

Robostar控制器

T1 系列 操作机运行说明书



- ☐ 使用及维护保养说明书
- ☒ 操作机运行说明书
- ☐ 程序说明书
- ☐ Uni-host说明书
- ☐ Alarm Code说明书

Robostar co., ltd.



Copyright 2017. ROBOSTAR Co., Ltd all rights reserved.

本使用说明书的制作权在Robostar 有限公司。

任何部分在没有得到Robostar的允许不能以其他形式或手段使用。

说明书中的内容可能在无预告的情况下变更。

产品保证相关

Robostar的产品在严格品质管理的监督下制造，其整体产品的保修期间自制造日起1年。在此期只对Robostar侧的过失导致的机械故障或正常使用中的设计及制造商的问题导致的故障无常服务。

以下几种情况无法保证无常服务。

- (1) 已满保修期间
- (2) 因贵公司或第三者不适当的修理、改造、移动或其他使用不注意导致的故障
- (3) 配件及润滑油使用了本公司指定的产品以外的产品而导致的故障
- (4) 因火灾、灾害、地震、风水灾害以及其他自然灾害导致的故障
- (5) 因喷料及被水淹等本公司产品配置以外环境中使用导致的故障
- (6) 消耗品消耗而导致的故障
- (7) 因未按照使用说明书中记载的保修检验作业内容执行而导致的故障
- (8) Robot修理以外的费用损失

Robostar有限公司地址及联系方式

- 总公司及工厂
京畿道安山市常绿区水仁路700
(沙士洞119-38)
700, Suin-ro, Sangnok-gu, Ansan-City,
Gyeonggi-do, Republic of South Korea
(15523)
- AS及产品咨询
- 营业咨询
TEL. 031-400-3600
FAX. 031-419-4249
- 顾客咨询
TEL. 1588-4428
- 水原工厂
京畿道水原市劝善区产业路155-37
(古索洞 997)
37, saneop-ro 155beon-gil,
Gwonseon-gu, Suwon-City,
Gyeonggi-do,
Republic of South Korea (16648)



www.robostar.co.kr

使用说明书的构成

本产品使用说明书的组成如下。首次使用本产品的顾客，请熟读所有说明书的内容后再使用。

■ 使用及维护保修说明书

在此部分，将叙述有关控制器的整体内容，即控制器的概述、安装及外部装置之间的接口方法等。

■ 操作及运行说明书

在此部分，将叙述整个控制器的使用方法及参数设置、JOB程序的编辑、机器人驱动等内容。

■ 程序说明书

在此部分，将叙述有关罗普伺达机器人程序RRL (Robostar Robot Language)及通过RRL编制机器人程序的方法。

■ Uni-host说明书

在此部分，将叙述罗普伺达的在线PC程序 - 单主机'

■ Alarm Code说明书

对报警代码进行说明。

本说明书的结构

本说明书结构如下.

■ 第一章. Teach Pendant (T/P) 结构及基本操作方法

Robot驱动中使用的T/P的结构和基本操作方法相关说明.

■ 第二章. EDIT 模式

Robot驱动所需的程序和Point、变量、参数输入方法相关说明.

■ 第三章. RUN 模式

驱动制作的Robot程序的方法相关说明.

■ 第四章. TEACH 模式

Robot的 JOG 模式移动, Point Teaching方法相关说明.

■ 第五章. I/O 模式

确认控制器外部信号输入输出的方法相关说明.

■ 第六章. ORIGIN 模式

控制器内部内置的ORIGIN 模式使用方法相关说明.

■ 第七章. INFO 模式

控制器信息及log确认, USB 使用等相关说明.

■ 第八章. MOVEMENT 模式

MOVEMENT 功能使用相关说明.

■ 第九章. SHUTDOWN

控制器正常结束相关说明.

■ 第十章. 附录

另外的参考事项相关说明.

■ 第十一章. 修正历史

修正完善内容相关记载.

목차

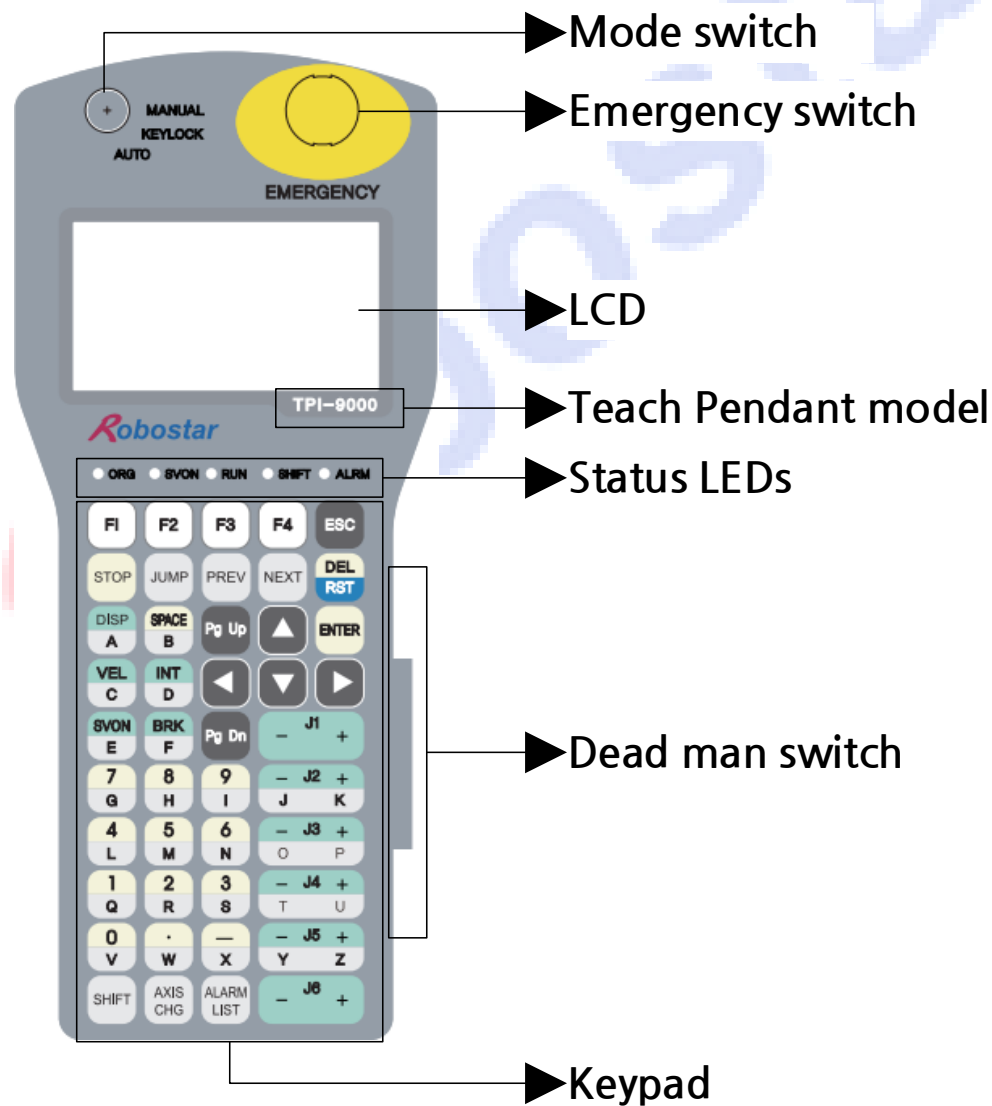
제1장	제어기 알람 코드 구성	오류! 책갈피가 정의되어 있지 않습니다.
제2장	제어기 알람 코드표.....	오류! 책갈피가 정의되어 있지 않습니다.
2.1	FILE SYSTEM.....	오류! 책갈피가 정의되어 있지 않습니다.
2.2	DEVICE.....	오류! 책갈피가 정의되어 있지 않습니다.
2.3	PROTECTION.....	오류! 책갈피가 정의되어 있지 않습니다.
2.4	RUN TIME	오류! 책갈피가 정의되어 있지 않습니다.
2.5	COMPILE.....	오류! 책갈피가 정의되어 있지 않습니다.
2.6	TRAJECTORY.....	오류! 책갈피가 정의되어 있지 않습니다.
2.7	EMERGENCY	오류! 책갈피가 정의되어 있지 않습니다.
2.8	SERVO BOARD.....	오류! 책갈피가 정의되어 있지 않습니다.
2.9	ETHERCAT SERVO DRIVER	오류! 책갈피가 정의되어 있지 않습니다.
제3장	개정	오류! 책갈피가 정의되어 있지 않습니다.

第1章 Teach Pendant (T/P) 结构及操作方法

- Teach Pendant (T/P)是控制器专用用户接口设备，提供Robot启动停止、Teaching、用户程序的生成及修正等功能。
- 本章节针对其结构及使用方法进行说明。

1.1 T/P 结构

- ✓ Teach Pendant的基本结构如下图。



1) Mode Switch

- Teach Pendant使用模式包括Auto 模式和Manual 模式两种，可以利用Mode 开关变更模式。
 - ① Manual 模式: 利用T/P控制Robot时使用.
 - ② Auto Mode: I/O 或通信等外部控制时使用.
- 此外, 为防止不必要的输入提供 KEYLOCK 功能.

2) Emergency Switch

- Robot动作的过程中发生异常动作时，或者系统发生异常时，紧急停止Robot动作而使用的开关.
- Emergency 开关按下 Teach Pendant 右上方红色开关即可停止Robot运行，开关维持按下的状态. 解除紧急停止时，顺时针方向旋转Emergency 开关即可解除紧急停止状态.

3) LCD

- 显示当前状态和很多其它信息给用户.

4) Status LEDs

- 显示Robot控制状态的部分，各LED的意义如下表.

LED	内容
ORG	原点信号亮灯
SVON	Servo电源ON时亮灯
RUN	执行程序驱动Robot时亮灯
SHIFT	使用SHIFT键时亮灯
ALARM	程序执行中发生Alarm时亮灯 程序执行中发生警告时闪烁

5) Dead Man Switch

- Deadman 黏贴在Teach Pendant侧面，是为了保证操作者的安全的开关. Robot 处于 CURR Mode时操作者如果判断为危险或者发生事故时，拿下此开关Servo 电源OFF 的同时Robot立即停止执行.

6) Keypad

- 提供菜单选择，数据输入，Robot操作等控制器操作所需的键.

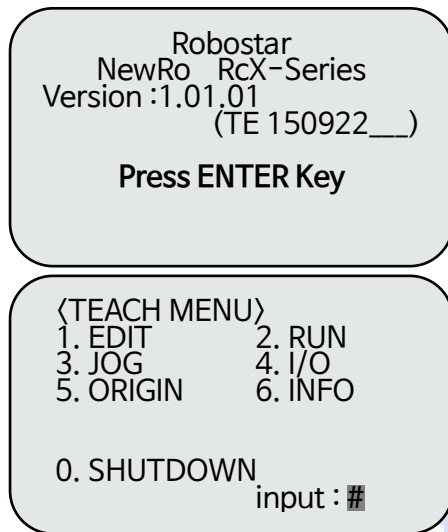
Robostar

1.2 LCD 画面结构及初始菜单

- ✓ Teach Pendant 连接到控制器供应电源可以确认如下画面，显示控制器类型 (ECAT(EtherCAT), RS422), 版本信息.
- ✓ 按下ENTER键切换到主画面.

Step 1.

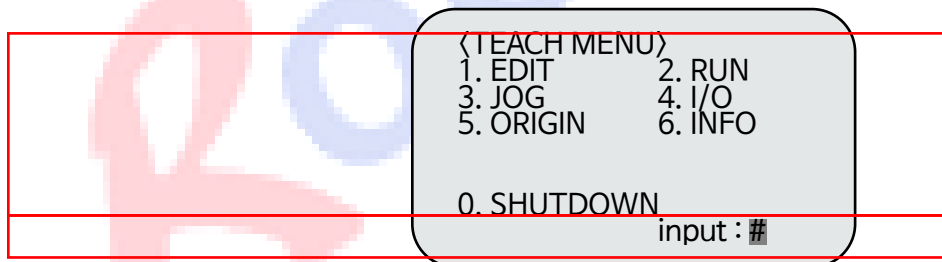
初始菜单画面



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始画面

- ✓ 出Alarm 消息等特殊情况，LCD 画面大致分为两种.



A. DATA 窗口

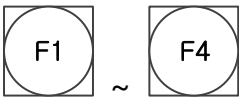
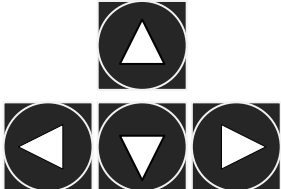
B. Command 窗口

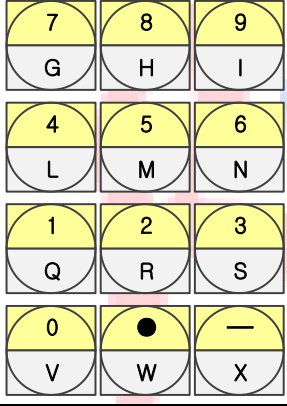
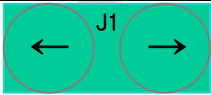
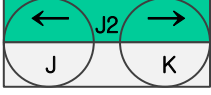
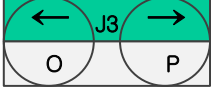
A. Data 窗口: 输出标题, 菜单, 输入数据, 状态信息等窗口.

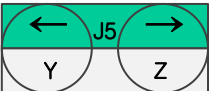
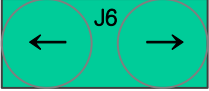
B. Command 窗口: 介绍菜单选择, 功能键输入, Update 情况等用户选择的窗口

1.3 操作键说明

- ✓ 针对各键的功能进行说明.

键	说明
	- 控制Robot的菜单功能键 F1~F6. - 下图中显示的Command 窗口中的部分相匹配. <EDIT POINT> ROBOT TYPE *R1 Q4AFL1 R2 NO-EXIT R3 NO-EXIT CURR MDI EDIT HOME     <OFFS> ZERO CALIBRATION T : 0 Z : 0 R1 : 0 R2 : 0 X : 0 6 : 0 T Z R1 R2    
	Robot驱动中执行停止功能.
	STEP 及编号移动时执行JUMP 功能.
	执行取消、恢复到前一画面等功能.
	转换到具有复数输入值的键值时使用.
	删除输入数据值或RESET时使用.
	切换画面Page时使用.
	完成输入或选择时使用.
	Cursor或当前位置移动时使用.
	- 监控快捷键 - 输入英文字母A时使用.

	- SPACE: 空一格 - 输入英文字母B时使用.
	- 执行速度变更. - 输入英文字母C时使用.
	- GLOBAL INTEGER 及 FLOAT变量快捷键 - 输入英文字母D时使用.
	- Servo电源 ON/OFF - 输入英文字母E时使用.
	- 点击Break解除 - 输入英文字母F时使用.
	移动到上一页
	移动到下一页
	轴变更Monitor
	Alarm List Monitor
	- 输入英文字母、数字时使用. - 支持输入小数点和负数.
	1轴Robot操作键
	2轴Robot操作键 - 输入英文字母J,K时使用.
	3轴Robot操作键 - 输入英文字母O,P时使用.

	• 4轴Robot操作键 • 输入英文字母T, R时使用.
	• 5轴Robot操作键 • 输入英文字母X,Y时使用.
	• 6轴Robot操作键

1.4 Alarm Reset

- ✓ 可以解除控制器Alarm.
- ✓ Alarm解除只能在Alarm LIST内执行.

Step 1.

选择Robot命令语

〈TEACH MENU〉
 1. EDIT 2. RUN
 3. JOG 4. I/O
 5. ORIGIN 6. INFO

 0. SHUTDOWN input : #



- 如下图Alarm LED亮灯或闪烁时 选择 "ALARM LIST" 键.



Step 2.

确认Alarm或警告内容

ECODE: 2011 [1/1]
 T/P emergency
 ROBOT: ALL

 [Description]
 1.Wen press TP
 EMG button, this EXIT



- 确认Alarm或警告内容时按下 "UP/DOWN" 键确认内容.

Step 3.

解除Alarm或警告

ECODE: 2011 [1/1]
 T/P emergency
 ROBOT: ALL

 [Description]
 1.Wen press TP
 EMG button, this EXIT



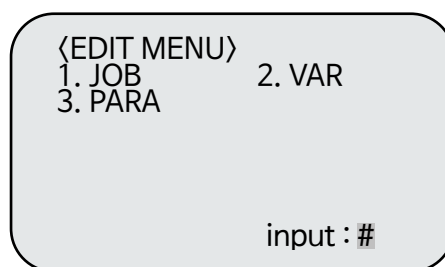
- 确认Alarm或警告的内容后解除Alarm 时按下 "RST" 按钮解除Alarm或警告.

〈TEACH MENU〉
 1. EDIT 2. RUN
 3. JOG 4. I/O
 5. ORIGIN 6. INFO

 0. SHUTDOWN input : #

第2章 EDIT 模式

- Edit 模式由3种模式组成，可以编辑Robot驱动所需的程序、变量、参数.
- Edit 模式如下图组成，提供编辑、保存及修正的功能.
 - 1) JOB Edit: 程序输入及编辑模式
 - 2) VAR (Variable) Edit: Robot控制器的GLOBAL (INTEGER, FLOAT) 编辑，确认模式
 - 3) PARA (Parameter) Edit: Robot形态及使用环境设定模式.



2.1 JOB

- JOB EDIT选择 Teach Pendant 画面中的 [1. JOB EDIT] 是显示在画面. 在此菜单内利用Robot语言(RRL)制作Robot文件 , 还可以进行文件复制、变更名称、删除等文件管理操作.

2.1.1 概要

- 保存、显示利用Robot语言 (RRL)制作的文件.
- 制作文件时Teach Pendant上的编辑功能.
- 一个文件最多可以输入2,000行
- 文件名称可以输入包含大写字母、数字的5位.
- 文件内容按照制作的日期顺序显示 , [JOB EDIT]中提供这些文件的生成, 复制, 修正, 查看内容等功能.

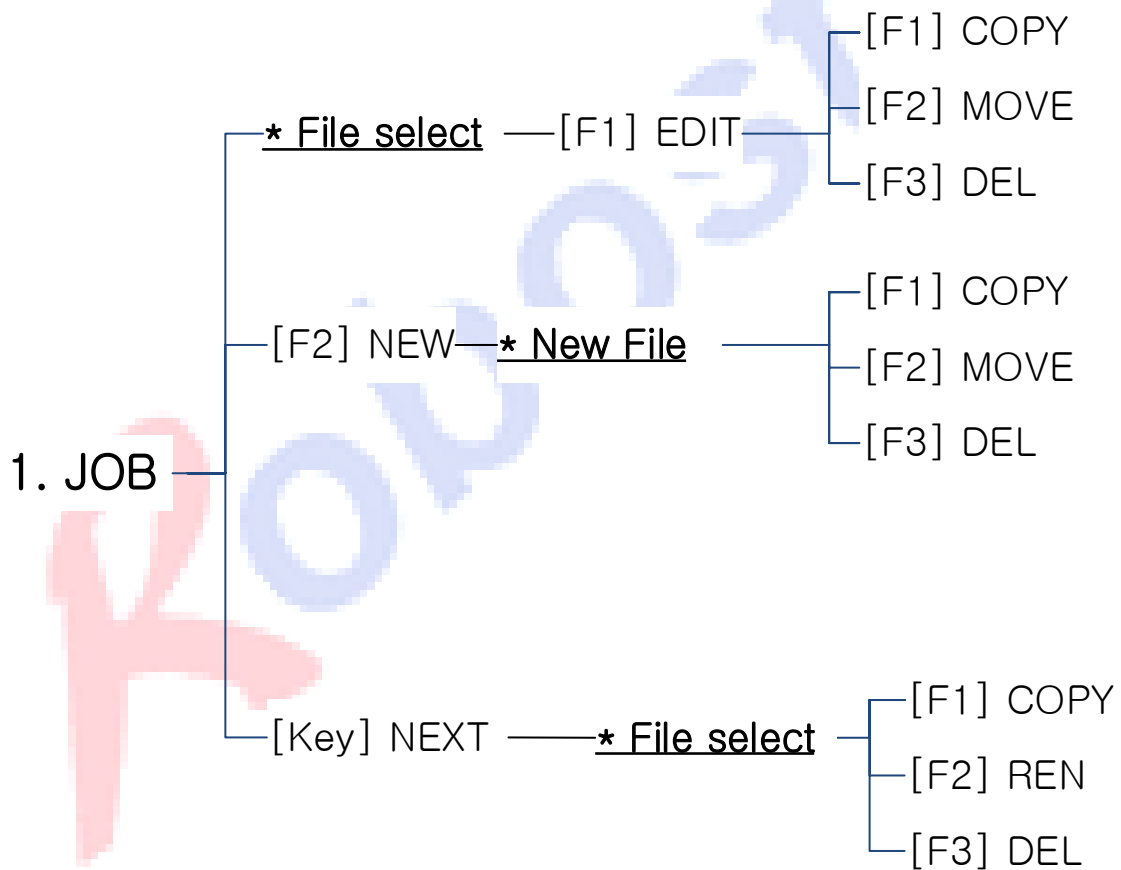
■ 1. JOB EDIT 种类

GROUP		内容	备注
1. JOB	[F1] EDIT	对现有JOB 内容编辑、复制、删除等管理的功能	
	[F2] NEW	生成新JOB 文件并编辑	
	[F1] COPY	复制现有JOB 文件复制为新JOB文件名	
	[F2] REN	变更现有JOB 文件名	
	[F3] DEL	删除选择的JOB 文件	

2.1.2 JOB 模式流程图

- Point大致由5个 Group组成.
 - EDIT: 对现有JOB文件内容进行编辑、复制、删除等操作.
 - NEW: 创建新JOB文件并制作JOB内容.
 - COPY: 复制现有JOB 文件为其它名称.
 - REN: 变更现有JOB 文件名为其它名.
 - DEL: 删除所选JOB 文件.

■ Job Menu Tree



2.1.3 JOB Group画面OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

ENTER

- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

EDIT选择

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

1

Q

- 选择 “1. EDIT”

POINT 选择

<EDIT MENU>
1. JOB 2. VAR
3. PARA

input : #

1

Q

- 选择 “1. JOB”

<JOB MENU> **(a)**
* 0. AA 8 STEP
1. DD 5 STEP
2. RF 7 STEP
(b) 3. WW **(c)** 12 STEP
4. ARE 100 STEP
5. QF 8 STEP

EDIT NEW **(d)**

- ① : Title bar
- ② : 文件名
- ③ : JOB Line个数
- ④ : Function

Step 3.

JOB文件编辑画面

```

<JOB MANU>
* 0.AA      8   STEP
  1.DD      5   STEP
  2.RF      7   STEP
  3.WW     12   STEP
  4.ARE    100  STEP
  5.QE      8   STEP
EDIT NEW    >

```



- 编辑、复制、删除现有JOB内容时，按下“F1”键选择“EDIT”。

创建新JOB文件

```

<JOB MANU>
* 0.AA      8   STEP
  1.DD      5   STEP
  2.RF      7   STEP
  3.WW     12   STEP
  4.ARE    100  STEP
  5.QE      8   STEP
EDIT NEW    >

```



- 创建新JOB文件时按下“F2”键选择“NEW”。

移动到下一个菜单

```

<JOB MANU>
* 0.AA      8   STEP
  1.DD      5   STEP
  2.RF      7   STEP
  3.WW     12   STEP
  4.ARE    100  STEP
  5.QE      8   STEP
EDIT NEW    >

```



- 移动到下一个菜单时按下“NEXT”键
- 有下一个菜单时菜单功能窗口中显示“<”。

JOB 文件复制

```

<JOB MANU>
* 0.AA      8   STEP
  1.DD      5   STEP
  2.RF      7   STEP
  3.WW     12   STEP
  4.ARE    100  STEP
  5.QE      8   STEP
< COPY REN DEL

```



- 现有 JOB文件复制为其它名称的文件时按下“F1”键选择“COPY”。

JOB 文件名变更

```

<JOB MANU>
* 0.AA      8   STEP
  1.DD      5   STEP
  2.RF      7   STEP
  3.WW     12   STEP
  4.ARE    100  STEP
  5.QE      8   STEP
< COPY REN DEL

```



- 现有JOB文件变更为其它名称时按下“F2”键选择“REN”。

JOB 文件删除

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
< COPY REN DEL		

移动到上一页

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
< COPY REN DEL		



- 删除现有 JOB文件时按下"F3" 键选择 "DEL" .



- 移动到上一页菜单时按下 "PREV" 键
- 有上一个菜单时菜单功能窗口中显示 "<" .

2.1.4 编辑方法

- 编辑JOB文件的功能.

■ OPEN 顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

Step 1.

移动页面

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
EDIT NEW		

<JOB MANU>		
* 6.TT	8	STEP
7.MM	5	STEP
8.FF	7	STEP
EDIT NEW		



- 利用"PAGE UP/DOWN" 键可以切换页面

Step 2.

Cursor移动

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
EDIT NEW		

<JOB MANU>		
0.AA	8	STEP
* 1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
EDIT NEW		



- 选择要编辑的文件
- 按下"UP/DOWN" 键选择要编辑的JOB文件
- 可以确认选择的JOB文件最前面显示"*"符号

Step 3

JOB文件编辑

<JOB MANU>		
0.AA	8	STEP
* 1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
EDIT	NEW	



- 编辑、复制、删除现有JOB内容时按下“F1”键选择“EDIT”。

Step 4

编辑画面

<EDIT JOB> ①	
STEP	INS ②
0000	func void main
③ 0001	take 1
0002	while 1 ④
0003	jmov ip[0]
0004	jmov ip[1]
COPY MOVE DEL ⑤	

- ① : Title bar
- ② : i → 插入模式, e → 编辑模式
- ③ : 程序行数(STEP)
- ④ : 程序内容
- ⑤ : Function

2.1.4.1 JOB Line移动

■ . OPEN 顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

编辑画面

Step 1

<JOB MANU>

* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP

EDIT NEW

<EDIT JOB>

STEP INS i

0000 func void main

*** 0001 take 1**

0002 while 1

0003 jmov ip[0]

0004 jmov ip[1]

COPY MOVE DEL

F1

- 编辑时按下“F1”键选择“EDIT”。

Step 2

JOB 内容编辑

<EDIT JOB>

STEP INS i

0000 func void main

*** 0001 take 1**

0002 while 1

0003 jmov ip[0]

0004 jmov ip[1]

COPY MOVE DEL

ENTER

- 变更选择的行JOB 内容时按下“ENTER”键

<EDIT JOB>

STEP INS

0000 func void main

*** 0001** ■

0002 take 1

0003 while 1

0004 jmov ip[1]

CMD FUNC OPER

- 新创建的JOB Line显示为“Cursor”。
- 现有JOB 内容下移到下一行。
- 请输入JOB 内容!

移动说明

Step 1.

页面移动

```

<EDIT JOB>
STEP  INS      i
0000 func void main
0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
  
```

```

EDIT JOB>
STEP  INS      i
* 0005 jmov ip[2]
0006 jmov ip[3]
0007 jmov ip[4]
0008 jmov ip[5]
0009 end
COPY  MOVE  DEL
  
```



- Teach Pendant 画面中无复制的内容时执行“**页面移动**”。
- 利用“PAGE UP/DOWN” 键可以进行画面切换

Step 2.

Cursor移动

```

EDIT JOB>
STEP  INS      i
* 0000 func void main
0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
  
```

```

<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
  
```



- 选择“**开始**”的Block Line时利用“**Cursor移动**”键移动..
- 按下“UP/DOWN” 键选择要编辑的JOB 文件
- 可以确认选择的Line最前面显示“*”符号

Step 3.

Jump

```

<EDIT JOB>
STEP  INS      i
* 0000 func void set_para()
0002 /*****
0003 /*****
0004 $ctype_num[1]
0005 $ctype_num[2]
COPY  MOVE  DEL

```



- 移动到输入的STEP编号时按下"JUMP"键

```

<EDIT JOB>
STEP  INS      i
* 0000 func void set_para()
0002 /*****
0003 /*****
0004 $ctype_num[1]
0005 $ctype_num[2]
STEP # =147

```



- 输入要移动的STEP编号后按下'ENTER'键

```

<EDIT JOB>
STEP  INS      i
* 0147 $e_fork_width[1]
0148 $e_fork_width[2]
0149 $e_fork_width[3]
0150 $e_fork_width[4]
0151 $e_fork_width[5]
COPY  MOVE  DEL

```

2.1.4.2 预处理器指令输入

- 预处理器指令包括“Robot指令, FUNC指令, 特殊符号” 3种.
- 可以利用提供的预处理器指令更方便的制作JOB.

■ . OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

Step 1

编辑画面

```
<JOB MANU>
* 0.AA      8    STEP
  1.DD      5    STEP
  2.RF      7    STEP
  3.WW     12    STEP
  4.ARE    100    STEP
  5.QE      8    STEP
EDIT NEW >
```

F1

- 要编辑时按下“F1”选择“EDIT”.

```
<EDIT JOB>
STEP INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
< COPY MOVE DEL
```

Step 2

JOB内容编辑

```
<EDIT JOB>
STEP INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
< COPY MOVE DEL
```

ENTER

- 在选择的Line中创建新的JOB内容时按下“ENTER”键

```
<EDIT JOB>
STEP INS      i
0000 func void main
* 0001 
0002 take 1
0003 while 1
0004 jmov ip[1]
< CMD FUNC OPER
```

- 新建的JOB Line显示为“Cursor”.
- 现有JOB 内容向下移动一行.
- 请输入JOB 内容!

Step 3

CMD 选择

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

```

<EDIT JOB>          DD
CMD INSTRUCT [1/10]
1: acc              2: amov
3: area:            4: bin
5: bmow             6: bout
7: cbin             8: cbout

```

Input #■



- 使用Robot指令时按下“F1”键选择“CMD”。

Step 4

FUNC 选择

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

```

<EDIT JOB>          DD
FUNC INSTRUCT [1/5]
1: abs              2: acos
3: asc              4: asin
5: atan             6: atan2
7: bins             8: chr

```

Input #■



- 使用函数指令时按下“F2”键选择“FUNC”。

Step 5

特殊符号选择

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```



- 使用特殊符号指令时按下“F3”键选择“OPER”。

<EDIT JOB> DD
OPER INSTRUCT [1/3]

1: +	2: -
3: *	4: /
5: !	6: %
7: &	8:

input #■

2.1.4.2.1 预处理器指令项

- 提供Robot指令.

■ CMD (Command) 项

- 黄色部分为按钮键编号.

命令语	内容	备注
acc	设置加速时间的百分比 (%)	
bin	以Byte为单位, 读取指定的Byte值.	
bout	以Byte为单位, 输出指定的Byte值.	
cbin	Fildbus, cc-link 输入以Byte为单位读取.	
cbout	Fildbus, cc-link 以Byte为单位输出.	
cdin	Fildbus, cc-link 输入以Double Word为单位读取.	
cdout	Fildbus, cc-link 以Double Word为单位输出.	
cfin	Fildbus, cc-link 输入以 float 单位读取.	
cfout	Fildbus, cc-link 以 float 单位输出.	
cin	Fildbus, cc-link 输入以Bit为单位读取.	
clog	保存包含时间信息的通信相关Log	
cnt	将通过输入端口进入的Pulse输入UP COUNT并保存	
continue	返回到重复语句的首部	
cout	Cclink以Bit单位输出.	
cthread	创建thread	
cwin	cclink 输入以Word为单位读取.	
cwout	Cclink以Word为单位输出.	
dec	设置减速时间的百分比 (%)	
din	从指定的Double Word单位输入中接收值	
dlay	迟延时间设置	
double	声明实数型变量	
dout	以Double Word为单位, 输出指定为Double Word的值.	
elif	执行条件判断	
else	执行条件判断	
end	结束	
for	执行Step重复数	

fos	到达轴线段目标点之前轨迹变更到下一个目标点.	
func	函数声明	
goto	执行语句分段	
here	读取当前周角度值变量	
if	执行条件判断	
in	1Bit为单位, 读取指定的Bit编号的ON(=1), OFF(=0) 值.	
initarm_clear	使用ARM FOLD 功能时保存Hand位置值	
initarm_read	使用ARM FOLD 功能时读取到达的Hand位置值	
initarm_write	使用ARM FOLD 功能时保存Hand位置值	
inpos	指定目标点到达	
int	整数型变数声明	
jmov	从当前位置PTP移动到目标点	
jntsync	设置PTP 同步模式	T1-X only
labl	指定分段点位置	
latch_clear	latch 环境初始化	
latch_init	latch 环境设置	
latch_pos	latch 位置	
latch_start	latch 开始	
latch_state	latch 状态	
latch_stop	latch 停止	
lmov	从当前位置线型插值移动到目标点	
lp_used	确认Local Point中是否保存值	
main	开始主 Program	
map_init	Mapping 相关内部变量初始化	T1-X only
map_read	Mapping 数据读取	
map_start	Mapping 开始	
map_stop	Mapping 停止	
mvr	对于整体距离的移动距离百分比值	
orgexec	执行Origin 功能	
out	1Bit为单位, 输出指定的Bit编号	
override	变更整体执行速度	
pin	以port为单位, 读取指定的输入Port值	
plup	设置PULL UP 动作用的Z轴位置值	

pos	声明位置型变量	
pout	以Port为单位，利用指定的Port输出指定值	
print	Log保存为文件	
print_read	读取通过PRINT函数保存的Log	
read_dist	读取累积距离及超过的Count	
error	返回最后的Error Code	
rerrcnt	返回当前发生的Alarm个数	
rerrcode	返回INDEX相应的Error Code	
rerrtext	返回INDEX相应的Error内容	
reset	Alarm Clear	
return	结束函数	
seterr	发生System emergency Alarm	
setlp	指定Local Point中保存Point值	
step	重复执行Block(增量)	
stop	停止Robot动作	
string	输入分隔符位置的字符串	
svof	Servo OFF	
svon	Servo ON	
tmr	系统定义变量 (Timer变量)	
to	Step重复执行	
vel	设置轴移动速度的百分比 (%)	
void	声明函数时无返回值	
while	执行条件重复执行	
win	Word为单位，读取指定的Word值。	
with	Robot动作中同时处理下一个执行语句	
wout	Word为单位，以指定的Word输出指定值。	
zcal_clear	解除设定值及初始化	
zcal_coef_gen	生成Z轴补偿多项式系数	
zcal_data_clr	初始化Z轴补偿函数生成中使用的位置及误差信息	
zcal_data_read	读取输入的位置及误差信息	
zcal_data_set	输入Z轴补偿函数生成中使用的位置及误差信息	
zcal_init	设置Z 补偿对象轴及参照的Hand轴	
zcal_opt	设置使用的附加轴	

zcal_read	返回Hand位置值对应的Z轴补偿值	
zcal_stop	停止Z轴补偿	
zcal_start	开始Z轴补偿	
zcal_state	确认Z轴补偿使用情况	

■ FUNC (Function) 项

- 黄色部分为按钮键编号.

GROUP	내용	비고
abs	取绝对值	
acos	反余弦函数	
asc	字符串第一个字母转换为字符代码	
asin	反正弦函数	
atan	反正切函数	
atan2	第2反正切函数	
bins	整数转换为二进制字符串	
chr	整数变为字符	
cos	余弦函数	
deg	弧度值转换为角度值	
dist	求两个位置变量之间的差	
exp	指数 e^x	
ftos	整数或实数转换为字符串	
htos	整数转换为16进制数的字符串	
ln	自然对数 $\log_e X$	
log	常用指数函数	
pow	指数函数	
rad	Angle转换为 radian 值	
max	最大值函数	
min	最小值函数	
reset	控制器Alarm解除	
rnd	四舍五入小数部实数值转换为整数值	
sin	正弦函数	
sleft	输出左侧部分字符串	
slen	返回字符串长度	
smid	从字符串指定位置提取指定长度字符串	
spos	返回字符串1 内字符串2所在的位置	
sqrt	平方根 $\sqrt{\quad}$	
sright	提取输入的字符串右侧部分	

strin	输入字符串到分隔符位置	
strout	字符串输出	
sval	字符串转换为数字	
tan	正切函数	

Robostar

■ OPER (Operator) 项

- 黄色部分为按钮键编号.

GROUP	내용	
+	二元运算符:加法,字符连接	
-	二元运算符:减法	
*	二元运算符:乘法	
/	二元运算符:除法	
!	逻辑运算符 : Negation	
%	二元运算符:取余	
&	比特运算符: AND	
	比特运算符: OR	
^	比特运算符: XOR ,指数	
→	比特运算符:NOT	
>	比较运算符: 大于	
<	比较运算符: 小于	
=	代入运算符	
,	单引号	
(	左括号,提高运算的有限顺序	
)	右括号, 提高运算的有限顺序	
[	左大括号	
]	右大括号	
“	双引号	
.	方位指针变量的各元素	
_	下划线文字	

2.1.4.2.2 CMD (Command)

- 提供Robot指令.
- Robot指令种类: 包含声明、位置控制、系统控制、分段、循环语句、程序执行、动作控制、Event、输入输出、通信、Task等.
- 下面通过例子学习下输入“while”语句的方法.

■ . 例子

Step 1.

选择Robot指令

```
<EDIT JOB>
STEP  INS
0000 func void main
* 0001 
0002 take 1
0003 while 1
0004 imov ip[1]
CMD FUNC OPER
```



- 使用Robot指令时按下“F1”键选择“CMD”.

Step 2.

Robot指令画面

```
<EDIT JOB>          DD
CMD INSTRUCT [1/8]
1: acc              2: bin
3: bout             4: cbin
5: cbout            6: cdin
7: cdout            8: cfin
```

Input #■

~

```
<EDIT JOB>          DD
CMD INSTRUCT [8/8]
1: wout
```

Input #■



- Teach Pendant 画面中没有要输入的指令时执行“页面移动”.
- 利用“PAGE UP/DOWN” 键可以切换画面
- 请参照‘2.1.4.2 预处理器项’的 ‘CMD 项’

Step 3.

While문 选择

```

<EDIT JOB>      DD
CMD INSTRUCT [7/8]
1: svon          2: tmr
3: to            4: vel
5: void          6: while
7: win          8: with

```

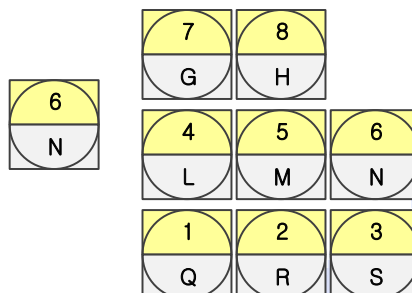
Input #■

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 while ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER

```

- 选择While语句时按下“6 While”键
- 选择指令时按下“数字键”



Step 5.

Edit 画面恢复及取消

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 while ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER

```

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 while
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER

```

ENTER

ESC

- “ENTER” 保存输入、恢复画面
- “ESC” 取消输入

2.1.4.2.3 FUNC (Function)

- 提供函数指令.
- 函数指令种类: 数学函数, 变换函数, 字符串函数
- 下面通过例子学习输入 “sval” 语句的方法.

■ 例子

Step 1.

选择Robot指令

```
<EDIT JOB>
STEP  INS
0000 func void main
* 0001 
0002 take 1
0003 while 1
0004 imov ip[1]
CMD  FUNC  OPER
```



- 使用Robot指令时按下“F2”键选择“FUNC”.

Step 2.

Robot指令画面

```
<EDIT JOB>          DD
FUNC INSTRUCT  [1/5]
1: abs          2: acos
3: asc          4: asin
5: atan         6: atan2
7: bins        8: chr
Input #■
```



- Teach Pendant 画面中没有要输入的指令时执行“页面移动”.

~

```
<EDIT JOB>          DD
CMD INSTRUCT  [5/5]
1: tan
Input #■
```



- 利用“PAGE UP/DOWN” 键可以切换画面
- 请参考‘2.1.4.2 预处理器项’的 ‘FUNC 项’

Step 3.

“sval” 选择

```

<EDIT JOB>      DD
CMD INSTRUCT [4/5]
1: slem          2: smid
3: spos          4: sqrt
5: sright        6: strin
7: strout        8: sval

```

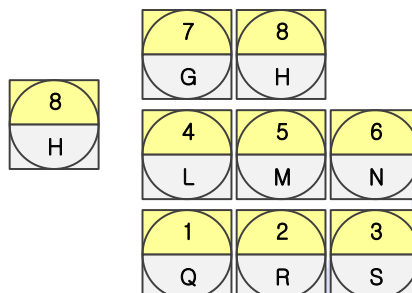
Input #■

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 sval ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER

```

- 选择“sval”时按下“8. sval”.
- 选择指令时按下“数字键”



Step 5.

Edit 画面恢复

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 sval ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER

```

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 sval
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER

```

ENTER

ESC

- “ENTER” 保存输入,画面恢复
- “ESC” 取消输入

2.1.4.2.4 OPER (Operation)

- 提供特殊符号.
- 特殊符号种类: 运算符
- 下面通过例子学习输入 “=” 等号的方法.

■ . 例子

Step 1.

选择Robot指令

```
<EDIT JOB>
STEP  INS
0000 func void main
* 0001 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER
```

F1

- 使用Robot指令时按下“ F3”键选择“OPER”.

Step 2.

Robot指令画面

```
<EDIT JOB>          DD
OPER INSTRUCT  [1/3]
1: +           2: -
3: *           4: /
5: !           6: %
7: &           8: |

input #■
```

```
<EDIT JOB>          DD
OPER INSTRUCT  [2/3]
1: ^           2: →
3: >           4: <
5: =           6: .
7: (           7: )

Input #■
```

```
<<EDIT JOB>         DD
OPER INSTRUCT  [3/3]
1: [           2: ]
3: ^           4: .

Input #■
```

Pg Up

Pg Dn

- Teach Pendant 画面中没有要输入的指令时执行“**页面移动**”
- 可以利用“PAGE UP/DOWN” 键进行画面切换
- 请参考‘2.1.4.2 预处理器项’的 ‘OPER 项’

Step 3.

“=”选择等号

```

<EDIT JOB>      DD
OPER INSTRUCT  [2/3]
1: ^            2: →
3: >           4: <
5: =           6: ;
7: (           7: )

```

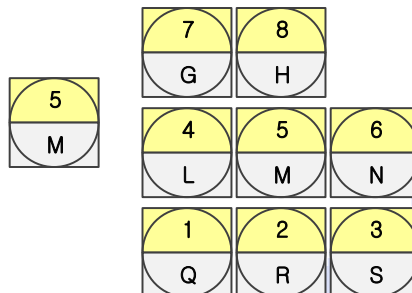
Input #■

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 = ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

- 选择“=” 时按下 “5. =”键
- 选择指令时按下“数字键”



Step 5.

Edit 画面恢复

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 = ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 =
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

ENTER

ESC

- “ENTER” 保存输入
- “ESC” 取消输入

2.1.4.3 输入JOB 内容

■ . OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

编辑画面

Step 1

```

<JOB MANU>
* 0.AA      8    STEP
  1.DD      5    STEP
  2.RF      7    STEP
  3.WW     12    STEP
  4.ARE    100   STEP
  5.QE      8    STEP
EDIT NEW    >
  
```

F1

- 输入JOB内容时按下“F1”键选择“EDIT”。

JOB内容编辑

Step 2

```

<EDIT JOB>
STEP INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY MOVE DEL
  
```

ENTER

- 在选择的Line创建新JOB 内容时按下“ENTER”键
- 新建的JOB Line显示为“Cursor”。
- 现有 JOB 内容内容向下移一行。
- 请编辑JOB 内容!

```

<EDIT JOB>
STEP INS
0000 func void main
* 0001 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD FUNC OPER
  
```

■ . 编辑顺序

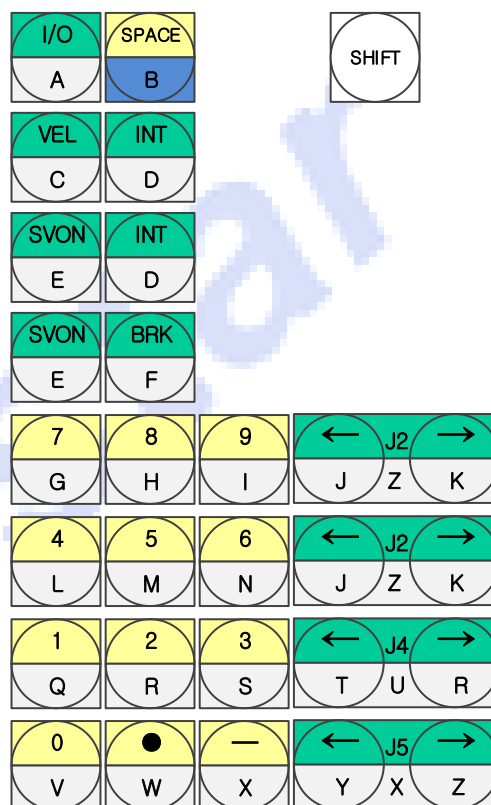
Step 1.

输入JOB 内容

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER
  
```

- 在选择的Line中按下数字键和文字键输入内容



Step 2.

输入内容

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER
  
```



- 试试直接输入aa=1!
- 首先要按下“SHIFT”按钮才能输入文字键。接下来按两次“A”,A”。

Step 3.

输入特殊符号

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```



- 输入等号“=”时按下“F3”键

```

<EDIT JOB>          DD
OPER INSTRUCT  [1/3]
1: +           2: -
3: *           4: /
5: !           6: %
7: &           8: |

```

input #■



- Teach Pendant 画面中无“等号”符号时进行“页面移动”。

```

<EDIT JOB>          DD
OPER INSTRUCT  [2/3]
1: ^           2: →
3: >           4: <
5: =           6: ,
7: (           8: )

```

Input #■



- 可以利用“PAGE UP/DOWN”键切换画面

等号 5번选择

```

<EDIT JOB>          DD
OPER INSTRUCT  [2/3]
1: ^           2: →
3: >           4: <
5: =           6: ,
7: (           8: )

```

Input #5■



- Teach Pendant中有等号时选择等号。
- 选择等号时按下“5. =”。

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

Step 4.

输入“1”

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```



- 首先解除SHIFT”功能。 接下来按下“1”号。



按下“ENTER”键恢复到EDIT 画面。



```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

Step 5.

输入“1”

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1 ■
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```



- 首先解除” SHIFT”功能。 接下来按下“1”号。



按下“ENTER”键恢复到EDIT 画面。



```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

Step 6.

JOB文件保存及取消

```

<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1
0002 take 1
0003 while 1
0004 jmov ip[1]
COPY  MOVE  DEL  JUMP

```



- 变更值之后按下“ESC” 键时画面中暗处 update 信息

```
<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1
0002 take 1
0003 while 1
0004 jmov ip[1]
Update ? (ENT/ESC)
```

ENTER

- 保存时按下“ENTER”键
- 取消时按下“ESC”键

```
<EDIT JOB>
STEP  INS
0000 func void main
* 0001 aa=1
0002 take 1
0003 while 1
0004 jmov ip[1]
save complete
```

ENTER

- 恢复到Edit 画面时按下“ENTER”键

```
<JOB MANU>
* 0.AA      8    STEP
  1.DD      5    STEP
  2.RF      7    STEP
  3.WW     12    STEP
  4.ARE    100   STEP
  5.QE      8    STEP
EDIT NEW      NEXT
```

- 恢复到JOB 文件选择画面

2.1.4.4 JOB Line插入

- 新建JOB 文件内Line后编辑内容.
- 新建的JOB Line从选择的Line开始.
- 继续按下“ENTER”键可以继续创建Line.

■ . OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

编辑画面

Step 1

```

<JOB MANU>
* 0.AA      8   STEP
  1.DD      5   STEP
  2.RF      7   STEP
  3.WW     12   STEP
  4.ARE    100  STEP
  5.QE      8   STEP
EDIT NEW      NEXT
        
```

F1

- 插入JOB Line时按下“F1”键选择“EDIT”.

```

<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
        
```

Step 2

JOB内容编辑

```

<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
        
```

ENTER

- 在选择的Line中创建新 JOB 内容时按下“ENTER”键
- 新创建的JOB Line显示为“Cursor”.
- 现有JOB内容向下移动一行.
- 请编辑JOB内容！

```

<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER
        
```


■ . 插入说明

Step 1

JOB 内容插入

```
<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
```

```
<EDIT JOB>
STEP  INS
0000 func void main
* 0001 
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER
```

ENTER

- 在选择的Line中创建新 JOB 内容时按下“ENTER”键

- 新创建的JOB Line显示为“Cursor”.
- 现有JOB内容向下移动一行.

- 请编辑JOB内容.

■ Cursor位置移动

- ① 请参考[2.1.4.1 JOB Line移动](#).

■ 内容输入方法

- ① 请参考[2.1.4.2 输入预处理器指令](#)
- ② 请参考[2.1.4.3 输入JOB 内容](#).

■ 保存及取消方法

- ① 请参考[2.1.4.3 输入JOB 内容](#) 中的JOB文件保存及取消方法.

2.1.4.5 JOB 内容修正

- 修正现有JOB文件内容.

■ . OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

Step 1

编辑画面

```

<JOB MANU>
* 0.AA      8    STEP
  1.DD      5    STEP
  2.RF      7    STEP
  3.WW     12    STEP
  4.ARE    100   STEP
  5.QE      8    STEP
EDIT NEW    >
  
```

Step 2

JOB内容编辑

```

<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL
  
```

ENTER

```

<EDIT JOB>
STEP  INS      i
0000 func void main
* 0001 
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER
  
```

- 修正JOB内容时按下“F1”键选择“EDIT”。
- 在选择的Line中创建新 JOB 内容时按下“ENTER”键
- 新创建的JOB Line显示为“Cursor”。
- 现有JOB内容向下移动一行。
- 请编辑JOB内容！

■ . 修正说明

Step 1

JOB 内容修正

```

<EDIT JOB>
STEP  INS          i
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL  JUMP

```



- 修正JOB 内容时按下“W”键

```

<EDIT JOB>
STEP  INS          e
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL  JUMP

```

- Teach Pendant 右上方显示“e”

Step 2

JOB파일 편집

```

<EDIT JOB>
STEP  INS          e
0000 func void main
* 0001 take 1
0002 while 1
0003 jmov ip[0]
0004 jmov ip[1]
COPY  MOVE  DEL  JUMP

```



- 新创建JOB内容时按下“ENTER”键

```

<EDIT JOB>
STEP  INS          e
0000 func void main
* 0001 take 1
0002 take 1
0003 while 1
0004 jmov ip[1]
CMD  FUNC  OPER

```

- 修正位置显示为“Cursor”.

■ Cursor位置移动

- ① 请参考[2.1.4.1 JOB Line移动](#)

■ 内容输入方法

- ① 请参考[2.1.4.2 输入预处理器指令](#)
- ② 请参考[2.1.4.3 输入JOB 内容](#).

■ 保存及取消方法

- ② 请参考[2.1.4.3 输入JOB 内容](#) 中的JOB文件保存及取消方法.

2.1.5 EDIT

- 可以查看选择的JOB文件内容，编辑、复制、移动、删除内容。详细项如下表。

■ 详细项

GROUP	内容	备注
[F1] COPY	在指定的Block Line复制粘贴	
[F2] MOVE	在指定的Block Line剪切粘贴	
[F3] DEL	删除指定的Block Line	

■ OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [F1] EDIT

Step 1.

页面移动

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
EDIT NEW		>

<JOB MANU>		
* 0.TT	8	STEP
1.MM	5	STEP
2.FF	7	STEP
EDIT NEW		>



- 可以利用“PAGE UP/DOWN”键切换画面

Step 2.

光标移动

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
5.QE	8	STEP
EDIT NEW		>



- 选择要编辑的文件
- 按下“UP/DOWN”键选择要编辑的JOB文件
- 可以确认选择的JOB文件最前面显示“*”符号

```

<JOB MANU>
0.AA      8    STEP
* 1.DD     5    STEP
- 2.RF     7    STEP
3.WW     12    STEP
4.ARE    100    STEP
5.QE      8    STEP
EDIT  NEW    >

```

Step 3

JOB文件编辑

```

<JOB MANU>
0.AA      8    STEP
* 1.DD     5    STEP
- 2.RF     7    STEP
3.WW     12    STEP
4.ARE    100    STEP
5.QE      8    STEP
EDIT  NEW    >

```



- 对现有内容编辑、复制、删除等操作时
按下“F1”键选择“EDIT”。

Step 4

编辑画面

```

<EDIT JOB>
STEP  INS
* 0000 func void main
- 0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
< COPY  MOVE  DEL

```

- 显示编辑画面。

2.1.5.1 COPY

- 指定要复制的模块，选择复制的Line，执行粘贴。
- Block个数可以无限制指定。

Step 1.

页面移动

```
<EDIT JOB>          DD
STEP  INS           i
* 0000 func void main
0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL
```

```
<EDIT JOB>          DD
STEP  INS           i
* 0005 end
0006 func void robot
0007 take 1
0008 while 1
0009 jmov ip[1]
COPY MOVE DEL
```



- Teach Pendant 画面中无复制的内容时进行“页面移动”。

- 可以利用“PAGE UP/DOWN” 键切换画面

Step 2.

光标移动

```
<EDIT JOB>          DD
STEP  INS           i
* 0000 func void main
0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL
```

```
<EDIT JOB>          DD
STEP  INS           i
0000 func void main
* 0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
< COPY MOVE DEL
```



- 选择“开始”的Block时用“光标移动”键移动..

- 按下“UP/DOWN” 键选择要编辑的JOB 文件

- 可以确认选择的Line最前面显示 “*” 符号

Step 3.

Block 指定

```

<EDIT JOB>          DD
STEP  INS            i
0000 func void main
* 0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```



- 指定要复制的Block时先按下“SHIFT”键后再按下“SPACE”键

```

<EDIT JOB>          DD
STEP  INS            i
0000 func void main
* 0001< release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```

- 选择的Line中显示“Block”.

```

<EDIT JOB>          DD
STEP  INS            i
0000 func void main
* 0001< release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```



- 按下“UP/DOWN”键指定要复制的内容为Block.

```

<EDIT JOB>          DD
STEP  INS            i
0000 func void main
0001< release
* 0002< take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```

Step 4.

Block 固定

```

<EDIT JOB>          DD
STEP  INS            i
0000 func void main
0001< release
* 0002< take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```



- 指定Block后再次按下“SPACE”键固定Block Line.

Step 5.

赋值为止选择方法 1

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001 { release
* 0002 { take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```



- 按下“PAGE UP/DOWN”键选择移动到复制的页面位置。



- 页面移动可以通过“*”符号确认。

```

<EDIT JOB>          DD
STEP  INS            i
* 0005 end
0006  func void robot
0007  take 1
0008  while 1
0009  jmov ip[1]
COPY  MOVE  DEL

```

赋值为止选择方法2

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001 { release
* 0002 { take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```



- 按下“UP/DOWN”键，Cursor移动并选择要复制的位置。



- Cursor移动通过“*”符号确认。

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001 { release
0002 { take 1
* 0003 while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```

赋值为止选择方法3

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001 { release
* 0002 { take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```



- 直接移动到输入的Line，选择要复制的位置。
- 移动到输入的Line时按下“JUMP”键。

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001 { release
* 0002 { take 1
0003  while 1
0004  jmov ip[0]
STEP #=9

```

- 输入要移动的Line编号9.

7	8	9
G	H	I
4	5	6
L	M	N


```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009 jmov ip[1]
COPY MOVE DEL

```

1 Q	2 R	3 S
0 V	● W	— X

Step 6.

执行复制.

```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009 jmov ip[1]
COPY MOVE DEL

```



- 复制时按下“F1”键选择“COPY”.

```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009 jmov ip[1]
Copy complete

```



- 确认信息后画面恢复时按下 “ENTER” 键

```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009 release
COPY MOVE DEL

```

- 显示复制的内容‘起始位置’.

2.1.5.2 MOVE

- 指定要剪切的Block，再把剪切的Block内容粘贴.
- Block个数可以无限制指定.

Step 1.

页面移动

```
<EDIT JOB>          DD
STEP  INS            i
* 0000 func void m-ain
0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL
```

```
<EDIT JOB>          DD
STEP  INS            i
* 0005 end
0006 func void robot
0007 take 1
0008 while 1
0009 jmov ip[1]
COPY MOVE DEL
```



- Teach Pendant 画面中无剪切内容时执行“页面移动”.

- 可以利用“PAGE UP/DOWN” 键切换画面

Step 2.

光标移动

```
<EDIT JOB>          DD
STEP  INS            i
* 0000 func void main
0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL
```

```
<EDIT JOB>          DD
STEP  INS            i
0000 func void main
* 0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL
```



- 选择“开始”的Block Line时按下“Cursor UP / DOWN”键移动..

- 按下“UP/DOWN” 键选择要编辑的JOB 文件

- 可以确认选择的Line最前面显示 “*” 符号

Step 3.

Block 指定

```

<EDIT JOB>          DD
STEP  INS           i
0000 func void main
* 0001 release
- 0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```



- 指定剪切的Block时先按下 “SHIFT” 键后再按下 “SPACE”



```

<EDIT JOB>          DD
STEP  INS           i
0000 func void main
* 0001< release
- 0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```

- 显示选择的Line的Block("<").

```

<EDIT JOB>          DD
STEP  INS           i
0000 func void main
* 0001< release
- 0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```



- 按下“UP/DOWN”键指定要复制的内容为Block.



```

<EDIT JOB>          DD
STEP  INS           i
0000 func void main
0001< release
* 0002< take 1
- 0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```

Step 4.

Block固定

```

<EDIT JOB>          DD
STEP  INS           i
0000 func void main
0001< release
* 0002< take 1
- 0003 while 1
0004 jmov ip[0]
COPY MOVE DEL

```



- 指定Block后再次按下“SPACE”键固定 Block Line.

Step 5.

粘贴位置选择方法1

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001< release
* 0002< take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```



- 按下“PAGE UP/DOWN”选择要复制的页面位置。



- 页面移动可以确认 “*” 符号。

```

<EDIT JOB>          DD
STEP  INS            i
* 0005 end
0006  func void robot
0007  take 1
0008  while 1
0009  jmov ip[1]
COPY  MOVE  DEL

```

붙여넣기 위치 선택 방법2

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001< release
* 0002< take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```



- 按下“UP/DOWN”键通过光标移动选择复制的位置。



- Cursor移动可以确认 “*” 符号。

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001< release
0002< take 1
* 0003< while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```

粘贴位置选择方法3

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001< release
* 0002< take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL

```



- 直接移动到输入的Line选择复制的位置。
- 移动到输入的Line时按下“JUMP”键。

```

<EDIT JOB>          DD
STEP  INS            i
0000  func void main
0001< release
* 0002< take 1
0003  while 1
0004  jmov ip[0]
STEP #=9

```

```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009  jmov ip[1]
COPY  MOVE  DEL

```

- 输入移动的Line编号9.

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

Step 6.

剪切后粘贴.

```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009  jmov ip[1]
COPY  MOVE  DEL

```



- 剪切结束后再粘贴时按下“F2”键选择“MOVE”.

```

<EDIT JOB>          DD
STEP  INS            i
0005  end
0006  func void robot
0007  take 1
0008  while 1
* 0009  jmov ip[1]
Copy complete

```



- 确认信息后恢复画面按下“ENTER”键

```

<EDIT JOB>          DD
STEP  INS            i
0007  take 1
0008  while 1
0009  release
0010  take 1
* 0011  end
COPY  MOVE  DEL  JUMP

```

- 粘贴内容最后一行显示“*” 符号.

2.1.5.3 DEL

- 指定要删除的Block并删除..
- Block个数可以无限制指定.

Step 1.

页面移动

```

<EDIT JOB>          DD
STEP  INS           i
* 0000 func void main
  0001 release
  0002 take 1
  0003 while 1
  0004 jmov ip[0]
COPY MOVE DEL JUMP
  
```

```

<EDIT JOB>          DD
STEP  INS           i
* 0005 end
  0006 func void robot
  0007 take 1
  0008 while 1
  0009 jmov ip[1]
COPY MOVE DEL JUMP
  
```



- Teach Pendant 画面中无删除内容时执行“页面移动”.

- 可以利用“PAGE UP/DOWN” 键切换画面

Step 2.

光标移动

```

<EDIT JOB>          DD
STEP  INS           i
* 0000 func void main
  0001 release
  0002 take 1
  0003 while 1
  0004 jmov ip[0]
COPY MOVE DEL JUMP
  
```

```

<EDIT JOB>          DD
STEP  INS           i
0000 func void main
* 0001 release
  0002 take 1
  0003 while 1
  0004 jmov ip[0]
COPY MOVE DEL JUMP
  
```



- 选择“开始”的Block Line时按下“Curs or UP/DOWN 移动”键移动..

- 按下“UP/DOWN” 键选择要编辑的JOB 文件

- 可以确认选择的Line最前面显示 “*” 符号

Step 3.

Block 指定

```

EDIT JOB>          DD
STEP  INS          i
0000 func void main
* 0001 release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL JUMP

```



- 选择要删除的Block时先按下 “SHIFT” 键后再按下 “SPACE” 键



- 开始位置显示“*”符号.

