

客服热线 (021) 5863-9595

绵密网络 专业服务

中达电通已建立了 41 个分支机构及服务网点，并塑建训练有素的专业团队，提供客户最满意的服务，公司技术人员能在 2 小时内回应您的问题，并在 48 小时内提供所需服务。

上海
电话:(021)6301-2827
传真:(021)6301-2307

南昌
电话:(0791)6255-010
传真:(0791)6255-102

合肥
电话:(0551)2816-777
传真:(0551)2816-555

南京
电话:(025)8334-6585
传真:(025)8334-6554

杭州
电话:(0571)8882-0610
传真:(0571)8882-0603

武汉
电话:(027)8544-8265
传真:(027)8544-9500

长沙
电话:(0731)8827-7881
传真:(0731)8827-7882

南宁
电话:(0771)5879-599
传真:(0771)2621-502

厦门
电话:(0592)5313-601
传真:(0592)5313-628

广州
电话:(020)3879-2175
传真:(020)3879-2178

济南
电话:(0531)8690-7277
传真:(0531)8690-7099

郑州
电话:(0371)6384-2772
传真:(0371)6384-2656

北京
电话:(010)8225-3225
传真:(010)8225-2308

天津
电话:(022)2301-5082
传真:(022)2335-5006

太原
电话:(0351)4039-475
传真:(0351)4039-047

乌鲁木齐
电话:(0991)6118-160
传真:(0991)6118-289

西安
电话:(029)8836-0640
传真:(029)88360640-8000

成都
电话:(028)8434-2075
传真:(028)8434-2073

重庆
电话:(023)8806-0306
传真:(023)8806-0776

哈尔滨
电话:(0451)5366-0643
传真:(0451)5366-0248

沈阳
电话:(024)2334-1612
传真:(024)2334-1163

长春
电话:(0431)8892-5060
传真:(0431)8892-5065

AH500 运动控制模块手册



AH500 运动控制模块手册



中达电通股份有限公司

地址：上海市浦东新区民夏路238号

邮编：201209

电话：(021)5863-5678

传真：(021)5863-0003

网址：<http://www.deltagreentech.com.cn>

AH-0101610-03
2015-10-31

中达电通公司版权所有
如有改动，恕不另行通知

www.delta.com.tw/ia



AH500 运动控制模块手册

目录

第 1 章 AH500 运动控制模块架构	
1.1 硬件 AH500 运动控制模块功能架构	1-2
1.2 软件 O100 主程序架构	1-5
1.2.1 O100 主程序手动功能运动方式	1-6
1.3 OX 运动子程序架构	1-7
1.4 Pn 子程序架构	1-9
1.5 O100、OX、Pn 整体程序区块规划之架构	1-11
1.5.1 程序架构介绍	1-11
第 2 章 硬件规格及配线方式	
2.1 硬件规格	2-2
2.1.1 一般规格	2-2
2.1.2 输入点电气规格	2-3
2.1.3 输出点电气规格	2-9
2.1.4 外观尺寸	2-13
2.1.5 模块部位介绍	2-15
2.2 配线方式	2-19
2.2.1 I/O 连接线及配线模块	2-19
2.2.2 输入接点配线	2-25
2.2.3 输出接点配线	2-28
2.2.4 与下位驱动器之配线	2-30
2.3 通讯端口	2-36
第 3 章 各种装置功能	
3.1 模块各装置编号一览表	3-2
3.2 数值、常数[K]/[H]、浮点数[F]	3-3
3.3 外部输入[X]/输出[Y]接点的编号及功能	3-5
3.4 辅助继电器[M]的编号及功能	3-7
3.5 特殊辅助继电器[SM]的编号及功能	3-7
3.6 内部继电器[S]的编号及功能	3-7

3.7	定时器[T]的编号及功能	3-8
3.8	计数器[C]的编号及功能	3-9
3.9	寄存器[D][V][Z]的编号及功能	3-14
3.9.1	数据寄存器[D]	3-14
3.9.2	变址用寄存器[V][Z]	3-14
3.10	特殊寄存器[SR]的编号及功能	3-15
3.11	指针[P]的编号及功能	3-15
3.12	特殊继电器[SM]及特殊寄存器[SR]	3-16
3.12.1	特殊继电器[SM]	3-16
3.12.2	特殊寄存器[SR]	3-20
3.13	特殊继电器[SM]及特殊寄存器[SR]功能说明	3-26
3.14	运动模式特殊寄存器[SR]	3-39
第 4 章	基本顺序指令	
4.1	基本指令一览表	4-2
4.2	基本指令说明	4-4
第 5 章	应用指令分类及基本使用	
5.1	应用指令一览表	5-4
5.2	应用指令的组成	5-8
5.3	应用指令对数值的处理方式	5-12
5.4	使用变址寄存器 V、Z 来修饰操作数	5-15
5.5	指令索引	5-16
5.6	应用指令说明	5-20
5.7	运动功能块一览表	5-129
5.8	运动功能块引脚介绍	5-131
5.8.1	输入/输出引脚功能定义	5-131
5.8.2	输入/输出引脚时序图	5-132
5.8.3	PMSoft 软件操作介绍	5-133
5.9	台达定义参数一览表	5-136
5.10	单轴运动功能块说明	5-139
5.10.1	绝对单段速运动	5-139
5.10.2	相对单段速运动	5-142

5.10.3	绝对两段速运动	5-145
5.10.4	相对两段速运动	5-148
5.10.5	插入单段速运动	5-150
5.10.6	插入两段速运动	5-154
5.10.7	寸动运动	5-156
5.10.8	手摇轮模式运动	5-159
5.10.9	原点回归运动	5-160
5.10.10	单轴运动停止	5-162
5.10.11	运动轴参数设定 I	5-164
5.10.12	运动轴参数设定 II	5-165
5.10.13	读取当前位置/速度	5-167
5.10.14	轴状态信息	5-168
5.10.15	设定当前位置	5-170
5.10.16	输入极性设定	5-171
5.10.17	电子齿轮运动	5-174
5.10.18	凸轮啮合	5-176
5.10.19	凸轮数据点读取	5-178
5.10.20	凸轮数据点写入	5-179
5.10.21	凸轮同步速率计算	5-181
5.10.22	凸轮曲线规划	5-182
5.10.23	凸轮曲线更新	5-184
5.11	多轴运动功能块说明	5-186
5.11.1	G 码参数设定	5-186
5.11.2	G 码运行	5-187
5.11.3	G 码停止	5-189
5.11.4	M 码读取	5-191
5.11.5	多轴绝对直线插补	5-193
5.11.6	多轴相对直线插补	5-194
5.11.7	多轴直线插补停止	5-196
5.12	网络功能块说明	5-198
5.12.1	伺服启动/停止	5-198
5.12.2	伺服重置	5-199
5.12.3	伺服参数写入	5-200

5.12.4	伺服参数读取	5-202
5.12.5	伺服回原点命令	5-204
5.12.6	伺服 DMCNET 通讯初始化	5-206
5.12.7	伺服捕捉功能	5-207
5.12.8	Ethernet IP Address 设定	5-209
5.13	其它功能块说明	5-210
5.13.1	SD 主程序备份	5-210
5.13.2	SD 装置备份	5-211
5.13.3	SD 装置回存	5-212
5.13.4	高速计数器	5-213
5.13.5	高速定时器	5-214
5.13.6	高速比较设定	5-216
5.13.7	高速比较重置	5-218
5.13.8	高速捕捉设定	5-219
5.13.9	高速捕捉遮蔽	5-222
5.13.10	中断设定	5-223
5.13.11	绝对型编码器	5-224
第 6 章 模块数据传递		
6.1	功能简介	6-2
6.2	参数介绍	6-2
6.3	功能使用	6-5
第 7 章 单轴运动功能		
7.1	单轴运动功能	7-2
7.2	单轴运动功能简介	7-13
7.3	JOG 模式使用介绍	7-14
7.3.1	专属对应特殊寄存器	7-14
7.3.2	运动操作说明	7-15
7.4	变速度模式使用介绍	7-16
7.4.1	专属对应特殊寄存器	7-16
7.4.2	运动操作说明	7-17
7.5	手摇轮操作模式使用介绍	7-18

7.5.1	专属对应的特殊寄存器	7-18
7.5.2	运动操作说明	7-19
7.6	原点回归模式使用介绍	7-20
7.6.1	专属对应的特殊寄存器	7-20
7.6.2	运动操作说明	7-22
7.7	单段速定位模式使用介绍	7-26
7.7.1	专属对应的特殊寄存器	7-26
7.7.2	运动操作说明	7-27
7.8	插入单段速定位模式使用介绍	7-28
7.8.1	专属对应的特殊寄存器	7-28
7.8.2	运动操作说明	7-29
7.9	连续两段速定位模式使用介绍	7-30
7.9.1	专属对应的特殊寄存器	7-30
7.9.2	运动操作说明	7-32
7.10	插入两段速定位使用介绍	7-33
7.10.1	专属对应的特殊寄存器	7-33
7.10.2	运动操作说明	7-34
7.11	状态标志与寄存器	7-35
第 8 章 电子凸轮功能		
8.1	电子凸轮 (E-CAM) 简介	8-2
8.2	电子凸轮的实现方式	8-3
8.2.1	初始设定	8-3
8.2.1.1	建立电子凸轮数据	8-3
8.2.1.2	输入/出脉冲形式设定	8-4
8.2.2	主从轴设定及操作	8-5
8.2.2.1	主轴来源设定	8-6
8.2.2.2	主轴启动角度设定	8-6
8.2.2.3	从轴输出设定	8-7
8.2.3	周期式电子凸轮启动/停止	8-8
8.2.3.1	周期式电子凸轮启动	8-8
8.2.3.2	周期式电子凸轮停止	8-9
8.3	电子凸轮数据 (E-CAM Data) 建立	8-11

8.3.1	PMSoft CAM Chart 绘制电子凸轮数据.....	8-11
8.3.1.1	根据主从轴的标准函数关系绘制	8-11
8.3.1.2	主从轴根据实际工作的点对点关系绘制	8-15
8.3.1.3	单笔电子凸轮数据 (E-CAM Data) 建立/变更.....	8-17
8.4	电子凸轮应用-飞剪应用	8-19
8.4.1	飞剪电子凸轮数据建立.....	8-21
8.4.2	功能块-凸轮飞剪曲线生成操作	8-21
8.4.3	功能块-凸轮动态变更操作.....	8-26
8.4.4	飞剪电子凸轮范例	8-26
 第 9 章 多轴插补运动		
9.1	多轴插补功能简介.....	9-2
9.2	O 指针/M 码指令及 G 码指令一览表	9-2
9.3	G 码指令的组成.....	9-3
9.4	G 码指令说明	9-6
9.5	O 指针/M 码指令	9-23
9.6	TO 指令使用说明.....	9-26
 第 10 章 高速计数与计时功能		
10.1	高速计数器.....	10-2
10.2	高速定时器.....	10-6
 第 11 章 高速捕捉与比较功能		
11.1	高速比较与捕捉功能使用格式.....	11-2
11.2	比较功能	11-2
11.3	清除输出功能.....	11-9
11.4	捕捉功能	11-10
11.5	屏蔽功能	11-15
 第 12 章 以太网网络设置		
12.1	功能简介	12-2
12.2	功能规格	12-2
12.3	参数功能介绍.....	12-2
12.4	PMSoft 通讯功能说明.....	12-3

12.5	MODBUS 通讯功能说明	12-7
12.6	以太网网络错误说明及排除方式	12-9
第 13 章 扩展保存装置		
13.1	功能简介	13-2
13.2	参数功能介绍	13-2
13.3	G 码读取与执行	13-4
13.4	装置备份及回存	13-5
13.4.1	备份	13-5
13.4.2	回存	13-8
13.5	程序备份及回存	13-9
13.5.1	备份	13-9
13.5.2	回存	13-9
13.6	韧体更新	13-10
第 14 章 DMCNET		
14.1	功能简介	14-2
14.2	功能规格	14-2
14.3	参数功能	14-3
14.4	DMCNET 联机说明	14-8
14.5	伺服参数读写说明	14-11
14.6	DMCNET 运动控制说明	14-13
14.7	DMCNET 错误说明及排除方式	14-18
14.7.1	增量型伺服连线范例	14-18
14.7.2	绝对型伺服连线范例	14-19
14.8	DMCNET 错误说明及排除方式	14-20
第 15 章 PMSoft 软件-USB 通讯设置		
15.1	功能简介	15-2
15.2	功能规格	15-2
15.3	PMSoft 通讯功能说明	15-2
附录 A 错误代码原因对照表		
A.1	错误代码原因对照表	A-2

MEMO



第1章 AH500 运动控制模块架构

目录

1.1	硬件 AH500 运动控制模块功能架构	1-2
1.2	软件 O100 主程序架构	1-5
1.2.1	O100 主程序手动功能运动方式	1-6
1.3	OX 运动子程序架构	1-7
1.4	Pn 子程序架构	1-9
1.5	O100、OX、Pn 整体程序区块规划之架构	1-11
1.5.1	程序架构介绍	1-11

1

台达 AH20MC-5A、AH15PM-5A、AH10PM-5A、AH05PM-5A 模块为高速定位、双轴线性及圆弧插补多功能可编程器，其特色是结合了基本指令、应用指令、运动指令及 G 码指令等功能，让程序编写设计上更多元化。

本手册以 AH500 运动控制模块的来统称 AH20MC-5A、AH15PM-5A、AH10PM-5A 及 AH05PM-5A 此三种模块。

本章节主要介绍 AH20MC-5A、AH15PM-5A、AH10PM-5A 及 AH05PM-5A 模块的基本架构，由于 AH500 运动控制模块结合了 PLC 顺序控制及多轴插补定位控制的功能，因此在程序的规划上，主要分为 O100 主程序、OX 运动子程序及 Pn 子程序等三大类，本章节将以此三大类的程序架构进行说明。基本指令、应用指令、运动指令及 G 码指令将于其它章节做详细介绍说明。

1.1 硬件AH500 运动控制模块功能架构

项目	规格			
	AH20MC-5A	AH10PM-5A	AH15PM-5A	AH05PM-5A
支持实体运动轴数	12 轴 (第 1~12 轴)	6 轴 (第 1~6 轴)	4 轴 (第 1~4 轴)	2 轴 (第 1~2 轴)
支持虚拟运动轴数	4 轴 (第 13~16 轴)	10 轴 (第 7~16 轴)	12 轴 (第 5~16 轴)	14 轴 (第 3~16 轴)
程序储存	内藏 64k Steps 储存器			
单位系	电机、机械、复合			
与系统主机串接说明	利用 HWCONFIG 设定数据传递区域，可设定寄存器传递的启示位置以及传递的数据空间长度。读写最大可分别为 400 个 D/M 寄存器。			
电机控制方式	台达伺服高速通讯 DMCNET 响应同步周期 1ms	三种脉冲输出模式：Pulse/Dir·FP (CW)/RP (CCW)·A/B；		
		第 1~4 采用差动输出， 第 5~6 轴采用集电极输出	均采用差动输出	
最快速度值	单轴最快 1M PPS 多轴插补轴最快 1M PPS			
输入信号	操作开关	STOP/RUN (停止/启动选择开关)		无

项目	规格			
	AH20MC-5A	AH10PM-5A	AH15PM-5A	AH05PM-5A
输入端子	X0.0+、X0.0-、 X0.1+、X0.1-、 X0.2+、X0.2-、 X0.3+、X0.3-、 X0.8+、X0.8-、 X0.9+、X0.9-、 X0.10+、X0.10-、 X0.11+、X0.11-、 X0.12+、X0.12-、 X0.13+、X0.13-、 X0.14+、X0.14-、 X0.15+、X0.15-	X0.0+、X0.0-、 X0.1+、X0.1-、 X0.2+、X0.2-、 X0.3+、X0.3-、 X0.8+、X0.8-、 X0.9+、X0.9-、 X0.10+、X0.10-、 X0.11+、X0.11-、 X0.12+、X0.12-、 X0.13+、X0.13-、 X0.14+、X0.14-、 X0.15+、X0.15-	X0.0+、X0.0-、 X0.1+、X0.1-、 X0.2+、X0.2-、 X0.3+、X0.3-、 X0.4+、X0.4-、 X0.5+、X0.5-、 X0.6+、X0.6-、 X0.7+、X0.7-、 X0.8+、X0.8-、 X0.9+、X0.9-、 X0.10+、X0.10-、 X0.11+、X0.11-、 X0.12+、X0.12-、 X0.13+、X0.13-、 X0.14+、X0.14-、 X0.15+、X0.15-	X0.0、X0.1、X0.8、 X0.9、X0.12、 X0.13
	输出信号	Y0.8、Y0.9、Y0.10、 Y0.11	Y0.0+、Y0.0-、 Y0.1+、Y0.1-、 Y0.2+、Y0.2-、 Y0.3+、Y0.3-、 Y0.4+、Y0.4-、 Y0.5+、Y0.5-、 Y0.6+、Y0.6-、 Y0.7+、Y0.7-、 Y0.8、Y0.9、 Y0.10、Y0.11	Y0.0+、Y0.0-、 Y0.1+、Y0.1-、 Y0.2+、Y0.2-、 Y0.3+、Y0.3-、 Y0.4+、Y0.4-、 Y0.5+、Y0.5-、 Y0.6+、Y0.6-、 Y0.7+、Y0.7-、 Y0.8、Y0.9、 Y0.10、Y0.11
外接通讯端口		Mini USB 通讯端口 Ethernet 通讯端口 DMCNET 通讯端口	Mini USB 通讯端口 □ Ethernet 通讯端口	Mini USB 通讯端口 Ethernet 通讯端口 □
扩充储存装置	Mini SD 卡，最大支持至 32GB		无	
基本指令	27 个			
应用指令	130 个			
M 码	OX0~99 (运动子程序/Positioning Program): M02 程序停止 (END) M00~M01、M03~M101、M103~M65535 做为程序暂停 (WAIT)，可自由使用。			



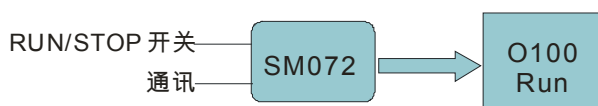
项目	规格			
	AH20MC-5A	AH10PM-5A	AH15PM-5A	AH05PM-5A
G 码	G0 (快速移动)、G1 (直线差补)、G2 (顺时针圆弧差补)、G3 (逆时针圆弧差补)、G4 (延迟时间)、G17 (XY 平面设定)、G18 (XZ 平面设定)、G19 (YZ 平面设定)、G90 (绝对坐标) 及 G91 (相对坐标)。			
计数器	6 组	6 组	6 组	1 组
高速捕捉/比较	8 组	8 组	8 组	3 组
中断装置	9 组	9 组	9 组	5 组

1.2 软件O100 主程序架构

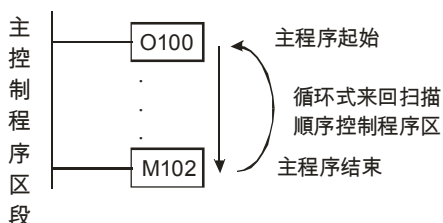
O100 主程序为 PLC 顺序控制程序，主要为 AH500 运动控制模块模块运作的顺序控制主程序。O100 主程序区段中，只支持基本指令及应用指令；使用这两种类型指令，除了提供 I/O 点信号数据处理及 Pn 子程序的调用，还控制启动 OX0~OX99 共 100 组的 OX 运动子程序，所以 O100 主程序为控程序的建立，再经由主控程序去设定及启动运动子程序，这是运作控制架构阶层关系。O100 主程序运作流程及特性说明如下：

1. 启动 O100 主程序方法有二种：

- 当上电状态中，AH500 运动控制模块的 RUN/STOP 开关，由 STOP→RUN 时，SM072 标志将自动为 ON，而 O100 主程序为 RUN 状态。
- 当上电状态中，可利用通讯的方式，将 SM072 标志设定为 ON，也可使 O100 主程序为 RUN 状态。



2. 其程序扫描方式为循环式来回扫描，当 O100 主程序启动后，将由 O100 起始标志进行扫描，扫描到 M102 主程序结束指令时，再重新返回 O100 起始标志持续的扫描，如下图：



3. O100 主程序停止运作方式有三种：

- 当上电状态中，AH500 运动控制模模块的 RUN/STOP 开关，由 STOP→RUN 时，SM072 标志将自动为 OFF，而 O100 主程序为 STOP 状态，此时 OX 及 Pn 子程序也会停止运作。
- 当上电状态中，可利用通讯的方式，将 SM072 标志设定为 OFF，使 O100 主程序为 STOP 状态，此时 OX 及 Pn 子程序也会停止运作。
- 当主程序于程序设计编译或程序运作出现错误时，O100 主程序也会自动停止运作。

4. O100 主程序有支持基本指令及应用指令两种，因此，可依实际需要设计控制程序，并在主程序中设定运动指令的参数，更可设定运动程序启动编号，进而启动 OX0~OX99 运动子程序。

- O100 主程序不支持运动指令及 G 码指令，因此，请将运动指令及 G 码规划在 OX0~OX99 运动子程序之中，请参考第 1.3 章说明。
- O100 主程序可调用 Pn 子程序，请参考第 1.4 章之说明。

1

5. 上述 O100 主程序说明整理如下表：

O100 主程序	说明
程序起始	O100 主程序起始标志(PMSoft 梯形图编辑模式下会自动设定·使用者不需撰写)
程序结束	M102 主程序结束指令(PMSoft 梯形图编辑模式下会自动设定·使用者不需撰写)
程序执行	1. AH500 运动控制模块上的 RUN/STOP 开关·STOP→RUN 状态 2. 利用通讯将 M1072 OFF→ON
运作特性	循环式来回扫描方式及运作
支持指令	基本指令及应用指令
程序数量	于程序中·O100 主程序只能有一组。
特色及功能	1. 为 PLC 顺序控制程序 2. 可启动 OX0~OX99 运动子程序及调用 Pn 子程序 3. 与 OX0~OX99 运动子程序及 Pn 子程序搭配使用时·3 组程序顺序可任意排列

1.2.1 O100 主程序手动功能运动方式

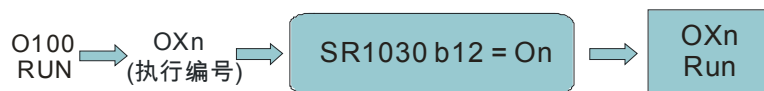
在 O100 主程序内·使用者可利用特殊寄存器·自行规画手动功能运动模式 (详细设定方法请参考第 7 章)。

1.3 OX运动子程序架构

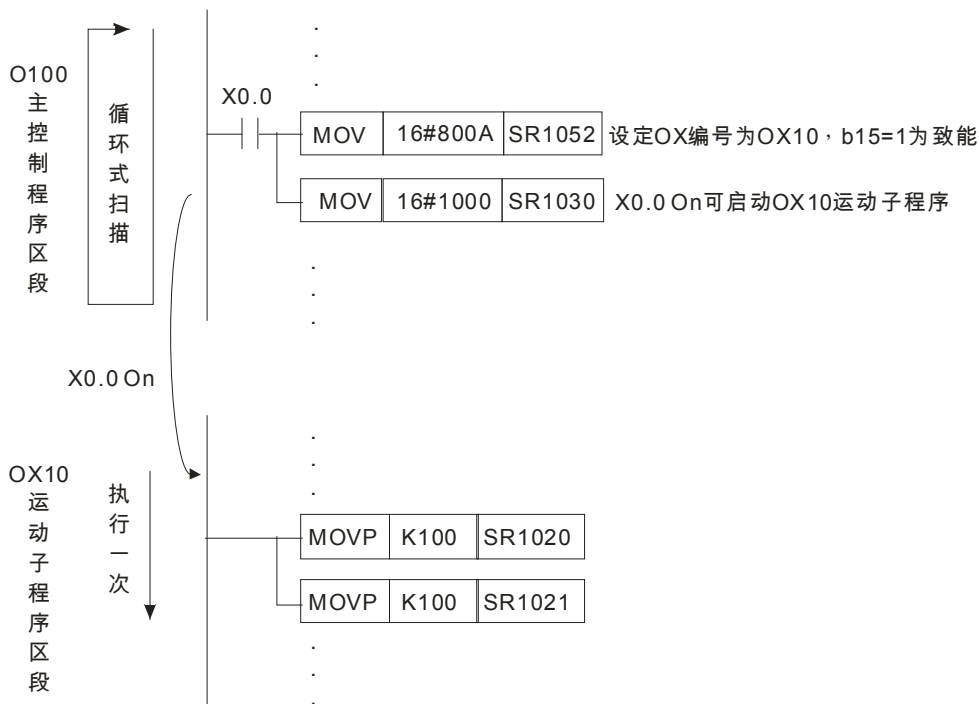
OX0~OX99 运动子程序为运动控制程序，主要为控制 AH500 运动控制模块主机进行运动轴运动之子程序。OX0~OX99 运动子程序区段中，有支持基本指令、应用指令、运动指令及 G 码指令，并在程序中可规划调用 Pn 子程序。主要提供使用者编写设计控制运动轴移动路径程序，下列将对 OX 运动子程序运作流程及特性说明。

1. 启动 OX0~OX99 运动子程序的方法如下：

- 当 O100 主程序为 RUN 状态下，可在 O100 主程序中设定 OX 执行编号 00~99 (SR1052 : 16#8000~16#8063) 后，设定运动轴运转命令 (SR1030) 的 b12 为 ON，即可启动 OX 运动子程序。
- 当启动 OX 运动子程序时，请确定无任何运动子程序正在运转中，才可进行启动。



2. 其程序扫描方式为每启动一次就执行一次，当 O100 主程序启动 OX 运动子程序后，将由 OX 运动子程序的起始标志进行执行，执行到 M2 运动子程序结束指令时，即结束运动子程序，如下图所示：



上图中当 X0 ON 时，将启动 OX10 运动子程序，执行到 M2 运动子程序结束指令时，即停止 (只执行一次，如须再执行可重新启动 X0 开关，即可重新启动 OX10 运动子程序)。

3. OX 运动子程序停止运作方式有四种：

- 当上电状态中，模块的 RUN/STOP 开关，由 RUN→STOP 时，SM072 标志将自动为 OFF，而 O100 主程序为 STOP 状态，OX 运动子程序也将停止运作。

