

Simple  friendly

 **Kawasaki**

川崎机器人控制器
E 系列

**KLogic/KLadder
操作手册**

Robot

川崎重工业株式会社

前言

本手册介绍了川崎机器人E系列控制器中的内置顺序控制软件KLogic和与其一同使用的编程软件KLadder。

本手册尽可能详细地介绍了使用本功能的标准操作方法。但无论怎样，都不可能把所有需避免的可能操作、条件或情况都完全地描述出来。所以，在操作中碰到任何未描述的问题或情况时，请联络川崎公司。请参阅本手册封底上的联系信息。

在仔细阅读本手册的同时，还必须熟读与机器人一起发运的E控制器的基本手册（包括安全手册）。只有当全部手册完整阅读并充分理解后，才能使用机器人。

-
1. 本手册并不构成对使用机器人的整个应用系统的担保。因此，川崎公司将不会对使用这样的系统而可能导致的事故、损害和(或)与工业产权相关的问题承担责任。
 2. 川崎公司郑重建议:所有参与机器人操作、示教、维护、维修、点检的人员，预先参加川崎公司准备的培训课程。
 3. 川崎公司保留未经预先通知而改变、修订或更新本手册的权利。
 4. 事先未经川崎公司书面许可，对本手册整体或其中的任何部分，均不可进行任何形式的再版、重印、翻印、转载或复制。
 5. 请把本手册小心存放好，使之保持在随时备用状态。如果机器人重新安装或移动到另一个地点，或者卖给另一个使用者，请务必将本手册与机器人放在一起。一旦出现丢失或严重损坏，请和您的川崎公司代理商联络。

Copyright © 2014 Kawasaki Heavy Industries Ltd. All rights reserved.

川崎重工 版权所有

符号

在本手册中，需要加以特别注意的事项带有下列符号。

为确保机器人的正确安全操作、防止人员伤害和财产损失，请遵守下述符号方框内的安全信息。

⚠ 危险

不遵守指出的内容，可导致即将临头的伤害或死亡。

⚠ 警告

不遵守指出的内容，可能会导致伤害或死亡。

⚠ 小心

不遵守指出的内容，可导致人身伤害和/或机械损伤。

[注意]

指示有关机器人规格、处理、示教、操作和维护的注意事项。

⚠ 警告

1. 手册中所给出的图表、顺序和细节说明的精确性和有效性，不能被证实是绝对正确的。因此，在使用本手册进行任何工作时，必须投于最大的注意力。如果出现了没有说明的问题，请与川崎机器系统公司联系。
2. 本手册描述的有关安全的内容适用于各单独的工作，不能应用于所有的机器人工作。为了安全地进行各项工作，请仔细阅读和充分理解安全手册、全部相关法律、规章制度、以及各章节的所有安全说明，并在实际工作中准备合适的安全措施。

凡例

1. 操作用硬件键和开关（按钮）

为了进行各种操作，在 D/E 系列控制器的操作面板和示教器上设有各种硬件键和开关。在本手册中，各硬件键和开关的名称用下面的方框框出。另外，为简化表达，“XXX”键或“XXX”开关的键或开关（按钮）等术语会被省略。当需要同时按两个或更多键时，如同下面的例子，按这些键的顺序通过“+”号来表示。

例

登录: 表示“登录”硬件键。

TEACH/REPEAT (示教/再现): 表示操作面板上的“TEACH/REPEAT（示教/再现）”模式转换开关。

A + 菜单: 表示按住 A 然后按 菜单 键。

2. 操作用软件键和开关

D/E 系列控制器为各种规格和情况的不同种类的操作，提供了显示在示教器画面上的软件键和开关。本手册中，软件键和开关的名称将用尖括号“<>”括起来。另外，为简化表达，“XXX”键或“XXX”开关的键或开关（按钮）等术语会被省略。

例

<ENTER>（输入）：表示显示在示教器画面上的“ENTER（输入）”键。

<下一页>：表示示教器画面上的“下一页”键。

3. 选择项目

一个选项通常需要从示教器画面的菜单或下拉式菜单中选择。而在本手册中，这些菜单项目的名称将被方括号[XXX]括起来。

例

[辅助功能]: 表示一个菜单中的选项“辅助功能”。要选择它，用箭头键移动光标到相应项目上，然后按 ↵ 键。为了详细描述，此过程必须每次都描述一遍，但为了简化表达，“选择[XXX]选项”将被用来替代详细描述。

目 录

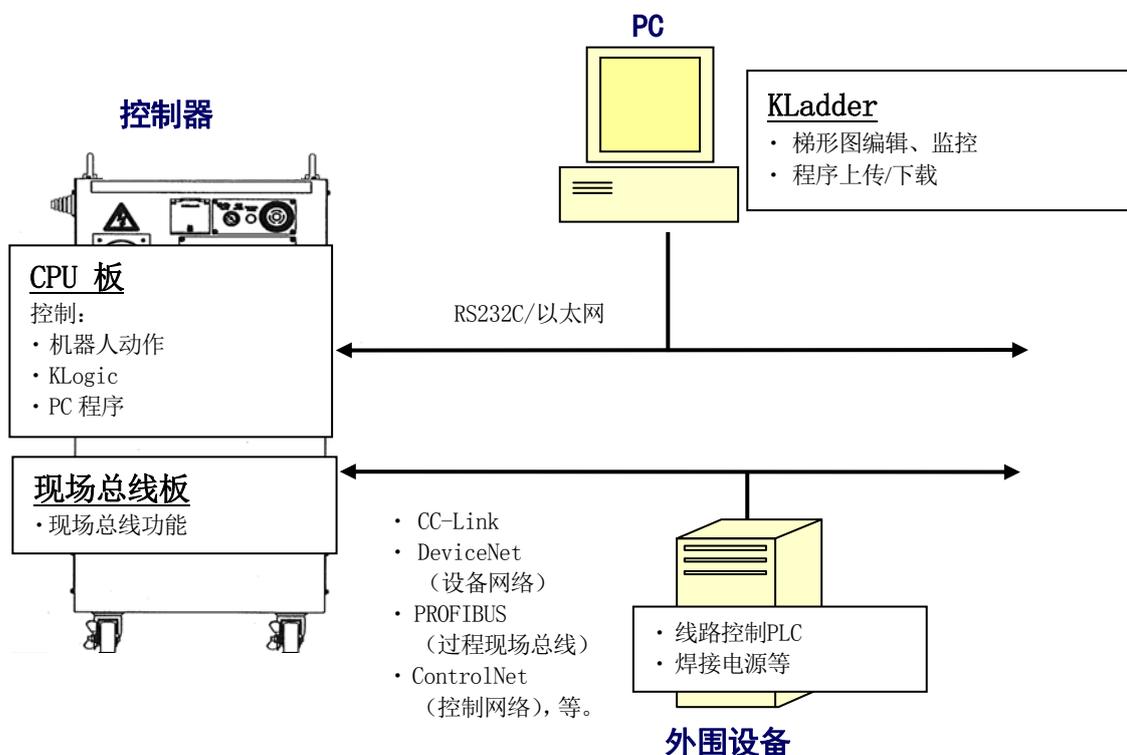
前言	i
符号	ii
凡例	iii
1.0	KLogic/KLadder 概要	1-1
1.1	KLogic 概要	1-1
1.1.1	KLogic 规格	1-2
1.1.2	KLogic 的画面显示	1-3
1.2	KLadder 概要	1-4
1.2.1	KLadder 功能	1-4
1.2.2	KLadder 规格	1-4
2.0	KLogic	2-1
2.1	基本运算部	2-1
2.1.1	用于基本运算部的助记符	2-1
2.1.2	基本运算部的程序	2-1
2.1.3	用于基本运算部的信号	2-2
2.1.4	基本运算程序执行中的改写	2-3
2.2	数值运算部	2-4
2.2.1	数值运算部的作用	2-4
2.2.2	数值运算部的执行	2-4
2.2.3	数值运算部的程序	2-4
2.2.4	用于数值运算部的信号	2-4
2.3	现场总线的信号分配	2-5
2.4	KLogic 显示器	2-7
2.4.1	Klogic 显示器的起动方法	2-7
2.4.2	Klogic 显示器画面的详细说明	2-9
2.4.2.1	基本运算部显示器	2-9
2.4.2.2	数值运算部显示器	2-10
2.5	KLogic 梯形图显示	2-11
2.5.1	KLogic 梯形图显示的起动方法	2-12
2.5.2	KLogic 梯形图显示的详细	2-13
2.5.2.1	梯形图显示画面的详细	2-13
2.5.2.2	信号检索画面	2-14
2.5.2.3	步骤检索画面	2-17
2.6	AS 语言	2-18

2.6.1	基本运算部的指令/命令	2-19
2.6.1.1	监控指令	2-19
2.6.1.2	程序命令	2-26
2.6.1.2.1	基本运算命令	2-26
2.6.1.2.2	应用运算命令	2-37
2.6.2	数值运算部的指令/命令	2-77
2.6.2.1	监控指令	2-77
2.6.2.2	基本运算部和数值运算部间的接口用命令	2-81
3.0	KLadder	3-1
3.1	安装和连接	3-1
3.1.1	KLadder 硬件配置要求	3-1
3.1.2	KLadder 安装程序	3-1
3.1.3	与控制器的连接方法	3-2
3.1.3.1	使用 RS232C 连接	3-2
3.1.3.2	使用以太网的连接	3-3
3.1.4	通信模块的设定	3-3
3.2	用 KLadder 编程	3-5
3.2.1	打开文件	3-5
3.2.2	建立新回路（插入新块）	3-5
3.2.3	配置符号	3-6
3.2.3.1	梯形图构成	3-8
3.2.4	更改符号	3-9
3.2.5	画并联回路	3-10
3.2.5.1	在同一块内插入第二行	3-10
3.2.5.2	移动符号 1 列	3-11
3.2.5.3	画 OR 图	3-11
3.2.6	移动/复制数据	3-12
3.2.7	转换数据	3-13
3.2.8	输入/编辑说明	3-14
3.2.9	输入/编辑寄存器	3-15
3.2.10	块编辑	3-16
3.2.10.1	编辑全部信号编号	3-16
3.2.10.2	更改应用命令的操作数	3-18
3.2.10.3	转换闭合和断开触点	3-20
3.2.11	块比较	3-21
3.2.12	检索	3-22
3.2.13	跳转	3-23
3.2.14	使用书签跳转	3-23

3.2.15	信号状态	3-24
3.2.16	触点表	3-24
3.2.17	程序检查	3-25
3.3	RC (机器人控制器) 菜单	3-27
3.3.1	从 RC 读取文件	3-27
3.3.1.1	程序的读取	3-27
3.3.1.2	说明的读取	3-28
3.3.2	写入到 RC (机器人控制器)	3-28
3.3.2.1	程序的写入	3-28
3.3.2.2	说明的写入	3-29
3.3.3	RC 的动作状态	3-30
3.3.4	RC 程序的停止	3-30
3.3.5	RC 程序的起动	3-31
3.3.6	I/O 信号的强制 ON/ OFF	3-31
3.4	监控菜单	3-34
3.4.1	RC 监控开始	3-34
3.4.2	寄存器·I/O 信号显示器	3-35
3.5	程序文件管理	3-36
3.5.1	数据的保存	3-36
3.5.2	文件和程序的二进制比较	3-36
3.5.3	说明的输入	3-36
3.6	选项菜单	3-37
3.6.1	自定义	3-37
3.6.1.1	字体的自定义	3-37
3.6.1.2	颜色的自定义	3-38
3.6.1.3	显示的自定义	3-40
3.6.2	KLadder 设定	3-41
3.6.2.1	文件设定	3-41
3.6.2.2	操作设定	3-42
3.6.2.3	设定快捷键	3-43
3.6.2.4	语言	3-44
3.7	打印文件	3-45
3.7.1	打印方法	3-45
3.7.2	回路图一览表	3-46
3.7.3	命令一览表	3-48
3.7.4	地址说明一览表	3-49
3.7.5	触点表一览表	3-50
3.7.6	信号一览表	3-51

1.0 KLogic/KLadder概要

本选件无需在外部设置控制面板或打印电路板，可通过在 E 控制器内装入 KLogic 软件来实现机器人系统的程序顺序控制。通过使用专用编程软件 KLadder，利用梯形图就可以编制顺序控制程序。



1.1 KLogic 概要

KLogic 通过使用机器人控制用 CPU 处理的一部分进行顺序控制。无需特别的硬件，在维护方面也省事。

由于 KLogic 软件的顺序控制程序是用 AS 语言编写的，因此，可以与机器人程序一同被管理。另外，通过使用多种规格的现场总线可以控制外围设备。

有关顺序控制的 AS 语言指令/命令，请参阅 2.6 AS 语言。

1.1.1 KLogic 规格

KLogic 由执行高速逻辑运算的基本运算部和执行四则运算的数值计算部组成。

基本运算部的规格如下：

		标准 KLogic	
编程语言		梯形图	
采样周期		自动设定为每 2 ms	
基本命令处理时间*		2 μs/ 步	
程序容量	梯形图	10000 步	
命令	逻辑运算 (基本命令)	13 种	
	(应用命令)	37 种	
信号种类	输入	2048	&X0000-&X07FF
	输出	2048	&Y0000-&Y07FF
	内部中间继电器	2048	&M0000-&M07FF
	保持型继电器	512	&K0000-&K01FF
	计时器	256	&T0000-&T00FF
	计数器	256	&C0000-&C00FF
	数据寄存器 **	2048	&D0000-&D07FF
	特殊继电器	256	&V0000-&V00FF
程序存储器		压缩闪存	
运算		在机器人主 CPU 中处理	

注* 应用运算命令的处理时间，请参阅 2.6.1.2.2 应用运算命令中的各命令。

注** 包括特殊寄存器 (&D0700-&D07FF)

数值计算部的规格如下：

		KLogic (数值计算部)
编程语言		AS 语言
程序执行方式		特殊 PC 程序执行方式
程序执行周期		程序中执行全部命令的合计时间
运算		在机器人主 CPU 中处理
程序存储器		压缩闪存

1.1.2 KLogic 的画面显示

KLogic 的执行状态和基本运算部的梯形图可以在示教器画面上显示。有关详细的操作方法，请参阅 2.4 KLogic 显示器和 2.5 KLogic 梯形图显示。

1.2 KLadder 概要

KLadder 是用于 Windows 系统计算机的，KLogic 专用的梯形图编程软件。根据梯形图编写顺序控制程序。

1.2.1 KLadder 功能

KLadder 的主要功能如下：

1. 使用梯形图编制/编辑程序
2. 向/从 KLogic 写入/读出程序
3. 信号状态的显示器

1.2.2 KLadder 规格

KLadder 的规格如下：

PC 规格	名称	KLadder
	OS	Windows95 以上
	CPU	Pentium 以上
	内存	推荐使用 32 M 以上
	硬件	15 M byte 以上
功能	梯形图编制	
	梯形图编辑	
	编辑	
	梯形图程序的传送	
	梯形图程序的执行	
	梯形图显示器	
与控制器的连接	RS232C/以太网	

有关在 PC 中安装及操作 KLadder 的方法，请参阅 3.0 KLadder。

2.0 KLogic

本章介绍 KLogic 的构成及在 KLogic 中使用的 AS 语言。

2.1 基本运算部

KLogic 由以下两个软件部分组成：

基本运算部……进行逻辑运算

数值运算部……进行四则运算等数值运算

有关基本运算部在本节介绍。有关数值运算部的介绍，请参阅 2.2 数值运算部。

2.1.1 用于基本运算部的助记符

用于基本运算部的助记符都是以“LSQ”开始的命令，并且作为 AS 语言处理。

例 LSQ STR, &X0000
 LSQ OUT, &Y0000

2.1.2 基本运算部的程序

基本运算部执行的程序必须使用“lsqpg”的名字。除名字固定为 lsqpg，并且程序命令以字母“LSQ”开始外，与机器人程序的执行方法相同。并且，该程序与机器人程序一样可以编辑、保存/加载。

通过使用专用的监控指令，控制基本运算部（以下称“基本运算程序”）所编制的程序的执行。（详情，请参阅 2.6 AS 语言。）

例 LSQEXECUTE LSQREWRITE
 LSQABORT LSQSTATUS 等

基本运算程序命令有 2 种：1. 基本运算命令，2. 应用运算命令。基本运算命令主要执行逻辑运算，应用运算命令执行计时、计数及数值输入。各命令的详细介绍请参阅 2.6 AS 语言。

基本运算程序与其他机器人程序不同，为了在 RAM（基本运算部不能处理助记符形式（lsqpg）的程序）上能高速处理从助记符形式转换到二进制形式。但是，程序以助记符形式保存，以便注意不要删除/更改必要的程序。（程序不以二进制形式保存，但每当控制器电源打开时由助记符形式转换。）

2.1.3 用于基本运算部的信号

用于基本运算部的信号以不同于标准机器人信号 (OX/ WX 等) 的形式编写。下表为标准 and KLadder 信号间的对应关系。(两种信号间的关系随各系统条件的不同而不同, 例如通用现场总线标准。有关详情, 请参阅 2.3。)

用于基本运算部信号种类间的对应关系如下:

信号种类	格式及上下限	信号数
输入	&X0000-&X07FF	2048
输出	&Y0000-&Y07FF	2048
内部中间继电器	&M0000-&M07FF	2048
保持型继电器	&K0000-&K01FF	512
计时器	&T0000-&T00FF	256
计数器	&C0000-&C00FF	256
数据寄存器	&D0000-&D07FF	2048
特殊继电器	&V0000-&V00FF	256

数据寄存器保存有每个数据 16 位的信息。其他信号保存有每个数据 1 位的信息。在数据寄存器中 &D0700 - &D07FF 为特殊寄存器, 与下表中所示的特殊继电器可以输出预先决定的数据, 如机器人状态, 当前时刻等。

特殊继电器一览表

编号	内容	
V0000	通常为 ON	不管运算状态如何总是 ON
V0001	通常为 OFF	不管运算状态如何总是 OFF
V0002	用户定义计时器 1	LSQ USC1 的输出结果
V0003	用户定义计时器 2	LSQ USC2 的输出结果
V0004	机器人错误	输出机器人错误*
V0005	进位标志	当 LSQ+, LSQ W+ 的运算结果进位时, 为 ON
V0006	借位标志	当 LSQ-, LSQ W- 的运算结果借位时, 为 ON
V0007	零标志	LSQ+, W+, -, W-, X, WX, /, W/ 的运算结果为零时, 为 ON
V0008	错误标志	当在 LSQ/, LSQ W/ 分配的寄存器为 0 时, 为 ON

注* 全部机器人错误中, 仅当错误指示灯亮灯时, 输出错误。

错误代码输出到特殊寄存器 D0700 中。错误发生时刻输出到特殊寄存器 D0701-D0703 中。

特殊寄存器一览表

编号	内容		
D0700	机器人错误代码*		
D0701	错误发生时间	(分/秒)	BCD 显示 (用 1 个字节表示二进制数位。)
D0702	错误发生时间	(日/时)	BCD 显示 (用 1 个字节表示二进制数位。)
D0703	错误发生时间	(年/月)	BCD 显示 (用 1 个字节表示二进制数位。)
D0704	当前时间	(分/秒)	BCD 显示 (用 1 个字节表示二进制数位。)
D0705	当前时间	(日/时)	BCD 显示 (用 1 个字节表示二进制数位。)
D0706	当前时间	(年/月)	BCD 显示 (用 1 个字节表示二进制数位。)
D0707	程序名	“pg” 后的编号。如果程序的前两个字符没有“pg” 的话, 为 FFFFh。	
D0708	步骤编号	当程序不存在时, 为 0。	
D0709	扫描时间	基本运算部	当前值
D070A	扫描时间	基本运算部	错误阈值
D070B	扫描时间	数值运算部	当前值

注* 在错误代码的开头表示错误类型的字母字符, 由以下数字来代替。

Pxxxx: 1xxxx Exxxx: 3xxxx
Wxxxx: 2xxxx Dxxxx: 4xxxx

2.1.4 基本运算程序执行中的改写

执行基本运算程序时, 可以将其改写为新程序。(请参阅 LSQ REWRITE 指令。)

改写流程如下:

1. 检查新程序 (lsqpg) 的语法是否正确。
2. 如果发现错误, 那么将显示错误信息。当前 (原) 程序继续执行。
3. 如果没有错误, 当前 (原) 程序执行到最后一步, 然后改写为新程序。
4. 在当前程序改写为新程序后, 立刻执行新程序。如果在新程序中有 LSQ START 命令, 那么从该步骤的下一步开始执行程序。

2.2 数值运算部

2.2.1 数值运算部的作用

数值运算部处理程序，该程序需要在一定周期内执行，不需高速处理。主要是由数值运算部进行四则运算。

2.2.2 数值运算部的执行

以与PC程序相同的方法来处理数值运算程序，并通过分配机器人主CPU处理时间来执行。因此，用于PC程序的大部分AS语言也都可以用在数值运算程序中。机器人动作和其他PC程序的执行不影响数值运算程序的执行。执行周期由程序中各命令的执行时间的总和来决定。

2.2.3 数值运算部的程序

数值运算部所编制程序的执行用专用命令来控制。

例

NUMEXECUTE
NUMABORT
NUMCONTINUE
NUMSTATUS
NUMKILL

另外，程序名固定为 numpg。数值运算程序不能用 KLadder 来编制。

2.2.4 用于数值运算部的信号

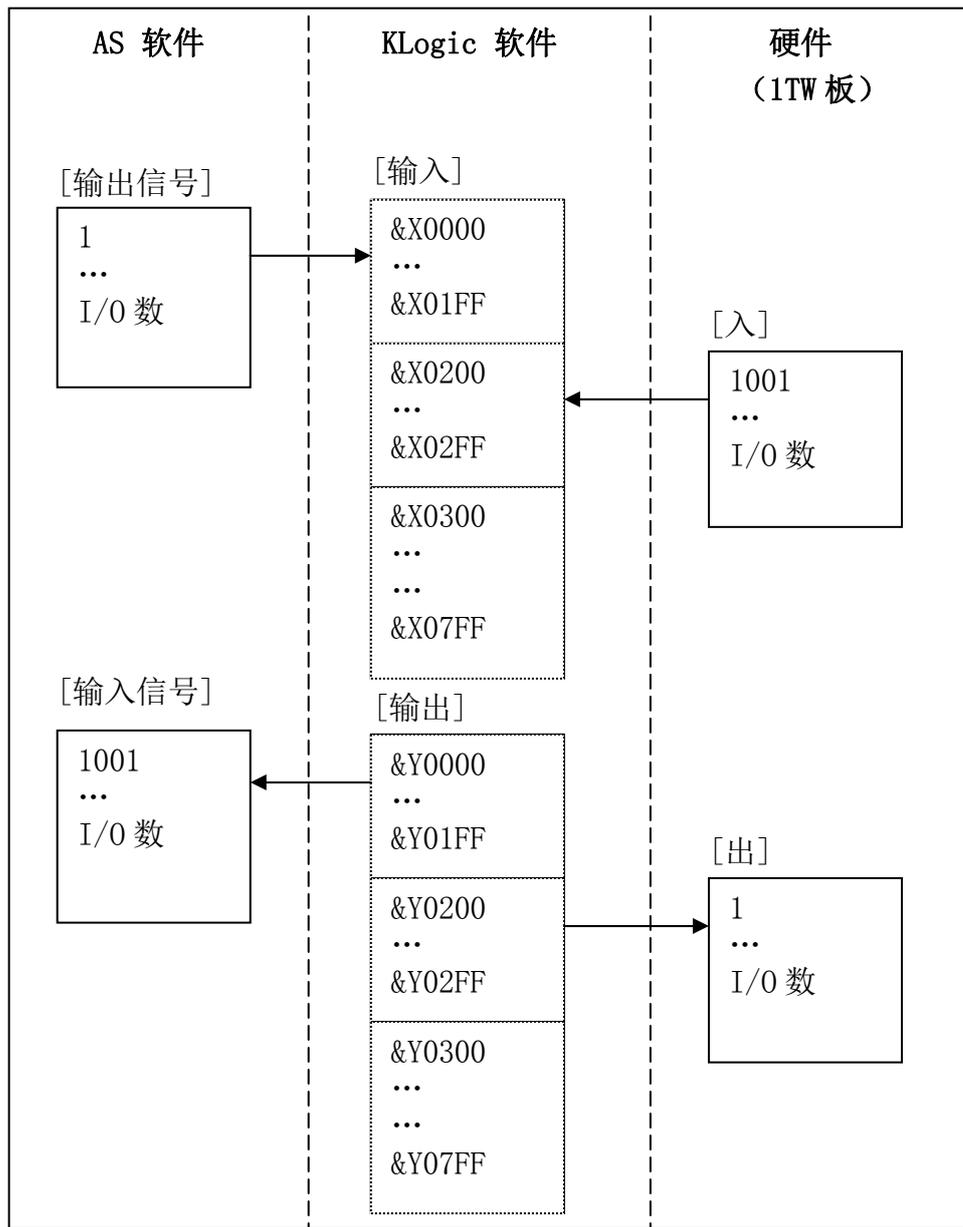
由于数值运算程序是机器人控制器 CPU 中处理的 PC 程序，所以其所使用的信号与机器人的相同。用于数值运算部 和基本运算部的信号间的对应关系如下表所示。

		数值运算部	基本运算部	对应关系
输入	标记法	信号	&Xnnnn	参阅 2.3 中的表
	范围	1-960	&X0000-&X03C0	
输出	标记法	信号	&Ynnnn	参阅 2.3 中的表
	范围	1-960	&Y0000-&Y03C0	
数据寄存器	标记法	变量	&Dnnnn	用 AS 语言读写
	范围		&D0000-&D07FF	