```

EDIT JOB>          DD
STEP  INS          i
0000 func void main
* 0001< release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL JUMP

```

```

EDIT JOB>          DD
STEP  INS          i
0000 func void main
* 0001< release
0002 take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL JUMP

```



- 按下“UP/DOWN”键指定要删除的内容作为Block.



```

EDIT JOB>          DD
STEP  INS          i
0000 func void main
0001< release
* 0002< take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL JUMP

```

Step 4.

Block 固定

```

EDIT JOB>          DD
STEP  INS          i
0000 func void main
0001< release
* 0002< take 1
0003 while 1
0004 jmov ip[0]
COPY MOVE DEL JUMP

```



- 指定Block后再次按下“SPACE”键 固 定 Block Line.

Step 5.

执行删除.

```

EDIT JOB>          DD
STEP  INS          i
0000  func void main
0001 { release
* 0002 { take 1
0003  while 1
0004  jmov ip[0]
COPY  MOVE  DEL  JUMP

```



- 按下“F3”键选择要删除的内容后选择“DEL”.

```

EDIT JOB>          DD
STEP  INS          i
0000  func void main
0001 { release
* 0002 { take 1
0003  while 1
0004  jmov ip[0]
Delete ok? (ENT/ESC)

```



- 删除时按下“ENTER”



- 取消删除时按下“ESC”

```

EDIT JOB>          DD
STEP  INS          i
0000  func void main
* 0001  while 1
0002  jmov ip[0]
0005  end
0006  func void robot
COPY  MOVE  DEL  JUMP

```

- 删除开始的位置显示“*”符号.

2.1.6 NEW

- 可以创建新JOB 文件并编辑JOB内容.
- 文件名一定要以英文字母开头，由英文字母或数字组成的5位字符以内.

■ OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [F2] NEW

Step 1.

创建新文件

<JOB MANU>		
* 0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
3.WW	12	STEP
4.ARE	100	STEP
50.QE	8	STEP
EDIT	NEW	>



- 创建新文件时按下“F2”键选择“NEW”.

Step 2.

创建新文件

<NEW JOB>	
FILE NAME =	NEW1



- 首先按下“SHIFT”按钮才能输入文字键.
接下来按顺序按下“N”, “E”, “W”键.
- 解除已选择的SHIFT”功能，按下“1”键.

EDIT JOB>	NEW1
STEP INS	1
* 0000	
0001	
0002	
0005	
0006	
COPY MOVE DEL	



■ 内容输入方法

- ① 请参考[2.2.4 2 预处理器指令输入方法](#)
- ② 请参考[2.2.4.1 JOB Line移动方法](#) JOB内容输入.

■ 保存及取消方法

- ① 请参考[2.2.4.1 JOB Line移动方法](#) JOB文件保存及取消方法.

2.1.7 COPY

- 复制现有JOB 文件保存为其它名称.
- 文件名一定要以英文字母开头，由英文字母或数字组成的5位字符以内.

■ OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [JUMP] 按钮 → [F1] COPY

Step 1.

页面移动方法

(JOB MANU)
 *0.AA 8 STEP
 1. DD 5 STEP
 2. RF 7 STEP
 3. WW 12 STEP
 4. ARE 100 STEP
 5. QQ 8 STEP
 (COPY REN DEL

(JOB MANU)
 *6.TT 8 STEP
 7. MM 5 STEP
 8. FF 7 STEP
 (COPY REN DEL



- Teach Pendant 画面中无复制的内容时进行“页面移动”.



- 可以利用“PAGE UP/DOWN” 键切换画面

Step 2.

光标移动方法

(JOB MANU)
 *0.AA 8 STEP
 1. DD 5 STEP
 2. RF 7 STEP
 (COPY REN DEL

(COPY REN DEL

(JOB MANU)
 0.AA 8 STEP
 *1.DD 5 STEP
 2. RF 7 STEP
 (COPY REN DEL

(COPY REN DEL



- 选择要编辑的文件.
- 按下“UP/DOWN” 键选择要编辑的JOB 文件



- 可以确认选择的JOB文件最前面显示“*” 符号

Step 3.

创建新文件

<JOB MANU>
 * 0.AA 8 STEP
 1.DD 5 STEP
 2.RF 7 STEP

< COPY REN DEL

<COPY JOB>
 JOB NAME = DD

New job name =

- 复制现有文件为其它文件时按下“F1”键选择“COPY”。



- 显示可以输入复制后的文件名的窗口。

Step 3.

输入复制后的文件名

<COPY JOB>
 JOB NAME = DD

New job name = COPY

<COPY JOB>
 JOB NAME = DD

Copy Complete

- 首先需要按下“SHIFT”按钮才能输入文字键。接下来按顺序输入“C”, “O”, “P”, “Y”键



- 输入文件名后按下“ENTER”完成。



Step 3.

输入复制后的文件名

<COPY JOB>
 JOB NAME = DD

Copy Complete

- 恢复到JOB MANU时按下“ENTER”键



<JOB MANU>

0.AA	8	STEP
1.DD	5	STEP
2.RF	7	STEP
<u>* 3.COPY</u>	<u>5</u>	<u>STEP</u>

2.1.8 REN (Rename)

- 现有JOB 文件更名.
- 文件名一定要以英文字母开头，由英文字母或数字组成的5位字符以内.

■ OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [JUNP] 按钮 → [F2] REN

Step 1.

页面移动方法

(JOB MANU)
 * 0.AA 8 STEP
 1.DD 5 STEP
 2.RF 7 STEP
 3.WW 12 STEP
 4.RE 100 STEP
 5QE 8 STEP
 (COPY REN DEL

(JOB MANU)
 * 6.TT 8 STEP
 7.MM 5 STEP
 8.FF 7 STEP
 (COPY REN DEL



- Teach Pendant 画面中无要变更名称的内容时按下“页面移动”.
- 可以利用“PAGE UP/DOWN” 键切换画面

Step 2.

光标移动方法

(JOB MANU)
 * 0.AA 8 STEP
 1.DD 5 STEP
 2.RF 7 STEP
 (COPY REN DEL

(JOB MANU)
 0.AA 8 STEP
 * 1.DD 5 STEP
 2.RF 7 STEP
 (COPY REN DEL



- 选择要编辑的文件.
- 按下“UP/DOWN” 键选择要编辑的JOB 文件
- 可以确认选择的JOB文件最前面显示“*” 符号

Step 3.

文件名变更

(JOB MANU)
0.AA 8 STEP
~~* 1.DD 5 STEP~~
2.RF 7 STEP

< COPY **REN** DEL

(REN JOB)
JOB NAME = DD

New job name =



- 现有文件更名时按下“F2”选择“REN”.
- 显示可以输入变更的文件名的窗口.

Step 3.

输入要变更的文件名

(REN JOB)
JOB NAME = DD

New job name = COPY

(REN JOB)
JOB NAME = DD

Rename Complete



- 首先需要按下“SHIFT”按钮才能输入文字键. 接下来按顺序输入“R”, “E”, “N”
- 输入文件名后按下“ENTER”完成

Step 3.

输入要变更的文件名

< REN JOB>
JOB NAME = DD

Rename Complete

<JOB MANU>
0.AA 8 STEP
* 1.REN 5 STEP
2.RF 7 STEP

< COPY REN DEL

ENTER

- 恢复到JOB MANU时按下“ENTER”键

2.1.9 DEL

- 可以删除选择的 JOB 文件.

■ OPEN顺序

- Top Screen → 1. Edit → 1. JOB → [JUMP]按钮 → [F3] DEL

Step 1.

页面移动方法

<JOB MANU>
 *0.AA 8 STEP
 1.DD 5 STEP
 2.RF 7 STEP
 3.WW 12 STEP
 4.RE 100 STEP
 5.QE 8 STEP
 < COPY REN DEL

<JOB MANU>
 *6.TT 8 STEP
 7.MM 5 STEP
 8.FF 7 STEP
 < COPY REN DEL



- Teach Pendant 画面中无要删除的文件时执行“页面移动”.



- 可以利用“PAGE UP/DOWN”键切换画面

Step 2.

光标移动方法

<JOB MANU>
 *0.AA 8 STEP
 1.DD 5 STEP
 2.RF 7 STEP
 < COPY REN DEL

<JOB MANU>
 0.AA 8 STEP
 *1.DD 5 STEP
 2.RF 7 STEP
 < COPY REN DEL



- 选择要删除的文件.
- 按下“UP/DOWN”键选择要删除的JOB文件



- 可以确认选择的JOB文件最前面显示“*”符号

Step 3.

删除文件

<JOB MANU>
0.AA 8 STEP
* 1.DD 5 STEP
2.RF 7 STEP

< COPY REN **DEL**

<DEL JOB>
JOB NAME = DD

Delete ok? (ENT/ESC)



- 删除现有文件时按下“F3”键
- 显示输入要删除的文件的窗口.

Step 4.

输入要删除的文件名

< DEL JOB>
JOB NAME = DD

Delete ok? (ENT/ESC)

<JOB MANU>
0.AA 8 STEP
*** 1.RF 7 STEP**

< COPY REN DEL



- 确定删除时按下“ENTER”键
- 取消删除时按下“ESC”键

2.2 VAR (Variable)

- ✓ GLOBAL 变量具有程序执行停止或电源切断时维持其原有值的特性。此外，根据使用用途选择整数型、实数型使用，可以通过画面变更数据或确认当前的数值。所有文件可以引用，程序结束后也保存内容，这与程序内的全局变量不同。

2.2.1 概要

- INTEGER设置整数型数据时，FLOAT设置实数型数据时使用。
- VARIABLE包含整数型和实数型，各分配1,000个。
- 输入变更的值后按下 [ENTER]键自动保存到内存。
- 编辑的变量数据的最终保存脱离VARIABLE模式时自动保存。

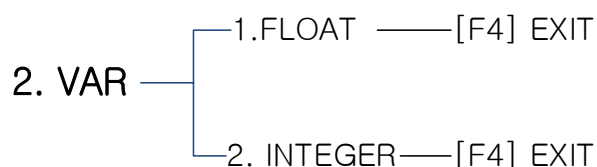
■ 2. VARIABLE种类

	GROUP	内容	备注
2. VAR	1. FLOAT	实数型变量	
	2. INTEGER	整数型变量	

2.2.2 VARIABLE模式流程图

- VARIABLE大致分为两个Group.
 - FLOAT：可以确认或编辑实数型变量值。
 - INTEGER: 可以确认或编辑整数型变量值。
 - FLOAT 输入值范围: -999999~999999
 - INTEGER 输入值范围：~32766~32767

■ Variable Menu Tree



2.2.3 VARIABLE Group OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version:1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. INFO 4. I/O
5. ORIGIN

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

EDIT 选择

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择“1. EDIT”

POINT 选择

<EDIT MENU>
1. JOB 2. VAR
3. PARA

input : #



- 选择“2. VAR”

INTEGER 选择

<EDIT VAR>
1. INT 2. FLOAT

input : #



- 编辑或确认INTEGE时选择 “1. INT”.

FLOAT 选择

<EDIT VAR>
1. INT

2. FLOAT

input : #



- 编辑或确认FLOAT时选择“2. FLOAT”.

2.2.4 VARIABLE 设定

- 输入方法与 '1. INTEGER' , '2. FLOAT' 相同.
- 通过例子说明输入变量值的方法.
 - Ex) F10=50 ← 输入实数型变量

Step 1.

移动到PARAMETER项

(EDIT VAR)
 1. INT **2. FLOAT**

 input :

 (FLOAT EDIT)
 F0 0
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0

 EXIT



- 选择“2. FLOAT”

- 确认实数型变量画面

Step 2.

移动到Cursor位置

(FLOAT EDIT)
F0 0
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0

 EXIT

(FLOAT EDIT)
 F0 0
F1 0
 F2 0
 F3 0
 F4 0
 F5 0

 EXIT



- 按下方向键移动到要编辑的值
- 通过Cursor位置可以确定移动结果

变更页码

〈FLOAT EDIT〉

F6	0
F7	0
F8	0
F9	0
F10	0
F11	0

EXIT



- 可以利用“PAGE UP/DOWN” 键切换画面



Step 3.

变量值变更

〈FLOAT EDIT〉

F6	0
F7	0
F8	0
F9	0
F10	5
F11	0

EXIT

- 按下要变更的值的数字键输入

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

〈FLOAT EDIT〉

F6	0
F7	0
F8	0
F9	0
F10	50
F11	0

EXIT



- 按下“ENTER” 键反应到画面中

Step 4.

删除变量值

〈FLOAT EDIT〉

F6	0
F7	0
F8	0
F9	0
F10	5.123
F11	0

EXIT



- 错误输入变量值时按下“DEL” 按顺序删除输入的值。

〈FLOAT EDIT〉

F6	0
F7	0
F8	0
F9	0
F10	5.12
F11	0

EXIT

입력값 범위 초과시

<FLOAT EDIT>

F6 0

F7 0

F8 0

F9 0

F10 1000000 ■

F11 0

JUMP

EXIT

<FLOAT EDIT>

F6 0

F7 0

F8 0

F9 0

F10 *

F11 0

JUMP

EXIT

- 实数型输入范围: -999999~999999

- 输入值超出范围时画面中显示“*” 符号.

2.2.5 JUMP (直接移动到编号)

- 直接移动到任意变量编号.
- 直接移动方法与 '1. INTEGER' , '2. FLOAT' 相同.

■ 移动方法

Step 1.

FLOAT画面

<INTEGER EDIT>

I0	0
I1	0
I2	0
I3	0
I4	0
I5	0

EXIT



- 移动到任意变量编号时按下“JUMP” 键

<INTEGER EDIT>

I0	0
I1	0
I2	0
I3	0
I4	0
I5	0

integer Num =

<INTEGER EDIT>

I0	0
I1	0
I2	0
I3	0
I4	0
I5	0

integer Num =50



- 输入要移动的编号: 50

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

- 完成输入后按下“ENTER” 键

<INTEGER EDIT>

I50	0
I51	0
I52	0
I53	0
I54	0
I55	0

EXIT

- 确认JUMP 移动画面.

2.2.6 INTEGER

- 监控、编辑GLOBAL整数型变量，详细项如下表.
- 输入范围：~32766~32767

■ 详细项

GROUP	内容	备注
[F4] EXIT	保存后退出	

■ OPEN顺序

- Top Screen → 1. Edit → 2. VAR → 1. INT

Step 1.

INTEGER 选择

<EDIT VAR>
 1. INT 2. FLOAT
 input #



- 选择“1. INT”

변수화면 확인

<INTEGER EDIT>
 I0 0
 I1 0
 I2 0
 I3 0
 I4 0
 I5 0
 EXIT

- 确认整数型变量画面

■ 设定方法

- ① 请参考[2.2.4 VARIABLE 设定](#).

■ 编号直接移动

- ① 请参考 [2.2.5 JUMP \(编号直接移动\)](#).

2.2.6.1 EXIT

- 退出变量显示画面.
- 退出变量显示画面时变量值保存到文件.
- 退出的方法有两种.

按下 “[F4] EXIT” 键退出的方法

按下 “ESC” 键退出的方法

■ 退出方法

Step 1.

INT 画面

<INTEGER EDIT>	
I0	0
I1	0
I2	0
I3	0
I4	0
I5	0
EXIT	

<EDIT VAR>	
1. INT	2. FLOAT
input #	



- 退出INT变量显示画面时按下 “F4” 键选择 “EXIT”.

- 确认INT 变量显示之前画面

2.2.7 FLOAT

- GLOBAL实数型变量监控及编辑详细项如下表.
- 输入范围：-999999~999999

■ 详细项

GROUP	内容	备注
[F4] EXIT	保存后退出	

■ OPEN顺序

- Top Screen → 1. Edit → 2. POINT → 3. VAR → 1. FLOAT

Step 1.

FLOAT 选择

<EDIT VAR>

1. INT

2. FLOAT

input #



- 选择2. FLOAT

确认变量画面

<FLOAT EDIT>

F0

0

F1

0

F2

0

F3

0

F4

0

F5

0

EXIT

- 确认实数型变量画面

■ 设定方法

- ① 请参考[2.2.4 VARIABLE 设定](#).

■ 编号直接移动

- ① 请参考 [2.2.5 JUMP \(编号直接移动\)](#).

2.2.7.1 EXIT

- 退出变量显示画面.
- 退出变量显示画面时变量值保存到文件.
- 退出的方法有两种.

按下 “[F4] EXIT” 键退出的方法

按下 “ESC” 键退出的方法

■ 退出方法

Step 1.

FLOAT 画面

Diagram illustrating the FLOAT EDIT screen and the EDIT VAR screen.

The FLOAT EDIT screen displays the following variables and values:

Variable	Value
F0	0
F1	0
F2	0
F3	0
F4	0
F5	0

The EXIT button is located at the bottom right of the FLOAT EDIT screen.

The EDIT VAR screen displays the following options:

Option	Value
1. INT	
2. FLOAT	

The input field is labeled "input #".



- 退出FLOAT 变量显示画面时按下 “F4” 键选择 “EXIT”.

- 确认FLOAT变量显示之前画面

2.3 参数(PARAMETER)

- ✓ 参数设定时非常重要的部分，Robot运行中发生的Error的一半以上原因都来自于参数值的错误设定。
- ✓ 参数的基本设定值请参考粘贴在Robot的标签内容。
- ✓ 参数的基本设定在产品出厂时设定，但在实际现场中安装并运行Robot时可能根据周边环境不得不修改参数值。此时请与本公司客服联系咨询。

2.3.1 概要

- 参数的编辑模式是设置Robot形态及使用环境的模式。
- Robot(机械部 + 控制器)运行前务必确认设定值，有必要时需要重新设置。
 - Robot的机械性、电气性规格相关数据
 - 原点恢复方法，特殊功能等Robot的基本动作方法
 - PC和数据通信规则
- Robot出厂时已设置基本参数。
- 完成设置时为避免使用上的错误等变更，请务必记录或备份参数设定。

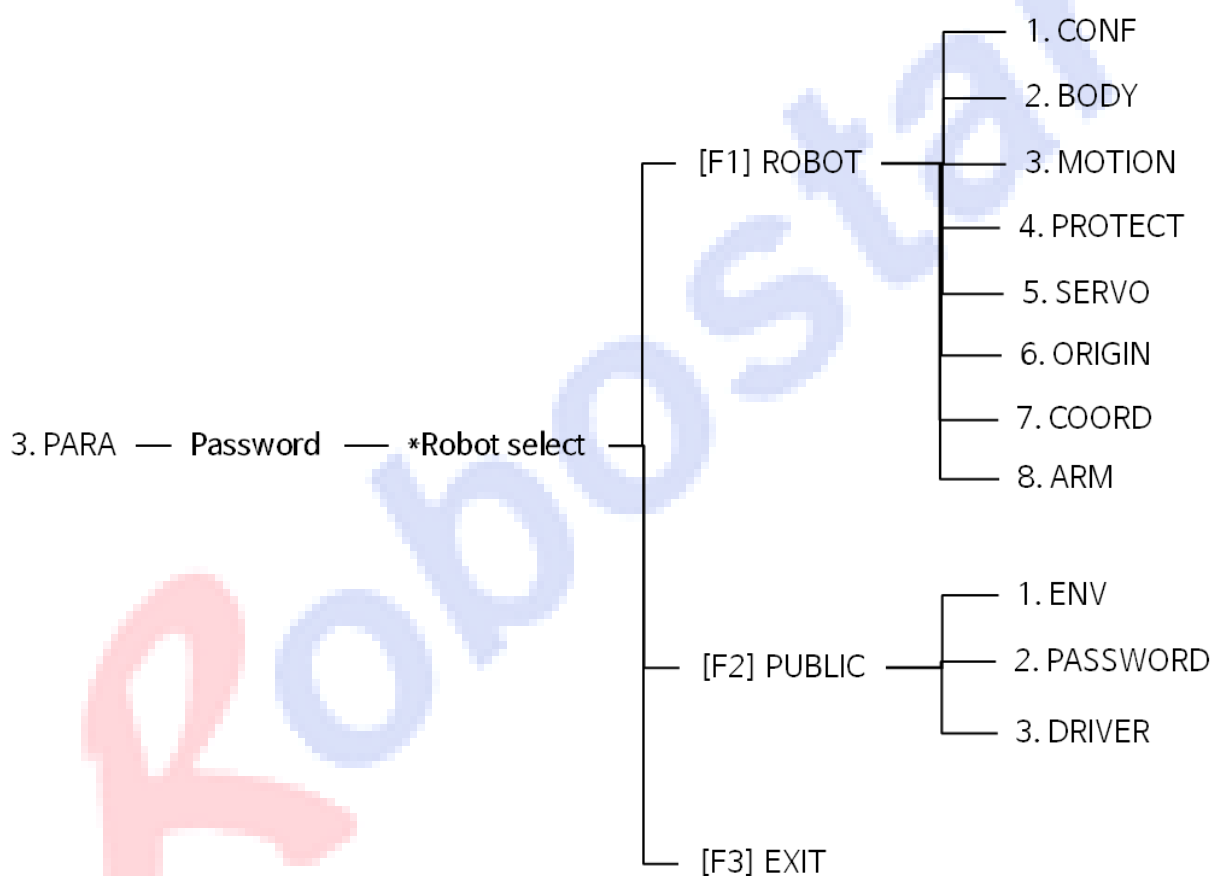
■ 参数Group的种类

GROUP		内容	备注
ROBOT	CONF	设置Robot使用环境。	
	BODY	设置Robot机械部的形态。	
	MOTION	设置Robot动作相关参数。	
	PROTECT	设置保护系统参数。	
	SERVO	设置Servo 电机和Driver相关参数。	
	ORIGIN	设置Origin 相关参数。	
	COORD	坐标系变换时使用。	
	ARM	设置Hand特殊功能相关参数。	
PUBLIC	ENV	设置外部通信, 扩展 I/O, Safety, 附加功能等动作环境。	
	PASSWORD	设置密码。	
	DRIVER	设置符合控制器的Driver 文件。	

2.3.1.1 参数模式流程图

- 参数大体分为 2个 Group.
 - ROBOT PARAMETER: 设置机械部的形态、动作、保护、Servo电机、电机驱动、坐标系等每个Robot都需要设定的参数值.
 - PUBLIC PARAMETER: 设置通信、外部I/O 端口, 密码等共同使用的参数值.

■ Parameter Menu Tree



2.3.1.2 参数Group画面OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RCM6-Series
Version :3.01.33
(TR 141009)
Press ENTER Key

- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)



- 按下ENTER Key 移动到初始菜单画面

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN input : #

Step 2.

选择EDIT

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN input : #



- 选择1. EDIT

选择PARAMETER

<EDIT MENU>
1. JOB 2. VAR
3. PARA

input : #



- 选择3. PRAR

Step 3.

输入PASSWORD

<EDIT MENU>
1. JOB 2. VAR
3. PARA

Password = #####



- 输入Password (初始值: "1111")

~



- 按下"ESC" 键取消输入Password

<EDIT MENU>
1. JOB 2. VAR
3. PARA

Invalid!!

- 输入错误Password 时显示 "Invalid"

Step 4.

选择Robot

<EDIT PARA>
ROBOT TYPE
*R1 AWDT1S026
R2 NO-EXIT
R3 NO-EXIT

ROBOT PUB EXIT



- 按下"UP/DOWN" 键选择编辑参数的 Robot
- 选择的Robot行最前面显示 "*"

选择ROBOT参数

<EDIT PARA>
ROBOT TYPE
*R1 AWDT1S026
R2 NO-EXIT
R3 NO-EXIT

ROBOT PUB EXIT



- 编辑ROBOT 参数时按下"F1" 键

选择PUBLIC 参数

<EDIT PARA>
ROBOT TYPE
*R1 AWDT1S026
R2 NO-EXIT
R3 NO-EXIT

ROBOT **PUB** EXIT



- 编辑PUBLIC 参数时按下 "F2" 键

取消编辑PARAMETER

<EDIT PARA>
ROBOT TYPE
*R1 AWDT1S026
R2 NO-EXIT
R3 NO-EXIT

ROBOT PUB **EXIT**



- 取消编辑恢复到之前画面时按下 "F4" 或 "ESC" 键

■ 参考事项

- Password 可以变更，变更方法请参考[2.5.3.2.1 密码变更方法](#).

2.3.1.3 参数设定

- 通过例子说明设定变更后的参数的方法。
 - Ex) 'ENC' 修正: [F1] ROBOT ⇒ 3. MOTION ⇒ 1. ENC

Step 1.

移动到要变更的PARAMETER 项

〈ROBOT PARAMETER〉
 1: CONF 2: BODY
3: MOTION 4: PROTECT
 5: SERVO 6: ORIGIN
 7: COORD

input : #



- 选择3: MOTION

〈MOTION〉
1: ENC 2: JONT
 3: LINR 4: INIT_V
 5: STOP_T

item #



- 选择1: ENC

Step 2.

移动Cursor位置

〈ENC〉
 ENCODER (pulse/rev)
 T : 131072 Z : 131072
 R1: 131072 R2: 131072
 X : 131072 6 : 131072
 R3: 131072 R4: 131072

〈ENC〉
 ENCODER (pulse/rev)
 T : 131072 Z : 131072
 R1: 131072 R2: 131072
 X : 131072 6 : 131072
 R3: 131072 R4: 131072



- 按下方向键移动到要编辑的值
- 通过Cursor位置确认移动结果

页面变更

〈ENC〉
 ENCODER (pulse/rev)
 Q1: 131072 V1: 131072
 Q2: 131072 V2: 131072
 Q3: 131072 V3: 131072
 Q4: 131072



- Robot轴超过 8 个 时 可以利用"PAGE UP/DOWN" 键切换画面

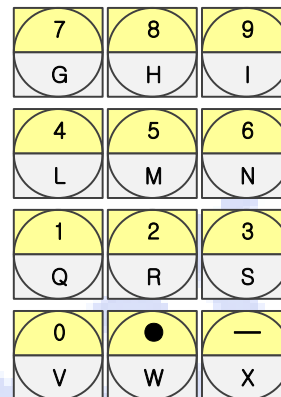
Step 3.

PARAMETER 值变更

```

<ENC>
ENCODER (pulse/rev)
T : 131072    Z : 131072
R1: 131072    R2: 100000
X : 131072    6 : 131072
R3: 131072    R4: 131072
  
```

- 按下要变更的数字值键输入



- ON/OFF 或者 TYPE 变更时按下 "ENTER" 键变更值

```

<ENC>
ENCODER (pulse/rev)
T : 131072    Z : 131072
R1: 131072    R2: 100000
X : 131072    6 : 131072
R3: 131072    R4: 131072
  
```

ENTER

- 按下"ENTER" 键反应到画面中

Step 4.

PARAMETER 值保存及取消

```

<ENC>
ENCODER (pulse/rev)
T : 131072    Z : 131072
R1: 131072    R2: 100000
X : 131072    6 : 131072
R3: 131072    R4: 131072
  
```

update? (enter/esc)

Range Error 发生时

```

<ENC>
ENCODER (pulse/rev)
T : 131072    Z : 131072
R1: 131072    R2: 0
X : 131072    6 : 131072
ind = 3, val = 0.00000
M=999999.00, m = 10000
range error!!!
  
```

ESC

- 变更值后按下"ESC" 键画面中显示 update 消息
- 设定值超出最大范围或未到达最小范围时输出 range error 错误信息

⟨MOTION⟩
1: ENC 2: JONT
3: LINR 4: INIT_V
5: STOP_T

item #

ENTER

ESC

- 按下"ENTER" 或 "ESC"键不暂时保存值，恢复到之前画面。

Range Error 未发生时

⟨MOTION⟩
1: ENC 2: JONT
3: LINR 4: INIT_V
5: STOP_T

item #

ENTER

ESC

- 按下"ENTER" 键暂时保存值恢复到上一级画面。
- 按下"ESC" 键不保存参数值恢复到上一级画面。

⟨ROBOT PARAMETER⟩
1: CONF 2: BODY
3: MOTION 4: PROTECT
5: SERVO 6: ORIGIN
7: COORD

input : #

ESC

- 移动到上一级画面

⟨EDIT PARA⟩
ROBOT TYPE
*R1 AWDT1S026
R2 NO-EXIT
R3 NO-EXIT

ROBOT PUB EXIT

ESC

- 移动到上一级画面

⟨EDIT MENU⟩
1. JOB 2. VAR
3. PARA

input : #

ESC

- 移动到EDIT菜单将变更的参数值反映到数据文件最终保存。

■ 参考事项

- ① PARAMETER 的永久保存时间点为移动到EDIT菜单时，而不是按下ENTER的时间点。

2.3.2 ROBOT PARAMETER

2.3.2.1 CONF (Configuration)

- 设置Robot的使用环境，详细项如下表.

■ 详细项

GROUP	内容	设定值
1: RDIS	选择是否使用所选Robot.	参照说明
2: XDIS	选择是否使用所选Robot的各轴.	参照说明
3: INIT	设置Robot机械部的规格到控制器.	出厂时设定

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 1: CONF

Step 1.

ROBOT CONF 移动

<ROBOT PARAMETER>
 1: CONF 2: BODY
 3: MOTION 4: PROTECT
 5: SERVO 6: ORIGIN
 7: COORD

input : #



- 选择1: CONF

ROBOT CONF 설정화면 확인

<ROBOT CONF>
 1: RDIS 2: XDIS
 3: INIT

item #

- ROBOT CONF 画面确认

2.3.2.1.1 RIDS (Robot Disable)

- 选择是否使用所选Robot.

Step 1.

RDIS 画面移动

〈ROBOT CONF〉
1: RDIS 2: XDIS
3: INIT

item #



- 选择1: RDIS

〈RDIS〉
ROBOT DISABLE
MASK : ☐ OFF

- 确定RDIS 设定画面

■ 设定方法

① 设定值说明

- ON: 当前Robot不运行.
- OFF: 当前Robot正在运行.

Step 1.

RDIS 设定方法

〈RDIS〉
ROBOT DISABLE
MASK : ☐ OFF



- 按下"ENTER" 键ON, OFF 反转
(ON -> OFF, OFF -> ON)

〈RDIS〉
ROBOT DISABLE
MASK : ☒ ON

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 的参数值保存及取消方法.

2.3.2.1.2 XDIS (Axis Disable)

- 选择是否使用所选Robot各轴.

Step 1.

XDIS画面移动

(ROBOT CONF)
 1: RDIS **2: XDIS**
 3: INIT

item #

(XDIS)
 AXIS DISABLE
 T : ☒ OFF Z : OFF
 R1: OFF R2: OFF
 X : OFF 6 : ON
 R3: OFF R4: OFF



- 选择2: XDIS

- 确认RDIS 设定画面

Step 2.

无法设定时

(ROBOT CONF)
 1: RDIS 2: XDIS
 3: INIT

cancel robot disable

(ROBOT CONF)
 1: RDIS 2: XDIS
 3: INIT

item #

- "Cancel robot disable" 信息输出到 Command 窗口



- 输入"ENTER" 或 "ESC"键恢复到之前画面， RDIS的值设定为OFF后再次执行 Step 1.

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 设定值说明

- ON: 选择的轴(Axis)不运行.
- OFF: 选择的轴(Axis)运行.

Step1.

XDIS 设定方法

```

<XDIS>
  AXIS DISABLE
T : OFF      Z : OFF
R1: OFF      R2: OFF
X : OFF      6 : ON
R3: OFF      R4: OFF
  
```

```

<XDIS>
  AXIS DISABLE
T : OFF      Z : OFF
R1: OFF      R2: OFF
X : OFF      6 : OFF
R3: OFF      R4: OFF
  
```

- 按下"ENTER" 键设置ON/OFF 值

■ 保存及取消方法

- ① 请参考 [2.5.1.3 参数设定](#) 参数值保存及取消方法.

■ 参考事项

- ① Disable(ON)的轴相应坐标值不会显示在Teach Pendant.
- ② Ex) 6轴设置为ON时

<XDIS 画面>

```

<XDIS>
  AXIS DISABLE
T : OFF      Z : OFF
R1: OFF      R2: OFF
X : OFF      6 : ON
R3: OFF      R4: OFF
  
```



<EDIT JNT 画面>

```

<EDIT PNT>      V: 5 BL
R2AWS1          CURR
T : 0            Z : 0
R1: 0            R2: 0
X : 0            6 : 0
R3: 0            R4: 0
  
```

2.3.2.1.3 INIT (Initialize)

- Robot机械部的规格设定在控制器上.

Step 1.

INIT 画面移动

<ROBOT CONF>
1: RDIS 2: XDIS
3: INIT

item #



- 选择3: INIT

Step 2.

输入PASSWORD

<ROBOT CONF>
1: RDIS 2: XDIS
3: INIT

Password = ####

CHANGE ROBOT MODEL

Do you want to
change? (ENT/ESC)



- 输入Password

~



- 按下"ESC" 键取消输入 Password



- 按下"ENTER" 键INIT 画面



- 取消编辑时按下 "ESC" 键移动到上一级画面

Step 3.

确认INIT 画面

<INIT-ROBOT TYPE>
1: TRANSFER
2: SCARA

item #

- 确认INIT 设定画面

※ 注意 ※

- INIT 设定在出厂时根据Robot机械部的状态设置最佳值.
- 请勿任意变更工厂最终检查者设定的值.
- 需要变更设定值时请咨询客服人员.

■ 设定方法

① ROBOT TYPE 选择

Step 1.

ROBOT TYPE 选择

〈INIT-ROBOT TYPE〉
1: TRANSFER
2: SCARA

item #

- 选择相应Robot类型

Step 2.

选择取消时

〈INIT-ROBOT TYPE〉
1: TRANSFER
2: SCARA

item #

ESC

- 按下"ESC" 键恢复到上一级画面

② 选择ROBOT APPLICATION

- OLD SEL: 显示之前选择的Application类型.
- 1. AF: Atmospheric FPD
2. AW: Atmospheric WAFER
3. VF: Vacuum FPD
4. VW: Vacuum WAFER
5. DEFAULT: 输入默认值

Step 1.

ROBOT APPLICATION 选择

〈ROBOT APPLICATION〉
OLD SEL:NONE
1: AF 2: AW
3: VF 4: VW
5: DEFAULT

item #



~



- 确认可选Robot Application类型后输入选择的编号移动到ARM TYPE 选择画面

〈ROBOT ARM TYPE〉
SEL:NONE
1: AFS 2: AFD
3: AF4 4: AF6

item #

- 确认ARM TYPE 选择 画面

Step 2.

选择取消时

〈ROBOT APPLICATION〉
SEL:NONE
1: AF 2: AW
3: VF 4: VW
5: NO-EXIT

item #

ESC

- 按下"ESC" 键恢复到上一级画面

③ ROBOT ARM TYPE 选择 (SCARA 类型在 ARM TYPE 选择过程中 SKIP)

- OLD SEL: 显示之前选择的 Arm 类型.
- 1. __S: Robot Arm为一个 (Single), 前面两位输出之前选择的Application Type.
- 2.__D: Robot Arm为两个 (Double)
- 3.__4: Robot Arm为四个
- 4.__6: Robot Arm为六个

Step 1.

ROBOT ARM TYPE 选择

〈ROBOT ARM TYPE〉
OLD SEL:NONE
1: AFS 2: AFD
3: AF4 4: AF6

item #



~



- 确认可选Robot Arm类型后输入选择的编号移动到FILE SELECT 画面

〈FILE SELECT〉
APP: AF A_TYPE:D
* 0. AFDT1S049-020
1. AFDT1S043-020-28

item #

- 确认FILE SELECT 画面

Step 2.

选择取消时

```

<ROBOT ARM TYPE>
SEL:NONE
1: AFS          2: AFD
3: AF4          4: AF6

```

item #



- 按下"ESC" 键恢复到之前画面(ROBOT APPLICATION 选择画面)

④ 选择文件

- 排列选择的Robot Application和Arm类型相应的文件显示在画面.

Step 1.

文件选择及移动

```

<FILE SELECT>
APP: AF      A_TYPE:D
* 0. AFDT1S049-020
- 1. AFDT1S043-020-28

```

item #



- 利用"上下方向键"变更 * 位置, 选择要初始化的文件
- 文件个数多时 可以 按下 "PAGE UP/DOWN" 键移动

确认整体文件名

```

<FILE SELECT>
APP: AF      A_TYPE:D
* 0. AFDT1S049-020
  1. AFDT1S043-020-28

```

item #



- 可以利用"左右方向键"确认整体文件名

```

<FILE SELECT>
APP: AF      A_TYPE:D
* 0. DT1S049-020
  1. DT1S043-020-281

```

item #

最终选择

<FILE SELECT>
 APP: AF A_TYPE: D
 0. AFDT1S049-020
 * 1. AFDT1S043-020-28

item #

ENTER

- 移动到所需文件后按下"ENTER" 键移动到下一个画面

Step 2.

设定有变更时

APPLICATION: AF
 ARM_TYPE: D
 FILE : AFDT1S043-020-281

update? (ENT/ESC)

ENTER

ESC

- 显示整体用户可选内容，输出update 信息
- 按下"ENTER" 键移动到设定更新画面
- 按下"ESC"键取消更新，恢复到上一级画面

设定无变更时

<ROBOT CONF>
 1: RDIS 2: XDIS
 3: INIT

item #

- 如果无变更则移动到上一级画面

Step 3.

选择取消时

APPLICATION: AF
 ARM_TYPE: D
 FILE : AFDT1S043-020-281

update? (ENT/ESC)

ESC

- 按下"ESC" 键恢复到之前画面(ROBOT TYPE 选择画面)

<ROBOT ARM TYPE>
 SEL:D
 1: AFS 2: AFD
 3: AF4 4: AF6

item #

- 确认ROBOT ARM TYPE 选择画面
- 可以通过SEL栏确认之前选择的Robot Arm类型

⑤ 设定更新

Step 1.

设定更新

INITIALIZE PARAMETER

Do you want to init
Para? [ENTER/ESC]

ENTER

- 按下"ENTER" 键输出是否初始化参数值的消息

INITIALIZED!!
Reset Please!

ENTER

- 按下"ENTER" 键完成更新

Press Any Key

Step 2.

取消更新

INITIALIZE PARAMETER

Do you want to init
Para? [ENTER/ESC]

ESC

- 按下"ESC" 键不保存参数值回到上级画面

<ROBOT CONF>
1: RDIS 2: XDIS
3: INIT

item #

■ 参考事项

- ① 设置Robot类型/系列/规格时 确认器具部和控制器的标签后设置.
- ② Robot标签的位置请参考 第7章 附录 [7.1 Robot标签位置](#) 内容.

2.3.2.2 BODY

- 设置Robot机械部形态，详细项如下表。

■ 详细项

GROUP	内容	范围	设定值
1: JNT	设置机械部各关节TYPE, 减速比, PITCH, 方向.	参照说明	
2: LENG	设置Robot的 Arm 长度.	参照说明	
3: TOOL	设置粘贴在机械部 Z轴 末端的作业Tool的 Offset 值.	-1,000~1,000	
4: OFFS	设置机械部坐标原点和控制器坐标原点之间的差值.	参照说明	
5: RANG	设置S/W Limit范围.	-999,999 ~ 999,999	
6: SYNC	设置轴同步情况	参照说明	
7: DSYNC	设置Z轴双同步情况.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 2: BODY

Step 1.

BODY 移动

<ROBOT PARAMETER>
 1: CONF 2: **BODY**
 3: MOTION 4: PROTECT
 5: SERVO 6: ORIGIN
 7: COORD

input : #



- 选择2: BODY

确定BODY 设定画面

<BODY>
 1: JNT 2: LENG
 3: TOOL 4: OFFS
 5: RANG

item #

- 确认BODY 画面

2.3.2.2.1 JNT (Joint)

- 设置机械部关节TYPE, 减速比, PITCH, 方向等于关节相关事项, 详细项如下表.

■ 详细项

GROUP	内容	范围	设定值
1: TYPE	设置关节类型.	参照说明	
2: GEAR	设置减速比.	0.001 ~ 1,000	
3: PTCH	设置直线移动轴的1次旋转移动距离.	0.1 ~ 1,000	
4: PULY1	设置Pulley 1的比例.	0.001 ~ 1,000	
5: PULY2	设置Pulley 2的比例.	0.001 ~ 1,000	
6: PULY3	设置Pulley 3的比例.	0.001 ~ 1,000	
7: DISP	设置关节方向.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 2: BODY -> 1: JNT

Step 1.

JNT 画面移动

<BODY>
 1: JNT
 3: TOOL
 5: RANG

2: LENG
 4: OFFS

item #



- 选择1: JNT

<JNT>
 GROUP: JOINT
 1: TYPE
 3: PTCH
 5: PULY2
 7: DISP

2: GEAR
 4: PULY1
 6: PULY3

item #

- 确认JNT 画面

1) TYPE

- 设置关节类型.

Step 1.

TYPE 画面移动

〈JNT〉
GROUP : JOINT
1: TYPE 2: GEAR
3: PTCH 4: PULY1
5: PULY2 6: PULY3
7: DISP

item #



- 选择1: TYPE

〈JNT〉
GROUP : JOINT
T : REV Z : TRAN
R1: LINK R2: LINK
X : TRAN 6 : LINK
R3: LINK R4: LINK

- 确认JNT 设定画面

Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

设定方法

- ① 设定值说明

- REV (Revolution Joint): 关节类型为旋转轴时使用.
- TRANS (Translation Joint): 关节类型为直线移动轴时使用.
- LINK: Link类型时使用.

Step 1.

RDIS 设定方法

〈JNT〉
GROUP : JOINT
T : REV Z : TRAN
R1: LINK R2: LINK
X : TRAN 6 : LINK
R3: LINK R4: LINK



- 按下"ENTER" 键选择所需类型
(按照REV -> LINK -> TRAN -> REV 顺序变化)


```
<JNT>
GROUP : JOINT
T : REV      Z : TRAN
R1: LINK     R2: LINK
X : TRAN     6 : REV
R3: LINK     R4: LINK
```

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2) GEAR

- 设置减速比.

Step 1.

GEAR 画面移动

<JNT> GROUP : JOINT
 1: TYPE 2: GEAR
 3: PTCH 4: PULY1
 5: PULY2 6: PULY3
 7: DISP

item #



- 选择2: GEAR

<GEAR>
 GEAR RATIO
 T : 80 Z : 1
 R1: 50 R2: 50
 X : 1 6 : 1
 R3: 0 R4: 0

- 确认GEAR设定画面

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 电机和器具的减速比 (包括Pully 比)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 减速比需要输入Robot [标签](#) 中显示的值.
- ② Robot标签位置请参考第7章 附录的 [7.1 Robot标签位置](#) 内容.

3) PTCH (Pitch)

- 设置直线移动周的1次旋转移动距离.

Step 1.

PTCH 画面移动

<JNT>
 GROUP : JOINT
 1: TYPE 2: GEAR
 3: **PTCH** 4: PULY1
 5: PULY2 6: PULY3
 7: DISP

item #



- 选择 3: PTCH

<PITCH>
 PITCH RATIO
 T : 1 Z : 10
 R1: 1 R2: 1
 X : 1 6 : 1
 R3: 1 R4: 1

- 确认PTCH 设定画面

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) 的PARAMETER 值变更.
- ② 设定值说明
 - 轴1次旋转器具移动距离(mm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

4) PULY1 ~ 3 (Pulley 1~3)

- 设置Pulley 1 ~3的比例.

Step 1.

PULY 画面移动

<JNT>
 GROUP : JOINT
 1: TYPE 2: GEAR
 3: PTCH 4: **PULY1**
 5: **PULY2** 6: **PULY3**
 7: DISP
 item #

<PULLEY1>
 PULLEY1 RATIO
 T : 1 Z : 1
 R1: 1 R2: 1
 X : 1 6 : 1
 R3: 1 R4: 1



- 选择4: PULY1

~

- 选择5: PULY2



- 选择6: PULY3

- PULY 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Pulley1~3 比例

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

5) DISP (Display Direction)

- 设置关节的方向.

Step 1.

DISP 画面移动

<JNT>
 GROUP : JOINT
 1: TYPE 2: GEAR
 3: PTCH 4: PULY1
 5: PULY2 6: PULY3
7: DISP

item #



- 选择7: DISP

<DISP>
 Display direction
 T : ☒ Z : +
 R1: - R2: +
 X : + 6 : -
 R3: - R4: -

- DISP 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 设定值说明
 - - : 坐标系上向负方向移动时使用.
 - +: 坐标系上向正方向移动时使用.

Step1.

DISP 设定方法

<DISP>
 Display direction
 T : - Z : +
 R1: - R2: +
 X : + 6 : ☒
 R3: - R4: -



- 按下"ENTER" 键设置 -/+ 值

<DISP>
Display direction
T : - Z : +
R1: - R2: +
X : + 6 : +
R3: - R4: -

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

■ 参考事项

- ① 出厂时已设定标准值..
② 设定标准值以外的值时无法完成补间动作、码垛动作中无法完成W轴补间.

2.3.2.2.2 LENG (Length)

- 设置Robot Arm 长度.

Step 1.

LENG 画面移动

⟨BODY⟩

1: JNT

3: TOOL

5: RANG

2: LENG
4: OFFS

item #



- 选择2: LENG

⟨LENG⟩

ARM LENGTH (mm)

A-ARM: 135

B-ARM: 135

- 确认LENG 画面

Step 2.

无法设定时

⟨BODY⟩

1: JNT

3: TOOL

5: RANG

2: LENG
4: OFFS

only use for SCARA!

- 直角坐标Robot时Command 窗口中输入
출처에 "only use for SCARA!" 信息

⟨BODY⟩

1: JNT

3: TOOL

5: RANG

2: LENG
4: OFFS

item #



- 按下"ENTER" 或 "ESC"键恢复到之前画面

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Robot Arm 长度(mm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超出最大范围或未达到最小范围时输出错误信息.

■ 参考事项

- ① 仅适用于水平多关节 (SCARA)/运送(TRANSFER) Robot.
- ② Arm的长度在工厂最终检查中利用JIG算出，显示在Robot主体背面的标签上，确认后输入即可.
- ③ Robot标签位置请参考第7章 附录的[7.1 Robot标签位置](#) 内容.

※ 注意 ※

- 错误的Arm 长度无法正确完成补间动作 (LMOV, AMOV, CMOV), 码垛动作 (PMOV). 即, Robot的移动会脱离实际Teaching的位置坐标.
- 最初安装Robot时请务必确认机械部的标签和控制器的设定值.

2.3.2.2.3 TOOL

- 设置粘贴在机械部Z轴末端的作业Tool的Offset 值.

Step 1.

TOOL 画面移动

(BODY)
 1: JNT 2: LENG
 3: **TOOL** 4: OFFS
 5: RANG

item #

(TOOL)
 TOOL OFFSET 0
 dX: 0
 dY: 0
 dZ: 0

T0 T1 T2 T3

(TOOL)
 TOOL OFFSET 0
 dX: 0
 dY: 0
 dZ: 0

T0 T1 T2 T3

(TOOL)
 TOOL OFFSET **0**
 dX: 0
 dY: 0
 dZ: 0

T0 T1 T2 T3



- 选择3: TOOL

- TOOL 画面确认



- 选择与要编辑的Tool匹配的键 ([F1] ~ [F4])

- 确认选择的Tool值，确认是否弹出正确的选择窗口

Step 2.

无法设定的Robot时

(BODY)
 1: JNT 2: LENG
 3: TOOL 4: OFFS
 5: RANG

no support !!!

- 没有使用TOOL 的Robot时画面中弹出 "no support!!!" 信息

<BODY>
 1: JNT 2: LENG
 3: TOOL 4: OFFS
 5: RANG

item #

ENTER

ESC

- 按下"ENTER" 或"ENTER" 혹은 "ESC" 키를 눌러 이전画面으로 복귀键回到之前画面

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 作业Tool的位置偏差(mm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超出最大范围或未达到最小范围时输出错误信息.

■ 参考事项

- ① SCARA Robot的直角坐标系中算出各方向 (X,Y,Z)的移动偏差 (Tool Offset).
- ② Tool Offset 值修正在下面的情况是必要的.

※ 注意 ※

- Z 轴末端(End point)上的作业 Tool Z轴黏贴时与中心线有偏差时
- W轴(第4个轴, 旋转运动轴)利用Tool末端执行补间动作(LMOV, AMOV), 码垛动作(PMOV)时

2.3.2.2.4 OFFS (Offset)

- 直接输入机械部坐标原点和控制器坐标原点之间的差值设置，或当前值设置为坐标原点，当前位置(pulse)作为差值保存并校正. 详细项如下.

■ 详细项

GROUP	内容	范围	设定值
1: EDIT	输入机械部坐标原点和控制器坐标原点之间的差值设置.	-1,000~1,000	
2: CALIB	设置当前位置值为坐标原点，保存当前位置 (pulse)为差值保存并校正.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 2: BODY -> 4: OFFS

Step 1.

OFFS 画面移动

⟨BODY⟩
1: JNT
3: TOOL
5: RANG

2: LENG
4: OFFS

item #



- 4: OFFS 选择

⟨OFFS⟩
OFFSET SETTING

EDIT CALIB

- OFFS 画面确认

1) EDIT (Offset Edit)

- 直接输入机械部坐标原点和控制器坐标原点之间的差值设置.

Step 1.

EDIT 画面移动

<OFFS>
OFFSET SETTING

EDIT CALIB



- 按下"F1" 键选择EDIT

<OFFS>
OFFSET SETTING
T : 0
Z : 0
R1: 0
R2: 0
X : 0
G : 0

- OFFSET EDIT 设定画面确认

■ Cursor位置移动

- 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- 设定值说明
 - 机械部坐标原点和控制器坐标原点之间的差值

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- Offset值记载在Robot标签.
- Robot标签位置请参考第7章附录的 [7.1 Robot标签位置](#) 内容.
- Offset 值校正**在以下情况下是必要的.

※ 注意 ※

- 冲突导致的器具部位置变化（歪曲）和Point不对齐
- 替换电机后发生位置变化
- 补间动作(AMOV, LMOV), 码垛动作(PMOV)不正确时

Robostar

2) CALIB (Offset Calibration)

- 当前位置值设置为坐标原点，保存当前位置 (pulse)为差值并校正。

Step 1.

CALIB 画面移动

〈OFFS〉
OFFSET SETTING

EDIT **CALIB**



- 按下"F2" 键选择CALIB

〈OFFS〉
ZERO CALIBRATION 1/4
T : 0 Z : 0
R1: 0 R2: 0

- OFFSET CALIB 设定画面确认

T Z R1 R2

■ 页面移动

- 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更。

■ 设定方法

- 设定值说明
 - 保存当前位置 (pulse)值。

Step 1.

CALIB 设定方法

〈OFFS〉
ZERO CALIBRATION 1/4
T : 2.61 Z : -5.25
R1: 0 R2: 10.47

- 移动Robot对齐到粘贴在Robot的箭头
(▲▼)

T Z R1 R2
X 6



- 选择进行校正的轴相匹配的键 ([F1] ~ [F4])

〈OFFS〉
ZERO CALIBRATION 1/4
T : 2.61 Z : -5.25
R1: 0 R2: 10.47



T Z R1 R2

② T轴选择例子

Step 1.

T轴选择

<OFFS>
ZERO CALIBRATION 1/4
T : 2.61 Z : -5.25
R1: 0 R2: 10.47



- 按下"F1"键选择 T轴

T Z R1 R2

Step 2.

选择是否保存

<OFFS>
ZERO CALIBRATION 1/4
T : 2.61 Z : -5.25
R1: 0 R2: 10.47



- 确认Encoder type 和是否保存

- 按下"ENTER"保存



- 按下"ESC" 取消保存

save T? (ENT/ESC)

选择保存时

<OFFS>
ZERO CALIBRATION 1/4
T : 0 Z : -5.25
R1: 0 R2: 10.47

- 完成设定后相应轴值变为0，回到之前的画面

T Z R1 R2

选择取消时

<OFFS>
ZERO CALIBRATION 1/4
T : 2.61 Z : -5.25
R1: 0 R2: 10.47

- 无变化回到之前画面

T Z R1 R2

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

■ 参考事项

- 无箭头(▼▲) 符号时， SCARA Robot A, B Arm 直线展开. Transfer Robot 会并行折叠与 L,R 轴相应的两个 Arm.

3) 判断修正的Offset 值是否正确的方法

- 修正的Offset 值是否正确可以通过Point Teaching画面确认.

Step 1.

JOG 画面移动

<TEACH MENU>
 1. EDIT 2. RUN
 3. JOG 4. I/O
 5. ORIGIN 6. INFO

 0. SHUTDOWN input : #



- 选择3. JOG

Step 2.

选择Robot

<JOG>
 ROBOT TYPE
 *R1 AWDT1S026
 R2 NO-EXIT
 R3 NO-EXIT

 JOINT HOME



- 按下"UP/DOWN" 键选择要编辑参数的 Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

<JOG>
 ROBOT TYPE
 *R1 AWDT1S026
 R2 NO-EXIT
 R3 NO-EXIT

 JOINT HOME



- 按下"F1" 键选择JOINT

<EDIT JNT> V: 5 BL
 AWDT1S026 CURR
 T : 0 Z : 0
 R1: 0 R2: 0
 X : 0 6 : 0

- 在CURR 画面确认是否所有轴的坐标值为"0"

PGET MDI CORD EXIT

2.3.2.2.5 RANG(Range)

- 设定S/W Limit范围.

Step 1.

RANG 画面移动

<BODY>
 1: JNT 2: LENG
 3: TOOL 4: OFFS
5: RANG

item #



- 选择5: RANG

<RANGE>
 Software Limit Range
 T : 5 335
 Z : -5 405
 R1: -260 260
 R2: -260 260
 X : -5 1505
 6 : 0 0

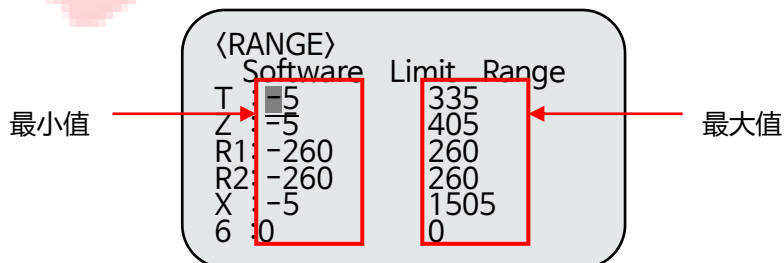
- RANG 画面确认

■ Cursor位置移动

- 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- 设定值说明
 - Limit 范围值 (旋转轴: degree, 直线移动轴: mm)
 - 左侧: Limit最小值, 右侧: Limit最大值



■ 保存及取消方法

- 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.2.2.6 SYNC (Synchronous Robot)

- 选择SCARA 类型的轴同步情况.

■ 详细项目

GROUP	内容	范围	设定值
1: TR	设定Transfer Robot的 SCARA 同步类型.	参照说明	
2: ROBOT	设置其它Robot的SCARA 同步类型.	参照说明	

■ OPEN 顺序

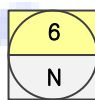
- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 2: BODY -> 6: SYNC

Step 1.

SYNC 画面移动

<BODY>
 1: JNT 2: LENG
 3: TOOL 4: OFFS
 5: RANG **6: SYNC**
 7: DSYNC

item #



- 选择6: SYNC

<SCARA SYNC TYPE>
 1: TR 2: ROBOT

item #

- 确认SYNC 画面

2.3.2.2.7 DSYNC

- 设置Z轴双同步情况.

■ 详细项目

GROUP	内容	范围	设定值
1: ENABLE	设置使用情况及双同步类型.	参照说明	
2: AXIS	设置Master轴和Slave轴误差允许范围.	参照说明	

■ OPEN 顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 2: BODY -> 7: DSYNC

Step 1.

DSYNC 画面移动

<BODY>
 1: JNT 2: LENG
 3: TOOL 4: OFFS
 5: RANG 6: SYNC
7: DSYNC

item #



- 选择7: DSYNC

<DUAL SYNC>
 1: ENABLE 2: AXIS

item #

- 确认DSYNC 画面 ,

1) ENABLE

- 设置使用情况及双同步类型.

Step 1.

ENABLE 画面移动

<DUAL SYNC>
1: ENABLE 2: AXIS

item #



- 选择1: ENABLE 선

<DSYNC ENABLE>
ENABLE SELECT
USED : ☐ OFF
TYPE : PAIR

- 确认ENABLE 画面

■ 光标位置移动

- ① 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- ① 设定值说明
 - USED: 设置功能使用情况.

Step 1.

设定方法

<DSYNC ENABLE>
ENABLE SELECT
USED : ☐ OFF
TYPE : PAIR



- 按下"ENTER"键ON,OFF反转
(ON -> OFF, OFF -> ON)

<DSYNC ENABLE>
ENABLE SELECT
USED : ☒ ON
TYPE : PAIR

- TYPE: PAIR(一对一起移动的类型), PARASITE(相互黏贴的类型) 类型中选择.

Step 1.

设定方法

```
<DSYNC ENABLE>
ENABLE SELECT
USED :ON
TYPE :PAIR
```

```
<DSYNC ENABLE>
ENABLE SELECT
USED :ON
TYPE :PARASITE
```

ENTER

- 按下"ENTER"键PAIR,PARASITE反转
(PAIR -> PARASITE,
PARASITE -> PAIR)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2) AXIS

- 设置Master轴和Slave轴误差允许范围.

Step 1.

AXIS 画面移动

<DUAL SYNC>
1: ENABLE **2: AXIS**

item #



- 选择2: AXIS

<DSYNC AXIS SET>
 AXIS NO.
MASTER : 3
SLAVE : 4
INRANGE : 0.2

- 确认AXIS画面

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.

① 设定值说明

- MASTER: Master轴编号
- SLAVE: Slave轴编号
- INRANGE: 误差允许范围 (mm)

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

① 设定值超出最大范围或未达到最小范围时输出错误信息.

2.3.2.3 MOTION

- 设定Robot动作相关参数，详细项如下表。

■ 详细项

GROUP	内容	范围	设定值
1: ENC	设置编码器一次旋转的脉冲数.	10,000~999,999	131,072
2: JONT	设置Joint Motion 相关变量.	参照说明	
3: LINR	设置Linear Motion 相关变量.	参照说明	
4: INIT_V	设置Override 初始速度值.	5 ~ 100	
5: STOP_T	设置Stop 减速时间.	10 ~ 1,000	
6: CUM_DIST	设置各轴移动距离累积功能使用情况及初始化设定值.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 3: MOTION

Step 1.

MOTION移动

<ROBOT PARAMETER>
 1: CONF 2: BODY
 3: **MOTION** 4: PROTECT
 5: SERVO 6: ORIGIN
 7: COORD

input : #



- 选择3: MOTION

MOTION 设定画面确认

<MOTION>
 1: ENC 2: JONT
 3: LINR 4: INIT_V
 5: STOP_T

item #

- MOTION 画面确认

2.3.2.3.1 ENC (Encoder)

- 设置编码器一次旋转的脉冲数..

Step 1.

ENC 画面移动

<MOTION>
 1: ENC
 3: LINR
 5: STOP_T
 2: JONT
 4: INIT_V

item #



- 选择1: ENC

<ENC>
 ENCODER (pulse/rev)
 T : 131072 Z : 131072
 R1: 131072 R2: 131072
 X : 131072 6 : 131072
 R3: 131072 R4: 131072

- ENC 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 编码器一次旋转的脉冲数 (pulse/rev)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 需要输入Robot [标签](#)上标记的值.
- ② Robot标签位置请参考第7章附录中的 [7.1 Robot标签位置](#) 内容.

※ 注意 ※

- 根据电机种类设定值不同，请务必确认电机规格.

2.3.2.3.2 JONT (Joint)

- 设置Joint Motion 相关变量.

Step 1.

JONT 画面移动

(MOTION)
 1: ENC
 3: LINR
 5: STOP_T
 2: JONT
 4: INIT_V
 item #

(JONT)
 JOINT MOTION: T
 Mv: 225 At: 1
 Jv: 39.81 Ia: 0.1
 [RPM]-----
 Mv: 3000 Jv: 530.8
 T Z R1 R2

(JONT)
 JOINT MOTION: T
 Mv: 225 At: 1
 Jv: 39.81 Ia: 0.1
 [RPM]-----
 Mv: 3000 Jv: 530.8
 T Z R1 R2

(JONT)
 JOINT MOTION: T
 Mv: 225 At: 1
 Jv: 39.81 Ia: 0.1
 [RPM]-----
 Mv: 3000 Jv: 530.8
 T Z R1 R2



- 选择2: JONT

- JONT 画面确认



- 选择要编辑的轴相匹配的键 ([F1] ~ [F4])

- 确认轴名称和值后，确认是否正确弹出选择的窗口

Cursor位置移动

- ① 请参考 [2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Mv: 最大速度 (旋转轴: degree/sec, 直线移动轴: mm/s)
 - At: 加减速时间(sec)

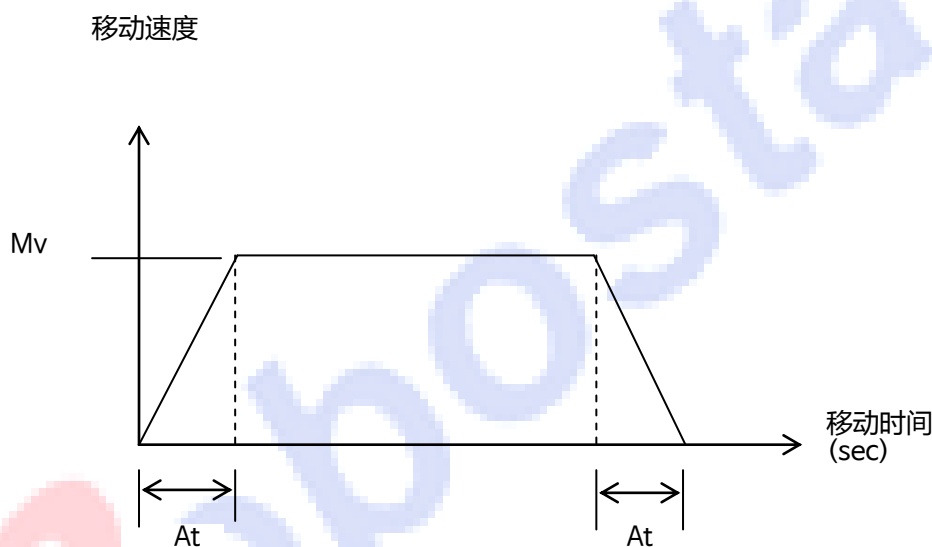
- Jv: Jog移动时的最大速度(旋转轴: degree/sec, 直线移动轴: mm/s)
- Ia: Inching Job单位移动距离(旋转轴: degree, 直线移动轴: mm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 详细说明



1) **Mv**: RUN MODE 中Robot移动时的各轴最高速度(旋转轴: degree/sec, 直线移动轴: mm/s)

- 作业程序中移动速度设置为最高速度(VEL 100%), 也有可能发生随着负荷量和移动距离下降到设定值以下的情况.
- 此外, Robot两轴以上同时移动到POINT时需要保持同步, 根据各轴移动距离有可能速度不同.

2) **At**: 各轴加减速时间 (sec)

- 同样适用于加速时间和减速时间.
- 根据负荷量及机械状态, 实际加减速时间可能比设定时间长.
- 过度的加减速设定可能导致机械部的震动及颤动现象, 对机械寿命也有影响.

• 请在不影响急剧性有效负重及Cycle Time的范围下调整加减速时间.
3) Jv : Point Teaching中利用 Teach Pendant的 Jog 键进行Robot轴移动时Robot移动速度 (旋转轴: degree/sec, 直线移动轴: mm/s) • Jog速度包含低速(Teach Pendant 显示: L), 中速(M), 高速(H) 3种.
4) It : Inching移动中的单位移动距离(旋转轴: degree, 直线移动轴: mm) • Inching移动包含最小 (Teach Pendant 显示: l), 中间(m), 最大(h) 3种.

2.3.2.3.3 LINR (Linear)

- 设置Linear Motion 相关变量.

Step 1.

LINR 画面移动