1

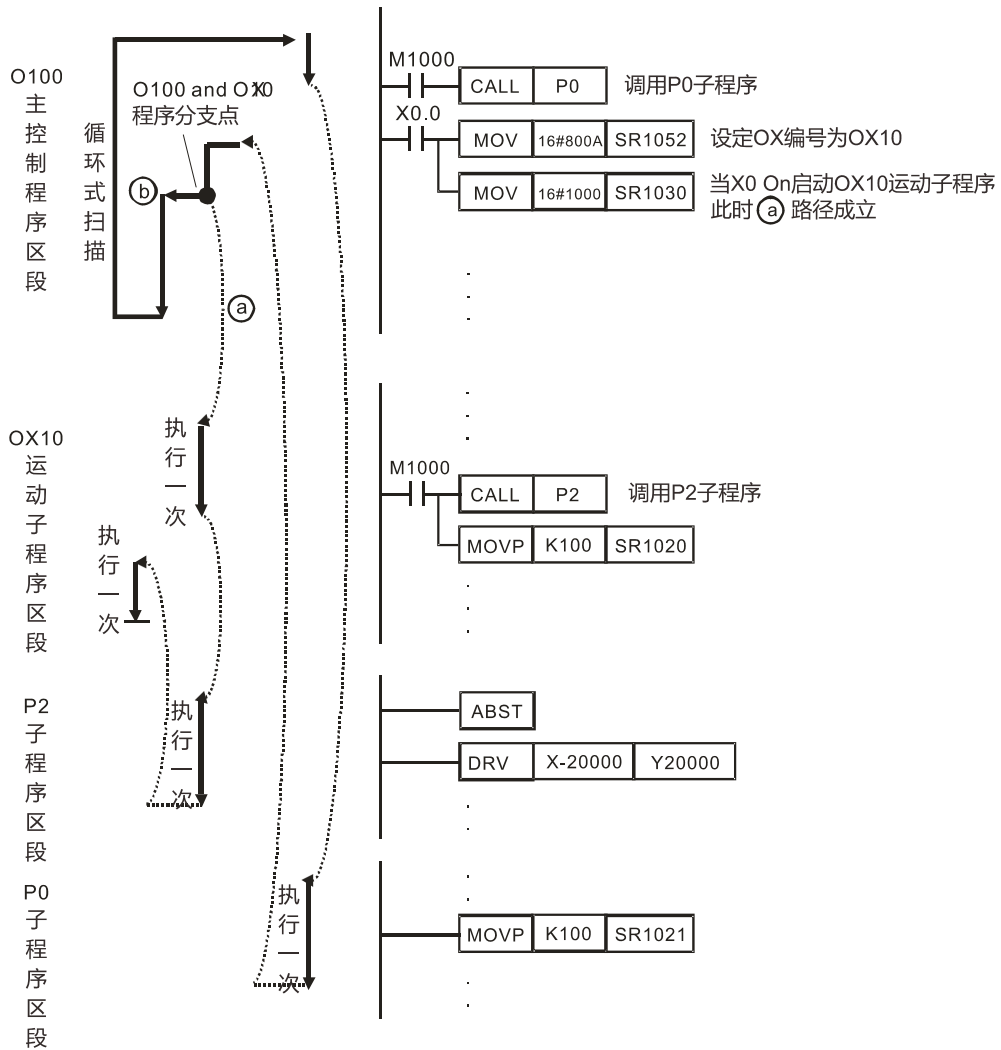
- 当上电状态中，可利用通讯的方式，将 SR1030 设定为 0 或 SM072 设为 OFF，也可使 OX 运动子程序停止运作。
 - 当运动子程序于程序设计编译或程序运作出现错误时，OX 运动子程序也会自动停止运作。
4. OX 运动子程序支持基本指令、应用指令、运动指令及 G 码指令四种，因此可依需求使用这四种指令来设计运动程序，及设定运动轴参数，以达到运动轴运动控制。
- 请将上述四种指令规划在 OX0~OX99 运动子程序之中！
 - OX 运动子程序有支持 Pn 子程序的调用，请参考第 1.4 章之说明！
5. 上述 OX 运动子程序说明整理如下表：

OX 运动子程序	说明
程序起始	OX 运动子程序 (OX0~OX99，共 100 组运动子程序) (PMSoft 梯形图编辑模式下会自动设定，使用者不需撰写)
程序结束	M2 运动子程序结束 (PMSoft 梯形图编辑模式下会自动设定，使用者不需撰写)
程序执行	1. 由 O100 主程序为 RUN 状态中，设定 SR1030_b12 为 1，即可启动 OX 运动子程序。 2. 由 O100 主程序为 RUN 状态中，利用通讯设定 SR1030_b12 为 1，亦可启动 OX 运动子程序。 注意：当启动 OX 运动子程序时，请确定无任何运动子程序正在运转中，才可进行启动。
运作特性	每启动一次只执行一次，若须再次执行可再重新启动一次即可。
支持指令	基本指令、应用指令、运动指令及 G 码指令，共四种 注意：使用基本指令、应用指令时，请避免使用脉冲型式指令。
程序数量	于程序中，OX 运动子程序能有 100 组，若须启动其它 OX 编号，可设定 SR1052 后，再启动 (设定 SR1030_b12 为 1) 即可。
特色及功能	1. 为运动子程序 (只能由 O100 主程序设计程序来启动) 2. 提供多轴控制，请参考 G 码指令！ 3. 提供程序设计及通讯方式，来启动 / 停止 OX 运动子程序。 4. 可调用 Pn 子程序 5. 与 O100 主程序及 Pn 子程序搭配使用时，3 组程序顺序可任意排列。

1.4 Pn 子程序架构

Pn 子程序为一般用子程序，主要为提供 O100 主程序及 OX 运动子程序之调用子程序使用。于 O100 主程序调用 Pn 子程序时，则 Pn 子程序区段支持基本指令及应用指令；若于 OX0~OX99 运动子程序调用 Pn 子程序时，则 Pn 子程序区段支持基本指令、应用指令、运动指令及 G 码。而不论在 O100 主程序或 OX 运动子程序中调用 Pn 子程序，都会在 Pn 子程序执行时，跳至 Pn 子程序进行动作，执行到 SRET 时将返回 CALL Pn 指令的下一行继续执行。

- 启动 Pn 子程序的方法如下：
 - 于 O100 主程序中调用 Pn 子程序，即可启动 Pn 子程序。
 - 于 OX 运动子程序中调用 Pn 子程序，即可启动 Pn 子程序。
- 其程序扫描方式为调用一次只执行一次，O100 主程序中调用 Pn 子程序后，将执行 Pn 子程序，执行到 SRET 子程序结束指令时，即结束 Pn 子程序，并回到 Pn 指针处的下一行程序继续执行；当由 OX 运动子程序调用 Pn 子程序后动作亦同，如下图之范例程序说明：



其中 P0 子程序区段中，可自由的编写基本指令及应用指令；于 P2 子程序区段中，可自由的编写基本指令、应用指令、运动指令及 G 码指令。

1

3. Pn 子程序停止运作方式有三种：

- 当上电状态中，AH500 运动控制模块的 RUN/STOP 开关，由 RUN→STOP 时，SR072 标志将自动为 OFF，而 O100 主程序为 STOP 状态，OX 子程序及 Pn 子程序都将停止运作。
- 当上电状态中，可利用通讯的方式，将 SR1030 设定为 0，也可使 OX 运动子程序停止运作，则所调用的 Pn 子程序也停止运作。
- 当 Pn 子程序于程序运作出现错误时，Pn 子程序也会自动停止运作。请参考附录之侦错码原因对照表！

4. 于 O100 主程序调用 Pn 子程序时，则 Pn 子程序只支持基本指令及应用指令；若于 OX0~OX99 运动子程序调用 Pn 子程序时，则 Pn 子程序支持基本指令、应用指令、运动指令及 G 码指令。

5. 上述 Pn 子程序说明整理如下表：

Pn 子程序	说明
程序起始	Pn 子程序起始标志 (P0~P255) (PMSoft 梯形图编辑模式下会自动设定，使用者不需撰写)
程序结束	SRET 子程序结束 (PMSoft 梯形图编辑模式下会自动设定，使用者不需撰写)
程序执行	1. 由 O100 主程序中，调用 Pn 子程序。 2. 由 OX 运动子程序中，调用 Pn 子程序。
运作特性	每调用一次只执行一次，若须再次执行可再重新调用一次即可
支持指令	1. 由 O100 调用时：支持基本指令及应用指令，共二种。 2. 由 OX 调用时：支持基本指令、应用指令、运动指令及 G 码指令，共四种 注意：由 OX 调用且使用基本指令、应用指令时，请避免使用脉冲型式指令。
程序数量	程序中，Pn 子程序能有 256 组。
特色及功能	1. 为一般用子程序 2. 提供给 O100 主程序及 OX 运动子程序调用用 3. 与 O100 主程序及 OX 运动子程序搭配使用时，3 组程序顺序可任意排列。

1

为了方便说明范例程序，故分成 (1) ~ (5) 区段，来进行说明，并将程序规划如下：



范例程序设计说明：

1. 范例程序中的编写顺序是由 (1) 到 (5) 的方式，而在排列顺序上是可以随意的摆放，而没有硬性的规则。
2. O100 主程序 (2) 只能有一组，且不可被其它程序调用，但可自由调用 OX 运动子程序及 Pn 子程序。
3. OX 运动子程序可被 O100 主程序及 Pn 子程序调用，并且可以调用 Pn 子程序。
4. Pn 子程序可被 O100 主程序及 OX 运动子程序调用，并且可以调用 OX 运动子程序。

注意事项：

1. 不可同时执行两个以上的 OX 运动子程序，故当 OX0 运动子程序执行时，则 OX3 运动子程序无法运作；反之，当 OX3 运动子程序执行时，则 OX0 运动子程序无法运作。
2. 由 O100 主程序或 Pn 子程序启动 OX 运动子程序后，就继续执行下一行程序，而不理会 OX 运动子程序。
3. OX 运动子程序启动后，只执行一次，若须再行运作，可重新启动，即可再动作一次。

针对前述之范例，其各程序区段所支持之指令，说明如下表：(O：有支援，X：不支援)

区段	O100 主程序	OX 运动子程序 (OX0、OX3)	P1 子程序	P2 子程序
基本指令	O	O	O	O
应用指令	O	O	O	O
运动指令	X	O	O	X
G 码	X	O	O	X
说明	-	-	由 OX 运动子程序调用，故有支持运动指令、G 码	由 O100 主程序调用，故不支持运动指令、G 码

补充说明：

	主程序	子程序	运动程序
摆放顺序	任意	任意	任意
程序执行	正常 RUN	主程序或运动程序调用	主程序或子程序调用
运作特性	循环运作	调用一次执行一次	调用一次执行一次
数量	1	256 个，使用者需求使用	100 个，使用者需求使用

1

MEMO

2

第2章 硬件规格及配线方式

目录


2.1	硬件规格.....	2-2
2.1.1	一般规格	2-2
2.1.2	输入点电气规格	2-3
2.1.3	输出点电气规格	2-9
2.1.4	外观尺寸	2-13
2.1.5	模块部位介绍.....	2-15
2.2	配线方式.....	2-19
2.2.1	I/O 连接线及配线模块	2-19
2.2.2	输入接点配线.....	2-25
2.2.3	输出接点配线.....	2-28
2.2.4	与下位驱动器之配线	2-30
2.3	通讯端口.....	2-36

2.1 硬件规格

本章节仅提供电气规格及安装配线说明，其它详细之程序设计及指令说明，请参考本手册其余章节；选购之周边装置详细说明，请参考产品随机手册。

2

2.1.1 一般规格

项目	规格
接头	高精密度端子接头，需转接至配线端子座做配线。
电源电压	5 VDC (-15~20%) · 24 VDC (-15~20%) (由主机经由内部总线供应)
消耗电力	2 W
绝缘电压	2,500 VDC
重量	150g
噪声免疫力	ESD (IEC 61131-2 · IEC 61000-4-2) : ±10 kV Air Discharge EFT (IEC 61131-2 · IEC 61000-4-4) : Communication I/O : ±4 kV CS (IEC 61131-2 · IEC 61000-4-6) : 0.15~80 MHz · 3 Vrms RS (IEC 61131-2 · IEC 61000-4-3) : 80~100 MHz · 10 V/m · 1.4~2.0 GHz
操作/储存环境	操作：-20~70°C (温度) · 5~95% (湿度) · 污染等级 2。 储存：-40~85°C (温度) · 5~95% (湿度)。
耐振动/冲击	国际标准规范 IEC61131-2 · IEC68-2-6 (TEST Fc) /IEC61131-2&IEC 68-2-27 (TEST Ea)。
标准	CE 

2.1.2 输入点电气规格

● AH20MC-5A

规格		项目	
		差动输入	
		高速 200kHz	
输入接线型式		独立接线	
输入信号电压		5~24 VDC	
最大电流输入值		15 mA	
动作位准	OFF→ON	20 us	
	ON→OFF	30 us	
反应时间/噪声抑制		10 ms/0.5 us	

端子	说明	响应特性	最大输入	
			电流值	电压值
X0.0+、X0.0- X0.1+、X0.1- X0.2+、X0.2- X0.3+、X0.3-	1. 差动信号输入 2. 端子功能： <ul style="list-style-type: none"> ● 高速计数器： <ul style="list-style-type: none"> (a) 计数器 0~计数器 5 的 Reset 信号输入 (b) X0.0 为计数器 0 Rest 输入端、X0.1 为计数器 1 Rest 输入端、X0.2 为计数器 2 和计数器 4 共享 Rest 输入端、X0.3 为计数器 3 和计数器 5 共享 Rest 输入端 ● 高速比较与捕捉： <ul style="list-style-type: none"> 可设定为高速捕捉功能的触发信号。 	200k Hz	15mA	5~24V
X0.8+、X0.8- X0.9+、X0.9-	1. 差动信号输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制：手摇轮脉冲输入 ● 高速计数器： <ul style="list-style-type: none"> (a) 计数器 0 的计数输入来源端 (b) X0.8 和 X0.9 为计数器 0 的 A 相与 B 相输入点 ● 高速比较与捕捉： <ul style="list-style-type: none"> 可设定为高速捕捉功能的触发信号。 ● 中断信号输入 	200k Hz	15mA	5~24V

2

2

端子	说明	响应特性	最大输入	
			电流值	电压值
X0.10+、X0.10- X0.11+、X0.11- X0.12+、X0.12- X0.13+、X0.13 X0.14+、X0.14- X0.15+、X0.15-	1. 差动信号输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制： <p>Axis1~Axis6 Dog 点脉冲输入·此功能用于单轴插入运动之用</p> ● 高速计数器： <p>(a) 计数器 1~计数器 5 的计数输入来源</p> <p>(b) A 相信号 X0.10 为计数器 1 输入点 X0.12 为计数器 2 和计数器 4 共享输入点 X0.14 为计数器 3 和计数器 5 共享输入点。</p> <p>(c) B 相信号 X0.11 为计数器 1 输入点 X0.13 为计数器 2 和计数器 4 共享输入点 X0.15 为计数器 3 和计数器 5 共享输入点。</p> ● 高速比较与捕捉： <p>可设定为高速捕捉功能的触发信号。</p> ● 中断信号输入 	200k Hz	15mA	5~24V

● AH10PM-5A

规格		项目	
		差动输入 高速 200kHz	集电极 100kHz (*1)
输入接线型式		独立接线	由端子 S/S 变换接线为 SINK 或 SOURCE
输入信号电压		5~24 VDC	24 VDC
最大电流输入值		15 mA	
动作位准	OFF→ON	20 us	
	ON→OFF	30 us	
反应时间/噪声抑制		10 ms/0.5 us	

端子	说明	响应特性	最大输入	
			电流值	电压值
X0.0+、X0.0- X0.1+、X0.1- X0.2+、X0.2- X0.3+、X0.3-	1. 差动信号输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制：Axis1~Axis4 PG 点脉冲输入 ● 高速计数器：计数器 0~计数器 5 的 Reset 信号输入。X0.0 为计数器 0 Rest 输入点、X0.1 为计数器 1 Rest 输入点、X0.2 为计数器 2 和计数器 4 Rest 共享输入点、X0.3 为计数器 3 和计数器 5 Rest 共享输入点。 ● 高速比较与捕捉：可设定为高速捕捉功能的触发信号。 	200k Hz	15mA	5~24V
X0.8、X0.9	1. 单相/AB 相输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制：手摇轮脉冲输入 ● 高速计数器： <ol style="list-style-type: none"> 1. 计数器 0 的计数来源 2. X0.8 和 X0.9 为计数器 0 的 A 相与 B 相输入点 ● 高速比较与捕捉：可设定为高速捕捉功能的触发信号。 ● 中断信号输入 	100k Hz(*1)	15mA	24V
X0.10、X0.11 X0.12、X0.13 X0.14、X0.15	1 单相/AB 相输入 2 端子功能： <ul style="list-style-type: none"> ● 运动控制：Axis1~Axis6 Dog 点脉冲输入 ● 高速计数器： <ol style="list-style-type: none"> (a) 计数器 1~计数器 5 的计数输入来源 (b) A 相信号 X0.10 为计数器 1 输入点 X0.12 为计数器 2 和计数器 4 共享输入点 X0.14 为计数器 3 和计数器 5 共享输入点。 (c) B 相信号 X0.11 为计数器 1 输入点 X0.13 为计数器 2 和计数器 4 共享输入点 X0.15 为计数器 3 和计数器 5 共享输入点。 ● 高速比较与捕捉：可设定为高速捕捉功能的触发信号。 ● 中断信号输入 	100k Hz(*1)	15mA	24V

*1. 若 AB 相输入需达到 200kHz 输入频率，需于每个通道并联 1kΩ (2W) 电阻。

2

● AH15PM-5A

规格	项目	差动输入	集电极
		高速 200kHz	100kHz (*1)
输入接线型式		独立接线	由端子 S/S 变换接线为 SINK 或 SOURCE
输入信号电压		5~24 VDC	24 VDC
最大电流输入值		15 mA	
动作位准	OFF→ON	20 us	
	ON→OFF	30 us	
反应时间/噪声抑制		10 ms/0.5 us	

端子	说明	响应特性	最大输入	
			电流值	电压值
X0.0+、X0.0- X0.1+、X0.1- X0.2+、X0.2- X0.3+、X0.3-	1. 差动信号输入 2. 端子功能： ● 运动控制：Axis1~Axis4 PG 点脉冲输入 ● 高速计数器：计数器 0~计数器 5 的 Reset 信号输入。X0.0 为计数器 0 Rest 输入点、X0.1 为计数器 1 Rest 输入点、X0.2 为计数器 2 和计数器 4 Rest 共享输入点、X0.3 为计数器 3 和计数器 5 Rest 共享输入点。 ● 高速比较与捕捉：可设定为高速捕捉功能的触发信号。	200k Hz	15mA	5~24V
X0.4、X0.5、 X0.6、X0.7	1 单相/AB 相输入 2 端子功能： ● 运动控制：Axis1~Axis4 Dog 点脉冲输入	100k Hz (*1)	15mA	24V
X0.8+、X0.8-、 X0.9+、X0.9-	1. 差动信号输入 2. 端子功能： ● 运动控制：手摇轮脉冲输入 ● 高速计数器 ■ 计数器 0 的计数来源 ■ X0.8 和 X0.9 为计数器 0 的 A 相与 B 相输入点 ● 高速比较与捕捉：可设定为高速捕捉功能的触发信号。 ● 中断信号输入	200k Hz	15mA	5-24V
X0.10、X0.11 X0.12、X0.13	1 单相/AB 相输入 2 端子功能：	100k Hz (*1)	15mA	24V

端子	说明	响应特性	最大输入	
			电流值	电压值
X0.14、X0.15 X1.0、X1.1	<ul style="list-style-type: none"> ● 运动控制 :Axis1~Axis4 LSP/NSP 点脉冲输入 X0.10 为 LSP0、X0.11 为 LSN0、 X0.12 为 LSP1、X0.13 为 LSN1、 X0.14 为 LSP2、X0.15 为 LSN2、 X1.0 为 LSP3、X1.1 为 LSN3。 ● 高速计数器 : <ul style="list-style-type: none"> ■ 计数器 1~计数器 5 的计数输入来源 ■ A 相信号 :X0.10 为计数器 1 输入点、X0.12 为计数器 2 和计数器 4 共享输入点、X0.14 为计数器 3 和计数器 5 共享输入点。 ■ B 相信号 :X0.11 为计数器 1 输入点、X0.13 为计数器 2 和计数器 4 共享输入点、X0.15 为计数器 3 和计数器 5 共享输入点。 ● 高速比较与捕捉 :可设定为高速捕捉功能的触发信号。 ● 中断信号输入 :X0.10、X0.11、X0.12、X0.13、X0.14、X0.15 			
X1.2、X1.3、 X1.4、X1.5	1 单相/AB 相输入	100k Hz (*1)	15mA	24V

*1. 若 AB 相输入需达到 200kHz 输入频率，需于每个通道并联 1kΩ (2W) 电阻。

● AH05PM-5A

项目		集电极
		高速 100kHz(*1)
规格		
输入接线型式		由端子 S/S 变换接线为 SINK 或 SOURCE
输入信号电压		24 VDC
最大电流输入值		15 mA
动作位准	OFF→ON	20 us
	ON→OFF	30 us
反应时间/噪声抑制		10 ms/0.5 us

2

端子	说明	响应特性	最大输入	
			电流值	电压值
X0.0、X0.1	1. 单相/AB 相输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制：Axis1~Axis2 PG 点脉冲输入 ● 高速计数器：X0.0 为计数器 0 Reset 输入点 3. 高速比较与捕捉：可设定为高速捕捉功能的触发信号。	100k Hz(*1)	15mA	24V
X0.8、X0.9	1. 单相/AB 相输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制：手摇轮脉冲输入 ● 高速计数器： <ul style="list-style-type: none"> ■ 计数器 0 的计数来源：X0.8 和 X0.9 为计数器 0 的 A 相与 B 相来源端 ● 高速比较与捕捉： <ul style="list-style-type: none"> 可设定为高速捕捉功能的触发信号。 3. 中断信号输入	100k Hz(*1)	15mA	24V
X0.12、X0.13	1. 单相/AB 相输入 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制：Axis1~Axis2 Dog 点脉冲输入 ● 高速比较与捕捉：可设定为高速捕捉功能的触发信号。 3. 中断信号输入	100k Hz(*1)	15mA	24V

*1. 若 AB 相输入需达到 200kHz 输入频率，需于每个通道并联 1kΩ (2W) 电阻。

2.1.3 输出点电气规格

● AH20MC-5A

项目		晶体管输出
规格		
最高交换 (工作) 频率		200 kHz
输出点配置		Y0.8~Y0.11
工作电压		5~30 VDC
最大输出电流值		40 mA
隔离方式		光耦合隔离
电流规格	电阻性	0.5A/1 点 (2A/COM)
	电感性	9W (24VDC)
	灯泡	2W (24VDC)
响应时间	OFF→ON	0.2 us
	ON→OFF	
输出过电流保护		N/A

端子	说明	响应特性	最大输出	
			电流值	电压值
Y0.8、Y0.9 Y0.10、 Y0.11	1. 集电极高速脉冲输出 2. 端子功能： ● 高速比较与捕捉：可作为高速比较的输出	200k Hz	15mA	24V

● AH10PM-5A

项目		差动输出	晶体管输出
规格			
最高交换 (工作) 频率		1 MHz	200 kHz
输出点配置		Y0.0~Y0.7	Y0.8~Y0.11
工作电压		5 VDC	5~30 VDC
最大输出电流值		40 mA	40 mA
隔离方式		数字隔离	光耦合隔离
电流规格	电阻性	<25 mA	0.5 A/1 点 (4 A/COM)
	电感性	--	12 W (24 VDC)
	灯泡	--	2 W (24 VDC)
最大输出延迟时间	OFF→ON	0.2 us	
	ON→OFF		
输出过电流保护		有	无

2

端子	说明	响应特性	最大输出	
			电流值	电压值
Y0.8、Y0.9 Y0.10、Y0.11	1. 集电极高速脉冲输出 2. 端子功能： ●运动控制： (a) Axis1~Axis4 CLR 点脉冲输出 (b) Y0.8、Y0.9 也可做为 Axis 5 的控制脉冲信号源，Y0.10、Y0.11 也可做为 Axis 6 的控制脉冲信号源。 A 相输出：Y0.8(Axis 5)、Y0.10(Axis 6) B 相输出：Y0.9(Axis 5)、Y0.11(Axis 6) ●高速比较与捕捉： 可作为高速比较的输出	200k Hz	15mA	24V
Y0.0+、Y0.0-、 Y0.1+、Y0.1-、 Y0.2+、Y0.2-、 Y0.3+、Y0.3-、 Y0.4+、Y0.4-、 Y0.5+、Y0.5-、 Y0.6+、Y0.6-、 Y0.7+、Y0.7-	1. 差动信号输出 2. 端子功能： ●运动控制： (a) Axis1~Axis4 脉冲控制输出 A 相输出：Y0.0(Axis 1)、Y0.2(Axis 2)、Y0.4(Axis 3)、Y0.6(Axis 4) B 相输出：Y0.1(Axis 1)、Y0.3(Axis 2)、Y0.5(Axis 3)、Y0.7(Axis 4) (b) Y0.0+、Y0.0-也可当 Axis 5 的 CLR，Y0.1+、Y0.1-也可当 Axis 6 的 CLR。	1M Hz	5mA	5V

● AH15PM-5A

项目		差动输出	晶体管输出
规格			
最高交换 (工作) 频率		1 MHz	200 kHz
输出点配置		Y0.0~Y0.7	Y0.8~Y0.11
工作电压		5 VDC	5~30 VDC
最大输出电流值		40 mA	40 mA
隔离方式		数字隔离	光耦合隔离
电流规格	电阻性	<25 mA	0.5 A/1 点 (4 A/COM)
	电感性	--	12 W (24 VDC)
	灯泡	--	2 W (24 VDC)

规格		项目	差动输出	晶体管输出
		反应时间	OFF→ON	0.2 us
ON→OFF				
输出过电流保护			有	无

端子	说明	响应特性	最大输出	
			电流值	电压值
Y0.8、Y0.9 Y0.10、Y0.11	1. 集电极高速脉冲输出 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制： <ul style="list-style-type: none"> ■ Axis1~Axis4 CLR 点脉冲输出 ● 高速比较与捕捉： <ul style="list-style-type: none"> 可作为高速比较的输出 	200k Hz	15mA	24V
Y0.0+、Y0.0-、 Y0.1+、Y0.1-、 Y0.2+、Y0.2-、 Y0.3+、Y0.3-、 Y0.4+、Y0.4-、 Y0.5+、Y0.5-、 Y0.6+、Y0.6-、 Y0.7+、Y0.7-	1. 差动信号输出 2. 端子功能： <ul style="list-style-type: none"> ● 运动控制： <ul style="list-style-type: none"> ■ Axis1~Axis4 脉冲控制输出 <ul style="list-style-type: none"> A 相输出：Y0.0 (Axis 1)、Y0.2 (Axis 2)、Y0.4 (Axis 3)、Y0.6 (Axis 4) B 相输出：Y0.1 (Axis 1)、Y0.3 (Axis 2)、Y0.5 (Axis 3)、Y0.7 (Axis 4) ■ Y0.0+、Y0.0-也可当 Axis 5 的 CLR、Y0.1+、Y0.1-也可当 Axis 6 的 CLR。 	1M Hz	5mA	5V