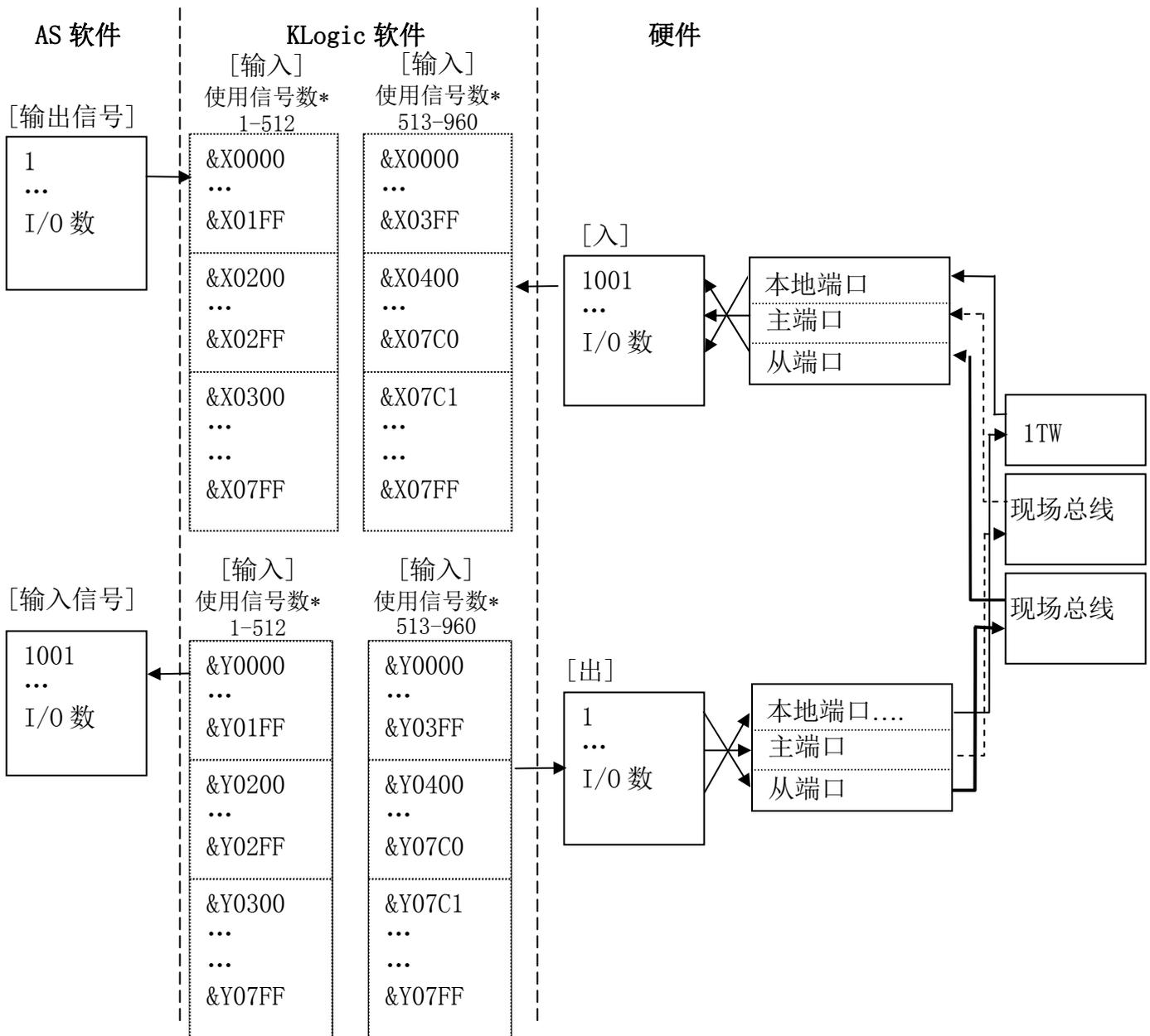
2.3 现场总线的信号分配

用于基本运算部的信号与用于机器人的信号（OX/ WX、SIG 否 L 等）之间的对应关系如下所示。此对应关系随通用现场总线的规格的不同而不同。

标准



通用现场总线 I/O 选项 ON



注* 使用信号数:

* ZSIG 否 L 指令的 DO / MAS 和 SLA 的合计数。

* 辅助 0611 I/O 信号数中的外部输出信号、主端口信号和从端口信号的总数。

2.4 KLogic 显示器

KLogic 显示器是在示教器的液晶显示屏上显示执行状态的功能。通过本功能可以显示基本运算部和数值运算部的执行状态和运算时间。

2.4.1 KLogic 显示器的起动方法



在示教器上按<显示器 1>或<显示器 2>显示左图的显示器菜单。



从菜单中选择[80. KLogic 显示器]，如左图所示另一菜单出现。从此菜单中选择[1. 基本运算]或[2. 数值运算]。



选择[1. 基本运算] 如左图所示基本运算部出现。



选择 [2. 数值运算] 如左图所示数值运算部出现。

2.4.2 KLogic 显示器画面的详细说明

2.4.2.1 基本运算部显示器

基本运算部显示器的各项目如下：

显示1 :KLogic 显示 :基本运算	
基本运算程序状态	: 程序执行中
最长执行周期	: 1000 ms
程序执行周期	: 2 ms
LSQPG	: 有

[基本运算程序状态]

显示基本运算程序的执行状态。根据执行状态显示“程序执行中”或“程序未执行”。

[最长执行周期]

显示完成一个执行周期的最长时间。如果完成此周期的实际所需时间超出设定时间的话，错误出现或程序停止。用 LSQ_CYCMAX 指令更改设定时间。单位为 ms。

[程序执行周期]

显示当前执行程序的执行时间。单位为 ms。

[LSQPG]

显示基本运算部程序 lsqpg 是否存在。lsqpg 存在时，显示为“有”，lsqpg 不存在时，显示为“无”。

2.4.2.2 数值运算部显示器

数值运算部显示器的各项目如下。

显示1 :KLogic 显示 :数值运算	
数值运算程序状态	: 程序执行中
程序执行时间	: 9 ms
程序执行时间(平均)	: 7 ms
NUMPG	: 有

[数值运算程序状态]

显示数值运算程序的执行状态。根据程序的执行状态显示“程序执行中”或“程序未执行”。

[程序执行时间]

显示完成当前执行程序所需时间。单位为 ms。

[程序执行时间（平均）]

显示完成当前执行程序所需平均时间。单位为 ms。

[NUMPG]

显示数值运算部程序 numpg 是否存在。numpg 存在时，显示为“有”， numpg 不存在时，显示为“无”。

2.5 KLogic 梯形图显示

在示教器画面中，与 KLadder 的显示相同，KLogic 的基本运算部可以用梯形图来显示。显示各触点的状态，该状态可以用于检查程序执行状态。



2.5.1 KLogic 梯形图显示的启动方法



在示教器上按[菜单] 如左图所示下拉式菜单出现。



在菜单中选择 [KLogic 梯形图显示]。显示执行中或刚刚执行的基本运算部程序(1sqpg)的梯形图。

如果基本运算部程序不存在，或存在但尚未执行的话，那么将出现空白画面。

2.5.2 KLogic 梯形图显示的详细

2.5.2.1 梯形图显示画面的详细



画面上的各按钮功能如下：

<上一页>，<下一页>

显示梯形图程序的前 4 行或后 4 行。按示教器上的 **S** 键+ **↑** **↓** 也与这两个按钮有相同的功能。

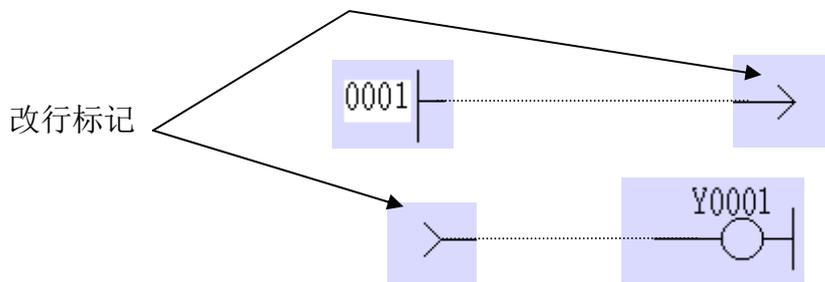
<信号检索>

显示信号检索画面。有关详情，请参阅 2.5.2.2 信号检索画面。

<步骤检索>

显示步骤检索画面。有关详情，请参阅 2.5.2.3 步骤检索画面。

1. 每次在画面上显示 4 行。
2. 1 行最大显示 11 块（回路）。超过 11 块的在下一行显示。当改行时，如下图所示在改行前回路的末端和改行后回路的始端显示改行标记。



3. 在每行左端的白底处显示步骤编号。
4. 在各行的末端显示回路线圈。在各行的右端显示线圈说明。
5. 在各触点上显示该触点说明。说明最多以 18 个字符来表示。
6. 在梯形图的最后块之后，显示[EOF]。
7. 当信号为 ON 的状态时，线圈、触点、计数器和计时器 (C0001, T0001, 等) 显示为红字。
8. 梯形图显示时，如果 lsqpg 更改的话，确认更新的对话框出现。确认更新后，在梯形图上显示更新程序。
9. 如果程序包括不能转换为梯形图的命令的话，只有命令之前的程序数据转换，在包括不能转换为梯形图的命令的行的末端显示???

2.5.2.2 信号检索画面

根据信号数检索梯形图内的触点和线圈。



按梯形图显示画面下部的
<信号检索>。



左图的弹出画面出现。
为了输入信号编号，在弹出画面中按<输入字符>按钮。



如左图的键盘画面出现。
用显示的键盘画面输入信号编号。

输入以表示信号种类的字母 (X, Y, M, K, T, C, D, V) 和 4 位以下的 16 进制数的信号编号。输入信号种类的字母和 1 位以上的 16 进制数来检索以该字母和数开始的触点和线圈。必须输入信号种类和 16 进制数，否则只输入其中的 1 个将无效。

例 输入 X0。 → 检索 X0000 - X0FFF 的所有信号。



按 **Enter** 键将从键盘画面返回到原弹出画面。

按<检索执行>，<下检索>，<上检索>时，信号检索开始。

<检索执行>

从梯形图的顶部开始检索，直到找到最初的应用步骤。并且，显示该应用步骤。

<上检索>

从步骤编号大到小的顺序检索，直到找到最初的应用步骤。并且，显示该应用步骤。

<下检索>

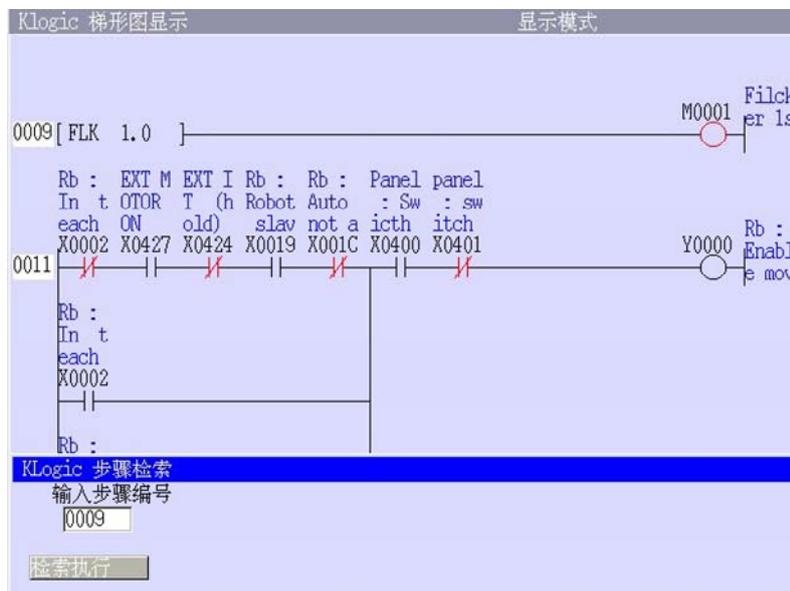
从步骤编号小到大的顺序检索，直到找到最初的应用步骤。并且，显示该应用步骤。

2.5.2.3 步骤检索画面

显示指定步骤编号的步骤。



按梯形图显示画面的下部的<步骤检索>。



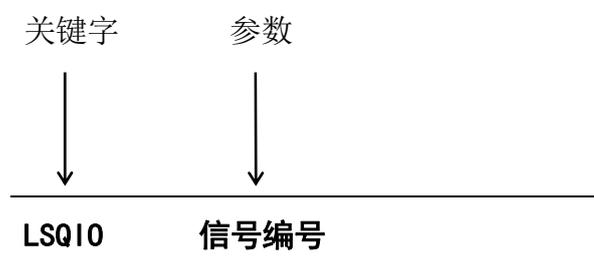
步骤检索的弹出画面出现。输入步骤编号并按<检索执行>，包括指定步骤在内的回路在画面的上部的第一行出现。

2.6 AS 语言

本节介绍 KLogic 用顺序控制专用的指令和命令。

基本运算程序可以用 AS 语言编写。但，一般建议用 KLadder 制作梯形图，然后转换为 AS 程序。另外，数值运算程序用在本节介绍的命令和其他 PC 程序中可使用的 AS 语言命令编制。

示例



可以省略输入用 标记的参数值。

必须在关键字和参数之间输入一个空格。

示例中 代表回车键 (Enter 键)。

2.6.1 基本运算部的指令/命令

2.6.1.1 监控指令

用于控制基本运算程序 (lsqpg) 的监控指令如下并在下页后介绍。

LSQEXECUTE	从第一步开始执行。
LSQABORT	执行中停止 lsqpg。
LSQREWRITE	改写 lsqpg 的内容。
LSQSTATUS	显示 lsqpg 的当前执行状态。
LSQIO	显示各种信号的当前状态/当前值。
LSQ_CYCMAX	设定 lsqpg 的最长执行周期。
LSQINIT	清除信号状态。

监控指令

LSQEXECUTE

功能

执行基本运算部的程序(lsqpg)。

说明

基本运算程序的程序名固定为“lsqpg”。如果找不到程序(lsqpg) 或该程序不存在, 那么错误产生:

(P1038) 程序不存在。

如果基本运算程序已经开始执行, 那么错误产生:

(P2055) [LSEQ] 程序已在运行中。

当使用不存在的信号编号等执行程序时, 错误产生:

[LSEQ] 编译错误 程序 LSQPG

步骤 2 内存步骤 2

(P2056) [LSEQ] 信号编号超出指定范围。

示例

```
>LSQEXECUTE
```

监控指令

LSQABORT

功能

当基本运算程序 (lsqpg) 正在运行时, 停止其运行。

示例

```
>LSQABORT
```

监控指令

LSQREWRITE

功能

改写当前运行中的基本运算程序 (lsqpg) 的内容。

说明

当基本运算程序运行时，改写其内容。程序完成运行后，更改生效。

为了改写程序，首先改写程序(lsqpg)，然后执行此指令。

在运行改写前的程序最后一步之后，立刻在修改的程序上改写。

如果在修改的程序中有文法错误的话，那么显示错误信息并继续执行改写前的原程序。

当在修改的程序中有 LSQ START 命令时，在此命令的下一步骤继续执行程序。

示例

>LSQREWRITE

监控指令

LSQSTATUS

功能

显示基本运算部的当前状态。

说明

1.	[LSEQ] 程序执行状态：执行中
2.	设定状态： 最长执行周期(ms)： 1000 程序执行周期 (ms)： 2
3.	程序 LSQPG 有

1. [LSEQ] 程序执行状态：

显示基本运算部的当前程序执行状态。根据状态显示信息“执行中”或“未执行”。

2. 设定状态：

显示基本运算部的设定。

最长执行周期(ms)：

显示完成一个执行周期所需的最长时间。如果完成此周期所需的实际时间比此设定时间长的话，那么错误出现。用 LSQ_CYC MAX 指令设定最长时间。

程序执行周期 (ms)：

显示执行程序 lsqpg 所需的时间。

3. 程序

显示基本运算程序 lsqpg 是否存在。根据程序是否存在，显示信息“LSQPG 有”或“LSQPG 无”。

 监控指令

LSQIO 信号编号

LSQIO/X 信号编号

LSQIO/Y 信号编号

LSQIO/M 信号编号

LSQIO/K 信号编号

LSQIO/T 信号编号

LSQIO/C 信号编号

LSQIO/CNTH 信号编号

LSQIO/D 信号编号

LSQIO/V 信号编号

功能

显示基本运算程序中使用的信号种类的当前状态。

参数

信号编号

可指定的信号编号随信号种类不同而不同。各信号种类显示 32 个信号编号。

例 1: 显示 1 - 32, 1001 - 1032, 2001 - 2032.

2: 显示 33 - 64, 1033 - 1064, 2033 - 2064

信号种类	限定符	格式及上下限	信号数	信号编号
输入	/X	&X0000-&X07FF	2048	1-64
输出	/Y	&Y0000-&Y07FF	2048	1-64
内部中间继电器	/M	&M0000-&M07FF	2048	1-64
保持型继电器	/K	&K0000-&K01FF	512	1-16
计时器	/T	&T0000-&T00FF	256	1-8
计数器	/C	&C0000-&C00FF	256	1-8
当前计数器值	/CNTH	-	256	1-4
数据寄存器	/D	&D0000-&D07FF	2048	1-32
特殊继电器	/V	&V0000-&V00FF	256	1-8

当省略信号编号时，自动输入 1。

说明

数据寄存器和当前计数器值以 16 位格式（显示为 16 进位数）来显示。其他信号种类以继电器状态（1 位）来显示的。

信号种类用限定符（/X, /Y, /M 等）来分类。如果没有限定符输入，那么显示输入（/X）、输出（/Y）、内部中间继电器（/M）、保持型继电器（/K）限定符的状态。

监控指令

LSQ_CYCMAX 最长执行周期

功能

设定基本运算部的最长执行周期。

参数

最长执行周期

设定基本运算程序所执行的最长周期。

单位：[ms]。输入范围：2-1000。

说明

当基本运算程序的执行时间比本指令设定的最长周期时间长时，以下错误产生：

(P2072) [LSEQ] 最大的执行循环数。

实际可接受周期为 8 ms 的倍数。对于不能被 2 整除的值，其商将作为设定值，并且余数被省略。

当前设定值可以用 LSQSTATUS 指令确认。

示例

>LSQ_CYCMAX 200

将最长执行周期设定为 200 ms。

监控指令

LSQINIT

功能

清除输入 (&X) 以外的全部信号。

说明

清除信号状态。

基本运算正在执行时，如下错误产生：

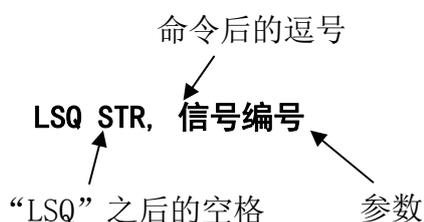
(P2055) [LSEQ] 程序已在运行中。

2.6.1.2 程序命令

2.6.1.2.1 基本运算命令

LSQ STR	开始指定闭合触点的运算。	执行时间:	2 μ s
LSQ STRN	开始指定断开触点的运算。	执行时间:	2 μ s
LSQ AND	串联连接指定闭合触点。	执行时间:	2 μ s
LSQ ANDN	串联连接指定断开触点。	执行时间:	2 μ s
LSQ OR	并联连接指定闭合触点。	执行时间:	2 μ s
LSQ ORN	并联连接指定断开触点。	执行时间:	2 μ s
LSQ ANDS	串联连接 2 个逻辑块。	执行时间:	2 μ s
LSQ ORS	并联连接 2 个逻辑块。	执行时间:	2 μ s
LSQ OUT	输出运算结果输出到指定的设备中。	执行时间:	2 μ s
LSQ SET	设定指定保持型继电器为 ON。	执行时间:	2 μ s
LSQ RST	复位指定保持型继电器为 OFF。	执行时间:	2 μ s
LSQ FST	无论任何条件都将运算结果打开 为 ON。	执行时间:	2 μ s
LSQ NOT	转换所保存的运算结果。	执行时间:	2 μ s

基本运算程序命令的格式.



各命令的说明如下:

步骤数: 表示每个命令可换算的步骤数。

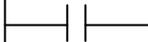
执行时间: 表示完成每个命令所需时间。

程序命令

LSQ STR, 信号编号

LSQ STRN, 信号编号

符号

LSQ STR 

LSQ STRN 

步骤数 1

执行时间 2 μs

功能

LSQ STR 开始指定闭合触点的运算。LSQ STRN 开始断开触点运算。将以信号编号指定的设备的 ON/OFF 信息作为运算结果保存到寄存器中。

参数

信号编号

指定回路中设置触点的地址。加“&”作为信号编号的最初字符。

信号种类								
X	Y	M	K	T	C	D	V	常数
是	是	是	是	是	是	否	是	否

示例

LSQ STR, &X0000

LSQ OUT, &Y0000



LSQ STRN, &X0001

LSQ OUT, &Y0001

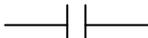


程序命令

LSQ AND, 信号编号

LSQ ANDN, 信号编号

符号

LSQ AND 

LSQ ANDN 

步骤数 1

执行时间 2 μs

功能

串联连接触点。LSQ AND 连接闭合触点，而 LSQ ANDN 连接断开触点。在指定设备的 ON/OFF 信息和寄存器的当前运算结果之间进行与运算(AND)，然后将新结果保存到寄存器中。

参数

信号编号

指定回路中设置触点的地址。加“&”作为信号编号的最初字符。

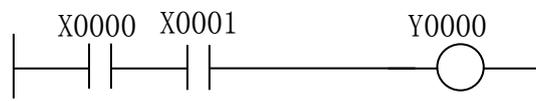
信号种类								
X	Y	M	K	T	C	D	V	常数
是	是	是	是	是	是	否	是	否

示例

LSQ STR, &X0000

LSQ AND, &X0001

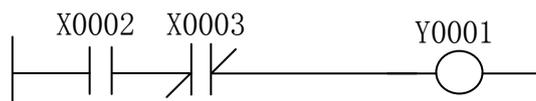
LSQ OUT, &Y0000



LSQ STR, &X0002

LSQ ANDN, &X0003

LSQ OUT, &Y0001



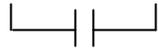
程序命令

LSQ OR, 信号编号

LSQ ORN, 信号编号

符号

LSQ OR



LSQ ORN



步骤数 1

执行时间 2 μs

功能

并联连接触点。LSQ OR 连接闭合触点，而 LSQ ORN 连接断开触点。在指定设备的 ON/OFF 信息和寄存器的当前运算结果之间进行或运算 (OR)，然后将新结果保存到寄存器中。

参数

信号编号

指定回路中设置触点的地址。加“&”作为信号编号的最初字符。

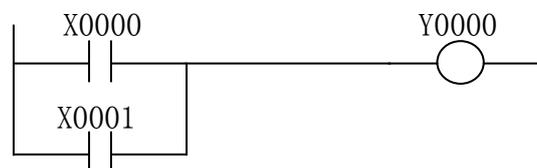
信号种类								
X	Y	M	K	T	C	D	V	常数
是	是	是	是	是	是	否	是	否

示例

LSQ STR, &X0000

LSQ OR, &X0001

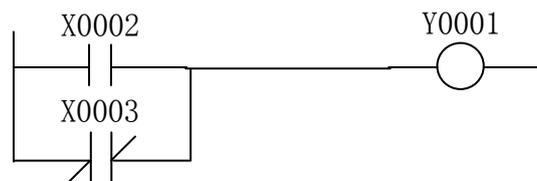
LSQ OUT, &Y0000



LSQ STR, &X0002

LSQ ORN, &X0003

LSQ OUT, &Y0001



程序命令

LSQ ANDS

符号

LSQ ANDS



步骤数 1

执行时间 2 μ s

功能

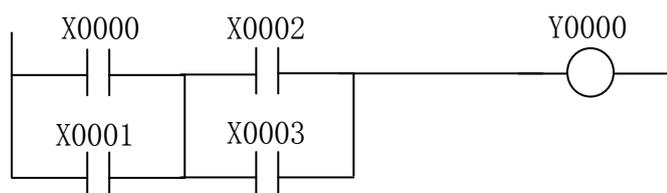
串联连接 2 个逻辑块。在 2 个逻辑块间进行与运算 (AND) ，然后将其结果保存到寄存器中。

