius I/O块。

表 2-12. 不可配置故障

不可配置故障 (作用)	说明
SBUS_FL (致命)	系统总线故障。PLC CPU不能访问 VME 总线。BUSGRT-NMI 错误。
HRD_CPU (致命)	PLC CPU硬件故障，如内存设备故障或串口故障。
HRD_SIO (诊断)	系统非致命硬件故障，如PCM上的串口故障。
SFT_SIO (诊断)	PCM或LAN接口模板不可恢复软件错误。
PB_SUM (致命)	上电或运行模式时程序或程序块检测失败。
LOW_BAT (诊断)	系统中CPU或其他模板电池电量低信号。
OV_SWP (诊断)	恒定时间扫描超时。
SY_FULL, IO_FULL (诊断)	PLC 故障表满 I/O故障表满
IOM_FLT (诊断)	I/O 模板上的点或通道—模板部分故障。
APL_FLT (诊断)	应用程序故障。
ADD_RCK (诊断)	新机架添加，多余或先前故障机架恢复。
ADD_IOC (诊断)	额外I/O总线控制器或I/O总线控制器重新启动。
ADD_IOM (诊断)	先前报故障的 I/O不再有故障或额外I/O模板。
ADD_SIO (诊断)	新的智能可选模块被添加，多余或重启。
NO_PROG (信息)	上电时无应用程序。应当只发生在PLC第一次上电或带后备电池的RAM保持程序错误时。
BAD_RAM (致命)	上电时程序内存冲突。程序不能读入或不能通过检测。
WIND_ER (信息)	窗口完成错误，通讯窗口服务预先占取以便下次扫描。发生在恒定时间扫描或微循环扫描方式下。
BAD_PWD (信息)	改变优先级的请求被拒绝，口令错误。
NUL_CFG (致命)	转换到运行模式时无配置。不设置任何配置数据的运行类似于将I/O扫描延迟。
SFT_CPU (致命)	CPU 软件故障。CPU检测到不可恢复错误。可能由看门狗定时器超时引起。
MAX_IOC (致命)	超过总线控制器的最大个数。系列90 PLC最多支持32个总线控制器。
STOR_ER (致命)	从编程器向PLC下装数据失败，某些PLC中的数据可能被损坏。

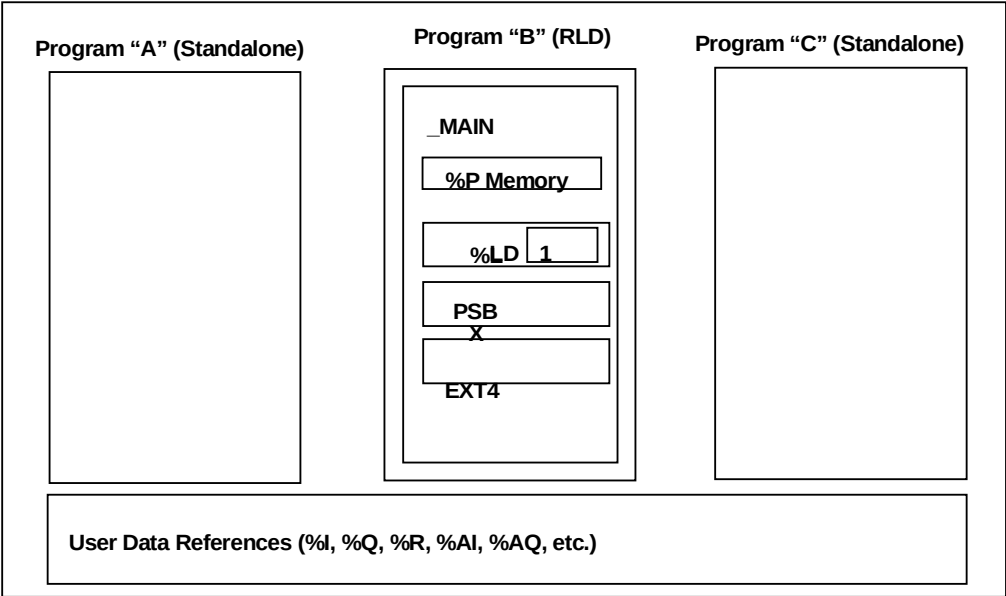
主要

有关故障和FIP定位变量的讨论参加手册《系列90™顺序流程图编程语言使用手册(GFK-0854)》第三章第一节SFC变量部分。

第三节: 程序的组织结构

用户程序包括了用于处理输入数据和控制输出数据的程序逻辑。程序逻辑在PLC中以重复方式执行。系列90-70 PLC最大运行16个用户程序，包含一个梯形图程序。表2-4和2-5列出了每种CPU模板所对于的程序大小和变量地址范围的限制。

下图中列出了3个用户程序，其中两个是独立的C程序。梯形图程序由四个程序块组成(分别是 MAIN, LD_1, PSB_X和 EXT4)。图中更详细的图解显示了各种内存类型的范围：除%P和%L外的所有其他类型变量都对独立C程序可用，%P变量对于所有的程序块可用，而程序块LD_1则有它自己的局部变量%L。对于独立程序和程序块的详细介绍在本节后面部分介绍。



梯形图逻辑程序

系列90-70 PLC的梯形图程序由一个或多个称为程序块的部分组成。90-70 PLC支持的四种程序块的类型如下：

表 2-13. 程序块类型

程序块类型	编程语言	大小限制	参数对个数	主要
LD	梯形图逻辑	16 KB Logicmaster;	n/a	
SFC	梯形图逻辑 /SFC	16 KB Logicmaster;	n/a	SFC 程序块不能用于 中断块。
PSB	梯形图逻辑	16 KB Logicmaster;	0–7	

External	C	64,000 bytes 0–7 External blocks cannot call any other blocks. External blocks are created using the C Programmer's Toolkit.

- SFC 编程 将在《90系列顺序流程图编程语言使用手册GFK-0854》中详细介绍。顺序流程图语言 (SFC)符合 IEC-标准，图形化状态语言尤其适用于顺序控制。
- LD (即继电器逻辑图形语言)是在Logicmaster 和控制软件中使用梯形图编辑窗口的一种语言。
- PSB (参数化子程序块) 是有自己的输入输出参数的梯形图程序块。

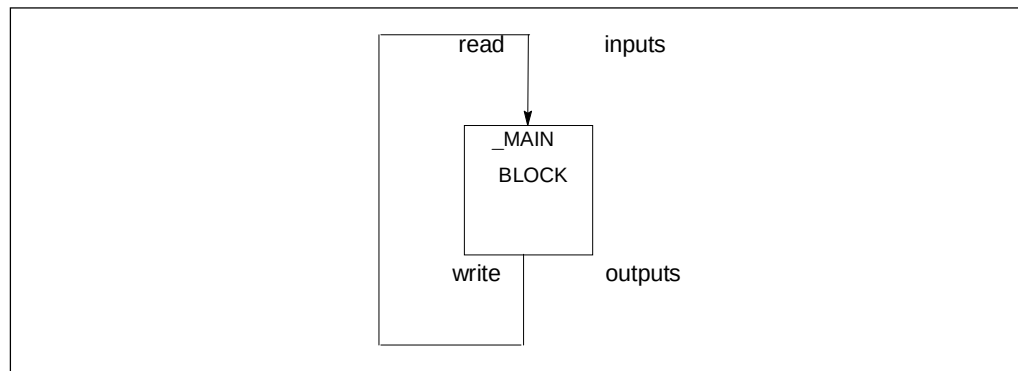
注意

程序中最多允许使用255个程序块。在一个给定程序块中最多调用64个程序块。一个程序块在一个程序中最多被调用255次。（一个程序块可执行无限多次，但在编程中，调用该程序块指令最多使用255次。



主程序块

梯形图程序都包含一个 `_MAIN` 程序块，程序的执行总是从 `_MAIN` 程序块开始的。

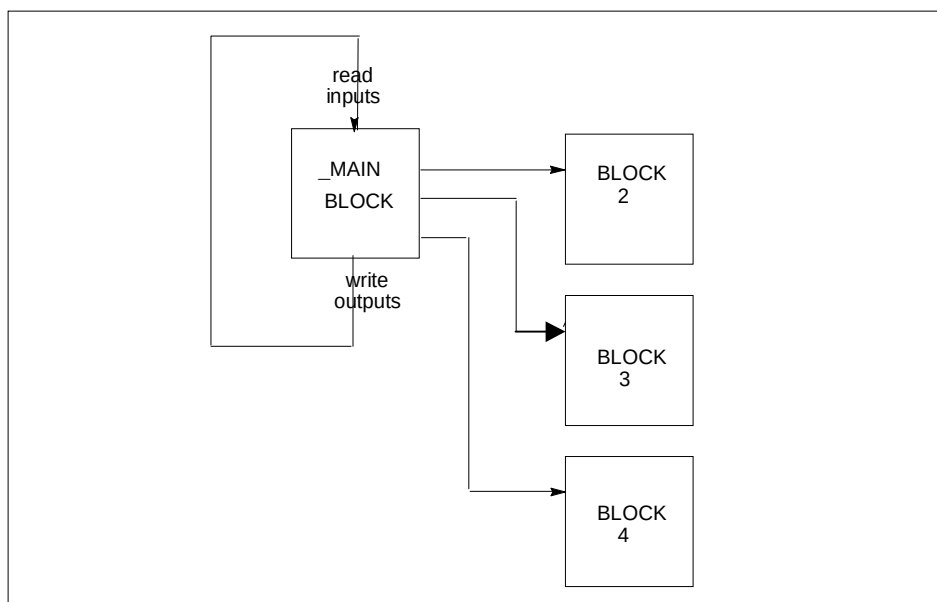


程序块

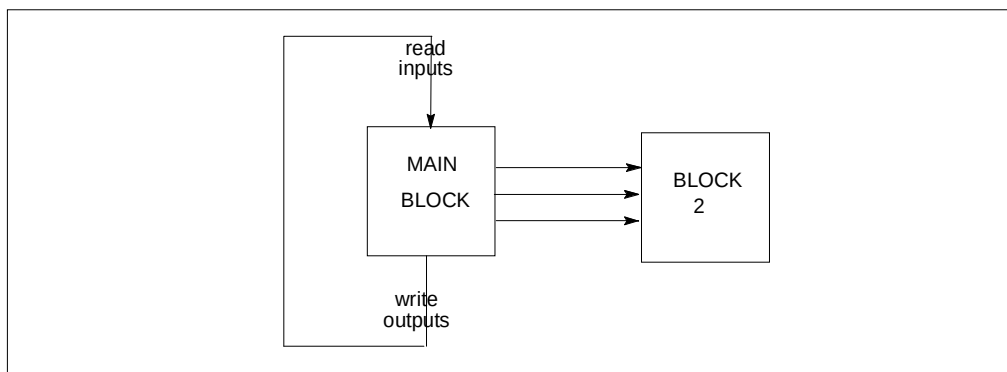
程序块作为一个程序结构可被重复调用，一些重复性的逻辑程序可使用一个程序块完成。使用 **Call** 指令来调用该程序块。这样可以减小程序量。将一个程序分为若干小程序块也有利于简化程序和降低程序所需逻辑指令的总量。

程序块使用举例

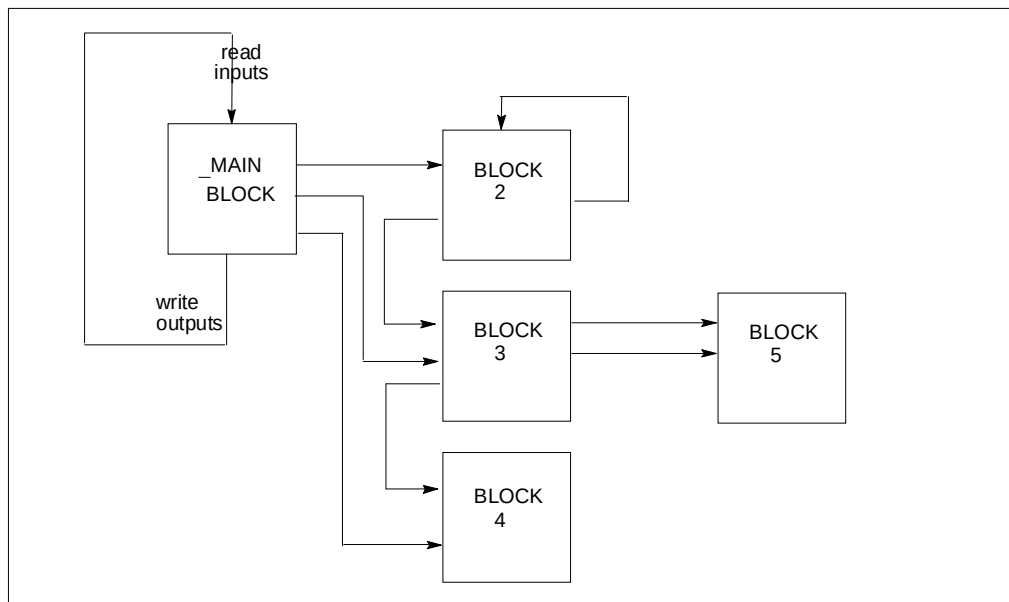
举例来说，如果一个程序可分为三个程序块，主程序块可根据需要调用其中任何一个（但程序块不能调用主程序），在本例中，主程序块可以只包含很少的逻辑，它主要对其他程序块进行时序的控制。



程序允许时，一个程序块可被多次调用。在程序中需要重复的逻辑可在一个程序块中完成。调用一次也就使程序执行了这些逻辑一次。



除了被MAIN程序块调用外，也可被其他程序块调用，甚至可以自调用。



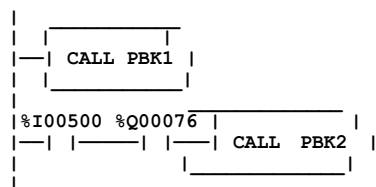
编程软件对程序块的调用的级别数没有限制。但PLC仅允许一定数量的嵌套调用，否则将会出现“应用程序堆栈溢出”的错误，并使PLC转换到Stop/Halt状态。对于731和732型CPU允许四层嵌套调用，其他型的CPU允许8层。程序中实际允许调用的层数还决定于程序块中所使用的非布尔型数据流的总量。如果使用的数据流总量小于171字，则可满足大部分的嵌套调用。嵌套调用的级数将主程序块计为第一层，上图所示为3级嵌套调用。

注意

必须在程序块声明中定义了的程序块才可在程序中调用。关于程序块声明的详细介绍，见《Logi-master 90-70 编程软件使用手册 (GFK-0263)》。

如何调用程序块

当一个程序块在被主程序块或其他程序块调用时，该程序块开始运行。

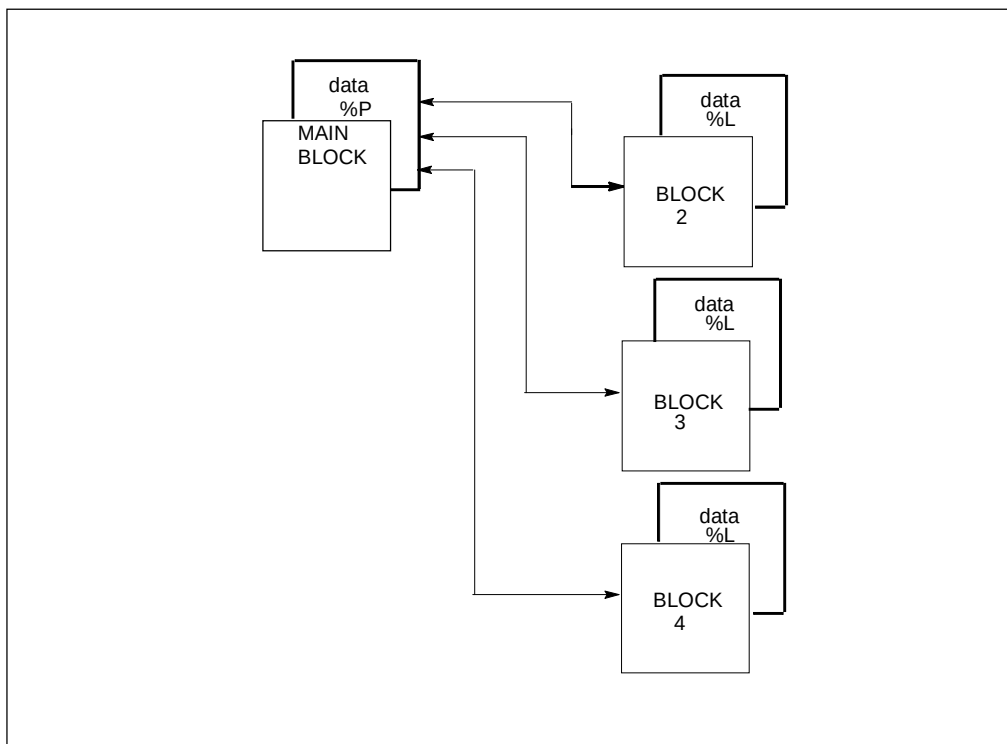


在上面的例子中，PBK1总是被调用。在调用时可以加一些条件逻辑，例如上面调用PBK2程序块，就需要%I00500 和 %Q00076都为ON状态。

程序块与局部数据

在梯形图程序中每个程序块都有与之相关的数据块。主程序的数据块用%P标记，其他程序块的数据块用%L标记。

数据块的大小由程序中使用的%P或程序块中使用的%L的最高地址决定。附录C“内存分配”提供了一个数据表来确认用户数据使用的总字节数以及还有多少字节可供程序使用。



梯形图程序中所有程序块都可使用与主程序块相关的%P数据。程序块可使用它们自己的%L数据，而%P数据对所有程序块是公用的。在主程序块中不可使用%L。

注意

外部程序块和参数化子程序块没有自己的%L数据，它们继承调用块的%L数据。

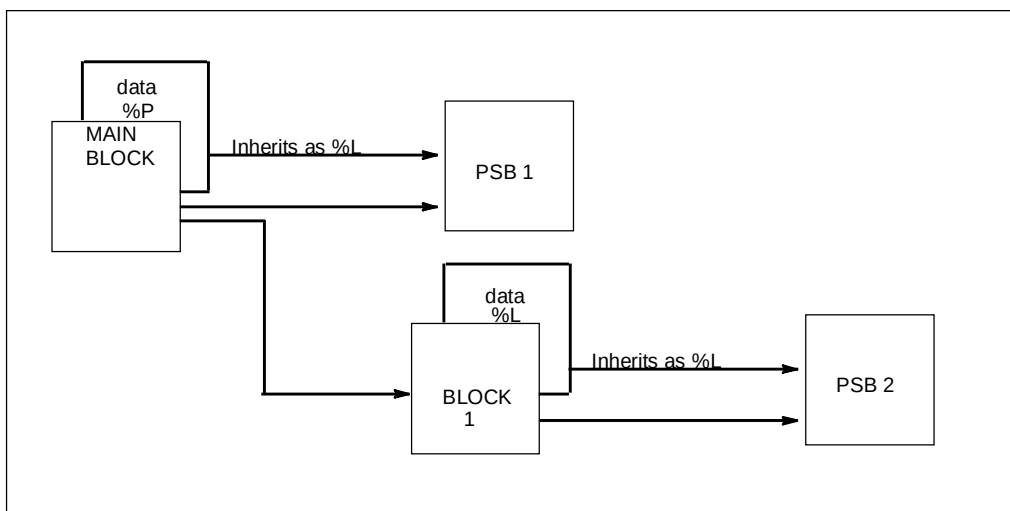
参数化子程序块

参数子程序块 (PSB) 是可选的，由用户定义的功能块，它可有0—7对输入/输出参数。

同其他程序块一样，参数子程序块可以被主程序块，其他程序块调用，也可以自调用。调用时需要为它指定参数。声明一个参数子程序块时，必须为它指定一个唯一的名称和需要使用的参数的对数及长度和类型。除使能和输出使能外的指定长度的每个参数可被定义为BOOL、WORD或 NWORD类型。BOOL长度范围为0到256；WORD和NWORD长度范围为1到1024。BOOL型的默认长度为1位，WORD和NWORD的默认长度为1字节。另外，可以为每个参数定义一个三个字符内的参数名。

参数化子程序块和局部数据

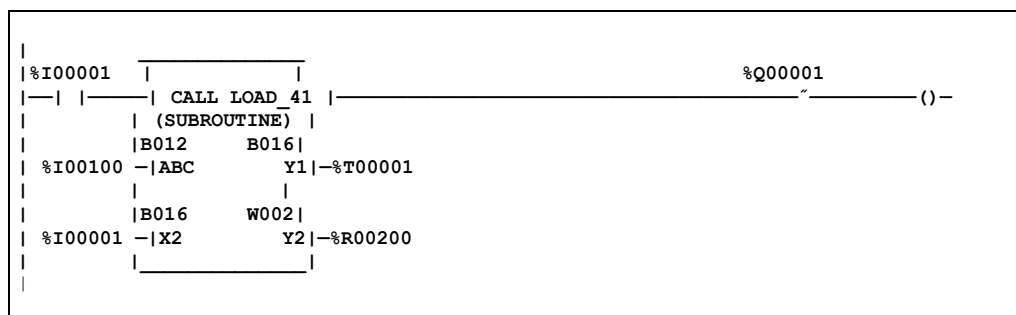
参数化子程序块可使用%P的全局数据。参数子程序块没有自己的%L 数据，但可继承调用它的程序块中的%L 数据。参数程序块同样继承%S接点，如FST_EXE。若在主程序中调用的参数程序块中使用了%L变量，无论用于参数化程序块的哪部分，则%L变量将由%P代替，例如，%L0005 = %P0005。



调用参数化子程序块

参数子程序块当在主程序块，其他程序块调用或自调用时执行程序逻辑。

下例中，若%I00001置位时，名称为LOAD_41的参数子程序块被执行。子程序块LOAD_41处理输入数据（变量地址为 %I00100 – %I00111 和 %I00001 – %I00016）并产生输出值（变量地址为 %T00001 – %T00016，以及寄存器内存地址 %R00200 – %R00201）。子程序逻辑也可控制参数子程序块的OK端的输出。这个例子显示了子程序块的调用（CALL）指令。



参数化子程序块中的形式参数

形式参数是参数化子程序块中用来与调用块之间传递数据的参数。它们的类型可为BOOL、WORD或 NWORD。NWORD型参数可用于任何多字型的操作数，但不能用于离散型数据。（NWORD是需要传给参数子程序块的整数个字）。

形式参数中，X定义为输入参数，Y定义为输出参数，后面跟的数字为该输入或输出参数的编号。例如，X2表示该参数在参数子程序块声明中的X2的位置上。X2后跟由子程序块声明中指定的长度，其值为1-16。

在参数子程序块中最多可有七对形式参数。形式参数的类型、数量和长度用如下格式表示：

ab[ccc]

此处	a = X	代表一个输入形式参数。
	a = Y	代表一个输出形式参数。
	b	参数个数 1 到 7。
	c	BOOL、WORD或 NWORD标记。

X1到X7和Y1到Y7可以被指定三个字符内的别名。

每个参数子程序块都有一个预定义的本地变量，YO，CPU每执行一次该程序块它就被置位一次。YO可由程序逻辑控制并作为程序块的一个输出状态。

所指定的参数是PLC变量或数据流，它们把地址（WORD或 NWORD型）或数据（BOOL型）传递给参数子程序块。参数在参数表中定义。

参数程序块中形式参数的限定

通常，BOOL型形式参数允许在所有接点、线圈和允许使用离散量变量(%I、%Q、%M、%T、%S、%G和%U)的功能块中使用。WORD和NWORD型形式参数允许在可使用寄存器变量(%R、%AI、%AQ、%P、%L和%UR)的功能块中使用。NWORD型参数仅用于多字型的参数（即DINT、DWORD或REAL）。NWORD是传递给参数子程序块的一组字数据。

下表列出了在参数子程序块中使用形式参数的几种例外及受限制的情况：

1. 触发型的接点和线圈，保持型线圈不允许使用形式参数。编辑器（编程软件中的编辑工具）会对这些功能块进行非保持型的等效替换。
+—(M)，+—(SM)，+—(/M)指令将会显示对应的警告信息。
2. BOOL型输入参数不能用作功能块的输出参数。
3. DO I/O 功能块不允许使用形式参数。
4. 多字型的功能块参数（即DINT、DWORD或REAL）仅允许以NWORD的形式参数出现。
5. 形式参数不允许作为以下功能块的参数：

功能块	参数
服务请求 (Service Request)	PARMS 输入参数
通讯请求 (COMMREQ)	IN 输入参数
DATA_INIT DATA_INIT_COMM DATA_INIT_PID DATA_INIT_ASCII	Q 输出参数

6. 参数子程序的BOOL型形式参数不可以由另一个参数子程序传递。
7. WORD型参数不能由其他子程序的NWORD输入参数传递。

PSB中的位型参数

PSB中的位型输出参数将会影响输出范围内的整个字节的状态。例如，一个长度为4的位型参数，连接到物理输出%Q6，将会影响从%Q1到%Q8及从%Q9到%Q16的整个字的输出。

由于对一个位型的数据组，组中的元素都是一个一个的被写入数据的。内存中的写操作和状态管理是按照字节的形式进行的，每元素的触发状态将被清除，处理队列末尾的元素。长度为1的位参数就没有此类问题，因仅一位的写操作在运行。然而，需要指出，此字节的其他位的触发状态将受到影响。

外部程序块

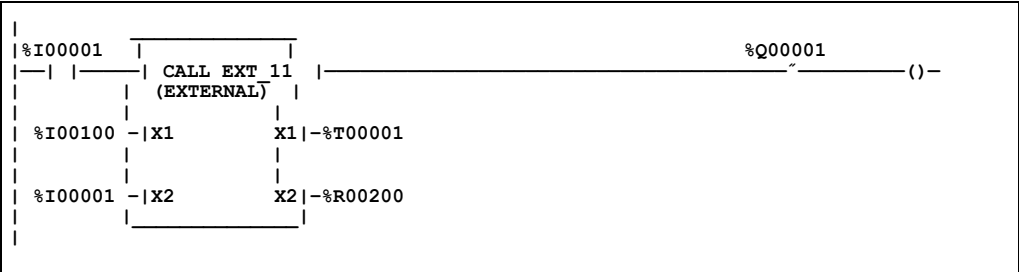
使用C程序工具包生成外部程序块。关于外部程序块的详细信息，参阅《系列90 C程序工具包(GFK-0646)》。

外部程序块的调用

通过编程软件中的库功能来向用户程序中添加外部块。

当外部块在程序逻辑的主程序块或其他程序块中被调用时执行。为方便数据的传递和数据的返回，外部块可以有0到7对的输入输出参数。

在下面的例子中，若%I00001置位，名为EXT_11的外部块被执行。块处理输入数据（变量地址为%I00100 – %I00111 和%I00001– %I00016）并产生程序块的输出数据值（变量地址%T00001 – %T00016和%R00200 – %R00201）。程序块中逻辑也可控制Enable Out的输出。

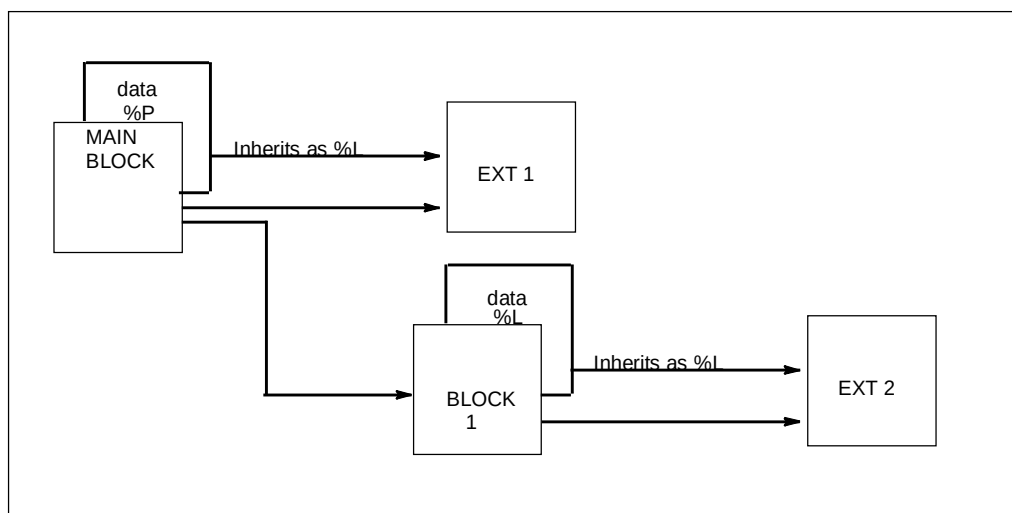


注意

与其他程序块不同，外部块不能调用其他程序块。

外部程序块和局部数据

外部块支持使用%P型全局数据。外部块没有自己的%L数据，但是可以继承调用它的程序块中的%L数据。外部块也继承%S接点，如FST_EXE接点的数据。如果%L数据在主程序块中出现的外部块中使用，则将对%L变量进行%P的等效转换(例如%L0005 = %P0005)。



局部数据的初始化

当一个外部块保存于PLC时，它的内部数据的一个副本将被保存。全局和静态初始化数据被保存，但如果静态变量在声明时没有赋初值，则必须由C应用程序初始化。（对于本节提到的“全局数据初始化”和“静态变量”部分内容，见《系列90 C程序工具包(GFK-0646)》第二章第六节。当PLC从Stop转换到 Run时，程序块的数据区通常要进行重新初始化。

外部/独立程序不能使用%L数据，但它内部使用的数据类似于局部数据。在独立程序内部使用的内部数据不会引起%L数据的混乱。

独立C程序

同外部块一样，独立程序由C程序工具包生成。然而，独立C程序最大可至564 KB。独立C程序不能调用其他的独立程序，也不能调用梯形图程序中的程序块。相应的，在梯形图程序中的程序块也不能调用独立C程序块。独立C程序块按照一系列可能的程序时序模式执行，本章第四节将对此作介绍。

注意

向文件夹中添加C程序的相关内容，见《Logicmaster90-70编程使用手册(GFK-0263)》。

一次最多可以使用16个独立程序。如果使用梯形图程序块，则最多可以使用15个独立程序，与梯形图程序共16个。

注意

由于独立C程序块是真正独立的程序，它们没有象LD程序中%L %P那样共享的内存类型。独立C程序块内部使用的局部变量不能混同于%L和 %P数据。

数据封装

每个独立C程序块都提供了获取用户变量的本地副本的方法。它能够处理本地数据组，而不是对用户的全局数据进行处理。这一特性由输入输出规范的使用来保证。

以下是对独立C程序块使用I/O规范时的运行步骤：

The following steps occur when using an I/O specification with a standalone C program:

1. 当程序被排序运行时，任何响应的输入规范由用户变量的全局数据复制到程序的局部数据区。
2. 程序执行时，对它的局部的输入输出数据进行处理。在运行时的任何中断不会影响到程序局部数据副本的输入或输出数据。
3. 程序完成时，它的局部输出规范将输出数据重新复制回用户数据变量区。

在使用输入输出规范时，有两点要特别注意。第一，若一个程序处于暂停状态，而在程序继续运行之前进行输出扫描，由于输出扫描值由用户变量区获得，输出值将保持一致。第二，若一个程序由另一程序中断，第一个程序就不在受由第二程序引起的全局数据的改变的影响，因为它所使用的是自己的局部数据。

使用输入规范的另一个好处就是对输入值进行更精确的采样。若程序的执行被更高优先级的程序延迟，输入规范可为程序提供一组比程序被延迟时更精确的一组数据，因为全局数据变量可能已被更高优先级的程序或输入扫描更新了。

输入、输出规范

与外部块不同，独立C程序与梯形图程序之间不允许有任何输入或输出参数。独立C程序块可使用输入或输出规范表，其中最多可有8个输入和8个输出。输入列可在程序开始执行时拷贝到程序中。输出列可在程序执行结束时由独立C程序拷贝到全局变量。需要说明与外部块不同的是参数是由变量来传递的，而不是变量值。这种操作对多周期扫描模式下的多个扫描时间分段的情况尤为重要。

小心

当PLC运行在多周期扫描模式时，程序可能正在执行当中，可能正在某行C代码，或逻辑和功能块中行执行中被延迟。如果程序使用%Q、%R等全局数据，在中断程序或输出扫描发生时可能会出现变量值的不一致。如果值没有由它的类型存取，则这种不一致也会在一个给定变量中发生。

如果程序使用全局数据并被多周期扫描延迟，则会产生数据的不连贯。数据在两个连续扫描中才会被更新。虽然数据在一个字节或一个字中连贯，但全局数据仅在使用它们基本类型的情况下才有效。而对于其中单个元素来说这些数据是不一致的。

这里举例说明输出数据时可能的不一致性。在下例中，一个给定的输出值将在几个特定位置才刷新数据。比如用户已在第二个程序中的输出列表中配置了%AQ1，并且变量%AQ1在两个程序开始执行时被赋值为0。

Differences Between Accessing Global and Local (Internal) Data

Program Accessing Global Data		Program Using I/O Specification	
<program begins>		<program begins>	
%AQ1 = 0	An output scan occurring after this line of code would output a 0 to AQ1.	LOC[1] = 0	An output scan occurring after this line of code would output a 0 to AQ1.
%AQ1 = %AQ1 + 2	An output scan occurring after this line of code would output a 2 to AQ1.	LOC[1] = LOC[1]+2	An output scan occurring after this line of code would output a 0 to AQ1.
%AQ1 = %AQ1 + 12	An output scan occurring after this line of code would output a 14 to AQ1.	LOC[1]=LOC[1] + 12	An output scan occurring after this line of code would output a 0 to AQ1.
%AQ1 = %AQ1 / 2	An output scan occurring after this line of code would output a 7 to AQ1.	LOC[1]=LOC[1]/2	An output scan occurring after this line of code would output a 0 to AQ1.
<program completes>		<program completes>	
Further output scans will output a 7 to AQ1.		The PLC copies the output specification from a local area back to the global areas when the program completes.	
		Further output scans will output a 7 to AQ1.	

注意

请注意独立C程序是真正独立的程序。它们不能访问梯形图程序中的本地内存(%L 和%P)。在其中使用的内部或“局部”数据不能混同于%L 和%P数据。

独立C程序和局部数据

PLC不为独立C程序提供本地数据副本。同样，它们不能访问梯形图程序的%P内存，也不能范围梯形图程序中程序块的%L内存。独立C程序的局部数据是在用于编制C程序的C源文件中被声明的。详细内容见《C程序工具包使用手册(GFK-0646)》。

局部数据初始化

当独立C程序保存至PLC时，内部数据的一个副本也同时被保存。当PLC由停止转入运行时，这些数据被重新初始化。

独立C程序中的I/O 列表数据

软件中为独立C程序定义了一些新的宏来与输入输出列表交互数据。关于独立C程序中的I/O 列表数据详细内容见《C程序工具包使用手册(GFK-0646)》。

I/O列表中的数据相关性

由于独立C程序可以被其他程序和中断块中断，在I/O列表中可能会出现数据时序的混乱。每个独立的数据表被限制最多为2048字节，其中最多16KB的输出和16KB的输入数据。90 70 PLC将按照如下规则运行。

在单独的I/O列表中的每个字节都是连贯的。

如果输入列表中的全部数据长度小于2048字节，则全部的输入列表是连贯一致的。

如果输出列表中的全部数据长度小于2048字节，则全部的输入列表是连贯一致的。

若输入或输出列表的总长度超过了2048字节，则总长度不超过2048字节的单独的列表组是时序一致的。见下表。

表 2-14. I/O列表的一致性

I/O specification	Length (bytes)
1	2000
2	48
Interrupts may occur	
3	1024
Interrupts may occur	
4	1026
Interrupts may occur	
5	10
6	20
7	20
8	20

梯形图与独立C程序的比较

在确定使用哪种编程语言编程时需要考虑某些方面的内容。下面总括了PLC所支持的几种编程语言的特点：

当考虑中断响应特性时，使用中断块较独立程序来说是首选的。处理一个中断独立程序的时间要远比使用中断块的时间长。

梯形图程序仅运行于全局变量数据。这将会在多周期扫描模式下运行的或是被延迟执行的梯形图程序引起数据时序一致性的问题。

每个独立C程序会占用8K的内存空间。

当程序中大量使用布尔指令时，要优先考虑使用梯形图方式。梯形图的编程方式更使用于继电器逻辑类型的编程。

表 2-15. 梯形图与独立C程序的比较s

	Programming Language	Program Size Limit	Data Types Accessible	Local Data Size	Scheduling Modes	Data Encapsulation	Block Types Supported
LD Ladder	Logic	smaller of 544K or available memory size, organized into 16K blocks.	All 8K	8k %P, %L per block	All*		No LD SFC PSB External
Standalone	C, using the C Programmer's Toolkit.	smaller of 564K or available memory size	All except %P, %L	unlimited, counts as part of program size	All * n/a		Yes

*Using Microcycle Sweep mode with an LD program is not recommended. LD programs always operate on global data directly, which can lead to inconsistent output values if the LD program is suspended mid-execution. Refer to section 4 of this chapter for more information on Microcycle Sweep mode.

程序运行的不同

数据的保持性

在独立C程序中没有线圈指令，故数据的保持性唯一的取决于内存类型。%Q 和%M都是保持型的。在C程序和梯形图程序中都用到的变量，其内存类型的保持性由梯形图中使用的类型来决定。详细内容见本章第二节“逻辑和数据的保持性”相关讨论。对于特殊内存类型保持性的介绍见表2-3。

全局数据

梯形图程序仅可使用全局数据区，故它们不能使用输入、输出列表。如果梯形图程序用在多周期扫描模式下，将会引起输出值时序的混乱。

执行中断

系统中梯形图逻辑中断块优先级最高。另外，它们不能被先占，而独立C程序则可以。

中断队列

在附加的中断队列中的程序和程序块有所不同。详细信息见2-64页“中断操作”及相关部分。

系统状态变量

当使用系统状态变量（旧版90-70参考书中的“快捷变量”）时，需注意如下几点：

变量 `FST_EXE` 在独立C程序中不可用。

`FST_SCN` 在C程序中使用时并不是指 `%S0001`，而是由C工具包提供同样功能用法的一个宏定义。

*对于系统状态变量的更多信息，见本章第二节表2-9。

第四节: PLC扫描模式和程序时序

一般扫描模式

一般扫描模式下，PLC每个扫描周期所消耗的时间不同。逻辑窗口在每扫描周期都完全执行。通讯和后台窗口按照设定的限制时间方式或运行完成的方式执行。一般扫描模式是PLC应用程序最常用的一种。

下图所示例子中列举了PLC在一般扫描模式下连续3个扫描周期内的执行情况。总的扫描时间可能由于在逻辑窗口，通讯窗口和后台窗口的时间的变化而不同。

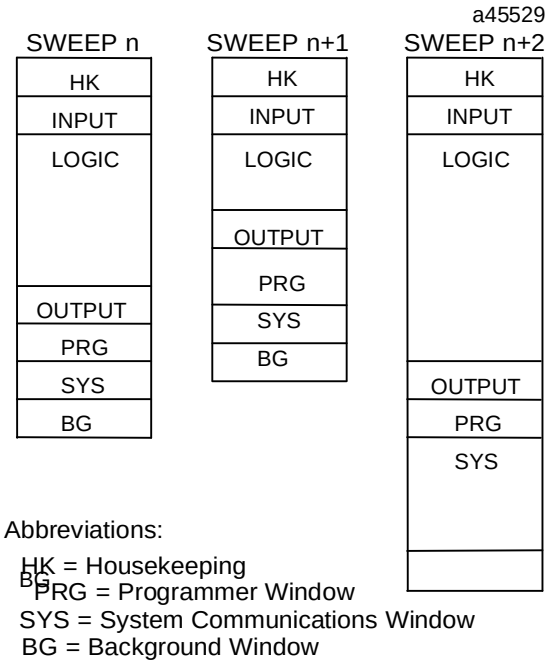


表 2-3. 一般扫描模式下的典型扫描

恒定时间扫描模式

在恒定时间扫描模式下，PLC扫描时间是一固定的常数。逻辑窗口在每个扫描周期内都完全执行，若仍有足够的时间完成本次扫描，则PLC将交替执行编程器通讯窗口，系统通讯窗口和后台窗口；运行它们以Run-to-Completion方式运行直到本次扫描时间到，才开始下一次的扫描。其中有一些或所有的通讯和后台窗口可以允许不执行完。当整个扫描时间达到所配置的时间常数时，通讯窗口和后台窗口终止。

使用恒定时间扫描模式的一个原因是确保按照固定的时间间隔刷新I/O数据。扫描时间常数可配置为3到255ms内的任意值。扫描时间以及扫描模式的改变可通过编程软件或通过程序的服务请求功能#1来进行。扫描时间没有默认值，若选择恒定时间扫描模式时必须为其设定该参数。

不管是以太网全局数据交换（EGD）的接收还是发送都将增加1ms的扫描周期的时间。当配置PLC为恒定时间扫描模式如果CPU看门狗超时将会引起扫描冲突。

如果PLC扫描时间超过了给定的时间，PLC将在PLC故障表产生一条超时报警信息，并在下次扫描开始时将OV_SWP(%SA0002)故障接点置位。当上一次扫描没有超时或PLC不在恒定时间扫描模式下时，该故障接点复位。由于超时扫描条件而引起的额外扫描时间不会影响到下次扫描的时间。

下图列出了恒定扫描时间为100ms的PLC的连续四个扫描周期的执行情况。虽总的扫描时间是一常数，但是也可能由于逻辑窗口的执行时间的延长而导致扫描超时。

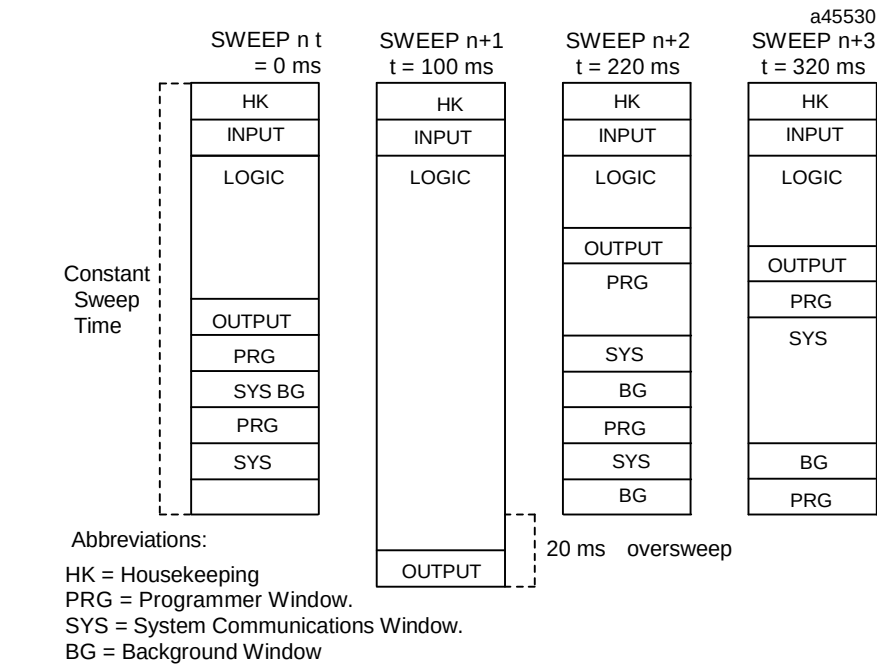


图 2-4. 恒定时间扫描模式的典型扫描

恒定窗口扫描模式

在恒定窗口模式下，PLC每个扫描周期所占用的时间是可变的。逻辑窗口在每个扫描周期内都完全执行，在这种模式下，PLC将交替执行编程器通讯窗口，系统通讯窗口和后台窗口；运行它们以Run-to-Completion方式运行直到给定的窗口时间到，才开始下一次的扫描。PLC总的扫描时间等于执行内务处理，输入扫描，逻辑窗口和输出扫描的时间与恒定窗口时间之和。该时间可能由于逻辑窗口每周期需要的时间不同而改变。

应用程序需要占用输出扫描和输入扫描之间的一段时间，允许输入在收到程序的输出数据后进行调整，这是恒定时间模式的理想情况。

恒定窗口模式的时间常数可配置为5到255ms之间的任意值。该常数也可由编程软件进行配置或由用户程序使用服务请求功能#3，#4，#5。

下图中列出了恒定窗口模式下的三个连续扫描周期的执行情况。总的扫描时间可能会由于逻辑窗口每次扫描的不同而不同。

但是通讯窗口和后台窗口的允许时间是固定的。根据所定的窗口时间的不同，一些通讯窗口或后台窗口可能被跳过，延迟或运行多次。

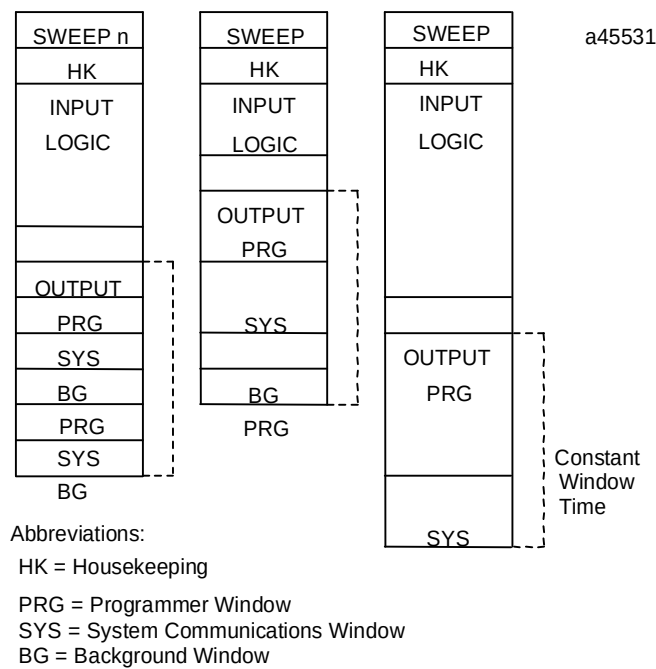


图 2-5. 恒定窗口模式的典型扫描

多周期扫描模式

在多周期扫描模式下，每个PLC扫描周期都开始于一个绝对时间—即与PLC由停止/运行转换相关的多重的基本循环时间。基本的循环时间决定了每个扫描所需时间（类似于恒定时间模式下的扫描时间常数）。用户可安排每个基于自己周期时间的执行时序，也可按照时间分段的方式运行多重PLC扫描。PLC将会交替执行通讯窗口和后台窗口，允许它们运行在Run-to-Completion方式下，直到下次扫描开始。

多周期扫描模式下运行某些程序的频率可以不同。对于重要的或运行时间要求严格的一些程序分配较多的时间。这种模式也允许程序当输入可用时执行多次。

虽然多周期模式有固定的扫描时间，但它与恒定时间模式有很大的不同。首先，用户程序不必每个周期都完全执行。为了维持基本循环时间和通讯、后台窗口时间，用户程序可以在执行中被延迟，并在下个扫描周期继续执行。同样，由扫描条件不同而引起的额外的扫描时间将会使下个扫描的时间变短。

在这种方式下每个扫描(with the exception of sweeps which follow an oversweep condition)开始于与PLC开始运行时刻相关的绝对时间。最后，确保每扫描有足够时间运行通讯窗口和后台窗口。必要时逻辑窗口将会被延迟开确保通讯窗口和后台窗口的运行时间。

基本循环时间和窗口时间值可配置为5到2550ms内的任意值。恒定窗口时间可以是5到255ms内的任意值。基本循环时间和恒定窗口时间可以在PLC停止状态下进行设定。它们不能在PLC的运行状态下修改。

多周期模式下，周期性的程序在优先的基础上执行。周期性的程序的优先级同它们的执行时间成反比（最小的周期优先级最高）。本章后续部分关于“用户程序执行”的讨论中对周期性程序和它们的执行作了详细介绍。

若PLC扫描超出所给定的基本循环时间，则PLC将在下次扫描开始时报超时报警信息于PLC故障表并置相应的故障状态位OV_SWP (%SA0002)。OV_SWP状态位在上一扫描不超时复位。由于超时条件造成的额外时间消耗将使下次扫描的时间缩短。

下图列出了在基本循环时间为100ms的多周期模式下的三个连续的PLC扫描周期。扫描时间是一常数，确保通讯和后台窗口运行于配置的窗口时间。在n和n+1次扫描中，逻辑窗口较早完成，一个I/O触发程序能够在该时间内完成。在n+2次扫描中，逻辑窗口由于没有完成而被延迟，以至于通讯和后台窗口能够在特定的窗口时间运行。无论何种情况，逻辑窗口对逻辑始终保持开放。

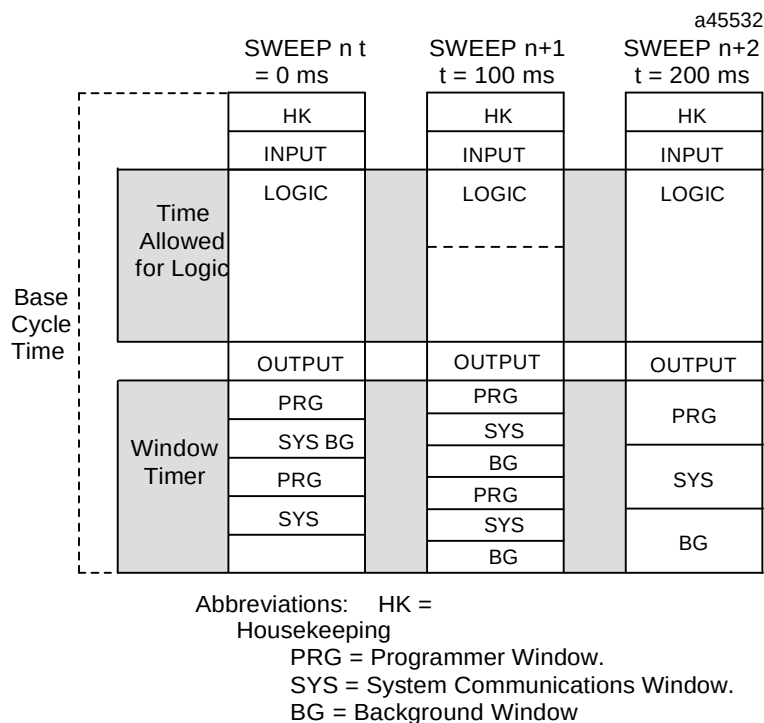


图 2-6. 多周期模式的典型扫描

注意

运行模式的逻辑存储不能用于多周期模式。同时也不支持单一程序测试特性。

微循环扫描模式输出扫描预计

微循环模式是由6.0版CPU才开始具有的一种新的扫描模式。在这种模式下，PLC按照与开始运行时刻相关的绝对时间开始每个周期的扫描。为了满足下一扫描开始的最后期限要求，CPU必须预计当前扫描输出扫描所需时间。

准7.00 CPU输出扫描预计

对于准7.00 CPU，对输出扫描的预计是在前次扫描周期的输出扫描进行的。例如，若第N个扫描周期的实际输出扫描时间为15.3ms，则CPU将会为第N+1个扫描周期分配15.3ms用于输出扫描，准7.00 CPU将会预计输出扫描为所配置的基本循环时间的三分之一。又如，若基本循环时间被配置为60ms，则第一次扫描的输出扫描将会预计为20ms。在这个例子中，若输出扫描实际并不是需要20ms，在PLC第一次扫描周期为逻辑窗口分配的时间将会比其它扫描周期的时间短。

当所编写的PLC逻辑第一次扫描时，确保在给定程序中逻辑执行完成优先于同样程序的其他执行时间。

7.00 和更新版CPU输出扫描预计

7.00 和更新版CPU，当前PLC扫描周期的实际输出数据量用于预计下次输出扫描时间。预计基于以经验为主的输出数据的类型和数量的测量时间。相当精确的估算结果，需按照逻辑窗口在保持用户配置的通讯窗口时间时所需的最大时间量来考虑。第一次扫描不需要特别的逻辑执行时间。

程序时序模式

90-70 PLC的每个用户程序能够执行于四种程序时序模式之一，依赖于扫描模式的制约。这部分将会简要介绍以下四种程序时序模式：

顺序模式

定时模式

I/O-触发模式

定周期模式

顺序模式	顺序程序同其他的所有时序程序执行于逻辑窗口。 顺序程序按照在编程软件中声明的时序，每扫描周期执行一次。顺序程序不支持微循环扫描模式。
定时模式	定时程序按照应用于PLC运行时刻的初始化延迟，以一定的时间间隔运行。定时程序是在PLC扫描周期的任何阶段基于优先基础的执行时序。定时程序不支持微循环模式。
I/O-触发	I/O-触发程序响应配置I/O中断时执行。PLC处于一般扫描模式，恒定时间扫描模式或恒定窗口扫描模式下的任何扫描阶段，I/O-触发程序优先执行。 在多周期扫描模式下，I/O-触发程序在逻辑窗口优先时序执行。在这种情况下，I/O-触发程序的执行可以在多个扫描周期内分段完成。
周期模式	周期程序由基于程序用户配置的周期来确定执行时间，来确定在逻辑窗口中与其他周期性程序一起执行的时间。周期程序按照与其他所有程序相关的优先级别执行，并可通过多次扫描周期来分段完成。周期程序仅用于多周期扫描模式下。

选择PLC的扫描模式和程序时序方式

下表列出了每种PLC扫描模式下的各种程序时序方式：

表 2-16. 各种PLC扫描模式下的各种程序时序方式

Sweep Mode	Program Scheduling Mode				Interrupt Blocks
		Ordered	I/O Triggered		
Timed			Periodic		
Normal			Yes	Yes	
Yes	No	Yes			
Constant			Yes	Yes	
Yes	No	Yes			
Constant Window	Yes		Yes		Yes
	No		Yes		
Microcycle		No		No	
Yes *	Yes	Yes			

* Executes in Logic Window only.

用户程序的执行

用户程序优先级

用户程序优先级是一种该程序与其他程序相关关系的属性。较高优先级的程序要比低优先级的程序先被执行。如果存在两个或多个具有相同优先级的程序处于相同的执行时序，则它们执行的时序将是不确定的。程序在执行过程中可以被较高优先级的程序或中断功能块延迟执行。

顺序程序都具有相同优先级，它们按照编程软件中指定的顺序依次执行。顺序程序较定时程序，I/O触发程序和中断功能块具有较低的优先级。

周期程序的优先级与其周期时间长短成反比（最小周期的优先级最高）。具有相同周期时间的多个程序的执行顺序是不确定的。周期程序较定时程序，I/O触发程序和中断功能块具有较低的优先级。

定时程序和I/O触发程序较顺序程序和周期程序具有更高的优先级。定时程序和I/O触发程序的优先级是由定时和I/O触发程序本身来决定的。定时程序和I/O触发程序的优先级范围为10-99（10为优先级最高），它们可运行于PLC扫描的任何阶段（即并不只是局限于逻辑窗口中运行）。定时程序和I/O触发程序在一般扫描模式、恒定时间扫描模式和恒定窗口模式下都按照这种方式运行。在多周期模式下，I/O触发程序在逻辑窗口执行，这种方式下的优先级范围为100-109（100为最高优先级）。

定时和I/O中断功能块在任何用户逻辑中都具有最高的优先级。

表 2-17. 定时程序和I/O触发程序的优先级

Scheduling Mode	Sweep Mode	
	Normal Sweep Constant Sweep Constant Window	Microcycle Sweep
Ordered * Executed in order	declared in the programming software.	Not supported.
Periodic *		Not supported. Smallest period has highest priority.
I/O-Triggered 100-109		10-99
Timed Not supported.		10-99

* Ordered and Periodic scheduling modes have lower priority than Timed and I/O-Triggered scheduling modes.

在一般扫描、恒定时间扫描和恒定窗口模式下的程序执行

在一般扫描、恒定时间扫描和恒定窗口模式下，90-70 PLC能够象执行定时程序和I/O中断功能块一样执行顺序程序和I/O触发程序。

顺序程序每扫描周期在逻辑窗口完全执行一次。程序按照编程软件中所设定的顺序依次执行。执行程序前首先要进行输入列表，并在程序完成后拷贝相应的输出列表。在这种方式下，必要时一个程序的输出可作为下一个程序的输入使用。

下图列出了在一般扫描模式下的两个顺序程序（A和B）的一个典型扫描周期：

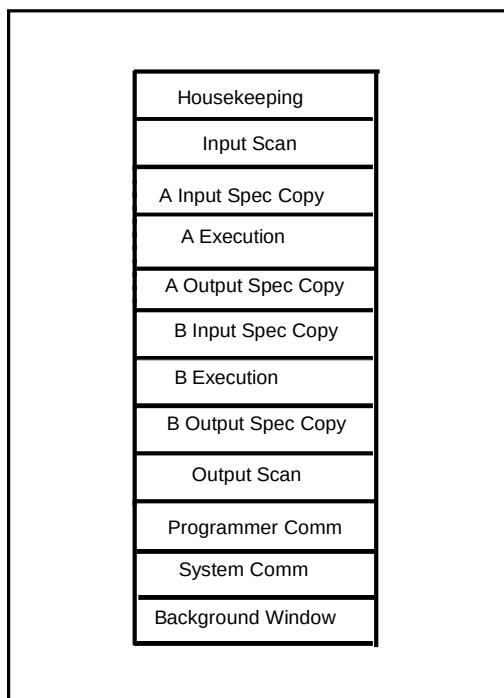


图 2-7. 顺序程序执行时序

在一般扫描、恒定时间扫描和恒定窗口模式下，定时程序和I/O触发程序可运行于PLC扫描周期的任何阶段。这些程序将顺序程序和具有较低优先级的定时和I/O触发程序优先执行。输入列表在程序确定执行时序时被复制（即达到一定的时间间隔或是发生I/O中断）。输出列表在程序完成时复制。

定时和I/O中断功能块可执行于PLC扫描周期的任何阶段。这些程序优先于其他所有程序，在PLC的用户逻辑里拥有最高的优先级。定时和I/O中断功能块没有输入或输出列表复制。

下图给出了两个顺序程序（A和B），一个优先级为10的I/O触发程序（C），一个优先级为20的定时程序（D）和一个I/O中断块在一般扫描模式下的一个典型PLC扫描周期的运行情况。

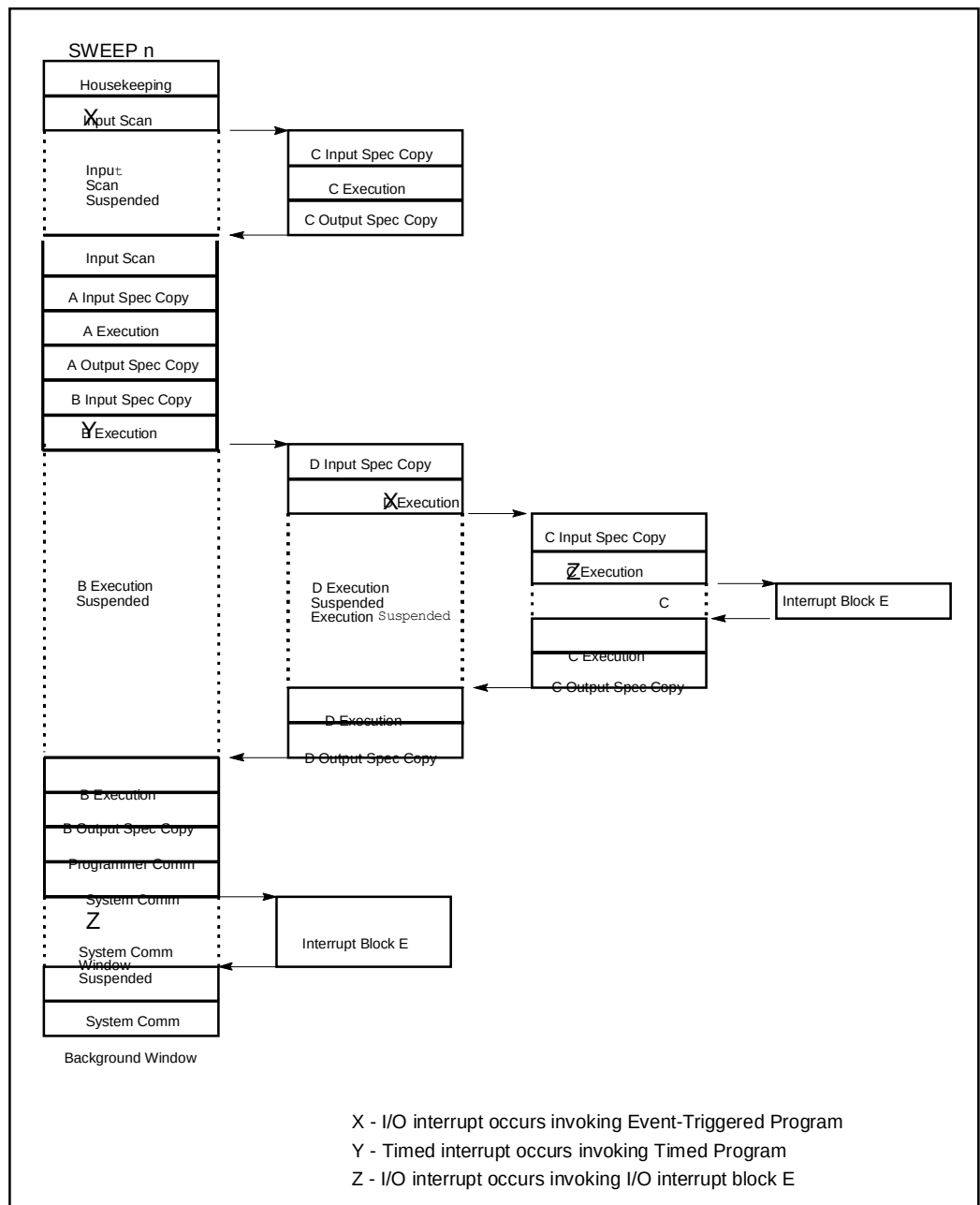


图 2-8. 顺序, 定时, I/O触发和中断功能块的执行时序

多周期扫描模式下的程序执行

在微循环扫描模式下，90-70可象执行定时和I/O中断块一样执行周期程序和I/O触发程序。

周期程序在逻辑窗口中执行。这些程序由程序的周期确定执行时序。例如，一个周期为1的程序将在PLC的每个扫描周期都被执行，而周期为2的将隔一周期执行一次。周期程序的优先级与其周期长短成反比（最短周期的优先级最高）。这些程序易于在逻辑窗口现有时间的基础上分段执行。与顺序程序不同，所有在给定扫描周期执行的周期程序输入列表拷贝在逻辑窗口开始时进行。也就是说，所有输入列表拷贝将在任何周期程序开始或继续执行之前进行。输出列表拷贝在程序完成时进行。

下图给出了在微循环扫描模式下两个周期程序（A和B）的一个典型扫描周期的执行情况：

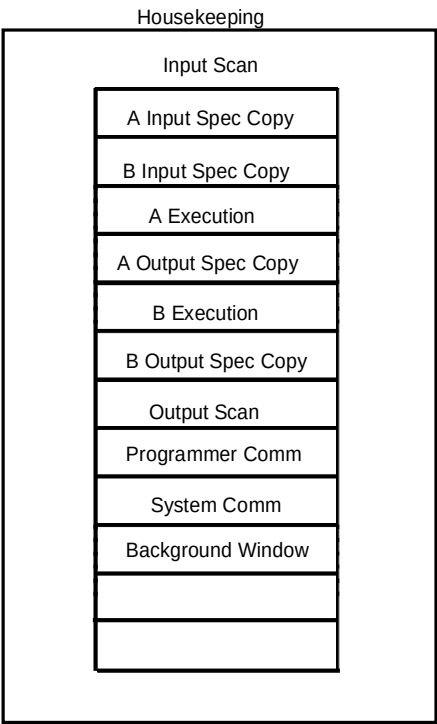


图 2-9. 周期程序执行时序

不同于其他扫描模式，当PLC运行于微循环扫描模式时，I/O触发程序仅在逻辑窗口执行。若I/O中断发生在逻辑窗口运行期间，或者说在逻辑窗口完成前，I/O触发程序将在当前扫描周期的逻辑窗口中进行。否则，将在PLC下个扫描周期的逻辑窗口执行。I/O触发程序将比周期程序和低优先级的I/O触发程序优先执行或恢复。这些程序同周期程序一样易于在逻辑窗口现有时间的基础上分段执行。输入列表拷贝与程序运行同时（即当I/O中断发生时）进行，而不象周期程序在逻辑窗口开始时进行。输出列表拷贝在程序完成时进行。

微循环扫描方式下定时和I/O中断块可在PLC扫描的任何阶段进行。这些块优先于其他所有程序执行，并在用户逻辑中具有最高的优先级。定时和I/O中断块没有输入或输出列表拷贝。

下图给出了两个连续微循环扫描周期内的两个周期程序（A和B）和一个I/O触发程序（C）的执行情况。周期程序A和B的周期都为1。

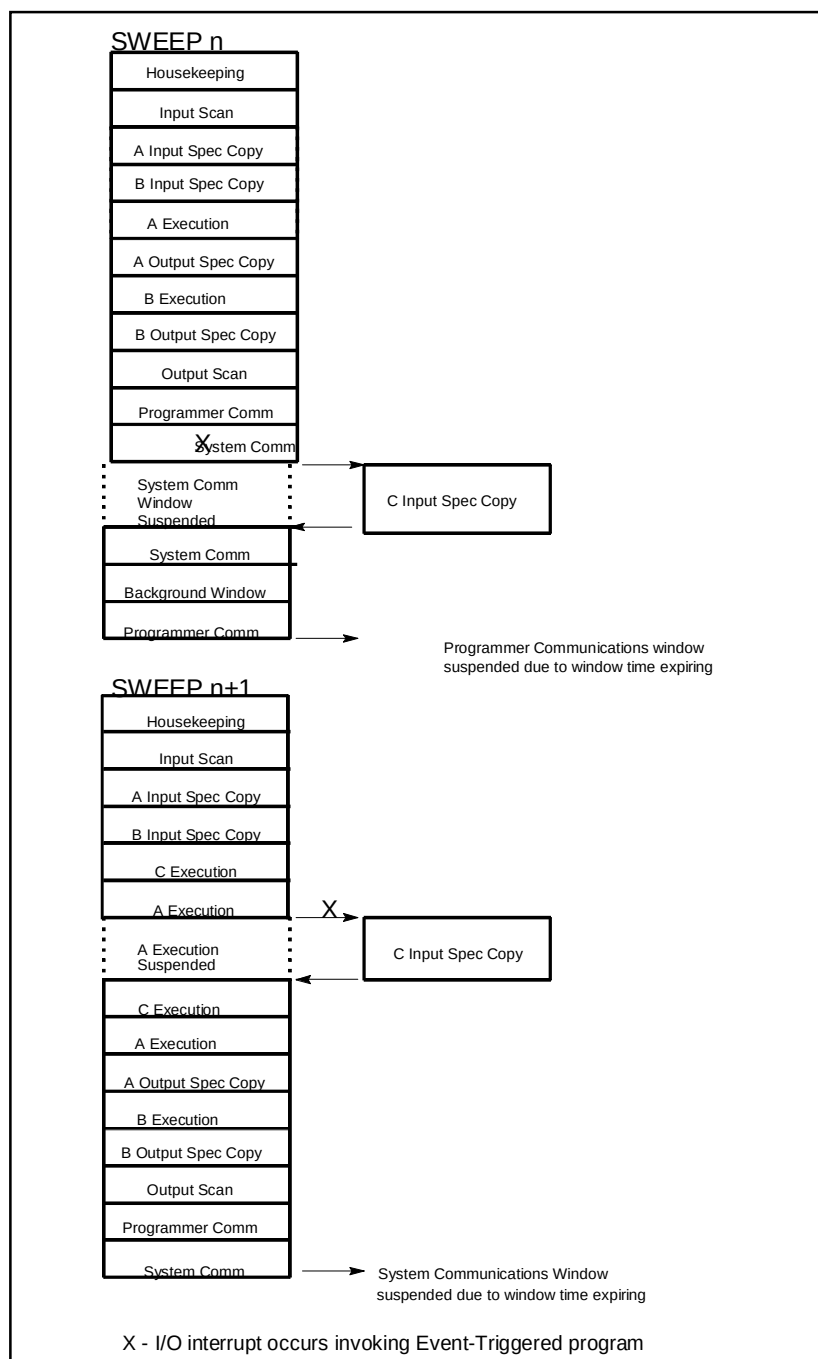


图 2-10. 周期和I/O触发执行时序

在微循环模式下的全局数据

被多重扫描延迟的程序中使用的全局数据（%R, %I, %Q等）将会引起数据的时序混乱。参考的数据将会来自两个连续的扫描周期。既然数据不会在一个字节或一个字中时序混乱，全局数据就可以仅用于它们的基本形式（字节或字等）；否则，个别的元素将会产生时序混乱。如果可能的话，输入和输出列表将用于存取和刷新全局数据区。

中断处理

90-70 PLC的用户程序处理中可使用两类中断：

I/O 中断	此类中断由90-70 I/O模板产生来显示离散量输入状态变化（上升沿或下降沿），模拟量范围限幅（低/高报警），高速计数事件和来自于第三方VME模板的中断。
定时中断	此类中断由90-70 PLC CPU产生，它按照初始延时基于用户定义的时间并与PLC运行时刻相关的时间进行。

两种类型的中断都可包含用户程序或功能块。

小心

中断功能块和程序的执行可被象其他定时程序和I/O触发程序之类的非中断逻辑中断。如果中断逻辑和被中断的逻辑同时存取相同的数据将会导致数据结果的不确定性发生。服务请求功能块#17或#32在访问共享数据时，可用于临时屏蔽I/O和定时中断块或程序。

I

中断处理和功能块时序

在系统用户逻辑中中断功能块具有最高的优先级，并可通过由定时或I/O中断块的响应通过编程来触发。定时或I/O中断程序触发的功能块在一般PLC扫描活动的执行中具有优先性。一般PLC扫描活动在中断功能块结束后恢复。最多可有64个中断功能块和16个定时中断功能块。

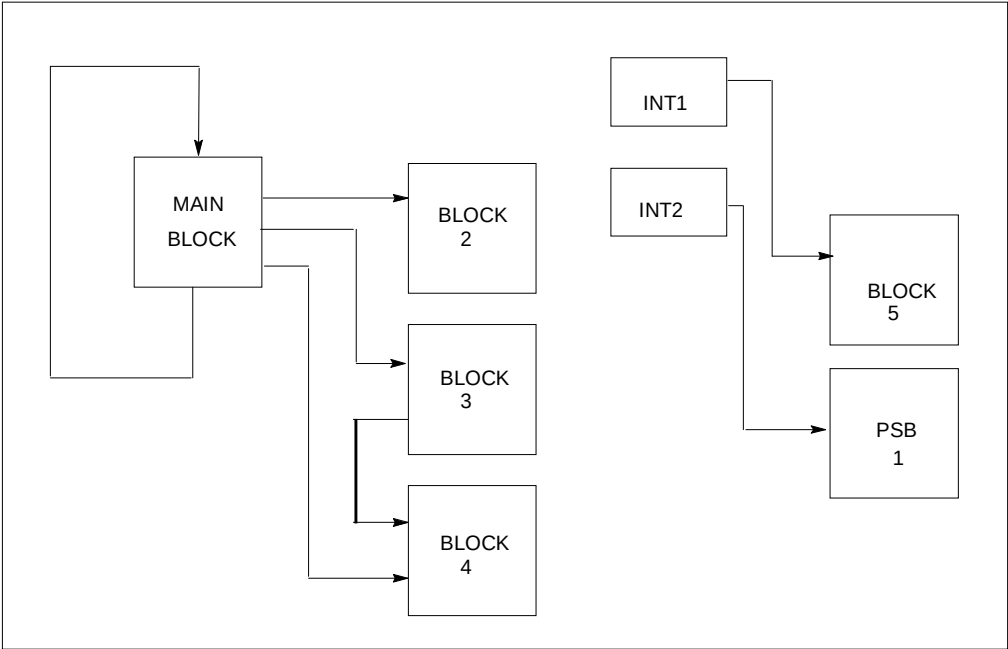
注意

使用于作为定时或I/O中断块的结果的功能块中时，计时器功能块不累计时间。

从6.0版开始的PLC CPU，梯形图中断块中可以调用其他程序块。在中断块执行中所使用的应用程序堆栈不同于梯形图程序中的。因此，嵌套调用限制不同于主程序块中对调用的限制。如果调用结果没有足够的站地址来完成本次调用，PLC将产生“应用程序堆栈溢出”错误日志，同时PLC将转入Stop/Fault模式。

注意

作为一个定时或I/O中断程序的结果来执行的功能块不能在主程序块或其他非中断功能块中调用，因为由程序块执行的部分代码没有重新加载。下例中，INT1，INT2，BLOCK5和PSB1不能在_MAIN，BLOCK2，BLOCK3或BLOCK4中调用。



I/O I中断程序块

程序块可以由特定的硬件模板的中断输入触发。例如，32点24VDC输入模板（IC697MDL650）的第一个输入可被配置为无论是该输入信号的上升沿或下降沿都产生一个中断。若模板以这种方式配置，输入可作为触发梯形图程序块或外部程序块的条件。

编辑一个I/O中断块，必须首先在编程软件中声明该程序块。它必须通过使用中断声明来与某中断相关联起来。

下图中列出了两个I/O中断的声明。在配置方向中，如果由一个停止按钮来的输入接线至输入1，触发器ST BUT调用梯形图程序块INT1。模板可被配置为上升沿或下降沿产生中断。梯形图程序块INT2则由%AI00009触发。

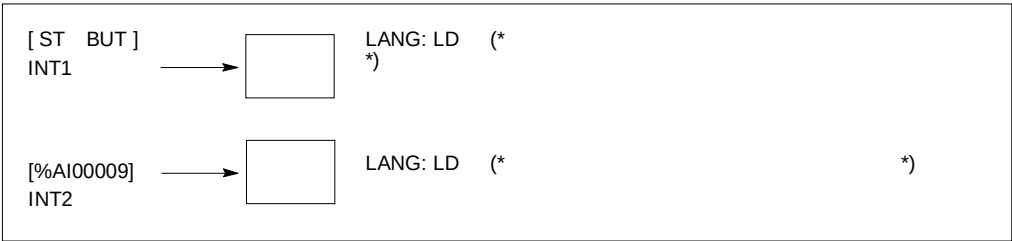


图 2-11. I/O 中断程序块的声明

注意

不带参数的参数子程序块(PSB) 和外部程序块（C程序块和C FBK）也可由一个中断输入来触发。（在控制软件中，无参数的子程序块被称为SUBR。）在这些类型的程序块中，局部数据（%L）来自于_MAIN程序块的局部数据（%P）。例如，%L0005 = %P0005。

定时中断程序块

在PLC从停止转入运行状态，经过一个初始延迟，程序块可按照用户设定的时间间隔来执行。在Logicismaster软件中，定时中断块的时间单位可为1.0秒，0.10秒，0.01秒。在控制软件中时间单位可为0.001秒（在任务定义对话框中简写为ms）。

必须先声明才能编辑一个定时中断块，它必须通过使用中断声明给定一定间隔和初始延迟时间，才能于定时中断相关联。

一个定时中断块的第一次执行将在PLC进入运行状态后的((DELAY * 时间单位) + (INTVL *时间单位))时间内进行。下图给出了两个定时中断的声明。梯形图程序块BLK1将会在PLC开始运行后的3秒，5秒，7秒等等时刻执行。梯形图程序块BLK2将会在PLC开始运行的两秒后开始并按照2秒的时间间隔执行。BLK2中缺少DELAY值说明在该块首次执行时没有初始延迟。

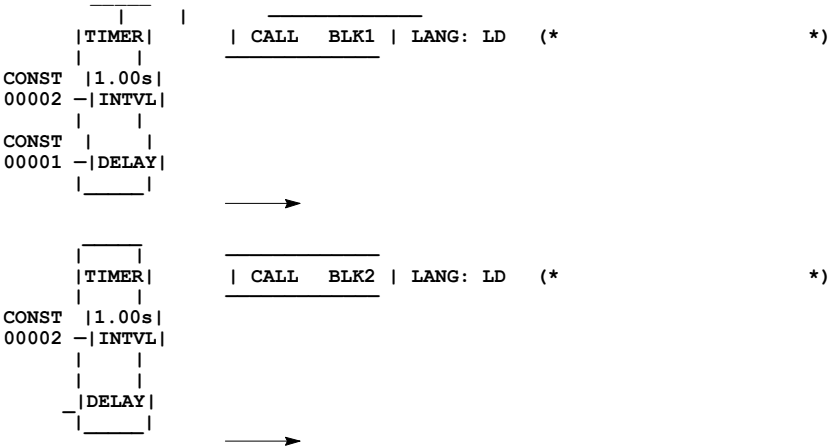


图 2-12. 定时中断块的声明

参数:

Parameter	Description
INTVL	INTVL is a constant value which will be multiplied by the time base of the interrupt to establish the frequency of execution of the associated block.
DELAY	DELAY is an optional field for the timed interrupt. It is a constant value which will be multiplied by the interrupt time base to establish an additional delay for the first execution of the associated block.
	Series 90™-70 PLC CPU Instruction Set Reference Manual–January 2000

Valid Memory Types:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ
%UR	const	none											
INTVL													
•													
DELAY													
•													
•													
=													

Valid data type, or place where power may flow through the function.

注意

对于不带参数的参数子程序块(PSB)和外部程序块(C块或C FBK)也可以由定时中断触发。
(在控制软件中不带参数的子程序块称为SUBR。)对于此类程序块，本地数据(%L)来。

中断处理和用户程序时序

I/O触发程序

从6.0版的PLC CPU开始，用户程序可以使用通过I/O中断激活程序的时序模式。当所配置的I/O中断发生时，I/O触发时序模式允许由用户程序调用，并与相应的输入和输出列表拷贝一起进行。I/O触发程序可在PLC扫描周期的任何阶段进行或仅在逻辑窗口中进行。有关I/O触发程序的时序和执行的详细信息，参见本节前面部分“用户程序执行”。

必须先 在编程软件中声明才能编辑一个定时中断块。时序模式必须设定为中断触发模式。

定时程序

从6.0版的PLC CPU开始，用户程序可以使用通过I/O中断激活程序的时序模式。当所配置的定时中断发生时，定时时序模式允许由用户程序调用，并与相应的输入和输出列表拷贝一起进行。定时程序可在PLC扫描周期的任何阶段进行或仅在逻辑窗口中进行。有关定时程序的时序和执行的详细信息，参见本节前面部分“用户程序执行”。

必须先在编程软件中声明才能编辑一个定时块。时序模式必须设定为定时中断模式。

定时程序的时间单位为毫秒。定时程序的首次执行发生在PLC进入运行模式时刻的[(初始延迟时间) + (时间间隔 * 时间单位)]毫秒后。

中断程序块和中断程序的比较

当选择程序或程序块来处理中断时应注意它们之间的差别。当选择使用程序块来处理定时或I/O中断时，程序块将在接收到中断指令时立即执行直到它运行完毕。中断块可在PLC扫描周期的任何阶段执行，不论当前PLC处于何种扫描模式。等待执行的定时中断块将在I/O中断块之前被执行，但是一旦中断块（定时或I/O中断块）开始执行，它将不会被中止。对于一个中断，如果该中断试图执行一个由于先前的中断而没有完全执行的定时或I/O中断块，那么该中断将进入等待队列，程序块将在中断程序块执行完毕后被重新执行。如果该中断块已经通过这种方式进入过等待序列一次，则所产生的任何其他的中断将被该程序块忽略。

在接收到中断后，定时和I/O触发程序立即进入执行序列（包括输入列表拷贝）。然而，程序实际的执行仍按照优先原则进行。与中断块不同，定时和I/O触发程序的执行可被其他具有较高优先级的定时和I/O触发程序延迟或提前。另外，对于一个中断，如果该中断试图执行一个由于先前的中断而没有完全执行的定时或I/O触发程序，那么将在PLC故障表中报“Program not Readied”的应用程序错误信息，并且此中断不会被程序响应。

当PLC处于一般扫描模式，恒定时间扫描模式或恒定窗口扫描模式时，中断程序可在PLC扫描周期的任何阶段被执行。当PLC处于多周期扫描模式时，中断程序只能在逻辑窗口中执行。

综上所述，中断块与中断程序注意有以下几点不同：

中断块

在收到中断指令时立即执行。

一旦中断块开始执行，就不能被其他逻辑中断。

如果该程序块由于先前的中断仍在执行中，另外的中断将被延迟执行。

可执行于扫描周期的任何阶段。

中断程序

根据优先原则进行排序执行。

可以被具有更高优先级的中断程序或中断块中断。

如果该程序由于先前的中断仍在执行中时发生另外的中断，则该中断不会被响应，并且会报相应的故障信息。

当PLC处于一般扫描模式，恒定时间扫描模式或恒定窗口扫描模式时，中断程序可在PLC扫描周期的任何阶段被执行。

当PLC处于多周期扫描模式时，中断程序只能在逻辑窗口中执行。

第五节: 运行/停止操作

运行方式

90-70 PLC支持四种运行/停止方式。可通过以下手段来改变PLC的运行方式：强制开关，编程软件，梯形图功能块以及C应用程序中的系统函数调用。在不同的运行方式之间的切换要受到基本优先权的限制，比如PLC强制开关的位置，安全密码等等。

Run/Outputs Enabled

在这种方式下，PLC运行用户程序并连续扫描输入，更新物理输出，包括Genius和现场总线的输出。编程器和系统通讯窗口运行于Limited，Run-to-Completion或Constant方式。

Run/Outputs Disabled

在这种方式下，PLC运行用户程序并连续扫描输入，但并不更新物理输出，包括Genius和现场总线的输出。这种方式下物理输出将保持在用户配置的默认状态。编程器和系统通讯窗口运行于Limited，Run-to-Completion或Constant方式。

Stop/IO Scan

在这种方式下，PLC不运行用户程序但进行输入和输出扫描。编程器和系统通讯窗口运行于Run-to-Completion方式。后台窗口被限制在10ms。

Stop/No IO Scan

在这种方式下，PLC不运行用户程序也不进行输入和输出扫描。编程器和系统通讯窗口运行于Run-to-Completion方式。后台窗口被限制在10ms。

注意

在运行方式下不能保存对%P和%L变量地址的改变，除非%P和%L变量是该块中首先保存的类型或者在新建的程序块中。

注意

对于6.0版所有型号 CPU，Stop/IOScan都不支持微循环扫描模式。

模式的转换

Stop-to-Run 转换

在CPU进行 Stop-to-Run 转换时，将进行一些操作，这些操作包括：

确认所选择的扫描模式和程序时序模式。

确认程序使用的变量及实际配置的大小。

为外部程序块和独立C程序重新初始化数据区。

清除非保持内存中的数据。

Run-to-Stop 转换

Wind-Down Period for Microcycle Sweep Mode

当PLC运行于微循环扫描模式时（参见第四节相关信息），在PLC收到停止指令后将会产生接下或逻辑解决时间段。接下时间段等于当前执行的程序完全执行总时间（不超过2.5秒）。如果当前执行的程序完成所需时间超过2.5秒，则将报相应的故障信息于PLC故障表，CPU将完成到停止方式的转换。在接下时间段内，将不会再有另外的程序（包括中断程序和中断块）在排到执行序列中。输入扫描，输出扫描，通讯窗口（编程器和系统通讯窗口），后台窗口将在接下时间段内继续进行。

注意

由上面的说明，当超过CPU接下时间，意味着在PLC转入停止模式前并没有完全执行所有程序。而且，当PLC再接到运行指令时，所有的程序将仍按照正常开始的节点重新开始运行。程序并不按照停止时刻的节点继续运行。

第六节 上电和掉电时序

上电时序

系统上电过程包括如下部分：

- 上电自检
- PLC 内存有效性检查
- 系统配置
- 智能可选模块自检完成
- 智能可选模块双端口检测
- I/O 系统初始化

上电自检

系统上电时，系统中的许多模块都会执行一个上电诊断性自检。系列90 70 PLC模板执行硬件检测和软件有效性检查。智能可选模块运行配置和板载微处理器检查，软件检查验证，本地硬件有效性检查，以及通知CPU自检完成。任何识别的检测在系统循环配置期间形成队列向CPU报告。

对CPU来说，上电可分为快速上电（热启）或完全上电（冷启），这主要由上电完成后CPU是否转入运行方式来决定。如果CPU运行于RUN模式下的所有条件均满足一存在有效的程序和配置，运行开关在RUN的位置，没有致命故障发生一那么CPU将执行快速上电过程。如果其中的任何一个条件不满足，将执行完全上电过程。

快速上电过程只执行CPU处理器和BCP（布尔运算协处理器）测试，以及最少量的RAM测试。快速上电的目的是启动CPU并尽可能的快速运行起来。对其他的检测，如ROM CRC，内存完全检测和外围设备检测仅在完全上电过程才进行。

如果当前电池电量不足，则于PLC故障表报电池电量低故障信息。

PLC内存有效性检测

系统上电的下一步是CPU对PLC内存有效性检测。首先，系统确认电池电量充足，并且由电池维持的RAM区域数据有效。检查应用程序RAM的已知区域的数据是否已经保留起来。接下来，如果有梯形图程序，则通过主程序块计算检验和。如果没有梯形图程序，则通过最小的独立C程序来计算检验和。

当系统确认应用程序RAM已经保存时，同时检测已知BCP（布尔运算协处理器）位缓存区域来确定BCP位缓存数据是否被保存。如果检测通过，则位缓存用于保持它的上电值。（非保持型输出数据在有停止转入运行模式是被清除。）如果检验和不对照或对应用程序RAM的保持性检测失败，则位缓存将判断为出错状态并清除该区域所有数据。此时PLC处于清除状态，同刚刚安装的CPU模块时的状况相同。所有的逻辑和配置文件必须重新从编程器下装到PLC。

系统配置

完成CPU自检后，下一步进行系统配置。首先它清除BCP位缓存中所有的系统诊断位。这样可避免系统掉电前存在，而现在不再出现的故障仍作为故障保留下来。接着，它对系统中的每块模板进行登记，并检查完成各模板自检。

CPU从每块模板读取信息，比较它们是否与用户配置信息相符。若实际配置与用户程序中的配置不符，则会报相应的配置错误于故障表。

智能可选模块的自检完成

智能可选模块可能会比CPU分配用来检测通讯介质或其他接口设备的更长时间来完成它们的自检。再智能模块完成它的初始化自检时，它会高速CPU完成它们的自检剩余部分所需时间。在这段时间内，CPU会提供出模板完成配置所需要的任何辅助信息，继而继续完成自检及自配置，如果模块没有及时向CPU报告它指定的信息，CPU会标记此模块出故障，并记录于故障表。当所有自检完成时，CPU会得到在模块上电自检时产生的各种信息并在故障表中记录故障信息。

智能选择模块的双端口检测

完成智能模块的自检并报告结果后，整个检测全部投入对用于CPU与智能模块通讯的双端口的检测。这些检测确认这两个模块可进行信息的传递，同时校验通讯协议所需的中断及信号传输容量。完成双端口检测后，对通讯信息进行初始化。

I/O 系统初始化

如果使用的是70系列的输入模板，则不需要对其作更多的配置。若使用70系列的输出模板，则它将被设定为默认状态。除非特别说明，在上电和故障状态，70系列的输出模板默认所有输出为OFF状态。对于总线发送模板（BTM），它要查询系统中存在什么类型的远程机架。在BTM响应的基础上，CPU将这些机架及相关插槽列入待配置的插槽列表中。

最后，I/O扫描进行自身的初始化。它通过建立I/O控制器的每路I/O总线与I/O连接之间的关系来初始化所有I/O控制器，并从I/O控制器获得所有的I/O配置数据。获得的I/O配置数据与用户的I/O配置进行比较，之间的差异报I/O故障表。接着I/O扫描器将各I/O模块上的每个I/O控制器送往I/O总线进行配置。I/O控制器初始化完成后，I/O扫描器将各I/O模块中出厂默认值改为应用程序设定值。

对于70系列的输入模板，当信号改变状态时，模板可以设置中断，此中断是否发生取决于信号从高电平到低电平或从低电平到高电平。而对70系列的输出模板，它们的默认状态可由“OFF”更改到“保持最后状态”。

掉电时序

当电源供给系统检测到输入的交流电源失电超过一个电源周期，将会发生系统掉电。机架背板的一根信号线被动变为低电平来表示此状态，此时CPU运行中断。从信号发生时起，还有5ms以上时间来完成整个掉电过程。

失电失内存数据的保存

由于应用程序的RAM和BCP内存区域是由后备电池来维持的, 以下类型的数据在一个电源周期内保存:

- 应用程序。
- 故障表和其他诊断变量。
- 程序和程序块校验和。
- 强制数据。
- 寄存器数据(%R), 局部数据 (%L)和程序寄存器 (%P) 中的变量
- 模拟量存储区 (%AI和%AQ)中的数据
- 数字量输入(%I)状态
- 保持型数字量输出 (%Q)的状态
- 保持型数字量中间量 (%M)的状态。

下列类型数据不能在一个电源周期内保存:

- 数字量临时内存 (%T)状态
- 用于非保持线圈 -()-%M和 %Q
- 系统内部离散量 (系统状态位, 故障位, 保留位等)

第七节 时钟和定时器

系列90-70 PLC提供的时钟和定时器包括：一个计时时钟，一个日历时钟和软件、硬件看门狗定时器。有三类定时功能块：延时接通块，延时断开块和开始复位型延时定时器。计时接点接通和断开（按照方波形式）时间值单位可为0.01秒，0.1秒，1.0秒，1分钟。

计时时钟

计时时钟以100ms的“时间段”来跟踪CPU自上电以来所经历的时间。系统失电后该时钟不保持，每次上电它都重新开始计时。每隔一秒硬件中断一次以使计数器更新一次。该秒计数器大约要在时钟开始计数后的100年后才会溢出（计数值回零）。

由于计时时钟为系统软件操作及定时功能块提供了基准，它不可由用户程序或编程器复位。但是，应用程序可通过使用服务请求功能块#16来读取计时时钟当前值。

日历时钟

系列90-70 PLC中的日历时钟由硬件日历时钟来实现。它包括以下七种时间功能：

年 (两位数)
月
日
时
分
秒
星期几

日历时钟在失电时由后备电池维持其现有状态。然而，除非用户初始化此时钟，否则其中的值没有任何意义。应用程序可通过使用服务请求功能块#7来读取或设置该时钟。它也可在编程软件中读取或设定。

日历时钟可处理月到月和年到年的日期转换，它自动补偿到2038年之间的闰年日期。

看门狗定时器

软件看门狗定时器

90-70 PLC 软件看门狗定时器用于检测“未完成扫描”条件。该定时器值通过编程软件进行设置，它的允许范围为10到2550ms，默认值为200ms。软件看门狗总是在每扫描周期从零开始。

软件看门狗定时器用于检测应用程序非正常的运转，以避免PLC扫描周期超出用户定义的时间范围。应用程序的非正常运转状况包括：

程序块过多的递归调用

过多的循环 (大循环数或每次循环需要占用大量的时间)

死循环

当选择软件看门狗的值时，设定的总要比扫描周期所最大需要值还要大，以避免发生意外的中止。对于恒定时间扫描和微循环扫描模式，选择软件看门狗时间时需考虑超时条件。

如果看门狗超时，则OK LED指示灯闪烁，CPU转入Stop/Halt模式，但某些功能仍然有效。在PLC故障表报故障，所有输出恢复到默认状态。如果使用串口或WSI通讯（不是以太网），则CPU将只与编程器进行通讯，其他的通讯和操作都无效。要重新开始，必须将包含CPU的机架电源重新上电。

为延长当前扫描周期超过软件看门狗计时器的值，应用程序可使用服务请求功能#8来重新启动软件看门狗定时器。然而，软件看门狗定时值只能通过配置软件来更改。

硬件看门狗定时器

有一备用电路专门为PLC提供附加保护。如果此电路被激活，PLC立即进入RESET模式。输出恢复为默认状态，不能进行任何形式的通讯，CPU停止运行。要恢复必须重新上电。

第八节 系统安全性

90-70 PLC支持如下三种安全设置：

1. 密码/权限等级
2. OEM 保护
3. 按键开关写保护

密码/权限等级

密码是系列90-70 PLC的一个可配置特性。对它的使用是可选的，可以通过编程软件来对它进行设定。密码设置的目的是当编程器处于Online或Monitor模式时，为PLC提供不同等级的访问权限。当编程器处于Offline状态时，不能使用密码。使用密码可限制：

- 改变 I/O 和 PLC配置数据
- 更改程序
- 读取 PLC数据
- 读取程序
- 锁定程序块

CPU默认状态是没有密码保护的。PLC每个权限等级都有一个密码。每个密码可是单独设定，也可以设定为相同的。它可以是1到7位的ASCII字符。只有编程器可以改变密码。

PLC密码保护用于限制用户只能进行选定PLC的功能的操作。设定密码后，由任何通讯路径对PLC的访问都是被限制的，除非用户输入正确的密码。一旦密码被成功接受，将被准许当前权限基本即以下的所有操作（例如，若提供LEVEL 3的权限等级，则权限等级为0，1，2，3的操作都是允许的）。如果PLC通讯被延迟，保护等级将自动恢复到最高未保护等级或权限等级2的最低权限等级。

表 2-18. 权限等级

Priv Level	Password	Access Description
4	有	可对所有配置和逻辑进行写操作。配置仅可在停止模式下修改；逻辑可在停止和运行模式下修改。可设定或修改任何权限等级密码。
3	有	配置和逻辑仅可在停止模式下修改，包括逐字改变，添加或删除程序逻辑及支配离散I/O。
2	有	可对任何数据内存进行写操作，包括触发/强制变量值，但不包括支配离散I/O。可对PLC进行运行或停止操作。可清除PLC和I/O故障表。注意：如果没有设定密码，此为默认状态。
1	有	可对任何PLC数据进行读操作，包括从PLC读故障表，运行数据报，检查逻辑和配置，下载程序和配置等。但不能更改任何数据，该等级下也不允许通过编程器将PLC转入运行模式。
0	无	可读PLC当前状态（包括PLC所支持的各种特性），读源文件名（6.0版以前称CP名），改变权限等级以及作为编程器登录。该等级下也不允许通过编程器将PLC转入运行模式。

注意

用户必须注意CPU上的运行模式开关不受密码保护。即使编程器不能将PLC在RUN和STOP之间切换，CPU上的开关是可以的。

编程器保护等级请求

与CPU连接后，编程器请求CPU转移至最高的非保护级，从而不需要特别的等级请求就可给予编程器最高的非保护级。

编程器要请求一个权限等级，必须提供出新的权限等级及其密码。如果编程器输入的该等级密码同PLC中所保存的密码不符，则该权限等级的修改将被拒绝并报故障信息于PLC故障表。当前权限等级不会改变。要改变到一个没有密码保护的权限等级，只需提供新的等级和一个空的密码。这种改变可以是向低等级改变，也可以是向高等级改变。

禁止密码保护

密码保护是一项可选功能。如果用户希望能够不使用密码保护，可以通过编程软件来禁用密码保护。

Note

禁用密码保护后如果需要重新启用密码保护功能，PLC必须在移除电池状态下重新上电。

密码保护也会阻止用在CPM 914, 915, 924, 925及所有CPX模板上的FLASH内存的固件升级。在对这些型号的模板进行固件升级之前，要禁用密码保护，在升级完成后重新启用即可。

OEM 保护

OEM保护类似于密码和权限等级；然而，OEM保护提供一个更高等级的安全性。OEM保护特性使用1到7个字符来启用和禁用。当OEM保护使能时，对PLC程序和配置的所有读写操作被禁止。

对OEM厂商软件投资的保护由被称为OEM key的特殊形式的密码来完成。当OEM key被赋予非空值时，CPU可进入一种模式，这种模式下的读写逻辑和写配置都是被禁止的。这就允许一个第三方的OEM来为PLC CPU编制控制程序接着设定OEM锁定模式来阻止最终用户来读取和修改程序。

注意

OEM保护也会阻止用在CPM 914, 915, 924, 925及所有CPX模板上的FLASH内存的固件升级。在对这些型号的模板进行固件升级之前，要禁用密码保护，在升级完成后重新启用即可。

写保护按键开关

CPU 780, 781, 782, 788, 789, 790, 914, 924, 915, 925; CGR 772 和935; 以及所有CPX型号90-70 CPU模板都有一个内存写保护按键开关。此按键开关位于前面板的上部，上面板机架夹子的上方。当在保护位置时，PLC程序和配置不能被修改或删除。

注意

写保护按键开关在写保护位置时，也会阻止用在CPM 914, 915, 924, 925及所有CPX模板上的FLASH内存的固件升级。在对这些型号的模板进行固件升级之前，要首先将该写保护开关拨至“写允许”位置。