```

<MOTION>
1: ENC          2: JONT
3: LINR         4: INIT_V
5: STOP_T
  
```

item #



- 选择3: LINR

```

<LINR>
  LINEAR MOTION
Mv : 1000      At : 0.50
Jv : 100       Ia : 10
  
```

- LINR 设定画面确认

Step 2.

无法设定时

```

<MOTION>
1: ENC          2: JONT
3: LINR         4: INIT_V
5: STOP_T
  
```

no support!!!

- 非SCARA Robot时Command窗口中显示 "no support!!" 信息

```

<MOTION>
1: ENC          2: JONT
3: LINR         4: INIT_V
5: STOP_T
  
```

item #



- 按下"ENTER" 或 "ESC"恢复到之前画面

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明

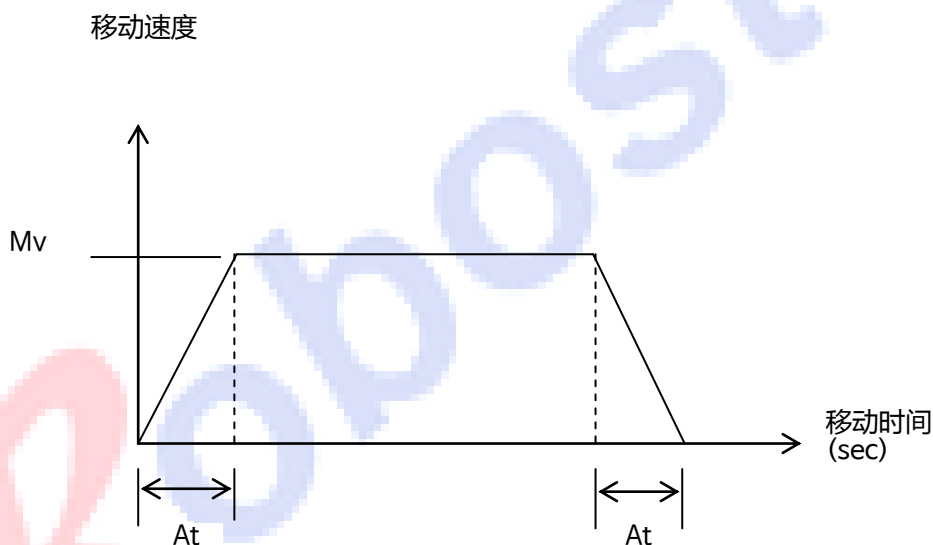
- Mv: 补间移动最大速度 (mm/s)
- At: 补间移动加减速时间(sec)
- Jv: Jog移动时补间移动最大速度 (mm/s)
- Ia: Inching Jog单位移动距离 (mm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 详细说明



1) **Mv**: RUN MODE 下Robot补间移动时的最高速度 (mm/s)

- 作业程序中移动速度设置为最高速度(VEL 100%)，也有可能发生随着负荷量和移动距离下降到设定值以下的情况
- 此外，使用FOS, FIX 等指令或连续性补间移动时根据急剧轨迹变更、负荷量、电机GAIN等，无法达到设定速度.
- 调整移送速度、负荷GAIN 等后操作.

2) **At**: 合成轴加减速时间 (sec)

- 同样适用于加速时间、减速时间.

- 根据负荷量及机械状态加减速时间可能比设定时间长.
- 过度的加减速设定可能导致机械部的震动及颤动现象, 对机械寿命也有影响.
- 请在不影响急剧性有效负重及Cycle Time的范围下调整加减速时间.

- **Jv:** Point Teaching中利用 Teach Pendant的 Jog 键进行Robot轴移动时Robot移动速度 (mm/s)
 - Jog速度包含低速(Teach Pendant 显示: L), 中速(M), 高速(H) 3种.
 - 机械部为SCARA Robot的情况下, 从POINT Teaching转换为XY坐标系后使用JOG 键, 且速度设置为高速(H)时, 根据位置可能以预期速度以上的速度运行, 请注意.

3) **It:** Inching移动(补间移动)中的单位移动距离(直线移动轴: mm)

- Inching移动包含最小 (Teach Pendant 显示: l), 中间(m), 最大(h) 3种..

2.3.2.3.4 INIT_V (Initial Velocity)

- 设置Override 初始速度值.

Step 1.

INIT_V 画面移动

<MOTION>
 1: ENC 2: JONT
 3: LINR 4: INIT_V
 5: STOP_T

item #



- 选择4: INIT_V

<INIT VEL>
 INIT VEL (%)
 VEL: 100

- INIT_V 画面确认

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Override 初始速度值 (%)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① Override 初始速度值可以在RUN 画面中利用 "UP/DOWN" 方向键操作, 反映在速度调整的功能的初始值.

2.3.2.3.5 STOP_T (Stop Time)

- 设置Stop减速时间.

Step 1.

STOP_V 画面移动

<MOTION>
 1: ENC 2: JONT
 3: LINR 4: INIT_V
5: STOP_T

item #



- 选择5: STOP_T

<STOP TIME>
 STOP TIME (%)
 EMG Rate : 100
 MOV Rate : 100
 JOG Rate : 100

- STOP_T 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Override初始速度值 (%)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 4. PARA -> 3: MOTION -> 2.JONT -> At 设定值为基准, 需要执行输入STOP Key相应的停止动作或需要缩短停止时间时, 设置此值并使用.
- ② 范围为 10~100 (%). (100: 使用默认At 设定值, 10: At 设定值的 10%)

2.3.2.3.6 CUM_DIST(Cumulated Distance)

- 设置各轴移动距离累积功能使用情况及初始化设定值.

■ 详细项目

GROUP	内容	范围	设定值
1: SET	设置累积距离初始化值.	0 ~ 999,999	
2: DATA	确认累积距离和初始化Count.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 3: MOTION-> 6:CUM_DIST

Step 1.

MOTION 画面移动

<MOTION>
 1: ENC 2: JONT
 3: LINR 4: INIT_V
 5: STOP_T **6: CUM_DIST**

item #



- 选择6: CUM_DIST

<CUM_DIST>
 1: SET 2: DATA

item #

- 确认CUM_DIST画面

1) SET

- 设置雷击距离初始化基准值.如果累积距离超出设定值,则进行初始化.

Step 1.

SET画面确认

<CUM_DIST>
1: SET 2: DATA

 item #

<SET>
 RESET CRITERIA
 T (rev): 0
 Z (km): 0
 R1 (km): 0
 R2 (km): 0
 X (km): 0
 6 (km): 0



- 选择1:SET

- SET画面确认

■ 游标位置移动

- 请参考 [2.5.1.3 参数设定的](#) 游标位置移动及页面变更.

■ 设定方法

① 设定值说明

- 0: 不使用移动距离累积功能.
- 允许范围值: 0~999,999
- 允许范围单位: 旋转, LINK轴的单位为一次旋转 (rev), 直线移动轴的单位为 (km)

단위

<SET>
 RESET CRITERIA
 T (rev): 0
 Z (km): 0
 R1 (km): 0
 R2 (km): 0
 X (km): 0
 6 (km): 0

■ 保存及取消方法

- 请参考 [2.5.1.3 参数设定的](#) 参数值保存及取消方法.

- 设定值超出最大范围或未达到最小范围时输出错误信息.

2) DATA

- 确认并设置累积的距离和初始化Count.

Step 1.

DATA 画面确认

{ CUM_DIST }
 1: SET **2: DATA**
 item #



- 确认2: DATA

{ DATA }
 DISTANCE: T (rev)
 DIST : 0
 COUNT: 0
 T Z R1 R2

- DATA 画面确认

Step 2.

轴选择

{ DATA }
 DISTANCE: T (rev)
 DIST : 1.25
 COUNT: 1
X 6 R3 R4



- 利用页面移动键及“F1 ~ F4”选择要确认的轴。

■ 游标位置移动

- 请参考[2.5.1.3 参数设定的](#) 游标位置移动及页面变更.

■ 设定方法

- ① 输入选择的轴的累积距离或SET画面中设定的初始化值为初始化Count值进行初始化或变更为所需的值.

累积距离

初始化次数

{ DATA }
 DISTANCE: X (km)
 DIST : 1.25
 COUNT : 1
 X 6 R3 R4

■ 保存及取消方法

■ 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

- ① 设定值超出最大范围或未达到最小范围时输出错误信息.

※注意※

- 除初始Robot设定或特殊情况, 请勿变更累积距离和初始化Count次数.

2.3.2.4 PROTECT

- 设置保护系统参数.

■ 详细项

GROUP	内容	范围	设定值
1: OVS	设置机械部的电机最大允许速度.	500~6,000	
2: OVA	设置机械部最大允许加速度.	3,000~100,000	
3: IPA	设置目标位置误差允许范围.	1.0 ~ 10,000	
4: IPE	到达目标位置的Error允许时间最大值.	1.0 ~ 1,000	
5: IPO	设置到达目标位置判定的最小维持时间.	1.0 ~ 1,000	
6: IRNG	设置In Range 范围或决定是否使用In Range Alarm.	参照说明	
7: URNG	设置用户In Range 使用情况, 范围, 输出接点, Z 轴允许范围等.	参照说明	
8: SENS	设置各轴Limit传感器 (H/W Limit Sensor) 使用情况 & 接点.	-1 ~ 64	
9: BELT	决定Timing Belt 有/无确定用的传感器使用情况 & 接点.	-1 ~ 64	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 4: PROTECT

Step 1.

PROTECT移动

(ROBOT PARAMETER)
 1: CONF 2: BODY
 3: MOTION **4: PROTECT**
 5: SERVO 6: ORIGIN
 7: COORD

input : #



- 选择4: PROTECT

PROTECT 设定画面确认

<PROTECT>

1: OVS	2: OVA
3: IPA	4: IPE
5: IPO	6: IRNG
7: URNG	8: SENS

item #

- PROTECT 画面确认

2.3.2.4.1 OVS (Over Speed Error)

- 设置机械部电机最大允许速度.

Step 1.

OVS 画面移动

<PROTECT>

1: OVS	2: OVA
3: IPA	4: IPE
5: IPO	6: IRNG
7: URNG	8: SENS

item #

<OVS>

OVER	SPEED (rpm)
T : 4500	Z : 4500
R1: 4500	R2: 4500
X : 4500	6 : 4500
R3: 4500	R4: 4500



- 选择1: OVS

- OVS 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

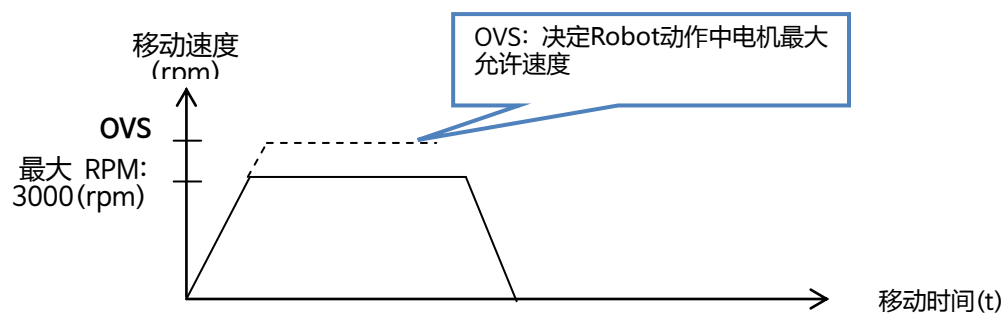
- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 最大允许速度 (rpm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 例子

**※ 注意 ※**

- 确认以下事项时，为了保证完美的移动角度，请修正最大允许速度。
 - ① 电机编码器是否有异常
 - ② 机械部连接电缆状态
 - ③ 连续的补间动作中Robot移动到下一个位置时，发生急剧的方向移动

2.3.2.4.2 OVA (Over Acceleration Error)

- 设置机械部最大允许加速度.

Step 1.

OVA 画面移动

<PROTECT>
 1: OVS
 3: IPA
 5: IPO
 7: URNG
 2: **OVA**
 4: IPE
 6: IRNG
 8: SENS

item #



- 选择2: OVA

<OVA>
 OVER ACCEL (rev/s²)
 T : 75000
 R1: 75000
 X : 75000
 R3: 75000
 Z : 75000
 R2: 75000
 6 : 75000
 R4: 75000

- OVA 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

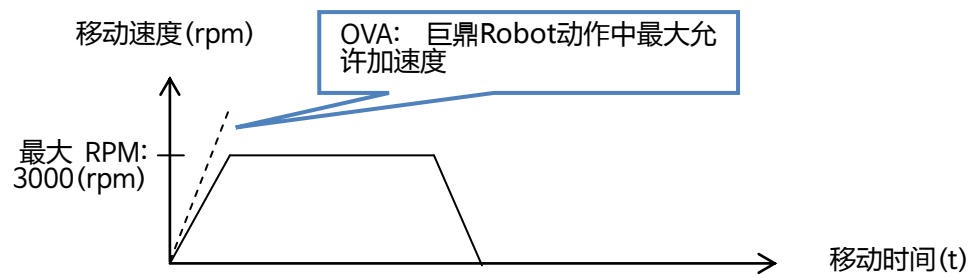
- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 最大允许加速度 (rev/s²)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 例子



※ 注意 ※

- 确认以下事项时，为了保证完美的移动角度，请修正最大允许加速度。
 - ① 电机编码器是否有异常
 - ② 机械部连接电缆状态
 - ③ 连续的补间动作中Robot移动到下一个位置时，发生急剧的方向移动

2.3.2.4.3 IPA (In Position Amount)

- 设置目标位置误差允许范围.

Step 1.

IPA 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA 4: IPE
 5: IPO 6: IRNG
 7: URNG 8: SENS

item #



- 选择3: IPA

<IPA>
 INPOS (1pulse)
 T : 333 Z : 1310
 R1: 386 R2: 386
 X : 655 6 : 200
 R3: 100 R4: 100

- IPA 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

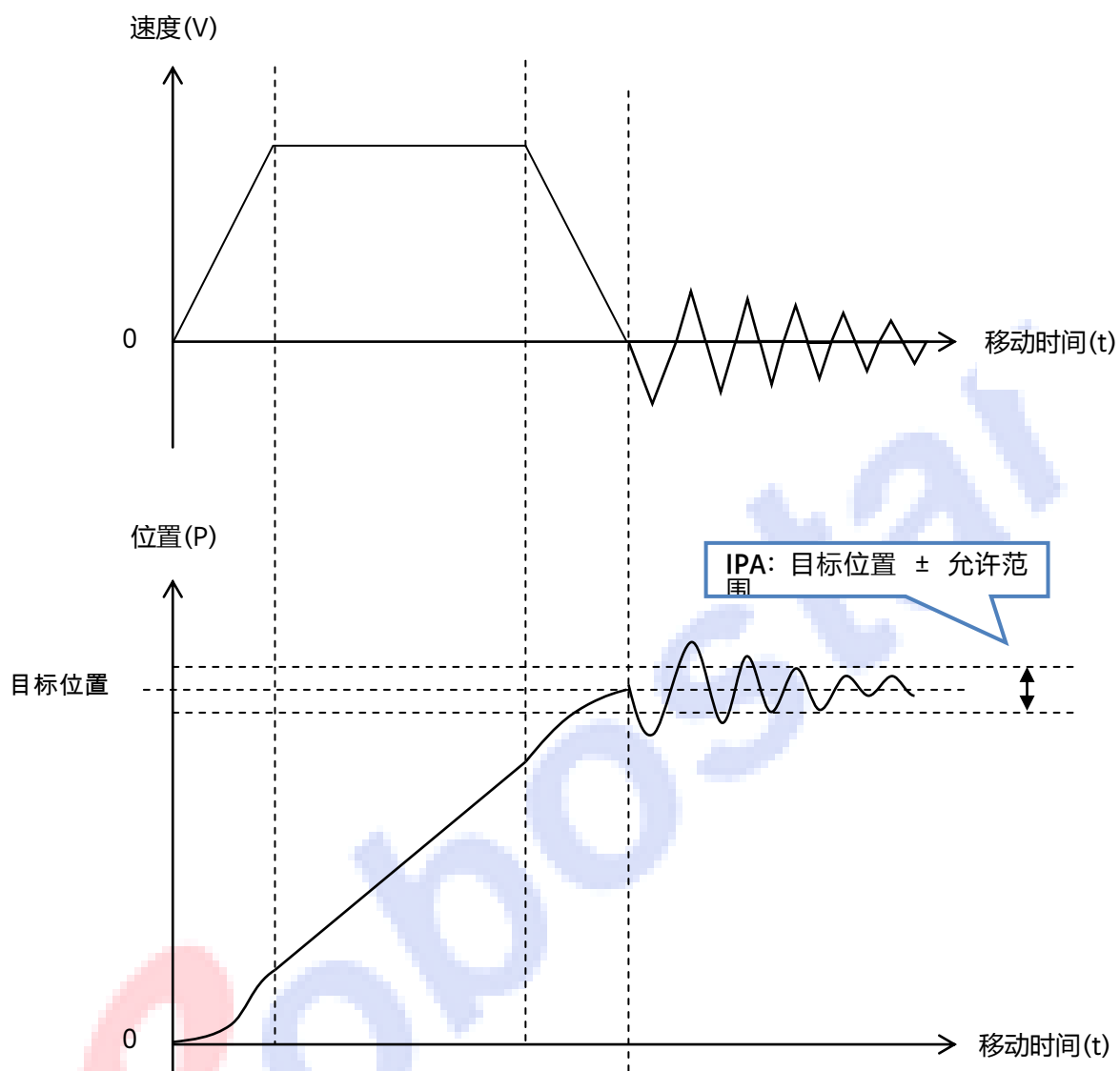
- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 目标位置误差允许范围 (pulse)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 例子
 - 1) 目标位置上的±误差允许范围设置为IPA 值.
 - 2) 下图显示 速度-时间, 位置-时间图表中的IPA 例子.



※ 注意 ※

- In Position Error在 IPE, IPA, IPO 允许值以上机械部晃动时发生。请确认以下状况。
 - ① 确认机械部组装状态是否正确
 - ② 确认负荷量是否在额定有效负荷内

2.3.2.4.4 IPE (In Position Error), IPO (In Position On-time)

- 设置到达目标位置的Error允许时间最大值 (IPE), 到达目标位置判定所需最小维持时间 (IPO).

Step 1.

IPE 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA **4: IPE**
 5: IPO 6: IRNG
 7: URNG 8: SENS

item #



- 选择4: IPE

<IPE>
 INPOS ERROR (1ms)
 T : 200 Z : 200
 R1: 200 R2: 200
 X : 200 6 : 200
 R3: 200 R4: 200

- IPE 画面确认

Step 1.

IPO 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA 4: IPE
5: IPO 6: IRNG
 7: URNG 8: SENS

item #



- 选择5: IPO

<IPE>
 INPOS ON Time (1ms)
 T : 2 Z : 2
 R1: 2 R2: 2
 X : 2 6 : 2
 R3: 2 R4: 2

- IPO 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明

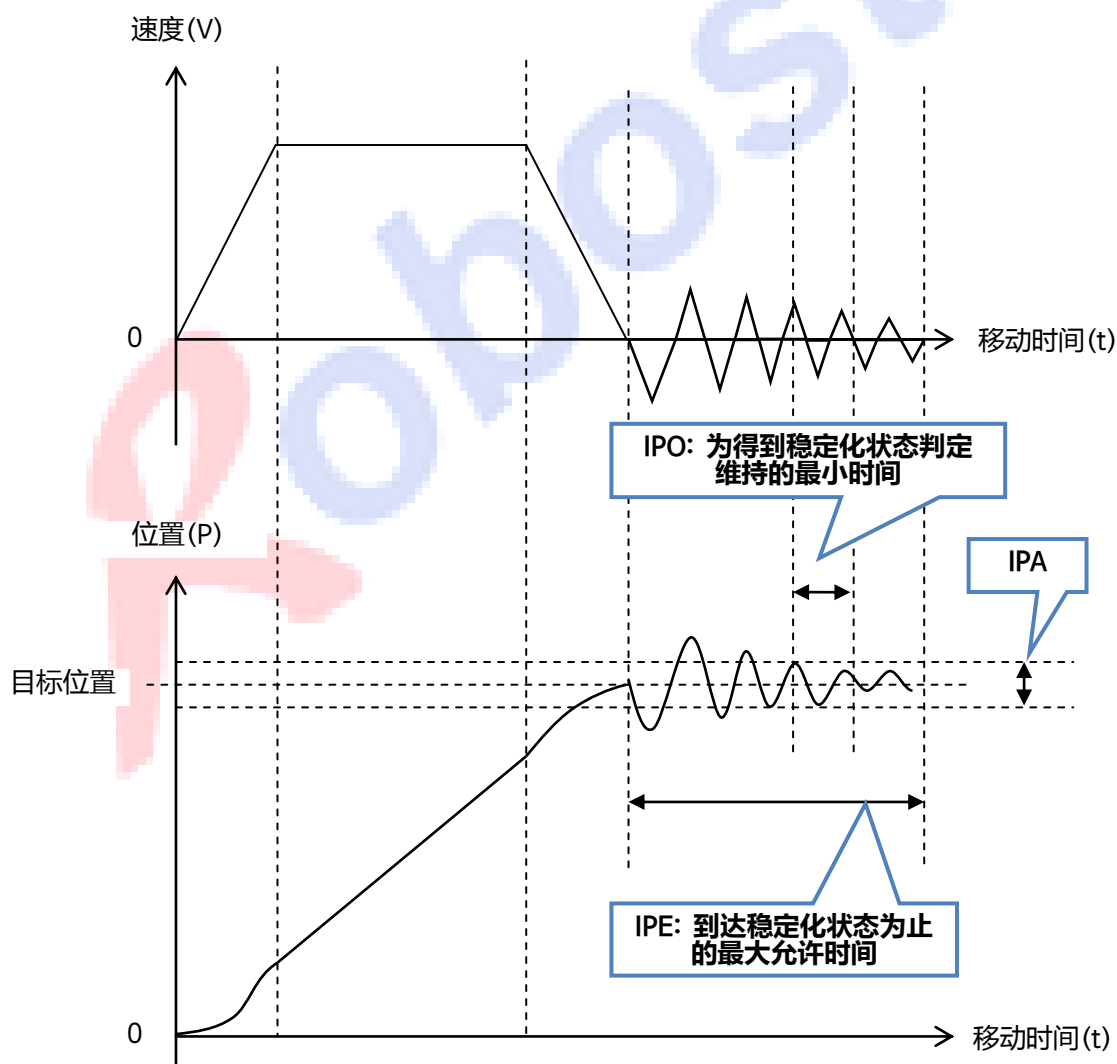
- IPE: 到达目标位置的Error允许时间最大值(ms)
- IPO: 到达目标位置判定所需最小维持时间 (ms)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法。
- ② 设定值超过最大范围或未达到最小范围时输出错误消息。

■ 参考事项

- ① 例子
 - 1) IPE 值设置到达稳定化状态为止的最大允许时间。
 - 2) IPO 值设置为得到稳定化状态判定维持的最小时间。
 - 3) 下图中显示了速度-时间，位置-时间图表中的IPE, IPO 例子。



※ 注意 ※

- In Position Error在 IPE, IPA, IPO 允许值以上机械部晃动时发生。请确认以下状况。
 - ① 确认机械部组装状态是否正确
 - ② 确认负荷量是否在额定有效负荷内

2.3.2.4.5 IRNG (In Range)

- 设置 In Range 范围或者决定 In Range 警报使用与否。

■ 详细项目

GROUP	内容	范围	设置值
1: IN RANGE	按各轴设置 In Range范围。	-999,999 ~ 999,999	
2: ALARM	按各轴决定 In Range警报使用与否，设置对象轴的速度。	参考说明	
3: JOG_SF	设置功能使用与否以及轴、对象轴的速度。	参考说明	

■ OPEN 顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 4: PROTECT -> 6: IRNG

Step 1.

IRNG 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA 4: IPE
 5: IPO **6: IRNG**
 7: URNG 8: SENS

item #



- 6: IRNG 选择

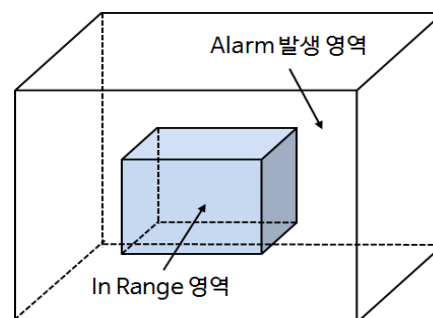
<IRNG>
 1. RANGE 2. ALARM
 3. JOG_SF

item #

- IRNG 画面确认

■ In Range 领域是?

- 指的是允许机器人机械部移动的领域，可以在参数窗编辑。机器人机械部移动到设置的 IN RANGE 参数领域之外的位置时，发生 IN RANGE 警报。



1) RANGE

- 设置 In Range 范围。

Step 1.

IN RANGE 画面移动

⟨IRNG⟩
1. RANGE 2. ALARM
3. JOG_SF

item #



- 1. RANGE 选择

⟨IRNG⟩ IN RANGE

T : 0	0
Z : 0	0
R1 : 0	0
R2 : 0	0
X : 0	0
6 : 0	0

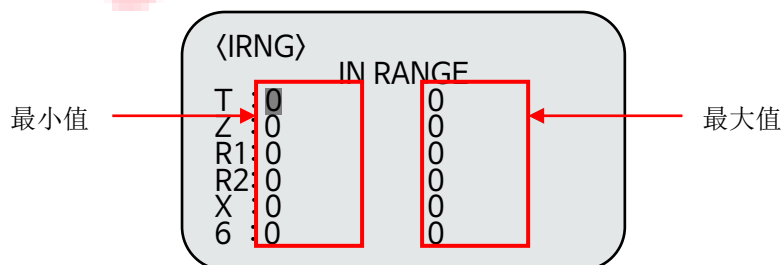
- IN RANGE 设置画面确认

■ 光标位置移动

- ① 请参考 [2.5.1.3 参数设置的](#) 光标位置移动以及页数变更。

■ 设置方法

- ① 请参考 [2.5.1.3 参数设置的](#) PARAMETER 值变更。
- ② 设置值说明
 - 允许范围值 (回转轴: degree, 直线移动轴: mm (可变更XY 坐标的机器人是 mm))
 - 左侧: 最小值, 右侧: 最大值



■ 保存以及取消方法

- ① 请参考 [2.5.1.3 参数设置的](#) 参数值保存以及取消方法。
- ② 设置值超过最大范围或者未达到最小范围时, 输出错误信息。

■ 参考事项

- ① 可以变更XY Coordinate 的机器人，用 X, Y, Z 值设置 Range。
- ② 机器人的所有轴都在 In Range 设置值范围内时才可以输出 I/O。

Robostar

2) ALARM

- 按各轴决定 In Range警报使用与否，设置对象轴的速度。

■ 详细项目

GROUP		内容	备注
JOG_SF	AXIS	按各轴决定 In Range 警报使用与否。在 In Range范围内决定各轴的限制速度。	
	VEL	在 In Range范围内决定各轴的限制速度。	
	DIST	在 In Range范围内决定各轴的限制移动距离。	

■ 功能设置

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 4: PROTECT -> 6:IRNG -> 2: ALARM

Step 1.

ALARM 画面移动

<IRNG>
1. RANGE
3. JOG_SF

2. ALARM

item #



- 2. ALARM 选择

<INRAG-ALARM>
1. AXIS 2. VEL
3. DIST

item #

- ALARM 设置画面确认

■ 光标位置移动

- 请参考 [2.5.1.3 参数设置](#)的光标位置移动以及页数变更。

(1) AXIS

- 设置值说明
 - ON: 目前 In Range 警报在动作。
 - OFF: 目前 In Range 警报不在动作。

Step 1.

ALARM 设置方法

〈ALM〉	
IN	RANGE ALARM
T : ☒ ON	T : OFF
R1: OFF	R2: OFF
X : OFF	6 : OFF
R3: OFF	R4: OFF

〈ALM〉	
IN	RANGE ALARM
T : ☐ OFF	T : OFF
R1: OFF	R2: OFF
X : OFF	6 : OFF
R3: OFF	R4: OFF

ENTER

- 点击"ENTER" 键, ON, OFF 会翻转 (ON -> OFF, OFF -> ON)

■ 保存以及取消方法

- ① 请参考[2.5.1.3 参数设置的](#) 参数值保存及取消方法。

(2) VEL (Velocity)

- 在 In Range 范围内决定各轴的限制速度。

Step 1.

ALARM 画面移动

<IRNG>
1. RANGE 2. ALARM
3. VEL 4. DIST

item #



- 3. VEL 选择

<IN RANGE VEL>
VELOCITY (rpm)
T : 1 T : 1
R1: 1 R2: 1
X : 1 6 : 1
R3: 1 R4: 1

- VEL VELOCITY 设置画面确认

■ 光标位置移动

- ① 请参考[2.5.1.3 参数设置的](#)光标位置移动以及页数变更。

■ 设置方法

- ① 请参考[2.5.1.3 参数设置的](#)PARAMETER 值变更。
- ② 设置值说明
 - 极限速度值 (rpm)

■ 保存以及取消方法

- ① 请参考[2.5.1.3 参数设置的](#)参数值保存以及取消方法。
- ② 设置值超过最大范围或者未达到最小范围时，输出错误信息。

(3) DIST (Distance)

- 在 In Range范围内决定各轴的限制移动距离。

Step 1.

ALARM 画面移动

<IRNG>
1. RANGE 2. ALARM
3. VEL 4. DIST

item #



- 4. DIST 选择

<IN RANGE DIST>
DISTANCE(mm)
T : 0.10 T : 1
R1: 1 R2: 1
X : 1 6 : 1
R3: 1 R4: 1

- DISTANCE 设置画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设置的](#)光标位置移动以及页数变更。

■ 设置方法

- 请参考[2.5.1.3参数设置的](#)PARAMETER 值变更。
- 设置值说明
 - 限制移动距离(mm)

■ 保存以及取消方法

- 请参考[2.5.1.3 参数设置的](#)参数值保存以及取消方法。
- 设置值超过最大范围或者未达到最小范围时，输出错误信息。

3) JOG_SF 设置

- 设置功能使用与否以及轴、对象轴的速度。
- 只有使用 JOG 功能时才适用的功能，掌握Hand 位置信息后 Hand的移动距离超过一定的距离时，为了防止由于移动到 X 轴的原因造成Hand 破损等事例，对象轴的速度自动变更且设置为非常低速的功能。
- Hand 距离信息请参考 IN RANGE(PROTECT-IRNG) 值。

■ 详细项目

GROUP		内容	备注
JOG_SF	ENABLE	设置 Jog Special Function 功能使用与否。	
	AXIS	设置适用 Jog Special Function 的对象或者Hand 轴。	
	VEL	限制对象轴的速度。	

■ 功能设置

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 4: PROTECT -> 6:IRNG -> 3: JOG_SF

Step 1.

JOG_SF 画面移动

<IRNG>
1. RANGE 2. ALARM
3. JOG_SF

item #



- 3. JOG_SF 选择

<JOG SPECIAL FUNC>
1.ENABLE 2.AXIS
3. VEL

item #

- JOG_SF 设置画面确认

(1) ENABLE

- 设置Jog Special Function 功能的使用情况.

Step 1.

ENABLE 画面移动

〈JOG SPECIAL FUNC〉
1.ENABLE 2.AXIS
 3. VEL

item #



- 选择1: ENABLE

〈ENABLE〉
 VELOCITY LIMIT
 ENABLE: ON

- ENABLE 画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

① 设定值说明

- ON: 当前 In Range Alarm在动作, OFF: 当前 In Range Alarm不在动作.

Step 1.

ALARM 设定方法

〈ENABLE〉
 VELOCITY LIMIT
 ENABLE: ON

〈ENABLE〉
 VELOCITY LIMIT
 ENABLE: OFF



- 按下"ENTER"键ON,OFF反转
- (ON -> OFF, OFF -> ON)

- 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

Robostar

(2) AXIS

- 设置应用Jog Special Function 的目标或Hand轴.

Step 1.

AXIS 画面移动

〈JOG SPECIAL FUNC〉
1. ENABLE 2. AXIS
3. VEL

item #



- 选择2: AXIS

〈ENABLE〉
VELOCITY LIMIT
ENABLE: ON

- ENABLE 画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

① 设定值说明

- HAND 轴: 设置确认位置信息的 HAND 轴 (例: R1, R2 轴)
- TARGET轴: HAND轴超出限定距离时设置限定速度的轴(例:T,X 轴)

Step 1.

AXIS 设定方法

〈AXIS〉
SET AXIS
T1: TARGET Z1: NONE
R1: HAND R2: HAND
X1: TARGET 6 : NONE
R3: NONE R4: NONE



- 按下"ENTER" 键按照NONE, TARGET, HAND 的顺序变更
- (NONE-)TARGET-)HAND-)NONE)

〈AXIS〉
SET AXIS
~~T1: TARGET~~ Z : NONE
~~R1: HAND~~ R2: HAND
~~X1: TARGET~~ 6 : NONE
R3: NONE R4: NONE

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

(3) VEL

- 限定Target轴的速度.

Step 1.

VEL 画面移动

〈JOG SPECIAL FUNC〉
1.ENABLE 2.AXIS
3. VEL

item #



- 选择3: VEL

〈TARGET MOVING VEL〉
VELOCITY (%)
T1: 1 Z1: 0
R1: 0 R2: 0
X1: 1 61: 0
R3: 0 R4: 0

- VEL 画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.

① 设定值说明

- JOG 速度的% 值 (HAND 轴超出限定范围时, TARGET 轴的速度限定为如下画面中设定的值.)

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

① 设定值超出最大范围或未达到最小范围时输出错误信息.

2.3.2.4.6 URNG (User In Range)

- 设置用户是否使用 In Range, 范围, 输出接点, Z轴允许范围等.
- User Range 可以保存1~20 使用.

■ 详细项

GROUP	内容	范围	设定值
1: DISABLE	设置是否使用In Range.	参照说明	
2: RANG	设置User In Range 范围.	-999,999 ~ 999,999	
4: OUT	选择I/O 输出接点.	0 ~ 320	
4: ZDTA	设置用户Z轴移动单位值.	-1.0 ~ 1,000	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 4: PROTECT -> 7: URNG

Step 1.

URNG 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA 4: IPE
 5: IPO 6: IRNG
 7: **URNG** 8: SENS

item #



- 选择7: URNG

<URNG (1/2)>
 1: URNG1 2: URNG2
 3: URNG1 4: URNG1
 5: URNG1 6: URNG1
 7: URNG1 8: URNG1
 9: URNG1 A: URNG1

item #

- URNG 画面确认

Step 2.

选择编辑区域

<URNG (1/2)>
 1: URNG1 2: URNG2
 3: URNG1 4: URNG1
 5: URNG1 6: URNG1
 7: URNG1 8: URNG1
 9: URNG1 A: URNG1

item #



- 可以利用“PAGE UP/DOWN” 键切换画面

<URNG (1/2)>

1: URNG1 2: URNG2
3: URNG1 4: URNG1
5: URNG1 6: URNG1
7: URNG1 8: URNG1
9: URNG1 A: URNG1

item #



~



- 选择需要编辑的区域

Step 3.

画面确认 (URNG1 例子)

<URNG1>
USER RANGE1
ENABLE : ☐ OFF

- DISABLE 状态画面确认

<URNG1>
USER RANGE1
ENABLE : ☒ ON

- ENABLE 状态画面确认

RANG OUT ZDTA

1) ENABLE

- 设置是否使用User In Range.

Step 1.

ENABLE 画面确认 (URNG1 例子)

〈URNG1〉
USER RANGE1
ENABLE : ☐ OFF

- DISABLE 状态画面确认

〈URNG1〉
USER RANGE1
ENABLE : ☒ ON

- ENABLE 状态画面确认

RANG OUT ZDTA

■ 设定方法

② 设定值说明

- ON: 使用User In Range(#).
- OFF: 不使用User In Range(#).

Step 1.

ENABLE 设定方法 (URNG1 例子)

〈URNG1〉
USER RANGE1
ENABLE : ☐ OFF



- 按下“ENTER”键反转 ON, OFF
(ON → OFF, OFF → ON)

〈URNG1〉
USER RANGE1
ENABLE : ☒ ON

RANG OUT ZDTA

■ 保存及取消方法

- ① 请参考 [2.5.1.3 参数设定](#) 参数值保存及取消方法.

2) RANG (Range)

- 设置User In Range 范围.

Step 1.

RANG 画面移动 (URNG1 例子)

〈URNG1〉
USER RANGE1
ENABLE : ☒ ON



- 按下"F1" 键移动到 RANG 画面

RANG OUT ZDTA

〈URNG1〉
USER RANGE 1
T : 0 0
Z : 0 0
R1 : 0 0
R2 : 0 0
X : 0 0
6 : 0 0

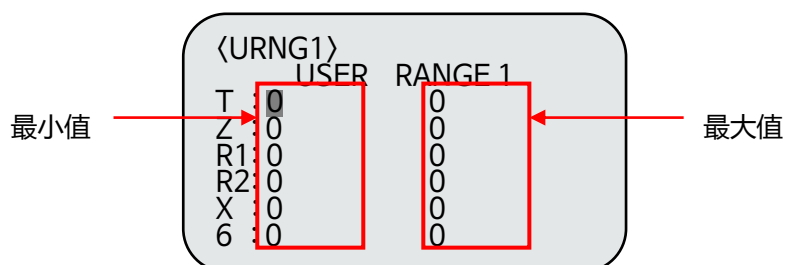
- RANG 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 允许的范围值(旋转轴: degree, 直线移动轴: mm (XY坐标可变换的Robot为 mm))
 - 左侧: 左小值, 右侧: 最大值



■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 可以变换XY Coordinate 对的Robot利用X, Y, Z 值设定 Range.
- ② Robot所有轴在User In Range 设定值范围内时才能输出I/O. (?)
- ③ 即使Robot超出User In Range 范围, 如果不在In Range 范围内就不会发生[Error Code: 1439 "IN RANGE Error"] Alarm.

3) OUT

- 选择I/O 输出接点.

Step 1.

OUT 画面移动 (URNG1 例子)

〈URNG1〉
USER RANGE1
ENABLE: ☐ ON

RANG **OUT** ZDTA

〈URNG1〉
INDEX: ☐ OUTPUT

F2

- 按下"F2" 键移动到 OUT 画面

- OUT 画面确认

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 用户定义的DIO编号 (-1~127)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① 输入User In Range 范围 IN/OUT 的 User IO Output 编号.

4) ZDTA (Z Delta Range)

- 设定用户 Z轴移动单位值.

Step 1.

ZDTA 画面移动 (URNG1 例子)

⟨URNG1⟩
USER RANGE1
ENABLE: ☒ ON

RANG OUT **ZDTA**

⟨URNG1⟩
Z DELTA (mm)
DELTA: 0



- 按下"F3" 键移动到 ZDTA 画面

- ZDTA 画面确认

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Z轴移动单位制(mm)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

■ 参考事项

- ① Z轴一次可移动的距离值.
- ② 设置JOG & AUTO 动作中一次可移动的距离, 限制连续移动时的移动量.
- ③ 即使Robot在User In Range 范围内, 当Z轴移动距离比设定值大时发生 **[Error Code: 1439 "IN RANGE Error"]** Alarm.

2.3.2.4.7 SENS (H/W Limit Sensor)

- 设置各轴Limit传感器 (H/W Limit Sensor) 使用情况和接点.

Step 1.

SENS 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA 4: IPE
 5: IPO 6: IRNG
 7: URNG **8: SENS**

item #



- 选择8: SENS

<SENS>

HW	Limit	Sensor	num
T	+: 1		-: 0
Z	+: 0		-: 0
R1	+: 0		-: 0
R2	+: 0		-: 0
X	+: 0		-: 0
6	+: 0		-: 0

- SENS 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 不使用Limit传感器时: -1
 - 使用Limit传感器时: 各轴Limit传感器DIO 编号 (0~64)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.2.4.8 BELT (Timing Belt)

- 决定Timing Belt 有/无确认用传感器使用情况及接点.

Step 1.

BELT 画面移动

<PROTECT>
 1: OVS 2: OVA
 3: IPA 4: IPE
 5: IPO 6: IRNG
 7: URNG 8: SENS
9: BELT

item #



- 选择9: BELT

<BELT SENS>
 BELT CHECK SENS NO
 ENABLE : ☐ OFF

- BELT 画面确认

Step 2.

ENABLE 设定

<BELT SENS>
 BELT CHECK SENS NO
 ENABLE : ☐ ON
 NUM1 : 0
 NUM2 : 0



- 使用"ENTER" 键选择使用情况.
(ON: 使用, OFF: 未使用)

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.

① 设定值说明

- 不使用Timing Belt确认传感器: -1
- 使用Timing Belt确认传感器: 各轴Limit传感器DIO 编号 (0~64) 中输入

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

① 设定值超出最大范围或未达到最小范围时输出错误信息.

2.3.2.5 SERVO

- 设置Servo电机和Driver相关参数.

■ 详细项

GROUP	内容	范围	设定值
1: ID	设置Driver ID.	参照说明	
2: M-TURN	初始化电机编码器的 MULTI-TURN DATA.	参照说明	
3: DRIVER	设置Driver相关参数. (EtherCAT 通信类型 Only)	参照说明	
4: BOARD	设置控制器基板相关参数. (RS422 通信类型 Only)	参照说明	
5: DELAY	设置Quick stop Servo Off delay. (EtherCAT 通信类型 Only)	10~9999	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 5: SERVO

Step 1.

SERVO移动

〈ROBOT PARAMETER〉
 1: CONF 2: BODY
 3: MOTION 4: PROTECT
5: SERVO 6: ORIGIN
 7: COORD

input : #



- 选择5: SERVO

SERVO 确认设置画面

〈SERVO〉
 1. ID 2. M-TURN
 3. DRIVER 4. BOARD
 5. DELAY

item #

- SERVO 画面确认

2.3.2.5.1 ID (Driver ID)

- 设置Driver ID.

Step 1.

ID 画面移动

<SERVO>

1. ID	2. M-TURN
3. DRIVER	4. BOARD
5. DELAY	

item #

<DRIVER ID>

Driver	ID	Setting
T : 0		Z : 1
R1 : 2		R2 : 3
X : 4		6 : 5
R3 : 6		R4 : 7



- 选择1: ID

- ID 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 未使用时: 0
 - 使用时: Driver ID 值 (1~16)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.
- ③ 设置ID 时如果输入重复的ID, 则在Command 窗口中输出 "Already Used!" 信息. 请重新设置不重复的ID.

Step 1.

重复输入相同 ID 时

<DRIVER ID>	
Driver	ID Setting
T : 0	Z : 1
R1 : 2	R2 : 0
X : 4	6 : 5
R3 : 6	R4 : 7

Already Used!

<SERVO>	
1. ID	2. M-TURN
3. DRIVER	4. BOARD
5. DELAY	

item #

ENTER

ESC

- 设置ID 时如果输入相同ID 保存，在Command 窗口中输出"Already Used!" 信息

- 按下"ENTER" 或 "ESC"键恢复到之前画面后重新设置

参考事项

- ① 重新设置Driver ID后请重启Driver.

2.3.2.5.2 M-TURN (Multi-turn Clear)

- 初始化电机编码器的MULTI-TURN DATA.

Step 1.

MULTI-TURN CLEAR 画面移动

<SERVO>
1. ID
3. DRIVER
5. DELAY

2. M-TURN
4. BOARD

item #



- 选择2: MULTI-TURN CLEAR

MULTI TURN CLEAR 1/4

T Z R1 R2

- MULTI-TURN CLEAR 画面确认

■ 设定方法

- ① 选择与Multi-turn 数据初始化轴匹配的功能键 (F1~F4)进行初始化.

Step 1.

MULTI-TURN CLEAR 设定

MULTI TURN CLEAR 1/4

T Z R1 R2



- 按下"PAGE UP/DOWN" 键移动到初始化 Multi-turn Data 的轴画面

MULTI TURN CLEAR 2/4

X 6 R3 R4



~



- 按下画面中与轴对应的功能键(F1~F4) 选择

② Q1 轴选择 例子

Step 1.

选择Q1 轴

MULTI TURN CLEAR 3/4

Q1

V1

Q2

V2

F1

- 按下"F1" 键选择Q1 轴

选择是否初始化

MULTI TURN CLEAR

Q1 Clear? (ENT/ESC)

ENTER

ESC

- 按下"ESC" 键取消Multi-turn Clear 后
恢复到之前画面
- 按下"ENTER"键初始化相应轴编码器的
Multi-turn 数据

MULTI TURN CLEAR 3/4

Q1

V1

Q2

V2

- 确认恢复画面

2.3.2.5.3 DRIVER (Driver Parameter) (EtherCAT 通信类型 Only)

- 设置Driver相关参数. ()

Step 1.

DRIVER PARAMETER 画面移动

<SERVO>
1. ID 2. M-TURN
3. DRIVER
5. DELAY

item #



- 选择3: DRIVER

<AXIS>
Select axis
1: T 2: Z
3: R1 4: R2
5: X 6: 6

input: #

- DRIVER PARAMETER 画面确认

Step 2.

选择编辑轴

<AXIS>
Select axis
1: T 2: Z
3: R1 4: R2
5: X 6: 6

input: #



- 按下"PAGE UP/DOWN" 键移动到要编辑的轴所在画面

<AXIS>
Select axis
1: T **2: Z**
3: R1 **4: R2**
5: X **6: 6**

input: #



~



- 输入与轴对应的数字

选择轴初始画面确认 (T轴例子)

<**T**-ax [3000:00]> 000
For manufacturer's use

Range: 0-0
Value: 0

JUMP EXIT

- 确认T轴初始画面

Step 3.

无法设定时

<AXIS>
Select axis
1: T 2: Z
3: R1 4: R2
5: X 6: 6

axis is disable

- 1) 无法使用轴 (disable 状态) 或 2) Driver 的 ID 未设定时 Command 窗口中先输出 "axis is disable" 信息

<AXIS>
Select axis
1: T 2: Z
3: R1 4: R2
5: X 6: 6

input: #

ENTER

ESC

- 按下 "ESC" 键或 "ENTER" 键恢复到之前画面
- 确认状况后处理

■ 页面移动

- ① 可以按下 "PAGE UP/DOWN" 键移动或, 选择 JUMP ("F3" 键) 移动到所需页面.
- ② JUMP 使用方法

Step 1.

JUMP 使用方法

<T-ax [3000:00]> 000
For manufacturer's use

Range: 0-0
Value: 0

EXIT

- 按下 T/P 的 JUMP 键激活画面

<T-ax [3000:00]> 000
For manufacturer's use

Range: 0-0
Value: 0

para # = ■

- Command 窗口更新, 可以确认被激活且可以输入值

<T-ax [3000:00]> 000
For manufacturer's use

Range: 0-0
Value: 0

para # = ■



~



- 输入要移动的页面编号确认更新画面

Step 2.

输入错误时

<T-ax [3000:00]> 000
For manufacturer's use

Range: 0-0
Value: 0

out of bound

- "out of bound" 信息输出到 Command 窗口

<T-ax [3000:00]> 000
For manufacturer's use

Range: 0-0
Value: 0

para # = ■

ENTER

ESC

- 按下"ESC" 键或 "ENTER"键恢复到之前画面
- 重新输入

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 请参考MINAS-A5B Series Panasonic技术文档第9章 List of parameters (157 ~ 171pp.).
(No. SX-DSV02472, EtherCAT Communication Specifications)
- ③ 输入设定值
 - Teach Pendant 画面中输入Range 范围内的值. 但, Range 范围为 0-0的项表示无输入范围且无法保存.
 - 与Teach Pendant 的画面和 MINAS-A5B Series Panasonic技术文档第9章 List 相对应

〈Teach Pendant 画面 - T轴 例子〉	
轴信息	Class No. Index
	<T-ax [3000:00]> 000 For manufacturer's use
Range	Range: 0-0
输入 値	Value: 0
	EXIT
〈List Table〉	

Class	No.	Title	Unit	Range	Size [byte]	Function / Contents
0	00	For manufacturer's use	—	—	2	Permanently set at 1.
						Select the control mode of the servo driver. 0: semi-closed control (position/velocity/torque control, selectable) 1-5: To be used by the manufacturer but not

④ 例外或错误事项

- 发生例外或错误事项，Command 窗口中输出下面的信息时，请按 "ENTER" 键或"ESC" 键恢复到之前画面后重新尝试。

Case 1.

Parameter Read 失败时

```

<T-ax [3000:00]> 000
Number of output
pulses per motor
revolution
Range:
Value: Comm Error!
JUMP EXIT

```

- 不输出Range 及 Value 值，画面中输出 "Comm Error!" 信息

Case2.

画面移动时Data Load 等待

```

<T-ax [3000:00]> 000
Number of output
pulses per motor
revolution
Range: 1-262144
Value: 2500
plz wait for a while

```

- 页面移动或 JUMP时Driver数据加载需要时间时，Command 窗口中输出 "plz wait for a while"信息

Case 3.

输入超出范围的值时