示例

```

LSQ STR,  &X0000
LSQ OR,   &X0001
LSQ STR,  &X0002
LSQ OR,   &X0003
LSQ ANDS
LSQ OUT,  &Y0000

```

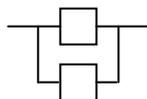


程序命令

LSQ ORS

符号

LSQ ORS



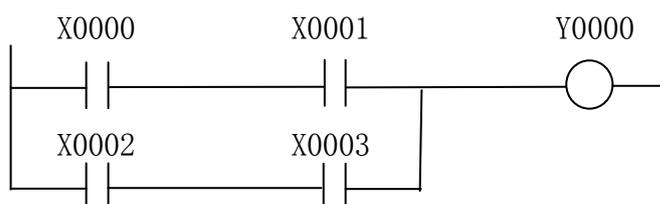
步骤数 1
执行时间 2 μ s

功能

并联连接 2 个逻辑块。在 2 个逻辑块间进行或运算 (OR) ，然后将其结果保存到寄存器中。

示例

LSQ STR, &X0000
LSQ AND, &X0001
LSQ STR, &X0002
LSQ AND, &X0003
LSQ ORS
LSQ OUT, &Y0000

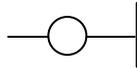


程序命令

LSQ OUT, 信号编号

符号

LSQ OUT



步骤数 1

执行时间 2 μs

功能

将保存在寄存器中的运算结果输出到指定的设备中。

参数

信号编号

指定保存运算结果的设备地址。加“&”作为信号编号的最初字符。

信号种类								
X	Y	M	K	T	C	D	V	常数
否	是	是	是	否	否	否	否	否

示例

LSQ STR, &X0000

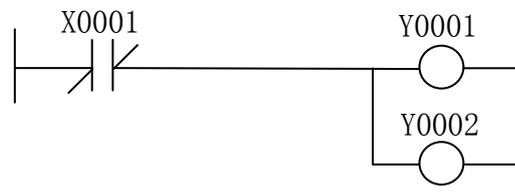
LSQ OUT, &Y0000



LSQ STRN, &X0001

LSQ OUT, &Y0001

LSQ OUT, &Y0002

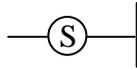


程序命令

LSQ SET, 信号编号

符号

LSQ SET



步骤数 1

执行时间 2 μs

功能

如果到此命令的运算结果为 ON (=1)，那么打开保持型继电器。

参数

信号编号

指定保持型继电器打开。

信号种类								
X	Y	M	K	T	C	D	V	常数
否	否	否	是	否	否	否	否	否

示例

LSQ STR, &X0000

LSQ SET, &K0000

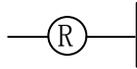


程序命令

LSQ RST, 信号编号

符号

LSQ RST



步骤数

1

执行时间

2 μ s

功能

如果到此命令的运算结果为 ON (=1)，那么关闭保持型继电器。

参数

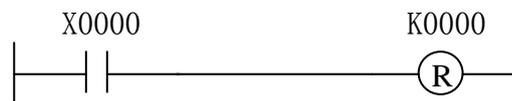
指定保持型继电器关闭。

信号种类								
X	Y	M	K	T	C	D	V	常数
否	否	否	是	否	否	否	否	否

示例

LSQ STR, &X0000

LSQ RST, &K0000

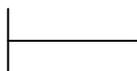


程序命令

LSQ FST

符号

LSQ FST



步骤数 1

执行时间 2 μ s

功能

无论任何条件都将运算结果打开 ON (=1)。

示例

LSQ FST

LSQ OUT, &Y0000



程序命令

LSQ NOT

符号

LSQ NOT



步骤数

1

执行时间

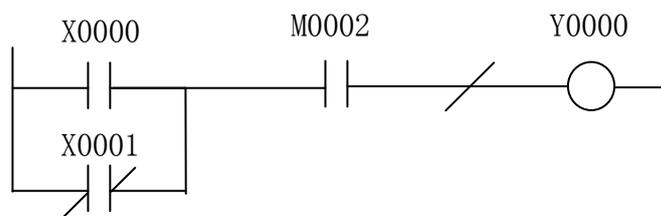
2 μ s

功能

转换保存在寄存器中运算结果的逻辑值（将 OFF 转换为 ON，将 ON 转换为 OFF。）

示例

LSQ STR, &X0000
LSQ ORN, &X0001
LSQ AND, &M0002
LSQ NOT
LSQ OUT, &Y0000



2.6.1.2.2 应用运算命令

LSQ TMR	为 100 ms 计时器设定计时器和计时器值。	执行时间	10 μ s
LSQ TMRS	为 100 ms 累时器设定计时器和计时器值。	执行时间	10 μ s
LSQ CNTH	增减计数器功能。	执行时间	10 μ s
LSQ PTS	在上升沿设定演算结果。	执行时间	10 μ s
LSQ NTS	在下降沿设定演算结果。	执行时间	10 μ s
LSQ MOVE	传送字节数据到寄存器。	执行时间	10 μ s
LSQ WMOVE	传送字数据到寄存器。	执行时间	10 μ s
LSQ MOV	传送 2 位十六进制的字节数据到寄存器。	执行时间	10 μ s
LSQ WMOV	传送 4 位十六进制的字数据到寄存器。	执行时间	10 μ s
LSQ MOVR	传送 3 位十进制的字节数据到寄存器。	执行时间	10 μ s
LSQ WMOVR	传送 5 位十进制的字数据到寄存器。	执行时间	10 μ s
LSQ WBMOV	传送字数据块。	执行时间	200 μ s
LSQ CMPEQ	比较块 1 的数据值是否等于块 2 的数据。	执行时间	10 μ s
LSQ CMPNEQ	比较块 1 是否不等于块 2 (!=)	执行时间	10 μ s
LSQ CMPGT	比较块 1 是否大于块 2 (>)	执行时间	10 μ s
LSQ CMPGE	比较块 1 是否大于等于块 2 (>=)	执行时间	10 μ s
LSQ CMPST	比较块 1 是否小于块 2 (<)	执行时间	10 μ s
LSQ CMPSE	比较块 1 是否小于等于块 2 (<=)	执行时间	10 μ s
LSQ WBCD	将 16 位二进制数据转换为 BCD 码数据, 并保存该结果。	执行时间	10 μ s
LSQ WBIN	将 BCD 数据转换为 16 位二进制数据, 并保存该结果。	执行时间	10 μ s
LSQ WBSRL	将字数据向左右位移的位数“n”。	执行时间	10 μ s
LSQ WUP	将字数据向高位位移。	执行时间	200 μ s
LSQ WDOWN	将字数据向低位位移。	执行时间	200 μ s
LSQ WBRST	复位位到 0。	执行时间	10 μ s
LSQ USC	定义用户设定块的 ON/OFF 功能。	执行时间	10 μ s
LSQ FLK	设定开/关闭的时间间隔。	执行时间	10 μ s
LSQ START	指定程序执行开始的步骤。	执行时间	10 μ s
LSQ +	8 位数据加法运算。	执行时间	10 μ s
LSQ W+	16 位数据加法运算。	执行时间	10 μ s
LSQ -	8 位数据减法运算。	执行时间	10 μ s

LSQ W-	16 位数据减法运算。	执行时间	10 μ s
LSQ X	8 位数据乘法运算。	执行时间	10 μ s
LSQ WX	16 位数据乘法运算。	执行时间	10 μ s
LSQ /	8 位数据除法运算。	执行时间	10 μ s
LSQ W/	16 位数据除法运算。	执行时间	10 μ s
LSQ AFLGCLR	设定应用操作命令标志的清除模式。	执行时间	2 μ s
LSQ AFLGCLRN	设定解除应用操作命令标志的清除模式。	执行时间	2 μ s

程序命令

LSQ TMR, 计时器, 计时器值

符号

LSQ TMR --[TMR 计时器 计时器值]

步骤数 2

执行时间 10 μ s

功能

到 LSQ TMR 命令为止的运算结果为 ON 时，开始计时，当计时器达到指定值时，打开指定设备 (&T0000—&T00FF)。

LSQ TMR 命令可以使用 2 种计时器：

1. 直接指定 100 ms 计时器
以常数直接指定值。该值以 100 ms 为单位。
2. 间接指定 100 ms 计时器
用保存在指定寄存器的值来指定值。该值以 100 ms 为单位。

参数

计时器

指定计时器。

计时器值

如果是直接指定计时器的话，以实数 0.0 - 6553.5 s 指定到计时器打开为止所等待的时间。以 0.1 s 为单位来指定。

如果是间接指定计时器的话，那么指定寄存器。经过了保存在指定寄存器的时间值后，计时器打开。

信号种类									
	X	Y	M	K	T	C	D	V	常数
计时器	否	否	否	否	是	否	否	否	否
计时器值（直接）	否	否	否	否	否	否	否	否	是
计时器值（间接）	是	是	是	是	否	否	是	否	否

[注意]

软件运算中的采样时间为 8 ms，因此如果指定时间值不是 8 的倍数，那么计时器实际上比指定时间打开的早。例如，如果指定 100 ms 的话，那么计时器实际在 8 的倍数 96 ms 时打开时，将比设定时间早 4 ms。

示例

直接指定计时器

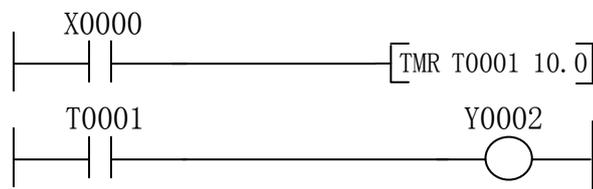
打开 X0000 10 秒后，打开 Y0002。

LSQ STR, &X0000

LSQ TMR, &T0001, 10.0

LSQ STR, &T0001

LSQ OUT, &Y0002



间接指定计时器

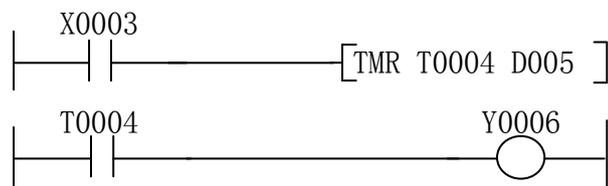
用保存在 D0005 的值来设定计时器值。X0003 打开后，经过 D0005 的时间值时，打开 Y0006。

LSQ STR, &X0003

LSQ TMR, &T0004, &D0005

LSQ STR, &T0004

LSQ OUT, &Y0006

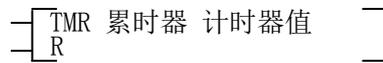


程序命令

LSQ TMRS, 累时器, 计时器值

符号

LSQ TMRS



步骤数

2

执行时间

10 μ s

功能

本命令之前的运算结果为 ON(=1)时，计时器运行。当累计时间达到指定值时，打开指定的设备(&T0000 - &T00FF)。

直到 R (reset) 输入为 ON 后，才复位计数器值（当前值）。

本命令所使用的计时器有 2 种类型：

直接指定 100 ms 累时器，以 100 ms 为单位增加。

计时器值以 100 ms 为单位来直接指定。

间接指定 100 ms 累时器，以 100 ms 为单位增加。

计时器值用保存在指定寄存器中的值来指定。该值以 100 ms 为单位。

参数

指定累时器

指定累计计时器。指定可能范围：T0000 - T00FF。

计时器值

如果是直接指定计时器的话，以实数 0.0 - 6553.5 s 指定到计时器打开为止所等待的时间。可以以 0.1 s 为增量来指定。

如果是间接指定计时器的话，那么指定寄存器。当保存在指定寄存器的时间值经过后，计时器打开。

信号种类									
	X	Y	M	K	T	C	D	V	常数
计时器	否	否	否	否	是	否	否	否	否
计时器值（直接）	否	否	否	否	否	否	否	否	是
计时器值（间接）	是	是	是	是	否	否	是	否	否

[注意]

软件运算中的采样时间为 8 ms，因此如果指定时间值不是 8 的倍数，那么计时器实际上比指定时间打开的早。例如，如果指定 100 ms 的话，那么计时器实际在 8 的倍数 96 ms 时打开时，将比设定时间早 4 ms。

示例

直接指定计时器

当 X0000 为 ON，累计时间达到 12.5 秒时，打开 Y0003，如果 X0001 为 ON 的话，关闭 Y0003。

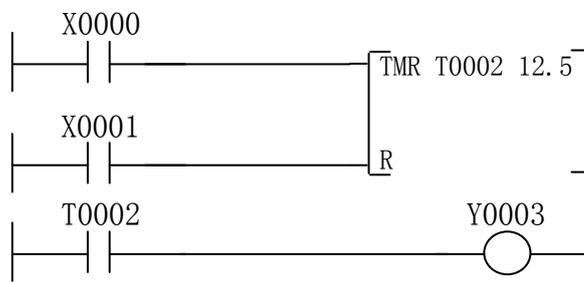
```
LSQ STR, &X0000
```

```
LSQ STR, &X0001
```

```
LSQ TMRS, &T0002, 12.5
```

```
LSQ STR, &T0002
```

```
LSQ OUT, &Y0003
```



间接指定计时器

当 X0004 为 ON，累计时间达到保存寄存器 D0007 的设定值时，打开 Y0008，如果 X0005 为 ON 的话，关闭 Y0008。

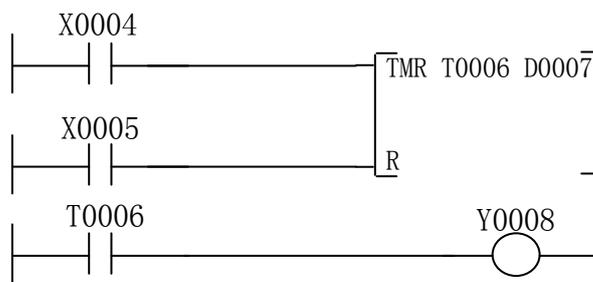
```
LSQ STR, &X0004
```

```
LSQ STR, &X0005
```

```
LSQ TMRS, &T0006, &D0007
```

```
LSQ STR, &T0006
```

```
LSQ OUT, &Y0008
```

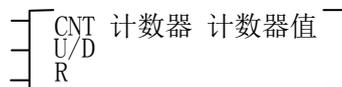


程序命令

LSQ CNTH, 计数器, 计数器值

符号

LSQ CNTH



步骤数

2

执行时间

10 μ s

功能

当 R（复位）输入为 OFF 并且 U/ D（增/减）输入为 ON 时，每次 CNT 计数器输入打开时，当前计数器值将减少-1。当 R（复位）输入和 U/ D 输入为 OFF 时，每次 CNT 输入打开时，当前计数器值将增加+1，然后当计数器值超过设定值时，打开指定计数器。当 R 输入为 ON 时，计数器值被复位到 0，并且指定计数器关闭。

本命令所使用的计数器有 2 种类型：

1. 直接指定增减计数器

通过输入常数来直接指定计数器值。

2. 间接指定增减计数器

用保存在指定寄存器中的值来设定计数器值。（设定范围：0 - 65535）。

参数

计数器

指定增减计数器。

计数器值

直接指定计数器时，指定整数值。设定范围：0 - 65535。

间接指定计数器时，指定寄存器。

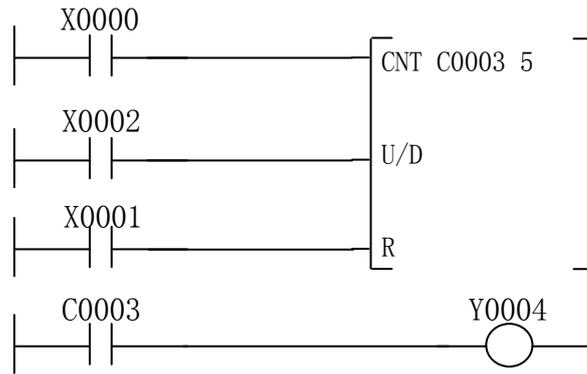
	信号种类								
	X	Y	M	K	T	C	D	V	常数
计数器	否	否	否	否	否	是	否	否	否
计数器值（直接）	否	否	否	否	否	否	否	否	是
计数器值（间接）	是	是	是	是	否	否	是	否	否

示例

直接计数器

比较: A = X0002 为 OFF 的状态下, X0000 打开的次数, B = X0002 为 ON 的状态下, X0000 打开的次数。如果 A-B 大于或等于 5 的话, 那么 Y0004 打开。如果 X0001 为 ON, 那么 Y0004 关闭。

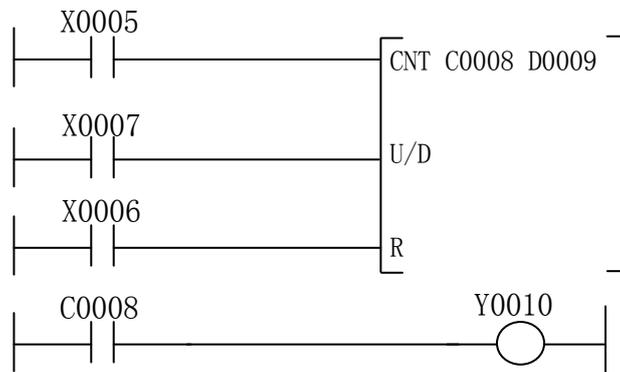
```
LSQ STR, &X0000
LSQ STR, &X0002
LSQ STR, &X0001
LSQ CNTH, &C0003, 5
LSQ STR, &C0003
LSQ OUT, &Y0004
```



间接计数器

比较: A = X0007 为 OFF 的状态下, X0005 打开的次数, B = X0007 为 ON 的状态下, X0005 打开的次数。如果 A-B 大于或等于保存在 D0009 中的设定值的话, 那么 Y0010 打开, 如果 X0006 为 ON, 那么 Y0010 关闭。

```
LSQ STR, &X0005
LSQ STR, &X0007
LSQ STR, &X0006
LSQ CNTH, &C0008, &D0009
LSQ STR, &C0008
LSQ OUT, &Y0010
```



程序命令

LSQ PTS**LSQ NTS**

符号

LSQ PTS 

LSQ NTS 

步骤数 1

执行时间 10 μs

功能

当 LSQ PTS 命令的上一演算结果由 OFF 转为 ON 时，本命令将 1 个扫描期间的演算结果打开为 ON(=1)。

当 LSQ NTS 命令的上一演算结果由 ON 转为 OFF 时，本命令将 1 个扫描期间的演算结果打开为 ON(=1)。

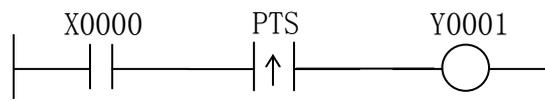
在一个程序中，示教 LSQ PTS 命令的总数加上示教 LSQ NTS 命令的总数和不能超过 255。

示例

LSQ STR, &X0000

LSQ PTS

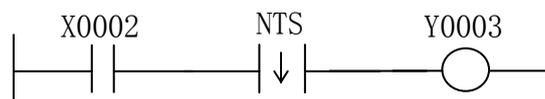
LSQ OUT, &Y0001



LSQ STR, &X0002

LSQ NTS

LSQ OUT, &Y0003



程序命令

LSQ MOVE, 源寄存器, 目的寄存器**LSQ WMOVE, 源寄存器, 目的寄存器**

符号

LSQ MOVE — MOVE 源寄存器, 目的寄存器 LSQ WMOVE — WMOVE 源寄存器, 目的寄存器

步骤数 2

执行时间 10 μs

功能

将源寄存器的数据传送到目的寄存器，保存并保持源寄存器中的数据不变。

LSQ MOVE 以字节为单位传送数据。

LSQ WMOVE 以字为单位传送数据。

参数

源寄存器

指定传送数据的源寄存器的地址，加“&”作为信号编号的最初字符。

目的寄存器

指定保存数据的寄存器的地址，加“&”作为信号编号的最初字符。

		信号种类								
		X	Y	M	K	T	C	D	V	常数
源寄存器		是	是	是	是	是	是	是	否	否
目的寄存器		否	是	是	是	是	是	是	否	否

示例

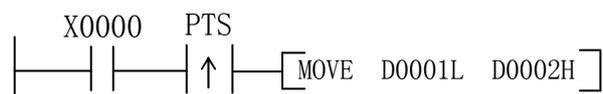
1 字节传送 (LSQ MOVE):

当 X0000 由 OFF 转变为 ON 时, 将数据寄存器 D0001 的低位 8 位传送到数据寄存器 D0002 的高位 8 位。

LSQ STR, &X0000

LSQ PTS

LSQ MOVE, &D0001L, D0002H

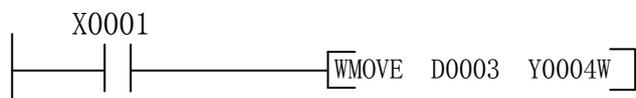


字传送 (LSQ WMOVE):

当 X0001 打开时, 将寄存器 D0003 中的数据传送到 Y0040 - Y004F。

LSQ STR, &X0001

LSQ WMOVE, &D0003, &Y0004W



程序命令

LSQ MOV, 源寄存器, 目的寄存器**LSQ WMOV, 源寄存器, 目的寄存器**

符号

LSQ MOV $\boxed{\text{MOV 源寄存器, 目的寄存器}}$ LSQ WMOV $\boxed{\text{WMOV 源寄存器, 目的寄存器}}$

步骤数 2

执行时间 10 μ s

功能

将源寄存器的十六进制常数传送到目的寄存器。

LSQ MOV 传送 2 位十六进制的字节数据。

LSQ WMOV 传送 4 位十六进制的字数据。

参数

源寄存器

指定要传送的十六进制数据的寄存器。LSQ MOV 的设定范围为 0 - FF。LSQ WMOV 的设定范围为 0 - FFFF。传送数据必须以十六进制常数来表示。

目的寄存器

指定要保存传送数据的寄存器的地址，加“&”作为信号编号的最初字符。

信号种类									
	X	Y	M	K	T	C	D	V	常数
源寄存器	否	否	否	否	否	否	否	否	是
目的寄存器	否	是	是	是	是	是	是	否	否

示例

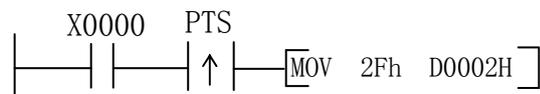
传送 2 位十六进制字节数据 (LSQ MOV):

当 X0000 由 OFF 转变为 ON 时, 将十六进制数据“2F”传送到数据寄存器 D0002 的高位 8 位。

LSQ STR, &X0000

LSQ PTS

LSQ MOV, ^H2F, &D0002H

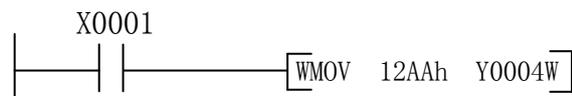


传送 4 位十六进制字数据 (LSQ WMOV) :

当 X0001 打开时，将十六进制数据“12AA”传送到寄存器 Y0040 - Y004F 中。

LSQ STR, &X0001

LSQ WMOV, ^H12AA, &Y0004W



程序命令

LSQ MOVR, 源寄存器, 目的寄存器

LSQ WMOVR, 源寄存器, 目的寄存器

符号

LSQ MOVR — MOVR 源寄存器, 目的寄存器

LSQ WMOVR — WMOVR 源寄存器, 目的寄存器

步骤数 2
执行时间 10 μs

功能

将源寄存器中的十进制常数转换为二进制数据, 并传送到目的寄存器。

LSQ MOVR 传送 3 位十进制的字节数据。

LSQ WMOVR 传送 5 位十进制的字数据。

参数

源寄存器

指定要传送的十进制数据的寄存器。LSQ MOVR 的设定范围为 0 - 255。LSQ WMOVR 的设定范围为 0 - 65535。

目的寄存器

指定要保存传送数据的寄存器的地址, 加 “&” 作为信号编号的最初字符。

	信号种类								
	X	Y	M	K	T	C	D	V	常数
源寄存器	否	否	否	否	否	否	否	否	是
目的寄存器	否	是	是	是	是	是	是	否	否

示例

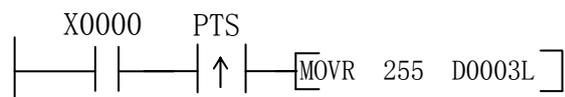
传送 3 位十进制字节数据 (LSQ MOVR):

当 X0000 由 OFF 转变为 ON 时, 将十进制数据 “255” (十六进制的 “FF”) 传送到数据寄存器 D0003 的低位 8 位。

LSQ STR, &X0000

LSQ PTS

LSQ MOVR, 255, &D0003L

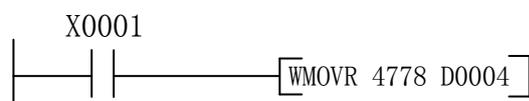


传送 5 位十进制字数据 (LSQ WMOVR) :

如果 X0001 为 ON 的状态, 那么将十进制数据 “4778” (十六进制的 “12AA”) 传送到数据寄存器 D0004 中。

LSQ STR, &X0001

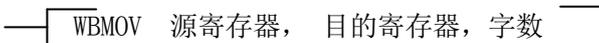
LSQ WMOVR, 4778, &D0004



程序命令

LSQ WBMOV, 源寄存器, 目的寄存器, 字数

符号

LSQ WBMOV 

步骤数 3

执行时间 200 μs

功能

将源寄存器的指定字数数据以相应字数传送到目的寄存器的内存中。最大传送数据数为 64 字。

参数

源寄存器

指定信号编号，其表示要传送数据的第一个数据的地址。加“&”作为信号编号的最初字符。

目的寄存器

指定信号编号，其表示要传送数据的目的寄存器中的第一个数据的地址。加“&”作为信号编号的最初字符。

字数

指定要传送数据的字数，设定范围为 1 - 64 的整数。

信号种类									
	X	Y	M	K	T	C	D	V	常数
源寄存器	是	是	是	是	是	是	是	否	否
目的寄存器	否	是	是	是	是	是	是	否	否
字数	否	否	否	否	否	否	否	否	是

示例

间接块传送字数据 (LSQ WBMOV)：

如果 X0001 为 ON 的状态，那么将 D0000 的 12 字数的数据传送到 D0020 的 12 字数的区域中。

LSQ STR, &X0001

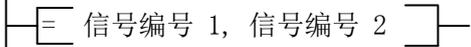
LSQ WBMOV, &D0000, &D0020, 12



程序命令

LSQ CMPEQ, 信号编号 1, 信号编号 2

符号

LSQ CMPEQ 

步骤数 2

执行时间 10 μs

功能

比较 2 个 16 位数据块，如果他们相等，那么打开演算结果为 ON(=1)。

参数

信号编号 1, 2

指定要比较保存数据的设备的信号编号。加“&”作为信号编号的最初字符。

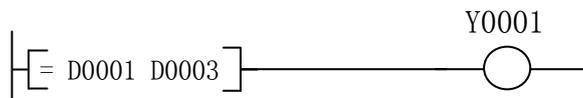
信号种类									
	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	否	是	是	是	是	是	是	否	否

示例

当 D0001 和 D0003 中的数据值相等时，打开 Y0001。

LSQ CMPEQ, &D0001, &D0003

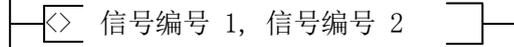
LSQ OUT, &Y0001



程序命令

LSQ CMPNEQ, 信号编号 1, 信号编号 2

符号

LSQ CMPNEQ 

步骤数 2

执行时间 10 μs

功能

比较 2 个 16 位数据块，如果他们不相等，那么打开演算结果为 ON(=1)。

参数

信号编号 1, 2

指定要比较保存数据的设备的信号编号。加“&”作为信号编号的最初字符。

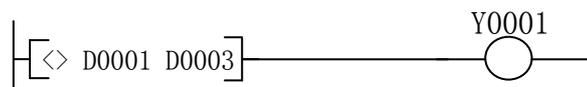
信号种类									
	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	否	是	是	是	是	是	是	否	否

示例

当 D0001 和 D0003 中的数据值不相等时，打开 Y0001。

LSQ CMPNEQ, &D0001, &D0003

LSQ OUT, &Y0001

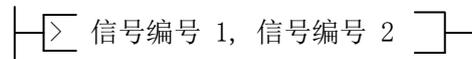


程序命令

LSQ CMPGT, 信号编号 1, 信号编号 2

符号

LSQ CMPGT



步骤数 2

执行时间 10 μs

功能

比较 2 个 16 位数据块，如果信号编号 1 的值比信号编号 2 的值大的话，那么打开演算结果为 ON (=1)。

参数

信号编号 1, 2

指定要比较保存数据的设备的信号编号。加“&”作为信号编号的最初字符。

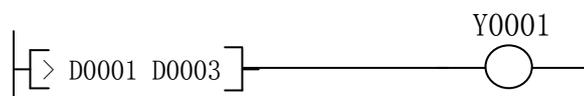
信号种类									
	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	否	是	是	是	是	是	是	否	否