第九节 系列 90-70 PLC I/O 系统

系列90-70 PLC I/O 系统提供了PLC与用户提供设备与装置之间的接口。此I/O系统支持机架型的70 I/O, Genius I/O系统及FIP I/O系统。Genius I/O总线控制器（GBC）模板提供了系列90-70 PLC CPU与Genius I/O总线之间的接口。FIP I/O总线控制器（FBC）模板提供了系列90-70 PLC CPU与FIP I/O总线之间的接口。另外还支持这三个I/O子系统，I/O系统也支持以太网接口和PCM模板。

系列90-70 PLC I/O 系统结构如下图所示：

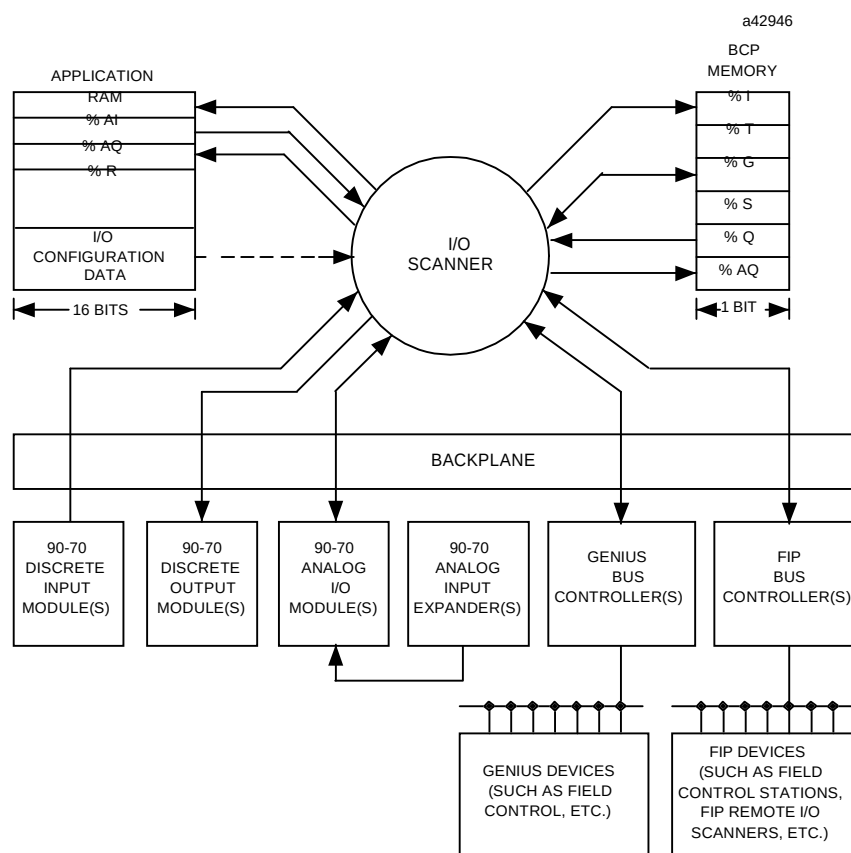


图 2-13. 系列 90-70 PLC I/O 系统结构

I/O 数据映射

离散量输入和输出按位存储在CPU的BCP位缓存中。模拟量I/O存储在专门分配的应用程序RAM区中。模拟数据总是以多路分离状态存储，而每个通道都需要一个字（16位）。

默认情况

上电时，70型离散量输入模块对该模块第一路输入默认不进行中断操作，并且输入滤波以低速进行。配置窗口提供了指定第一路输入可为一个中断输入及输入滤波可以以高速或低速进行的能力。如果用户要改变它，新缺省值将在上电过程中由CPU配置该模板，否则该模板配置不被通过。70型离散量输出模板默认状态为OFF。配置窗口提供了CPU在Run/Enabled 到 Run/Disabled 或Stop模式相互转换的能力。系统中止时也使用这些默认信息。

Genius I/O

以下各部分介绍了系列90-70 PLC系统中Genius I/O的使用。关于Genius I/O块类型，配置和设定的内容，见《Genius I/O系统用户手册（GEK-90486-1 和 -2）》。

Genius I/O 总线配置

系列90-70 PLC中使用的总线控制器控制一个单独的Genius I/O总线。任何类型的Genius I/O块都可加到该总线上。

在I/O故障表中，列出了故障的机架号，槽号，总线，模板号及I/O点数。1号总线对应单通道Genius总线控制器。

Genius I/O 数据映射

Genius I/O的离散量输入输出按位存储在CPU的位缓存中。Genius I/O的模拟量数据存储在专门为其分配的应用程序RAM中（%AI 和 %AQ）。模拟量数据总是按照一个字来存储一个通道的数据进行的。

模拟量分组模块只占用（在输入和输出数据内存区）输入输出实际需要的内存空间。例如，Genius I/O 115VAC 分组模块IC660CBA100有四个输入和两个输出；它就占用四个字的模拟量输入内存(%AI)和两个字的模拟量输出内存(%AQ)。

离散量分组模块（每点可用手持式监视器（HHM）来配置的输入，输出或带反馈的输出）需要占用的离散输入内存(%I)和离散输出内存(%Q)等于它的物理空间。因此，8 I/O 115 VAC离散量分组模块(IC660CBD100)就需要8位的%I内存和8位的%Q内存，不管该模块是如何配置的。

下面四种 Genius I/O模块需要分配模拟量内存:

- 6通道模拟量分组模块
- 6通道热电偶模块
- 6通道 RTD模块
- 4通道 Gauge/mV 模拟量输入模块

其中热电偶和RTD模块也称作低等模拟量输入模块。

模拟量分组模块

模拟量分组模块包含四个通道模拟量输入和两通道模拟量输出。当一个模块数据要进入Genius I/O总线时, 它以一个广播控制信息将所有四个通道的输入数据以广播的形式发出。然后, 当总线控制器工作时, 它以一个直接的控制信息将两个通道的输出数据发送出去。

低等模拟量模块

与模拟量分组模块不同, 低等模拟量模块仅是一个输入模块, 每块有六个通道。

默认状态

Genius I/O模块有许多缺省值, 它们可通过使用Genius I/O HHM来进行设置。这些缺省项包括:

- 故障报告
- 范围选择
- 模拟量输入输出域
- 输入滤波次数
- 报警输入模式
- 输出保持最后状态
- 输出缺省值

这些缺省值存储于模块自身上的EEPROM中。系列90-70 PLC的配置程序仅允许修改这些缺省值中的一个小子集。有关更多信息, 参见《Genius I/O系统用户手册 (GEK-90486-1 -2)》。

使用COMMREQ功能块, 应用程序能够请求总线控制器改变指定块的默认状态。但是, 这种改变仅当模块不在Config Protect模式时才有效。在Config Protect模式下, 只有使用HHM才能更改这些默认值。有关Genius I/O的COMMREQ功能块的使用格式见《Genius 总线控制器用户手册 (GFK-0398)》。

Genius 全局数据通信

系列90-70 PLC支持通过公用一个Genius I/O总线在多PLC系统中共享数据。这为自动地重复传输%G, %I, %Q, %AI, %AQ和 %R数据提供了一种手段。使用全局数据时不需要特别的应用程序, 它集成于I/O扫描中。所有与Genius I/O兼容的GE Fanuc PLC都可向90-70PLC发送数据或从90-70 PLC获取数据。编程软件用于配置接收和发送Genius I/O总线上的全局数据。

注意

当90-70 PLC 处于STOP/NOIO模式时, Genius全局数据的通讯不再进行。然而, 在STOP/IOSCAN模式时, Genius全局数据的通讯仍在继续进行。

FIP I/O

在90-70 PLC系统中使用FIP I/O的相关信息见后面段落。关于FIP I/O的特殊信息, 如类型, 配置和设定等, 参见《系列90-70 FIP总线控制器用户手册(GFK-1038)》。

FIP I/O 总线的配置

90-70 PLC 使用FIP总线控制器来控制一个单独的FIP I/O总线。目前支持的有90-30 FIP远程I/O扫描器, FIP总线接口单元, 普通FIP I/O模板配置。所有的FIP I/O接口模板(例如, FIP远程I/O扫描器)必须向90-70 FIP总线控制器提供输入数据, 以使FIP总线控制器在下一个正常输入扫描中具有与提供给90-70 CPU相同的数据。类似地, 当90-70 CPU运行下一个输出扫描时, FIP总线控制器接收此输出数据, 并传递给相应的FIP I/O接口模板, 然后再刷新本地I/O。

90-30 FIP远程 I/O 扫描器

FIP远程 I/O 扫描器提供了一种在FIP I/O网络中使用90-30 I/O作为远程I/O站的功能。FIP远程 I/O 扫描器模板提供了与FIP I/O网络之间的通讯接口(与90-70 FIP总线控制器之间的通讯), 同时也提供了对本地90-30 I/O模板的扫描功能。(更多关于90-30 FIP远程 I/O 扫描器的信息, 请参考《90-30 FIP远程 I/O 扫描器用户手册GFK-1037》)。

FIP 总线接口单元(现场控制)

FIP 总线接口单元提供了一种在FIP I/O网络中使用现场控制 I/O作为远程I/O站的功能。FIP总线接口单元模板提供了与FIP I/O网络之间的通讯接口(与90-70 FIP总线控制器之间的通讯), 同时也提供了对本地现场控制模板的扫描功能。(更多关于FIP 总线接口单元的信息, 请参考《FIP 总线接口单元用户手册GFK-1175》)。

一般FIP I/O

一般FIP I/O允许配置FIP远程 I/O 扫描器和FIP 总线接口单元之外的其他FIP I/O。这就允许90-70 CPU和90-70 FIP总线控制器为一般FIP I/O设备分配I/O参考地址。配置选择也允许90-70 FIP总线控制器在FIP I/O网络中确认一般FIP I/O模块。

FIP I/O 故障数据

在I/O故障表中，FIP远程 I/O 扫描器和FIP 总线接口单元控制的远程节点的故障信息包括：机架号，槽号，FIP ID，远程机架号以及远程槽号。一般FIP I/O不能返回故障信息。

FIP I/O 数据映射

FIP I/O离散量输入和输出按位存储于CPU的位缓存中， FIP I/O的模拟量数据存储在专门为其分配的应用程序RAM中（%AI 和 %AQ）。模拟量数据总是按照一个字来存储一个通道的数据进行的。

模拟量分组模块只占用（在输入和输出数据内存区）输入输出实际需要的内存空间。例如，有四个输入和两个输出的模块；它就占用四个字的模拟量输入内存(%AI)和两个字的模拟量输出内存(%AQ)。

默认状态

FIP I/O模块有许多缺省值，它们可通过使用编程软件来进行设置。这些缺省项包括：

范围选择

模拟量输入比例

模拟量输出比例

离散量输出默认状态Off (固定)

模拟量输出默认状态保持最后状态(固定)

诊断数据收集

90-70 PLC I/O系统的诊断数据通过以下两种方式之一获得：

1. 如果I/O模板有相关的总线控制器（Genius或FIP总线控制器），则由总线控制器向CPU提供诊断数据。
2. 如果I/O模板是70型I/O模板，则由CPU I/O扫描子系统产生基于I/O模板数据的诊断位。

诊断位来自I/O模板给它们I/O控制器（Genius或FIP总线控制器或90-70 CPU）的诊断数据。诊断位总是体现与当前故障状态相关的模板。当故障发生时诊断位置位，故障清除时复位。

一般来讲，第三方I/O模板的诊断数据不通过90-70 PLC获得。这些板子的诊断信息可由应用程序使用VME读/写功能块来得到。从90-70 5.50版CPU开始，90-70系统支持经GE Fanuc授权允许的第三方I/O模板的开发。这些板子被配置为“3rd PartyVME”，接口形式为“I/O扫描”。开发确认的I/O扫描接口板也能向90-70 CPU提供离散量和模拟量的诊断信息。

离散量 I/O 的诊断信息

每个离散量I/O点的诊断信息由90-70 PLC保持。在应用程序RAM中专门为离散量诊断数据分配两个内存块。一个与%I内存相关，另一与%Q内存相关。诊断内存中的位与I/O点相对应。该位指示了相关I/O数据的有效性。每个离散量点都有一个故障变量，它们可以使用两个专用接点来查询：故障接点(-[FAULT]-)和无故障接点(-[NOFLT]-)。可通过编程软件使PLC仅收集错误数据。下表显示了故障和无故障接点的各种状态。

Condition	[FAULT]	[NOFLT]
Fault Present	ON	OFF
Fault Absent	OFF	ON

模拟量 I/O 诊断数据

PLC CPU对应每个模拟量通道都有诊断信息，这些通道对应于70模拟量输入模板，70模拟量输出模板，Genius模拟量模板等等。CPU为模拟量诊断数据分配两个内存块。一个与%AI内存相关，另一与%AQ内存相关。每个I/O通道占用一字节的诊断内存。因此每个模拟量I/O通道使用两个字节的%AI 和%AQ内存，诊断数据内存是数据内存大小的一半。

模拟量诊断数据包含了诊断数据和带高报警和低报警位的过程数据。诊断数据为-[FAULT]-和-[NOFLT]-接点数据。而过程数据为-[HIALR]- 和-[LOALR]-接点数据。为模拟量诊断数据分配的内存是每个模拟量输入或输出的一个字对应一个字节。当模拟量故障接点在应用程序中给定时，PLC对诊断字节中除去过程位的所有位进行“或”运算。如果任一个诊断位为ON则报警接点ON；仅当所有位为OFF时，报警接点为OFF。

Chapter 3

故障的解释与排除

本章将帮助用户发现并排除90-70 PLC系统故障。它将解释出现在PLC故障表中的故障描述，以及出现在I/O故障表中的故障分类。

本章中，每个故障解释都给出了PLC故障表中的故障描述或I/O故障表中的故障分类。找到与编程器屏幕上显示的应用程序故障条目相同的故障描述或故障种类。在它的下面是引起该故障可能的原因及纠正的方法。

第三章包括以下各部分：

Section	Title	Description	Page
1	系统故障处理	介绍 PLC 系统故障 (SY_FLT) 和 I/O故障(IO_FLT)。包括配置故障，更改故障响应，非配置故障及定位故障变量（机架，槽号，总线及FIP定位变量）。	3-2
2	故障处理	介绍了系列90-70 PLC 中可能出现的故障类型及其如何在故障表中显示。也包括了PLC和I/O故障表显示的说明。	3-10
3	PLC故障表 解释	提供每个PLC故障的说明及纠正故障的方法。	3-16
4	I/O 故障表 解释	提供每个I/O故障的说明及纠正故障的方法。	3-38

第一节 系统故障处理 (General)

下列系统故障变量可用于区分所产生的故障的类型。

系统故障变量	说明
ANY_FLT	系统任何故障。
SY_FLT	90系列PLC 任何系统故障。
IO_FLT	I/O 故障
SY_PREP	PLC 故障表中一个新条目。
IO_PREP	I/O 故障表中一个新条目。
HRD_FLT	硬件故障。
SFT_FLT	软件故障。

系统上电时，要复位全部的系统故障变量。如果发生故障，在故障产生后的下一个扫描周期受影响的on转换状态变量置位。如果故障一直存在，系统故障变量也会一直处于置位状态，直到PLC被清空或程序被清空。故障变量ANY_FLT在发生任何故障时都会置位。SY_PREP和IO_PREP在PLC和I/O故障表中包含故障信息时置位。

系统故障变量

当系统故障变量置位时，其附加的故障变量也同时置位。下表给出了各种其他的错误类型。标注星号的为可配置的系统故障变量。

表 3-1. 系统故障变量

故障类型	PLC系统故障 (SY_FLT)	I/O故障 (IO_FLT)
硬件故障 (HRD_FLT)	SBUS_ER 系统总线故障 HRD_CPU PLC CPU 硬件故障 HRD_SIO模板硬件故障 SBUS_FL 系统总线失败*	
软件故障 (SFT_FLT)	SFT_SIO 智能模板故障 SFT_CPU PLC 软件故障* MAX_IOC总线控制器过多* STOR_ER 编程器下装失败 *	SFT_IOC I/O 控制器软件故障
其他故障	PB_SUM 程序块检测失败 LOW_BAT 电池低电量 OV_SWP 恒定时间扫描超时 SY_FULL PLC故障表满 IO_FULL I/O故障表满 APL_FLT 应用程序错误 NO_PROG 上电后无应用程序* BAD_RAM 程序内存冲突* WIND_ER 未完成窗口服务 * BAD_PWD 口令错误 * NUL_CFG 无配置 * LOS_SIO 丢失可选模块* ADD_RCK多余的扩展机架* ADD_SIO 多余可选模块* CFG_MM 配置不匹配 * LOS_RCK丢失机架 *	LOS_IOC 丢失I/O控制器 LOS_IOM 丢失I/O模板 ADD_IOC 多余I/O 控制器 ADD_IOM 多余I/O模板 IOC_FLT 总线或I/O控制器错误 IOM_FLT I/O 模板错误

* 可配置系统故障变量。.

可配置的故障响应

有些故障一旦发生，PLC必须停止运行；而对于有些故障的响应要取决于应用程序的属性。所有的故障都将会发生如下三种响应之一。

Fault Action	Description
致命故障	此类故障停止PLC运行，诊断变量置位，报故障信息于故障表。
诊断故障	此类故障不停止PLC运行，但诊断变量置位，报故障信息于故障表。
信息类故障	此类故障报故障信息于故障表，但不引起其他响应。

有些故障称为“不可配置”故障，它们的故障响应不能更改。另外一些称为“可配置”故障，如果更改适合于应用程序则这些故障的响应类型可以更改。下表所列为可配置故障：

表 3-2. 可配置故障的故障变量

Fault (Default Action)	Description	May Also Be Set
SBUS_ER (诊断)	系统总线错误。 (BSERR*信号由VME系统总线产生。)	HRD_FLT SY_PRE, SY_FLT 当BSERR*出现时，其他变量也可能由访问类型决定。
SFT_IOC* (诊断)	I/O控制器中的不可恢复软件错误。	IO_FLT, IO_PRE SFT_FLT
LOS_RCK** (诊断)	缺少机架 (BRM失败，掉电)，或丢失配置的机架。	SY_FLT, SY_PRE IO_FLT,
LOS_IOC*** (诊断)	丢失 I/O 控制器通道或丢失配置的总线控制器。	IO_PRE IO_FLT, IO_PRE
LOS_IOM (诊断)	丢失 I/O 模板 (无响应)或丢失已配置的I/O模板。	IO_FLT, IO_PRE
LOS_SIO (诊断)	丢失智能模块(无响应)或丢失已配置的模板。	SY_FLT, SY_PRE
IOC_FLT (诊断)	非致命的总线或 I/O控制器错误，10秒中大于10次总线故障 (错误率可配置)。	IO_FLT, IO_PRE
CFG_MM (致命)	再上电或运行期间发现错误的模板类型。PLC不检查单个板的配置参数设置，如Genius I/O模板。	SY_FLT, SY_PRE

* SFT-IOC 软件故障将与设定的LOS_IOC有相同的故障响应。

** 当报告 “ Loss of Rack or Addition of Rack fault ” 故障信息时，通常单独的丢失或增加模板在该机架中不再产生。

*** 即使LOS-IOC故障被配置为致命故障， PLC也不会进入STOP/FAULT状态除非同时发生GENIUS总线控制器内部错误。

注意

若登录于故障表的故障响应是信息类的，则不能用于配置响应。例如，SBUS_ERR登录的故障是信息类的，但用户配置为致命故障，则该故障仍按照信息类故障响应。

不可配置故障

对于不可配置故障，故障响应不可更改。下表所列为不可配置故障：

表 3-3. 不可配置故障

Fault	Description	Result
SBUS_FL (致命)	系统总线故障。PLC CPU 不能访问VME总线。 BUSGRT*NMI 错误。	置位SY_FLT, HRD_FLT和SY_PRES.
HRD_CPU (致命)	PLC CPU硬件故障，如内存设备错误或串口错误。	SY_FLT, SY_PRES HRD_FLT
HRD_SIO (诊断)	系统中模板的非致命硬件故障，如PCM模板上的串口错误。	SY_FLT, SY_PRES HRD_FLT
SFT_SIO (诊断)	PCM或LAN接口模板的不可恢复软件错误。	SY_FLT, SY_PRES SFT_FLT
PB_SUM (致命)	在上电或运行模式期间的程序检测失败。	SY_FLT, SY_PRES
LOW_BAT (诊断)	系统中CPU或其他模板的低电量信号。	SY_FLT, SY_PRES
OV_SWP (诊断)	定时间扫描超时。	SY_FLT, SY_PRES
SY_FULL IO_FULL (诊断)	PLC故障表满(16条)。 I/O 故障表满 (32条)。	SY_FLT, SY_PRES IO_FLT, IO_PRES
APL_FLT (诊断)	应用程序错误。	SY_FLT, SY_PRES
ADD_RCK** (诊断)	增加一个新的机架或先前的故障机架恢复。	SY_FLT, SY_PRES
ADD_IOC (诊断)	先前的故障 I/O控制器不再报故障。	IO_FLT, IO_PRES
ADD_IOM (诊断)	先前的故障 I/O模板不再报故障。	IO_FLT, IO_PRES
ADD_SIO (诊断)	增加一个新的模板或先前的故障模板恢复。	SY_FLT, SY_PRES
IOM_FLT (诊断)	I/O模板上的点或通道故障，模板的部分故障。	IO_FLT, IO_PRES

表 3-3. 不可配置故障 - 续表

Fault	Description	Result
NO_PROG (信息)	上电时无应用程序。 仅发生在PLC第一次扫描或者由电池维持的RAM中的程序失败。	不会置位任何故障变量。PLC将不会进入运行模式，他将继续停止模式下的扫描，除非有程序下装。这可以是一个空程序。置位 SY_FLT 和 SY_PRES。
BAD_RAM (致命)	上电时，程序内存冲突。程序不能读或不能通过测试。	Sets SY_FLT and SY_PRES.
WIND_ER (信息)	窗口完成错误。编程器服务或逻辑窗口被跳过，在定时间扫描和微循环扫描方式下时。	Sets SY_FLT and SY_PRES.
BAD_PWD (信息)	口令错误，对优先级修改请求被拒绝。	Sets SY_FLT and SY_PRES.
NUL_CFG (致命)	转入运行模式时无配置。Running without a configuration is equivalent to suspending the I/O scans.	Sets SY_FLT and SY_PRES.
SFT_CPU (致命)	CPU software fault. A non-recoverable error has been detected in the CPU. May be caused by Watchdog Timer expiring.	PLC immediately transitions to Error Sweep mode. The only activity permitted is communication with the programmer. To be cleared, PLC power must be cycled. Sets SY_FLT, SY_PRES, and SFT_FLT. Sets SY_FLT, SY_PRES, and SFT_FLT.
MAX_IOC (致命)	The maximum number of bus controllers has been exceeded. The Series 90 PLC supports 32 bus controllers.	PLC will not transition to Run mode.
STOR_ER (致命)	Download of data to PLC from the programmer failed; some data in PLC may be corrupted.	This fault is not cleared at power-up, intervention is required to correct it. Sets SY_FLT and SY_PRES.

**When a Loss of Rack or Addition of Rack fault is logged, individual loss or add faults for each module in that rack are usually not generated.

故障接点

故障 (-[FAULT]-) 和无故障 (-[NOFLT]-)接点可用于查找系统中存在的各种故障。这些接点不能被支配。下表列出了故障接点 and 无故障接点的各种状态。

Condition	[FAULT]	[NOFLT]
有故障	ON	OFF
无故障	OFF	ON

故障定位变量(机架，槽，总线，模板)

如果进行过硬件配置的话，系列90-70 PLC支持为每机架，槽，总线，模板保留故障名。通过在FAULT 和 NOFLT接点指令上编辑这些名称，逻辑可以定位与所配置的机架和模板相关的故障。

故障变量的格式

表 3-4. 故障变量名称

Fault Reference Type	Reserved Name	Comment
机架	RACK_0r	r 为机架号 0 到7。
槽	SLOT_rs	r 为机架号 0到 7， s为槽号 0 到 9。
总线	BUS_rsb (仅用于Genius)	r 为机架号 0到 7， s为槽号 0 到 9， b 为 Genius 总线号1或2。
模板	M_rsbmm (仅用于Genius)	r 为机架号 0到 7， s为槽号 0 到 9， b 为 Genius 总线号1或2， mm 为 串行总线地址(SBA)数 00 到 31。
FIP 模板	F_rsmmm (仅用于FIP)	r 为机架号 0到 7， s为槽号 0 到 9， mmm 为 FIP 站地址 000 到 255。

这些错误名在程序中仅可用于FAULT 和 NOFLT接点。保留故障名总是存在的。启用一些特殊的功能选项是必要的，如故障点。

这些故障名不同于%SA, %SB, %SC，或任何其他变量类型。仅显示其名称，故障变量是不能在变量表屏幕中显示的。

一个机架的故障名的书写格式为RACK_0r，这里机架号为0到7的数。例如，下例中的RACK_01表示机架1。

一个槽的故障名的书写格式为SLOT_rs，这里r为0到7的机架号，s为0到9的槽号。例如，下例中SLOT_15表示机架1， 5槽。

```
|
| RACK_01 SLOT_15                                %Q00002
|-[FAULT]-[NOFLT]-----<-----< )-
```

一个总线的故障名的书写格式为BUS_rsb，这里r为0到7的机架号，s为0到9的槽号，b为总线号1或2。例如，BUS_241表示机架2， 4槽，总线1。

一个模板的故障名的书写格式为M_rsbmm，这里r为0到7的机架号，s为0到9的槽号，b为总线号1或2，mm为模板号00到31。例如，M_26128表示机架2， 6槽，总线1，模板28。

一个FIP模板的故障名的书写格式为F_rsmmm，这里r为0到7的机架号，s为0到9的槽号，mmm为FIP站地址。

Behavior of Fault References

上电时，PLC中所有故障定位变量被复位。当一个故障登录时，PLC改变受影响的变量的状态。故障变量将保持此状态，直到有以下操作之一发生时：

PLC故障表和I/O故障表都通过编程软件清除，或者分别清除两个故障表，或者完全清除PLC内存。

相关设备（机架，I/O模板或Genius设备）重新添加到系统中。无论何时如果发生“Addition of . . .”故障，PLC将初始化与此设备相关的所有故障变量到NoFlt状态。这些变量将保持NoFlt状态直到与此设备相关的其他错误报告产生。（这可能需要几秒的时间来重新划分I/O故障，尤其当总线控制器重启时。

对于 FIP 故障，参见《系列90-70 FIP 总线控制器用户手册(GFK-1038)》。

注意

这些故障变量只是为了信息提示的目的而置位的。它们不能用于限定I/O数据。I/O点故障数据（见3-9页）可用于限定I/O数据。作为设定故障定位变量的故障状态PLC不停止运行。

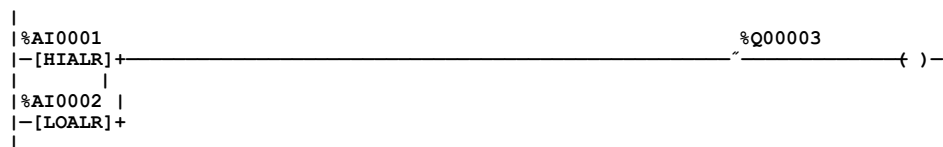
故障变量具有级联效应。如果在位于机架5，槽6，总线1，模板29的模板发生故障，则以下故障变量都被置位：RACK_05, SLOT_56, BUS_561, M_56129。在故障表中只会登录一条故障信息来标记此模板故障。在这种情况下，故障表并不指示附属于机架，槽和总线的故障。在FIP设备的机架5，槽6，模板29产生故障将置位如下故障变量：RACK_05, SLOT_56, F_56029。

FIP远程I/O扫描设备的模板故障将只引起一个模板错误类型的故障变量置位。例如，若站号为43，位于90-70机架3，9槽的FIP模板的模板为FIP远程I/O扫描器，在FIP远程I/O扫描器的机架1，槽5有一块故障模板，则将仅有故障变量F_39043置位。

报警接点

高报警 (-[HIALR]-) 和低报警 (-[LOALR]-) 接点用来表示模拟量输入模板的比较器功能状态。然而，要使用故障点首先要在硬件配置中启用该功能。关于启用点故障变量的详细内容，参见《Logicmaster用户手册(GFK-0263)》或者CIMPPLICITY中硬件配置的在线帮助。

下例中使用了高报警和低报警接点。



注意

HIALR 和 LOALR 接点不会在故障表中产生条目信息。

点故障

点故障适合于外部I/O故障，虽然它们总是由与高等级的内部硬件相关的失败引起的（例如，IOC失败或机架丢失）。为了使用点故障，它们必须在硬件配置中启用。关于启用点故障变量的更多内容，见《Logicmaster用户手册(GFK-0263)》或者CIMPPLICITY中硬件配置的在线帮助。

当启用时，每个离散I/O点故障的布尔变量和每个模拟量I/O通道级错误的字节变量都会在PLC内存中分配空间。用于点故障的PLC内存包括整个变量表的内存大小。以上介绍的FAULT和NOFLT接点用于访问点故障。

第二节 故障处理

当影响系统操作和运行的某些故障或状态发生时，系列90-70 PLC系统就会发生故障。这些情况，比如丢失I/O模块或丢失机架，可能会影响PLC控制某机器或某过程的能力。同时，这些状态也可能会有有利的影响，比如当一块新模板进入在线状态并正好利用它。或者，这些状态仅作为一种报警，如电池电量低信号表示保护PLC内存的电池需要更换了。

关于系统状态/故障参考的详细内容，见第二章，第三节“程序组织”。

报警处理器

上述情况或错误本身称为故障。当由CPU接收并处理一个故障时，称为一个报警。CPU中处理这些情况的软件称为报警处理器。它通过编程软件与用户进行交互。实际应用中，所检测到任何一个故障将被记入故障表，并在PLC故障表屏幕或I/O故障表屏幕上显示。

故障分类

系列90-70 PLC检测几类故障。它们是内部故障，外部故障和操作故障。

表3-5. 故障分类

Fault Class	Examples
内部故障	模板无响应。 电池电量不足。 内存检测错误。
外部 I/O 故障	丢失机架或模板。 丢失Genius I/O 块。
操作故障	通讯失败。 配置失败。 口令输入错误。

系统对故障的反应

一般来说，硬件故障不是需要系统停机就是系统可以承受此种故障。虽然I/O故障是可以由PLC系统承受的，但可通过应用程序或受控过程将它配置为不可承受故障。操作故障一般是可承受的。系列90-70 PLC的故障按不同的属性有如下三种分类方法：

表 3-6. 故障属性

属性	说明
故障表影响范围	I/O 故障表
故障的作用类	PLC 故障表
	致命的
	诊断的
	提示类的
故障的反应	可配置的
	不可配置的

故障表

设置两个故障表—PLC故障表和I/O故障表—使得故障更容易查找。

故障作用类

致命故障将在对应故障表中记录故障信息，置位所有的诊断变量，并停止PLC的运行。诊断故障将在对应故障表中记录故障信息，置位所有的诊断变量。信息类故障只在对应故障表中记录故障信息。

表 3-7. 故障的作用类

Fault Action	Response by CPU
致命的	故障表记录，置位诊断变量，停止PLC的运行。
诊断的	故障表记录，置位诊断变量。
信息类的	故障表记录。

可通过编程软件更改某些故障的作用类。此类应用中有两种分类：致命的和非致命的。这相当于PLC故障作用类中的致命故障和诊断故障。只有致命故障会引起系统中止。另外，信息类故障只把故障记入故障表。

当CPU检测到一个故障时，它将使用默认的故障作用类进行处理。对于可通过软件更改的故障作用类，CPU将按照软件配置的进行处理，这可以是系统默认值，也可以是用户自定义的。

故障响应

故障响应指一个故障是否具有改变其故障作用类的能力。可以更改的就称为可配置故障，不可更改的就称为不可配置故障。不可配置故障都是致命的或信息类的故障。同样，不可配置故障也不能引起应用程序的变量置位，也不能产生与故障检测相关的报警块。某些不可配置故障也有其他与之相关的影响。一般来讲，这些影响将会改变CPU运行模式(Stop, Run/Disabled, Run/Enabled)。比如当系统检测到空系统配置时，这种影响就是禁用I/O。

PLC故障表

PLC故障表用于显示PLC故障，如口令错误，PLC配置不符，奇偶校验错误，通讯错误等。

编程器可以在任何操作模式。然而，在Offline模式下是不能显示故障表的。在Online 或Monitor模式下，才可显示PLC故障数据。在Online模式下，才可清除故障表（此操作是受密码保护的）。

Field	Description
显示的顶端故障	此处显示PLC当前故障的索引号，它将显示在屏幕的第一行。
总故障	自上次清除故障表以来的故障总数。
上次清除的表	记录上次从故障表中清除故障的日期和时间，此信息由PLC维护。
溢出登记	故障表溢出而丢失故障信息引起的故障条目总数的丢失。PLC故障表最多容纳40条故障记录（6.0版以前为16— 在CIMPPLICITY 软件中可配置1）。
PLC 时间/日期	当前日期和时间。也由PLC来维护。

注意

PLC故障表的容量大小是可配置的（在CIMPPLICITY控制软件中默认值为16，最大值为40）。多余的故障（超过了所配置的数量限制）将引起故障表的溢出，会使故障信息丢失。故障表满时会置位系统变量SY_FULL (%S0009)。

用户定义故障

用户定义故障也可登记于PLC故障表。当用户定义故障发生时，它作为“应用信息（错误代码）”记录于相应故障表，其后可以有由用户输入的24个字符的说明信息，该信息必须以空字符结束，如0，且此空字符不记入24个字符数量中。若所包含的信息超过24字符，则只能显示前24个字符。

某些用户定义故障可置位一些系统状态变量(%SA0081–%SA0112)。

注意

需要使用服务请求21来定义用户定义故障（见第四章）。

I/O 故障表

I/O故障表用于显示I/O故障，如线路故障，地址冲突，强制回路及I/O总线故障。

编程器可以在任何操作模式。然而，在Offline模式下是不能显示故障表的。在Online 或Monitor模式下，才可显示PLC故障数据。在Online模式下，才可清除故障表（此操作是受密码保护的）。

Field	Description
显示的顶端故障	此处显示PLC当前故障的索引号，它将显示在屏幕的第一行。
总故障	自上次清除故障表以来的故障总数。
故障说明	关于当前I/O故障表中高亮显示的故障类型的详细说明。
上次清除的表	记录上次从故障表中清除故障的日期和时间，此信息由PLC维护。
溢出登记	故障表溢出而丢失故障信息引起的故障条目总数的丢失。I/O故障表最多容纳32条故障记录。
PLC 时间/日期	当前日期和时间。也由PLC来维护。。

注意

I/O故障表的容量大小是可配置的（在SIMPLICITY控制软件中默认值为32，最大值为40）。多余的故障（超过了所配置的数量限制）将引起故障表的溢出，会使故障信息丢失。故障表满时会置位系统变量IO_FULL (%S0010)。

读取附加的故障信息

由编程软件显示的故障表包含与故障相关的基本信息。对于附属于每个故障的附加信息，在编程软件中双击该故障将会显示详细的说明窗口。

每个故障信息的最后一项，故障纠正，列出了纠正该故障的方法。注意某些故障纠正的方法包含以下语句：

Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

这第二句是说，你必须告知现场服务你直接从故障表中读到的信息和十六进制信息。这样现场服务人员才将给你进一步排除该故障的方法。

第三节 PLC 故障表解释

每个故障解释包括了该故障的说明及其排除方法。许多故障说明有多种可能的触发原因。这种情况下，对于相同的故障说明，可通过故障代码和附加故障信息来区分不同的故障状态。如下例所示，故障代码是第五组数字中的前两位十六进制数。

01	000000	01030100	0902	0200	000000000000
					故障代码 (第五组数字中的前两位)

某些故障可能是由PLC CPU板或扩展内存板上的随即存取内存错误引起的。由于系统掉电或后备电池电压过低也会引起同样的故障。当损坏的内存引起此类故障时，为了避免过多的复杂指令，排除该故障的方法通常简单的用下述语句来表示：

Perform the corrections for Corrupted Memory.

其含义是：

- 1. 若系统掉电，需要重新安装电池。电池电压可能不足以维持内存中的数据。
- 2. 重新安装扩展内存板，内存板上的集成电路可能已损坏。
- 3. 更换PLC CPU板，CPU板上的集成电路可能已损坏。

注意

关于故障组的值的信息，见附录B。

可配置故障

可配置故障能够更改它们的故障作用类（致命的或诊断的）。CPU可由配置软件来设定故障作用类，可以是默认的，也可以是由用户选择的故障作用类。本节中，可配置故障中所列的故障类都是默认状态下的配置。

丢失或缺少机架

当系统不能与扩展机架通讯时就会产生丢失或缺少机架的故障，引起该故障可能的原因有：主机架BTM（总线发送模板）故障，扩展机架BRM（总线接收模板）故障，扩展机架电源故障，扩展机架上电后没有响应配置文件中的配置信息。该组故障默认的故障作用类为诊断故障。

错误代码:	1
名称:	机架丢失
说明 :	当 PLC 主机架不能与扩展机架通讯时产生该故障。系统中的每个扩展机架都可能引起该故障。
排除方法 :	(1) 将系统断电，确认BTM和BRM在各自机架中插接良好，所有的电缆都正确连接并接触良好。 (2) 更换电缆。 (3) 更换BRM。 (4) 更换BTM。
错误代码:	2
名称:	机架无响应
说明:	当来自编程器的配置文件指示某个扩展机架应存在于系统中，但该机架无响应时就会产生该错误。
排除方法 :	(1) 检查电源模板后的机架跳线，先检查报错机架，再检查所有机架，看是否有重复的机架号。 (2) 若不需此机架，则直接更新配置文件。 (3) 若该机架已存在，则需在硬件配置中添加该机架。 (4) 系统断电，确认BTM和BRM在各自机架中插接良好，所有的电缆都正确连接并接触良好。 (5) 更换电缆。 (6) 更换BRM。 (7) 更换BTM。 (8) 检查最末机架 BRM上的终端插头。

可选模块丢失或缺少

当Genet, PCM, BTM或BRM响应失败时,会发生可选模块丢失或缺少故障组。如果缺少模板或操作过程中该模板无响应就会出现此类故障。该故障缺省配置为诊断故障。

故障代码: 名称: 说明: 排除方法:	3 发现扩展机架的总线发送模板 当扩展机架中发现有总线发送模板时PLC产生此故障。 系统断电,移除扩展机架BTM。
故障代码: 名称: 说明: 排除方法:	16 模拟扩展模板置于基准转换模板的左侧 在机架中,模拟扩展模板置于基准转换模板的左侧。 系统断电,将模拟扩展模板移至基准转换模板的右侧。
故障代码: 名称: 说明: 排除方法:	19 丢失模拟扩展模块 基准转换模板同模拟扩展模板之间不能通讯。 (1) 确认基准转换模板同模拟扩展模板之间的连接。 (2) 更换模拟扩展模板 (3) 如果与所有的模拟扩展模板都不能通讯,则要更换基准转换模板。
故障代码: 名称: 说明: 排除方法:	2C, 2D 可选模块软启动失败 软启动之后PLC CPU不能与可选模块重新建立通讯。 (1) 尝试再次软启动。 (2) 更换可选模块。 (3) 系统断电,确认BTM和BRM在各自机架中插接良好,所有的电缆都正确连接并接触良好。 (4) 更换电缆。 (5) 更换 BRM (总线接收模板)。 (6) 更换 BTM (总线发送模板)。 (7) 向 GE Fanuc PLC 现场服务报告故障。
故障代码: 名称: 说明: 排除方法:	3B 丢失或缺少通讯驱动 使用 FULL MAIL配置模式的第三方VME模板与PLC CPU之间的通讯失败时,PLC产生此错误。 (1) 使用正确的通讯参数更新配置文件。 (2) 更换模板通讯驱动。 (3) 从配置文件中移除该模板。 (4) 更换模板。

故障代码: 名称: 说明: 排除方法:	3C 模板处于固件升级模式 当 PLC 发现模板处于固件升级模式时,产生该错误。处于这种模式下的模板将不能与 PLC CPU进行通讯。 (1) 完成该模板的固件升级。 (2) 通过按钮重启模板。 (3) 重启整个系统。 (4) 重启包含此模块的机架。
故障代码: 名称: 说明: 排除方法:	41 不能建立 VME 通讯 当发现有模板处于独立模式时产生该错误。处于独立模式的模板看上去运行正常,但它并不与PLC进行通讯。 (1) 重启该模板。 (2) 重启整个系统。 (3) 重启包含此模块的机架。
故障代码: 名称: 说明: 排除方法:	FF 可选模块通讯失败 当 PLC CPU与可选模块之间通讯失败时产生该错误。 (1) 检查总线的反常活动 (2) 更换直接请求的智能可选模块。 (3) 检查并行编程电缆是否正确连接。
故障代码: 名称: 说明: 排除方法:	4B CFG_486_NOT_COMPAT 当前安装的CPX PLC固件与 ESCM固件不兼容(内嵌的串行通讯模块固件—控制串口 1和 2的)。 Byte 0 = 最小所需的 CPX 主固件版本 Byte 1 = 最小所需的 CPX 主固件版本 Byte 2 = 当前安装的主固件版本 Byte 3 = 当前安装的次固件版本 CPX PLC固件需要同 ESCM 固件作兼容升级。
故障代码: 名称: 说明: 排除方法:	4C CFG_ESCM_NOT_COMPAT 当前安装的PLC固件与 ESCM固件不兼容(内嵌的串行通讯模块固件—控制串口 1和 2的)。 Byte 0 = 最小所需的 ESCM 主固件版本 Byte 1 = 最小所需的 ESCM 主固件版本 Byte 2 = 当前安装的主固件版本 Byte 3 = 当前安装的次固件版本 CPX PLC固件需要同 ESCM 固件作兼容升级。
故障代码: 名称: 说明: 排除方法:	51 ESCM_RESET_REQ ESCM (内嵌的串行通讯模块) 请求重新启动,但当前情况不允许。大多数可能的原因是ESCM 正在导入加载。 ESCM当完成升级时重新上电。

故障代码:	其他
名称:	模板配置失败
说明:	当上电或配置期间模板错误将会产生此类错误。
排除方法:	(1) 系统断电，更换该模板。 (2) 若此模板在扩展机架中，确认BTM/BRM电缆连接紧固，模板插接正确，并确认扩展机架地址。 (3) 更换 BTM。 (4) 更换BRM。 (5) 更换机架。

增加的或额外的机架

当已配置的扩展机架无法与PLC通讯或上电时发现一个未配置的机架时，就会产生此错误组。该故障缺省为诊断性故障。

错误代码:	1
名称:	多余机架
排除方法:	(1) 检查电源模板后的机架跳线是否正确。 (2) 更新包含扩展机架的配置文件。 (3) 从硬件配置中移除扩展机架。 <u>注意:</u> 如果是机架刚刚上电，则不需再去排除。

复位，增加或额外可选模板

当可选模块(PCM, BTM等)在线或被复位，或在机架中装置了一个模块，但并没有在配置中配置时将会产生此类错误。该错误默认为诊断性故障。

Error Code:	1
Name:	额外的可选模块
Correction:	(1) 升级包含该模板的配置文件。 (2) 从系统中移除该模板。.
Error Code:	2
Name:	模板重启完成
Description:	模板重启完成。
Correction:	无
Error Code:	3
Name:	LAN 接口重启完成，正在运行应用程序
Description:	LAN 接口重启完成，正在运行应用程序。
Correction:	见 LAN接口手册 GFK-0868 或 GFK-0869 (以前版本为GFK-0533)。

系统配置不匹配

当插槽中的模板与配置文件中的规格不同时就会产生配置不匹配故障。该故障默认为致命故障。当由于Genius块而导致I/O扫描器产生不匹配，故障附加数据区的第二个字节包含了不匹配块的总线地址。

错误代码:	2
名称:	Genius I/O块数量不匹配
说明:	当配置与实际Genius I/O块的数量不一致时会产生此类故障。
排除方法:	(1) 更换与配置相一致的Genius I/O块。 (2) 更新配置文件。
错误代码:	4
名称:	I/O 类型不匹配
说明:	当配置的Genius分组块的I/O类型与实际不一致时产生此故障。
排除方法:	(1) 移除错误的Genius模板并按照配置文件安装正确模板。 (2) 更新在配置文件中描述的Genius模板与实际安装相一致。
错误代码:	7
名称:	子板不匹配
说明:	当配置文件中的（扩展）内存子板大小与PLC CPU上实际的内存大小不相同时，将会产生此故障。
排除方法:	(1) 更换模板。 (2) 更换与配置文件中容量相同的子板。 (3) 更新为与PLC CPU上实际安装的子板容量相同的配置文件。
错误代码:	8
名称:	模拟扩展模板不匹配
说明:	当配置文件中模拟扩展模板型号与实际不相同时将会产生此错误。
排除方法:	(1) 更换与配置相符的模拟扩展模板。 (2) 更新配置文件。
错误代码:	9
名称:	Genius I/O 块大小不匹配
说明:	当实际块的大小与配置文件不匹配时，PLC产生此错误。
排除方法:	重新配置块。

错误代码:	A
名称:	不支持的特性
说明:	当前版本的模板不支持所配置的梯形。
排除方法:	(1) 升级该模板到支持该特性的版本。 (2) 更改模板配置。
错误代码:	B
名称:	版本A BTM 不位于最右槽
说明:	BTM (版本为 A) 不在机架的最右槽。
排除方法:	(1) 将BTM移至机架所有其他模板的最右槽。 (2) 升级BTM至更新的版本 (版本 B或更新的版本)。
错误代码:	E
名称:	LAN 重复 MAC 地址
说明:	LAN 接口模板与局域网中的其他设备有相同的MAC地址，该模板不能正常工作于此网络。
排除方法:	(1) 更改模板MAC地址。 (2) 更改其他设备的MAC地址。
错误代码:	F
名称:	LAN的重复MAC地址被解析
说明:	先前重复的 MAC 地址能够被解析。模板又回到网络中。 此为信息类提示。
排除方法:	不需。
错误代码:	10
名称:	LAN MAC地址不匹配
说明:	由软开关应用程序所配置的MAC地址同软件中存储的配置不匹配。
排除方法:	更改软开关应用或软件中的 MAC 地址。
错误代码:	11
名称:	LAN 软开关/调制解调器不匹配
说明:	LAN 模板的配置同调制解调器或软开关应用程序的配置不匹配。
排除方法:	(1) 改为正确调制解调器类型的配置。 (2) 参照 LAN 接口手册来进行配置设定。
错误代码:	17
名称:	非法内存变量
说明:	逻辑程序中使用的内存变量超出了允许的范围。
排除方法:	更新配置文件并下装到PLC。
错误代码:	1E
名称:	变量长度不匹配
说明:	当配置中的模板的I/O变量长度同板子的时间数据大小不相符时，PLC产生此错误。
排除方法:	以正确的变量长度更新配置文件。

错误代码:	1F
名称:	非法的配置参数
说明:	当PLC确定模板配置中的临界参数不能被接受时，产生此错误。
排除方法:	以正确值更新配置文件。
错误代码:	20
名称:	新配置需要重启
说明:	当PLC确定存储的配置试图更改相应模板的临界配置值时，产生此错误。除非该模板重启，否则新的配置不能生效。
排除方法:	(1) 重启整个系统。 (2) 重启包含该模板的机架。
错误代码:	27
名称:	不能解析或禁用的中断变量
说明:	当中断触发变量超出范围或在I/O模板配置中禁用时，产生该错误。T
排除方法:	(1) 移除或更正中断触发变量。 (2) 更新配置文件来允许该中断。
错误代码:	1D
名称:	不相容的时序模式
说明:	程序的时序模式与存储的扫描模式不相容。在由停止转运行时报告此错误。
排除方法:	(1)更改扫描模式，重试一次。 (2) 更改时序模式或在程序声明中删除不相容的程序。
错误代码:	24
名称:	I/O描述不匹配
说明:	I/O描述与程序中给的不一致。
排除方法:	通过更改I/O描述声明或C程序中的宏定义来排除此错误。
错误代码:	25
名称:	控制器变量超出范围
说明:	在触发器，禁用的或I/O描述中使用的变量超出了配置所允许的范围。
排除方法:	修改超出的变量到允许的范围，或增加该种变量所配置的空间。
错误代码:	26
名称:	损坏的程序描述
说明:	程序I/O 描述冲突。
排除方法:	联系 GE Fanuc 现场服务。
错误代码:	其他
名称:	模板或配置不匹配
说明:	当插槽中的模板与配置文件中的型号不一致时，产生此错误。
排除方法:	(1) 更换为配置文件中所配置的型号模板。 (2) 更新配置文件。

系统总线错误

当PLC CPU接收到来自于总线系统的一个不可配置的中断总线错误时，会产生此系统总线错误。该错误默认为诊断性故障。

错误代码:	4
名称:	不可识别的VME 中断源
说明:	当一个模板产生一个CPU不期望得到的中断（非配置的或不可识别的）时，会产生此错误。
排除方法:	(1) 确认配置中断的所有模板在程序逻辑中都有相应的中断声明。 (2) 确认数据线IRQ6和 IRQ7没有第三方VME模板产生中断。
错误代码:	其他
名称:	系统总线错误
说明:	The当PLC检测到一个VME背板总线上错误信号时，产生该错误，比如奇偶校验错误。
排除方法:	(1) 确保所有的扩展机架电缆连接正确接触良好。 (2) 采取措施降低系统噪音。

PLC CPU硬件故障

当PLC CPU检测到一个硬件故障时产生此故障。例如RAM区故障或通讯端口故障。当RAM区故障发生，故障地址保存于故障附加数据区的前四个字节中。

当PLC CPU硬件故障发生，PLC上的OK指示灯将会闪烁，表明此故障没有严重到不允许通过编程器通讯检索此故障。该组故障默认为致命性故障。

错误代码:	6Eh
名称:	日历时钟无后备电池
说明:	由后备电池维持的日历数据将丢失。
排除方法:	(1) 更换电池。更换完成前主机架不能断电。需使用编程器对日历重新设定。 (2) 更换模板。
错误代码:	其他
排除方法:	更换模板。.

模板硬件故障

当PLC CPU检测到系统中的任一块模板上有非致命硬件故障时，产生此故障。例如PCM板上的串口故障。该故障默认为诊断性故障。

错误代码:	1A0
名称:	缺少12伏供电
说明:	LAN 接口模板需要12伏电源供电才能运转。
排除方法:	(1) 安装或更换GE Fanuc 100 W电源模块。 (2) 连接外部VME电源供12伏电。
错误代码:	1C2-1C6
名称:	LAN接口硬件故障
说明:	参见 LAN接口手册， GFK-0868 或GFK-0869 (以前版本 GFK-0533)
错误代码:	其他
名称:	模板硬件故障
说明:	检测到模板硬件故障。
排除方法:	更换受影响的模板。

可选模板软件故障

当PCM模板上发生不可恢复的软件故障时会引起该故障。当从一个模块上读取的识别数据表明该模块是GE Fanuc模块但不是支持的类型时，也会产生该故障。此故障默认为致命故障。

错误代码:	1
名称:	不支持的模板类型
说明:	当从一个模块上读取的识别数据表明该模块是GE Fanuc模块但不是支持的类型时，也会产生该故障。
排除方法:	(1) 下装配置文件并确认该文件中的模板类型。如果有错误，请在下装正确的配置文件后重试。 (2) 通过编程器显示PLC故障表。联系GE Fanuc PLC的现场服务，提供故障条目中所包含的全部信息。
错误代码:	2, 3
名称:	COMMREQ 频率过高
说明:	COMMREQ向模板的发送要比对它们的处理快。
排除方法:	更改向受影响模板发送COMMREQ的PLC程序到一个较低的访问频率，或者在下次发送前监控每个COMMREQ 的完成状态。
错误代码:	4
名称:	机架中BTM 超过一块
说明:	机架中BTM 超过一块。
排除方法:	从机架中移除BTM模板，一个CPU机架中只能有一块。
错误代码:	191, 195
名称:	LAN 接口软件故障
说明:	参见 LAN接口手册， GFK-0868 或GFK-0869 (以前版本 GFK-0533)
错误代码:	其他
名称:	可选模块软件故障
说明:	在可选模块上检测到软件故障。
排除方法:	(1) 对相应可选模块重新下装软件。 (2) 更换模板。

程序（块）校验故障

当PLC CPU检测到由PLC接收的程序块中存在故障状态时，会产生该故障组。在RUN模式后台检验底板期间也会产生此故障。这些情况下，PLC故障表给出的故障附加数据区将记录该错误发生的程序或程序块的名称。该故障组默认为致命故障。

错误代码:	全部
名称:	程序或程序块校验错误
说明:	当程序或程序块冲突时，产生此错误。
排除方法:	(1) 清除PLC内存并重新存储。 (2) 检查应用程序是否有错。 (3) 通过编程器显示PLC故障表。联系GE Fanuc PLC的现场服务，提供故障条目中所包含的全部信息。

电池电量不足信号

当PLC CPU检测到CPU板上的电池电量不足，CPU内存子板或PCM之类的模板报告电池低电量时，产生此故障组。该故障组默认为诊断故障。

错误代码:	0
名称:	电池信号故障
说明:	CPU模板 (或其他带电池的模板)电池已不能正常工作。
排除方法:	更换电池，在更换完成前不能切断该机架供电。
错误代码:	1
名称:	电池低电量信号
说明:	CPU或其他模板上的电池低电量。
排除方法:	更换电池，在更换完成前不能切断该机架供电。

恒定时间或微循环扫描超时

当PLC CPU运行于恒定时间或微循环扫描方式时检测到扫描周期超过规定的时间时，会产生此故障组。故障附加数据的八个字节中包含此程序文件夹的名称。此故障组默认为诊断故障。

错误代码:	0 恒定时间扫描时 1 微循环扫描时
排除方法:	若为恒定时间扫描方式: (1) 增加扫描周期设定的时间。 (2) 从应用程序中移除部分逻辑。 若为微循环扫描方式 (1) 增加扫描时间设定。 (2) 修改执行的间隔给程序更多的时间执行。

PLC 故障表满

当PLC故障表达到所配置数量的上限时（见3-13页），将会产生此故障组。该故障组默认为诊断故障。

错误代码:	0
排除方法:	清除PLC故障表。

I/O 故障表满

当I/O故障表达到所配置数量的上限时（见3-14页），将会产生此故障组。为了避免丢失新增的故障，需要从故障表中清除以前的故障条目。该故障组默认为诊断故障。

错误代码:	0
排除方法:	清除 I/O故障表。

应用程序错误

当PLC CPU检测到用户程序中的错误时，将产生此错误组。此故障默认为诊断故障。

错误代码:	1
名称:	间接地址超限
说明:	当属于某个功能块的参数为间接地址变量时（即该参数是包含参数值的内存类型的地址）并且此间接变量的内容超出了该内存类型所允许的范围，PLC就会报此故障组。例如，某系统定义了500个%R寄存器，如果参数地址在%R00100中，则当%R00100中的数超出0到500的范围的话，就会产生此错误。 故障附加数据区包含在被调用的偏移地址的前两个字节中，段选择和偏移在接下来的四个字节中，功能调用的程序或程序块名称位于后八个字节中。
排除方法:	(1) 改正间接变量。 (2) 如果可能的话，增加现有寄存器数量。
错误代码:	2
名称:	软件看门狗定时器溢出
说明:	软件看门狗定时器溢出时将产生此错误。此时PLC CPU中断执行用户程序并转入STOP模式，恢复的唯一方法就是重新为PLC上电。引起定时器溢出的原因有：循环，跳转，程序过长等。
排除方法:	(1) 确定引起定时器溢出的原因（逻辑执行，外部事件等）并纠正。 (2) 使用系统服务功能块重新启动看门狗定时器。
错误代码:	5
名称:	COMMREQ WAIT 模式不支持
说明:	收到COMMREQ的模板不支持 WAIT模式的COMMREQ。
排除方法:	使用 NOWAIT模式 COMMREQ。
错误代码:	6
名称:	COMMREQ 不正确的任务 ID
说明:	由 COMMREQ选择的任务在可选模块中不存在。
排除方法:	更正任务ID。
错误代码:	7
名称:	应用重新堆栈溢出
说明:	程序调用的级数超出了PLC允许的范围。
排除方法:	增大程序的堆栈大小或调整应用重新减少嵌套。
错误代码:	8 到 D
名称:	LAN 接口应用程序错误。
说明:	详细信息 见LAN 接口手册 GFK-0868 或GFK-0869 (以前版本为 GFK-0533)。

错误代码:	0E
名称:	外部块运行时间故障
说明:	运行时间故障发生在外部块执行期间。
排除方法:	根据故障信息, 在外部块中修改相关问题。
错误代码:	0F
名称:	SORT 中断错误
说明:	在定时或I/O中断块中的SORT功能块执行的同时在其他程序块中也在执行 SORT 功能块。
排除方法:	不要在中断块和非中断块同时使用 SORT功能块。
错误代码:	11
名称:	单独的运行时间故障
说明:	在独立程序运行时会产生该错误。
排除方法:	根据故障信息, 在独立程序中修改相关问题。
错误代码:	1C
名称:	程序超出转入停止时间
说明:	在 PLC收到停止命令后的停止周期时间(通常2.5秒)内程序没有完成执行。
排除方法:	程序进入无穷循环或者是程序太长执行不完。修正错误代码或修改程序。
错误代码:	1D
名称:	程序未准备好
说明:	需准备好的程序时序在前一次执行中未完成。基本循环时间太短(周期程序), 或者中断率太高(I/O触发或定时程序)。
排除方法:	(1) 增加基本循环时间或降低中断率。 (2) 增加程序间隔时间来使程序执行完。

不可配置故障

不可配置故障的作用类不能改变。致命故障将导致PLC在发生故障的扫描周期结束时进入STOP模式。诊断故障记入故障表, 并将相应的故障接点置位。信息类故障只报故障于故障表。

系统总线故障

当PLC CPU软件接收到来自总线系统的不可配置中断总线故障时，将产生此故障组。此故障默认为致命故障。

错误代码:	1
名称:	总线严重故障
说明:	当 PLC CPU不能按照需要对VME总线进行控制时，PLC操作软件产生此故障。
排除方法:	(1) 确保任何可成为总线主控制器非GE Fanuc模板已放弃对VME总线的控制权，当PLC CPU发出此请求后。 (2) 更换 PLC CPU模板。

上电时无用户程序

当PLC上电但无用户程序时会产生此故障组。PLC上电后若检测到缺少用户程序，则控制器将会处于停止状态，运行停止模式下的PLC扫描直到有合法的程序下装到PLC。该故障组是信息类故障。

排除方法:	在尝试转入运行模式时首先下装程序。

上电时用户程序受损

当PLC CPU检测到用户RAM受损时，产生该故障组。PLC CPU将保持在停止模式，直到有效的用户程序及配置文件下装。该故障默认为致命故障。

错误代码:	1
名称:	上电时用户RAM受损
说明:	当PLC CPU检测到用户RAM受损时，产生该故障组。
排除方法:	(1) 在无电池情况下，重新上电。 (2) 检查C 应用程序中的错误。 (3) 更换 PLC CPU板上的电池。 (4) 更换PLC CPU上的控制内存板。 (5) 更换 PLC CPU。
错误代码:	2
名称:	检测到非法的布尔操作码
说明:	当PLC检测到错误指令时将产生此故障。
排除方法:	(1) 恢复用户程序和变量。 (2) 检查C 应用程序中的错误。 (3) 更换PLC CPU上的控制内存板。 (4) 更换 PLC CPU。
错误代码:	6
名称:	远程 I/O 扫描器 EEPROM损坏
说明:	上电时在远程 I/O 扫描器 EEPROM中的配置损坏。
排除方法:	远程 I/O 扫描器恢复配置。

窗口完成故障

窗口完成故障组由PLC预逻辑和扫描结尾处理软件所产生。故障附加数据包含了当错误发生时正在处理的任务名称。该故障组为信息类故障。

错误代码:	0
名称:	窗口完成故障
说明:	PLC运行于恒定时间扫描模式且在编程器窗口开始执行前已经到达恒定时间设定值时, 产生该错误。
排除方法:	增加恒定时间扫描的设定时间
错误代码:	1
名称:	逻辑窗口被跳过
说明:	由于没有时间来运行, 逻辑窗口被跳过。T
排除方法:	(1) 增加基本循环时间。 (2) 减少通讯窗口时间。

口令错误

当PLC CPU收到一个改变到一个新的优先级的请求, 并且包含在此请求中的此优先级的口令非法时, 将产生此错误组。该故障组默认为信息类故障。

错误代码:	0
排除方法:	输入正确口令重试。

运行模式无系统配置

当PLC需要从停止模式转换到运行模式, 配置文件不存在时, 将发生运行模式无系统配置错误组。此时PLC进入运行模式, 但无I/O扫描开始。此错误的影响就是触发延迟I/O功能。此故障组为信息类故障。

错误代码:	0
排除方法:	下装配置文件。

PLC CPU 系统软件错误

由90-70 PLC CPU的操作软件产生PLC CPU 系统软件错误故障组。它们会在系统操作的多种情况下出现。当致命故障发生，PLC CPU立即转入一种特殊的故障扫描模式。当PLC运行于此模式下时所运行的唯一活动就是与编程器进行通讯。排除此状态的唯一方法就是PLC重新上电。此故障组为致命故障。

错误代码:	14, 27
名称:	PLC程序内存损坏
说明:	当某些 PLC操作软件问题发生时，会出现此错误。它们不应发生在产品系统中。
排除方法:	(1) 通过编程器显示PLC故障表。联系GE Fanuc PLC的现场服务，提供故障条目中所包含的全部信息。 (2) 修正损坏的内存。
错误代码:	52
名称:	背板通讯故障
说明:	PLC试图进行需要背板通讯的应用请求，而收到拒绝的响应信号时产生此故障。
排除方法:	(1) 检查总线的反常活动。 (2) 更换直接请求的智能可选模块。 (3) 检查并行编程电缆是否正确连接。
错误代码:	5A
名称:	用户中止请求
说明:	PLC应用程序在执行 SVCREQ #13 （用户中止）时产生的报警信息。
排除方法:	不需要，仅为报警信息。
错误代码:	7B
名称:	远程 I/O 扫描器通讯脉冲错误
说明:	参见《系列90-70远程 I/O 扫描器用户手册GFK-0579》。
排除方法:	不需要，仅为报警信息。
错误代码:	94
名称:	单元中包含不匹配的固件，建议立即升级
说明:	此故障发生在冗余状态改变时，且冗余CPU包含不兼容的固件。
排除方法:	确保冗余CPU包含兼容的固件。
错误代码:	其他
名称:	PLC CPU 外部系统错误
说明:	外部系统错误不应发生在产品系统中。
排除方法:	通过编程器显示PLC故障表。联系GE Fanuc PLC的现场服务，提供故障条目中所包含的全部信息。

总线控制器过多

当PLC操作软件中的I/O扫描器部分检测到多于最大数量（32）的总线控制器被定义时，产生总线控制器过多故障组。系统中为70型I/O提供总线支持的PLC CPU本身就是一个总线控制器。该故障组为致命故障。

注意

为冗余和非冗余系统模块使用的Genius总线控制器是作为两个总线控制器存在的。

排除方法:	(1) 确定是总线控制器的模板并将多余的移除。 (2) 从配置文件中删除总线控制器并下装到PLC CPU。 (3) 若总线控制器从一个插槽移到另一个插槽，而在移动前无此故障发生，请为此机架重新上电。模板都不应在机架带电情况下进行拔插。 (4) 通过编程器显示PLC故障表。联系GE Fanuc PLC的现场服务，提供故障条目中所包含的全部信息。

存储时通讯失败

当程序或程序块和其他数据向PLC中存储期间可能会报存储时通讯失败故障组。向PLC中存储的命令和数据流由一个特殊的开始时序命令开始，并由结束时序命令结束。如果执行存储时与编程设备之间的通讯中断或任何其他原因导致存储中止，将会报此故障。系统中如果存在此故障，则编程器就不能使之转入运行模式。

此故障在重新上电时不自动清除，必须由用户清除此状态。此故障为致命故障。

错误代码:	0
排除方法:	清除故障重新下装程序和配置。

运行模式存储失败

错误代码:	1
说明:	在运行模式存储时通讯丢失，掉电等。新的重新未被激活而被删除。
排除方法:	重新运行存储，此故障为诊断故障。
错误代码:	2
说明:	在运行模式存储，清除旧程序或程序块时通讯丢失，掉电等。新的程序或程序块被安装，保留的程序和程序块被清除。
排除方法:	不需要，此为信息类故障。
错误代码:	3
说明:	在运行模式存储时掉电。
排除方法:	删除并恢复程序，此为致命故障。

第四节 I/O 故障表解释

I/O故障表报告数据有三类：

故障范畴

故障类型

故障说明

所有故障都有故障范畴，但不是所有都有故障类型和故障组。
关于每个故障的附加信息，通过在它的说明窗口双击来获取。

下表列出了每种故障范畴的说明信息。

表 3-8. 故障范畴说明

故障范畴	故障类型	故障说明	故障数据
回路故障	离散量故障	丢失用户侧电源	回路配置 *
		用户接线短路	回路配置 *
		持续过电流	回路配置 *
		低电流或无电流	回路配置 *
		开关温度过高	回路配置 *
		开关故障	回路配置 *
		点故障	回路配置 *
		输出熔断器损坏	回路配置 *
	模拟量故障	输入通道低报警	回路配置 *
		输入通道高报警	回路配置 *
		输入通道超过下限	回路配置 *
		输入通道超过上限	回路配置 *
		输出通道开路	回路配置 *
		输出通道低于下限	回路配置 *
		输出通道超过上限	回路配置 *
		非法数据	回路配置 *
		扩展通道无响应	回路配置 *
	低级模拟量故障	输入通道低报警	回路配置 *
		输入通道低报警	回路配置 *
		输入通道低于下限	回路配置 *
		输入通道高于上限	回路配置 *
		输入通道开路	回路配置 *
		接线错误	回路配置 *
		内部故障	回路配置 *
		输出通道短路	回路配置 *
		非法数据	回路配置 *
	GENA 故障	GENA 回路故障	GENA 故障字节2
	远程 I/O 扫描器故障	远程I/O扫描器回路故障	字节 1: 回路类型 字节 2: I/O类型
IOC未接通			超时 未知状态 未知信息状态 VME总线错误
添加 IOC			
I/O 模板未接通			
添加 I/O 模板			
额外I/O模板			
块未接通	故障无特性	通讯丢失	
添加块			
额外块			
I/O总线故障	总线错误总线输出禁止		
全局内存错误			分支组全局变量名称

*见下页。

表 3-8. 故障范畴说明 - 续

故障范畴	故障类型	故障说明	故障数据
模板故障	数据转发器故障	EPROM或NVRAM 错误共享内存失败，配置不匹配，看门狗超时，输出熔断器损坏。t	
IOC软件故障			
IOC硬件故障			
强制回路			块配置 * 离散量/模拟量显示*
Unforced Circuit			块配置 * 离散量/模拟量显示*

*见下表.

在许多故障范畴信息中包括3类故障特性数据，它们是回路配置，程序块配置，模拟/离散量指示。其代码所表示的含义见下表：

值	说明
	回路配置
1	回路为输入。
2	回路为输出。
3	回路为一个带反馈的输出。
	程序块配置
1	程序块仅配置为输入。
2	程序块仅配置为输出。
3	程序块仅配置为输入和输出。
	离散/模拟 指示
1	程序块为离散量块。
2	程序块为模拟量块。

回路故障

回路故障有四种故障类型，其中有三种有故障说明。它们都有相关的故障特性数据。回路故障专门应用于Genius I/O模板。默认为诊断性故障。下表给出了各种回路故障的说明。

表 3-9. 回路故障说明

故障 范畴	故障类型	故障说明	故障特性数据
回路故障	离散量故障	丢失用户侧电源	回路配置
		用户接线短路故障	回路配置
		持续过电流	回路配置
		低电流或无电流	回路配置
		开关温度过高	回路配置
		开关错误	回路配置
		点故障	回路配置
		输出熔断器断路	回路配置
	模拟量故障	输入通道低报警	回路配置
		输入通道高报警	回路配置
		输入通道超出低限	回路配置
		输入通道超出高限	回路配置
		输出通道开路	回路配置
		输出通道超出低限	回路配置
		输出通道超出高限	回路配置
		非法数据	回路配置
		扩展通道无响应	回路配置
	低等级 模拟量故障	输入通道低报警	回路配置
		输入通道高报警	回路配置
		输入通道超出低限	回路配置
		输入通道超出高限	回路配置
		输入通道开路	回路配置
		接线错误	回路配置
		内部故障	回路配置
		输入通道过短	回路配置
		非法数据	回路配置
	远程故障	远程 I/O 扫描器故障	

离散量故障

离散量故障有八种故障说明。在此故障报告中，有一种以上的情况发生。

名称:	用户电源未接通
说明:	当Genius I/O 模板侧的供电线路断开时，Genius总线控制器产生此错误。
排除方法:	(1) (仅对于隔离的I/O模块有效) 初始化“Pulse Test” COMREQ #1。在I/O模板上的脉冲测试功能可被启动和禁用。 (2) 纠正电源故障。
名称:	用户接线短路
说明:	当Genius总线控制器检测到某Genius模板接线短路时，产生此故障，短路电流定义为大于20A。
排除方法:	排除短路故障。
名称:	持续过电流
说明:	当Genius总线控制器检测到用户接线中流过大于2A的电流时，会产生此故障。
排除方法:	排除造成过电流的原因。
名称:	低电流或无电流
说明:	当用户电路中没有或有很小的电流时，Genius总线控制器产生此故障。
排除方法:	排除造成此情况的原因。
名称:	开关温度过高
说明:	当Genius总线控制器 (GBC)报告Genius智能开关温度过高时，产生此错误。
排除方法:	(1) 确保可为安装的模板提供足够的对流空间。 (2) 降低此板周围的环境温度。 (3) 安装RC缓冲器。
名称:	开关故障
说明:	当Genius总线控制器 (GBC)报告Genius智能开关故障时，产生此错误。
排除方法:	(1) 检查Genius输出的分路（按钮）。 (2) 更换Genius I/O模板。
名称:	点故障
说明:	当PLC 检测到Genius I/O模板上的单个点故障时，产生此错误。
排除方法:	更换Genius I/O 模板。
名称:	输出熔断器熔断
说明:	当PLC检测到Genius I/O输出模板上的熔断器熔断时，产生此错误。
排除方法:	(1) 确定并排除造成此现象的原因，然后更换熔断器。 (2) 更换模板。

模拟量故障

模拟量故障有九种故障说明，会有一种以上的情况出现在此故障的专门报告中。

名称:	输入通道信号过低报警
说明:	当Genius模拟量模板输入通道上信号太弱时，Genius总线控制器产生此故障。
排除方法:	纠正造成低报警的原因。
名称:	输入通道高报警
说明:	当Genius模拟量模板输入通道上信号过高时，Genius总线控制器产生此故障。
排除方法:	纠正造成高报警的原因。
名称:	输入通道低于低限值
说明:	当Genius模拟量模板输入通道上信号低于低限值时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道高于高限值
说明:	当Genius模拟量模板输入通道上信号高于高限值时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道断路
说明:	当Genius模拟量模板输入通道发生断路故障时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输出通道低于低限值
说明:	当Genius模拟量模板输出通道上信号低于低限值时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输出通道高于高限值
说明:	当Genius模拟量模板输出通道上信号高于高限值时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	非法数据
说明:	当Genius模拟量模板输入通道检测到无效数据时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	扩展通道无响应
说明:	当来自多路模拟输入模块的扩展通道上的数据无响应时，PLC产生此错误。
排除方法:	(1) 检查模板接线。 (2) 更换模板。

低等级模拟量故障

低等级模拟量故障有九种故障说明。会有一种以上的情况出现在此故障的专门报告中。

名称:	输入通道低报警
说明:	当Genius模拟量模板输入通道上信号太弱时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道高报警
说明:	当Genius模拟量模板输入通道上信号过高时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道低于下限值
说明:	当Genius模拟量模板输入通道上信号低于低限值时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道高于上限值
说明:	当Genius模拟量模板输入通道上信号高于高限值时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道开路
说明:	当Genius模拟量模板输入通道发生断路故障时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	接线错误
说明:	当Genius模拟量模板输入检测到一个错误的RTD连接或热电偶时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	内部故障
说明:	当Genius模拟量模板输入检测到一个有内部错误的RTD或冷连接的热电偶传感器故障时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。
名称:	输入通道短路
说明:	当Genius总线控制器检测到RTD或应变模块上输入通道短路时，会产生此故障。
排除方法:	纠正造成现象的原因。
名称:	无效数据
说明:	当Genius模拟量模板输入通道检测到无效数据时，Genius总线控制器产生此故障。
排除方法:	纠正造成现象的原因。

GENA故障

GENA故障没有相关的故障说明。GENA故障字节2是故障特定数据的第一个字节。

说明:	当Genius I/O 操作软件检测到Genius I/O总线上的GENA模块发生故障时,它就产生此故障。
排除方法:	更换GENA模块。

IOC未接通(I/O控制器)

该故障类别没有相关的故障类型和故障说明。该故障默认为致命故障。

名称:	IOC未接通或空缺
说明:	当PLC不能与I/O控制器通讯并在配置文件中存在此I/O控制器时产生此错误。
排除方法:	(1) 确认该插槽/总线地址中的模块是否正确。 (2) 重新检查配置文件是否正确。 (3) 更换模板。 (4) 在编程器上显示此PLC故障表, GE Fanuc PLC 现场服务联系,并向它们提供故障表中记录的所有信息。

额外的IOC (I/O 控制器)

该故障类别没有相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	额外的 IOC
说明:	当出过故障的IOC重新恢复运行时,或当系统中发现一个IOC,但配置文件指示在那个插槽中无IOC时,PLC操作软件产生此故障。
排除方法:	(1) 若故障模块在远程机架并且是由于某个远程机架循环上电而返回时,则没有必要再采取措施来排除故障。 (2) 更新配置文件或移除该模板。

I/O模板未接通

该故障组适用于70型离散量和模拟量I/O模板。没有与之相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	I/O模板未接通
说明:	当PLC检测到一个70型的模板不再对来自PLC CPU的命令有响应, 或配置文件中指示某个模板占据一个槽而该槽中没有该模板存在时, PLC就会产生此故障。
排除方法:	(1) 更换该模板。 (2) 更正配置文件。 (3) 在编程器上显示此PLC故障表, GE Fanuc PLC 现场服务联系, 并向它们提供故障表中记录的所有信息。

附加 I/O模板

该故障组适用于70型离散量和模拟量I/O模板。没有与之相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	附加 I/O模板
说明:	当出过故障的I/O模板重新恢复运行时, PLC产生此故障。
排除方法:	(1) 若故障模块在远程机架并且是由于某个远程机架循环上电而返回时, 则没有必要再采取措施来排除故障。 (2) 更新配置文件或移除该模板。

额外 I/O模板

该故障组适用于70型离散量和模拟量I/O模板。没有与之相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	额外 I/O模板
说明:	当 PLC检测到一个70型I/O模板但在配置文件中该槽为空时, 产生此故障。
排除方法:	(1) 移除该模板 (它可能在错误的槽位上)。 (2) 更新并存储新的配置文件。

块未接通

该故障组适用于GENIUS模板。没有与之相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	块未接通
说明:	当PLC从GBC上收到一个块未接通的故障，而其原因又没有给出时，PLC就会产生此故障。
排除方法:	(1) 确认电源和接线正确。 (2) 更换此块。
名称:	块未接通 - A/D 通讯故障
说明:	当Genius I/O操作软件检测到它与Genius I/O块的通讯中断时，产生此错误。
排除方法:	(1) 确认电源和串行总线接线正确。 (2) 更换此块。

附加块

该故障组适用于GENIUS模板。没有与之相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	附加块
说明:	当Genius I/O操作软件检测到与Genius I/O块的通讯中断的模板重新与之建立通讯时，产生此错误。
排除方法:	不需要，仅为提示信息。

额外块

该故障组适用于GENIUS模板。没有与之相关的故障类型和故障说明。该故障默认为诊断性故障。

名称:	额外块
说明:	当 PLC检测到串行地址总线上的Genius I/O块，而此块并未出现在配置文件中时，PLC操作软件会产生此故障。
排除方法:	(1) 移除或重新配置此块（它可能位于错误的串行总线地址上）。 (2) 更新并存储新的配置文件。

I/O 总线错误

该故障有两种相关的故障类型。此故障默认为诊断性故障。

名称:	总线故障
说明:	Genius总线控制器(GBC)检测到Genius I/O总线上的错误时, 操作软件会产生此错误。(通常当错误率超过GBC中的设置时产生—此值默认为10, 即在10秒期间有10个错误产生。)
排除方法:	(1) 确定导致总线故障的原因并纠正。 (2) 更换Genius总线控制器。 (3) 需要时, 可将默认值10适当提高, 但应该进行总线的电气检查—使用示波器进行波形检查。
名称:	禁止总线输出
说明:	Genius总线控制器在等待PLC CPU进行I/O扫描时超时, 操作软件会产生此错误。
排除方法:	(1) 更换 PLC CPU。 (2) 在编程器上显示此PLC故障表, GE Fanuc PLC 现场服务联系, 并向它们提供故障表中记录的所有信息。
名称:	SBA冲突
说明:	Genius总线控制器(GBC)检测到串行总线地址与总线上的其他设备冲突。
排除方法:	调整相冲突的一个串行总线地址。

模板故障

该故障类别有一种故障类型，初始故障，有八种故障说明，无故障特性数据。该故障缺省为诊断性故障。

名称:	配置内存故障
说明:	Genius总线控制器检测到Genius 模块上的EEPROM或NVRAM故障时，产生此错误。
排除方法:	更换Genius块的电子模块。
名称:	校正内存失败
说明:	Genius总线控制器校正内存失败时，产生此错误。
排除方法:	更换Genius块的电子模块。
名称:	共享 RAM故障
说明:	Genius总线控制器检测到Genius共享RAM故障时，产生此错误。
排除方法:	更换Genius块的电子模块。
名称:	看门狗超时
说明:	Genius总线控制器检测到Genius输入模板看门狗定时器超时，产生此错误。
排除方法:	更换 70型输入模板。
名称:	输出熔断器烧断
说明:	Genius总线控制器检测到Genius输出熔断器烧断时，产生此错误。
排除方法:	(1) 确定并修复熔断的原因，并更换熔断器。 (2) 更换此模板。
名称:	模板故障
说明:	模板内部检测到有故障。
排除方法:	更换受影响的模板。

IOC (I/O控制器)软件故障

IOC 软件故障适用于任何类型的I/O控制器。无与之相关的故障类型或故障说明。该故障为致命故障。

名称:	报文队列满，读写队列满
说明:	过多的报文或读写请求发送到总线控制器。
排除方法:	调整系统以减少向Genius总线控制器的请求发送率。
名称:	响应丢失
说明:	Genius总线控制器在收到报文或读写请求后不能响应。
排除方法:	调整系统以减少向Genius总线控制器的请求发送率。

IOC (I/O控制器)硬件故障

该故障类别无相关的故障类型和故障说明。此故障默认为诊断性故障。

说明:	Genius操作软件检测到总线通讯硬件故障或波特率不匹配时，产生此错误。
排除方法:	(1) 确定配置文件中的波特率设定，使之与总线上每个通讯块的波特率相一致。 (2) 必要时修改配置文件并存储于PLC。 (3) 更换Genius总线控制器。 (4) 选择性地把每个块从总线上取下，直到检查出故障块，再更换此故障块。

强制与解除强制回路

该故障类别报告的是点状态，因此从技术上讲不是故障，也没有相关的故障类型或故障说明。当一个Genius I/O点用手持监控器强制或解除强制时，会产生此报告。此为一提示性信息。

故障特定数据包含以下数据：

字节号	说明
1	块配置
2	模拟量/离散量信息

块开关

该故障类别无故障类型或故障说明。此故障默认为诊断性故障。

名称:	块开关
说明:	当冗余Genius总线上的Genius块从一个总线切换到另一个总线时，PLC产生此错误。
排除方法:	(1) 不需采取措施就能保持该模块运行。 (2) 修正总线故障： (a) 确认总线的接线。 (b) 更换I/O控制器。 (c) 更换总线开关模板(BSM)。

Chapter 4

第四章 继电器功能

PLC编程即为创建应用程序的过程。第四章到第十二章介绍了在对90-70 可编程序控制器进行梯形图逻辑编程时可能要用到的编程指令。

如果尚未安装Logicmaster 90编程软件，请参阅《Logicmaster 90-70编程软件使用手册GFK-0263》，该手册介绍了如何创建，转换，编辑和打印程序。

配置就是为系统中的硬件分配逻辑地址及其他特性的过程，该过程使用配置软件既可在编程前也可在编程后进行，然而推荐首先进行配置。如果您还没有进行硬件配置，请参阅《Logicmaster 90-70编程软件使用手册GFK-0263》，以确定开始编程的最佳时期。

本章介绍梯形图中使用的接点，线圈和逻辑连接线的使用方法。

功能	页码
使用接点	4-2
使用线圈	4-3
常开接点和常闭接点	4-4
正跳变和负跳变接点	4-4
故障接点和无故障接点	4-7
高报警接点和低报警接点	4-7
线圈和负线圈	4-8
保持线圈和负向保持线圈	4-8
正跳变和负跳变线圈	4-9
置位和复位线圈	4-10
保持型置位和复位线圈	4-11
水平和垂直连接线	4-11
续行线圈和续行接点	4-12

使用接点

接点用于监控一个变量的状态。接点是否通过电流取决于被监控变量的状态或情况和该接点的类型。变量为1时称为ON状态，变量为0时称为OFF状态。

表 4-1. 接点的类型

Type of Contact	Display	Contact Passes Power to Right:
常开接点	— —	当变量为ON时。
常闭接点	— /—	当变量为OFF 时。
正跳变接点	— ↑ —	当变量跳变为ON时。
负跳变接点	— ↓ —	当变量跳变为OFF时。
故障接点	—[FAULT]—	当变量有点故障时。
无故障接点	—[NOFLT]—	当变量无点故障时。
高报警接点	—HIALR]—	当变量超过高报警时。
低报警接点	—LOALR]—	当变量超过低报警时。
续行接点	<+>—	当其前面的续行线圈ON时。

使用线圈

线圈用于控制离散变量。必须使用条件逻辑来控制给线圈的电流。线圈直接触发动作，它不再向右传递电流。如果使用线圈条件的结果来执行程序的另外一些逻辑，此线圈应使用内部变量或者续行线圈/接点组合。

线圈总是位于一行逻辑的最右端，一个程序横档程序最多有八个线圈。

要根据期望得到的程序响应来选择所使用的线圈的类型。保持线圈的状态在PLC掉电或从STOP状态转入RUN模式时将被保存。非保持线圈的状态在PLC掉电或从STOP状态转入RUN模式时将被清零。

表 4-2. 线圈类型

线圈类型	显示	到线圈的电流	结果
线圈（常开）	-()-	ON	置变量为 ON。
		OFF	置变量为 OFF。
负向线圈	-(/)-	ON	置变量为 OFF。
		OFF	置变量为 ON。
保持线圈	-(M)-	ON	置变量为 ON，并保持。
		OFF	置变量为 OFF，并保持。
负向保持线圈	-(/M)-	ON	置变量为 OFF，并保持。
		OFF	置变量为 ON，并保持。
正跳变线圈	-(↑)-	OFF → ON	若变量为OFF，则将它ON一个扫描周期。
负跳变线圈	-(↓)-	ON → OFF	若变量为OFF，则将它ON一个扫描周期。
置位线圈	-(S)-	ON	置变量为 ON直到由 -(R)- 复位为OFF。
		OFF	不改变线圈状态。
复位线圈	-(R)-	ON	复位变量为OFF直到由 -(S)-置位为ON。
		OFF	不改变线圈状态。
保持型置位线圈	-(SM)-	ON	置变量为 ON直到由 -(RM)- 复位为OFF。
		OFF	不改变线圈状态。
保持型复位线圈	-(RM)-	ON	复位变量为直OFF到由 -(SM)- 复位为ON。
		OFF	不改变线圈状态。
续行线圈	——<+>	ON	置下面的续行接点ON。
		OFF	置下面的续行接点OFF。

注意

对于参数子程序块的限制，参见第二章第二节“参数子程序块中形式参数的限制”部分内容。

常开接点 

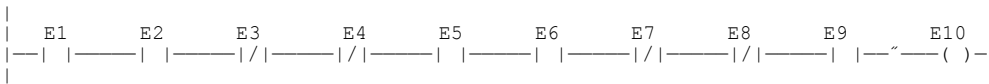
若相对应的变量ON（1）时，常开接点如同开关动作一样使电流通过。

常闭接点 

若相对应的变量OFF（0）时，常闭接点如同开关动作一样使电流通过。

例:

下图为梯形图程序中的一行，其上有别名为E1到E10的十个元件。当E1, E2, E5, E6和E9 为ON且变量E3, E4, E7和 E8 为 OFF时，线圈E10接通，为ON状态。



正跳变接点 

负跳变接点 

当与负跳变接点相关联的变量从ON (1)到OFF(0)变化时，它接通而通过电流。从当前ON (1)到下一个写入值OFF (0)的转变确定了此跳变。这些写入可在PLC扫描周期里发生多次，结果使跳变位仅在扫描周期的部分时间内被置位；或者发生在几个独立的扫描中，结果使跳变位在不只一个扫描周期内被置位。对于使用跳变线圈或置位复位线圈的情况下不要再使用跳变接点。

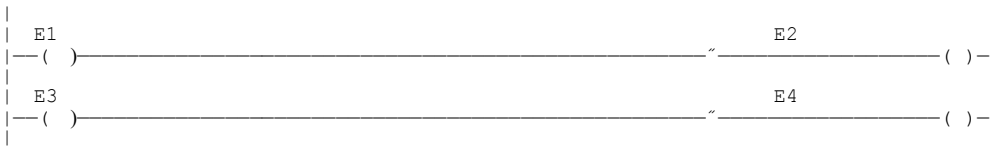
跳变接点的附加说明

当变量点被写入时，该变量的跳变位将受影响。当变量点从OFF变化到ON或者从ON到OFF时，跳变位被置位。如果是从ON到ON或从OFF到OFF，则跳变位被复位。写入源并不重要，它可以是一个输出线圈，一个功能块的输出，输入扫描，一个输入中断，一个PCM SYSWRITE，一个从程序来的数据改变，或是Genius数据报文。一旦变量点被写入，则跳变位状态立即发生变化。跳变位不能通过它自身的扫描来改变状态，仅在变量点写入时改变。为了清除跳变位必须对该点变量进行写入操作，否则它将会出现“阻塞”现象。除了已配置的输入点外，每次扫描不会自动清除跳变位，只有当读入输入数据且输入点的状态与上一次扫描所读状态一致时，跳变位才被清除。

强制并不对跳变位发生作用。如果对一个强制点写操作，跳变位也会被清除。例如，一个强制输入点的跳变位在输入扫描时会被清除。

例 1:

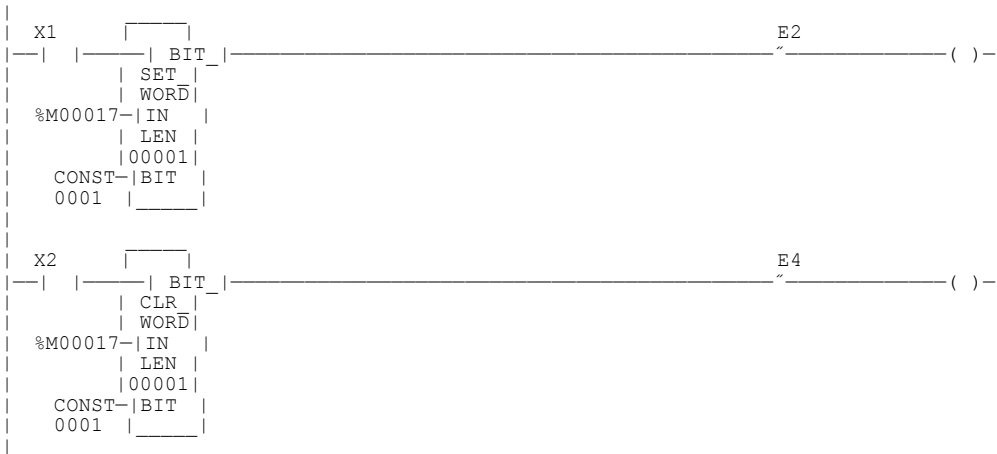
本例说明了正跳变和负跳变接点的使用方法。当在一次扫描中元件E1从OFF到ON变化时，线圈E2也为ON；元件E3从ON到OFF变化时，线圈E4也为ON。



例 2:

本例中，%M00017由BSET功能块置位后又由BCLR功能块复位。正跳变接点X1激活BSET，负跳变接点X2激活BCLR。

与%M00017相关的正跳变接点在%M00017由BCLR功能块复位前一直保持ON。这是因为该位仅当接点X1由OFF变化到ON时才被写入。同样，与%M00017相关的负跳变接点在%M00017由BSET功能块置位前一直保持ON状态。



故障接点

-[FAULT]-

故障接点用于检测离散量或模拟量变量的故障，或来定位故障（机架，槽，总线）。如果希望得到I/O点故障报告，则 必须在CPU配置中将I/O故障接点功能启用。详细内容参见第二章，第三节，“程序组织和用户数据”部分。

当与故障接点相关的变量或位置有点故障发生时，故障接点通电。

注意

使用故障或无故障接点用于物理变量地址(%I, %Q, %AI, %AQ)时需确保正确的模板状态指示。（当使用机架，槽，总线，故障或无故障接点的模板变量，当相关故障从故障表中清除时给定模板的故障显示也被清除。）

无故障接点

-[NOFLT]-

无故障接点也用于检测离散量或模拟量变量的故障，如果希望得到I/O点故障报告，也必须在CPU配置中将I/O故障接点功能启用。详细内容参见第二章，第三节，“程序组织和用户数据”部分。

当与无故障接点相关的变量或位置无点故障发生时，故障接点通电。

注意

使用故障或无故障接点用于物理变量地址(%I, %Q, %AI, %AQ)时需确保正确的模板状态指示。（当使用机架，槽，总线，故障或无故障接点的模板变量，当相关故障从故障表中清除时给定模板的故障显示也被清除。）

高报警接点

-[HIALR]-

高报警接点用于检测与模拟变量相关的高报警。要使用此接点和低报警接点必须在CPU配置中使能该项功能。（在配置软件菜单中，选择“内存分配和电故障使能”，把电故障变量项从**DISABLED** 改为 **ENABLED**）

若与模拟变量相关的高报警位置位时，高报警接点接通。

低报警接点

-[LOALR]-

低报警接点用于检测与模拟变量相关的低报警。和上面介绍的高报警相同，要使用此接点接点必须在CPU配置中使能该项功能。

若与模拟变量相关的低报警位置位时，低报警接点接通。

线圈 **-()-**

当线圈接受到电流时，与之相关的离散变量ON。它是非保持型的，故不能用于系统状态变量(%SA, %SB, %SC或%G)。

例:

本例中，当变量E1为ON且E2为OFF时，线圈E3得电。



负向线圈 **-(/)-**

当不得电时，负向线圈设定与之相关离散变量为ON状态。它是非保持型的，故不能用于系统状态变量(%SA, %SB, %SC或%G)。

例:

本例中，当E1为OFF时E2得电。



保持型线圈 **-(M)-**

同常开线圈一样，当线圈接受到电流时，与之相关的离散变量ON。保持型线圈的状态在失电时能够保持。故它不能用于严格的非保持型内存变量(%T)。

负向保持型线圈 **-(/M)-**

当线圈未接受到电流时，与之相关的离散变量ON。保持型线圈的状态在失电时能够保持。故它不能用于严格的非保持型内存变量(%T)。

正跳变线圈

-(↑)-

如果与正跳变线圈相对应的变量为OFF，当线圈得电它将置为ON状态直到下次执行到该线圈。（如果包含此线圈的程序行在后来扫描中被跳过，则它将仍然保持ON状态。）该线圈当作一个单触发来使用。

不要使用外部设备（如PCM）向用作正跳变线圈的变量写入数据，因为那样会破坏此线圈的单触发属性。

注意

不要在跳变线圈上使用跳变接点，因为线圈使用跳变位来存储到线圈的电流值。

跳变线圈可用于保持型和非保持型内存变量（%Q, %M, %T, %G, %SA, %SB或%SC）。

负向跳变线圈

-(↓)-

如果与负向跳变线圈相对应的变量为OFF，当线圈断电它将置为ON状态直到下次执行到该线圈。

注意

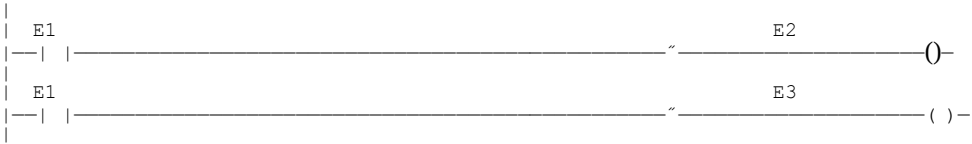
不要在跳变线圈上使用跳变接点，因为线圈使用跳变位来存储到线圈的电流值。

不要在跳变线圈上使用跳变接点，因为线圈使用跳变位来存储到线圈的电流值。

跳变线圈可用于保持型和非保持型内存变量（%Q, %M, %T, %G, %SA, %SB或%SC）。

例:

本例中，当变量E1从OFF到ON，线圈E2和E3得电，此时E2变为ON并保持一个逻辑扫描。当E1从ON到OFF，E2和E3失电，此时E3变为ON一个扫描周期。



置位线圈

-(S)-

置位和复位线圈为非保持型线圈，用于保持变量（如E1）状态ON或者OFF。当置位线圈得电，它将与之相关离散变量置位(不管线圈本身是否得电)直到有复位线圈复位此变量才改变状态(如下例中的E1)。

注意

置位线圈向给定变量的跳变位写入一个不确定值。
不要对同一变量同时使用置位线圈和跳变接点。

复位线圈

-(R)-

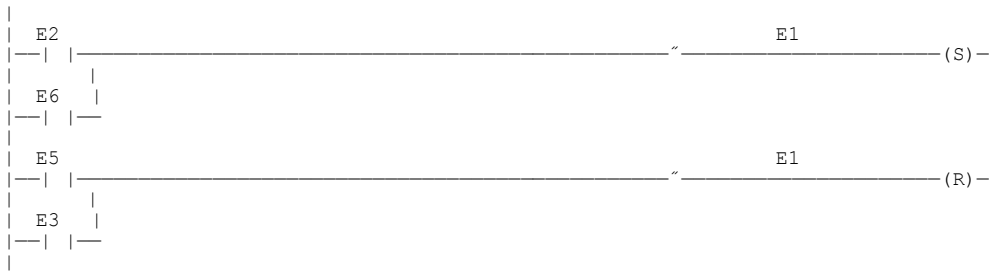
当线圈得电，将相关变量复位。该变量将一直保持OFF状态，直到置位线圈置位相同的变量（如下例中的E1）。最近处理的置位或复位线圈优先起作用。

注意

复位线圈向给定变量的跳变位写入一个不确定值。
不要对同一变量同时使用复位线圈和跳变接点。

例:

本例中，当变量E2或E6为ON时，线圈E1变为ON；当变量E3或E5为ON时，线圈E1变为OFF。



保持型置位线圈

-(SM)-

保持型置位复位线圈同置位复位线圈类似，但当PLC从STOP模式转到RUN模式或失电后其状态仍然保持。当保持型置位线圈得电时，它将置位与之相关离散变量。该变量将保持ON直到有保持型复位线圈将它复位。

注意

保持型置位线圈向给定变量的跳变位写入一个不确定值。
不要对同一变量同时使用保持型置位线圈和跳变接点。（参见第二章“跳变与强制”部分内容。）

保持型复位线圈

-(RM)-

该线圈得电将置相关离散变量为OFF状态。该变量将保持OFF直到有保持型置位线圈将它置位。当PLC从STOP模式转到RUN模式或失电后其状态仍然保持。

注意

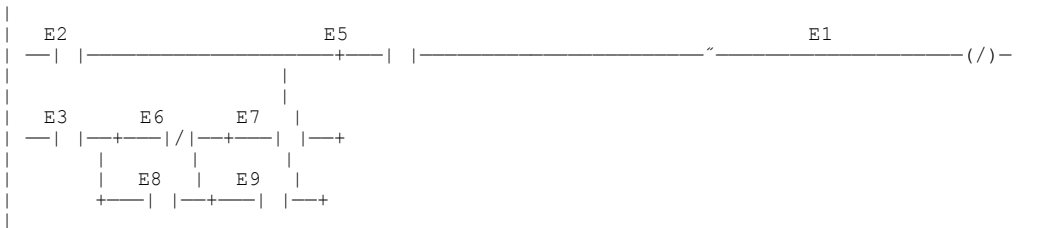
保持型复位线圈向给定变量的跳变位写入一个不确定值。
不要对同一变量同时使用保持型复位线圈和跳变接点。（参见第二章“跳变与强制”部分内容。）