```

<T-ax [3000:00]> 000
Number of output
pulses per motor
revolution
Range: 1-262144
Value: 999999
wrong data!!

```

- 输入超出范围(Range 值)的值时，Command 窗口中输出 "wrong data!!" 信息

■ 保存及结束方法

① 保存方法 (SAVE)

Step 1.

SAVE 选择

<T-ax [3000:00]> 000
 Number of output
 pulses per motor
 revolution
 Range: 1-262144
 Value: 2500

SAVE JUMP EXIT



- 按下"F1" 键进行 SAVE

确认保存等待信息

<T-ax [3000:00]> 000
 Number of output
 pulses per motor
 revolution
 Range: 1-262144
 Value: 2500

plz wait for a while

- 如果保存无异常，则在画面中输出"plz wait for a while" 信息。
- 进行迅速时可能无法用肉眼确认

保存完成 (reboot 请求)

<T-ax [3000:00]> 000
 Number of output
 pulses per motor
 revolution
 Range: 1-262144
 Value: 2500

Save OK. plz reboot!



- 完成保存后按下 "ESC" 或 "ENTER" 键
恢复到之前画面

保存完成 (reboot 未请求)

<T-ax [3000:00]> 000
 Number of output
 pulses per motor
 revolution
 Range: 1-262144
 Value: 2500

Save OK!!



Step 2.

保存失败时

<T-ax [3000:00]> 000
 Number of output
 pulses per motor
 revolution
 Range: 1-262144
 Value: 2500

Save Error!!

- 由于系统错误等原因保存失败时，
Command 窗口中输出 "Save Error!!"
信息

② 结束方法 (EXIT)

- 按下"ESC" 键或者EXIT移动到上一级画面.

Step 1.

结束 (ESC)

<T-ax [3000:00]> 000
Number of output
pulses per motor
revolution
Range: 1-262144
Value: 2500

SAVE JUMP EXIT

ESC

- 按下"ESC" 键执行结束

结束 (选择EXIT)

<T-ax [3000:00]> 000
Number of output
pulses per motor
revolution
Range: 1-262144
Value: 2500

SAVE JUMP EXIT

F4

- 按下"F4" 键进行EXIT

恢复到上一级画面

<AXIS>
Select axis
1: T 2: Z
3: R1 4: R2
5: X 6: 6
input: #

- 确认恢复的上一级画面

■ 参考事项

- ① 保存时如果确认重启请求信息时, 请务必重启Driver.

2.3.2.5.4 BOARD (Board Parameter) (RS422 通信类型 Only)

- 设置控制器基板相关参数.

■ 详细项

GROUP	内容	范围	设定值
1: WATT	设置电机容量.		
2: OND	设置Brake ON delay.		
3: OFD	设置Brake OFF delay.		
4: SENS	设置各轴Limit传感器方向.		
5: GAIN	设置各轴Gain值.		
6: TQR	设置Torque Limit.		
7: HWL	设置是否使用Limit 传感器.		

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 5: SERVO-> 4: BOARD

Step 1.

BOARD PARAMETER 画面移动

<SERVO>
 1. ID 2. M-TURN
 3. DRIVER 4. BOARD
 5. DELAY



- 选择4: BOARD

<MOTOR>
 1: WATT 2: OND
 3: OFD 4: SESN
 5: GAIN 6: TQR
 7. HWL

input: #

- MOTOR PARAMETER 画面确认

1) WATT

- 设置电机容量.

Step 1.

Motor watt 画面移动

<MOTOR>
 1: WATT 2: OND
 3: OFD 4: SESN
 5: GAIN 6: TQR
 7: HWL

item #



- 1. WATT 选择

<MOTOR>
 Motor [Watt]
 1 : 103 2 : 103
 3 : 103 4 : 103
 5 : 103 6 : 103

- Motor Watt 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.

- ② 设定值说明

- 请参考标签内容中‘ENC TYPE’.

ABS : Absolute Type Encoder

INC : Incremental Type Encoder

- PANASONIC MINAS MOTOR A4 SERIES

Model	Watt	设定值	ENCODER PULSE	BRAKE	ENC TYPE
MSMD5AZS1S	50W	53	131072	without	ABSOLUTE 17BIT
MSMD5AZS1T	50W	53	131072	with	
MSMD012S1S	100W	103	131072	without	
MSMD012S1T	100W	103	131072	with	
MSMD022S1S	200W	203	131072	without	
MSMD022S1T	200W	203	131072	with	
MSMD042S1S	400W	403	131072	without	

MSMD042S1T	400W	403	131072	with	
MSMD082S1S	750W	803	131072	without	
MSMD082S1T	750W	803	131072	with	
MSMA102S1G	1000W	1003	131072	without	
MSMA102S1H	1000W	1003	131072	with	
MSMD012P1S	100W	101	10000	without	INCREMENTAL 2500 PULSE
MSMD012P1T	100W	101	10000	with	
MSMD022P1S	200W	201	10000	without	
MSMD022P1T	200W	201	10000	with	
MSMD042P1S	400W	401	10000	without	
MSMD042P1T	400W	401	10000	with	
MSMD082P1T	750W	801	10000	without	
MSMD082P1T	750W	801	10000	with	

• SEWOO INDUSTRIAL SYSTEM

Model	Watt	设定值	ENCODER PULSE	DIGITAL ENCODER	ENC TYPE
PS20C	400W	402	96000	HEIDENHAIN 1um	LINEAR
PM40C	750W	702	240000	HEIDENHAIN 1um	
TM20C	750W	802	120000	HEIDENHAIN 1um	

• SWL-PL30C 电机ID 附加内容

Model	Motor ID	
	1um	20um
	设定值	设定值
SWL-PM10C	402	422
SWL-PM20C	502	522
SWL-PM30C	602	622
SWL-PM40C	702	722
SWL-PL10C	902	922
SWL-PL20C	1002	1022

■ 保存及取消方法

- ① 请参考 [2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2) OND (ON delay)

- 设置Brake ON delay.

Step 1.

ON delay 画面移动

<MOTOR>
 1: WATT
 3: OFD
 5: GAIN
 7: HWL
 2: **OND**
 4: SESN
 6: TQR

item #



- 选择2. OND

<OND>
 BK ON DELAY(msec)
 1 : 15
 3 : 15
 5 : 15
 2 : 200
 4 : 15
 6 : 15

- Brake ON delay 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 设置Brake ON 信号输出时间点和Servo OFF 输出时间点之间的差
 - Brake ON/OFF 时间延迟单位 : msec
 - Servo ON 信号发生时Brake OFF 信号/ ServoOFF 信号发生时 Brake ON 信号从控制器同时输出.
 - Brake ON/OFF具有机械性因素, 比 Servo ON/OFF信号有延迟
 - 为了弥补Brake ON/OFF信号的延迟, 可以在 Brake ON/OFF信号中设置延迟 (OND,OFD)..

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

3) OFD (OFF delay)

- 设置Brake OFF delay.

Step 1.

OFF delay 画面移动

<MOTOR>
 1: WATT 2: OND
 3: **OFD** 4: SESN
 5: GAIN 6: TQR
 7: HWL

item #



- 选择3. OFD

<OND>
 BK OFF DELAY(msec)
 1 : 5 2 : 200
 3 : 5 4 : 5
 5 : 5 6 : 5

- Brake OFF delay 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 设置Brake ON 信号输出时间点和Servo OFF 输出时间点之间的差
 - Brake ON/OFF 时间延迟单位 : msec
 - Servo ON 信号发生时Brake OFF 信号/ ServoOFF 信号发生时 Brake ON 信号从控制器同时输出.
 - Brake ON/OFF具有机械性因素, 比 Servo ON/OFF信号有延迟
 - 为了弥补Brake ON/OFF信号的延迟, 可以在 Brake ON/OFF信号中设置延迟 (OND,OFD).

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

4) SENS (Sensor Limit)

- 设置各轴Limit传感器方向.

Step 1.

SENSOR Limit 画面移动

<MOTOR>
 1: WATT 2: OND
 3: OFD 4: **SESN**
 5: GAIN 6: TQR
 7: HWL

item #



- 选择4. SENS

<SENS>
 HW limit sensor dir
 1 :- 2 :-
 3 :- 4 :-
 5 :- 6 :-

- HW Limit sensor 方向设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 按下[ENTER]key循环变更 (+ → - → +)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

5) GAIN

- 设置各轴电机GAIN值.

Step 1.

GAIN 画面移动

<MOTOR>
 1: WATT 2: OND
 3: OFD 4: SESN
 5: **GAIN** 6: TQR
 7: HWL

item #



- 选择5. GAIN

<GAIN>
 1 : PVG



- 选择1. PVC

<PVG>
 POS/VEL GAIN:1
 Kpp: 35 Kpf: 0
 Kv: 150 Kvi: 150
 Kv: 150 IR: 3

1 2 3 4

- Position Velocity Gain 设定画面确认

Step 2.

轴移动

<PVG>
 POS/VEL GAIN:1
 Kpp: 35 Kpf: 0
 Kv: 150 Kvi: 150
 Kv: 150 IR: 3

1 2 3 4



- 1轴: F1, 2轴: F2, 3轴: F3, 4轴: F4



- 5轴: F1, 6轴: F2

Cursor位置移动

- 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

设定方法

- 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- 设定值说明
 - Kpp: 位置比例Gain (Position Proportional Gain)

③ 决定控制系统的动特性（固有振动数的控制变量，根据机械部的形象、有效负载重量不同，作为机械部整体系统利益，根据Robot形态不同

- 设定范围：1.0 ~ 100 [Hz]（建议以2为单位调整）
- 调整范围：30 ~ 40（初始值：35）

• Kpf：速度控制器转向补偿Gain (Feedforward Gain)

③ 决定对于加减速位置指令的追踪特性.

④ Kpf 值小时，整体动作时间变慢。因此，需要减少循环时间时调整.

⑤ 增加设定值时，机械部可能发生振动或噪音

- ➔ 设定范围：0.0 ~ 100 [%]（建议以5为单位调整）
- ➔ 调整范围：30 ~ 60（初始值：0）

• Kvf：电流控制器转向补偿Gain (Feedforward Gain)

③ 除发生严重振动以外，无需变更

- 设定范围：1.0 ~ 200 [Hz]（建议以10 为单位调整）
- 调整范围：100 ~ 150（初始值：150）

• Kvi：速度积分Gain (Velocity Intergral Gain)

补偿速度控制器正常状态的误差的值，根据电机容量及特性决定

- 设定范围：1.0 ~ 200 [Hz]（建议以10 为单位调整）
- 调整范围：100 ~ 150（初始值：150）

• Kvp：速度比例Gain (Velocity Proportional Gain)

③ 根据速度控制器追踪特性值决定

④ 增加Kvp 值时：能改善电机的驱动扭矩及动特性，但会发生振动

⑤ 减少Kvp 值时：电机的驱动扭矩变小，动特性变慢

- ➔ 设定范围：1.0 ~ 200 [Hz]（建议以10为单位调整）
- ➔ 调整范围：100 ~ 150（初始值：150）

• IR：惯性比 (Inertia Ratio)

③ 电机惯性 (Motor Interia) 对符合惯性的比例，惯性比对于Kvp, Kvi, Kvf时比例常数.

⑥ 惯性比 = (负荷惯性+ 电机惯性)/电机惯性

→ 设定范围：1.0 ~ 50 (建议以2.0 为单位调整)

→ 调整范围：1.0 ~ 30 (初始值：2.5)

• Gain Tunning顺序

Kpp	35	Kpf	60
Kvf	150	Kvi	150
Kvp	150	IR	2.5

1. 上述初始值或用户在Robot上安装Tool导致Robot配置与出厂状态不同，或者Robot安装环境从大气状态下变为真空状态产生振动时，将IR上升2单位，试图从低速驱动到高速。

2. 第1步骤过程中因惯性比(IR) 上升，振动或噪音消失后，也以2.0为单位增加，将其设置为比噪音开始发生的IR 值小 1.0~2.0 左右。

3. Kpp从35上升2单位，从低速驱动到高速，将其设置为比噪音开始发生的Kpp 值小1.0~2.0 左右。

4. 为了提高整体应答性， Kpf设置为 60%

■ **保存及取消方法**

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法。
- ② 设定值超过最大范围或未达到最小范围时输出错误消息。

6) TRQ (Torque)

- 设置Torque Limit

Step 1.

Torque 画面移动

<MOTOR>
 1: WATT 2: OND
 3: OFD 4: SESN
 5: GAIN **6: TQR**
 7: HWL

item #



- 选择6. TRQ

<TRQ>
 TRQ LIMIT [%]
 1 : 300 2 : 300
 3 : 300 4 : 300
 5 : 300 6 : 300

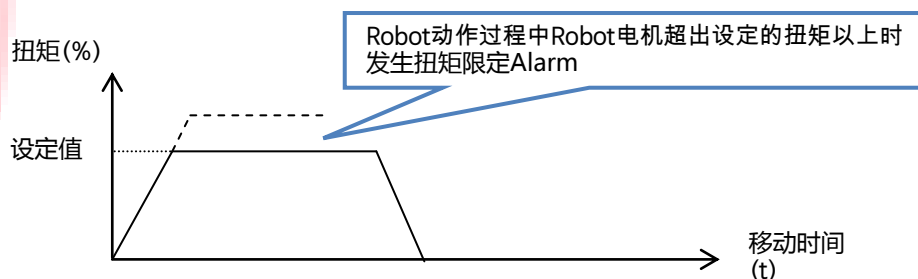
- Torque Limit 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 出厂设定值为 '300'
 - 设定值 : 1...300
 - 为了应对外部冲击的敏感度, 降低 Torque值再使用.



■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

7) HWL(Limit sensor)

- 设置是否使用HW Limit sensor.

Step 1.

HW Limit sensor 画面移动

<MOTOR>
 1: WATT 2: OND
 3: OFD 4: SESN
 5: GAIN 6: TQR
7. HWL

item #



- 选择7. HWL

<HWLIMIT>
 HW limit disable
 1 : ON 2 : ON
 3 : ON 4 : ON
 5 : ON 6 : ON

- HW Limit sensor 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 按下[ENTER] (ON → OFF → ON)
 - ON : 未使用
 - OFF : 使用

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.2.5.5 DELAY (EtherCAT 通信类型 Only)

- 设置Driver执行quick stop 指令时使用的Servo Off 为止的Delay时间.

Step 1.

ID 画面移动

〈SERVO〉
1. ID 2. M-TURN
3. DRIVER 4. BOARD
5. DELAY

item #



- 选择5: DELAY

〈SERVO OFF DELAY〉
q-stop sv off delay
T : 300 Z : 300
R1: 300 R2: 300
X : 300 6 : 300
R3: 300 R4: 300

- DELAY 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - Delay时间范围: 10~9999(ms)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.2.6 ORIGIN

- 设置Origin 相关参数.
- 利用用户设定的方法Scanning初始位置设定坐标原点的功能.

■ 详细项

GROUP	内容	范围	设定值
1:ENABLE	设置是否使用.		
2: VEL	设置ORIGIN 执行速度.		
3: SEQ	设置ORIGIN 执行时的动作有优先顺序.		
4: RULE	设置ORIGIN 执行方式.		

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 6: ORIGIN

Step 1.

ORIGIN移动

〈ROBOT PARAMETER〉
 1: CONF 2: BODY
 3: MOTION 4: PROTECT
 5: SERVO **6: ORIGIN**
 7: COORD

input : #



- 选择6: ORIGIN

ORIGIN设定画面确认

〈ORG〉
 1. ENABLE 2. VEL
 3. SEQ 4. RULE

item #

- ORIGIN 画面确认

2.3.2.6.1 ENABLE

- 设置是否使用ORIGIN.

Step 1.

ORIGIN 画面移动

<ORG>
1. ENABLE 2. VEL
3. SEQ 4. RULE

item #



- 选择1: ID

<ORG ENABLE>
ORIGIN ENABLE
ENABLE : **OFF**

- OIGIN ENABLE 画面确认

■ 设定方法

① 设定值说明

- ON: 使用ORIGIN 功能.
- OFF: 不使用ORIGIN 功能.

Step 1.

ENABLE 设定方法 (例子)

<ORG ENABLE>
ORIGIN ENABLE
ENABLE : **OFF**



- 按下"ENTER" 键反转 ON, OFF
(ON -> OFF, OFF -> ON)

<ORG ENABLE>
ORIGIN ENABLE
ENABLE : **ON**

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.2.6.2 VEL

- 设置ORIGIN 执行速度.

■ 详细项

GROUP	内容	范围	设定值
1: SEARCH VEL	设置从当前位置寻找Origin 传感器的速度.	1 ...50	10
2: ORIGIN VEL	设置传感器感知后寻找Z相的速度.	1 ...50	5
3: HOME VEL	设置ORIGIN 完成后寻找HOME 位置的速度.	1 ...50	20

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 6: ORIGIN-> 2: VEL

Step 1.

VEL 画面移动

<ORG>
 1. ENABLE 2. VEL
 3. SEQ 4. RULE

item #



- 选择2: VEL

<VEL>
 1. SEARCH VEL
 2. ORIGIN VEL
 3. HOME VEL

item #

- VEL 画面确认

1) SEARCH VEL

- 设置从当前位置寻找ORIGIN 传感器的速度.

Step 1.

VEL 画面移动

<VEL>
1. SEARCH VEL
2. ORIGIN VEL
3. HOME VEL

item #



- 选择1. SEARCH VEL

<SEARCH VLE>
VEL (%)
1 : 10 2 : 10
3 : 10 4 : 10
5 : 10 6 : 10

- SEARCH VEL 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2) ORIGIN VEL

- 设置ORIGIN传感器感知后寻找Z相的速度.

Step 1.

VEL 画面移动

<VEL>
1. SEARCH VEL
2. ORIGIN VEL
3. HOME VEL

item #



- 选择2. ORIGIN VEL

< ORIGIN VEL >
VEL (%)
1 : 5 2 : 5
3 : 5 4 : 5
5 : 5 6 : 5

- ORIGIN VEL 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

3) HOME VEL

- 设置ORIGIN 完成后移动到 HOME 位置的速度.

Step 1.

VEL 画面移动

<VEL>
1. SEARCH VEL
2. ORIGIN VEL
3. HOME VEL

item #



- 选择3. HOME VEL

< HOME VEL>
HOME VEL (%)
VLE: 20

- HOME VEL 设定画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.2.6.3 SEQ (Sequence)

- 设置ORIGIN 执行时动作优先顺序.
- 设置ORIGIN 动作优先顺序为最危险的顺序.
- 可以将各轴优先顺序设置为相同顺序.

Step 1.

ORIGIN 画面移动

<ORG>
1. ENABLE 2. VEL
3. SEQ 4. RULE

item #



- 选择3: SEQ

<SEQ>
origin sequence
1 :2 2 :2
3 :1 4 :1
5 :2 6 :2

- OIGIN SEQUENCE 画面确认

■ 设定方法

① 设定值说明

- 动作优先顺序 : 1 → 2 → 3 → 4
- 设定值 范围 : 1...4

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.2.6.4 RULE

- 设置ORIGIN 执行方式.

Step 1.

ORIGIN 画面移动

<ORG>
1. ENABLE 2. VEL
3. SEQ 4. RULE

item #



- 选择4: ORIGIN RULE

<RULE>
ORIGIN RULE
1 :4 2 :4
3 :4 4 :4
5 :4 6 :4

- OIGIN RULE 画面确认

Step 2

ORIGIN RULE 画面移动

<RULE>
ORIGIN RULE
1 :4 2 :4
3 :4 4 :4
5 :4 6 :4



- 选择1: 1… 6轴位置

<ORG RULE>
1 axis rule
RULE :4
DIR :CW
SENSOR :NEL
Z_PHASE :OFF
ORG DEC :OFF
EXIT

- OIGIN RULE 设定画面确认

■ 设定方法

- 请参考2.5.1.3 参数设定的 PARAMETER 值变更.

① 设定值说明(EtherCAT Communication Type)

- 工厂出厂时设定值为 '103'
- 设定值除下表中列出的值禁止使用其它值.

Rule No.	Description
100	当前位置
103	Home传感器的(+)方向搜索后向Index Pulse (Z相) 反方向移动
104	Home 传感器的 (+)方向搜索后向Index Pulse (Z相) 方向移动
105	Home 传感器的 (-)方向搜索后向Index Pulse (Z相) 反方向移动
106	Home 传感器的 (-)方向搜索后向Index Pulse (Z相) 方向移动
119	与103号相同但不确认Index Pulse(Z相)
120	与104号相同但不确认Index Pulse(Z相)
121	与105号相同但不确认Index Pulse(Z相)
122	与106号相同但不确认Index Pulse(Z相)

② 设定值说明(RS422 Communication Type)

- 工厂出厂时设定值为 '4'
- 设定值除下表中列出的值禁止使用其它值.
- ■■■ 橙色格表示常用的RULE

Rule No.	Description
0	读取当前MOTOR的位置值
1	+ 方向移动, CW Limit传感器确认
2	+ 方向移动, CW Limit传感器确认后Z相补偿
3	- 方向移动, CCW Limit传感器确认
4	- 方向移动, CCW Limit传感器确认后Z相补偿
13	+ 方向移动, Damper碰撞后Z相补偿
14	- 方向移动, Damper碰撞后Z相补偿

2.3.2.7 COORD (Coordinate)

- 返回坐标系时使用.
- 当前不支持.

Robostar

2.3.2.8 ARM (Arm Specialized Functions)

- 设置Hand特殊功能相关参数.

■ 详细项目

GROUP	内容	范围	设定值
1: AXIS	设置各轴Hand, Z轴类型.	参照说明	
2: INIT	设置Hand轴的初始位置.	参照说明	
3: FOLD	设置Hand轴Folding功能使用情况.	参照说明	
4: ZCAL	设置Z轴补偿使用情况及必要数据.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 8: ARM

Step 1.

ARM移动

(ROBOT PARAMETER)
 1: CONF 2: BODY
 3: MOTION 4: PROTECT
 5: SERVO 6: ORIGIN
 7: COORD **8: ARM**

input : #



- 选择8: ARM

ARM FUNC设定画面确认

(ARM FUNC)
 1: AXIS 2: INIT
 3: FOLD 4: ZCAL

item #

- ARM FUNC 画面确认

2.3.2.8.1 AXIS

- 设置各轴Hand, Z轴类型.

Step 1.

AXIS 画面移动

〈ARM FUNC〉
 1: **AXIS** 2: INIT
 3: FOLD 4: ZCAL

item #



- 选择1: AXIS

〈ARM-AXIS〉
 SET AXIS
 T : NONE Z : NONE
 R1: HAND R2: HAND
 X : NONE 6 : NONE

- AXIS 画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

① 设定值说明

- HAND: 设置为Hand轴.
- OPTION: 设置为附加轴.
- Z: 设置为上下 (Z)轴.
- NONE: 除Hand、附加、上下 (Z)轴的轴.

Step 1.

ENABLE 设定方法 (例)

〈ARM-AXIS〉
 SET AXIS
 T : ~~NONE~~ Z : NONE
 R1: HAND R2: HAND
 X : NONE 6 : NONE



- 按下“ENTER” 键变更设置 (NONE -> HAND -> OPTION -> Z -> NONE)


```
<ARM-AXIS>  
      SET  AXIS  
T : ☒ HAND  Z : NONE  
R1: HAND  R2: HAND  
X : NONE   6 : NONE
```

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.2.8.2 INIT

- 保存Hand轴的初始位置值.

Step 1.

INIT 画面移动

<ARM FUNC>
 1: AXIS
 3: FOLD
 2: **INIT**
 4: ZCAL

item #



- 选择2: INIT

<ARM-INIT>
 SET ARM INIT POS
 HAND : -1300
 OPTION: -

CLR

EXIT

- INIT 画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- ① [F1] CLR: 初始化输入的初始位置数据。初始化完成后输出为"-".
- ② [F4] EXIT: 恢复到以前画面.
- ③ 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.2.8.3 FOLD

- 设置Hand轴折叠到基准位置的功能使用情况.

Step 1.

FOLD 画面移动

〈ARM FUNC〉
1: AXIS 2: INIT
3: FOLD 4: ZCAL

item #



- 选择3: FOLD

〈FOLD ARM - ENABLE〉
ENAB FOLDING ARM
ENABLE : ☒ ON

- FOLD 画面确认

■ 设定方法

① 设定值说明

- ON: 使用功能.
- OFF: 不使用功能.

Step 1.

FOLD 设定方法

〈FOLD ARM - ENABLE〉
ENAB FOLDING ARM
ENABLE : ☒ ON

〈FOLD ARM - ENABLE〉
ENAB FOLDING ARM
ENABLE : ☐ OFF



- 按下"ENTER"键ON,OFF反转
(ON → OFF, OFF → ON)

■ 保存及取消方法

- 你参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.2.8.4 ZCAL

- 设置校正Hand轴下垂现象的功能相关内容.

■ 详细项目

GROUP	内容	范围	设定值
1: ENABLE	设置Z轴补偿使用情况.	参照说明	
2: SET/DISP	输入或确认多项式系数.	参照说明	
3: DATASET	输入多项式生成所需的HAND 位置及高度误差.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F1] ROBOT -> 8: ARM-> 4: ZCAL

Step 1.

URNG 画面移动

<ARM FUNC>

1: AXIS
3: FOLD

2: INIT
4: ZCAL

item #



- 选择4: ZCAL

<Z CAL>

1: ENALBE
3: DATASET

2: SET/DISP

item #

- ZCAL 画面确认

1) ENABLE

- 设置Z轴补偿使用情况.

Step 1.

ENABLE 设定

<Z_CAL>
1: **ENALBE** 2: SET/DISP
3: DATASET

item #



- 选择1. ENABLE

<Z_CAL - ENABLE>
ZCAL_ENABLE
ENABLE: SET_BY_JOB

- ENABLE 设定画面确认

■ 设定方法

① 设定值说明

- SET_BY_JOB: 通过JOB 输入Hand位置对应的下垂误差到JOB的模式.
- COEFFICIENT: 直接输入多项式系数的模式.
- OFF: 不使用Z轴补偿.

Step 1.

ENABLE 设定方法 (例)

<Z_CAL - ENABLE>
ZCAL_ENABLE
ENABLE: ☒ SET_BY_JOB



- 按下"ENTER" 键变更设置 (OFF -> SET_BY_JOB -> COEFFICIENT)

<Z_CAL - ENABLE>
ZCAL_ENABLE
ENABLE: ☒ COEFFICIENT

■ 保存及取消方法

- ① 请参考 [2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2) SET/DISP

- 输入多项式的系数.
- **ENABLE**设置为 **COEFFICIENT**的情况下才可以编辑.

Step 1.

SET/DISP设定

<Z_CAL>
1: ENALBE **2: SET/DISP**
3: DATASET

item #



- 选择2. SET/DISP

<Z_CAL - SET/DISP>
1: 5TH ORD 2: 4TH ORD
3: 3RD ORD 4: 2ND ORD
5: 1ST ORD 6: CONSTANT

item #

- 系数选择画面确认

Step 2.

系数选择

<5TH ORDER>
COEF: $a \cdot E - b \cdot x^5$
INPUT NUM(a):
-3.731886194722607
EXPONENTIAL(b): 23

item #

- 输入各次数失误值及指数系数

■ 游标位置移动

- ① 请参考[2.5.1.3 参数设定的](#) 游标位置移动及页面变更.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.
- ② 设定值说明
 - Input Num(a): 输入5次多项式的各次数失误值.
 - Coefficient(b): 输入5次多项式的指数系数.

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

3) DATASET

- 输入多项式生成算法中使用的Hand位置及误差值.
- **ENABLE设置为 SET_BY_JOB时才可以编辑.**

Step 1.

DATASET 设定

<Z_CAL>
1: ENALBE 2: SET/DISP
3: DATASET

item #

<Z_CAL - DATASET>
X(POS) Z(ERR)

0 :-1800	0
1 :-1589	0.25
2 :-1378	0.50
3 :-116639	0.75
4 :-956	1
CLR	EXIT



- 选择3. DATASET

- DATASET 画面确认

■ 光标位置移动

- ① 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- ① [F1] CLR: 初始化输入的初始位置数据。初始化完成时输出为"-".
- ② [F4] EXIT: 恢复到之前画面.
- ③ 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.
- ④ 设定值说明
 - **最少输入 5~ 最大 30个才可以生成多项式.** 不输入中间数据时无法生成多项式.
 - X(POS): 输入Hand的位置.
 - Z(ERR):输入Hand位置相应的误差值.

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.3 PUBLIC PARAMETER

2.3.3.1 ENV

■ 详细项目

GROUP	内容	范围	设定值
Page 1	1: BGT	设置是否自动执行Background task.	参照说明
	2: CONTACT	设置Auto mode, Battery, Fan, Light的节点Index.	参照说明
	3: SAFETY	设置Safety unit type.	参照说明
	4: COMM	与PC连接时的 RS232C 通信规格.	参照说明
	5: EXT IO	设置是否使用External I/O.	参照说明
	6: NAME	设置控制器名.	参照说明
	7: ALARM	设置是否使用FAN, SMPS Alarm.	参照说明
	8: FBUS	设置Fieldbus类型及Counter.	0 ~ 999
	9: TIME	设置系统时间.	参照说明
Page 2	1: ETH MO	设置以太网监控功能.	参照说明
	2: AC_RELAY	设置AC Relay类型和Power开关的连接情况,	参照说明

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F2] PUB -> 1: ENV

Step 1.

ENV移动

<PUBLIC PARAMETER>
 1: ENV 2: PASSWORD
 3: DRIVER

 item #



- 选择1: ENV

ENV 确认设置画面

<PUBLIC-ENV>

1: BGT	2: CONTACT
3: SAFETY	4: RS232C
5: EXT IO	6: NAME
7: ALARM	8: FBUS
9: TIME	

item #

- ENV 画面确认

2.3.3.1.1 BGT (Background Task)

- 设置是否自动执行Background task.

Step 1.

BGT 画面移动

<PUBLIC-ENV>
 1: **BGT** 2: CONTACT
 3: SAFETY 4: RS232C
 5: EXT IO 6: NAME
 7: ALARM 8: FBUS
 9: TIME

item #



- 选择1: BGT

<ENV-BGT>
 1: ENABLE

item #

- BGT 画面确认

Step 2.

移动ENABLE 设定画面

<ENV-BGT>
 1: **ENABLE**

item #



- 选择1: ENABLE

<BGT>
 AUTO RUN : NO
 1: YES 2: NO

input: #

- ENABLE 画面确认

■ 设定方法

① 设定值说明

- 1(YES): 使用AUTO RUN.
- 2(NO): 不使用AUTO RUN.

Step 1.

ENABLE 设定方法

〈BGT〉
 AUTO RUN : NO
1: YES 2: NO

input: #



- 输入数字 "1" 或 "2" 键决定是否使用 AUTO RUN

选择YES时

〈BGT〉
 AUTO RUN : NO
 1: YES 2: NO

YES selected !

- 如果选择YES , Command 窗口中输出 "YES selected !" 信息

选择NO时

〈BGT〉
 AUTO RUN : NO
 1: YES 2: NO

NO selected !

- 如果选择NO , Command 窗口中输出 "NO selected !" 信息

〈BGT〉
 AUTO RUN : NO
 1: YES 2: NO

update? (enter/esc)



- 按下"ESC" 键移动到Update 画面

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.2 CONTACT

- 设置Auto mode, Battery, Fan, Light的接点.

Step 1.

CONTACT 画面移动

〈PUBLIC-ENV〉
 1: BGT
 3: SAFETY
 5: EXT IO
 7: ALARM
 9: TIME
 2: **CONTACT**
 4: RS232C
 6: NAME
 8: FBUS

item #



- 选择2: CONTACT

〈ENV-CONTACT〉

AUTO MODE: -1
 BATTERY : -1
 FAN : -1
 LIGHT : -1

- CONTACT 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更请参考.

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 未使用时: -1
 - 使用时: 输出接点编号(0~64)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.
- ③ 设置接点Index时如果重复设置相同Index , Command 窗口中输出 "SAME INDEX" 信息. 请注意不重复并重新设置.

Step 1.

重复输入相同Index时

<ENV-CONTACT>

AUTO MODE: 1
BATTERY : 1
FAN : -1
LIGHT : -1

SAME INDEX

<PUBLIC-ENV>

1: BGT	2: CONTACT
3: SAFETY	4: RS232C
5: EXT IO	6: NAME
7: ALARM	8: FBUS
9: TIME	

item #

ENTER

ESC

- 设置接点Index时，如果输入相同Index并保存时Command窗口输出 "SAME INDEX!" 信息

- 按下"ENTER" 键或 "ESC"键恢复到之前画面，重新设置

2.3.3.1.3 SAFETY

- 设置Safety unit type.

Step 1.

SAFETY 画面移动

<PUBLIC-ENV>
 1: BGT 2: CONTACT
 3: **SAFETY** 4: RS232C
 5: EXT IO 6: NAME
 7: ALARM 8: FBUS
 9: TIME

item #



- 选择3: SAFETY

<HW CONF-SAFETY>
 SAFETY SELECT
 TYPE : ☒ 4UT

 GROUP : CT
 CATE : C4
 UTYPE : UNIT

- SAFETY 画面确认

■ 设定方法

① 设定值说明

- TYPE: 表示Safety Unit 的类型.

Step 1.

SAFETY 设定方法

<HW CONF-SAFETY>
 SAFETY SELECT
 TYPE : ☒ 4UT

 GROUP : CT
 CATE : C4
 UTYPE : UNIT



按下"ENTER" 键变换类型，相应类型的详细内容以 GROUP, CATEGORY, UNIT TYPE显示在下方.

<HW CONF-SAFETY>
 SAFETY SELECT
 TYPE : ☒ X7S

 GROUP : CX
 CATE : C4
 UTYPE : SERIAL

② 项说明

- 根据GROUP, CATEGORY, TYPE 类别分类如下. 根据控制器类型设置适合的类

型.

SAFETY TYPE	GROUP	CATEGORY	TYPE	备注
S4PO	SD	C4	PLC(OMRON)	
S4PZ	SD	C4	PLC(PILZ)	
L4PO	GP	C4	PLC(OMRON)	
L4PZ	GP	C4	PLC(PILZ)	
E4UT	CE	C4	UNIT	
B2	AB	C2	NONE	
C4PZ	CT	C4	PLC(PILZ)	
C4UT	CT	C4	UNIT	
CX7S	Not Defined	Not Defined	CX7 RS422	

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.4 COMM

- 设置与PC连接时的通讯规格.

Step 1.

RS232C 画面移动

<PUBLIC-ENV>

1: BGT

3: SAFETY

5: EXT IO

7: ALARM

9: TIME

2: CONTACT

4: RS232C

6: NAME

8: FBUS

item #

<ENV-RS232C>

BAUD: 115200

USED: OFF



- 选择4: RS232C

- RS232C 画面确认

Cursor位置移动

- ① 请参考2.5.1.3 参数设定 Cursor位置移动及页面变更.

设定方法

- 请参考2.5.1.3 参数设定的 PARAMETER 值变更.

- ① 设定值说明
 - USED: 决定使用情况. (ON/OFF)
 - TYPE, SUB_MENU: 设置通信类型及使用所需的内容.

TYPE	SUB_MENU	DISPLAY
RS-232C	BAUD RATE	<ENV-COMM> USED : OFF TYPE : RS-232C BAUD : 115200

ETHERNET	IP ADDRESS	<ENV-COMM> USED : OFF TYPE : ETHERNET IP : ☒XX.XXX.XXX.XXX

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.5 EXT IO (External I/O)

- 设置是否使用External I/O.

Step 1.

EXT IO 画面移动

<PUBLIC-ENV>
 1: BGT 2: CONTACT
 3: SAFETY 4: RS232C
 5: **EXT IO** 6: NAME
 7: ALARM 8: FBUS
 9: TIME

item #



- 选择5: EXT IO

<ENV-EXT IO>
 EXT IO ENABLE
 MASK: ☐ OFF

- EXT IO 画面确认

■ 设定方法

① 设定值说明

- ON: 使用External I/O.
- OFF: 不使用External I/O.

Step 1.

EXT IO 设定方法

<ENV-EXT IO>
 EXT IO ENABLE
 MASK: ☐ OFF

<ENV-EXT IO>
 EXT IO ENABLE
 MASK: ☒ ON



- 按下“ENTER” 键反转 ON, OFF
(ON → OFF, OFF → ON)

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.6 NAME

- 设置控制器名.

Step 1.

NAME 画面移动

<PUBLIC-ENV>
 1: BGT 2: CONTACT
 3: SAFETY 4: RS232C
 5: EXT IO 6: NAME
 7: ALARM 8: FBUS
 9: TIME

item #



- 选择6:NAME

<ENV-NAME>
 Controller name set

CUR:
 NEW: ■

SAVE

- NAME 画面确认

■ 设定方法

① 设定值说明

- 控制器名 (15字, 可以输入A~Z, 0~9)

Step 1.

NAME 设定方法

<ENV-NAME>
 Controller name set

CUR:
 NEW: ■

SAVE

- 使用"SHIFT" + 英文字母, 数字, SPACE
键输入控制器名

<ENV-NAME>
 Controller name set

CUR:
 NEW: CON_NAME1 ■

SAVE

■ 保存及结束方法

① 保存方法 (SAVE)

Step 1.

选择SAVE

〈ENV-NAME〉
Controller name set

CUR:
NEW: CON_NAME1■

SAVE

- 按下"F4" 键进行 SAVE

② 除保存的内容请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.7 ALARM

- 设置是否使用FAN, SMPS Alarm.

Step 1.

ALARM 画面移动

<PUBLIC-ENV>
 1: BGT 2: CONTACT
 3: SAFETY 4: COMM
 5: EXT IO 6: NAME
 7: **ALARM** 8: FBUS
 9: TIME

item #



- 选择7: ALARM

<PUB-ALARM>
 Alarm Disable
 FAN: ☐ OFF SMPS: OFF
 DIO: OFF GPIO: OFF

- ALARM 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ① 设定值说明
 - ON: 不使用Alarm.
 - OFF: 使用Alarm.

Step1.

ALARM 设定方法

<PUB-ALARM>
 Alarm Disable
 FAN: OFF SMPS: **OFF**
 DIO: OFF GPIO: OFF



- 按下"ENTER"键ON,OFF反转
(ON -> OFF, OFF -> ON)

〈PUB-ALARM〉
Alarm Disable
FAN: OFF SMPS: **ON**
DIO: OFF GPIO: OFF

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.8 FBUS (Fieldbus)

- 设置Fieldbus类型及Counter.

■ 详细项目

GROUP	内容	范围	设定值
1: SET	设置Fieldbus类型.	参照说明	
2: COUNT	设置Fieldbus Counter.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F2] PUB -> 8: FBUS

Step 1.

FBUS 移动

<PUBLIC-ENV>
 1: BGT 2: CONTACT
 3: SAFETY 4: RS232C
 5: EXT IO 6: NAME
 7: ALARM **8: FBUS**
 9: TIME

item #



- 选择8: FBUS

FBUS 设定画面确认

<FBUS>
 1: SET 2: COUNT

item #

- FBUS 设定画面确认

1) SET (Version 1.01.13 以上支持)

- 设置Fieldbus类型.

Step 1.

SET 画面移动

<FBUS>
1: SET
2: COUNT

item #

<FBUS SET>
FIELD BUS SET

TYPE : CCLINK V2
STNO : 1
RATE : 156K
CYCL : 1



- 选择1: SET

- FBUS SET 画面确认

■ 光标位置移动

- 请参考[2.3.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

① 设定值说明

- TYPE : 选择Fieldbus type.(CCLINK V1, CCLINK V2, ETHERCAT)
- STNO : 设置站号(Station Number).
(但, 仅在TYPE为 CCLINK V2时可以设置.)
- RATE : 选择通讯速度.(10M ~ 156K)
(但, 仅在TYPE为 CCLINK V2时可以设置.)
- CYCL : 选择周期(Extension cycle).(1 ~ 8)
(但, 仅在TYPE为 CCLINK V2时可以设置.)
- OCST : 选择占领战(Occupied Station).(1 ~ 4)
(但, 仅在TYPE为 CCLINK V2时可以设置.)

Step 1.

TYPE 设定方法

〈FBUS SET〉
FIELD BUS SET
TYPE: CCLINK V1

〈FBUS SET〉
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4

〈FBUS SET〉
FIELD BUS SET
TYPE: ETHERCAT

ENTER

- 按下“ENTER”键变更TYPE
(CCLINK V1 -> CCLINK V2 -> ETHERCAT -> CCLINK V1)
- STAT, RATE, CYCL 项目仅可以在 TYPE 为 CCLINK V2 时设定。

Step 2.

STNO 值设定方法

〈FBUS SET〉
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4

〈FBUS SET〉
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4

ENTER

- 按下要变更的值数字键输入

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

- 按下“ENTER”键反应到画面

Step 3.

RATE 值设定方法

```
<FBUS SET>
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4
```

- 按下"ENTER" 键变更RATE
(156K -> 625K -> 2.5M -> 5M -> 10M
-> 156K)

```
<FBUS SET>
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4
```

ENTER

- 按下"ENTER" 键反映到画面

Step 4.

CYCL 值设定方法

```
<FBUS SET>
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4
```

- 按下"ENTER" 键变更CYCL
(1 -> 2 -> 4 -> 8 -> 1)

```
<FBUS SET>
FIELD BUS SET
TYPE: CCLINK V2
STNO: 1
RATE: 156K
CYCL: 1
OCST: 4
```

ENTER

- 按下"ENTER" 键反映到画面

■ 保存及取消方法

- 请参考[2.3.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.3.1.9 TIME (System Time)

- 设置系统时间.

Step 1.

TIME 画面移动

〈PUBLIC-ENV〉	
1: BGT	2: CONTACT
3: SAFETY	4: RS232C
5: EXT IO	6: NAME
7: ALARM	8: FBUS
9: TIME	
item #	



- 9: TIME 选择

SYSTEM TIME SET	
C: 15/09/08	10:44:22
N: 15/09/08	10:44:22
ESC	

- TIME 画面确认

■ Cursor位置移动

- ① 使用左右方向键移动Cursor位置.
- ② 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

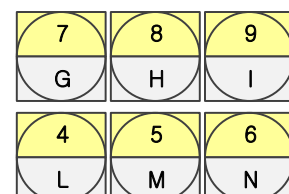
- ① 画面内容说明
 - C: 当前系统时间
 - N: 新系统时间
- ② 设定值说明
 - 输入系统时间

Step 1.

TIME 值设定方法

SYSTEM TIME SET	
C: 15/09/08	10:44:22
N: 15/09/08	10:44:22
ESC	

- 将Cursor移动到所需位置，利用数字键输入年/月/日和时信息



1	2	3	0
Q	R	S	V

SYSTEM TIME SET

C: 15/09/08 10:44:22
N: 15/09/08 11:07:22

ESC



- 按下"ENTER" 键反映到画面

确认修正完成信息

SYSTEM TIME SET

C: 15/09/08 10:44:22
N: 15/09/08 11:07:22

Time set complete!!!

- 时间信息保存完成后，Command窗口中输出 "Time set complete!!!" 信息

■ 保存及取消方法

- ① 按下"F4" 键可以执行ESC 功能.
- ② 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.

2.3.3.1.10 EXT ETH(External Ethernet)

- 为了通过外部以太网通信确认数据，设置使用情况和IP.

Step 1.

EXT MO 画面移动

<PUBLIC-ENV>
 1: BGT 2: CONTACT
 3: SAFETY 4: COMM
 5: EXT IO 6: NAME
 7: ALARM 8: FBUS
 9: TIME

item #



- 利用"PAGE UP/DOWN"移动到下一页

<PUBLIC-ENV>
1: EXT MO 2: AC_RELAY

item #

- 选择1: EXT MO

<ENV-ETH MONI>
 USED: ☐ OFF
 IP : XXX.XXX.XXX.XXX

- EXT MO 画面确认

■ 光标位置移动

- 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- 请参考[2.5.1.3 参数设定的](#) PARAMETER 值变更.

① 设定值说明

- USED: 选择是否使用以太网监控功能. (ON: 使用, OFF: 未使用)
- IP: 输入IP address.

Step 1.

USED设置

<ENV-ETH MONI>
USED: ☐ OFF
IP : XXX.XXX.XXX.XXX

ENTER

- 按下"ENTER" 键变更 ON, OFF
(ON-)OFF, OFF-)ON)

<ENV-ETH MONI>
USED: ☒ ON
IP : XXX.XXX.XXX.XXX

■ 保存及取消方法

- 请参考[2.3.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.3.1.11 AC_RELAY

- 设置AC_Relay 接点类型和 Power switch 信号连接情况.

Step 1.

AC_RELAY 画面移动

⟨PUBLIC-ENV⟩
1: BGT 2: CONTACT
3: SAFETY 4: COMM
5: EXT IO 6: NAME
7: ALARM 8: FBUS
9: TIME

item #



- 利用"PAGE UP/DOWN"移动到下一页

⟨PUBLIC-ENV⟩
1: EXT MO **2: AC_RELAY**

item #



- 选择2: AC_RELAY

⟨ENV-AC_RELAY⟩
RELAY/PW CONTACT
AC_RELAY : NC TYPE
PW_SWITCH: DISCONNECT

- AC_RELAY 画面确认

■ 光标位置移动

- ① 使用上下方向键移动光标的位置，
- ② 请参考[2.5.1.3 参数设定的](#) 光标位置移动及页面变更.

■ 设定方法

- ① 画面内容说明
 - AC_RELAY: 设置 AC RELAY 接点类型 (NC(Normal Close)/ NO(Normal Open))
 - PW_SWITCH: 选择Power switch 信号连接情况 (CONNECT/DISCONNECT)
- ② 设定值说明
 - 设置AC_RELAY 接点

Step 1.

AC RELAY接点类型设置

```

<ENV-AC_RELAY>
RELAY/PW CONTACT
AC_RELAY : NC TYPE
PW_SWITCH: DISCONNECT

```

```

<ENV-AC_RELAY>
RELAY/PW CONTACT
AC_RELAY : NO TYPE
PW_SWITCH: DISCONNECT

```

ENTER

- AC_RELAY位置上游标闪烁时，按下"ENTER"键变更 (NC TYPE -> NO TYPE, NO TYPE->NC TYPE)

- 设置Power switch 信号连接情况

Step 1.

SMPS类型设置

```

<ENV-AC_RELAY>
RELAY/PW CONTACT
AC_RELAY : NO TYPE
PW_SWITCH: DISCONNECT

```

```

<ENV-AC_RELAY>
RELAY/PW CONTACT
AC_RELAY : NO TYPE
PW_SWITCH: CONNECT

```

ENTER

- 移动方向键后，PW_SWITCH位置上游标闪烁时，按下"ENTER"键变更 (CONNECT -> DISCONNECT, DISCONNECT->CONNECT)

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值保存及取消方法.

2.3.3.2 PASSWORD

- 设置密码.

■ 详细项

GROUP	内容	范围	设定值
1: CHANGE	变更PASSWORD.		
2: DISABLE	决定PASSWORD 使用情况.		

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F2] PUB -> 2: PASSWORD -> 输入 PASSWORD

Step 1.

PASSWORD移动

〈PUBLIC PARAMETER〉
1: ENV 2: PASSWORD

item #



- 选择2: PASSWORD

Step 2.

输入PASSWORD

〈PUBLIC PARAMETER〉
1: ENV 2: PASSWORD

Password = ####



- 输入Password (初始值: "1111")

~



- 按下"ESC" 键取消输入 Password

〈EDIT MENU〉
1. JOB 2. POINT
3. VAR 4. PARA

Invalid!!

- 输入错误的Password 时显示"Invalid"

Step 2.

PASSWORD 画面确认

〈ETC-PASS〉
1: CHANGE 2: DISABLE

item #

- PASSWORD 画面确认

2.3.3.2.1 CHANGE

- 变更PASSWORD.

Step 1.

CHANGE 画面移动

<ETC-PASS>
1: CHANGE 2: DISABLE

item #



- 选择1: CHANGE

<PASS-CHANGE>
Change password
VALUE = 1111

- CHANGE 画面确认

■ 设定方法

- ① 请参考[2.5.1.3 参数设定](#) PARAMETER 值变更.
- ② 设定值说明
 - 输入4位数字

■ 保存及取消方法

- ① 请参考[2.5.1.3 参数设定](#) 参数值保存及取消方法.
- ② 设定值超过最大范围或未达到最小范围时输出错误消息.

2.3.3.2.2 DISABLE

- 决定是否使用PASSWORD.

Step 1.

DISABLE 画面移动

〈ETC-PASS〉
1: CHANGE **2: DISABLE**

item #



- 选择2: DISABLE

〈PASS-DISABLE〉
Password Disable
PARA: OFF BRK: OFF
ORG : OFF

- DISABLE 画面确认

■ Cursor位置移动

- ① 请参考[2.5.1.3 参数设定](#) Cursor位置移动及页面变更.

■ 设定方法

- ② 选择Password 输入窗口使用情况.
 - PARA: Parameter 相关内容编辑时
 - BRK: break 相关内容编辑时
 - ORG: Origin 相关内容编辑时
- ③ 设定值说明
 - ON: 不询问Password.
 - OFF: 询问Password.

Step1.

DISABLE 设定方法

〈PASS-DISABLE〉
Password Disable
PARA: OFF BRK: OFF
ORG : OFF



- 按下"ENTER" 键反转 ON, OFF
(ON → OFF, OFF → ON)

〈PASS-DISABLE〉
Password Disable
PARA: **ON** BRK: OFF
ORG : OFF

Robostar

2.3.3.3 DRIVER

- 设置适合控制器的Driver文件.

■ 详细项

GROUP	内容	范围	设定值
1: CONFIG	设置Driver文件.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F2] PUB -> 3: DRIVER -> 输入CONFIG

Step 1.

DRIVER移动

〈PUBLIC PARAMETER〉
1: ENV 2: PASSWORD
3: DRIVER

item #



- 选择3: DRIVER

DRIVER 设定画面确认

〈PUBLIC-DRIVER〉
1: CONFIG

item #

- DRIVER 画面确认

2.3.3.3.1 CONFIG

- 设置Driver文件.

Step 1.

CONFIG 画面移动

<PUBLIC-DRIVER>

1: CONFIG

item #



- 选择1: CONFIG

<NUMBER OF DRIVERS>

Num(0~)16) = ■

- CONFIG 画面确认

■ 设定方法

- ① 输入控制器的Driver 个数

- 输入值范围: 0~16

Step 1.

输入Driver个数

<NUMBER OF DRIVERS>

Num(0~)16) = ■



~

- 输入控制器的Driver个数



选择或取消

<NUMBER OF DRIVERS>

Num(0~)16) = 16 ■

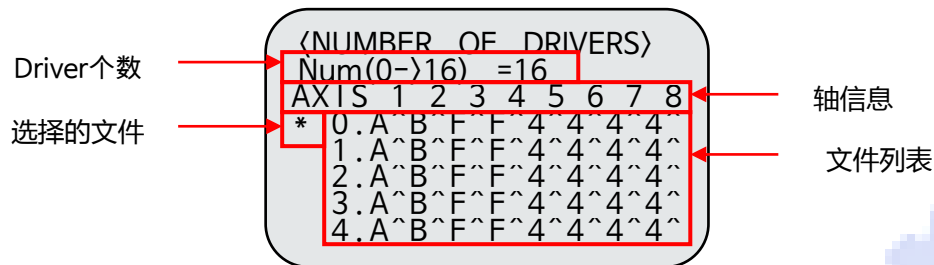


- 按下"ENTER" 键移动到文件选择窗口



- 按下"ESC" 键恢复到之前画面

② 文件选择窗口画面说明



③ 选择文件

Step 1.

选择文件及移动

```

<NUMBER OF DRIVERS>
Num(0->16) =16
AXIS 1 2 3 4 5 6 7 8
* 0.A^B^F^F^4^4^4^4^
  1.A^B^F^F^4^4^4^4^
  2.A^B^F^F^4^4^4^4^
  3.A^B^F^F^4^4^4^4^
  4.A^B^F^F^4^4^4^4^
  
```



- 操作"上下方向键"变更 * 位置，选择要初始化的文件
- 文件个数多时 可以 按下 "PAGE UP/DOWN" 键移动

确认整体文件名

```

<NUMBER OF DRIVERS>
Num(0->16) =16
AXIS 1 2 3 4 5 6 7 8
* 0.A^B^F^F^4^4^4^4^
  1.A^B^F^F^4^4^4^4^
  2.A^B^F^F^4^4^4^4^
  3.A^B^F^F^4^4^4^4^
  4.A^B^F^F^4^4^4^4^
  
```



- 按下"左右方向键"可以确认整个文件名
- 可以通过AXIS 行输出的值确认相应到第几个轴

```

<NUMBER OF DRIVERS>
Num(0->16) =16
AXIS 9 A B C D E F G
* 0.1^1^1^1^1^1^1^1^
  1.1^1^1^1^1^1^1^2^
  2.1^1^1^1^1^1^1^3^
  3.1^1^1^1^1^1^1^4^
  4.1^1^1^1^1^1^1^5^
  
```

最终选择

(NUMBER OF DRIVERS)
 Num(0~16) = 16
 AXIS 1 2 3 4 5 6 7 8
 * 0.A^B^F^F^4^4^4^4^

ENTER

- 移动到所需文件后按下 "ENTER" 键移动到下一个画面

Step 2.

更新或取消

CUR:1^2^7%1^1^

NEW:A^B^F^F^4^4^4^4^1^1^1^1^1^1^1^

update? (ENT/ESC)

ENTER

ESC

- 当前设置的Driver文件和用户选择的内容与 update 信息一起显示
- 按下"ENTER" 键保存文件名和Driver个数后恢复到上一级画面
- 按下"ESC" 键取消更新后恢复到上一级画面

(PUBLIC-DRIVER)
 1: CONFIG

item #

- 移动到上一级画面

■ 参考事项

① 文件名意义说明

文件名: (3) (1)A (2)^ B^F^F^4^4^4^4^1^1^1^1^1^1^1^1^1^

(1) 根据电机容量的标记方法

Motor 용량	Driver		标记
	Frame	Model	
50	A	1505	1
100			
200		1507	2
400	B	2510	4
750	C	3520	7
1K	D	5510	A
1.5K			
2K	E	7364	B

3K	F	A390	C
4K		B3A2	F
5K			
7.5K	G		G

(2) 根据Driver类型的标记方法

Driver类型	BA	BD	BL	B2
标记	^	#	%	-

(3) Driver个数= 整体文件长度/ 2 (ex, 32 / 2 = 16)

2.3.3.4 MOVEMENT

■ 详细项目

GROUP	内容	范围	设定值
1: JOB	设置MOVEMENT功能中使用的JOB程序.	参照说明	
2: DISP	设置MOVEMENT画面类型.	参照说明	

■ OPEN顺序

- Top Screen -> 1. Edit -> 3. PARA -> [F2] PUB -> 输入4: MOVEMENT

Step 1.

MOVEMENT 移动

〈PUBLIC PARAMETER〉
 1: ENV 2: PASSWORD
 3: DRIVER 4: MOVEMENT

item #



- 选择4: MOVEMENT

MOVEMENT 设定画面确认

〈PUBLIC-MOVEMENT〉
 1. JOB 2. DISP

item #

- MOVEMENT 画面确认

2.3.3.4.1 JOB

- 选择MOVEMENT 功能中使用的 JOB 程序.

Step 1.

JOB 画面移动

<PUBLIC-MOVEMENT>
1. JOB 2. DISP

 item #

 <MOVEMENT-JOB>
 MOVEMENT DEFAULT JOB
 JOB NAME =

 FIND SAVE



- 选择1: JOB

- JOB 画面确认

■ 设定方法

- 按下[F1] FIND 键确认 JOB 列表后, 选择 JOB 文件并按下[ENTER] 键.

Step1.

JOB 设定方法

<MOVEMENT-JOB>
 MOVEMENT DEFAULT JOB
 JOB NAME =

 FIND SAVE



- 按下[F1] 键确认 JOB 选择列表

Step 2

JOB程序选择

<MOVEMENT-JOB>
 F: 39
 * 0. AFOLD
 1. AGT
 2. BGT
 3. MANUL



- 按下"上/下 方向键"移动到所需的文件

Step 3.

JOB选择后确认

<MOVEMENT-JOB>

F: 39
0. AFOLD
1. AGT
2. BGT

* 3. MANUL

ENTER

- 移动后按下 "ENTER" 键选择 JOB 程序

<MOVEMENT-JOB>
MOVEMENT DEFAULT JOB
JOB NAME = MANUL

FIND

SAVE

■ 保存及取消方法

- 请参考[2.5.1.3 参数设定的](#) 参数值取消方法.

Step1.

JOB保存方法

<MOVEMENT-JOB>
MOVEMENT DEFAULT JOB
JOB NAME = MANUL

F4

- 按下 [F4] 键保存

FIND

SAVE

<PUBLIC-MOVEMENT>
1. JOB 2. DISP

- 保存后恢复到以前画面

item #

2.3.3.4.2 DISP

- 选择输出到Main菜单的MOVEMENT 菜单的画面类型.

Step 1.

DISP 画面移动

<PUBLIC-MOVEMENT>
1. JOB 2. DISP

item #

F2

- 选择2: DISP

<MOVEMENT-DISP TYPE>
TYPE: TYPE 1 ■

- DISP 画面确认

■ 设定方法

① 选择输出画面类型.

- TYPE 1: 用户直接输入Point信息
- TYPE 2: 用户输入STAGE, SIZE 等信息, 利用相应信息自动设置Point信息

Step1.

输出画面设定方法

<MOVEMENT-DISP TYPE>
TYPE: TYPE 1 ■

<MOVEMENT-DISP TYPE>
TYPE: TYPE 2 ■

ENTER

- 按下"ENTER" 键反转TYPE 1, TYPE 2
(TYPE 1 → TYPE 2, TYPE 2 → TYPE 1)

■ 保存及取消方法

- 请参考 [2.5.1.3 参数设定的](#) 参数值保存及取消方法.

第3章 RUN 模式

- RUN模式中执行EDIT 模式的 JOB 菜单中利用Robot语言制作的JOB 文件(RRL).
- RUN 模式中执行JOB 文件前一定要执行原点设定.
- 本章节针对RUN 模式的执行方法及监控方法等具体说明.

3.1 概要

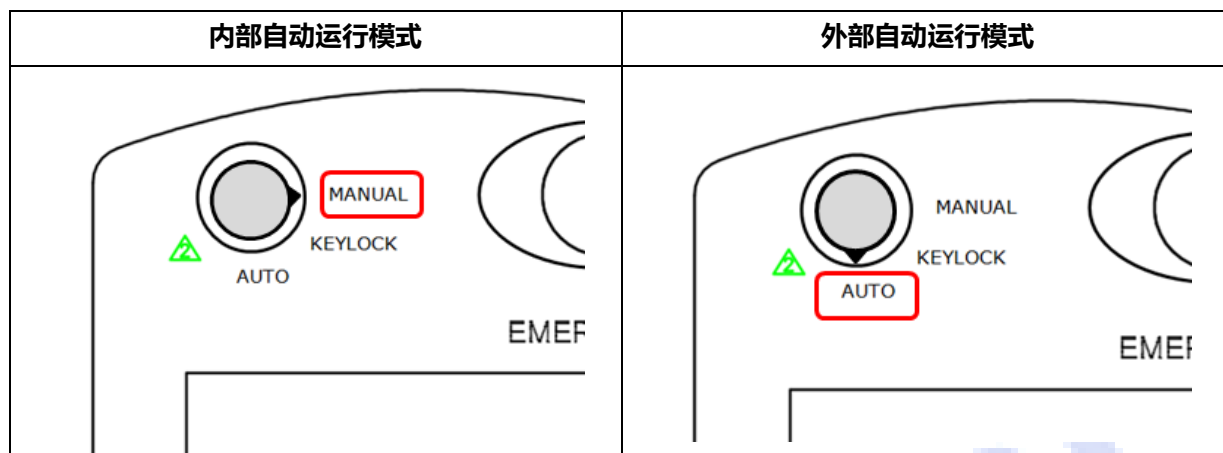
- ✓ 从JOB文件列表中选择要执行的作业后执行即可.
- ✓ Teach Pendant中的 RUN模式执行方法有以下两种.
 - 1) MANUAL RUN (内部自动运行模式) - AUTO/STEP RUN
 - 2) AUTO RUN (外部自动运行模式)

■ RUN 模式种类

GROUP		内容	备注
MANUAL RUN (内部自动 运行模式)	AUTO/STEP RUN	STEP: 程序以 1Step(1行)单位执行并停止. AUTO: 执行程序整体.	
AUTO RUN (外部自动运行模式)		从Sequence等外部机械执行Robot程序的模式.	通常设备的自 动运行模式

■ RUN 模式选择方法

- ① 如下图，旋转Teach Pendant 上端开关选择自动运行模式.
- ② 运行模式的变更只有在最上位菜单时才可以.

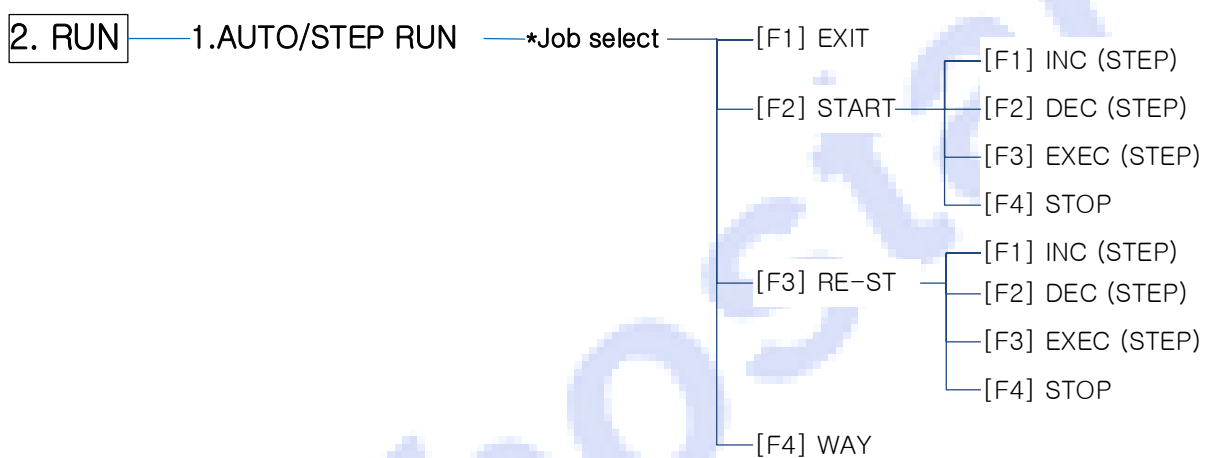


3.2 RUN 模式流程图

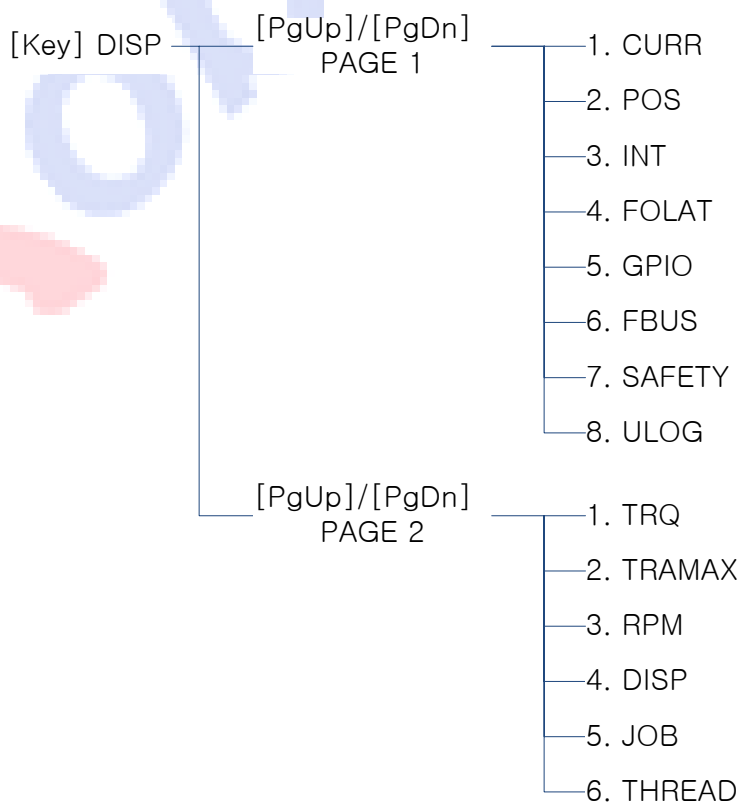
■ 外部自动运行模式 Menu Tree

*Thread Select — [F1] SEL

■ 内部自动运行模式 Menu Tree



■ Monitoring Menu Tree



3.3 AUTO RUN (外部自动运行模式)

- ✓ 外部自动运行模式是通常设备的自动运行模式，重复执行设定的Robot程序.
- ✓ 旋转Teach Pendant 上端开关选择外部自动运行模式后可以确认Thread 信息和状态.
- ✓ 同时可以使用4个Thread, Robot正在执行动作时STATE显示为RUN. 此外，Robot没有执行作业时显示为IDLE 状态.
- ✓ 按下"**F1**" 键选择作业，输出可以确认作业内容和状态值的监控画面.

■ AUTO RUN 初始画面

〈AUTO : TH INFO〉	
THREAD	STATE
*T H1	RUN
TH2	IDLE
TH3	IDLE
TH4	IDLE
SEL	EXIT

3.4 MANUAL RUN (内部自动运行模式)

✓ 首先, 建议执行STEP RUN确认程序是否有异常, 再执行AUTO RUN.

3.4.1 Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

ENTER

- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择RUN

<TEACH MENU>
1. EDIT **2. RUN**
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

2
R

- 选择2. RUN

<RUN>
SELECT RUN MODE
1. AUTO/STEP RUN

input : #

1
Q

- 选择AUTO/STEP RUN

<RUN MODE>
F: 40
* 0. test_bitop5
1. test_rest
2. bb4
3. out_test5
4. out_test3
5. test_rest2

- 文件列表画面确认

```
<MAN : INS>
F: test4      S:2    S
T: 1      V: 5  Q4AFL1
  a = 0

  if a == 0
    b = 1
EXIT  START  RE-ST  WAY
```

- AUTO/STEP RUN 画面确认

3.4.2 AUTO/STEP RUN

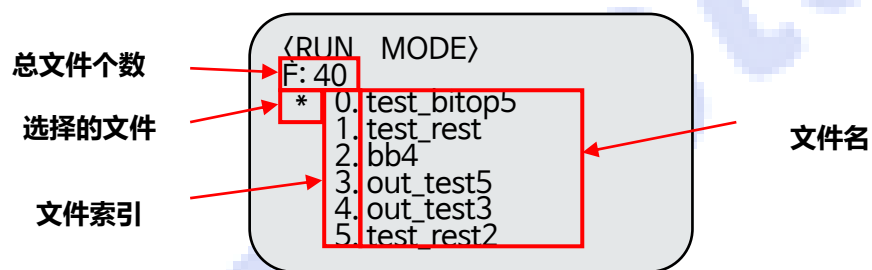
※ 注意 ※

- 建议先执行STEP RUN确认程序是否有异常，再执行AUTO RUN..

3.4.2.1 画面说明

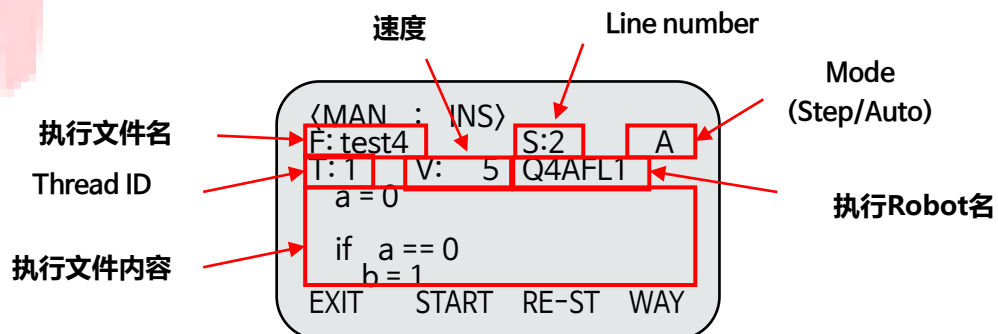
■ 画面说明

① 文件选择 画面



② AUTO/STEP RUN 画面

- 1) **EXIT**: 结束
- 2) **START**: 从文件头开始执行
- 3) **RE-ST(Restart)**: 从停止的位置重新开始执行
- 4) **WAY**: 执行模式变更 (STEP->AUTO, AUTO->STEP)



- 1) **STEP 模式时**, START 及 RE-ST执行时的画面
- 2) INC: 移动到上一个STEP
- 3) DEC: 移动到下一个STEP
- 4) EXEC: 执行
- 5) STOP: 执行停止及上一画面恢复

```
<MAN : INS>
F: test4      S:2      S
T: 1      V: 5 Q4AFL1
a = 0
  if a == 0
    b = 1
INC      DEC      EXEC      STOP
```

③ **AUTO 模式时**, START 及 RE-ST 执行时的画面

- 1) STOP: 执行停止及上一画面恢复

