



警告

应该由合格的安装人员进行安装，并且
安装要符合所有国家法规和地方法规



Hi5a 控制器功能说明书

内置 PLC





本手册内的信息为 Hyundai Robotics 所有。
未经 Hyundai Robotics 书面授权、不得复制全部或部分内容。
本手册不得提供给第三方、不得用于其它用途。

Hyundai Robotics 保留不经过事先通知而修改本手册的权利。

韩国语印刷 - 2023 年 4 月、第 10 版
Hyundai Robotics Co., Ltd. 版权所有© 2023

地址:北京市朝阳区望京东路 8 号 锐创国际中心 A 座 1101 室
电话:010 8417-7788
主页 :www.hyundai-robotics.com





目 录

1. 简要	1-1
2. 控制器设置	2-1
2.1. 内置 PLC 的模式设置	2-2
2.2. 在控制器 TP 中监测 Relay 状态	2-4
2.3. 内藏 Scan Time	2-4
2.4. 内置 PLC 启动分配时间调整	2-4
3. 输入出 Diagram	3-1
4. Relay 规格	4-1
4.1. Relay 分数	4-2
4.2. Relay 说明	4-4
4.2.1. SP Relay	4-5
4.2.2. SW Relay	4-7
4.2.3. SW40~59 Relay - 以 16bit 读取机器人当前姿势	4-20
4.2.4. SW40~59 Relay - 以 32bit 读取机器人当前姿势	4-21
4.2.5. SW220~249 Relay - 系统变量	4-22
4.2.6. SW220~249 Relay - 主板存储空间	4-23
4.2.7. SW220~249 Relay - 模拟输入输出	4-24
4.2.8. SW220~249 Relay - 日期/时间	4-25
4.2.9. SW220~249 Relay - GE 变量	4-26
4.3. 继电器间接地址指定	4-27
4.4. 计时器&计数器 Relay	4-28
5. 命令语说明	5-1
5.1. 命令语表	5-2
5.2. 可使用的 Operands	5-5
5.3. Ladder Diagram 与 Mnemonic 的关系	5-8
6. 命令语详细说明	6-1
6.1. XIC(Examine if Closed):Close 检查	6-2
6.2. XIO(Examine if Open):Open 检查	6-2
6.3. INV(Inverting):反转	6-3
6.4. EQU(Equal):检查是否相同	6-3

6. 5. NEQ(Not Equal):检查是否不同.....	6-4
6. 6. LES(Less Than):检查是否小于基准值.....	6-4
6. 7. GRT(Greater Than):检查是否大于基准值.....	6-5
6. 8. LEQ(Less Than or Equal):检查是否比基准值小或相同	6-5
6. 9. GEQ(Greater Than or Equal):检查是否比基准值大或相同	6-6
6. 10. OTE(Output Energize):一般输出.....	6-6
6. 11. OTL(Output Latch):Latch 输出	6-7
6. 12. OTU(Output Unlatch):Unlatch 输出.....	6-7
6. 13. OSR(One Shot Rising):One Shot 输出.....	6-8
6. 14. RES(Reset):复位	6-8
6. 15. TON(Time On Delay):计时器.....	6-9
6. 16. CTD(Count Down):计数器.....	6-9
6. 17. ADD(Add):加	6-10
6. 18. SUB(Subtract):减	6-11
6. 19. MUL(Multiply):乘	6-11
6. 20. DIV(Divide):除	6-12
6. 21. POW(Power):累乘	6-12
6. 22. TOD(Convert to BCD):变换为 BCD 值.....	6-13
6. 23. FRD(Convert form BCD to Integer):变换为 integer.....	6-14
6. 24. SEG(7' Segment):变换为 7' Segment 值	6-15
6. 25. MOV(Move):移动	6-16
6. 26. COP(Copy data):Copy	6-17
6. 27. CCOP(Conditional Copy data):按条件复制	6-18
6. 28. ROT(Rotating Output):按序输出.....	6-19
6. 29. FOR(FOR):Block 反复.....	6-20
6. 30. NEXT(NEXT):Block Next.....	6-21
6. 31. LBL(Label):Label 指定	6-21
6. 32. JMP(Jump):Jump	6-22
6. 33. CALL(Call):Sub Ladder 呼叫	6-23
6. 34. END(End):关闭 Ladder	6-23

图纸目录

图 1.1 Hi5a I/O Connection.....	1-2
图 2.1 内置 PLC 模式设置.....	2-2
图 2.2 内置 PLC Off 状态	2-2
图 2.3 内置 PLC STOP 状态.....	2-2
图 2.4 内置 PLC 启动时间设置	2-4
图 3.1 输入出 Diagram.....	3-2

表格目录

表 4-1 Relay 分数.....	4-2
表 5-1 命令语表.....	5-2
表 5-2 可使用的 Operands.....	5-5







HD

HYUNDAI
ROBOTICS

1

概要



1. 简要

Hi5a 控制器的内置 PLC 是将常用 PLC 的功能体现到控制器的功能。如下图所示、在连接于控制器的 PC 或笔记本电脑启动 HRLadder 后编制/编辑 Ladder 程序、并下载到控制器或上传到控制器的 Ladder 程序、还可监测在控制器启动的程序状况。关于 HRLadder 请参考另外的功能说明书。

通过上级工艺或者现场总线或者实时以太网可以连接 Hi5a 控制器的 I/O。利用内置的 DeviceNet 主机可以连接 DeviceNet 从属模块、利用 BD580 或者 BD58A、可通过物理信号连接诸多装置。如上所述、内置 PLC 可通过 Ladder Logic 控制已连接的输入输出信号。

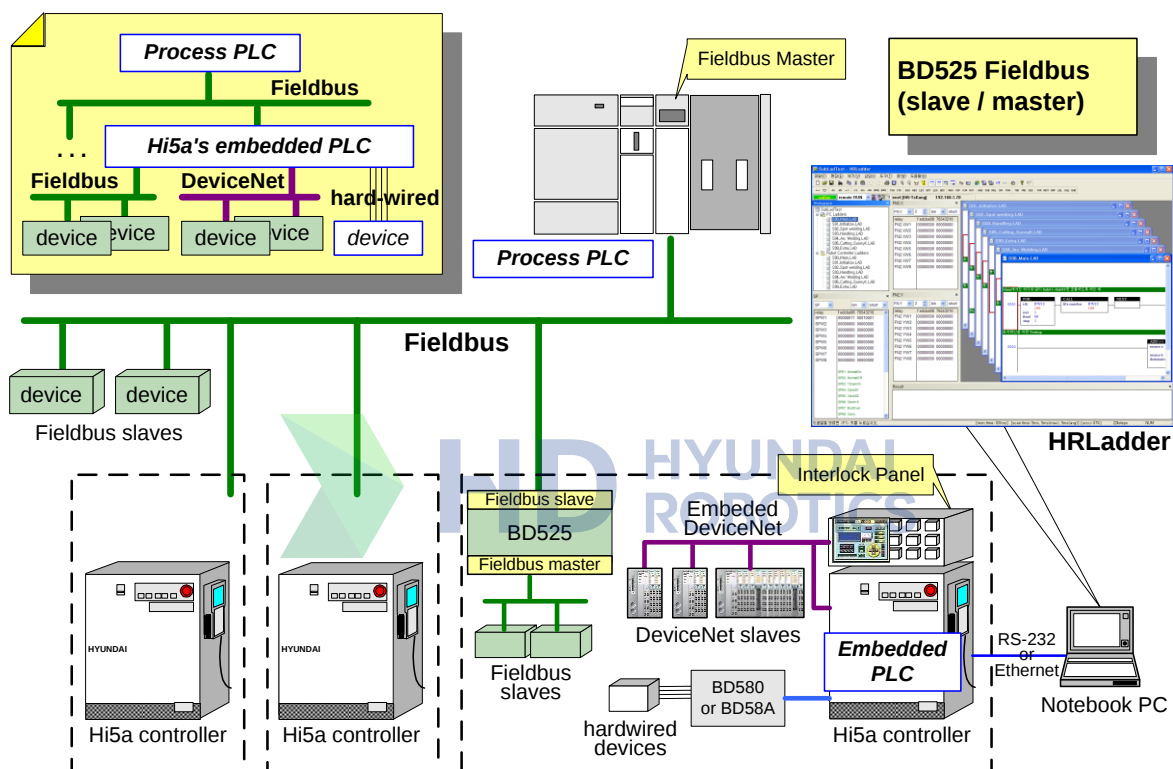


图 1.1 Hi5a I/O Connection



HD

HYUNDAI
ROBOTICS

2

控制器设置



2. 控制器设置

内置 PLC

2.1. 内置 PLC 的模式设置

内置 PLC 的工作模式、可在『[F7]:条件设置』→『[F1]:应用条件』→『5:内置 PLC 模式』的 <关闭、停止、R-停止、R-运行、运行>中选一设置。

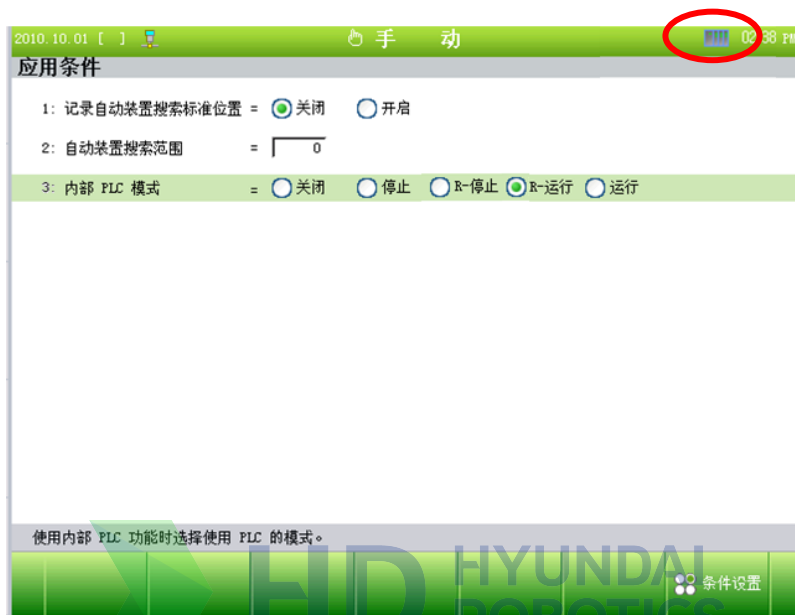


图 2.1 内置 PLC 模式设置

根据所选模式、在 TP 画面右上方显示图标。如上图所示、如果是[PLC=<R-运行>]或[PLC=<运行>]状态、就会显示 PLC 图标。如下图所示、[PLC=<关闭>]时图标会消失、[PLC=<停止>]状态时会在 PLC 图显示红色禁止标记。



图 2.2 内置 PLC Off 状态



图 2.3 内置 PLC STOP 状态

关闭

内置 PLC 的所有功能都会处于无效状态。这时、机器人控制器的数码输出(Digital Output)- D01~D04096 会自动输出物理性输出(Physical Output)-Y1~Y4096、并且物理性输入 X1~X4096 会自动输入 DI1~DI4096。这时各适用 DI、DO、X 和 Y 的属性。

例如、D01 为副逻辑、Y1 也属副逻辑时如果用 D01=1 输出、D01 会呈副逻辑、0 被输入为 Y1、Y1 的属性会呈副逻辑、因此实际的输出值为 Y1=1。

R-停止 / 停止

停止内置 PLC 的工作。R-停止 显示在 HRLadder 可以变更的 Remote 状态。如果设为 STOP、就不能在 HRLadder 变更工作模式。内置 PLC 被 Stop 时、PLC 的输出信号 DI、Y、R Relay 会自动变为 0。但、除 PLC 以外的 HRBasic 或分配中、Y 输出值为“SP11=On”时、Y 输出就会保持之前值。

R-运行 / 运行

启动内置 PLC。R-运行 显示可在 HRLadder 变更的 Remote 状态、如果设为 Run、就不能在 HRLadder 变更工作模式。



2.2. 在控制器 TP 中监测 Relay 状态

可在『[F1]:服务』 → 『1:监测』 → 『6:PLC Relay 数据』 → 『1:PLC X 继电器 (外部输入)』 → 『11:PLC SW 继电器 (系统存储器)』监测 Relay 状态。

2.3. 内藏 Scan Time

在 HRLadder 的下方状态栏中显示“scan time”、因 Step 数增加而不能在 5msec 内处理时、scan time 会自动增加 (按 5msec 单位)。

2.4. 内置 PLC 启动分配时间调整

在特殊用途中需要加快内置 PLC Scan 时间时、请向本公司工程师申请、以便在 Hi5a 机器人调整 PLC 所需的分配时间。在手动模式的『[F2]:系统』 → 『2:控制参数』 → 『1:控制环境设置』 → 『7:PLC 执行时间设置』中具有调整分配时间的菜单。PLC 启动时间设置是指在机器人控制器应处理的作业中设置 PLC 处理时间、默认值为 2.0/20[ms]、如果设置值过大、就不能处理机器人移动所需的动作计算等、必须向本公司工程师提出申请、确认要变更的系统的残余资源后、以此为基础计算分配时间。