连接线

水平和垂直连接线用于连接梯形图逻辑中的元件。其目的是实现梯形图逻辑中从左至右的电流通路。

例:

本例中，两条平行连接线用于连接接点E2和E5，垂直连接线用于连接E3,E6,E7,E8和E9到E2。



续行线圈 (---<+>) 和续行接点 (<+>---)

续行线圈(---<+>) 和续行接点 (<+>---)用于当继电器逻辑超过十列时的延续。最近执行的续行线圈的状态就是下一个续行接点的状态。如果在续行接点前无续行线圈则该续行接点将无电流通过。当PLC由STOP模式转为RUN模式时，续行接点的状态会被清除，并且在线圈未由PLC转入RUN状态而置位前不会再有电流通过。

每程序行只允许有一个续行线圈，续行接点必须位于第一列，而续行线圈必须位于第十列。

注意

续行线圈和接点是不能再进入执行，它们必须不用在不同的中断块中。

Chapter 5

第五章 计时器和计数器

本章介绍了如何使用延时接通，延时断开和秒表型定时器，增计数器和减计数器。与这些功能块相关的数据在电源周期内均是保持的。

缩写	功能	页码
ONDTR	保持型延时接通计时器	5-3
OFDT	延时关断计时器	5- 6
TMR	简单延时接通计时器	5- 9
UPCTR	递增计数器	5-12
DNCTR	递减计数器	5-14

计时器和计数器功能块数据需求

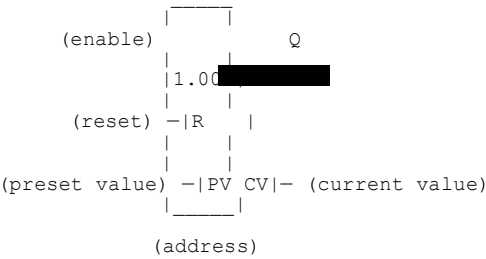
每个计时器或计数器使用三个字（寄存器）%R, %P或%L来存储以下数据：

当前值 (CV)	word 1
设定值 (PV)	word 2
控制字	word 3

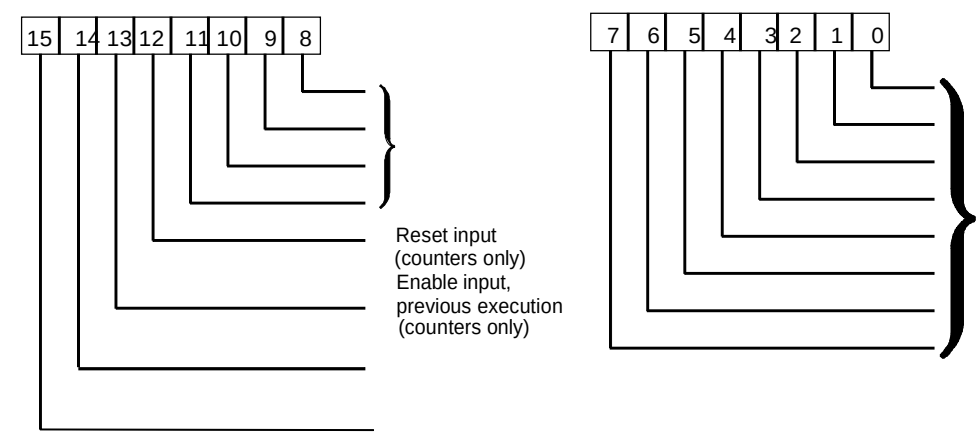
注意

当使用计时器功能块时不允许其他功能块再使用这些寄存器。

当在梯形图中输入一个计时器或计数器时，必须在此功能块的下面部分直接给出三个字的地址。例如：



控制字为与功能块相关的布尔输入和输出状态，其格式如下图所示：



Bit 0 到 13 用于计时器精度，bit 0 到 10 不用于计数器。

保持型延时接通计时器ONDTR

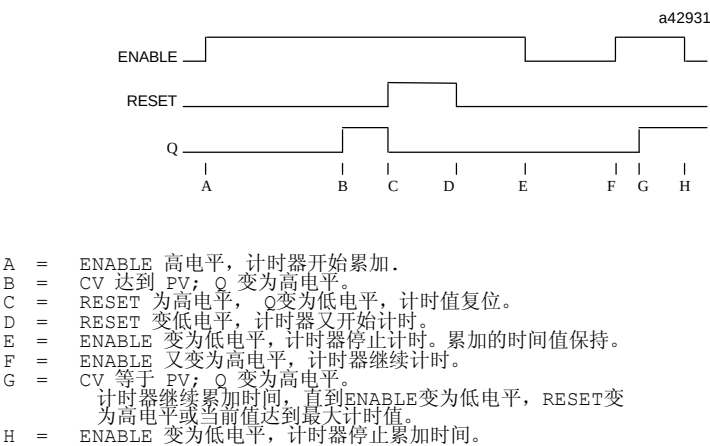
保持型延时接通计时器 (ONDTR)得电开始累加计时，失电停止计时并保持当前计时值。计时单位有秒（默认选项），0.1秒，0.01秒。时间单位的取值范围为0到+32,767。计时器的当前值失电保持，在重新得电时不会自动初始化此数据。

当ONDTR首次得电，它开始累加计时（当前值）。当梯形图逻辑中使用该计时器时，它的当前值将被刷新。

注意

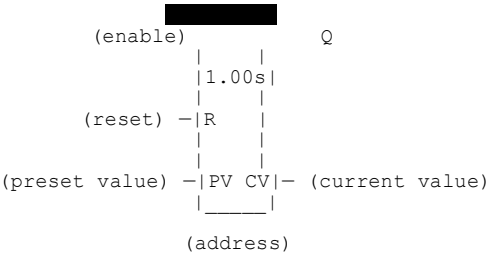
如果在一次CPU扫描中，多次出现具有相同变量地址的相同的计时器被执行，则这些计时器的当前值是相同的。

当前值(CV)等于或大于设定值(PV)时，输出Q得电。只要计时器继续在得电状态，它的计时值就会一直累加，一直达到它的计时最大值。一旦达到最大值，则它将保持并使输出Q保持得电状态而不受使能输入的影响。



当计时器失电，当前计时值停止累加并保持。输出Q若已经得电，将保持得电状态。当计时器功能块又重新得电，当前值从保持值开始累加。当复位端(R)得电，当前值被清零，输出Q在当前值达到设定值前不会再得电。

如果PV=0且计时器使能，计时器的输出将得电。这种情况下，即使撤除使能信号或使复位端得电都不会改变计时器输出的得电状态。



当ONDTR用在不是每次扫描都要被调用的程序块中时，该计时器在调用该程序块期间累计时间，除非它被复位。这就是说，它起到了运行于某个程序中的计时器作用，具有比主程序块中计时器慢得多的计时时间。由于该程序块可能很长时间不被调用，所以必须对计时器进行编程以允许它的catch-up特性。

例如，若在一个程序块中的计时器被复位且该程序在4分钟之间未被调用，当再调用程序块时，这4分钟已被累计上了。这段时间被加入到计时时间中，除非计时器被重新复位。

参数:

参数	说明
地址	ONDTR 功能块使用3个连续字（寄存器）%R, %P或%L 内存来存储下面的数据: • 当前值(CV) = word 1. • 设定值(PV) = word 2. • 控制字 = word 3. 当你在梯形图中使用ONDTR，必须为它指定这三个连续字的地址，写在代表此功能块的图形块的下方。详细信息见4-13页。 <u>注意:</u> 不要再在其他指令中使用这些地址。 <u>小心:</u> 变量重复使用会造成计时器运行的失常。
enable	当使能得电时，计时器的当前值开始累计。
R	当R得电时，计时器值被复位到0。
PV	当计时器被复位或被选通时，PV值被写入计时器作为预置值。对于一个寄存器(%R)PV变量，PV参数被指定为此地址参数的第二个字。例如当地址参数为%R00001则%R00002即为PV参数。
Q	当前值大于或等于设定值时，输出Q得电。由于在系统上电时不自动初始化Q的状态，Q的状态将掉电保持。
CV	CV 为计时器当前计时值。

有效的存储器类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
address									•	•	•			•		
enable	•															
R			•													
PV	•	•	•	•	•		•		•	•	•	•	•	•	•	•
Q	•															•
CV	•	•	•	•	•		•		•	•	•	•	•	•		•

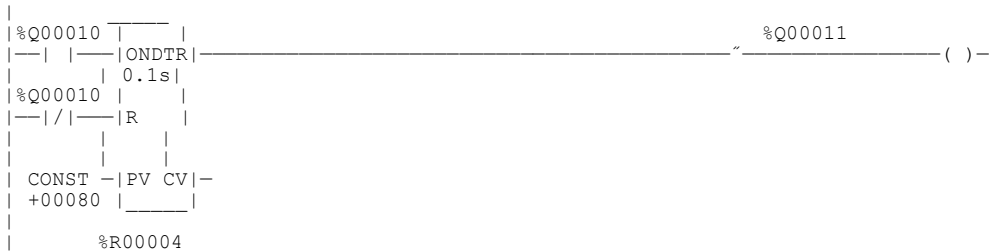
- 可使电流流经功能块的变量或地址。

注意

对参数子程序块的限制，见第二章第二节“参数子程序块形式参数的限制”部分内容。

例:

本例中，保持型延时接通计时器在%Q00010接通后8秒产生一个信号(%Q00011)，当%Q00010失电该信号也随之关闭。



延时关断计时器OFDT

延时关断计时器(OFDT)当失电时计时值开始累计，在得电时计时值复位为零。计时单位有秒（默认选项），0.1秒，0.01秒。时间单位的取值范围为0到+32,767。计时器的当前值失电保持，在重新得电时不会自动初始化此数据。

当OFDT首次得电，它开始向右侧传送电流且当前值(CV)被置为零。功能块得电期间输出将一直保持ON。如果功能块不再由左侧得到电流，它将继续为右侧提供电流，计时器开始在CV中累计时间。

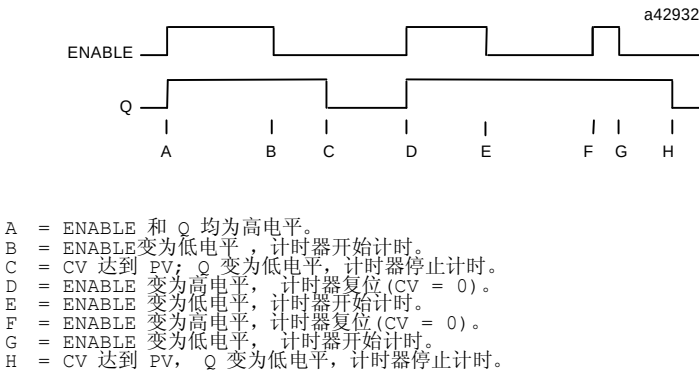
注意

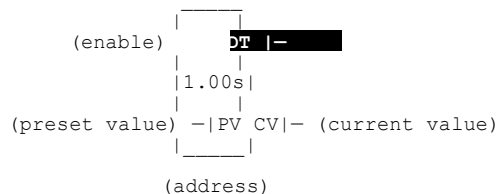
如果在一次CPU扫描中，多次出现具有相同变量地址的相同的计时器被执行，则这些计时器的当前值是相同的。

OFDT块在设定值为零或者为负数时，不通过电流。

每次计时功能随使能逻辑被设为OFF时，当前值更新为由计时器变为OFF时刻开始的逝去的时间值。当当前值(CV)等于设定值(PV)时，功能块停止向右传送电流。当前值不会超过设定值。

当此功能块重新得电，当前值被复位为零。





当OFDT用在不是每次扫描都要被调用的程序块中时，该计时器在调用该程序块期间累计时间，除非它被复位。这就是说，它起到了运行于某个程序中的计时器作用，具有比主程序块中计时器慢得多的计时时间。由于该程序块可能很长时间不被调用，所以必须对计时器进行编程以允许它的catch-up特性。

例如，若在一个程序块中的计时器被复位且该程序在4分钟之间未被调用，当再调用程序块时，这4分钟已被累计上了。这段时间被加入到计时时间中，除非计时器被重新复位。

参数:

参数	说明
地址	OFDT 功能块使用3个连续字（寄存器）%R, %P或%L 内存来存储下面的数据: • 当前值 (CV) = word 1. • 设定值 (PV) = word 2. • 控制字 = word 3. 当你在梯形图中使用OFDT，必须为它指定这三个连续字的地址，写在代表此功能块的图形块的下方。详细信息见4-13页。 注意: 不要再在其他指令中使用这些地址。 小心: 变量重复使用会造成计时器运行的失常。
enable	当使能得电时，计时器的当前值开始累计。
PV	当计时器被复位或被选通时，PV值被写入计时器作为预置值。对于一个寄存器(%R)PV变量，PV参数被指定为此地址参数的第二个字。例如当地址参数为%R00001则%R00002即为PV参数。
Q	当前值小于设定值时，输出Q得电。由于在系统上电时不自动初始化Q的状态，Q的状态将掉电保持。
CV	CV 为计时器当前计时值。

有效的内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	UR	const	none
address									•	•	•			•		
enable	•															
PV	•	•	•	•	•		•		•	•	•	•	•	•	•	•
Q	•															•
CV	•	•	•	•	•		•		•	•	•	•	•	•		•

• 可使电流经功能块的变量或地址。

注意

对参数子程序块的限制，见第二章第二节“参数子程序块形式参数的限制”部分内容。

例:

本例中，当输入信号(%I00001)ON时，OFDT计时器用于关断输出(%Q00001)。输出在输入关断0.3秒后也将关断。



延时接通计时器TMR

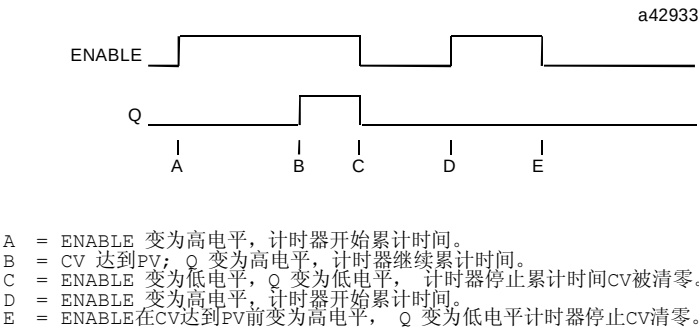
简单延时接通计时器(TMR)当得电时计时值开始累计，在失电时计时值复位为零。计时单位有秒（默认选项），0.1秒，0.01秒。时间单位的取值范围为0到+32,767。计时器的当前值失电保持，在重新得电时不会自动初始化此数据。

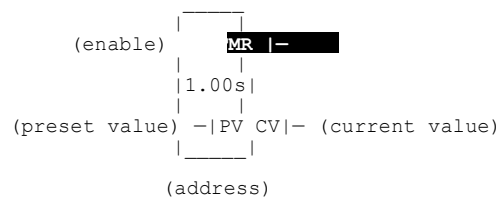
当TMR得电，它开始累计当前值(CV)。当逻辑中使用此功能块失，当前值被刷新为计时器自最近一次复位开始的消逝的时间值。

注意

如果在一次CPU扫描中,多次出现具有相同变量地址的相同的计时器被执行，则这些计时器的当前值是相同的。另外，当前一次扫描时间大于设定值时，TMR将不起作用（向右通电流）。

当使能逻辑保持ON时当前值将一直被刷新。当当前值等于或大于设定值时，功能块向右传送电流。计时器继续累计直到达到最大计时值。当使能参数由ON跳变到OFF时，计时器停止累计时间并将当前值复位为零。





当TMR用在不是每次扫描都要被调用的程序块中时，该计时器在调用该程序块期间累计时间，除非它被复位。这就是说，它起到了运行于某个程序中的计时器作用，具有比主程序块中计时器慢得多的计时时间。由于该程序块可能很长时间不被调用，所以必须对计时器进行编程以允许它的catch-up特性。

例如，若在一个程序块中的计时器被复位且该程序在4分钟之间未被调用，当再调用程序块时，这4分钟已被累计上了。这段时间被加入到计时时间中，除非计时器被重新复位。

参数:

参数	说明
地址	TMR 功能块使用3个连续字（寄存器）%R, %P或%L 内存来存储下面的数据： • 当前值 (CV) = word 1. • 设定值 (PV) = word 2. • 控制字 = word 3. 当你在梯形图中使用TMR，必须为它指定这三个连续字的地址，写在代表此功能块的图形块的下方。详细信息见4-13页。 注意: 不要再在其他指令中使用这些地址。 小心: 变量重复使用会造成计时器运行的失常。
enable	当使能得电时，计时器的当前值开始累计。 TMR 未被使能时，当前值复位为零且Q被关断。
PV	当计时器被复位或被选通时，PV值被写入计时器作为预置值。对于一个寄存器(%R)PV变量，PV参数被指定为此地址参数的第二个字。例如当地址参数为%R00001则%R00002即为PV参数。
Q	当TMR使能且当前值大于或等于设定值时，输出得电。
CV	CV 为计时器当前计时值。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
address									•	•	•			•		
enable	•															
PV	•	•	•	•	•		•		•	•	•	•	•	•	•	•
Q	•															•
CV	•	•	•	•	•		•		•	•	•	•	•	•		•

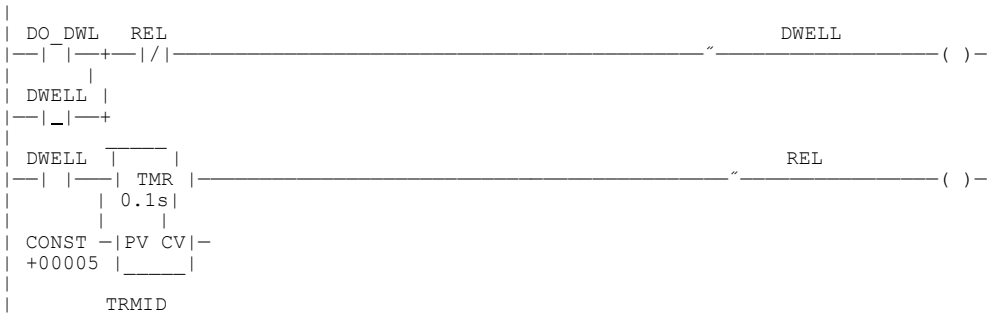
- 可使电流流经功能块的变量或地址。

注意

对参数子程序块的限制，见第二章第二节“参数子程序块形式参数的限制”部分内容。

例:

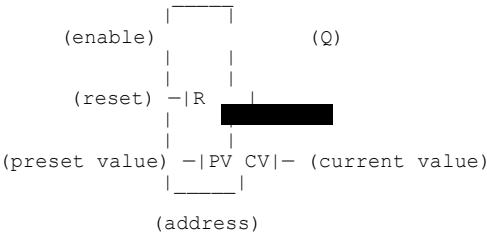
本例中，延时接通计时器TMRID用于控制线圈DWELL得电的时间长度。当常开接点DO_DWL得电时，线圈DWELL也得电。线圈DWELL得接点使线圈DWELL得电，同时开启计时器TMRID。当TMRID达到它的设定值0.5秒时，线圈REL得电，中断线圈DWELL的自锁状态。而接点DWELL中断至TMRID的电流，复位当前值并使线圈失电。于是回路又开始等待接点DO_DWL的下一动作。



增计数器UPCTR

增计数器功能块(UPCTR)用于计数到设定值。计数范围为0到+32,767。当增计数器被置ON，当前值(CV)被复位为零。其使能输入每次从OFF变为ON，当前值便递加1。当前值可以大于设定值，只要当前值大于或等于设定值，输出就为ON。

UPCTR功能块输出 (Q)的状态是掉电保持的，在电源上电时不自动初始化其计数值。



参数:

参数	说明
地址	UPCTR 功能块使用3个连续字（寄存器）%R, %P或%L 内存来存储下面的数据： • 当前值(CV) = word 1. • 设定值(PV) = word 2. • 控制字 = word 3. 当你在梯形图中使用UPCTR，必须为它指定这三个连续字的地址，写在代表此功能块的图形块的下方。详细信息见4-13页。 注意： 不要在其他增计数器，减计数器，或其他指令上再使用这些地址，否则将导致不正确的结果。 小心： 重复使用变量将导致不可预料的运行结果。
enable	对于一个使能的正跳变，当前计数值增1。
R	当R通电，将把当前值复位为零。
PV	当计时器被复位或被选通时，PV值被写入计时器作为预置值。对于一个寄存器(%R)PV变量，PV参数被指定为此地址参数的第二个字。例如当地址参数为%R00001则%R00002即为PV参数。
Q	当当前值大于或等于设定值时，输出Q得电。 (Q)的状态是掉电保持的，在电源上电时不自动初始化其计数值。
CV	CV 为当前计数器的计数值。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
address									.	.	.			.		
enable	.															
R	.															
PV	.	.	.	.	.		.		.	.	.	.	.	.	.	.
Q	.															.
CV	.	.	.	.	.		.		.	.	.	.	.	.		.

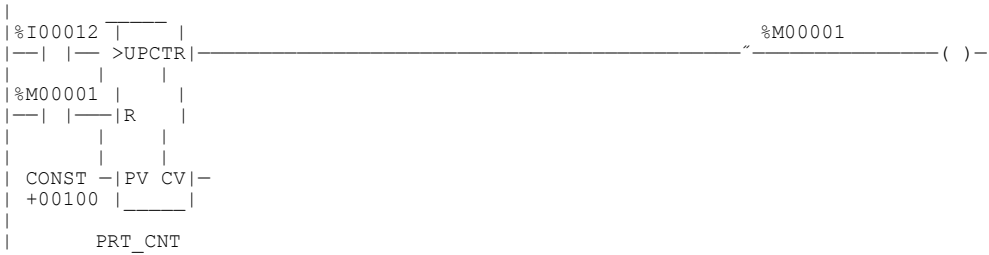
- 可使电流流经功能块的变量或地址。

注意

对参数子程序块的限制，见第二章第二节“参数子程序块形式参数的限制”部分内容。

例:

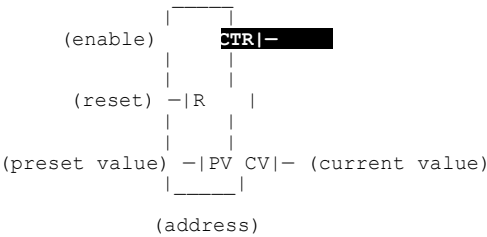
本例中，输入信号%I00012每次从OFF变为ON，增计数器PRT_CNT增加1，当计数值到达100时，内部线圈%M00001得电。当%M00001为ON时，计数值被复位为零。



DNCTR

减计时器功能块(DNCTR)用于从预置置开始减计数为零。预设定值最小为零，最大为+32,767。当前值最小为-32,768。复位时，计数器当前值(CV)被设定为预设定值(PV)。当使能输入由OFF跳变为ON，当前值减1。当前值小于或等于0时，输出为ON。

减计时器功能块(DNCTR)的输出(Q)状态掉电保持，在上电时不自动初始化。



参数:

参数	说明
地址	DNCTR 需使用三个连续字（寄存器）的 %R, %P或%L 来存储下面数据： • 当前值 (CV) = word 1. • 设定值 (PV) = word 2. • 控制字 = word 3. 当你在梯形图中使用DNCTR，必须为它指定这三个连续字的地址，写在代表此功能块的图形块的下方。详细信息见4-13页。 注意： 不要在其他增计数器，增计数器，或其他指令上再使用这些地址，否则将导致不正确的结果。 小心： 重复使用变量将导致不可预料的运行结果。
enable	对于一个使能的正跳变，当前计数值减1。
R	当R通电，将把当前值复位为设定值。
PV	当计时器被复位或被选通时，PV值被写入计时器作为预置值。对于一个寄存器(%R)PV变量，PV参数被指定为此地址参数的第二个字。例如当地址参数为%R00001则%R00002即为PV参数。
Q	当前值小于或等于0时输出Q得电。输出(Q)状态掉电保持，在上电时不自动初始化。
CV	CV 为当前计数器的计数值。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
address									•	•	•			•		
enable	•															
R	•															
PV	•	•	•	•	•		•		•	•	•	•	•	•	•	•
Q	•															•
CV	•	•	•	•	•		•		•	•	•	•	•	•		•

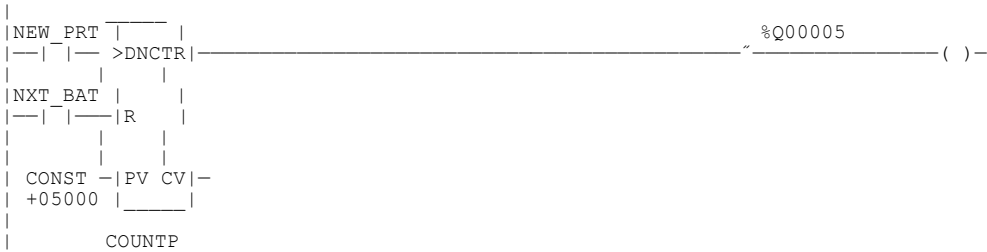
- 可使电流流经功能块的变量或地址。

注意

对参数子程序块的限制，见第二章第二节“参数子程序块形式参数的限制”部分内容。

例:

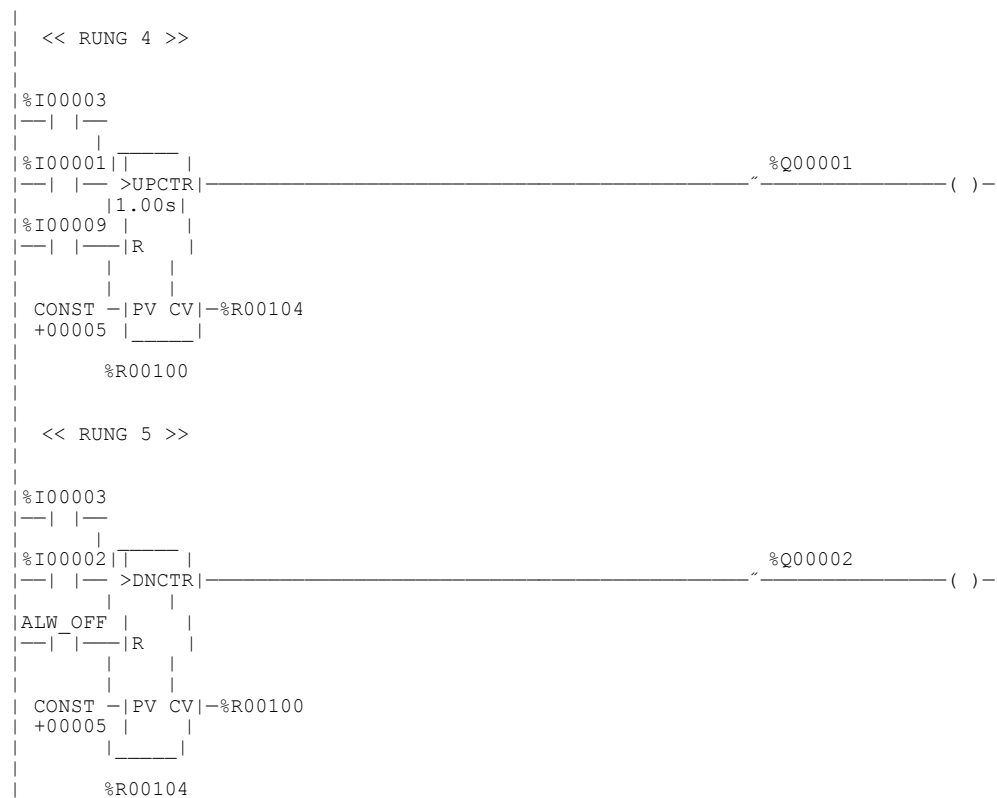
本例中，减计数器COUNTP将对NEW_PRT信号累次5000次跳变后使输出%Q00005得电。



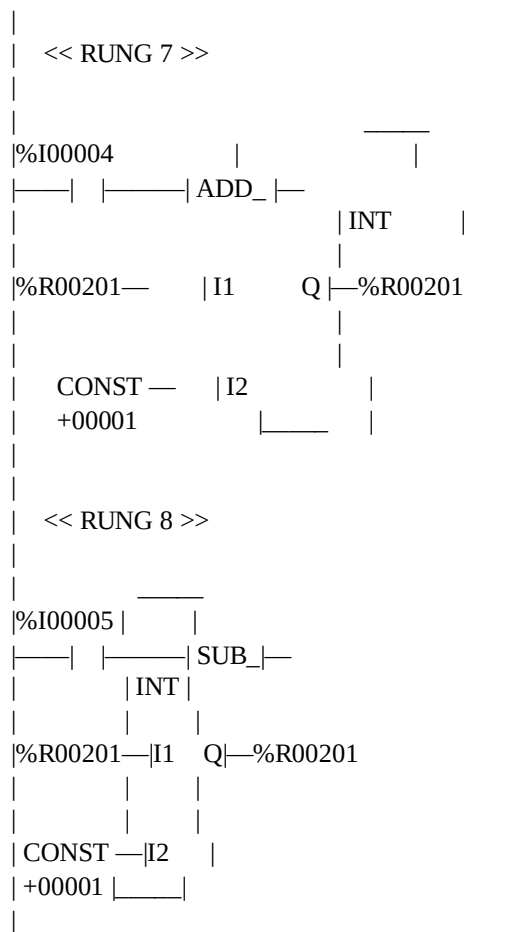
增/减计数器配合使用举例：

本例中，PLC用于跟踪一个临时存储区域的部件的数量。使用90-70指令集有两种方法完成上述功能。

第一种方法是通过使用一组增/减计数器来对共享寄存器或当前值进行累计。当部件进入存储区域，增计数器加1，存储区部件数的当前值增加1；当一个部件离开存储区域，减计数器减1，存储区部件总量减1。为避免同共享寄存器的冲突，两个计数器使用不同的寄存器地址，但它们具有与其他寄存器累计值相同的当前值(CV)地址。这是本应用优先选用的方法。



第二种方法如下所示，使用 ADD 和 SUB 功能块来进行部件跟踪。



Chapter 6

第六章 数学运算功能块

本章介绍了 Logicmaster 90 数学运算功能块：

缩写	功能	说明	页码
ADD	加法	两个数相加。	6-2
SUB	减法	一个数减去另一个数。	6-2
MUL	乘法	两个数相乘。	6-2
DIV	除法	一个数除以另一个数，得商。	6-2
MOD	取余	一个数除以另一个数，得余数。	6-4
SQRT	平方根	求一个整型数或实型数的平方根。	6-6
ABS	绝对值	求一个整型，双精度整型或实型数的绝对值。	6-8
SIN, COS, TAN, ASIN, ACOS, ATAN	三角函数	对输入端IN的实型数进行相应的三角运算。	6-10
LOG, LN EXP, EXPT	对数/指数运算	对输入端IN的实型数进行相应的对数/指数运算。	6-12
RAD, DEG	弧度转换	对输入端IN的实型数进行相应的弧度换算。	6-14

注意

除法和取余是两个类似的功能，区别在于它们的输出，除法的输出是得到的商，而取余的输出是得到的余数。

数学运算MATH (ADD, SUB, MUL, DIV)

数学运算功能块包括加，减，乘，除。当功能块得电时，对输入参数I1和I2进行相应的数学运算。这两个参数的数据类型必须相同。输出Q的数据类型同I1和I2。

注意

DIV运算舍去余数，而不是得到最接近的整数。
(例如, $24 \text{ DIV } 5 = 4$ 。)

数学功能块操作数的类型可为：

数据类型	说明
INT	有符号整型。
UINT	无符号整型。
DINT	双精度有符号整型。
REAL	浮点型。
MIXED	混合型仅用于MUL和DIV功能。

Note

MUL_MIXED 的输入为INT，MUL_MIXED 输出为DINT。 DIV_MIXED i 输入 I1 为 DINT，DIV_MIXED 输入 I2 和输出为INT。

默认的数据类型为有符号整型，可在选择该功能块后对此进行修改。有关数据类型的更多信息参见第二章第三节，“程序的组织与用户数据”部分。

如果对于INT或DINT进行运算的结果溢出，则输出值将会设定为此数据类型能够达到的最大值。对于有符号数，符号指示了相应的溢出方向。如果操作数为带符号的或双精度整型数则MUL和DIV运算的结果的符号取决于I1和I2的符号。如果操作数为UINT的发生溢出，则输出变量被设为该数据类型的最大值。

如果运算结果没有溢出，则OK输出为ON，除非有非法的REAL运算发生：

对于 ADD, $(+\infty) + (-\infty)$

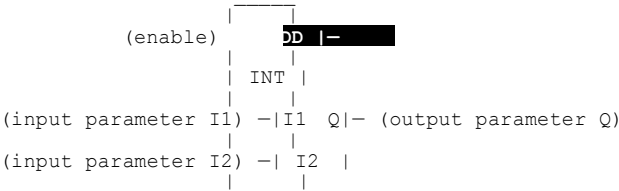
对于 SUB, $(+\infty) + (\pm)$

对于 MUL, $0 \times (\infty)$

对于 DIV, $0 \div 0$

对于 DIV, $00 \div 00$

I1 和/或 I2 为 NaN (不是数字)，这种情况ok被置为 OFF。



参数:

参数	说明
enable	当功能块使能，便开始相应运算。
I1	第一个操作数 I1 为一常数或变量 (I1 在数学运算符左侧，即 I1 - I2。)
I2	第二个操作数 I2 为一常数或变量 (I2在数学运算符右侧，即 I1 - I2。)
ok	未出现被0除时OK得电，除非发生I1 和/或 I2 为 NaN的非法操作。
Q	输出 Q 为运算的结果。

有效的数据类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
I1	•	o	o	o	o		o		•	•	•	•	•	•	•	
I2	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o		o		•	•	•	•	•	•		

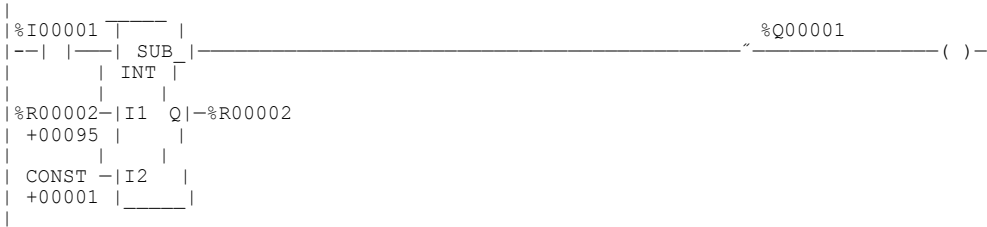
注意: 间接变量适用于所有的寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).
• 对电流可通过的功能块有效。
o 对于UINT 或 INT类型有效; 不适用于 DINT 或 REAL.型变量。

注意

对于参数子程序中使用的限制，见第二章“参数子程序块中形式参数的限制”部分内容。

例:

下例中，当输入%I00001置位，只要减法运算结果不溢出，则%R00002减去1，且线圈%Q00001变为ON。



取余功能块MOD (INT, UINT, DINT)

取余功能块 (MOD)用于两个相同类型的数据相除，来得到它们的余数。如果对于有符号整数或双精度有符号整数进行取余运算时，其结果的符号总是与I1的符号相同。

取余功能块的操作数类型有以下几种：

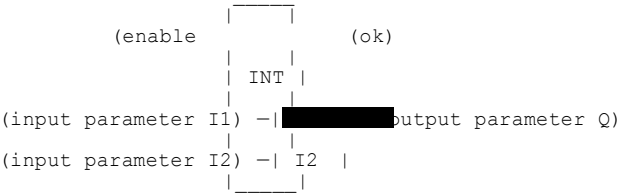
数据类型	说明
INT	带符号整型
DINT	双精度带符号整型
UINT	无符号整型

默认数据类型为带符号整型，然而，可以在选择该功能块后改变它的类型。有关数据类型的详细说明，见第二章第三节“程序组织和用户数据”部分内容。

当此功能块得电，它就用输入I1除输入参数I2。这两个参数必须是相同的数据类型。输出Q由下面的公式计算：

$$Q = I1 - ((I1 \text{ DIV } I2) * I2)$$

这里 DIV 产生一个整型数， Q与输入参数I1和I2具有相同的数据类型。当功能块使能时，OK 总是处在ON状态。除非发生除零错误。此时，OK变为OFF。



参数:

参数	说明
enable	功能块使能，运算才开始。
I1	被除数，是常量或变量。
I2	除数，是常量或变量。
ok	如果运算中无除零错误，则OK为ON。
Q	Q中为I1除以I2所得余数。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
I1	•	o	o	o	o		o		•	•	•	•	•	•	•	
I2	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o		o		•	•	•	•	•	•		

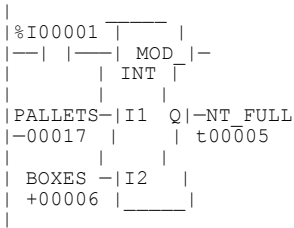
注意: 间接变量对所有寄存器变量有效(%R, %P, %L, %AI, %AQ, 和%UR).
• 可使电流流经功能块的变量或地址。
o 对于UINT 或INT 型数据有效, 而对于 DINT无效。.

注意

对于参数子程序中使用的限制, 见第二章“参数子程序块中形式参数的限制”部分内容。

例:

本例中, 当%I00001为ON时, 整型数PALLETS 与BOXES进行除法运算后所得余数送NT_FULLL 中。



平方根功能块SQRT (INT, DINT, REAL)

平方根功能块(SQRT)用于求一个数的平方根。进行运算时，输出Q为输入IN的平方根的整数部分。输出Q必须与输入IN具有相同的数据类型。

SQRT 功能的操作数类型有：

数据类型	说明
INT DINT REAL	带符号整型. 双精度带符号整型 浮点型

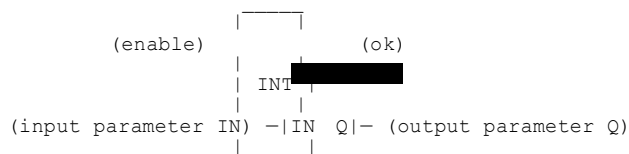
默认数据类型为带符号整型，然而，可以在选择该功能块后改变它的类型。有关数据类型的详细说明，见第二章第三节“程序组织和用户数据”部分内容。

如果功能块运算结果无溢出则OK为ON，除非以下两种非法REAL操作数其中之一存在：

IN < 0。

IN 为 NaN (非数字)。

这种情况ok为 OFF。



参数:

参数	说明
enable	功能块使能，运算才开始。
IN	IN 包含了一个需要进行平方根运算的常数或变量。若IN小于零，则功能块将得电。
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或IN 为 NaN。
Q	Q 包含了对 IN进行平方根运算的结果。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
Il	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o		o		•	•	•	•	•	•	•	

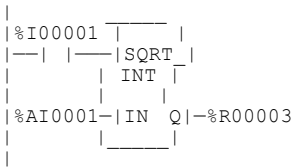
注意: 间接变量对所有寄存器变量有效(%R, %P, %L, %AI, %AQ, 和%UR).
• 可使电流流经功能块的变量或地址。
o 对于UINT 或INT 型数据有效, 而对于 DINT和REAL无效。.

注意

对于参数子程序中使用的限制, 见第二章“参数子程序块中形式参数的限制”部分内容。

例:

本例中, 当%I00001为ON时, 对于存储在%AI0001 中的整型数进行平方根运算的结果送%R00003。



绝对值运算ABS (INT, DINT, REAL)

绝对值功能块 (ABS)用于求一个整型数，或双精度整型数，或浮点数的绝对值。当此功能块运行，将把输入IN的数据取绝对值后送输出Q。

ABS 功能块操作数类型有：

数据类型	说明
INT	带符号整型
DINT	双精度带符号整型
REAL	浮点型

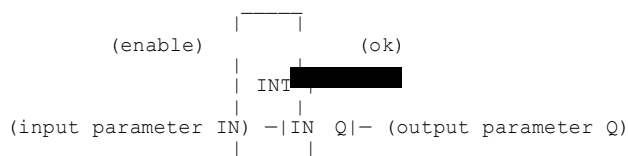
默认数据类型为带符号整型，然而，可以在选择该功能块后改变它的类型。有关数据类型的详细说明，见第二章第三节“程序组织和用户数据”部分内容。

OK输出为ON状态，除非出现下列情况之一：

对于INT 类型， IN为 MININT。

对于DINT类型， IN为 MINDINT。

对于REAL类型， IN为 NaN (非数字)。



参数:

参数	说明
enable	当功能块使能，进行相应的运算。
IN	IN 包含了需要进行运算的整型或实型数。
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或IN 为 NaN。
Q	输出Q包含了IN的绝对值。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o		o		•	•	•	•	•	•	•	

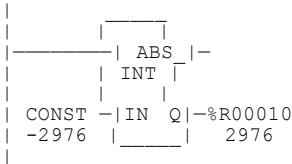
注意: 间接变量对所有寄存器变量有效(%R, %P, %L, %AI, %AQ, 和%UR).
• 可使电流流经功能块的变量或地址。
o 对于UINT 或INT 型数据有效, 而对于 DINT和REAL无效。.

Note

对于参数子程序中使用的限制，见第二章“参数子程序块中形式参数的限制”部分内容。

例:

此例中，对 -2976取绝对值得2976，结果送 %R00010。



三角运算功能块(SIN, COS, TAN, ASIN, ACOS, ATAN)

SIN, COS和TAN功能分别用于求输入数据的正弦值, 余弦值和正切值。一旦功能块被选通, 便计算出单位为弧度的输入量的正弦值, 余弦值或正切值, 并将结果存储于输出Q。IN和Q均为浮点数。

ASIN, ACOS和ATAN功能分别用于求输入数据的反正弦值, 反余弦值和反正切值。一旦功能块被选通, 便计算出单位为弧度的输入量的反正弦值, 反余弦值或反正切值, 并将结果存储于输出Q。IN和Q均为浮点数。

SIN, COS和TAN功能块的输入值范围很大, $-2^{63} < IN < +2^{63}$, ($2^{63} = 9.22 \times 10^{18}$)。

ASIN 和 ACOS功能块的输入值范围很小, $-1 < IN < 1$ 。给定一个IN参数的有效值, ASIN_REAL功能块将求出Q, 使得:

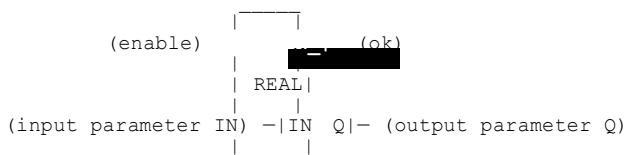
$$\text{ASIN (IN)} = -\frac{\pi}{2} \leq Q \leq \frac{\pi}{2}$$

ACOS_REAL功能块的输出Q为:

$$\text{ACOS (IN)} = \pi$$

ATAN功能块的输入值范围很大, $-\infty \leq IN \leq +\infty$ 给定一个IN参数的有效值, ASIN_REAL功能块将求出Q, 使得:

$$\text{ATAN (IN)} = -\frac{\pi}{2} \leq Q \leq \frac{\pi}{2}$$



注意

这些功能块仅用在支持浮点运算的系列90-70CPU。

参数:

参数	说明
enable	当功能块使能，运行相应的运算。
IN	IN 包含需要进行三角运算的实型数。
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或IN 为 NaN。
Q	Q 包含IN的三角运算结果。

Valid Memory Types:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•								•	•	•	•	•	•	•	
ok	•															•
Q	•								•	•	•	•	•	•		

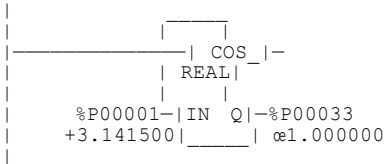
注意: 间接变量适用于所有寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).
• 可使电流流经功能块的变量或地址。

注意

对于参数子程序中使用的限制，见第二章“参数子程序块中形式参数的限制”部分内容。

例:

本例中，%P00001中的COS值送 %P00033。



对数/指数功能(LOG, LN, EXP, EXPT)

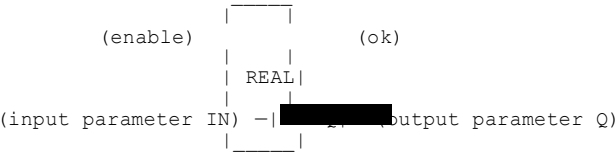
LOG， LN和EXP有两个输入参数和两个输出参数。一旦选通函数，则对输入IN中的实数执行相应的对数/指数运算并将结果送入输出Q中。

对于 LOG功能，对IN求以10为底的对数结果放入Q中。

对于 LN功能，对IN求自然对数，结果放入Q中。

对于 EXP 功能， 计算 e 的IN次幂，结果放入Q中。EXPT功能块有三个输入参数和两个输出参数，一旦选通，将计算I1和I2次幂运算，结果放入到Q中。

ok 端为ON状态，除非IN为负或不是一个数。



参数:

参数	说明
enable	当功能块使能，进行相应的运算。
IN	IN 包含要进行运算的实型数。
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或IN 为 NaN或负数。
Q	输出 Q包含了IN的对数或指数运算结果。

Note

这些功能块仅用在支持浮点运算的系列90-70CPU。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN*	.								.	.	.	.	.	.	.	
ok	.															.
Q	.								.	.	.	.	.	.	.	

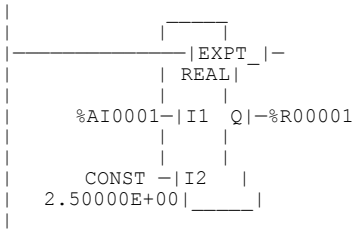
注意: 间接变量适用于所有寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).
* 对于 EXPT功能块, 输入 IN由输入参数 I1 和 I2来代替。
• 可使电流流经功能块的变量或地址。
.

注意

对于参数子程序中使用的限制，见第二章“参数子程序块中形式参数的限制”部分内容。

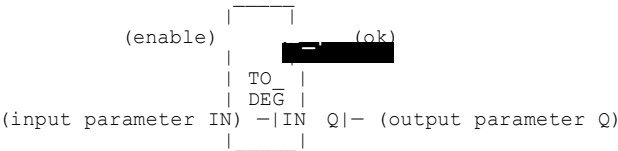
例:

本例中，将%AI0001中的值计算其2.5次幂，结果放入%R00001中。



弧度转换(RAD, DEG)

一旦选通该功能块，将对实数输入IN进行适当的转换运算(RAD_TO_DEG或DEG_TO_RAD)并将结果放入输出Q中。输出OK将得电，除非IN为NaN (非数字)。



参数:

参数	说明
enable	当功能块使能，进行相应的运算。
IN	IN 包含要进行运算的实型数。
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或IN 为 NaN或负数。
Q	输出 Q包含了IN的运算结果。

注意

这些功能块仅用在支持浮点运算的系列90-70CPU。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•								•	•	•	•	•	•	•	
ok	•															•
Q	•								•	•	•	•	•	•		

注意: 间接变量适用于所有寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).

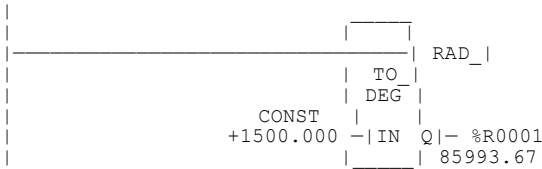
• 可使电流流经功能块的变量或地址。

注意

对于参数子程序中使用的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

Example:

例中，+1500 转换为 DEG 并将结果送 %R0001。



Chapter 7

第七章 关系运算功能块

关系运算功能块用于比较两个数字。本章将介绍如下关系功能块：

缩写	功能	说明	页码
EQ	等于	比较两个数是否相等。	7-2
NE	不等于	比较两个数是否不等。	7-2
GT	大于	比较其中一个数是否大于另一个数。	7-2
GE	大于等于	比较其中一个数是否大于或等于另一个数。	7-2
LT	小于	比较其中一个数是否小于另一个数。	7-2
LE	小于等于	比较其中一个数是否小于或等于另一个数。	7-2
CMP	比较	比较其中一个数是否小于或等于或大于另一个数。	7-4
RANGE	范围	确定一个数是否在某个范围。	7-6

EQ, NE, GT, GE, LT和 LE (INT, UINT, DINT, REAL)

关系功能块用于确定两个值之间的关系。当接通后，它比较输入I1和I2参数。二者必须为同一数据类型。

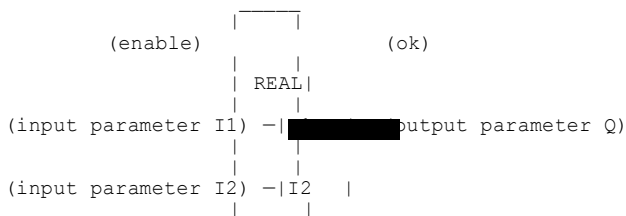
关系功能块的操作数类型有

数据类型	说明
INT	带符号整型
UINT	无符号整型
DINT	双精度带符号整型
REAL	浮点型

默认的数据类型为带符号整型。可以比较的数据类型有，带符号整型，无符号整型，双精度带符号整型；在选择关系功能块后在选择新的数据类型就可以。比较两个不同类型的数据，必须首先用正确的转换功能块（见第十二章，转换功能块）将它们转换为相同的整型数。

若输入参数 I1 和 I2 与确定的函数关系匹配，则输出Q接通被置为ON (1)，否则将设为OFF (0)。

输出OK在该功能使能时一直为ON，除非I1和/或I2为NaN（非数字）。



参数:

参数	说明
enable	当功能块使能，进行相应的运算。
I1	I1 包含了用于比较的第一个常数或变量值。 (I1在关系符号的左侧，如I1 < I2)
I2	I2包含了用于比较的第二个常数或变量值。 (I2在关系符号的右侧，如I1 < I2)
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或 IN 为 NaN。
Q	输出 Q当I1和I2符号特定的关系时接通。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
I1	•	o	o	o	o		o		•	•	•	•	•	•	•	
I2	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o		o		•	•	•	•	•	•		

注意: 间接变量适用于所有寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).
 0 可使电流流经功能块的变量或地址。
 对于 INT 或 UINT 类型数据有效; 但对 DINT 或 REAL 无效。

注意

对于参数子程序中使用的限制,见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例：

此例中，两个双精度带符号整型数PRT_MDE 和 BIN_FUL当%I00001接通时进行比较。若PRT_MDE小于 BIN_FUL，则线圈%Q00002得电接通。



CMP (INT, UINT, DINT, REAL)

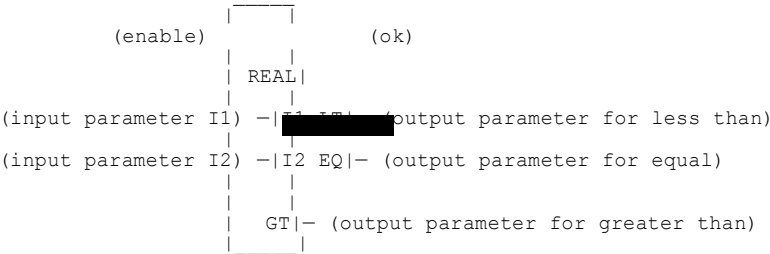
使用比较功能块 (CMP) 来比较一个数是小于，等于还是大于另一个数。当功能块被选通，它将比较I1和I2的值。这两个值须为同数据类型的数。

The CMP 功能块操作数类型有：

数据类型	说明
INT	带符号整型.
UINT	无符号整型
DINT	双精度带符号整型
REAL	浮点型

默认的数据类型为带符号整型。可以比较的数据类型有，带符号整型，无符号整型，双精度带符号整型；在选择关系功能块后在选择新的数据类型就可以。比较两个不同类型的数据，必须首先用正确的转换功能块（见第十二章，转换功能块）将它们转换为相同的整型数。
LT， EQ和GT作为比较的结果被置为 ON (1)或 OFF (0) 。

输出OK在该功能使能时一直为ON，除非I1和/或I2为NaN（非数字）。



参数:

参数	说明
enable	当功能块使能，进行相应的运算。
I1	I1 包含了用于比较的第一个常数或变量值。
I2	I2 包含了用于比较的第二个常数或变量值。
ok	如果功能块运算结果无溢出则OK为ON，除非出现非法的操作数或 IN 为 NaN。
LT	当I1小于I2时输出LT接通。
EQ	当I1等于I2时输出EQ接通。
GT	当I1大于I2时输出GT接通。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
I1	•	o	o	o	o		o		•	•	•	•	•	•	•	
I2	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
LT	•															•
EQ	•															•
GT	•															•

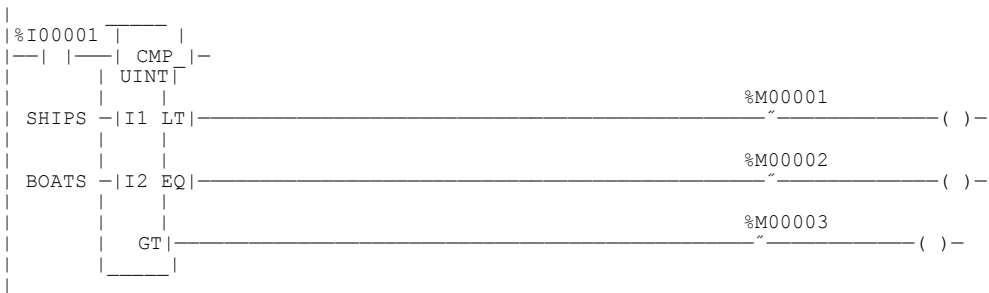
注意: 间接变量适用于所有寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).
o 可使电流流经功能块的变量或地址。
对于 INT 或 UINT 类型数据有效; 但对 DINT 或 REAL 无效。.

注意

对于参数子程序中使用的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

此例中，当%I00001为 ON时，整型变量SHIPS与变量BOATS进行比较，中间线圈%M0001, %M0002和 %M0003 根据比较所得结果来接通或断开。



RANGE (INT, UINT, DINT, WORD, DWORD)

RANGE 功能块用于比较一个单独的输入的值与两个限幅值进行比较以确定该输入变量是否落在限制范围内。

注意

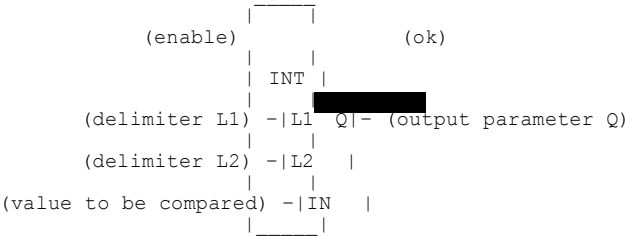
该功能块仅对5.0版或更新版的CPU可用。

RANGE 功能块操作数类型有：

数据类型	说明
INT	带符号整型
DINT	双精度带符号整型
UINT	无符号整型
WORD	字数据类型
DWORD	双字数据类型

默认的数据类型为带符号整型。可以比较的数据类型有，带符号整型，无符号整型，双精度带符号整型；在选择关系功能块后在选择新的数据类型就可以。比较两个不同类型的数据，必须首先用正确的转换功能块（见第十二章，转换功能块）将它们转换为相同的整型数。

当功能块使能，RANGE功能块将输入参数IN中的值与L1和L2中的两个限制值进行比较。当IN的值在L1与L2之间时，输出Q为ON。否则Q为OFF。如果运算成功，则OK将得电。



注意

限幅参数L1和L2代表了一个数值范围的端点。它们没有固定的大小或是高低关系，因此，一个要求0到100范围的条件可以把L1赋值为0，L2赋值为100或者把L1赋值为100，L2赋值为0。无论上面哪种情况，输入值为45时，都将使输出Q为ON。

参数:

参数	说明
enable	当功能块使能，进行相应的运算。
L1	L1 包含了该范围的起点。
L2	L2 包含了该范围的终点。
IN	IN包含了同L1和L2进行比较的输入值。
ok	ok 输出接通除非有错误发生。
Q	当输入IN的值在L1和L2范围内时，输出 Q得电。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
L1		o	o	o	o		o		•	•	•	•	•		•†	
L2		o	o	o	o		o		•	•	•	•	•		•†	
IN		o	o	o	o		o		•	•	•	•	•			
ok		o	o	o	o		o		•	•	•	•	•			
Q	•															•

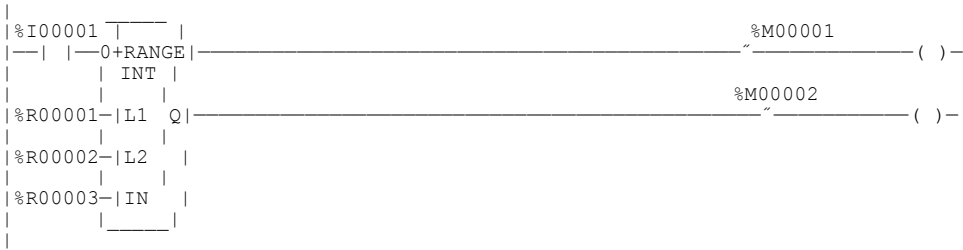
注意: 间接变量适用于所有寄存器类型 (%R, %P, %L, %AI, %AQ, and %UR).
o 可使电流流经功能块的变量或地址。
† 对于INT, WORD或 UINT 数据类型有效, 但对 DINT 或 DWORD无效。.
常数仅可以是整数或双精度带符号整数。

注
意

对于参数子程序中使用的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

此例中，%R00001中的值为10，%R00002中的值为50。当输入%R00003中的值为大于等于10并且小于等于50时，OK输出为ON。



Chapter 8

第八章 位运算功能块

位运算功能块完成对位串的比较，逻辑和移动运算。最大位串长度为256个字或双字。为操作功能块需要WORD或DWORD型数据，默认数据类型为WORD。B

尽管数据必须为16位或32位的，但是实际上位运算功能块对这些数据进行运算时是以连续位串形式实现的。第一个字的第一位称为最小有效位(LSB)，最后一个字的最后一位称为最大有效位(MSB)。例如，你声明了从%L00100开始的三个字，则位运算将以48个连续位的形式对它进行操作。

%L00100	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	← bit 1 (LSB)
%L00101	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	
%L00102	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	

↑
(MSB)

注意

在多次功能中出现重复使用的地址将会导致不确定的结果。

本章介绍以下位运算功能：

缩写	功能	说明	页码
AND	逻辑与	如果位串I1同位串I2对应位均为1，则输出位串的对 应位也为1。	8-3
OR	逻辑或	如果位串I1同位串I2对应位均为1或有一个为1，则 输出位串的对位也为1。	8-3
XOR	逻辑异或	如果位串I1同位串I2对应位不同，则输出位 串的对位也为1。	8-5
NOT	逻辑非	使输出位串Q的各个位与输入串的各个位相反。	8-7
SHL	左移	使输入位串的各个位同时左移指定的位数。	8-9
SHR	右移	使输入位串的各个位同时右移指定的位数。	8-9
ROL	循环左移	使输入位串的各个位循环左移指定的位数。	8-12
ROR	循环右移	使输入位串的各个位循环左移指定的位数。	8-12
BTST	位测试	测试某位当前值是1还是0。	8-14
BSET	位置位	将位串中的某位置位。	8-16
BCLR	位清除	将位串中的某位置位。	8-16
BPOS	位定位	用于确定1在位串中的位置。	8-18
MCMP	位屏蔽比较	用于比较位串1中的位与位串2中的相应位。	8-20

注意

对于所有的位运算，功能块的不明确位类型的位组将会影响写入的字节，字或
双字中所有位的状态改变。请仔细阅读第二个例子以更好的了解此特点。

与和或(WORD, DWORD)

一旦被选通，AND或OR功能块开始从位串I1和位串I2的最小有效位开始，对两个位串中的每一位进行比较。

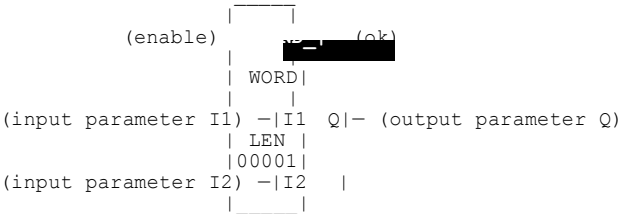
对于AND功能检测的每两位如果同时为1，那么输出位串Q的相应位也为1，如果两位之中有一为0或者都为0，则Q的相应为也置0。

AND功能块可用来建立屏蔽或隔离，此时只有某些位通过，而其他所有位被置0。这一功能可用来把内存中指定区域的值清零，即把这些位与所有位均为零的位串作逻辑与运算就可以了。I1和I2可以重叠使用。

对于OR功能块，它检测的每两位数，如果其中之一或两个都为1，则位串Q的相应位也为1，如果均为0，则Q相应的为也为0。

OR功能块可用于对位串进行组合，可通过使用一个简单的逻辑结构来实现对几个输出的控制。此功能等同于两个并行继电器触点的作用。它可用来实现直接用输入状态驱动指示灯或给状态灯添加闪烁的功能。

对上述两种功能块，其操作数的长度最多为256字或双字。当功能块选通，它就向右传递电流。



参数:

参数	说明
enable	当功能块使能，开始进行相应的运算。
I1	I1 包含了第一个字的常数或变量。
I2	I2 包含了第二个字的常数或变量。
ok	使能时，OK总为接通状态。
Q	输出 Q 包含了运算的结果。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
I1	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
I2	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o	o	o		•	•	•	•	•	•		

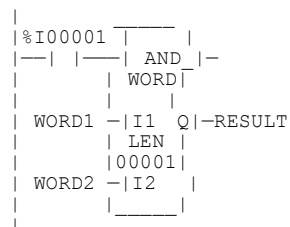
注意:	对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
•	可使电流流经功能块的变量或地址.
+	对于WORD 数据类型适用, 但不可用 DWORD.
o	%SA, %SB, %SC 可用, %S 不能使用.

注意

对于在参数子程序中使用的限制，见第二章第二节，“参数子程序块中形式参数的使用限制”部分内容。

例：

此例中，一旦%I00001置位，则检测由WORD1和WORD2表示的两个16位串。逻辑与运算后的结果被放到输出RESULT中。



WORD1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
WORD2	1	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1
RESULT	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0

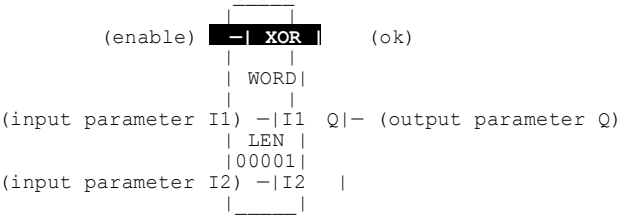
XOR (WORD, DWORD)

异或功能块 (XOR)用于比较位串I1和位串I2的对应位。如果两个位不同，则输出位串的相应位置1。

一旦被选通，AND或OR功能块开始从位串I1和位串I2的最小有效位开始，对两个位串中的每一位进行比较。如果只有其中之一为1，则输出Q的相应位为1。位串最长为256个字或双字。当功能块选通，它就向右传递电流。

如果位串I2与输出串Q在相同的地址开始，则当位串I1某位为1时，将使得I2中的对应位的状态在0和1之间交替变化，只要该功能块使能，则在每个扫描周期中都会改变状态。以双倍于所需的闪烁频率向此功能块施加脉冲电流的方法编程可得到更长的循环周期，电流脉冲周期应与一次扫描的时间一样。（脉冲型线圈或自复位式计时器。）

XOR功能块可用于对两个位串进行快速比较，或者用于实现每两次扫描，一组位以一个ON状态频率进行闪烁。



参数:

参数	说明
enable	当功能块使能，开始进行相应的运算。
I1	I1包含了第一个字的常数或变量。
I2	I2包含了第二个字的常数或变量。
ok	使能时，OK总为接通状态。
Q	输出 Q 包含了运算的结果。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
I1	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
I2	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o	o†	o		•	•	•	•	•	•	•	

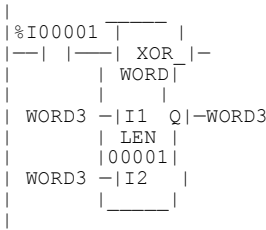
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流流经功能块的变量或地址。
o 对于WORD 数据类型适用, 但不可用 DWORD.
† %SA, %SB, %SC 可用; %S不能使用。.

注意

对于在参数子程序中使用的限制, 见第二章第二节, “参数子程序块中形式参数的使用限制” 部分内容。

例:

本例中, 当输入 %I00001置位, 别名为WORD3 的16位位串将被清零。

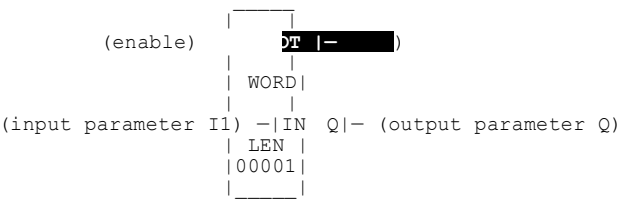


I1 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
I2 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
Q (WORD3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOT (WORD, DWORD)

NOT功能块用于对输入位串I1按位取反后得到输出位串Q。

在该功能块每次扫描通电时，所有位都改变其状态，使输出为I1的逻辑补码。它可处理的长度范围为1到256字或双字。功能块使能它就向右传送电流。



参数:

参数	说明
enable	当功能块使能，开始进行相应的运算。
I1	I1 包含了要被取反的变量。
ok	当使能得电，OK总是接通的。
Q	输出 Q 包含了I1的非。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
Il	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o	o	o		•	•	•	•	•	•	•	

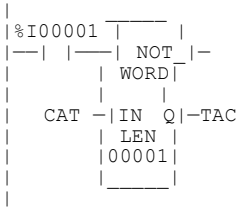
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流流经功能块的变量或地址。
o 对于WORD 数据类型适用, 但不可用 DWORD.
† %SA, %SB, %SC 可用; %S不能使用。.

注意

对于在参数子程序中使用的限制, 见第二章第二节, “参数子程序块中形式参数的使用限制” 部分内容。

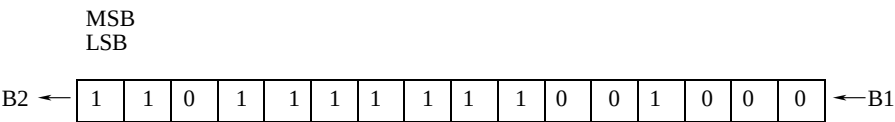
例:

本例中, 当输入 %I00001置位, 别名为TAC的位串为位串CAT的取反。

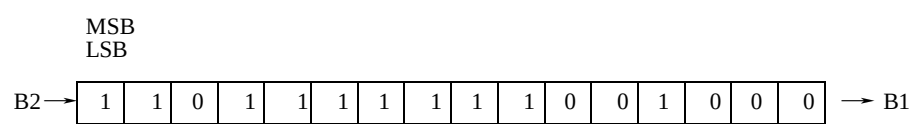


SHL 和 SHR (WORD, DWORD)

左移功能块(SHL) 用于对一个字或一组字的所有位向左移动一定位数。移动时，一定数目的位从位串的左端移出，同时相应的位被移入串的右端。



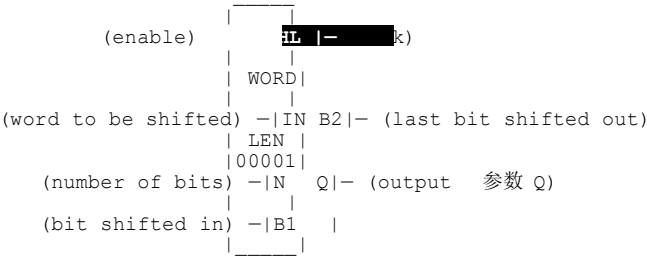
右移功能块(SRL) 用于对一个字或一组字的所有位向右移动一定位数。移动时，一定数目的位从位串的右端移出，同时相应的位被移入串的左端。



这两个功能块都可处理1到256字或双字的位串。
移动的位数要在0与该位串的长度之间，否则，该功能不能执行，而且与不通过电流。

通过输入参数B1可声明要移入位串起始部分的位。如果一个大于1的长度被声明为要被移动的位数，那么这些位的每一位填入相同的值（0或1）。这可为：

- 另一个程序的布尔输出。
 - 全部是1。为此，须用特殊变量ALW_ON 作为输入B1的允许条件。
 - 全部是0。为此，须用特殊变量ALW_OFF 作为输入B1的允许条件。SHL 或 SHR功能块向右传递电流，除非声明移动的位数大于位串的总长度或等于零。
- 输出 Q 是输入串的移动后的拷贝。如果想将输入串本身移位，则输出参数Q必须使用与输入参数IN一样的内存地址。每次扫描只要功能块使能，则整个移位串就被写入。例如，如果移动了四位，则B2将是第四位要移出的。



参数:

参数	说明
enable	当功能块使能，开始进行相应的移位。
IN	IN包含了第一个要进行移位操作的数。
N	需要移动的位数。
B1	B1 将被移入位串的位的值。
B2	B2 将被移出的位的值。
ok	当移位功能块使能并且移动长度不超过总位数时，OK接通。
Q	经过移位的位串的第一个字。
LEN	LEN 为需要移动的总的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
N	•	•	•	•	•		•		•	•	•	•	•	•	•	
B1	•															
B2	•															•
ok	•															•
Q	•	o	o	o	o	o†	o		•	•	•	•	•	•	•	

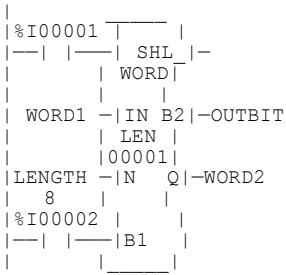
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流流经功能块的变量或地址。
o 对于WORD 数据类型适用，但不可用 DWORD.
† %SA, %SB, %SC 可用; %S不能使用。.

注意

对于在参数子程序中使用的限制，见第二章第二节，“参数子程序块中形式参数的使用限制” 部分内容。

例:

本例中， 当输入 %I00001置位，则别名为WORD2的输出位串是输入位串WORD1移动LENGTH位后的拷贝，并在输出串（WORD2）的起始部分位被置成%I00002中的值。



ROL 和 ROR (WORD, DWORD)

循环左移(ROL)用于对位串的所有位循环向左移动一定位数。从输入位串左边移出给定位数的位从右端移入该位串。

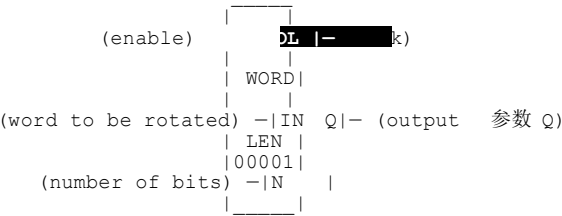
循环右移(ROR)用于对位串的所有位循环向右移动一定位数。从输入位串右边移出给定位数的位从左端移入该位串。

它可处理的长度范围为1到256字或双字。

移动的位数要在0与该位串的长度之间，否则，该功能不能执行，而且与不通过电流。

ROL 或 ROR功能块向右传递电流，除非声明移动的位数大于位串的总长度或等于零。

结果放在输出位串Q中。如果要对输入串本身进行循环移位，那么输出参数Q的地址须与输入参数IN的地址相同。在每次该功能使能的扫描中，整个循环移位后的位串将写入。



参数:

参数	说明
enable	当功能使能，循环移位进行。
IN	IN 要被移位的第一个字。
N	N被移串的位的数量。
ok	当移位功能块使能并且移动长度不超过总位数时，OK接通。
Q	输出移动后的第一个字。
LEN	LEN为被移动字的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
N	•	•	•	•	•		•		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o	o†	o		•	•	•	•	•	•		

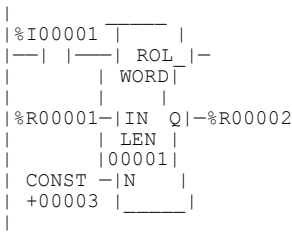
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流经功能块的变量或地址。
o 对于WORD 数据类型适用, 但不可用 DWORD.
† %SA, %SB, %SC 可用; %S不能使用。.

注意

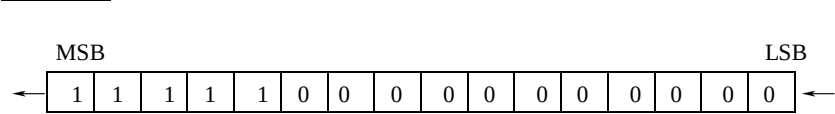
对于在参数子程序中使用的限制, 见第二章第二节, “参数子程序块中形式参数的使用限制” 部分内容。

例:

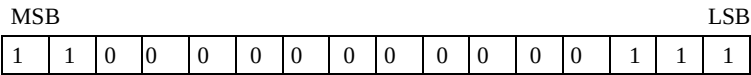
本例中, 当输入 %I00001置位, 输入位串%R00000将被循环左移3位, 并将结果放在%R00002中。该功能块执行后, 输入位串%R00001并没有改变。如果IN和Q使用相同的变量, 则在此将发生循环的移位。



%R00001:



%R00002 (after %I00001 is set):

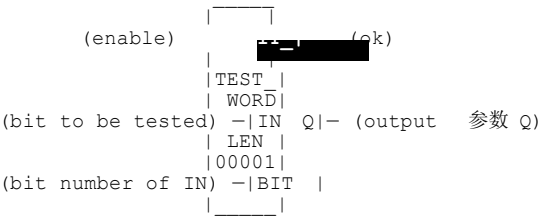


BTST (WORD, DWORD)

位测试 (BTST)功能用于对位串的某一位进行测试，以得知当前位是1还是0。测试结果将放入Q中。

每次扫描时，BTST的输出Q将与指定的状态相同。如果使用一个寄存器而不是用一个常数来声明位的数目，则同是一个功能块在下次扫描时可以测试与这次不同的位。如果参数BIT的值超出范围($1 \leq \text{BIT} \leq (16 * \text{LEN})$)，那么输出Q为OFF。

它可处理的长度范围为1到256字或双字。



参数:

参数	说明
enable	当功能使能，循环移位进行。
IN	IN 要被操作数据的第一个字。
BIT	BIT 要被测试的IN中的位数。有效的范围为($1 \leq \text{BIT} \leq (16 * \text{LEN})$)。
ok	当移位功能块使能并且BIT长度不超过总位数时，OK接通。
Q	当被测试位为1时接通。
LEN	LEN 被测试的数据的字的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	o	o	o	o	o	o		•	•	•	•	•	•	•	
BIT	•	•	•	•	•		•		•	•	•	•	•	•	•	
ok	•															•
Q	•															

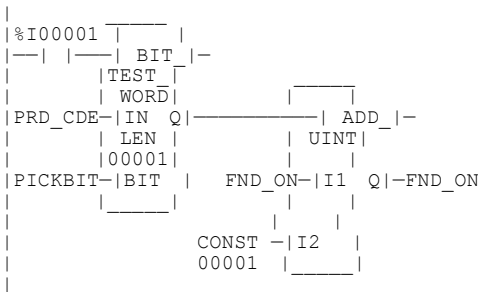
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流流经功能块的变量或地址。
o 对于WORD 数据类型适用，但不可用 DWORD.

注意

对于在参数子程序中使用的限制，见第二章第二节，“参数子程序块中形式参数的使用限制” 部分内容。

例:

本例中， 当输入%I00001置位， 包含在变量PICKBIT中指定位置处的位将被测试。该位是位串PRD_CDE的一部分。如果为1， 则输出Q向ADD功能块传递电流， 使ADD功能块的输入I1加1。

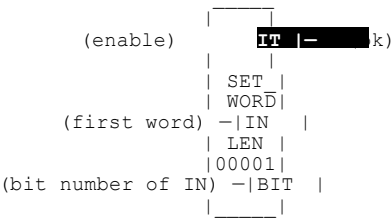


BSET 和 BCLR (WORD, DWORD)

位置位功能 (BSET) 用于将位串中的某位置位。位清除功能(BCLR)用于将位串中的某位清零。

每一次功能块使能，BSET将指定位置位，而BCLR将指定位清零。如果使用变量来代替常数的给定位数，则在下次扫描时对不同的位进行置位和清零。

它可处理的长度范围为1到256字或双字。如果参数BIT的值超出范围($1 \leq \text{BIT} \leq (16 * \text{LEN})$)，那么输出Q为OFF。



参数:

参数	说明
enable	当功能使能，循环移位进行。
IN	要被操作数据的第一个字。
BIT	要被测试的IN中的位数。有效的范围为($1 \leq \text{BIT} \leq (16 * \text{LEN})$)。
ok	当移位功能块使能时，OK接通。
LEN	LEN 被测试的数据的字的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.	o	o	o	o	o†	o		.	.	.	.	.	.	.	
BIT	.	.	.	.	.		.		.	.	.	.	.	.	.	
ok	.															.

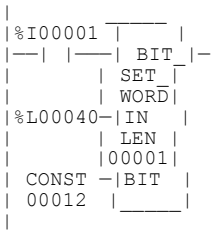
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流流经功能块的变量或地址。
o 对于WORD 数据类型适用, 但不可用 DWORD.
† 对于%SA, %SB, %SC可用; %S 不能使用。

注意

对于在参数子程序中使用的限制, 见第二章第二节, “参数子程序块中形式参数的使用限制” 部分内容。同时, 需注意对于所有的位运算, 当功能块的位组为不明确的位类型时, 将会引起所有写入的字节, 字和双字的位的改变(线圈和接点)。请看下面的第二个例子极其说明, 它说明了对对字节和字块使用位运算之前需了解的内容。

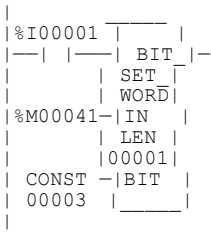
例 1:

本例中, 当 %I00001 置位, 从%L00040开始的第12位被置1。



例 2:

本例中, M41–M48将被解析为一个写入的跳变状态。这些位不可以如期望的那样当作一个跳变接点和线圈来使用。如果希望使用位 Op 功能在跳变功能的联合中, 位Op功能需要是BIT型。



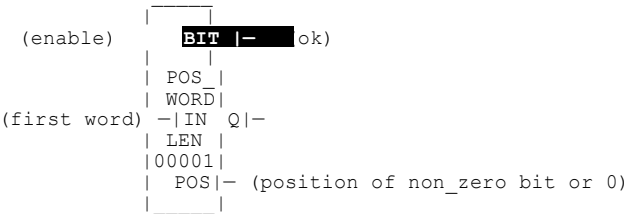
BPOS (WORD, DWORD)

位定位 (BPOS)功能用于在一个位串中查找为1的位。

每次扫描，该功能块对从IN开始的位串进行查找。当该功能块停止查找时，表明已经找到某位为1或已找遍整个位串。

参数POS存储位串中第一个非零位的位置，如果没有找到，则POS被置0。

位串长度可以为1到256字。只要使能ON，则功能块就向右传递电流。



参数:

参数	说明
enable	当功能使能，位查找进行。
IN	要被操作的数据的第一个字。
ok	当移位功能块使能时，OK接通。
POS	位串中第一个非零位的位置，如果没有找到，则POS被置0。
Q	当找到为1的位输出 Q得电。
LEN	LEN 被测试的数据的字的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.	o	o	o	o	o†	o		.	.	.	.	.	.	.	.
POS	.	.	.	.	.		.		.	.	.	.	.	.	.	.
ok	.															.
Q	.															.

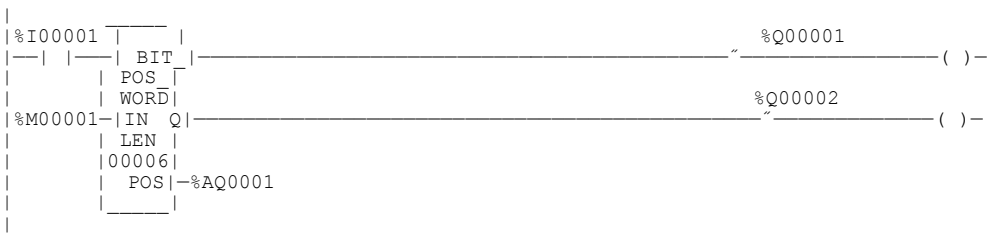
注意: 对于任何寄存器类型的间接变量均可用 (%R, %P, %L, %AI, %AQ, 和 %UR).
• 可使电流流经功能块的变量或地址。
o 对于WORD 数据类型适用, 但不可用 DWORD.

注意

对于在参数子程序中使用的限制, 见第二章第二节, “参数子程序块中形式参数的使用限制” 部分内容。

例:

本例中, 若 %I00001置位, 对开始于 %M00001的位串进行搜索, 直到6个字全部扫描完毕或找到第一个非0位。线圈%Q00001接通。如果发现等于1的位, 则其在位串中的位置被记入 %AQ0001 同时 %Q00002接通。如果%M00001为0, 则向%AQ0001中写入的是2。



MCMP (WORD, DWORD)

屏蔽比较功能块(MCMP)用于比较两个位串中的内容。输入位串I1可包含输出的状态，如线圈或电机启动器。输入位串I2可包含输入状态反馈，例如限位开关或接点。

每次扫描得到电流时，该功能块开始将第一个位串中的位与第二个位串中的对应位进行比较，直到发现不同状态的位或完成对整个位串的比较。

输入BIT用于存储下一次比较的起始位的位数。通常，它与上次发生比较不一致的位的位数相同。由于上次不一致位号被存储在输出BN中，故BIT和BN可以使用同一个变量。比较实际是从BIT后的第1位开始的，因此，BIT的初始值应该比要比较的第一位小1（如，要从%I00001开始比较BIT应设为0）。

如果希望下次比较从位串的其他位开始，可以为BIT和BN指定不同的变量。如果BIT中的值超出了位串的末端，那么BIT在下次比较开始前会被复位到输入位串的起始值。

功能块使能就会向右传送电流。该功能块的其他输出取决于对应屏蔽位的状态，如下所述：

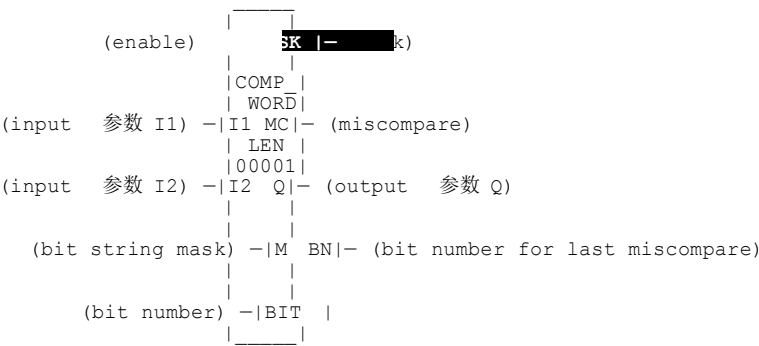
如果 I1 和 I2 的所有位均相同

如果 I1 和 I2 的所有对应位均相同，则功能块将“不匹配”输出MC置为0，并将BN置为输入串中的最高位数，然后比较停止。在进行下一次的MCMP时，它将被复位到1。

如果发现不一致的位

当两个被比较的位不同时，功能块将检查串M（屏蔽字）中对应序号的位，如果该屏蔽位为1，则继续进行下一位的比较，直到发现另一个不一致的位或比较完成。如果检测到的不一致位的屏蔽位为0，则功能块将：

1. 置相应屏蔽位为1。
2. 置MC输出为1。
3. 更新输出位串Q 使之与新的屏蔽串M相匹配。
4. 将位数输出BN置位不匹配的位的位号(如, 6)。
5. 停止比较。



参数:

参数	说明
enable	功能块选通逻辑。
I1	被比较的第一个位串变量。
I2	被比较的第二个位串变量。
M	屏蔽位串变量。
BIT	下一次开始比较的位号变量。比较将从BIT后的第一位开始。有关此参数的详细解释见上一页第三段。
ok	使能时OK为ON。
MC	用户逻辑确定是否出现不匹配位。
Q	屏蔽位串的拷贝输出。
BN	最近一次出现不匹配的位号。
LEN	LEN 位串中含有字的个数。

Chapter 9

第九章 数据移动功能块

数据移动功能块提供基本的数据移动功能。本章介绍如下一些数据移动功能块：

缩写	功能	说明	页码
MOVE	移动	以单独的位拷贝数据。MOVE_BIT功能最长为256位，其他功能块的最大长度数为32767。数据未经类型转换就可移入不同类型的变量中。	9-2
BLKMOV	块移动	将7个常数组成的块拷贝入指定的内存区域，常数为功能块的输入部分。	9-4
BLKCLR	块清除	将数据块中的内容清零。该功能块可用于清除位(%I, %Q, %M, %G, %T)或字(%R, %P, %L, %AI, %AQ)。最长为256字。	9-6
SHFR	移位寄存器	向表中移入一个或多个数据字。最长为256字。	9-8
BITSEQ	位序列	通过位的数组来完成位的移动。最长为256字。	9-11
SWAP	交换	交换一个字中的两个字节或一个双字中的两个字的位置。最长为256字。	9-15
COMMREQ	通讯请求	允许程序同智能模板进行通讯，如，总线控制器，可编程协处理器模板，或子网模板。	9-17
VMERD	VME 读	从VME总线读数据。长度数最大为32,767字。	9-24
VMEWRT	VME 写	向VME总线写数据。长度数最大为32,767字。	9-26
VMERMW	VME 读/改/写	使用VME总线上读/改/写循环来更新数据。	9-28
VMETST	VME 测试和设定	手动设置VME总线上的指示灯。	9-30
VME_CFG_RD	VME 读配置	读 VME模板的配置。	9-33
VME_CFG_WRT	VME 写配置	写 VME模板的配置。	9-36
DATA_INIT	数据初始化	向指定区域拷贝一个常量数据块。	9-39
DATA_INIT_COMM	数据初始化通讯请求	通过一个常量数据块初始化 COMMREQ 功能块。长度需要与整个功能块的命令块长度相等。	9-42
DATA_INIT_ASCII	数据初始化 ASCII	向给定范围拷贝一个常量ASCII文本块。长度须为偶数。	9-45

MOVE (INT, UINT, DINT, BIT, WORD, DWORD, REAL)

使用 MOVE功能块从某处拷贝数据（按位）到另一处。由于数据是以位的形式拷贝的，故目的地址的数据类型不必与源数据类型相同。

MOV功能块有两个输入参数和两个输出参数。当功能块被选通，它将输入参数IN按位拷贝到输出Q。如果数据是从离散内存中的一处移动至另一处（如从%I移到%T），那么与此离散内存相关联的跳变信息也将被拷贝到新地址，输入的数据不会改变，除非源地址与目标地址有重叠。

是需要被移动的输入IN即可以是变量也可以是常数。如果为常数，那么该常数将被放入由输出变量指定的地址。例如，如果IN指定为常数4，则4被放入由Q指定的内存空间。如果输入为常数并且需要移动的长度大于1，则该常数将被放入从Q开始的指定长度范围内的每个元素中。例如，IN为常数9且长度为4，则9将被放入Q及其后三个顺序地址处。

LEN 操作数有以下几种使用的场合：

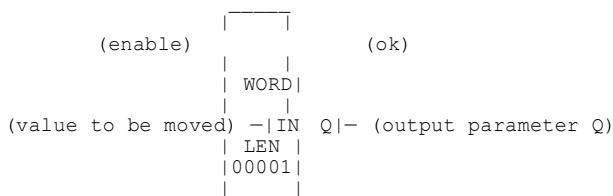
对于 MOVE_INT, MOVE_UINT和 MOVE_WORD为所移动的字数。

对于MOVE_DINT 和 MOVE_DWORD为所移动的字数。

对于 MOVE_BIT为所需移动的位。

对于 MOVE_REAL为所需移动的实型数的数量。

一旦该功能块选通，即向右传送电流。



参数

参数	说明
enable	当功能块使能，进行移动操作。
IN	IN包含了所要移动的值。对于MOVE_BIT，可用于所有的离散量，它不需要按字节排列。然而，从给定地址开始的1位将在线显示。
ok	此功能块使能时，OK为ON。
Q	执行移动功能后，IN中的数据被写入Q。对于MOVE_BIT，可用于所有离散量，它不需要按字节排列。然而，从给定地址开始的1位将在线显示。
LEN	LEN 必须为 1到 32,767之间的数，而对于MOVE_BIT则只能是1到256，除非IN为常数。对于MOVE_BIT，当IN为常数LEN 必须取值在 1 到 16之间。在此范围内的所有字节的跳变变量也随之改变。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	o	o	o	o	Δ	o	o	•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o	†	o	o	•	•	•	•	•	•		

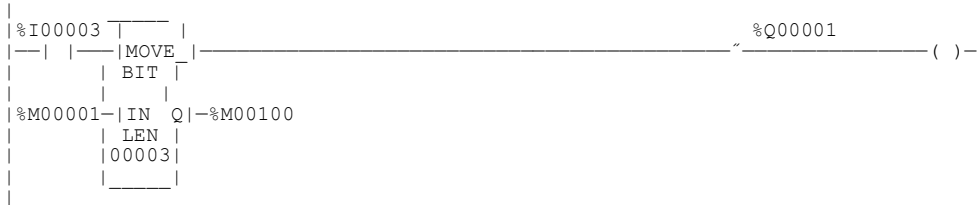
Note: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.
o Valid reference for INT, UINT, BIT, or WORD data only; not valid for DINT, DWORD, or REAL. For MOVE_BIT, discrete user references %I, %Q, %M, and %T need not be byte aligned. %U is allowed for MOVE_BIT only.
Δ Valid reference for BIT or WORD data only; not valid for INT, UINT, DINT, DWORD, or REAL.
† %SA, %SB, %SC only; %S cannot be used.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，当 %I00003 置位， %M00001，%M00002和%M00003三个位分别被复制到 %M00100， %M00101和%M00102。 线圈 %Q00001为ON。



BLKMOV (INT, UINT, DINT, WORD, DWORD, REAL)

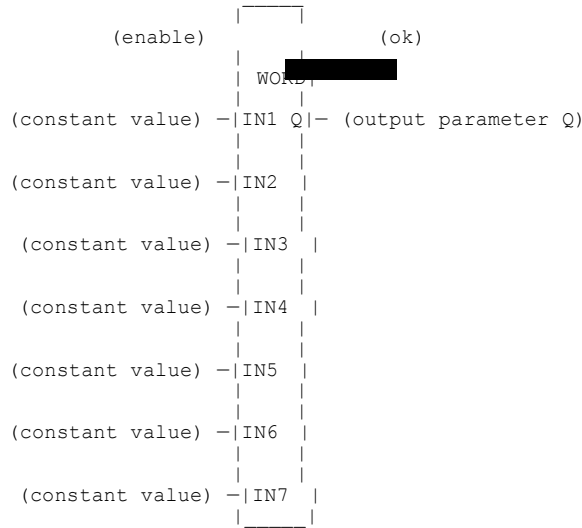
块移动功能块(BLKMOV) 将给定的7个常数拷贝到指定地址。

BLKMOV功能块有8个输入参数和2个输出参数。当功能块使能，它将常数值拷贝到从输出Q给定地址开始的连续地址中。输出Q可作为其他程序功能的输入。

注意

对于 BLKMOV_INT,, IN1 x IN 7以带符号十进制数显示；对于 BLKMOV_WORD, IN1 x IN7以十六进制数显示。

该功能块一旦使能，即向右传送电流。



参数

参数	说明
enable	一旦选通，便开始进行移动操作。
IN1 - IN7	IN1到 IN7 包含了7个常数值。
ok	一旦功能块使能，OK为ON。
Q	输出 Q 包含了被移动到的数组的第一个。 IN1 被移至 Q。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN1-IN7															•	
ok	•															•
Q		o	o	o	o	Δ†	o		•	•	•	•	•	•		

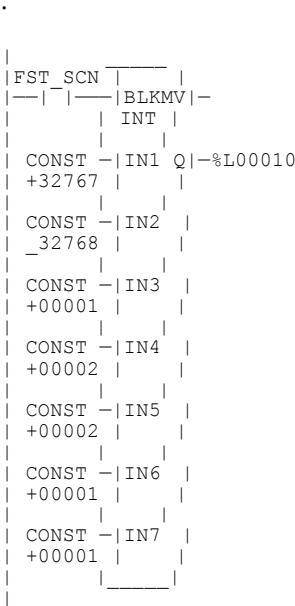
Note: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.
o Valid reference for INT, UINT, or WORD only; not valid for DINT, DWORD, or REAL.
Δ Valid reference for WORD data only; not valid for INT, UINT, DINT, DWORD, or REAL.
† %SA, %SB, %SC only; %S cannot be used.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

本例中，当功能块使能，当输入FST_SCN为ON时，BLKMOV 复制7个输入常数到%L00010至%L00016的内存地址中。



BLKCLR (WORD)

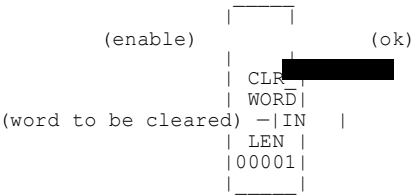
块清除功能块 (BLKCLR) 用于将指定的数据块清为零。

BLKCLR功能块有两个输入参数和一个输出参数。当功能块使能，它就向IN指定的内存地址写入0。当对离散量内存(%I, %Q, %M, %G, %T)清除数据时，相应的跳变信息也被清除。

注意

输入参数IN 不进行线圈检查。

功能块一旦选通，便开始向右传递电流。



参数

参数	说明
enable	当功能块使能，数列被清零。
IN	IN被清零数列的第一个字。
ok	功能块一旦选通，OK为ON。
LEN	LEN 必须为 1 到 256 字。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN		•	•	•	•	•†	•		•	•	•	•	•	•		
ok	•															•

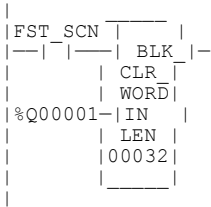
Note: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.
† %SA, %SB, %SC only; %S cannot be used.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

本例中，上电时，从%Q00001开始的32个字的%Q内存 (512点)全部被清零。与之相关的跳变信息也随之被清除。



SHFR (BIT, WORD, DWORD)

移位功能块(SHFR)用于将一个或多个数据字或位从某个变量地址移至某个指定内存区域。例如，一个字可以被移入至一个长度为5个字的内存区中。作为此移动的一个结果，另外的数据字将从该数据区的末端移出。

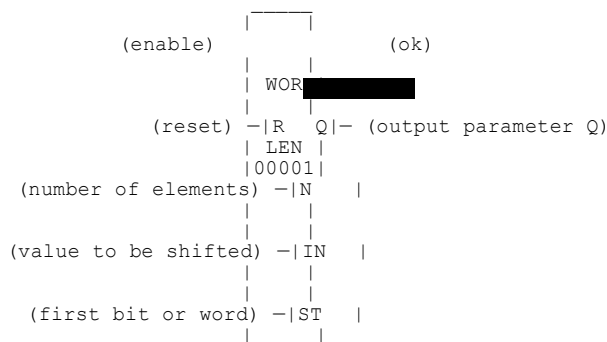
注意

当指定变量地址在进行多字的操作时，如果发生数据地址重叠的情况，将导致不可预料的后果。

SHFR功能块有5个输入参数和2个输出参数。复位输入端(R)的优先级要高于使能端。当复位端为ON时,从移位寄存器开始,由LEN给定长度的数据将被清零。LEN决定了寄存器移位的长度。

当该功能块使能且复位端没接通时，它将移位寄存器中的参数向下移动参数N个元素（位或字）的数量。移位寄存器最后一个元素将被移入Q。IN最右端一个元素将被移入从ST开始的腾空的元素位置。移位寄存器的内容可通过程序读取，因为它们是在可寻址内存区域中的绝对地址。

一旦功能块通过使能逻辑使能，便向右传递电流。



参数:

参数	说明
enable	当该功能块使能且复位端没接通时，执行该功能块。
R	当R端接通，在ST端的移位寄存器均被清零。
N	N 将被移入移位寄存器的元素的数量。
IN	IN 将被移入移位寄存器的元素的第一个字或位。 对于 SHFR_BIT适用于任何离散变量，它不需进行字节调整。然而，从指定的变量地址开始的1位，将被在线显示。
ST	ST 为移位寄存器的元素的第一个字或位。对于SHFR_BIT适用于任何离散变量，它不需进行字节调整。然而，从指定的变量地址开始的16位，将被在线显示。
ok	当使能时，OK为ON。
Q	输出 Q为从移位寄存器移出的位或字。 对于 SHFR_BIT适用于任何离散变量，它不需进行字节调整。然而，从指定的变量地址开始的16位，将被在线显示。
LEN	LEN 必须在1 到 256 之间，可以是位，字或双字。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
R	•															
N																•
IN	*	o	o	o	o	o	o		•	•	•	•	•	•	•	
ST		o	o	o	o	o†	o		•	•	•	•	•	•	•	
ok	•															•
Q	*	o	o	o	o	o†	o	o	•	•	•	•	•	•	•	

Note: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.
* Valid reference for WORD or DWORD data only; not valid for
BIT. o Valid reference for BIT or WORD data only; not valid for
DWORD.
† For SHFR_BIT, discrete user references %I, %Q, %M, and %T need not be byte aligned.
%SA, %SB, %SC only; %S cannot be used.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例 1:

本例中，移位寄存器运行于从%R00001到 %R00100的寄存器内存上。当复位端变量CLEAR为ON时，移位寄存器中的字被置0。

当变量 NXT_CYC为ON且CLEAR为OFF时，处于%Q00033的输出状态表中的字将被移入移位寄存器中。从移位寄存器中移出的字被存储在输出%M00005中。此例中需注意的，由LEN指定的长度与被一旦的数据量N并不相同。

```

| NXT_CYC |-----|
|-----| SHFR |-----|
|-----| WORD |-----|
| CLEAR   |-----| R   Q |-%M00005
|-----| LEN  |-----|
|-----| 00100 |-----|
| CONST -|N   |-----|
| 00001  |-----|
| %Q00033 | IN   |-----|
|-----| ST   |-----|
| %R00001 |-----|

```

例 2:

本例中，移位寄存器运行于%M00001到 %M00100的寄存器内存。当复位端变量CLEAR为ON时，移位寄存器中的字被置0。

当变量 NXT_CYC为ON且CLEAR为OFF时，处于%M00001到 %M00100中的数据将被向下移动一位。%Q00033中的值被移入%M00001，同时%M00100中的值被移出并写入到%M00200。

```

| NXT_CYC |-----|
|-----| SHFR |-----|
|-----| BIT  |-----|
| CLEAR   |-----| R   Q |-%M00200
|-----| LEN  |-----|
|-----| 00100 |-----|
| CONST -|N   |-----|
| 00001  |-----|
| %Q00033 | IN   |-----|
|-----| ST   |-----|
| %R00001 |-----|

```

BITSEQ (BIT)

位序列发生器 (BITSEQ)通过一个位数组执行位序列移动功能，BITSEQ功能块有5个输入参数和一个输出参数。功能块的运行要取决于上一次参数EN的值，如下表所示：

R 当前 执行状态	EN 上次 执行状态	EN 当前 执行状态	位序列执行状态
OFF	OFF	OFF	不工作。
OFF	OFF	ON	增加或减少1。
OFF	ON	OFF	不工作。
OFF	ON	ON	不工作。
ON	ON/OFF	ON/OFF	复位。

复位端 (R) 优先于使能端(EN)对序列发生器进行复位。当R端ON时，当前步数被设置为通过步数参数输入的数值。如果没有步数值被送入，步数被设为1，除了由当前步指明的位为1外，位序列发生器中的其他位均被置0。

当EN为ON且R为OFF时，由当前步所指明的位被清零。而当前步数值是增还是减取决于方向参数。由新步数指定的位被置1。

当步数值不断增加并超出范围（ $1 \leq \text{步数} \leq \text{LEN}$ ）后，它被置为1。

当步数值不断减小并超出范围（ $1 \leq \text{步数} \leq \text{LEN}$ ）后，它被置为LEN。r

参数ST是可选的，如果不用它，只是没有位被置位或清零，对BITSEQ的执行无任何影响。BITSEQ的步数在允许的范围内进行循环。

位序列的内存要求

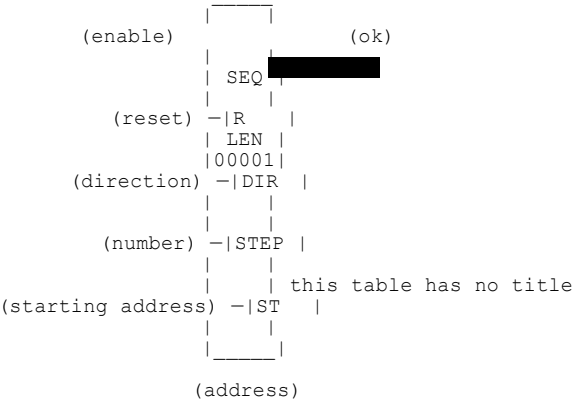
每个位序列需使用三个%R, %L或%P内存字 (寄存器) 来存储以下数据:

当前步数	word 1
位序列长度 (位)	word 2
控制字	word 3

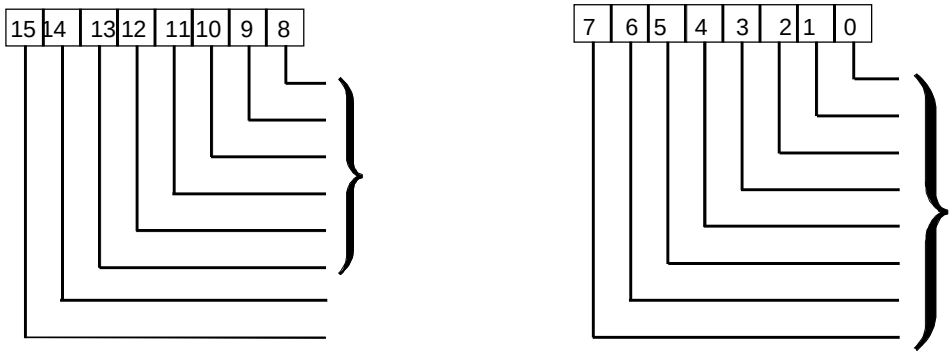
注意

其他功能块不能再向这些寄存器写入数据。

当输入一个位序列后，必须在代表此功能块的图块下方输入这三个连续字的地址，如：



控制字用于存储与此功能块有关的布尔输入输出状态，如下面格式所示：



注意

0到13位未使用。

参数

参数	说明
address	位序列需使用三个%R, %L或%P内存字 (寄存器) 来存储以下数据: • 当前步数 = word 1. • 位序列长度 = word 2. • 控制字 = word 3. 当输入一个位序列后, 必须在代表此功能块的图块下方输入这三个连续字的地址。详细信息见上页 注意: 不要再在其他指令中使用这些地址。 小心: 变量重叠将导致不正确的运行结果。
enable	当使能为ON, 如果上一次扫描时使能为OFF且R为OFF则执行位序列功能。
R	当R为ON, 位序列中的步数被置为STEP中的值 (默认为1), 除了当前步所指为1外, 其余位均清零。
DIR	当 DIR 为ON时, 位序列数在移位前增加, 否则就先减小。
STEP	当R 为ON, 步数被设置为该值。
ST	ST 包含了位序列的第一个字。
ok	一旦选通该功能块, OK为ON。
LEN	LEN 必须为1到256字。

注意

在BITSEQ功能块进行线圈检查时, 将从ST开始检查16位, 即使LEN的值小于16。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
address									•					•		
enable	•															
R	•															
DIR	•															
STEP		•	•	•	•		•		•	•	•	•	•	•	•	•
ST		•	•	•	•	†	•		•	•	•		•	•		•
ok	•															

• Valid reference or place where power may flow through the function.
† %SA, %SB, %SC only; %S cannot be used.