```
<MAN : INS>
F: test4      S:2      A
T: 1      V: 5 Q4AFL1
a = 0
  if a == 0
    b = 1
                                STOP
```

3.4.2.2 选择执行文件

- AUTO/STEP RUN 模式中选择执行文件.

■ 选择执行文件

Step 1.

选择AUTO/STEP RUN

```
<RUN>
SELECT RUN MODE
1. AUTO/STEP RUN
```

input : #



- 选择1.AUTO/STEP RUN

```
<RUN MODE>
F: 40
* 0. test_bitop5
  1. test_rest
  2. bb4
  3. out_test5
  4. out_test3
  5. test_rest2
```

- 文件列表画面确认

Step 2.

选择JOB 文件

```
<RUN MODE>
F: 40
* 0. test_bitop5
- 1. test_rest
  2. bb4
  3. out_test5
  4. out_test3
  5. test_rest2
```



- 按下"PAGE UP/DOWN" 键移动页面

```
<RUN MODE>
F: 40
* 6. test1
- 7. test2
  8. test3
  9. test4
 10. test5
 11. test6
```



- 按下"上下方向键" 移动到所需文件

```
<RUN MODE>
F: 40
  6. test1
  7. test2
  8. test3
* 9. test4
- 10. test5
 11. test6
```



- 按下"ENTER" 键选择最终文件

```
<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 5  Q4AFL1
  a = 0

  if a == 0
    b = 1
  EXIT      START  RE-ST  WAY
```

- AUTO/STEP RUN 画面确认



■ 程序执行前注意事项

输入作业程序后执行作业时请注意一下安全事故.

第一, 操作者务必移动到Robot作业区域外.

第二, 请做好按下紧急停止(EMERGENCY) 开关的准备.

第三, 请再次确认Robot的作业Point及移动路径.

第四, 在低速环境下执行作业后渐渐加速.

3.4.2.3 执行速度变更

- AUTO/STEP RUN 模式下变更执行速度.

■ 执行速度变更

- ① 作业执行速度V如果在作业程序中没有设置, 则为最高速度 (参数 Mv 设定值)的 20%.
- ② 可以以5% 为单位调整, 范围为 5~100%.

■ 执行速度变更

Step 1.

执行速度变更

```
<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 5  Q4AFL1
a = 0
```

```
if a == 0
  b = 1
EXIT  START  RE-ST  WAY
```



- 按下上下方向键调整执行文件的作业速度

```
<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 20 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
EXIT  START  RE-ST  WAY
```



作业中变更执行速度

```
<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 5  Q4AFL1
a = 0
```

```
if a == 0
  b = 1
INC   DEC   EXEC  STOP
```

- STEP/AUTO 作业画面中也可以按下上下方向键调整执行文件的速度

※ 注意 ※

- 执行开始画面中提高速度执行START时, 可能发生速度RESET的现象.
- RE-ST时继续维持原有速度.

3.4.2.4 执行STEP RUN

- AUTO/STEP RUN 模式中执行STEP 模式.
- STEP 模式中以 1STEP 为单位执行作业.

STEP 模式设定

Step 1.

STEP模式设定

```
<MANUAL : INS>
F: test4      S:2
T: 1 V: 5 Q4AFL1 A
a = 0
```

```
if a == 0
  b = 1
EXIT START RE-ST WAY
```

```
<MANUAL : INS>
F: test4      S:2
T: 1 V: 5 Q4AFL1 S
a = 0
```

```
if a == 0
  b = 1
EXIT START RE-ST WAY
```

F4

- 按下"F4" 键，如果为A(AUTO)时变更为S(STEP)

STEP 模式执行

Step 1.

STEP模式执行

```
<MANUAL : INS>
F: test4      S:2 S
T: 1 V: 5 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
EXIT START RE-ST WAY
```

```
<MANUAL : INS>
F: test4      S:2 S
T: 1 V: 5 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
INC DEC EXEC STOP
```

F2

- 按下"F2" 键，从执行文件的第一行开始执行 STEP 模式

- 执行画面确认

LED 确认

ORG SVON **RUN** SHIFT ALRM

- 确认RUN LED亮灯

重新执行STEP 模式

```

<MANUAL : INS>
F: test4      S:2      S
T: 1 V: 5 Q4AFL1
a = 0

if a == 0
  b = 1
EXIT START RE-ST WAY

```



- 按下"F3" 键 从STOP 时的STEP line开始执行

Step 2.

执行1 STEP

```

<MANUAL : INS>
F: test4      S:2      S
T: 1 V: 5 Q4AFL1
a = 0

if a == 0
  b = 1
INC DEC EXEC STOP

```



- 按下"F3" 键执行 1 STEP
- 重复所需STEP 按下Key

```

<MANUAL : INS>
F: test4      S:3      S
T: 1 V: 5 Q4AFL1
a = 0

if a == 0
  b = 1
INC DEC EXEC STOP

```

- 可以确认第二行S:的编号变更

执行INC

```

<MANUAL : INS>
F: test4      S:2      S
T: 1 V: 5 Q4AFL1
a = 0

if a == 0
  b = 1
INC DEC EXEC STOP

```

- 移动到下一个STEP

执行DEC

```

<MANUAL : INS>
F: test4      S:2      S
T: 1 V: 5 Q4AFL1
a = 0

if a == 0
  b = 1
INC DEC EXEC STOP

```

- 移动到上一个STEP

Step 3.

STEP 模式结束

```

<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 5 Q4AFL1
a = 0
if a == 0
  b = 1
INC      DEC      EXEC STOP

```



- 按下"F4" 键停止 STEP 模式恢复到上一个画面

```

<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 5 Q4AFL1
a = 0
if a == 0
  b = 1
EXIT      START RE-ST WAY

```

- 确认上一个画面

STEP 模式结束

Step 1.

STEP 模式结束

```

<MANUAL : INS>
F: test4      S:2      S
T: 1      V: 5 Q4AFL1
a = 0
if a == 0
  b = 1
EXIT      START RE-ST WAY

```

- 按下"F1" 键恢复到主画面

```

<TEACH MENU>
1. EDIT      2. RUN
3. INFO      4. I/O
5. ORIGIN

0. SHUTDOWN input : #

```

- 主画面确认

3.4.2.5 执行AUTO RUN

- AUTO/STEP RUN 模式中执行 AUTO 模式.
- AUTO 模式中所有STEP按顺序执行作业.

■ AUTO 模式设定

Step 1.

AUTO 模式设定

```
<MANUAL : INS>
F: test4      S:2
T: 1 V: 5 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
EXIT START RE-ST WAY
```



- 按下"F4" 键, 如果为S(STEP)时变更为 A(AUTO)

```
<MANUAL : INS>
F: test4      S:2
T: 1 V: 5 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
EXIT START RE-ST WAY
```

■ AUTO 模式执行

Step 1.

AUTO 模式执行

```
<MANUAL : INS>
F: test4      S:2
T: 1 V: 5 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
EXIT START RE-ST WAY
```



- 按下"F2" 键从执行文件的第一行开始执行 AUTO模式

```
<MANUAL : INS>
F: test4      S:2
T: 1 V: 5 Q4AFL1
a = 0
```

```
if a == 0
  b = 1
STOP
```

- 输出执行画面, 自动执行

LED 确认

- 确认RUN LED亮灯

○ORG ○SVON ●RUN ○SHIFT ○ALRM

重新执行AUTO 模式

```

<MANUAL : INS>
F: test4      S:2      A
T: 1      V: 5  Q4AFL1
a = 0

if a == 0
  b = 1
EXIT      START  RE-ST  WAY
  
```



- 按下"F3" 键 从STOP 时的STEP line开始执行

Step 2.

AUTO 模式结束

```

<MANUAL : INS>
F: test4      S:2      A
T: 1      V: 5  Q4AFL1
a = 0

if a == 0
  b = 1
  
```

STOP



- 按下"F4" 键停止 AUTO 模式恢复到上一个画面

```

<MANUAL : INS>
F: test4      S:2      A
T: 1      V: 5  Q4AFL1
a = 0

if a == 0
  b = 1
EXIT      START  RE-ST  WAY
  
```

- 上一个画面确认

■ AUTO 模式结束

Step 1.

AUTO 模式结束

```

<MANUAL : INS>
F: test4      S:2      A
T: 1      V: 5  Q4AFL1
a = 0

if a == 0
  b = 1
EXIT      START  RE-ST  WAY
  
```

- 按下"F1" 键恢复到主画面

```

<TEACH MENU>
1. EDIT      2. RUN
3. INFO      4. I/O
5. ORIGIN
  
```

- 主画面确认

```

0. SHUTDOWN
input : #
  
```

3.5 监控

- ✓ 为诊断系统各种状态而提供.

3.5.1 概要

- 执行RUN模式的过程中可以实时确认作业内容及状态值.
- 可以监控AUTO RUN和 MANUAL RUN.
- 对于提供的功能请参考下面的详细项.

■ 详细项

GROUP		内容	显示单位
Page1	1 :CURR	可以确认Robot各轴的当前位置.	-移动轴: mm/s
	2: POS	可以确认Robot各轴的Teaching位置.	旋转轴: degree/s
	3: INT	可以确认Edit中编辑的Integer 值.	-
	4: FLOAT	可以确认Edit中编辑的Float 值.	-
	5: GPIO	可以确认GPIO的 I/O 值.	-
	6: FBUS	可以确认Fieldbus的 I/O 值.	-
	7: SAFETY	可以确认SAFETY 状况.	-
	8: ULOG	可以确认用户JOB log.	-
Page2	1: TRQ	可以确认Robot各轴的Torque.	%
	2: TRQMAX	可以确认Robot各轴的最大扭矩.	%
	3: LOAD	可以确认各轴的平均负荷率.	%
	4: RPM	可以确认各轴的RPM速度.	RPM
	5: DIST	电机总驱动距离	-
	6: JOB	可以确认当前执行的JOB内容.	-
	7: MVR	两个POINT之间移动时, 可以确认整体距离对当前位置的百分比值.	%
	6: THREAD	可以设置执行文件的Thread ID. 仅在MANUAL RUN 时激活.	-
Page 3	1: INSHAPE	可以确认给予Motion的变化量.	

3.5.2 监控画面输出方法

3.5.2.1 AUTO RUN 模式时

- 选择Thread 后第一个画面输出INC窗口.

Step 1.

选择THREAD

```

<AUTO : TH INFO>
THREAD      STATE
*T H1      RUN
TH2        IDLE
TH3        IDLE
TH4        IDLE
SEL
  
```

- 接入电源后画面中显示 Thread 信息

```

<AUTO : TH INFO>
THREAD      STATE
*T H1      RUN
TH2        IDLE
TH3        IDLE
TH4        IDLE
SEL
  
```



- 按下"UP/DOWN" 键选择要确认信息的 Thread



- 选择的 Thread行最前面显示"*" 符号

选择SEL

```

<AUTO : TH INFO>
THREAD      STATE
*T H1      RUN
TH2        IDLE
TH3        IDLE
TH4        IDLE
SEL
  
```



- 按下"F1" key选择 "SEL".

确认INS窗口

```

<AUTO : INS>
F: test4      S:2      A
T: 1      V: 5 Q4AFL1
a = 0
if a == 0
  b = 1
  
```

- 可以确认INS 窗口
- 红色框内的内容根据程序执行STEP 有变化

Step 2.

确认DISPLAY 选择窗口


```

<AUTO : INS>
F: test4      S:2      A
T: 1      V: 5      Q4AFL1
a = 0

if a == 0
b = 1

```



- 按下"DISP"键确认DISPLAY 选择窗口

```

<AUTO : DISP>      (1/2)
1.CURR      2. POS
3. INT      4. FLOAT
5. GPIO     6. FBUS
7. SAFETY   8. ULOG

input : #

```

- DISPLAY 输出窗口确认

Step 3.

页面选择

```

<AUTO : DISP>      (1/2)
1.CURR      2. POS
3. INT      4. FLOAT
5. GPIO     6. FBUS
7. SAFETY   8. ULOG

input : #

```



- "可以利用"PAGE UP/DOWN" 键切换画面

```

<AUTO : DISP>      (2/2)
1. TRQ      2. TRQMAX
3. RPM      4. DIST
5. JOB

```



Step 4.

取消选择监控菜单画面

```

<AUTO : DISP>      (1/2)
1.CURR      2. POS
3. INT      4. FLOAT
5. GPIO     6. FBUS
7. SAFETY   8. ULOG

input : #

```



- 按下"ESC" 键恢复到上一个画面

```

<AUTO : DISP>      (1/2)
1.CURR      2. POS
3. INT      4. FLOAT
5. GPIO     6. FBUS
7. SAFETY   8. ULOG

input : #

```

```
<AUTO : DISP> (1/2)
1.CURR          2.POS
3.INT           4.FLOAT
5.GPIO          6.FBUS
7.SAFETY        8.ULOG
```

input : #

```
<AUTO : INS>
F:test4 S:2 A
T:1 V: 5 Q4AFL1
a = 0
if a == 0
b = 1
```

3.5.2.2 MANUAL RUN 模式时

Step 1.

监控菜单画面输出

```

<MAN : INS>
F: test4      S:2      S
T: 1      V: 5  Q4AFL1
a = 0

```

```

if a == 0
  b = 1
EXIT      START  RE-ST  WAY

```

```

<MAN : INS>
F: test4      S:2      S
T: 1      V: 5  Q4AFL1
a = 0

```

```

if a == 0
  b = 1
INC      DEC  EXEC  STOP

```

```

<MAN: DISP>      (1/2)
1. CURR          2. POS
3. INT           4. FLOAT
5. GPIO          6. FBUS
7. SAFETY        8. ULOG

```

input : #



- 按下"DISP"键确认DISPLAY 选择窗口

- 监控画面确认

Step 2.

页面选择

```

<MAN : DISP>      (1/2)
1. INS            2. POS
3. RPM            4. TRQ
5. TRQMAX         6. GPIO
7. SAFETY         8. FBUS

```

input : #



- "可以利用"PAGE UP/DOWN" 键切换画面

```

<MAN: DISP>      (2/2)
1. TRQ            2. TRQMAX
3. RPM            4. DIST
5. JOB            6. THREAD

```



Step 3.

取消选择监控菜单画面

```
< MAN: DISP> (1/2)
1.CURR      2. POS
3. INT      4. FLOAT
5. GPIO     6. FBUS
7. SAFETY   8. ULOG
```

input : #

```
<MAN : INS>
F: test4      S:2      S
T: 1          V: 5     Q4AFL1
a = 0
if a == 0
  b = 1
EXIT  START  RE-ST  WAY
```

ESC

- 按下"ESC" 键恢复到上一个画面

※ 注意 ※

- MANUAL RUN 模式中监控用的功能键说明不会显示在画面.

3.5.3 监控功能及画面说明

3.5.3.1 CURR (Current Position)

- 可以确认Robot各轴的当前位置.

Step 1.

CURR 画面移动

```
< AUTO: DISP> (1/2)
1.CURR      2. POS
3.INT       4. FLOAT
5.GPIO      6. FBUS
7.SAFETY    8. ULOG
```

input : #



- 选择1: CURR

Step 2.

选择ROBOT

```
<SELECT ROBOT>
ROBOT TYPE
*R1      Q4AFL1
R2      NO-EXIT
R3      NO-EXIT
```

SEL

EXIT



- 按下 "UP/DOWN" 键选择要确认 Position的Robot
- 可以确认选择的Robot一行最前面的 "*" 符号

```
<SELECT ROBOT>
ROBOT TYPE
*R1      Q4AFL1
R2      NO-EXIT
R3      NO-EXIT
```

SEL

EXIT



- 按下"F1" 键移动到 POS 画面
- 按下"F4" 键恢复到上一个画面

Step 3.

CURR 画面确认

```
<AUTO : CURR>
F: test4      S:2      A
T: 1          V: 5      Q4AFL1
a = 0
T : 0          Z : 0
R1:            R2: 0
X : 0          6 : 0
```

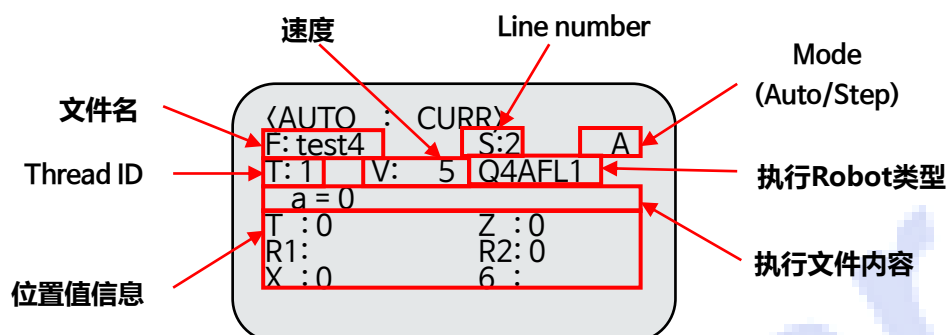
- Current 画面确认

■ 功能及画面说明

① 画面说明

- 1) 可以确认位置值信息栏中的当前位置值信息.

2) 位置值信息栏中没有输出说明轴Disable.



② 功能说明

- **[ESC]:** 移动到上一级画面

3.5.3.2 POS (Current Position)

- 可以确认Robot各轴的Teaching位置 (Local position).

Step 1.

POS 画面移动

```
< AUTO: DISP> (1/2)
1. CURR      2. POS
3. INT       4. FLOAT
5. GPIO      6. FBUS
7. SAFETY    8. ULOG
```

input : #



- 选择2: POS

Step 2.

选择ROBOT

```
<SELECT ROBOT>
ROBOT TYPE
*R1      Q4AFL1
R2      NO-EXIT
R3      NO-EXIT
```

SEL

EXIT



- 按下 "UP/DOWN" 键选择要确认 Position的Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

```
<SELECT ROBOT>
ROBOT TYPE
*R1      Q4AFL1
R2      NO-EXIT
R3      NO-EXIT
```

SEL

EXIT



- 按下"F1" 键移动到 POS 画面



- 按下"F4" 键移动到上一个画面

Step 3.

POS 画面确认

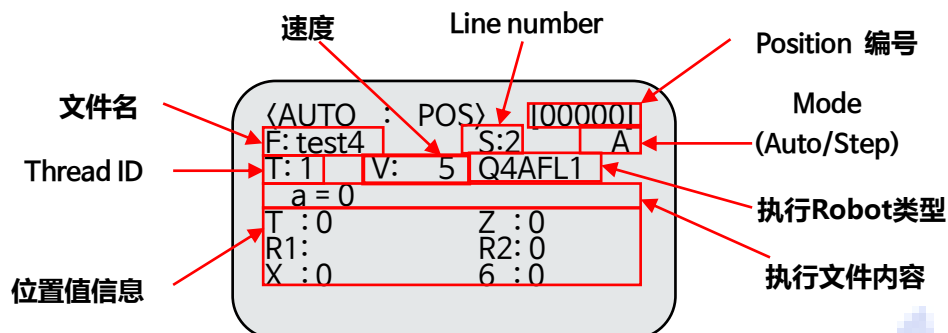
```
<AUTO : POS> [00000]
F: test4      S:2      A
T: 1          V: 5      Q4AFL1
a = 0
T : 0          Z : 0
R1:            R2: 0
X : 0          6 : 0
```

- Local position 画面确认

■ 功能及画面说明

① 画面说明

- 可以确认位置值信息栏中的当前位置值信息.
- 位置值信息栏中没有输出说明轴Disable.



② 功能说明

- **[PAGE UP/DOWN]**: 按顺序移动到所需索引
- **[JUMP]**: 直接移动到所需索引

Step 1

索引直接移动

```

<AUTO : POS> [00000]
F: test4 S:2 A
T: 1 V: 5 Q4AFL1
a = 0
T : 0 Z : 0
R1: R2: 0
X : 0 6 : 0
  
```



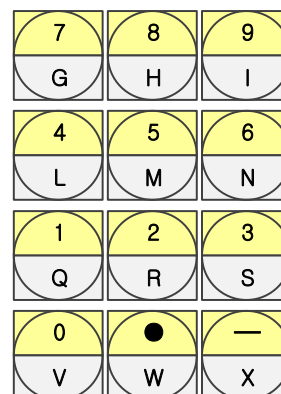
- 直接移动到输入的Point编号时按下 "JUMP" 键

```

<AUTO : POS> [00000]
F: test4 S:2 A
T: 1 V: 5 Q4AFL1
a = 0
T : 0 Z : 0
R1: R2: 0
X : 0 6 : 0
Point # = 10
  
```



- 输入要移动的Point编号“10”



- 输入完成后按下“ENTER”键

- **[ESC]**: 移动到上一级画面

3.5.3.3 INT (Integer)

- 可以确认Edit中编辑的Integer 值.

Step 1.

GLBINT 画面移动

< AUTO: DISP > (1/2)
 1. CURR 2. POS
 3. **INT** 4. FLOAT
 5. GPIO 6. FBUS
 7. SAFETY 8. ULOG

input : #



- 选择3: INT

Step 2.

< AUTO: **INTEGER** >

I0	■
I1	0
I2	0
I3	0
I4	0
I5	0

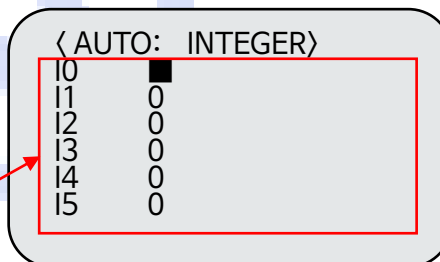
- INTEGER 画面确认

■ 功能及画面说明

① 画面说明

- Integer 信息栏中可以确认 Integer 索引和相应值.

Integer 信息



② 功能说明

- [UP/DOWN]: 顺序移动到编号

Step 1.

Cursor位置移动

< AUTO: INTEGER >

I0	■
I1	0
I2	0
I3	0
I4	0
I5	0



- 按下方向键移动到要编辑的值
- 可以通过Cursor位置确认移动结果

< AUTO: INTEGER>

10	0
11	0
12	0
13	0
14	0
15	0

- **[PAGE UP/DOWN]:** 移动到索引编号页面

Step 1.

< AUTO: INTEGER>

10	0
11	0
12	0
13	0
14	0
15	0

Pg Up

- 可以利用“PAGE UP/DOWN” 键切换画面

< AUTO: INTEGER>

16	0
17	0
18	0
19	0
110	0
111	0

Pg Dn

- **[JUMP]:** 直接移动到索引编号

Step 1

直接移动到索引

< AUTO: INTEGER>

10	0
11	0
12	0
13	0
14	0
15	0

JUMP

- 直接移动到输入的Point编号时按下 “JUMP”键

- 输入要移动的Point编号“10”

< AUTO: INTEGER>

10	0
11	0
12	0
13	0
14	0
15	0

Integer Num # =10

ENTER

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

< AUTO: INTEGER>

~~10~~ ■
I1 0
I2 0
I3 0
I4 0
I5 0

- 输入完成后按下“ENTER”键

• 编辑Integer值

Step 1.

变量值变更

< AUTO: INTEGER>

~~10~~ ■
I1 0
I2 0
I3 0
I4 0
I5 0

- 按下要变更的值数字键输入变更值 “10”

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

< AUTO: INTEGER>

~~10~~ 10 ■
I1 0
I2 0
I3 0
I4 0
I5 0



- 按下“ENTER” 键反映到画面

• 删除Integer值

Step 1.

删除变量值

< AUTO: INTEGER>

~~10~~ 10 ■
I1 0
I2 0
I3 0
I4 0
I5 0



- 错误输入变量值时按下“DEL” 键，按顺序删除输入值。

< AUTO: INTEGER>

I0	1
I1	0
I2	0
I3	0
I4	0
I5	0

- **[ESC]:** 移动到上一级画面

< AUTO: INTEGER>

I0	0
I1	0
I2	0
I3	0
I4	0
I5	0

ESC

- 确认结果

< AUTO : INS>

F: test4 S:2 A

T: 1 V: 5 Q4AFL1

a = 0

if a == 0
b = 1

3.5.3.4 FLOAT(Float)

- 可以确认Edit中编辑的Float 值.

Step 1.

GLBINT 画面移动

< AUTO: DISP > (1/2)
 1. CURR 2. POS
 3. INT **4. FLOAT**
 5. GPIO 6. FBUS
 7. SAFETY 8. ULOG

input : #



- 4. FLOAT选择

Step 2.

< AUTO: **FLOAT** >

F0 ■
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0

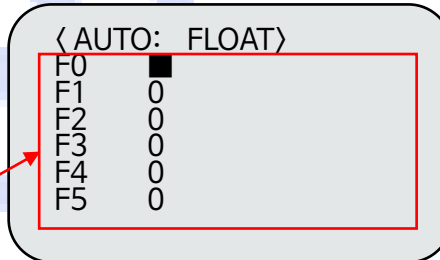
- FLOAT 画面确认

■ 功能及画面说明

① 画面说明

- 可以确认Float 新西兰中的Float 索引和相应值.

Float 信息



② 功能说明

- [UP/DOWN]: 按编号顺序移动

Step 1.

Cursor位置移动

< AUTO: FLOAT >

F0 ■
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0



- 按下方向键移动到要编辑的值
- 通过Cursor位置确认移动结果

< AUTO: FLOAT >

F0 0
F1 0
 F2 0
 F3 0
 F4 0
 F5 0

- [PAGE UP/DOWN]: 移动到Index编号页面

Step 1.

< AUTO: FLOAT >

F0 0
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0

Pg Up

- 可以利用"PAGE UP/DOWN" 键切换画面

< AUTO: FLOAT >

F6 0
 F7 0
 F8 0
 F9 0
 F10 0
 F11 0

Pg Dn

- [JUMP]: 直接移动到Index编号

Step 1

直接移动到Index

< AUTO: FLOAT >

F0 0
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0

JUMP

- 直接移动到输入的Point编号时按下"JUMP"键

< AUTO: FLOAT >

F0 0
 F1 0
 F2 0
 F3 0
 F4 0
 F5 0
Float Num # =10

ENTER

- 输入要移动的Point编号"10"

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

< AUTO: FLOAT >

F10 ■
F11 0
F12 0
F13 0
F14 0
F15 0

- 输入结束后按下“ENTER”键

- 编辑Float值

Step 1.

变量值变更

< AUTO: FLOAT >

~~F0~~ ■
F1 0
F2 0
F3 0
F4 0
F5 0

- 按下要变更的值数字键输入“10.0”

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

< AUTO: FLOAT >

~~F0~~ 10.0 ■
F1 0
F2 0
F3 0
F4 0
F5 0

ENTER

- 按下“ENTER”键反映到画面

- 删除Float值

Step 1.

删除变量值

< AUTO: FLOAT >

~~F0~~ 10.0 ■
F1 0
F2 0
F3 0
F4 0
F5 0

DEL
RST

- 错误输入变量值时按下“DEL”键按顺序删除输入的值。

< AUTO: FLOAT >

F0 10.■

F1 0

F2 0

F3 0

F4 0

F5 0

- [ESC]: 移动到上一级画面

< AUTO: FLOAT >

F0 ■

F1 0

F2 0

F3 0

F4 0

F5 0

ESC

- 确认结果

< AUTO : INS >

F: test4 S:2 A

T:1 V: 5 Q4AFL1

a = 0

if a == 0

b = 1

3.5.3.5 GPIO

- 可以确认GPIO的 I/O 值.

Step 1.

GPIO 画面移动

〈 AUTO: DISP〉 (1/2)
 1. CURR 2. POS
 3. INT 4. FLOAT
 5. **GPIO** 6. FBUS
 7. SAFETY 8. ULOG

input : #



- 选择5:GPIO

Step 2.

〈 I/O - GPIO〉
 1. IN 2. OUT

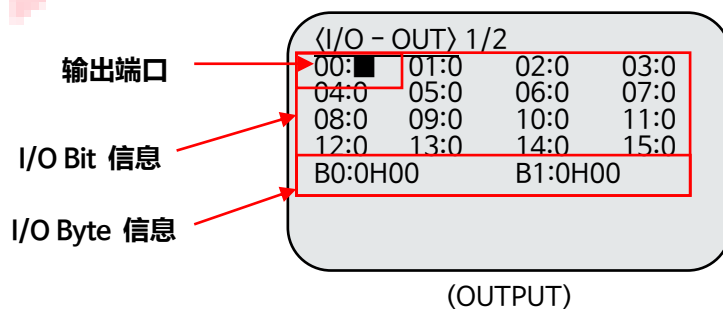
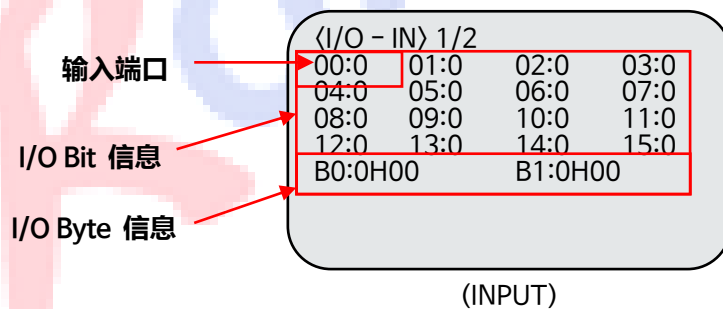
input #

- GPIO 画面确认

■ 功能及画面说明

① 画面说明

- 1) GPIO 信息栏中可以确认当前Display 窗口的输出Port和各Index相应的GPIO Pin输出或输入值.



② 功能说明

- [PAGE UP/DOWN]: 输入输出信号确认方法

Step 1.

IN/OUT Index页面变更

<I/O - OUT> 1/2

00:0	01:0	02:0	03:0
04:0	05:0	06:0	07:0
08:0	09:0	10:0	11:0
12:0	13:0	14:0	15:0
B0:0H00		B1:0H00	



- 可以利用“PAGE UP/DOWN”键切换画面



<I/O - OUT> 2/2

16:0	17:0	18:0	19:0
20:0	21:0	22:0	23:0
24:0	25:0	26:0	27:0
28:0	29:0	30:0	31:0
B2:0H00		B3:0H00	

- [UP/DOWN]: 选择输出的Port

Step 2.

信号位置移动

<I/O - OUT> 2/2

16:0	17:0	18:0	19:0
20:0	21:0	22:0	23:0
24:0	25:0	26:0	27:0
28:0	29:0	30:0	31:0
B2:0H00		B3:0H00	



- 按下方向键移动到要输出的编号
- 通过Cursor位置可以确认移动的结果



<I/O - OUT> 2/2

16:0	17:0	18:0	19:0
20:0	21:0	22:0	23:0
24:0	25:0	26:0	27:0
28:0	29:0	30:0	31:0
B2:0H00		B3:0H00	

- 输出信号 ON/OFF

Step 3.

信号 ON

```

<I/O - OUT> 2/2
16:0  17:0  18:0  19:0
20:0  21:0  22:0  23:0
24:0  25:0  26:0  27:0
28:0  29:0  30:0  31:0
B2:0H00  B3:0H00

```

EXIT

ENTER

- 开启信号时按下“ENTER”键

```

<I/O - OUT> 2/2
16:0  17:0  18:0  19:0
20:1  21:0  22:0  23:0
24:0  25:0  26:0  27:0
28:0  29:0  30:0  31:0
B2:0H00  B3:0H00

```

EXIT

信号OFF

```

<I/O - OUT> 2/2
16:0  17:0  18:0  19:0
20:1  21:0  22:0  23:0
24:0  25:0  26:0  27:0
28:0  29:0  30:0  31:0
B2:0H00  B3:0H00

```

EXIT

ENTER

- 关闭信号时按下“ENTER” 键

```

<I/O - OUT> 2/2
16:0  17:0  18:0  19:0
20:0  21:0  22:0  23:0
24:0  25:0  26:0  27:0
28:0  29:0  30:0  31:0
B2:0H00  B3:0H00

```

EXIT

- [ESC]: 移动到上一级画面

Step 4.

```

<I/O - GPIO>
1. IN      2. OUT

```

input #

ESC

- 确认结果

```

<AUTO : INS>
F: test4      S:2      A
T: 1          V: 5      Q4AFL1
a = 0
if a == 0
b = 1

```

3.5.3.6 FBUS (Fieldbus)

- 可以确认Fieldbus的 BIT, WORD 值.

Step 1.

FBUS 画面移动

〈AUTO: DISP〉 (1/2)
 1. CURR 2. POS
 3. INT 4. FLOAT
 5. GPIO 6. **FBUS**
 7. SAFETY 8. ULOG

input : #



- 选择6: FBUS

〈FIELDBUS〉
 1. B-IN 2. B-OUT
 3. W-IN 4. W-OUT
 5. F-IN 6. F-OUT

input : #

- FBUS 画面确认

Step 2.

FBUS 未连接时

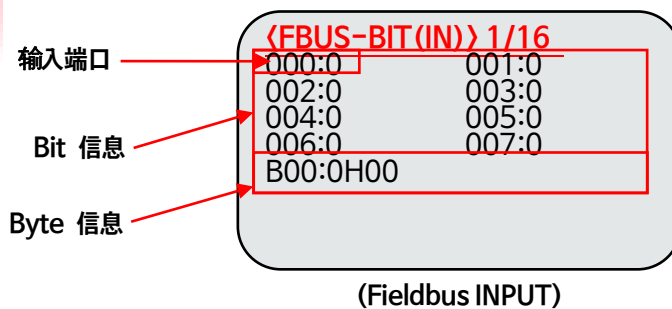
〈AUTO : FBUS〉 (1/2)
 1. INS 2. POS
 3. RPM 4. TRQ
 5. TRQMAX 6. GPIO
 7. SAFETY 8. FBUS

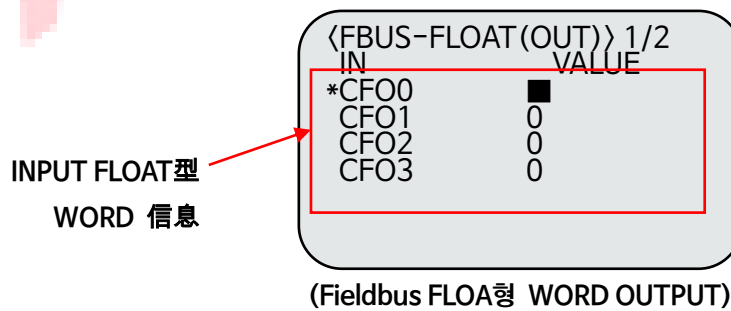
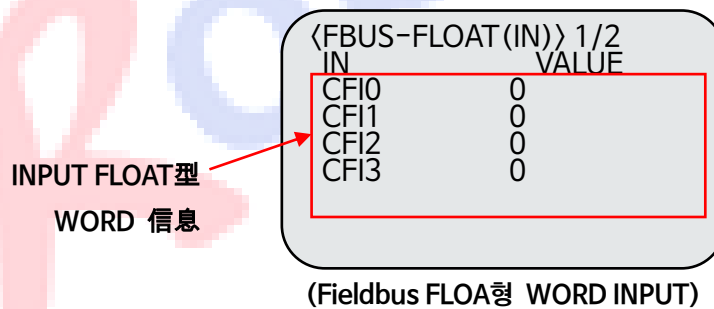
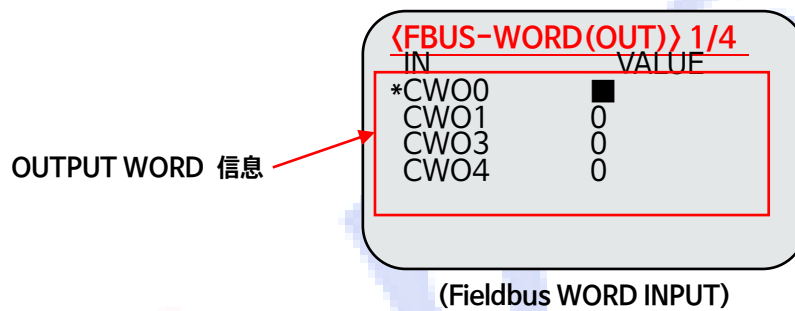
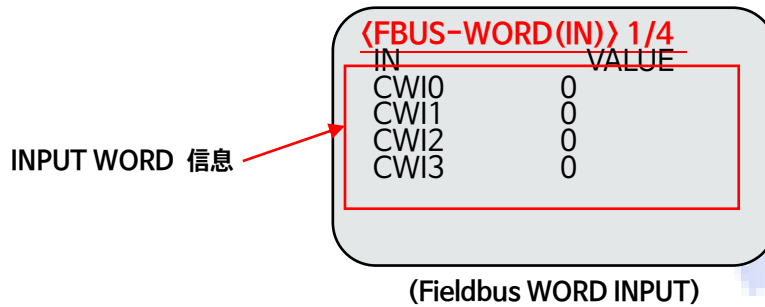
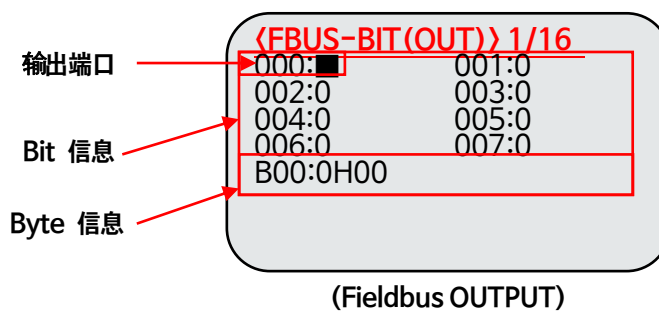
- 没有连接Fieldbus 时画面无变化.

■ 功能及画面说明

① 画面说明

- Fieldbus 信息栏中可以确认 Fieldbus Index信息及输出或输入信息.





② 功能说明

• [PAGE UP/DOWN]: 输入输出信号确认方法

Step 1.

BIT (IN/OUT) 索引页面变更

<FBUS-BIT (IN)> 1/16

000:0	001:0
002:0	003:0
004:0	005:0
006:0	007:0
B00:0H00	

Pg Up

- 利用"PAGE UP/DOWN"键切换画面

<FBUS-BIT (IN)> 2/16

008:0	009:0
010:0	011:0
012:0	013:0
014:0	015:0
B01:0H00	

Pg Dn

WORD/FLOAT索引页面变更

<FBUS-WORD (IN)> 1/4

IN	VALUE
CW10	0
CW11	0
CW12	0
CW13	0

EXIT

Pg Up

- WORD DATA 区域显示 0~15 范围.

<FBUS-WORD (IN)> 4/4

IN	VALUE
CW112	0
CW113	0
CW114	0
CW115	0

EXIT

Pg Dn

• [UP/DOWN]: 选择输出的端口

Step 2.

BIT位置移动

<FBUS-BIT (OUT)> 3/16

016:0	017:0
018:0	019:0
020:0	021:0
022:0	023:0
B02:0H00	



- 按下方向键移动到要输出的编号
- 通过游标的位置可以确认移动结果

⟨FBUS-BIT(OUT)⟩ 3/16
 017:0 018:0
 019:0 020:0
 021:0 022:0
 023:0 024:0
 B02:0H00



WORD/FLOAT 位置移动

⟨FBUS-WORD(OUT)⟩ 2/4
 IN VALUE
 *CWO4 0
 CWO5 0
 CWO6 0
 CWO7 0

SET EXIT



- 按下方向键移动到要输出的编号



- 通过游标的位置可以确认移动结果

⟨FBUS-WORD(OUT)⟩ 2/4
 IN VALUE
 CWO4 0
 *CWO5 0
 CWO6 0
 CWO7 0

SET EXIT

- [JUMP]: 移动到相应Bit或Port.

Step 3.

BIT位置移动

⟨FBUS-BIT(IN)⟩ 1/16
 000:0 001:0
 002:0 003:0
 004:0 005:0
 006:0 007:0
 B00:0H00

fbus bit = 120

⟨FBUS-BIT(IN)⟩ 1/16
 120:0 121:0
 122:0 123:0
 124:0 125:0
 126:0 127:0
 B00:0H00



- 按下Jump 按钮输入要移动的bit 值

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

WORD/FLOAT 移动

```

<FBUS-WORD(IN)> 1/4
IN      VALUE
CWI0    0
CWI1    0
CWI2    0
CWI3    0

```

fbus_port = 12

```

<FBUS-WORD(IN)> 4/4
IN      VALUE
CWI12   0
CWI13   0
CWI14   0
CWI15   0

```

EXIT



- 按下Jump 按钮输入要移动的port值

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

• 输出 BIT ON/OFF

Step 4.

信号 ON

```

FBUS-BIT(OUT)> 3/16
017:0          018:0
019:0          020:0
021:0          022:0
023:0          024:0
B02:0H00

```



- 开启信号时按下“ENTER” 键

```

FBUS-BIT(OUT)> 3/16
017:0          018:0
019:0          020:1
021:0          022:0
023:0          024:0
B02:0H00

```

信号 OFF

```

FBUS-BIT(OUT)> 3/16
017:0          018:0
019:0          020:1
021:0          022:0
023:0          024:0
B02:0H00

```



- 关闭信号时按下“ENTER” 键

```

FBUS-BIT(OUT)> 3/16
017:0          018:0
019:0          020:0
021:0          022:0
023:0          024:0
B02:0H00

```


编辑OUTPUT WORD值

OUT	VALUE
CWO1	0
CWO2	0
CWO3	0
CWO4	0
CWO5	0

OUT	VALUE
CWO1	0
CWO2	0
CWO3	10
CWO4	0
CWO5	0

• 编辑值

7	8	9
G	H	I
4	5	6
L	M	N
1	2	3
Q	R	S
0	●	—
V	W	X

3.5.3.7 SAFETY

- 可以确认SAFETY 状态.

Step 1.

SAFETY 画面移动

<<AUTO: DISP>> (1/2)
 1.CURR 2. POS
 3. INT 4. FLOAT
 5. GPIO 6. FBUS
7. SAFETY 8. ULOG

input : #



- 选择7: SAFETY

Step 2.

<I/O - SAFETY>
 1. IN 2. OUT

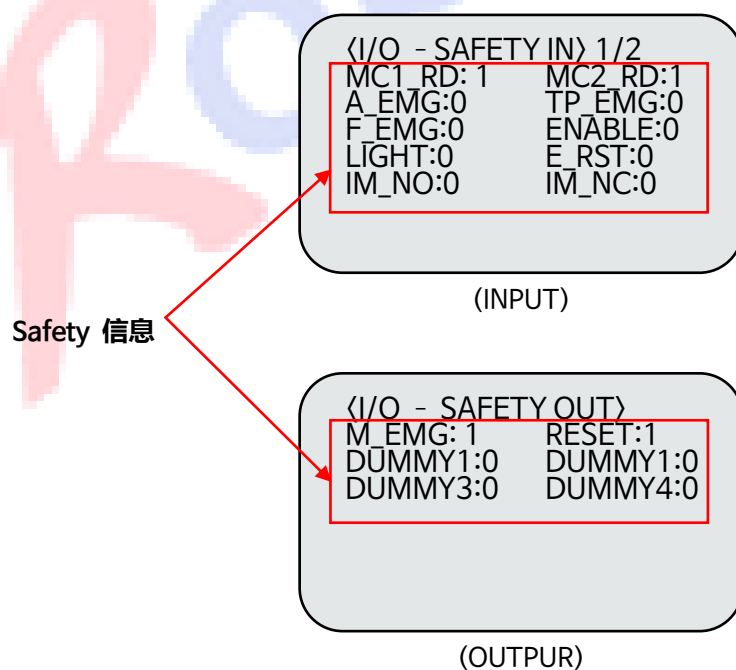
input : #

- SAFETY 画面确认

■ 功能及画面说明

① 画面说明

- 1) Safety 信息栏中可以确认Safety 输出或输入信息.



② 功能说明

• [PAGE UP/DOWN]: 输入信号确认方法

Step 1.

页面变更

⟨I/O - Safety IN⟩ **1/2**
 MC1_RD:1 MC2_RD:1
 A_EMG:0 TP_EMG:0
 F_EMG:0 ENABLE:0
 LIGHT:0 E_RST:0
 IM_NO:0 IM_NC:0
 EXIT



- 可以利用“PAGE UP/DOWN”键切换画面

⟨I/O - Safety IN⟩ **2/2**
 SMPS_F:1 FAN_F:1
 DUMMY2:0 DUMMY6:0
 DUMMY7:0 DUMMY8:0
 EXIT



• [ESC]: 移动到上一级画面

⟨I/O - SAFETY⟩
 1. IN 2. OUT

input #



- 确认结果

⟨AUTO : INS⟩
 F: test4 S:2 A
 T:1 V: 5 Q4AFL1
 a = 0
 if a == 0
 b = 1

③ SAFETY 信息说明

Port		说明
IN	MC1_RD	检查MC1 状态
	MC2_RD	检查MC2 状态
	A_EMG	Auto模式时的Alarm
	M_EMG	T/P Alarm

	S_EMG	Front 开关Alarm
	TP_EMG	设备 Enable or Hold
	F_EMG	CURTAIN感知
	ENABLE	外部RESET
	LIGHT	T/P 模式和设备模式不一致时Alarm (Manual)
	E_RST	T/P 模式和设备模式不一致时Alarm (Auto)
	IM_NO	UPS AC 电源不接入时的Alarm
	IM_NC	Fan Alarm
	SMPS_F	DUMMY
	FAN_F0	DUMMY
	DUMMY2	DUMMY
	DUMMY6	DUMMY
	DUMMY7	MC1 状态检查
	DUMMY8	MC2 状态检查
OUT	M_EMG	Main Alarm
	RESET	Reset
	DUMMY1	DUMMY
	DUMMY3	DUMMY
	DUMMY4	DUMMY
	DUMMY5	DUMMY

3.5.3.8 ULOG (User Log)

- 可以确认用户JOB log.

Step 1.

SAFETY 画面移动

(AUTO: DISP) (1/2)
 1. CURR 2. POS
 3. INT 4. FLOAT
 5. GPIO 6. FBUS
 7. SAFETY 8. **ULOG**

input : #



- 选择8: ULOG

Step 2.

<MAN : ULOG>

000: ■
 001: —
 002:
 003:
 004:
 005:
 EXIT START RE-ST WAY

- ULOG 画面确认

■ 功能及画面说明

① 画面说明

- 用户在JOB 上使用 PRINT 指令确认保存在内存的Log.

Float 信息

<MAN : ULOG>

000: ■
 001: —
 002:
 003:
 004:
 005:
 EXIT START RE-ST WAY

② 功能说明

- [UP/DOWN]: 按编号顺序移动

Step 1.

Cursor位置移动

<MAN : ULOG>

000: ■
 001:
 002:
 003:
 004:
 005:
 EXIT START RE-ST WAY



- 按下方向键移动到要确认的值
- 移动到Cursor位置确认结果

```

<MAN : ULOG>
000:
001: ■
002:
003:
004:
005:
EXIT START RE-ST WAY

```

- **[PAGE UP/DOWN]:** Index编号页面移动

Step 1.

```

<MAN : ULOG>
000:
001: ■
002:
003:
004:
005:
EXIT START RE-ST WAY

```



- 可以利用“PAGE UP/DOWN” 键切换画面

```

<MAN : ULOG>
006:
007: ■
008:
009:
010:
011:
EXIT START RE-ST WAY

```



- **[JUMP]:** 直接移动到编号

Step 1

Index直接移动

```

<MAN : ULOG>
000:
001: ■
002:
003:
004:
005:
EXIT START RE-ST WAY

```



- 直接移动到输入的Point编号时按下 “JUMP”键

```

<MAN : ULOG>
000:
001:
002:
003:
004:
005:
Float Num # =10

```



- 输入要移动的Point编号“10”

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

<MAN : ULOG>

010: ■

011:

012:

013:

014:

015:

EXIT START RE-ST WAY

- 输入完成后按下“ENTER”键

- **[ESC]**: 移动到上一级画面

<MAN : ULOG>

010: ■

011:

012:

013:

014:

015:

EXIT START RE-ST WAY

<MAN : INS>

F: test4 S:2 A

T: 1 V: 5 Q4AFL1

a = 0

if a == 0

b = 1

ESC

- 确认结果

3.5.3.9 TRQ (Torque)

- 可以确认Robot各轴的Torque.

Step 1.

TRQ 画面移动

〈AUTO : DISP〉 (2/2)	
1. TRQ	2. TRQMAX
3. LOAD	4. RPM
5. DIST	6. JOB
7. MVR	8. THREAD
input : #	



- 选择1: TRQ

Step 2.

选择ROBOT

〈SELECT ROBOT〉	
ROBOT TYPE	
* R1	Q4AFL1
R2	NO-EXIT
R3	NO-EXIT
SEL	EXIT



- 按下"UP/DOWN" 键选择要确认Position的Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

〈SELECT ROBOT〉	
ROBOT TYPE	
*R1	Q4AFL1
R2	NO-EXIT
R3	NO-EXIT
SEL	EXIT



- 按下"F1" 键移动到TRQ 画面



- 按下"F4" 键恢复到上一个画面

Step 3.

TRQ 画面确认

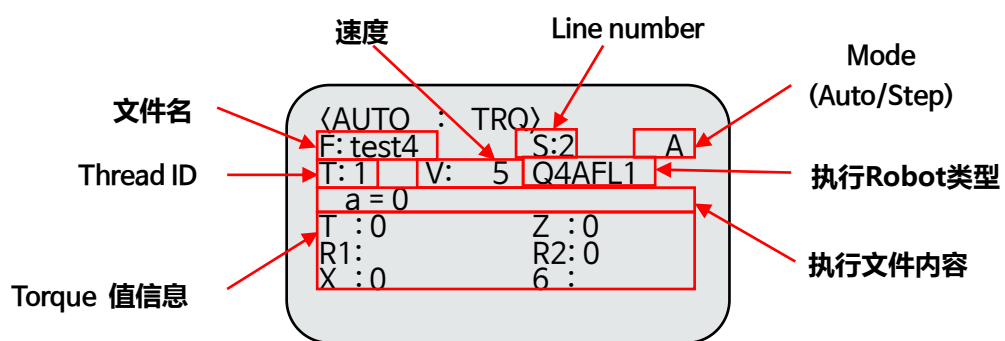
〈AUTO : TRQ 〉	
F: test4	S:2 A
T: 1 V: 5	Q4AFL1
a = 0	
T1: 0	Z1: 0
R1: 0	R2: 0
X1: 0	G1: 0

- TRQ 画面确认

■ 功能及画面说明

① 画面说明

- 1) Torque 值信息栏中可以确认当前Torque 值.
- 2) Torque 值信息栏中不输出值时说明轴Disable.



② 功能说明

- **[ESC]**: 移动到上一级画面

3.5.3.10 TRQMAX (Max Torque)

- 可以确认Robot各轴的最大扭矩。

Step 1.

TRQMAX 画面移动

⟨AUTO : DISP⟩ (2/2)
1. TRQ 2. **TRQMAX**
3. LOAD 4. RPM
5. DIST 6. JOB
7. MVR 8. THREAD

input : #



- 选择2: TRQMAX

Step 2.

选择ROBOT

⟨SELECT ROBOT⟩
ROBOT TYPE
***R1** **Q4AFL1**
~~R2~~ NO-EXIT
R3 NO-EXIT

SEL

EXIT



- 按下"UP/DOWN"键选择要确认Position的Robot



- 可以确认选择的Robot一行最前面的"*"符号

⟨SELECT ROBOT⟩
ROBOT TYPE
*R1 Q4AFL1
R2 NO-EXIT
R3 NO-EXIT

SEL

EXIT



- 按下"F1"键移动到 TRQMAX 画面



- 按下"F4"键恢复到上一个画面

Step 3.

TRQMAX 画面确认

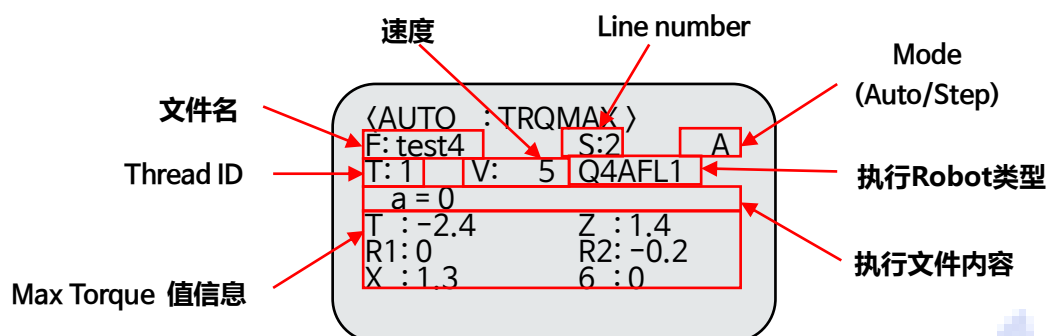
⟨AUTO : **TRQMAX**⟩
F: test4 S: 2 A
T: 1 V: 5 Q4AFL1
a = 0
T1: -2.4 **Z1: 1.4**
R1: 0 **R2: -0.2**
X1: 1.3 **G1: 0**
DISP

- TRQMAX 画面确认

■ 功能及画面说明

① 画面说明

- Max Torque值信息栏中可以确认当前Max Torque值。
- Max Torque值信息栏中不输出值时说明轴Disable。



② 功能说明

- **[ESC]:** 移动到上一级画面

3.5.3.11 LOAD (Over Load)

- 可以确认各轴的平均负荷率.

Step 1.

LOAD 画面移动

〈AUTO : DISP〉 (2/2)
 1. TRQ 2. TRQMAX
3. LOAD 4. RPM
 5. DIST 6. JOB
 7. MVR 8. THREAD

input : #



- 选择3: LOAD

Step 2.

选择ROBOT

〈SELECT ROBOT〉
 ROBOT TYPE
***R1** Q4AFL1
 R2 NO-EXIT
 R3 NO-EXIT

SEL EXIT



- 按下"UP/DOWN" 键移动到确认平均负荷率的Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

〈SELECT ROBOT〉
 ROBOT TYPE
***R1** Q4AFL1
 R2 NO-EXIT
 R3 NO-EXIT

SEL **EXIT**



- 按下"F1" 键移动到 LOAD 画面



- 按下"F4" 键恢复到上一个画面

Step 3.

LOAD 画面确认

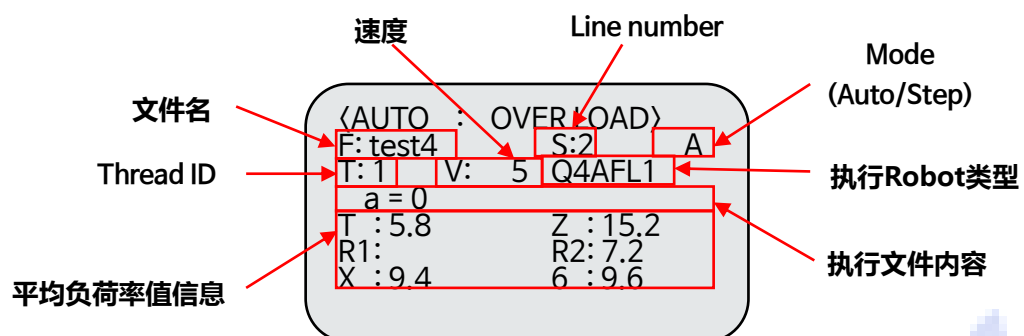
〈AUTO : **OVER LOAD**〉
 F: test4 S:2 A
 T: 1 V: 5 Q4AFL1
 a = 0
 T : 0 Z : 0
 R1: 0 R2: 0
 X : 0 6 : 0

- Over load 画面确认

■ 功能及画面说明

① 画面说明

- 1) 平均负荷率值新画面中可以确认负荷率值.
- 2) 平均负荷率值信息栏中不输出值时说明轴Disable.
- 3) 平均负荷率值输出为0时说明没有完成Driver F/W的升级.



② 功能说明

- **[ESC]:** 移动到上一级画面

3.5.3.12 RPM

- 可以确认Robot各轴的RPM速度.

Step 1.

RPM 画面移动

```

<AUTO : DISP>      (2/2)
1. TRQ              2. TRQMAX
3. LOAD              4: RPM
5. DIST              6: JOB
7. MVR               8. THREAD
  
```

input : #



- 选择4: RPM

Step 2.

ROBOT 选择

```

<SELECT ROBOT>
ROBOT TYPE
*R1              Q4AFL1
R2              NO-EXIT
R3              NO-EXIT
  
```

SEL

EXIT



- 按下 "UP/DOWN" 键选择要确认 Position的Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

```

<SELECT ROBOT>
ROBOT TYPE
*R1              Q4AFL1
R2              NO-EXIT
R3              NO-EXIT
  
```

SEL

EXIT



- 按下"F1" 键移动到RPM 画面



- 按下"F4" 键恢复到上一个画面

Step 3.

RPM 画面确认