图 2.4 内置 PLC 启动时间设置



HD

HYUNDAI
ROBOTICS

3

输入出 Diag
ram



3. 输入出 Diagram

内置 PLC

Hi5a 机器人控制器的输入出 Diagram 如下图所示。

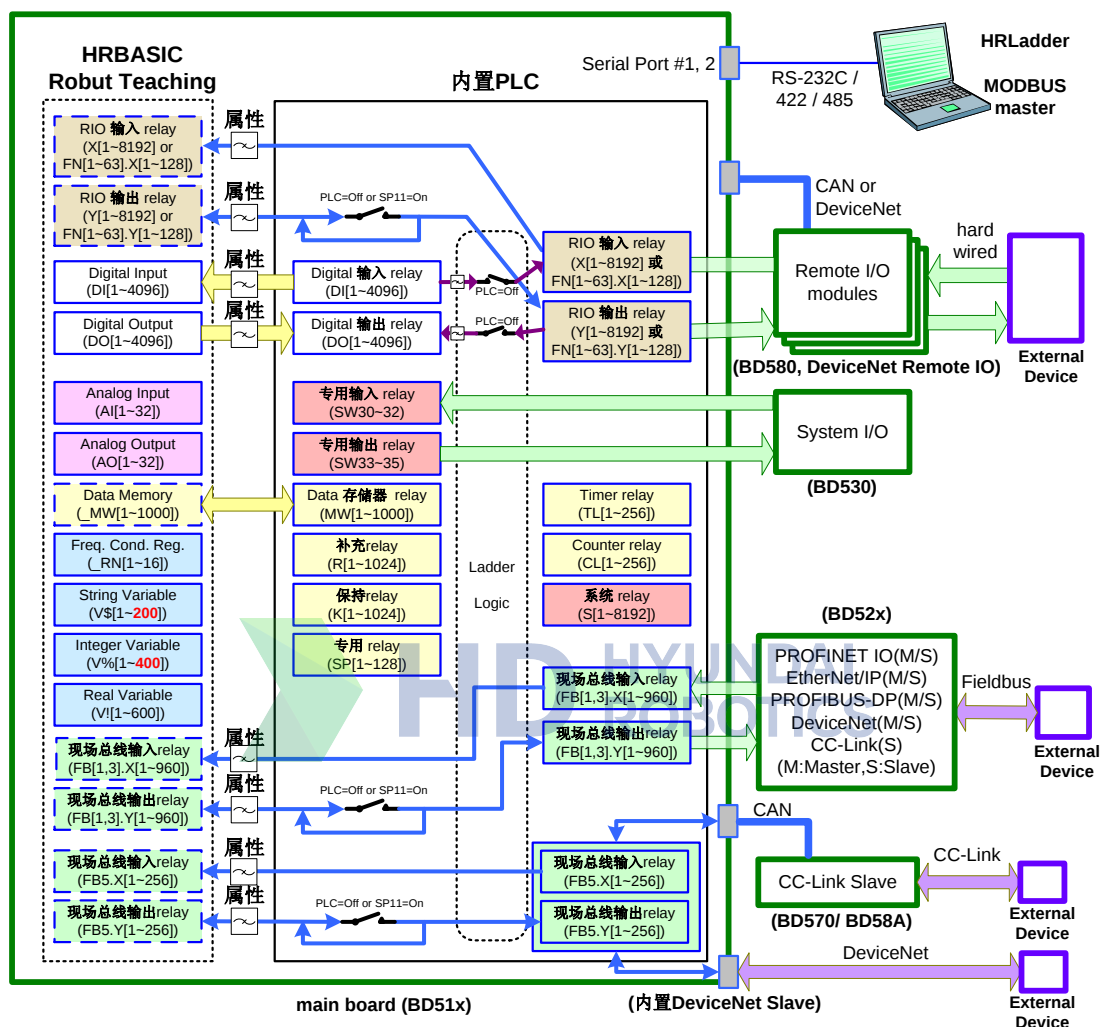


图 3.1 输入出 Diagram

即使 PLC 正在使用、只要让 SP11 打开、就可以在作业程序 (HRBASIC) 中、直接控制 RIO 输出 (Y [1~8192]、FN[1~63].Y[1~128])、以及现场总线输出 (FB[1, 3, 5].Y[1~960]) 继电器。

不过对于输入继电器 (X[1~8192]、FN[1~63].X[1~128]、FB[1, 3, 5].X[1~960]) 来讲、不管 PLC 是否使用、都可随时在机器人作业程序 (HRBASIC) 中读到。

上图的系统存储器 (System Memory) 是为特殊用途预约 (reserved) 的装置、今后会根据控制器版本变更用途。



HD

HYUNDAI
ROBOTICS

4

Relay 规格



4. Relay 规格

内置 PLC

4.1. Relay 分数

表 4-1 Relay 分数

Relay 名称	分数	Relay (1bit)	Relay (Byte、Word)	(Long、Float)
数码输入出 Relay	输入 4096 分	DI1~4096	DIB1~512	DIL1~128
			DIW1~256	DIF1~128
	输出 4096 分	DO1~4096	DOB1~512	DOL1~128
			DOW1~256	DOF1~128
RIO 输入出 Relay (BD580/DeviceNet)	输入 8192 分 每 Node 为 128 分 (Max 64Node)	FN1~64. X1~128 (或 X1~8192)	FN1~64. XB1~16 (or XB1~1024)	FN1~64. XL1~4 (or XL1~256)
			FN1~64. XW1~8 (or XW1~512)	FN1~64. XF1~4 (or XF1~256)
	输出 8192 分 每 Node 为 128 分 (Max 64Node)	FN1~64. Y1~128 (或 Y1~8192)	FN1~64. YB1~16 (or YB1~1024)	FN1~64. YL1~4 (or YL1~256)
			FN1~64. YW1~8 (or YW1~512)	FN1~64. YF1~4 (or YF1~256)
Fieldbus Relay (PROFINET IO, EtherNet/IP, CC-Link IE, PROFIBUS-DP, DeviceNet)	输入 960 分	FB1. X1~960	FB1. XB1~120	FB1. XL1~30
			FB1. XW1~60	FB1. XF1~30
	输出 960 分	FB1. Y1~960	FB1. YB1~120	FB1. YL1~30
			FB1. YW1~60	FB1. YF1~30
	输入 960 分	FB3. X1~960	FB3. XB1~120	FB3. XL1~30
			FB3. XW1~60	FB3. XF1~30
	输出 960 分	FB3. Y1~960	FB3. YB1~120	FB3. YL1~30
			FB3. YW1~60	FB3. YF1~30
Fieldbus Relay (CC-Link, CC-Link IE)	输入 960 分	FB5. X1~960	FB5. XB1~120	FB5. XL1~30
			FB5. XW1~60	FB5. XF1~30
	输出 960 分	FB5. Y1~960	FB5. YB1~120	FB5. YL1~30

4. Relay 规格

Relay 名称	分数	Relay(1bit)	Relay(Byte、Word)	(Long、Float)
			FB5. YW1 ~ 60	FB5. YF1 ~ 30
辅助 Relay	1024 分	R1~1024	RB1~128	RL1~32
			RW1~64	RF1~32
保存 Relay	1024 分	K1~1024	KB1~128	KL1~32
			KW1~64	KF1~32
特殊 Relay	128 分	SP1~128	SPB1~16	SPL1~4
			SPW1~8	SPF1~4
计时 Relay	256 分	T1~256	TB1~1024	TL1~256
			TW1~512	TF1~256
计数 Relay	256 分	C1~256	CB1~1024	CL1~256
			CW1~512	CF1~256
数据存储器	16000 分	M1~16000	MB1~2000	ML1~500
			MW1~1000	MF1~500
系统存储器	8192 分 (Reserved)	S1~8192 (Reserved)	SB1~1024	SL1~256
			SW1~512	SF1~256

参考) Relay 形式:Byte=8bit、 Word=16bit、 Long=32bit、 Float=(32bit)

4.2. Relay 说明

各 Relay 使用首字母(B/W/L/F)、可以 bit、byte、word、long、float 形式指定。但、计时器和计数器的 bit(T/C)在 long(TL/CL)值为 0 时表示激活状态(activation)、非为 0 时表示非激活状态。

- (1) 数码输入出 Relay:可在 HRBasic 或各种输入出分配中使用的信号。
- (2) RIO 输入出 Relay:
BD580 电路板的输入输出信号,或者是内置 DeviceNet 主机连接的 DeviceNet 主机从属节点的输入输出信号。
- (3) Fieldbus Relay:
通过 BD521, BD522, BD525, BD570, BD58A 电路板连接的现场总线或者实时以太网的输入输出信号。
- (4) 辅助 Relay:在 PLC 程序中使用的辅助 Relay。
- (5) 保存 Relay:即使关闭电源、也会保存 On/Off 状态的 Relay。
- (6) 特殊 Relay:为特殊目的而定义的 Relay。请参考 4.2.1 节。
- (7) 计时器 Relay:
进行计时器工作的 Relay、设置值为 0 时接点会 On。(停电时保存)
- (8) 计数器 Relay:
进行计数器工作的 Relay、设置值为 0 时接点会 On。(停电时保存)
- (9) 数据存储器:
保存或读取应用命令的任一数据时使用、在 HRBasic 也可 access、与 HABasic 收发大量数据时可以使用。(停电时保存)
- (10) 系统存储器(Reserved):用于读取或写入控制器内部的系统值。请参考 4.2.2 节。

4.2.1. SP Relay

Relay 编号	说明	其他
SP01	时常 On Relay	Controller states
SP02	时常 Off Relay	
SP03	在开始运行后的第一个 Scan 时启动的 Relay	Internal timer
SP04	0.1 秒 clock (0.05 秒 On → 0.05 秒 Off)	
SP05	0.2 秒 clock (0.1 秒 On → 0.1 秒 Off)	
SP06	1 秒 clock (0.5 秒 On → 0.5 秒 Off)	
SP07	不能在 BCD 计算变换时启动。.	启动 TOD 或 FRD 时
SP08	计算结果有 carry 时 On.	执行算数命令时
SP09	2 秒 clock (1 秒 On → 1 秒 Off)	Internal timer
SP10	4 秒 clock (2 秒 On → 2 秒 Off)	
SP11	如果是 On、就在除 PLC 以外的应用 (HRBasic、分配) 中允许直接向 YRelay 输出。	
SP12	Label 非为常数时 On	
SP13	Label 的数量为 100 以上时 On	
SP14	Label 重复时 On	
SP15	Label 为 0 以下或没有可 Jump 的 Label 编号时	
SP16	在 Simulation 状态中进行 Modbus Test	
SP17	Scan Time 超过 5 秒钟时 On (这时处于 5 秒之后的命令不会被启动。)	
SP18	没有通过 Call 呼叫的 Subladder 时 On	
SP19	0 = 内置 Devicenet 主机错误/警告激活 (基本值) 1 = 内置 Devicenet 主机错误/警告省略	从 V40.06-00 版本开始支持
SP20	0 = FB1 通信渠道错误/警告激活 (基本值) 1 = FB1 通信渠道错误/警告省略	

Relay 编号	说明	其他
SP21	0 = FB3 通信渠道错误/警告激活（基本值） 1 = FB3 通信渠道错误/警告省略	
SP22	0 = FB5 通信渠道错误/警告激活（基本值） 1 = FB5 通信渠道错误/警告省略	
SP23~SP128	Reserved	



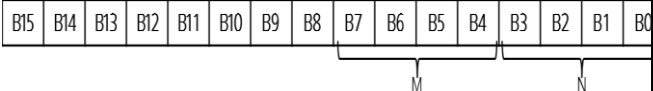
4.2.2. SW Relay

No.	说明	其它
SW3	PLC 启动模式 (4:PLC OFF、5:没有程序、0:STOP、1:R. STOP、2:R. RUN、3:RUN)	
SW4	扩张 IO Board 数	
SW5	Main SW Version 的 2 nd + 1 st	40.05-04v → &H05 28
SW6	I/O version + Main SW Version 的 3 rd	40.05-04 → &H43 04
SW10	Scan time	
SW11	分配时间	
SW12	最大占有时间	
SW13	平均占有时间	
SW14	Ladder 的总 Step 数	
SW15	Ladder 程序的 CHECKSUM	
SW17	占有率 (%)	
SW20	控制器的当前程序编号	
SW21	控制器的当前 Step 编号	
SW22	控制器的当前功能编号	
SW23	控制器的主程序编号	
SW24	FB1~FB4 active	0 th ~3 rd bit
SW25	读取 Hilscher 状态信息 (1~4) (1:FB1 信息、2:FB2 信息...)	
SW26	LSB:Hilsche COM GlobalBits MSB:reserved	(只有 Master 有效)

No.	说明	其它
SW27	LSB: 错误 Node 编号 MSB: 错误代码	(只有 Master 有效)
SW28	Bus 错误计数器	(只有 Master 有效)
SW29	DeviceNet: Bus OFF 计数器 Profibus-DP: Time Out 计数器	(只有 Master 有效)
SW30	专用输入 1	
SW31	专用输入 2	
SW32	专用输入 3	
SW33	专用输出 1	
SW34	专用输出 2	
SW35	专用输出 3	
SW37	LSB: 对指定的 node 编号、每一秒邀请 slave_diag	
SW38	对 slave_diag 邀请的回复值	
SW39	slave_diag 邀请结果的 Station Status LSB: StationStatus1 MSB: StationStatus2	
SW40~59	读取机器人当前人姿势 (pose)。请参考 4.2.3 节、4.2.4 节。	
SW60~69	指定间接地址地址领域 (4.3 间接地址指定参考)	-1: SW61 ~ -9: SW69
SW70	驱动时间类型 (0: 无、1: 驱动时间 (累计)、2: 总驱动时间 0) 注意) SW(n): MSW (上位)、SW(n+1): LSW	
SW71~SW72	检测时间 (10msec 单位)	
SW73~SW74	Cycle 时间 (10msec 单位)	
SW75	Cycle 数	