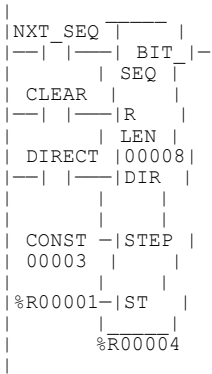
注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制” 部分内容。

例:

此例中，对于寄存器%R00001进行位序列操作。其静态数据存在R00004, R00005和 R00006中，当CLEAR为ON时，位序列被清零同时当前步被置为3。%R00001的前八位被置0。

当 NXT_SEQ 为ON且 CLEAR为OFF时，由步数3对应的位被清零，而步数2或步数4的位（由方向参数DIR决定的）被置为1。



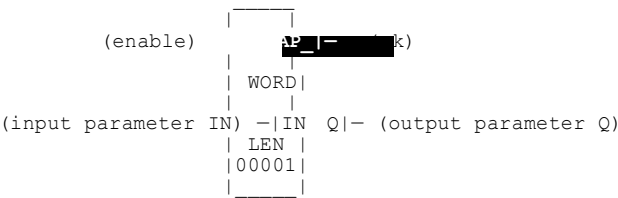
SWAP (WORD, DWORD)

SWAP 功能块用来交换字中的两个字节或双字中的两个字的位置。

SWAP功能块可对于长度大于1的内存区域进行变换，以对该段指定长度的数据进行相应的数据变换。

SWAP功能块有两个输入参数和两个输出参数。当SWAP功能块选通，它将对给定区域的每个字或双字执行变换操作，变换结果被放入输出Q。

SWAP 功能块使能，就向右传递电流。



参数

参数	说明
enable	一旦选通，进行相应的交换。
IN	IN 包含了需要变换的数据。
ok	功能块使能，OK为ON。
Q	输出Q 为变换后的数据。
LEN	LEN 必须在 1 到 256字。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	o	o	o	o		o		•	•	•	•	•	•	•	
ok	•															•
Q	•	o	o	o	o		o		•	•	•	•	•	•		

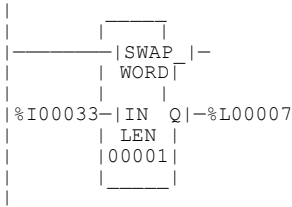
Note: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.
o Valid reference for WORD data only; not valid for DWORD.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，%I00033到%I00048的两个字节进行交换，结果放在%L00007中。



COMMREQ

当程序需要同智能模板如总线控制器，可编程序协处理器，或局域网接口模板通讯时，使用通讯请求功能块(COMMREQ)来完成此操作。在下一页的叙述中，说明了COMMREQ功能块的格式。若需要编制针对某个类型设备的COMMREQ程序，需要更多的其他信息。使用COMMREQ功能块来为模板编程的这些要求需要查看相应模板的手册。

注意

如果使用串行通讯，见《系列90™PLC串行通讯用户手册(GFK-0582)》。如果使用MMS-Ethernet通讯，见《系列90™PLCMMS-Ethernet通讯用户手册(GFK-0686)》。

下面是可被请求的通讯类型的一个例子，GENIUS总线控制器使用以下类型的信息。

命令	说明
1	脉冲测试输出。
2	从总线设备读配置数据。
3	向总线设备读配置数据。
4	读诊断。
5	清除回路错误。
6	清除所有电路错误。
7	分配监视器。
8	启用/禁用输出。
9	启用/禁用全局数据。
10	切换总线开关模块。
11	读设备。
12	写设备。
13	出列故障报文。
14	发送故障报文。
15	获取故障报文。

发送故障报文和获取故障报文命令 (14 和 15条) 可用于传送某些附加信息，对于总线控制器的编程信息，见《总线控制器手册，GFK-0398》。

注意这些设备也在系统通讯窗口与PLC进行全局数据交换，这将在每次扫描末尾自动执行。这种通讯不需借助程序来完成。

COMMREQ功能块有4个输入参数和两个输出参数。当该功能块被选通，一个命令数据块发送到通讯任务。命令块起始地址由输入参数IN指定。与之通讯的设备的SYSID可通过输入其机架号和槽号来识别。

COMMREQ 可在发送信息后等待应答，也可在发送信息后不等待应答。如果命令块指定程序不等待应答，则命令块内容发送给接收设备后，程序立即重新执行。（无超时时间值），这种方式被称为**NOWAIT**模式。

如果命令块指定程序等待应答，则命令块内容发送到接收设备后CPU开始等待应答。PLC等待应答的最大时间由命令块给定。如果设备在此时间段内无应答，则程序重新执行。这种方式称为**WAIT**模式。

功能块通过电流，除非超时或超时周期时间被设定为0。输出参数FT(功能块故障)在下述情况下置为ON：

1. 指定的目标地址不存在。
2. 指定的任务对设备无效。
3. 数据长度为0。

功能块故障输出可能有以下几种状态：

使能	错误	功能块错误输出
激活	无	OFF
激活	有	ON
未激活	未执行	OFF

命令块

命令块为COMMREQ提供了所需的附加信息。

命令块的地址由输入参数IN来指定。该地址可使用任何字形式的用户变量(%R, %L, %P, %AI, %AQ)。命令块的长度取决于被发送的特定的命令。

命令块的结构如下：

数据块长度	address
等待/不等待标志	address + 1
状态指针内存类型	address + 2
状态指针偏移	address + 3
空闲超时值	address + 4
最大通讯时间	address + 5
数据块	address + 6
	到
	address + 133

可用 MOV, BLKMOV或 DATA_INIT功能块将命令块所需数据放入指定的内存区域中。

输入功能块信息时，参照以下定义：

数据块长度 命令块中从地址+6开始到末尾数据字的长度。该值可在1到128字之间。每个COMMREQ 命令都有自己的数据长度。在输入数据块长度时，必须确认命令块在寄存器限制的范围内。

等待/不等待标志: 这个标志用来选择程序是否等待CCM通讯的完成。

目的	输入值
不等待	0
等待应答	1

标志位存放在地址+ 1处的最低有效位(LSB)中。该字的其余位由0填充。

注意

等待模式下 COMMREQ不能直接连接串口1 和 2， 对于7.0版或更新的CPX CPU。

状态指针内存类型: 两个状态指针字指明了PLC内存中放置COMMREQ执行完成后设备返回的状态字所写入的位置。

状态指针内存类型	address + 2
状态指针偏移	address + 3

状态指针内存类型包含一个数字码，为状态字规定了用户变量内存类型，下表列出了每种变量类型码。

内存类型		输入数值 *
%I	离散量输入表 (BIT模式)	70
%Q	离散量输出表 (BIT 模式)	72
%I	离散量输入表 (BYTE 模式)	16
%Q	离散量输出表 (BYTE 模式)	18
%R	寄存器内存	8
%AI	模拟量输入表	10
%AQ	模拟量输出表	12

* 均为十进制数。

注意

输入的值决定了所采用的模式。例如，如果输入%I 的位模式为70，则偏移量将按位来计。另一方面如果%I 值为16则偏移量将按字节来计。
地址+ 2处的高字节应为0。

状态指针偏移: 地址+ 3处为在所选内存中的状态字的偏移量。

注意

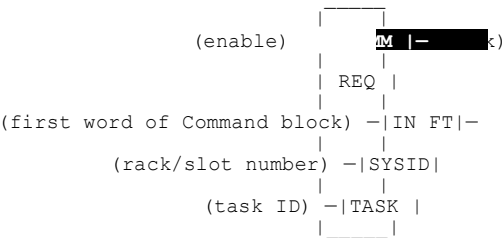
这些状态指针偏移是一个基于0的值。例如，%R00001为偏移为0的寄存器表。

空闲超时值：空闲超时值为PLC CPU等待接收COMMREQ 的设备应答的最大时间。该值在**NOWAIT**模式下不起作用。如果选择**WAIT**模式，则地址+ 4为以100ms为时间单位的超时时间值。

最大通讯时间：此值在address +5处，它给定了PLC CPU等待设备完成COMMREQ的最大时间。此时间以100ms为时间单位并在**NOWAIT**模式下不起作用。

数据块

数据块包含了命令的参数。数据块起始于地址+ 6的命令数，它用于区分不同的通讯类型。有关特定的COMMREQ命令格式见特殊设备的（如，PCM，GBC，通讯）手册。



参数

参数	说明
enable	使能时，执行通讯请求命令。
IN	IN命令块的第一个字。
SYSID	SYSID 包含了通讯设备的机架号 (最大有效字节)和槽号 (最低有效字节) 。
TASK	TASK 目标设备的任务处理 ID。
FT	FT 当通讯失败时接通，在9 -2页作过详细说明。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN									•	•	•	•	•	•		
SYSID		•	•	•	•		•		•	•	•	•	•	•	•	
TASK									•	•	•	•	•	•	•	
ok			•													
•																
FT			•													
•																

注意: 间接变量对所有寄存器变量均有效。(%R, %P, %L, %AI, %AQ, %UR).
• 可使电流流经功能块的变量或地址。.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

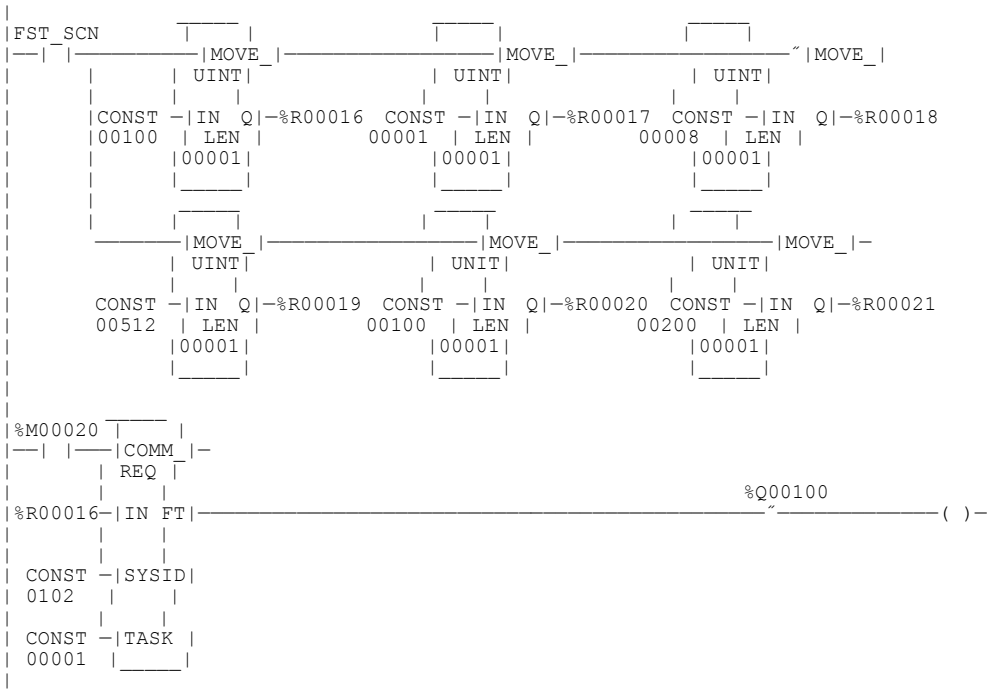
例 1:

此例中，当使能输入 %M00020为ON，开始于%R00016的命令块被发送给位于PLC的1号机架，2槽的通讯任务为1的设备。如果出现错误，%Q00100变为ON。



例 2:

本例说明如何使用MOVE功能块来输入例1中所述的COMMREQ的命令块。



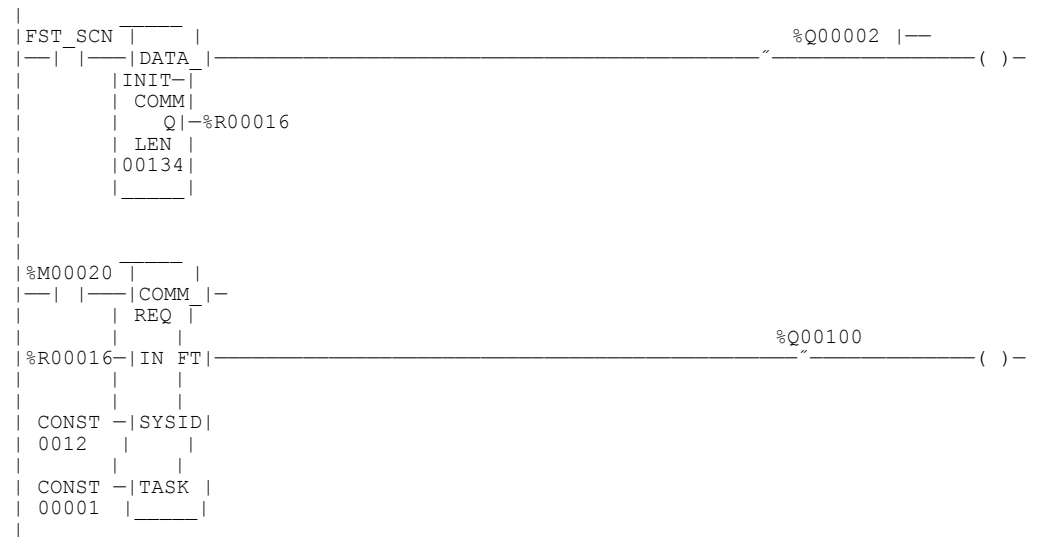
COMMREQ的输入 IN指定%R00016为命令块的起始地址。连续的变量如下所示:

数据块长度	%R00016
等待/不等待标志	%R00017
状态指针内存类型	%R00018
状态指针偏移	%R00019
空闲超时时间	%R00020
最大通讯时间	%R00021
数据块	%R00022至数据块结束

MOVE 功能块实现了为COMMREQ提供以下命令块数据的功能。第一个MOVE功能向%R00016中写入待通讯数据的长度。第二个MOVE向%R00017中写入1，即指定为**WAIT**模式。

第三个MOVE 功能块将8放入%R00018。即规定了寄存器作为状态指针的位置。下一个MOVE将512放入%R00019，因此状态指针被定位于%R00513。另两个MOVE将常量100和200分别放入%R00020和%R00021，100即为10ms的空闲超时时间，200则代表了20ms的最大通讯时间。

例2中的六个MOVE功能块可以只使用一个DATA_INIT_COMM功能块来完成，如下图所示：



在Logicmaster中，将光标移动到 DATA_INIT_COMM 功能块上并按 **Zoom (F10)** 来进入DATA_INIT_COMM窗口

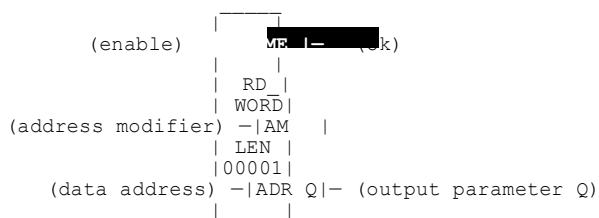
VMERD (BYTE, WORD)

VME读功能块 (VMERD)用于从VME总线读取数据

注意

使用VME 功能块 (VMERD, VMEWRT, VMERMW, VMETST)时，需要得到给出VME板的地址的正确的附加信息。这些信息需要从一个或两个源程序中获得。对于一个规范的VME模板，卖方会提供正确使用它的说明。另外，还可参考《选择第三方VME模板指南GFK-0448》。

VMERD功能块有3个输入参数和一个输出参数。当该功能块使能，它在ADR和地址偏移AM指定的地址处访问VME模板，并将长度为LEN的数据拷贝到输出Q开始的PLC地址中。如果这些操作成功完成，则向右传递电流。



参数:

参数	说明
enable	使能时，进行VME读操作。
AM	AM地址偏移
ADR	待读数据的地址。
ok	功能块使能并成功读取数据，OK为ON。
Q	由ADR和AM指定的地址处读来的数据。
LEN	LEN 可以为 1到 32,767。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
AM	.								.	.	.	.	.	.	.	
ADR	.								.	.	.	.	.	.	.	
ok	.															.
Q	o	o	o	o	o		o		.	.	.	.	.	.		

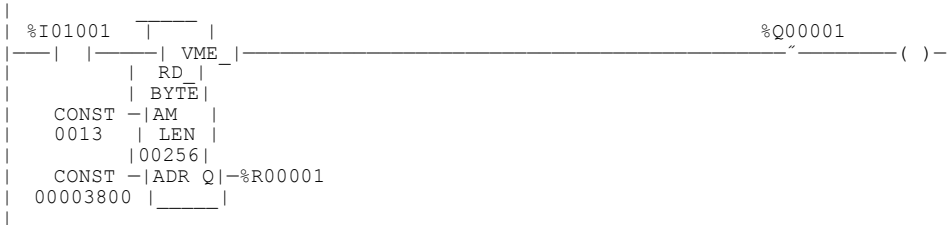
注意: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.
o Valid reference for BYTE data only; not valid for WORD.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，当使能输入%I01001变为ON，它将从VME总线地址为3800处短地址空间内读取256字节放入%R00001 到 %R00128寄存器中。（对于具有BTM的多机架系统，这应处在第4机架上。）除非在读数据时发生错误，线圈%Q00001将一直为ON。



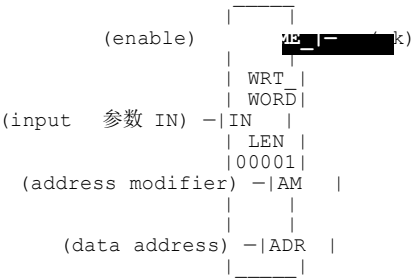
VMEWRT (BYTE, WORD)

VME写功能块 (VMEWRT)用于向VME总线写数据。

注意

使用VME 功能块 (VMERD, VMEWRT, VMERMW, VMETST)时，需要得到给出VME板的地址的正确的方式的附加信息。这些信息需要从一个或两个源程序中获得。对于一个规范的VME模板，卖方会提供正确使用它的说明。另外，还可参考《选择第三方VME模板指南GFK-0448》。

VMEWRT功能块有4个输入参数一个输出参数，当它使能时，功能块从输入参数IN拷贝数据写入到由在ADR和地址偏移AM指定的地址处。在成功的转移数据到指定地址后，功能块向右传递电流。



参数:

参数	说明
enable	使能时，进行VME写操作。
IN	IN 包含了要写入地址的数据。
AM	AM 地址偏移。
ADR	待写数据的地址。
LEN	LEN 可为 1 到 32,767。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.	o	o	o	o		o		.	.	.	.	.	.	.	
AM	.	.	.	.	.		.		.	.	.	.	.	.	.	
ADR	.								.	.	.	.	.	.	.	
ok	.															.

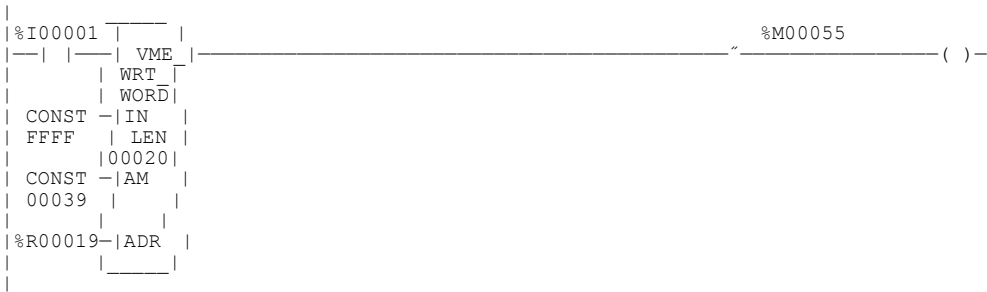
注意: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function. o Valid reference for BYTE data only; not valid for WORD.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中， 当使能输入 %I00001为ON， 十六进制数FFFF 被写入VME总线上的20个数中的每一个字，第一个字（最低地址）的位置由%R00019(低字)和%R00020 (高字)中的内容决定。内部变量%M00055将被置位， 除非写数据过程中发生错误。



VMERMW (BYTE, WORD)

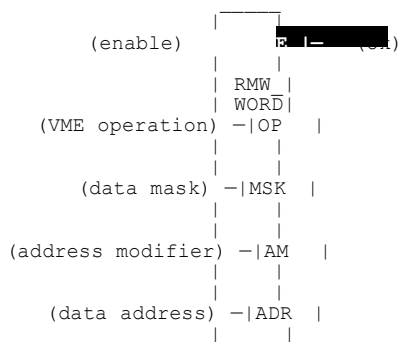
VME读/改/写功能块 (VMERMW) 用于更新VME总线上的数据元素。

注意

使用VME 功能块 (VMERD, VMEWRT, VMERMW, VMETST)时，需要得到给出VME板的地址的正确的附加信息。这些信息需要从一个或两个源程序中获得。对于一个规范的VME模板，卖方会提供正确使用它的说明。另外，还可参考《选择第三方VME模板指南GFK-0448》。

VMERMW功能块有5个输入参数和1个输出参数。当该功能块选通，它在ADR和地址偏移AM指定的地址处读取VME总线数据，这些数据将同数据屏蔽字MSK作由输入OP来决定的相应的逻辑运算(AND/OR)。MSK是一个字。如果要运算的是字节，则仅使用MSK的低8位。

结果将写回到读取它们的VME地址中。如果这些操作成功完成，则向右传递电流。



参数:

参数	说明
enable	使能时，VME功能块运行。
OP	OP指定数据是否要与MSK 进行AND或 OR运算。
MSK	MSK 为屏蔽数据。
AM	AM 地址偏移。
ok	功能块使能并成功操作数据，OK为ON。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
OP															.	
MSK	.	.	.	.	.		.		.	.	.	.	.	.	.	
AM	.	.	.	.	.		.		.	.	.	.	.	.	.	
ADR	.								.	.	.	.	.	.	.	
ok	.															.

注意: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the function.

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，使能输入 %M00044为ON，十六进制数80H 将与从VME总线上的0机架（主机架）中的地址为060010H处使用标准非特权数据存取方式读出的数据进行“或”运算。如果没有发生错误，将使线圈%Q00027置为ON。



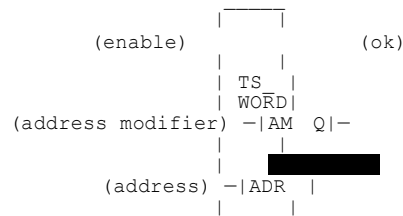
VMETST (BYTE, WORD)

使用 VME 测试和置位功能块 (VMETST) 来处理在VME总线上的信号量。VMETST用于交换当前信号量位置中的布尔数值ON(1)。如果该值已经为ON，则VMETST功能不获取信号量。如果为OFF，则将信号量复位并且VMETST获得该信号量并使用它控制的内存区域。清除该信号是通过VMEWRT向该信号量地址中写入0来实现的。

注意

使用VME 功能块 (VMERD, VMEWRT, VMERMW, VMETST)时，需要得到给出VME板的地址的正确的方式的附加信息。这些信息需要从一个或两个源程序中获得。对于一个规范的VME模板，卖方会提供正确使用它的说明。另外，还可参考《选择第三方VME模板指南GFK-0448》。

VMETST功能块有3个输入参数和2个输出参数。当该功能块被选通，布尔值ON在由ADR指定地址和AM指定地址偏移量的地址的数据进行交换。如果该信号量存在(OFF)并被获取，则VMETST功能块置输出Q为ON。当功能块使能且在执行中无错误产生时，向右传递电流。



参数:

参数	说明
enable	使能时，VME 测试和置位运行。
AM	AM为地址偏移量。
ADR	ADR 包含了信号量的地址。
ok	功能块使能并成功操作数据，OK为ON。
Q	如果信号位有效(OFF)输出Q被置为ON，否则Q为OFF。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
AM	.								.	.	.	.	.	.	.	
ADR	.								.	.	.	.	.	.	.	
ok	.															.
Q	.															.

注意: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).

- Valid reference or place where power may flow through the function.

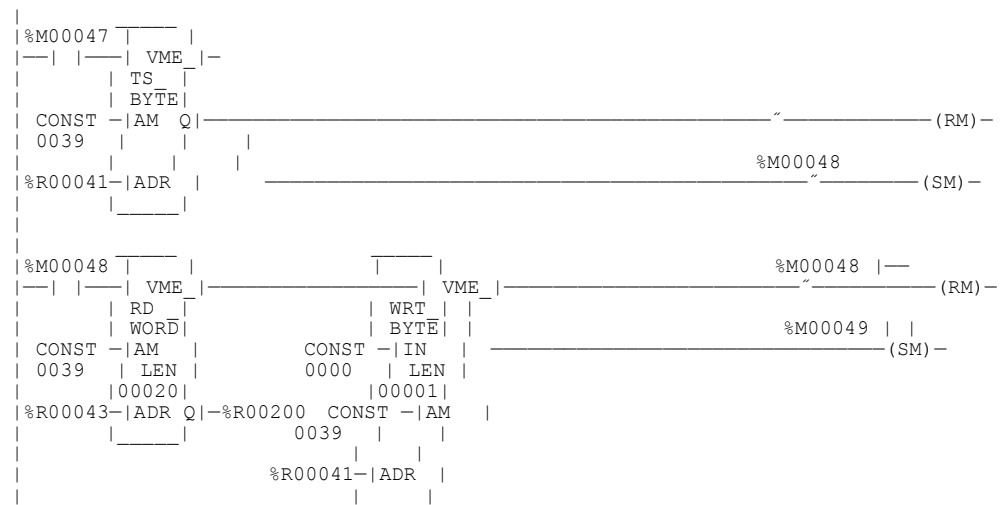
注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制” 部分内容。

例:

此例中，由VMERD, VMEWRT和VMETST功能块来读取被一个信号量所保护的数据。当使输入%M00047为ON时执行VMETST来获取信号量。信号量VME地址存储在使用标准无优先级数据访问的%R00041和%R00042中。当此操作成功，线圈%M00047复位，%M00048置位。%M00048置位后，VMERD功能块开始读取数据（VME地址存储在%R00043和 %R00044中的20个字的数据被读入到%R00200到%R00219中）。当读取完成（没有被中断或出现程序错误），VMEWRT功能块放弃该信号量。当VMEWRT成功执行后，线圈%M00048复位。%M00049指向目前有效的新数据。

如果信号量无效，VMERD和VMEWRT将不被执行。结果是设置%M00047使PLC在每次扫描时均检测该信号量直到该信号量有效为止。当信号量变有效时，信号将被获取，数据被读入，然后信号被放弃。在%M00047再次置位前，不会再有进一步的动作。



VME_CFG_RD

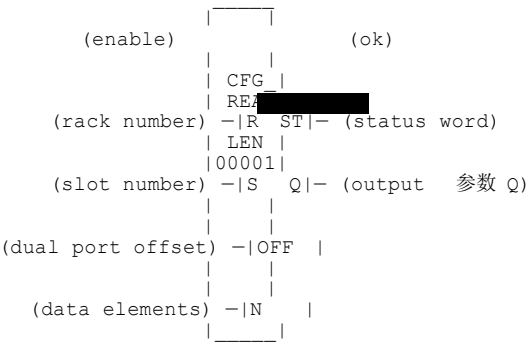
VME_CFG_RD功能块用于从VME总线读取数据。该功能块有5个输入参数和3个输出参数。当功能块被选通，它从由机架(R)，槽(S)，双端口偏移(OFF)指定的VME总线地址读取数据元素(N)。读到的数据放入输出Q。操作状态放在状态字输出(ST)中。该功能块的输出数据列的长度由参数LEN声明。

注意

这里由机架号和槽号指定的模板必须被在总线接口模式配置为第三方VME模板，此功能块才能正常执行。另外，用于表明了AM码，模板双端口的位位置和大，以及总线接口类型的参数都要在模板配置中声明才能正确的运行。关于配置第三方VME模板的详细信息，见《Logicmaster 90-70编程软件用户手册 (GFK-0263)》。

如果此功能块成功完成，则OK为ON；否则，为OFF。当在以下一些情况下，OK也会为OFF状态：

- 给定的数据元素数(N)大于给定长度(LEN)。
- 机架/插槽值 (R 和 S) 不在有效的VME位置。
- 双端口偏移 (OFF)的最高有效字节不为0。
- 双端口地址加双端口偏移的最高有效字节不为0。
- 读操作超出了双端口的末端地址。
- 在总线接口模式中没有对第三方VME模板进行机架/槽的配置。
- 如果双端口偏移为偶数，只配置奇数字节。如果双端口偏移为奇数，则只配置字或单字。



参数:

参数	说明
enable	使能时, 进行数据初始化。
R	所指定的机架号。
S	指定的槽号。
OFF	双端口偏移量。
N	从VME总线读取的数据总量。
ok	执行过程无错误, 则OK为ON。
ST	操作的状态。
Q	功能块运行, 数据被读入Q。
LEN	LEN 输出数据的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
R	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
S	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
OFF	•					•			•	•	•	•	•	•	•	
N	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
ok	•															•
ST	•	•	•	•	•	†	•		•	•	•	•	•	•		
Q	•	•	•	•	•	†	•		•	•	•	•	•	•		

• 可使电流流经功能块的变量或地址。

† %SA, %SB, %SC可以, %S不可用。

注意

对于在参数子程序块中使用的限制, 见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，当使能ON，位于机架1，槽3，双端口偏移由%R00100指定的VME数据被读入%R00101到%R00116中的%R00101到%R00110。如果发生错误，则状态字%AQ0001会有相应的错误代码。

```

| %I00001 | _____ |
|-----| |-----| VME |-----|
|          | CFG |
|          | READ|
| CONST -|R ST|-%AQ0001
| 00001 | LEN |
|          | 00016|
| CONST -|S Q|-%R00101
| 00003 | _____ |
| %R00100-|OFF |
|-----| |-----|
|          |
| CONST -|N |
| 00010 | _____ |
|-----|

```



VME_CFG_WRITE

VME_CONFIG_WRITE功能块用于向VME总线写数据。该功能块有6个输入参数和2个输出参数。当功能块选通，从输入数据列(IN)来的数据元素(N)将写入由机架(R)，槽(S)，双端口偏移(OFF)指定的VME总线地址。操作的状态放在状态字输出(ST)中，该功能块的输出数据列的长度由参数LEN声明。

注意

这里由机架号和槽号指定的模板必须被在总线接口模式配置为第三方VME模板，此功能块才能正常执行。另外，用于表明了AM码，模板双端口的位和大小，以及总线接口类型的参数都要在模板配置中声明才能正确的运行。关于配置第三方VME模板的详细信息，见《Logicmaster 90-70编程软件用户手册(GFK-0263)》。

如果此功能块成功完成，则OK为ON；否则，为OFF。当在以下一些情况下，OK也会为OFF状态：

给定的数据元素数(N)大于给定长度(LEN)。

机架/插槽值 (R 和 S) 不在有效的VME位置。

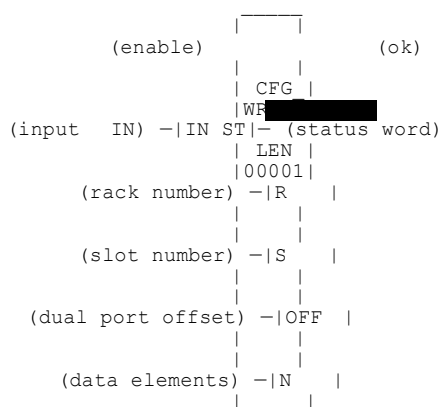
双端口偏移 (OFF)的最高有效字节不为0。

双端口地址加双端口偏移的最高有效字节不为0。

读操作超出了双端口的末端地址。

在总线接口模式中没有对第三方VME模板进行机架/槽的配置。

如果双端口偏移为偶数，只配置奇数字节。如果双端口偏移为奇数，则只配置字或单字。



参数:

参数	说明
enable	使能时，进行数据初始化。
R	所指定的机架号。
S	指定的槽号。
OFF	双端口偏移量。
N	从VME总线写入的数据总量。
ok	执行过程无错误，则OK为ON。
ST	操作的状态。
Q	功能块运行，数据被写入Q。
LEN	LEN 输出数据的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
R	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
S	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
OFF	•					•			•	•	•	•	•	•	•	
N	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
ok	•															•
ST	•	•	•	•	•	†	•		•	•	•	•	•	•	•	

• 可使电流流经功能块的变量或地址。

† %SA, %SB, %SC可以， %S不可用。

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，当使能ON，%R00101到%R00116的数据列的%R00101到%R00110的数据被写入机架1，槽3和由%R00100指定的双端口偏移的VME总线上。如果出现错误，则状态字%AQ0001中将会有相应的错误代码。

```

| %I00001 |-----|
|-----|-----| VME |-----|
|          | CFG |
|          | WRITE|
| %R00101 | IN ST | %AQ00001
|          | LEN |
|          | 00016|
| CONST -| R |
| 00001 | |
|          |
| CONST -| S |
| 00003 | |
|          |
| %R00100 | OFF |
|          |
| CONST -| N |
| 00010 |-----|

```

DATA_INIT

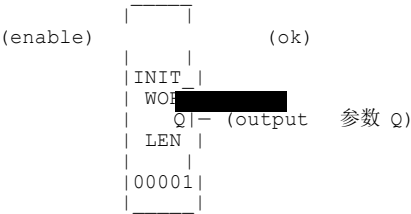
(INT, UINT, DINT, WORD, DWORD, REAL)

数据初始化功能块(DATA_INIT) 用于向某个地址区域中复制一个常量数据块。它有1个输入参数和2个输出参数。当功能块选通, 它复制常量数据到输出Q。该功能块的常数数据长度参数(LEN) 给定了将有多少常数被复制到输出Q开始的连续地址上。

注意

输出参数不进行线圈检查。

功能块使能后便向右传递电流。



注意

当 DATA_INIT指令在程序中使用时, 常数被初始化为0。使用缩放窗口功能可改变这些常量的内容（详见下面的描述）。

参数:

参数	说明
enable	使能时, 数据初始化运行。
ok	功能块使能OK便为ON。
Q	当数据初始化运算, 常数被写入Q。
LEN	指明写入Q中数据的长度。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
ok	•															•
Q		o	o	o	o		o		•	•	•	•	•	•		

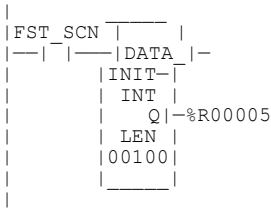
注意: Indirect referencing is available for all register references (%R, %P, %L, %AI, %AQ, and %UR).
• Valid reference or place where power may flow through the
function. o Valid reference for INT, UINT, or WORD data only..

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制” 部分内容。

例:

此例中， 首次扫描 (FST_SCN)时， 100 个字的初始数据被复制到%R00005到%R00104中。



进入 DATA_INIT_type 功能块

在 Logicmaster中，按**Zoom (F10)**键来进入DATA_INIT_type 功能块。

DATA_INIT_type 包含一个**New Value**区域作为一个小命令行，一个**Element**区域来指明当前光标所在的数据元素，一个**Length**区域来表明该DATA_INIT_type指令包含了多少个元素，以及常量数据块。 **Element**和**Length**区域不可编辑。

要改变一个值，先将光标移动到要改变的元素上，然后在**New Value**区域输入新的值就可以了。

为了帮助定位光标，窗口在数据块左侧包含了一个起始元素标记行。**Element** 区域包含了当前光标所在位置的元素的号。窗口的数据块部分可滚屏和翻页。

DATA_INIT_COMM

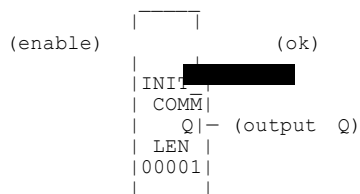
数据初始化通讯请求功能块(DATA_INIT_COMM)用于通过一个常量数据块对COMMREQ功能块进行初始化。COMMREQ的IN参数必须同DATA_INIT_COMM功能块的输出Q相对应。

DATA_INIT_COMM功能块有1个输入参数和2个输出参数。当该功能块选通，它将拷贝常数到输出Q。LEN参数指定了将被拷贝至输出Q开始的连续地址的常数的个数。该值必须等于等于COMMREQ功能块中整个命令块的大小。

注意

输出参数不进行线圈检查。

功能块使能后便向右传递电流。



注意

当 DATA_INIT指令在程序中使用，常数被初始化为0。使用缩放窗口功能可改变这些常量的内容（详见下面的描述）。

参数:

参数	说明
enable	功能块使能，数据初始化运行。
ok	功能块使能后OK为ON。
Q	数据初始化进行时，常量数据被写入Q。
LEN	LEN参数指定了将被拷贝至输出Q开始的连续地址的常数的个数。该值必须等于等于COMMREQ功能块中整个命令块的大小。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
ok	•															•
Q									•	•	•	•	•	•		

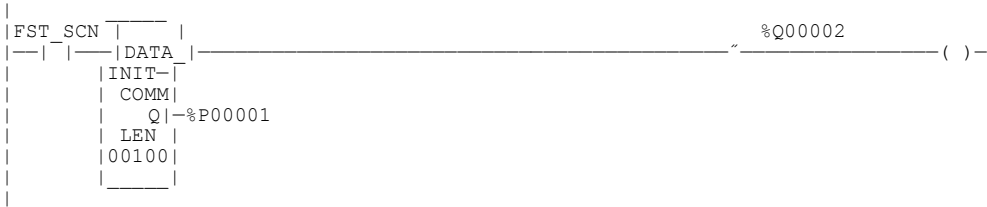
注意: 间接变量对所有寄存器变量都有效 (%R, %P, %L, %AI, %AQ, and %UR).。
• 可使电流流经功能块的变量或地址。

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。。

例:

此例中， 在第一次扫描(FST_SCN)时， 一个包括100字长的数据和6个字的头命令被复制到 %P00001到%P00099， 同时%Q00002得电。



进入 DATA_INIT_COMM 功能块

在 Logicmaster中，按**Zoom (F10)**键来进入DATA_INIT_COMM 功能块。

DATA_INIT_COMM窗口包含一个**New Value**区域作为一个小命令行，一个**Element**区域来指明当前光标所在的数据元素， **Element**区域不可编辑。

窗口还包含了COMMREQ命令块的前六个命令字，每个字都有标记表明该字的用途。这前六个字为：**Data Length, Wait flag, Timeout, Status Memory, Status Offset, 和 Max comm time**。

数据元素块和**Data Element**左侧的标记行与COMMREQ功能块中的数据块部分一致。例如，数据块元素标记为1，则表明该数据块起始于此指令的第七个字。

窗口的数据块部分可滚屏和翻页。

注意

数据块的信息均以十六进制显示。

DATA_INIT_ASCII

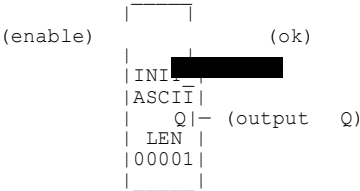
ASCII数据初始化功能块 (DATA_INIT_ASCII) 用于向某地址区域拷贝ASCII码文本。

DATA_INIT_ASCII功能块有1个输入参数和2个输出参数。当功能块选通，它将常量数据拷贝至输出Q。常量数据长度LEN声明了将被拷贝到从输出Q开始的连续地址中常量文本的字节数。LEN必须为偶数。

注意

输出参数不进行线圈检查。

当功能块选通便向右传递电流。



注意

当 DATA_INIT指令在程序中使用，常数被初始化为空。使用缩放窗口功能可改变这些常量的内容（详见下面的描述）。

参数:

参数	说明
enable	功能块使能，数据初始化运行。
ok	功能块使能后OK为ON。
Q	数据初始化进行时，常量数据被写入Q。
LEN	LEN参数指定了将被拷贝至输出Q开始的连续地址的文本常数的字节个数。该值必须为偶数。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
ok	•															•
Q		•	•	•	•		•		•	•	•	•	•	•		

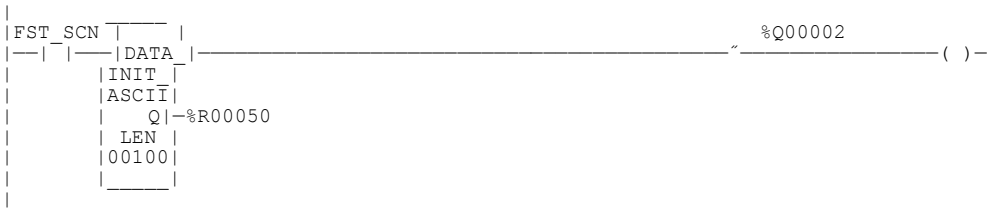
注意: 间接变量对所有寄存器变量都有效 (%R, %P, %L, %AI, %AQ, and %UR)。
• 可使电流流经功能块的变量或地址。

注意

对于在参数子程序块中使用的限制，见第二章第二节“参数子程序块中形式参数的使用的限制”部分内容。

例:

此例中，第一次扫描 (FST_SCN)时，100个字节的ASCII码文本被拷贝到%R00050到%R00149中，同时输出%Q00002 得电。



进入 DATA_INIT_ASCII 功能块

在Logicmaster中，按**Zoom (F10)**键来进入DATA_INIT_ASCII 功能块。

DATA_INIT_ASCII窗口包括一个小命令行，一个**Function Len**区域，表明该指令包含了多少个字符字节，一个**Element No.**标明当前光标所在所指向的位置和ASCII文本块。
输入ASCII文本可将光标移动到指定位置，然后在命令行上输入引号（“）其后接着键入要输入的文本即可。如“This is a test。（见下页中的例子）。

可以一次输入与命令块同样长度的字符，当要输入不可打印字符时，可以用反斜杠加上该字符的三位等效十进制数。（如“\010表示换行符）在命令行输入文本后，按回车键结束。
按退出键退出当前窗口返回到梯形图逻辑画面。

DATA_INIT_DLAN

数据初始化功能块DLAN (DATA_INIT_DLAN) 用于DLAN系统,它是一个限制可用性的特殊的系统。要了解DLAN系统的相关内容, 见《*DLAN/DLAN+* 接口模板使用手册(GFK-0729B)》。

注意

此模板对于一般订货硬件不可用。

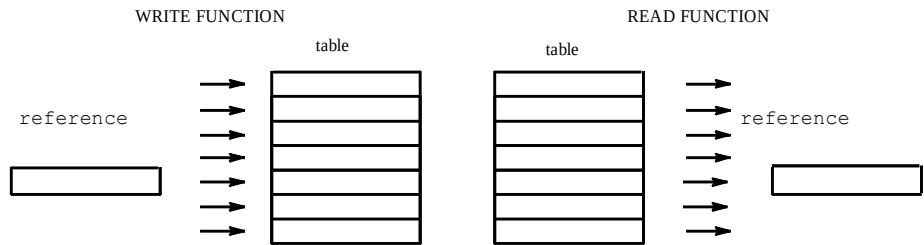
Chapter 10

第十章 数据表功能块

数据表提供自动数据移动能力。数据表功能块用于输入数据或从表中拷贝数据。为正确使用这些功能，程序必须包括其他的逻辑指令来移动表中的数据 and 初始化表的指针。

数据表数据的移入和移出

所有的表写入功能块都是从指定的变量地址读取数据并填入表中。类似地，所有表读出功能块均能将表中数据拷贝到指定区域中。



每次调用这些功能块都有一个值被写入或读出。程序中其他逻辑必须把新数值放入一个变量中，或从变量中通过读功能来获取数据。

表指针初始化

读写功能块均借助指针来跟踪当前表格位置。必须使用附加程序逻辑来初始化此数据以使该功能块从正确的位置进行读写操作。通常，变量中的值初始化为0。

本章介绍以下数据表功能块：

缩写	功能	说明	页码
TBLRD	读数据表	从表中指定地址处拷贝数据到输出变量中。	10-3
TBLWR	写入数据表	将输入变量数据写入表中指定位置。	10-5
LIFORD	LIFO读	移动指针位置入口，同时指针减1。配合LIFOWRT使用。	10-7
LIFOWRT	LIFO写	增加数据指针值并将数据写入。配合LIFORD使用。	10-9
FIFORD	FIFO读	移动表底部的条目，同时指针值减1。配合FIFOWRT使用。	10-11
FIFOWRT	FIFO写	增加表指针值，并向数据表写入数据。配合FIFORD使用。	10-13
SORT	排序	按升序对数据进行排序。	10-15
ARRAY_MOVE	数列移动	从源数列中拷贝指定数量的元素到指定的目标数列中。	10-17
SRCH_EQ	寻找相等	在所有数列中查找与给定值相等的元素。	10-21
SRCH_NE	查找不等	在所有数列中查找与给定值不相等的元素。	10-21
SRCH_GT	查找大于	在所有数列中查找大于给定值的元素。	10-21
SRCH_GE	查找大于或等于	在所有数列中查找大于或等于给定值的元素。	10-21
SRCH_LT	查找小于	在所有数列中查找小于给定值的元素。	10-21
SRCH_LE	查找小于或等于	在所有数列中查找小于或等于给定值的元素。	10-21
ARRAY_RANGE	数列范围	确定某值是否处于两个表给定的范围内。	10-24

表中所有值必须为相同的类型，可以是带符号整型(INT)，双进位带符号整型(DINT)，无符号整型(UINT)，字(WORD)，或双字(DWORD)。 BYTE数据类型可用于 SRCH_EQ, SRCH_NE, SRCH_GT, SRCH_LT, SRCH_GE, SRCH_LE, 和 ARRAY_MOVE； BIT也可用于 ARRAY_MOVE。

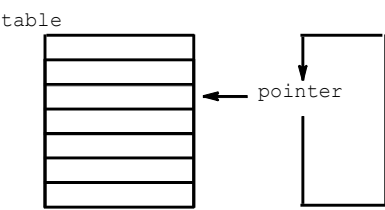
注意

REAL型数据不可用于数据表功能块，然而，可以使用DWORD数据来运算实型数据。

数据表所允许的最大长度为32,767个元素，但是 SORT功能块被限制为64元素。

TBLRD (INT, UINT, DINT, WORD, DWORD)

表读出功能块(TBLRD)用来顺序地读出一个表中的值，该表永远不会变满。当指针达到表底后会自动回绕到表的起始位置。



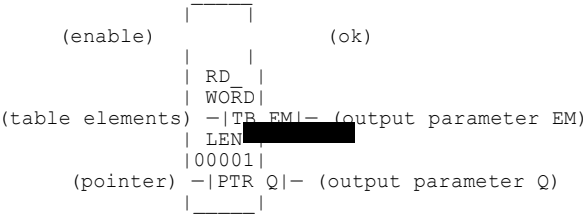
- 1. 功能块指针值增1。
- 2. 将指针所指数据拷贝到输出参数Q中，必须用附加程序逻辑从输出变量中获取数据。
- 3. 每次执行该指令时，重复步骤1和2，直至表空为止(PTR= LEN)。
- 4. 当表满以后，指针回到表的起始位置。

注意

TBLRD 和 TBLWRT功能块可以对同一个表或不同的表进行操作。通过给指针指定不同的变量，这些功能块可以在不同的位置或以不同的速率对同一数据进行存取。

TBLRD功能块有3个输入参数和3个输出参数。当功能块选通，指针(PTR)增加1。如果新的指针位置指向表中最后一个元素，则输出EM置为ON。下次该功能块执行时，指针会自动回到1。指针增加后，新指针所指向的位置的内容被拷贝到输出Q。

该功能块一旦得电，就向右传递电流。



参数:

参数	说明
enable	使能时，执行表读出。
TB	TB包含了表中的元素。
PTR	PTR 指针增1，并指向下一被读的元素。
ok	功能块使能则OK为ON。
EM	当读取表中最后一个元素时，输出EM为ON。
Q	从表中读出的数据。
LEN	LEN长度必须在 1到 32,767范围内。

有效内存类型:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
TB	•	o	o	o	o	Δ	o		•	•	•	•	•	•		
PTR		•	•	•	•		•		•	•	•	•	•	•		
ok	•															•
EM	•															•
Q	•	o	o	o	o	Δo	o		•	•	•	•	•	•		

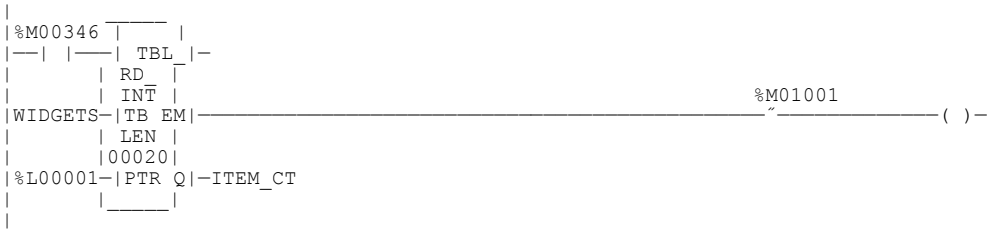
注意: 间接变量对所有寄存其类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
• 可使电流流经功能块的地址或变量。
o 对INT, UINT,或 WORD数据有效，但对DINT或DWORD无效。
Δ 对WORD 数据有效，但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

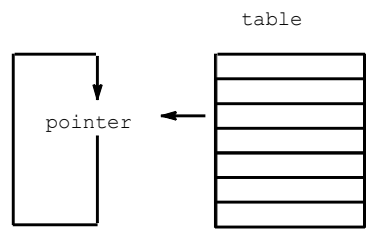
例:

本例中， WIDGETS 为一个有20个元素的数据表，当输入%M00346 为ON时，指针增1同时表中上一个元素被拷贝到ITEM-CT中。%L00001起到了进入此数据表的指针的作用。%M01001用于当表中数据全部被存取后发出信号。



TBLWRT (INT, UINT, DINT, WORD, DWORD)

表写入功能块 (TBLWRT)用于对一个永远不会填满的表中的值进行连续更新。当指针达到表底部时，它会自动回到表起始处。

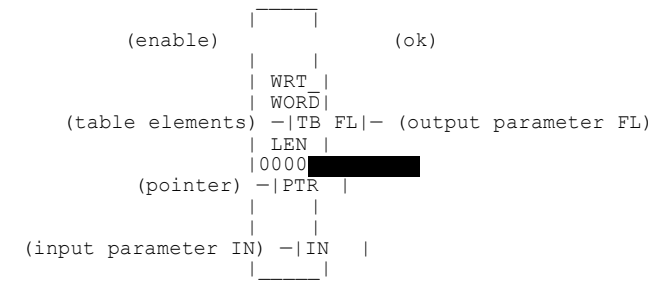


- 1. 功能表指针增1。
- 2. 功能块把由输入参数IN输入的数拷贝至当前指针所指的表格中。（它将改写该位置的当前值）必须用其他逻辑程序将数据送入输入变量。
- 3. 每次执行该指令时，重复步骤1和2，直至表满为止(PTR= LEN)。
- 4. 当表满以后，指针回到表的起始位置。

注意

TBLRD 和 TBLWRT功能块可以对同一个表或不同的表进行操作。通过给指针指定不同的变量，这些功能块可以在不同的位置或以不同的速率对同一数据进行存取。

TBLWRT功能块有4个输入参数和2个输出参数。当功能块选通，指针增1。如果新的指针位置指向表中最后一个元素，则输出FL置为ON。下次该功能块执行时，指针会自动回到1。指针增加后，将输入变量的内容写入到新指针所指向的位置覆盖原来存储在那里的数据。该功能块一旦得电，就向右传递电流。



参数:

参数	Description
enable	使能时，执行表读出。
TB	TB包含了表中的元素。
PTR	PTR 指针增1，并指向下一被写的元素。
ok	功能块使能则OK为ON。
FL	当写入表中最后一个元素时，输出FL为ON。
IN	写入表中的数据。
LEN	LEN长度必须在 1到 32,767范围内。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable		•														
TB		o	o	o	o	Δ†	o		•	•	•	•	•	•		
PTR		•	•	•	•		•		•	•	•	•	•	•		
IN	•	o	o	o	o	Δ†	o		•	•	•	•	•	•		
ok	•															•
FL	•															•

注意: 间接变量对所有寄存其类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
• 可使电流流经功能块的地址或变量。
o 对INT, UINT,或 WORD数据有效，但对DINT或DWORD无效。
Δ 对WORD 数据有效，但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

例:

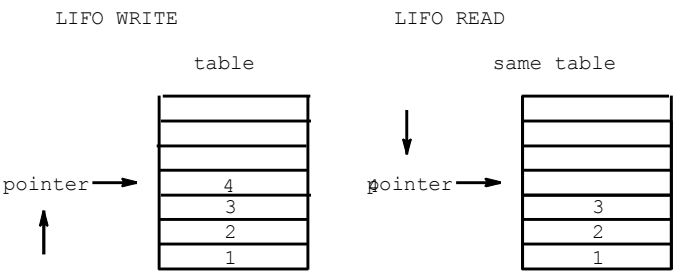
此例中，WIDGETS为含有20个整型数的表。当使能输入%I00012为ON，指针值增加1，并将%P00077中的内容写入到当前指针所指向的位置。%L00001作为指针进入数据表。



LIFORD

(INT, UINT, DINT, WORD, DWORD)

后人先出 (LIFO)和后如先出读功能块(LIFORD) 用于从表中移出数据。数值总是从表的顶端移出。如果指针到达最后一个元素位置且表已满，则必须使用LIFORD功能块来移动指针所在的条目并使指针值减1。LIFORD功能块应与LIFOWRT配合使用，后者使指针增加并向表中写入数据。



参数:

参数	Description
enable	使能时，执行表读出。
TB	TB包含了表中的元素。
PTR	PTR 指针增1，并指向下一被写的元素。
ok	功能块使能则OK为ON。
EM	当读出表中最后一个元素时，输出EM为ON。
Q	读出的表中的数据。
LEN	LEN长度必须在 1到 32,767范围内。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
TB	•	o	o	o	o	o	Δ†	o		•	•	•	•	•	•	
PTR		•	•	•	•		•		•	•	•	•	•	•		
ok	•															•
EM	•															•
Q	•	o	o	o	o	o	Δ†	o		•	•	•	•	•	•	

注意: 间接变量对所有寄存其类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
• 可使电流流经功能块的地址或变量。
o 对INT, UINT,或 WORD数据有效，但对DINT或DWORD无效。
Δ 对WORD 数据有效，但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

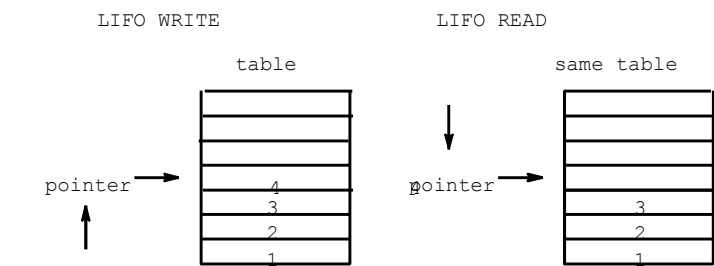
例:

此例中，PRODUCT为一个LIFO表，它有100个字的元素。当使能输入PACK_IT为ON，表顶端的数据被拷贝到由CART表示的变量中。由STK_PTR来确定该表的指针。输出线圈EMPTY当表空时得电。



LIFOWRT (INT, UINT, DINT, WORD, DWORD)

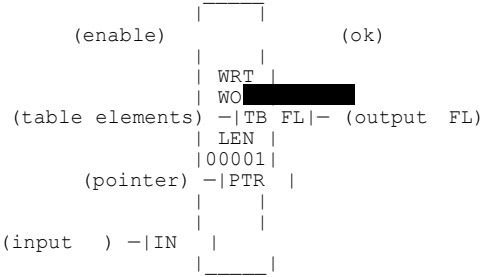
后人先出 (LIFO)和后如先出写功能块(LIFOWRT) 用于增加指针值并向表中移入数据在表的指针上方增加条目。数值总是从表的顶端移入。如果指针到达最后一个元素位置且表已满，则不能有更多的值再由LIFOWRT功能块添加进表中。必须使用LIFORD功能块移除指针处的条目并使指针减1。



- 执行时序：
1. 指针值增1。
 2. 将输入参数IN数据拷贝至指针指向的位置，它将覆盖以前的存储值。然后必须使用另外的逻辑程序将数据放到其输入变量中。
 3. 每次执行该指令时，重复步骤1和2，直至表空为止(PTR= LEN)。
 4. 表满时，指针不返回起始位置。

LIFOWRT功能块有4个输入参数和2个输出参数。当功能块选通，指针增1同时将新的数据拷贝至指针所指向的位置。如果此时新的指针指向表的底端，则没有数据写入且不会有电流流向右侧。指针总是指向表中最近输入的一个元素。如果表满，则不可以再向此表中添加条目。

若成功执行，该功能块向右传递电流。



参数:

参数	说明
enable	使能时, 执行表写入。
TB	TB包含了表中的元素。
PTR	PTR 指针增1, 并指向下一被写的元素。
ok	功能块使能PTR < LEN, 则OK为ON。
FL	当写入表中最后一个元素时, 输出FL为ON。
IN	写入表中的数据。
LEN	LEN长度必须在 1到 32,767范围内。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
TB	•	o	o	o	o	o	Δ†	o		•	•	•	•	•	•	
PTR		•	•	•	•		•		•	•	•	•	•	•		
IN	•	o	o	o	o	o	Δ	o		•	•	•	•	•	•	
ok	•															•
FL	•															•

注意: 间接变量对所有寄存其类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
 • 可使电流流经功能块的地址或变量。
 o 对INT, UINT,或 WORD数据有效, 但对DINT或DWORD无效。
 Δ 对WORD 数据有效, 但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制, 见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

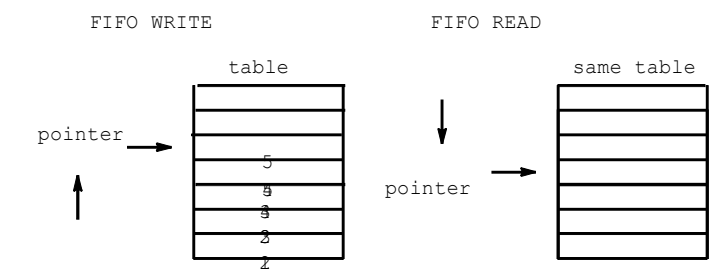
例:

此例中, PRODUCT为有100个字元素的LIFO数据表。当使能输入STORE为ON, 一个来自NEW_ITEM的数据被拷贝到由 STK_PTR所指向的表的位置。当PTR = LEN, 输出 FL 将通过电流使线圈FULL 得电。在数据被拷贝出之前不能再有来自NEW_ITEM的数据通过LIFORD功能块添加到表中。



FIFORD (INT, UINT, DINT, WORD, DWORD)

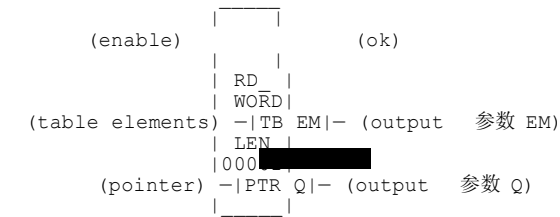
先入先出读功能块(FIFORD)用于从数据表移出数据。数据值总是从数据表底部移出。如果指针到达最后一项而该表变满时，就必须用FIFORD功能块移动该指针位置的条目并使指针减1。该功能块应与FIFOWRT功能块配合使用，后者可使指针增加并向表中写入数据。



执行时序

1. 功能块把此表顶部数据拷贝到输出Q中。然后必须使用其他的程序逻辑将数据放入其输入变量中。
2. 此表中剩余项被拷贝到表中较低序号的位置处。
3. 指针减1。
4. 每次执行该指令均重复步骤1和2，直到表空(PTR = LEN)为止。
5. 表满后，指针并不返回到起始位置。

FIFORD功能块有3个输入参数和3个输出参数。当功能块选通，将指针所指向的数据拷贝至输出Q同时指针减1。接下来表中的元素依次向下移动一个位置，即首先由表中位置2的数据移至位置1。最后指针减1。如果此时新的指针值为0，则输出EM置为ON，因此EM指示了表是否为空。该功能块读取的指针数据在表的范围内，就向右传递电流。



参数:

参数	说明
enable	使能时，执行表读出。被读的总是FIFO表第一个元素。
TB	TB包含了表中的元素。
PTR	PTR 指针增1，并指向下一被写的元素。
ok	功能块使能则OK为ON。
EM	当读出表中最后一个元素时，输出EM为ON。
Q	读出的表中的数据。
LEN	LEN长度必须在 1到 32,767范围内。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
TB	•	o	o	o	o	Δ	o		•	•	•	•	•	•		
PTR		•	•	•	•		•		•	•	•	•	•	•		
ok	•															•
EM	•															•
Q		•	o	o	o	o	Δ†	o		•	•	•	•	•	•	

注意: 间接变量对所有寄存器类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
 • 可使电流流经功能块的地址或变量。
 o 对INT, UINT,或 WORD数据有效，但对DINT或DWORD无效。
 Δ 对WORD 数据有效，但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

例:

此例中，PRODUCT为一个有100个字的 FIFO 表，当使能输入PACK_IT为ON，PRODUCT由指针STK_PTR所指的表中位置处的数据被拷贝至由CART制定的变量地址中，该表指针应指向底部，即最早进入表的数据项处。在STK_PTR中的值将减少。在PRODUCT表中最早的数据项的拷贝将放在每个表位置的后面，作为下一次PACK_IT触发时被拷贝出的当前值。当PTR=0后，输出点EM将通过电流从而接通线圈EMPTY。如果没有再使用FIFOWRT功能首先把数据写入表中，就不会从该表再读出数据。

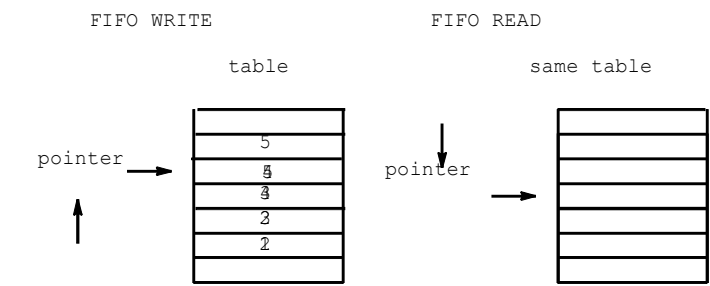
```

| PACK_IT |-----|
|-----|-----| FIFO |-----|
|          | RD  |
|          | WORD|
| PRODUCT--| TB EM|-----"-----" EMPTY ( )-----|
|          | LEN |
|          | 00100|
| STK_PTR--| PTR Q|--CART
|          |-----|

```

FIFOWRT (INT, UINT, DINT, WORD, DWORD)

先入先出写功能块 (FIFOWRT)用来指针增1同时将一个条目加到表中新的指针位置。数据总是从表底部移入。如果指针已达到最后位置且表已满，则FIFOWRT将不再向此表写入数据。接下来必须使用FIFORD功能块移动在指针位置处的入口并使指针减1。



动作时序:

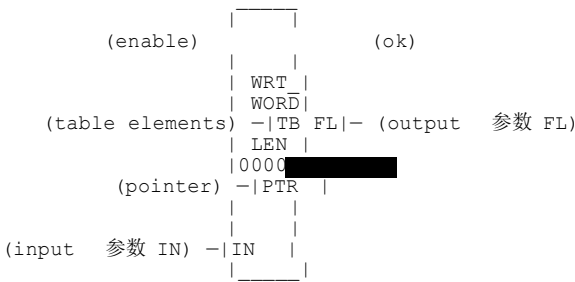
1. 指针增1。

2. 把输入参数IN中的数据拷贝到由指针指向的表的位置处（将覆盖该位置的当前值）。应使用其他程序逻辑将数据放入输入变量中。

3. 每次执行该指令均重复步骤1和2，直到表空(PTR = LEN)为止。

4. 表满后，指针并不返回到起始位置。

FIFOWRT功能块有4个输入参数和2个输出参数。当功能块选通，指针增1同时将新的数据拷贝至指针所指向的位置。如果此时新的指针指向表的底端，则没有数据写入且不会有电流流向右侧。指针总是指向表中最近输入的一个元素。如果表满，则不可以再向此表中添加条目。FIFOWRT功能块成功执行将向右传送电流。



参数:

参数	说明
enable	使能时，执行表写入。
TB	TB包含了表中的元素。
PTR	PTR 指针增1，并指向下一被写的元素。
ok	功能块使能PTR < LEN，则OK为ON。
FL	当写入表中最后一个元素时，输出FL为ON。
IN	写入表中的数据。
LEN	LEN长度必须在 1到 32,767范围内。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable		•														
TB		o	o	o	o	Δ†	o		•	•	•	•	•	•		
PTR		•	•	•	•		•		•	•	•	•	•	•		
IN	•	o	o	o	o	Δ	o		•	•	•	•	•	•	•	
ok	•															•
FL	•															•

注意: 间接变量对所有寄存其类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
• 可使电流流经功能块的地址或变量。
o 对INT, UINT,或 WORD数据有效，但对DINT或DWORD无效。
Δ 对WORD 数据有效，但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制” 部分内容。

例:

此例中，PRODUCT为100个字元素的FIFO 表。 当使能输入UNPACK为ON，来自P_CODE的一个数据项被拷贝到指针STK_PTR指向的位置。当PTR = LEN，输出FL将接通使线圈FULL得电。如果不使用FIFO功能块读出表中第一个数据，则P_CODE将不再有数据写入表中。



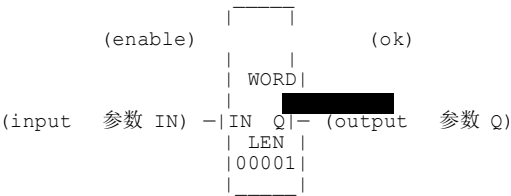
SORT (INT, UINT, WORD)

SORT功能块用于对一组数按照升序排序。它有2个输入参数和2个输出参数。EN为布尔型使能输入，IN为待排序的数列。OK为一个布尔型输出用于提供电流，Q是一个整型数组，它给出了已被排序的元素在原数列中的索引。Q的大小与IN相同。另外SORT还有一个待排序元素数量参数(LEN)。

SORT功能块最多只能对含64个元素的数列进行排序。当EN为ON时，IN中所有元素按升序排序。为了给出各个已排序的元素在未排序数组中的原来位置而生成了Q数列。OK始终为ON状态。

注意

不要使用SORT功能块于定时或触发输入程序块。



参数:

参数	说明
enable	使能时，开始排序操作。
IN	待排序数列。 排序结果又写回IN中。
ok	当LEN小于65，OK即为ON。但编程器本身不对长度作64的限制。如果LEN超过64，则SORT功能块只对前64个元素排序。
Q	输出 Q包含一个索引数组，它给出排序后各元素在原数组中的位置。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN		•	•	•	•		•		•	•	•	•	•	•		
ok	•															•
Q	•	•	•	•	•		•		•	•	•	•	•	•		

注意: 间接变量对所有寄存其类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
• 可使电流流经功能块的地址或变量。

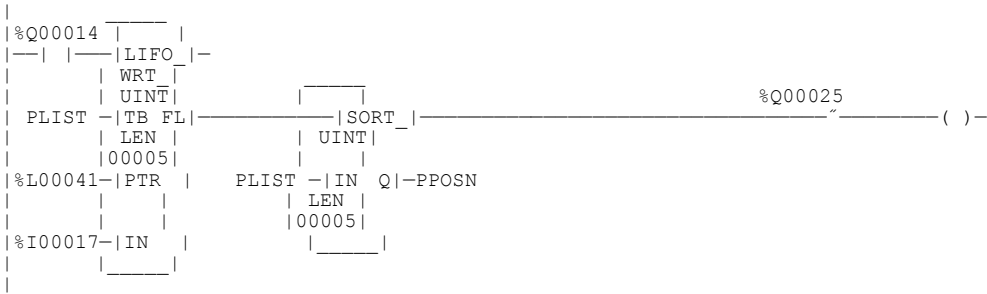
注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

例:

此例中，每次%Q00014为ON，新的部件号 (%I00017 - %I00032)被加入到PLIST表中。当数组被填入后，便开始排序工作，同时使输出%Q00025置为ON。数列PPOSN含有排序后的元素在PLIST中的原始位置。

如果 PLIST含有5个元素，排序前为25, 67, 12, 35, 14 ，排序后其数据为12, 14, 25, 35, 67。PPOSN中得值为3, 5, 1, 4, 2。



ARRAY_MOVE (INT, UINT, DINT, BIT, BYTE, WORD, DWORD)

数列移动功能块 (ARRAY_MOVE) 用于将指定数量数据从源数据列中拷贝到目的数列。

ARRAY_MOVE功能块有6个输入参数和2个输出参数。当功能块选通，在计数指示器(N)中指定数量的元素将被从由索引位置(SR + SNX - 1)开始的输入数列中提取出来。这些元素将被写入由(DS + DNX - 1)开始的输出数列中。操作数LEN说明组成每个数组的元素。

对于ARRAY_MOVE_BIT，当源数列和/或目的数列起始地址的参数选用与字有关的内存时，指定字的最低有效位将作为该数列的第一位。所显示的值包括16位而不管该数列的长度。

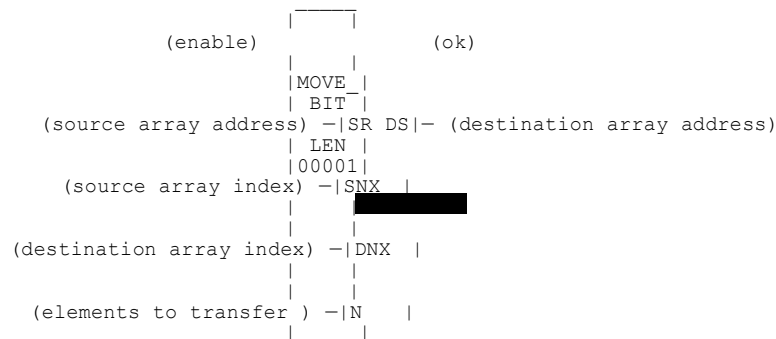
在ARRAY_MOVE指令中的索引都是基于1的。在使用ARRAY_MOVE时，由起始地址和长度确定的源数列和目的数列以外的元素不能被赋予变量地址。

OK输出将得电，除非有以下情况发生：

使能为OFF。

(N + SNX)大于LEN。

(N + DNX)大于 LEN。



参数:

参数	说明
enable	使能时，功能块执行。
SR	SR包含了源数列的起始地址。对于 ARRAY_MOVE_BIT，可以使用任何变量，它不需进行字节调整。但是，将在线显示起始于指定地址的1位。
SNX	SNX包含源数列的索引。
DNX	DNX包含目的数列的索引。
N	N为计数指示。
ok	功能块使能则OK为ON，除非NX超出范围
DS	含有目的数列的起始地址。对于ARRAY_MOVE_BIT，可使用任何变量，它无需字节调整。但是将在线显示指定地址的起始位。
LEN	LEN 指明了由SR和 DS开始的元素数。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
SR	•	o	o	o	o	Δ	o	o	•	•	•	•	•	•		
SNX	•	•	•	•	•		•		•	•	•	•	•	•	•	
DNX	•	•	•	•	•		•		•	•	•	•	•	•	•	
N	•	•	•	•	•		•		•	•	•	•	•	•	•	
DS	•	o	o	o	o	†	o	o	•	•	•	•	•	•		
ok	•															•

- 间接变量对所有寄存器类型都有效。对于 ARRAY_MOVE_MOVE_BIT，离散量 %I, %Q, %M, %T不需字节调整。
- o 对于 INT, UINT, BIT, BYTE, WORD型数据有效，而 DINT 或 DWORD。
- %U仅对 ARRAY_MOVE_BIT 有效。
- Δ 对于 BIT, BYTE, WORD 有效，而对INT, UINT, DINT, DWORD无效。
- † 仅对%SA, %SB, %SC有效，而%S无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制”部分内容。

例 1:

此例中， 数列%R00001 - %R00016中的%R00003 - %R00007 被读出并写入%R00100 - %R00115数列中的 %R00104 - %R00109。

```

| %I00001 | _____ |
|-----| |-----| ARRAY |-----|
|          | MOVE |
|          | WORD |
| %R00001 | SR DS | %R00100
|          | LEN |
|          | 00016 |
| CONST  - | SNX |
| 00003   |     |
|          |     |
| CONST  - | DNX |
| 00005   |     |
|          |     |
| CONST  - | N   |
| 00005   | _____ |
|          |

```

例 2:

SR和DS使用了位型内存， %M00009 - %M00024数列中的%M00011 - %M00017被读出并写入%Q00022 - %Q00037数列中的%Q00026 - %Q00032。

```

| %I00001 | _____ |
|-----| |-----| ARRAY |-----|
|          | MOVE |
|          | BIT  |
| %M00009 | SR DS | %Q00022
|          | LEN |
|          | 00016 |
| CONST  - | SNX |
| 00003   |     |
|          |     |
| CONST  - | DNX |
| 00005   |     |
|          |     |
| CONST  - | N   |
| 00007   | _____ |
|          |

```

例 3:

SR和DS使用了字型内存，先读出%R00001中的全部16位和%R00002中的4位组成数列中从%R00001第3有效位到%R00002中第2个最低有效位，然后写入由%R00100中的全部16位和%R00101中4位组成数列中的从%R00100第5有效位到%R00101第4位有效位中。

```

|
| %I00001 |-----|
|-----| |-----| ARRAY |-----|
|          | MOVE |
|          | BIT  |
| %R00001 |-----| SR DS |-----| %R00100
|          | LEN  |
|          | 00020 |
| CONST  -| SNX  |
| 00003  |      |
|          |      |
| CONST  -| DNX  |
| 00005  |      |
|          |      |
| CONST  -| N    |
| 00016  |-----|
|

```

SRCH_EQ和SRCH_NE (INT, UINT, DINT, BYTE, WORD, DWORD) SRCH_GT 和 SRCH_LT SRCH_GE 和 SRCH_LE

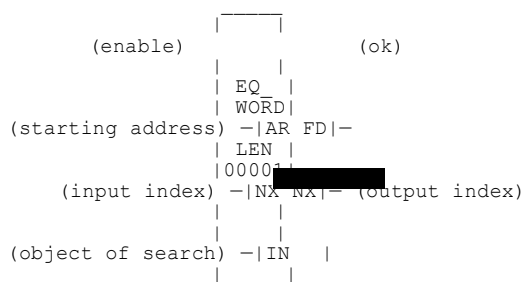
使用下表中列出的相应查找功能可实现对数列中的值的特定操作。

缩写	功能	说明
SRCH_EQ	查找相等	查找与给定值相等的数列中的值。
SRCH_NE	查找不相等	查找与给定值不相等的数列中的值。
SRCH_GT	查找大于	查找大于给定值的数列中的值。
SRCH_GE	查找大于或等于	查找大于或等于给定值的数列中的值。
SRCH_LT	查找小于	查找小于给定值的数列中的值。
SRCH_LE	查找小于或等于	查找小于或等于给定值的数列中的值。

以上各功能块均有4各输入参数和2各输出参数。当功能块选通，便开始从(AR + 输入NX)位置进行查找。也即从数列的起始地址(AR)与此数之和的索引量（输入NX）处开始查找。

查找工作将一直进行到发现所查找目标(IN)或到达数列末尾为止。如果找到符合条件的元素，输出参数(FD)将置为ON，同时输出参数(NX)将输出此元素在该数组中的相应位置。如果指定数列结束都未找到符合条件的元素，则输出参数FD将置为OFF且输出NX置为ON。

输入NX的有效范围是0到LEN - 1，为从数组的第1各元素开始查找，NX应被置为0。在执行此功能时，该值会增1，因此输出NX范围为1到LEN。如果输入NX的值不在有效范围内(< 0 或 ≥ LEN)，那么输出NX被置为默认值0。



参数:

参数	说明
enable	使能时，开始执行查找。
AR	包含被查找数列的起始地址。
Input NX	含有此数列的索引，查找从此处开始。
IN	被查找的对象。
Output NX	包含所找到的元素在数组中的位置。
FD	表示已发现一个数组元素符合条件
ok	执行过程中无错误发生且NX无超界，则为ON。
LEN	指明从AR开始的数列元素个数，应介于1与32,767字或字节间。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
AR	•	o	o	o	o	Δ	o		•	•	•	•	•	•		
NX in	•	•	•	•	•		•		•	•	•	•	•	•	•	
IN	•	o	o	o	o	Δ	o		•	•	•	•	•	•		
NX out	•	•	•	•	•		•		•	•	•	•	•	•	•	
FD	•															•
ok	•															•

• 可使电流流经功能块的变量或地址。
o 对于INT, UINT, BYTE, WORD 有效但对于 DINT 或DWORD无效。
Δ 对于BYTE,or WORD有效但对于INT, UINT, DINT或 DWORD无效。.

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制” 部分内容。

例 1:

数列AR定义在内存地址%R00001 - %R00005上。当EN为ON时，功能块将在%R00004和%R00005查找与IN相等的元素。如果%R00001 = 7, %R00002 = 9, %R00003 = 6, %R00004 = 7, %R00005 = 7, %R00100 = 7, 那么查找工作将从%R00004开始，FD变为ON，同时数值4被写入%R00101。

```

| %I00001 | _____ |
|-----| |-----| SRCH |-----|
|          | EQ      |
|          | INT      |
| %R00001 | AR FD |-----| %Q00001 |-----| ( )-----|
|          | LEN      |
|          | 00005 |
| CONST  -| NX NX |-%R00101
| 00003   |
| %R00100 | IN      |
|          | _____ |

```

例 2:

数列AR定义在内存地址%AI0001 - %AI0016上。数列元素中的值为100, 20, 0, 5, 90, 200, 0, 79, 102, 80, 24, 34, 987, 8, 0, 500。起初%AQ0001为5，当EN为ON，每次扫描时将搜寻数列中为0的元素。第一次扫描时将从%AI0006开始，并在%AI0007找到与之相匹配的值，于是FD为ON，而%AQ0001为7。第二次扫描将从%AI0008开始并在%AI0015找到0，于是FD仍为ON，而%AQ0001为15。第三次扫描将从%AI0016开始，由于直到数列结束也再没有0，故FD变为OFF而%AQ0001被置为0。下次扫描将从数组的起始处开始查找。

```

| %I00001 | _____ |
|-----| |-----| SRCH |-----|
|          | EQ      |
|          | WORD    |
| %AI0001 | AR FD |-----| %M00001 |-----| ( )-----|
|          | LEN      |
|          | 00016 |
| %AQ0001 | NX NX |-%AQ0001
|          |
| CONST  -| IN      |
| 0000   | _____ |

```

ARRAY RANGE (INT, DINT, WORD, DWORD)

ARRAY RANGE功能块用于确定一个单输入值是否落在两个分别规定了上下限的数列之间。输出的将是一个位数列，当输入值大于或等于下限且小于等于上限值时，此位数列的各个位将置为ON。当输入值超出上下限或下限值大于上限值时，输出将为OFF。

注意

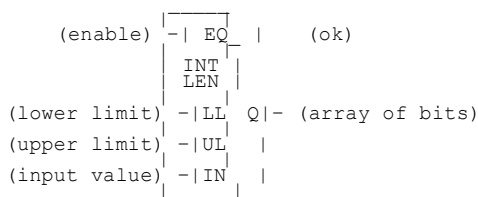
此功能块仅适用于版本5或以上的CPU。

ARRAY RANGE功能块的操作数可有以下类型：

Data Type	说明
INT	带符号整型
DINT	双精度带符号整型
UINT	无符号整型
WORD	字
DWORD	双字

默认的数据类型为带符号整型，可以在选择此功能块后再对其进行修改。有关数据类型的更详细说明，参加第2-16页“数据类型”部分内容。

当功能块使能，ARRAY RANGE功能块把输入参数IN中的值与LL和UL进行比较。对于IN值大于等于LL且小于等于UL，输出Q置一位ON。对于输入值超出范围或LL超出UL中的值时，输出Q设置一位OFF。如果功能块成功执行，输出OK将得电。



参数:

参数	说明
enable	使能为ON后，开始执行该功能。
LL	所给的范围的下限。
UL	所给的范围的上限。
IN	被比较范围的值。
ok	如果无错误发生，OK为ON。
Q	当IN中的值落在LL和UL范围内时，输出Q为ON。 注意： Q 并不作调整，它以位格式显示。对于第一个元素显示0 (OFF)或1 (ON)。对于离散变量，它将代表显示的变量。对于字变量，它将代表所显示的变量的低位。
LEN	数列中含有的元素个数。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
LL		o	o	o	o		o		•	•	•	•	•		•†	
UL		o	o	o	o		o		•	•	•	•	•		•†	
IN		o	o	o	o		o		•	•	•	•	•			
ok		o	o	o	o		o		•	•	•	•	•			
Q	•															•

注意: 间接变量对所有寄存器类型均有效(%R, %P, %L, %AI, %AQ, and %UR).
— 可使电流流经功能块的地址或变量。
o 对INT, UINT,或 WORD数据有效，但对DINT或DWORD无效。.
Δ 对WORD 数据有效，但对INT, UINT, DINT,或 DWORD.无效。

注意

对于在参数子程序中的限制，见第二章第二节“参数子程序中的形式参数的使用限制” 部分内容。

例 1:

此例中，从%R00001到%R00008的下限(LL)值为1, 20, 30, 100, 25, 50, 10和200。%R00100到%R00108的上限(UL)值为40, 50, 150, 2, 45, 90, 250和47。输出Q值将被放入%R00200的前8位。Q中位值从低到高的次序为: 1, 1, 1, 0, 1, 0, 1, 和0。对于%R00200的低位所显示的位值被置为ON (1)。输出OK将置为ON (1)。

```

| %I00001 | _____ | %Q00001
| ---|
| ---| ARRAY | _____ ~ _____ ( ) ---
|         | RANGE |
|         | INT  |
|         | _____ |
| %R00001 | LL  Q | %R00200
|         | LEN  |
|         | 00008 |
| %R00100 | UL  |
|         | _____ |
|         | 40  | IN  |
|         | _____ |

```

例 2:

此例中，下限(LL)数列为%T00001到%T00016, %T00017到%T00032, %T00033 到 %T00048。下限为100, 65和1。上限值为29, 165和2。输出Q的值0, 1和0被放入 %Q00001到%Q00003中。所显示的位值将为0 (OFF)，它代表了%Q00001的值。输出 OK为ON。

```

| %I00001 | _____ | %M00001
| ---|
| ---| ARRAY | _____ ~ _____ ( ) ---
|         | RANGE |
|         | INT  |
|         | _____ |
| %T00001 | LL  Q | %Q00001
|         | LEN  |
|         | 00003 |
| %T00049 | UL  |
|         | _____ |
|         | 65  | IN  |
|         | _____ |

```

Chapter 11

第十一章 转换功能块

使用转换功能块可以将一个数据项从一种类型变换为另一种类型。许多程序指令，如，数学运算指令，只能使用某一类型的数据。本章主要介绍以下各转换功能块：

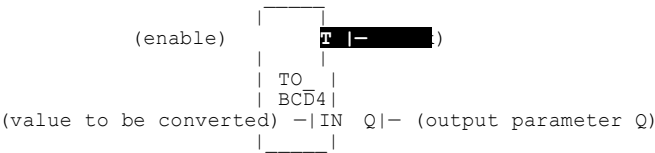
缩写	功能	说明	页码
BCD-4	转换为BCD-4型	将无符号或带符号整型转换为4位BCD码。	11-2
BCD-8	转换为BCD-8型	将双精度带符号整型转换为8位BCD 码。	11-4
UINT	转换为无符号整型	将BCD-4型，带符号整型，或双精度整型转换为无符号整型。	11-6
INT	转换为带符号整型	将BCD-4型，无符号整型，或双精度整型转换为带符号整型。	11-8
DINT	转换为双精度带符号整型	将BCD-4型，带符号整型，或无符号整型转换为双精度整型。	11-10
REAL	转换为实型	将BCD-4型，BCD-8型，无符号整型，带符号整型，或双精度整型转换为实型。	11-12
TRUN	舍去	舍去实型数的小数部分。	11-14

BCD-4 (INT, UINT)

该功能块用于将无符号或带符号整型数据转换为等值的4位BCD码输出。它并不改变原来的值。输出数据可直接作为其他程序功能块的输入。

数据能够被转换成BCD码格式以驱动BCD编码的LED显示或供给外部设备，如高速计数器。

当此功能块接通时，它便开始进行转换，并将结果送输出Q。此功能块得电便向右传递电流，除非转换结果超出了0到9999的范围。



参数:

参数	说明
enable	当功能块使能，开始进行转换。
IN	需转换为BCD-4码的整型数值的一个变量。
ok	转换中无错误发生，OK即为ON。
Q	IN中的值转换为BCD-4码后的结果。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	•	•	•	•		•		•	•	•	•	•	•	•	
ok	•															•
Q	•	•	•	•	•		•		•	•	•	•	•	•		

注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。

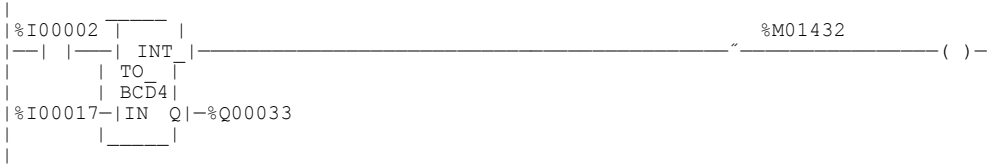
• 可使电流流经功能块的变量或地址。

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

本例中，当输入%I00002置位且无错误产生时，位于%I00017到%I00032中的整型数将被转换为4位BCD码，并将此转换结果存至%Q00033到%Q00048。线圈%M01432用于检测转换工作是否成功。



BCD-8 (DINT)

该功能块用于将无符号或带符号整型数据转换为等值的8位BCD码输出。它并不改变原来的值。输出数据可直接作为其他程序功能块的输入。

当此功能块接通时，它便开始进行转换，并将结果送输出Q。此功能块得电便向右传递电流，除非转换结果超出了0到99999999的范围。



参数:

参数	说明
enable	当功能块使能，开始进行转换。
IN	需转换为BCD-8码的整型数值的一个变量。
ok	转换中无错误发生，OK即为ON。
Q	IN中的值转换为BCD-8码后的结果。

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.	.	.	.	.		.		.	.	.	.	.	.	.	
ok	.															.
Q	.	.	.	.	.		.		.	.	.	.	.	.		

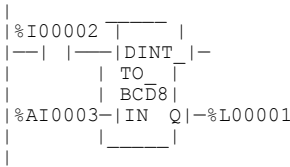
注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。
可使电流流经功能块的变量或地址。

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

此例中，只要输入 %I00002 为ON且无错误存在，位于%AI0003 中的带符号整数将转换为8位BCD码将结果存放于%L00001 到 %L00002中.

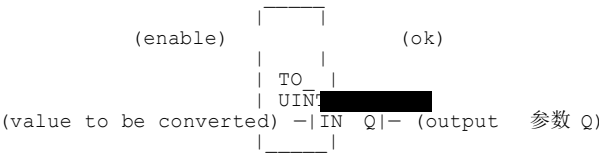


UINT (INT, DINT, BCD-4, REAL)

该功能用于输出带符号整型、双精度带符号整型、BCD-4或实型数据的等值无符号整型值。它并不改变原数据。转换的结果可直接作为其它功能块的输入。

→UINT功能的一个用途就是将 I/O结构中的BCD码数据转换成整型数据并存储到内存中。这可以为BCD 指型轮或外部BCD 指型轮或外部BCD电器元件，如高速计数器或位置编码器，提供一个接口。

当它被选通后，便开始转换并通过Q输出结果。当此功能被选通便向右传递电流，除非转换结果超出0到65,535这个范围。



参数:

参数	说明
使能enable	使能为ON后，开始转换工作
IN	包含待转换的无符号整型数值的变量
ok	转换无错误产生，OK为ON
Q	包含IN中数值的无符号整型形式

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	0	0	0	0		0		•	•	•	•	•	•	•	
ok	•															•
Q	•	•	•	•	•		•		•	•	•	•	•	•		

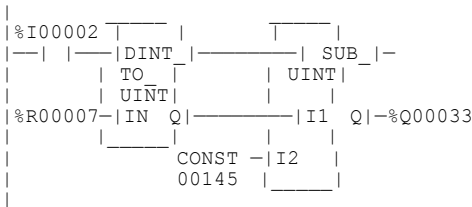
注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。
• 可使电流流经功能块的变量或地址。
0 对DINT_TO_UINT..无效

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

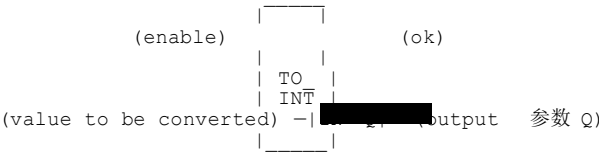
此例中，只要输入 %I00002 置位且不存在错误，则位于%R00007 中的双精度带符号整数将转换为一个无符号整型并传送给SUB（减法）功能块，此功能作减145的运算，其结果被放入到%Q00033。



INT (UINT, DINT, BCD-4, REAL)

该功能用于输出带符号整型、双精度带符号整型或BCD-4数据的等值带符号整型值。原来的数据并不会被该功能改变。其输出的结果可直接作为其它功能块的输入。

当它被选通后，便开始转换并通过Q输出结果。当此功能被选通便传递电流，除非数据超出范围。



参数表

参数	说明
enable	使能为ON后，开始转换工作
IN	包含待转换的带符号整型数值的变量
ok	只要enable为ON，OK为ON，除非数据超出范围
Q	包含IN中数值的带符号整型形式

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.	0	0	0	0		0		.	.	.	.	.	.	.	
ok	.															.
Q	.	.	.	.	.		.		.	.	.	.	.	.		