示例

当 D0001 的数据值比 D0003 的数据值大时，打开 Y0001。

LSQ CMPGT, &D0001, &D0003

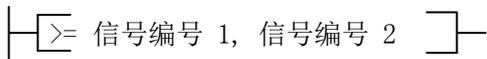
LSQ OUT, &Y0001



程序命令

LSQ CMPGE, 信号编号 1, 信号编号 2

符号

LSQ CMPGE 

步骤数 2

执行时间 10 μs

功能

比较 2 个 16 位数据块，如果由信号编号 1 设定的寄存器的数据值大于或等于由信号编号 2 设定的寄存器的数据值的话，那么打开演算结果为 ON (=1)。

参数

信号编号 1, 2

指定要读取并比较的保存数据的设备的信号编号。加“&”作为信号编号的最初字符。

		信号种类								
		X	Y	M	K	T	C	D	V	常数
信号编号 1		是	是	是	是	是	是	是	否	否
信号编号 2		否	是	是	是	是	是	是	否	否

示例

当 D0001 的数据值大于或等于 D0003 的数据值时，打开 Y0001。

LSQ CMPGE, &D0001, &D0003

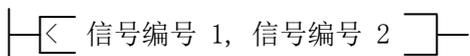
LSQ OUT, &Y0001



程序命令

LSQ CMPST, 信号编号 1, 信号编号 2

符号

LSQ CMPST 

步骤数 2

执行时间 10 μs

功能

比较 2 个 16 位数据块，如果由信号编号 1 设定的寄存器的数据值小于由信号编号 2 设定的寄存器的数据值的话，那么打开演算结果为 ON(=1)。

参数

信号编号 1, 2

指定要读取并比较的保存数据的设备的信号编号。加“&”作为信号编号的最初字符。

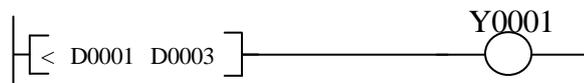
		信号种类								
		X	Y	M	K	T	C	D	V	常数
信号编号 1		是	是	是	是	是	是	是	否	否
信号编号 2		否	是	是	是	是	是	是	否	否

示例

当 D0001 的数据值比 D0003 的数据值小时，打开 Y0001。

LSQ CMPST, &D0001, &D0003

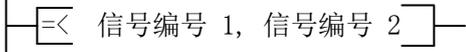
LSQ OUT, &Y0001



程序命令

LSQ CMPSE, 信号编号 1, 信号编号 2

符号

LSQ CMPSE 

步骤数 2

执行时间 10 μs

功能

比较 2 个 16 位数据块，如果由信号编号 1 设定的寄存器的数据值小于或等于由信号编号 2 设定的寄存器的数据值的话，那么打开演算结果为 ON (=1)。

参数

信号编号 1, 2

指定要读取并比较的保存数据的设备的信号编号。加“&”作为信号编号的最初字符。

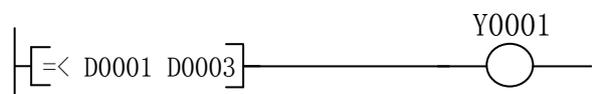
信号种类									
	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	否	是	是	是	是	是	是	否	否

示例

当 D0001 的数据值小于或等于 D0003 的数据值时，打开 Y0001。

LSQ CMPSE, &D0001, &D0003

LSQ OUT, &Y0001



程序命令

LSQ WBCD, 源寄存器, 目的寄存器

符号

LSQ WBCD — WBCD 源寄存器, 目的寄存器

步骤数 2
执行时间 10 μs

功能

将源寄存器的 16 位二进制数据转换为 BCD 码数据，并在目的寄存器中保存。

参数

源寄存器

指定保存二进制数据转换为 BCD 数据的寄存器的信号编号。加 “&” 作为信号编号的最初字符。

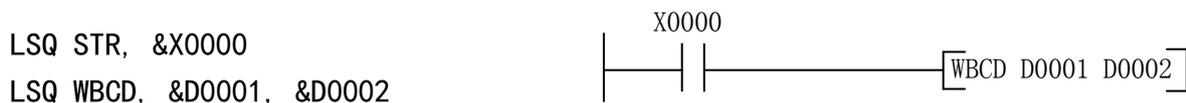
目的寄存器

指定要保存转换后数据的寄存器的信号编号。加 “&” 作为信号编号的最初字符。

信号种类									
	X	Y	M	K	T	C	D	V	常数
源寄存器	是	是	是	是	是	是	是	否	否
目的寄存器	否	是	是	是	是	是	是	否	否

示例

当 X0000 为 ON 的状态时，本程序将保存在 D0001 中的 16 位二进制数据转换为 4 位 BCD 码数据，然后在 D0002 中保存。



如果 D0001 的值为 “1200”，那么在 D0002 中保存 “4608”。

程序命令

LSQ WBIN, 源寄存器, 目的寄存器

符号

LSQ WBIN — WBIN 源寄存器, 目的寄存器

步骤数 2
执行时间 10 μs

功能

将保存在源寄存器的 BCD 码数据转换为 16 位二进制数据，并在目的寄存器中保存。

参数

源寄存器

指定保存 BCD 数据转换为 16 位二进制数据的寄存器的信号编号。加“&”作为信号编号的最初字符。

目的寄存器

指定要保存转换后数据的寄存器的信号编号。加“&”作为信号编号的最初字符。

信号种类									
	X	Y	M	K	T	C	D	V	常数
源寄存器	是	是	是	是	是	是	是	否	否
目的寄存器	否	是	是	是	是	是	是	否	否

示例

当 X0000 为 ON 的状态时，本程序将保存在 D0001 中的 4 位 BCD 码数据转换为 16 位二进制数据，然后在 D0002 中保存。



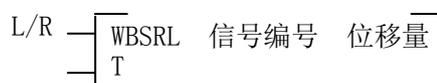
如果 D0001 的值为“1200”，那么在 D0002 中保存“04B0”。

程序命令

LSQ WBSRL, 信号编号, 位移量

符号

LSQ WBSRL



步骤数 2

执行时间 10 μs

功能

当 L/R 输入为 ON，T 由 OFF 切换为 ON 时，本命令将指定的寄存器内容以在位移量参数中设定的“n”位数向最高位向左位移。此时，从低位（最右边的）开始将位数设为 0。

当 L/R 输入为 OFF，T 由 OFF 切换为 ON 时，本命令将指定的寄存器内容以在位移量参数中设定的“n”位数向最低位向右位移。此时，从高位（最左边的）开始将位数设为 0。

参数

信号编号

指定包括要位移数据在内的寄存器的地址。加“&”作为信号编号的最初字符。

位移量

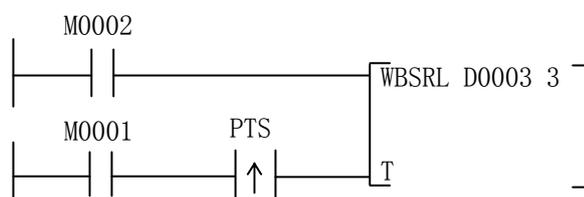
指定要位移的位数。设定范围：0 - 16。

信号种类									
	X	Y	M	K	T	C	D	V	常数
信号编号	否	是	是	是	是	是	是	否	否
位移量	否	否	否	否	否	否	否	否	是

示例

假设 M0002 保存 L/R 输入，并且 M0002 为 OFF，当 M0001 由 OFF 切换为 ON 时，D0003 向右位移 3 位。如果在执行本程序前 D0003 的值为“2000H”的话，那么执行后该值变为“0400H”。

```
LSQ STR, &M0002
LSQ STR, &M0001
LSQ PTS
LSQ WBSRL, &D0003, 3
```



程序命令

LSQ WUP, 信号编号, 位移量

符号

LSQ WUP $\left[\text{WUP} \text{ 信号编号, 偏移量} \right]$

步骤数 2

执行时间 200 μ s

功能

将字数据向高位位移。位移量由参数设定，并从指定寄存器开始向高位（信号编号大的方向）位移。最低位数据为 0。

参数

信号编号

指定包括要位移数据在内的寄存器地址。加“&”作为信号编号的最初字符。

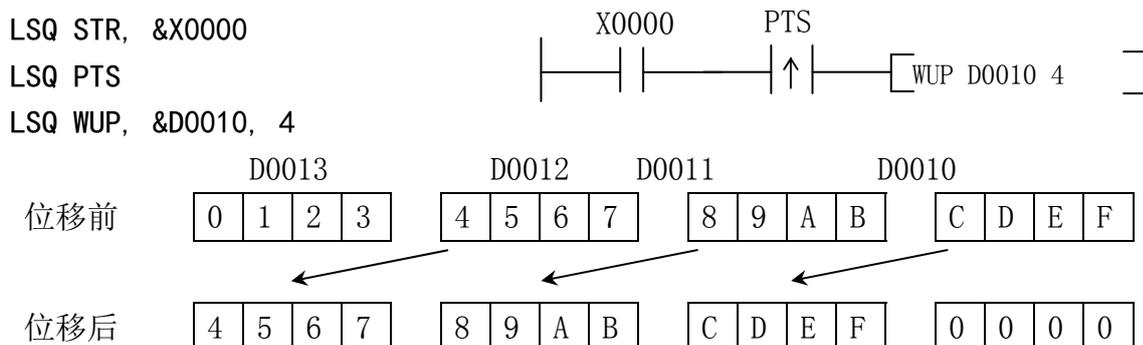
位移量

指定要位移的位数。设定范围：0 - 64。

信号种类									
	X	Y	M	K	T	C	D	V	常数
信号编号	否	是	是	是	是	是	是	否	否
位移量	否	否	否	否	否	否	否	否	是

示例

当打开 X0000 时，如下所示本程序从 D0010 开始以 4 个字向高位(向左)位移寄存器 D0010。



程序命令

LSQ WBRST, 位, 信号编号

符号

LSQ WBRST $\left[\text{WBRST 位, 信号编号} \right]$

步骤数 2
执行时间 10 μ s

功能

在以“信号编号”设定的寄存器中，将以参数“位”设定的数据复位到 0。

参数“位”的低位 4 位（0-15）为有效，其余位为无效。

参数

位

指定要复位数据的位置。（0, 1, …..15）

信号编号

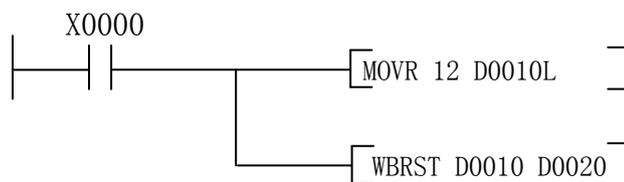
指定要复位的保存位的寄存器的信号编号。加“&”作为信号编号的最初字符。

信号种类									
	X	Y	M	K	T	C	D	V	常数
位	否	是	是	是	是	是	是	否	否
信号编号	是	是	是	是	是	是	是	否	否

示例

当 X0000 为 ON 的状态时，在 D0020 中复位第 12 位。

LSQ STR, &X0000
LSQ MOVR, 12, &D0010L
LSQ WBRST, &D0010, &D0020



执行本命令之前，如果 D0020 的值为“1234H”，那么当执行本命令时，复位第 12 位，并且 D0020 的值变为“234H”。

程序命令

LSQ USC, 计时器编号, ON 扫描数, OFF 扫描数

符号

LSQ USC — [USC 计时器编号 ON 扫描数 OFF 扫描数]

步骤数 1
执行时间 10 μs

功能

为特殊继电器的 USC (用户设定块) 1 和 2 (V0002 和 V0003) 设定条件。以指定时间 (“ON 扫描数”) 打开指定计时器, 并以指定时间 (“OFF 扫描数”) 关闭指定计时器。本命令执行后的下一扫描起打开/关闭计时器。

当将 0 指定给 ON 扫描数时, 执行本命令时, 计时器关闭。

如果以 0 以外的数设定 ON 扫描数, 0 为 OFF 扫描数的话, 计时器保持为 ON。

计时器从 OFF 开始。1 次扫描时间为 8 ms。

参数

计时器编号

指定要激活的计时器编号。指定为 1 或 2。

ON 扫描数

用整数来指定 ON 扫描数, 设定范围: 0 - 255。

OFF 扫描数

用整数来指定 OFF 扫描数, 设定范围: 0 - 255。

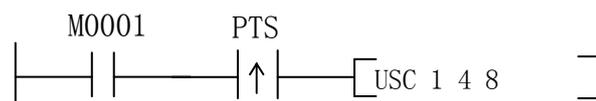
示例

当 M0001 从 OFF 切换为 ON 时, USC 1 打开 4 次扫描, 并关闭 8 次扫描。

LSQ STR, &M0001

LSQ PTS

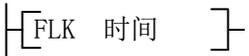
LSQ USC, 1, 4, 8



程序命令

LSQ FLK, 时间

符号

LSQ FLK 

步骤数 1

执行时间 10 μs

功能

在指定时间间隔打开/关闭(ON/OFF)。

在 1 个程序中，本命令最多可以使用 15 次。

参数

时间

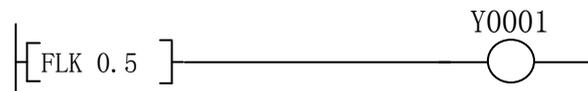
以 0.1 s 为单位指定 ON 和 OFF 之间的间隔。指定范围：0.0 - 6553.5 s。

示例

每 0.5 秒打开/关闭(ON/OFF) Y0001。

LSQ FLK, 0.5

LSQ OUT, &Y0001



程序命令

LSQ +, 信号编号 1, 信号编号 2, 信号编号 3

LSQ W+, 信号编号 1, 信号编号 2, 信号编号 3

符号

 LSQ + $\text{---} \left[+ \text{ 信号编号 1, 信号编号 2, 信号编号 3} \right]$

 LSQ W+ $\text{---} \left[W+ \text{ 信号编号 1, 信号编号 2, 信号编号 3} \right]$

步骤数 3

执行时间 10 μ s

功能

将信号编号 1 和信号编号 2 的寄存器的内容加起来，并把该结果保存到信号编号 3 的寄存器中。全部数据作为二进制数据处理。如果该结果发生进位，那么寄存器 V0005（进位标志）打开。如果该结果为 0，那么寄存器 V0007（零标志）打开。

LSQ +命令以 2 位十六进制的字节为单位进行加法运算。

LSQ W+命令以 4 位十六进制的字为单位进行加法运算。

参数

信号编号 1, 信号编号 2

指定保存加运算数据的寄存器的信号编号。加“&”作为信号编号的最初字符。

信号编号 3

指定保存结果的寄存器的信号编号。加“&”作为信号编号的最初字符。

	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	是	是	是	是	是	是	是	否	否
信号编号 3	否	是	是	是	是	是	是	否	否

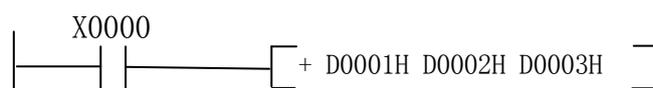
示例

字节加法运算 (LSQ+):

当 X0000 为 ON 的状态时，下面的程序是将 D0001H 和 D0002H 的寄存器的值加起来，并将该结果保存到 D0003H 的寄存器中。

LSQ STR, &X0000

LSQ +, &D0001H, &D0002H, &D0003H

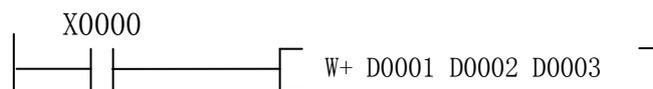


字加法运算 (LSQ W+):

当 X0000 为 ON 的状态时, 下面的程序是将 D0001 和 D0002 的寄存器的值加算起来, 并将该结果保存到 D0003 的寄存器中。

LSQ STR, &X0000

LSQ W+, &D0001, &D0002, &D0003



程序命令

LSQ -, 信号编号 1, 信号编号 2, 信号编号 3

LSQ W-, 信号编号 1, 信号编号 2, 信号编号 3

符号

LSQ - — [- 信号编号 1, 信号编号 2, 信号编号 3]

LSQ W- — [W- 信号编号 1, 信号编号 2, 信号编号 3]

步骤数 3

执行时间 10 μs

功能

将信号编号 1 和信号编号 2 的寄存器的内容相减，并把该结果保存到信号编号 3 的寄存器中。全部数据作为二进制数据处理。如果该结果发生借位，那么寄存器 V0006（借位标志）打开。如果该结果为 0，那么寄存器 V0007（零标志）打开。

LSQ - 命令以 2 位十六进制的字节为单位进行减法运算。

LSQ W- 命令以 4 位十六进制的字为单位进行减法运算。

参数

信号编号 1, 信号编号 2

指定保存减运算数据的寄存器的信号编号。加“&”作为信号编号的最初字符。

信号编号 3

指定保存结果的寄存器的信号编号。加“&”作为信号编号的最初字符。

	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	是	是	是	是	是	是	是	否	否
信号编号 3	否	是	是	是	是	是	是	否	否

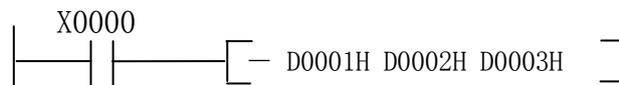
示例

字节减法运算 (LSQ -):

当 X0000 为 0N 的状态时，下面的程序是从 D0001H 的寄存器的值减去 D0002H 的寄存器的值，并将该结果保存到 D0003H 的寄存器中。

LSQ STR, &X0000

LSQ -, &D0001H, &D0002H, &D0003H

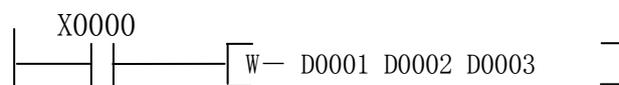


字减法运算 (LSQ W -):

当 X0000 为 ON 的状态时, 下面的程序是从 D0001 的寄存器的值减去 D0002 的寄存器的值, 并将该结果保存到 D0003 的寄存器中。

LSQ STR, &X0000

LSQ W-, &D0001, &D0002, &D0003



程序命令

LSQ X, 信号编号 1, 信号编号 2, 信号编号 3

LSQ WX, 信号编号 1, 信号编号 2, 信号编号 3

符号

LSQ X $\left[\text{X 信号编号 1, 信号编号 2, 信号编号 3} \right]$ LSQ WX $\left[\text{WX 信号编号 1, 信号编号 2, 信号编号 3} \right]$

步骤数 3

执行时间 10 μ s

功能

将信号编号 1 和信号编号 2 的寄存器的内容相乘，并把该结果保存到信号编号 3 的寄存器中。全部数据作为二进制数据处理。

LSQ X 命令以 2 位十六进制的字节为单位进行乘法运算。

LSQ WX 命令以 4 位十六进制的字为单位进行乘法运算。

参数

信号编号 1, 信号编号 2

指定保存乘法运算数据的寄存器的信号编号。加“&”作为信号编号的最初字符。

信号编号 3

指定保存结果的寄存器的信号编号。加“&”作为信号编号的最初字符。不能指定 8 位。

	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	是	是	是	是	是	是	是	否	否
信号编号 3	否	是	是	是	是	是	是	否	否

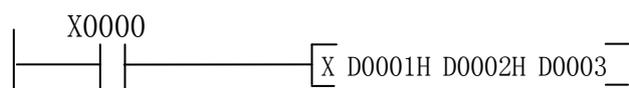
示例

字节乘法运算 (LSQ X):

当 X0000 为 0N 的状态时，下面的程序是将 D0001H 和 D0002H 的寄存器的值相乘，并将该结果保存到 D0003 的寄存器中。

LSQ STR, &X0000

LSQ X, &D0001H, &D0002H, &D0003

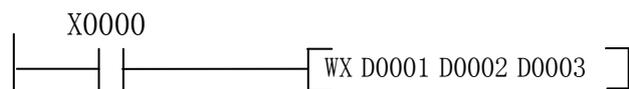


字乘法运算 (LSQ WX):

当 X0000 为 ON 的状态时, 下面的程序是将 D0001 和 D0002 的寄存器的值相乘, 并将该结果保存到 D0003 的寄存器中。

LSQ STR, &X0000

LSQ WX, &D0001, &D0002, &D0003



程序命令

LSQ /, 信号编号 1, 信号编号 2, 信号编号 3

LSQ W/, 信号编号 1, 信号编号 2, 信号编号 3

符号

LSQ / $\left[\text{/ 信号编号 1, 信号编号 2, 信号编号 3} \right]$ LSQ W/ $\left[\text{w/ 信号编号 1, 信号编号 2, 信号编号 3} \right]$

步骤数 3

执行时间 10 μ s

功能

信号编号 1 的寄存器的内容除以信号编号 2 的寄存器的内容，并把商保存到信号编号 3 的寄存器中，余数保存到信号编号 3 的下一寄存器中。如果由信号编号 3 指定的寄存器为最后一个寄存器的话，那么不保存余数。全部数据作为二进制数据处理。如果信号编号 2 的寄存器值为 0，那么寄存器 V0008（错误标志）打开。

LSQ / 命令以 2 位十六进制的字节为单位进行除法运算。

LSQ W/ 命令以 4 位十六进制的字为单位进行除法运算。

功能

信号编号 1, 信号编号 2

指定保存除法运算数据的寄存器的信号编号。加“&”作为信号编号的最初字符。

信号编号 3

指定保存结果的寄存器的信号编号。加“&”作为信号编号的最初字符。

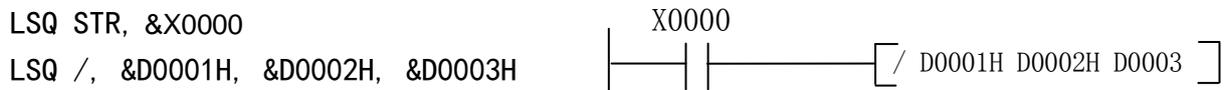
	X	Y	M	K	T	C	D	V	常数
信号编号 1	是	是	是	是	是	是	是	否	否
信号编号 2	是	是	是	是	是	是	是	否	否
信号编号 3	否	是	是	是	是	是	是	否	否

示例

字节除法运算 (LSQ /):

当 X0000 为 0N 的状态时，下面的程序是从 D0001H 的寄存器的值除以 D0002H 的寄存器的

值，并将该结果保存到 D0003H 的寄存器中。



字除法运算 (LSQ W/):

当 X0000 为 ON 的状态时，下面的程序是从 D0001 的寄存器的值除以 D0002 的寄存器的值，并将该结果保存到 D0003 的寄存器中。



有关 +, W+, -, W-, X, WX, /, W/ 命令

字节数据命令的参数配置:

当将数据寄存器 (D: 16 位数据) 传送给信号 1, 2, 3 时

→ D0002H: 表示寄存器 D0002 的高位 8 位数据。

D0002L: 表示寄存器 D0002 的低位 8 位数据。

当将 X, Y, M, K, T, C, D (1 位数据) 传送给信号 1, 2, 3 时

→ Y0000L: 表示 &Y0000 - &Y0007.

Y0000H: 表示 &Y0008 - &Y000F.

字数据命令的参数配置:

当将数据寄存器 (D: 16 位数据) 传送给信号 1, 2, 3 时

→ D0002: 表示寄存器 D0002。

当将 X, Y, M, K, T, C, D (1 位数据) 传送给信号 1, 2, 3 时

→ Y0000W: 表示 &Y0000 - &Y000F.

Y0001W: 表示 &Y0010 - &Y001F.