No.	说明	其它
SW76~SW77	焊接机 1 焊接时间(10msec 单位)	
SW78	焊接机 1 焊接次数	
SW79~SW80	焊接机 2 焊接时间(10msec 单位)	
SW81	焊接机 2 焊接次数	
SW82~SW83	焊接机 3 焊接时间(10msec 单位)	
SW84	焊接机 3 焊接次数	
SW85~SW86	焊接机 4 焊接时间(10msec 单位)	
SW87	焊接机 4 焊接次数	
SW88~SW89	Wait、DI 等待时间(10msec 单位)	
SW90~SW91	计时器等待时间(10msec 单位)	
SW70	驱动时间类型 (0:无、3:最近驱动时间、4:总驱动时间) 添加最近驱动时间、Swapping SW、以便使用 SL。 即、SW(奇数):LSW(下位)、SW(偶数):MSW	
SL36	检测时间(10msec 单位)	
SL37	Cycle 时间(10msec 单位)	
SL38	Cycle 数	
SL39	焊接机 1 焊接时间(10msec 单位)	
SL40	焊接机 1 焊接次数	
SL41	焊接机 2 焊接时间(10msec 单位)	
SL42	焊接机 2 焊接次数	
SL43	焊接机 3 焊接时间(10msec 单位)	

No.	说明	其它
SL44	焊接机 3 焊接次数	
SL45	焊接机 4 焊接时间(10msec 单位)	
SL46	焊接机 4 焊接次数	
SL47	Wait、DI 等待时间(10msec 单位)	
SL48	计时器 等待时间(10msec 单位)	10msec 单位
SW70	运行时间种类 (5:运行信息(电弧)) 添加最近运行时间、同时为了能够使用 SL、对 SW 进行 Swapping。 即 SW(奇数):LSW(下位)、SW(偶数):MSW	
SL36	测试时间(10msec 单位)	
SL37	循环时间(10msec 单位)	
SL38	循环次数	
SL39	电弧焊接进行时间(10msec 单位)、以 1 天的单位设置	
SL40	电弧焊接进行日期(1 天单位)	
SW81	ARC Retry 进行次数	
SW82	ARC Overlap 进行次数.1:Arc Off	
SW83	ARC Overlap 进行次数.2:Wire Off	
SW84	ARC Overlap 进行次数.3:Limit Over	
SW85	ARC Overlap 进行次数.4:Gas Off	
SW86	ARC Overlap 进行次数.5:Coolant Off	
SW87	ARC 自动熔敷清除 进行次数	
SL45	1 Cycle 期间电弧运行时间	

No.	说明	其它
SL46	Tip 使用时间	
SL47	Tip 使用时间限制值	
SW99	设置是否在切换模式时关闭电机(Off) (0:未处理, 1:转换手动<->自动时关闭电机, 2:切换模式时关闭电机)	
SW100~SW109	Program Count 相关 reserved	
SW110	各轴速度启用 (Enable) (SW110=0xKLMN, 16 进制)  N (B3~B0): 分辨率 (0x1=10 倍, 0x2=100 倍, 0x3=1000 倍) M: 其他选项 (B4:1=绝对值, B5:0=轴速度, 1=电机速度)	按比特 (Bit) 确定设定值后, 应将其转换为 10 进制后输入。 例) 当分辨率为 1000 倍, 无绝对值, 设置电机速度时, N = 0x3, M = 0x2 => 0x23 = 输入 “35”
SW111~SW126	各轴(或马达)的速度 (16 轴)	
SW130~SW133	TP KEYPAD 信息	
SW135	Get Applet	
SW136	Set Applet	
SW140	内置 Devicenet 主机错误从站列表 (0~15)	Bit0=1:Node 0 Error Bit15=1:Node 15 Error
SW141	内置 Devicenet 主机错误从站列表 (16~31)	Bit0=1:Node 16 Error Bit15=1:Node 31 Error
SW142	内置 Devicenet 主机错误从站列表 (32~47)	Bit0=1:Node 32 Error Bit15=1:Node 47 Error
SW143	内置 Devicenet 主机错误从站列表 (48~63)	Bit0=1:Node 48 Error Bit15=1:Node 63 Error
SW144	内置 Devicenet 主机状态*1)	

*1) 内置 Devicenet 主机状态

Bit 15~9 :Reserved

Bit 8 :初始化状态 (1=初始化进行中, 0=初始化结束)

Bit 4 :主机运行状态 (0=RUN, 1=IDLE (PLC STOP))

Bit 3~0 :Devicenet 状态 (0=Running, 1=Reset (Out-of-box), 2=Init (Out-of-box),
3=Reset (Normal), 4=Init (Normal), 5=Duplicated MAC ID)

No.	说明	其它
SW145	请求内置 Devicenet 主机连接从站状态信息时的电路板号码	
SW146	SW145 节点号的从站状态*2)	
SW150	Arc Welding Info	1=Enable(Digital Only)
SW151	Arc Welding Real Current	
SW152	Arc Welding Real Voltage	
SW153	Arc Welder Error1	
SW154	Arc Welder Error2	
SW155	Wire Feeding Speed	
SW156	Arc Welding SI 1~8 号 (下位 byte), SO 1~8 号 (上位 byte)	从 V40.04-00 版本开始支持
SW160	Devicenet Explicit message 请求标识 0x0001= 请求 Devicenet Explicit message 应答标识 0x001X=Explicit message 请求中 0x0000=Explicit message 处理结束 0x002X=Explicit message 超时 其他= 错误应答	结果 =>SW166 ~ SW179
SW161	Explicit message Request:Node ID	
SW162	Explicit message Request:Service	

Check, (6=Fault)

*2) 从站状态

Bit 15 :1=使用节点, 0=未使用节点

Bit 14 :生产商 (1=CREVIS, 0=其他)

Bit 13~12 :节点状态 (1=Standby, 2=Connection Fault, 3=Configuration Fault)

Bit 11 :0=IO mode(主机和 IO 数据交换状态), 1=非IO mode

Bit 9~0 :从站状态

0x000=Online, 0x001=Offline, 0x002=IO Connection Closed,

0x004=Input Size Error, 0x008=Output Size Error, 0x010=Vendor ID mismatch,

0x020=Device type mismatch, 0x040=Production code mismatch,

0x080=CCV mismatch, 0x100=IO size is too big, 0x200=No connection

No.	说明	其它
SW163	Explicit message Request:Class	
SW164	Explicit message Request:Instance	
SW165	Explicit message Request:Attribute	
SW166	Explicit message Request/Response:Size of Data	Max. 26
SW167~SW179	Explicit message Request/Response:Service Data (SB333~SB358)	Max. 26 bytes
SW180	获取 SPOT GUN 之磨损量所需的 GUN 编号 (0:无效、1~:GUN 编号)	
SW181	移动电极磨损量	
SW182	固定电极磨损量	
SW183	枪搜索状态	
~SW189	SPOT GUN reserved	
SW190	MON_AXIS_CTRL_OFF	每一个 Bit 为 1 轴
SW195	滚边加压力[N]	
SW200	轴别累计距离监测模式 (支持 dword 是 V31.11-00~、V32.01-00~)	0:Off、 1:read、2:write(word) 3:read、4:write(dword)
SW201~SW216	轴别累计距离监测 (1 轴~16 轴) dword 的话 SL101~SL108 (1~8 轴)	单位:km、rad 发生 Overflow、就都会成为 0。
SW220	Slot1 代码 (0:功能 Off)	[_IDX] 0-base:读取值 30000-base:设定值 如) 在 SW223 设置 352 的话、 以 SW230 的值 设置_SPDRATE 值
SW221	Slot1 数据低位字	
SW222	Slot1 数据高位字	
SW223~ SW224, 225	Slot2 代码 Slot2 数据低位字, 高位字	

No.	说明	其它
SW226~ SW227, 228	Slot3 代码 Slot3 数据低位字, 高位字	
SW229~ SW230, 231	Slot4 代码 Slot4 数据低位字, 高位字	
SW232~ SW233, 234	Slot5 代码 Slot5 数据低位字, 高位字	
SW235~ SW236, 237	Slot6 代码 Slot6 数据低位字, 高位字	
SW238~ SW239, 240	Slot7 代码 Slot7 数据低位字, 高位字	
SW241~ SW242, 243	Slot8 代码 Slot8 数据低位字, 高位字	
SW244~ SW245, 246	Slot9 代码 Slot9 数据低位字, 高位字	
SW247~ SW248, 249	Slot10 代码 Slot10 数据低位字, 高位字	
SW250	(子任务 1) 生成状态	
SW251	(子任务 1) 当前 程序号码	
SW252	(子任务 1) 当前步骤号码	
SW253	(子任务 1) 当前功能号码	
~SW259	为(子任务 1)进行 reserved	
SW260	(子任务 2) 生成状态	
SW261	(子任务 2) 当前程序号码	
SW262	(子任务 2) 当前步骤号码	
SW263	(子任务 2) 当前功能号码	
~SW269	为(子任务 2)进行 reserved	
SW270	(子任务 3) 生成状态	
SW271	(子任务 3) 当前程序号码	
SW272	(子任务 3) 当前步骤号码	

No.	说明	其它
SW273	(子任务 3) 当前功能号码	
~SW279	为(子任务 3) 进行 reserved	
SW280	(子任务 4) 生成状态	
SW281	(子任务 4) 当前程序号码	
SW282	(子任务 4) 当前步骤号码	
SW283	(子任务 4) 当前功能号码	
~SW289	为(子任务 4) 进行 reserved	
SW290	(子任务 5) 生成状态	
SW291	(子任务 5) 当前程序号码	
SW292	(子任务 5) 当前步骤号码	
SW293	(子任务 5) 当前功能号码	
~SW299	为(子任务 5) 进行 reserved	
SW300	(子任务 6) 生成状态	
SW301	(子任务 6) 当前程序号码	
SW302	(子任务 6) 当前步骤号码	
SW303	(子任务 6) 当前功能号码	
~SW309	为(子任务 6) 进行 reserved	
SW310	(子任务 7) 生成状态	
SW311	(子任务 7) 当前程序号码	
SW312	(子任务 7) 当前步骤号码	
SW313	(子任务 7) 当前功能号码	

No.	说明	其它
~SW319	为(子任务 7)进行 reserved	
SW320	当前机器人的工具号	
SW321	PRM 回生电力:当前值 (下位 16Bit)	SL161
SW322	PRM 回生电力:当前值 (上位 16Bit)	
SW323	PRM 回生电力:累计值 (下位 16Bit)	SL162
SW324	PRM 回生电力:累计值 (上位 16Bit)	
SW325	PRM 回生电力:5 分平均值 (下位 16Bit)	SL163
SW326	PRM 回生电力:5 分平均值 (上位 16Bit)	
~SW329	Reserved	
SW330	BD525 主:装置状态 *3)	
SW331	BD525 主:设定从站个数	0~65535
SW332 ~ SW339	BD525 主:设定从站清单	位清单
SW340	BD525 主:通信状态 (下位 8bit)*4) BD525 主:从站状态 (上位 8Bit)*5)	

*3) BD525 主:装置状态
 Bit 0:1=准备 (Ready)
 Bit 1:1=运转 (Running)
 Bit 2:1=通信 (Bus On)
 Bit 3:1=设定锁定 (Configuration locked)
 Bit 4:1=等待新的设定 (New Configuration)
 Bit 5:1=需要重置 (Restart required)
 Bit 6:1=重置激活 (Restart required Enable)
 Bit 7~31:Reserved

*4) BD525 主:通信状态
 0=未知, 1=未配置, 2=停止, 3=闲置, 4=操作

*5) BD525 主:从站状态
 0=未定义, 1=OK(无故障), 2=失败(一个以上的从站错误)

No.	说明	其它
SW341	BD525 主:动作从站个数	0~65535
SW342 ~ SW349	BD525 主:动作从站清单	位清单
SW350	BD525 主:通信错误累积次数	0~65535
SW351	BD525 主:错误从站个数	0~65535
SW352 ~ SW359	BD525 主:错误从站清单	位清单
SW360	CC-Link IE 链接及运行状态	参考 CC-Link IE 功能说明书
SW361	CC-Link IE 周期性通讯状态	参考 CC-Link IE 功能说明书
SW366	低位字节: <code>iot_converter_client</code> 高位字节: <code>iot_converter_server</code>	从 Hi5a V40.25-00 开始支持 ON:1, OFF:0
SW367	低位字节: <code>iot_connector</code> 高位字节: <code>iot_ftp</code>	
SW370 ~ SW385	各轴编码器温度 (J1~J16) 服务/监测/系统特性数据/电机编码器数据/监测各轴编码器温度当前值 参考) 间接地监测电机温度	从 Hi5a V40.15-00 开始支持 °C
SW388	主板温度 (单位: 0.1°C)	从 Hi5a V40.17-00 开始支持
SW389	主板的 CPU 温度 (单位: 0.1°C)	从 Hi5a V40.17-00 开始支持
SW390	控制器序列号 服务/系统诊断/监测系统版本界面上的序列号	从 Hi5a V40.15-00 开始支持
~SW394	Reserved	
SW395	监测机器人状态 (0:stop, 1:run, 2:wait)	从 Hi5a V40.15-00 开始支持
SW396	是否脱离路径 (1: 脱离路径, 0: 正常路径, <0: 无法判断)	从 Hi5a V40.19-00 开始支持
SW397	限制步骤前进/后退 (输入“1”: 限制, 输入其他数字: 正常运行)	
~SW399	Reserved	
SW400	各轴速度启用 (Enable) (SW110=0xKLMN, 16 进制)	从 Hi5a V40.15-00 开始支持