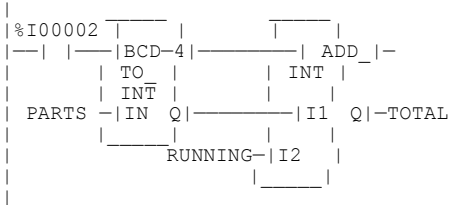
注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。
可使电流流经功能块的变量或地址。
O 对DINT_TO_UINT..无效

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

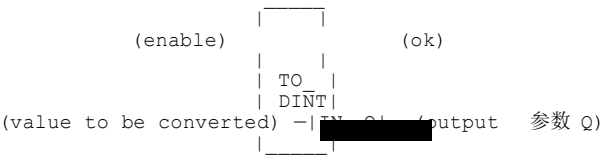
此例中，只要输入 %I00002 置位，在ARTS中的BCD-4 型数据被转换为带符号整型并传送给ADD（加法）功能块，与RUNNING中的数值相加，其和放入到TOTAL变量中。



DINT (INT, UINT, BCD-8, REAL)

该功能用于输出带符号整型、双精度带符号整型或BCD-8数据的等值带符号整型值。原来的数据并不会被该功能改变。其输出的结果可直接作为其它功能块的输入。

当它被选通后，便开始转换并通过Q输出结果。当此功能被选通便传递电流，除非数据超出范围。



参数:

参数	说明
enable	使能为ON后，开始转换工作
IN	包含待转换的带符号整型数值的变量
ok	只要enable为ON，OK为ON，除非数据超出范围
Q	包含IN中数值的带符号整型形式

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	•															
IN	•	0	0	0	0		0		•	•	•	•	•	•	•	
ok	•															•
Q	•								•	•	•	•	•	•		

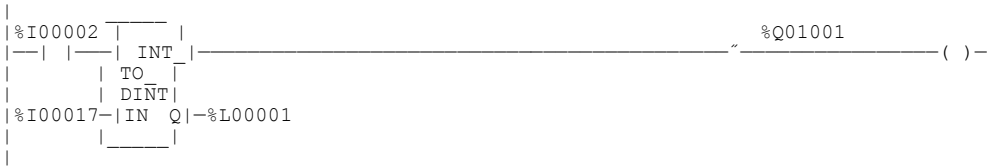
注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。
可使电流流经功能块的变量或地址。
O 对DINT_TO_UINT..无效

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

此例中，只要输入 %I00002 置位，位于输入%I00017处的整型数据被被转换为双精度带符号整型并将结果放入%L00001中，当此功能成功执行后，其输出%Q01001被置位。

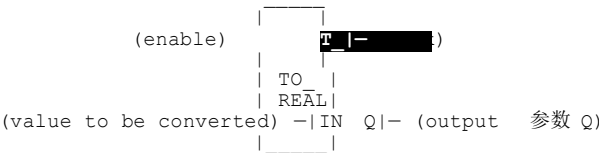


REAL (INT, UINT, DINT, BCD-4, BCD-8)

转换为实型功能块用于将输入的数据转换为实型输出。原来的数据并不会被该功能改变。其输出的结果可直接作为其它功能块的输入。

当它被选通后，便开始转换并通过Q输出结果。当此功能被选通便传递电流，除非数据超出范围。

从DINT或BCD-8转换为REAL时，有可能会损失精度，由于REAL的有效位被限制为24位。



参数:

参数	说明
enable	使能为ON后，开始转换工作
IN	包含待转换为REAL的整型数值的变量
ok	只要enable为ON，OK为ON，除非数据超出范围
Q	包含IN中数值的REAL型形式

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.	o	o	o	o		o		.	.	.	.	.	.	.	
ok	.															.
Q	.								.	.	.	.	.	.		

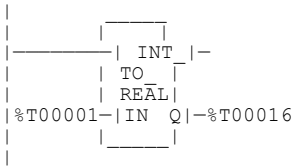
注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。
可使电流流经功能块的变量或地址。
o 对于DINT_TO_REAL或BCD8_TO_REAL无效。..

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

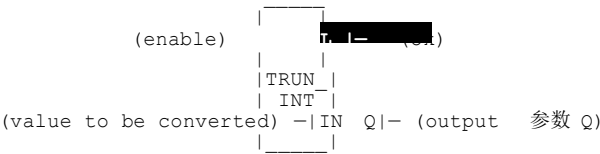
此例中，输入IN中的整型值为678。被存入%T00016的输出结果为678.000。



TRUN (INT, DINT)

舍余功能块用于将实型数字的小数部分舍去。原来的数据并不会被该功能改变。其输出的结果可直接作为其它功能块的输入。

当它被选通后，便开始转换并通过Q输出结果。当此功能被选通便传递电流，除非转换结果数据超出范围或IN为NaN（非数字）。



参数:

参数	说明
enable	使能为ON后，开始转换工作
IN	包含待转换为REAL的整型数值的变量
ok	只要enable为ON时无错误发生，OK为ON，除非转换结果数据超出范围或IN为NaN（非数字）。
Q	包含IN中数值的INT或DINT型形式

有效内存类型:

参数	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
IN	.								.	.	.	.	.	.	.	
ok	.															.
Q	.	o	o	o	o		o		.	.	.	.	.	.		

注意: 间接变量对所有的寄存器变量均有效 (%R, %P, %L, %AI, %AQ, %UR).。
可使电流流经功能块的变量或地址。
o 对于 REAL_TRUN_INT无效。..

注意

对于在参数子程序块中使用时的限制，见第二章第二节“参数子程序块中形式参数的限制”部分内容。

例:

此例中，所显示的实型常数被取整，其整型输出结果562被存入%T00001。

```
|
|          |-----|
|          | REAL |---
|          | TRUN |
|          | INT  |
|          |-----|
|          CONST -|IN  Q|-%T00001
|          5.62987E+02|-----|
|
```

本章将对于限制程序的执行和改变CPU执行应用程序的方式的控制功能块进行介绍。
(可参考第二章第一节关于CPU扫描的描述)

功能块	说明	
CALL	使程序转去执行另一特定的程序块	
CALL EXTERNAL	使程序转去执行另一特定的外部块	
CALL SUBROUTINE	使程序转去执行另一特定的参数子程序块	
DOIO	立即对指定范围内的输入输出进行一次扫描。(包括在DO I/O指令所声明的地址范围内的模块的全部输入或输出点均被扫描。不对局部I/O模块进行更新)。可选择的是, 已扫描I/O的拷贝可放在内部存储器中, 而不是在实际的输入点上。	
SUSIO	中止对所有正常I/O点的更新扫描(除了由DO I/O指令所制定的那些I/O)。	
MCR	编制一个主控继电器程序, MCR可使处于MCR和ENDMCR之间的所有逻辑行即使没有电流也可执行。	
END	用于指明其后的各级逻辑	

GFK-0265J

	MCR		在正常条件下运行。			
	JUMP	FOR, END- FOR, EXIT	使程序转向逻辑中循环执行某一逻辑位置（由LABEL指出）			
	LABEL		用于声明JUMP所转向的位置。			
	COM		在程序中加入一段注释。			

GFK-0265J
12-1

12

功能 块	说明	
SVC	请求进行下列PLC服务功	

GFK-0265J

12-2

REQ	能之一： 改变/读取定值扫描计时器 读窗口值 改变编程器通讯窗口状态 和时间 改变系统窗口状态和时间 改变任务模式和时间 改变/读取校验和任务状态 和字的序号 改变/读取时钟状态和值 复位Watchdog定时器 读取从扫描开始的扫描时 间 读取文件夹名称 读取PLC ID 读取PLC的运行状态 终止PLC 清除故障表 读取最后登录故障表的内 容 读取已过去的时间 屏蔽/非屏蔽I/O中断 读取I/O过载状态 设置运行允许/禁止 读故障表 登录用户定义的PLC故障 屏蔽/非屏蔽定时中断 读主校验和 允许/禁止EXE块校验和 任务选择开关 写入交换传输区域 读取交换传输区域 追加/继续I/O中断 ESCM端口状态	

GFK-0265J

GFK-0265J

GFK-0265J

PID	提供两种PID（比例/积分/微分）闭环控制算法：标准ISA PID算法和单独项算法	

CALL

使用CALL功能引导程序转去执行另一特定的程序块。可以在任何一个MAIN程序块，其他程序块，参数子程序块中使用CALL功能。

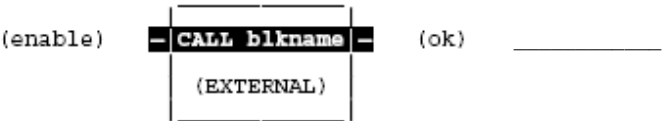
说明

一个程序块中最多允许调用功能块的最大数量为64。声明的程序块最大数量为255。允许调用的子程序最多为255。

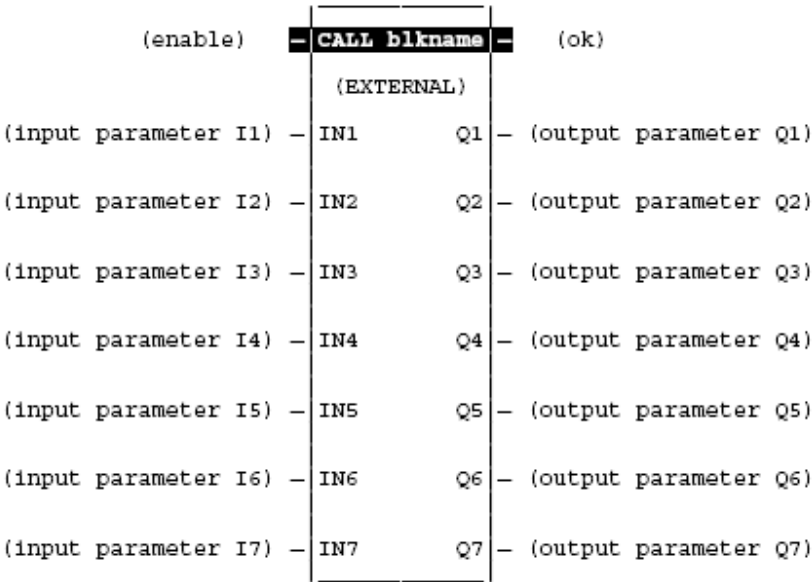
当CALL功能块得电后，它是扫描立即转到所指定的程序块并执行。执行完毕后，控制返回到CALL指令的下一条指令。

CALL EXTERNAL

使用CALL EXTERNAL功能块使程序转去执行在Logicmaster软件之外建立的程序块。八个CALL EXTERNAL 功能块的区别在于传送输入和输出数据块的参数不同。输入输出参数的长度可以是一位或一个字。当某个继电器指令连接了输入输出参数中的一个时，该参数长度为一位。所有的数据流长度将为一位。每个功能块均有使能（ENABLE）和OK参数。功能块被接通后，将执行指定的外部程序块。这个外部程序块使用向该块输入的数据，并产生输出数据。参数OK由外部块中的逻辑进行控制。程序块执行完毕后，控制返回到程序中CALL指令后面的指令。无输入和输出参数的CALL EXTERNAL功能块得形式如下：



CALL EXTERNAL功能可以有7对参数，如下图所示：



参数表:

参数	说明
使能	使能为ON后,开始执行该功能块
IN1-IN7	指明外部块中要使用的数据的起始地址
OK	由外部块中的逻辑控制OK输出
Q1-Q7	输出数据的起始地址

有效的数据类型

	f l o w		% Q	% M	% T	% S	% G	% U	% R	% P	% L	% A I	% A Q	% U R	c o n s t		n o n e

.可使电流流经功能块的变量或地址

例：

本例中，如果%I0001 置位后，名为C_BLK20的外部块将执行。C_BLK20块运行于参考地址%L00100中输入数据上，并产生输出数据放在%T0001中。C_BLK20中的逻辑控制Q1输出。



GFK-0265J
12-5

Chapter 12 Control Functions

12-11

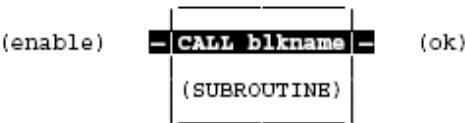
GFK-0265J

CALL SUBROUTINE

使用CALL SUBROUTINE 功能块使程序转去执行所指定的参数子程序。参数子程序块（PSBs）是用户自定义型功能块，它配有0~7对输入/输出参数。在程序执行时可以调用一个参数子程序块，但该参数子程序块必须在CALL指令之前先由块声明编辑器进行声明。有关声明子程序块的详情，请看本书关于第二章第三节和Logicmaster 90-70 软件用户手册（GFK-0263）中有关内容。

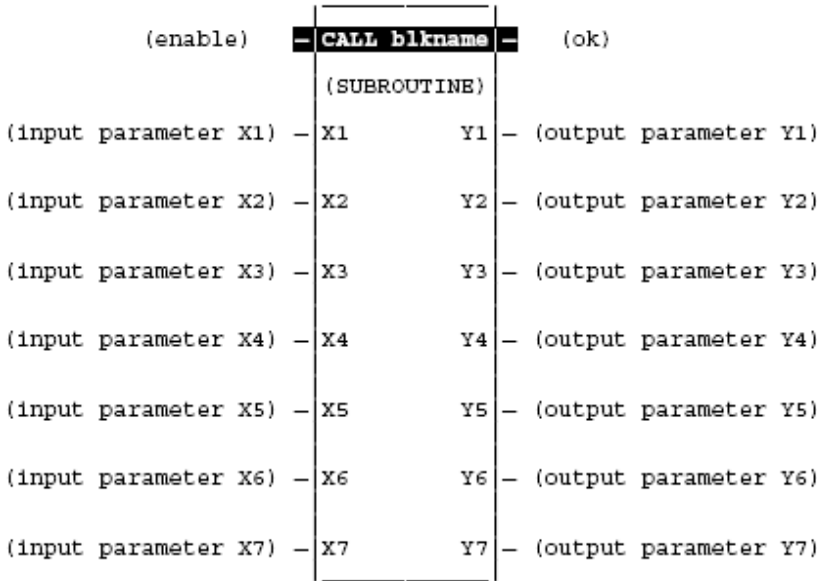
该功能块接通之后，便执行指定的参数子程序。子程序中的逻辑将对该功能块输入数据操作并产生输出数据块。参数OK也由子程序中的逻辑来控制。执行完毕后，控制返回到逻辑中紧跟CALL指令的指令处。

CALL SUBROUTINE 功能无输入/输出参数时的形式如下图：



在这种零参数PSB调用时，OK始终为ON。

CALL SUBROUTINE功能块可以有7对参数，如下图：



参数表

	参数	说明
	使能 (enable)	使能为ON后，开始执行此功能
	X1-X7	声明参数子程序块要使用的输入数据的起始地址
	OK	由参数子程序块内部逻辑通过PSB形式参数YO[001]控制
	Y1-Y7	子程序块输出数据的起始地址

有效的存储器类型

Parameter	flow	%I	%Q	%M	%T	%S			%R	%P	%L	%AI	%AQ	%UR	const	none

- 可使电流流经功能块的变量或地址
- +仅对位参数（当位长度位1时）
- ++当字长度为1时

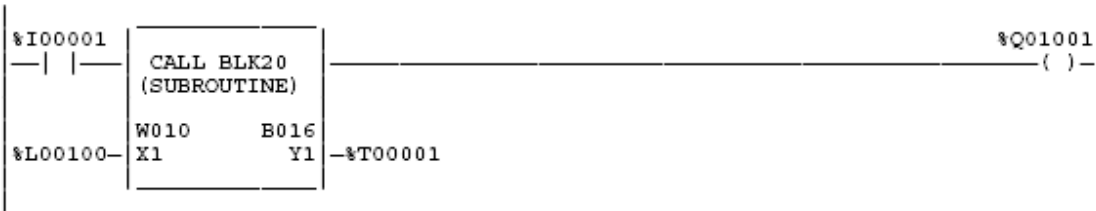
说明：
PSB(参数子程序块)形式**BIT**(位)参数不允许被当作**CALL SUBROUTINE**功能块的参数。而**PSB**形式字或**NWORD**参数可以作为**CALL SUBROUTINE**的参数。对于参数子程序块的限制，请看第二章第三节中有关内容。

说明：
 输入参数可以使用间接变量，但不作为形式参数。在一些应用程序中，可以要求**%S**变量(如**ALW-ON**和**ALW-OFF**)来代替所要求的但未使用的已分配的输入参数。**%S**变量可以不分配给字类型的输入参数，而分配给位类型输入参数。

12

例：

本例中，当**%I00001**置位后，将执行名为**BLK20**的参数子程序块。块**BLK20**对位于变量地址**%L00100~%L00109**中的输入数据操作并产生输出数据块**%T00001~%T00016**。块**BLK20**中的逻辑控制输出**Y1**。



GFK-0265J

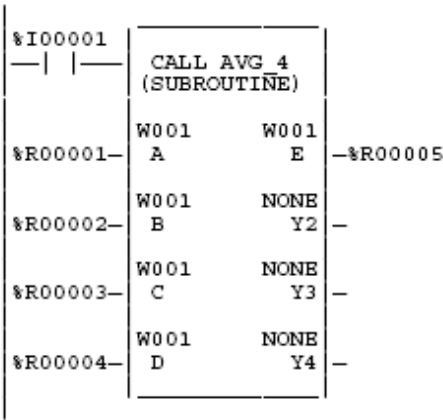
参数子程序块的例子

参数子程序块可用于建立用户自定义功能的库，这些功能块可以被调用并且可以从一个应用程序移植到另一个程序中。例如：如果一个数学公式或算法将被多次调用，那么你可以为此功能建立一个参数子程序块，然后再程序中需要时即可调用这个参数程序块。

例如，有下面一个数学公式：

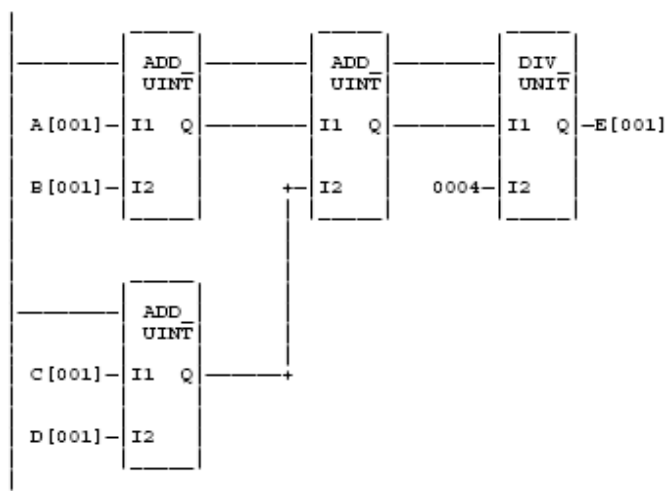
$$E = (A + B + C + D) / 4$$

其中，E为字输出，A，B，C，D为字输入。定义一个参数子程序块名为AVG_4，按下图调用：



本例中，%R00001，%R00002，%R00003，%R00004中数值的平均值将被放入%R00005。

该参数子程序块内部逻辑可以编辑如下图所示：



DOIO

The DO I/O (DOIO)功能用于在程序运行过程中对输入或输出进行更新一次扫描。该功能块可以与SUSIO功能一起使用，SUSIO用于停止正常I/O扫描。DODI功能还可以用于对正常I/O扫描工作之外所选定的I/O点进行更新。如果指定了输入变量，那么该功能可以使程序逻辑取得输入点的最新值。如果指定了输出变量，那么DO I/O功能则根据存储在I/O内存中最新值去更新输出值。在增加的全部I/O模块中使用I/O点，如果有必要的话，PLC将在此功能执行时调整变量。DO I/O对未配置的I/O模块不进行扫描。

说明

DOIO功能仅支持系列90-70 I/O模块，而不支持Genius I/O模块，也不会影响EGD的发送和接受。更多详情参考第四章关于90系列PLC TCP/IP以太网通讯的用户手册，GFK-1541。

DOIO功能块有4个输入参数和一个输出参数。当它得电且确定输入变量时，将对从起始地址ST到结束地址END间的I/O点进行扫描。如果给ALT也指定了变量，那么新输入值的拷贝将被放入到从此变量开始的内存中，而实际输入点并不被更新。ALT的长度必须与所扫描的变量类型的长度相同。如果ST和END均为离散变量，那么ALT也必须是离散变量。如果没有给ALT指定变量，则实际输入点将被更新。

当DOIO功能得电且指定了输出变量时，这些从变量ST开始到END结束的输出点被写入到输出模板。如果要从除%Q，%AQ以外的内部存储器向输出模板写入输出量，那么就可为ALT指定起始变量。向输出模块写入的范围由起始ST与结束地址END确定。

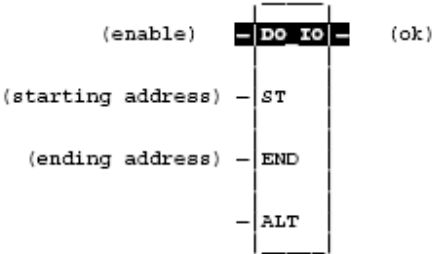
该功能的执行一直持续到报告了所选定范围的所有输入量或在I/O卡上已经使用了所有输出。然后返回到紧随DO I/O的下一条指令。

当此功能块得电时，便向右传递电流，除非出现如下情况之一：

- 所指定类型的所有变量在选定范围内都不存在。
- CPU不能正常地处理由该功能所建立的临时I/O表。
- 所指定范围内的I/O模块出现“Loss of I/O Module (I/O模块丢失)”故障。

说明

如果DOIO功能与时间或I/O中断一起使用，那么与已扫描输入点有关的跳变继电器的操作将不能如预期的那样进行。



参数表

参数	说明
使能 (enable)	使能为 ON 后，执行指定的输入或输出点扫描
ST	待扫描的输入或输出点(字)集合的起始地址
END	待扫描的输入或输出点(字)集合的结束地址
ALT	对输入扫描，ALT 给出存储输入点/字扫描值的内存地址 对输出扫描，ALT 给出向 I/O 点发送数据的内存地址
OK	对输入或输出的扫描正常结束后，OK 为 ON

有效的存储器类型

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
enable	.															
ST		.	.									.	.	.		
END		.	.									.	.	.		
ALT		.	.	.	.		.		.			.	.	.		.
ok	.															.

说明：间接变量对所有寄存器变量均有效

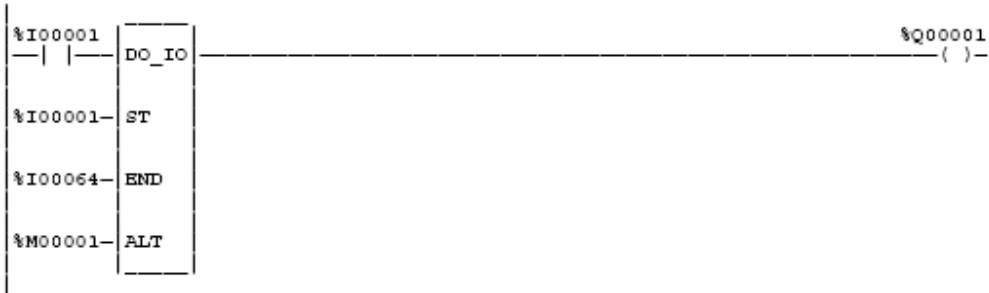
- 可使电流流经功能块的变量或地址
 - 仅为 INT,UINT 或 WORD 有效，但对 DINT 或 DWORD 无效的变量。

说明

对于参数子程序块的限制，请看第二章第三节相关内容。

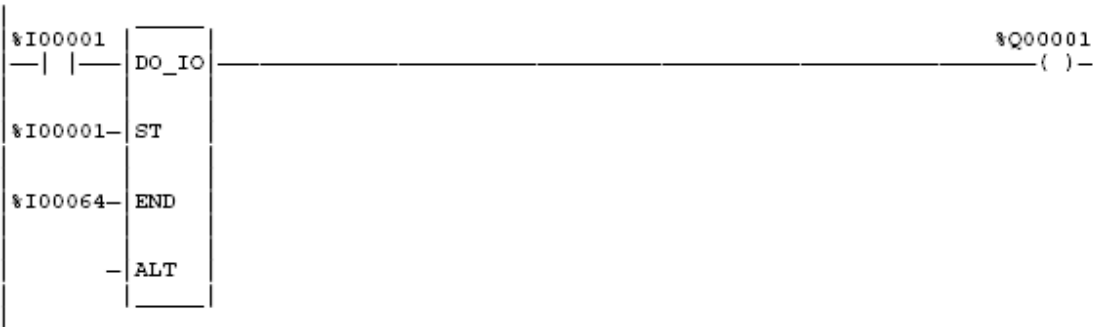
输入示例 1：

在本例中，当允许输入%I00001为ON时，将对%I0001~%I00064进行扫描同时接通%Q00001。扫描得到的输入值将被放入%M00001~%M00064中。实际输入点并不更新。因此，这可以用于比较当输入点的值与扫描开始时的输入点的值。I



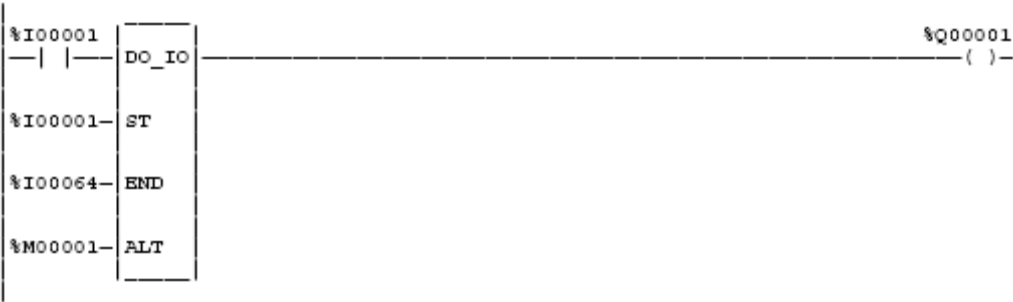
示例2:

本例中,当%I00001为ON时，将对%I00001到%I00064进行扫描同时接通%Q00001。扫描得到的输入点值将放入到从%I00001到%I00064的输入状态存储器中。因此，这种形式的功能允许输入点在CPU扫描的程序执行期间被扫描一次或多次。.



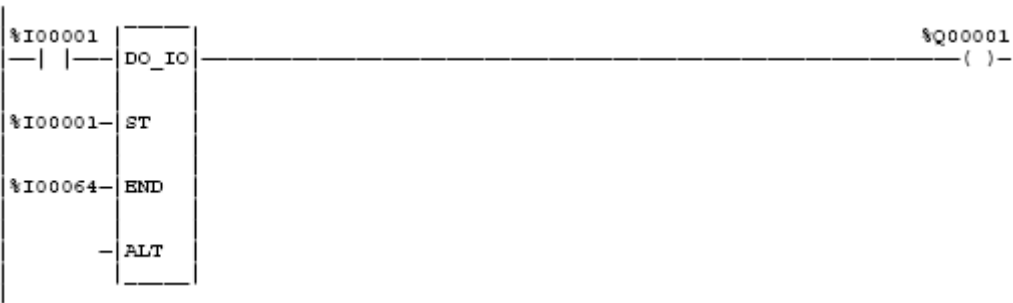
输出示例1:

本例中，当允许输入%I00001为ON时，位于%R00001到%R00004变量中的值将被写入到模拟输出通道%AQ0001到%AQ0004中，同时使%Q00001接通。在%AQ0001到%AQ0004中的值不被写到模拟量输出模块中。



示例2:

本例中，当%I00001为ON时位于%AQ0001到%AQ0004变量中的值将被写入模拟量输出通道%AQ0001到%AQ0004中且使%Q00001接通。

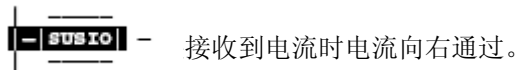


中止I/O(SUSIO)功能用于使一次CPU扫描停止正常的I/O扫描工作。在下一次输出扫描过程中，所有的输出均保持在当前状态。在下一次输入扫描过程中，并不使用输入点的数据对输入变量进行更新。但是，在扫描的输入点扫描期间，CPU将检验Genius总线控制器已完成对它们的先前输出进行更新。

说明

SUSIO将中止所有的I/O包括模拟量和离散量，不论是集成I/O还是Genius I/O或EGD数据。详情参考第四章90系列PLC TCP/IP以太网通讯用户手册，GFK-1541。

SUSIO功能块有一个输入和一个输出。当它得电时，便停止使用所有的I/O除了由DOIO功能所指定的I/O。如果该功能被放在梯形图的左侧，如果没有允许逻辑去控制它的执行，则不会运行常规的I/O扫描。

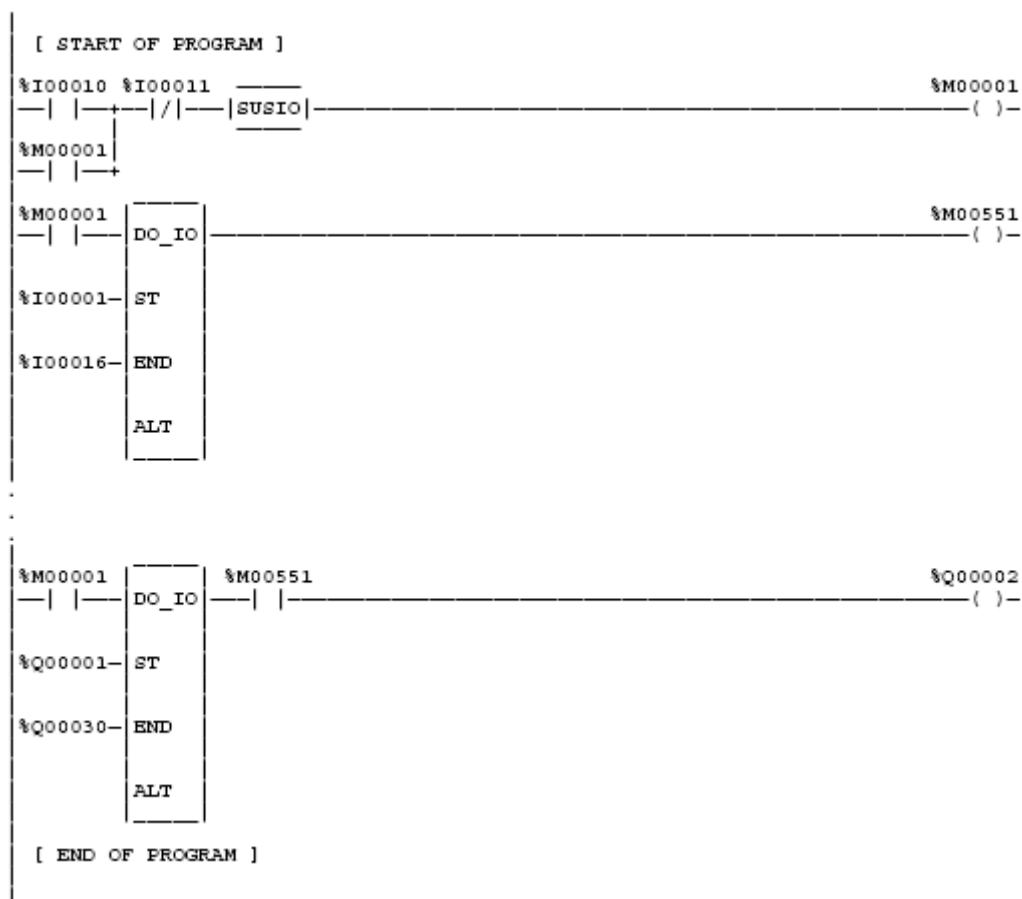


示例：

本例介绍了使用SUSIO和DOIO功能停止常规I/O扫描，但对指定的I/O进行扫描。输入%I00010和%I00011与%M00001触点形成一个自锁电路。于是在每一个扫描周期中保持SUSIO功能处于激活状态直到%I00011动作为止。如果该输入在SUSIO已激活后未被DOIO扫描，那么就只能通过使PLC断电才能禁止SUSIO。

两个DOIO均正常执行完毕后输出%Q00002变为ON。如此构造这一梯级是为了实现即使有一个DOIO块未能将其OK输出置位也可以执行两个DOIO操作。对于正常中止的I/O输出%Q00002要一直到DOIO和%Q00002在其梯级运行后才进行更新。这一切要等到%Q00002置位后的扫描时才发生。在执行某个DOIO功能后被置位的输出要等到另一个DOIO功能执行时才更新，一般是在下一次扫描时进行。由于这一延时，所以在大多数用到SUSIO和DOIO的程序均把SUSIO功能放在程序的第一级，处理输入的DOIO功能放在第二级，而把处理输出的DOIO功能放在最后一级。

处理输出的DOIO功能的范围为%Q00001~%Q00030。如果处于该范围内的模块是一个32点模块，那么实际上将会扫描该整块模块。一个DOIO功能不会在I/O模块扫描期间中断。



MCR

处于激活的主控制器(MCR)与相应的结束主控制器(ENDMCR)间的所有梯级按负逻辑执行。使用MCR功能可以绕过部分程序逻辑。一个与MCR相关的ENDMCR功能用于恢复程序的正常执行。与JUMP指令不同的是，MCR 只能出现在程序的向前方向上。ENDMCR指令必须出现在程序忠与其相应的MCR指令之后。

说明

在一个有效MCR范围内不能进行程序块调用。但是，在程序块中的任何计时器将会继续累计时间。

对于每一个ENDMCR指令只能对应一条MCR指令。一个MCR/ENDMCR指令对可以嵌套在其它MCR/ENDMCR对中。MCR功能有一个使能布尔量输入(EN)而且可以给它指定一个名字。这个名字也必须与和它对应的ENDMCR名字相同。MCR功能没有输出，因此在一级中MCR后就不再有其它指令。

-[MCR]-

示例:

本例中，当%I00002允许电流进入MCR功能后，程序将继续执行下去即使无电流进入线圈，直到达到相应的ENDMCR为止。如果%I00001和%I00003为ON，则%Q00001将断开，而%Q00003保持ON。



ENDMCR

使用结束主控继电器(ENDMCR)指令用于在一个MCR功能后重新开始正常程序的执行。当与ENDMCR有关的MCR激活时，ENDMCR使程序执行恢复正常。当与ENDMCR功能相关的MCR未被激活时，该ENDMCR将不起作用。

ENDMCR有一个负逻辑布尔型使能输入(EN).该指令的使能端必须放在电源侧，它的执行是无条件的。ENDMCR功能也有一个名字，该名字用于识别ENDMCR并与其相应的MCR联系起来。ENDMCR功能没有输出，在一级中ENDMCR指令后不能有任何其它指令。

[END MCR] -

→ EN

```

| CLEAR
- [ END MCR ] MCR指令用于结束名字为 CLEAR 的

```

```

| CLEAR
- [ END MCR ]

```

JUMP

使用JUMP功能可以跳过一部分程序逻辑。JUMP既可向前跳转也可以向后跳转。程序将转向LABEL所声明处继续执行。

JUMP指令与MCR指令相似，不同的是在JUMP范围内的线圈不按负逻辑执行。当JUMP激活后，在它的范围内的所有线圈均被激活。这些包括与计时器，计数器，自锁接点和继电器相关的线圈。

JUMP指令一般都被放在当前梯级的第九和第十列；因此在该级的JUMP指令后不能再出现其它指令。电流将直接从此指令跳转到LABEL所标明的梯级处。

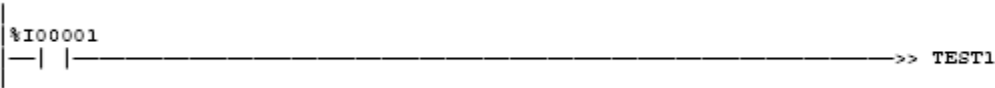


注意

避免因使用向前或向后JUMP指令而导致无限循环，一个向后跳转JUMP必须是有条件的。

示例:

本例中，只要JUMPTEST1被激活，那么电流将被传到LABEL TEST1处。



LABEL

LABEL指令起到了指明JUMP的目的地的作用。使用LABEL指令来重新开始一个JUMP指令以后的程序正常运行。

一个程序中只能有一个专用名字的LABEL。如果没有一个匹配的JUMP/LABEL对的程序，虽然可以建立并存储到PLC中，但是并不被执行。

LABEL指令无输入参数和输出参数；因此在LABEL指令前后均不能出现其它指令。

示例:

本例中，从JUMP TEST1来的电流将从LABEL TEST1处重新流通。

| **TEST1** :

COMMENT

使用COMMENT功能可以给程序添加注释。一个注释区最多允许有2048个文本字符。在逻辑图中该功能块如下图所示：



将光标移到(*COMMENT*)处按F10(缩放键)，即可读取注释内容并进行编辑。注释内容可以被打印出来。

通过使用注释文件可以打印出更长内容的主食内容。实现方法如下：

1. 建立注释：
 - A. 在要读入其它文件的点处输入注释
 - B. 将光标移到一新行的起始处并键入\I 或 \i，接着输入文件所在的路径和文件名，如下例所示：

```
\I d:\text\commnt1
```

如果该文件与程序在同一路径中时，可以省略路径。

- C. 继续编程，或返回MS-DOS。
2. 在退出编辑器后，用任何MS-DOS兼容软件生成一个文本文件。给此文件一个文件名(已录入在注释中)，然后把其放入在注释制定的路径中。

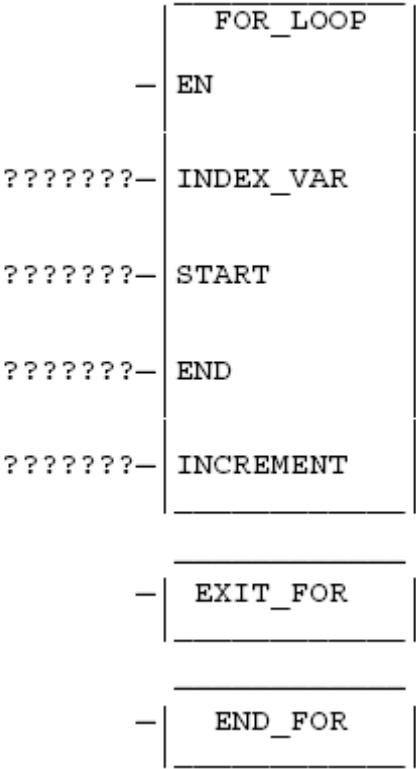
FOR、END-FOR 和 EXIT 指令可以重复某一级逻辑若干次，同时不断改变循环变量 FOR INDEX-VAR，被重复的逻辑必须介于 FOR 与 END-FOR 指令之间。

FOR 指令有 5 个输入参数但没有输出参数。当 FOR 指令的输入端 EN 接通时，参数 START、END 和 INCRECEMENT 的值将被保存。这些保存的值用于确定在 FOR 和 END-FOR 指令之间梯级被执行的次数。在 FOR 循环执行过程中改变 START 和 END 参数并不影响循环的次数。

当有电流流入 EXIT 指令时，FOR 循环即被终止而直接转去执行紧随 END-FOR 指令后的语句。

在 FOR 指令后可以没有其它指令。一个 EXIT 语句必须处于 FOR 与 END-FOR 指令之间且可在 EXIT 后也没有其它指令。

END-FOR 语句在此梯级中必须是唯一的指令。



一个 FOR-LOOP 能够给其循环变量赋予一个不断减少的值，这只要设定一个增量即可。如果 START 为 21，END 值为 1，

增量为-5 时，FOR 循环将执行 5 次，而每次循环变量递减 5。该循环变量值将依次是：21，16，11，6，1。

当 START 与 END 的值设置相同时，FOR 循环语句只执行一次。

当无论从 START 开始增加还是从 START 开始递减都无法达到 END 时，在 FOR 循环中的语句将不被执行。例如，START 为 10，END 为 5，增量为 1 时，将会直接去执行 END-FOR 语句之后的语句。

说明

当在对 FOR 循环首次测试时，如果 EN 参数得电，那么在 FOR 和其对应的 END-FOR 语句之间的梯级将按 START、END 和 INCRECEMENT 中的初始值执行规定的次数。这样的重复执行将发生在 PLC 的一次单一扫描过程中，同时这个循环如果超长，那么将引起监视定时器产生终止信号。

FOR 循环允许嵌套，最多可以有 5 层嵌套。每个 FOR 指令必须有与其相应的 END-FOR 语句相匹配。

在 FOR 循环中允许使用 JUMP 和 MCR 指令嵌套，每一对 MCR 和 ENDMCR 或 JUMP 和 LABEL 指令必须全部都在一对 FOR/END-FOR 指令之间或者全部都在 FOR/END-FOR 指令之外。不允许跳入或跳出某一 FOR/END-FOR 循环。

参数表

参数	说明
使能(enable)	使能为 ON 后，开始执行循环。
INDEX-VAR	含循环变量。循环执行完毕后，该值不确定。最好不要

GFK-0265J

	在循环中改变该值。
START	循环变量的初值。
END	循环变量的终值。
INCREMENT	循环增量（默认为 1）

有效的存储器类型

Hf al ro aw n e t e r															
E n a b l e															
I N I E X															
S T A F T															
E N I															
I															

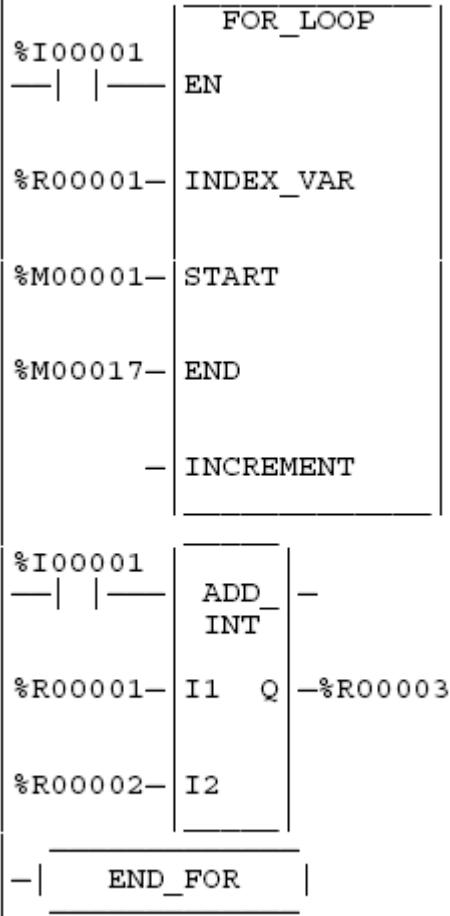
	N C E E M E N T															

。可使电流流经功能块的变量或地址。

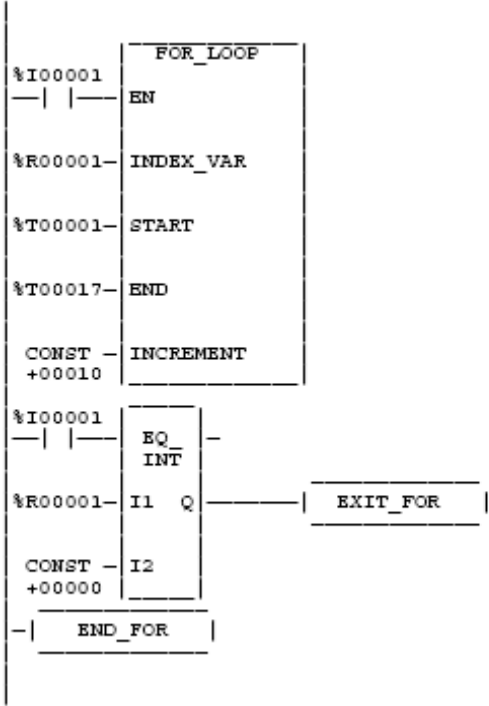
说明

对于参数化子程序块的改制，见 2-24 页。

例 1:
本例中，%M00001（START）的值为 1，%M00017（END）的值为 10。循环变量 INDEX-VAR（%R00001）由 INCREMENT（缺省时为 1）在 1 开始递增到 10。循环体中的 ADD 指令将被执行 10 遍，将 I1（%R00001）中的值 1，2，。。。。，10 加到 I2（%R00003）中。



本例中，%T00001(START)为-100，%T00017(END)为 100。循环变量 INDEX-VAR (%R00001) 增量为 10，从-100 开始直至 100。循环体的 EQ 指令将执行 21 遍，I1 (%R00001) 中值将依次为-100， -90， -80， -70， -60， -50， -40， -30， -20， -10， 0， 10， 20， 30， 40， 50， 60， 70， 80， 90， 100。当 I1 (%R00001) 中的值变为 0 时，EXIT 指令被激活并直接转到 END-FOR 语句之后的语句去执行。



SVCREQ

使用服务请求（SVCREQ）功能可以申请下列 PLC 服务功能：

表 12-1 服务请求功能

功能	说明	页码
1	改变/ 读取 恒定 扫描 定时 器	12-28
2	读窗 口值	12-31
3	改变 编程 器通 讯窗 口状 态和 值	12-32
4	改变 系统 通讯 窗口 状态 和值	12-33
5	保留	12-34
6	改变/ 读取 校验 和任 务状 态和 字数	12-36

7	改变/ 读取 日历 时钟 状态 和值	12-38
8	复位 监控 定时 器	12-43
9	读取 扫描 开始 后的 扫描 时间	12-43
10	读取 本模 块所 在程 序名	12-44
11	读取 PLC ID	12-45
12	读取 PLC 运行 状态	12-46
13	关断 PLC	12-47
14	清除 故障 表	12-48
15	读出 故障 表中 最后 登录 的信 息	12-49

16	读取 耗时 时钟	12-53
17	屏蔽/ 去除 屏蔽 I/O 中 断	12-54
18	读取 I/O	12-55
19	设置 运行 允许/ 禁止	12-57
20	读取 故障 表	12-58
21	登录 用户 定义 的 PLC 故障 表	12-62
22	屏蔽/ 去除 屏蔽 时钟 中断	12-64
23	读取 主校 验和	12-65
24	保留	---
25	允许/ 禁止 EXE 块的 验和	12-67
26	转换 任务	12-68

GFK-0265J

	(活动到备用和备用到活动)	
27	写入交换传输区	12-69
28	读取交换传输区	12-69
29-31	保留	---
32	追加/恢复 I/O 中断	12-70
39	ESCM 端口状态	12-72

12

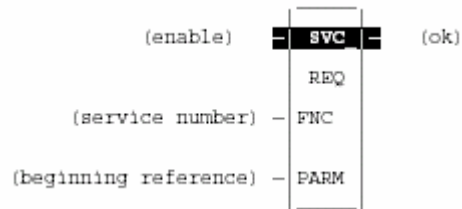
44	逻辑驱动型动态 EGD 通讯	12-74

SVCREQ 功能块有三个输入参数和一个输出参数。当它得电时，便申请 PLC 执行所指定的功能（FNC）。功能块所需的参数从 PARM 指定的变量处开始。SVCREQ 功能块传递电流除

非出现错误的参数，非法的功能号或超出范围的变量地址。其它的故障因素将在后面加以介绍。

为 PARM 指定的变量可以是任何一种类型的存储器（%R，%AI，%AQ，%P，或%L）。该变量将是该功能块组“参数块”的第一组数据。接下来的 16 位存储单元将存储附加参数。所需变量的总数将由所用 SVCREQ 功能块的类型来决定。

参数块既可作为此功能块的输入，也可作为在此功能块执行后输出数据的存储单元。因此，由此功能块返回的数据可以由 PARM 所确定的同一地址处存取。



参数表

参数	说明
使能 enable	当使能被接通后，开始执行该请求服务
FNC	含有所申请服务的序号或变量地址
PARM	含有所申请服务的参数块的起始地址
OK	功能块正确执行，OK即为 ON

可用的存储器类型

存储器类型															

	E n a b l e															
	F N C															
	F A F N															
	C F															

说明：间接变量对所有的寄存器变量（%R，%P，%L，%AI，%AQ

和%UR）均有效。
。可使电流通过功能块的变量或地址。
说明
对于参数子程序块的限制，请看第二章第三节中有关内容。

示例：

本例中，当使能输入%I00001 为 ON 后，1 号 SVCREQ 功能被调用，其参数子程序块的地址起始于%L00001。如果成功执行完毕，输出线圈 %M00001 将被置为 ON。



SVCREQ #1: 改变/读取恒定扫描计时器

使用 SVCSEQ #1 功能可以:

- 禁止恒定扫描时间模式
- 允许恒定扫描时间模式并使用旧计时器的值
- 允许恒定扫描时间模式并使用新的计时器的值
- 仅仅设定一新的时间值
- 读取恒定扫描模式状态和计时器值

参数块的长度为 2 个字。
为禁止恒定扫描时间模式，可以输入有以下参数块的 SVCREQ#1 功能：

0	地址
忽略	地址+1

当允许恒定扫描时间模式，可以输入有以下参数块的 SVCREQ#1 功能：

0	地址
0 或 计时 器的 值	地址+1

说明

如果想让计时器选用新的时间值，在第 2 个字中输入这个值。
如果不改变计时器的值，在第 2 个字中输入 0。如果计时器值并不存在，那么写入 0 会使该功能块的输出参数，以将 OK 置为 OFF。

如果只想改变计时器的值而不改变对扫描模式状态的选择，那么输入 SVCREQ#1 功能有以下参数块：

2	地址
新的 时间 值	地址+1

如果要读取计时器的值和状态而不改变其它值，那么参数块应为：

3	地址
忽略	地址+1

说明

使用上页所示参数块的 SVCREQ #1 功能之后，6 或者更高版本的 CPU 返回值表示正常扫描，1 表示固定扫描，2 表示循环扫描。不要与下面所示的输入值相混淆。

本功能将成功执行，除非出现下列情况：

1. 输入了 0、1、2、3 以外的数作为所请求的操作：

0	禁止恒定扫描时间模式
1	允许恒定扫描时间模式
2	设定新时间值
3	读取恒定扫描模式的状态和时间值

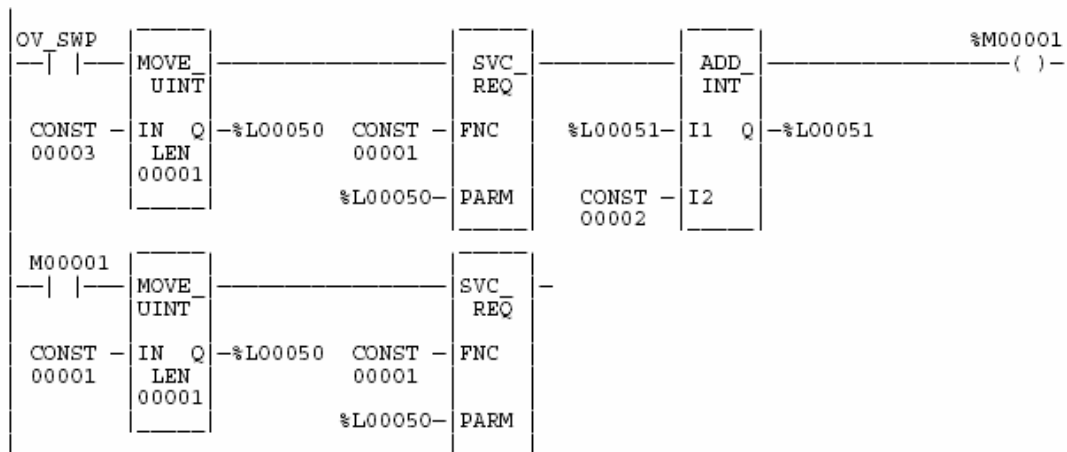
2. 时间值大于 2500ms(2.5 秒)
3. 采用恒定扫描模式但计时器值并未编制，或者此计时器值为 0
该功能执行后，将把计时器状态和值返回到相同的参数块变量。

0=禁止	地址 地址+1
1=允许	
当前时间值	

如果字地址+1 处的内容为 16 进制的 FFFF，那么表明计时器值还未进行编程。

例：

当使能触点 OV_SWP 置位后，将读取恒定扫描计时器的值，并将该计时器再加 2 毫秒，再把此新计时器值送回 PLC。参数块位于本地内存 %L00050。由于 MOVE 和 ADD 功能块各要占据三个触点的位置，所以本例采用一离散内部线圈 %M00001 作为保存第一梯级成功执行结果的临时地址，在任何一次扫描时，如果 OV_SWP 来被置位，那么 %M00001 将断开。



SVCREQ #2: 读取窗口值

使用 SVCREQ #2 功能可以获得编程器通讯窗、系统通讯窗口和后台 窗口的当前模式时间值。每个窗口均有三种模式：

模式名称	值	说明
限制模式	0	窗口的执行时间的时限为其默认值或为通过 SVCREQ #3 功能为编程器通

		讯窗口或使用 S V C R E Q #4 功能为系统通讯窗口设定的值。如果不再有要完成的任务，该

		窗口将终止。
恒定模式	1	每个窗口将运行在“运行到完成”模式中，P L C 轮流进入三个窗口，总时间

		等于各窗口之间之和。如果一个窗口处于恒定模式，那么其余的两个也自动地变为恒定模式

		。如果 P L C 运行在恒定模式下而某一窗口的运行时间未通过相关 S V C R E Q 功能被确定

		，那么在此恒定窗口时间计算中使用此窗口的默认时间。
运行到完成模式	2	不管窗口时间是默认时间值或是通

		过 S V C R E Q 功 能 所 确 定 的 值 ， 窗 口 都 要 运 行 到 它 所 包 含 的 所 有 任 务 都 被 执 行 。

当模式为限制模式（0）或时间值为 0 时，窗口将被禁止。

参数块的长度为三个字：

GFK-0265J

高字节

低字节

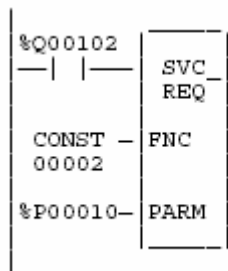
编
程
器
窗
口
系
统
通
讯
窗
口
保
留

模 式	模 式 (m s)
模 式	数 值 (m s)

该功能只有输出参数，无须为它的参数块输入数值，三个窗口的输出值均以毫秒为单位。

示例：

当使能输入%Q00102 置位后，PLC 操作系统将三个窗口的当前时间值放入从%P00010 开始的参数块中。在接下来的三个 SVCREQ 功能介绍中将包括读取窗口值的离子。



SVCREQ #3：改变编程器通讯窗口的状态和值

使用 SVCREQ #3 功能可以允许/禁止编程器通讯窗口。这种变化将发生在该功能被调用的同一次 CPU 扫描过程中。

当窗口被允许后，该功能将返回窗口的当前时间值。当窗口被禁止后，它将一直保持禁止状态知道再次被允许后为止。对于 3 号功能，SVCREQ 将始终向右传递电流。

参数块长度为 1 个字节。

为禁止编程器的窗口，应输入有下述参数块的 SVCREQ #3 功能：

	高字节	低字节
地址	0	0

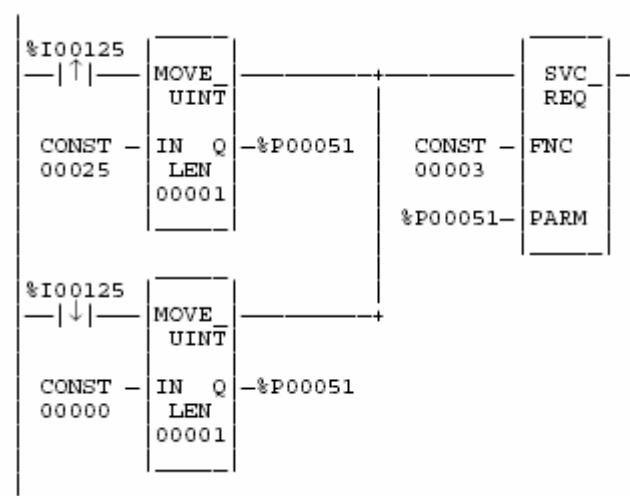
为允许编程器窗口，应输入以下参数块：

模 式	数 值 (1 ~ 24 4 m s)

地址

示例：

本例中，当使能输入%I00125 发生正向跳变时，编程器窗口将被允许并赋予 25ms 时间值。当使能输入断开时，窗口将被禁止。参数块位于全局内存%P00051 中。



SVCREQ # 4 : 改变系统通讯窗口的状态和值

GFK-0265J

使用 SVCREQ # 4 功能可以允许/禁止系统通讯窗口。这种变化将发生在该功能被调用的同一次扫描过程中。当窗口被允许后，该功能将返回窗口当前的时间值。当窗口被禁止后，它将一直保持禁止状态知道再一次被允许为止。SVCREQ # 4 功能将始终向右传递电流。

参数块的长度为 1 个字。

为禁止编程器的窗口，应输入有下述参数块的 SVCREQ # 4 功能：

	高字节	低字节
地址	0	0

为允许编程器窗口，应输入以下参数块：

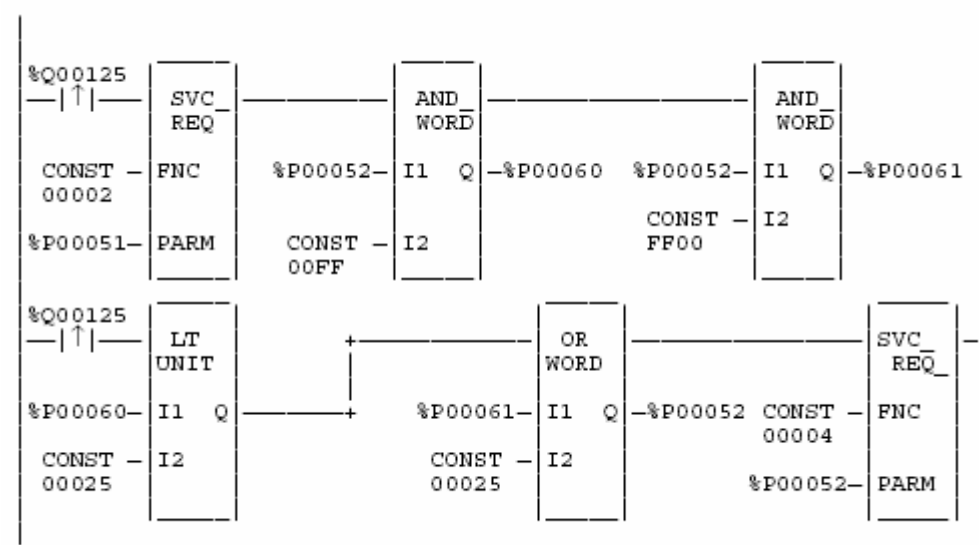
	高字节	低字节
地址	模式	数值 (1 ~ 255 ms)

示例：
本例中，当使能输出%Q00125 发生正跳变时，将读取系统通讯窗口的值。如果该值大于或等于 25ms，那么该值将不变

化。如果小于 25ms，那么该值将被改为 25ms。无论那种情况，

当逻辑执行完毕后窗口均被使能。三个窗口的参数数据块位于 %P00051。由于系统通讯窗口的时间值是处在从 SVCREQ # 2

功能返回的数据块中的第二个值中，所以系统通讯窗口的新的窗口时间位于%P00052 中。



SVCREQ # 5 ： 改变后台任务窗口模式和定时器值

使用 SVCREQ # 5 改变后台任务窗口模式和定时器值。这种变化在该功能被调的同一个 CPU 扫描周期内发生。

使用 SVCREQ # 4 功能块来使能或者禁止系统通讯窗口。这种变化在该功能被调的同一个 CPU 扫描周期内发生。

SVCREQ # 4 功能向右导通电流，除非发生以下情况：

- 1. 选择模式 0（限制），1（固定）或 2（运行到完成）以外的模式。
- 2. PLC 运行在循环扫描模式下，而没有选择模式 1（固定扫描）。

SVCREQ # 5 总是向右导通电流。
参数块的长度为 1。

为禁止编程器的窗口，应输入有下述参数块的 SVCREQ # 5 功能：

高字节 低字节

地址

0	0

为允许编程器窗口，应输入以下参数块：

高字节 低字节

模 式	数 值 (1 ~ 25 5 m s)

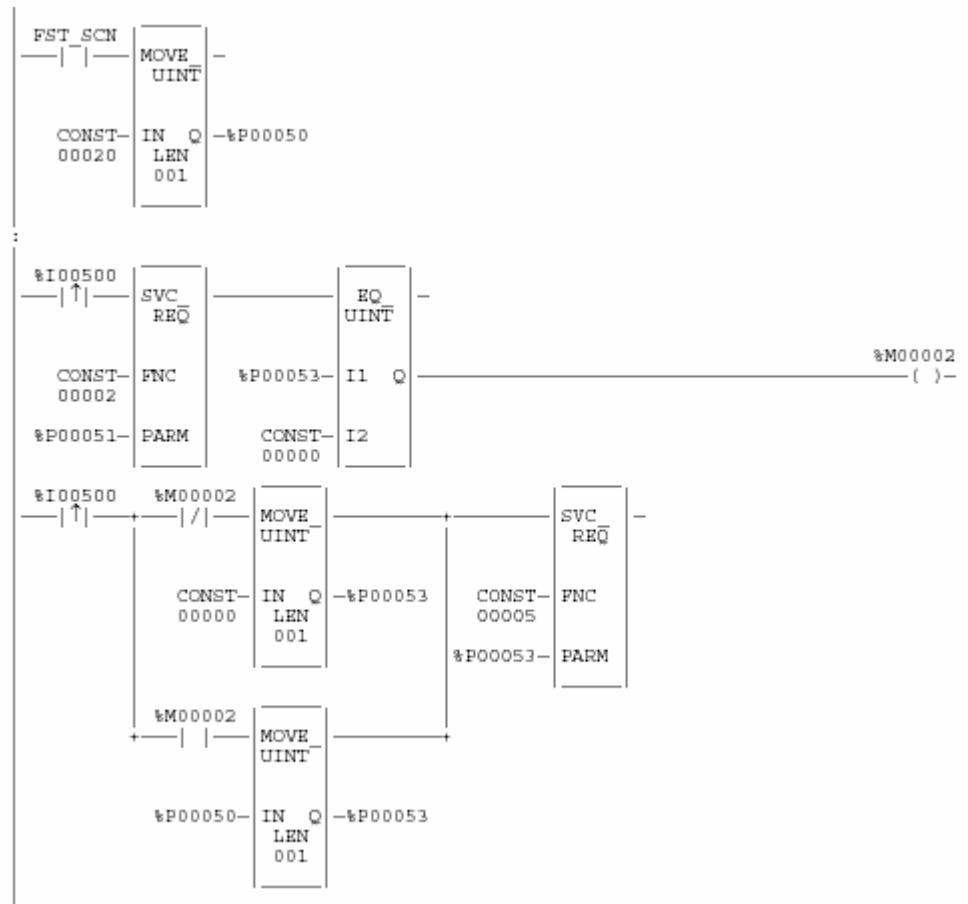
地址

Example:

In the following example, when enabling contact FST_SCN is set in the first scan, the MOVE function establishes a default value of 20 ms for the background task window, using a parameter block beginning at %P00050. Later in the program when input %I00500 transitions on, the state of the background task window toggles on and off. The parameter block for all three windows is at location %P00051. The time for the background task window is the third value in the parameter block returned from the Read Window Values function (function #2); therefore, the location of the existing window time for the system communications window is %P00053.

示例:

本例中，当结点 FST_SCN 在第一次扫描时置位的时候，使用 %P00050 为起始地址的参数块，MOVE 功能块为后台任务窗口设定 20ms 的默认值。之后



SVCREQ # 6：改变/读取校验和任务状态和校验和字数

使用 SVCREQ # 6 号功能可以读取当前的字数或设置一个新的字数。默认时，将有 16 个字作校验和。除非输入 0 和 1 以外的数作为请求的操作，该功能均能成功地完成。

参数块的长度为 2 个字。
为读取当前的字数，参数块应写为：

0	地址
忽略	地址+1

在该功能完成后，该功能把当前的校验和返回到参数块中的第二个字中。对读操作未指定范围；返回的值是当前正被校验和的字数。

0	地址
当前字数	地址+1

为设置一个新字数，输入有以下参

数块的 SVCREQ # 6 功能

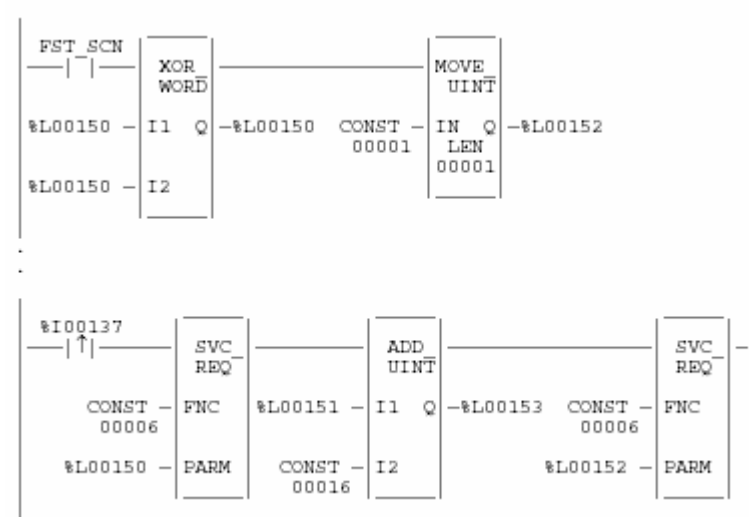
0	地址
新字数	地址+1

输入 1，使 PLC 将所

校验和的字数调整为参数块的第 2 个字中所指定的值，该值将被按 8 的倍数进行舍入。为禁止校验和，可将数值设为 0。

示例：

本例中，当使能出点 FST_SCN 置位后，将建立校验和功能块和参数块。接着当输入 %I00137 发生正跳变后，从 PLC 操作系统中读入待校验和的字数。这个字数在 ADD_UINT 功能块中增加 16，运算结果放在“保持新设定字数”参数中。第二个 SVCREQ # 6 功能将请求 PLC 设定新的字数。



上例中的参数块放在%L00150处。它包括以下内容：

0=	%L00150
读	%L00151
取	
当	%L00152
前	%L00153
值	
保	
持	
当	
前	
值	
1=	
设	
置	
新	
值	
保	
持	
新	
值	

7: 改变/读取日历时钟状态和值

使用该功能可以读取或设置 PLC 中的日历时钟。
参数块的长度由数据格式确定。每个数字分离式 BCD 码需要 9 个字。BCD 格式需要 6 个字。组合 ASCII 码需要 12 个字。

0=读取 时间和 日期	地址
1=设置 时间和 日期	地址+1
0=数值 数据格 式 1=BCD 格式 2=分离 式 BCD 格式 3=组合 式 ASCII 格式	地址+2~结 束
数据	

在第一个字中规定该功能是读还是改变数值。

- 0 = 读取
- 1 = 改变

第二个字中规定数据格式：

- 0 = 数值
- 1 = BCD
- 2 = 分离式 BCD
- 3 = 组合式 ASCII，带空格和冒号

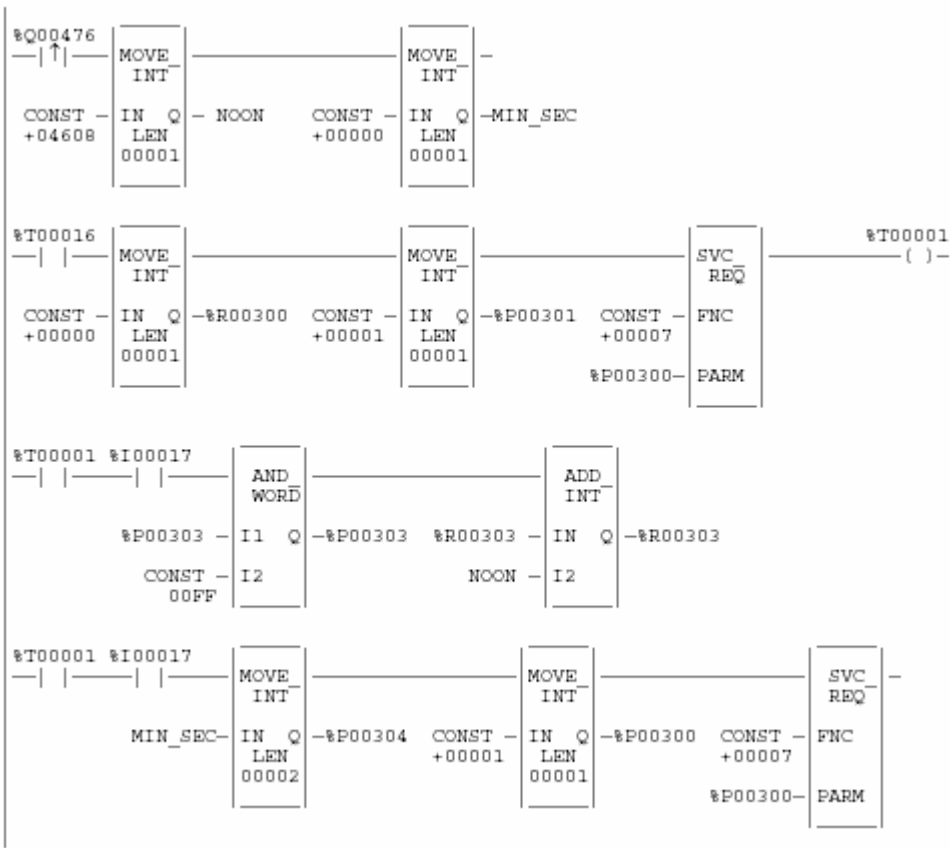
从第三个字到参数块的末尾包括由读功能返回数据，或包括由改变功能所需的新数据。在此两种情况下，这些数据字的格式都是一样的，当读取日期和时间时，连输入时参数块中的字（地址+2 到地址+8）将被忽略。

下列情况导致功能块执行失败：

- 输入了 0 和 1 以外的数值作为请求操作（见后面所述）
- 声明了一个非法的数据格式
- 所提供的数据不是所需的格式

示例：

本例中，当输出%Q00476 为 ON 后，将建立第一个读取日期和时间的功能块的参数块，然后使用 BCD 格式设置为中午 12 点。参数块位于全局数据地址%P00300 处。数组 NOON 在程序中其它地方已设置为 12.0 和 0。（数组 NOON 也必须含有 %R00300 中的数据）BCD 格式的参数块需要 6 个连续的内存地址。



参数块的内容

四种不同数据格式的参数块内容将在下面进行介绍。对所有的数据格式，小时是 24 小时制，周几是用数值表示的。

值	星期
1	星期日
2	星期一
3	星期二
4	星期三

	三
5	星 期 四
6	星 期 五

7	星 期 六

使用数字格式改变/读取日期和时间

在数字格式中，年、月、日、小时、分、秒和周几均各占有一个无符号整数。相应的参数块为：

输出举

例：

（数字格

式）

（1988. 6. 15 星期三， 12: 15: 30）
高字节 低字节

1	地址	0
=	地址+1	0
改	地址+2	88
变	地址+3	06
或	地址+4	15
0	地址+5	12
=	地址+6	15
读	地址+7	30
取	地址+8	04
0		
年		
月		
日		
小		
时		
分		
秒		
周		
几		

使用 BCD 格式改变/读取日期和时间

在 BCD 格式中每个时间和日期项各占一个字节。这种格式共需 6 个字。最后一个字节不用。当设置日期和时间时，该字节被忽略；当读取日期和时间时，该字节将返回一个空字符（00）。

输出举例：
(数字格式)
(1988. 7. 3

星期一， 2: 45: 30PM)
高字节 低字节

地址

地址+1

地址+2

地址+3

地址+4

0	
1	
07	88
14	03
30	45
00	02

1 = 改变 或 0 = 读取	
1	
月	年
小时	日
秒	分
(NULL)	周 几

地
址
+
5

使用分离式 BCD 格式改变/读取日期和时间

使用这种格式，每一个时间和日期项各占有一个字节的低 4 位，高 4 位始终为 0。这种格式需要 9 个字节。

输出举
以后需

例：

BCD 格式读取日期和时间

(1989. 12. 28 星期四, 9: 34: 57)

GFK-0265J

高字节	低字节	
1	地址	0
=	地址+1	2
	地址+2	08
改	地址+3	09
变	地址+4	01
	地址+5	02
或	地址+6	02
	地址+7	08
0	地址+8	00
=		09
读		03
取		04
2		05
		07
年		00
		05
月		
日		
小		
时		
分		
秒		
周		
几		

使用组合式 ASCII 码（带空格和冒号）改变/读取日期和时间

在这种格式中，每个日期和时间项都是一个 ASCII 码格式字节。另外，数据中的空格和冒号也被嵌入到数据中并打印或显示出来。该格式需要 12 个字。

输出举例：
(数字格式)
(1988. 7. 3

星期一， 2: 45: 30PM)
高字节 低字节

地址
地址+1
地址+2
地址+3
地址+4
地址+5
地址+6
地址+7
地址+8
地址+9
地址+10
地址+11

0	
3	
39	38
31	20
20	30
32	30
32	20
3A	33
33	31
30	3A
20	30
33	30

1 = 改变 或 0 = 读取	
3	
年	年
月	空 格
、	—

S
V

口	口
小 时	空 格
:	小 时
分	分
秒	:
空 格	秒
周 几	周 几

C
R
E
Q

8

:
复
位
监

控定时器

在扫描过程中使用 SVCREQ # 8 功能可以复位监控定时器。当
监控定时器计满后，PLC 将关断且不发出任何警告，该功能允

许计时器在时间消耗内容中继续计时（例如，当等待通讯应答式）

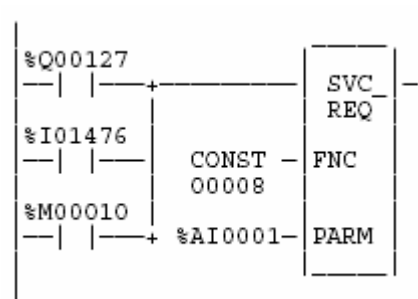
注 意

使用功能时应首先确认这样做不会对所控制的过程产生不利的影响。

该功能块没有相应的参数块；但在编程时要求给 PARM 参数指定一个地址入口，该地址未使用。

例：

本例中，当使能输出%Q00127 或输出%I01476 或内部线圈 %M00010 置位后，将复位监控定时器。



SVCREQ # 9：读取从扫描开始到当前所经历的时间

该功能用于读取从扫描开始到当前所经历的毫秒数。数据格式为 16 位无符号整型。参数块是一个输出参数块，长度为 1 个字。

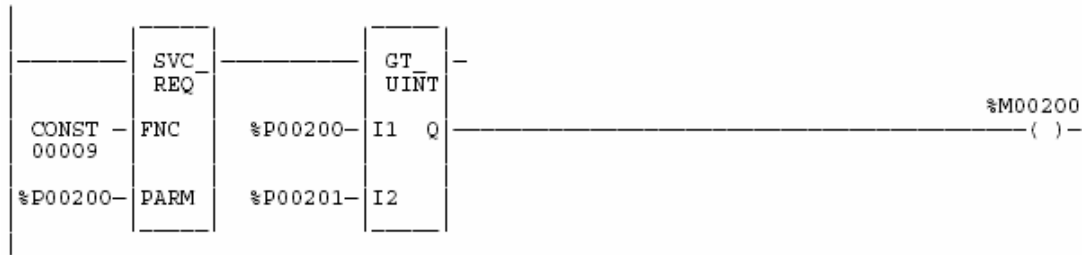
从
扫
描
开
始
的
时
间

地址

例：

本例中，从扫描开始所经历的时间总是被读入到 %P00200 中。

如果该值大于 %P00201 中的值，内部线圈 %M00200 将被接通。



说 明

SVCREQ 在不同的循环扫描模式下有不同的功能。它反应了程序开始运行所经过的时间。

SVCREQ # 10 ： 读取此模块所在程序名

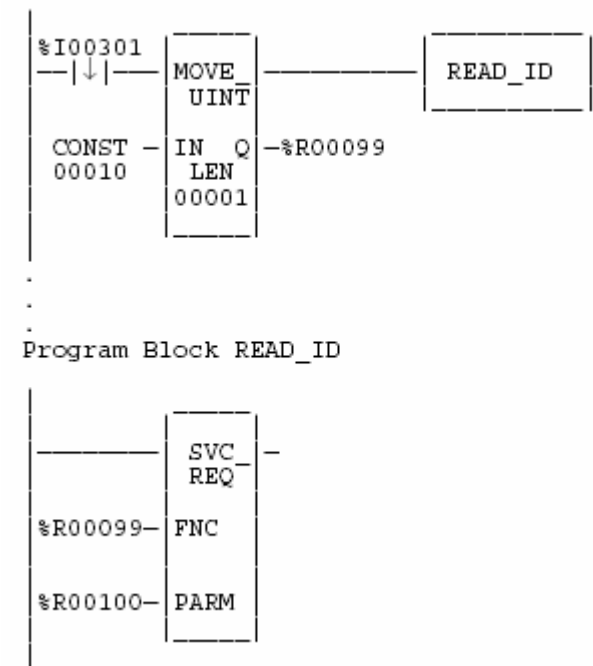
该功能用于读出当前正在执行的程序块的名称。
输出参数块有 4 个字长。它将返回 8 个 ASCII 字符；最后将是一个空字符（00h）。如果程序名小于 7 个字符，那么将在末尾添加空字符。

低字节	高字节
字符 1	字符 2
字符 3	字符 4
字符 5	字符 6
字符 7	00

地址
地址+1
地址+2
地址+3

示例：

本例中，使能输入%I00301 发生负跳变，10 被装入寄存器地址 %R00039 中，它是读程序 ID 功能的功能代码。然后程序块 READ_ID 被调用并取回 ID。参数块处于 %R00100 地址中。READ_ID 也将在下一个例子中被使用。



SVCREQ # 11 : 读 PLC ID

该功能用于读取 90 系列 PLC 正在执行的程序名。

输出参数块长度为 4 个字。它将返回八个 ASCII 字符，最后一个为空格字符 (00h)。如果 PLC ID 少于七个字符，那么将会在其末尾补空格字符。

低字节

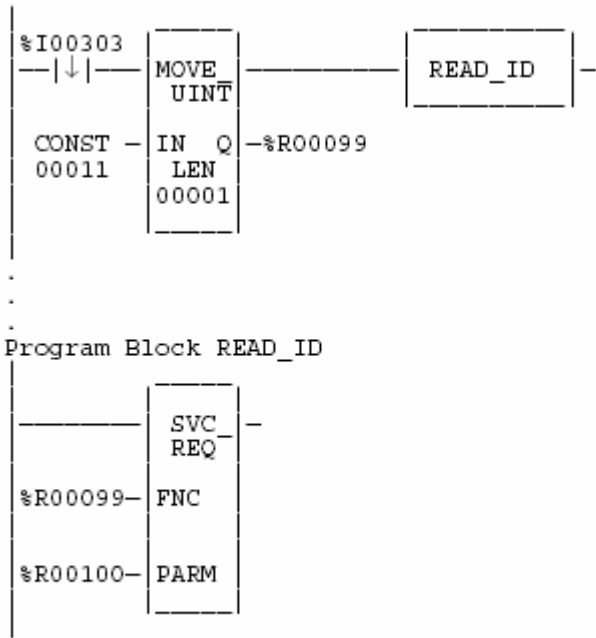
高字节

GFK-0265J

字符 1	字 符 2
字 符 3	字 符 4
字符 5	字 符 6
字符 7	00

地址
地址+1
地址+2
地址+3

例：
本例中，使能输入%I00303 发生负跳变时，寄存器%R00099 被装入 11，该值为读 PLC ID 功能的功能代号。程序块 READ_ID 继而被调用并取回 ID 数据。
参数块位于%R00100。除了使能触点和功能号外，其余均与上例相同。

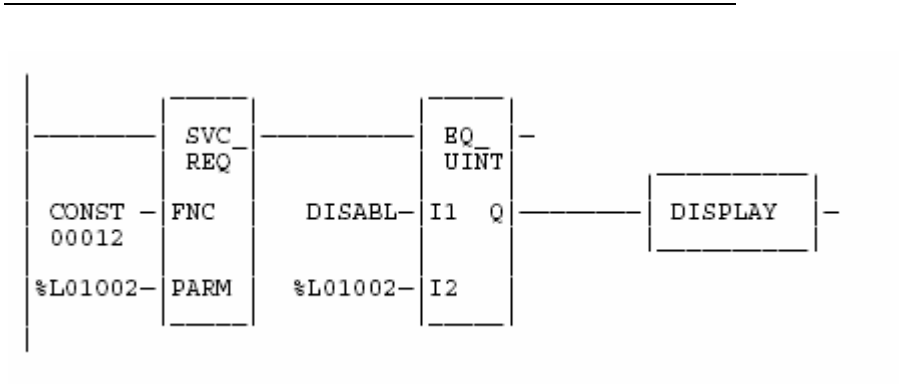


SVCREQ # 12：读取 PLC 的运行状态

该功能用于读出当前 PLC CPU 的运行状态。参数块仅为一个输出参数块，长度为 1 个字。

1	地址
=	
运行	
/	
禁止	
2	
=	
运行	
/	
使能	

例：
本例中，PLC 的运行状态总是被读入到地址%L01002 中。如果状态为运行/禁止时，CALL 功能调用程序块 DISPLAY。



SVCREQ # 13 : 关断（停止）PLC

该功能用于在本次扫描结束时停止 PLC。所有输出都将在下次 PLC 扫描时变为开始时设定的默认状态。PLC 故障表将登录一

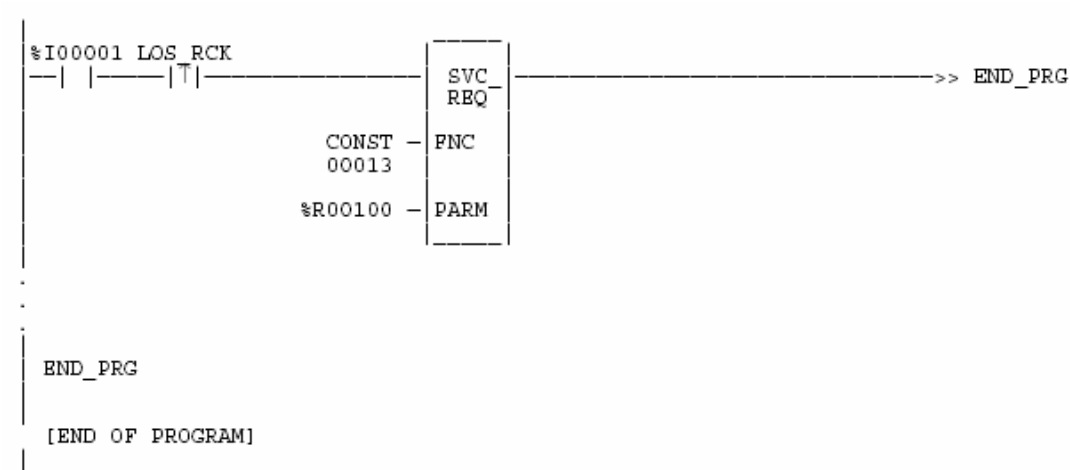
GFK-0265J

个故障信息，并将仅执行“SHUT DOWN PLC”（关断 PLC）功能块。

本功能块没有参数块；但 Logicmaster 90-70 软件需要为 PARM 参数指定一个入口地址。

例：
本例中，当使能输入%I00001 置位且发生机架未接通的故障时，将执行 SVCREQ # 13 服务功能。由于不需要参数块，所以输入 PARM 未被使用；但是编程软件需要给 PARM 参数指定一个入口地址。

本例使用一个 JUMP 功能，如果成功执行了关断 PLC 功能，程序直接跳到程序结尾处强迫关断 PLC。使用 JUMP 和 LABEL，是因为知道执行此功能的扫描到结尾时 PLC 才会转到 STOP 模式。



SVCREQ # 14 : 清故障表

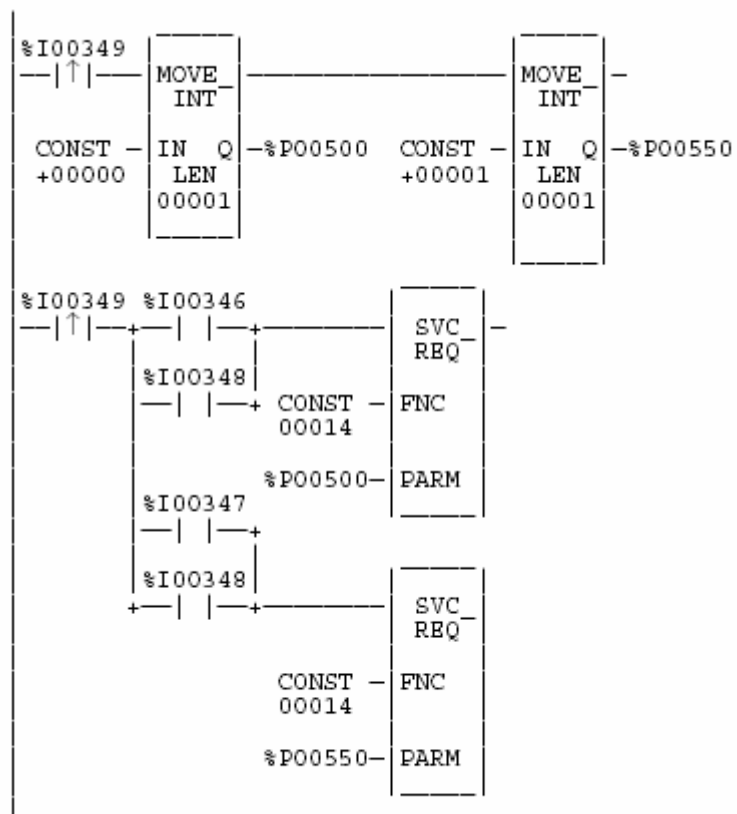
该功能用于清除 PLC 或 I/O 故障表。该功能的输出被设置为 ON，除非输入 0 和 1 以外的数作为请求的操作（如下）。

参数块仅是一个输入参数块；其长度为 1 个字。

0 = 清 PLC 故障 表	地址
1 = 清 I/O 故障 表	例： 本例中，当输入%I00346 为 ON 且输入 %I00349 发生

正跳变，PLC 故障表将被清空。当输入%I00347 为 ON 且 %I00349 发生正跳变，I/O 故障表将被清空。当输入%I00348 为 ON 且%I00349 发生正跳变，两个表均被清空。

PLC 故障表的参数块位于%P00500；I/O 故障表的参数块位于 %P00500。两个参数块可在程序中任何地方建立。



SVCREQ # 15 ： 读取最后登录的故障表信息

该功能用于读取 PLC 或 I/O 故障表中最后登录的信息。只要不输入 0 和 1 以外的书作为请求的操作（见下）或者故障表不是空的，那么该功能块的输出即 ON。（有关故障表入口详情，见第三章中的故障说明与排除）

参数块长度为 22 个字。输入参数块为如下格式：

0 = 清 空 P L C 故 障 表	地址 输出参数的格式 取决于是从 PLC 故障表还是从 I/O 故障表读出 数据。 PLC 故障表输出 格式 I/O 故障表输出 格式 高字节 低字 高字节 低字节
1 = 清 空 I/ O 故 障 表	

0	
	备用
P L C 故障地址	
故障组别和动作	
故障代码	
故障描述数据	
时间标记	

地址
地址+1
地址+2
地址+3
地址+4
地址+5
地址+6
地址+7
地址+8
地址+9
地址+10
地址+11
地址+12
地址+13
地址+14
地址+15
地址+16
地址+17
地址+18
地址+19
地址+20
地址+21

1	
给定	长 / 短
给定	
I/O 故障地址	
故障组别和动作	
故障类型	故障类别
	故障说明
故障描述数据	
时间标记	

GFK-0265J

地址+1 的最低有效字节处：长/短指示器定义了故障表入口中列出的故障描述数据的长度。

如上页内容所述，输出参数块的格式取决于此功能是否读取 PLC 故障表，I/O 故障表，扩展 PLC 故障表或者扩展 I/O 故障表。扩展格式如下所示：

扩展 PLC 故障表输出格式		扩展 I/O 故障表输出格式
高字节	低字节	高字节
低字节		

80h	
备用	
PLC故障地址	
故障组别和动作	
故障代码	
故障描述数据	
时间标记	
	保留

地址
地址+1
地址+2
地址+3
地址+4
地址+5
地址+6
地址+7
地址+8
地址+9
地址+10
地址+11
地址+12
地址+13
地址+14
地址+15
地址+16
地址+17
地址+18
地址+19
地址+20
地址+21
地址+22
地址+23

GFK-0265J

81h	
给定	长 / 短
给定	
I/O 故障地址	
故障组别和动作	
故障类型	故障类别
	故障说明
故障描述数据	
时间标记	
保留	

12486

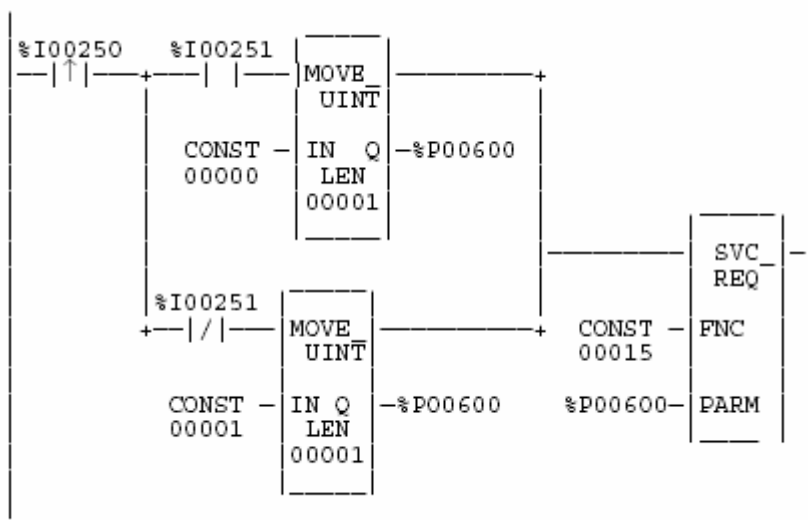
因为扩展故障表比标准故障表多 2 个字长（4 个字节），需要读取的末位寄存器的数量增加了。下表列出了在每个故障表中需要读取的末位故障入口的字节数量。

故障表	字节数量
PLC 故障表扩展 PLC 故障表	00=8 字节 (短整型) 01=24 字节 (长整型)
I/O 故障表扩展 I/O 故障表	02=5 字节 (短整型) 03=21 字节 (长整型)

例 1:

在本例中，当输入%I00251 为 ON 且输入%I00250 发生正跳变时，PLC 故障表中最后登录的信息被读入参数块。当输入%I00251 为 OFF 且%I00250 发生正跳变时，在 I/O 故障表中最

后登录的信息被读入参数块。该参数块位于全局内存%P00600处。



例 2:

在下例中，除 0 机架第 9 槽和 1 机架第 9 槽的模块出现故障外，其它任何一个 I/O 模块发生故障都会导致 PLC 关断。但如果故障发生在 0 机架和 1 机架的第 9 槽模块上，系统仍将继续运行。在第一次扫描时建立“表类型”参数。触点 IO_PRESS 置位时，表明 I/O 故障表有故障信息。当故障被登录进故障表后，PLC CPU 便在扫描中设置上升跳变触点。如果故障是在连续两次扫描中被记入表内，那么连续两次扫描中正跳变触点均被置位。

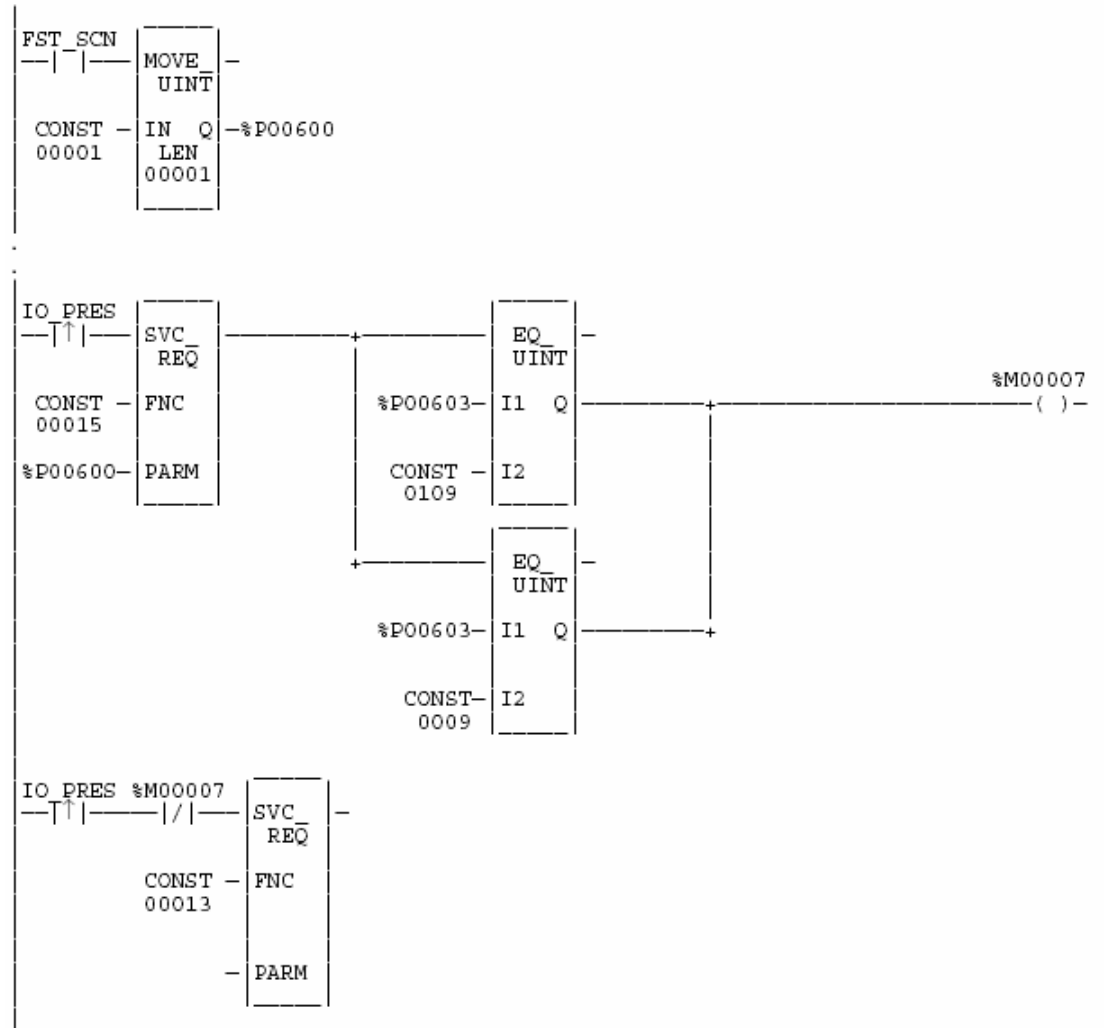
该例使用了位于全局内存%00600 处的参数块。SVCREQ 功能执行后，此参数块的第 4、5、6 字将包含有发生故障的 I/O 模块的地址：

1	
长/ 短	
变 量 地 址	
机 架 号	槽 号
I/O 总 线 号	总 线 地 址
点 地 址	

%P00600
%P00601
%P00602
%P00603
%P00604
%P00605

故障数据

在程序中，EQ_UINT 模块对表中的机架/槽的地址和 16 进制常数进行比较。当发生故障的机架/槽号与以上给定值一致时，内部线圈%M00007 被接通。如果%M00007 为 ON，它的常闭接点将打开从而禁止 PLC 关断。相反，如果%M00007 因为故障出现在不同的模块上而变为 OFF，则常闭接点将闭合并使 PLC 关断。



SVCREQ # 16 ： 读取耗时时钟

该功能可以读取系统的耗时时钟。该时钟跟踪从PLC上电以来所经过的秒数。该时钟每隔 100 年翻转一次。

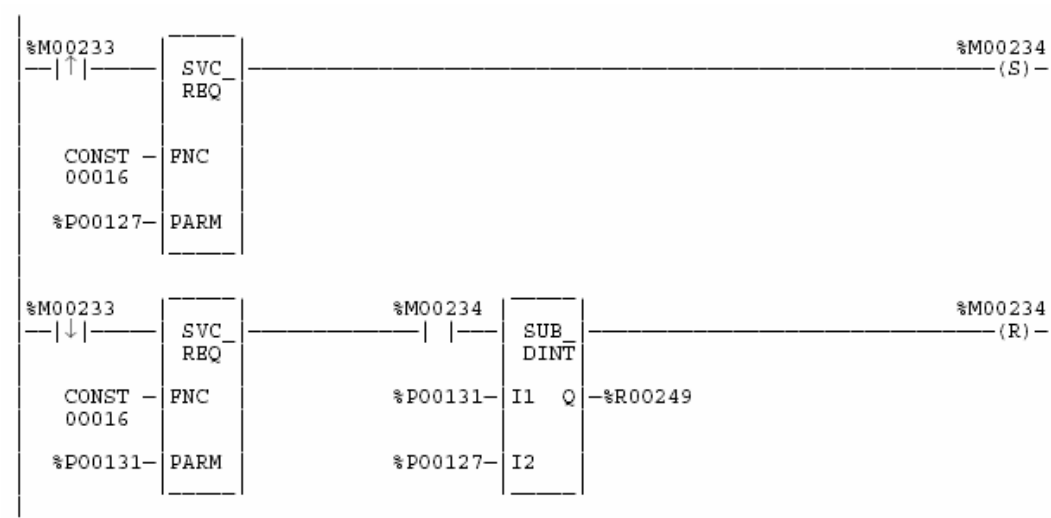
参数块仅是一个输出块，长度为 3 个字节。

上 电 以 来 的 秒 数 （ 低 位）	地址 地址+1 地址+2 前两个字是 已经过的时 间秒数。最 后一个字为 当前秒中 100 微秒数。
上 电 以 来 的 秒 数 （ 高 位）	例： 本例中，当 内部线圈 %M00233 发 生正跳变 时，读出耗 时时钟，同 时使线圈 %M00234 接 通。当 %M00233 发 生负跳变 时，该值又 被读取一
100 微 秒 时 间 段	