程序命令

LSQ AFLGCLR

LSQ AFLGCLRN

符号

LSQ AFLGLR [AFLGCLR]

LSQ AFLGCLRN [AFLGCLRN]

步骤数 1

执行时间 2 μs

功能

LSQ AFLGCLR 命令使应用操作命令标志清除模式有效。

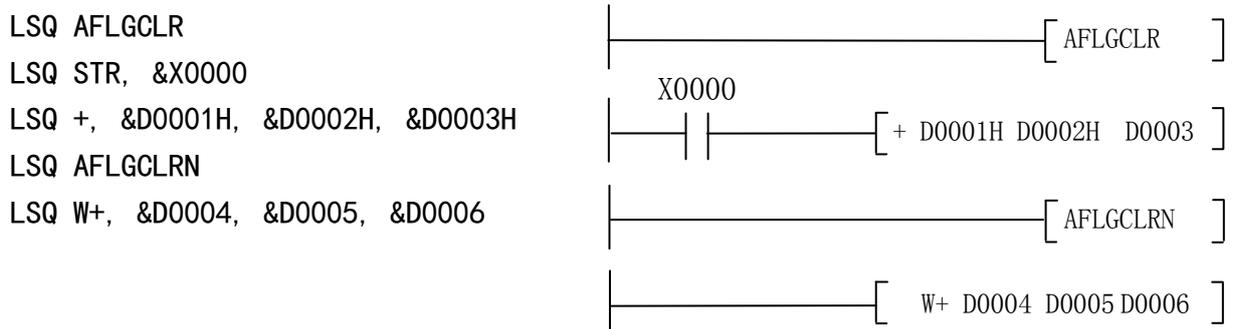
以下 4 种标志用于清除应用操作命令：进位标志、借位标志、零标志、错误标志。

LSQ AFLGCLRN 命令使应用操作命令标志清除模式无效。

用 AFLGCLR 命令清除的标志不能恢复。

示例

在下面的程序中，在应用操作命令标志清除模式有效的状态下，X0000 打开为 ON。将 D0001H 和 D0002H 的寄存器的值加起来，并把该结果保存到 D0003H 的寄存器中。然后，使应用操作命令标志清除模式无效。并且，将 D0004 和 D0005 的寄存器的值加起来，把该结果保存到 D0006 的寄存器中。



2.6.2 数值运算部的指令/命令

2.6.2.1 监控指令

以下指令控制数值运算程序（numpg）。

NUMEXECUTE	从第一步开始执行 numpg。
NUMABORT	执行过程中停止 numpg。
NUMCONTINUE	用 NUMABORT 指令停止后重新开始 numpg。
NUMKILL	显示 numpg 的当前执行状态。
NUMSTATUS	初始化包括 numpg 在内的程序栈。

监控指令

NUMEXECUTE

功能

执行数值运算部的程序 (numpg)。

示例

数值运算程序的程序名固定为“numpg”。如果找不到程序 (numpg) 或该程序不存在，那么错误产生：

(P1038) 程序不存在。

如果数值运算程序 numpg 已经开始执行，那么错误产生：

(P2055) [LSEQ] 程序已在运行中。

示例

>NUMEXECUTE

监控指令

NUMABORT

功能

停止当前执行中的数值运算程序 (numpg)。

示例

>NUMABORT

监控指令

NUMCONTINUE NEXT

功能

用 NUMABORT 指令停止后重新开始数值运算程序 (numpg) 的执行。

参数

如果输入了 NEXT，那么在中断步骤的下一步重新开始程序的执行。如果未输入 NEXT，那么在中断步骤重新开始程序的执行。如果用 WAIT 命令等暂停当前程序，一旦输入 NEXT 将解除等待状态。

示例

```
>NUMCONTINUE
```

监控指令

NUMKILL

功能

初始化包括数值运算程序 (numpg) 在内的程序栈。

说明

用 NUMABORT 指令，解除暂停数值运算程序的状态。但，执行本指令后，用 NUMCONTINUE 指令或其他指令不能重新开始程序。

监控指令

NUMSTATUS

功能

显示数值运算部的执行状态。

说明

1.	NUM 状态	: 程序 执行中
2.	程序执行时间	: 808 ms
	程序执行时间 (平均)	: 808 ms
3.	程序 NUMPG 有	
	程序名	步骤编号
	numpg	3 C=DREG (1)

1. NUM (数值运算程序执行) 状态

显示数值运算部当前的执行状态。根据状态, 将显示信息“程序执行中”或“程序未执行”。

2. 程序执行时间

显示正在执行程序所经过的执行时间和平均执行时间。

3. 程序

显示数值运算程序 (numpg) 是否存在。根据情况, 将显示信息“NUMPG 有”或“NUMPG 无”。

2.6.2.2 基本运算部和数值运算部间的接口用命令

数据寄存器用于基本运算部和数值运算部间的接口。以下指令用于从数值运算部向数据寄存器读取或写入数据。

DREG	向寄存器写入数据。
DREG (寄存器编号)	从寄存器读取数据。
DREGS	向寄存器写入全部数据。
DREGR	从寄存器读取全部数据。
DREGINIT	清除寄存器的全部内容。

监控指令 / 程序命令

DREG 数据寄存器编号 = 设定值

功能

向数据寄存器写入数据。

参数

数据寄存器编号

0 - 2047 (相当于&D0000 - &D07FF)

设定值

0 - 65535

说明

将参数“设定值”中所指定的值代入到指定的数据寄存器编号中。如果指定值超出设定范围，该值小于下限，那么将最小值代入数据寄存器；该值超过上限，那么将最大值代入数据寄存器。并且，小数点以下的舍去代入。

示例

DREG 1=a

如果 a=256.1 那么在&D0001 中代入 256。

实数函数

DREG (数据寄存器编号)

功能

从数据寄存器中读取数据。

参数

数据寄存器编号

0 - 2047 (相当于&D0000 - &D07FF)

说明

返回指定数据寄存器的内容。

示例

`c=DREG(b)`

如果 b=2, 那么将&D0002 的内容代入变量 c 中。

监控指令 / 程序命令

DREGS 数据寄存器编号 = 数组变量名[元素编号], 数据数

功能

向数据寄存器写入全部数据。

参数

数据寄存器编号

0 - 2047 (相当于&D0000 - &D07FF), 表示要写入的数据寄存器范围的第一数据寄存器。

数组变量名

指定包括要写入数据寄存器的实数数组的数组变量名。

元素编号

指定要写入数据寄存器的实数数组的第一个数组元素编号。

数据数

指定要写入数据寄存器的元素总数, 包括第一个元素。

说明

将在数组变量中设定的实数值, 以指定数据数代入到从指定的数据寄存器编号开始的数据寄存器中。如果数组变量的实数值超出设定范围, 该值小于下限, 那么将最小值代入数据寄存器; 该值超过上限, 那么将最大值代入数据寄存器。并且, 小数点以下的舍去代入。

示例

DREGS 5=a[2], 3

如果 a[2]=256.1, a[3]=123.4, a[4]=999.9,

那么在&D0005 中代入 256,

在&D0006 中代入 123,

在&D0007 中代入 999。

监控指令 / 程序命令

DREGR 数组变量名[元素编号] = 数据寄存器编号, 数据数

功能

从数据寄存器读取全部数据。

参数

数组变量名

指定包括要保存的从数据寄存器读取的实数数组的数组变量名。

元素编号

指定数组变量的第一个数组元素编号，该数组变量保存从数据寄存器读取的数据。

数据寄存器编号

0 - 2047（相当于&D0000 - &D07FF），表示要读取的数据寄存器范围的第一数据寄存器。

数据数

指定从数据寄存器读取的数据数。

说明

读取连续的数据寄存器的内容，并将其保存在指定的数组变量中。

示例

DREGR a[1]=b, c

如果 b=2, c=3

那么&D0002 的值为 a[1],

&D0003 的值为 a[2],

&D0004 的值为 a[3]。

监控指令

DREGINIT

功能

清除数据寄存器的全部内容。

说明

设定全部数据寄存器为 0 。

基本运算或数值运算程序当前正在执行中，执行 DREGINIT 指令，在此显示的错误产生并且不能执行本指令。

(P1042) NUM 程序运行中。

示例

```
>DREGINIT  
确认! 执行吗? (是: 1, 否: 0) 1  
>
```

3.0 KLadder

KLadder 是 KLogic 专用的梯形图编程应用软件，其在 MS Windows 中运行并操作。将用 KLadder 画的梯形图转换为 KLogic 的顺序程序。

3.1 安装和连接

3.1.1 KLadder 硬件配置要求

KLadder 的最低硬件配置要求如下：

	硬件配置要求
OS	Windows 95 以上
CPU	Pentium 以上
内存	16MB, 建议 32MB 以上

3.1.2 KLadder 安装程序

给您一张 KLadder 软盘，安装软件到您的计算机上。安装程序如下。

1. 向硬盘驱动器安装：

将软盘中的以下文件复制到您计算机的硬盘驱动器的任意目录中。

KCODECHG. DLL
KCONVERT. DLL
KLADDER. EXE
KLADDERENG. DLL
KLOGICCOMM. DLL
MSGEG. DLL
MSGJP. DLL

2. 创建快捷图标

为了方便使用，创建 KLADDER. EXE 的快捷图标。

从菜单栏中的[选项] - [配置]中更改显示的语言。在出现的对话框中点击[语言] 标签并选择想要的语言。在 KLadder 下一次启动时，更改后的语言有效。更多详情，请参阅 3.6.2.4 语言。

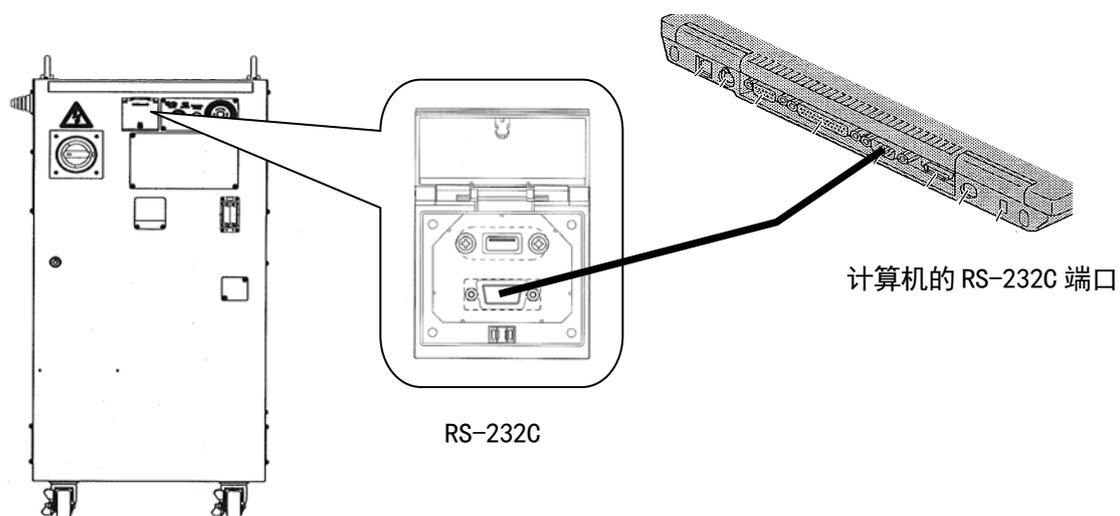
3.1.3 与控制器的连接方法

连接安装了 KLadder 的计算机和机器人控制器。有两种连接方法：

1. RS232C
2. 以太网 (TCP/IP)

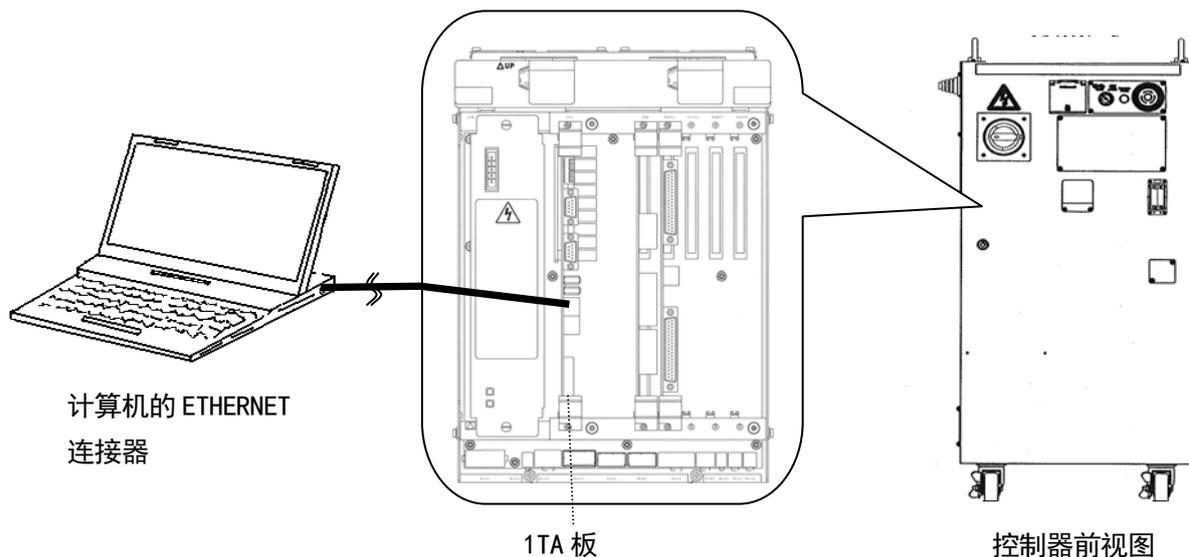
3.1.3.1 使用 RS232C 连接

连接计算机上的 RS232C 端口和机器人控制器上的终端端口。



3.1.3.2 使用以太网的连接

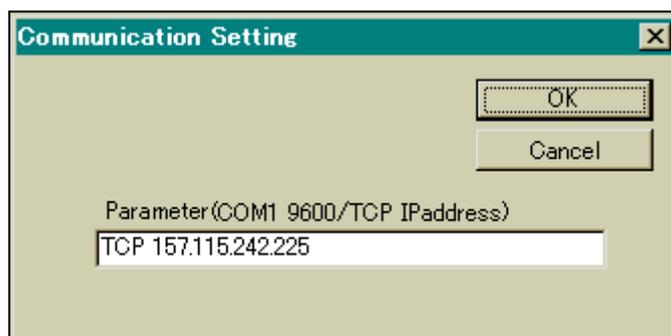
连接计算机的以太网端口和机器人控制器内部的以太网端口。在机器人控制器中需要 1TA 子板。



3.1.4 通信模块的设定

选择 KLadder 和 RC（机器人控制器）之间进行通信的模块。

1. 启动 KLadder，选择 [Setup（设定）] - [Setup Communication Module（通信模块的设定）]。



2. 设定参数。

(1) 串行连接 (RS232C)：

以 [Connecting port (连接端口)] [Baud rate (波特率)] 的顺序输入。

例 COM1 9600

(2) 以太网连接：

以 TCP [IP Address (IP 地址)] 的顺序输入。

例 TCP 192.168.0.2

输入后，点击<OK>。之后反映输入的设定。

3.2 用 KLadder 编程

本节介绍用 KLadder 编程的顺序和方法。

3.2.1 打开文件

启动 KLadder。

要创建一个新文件，选择[File (文件)] - [New file (新文件)]，或点击 。

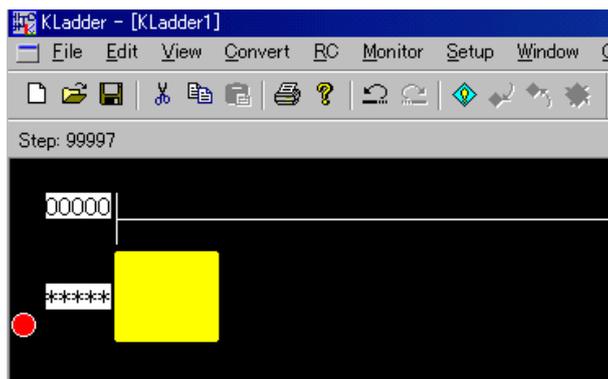
在新文件中自动显示 START。要取消 START 显示，从菜单栏的[Option (选项)] - [Configuration (配置)] 中选择 [Operation (操作)]，并取消已选择的[Create START automatically when new file is open (当新文件打开时，自动创建 START)]。

打开已存文件时，选择[File (文件)] - [Open (打开)]，或点击 。对话框[Open File (打开文件)]出现，因此双击需要的文件。

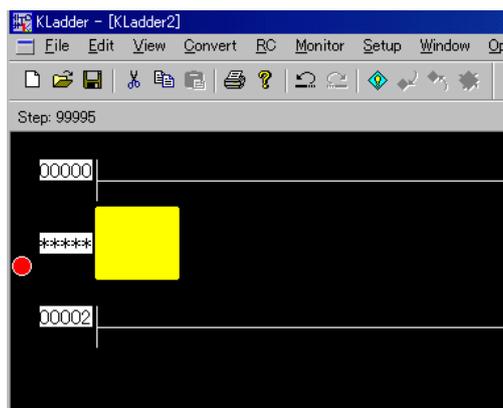
3.2.2 建立新回路（插入新块）

要建立新回路，先建一新块。

1. 点击要建新块的位置。光标（黄色箱）位于该位置。



2. 在键盘上按 **Ctrl** + **Insert**，或点击 。

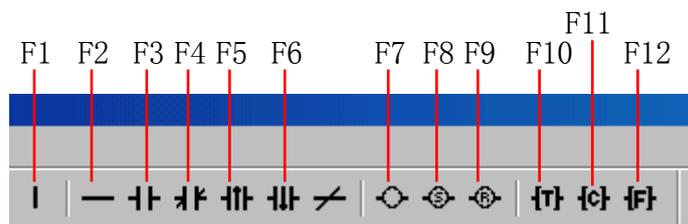


在已存块之间插入新块，可以在选择位置的前或后插入。设定是否在光标的前或后插入新块，选择[Option (选项)] - [Settings (设定)]。(请参阅 3.6.2.2 操作设定。)

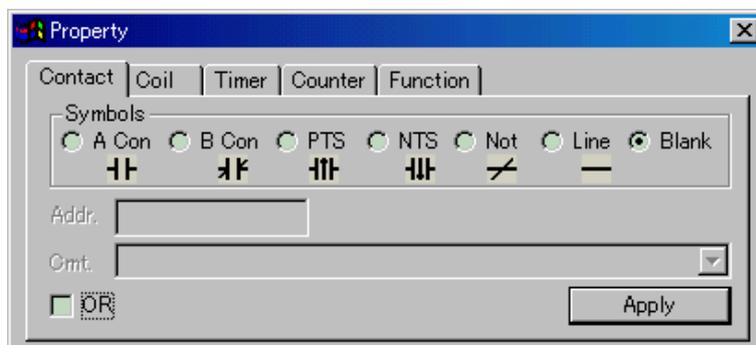
3.2.3 配置符号

按以下程序在梯形图中配置电子符号。

1. 点击要配置符号的位置。光标（黄色箱）移到该位置。
2. 在工具栏上点击想要的符号，或如下图在键盘上按相应的功能键。



3. 在图中配置选择的符号。
4. 移动光标到符号并按 **Enter**，或双击符号。如下所示[Property (属性)]对话框显示。
注： 如果在[Option (选项)]- [Configuration (配置)] - [Operation (操作)] 中选择了[Activate Property Dialog when symbol is input. (输入符号时，属性对话框有效。)]，那么在配置符号后，[Property (属性)]对话框立刻显示。



5. 选择要配置的符号，并在[Addr. (ess) (地址)]和[Cmt. (Comment) (说明)]栏中输入信号编号和说明。可使用的信号如下：

信号种类	格式及上下限	信号数
输入	X0000-X07FF	2048
输出	Y0000-Y07FF	2048
内部中间继电器	M0000-M07FF	2048
保持型继电器	K0000-K01FF	512
计时器	T0000-T00FF	256
计数器	C0000-C00FF	256
数据寄存器	D0000-D07FF	2048
特殊继电器	V0000-V00FF	256

无论 Caps Lock (大小写锁定) 如何, 以大写字母输入[Address (地址)]。并且如下可以省略输入:

例: 为 M0001 输入 M1
 为 M003A 输入 M3A。

触点侧或线圈侧都可以输入[Comment (说明)], 输入的说明显示在两侧。

3.2.3.1 梯形图构成

建议梯形图如下构成。



1. 输入部

将输入信号到 KLogic 的设备，如按钮、限位开关等写入到输入部。使用“X”作为信号识别符。设定可能范围：X0000–X07FF（2048 点）。

2. 逻辑部

将不需要驱动电磁阀、电磁开关等外部设备的暂时记忆等的内部中间继电器写入逻辑部。使用“M”作为信号识别符。设定可能范围：M0000–M07FF（2048 点）。

计时器

将计时器写入逻辑部。使用“T”作为信号识别符。使用以下种类的计时器。设定可能范围：T0000–T00FF（256 点）。

计时器 (LSQ TMR)	直接	100 ms 单位
	间接	100 ms 单位
累时器 (LSQ TMRS)	直接	100 ms 单位
	间接	100 ms 单位

计数器

使用以下类型的计数器。使用“C”作为信号识别符。设定可能范围：C0000–C00FF（256 点）。

增减计数器 (LSQ TMR)	直接
	间接

保持型继电器

此继电器用于条件设定等，即使电源切断后也必须保存在内存中。将保持型继电器写入逻辑部。使用“K”作为信号识别符。设定可能范围：K0000–K01FF（512 点）。

特殊继电器

特殊继电器为内部中间继电器，其主要用于表示 KLogic 的输出动作状态或应用命令的

运算结果。使用“V”作为信号识别符。设定可能范围：V0000-V00FF（256点）。

编号	内容	
V0000	通常 ON	无论动作状态如何通常为 ON
V0001	通常 OFF	无论动作状态如何通常为 OFF
V0002	用户定义计时器 1	输出 LSQ USC1 的结果
V0003	用户定义计时器 2	输出 LSQ USC2 的结果
V0004	机器人错误	输出机器人错误*
V0005	进位标志	LSQ+, LSQ W+的运算结果进位时打开
V0006	借位标志	Turns ON LSQ-, LSQ W-的运算结果借位时打开。
V0007	零标志	LSQ+, W+ ,-, W-, X, WX, /, W/的结果为 0 时, 打开。
V0008	错误标志	当 LSQ/, LSQ W/中被分割的寄存器为 0 时, 打开。

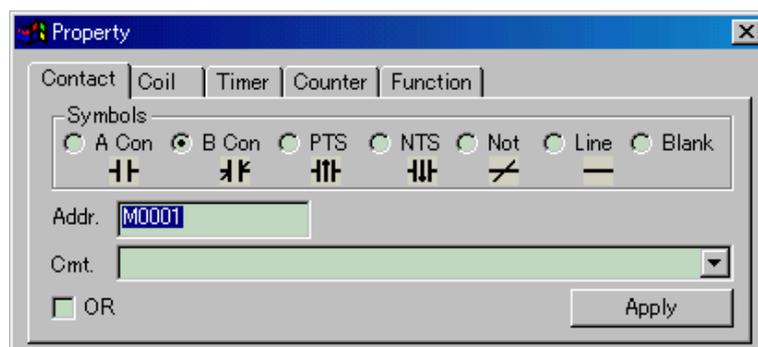
3. 输出部

将应答来自 KLogic 输出信号的设备写入输出部。使用“Y”作为信号识别符。设定可能范围：Y0000-Y07FF（2048点）。

3.2.4 更改符号

按以下程序更改梯形图中的电气符号：

1. 双击要更改的符号，或移动光标到符号并按 Enter（回车）。
2. 如下图所示 [Property（属性）] 对话框出现。



选择想要的符号，输入符号的地址或说明等。进行必要的更改后，按 Enter 或点击 < Apply（应用）>。

3.2.5 画并联回路

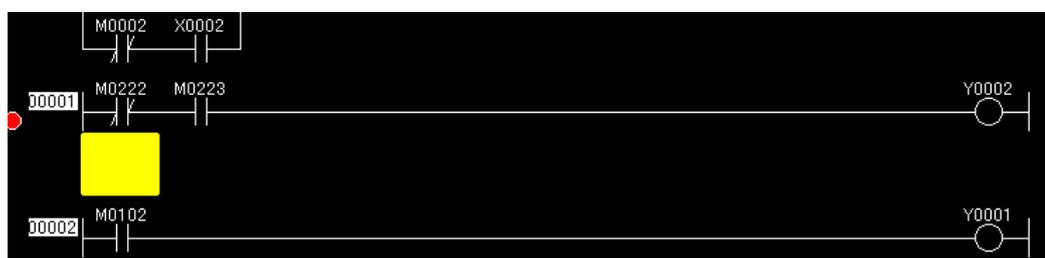
3.2.5.1 在同一块内插入第二行

为了画并联(OR)回路，在同一块内插入第二行。

1. 点击要插入第二行的位置，并在该位置放置光标（黄色箱）。



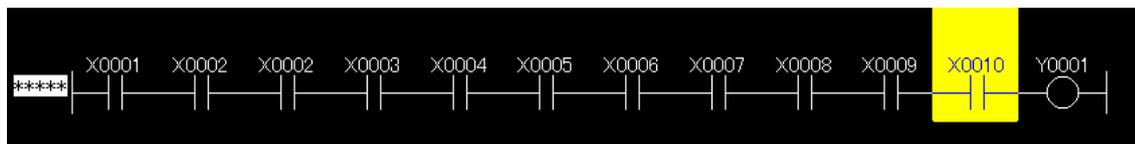
2. 按键盘上的 **Shift (转换)** + **Insert (插入)** 或点击 。或，选择 [Edit (编辑)] - [New line insert (新 1 行插入)]。在光标下插入新行，该新行后都各往下串一行。



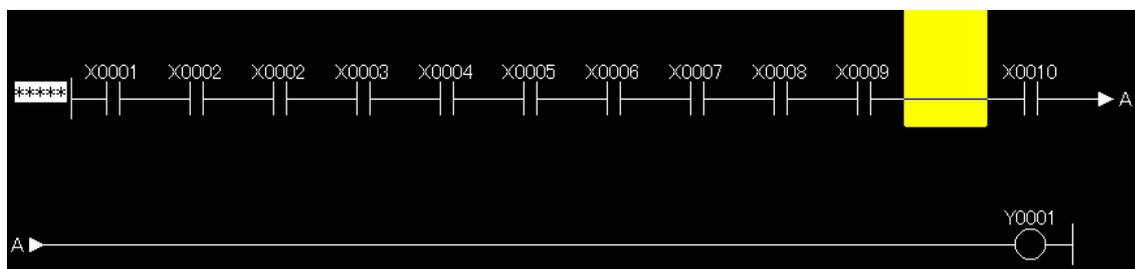
3.2.5.2 移动符号 1 列

按 **Insert (插入)** 键将图上的电气符号向右侧移动 1 列。

1. 点击要移动的符号，并在该符号上放置光标（黄色箱）。

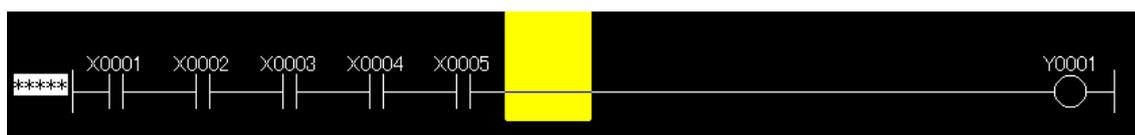


2. 按 **Insert (插入)**。光标上的符号和光标右侧的符号向右移动 1 列。当超过 1 行配置的最多符号数时，该行自动返回。



当 **Insert (插入)** 键无效时

如下图所示，线圈前的横线，**Insert (插入)** 键无效。在这种情况下，在线圈上放置光标，然后按 **Insert (插入)**。该行自动返回。



3.2.5.3 画 OR 图

为了画 OR 的竖线，点击工具栏上的  或选中 [Property (属性)] 对话框的选择框。

1. 移动光标（黄色箱）或点击想要的位置。



2. 点击 **I** 或按 **Enter** 或双击来显示 [Property (属性)] 对话框。选中选择框 [OR]。在光标前插入 OR 线。



3. 为了删除图中画的 OR 线，移动光标到该线并点击 **I**，或按 **Enter**，或双击 OR 线，[Property (属性)] 对话框出现，不选选择框 [OR]。

3.2.6 移动/复制数据

可以剪切、复制、粘贴回路数据。有两种剪切、复制、粘贴数据的方法。

仅使用鼠标时

右击行首并拖拽，选择目的回路。松开右按钮显示菜单。移动数据时，选择 [Cut (剪切)]，或复制数据时，选择 [Copy (复制)]。移动光标 (黄色箱) 到目标位置。右击并从下拉菜单中选择 [Paste (粘贴)]。

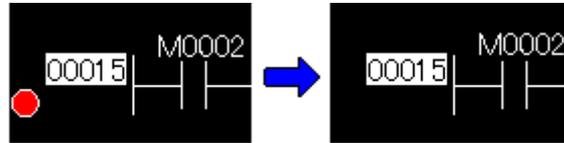
使用键盘上的快捷键

点击行首，并拖拽来选择想要的的数据。要剪切，按 **Ctrl**+**X**，要复制，按 **Ctrl**+**C**。要粘贴，移动光标 (黄色箱) 到目标位置，并按 **Ctrl**+**V**。