No.	说明	其它
	B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0 M N N (B3~B0): 分辨率 (0x1=10 倍, 0x2=100 倍, 0x3=1000 倍) M: 其他选项 (B4:1=绝对值, B5:0=轴速度, 1=电机速度) 参考) 相同于 SW110。若要同时查看两种选项, 可与 SW110 一同使用。	速度单位 - 直动轴: mm/sec - 旋转轴: deg/sec - 电机速度: RPM
SW401~	监测各轴 (或电机) 的速度 (16 个轴)	从 Hi5a V40.15-00 开始支持
SW420	服务/监测/系统特性数据/负载率数据 (1:I/Ir, 2:i/Ip, 3:连续负载率)	
SW421~ SW436	监测各轴 (或电机) 的负载率 (16 个轴)	单位: %
SW440	设置适用于润滑脂自动加注机的配件	从 Hi5a V40.18-00 开始支持
SW441~ SW456	按配件分类的润滑脂自动加注标准距离	从 Hi5a V40.18-00 开始支持 距离单位 - 直动轴: km - 旋转轴: rad
SW461~ SW476	分配到每个配件的轴号	从 Hi5a V40.19-00 开始支持
SW480	Low Grease 警告 (0: 正常, 1~16: 相关配件的加注机出现异常)	从 Hi5a V40.19-00 开始支持
SW481	Empty 警告 (0: 正常, 1~16: 相关配件的加注机出现异常)	从 Hi5a V40.19-00 开始支持
SW482	Overload 警告 (0: 正常, 1~16: 相关配件的加注机出现异常)	从 Hi5a V40.19-00 开始支持
SW483	FAN 异常信息监测 - 第 0 位 Bit :R-FAN 板信息 (0: 正常, 1: 异常) - 第 8 位 Bit :R-FAN 板信息 (0: 正常, 1: 异常) 15 8 0 예비7 예비6 예비5 예비4 예비3 예비2 예비1 C-FAN 예비7 예비6 예비5 예비4 예비3 예비2 예비1 R-FAN	从 Hi5a V40.22-00 开始支持
SW484	监测 LCD Handle 撞击传感器数据 - 按位 (Bit) 分类的信息 (0: 未检测, 1: 已检测) (Bit0 : LSR1, Bit1 : LSR2, Bit8 : RSR1, Bit9 : RSR2) 15 9 8 7 1 0 R-예비6 R-예비5 R-예비4 R-예비3 R-예비2 R-예비1 RSR2 RSR1 L-예비6 L-예비5 L-예비4 L-예비3 L-예비2 L-예비1 LSR2 LSR1 - 仅在使用撞击传感器时上传监测数据 (路线 : F2 系统 > 4: 应用参数 > 29: 手碰撞检测传感器设置)	从 Hi5a V40.26-00 开始支持

No.	说明	其它
SW485, SW486	SW485:错误, 警告编号 SW486:最高位 bit (0:错误, 1:警告) 第 1~15 位 bit:错误附加信息(轴号等)	从 Hi5a V40.25-00 开始 支持



4.2.3. SW40-59 Relay - 以 16bit 读取机器人当前姿势

No.		说明	其它
	SW40	以 16bit 读取机器人位置 (SW41~SW56) 0:无 1:当前 base 坐标值 (1mm 或 0.1deg) 2:指令 base 坐标值 (1mm 或 0.1deg) 3:当前轴坐标值 (0.1mm 或 0.1deg) 4:指令前坐标值 (0.1mm 或 0.1deg) 5:当前轴坐标值 (1mm 或 0.1deg) 6:指令前坐标值 (1mm 或 0.1deg)	
SL21	SW41	X 坐标值或 J1 轴值	
	SW42	Y 坐标值或 J2 轴值	
SL22	SW43	Z 坐标值或 J3 轴值	
	SW44	RX 坐标值或 J4 轴值	
SL23	SW45	RY 坐标值或 J5 轴值	
	SW46	RZ 坐标值或 J6 轴值	
SL24	SW47 ~52	T1~T6 轴值	

4.2.4. SW40-59 Relay - 以 32bit 读取机器人当前姿势

No.		说明	其它
	SW40	以 32bit 读取机器人位置 (SL21~SL29) 0: 无 ... 10: 当前 base 坐标值 + 附加 3 轴 20: 指令 base 坐标值 + 附加 3 轴 1m: 当前轴坐标值 (m 轴 ~ m+8 轴) 2m: 指令前坐标值 (m 轴 ~ m+8 轴)	(0.001mm or 0.001deg) 从 Hi5a V40.25-00 开始支持
SL21	SW41	X 坐标值	
	SW42	Jm+0 轴值	
SL22	SW43	Y 坐标值	
	SW44	Jm+1 轴值	
SL23	SW45	Z 坐标值	
	SW46	Jm+2 轴值	
SL24	SW47	RX 坐标值	
	SW48	Jm+3 轴值	
SL25	SW49	RY 坐标值	
	SW50	Jm+4 轴值	
SL26	SW51	RZ 坐标值	
	SW52	Jm+5 轴值	
SL27	SW53	T1 坐标值	
	SW54	Jm+6 轴值	
SL28	SW55	T2 坐标值	
	SW56	Jm+7 轴值	
SL29	SW57	T3 坐标值	
	SW58	Jm+8 轴值	

4.2.5. SW220~249 Relay - 系统变量

代码

10	SPDRATE ; 启动速度 (%)
20	JOGSPD ; STEP 前进/后退时最高速 (mm/sec)

读取/写入

读取	继续更新
写入	继续更新

示例(例如, 当使用 slot1 时)

SW220	10(dec)	读取 SPDRATE 值。85%
SW221	85(dec)	
SW222	-	
SW220	30020(dec)	JOGSPD=100 mm/sec
SW221	100	
SW222		

4.2.6. SW220~249 Relay - 主板存储空间

代码

210	低位字 Free Block 数量, 高位字 Total Block 数量 只能读取
-----	---



4.2.7. SW220~249 Relay - 模拟输入输出

代码

301~316	AI1 ~ AI16 (单位:0.01V)值:低位字 只能读取
351~366	A01 ~ A016 (单位:0.01V)值:低位字

示例(例如, 当使用 slot1 时)

SW220	302(dec)	
SW221	-925(dec)	读取 AI2 值。-9.25V
SW222	-	
SW220	30353(dec)	
SW221	380(dec)	A03=3.80V
SW222	-	

4.2.8. SW220~249 Relay - 日期/时间

代码

401	日期:高位字 YYYY(dec), 低位字 MMDD(dec)
402	时间(24 小时制) :高位字 00HH(dec), 低位字 MMSS(dec)

读取/写入

读取	Hi5a 控制器的当前日期/时间值以 Slot 数据继续更新。(周期约为每秒 1 次)
写入	当代码值从其他值变更为 30401 或 30402 时, 则以 Slot 数据输入的日期/时间值立即将设置在 Hi5a 控制器中。

示例(例如, 当使用 slot1、slot2 时)

SW220	401(dec)	Index: 读取日期 (0-based)
SW221	0523(dec)	2012 年 5 月 23 日
SW222	2012(dec)	
SW223	402(dec)	Index: 读取时间 (0-based)
SW224	3650(dec)	15 时 36 分 50 秒
SW225	15(dec)	
SW220	30401(dec)	Index: 日期设置 (30000-based)
SW221	0523(dec)	2012 年 5 月 23 日
SW222	2012(dec)	
SW223	30402(dec)	Index: 时间设置 (30000-based)
SW224	3650(dec)	15 时 36 分 50 秒
SW225	15(dec)	

4.2.9. SW220~249 Relay - GE 变量

代码

1101~1128	GE1 ~ GE128 GE 变量的 index 为代码 - 1000
-----------	--

读取/写入

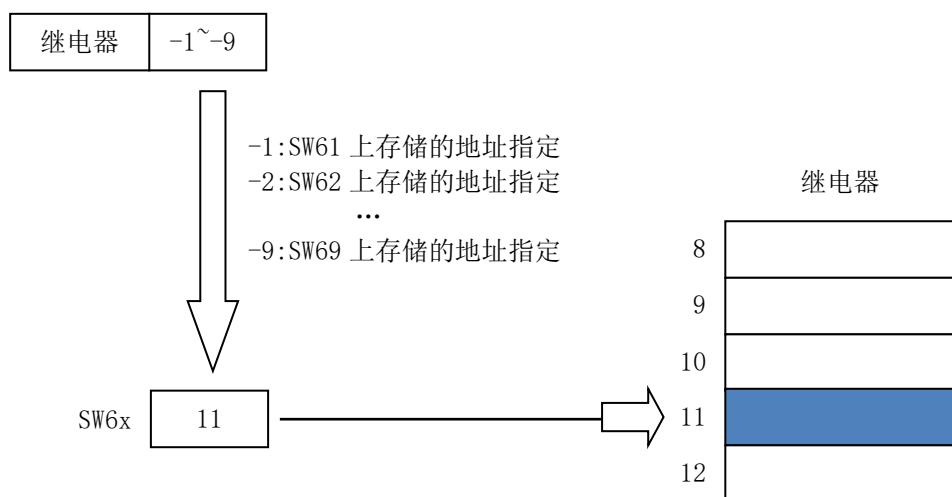
读取	继续更新 将 GE[index] ~ GE[index+3]的 4byte 读取为低位字、高位字。
写入	当系统变量 index 值变更时 (须为 31001~31128) 将已输入在数据低位字中的值转换为 (类型转换为) byte 并设置在 GE 变量中。

示例 (例如, 当使用 slot1、slot2 时)

SW220	31001 (dec)	Index:GE 设置 (30000-based)
SW221	2 (hex)	在 GE1 中设置 2
SW222	-	
SW223	31004 (dec)	
SW224	37 (hex)	在 GE4 中设置 55
SW225	-	
SW220	1001 (dec)	Index:GE1~读取 (0-based)
SW221	0002 (hex)	
SW222	3700 (hex)	
SW223	1002 (dec)	Index:GE2~读取 (0-based)
SW224	0000 (hex)	
SW225	0037 (hex)	

4.3. 继电器间接地址指定

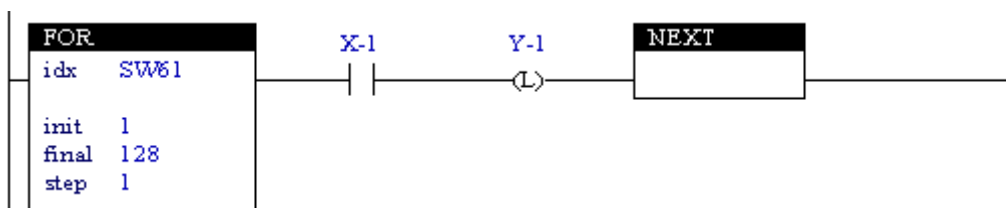
SW61~SW69 是指定间接地址的系统内存。不管采用何种中继形式，如果用“-1” ~ “-9”的值指定中继地址，则用 SW61~SW69 上储存的值的地址进行指定。



例如，在 SW61=12, SW65=34, SW69=56 的情况下，如下存储继电器。

MW-1 → MW12, MW-5 → MW34, MW-9 → MW56
 FN-1.X3 → FN12.X3, FN-5.X3 → FN34.X3, FN-9.X3 → FN56.X3
 X-1 → X12, X-5 → X34, X-9 → X56
 Y-1 → Y12, Y-5 → Y34, Y-9 → Y56

下列内置 PLC 例题利用 FOR/NEXT 门和间接地址指定方式制备输出相当于 X1~X128 的输入信号的 Y1~128 信号的动作的示例。



4.4. 计时器&计数器 Relay

(1) 计时器和计数器 Relay 只支持 down-counting。

用户可按 10msec 单位设置计时器 Base。

计时器值在内部以 32bit 值处理、可以计时到 2、147、483、647[msec] (约 597 小时)。

(2) 计时器 / 计数器值具有以下意义。

计时器 & 计数器值	说明
0	接点 On (=counting 完毕)
-1	接点 Off
其外	接点 Off; timing & counting (进行中)

(3) 连接计时器 / 计数器 Relay 的 Rung 处于非激活状态、

TON:TL(计时器)值就会成为-1。

CTD:CL(计数器)值就会继续保持。

(4) 连接计时器 / 计数器 Relay 的 Rung 处于激活状态、

TON

如果 TL 值小于 0、TL 的起初值会保存为“计时器 Base x Freeset x 10”、TL 值大于 0、每 5msec 会减少 5。

CTD

如果 CL 值小于 0、CL 的起初值会成为 Freeset 值、如果 CL 值大于 0、每次从非激活状态变为激活状态时其数值就会减少 1。



HD

HYUNDAI
ROBOTICS

5

命令语说明



5. 命令语说明

内置 PLC

5.1. 命令语表

表 5-1 命令语表

命令语				说明
编号	Mnemonic	名称	标记	
	RUNG	Rung	┌─┐	显示 rung。
	BST	Branch Start	└─┐	Branch 的开始
	BND	Branch End	┐─└	Branch 的结束
	NXB	Nested Branch	┌、┐	Branch 的重复
Logic 检查命令				检查结果、 是表示 Rung 被激活/(否: 非激活)
1	XIC	Examine if Closed	- -	检查接点是否关闭(A 接点)
2	XIO	Examine if Open	- / -	检查接点是否打开(B 接点)
3	INV	Inverting	-//-	Rung 的结果反转(inverting)
4	EQU	Equal	-目-	检查是否相同(=)
5	NEQ	Not Equal	-目-	检查是否不同(<>)
6	LES	Less Than	-目-	检查是否小于基准值(<)
7	GRT	Greater Than	-目-	检查是否大于基准值(>)
8	LEQ	Less Than 或 Equal	-目-	检查是否比基准值(<=)小或相同。
9	GEQ	Greater Than 或 Equal	-目-	检查是否比基准值(>=)大或相同。
输出命令				
10	OTE	Output Energize	-()-	输出 Rung 的状态 (激活:ON/非激活:OFF)
11	OTL	Output Latch	-(L)-	Rung 处于激活状态、就输出为 ON(high)
12	OTU	Output Unlatch	-(U)-	Rung 处于激活状态、就输出为 OFF(low)