遍。这两个值之差将被计算出来并将差值放入寄存器内存 %R00250 中。

第一次读取值的参数块在%P00127 处；第二次读取的参数块在 %P00131 中。计算结果将忽略 100 微秒时间段值，这是由于

DINT 类型为有符号数值。该计算值在 PLC 上电后 50 年内准确无误。



SVCREQ # 17 : 屏蔽/去除屏蔽 I/O 中断

该功能用于屏蔽/去除屏蔽输入板的中断。当屏蔽掉某一中断时，即使发生输入跳变并导致中断产生，PLC CPU 也不会去执行相应的中断块。

参数块仅是一个输入参数块；其长度为 3 个字。

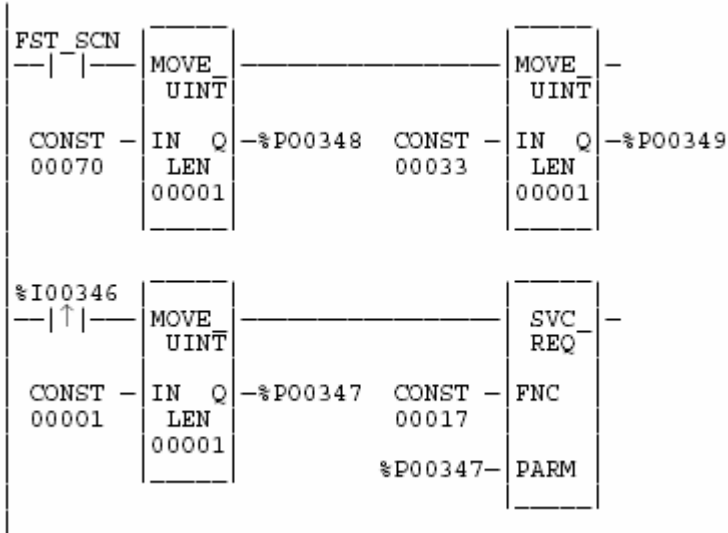
0=	地址
去	地址+1
除	地址+2
屏蔽	
输入	
1=	“内存类型”为一个十进制数，它位于字地址+1的字节中。它与输入的内存类型相对应：
屏蔽	
输入	
内存	70 = I 内存，位模式
类型	
变量	10 = AI 内存
	出现下述情况将

导致该功能执行失败：

- 输入 0 或 1 以外的数作为请求操作
- 被屏蔽或去除屏蔽的输入的内存类型不是%I 或%AI。
- 该 I/O 板不是合适的输入模块
- 该板的中断被禁止
- 所声明的变量对 90 系列输入模块不是一个可屏蔽的通道。

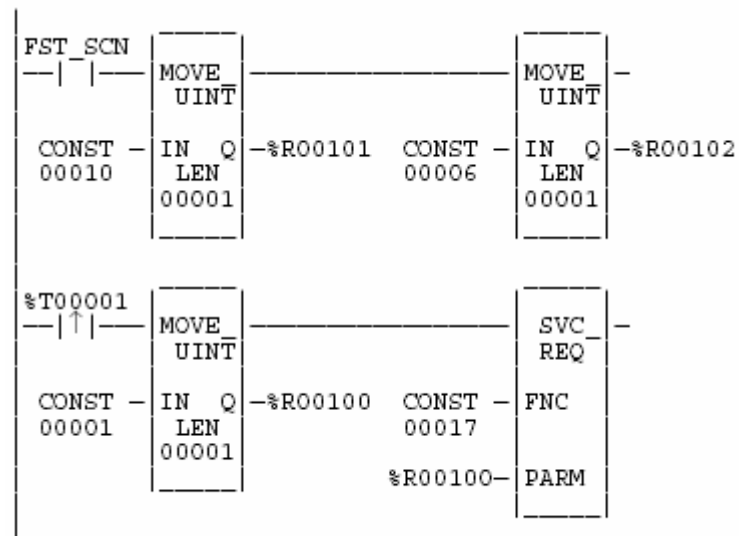
例 1：

当%I00346 发生正跳变，来自输入%I00033 的中断被屏蔽。位于%P00347 处的参数块将在第一次扫描时建立。



例 2:

在下面的例子中，当%T00001 正跳变时，%AI0006 输入的警告中断被屏蔽。%R00100 开始的参数块在第一次扫描的时候被赋值。



SVCREQ # 18 : 读取 I/O 强制状态

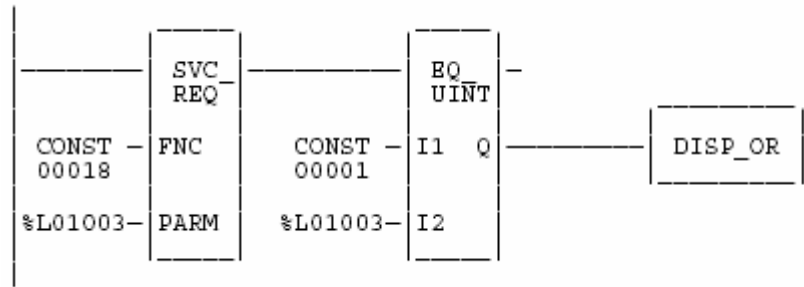
GFK-0265J

12-95

该功能用于读取 CPU 中当前的强制状态。
 参数块仅是一个输出块；长度为一个字。

0	地址
=	
无	
强	
制	
1	例：
=	
有	
强	
制	

本例中，I/O 的强制状态被不断地读到%L01003 中。如果存在强制，那么调用程序块 DSP_OR。



说 明

SVCREQ # 18 功能不检测%G 或%M 内存类型的强制状态。可以使用%S0011（OVR_PRE）来检测%I，%Q，%G 和%M 内存类型的强制状态。

SVCREQ # 19 : 设置运行允许/禁止

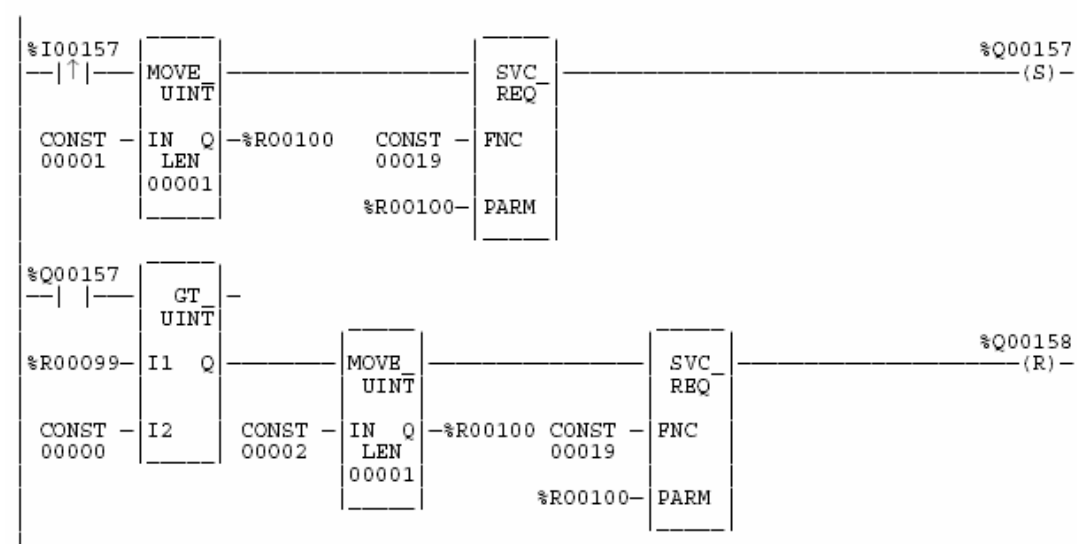
该功能允许梯形图程序去控制 CPU 的运行模式。

参数块给出所执行功能的选择。如果该功能成功完成，输出 OK 将变为 ON。如果所请求的操作模式不是“设置运行禁止”模式 1 或“设置运行允许”模式 2，OK 输出置为 OFF。

地 址	1 = 设 置 运 行 禁 止 模 式 2 = 设 置 运 行 允 许 模 式

例：

当输入%I00157 发生正跳变时，设置运行禁止模式。在该功能成功执行后，%Q00157 接通。当%Q00157 为 ON 且寄存器 %R00099 大于 0，该模式又将改变为运行允许模式。该功能成功完成后，线圈%Q00157 断开。



SVCREQ # 20 : 读取故障表

该功能用于读取整个 PLC 或 I/O 故障表并把它返回到指定的寄存器中。第 1 个输入参数指定被读取的故障表。第 2 个输入参数（现在始终为 0）允许以后对该功能进行扩展，以读取指定的故障信息。故障表中的数据放在紧随输入参数的参数块中。

若该功能成功完成，则 OK 输出置为 ON。如果所请求的操作不是“0”（读取 PLC 故障表）或“1”（读 I/O 故障表）时，

输出 OK 将置为 OFF。如果指定的故障表是空的，该功能块把 OK 输出置为 ON，但不返回任何数据。

参数块是一个输入和输出参数块。输入参数块的格式为：

地址	0 = 读 取 P L C 故 障 表
	1 = 读 取 I/ O 故 障 表
地址 + 1	总 是 为 0

输出参数块的格式为：

地址	0 = P L C 故 障 表 1

	= I/ O 故障 表
地址+1	总 为 0
地址+2 地址+14	未 使 用
地址+15 +16 +17	上 一 次 清 空 故 障 表 到 现 在 所 经 历 的 时 间
地址+18	上 次 清 空 故 障 表 后 的 故 障

GFK-0265J

	数
地址+19	队 列 中 故 障 的 数 量
地址+20	读 取 到 的 故 障 数
地址+21	故 障 数 据 的 开 始

故障表中每一项故障信息长度为 21 个字（42 个字节）。PLC 故障表最多可登录 16 项故障而 I/O 故障表最多可达 32 项。如果故障表清空，则不会有数据返回。

说 明

SVCREQ # 20 功能需要有 693 个以上的寄存器可用时才能正常工作。

参数块是输入输出参数块。读扩展 PLC 故障表（80h），读扩展 I/O 故障表（81h），输入参数块格式如下：

地 址	80h = 读扩 展 PLC 故障 表 81h = 读扩 展 I/O 故障 表
地 址 + 1	所读 取故 障的 起始 位置
地 址 + 2	所读 取的 故障 数量

输出参数块格式如下：

地址	80h = PLC 故 障 表 81h = I/O 故 障 表
地址+1	所

	读取的故障表起始位置
地址+2 地址+14	未使用
地址+15 +16 +17	上一次清空故障表到现在所经历的时间
地址+18	上次清空故障表后的故

	障 数
地址+19	队 列 中 故 障 的 数 量
地址+20	读 取 到 的 故 障 数
地址+21	PLC 名 称
地址+22	
地址+23	
地址+24	
地址+25	
地址+26	
地址+27	
地址+28	
地址+29	
地址+30	
地址+31	
地址+32	
地址+33	
地址+34	
地址+35	
地址+36	
地址+37	故 障 数 据 的 开 始

GFK-0265J

读取扩展 PLC 故障表（80h）以及读取扩展 I/O 故障表（81h）时，每个扩展故障表的入口地址都是 23 个字长（46 字节）。最多可以有 40 个 PLC 故障表入口和 40 个 I/O 故障表入口。默认值为 16 个 PLC 故障表入口和 32 个 I/O 故障表如。如果故障表读数为空，则没有数据返回。

说 明

对于非扩展格式故障表，SVCREQ # 20 功能只有 693 个连续寄存器可用（从起始点开始）的情况下才能正常工作。而对于扩展格式，PLC 计算需要读取的入口数量，如果寄存器数量不够，将返回一个错误。

扩展故障表格式总长度为头部长度+（故障入口数量）*（故障入口长度）。

下表显示了 PLC 故障表入口和 I/O 故障表入口的返回格式。

地 址	PLC 故 障 表	I/O 故 障 表
地 址 + 21	长/ 短	长/ 短
地 址 + 22	备 用	参 考 地 址
地 址 + 23	PLC 故 障 地 址	I/O 故 障 地 址
地 址 + 24	PLC 故 障 地 址	I/O 故 障 地 址
地	故	I/O

址 + 25	障 组 别 和 动 作	故 障 地 址
地 址 + 26	错 误 代 码	故 障 组 别 和 动 作
地 址 + 27	故 障 附 加 数 据	故 障 目 录 和 类 型
地 址 + 28	故 障 附 加 数 据	故 障 描 述
地 址 + 29	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 30	故 障 附 加 数 据	故 障 详 细 说 明

GFK-0265J

		数 据
地 址 + 31	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 32	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 33	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 34	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 35	故 障 附 加 数 据	故 障 详 细 说 明

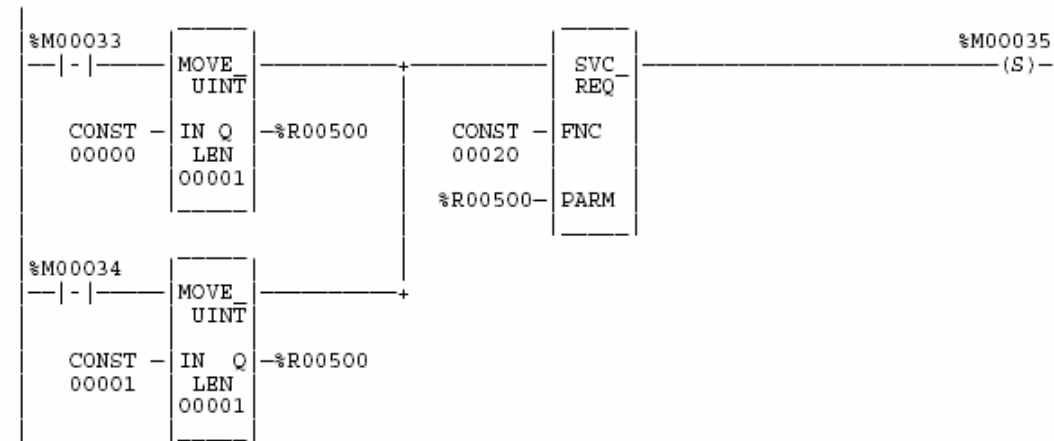
		数 据
地 址 + 36	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 37	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 38	故 障 附 加 数 据	故 障 详 细 说 明 数 据
地 址 + 39	时 间 标 记	时 间 标 记
地 址 + 40	时 间 标 记	时 间 标 记
地 址 + 41	时 间 标 记	时 间 标 记

地址+21 的第一个字节长/短指示器定义了故障入口所标明的故障数据的数量。PLC 故障表中，长/短值为 00 表明故障入口中

有 8 个字节的故障附加数据，01 表明有 24 个故障附加数据。
I/O 故障表中，02 表明有 5 个字节的故障详细说明数据，03 表明有 21 个字节故障详细说明数据。
各块的详细说明参见附录 B，“Logicmaster 90-70 软件故障表使用说明”。

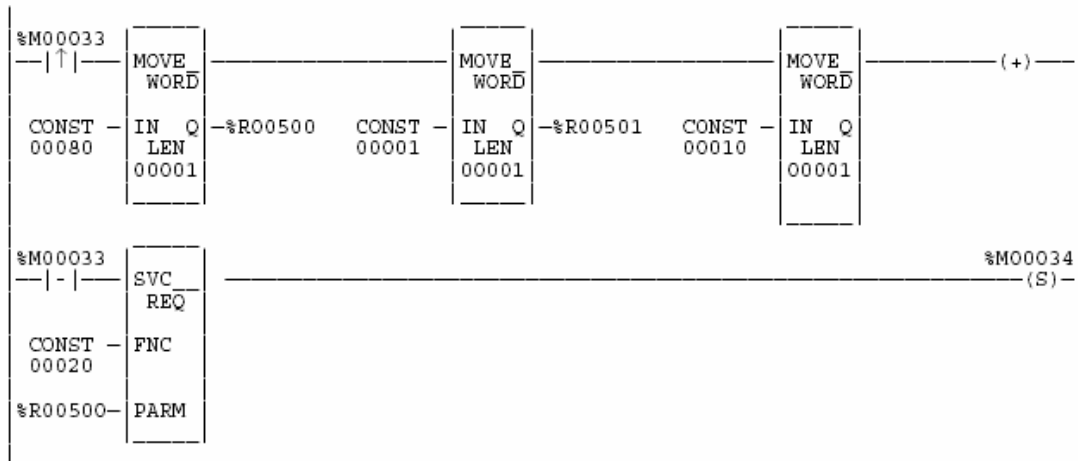
示例：

下例中，当输入%M00033 发生正跳变，将读 PLC 故障表。当 %M00034 发生正跳变，则读 I/O 故障表。参数块位于 %R00500。该功能成功完成后，线圈%M00035 接通。



扩展格式示例：

下例中，当输入%M00033 发生正跳变时，将读取扩展 PLC 故障表。参数块位于 %R00500。%R00500 包含了故障表类型（PLC 扩展）； %R00501 包含了起始故障， %R00502 包含了从 %R00501 中故障开始的故障数量。当 SVCREQ 功能成功执行，线圈%M00034 接通。



SVCREQ # 21：用户自定义故障的输入

该功能用来定义一个能在 PLC 故障表中显示出来的故障。该故障包括二进制信息或 ASCII 码信息。该用户自定义的故障代码开始于十六进制的 0。

鼓掌的错误代码必须介于 0~2047 之间，以便显示 “Application Msg:” 信息。如果错误代码介于十进制的 81~112 之间，那么 PLC CPU 将在 %SA 系统内存中设置相同个数的故障位。这就允许多达 32 位可被分别设置。

错误代码	状态位
0~80	没

GFK-0265J

81~112 113~2047 2048~32767	有位 被 设置 设置 设置 % S A 没有 位 被 设置 保留

当 EN 被激活时，由 IN 给定的故障数据数组被当作一项故障被输入到 PLC 故障表中。如果 EN 未被激活，OK 位将被清除。如果错误代码超出范围，OK 位将被清除且不输入该故障。

参数块仅是一个输入参数块，它的格式如下：

参 数	高 位	低 位
地 址	错误代码	
地 址 + 1	文 本 2	文 本 1
地 址 + 2	文 本 4	文 本 3
地 址 +	文 本 6	文 本 5

3		
地址 + 4	文本 8	文本 7
地址 + 5	文本 10	文本 9
地址 + 6	文本 12	文本 11
地址 + 7	文本 14	文本 13
地址 + 8	文本 16	文本 15
地址 + 9	文本 18	文本 17
地址 + 10	文本 20	文本 19
地址 + 11	文本 22	文本 21
地址 + 12	文本 24	文本 23

输入参数数据允许选择一个 0~2047 之间的错误代码和文本信息，此文本将被放入一个长 PLC 故障的附加区中。该功能块将填写 PLC 故障地址，故障组和故障动作。

故障文本的 1~24 字节用于传递故障的二进制或 ASCII 码数据。如果故障文本数据的第一个字节非 0，那么该数据将是 ASCII 码信息串。可以在故障表中的说明区中显示出这些信息。如果该信息小于 24 个字符，那么这个 ASCII 码串必须用 NULL 字节来结束。编程器会显示提示信息：“Application

Msg:”，并在其后显示该 ASCII 数据。当输入代码介于 1 和 2047 之

间时，该错误代码号将紧跟“Msg:”显示出来。如果故障代码超过 2047 它将被转换为故障代码 0。

如果文本的第一个字节为 0，那么在故障说明区中仅显示“Application Msg:”信息。其余的 1~23 个字节将被认为用户自定义故障数据的二进制信息。通过热键 CTRL-F 可以在 Logicmaster 90-70 编程器 PLC 故障显示处显示该数据。

更详细的信息，可以参阅《编程软件用户手册》（GFK-0263）第五章“PLC 控制与状态”。

示例：

本例中，传递给 IN1 的值为故障错误代码，该值为 16 进制的 0057，代表错误代码 87 并作为故障信息的一部分出现。以下输入的值是故障信息文本的 ASCII 码。对于 IN2 输入 2D45。低字节 45 对应字符 E 而高字节 2D 对应字符_。以此类推，接下来是字符 S、T、O、P、O、N。最后一个字符 00 为一个空字符以结束该字符串。因此，这些字符连起来便是 E_STOP ON。

FST - EXE		BLKMW		%Q00001	
-- --		WORD		--()--	
CONST	IN1	Q	→P00001		
0057					
CONST	IN2				
2D45					
CONST	IN3				
5453					
CONST	IN4				
504F					
CONST	IN5				
4F20					
CONST	IN6				
004E					
CONST	IN7				
%I00050		SVC		-	
-- --		REQ			
CONST	FNC				
00021					
%P00001	PARM				

SVCREQ # 22 ： 屏蔽/去除屏蔽计时中断

该功能用于屏蔽计时中断或去除计时中断的屏蔽并读取当前的屏蔽。当中断被屏蔽后，PLC CPU 将不会去执行与计时中断相对应的中断块。把计时中断作为一个组来屏蔽或去除屏蔽。它们不能被单独屏蔽或去除屏蔽。

只要不在参数块中输入 0、1 以外的数据作为请求操作或屏蔽值，该功能均能被成功执行。

参数块是一个输入和输出参数块。

为确定当前的屏蔽，使用如下格式：

0	地址
=	
读	
取	
中	
断	
屏	
蔽	

PLC 将返回如下格式的信息：

0 = 读 取 中 断 屏 蔽
0 = 计 时 中 断 屏 蔽 被 去 除 1 = 计 时 中 断 被 屏 蔽

地址
地址+1

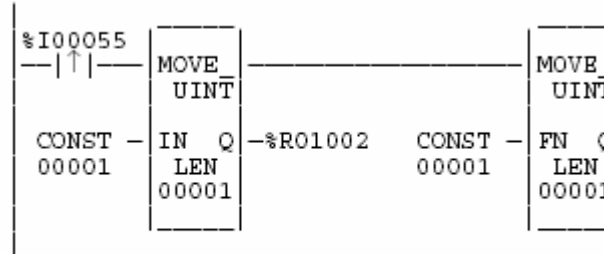
要改变当前的屏蔽，使用如下格式：

1 = 屏蔽 / 去除 屏蔽 计时 中断
0 = 去除 计时 中断 屏蔽 1 = 屏蔽

地址
地址+1

例：

当输入
%I00055 发生
正跳变后，将
屏蔽计时中
断。



SVCREQ #
23 : 读取主

中 断

校验和

该功能可以为
用户程序和配置返回主校验和。它也可以从该服务请求所建立的
块返回校验和。如果这个块是_MAIN，那么对_MAIN 的校
验和将作为该块的校验和被返回。

当“运行模式存储”被激活时，程序校验和可以无效直到存储
工作结束为止。为了确定何时校验和有效，在输出程序块的起
始部分提供了三个标志（程序块校验和，主程序检验和主配置
检验和）。

对此服务请求没有输入参数块。输出参数块格式如下，它需要 15 个字内存。

输出参数块

字地址

程序块检验和有效（0=无效，1=有效）		地址
主程序检验和有效（0=无效，1=有效）		
主配置检验和有效（0=无效，1=有效）		
程序块个数（包括_MAIN）		
用户程序的字节数（DWORD 类型）		地址 + 1
程序加法检验和		
程序 CRC 检验和（DWORD 类型）		
配置数据的字节数		
配置加法检验和		地址 + 2
配置 CRC 检验和 （DWORDLEIXING ）		
总是为0	* 该程序块的加法检验和	
“This” 程序块 CRC 检验和		地址 + 4

地址+6

地址+7

地址+9

地址+10

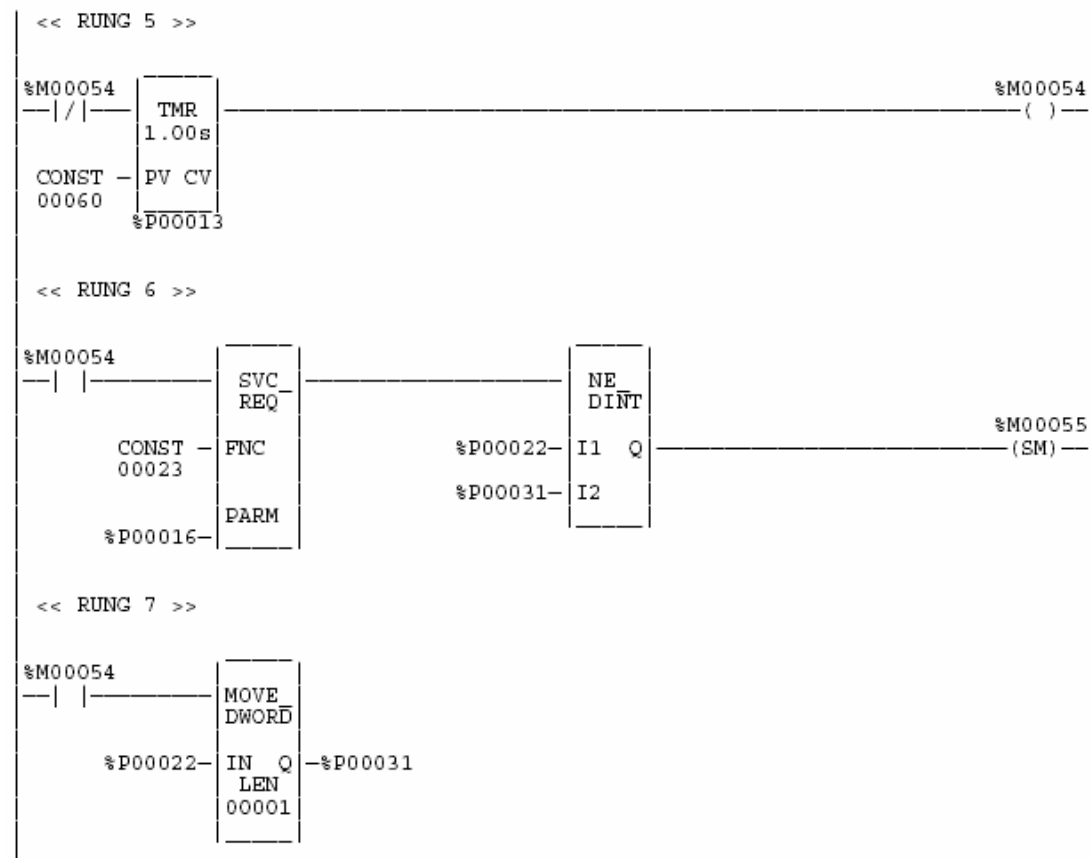
地址+11

地址+13

地址+14

例：

本例中，当使用寄存器%P00013~%P00015 的计时器计时满后，开始执行检验和读取。检验和数据返回到寄存器%P00016~%P00030。位于寄存器%P00022 和%P00023 中的主程序检验和（该检验和数据为一个 DWORD 数据类型并且占用两个连续的寄存器）同上一次所保存的主程序检验和进行比较。如果不相同，线圈%M00055 将保持为 ON。当前主程序检验和数据被保存到寄存器%P00031~%P00032 中。



该功能用于允许或禁止在后台检验和计算中包含 EXE 块。默认情况为包含该检验和的计算。

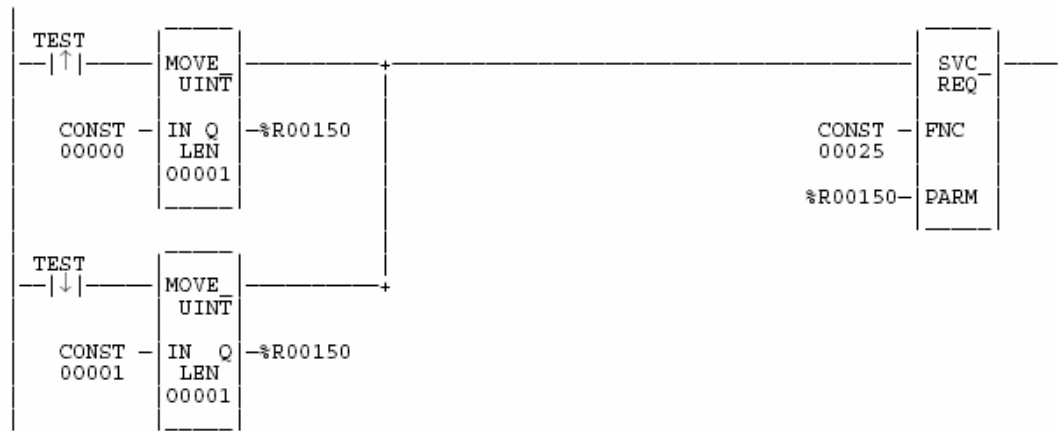
该服务请求仅需要一个输入参数块。

0 = 禁止在 检验和的计 算中包含 EXE 块 1 = 允许在 检验和的计 算中包含 EXE 块	地址 在执行该 服务请 求后， 参数

块不发生变化。

例：

本例中，当线圈 TEST 从 OFF 变为 ON 时，SVCREQ # 25 便禁止在后台检验和计算中包含 EXE 块。当线圈 TEST 从 ON 变为 OFF 时，该功能块将允许在后台检验和计算中包含 EXE 块。



SVCREQ # 26 : 任务切换

说 明

该功能仅在 780 CPU 模块（IC697CPU780）有冗余时才有效，在 IC697CGR772 和 IC697CGR935 增强型冗余 CPU 上也有效。更多关于冗余 CPU 的信息参考 90-70 系列冗余 CPU 用户手册，GFK-0827。更多关于增强型冗余 CPU 的信息参考 90-70 系列冗余 CPU 用户手册，GFK-1527。

如果 CPU 之间同步且满足相互切换请求的时序要求，则本功能可以使 CPU 组在下次扫描时相互切换。（现用 CPU 切换为备用 CPU 而备用 CPU 转为活动 CPU）。两次手动切换的时间间隔不能少于 10 秒。这种切换允许失败或再同步（未使用 10 秒中限制）。

说 明

当 SVCREQ # 26 功能块输出电流时，表明在下次扫描时可能会进行 CPU 组切换。但并不表明已产生切换或下次扫描时一定会进行切换。

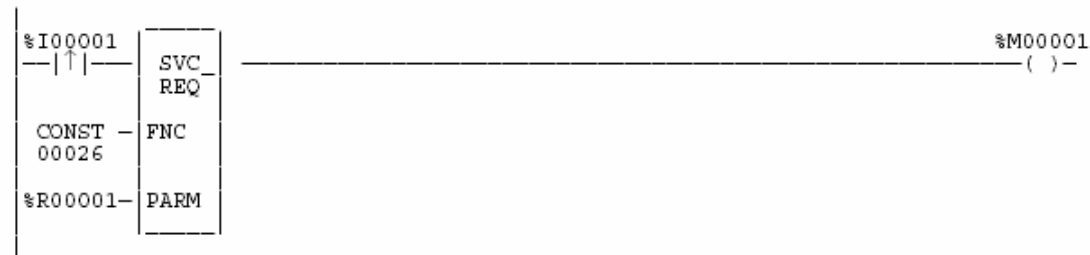
该功能没有相关的参数块；但是编程软件要求给参数 PARM 指定一个入口。在此输入任何适当的变量；且它不应被使用。

例：

本例中，位于操作台上的某个开关接在 %I00001 处，即 SVCREQ # 26 功能块。当开关闭合时，将激活 SVCREQ # 26 功能块，从而在 CPU 间产生任务切换。

GFK-0265J

12-123



10 秒限制允许 SVCREQ 在两个 CPU 中同时使用，当两个 CPU 同时检测到输入时，仅发生一次切换。

SVCREQ # 27 和 SVCREQ # 28 : 写入/读取交换传送区

SVCREQ # 27 和 SVCREQ # 28 对于 IC697CPU780 和 IC697CGR935CPU 上使用冗余 CPU 有效。关于此服务请求的使用信息参考手册 GFK-1527（在系列 90-70 增强型冗余 CPU 从后备单元到活动单元的编程数据交换手册）

SVCREQ # 32 : 中止/恢复 I/O 中断

使用 SVCREQ # 32 中止一系列 I/O 中断，并且将这些中断放到队列中直到中断恢复。这一系列的 I/O 中断能够在 90-70 高速计数器上进行。I/O 中断队列的数量取决于 I/O 模块的容量。PLC CPU 通知 I/O 模块中止还是恢复 I/O 中断。I/O 模块的默认状态是恢复 I/O 中断。中止请求应用于所有 I/O 模块相关的 I/O 中断。中断应该在单个周期内中止和恢复。

本服务请求只在输入参数块中使用。长度为 3 个字长。

GFK-0265J

12-125

除非出现以下情况，此服务将成功执行：

- 第一个参数是 0 或 1 以外的数
- 存储器类型参数不是 70（%I 存储器）
- 与指定地址相关的 I/O 模块不符合此操作。（模块必须是 90-70 高速计数器）
- 指定的参考地址不是高速计数器的第一个 %I 参考地址。
- PLC CPU 和 I/O 模块之间通讯失败。（模块丢失或发生致命错误。）

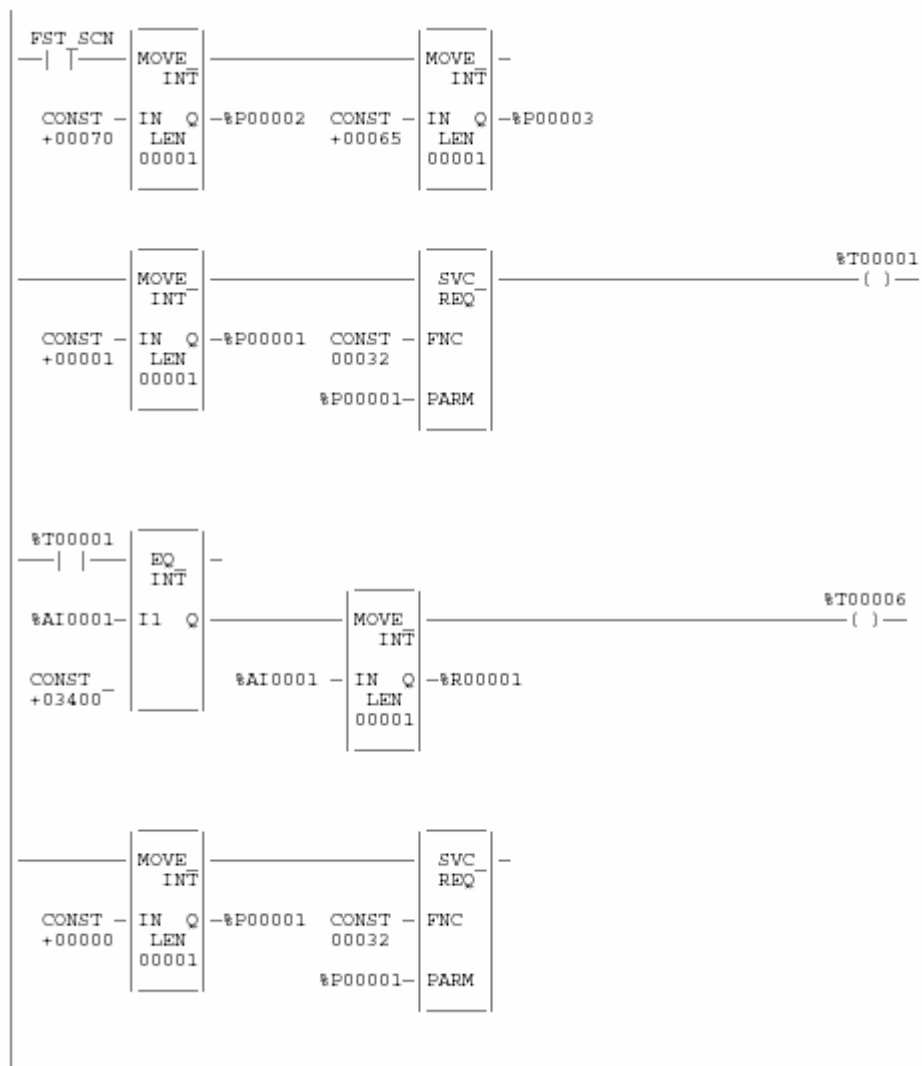
例：

下例中，以 %I00065 为起始参考地址的高速计数器模块中断将在 CPU 处理完第二行逻辑之后恢复。如果没有恢复中断，HSC 的中断将在第三行的执行中发生，如果 %R00001 的值不是 3400 %T00006 将被置位。（%AI 是高速计数器的第一个非离散量输入地址。）

说 明

除非被恢复或屏蔽，I/O 中断将正常中断功能块的执行。此服务请求最多应用是防止中断作用来进行诊断或其它目的。

例：



SVCREQ # 39 : ESCM 端口状态

使用服务请求# 38 ESCM（嵌入式串口模块）端口状态为 PLC CPU 提供端口 1 和 2 的状态。参数中的指定地址应该设置为 1 或 2（表明端口是 1 还是 2）。端口状态以下面的值返回。

1 = Port 1 2 = Port 2	地址
状态返回 位置	地址+1

除非发生以下情况，程序将正常执行：

- 第一个参数不是 1 或 2
- CPU 不支持端口 1 和 2 的使用

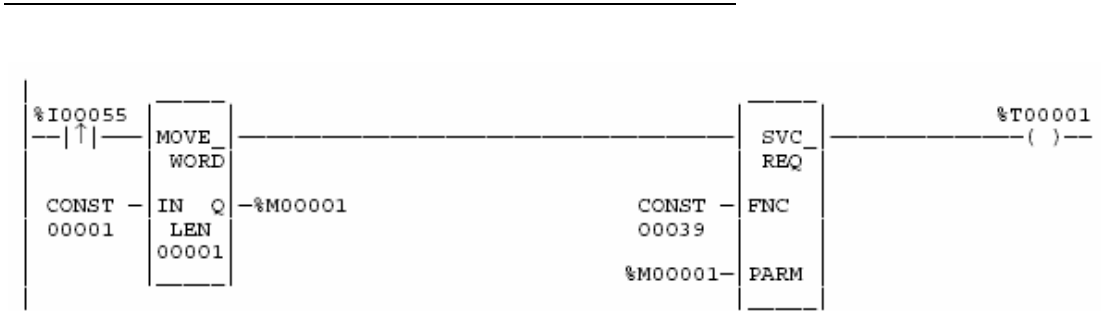
说 明

只有 CPX 和 CGR 的 CPU 模块支持端口 1 和 2。

例：

在下例中，当服务请求# 39 执行时，CPU 由命令字（即下例所示的%M00001）决定端口号，然后使此端口号生效。之后重新获取当前相应的 ESCM 状态字，并且将此状态字放入 %M00017 中。

如果端口号无效，或者输入电流为 OFF 状态，或者当前 CPU 不支持端口 1 和端口 2 的使用，那么输出电流为 OFF。如果端口有效，CPU 支持端口 1 和端口 2 的使用，输出电流将为 ON。



返回值

ESCM 状态信息以位编码的字格式进行传输。前四位中（以重要性最低的位开始）的每一位代表了 ESCM 状态的一些信息(如下表所示)：

状态	位所在位置	
PORTN_OK	0	

PORTN_ACTIVE	1	

PORTN_DISABLED	2	
PORTN_FUSE_BLOWN	3	

GFK-0265J

GFK-0265J

12-132

RESERVED	4~16	

SVCREQ # 44 : 逻辑驱动型动态 EGD 通讯

逻辑驱动型动态 EGD 通讯在 7.91 和更新的 CPX CPU 上可用，这种功能允许通过逻辑程序动态建立和中止 EGD 数据交换。这种功能利用 SVCREQ # 44 来建立、中止和监控数据交换。EGD 数据交换不必对硬件进行配置；但是必须通过配置软件对适配器进行配置。

适配器模块 IC697CMM742 2.70 或者更新的版本支持逻辑驱动型 EGD 数据交换。

每个 PLC 最多允许有 32 个逻辑驱动型数据交换。总共可能有 255 个数据交换。每个逻辑驱动型数据交换最多可以有 8 个变量。

在同一个 CPU 中如果有另外一个 SVCREQ # 44 正在执行，SVCREQ # 44 不能来自中断程序块。

GFK-0265J

12-134

说 明

在数据交换建立的扫描周期中，CPU 在开始扫描进程时需要更多的时间。另外，如果要在一个扫描过程中开始多个 LD EGD 连接，看门狗的超时时间可能需要进行调整。

服务请求功能块

表 12-2 SVCREQ # 44 功能块的通用格式

状态字 指示请求成功或是失败。参考每个命令的一系列可能响应。用户需要 将此区域初始化为 0。	地 址
命令字 指示将要执行的命令 1 设置本地 producer（发送器）ID 号 2 重新获得本地 producer（发送器）ID 号 3 建立一个 produced（发送）数据 交换 4 建立一个 consumed（接收）数 据交换 5 中止 produced （发送）数据交换 6 中止 consumed （接收）数据交换 7 每个扫描周期 刷新 production（发送） 数据	地 址 + 1
请求指定信息功能快 其它区域的格式取决于所选择的命令。参考后面关于每个命令字的说明。	地 址 + 2

	 地址 + N

返回的状态值

下面的常见错误能够返回到状态字中。其它的值可能出现于其它命令。

表 12-3. 适用于所有命令的命令状态

值	说明	通过 电 流
-18	无效请求，不在1-7范围之内。	否
-19	参数块	否

	长度超过内存范围	
-23	服务请求# 44 被中断。即在上一个命令完成之前，发出新的命令。	否

服务请求命令的详细内容

表 12-2 列出了 7 个本服务请求能够初始化的命令。每个命令的详细内容在本节进行说明。

命令 1-设置本地 Producer（发送）ID

本地 Producer（发送）ID 将本 PLC 和其它数据交换的接受和发送器。这个本地发送器在数据交换创建之前必须已经存在并且有一个非零值。有两种方法来创建一个发送器 ID：1）利用控制软件配置此 ID，或者 2)利用本地发送器 ID 设置命令。一旦设置了一个非零值，本地发送器 ID 将不能通过逻辑进行修改。

本地发送器 ID 通过 4 个字进行指定。每个字包含了由点（.）进行分割的 IP 地址的一部分。当用此命令来指定本地发送器 ID 时，4 个字中只须包含一个非 0 值即可；但是，要确保每个 PLC 有唯一 IP 地址。如果硬件配置尚未保存到 PLC CPU 中，那么不允许对 ID 进行配置。

如果本地发送器通过这个命令进行配置，当 CPU 转为 STOP 状态或者电流在 CPU 循环后将重设为 0.0.0.0。

表 12-4. 设置本地发送器 ID 的命令格式

状态 参见表 12-5。用户应该将这个区域初始化设置为 0。	地址
本地发送器设置命令 总为 1。	地址 + 1
本地发送器 ID 4 个字包括了发送器 ID 的 4 个部分。每个字的值从 0 到 255。至少一个字为非零值。	地址 + 2 到地址 + 5

如下表所示，命令状态字在设置服务请求完成之后返回。、

表 12-5. 设置本地发送器 ID 命令的状态

值	说明	通过电流是
1	ID 按照请求进行设置或者指定的 ID 与已存在的 ID 相同。	

-2	请求的ID与保存在已下载配置中的值不同。	否
-3	请求的ID与通过上面“设置发送器ID”服务请求	否

	所设置 的值不同。 。	
-11	没有保存的 硬件配置。 。	否
-16	无效的本地 发送器 ID（ 0 或超过范 围）	否

命令 2-重新获得本地发送器 ID

当 ID 已经通过控制软件设置好后，此命令将有用。如果此命令返回 0.0.0.0，本地发送器 ID 或者不存在或者值为 0，必须通过逻辑驱动交换设置为非 0 值。

表 12-6 重新获取本地发送器 ID 命令格式

状态字 参见表 12-7。用户须将此区域初始化设置为 0。	地址
重新获取本地发送器 ID 命令 总为 2。	地址 + 1
本地发送器 ID 4 个字包含了 CPU 当前发送器 ID 的 4 个部分。每个返回的字在 0 到 255 范围之内。	地址 + 2 到地址 + 5

如下表所示，命令状态在重新获取本地发送器 ID 命令执行完之后进行返回。

表 12-7 重新获取本地发送器 ID 命令状态

值	说明	通过电流
1	ID 通过设置本地发送	是

	器 ID 进 行 配 置 。	
2	ID 在 保 存 的 配 置 中 进 行 配 置 。	是
-1	本 地 发 送 器 ID 没 有 进 行 设 置 。	否

命令 3 和 4-建立发送交换/接收交换

数据交换的建立要求程序进行复合扫描。然而，如果满足下列条件，在扫描过程中将通过电流：

- 本地发送 ID 不是 0.0.0.0。
- 需要建立的交换（通过发送 ID 和交换 ID 进行区分）不存在。
- 变量地址有效并且指定的变量数量没有超过允许的最大数量（8）。
- 交换的创建没有超过逻辑驱动的最大数量（32）或者允许的最大交换数量（255）。
- 有足够的用户存储区可用来创建数据交换。

如果以上的任一条件不满足，电流将不能通过，命令状态字将指示失败的原因。交换将不能建立并且交换状态字不会改变。

如果电流通过，此交换将创建，但是此交换还不能被用户程序使用。命令状态字将被置为 1，交换状态字将被设置为 IPLC_CREATE_IN_PROGRESS（24）。用户程序必须在后续扫描中监控交换状态字以判断交换何时创建完成（由 IPLC_CREATE_IN_PROGRESS 转变）。交换状态字设置为失败或完成之前最多有 12 秒时间。

如果交换状态字被设置为表 10 中的一个值时，交换没有被成功创建，同时本地交换定义将被删除。逻辑驱动交换的可能状态字的完整列表，参考表 12-19。

表 12-8 建立交换命令的交换状态字

交 换 状 态 字	说明
12	在适配器模块中没有足够的双端口寄存器。
18	适配器模块失败。

GFK-0265J

26	适配器模块在请求发出 6 秒之内没有响应。
28	其它失败。

交换能够类似地建立。如果有请求，用户程序在本次扫描过程中可以开始所有交换的创建。如果任何与交换相关的后续请求没有成功完成，创建将失败，当创建完成时，必须重新开始。

每个交换中最多允许 8 个变量。

对于每个逻辑驱动交换，PLC 用户需要的内存数量将在交换建立的时候进行分配。逻辑驱动交换使用的内存取决于配置和逻辑的可用内存数量。每个交换大概分配 $176+N*16$ 位，N 是交换中使用的变量数。

逻辑交换使用的总的用户内存数量和用户内存保留数量在当前版本的 Logicsmaster 90-70 是不可见的。

使用逻辑驱动 EGD 要求系统通讯窗口有足够的时间。如果系统通讯窗口没有运行，逻辑驱动 EGD 建立的交换将失败，交换状态字将设置为 26。

创建发送和接收交换的格式是不同的。此主题包含在下面几节中。

建立发送交换的命令格式

建立发送交换命令的格式如下所示。

表 12-9 建立发送交换命令的格式

GFK-0265J
12-145

状态 参见表 12-10。用户应 该将此区域初始化设置 为 0。	地 址
建立发送交换命令 总为 3。	地 址 + 1
保留 4 个字。必须设置为 0。	地 址 + 2 到 地 址 + 5
交换 ID 允许范围：32769 到 49151。	地 址 + 6
保留 2 个字。必须设置为 0。	地 址 + 7 地 址 + 8
适配器所在的机架号/ 槽号 高字节 = 机架号 低字节 = 槽号	地 址 + 9
发送周期 10 到 3,600,000 毫秒（2 个字）	地 址 + 10 地 址 +

	11
交换状态字定位 4 个字。关于此地址的格式参考 12-84 页“交换状态字”。不能使用 NULL 选择器。	地 址 + 12 到 地 址 + 15
目标地址类型 1 = IP 地址 2 = 名字字符串（必须以 0 结尾） 3 = 多播组 ID	地 址 + 16
目标地址值 16 个字。	地 址 + 17 到 地 址 + 32
变量数目 1 到 8	地 址 + 33
配置数据 每个变量 6 个字。关于此地址格式参考 12-86 页“可变目标地址”。	地 址 + 34 到 ...

下表列举了建立发送交换命令的可能状态值。

表 12-10 建立发送交换命令的状态字

值	说明	通过电流是
1	交换信息被接受。	是
-1	本地发送器 ID 没有设置。	否
-5	交换 ID 超出范围。	否
-6	交换已经	否

GFK-0265J

	存在。	
-7	交换太多。	否
-8	变量太多。	否
-9	超过最大变量数。	否
-10	可用资源不足。	否
-12	机架/槽的位置配置不合适，	否

	或者不支持EGD。	
-13	机架/槽无效或超出范围。	否
-14	没有指定状态字地址或状态字地址无效。	否
-16	保留字（3-	否



	6) 无效。 （ 必须 为 0。 ）	
-22	无效的 地址 类型。 必须 在 1-3 范围 内。	否
-24	为目 标地 址指 定的 值无 效。	否
-101 到	-101 表	否



-134	示 字 的 参 数 错 误 。	
-135 到 -177	-101 表 示 以 字 开 始 的 指 定 的 变 量 无 效 。	否

以下规则适用于发送交换：

- 同一个 PLC 内不允许存在两个发送交换。如果发送交换在同一个 PLC 内已经存在，此请求将被拒绝。
- 数据发送在交换创建完成之后立刻开始（可能需要几个扫描周期）。

建立接收交换命令的格式

建立接收交换命令的格式如下所示。

表 12-11 建立接收交换命令的格式

状态 参见表 12-12。用户应 该将此区域初始化设置 为 0。	地 址
建立接收交换命令 总为 4。	地 址 + 1
发送器 ID 4 个字。不能为 0.0.0.0。这个 4 个字是 交换发送器的 ID。而 不是当前 CPU 的本地 发送器 ID。	地 址 + 2 到 地 址 + 5
交换 ID 允许范围：对于逻辑驱 动交换为 32769 到 49151 或者对于静态交 换为 1 到 16383。	地 址 + 6
保留 2 个字。必须设置为 0。	地 址 + 7 地 址 + 8
适配器模块的机架/槽	地

GFK-0265J

高字节 = 机架号 低字节 = 槽号	址 + 9
接收周期 10 到 3,600,000 毫秒（2 个字）	地 址 + 10 地 址 + 11
交换状态字定位 4 个字。关于这个地址 参考 12-84 页“交换状 态字”。不能使用 NULL 选择器。	地 址 + 12 到 地 址 + 15
组 ID 0 使用点对点格式 1-32 使用多播组方式	地 址 + 16
更新超时时间 以毫秒为单位，必须为 0（没有使用）或者设置 为大于接收周期。范围 与接受周期相同。建议 超时时间至少设置为比 当前接收周期错误超时 时间大 20ms.	地 址 + 17 地 址 + 18
时间标志位置 4 个字。当 CPU 发送数 据时，时间标志用来记 录时间。如果此区域地 址使用了 NULL 选择 器，时间标记信息将无 效。关于这个地址的格 式和定位于这个地址的	地 址 + 19 到 地 址 +

GFK-0265J

时间标记的格式，参考 12-86 页“时间标 记”。	22
保留 10 个字。必须设置为 0。	地 址 + 23 到 地 址 + 32
变量数量 1 到 8	地 址 + 33

12

配置数据 每个变量 6 个字。关于 这个地址的格式参考 12-86 页“变量地址定 义”。	地 址 + 34 到 ...

建立接收交换命令的可能的状态字如下所示。

表 12-12 建立接收交换命令的状态字

值	说 明	通 过 电 流
1	交 换 信	是

GFK-0265J

	息被接受。	
-1	本地发送器ID没有设置。	否
-5	交换ID超出范围。	否
-6	交换已经存在。	否
-7	交换太多。	否
-8	变量太多	否

	。	
-9	超过最大变量数。	否
-10	可用资源不足。	否
-12	机架/槽的位置配置不合适，或者不支持EGD。	否
-13	机架/槽无效	否

	或超出范围。	
-14	没有指定状态字地址或状态字地址无效。	否
-16	保留字（3-6）无效。（必须为0。）	否
-22	无效	否

	的地址类型。必须在1-3范围内。	
-24	为目标地址指定的值无效。	否
-101到-134	-101表示字的参数错误。	否
-135到-177	-101表示以	否



	字 开 始 的 指 定 的 变 量 无 效 。	

以下规则适用于接收交换：

- 同一个 PLC 内不允许存在两个交换。如果发送交换在同一个 PLC 内已经存在（静态或动态，接收或发送），此请求将被拒绝。
- 数据接收在交换创建完成之后立刻开始（可能需要几个扫描周期）。

命令 5 和 6 终止一个发送交换/接收交换

当 CPU 转换为 STOP 状态或电流循环完成后，逻辑驱动 EGD 交换将在指定的 CPU 上自动终止。可以使用发送交换或接收交换终止命令来终止这些交换。但是，这些命令不能终止用控制软件为这个 CPU 建立的交换。

终止可能需要几个扫描周期来完成。如果交换能够完成，象命令状态字一样，在完成的时候将返回 1，交换状态字将被设置为 IPLC_EXCHANGE_DELETED（30）。如果交换不能完成，在命令状态字中将返回一个错误状态，交换状态字不会受其影响。这些命令格式如下所示。

表 12-13 终止发送交换的命令格式

状态 参见表 12-15。这个区域应该初始化设置为 0。	地址
终止发送交换命令 总为 5。	地址 + 1
保留 4 个字。必须设置为 0。	地址 + 2 到 地址 + 5
交换 ID 允许范围：32769 到 49151。	地址 + 6
保留 必须设置为 0。	地址 + 7

表 12-14 终止接收交换的命令格式

状态 参见表 12-15。这个区域应该初始化设置为 0。	地址
终止接收交换命令 总为 6。	地址

	+ 1
保留 4 个字。必须设置为 0。	地 址 + 2 到 地 址 + 5
发送 ID 4 个字。必须与创建的 交换 ID 相一致。	地 址 + 6
交换 ID 允许范围：对于逻辑驱 动交换为 32769 到 49151，而对于静态交 换为 1 到 16383。	地 址 + 7
保留 必须设置为 0。	地 址 + 7

以上两种终止命令可能的命令状态如下表所示。

表 12-15 终止交换命令的状态字

值	说明	通 过 电

		流
1	交换按照请求已经删除。	是
-1	本地发送器 ID 没有设置。	否
-5	交换 ID 超出范围。	否
-16	终止发送 ID 保留的 3 个字 (3 到 6) 必须为 0。 终止接收 ID 无效发送器 ID	否
-108		否

命令 7 每个扫描周期刷新发送数据

在默认情况下，发送数据在每个发送周期内只更新一次。每个扫描周期刷新发送数据命令使得逻辑驱动发送交换在 CPU 的每个扫描周期内进行更新，除非为交换指定了发送周期。这个请求的作用不会被取消除非将 CPU 转为 STOP 模式或者 CPU 电流循环完成。静态交换不会受此命令的影响，同时在逻辑驱动交换中静态交换也不能使用这个命令。

交换状态字将在每个 CPU 扫描中进行更新，但是，正在传输的书将按照指定的发送周期进行发送。

必须将本地发送器 ID 设置（通过配置或通过设置本地发送器 ID 命令）为优先于每个扫描周期刷新发送数据命令的中断。每个扫描周期刷新发送数据必须设置为优先于建立逻辑驱动交换（发送或者接收）。

表 12-16 每个扫描周期刷新发送数据命令格式

状态 参见表 12-17。这个区域 必须被初始化设置为 0。	地 址
每个扫描周期刷新发送 数据命令 总为 7。	地 址 + 1
保留 2 个字。必须设置为 0。	地 址 + 2 地 址 + 3

如表 12-17 所示，命令状态字将在每个扫描周期刷新发送数据命令完成之后返回。

表 12-17 每个扫描周期刷新发送数据命令状态

值	说 明	通 过 电

		流
1	成功。	是
-1	本地发送器 I D 没有设置。	否
-23	服务请求 #4 4 通过电流被中断。	否
-25	相关的命令无效。此请	否

GFK-0265J



	求的优先级必须高于任何建立交换的请求。	
-103	参数块保留字 3 不为 0。	否
-104	参数块保留字 4 不为 0	否

	。	

关于逻辑驱动动态 EGD 的附加说明

分配交换 ID

对于本地 PLC 的每个交换，不论是静态的，动态的，接收还是发送，发送器 ID 和交换 ID 必须唯一。

对于逻辑驱动发送交换，本地 PLC 的每个交换必须有唯一的交换 ID。这些交换 ID 的可用范围为 32769 到 49151。建议客户使用这个范围内的一个方便的子集（如 40000-49151）。

对于逻辑驱动接收交换，发送器 ID 和交换 ID 必须与实际发送器的相匹配。使用逻辑驱动动态交换时，交换 ID 必须在 32769 到 49151 范围之内。使用静态交换，交换 ID 必须在 1-16383 范围内。

用户有责任防止发送交换 ID 的频繁重复使用。

重启/重新开始以太网模块的作用

如果以太网接口通过按钮或站管理器重新启动命令重新启动，PLC CPU 将把这个以太网接口相关的逻辑驱动交换的状态字设置为 IPLC_FAILED_ADAPTER (18)，并且终止所有相关的逻辑驱动交换。用户程序应该监控以太网接口的状态字和逻辑驱动交换的交换状态字，在适当的时候重新建立交换。如果系统存在复合以太网接口模块，而且只有一个进行重启，那么只有分配给相应以太网模块的逻辑驱动交换会终止。

交换状态字

交换状态字提供了交换的目前状态。对于静态交换，交换状态字在整个交换过程中进行更新。交换状态字的格式如下所示。由于总是假设其长度为一个字长，因此没有提供长度区域。

NULL 选择器不能用于这种内存类型，由于必须存在交换状态字。

表 12-18 交换状态字地址格式

字	说明
1	PLC 内存类型 可能的内存类型参考表 12-19。
2	PLC 内存偏移地址 关于偏移地址的计算参考表 12-20。基于 0 偏移地址内存类型。
3	保留 必须设置为 0。
4	保留 必须设置为 0。

下表列举了逻辑驱动交换中可以被写入交换状态字的可能值。

表 12-19 交换状态字值

状态助记符	值	
IPLC_NO_NEW_DATA	0	
IPLC_NO_ERROR	1	
IPLC_NEW_DATA	1	
GOOD_DATA_UNSYNC_TIMESTAMP	3	
IPLC_SPEC_MISMATCH	4	
IPLC_REFRESH_ERR	6	
IPLC_DATA_AND_REFRESH_ERROR	7	
IPLC_NO-NETWORK	10	
IPLC_LOCAL_NO_RESOURCE	12	
IPLC_WAITING_NAME_RESOLUTION	16	
IPLC_FAILED_ADAPTER	18	
IPLC_MOD_NOT_CAPABLE	22	
IPLC_CREATE_IN_PROGRESS	24	
IPLC_CREATE_NO_RESPONSE	26	

GFK-0265J

IPLC_CREATE_FAILED	28	
IPLC_EXCHANGE_DELETED	30	

变量地址定义

以下格式用于变量列表的每个变量。

表 12-20 变量格式

字	说明
1	PLC 内存类型 可能的内存类型参考表 12-23。
2	PLC 内存偏移地址 偏移地址的计算参考表 12-23。基于 0 的偏移地址内存类型。
3	保留 必须设置为 0。
4	保留 必须设置为 0。
5	变量长度（字节） 最大 1400 字节。长度计算参考表 12-23。
6	保留 必须设置为 0。

时间标记

下面的格式用于时间标记地址。如果用户不想使用时间标记，可以为内存类型区域指定 NULL 选择器（255）。

表 12-21 时间标记地址

字	说明
1	PLC 内存类型 可能的内存类型参考表 12-23。
2	PLC 内存偏移地址 偏移地址的计算参考表 12-23。基于 0 的偏移内存类型。
3	保留 必须设置为 0。
4	保留 必须设置为 0。

时间标记的长度通常为 4 个字，如下表所示。

表 12-22 POSIX 时钟时间标记格式

字	说明
1	POSIX 时钟 – 秒值(低字节)
2	POSIX 时钟 – 秒值（高字节）
3	POSIX 时钟 – 十亿分之一秒值（低字节）
4	POSIX 时钟 – 十亿分之一秒值（高字节）

交换状态字，变量和时间标记的内存类型

建立交换状态字、变量和时间标记地址的时候可以使用下表。

表 12-23 PLC 内存类型格式

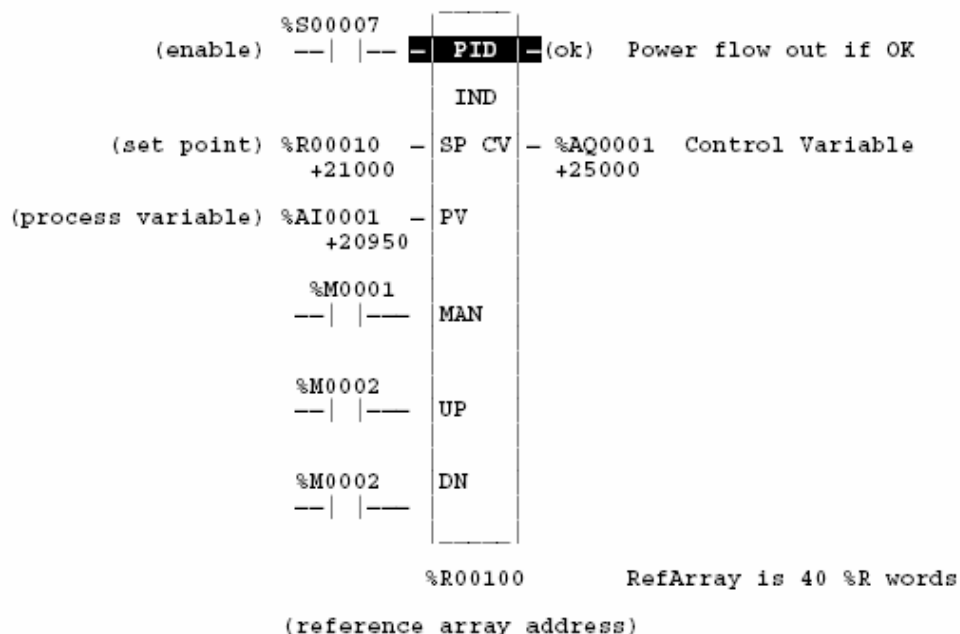
PLC 内存 类型	代 码	交 换 状 态 字	
%R 内 存	8	√	
%AI 内存	10	√	
%AQ 内存	12	√	
%I 内 存	16	√	
%Q 内存	18	√	
%T 内 存	20	√	
%M 内存	22	√	
%SA 内存	24		
%SB 内存	26		
%SC 内存	28	√	
%S 内 存	30		
%G 内存	56		
%I 无 效表	114		
%Q 无效 表	116		
%T 无 效表	118		
%M 无效 表	120		

%SA 无效 表	122		
%SB 无效 表	124		
%SC 无效 表	126		
%S 无 效表	1128		
%G 无效 表	130		
NULL (没 有地 址)	255		

PID

比例、积分和微分（PID）控制功能块通常用来进行闭环过程控制算法。系列 90 PID 功能块根据要求的过程设置点比较过程变量反馈，并且更新基于偏差的控制变量输出。

模块在需要的时间间隔内使用 PID 闭环比例参数和其它保存在 40 个 16 位字（页 12-90 中讨论）的数列来实现 PID 算法。所有的参数是 16 位的整数与 16 位模拟量过程变量兼容。这就允许%AI 内存用于输入过程变量，%AQ 用于输出控制变量。下例显示了典型的输入。



由于设置点和过程变量以及输出控制变量使用频繁，所以简写为 SP，PV 和 CV。由于用 16 个整数进行表示，许多参数在 PV 或 CV 的范围中进行定义。比如，SP 输入必须与 CV 在相同的范围内，因为 PID 功能块通过对这两个书作减法来计算偏差。PV 和 CV 可能为-32000 或者 0 到 32000 来与模拟量量程匹配，或者从 0 到 10000 来显示变量为 0.00%到 100.00%。PC 和 CV 必须有相同的量程，这种情况下，PID 积分中才包含量程因素。

说 明

PID 功能块在 1 毫秒内只能执行一次。因此，如果设置为每个扫描周期执行一次而扫描周期小于 10 毫秒，结果将不同。在这种情况下，PID 功能块在经过的累计时间不到 10 毫秒的情况下将不会运行；比如，扫描时间为 9 毫秒，PID 功能块将在其每次执行完之后 18 秒的另一个扫描周期中执行。

参数表

参数	说明
使能 (enable)	当 enable 为 ON 后，开始执行 PID 运算。
SP	为控制环节的设定值。
PV	为控制环节的过程变量。
MAN	激活后，PID 处于手动模式。
UP	激活时，若在手动模式下，使 CV 值上调。
DN	激活时，若在手动模式下，使 CV 值下降。
地址	PID 功能块所使用的内存地址，由 %R，%P 或 %L 内存中 40 个连续的寄存器组成。 说明：该地址不允许被其它功能使用。 注意：地址重叠时，会导致 PID 不可预测的执行结果。
OK	PID 功能成功执行后，OK 为 ON。
CV	PID 控制变量输出

（UP 或 DN 参数）在 PID 功能每次执行中增或减 1。

有效的内存类型：

- 可使电流通过功能块的变量或地址

PID 参数块:

除了 2 个输入字和 3 个手动控制接点外，PID 功能块在数列中使用参数的其它 13 个参数。这些参数必须在调用 PID 功能块之前设置。PLC 使用其它参数，并且这些参数块是不可配置的。下表所示的%Ref 与 PID 功能块底部 Ref 数列的地址相同。加号后面的数字是数列的偏移地址。比如，Ref 数列以%R100 开始，%R113 包含了手动模式下用来设置控制变量和积分器的手动命令。

表 12-4 PID 参数总览

寄存器	参 数	低位
%Ref+0000	环 号	整数
%Ref+0001	算 法	N/A； PLC 设 置和保存
%Ref+0002	采 样 周 期	10 毫秒

%Ref+0003	死 区 +	PV 值
%Ref+0004	死 区 —	PV 值
%Ref+0005	比 例 增 益	0.01CV%/PV%
%Ref+0006	微 分	0.01 秒
%Ref+0007	积 分 率	重复/1000 秒
%Ref+0008	偏 置	CV 值
%Ref+0009	上 阈 值	CV 值
%Ref+0010	下 阈 值	CV 值
%Ref+0011	最 快 上 升 时 间	秒/全速
%Ref+0012	配 置	使用低 5 位

GFK-0265J

	字	
%Ref+0013	手 动 命 令	CV 值
%Ref+0014	控 制 字	除非 Bit1 进行 设置，由 PLC 保持。
%Ref+0015	内 部 S P	N/A； PLC 设 置和保持。
%Ref+0016	内 部 C V	N/A； PLC 设 置和保持。
%Ref+0017	内 部 P V	N/A； PLC 设 置和保持。

12

%Ref+0018	输 出	N/A； PLC 设 置和保持。

%Ref+0019	微 分 项	N/A； PLC 设置

GFK-0265J

	存 储	和保 持。
%Ref+0020	积 分 项 存 储	N/A; PLC 设置 和保 持。

表 12-4 PID 参数总览（接上页）

%Ref+0021		
%Ref+0022	上升 项存 储	N/A; PLC 设置 和保 持。
%Ref+0023	时钟 （最 后执 行时 间）	N/A; PLC 设置 和保 持。
%Ref+0024		
%Ref+0025		
%Ref+0026	Y 保 持存 储	N/A; PLC 设置 和保 持。
%Ref+0027	SP, PV 的下 限	PV 值

%Ref+0028	SP, PV 的上 限	PV 值
%Ref+0029 到 %Ref+0034	预留 给内 部	N/A
%Ref+0035 到 %Ref+0039	预留 给内 部	N/A

注意每个PID功能块必须使用不同的40个字的数列，即使所有13个用户参数相同，因为内部PID数据数列中其它的字也被使用。确定数列没有超过内存边界。

组态用户参数时，选择PID功能块并按F10键进入用户参数显示界面；然后用箭头键选择字段，键入期望值。用户可用0作为多数默认值，但不包括CV上阈值，PID模块的操作CV上阈值应大于CV下阈值。如果用户参数中有错误的话，PID将无电流通过，所以修改数据时可用一个临时线圈进行监控。

一旦选定合适的 PID 值，就应定义为 BLKMOV 的常数项，以便需要时可以重载缺省 PID 用户参数。

PID 指令的操作

常规自动操作将在每次扫描时调用 PID 模块，有电流时为 Enable，无电流时为 Manual 输入触点。该模块即把当前 PLC 逝去时钟与赏赐贮存在内部 RefArray 的 PID 解答时间进行比较。如果此时间大于 RefArray 的定义在第 3 个字 (%Ref+2) 采样周期的话，PID 运算就用时间差进行解答，并且把上次解答时间和控制变量进行刷新。在 Automatic 方式下，输出 CV 置于手动命令参数 %Ref+13 中。

如果同时为 Enable 和 Manual 输入点送电，则 PID 模块将置于 Manual 方式，输出 CV 从 Manual 命令参数 %Ref+13 中置位。如果 UP 或 DN 中的一个得电，则 Manual 命令字在每次 PID 解答时进行逐一 CV 值的递增或递减。对于输出 CV 的更快的手动修改，直接或从 Manual 命令字进行加/减 CV 值也可以。

PID 模块可用 CV 上阈值和 CV 下阈值参数限制 CV 输出。如果定义了一个正的最小循环时间，则可用来限制 CV 输出的修改速度。如果 CV 范围或速限超出，则积分器中存贮值将进行调用，以确保 CV 值在限制范围内。这种防复位功能意味着即使误差试图使 CV 高于（或低于）阈值一段时间，CV 输出也将在误差范围内进行变化。

这种自动方式下利用 Manual 命令跟踪 CV 和 Manual 方式之间进行的无震动切换。CV 上、下阈值和最小循环时间仍用于手动方式的 CV 输出，而存贮在积分器中的内部值则会被刷新。其意义是如果用户即将进入到手动方式的 Manual 命令，CV 输

出不会改变最小循环时间速率的限制内容，也不会高过或低于 CV 上阈值或下阈值。

注 意

PID 功能在每次扫描中不得调用一次以上。

表 12-4 简单介绍了 PID 功能模块的一些参数，表 12-5 为其参数的详细说明。每个参数名称后括号中的数字即 RefArray 中的偏置量。

表 12-5 PID 参数详细说明

数 据 项	说明
回 路 数 (0 0)	回路数是标明 PID 模块的有效参数。该参数是操作员接口设备所定义的一个无符号整数，可为 PLC 提供一个功用标志。若操作逻辑由 Logic master 90-70 软件监控，则回路数可在模

	块地址下加以显示。
算 法 (0 1)	算法是由 PLC 设置的一个无符号整数，用以识别功能模块正在使用什么算法。ISA 算法帝国难以为算法 1，而独立算法则定义为 2。
采 样 周 期 (0 2)	PID 算法之间的最短时间，以 10 毫秒为增量计时。如，以 10 计为 100ms 采样周期。 (UINT 值范围为最大 65535，代表 10.9 分钟的采样周期。) 如果为 0，则功能块被调用(参考下一节 PID 模块计划表)。如果当前 PLC 逝去时间刚好或小于上次 PID 应答时间加上采样周期的话则 PID 算法进行。切记，90-70 不会使用低于 10 毫秒的应答时间(参考页 12-88 的说明)。所以此扫描会跳过那些比较短的扫描时间。该功能可补偿实际逝去时间为 100us 之内。如果其值设置为 0，则每次功能块使能，否则限制为最小值 10ms。
死 区 (+ /-) (0 3/0 4)	INT 值定义 PV 值的上、下死区限。如果不需要死区，则这些值均应为零。如果 PID 误差 (SP-PV) 或 (PV-SP) 介于死区 (+) 和死区 (-) 之间，则功能块将用零误差解决 PID 计算。如果不是零，则死

	区 (+) 必须大于零或死区 (—) 小于零, 或者是 PID 功能块不发生作用。
比例增益 -kp (0 5)	即 INT 数, 在 ISA 版中称为控制器增益, KC, 用于确定 CV 值的变化。以 0.00%/ %形式显示, 隐含两位有效数字, 如输入 kp450, 则显示为 4.50, 结果是 kp 误差/100 或 450*误差/100 为 PID 输出的精度。kp 为调节 PID 回路时的主要增益。
微分增益 -kd (0 6)	INT 值, 用以确定 CV 值的误差。键入代表 10ms 的低位时间, 则显示为 0.00 秒, 隐含两位有效数字。如, kd 为 120 则显示为 1.20 秒, 其结果是 kd*△误差/△时间或 120*4/3 精度的 PID 输出, 此前提是误差为每 30ms 变化 4 个 PV 值。kd 可用于提高低速回路响应, 但对于 PV 输入噪音很敏感。
积分速率增益 -ki (0 7)	INT 值用以设置积分速率。显示为带 3 位隐有效数字 0.000Repeats/秒。如, ki 输入为 1400 则结果显示为 1400Repeats/秒, 即 ki*误差*dt 或 1400*20*50/1000, 即误差为 20PV 值和 50ms PLC 扫描时间的精度。(采样周期为 0)。ki

	为继 kp 之后的次级增益。
CV 偏置/ 输出 (0 8)	在到达速率和振幅阈值到达之前 CV 计数器上增加整型值。只有使用 Kp 比例增益或 PID 环的前馈控制输出来自其它控制环情况下，使用非零 CV 值。

表 4-5 PID 参数详细说明（续）

数据项	说明
CV 上/ 下 阈值 (0 9/1 0)	CV 计数器中的 INT 值定义了 CV 最高和最低值。这些值是必须的，上阈值必须比下阈值的值大，否则 PID 功能块不能正常工作。这些值通常用来定义基于物理限制 CV 输出界限，也用来为 LM90 中的 CV 标定棒图显示或 ADS PID 显示。当到达 CV 阈值时功能块防复位功能将修改积分器的值。
最小 循环	用一个正的无符号数值来定义 CV 的最小秒数，以将输出定义为 0 到满量程的 100%或者

时间 (11)	32000 CV 计数。这是一个倒计时速率限制，CV 输出的变化按照此速率进行。如果是正值，CV 不能改变超过 32000CV 计数时间 Delta 时间（秒），这是由最小循环时间决定的。比如，如果采样周期为 2.5 秒，最小循环时间为 500 秒，每个 PID 处理期间，CV 的变化不能超过 $32000 \times 2.5 / 500$ 或 160 CV 计数。由于阈值限制，如果 CV 速率超限，调整积分器值的防终止特性将起作用。如果最小循环时间为 0，则没有 CV 速率限制。当调节或整定 PID 环增益时，确定最小循环时间设置为 0。
配置字	配置字低 5 位用来修改 3 个标准 PID 设定。其它位应该设置为 0。设置低位为 1，从标准（SP-PV）到（PV-SP）修改标准 PID 偏差项，保留反馈项的标记。设置秒位到 a 1 来将输出极性取反，将 PID CV 输出置为负值，而不是通常的正值。设置第四位为 1，使用偏差项的常规修改 P V 反馈项，从而修改微分项。配置字低 5 位如下详细定义： Bit 0 = 偏差项。当

	这一位设置为 0，偏差项为 SP-PV。 当这一为设置为 1 时，偏差项为 PV-SP。 Bit 1 = 输出极性。当这一位设置为 0 时，CV 输出表示了 PID 计算的输出。当这一位设置为 1，CV 输出表示了 PID 计算的负输出。 Bit 2 = 作用于 PV 的微分项。当这一位设置为 0，微分项应用于偏差项。当这一位设置为 1 时，微分项作用于 P V。所有保留位应设置为 0。 Bit 3 = 死区项。当这一位设置为 0，没有选择死区项。如果偏差在死区范围之内，则偏差项强制为 0。否则偏差项不受死区的影响。如果死区位设置为 1，则选择了死区项。如果偏差在死区范围之内，偏差强制为 0。但是，如果偏差超过了死区范围，偏差按照死区范围进行减少（偏差=偏差-死区范围）。 Bit 4 = 防复位项。当这一位设置为 0，防复位项使用复位返回计算。当输出超出阈值，将产生阈值输出的值代替累积 Y 保持值(在 12-95 页定义)。当此位设置为 1，在计算开始时，用 Y 项的值来代替累积的 Y 项的值。通过这种方法，预设阈值将一直保持到输出超过阈值范围。 说明：防复位动作位只在 6.01 或更新的 90-70 CPU 中有效。 切记，位的设置是以 2

	的指数进行的。比如， 设置控制字为 0 为默认 的 PID 配置，加 1 将偏 差项从 SP-PV 改为 PV-SP，或者加 2 将输 出极性由 CV=PID 输出 变为 CV=-PID 输出， 或者加 4 将微分项从偏 差速率改变为 PV 改变 速率，等。

表 12-5 PID 参数详细说明（续）

数 据 项	说明
手 动 指 令 （ 13 ）	这是一个整型值，当 PID 功能 块处于自动模式下时，设置为 当前 CV 输出。当功能块切换 到手动模式下时，这个值用来 设置 C V 输出和积分器的内部 值，范围为上下阈值和循环时 间限制。
控 制 字 （ 14 ）	这是一个内部参数，通常以 0 开头。 如果低位设置为 1，P I D 功 能块的这个字和其它内部 S P，P V 和 C V 参数必须用作 远程操作（如下所示）。这将 允许远程操作接口设备（如计 算机）从 P L C 程序外部进行

GFK-0265J

CV (16)	(不可配置由 P L C 保持) 跟踪 C V 输出。
PV (17)	(不可配置由 P L C 保持) 跟踪 P V 输入, 如果 Override = 1, 必须由外部设置。
输出 (18)	(不可配置由 P L C 保持) 这是一个有符号字, 在应用可选倒置之前, 代表了功能块输出。如果没有配置输出倒置, 且控制字中的输出极性位设置为 0, 这个值将等于 CV 输出。如果选择了倒置, 输出极性位设置为 1, 这个值将等于 CV 输出的负值。
差分 项 存 贮 (19)	内部使用, 保存中间值。不要 向此位置写值。
内 部 项 存 贮 (20/ 21)	内部使用, 保存中间值。不要 向此位置写值。
循 环 项 存 贮 (	内部使用, 保存中间值。不要 向此位置写值。

GFK-0265J

22)	
时 钟 (23- 25)	内部消耗时间存储（上一 PID 执行时间）。不要向此位置写值。
Y 余 数 (26)	保留余数作积分器部分换算的 0 稳态误差。
降 低 范 围 上 升 范 围	PV 计数中可选的整型值，用来定义 SP 和 PV 在 Logicmaster Zoom 键水平棒图和 ADS PID 面板显示的最低和最高显示值，。

保留 (29-34) (35-39)	29-34 保留作为内部使用；35-39 保留作为外部使用。保留作为 GE Fanuc 使用，所以不能作为其它用途。

RefArray 内部参数

如前页 12-6 所述，PID 功能块可读取 13 个用户参数，而使用其余的 40 个字 RefArray 用于内部 PID 存贮。一般情况下用户

GFK-0265J

不需进行修改。如果用户在长时间延迟后在 Auto 模式下正调用 PID 模块，可用 SVC-REQ #16 加装当前 PLC 逝去时间时钟到 %Ref+23 去更新上次 PID 处理时间以避免积分器沙锅内发生多级变化。若已设置低位控制字 (%Ref+14) 为 1，随后的 4 位控制字必须设置为控制 PID 模块输入触点，内部 SP 和 PV 必须进行设置，这是因为用户已把 PID 模块控制从梯形逻辑中撤出。内部参数字为：

PID 算法选择 (PID ISA 或 PID IND) 和增益

PID 模块可选择独立项 (PID-IND) 或标准 ISA (PID-ISA) 两种 PID 算法之一进行编程。其间算法差别为如何定义积分和微分增益。为了理解这些区别，需要理解以下内容：

PID 算法选择 (PIDISA 或 PININD) 与增益

PID 功能块可以通过编程选择独立项 (PID_IND) 或 ISA (PID_ISA) 的 PID 算法。两种算法的唯一不同之处在于如何定义积分和微分增益。为了理解这些区别，需要理解如下内容：

两种 PID 类型由 SP-PV 计算偏差项，如果偏差项 (配置字 %Ref+12 的低位) 设置为 1，则变为 PV-SP 的取反模式。如果不想让 CV 输出按照通常的 CV 增加 PV 增加的方式，而是按照 PV 输入相反的方向变化，则需要使用取反模式。

Error = (SP-PV) 或 (PV-SP) 如果配置字的低位设置为 1。

微分项通常基于从上一次 PID 处理开始的偏差项的变化，如果 SP 发生了变化，将会引起输出的很大变化。如果不需要这样，配置字的第三位设置为 1，通过 PV 的变化来计算微分项。dt (或 Delta 时间) 通过将当前 PLC 消耗时间减去此块上一个 PID 处理时间决定的。

dt = 当前 PLC 消耗时间时钟 - 上一 PID 处理 PLC 消耗时间
微分 = (偏差 - 上一偏差) / dt 或 (PV - 上一 PV) / dt 如果配置字的第 3 位设置为 1。

独立项 PID(PID_IND)算法计算输出如下：

GFK-0265J

PID 输出 = $K_p \times \text{偏差} + K_i \times \text{偏差} \times dt + K_d \times \text{微分项} + \text{CV 偏置}$

标准 ISA(PID_ISA)算法有如下不同的格式:

PID 输出 = $K_c \times (\text{偏差} + \text{偏差} \times dt / T_i + T_d \times \text{微分}) + \text{CV 偏置}$

公式中 K_c 是控制器增益, T_i 是积分时间而 T_d 是微分时间。
ISA 的优点在于可以通过调整 K_c 来改变积分和微分项, 比例项为 1, 这使得控制环调整更加容易。如果有 PID 增益形式为 T_i 和 T_d , 使用 $K_p = K_c$ $K_i = K_c / T_i$ $K_d = K_c / T_d$ 对它们进行转换, 用作 PID 用户参数输入。

上面所述的 CV 偏置项是 PID 组成之外的附加项。当只使用比例项 K_p 增益并且想保持 CV 为一个非零值, 这时 PV 等于 SP 且偏差为 0。在这种情况下, 将 CV 偏置设置为需要的 CV 值, 当 PV 在 SP 上时。CV 偏置同样可以用来进行反馈控制, 这时, 另外的 PID 环或控制算法用来调整这个 PID 环的 CV 输出。。

如果使用积分项 K_i 增益, CV 偏置一般设置为 0, 因为积分项作为自动的偏置项。在手动模式下启动, 使用手动命令字

(%Ref+13) 将积分项设置为需要的 CV 值, 然后切换到自动模式。如果 K_i 为 0 时, 这也将起作用, 除非积分器在进入自动模式后, 没有按照偏差进行调整。

下图显示了 PID 算法工作过程

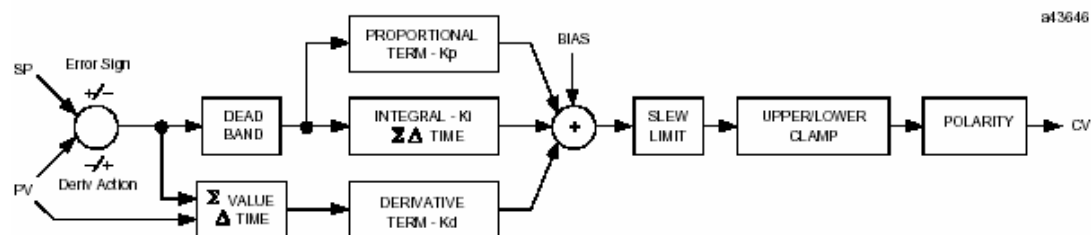


图 12-1 独立项算法 (PIDIND)

ISA 算法 (PIDISA) 也与此类似, 除非 K_p 增益由 K_i 和 K_d 换算出来, 这样积分增益为 $K_p \cdot K_i$ 和微分增益为 $K_p \cdot K_d$ 。偏差标记, 微分项和极性通过用户参数配置字的位进行设置。

CV 幅值和速率极限

该功能块不可直接向 CV 发送计算后的 PID 输出。这两种 PID 运算均可限制输出控制变量的幅值一和变化速率。如果规定值大于 0, 则最大变化速率由 $100\%CV$ (32000) 除以最小循环时间。例如, 如果最小循环时间为 100 秒, 则速率极限为 $320CV$ 值/秒。若 dt 处理时间为 50ms, 则新的 CV 输出不得大于 $320 \times 50/1000$ 或 $16CV$ 值。

CV 输出然后与 CV 上、下阈值进行对比。若有一限值超出, 则 CV 输出置位于该阈值。如果速率或副值超出修改的 CV 值, 则内部积分器值会调整至该限制匹配以避免重置。最后模块会检验输出极性 (配置字 %Ref+12 的第二位) 如果为 1 的话, 则改变输出的极性。

CV=阈值 PID 输出或输出极性位设置时的阈值 PID 输出。

如果模块处于自动模式, 则最终 CV 置于手动请求 %Ref+13。

若模块处于手动模式, 则 PID 等式会跳过而由手动请求置位的 CV, 但所有的速率和幅值极限仍要进行检验。即手动请求不

能大于 CV 上阈值, 并且输出不能如最小循环时间所允许的快速改变。

采样周期和 PID 模块计划表

PID 模块是模拟控制功能块的数字补充, 所以 PID 输出公式中的 dt 采样时间不是模拟控制的无穷小的小值采样时间。多数控制过程可近似为一个带有 1 或 2 级延迟的增益, 可能为一纯时间延迟。PID 模块可把 CV 输出设置给该过程并使用过程反馈 PV 来确定误差以调整下次 CV 输出。某关键过程参数即总的时间常数, 此常数为当 CV 改变时的 PV 快速响应。正如下文中讨论的设置回路增益, 总时间常数 T_p+T_c , 为 CV 跃变时的 PV 达到最终值的 63%所需时间。PID 模块只有在采样时间低于总时间常数一半才能用于控制过程, 较大的采样周期将使其不稳定。

采样周期不应大于总时间常数的 1/10（或特殊情况下的 1/5）。例如，如果 PV 近似为最终值 2 秒的 2/3 时，则采样周期应小于 0.2 秒，特殊情况下为 0.4 秒。另外，采样周期不能太短，如果小于总时间常数的 1/1000，或 $k_i \times \text{误差} \times dt$ 的 PID 积分器四舍五入为 0。例如，一个非常缓慢的过程（需要 10 小时或 36000 秒达到 63%）应该使用 40 秒或更长的扫描时间。

除非过程非常快，一般不必使用 0 采样周期来处理每次 PID 扫描的 PID 运算。如果很多 PID 回路都使用大于扫描时间的采样周期的话，则如果许多回路可在同时结束处理运算时，可以设置较宽的 PLC 扫描时间变化范围。最简单的解决方法是通过一个设置为 0 的位数组进行排列一个或多个位，使电流传至各单个 PID 模块。

确定过程特征

PID 回路增益、 K_p 、 K_i 和 K_d 可由所控制的过程特征确定。其中两个关键问题为：

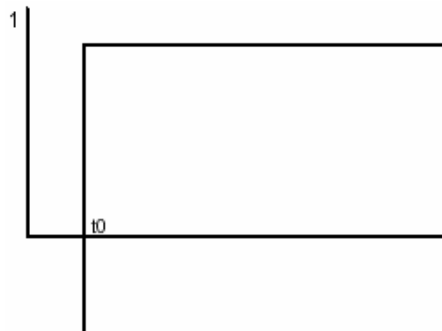
1. 当通过固定量改变 CV 时，PV 改变有多大？开路增益是多少？
2. 系统响应有多快，或者说 CV 输入跃变后 PV 改变有多快？

许多过程可由过程增益，初级或次级顺序延迟及一个纯延迟而模拟。在频率范围内，含一纯时间延迟的初级顺序延迟系统的转换功能如下：

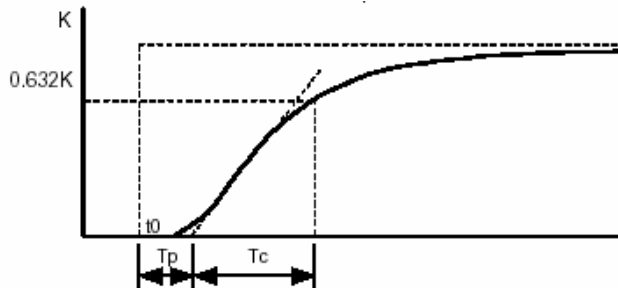
$$PV(s)/CV(s) = G(s) = K \cdot e^{(-T_p s)} / (1 + T_c s)$$

在时间范围内绘出时间 t_0 的阶跃响应曲线，即有开路单位作用曲线：

过程 CV 单位阶跃输出
单位作用曲线输入



来自过程的 PV



由 PV 单位作用作用曲线可得出以下过程模型参数：

k	过程开路增益 = PV 最终改变/ t_0 时的 CV 改变 (注意 k 无脚标)
T_p	t_0 之后但在过程输出 PV 开始传送前的过程时间延迟或死区时间。
T_c	第一顺序过程时间常数， T_p 之后 PV 达到最终 PV 值的 63.2% 时所需时间

测量这些参数的最快捷的方法是把 PID 模块置于手动模式并把 CV 输出作小幅跃变，改变手动请求命令 %Ref+13，并绘出 PV 响应对应时间的曲线。对于慢速过程，该工作可由手工方式完成，但对于快速过程，将借助图形记录仪或计算机图形数据记录包。CV 跃变幅度应足够引起 PV 的可观察的变化，但也不能太大，否则会破坏所测量的过程。最佳幅度为 CV 上、下阈值之差的 2~10%。

设置用户参数

GFK-0265J

由于所有的 PID 参数总是取决于所控制的过程，所以没有现成的预定值，但用迭代法可找到适用的回路增益。

1. 设置所有的用户参数为 0，然后设定 CV 上、下阈值为可能的最高和最低 CV 值。设定采样周期为预定的过程时间常数（上文所述）至 100。
2. 把模块置于手动模式并在不同值下设置手动请求（%Ref+13）以检查 CV 是否可以为上、下阈值。记下 CV 点的 PV 值并装入 SP。
3. 设定较小的增益，如 $100 \times \text{最大 CV} / \text{最大 PV}$ ，中止手动模式。使 SP 值在 PV 范围内的 2~10% 内变化，并观察 PV 响应情况。如 PV 阶跃响应太慢则增大 K_p ，若 PV 突增并且震荡且无一稳态值则减小 K_p 。
4. K_p 找到后，开始增大 K_i 达到突增，该突增应在 2~3 个循环中减弱至一稳态值。这可能需要减少 K_p 。还可以尝试不同的阶跃幅度和 CV 操作点。

5. 找到稳定的 K_p 和 K_i 增益后，尝试增加 K_d 以获得较快的输入改变响应，并不产生震荡。 K_d 通常不是必须的，在 PV 噪音下不能工作。
6. 检查不同 SP 操作点时的增益，必要时加上死区和最小循环时间。有些反向过程还要求设置配置字误差符号或极性位。

设置回路增益——Ziegler 和 Nicholas 调节方法

一旦 K 、 T_p 和 T_c 等 3 个过程模型参数确定后，即可用来评价初始 PID 回路增益。下面的调节方法由 Ziegler 和 Nicholas 在 1940 年代发展起来。幅比即环路响应中第二峰值与第一峰值之比。

1. 计算作用速率：

$$R = K/T_c$$

2. 只对比例控制，计算 K_p

$$K_p = 1 / (R \cdot T_p) = T_c / (K \cdot T_p)$$

3. 对于比例和积分控制

$$K_p = 0.9 / (R \cdot T_p) = 0.9 \cdot T_c / (K \cdot T_p)$$

$$K_i = 0.3 \cdot K_p / T_p$$

4. 对于 PID 控制

$$K_p = G / CR^* \quad G \text{ 为 } 1.2-2.0$$

$$K_i = 0.5 \cdot K_p / T_p$$

$$GFK-0265J$$

$$K_d = 0.5 * K_p * T_p$$

5. 检查采样周期是否在 $(T_p + T_c) / 10$ 至 $(T_p + T_c) / 1000$ 之间。

还有一种“理想调节”方法，可获得最佳的 SP 改变响应，仅由 T_p 过程延迟或死区时间延迟。

$$K_p = 2 * T_c / (3 * K * T_p)$$

$$K_i = T_c$$

$$K_d = K_i / 4 \quad \text{如果使用微分项。}$$

如果确定了初始增益，则必须转换成整型用户参数。为了避免换算问题，过程增益 K 应按输入 PV 值的改变除以 CV 值的输出阶跃改变，不以过程 PV 或 CV 工程单位计量。所有时间以秒计量。 K_p 、 K_i 和 K_d 确定后， K_p 和 K_d 可乘以 100 并按整数输入， K_i 可乘以 1000 并输入到用户参数 %RefArray。

PID 调用示例

下例中 PID 采样周期 100ms， K_p 增益为 4.00， K_i 增益为 1.500。Set Point 存入 %R1，Control Variable 输出存于 %AQ2，而过程变量存入 %AI3。CV 上、下阈值应进行设置，本例中为 20000 和 4000，任选死区为 + 5 和 - 5。40 个字的 RefArray 起始于 %R100。通常用户参数可通过 PID Zoom 键 F10 在 RefArray 中进行设置。但是 %M6 能够通过保存在逻辑中的常数重新初始化从 %R102 (%Ref+2) 开始的 14 个字。这个功能块能够通过 %M1 切换为手动模式，所以手动请求，%R113，能够进行调整。在每个 100 毫秒处理过程中，位 %M4 或 %M5 设置为 1 能够用来增加或减少 %R113 和 PID CV 和积分器的值。对于更快的手动操作，每个 PLC 扫描中，位 %M2 和

%M3 能够用来增加或减少 %R2 的值到 %R113 中，或来自 %R113 的值。当 PID 正常时，%T1 输出置位。

本附录包含指令和90-70系列CPU模板的头定时。这个定时信息可以用来预报CPU扫描时间。

指令定时

90-70系列PLC支持很多不同函数和函数块。表A-1，从下页开始，列出执行时间（ms）和这些CPU支持的函数的内存大小。:

1. 模板 CPM 924/925
2. 模板 CPM 914/915
3. 模板 CPU 781/782
4. 模板 CPU 788/89
5. 模板 CPU 731R和更新的, CPU 732, CPU 771P 和更新的, 和 CPU 772
6. 模板 CPX 935 和 CGR 935
7. 模板 CPX 928
8. 模板 CPX 772 和 CGR 772, 和 CPX 782

注意

本手册印刷时，所有CPU的所有定时信息不可用。附加信息将加到以后版本中。

2个执行时间展示给每个函数:

执行时间	说明
允许	当能流输入或流过函数时，必须的时间，执行函数或函数块。典型地，最好情况时间是，当块使用的数据包含在用户RAM（字定向内存）。
不允许	当其不允许，必须的时间执行函数。