粘贴数据时，该数据被插入到已选择的行，并且没有数据被改写。

3.2.7 转换数据

在创建/编辑 1 个回路后，梯形图数据转换为助记符数据。未转换的块在行首显示红点。转换成功后，红色标记消失。（参阅下图。）



要转换数据，在菜单栏的[Convert（转换）]中选择[One block（1 块）]或[Edited blocks（编辑块）]。

[One block（1 块）]

仅转换放置光标的块。当错误产生时，错误信息出现在画面上。



[Edited blocks（编辑块）]

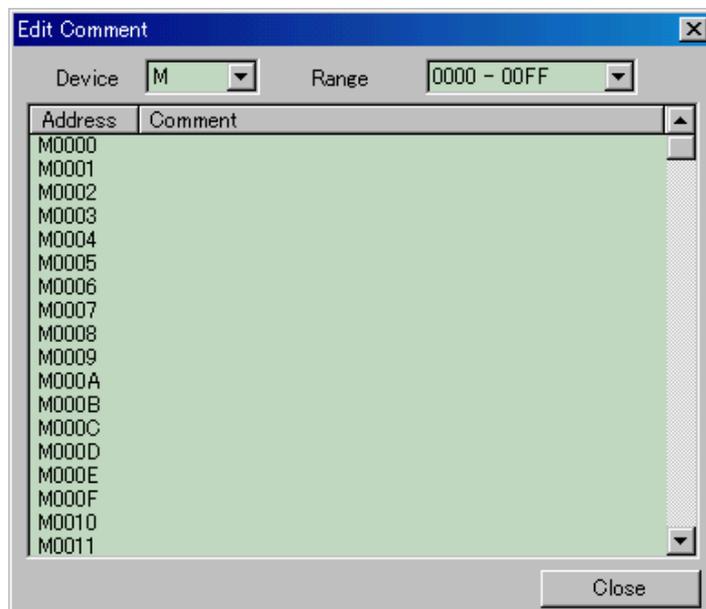
转换所有编辑的还未转换的块。如果错误产生的话，在错误出现处信息显示，并且转换停止。在这种情况下，错误产生后的块不转换。



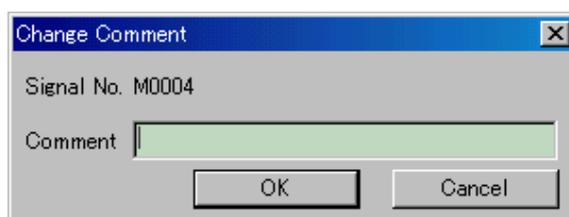
3.2.8 输入/编辑说明

用[Edit Comment (编辑说明)]列表框可以输入或编辑线圈等的说明。

1. 从菜单栏中点击[Edit (编辑)] - [Edit Comment (编辑说明)]。如下的对话框出现。



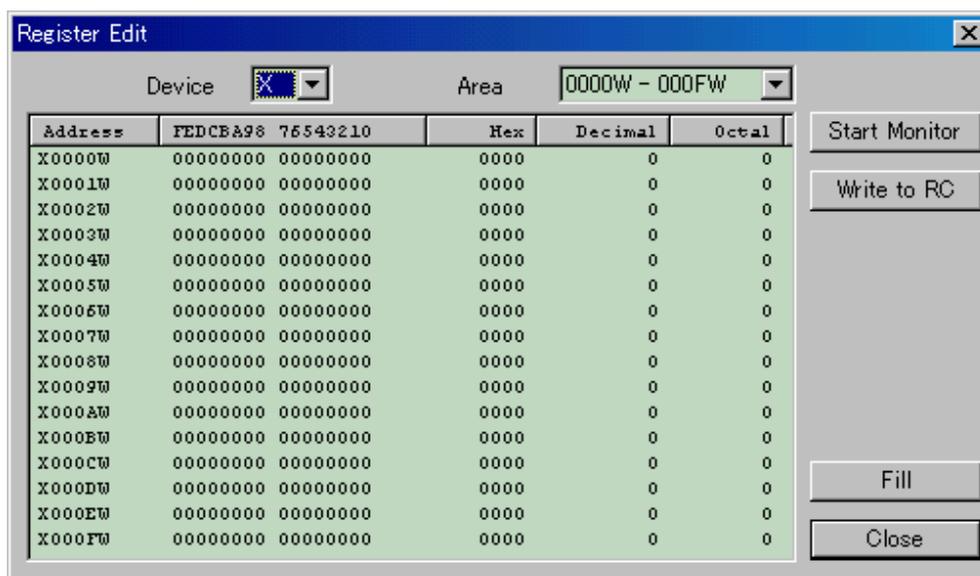
2. 在[Device (设备)]上点击下拉箭头，并选择想要的设备。信号编号(地址)的一览表出现。双击要编辑说明的信号编号。
3. [Change Comment (更改说明)]对话框出现。输入/编辑说明并点击<OK>。



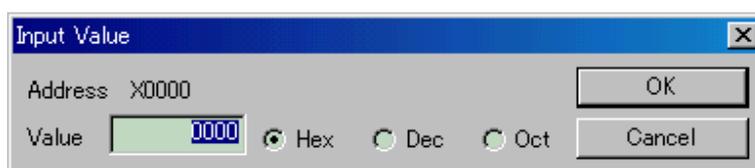
3.2.9 输入/ 编辑寄存器

按以下程序设定寄存器。

1. 在菜单栏中点击 [Edit (编辑)] - [Edit Register Data (编辑寄存器数据)]。如下的对话框出现。



1. 通过选择 [Device (设备)] 和 [Area (显示范围)] 来显示想要的寄存器。
3. 双击想要的信号编号 (地址), 显示如下画面。



4. 选择设定值是否用十六进制、十进制或八进制数来表示, 并在 [Value (设定值)] 中输入数值。点击 <OK>。

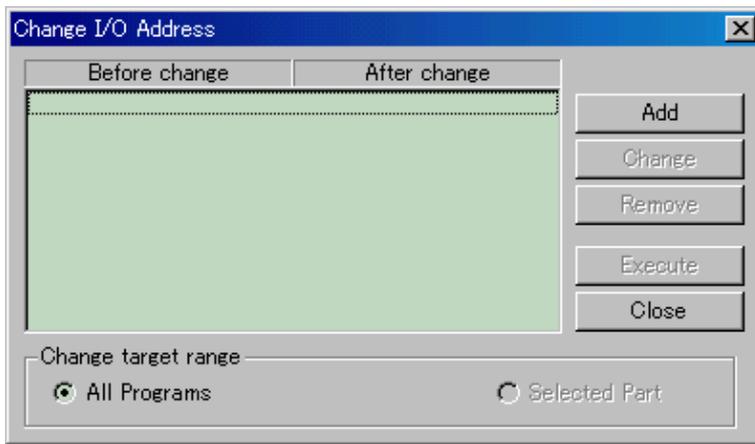
3.2.10 块编辑

3.2.10.1 编辑全部信号编号

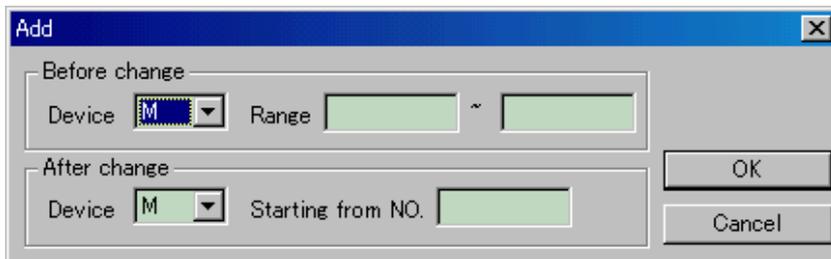
可以同时编辑线圈和触点的全部信号编号。要更改作为应用命令操作数（寄存器和常数）使用的信号编号，请参阅 3.2.10.2 更改应用命令的操作数。

更改全部程序的信号编号

1. 选择 [Edit (编辑)] - [Contact Change (触点更改)] - [I/O Address (I/O 地址)]。

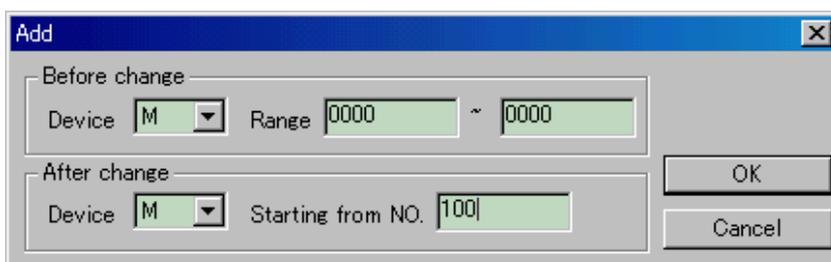


2. 点击<Add (追加)>。

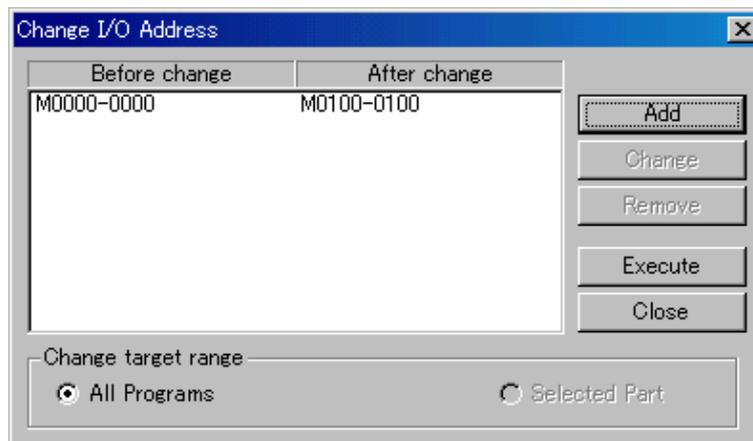


3. 设定要更改的设备。

例 将 M0000 更改为 M0100:



4. 点击<OK>。在变更一览表中显示变更的设备。



5. 点击<Execute (执行)>来更改。

更改指定范围的信号编号

1. 选择要更改数据的范围。要选择范围，点击步骤编号处并拖拽。选择范围为高亮显示。



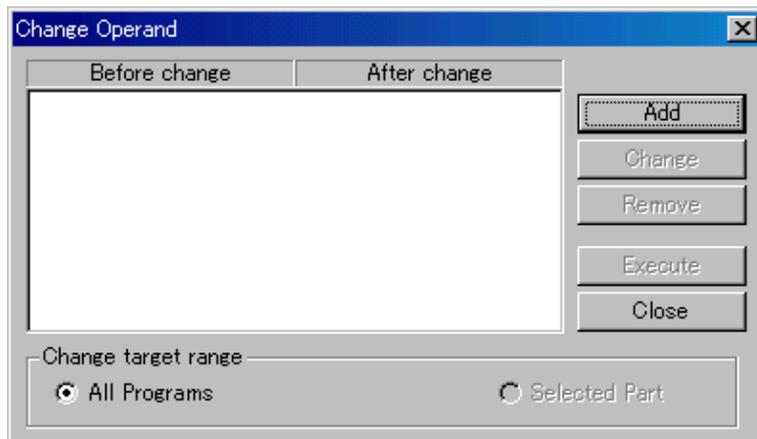
2. 选择[Edit (编辑)] - [Contact Change (触点更改)] - [I/O Address (I/O 地址)]。
3. 选中[Selected Part (选择的范围)]选择框。以下为按上述介绍的更改信号编号的程序进行操作。

3.2.10.2 更改应用命令的操作数

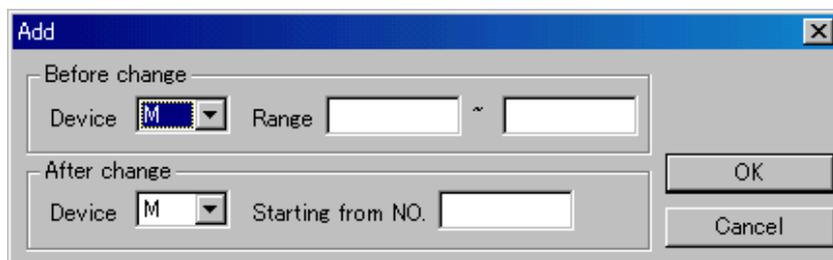
可以同时编辑全部用于应用命令的信号编号。（有关应用命令的更多信息，请参阅 2.6 AS 语言。）

更改全部程序的信号编号

1. 选择 [Edit (编辑)] - [Contact Change (触点更改)] - [Function Operand (应用命令操作数)]。

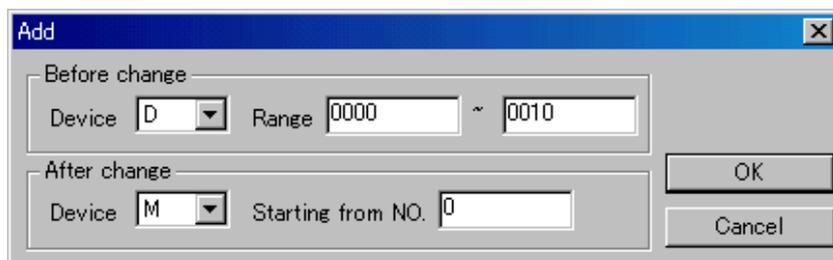


2. 点击 <add (追加)>。

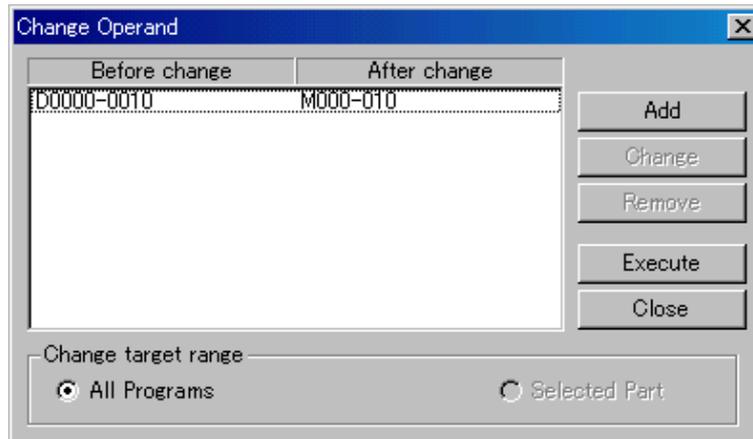


3. 选择要更改的操作数。

例 将作为操作数使用的信号 D0000 - D0010 更改为 M000W - M010W:



4. 点击<OK>。在变更一览表中显示变更的设备。



5. 点击<Execute (执行)>来更改。

更改指定范围的信号编号

1. 选择要更改数据的范围。要选择范围，点击步骤编号处并拖拽。选择范围为高亮显示。



2. 选择[Edit (编辑)] - [Contact Change (触点更改)] - [Function Operand (应用命令操作数)]。
3. 选中[Selected Part (选择的范围)]选择框。以下为按上述介绍的更改信号编号的程序进行操作。

3.2.10.3 转换闭合和断开触点

可以同时转换选择的信号编号的闭合触点（A 触点）和断开触点（B 触点）

1. 选择[Edit（编辑）] - [Contact Change（触点更改）] - [AB Contact（AB 触点）]。



2. 输入信号编号并点击<Execute（执行）>。

更改指定范围的信号编号

1. 选择要更改数据的范围。要选择范围，点击步骤编号处并拖拽。选择范围为高亮显示。



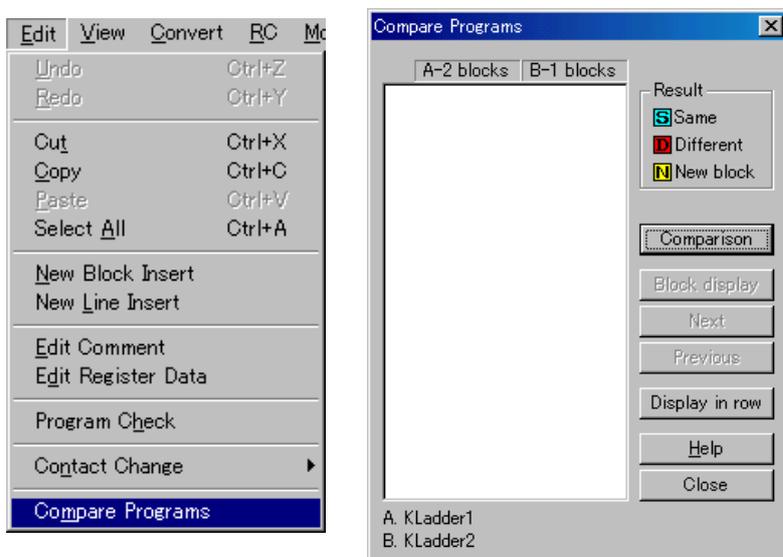
2. 选择[Edit（编辑）] - [Contact Change（触点更改）] - [AB Contact（AB 触点）]。
3. 选中[Selected（选择的部分）]选择框。以下为按上述介绍的更改信号编号的程序进行操作。

3.2.11 块比较

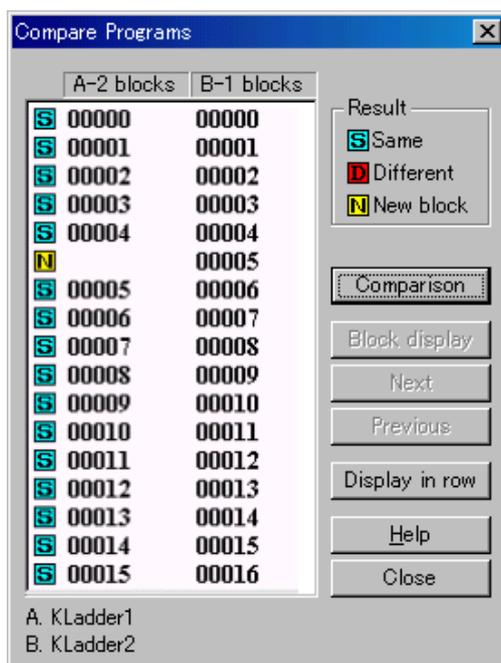
可以比较 2 个数据文件，找出不同的块或仅在 1 个文件中存在的块。本功能可以用于比较更改前更改后的程序并掌握修改处等。在这种情况下，以不同名来保存编辑前和编辑后的程序。

1. 显示要比较的 2 个数据文件。

选择[Edit (编辑)] - [Compare Programs (比较程序)]。如下的对话框出现。点击 <Comparison (比较)>，开始 2 个数据的比较。



2. 相同块显示为[S]，不同块显示为[D]，在 1 个文件中存在的块显示为[N]。显示的数值表示块数。

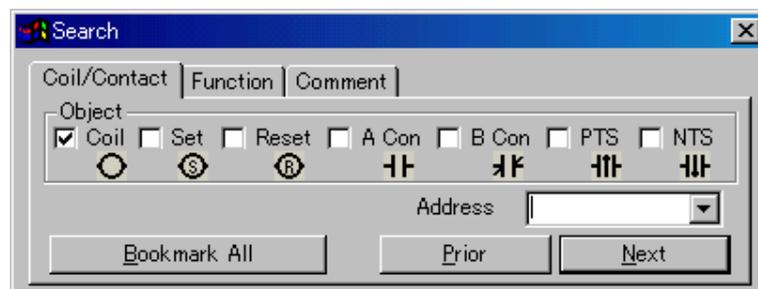


3. 在一览表中选择块编号，并点击<Block display (块显示)>。在梯形图中相应的块为高亮显示。双击块编号也会同样显示。
4. 点击<Next (下一个)>或<Previous (上一个)>显示下一个或上一个不同的或仅在 1 个文件中存在的块。

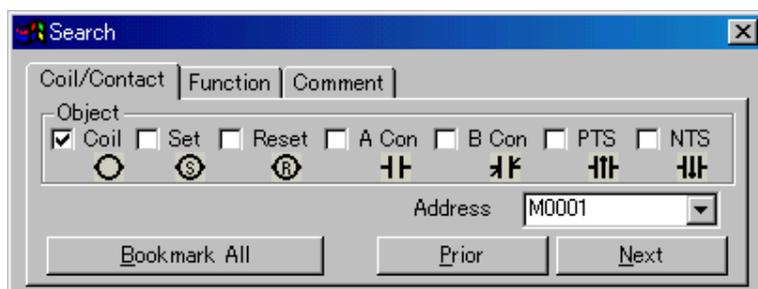
3.2.12 检索

可以指定字符串检索活动窗口中显示的回路数据。

1. 选择[View (显示)] - [Search (检索)]。如下的菜单出现。



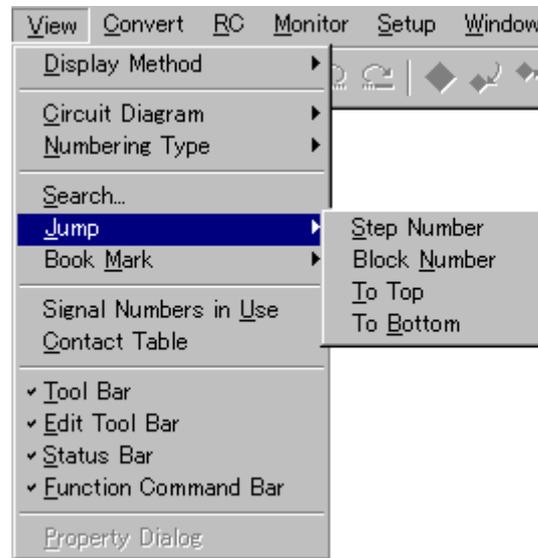
2. 根据要检索的项目，点击 [Coil/ Contact (线圈/触点)]、[Function (功能)]、或 [Comment (说明)] 标签。
3. 在 [Object (检索对象)] 中选中要检索的设备类型的选择框。
4. 在 [Address (地址)] (或 [Text (文本)]) 中输入要检索的字符串，然后点击<Previous (上一个)>或<Next (下一个)> 开始检索。



点击<Bookmark All (全部书签)>来设定满足检索条件的所有数据的书签。(请参阅 3.2.14 使用书签跳转跳转。)

3.2.13 跳转

显示指定回路。选择[View (显示)] - [Jump (跳转)]并指定跳转目标。



Step number (步骤编号) 显示指定步骤编号的回路。

Block number (块编号) 显示指定块编号的回路。

To top (到首) 显示文件中的第一回路。

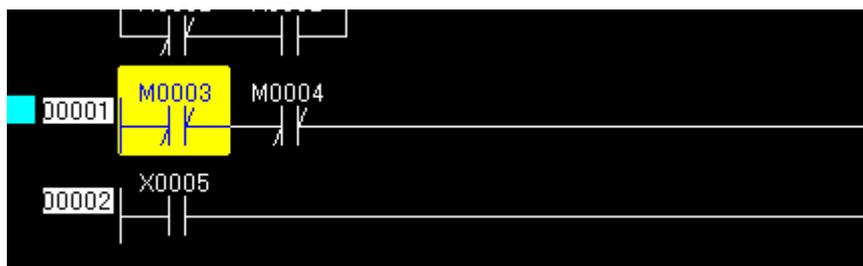
To bottom (到末) 显示文件中的最后回路。

3.2.14 使用书签跳转

可以使用书签迅速跳过回路图中特定处。

设定书签

1. 在目标行放置光标 (黄色箱)。
2. 点击 。在选择行前显示小蓝色标记, 请看如下图的左侧。



跳转到书签

要跳转到光标下面的书签，点击 。要跳转到上述光标上面的书签，点击 。

要解除书签，

1. 将光标（黄色箱）放到要解除的书签行。
2. 点击 。

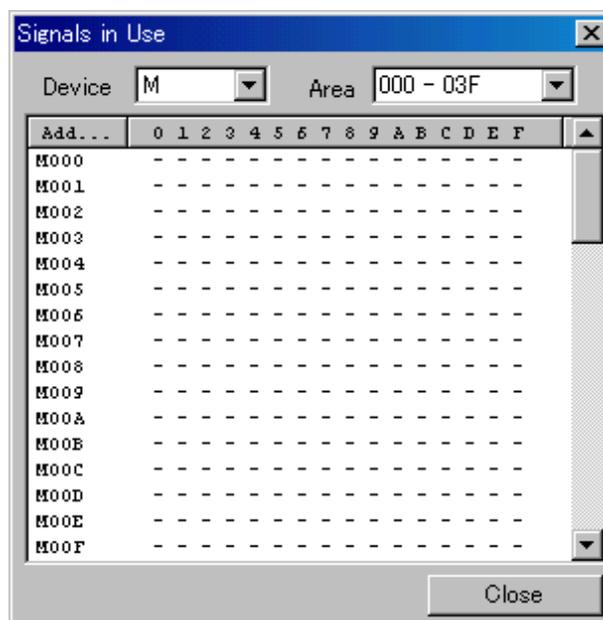
要解除全部书签，点击 。

在下拉菜单的[View（显示）] - [Bookmark（书签）]中可以进行上述操作。

3.2.15 信号状态

显示活动窗口中显示的文件使用的信号编号（地址）。

选择[View（显示）] - [Signal Numbers in Use（使用的信号编号）]，并从[Device（设备）]一览表中选择设备。



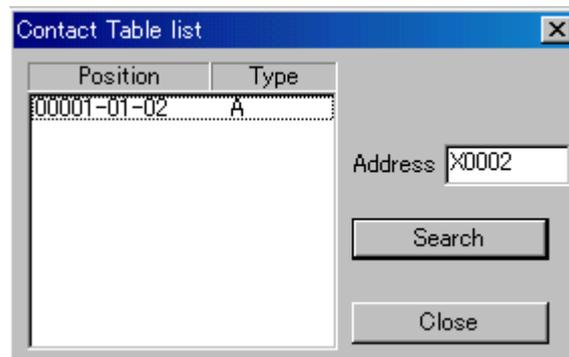
3.2.16 触点表

将线圈和触点的位置用信号编号来指定，并显示在一览表中。

1. 选择[View（显示）] - [Contact Table（触点表）]。



2. 在[Address (地址)]中输入要检索的信号编号，并点击< Search (检索)>或按 **Enter**。在一览表中显示检索结果。信号的位置显示为“块编号-行编号-列编号”。