5. 命令语说明

命令语				说明
编号	Mnemonic	名称	标记	
13	OSR	One Shot Rising	-(OSR)-	Rung 处于激活状态、只在 scan 期间输出 ON
14	RES	Reset	-(RES)-	Rung 处于激活状态、对计时器或计数器进行复位
计时器及计数器命令				
15	TON	Time On Delay	-[目]-	Rung 处于激活状态时计时器会工作
16	CTD	Count Down	-[目]-	Down Count Rung 的激活时间(非激活→激活)
计算命令				
17	ADD	Add	-[目]-	Rung 处于激活状态、(+)计算
18	SUB	Subtract	-[目]-	Rung 处于激活状态、(-)计算
19	MUL	Multiply	-[目]-	Rung 处于激活状态、(x)计算
20	DIV	Divide	-[目]-	Rung 处于激活状态、(/)计算
21	POW	Power	-[目]-	Rung 处于激活状态、(^:累乘)计算
数据变换命令				
22	TOD	Convert int. to BCD	-[目]-	Rung 处于激活状态、变换为 BCD
23	FRD	Convert form BCD to int.	-[目]-	Rung 处于激活状态、变换为 integer
24	SEG	7' Segment	-[目]-	Rung 处于激活状态、变换为 7' Segment 值
移动及复制命令				
25	MOV	Move	-[目]-	Rung 处于激活状态、复制一个数据
26	COP	Copy data	-[目]-	Rung 处于激活状态、复制多个数据
27	CCOP	Conditional Copy data	-[目]-	根据 Rung 状态复制多个数据
应用命令				

命令语				说明
编号	Mnemonic	名称	标记	
28	ROT	Rotating Output	-目-	Rung 处于激活状态、按序输出
Block 控制命令				
29	FOR	For	-目-	Rung 处于激活状态、到 NEXT 反复执行
30	NEXT	Next	-目-	如果是反复次数以内、就用 FOR 文 JUMP
31	LBL	Label	-目-	用 Jump 命令指定要 Jump 的位置
32	JMP	Jump	-目-	Rung 处于激活状态、就向 LabelJump (Label<0 时、就会越过-n 个 Next)
33	CALL	Call	-目-	Rung 处于激活状态、就呼叫 Sub-ladder
34	END	End	-目-	Rung 处于激活状态、Sub-ladder end

5.2. 可使用的 Operands

表 5-2 可使用的 Operands

inst	Relay		Input		Output		Timer		Counter		Memory		Const (32bit) (L/F)
	Type		X, DO		Y, DI, R, K, SP		T		C		M, S		
	arg	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m		
XIC			x		x		x		x		x		x
XIO			x		x		x		x		x		x
EQU	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
LES	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
GRT	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
LEQ	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
GEQ	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
OTF		x	x		x	x	x	x	x	x	x		x
OTL		x	x		x	x	x	x	x	x	x		x
OUT		x	x		x	x	x	x	x	x	x		x
OSR		x	x		x	x	x	x	x	x	x		x
RES		x	x	x	x		x		x	x	x		x
TON	tm	x	x	x	x		x	x	x	x	x		x
	bs	x		x		x		x		x			
	pst	x		x		x		x		x			
CTD	cnt	x	x	x	x	x	x	x	x	x	x		x
	pst	x		x		x		x		x			
ADD	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
	dst	x	x	x		x		x		x			x
SUB	sA	x		x		x		x		x			
	sB	x		x		x		x		x			
	dst	x	x	x		x		x		x			x
MUL	sA	x		x		x		x		x			
	sB	x		x		x		x		x			

inst	Relay	Input		Output		Timer		Counter		Memory		Const (32bit) (L/F)
	Type	X, DO		Y, DI, R, K, SP		T		C		M, S		
	arg	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	
	dst	x	x	x		x		x		x		x
DIV	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
POW	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
TOD	src	x	u	x	u	x	u	x	u	x	u	u
	dst	x	x	x	u	x	x	x	x	x	u	x
FRD	src	x	u	x	u	x	u	x	u	x	u	u
	dst	x	x	x	u	x	x	x	x	x	u	x
SEG	src	x	u	x	u	x	u	x	u	x	u	u
	dst	x	x	x	u	x	x	x	x	x	u	x
MOV	src	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
COP	src					x		x				
	dst	x	x			x		x				x
	len	x		x		x		x		x		
CCOP	sA					x		x				
	sB					x		x				
	dst	x	x			x		x				x
	len	x		x		x		x		x		
ROT	st	x		x		x		x		x		
	cnt	x		x	x	x		x		x		
	tm	x	x	x	x		x	x	x	x	x	x
	rep	x		x		x		x		x		
	out	x	x	x		x	x	x	x	x		x
	rst		x		x		x		x	x	x	x
	tmp	x	x	x		x	x	x	x	x		x
FOR	idx	x	x	x		x	x	x	x	x		x
	init	x		x		x		x		x		
	final	x		x		x		x		x		
	step	x		x		x		x		x		

5. 命令语说明

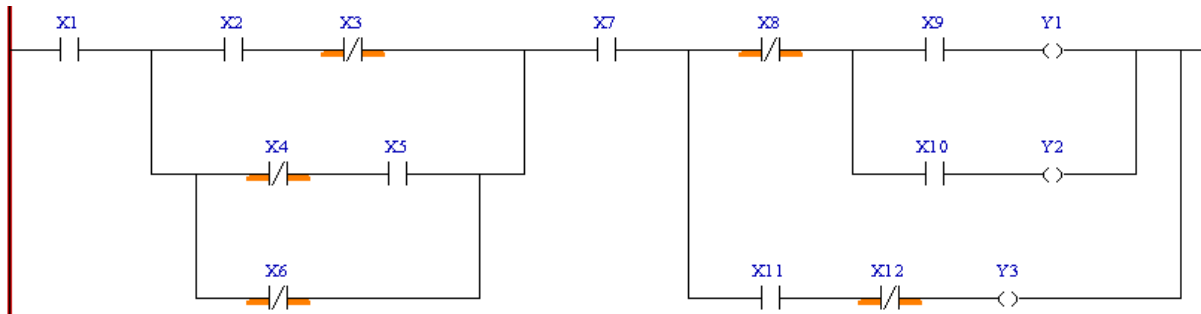
	Relay	Input		Output		Timer		Counter		Memory		Const (32bit) (L/F)
	Type	X, DO		Y, DI, R, K, SP		T		C		M, S		
	arg	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	n	(B, W, L, F) _m	
LRI	label	x	x	x	x	x	x	x	x	x	x	
JMP	label	x		x		x		x		x		
CALL	S file	x		x		x		x		x		

- (1) 表中 ‘x’ 标记表示不能使用。
- (2) Relay(relay)的 n 表示 bit 编号、(B、W、L、F)_m 是区分 Byte(8bit)、Word(16bit)、Long(32bit)、Float 形式的字母、m 表示各形式的数据编号。(例、X1、XB2、XW2、XL2、XF2)
- (3) 表中的 ‘u’ 标记显示以没有符号的(unsigned) 值处理。
- (4) 因此、输入命令语的可通过选择 Relay 形式(B、W、L、F)来选择数据尺寸(data size)。
- (5) 但、在 COP(复制)和 CCOP(按条件复制) 命令中、如果 src 不是常数、src 与 dst 的 Relay 形式应相同。
- (6) 输入 Bit 值时可以 Bite(B)或 Word(W)、或 Long Word(L)的相对值输入。即、例如 X34=XB5/2=XW3/2=XL2/2、可以多种形式输入相同的 bit 编号。
- (7) 常数(constant)
常数的数据尺寸为 32bit 数据形式具有 2 进数、10 进数、16 进数、以&B 开始、就表示 2 进数；以&H 开始就表示 16 进数；没有词头的数字则表示 10 进数。因此、125 可以&H7D 或&B01111101 等形式输入。
- (8) 在表格中、没有 ‘u’ 标记的所有 Relay 的(B、W、L、F)都以有符号的(signed) 值处理。
例如、在 MOV 命令中、把 B(8bit)值移动到 W(16bit)时、signed bit 会扩张、应注意。即、如果把 RB(-1=&HFF)移动到 RW、就会成为 0xFFFF(-1)、而不是&H00FF(255)。
但、与 BCD 数据有关的 TOD 和 FRD 命令、变换为了 7’ Segment 数据的 SEG 命令会以没有符号的值(unsigned)处理。

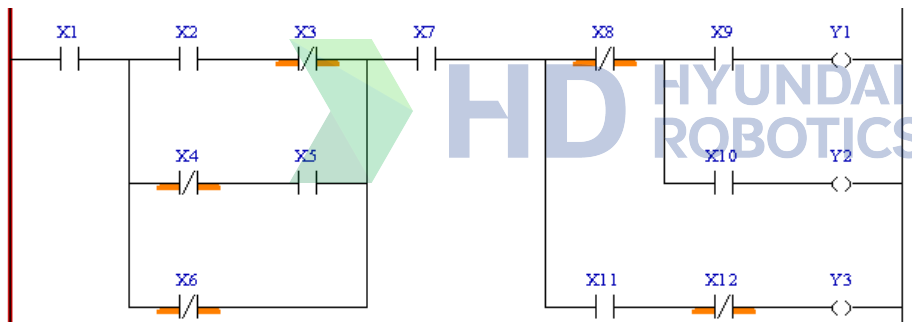
5.3. Ladder Diagram 与 Mnemonic 的关系

用 Ladder 编制的文件会自动变换为 Mnemonic、并向控制器发送。

(1) Ladder diagram



或点击 HRLadder 的 branch gap []、显示以下数据。



(2) Mnemonic

```
SOR XIC X1 BST XIC X2 XIO X3 NXB BST XIO X4 XIC X5 NXB XIO X6 BND BND
XIC X7 BST XIO X8 BST XIC X9 OTE Y1 NXB XIC X10 OTE Y2 BND
NXB XIC X11 XIO X12 OTE Y3 BND
```



HD

HYUNDAI
ROBOTICS

6

命令语详细说明



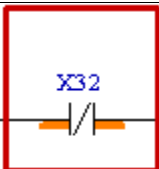
6. 命令语详细说明

内置 PLC

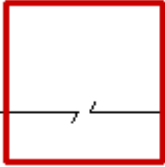
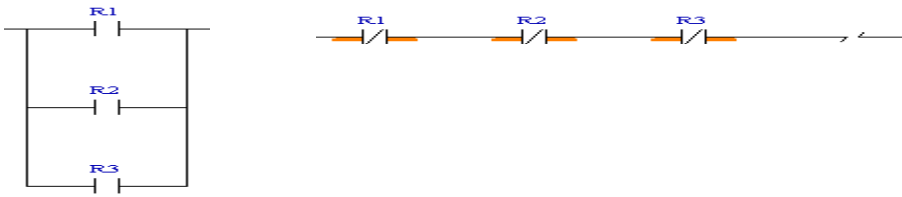
6.1. XIC(Examine if Closed):Close 检查

标记	可使用的 Relay 组合																												
X34	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(L/F)</td></tr><tr><th>inst</th><td>arg. n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>XIC</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	arg. n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)	XIC	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const (32bit)																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																							
inst	arg. n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)																							
XIC	X	X	X	X	X	X																							
说明	因数值为 1、就会激活 (active)Rung；如果是 0、就会处于非激活状态。																												
使用例	XIC(X2)；XIC(DO2)；XIC(Y2)；XIC(DI2)；XIC(R2)；XIC(K2)；XIC(SP1)；XIC(T2)；XIC(C2)；																												
输入方法	可以把 X18 输入为 XB3/2 或 XW2/2 或 XL1/2。																												

6.2. XIO(Examine if Open):Open 检查

标记	可使用的 Relay 组合																												
X32 	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td></td></tr><tr><th>inst</th><td>arg. n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>XIO</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S		inst	arg. n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)	XIO	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const (32bit)																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S																								
inst	arg. n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)																							
XIO	X	X	X	X	X	X																							
说明	因数值为 0、就会激活(active)Rung；如果是 1、就会处于非激活状态。																												
使用例	XIO(X2)；XIO(DO2)；XIO(Y2)；XIO(DI2)；XIO(R2)；XIO(K2)；XIO(SP1)；XIO(T2)；XIO(C2)；																												
输入方法	可以把 X18 输入为 XB3/2 或 XW2/2 或 XL1/2。																												

6.3. INV(Inverting):反转

标记	说明
	反转 Rung 之前的结果（激活$\leftrightarrow$非激活）。 此处理方式根据 De Morgan 的规律成为 $\neg(A \times B) = \neg A + \neg B$ 或 $\neg(A + B) = \neg A \times \neg B$、原本应以多个 Branch 构成 OR Logic、但如果使用此方式、就无需 Branch 也可简单构成 AND Logic。 即、$(R1+R2+R3) = \neg(\neg R1 \times \neg R2 \times \neg R3)$、以下两个 Logic 的结果会相同。 

6.4. EQU(Equal):检查是否相同

标记	EQU source aXB3 0 source b100																																					
Relay 组合	<table><tr><th rowspan="2">relay type</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg.</th><th>n (B, W, L, F)</th><th>m (B, W, L, F)</th><th>n (B, W, L, F)</th><th>m (B, W, L, F)</th><th>n (B, W, L, F)</th><th>m (B, W, L, F)</th></tr><tr><td rowspan="2">EQU</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B, W, L, F)	m (B, W, L, F)	n (B, W, L, F)	m (B, W, L, F)	n (B, W, L, F)	m (B, W, L, F)	EQU	sA	X	X	X	X	X	X	sB	X	X	X	X	X	X
relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																														
	X, DO	Y, DI, R, K, SP	T	C	M, S																																	
inst	arg.	n (B, W, L, F)	m (B, W, L, F)	n (B, W, L, F)	m (B, W, L, F)	n (B, W, L, F)	m (B, W, L, F)																															
EQU	sA	X	X	X	X	X	X																															
	sB	X	X	X	X	X	X																															
因数	sA: 要比较的第一个 Relay (或数字) sB: 要比较的第二个 Relay (或数字)																																					
说明	比较两个值时如果相同、Rung 就会被激活。(激活接点)																																					
使用例	EQU(XB3、100): XB3 的值为 100、Rung 就会被激活。 EQU(XB3、RW4): XB3 与 RW4 的值相同、Rung 就会被激活。																																					