注意

定时器都更新，每次它们在逻辑中达到扫描时间总和。

表 A-1. 指令定时

函数组	函数	允许				不允许				增加				大小
		924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
定时器	ONDT	6.5	10.5	34.5	78	5.5	8	22.5	53	-	-	-	-	18
	OFDT	6	9.5	32.5	68	6	9.5	30	65.5	-	-	-	-	15
	TMR	7.5	10	37	79	8.5	10.5	32	62	-	-	-	-	15
	UPCTR	8.5	11	28.5	67	8	11	29	63	-	-	-	-	18
	DNCTR	6	9.5	28.5	66	6	9.5	28	62.5	-	-	-	-	18
数学公式	ADD (INT)	5	6	22	34.5	4	5.5	14.5	18	-	-	-	-	15
	ADD (DINT)	7	9.5	34.5	63.5	6	6.5	19.5	30.5	-	-	-	-	15
	SUB (INT)	5	6.5	17	35	4	4.5	10.5	19	-	-	-	-	15
	SUB (DINT)	8	10	36	63.5	6	7.5	20	30	-	-	-	-	15
	MUL (INT)	5	6.5	21	38	4	4.5	11.5	18.5	-	-	-	-	15
	MUL (DINT)	8	10.5	35.5	76.5	5.5	6.5	19.5	31	-	-	-	-	15
	DIV (INT)	5.5	7	22.5	41	4.5	4.5	14.5	19	-	-	-	-	15
	DIV (DINT)	8.5	12	41	82	5.5	6.5	18	30.5	-	-	-	-	15
	MOD (INT)	5.5	7.5	23.5	41	4	5	12.5	18.5	-	-	-	-	15
	MOD (DINT)	10.5	14	46.5	85	5.5	7.5	21	31	-	-	-	-	15
	ABS (INT)	5.5	7.5	25.5	50	4.5	5.5	18	25.5	-	-	-	-	12
	ABS (DINT)	7	8.5	28.5	51.5	5	7	18.5	26.5	-	-	-	-	12
	SQRT (INT)	10	14	40.5	82.5	4	4	11	16.5	-	-	-	-	12
	SQRT (DINT)	15.5	27.5	80.5	166	5	6.5	16.5	26.5	-	-	-	-	12
关系	EQ (INT)	6	8	21	37	3.5	4.5	12.5	17.5	-	-	-	-	15
	EQ (DINT)	6.5	9	30	58.5	5	7	18.5	30.5	-	-	-	-	15
	NE (INT)	6	8	20	36.5	3.5	4.5	14	18.5	-	-	-	-	15
	NE (DINT)	6.5	10.5	31	58	5	7	20	30.5	-	-	-	-	15
	GT (INT)	6.5	9	24	43.5	4	4.5	13.5	17.5	-	-	-	-	15
	GT (DINT)	7.5	10.5	28.5	61	5	7.5	18	30.5	-	-	-	-	15
	GE (INT)	6.5	9	25	41.5	3.5	4.5	14	17.5	-	-	-	-	15
	GE (DINT)	6.5	9.5	31.5	58.5	5	7	20	30.5	-	-	-	-	15
	LT (INT)	7.5	9.5	27.5	43.5	3.5	4.5	14.5	17.5	-	-	-	-	15
	LT (DINT)	7.5	10.5	35	61	5	7	20	30.5	-	-	-	-	15
	LE (INT)	6	8.5	20	41.5	3.5	4.5	14	17.5	-	-	-	-	15
	LE (DINT)	6.5	10.5	32	58.5	5	7	21.5	30.5	-	-	-	-	15
	CMP (INT)	10.5	12.5	34.5	62	9	14	33.5	55.5	-	-	-	-	21
	CMP (DINT)	12	14.5	41	79.5	9	12	32	56	-	-	-	-	21
	RANGE (INT)	10.5	16.5	46	82.5	5.5	8	26	33	-	-	-	-	
	RANGE (DINT)	9	14.5	53.5	96.5	8	9	21.5	34	-	-	-	-	
位操作	AND (WORD)	9.5	13.5	53.5	116.5	5	7	28.5	35	0.7	1.5	8.2	18.1	18
	AND (DWORD)	9.5	15	58	123.5	6	7.5	25.5	36	1	2	10.1	24.4	18
	OR (WORD)	8.5	14	57	118	5.5	7	25	35	0.8	1.6	9	19.6	18
	OR (DWORD)	9.5	15	63	126.5	6.5	8.5	24.5	36	1.2	2.3	11.7	27.5	18
	XOR (WORD)	8.5	14	58.5	118	5	7	24.5	35	0.8	1.6	8.9	19.3	18
	XOR (DWORD)	9.5	15	65	124.5	5.5	7.5	26.5	34.5	1.1	2.3	11.6	26.6	18
	NOT (WORD)	7	10.5	42.5	76.5	5.5	6.5	22	32.5	0.3	0.7	3.5	8.4	15
	NOT (DWORD)	7.5	11	41	77.5	5.5	7	25.5	32	0.3	0.7	3.6	10.3	15
	MCMP (WORD)	14	24.5	97.5	213	7	10.5	36.5	56.5	0.6	1.3	3.8	8.4	30
	MCMP (DWORD)	15.5	27	100	221	7.5	11	35	56	1.3	2.7	7.7	16.9	30
	SHL (WORD)	11	17.5	61	137.5	5.5	8.5	25.5	41.5	0.6	1	3.9	10.2	24
	SHL (DWORD)	12	18.5	66.5	147	6	8.5	27	43	1.2	2.1	7.9	20.5	24
	SHR (WORD)	10.5	16.5	60.5	137.5	5	8	26.5	41	0.6	1	3.9	10.1	24
	SHR (DWORD)	12	18	68	147	6	8.5	25	43	1.2	2.1	7.9	20.3	24
	ROL (WORD)	12.5	18	55	117.5	5	7.5	24	33.5	0.6	1.1	4	9.4	18
	ROL (DWORD)	14	20	57.5	127	6	7.5	23.5	35.5	1.3	2.2	8.2	19	18
	ROR (WORD)	8.5	14	51	113	5	7	20.5	33.5	0.6	1.1	4.1	9.4	18
	ROR (DWORD)	10	16	55.5	122	6.5	7.5	21.5	35.5	1.3	2.3	8.2	19	18
	BTST (WORD)	7.5	11	46.5	92	5	7	22.5	35	-	-	-	-	18
	BTST (DWORD)	7	11	43	91	4	6	21.5	34	-	-	-	-	18
	BSET (WORD)	9	11	42.5	89.5	4	5.5	23	32.5	-	-	-	-	15
	BSET (DWORD)	7.5	10.5	40.5	88.5	4	5.5	18.5	31.5	-	-	-	-	15
	BCLR (WORD)	6	10.5	44	89.5	4	5.5	20.5	33.5	-	-	-	-	15
	BCLR (DWORD)	6	10	42	88.5	4	5.5	20	32.5	-	-	-	-	15
	MOVE (BIT)	14	19	58	117.5	3.5	5.5	21	31	0.8	1.2	3.2	6	15
	BPOS (WORD)	10	15	53	108	4	6	22.5	34	0.2	0.4	1.8	4.1	18
	BPOS (DWORD)	13	20.5	81	157.5	4	6	22	35	0.4	0.8	3.1	6.9	18
	SHFR (BIT)	11	20.5	70.5	161.5	4	5.5	12.5	27.5	0.3	0.7	3.1	6.1	24

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPI。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字 的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数块	函数	允许				不允许				增加				大小
		924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
数据移动	MOVE (WORD)	4.5	5.5	14.5	29.5	3.5	3.5	11.5	18.5	0.05	0.1	0.4	0.8	15
	MOVE (DWORD)	7.5	9.5	32.5	56.5	5.5	6.5	18	31	0.2	0.2	1	1.7	15
	BLKMOV (WORD)	5.5	8	21	36	3	3.5	7.5	16.5	-	-	-	-	30
	BLKMOV (DWORD)	7	10	28.5	55	6.5	9	28.5	50.5	-	-	-	-	44
	SWAP (WORD)	5.5	7.5	25	45.5	3	3.5	12	17.5	0.3	0.6	1.7	3.5	15
	SWAP (DWORD)	7	10.5	36.5	65	5	6.5	20.5	31	0.5	0.9	2.7	5.6	15
	BLKCLR	4.5	6	19	30.5	3	3.5	10.5	16.5	0.1	0.2	0.6	1.4	12
	BITSEQ	10	15	46.5	95	7	11.5	38.5	80.5	-	-	-	-	24
	SHFR (WORD)	11	15.5	54	113.5	4	5	11.5	28.5	0.05	0.05	0.3	0.6	24
	SHFR (DWORD)	11.5	16.5	54	117	5	6	14.5	30.5	0.1	0.1	0.8	1.3	24
	SORT	258	423	1403.5	2894	5	6	18.5	28	2.02	3.61	14.43	36.92	15
数据表	TBLRD (INT)	5.5	8.5	27.5	53.5	3.5	4	11	17.5	-	-	-	-	21
	TBLRD (DINT)	7.5	11.5	28.5	58.5	4	4	12.5	18	-	-	-	-	21
	TBLWR (INT)	6	9	30	60.5	3.5	3.5	11.5	17.5	-	-	-	-	21
	TBLWR (DINT)	8.5	14	52	104.5	6	8.5	19.5	37.5	-	-	-	-	21
	FIFORD (INT)	5.5	9	29	58.5	2.5	3	9.5	17	-	-	-	-	21
	FIFORD (DINT)	7	11	31.5	61	3.5	4	12	18	-	-	-	-	21
	FIFOWRT (INT)	5.5	8	25.5	55	3	4	11.5	17.5	-	-	-	-	21
	FIFOWRT (DINT)	7	13	45	85	5	7.5	21.5	37.5	-	-	-	-	21
	LIFORD (INT)	6	9	27.5	52	2.5	3.5	11.5	17	-	-	-	-	21
	LIFORD (DINT)	7.5	12	29.5	57	3.5	4	11.5	18	-	-	-	-	21
	LIFOWRT (INT)	5.5	8	30	55	2.5	3.5	11.5	17	-	-	-	-	21
	LIFOWRT (DINT)	8	14	44.5	85	6	9	23.5	37.5	-	-	-	-	21
	ARRAY_MOVE (BIT)	18.5	27.5	91	197.5	5.5	9	23	41.5	0.01	0.01	0.10	0.30	-
	ARRAY_MOVE (BYTE)	12	20.5	74	163	6	9.5	23	42.5	0.10	0.21	0.50	1.20	-
	ARRAY_MOVE (WORD)	12	20.5	76	164	5.5	9.5	23	41.5	0.03	0.04	0.10	0.30	-
	ARRAY_MOVE (DWORD)	12	20.5	76	165	6.5	9.5	25.5	43	0.09	0.09	0.30	0.60	-
	SRCH (BYTE)	9.5	15	56	119	5.5	8.5	25	40.5	0.03	0.07	0.30	0.60	-
	SRCH (WORD)	9.5	15	57	122.5	5.5	8.5	25.5	41	0.04	0.08	0.30	0.60	-
	SRCH (DWORD)	10	16	60	132	6	8.5	24	41	0.12	0.25	0.80	1.50	-
	ARRAY_RANGE (WORD)	12	16.5	60.5	124	5.5	8	22.5	38	0.24	0.46	2	4.80	-
	ARRAY_RANGE (DWORD)	11.5	16.5	57.5	127.5	7.5	8	20	38	0.25	0.50	2.10	4.80	-
转换	to INT (UINT)	4.5	6.5	21.5	32	3	3.5	10.5	16.5	-	-	-	-	12
	to INT (DINT)	5.5	7	23	40	4.5	5.5	16	26	-	-	-	-	12
	to INT (BCD-4)	5	7	20	38.5	3.5	3.5	9	17	-	-	-	-	12
	to DINT (INT)	6.5	7.5	21	33	4.5	5	11	17.5	-	-	-	-	12
	to DINT (UINT)	5.5	7	23.5	32.5	4	4.5	11	17.5	-	-	-	-	12
	to DINT (BCD-8)	11.5	13	44.5	76.5	5	6	19.5	26.5	-	-	-	-	12
	to UINT (INT)	4.5	6	16.5	32	4	5	7.5	17	-	-	-	-	12
	to UINT (DINT)	5.5	6.5	19.5	39.5	5	5.5	15	26	-	-	-	-	12
	to UINT (BCD-4)	5	7.5	20	38.5	3.5	3.5	7.5	17	-	-	-	-	12
	to BCD-4 (INT)	7	9.5	29	44	3.5	3.5	16	16.5	-	-	-	-	12
	to BCD-4 (UINT)	5.5	7.5	25	43	3	3.5	12	16.5	-	-	-	-	12
	to BCD-8 (DINT)	8.5	13	38.5	77	5	6	18.5	26.5	-	-	-	-	12

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPL。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数组	函数	允许				不允许				增加				大小
		924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
控制	JUMP	2.5	2.5	6.5	13	2	2	2.5	5	-	-	-	-	-
	FOR/NEXT	13.5	19.5	57	124	5.5	8.5	22	52.5	-	-	-	-	-
	MCR/ENDMCR Combined	7	11	28	51	6	9.5	28	46.5	-	-	-	-	9
	DOIO	25.5	40	122	243.5	3.5	4.5	11	20.5	-	-	-	-	15
	DOIO with ALT	32.5	47	140.5	311.5	4	5	11.5	20.5	-	-	-	-	15
	SUSIO	3.5	4.5	12	22	3	4	7	15.5	-	-	-	-	6
	COMMREQ	113	137	350	790	9	11	22.5	38.5	-	-	-	-	18
	CALL/RETURN (LD)	23	31	90.5	229.5	2.5	2.5	5	6	-	-	-	-	15
	CALL/RETURN (SFC)	104.5	143	390	731.5	2.5	2.5	5	6	-	-	-	-	15
	CALL/RETURN (PSB)	16.5	23.5	73.5	145	3	3	3.5	7	-	-	-	-	15
	CALL/RETURN (External Block)	42.5	74	262.5	309.5	2.5	2.5	3	6	-	-	-	-	15
	PIDISA	6.5	11	40	75.5	6	9.5	33.5	65	-	-	-	-	27
	PIDIND	6	11	41.5	76.5	6.5	10	31	66.5	-	-	-	-	27
	VMERD (BYTE)	16	27.5	89.5	173.5	4.5	7.5	20.5	35	0.8	0.8	0.9	1.2	18
	VMERD (WORD)	16	27.5	89.5	171.5	4.5	7.5	19	34	0.8	0.8	0.9	1.2	18
	VMEWRT (BYTE)	19	30	102.5	192.5	5.5	8	20	36.5	0.8	0.8	0.8	0.9	18
	VMEWRT (WORD)	19	30	97.5	188	4.5	7.5	19.5	34	0.8	0.8	0.8	0.9	18
	VMERMW	21	33.5	99.5	191.5	4.5	7.5	20.5	35	-	-	-	-	18
	VMETST	21	28.5	86	155.5	6.5	9.5	25.5	39	-	-	-	-	15
	VME_CFG_RD	31.5	50.5	170.5	345.5	5.5	9.5	26.5	45	-	-	-	-	-
	VME_CFG_WRT	26.5	43	159.5	333.5	5.5	8.5	28	45	-	-	-	-	-
	SVCREQ:													12
	#1	12	23	79	97	4.5	6	14	26	-	-	-	-	12
	#2	12	21	77	86	4.5	6	14	26	-	-	-	-	12
	#3	12	22	76	85	4.5	6	14	26	-	-	-	-	12
	#4	11	21	74	85	4.5	6	14	26	-	-	-	-	12
	#6	13	22	82	93	4.5	6	14	26	-	-	-	-	12
	#7	24	41	152	288	4.5	6	14	26	-	-	-	-	12
	#8	20	37	143	279	4.5	6	14	26	-	-	-	-	12
	#9	23	36	130	202	4.5	6	14	26	-	-	-	-	12
	#10	15	25	90	127	4.5	6	14	26	-	-	-	-	12
	#11	12	22	82	117	4.5	6	14	26	-	-	-	-	12
	#12	11	20	77	82	4.5	6	14	26	-	-	-	-	12
	#13	-	-	-	-	-	-	-	-	-	-	-	-	12
	#14	150	203	643	963	4.5	6	14	26	-	-	-	-	12
	#15	15	25	91	156	4.5	6	14	26	-	-	-	-	12
	#16	19	31	115	148	4.5	6	14	26	-	-	-	-	12
	#17	23	40	94	346	4.5	6	14	26	-	-	-	-	12
	#18	2939	2998	3199	1203	4.5	6	14	26	-	-	-	-	12
	#19	34	52	77	92	4.5	6	14	26	-	-	-	-	12
	#20	26	42	128	248	4.5	6	14	26	-	-	-	-	12
	#21	138	178	586	1558	4.5	6	14	26	-	-	-	-	12
	#22	11	21	77	85	4.5	6	14	26	-	-	-	-	12

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPI。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字 的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数组	函数	允许				不允许				增加				大小
		924/ 925	914 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
浮点	Math:													
	ADD_REAL	8.5	13.5	52.5	108	5.5	6.5	19	31	-	-	-	-	-
	SUB_REAL	8.5	13	51.5	108	5.5	6.5	22.5	31	-	-	-	-	-
	MUL_REAL	8.5	13	51	108	5.5	6.5	21	31	-	-	-	-	-
	DIV_REAL	9.5	14	54	109	5.5	7	22	31	-	-	-	-	-
	ABS_REAL	7	11	42.5	85	5	6.5	17	26.5	-	-	-	-	-
	SQRT_REAL	8	13	43.5	90.5	6	7	14	26.5	-	-	-	-	-
	Trigonometric:													
	SIN	10.5	20.5	63.5	126.5	3.5	6	16	25.5	-	-	-	-	-
	COS	10	20	61.5	121.5	2.5	5.5	14	25.5	-	-	-	-	-
	TAN	9	18	58.5	114.5	3	5.5	14	25	-	-	-	-	-
	ASIN	6.5	13.5	59	132	2.5	5.5	12.5	25.5	-	-	-	-	-
	ACOS	8.5	17.5	69.5	165	2.5	5.5	14.5	25	-	-	-	-	-
	ATAN	10.5	19.5	54	110.5	3.5	5.5	16	25	-	-	-	-	-
	Logarithmic:													
	LOG	6	13	48	100.5	2.5	5.5	13.5	25.5	-	-	-	-	-
	LN	7	13.5	47.5	99.5	2.5	5.5	15.5	25	-	-	-	-	-
	EXPT	10.5	21.5	92	228	3	6	17.5	30	-	-	-	-	-
	EXP	11	22.5	90.5	194	2.5	5.5	14	25.5	-	-	-	-	-
	Comparison:													
	EQ_REAL	7.5	11.5	40	87	5	6	17.5	31	-	-	-	-	-
	NE_REAL	7	10.5	41	85	5	6	20	31	-	-	-	-	-
	GT_REAL	7.5	11.5	46	92	5	6	15	31	-	-	-	-	-
	GE_REAL	7.5	11	46	92	5	6	13.5	30	-	-	-	-	-
	LT_REAL	8	11	43.5	90	5.5	8	19.5	31	-	-	-	-	-
	LE_REAL	7.5	11	45	90	5	6	21.5	31	-	-	-	-	-
	CMP_REAL	17	20	51	114	13	16.5	35.5	64.5	-	-	-	-	-
	Data Move:													
	MOVE_REAL	6	9	29	55.5	3.5	6	14	30	-	-	-	-	-
	Conversion:													
	REAL_TO_INT	7	12.5	43	96.5	4	6	17	26	-	-	-	-	-
	REAL_TO_UINT	8.5	13	46.5	112.5	4	6	16	25.5	-	-	-	-	-
	REAL_TO_DINT	8	13.5	41.5	94	5	7	14.5	25.5	-	-	-	-	-
	INT_TO_REAL	6	10	36.5	68	4	6	14.5	25	-	-	-	-	-
	UINT_TO_REAL	7.5	10.5	30	70.5	4	6	11.5	26	-	-	-	-	-
	DINT_TO_REAL	5.5	9.5	28	65.5	5	5.5	14	25.5	-	-	-	-	-
	REAL_TRUN_INT	8.5	13.5	50	126.5	5	7	15.5	26	-	-	-	-	-
	REAL_TRUN_DINT	8	13.5	51.5	121.5	4	6.5	14.5	26	-	-	-	-	-
	DEG_TO_RAD	9	14.5	43.5	101	5.5	7	12	26	-	-	-	-	-
	RAD_TO_DEG	7.5	14	46	105	4	6	12	26	-	-	-	-	-
	BCD4_TO_REAL	7.5	13	36	92	4	5.5	15	25.5	-	-	-	-	-
	BCD8_TO_REAL	9.5	15	39.5	100	4	6	13.5	26	-	-	-	-	-

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPL。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数组	函数	允许			不允许			增加			大小
		935	928	CPX782 CPX772	935	928	CPX782 CPX772	935	928	CPX782 CPX772	
定时器	ONDIR	5.5	6	23.5	5	5	16.5	-	-	-	18
	OFDT	5.5	5.5	21	5.5	5.5	20	-	-	-	15
计数器	TMR	6	6.5	23.5	5.5	6	20.5	-	-	-	15
	UPCTR	5.5	5.5	19	5	5	18	-	-	-	18
	DNCTR	5.5	5.5	19	5	5	18	-	-	-	18
数学运算	ADD (INT)	4	4	10.5	3.5	3.5	7.5	-	-	-	15
	ADD (DINT)	5.5	6	20	4.5	5	11.5	-	-	-	15
	SUB (INT)	4	4	10.5	3.5	3.5	7.5	-	-	-	15
	SUB (DINT)	5.5	6	20	4.5	4.5	11	-	-	-	15
	MUL (INT)	4	4	11	3.5	3.5	8	-	-	-	15
	MUL (DINT)	6	6	21	4.5	4.5	11.5	-	-	-	15
	DIV (INT)	4.5	4.5	11.5	3.5	3.5	8	-	-	-	15
	DIV (DINT)	6.5	6.5	21	5.5	5.5	11.5	-	-	-	15
	MOD (INT)	4.5	4.5	12	3.5	3.5	8	-	-	-	15
	MOD (DINT)	7	7	25	5	5	12	-	-	-	15
	ABS (INT)	4.5	5	17	3.5	4	9.5	-	-	-	12
	ABS (DINT)	5.5	5.5	17	4.5	4.5	10	-	-	-	12
	SQRT (INT)	6.5	6.5	20	3	3	6.5	-	-	-	12
	SQRT (DINT)	9	9	37.5	4.5	4.5	10	-	-	-	12
关系	EQ (INT)	5.5	5.5	14	3	3.5	7	-	-	-	15
	EQ (DINT)	6	6	19.5	4.5	4.5	11	-	-	-	15
	NE (INT)	5.5	5.5	13	3	3.5	7	-	-	-	15
	NE (DINT)	6	6	19	4.5	4.5	11	-	-	-	15
	GT (INT)	6.5	6.5	16	3	3.5	7	-	-	-	15
	GT (DINT)	7	7	20.5	4.5	4.5	12	-	-	-	15
	GE (INT)	5.5	5.5	15.5	3	3.5	7	-	-	-	15
	GE (DINT)	6	6	19.5	4.5	4.5	11	-	-	-	15
	LT (INT)	6.5	6.5	17	3	3.5	7.5	-	-	-	15
	LT (DINT)	7	7	20.5	4.5	4.5	11.5	-	-	-	15
	LE (INT)	5.5	5.5	15.5	3	3.5	7	-	-	-	15
	LE (DINT)	6	6	20	4.5	4.5	11	-	-	-	15
	CMP (INT)	9.5	9.5	22.5	7.5	8	20.5	-	-	-	21
	CMP (DINT)	10	10.5	27.5	8	8	21	-	-	-	21
	RANGE (INT)	7	7.5	28	4.5	4.5	13	-	-	-	
	RANGE (DINT)	7.5	7.5	30.5	4.5	5	13.5	-	-	-	
位操作	AND (WORD)	6.5	7	32	4.5	4.5	13	0.5	0.5	5.1	18
	AND (DWORD)	7.5	7.5	34	5	5	13.5	0.6	0.7	6	18
	OR (WORD)	6.5	7	32.5	4.5	4.5	13	0.5	0.5	5.4	18
	OR (DWORD)	7.5	8	35.5	5	5	14	0.7	0.8	7.3	18
	XOR (WORD)	6.5	7	32.5	4.5	4.5	13.5	0.5	0.5	5.4	18
	XOR (DWORD)	7.5	7.5	34.5	5	5	13.5	0.7	0.8	7	18
	NOT (WORD)	5.5	5.5	23.5	4	4	12	0.2	0.2	1.9	15
	NOT (DWORD)	6	6	23.5	5	5	12	0.2	0.2	1.6	15
	MCMP (WORD)	9.5	10	58.5	6.5	6.5	18	0.4	0.4	3.3	30
	MCMP (DWORD)	10.5	11	62	7	7	18	0.8	0.9	6.7	30
	SHL (WORD)	9.5	9.5	37.5	4.5	4.5	13.5	0.4	0.5	2.3	24
	SHL (DWORD)	10.5	10.5	40	5.5	5.5	14.5	0.9	1.1	4.8	24
	SHR (WORD)	8.5	8.5	39	4.5	4.5	15	0.4	0.5	2.3	24
	SHR (DWORD)	9.5	9.5	41	5.5	5.5	15.5	0.9	1.1	4.7	24
	ROL (WORD)	7.5	7.5	32	4.5	4.5	12.5	0.4	0.5	2.3	18
	ROL (DWORD)	8	8	34.5	5	5	13	1	1.1	4.8	18
	ROR (WORD)	6.5	6.5	31.5	4.5	4.5	12.5	0.5	0.5	2.5	18
	ROR (DWORD)	8	8	34	5	4.5	14	1	1.1	5.1	18
	BTST (WORD)	6.5	6.5	26	4.5	4.5	12.5	-	-	-	18
	BTST (DWORD)	6.5	6.5	25	4.5	4.5	11.5	-	-	-	18
	BSET (WORD)	5.5	5.5	25	4	4	11	-	-	-	15
	BSET (DWORD)	5.5	5.5	23.5	4.5	4.5	9.5	-	-	-	15
	BCLR (WORD)	5.5	5.5	24.5	4	4	11	-	-	-	15
	BCLR (DWORD)	5.5	5.5	24.5	4	4	11	-	-	-	15
	MOVE (BIT)	11	11	36	4	4	11.5	0.7	0.7	1.8	15
	BPOS (WORD)	6.5	6.5	30	4	4	11	0.1	0.1	1.3	18
	BPOS (DWORD)	8.5	8.5	40.5	4	4	11	0.2	0.2	2.3	18
	SHFR (BIT)	8	8	43	3.5	3.5	8.5	0.2	0.2	1.8	24

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPI。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数组	函数	允许			不允许			增加			大小
		935	928	CPX772 CPX782	935	928	CPX772 CPX782	935	928	CPX772 CPX782	
数据移动	MOVE (WORD)	4	4	9.5	3	3	7	0.1	0.1	0.3	15
	MOVE (DWORD)	5.5	5.5	18.5	4.5	4.5	11	0.2	0.2	0.8	15
	BLKMOV (WORD)	4.5	4.5	13	2.5	2.5	5	-	-	-	30
	BLKMOV (DWORD)	6	6	18	5.5	5.5	19	-	-	-	44
	SWAP (WORD)	4.5	4.5	14	2.5	2.5	6.5	0.2	0.2	1.3	15
	SWAP (DWORD)	6	6	20.5	4.5	4.5	11	0.3	0.3	1.5	15
	BLKCLR	4	4	10.5	2.5	2.5	5.5	0.1	0.2	0.3	12
	BITSEQ	9	9	28.5	6	6	24	-	-	-	24
	SHFR (WORD)	7	7	32.5	3.5	3.5	9	0.1	0.1	0.3	24
	SHFR (DWORD)	8	8	34	4.5	4.5	10	0.2	0.2	0.7	24
	SORT	215.5	260	788	4.5	4	13	1.38	1.53	8.8	15
数据表	TBLRD (INT)	4	4	15.5	2	2	4.5	-	-	-	21
	TBLRD (DINT)	5	5	17	3	3	5.5	-	-	-	21
	TBLWR (INT)	4.5	4.5	17.5	2	2	5	-	-	-	21
	TBLWR (DINT)	6.5	6.5	31.5	4	4	12.5	-	-	-	21
	FIFORD (INT)	4	4	16.5	2	2	5.5	-	-	-	21
	FIFORD (DINT)	5	5	18	3	3	5.5	-	-	-	21
	FIFOWRT (INT)	4	4	15	2	2	5	-	-	-	21
	FIFOWRT (DINT)	5.5	5.5	25.5	4	4	13	-	-	-	21
	LIFORD (INT)	4	4	14	2	2	5	-	-	-	21
	LIFORD (DINT)	5	5	17	3	3	6	-	-	-	21
	LIFOWRT (INT)	4	4	15	2	2	5	-	-	-	21
	LIFOWRT (DINT)	5.5	5.5	26	4.5	4.5	13.5	-	-	-	21
列	ARRAY_MOVE (BIT)	9.5	10	52.5	5	5	14.5	0.01	0.01	0.07	-
	ARRAY_MOVE (BYTE)	8.5	9	44	5	5	15	0.07	0.07	0.40	-
	ARRAY_MOVE (WORD)	8.5	9	45.5	5	5	15.5	0.04	0.06	0.10	-
	ARRAY_MOVE (DWORD)	9.5	9.5	46	5.5	5.5	16	0.09	0.12	0.30	-
	SRCH (BYTE)	8	8	36	5	5	14.5	0.02	0.02	0.10	-
	SRCH (WORD)	8	8	35.5	5	5	14.5	0.02	0.02	0.10	-
	SRCH (DWORD)	8	8.5	38.5	5	5	14.5	0.08	0.08	0.40	-
	ARRAY_RANGE (WORD)	7.5	7.5	38	5	5	14	0.15	0.16	1.30	-
	ARRAY_RANGE (DWORD)	7.5	7.5	37.5	5	5	14	0.16	0.17	1.30	-
变换	to INT (UINT)	4	4	11.5	3	3	5.5	-	-	-	12
	to INT (DINT)	5	5	13.5	4	4	9.5	-	-	-	12
	to INT (BCD-4)	4.5	4.5	12	3	3	6	-	-	-	12
	to DINT (INT)	5	5	12	3.5	3.5	5.5	-	-	-	12
	to DINT (UINT)	5	5	11.5	3.5	3.5	6.5	-	-	-	12
	to DINT (BCD-8)	6.5	6.5	19.5	4.5	4	9.5	-	-	-	12
	to UINT (INT)	4	4	10.5	3	3	6.5	-	-	-	12
	to UINT (DINT)	5	5	13.5	4	4	9	-	-	-	12
	to UINT (BCD-4)	4.5	4.5	12.5	3	3	6.5	-	-	-	12
	to BCD-4 (INT)	4.5	4.5	12	3	3	6	-	-	-	12
	to BCD-4 (UINT)	4.5	4.5	13	3	3	6	-	-	-	12
	to BCD-8 (DINT)	7	7	18	4.5	4.5	9.5	-	-	-	12

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPI。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字 的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数组	函数	允许			不允许			增加			大小
		935	928	CPX772 CPX782	935	928	CPX772 CPX782	935	928	CPX772 CPX782	
控制	JUMP	2.5	2.5	3.5	2	2	3	-	-	-	-
	FOR/NEXT	10.5	11	38	5	5	16.5	-	-	-	-
	MCR/ENDMCR Combined	6	6.5	18.5	5	5.5	17.5	-	-	-	9
	DOIO	13.5	15	81.5	3	3	7.5	-	-	-	15
	DOIO with ALT	14	16	95.5	3	3	7.5	-	-	-	15
	SUSIO	3	3	7.5	2.5	2.5	6	-	-	-	6
	COMMREQ	84.5	90	240.5	8.5	8.5	16	-	-	-	18
	CALL/RETURN (LD)	23	24.5	57.5	2	2	2.5	-	-	-	15
	CALL/RETURN (SFC)	82	88	251	2	2	2.5	-	-	-	15
	CALL/RETURN (PSB)	15.5	17.5	47.5	3	3	3.5	-	-	-	15
	CALL/RETURN (External Block)	27.5	31.5	152.5	2	2	2.5	-	-	-	15
	PIDISA	6	6	22.5	6	6	20.5	-	-	-	27
	PIDIND	6	6	23	6	6	20.5	-	-	-	27
	VMERD (BYTE)	13.5	13.5	53	4	4	13	0.8	0.8	0.9	18
	VMERD (WORD)	13.5	13.5	50.5	4	4	12	0.8	0.8	0.9	18
	VMEWRT (BYTE)	14.5	14.5	55	4	4	13	0.8	0.8	0.8	18
	VMEWRT (WORD)	14.5	14.5	54.5	4	4	12.5	0.8	0.8	0.8	18
	VMERMW	15	15.5	56.5	4	4	13	-	-	-	18
	VMETST	17	17	49	6	6	14.5	-	-	-	15
	VME_CFG_RD	26	26.5	104	4.5	4.5	16.5	-	-	-	-
	VME_CFG_WRT	17.5	18	93	4.5	4.5	16	-	-	-	-
	SVCREQ:										12
	#1	9	9	49	3.5	3.5	10.5	-	-	-	12
	#2	9	9	49	3.5	3.5	10.5	-	-	-	12
	#3	9	9	48	3.5	3.5	10.5	-	-	-	12
	#4	8	9	48	3.5	3.5	10.5	-	-	-	12
	#6	11	11	50	3.5	3.5	10.5	-	-	-	12
	#7	19	21	96	3.5	3.5	10.5	-	-	-	12
	#8	22	25	120	3.5	3.5	10.5	-	-	-	12
	#9	19	19	71	3.5	3.5	10.5	-	-	-	12
	#10	11	11	54	3.5	3.5	10.5	-	-	-	12
	#11	10	10	50	3.5	3.5	10.5	-	-	-	12
	#12	9	9	48	3.5	3.5	10.5	-	-	-	12
	#13	-	-	-	-	-	-	-	-	-	12
	#14	128	159	370	3.5	3.5	10.5	-	-	-	12
	#15	10	10	55	3.5	3.5	10.5	-	-	-	12
	#16	16	17	64	3.5	3.5	10.5	-	-	-	12
	#17	14	15	91	3.5	3.5	10.5	-	-	-	12
	#18	2985	2986	546 (772) 3025 (782)	3.5	3.5	10.5	-	-	-	12
	#19	24	24	112	3.5	3.5	10.5	-	-	-	12
	#20	20	23	99	3.5	3.5	10.5	-	-	-	12
	#21	77	91	381	3.5	3.5	10.5	-	-	-	12
	#22	9	9	47	3.5	3.5	10.5	-	-	-	12

- 注意:
1. 时间 (ms) 基于5.0版Logicmaster 90-70软件。在本手册发行时, 不可用的信息 (用-表示), 参考CPU的IPI。
 2. 关于表函数, 增加是按指定长度单位来的。关于位操作函数, ms/位。关于数据移动函数, ms/位或字的数量。
 3. 允许的时间是单个长单位类型 %R。
 4. COMMREQ 时间已经在CPU和EX7间测量, 不需要等待。
 5. DOIO 是输出值到输出模板的时间。

表 A-1. 指令定时- 续

函数组	函数	允许			不允许			增加			大小
		935	928	CPX772 CPX782	935	928	CPX772 CPX782	935	928	CPX772 CPX782	
浮点	Math:										
	ADD_REAL	7	7	21.5	4.5	4.5	11	-	-	-	-
	SUB_REAL	7	7	21	4.5	4.5	11	-	-	-	-
	MUL_REAL	7	7	21.5	4.5	4.5	11	-	-	-	-
	DIV_REAL	7	7	21.5	4.5	4.5	12	-	-	-	-
	ABS_REAL	6	6	17.5	4.5	4.5	10	-	-	-	-
	SQRT_REAL	6.5	6.5	18	4.5	4.5	10.5	-	-	-	-
	Trigonometric:										
	SIN	8	8	19.5	3	3	9	-	-	-	-
	COS	8	8	20	3	3	9	-	-	-	-
	TAN	7.5	7.5	19.5	3	3	9.5	-	-	-	-
	ASIN	6	6	17.5	3	3	9.5	-	-	-	-
	ACOS	7	7	19	3	3	9.5	-	-	-	-
	ATAN	8	8	20	3	3	9.5	-	-	-	-
	Logarithmic:										
	LOG	5.5	5.5	19	3	3	9	-	-	-	-
	LN	5.5	5.5	18.5	3	3	9	-	-	-	-
	EXPT	8.5	8.5	27	3	3	10.5	-	-	-	-
	EXP	9	9	20	3	3	9	-	-	-	-
	Comparison:										
	EQ_REAL	7	7	21	5	5	12	-	-	-	-
	NE_REAL	7	7	19.5	5	5	11.5	-	-	-	-
	GT_REAL	7	7	20.5	5	5	11.5	-	-	-	-
	GE_REAL	7	7	20.5	5	5	11.5	-	-	-	-
	LT_REAL	7	7	19.5	5	5	11.5	-	-	-	-
	LE_REAL	7	7	19.5	5	5	11	-	-	-	-
	CMP_REAL	16.5	16.5	32	13.5	13.5	25.5	-	-	-	-
	Data Move:										
	MOVE_REAL	4.5	4.5	17.5	3.5	3.5	10	-	-	-	-
	Conversion:										
	REAL_TO_INT	6	6	18	3.5	3.5	9.5	-	-	-	-
	REAL_TO_UINT	6	6	20	3.5	3.5	9.5	-	-	-	-
	REAL_TO_DINT	5.5	5.5	18	3.5	3.5	9.5	-	-	-	-
	INT_TO_REAL	5	5	17	3.5	3.5	9.5	-	-	-	-
	UINT_TO_REAL	5	5	17.5	3.5	3.5	9.5	-	-	-	-
	DINT_TO_REAL	4.5	4.5	16	3.5	3.5	9.5	-	-	-	-
	REAL_TRUN_INT	6	6	20.5	3.5	3.5	10	-	-	-	-
	REAL_TRUN_DINT	6	6	20	3.5	3.5	9	-	-	-	-
	DEG_TO_RAD	5.5	5.5	17.5	3.5	3.5	9.5	-	-	-	-
	RAD_TO_DEG	6	6	18.5	3.5	3.5	9.5	-	-	-	-
	BCD4_TO_REAL	6	6	19.5	3.5	3.5	9.5	-	-	-	-
	BCD8_TO_REAL	6.5	6.5	21.5	3.5	3.5	9	-	-	-	-

- 注意:
1.

时间（ms）基于5.0版Logicmaster 90-70软件。在本手册发行时，不可用的信息 (用-表示), 参考CPU的IPI。
2.

关于表函数，增加是按指定长度单位来的。关于位操作函数，ms/位。关于数据移动函数，ms/位或字的数量。
3.

允许的时间是单个长单位类型 %R。
4.

COMMREQ 时间已经在CPU和EX7间测量，不需要等待。
5.

DOIO 是输出值到输出模板的时间。

头扫描插入时间

附录的这部分包含90-70系列PLC CPU的头定时信息。这些信息可也用来估计逻辑执行时间，预知每个90-70系列CPU的扫描时间。这部分中的信息由基本扫描时间加上扫描插入时间组成，相对应的CPU模板： 731, 732, 771, 772, 781,782, 914, 915, 924,925, 928 和 935。预知扫描时间由加扫描插入时间，基本扫描，和估计的逻辑执行时间计算得到。

本附录的末尾提供预知扫描时间的2个例子，扫描插入时间有4个基本部分组成：

1. 编程器通讯扫描插入
2. I/O 扫描和故障扫描插入
3. 以太网全局数据扫描插入
4. 智能选择模板(PCMs 和LAN 模板) 扫描插入
5. I/O 中断程序和扫描插入

这些部分的每个描述函数，提供每个CPU模板的通讯时间表。

这些表的信息可以用来预知扫描时间，在给定配置情况下。

表包含什么

下表包含扫描90-70系列PLC头函数的插入时间。基本扫描时间是空MAIN程序块执行的时间，没有保存配置和没有窗口激活。剩余的时间值作为扫描插入时间，其是请求函数加到扫描的时间。扫描插入时间是名义上的。

注意

表中列有2类扫描插入数，那些每个扫描周期都插入扫描和那些只有调用时插入扫描。每个扫描周期都插入扫描的函数在列表中粗体列出。

表的一部分，函数同插入扫描一起表示。这意味着，在函数发生时，扫描没有设定阶段。例如，当PLC CPU扫描的输入或输出扫描阶段，发生所有I/O模板的扫描。然而，I/O中断完全与扫描不同步，将中断程序中的任何函数。

通讯函数（除了高优先级的编程器请求外）都在扫描中两个窗口（编程器通讯窗口和系统通讯窗口）的一个中处理。各种服务请求的扫描插入时间是定义函数的所有最小扫描插入时间，窗口时间进行调整，于是在给定的PLC扫描中不会发生窗口限制。这意思是，尽可能，一个窗口（连续逻辑扫描间）中完成每个函数。这些函数的扫描插入可以延伸到多个扫描（受限制的），通过调节窗口时间为比证明的扫描插入时间更小的值。对于编程器，默认时间为10ms;因此，这部分列举的一些函数将自然地限制连续扫描。

基本扫描时间

每个CPU模板的基本扫描时间展示如下。这时间是给空MAIN程序块的，没有连接编程器，没有装载配置，系统中除了CPU没有其它模板。下面图展示了整个扫描阶段和基本扫描阶段，于是扫描的可选部分插入图。

表 A-2. 基本扫描与完整扫描阶段比较

基本扫描	完整扫描
<扫描开始>	<扫描开始>
扫描管理	扫描管理
空输入扫描 *	输入扫描 *
程序逻辑执行	EGD 消耗扫描***
空输出扫描 *	程序逻辑执行
	输出扫描 *
	EGD 产品扫描 ***
	测验丢失I/O模板 **
	编程器通讯窗口
	系统通讯窗口
<扫描结束>	<扫描结束>

* 如果I/O被悬挂，则跳过输入和输出扫描。
** 检测丢失I/O模板，只有如果“Loss of ...”故障已经记录90-70系列I/O模板。
*** 如果没有以太网全局数据 (EGD)交换配置，于是跳过消耗和产品扫描。
对于基本扫描，配置的缺少意味着扫描的输入和输出阶段为空（例如，检查配置和结束）。
没有I/O模板或智能I/O模板（GBC，PSM，等）的配置将有同样的效果。

基本扫描中，逻辑执行时间不为0。执行空MAIN程序的时间包括在内，于是你只需要加实际编程函数的估计的执行时间。基本扫描也假定没有丢失I/O模板。编程器连接的缺少，意思是编程器通讯窗口从没打开。智能选择模板的缺少，意思是系统通讯窗口从没有打开。

下表给出CPU模板的基本扫描时间（ms）。

表 A-3. 基本扫描时间

CPU 模板	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
基本扫描时间	.3731	.4001	.1525	.2074	1.0687	.7025	1.0687	2.00

编程器扫描插入时间

下表展示编程器扫描插入时间（ms）。时间分解为并行和串行，因为这些接口很不同。（表中的每项在表末尾更详细）。

表 A-4. 编程器扫描插入时间*

扫描插入时间	CPX 935	CPX 928	914/ 925	924/ 925	CPX 782	781/782 788/789	CPX 772	731/731 771/772
并行编程器								
编程器窗口	-	-	-	-	-	-	-	0.75
参考表监视器	-	-	-	-	-	-	-	4.90
编辑器监视器	-	-	-	-	-	-	-	4.90
字与字变换	-	-	-	-	-	-	-	28.10
ALT-S 存储	-	-	-	-	-	-	-	14.80
高优先级请求	-	-	-	-	-	-	-	9.90
串行编程器								
编程器窗口	-	-	-	-	-	-	-	0.50
参考表监视器	-	-	-	-	-	-	-	5.60
编辑器监视器	-	-	-	-	-	-	-	6.40

* 上表中需要的时间信息，在本手册发行时，并不是所有的都可用。（少量）

注意

上面表A-4中粗体函数连续插入扫描中。所有其它函数只有在调用时才插入扫描。

表中每项详细说明如下。

扫描插入项	说明
编程器窗口	并行: 打开编程器窗口的时间，但不处理任何请求。编程器并行连接。没有参考值被监视。 串行: 除了是串行连接外，其它与上面一样。不象并行编程器，扫描插入不能限制在单一扫描内。串行通讯，除了在编程通讯窗口内完成处理时间服务外，也通过RS422/484串行连接接收数据处理时间，这个串行通讯处理时间异步。
参考表监视器	扫描插入刷新参考表屏（%R表用来作为示例。）组合表展示微小的大量插入。扫描插入不能连续，依靠PLC的扫描时间和编程软件的主机的速度。
编辑器监视器	当监视梯形图逻辑时，扫描插入刷新编辑器屏。表中给定的时间是一个逻辑屏包含1接点，2线圈和11寄存器的时间。如同参考表扫描插入，插入可能不连续。
字与字变换	扫描插入，改变MOVE_UINT函数的一个常量输入从1到2。这是能做的最小变换。一个线圈的变化要求线圈的更新和保留在PLC内的保持图形。如果函数块的%Q 或 %M 参考地址变化了，于是线圈使用和保持图形的变化尺寸可能更大。如果采用并行编程器（在731CPU上，最坏情况为35ms），大量的字与字变化将要很小的扫描插入。如果采用串行编程器，大量的字与字变化将要很大的扫描插入。时间与要求的字节数成线性，最坏的情况将比并行时最坏情况稍微大一点。
ALT-S 存储	扫描插入存储一个933字节程序块。扫描插入的大小与程序块大小成线性比。
高优先级请求 (只有并行)	有几个编程器高优先级请求：改变PLC模式(包括ALT-R)，读取恒扫描状态/时间，读窗口时间，改变恒扫描状态/时间，改变窗口时间。不象其它编程器服务请求，其在编程器通讯窗口内服务，高优先级请求中断扫描，编程器向它们提问时，异步服务。最坏情况，高优先级请求是监视/改变窗口时间。这个时间如下所示。

I/O 扫描和I/O 故障扫描插入

I/O 扫描插入有2部分，90-70 系列I/O和 Genius I/O。计算I/O扫描插入的公式是：

$$\boxed{\text{I/O 扫描插入}} = \boxed{\text{I/O 扫描头}} + \boxed{90*70\text{系列 I/O 扫描}} + \boxed{\text{Genius I/O扫描}}$$

下表以毫秒方式表示CPU I/O扫描头：

表 A-5. I/O 扫描头 *

CPU 模板	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
I/O 扫描头	.0293	.0321	.1348	1382	.0638	.1756	.0631258	0.38

* 时间是毫秒。当本手册印刷时，信息不可用（用——表示），参考每个CPU的IPI。

注意

I/O 扫描头连续插入扫描

Sweep Impact of Series 90-70 I/O Modules

90-70 系列I/O模板的I/O扫描是如此插入的，通过模板的位置和参考地址。I/O扫描有几个基本部分。

I/O 扫描	说明
I/O 扫描头	包括输入和输出扫描的建立和主机架的选择。
机架建立时间	因为VME总线上扩展机架的寻地址，所以每个扩展机架分别选择。每个扩展机架的固定的头，不管机架内的模板数。
每个模板建立时间	每个90-70系列 I/O模板有一个固定的建立扫描时间。
字节传输时间	主机架中模板的字节传输比扩展机架中的快。字节传输时间不同，因为主机架中使用不同 I/O模板。

另外，模拟输入扩展模板（同Genius块一样）能组成单一传输，连续参考地址用于连续槽地址的模板。连续地址模板的每个顺序称为扫描片段。每个另外扫描片段有一时间处罚。

下面的公式可以用来计算I/O模板扫描插入。计算包含模拟扩展器的时间（或者与先前的模板组到同样扫描片段或组到单独新扫描片段。）扫描插入时间在表A-7中能找到。

表 A-6. 工作表 A: I/O 模板扫描时间

扩展机架数	_____	
每个扩展机架扫描插入	x _____	= _____
主机架离散I/O模板数	_____	
主机架每个离散I/O模板扫描插入	x _____	= _____
扩展机架离散I/O模板数	_____	
扩展机每个离散I/O模板扫描插入	x _____	= _____
主机架模拟输入基础和输出模板数	_____	
主机架每个模拟输入基础和输出模板扫描插入	x _____	= _____
主机架模拟输入扩展模板（同片段）数	_____	
主机架每个模拟输入扩展模板（同片段）扫描插入	x _____	= _____
主机架模拟输入扩展模板（新片段）数	_____	
主机架每个模拟输入扩展模板（新片段）扫描插入	x _____	= _____
扩展机模拟输入基础和输出模板数	_____	
扩展机每个模拟输入基础和输出模板扫描插入	x _____	= _____
扩展机模拟输入扩展模板（同片段）数	_____	
扩展机每个模拟输入扩展模板（同片段）扫描插入	x _____	= _____
扩展机模拟输入扩展模板（新片段）数	_____	
扩展机每个模拟输入扩展模板（新片段）扫描插入	x _____	= _____
预知90-70系列 I/O模板扫描插入	_____	

注意

如果允许点故障，代替允许点故障的相应时间，在下表中表示。

每点或每通道的大概平均值表示如下表。这些平均值基于离散的1024点（512输入和512输出）和模拟128通道（96输入和32输出）。96模拟输入通道包含2个基本模板和5个扩展板。实际值将与大概平均值不同，取决于系统I/O配置。

注意

在本手册印刷时，并不是所有的扫描时间信息都可用（下表中空白处）。参考指定CPU的IPI。

表 A-7. 70模板，I/O模板和机架的扫描插入时间 *

	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
机架建立 每个扩展机 架	.0010	.0010	.0010	.0010	.0010	.0010	.0010	.0500
离散 I/O 模板								
主机架中每个I/O模板	.0074	.0086	.0209	.0105	.0144	.0090	.0144	.0800
主机架中每个I/O模板								
w/点故障允许	.0084	.0125	.0136	.0156	.0220	.0283	.0220	.1200
扩展机架中每个I/O模板	.0119	.0144	.0226	.0110	.0154	.0102	.0154	.0900
扩展机架中每个I/O模板	.0135	.0184	.0398	.0183	.0273	.0149	.0273	.1300
w/点故障允许								
每个故障信息 **	.1314	.1781	.4256	.1570	.3980	.1189	.3980	1.700
每点粗略平均值（无点故障）	-	-	-	-	-	-	-	3.1μs
每点粗略平均值（无点故障）	-	-	-	-	-	-	-	4.4μs

表 A-7. 70模板，I/O模板和机架的扫描插入时间- 续

	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
模拟I/O模板								
主机架中每个输入/输出模板	.0203	.0230	.0279	.0160	.0501	.0149	.0501	0.08
主机架中每个输入/输出模板 W/ 点故障允许	.0259	.0281	.0308	.0223	.0579	.0207	.0579	0.12
扩展机架中每个输入/输出模板	.0313	.0359	.0321	.0318	.0472	.0480	.0472	0.10
扩展机架中每个输入/输出模板 W/ 点故障允许	.0451	.0499	.0464	.0523	.0688	.0761	.0688	0.15
主机架中同片段内每个输入扩展模板	.0166	.0183	.0152	.0169	.0187	.0188	.0187	0.02
主机架中同片段内每个输入扩展模板 W/ 点故障允许	.0249	.0271	.0242	.0259	.0279	.0275	.0279	0.03
主机架中新片段中每个输入扩展模板	.0189	.0199	.0203	.0227	.0267	.0366	.0267	0.03
主机架中新片段内每个输入扩展模板 W/ 点故障允许	.0272	.0308	.0279	.0324	.0395	.0514	.0395	0.05
扩展机架中同片段内每个输入扩展模板	.0537	.0534	.0558	.0567	.0277	.0570	.0277	0.06
扩展机架中同片段内每个输入扩展模板 W/ 点故障允许	.0796	.0822	.0824	.0836	.0558	.0856	.0558	0.10
扩展机架内新片段中每个输入扩展模板	.0557	.0566	.0598	.0653	.0356	.0705	.0356	0.10
扩展机架内新片段中每个输入扩展模板 W/ 点故障允许	.0851	.0861	.0883	.0934	.0675	.1047	.0675	0.15
每个故障信息 **	.1613	.2100	.2597	.3296	.5362	.9362	.5362	2.10
每点粗略平均值（无点故障）	-	-	-	-	-	-	-	10.8μs
每点粗略平均值（无点故障）	-	-	-	-	-	-	-	14.0μs

* 时间为毫秒，除了那些指定为微秒的。
** 离散的90-70系列模板故障总是由PLCCPU获得。当一个发生时，在一个IO扫描段内，其总是登记着。只有当点故障允许时，这些故障才能获得。

注意

表A-7中粗体类函数连续插入扫描中。只有其它函数只有在调用时插入扫描中。也是，在本手册发行时，并不是所有的定时信息都可用。（空白处）

Genius I/O 和 GBC的扫描插入

Genius I/O和 Genius 总线控制器(GBC)的扫描插入同90-70系列I/O类似。有一个能计算的I/O扫描头，在90-70I/O扫描和Genius I/O间。

也有每个Genius总线控制扫描插入，每个扫描片段扫描插入，和一传输时间（每个字）插入所有I/O数据。

扫描插入每个Genius总线控制器有3部分：

1. 扫描插入，打开系统通讯窗口。当第1个智能选择模板（Genius总线控制器是一个）置于系统中时，其只加一次。
2. 扫描插入，允许每个Genius 总线控制器取得背景信息（数据框）。这部分是系统中每个Genius 总线控制器的插入。

注意

第1和第2部分（Genius总线控制器的扫描插入）可以通过关闭系统通讯窗口（设定它的时间为0）来消除。这将只有进行，在处理的鉴定阶段，以减少扫描时间，确保最小扫描时间。输入信息将超时，当窗口关闭时，COMMREQ将停止工作。PCM和LAN模板间通讯将停止。

3. 扫描插入，扫描 Genius总线控制器。本插入导致PLC CPU通报Genius总线控制器，其新输出数据已经传输，命令Genius总线控制读取它的输入数据。在通报Genius 总线控制器时，PLC已经结束另一个扫描，继续运行。

Genius I/O扫描片段包含同一总线（有连续参考地址和连续总线地址）上的Genius 块。处理输入扫描片段的一个扫描片段的时间比输出扫描片段的时间高级。处理的扫描片段同模拟，离散，全局数据扫描片段相同。离散数据一次传输一个字节，比传输模拟数据花更长时间，模拟数据每次传输一个字。全局数据应该计算为离散或模拟的，依据源或目标的参考内存。

注意

在本手册印刷时，并不是所有扫描时间信息都可用（下表中空白处）。参考指定CPU的IPI。

表 A-8. Genius I/O 和 GBC的扫描插入时间*

	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
Genius 总线控制器								
打开系统通讯窗口	.0583	.0631	.0432	.0616	.1625	.1890	.1625	.6000
每个 Genius 总线控制器取的背景信息	.0072	.0083	.0081	.0086	.0160	.0200	.0160	.1000
每个 Genius 总线控制器 I/O 扫描	.5658	.5619	.4237	.6476	.5215	.8110	.5215	.7000
第1 Genius总线控制器 ** 后来的	-	-	-	-	-	-	-	2.4000
Genius 总线控制器	-	-	-	-	-	-	-	1.6000
Genius I/O 块								
每个 I/O 块扫描片段	.0070	.0102	.0270	.0300	.0314	.0444	.0314	.0300
每个 I/O 块扫描片段								
w/点故障允许	.0259	.0266	.0540	.0600	.0584	.0725	.0584	.0900
主机架中每个字节								
离散 I/O 数据	.0451	.0460	.0015	.0017	.0475	.0031	.0475	.0032
扩展机架中每个字节								
离散 I/O 数据	.0468	.0471	.0020	.0025	.0442	.0045	.0442	.0045
主机架中每个字								
模拟 I/O 数据	.0634	.0643	.0011	.0011	.0604	.0042	.0604	.0029
扩展机架中每个字								
模拟 I/O 数据	.0654	.0651	.0040	.0057	.0623	.0131	.0623	.0077
异步事件								
故障信息	-	-	-	-	-	-	-	2.00

*时间为毫秒，除了那些指定为微秒的。

**第1GBC的额外时间与下一表中第1智能选择模板的时间相同。着个时间应该只计算一次。

注意

表A-8中粗体类型的函数连续插入扫描。其他函数只有在调用时插入扫描。

注意

在本手册印刷时，并不是需要所有的定时信息（空白处）。

使用下面公式预算针对Genius I/O的扫描插入。扫描插入时间可以从表A-8中查到。

表 A-9. 工作表 B: Genius I/O 扫描时间

打开系统通讯窗口		=	
GBC I/O 扫描			
GBC 取得背景信息s	+		
GBC数	x		
输入块扫描片段数			
输入块扫描片段扫描插入	x		
输出扫描片段数			
输出块扫描片段扫描插入	x		
主机架GBC上离散I/O数据的字节			
主机架GBC上离散I/O数据的扫描插入/字节	x		
扩展机架GBC上离散I/O数据的字节			
扩展机架GBC上离散I/O数据的扫描插入/字节	x		
主机架GBC上模拟I/O数据的字			
主机架GBC上模拟I/O数据的扫描插入/字	x		
扩展机架GBC上模拟I/O数据的字节			
扩展机架GBC上模拟I/O数据的扫描插入/字	x		
预知 Genius I/O 扫描插入			

FIP I/O 和 FBC的扫描插入

FIP I/O 和 FIP总线控制器（FBC）的扫描插入类似与90-70系列I/O。在90-70系列I/O扫描和FBC I/O扫描间，只有一个I/O扫描头计算一次。也有一个每个FIP总线控制器扫描插入，一个每个扫描片段扫描插入，一个传输时间（每个字）扫描插入所有I/O数据。

扫描插入每个FIP总线控制器有3部分：

1. 扫描插入，打开系统通讯窗口。当第1个智能选择模板（Genius总线控制器是一个）置于系统中时，其只加一次。
2. 扫描插入，允许每个FIP总线控制器取得背景信息（数据框）。这部分是系统中每个FIP总线控制器的插入。

注意

第1和第2部分(FIP总线控制器的扫描插入)可以通过关闭系统通讯窗口(设定它的时间为0)来消除。这将只有进行，在处理的鉴定阶段，以减少扫描时间，确保最小扫描时间。输入信息将超时，当窗口关闭时，COMMREQ将停止工作。PCM和LAN模板间通讯将停止。

3. 扫描插入，扫描FIP总线控制器。本插入导致PLC CPU通报FIP总线控制器，其新输出数据已经传输，命令FIP总线控制读取它的输入数据。在通报FIP总线控制器时，PLC已经结束另一个扫描，继续运行。

FIP I/O扫描片段包含同一总线（有连续参考地址包含多达256字节）上的FIP传输变量（TVA）。处理扫描片段与模拟和离散扫描片段。

注意

在本手册印刷时，一些扫描时间信息不可用（下表中空白处）。

表 A-10. FIP I/O 和 FBC的扫描插入时间 *

	CPX 935 **	CPX 928 **	924/ 925**	914/ 915**	CPX 782**	781/ 782 **	CPX 772**
FIP 总线控制器							
打开系统	.0583	.0631	.0432	.0616	.1625	.1890	.1625
通讯窗口							
每个FIP总线控制器	.0072	.0083	.0081	.0086	.0160	.0200	.0160
取得背景信息	.0839	.1147	—	—	.0931	—	.0931
每个FIP总线控制器I/O 扫描							
第1FIP总线控制器 ***							
后来的FIP总线控制器							
FIP I/O 块							
每个I/O块扫描片段	.0061	.0132	.0114	.0116	.0161	.0025	.0161
每个 I/O 块扫描片段							
w/点故障允许	.0148	.0176	.0119	.0153	.0323	.0670	.0323
主机架中每个字节离散I/O数据	.0018	.0018	.0019	.0024	.0019	.0025	.0019
扩展机架中每个字节离散I/O数据	.0025	.0021	.0024	.0026	.0024	.0044	.0024
主机架中每个字模拟I/O数据	.0012	.0012	.0008	.0011	.0011	.0071	.0011
扩展机架中每个字模拟I/O数据	.0024	.0024	.0043	.0032	.0024	.0098	.0024
异步事件	—	—	—	—	—	—	—
故障信息	—	—	—	—	—	—	—

* 时间为毫秒，除了那些指定为微秒的。

** 有典型扫描插入。每个扫描片段将招致另外的2.1ms，但这不是典型的。当使用异步扫描，设定只包含1FBC，可以避免这个额外的扫描插入。

*** 第1FBC的额外时间与下表中第1智能选择模板的时间相同。这个时间应该只计算一次。

注意

表A-8中粗体类型的函数连续插入扫描。其他函数只有在调用时插入扫描。

也请注意，空白线表示信息在这里不可用。

使用下面公式预算针对FIP I/O的扫描插入。扫描插入时间可以从表A-10中查到。

表 A-11. 工作表B: FIP I/O 扫描时间

打开系统通讯窗口	_____	=	_____
FBC I/O 扫描	_____		
FBC取得背景信息	+ _____	=	_____
FBC数量	x _____	=	_____
输入块扫描片段数	_____		
输入块扫描片段扫描插入	x _____	=	_____
输出块扫描片段数	_____		
输出块扫描片段扫描插入	x _____	=	_____
主机架FBC上离散I/O数据字节	_____		
主机架离散I/O数据的扫描插入/字节	x _____	=	_____
扩展机架FBC上离散I/O数据字节	_____		
扩展机架离散I/O数据的扫描插入/字节	x _____	=	_____
主机架FBC上模拟I/O数据字	_____		
主机架模拟I/O数据的扫描插入/字	x _____	=	_____
扩展机架FBC上模拟I/O数据字	_____		
扩展机架模拟I/O数据的扫描插入/字	x _____	=	_____
预知 FIP I/O 扫描插入	_____		

以太网全局数据扫描插入

依靠CPU扫描时间和一个以太网全局数据（EGD）交换的周期期间的关系，交换数据每个扫描被传输，或在数个扫描后周期性传输。因此，扫描插入将基于交换数量变化，这些交换数安排在扫描时传输。然而，在PLC运行的一些点，所有交换将安排在同一扫描内传输数据。因此，当计算最坏情况扫描插入时，必须考虑所有交换。

以太网全局数据 (EGD) 扫描插入有2部分，消耗扫描和生产扫描：

$$\text{EGD 扫描插入} = \text{消耗扫描} + \text{生产扫描}$$

每个以太网全局数据交换配置为消耗或生产，扫描时间能增加1ms。在配置PLC为恒定扫描模式和设定CPU看门狗超时，这个扫描插入应该考虑进去。

这里，消耗和生产扫描包含2部分，交换头和字节传输时间：

$$\text{扫描时间} = \text{交换头} + \text{字节传输时间}$$

交换头

交换头包括建立每个交换的时间，在扫描时将被传输。当计算扫描插入时，包括每个的头时间。

交换头*

CPU模板		
交换类型	924/925	914/915
消耗	61	106
生产	88	133

* 时间为ms。

字节传输时间

这是在PLC CPU模板和以太网模板间传输数据要求的时间。下表中列出的时间表示传输一个数据字节的时间。

字节传输时间*

CPU模板		
交换类型	924/925	914/915
消耗	2.4	2.4
生产	1.8	1.8

*时间为ms。

表 A-12. 工 作 表：以太网全局数据扫描时间

消耗的交换数	_____		
每个交换扫描插入	x	__	= _____
所有消耗的交换中数据字节数	_____		
每个消耗的数据字节扫描插入	x		= _____
产生的交换数	_____		
每个交换扫描插入	x	__	= _____
所有产生的交换中数据字节数	_____		
每个产生的数据字节扫描插入	x	__	= _____
	预知以太网全局数 据扫描插入		_____

注意

如果PLC设定为Microcycle扫描模式，EGD 扫描插入时间分派给逻辑窗口。
因此，可用来执行用户程序的时间减去以太网全局数据交换要求的时间。

智能选择模板的扫描插入

智能选择模板包括可编程处理器模板（PCM）,数字显示处理器（ADC）,图形显示处理器（GDC），MAP 网络 模板，以太网模板，和Genius 总线控制器用于Genius 网络。这些智能选择模板的扫描插入极不确定。系统通讯窗口的打开和每个模板的读取与CPU内存的读和写要求的扫描插入相比，简直是很小的插入。

下面公式表示怎么计算每个模板的固定扫描。

PCM	=	取得扫描插入 + 时钟刷新插入每 1/2 秒。
ADC	=	取得扫描插入 + 时钟刷新插入每 1/2 秒。
GDC	=	取得扫描插入 + 时钟刷新插入每 1/2 秒。
MAP LAN	=	取得扫描插入+ 网络 I/O 扫描插入
以太网	=	取得扫描插入 + 网络 I/O 扫描插入
GBC	=	取得扫描插入 + GBC I/O 扫描插入
FBC	=	取得扫描插入+ FBC I/O 扫描插入

下表以ms表示智能选择模板固定扫描插入时间。也包含读和写PLC系统内存（包括所有内存，除了%P和%L，其有点慢）扫描插入时间。读和写服务请求有2个边界条件，以改变时间。因此，将时间分解为3部分以反映这些边界条件。

表 A-13. 智能选择模板的固定扫描插入时间 *

扫描插入项	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
智能选择模板	-	-	-	-	-	-	-	-
第1模板	-	-	-	-	-	-	-	0.60
(打开通讯窗口)	-	-	-	-	-	-	-	0.10
每个模板 (读取)	.0751	.0833	.0471	.0586	.1960	.0665	.1960	0.15
网络模板I/O扫描								
PCM, ADC, GDC时钟	.1630	.1434	.1386	.1455	.3651	.3248	.3651	0.60
刷新								
PLC 内存允许读取								
IOM	.4940	.5586	.3990	.4450	1.252	1.206	1.252	3.00
读 / 写 1 到 3 字 *	.6320	.6964	.4830	.5790	1.420	1.393	1.420	3.10
读 / 写 4 到 128 字	.2274	.2495	.6530	.8030	.4140	1.835	.4140	1.00
读/写 每个附加的128字								

* 时间为ms。在本手册印刷时，并不是所有时间信息都需要（空白处）。参考每个CPU的IPI信息。

** 读取只能放3个字到基本信息中。在256字节文本寄存器前需要4个字才能写。

注意

上表中粗体类型函数连续插入扫描。所有其它函数只有在调用时插入扫描。

I/O 中断完成和扫描插入

对于I/O中断块或I/O触发程序，有几个重要性能号。一个I/O中断的扫描插入调用一个空程序或块，测量现场中断的所有时间，开始程序或块，退出程序或块，重新开始中断任务。最大I/O中断速率反应以相同速率调用最小程序或块的I/O中断的限制。执行逻辑的时间包含在中断程序或块中，将影响限制，引起PLC花更多的时间执行I/O中断，减小最大I/O中断速率。

最小的，典型的，和最大的中断响应时间反映一个时间值，从单一I/O模板检测到输入脉冲，直到I/O中断程序或块执行梯形图的第1行或C代码。最小响应时间反映了300ms输入卡滤波时间+I/O中断程序或块内第1梯形逻辑行的运行时间。最小响应时间只有在系统内没有智能选择模板和没有连接编程器时才可以取得。典型响应时间是最小响应时间加上731CPU模板反映的最大中断2ms。这个中断反应时间可以利用，除了当下面操作发生时：

接编程器

逻辑存储，运行模式存储，或字与字变换发生。

一个故障条件（登记一个故障）发生。

另一个I/O中断发生。

CPU正从一个I/O控制器（如Genius总线控制器或FIP总线控制器）传输大量输入（或输出）数据。重负载I/O控制器应该随时放在主机架中。

任何一个这些时间延长中断反应时间（从中断卡发出中断信号给CPU，到CPU执行中断），远离典型值。然而，先前I/O中断的处理期间的中断反应时间很大。I/O中断是连续处理的，因此，一个单一I/O中断的中断反应时间受所有先前中断块的持续时间的影响。（最坏的情况是系统中每个I/O中断同时发生，因此它们中的一个等待所有其它的完成了才开始。）

下面表示的最大响应时间没包含2个极大事件。

表 A-14. I/O 中断块完成和扫描插入时间 *

扫描插入项	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
I/O 中断扫描插入	-	-	-	-	-	-	-	0.95
I/O 中断响应时间								
输入卡滤波时间	-	-	-	-		-	-	0.30
+ 典型中断反应时间	-	-	-	-		-	-	1.40
+ 中断逻辑时间	-	-	-	-		-	-	0.48
最小响应时间	.3851	.3851	.313	-	.5402	.675	.5402	0.78
典型响应时间	.3873	.3873	.316		.5421	.679	.5421	2.18
最大响应时间	.5424	.5424	.627		1.0528	1.633	1.0528	3.60
I/O 中断速率限制								450 ints/sec
I/O 中断速率限制 w/无 IOM								750 ints/sec

* 时间为ms。在本手册印刷时，并不是所有时间信息都需要（空白处）。

表 A-15. I/O-触发中断完成和扫描插入时间 *

扫描插入项	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772
I/O 中断扫描插入							
I/O 中断响应时间							
输入卡滤波时间							
+ 典型中断反应时间							
+ 中断逻辑时间							
最小响应时间	.3046	.3259			.4796		.4796
典型响应时间	.3072	.3273			.4797		.4797
最大响应时间	.4618	.4822			.9921		.9921
I/O 中断速率限制							
I/O 中断速率限制 w/无 IOM							

* 时间为ms。在本手册印刷时，并不是所有时间信息都需要（空白处）。

下面的表格是编程器扫描插入，智能选择模板和I/O中断的扫描插入时间工作表。（参考表 A-4, A-10, A-11。）

注意

在本手册印刷时，并不是所有时间信息都需要（空白处）。参考每个CPU的 IPI信息。

表 A-16. 工 作 表 C: 编程器, IOM, I/O 中断扫描时间

编程器扫描插入	=	_____	
IOM—第1模板（打开通讯窗口）		_____	
IOM—每个模板（取得）	+	_____	
网络模板I/O扫描	+	_____	
总的 IOM 扫描插入	=	_____	
PLC 内存访问IOM	=	_____	
I/O 中断扫描插入		_____	
I/O中断响应时间	+	_____	= _____
预知扫描时间（其它）		_____	

定时中断执行

一个定时的中断的扫描插入调用一个空程序块或定时的程序测量现场中断的全部时间，开始程序或块，退出程序或块，重新启动中断任务。最小的，典型的，和最大的中断响应时间反应一个时间值，从单一I/O模板检测到输入脉冲，直到I/O中断程序或块执行梯形图的第1行或C代码。最小响应时间只有在系统内没有智能选择模板和没有连接编程器时才可以取得。典型响应时间是最小响应时间加上CPU模板反映的最大反应时间。这个中断反应时间可以利用，除了当下面操作发生时：

接编程器

逻辑存储，SLT-S存储，运行模式存储，或字与字变换发生。

一个故障条件（登记一个故障）发生。

另一个I/O中断发生。

任何一个这些事件延长中断反应时间，远离典型值。然而，先前I/O中断的处理期间的中断反应时间很大。对于中断，最坏的情况是系统中每个I/O中断同时发生，因此它们中的一个等待所有其它的完成了才开始。

下面表示的最大响应时间没包含2个极大事件。

表 A-17. 定时的中断执行和扫描插入时间 *

扫描插入项	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
定时的中断扫描插入	-	-	-	-	-	-	-	1.15
定时的中断响应时间	-	-		-	-		-	1.40
典型中断反应时间	-	-		-	-		-	0.68
+ 中断逻辑时间	-		.108	.147	-	.227		0.68
最小响应时间	-		.143	.185		.346		2.08
典型响应时间	-		.163	.219		.464		3.50
最大响应时间								

* 时间为ms。在本手册印刷时，并不是所有时间信息都需要（空白处）。

表 A-18. I/O-触发中断执行和扫描插入时间

扫描插入项	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772
定时的中断扫描插入	-	-	-	-	-	-	-
定时的中断响应时间	-	-	-	-			
典型中断反应时间	-	-	-	-			
+中断逻辑时间							
最小响应时间	-	-	-	-			
典型响应时间	-	-	-	-			
最大响应时间	-	-	-	-			

* 时间为ms。在本手册印刷时，并不是所有时间信息都需要（空白处）。

计算预知扫描时间的例子

下面两个例子说明怎么计算预知扫描时间。第1例子是一个小系统，第2个是一个大系统。这些扫描时间估计值的两个都不包括逻辑执行时间。在这些系统中，计算的扫描为平常扫描时间同点故障不允许，PCM空闲，和不连编程器。计算中使用的时间从表A-3, A-5, A-7, A-8, 和 A-10中取得。计算预知扫描时间的采样格式在后面的例子中提出。

小系统

PS	CPU 731	BTM	32PT Input	32PT Input	32PT Output	32PT Output	8CHN Analog Input	4CHN Analog Output	PCM
0	1	2	3	4	5	6	7	8	9

主机架

扫描计算

预知扫描

=

基本扫描

+

I/O 扫描

PCM 插入

基本扫描时间	=	2.00
I/O 扫描插入 = I/O 扫描头+ 90-70系列 I/O扫描插入 主机架离散I/O模板数		
扫描插入时间每个离散I/O模板	x 4 0.08	= 0.32
主机架类似基础和输出模板数		
扫描插入时间每个类似基础和输出模板	x 2 0.08	= 0.16
I/O 扫描插入 = 0.32 + 16	.48	= 0.48
I/O 扫描头	.38	= 0.38
PCM 插入		
第1模板 (打开通讯窗口)	.60	
每个模板 (读取)	+ .10	= 0.70
预知扫描时间	=	4.04

注意

时间为ms.

大系统

PS	CPU 781	BTM	GBC 20Blks	GBC 20Blks	GBC 16Blks	GBC 16Blks	PCM	PCM	ENET LAN
0	1	2	3	4	5	6	7	8	9

主机架

PS	BRM	32PT Input	32PT Input	32PT Output	32PT Output	8CHN Analog Input	16CHN Analog Expndr	4CHN Analog Output	4CHN Analog Output
0	1	2	3	4	5	6	7	8	9

机架1

PS	BRM	32PT Input	32PT Input	32PT Output	32PT Output	8CHN Analog Input	16CHN Analog Expndr	4CHN Analog Output	4CHN Analog Output
0	1	2	3	4	5	6	7	8	9

机架2

注意

本例子中使用的Genius块配置是一16点组成（QI）块，这些块有相邻的总线参考地址。

预知扫描计算

预知扫描 = 基本扫描+ IOM 插入

基本扫描时间	0.10	=	0.10
I/O 扫描头		0.20	
90-70 I/O 扫描插入 (看表 ____, 工作表 A)		+ 0.80	
Genius I/O 扫描插入(看表 ____, 工作表 B) I/O 扫描头		+ 5.88	= 6.88
PCM 插入		x .30	
PCM数		2	
LAN 插入		0.60	
IOM插入		+ .08	= .68
	预知扫描时间	=	8.66

注意

时间为ms.

表 A-19. 工作表 A

扩展机架数	2	
扫描插入每个扩展机架	x .03	= .06
主机架离散I/O模板数	_____	
扫描插入主机架每个离散I/O模板	x _____	= _____
扩展机架离散I/O模板数	8	
扫描插入扩展机架每个离散I/O模板	x .05	= .40
主机架模拟输入和输出模板数	6	
扫描插入主机架每个模拟输入和输出模板	x .04	= .24
主机架扩展模板（同片段）模拟输入数	_____	
扫描输入主机架扩展模板（同片段）模拟输入	x _____	= _____
主机架扩展模板（新片段）模拟输入数	_____	
扫描输入主机架扩展模板（新片段）模拟输入	x _____	= _____
扩展机架模拟输入基础和输出模板数	_____	
扫描输入扩展机架每个模拟输入基础和输出模板	x _____	= _____
扩展机架模拟输入基础和输出模板（同片段）数	2	
扫描输入扩展机架每个模拟输入基础和输入模板（同片段）	x .05	= .10
扩展机架模拟输入基础和输出模板（新片段）数	_____	
扫描输入扩展机架每个模拟输入基础和输入模板（新片段）	x _____	= _____
预知90-70系列I/O模板扫描时间	.80	

表 A-20. 工作表 B

打开系统通讯窗口	0.30	=	0.30
GBC I/O 扫描			0.94
GBC 取得背景信息	x	0.04	= 0.98
GBC数	x	4	= 3.92
输入块扫描片段数			4
输入块扫描片段——扫描插入	x	0.02	= 0.08
输出块扫描片段数			4
输出块扫描片段——扫描插入	x	0.02	= 0.08
主机架GBC上离散I/O数据字节			288
主机架离散I/O数据的扫描插入/字节	x	.0018	= .5184
扩展机架GBC上离散I/O数据字节		_____	
扩展机架离散I/O数据的扫描插入/字节	x	_____	= _____
主机架GBC上模拟I/O数据字			_____
主机架模拟I/O数据的扫描插入/字	x	_____	= _____
扩展机架GBC上模拟I/O数据字			_____
扩展机架模拟I/O数据的扫描插入/字	x	_____	= _____
预知Genius I/O扫描时间			5.88

表 A-21. 示例工作表 A

扩展机架数	_____	
扫描插入每个扩展机架	x _____	= _____
主机架离散I/O数据字节数	_____	
主机架扫描插入每个离散I/O模板	x _____	= _____
扩展机架离散I/O数据字节数	_____	
扩展机架扫描插入每个离散I/O模板	x _____	= _____
主机架模拟输入基础和输出模板数	_____	
主机架扫描插入每个模拟输入基础和输出模板	x _____	= _____
主机架模拟输入扩展模板（同片段）数	_____	
主机架扫描插入每个模拟输入扩展模板（同片段）	x _____	= _____
主机架模拟输入扩展模板（新片段）数	_____	
主机架扫描插入每个模拟输入扩展模板（新片段）	x _____	= _____
扩展机架模拟输入基础和输出模板数	_____	
扩展机架扫描插入每个模拟输入基础和输出模板	x _____	= _____
扩展机架模拟输入扩展模板（同片段）数	_____	
扩展机架扫描插入每个模拟输入扩展模板（同片段）	x _____	= _____
扩展机架模拟输入扩展模板（新片段）数	_____	
扩展机架扫描插入每个模拟输入扩展模板（新片段）	x _____	= _____
预知I90-70系列I/O模板扫描时间	_____	

表 A-22. 示例工作表 B

编程器扫描插入	= _____	
IOM—第1模板 (打开通讯窗口)	_____	
IOM—每个模板(取得)	+ _____	
LAN 模板 I/O 扫描	+ _____	
总 IOM 扫描插入	= _____	
PLC 内存访问IOM	= _____	
I/O中断扫描插入	_____	
I/O 中断响应时间	+ _____	= _____
预知扫描时间（其它）	_____	

相关CPU性能比较

本部分包含相关的CPU性能比较，基于实验室测试90-70系列CPU的结果。

测试程序

测试程序用来执行测试，包含190K字节文件夹在CPX935 CPU上运行。扫描时间大概是115ms。在这个测试程序中，主块调用同样的子程序（S1）4次。S1子程序，按顺序，调用60个子程序（S2-S61），每个包含同样的逻辑。逻辑包含：

- 140 接点
- 50-70 线圈
- 5 ADD
- 2 DNCTR
- 5 UPCTR
- 10 EQ_INT
- 3 FIFO RD
- 12 MOVE
- 1 MUL_INT
- 1 NE_INT
- 8 ONDTRS
- 1 CALL
- 2 Bit Shift
- 4 SUB_INT
- 4 TMR
- 1 DIVIDE
- 1 PI-ISA

说明图

CPX935运行测试程序的执行时间分配为100%，其它CPU与这个标准进行比较。图说明CPX935有最快时间。CPU782时间占用最长，比CPX935时间长446%。

注意

下图中的数字是由上面测试程序测得。因为不同程序，它们将不会完全相同，但提供一个基本指导，与CPU性能有关。

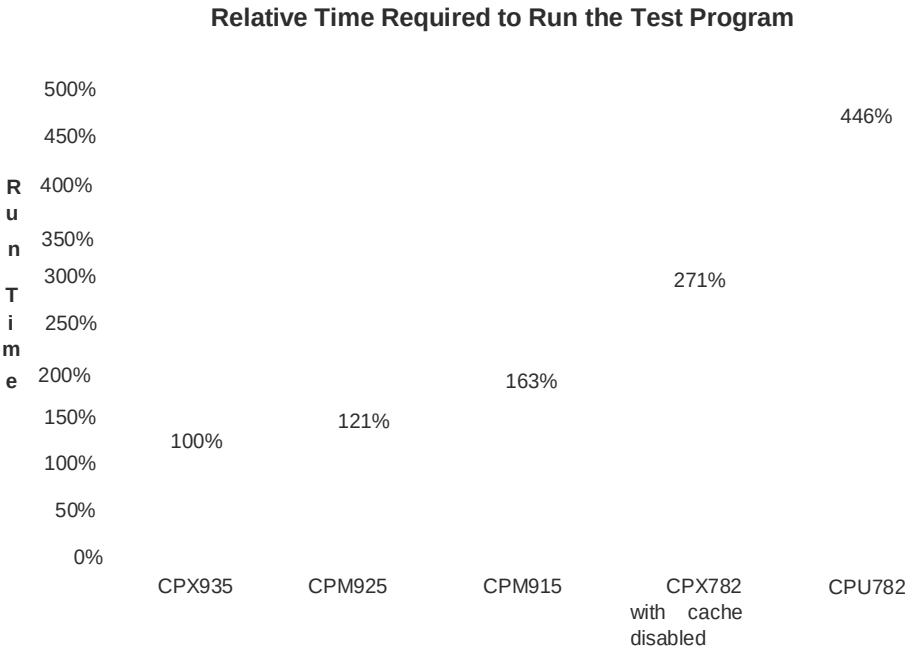
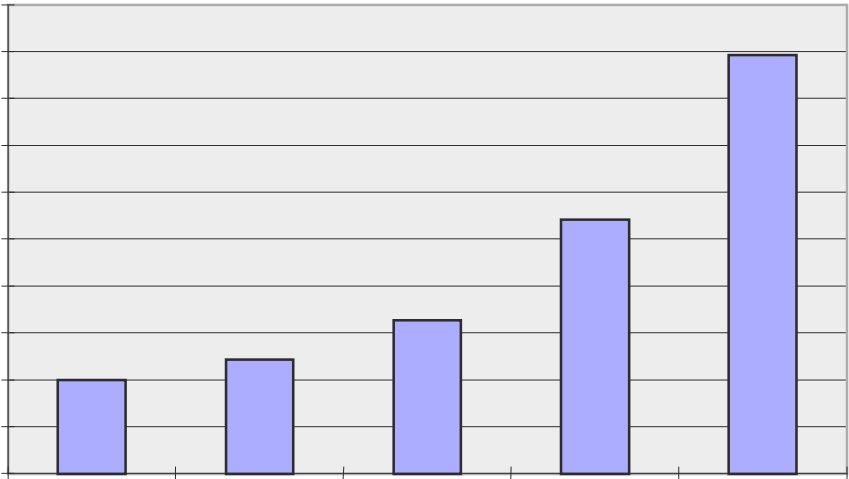




图 A-1. 相关CPU性能图



附录B

Logicmaster 90-70 软件的故障解释

90-70系列PLC包括两个故障表，I/O故障表是I/O设备（包括I/O控制器）发生的故障，和PLC内部故障的PLC故障表。本附录中的信息帮助你在读故障表时了解详细的故障信息。两个故障表都包括同样的信息。

PLC故障表包括：

故障位置

故障描述

故障的日期和时间

I/O 故障表包括

故障位置

回路号

变量地址

故障目录

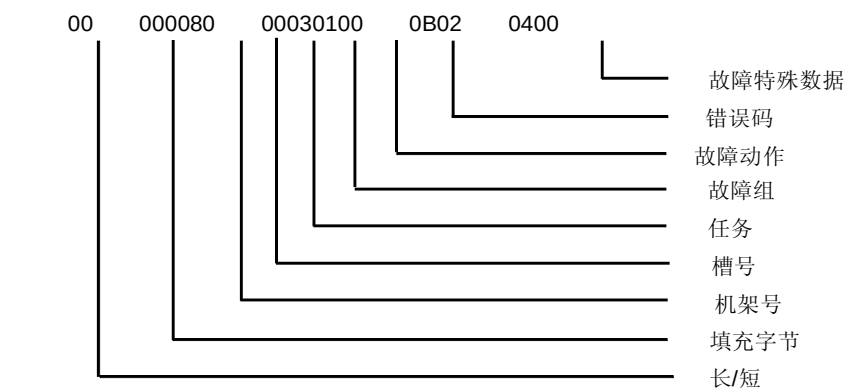
故障类型

故障日期和时间

90-70 系列PLC包括对系统故障诊断有用的每个故障的附加的信息。信息被称为CTRL-F故障，或者故障详情。可以通过选择膏梁的故障信息或者同时按CTRL和F键来查看这些信息。

PLC故障表

下图显示每个故障代号的意义



下图描述故障表每一部分的意义。还包括每部分中值的范围。

长/短指示

此位显示了故障包括8个字节或24个字节的故障数据

类型	代码	故障外部数据
8 位	00	短
24 位	01	长

空字节

这三个字节是填充字节，使PLC的故障表和I/O故障表具有相同的长度

机架

机架号是从0到7的数字。0号是主机架，包括PLC。机架号1到7是扩展机架，通过主机架上的总线通讯模块和扩展机架上的总线接收模块连接。，

槽号

槽号的范围是0到9。PLC的cpu通常在主机架的一号槽上（槽号0）。

任务号

任务号的范围是0到+65535。有时任务号包括PLC任务的附加信息，但是，通常情况下这个数值可以被忽略。

PLC故障组

故障组是最高级别的故障分类。它显示了通常的故障目录。系统软件显示的故障描述文字基于故障组和故障码

表B-1列出了PLC故障表中可能的故障组名。故障组号小于80（十六进制）的是可以屏蔽的故障，故障组号大于等于80（十六进制）的是不可屏蔽故障。

最后的不可屏蔽的故障组，附加的PLC故障码被定义为PLC不需要知道故障码的情况下处理系统中的新的故障条件。所有的未定义PLC类型故障码属于此组。

表 B-1. PLC 故障组

组号		组名	类别
十进制码	十六进制码		
1	1	丢失或者检测不到机架	致命
4	4	丢失或者检测不到模块	诊断性
5	5	额外的或增加的机架	诊断性
8	8	额外的或增加的模块	诊断性
11	B	系配置不匹配	致命
12	C	系统总线故障	诊断性
13	D	PLC CPU硬件故障	致命
14	E	非致命模块硬件故障	诊断性
16	10	可选模块软件故障	诊断性
17	11	程序模块校验值故障	致命
18	12	电池低电压故障	诊断性
19	13	扫描周期超时	诊断性
20	14	PLC 系统故障表满	诊断性
21	15	I/O故障表满	诊断性
22	16	用户程序错	诊断性
—	—	附加 PLC 故障码	
128	80	系统总线故障	致命
129	81	上电时无用户程序	提醒
130	82	检测到计算的用户RAM	致命
131	83	常规扫描周期内窗口程序未完成（就是在指定时间内未完成所有的窗口程序	提醒
132	84	密码错误	提醒
134	86	RUN模式下无系统配置	提醒
135	87	PLC CPU 软件故障	
136	88	系统中发现多于允许的I/O总线控制器数量	致命
137	89	PLC序列存储故障	

* PLC OK 指示灯会闪亮来显示故障并不严重到需要以阻止通讯的方式从故障表信息中读取故障

PLC故障动作

故障动作有三种，故障的动作作为其中的一种

B-2.表 PLC 故障动作

故障动作	相关的CPU 动作	故障码
报告	记录在故障表中.	
报警	记录在故障表中，置位故障变量	2
致命	记录在故障表中，置位故障变量 PLC停止	3

错误码

故障码进一步描述故障。每个故障组有自己的故障码。下面的第一个表显示了PLC软件错误码组(Group 87H)中的错误码。

B-3.表 PLC CPU 软件故障总的故障码

十进制	十六进制	描述		
20	14	PLC 程序内存错误		
39	27	PLC 程序内存错误		
82	52	背板 通 讯 失 败 .		
90	5A	PLC 因为系统请求# 13.停止		
123	7B	远程 I/O 扫描通讯故障		
149	95	上 电 时 存 储 到 flash时 发 生 故 障 注意: 故障额外数据的首字指示了存储失败的原因		
		额外的故障		
		设备不可用	CF	系统中有不可用的特殊设备
		错误的设备数据	CC	设备存储的数据被中断并且不可用,或者闪存未被初始化
		设备RW错误	CB	读写闪存设备时发生错误
		闪存与CPU不匹配	8E	闪存中的数据与PLC中的CPU的固件版本不匹配
All	All	闪存中项的项未被发现	8D	一个或者更多的特定的项未在闪存中发现
All	All	PLC CPU 内部系统错误		

下表显示所有的其它故障组中的故障码

B-4.表 PLC CPU 故障表中的故障码

十进制	十六进制	描述
PLC 选项模块故障码		
3	3	扩展机架发现总线传输模块
22	16	在基本转换模块左面发现模拟量扩展板
25	19	模拟量扩展模块丢失
44	2C	选项模板软复位失败
45	2D	选项模板软复位失败
59	3B	通讯驱动器丢失
60	3C	模块在固件升级模式
65	41	模块在单独运行模式 邮件系统未初始化
75	4B	配置ESCM不兼容—升级端口1和2来配合CPU
76	4C	配置 486不兼容—升级固件使端口1和2相配合
81	51	ESCM 复位请求—端口1和2请求复位。 当端口1和2完成升级后要重新上电一次
255	FF	选项模块通讯故障
增加的机架组的错误码		
1	1	增加的机架
增加的选项模块的故障码		
2	2	模块重启完成
3	3	LAN 介面重启完成，启动应用程序
其它		其它的选项模块故障
LAN模块硬件故障码		
1	1	非致命的 LAN 硬件故障
416	1A0	需 要 的 12V PS 故障或检测不到
450	1C2	LAN 控制器流量过低/过高，重新启动
451	1C3	LAN 介面故障，网络切换到off
452	1C4	LAN 网络问题，性能下降
453	1C5	LAN 服务器网络错误，试图修复
454	1C6	LAN 接收器错误，网络在修复前中断
选项模块软件故障码		
1	1	不支持的模板类型
2	2	COMREQ 邮箱发件箱满，开始COMREQ请求
3	3	COMREQ_ 邮箱满的返回信息
4	4	机架中有超过一个BTM
5	5	背板通讯与PLC，失去请求
10	A	与 LAN交换信息发生错误
11	B	源 (位置, tbl 溢出, etc.) 错误
12	C	VME 背板错误
13	D	用户程序错误
401	191	模块软件中断，需要重新加载
402	192	LAN 系统软件错误，重启
403	193	LAN 系统软件故障，退出配置重启
404	194	LAN 系统软件故障，重启LAN I/F
405	195	LAN 系统软件故障，重新初始化LLC

表 B-4. PLC CPU故障码——接上

十进制	十六进制	描述
<i>系统配置不匹配的故障码</i>		
2	2	智能模块号编号不匹配
4	4	智能模块 I/O 类型不匹配
7	7	子板不匹配
8	8	模拟量扩展板不匹配
9	9	智能模块广播控制数据 (bcd) 长度不匹配
10	A	不支持的特性
11	B	版本 A BTM 不在 Right 槽
14	E	LAN 重复的MAC 地址
15	F	LAN重复的 MAC 地址已解决
16	10	LAN MAC 地址不匹配
17	11	LAN 软开关/解调器不匹配
19	13	智能模块直接控制数据 (DCD) 长度不匹配
23	17	程序超过内存限制
29	1D	不兼容的进度模式
30	1E	变量长度不匹配
31	1F	无效的配置参数
32	20	新的配置, 需要重启
36	24	I/O 规格不匹配
37	25	控制变量超范围
39	27	错误的中断触发
<i>系统总线错误的错误码</i>		
4	4	未定义的 VME 中断错误
其它的		系统总线错误
<i>系统模块校验值错误码</i>		
0	0	90-70系列PLC系统模块头部的校验值错误
1	1	90-70系列PLC 存储的OMF 记录校验值错误
2	2	90-70系列PLC 存储的OMF 记录校验值错误
3	3	系统或者系统模块校验值错误
<i>电池低电压故障码</i>		
0	0	PLC CPU 或其它模块的电池错误
1	1	PLC CPU 或其它模块的电池低电压
2	2	VME背板电池错误
3	3	VME背板电池低电压

表 B-4. PLC CPU故障码—接上 - Continued

十进制	十六进制-	描述
<i>用户故障组的故障码</i>		
1	1	非直接变量地址超范围
2	2	PLC 看门狗计时器时间到
3	3	GBC COMREQ
4	4	GBC 背景信息 – 错误的智能总线请求
5	5	COMREQ – 此命令的等待模式不可用
6	6	COMREQ – 错误的任务 ID
7	7	应用堆栈超范围
8	8	LAN 数据内存耗尽– 检查程序，重新启动
9	9	错误的远程应用程序请求，忽略请求
10	A	错误的本地应用程序请求，忽略请求
11	B	LAN I/F 容量超范围，忽略请求
12	C	LAN 程序/软件不匹配，运行空白程序
13	D	LAN I/F 无法初始化 – 检查程序，运行空白程序
14	E	外部模块上检测凹运行时间错误
15	F	中断中的 SORT 功能未被执行
17	11	单独的运行时间错误
28	1C	程序超过Wind Down时间
29	1D	程序未准备好
<i>系统总线故障组故障码</i>		
1	1	操作系统
<i>上电时中断的用户RAM故障码</i>		
1	1	上电时用户 RAM 错误
2	2	检测到非法布尔操作故障码
5	5	解析 OMF时第二次传递的部分存储故障
6	6	远程 I/O 扫描EEPROM故障; 配置丢失
<i>PLC CPU 硬件故障的故障码</i>		
2169	879	远程 I/O 扫描 Comm 故障，检查总线
2172	87C	远程 I/O 扫描串口总线地址冲突
2048	800	远程I/O 扫描硬件故障
到	到	
4095	FFF	
其它代码		PLC CPU 硬件故障

PLC 故障额外数据

PLC故障表中的故障额外数据段包括故障目录的细节。下面是一些PLC故障表中可能存在的故障的例子:

系统配置不匹配

下面是系统配置不匹配错误码组中的错误码

表 B-5. PLC 故障额外码—系统配置不匹配

故障额外码 位号	模块数字不匹配 (2 位数字)
[0]	FF
[1]	总线地址
[2]	安装模块的型号号
[3]	配置的型号号
故障额外码 位号	I/O 类型不匹配 (4 个数字)
[0]	FF
[1]	总线地址
[2]	安装模块的I/O 类型
[3]	配置模块的I/O类型
故障额外码 位号	BCD长度不匹配 (9个数字)
[0]	FF
[1]	总线地址
[2]	模块的广播数据长度
[3]	配置的模块广播数据长度
故障额外码 位号	不支持的故障 (10 Decimal)
[0]	0A –不支持的 ESCM配置
[1]	01 – 不支持的波特率
[2]	02 –不支持的协议
	00 – 端口 1 (RS-232端口)
	01 – 端口 2 (RS-485端口)
位号	不支持的特性 (10 个数字)
[0]	45 - ASCII 'E' *
[1]	53 - ASCII 'S'
[2]	43 - ASCII 'C'
[3]	4D - ASCII 'M'
[4]	50 - ASCII 'P'
[5]	31 - ASCII '1' Port 1
(RS-232 端口)	
故障额外码 位号	DCD 长度不匹配 (19位)
[0]	FF
[1]	总线地址
[2]	模块的直接数据长度
[3]	配置的模块的直接数据长度

*显示了端口1和端口2在使能状态，但是CPU不支持嵌入的串口模块部件。 在第六位（偏移地址5）的ASCII 值显示了在存储的配置中的哪个端口是使能的。

下表显示了当模块号不匹配时显示的号码

表 B-6. 智能模块故障代号

号码		描述
十进制	十六进制	
4	4	智能网络介面(GENI)
5	5	B 相手持式监视器
6	6	B相6系列带故障诊断智能总线控制器
7	7	B相6系列不带故障诊断智能总线控制器
8	8	PLCM/六系列
9	9	PLCM 90-40系列
10	A	90-70 系列单通道总线控制器
11	B	90-70 系列双通道总线控制器
12	C	90-10 系列智能通讯模块
13	D	90-30 系列智能通讯模块
32	20	高速计数器
69	45	B相 115Vac 8-点 (2 安) 成组模块
70	46	B 相115Vac/125Vdc 8-点隔离模块
70	46	B 相115Vac/125Vdc 8-点隔离模块无开关状态报警
71	47	B 相220Vac 8-点成组模块
72	48	B 相24-48Vdc 16-点接近开关接收模块
72	48	B 相24Vdc 16-点接近开关接收模块
73	49	B 相24-48Vdc 16-点源模块
73	49	B 相24Vdc 16-点接近开关源模块
74	4A	B 相12-24Vdc 32-点接收模块
75	4B	B 相12-24Vdc 32-点源模块
76	4C	B 相12-24Vdc 32-点 5V 逻辑模块
77	4D	B 相115Vac 16-点象限状态输入模块
78	4E	B 相12-24Vdc 16点象限状态输入模块
79	4F	B 相115/230Vac 16-点常开继电器模块
80	50	B 相115/230Vac 16-点常闭继电器模块
81	51	B 相115Vac 16-点AC输入模块
82	52	B 相115Vac 8-点低泄漏成组模块
127	7F	智能网络适配器 (GENA)
131	83	B 相115Vac 4-输入2输出模拟量模块
132	84	B 相24Vdc 4-输入2输出模拟量模块
133	85	B 相220Vac 4输入2输出模拟量模块
134	86	B 相115Vac 热电偶输入模块
135	87	B 相24Vdc 热电偶输入模块
136	88	B 相115Vac RTD 输入模块
137	89	B 相24/48Vdc RTD 输入模块
138	8A	B 相115Vac 张力测量/mV 模拟量输入模块
139	8B	B 相24Vdc张力测量/mV 模拟量输入模块
140	8C	B 相115Vac 4输入2输出电流源模拟量源模块
141	8D	B 相24Vdc 4输入2输出电流源模拟量源模块-

如果模块号是 7FH (智能网络适配器), 模块可能是下表中的一个
(包括GENA 应用 ID 的变量)

Table B-7. GENA 应用 ID 号

号码		描述
十进制	十六进制	
131	83	115Vac/230Vac/125Vdc 电源监视模块 24/48Vdc 电源监视模块 智能远程 90-70 机架控制器
132	84	
160	A0	

如果系统配置不匹配是I/O类型不匹配, 安装的模块I/O类型是下面其中的一种:

表B-8. 智能安装模块I/O 类型

值	描述
01	只有输入
02	只有输出
03	混合型

当系统配置不匹配是I/O类型不匹配时, 配置的模块I/O类型是下面其中的一种 (所有的值都是十六进制)

表 B-9. 智能配置模块 I/O 类型

值		描述
十进制	十六进制	
0	0	数字量输入
1	1	数字量输出
2	2	模拟量输入
3	3	模拟量输出
4	4	成组的数字量
5	5	成组的模拟量
20	14	模拟量输入 数字量输入
21	15	模拟量输入 数字量输出
24	18	模拟量输入, 成组的数字量
30	1E	模拟量输出, 数字量输入
31	1F	模拟量输出, 数字量输出
34	22	模拟量输出, 成组的数字量
50	32	成组的模拟量, 数字量输入
51	33	成组的模拟量, 数字量输出
54	36	成组的模拟量, 成组的数字量

程序模块校验值错误

错误程序模块的名字包含在故障特殊数据字段的头八位中

PLC CPU 硬件故障(RAM 故障)

对于一个PLC CPU 中的RAM故障（故障被报告为PLC CPL硬件的其中一个故障），故障的地址包含在字段的前四个字节中。

应用故障

非直接变量溢出：首俩字节包括调用的偏移值地址。这个故障地址（字段选择和偏移）位于下四个字节中。程序模块的名字在下面的8个字节中

错误的COMREQ 状态指针：第一个字节包括十六进制数FF。后面的四个字节包括状态PLC CPU不能写的指针的字段选择和偏移值

错误的智能总线请求：使用四个字节，除非请求是一个读或写的设备。在这俩种情况下，使用8个字节。

表 B-10. 故障特殊数据- 错误的智能总线请求

故障特殊数据	错误的智能总线请求
[0]	FF
[1]	总线地址请求
[2]	功能码
[3]	子程序码
[4]	字段选择，如果是读/写设备
[5]	偏置的LSB, 如果是读/写设备
[6]	偏置的MSB ，如果是读/写设备
[7]	长度，如果是读/写设备

PLC 故障时刻

六个字节的故障时刻信息是发生故障时PLC CPU记录的系统时钟值。(值已BCD码格式存储。)

表B-11. PLC故障时刻

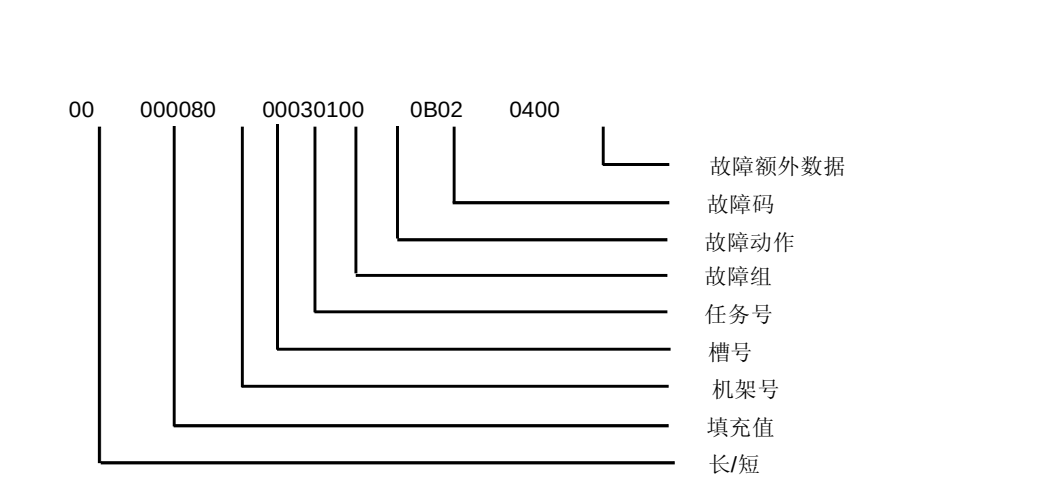
位号	描述
1	秒
2	分
3	小时
4	日
5	月
6	年

智能总线I/O不匹配的例子

智能总线I/O类型不匹配的具体解释的例子如下。(所有的数据是十六进制。)

字段	值	描述
长/短	00	包括8个字节的故障额外数据
机架号	00	主机架（机架号 0)
槽号	03	槽3. 在这个配置中，槽3包括了一个智能总线控制器
任务号	01	单通道总线控制器只有一个任务号
故障组	0B	系统配置不匹配故障
故障动作	02	诊断故障
错误码	04	错误码 04 在系统配置不匹配故障组中表示一个智能模块的I/O类型不匹配
故障额外数据		IO 类型不匹配错误有四位的故障额外数据
[0]	FF	标志位
[1]	16	串行总线地址: 十六进制的16相当于十进制的22。
[2]	01	安装的模块类型。01是输入模块
[3]	04	配置的模块I/O类型。04是混合型模块

存储在系统软件中断配置文件中数据显示串口总线地址22的设备是一个混合类型的模块。但是从智能总线控制器接收到的读ID返回信息表明设备被配置为只是输入的。智能总线控制器当检测到这种不匹配时会记录一个不匹配故障。

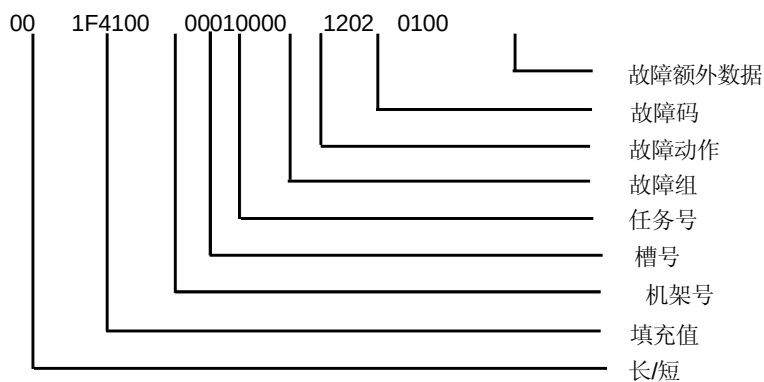


低电池电压信号的例子

低电池电压故障的例子描述如下（所有的数据都是16进制）

字段	值	描述
长/短	00	这个数据包括8个字节的额外数据
机架号	00	主机架 (机架号0).
槽号	01	槽 1. 在所有的配置中, PLC的CPU都安装在机架号0槽号1的位置
任务号	00	PLC CPU.
故障组	12	低电池电压信号
故障组	02	检测到故障.
故障码	01	故障码 01 在低电池电压故障组中是检测到PLC CPU其中的一个 电池低电压
故障额外数据		低电压故障组无故障额外数据.