3. 双击一览表中的触点。光标移动到梯形图中的触点。

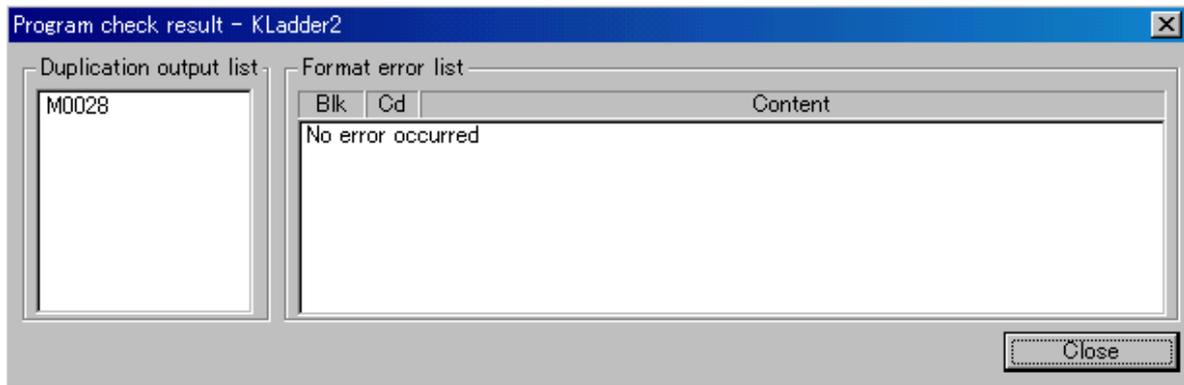
3.2.17 程序检查

检查所选择文件的错误，在一览表中显示错误。双击一览表中的错误，检索回路上的错误处，并显示。

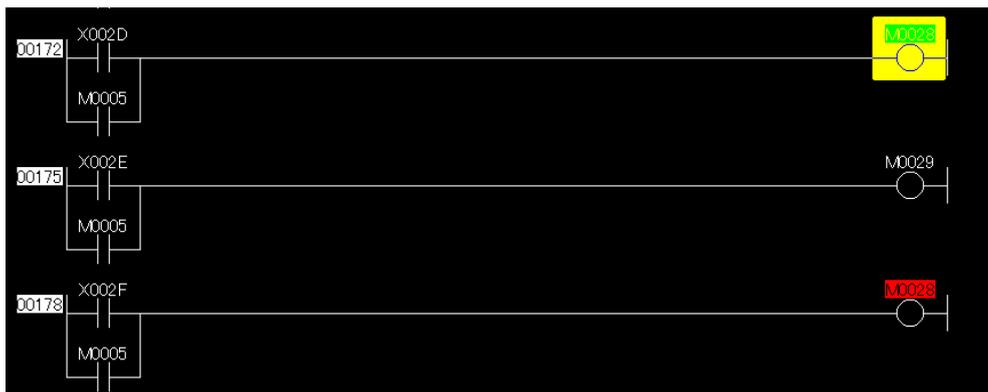
选择 [Edit (编辑)] - [Program Check (程序检查)]。[Program check result (程序检查结果)]对话框显示。

在 1 个程序中有重复输出时

1. 当在 1 个程序中使用的信号编号超过 1 次时，[Duplication output list (重复输出一览表)] 中显示信号编号。

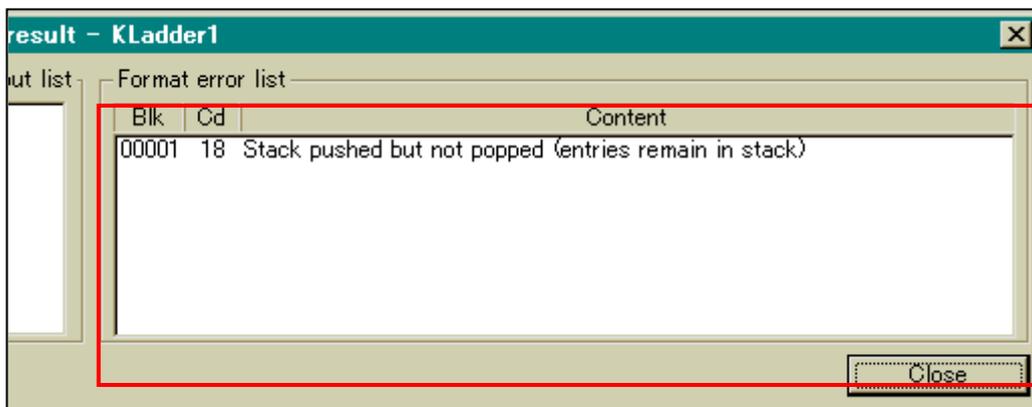


2. 双击一览表的信号编号，检索程序，显示梯形图的重复信号。



有格式错误时

1. 程序中有语法错误时，[Format error list (格式错误一览表)] 中显示所有错误。



2. 双击一览表的错误，检索程序，显示梯形图的错误。

3.3 RC（机器人控制器）菜单

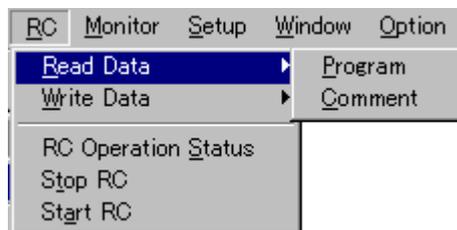
从本菜单进行从/向机器人控制器（RC）读写程序操作、机器人动作状态确认。

3.3.1 从 RC 读取文件

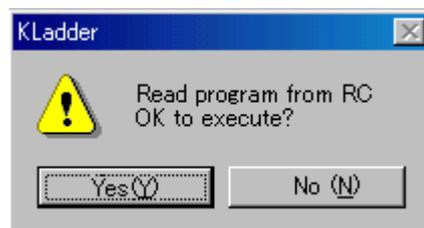
将保存在机器人控制器内存中的程序和说明读取到 KLadder 应用中。

3.3.1.1 程序的读取

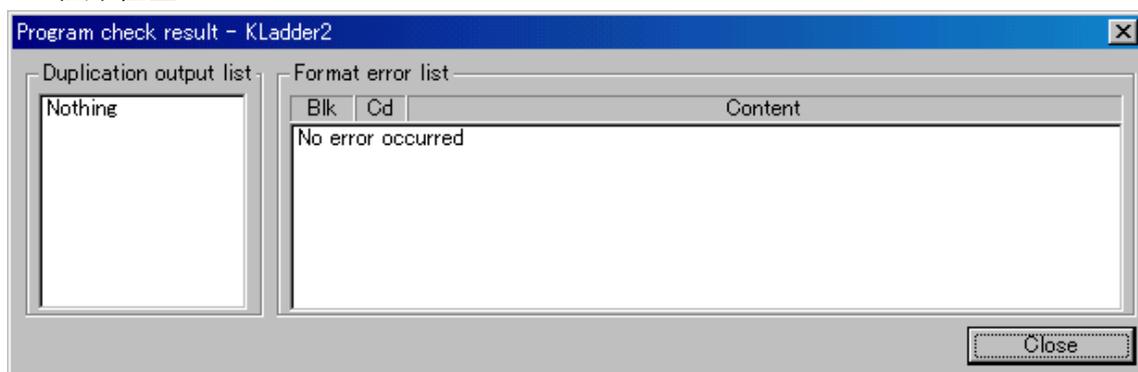
1. 从 RC 菜单的[Read Data（读取数据）] 中点击[Program（程序）]。



2. 点击<Yes（是）>，如下信息出现。



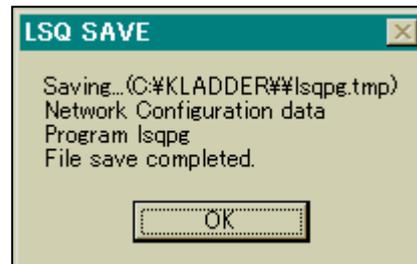
3. 读取程序到 KLadder 后，检查程序。当检查结束时，如下的画面出现。（请参阅 3.2.17 程序检查。）



3.3.1.2 说明的读取

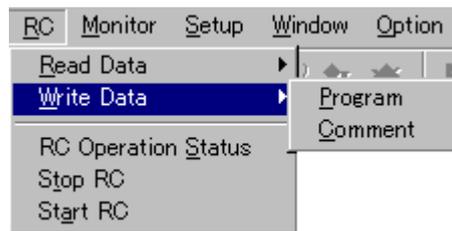
从机器人控制器读取说明。

1. 选择[RC]—[Read Data (读取数据)]—[Comment (说明)]。
2. 点击<OK>，如下信息出现。



3.3.2 写入到 RC (机器人控制器)

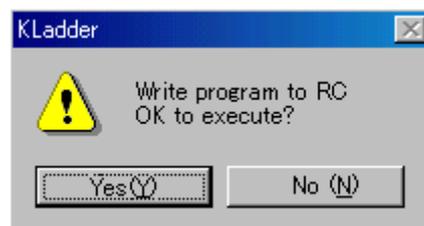
用 KLadder 将程序和说明写入到机器人控制器内存中。



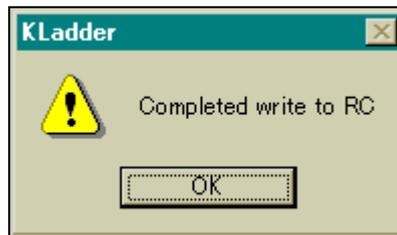
3.3.2.1 程序的写入

将程序写入到机器人控制器中。

1. 选择[RC] - [Write Data (写入数据)] - [Program (程序)]。
2. 如下信息出现，点击< Yes (是) >。



3. 写完后，如下信息出现。



当机器人控制器的程序正在运行中，写入程序时，确认信息出现。

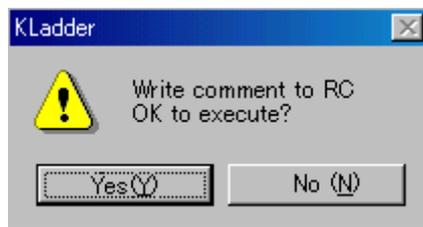


如果点击< Yes（是）>，选择新的写入程序，在执行当前运行程序的最后一步后，将立即执行。

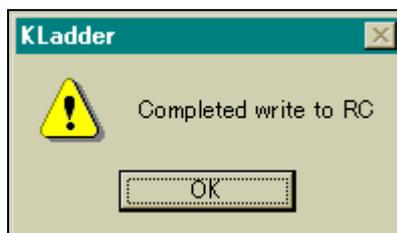
3.3.2.2 说明的写入

将说明写入到 RC。

1. 选择[RC] - [Write Data（写入数据）] - [Comment（说明）]。
2. 如下信息出现，点击< Yes（是）>。



3. 写入完成后，显示如下信息。



3.3.3 RC 的动作状态

显示机器人控制器的当前的动作状态。

1. 选择[RC] - [RC Operation Status (RC 动作状态)]。
2. 显示机器人控制器的当前的动作状态信息。

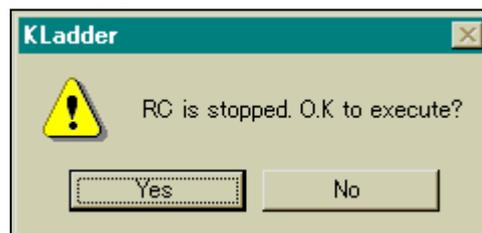


有关检查动作状态的更多详情，请参阅 2.6.1.1 LSQSTATUS。

3.3.4 RC 程序的停止

使用 KLadder 可以停止机器人控制器运行中的程序。

1. 点击[RC] - [Stop RC (停止 RC)]。
2. 如下信息出现，点击< Yes (是) >。



3. 程序停止时，如下信息出现。



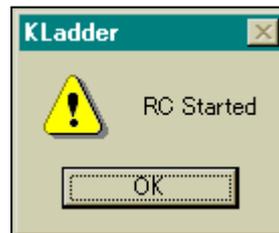
3.3.5 RC 程序的起动

使用 KLadder 可以起动机器人控制器程序。

1. 选择[RC]—[Start RC (起动 RC)]。
2. 如下信息出现，点击< Yes (是) >。



3. 程序起动时，如下信息出现。



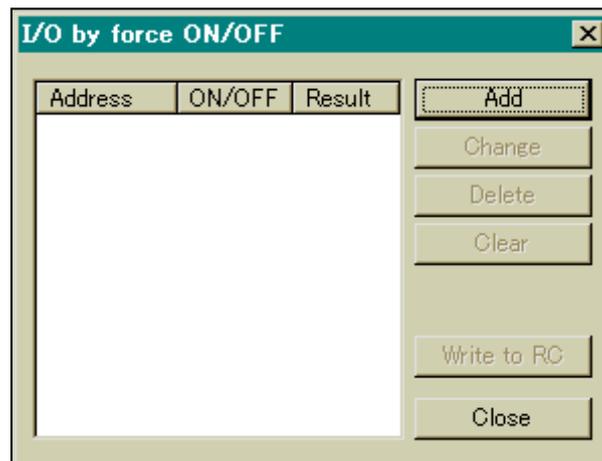
3.3.6 I/O 信号的强制 ON/ OFF

使用 KLadder，将机器人控制器中执行的用于程序的 I/O 信号强制 ON/OFF。

1. 从菜单栏的[Setup (设定)]中选择 [I/O Forced ON/OFF (I/O 信号的强制 ON/OFF)]。



2. 在出现的对话框中设定/ 更改信号编号 (地址)。



设定信号编号

点击<Add（追加）>。

在出现的对话框中输入信号编号（地址）。设定可能的信号编号范围：

Y0000 - Y07FF

M0000 - M07FF

K0000 - K01FF



点击 并选择 ON 或 OFF。

更改信号编号

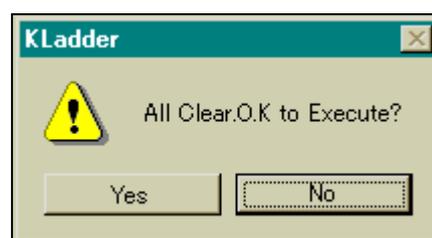
选择要更改的信号编号，并点击<Change（更改）>。输入信号编号，并在出现的对话框中选择 ON 或 OFF。

删除信号编号

选择要删除的信号编号，并点击<删除>。

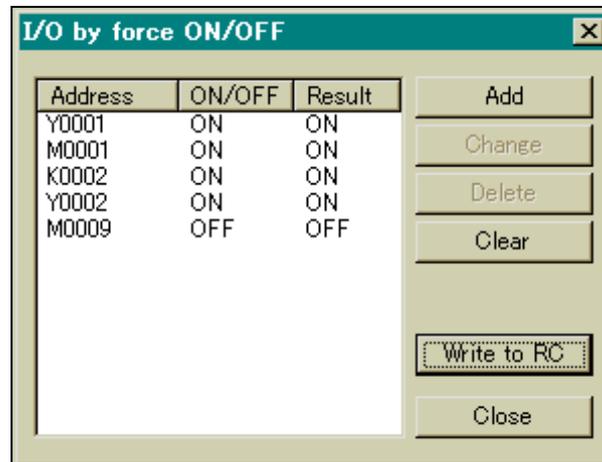
清除信号编号

点击<Clear（清除）>，如下信息出现。点击< Yes（是）>，删除所有信号编号。



3. 将更改的信号写入到机器人控制器。

点击< Write to RC (向 RC 写入) >。根据是否编辑修改的信号，将强制执行 ON/ OFF。从机器人控制器读取结果，并在[Result (结果)]栏中显示。



但，在机器人控制器起动过程中，顺序程序运算结果的信号状态优先于在此设定的信号状态。

[注 意]

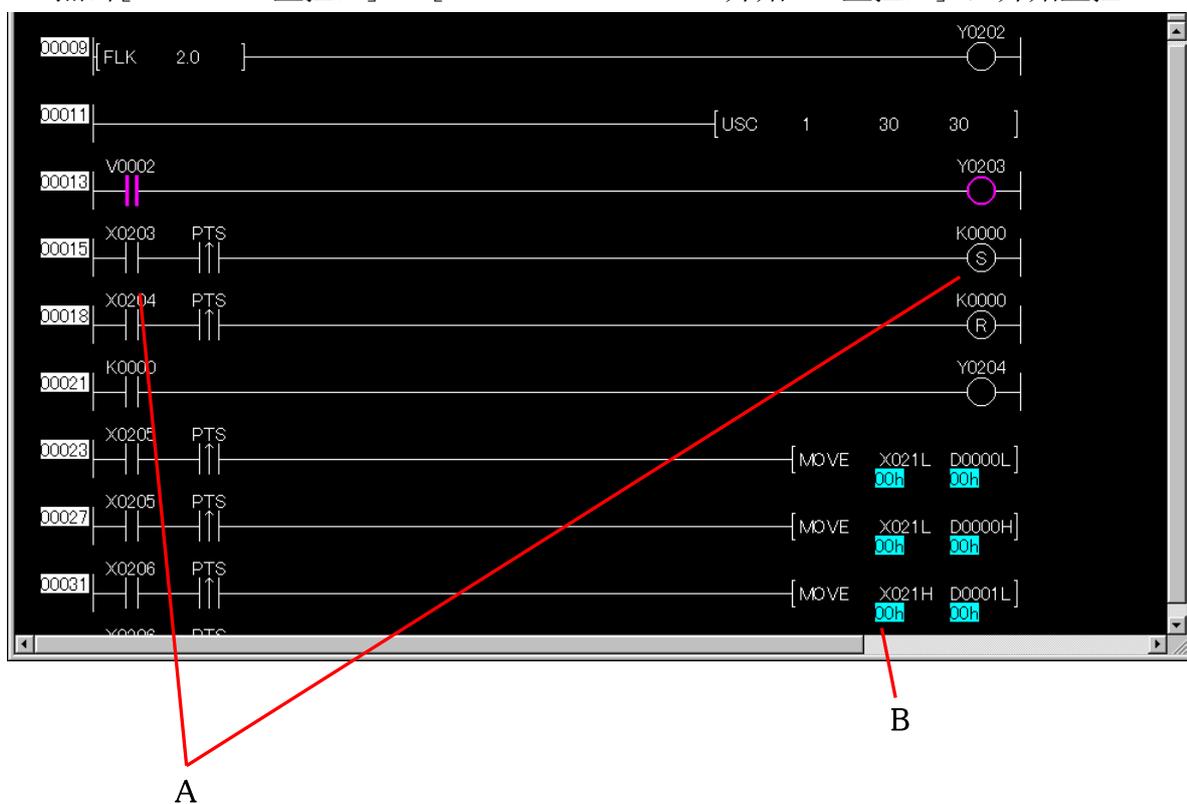
本功能强制信号 ON/OFF，但不能强制信号在程序中保持相同的状态。因此，当机器人控制器正在运行程序时，即使信号强制 ON/ OFF，其状态也将根据顺序运算结果或最新结果变化。

3.4 监控菜单

3.4.1 RC 监控开始

当用串联或以太网连接已安装 KLadder 的 PC 和机器人控制器时,在画面上显示程序的梯形图时,可以监控程序执行状态。

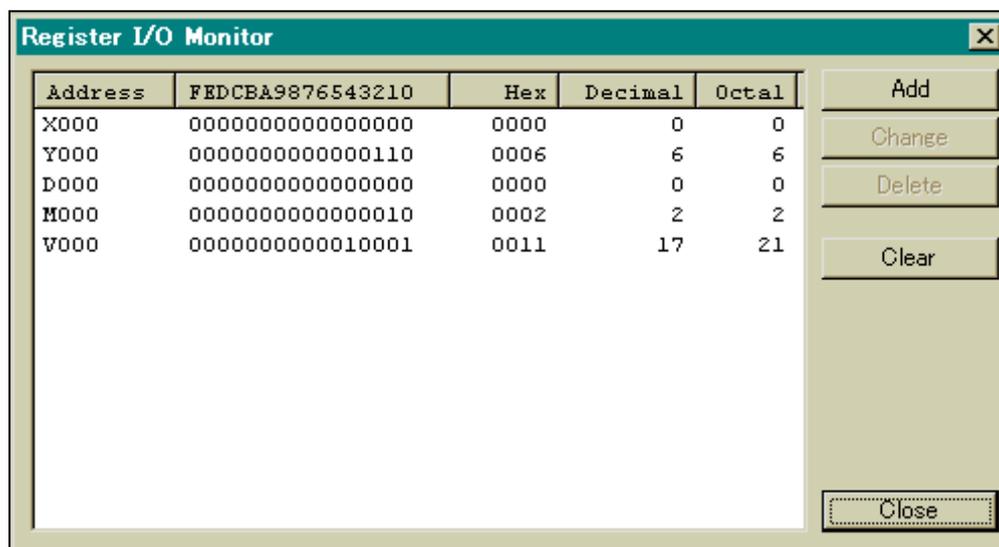
1. 选择[RC] - [Read Data (读取数据)] - [Program (程序)], 从机器人控制器读取程序数据。
2. 点击[Monitor (监控)] - [Start RC-Monitor (开始 RC-监控)], 开始监控。



- A. 用红色显示当前动作中的输入输出符号。
 - B. 用蓝色显示当前动作中的寄存器的信号编号。
3. 选择[Monitor (监控)] - [Register (寄存器) + I/O Address Monitor (I/O 地址监控)], 显示选择的寄存器和 I/O 信号编号。

3.4.2 寄存器·I/O 信号显示器

选择[Start RC-Monitor (开始 RC-监控)], 显示被监控的寄存器和 I/O 信号。



< Add (追加) >

输入另一个要监控并显示的信号编号。

< Change (更改) >

选择信号编号, 并点击此按钮来更改该信号。

< Delete (删除) >

选择信号编号, 并点击此按钮来删除此画面中被监控的信号。

< Clear (清除) >

删除此监控功能画面中的全部信号编号。

3.5 程序文件管理

3.5.1 数据的保存

改写数据

选择[File (文件)] - [Save (保存)], 改写制作中的文件。当文件是新建文件, 输入文件名的对话框将出现。点击也可以保存文件。

另存为

选择[File (文件)] - [Save as (另保存为)], 将当前文件另存为 1 个新文件。原文件保持不变。

3.5.2 文件和程序的二进制比较

比较当前文件和已选择的文件, 并显示比较结果。选择[File (文件)] - [Compare Program and File (比较程序和文件)]。

3.5.3 说明的输入

可以将选择的说明文件读取到活动窗口所显示的文件中。选择[File (文件)] - [Import Data (输入数据)] - [Comment (说明)]。

数据文件名和说明文件名不同时, 使用本功能。文件名相同时, 与数据文件一同读取说明 (只有在[Option (选项)] - [Configuration (配置)] - [File (文件)]菜单中的[When reading program (读取程序时)]选中[Read comment together with program (与程序一同读取说明)]时)。

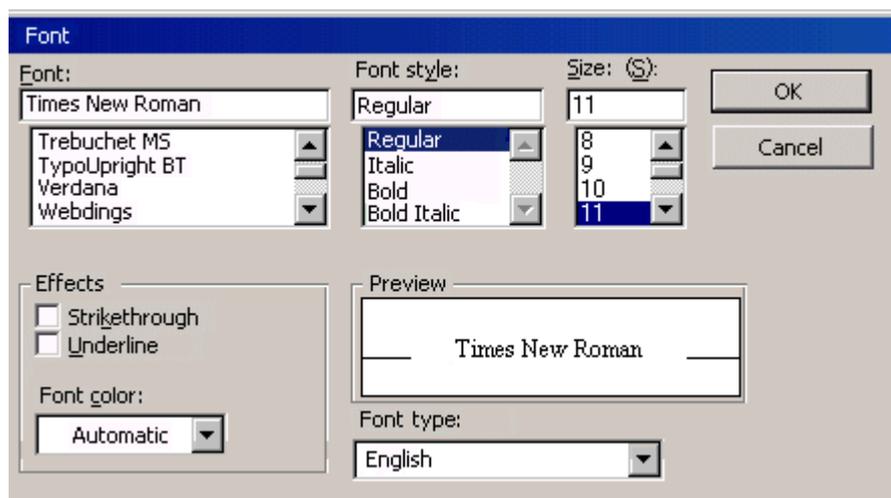
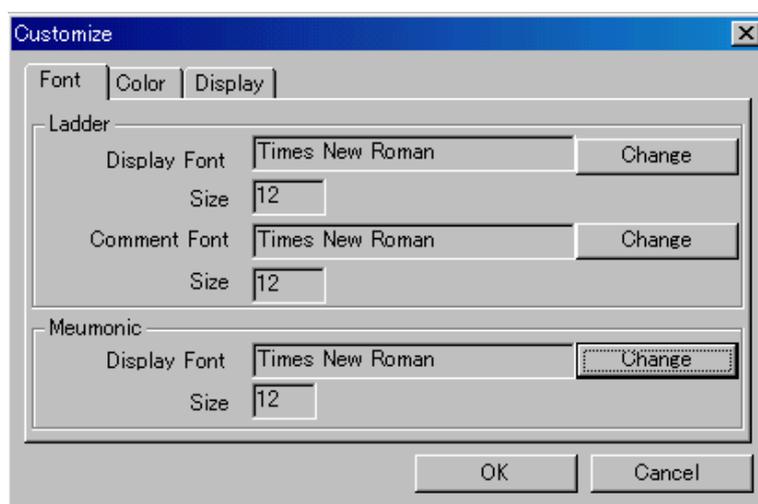
3.6 选项菜单

3.6.1 自定义

可以自定义 KLadder 画面显示属性，更改字符的字体和大小、显示的颜色等等。从菜单栏 [Option (选项)] 中选择 [Customize (自定义)]。

3.6.1.1 字体的自定义

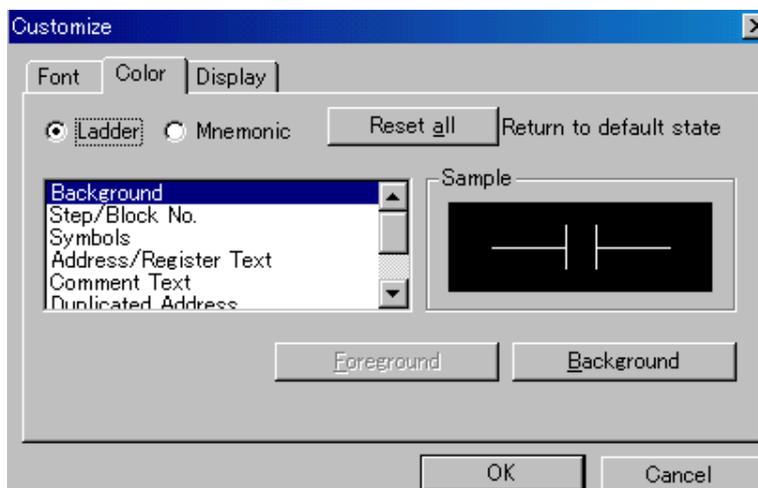
可以更改梯形图显示和助记符显示中的字符的字体和大小。当更改字符大小时，梯形图中显示的大小也随之更改。从菜单栏 [Option (选项)] 中选择 [Customize (自定义)]，并点击出现的对话框中的 [Font (字体)] 标签。