2

2

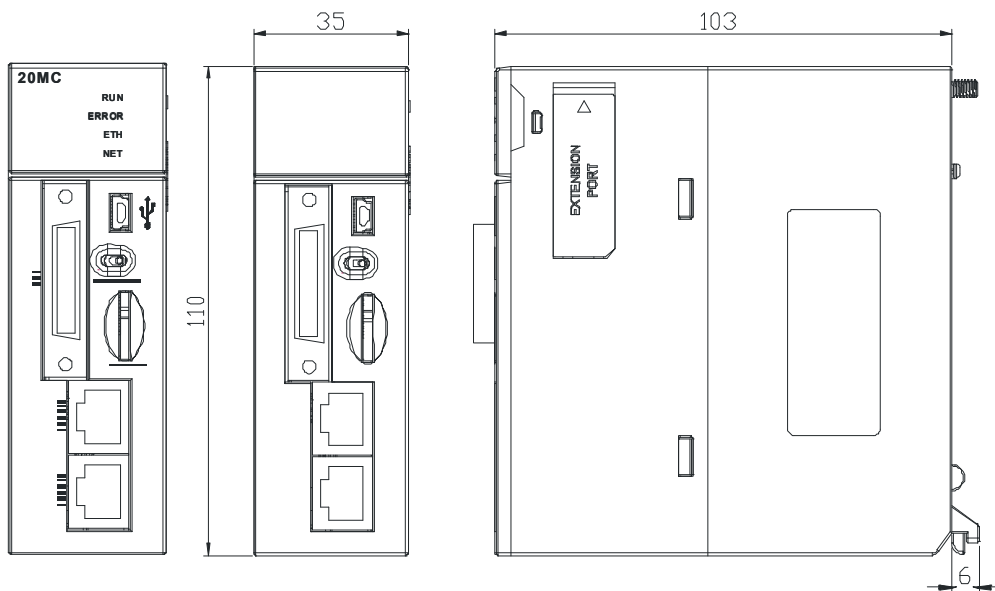
● AH05PM-5A

规格 \ 项目		双端差动输出	单端共点晶体管输出
最高交换 (工作) 频率		1 MHz	200 kHz
输出点配置		Y0.0~Y0.3	Y0.8~Y0.9
工作电压		5 VDC	5~30 VDC
最大输出电流值		40 mA	40 mA
隔离方式		数字隔离	光耦合隔离
电流规格	电阻性	<25 mA	0.5 A/1 点 (4 A/COM)
	电感性	--	12 W (24 VDC)
	灯泡	--	2 W (24 VDC)
响应时间	OFF→ON	0.2 us	
	ON→OFF		
输出过电流保护		有	无

端子	说明	响应特性	最大输出	
			电流值	电压值
Y0.8、Y0.9	1. 集电极高速脉冲输出 2. 端子功能： ● 运动控制：Axis1~Axis2 CLR 点脉冲输出。 ● 高速比较与捕捉：可提供高速比较功能的输出。	200k Hz	15mA	24V
Y0.0+、Y0.0-、Y0.1+、Y0.1-、Y0.2+、Y0.2-、Y0.3+、Y0.3-	1. 差动信号输出 2. 端子功能： ● 运动控制 (Axis1~Axis2 脉冲控制输出)： A 相输出：Y0.0(Axis 1)、Y0.2(Axis 2) B 相输出：Y0.1(Axis 1)、Y0.3(Axis 2)	1M Hz	5mA	5V

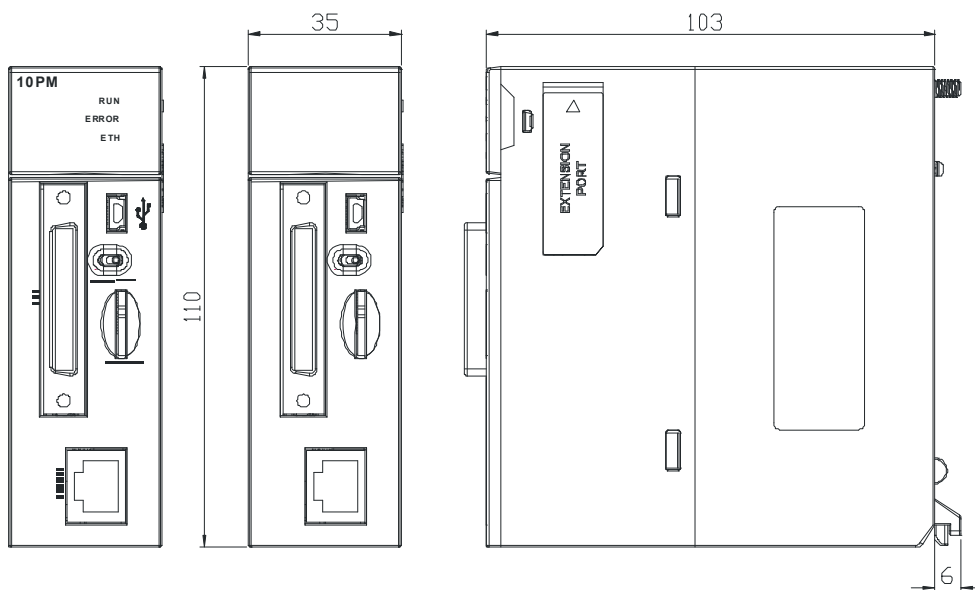
2.1.4 外观尺寸

- AH20MC-5A



单位：mm

- AH10PM-5A

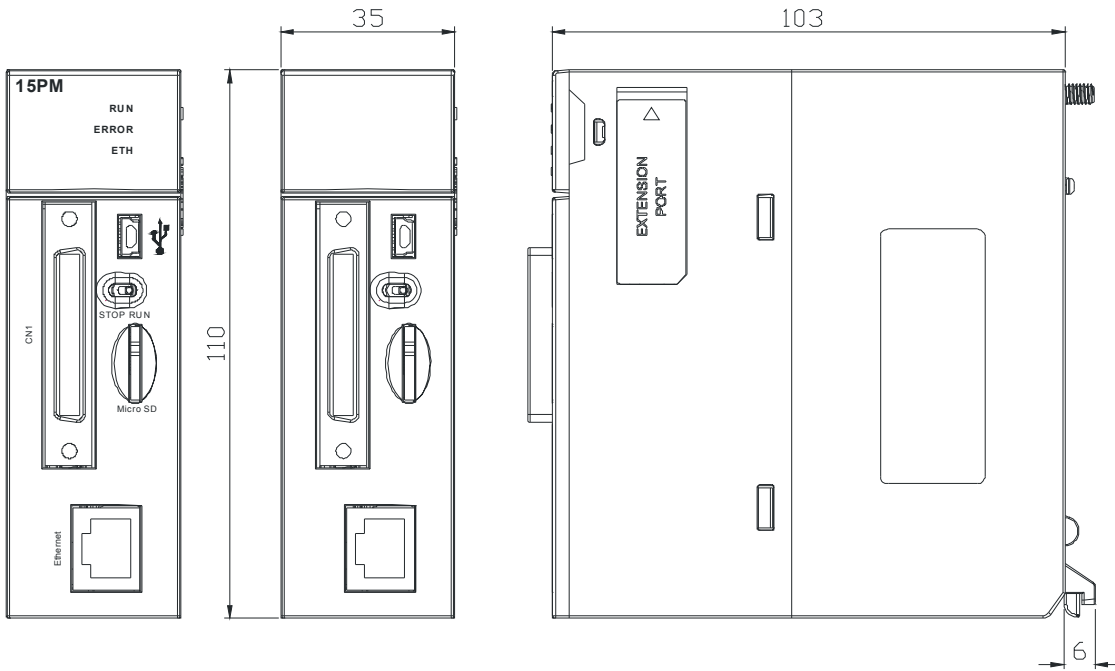


单位：mm

2

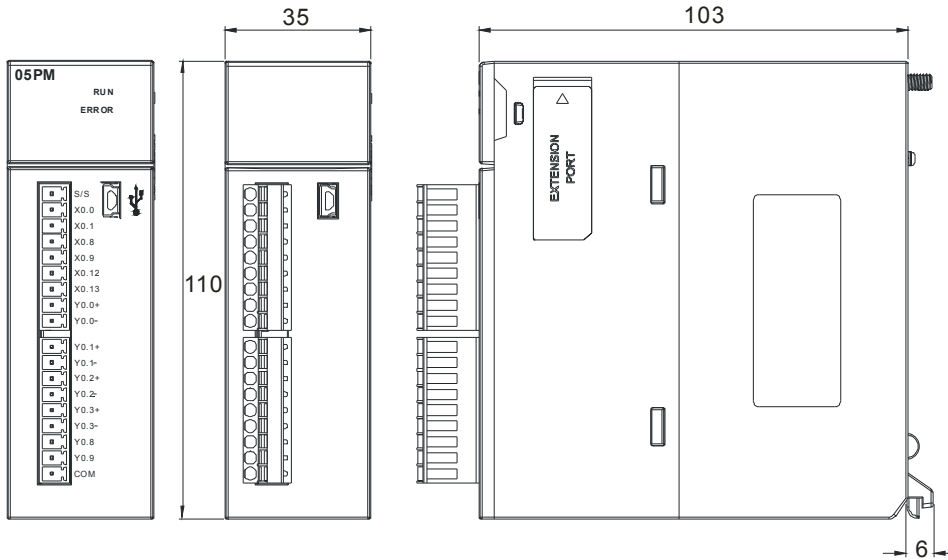
2

● AH15PM-5A



单位：mm

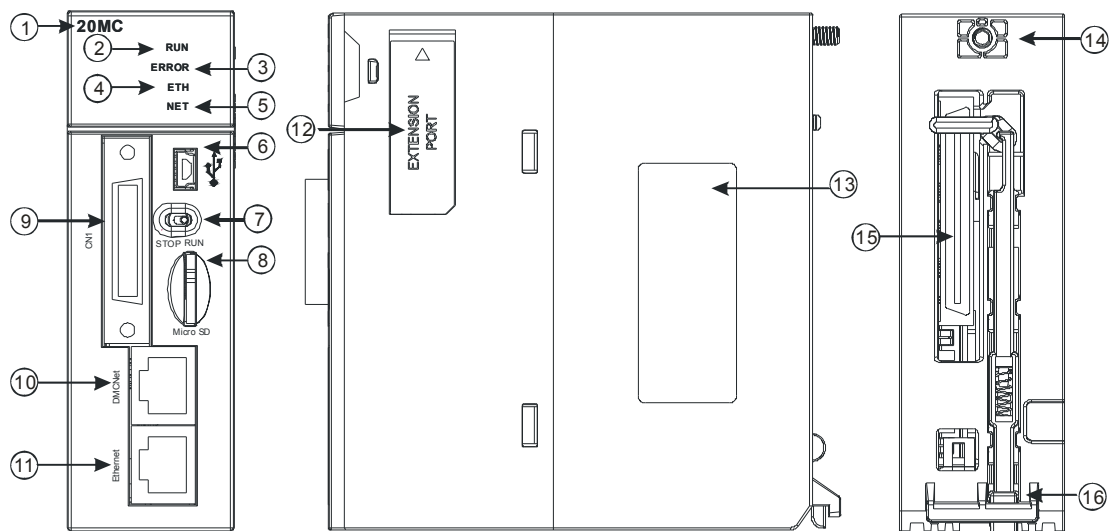
● AH05PM-5A



单位：mm

2.1.5 模块部位介绍

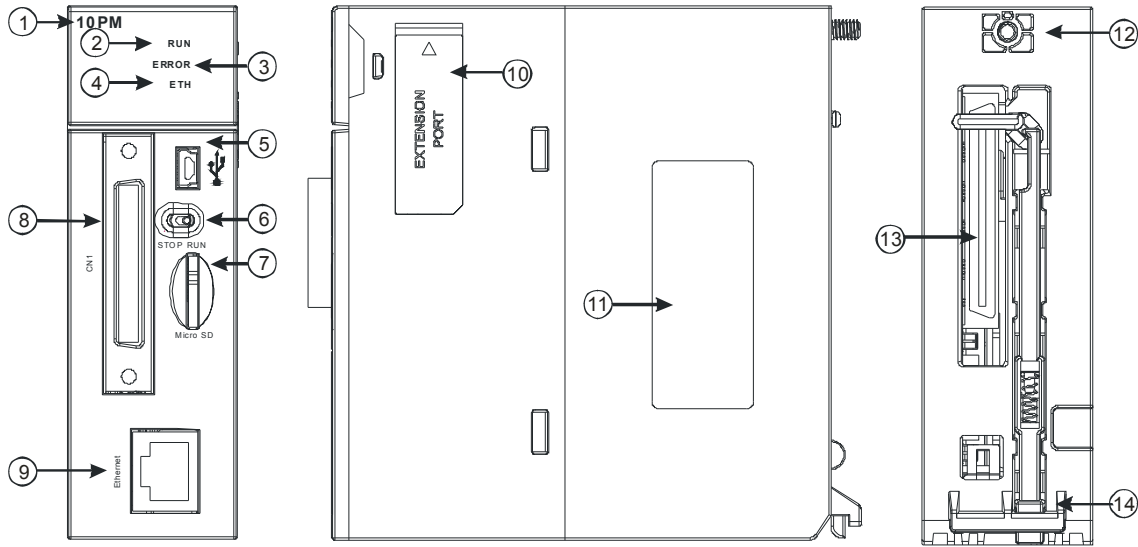
● AH20MC-5A



序号	名称	说明
1	机种名称	模块机种名称
2	运行指示灯 (绿灯)	指示模块的运行状态 常亮：模块状态为 RUN；灯灭：模块状态为 STOP
3	错误指示灯 (红灯)	指示模块的错误状态 闪烁：模块异常
4	Ethernet 联机指示灯 (绿灯)	指示 Ethernet 联机状态 常亮：Ethernet 联机中；灯灭：Ethernet 断线
5	DMC 联机指示灯 (绿灯)	指示 DMC 联机状态 常亮：DMC 联机中；灯灭：DMC 断线
6	USB 通讯端口	提供 mini USB 通讯接口
7	RUN/STOP 开关	RUN：执行使用者程序运行 STOP：停止使用者程序运行
8	SD 卡插槽	提供 SD 卡储存接口
9	CN1 I/O 端子	提供脉冲类型输入输出配置
10	DMC 通讯端口	提供 DMC 通讯接口
11	Ethernet 通讯端口	提供 Ethernet 通讯接口
12	EXTENSION PORT	提供更新使用接口
13	标签	铭牌
14	固定螺丝	固定模块
15	背板连接口	连接背板插槽
16	模块固定卡口	固定模块

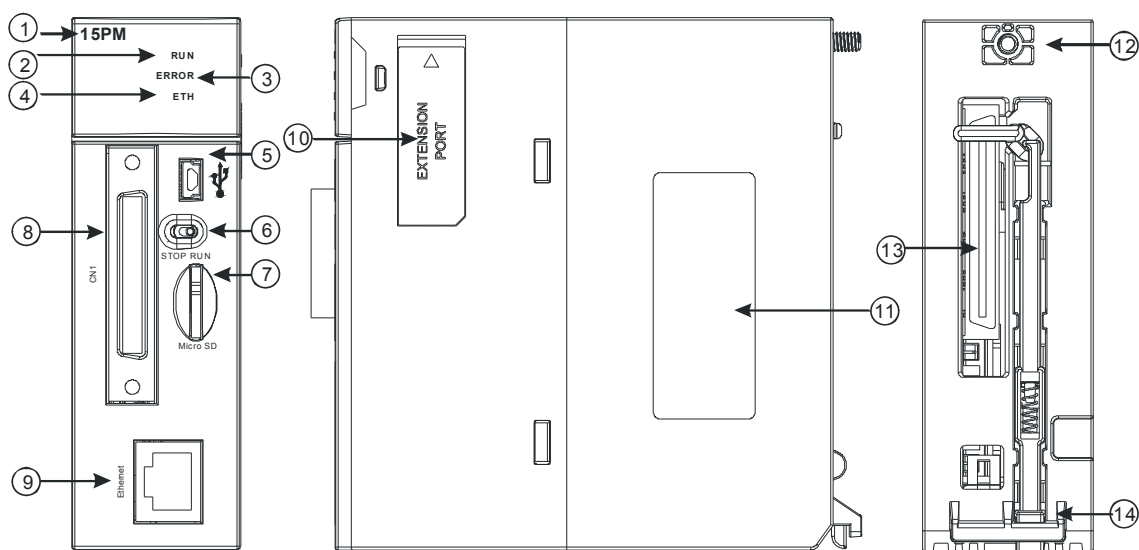
● AH10PM-5A

2



序号	名称	说明
1	机种名称	模块机种名称
2	运行指示灯 (绿灯)	指示模块的运行状态 常亮：模块状态为 RUN 灯灭：模块状态为 STOP
3	错误指示灯 (红灯)	指示模块的错误状态 闪烁：模块异常
4	Ethernet 联机指示灯 (绿灯)	指示 Ethernet 联机状态 常亮：Ethernet 联机中 灯灭：Ethernet 断线
5	USB 通讯端口	提供 mini USB 通讯接口
6	RUN/STOP 开关	RUN：执行使用者程序运行 STOP：停止使用者程序运行
7	SD 卡插槽	提供 SD 卡储存接口
8	CN1 I/O 端子	提供脉冲类型输入输出配置
9	Ethernet 通讯端口	提供 Ethernet 通讯接口
10	EXTENSION PORT	提供更新使用接口
11	标签	铭牌
12	固定螺丝	固定模块
13	背板连接口	连接背板插槽
14	模块固定卡口	固定模块

● AH15PM-5A

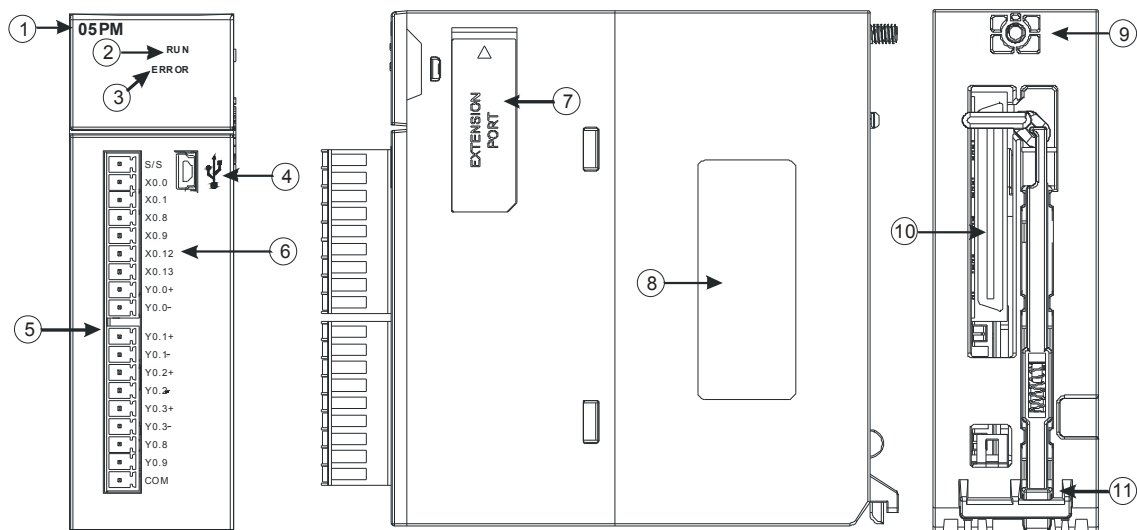


序号	名称	说明
1	机种名称	模块机种名称
2	运行指示灯 (绿灯)	指示模块的运行状态 常亮：模块状态为 RUN 灯灭：模块状态为 STOP
3	错误指示灯 (红灯)	指示模块的错误状态 闪烁：模块异常
4	Ethernet 联机指示灯 (绿灯)	指示 Ethernet 联机状态 常亮：Ethernet 联机中 灯灭：Ethernet 断线
5	USB 通讯端口	提供 mini USB 通讯接口
6	RUN/STOP 开关	RUN：执行用户程序运行 STOP：停止用户程序运行
7	SD 卡插槽	提供 SD 卡储存接口
8	CN1 I/O 端子	提供脉冲类型输入输出配置
9	Ethernet 通讯端口	提供 Ethernet 通讯接口
10	EXTENSION PORT	提供更新使用接口
11	标签	铭牌
12	固定螺丝	固定模块
13	背板连接口	连接背板插槽
14	模块固定卡口	固定模块

2

● AH05PM-5A

2



序号	名称	说明
1	机种名称	模块机种名称
2	运行指示灯 (绿灯)	指示模块的运行状态 常亮：模块状态为 RUN 灯灭：模块状态为 STOP
3	错误指示灯 (红灯)	指示模块的错误状态 闪烁：模块异常
4	USB 通讯端口	提供 mini USB 通讯接口
5	输入输出端子	提供脉冲类型输入输出配置
6	输入输出端子配置	端子配置
7	EXTENSION PORT	提供更新使用接口
8	标签	铭牌
9	固定螺丝	固定模块
10	背板连接接口	连接背板插槽
11	模块固定卡口	固定模块

2.2 配线方式

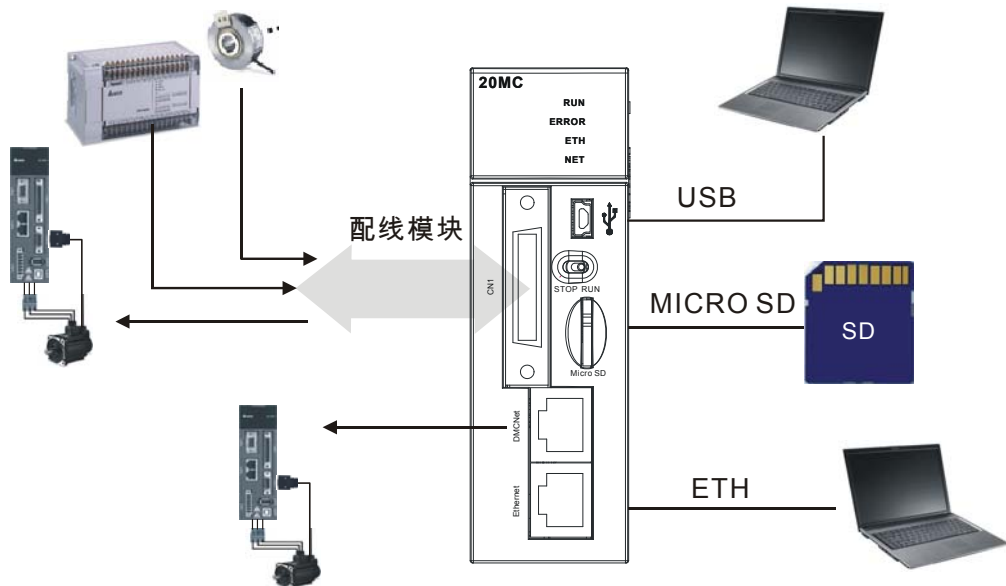
本机为开放型 (OPEN TYPE) 机壳，因此使用者使用本机时，必须将之安装于具防尘、防潮及免于电击/冲击意外之外壳配线箱内。另必须具备保护措施 (如：特殊之工具或钥匙才可打开) 防止非维护人员操作或意外冲击本体，造成危险及损坏。

交流输入电源不可连接于输入/出信号端，否则可能造成严重损坏，请在上电之前再次确认电源配线。本体上之接地端子 (⊕) 务必正确的接地，可提高产品抗噪声能力。

2

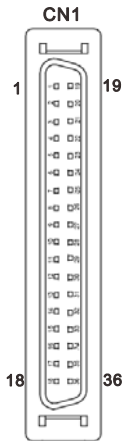
2.2.1 I/O 连接线及配线模块

1. AH20MC-5A 外部装置图



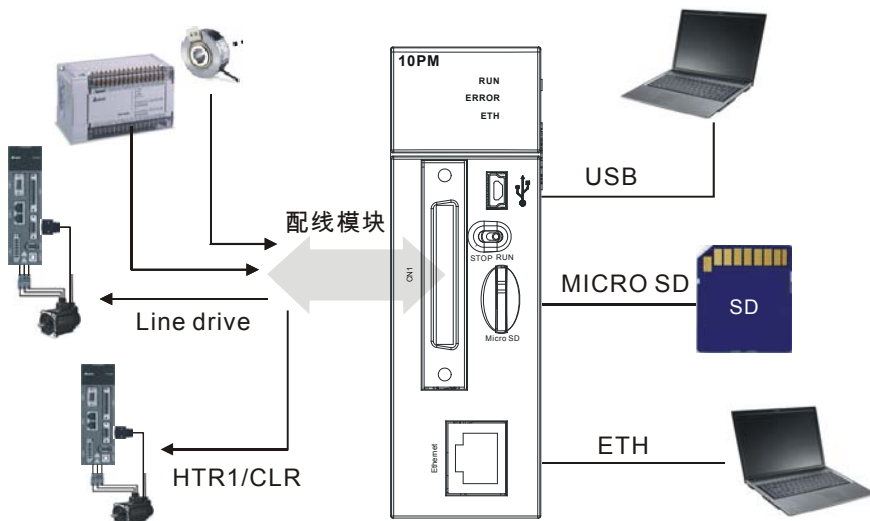
2. AH20MC-5A CN1 引脚

2

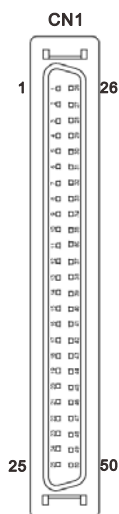


引脚	端子	功能		引脚	端子	功能	
		脉冲	计数			脉冲	计数
1	C3	.	COM3	19	Y0.11	.	Out3
2	C2	.	COM2	20	Y0.10	.	Out2
3	C1	.	COM1	21	Y0.9	.	Out1
4	C0	.	COM0	22	Y0.8	.	Out0
5	.	.	.	23	.	.	.
6	.	.	.	24	.	.	.
7	X0.3-	.	Rst3-/ Rst5-	25	X0.3+	.	Rst3+/ Rst5+
8	X0.15-	DOG3-	CntB3-/ CntB5+	26	X0.15+	DOG3+	CntB3+/CntB5+
9	X0.14-	DOG2-	CntA3-/ CntA5+	27	X0.14+	DOG2+	CntA3+/CntA5+
10	X0.2-	.	Rst2-/ Rst4-	28	X0.2+	.	Rst2+/ Rst4+
11	X0.13-	DOG1-	CntB2-/ CntB4-	29	X0.13+	DOG1+	CntB2+/CntB4+
12	X0.12-	DOG0-	CntA2-/ CntA4-	30	X0.12+	DOG0+	CntA2+/CntA4+
13	X0.1-	.	Rst1-	31	X0.1+	.	Rst1+
14	X0.11-	DOG5-	CntB1-	32	X0.11+	DOG5+	CntB1+
15	X0.10-	DOG4-	CntA1-	33	X0.10+	DOG4+	CntA1+
16	X0.0-	.	Rst0-	34	X0.0+	.	Rst0+
17	X0.9-	MPGB-	CntB0-	35	X0.9+	MPGB+	CntB0+
18	X0.8-	MPGA-	CntA0-	36	X0.8+	MPGA+	CntA0+