```

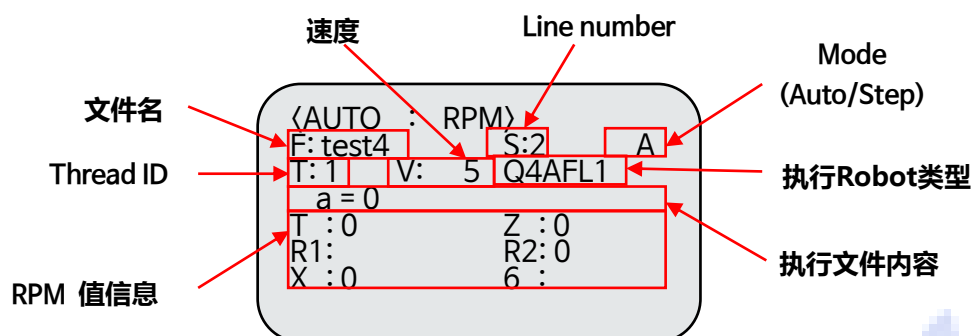
<AUTO : RPM>
F: test4          S:2      A
T: 1              V: 5    Q4AFL1
a = 0
T1: 0             Z1: 0
R1:               R2: 0
X1: 0             6T:
  
```

- RPM 画面确认

■ 功能及画面说明

① 画面说明

- 1) RPM 值信息栏中可以确认RPM 值.
- 2) RPM 值信息栏中不输出值时说明轴Disable.



② 功能说明

- **[ESC]**: 移动到上一级画面

3.5.3.13 DIST (Distance)

- 可以确认Robot各轴的累积距离。

Step 1.

DIST 画面移动

{AUTO : DISP} (2/2)
 1. TRQ 2. TRQMAX
 3. LOAD 4: RPM
 5. **DIST** 6: JOB
 7. MVR 8. THREAD

input : #



- 选择5: DIST

Step 2.

选择ROBOT

{SELECT ROBOT}
 ROBOT TYPE
 *R1 Q4AFL1
 R2 NO-EXIT
 R3 NO-EXIT

SEL

EXIT



- "UP/DOWN"键选择要确认Position的Robot



- 可以确认选择的Robot一行最前面的"*" 符号

{SELECT ROBOT}
 ROBOT TYPE
 *R1 Q4AFL1
 R2 NO-EXIT
 R3 NO-EXIT

SEL

EXIT



- 按下"F1" 键移动到DIST 画面



- 按下"F4" 键恢复到上一个画面

Step 3.

DIST 画面确认

{AUTO : **DIST**}
 F: test4 S:2 A
 T: 1 V: 5 Q4AFL1
 a = 0
 T1:0 Z1:0
 R1: R2:0
 X1:0 61:

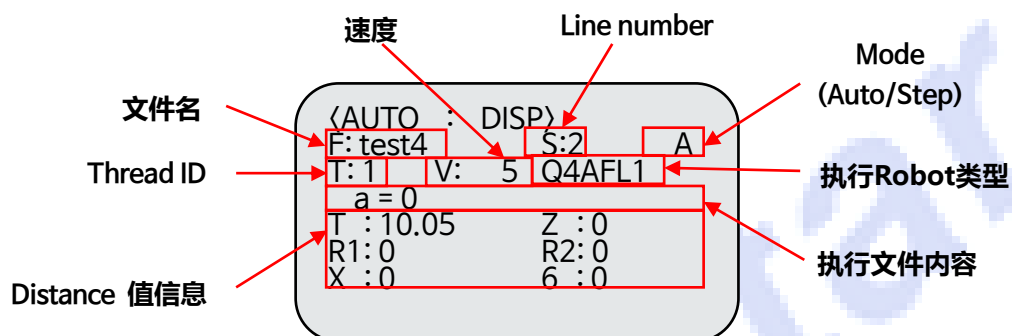
- DIST 画面确认

■ 功能及画面说明

① 画面说明

- Distance值信息栏中可以确认累积距离值。
- Distance值信息栏中不输出值时说明轴Disable。

- 3) 直线轴时的单位: Km
- 4) 旋转轴时的单位: 减速器基准旋转数1,000,000次为单位
 - 因Teach pendant 位数限定, 如下表示
 - Ex) T 轴旋转10,050,000次时显示在画面的内容: 10.05 (小数点 2位以下省略)



② 功能说明

- **[ESC]:** 移动到上一级画面

3.5.3.14 JOB

- 可以确认当前执行的JOB信息.

Step 1.

JOB 画面移动

(AUTO : DISP) (2/2)
 1. TRQ 2. TRQMAX
 3. LOAD 4. RPM
 5. DIST **6. JOB**
 7. MVR 8. THREAD

 input : #

(AUTO : JOB)
 T1 test4 — main

 DISP



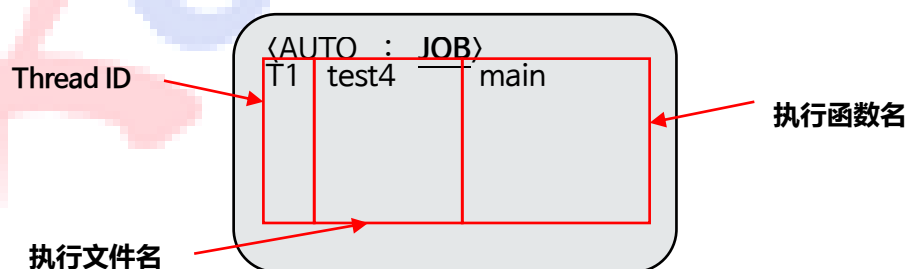
- 选择6: JOB

- JOB 画面确认

■ 功能及画面说明

① 画面说明

- 1) Thread ID 栏中可以确认ID.
- 2) 执行文件名栏中可以确认当前执行的文件名称.
- 3) 执行函数名栏中可以确认当前执行的函数名.



② 功能说明

- **[ESC]**: 移动到上一级画面

3.5.3.15 MVR (Move Rate)

- 移动两个POINT 之间时，可以确认当前位置对整体距离的百分比.

Step 1.

MVR 画面移动

<AUTO : DISP> (2/2)
 1. TRQ 2. TRQMAX
 3. LOAD 4: RPM
 5. DIST 6: JOB
7. MVR 8. THREAD

input : #



- 选择7: MVR

Step 2.

选择ROBOT

<SELECT ROBOT>
 ROBOT TYPE
*R1 Q4AFL1
 R2 NO-EXIT
 R3 NO-EXIT

SEL

EXIT



- 按下"UP/DOWN" 键 选择要确认 Position的Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

<SELECT ROBOT>
 ROBOT TYPE
 *R1 Q4AFL1
 R2 NO-EXIT
 R3 NO-EXIT

SEL

EXIT



- 按下"F1" 键移动到 MVR 画面



- 按下"F4" 键恢复到上一个画面

Step 3.

MVR 画面确认

<AUTO : MVR>
 F: test4 S:2 A
 T: 1 V: 5 Q4AFL1
 a = 0

MVR : 0

DISP

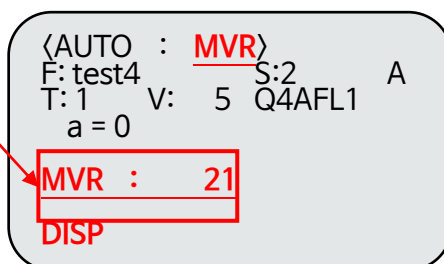
- MVR 画面确认

■ 功能及画面说明

① 画面说明

- 当前执行中的作业移动两个POINT 之间时，可以实时确认当前位置对整体距离的百分比.

Move Rate
(%)



⟨AUTO : MVR⟩
F: test4 S:2 A
T: 1 V: 5 Q4AFL1
a = 0
MVR : 21
DISP

② 功能说明

- [ESC]: 移动到上一级画面

3.5.3.16 THREAD

- 可以设置执行文件的Thread ID. 仅在MANUAL RUN 时激活.

Step 1.

THREAD 画面移动

<AUTO : DISP> (2/2)
 1. TRQ 2. TRQMAX
 3. LOAD 4: RPM
 5. DIST 6: JOB
 7. MVR **8. THREAD**

input : #



- 选择8: THREAD

< MANUAL: TH INFO>
 THREAD STATE
 *T H1 RUN
 TH2 RUN
 TH3 IDLE
 TH4 IDLE

SEL

EXIT

- THREAD 画面确认

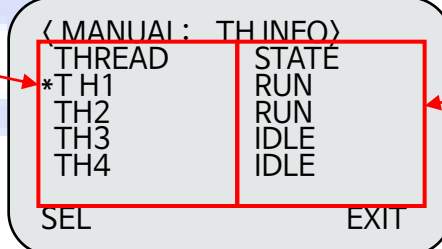
■ 功能及画面说明

① 画面说明

- 1) 确认当前执行中的Thread进行状态, 选择要监控的Thread.

Thread ID

*: 选择 ID



Thread 状态

② 功能说明

- [ESC]: 移动到上一级画面

3.5.3.17 17 INSHAPE

- 可以确认给予Motion的变化量.

Step 1.

THREAD 画面移动

〈AUTO : DISP〉 (3/3)
1. INSHAPE

input : #



- 选择1: INSHAPE

Step 2.

ROBOT 选择

〈SELECT ROBOT〉
ROBOT TYPE
*R1 Q4AFL1
R2 NO-EXIT
R3 NO-EXIT

SEL

EXIT



- 按下"UP/DOWN" 键选择确认 Position 的Robot



- 选择的Robot行首可以看到"*" 标识

〈SELECT ROBOT〉
ROBOT TYPE
*R1 Q4AFL1
R2 NO-EXIT
R3 NO-EXIT

SEL

EXIT



- 按下"F1"移动到MVR画面



- 按下"F4" 键恢复到之前的画面

Step 3.

INSHAPE 画面确认

〈AUTO : ISP〉
F: test4 S:2 A
T: 1 V: 5 Q4AFL1
a = 0
AXIS: T Used: OFF
T: 0.5 D: 1.5

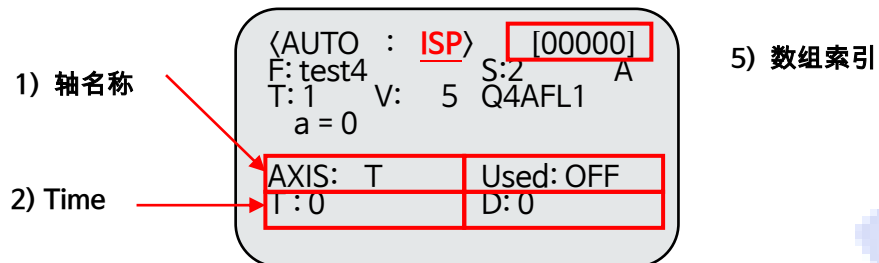
- INSHAPE 画面确认

■ 功能及画面说明

① 画面说明

- 1) 轴名称: 确认相应轴的名称.
- 2) Time (0~1): 总移动时间为1时, 确认位置变化产生的时间.
- 3) Distance: 表示位置变化量.

- 4) 使用情况: 表示使用情况.
- 5) 数组索引: 标识位置Point数组的索引, .

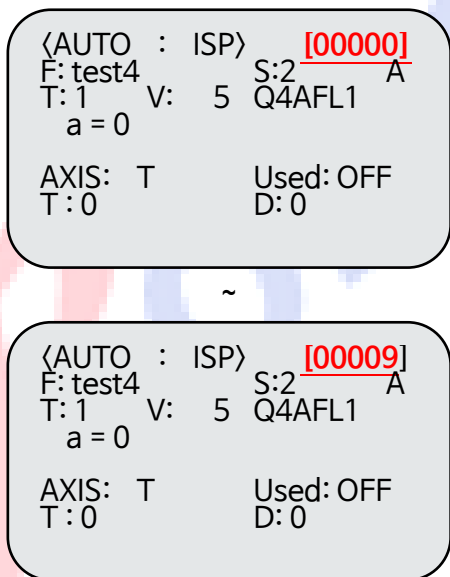


② 功能说明

- **[ESC]**: 移动到上一个画面
- **[PAGE UP/DOWN]**: 移动到索引编号页面(0~9凡违反)

Step 1.

移动索引页面



- 利用"PAGE UP/DOWN" 检可以移动索引

- **[AXIS CHG]**: 轴变更

Step 1.

轴变更

〈AUTO : ISP〉 [00000]
F: test4 S:2 A
T:1 V: 5 Q4AFL1
a = 0

AXIS: T Used: OFF
T:0 D:0



- 利用"AXIS CHG" 键可以切换轴

〈AUTO : ISP〉 [00000]
F: test4 S:2 A
T:1 V: 5 Q4AFL1
a = 0

AXIS: Z Used: OFF
T:0 D:0

第4章 TEACH

- 可以利用JOB动作操作Robot.
- JOG LOCAL POINT拥有Robot移动时位置相关的数据. Robot位置数据包含JOINT 坐标值，内置Robot位置数据.
- 每个Robot的位置数据分配了LOCAL POINT，只能在分配的Robot内使用.
- 每个Robot的最大保存个数为 40,000个.
- 大部分位置Teaching模式中，用户移动Robot并保存位置值，此时使用Teach Pendant [PGET]键保存更方便.
- 修正的位置信息退出JOG模式时自动永久保存.
(但，前提是位置信息保存在内存.)

4.1 概要

- ✓ Robot位置Teaching模式中根据Robot种类可以选择1~2 种坐标系.
- ✓ SCARA 类型的Robot使用[JOINT,直角]坐标系，其余类型的Robot只使用[JOINT]坐标系.
- ✓ HMOV: 用户可以指定HOME位置使用， HOME 位置可以指定HMOV 0~3. 执行ORIGIN 模式时，在HMOV 0 号输入位置信息并使用.
- ✓ ORIGIN 模式在电机种类为[INCREMENTAL, LINEAR]时执行.

<TEACH> ROBOT *R1 R2 R3	TYPE R2AWS1 NO-EXIT NO-EXIT
JOINT HOME	

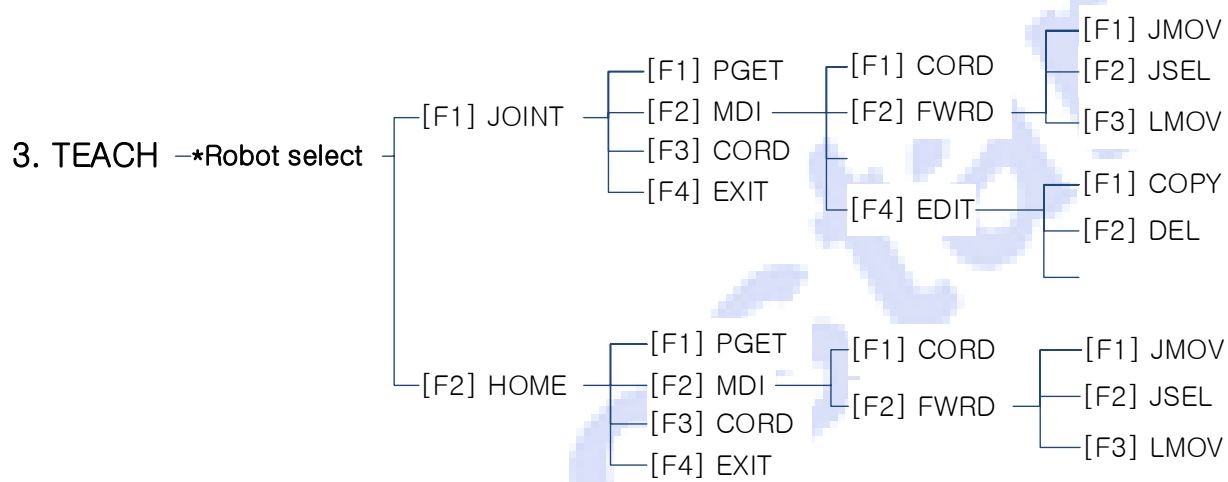
POINT种类

	GROUP	内容	备注
3. TEACH	[F1] JOINT	JOINT数据位置Teaching	
	[F2] HOME	HMOV数据位置Teaching	

4.2 TEACH 模式流程图

- ✓ JOG大体分为2个Group.
 - JOINT: 编辑JOINT数据位置变量或读取当前位置到JOINT数据.
 - HOME: 编辑HMOV数据位置变量或读取当前位置到 HMOV数据.
 - 可以获取或编辑位置数据.

■ TEACH Menu Tree



4.3 TEACH Group 画面 OPEN 顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. TEACH 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择JOG

<TEACH MENU>
1. EDIT 2. RUN
3. TEACH 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择“3. TEACH”

Step 3.

选择Robot

<TEACH>
ROBOT TYPE
*R1 R2AWS1
-R2 NO-EXIT
R3 NO-EXIT

JOINT HOME



- 按下“UP/DOWN” Cursor 键选择要编辑的参数的Robot



- 可以确认选择的Robot一行最前面的“*”符号

选择JOINT 数据

<TEACH>
ROBOT TYPE
*R1 R2AWS1
-R2 NO-EXIT
R3 NO-EXIT

JOINT HOME



- 编辑或获取JOINT 数据时按下 “F1” 键选择 “JOINT”。

选择HMOV 数据

< TEACH >	
ROBOT	TYPE
*R1	R2AWS1
R2	NO-EXIT
R3	NO-EXIT

JOINT HOME



- 编辑或获取HMOV 数据时按下"F2"键选择"HOME".

4.4 位置变量编辑设定

- ✓ 通过例子学习如何输入位置变量值..
- ✓ HMOV 数据编辑同上.
 - Ex) 修改各 “P10” 轴位置变量.

Step 1.

移动到位置变量编辑画面

< TEACH >	
ROBOT	TYPE
*R1	R2AWS1
R2	NO-EXIT
R3	NO-EXIT

JOINT HOME

F1

- 编辑或获取JOINT 数据时按下“F1” 键选择“JOINT”.

位置变量设定画面

<EDIT JNT>		V:	5	BL
Q4AFL		CURR		
1:0		2:0		
3:0		3:0		
6:0		6:0		

PGET MDI CORD EXIT

F2

- 获取、编辑、保存ROBOT 当前位置，利用保存的额位置执行 Robot 动作等时按下 “F2” 键选择 “MDI”.

位置变量设定画面

<EDIT JNT> V:		5	BL
Q4AFL	PO		US
1:0	2:0		
3:0	42:0		
5:0	6:0		

CORD FWRD EDIT

- 位置数据编辑设定画面

Step 2.

Point编号页面变更移动

<EDIT JNT> V:		5	BL
Q4AFL	PO		US
1:0	2:0		
3:0	3:0		
6:0	6:0		

CORD FWRD EDIT

Pg Up

Pg Dn

- 可以利用“PAGE UP/DOWN” 键进行Point移动

```

<EDIT JNT> V: 5 BL
Q4AFL P1 US
1:0 2:0
3:0 3:0
6:0 6:0

```

CORD FWRD EDIT

直接移动到Point编号

```

<EDIT JNT> V: 5 BL
Q4AFL P0 US
1:0 2:0
3:0 3:0
6:0 6:0

```

CORD FWRD EDIT



- 确认移动的Point编号

- 直接移动到输入的Point编号时按下"JUMP"键

```

<EDIT JNT> V: 5 BL
Q4AFL P0 US
1:0 2:0
3:0 3:0
6:0 6:0

```

Point # = 10



- 输入要移动的Point编号"10"

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	• W	— X

- 输入完成后按下"ENTER"键

```

<EDIT JNT> V: 5 BL
Q4AFL P10 US
1:0 2:0
3:0 3:0
6:0 6:0

```

CORD FWRD EDIT

Step 3.

轴 Cursor位置移动

```

<EDIT JNT> V: 5 BL
Q4AFL P0
1:0 2:0
3:0 4:0
5:0 6:0

```

CORD FWRD EDIT



- 按方向键移动到要编辑的轴
- 可以通过Cursor位置确认移动结果

```

<EDIT JNT> V:    5  BL
Q4AFL           P0
1:0             2:■
3:0             4:0
5:0             6:0

```

CORD FWRD EDIT

- 确认移动的Cursor

Step 4.

轴间光标移动

```

<EDIT JNT> V:    5  BL
Q4AFL           P0
1:■            2:0
3:0            4:0
5:0            6:0

```

CORD FWRD EDIT

ENTER

- 各轴间光标移动时按下 "ENTER" 键
- Cursor按顺序移动.

```

<EDIT JNT> V:    5  BL
Q4AFL           P0
1:0             2:■
3:0             4:0
5:0             6:0

```

CORD FWRD EDIT

- 确认完成移动的 Cursor

Step 5.

变更位置变量值

```

<EDIT JNT> V:    5  BL
Q4AFL           P10
1:0             2:-10.5■
3:0             4:0
5:0             6:0

```

CORD FWRD EDIT

ENTER

- 按下要变更的值数字键输入
- 按下"ENTER" 键反映到画面

7	8	9
G	H	I
4	5	6
L	M	N
1	2	3
Q	R	S
0	●	—
V	W	X

Step 7.

删除变量值

<EDIT JNT> V:		5	BL
Q4AFL	P10		
1:0	2:-10.5		
2:0	4:0		
5:0	6:0		
CORD		FWRD	EDIT



- 错误输入变量值时按下“DEL”键按顺序删除输入的值。

<EDIT JNT> V:		5	BL
Q4AFL	P10		
1:0	2:-10.		
3:0	4:0		
5:0	6:0		
CORD		FWRD	EDIT

Step 8.

保存及取消

<EDIT JNT> V:		5	BL
Q4AFL	P10		
1:0	2:-10.5		
3:0	3:0		
5:0	6:0		
CORD		FWRD	EDIT



- 变更值后按下“ESC”键，update信息输出到画面

<EDIT JNT> V:		5	BL
Q4AFL	P10		
1:0	2:-10.5		
3:0	4:0		
5:0	6:0		
Update ? (ENT/ESC)			



- 保存按下“ENTER”
- 取消按下“ESC”



<EDIT JNT> V:		5	BL
Q4AFL	P10		
1:0	2:-10.5		
3:0	4:0		
5:0	6:0		
save complete			



- 保存完成后恢复到上一个画面按下“ENTER”键

<EDIT JNT> V:		5	BL
Q4AFL	P10		
1:0	2:-10.5		
3:0	4:0		
5:0	6:0		
CORD		FWRD	EDIT

- 恢复画面

4.5 ROBOT 驱动方法

- ✓ 可以使用速度变更、电机Break解除、Inch模式.

4.5.1 速度变更

- 变更ROBOT 动作速度.
- 切换功能

■ . 上/中/下 速度变更

Step 1. 上(BH)→ 中(BM) → 下(BL) 变更

〈EDIT JNT〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 变更速度时按下“VEL”键
- 速度值“下” → BL

〈EDIT JNT〉 V: 50 BM
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 变更速度时按下“VEL”键
- 速度值“中” → BM

〈EDIT JNT〉 V: 100 BH
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 变更速度时按下“VEL”键
- 速度值“上” → BH

■ . 10 %” 为单位变更速度

Step 1. “10 %”为单位变更速度

〈EDIT JNT〉 V: 50 BH
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 速度以 10%为单位变更时 “BH” 状态下按下 “▲,▼” 键
- 速度增加 10%时“▲”
- 速度减少 10%时 “▼”

```

<EDIT JNT> V: 50 BH
Q4AFL      CURR
1:0        2:0
3:0        4:0
5:0        6:0

```

PGET MDI CORD EXIT



- 显示增加的速度.

```

<EDIT JNT> V: 40 BH
Q4AFL      CURR
1:0        2:0
3:0        4:0
5:0        6:0

```

PGET MDI CORD EXIT



- 显示减少的速度.

■ . 5 %” 为单位变更速度

Step 1.

“5 %”为单位变更速度

```

<EDIT JNT> V: 50 BM
Q4AFL      CURR
1:0        2:0
3:0        4:0
5:0        6:0

```

PGET MDI CORD EXIT



- 速度以 5%为单位变更时 “BM” 状态
下按下 “▲,▼” 键
- 速度增加 5%时 “▲”
- 速度减少 5%时 “▼”

```

<EDIT JNT> V: 55 BM
Q4AFL      CURR
1:0        2:0
3:0        4:0
5:0        6:0

```

PGET MDI CORD EXIT



- 显示增加的速度.

```

<EDIT JNT> V: 45 BM
Q4AFL      CURR
1:0        2:0
2:0        4:0
5:0        6:0

```

PGET MDI CORD EXIT



- 显示减少的速度.

■ “.1 %” 为单位变更速度

Step 1.

“1 %”为单位变更速度

<EDIT JNT> V: 50 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 速度以 1%为单位变更时 “BL” 状态
下按下 “▲,▼” 键
- 速度增加1%时“▲”
- 速度减少 1%时 “▼”

<EDIT JNT> V: 51 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 50 6:0

PGET MDI CORD EXIT



- 显示增加的速度.

<EDIT JNT> V: 49 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 显示减少的速度.

4.5.2 使用INCH 模式

- 按一次JOG键可以移动用户设定的移动距离.
- 设定值请参考 ‘MOTION’ 参数
- ‘Ia’ 设定值为 1mm 时移动距离计算方法
 Bl → 以(1mmx.0.1)的移动距离移动.
 Bm → 以(1mmx.0.5) 的移动距离移动.
 Bh → 以(1mmx.1) 的移动距离移动.

Step 1.

使用设定

〈EDIT JNT〉 V: 5 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 使用移动距离时按下“SPACE”键
- 取消方法也相同.

〈EDIT JNT〉 V: 5 Bl
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT

- 显示INCH 模式

Step 2.

上(Bh)→ 中(Bm) → 下(Bl)变更

〈EDIT JNT〉 V: 5 Bl
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 要变更移动距离时按下“VEL”键
- 移动距离“下”→ Bl

〈EDIT JNT〉 V: 50 Bm
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 要变更移动距离时按下“VEL”键
- 移动距离“中”→ Bm

<EDIT JNT> V: 100 Bh
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 要变更移动距离时按下“VEL”键
- 移动距离“上”→ Bm

4.5.3 BRAKE 解除

- 可以解除Brake电机的Brake.
- 器具部移动到安全位置后解除Brake.
- Brake解除只有在“**CURR**”画面中进行.
- 通过Z轴Brake解除方法学习Brake解除方法.

Step 1.

Brake解除

〈EDIT JNT〉 V: 5 BL
Q4AFL **CURR**
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 解除Brake时“CURR”画面 中 按 下 “BRK”键

〈EDIT JNT〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

Brake manually?

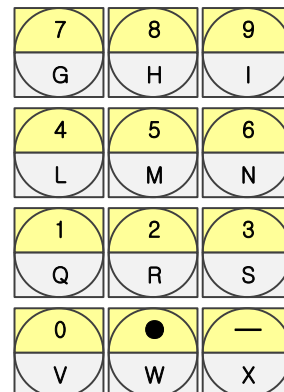


- 解除Brake时按下“ENTER” 键

〈EDIT JNT〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

Password =

- 输入PASSWORD “1111”



〈EDIT JNT〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

Password =1111

〈BRAKE STATE〉 0/0
Q4AFL CURR
1:ON 2: ON
3: ON 4: ON
5: ON 6: ON

Step 2.

Cursor位置移动

〈BRAKE STATE〉		0/0
Q4AFL	CURR	
1: ON ■	2: ON	
3: ON	4: ON	
5: ON	6: ON	

〈BRAKE STATE〉		0/0
Q4AFL	CURR	
1: ON	2: ON ■	
3: ON	4: ON	
5: ON	6: ON	



- 按下方向键移动到要解除的轴
- 可以通过Cursor的位置确认移动结果

Step 3.

选择Brake轴

〈BRAKE STATE〉		0/0
Q4AFL	CURR	
1: ON	2: ON ■	
3: ON	4: ON	
5: ON	6: ON	

〈BRAKE STATE〉		0/0
Q4AFL	CURR	
1: ON	2: ON ■	
3: ON	4: ON	
5: ON	6: ON	
Z OFF? (ENT/ESC)		



- 解除Z轴电机Brake时按下“ENTER”键

Brake解除/取消

〈BRAKE STATE〉		0/0
Q4AFL	CURR	
1: ON	2: ON ■	
3: ON	4: ON	
5: ON	6: ON	
Z OFF? (ENT/ESC)		

〈BRAKE STATE〉		0/0
Q4AFL	CURR	
1: ON	2: OFF ■	
3: ON	4: ON	
5: ON	6: ON	



- Brake解除时按下“ENTER”键
- 取消时按下“ESC”键

4.5.4 电机电源 ON/OFF

- 可以ON/OFF servo电机电源.
- Servo电源接入只能在“2. POINT” → “CURR”画面中进行.
- 切换功能 : ON → OFF → ON

Step 1.

电机电源 ON

〈EDIT JNT〉 V: 5 BL
Q4AFL **CURR**
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 电机电源“ON”时按下“SVON”
- Teach Pendant的Servo电源LED亮灯



Step 2.

电机电源 OFF

〈EDIT JNT〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 电机电源“OFF”时按下“SVON”
- Teach Pendant的Servo电源LED亮灯



4.5.5 轴Monitor变更

- 使用6轴以上的轴时无法在Teach Pendant显示所有轴. 通过轴变更可以显示6轴以上的轴.
- 切换方式 : [1~6]轴, → [7~12]轴 → [13~15]轴

Step 1.

显示1~6轴

```

<EDIT JNT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
  
```

PGET MDI CORD EXIT



- 变更轴时按下"AXIS CHG"键

显示7~12轴

```

<EDIT JNT> V:    5  BL
Q4AFL          CURR
7:0            8:0
9:0           10:0
11:0           12:0
  
```

PGET MDI CORD EXIT



- 变更轴时按下"AXIS CHG"键

显示13~16轴=

```

<EDIT JNT> V:    5  BL
Q4AFL          CURR
13:0           14:0
15:0           16:0
  
```

PGET MDI CORD EXIT



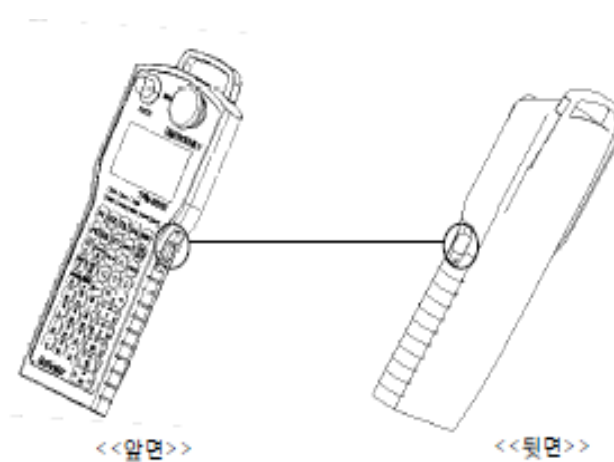
- 变更轴时按下"AXIS CHG"键

4.5.6 手动模式下位置Teaching

- 可以手动操作Robot.

4.5.6.1 Pendant 基本操作及Deadman开关说明

- Teach Pendant双手操作，Deadman开关利用适当的力量操作



- Deadman开关功能在手动运行 Robot的过程中预防无法预测的事故非常重要. 操作者在Robot运行中遇到无法预测的事故时，按下Deadman开关的力量过弱或过强.
- Deadman开关包含三种位置.
 - i. 不按下开关或轻按→ Robot 停止
 - ii. 正常力量按下开关→ Robot 动作
 - iii. 强力按下开关→ Robot 停止
- 不按下Deadman开关操作时显示如下信息

```

<EDIT JNT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
  
```

Please push deadman!

4.5.6.2 Robot手动移动

Step 1.

确认安全.

- 确认Robot 安装状态是否正常
- 确认出操作者 Robot 动作范围内是否有人

Step 2.

开启Robot 控制器电源.

- 顺时针方向旋转控制器电源开启电源.
- 控制器启动后Segment的 LED开启, 电源指示灯亮起.
- 控制器启动时间大约需要30秒.

Step 3.

Teach Pendant模式开关变为手动

- Mode Switch变更为 “Manual”模式.
- 操作方法请参考 [1.1 T/P 结构](#).

Step 4.

移动到CURR 画面

- 只有在CURR 模式移动 Robot.
- 移动到CURR 画面的方法请参考以下内容.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

EDIT选择

ENTER

- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

<TEACH MENU>
 1. EDIT 2. RUN
3. TEACH 4. I/O
 5. ORIGIN 6. INFO

 0. SHUTDOWN
 input : #



- “3. TEACH” 选择

Robot 选择

<TEACH>
 ROBOT TYPE
***R1** R2AWS1
 R2 NO-EXIT
 R3 NO-EXIT

 JOINT HOME



- 按下“UP/DOWN” Cursor键选择要选择的Robot



- 可以确认选择的Robot一行最前面的“*” 符号

JOINT 数据 选择

<TEACH>
 ROBOT TYPE
***R1** R2AWS1
 R2 NO-EXIT
 R3 NO-EXIT

JOINT HOME



- 编辑或获取JOINT 数据时按下 “F1” 键

CURR 画面

<EDIT JNT> V: 5 BL
 Q4AFL **CURR**
 1:0 2:0
 3:0 4:0
 5:0 6:0

 PGET MDI CORD EXIT

- ROBOT 当前位置值画面确认
- 可以驱动Robot的画面

Step 5.

开启电机电源.

- 开启Servo 电机电源.
- 按下Teach Pendant “SVON” 键



- Teach Pendant的Servo电源 LED亮灯



Step 6

速度变更.

- 移动Robot时速度快有可能发生危险，请降低速度使用。

上(BH)→ 中(BM) → 下(BL) 变更

〈EDIT JNT〉 V: 5 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 变更速度时按下“VEL”键
- 速度值 “下” → BL

〈EDIT JNT〉 V: 50 BM
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 变更速度时按下“VEL”键
- 速度值 “中” → BM

〈EDIT JNT〉 V: 100 BH
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 变更速度时按下“VEL”键
- 速度值 “上” → BH

“10 %”为单位变更速度

〈EDIT JNT〉 V: 50 BH
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 速度以10%为单位变更时，“BH” 状态下按下“▲,▼” 键
- 速度增加 10%时“▲”
- 速度减少 10%时 “▼”

“5 %”为单位变更速度

〈EDIT JNT〉 V: 50 BM
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 速度以 5%为单位变更时，“BM” 状态下按下 “▲,▼” 键
- 速度增加 5%时“▲”
- 速度减少 5%时 “▼”

“1 %”为单位变更速度

<EDIT JNT> V: 50 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- 速度以 1%为单位变更时，“BL” 状态下按下 “▲,▼” 键
- 速度增加 1%时“▲”
- 速度减少 1%时 “▼”

Step 7.

实际运行Robot各轴.

- 禁止Robot 动作范围内人员出入.
- 只能在CURR 画面进行 Robot动作.
- 按下Keypad的 “移动方向”键.
- 变更轴时按下“AXIS CHG”键

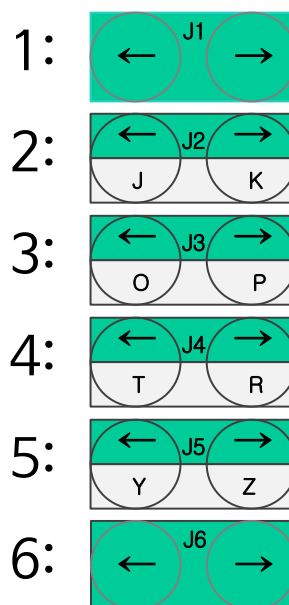


1~6轴 CURR 画面

<EDIT JNT> V: 50 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT

- 移动时按下“移动方向” 键



7~12轴 CURR 画面

```

<EDIT JNT> V: 5 BL
Q4AFL CURR
7:0      8:0
9:0      10:0
11:0     12:0

```

PGET MDI CORD EXIT

13~16轴 CURR 画面

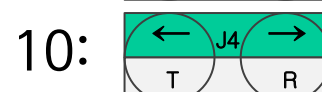
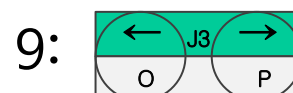
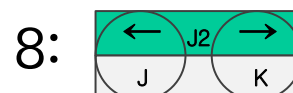
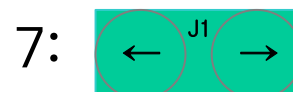
```

<EDIT JNT> V: 5 BL
Q4AFL CURR
13:0    14:0
15:0    16:0

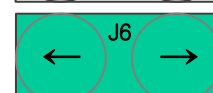
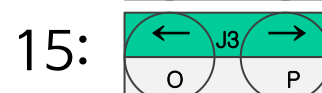
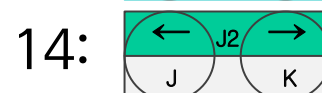
```

PGET MDI CORD EXIT

- 移动时按下“移动方向”键



- 移动时按下“移动方向”键



4.6 ROBOT JOINT

✓ JOINT数据位置变量编辑及获取当前位置方法如下详细项.

详细项目

GROUP		内容	备注
Page 1	[F1] PGET	JOINT 数据获取 ROBOT当前位置	(获取当前位置)
	[F2] MDI	编辑JOINT 数据的位置变量	
	[F3] CORD	变更当前位置数据坐标系(JOINT → 直角)	
	[F4] AFLD	仅将设定的Hand轴移动到设定的基准位置.	
Page 2	[F1]JSYN	设置为同步模式.	SCARA only
	[F2] ZCAL	设置为Z轴补偿模式.	

OPEN顺序

- Top Screen -> 3. JOG -> ROBOT SELECT -> [F1] JOINT

Step 1.

选择JOINT数据

<TEACH>
ROBOT
*R1
R2
R3

TYPE
R2AWS1
NO-EXIT
NO-EXIT

JOINT HOME



- 选择JOINT数据时按下"F1" 键 选择 "JOINT".

JOINT 数据画面

<EDIT JNT> V: 5 BL
Q4AFL
1:0
3:0
5:0

CURR
2:0
4:0
6:0

PGET MDI CORD EXIT

- JOINT数据 Current 画面确认

选择Position Get

<EDIT JNT> V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT

选择Position Get



- JOINT数据获取当前位置时按下"F1"键选择"PGET".

<EDIT JNT> V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT

选择Position Get



- 编辑JOINT数据时按下"F2" 键选择"MDI".

<EDIT JNT> V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT

选择Position Get



- 变更选择的 Robot坐标系时按下 "F3" 键选择"CORD".

<EDIT JNT> V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 退出JOINT数据.

4.6.1 CURR (Current)

- 可以确认Robot 当前位置

Step 1.

CURR 画面移动

(EDIT JNT) V: 5 BL
Q4AFL **CURR**
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT

- 确认ROBOT 当前位置值画面

4.6.2 PGET (Position Get)

- 获取Robot 当前位置到 JOINT 数据.
- 通过给Point编号 “5” 获取当前位置的练习学习 “Position Get”

Step 1.

PGET 画面移动

〈EDIT JNT〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- 获取当前位置时按下 “F1” 键

- 选择赋予当前位置的Point编号.
- 恢复到上一个画面按下 “ESC”
- 输入编号利用 “数字键”

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

〈EDIT JNT〉
Select point number
Range (0 ~39999)
Point num: ■

〈EDIT JNT〉
Select point number
Range (0 ~39999)
Point num: 5

Update? (ENT/ESC)



- 保存到内存按下“ENTER”
- 取消并恢复到上一个画面按下“ESC”

〈EDIT JNT〉
Select point number
Range (0 ~39999)
Point num: 5

Update complete

Step 2.

画面恢复

<EDIT JNT>
Select point number
Range (0 ~ 39999)
Point num: ■

Update complete

<EDIT JNT> V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- Currnet 画面恢复

4.6.3 MDI (Manual data input)

- 可以进行Robot位置的编辑、复制、移动、删除、以保存的位置执行Robot动作等，详细项如下表。
- 有数据的状态下限时 “US”
- 无数据的状态下限时 “NW”

(EDIT JNT) V: 5 BL
 Q4AFL P0 **NW**
 1:- 2:-
 3:- 4:-
 5:- 6:-
 PGET MDI CORD EXIT

(EDIT JNT) V: 5 BL
 Q4AFL P0 **US**
 1:0 2:0
 3:0 4:0
 5:0 6:0
 PGET MDI CORD EXIT

详细项

GROUP	内容	备注
[F1] CORD	变更位置数据坐标系(JOINT → 直角).	
[F2] FWRD	以保存的位置执行Robot 动作.	
[F4] EDIT	位置数据复制及删除	

OPEN顺序

- Top Screen -> 3. JOG -> ROBOT SELECT -> [F1] JOINT -> [F2] MDI

Step 1.

ROBOT位置变量设定画面

(EDIT JNT) V: 5 BL
 Q4AFL P0 US
 1:0 2:0
 3:0 4:0
 5:0 6:0
 PGET **MDI** CORD EXIT



- 位置数据编辑、保存、以保存的位置执行Robot 动作等时按下“F2” 键选择“MDI”。

ROBOT位置数据编辑画面

(EDIT JNT) V: 5 BL
 Q4AFL **P0** US
 1:0 2:0
 3:0 4:0
 5:0 6:0
 CORD FWRD EDIT

- 位置数据编辑设定画面

选择坐标系变更画面

```

<EDIT JNT> V: 5 BL
Q4AFL        P0  US
1:0          2:0
3:0          4:0
5:0          6:0

```

CORD FWRD EDIT



- 变更位置数据坐标系时按下"F1" 键选择"CORD".

以保存的位置执行Robot 动作

```

<EDIT JNT> V: 5 BL
Q4AFL        P0  US
1:0          2:0
3:0          4:0
5:0          6:0

```

CORD FWRD EDIT



- 以保存的位置执行Robot动作时按下"F2" 键选择 "FWRD".

复制及删除

```

<EDIT JNT> V: 5 BL
Q4AFL        P0  US
1:0          2:0
3:0          4:0
5:0          6:0

```

CORD FWRD EDIT



- 现有位置数据复制到另一个位置数据或删除时按下F4" 键选择"EDIT" 键.

直接移动到位置数据编号

```

<EDIT JNT> V: 5 BL
Q4AFL        P0  US
1:0          2:0
3:0          4:0
5:0          6:0

```

CORD FWRD EDIT



- 直接移动到要编辑的Point编号时按下"JUMP " 键

4.6.3.1 位置变量编辑

- 可以编辑位置变量

Step 1.

MDI 画面移动

〈EDIT JNT〉 V:		5	BL
Q4AFL	PO		US
1:0	2:0		
3:0	4:0		
5:0	6:0		
CORD FWRD		EDIT	

- 编辑ROBOT 位置变量

■ Point编号移动

- ① 请参考[3.4 位置变量编辑设定](#) 的 “Point编号移动及页面变更” 。

■ Cursor位置移动

- ① 请参考[3.4 位置变量编辑设定](#) 的 “Cursor位置移动及轴间移动Cursor” 。

■ 设定方法

- ① 请参考[3.4 位置变量编辑设定](#) 的 “位置变量值变更” 。

■ 编辑保存及取消方法

- ① 请参考[3.4 位置变量编辑设定](#) “保存及取消” 。

4.6.3.2 CORD (Coordinate)

- 可以变更位置变量值坐标系
- 坐标系以[JOINT → 直角]顺序显示.
- 切换功能.

Step 1.

COORD 画面移动

```

<EDIT JNT> V:    5  BL
Q4AFL          P0  US
1:0            2:0
3:0            4:0
5:0            6:0
  
```

CORD FWRD EDIT



- 变更坐标时按下“F1”键选择“CORD”.

```

<EDIT JNT> V:    5  BL
Q4AFL          P0  US
1:0            2:0
3:0            4:0
5:0            6:0
  
```

not work!!!

- 无法使用的Robot 种类键操作时显示
"not work!!!"

■ 参考事项

- ① 只能在“SCARA” Robot 种类执行，其余Robot无法使用.
- ② 输出“not work!!!”信息.

4.6.3.3 FWRD (Forward)

- 以保存的位置执行Robot 动作.

■ 详细项

GROUP	内容	备注
[F1] JMOV	以保存的位置变量值所有轴进行 JOINT 动作	
[F2] JSE	以保存的位置变量值指定轴进行动作.	
[F3] LMO	以保存的位置变量值所有轴进行直角动作.	

■ ROBOT 轴编号

ROBOT AXIS	值	内容
T	0	电机 1轴
Z	1	电机 2轴
R1	2	电机 3轴
R1	3	电机 4轴
X	4	电机 5轴
Y	5	电机 6轴
R3	6	电机 7轴
R4	7	电机 8轴
Q1	8	电机 9轴
V1	9	电机 10轴
Q2	10	电机 11轴
V2	11	电机 12轴
Q3	12	电机 13轴
V3	13	电机 14轴
Q4	14	电机 15轴
V4	15	电机 16轴

■ OPEN顺序

- Top Screen → 3. JOG → 2. ROBOT SELECT → [F1] JOINT → [F2] MDI → [F2] FWRD

Step 1.

ROBOT位置变量设定画面

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0              2:0
3:0              4:0
5:0              6:0
  
```

PGET **MDI** CORD EXIT

F2

- 获取ROBOT 当前位置、编辑、保存、以保存的位置执行Robot 动作等时按下"F2" 键选择“MDI”.

ROBOT位置数据编辑画面

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0              2:0
3:0              4:0
5:0              6:0
  
```

CORD **FWRD** JUMP EDIT

F2

- 位置数据编辑设定画面
- 以保存的位置执行Robot 动作时按下"F2" 键选择“FWRD”.

移动到位置变量值

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0              2:0
3:0              4:0
5:0              6:0
  
```

JSEL JMOV LMOV

- 以位置变量值执行Robot 动作

Step 1.

所有轴 JOINT 动作.

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0              2:0
3:0              4:0
5:0              6:0
  
```

JMOV JSEL LMOV

F1

- 所有轴进行JOINT 动作时按下"F1" 键选择“JMOV”.

指定的轴动作

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0              2:0
3:0              4:0
5:0              6:0
  
```

JMOV **JSEL** LMOV

F2

- 指定的轴执行JOINT 动作时按下 "F2" 键选择“JSEL”.

所有轴直角动作

〈EDIT JNT〉 V: 5 BL
Q4AFL P0 US
1:0 2:0
3:0 4:0
5:0 6:0

JMOV JSEL **LMOV**



- 所有轴执行直角动作时按下"F3"键选择“LMOV”。

4.6.3.3.1 JMOV

- 所有轴执行 JOINT 动作.

■ 动作方法

Step 1.

投入Servo电源

```

<EDIT JNT> V:    5  BL
Q4AFL          P0  US
1:0            2:0
3:0            4:0
5:0            6:0
  
```

JMOV JSEL LMOV



- 执行前电机接入电源.

Step 2.

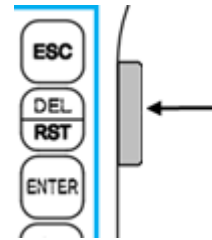
Deadman switch

```

<EDIT JNT> V:    5  BL
Q4AFL          P0  US
1:0            2:0
3:0            4:0
5:0            6:0
  
```

JMOV JSEL LMOV

- 执行前利用适当的力量抓住Deadman switch..



Step 3.

JOINT 动作选择画面

```

<EDIT JNT> V:    5  BL
Q4AFL          P0  US
1:0            2:0
3:0            4:0
5:0            6:0
  
```

JMOV JSEL LMOV

- 执行所有轴JOINT 动作时按下"F1" 键选择“JMOV”.



- 开始按下 “ENTER”
- 取消按下 “ESC”



```

<EDIT JNT> V:    5  BL
Q4AFL          P0  US
1:0            2:0
3:0            4:0
5:0            6:0
  
```

Ready ? (ENT/ESC)

- Robot 动作中停止时按下“ESC”键

```

<EDIT JNT> V: 5 BL
Q4AFL P0 US
1:0 2:0
3:0 4:0
5:0 6:0

```

Prease ESC to stop

```

<EDIT JNT> V: 5 BL
Q4AFL P0 US
1:0 2:0
3:0 4:0
5:0 6:0

```

forward complete

```

<EDIT JNT> V: 5 BL
Q4AFL P0 US
1:0 2:0
3:0 4:0
5:0 6:0

```

JMOV JSEL LMOV



- 动作完成后恢复画面时按下“ENTER”键

- 恢复画面

4.6.3.3.2 JSEL

- 利用保存的位置变量值仅对指定的一个轴进行动作。

■ 动作方法

Step 1.

接入Servo电源

```
<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0
```

JMOV JSEL LMOV



- 执行前电机接入电源。

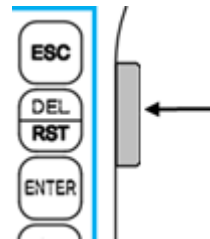
Step 2.

Deadman switch

```
<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0
```

JMOV JSEL LMOV

- 执行前利用适当的力量抓住 Deadman switch..



Step 3.

指定的轴动作选择画面

```
<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0
```

JMOV JSEL LMOV



- 指定轴进行JOINT 动作时按下“F2” 键选择“JSEL”

```
<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0
```

Forward axis #=1



- 选择要动作的轴。
- 输入电机2轴编号“1”
- 输入完成后按下“ENTER”键

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

Ready ? (ENT/ESC)

ENTER

- 开始按下 “ENTER”
- 取消按下 “ESC”

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

Prease ESC to stop

- Robot 动作中停止时按下“ESC”键

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

forward complete

ENTER

- 动作完成后恢复画面时按下 “ENTER” 键

```

<EDIT JNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

JSEL JMOV LMOV

- 恢复画面

■ 确认轴编号

- ① 请参考[3.6.3. MDI](#) “ROBOT 轴编号”。

4.6.3.3.3 LMOV

- 所有轴进行直角动作.

■ 动作方法

Step 1.

接入Servo电源

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

JSEL  JMOV  LMOV

```



- 执行前电机接入电源.

Step 2.

Deadman switch

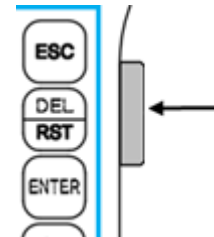
```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

JSEL  JMOV  LMOV

```

- 执行前利用适当力量抓住Deadman switch..



Step 3.

直角动作选择画面

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

JSEL  JMOV  LMOV

```



- 所有轴执行直角动作时按下 "F3" 键选择“LMOV”.

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

Ready ? (ENT/ESC)



- 开始按下 “ENTER”
- 取消按下 “ESC”
- Robot动作中停止时按下“ESC”键


```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

Prease ESC to stop

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

forward complete

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

JSEL JMOV LMOV

ENTER

- 动作完成后恢复画面时按下“ENTER”键

ENTER

- 恢复画面

Step 4.

选择无法使用的Robot时

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

JSEL JMOV **LMOV**

```

<EDIT PNT> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

only use for SCARA!

```

<EDIT MDI> V:      5  BL
Q4AFL             P0  US
1:0               2:0
3:0               4:0
5:0               6:0

```

JSEL JMOV LMOV

F3

- 无法使用的Robot 种类的键操作时显示"only use for SCARA!"
- 恢复画面时按下“ENTER”键

- 恢复画面

■ 参考事项

- ① 只能在“SCARA” Robot 种类执行，其余Robot无法使用.
- ② 输出"only use for SCARA!" 信息.

Robostar

4.6.3.4 EDIT

- 可以复制现有位置数据到另一个位置数据或删除.
- 有数据的状态下显示 “US”
- 无数据的状态下显示 “-- ”

■ 详细项

GROUP	内容	备注
[F1] COPY	现有位置数据复制到另一个位置数据	
[F2] DEL	删除位置数据	
[F3] JUMP	移动到输入的Point编号	

■ OPEN顺序

- Top Screen → 3. JOG → ROBOT SELECT → [F1] JOINT → [F2] MDI → [F4] EDIT

Step 1.

ROBOT位置变量设定画面

<EDIT JNT> V: 5 BL
 Q4AFL CURR
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET MDI CORD EXIT



- ROBOT位置编辑、保存、以保存的位置执行Robot 动作时按下“F2”键

Step 2.

位置数据修正

<EDIT MDI> V: 5 BL
 Q4AFL P0 US
 1:0 2:0
 3:0 4:0
 5:0 6:0

CORD FWRD EDIT



- 位置数据修正画面

复制及删除画面

<EDIT EDIT>
Q4AFL1
P0 US P1 US
P2 -- P3 --
P4 -- P5 --
P6 -- P7 --
P8 -- P9 --
COPY DEL

- 复制及删除画面

<EDIT EDIT>
Q4AFL1
P0 US P1 US
P2 -- P3 --
P4 -- P5 --
P6 -- P7 --
P8 -- P9 --
COPY DEL



- 现有位置数据复制到另一个位置数据时
按下"F1" 键

<EDIT EDIT>
Q4AFL1
P0 US P1 US
P2 -- P3 --
P4 -- P5 --
P6 -- P7 --
P8 -- P9 --
COPY **DEL**



- 删除位置数据时按下"F2" 键

4.6.3.4.1 COPY

- 可以将现有位置数据复制到另一个位置数据。

复制方法

Step 1.

Point编号页面移动

```

<EDIT EDIT>
Q4AFL1
■P0  US      P1  US
P2  --      P3  --
P4  --      P5  --
P6  --      P7  --
P8  --      P9  --
COPY  DEL
  
```



- 利用"PAGE UP/DOWN" 键移动Point

```

<EDIT EDIT>
Q4AFL1
■P10 US      P11 US
P12 US      P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
COPY  DEL
  
```



- 通过Cursor确认移动到的Point编号

Step 2.

Cursor位置移动

```

<EDIT EDIT>
Q4AFL1
■P10 US      P11 US
P12 US      P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
COPY  DEL
  
```



- 按下方向键移动到要选择的Point编号.
- 通过Cursor的位置确认移动结果

```

<EDIT EDIT>
Q4AFL1
P10 US      P11 US
■P12 US      P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
COPY  DEL
  
```

Step 3.

选择要复制的Point编号

```

<EDIT EDIT>
Q4AFL1
P10 US      P11 US
P12 US      P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
COPY DEL

```



- 选择要复制的Point编号时按下“ENTER”键

```

<EDIT EDIT>
Q4AFL1
P10 US      P11 US
*P12 US    P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
COPY DEL

```

- 选择后显示“*”符号

```

<EDIT EDIT>
Q4AFL1
P10US      P11 US
*P12 US    P13 --
*P14 US    P15 --
*P16 US    P17 --
P18 --      P19--
COPY DEL

```

- 选择多行时每行显示“*”符号.

Step 4.

Point复制

```

<EDIT EDIT>
Q4AFL1
P10 US      P11US
*P12 US    P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
COPY DEL

```



- 复制时按下“F1”键

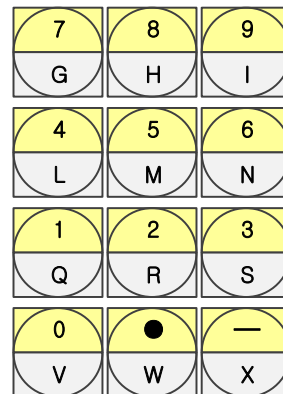
输入要复制的编号

```

<EDIT EDIT>
Q4AFL1
P10 US      P11 US
*P12 US    P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
point #=14

```

- 复制的 Point编号 “14”



```
<<EDIT EDIT>
Q4AFL1
P10 US      P11 US
* P12 US    P13 --
P14 --      P15 --
P16 --      P17 --
P18 --      P19 --
copy complete■
```

```
<EDIT EDIT>
Q4AFL1
P10 US      P11 US
P12 US      P13 --
■ P14 US    P15 --
P16 --      P17 --
P18 --      P19 --
COPY DEL
```

- 显示复制完成信息.

- 复制的 Point编号上显示“US”.

4.6.3.4.2 DEL

- 可以删除现有位置数据.
- 删除后变为无数据状态, 显示为 "--" .

■ 删除方法

Step 1.

Point编号页面移动

<EDIT EDIT>			
Q4AFL1			
■P0	US	P1	US
P2	--	P3	--
P4	--	P5	--
P6	--	P7	--
P8	--	P9	--
COPY DEL			

<EDIT EDIT>			
Q4AFL1			
■P10	US	P11	US
P12	US	P13	--
P14	--	P15	--
P16	--	P17	--
P18	--	P19	--
COPY DEL			



- 可以利用"PAGE UP/DOWN" 键移动 Point



- 通过Cursor确认移动的Point编号

Step 2.

Cursor位置移动

<EDIT EDIT>			
Q4AFL1			
■P10	US	P11	US
P12	US	P13	--
P14	--	P15	--
P16	--	P17	--
P18	--	P19	--
COPY DEL			

<EDIT EDIT>			
Q4AFL1			
P10	US	P11	US
■P12	US	P13	--
P14	--	P15	--
P16	--	P17	--
P18	--	P19	--
COPY DEL			



- 按下方向键移动到要选择的Point编号.
- 通过Cursor位置确认移动的结果

Step 3.

选择Point编号


```

<EDIT EDIT>
Q4AFL1
P10  US      P11  US
P12  US      P13  --
P14  --      P15  --
P16  --      P17  --
P18  --      P19  --
COPY  DEL

```



- 确认Point编号后选择时按下“ENTER”键

位置数据移动

```

<EDIT EDIT>
Q4AFL1
P10  US      P11  US
*P12 US      P13  --
P14  --      P15  --
P16  --      P17  --
P18  --      P19  --
COPY  DEL

```

- 选择后显示“*”符号

```

<EDIT EDIT>
Q4AFL1
P10  US      P11  US
*P12 US      P13  --
*P14 US      P15  --
*P16 US      P17  --
P18  --      P19  --
COPY  DEL

```

- 选择多行时每行显示“*”符号。

Step 4.

选择删除

```

<EDIT EDIT>
Q4AFL1
P10  US      P11  US
*P12 US      P13  --
P14  --      P15  --
P16  --      P17  --
P18  --      P19  --
COPY  DEL

```



- 删除时按下“F2”键

```

<EDIT EDIT>
Q4AFL1
P10  US      P11  US
*P12 US      P13  --
P14  --      P15  --
P16  --      P17  --
P18  --      P19  --
delete? (ENT/ESC)

```

- 删除按下“ENTER”
- 取消按下“ESC”

```

<EDIT EDIT>
Q4AFL1
P10  US      P11  US
*P12 US      P13  --
P14  --      P15  --
P16  --      P17  --
P18  --      P19  --
delete complete

```

- 显示复制完成信息
- 恢复画面时按下“ENTER”键

<EDIT EDIT>
Q4AFL1
P10 US P11 US
~~* P12 --~~ P13 --
P14 -- P15 --
P16 -- P17 --
P18 -- P19 --
COPY DEL

- 删除的Point编号显示“--”。

4.6.3.4.3 JUMP

- 可以直接移动到输入的Point编号.

Step 1

移动到输入的Point编号

<EDIT EDIT>

Q4AFL1

■P0U SPUS

P2 --

P3 --

P4 --

P5 --

P6 --

P7 --

P8 --

P9 --

COPY DEL



- 移动到输入的Point编号时按下 "JUMP" 键

<EDIT EDIT>

Q4AFL1

■P0 US

P1 US

P2 --

P3 --

P4 --

P5 --

P6 --

P7 --

P8 --

P9 --

Point # =10

- 输入要移动的Point编号 "10"

<EDIT EDIT>

Q4AFL1

■P10 US

P11 US

P12 --

P13 --

P14 --

P15 --

P16 --

P17 --

P18 --

P19 --

COPY DEL

- 确认移动完成的Point编号

4.6.4 CORD (Coordinate)

- 可以变更或监控位置变量值坐标系.
- 坐标系以 [JOINT → 直角] 的顺序显示.
- 切换功能.

Step 1.

COORD 画面移动

〈EDIT PNT〉	V: 5 BL
Q4AFL	CURR
1:0	2:0
3:0	4:0
5:0	6:0

not work!!!



- 操作无法使用的Robot种类的键时显示 "not work!!!"

■ 参考事项

- ① 只能执行在 "SCARA" Robot 种类, 其余Robot无法使用.
- ② 输出 "not work!!!" 信息.

4.6.5 AFLD (Arm Folding)

- 仅将设定的Hand轴移动到设定的基准为止.
- 使用情况及基准位置可以在Robot参数中编辑.

Step 1.

AFLD 画面移动

```
<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
```

PGET MDI CORD **AFLD**>

F4

- 按下"F4" 键执行 AFLD 功能

Step 2.

确认执行情况

```
<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
```

Fold arms? (ENT/ESC)

ENTER

- 按下"ENTER" 键执行 AFLD 功能

ESC

- 按下 "ESC" 键恢复到之前画面

执行AFLD 功能

```
<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
```

Press ESC to stop

ESC

- 执行功能.
- 可以按下"ESC" 键取消.

输出完成信息

```
<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
```

forward complete

ENTER

- 确认完成信息后 , 按下 "ENTER" 键恢复到之前画面.

■ 参考事项

- ① 作为JMOV 动作进行.
- ② 功能使用失败情况如下.

Step 3.

失败 - 无法使用功能

```

<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

Func is disabled!

- 输出"Func is disabled!" 信息

失败 - 非Servo ON状态

```

<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

please servo on!

- 输出"please servo on!" 信息

失败 - Over range

```

<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

fwd fail - OVR[1]

- 与"fwd fail" 信息，告知超出 Range 的轴编号

失败 - 其它

```

<LP JOINT> V:    5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

foward fail

- 其它 forward 失败时,
输出"forward fail" 信息

4.6.6 JSYN (JMOV 同步移动)

- 只可以在SCARA Robot中使用.
- SCARA Robot在特定点附近进行直线补间Motion (LMOV)时, 由于急剧的速度变化发生Over Speed Alarm。为解决这种问题, 对于ROSEP 类型同步启动 P2P Motion(JMOV)提供经过特定点的直线Motion功能.

- 目前不支持.

4.6.7 ZCAL(Z Calibration)

- 可以激活Z轴补偿因重力导致的Hand下垂情况的功能。
- 只有在Hand轴在基准点时可以使用。
- 使用情况及补偿值可以在Robot参数中编辑。

Step 1.

ZCAL 画面移动

```
<LP JOINT> V:    5 BL
Q4AFL          CURR
1:0            2:-20.57
3:0            4:0
5:0            6:0
```

PGET MDI CORD AFLD>



- 按下"NEXT" 键移动到下一个菜单窗口

```
<LP JOINT> V:    5 BL
Q4AFL          CURR
1:0            2:-20.57
3:0            4:0
5:0            6:0
```

<JSYN **ZCAL**



- 按下"F2" 键开始ZCAL 功能

Step 2.

直线ZCAL 功能

```
<LP JOINT> V:    5 BL
Q4AFL          CURR
1:0            2:-20.57
3:0            4:0
5:0            6:0
```

<JSYN ZCAL



- 按下"NEXT" 键移动到下一个菜单窗口

ZCAL 画面确认

```
<LP JOINT> V:    5 BL
Q4AFL          CURR
1:0            2:-20.57
3:10.1         4:0
5:0            6:0
```

ZCAL (2 : -20.57)

<JSYN ZCAL

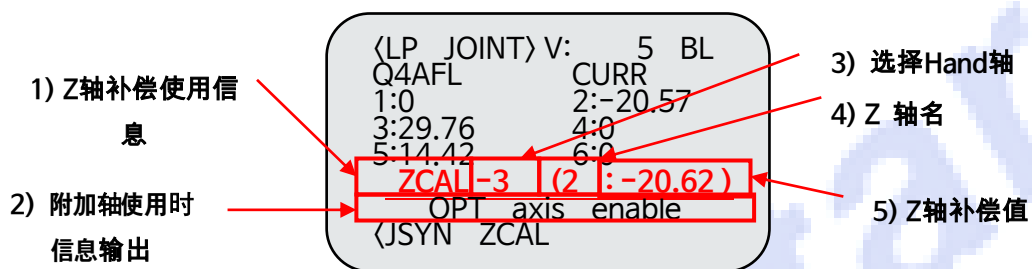
- 确认ZCAL执行画面。

■ 画面及功能说明

① 画面说明

- 1) Z轴补偿使用信息: Z轴补偿激活时输出的信息 ,

- 2) 附加轴使用信息: Z轴补偿中使用附加轴时输出的信息.
- 3) 选择Hand轴: 输出位于基准点的Hand信息. 基准点上存在多个Hand时, 将最先移动的Hand输出到画面.
- 4) Z轴名: Z轴名输出到画面.
- 5) Z轴补偿值: 根据Hand位置可以确认应用补偿式的值. 输出实际位置值.



② 功能说明

- 1) **[J1~J6]:** 可以完成与现有相同的JOG动作, 其中移动Hand轴时Z轴补偿值变更.与设置为附加轴的Hand轴同步移动, 不接收KEY的输入.
- 2) **[F2]:** 开始或结束Z轴补偿. 结束时设置为当前位置值.

Step 1.

ZCAL 结束

```

<LP JOINT> V: 5 BL
Q4AFL CURR
1:0 2:-20.57
3:29.76 4:0
5:14.42 6:0
ZCAL-3 (2 : -20.62)
OPT axis enable
zcal off? (ENT/ESC)
  
```



- 按下"F2" 按钮输出 Z轴补偿结束确认信息

```

<LP JOINT> V: 5 BL
Q4AFL CURR
1:0 2:-20.62
3:29.76 4:0
5:14.42 6:0
<JSYN ZCAL
  
```



- 按下"ENTER" 键结束Z轴补偿后, 变更为当前位置值

```

<LP JOINT> V:      5 BL
Q4AFL      CURR
1:0         2:-20.57
3:29.76     4:0
5:14.42     6:0
      ZCAL-3 (2 : -20.62)
      OPT axis enable
<JSYN ZCAL

```

ESC

- 按下"ESC" 键维持 Z轴补偿

■ 参考事项

- ① 没有设置附加轴时，不会输出 "OPT axis enable" 信息.
- ② 功能使用失败的情况如下.

Step 2.

失败 – Hand不是基准点时

```

<LP JOINT> V:      5 BL
Q4AFL      CURR
1:0         2:-20.57
3:10.1      4:100
5:0         6:0

```

Wrong start pos!

ENTER

ESC

- 按下"ENTER" 键执行AFLD 功能
- 按下 "ESC" 键恢复到之前画面

4.7 ROBOT HOME

✓ 编辑HMOV数据位置变量及当前位置获取方法的详细项如下表.

■ 详细项

GROUP		内容	备注
Page 1	[F1] PGET	HOME 数据中获取ROBOT当前位置	(获取当前位置)
	[F2] MDI	编辑HOME 数据的位置变量	
	[F3] CORD	变更当前位置数据坐标系(JOINT → 直角)	
	[F4] AFLD	仅将设定的Hand轴移动到设定的基准位置.	
Page 2	[F1]JSYN	设置为同步模式.	SCARA only
	[F2] ZCAL	设置为Z轴补偿模式.	

■ OPEN顺序

- Top Screen -> 3. JOG -> ROBOT SELECT -> [F2] HOME

Step 1.

选择JOINT数据

```

< TEACH >
ROBOT      TYPE
*R1        R2AWS1
R2         NO-EXIT
R3         NO-EXIT
  
```

JOINT **HOME**



- 选择JOINT数据时按下"F1" 键选择“HOME”.

HOME 数据画面

```

<EDIT HOME> V:  5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0
  
```

PGET MDI CORD EXIT

选择Position Get

- HOME 数据 Current 画面确认

```

<EDIT HOME> V:  5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

PGET MDI CORD EXIT

选择Position Get



- HOME数据获取当前位置时按下"F1"键

```

<EDIT HOME> V:  5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

PGET MDI CORD EXIT

Position Get 选择



- 编辑HOME数据时按下"F2" 键

```

<EDIT HOME> V:  5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

PGET MDI CORD EXIT

Position Get 选择



- 变更选择的 Robot坐标系时按下 "F3" 键

```

<EDIT HOME> V:  5  BL
Q4AFL          CURR
1:0            2:0
3:0            4:0
5:0            6:0

```

PGET MDI CORD EXIT



- 退出HOME数据.