6.5. NEQ(Not Equal):检查是否不同

标记	NEQ source aXB3 0 source b100																																								
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td>NEQ</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	NEQ	sA	X	X	X	X	X			sB	X	X	X	X	X	
	relay	Input	Output	Timer	Counter	Memory	const																																		
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																		
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																		
NEQ	sA	X	X	X	X	X																																			
	sB	X	X	X	X	X																																			
因数	sA: 要比较的第一个 Relay (或数字) sB: 要比较的第二个 Relay (或数字)																																								
说明	比较两个值时不同、Rung 就会被激活。(激活接点)																																								
使用例	NEQ(XB3、100):XB3 的值非为 100、 Rung 就会被激活。 NEQ(XB3、RW4):XB3 的值与 RW4 的值不同、Rung 就会被激活。																																								

6.6. LES(Less Than):检查是否小于基准值

标记	LES source aXB3 0 source b100																																							
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td rowspan="2">LES</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	LES	sA	X	X	X	X	X		sB	X	X	X	X	X	
	relay	Input	Output	Timer	Counter	Memory	const																																	
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																	
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																	
LES	sA	X	X	X	X	X																																		
	sB	X	X	X	X	X																																		
因数	sA: 要比较的第一个 Relay (或数字) sB: 要比较的第二个 Relay (或数字)																																							
说明	如果 sA 小于 sB、 Rung 就会被激活。(接点激活)																																							
使用例	LES(XB3、100):XB3 的值小于 100、Rung 就会被激活。 LES(XB3、RW4):XB3 的值小于 RW4 的值、Rung 就会被激活。																																							

6.7. GRT (Greater Than) : 检查是否大于基准值

标记	GRT source aXB3 0 source b100																																					
Relay 组合	<table><tr><th rowspan="2">relay type</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th></tr><tr><td rowspan="2">GRT</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	GRT	sA	X	X	X	X	X	X	sB	X	X	X	X	X	X
relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																														
	X, DO	Y, DI, R, K, SP	T	C	M, S																																	
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m																															
GRT	sA	X	X	X	X	X	X																															
	sB	X	X	X	X	X	X																															
因数	sA: 要比较的第一个 Relay (或数字) sB: 要比较的第二个 Relay (或数字)																																					
说明	sA 大于 sB、Rung 就会被激活。(接点激活)																																					
使用例	GRT(XB3、100):XB3 的值大于 100、Rung 就会被激活。 GRT(XB3、RW4):XB3 的值大于 RW4 的值、Rung 就会被激活。																																					

6.8. LEQ (Less Than or Equal) : 检查是否比基准值小或相同

标记	LEQ source aXB3 0 source b100																																								
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n (B, W, L, F) m</th><th>n (B, W, L, F) m</th><th>n (B, W, L, F) m</th><th>n (B, W, L, F) m</th><th>n (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td>LEQ</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)	LEQ	sA	X	X	X	X	X			sB	X	X	X	X	X	
	relay	Input	Output	Timer	Counter	Memory	const																																		
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																		
inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)																																		
LEQ	sA	X	X	X	X	X																																			
	sB	X	X	X	X	X																																			
因数	sA: 要比较的第一个 Relay (或数字) sB: 要比较的第二个 Relay (或数字)																																								
说明	sA 小或同于 sB、Rung 就会被激活。(接点激活)																																								
使用例	LEQ (XB3、100): XB3 的值比 100 小或相同、Rung 就会被激活。 LEQ (XB3、RW4): XB3 的值比 RW4 的值小或相同、Rung 就会被激活。																																								

6. 9. GEQ(Greater Than or Equal):检查是否比基准值大或相同

标记	GEQ source aXB3 0 source b100																																
Relay 组合	<table><tr><th rowspan="2">relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg. n:(B,W,L,F) m</th><th>n:(B,W,L,F) m</th><th>n:(B,W,L,F) m</th><th>n:(B,W,L,F) m</th><th>n:(B,W,L,F) m</th><th></th></tr><tr><td rowspan="2">GEQ</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	type X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg. n:(B,W,L,F) m	n:(B,W,L,F) m	n:(B,W,L,F) m	n:(B,W,L,F) m	n:(B,W,L,F) m		GEQ	sA	X	X	X	X		sB	X	X	X	X	
relay	Input		Output	Timer	Counter	Memory	const (32bit) (L/F)																										
	type X, DO	Y, DI, R, K, SP	T	C	M, S																												
inst	arg. n:(B,W,L,F) m	n:(B,W,L,F) m	n:(B,W,L,F) m	n:(B,W,L,F) m	n:(B,W,L,F) m																												
GEQ	sA	X	X	X	X																												
	sB	X	X	X	X																												
因数	sA:要比较的第一个 Relay(或数字) sB:要比较的第二个 Relay(或数字)																																
说明	sA 比 sB 大或相同、Rung 就会被激活。(接点激活)																																
使用例	GEQ(XB3、100):XB3 的值比 100 大或相同、Rung 就会被激活。 GEQ(XB3、RW4):XB3 的值比 RW4 的值大或相同、Rung 就会被激活。																																

6. 10. OTE(Output Energize):一般输出

标记	可使用的 Relay 组合																												
Y23 ()	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><td>arg. n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>OTE</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg. n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	OTE	x	x	x	x	x	x
relay	Input	Output	Timer	Counter	Memory	const																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																							
inst	arg. n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																							
OTE	x	x	x	x	x	x																							
说明	根据 Rung 状态输出输出信号。 即、Rung 处于激活状态时以 On(high)输出信号、处于非激活状态时以 Off(low) 输出。																												
因数例	Y2; DI2; R2; K2; SP1;																												
输入方法	可以把 Y18 输入为 YB3/2 或 YW2/2 或 YL1/2。																												

6.11. OTL(Output Latch):Latch 输出

标记	可使用的 Relay 组合																																
Y20 (L)	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th></th><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(L/F)</td></tr><tr><th>inst</th><th>arg.</th><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>OTL</td><td></td><td>X X</td><td>X</td><td>X X</td><td>X X</td><td>X X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const (32bit)		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	OTL		X X	X	X X	X X	X X	X
	relay	Input	Output	Timer	Counter	Memory	const (32bit)																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																										
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																										
OTL		X X	X	X X	X X	X X	X																										
说明	Rung 处于激活状态、 就会以 On(high) 输出信号。 但如果 Rung 处于非激活状态、输出就不会变更。																																
因数例	Y2; DI2; R2; K2; SP1;																																
输入方法	可以把 Y18 输入为 YB3/2 或 YW2/2 或 YL1/2。																																

6.12. OTU(Output Unlatch):Unlatch 输出

标记	可使用的 Relay 组合																												
Y21 	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(L/F)</td></tr><tr><th>inst</th><td>arg. n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>OTU</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	arg. n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	OTU	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const (32bit)																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																							
inst	arg. n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																							
OTU	X	X	X	X	X	X																							
说明	Rung 处于激活状态、就会以 Off (low) 输出信号。 但如果 Rung 处于非激活状态、输出就不会变更。																												
使用例	Y2; DI2; R2; K2; SP1;																												
输入方法	可以把 Y18 输入为 YB3/2 或 YW2/2 或 YL1/2。																												

6.13. OSR(One Shot Rising):One Shot 输出

标记	可使用的 Relay 组合																												
Y22 (OSR)	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><td>arg.</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>OSR</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	OSR	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																							
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																							
OSR	X	X	X	X	X	X																							
说明	Rung 处于激活状态、 只在一次 scan 期间以 On(high)输出。 即、 Rung 从非激活状态变更为激活状态时、 只在一个 scan 过程中 On 该 Relay。																												
因数例	Y2; DI2; R2; K2; SP1;																												
输入方法	可以把 Y18 输入为 YB3/2 或 YW2/2 或 YL1/2。																												

6.14. RES(Reset):复位

标记	可使用的 Relay 组合																												
T3 (RES)	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><td>arg.</td><td>n : (B,W,L,F) m</td><td>n : (B,W,L,F) m</td><td>n : (B,W,L,F) m</td><td>n : (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>RES</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	(L/F)	RES	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																							
inst	arg.	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	(L/F)																							
RES	X	X	X	X	X	X																							
说明	如果 Rung 处于激活状态、 就会清除输出(-1)。																												
因数例	T2; C2;																												

6.15. TON(Time On Delay):计时器

标记	TON timer T2 =0 timer base(1/100s) 100 preset 2																																											
Relay 组合	<table><tr><td rowspan="2"></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td rowspan="2">const (32bit) (L/F)</td></tr><tr><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td></tr><tr><td>inst</td><td>arg.</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td></tr><tr><td rowspan="3">TON</td><td>tm</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>bs</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>pst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	TON	tm	X	X	X	X	X	X	bs	X	X	X	X	X	X	pst	X	X	X	X	X	X
	relay		Input	Output	Timer	Counter	Memory	const (32bit) (L/F)																																				
	type	X, DO	Y, DI, R, K, SP	T	C	M, S																																						
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m																																						
TON	tm	X	X	X	X	X	X																																					
	bs	X	X	X	X	X	X																																					
	pst	X	X	X	X	X	X																																					
因数	tm:指定计时器 Relay 编号。 bs:把计时器的单位(如果是 100、就按 1 秒单位设置；如果是 10、就按 0.1 秒 单位设置、 ……)设为 Relay 或数字。 pst:把 Freeset 值(时间[msec]=bs*pst*10) 设置为 Relay 或数字。																																											
说明	计算 Rung 的激活时间、过设置时间(bs x pst x 10) [msec]后该计时器 Relay 就会 On(high)。 但如果 Rung 处于非激活状态、就会即时清除(-1)。 参考) TL 的值为 1msec 单位。																																											
使用例	TON(T2、100、5):利用计时器把 T2 设置为 (100x5x10[msec]=5[sec]) 秒。 TON(T3、RW3、RW4):利用计时器把 T3 设置为与(RW3) x (RW4) x10 的值相同。																																											

6.16. CTD(Count Down):计数器

标记	CTD counter C2=1 CL2=0 preset 100																																					
Relay 组合	<table><tr><th rowspan="2"> </th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td></tr><tr><th>inst</th><th>arg.</th><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td></td></tr><tr><td rowspan="2">CTD</td><td>cnt</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>pst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m		CTD	cnt	X	X	X	X	X	X	pst	X	X	X	X	X	
	relay		Input	Output	Timer	Counter	Memory	const (32bit) (L/F)																														
	type	X, DO	Y, DI, R, K, SP	T	C	M, S																																
inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m																																
CTD	cnt	X	X	X	X	X	X																															
	pst	X	X	X	X	X																																
因数	cnt:指定计数器 Relay 编号。 pst:用 Replay 或数字设置 Freeset 值。																																					
说明	Down Count Rung 的上升(从非激活变为激活)。 如果 CL 值为 0、 该计数器就会 On(high)、不再进行 Count。 Rung 处于激活时如果该 CL 值为负数、就把 pst 值保存于 CL。 参考) 即使 Rung 处于非激活状态、CL 也不会清除(-1)。																																					
使用例	CTD(C2、100):将 C2 设置为(100)个 Down-Counter。 CTD(C3、RW3):将 C3 设置为(RW3)值 Down-Counter。																																					

6. 17. ADD (Add) :加

标记	ADD source aXB3 0 source b50 destinationRW3 0																																															
Relay 组合	<table><tr><th rowspan="2">inst</th><th>relay type</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>arg.</th><th>n : (B,W,L,F)</th><th>m</th><th>n : (B,W,L,F)</th><th>m</th><th>n : (B,W,L,F)</th><th>m</th><th>n : (B,W,L,F)</th><th>m</th></tr><tr><td rowspan="3">ADD</td><td>sA</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td>X</td></tr></table>	inst	relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	arg.	n : (B,W,L,F)	m	n : (B,W,L,F)	m	n : (B,W,L,F)	m	n : (B,W,L,F)	m	ADD	sA	X		X		X		X		sB	X		X		X		X		dst	X	X	X		X		X	X
inst	relay type		Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																						
	arg.	n : (B,W,L,F)	m	n : (B,W,L,F)	m	n : (B,W,L,F)	m	n : (B,W,L,F)	m																																							
ADD	sA	X		X		X		X																																								
	sB	X		X		X		X																																								
	dst	X	X	X		X		X	X																																							
因数	sA: 计算所需的第一个 Relay (或数字) sB: 计算所需的第二个 Relay (或数字) dst: 保存结果的 Relay																																															
说明	Rung 处于激活状态、 在 sA 值添加 sB 的值后保存于 dst。 如果计算结果发生 Overflow、就设置为 SP8=1。																																															
使用例	ADD (XB3、RB2、RW3) : (RW3)=(XB3)+(RB2) ADD (XB3、RB2、RB2) : (RB2)=(XB3)+(RB2) ADD (XB3、50、RW3) : (RW3)=(XB3)+50																																															

6.18. SUB(Subtract):减

标记	SUB source aXB3 0 source b50 destinationRW3 0																																														
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>(L/F)</th></tr><tr><td rowspan="3">SUB</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	SUB	sA	X	X	X	X	X		sB	X	X	X	X	X		dst	X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																								
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																								
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																																								
SUB	sA	X	X	X	X	X																																									
	sB	X	X	X	X	X																																									
	dst	X	X	X	X	X	X																																								
因数	sA: 计算所需的第一个 Relay(或数字) sB: 计算所需的第二个 Relay(或数字) dst: 保存结果的 Relay																																														
说明	Rung 处于激活状态、 从 sA 值减去 sB 的值后保存于 dst。																																														
使用例	SUB(XB3、RB2、RW3) : (RW3)=(XB3)-(RB2) SUB(XB3、RB2、RB2) : (RB2)=(XB3)-(RB2) SUB(XB3、50、RW3) : (RW3)=(XB3)-50																																														