3. AH10PM-5A 外部装置图



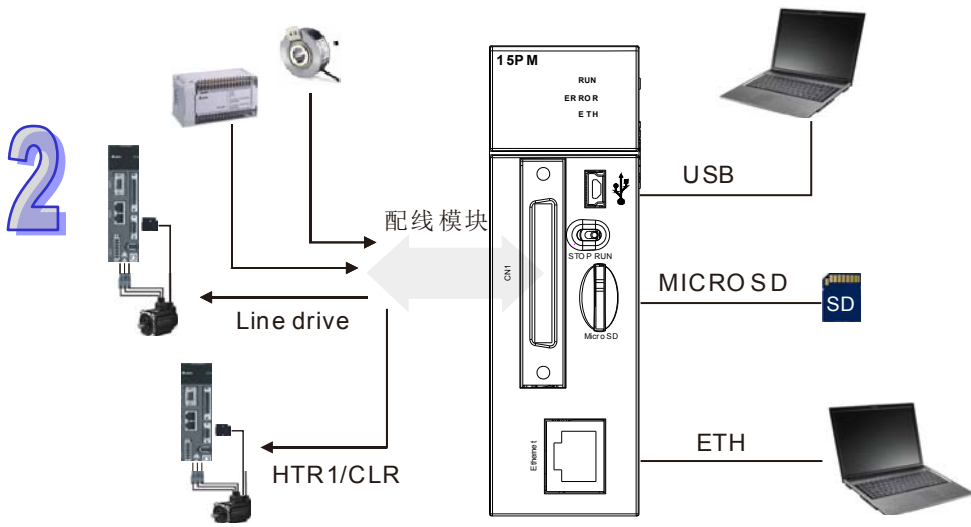
4. AH10PM-5A CN1 引脚



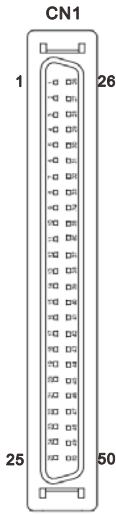
引脚	端子	功能		引脚	端子	功能	
		脉冲	计数			脉冲	计数
1	C3	COM3		26	Y0.11	CLR3/B5	
2	C2	COM2		27	Y0.10	CLR2/A5	
3	C1	COM1		28	Y0.9	CLR1/B4	
4	C0	COM0		29	Y0.8	CLR0/A4	
5	NC	.	.	30	NC	.	
6	Y0.7-	B3-		31	Y0.7+	B3+	
7	Y0.6-	A3-		32	Y0.6+	A3+	
8	Y0.5-	B2-		33	Y0.5+	B2+	
9	Y0.4-	A2-		34	Y0.4+	A2+	
10	Y0.3-	B1-		35	Y0.3+	B1+	
11	Y0.2-	A1-		36	Y0.2+	A1+	
12	Y0.1-	B0-/CLR5-		37	Y0.1+	B0+/CLR5+	
13	Y0.0-	A0-/CLR4-		38	Y0.0+	A0+/CLR4+	
14	NC	.	.	39	NC	.	
15	NC	.	.	40	S/S	S/S	S/S
16	X0.15	DOG3	CntB3/CntB5	41	X0.14	DOG2	CntB3/CntA5
17	X0.13	DOG1	CntB2/CntB4	42	X0.12	DOG0	CntA2/CntA4
18	X0.11	DOG5	CntB1	43	X0.10	DOG4	CntA1
19	X0.9	MPGB	CntB0	44	X0.8	MPGA	CntA0
20	NC	.	.	45	NC	.	.
21	NC	.	.	46	NC	.	.
22	X0.3-	Pg3-	Rst3-/Rst5-	47	X0.3+	Pg3+	Rst3+/Rst5+
23	X0.2-	Pg2-	Rst2-/Rst4-	48	X0.2+	Pg2+	Rst2+/Rst4+
24	X0.1-	Pg1-	Rst1-	49	X0.1+	Pg1+	Rst1+
25	X0.0-	Pg0-	Rst0-	50	X0.0+	Pg0+	Rst0+

2

5. AH15PM-5A 外部装置图



6. AH15PM-5A CN1 引脚

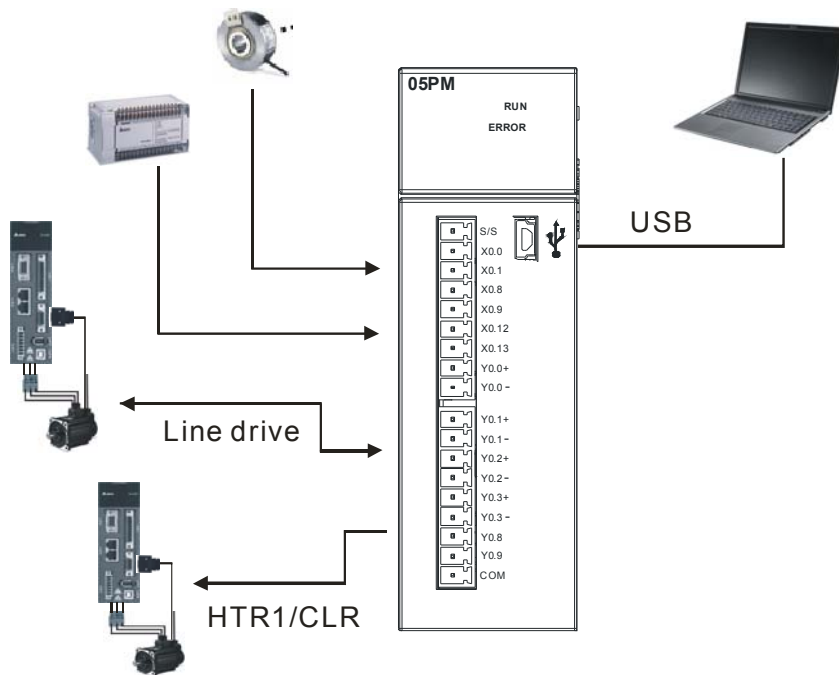


引脚	端子	功能		引脚	端子	功能	
		脉冲	计数			脉冲	计数
1	Y0.11	CLR3	-	26	Y0.10	CLR2	-
2	Y0.9	CLR1	-	27	Y0.8	CLR0	-
3	COM	COM	-	28	Y0.7+	B3+	-
4	Y0.7-	B3-	-	29	Y0.6+	A3+	-
5	Y0.6-	A3-	-	30	Y0.5+	B2+	-
6	Y0.5-	B2-	-	31	Y0.4+	A2+	-
7	Y0.4-	A2-	-	32	Y0.3+	B1+	-
8	Y0.3-	B1-	-	33	Y0.2+	A1+	-
9	Y0.2-	A1-	-	34	Y0.1+	B0+	-
10	Y0.1-	B0-	-	35	Y0.0+	A0+	-
11	Y0.0-	A0-	-	36	S/S	S/S	S/S
12	X1.5	CHG3	-	37	X1.4	CHG2	-
13	X1.3	CHG1	-	38	X1.2	CHG0	-
14	X1.1	LSN3	-	39	X1.0	LSP3	-
15	X0.15	LSN2	CntB3/CntB5	40	X0.14	LSP2	CntB3/CntA5
16	X0.13	LSN1	CntB2/CntB4	41	X0.12	LSP1	CntA2/CntA4
17	X0.11	LSN0	CntB1	42	X0.10	LSP0	CntA1
18	X0.9-	MPGB-	CntB0-	43	X0.9+	MPGB+	CntB0+
19	X0.8-	MPGA-	CntA0-	44	X0.8+	MPGA+	CntA0+
20	X0.7	DOG3	-	45	X0.6	DOG2	-
21	X0.5	DOG1	-	46	X0.4	DOG0	-
22	X0.3-	Pg3-	Rst3-/Rst5-	47	X0.3+	Pg3+	Rst3+/Rst5+
23	X0.2-	Pg2-	Rst2-/Rst4-	48	X0.2+	Pg2+	Rst2+/Rst4+
24	X0.1-	Pg1-	Rst1-	49	X0.1+	Pg1+	Rst1+
25	X0.0-	Pg0-	Rst0-	50	X0.0+	Pg0+	Rst0+

2

7. AH05PM-5A 外部装置图

2



8. AH05PM-5A 引脚

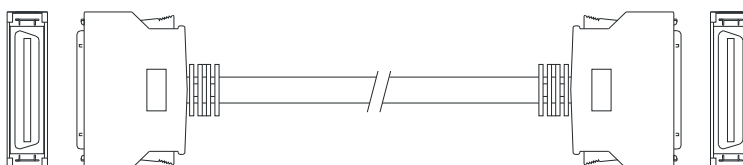
端子	功能		端子	功能	
	脉冲	计数		脉冲	计数
S/S	S/S	S/S	Y0.1+	B0+	-
X0.0	PG0	Rst0	Y0.1-	B0-	-
X0.1	PG1	-	Y0.2+	A1+	-
X0.8	MPGA	CntA0	Y0.2-	A1-	-
X0.9	MPGB	CntB0	Y0.3+	B1+	-
X0.12	DOG0	-	Y0.3-	B1-	-
X0.13	DOG1	-	Y0.8	CLR0	-
Y0.0+	A0+	-	Y0.9	CLR1	-
Y0.0-	A0-	-	COM	-	-

运动控制模块的 CN1 使用 I/O 连接线与配线模块连接，可将所需的线路配置在配线模块上的输入/输出端子台。

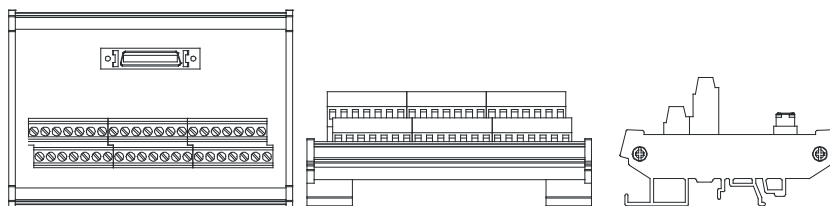
■ I/O 连接线 DVPACAB7D10/ DVPACAB7E10

DVPACAB7D10 (36Pin) : AH04HC-5A 与 AH20MC-5A 连接线

DVPACAB7E10 (50Pin) : AH10PM-5A 连接线

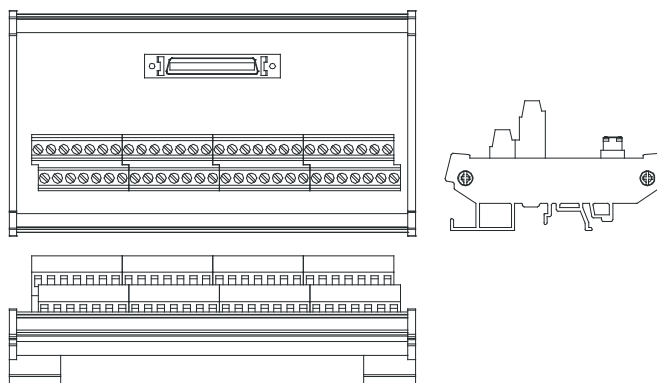


■ AH20MC-5A 配线模块 DVPAETB-IO16C 与端子配置图



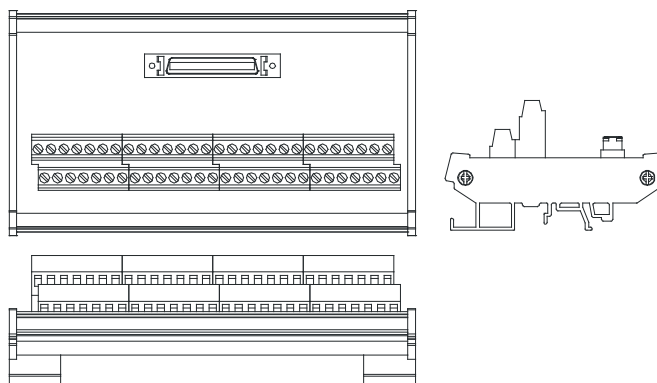
C3	C2	C1	C0	NC	NC	X0.3-	X0.15-	X0.14-	X0.2-	X0.13-	X0.12-	X0.1-	X0.11-	X0.10-	X0.0-	X0.9-	X0.8-	24G	24G	FE
Y0.11	Y0.10	Y0.9	Y0.8	NC	NC	X0.3+	X0.15+	X0.14+	X0.2+	X0.13+	X0.12+	X0.1+	X0.11+	X0.10+	X0.0+	X0.9+	X0.8+	NC	24V	24V

■ AH10PM-5A 配线模块 DVPAETB-IO24C 与端子配置图



上左 1	C3	C2	C1	C0	N/C	Y0.7-	Y0.6-	Y0.5-	Y0.4-	Y0.3-	Y0.2-	Y0.1-	Y0.0-	N/C
上左 15	N/C	X0.15	X0.13	X0.11	X0.9	N/C	N/C	X0.3-	X0.2-	X0.1-	X0.0-	24G	24G	FE
下左 1	Y0.11	Y0.10	Y0.9	Y0.8	N/C	Y0.7+	Y0.6+	Y0.5+	Y0.4+	Y0.3+	Y0.2+	Y0.1+	Y0.0+	N/C
下左 15	S/S	X0.14	X0.12	X0.10	X0.8	N/C	N/C	X0.3+	X0.2+	X0.1+	X0.0+	N/C	24V	24V

■ AH15PM-5A 配线模块 DVPAETB-IO34C 与端子配置图



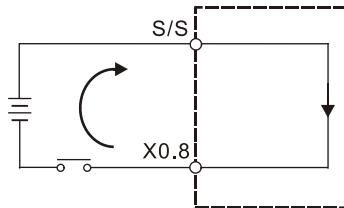
上左 1	Y0.11	Y0.9	COM	Y0.7-	Y0.6-	Y0.5-	Y0.4-	Y0.3-	Y0.2-	Y0.1-	Y0.0-	X1.5	X1.3	X1.1
上左 15	X0.15	X0.13	X0.11	X0.9-	X0.8-	X0.7	X0.5	X0.3-	X0.2-	X0.1-	X0.0-	24G	24G	FE
下左 1	Y0.10	Y0.8	Y0.7+	Y0.6+	Y0.5+	Y0.4+	Y0.3+	Y0.2+	Y0.1+	Y0.0+	S/S	X1.4	X1.2	X1.0
下左 15	X0.14	X0.12	X0.10	X0.9+	X0.8+	X0.6	X0.4	X0.3+	X0.2+	X0.1+	X0.0+	N/C	24V	24V

2.2.2 输入接点配线

输入点之入力信号为直流电源 DC 输入，DC 型式共有两种接法：

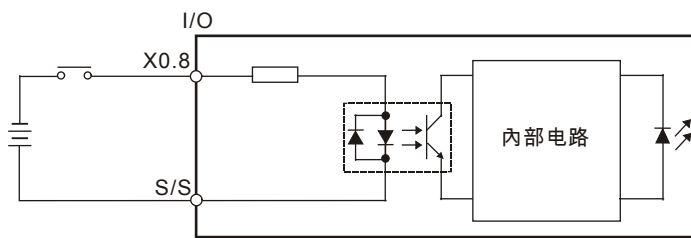
1. SINK 定义如下：

直流形式 (DC Signal IN)



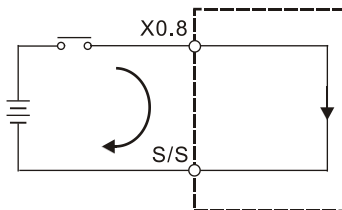
SINK 模式 (电流流入共享端 S/S)

输入点回路等效电路



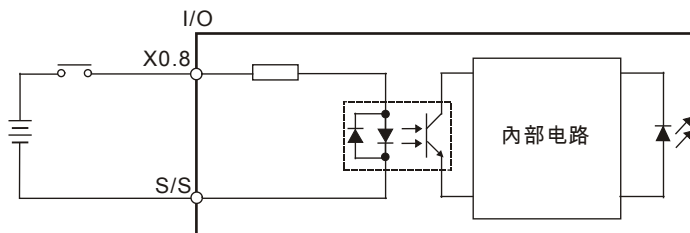
2. SOURCE 定义如下：

直流形式 (DC Signal IN)



SOURCE 模式 (电流流出共享端 S/S)

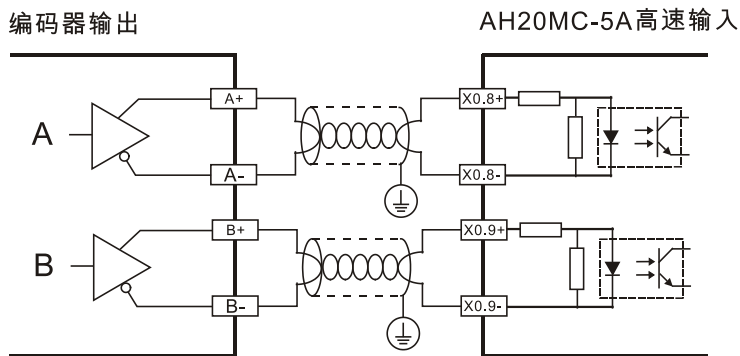
输入点回路等效电路



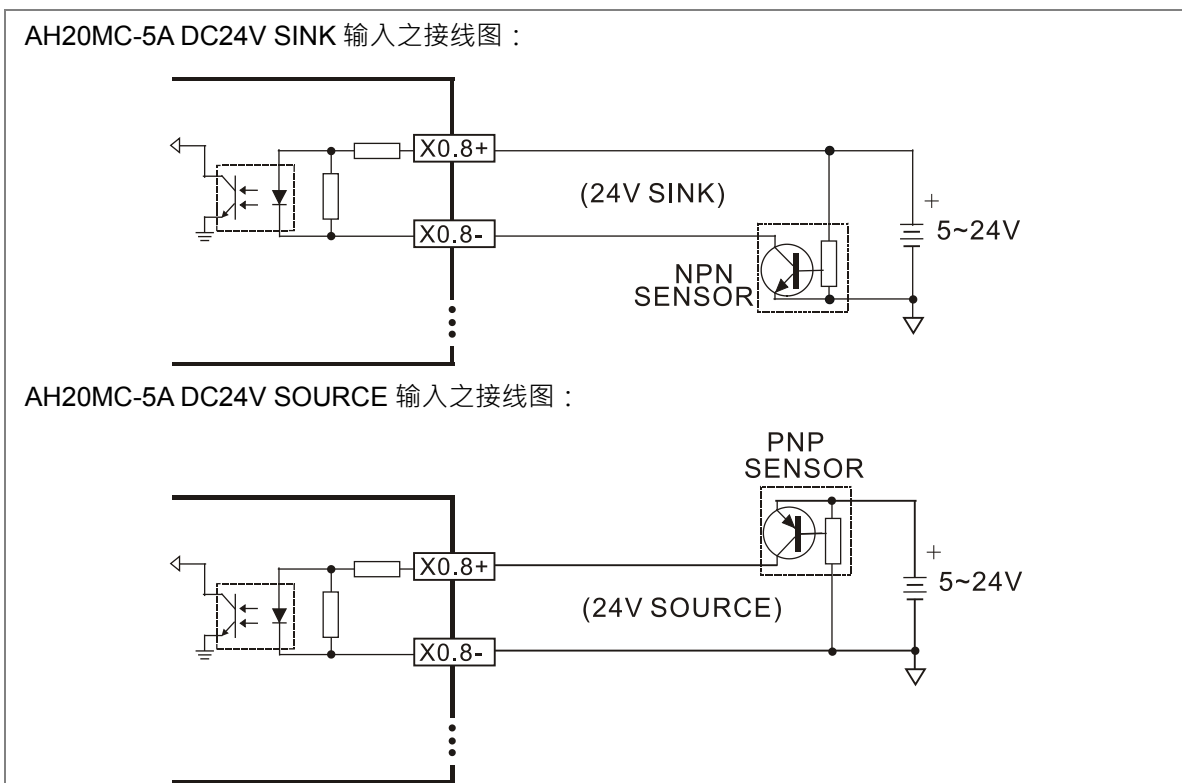
3. 差动输入之配线

以 AH20MC-5A 为例,其 X0.8+~X0.15+及 X0.8~-X0.15-均为 DC5~24V 高速输入电路。此电路最高工作频率可达 200kHz,主要使用在连接差动(双线式) Line Driver 输出电路。

差动输入之接线图(高速、高噪声时使用):



若工作环境频率不高(小于 50kHz)且噪声较低,亦可使用 DC5~24V SINK/SOURCE 之单端输入方式。以 AH20MC-5A 为例,DC5~24V SINK 与 SOURCE 输入之接线图,如下所示:

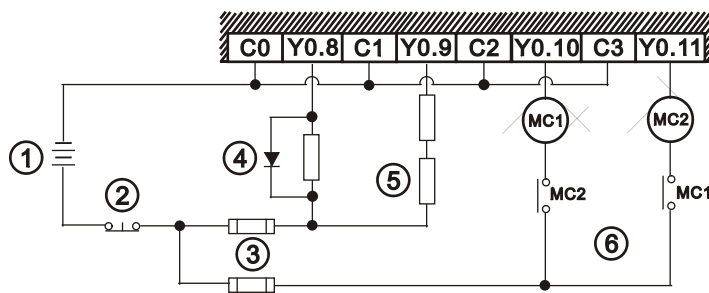
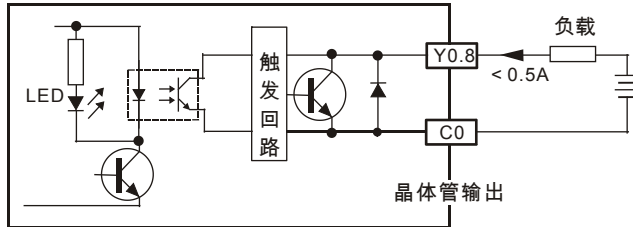


2

2.2.3 输出接点配线

1. 晶体管输出回路配线

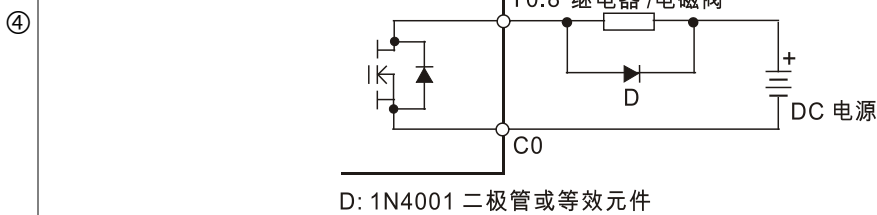
2



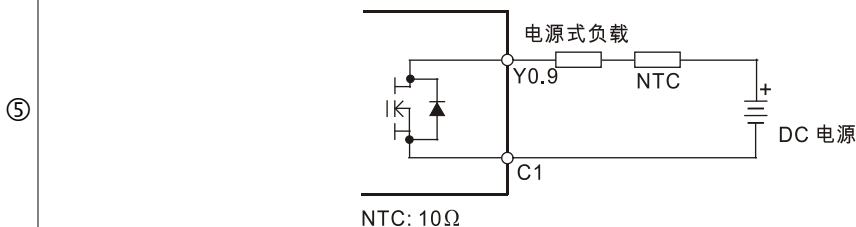
- | | | | | | |
|---|--------|---|------|---|------------|
| ① | 直流电源供应 | ② | 紧急停止 | ③ | 电路回路保护用保险丝 |
|---|--------|---|------|---|------------|

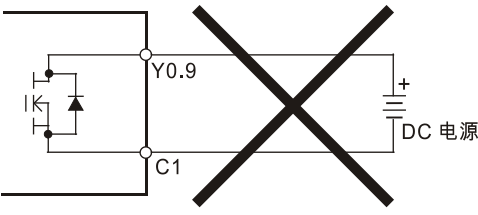
因晶体管模块输出均为集电极输出 (Open Collector) · 若 Y0.8 设定为脉冲输出 · 为确保晶体管模块能动作正常 · 必须维持经提升电阻的输出电流大于 0.1A。

直流负载使用继电器、电磁阀：并联二极管吸收负载关断时的突波电压。



直流负载使用灯泡 (白炽灯)、电源式负载：串联热敏电阻吸收负载启动时的突波电流。



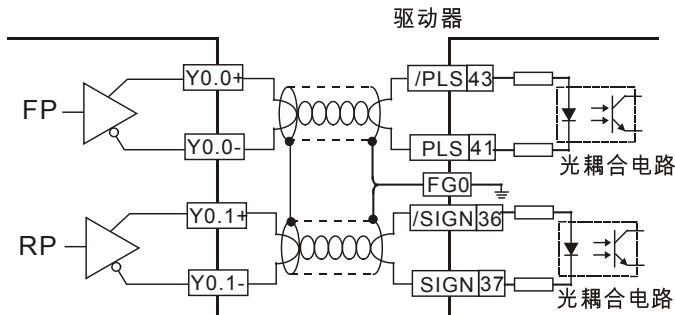
⑤	<p>不可以把输出点 Y0.9 直接接在 I/O 电源上，必须将输出配线接上负载。</p> 
⑥	<p>互斥输出：例如，将 Y0.10 与 Y0.11 用以控制对应马达的正转及反转，使外部电路形成互锁，配合 PLC 内部程序，确保任何异常突发状况发生时，均有安全的保护措施。</p>

2

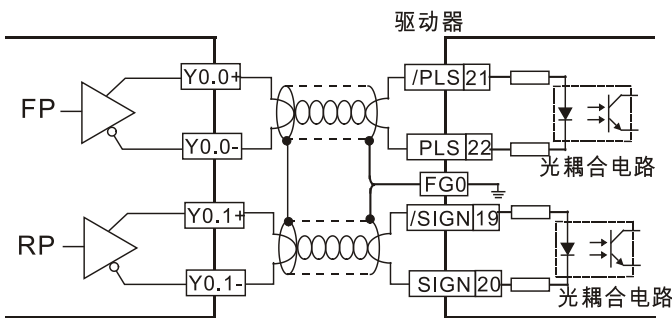
2. 差动输出之配线图

以下以 AH10PM-5A 为例，说明差动输出之配线。

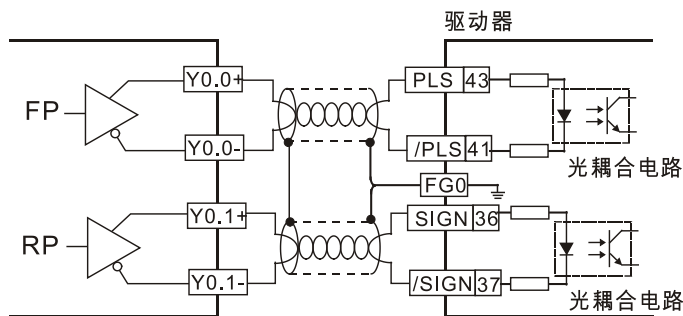
- A10PM-5A 差动输出与 ASDA-A、ASDA-A+及 ASDA-A2 系列驱动器