这个故障发生在PLC CPU检测到其中的一个电池低电压时。

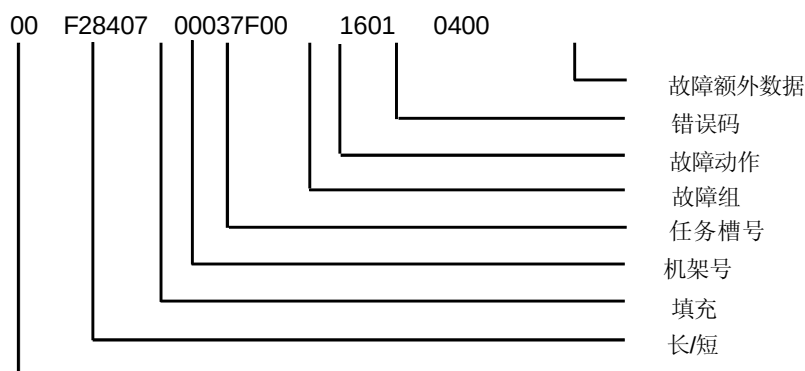


用户应用故障的例子

用户应用故障的例子如下表所列（所有的数据都是十六进制）

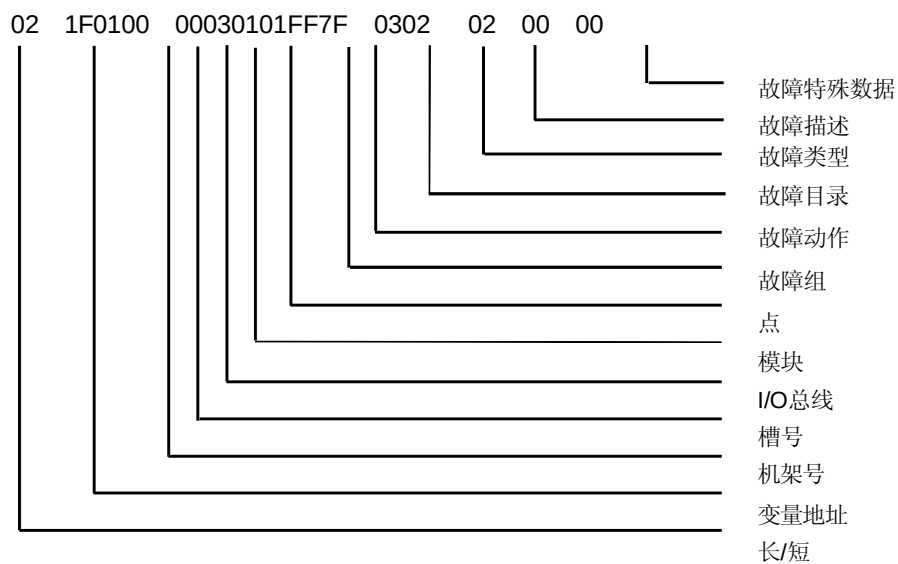
字段	值	描述
长/短	00	故障包括8个字节的额外数据
机架号	00	主机架 (机架号 0)
槽号	03	槽 3. 在这个配置中, 槽 3 包括智能总线控制器
任务号	7F	当GBC记录一个PLC故障列表中的用户应用程序故障时, 会将一个7F放到故障任务号这位。
故障组	16	用户应用程序故障
故障动作	01	信息故障
故障码	04	应用程序故障组中的错误码04表示一个错误的智能总线请求。当智能总线控制器接收到总线上不能完全接收到数据的其它设备一个读或写的数据库的命令时发生此故障。
故障额外数据		错误的智能总线请求错误的的数据有八个字节
[0]	FF	标准位
[1]	1D	串行总线地址: 十六进制的1D表示十进制的29。
[2]	20	数据库中的功能码: 一个 GE Fanuc 的数据库
[3]	1E	数据库中的子程序: 一个读设备。
[4]	08	字段: 08是一个%R内存
[5]	FF	偏置地址的最小的有意义的字节
[6]	FF	偏置地址中最大的有意义的字节
[7]	03	读取的数据的长度: 3个字

智能总线控制器接收到从串口总线地址29来的读入数据, 需要从%R65536开始的%R的三个双字内存。因为这超过了%R的最大极限, 总线控制器会在PLC的故障列表中记录一个错误信息。



I/O 故障表

下表显示了每个故障代码中每一部分的详细信息



下面的章节描述了I/O故障表中每段的意义。包括每段的取值范围。

长/短

此字节显示故障是包括5个字节还是21个字节的故障信息。

B-12表. I/O 故障表格式

类型	对应代码	故障数据长度
短	02	5字节
长	03	21字节

变量地址

变量地址包括一个三个字节的地址，包括I/O内存的地址和与故障点通讯的内存的位置（或者偏移值）。当智能模块故障或者模拟量模块发生故障时，变量地址是模块发生故障的第一个点的地址。

表B-13. I/O 变量地址

字节	描述	范围
0	内存类型	0 - FF
1-2	偏移值	0 - 12K（十进制）

内存类型位是下列值其中的一个。

表B-14. I/O 变量地址内存类型e

名称	值（十六进制）
模拟量输入	0A
模拟量输出	0C
模拟量组	0D
数字量输入	10 or 46
数字量输出	12 or 48
数字量组	1F

I/O 故障地址

I/O故障地址六位的地址包括机架号，槽号，总线号，模块号和发生故障的故障点号。点地址是一个字，其它的地址是一个字节。所有的五个值可能在一个故障中不会同时存在。

当 I/O 故障地址不包括所有的五个地址时，在地址中用一个十六进制的7F来显示重要地址已经结束。例如，如果7F出现在总线位中，就表明故障是一个模块故障。只有机架号和槽号值是重要地址。

机架号

机架号的范围是0到7。0表明是主机架，也就是包含PLC的机架。1到7号是扩展机架，通过主机架上的总线传输模块和扩展机架上的总线接收模块连接到PLC。

槽号

槽号的范围是0到9。PLC CPU总是在主机架的第一槽上（机架号0）。

I/O 总线

I/O总线号的范围是0到15。当槽中的模块是单通道智能总线控制器时，这位总是1。当模块是模拟量模块时，这位标明哪个扩展机架发生了错误。

模块号

模块号标示发生故障的智能总线模块串口总线地址。

点号

点号的范围是1到1024（十进制）。当故障是点类型的故障时，它标明模块的哪个点发生故障。

I/O 故障组

故障组是故障中最高的级别分类。它标明了故障的通常的目录。软件显示的故障描述的文字部分是基于故障组和错误码。表B1-15列出了I/O故障表中可能的故障组。组号小于80（十进制）的是可屏蔽的故障。

最后的不能屏蔽的故障组，附加的I/O故障码，被定义为PLC 不必知道故障码的新的故障条件。所有的认不出的I/O类型的故障码属于此组。

表 B-15. I/O 故障组

组	组名	故障类型
2	丢失的, IOC	致命
3	丢失的I/O 模块	诊断性
6	额外的 IOC	诊断性
7	额外的 I/O模块	诊断性
9	IOC 或 I/O总线故障	诊断性
A	I/O 模块故障	诊断性
F	IOC 软件故障	致命
—	附加的 I/O 故障码	具体见下面描述

I/O 故障动作

故障动作标明PLC CPU在发生故障时应该采用的动作。下表的是可能的故障动作。

表 B-16. PLC 故障动作

故障动作	CPU 的故障动作	代码
提示型	记录在故障列表中	1
诊断性	记录在故障列表中 置位故障变量	2
致命	记录在故障列表中e 置位故障变量 置于停止模式	3

I/O 故障目录

I/O故障目录标明了故障的通常的分类。与 I/O 故障组相似，讨论的更早些。 I/O 故障目录的描述见下

表 B-17. I/O 故障目录

十进制码	十六进制	描述
1	1	回路故障：短路，开环 等等
2	2	模块丢失：模块不响应
3	3	增加的模块：发现新模块
4	4	未使用的目录
5	5	未使用的目录
6	6	智能总线故障
7	7	通用内存故障t
8	8	EEPROM 故障，看门狗计时器时间到
9	9	新增加的 IOC
10	A	丢失的 IOC
11	B	IOC软件故障
12	C	强制的回路： 智能 I/O 的点被 HHM 强制
13	D	未被强制的回路： HHM 的强制被解除
14	E	失去90 系列的整体卡
15	F	额外的 90整体卡
16	10	发现增加的 90整体卡
17	11	发现额外的智能模块
18	12	一个 IOC 检测到一个硬件故障或一个波特率不匹配
19	13	智能总线控制器因为太多的故障发生而停止报告故障
20	14	I/O模块配置不匹配故障
21	15	GBC 软件意外
22	16	冗余智能模块转换角色
23	17	冗余总线上的模块不是激活状态

I/O 故障类型

I/O 故障类型元件的分故障故障目录包括回路故障，模块故障，I/O总线故障，丢失模块故障，意外故障，I/I配置不匹配，和GBC软件意外故障。其它故障目录没有定义，当故障目录不是这7种时总是置位0。表 B-18 列出了I/O 故障类型。

表 B-18. I/O 故障类型

号	描述
<i>回路故障目录的I/O 故障类型</i>	
1	数字量 I/O 点回路故障
2	模拟量 I/O 通道回路故障
3	GENA故障
4	低标准模拟量输入通道故障
5	远程I/O 扫描发生故障
<i>模块故障组目录的I/O 故障类型</i>	
0	模块故障(EEPROM, 看门狗)
1	模拟量到数字量同性故障感众标定故障
5	超过范围造成的用户标定错误
<i>I/O总线故障目录的I/O故障类型</i>	
0	因为PLC CPU和智能IOC之间的通讯超时，智能 IOC切断所有总线上的输出
1	智能总线故障 (在时间范围内，GBC没有中断来启动总线。)
3	智能 IOC 检测到SBA和另一个设备的冲突。
<i>丢失模块目录的I/O 故障类型</i>	
0	无特殊的说明
1	A/D通讯丢失
<i>意外故障目录的I/O 故障类型</i>	
1	智能 IOC 检测到智能I/O总线的一个高的错误计数器在并在最少1.5秒之内切断总线
<i>配置不匹配目录的I/O 故障类型</i>	
2	I/O扫描器检测到的模块号不匹配
3	I/O扫描器检测到I/O模块不存在
4	I/O 检测器检测到I/O类型不匹配
8	检测到完整的模拟量模块，扩展模拟量模块不匹配
9	广播数据长度不匹配
A	不支持的配置特性

表 B-18. I/O 故障类型——接上

<i>GBC 软件意外故障目录的I/O故障类型</i>	
1	输入数据队列满
2	GBC中的读写请求满。请求可能来自智能总线或者从COMMREQ来
3	从GBC到PLC的低优先级邮箱队列满。Plc 的响应丢失
4	在PLC完成初始化前，智能背景信息需要PLC动作。此信息可以忽略。
5	因为GBC软件版本太旧，智能报告故障信息未被执行
6	内部 GBC内存的意外使用。使用者应该确认COMMREQ的使用。

I/O 故障描述

当I/O故障目录是一个回路故障（数字量回路故障，模拟量回路故障，低电位模拟量故障）或者模块故障时，I/O 故障描述元件提供了一个特殊的故障码。其它故障未定义，通常设置为0。下面两个表是故障的描述。

表 B-19. I/O故障描述

号	描述
<i>数字量回路故障的故障描述</i>	
01	失去用户侧电源.
02	用户接线短路(对于智能模块，电流超过20安培).
04	持续过电流 (对于智能模块，电流超过2安培).
08	电流过小或无电流
10	节点温度过高
20	冗余智能总线切换失败
83	90-70系列 I/O 单独点故障(也显示在 I/O 模块故障目录中).
84	90-70 系列输出快熔断(也显示在I/O模块故障目录中)
<i>模拟量回路故障的故障描述</i>	
01	输入通道低报警.
02	输入通道高报警
04	输入通道低范围
08	输入通道超范围
10	输入通道开环检测
20	输出通道低范围
40	输出通道超范围
80	扩展通道未响应.
80	智能电流源模拟量模块反馈错误.

表 B-19. I/O 故障描述—接上

号	描述
低电压模拟量回路错误的故障描述*	
20	不正确的 RTD 连接或者热电偶连接错误.
40	热电偶模拟的冷节点传感器故障或者RTD模块的内部故障。.
80	输入通道短路 (只在智能 RTD 和张力应变模块中).
模块故障的描述**	
08	智能EEPROM 或 NVRAM 失败.
20	智能标定内存失败
40	智能共享RAM故障
80	智能内部回路故障t.
81	看门狗时间计数器到时 (只在数字量 I/O模块).
82	数字量 I/O模块辅助故障
83	90-70系列 I/O 单独点故障（也在回路故障目录中显示）
84	90-70系列输出快熔故障（也在回路故障目录中显示）
GENA故障的故障描述	
80	GENA模拟量或数字量点错误
87	远程 I/O 扫描器故障

* 当下面的低电压模拟量故障类型发生时，包括下面的模拟量错误 AI_LOW_ALARM, AI_HI_ALARM, AI_UNDER_RANGE, AI_OVER_RANGE, and OPEN_WIRE.

** 这些故障被报告为 HEADEND_FAULT 故障类型

I/O 故障特殊数据

I/O故障表可能包括最多21位 I/O 故障特殊数据。通常情况下，这部分包括与故障有关系的特殊信息。不是所有的数据都包括I/O故障特别数据。这段字节描述了有特别数据的。只有一个使用五个字节的I/O故障特殊数据，全局内存故障使用21位的数据。如果故障没在列表中，可能不包括I/O特殊数据。

智能总线控制器发生的故障最少有一个I/O故障特殊数据。其它字节存在的情况下这个字节是附加的数据。

回路故障

回路故障字段使用一个或者两个字节的故障特殊数据。当智能总线控制器报告这个故障时，智能总线控制器产生这个第一个字节。如果故障类型是GENA故障，第二个字节包括从GENA模块发出的报告故障信息。如果故障类型不是一个GENA故障，第二个字节包括回路的配置信息，如表B-20所示。

失去或增加模块

在发生丢失模块或增加模块的故障时，使用四个字节的特使数据。第一个字节是编码，入表B-20所示。第二个字节包括模块配置信息，编码入表B-20所示。第三位标明可能使用的输入回路数量。

通用内存故障

通用内存故障有10个字节的数据。第一个字节包括子网组号。其它9个字节包括通用变量名，格式为空端子的字符串。

强制/非强制的回路

当回路增加或移除强制值（强制回路或者不强制回路）时，会包括三个字节的故障特殊数据。第一个字节包括按照表B-20格式的编码。第三位包括包括数字量/模拟量信息。

IOC丢失

当PLC记录到一个IOC丢失故障时，记录会包括一个以表B-20为格式的I/O故障特别数据。

其它的 I/O 故障

除了以上列出的故障，当智能总线控制器报告下列故障中的一个时，在I/O故障表中包括一个以表B-20格式的数据。

- I/O总线故障
- 模块故障
- IOC 软件故障
- 额外智能模块
- IOC 硬件故障
- 过多的故障
- GBC 软件意外

模块开关

当故障目录是模块开关故障时，使用五个字节的特殊数据。第一个字节是编码，如表B-20所示。第三个字节标明可能使用的输入回路数量，第四个字节标明可能使用的输出回路的数量。最后一位（第5位）故障特殊数据包括发生切换故障的智能总线控制器的机架号/槽号。格式是槽号在低四位，机架号在高四位。

符号型错误特殊数据

下表列出的数据包括4种故障特殊数据:

- 模块回路配置
- 模块使用指示
- shuzl/模拟量指示
- IOC 丢失故障码

表 B-20. I/O 故障特殊数据

十进制	十六进制	描述
<i>回路配置</i>		
	1	输入回路
	2	输出回路
	3	带反馈的输出回路
<i>模块配置</i>		
	1	配置为只是输入 Block
	2	配置为只是输出
	3	配置为输入输出混合型
<i>数字量/模拟量指示</i>		
	1	数字量模块
	2	模拟量模块
<i>失去IOC 故障码</i>		
1	1	IOC与PLC失去响应
2	2	CPU 和 IOC失去同步
3	3	CPU/IOC通讯失败
4	4	VME 总线故障
5	5	VME 总线故障
6	6	CPU/IOC 通讯错误
7	7	CPU/IOC 通讯错误ailed
8	8	IOC 错误的响应了 CPU的请求
9	9	CPU/IOC 通讯错误
10	A	VME总线错误
11	B	BME 总线错误
12	C	CPU/IOC 通讯错误
13	D	CPU/IOC 通讯错误
14	E	CPU/IOC 通讯错误
15	F	IOC 错误的响应了一个CPU的请求
16	10	CPU/IOC 通讯错误
17	11	CPU/IOC 通讯错误
18	12	IOC 错误的响应了一个CPU的请求
19	13	CPU/IOC 通讯错误
20	14	CPU/IOC 通讯错误
21	15	内部 I/O 扫描器故障
22	16	IOC 错误的响应了一个CPU的请求
23	17	IOC 错误的响应了一个CPU的请求
24	18	IOC 错误的响应了一个CPU的请求
25	19	IOC 错误的响应了一个CPU的请求
26	1A	IOC 错误的响应了一个CPU的请求
27	1B	CPU/IOC 通讯错误
28	1C	当读输入数据时发生 VME 总线故障

表B-20. I/O 故障特殊数据 ——接上

Decimal Number	Hex Code	
29	1D	在读取输入诊断时发生VME总线错误
30	1E	CPU/IOC 通讯错误
31	1F	在写输出数据到IOC时发生VME总线故障
32	20	CPU/IOC 通讯错误
33	21	CPU/IOC 通讯错误
34	22	CPU/IOC 通讯错误
35	23	VME总线错误
36	24	VME总线错误
37	25	冗余I/O模块中不能从IOC读取数据
38	26	冗余I/O模块不能从IOC中读取数据
39	27	IOC 不支持配置的I/O 冗余
40	28	IOC 不能响应CPU请求
41	29	I/O 扫描器检测到系统中太多的IOC。
42	2A	I/O 扫描器检测到系统中太多的IOC
43	2B	VME总线错误
44	2C	VME 总线错误

特定故障的故障动作

强制的/非强制的回路故障被是信息型故障。其它所有的故障是诊断性和致命型。

.模块号不匹配，I/O类型不匹配和不存在I/O模块故障在系统配置不匹配故障表中的PLC故障表中报告。不是在I/O故障表中报告。

I/O 故障时间标记

六位的时间标记是PLC CPU在发生故障时系统时钟值。值是以BCD码格式标出。

表 B-21. I/O 故障时间标志

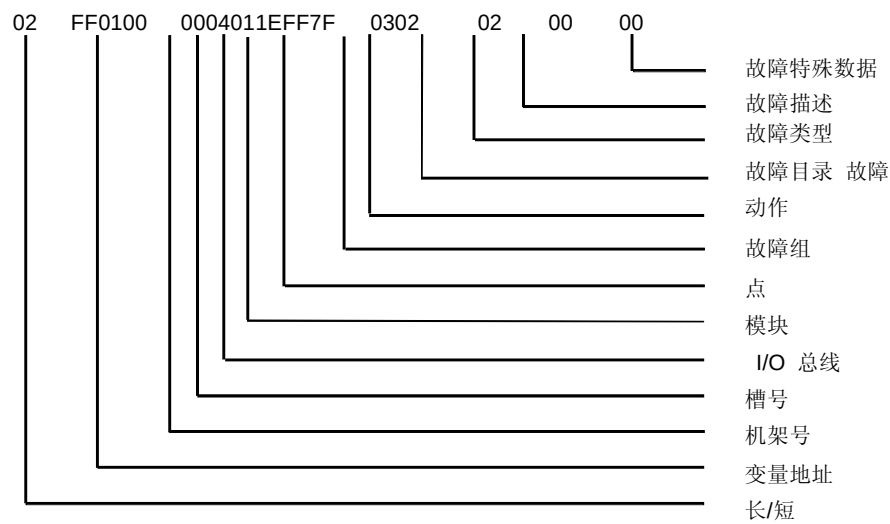
位号r	描述
1	秒.
2	分.
3	小时
4	天.
5	月.
6	年.

失去I/O 模块故障的例子

失去 I/O 故障的例子如下。（所有的数据是十六进制）

字段	值	描述
长/短	02	故障包括5个字节的故障信息。
变量地址	FF0100	FF 表示变量地址不是给定的。故障发生在整个模块上。
机架号	00	主机架 (机架号 0)。
槽号	04	槽号 4。在这个配置中，智能总线控制器包含在第四槽
总线	01	总线控制器在总线1上。当这位是重要的时，单通道的总线控制器在这个字段总是显示1。
模块号	1E	1E 是十进制的30。这个故障记录的是串行总线地址为30的智能设备
点	FF7F	在第一个字节的 FF (低字节) 显示在这个故障列表中此点无意义
故障组	03	失去 I/O 模块故障。
故障动作	02	诊断性故障
故障目录	02	检测不到模块。
故障类型	00	失去模块目录中的00 故障类型显示没有特别的故障发生。
故障描述	00	无故障描述信息
故障特殊数据	84	GBC 报告一个丢失的设备。只有一个字节的故障特殊数据是重要的。

在第四槽的智能总线控制器发现在1号总线的串行总线地址30在三个连续的总线扫描周期未能发送数据。总线控制器然后在I/O故障表中标记这个设备为丢失的设备。故障类型和故障描述在这个故障中无意义。故障特殊数据包括从GBC响应故障的一个字节的信息。

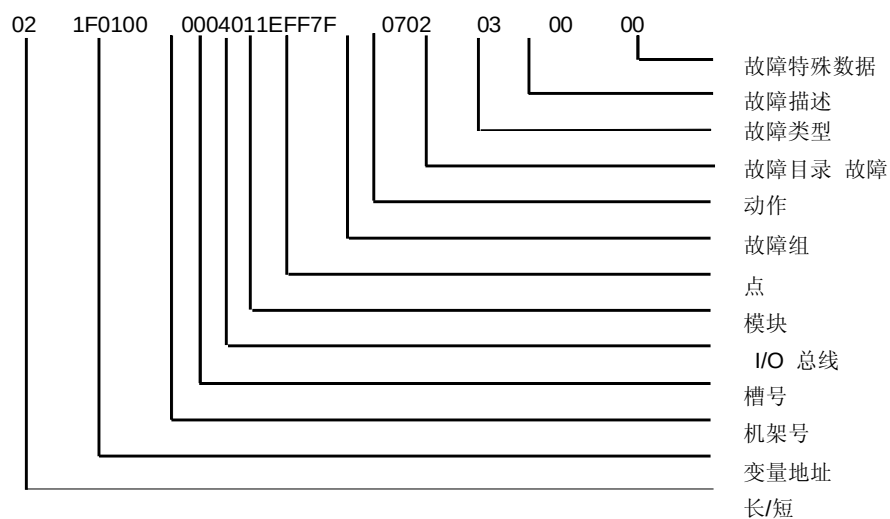


附加的 I/O 模块故障例子

附加的 I/O模块故障的例子如下 (所有的数据都是十六进制)

字段	值	描述
长/短	02	故障包括5个字节的故障信息data.
变量地址	1F0100	1F (十进制31) 显示模块有数字量成组的变量地址。软件在变量地址栏的%QI显示。 在数字量成组地址空间 显示地址为000 (显示的十六进制数据先显示低字节，然后是高字节)
机架号	00	主机架 (机架号 0).
槽号	04	槽4. 在这个配置中，槽4包括智能总线控制器。
总线	01	总线控制器中的总线 1。当这位是重要的时，单通道总线控制器总是显示为1。
模块号	1E	1E 是十进制的30。这个故障在智能总线地址30中被记录。
点	FF7F	FF是第一个 (低位) 显示在故障表中是无意义的。
故障组	07	附加的 I/O模块故障
故障动作	02	诊断性故障
故障目录	03	故障目录 03 是模块附加故障
故障类型	00	附加的模块故障无故障类型数据 其中的00是无意义的
故障描述	00	无故障描述信息
故障特别数据	80	GBC 报告一个附加地址。只有一个字节的故障特殊数据是重要的

在主机架的槽号为4的总线控制器接收到从串口总线地址为30的设备来的数据，显示显示设备在总线上重新被发现。从PLC cpu中的程序软件配置表中存储，CPU显示了设备的第一个点是%QI0001。故障类型和故障描述在这个故障中是无意义的。故障特殊数据包括从GBC显示故障来的一个字节。

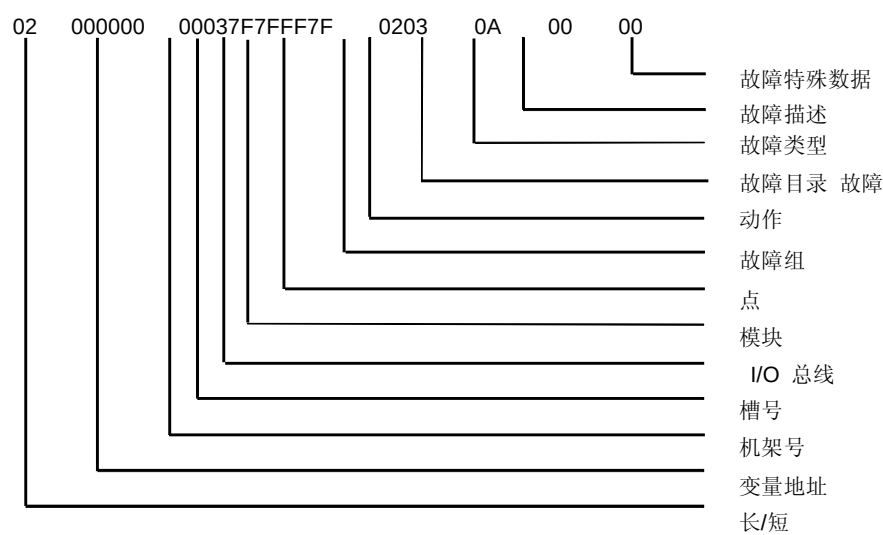


失去 IOC (I/O控制器)

失去 IOC 故障在下表中解释 (所有的数据是十六进制.)

字段	值	描述
长/短	02	这个故障总共包括五个字节的数据
变量地址	000000	本字段都是0表示在这个故障总变量地址是无意义的。.
机架号	00	主机架 (机架号 0).
槽号	03	槽 3. 在这个配置中, 智能总线控制器在第三槽中
总线	7F	因为是一个但通道 的GBC,并不需要总线号。7F 显示总线号是不重要的。
模块号	7F	十六进制的模块号7F显示故障中的模块号是无意义的。
点	FF7F	第一个字节的FF (1低字节) 显示在这个故障中的点故障是无意义的。
故障组	02	失去 IOC 故障.
故障动作	03	致命故障.
故障目录	0A	故障目录 0A (十进制的10i) 是一个失去 IOC故障.
故障类型	00	在此故障中无故障类型数据.
故障描述	00	没有故障描述信息
故障特殊数据	00	在这个故障中无故障特殊数据, 所以在失去IOC的故障中附加信息是无意义的。

检测到失去IOC故障。致命动作显示PLC CPU不会在故障清楚前将模块传送RUN模式。故障类型和故障描述在这个故障中是无意义的, 故障特殊数据对于PLC CPU来说是无意义的。

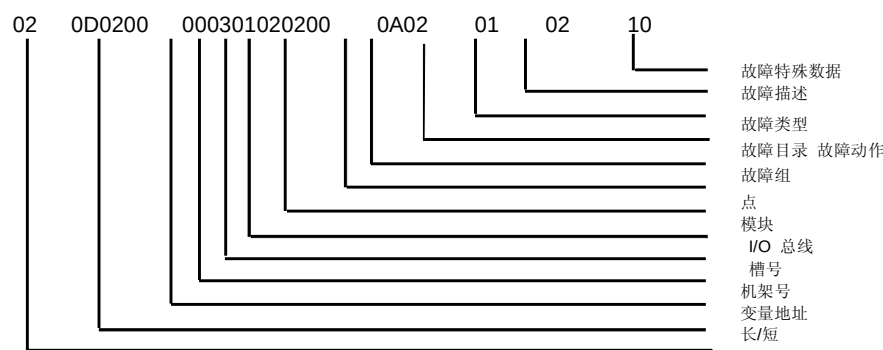


回路故障

回路故障在下表中解释(所有的数据是十六进制)

字段	值	描述
长/短	02	这个故障包括5个特殊字节
变量地址	0D0200	0D (十进制13)显示模块有成组的模拟量变量地址。软件在变量地址栏的%AQ显示。 在模拟量成组地址空间 显示地址为0200 (显示的十六进制数据先显示低字节，然后是高字节)
机架号	00	主机架 (机架号 0).
槽号	03	槽 3. 在这个配置中，智能总线控制器在槽3中。
总线号	01	总线控制器是总线1。当这个字节是重要的时，单通道总线控制器总在字段显示为1。
模块号	02	总线设备在串行总线地址02中记录。
点	0200	串行总线地址02在故障中记录的点2。
故障组	0A	0A 是十进制的10。 I/O 模块故障
故障动作	02	诊断性故障
故障目录	01	故障目录 01是回路故障
故障类型	02	回路故障目录中的故障类型 02是模拟量 I/O通道
故障描述	10	在模拟量回路描述中的故障描述中的十六进制10是一个开环故障
故障特殊数据	9B03	The GBC 报告一个回路故障。 只有字节 9B 是重要的.

在主机架中的在槽号3的智能总线控制器从串行总线地址2的模拟量模块接收到回路故障的报告故障信息。模块报告一个点2的开环故障。从编程软件软件的配置表中存储的数据，PLC CPU在串行总线地址02是标记为%AI和%AQ用户变量并且与位置0002的点2通讯。



Appendix

C

附录C 指令助记符

在程序的显示和编辑模式下，可以通过键入AND字符(&)和指令助记符的方法快速输入和查找程序指令。对于某些指令，也可通过变量地址或别名，标记名，或位置参考地址来指明。

本附录列出了Logicmaster 9070编程软件在编辑指令时可使用的助记符。在编程时，可以按 **ALT** 和 **I** 键来调出显示这些助记符的帮助画面。

表 C-1. 接点, 线圈, 连线, 记时器和计数器指令

指令	助记符									
	ALL	INT	UINT	DINT	BIT	BYTE	WORD	DWORD	REAL	MIXED
<u>接点</u>										
Any Contact	&CON	&CON								
-	&NOCON	&NOCON								
- /	&NCCON	&NCCON								
-	&PCON	&PCON								
-	&NCON	&NCON								
-[FAULT]-	&FA	&FA								
-[NOFLT]-	&NOF	&NOF								
-[HIALR]-	&HI	&HI								
-[LOALR]-	&LOA	&LOA								
<+>—	&CONC	&CONC								
<u>线圈</u>										
Any Coil	&COI	&COI								
-()-	&NOCOI	&NOCOI								
-(/)-	&NCCOI	&NCCOI								
-()-	&PCOI	&PCOI								
-()-	&NCOI	&NCOI								
-(S)-	&SL	&SL								
-(r)-	&RL	&RL								
-(SM)-	&SM	&SM								
-(RM)-	&RM	&RM								
-(M)-	&NOM	&NOM								
-(/M)-	&NCM	&NCM								
—<+>	&COILC	&COILC								
<u>连线</u>										
Horizontal	&HO	&HO								
Vertical	&VE	&VE								
<u>计时器</u>										
ONDTR										
OFDT	&OF	&OF								
TMR	&TM	&TM								
<u>计数器</u>										
UPCTR	&UP	&UP								
DNCTR	&DN	&DN								

表 C-2. 数学运算，关系和位操作指令

指令	Mnemonic									
	ALL	INT	UINT	DINT	BIT	BYTE	WORD	DWORD	REAL	MIXED
<u>数学运算指令</u>										
ADD	&AD	&AD_I	&AD_UI	&AD_DI						
SUB &AD_R	&SUB	&SUB_I	&SUB_UI	&SUB_DI						
MUL	&MUL	&MUL_I	&MUL_UI	&MUL_DI					&MUL_R	&MUL_M
DIV	&DIV	&DIV_I	&DIV_UI	&DIV_DI					&DIV_R	&DIV_M
MOD	&MOD	&MOD_I	&MOD_UI	&MOD_DI						
SQRT	&SQ	&SQ_I	&SQ_DI						&SQ_R	
ABS	&ABS	&ABS_I	&ABS_DI						&ABS_R	
SIN	&SIN	&SIN								
COS	&COS	&COS								
TAN	&TAN	&TAN								
IASIN	&ASIN	&ASIN								
ACOS	&ACOS	&ACOS								
ATAN	&ATAN	&ATAN								
LOG	&LOG	&LOG								
LN	&LN	&LN								
EXP	&EXP	&EXP								
EXPT	&EXPT	&EXPT								
DEG	&DEG	&DEG								
RAD	&RAD	&RAD								
<u>关系指令</u>										
EQ	&EQ	&EQ_I	&EQ_UI	&EQ_DI						
NE	&NE	&NE_I	&NE_UI	&NE_DI					&NE_R	
GT	&GT	&GT_I	&GT_UI	&GT_DI					&GT_R	
GE	&GE	&GE_I	&GE_UI	&GE_DI					&GE_R	
LT	&LT	&LT_I	&LT_UI	&LT_DI					&LT_R	
LE	&LE	&LE_I	&LE_UI	&LE_DI					&LE_R	
CMP	&CM	&CM_I	&CM_UI	&CM_DI					&CM_R	
<u>位操作指令</u>										
AND	&AN						&AN_W	&AN_DW		
OR	&OR						&OR_W	&OR_DW		
XOR	&XO						&XO_W	&XO_DW		
NOT	&NOT						&NOT_W	&NOT_DW		
SHL	&SHL						&SHL_W	&SHL_DW		
SHR	&SHR						&SHR_W	&SHR_DW		
ROL	&ROL						&ROL_W	&ROL_DW		
ROR	&ROR						&ROR_W	&ROR_DW		
BTST	&BT						&BT_W	&BT_DW		
BSET	&BS						&BS_W	&BS_DW		
BLCR	&BCL						&BCL_W	&BCL_DW		
BPOS	&BP						&BP_W	&BP_DW		
MCMP	&MCM						&MCM_W	&MCM_DW		

表 C-3. 数据移动和数据表操作

指令	Mnemonic									
	ALL	INT	UINT	DINT	BIT	BYTE	WORD	DWORD	REAL	MIXED
<u>数据移动</u>										
MOVE	&MOV	&MOV_I	&MOV_UI	&MOV_DI	&MOV_BI		&MOV_W	&MOV_DW	&MOV_R	
BLKMOV	&BLKM	&BLKM_I	&BLKM_UI	&BLKM_DI			&BLKM_W	&BLKM_DW	&BLKM_R	
BLKCLR	&BLKC									
SHFR	&SHF				&SHF_BI		&SHF_W	&SHF_DW		
BITSEQ	&BI									
SWAP	&SW						&SW_W	&SW_DW		
COMMREQ	&COMMR									
VMERD	&VMERD					&VMERD_BY	&VMERD_W			
VMEWRT	&VMEW					&VMEWR_BY	&VMERD_W			
VMERMW	&VMERM					&VMERM_BY	&VMERM_W			
VMETST	&VMET					&VMET_BY	&VMET_W			
VME_CFG_RD	&CFG_RD									
VME_CFG_WRT	&CFGWRT									
DATA_INIT	&DINI									
DATA_INIT_COMM	&DCO									
DATA_INIT_ASCII	&DA									
<u>数据表</u>										
TBLRD	&TBLR	&TBLR_I	&TBLR_UI	&TBLR_DI			&TBLR_W	&TBLR_DW		
TBLWR	&TBLW	&TBLW_I	&TBLW_UI	&TBLW_DI			&TBLW_W	&TBLW_DW		
LIFORD	&LIFOR	&LIFOR_I	&LIFOR_UI	&LIFOR_DI			&LIFOR_W	&LIFOR_DW		
LIFOWRT	&LIFOW	&LIFOW_I	&LIFOW_UI	&LIFOW_DI			&LIFOW_W	&LIFOW_DW		
FIFORD	&FIFOR	&FIFOR_I	&FIFOR_UI	&FIFOR_DI			&FIFOR_W	&FIFOR_DW		
FIFOWRT	&FIFOW	&FIFOW_I	&FIFOW_UI	&FIFOW_DI			&FIFOW_W	&FIFOW_DW		
SORT	&SORT	&SORT_I	&SORT_UI				&SORT_W			
ARRAY_MOVE	&AR	&AR_I	&AR_UI	&AR_DI	&AR_BI	&AR_BY	&AR_W	&AR_DW		
SRCH_EQ	&SRCHE	&SRCHE_I	&SRCHE_UI	&SRCHE_DI		&SRCHE_BY	&SRCH_W	&SRCH_DW		
SRCH_NE	&SRCHN	&SRCHN_I	&SRCHN_UI	&SRCHN_DI		&SRCHN_BY	&SRCHN_W	&SRCHN_DW		
SRCH_GT	&SRCHGT	&SRCHGT_I	&SRCHGT_UI	&SRCHGT_DI		&SRCHGT_BY	&SRCHGT_W	&SRCHGT_DW		
SRCH_GE	&SRCHGE	&SRCHGE_I	&SRCHGE_UI	&SRCHGE_DI		&SRCHGE_BY	&SRCHGE_W	&SRCHGE_DW		
SRCH_LT	&SRCHLT	&SRCHLT_I	&SRCHLT_UI	&SRCHLT_DI		&SRCHLT_BY	&SRCHLT_W	&SRCHLT_DW		
SRCH_LE	&SRCHLE	&SRCHLE_I	&SRCHLE_UI	&SRCHLE_DI		&SRCHLE_BY	&SRCHLE_W	&SRCHLE_DW		

表 C-4. 转换和控制操作指令

指令	助记符									
	ALL	INT	UINT	DINT	BIT	BYTE	WORD	DWORD	REAL	MIXED
<u>转换指令</u>										
to BCD-4	&BCD4	&BCD4	&BCD4_UI							
to BCD-8				&BCD8						
to UINT	&UI	&UI		&UI_DI					&UI_R	
to INT	&I	&I	&I_DI						&I_R	
to DINT	&DIN	&DIN	&DIN_UI						&DIN_R	
to REAL	&R	&R	&R_UI	&R_DI						
BCD-4 to UINT	&UI_BCD4									
BCD-4 to INT	&I_BCD4									
BCD-8 to DINT	&DI_BCD8									
BCD-4 to REAL	&R_BCD4									
BCD-8 to REAL	&R_BCD8									
TRUN	&TRINT	&TRINT		&TRDINT						
<u>控制</u>										
CALL	&CA									
DOIO	&DO									
SUSIO	&SUS									
MCR	&MCR									
ENDMCR	&ENDMCR									
JUMP	&JUMP									
LABEL	&LABEL									
COMMENT	&COMME									
SVCREQ	&SV									
PIDISA	&PIDIS									
PIDIND	&PIDIN									
FOR	&FOR									
END_FOR	&ENDFOR									
EXIT	&EXIT									

Appendix D

附录D 内存分配

CPU 内存大小是PLC内存中可供用户应用程序使用的内存的字节数。CPU内存的大小由CPU的型号和所选的子板的型号决定。目前有如下一些种类可选。

CPU	内存大小
32K	32,768
64K	65,536
128K	131,072
256K	262,144
512K	524,288
1M	1,048,576
6M	6,291,456

CPU内存的占用可分为如下若干项：

寄存器内存大小 (%R)

字节数 = %R 配置的变量地址 2

模拟量输入 (%AI)

点故障使能：字节数 = %AI配置的变量地址 3

点故障禁用：字节数 = %AI配置的变量地址 2

模拟量输出 (%AQ)

点故障使能：字节数 = %AI配置的变量地址 3

点故障禁用：字节数 = %AI配置的变量地址 2

离散量点故障

如果点故障使能：

字节数 = 128对于 73x 型CPU

字节数 = 512对于 77x 型CPU

字节数 = 3072 对于78x 和 9xx 型CPU

故障表

字节数 = (I/O故障条目配置 + PLC故障条目配置 - 48) 48

注意

默认故障表大小 (16 PLC 故障, 32 I/O故障) 不需要占用用户内存。

以太网全局数据

字节数 = 0 若没有配置以太网全局数据交换

C 测试连接

字节数 = 11772 当连接建立；当无连接时不占用用户内存

I/O扫描设置文件

基于扫描设定使用的数量。

注意

如果每个扫描都扫描用户所有I/O(默认)，则需要32 字节的用户内存。.

模板配置文件

仅适用于 FBC 模板

名称解析文件

字节数 = 0若无用户定义的名称解析。

用户协议文件

字节数 = 0若无由用户建立的协议文件。

用户程序

见下面对用户程序的描述。

对于冗余产品有额外的需求，如788, 789, 和CGR935CPU，见相关的用户手册。

用户程序内存使用

CPU内存大小减去以上配置项所占用内存剩余的内存部分可全部用于用户程序逻辑。然而，多个程序可以共同利用1M 或更大的CPU内存。

每个用户创建的程序都有单独最大程序量。它的大小取决于程序类型。对于程序的大小限制如下表所示。要说明的是，最大程序量仅适用于有1M及以上内存的CPU，对于内存小的CPU，程序量由CPU的内存大小来决定。

程序类型	堆栈大小	最大程序量 *
LD/SFC 程序	12K (默认为 12K)	544K 堆栈大小不计入程序或整个用户内存。
LD/SFC 程序	> 12K	544K – 堆栈大小
独立 C程序	N/A	564K – 堆栈大小

*最大程序量仅适用于有1M及以上内存的CPU，对于内存小的CPU，程序量由CPU的内存大小来决定。.

以下用户程序项计入CPU内存数量，但不占用相关程序的用户程序存储空间。

C 初始化数据

独立C程序和EXE程序块的初始化数据区独立于程序映象数据区，它不计入用户程序的大小。这部分内存不占用CPU内存空间。详见《系列90-70 PLC C程序工具包用户手册》。

独立 C 程序控制结构

创建并存储于CPU的每个独立C程序块占用8K (8192字节)，下列用户程序相关项计入用户程序内存空间，同时也需要占用CPU的内存资源。i

%P 程序内存 ¹

如果最大% P只用到 %P00128 或以下，则: 字节数 = 399

如果最大% P用到超过 %P00128:
 字节数 = 使用的最高 %P (延续到下一连续32个) 2 + 143

¹ 延续到下一个32个的%P和 %L计算: 将变量地址除以32，向上约等于一个整数，然后再乘以32。例如167 %P的变量区域为192，因为167 / 32 =5.21875，向上约等于6，而6 X 32 = 192。

%L 程序块内存使用¹

对于每个程序块，

如果最大 %L 等于或小于

%L0064，则：字节数 = 271

如果最大 %L 大于 %L0064：

字节数 = 使用的最高 %L 延续到下一连续32个) $2 + 143$

中断程序块控制结构

142 字节每中断块

程序逻辑和管理**C 数据区**

独立C程序和EXE程序块的数据区被当作用户程序的一部分而被计入总的程序量中。

可配置堆栈大小

见上页表对程序大小的限制。

注意

第一个 12K 的 LD/SFC 程序堆栈不计入CPU的内存占用中。

同样，要说明的是程序块的大小决定于程序自身的大小和逻辑/寄存器占用的内存大小之和 (%L)。关于程序块大小见调用指令于表 A-1 (附录 A)。

注意

如果用户使用具有1M及以上的内存的CPU来编制LD/SFC逻辑，则它们可以通过改变%P 或 %L为%R，%AI或 %AQ来获得更多的内存程序空间。比如可以改变需要重复编译的程序块，并在停止模式存储于CPU。

¹ 延续到下一个32个的%P和 %L计算：将变量地址除以32，向上约等于一个整数，然后再乘以32。例如167 %P的变量区域为192，因为 $167 / 32 = 5.21875$ ，向上约等于6，而 $6 \times 32 = 192$ 。

Appendix

E

附录E 功能键

下表列出了可在Logicmaster 90-70软件环境中使用的功能键。这些信息也可通过按ALT- K 键来显示在编程器屏幕上。

键 序列	说明	键 序列	说明
软件包中可用功能键			
ALT-A	退出。	CTRL-Break	退出软件。
ALT-C	清除区域。	Esc	退出缩放。
ALT-M	改变编程器模式。	CTRL-Home	先前命令行内容。
ALT-R	更改PLC RUN/STOP 状态。	CTRL-End	下一命令行内容。
ALT-E	强制状态区域。	CTRL-	区域的光标左侧。
ALT-L	列出目录文件。	CTRL-	区域的光标左侧。
ALT-P	打印屏幕。	CTRL-D	减少变量地址。
ALT-H	帮助。	CTRL-U	增加变量地址。
ALT-K	功能键帮助。	Tab	更改/增加区域内容。
ALT-I	指令助记符帮助。	Shift-Tab	更改/减少区域内容。
ALT-T	开始 TEACH 模式。	Enter	确认区域内容。
ALT-Q	停止 TEACH 模式。	CTRL-E	显示上一系统错误。
ALT-Z	暂停 TEACH 重放。	F12 (或-)	强制离散变量
ALT-n	回放文件 n (n = 0 thru 9)。	F11 (或*)	控制离散变量。
仅用于编辑模式的功能键			
ALT-B	强制文本编辑铃。	Keypad +	接受程序行。
ALT-D	删除程序行或其元素。	Enter	接受程序行。
ALT-S	存储块到 PLC 和磁盘。	CTRL-PgUp	先前的程序行。
ALT-X	显示缩放等级 & 块状态。	CTRL-PgDn	下一程序行。
ALT-V	变量表窗口。	~	水平连线。
ALT-W	显示PSB 参数表		竖直连线。
ALT-F2	进入操作数变量表。	Tab	进入下一个操作区。
特殊功能键			
ALT-G	单个扫描测试。仅用于程序编辑器，变量表编辑器和状态故障表。		
ALT-N	强制显示选项。仅用于程序编辑器，变量表编辑器。		
ALT-O	口令占用。仅用于配置软件的密码屏幕。		
ALT-U	更新磁盘。		

下页中的帮助卡片包含了功能键帮助列表，同时也列出了Logicmaster 90 软件中可用的指令助记符的帮助文本。 此卡片在本手册中共打印3份，便于从该手册中取下，方便用户查找。

本页包含了GFJ-056B的第一面。

本页包含了 GFJ-056B 的第二面。

Appendix F

附录F 使用浮点数

浮点数

编程软件提供了编辑，显示，存储和得到实型数值的手段。然而为了使用浮点数，你必须使用 732, 772, 782 , 914或更高版本的CPU。浮点数以十进制科学计数法的形式表示，共有六位有效数字。

注意

在参数子程序(PSB)中使用浮点数通常使用NWORD型参数，而不是WORD型。

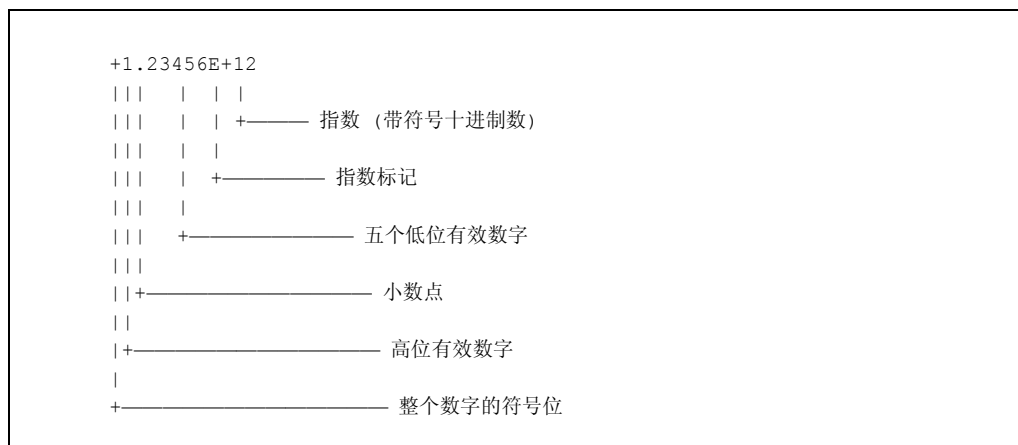
注意

本手册中，“浮点数”和“实型”都可用于说明浮点数及对其在软件中进行显示或登录。

在软件中使用如下格式。实型数的数值范围为 9999999999到 .0001，显示无指数并且最高不超过六位或七位的有效数字。例如：

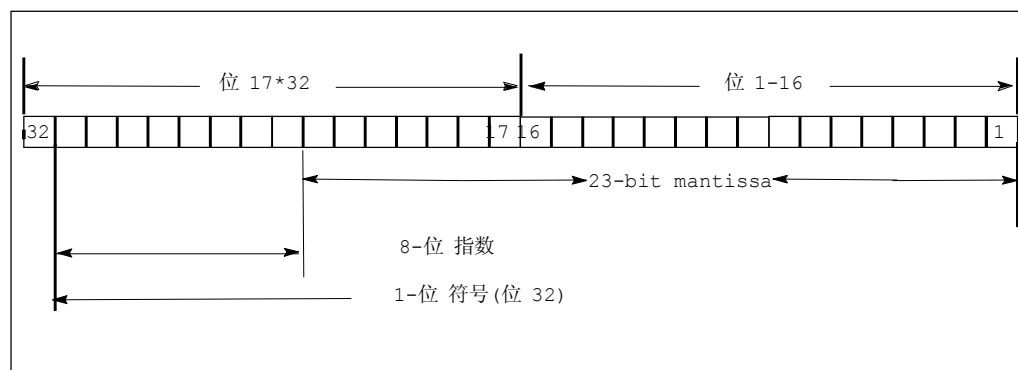
输入数	显示	说明
.000123456789	+.0001234567	十位数，六位或七位有效数字。
-12.345e-2	-.1234500	七位数，六位或七位有效数字。
1234	+1234.000	七位数，六位或七位有效数字。

输出范围见下图，仅六位有效数字被显示，且按照下面的格式进行显示：

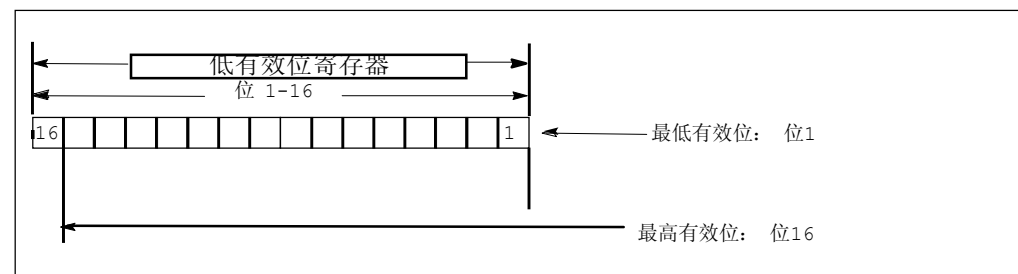


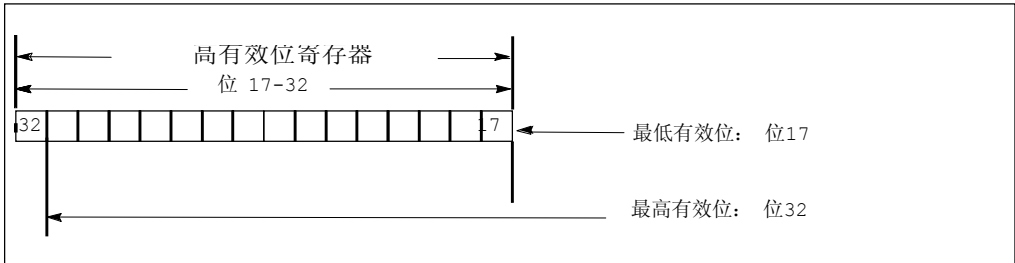
浮点数的内部格式

浮点数是按照精确的IEEE-标准格式存储的。这种格式需要32位，它们被转换为两个相邻的16位PLC寄存器。这些位的编码格式如下图所示。



一个单独的浮点数使用的寄存器如下图所示。图中，如果浮点数占用了寄存器R5和R6，则R5为低有效位寄存器，R6为高有效位寄存器。





浮点值

使用下表来对存储在两个寄存器中的二进制数进行相应浮点数值计算。

指数 (e)	尾数 (f)	浮点数的值
255	非零	不是一个有效数字 (NaN)。
255	0	$-1^s *$
$0 < e < 255$	任意值	$-1^s * 2^{e-127} * 1.f$
0	非零	$-1^s * 2^{-126} * 0.f$
0	0	0

f = 尾数. 尾数为二进制部分
e = 指数. 指数为整型数 E 比如, E+127为 2的幂, 而尾数为浮点值的累计增加。
s = 符号位.
* = 乘法运算.

例如, 对于浮点数12.5。IEEE 标准浮点数的二进制表示形式为:

01000001 01001000 00000000 00000000

或41480000 hex为其十六进制表示形式。最高有效位（符号位）为零(s=0), 接下来8位最高有效位为10000010 或表示为十进制的130(e=130)。

尾数以小数点后的最高有效位23位开始的二进制数形式存储。因此, 尾数的最高有效位是2⁻¹的倍数, 最高有效位下一位为2⁻² 的倍数, 依此直到最低有效位为2⁻²³。最后23位（尾数）为:

1001000 00000000 00000000

那么尾数值为 .5625 (即2⁻¹ + 2⁻⁴).

由于e > 0 和e < 255, 可使用上面表中第三个公式计算:

数字 = $-1^s * 2^{e-127} * 1.f$
= $-1^0 * 2^{130-127} * 1.5625$
= $1 * 2^3 * 1.5625$
= $8 * 1.5625$
= 12.5

因此，可以看出以上的二进制代码是正确的。

以这种格式存储的数字范围为 $\pm 1.401298\text{E}-45$ 到 $\pm 3.402823\text{E}+38$ 和数字零。

输入和显示浮点数

在尾数中，可以有六位或七位有效数字的精度数字被输入和存储；然而，软件只显示这些数字的前六位。尾数前可带正负符号。若没有输入符号，则浮点数将按照正值来存储。

如果输入带指数的数字，则指数前必须加字母“E”或“e”并且尾数必须包含小数点，以避免同十六进制数相混淆。指数部分可带符号，但如果未输入符号，则按照正值计入。如果没有输入指数，则按0处理。浮点数输入时不允许有空格。

为了便于编程者使用，软件支持命令行和数据录入区域两种方式输入数据。这些格式包括整型，十进制数，带指数部分的十进制数。一旦用户完成输入数据并按**Enter** 键，这些数据便转换成标准格式显示出来。

合法的浮点数条目及其标准化显示的例子如下表所示。

输入	显示
250	+250,0000
+4	+4.000000
-2383019	-2383019.
34.	+34.00000
-.0036209	-.003620900
12.E+9	+1.20000E+10
-.0004E-11	-4.00000E-15
731.0388	+731.0388
99.20003e-29	+9.92000E-28

非法的浮点数输入条目如下表所示。

非法条目	说明
-433E23	缺少小数点。
10e-19	缺少小数点。
1 0.e19	尾数的数字或字母之间不能包含空格。这将被记作10.e0，同时会有相应的错误提示信息。
4.1e1 9	指数的数字或字母之间不能包含空格。这将被记作4.1e0，同时会有相应的错误提示信息。

浮点数及其运算错误

当数字大于 $3.402823\text{E}+38$ 或小于 $-3.402823\text{E}+38$ ，则REAL功能块就会产生溢出。如果溢出出现，功能块使能输出将置为OFF；结果将被置为正无穷大(大于 $3.402823\text{E}+38$ 时)或负无穷大(小于 $-3.402823\text{E}+38$ 时)。可通过检测使能输出来查找溢出发生的位置。

```
POS_INF  = 7F800000h - IEEE 十六进制正无穷大
NEG_INF  = FF800000h - IEEE 十六进制负无穷大
```

如果由于溢出引起的无穷大的数据被用作其他REAL功能块的操作数，可能会造成不确定的结果。不确定的结果可能被表示为非数字(NaN)。例如，正无穷与负无穷之和是不确定的。当ADD_REAL 功能块使用正无穷和负无穷作为其操作数时，其结果将产生一个NaN。

每个REAL 功能块仅产生与此功能块相关的特定的 NaN。

```
NaN_ADD   = 7F81FFFFh - 十六进制实型加法错误值
NaN_SUB   = 7F81FFFFh - 十六进制实型减法错误值
NaN_MUL   = 7F82FFFFh - 十六进制实型乘法错误值
NaN_DIV   = 7F83FFFFh - 十六进制实型除法错误值
NaN_SQRT  = 7F84FFFFh - 十六进制实型平方根错误值
NaN_LOG   = 7F85FFFFh - 十六进制实型对数错误值
NaN_POW0  = 7F86FFFFh - 十六进制实型指数错误值
NaN_SIN   = 7F87FFFFh - 十六进制实型正弦错误值
NaN_COS   = 7F88FFFFh - 十六进制实型余弦错误值
NaN_TAN   = 7F89FFFFh - 十六进制实型正切错误值
NaN_ASIN  = 7F8AFFFFh - 十六进制实型反正弦错误值
NaN_ACOS  = 7F8BFFFFh - 十六进制实型反余弦错误值
NaN_BCD   = 7F8CFFFFh - BCD-4 到实型错误
REAL_INDEF = FFC00000 - 不确定实型，0除0错误
```

当一个NaN 的结果被送入其他功能块，它会通过其他功能块而进入结果中。例如如果NaN_ADD是SUB_REAL功能块的第一个操作数，则SUB_REAL的结果将为NaN_ADD。由于此特性传递NaN通过功能块，可以通过这种方法来识别发生NaN的地方。

注意

对于 NaN，使能输出为Off。

%

%G 变量
%G变量和CPU内存, 2-15

:

:以太网全局数据
扫描影响时间, A-24

@

@ 间接变量标记, 2-11

A

ABS, 6-8
绝对值根功能, 6-8
ACOS, 6-10
ADD, 6-2
加法功能块, 6-2
报警接点, 3-8
报警错误代码, B-6
报警处理器, 3-10
ALT 键, E-1
ALW_OFF, 2-21
ALW_ON, 2-21
模拟量I/O诊断数据, 2-87
模拟量输入寄存器变量 (%AI), 2-11
模拟量输出寄存器变量(%AQ), 2-11
AND, 8-3
应用程序任务执行, 2-5
 内务处理, 2-4
 输入扫描, 2-4
 输出扫描, 2-5
ARRAY_MOVE, 10-17
ASIN, 6-10
ATAN, 6-10

B

后台窗口, 2-7
以10为底对数功能块, 6-12
基本扫描时间, A-11
BCD-4, 11-2
BCD-8, 11-4
BCLR, 8-16
位清除功能块, 8-16
位操作功能块
 AND, 8-3
 BCLR, 8-16
 BPOS, 8-18
 BSET, 8-16
 BTST, 8-14

MCMP, 8-20
NOT, 8-7
OR, 8-3
ROL, 8-12
ROR, 8-12
SHL, 8-9
SHR, 8-9
XOR, 8-5

位位置功能块, 8-18
位排序功能块, 9-11
置位功能块, 8-16
位检测功能块, 8-14
BITSEQ, 9-11

 内存需求, 9-12

BLKCLR, 9-6
BLKMOV, 9-4
块清除功能块, 9-6
块移动功能块, 9-4
BPOS, 8-18
BSET, 8-16
BTST, 8-14
总线配置, FIP I/O, 2-85
总线配置, Genius I/O, 2-83

C

C 测试工具, 1-1
 编程器通讯窗口, 2-6
C 程序结构
 大小限制, 2-30
C 程序, 2-42
 数据封装, 2-42
C v. RLD 程序, 2-47
估算扫描时间, A-31
CALL, 12-3
CALL_EXTERNAL, 12-4
调用外部功能块, 12-4
调用功能块, 12-3
CALL子程序块, 12-6
调用子程序功能块, 12-6
检测, 12-36
任务状态检测, 12-36
故障等级, 3-11
时钟, 2-77
 历史计时时钟, 2-77
 日历时钟, 2-77
 使用 SVCREQ 功能块 #16 读取时钟, 2-77
 使用 SVCREQ 功能块 #7 读取和设定, 2-77
CMP, 7-4
线圈, 4-8
线圈, 4-3
 线圈, 4-8
 续行线圈, 4-12

- 负线圈, 4-8
- 负向保持线圈, 4-8
- 负跳变线圈, 4-9
- 正跳变线圈, 4-9
- RESET线圈, 4-10
- 保持线圈, 4-8
- 保持型复位线圈, 4-11
- 保持型置位线圈, 4-11
- 置位线圈, 4-10
- 注释, 12-20
- 注释功能块, 12-20
- COMMREQ, 9-17
- 通讯请求功能块, 9-17
- 比较功能块, 7-4
- 可配置故障作用, 3-4
 - 诊断性故障, 3-4
 - 致命故障, 3-4
 - 信息类故障, 3-4
- 可配置故障, 3-17
- 配置, 4-1
- 系统配置, 2-74
- 恒定时间扫描, 2-10
- 恒定窗口模式, 2-10
- 接点, 4-2
 - 续行接点, 4-12
 - 故障接点, 4-7
 - 高报警接点, 4-7
 - 低报警接点, 4-7
 - 负跳变接点, 4-4
 - 无故障接点, 4-7
 - 常闭接点, 4-4
 - 常开接点, 4-4
 - 正跳变接点, 4-4
- 续行接点, 4-12
- 续行接点, 4-12
- 控制功能块
 - CALL, 12-3
 - CALL EXTERNAL, 12-4
 - CALL SUBROUTINE, 12-6
 - COMMENT, 12-20
 - DOIO, 12-10
 - ENDMCR, 12-17
 - FOR, END_FOR, and EXIT, 12-21
 - JUMP, 12-18
 - LABEL, 12-19
 - MCR, 12-16
 - PID, 12-88
 - SUSIO, 12-14
 - SVCREQ, 12-25
- 便捷变量, 见系统状态变量
- 转换功能块
 - BCD-4, 11-2
 - BCD-8, 11-4
 - DINT, 11-10
 - INT, 11-8
 - REAL, 11-12
 - TRUN, 11-14

- UINT, 11-6
- 转换为 BCD-4 功能块, 11-2
- 转换为 BCD-8 功能块, 11-4
- 转换为双精度带符号整型功能块, 11-10
- 转换为实型功能块, 11-12
- 转换为带符号整型功能块, 11-8
- 转换为无符号整型功能块, 11-6
- 内存损坏, 3-16
- COS, 6-10
- 余弦功能块, 6-10
- 计数器
 - DNCTR, 5-14
 - 功能块数据, 5-1
 - UPCTR, 5-12
- CPU
 - 执行图表, A-39
- CPU 默认内存大小, 2-14
- CPU 运行数据, A-1
 - 基本扫描时间, A-11
 - 扫描时间的估算, A-31
 - FIP I/O 扫描影响时间
 - 工作表, A-23
 - Genius I/O 扫描影响时间, A-18
 - 工作表, A-20
 - I/O 中断运行和影响时间, A-27
 - I/O 模板扫描影响时间
 - 工作表, A-15
 - I/O 扫描和 I/O 故障扫描影响时间, A-14
 - 指令定时, A-1
 - 编程器扫描影响时间, A-12
 - 以太网全局数据扫描影响时间, A-24
 - FIP I/O 和 FBC扫描时间, A-21
 - Genius I/O和GBC扫描时间, A-17
 - 智能可选模块扫描时间, A-26
 - 系列90-70 I/O模板扫描时间, A-14
- CPU 冗余, 3-35
- CTRL 键, E-1

D

- 通讯窗口数据一致性, 2-8
- 数据流限制, 2-32
- 数据初始化功能块, 9-40
- ASCII数据初始化功能块, 9-46
- 通讯请求数据初始化功能块, 9-43
- DLAN数据初始化功能块, 9-48
- 数据映射, 2-83
 - 默认情况, 2-83
 - FIP I/O 数据映射, 2-86
 - Genius I/O数据映射, 2-83

数据移动功能块

- BITSEQ, 9-11
- BLKCLR, 9-6
- BLKMOV, 9-4
- COMMREQ, 9-17
- DATA_INIT, 9-40
- DATA_INIT_ASCII, 9-46
- DATA_INIT_COMM, 9-43
- DATA_INIT_DLAN, 9-48
- MOVE, 9-2
- SHFR, 9-8
- SWAP, 9-15
- VME_CFG_RD, 9-34
- VME_CFG_WRT, 9-37
- VMERD, 9-25
- VMERMW, 9-29
- VMETST, 9-31
- VMEWRT, 9-27

数据保持性, 2-16

数据范围, 2-18

数据表功能块

- ARRAY_MOVE, 10-17
- FIFORD, 10-11
- FIFOWRT, 10-13
- LIFORD, 10-7
- LIFOWRT, 10-9
- SORT, 10-15
- SRCH_EQ, 10-21
- SRCH_GE, 10-21
- SRCH_GT, 10-21
- SRCH_LE, 10-21
- SRCH_LT, 10-21
- SRCH_NE, 10-21
- TABLE_RANGE, 7-6, 10-24
- TBLRD, 10-3
- TBLWRT, 10-5

数据类型, 2-19

- DATA_INIT, 9-40
- DATA_INIT_ASCII, 9-46
- DATA_INIT_COMM, 9-43
- DATA_INIT_DLAN, 9-48
- DEG, 6-14

诊断数据, 模拟量 I/O, 2-87

诊断性故障, 3-4, 3-12

- 添加块, 3-47
- 添加 I/O 模板, 3-46
- 添加 IOC, 3-45
- 添加额外机架, 3-21
- 应用程序故障, 3-30
- 块开关, 3-50
- 回路故障, 3-41
- 恒定时间扫描超时, 3-29
- 外部块, 3-47
- 额外 I/O 模板, 3-46
- I/O 总线故障, 3-48
- I/O 故障表满, 3-29
- IOC 硬件失败, 3-50
- 块未接通, 3-47

I/O 模板未接通, 3-46

可选模块未接通或丢失, 3-18

电池低电量, 3-28

模板故障, 3-49

模板硬件故障, 3-26

PLC 系统故障表满, 3-29

- 程序微循环时间超时, 3-29
- 重启, 添加, 或多余可选模块, 3-21

系统总线错误, 3-25

诊断信息, 离散量 I/O, 2-87

DINT, 11-10

离散量 I/O 诊断信息, 2-87

离散变量, 2-12

- 大小和缺省设置, 2-14

DIV, 6-2

除法功能块, 6-2

DNCTR, 5-14

DOIO 功能块, 12-10

减计数器, 5-14

E

历史时钟, 2-77

结束主控制继电器功能块, 12-17

END_FOR, 12-21

指令结束, 12-21

ENDMCR, 12-17

EQ, 7-2

等于功能块, 7-2

错误代码, B-6

ESCM 端口状态 (SVCREQ #39), 12-72

以太网全局数据

- 动态逻辑驱动, 12-74

以太网全局数据扫描影响时间, A-24

EXIT, 12-21

退出指令, 12-21

EXP, 6-12

指数功能块, 6-12

- 激励 e, 6-12

- 激励 X, 6-12

EXPT, 6-12

F

致命故障, 3-4, 3-12

- 存储期间通讯故障, 3-36

- 上电时用户程序损坏, 3-33

IOC 软件故障, 3-49

IOC 未接通, 3-45

机架未接通或丢失, 3-17

可选模板软件故障, 3-27

PLC CPU 硬件故障, 3-26

PLC CPU 系统软件故障 3-35

程序块检测失败, 3-28

运行模式存储失败, 3-37

- 系统总线故障, 3-32
- 系统配置不符, 3-22
- 总线控制器过多, 3-36
- 故障作用, 3-12
 - 诊断故障, 3-12
 - 致命故障, 3-12
 - 故障响应, 3-12
 - I/O 故障作用, B-21
 - 信息故障, 3-12
 - PLC 故障作用, B-5
- 故障信息, B-22
- 故障接点, 4-7
- 故障接点, 3-6
- 故障说明, B-24
- 故障解释及纠正, 3-1
 - 获取附加故障信息, 3-15
 - 添加块, 3-47
 - 添加 I/O 模板, 3-46
 - 添加 IOC, 3-45
 - 添加额外机架, 3-21
 - 模拟量故障, 3-43
 - 应用程序故障, 3-30
 - 块开关, 3-50
 - 回路故障, 3-41
 - 存储时通讯失败, 3-36
 - 可配置故障, 3-17
 - 恒定时间扫描超时, 3-29
 - 上电时用户程序损坏, 3-33
 - 离散量故障, 3-42
 - 错误代码, B-6
 - 外部块, 3-47
 - 外部 I/O 模板, 3-46
 - 故障信息, 3-38
 - 故障说明, 3-38
 - 故障说明, 3-38
 - 故障处理, 3-10
 - 故障类型, 3-38
 - 故障类型, 3-38
 - 强制和解除强制回路, 3-50
 - GENA故障, 3-45
 - I/O 总线故障, 3-48
 - I/O故障组, B-20
 - I/O 故障表, 3-14
 - I/O 故障表解释, 3-38
 - I/O 故障表满, 3-29
 - 解释故障表, B-1
 - IOC 硬件失败, 3-50
 - IOC 软件故障, 3-49
 - 块未接通, 3-47
 - I/O 模板未接通, 3-46
 - IOC未接通, 3-45
 - 可选模块未接通或丢失, 3-18
 - 机架未接通或丢失, 3-17
 - 电池低电量信号, 3-28
 - 低等模拟量故障, 3-44
 - 微循环扫描超时, 3-29
 - 模板故障, 3-49
 - 模板硬件故障, 3-26
 - 上电时无用户程序, 3-32
 - 不可配置故障, 3-31
 - RUN方式无系统配置, 3-34
 - 可选模块软件故障, 3-27
 - 密码输入错误, 3-34
 - PLC CPU 硬件故障, 3-26
 - PLC CPU 系统软件故障, 3-35
 - PLC 故障组, B-3
 - PLC 故障表解释, 3-16
 - PLC 系统故障表满, 3-29
 - 程序块检测失败, 3-28
 - 程序微循环扫描超时, 3-29
 - 重启, 添加, 或多余可选模块, 3-21
 - 运行模式存储失败, 3-37
 - 系统总线错误, 3-25
 - 系统总线失败, 3-32
 - 系统配置不符, 3-22
 - 总线控制器过多, 3-36
 - 窗口完成失败, 3-34
- 故障组, B-3, B-20
- 故障处理, 3-10
 - 报警处理器, 3-10
 - 用户自定义故障信息, 12-62
- 故障定位变量, 3-7
- 故障变量
 - 报警接点, 3-8
 - 可配置故障作用, 3-4
 - 故障接点, 3-6
 - 故障定位变量, 3-7
 - 不可配置故障, 3-5
 - 点故障, 3-9
- 故障响应, 3-12
- 故障表
 - 故障表解释, B-1
- 故障类型, B-23
- 故障
 - 获取附加故障信息, 3-15
 - 添加块, 3-47
 - 添加 I/O 模板, 3-46
 - 添加IOC, 3-45
 - 添加额外机架, 3-21
 - 模拟量故障, 3-43
 - 应用程序故障, 3-30
 - 块开关, 3-50
 - 回路故障, 3-41
 - 故障等级, 3-11
 - 存储期间通讯失败, 3-36
 - 可配置故障, 3-17
 - 恒定时间扫描超时, 3-29
 - 上电时用户程序损坏, 3-33
 - 离散量故障, 3-42
 - 显示用户自定义故障, 3-13
 - 错误代码, B-6
 - 解释和纠正, 3-1
 - 额外块, 3-47
 - 额外 I/O 模板, 3-46
 - 故障作用, 3-12
 - 故障属性, 3-11
 - 故障说明, B-24

故障处理, 3-10
 故障响应, 3-12
 强制和解除强制回路, 3-50
 GENA故障, 3-45
 I/O bus故障, 3-48
 I/O故障作用, B-21
 I/O故障信息, B-22
 I/O故障组, B-20
 I/O故障表, 3-14
 I/O故障表解释, 3-38
 I/O故障表满, 3-29
 I/O故障类型, B-23
 解释故障表, B-1
 IOC硬件故障, 3-50
 IOC软件故障, 3-49
 块未接通, 3-47
 I/O 模板未接通, 3-46
 IOC未接通, 3-45
 可选模块未接通或丢失, 3-18
 机架未接通或丢失, 3-17
 电池低电量信号, 3-28
 低等模拟量故障, 3-44
 微循环扫描超时, 3-29
 模板故障, 3-49
 模板硬件故障, 3-26
 上电时无用户程序, 3-32
 不可配置故障, 3-31
 RUN模式无系统配置, 3-34
 可选模块软件故障, 3-27
 密码错误, 3-34
 PLC CPU 硬件故障, 3-26
 PLC CPU 系统软件故障, 3-35
 PLC故障作用, B-5
 PLC故障组, B-3
 PLC故障表解释, 3-16
 PLC系统故障表满, 3-29
 程序块检测失败, 3-28
 重启, 添加, 或多余可选模块, 3-21
 运行模式存储失败, 3-37
 系统总线错误, 3-25
 系统总线失败, 3-32
 系统配置不符, 3-22
 系统对故障的反应, 3-11
 总线控制器过多, 3-36
 用户自定义, 2-23, 3-13, 12-62
 窗口完成失败, 3-34
 FIFO读功能块, 10-11
 FIFO 写功能块, 10-13
 FIFORD, 10-11
 FIFOWRT, 10-13
 FIP I/O, 2-85
 默认情况, 2-86
 FIP I/O 总线配置, 2-85
 FIP I/O 数据映射, 2-86
 FIP I/O 扫描时间
 工作表, A-23
 浮点数, F-1

输入和显示浮点数, F-4
 浮点数及其运算中的错误, F-5
 浮点数的内部格式, F-2
 浮点数的值, F-3
 FOR, 12-21
 For 指令, 12-21
 FST_SCN, 2-21

G

GE, 7-2
 Genius 全局数据, 2-15
 Genius I/O, 2-83, 2-84
 模拟量组块, 2-84
 默认状态, 2-84
 诊断数据收集, 2-86
 Genius I/O 总线配置, 2-83
 Genius I/O 数据映射, 2-83
 低等模拟量块, 2-84
 Genius I/O 扫描时间
 工作表, A-20
 全局数据
 Genius, 2-15
 全局数据通讯, 2-85
 微循环扫描中的全局数据, 2-64
 全局数据变量 (%G), 2-13
 大于功能块, 7-2
 大于等于功能块, 7-2
 GT, 7-2

H

高报警接点, 4-7
 内务处理, 2-4

I

I/O 数据映射, 2-83
 默认状态, 2-83
 FIP I/O 数据映射, 2-86
 Genius I/O 数据映射, 2-83
 I/O 故障扫描时间, A-14
 I/O 故障表, 3-14, B-18
 块, B-20
 解释, 3-38
 故障作用, B-21
 故障地址, B-19
 故障信息, B-22
 故障说明, B-24
 故障组, B-20
 故障特定数据, B-25
 故障时间标记, B-28
 故障类型, B-23
 I/O总线, B-19

- 解释故障表, B-1
- 长/短指示器, B-18
- 点, B-20
- 机架, B-19
- 参考地址, B-18
- 插槽, B-19
- 象征故障特性数据, B-26
- I/O中断运行和扫描时间, A-27
 - 工作表, A-28
- I/O 模板扫描时间
 - 工作表, A-15
- I/O 扫描时间, A-14
- I/O 扫描影响时间, A-18
- I/O 系统初始化, 2-75
- I/O 系统, 系列 90-70 PLC
 - 模拟量 I/O 诊断数据, 2-87
 - 离散量 I/O 诊断信息, 2-87
 - FIP I/O, 2-85
 - Genius I/O, 2-83
 - I/O 数据映射, 2-83
- I/O-触发中断程序, 2-68, 2-69
- I/O-触发程序, 2-56
- 间接变量寄存器地址,
 - 2-11
- @符号, 2-11
- 信息类故障, 3-4, 3-12
 - 强制和解除强制, 3-50
 - 上电时无用户程序, 3-32
 - 运行模式无系统配置, 3-34
 - 密码错误, 3-34
 - 窗口完成失败, 3-34
- 输入变量 (%I), 2-12
- 输入扫描, 2-4
- 指令助记符, C-1
- 指令定时, CPU, A-1
- 指令, 程序, 4-1
 - 指令助记符, C-1
- INT, 11-8
- 智能可选模块自检完成
 - 2-74
- 智能可选模块, A-26
 - 扫描时间, A-26
- 内部变量 (%M), 2-12
- 中断处理
 - 定时中断, 2-67
- 中断处理, 2-65
 - 独立程序-I/O-触发中断,
 - 2-68, 2-69
 - 触发中断块, 2-66
- 反余弦功能块, 6-10
- 反正弦功能块, 6-10
- 反正切功能块, 6-10
- IO_FULL, 2-21

J

- JUMP, 12-18
- 跳转功能块, 12-18

K

- 按键开关, 2-81

L

- LABEL, 12-19
- 标签功能, 12-19
- LE, 7-2
- 小于功能块, 7-2
- 小于等于功能块, 7-2
- LIFO 读功能块, 10-7
- LIFO 写功能块, 10-9
- LIFORD, 10-7
- LIFOWRT, 10-9
- 连线, 4-11
- LN, 6-12
- 局部寄存器变量 (%L), 2-11
- LOG, 6-12
- 对数功能块, 6-12
 - 以10为底对数, 6-12
 - 自然对数, 6-12
- 逻辑 AND 功能块, 8-3
- 逻辑 NOT 功能块, 8-7
- 逻辑 OR 功能块, 8-3
- 逻辑 XOR 功能块, 8-5
- 逻辑驱动动态以太网全局数据(SVCREQ #44), 12-74
- 低报警接点, 4-7
- LT, 7-2

M

- 维护, 3-1
- 映射, I/O data, 2-83
 - 默认状态, 2-83
 - FIP I/O 数据映射, 2-86
 - Genius I/O 数据映射, 2-83
- 屏蔽比较功能块, 8-20
- 主控制继电器功能块, 12-16
- 数学功能块
 - ABS, 6-8
 - ACOS, 6-10
 - ADD, 6-2
 - ASIN, 6-10
 - ATAN, 6-10
 - COS, 6-10
 - DEG, 6-14
 - DIV, 6-2

EXP, 6-12
EXPT, 6-12
LN, 6-12
LOG, 6-12
MOD, 6-4
MUL, 6-2
RAD, 6-14
SIN, 6-10
SQRT, 6-6
SUB, 6-2
TAN, 6-10
MCMP, 8-20
MCR, 12-16
内存
 掉电后数据内存保持性, 2-76
内存类型
 %I, %Q, %R, %AI, %AQ, 9-19
内存, 损坏, 3-16
微循环
 程序微循环时间超时故障, 3-29
 掉电周期时间, 2-72
微循环扫描方式, 2-10
微循环扫描时间, 2-72
MOD, 6-4
运行方式, 2-71
 运行/输出禁止, 2-71
 运行/输出允许, 2-71
 停止/IO扫描, 2-71
 停止/不扫描IO, 2-71
取模功能块, 6-4
MOVE, 9-2
移动功能块, 9-2
MUL, 6-2
乘法运算功能块, 6-2

N

自然对数功能块, 6-12
NE, 7-2
负线圈, 4-8
负向保持线圈, 4-8
正跳变线圈, 4-9
负跳变线圈, 4-4
嵌套, 2-32
无故障接点, 4-7
不可配置故障, 3-5, 3-31
一般扫描, 2-10
一般扫描方式
 应用程序任务执行, 2-4, 2-5
 编程器通讯窗口, 2-4, 2-6
 系统通讯窗口, 2-4, 2-7
常闭接点, 4-4
常开接点, 4-4
NOT, 8-7
不等功能块, 7-2

O

OEM 保护, 2-81
OFDT, 5-6
延时关断计时器, 5-6
延时接通计时器, 5-3
ONDTR, 5-3
PLC系统运行, 2-1
可选模块双端口测试, 2-74
可选模块自检完成, 2-74
OR, 8-3
顺序程序, 2-56
顺序时序模式, 2-56
输出变量 (%Q), 2-12
输出扫描, 2-5
管理扫描影响时间
 基本扫描时间, A-11
 估算扫描时间, A-31
FIP I/O扫描影响时间
 工作表, A-23
Genius I/O 扫描影响时间, A-18
 工作表, A-20
I/O中断运行及影响时间, A-27
I/O扫描影响时间
 工作表, A-15
I/O 扫描和 I/O 故障扫描, A-14
编程器扫描影响时间, A-12
FIP I/O 和 FBC影响扫描, A-21
Genius I/O 和 GBC扫描时间, A-17
可选智能模块扫描时间, A-26
系列 90-70 I/O模板扫描时间, A-14
强制, 2-16

P

参数子程序块, 2-35
参数子程序块
 如何调用参数子程序块, 2-36
 定义形式参数, 2-37
 使用形式参数的限制, 2-38
密码, 2-79
运行
 CPU图表, A-39
周期时序模式, 2-56
PID, 12-88
PLC故障表, B-2
 错误代码, B-6
 解释, 3-16
 故障作用, B-5
 故障附加数据, B-10
 故障组, B-3
 故障时间标记, B-14

- 故障表解释, B-1
- 长/短指示器, B-2
- 机架, B-2
- 插槽, B-2
- 空槽, B-2
- 任务, B-3
- PLC 内存确认, 2-73
- PLC 扫描周期, 2-2
 - 应用程序任务执行, 2-5
 - 编程器通讯窗口, 2-6
 - STOP 模式, 2-9
 - 系统通讯窗口, 2-7
- PLC扫描模式, 2-10
 - 恒定时间扫描, 2-10
 - 恒定窗口, 2-10
 - 微循环, 2-10
 - 常开扫描, 2-10
- PLC 系统运行, 2-1
- 点故障, 3-9
- 正跳变线圈, 4-9
- 正跳变接点, 4-4
- 激励e 功能块, 6-12
- 激励 X功能块, 6-12
- 掉电时序, 2-72, 2-73, 2-75
 - 微循环, 2-72
- 上电自检, 2-73
- 上电时序, 2-73
 - I/O 系统初始化, 2-75
 - 可选模板双端口测试, 2-74
 - 可选模板自检完成, 2-74
 - PLC内存确认, 2-73
 - 上电自检, 2-73
 - 系统配置, 2-74
- 优先级, 2-80
- 程序块, 2-29
 - 使用程序块举例, 2-30
 - 如何调用程序块, 2-33
 - 主程序块, 2-29
 - 嵌套, 2-32
 - 参数子程序块, 2-35
 - 程序块和局部数据, 2-34, 2-35
 - 用户自定义程序块, 2-35
- 程序块
 - 大小限制, 2-30
- 程序组织和用户数据
 - 用户变量, 2-11
- 程序组织和用户数据
 - 浮点数, F-1
- 程序寄存器变量 (%P), 2-11
- 程序时序, 2-57
- 程序结构
 - 如何调用程序块, 2-33
 - 如何调用参数子程序块, 2-36
 - I/O-触发中断程序块, 2-68, 2-69
 - 主程序块, 2-29
 - 参数子程序块, 2-35

- 程序块和局部数据, 2-34, 2-35
- 定时中断, 2-67
- 触发中断块, 2-66
- 用户自定义功能块, 2-35
- 编程器通讯窗口, 2-6
- 编程器扫描时间, A-12
- 编程指令, 4-1
 - 指令助记符, C-1
- 编程时序模式, 2-56, 2-57
 - I/O-触发, 2-56
 - 顺序, 2-56
 - 周期, 2-56
 - 定时, 2-56
- 比例积分微分调节器 (PID), 12-88
- 保护等级请求, 2-80
- PSB, 2-30

R

- RAD, 6-14
- 弧度转换功能块, 6-14
- REAL, 11-12
- 变量, 用户, 2-11
 - %G 变量和 CPU 内存, 2-15
 - 模拟量输入寄存器变量 (%AI), 2-11
 - 模拟量输出寄存器变量(%AQ), 2-11
 - 相关转变和强制, 2-16
 - 数据范围, 2-18
 - 数据类型, 2-19
 - 离散变量, 2-12
 - 全局数据变量 (%G), 2-13
 - 间接变量, 2-11
 - 输入变量 (%I), 2-12
 - 内部变量 (%M), 2-12
 - 局部寄存器变量 (%L), 2-11
 - 输出变量 (%Q), 2-12
 - 程序寄存器变量 (%P), 2-11
 - 寄存器变量, 2-11
 - 大小及默认值, 2-14
 - 系统故障变量, 3-3
 - 系统寄存器变量 (%R), 2-11
 - 系统状态变量, 2-20
 - 系统状态变量 (%S), 2-13
 - 系统状态/故障变量, 2-20
 - 临时变量 (%T), 2-12
- 寄存器变量, 2-11
 - 模拟量输入寄存器 (%AI), 2-11
 - 模拟量输出寄存器 (%AQ), 2-11
 - 全局数据变量 (%G), 2-13
 - 间接变量, 2-11
 - 输入变量 (%I), 2-12
 - 内部变量 (%M), 2-12
 - 局部寄存器变量(%L), 2-11
 - 输出变量(%Q), 2-12
 - 程序寄存器变量 (%P), 2-11
 - 大小和默认值, 2-14
 - 系统寄存器变量 (%R), 2-11
 - 系统状态变量 (%S), 2-13

临时变量 (%T), 2-12

关系功能块

CMP, 7-4

EQ, 7-2

GE, 7-2

GT, 7-2

LE, 7-2

LT, 7-2

NE, 7-2

继电器功能块

线圈, 4-8

线圈, 4-3

接点, 4-2

续行线圈, 4-12

续行接点, 4-12

故障接点, 4-7

高报警接点, 4-7

连接线, 4-11

低报警接点, 4-7

负线圈, 4-8

负保持线圈, 4-8

负跳变线圈, 4-9

负跳变接点, 4-4

无故障接点, 4-7

常闭线圈, 4-4

常开接点, 4-4

正跳变线圈, 4-9

正跳变接点, 4-4

复位线圈, 4-10

保持线圈, 4-8

保持型复位线圈, 4-11

保持型置位线圈, 4-11

置位线圈, 4-10

复位线圈, 4-10

掉电后数据内存的保持性, 2-76

保持线圈, 4-8

保持型复位线圈, 4-11

保持型置位线圈, 4-11

逻辑和数据的保持性, 2-16

RLD v. 独立 C 程序, 2-47

ROL, 8-12

ROR, 8-12

循环左移功能块, 8-12

循环右移功能块, 8-12

运行/停止操作, 2-71

运行/输出禁止, 2-71

运行/输出允许, 2-71

停止/IO扫描, 2-71

停止/无 IO扫描, 2-71

S

时序模式, 2-56, 2-57

I/O-触发, 2-56

顺序, 2-56

周期, 2-56

定时, 2-56

时序程序, 2-57

查找数列移动功能块, 10-17

查找相等功能块, 10-21

查找大于功能块, 10-21

查找大于或等于功能块, 10-21

查找小于功能块, 10-21

查找小于或等于功能块, 10-21

查找不等功能块, 10-21

安全性, 系统, 2-79

优先级, 2-80

自检

I/O 系统初始化, 2-75

可选模块双端口测试, 2-74

可选模块自检完成, 2-74

上电自检, 2-73

系列 90-70 PLC I/O系统, 2-82

模拟量 I/O 诊断数据, 2-87

离散量 I/O 诊断信息, 2-87

FIP I/O, 2-85

Genius I/O, 2-83

I/O 数据映射, 2-83

复位请求功能块, 12-25

置位线圈, 4-10

SHFR, 9-8

左移功能块, 8-9

移动寄存器功能块, 9-8

右移功能块, 8-9

SHL, 8-9

SHR, 8-9

SIN, 6-10

正弦功能块, 6-10

SORT, 10-15

分类功能块, 10-15

SQRT, 6-6

平方根功能块, 6-6

SRCH_EQ, 10-21

SRCH_GE, 10-21

SRCH_GT, 10-21

SRCH_LE, 10-21

SRCH_LT, 10-21

SRCH_NE, 10-21

独立 C 程序, 2-27, 2-42

数据封装, 2-42

内部数据, 2-42, 2-44, 2-45, 2-47

独立 C v. RLD 程序, 2-47

STOP 模式, 2-9

SUB, 6-2

减法功能块, 6-2

SUSIO, 12-14

延迟 I/O 功能块, 12-14

SVCREQ, 12-25

更改后台任务窗口状态和值(#5), 12-34

更改编程器通讯窗口(#3), 12-32
 更改系统通讯窗口 (#4), 12-33
 更改或读取检测任务状态及检测字数 (#6), 12-36
 更改或读取恒定扫描时间 (#1), 12-28
 更改或读取日历时钟 (#7), 12-38
 清除故障表 (#14), 12-48
 禁用/启用 EXE块检测 (#25), 12-67
 ESCM 端口状态 (#39), 12-72
 逻辑驱动动态以太网全局数据 (#44), 12-74
 屏蔽/解除屏蔽 I/O中断(#17), 12-54
 屏蔽/解除屏蔽定时中断(#22), 12-64
 读取历史时钟 (#16), 12-53
 读取故障表 (#20), 12-58
 读取文件夹名 (#10), 12-44
 从反向区域读取数据 (#28), 12-69
 读I/O强制状态 (#18), 12-56
 读最新故障表条目 (#15), 12-49
 读取主检测 (#23), 12-65
 读取PLC ID (#11), 12-45
 读取 PLC运行状态 (#12), 12-46
 读取扫描时间 (#9), 12-43
 读取窗口值 (#2), 12-31
 重启看门狗定时器 (#8), 12-42
 任务开关 (#26), 12-68
 设置运行使能/禁止 (#19), 12-57
 停止(stop) PLC (#13), 12-47
 延迟 I/O 中断 (#32), 12-70
 用户自定义故障信息(#21), 12-62
 反向区域写入 (#27), 12-69
 SWAP, 9-15
 交换功能块, 9-15
 扫描影响
 以太网全局数据, A-24
 扫描, PLC, 2-2
 应用程序任务执行, 2-5
 编程器通讯窗口, 2-6
 STOP模式, 2-9
 系统通讯窗口, 2-7
 SYS_FULL, 2-21
 系统通讯窗口, 2-7
 系统配置, 2-74
 系统故障变量, 3-3
 硬件故障, 3-3
 其他故障, 3-3
 软件故障, 3-3
 系统运行, 2-1
 时钟和定时器, 2-77
 密码, 2-79
 掉电时序, 2-73, 2-75
 掉电时序和微循环模式, 2-72
 上电时序, 2-73
 P节 1: 基本 PLC 扫描综述LC
 扫描总结

掉电后的内存数据保持性, 2-76
 系列 90-70 PLC I/O 系统, 2-82
 系统安全性, 2-79
 系统寄存器变量 (%R), 2-11
 系统状态变量, 2-20
 使用独立 C程序的不同, 2-48
 系统状态变量 (%S), 2-13, 2-20
 系统状态/故障变量, 2-20
 系统状态变量, 2-21

T

T_100MS, 2-21
 T_10MS, 2-21
 T_MIN, 2-21
 T_SEC, 2-21
 TABLE RANGE, 7-6, 10-24
 表范围功能块, 7-6, 10-24
 表读取功能块, 10-3
 表写入功能块, 10-5
 TAN, 6-10
 正切功能块, 6-10
 TBLRD, 10-3
 TBLWRT, 10-5
 临时寄存器 (%T), 2-12
 定时中断, 2-67, 12-64
 定时时序模式, 2-56
 日历时钟, 2-77, 12-38
 使用SVCREQ 功能块 #16 读取时钟, 2-77
 使用 SVCREQ 功能块 #7 t读取和设定时钟, 2-77
 计时器, 2-77
 功能块数据, 5-1
 OFDT, 5-6
 ONDTR, 5-3
 TMR, 5-9
 看门狗计数器, 2-78
 使用 SVCREQ功能块 #8重启计时器, 2-78
 定时, 指令, A-1
 TMR, 5-9
 转换, 2-16
 触发中断块, 2-66
 发现和解决故障, 3-1
 获取附加故障信息, 3-15
 可配置故障, 3-17
 故障处理, 3-10
 I/O 故障表, 3-14
 I/O 故障表解释, 3-38
 故障表解释, B-1
 不可配置故障, 3-31
 PLC故障表解释, 3-16
 发现和解决故障类别, 3-38