6.19. MUL(Multiply):乘

标记	MUL source aXB30 source b50 destinationRW30																																														
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td rowspan="3">MUL</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	MUL	sA	X	X	X	X	X		sB	X	X	X	X	X		dst	X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																								
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																								
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																								
MUL	sA	X	X	X	X	X																																									
	sB	X	X	X	X	X																																									
	dst	X	X	X	X	X	X																																								
因数	sA: 计算所需的第一个 Relay (或数字) sB: 计算所需的第二个 Relay (或数字) dst: 保存结果的 Relay																																														
说明	Rung 处于激活状态、在 sA 值乘于 sB 的值后保存于 dst。 如果在计算结果发生 Overflow、就设置为 SP8=1。																																														
使用例	MUL (XB3、RB2、RW3) : (RW3)=(XB3) x (RB2) MUL (XB3、RB2、RB2) : (RB2)=(XB3) x (RB2)																																														

	MUL (XB3、50、RW3) : (RW3)=(XB3) x50
--	------------------------------------

6.20. DIV(Divide) :除

标记	DIV source aXB30 source b50 destinationRW30																																																								
Relay 组合	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><th colspan="2">X, DO</th><th colspan="2">Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>arg.</th><th></th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th></tr><tr><td rowspan="3">DIV</td><td>sA</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td>X</td></tr></table>	inst	relay	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO		Y, DI, R, K, SP		T	C	M, S	arg.		n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	DIV	sA	X		X		X		X		sB	X		X		X		X		dst	X	X	X		X		X	X
inst	relay		Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																															
	type	X, DO		Y, DI, R, K, SP		T	C	M, S																																																	
arg.		n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m																																																
DIV	sA	X		X		X		X																																																	
	sB	X		X		X		X																																																	
	dst	X	X	X		X		X	X																																																
因数	sA: 计算所需的第一个 Relay (或数字) sB: 计算所需的第二个 Relay (或数字) dst: 保存结果的 Relay																																																								
说明	Rung 处于激活状态、 将 sA 值除以 sB 值后保存于 dst。 如果 sB 为 0 或计算结果发生 Overflow 就设置为 SP8=1。																																																								
使用例	DIV (XB3、RB2、RW3) : (RW3)=(XB3) / (RB2) DIV (XB3、RB2、RB2) : (RB2)=(XB3) / (RB2) DIV (XB3、50、RW3) : (RW3)=(XB3) /50																																																								

6.21. POW(Power) :累乘

标记	POW source aXB3 source b3 destinationRW3																																																												
Relay 组合	<table><tr><th rowspan="2"></th><th>relay</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><th colspan="2">X, DO</th><th colspan="2">Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg.</th><th>n</th><th>(B,W,L,F)</th><th>m</th><th>n</th><th>(B,W,L,F)</th><th>m</th><th>n</th><th>(B,W,L,F)</th><th>m</th></tr><tr><td rowspan="3">POW</td><td>sA</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td></tr></table>		relay	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO		Y, DI, R, K, SP		T	C	M, S	inst	arg.	n	(B,W,L,F)	m	n	(B,W,L,F)	m	n	(B,W,L,F)	m	POW	sA	X		X		X		X			sB	X		X		X		X			dst	X	X	X		X		X		X
	relay		Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																																			
	type	X, DO		Y, DI, R, K, SP		T	C	M, S																																																					
inst	arg.	n	(B,W,L,F)	m	n	(B,W,L,F)	m	n	(B,W,L,F)	m																																																			
POW	sA	X		X		X		X																																																					
	sB	X		X		X		X																																																					
	dst	X	X	X		X		X		X																																																			
因数	sA: 计算所需的第一个 Relay(或数字) sB: 计算所需的第二个 Relay(或数字) dst: 保存结果的 Relay																																																												
说明	Rung 处于激活状态、 在 sA 值累乘 sB 值后保存于 dst。 如果计算结果发生 Overflow、就设置为 SP8=1。																																																												

使用例	POW(2、3、RW3) : (RW3) = $2^3 = 2 * 2 * 2 = 8$ POW(XB3、RB2、RW3) : (RW3) = (XB3) ^ (RB2) POW(XB3、RB2、RB2) : (RB2) = (XB3) ^ (RB2) POW(XB3、3、RW3) : (RW3) = (XB3) ^ 3
-----	--

6.22. TOD(Convert to BCD):变换为BCD值

标记	TOD sourceXB30 destinationRB20																														
Relay 组合	<table><tr><td rowspan="2"></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td rowspan="2">const (32bit) (L/F)</td></tr><tr><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td></tr><tr><td>inst</td><td>arg.</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td></td></tr><tr><td>TOD</td><td>src dst</td><td>X X</td><td>U X</td><td>X X</td><td>U X</td><td>X X</td><td>U X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m		TOD	src dst	X X	U X	X X	U X	X X	U X
	relay		Input	Output	Timer	Counter	Memory	const (32bit) (L/F)																							
	type	X, DO	Y, DI, R, K, SP	T	C	M, S																									
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m																									
TOD	src dst	X X	U X	X X	U X	X X	U X																								
因数	src:原本(source) Relay(或数字) dst:目的地(destination) Relay																														
说明	Rung 处于激活状态、将 src 的值变换为 CD 后保存于 dst。 此命令为 BCD 形式、使用在在 7' 显示值的装置时方便。 如果 dst 为 Bite(B) 形式、就变换为两个 Digit、如果是 Word(W) 形式、就会变换为 4 个 Digit。 如果 Src 的值大于 Digit 数、就设置为 SP7=1。																														
使用例	TOD(XB3、RB2):将 XB3 的值变换为 BCD 后保存于 RB2。 (参考、BCD(Binary Coded Decimal)表示 4bit 的代码值可以具备 0~9 范围值得数字。即、在 BCD 以 4bit 显示的数字 0~F 中不使用 A~F。) 如果是 (XB3)=&H7B(123)、就把&H23(35)保存于 RB2、因为是&H7B(123)>&H63(99)、设置为 SP7=1。																														

6.23. FRD(Convert form BCD to Integer):变换为 integer

标记	FRD source XB3 0 destination RB3 0																																							
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td rowspan="2">FRD</td><td>src</td><td>X</td><td>U</td><td>X</td><td>U</td><td>X</td><td>U</td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	FRD	src	X	U	X	U	X	U	dst	X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																	
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																	
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																	
FRD	src	X	U	X	U	X	U																																	
	dst	X	X	X	X	X	X																																	
因数	src:原本(source) Relay(或数字) dst:目的地(destination) Relay																																							
说明	如果 Rung 处于激活状态、 就把 src 的 BCD 值变换为 integer 后保存于 dst。 此命令在接收以 BCD 形式输出的 Cam 开关值时可方便使用。 如果 src 的值非为 BCD、就设置为 SP7=1。 并且、 src 为 Word(W)形式、dst 为 Bite(B)形式时、要变换的 src 最大值为&H9999、因此 integer 的变换结果成为 9999(&H270F)、超过 Bite 的范围&Hff、从而发生 0 verflow、这时设置为 SP7=1。																																							
使用例	FRD(XB3、RB2):XB3 的值(BCD)变换为 integer 后保存于 RB2。 如果是 (XB3)=&H23(35)、 就把 23=&H17 保存于 RB2。																																							

6. 24. SEG (7' Segment) : 变换为 7' Segment 值

标记	SEG sourceXB3 0 destinationRW3 0																																								
Relay 组合	<table><tr><td></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td></td><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>arg.</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>(L/ F)</td></tr><tr><td>SEG</td><td>src</td><td>X</td><td>u</td><td>X</td><td>u</td><td>X</td><td>u</td></tr><tr><td></td><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/ F)	SEG	src	X	u	X	u	X	u		dst	X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																		
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																		
inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/ F)																																		
SEG	src	X	u	X	u	X	u																																		
	dst	X	X	X	X	X	X																																		
因数	src:原本 (source) Relay (或数字) dst:目的地 (destination) Relay																																								
说明	Rung 处于激活状态、将 src 的值变换为 7' Segment 值 (8bit) 后保存于 dst。 Dst 为 Word (W) 形式、就把 2 个 7' Segment 值 (8bit) 保存于 dst。																																								
7' Segment 数据	<pre>#define SEGM_A 0x01 #define SEGM_B 0x02 #define SEGM_C 0x04 #define SEGM_D 0x08 #define SEGM_E 0x10 #define SEGM_F 0x20 #define SEGM_G 0x40 #define SEGM_DP 0x80</pre> <pre>#define SEG_D_0 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_E SEGM_F) #define SEG_D_1 (SEGM_B SEGM_C) #define SEG_D_2 (SEGM_A SEGM_B SEGM_G SEGM_E SEGM_D) #define SEG_D_3 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_G) #define SEG_D_4 (SEGM_B SEGM_C SEGM_F SEGM_G) #define SEG_D_5 (SEGM_A SEGM_C SEGM_D SEGM_F SEGM_G) #define SEG_D_6 (SEGM_A SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_7 (SEGM_A SEGM_B SEGM_C) #define SEG_D_8 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_9 (SEGM_A SEGM_B SEGM_C SEGM_F SEGM_G) #define SEG_D_A (SEGM_A SEGM_B SEGM_C SEGM_E SEGM_F SEGM_G) #define SEG_D_B (SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_C (SEGM_A SEGM_D SEGM_E SEGM_F) #define SEG_D_D (SEGM_B SEGM_C SEGM_D SEGM_E SEGM_G) #define SEG_D_E (SEGM_A SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_F (SEGM_A SEGM_E SEGM_F SEGM_G)</pre>																																								
使用例	SEG (XB3、RW3): 相当于 XB3 值的 7' Segment 值保存于 RW3。 如果 (XB3) = (&H17)、 就把结合上述 SEG_D_1 (SEGM_B SEGM_C=0x02 0x04=0x06) =&H06 和 上述 SEG_D_7 (SEGM_A SEGM_B SEGM_C=0x01 0x02 0x04=0x07) =&H07 的 &H0607 保存于 RW3。																																								

6. 25. MOV (Move) :移动

标记	MOV source55 destinationRB2 0																																								
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td>MOV</td><td>src</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	MOV	src	X	X	X	X	X			dst	X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																		
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																		
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																		
MOV	src	X	X	X	X	X																																			
	dst	X	X	X	X	X	X																																		
因数	src:原本(source) Relay(或数字) dst:目的地(destination) Relay																																								
说明	Rung 处于激活状态、 就把 src 的值复制到 dst。 如果、 src 为 Word(W)形式、 dst 为 Bite(B)形式、 在 src 的值中只把下位 Bite 复制到 dst。 并且、 内置 PLC 的所有数据都以有符号的数据处理、 src 为 Bite(B)形式、 值为 -1 (&Hff)时复制为 Word(W)形式的 dst、 就会复制为-1 (&HFFFF)。(&H00ff 成为 255 的值。)																																								
使用例	MOV (XB3、 RB2) :XB3 的值复制到 RB2。 MOV (55、 RB2) :将 55 复制到 RB2。 MOV (-30、 RB2) :将-30 复制到 RB2。																																								

6. 26. COP (Copy data) : Copy

标记	COP source XB2 0 destination YB3 0 length 3																																													
Relay 组合	<table><tr><th rowspan="2">inst</th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>arg.</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>(L/F)</th></tr><tr><td rowspan="3">COP</td><td>src</td><td></td><td></td><td>X</td><td>X</td><td></td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td></tr><tr><td>len</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	inst	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	arg.	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	(L/F)	COP	src			X	X			dst	X	X	X	X		X	len	X	X	X	X	X	
inst	relay		Input	Output	Timer	Counter	Memory	const																																						
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																							
arg.	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	(L/F)																																							
COP	src			X	X																																									
	dst	X	X	X	X		X																																							
	len	X	X	X	X	X																																								
因数	src:原本(source) Relay 或想要充填的值(const) dst:目的地(destination) Relay len:数量(length) Relay(或数字)																																													
说明	Rung 处于激活状态、 从 src 位置开始按 len 数复制到 dst 位置。 Src 为数字时、 从 dst 开始按 len 数充填 src 值。这时如果 dst 为 bit 形式、 src 的数字为 0 就用 OFF 充填、 如果不是 0 就用 ON 充填。 Src 为 Relay 时、 src 和 dst 的数据形式应相同。 即、 如果 src 为 Bit、dst 也应为 Bit; 如果 src 为 Bite(B)、 dst 也应为 Bite(B); src 为 Word(W)、dst 也应是(W)。 如果 src+len 大于 src Relay 最大数量或 dst+len 大于 dst Relay 最大数量、 就按 Relay 最大数量进行复制。																																													
使用例	按指定数复制的例) COP (X2、 Y3、 4):Y3=X2、 Y4=X3、 Y5=X4、 Y6=X5 COP (XB2、 YB3、 3):YB3=XB2、 YB4=XB3、 YB5=XB4 COP (XW2、 YW3、 2):YW3=XW2、 YW4=XW3 用指定值(const)充填的例) COP (0、 Y3、 4):Y3=OFF、 Y4=OFF、 Y5=OFF、 Y6=OFF COP (1、 Y3、 4):Y3=ON、 Y4=ON、 Y5=ON、 Y6=ON COP (25、 Y3、 4):Y3=ON、 Y4=ON、 Y5=ON、 Y6=ON COP (25、 YB3、 3):YB3=25、 YB4=25、 YB5=25 COP (&H55AA、 YW3、 2):YW3=&H55AA、 YW4=&H55AA COP (0、 MW3、 50):MW3=0、 MW4=0、 MW5=0、 MW6=0、 ~、 MW52=0																																													