- 10PM-5A 差动输出与 ASDA-B 系列驱动器



- 10PM-5A 差动输出与 ASDA-AB 系列驱动器

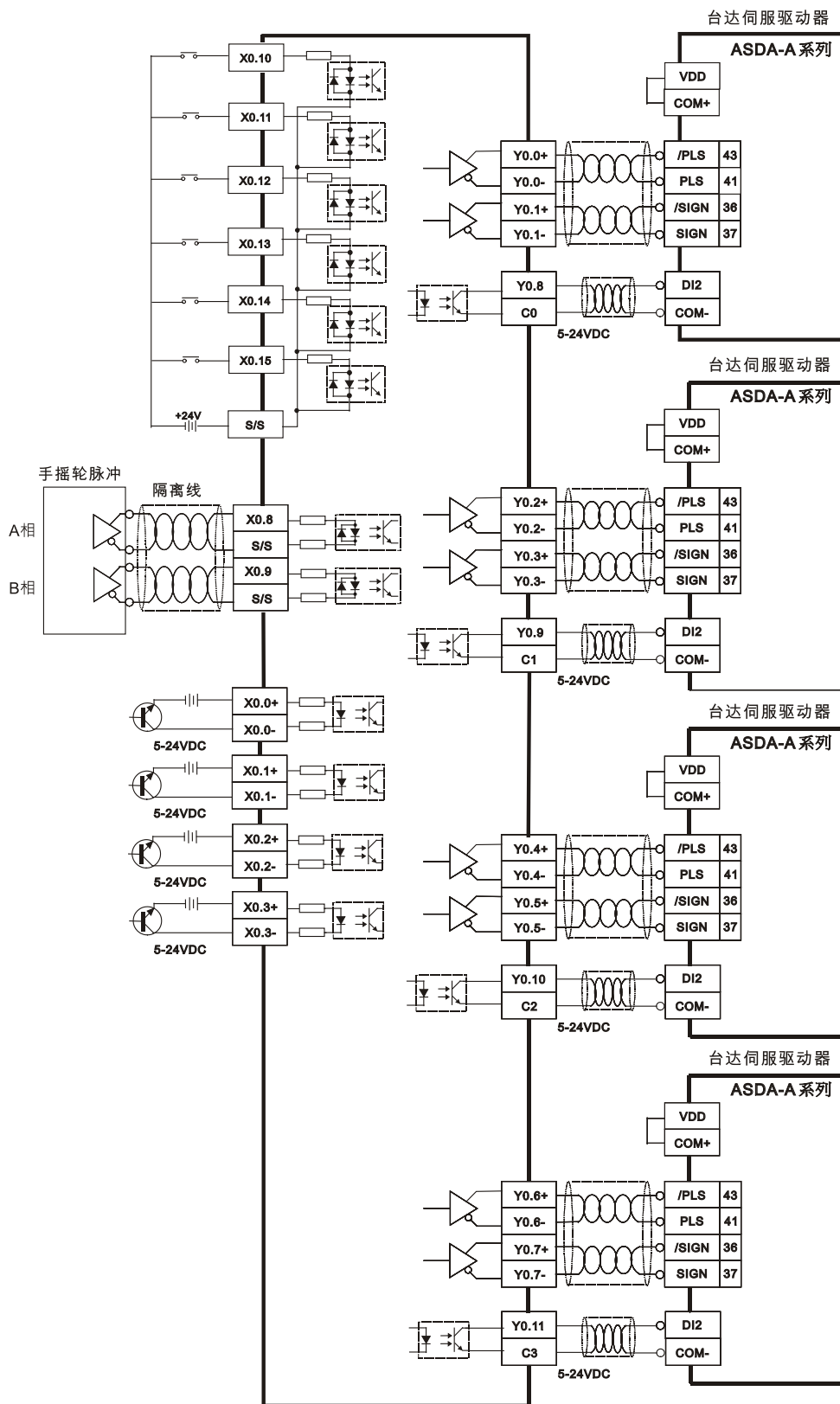


2.2.4 与下位驱动器之配线

以下以 AH10PM-5A 为例，说明与下位驱动器之配线。

2

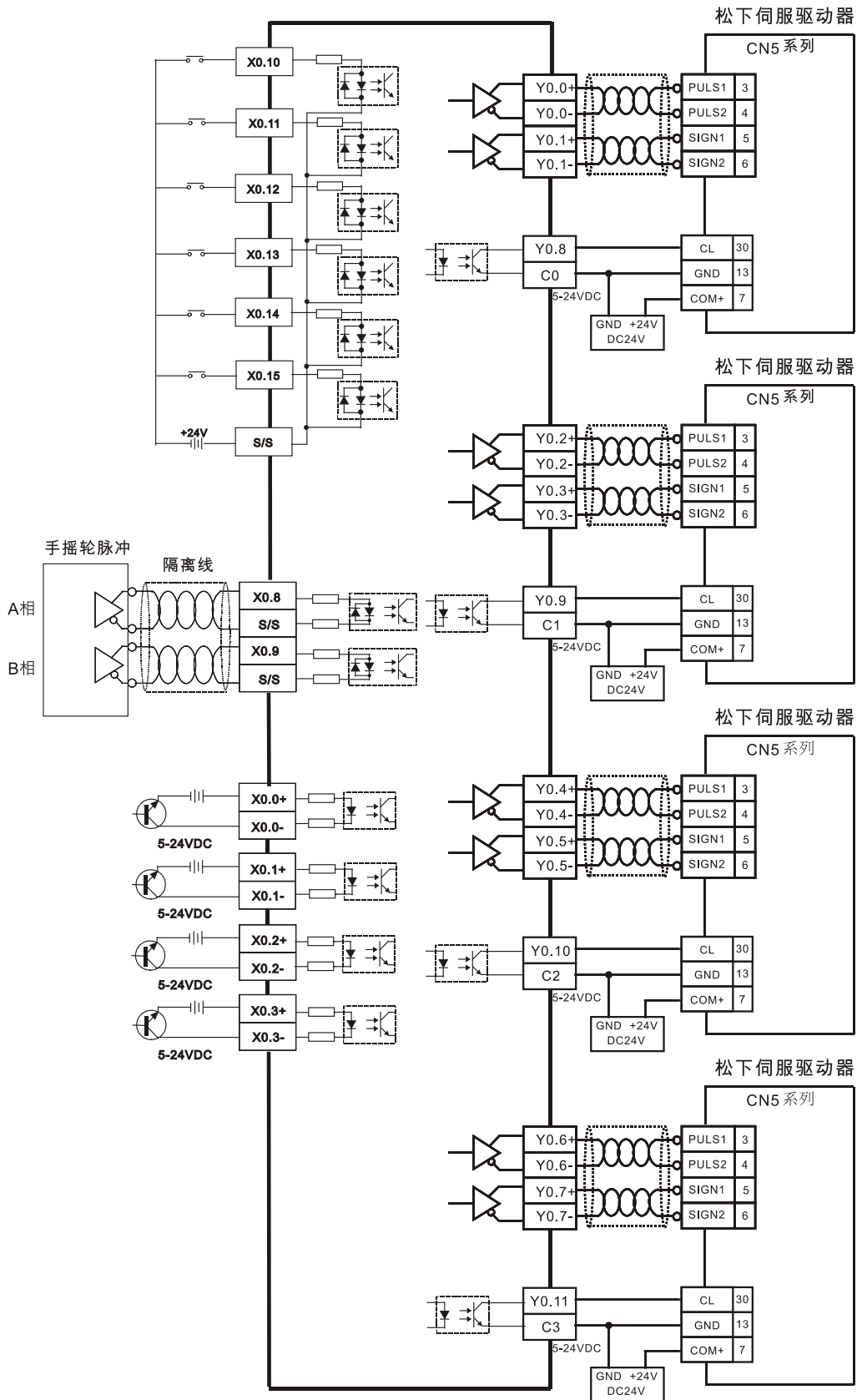
AH10PM-5A 与台达 ASDA-A 系列伺服驱动器之配线图



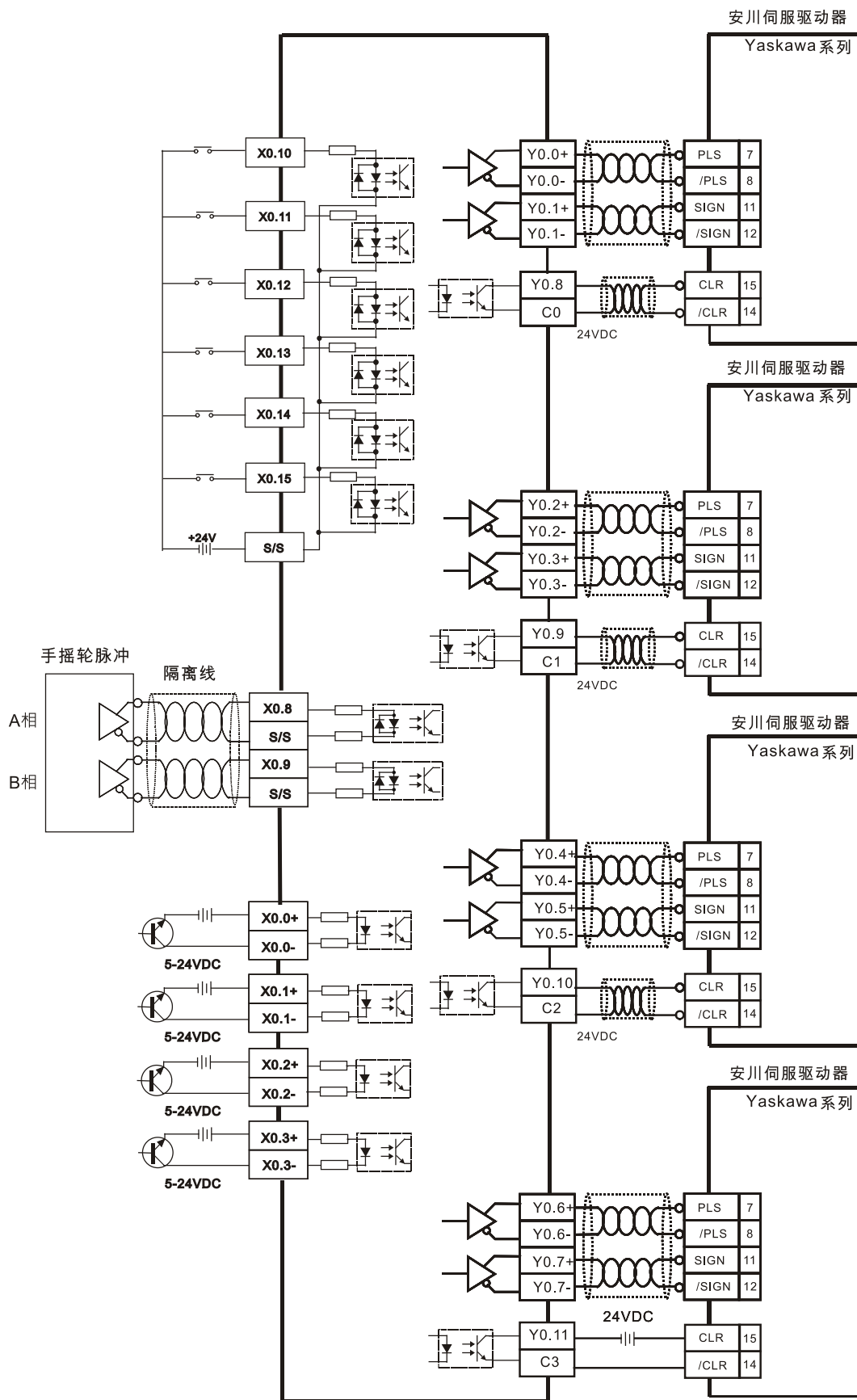
2

AH10PM-5A 与松下 CN5 系列伺服驱动器之配线图

2



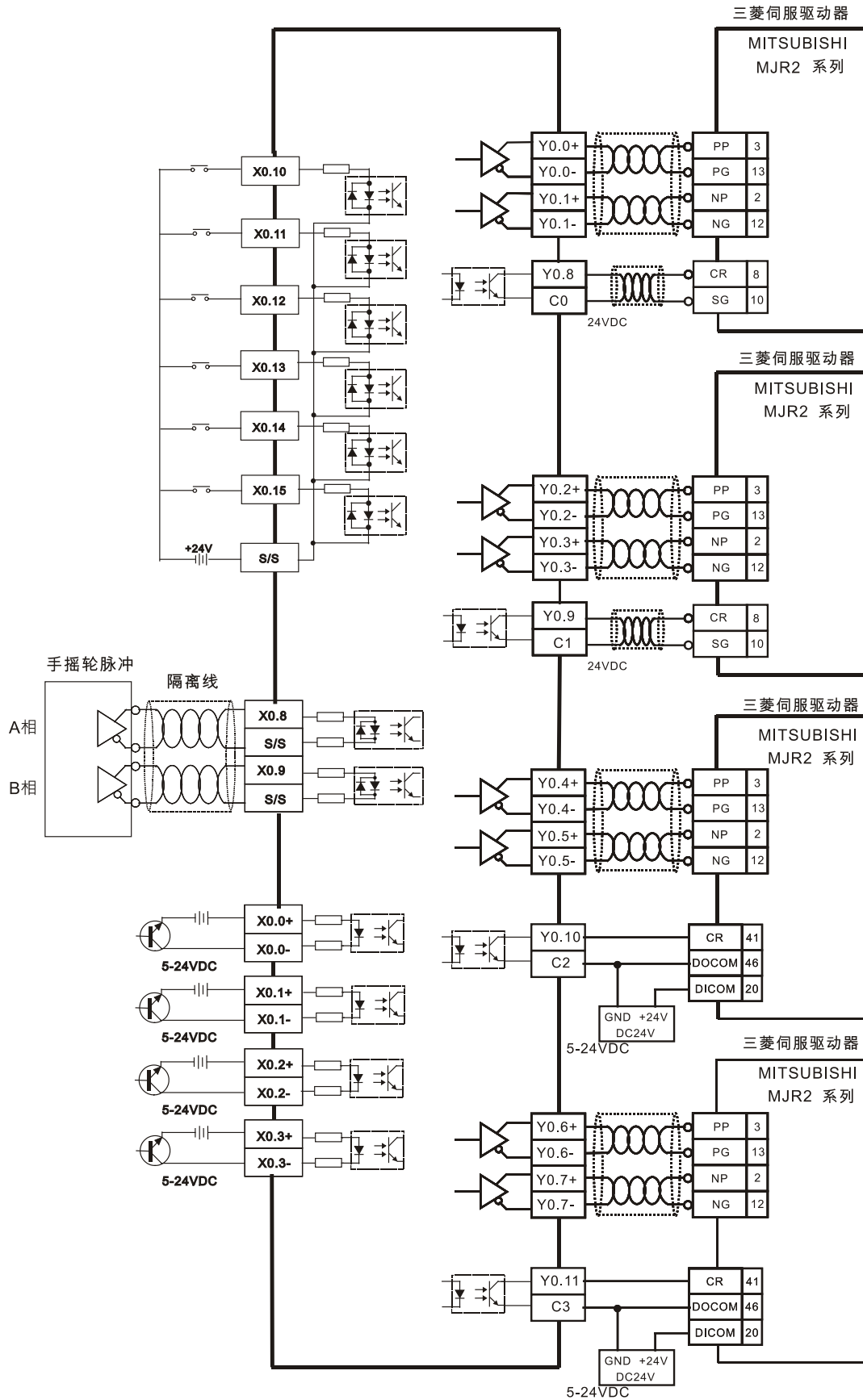
AH10PM-5A 与安川伺服驱动器之配线图



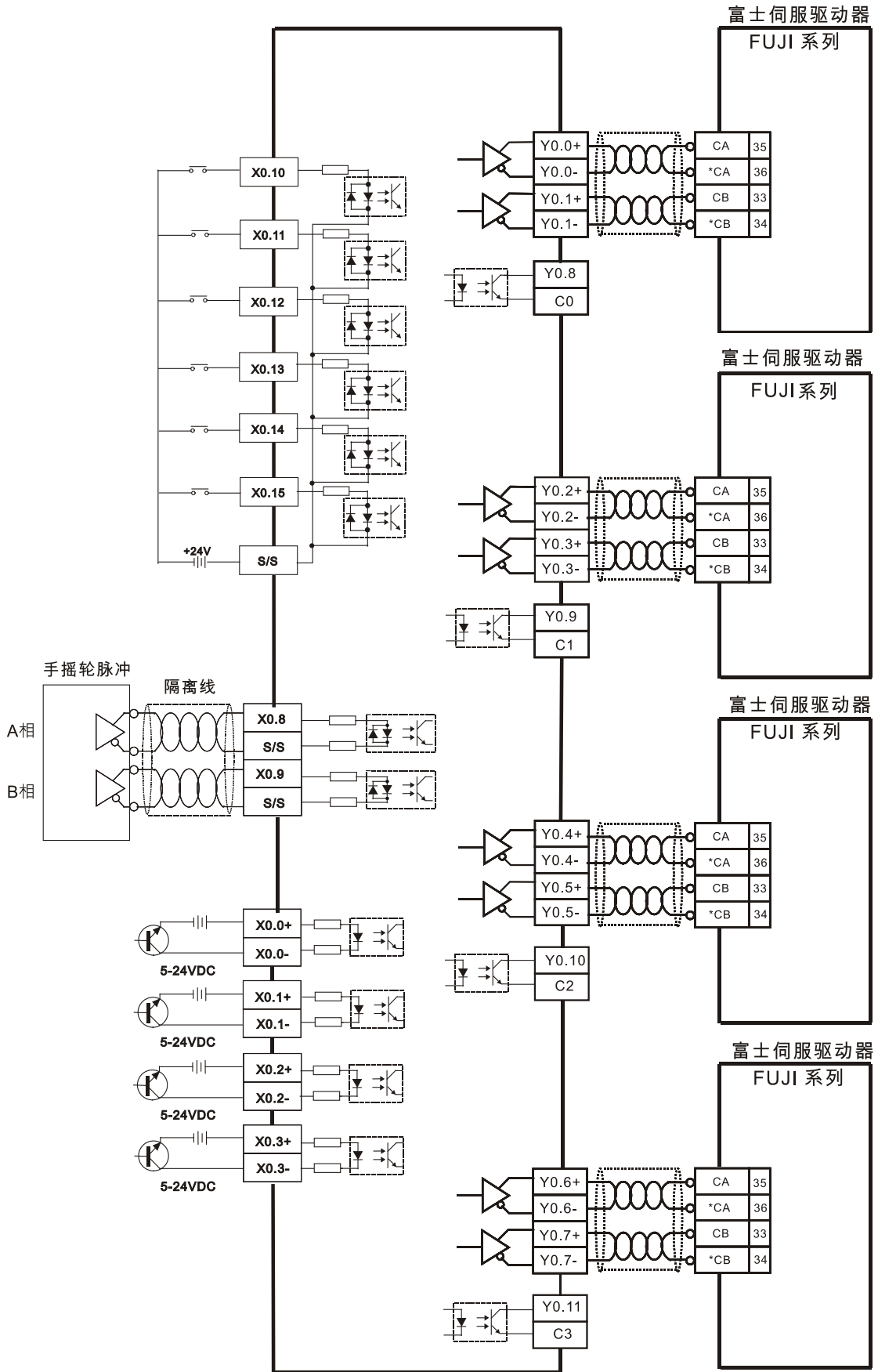
2

AH10PM-5A 与三菱 MJR2 系列伺服驱动器之配线图

2



AH10PM-5A 与富士伺服驱动器之配线图



2

2.3 通讯端口

AH05PM-5A 的通讯端口有内建 Mini USB ; AH20MC-5A、AH10PM-5A 与 AH15PM-5A 的通讯端口有内建 Mini USB 及 Ethernet。另外 AH20MC-5A 内建 DMCNET，请参考以下之说明。

2

Mini USB : USB 通讯端口，可做从站。为主要程序编辑通讯端口，可用于 MODBUS ASCII 或 RTU 模式。

Ethernet : NET 通讯端口，可用于 MODBUS TCP/IP 主站模式。

DMCNET : DMCNET 通讯端口，做为与伺服控制之用，仅 AH-20MC 机种。

通讯架构：

通讯参数 \ 通讯端口	Mini USB
波特率	9,600~57,600 bps
数据位长度	7~8 位
奇偶校验位	偶校验 EVEN/奇校验 ODD/无校验 NONE
停止数据位长度	1~2 数据位
通讯格式设定寄存器	SR36
ASCII 模式	从站有效
RTU 模式	从站有效
读写数据长度(ASCII 模式)	100 个寄存器
读写数据长度 (RTU 模式)	100 个寄存器
通讯参数 \ 通讯端口	Ethernet
传输速率	10/100M bps
数据类型	MODBUS TCP
读写数据长度	100 个寄存器
通讯参数 \ 通讯端口	DMCNET
波特率	10M bps (A/B 两通道)
数据类型	DMCNET 封包格式
最大支持轴数	12 轴

Mini USB 通讯端口的预设通讯格式

- MODBUS ASCII 模式
- 7 个数据位
- 1 个停止位
- 偶校验位 (EVEN)
- 9600bps 波特率

Mini USB

1. 通讯端口 Mini USB 主要为提供 AH500 运动模块程序上/下传，其支持 MODBUS 通讯格式 (ASCII/RTU 模式)，且通讯速度为 9,600~57,600 bps。

Ethernet

1. 通讯端口 Ethernet 为 RJ45 接口之通讯端口，主要为提供从站通讯联机，其支持 MODBUS TCP/IP 通讯格式，且通讯速度为 10/100M bps。
2. Ethernet 通讯除 MODBUS TCP 通讯外，还提供 AH500 运动模块程序上/下传与监控用。

DMCNET (仅 AH20MC-5A 支援)

1. 通讯端口 DMCNET 为 RJ45 接口之通讯端口，主要为提供与台达网络型伺服驱动器联机，其支持 DMCNET 通讯格式，且通讯速度为每一信道 10M bps，同时两通道联机。
2. 此通讯端口做为控制台达网络型伺服驱动器使用。
3. 台达 DMCNET 电缆型号如下：TAP-CB03/05/10/20/30/100。

MEMO

2

3

第3章 各种装置功能

目录

3.1	模块各装置编号一览表	3-2
3.2	数值、常数[K]/[H]、浮点数[F]	3-3
3.3	外部输入[X]/输出[Y]接点的编号及功能	3-5
3.4	辅助继电器[M]的编号及功能	3-7
3.5	特殊辅助继电器[SM]的编号及功能	3-7
3.6	内部继电器[S]的编号及功能	3-7
3.7	定时器[T]的编号及功能	3-8
3.8	计数器[C]的编号及功能	3-9
3.9	寄存器[D][V][Z]的编号及功能	3-14
3.9.1	数据寄存器[D]	3-14
3.9.2	变址用寄存器[V][Z]	3-14
3.10	特殊寄存器[SR]的编号及功能	3-15
3.11	指针[P]的编号及功能	3-15
3.12	特殊继电器[SM]及特殊寄存器[SR]	3-16
3.12.1	特殊继电器[SM]	3-16
3.12.2	特殊寄存器[SR]	3-20
3.13	特殊继电器[SM]及特殊寄存器[SR]功能说明	3-26
3.14	运动模式特殊寄存器[SR]	3-39

3.1 模块各装置编号一览表

■ 装置列表

形式	装置名称		装置数	范围
位装置	输入装置	X	256	X0.0~X15.15
	输出装置	Y	256	Y0.0~Y15.15
	辅助继电器	M	4096	M0~M4095
	特殊辅助标志	SM	16384	SM0~SM16383
	步进点继电器	S	1024	S0~S1023
字符装置	输入装置	X	16	X0~X15
	输出装置	Y	16	Y0~Y15
	数据寄存器	D	10000	D0~D9999
	特殊数据寄存器	SR	16384	SR0~SR16383
	定时器	T	256	T0~T255
	16 位计数器	C	200	C0~C199
	32 位计数器	C	16	C240~C255
	32 位高速计数器	C	6	C200 · C204 · C208 · C212 · C216 · C220 (各机种详细范围, 请见第 3.8 节)
	变址寄存器	V	6	V0~V5
Z		8	Z0~Z7	
指针装置	指针	P	256	P0~P255
常数	十进制	K	16 位运算: -32768~32767 32 位运算: -2147483648~2147483647	
	十六进制	H	16 位运算: H0~HFFFF 32 位运算: H0~HFFFFFFFF	
	浮点数	F	32 位运算: $\pm 1.17549435^{-38} \sim \pm 3.40282347^{+38}$	

3.2 数值、常数[K]/[H]、浮点数[F]

常数	K	10 进制	16 位运算：-32768~32767 32 位运算：-2147483648~2147483647
	H	16 进制	16 位运算：H0~HFFFF 32 位运算：H0~HFFFFFFF
浮点数	F	32-bit 的长度	32 位运算： $\pm 1.17549435 \cdot 38 \sim \pm 3.40282347 \cdot 38$

AH500 运动定位模块内部依据各种不同控制目的，共使用 3 种数值类型执行运算的工作，各种数值的任务及功能说明如下：

1. 二进制 (Binary Number · BIN)

模块内部之数值运算或储存均采用二进制，二进制数值及相关术语如下：

位 (Bit)： 位为二进制数值之最基本单位，其状态非 1 即 0。

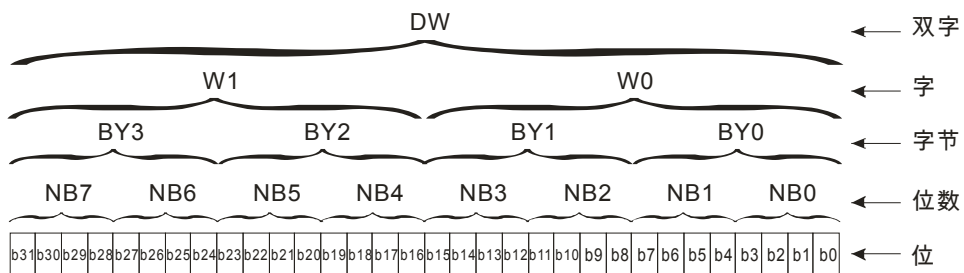
位数 (Nibble)： 由连续的 4 个位所组成 (如 $b_3 \sim b_0$) 可用以表示一个位数之 10 进制数字 0~9 或 16 进制之 0~F

字节 (Byte)： 是由连续之两个位数所组成 (亦即 8 位， $b_7 \sim b_0$)。可表示 16 进制之 00~FF

字 (Word)： 是由连续之两个字节所组成 (亦即 16 位， $b_{15} \sim b_0$) 可表示 16 进制之 4 个位数值 0000~FFFF

双字 (Double Word)： 是由连续之两个字符组所组成 (亦即 32 位， $b_{31} \sim b_0$)。可表示 16 进制之 8 个位数值 00000000~FFFFFFFF