点击 < Change (更改) >。在 [Font (字体)] 任意设定的如下的对话框出现。

3.6.1.2 颜色的自定义

可以更改显示的颜色。在菜单栏的[Option (选项)]中选择[Customize (自定义)]。在出现的对话框中点击[Color (颜色)]标签。



1. 选择[Ladder (梯形)] 或[Mnemonic (助记符)]。
2. 点击一览表中要更改的颜色。
3. 可以更改选择项目的前景色和/或背景色。点击< Foreground (前景色)>或< Background (背景色)>。[Set color (设定颜色)]对话框出现, 选择想要的颜色, 并点击 <OK>。

Background (背景色) (Ladder, Mnemonic)

数据窗口的背景色。点击< Background (背景色)> 按钮来更改此颜色。

Step/ Block No. (步骤/块编号) (Ladder, Mnemonic)

步骤或块编号的颜色 (前景色) 和其背景色。

Symbols (符号) (Ladder)

符号的颜色。点击 <Foreground (前景色)>按钮, 更改其颜色。

Address/ Registers Text (地址/ 寄存器文本) (Ladder)

信号编号和寄存器的颜色 (前景色) 和其背景色。

Comment Text (说明文本) (Ladder, Mnemonic)

说明的颜色 (前景色) 和其背景色。

Duplicated signals (重复信号) (Ladder)

程序中重复的线圈信号编号的颜色和其背景色。

Monitor Active Symbols (监控活动符号) (Ladder)

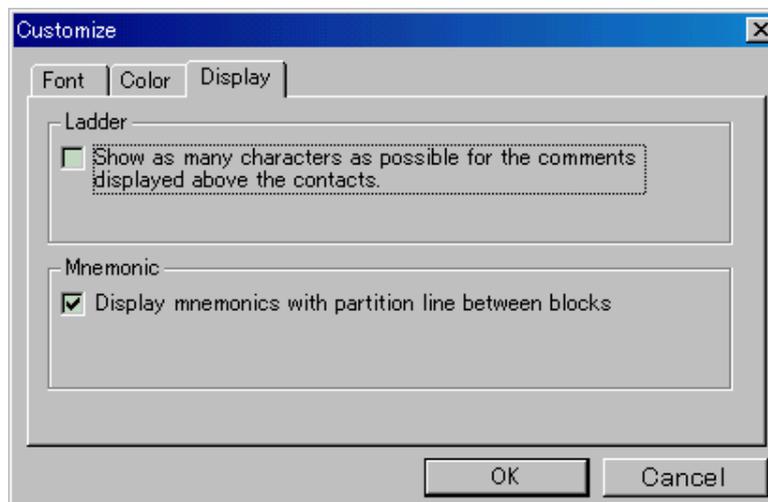
监控 RC 时正在处理的输入和输出符号的颜色。点击 < Foreground (前景色) > 按钮来更改此颜色。

Monitor Active Address (监控活动地址) (Ladder)

监控 RC 时正在处理的地址 (信号编号) 和设定值的颜色 (前景色) 及其背景色。

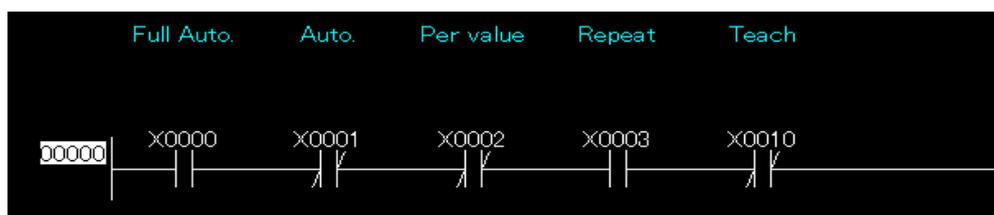
3.6.1.3 显示的自定义

如下所示可以更改显示的属性。在菜单栏中的[Option (选项)]中选择[Customize (自定义)]。在出现的对话框中点击 [Display (显示)] 标签。



[Show as many characters as possible for the comments displayed above the contacts. (触点上的说明显示尽可能多的字符。)]

当选择此选项，并且从菜单栏中选择[View (显示)]-[Circuit Diagram (回路图)]-[Circuit-Comment2 (回路-说明2)]时，触点上的说明显示如下。(最多30个字符。)(随字体类型/大小字符的变化，显示数变化。)



[Display mnemonic with partition lines between blocks. (在块之间插入分割线显示助记符。)]

如下图所示在块之间画 1 条分割线。

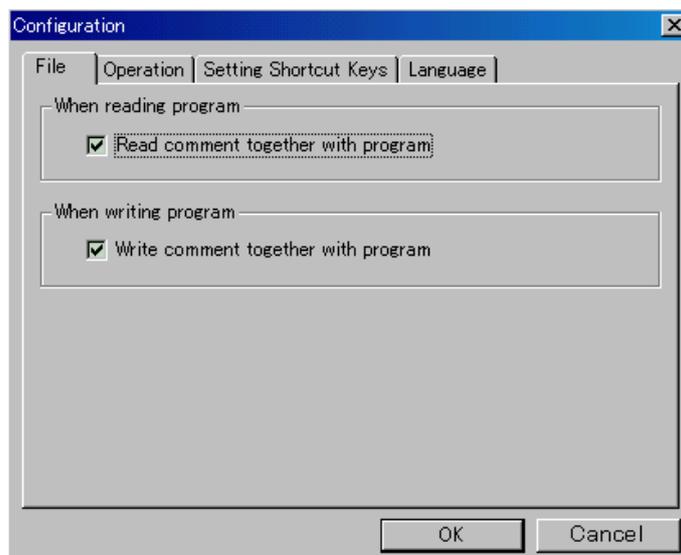
00000	LSQ STR	&V0001
00001	LSQ TMR	&T0000 200.0
00003	LSQ STR	&M0001
00004	LSQ AND	&M0002
00005	LSQ OUT	&Y0001
00006	LSQ STR	&X0001
00007	LSQ ANDN	&M0003
00008	LSQ OUT	&Y0002

3.6.2 KLadder 设定

可以在[Option (选项)] - [Configuration (配置)]中更改 KLadder 操作设定。

3.6.2.1 文件设定

读/写程序时设定条件。在菜单栏中选择[Option (选项)] - [Configuration (配置)], 并在出现的对话框中点击[File (文件)]图标。



[When reading program (读取程序时)]

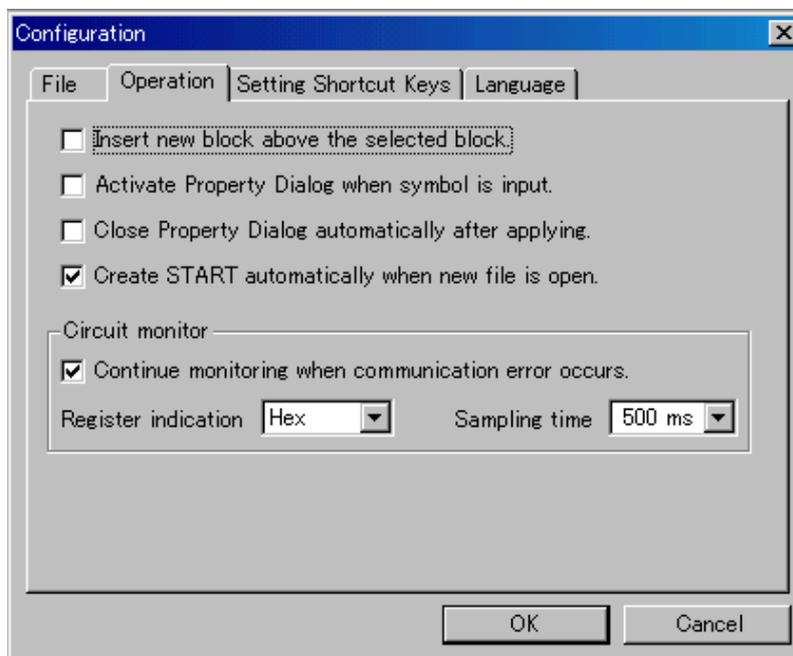
选择[Read comment together with program (与程序一同读取说明)], 向 KLadder 读取程序时, 也将一同读取说明文件。

[When writing program (写程序时)]

选择[Write comment together with program (与程序一同写说明)], 当从 KLadder 向机器人控制器写入程序时, 也将一同写入说明文件。

3.6.2.2 操作设定

在本菜单中进行梯形图画面和回路监控的操作设定。在菜单栏中选择[Option (选项)] - [Configuration (配置)], 并在出现的对话框中点击[Operation (操作)]图标。



[Insert new block above the selected block. (在选择的块上插入新块。)]

选择此选择框，在光标的位置的上一行插入新块。如果不选择此选择框，在光标位置的下一行插入新块。

[Activate Property Dialog when symbol is input. (输入符号时，属性对话框有效。)]

选择此选择框，自动显示输入信号编号、说明等的属性对话框，每次在回路上配置新的符号。(按键盘的功能键或工具栏上的符号图标的任何一个时，属性对话框立刻显示。)如果不选择此选择框，属性对话框不出现。

[Close Property Dialog automatically after applying. (适用后，自动关闭属性对话框。)]

选择此选择框，按<Apply (应用)>按钮后，属性对话框自动关闭。

[Create START automatically when a new file is open. (当打开一个新文件时，自动创建 START。)]

选择此选择框，START 作为第一命令打开一个新文件。

[Continue monitoring when communication error occurs. (通信错误产生时，继续监控。)]

监控过程中错误产生时，选择此选择框，并继续监控。如果不选择此选择框，即使错误产生，监控也停止。

[寄存器显示]

显示寄存器内容时，设定显示类型。从 16 进制、10 进制、8 进制中选择。

[Sampling time (取样时间)]

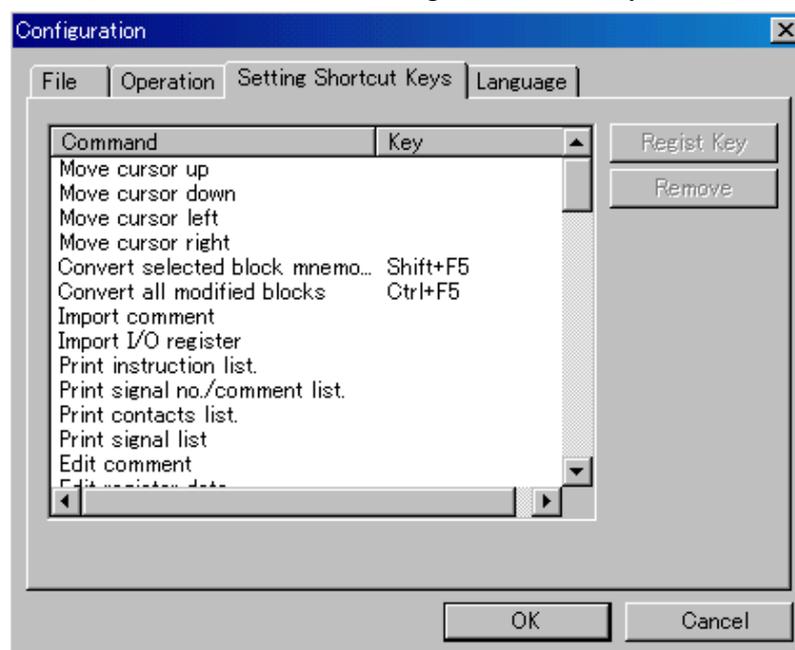
设定监控时的取样时间。设定范围为 50 ms - 3000 ms。

[注 意]

较短的取样时间会使机器人 CPU 的负荷增大。这样将会影响 PC 程序的处理速度等设定，因此不要进行不必要的高速设定。

3.6.2.3 设定快捷键

可以为经常使用的指令定义快捷键。从菜单栏中选择[Option (选项)] - [Configuration (配置)]，并在出现的对话框中点击[Setting Shortcut Keys (设定快捷键)]标签。



<Regist key (登录键)>

为已选择的指令定义快捷键。

1. 从一览表中选择指令。

2. 点击<Regist key（登录键）>。
3. 在键盘上，按作为指令的快捷键使用的键，也可以并用 **Shift** 和 **Ctrl** 来登录快捷键。

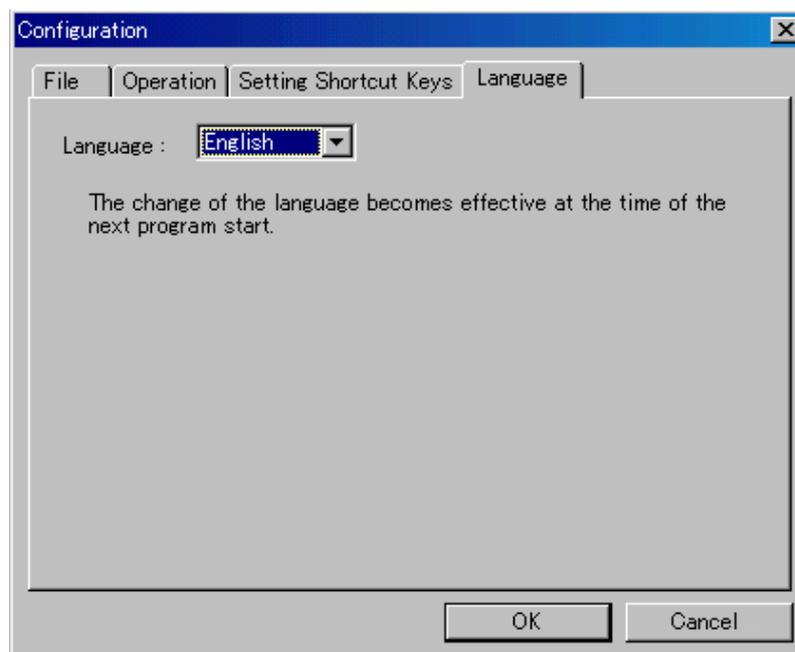
< Remove（删除）>

删除登录的快捷键。

1. 从一览表中选择要删除的指令。
2. 点击<Remove（删除）>。

3.6.2.4 语言

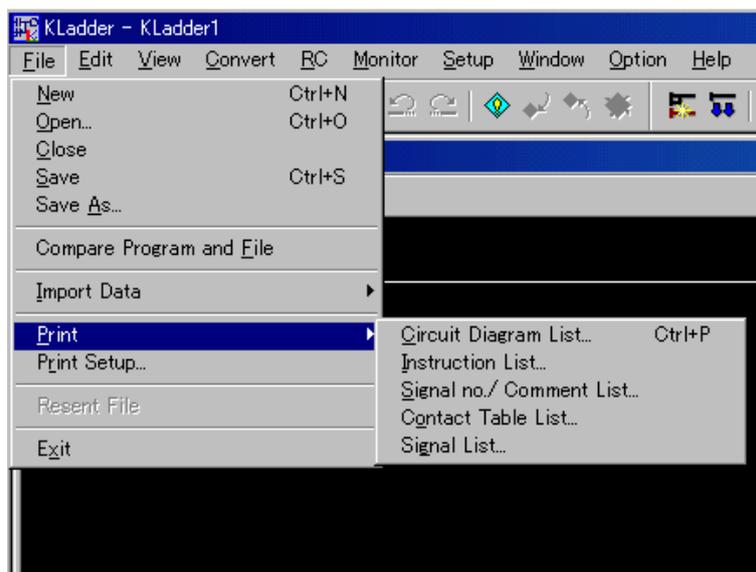
选择用于 KLadder 显示的语言。从菜单栏中选择[Option（选项）] - [Configuration（配置）]，并在出现的对话框中点击[Language（语言）]标签。选择想要的语言。从下一次 KLadder 启动起显示语言变更。



3.7 打印文件

3.7.1 打印方法

可以打印出当前选择的文件内容。选择 [File (文件)] - [Print (打印)]。从出现的菜单中选择要打印的项目。



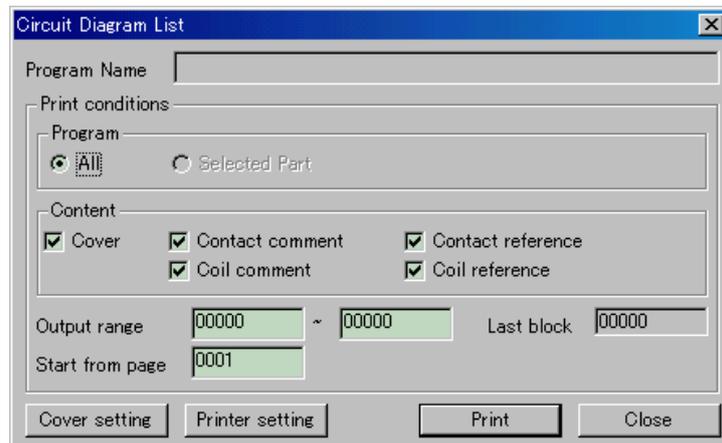
可以打印的项目如下：

- Circuit Diagram list (回路图一览表)
- Instruction List (命令一览表)
- Signal no./Comment List (信号编号/说明一览表)
- Contact Table List (触点表一览表)
- Signal List (信号一览表)

3.7.2 回路图一览表

选择[File (文件)] - [Print (打印)] - [Circuit Diagram List (回路图一览表)]

来打印梯形图。点击工具栏上的  也可以打印梯形图。



[A11 (全部)]

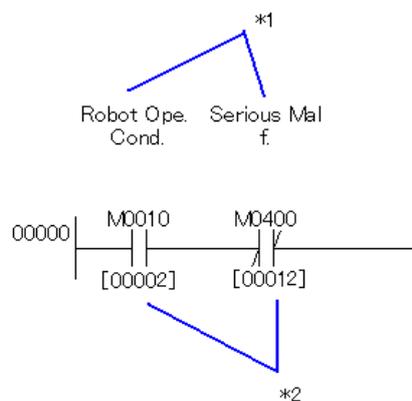
打印活动窗口内显示的当前全部梯形图。

[Selected part (选择部分)]

仅打印选择 (高亮) 区域。

[Cover (封面)]

打印封面。在封面上打印在<Cover setting (封面设定)>中设定的制作者、程序名和日期。



[Contact comment (触点说明)]

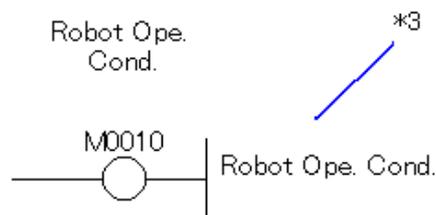
打印上述触点信号编号的说明。 *1

[Contact reference (触点参考)]

打印触点下的线圈存在块数。 *2

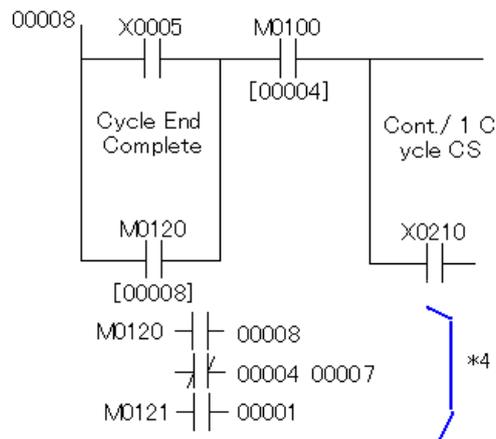
[Coil comment (线圈说明)]

打印线圈说明。 *3



[Coil reference (线圈参考)]

打印各线圈的触点存在块数。 *4



[Output range (输出范围)]

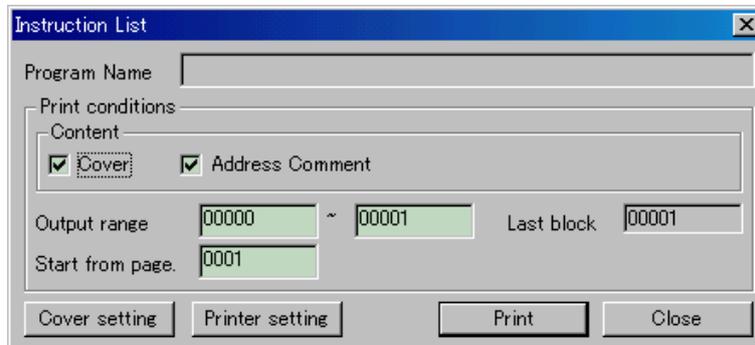
指定要打印的块编号。

[Start from page (开始页面)]

打印一览表的第一页面的页数。

3.7.3 命令一览表

选择[File (文件)] - [Print (打印)] - [Instruction List (命令一览表)]来打印用于文件中的命令一览表。在出现的对话框中设定必要的条件并点击 <Print (打印)>。



[Cover (封面)]

打印封面。在封面上打印在<Cover setting (封面设定)>中设定的制作者、程序名和日期。

[Address Comment (地址说明)]

打印线圈的说明。

[Output Range (输出范围)]

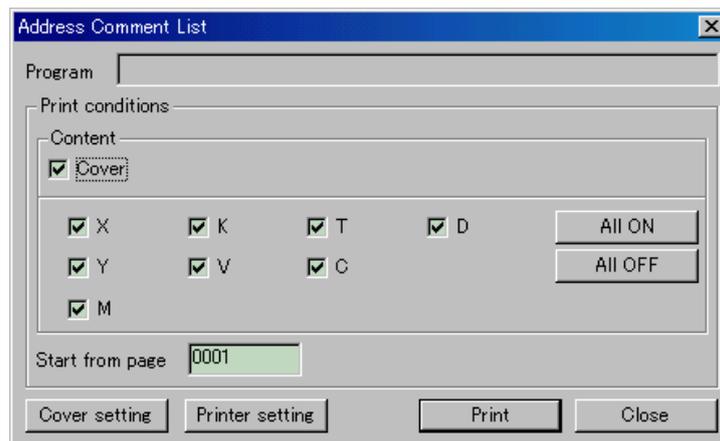
用步骤编号指定打印范围。为[Program (程序)]选择[A11 (全部)]时，无效。

[Start from page (开始页面)]

打印一览表的第一页面的页数。

3.7.4 地址说明一览表

选择[File (文件)] - [Print (打印)] - [Signal no./ Comment List (信号编号/说明一览表)] 来打印文件中信号的说明一览表。在出现的对话框中设定必要的条件，并点击 <Print (打印)>。



[Cover (封面)]

打印封面。在封面上打印在<Cover setting (封面设定)>中设定的制作者、程序名和日期。

<All ON (全部ON)>

选择全部打印设备。

<All OFF (全部OFF)>

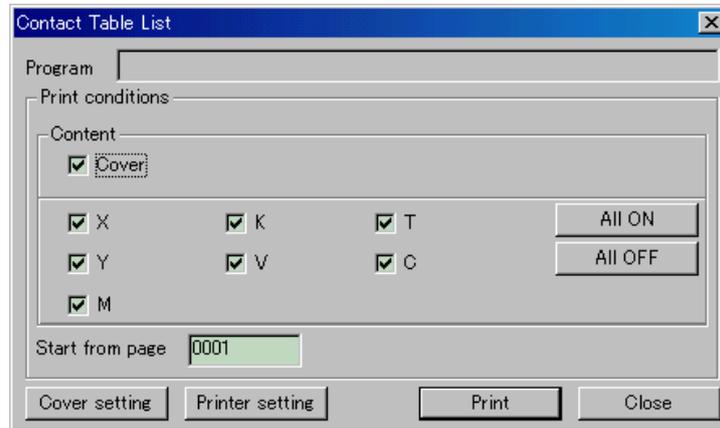
取消全部打印设备的选择。

[Start from page (开始页面)]

打印一览表的第一页面的页数。

3.7.5 触点表一览表

选择[File (文件)] - [Print (打印)] - [Contact Table List (触点表一览表)], 打印文件中各线圈的触点使用位置的一览表。在出现的对话框中设定必要的条件, 并点击<Print (打印)>。



[Cover (封面)]

打印封面。在封面上打印在<Cover setting (封面设定)>中设定的制作者、程序名和日期

<All ON (全部ON)>

选择全部打印设备。

<All OFF (全部OFF)>

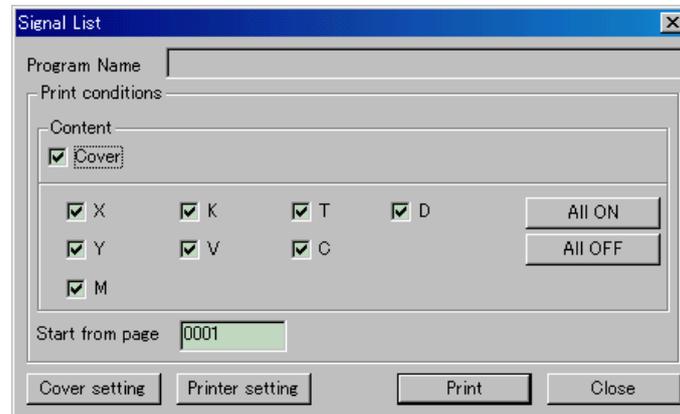
取消全部打印设备的选择。

[Start from page (开始页面)]

打印一览表的第一页面的页数。

3.7.6 信号一览表

选择[File (文件)] - [Print (打印)] - [Signal List (信号一览表)], 打印显示信号状态的一览表。在出现的对话框中设定必要的条件, 并点击 <Print (打印)>。



[Cover (封面)]

打印封面。在封面上打印在<Cover setting (封面设定)>中设定的制作者、程序名和日期。

<All ON (全部ON)>

选择全部打印设备。

<All OFF (全部 OFF)>

取消全部打印设备的选择。

[Start from page (开始页面)]

打印一览表的第一页面的页数。



川崎机器人控制器 E 系列
KLogic/ KLadder 操作手册

2014 年 8 月：第一版

川崎重工业株式会社出版

90210-1083DCA