4.7.1 CURR (Current)

- 可以确认Robot 当前位置

Step 1.

CURR 画面移动

<EDIT HOME> V: 5 BL
Q4AFL **CURR**
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT

- 确认ROBOT 当前位置值画面

4.7.2 PGET (Position Get)

- 获取Robot 当前位置到 HOME 数据.
- 通过给Point编号 “5” 获取当前位置的练习学习 “Position Get”

Step 1.

PGET 画面移动

〈EDIT HOME〉 V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT

F1

- 获取当前位置时按下 “F1” 键

〈EDIT HOME〉
Select home number
Range (0 ~3)
Home num: ■

〈EDIT HOME〉
Select home number
Range (0 ~3)
Home num: 5

Update? (ENT/ESC)

〈EDIT HOME〉
Select home number
Range (0 ~3)
Home num: 5

Update complete

- 输入赋予当前位置的Point编号.
- 恢复到上一个画面按下“ESC”
- 输入编号利用 “数字键”

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

ENTER

- 保存到内存按下“ENTER”
- 取消并恢复到上一个画面按下“ESC”

Step 2.

画面恢复

<EDIT HOME>
Select point number
Range (0 ~ 39999)
Point num: ■

Update complete

<EDIT HOME> V: 5 BL
Q4AFL CURR
1:0 2:0
3:0 4:0
5:0 6:0

PGET MDI CORD EXIT



- Currnet 画面恢复

4.7.3 MDI (Manual data input)

- 可以进行Robot位置的编辑、复制、移动、删除、以保存的位置执行Robot动作等，详细项如下表。

详细项

GROUP	内容	备注
[F1] CORD	变更位置数据坐标系 (JOINT → 直角).	
[F2] FWRD	以保存的位置执行Robot 动作.	

OPEN顺序

- Top Screen → 3. JOG → ROBOT SELECT → [F2] HOME → [F2] MDI

Step 1.

ROBOT位置变量设定画面

(EDIT HOME) V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET **MDI** CORD EXIT

F2

- 位置数据编辑、保存、以保存的位置执行Robot 动作等时按下"F2" 键选择“MDI”。

ROBOT位置数据编辑画面

(EDIT HOME) V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

CORD FWRD

- 位置数据编辑设定画面

选择坐标系变更画面

((EDIT HOME) V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

CORD FWRD EDIT

F1

- 变更位置数据坐标系时按下"F1" 键选择“CORD”。

以保存的位置执行Robot 动作

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

CORD **FWRD** EDIT



- 以保存的位置执行Robot动作时按下"F2"键选择"FWRD".

复制及删除

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

CORD FWRD **EDIT**



- 现有位置数据复制到另一个位置数据或删除时按下F4"键选择"EDIT"键.

直接移动到位置数据编号

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

CORD FWRD



- 直接移动到要编辑的Point编号时按下"JUMP"键

4.7.3.1 位置变量编辑

- 可以编辑位置变量

Step 1.

MDI 画面移动

<EDIT HOME> V: 5 BL		
Q4AFL	H0	
1:0	2:0	
3:0	4:0	
5:0	6:0	
CORD	FWRD	EDIT

- 编辑ROBOT 位置变量

■ Point编号移动

- ① 请参考[4.4 位置变量编辑设定](#) 的 “Point编号移动及页面变更” 。

■ Cursor位置移动

- ① 请参考[4.4 位置变量编辑设定](#) 的 “Cursor位置移动及轴键移动Cursor” 。

■ 设定方法

- ① 请参考[4.4 位置变量编辑设定](#) 的 “位置变量值变更” 。

■ 编辑保存及取消方法

- ① 请参考[4.4 位置变量编辑设定](#) “保存及取消” 。

4.7.3.2 CORD (Coordinate)

- 可以变更位置变量值坐标系
- 坐标系以[JOINT → 直角]顺序显示.
- 切换功能.

Step 1.

COORD 画面移动

(EDIT HOME) V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

CORD FWRD EDIT



- 变更坐标时按下“F1”键选择“CORD”.

(EDIT HOME) V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

not work!!!

- 无法使用的Robot 种类键操作时显示
"not work!!!"

■ 参考事项

- ① 只能执行在“SCARA” Robot 种类，其余Robot无法使用.
- ② 输出“not work!!!”信息.

4.7.3.3 FWRD (Forward)

- 以保存的位置执行Robot 动作.

■ 详细项

GROUP	内容	备注
[F1] JMOV	以保存的值为变量值所有轴进行 JOINT 动作	
[F2] JSE	以保存的值为变量值指定轴进行动作.	
[F3] LMO	以保存的值为变量值所有轴进行直角动作.	

■ ROBOT 轴编号

ROBOT AXIS	值	内容
T	0	电机 1轴
Z	1	电机 2轴
R1	2	电机 3轴
R1	3	电机 4轴
X	4	电机 5轴
Y	5	电机 6轴
R3	6	电机 7轴
R4	7	电机 8轴
Q1	8	电机 9轴
V1	9	电机 10轴
Q2	10	电机 11轴
V2	11	电机 12轴
Q3	12	电机 13轴
V3	13	电机 14轴
Q4	14	电机 15轴
V4	15	电机 16轴

■ OPEN顺序

- Top Screen → 3. JOG → 2. ROBOT SELECT → [F2] HOME → [F2] MDI → [F2] FWRD

Step 1.

ROBOT位置变量设定画面

〈EDIT HOME〉 V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

PGET **MDI** CORD EXIT

F2

- 获取ROBOT 当前位置、编辑、保存、以保存的位置执行Robot 动作等时按下"F2" 键选择“MDI”.

ROBOT位置数据编辑画面

〈EDIT HOME〉 V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

CORD **FWRD** JUMP EDIT

F2

- 位置数据编辑设定画面
- 以保存的位置执行Robot 动作时按下"F2" 键选择“FWRD”.

移动到位置变量值

〈EDIT HOME〉 V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

JSEL JMOV LMOV

- 以位置变量值执行Robot 动作

Step 1.

所有轴JOINT动作.

〈EDIT HOME〉 V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

JMOV JSEL LMOV

F1

- 所有轴进行JOINT 动作时按下"F1" 键选择“JMOV”.

指定的轴动作

〈EDIT HOME〉 V: 5 BL
 Q4AFL H0
 1:0 2:0
 3:0 4:0
 5:0 6:0

JMOV **JSEL** LMOV

F2

- 指定的轴执行JOINT 动作时按下 "F2" 键选择“JSEL”.

所有轴直角动作

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

JMOV JSEL **LMOV**



- 所有轴执行直角动作时按下"F3"键选择“LMOV”。

4.7.3.3.1 JMOV

- 所有轴执行 JOINT 动作.

■ 动作方法

Step 1.

接入Servo电源

<EDIT HOME> V: 5 BL

Q4AFL H0

1:0 2:0

3:0 4:0

5:0 6:0

JMOV JSEL LMOV



- 执行前电机接入电源.

Step 2.

Deadman switch

<EDIT HOME> V: 5 BL

Q4AFL H0

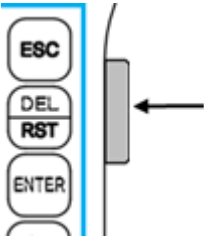
1:0 2:0

3:0 4:0

5:0 6:0

JMOV JSEL LMOV

- 执行前利用适当力量抓住Deadman switch..



Step 3.

JOINT 动作选择画面

<EDIT HOME> V: 5 BL

Q4AFL H0

1:0 2:0

3:0 4:0

5:0 6:0

JMOV JSEL LMOV



- 执行所有轴JOINT 动作时按下"F1" 键选择“JMOV”.

<EDIT HOME> V: 5 BL

Q4AFL H0

1:0 2:0

3:0 4:0

5:0 6:0

Ready ? (ENT/ESC)



- 开始按下 “ENTER”
- 取消按下 “ESC”



<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

Prease ESC to stop

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

forward complete

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

JMOV JSEL LMOV

- Robot动作中停止时按下“ESC”键



- 动作完成后恢复画面时按下“ENTER”键

- 恢复画面

4.7.3.3.2 JSEL

- 利用保存的为值变量值仅对指定的一个轴进行动作。

■ 动作方法

Step 1.

接入Servo电源

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JMOV JSEL LMOV



- 执行前电机接入电源.

Step 2.

Deadman switch

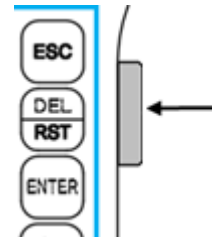
```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JMOV JSEL LMOV

- 执行前利用适当力量抓住Deadman switch..



Step 3.

指定的轴动作选择画面

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JMOV JSEL LMOV



- 指定轴进行JOINT 动作时按下"F2" 键选择"JSEL"

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

Forward axis #=1



- 选择要动作的轴.
- 输入电机2轴编号 "1"
- 输入完成后按下"ENTER"键

```

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

```

Ready ? (ENT/ESC)

ENTER

- 开始按下 “ENTER”
- 取消按下 “ESC”

```

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

```

Prease ESC to stop

- Robot动作中停止时按下“ESC”键

```

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

```

forward complete

ENTER

- 动作完成后恢复画面时按下“ENTER”键

```

<EDIT HOME> V: 5 BL
Q4AFL H0
1:0 2:0
3:0 4:0
5:0 6:0

```

JSEL JMOV LMOV

- 恢复画面

■ 确认轴编号

- ① 请参考[3.6.3. MDI “ROBOT 轴编号”](#)。

4.7.3.3.3 LMOV

- 所有轴进行直角动作.

■ 动作方法

Step 1.

接入Servo电源

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JSEL JMOV LMOV



- 执行前电机接入电源.

Step 2.

Deadman switch

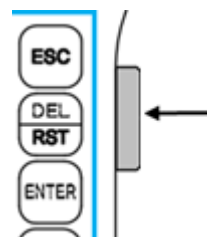
```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JSEL JMOV LMOV

- 执行前利用适当力量抓住Deadman switch..



Step 3.

直角动作选择 画面

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JSEL JMOV **LMOV**

- 所有轴执行直角动作时按下 "F3" 键选择“LMOV”.



- 开始按下 “ENTER”
- 取消按下 “ESC”

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

Ready ? (ENT/ESC)

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

Prease ESC to stop

- Robot动作中停止时按下“ESC”键

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

forward complete

ENTER

- 动作完成后恢复画面时按下“ENTER”键

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JSEL JMOV LMOV

ENTER

- 恢复画面

Step 4.

选择无法使用的Robot时

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JSEL JMOV **LMOV**

- 无法使用的Robot 种类的键操作时显示"only use for SCARA!"

F3

- 恢复画面时按下“ENTER”键

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

only use for SCARA!

```

<EDIT HOME> V: 5 BL
Q4AFL      H0
1:0        2:0
3:0        4:0
5:0        6:0

```

JSEL JMOV LMOV

- 恢复画面

■ 参考事项

- ① 只能执行在“SCARA” Robot 种类，其余Robot无法使用.
- ② 输出"only use for SCARA!" 信息.

Robostar

4.7.4 CORD (Coordinate)

- 可以变更或监控位置变量值坐标系.
- 坐标系以 [JOINT → 直角] 的顺序显示.
- 切换功能.

Step 1.

COORD 画面移动

```

<EDIT HOME> V: 5 BL
Q4AFL      CURR
1:0        2:0
3:0        4:0
5:0        6:0
  
```

not work!!!



- 操作无法使用的Robot种类的键时显示 "not work!!!"

■ 参考事项

- ① 只能执行在 "SCARA" Robot 种类, 其余Robot无法使用.
- ② 输出 "not work!!!" 信息.

4.7.5 AFLD (Arm Folding)

- 与[4.6.5. AFLD \(Arm Folding\)](#) 相同.

4.7.6 JSYN (JMOV 同步移动)

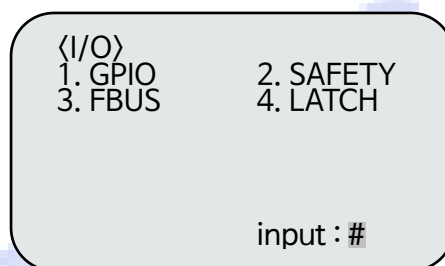
- 与[4.6.6. JSYN \(JMOV 同步移动\)](#)相同.

4.7.7 ZCAL (Z calibration)

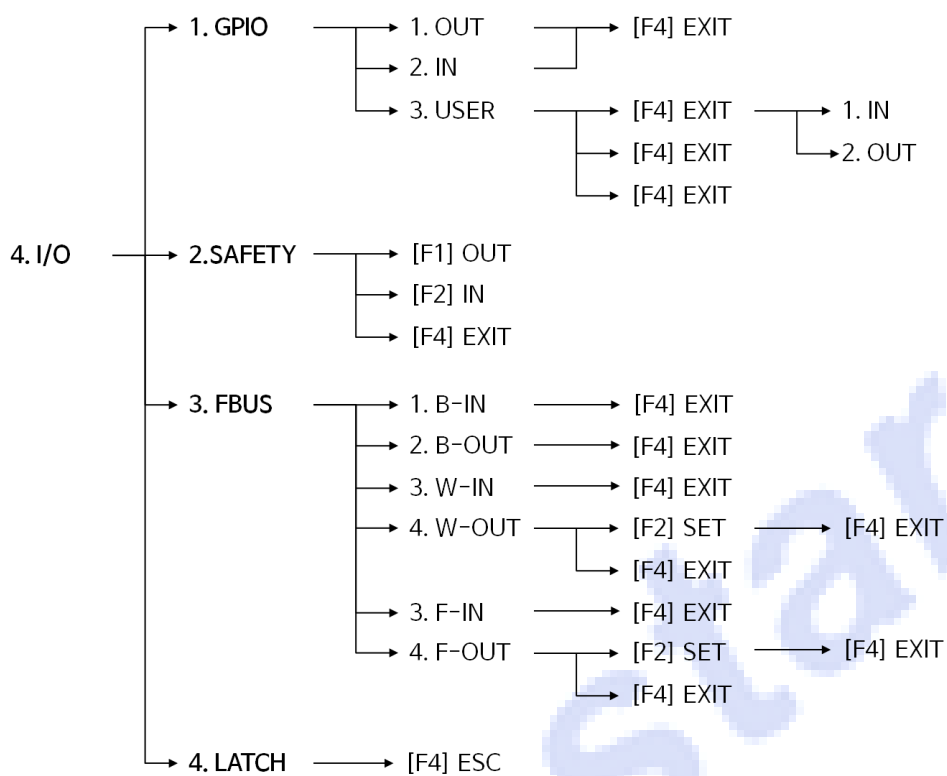
- 与[4.6.7. ZCAL \(Z calibraion\)](#) 相同.

第5章 I/O 模式

- I/O模式大体由3种模式组成，与周边设备(PLC, 传感器, 电磁阀 等)连接的状态下确认周边设备的动作情况.
- 确认用户I/O, SAFETY I/O, LIMIT센서 I/O, HIGHT SPEED I/O等状态.
- I/O模式如下图组成, 提供各种INPUT/OUTPUT 接点的输出、状态确认功能
 - 1) GPIO: 用户I/O 输入输出确认及设定模式
 - 2) SAFETY : Robot INTERLOCK 接点确认模式
 - 3) FBUS : CC-LINK BIT 输入输出确认模式
- 选择性菜单(RS422 通信类型控制器时可以使用)
 - 4) LATCH: 确认与Servo基板连接的Latch 传感器信号值的模式



■ I/O Menu Tree



5.1 GPIO (General Purpose Input Output)

- ✓ 可以显示用户I/O 输入接点或控制输出接点.

5.1.1 概要

- 确认与设备 (Robot,外部设备)连接的I/O 输入输出信号状态.
- 输入输出接点一般有32点.
- 连接扩展I/O 基板时加32点, 可以供使用64个接点.

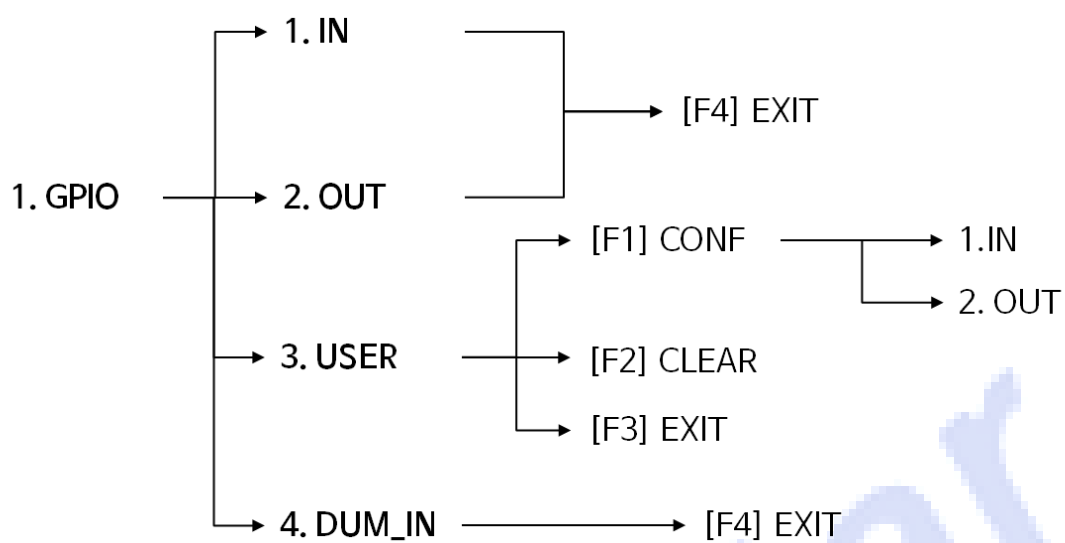
■ 2. GPIO 种类

	GROUP	内容	备注
1 GPIO	1. IN	INPUT	
	2. OUT	OUTPUT	
	3. USER	显示用户设定的I/O 输入输出接点信号	
	4.DUM_IN	显示Dummy 输入接点信号	

5.1.2 GPIO 模式流程图

- GPIO大体分为4个Group.
 - IN: 显示任意I/O 输入接点信号.
 - OUT: 输出任意I/O 输出接点.
 - USER: 显示用户指定的I/O 输入输出接点信号.
 - DUM_IN: 显示除IN中显示的, 作为特殊用途使用的 I/O 输入接点信号.

■ GPIO Menu Tree



5.1.3 GPIO Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version:1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择I/O

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择“4. I/O”

选择GPIO

<I/O>
1. GPIO 2. SAFETY
3. FBUS

input : #



- 选择“1. GPIO”

<I/O - GPIO>
1. IN 2. OUT
3. USER 4. DUM_IN

input #

- GPIO 画面确认

Step 3.

INPUT

〈I/O - GPIO〉	
<u>1. IN</u>	2. OUT
3. USER	4. DUM_IN
input #	



- 确认INPUT 接点时按下“1. IN”键

Step 4.

OUTPUT

〈I/O - GPIO〉	
1. IN	<u>2. OUT</u>
3. USER	4. DUM_IN



- 输出OUTPUT 接电信号时按下 “2. OUT”键

Step 5.

确认用户设定的I/O

〈I/O - GPIO〉	
1. IN	2. IN
<u>3. USER</u>	4. DUM_IN



- 确认用户设定的I/O 接点或输出时按下 “3. USER”键

Step 6.

确认DUMMY INPUT

〈I/O - GPIO〉	
1. OUT	2. IN
3. USER	<u>4. DUM_IN</u>



- 确认用户设定的I/O 接点或输出时按下 “4. DUM_IN”键

5.1.4IN (Input)

- 可以确认任意I/O 输入接点信号状态.

■ OPEN顺序

- 1. GPIO → 1. IN

Step 1.

选择OUT

<I/O - GPIO>	
1. IN	2. OUT
3. USER	4. DUM_IN
input #	



- 确认输入接点信号时按下 "1. IN"键

确认输入接点1/2 显示

<I/O - IN> 1/2			
00:0	01:0	02:0	03:0
04:0	05:0	06:0	07:0
08:0	09:0	10:0	11:0
12:0	13:0	14:0	15:0
B0:0H00	B1:0H00		
EXIT			

- 显示输入接点

■ 输入接点信号确认方法

- 通过例子说明确认INPUT接点信号状态的方法.
 - Ex) 确认INPUT 端口20号的信号状态.

Step 1.

页面变更

<I/O - IN> 1/2			
00:0	01:0	02:0	03:0
04:0	05:0	06:0	07:0
08:0	09:0	10:0	11:0
12:0	13:0	14:0	15:0
B0:0H00	B1:0H00		
EXIT			



- 可以利用"PAGE UP/DOWN" 键切换画面

(I/O - IN) 2/2

16:0	17:0	18:0	19:0
<u>20:1</u>	21:0	22:0	23:0
24:0	25:0	26:0	27:0
28:0	29:0	30:0	31:0
B2:0H00	B3:0H00		

EXIT

5.1.5 OUT (Output)

- 输出任意I/O 输出接点.

■ OPEN顺序

- 1. GPIO → 2. OUT

Step 1.

选择OUT

<I/O - GPIO>
1. IN **2. OUT**
3. USER 4. DUM_IN



- ON/OFF输出接点时按下"2 OUT"键

确认输出接点1/2 显示

<I/O - OUT> 1/2
00: 01:0 02:0 03:0
04:0 05:0 06:0 07:0
08:0 09:0 10:0 11:0
12:0 13:0 14:0 15:0
B0:0H00 B1:0H00

EXIT

- 显示输出接点

■ 输出接点信号控制方法

- 通过例子说明OUTPUT ON/OFF 的方法.
- 按下 "ENTER" 按顺序输出 ON → OFF → ON
 - Ex) ON/OFF OUTPUT 端口 20号.

Step 1.

页面变更

<I/O - OUT> **1/2**
00: 01:0 02:0 03:0
04:0 05:0 06:0 07:0
08:0 09:0 10:0 11:0
12:0 13:0 14:0 15:0
B0:0H00 B1:0H00

EXIT



- 可以利用"PAGE UP/DOWN" 键切换画面

<I/O - OUT> 2/2
 16:■ 17:0 18:0 19:0
 20:0 21:0 22:0 23:0
 24:0 25:0 26:0 27:0
 28:0 29:0 30:0 31:0
 B2:0H00 B3:0H00

EXIT

Step 2.

Cursor位置移动

<I/O - OUT> 2/2
 16:■ 17:0 18:0 19:0
 20:0 21:0 22:0 23:0
 24:0 25:0 26:0 27:0
 28:0 29:0 30:0 31:0
 B2:0H00 B3:0H00

EXIT

<I/O - OUT> 2/2
 16:0 17:0 18:0 19:0
 20:■ 21:0 22:0 23:0
 24:0 25:0 26:0 27:0
 28:0 29:0 30:0 31:0
 B2:0H00 B3:0H00

EXIT



- 按下方向键移动到要输出的编号
- 可以通过 Cursor的位置确认移动结果

Step 3.

信号ON/OFF

开启信号

<I/O - OUT> 2/2
 16:0 17:0 18:0 19:0
 20:1 21:0 22:0 23:0
 24:0 25:0 26:0 27:0
 28:0 29:0 30:0 31:0
 B2:0H00 B3:0H00

EXIT



- 开启信号时按下“ENTER” 键

关闭信号

<I/O - OUT> 2/2
 16:0 17:0 18:0 19:0
 20:0 21:0 22:0 23:0
 24:0 25:0 26:0 27:0
 28:0 29:0 30:0 31:0
 B2:0H00 B3:0H00

EXIT



- 关闭信号时按下 “ENTER” 键

5.1.6 USER (用户 I/O 设定)

- 确认用户设定的I/O 状态.

■ USER 种类

	GROUP	内容	备注
[F3] POINT	[F1]. CONF	I/O 设定	
	[F2]. CLEAR	删除用户设定	
	[F4.] EXIT	退出	

■ OPEN顺序

- 1. GPIO → 3. USER

Step 1.

选择OUT

<I/O - GPIO>
1. OUT
3. USER
2. IN
4. DUM_IN

input #



- 确认用户设定的I/O 目录时按下 "3. USER"键

用户 I/O 画面

<I/O - USER>
NONE:--
NONE:--
NONE:--
NONE:--
NONE:--
NONE:--
CONF CLEAR EXIT

NONE:--
NONE:--
NONE:--
NONE:--
NONE:--
NONE:--
NONE:--

- 确认用户I/O 显示画面

■ 用户 I/O 设定方法

- 用户可以直接设定所需的I/O.
- 设定接点最多可以设定12个.
- 用户I/O 设定时左侧为" INPUT" , 右侧为" OUTPUT"
- 可以设置INPUT/OUTPUT .

Step 1.

Cursor位置移动

<I/O USER>
 IN01: ■ OUT01: 1
 NONE: -- NONE: --
 NONE: NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 CONF CLEAR EXIT

<I/O USER>
 IN01: 0 OUT01: 1
 NONE ■ NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 CONF CLEAR EXIT



- 按下方向键移动到要输出的编号
- 可以通过Cursor的位置确认结果

Step 2.

用户I/O 设定画面

<I/O USER>
 IN01: 0 OUT01: 1
 NONE ■ NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 NONE: -- NONE: --
 CONF CLEAR EXIT

<I/O USER SET>
 1. IN 2. OUT
 input : #



- 设定用户I/O时按下“F1” 键

- 选择用户要设定的“INPUT/OUTPUT”.

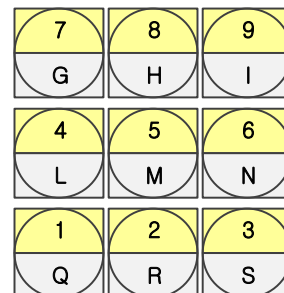
Step 3.

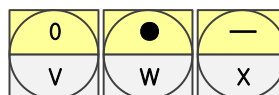
INPUT 设定

<USER SET - IN>
 Index # =2



- 输入INPUT 端口编号.
- 输入结束后按下“ENTER”





```

<I/O USER>
IN01: 0      OUT01: 1
IN02: ■ NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
CONF CLEAR  EXIT
  
```

- 显示“端口编号 2”

Step 4.

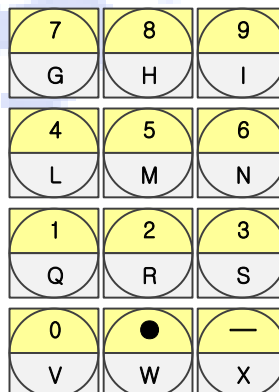
OUTPUT 设定

```

<USER SET - OUT>
  
```

Index # =2

ENTER



- 输入OUTPUT 端口编号.
- 输入结束后按下“ENTER”

```

<I/O USER>
IN01: 0      OUT01: 1
IN02: 1      OUT02: ■
NONE: --    NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
CONF CLEAR  EXIT
  
```

- 显示“端口编号 2”

Step 5.

保存及取消

```

<I/O USER>
IN01: 0      OUT01: 1
IN02: 1      OUT02: ■
NONE: --    NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
NONE: --    NONE: --
CONF CLEAR  EXIT
  
```

F4

- 输入“F4” 端口编号.

```
<I/O USER SET>
IN01:0      OUT01:1
IN02:1      OUT02:■
NONE:--     NONE:--
NONE:--     NONE:--
NONE:--     NONE:--
NONE:--     NONE:--
```

update ? (ENT/ESC)



- 保存时按下“ENTER”
- 取消按下“ESC”

```
<I/O USER>
IN01:0      OUT01:1
IN02:1      OUT02:■
NONE:--     NONE:--
NONE:--     NONE:--
NONE:--     NONE:--
NONE:--     NONE:--
CONF CLEAR  EXIT
```

- 显示用户 I/O 画面

5.1.7 DUM_IN (DUMMY INPUT)

- 显示作为特殊用途使用的I/O 输入接点信号.

■ OPEN顺序

- 1. GPIO → 4. DUM_IN

Step 1.

选择OUT

<I/O - GPIO>
1. IN 2. OUT
3. USER 4. DUM_IN

input #



- 确认DUMMY INPUT 信号时按下"4. DUM_IN"键

确认输入接点1/2 显示

<DUMMY - IN>
DUMMY_IN 00 = 1
DUMMY_IN 01 = 0
DUMMY_IN 02 = 0

- 显示输入接点

■ 功能及画面说明

① 画面说明

- 1) 输出连接到节点的Input信号.当前只使用DUMMY_IN 00, 输出Power switch 信号.

(1表示电源供应中, 0 时表示电源切断)

② 功能说明

- **[ESC]:** 恢复到上一个画面

5.2 SAFETY

- ✓ Safety Part是对外部接口 (Curtain感知, Door Open 等)的 Emergency由硬件组成的部分. 这与MAIN部件的权限无关 , 在Emergency 状况下通过控制器的硬件 切 断 Robot(电机)的驱动电源的方法 , 可以有效对应危险状况. Safety Part中起到关键作用的是 Safety Controller(Unit, PLC).

5.2.1 概要

- Safety Part由Front Board的 Emergency Circuit和控制器内部Safety Unit(PLC)组成. 可以确认各Safety Part的信号
- Safety 画面中OUT 模式优先显示.

■ 2. SAFETY种类

	GROUP	内容	备注
2 SAFETY	1. IN	INPUT	
	2. OUT	OUTPUT	

5.2.2 SAFETY 模式流程图

- SAFETY大体分为两个Group.
 - IN: 显示任意SAFETY I/O 输入接点信号.
 - OUT: 输出任意SAFETY I/O 输出接点.

■ SAFETY Manu Tree



5.2.3 SAFETY Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version:1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER键移动到初始菜单画面

Step 2.

选择I/O

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择“4. I/O”

选择SAFETY

<I/O>
1. GPIO 2. SAFETY
3. FBUS

input : #



- 选择“2. SAFETY”

Step 3

INPUT

<I/O - SAFETY>
1. IN 2. OUT

input : #

- 选择Safety INPUT时按下 “1. IN”键.

Step 4

OUTPUT

<I/O - SAFETY>
1. IN 2. OUT

input : #



- 选择Safety INPUT时按下“2. OUT”键.

5.2.4 IN (Input)

- 可以确认Safety I/O 输入接点信号状态，详细项如下表。

2. SAFETY I/O INPUT 种类

种类	内容	备注
MC1_RD	MC1 状态确认	
MC2_RD	MC2 状态确认	
A_EMG	Auto模式时的Alarm	
TP_EMG	T/P Alarm	
F_EMG	Front 开关Alarm	
ENABLE	设备 Enable or Hold	
LIGHT	Curtain感知	
E_RST	外部重置	
IM_NO	T/P 模式与设备模式不一致时的Alarm (Manual)	
IM_NC	T/P 模式与设备模式不一致时的Alarm (Auto)	
SMPS_F	UPS AC 电源无法接入时的Alarm	
FAN_F	Fan Alarm	
DUMMY2	DUMMY	
DUMMY6	DUMMY	
DUMMY7	DUMMY	
DUMMY8	DUMMY	

OPEN顺序

- 2. SAFETY → 1. IN

Step 1.

INPUT 选择

<I/O - SAFETY>
1. IN 2. OUT

input : #



- 确认Safety I/O 输入接点信号时按下 "1. IN" 键

```

<I/O - Safety IN> 1/2
MC1_RD:1    MC2_RD:1
A_EMG:0     TP_EMG:0
F_EMG:0     ENABLE:0
LIGHT:0     E_RST:0
IM_NO:0     IM_NC:0
EXIT

```

- 显示Safety I/O 输入接点

Step 2.

页面变更

```

<I/O - Safety IN> 1/2
MC1_RD:1    MC2_RD:1
A_EMG:0     TP_EMG:0
F_EMG:0     ENABLE:0
LIGHT:0     E_RST:0
IM_NO:0     IM_NC:0
EXIT

```

Pg Up

- 可以利用”PAGE UP/DOWN” 键切换画面

```

<I/O - Safety IN> 2/2
SMPS_F:1    FAN_F:1
DUMMY2:0    DUMMY6:0
DUMMY7:0    DUMMY8:0
EXIT

```

Pg Dn

面

5.2.5 OUT (Output)

- 可以确认Safety I/O 输出接点信号，详细项如下表

2. SAFETY I/O OUTPUT 种类

种类	内容	备注
M_EMG	MAIN Alarm	
RESET	重置	
DUMMY1	DUMMY	
DUMMY2	DUMMY	
DUMMY3	DUMMY	
DUMMY4	DUMMY	

OPEN顺序

- 2. SAFETY → 2. OUT

Step 1.

选择OUT

⟨I/O - SAFETY⟩
1. IN **2. OUT**

input : #

Safety 输出接点

⟨I/O - Safety OUT⟩
M_EMG:1 RESET:1
DUMMY1:0 DUMMY1:0
DUMMY3:0 DUMMY4:0

EXIT



- 确认Safety I/O 输出接点信号时按下 "2. OUT"键

- 显示Safety I/O 输出接点

5.3 FBUS (Fieldbus)

✓ CC-LINK 输入输出功能

5.3.1 概要

- 确认连接到设备 (Robot,외부장치)의CC-LINK 输入输出信号状态.
- 输入输出接点各有128点.
- WORD DATA 输入输出各由0~15个区域组成.
- FLOAT DATA 输入输出各由0~7个区域组成, 供有WORD 区域.
每个FLOAT编号分配 2WORD.

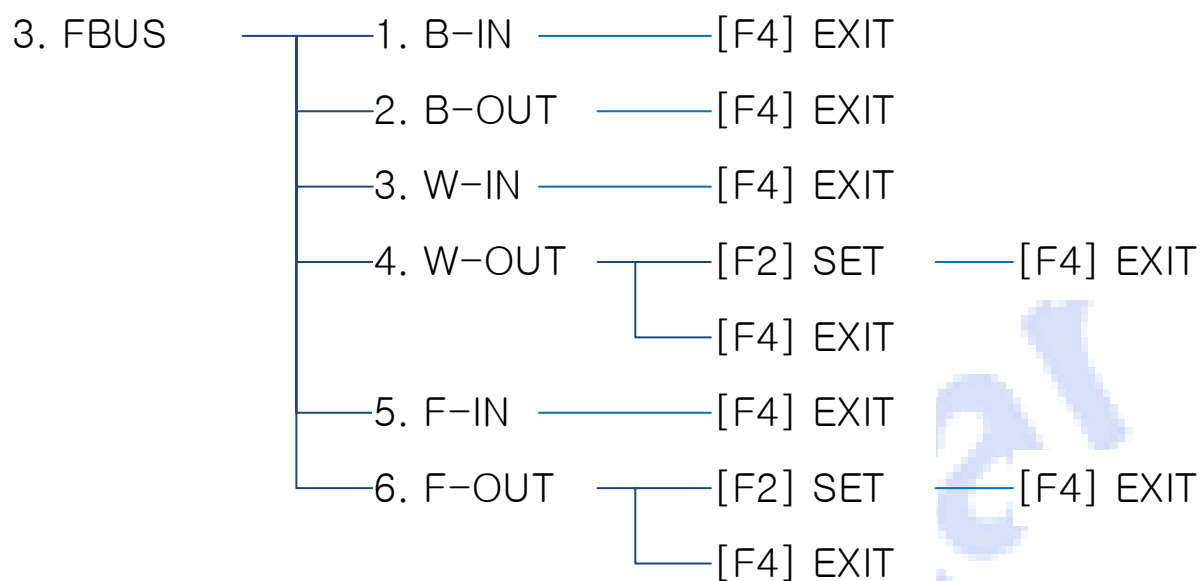
■ 3. FBUS种类

	GROUP	内容	备注
3. FBUS	1. B-IN	FILEDBUS- BIT 输入接点的信号	1 WORD
	2. B-OUT	FILEDBUS- BIT 输出接点的输出	
	3. W-IN	FILEDBUS- WORD DATA 输入显示	
	4. W-OUT	FILEDBUS- WORD DATA 输出	
	5. F-IN	FILEDBUS- FLOAT DATA 输入显示	2 WORD
	6. F-OUT	FILEDBUS- FLOAT DATA 输出	

5.3.2 FBUS 模式流程图

- FBUS大体由6个Group组成.
 - B-IN: 显示任意的I/O 接点输入信号..
 - B-OUT: 输出任意的 I/O 接点输出.
 - W-IN: 显示任意的WORD DATA 输入.
 - W-OUT: 输出任意的WORD DATA 输出.
 - F-IN: 显示任意的FLOAT DATA 输入.
 - F-OUT: 输出任意的FLOAT DATA 输出.

■ FBUS Manu Tree



5.3.3 FBUS Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version:1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER 键移动到初始菜单画面

Step 2.

选择I/O

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择“4. I/O”

选择FBUS

<I/O>
1. GPIO 2. SAFETY
3. FBUS

input : #



- 选择“3. FBUS”

FIELD BUS 画面

<FIELD BUS>
1. B-IN 2. B-OUT
3. W-IN 4. W-OUT
5. F-IN 6. F-OUT

input : #

- 画面确认

Step 3.

BIT输入信号

〈FIELD BUS〉

1. B-IN	2. B-OUT
3. W-IN	4. W-OUT
5. F-IN	6. F-OUT

input : #



- 确认BIT输入信号时按下“1. BIT-IN”键

Step 4.

BIT输出信号

〈FIELD BUS〉

1. B-IN	2. B-OUT
3. W-IN	4. W-OUT
5. F-IN	6. F-OUT

input : #



- 确认BIT输出信号时按下“2. B-OUT”键

Step 5.

WORD DATA 输入

〈FIELD BUS〉

1. B-IN	2. B-OUT
3. W-IN	4. W-OUT
5. F-IN	6. F-OUT

input : #



- 确认WORD DATA 输入值时按下“3. W-IN”键

Step 6.

WORD DATA 输出

〈FIELD BUS〉

1. B-IN	2. B-OUT
3. W-IN	4. W-OUT
5. F-IN	6. F-OUT

input : #



- 输出WORD DATA时按下“4. W-OUT”键

Step 7.

输入FLOAT型 WORD DATA

〈FIELD BUS〉

1. B-IN	2. B-OUT
3. W-IN	4. W-OUT
5. F-IN	6. F-OUT

input : #



- 输出FLOAT型WORD DATA时按下“5. F-IN”键

Step 8.

输出FLOAT型 WORD DATA

<FIELD BUS>

1. B-IN

2. B-OUT

3. W-IN

4. W-OUT

5. F-IN

6. F-OUT

input : #



- 输出FLOAT型 WORD DATA时按下“6. F-OUT”键

5.3.4 B -IN (BIT Input)

- 可以确认任意的CC-LINK BIT 输入接点的信号状态，其项目如下表。

■ OPEN顺序

- 3. FBUS → 1. B-IN

Step 1.

选择BIT INPUT

〈FIELD BUS〉
 1. B-IN 2. B-OUT
 3. W-IN 4. W-OUT
 5. F-IN 6. F-OUT

input : #



- 确认BIT INPUT 信号时按下 “1. BIT-IN” 键

确认输入接点显示

〈FBUS-BIT (IN)〉 1/16
 000:0 001:0
 002:0 003:0
 004:0 005:0
 006:0 007:0
 B00:0H00

EXIT

~

〈FBUS-BIT (IN)〉 16/16
 120:0 121:0
 122:0 123:0
 124:0 125:0
 126:0 127:0
 B15:0H00

EXIT



- BIT 输入接点可以表示0~127范围.

■ 输入接点信号确认方法

1) 页面移动键

- Ex) 确认输入端口编号20号信号状态.

Step 1.

页面变更(页面1)

Step 2.

页面 2

〈FBUS-BIT (IN)〉 1/16

000:0	001:0
002:0	003:0
004:0	005:0
006:0	007:0
B00:0H00	

EXIT



- 利用"PAGE UP/DOWN"可以切换画面

〈FBUS-BIT (IN)〉 2/16

008:0	009:0
010:0	011:0
012:0	013:0
014:0	015:0
B01:0H00	

EXIT



- 利用"PAGE UP/DOWN" 键可以切换画面

Step 3.

页面 3

〈FBUS-BIT (IN)〉 3/16

016:0	017:0
018:0	019:0
020:0	021:0
022:0	023:0
B02:0H00	

EXIT

- 确认BIT接点编号

2) JUMP 功能移动

Step 1.

JUMP移动

〈FBUS-BIT (IN)〉 1/16

000:0	001:0
002:0	003:0
004:0	005:0
006:0	007:0
B00:0H00	

fbus bit = 120■

- 按下Jump 按钮输入要移动的bit值

7	8	9
G	H	I
4	5	6
L	M	N
1	2	3
Q	R	S
0	●	—
V	W	X

5.3.5 B-OUT (BIT Output)

- 可以输出任意的 CC-LINK BIT 输出接点，其项目如下表。

■ OPEN顺序

- 3. FBUS → 2. B-OUT

Step 1.

BIT OUTPUT选择

〈FIELDBUS〉
1. B-IN **2. B-OUT**
3. W-IN 4. W-OUT
5. F-IN 6. F-OUT

input : #



- 确认BIT OUTPUT信号时，按下“2. BIT-OUT” 键

确认输出接点显示

〈FBUS-BIT(OUT)〉 1/16
000:■ 001:0
002:0 003:0
004:0 005:0
006:0 007:0
B00:0H00

EXIT

~

〈FBUS-BIT(OUT)〉 16/16
120:■ 121:0
122:0 123:0
124:0 125:0
126:0 127:0
B15:0H00

EXIT



- BIT输出接点可以显示0~127范围.

■ 输出接点信号控制方法

1) 页面移动

- 页面移动键

Step 1.

页面变更 (页面1)

Step 2.

页面 2

<FBUS-BIT(OUT)> 1/16

000:0	001:0
002:0	003:0
004:0	005:0
006:0	007:0
B00:0H00	

EXIT

Pg Up

Pg Dn

- 利用"PAGE UP/DOWN" 键可以切换画面

Step 3.

页面 3

<FBUS-BIT(OUT)> 2/16

008:0	009:0
010:0	011:0
012:0	013:0
014:0	015:0
B01:0H00	

EXIT

Pg Up

Pg Dn

- 利用"PAGE UP/DOWN" 键可以切换画面，

<FBUS-BIT(OUT)> 3/16

016:0	017:0
018:0	019:0
020:0	021:0
022:0	023:0
B02:0H00	

EXIT

- 确认BIT接点编号

- JUMP 键

Step 1.

JUMP 移动

<FBUS-BIT(OUT)> 1/16

000:0	001:0
002:0	003:0
004:0	005:0
006:0	007:0
B00:0H00	

fbus bit = 120

JUMP

- 按下Jump 按钮输入要移动的bit 值

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

2) 游标移动

Step 1.

游标位置移动

〈FBUS-BIT(OUT)〉 3/16
 016:0 017:0
 018:0 019:0
 020:0 021:0
 022:0 023:0
 B02:0H00

EXIT

〈FBUS-BIT(OUT)〉 3/16
 017:0 018:0
 019:0 020:0
 021:0 022:0
 023:0 024:0
 B02:0H00

EXIT



- 按下方向键移动到要输出的编号
- 通过游标的位置可以确认移动结果

3) 输出信号控制

- 按下“ENTER” 顺序输出 ON → OFF → ON

Step 1.

信号ON/OFF

开启信号

FBUS-BIT(OUT) 3/16
 017:0 018:0
 019:0 020:0
 021:0 022:0
 023:0 024:0
 B02:0H00

EXIT



- 开启信号时按下“ENTER” 键

关闭信号

FBUS-BIT(OUT) 3/16
 017:0 018:0
 019:0 020:1
 021:0 022:0
 023:0 024:0
 B02:0H00

EXIT



- 关闭信号时按下“ENTER” 键

5.3.6 W-IN (WORD Input)

- 可以确认任意的CC-LINK WORD DATA 输入值，项目如下表.

■ OPEN顺序

- 3. FBUS → 3. W-IN

Step 1.

选择WORD DATA INPUT

〈FIELDBUS〉
1. B-IN 2. B-OUT
3. W-IN 4. W-OUT
5. F-IN 6. F-OUT

input : #



- 确认WORD DATA 输入值时按下
“3.W-IN” 键

WORD DATA输入显示确认

〈FBUS-WORD(IN)〉 1/4
IN VALUE
CWI0 0
CWI1 0
CWI2 0
CWI3 0

EXIT

~

〈FBUS-WORD(IN)〉 4/4
IN VALUE
CWI12 0
CWI13 0
CWI14 0
CWI15 0

EXIT



- WORD DATA 区域可以显示 0~15范围.

■ WORD DATA 输入值确认方法

- 1) 页面移动

- 页面移动键

Step 1.

页面变更 (页面 1)

《FBUS-WORD(IN)》 1/4

IN	VALUE
CWI0	0
CWI1	0
CWI2	0
CWI3	0

EXIT



- 可以利用"PAGE UP/DOWN" 键切换画面

- JUMP 键

Step 1.

JUMP 移动

《FBUS-WORD(IN)》 1/4

IN	VALUE
CWI0	0
CWI1	0
CWI2	0
CWI3	0

fbus_port = 12 ■《FBUS-WORD(IN)》 4/4

IN	VALUE
<u>CWI12</u>	0
CWI13	0
CWI14	0
CWI15	0

EXIT



- 按下Jump按钮输入要移动的Port值

7	8	9
G	H	I
4	5	6
L	M	N
1	2	3
Q	R	S
0	●	—
V	W	X

2) 输入值确认

- Ex) 确认WORD区域 4号输入值.

Step 1.

页面 2

《FBUS-WORD(IN)》 2/4

IN	VALUE
<u>CWI4</u>	0
CWI5	0
CWI6	0
CWI7	0

EXIT

- 确认WORD 区域 4号输入值

5.3.7 W-OUT(WORD Output)

- 可以输出任意的WORD DATA 输出值，其项目如下表。

■ 4. W-OUT种类

	GROUP	内容	备注
4. W-OUT	[F1] SET	输出WORD DATA输出值.	
	[F4] EXIT	退出	

■ OPEN顺序

- 3. FBUS → 4. W-OUT

Step 1.

WORD DATA OUTPUT选择

<FIELDBUS>
 1. B-IN 2. B-OUT
 3. W-IN **4. W-OUT**
 5. F-IN 6. F-OUT

input : #



- 输出WORD DATA 输出值时按下 “4. W-OUT” 键

WORD DATA 输出显示确认

<FBUS-WORD(OUT)> 1/4

IN	VALUE
*CWO0	0
CWO1	0
CWO3	0
CWO4	0

SET

EXIT

~

<FBUS-WORD(OUT)> 4/4

IN	VALUE
CWO12	0
CWO13	0
CWO14	0
CWO15	0

EXIT



- WORD DATA 区域可以显示 0~15 范围.

5.3.7.1 SET

- 输出WORD DATA 输出值.
- 输出值可以输出 2byte (-32768~32767)范围.
- 通过例子说明CCLINK WORD 区域输出输出值的方法.
 - Ex) 输出WORD区域 5号输出值.

■ 页面变更

1) 页面移动键

Step 1.

页面变更 (页面1)

<u><FBUS-WORD(OUT)> 1/4</u>	
IN	VALUE
*CWO0	0
CWO1	0
CWO2	0
CWO3	0
SET	EXIT



- 可以利用“ PAGE UP/DOWN” 键切换画面



页面2

<u><FBUS-WORD(OUT)> 2/4</u>	
IN	VALUE
*CWO4	0
CWO5	0
CWO6	0
CWO7	0
SET	EXIT

- WORD DATA 确认

2) JUMP键

JUMP移动

<u><FBUS-WORD(OUT)> 1/4</u>	
IN	VALUE
CWO0	0
CWO1	0
CWO2	0
CWO3	0
<u>fbus port = 12</u>	



- 按下Jump按钮输入要移动的Port值

7	8	9
G	H	I
4	5	6
L	M	N
1	2	3
Q	R	S

<FBUS-WORD(OUT)> 4/4

IN VALUE

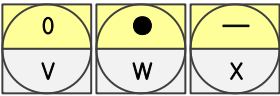
CWIO20

CWIO30

CWIO40

CWIO50

EXIT



■ 光标移动

Step 1.

光标位置移动

<FBUS-WORD(OUT)> 2/4

IN VALUE

*CWO40

~~CWO50~~

CWO60

CWO70

SET EXIT

<FBUS-WORD(OUT)> 2/4

IN VALUE

CWO40

*CWO50

~~CWO60~~

CWO70

SET EXIT



- 按下方向键移动到要输出的编号
- 通过光标的位置可以确认移动结果

■ WORD DATA 输出值输出

Step 1.

WORD DATA 5号画面

<FBUS-WORD(OUT)> 2/4

IN VALUE

CWO40

*CWO50

~~CWO60~~

CWO70

SET EXIT

<FBUS-WORD(OUT)>

CWO0 : 0

NEW : ■

SET EXIT



- 光标移动到要输出的WORD DATA后
输出输出值时按下“F1” 键
- WORD DATA 输出画面确认

Step 2.

输入输出值

<FBUS-WORD(OUT)>

CW00 : 0
NEW : ■

SETEXIT

<FBUS-WORD(OUT)>

CW00 : 0
NEW : 50■

SETEXIT



- 输入输出值 : 50

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

Step 3.

删除

<FBUS-WORD(OUT)>

CW00 : 0
NEW : 501■

SETEXIT

<FBUS-WORD(OUT)>

CW00 : 0
NEW : 50■

SETEXIT



- 错误输入值时按下 “DEL” 键顺序删除输入的值。

Step 4.

输出值输出

<FBUS-WORD(OUT)>

CW00 : 0
NEW : 50■

SETEXIT

<FBUS-WORD(OUT)> 2/4

INVALUE

CW040

*CW0550

CW060

CW070

SETEXIT



- 保存时按下 “ENTER” 键
- 取消时按下 “ESC” 键

5.3.8 F-IN (FLOAT Input)

- 可以确认任意的 CC-LINK FLAOT DATA 输入值，其项目如下表。
- 共享WORD区域，每个 FLOAT 编号分配两个 WORD。
- 可以接收实数型DATA。

■ OPEN顺序

- 3. FBUS → 5. F-IN

Step 1.

FLAOT DATA INPUT 选择

〈FIELD BUS〉	
1. B-IN	2. B-OUT
3. W-IN	4. W-OUT
5. F-IN	6. F-OUT

input : #



- 确认FLOAT DATA 输入值时按下 “5 F-IN” 键

FLOAT형 WORD DATA 输入显示确认

〈FBUS-FLOAT (IN)〉 1/2	
IN	VALUE
CFI0	0
CFI1	0
CFI2	0
CFI3	0

EXIT



- FLAOT DATA 区域可以显示0~7范围.

~

〈FBUS- FLOAT (IN)〉 2/2	
IN	VALUE
CFI4	0
CFI5	0
CFI6	0
CFI7	0

EXIT



■ FLOAT DATA 输入值确认方法

1) 页面移动

- 页面移动键

Step 1.

页面变更 (页面1)

<u><FBUS-FLOAT (IN)> 1/2</u>	
IN	VALUE
CFI0	0
CFI1	0
CFI2	0
CFI3	0
EXIT	



- 可以利用“ PAGE UP/DOWN” 键切换画面



Step 2.

页面2

<u><FBUS- FLOAT (IN)> 2/2</u>	
IN	VALUE
<u>CFI4</u>	<u>0</u>
CFI5	0
CFI6	0
CFI7	0
EXIT	

- 确认FLOAT DATA 区域 4号输入值

- JUMP键

JUMP移动

<u><FBUS-FLOAT (IN)> 1/2</u>	
IN	VALUE
CFI 0	0
CFI 1	0
CFI 2	0
CFI 3	0
<u>fbus port = 4■</u>	

<u><FBUS-FLOAT (IN)> 2/2</u>	
IN	VALUE
<u>CFI 4</u>	0
CFI 5	0
CFI 6	0
CFI 7	0
EXIT	



- 按下Jump按钮输入要移动的Port值

7	8	9
G	H	I
4	5	6
L	M	N
1	2	3
Q	R	S
0	●	—
V	W	X

5.3.9 F-OUT(FLOAT Output)

- 可以输出任意的FLOAT DATA 输出值，其项目如下表.

6. F-OUT的种类

	GROUP	内容	备注
6. F-OUT	[F1] SET	输出FLOAT DATA 输出值.	
	[F4] EXIT	退出	

OPEN顺序

- 3. FBUS → 6. F-OUT

Step 1.

FLAT DATA OUTPUT 选择

<FIELDBUS>
1. B-IN
3. W-IN
5. F-IN

2. B-OUT
4. W-OUT
6. F-OUT

input : #



- 输出FLOAT DATA 输出值时按下 “6 F-OUT” 键

WORD DATA 输出 1/4 显示确认

<FBUS-FLOAT(OUT)> 1/4
IN VALUE

*CF00 0
CF01 0
CF02 0
CF03 0

SET EXIT



- FLOAT DATA 区域可以显示0~7范围.

~

<FBUS-FLOAT(OUT)> 2/2
IN VALUE

CF04 0
CF05 0
CF06 0
CF07 0

EXIT



5.3.9.1 SET

- 输出FLOAT型 WORD DATA 输出值.
- 输出值可以输出 合算2WORD 的4byte的范围.
- 通过例子说明FLOAT区域输出值输出的方法.
 - Ex) 输出FLOAT区域 5号输出值.

■ 页面变更

1) 页面移动键

Step 1.

页面变更 (页面1)

<FBUS-FLOAT (OUT)> 1/2

IN	VALUE
*CFO0	0
CFO1	0
CFO2	0
CFO3	0

SET

EXIT



- 可以利用“ PAGE UP/DOWN” 键切换画面

页面2

<FBUS-FLOAT (OUT)> 2/2

IN	VALUE
*CFO4	0
CFO5	0
CFO6	0
CFO7	0

SET

EXIT

- FLOAT DATA 确认

2) JUMP键

JUMP移动

<FBUS-FLOAT (OUT)> 1/2

IN	VALUE
CFO 0	0
CFO 1	0
CFO 2	0
CFO 3	0

fbus_port = 4 ■



- 按下Jump按钮输入要移动的Port值

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S

<FBUS-FLOAT(OUT)>> 2/2

IN VALUE

CFO 4

0

CFO 5

0

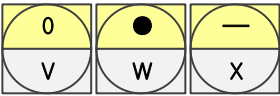
CFO 6

0

CFO 7

0

EXIT



■ 光标移动

Step 1.

光标位置移动

<FBUS- FLAOT (OUT)>> 2/2

IN VALUE

*CFO0

0

CFO1

0

CFO2

0

CFO3

0

SET EXIT



• 按下方向键移动到要输出的编号

<FBUS- FLAOT (OUT)>> 2/2

IN VALUE

CFO4

0

*CFO5

0

CFO6

0

CFO7

0

SET EXIT



• 通过光标的位置可以确认移动结果

■ FLOAT DATA 输出值输出

Step 1.

FLAOT DATA 5号画面

<FBUS-FLAOT(OUT)>> 2/2

IN VALUE

CFO4

0

*CFO5

0

CFO6

0

CFO7

0

SET EXIT



• 光标移动到要输出的FLOAT DATA 后，输出输出值时按下 “F1” 键

<FBUS- FLAOT (OUT)>>

CFO0 : 0

NEW : ■

SET EXIT

• FLOAT DATA 输出画面确认

Step 2.

输入输出值

<FBUS- FLOAT (OUT)>

CF00 : O

NEW : ■

SET

EXIT

<FBUS- FLOAT (OUT)>

CF00 : O

NEW : 50■

SET

EXIT

- 输入输出值 : 50

7 G	8 H	9 I
4 L	5 M	6 N
1 Q	2 R	3 S
0 V	● W	— X

ENTER

Step 2.

删除

<FBUS- FLOAT (OUT)>

CF00 : O

NEW : 501■

SET

EXIT

<FBUS- FLOAT (OUT)>

CF00 : O

NEW : 50■

SET

EXIT



- 错误输入输入值时按下“DEL”键按顺序删除输入值。

Step 3.

输出值输出

<FBUS-WORD(OUT)>

CF00 : O

NEW : 50■

SET

EXIT

<FBUS-WORD(OUT)> 2/4
IN VALUE

CF04	0
*CF05	50
CF06	0
CF07	0

SET

EXIT

ENTER

- 保存时按下“ENTER”键
- 取消时按下“ESC”键

5.4 LATCH (RS422 通信类型 only)

- ✓ 可以确认与Servo基板连接的Latch 传感器信号值.
- ✓ 最多可以粘贴6轴, 各轴最多 4个传感器.

■ OPEN顺序

- Top Screen -> 4. I/O -> 4. LATCH

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

ENTER

- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER 键移动到初始菜单画面

Step 2.

选择I/O

<TEACH MENU>
1. EDIT 2. RUN
3. JOG **4. I/O**
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择“4. I/O”

选择LATCH

<I/O>
1. GPIO 2. SAFETY
3. FBUS **4. LATCH**

input : #



- 选择“4. LATCH”

```

<I/O - LATCH>
  LATCH  SENSOR
    1234  1234
T : 0000  Z : 0000
L : 0000  R : 0000
X : 0000  6 : 0000
          ESC

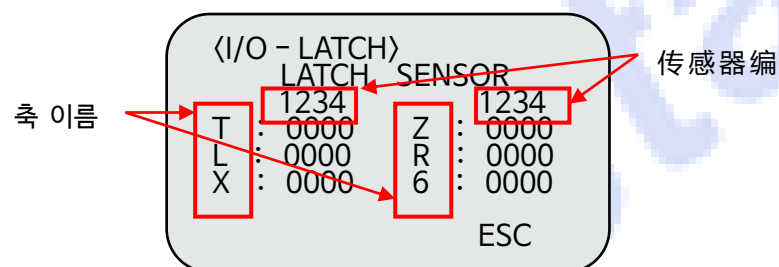
```

- LATCH 画面确认

■ 功能及画面说明

① 画面说明

- 1) 可以确认各轴Latch 传感器状态.



② 功能说明

- **[ESC]:** 移动到上一级画面

第6章 Origin 模式

- 确认Robot原点位置的一系列过程.
- 通常使用相对位置检测式编码器 (Incremental encoder)的Robot在切断电源后无法记住当前Robot的位置，供应电源后移动到Robot的基准位置后，将那个位置定为原点运行Robot，所以需要Origin 功能.
- ORIGIN：可以通过扫描粘贴在器具的原点传感器确认坐标原点位置.