二进制系统中位、位数、字节、字、及双字的关系如下图所示：



2. 十进制 (Decimal Number , DEC) , 十进制在模块系统应用的时机如下 :

- 做为定时器 T、计数器 C 等的设定值 , 例 : TMR T0 K50 (K 常数)
- S、M、SM、T、C、D、SR、V、Z、P 等装置的编号 , 例 : M10、T30 (装置编号)
- 在应用指令中做为操作数使用 , 例 : MOV K123 D0 (K 常数)
- 外部输入 : X0.0~X0.15、X1.0~ X 1.15... (装置编号)
- 外部输出 : Y0.0~Y0.15、Y1.0~Y1.15... (装置编号)

3. 十六进制 (Hexadecimal Number , HEX) , 十六进制在 AH500 运动模块系统应用的时机如下 :

- 在应用指令中做为操作数使用 , 例 : MOV H1A2B D0 (H 常数)

常数 K :

十进制数值在 AH500 运动定位模块系统中 , 通常会在数值前面以一“K”字表示 , 例 : K100 , 表示为十进制 , 其数值大小为 100 。

例外 :

当使用 K 再搭配位装置 M、S 可组合成为位数、字节、字符组或双字符组形式的数据。

例 : K4M100。在此 K1 代表一个 4 bits 的组合 , K2~K4 分别代表 8、12 及 16 bits 的组合。

常数 16# :

16 进位数值在 AH500 运动定位模块中 , 通常在其数值前面以一“16#”字符表示 , 例 : 16#100 , 其表示为 16 进位 , 数值大小为 256 。

浮点数 F :

浮点数值在 AH500 运动定位模块中 , 通常在其数值前面以一“F”字符表示 , 例 : F3.123 , 其表示为浮点数值 , 数值大小为 3.123 。



3.3 外部输入[X]/输出[Y]接点的编号及功能

1. 输入 X 装置：

输入 X 装置其编号采用十进制编码，输入 X 装置的编号固定从 X0.0 开始算，依照各运动控制模块输入 X 装置，编号定义略有不同，而最大输入点数可达 256 点，其范围如下：X0.0~X15.15。

2. 输出 Y 装置：

输出 Y 装置其编号亦采用十进制编码，输出 Y 装置的编号固定从 Y0.0 开始算，依照各运动控制模块输出 Y 装置，编号定义略有不同，而最大输出点数可达 256 点，其范围如下：Y0.0~Y15.15。

3. 输入 X 装置的功能：

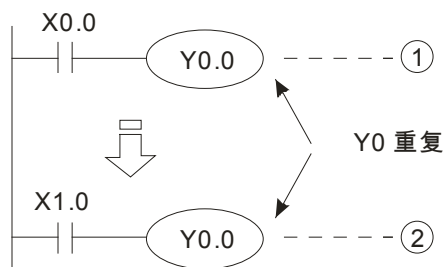
输入 X 装置的功能：输入 X 装置与输入装置连接，读取输入信号进入 AH500 运动控制模块。每一个输入 X 装置的 A 或 B 接点于程序中使用次数没有限制。输入 X 装置之 ON/OFF 只会跟随输入装置的 ON/OFF 做变化。

4. SM304 标志强制 ON/OFF X 输入点：

SM304=OFF 时，不可使用 PMSoft 来强制输入 X 装置之 ON/OFF，SM304=ON 时，允许 PMSoft 来强制 AH500 运动控制模块输入 X 装置之 ON/OFF，但此时 AH500 运动控制模块外部扫描更新输入点信号功能关闭。

5. 输出 Y 装置的功能：

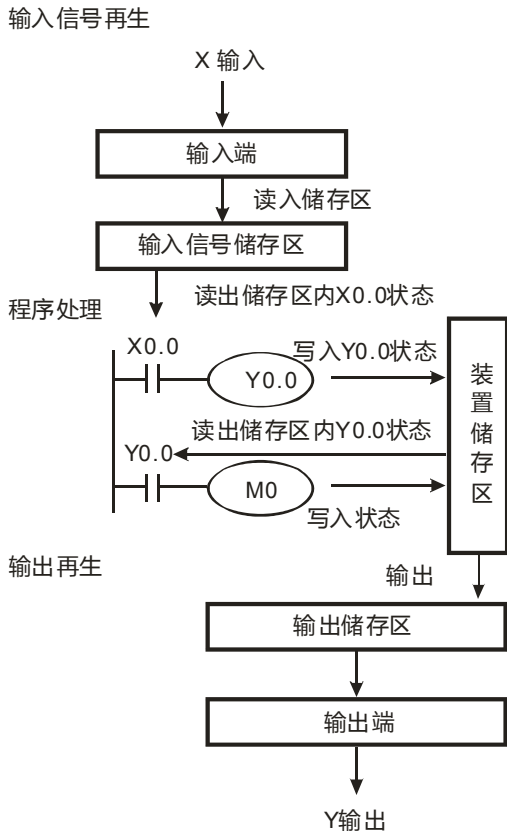
输出 Y 装置的任务就是送出 ON/OFF 信号来驱动连接输出 Y 装置的负载。每一个输出 Y 装置的 A 或 B 接点于程序中使用次数没有限制，但输出线圈 Y 的编号，在程序建议仅能使用一次，否则依程序扫描原理，其输出状态的决定权会落在程序中最后的输出 Y 的电路。



Y0.0 的输出最后会由电路②决定，亦即由 X1.0 的 ON/OFF 决定 Y0.0 的输出。

对于程序的处理流程（结束再生方式）

3



输入信号再生：

1. AH500 运动定位模块在执行程序之前会将外部输入信号的 ON/OFF 状态一次读入至输入信号储存区内。
2. 在程序执行中若输入信号作 ON/OFF 变化，但是输入信号储存区内的状态不会改变，一直到下一次扫描开始时再读入输入信号新的 ON/OFF 状态。
3. 外部信号 ON→OFF 或 OFF→ON 变化到程序内接点认定为 ON/OFF 时期间约有 10ms 的延迟（但可能会受程序扫描时间的影响）。

程序处理：

AH500 运动定位模块读取输入信号储存区内各输入信号之 ON/OFF 状态后开始从程序起始位置依序执行程序中的每一指令，其处理结果即各输出线圈之 ON/OFF 也逐次存入各装置储存区。

输出再生：

1. 当执行到 M102 指令时将装置储存区内 Y 的 ON/OFF 状态送到输出 LATCH 储存区内。
2. ON→OFF 或 OFF→ON 变化到接点 ON/OFF 时期间约有 10ms 的延迟。

3.4 辅助继电器[M]的编号及功能

辅助继电器的编号 (以 10 进制编号): M0~M4095, 共 4096 点。

辅助继电器的功能: 辅助继电器 M 与输出装置 Y 一样有输出线圈及 A、B 接点, 而且于程序当中使用次数无限制, 使用者可利用辅助继电器 M 来组合控制回路, 但无法直接驱动外部负载。当继电器于 AH500 运动模块运转时, 若遇到停电, 其状态将全部被复位为 OFF, 再送电时其状态仍为 OFF。

3.5 特殊辅助继电器[SM]的编号及功能

辅助继电器的编号 (以 10 进制编号): SM0~SM16383, 共 16384 点。

特殊辅助继电器的功能: 特殊辅助继电器 SM 与输出 Y 装置一样有输出线圈及 A、B 接点, 而且于程序当中使用次数无限制, 使用者可利用特殊辅助继电器 SM 来组合控制回路, 但无法直接驱动外部负载。每一个特殊用辅助继电器均有其特定之功用, 未定义的特殊用辅助继电器请勿使用。

3.6 步进点继电器[S]的编号及功能

步进点继电器的编号 (以 10 进制编号): S0~S1023, 共 1024 点。

步进点继电器的功能: 步进点继电器 S 的装置编号为 S0~S1023 共 1024 点, 各步进点继电器 S 与输出 Y 装置一样有输出线圈及 A、B 接点, 而且于程序当中使用次数无限制, 但无法直接驱动外部负载。步进点继电器 S 可当作一般的辅助继电器使用。

3.7 定时器[T]的编号及功能

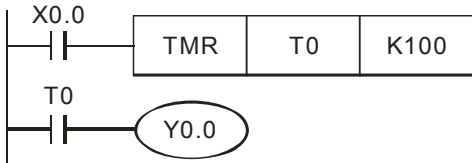
定时器的编号 (以 10 进制编号): T0~T255, 共 256 点。可使用参数设定变更成停电保持区域。

定时器的功能: 定时器是以 10ms 为一个计时单位, 计时方式采上数计时, 当定时器当前值=设定值时输出线圈导通, 设定值为 10 进制 K 值, 亦可使用数据寄存器 D 当成设定值。

定时器之实际设定时间=计时单位*设定值。

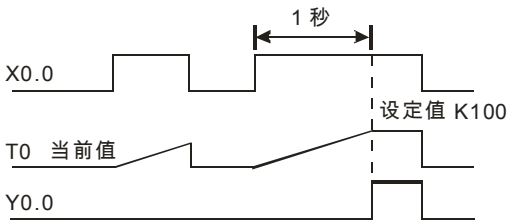
1. 一般用定时器在 TMR 指令执行时计时一次, 在 TMR 指令执行时, 若计时到达, 则输出线圈导通。

3



当 X0.0=ON 时, 定时器 T0 之当前值以 10ms 采上数计时, 当定时器当前值=设定值 K100 时, 输出线圈 T0=ON。

当 X0.0=OFF 或停电时, 定时器 T0 之当前值清除为 0, 输出线圈 T0 变为 OFF。



设定值的指定方法: 定时器之实际设定时间=计时单位*设定值。

- 常数指定 K: 设定值直接指定常数 K 值
- 变址 D: 设定值使用数据寄存器 D 做变址

3.8 计数器[C]的编号及功能

计数器的编号 (以 10 进制编号) :

1. AH20MC-5A 机种

功能说明	范围		备注
16 位上数	C0~C199 · 200 点	合计 216 点	CNT (DCNT) 指令所指定的计数器 · 若计时到达则此同编号 C 的接点将会 ON 。
32 位上/下数	C240~C255 · 16 点 (积分型)		
32 位 高速计数器	C200 · C204 · C208 · C212 · C216 · C220 · 6 点	合计 6 点	C200 输入接点 X0.8+/X0.8-/X0.9+/X0.9- C204 输入接点 X0.10+/X0.10-/X0.11+/X0.11- C208 输入接点 X0.12+/X0.12-/X0.13+/X0.13- C212 输入接点 X0.14+/X0.14-/X0.15+/X0.15- C216 输入接点 X0.12+/X0.12-/X0.13+/X0.13- C220 输入接点 X0.14+/X0.14-/X0.15+/X0.15-

2. AH10PM-5A 机种

功能说明	范围		备注
16 位上数	C0~C199 · 200 点	合计 216 点	CNT (DCNT) 指令所指定的计数器 · 若计时到达则此同编号 C 的接点将会 ON 。
32 位上/下数	C240~C255 · 16 点 (积分型)		
32 位 高速计数器	C200 · C204 · C208 · C212 · C216 · C220 · 6 点	合计 6 点	C200 输入接点 X0.8/X0.9 C204 输入接点 X0.10/X0.11 C208 输入接点 X0.12/X0.13 C212 输入接点 X0.14/X0.15 C216 输入接点 X0.12/X0.13 C220 输入接点 X0.14/X0.15

3

3. AH15PM-5A 机种

功能说明	范围		备注
16 位上数	C0~C199 · 200 点	合计 216 点	CNT (DCNT) 指令所指定的计数器 · 若计时到达则此同编号 C 的接点将会 ON 。
32 位上/下数	C240~C255 · 16 点 (积算型)		
32 位 高速计数器	C200 · C204 · C208 · C212 · C216 · C220 · 6 点	合计 6 点	C200 输入接点 X0.8/X0.9 C204 输入接点 X0.10/X0.11 C208 输入接点 X0.12/X0.13 C212 输入接点 X0.14/X0.15 C216 输入接点 X0.12/X0.13 C220 输入接点 X0.14/X0.15

4. AH05PM-5A 机种

功能说明	范围		备注
16 位上数	C0~C199 · 200 点	合计 216 点	CNT (DCNT) 指令所指定的计数器 · 若计时到达则此同编号 C 的接点将会 ON 。
32 位上/下数	C240~C255 · 16 点 (积分型)		
32 位 高速计数器	C200	合计 1 点	C200 输入接点 X0.8/X0.9

计数器特点：

项目	16 位计数器		32 位计数器	
			一般型	高速型
类型	一般型		一般型	高速型
计数方向	上数		上、下数	
设定值	0~32,767		-2,147,483,648~+2,147,483,647	
设定值的指定	常数 K 或数据寄存器 D		常数 K 或数据寄存器 D (指定 2 个)	
当前值的变化	计数到达设定值就不再计数		计数到达设定值后 · 仍继续计数	
输出接点	计数到达设定值 · 接点导通并保持		上数到达设定值接点导通并保持 ON 下数到达设定值接点复位成 OFF	
复位动作	RST 指令被执行时当前值归零 · 接点被复位成 OFF			
接点动作	在扫描结束时 · 统一动作		在扫描结束时 · 统一动作	计数到达后立即动作 · 与扫描周期无关

计数器的功能：

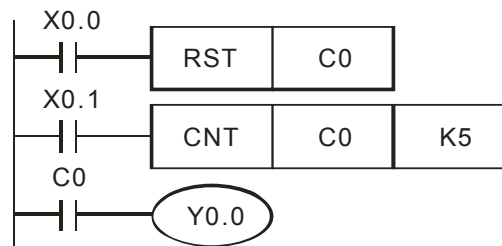
计数器之计数脉冲输入信号由 OFF→ON 时，计数器当前值等于设定值时输出线圈导通，设定值为 10 进制 K 值，亦可使用数据寄存器 D 当成设定值。

16 位计数器：

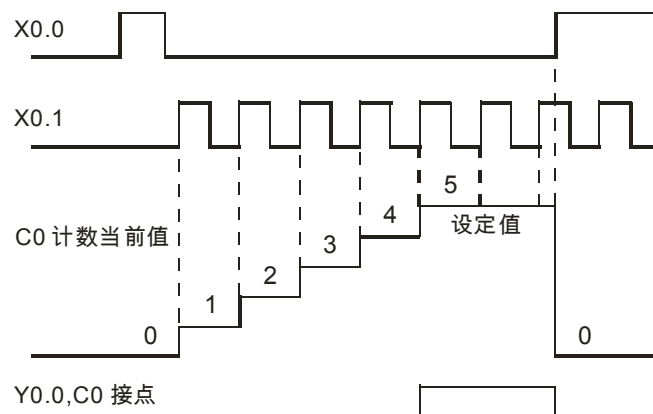
1. 16 位计数器的设定范围：K0~K32,767。(K0 与 K1 相同，在第一次计数时输出接点马上导通)
2. 若使用 MOV 指令、将一个大于设定值的数值传送到 C0 当前值寄存器时，在下次 X1 由 OFF→ON 时，C0 计数器接点即变成 ON，同时当前值内容变成与设定值相同。
3. 计数器之设定值可使用常数 K 直接设定或使用寄存器 D 中之数值作间接设定。
4. 设定值若使用常数 K 仅可为正数，使用数据寄存器 D 做为设定值可以是正负数。计数器当前值由 32,767 再往上累计时则变为-32,768。

范例：

```
LD    X0.0
RST   C0
LD    X0.1
CNT   C0 K5
LD    C0
OUT   Y0.0
```



1. 当 X0.0=ON 时 RST 指令被执行，C0 的当前值归零，输出接点被复位为 OFF。
2. 当 X0.1 由 OFF→ON 时，计数器之当前值将执行上数（加一）的动作。
3. 当计数器 C0 计数到达设定值 K5 时，C0 接点导通，C0 当前值 = 设定值 = K5。之后的 X0.1 触发信号 C0 完全不接受，C0 当前值保持在 K5 处。



32 位一般用加减计数器：

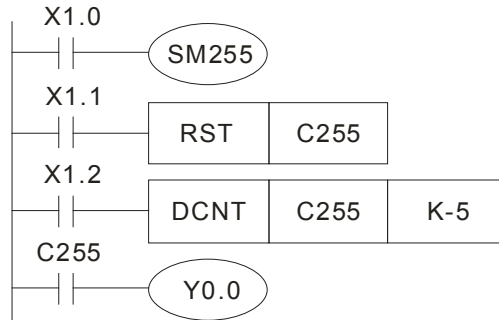
1. 32 位一般用计数器的设定范围：K-2,147,483,648~K2,147,483,647。32 位一般用加减计数器切换上下数用特殊辅助继电器：由 SM240~SM255 来决定。例：SM240=OFF 时决定 C240 为加算，SM240=ON 时决定 C240 为减算其余类推。
2. 设定值可使用常数 K 或使用数据寄存器 D 做为设定值可以是正负数，若使用数据寄存器 D 则一个设定值占用两个连续的数据寄存器。

3. 一般用计数器在模块停电的时候，计数器当前值即被清除，若为停电保持型计数器（积分型），则会将停电前的当前值及计数器接点状态储存着，复电后会继续累计。
4. 计数器当前值由 2,147,483,647 再往上累计时则变为-2,147,483,648。同理计数器当前值由 -2,147,483,648 再往下递减时，则变为 2,147,483,647。

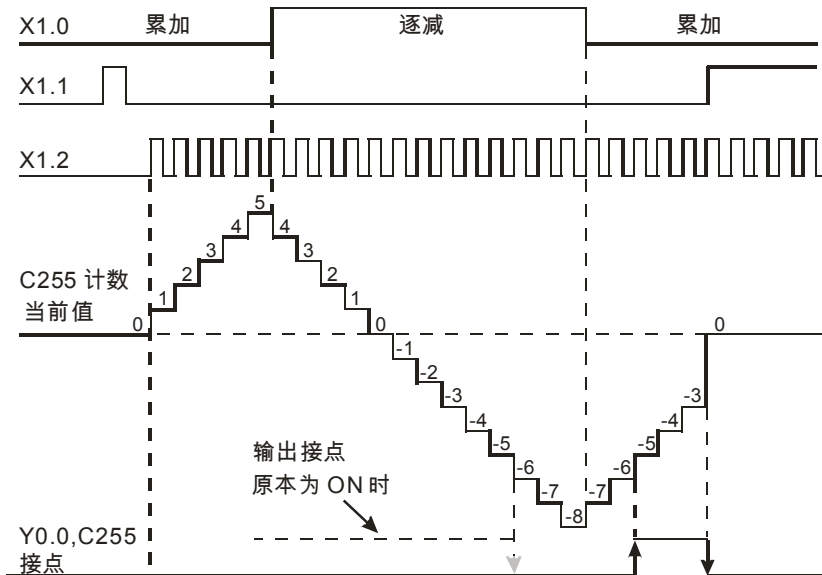
范例：

```

LD    X1.0
OUT   SM255
LD    X1.1
RST   C255
LD    X1.2
DCNT  C255 K-5
LD    C255
OUT   Y0.0
    
```



1. X1.0 驱动 SM255 来决定 C255 为加算或减算。
2. 当 X1.1 由 OFF→ON 时，RST 指令执行，C255 之当前值被清除为 0，且接点变为 OFF。
3. 当 X1.2 由 OFF→ON 时，计数器之当前值将执行上数（加一）的动作或下数（减一）的动作。
4. 当计数器 C255 之当前值从 K-6→K-5 变化时，C255 接点由 OFF→ON。当计数器 C255 之当前值从 K-5→K-6 变化时，C255 接点由 ON→OFF。
5. 若使用 MOV 指令 PMSoft 将一个大于设定值的数值传送到 C255 当前值寄存器时，在下次 X1.1 由 OFF→ON 时，C255 计数器接点即变成 ON，同时当前值内容变成与设定值相同。



32 位高速计数器：

1. 32 位高速计数器的设定范围：K-2,147,483,648~K2,147,483,647。
2. 计数模式设定

计数器 编号	计数模式		清除信 号设定	清除信号 外部接脚	外部输入接脚*1*2
	装置	设定值*3			
C200	K1SM200	0 : U/D 1 : P/D 2 : A/B (1 倍频) 3 : 4A/B (4 倍频)	SM203	X0.0+ 、 X0.0-	X0.8 、 X0.9 、 S/S
C204	K1SM204		SM207	X0.1+ 、 X0.1-	X0.10 、 X0.11 、 S/S
C208	K1SM208		SM211	X0.2+ 、 X0.2-	X0.12 、 X0.13 、 S/S
C212	K1SM212		SM215	X0.3+ 、 X0.3-	X0.14 、 X0.15 、 S/S
C216	K1SM216		SM219	X0.2+ 、 X0.2-	X0.12 、 X0.13 、 S/S
C220	K1SM220		SM223	X0.3+ 、 X0.3-	X0.14 、 X0.15 、 S/S

*1. AH20MC-5A 机种输入端为差动线路; AH15PM 机种的 X0.8 · X0.9 输入端为差动线路 ,
AH05PM/10PM-5A 机种输入端为集电极线路; AH15PM-5A 机种之 X0.10~X0.15 为集电极线
路。

*2. 只有 AH05PM/10PM-5A 机种需接 S/S ; AH15PM 机种的 X0.10~X0.15 需接 S/S 。

*3. U/D : 上数/下数 · P/D : 脉冲/方向 · A/B : A 相/B 相

- C200 由 SM200/SM201 选择计数模式 · 输入信号由 X0.8 · X0.9 控制 ; 清除信号由 SM203 启动 · 由 X0.0 控制。
 - C204 由 SM204/SM205 选择计数模式 · 输入信号由 X0.10 · X0.11 控制 ; 清除信号由 SM207 设定 · 由 X0.1 控制。
 - C208 由 SM208/SM209 选择计数模式 · 输入信号由 X0.12 · X0.13 控制 ; 清除信号由 SM211 设定 · 由 X0.2 控制。
 - C212 由 SM212/SM213 选择计数模式 · 输入信号由 X0.14 · X0.15 控制 ; 清除信号由 SM215 设定 · 由 X0.3 控制。
 - C216 由 SM216/SM217 选择计数模式 · 输入信号由 X0.12 · X0.13 控制 ; 清除信号由 SM219 设定 · 由 X0.2 控制。
 - C220 由 SM221/SM220 选择计数模式 · 输入信号由 X0.14 · X0.15 控制 ; 清除信号由 SM223 设定 · 由 X0.3 控制。
3. 设定值可使用常数 K 或使用数据寄存器 D 。 设定值可以是正负数 · 若使用数据寄存器 D 则一个设定值占用两个连续的数据寄存器 。
 4. 一般用计数器在模块停电的时候 · 计数器当前值即被清除 · 若为停电保持型 (积分型) 计数器 · 则会将停电前的当前值及计数器接点状态储存着 · 复电后会继续累计 。
 5. 计数器当前值由 2,147,483,647 再往上累计时则变为 -2,147,483,648 。 同理 · 计数器当前值由 -2,147,483,648 再往下递减时 · 则变为 2,147,483,647 。

3.9 寄存器[D][V][Z]的编号及功能

一般用	D0~D9999 · 10,000 点。	合计 10,014 点
变址用寄存器 V、Z	V0~V5、Z0~Z7 · 14 点	

寄存器依其性质可区分为下列 3 种：

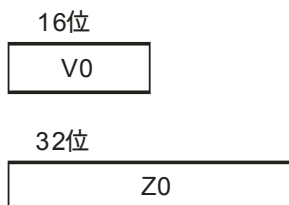
1. 一般用寄存器：当模块由 RUN→STOP 或断电时，寄存器内的数值数据会被清除为 0。如果让 SM33=ON 时，则 AH500 运动模块由 RUN→STOP 时，数据会保持不被清除，但断电时仍会被清除为 0。
2. 停电保持用寄存器：当模块断电时此区域的寄存器数据不会被清除，仍保持其断电前之数值。
清除停电保持用寄存器的内容值，可使用 RST 或 ZRST 指令。
目前 AH500 运动模块尚无停电保持用寄存器
3. 变址用寄存器[V]、[Z]：变址寄存器 V 为 16 位寄存器，而 Z 为 32 位寄存器，V0~V5、Z0~Z7 共计 14 点。

3

3.9.1 数据寄存器[D]

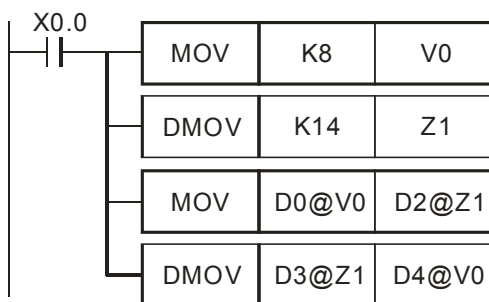
用于储存数值数据，其数据长度为 16 位 (-32,768~+32,767)，最高位为正负号，可储存 -32,768~+32,767 之数值数据，亦可将两个 16 位寄存器合并成一个 32 位寄存器 (D+1、D 编号小的为下 16 位) 使用，而其最高位为正负号，可储存 -2,147,483,648~+2,147,483,647 之数值数据。

3.9.2 变址用寄存器[V][Z]



V 与一般的数据寄存器一样的都是 16 位的数据寄存器，它可以自由的被写入及读出，若当一般寄存器用，仅能使用在 16 位的指令里。

Z 为 32 位的数据寄存器，若当一般寄存器用，仅能使用在 32 位的指令里。



当 X0.0=ON 时，V0=8、Z1=14。

D8 寄存器内容被搬到 D16；D17 寄存器内容被搬到 D12

如果使用变址寄存器 V、Z 来修饰操作数时，16 位指令及 32 位指令，皆可混用。(如左范例)

变址寄存器与一般的操作数相同可用来做为搬移或比较，可用于字符装置 (KnM、KnS、T、C、D、SR) 及位装置 (X、Y、M、S、SM)。

V0~V5、Z0~Z7 共计 14 点

※常数与部分指令并不支持变址用法，使用变址寄存器 V、Z 来修饰操作数请参考第 5.4 章。

3.10 特殊寄存器[SR]的编号及功能

用于储存数值数据，其数据长度为 16 位 (-32,768~+32,767)，最高位为正负号，可储存 -32,768~+32,767 之数值数据，亦可将两个 16 位寄存器合并成一个 32 位寄存器 (D+1、D 编号小的为下 16 位) 使用，而其最高位为正负号，可储存 -2,147,483,648~+2,147,483,647 之数值数据。

特殊用	SR0~SR16383，16,384 点。	合计
变址用寄存器 V、Z	V0~V5、Z0~Z7，14 点。	16,400 点

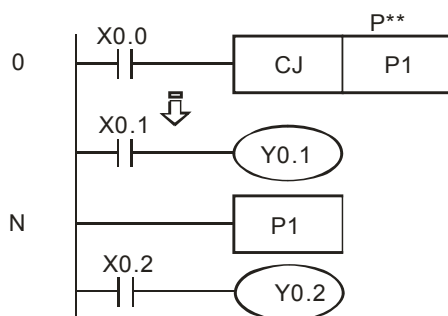
特殊用寄存器：每个特殊用途寄存器均有其特殊定义与用途，主要做为存放系统状态、错误信息、监视状态之用，未定义的特殊用寄存器请勿使用。请参考第 3.12 节特殊继电器及特殊寄存器和第 3.13 节特殊辅助继电器及特殊寄存器群组功能说明。