6.27. CCOP(Conditional Copy data):按条件复制

标记	CCOP source a X2 0 source b R5 0 destination Y3 0 length 3
----	---

6.28. ROT(Rotating Output):按序输出

标记	ROT start relay MW51 0 count 6 timer relay T5 1 repeat time(1/100s) 200 out relay MW58 0 reset relay X3 0 temp relay MW60 0																																																																																																																						
Relay 组合	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th colspan="2">Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><th colspan="2">X, DO</th><th colspan="2">Y, DI, R, K, SP</th><th>T</th><th>C</th><th colspan="2">M, S</th></tr><tr><th></th><th>arg.</th><th>n</th><th>(B, W, L, F)</th><th>m</th><th>n</th><th>(B, W, L, F)</th><th>m</th><th>n</th><th>(B, W, L, F)</th><th>m</th><th>n</th><th>(B, W, L, F)</th></tr><tr><td rowspan="6">ROT</td><td>st</td><td>X</td><td></td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>cnt</td><td>X</td><td></td><td></td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>tm</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>rep</td><td>X</td><td></td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>out</td><td>X</td><td></td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>rst</td><td>X</td><td></td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>tmp</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	inst	relay	Input		Output		Timer	Counter	Memory		const (32bit) (L/F)	type	X, DO		Y, DI, R, K, SP		T	C	M, S			arg.	n	(B, W, L, F)	m	n	(B, W, L, F)	m	n	(B, W, L, F)	m	n	(B, W, L, F)	ROT	st	X			X		X		X		X		cnt	X			X	X	X		X		X		tm	X		X	X	X		X	X	X	X		rep	X			X		X		X		X		out	X		X	X		X		X		X		rst	X		X	X		X		X	X	X		tmp	X		X	X	X	X	X	X	X	X	X
inst	relay		Input		Output		Timer	Counter	Memory		const (32bit) (L/F)																																																																																																												
	type	X, DO		Y, DI, R, K, SP		T	C	M, S																																																																																																															
	arg.	n	(B, W, L, F)	m	n	(B, W, L, F)	m	n	(B, W, L, F)	m	n	(B, W, L, F)																																																																																																											
ROT	st	X			X		X		X		X																																																																																																												
	cnt	X			X	X	X		X		X																																																																																																												
	tm	X		X	X	X		X	X	X	X																																																																																																												
	rep	X			X		X		X		X																																																																																																												
	out	X		X	X		X		X		X																																																																																																												
	rst	X		X	X		X		X	X	X																																																																																																												
tmp	X		X	X	X	X	X	X	X	X	X																																																																																																												
因数	st:开始(start) Relay cnt:数量(count) Relay(或数字) tm:计时器(timer) Relay(1/100sec 单位) rep:指定反复(repeat)时间的 Relay(或数字) out:输出(output)结果的 Relay rst:复位(reset)输出状态的 Relay tmp:显示数值输出方向的 Relay																																																																																																																						
说明	Rung 处于激活状态、 在 rep 时间内、从 st Relay 到 cnt 范围内的 Relay 值中非为 0 的值反复输出到 out Relay。 如果、 向 rst Relay 输入信号、从 st Relay 按 cnt 数充填 0、 将计时器值初始化为 rep 值、outRelay 也以 0 输出。 输出错误编号的装置只有一个、 且所发生的错误类型多时可方便使用此命令、 以便在指定时间内输出错误编号。																																																																																																																						
使用例	----- (错误条件 1) -----[MOV (21, MW51)]---- ----- (错误条件 2) -----[MOV (22, MW52)]---- ----- (错误条件 3) -----[MOV (23, MW53)]---- ----- (错误条件 4) -----[MOV (24, MW54)]---- ----- (错误条件 5) -----[MOV (25, MW55)]---- ----- (错误条件 6) -----[MOV (26, MW56)]---- ---[ROT(MW51、 6、 T5、 200、 MW58、 X3、 MW60)]--- -----[TOD(MW58、 YB3)]---- 上述 Logic 在错误条件 1~6 中发生一个以上错误、 错误编号就会保存于 MW51~MW56、 根据 ROT 命令发生的错误编号会在 MW58 保存 2 秒钟、并通过 TOD 命令变换为 BCD 值、按序显示于与 YB3 连接的显示装置。 如果、 信号被输入到连接于外部错误复位的 X3、 保存错误编号的 MW51~MW56 内容就会被清除为 0、 MW58 和 MW60 也被清除为 0、在显示装置显示 0。																																																																																																																						

6. 29. FOR (FOR) :Block 反复

标记	FOR idx SW61 initial 1 final 256 step 1																																												
Relay 组合	<table><tr><th rowspan="2">inst</th><th>relay type</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>arg.</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>(L/F)</th></tr><tr><td rowspan="4">FOR</td><td>idx</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>init</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>final</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>step</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	inst	relay type	Input	Output	Timer	Counter	Memory	const (32bit)	arg.	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	(L/F)	FOR	idx	X	X	X	X	X	X	init	X	X	X	X	X	X	final	X	X	X	X	X	X	step	X	X	X	X	X	X
inst	relay type		Input	Output	Timer	Counter	Memory	const (32bit)																																					
	arg.	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	(L/F)																																						
FOR	idx	X	X	X	X	X	X																																						
	init	X	X	X	X	X	X																																						
	final	X	X	X	X	X	X																																						
	step	X	X	X	X	X	X																																						
因数	idx:目录(index) Relay init:初始值(initial value) Relay 或数字 final:结束值(final value) Relay 或数字 step:作为步骤(step)的 Relay 或数字																																												
说明	Rung 处于激活状态、 从 init 到 final 按 Step 增加 idx Relay 值并反复执行到 NEXT Block。 执行 FOR 文时、在无条件 idx 代入 init 值。 FOR/NEXT 文最大可以重叠至 10 个。 → FOR() FOR() FOR() ...NEXT NEXT NEXT step>0 时、如果 init>final、就不执行、直接 Jump 到 NEXT。 step<0 时、如果 init<final 、就不执行、直接 Jump 到 NEXT。 Final 和 step 即使指定变数、也只适用首次开始 for 文时的值。 特别在 FOR 文的中间脱离时、可以使用 JMP(负数)命令。(请参考 JMP 命令说明) 注意) FOR 命令不进行 Branch 所需的另外处理。																																												
使用例	FOR(RB10、1、4、1):从 RB10=1 到 4 逐一增加、反复执行到 NEXT 命令。 { --[FOR(SW61、1、256、1)]--[XIC(X-1)]--[OTL(Y-1)]--[NEXT]--} :从 SW61=1 到 256 逐一增加、反复执行 {XIC(X-1)、 OTL(Y-1)} 命令。 即、 idx 使用相对 Addressing(SW61~SW69)所需的 Relay、 其 XIC 的 XRelay 和 OTL 的 YRelay 为“-1”、适用 SW61 值的编号、因此只把在 X1~X256 中属于 High 信号编号的 YRelay 编号输出为 High、未输入编号的 Y 输出保持之前状态。 参考) 相对 Addressing 是指、不管任何形式的 Relay 将其编号设为-1~9 值、就按保存于 SW61~SW69 的值指定 Relay 编号的方式。																																												

6. 30. NEXT(NEXT):Block Next

标记	
因数	无
说明	按 FOR 命令的 step 工作如下。 step>0 时、反复执行到 idx Relay 值比 final 小或相同为止、 step<0 时、反复执行到 idx Relay 值比 final 大或相同为止。 在没有 FOR 文的情况下执行 NEXT 文时、NEXT 命令会被忽视。 注意) FOR/NEXT 命令不会另行 Branch 处理、如果在 Branch 内记录 FOR 命令、在 Branch 外部或其它 Branch 内记录 NEXT、FOR 命令就会进行误工作。
使用例	请参考 FOR 命令的使用例。

6. 31. LBL(Label):Label 指定

标记	LBL label 99																																
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><th>arg.</th><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>LBL</td><td>label</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	LBL	label	x	x	x	x	x	
	relay	Input	Output	Timer	Counter	Memory	const																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																										
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																										
LBL	label	x	x	x	x	x																											
因数	label:Label 编号、 用大于 0 的数字(const)指定编号。																																
说明	利用 JMP 命令指定要 jump 的 Label 位置。 在 LBL 命令、指定位置与 Rung 的激活/非激活无关。 在一个 file 内有相同编号的 LBL 命令时、																																
使用例	与 JMP 命令一起使用、请参考 JMP 命令说明。																																

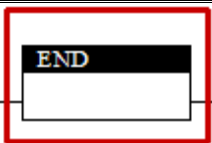
6. 32. JMP (Jump) : Jump

标记	JMP label 99																																
Relay 组合	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><th>arg.</th><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>JMP</td><td>label</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	JMP	label	x	x	x	x	x	
	relay	Input	Output	Timer	Counter	Memory	const																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																										
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																										
JMP	label	x	x	x	x	x																											
因数	label:Label 编号数字(const) 或 Relay																																
说明	Rung 处于激活状态、 Jump 到有与用 label 指定的 Label 值相同之 LBL 命令的位置。特别 label<0 时、可在 FOR 文的中间脱离时使用。(按负数指定值 skip) 注意 1) Label 的位置处于比 JMP 命令高的位置、而且 JMP 命令前面没有条件时、可能会陷入无限 Roof、应注意。这时、Scan Time 会超过 5 秒、因此被设置为 SP17=0n。 注意 2) 在 FOR/NEXT 命令 Block 内使用 JMP(正数)、不考虑脱离 Block 时的情况、因此 Block 控制可能会发生错误。这时应采取措施、以便利用 JMP(负数)Jump 到 NEXT 命令。																																
使用例	{ --[XIC(X21)]--[JMP(25)]--[.....]--[LBL(25)]--}:如果 X21Relay 为 1、就会根据 JMP(25)命令 Jump 到有 LBL(25)命令的位置。LBL 命令位于其他 Rung 也会正常工作。																																

6. 33. CALL(Call):Sub Ladder 呼叫

标记	CALL file number 21																																			
Relay 组合	<table><tr><td> relay </td><td> Input </td><td> Output </td><td> Timer </td><td> Counter </td><td> Memory </td><td> const (32bit) </td></tr><tr><td> type </td><td> X, DO </td><td> Y, DI, R, K, SP </td><td> T </td><td> C </td><td> M, S </td><td> (L/F) </td></tr><tr><td> inst </td><td> arg. </td><td> n </td><td> n </td><td> n </td><td> n </td><td> n </td></tr><tr><td></td><td> (B,W,L,F) m </td><td> (B,W,L,F) m </td><td> (B,W,L,F) m </td><td> (B,W,L,F) m </td><td> (B,W,L,F) m </td><td> (L/F) </td></tr><tr><td>CALL</td><td>S_file</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	arg.	n	n	n	n	n		(B,W,L,F) m	(B,W,L,F) m	(B,W,L,F) m	(B,W,L,F) m	(B,W,L,F) m	(L/F)	CALL	S_file	X	X	X	X	
relay	Input	Output	Timer	Counter	Memory	const (32bit)																														
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																														
inst	arg.	n	n	n	n	n																														
	(B,W,L,F) m	(B,W,L,F) m	(B,W,L,F) m	(B,W,L,F) m	(B,W,L,F) m	(L/F)																														
CALL	S_file	X	X	X	X																															
因数	file number:指定 Sub Ladder 文件编号(1~99)的 Relay 或数字																																			
说明	Rung 处于激活状态、 就呼叫指定为(file number)的编号之 Sub Ladder。 Sub Ladder 的文件名为 S01xxxx.LAD ~ S99xxxx.LAD、最大具有 99 个、 用户可任一设置文件名的 xxxx、最大可设置 15 字符。																																			
使用例	X23 FOR idx RW1 init 1 final 99 step 1 CALL file number RW1 NEXT 如果 X23 信号被输入、 就会按序呼叫从 S01xxxx.LAD 到 S99xxxx.LAD 的文件。 CALL 命令执行结果、如果该编号的 Sub Ladder 不存在或数值范围超出 1~99、就会被设置为 SP18=On; 如果正常执行、就会成为 SP18=Off。因此、根据需要须有 SubLadder 时、可在呼叫后利用 SP18 检测错误。 从 Ladder 通过 CALL 命令呼叫 1~99 号的 SubLadder、 按应用编号安排 SubLadder 编号、 并用控制器导入应用所需的 SubLadder、就可以期待该应用 Ladder 自动启动的效果。																																			

6. 34. END(End):关闭 Ladder

标志	
人数	无
说明	Rung 处于激活状态、就关闭现在启动的 Ladder。 如果现在启动的 Ladder 为 Sub Ladder、就 Return 为 Main Ladder; 如果当前启动的 Ladder 为 Main Ladder、就关闭后重新启动。
使用例	 需要初始化步骤时、用 S01_Initialize.LAD 文件编制处理化过程、然后在 Main Ladder 呼叫 1 号、在 S01 的第一个 rung 添加、SP3 是在开始运行后的第一个 Scan 启动的 Relay、S01 只执行一次、下一个 Scan 开始会在第一个 Rung Return。



- **Daegu Office (Head Office)**

50, Techno sunhwan-ro 3-gil, yuga, Dalseong-gun, Daegu, 43022, Korea

- **GRC**

477, Bundangsuseo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13553, Korea

- **대구 사무소**

(43022) 대구광역시 달성군 유가읍 테크노순환로 3 길 50

- **GRC**

(13553) 경기도 성남시 분당구 분당수서로 477

- **ARS : +82-1588-9997 (A/S center)**

- **E-mail : robotics@hyundai-robotics.com**