〈ORIGIN〉		
	DIS	SEQ
1:	OFF	2
2:	OFF	2
3:	OFF	2
4:	OFF	2
5:	OFF	2
ORG		EXIT

■ ORIGIN Menu Tree



6.1 ORIGIN

6.1.1 概要

- 利用用户设定的方法扫描初始位置设定坐标原点的功能.
- 为了使用电源重置时丢失当前位置的电机(Incremental Motor, Linear Motor) , 控制器启动后一定要执行Origin.

■ 1. ROBOT种类

	GROUP	内容	备注
5. ORIGIN	[F1] ORG	对于选择的Robot进行Origin 动作.	
	[F4] EXIT	退出Top screen(MAIN) 画面.	

6.1.2 ORIGIN 模式流程图

- ORIGIN大体分为两个Group.
 - ORG: 执行Origin 动作
 - EXIT: 退出到INFO 画面.

6.1.3 ORIGIN Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择ORIGIN

<MAUAL MODE>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 选择5.ORIGIN

Step 3.

选择Robot

<ORIGIN>		
RBT	TYPE	ORG
*R1	R2AWS1	ON
R2	NO-EXIT	---
R3	NO-EXIT	---

SEL

EXIT



- 按下"UP/DOWN" 键选择Robot
- 可以确认选择的Robot一行最前面的 "*" 符号

Robot 名称确认

<ORIGIN>		
RBT	TYPE	ORG
*R1	R2AWS1	ON
R2	NO-EXIT	---
R3	NO-EXIT	---

SEL

EXIT



- 确认输入的Robot 名称时选择Robot 选择后按下 "F1"键选择 "SEL".

Step 4

ORIGIN 画面

<ORIGIN>			
	DIS	SEQ	ORG
1:	OFF	2	ON
2:	OFF	2	ON
3:	OFF	2	ON
4:	OFF	2	ON
5:	OFF	2	OFF
ORG			EXIT

- “ORIGIN” 执行画面

6.1.3.1 ORG (Origin)

- 执行选择的 Robot 的 Origin.

Step 1

执行ORIGIN

<ORIGIN>	DIS	SEQ	ORG
1:	OFF	2	ON
2:	OFF	2	ON
3:	OFF	2	ON
4:	OFF	2	ON
5:	OFF	2	OFF
ORG			EXIT

<ORIGIN>
Forwarding

Axis Info :
3 4

Press ESC to stop

- 按下“F1”键执行“ORIGIN”动作

- Origin Scanning

6.1.3.2 EXIT

- 退出[ORIGIN] 画面.
- 退出的方法有两种.

按下 “[F4] EXIT” 键退出的方法

按下 “ESC” 退出的方法

■ 退出方法

Step 1

ORIGIN 画面

<ORIGIN>		
DIS	SEQ	ORG
1: OFF	2	ON
2: OFF	2	ON
3: OFF	1	ON
4: OFF	1	ON
5: OFF	2	OFF
ORG		EXIT

F4

- 退出[ORIGIN] 画面时按下
“F4” 键

<ORIGIN>		
RBT	TYPE	ORG
*R1	R2AWS1	ON
R2	NO-EXIT	---
R3	NO-EXIT	---
SEL		EXIT

F4

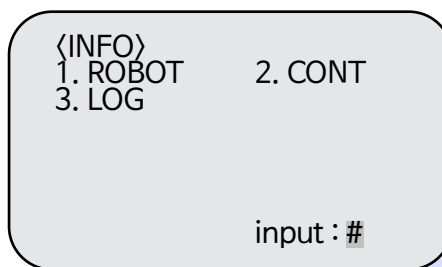
- 退出Main 画面时按下
“F4” 键

<MANUAL MODE>	
1. EDIT	2. RUN
3. JOG	4. I/O
5. ORIGIN	6. INFO
0. SHUTDOWN	
input : #	

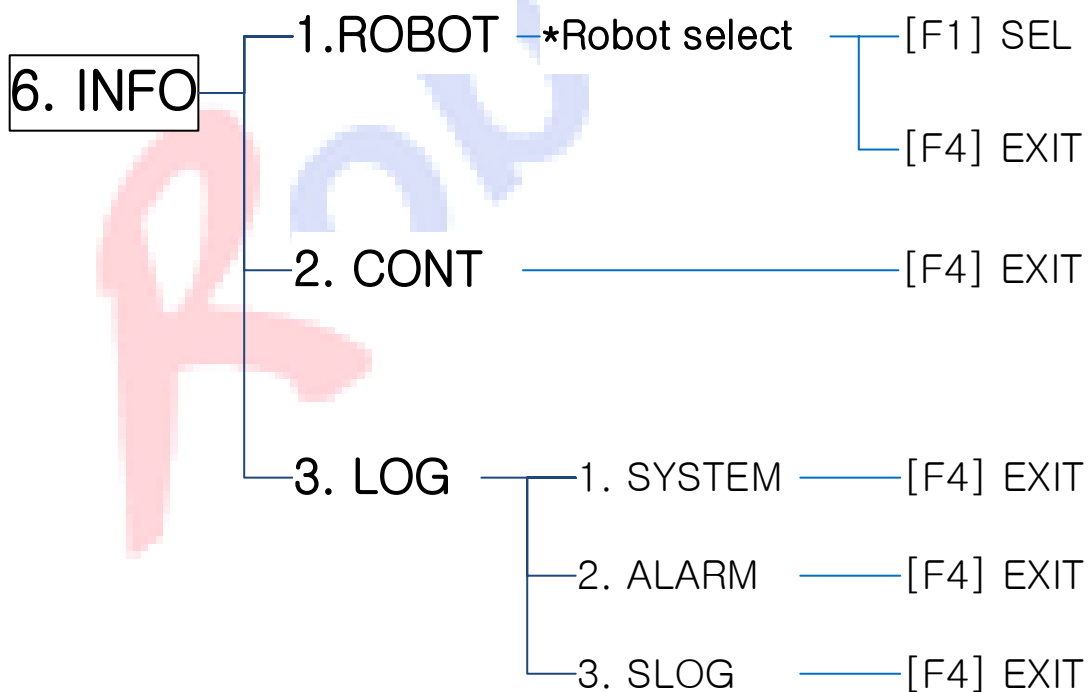
- Main画面确认

第7章 INFO 模式

- 可以诊断系统各种状态的模式.
- INFO 模式由3种(ROBOT, CONT, LOG) 组成.
 - ROBOT : 可以确认Robot 类型/系列/规格设定信息.
 - CONT: 可以确认控制器F/W 信息.
 - LOG: 可以确认系统启动日志、系统Alarm日志、用户日志等.
 - USB: 通过USB下载控制器上的文件或控制器上传USB上的文件的功能.



■ INFO Menu Tree



7.1 ROBOT (ROBOT 类型/规格/系列)

7.1.1 概要

- 可以确认Robot 类型/系列/规格等设定信息.

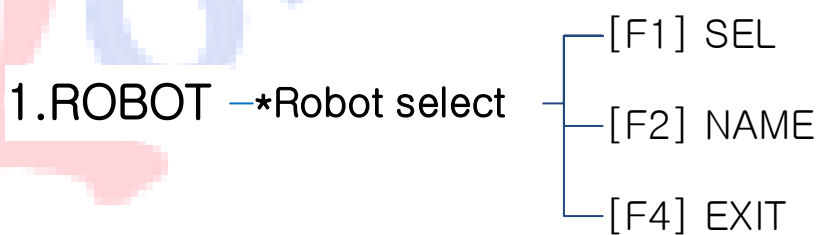
■ 1. ROBOT种类

	GROUP	内容	备注
1 ROBOT	[F1] SEL	选择Robot	
	[F2] NAME	输入ROBOT 名称	
	[F4] EXIT	INPUT	

7.1.2 ROBOT 模式流程图

- GPIO大体分为两个Group.
 - SEL: 选择任意Robot.
 - NAME : 输入Robot 名称.
 - EXIT: 退出到INFO 画面.

■ ROBOT Menu Tree



7.1.3 ROBOT Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version:1.01.01
(E 150922)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择INFO

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN **6. INFO**

0. SHUTDOWN
input : #



- 选择6. INFO

选择ROBOT

<INFO>
1. ROBOT 2. CONT
3.LOG

input : #



- 选择1. ROBOT

Step 3.

选择Robot

<EDIT PARA>
ROBOT TYPE
*R1 R2AWS1
R2 NO-EXIT
R3 NO-EXIT

SEL NAME EXIT



- 按下"UP/DOWN" 键选择Robot
- 可以确认选择的Robot一行最前面的 "*" 符号

Robot 名称确认

```
<INFO ROBOT>
ROBOT      TYPE
*R1        R2AWS1
R2         NO-EXIT
R3         NO-EXIT
```

SEL NAME EXIT



- 确认输入的Robot 名称时选择Robot 选择后按下“F1”键选择“SEL”。

Robot 名称变更

```
<INFO ROBOT>
ROBOT      TYPE
*R1        R2AWS1
R2         NO-EXIT
R3         NO-EXIT
```

SEL **NAME** EXIT

- 变更输入的Robot 名称时选择Robot 选择后按下“F2”键选择“NAME”。

Step 3.

Robot 名称确认

```
<ROBOT 1 INFO>
NAME: Q4AFL
```

EXIT

- 确认用户输入的Robot 名称。

7.1.3.1 SEL (Select)

- 确认用户输入的各Robot名称.

Step 1.

选择Robot

<EDIT PARA>	
ROBOT	TYPE
*R1	R2AWS1
R2	NO-EXIT
R3	NO-EXIT

SEL	NAME	EXIT



- 按下"UP/DOWN" 键选择Robot



- 可以确认选择的Robot一行最前面的 "*" 符号

Robot 名称确认

<INFO ROBOT>	
ROBOT	TYPE
*R1	R2AWS1
R2	NO-EXIT
R3	NO-EXIT

SEL	NAME	EXIT



- 确认输入的Robot 名称时选择Robot 选择后按下 "F1"键选择 "SEL".

Step 2.

Robot 名称确认

<ROBOT 1 INFO>	
NAME: Q4AFL	

EXIT

- 确认用户输入的Robot名称.

7.1.3.2 NAME

- 可以确认用户输入的各Robot名称.

Step 1.

Robot 选择

<EDIT PARA>	
ROBOT	TYPE
*R1	R2AWS1
R2	NO-EXIT
R3	NO-EXIT
SEL NAME EXIT	



- 按下"UP/DOWN" 键选择Robot



- 可以确认选择的Robot一行最前面的
"*" 符号

Robot 名称确认

<INFO ROBOT>	
ROBOT	TYPE
*R1	R2AWS1
R2	NO-EXIT
R3	NO-EXIT
SEL NAME EXIT	



- 确认输入的Robot 名称时选择Robot
选择后按下 "F2"键选择 "NAME".

Step 2.

Robot 名称

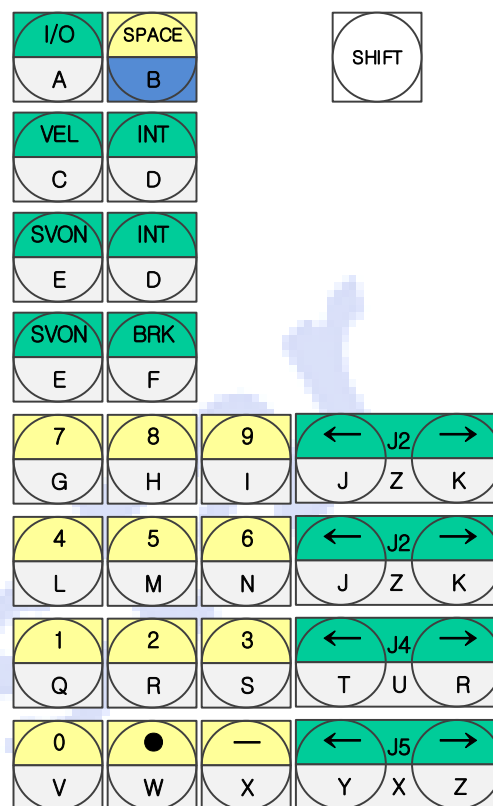
<ROBOT 1 INFO>	
NAME: Q4AFL	
NEW : ■	
EXIT	

- 变更用户输入的Robot名称.

Step 3.

输入新名称

- 按下数字和文字键输入新名称内容



Step 4.

输入“Q”

(ROBOT 1 INFO)
 NAME: Q4AFL
 NEW : ■
 EXIT

(ROBOT 1 INFO)
 NAME: Q4AFL
 NEW : Q■
 EXIT



- 开始输入Q4AFL1!



- 首先要按下“SHIFT”键才能输入文字键。接下来按下文字“Q”键。

Step 5

输入“4”

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q■

EXIT



- 首先解除“SHIFT”功能. 接下来按下数字“4”键.

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4■

EXIT



Step 6.

输入“AFL”

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4■

EXIT



- 首先需要按下“SHIFT”键才能输入文字键. 接下来按顺序按下文字键“A”, “F”, “L”.

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4AFL■

EXIT



Step 7

输入“1”

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4AFL■

EXIT



- 首先解除“SHIFT”功能. 接下来按下数字“1”键.

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4AFL1■

EXIT



Step 8.

保存及取消

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4AFL1 ■

EXIT



- 输入名称后按下“ENTER”键，update 信息显示在画面

<ROBOT 1 INFO>
NAME: Q4AFL

NEW : Q4AFL1 ■

update ? (ENT/ESC)



- 保存时按下 “ENTER”键
- 取消按下 “ESC”键

<INFO ROBOT>
ROBOT TYPE
*R1 R2AWS1
-R2 NO-EXIT
R3 NO-EXIT

SEL	NAME	EXIT

7.1.4 EXIT

- 退出[INFO ROBOT] 画面.
- 退出方法有两种.

按下 “[F4] EXIT” 键退出的方法

按下 “ESC” 键退出的方法

■ 退出方法

Step 1.

FLOAT 画面

<INFO ROBOT>	
ROBOT	TYPE
*R1	R2AWS1
-R2	NO-EXIT
R3	NO-EXIT
SEL NAME EXIT	

<INFO>	
1. ROBOT	2. CONT
3. LOG	
input : #	



- 退出[INFO ROBOT] 画面时按下 “F4” 键

- INFO 画面确认

7.2 CONT (Control)

7.2.1 概要

- 可以确认控制器F/W 信息.

■ ROBOT Menu Tree

2. CONT ————— [F4] EXIT

7.2.2 ROBOT Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version :1.01.01
(E 150922)

Press ENTER Key

ENTER

- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO

0. SHUTDOWN
input : #

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择INFO

<TEACH MENU>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN **6. INFO**

0. SHUTDOWN
input : #



- “6. INFO” 选择

选择控制器

<INFO>
1. ROBOT 2. CONT
3.LOG

input : #



- 确认控制器F/W 版本信息时按下“ 2. CONT”键

Step 3.

确认F/W 版本

<INFO - CONT>
Ver: RcX
1.01.01
(E 150922)

EXIT

- 可以确认控制器F/W 信息.

Step 4.

退出

<INFO - CONT>
Ver: RcX
1.01.01
(E 150922)

EXIT

<INFO>
1. ROBOT 2. CONT
3.LOG

input : #

- 确认控制器F/W信息后退出 INFO 画面时按下“EXIT”键

7.3 LOG

7.3.1 概要

- 可以确认系统启动日志、系统Alarm日志、用户日志等.

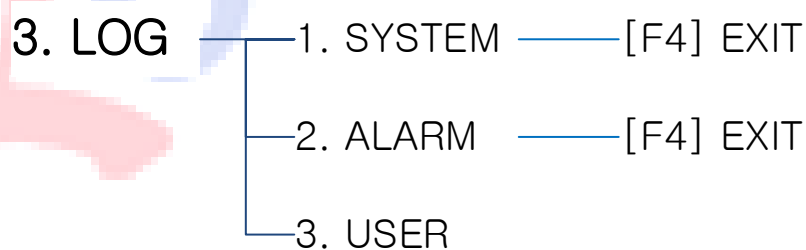
■ 3. LOG 种类

	GROUP	内容	备注
3. LOG	1. SYSTEM	控制器电源ON/OFF 历史记录	
	2. ALARM	控制器Alarm记录	
	3. USER	用户设定记录	

7.3.2 ROBOT 模式流程图

- LOG 大体由3个Group组成.
 - SYSTEM: 记录按照时间/日期ON/OFF控制器电源的内容.
 - ALARM : 记录各时间/日期的控制器Alarm内容.
 - USER: 记录各时间/日期用户制作的JOB程序内容.

■ 3. LOG Menu Tree



7.3.3 LOG Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RCM6-Series
Version :3.01.33
(TR 141009)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. INFO 4. I/O
5. ORIGIN

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER Key 移动到初始菜单画面

Step 2.

选择INFO

<TEACH MENU>
1. EDIT 2. RUN
3. INFO 4. I/O
5. ORIGIN

0. SHUTDOWN
input : #



- 选择“3. INFO”

选择ROBOT

<INFO>
1. ROBOT 2. CONT
3.LOG

input : #



- 选择“3. LOG”

确认控制器电源记录

<INFO - LOG>
1. SYSTEM 2. ALARM
3.USER

input : #



- 确认控制器电源ON/OFF 记录时按下“1. SYSTEM” 键

确认控制器Alarm记录

<INFO - LOG>
1. SYSTEM 2. ALARM
3. USER

input : #

确认用户设定记录

<INFO - LOG>
1. SYSTEM 2. ALARM
3. USER

input : #



- 确认控制器Alarm记录时按下“2 .ALARM”键



- 确认用户设定记录时按下“3. USER”键

7.3.3.1 SYSTEM

- 记录了System 电源 ON/OFF 历史.

Step 1.

确认控制器电源记录

<INFO - LOG>
1. SYSTEM 2. ALARM
3. USER

input : #

SYSTEM LOG
92: Open the system msg
93: [15-09-22 13:55:04]
94: Open the system msg
95: [15-09-24 13:58:02]
96: Open the system msg
97: [15-09-25 13:58:02]
EXIT



- 确认控制器电源ON/OFF 记录时按下
“1. SYSTEM” 键

7.3.3.2 ALARM

- 记录了System alarm历史.

Step 1.

确认控制器Alarm记录

<INFO - LOG>
1. SYSTEM 2. ALARM
3. USER

input : #

SYSTEM LOG
93: [15-09-22 13:55:04]
94: Open the alarm file ms
95: [15-09-24 14:58:02]
94: Open the alarm file ms
97: [15-09-25 15:58:02]
98: Open the alarm file ms
EXIT



- 确认控制器Alarm记录时按下“2. ALARM”键

7.3.3.3 USER

- 记录了用户输入的 'print 指令Log 历史.

Step 1.

确认用户设定历史

<INFO - LOG>
1. SYSTEM 2. ALARM
3. USER

input : #

SYSTEM LOG
00: [15-09-22 13:55:04]
01: [0:0] save test
02: [15-09-24 14:58:02]
03: [0:0] save test
04: [15-09-25 15:58:02]
05: [0:0] save test
EXIT



- 确认用户设定历史时按下“3. USER”键

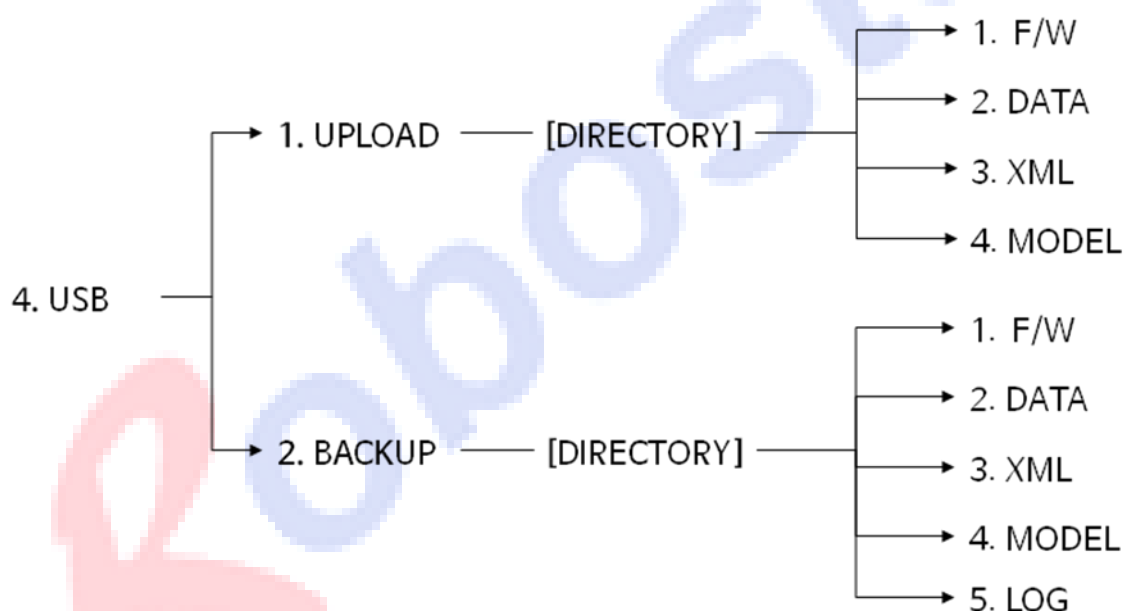
7.4 USB (Version 1.01.13 以上支持)

7.4.1 概要

- 通过USB下载控制器上的文件 (Firmware, JOB, Point, Parameter, Global Integer, Global Float, XML, MODEL 등)或控制器上传USB上的文件的功能.

7.4.2 USB 模式流程图

- USB 画面大体分为两个Group.
 - UPLOAD: 将USB中的文件复制到控制器.
 - BACKUP: 将控制器上的文件复制到 USB.



7.4.3 USB Group画面 OPEN顺序

Step 1.

初始菜单画面

Robostar
NewRo RcX-Series
Version:1.01.13
(TE170209__)
Press ENTER Key

<MANUAL MODE>
1. EDIT 2. RUN
3. JOG 4. I/O
5. ORIGIN 6. INFO
7. MOVE
0. SHUTDOWN
input : #



- 投入电源LCD上输出画面(发生错误时，请按照错误代码处理)

- 按下ENTER Key移动到初始画面

Step 2.

INFO 选择

<MANUAL MODE>
1. EDIT 2. RUN
3. TEACH 4. I/O
5. ORIGIN **6. INFO**
7. MOVE
0. SHUTDOWN
input : #



- 选择 "6. INFO"

USB 选择

<INFO>
1. ROBOT 2. CONT
3. LOG **4. USB**
input : #



- 选择 "4. USB"

Upload

<INFO - USB>
1. UPLOAD 2. BACKUP
input : #



- USB上的文件上传到控制器时按下 "1. UPLOAD" 键

确认控制器Alarm历史

<INFO - LOG>
1. UPLOAD 2. BACKUP

input : #



- 控制器上的文件备份到USB时，按下
“2. BACKUP” 键

7.4.3.1 UPLOAD

- USB中的文件上传到控制器，其详细项目如下表.

■ 详细项目

	GROUP	内容	备注
1. UPLOAD	1. F/W	升级Firmware.	
	2. DATA	升级JOB, POINT, INTEGER, FLOAT, PARAMETER文件.	
	3. XML	升级XML文件.	
	4. MODEL	升级MODEL文件.	

7.4.3.1.1 F/W

- 升级, Firmwar.

Step 1.

选择上传

<INFO - USB>
1. UPLOAD 2. BACKUP

input : #



- 选择 “1. UPLOAD”

USB路径选择

<SELECT DIR>
 PATH: /
 *TR-1.2.3
 boot
 css
 dists
 abc

ENT SEL



- 上下移动光标选择上传的数据所在的文件夹.
- 选择文件夹后按下 “F4” 键, 显示*标识的文件夹选择文作业文件夹.
- ENT移动到当前 * 标识所在文件夹下面的文件夹.

F/W选择

<INFO - UPLOAD>
1. F/W 2. DATA
 3. XML 4. MODEL

PATH: /TR-1.2.3
 input : #



- 将USB中的 Firmware 文件上传到控制器时按下 “1. F/W” 键

Firmware update

<UPDATE FIRMWARE>
 Do you want to
 update the firmware
 file?

(ENT/ESC)



- 输出是否确认上传Firmwarefile的信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键时恢复到之前画面.

Firmware update 确认

〈UPDATE FIRMWARE〉
Please wait

〈UPDATE FIRMWARE〉
Update Success!!

Press Any Key

- 正常结束Firmware上传，输出，
Update Success.

7.4.3.1.2 DATA

- 上传DATA. 其详细项目如下表.

■ 详细项目

GROUP		内容	备注
2. DATA	1. JOB	更新JOB 文件.	
	2. PNT	更新POINT 文件.	
	3. INT	更新INTEGER文件.	
	4. FLT	更新FLOAT文件.	
	5. PARA	更新PARAMETER文件.	
	0. ALL	全部更新JOB, POINT, INTEGER, FLOAT, PARAMETER 文件.	

1) JOB

Step 1.

DATA 选择

<INFO - UPLOAD>
 1. F/W **2. DATA**
 3. XML 4. MODEL

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

JOB 选择

<UPLOAD - DATA>
1. JOB 2. PNT
 3. INT 4. FLT
 5. PARA
 0. ALL

PATH: /TR-1.2.3
 input : #



- 选择 “1. JOB”

JOB update

<UPLOAD JOB FILE>
 Do you want to
 upload the job
 file?

(ENT/ESC)

- 输出是否上传job file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与上传的文件相同名称的文件时, 弹出警告窗口.



<UPLOAD JOB FILE>
 [CAUTION]
 AGT.JOB
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP:覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

JOB update 确认

<UPLOAD JOB FILE>
 Copying Success!!

Press Any Key

- JOB上传正常结束后 , 输出 Copying Success.

2) PNT

Step 1.

DATA 选择

<INFO - UPLOAD>
1. F/W
2. DATA
3. XML
4. MODEL

PATH: /TR-1.2.3
input : #



- 选择 “2. DATA”

PNT 选择

<UPLOAD - DATA>
1. JOB
2. PNT
3. INT
4. FLT
5. PARA
0. ALL

PATH: /TR-1.2.3
input : #



- 选择 “2. PNT”

PNT update

<UPLOAD PNT FILE>
Do you want to
upload the point
files?

(ENT/ESC)



- 输出是否上传point file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.

PNT update 确认

<UPLOAD PNT FILE>
Copying Success!!

Press Any Key

- POINT上传正常结束后 , 输出 Copying Success.

3) INT

Step 1.

DATA 选择

<INFO - UPLOAD>
 1. F/W **2. DATA**
 3. XML 4. MODEL

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

INT 选择

<UPLOAD - DATA>
 1. JOB 2. PNT
3. INT 4. FLT
 5. PARA
 0. ALL

PATH: /TR-1.2.3
 input : #



- 选择 “3. INT”

INT update

<UPLOAD GIT FILE>
 Do you want to
 upload the GIT
 files?

(ENT/ESC)



- 输出是否上传integer file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.

INT update确认

<UPLOAD GIT FILE>
 Upload Success!!

Press Any Key

- integer file上传正常结束后, 输出 Upload Success.

4) FLT

Step 1.

DATA 选择

<INFO - UPLOAD>
1. F/W
3. XML
2. **DATA**
4. MODEL

PATH: /TR-1.2.3
input: #



- 选择 “2. DATA”

FLT 选择

<UPLOAD - DATA>
1. JOB
3. INT
5. PARA
0. ALL
2. PNT
4. **FLT**

PATH: /TR-1.2.3
input: #



- 选择 “4. FLT”

FLT update

<UPLOAD GFT FILE>
Do you want to
upload the GFT
files?

(ENT/ESC)



- 输出是否上传float file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.

FLT update 确认

<UPLOAD GFT FILE>
Upload Success!!

Press Any Key

- float file上传正常结束后, 输出 Upload Success.

5) PARA

Step 1.

DATA 选择

<INFO - UPLOAD>
1. F/W
3. XML
2. DATA
4. MODEL

PATH: /TR-1.2.3
input : #



- 选择 “2. DATA”

PARA 选择

<UPLOAD - DATA>
1. JOB
3. INT
5. PARA
0. ALL
2. PNT
4. FLT

PATH: /TR-1.2.3
input : #



- 选择 “5. PARA”

PARA update

<UPLOAD PARAMETER>
Do you want to
upload the
parameter files?

(ENT/ESC)



- 输出是否上传parameter file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.

PARA update 确认

<UPLOAD PARAMETER>
Upload Success!!

Press Any Key

- parameter file上传正常结束后, 输出 Upload Success.

6) ALL

Step 1.

DATA 选择

<INFO - UPLOAD>
 1. F/W **2. DATA**
 3. XML 4. MODEL

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

ALL 选择

<UPLOAD - DATA>
 1. JOB 2. PNT
 3. INT 4. FLT
 5. PARA
0. ALL

PATH: /TR-1.2.3
 input : #



- 选择 “0. ALL”

ALL update

<UPLOAD DATA>
 Do you want to
 upload the all
 data files?

(ENT/ESC)



- 输出是否上传所有 file 的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.

ALL update 확인

<UPLOAD DATA>
 Upload Success!!

Press Any Key

- 所有file上传正常结束后 , 输出 Upload Success.

7.4.3.1.3 XML

- 上传XML 文件.

Step 1.

选择XML

<INFO - UPLOAD>
 1. F/W 2. DATA
3. XML 4. MODEL

PATH: /TR-1.2.3
 input : #



- 选择 “3. XML”

XML update

<UPLOAD XML FILE>
 Do you want to
 upload the xml
 file?

(ENT/ESC)



- 输出是否上传xml file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与上传的文件相同名称的文件时, 弹出警告窗口.

<UPLOAD XML FILE>
 [CAUTION]
 2^2^7%1^1^ .xml
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP:覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

XML update 确认

<UPLOAD XML FILE>
 Upload Success!!

Press Any Key

- xml file 上传正常结束后, 输出 Upload Success.

7.4.3.1.4 MODEL

- 上传MODEL 文件.

Step 1.

MODEL 选择

<INFO - UPLOAD>
1. F/W 2. DATA
3. XML 4. MODEL

PATH: /TR-1.2.3
input : #



- 选择 “4. MODEL”

MODEL update

<UPLOAD MODEL FILE>
Do you want to
upload the model
file?

(ENT/ESC)

- 输出是否上传model file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与上传的文件相同名称的文件时, 弹出警告窗口.



<UPLOAD MODEL FILE>
[CAUTION]
UAF4B1L100Q-240-300.
already exists!!
Do you want to
overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP:覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

MODEL update 确认

<UPLOAD MODEL FILE>
Upload Success!!

Press Any Key

- model file 上传正常结束后, 输出 Upload Success.

7.4.3.2 BACKUP

- 将控制器上的文件保存到 USB，详细项目如下表。

■ 详细项目

GROUP		内容	备注
2. BACKUP	1. F/W	备份Firmware.	
	2. DATA	备份JOB, POINT, INTEGER, FLOAT, PARAMETER文件.	
	3. XML	备份XML文件.	
	4. MODEL	备份MODEL文件.	
	5. LOG	备份LOG 文件.	

7.4.3.2.1 F/W

- 备份Firmware.

Step 1.

选择上传

<INFO - USB>
1. UPLOAD **2. BACKUP**

input : #



- 选择 “2. BACKUP”

USB 路径选择

<SELECT DIR>
PATH: /
*TR-1.2.3
boot
css
dists
abc
MAKE DEL ENT SEL



- 上下移动光标选择上传的数据所在的文件夹.
- 选择文件夹后按下 “F4” 键，显示*标识的文件夹选择文作业文件夹.
- ENT移动到当前 * 标识所在文件夹下面的文件夹.
- MAKE生成文件夹.
- DEL删除文件夹.

F/W 选择

<INFO - BACKUP>
1. F/W 2. DATA
3. XML 4. MODEL

PATH: /TR-1.2.3
input : #



- 将USB中的 Firmware 文件上传到控制器时按下 “1. F/W” 键

Firmware backup

<BACK UP FIRMWARE>
Do you want to
backup the firmware
file?

(ENT/ESC)



- 输出是否确认备份Firmware file的信息，需要上传时按下 “ENTER” 键
- 按下 “ESC” 键时恢复到之前画面.

Firmware存在时

〈BACK UP FIRMWARE〉
[CAUTION]
firmware already
exists!!
Do you want to
overwrite it?

(ENT/ESC)



- USB中存在要保存的Firmware时，确认是否覆盖。
- 选择ENTER，USB上的fireware被控制器上的firmware覆盖掉。

Firmware backup 确认

〈BACK UP FIRMWARE〉
Please wait

〈BACK UP FIRMWARE〉
Backup Success!!

Press Any Key

- Firmware 备份正常结束后，输出 Backup Success.

7.4.3.2.2 DATA

- 备份DATA. 详细项目如下表.

■ 详细项目

GROUP		内容	备注
2. DATA	1. JOB	备份JOB 文件.	
	2. PNT	备份POINT文件.	
	3. INT	备份INTEGER文件.	
	4. FLT	备份FLOAT文件.	
	5. PARA	备份PARAMETER文件.	
	0. ALL	备份全部JOB, POINT, INTEGER, FLOAT, PARAMETER 文件.	

1) JOB

Step 1.

DATA 选择

<INFO - BACKUP>
 1. F/W **2. DATA**
 3. XML 4. MODEL
 5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

JOB 选择

<BACKUP - DATA>
1. JOB 2. PNT
 3. INT 4. FLT
 5. PARA
 0. ALL

PATH: /TR-1.2.3
 input : #



- 选择 “1. JOB”

JOB backup

<BACK UP JOB FILE>
 Do you want to
 backup the job
 files?

(ENT/ESC)

- 输出是否备份job file的确认信息, 需要备份时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与备份的文件相同名称的文件时, 弹出警告窗口.



<BACK UP JOB FILE>
 [CAUTION]
 AGT.JOB
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP:覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

JOB backup 确认

<BACK UP JOB FILE>
 Copying Success!!

Press Any Key

- JOB 备份正常结束后, 输出 Copying Success.

Step 1.

<INFO - BACKUP>
 1. F/W
 3. XML
 5. LOG
 2. **DATA**
 4. MODEL

- 选择 “2. DATA”

(BACKUP - DATA)
 1. JOB
 3. INT
 5. PARA
 0. ALL

- 选择 “2. PNT”

Do you want to backup the point files?

(ENT/ESC)

- 输出是否备份point file的确认信息, 需要上传时按下 “**ENTER**” 键
- **按下 “ESC” 键**恢复到之前画面.
- 存在与要备份的文件相同名的文件时, 弹出警告窗口.

ENTER

<BACK UP PNT FILE>
 [CAUTION]
 RBT0.PNT
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

〈BACK UP PNT FILE〉
Copying Success!!

Press Any Key

- POINT 备份正常结束后，输出 Copying Success.

3) INT

Step 1.

DATA 选择

<INFO - BACKUP>
 1. F/W **2. DATA**
 3. XML 4. MODEL
 5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

INT 选择

<BACKUP - DATA>
 1. JOB 2. PNT
3. INT 4. FLT
 5. PARA
 0. ALL

PATH: /TR-1.2.3
 input : #



- 选择 “3. INT”

INT backup

<BACK UP GIT FILE>
 Do you want to
 backup the GIT
 files?

(ENT/ESC)

- 输出是否备份integer file的确认信息，需要上传时按下 “ENTER” 键



<BACK UP GIT FILE>
 [CAUTION]
 GIT already
 exists!!
 Do you want to
 overwrite it?

(ENT/ESC)

- 按下 “ESC” 键恢复到之前画面
- GIT 文件存在时弹出警告窗口。

INT backup 确认

<BACK UP GIT FILE>
 Backup Success!!

Press Any Key

- integer file 备份正常结束后，输出 Backup Success.

4) FLT

Step 1.

DATA 选择

<INFO - BACKUP>
 1. F/W
 3. XML
 5. LOG
 2. DATA
 4. MODEL

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

FLT 选择

<BACKUP - DATA>
 1. JOB
 3. INT
 5. PARA
 0. ALL
 2. PNT
 4. FLT

PATH: /TR-1.2.3
 input : #



- 选择 “4. FLT”

FLT update

<BACK UP GFT FILE>
 Do you want to
 backup the GFT
 files?

(ENT/ESC)



- 输出是否备份float file的确认信息，需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面

<BACK UP GFT FILE>
 [CAUTION]
 GFT already
 exists!!
 Do you want to
 overwrite it?

(ENT/ESC)

FLT update 确认

<BACK UP GFT FILE>
 Backup Success!!

Press Any Key

- float file 备份正常结束后，输出 Backup Success.

5) PARA

Step 1.

DATA 选择

<INFO - BACKUP>
 1. F/W **2. DATA**
 3. XML 4. MODEL
 5. LOG
 PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

PARA 选择

<BACKUP - DATA>
 1. JOB 2. PNT
 3. INT 4. FLT
5. PARA
 0. ALL
 PATH: /TR-1.2.3
 input : #



- 选择 “5. PARA”

PARA backup

<BACK UP PARAMETER>
 Do you want to
 backup the
 parameter files?

(ENT/ESC)



- 输出是否备份parameter file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面

<BACK UP PARAMETER>
 [CAUTION]
 SY0.PAR
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

PARA backup 确认

<BACK UP PARAMETER>
 Backup Success!!

Press Any Key

- parameter file备份正常结束后, 输出 Backup Success.

6) ALL

Step 1.

DATA 选择

⟨INFO - BACKUP⟩
 1. F/W **2. DATA**
 3. XML 4. MODEL
 5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “2. DATA”

ALL 选择

⟨BACKUP - DATA⟩
 1. JOB 2. PNT
 3. INT 4. FLT
 5. PARA
0. ALL

PATH: /TR-1.2.3
 input : #



- 选择 “0. ALL”

ALL backup

⟨BACK UP DATA⟩
 Do you want to
 backup the all
 data files?

(ENT/ESC)



- 输出是否备份所有 file 的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面

ALL backup 确认

⟨BACK UP DATA⟩
 Backup Success!!

Press Any Key

- 所有 file 备份正常结束后, 输出 Backup Success.

7.4.3.2.3 XML

- 备份XML 文件.

Step 1.

XML 选择

<INFO - BACKUP>
 1. F/W 2. DATA
3. XML 4. MODEL
 5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “3. XML”

XML backup

<BACK UP XML FILE>
 Do you want to
 backup the xml
 file?

(ENT/ESC)



- 输出是否备份xml file的确认信息, 需要备份时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与备份的文件相同名称的文件时, 弹出警告窗口.

<BACK UP XML FILE>
 [CAUTION]
 2^2^7%1^1^ .xml
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP:覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

XML backup 确认

<BACK UP XML FILE>
 Copying Success!!

Press Any Key

- xml file 备份正常结束后, 输出 Copying Success.

7.4.3.2.4 MODEL

- 备份MODEL 文件.

Step 1.

MODEL 选择

<INFO - BACKUP>
 1. F/W 2. DATA
 3. XML **4. MODEL**
 5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “4. MODEL”

MODEL backup

<BACK UP MODEL FILE>
 Do you want to
 upload the model
 file?

(ENT/ESC)



- 输出是否备份model file的确认信息, 需要上传时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与要备份的文件相同名的文件时, 弹出警告窗口.

<BACK UP MODEL FILE>
 [CAUTION]
 UAF4B1L100Q-240-300.
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP:覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

MODEL backup 确认

<BACK UP MODEL FILE>
 Copying Success!!

Press Any Key

- model file 备份 正常结束后, 输出 Copying Success.

7.4.3.2.5 LOG

- 备份LOG. 详细项目如下表.

■ 详细项目

GROUP		内容	备注
5. LOG	1. JOB	备份WLOG制作的Log(JOB Log).	
	2. COMM	备份CLOG制作的Log (通信Log).	

1) JOB

Step 1.

LOG 选择

<INFO - BACKUP>
 1. F/W 2. DATA
 3. XML 4. MODEL
5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “5. LOG”

JOB LOG 选择

<BACKUP - LOG>
1. JOB 2. COMM

PATH: /TR-1.2.3
 input : #



- 选择 “1. JOB”

JOB LOG backup

<BACK UP J-LOG FILE>
 Do you want to
 backup the J-log
 files?

(ENT/ESC)

- 输出是否备份job log file的确认信息, 需要备份时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与备份的文件相同名称的文件时, 弹出警告窗口.



<BACK UP J-LOG FILE>
 [CAUTION]
 160220_160440.log
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP: 覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

JOB LOG backup 确认

<BACK UP J-LOG FILE>
 Copying Success!!

Press Any Key

- JOB LOG 备份正常结束后, 输出 Copying Success.

2) COMM

Step 1.

LOG 选择

<INFO - BACKUP>
 1. F/W 2. DATA
 3. XML 4. MODEL
5. LOG

PATH: /TR-1.2.3
 input : #



- 选择 “5. LOG”

COMM LOG 选择

<BACKUP - LOG>
 1. JOB 2. COMM

PATH: /TR-1.2.3
 input : #



- 选择 “2. COMM”

COMM LOG backup

<BACK UP C-LOG FILE>
 Do you want to
 backup the C-log
 files?

(ENT/ESC)

- 输出是否备份通信Log file的确认信息, 需要备份时按下 “ENTER” 键
- 按下 “ESC” 键恢复到之前画面.
- 存在与备份的文件相同名称的文件时, 弹出警告窗口.



<BACK UP C-LOG FILE>
 [CAUTION]
 comm_170221_00.log
 already exists!!
 Do you want to
 overwrite it?

ONE ALL SKIP STOP

ONE: 允许覆盖相应文件. 有重复的文件是输出警告窗口.

ALL: 允许覆盖所有重复文件. 不会再询问是否覆盖重复文件.

SKIP: 覆盖相应文件, 复制下一个文件.

STOP: 停止复制.

COMM LOG backup 确认

<BACK UP J-LOG FILE>
 Copying Success!!

Press Any Key

- 通信 LOG 备份正常结束后, 输出 Copying Success.

第8章 MOVEMENT模式

- 为了确认Robot位置是否正确设置在用户所需的位置，给特定变量值（例：I20~I26）分配单位Motion进行用户选择的JOB程序（例子：MANUL.JOB）的阶段。此过程的难度在于需要完美理解相应单位Motion中用数字标记的特定变量值。因此，为解决这种不变，在MOVEMENT 参数中设置JOB 程序和画面类型后，使用主菜单的 MOVEMENT 功能。
- MOVEMENT 功能为用户输入单位Motion对应的命令和坐标Point值或STAGE, SIZE 等时，自动设置符合特定变量值（例：I20~I26）变量的功能。

8.1 概要

- ✓ MOVEMENT 模式根据参数设定提供两种画面。根据使用用途设置适合的类型。

■ MOVEMENT 画面种类

- CMD: Robot执行的命令
- GP: Robot移动的位置Point
- SLOT: Slot编号
- ARM: 选择上下Hand
- VEL: Robot速度
- STAGE, SIZE: STAGE 编号及尺寸
- SLOT: Slot编号
- MULTI: 设置磁带类型

TYPE 1

```

(MOVEMENT)
CMD  :HOME
GP   :100
SLOT :1
ARM  :DN
VEL  :5
EXIT  START
  
```

TYPE 2

```

(MOVEMENT)
CMD  :HOME
STAGE :1      SIZE :1
SLOT :1      ARM  :DN
MULTI :CASSETTE
VEL   :5
[GP:100]
EXIT  START
  
```

■ MOVEMENT 命令说明

COMMAND	INDEX	说明
HOME	1	HOME
AFOLD	2	ARM FOLD
ADSOL	3	吸附
DESOL	4	破坏吸附
PUT_RD	5	收纳前等待位置移动(put ready)
GET_RD	6	提取前等待位置移动(get ready)
PUT	7	收纳动作
GET	8	提取动作
PUT_TE	12	put Teaching位置移动
GET_TE	13	get Teaching位置移动
GET_RET	16	提取 RETRACT 移动
GET_EXT	17	提取 EXTEND 移动
PUT_RET	18	收纳 RETRACT 移动
PUT_EXT	19	收纳 EXTEND 移动
AFOLD_N	26	无Work确认 ARM FOLD 移动
AFOLD_C	27	确认Work后 ARM FOLD 移动
MOVE_AP	41	绝对位置移动
MOVE_RP	42	相对位置移动
MOVE_TE	43	Teaching位置移动
MAP	50	Mapping动作
MAP_SET	51	Mapping位置设定
CHK_TE	300	Teaching位置确认

8.2 TYPE 1 模式使用方法

- ✓ 输入CMD, Slot编号, ARM TYPE, 进行速度. 可以修改Point编号.

Step 1.

选择项目

```
<MOVEMENT>
CMD  : HOME ■
GP   : 100
SLOT : 1
ARM  : DN
VEL  : 5
```

EXIT START

```
<MOVEMENT>
CMD  : HOME
GP   : 100
SLOT : 1
ARM  : DN ■
VEL  : 5
```

EXIT START



- 按下上下方向键选择要编辑的项目

Step 2.

CMD 输入

```
<MOVEMENT>
CMD  : HOME ■
GP   : 100
SLOT : 1
ARM  : DN
VEL  : 5
```

EXIT START



- 按下ENTER 键激活CMD 列表设定窗口

页面移动

```
<SELECT CMD> 2/2
1. PUT_RET      2. PUT_EXT
3. AFOLD_N      4. AFOLD_C
5. MOVE_AP      6. MOVE_RP
7. MOVE_TE      8. MAP
9. MAP_SET      A. CHK_TE
```

PREV NEXT

EXIT



- 按下[F1] PREV 键移动到上一个页面



- 按下[F2] NEXT键移动到下一个页面

选择命令

```
<SELECT CMD> 1/2
1. HOME        2. AFOLD
3. ADSOL       4. DESOL
5. PUT_RD      6. GET_RD
7. PUT         8. GET
9. PUT_TE      A. GET_TE
B. GET_RET     C. GET_EXT
PREV NEXT      EXIT
```



- 输入所需命令相应的键选择命令



- 恢复到之前画面



取消输入

(SELECT CMD) 1/2
 1. HOME 2. AFOLD
 3. ADSOL 4. DESOL
 5. PUT_RD 6. GET_RD
 7. PUT 8. GET
 9. PUT_TE A. GET_TE
 B. GET_RET C. GET_EXT
 PREV NEXT **EXIT**



- 按下[F4] EXIT 键恢复到之前画面

Step 3.

GP, SLOT, VEL 编辑

(MOVEMENT)
 CMD : PUT_RET
 GP : ■
 SLOT : 1
 ARM : DN
 VEL : 5
 EXIT START



- 移动到需要编辑的项目
- 按下"DEL" 键删除Point信息

(MOVEMENT)
 CMD : PUT_RET
 GP : 110 ■
 SLOT : 1
 ARM : DN
 VEL : 5
 EXIT START



- 输入所需值



错误

(MOVEMENT)
 CMD : PUT_RET
 GP : **77777**
 SLOT : 1
 ARM : DN
 VEL : 5

GP is not valid!!

- 输入超出范围的值时发生Error信息

Step 4.

ARM 编辑

(MOVEMENT)
 CMD : PUT_RET
 GP : 105
 SLOT : 1
 ARM : DN ■
 VEL : 5
 EXIT START



- 按下"ENTER" 键选择ARM TYPE

Step 5.

执行

```

<MOVEMENT>
CMD  : HOME
GP   : 100
SLOT : 1
ARM  : DN
VEL  : 5

```

EXIT **START**• 按下[F2] **START** 键执行

```

<MOVEMENT>
CMD : HOME
GP : 100          V: 5

T : 214    Z : 20.39
L : -135   R : -135
X : 0      S : 0
                STOP

```

执行中停止

```

<MOVEMENT>
CMD : HOME
GP : 100          V: 5

T : 214    Z : 20.39
L : -135   R : -135
X : 0      S : 0
                STOP

```

STOP• 按下[F4] **STOP** 键停止

Step 6.

结束

```

<MOVEMENT>
CMD  : HOME
GP   : 100
SLOT : 1
ARM  : DN
VEL  : 5

```

EXIT START• 按下[F1] **EXIT** 或 "ESC" 键结束功能

8.3 TYPE 2 模式使用方法

- ✓ 设置CMD, STAGE, SIZE, Slot编号, ARM 类型, Multi类型, 速度. 使用固定 GP Point.

Step 1.

项目选择

```

<MOVEMENT>
CMD : HOME
STAGE : 1      SIZE : 1
SLOT : 1      ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 105]
EXIT START
  
```

```

<MOVEMENT>
CMD : HOME
STAGE : 1      SIZE : 1
SLOT : 1      ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 105]
EXIT START
  
```



- 按下方向键选择要编辑的项目

Step 2.

输入CMD

```

<MOVEMENT>
CMD : HOME
STAGE : 1      SIZE : 1
SLOT : 1      ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 105]
EXIT START
  
```



- 按下ENTER 键激活 CMD列表设定窗口

页面移动

```

<SELECT CMD>
1. PUT_RET      2. PUT_EXT
3. AFOLD_N      4. AFOLD_C
5. MOVE_AP      6. MOVE_RP
7. MOVE_TE      8. MAP
9. MAP_SET      A. CHK_TE
  
```

PREV NEXT EXIT



- 按下[F1] PREV 键移动到上一个页面
- 按下[F2] NEXT键移动到下一个页面

选择命令

```

<SELECT CMD>
1. HOME      2. AFOLD
3. ADSOL     4. DESOL
5. PUT_RD    6. GET_RD
7. PUT       8. GET
9. PUT_TE    A. GET_TE
B. GET_RET   C. GET_EXT
PREV NEXT    EXIT
  
```



- 输入所需命令相应的键选择命令
- 恢复到之前画面

取消输入

```

<SELECT CMD> 1/2
1. HOME      2. AFOLD
3. ADSOL     4. DESOL
5. PUT_RD    6. GET_RD
7. PUT       8. GET
9. PUT_TE    A. GET_TE
B. GET_RET   C. GET_EXT
PREV NEXT   EXIT
  
```



- 按下[F4] EXIT 键恢复到之前画面

Step 3.

STAGE, SIZE, SLOT, VEL 编辑

```

<MOVEMENT>
CMD : HOME
STAGE : 1 ■ SIZE : 1
SLOT : 1 ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 100]
EXIT START
  
```



- 移动到需要编辑的项目
- 按下"DEL" 键删除Point信息

```

<MOVEMENT>
CMD : HOME
STAGE : 10 ■ SIZE : 1
SLOT : 1 ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 100]
EXIT START
  
```



- 输入所需值



错误

```

<MOVEMENT>
CMD : HOME
STAGE : 100 SIZE : 1
SLOT : 1 ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 100]
STAGE is not valid!!
  
```

- 输入超出范围的值时发生Error信息

Step 4.

ARM, MULTI 编辑

```

<MOVEMENT>
CMD : HOME
STAGE : 10 SIZE : 1
SLOT : 1 ARM : DN
MULTI : CASSETTE ■
VEL : 5
[GP: 100]
EXIT START
  
```



- 按下"ENTER" 键选择

Step 5.

执行

```

<MOVEMENT>
CMD : HOME
STAGE : 1      SIZE : 1
SLOT : 1      ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 100]
EXIT START

```

F2

• 按下[F2] **START** 键执行

```

<MOVEMENT>
CMD : HOME
GP : 100      V: 5

T : 214      Z : 20.39
L : -135     R : -135
X : 0        S : 0
                STOP

```

执行中停止

```

<MOVEMENT>
CMD : HOME
GP : 100      V: 5

T : 214      Z : 20.39
L : -135     R : -135
X : 0        S : 0
                STOP

```

F4

• 按下[F4] **STOP** 键停止

Step 6.

结束

```

<MOVEMENT>
CMD : HOME
STAGE : 1      SIZE : 1
SLOT : 1      ARM : DN
MULTI : CASSETTE
VEL : 5
[GP: 100]
EXIT  START

```

F1

• 按下[F1] **EXIT** 或 "**ESC**" 键结束功能

ESC

第9章 SHUTDOWN 模式

- 非正常结束控制器时，可能导致CF损坏，因此请务必正常结束后关闭控制器电源开关。
电源结束后控制器正面部的Segment灭灯。此时，关闭控制器电源开关即可。
- 对于安装电池的控制器，即使发生电源切断等非正常结束状况，也会最终保存修改的数据并正常结束。但，重新供应电源时，请务必保证正面部Segment灭灯后，再供应电源。

9.1 概要

- ✓ SHUTDOWN 模式提供控制器正常结束、Driver Alarm Reset功能。

SHUTDOWN MODE
1. SYSTEM SHUTDOWN

■ 0. SHUTDOWN种类

	GROUP	内容	备注
0. SHUTDOWN	1. SYSTEM SHUTDOWN	控制器正常结束	

9.2 SHUTDOWN 模式流程图

- ✓ SHUTDOWN大体由两个Group组成.
 - SYSTEM SHUTDOWN: 正常结束控制器.

■ SHUT DOWN Manu Tree

0. SHUTDOWN → 1. SYSTEM SHUTDOWN

9.3 SHUTDOWN Group 画面 OPEN 顺序

Step 1.

初始菜单画面

Robostar
NewRo RCM6-Series
Version :3.01.33
(TR 141009)

Press ENTER Key

<TEACH MENU>
1. EDIT 2. RUN
3. INFO 4. I/O
5. ORIGIN

0. SHUTDOWN
input : #



- 接入电源后LCD中输出的画面(发生错误时请按照Error code进行处理)

- 按下ENTER 键移动到初始菜单画面

Step 2.

输入PASSWORD

<TEACH MENU>
1. EDIT 2. RUN
3. INFO 4. I/O
5. ORIGIN

0. SHUTDOWN
Password=



~



- 输入Password (初始值: "1111")
- 按下"ESC" 键取消输入 Password

<TEACH MENU>
1. EDIT 2. RUN
3. INFO 4. I/O
5. ORIGIN

0. SHUTDOWN
Invalid!!

- 输入错误的Password 时显示"Invalid"

控制器正常结束

SHUTDOWN MODE
1. SYSTEM SHUTDOWN

input : #



- 控制器正常结束

9.4 SYSTEM SHUTDOWN

- ✓ 正常结束控制器.
- ✓ 预防异常结束控制器时对CF产生的损伤.

■ OPEN顺序

- Top Screen → 0. SHUTDOWN → Password → 1. SYSTEM SHUTDOWN

Step 1.

控制器SHUTDOWN

SHUTDOWN MODE
1. SYSTEM SHUTDOWN

input : #



- 正常结束控制器时按下”1”键

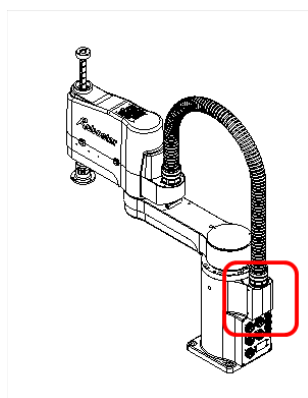
- 结束画面

第10章 附录

10.1 Robot 标签位置

- ✓ Robot 标签位于两个位置.
 - 1) 器具部标签: 粘贴在Robot 背面连接器商标.
 - 2) 控制器标签: 粘贴在控制器上方商标.
- ✓ Ex) SCARA Robot时

- 器具部标签(商标)



标签

说明

New Ro AC ROBOT SCARA ROBOT		
MODEL : RBS60-4A		
LENGTH : A = 300, B = 300		
STROKE : Z = 200		
CONTROLLER : RCQ-4211		
Ser. NO : 2103 - 007		

- MODEL: 请参考器具部类型、使用说明书类型说明
- LENGTH, STROKE:
 - A轴: 300mm, B轴 300mm, Z 轴: 200mm
- CONTROLLER: 控制器类型 ,
 - A(X): 400W, B(Y): 200W, Z: 100W, W: 100W
- 制造编号: 2001年 3月 7号产品

Axis	Motor	Encoder
A	MS/400W	Abs.131072
B	MS/200W	Abs.131072
Z	MS/100W	Abs.131072
W	MS/100W	Abs.131072

- Axis: 显示各轴
- Motor: 各轴电机容量
 - MS/400W -> Minas Motor 400W
- Encoder: 各轴电机编码器类型, 脉冲值
 - Abs.131072: Absolute Encoder, 131072 (脉冲)

Axis	R	Length	offset
A	1/50	299.92	3.29
B	1/50	299.95	3.82
Z	1/15	200	
W	1/20	360	

- Axis: 显示各轴
- R: 各轴减速比
- Length: 各轴长度 (mm)
- offset: A, B 轴 offset 值

- 控制器标签 (商标) - 粘贴在控制器上方

Robostar
www.robostar.co.kr

경기도 안산시 상록구 사사동 119-38
(426-220)
㈜ 로보스타

119-38, Sasa-dong, Sangnok-gu,
Ansan-City, Gyeonggi-do, Republic of
South Korea (426-220)
Robostar.Co.,Ltd

제어기 사양 CONTROLLER SPECIFICATION		IPM 용량 IPM CAPACITY	주의사항 CAUTION
로봇 타입 ROBOT TYPE			경고 DANGER
모델명 MODEL NO.			고압에 주의하십시오. 제어가 점검 전 반드시 전원을 꺼주십시오. CAUTION! HIGH VOLTAGE! DISCONNECT POWER BEFORE SERVICING.
시리얼 넘버 SERIAL NO.			위험 WARNING
제조 일자 DATE			감전의 위험이 있으니, 제어가 점검 전에는 반드시 전원 스위치를 끄고, 전원 케이블을 분리한 후, 3분 이상 대기 후 점검을 하십시오. DON'T TOUCH INSIDE OF CONTROLLER BEFORE TURNING OFF POWER SWITCH AND DISCONNECTING POWER CABLE. WAITING 3 MINUTES OR MORE OR YOU CAN GET ELECTRIC SHOCK.
용량 / 출력 CAPACITY / OUTPUT	5kVA / 5kW		주의 CAUTION
입력 전원 INPUT POWER	MAIN 1Phase AC 220V(±10%, 50-60Hz) CONTROL 1Phase AC 220V(±10%, 50-60Hz)		부상의 위험이 있으니, 제어가 내부로 손이나 이물질을 넣지 마십시오. DON'T PUT FINGERS OR FOREIGN OBJECTS INTO OPENING. OR PERSONAL INJURY MAY OCCUR.
크기 SIZE	620 x 480 x 173 (W x D x H, mm)		감전 및 오동작의 원인이 될 수 있으니, 제어가 외부 접지 단자에 접지를 연결하십시오. GROUND THE EARTH TERMINAL OF THE CONTROLLER WITHOUT FAIL. OR YOU CAN GET ELECTRIC SHOCK.
중량 WEIGHT	25Kg		
사용 환경 CONDITION	0-40°C, 20-80% RH		
소프트웨어 버전 SOFTWARE Ver		옵션 카드 OPTION CARD	
MAIN		DIGITAL I/O	
SERVO		CC-LINK	
FRONT		PROFIBUS	
I/O		ETHERNET	
T/P		DEVICENET	

10.2 TPI-9000 membrane

F1	F2	F3	F4	ESC
STOP	JUMP	PREV	NEXT	DEL RST
DISP A	SPACE B	Pg Up	▲	ENTER
VEL C	INT D	◀	▼	▶
SVON E	BRK F	Pg Dn	← J1 →	
7 G	8 H	9 I	← J2 →	→ J2
4 L	5 M	6 N	← J3 →	→ J3
1 Q	2 R	3 S	← J4 →	→ J4
0 V	● W	— X	← J5 →	→ J5
SHIFT	AXIS CHG	ALARM LIST	← J6 →	

10.3 各通信类型控制器列表

通信类型	控制器
EtherCAT	RcT1 Series
RS422	TX1 Series

第11章 修正

日期	修正内容
2015年 10月	初始版本
2015年 12月	3. JOG → 3. TEACH 变更
2016年 7月	- 第二章 EDIT 2.1.4.2.1 预处理器指令项变更事项修正 2.3 参数中 2.3.2.5 SERVO, 2.3.2.6 ORIGIN, 2.3.3.1 ENV 修正事项变更 - 第五章 I/O 5.1.7 DUM_IN 追加, 5.4 LATCH 追加 - 第六章 Origin - 内容追加
2017年 3月	- 第二章 参数(PARAMETER) 2.3.2.3.6 CUM_DIST 追加, 2.3.2.5.8 BELT 追加, 2.3.2.8 ARM 追加, 2.1.3.1 反映变更事项 - 第三章 RUN 3.5.3.17 INSHAPE 追加 - 第四章 TEACH 4.6.5 AFLD, 4.6.6 JSYN, 4.6.7 ZCAL 追加 4.7.5 AFLD, 4.7.6 JSYN, 4.7.7 ZCAL 追加 - 第七章 INFO 7.4 USB 追加
2017年 4月	- 第八章 SHUTDOWN 模式 变更为第九章 - 第八章 MOVEMENT 功能 追加

T1 SERIES: T1 CONTROLLER

OPERATING MANUAL

FIRST EDITION SEPTEMBER 2015

ROBOSTAR CO, LTD
ROBOT R&D CENTER