3.11 指针[P]的编号及功能

指针装置 P	P0~P255，256 点。
--------	----------------

指针 P：搭配应用指令 API 00 CJ、API 256 CJN、API 257 JMP 使用，为 CJ、CJN、JMP 的位置指针。详细说明请参考第 5 章节中的 CJ、CJN、JMP 指令使用说明。

CJ 条件跳跃：



当 X0.0=ON 时程序自动从地址 0 跳跃至 N (即指定之标签 P1) 继续执行，中间地址跳过不执行。
当 X0.0=OFF 时程序如同一般程序由地址 0 继续往下执行，此时 CJ 指令不被执行。

3.12 特殊继电器[SM]及特殊寄存器[SR]

3.12.1 特殊继电器[SM]

特殊辅助继电器 (SM) 及特殊数据寄存器 (SR)，它的种类及功能如下所示。下列各表中，在编号右上角有“*”标记可参考下一节的功能说明，像其中属性栏中标示为“R”者，表示仅可作读取的动作，若标示为“R/W”，表示可作读写的动作。另若标示为“-”，表示无变化。标示为“#”，则表示系统会依照 AH500 运动模块状态作设定，使用者可读取该设定值对照手册之说明，进一步了解系统信息。

SM 编号	功能说明	适用机种 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 值
		05 M	15 M	10 M	20 M					
000	运转监视常开接点 (a 接点)。RUN 中常时 ON，a 接点。RUN 的状态下，此接点 ON	○	○	○	○	OFF	ON	OFF	R	OFF
001	运转监视常闭接点 (b 接点)。RUN 中常时 OFF，b 接点。RUN 的状态下，此接点 OFF	○	○	○	○	ON	OFF	ON	R	ON
002	启动正向 (RUN 的瞬间 'ON') 脉冲。初期脉冲，a 接点。RUN 的瞬间，产生正向的脉冲，其宽度=扫描周期	○	○	○	○	OFF	ON	OFF	R	OFF
003	启动负向 (RUN 的瞬间 'OFF') 脉冲。初期脉冲，a 接点。RUN 的瞬间，产生负向的 PULSE，PULSE 的宽度=扫描周期	○	○	○	○	ON	OFF	ON	R	ON
008	扫描逾时定时器 (WDT) ON	○	○	○	○	OFF	OFF	-	R	OFF
009	LV 信号曾经发生过纪录	○	○	○	○	OFF	-	-	R	OFF
011	10ms 时钟脉冲，5ms ON/5ms OFF	○	○	○	○	OFF	-	-	R	OFF
012	100ms 时钟脉冲，50ms ON/50ms OFF	○	○	○	○	OFF	-	-	R	OFF
013	1s 时钟脉冲，0.5s ON/0.5s OFF	○	○	○	○	OFF	-	-	R	OFF
014	1min 时钟脉冲，30s ON/30s OFF	○	○	○	○	OFF	-	-	R	OFF
020	零点旗标 (SFRD、SFWR 指令使用)	○	○	○	○	OFF	-	-	R	OFF
022	进位旗标 (SFWR、RCR、RCL 指令使用)	○	○	○	○	OFF	-	-	R	OFF
034	Y 输出全部禁止	○	○	○	○	OFF	-	-	R/W	OFF
039	固定扫描时间	○	○	○	○	OFF	-	-	R/W	OFF
072	AH500 运动模块 RUN 指令执行 (通讯)	○	○	○	○	OFF	ON	OFF	R/W	OFF

SM 编号	功能说明	适用机种 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 值
		05 M	15 M	10 M	20 M					
073	语法检查标志	○	○	○	○	OFF	ON	OFF	R/W	OFF
087	LV 信号动作标志	○	○	○	○	OFF	-	-	R/W	OFF
161	8 位处理模式 (ON 时 8 位模式/OFF 时 16 位模式)	○	○	○	○	OFF	-	-	R/W	OFF
200	C200 高速计数模式设定	○	○	○	○	OFF	-	-	R/W	OFF
201						OFF	-	-	R/W	OFF
203	C200 高速计数重置设定	○	○	○	○	OFF	-	-	R/W	OFF
204	C204 高速计数模式设定	X	○	○	○	OFF	-	-	R/W	OFF
205						OFF	-	-	R/W	OFF
207	C204 高速计数重置设定	X	○	○	○	OFF	-	-	R/W	OFF
208	C208 高速计数模式设定	X	○	○	○	OFF	-	-	R/W	OFF
209						OFF	-	-	R/W	OFF
211	C208 高速计数重置设定	X	○	○	○	OFF	-	-	R/W	OFF
212	C212 高速计数模式设定	X	○	○	○	OFF	-	-	R/W	OFF
213						OFF	-	-	R/W	OFF
215	C212 高速计数重置设定	X	○	○	○	OFF	-	-	R/W	OFF
216	C216 高速计数模式设定	X	○	○	○	OFF	-	-	R/W	OFF
217						OFF	-	-	R/W	OFF
219	C216 高速计数重置设定	X	○	○	○	OFF	-	-	R/W	OFF
220	C220 高速计数模式设定	X	○	○	○	OFF	-	-	R/W	OFF
221						OFF	-	-	R/W	OFF
223	C220 高速计数重置设定	X	○	○	○	OFF	-	-	R/W	OFF
240	C240 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R/W	OFF
241	C241 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R/W	OFF
242	C242 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R/W	OFF
243	C243 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R/W	OFF
244	C244 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R/W	OFF
245	C245 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R/W	OFF
246	C246 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
247	C247 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
248	C248 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
249	C249 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
250	C250 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
251	C251 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
252	C252 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF

3

SM 编号	功能说明	适用机种*1				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 值
		05 M	15 M	10 M	20 M					
253	C253 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
254	C254 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
255	C255 计数模式设定 (ON 时为下数)	○	○	○	○	OFF	-	-	R	OFF
303	XCH 指令高低位交换标志	○	○	○	○	OFF	-	-	R/W	OFF
304	X 输入点可强制 ON/OFF	○	○	○	○	OFF	-	-	R/W	OFF
920	O100 弧度/角度使用标志	○	○	○	○	OFF	-	-	R/W	OFF
952	O100 Ready 标志	○	○	○	○	OFF	-	-	R	OFF
953	O100 错误标志	○	○	○	○	OFF	-	-	R	OFF
968	O100 零标志 (O100 Zero flag)	○	○	○	○	OFF	-	-	R	OFF
969	O100 借位标志 (O100 Borrow flag)	○	○	○	○	OFF	-	-	R	OFF
970	O100 进位标志 (O100 Carry flag)	○	○	○	○	OFF	-	-	R	OFF
971	O100 浮点数运算错误标志	○	○	○	○	OFF	-	-	R	OFF

*1 : 05M=AH05PM-5A · 15M=AH15PM-5A · 10M=AH10PM-5A · 20M=AH20MC-5A

运动程序 OX 状态标志

SM 编号	功能说明	适用机种*1				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 值
		05 M	15 M	10 M	20 M					
1016	OX 弧度/角度使用标志	○	○	○	○	OFF	-	-	R/W	OFF
1049	OX 错误旗标与清除	○	○	○	○	OFF	-	-	R/W	OFF
1050	OX M 码 ON 标志 (OX 启动时自动清除)	○	○	○	○	OFF	-	OFF	R	OFF
1051	OX M00 ON 标志 (OX 启动时自动清除)	○	○	○	○	OFF	-	-	R	OFF
1052	OX M02 ON 标志 (OX 启动时自动清除)	○	○	○	○	OFF	ON	-	R	OFF
1064	OX 零点标志 (OX Zero flag)	○	○	○	○	OFF	-	-	R	OFF
1065	OX 借位标志 (OX Borrow flag)	○	○	○	○	OFF	-	-	R	OFF
1066	OX 进位标志 (OX Carry flag)	○	○	○	○	OFF	-	-	R	OFF
1067	OX 浮点数运算错误标志	○	○	○	○	OFF	-	-	R	OFF

*1 : 05M=AH05PM-5A · 15M=AH15PM-5A · 10M=AH10PM-5A · 20M=AH20MC-5A

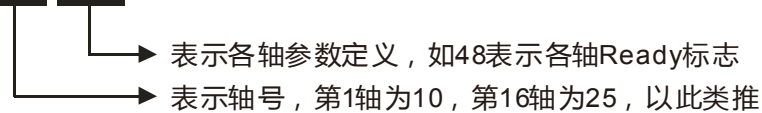
运动轴的特殊继电器 (SM)

运动轴的特殊继电器 (SM) 使用特殊继电器 (SM1000) 到特殊继电器 (SM2599) · 并且每一轴将使用 100 个特殊继电器 (SM) · 命名规则如下 :

- 每一轴均含有相同数量和相同定义的参数

特殊继电器 (SM) 以 SM1000 为轴参数开始 · 并且 100 组参数为轴参数的间隔定义如下

SM mn xx



以下为例 :SM1048,表示第 1 轴的 Ready 标志 ·而 SM1148 表示第 2 轴的 Ready 标志 ·SM1548 表示第 6 轴的 Ready 标志 ·

- 以下为轴参数的定义与范围 ·

轴号	1	2	3	4	5	6
SM 范围	1000~1099 (mn=10)	1100~1199 (mn=11)	1200~1299 (mn=12)	1300~1399 (mn=13)	1400~1499 (mn=14)	1500~1599 (mn=15)
轴号	7	8	9	10	11	12
SM 范围	1600~1699 (mn=16)	1700~1799 (mn=17)	1800~1899 (mn=18)	1900~1999 (mn=19)	2000~2099 (mn=20)	2100~2199 (mn=21)
轴号	13	14	15	16	SM2600 以后目前未使用	
SM 范围	2200~2299 (mn=22)	2300~2399 (mn=23)	2400~2499 (mn=24)	2500~2599 (mn=25)		

SM 编号	功能说明	适用機種 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂值
		05 M	15 M	10 M	20 M					
mn04	电子凸轮周期运动停止	○	○	○	○	OFF	-	-	R/W	OFF
mn08	电子凸轮主轴偏移设定启动	○	○	○	○	OFF	-	-	R/W	OFF
mn17	指定轴停止指定角度标志	○	○	○	○	OFF	-	-	R/W	OFF
mn48	指定轴 Ready Flag	○	○	○	○	ON	ON	ON	R	ON
mn49	运动错误标志与清除 (启动时自动清除)	○	○	○	○	OFF	-	-	R/W	OFF
mn69	电子凸轮每周期开始标志	○	○	○	○	OFF	-	-	R/W	OFF

*1 : 05M=AH05PM-5A · 15M=AH15PM-5A · 10M=AH10PM-5A · 20M=AH20MC-5A

3.12.2 特殊寄存器[SR]

3

SR 编号	功能说明	适用机种 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	停 电 保 持	出 厂 值
		05 M	15 M	10 M	20 M						
000	程序扫描逾时定时器 (WDT) (单位: ms)	○	○	○	○	200	-	-	R/W	否	200
002	程序容量	○	○	○	○	65535	-	-	R	否	65535
003	程序内存内容总和	○	○	○	○	-	-	-	R	否	0
004	程序内存内容及 CAM 表总和	○	○	○	○	-	-	-	R	否	0
005	AH500 运动模块机种系统程序版本 (以出厂时的版本显示)	○	○	○	○	#	-	-	R	否	#
008	WDT 定时器 ON 之 STEP 地址	○	○	○	○	0	-	-	R	否	0
009	纪录 LV 信号曾经发生过次数	○	○	○	○	0	-	-	R	否	0
010	当前扫描时间 (单位: 1ms)	○	○	○	○	0	-	-	R	否	0
011	最小扫描时间 (单位: 1ms)	○	○	○	○	0	-	-	R	否	0
012	最大扫描时间 (单位: 1ms)	○	○	○	○	0	-	-	R	否	0
020	X 输入滤波设定 (单位: ms)	○	○	○	○	10	-	-	R/W	否	10
039	固定扫描时间 (单位: ms)	○	○	○	○	0	-	-	R/W	否	0
121	AH500 运动模块通讯地址	○	○	○	○	-	-	-	R/W	是	1
200	SD 卡备份 M 辅助继电器开始地址	X	○	○	○	0	-	-	R/W	否	0
201	SD 卡备份 M 辅助继电器结束地址	X	○	○	○	0	-	-	R/W	否	0
202	SD 卡备份 T 定时器开始地址	X	○	○	○	0	-	-	R/W	否	0
203	SD 卡备份 T 定时器结束地址	X	○	○	○	0	-	-	R/W	否	0
204	SD 卡备份 C16 位计数器开始地址	X	○	○	○	0	-	-	R/W	否	0
205	SD 卡备份 C16 位计数器结束地址	X	○	○	○	0	-	-	R/W	否	0
206	SD 卡备份 C32 位计数器开始地址	X	○	○	○	0	-	-	R/W	否	0
207	SD 卡备份 C32 位计数器结束地址	X	○	○	○	0	-	-	R/W	否	0
208	SD 卡备份 S 步进点开始地址	X	○	○	○	0	-	-	R/W	否	0
209	SD 卡备份 S 步进点结束地址	X	○	○	○	0	-	-	R/W	否	0
210	SD 卡备份 D 寄存器开始地址	X	○	○	○	0	-	-	R/W	否	0
211	SD 卡备份 D 寄存器结束地址	X	○	○	○	0	-	-	R/W	否	0
212	SD 卡备份 W 寄存器开始地址	X	○	○	○	0	-	-	R/W	否	0
213	SD 卡备份 W 寄存器结束地址	X	○	○	○	0	-	-	R/W	否	0

SR 编号	功能说明	适用机种 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	停 电 保 持	出 厂 值
		05 M	15 M	10 M	20 M						
214	SD 卡存取控制寄存器	X	○	○	○	0	-	-	R/W	否	0
400	中断致能	○	○	○	○	0	-	-	R/W	否	0
401	时间中断周期 (单位 : ms)	○	○	○	○	0	-	-	R/W	否	0
700	OX 程序执行编号	○	○	○	○	0	-	-	R	否	0
702	OX 程序执行 STEP 号码	○	○	○	○	0	-	-	R	否	0
703*	OX 执行 M 码	○	○	○	○	0	-	-	R	否	0
704	OX 等待时间设定值	○	○	○	○	0	-	-	R	否	0
705	OX 等待时间目前值	○	○	○	○	0	-	-	R	否	0
706	OX RPT 指令设定值	○	○	○	○	0	-	-	R	否	0
707	OX RPT 指令目前值	○	○	○	○	0	-	-	R	否	0
708 709	X/Y 轴补偿值	○	○	○	○	0	-	-	R/W	否	0
710 711	中心点补偿值	○	○	○	○	0	-	-	R/W	否	0
712 713	半径补偿值	○	○	○	○	0	-	-	R/W	否	0
796 797	连续插补减速速度	○	○	○	○	0	-	-	R/W	否	0
798	G 码移动速度百分比设定	○	○	○	○	0	-	-	R	否	0
799	输入端子极性	○	○	○	○	0	-	-	R/W	否	0
800	输入端子状态	○	○	○	○	0	-	-	R	否	0
802	O100 错误编号	○	○	○	○	0	-	-	R/W	否	0
803	O100 错误 STEP 位置	○	○	○	○	0	0	-	R/W	否	0
804	输入端子极性 2	○	○	○	○	0	-	-	R/W	否	0
805	输入端子状态 2	○	○	○	○	0	-	-	R	否	0
806	输入端子滤波系数设定	○	○	○	○	0	-	-	R/W	否	0
808	Ethernet IP	X	○	○	○	-	-	-	R/W	否	100
809	Ethernet IP	X	○	○	○	-	-	-	R/W	否	49320

*1 : 05M=AH05PM-5A · 15M=AH15PM-5A · 10M=AH10PM-5A · 20M=AH20MC-5A

运动程序 OX 相关

3

SR 编号	功能说明	适用机种*1				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂值
		05 M	15 M	10 M	20 M					
1041	OX 错误编号	○	○	○	○	0	-	-	R	0
1049	OX 0~99 停止模式 (K1 : 下次启动走完未完行程 · K2 : 下次启动执行下一指令 · 其它 : 重新开始)	○	○	○	○	OFF	-	-	R/W	0
1052	设定 OX 程序号码	○	○	○	○	OFF	-	OFF	R/W	0
1053	OX 错误 STEP 位置	○	○	○	○	OFF	-	-	R	0

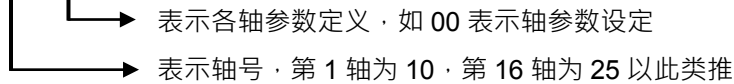
*1 : 05M=AH05PM-5A · 15M=AH15PM-5A · 10M=AH10PM-5A · 20M=AH20MC-5A

运动轴的特殊寄存器 (SR)

运动轴的特殊寄存器 (SR) 使用特殊寄存器 (SR1000) 到特殊寄存器 (SR2599) · 并且每一轴将使用 100 个特殊寄存器 (SR) · 命名规则如下 :

- 每一轴均含有相同数量和相同定义参数
- 特殊寄存器 (SR) 以 SR1000 为轴参数开始 · 并且 100 组参数为轴参数的间隔 · 并且 100 组参数为轴参数的间隔定义如下 · 而 SR1500 表示第 6 轴的不同定义参数。

SR mn xx



- 以下为例 : SR1000 · 表示第 1 轴的参数设定 · 而 SR1100 表示第 2 轴的参数设定 · SR1500 表示第 6 轴的参数设定。
- 以下为轴参数的定义与范围。

轴号	1	2	3	4	5	6
SR 范围	1000~1099 (mn=10)	1100~1199 (mn=11)	1200~1299 (mn=12)	1300~1399 (mn=13)	1400~1499 (mn=14)	1500~1599 (mn=15)
轴号	7	8	9	10	11	12
SR 范围	1600~1699 (mn=16)	1700~1799 (mn=17)	1800~1899 (mn=18)	1900~1999 (mn=19)	2000~2099 (mn=20)	2100~2199 (mn=21)
轴号	13	14	15	16	SR2600 以后目前未使用	
SR 范围	2200~2299 (mn=22)	2300~2399 (mn=23)	2400~2499 (mn=24)	2500~2599 (mn=25)		

SR 编号	功能说明	适用機種 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	停电 保持	出 厂 值
		05 M	15 M	10 M	20 M						
mn00	指定轴参数设定	○	○	○	○	-	-	-	R/W	否	0
mn01	指定轴背隙补偿	○	○	○	○	-	-	-	R/W	否	0
mn02	指定轴电机一圈所需脉冲数 A (Low word)	○	○	○	○	-	-	-	R/W	否	2000
mn03	指定轴电机一圈所需脉冲数 A (High word)										
mn04	指定轴电机一圈之移动距离 B (Low word)	○	○	○	○	-	-	-	R/W	否	1000
mn05	指定轴电机一圈之移动距离 B (High word)										
mn06	指定轴最高速度 V_{MAX} (Low word)	○	○	○	○	-	-	-	R/W	否	500K
mn07	指定轴最高速度 V_{MAX} (High word)										
mn08	指定轴启动速度 V_{BIAS} (Low word)	○	○	○	○	-	-	-	R/W	否	0
mn09	指定轴启动速度 V_{BIAS} (High word)										
mn10	指定轴寸动 JOG 速度 V_{JOG} (Low word)	○	○	○	○	-	-	-	R/W	否	5000
mn11	指定轴寸动 JOG 速度 V_{JOG} (High word)										
mn12	指定轴原点回归速度 V_{RT} (Low word)	○	○	○	○	-	-	-	R/W	否	50K
mn13	指定轴原点回归速度 V_{RT} (High word)										
mn14	指定轴原点回归减速速度 V_{CR} (Low word)	○	○	○	○	-	-	-	R/W	否	1000
mn15	指定轴原点回归减速速度 V_{CR} (High word)										
mn16	指定轴零点信号数 N	○	○	○	○	-	-	-	R/W	否	0
mn17	指定轴补充距离 P	○	○	○	○	-	-	-	R/W	否	0
mn18	指定轴原点位置定义 HP (Low word)	○	○	○	○	-	-	-	R/W	否	0
mn19	指定轴原点位置定义 HP (High word)	○	○	○	○	-	-	-	R/W	否	0
mn20	指定轴加速时间 T_{ACC}	○	○	○	○	-	-	-	R/W	否	500
mn21	指定轴减速时间 T_{DEC}	○	○	○	○	-	-	-	R/W	否	500
mn22	指定轴目标位置 (I) P (I) (Low word)	○	○	○	○	○	-	-	R/W	否	0
mn23	指定轴目标位置 (I) P (I) (High word)	○	○	○	○	○	-	-	R/W	否	0

3

SR 编号	功能说明	适用机种*1				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	停电 保持	出 厂 值
		05 M	15 M	10 M	20 M						
mn24	指定轴运转速度(I)V(I)(Low word)	○	○	○	○	1000	-	-	R/W	否	1000
mn25	指定轴运转速度(I)V(I)(High word)	○	○	○	○	1000	-	-	R/W	否	1000
mn26	指定轴目标位置(II)P(II)(Low word)	○	○	○	○	0	-	-	R/W	否	0
mn27	指定轴目标位置(II)P(II)(High word)	○	○	○	○	0	-	-	R/W	否	0
mn28	指定轴运转速度(II)V(II)(Low word)	○	○	○	○	2000	-	-	R/W	否	2000
mn29	指定轴运转速度(II)V(II)(High word)										
mn30	指定轴运转命令	○	○	○	○	0	-	0	R/W	否	0
mn31	指定轴工作模式	○	○	○	○	0	-	-	R/W	否	0
mn32	指定轴当前位置 CP (PLS) (Low word)	○		○	○	0	-	-	R/W	否	0
mn33	指定轴当前位置 CP (PLS) (High word)	○	○	○	○	0	-	-	R/W	否	0
mn34	指定轴当前速度(PPS)(Low word)	○	○	○	○	0	0	0	R/W	否	0
mn35	指定轴当前速度(PPS)(High word)										
mn36	指定轴当前位置 CP (unit) (Low word)	○	○	○	○	0	-	-	R/W	否	0
mn37	指定轴当前位置 CP (unit) (High word)										
mn38	指定轴当前速度 CS (unit) (Low word)	○	○	○	○	0	0	0	R/W	否	0
mn39	指定轴当前速度 CS (unit) (High word)										
mn40	指定轴执行状态	○	○	○	○	0	-	-	R	否	0
mn41	指定轴错误编号	○	○	○	○	0	-	-	R	否	0
mn42	指定轴电子齿轮(分子)	○	○	○	○	-	-	-	R/W	否	1
mn43	指定轴电子齿轮(分母)	○	○	○	○	-	-	-	R/W	否	1
mn44	指定轴手摇轮输入频率(Low word)	○	○	○	○	0	0	-	R/W	否	0
mn45	指定轴手摇轮输入频率(High word)	○	○	○	○	0	0	-	R/W	否	0
mn46	指定轴累计手摇轮输入脉冲个数(Low word)	○	○	○	○	0	-	-	R/W	否	0

SR 编号	功能说明	适用機種 ^{*1}				OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	停电 保持	出 厂 值
		05 M	15 M	10 M	20 M						
mn47	指定轴累计手摇轮输入脉冲个数 (High word)	○	○	○	○	0	-	-	R/W	否	0
mn48	指定轴手摇轮输入响应速度	○	○	○	○	-	-	-	R/W	否	5
mn50	指定轴电气零点 (Low word)	○	○	○	○	-	-	-	R/W	否	0
mn51	指定轴电气零点 (High word)	○	○	○	○	-	-	-	R/W	否	0
mn68	指定轴 DMCNET 伺服编码器当前位置 (Low word)	X	X	X	○	-	-	-	R	否	0
mn69	指定轴 DMCNET 伺服编码器当前位置 (High word)	X	X	X	○	-	-	-	R	否	0
mn72	指定轴 DMCNET 伺服控制命令	X	X	X	○	-	-	-	R/W	否	0
mn73	指定轴 DMCNET 伺服状态	X	X	X	○	-	-	-	R	否	0
mn74	指定轴 DMCNET 伺服错误码 (Low word)	X	X	X	○	-	-	-	R/W	否	0
mn75	指定轴 DMCNET 伺服错误码 (High word)	X	X	X	○	-	-	-	R/W	否	0
mn76	指定轴 DMCNET 伺服参数读写 Data (Low word)	X	X	X	○	-	-	-	R/W	否	0
mn77	指定轴 DMCNET 伺服参数读写 Data (High word)	X	X	X	○	-	-	-	R/W	否	0
mn78	指定轴 DMCNET 伺服参数位置	X	X	X	○	-	-	-	R/W	否	0
	指定轴 DMCNET 回原点模式方式	X	X	X	○	-	-	-	R/W	否	0

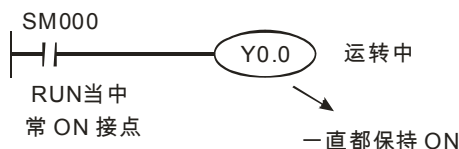
*1 : 05M=AH05PM-5A · 10M=AH10PM-5A · 15M=AH15PM-5A · 20M=AH20MC-5A

3.13 特殊继电器[SM]及特殊寄存器[SR]功能说明

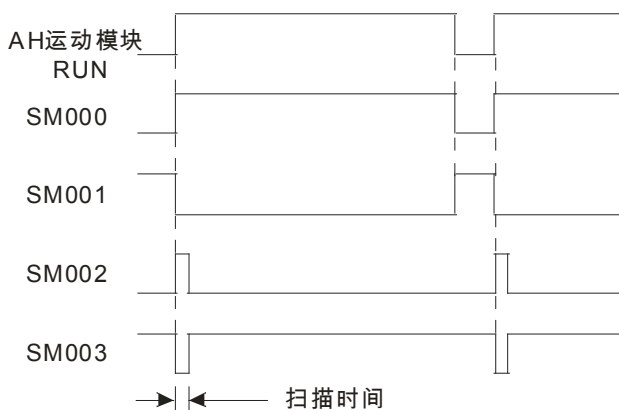
运转标志

SM000~SM003

1. SM000 : SM000 为 RUN 中时，常 ON 接点，即运转监视常开接点 (A 接点)，AH500 运动模块于 RUN 的状态下，SM000 保持为 ON。



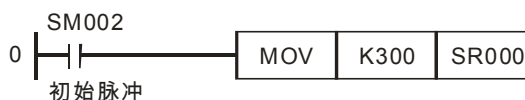
2. SM001 : SM001 为 RUN 中时，常 OFF 接点，即运转监视常闭接点 (B 接点)，AH500 运动模块于 RUN 的状态下 SM001 保持为 OFF。
3. SM002 : 开始 RUN 的第一次扫描 ON，之后保持为 OFF。该脉冲的宽度为一次扫描时间，当要作各种初始设定工作时使用本接点。
4. SM003 : 开始 RUN 的第一次扫描 OFF，之后一直 ON。即启动负向 (RUN 的瞬间 'OFF') 脉冲。



监控定时器

SR000

1. 监控定时器专门用来监视扫描时间，当扫描时间超过监控定时器的设定时间时，ERROR 红色指示灯长亮，输出全部变成 OFF。
2. 监控定时器时间的初始值为 200ms，当程序长或是运算过于复杂时，可于程序中使用 MOV 指令来变更监控定时器的设定值，如下所示，将监控定时器的设定值变更为 300ms。

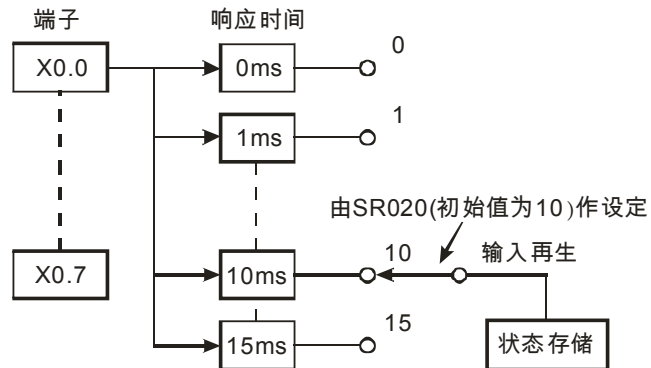


3. 监控定时器最大可设定至 32,767ms，但必须注意，监控定时器设定过大时，运算异常发生的检出时机将会跟着被拖慢。因此，若非复杂的运算使得扫描时间超过 200ms，一般的情况下请维持在 200ms 以下较佳。
4. 指令运算过于复杂会造成扫描时间过长，扫描时间是否超过 SR000 的设定值，请监视 SR010~SR012。此种情况下，可变更 SR000 的设定值。

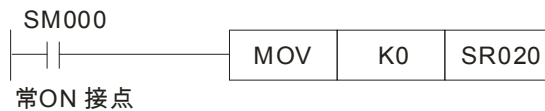
输入端
响应时间
的调整

SR020

1. X 输入端，可由 SR020 的内容来设定输入端接收脉冲的反应时间，设定范围 0~20，单位 ms。
电源 OFF→ON 变化时，SR020、SR021 的内容自动变成 10。



2. 如果程序中执行下面的程序时，X 输入端的反应时间被设定为 0ms，由于，输入端均串接 RC 滤波回路的关系，输入端最快的反应时间为 50μs。

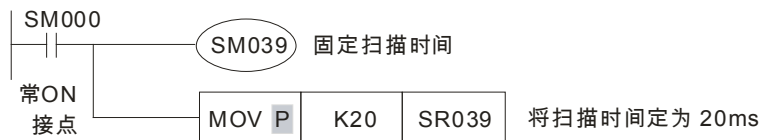


3. 当程序中使用高速计数器、中断插入等功能时，则此设定值无效。

固定
扫描时间

SM039、SR039

1. 将 SM039 为 ON，则程序的扫描时间固定以 SR039 的内容来决定，当程序执行完毕，必须等待固定的扫描时间到达时才执行下一次的扫描，如果 SR039 的内容小于实际上程序的扫描时间时，则以实际上程序的扫描时间为主。



2. SR010~SR012 所显示的扫描时间亦包括固定的扫描时间。

3

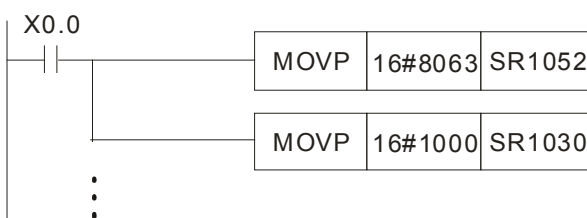
设定 OX
程序编号

SR1052

由 SR1052 可指定欲执行的 OX 程序编号，其设定步骤如下：

1. 首先设定 SR1052 的 b[14]=1 或 b[15]=1 或 b[14]=b[15]=1。(前述三种条件只要成立其中一项即可)，同时将 SR1052 的 b[0~13]写入 K99 (=16#63)，即为设定 OX 编号数值为 OX99。结合前述两条件，因此将 16#8063 写至 SR1052 寄存器之中。
2. 上述步骤 1 设定完成后，设定 SR1030 为 16#1000，即可启动 SR1052 所指定的 OX 编号程序。

上述步骤 1~2 之范例程序如下：



在 O100 主程序中，设计由 X0.0 启动子程序 OX99，并且执行 OX99 内之程序。六组计数功能，其设定数据如下表所示：

高速计数
功能

SM200、C200
SM204、C204
SM208、C208
SM212、C212
SM216、C216
SM220、C220

组别	计数器 编号	计数模式		清除计数 外部接脚	外部输入接脚 *2
		装置	设定值*1		
第一组	C200	K1SM200	0 : U/D 1 : P/D 2 : A/B (1 倍频) 3 : 4A/B (4 倍频)	X0.0+、X0.0- SM203	X0.8、X0.9、 S/S
第二组	C204	K1SM204		X0.1+、X0.1- SM207	X0.10、X0.11、 S/S
第三组	C208	K1SM208		X0.2+、X0.2- SM211	X0.12、X0.13、 S/S
第四组	C212	K1SM212		X0.3+、X0.3- SM215	X0.14、X0.15、 S/S
第五组	C216	K1SM216		X0.2+、X0.2- SM219	X0.12、 X0.13、S/S
第六组	C220	K1SM220		X0.3+、X0.3- SM223	X0.14、 X0.15、S/S

*1. U/D：上数/下数，P/D：脉冲/方向，A/B：A相/B相

*2. AH05PM/10PM-5A 输入为集电极线路，AH20MC-5A 输入为差动线路，AH15PM-5A 的 X0.8、X0.9 输入为差动线路，AH15PM-5A 的 X0.10~X0.15 输入为集电极线路。

3

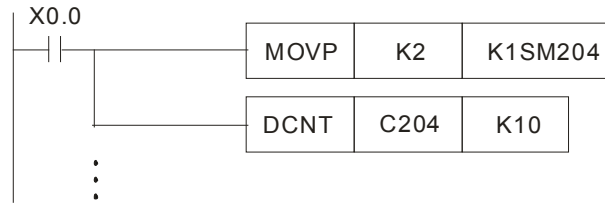
**高速计数
功能**

SM200、C200
SM204、C204
SM208、C208
SM212、C212
SM216、C216
SM220、C220

启动第二组高速计数器，其设定步骤如下：

1. 首先设定第二组计数器模式为 A/B 相(一倍频)计数模式(K1SM204=K2)。
2. 启动 C204 开始计数。

上述步骤 1~2 之范例程序如下：

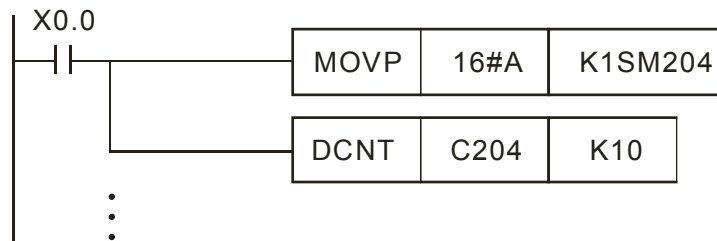


3. 若设定由外部信号清除目前计数值，则模式设定为 (K1SM204=16#A)。

SM207	SM206	SM205	SM204
1	0	1	0

4. 启动 C204 开始计数，当接点 X0.1 为 ON，C204 立即被清除为零。

上述步骤 3~4 之范例程序如下：



**高速计时
功能**

SM200、C201
SM204、C205
SM208、C209
SM212、C213

高速计时功能，其设定数据如下表所示：

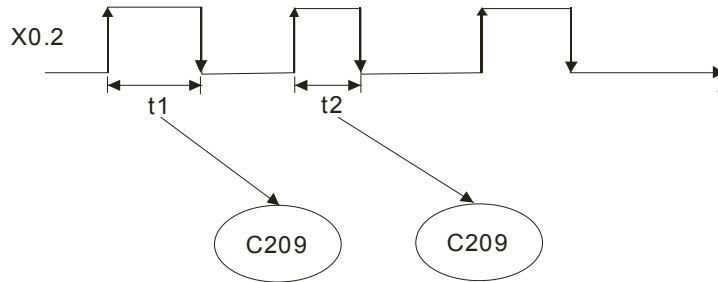
组别	计数装置	计时模式				启动计时外部信号	计时储存装置
		装置	设定值				
			Bit3	Bit2	Bit1	Bit0	
第一组	C200	K1SM200	x	启动计时	x	模式设定	X0.0+、X0.0- C201
第二组	C204	K1SM204	Bit2：启动计时功能				X0.1+、X0.1- C205
第三组	C208	K1SM208	Bit0：等于 0 为一般模式，计时区间为上升沿至下降沿				X0.2+、X0.2- C209
第四组	C212	K1SM212	等于 1 为周期模式，计时区间为上升沿至下一个上升沿				X0.3+、X0.3- C213

高速计时
功能

SM200、C201
SM204、C205
SM208、C209
SM212、C213

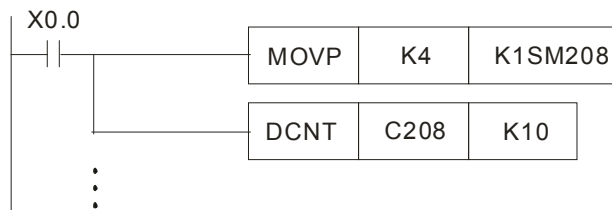
范例一：指定使用第三组定时器为一般模式

1. 首先设定第三组定时器计时模式为一般模式 (Bit0=0)、启动计时功能 (Bit2=1)，则装置 K1SM208=K4。
2. 启动 C208 开始计数，计算 X0.2 外部触发信号上升沿至信号的下降沿之间间隔时间，将此值写入 C209，写入数值单位为 0.01us，如下图所示。



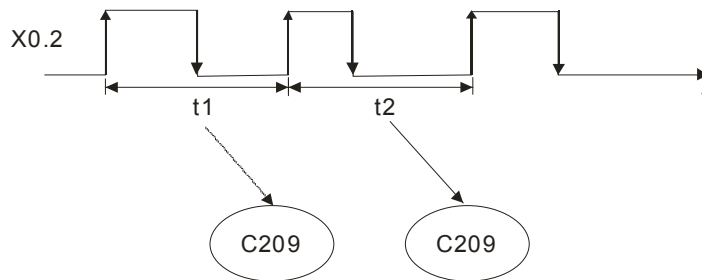
一般模式，单位 0.01μs

上述步骤之范例程序如下：



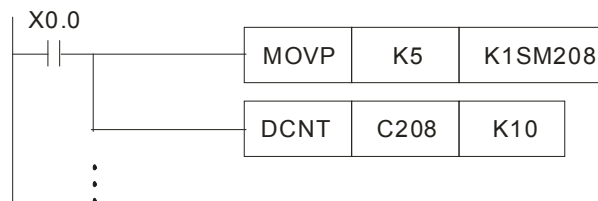
范例二：指定使用第三组定时器为周期模式

1. 若设定计时方式为周期模式 (Bit0=1)，则装置 K1SM208=K5。
2. 启动 C208 开始计数，计算 X0.2 外部触发信号上升沿至下个上升沿之间间隔时间，将此值写入 C209，写入数值单位为 0.01us，如下图所示。



周期模式，单位 0.01μs

上述步骤之范例程序如下：



3. 周期模式用于测频应用。

X输入点可
强制ON/OFF

SM304

中断寄存器

SR400、SR401

当 SM304=ON 时，AH500 运动模块上之 X 输入点可利用周边装置如 PMSoft 做强制 ON-OFF 之动作。

1. SR400 为中断致能寄存器，欲启动以下 9 个中断请将对应之 BIT# ON，说明如下：

bit#	中断	中断编号
0	时间中断	I0
1	外部端子 X0.8	I1
2	外部端子 X0.9	I2
3	外部端子 X0.10	I3
4	外部端子 X0.11	I4
5	外部端子 X0.12	I5
6	外部端子 X0.13	I6
7	外部端子 X0.14	I7
8	外部端子 X0.15	I8

2. 当启动之中断种类为时间中断，则可在 SR401 输入时间中断之周期。
3. 中断的种类分别为时间中断与外部中断，说明如下：
 - 外部中断：当外部端子的输入信号于上升沿或下降沿触发时，不受扫描周期影响，立即中断目前执行中程序而立即跳至指定的中断子程序处执行，当中断子程序执行完毕后即返回中断前执行的程序。
 - 时间中断：PM 每隔时间中断之周期（SR401）便自动中断目前执行之程序而跳至时间中断子程序执行。

清除执行中
M码

SM1050、SR703

- ◆ 欲清除 OX M 码命令时，可执行 SM1050=0，当 SM1050 被执行时，会同时清除 SR703。
- ◆ 上述中的 SM1050 为 OX M 码的被执行标志，而 SR703 为 OX 的 M 码执行寄存器。

Ready 标志

SMmn48
(SM1048、
SM1148、
....SM2548)

1. 每一个运动轴各有 Ready 标志，第 1 轴：SM1048，第 2 轴：SM1148，第 3 轴：SM1248，第 4 轴：SM1348，.....，第 15 轴：SM2448，第 16 轴：SM2548。可用来判断行程是不是还在运转。
2. 以下为第 1 轴 Ready 标志动作说明：第 1 轴在运转前 SM1048 为 ON，开始运转后 SM1048 变为 OFF，运转完后则变为 ON。

3

单轴运动
固定角度停止

SMmn17
(SM1017、
SM1117、
...SM2517)

1. 单轴固定角度停止设定参数表

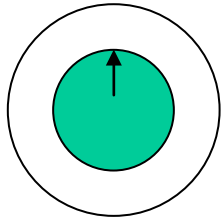
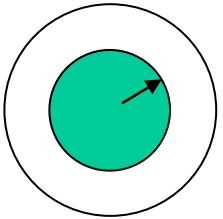
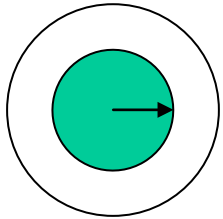
参数功能	1 轴	2 轴	3 轴	4 轴
停止于指定角度	SM1017	SM1117	SM1217	SM1317
停止角度	SR1023 ~ SR1022	SR1123 ~ SR1122	SR1223 ~ SR1222	SR1323 ~ SR1322
一周期脉冲数	SR1027 ~ SR1026	SR1127 ~ SR1126	SR1227 ~ SR1226	SR1327 ~ SR1326
参数功能	5 轴	6 轴	7 轴	8 轴
停止于指定角度	SM1417	SM1517	SM1617	SM1717
停止角度	SR1423 ~ SR1422	SR1523 ~ SR1522	SR1623 ~ SR1622	SR1723 ~ SR1722
一周期脉冲数	SR1427 ~ SR1426	SR1527 ~ SR1526	SR1627 ~ SR1626	SR1727 ~ SR1726
参数功能	9 轴	10 轴	11 轴	12 轴
停止于指定角度	SM1817	SM1917	SM2017	SM2117
停止角度	SR1823 ~ SR1822	SR1923 ~ SR1922	SR2023 ~ SR2022	SR2123 ~ SR2122
一周期脉冲数	SR1827 ~ SR1826	SR1927 ~ SR1926	SR2027 ~ SR2026	SR2127 ~ SR2126
参数功能	13 轴	14 轴	15 轴	16 轴
停止于指定角度	SM2217	SM2317	SM2417	SM2517
停止角度	SR2223 ~ SR2222	SR2323 ~ SR2322	SR2423 ~ SR2422	SR2523 ~ SR2522
一周期脉冲数	SR2227 ~ SR2226	SR2327 ~ SR2326	SR2427 ~ SR2426	SR2527 ~ SR2526

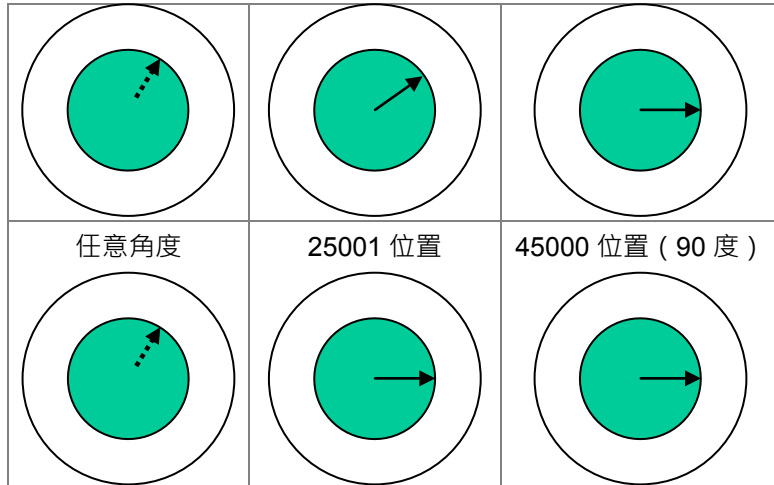
- 在 JOG+ 运动模式中，用来控制第 1 ~ 16 轴固定在某特定角度停止。
- 以第 1 轴为例，设定 SM1017=1，运行一周脉冲数设定于 (SR1027、SR1026)，欲停止的角度设定在 (SR1023、SR1022)，启动 JOG+ 模式运动。

单轴运动
固定角度停止

SMmn17
(SM1017、
SM1117、
...SM2517)

- 举例说明，一周脉冲数为 20000，欲停止角度为 90 度，则各种停止状况如下图所示：

起始位置	JOG 运动停止位置	最后停止位置
0 度位置	4000 位置	5000 位置 (90 度)
		
任意角度	63500 位置	65000 位置 (90 度)



运动错误清除

SMmn49
(SM1049、
SM1149、
....SM2549)

SRmn41
(SR1041、
SR1141、
....SR2541)

当第 1~16 轴发生运动错误时，错误标志分别为各自对应的 SM 装置内，且会将错误信息分别存放至各自对应的 SR 寄存器之中。

欲进行除错时，请清除错误信息寄存器与错误标志。

连续插补

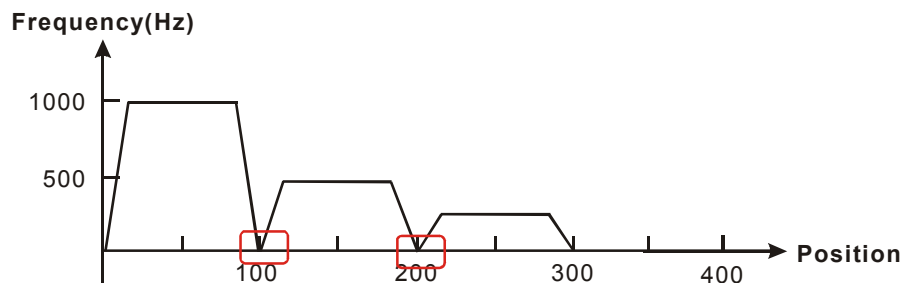
SR797、SR796

连续插补

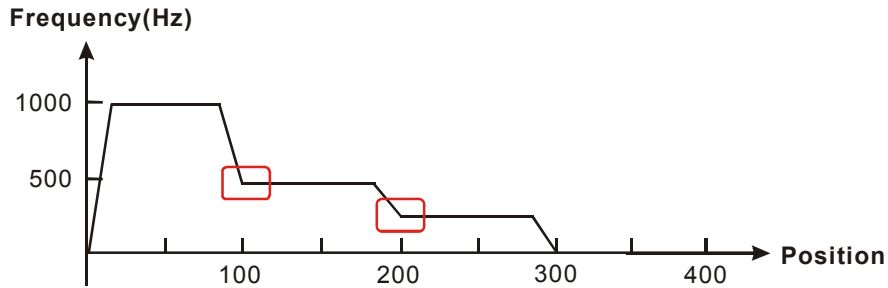
SR797、SR796

当使用 SR797、SR796 设定最大减速速度(kHz)时，路径减速下降时会以 SR797、SR796 设定值与实际减速速度值，二者取小的频率值做为转折点。

设定连续插补 SR797、SR796 为 k0 (关闭连续插补功能)：设定为 k0，则不管减速速度多少，都会降至 0kHz 再加速，如下图所示。



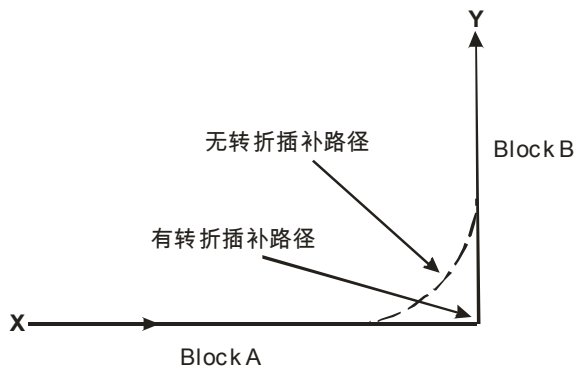
设定连续插补 SR797..796=k500(开启连续插补功能) 减速速度会与 SR797..796 比较，以较小值为低点再加速，如下图所示。



连续路径示意图

在未设定 SR797..796 连续插补时，X→Y 路径为直角转折后向上；在设定连续插补 SR797..796 后，X→Y 路径为较平滑的曲线向上。

```
G01 X100 F1000 ; ( Block A )
Y100 ; ( Block B )
```



Block A：执行 X 轴移动路径；Block B：执行 Y 轴移动路径

设定G码
移动速度
百分比

SR798

1. 当 SR798 设为 100 时，G 码移动速度以原设定速度移动；当 SR798 设为 1000 时，G 码移动速度以原设定速度 10 倍移动；当 SR798 设为 50 时，G 码移动速度以原设定速度一半移动；以此类推。
2. SR798 设定百分比使 G 码移动速度超过 500k (HZ) 时，则会以 500k (HZ) 速度移动。

设定输入
端子极性

SR799

欲设定下表内之输入端子的极性为 a 接点时，请将 bit# OFF。若为 b 接点时，则将 bit# ON 即可。

bit#	输入端子极性	bit#	输入端子极性
0	X0.0	8	X0.8
1	X0.1	9	X0.9
2	X0.2	10	X0.10
3	X0.3	11	X0.11
4	X0.4 ^{*1}	12	X0.12
5	X0.5 ^{*1}	13	X0.13
6	X0.6 ^{*1}	14	X0.14
7	X0.7 ^{*1}	15	X0.15

*1：仅 AH15PM-5A 有此输入点功能

读取输入
端子状态

SR800

当 bit# 为 ON 时，表示有信号输入。反之，则当 bit# 为 OFF 时，表示无信号输入。

bit#	输入端子状态	bit#	输入端子状态
0	X0.0	8	X0.8
1	X0.1	9	X0.9
2	X0.2	10	X0.10
3	X0.3	11	X0.11
4	X0.4 ^{*1}	12	X0.12
5	X0.5 ^{*1}	13	X0.13
6	X0.6 ^{*1}	14	X0.14
7	X0.7 ^{*1}	15	X0.15

*1：仅 AH15PM-5A 有此输入点功能

输入端子
滤波系数设定

SR806

1. 此功能为端子硬件滤波设定
2. 输入端子，可由 SR806 之 Low Byte 内容设定输入端之滤波系数。

3. 滤波系数 = $\frac{85000}{2^{N+4}}$ (KHz) · N=1~19；下表为换算之滤波值。

N	kHz	N	kHz
1	2656.25	11	2.593994
2	1328.125	12	1.296997
3	664.0625	13	0.648499
4	332.0313	14	0.324249
5	166.0156	15	0.162125
6	83.00781	16	0.081062
7	41.50391	17	0.040531
8	20.75195	18	0.020266
9	10.37598	19	0.010133
10	5.187988		

4. 当 SR806 设定为 0 时，表示不启动外部端子之滤波功能。
5. 举例说明，SR806=16#000A 表示设定输入端子 X0.0、X0.1、X0.2、X0.3、X0.8、X0.9、X0.10、X0.11、X0.12、X0.13、X0.14、X0.15 之滤波系数为 $\frac{85000}{2^{10+4}} = 5.187988$ (KHz)；输入频率高于 5.187988KHz 之信号将会被滤除。

O100 错误
发生检查SM953、SR802、
SR803

1. 当 O100 程序发生错误时，O100 错误标志 SM953 会 SET ON，错误信息会被存在 SR802，错误 STEP 位置会存放在 SR803。
2. 错误信息请参阅附录 A 的错误信息表。

轴参数设定

SR1000 为第 1 轴参数设定，SR1100 为第 2 轴参数设定，SR1200 为第 3 轴参数设定，SR1300 为第 4 轴参数设定，第 5~16 轴以此类推。说明如下：

SRmn00
(SR1000、
SR1100、
...SR2500)

bit#	轴参数设定	bit#	轴参数设定
0	单位系设定*1	8	原点回归方向*3
1		9	原点回归模式*3
2	-	10	原点回归 DOG 下降沿检测*3
3	-	11	脉冲旋转方向*3
4	脉冲型式*2	12	相对绝对坐标*3
5		13	DOG 触发启动模式*3
6*	-	14	曲线选择*3
7	-	15	-

*1 :

b1	b0	单位系
0	0	电机单位
0	1	机械单位
1	0	复合单位
1	1	

	电机单位	复合单位	机械单位
位置	pulse	um	
	pulse	m deg	
	pulse	10 ⁻⁴ inch	
速度	pulse/sec		cm/min
	pulse/sec		10deg/min
	pulse/sec		inch/min

*2 :

b5	b4	说明
0	0	正向脉冲+反向脉冲
0	1	脉冲+方向
1	0	A/B 相脉冲 (2 相 2)
1	1	

3

轴参数设定

SRmn00
(SR1000、
SR1100、
....SR2500)

*3 :

bit#	说明
8	b[8]=0 : 执行原点回归方向 · 往当前位置 (CP) 内容值递减方向执行 b[8]=1 : 执行原点回归方向 · 往当前位置 (CP) 内容值递增方向执行
9	b[9]=0 : 正常模式 ; b[9]=1 : 覆写模式
10	b[10]=0 : 原点回归 DOG 下降沿检测 b[10]=1 : 原点回归 DOG 上升沿检测
11	b[11]=0 : 当正转运动时 · 当前位置 (CP) 内容值递增 b[11]=1 : 当正转运动时 · 当前位置 (CP) 内容值递减
12	b[12]=0 : 设定为绝对坐标定位 ; b[12]=1 : 设定为相对坐标定位
13	b[13]=0 : DOG 信号上升沿触发 b[13]=1 : DOG 信号下降沿触发 (插入单段速定位运动模式、插入二段速定位运动模式有效)
14	b[14]=0 : 为加速采梯形加速曲线 b[14]=1 : 为加速采 S 加速曲线

轴运转命令设定

SRmn30
(SR1030、
SR1130、
....SR2530)

SR1030 为第 1 轴运转命令设定 · SR1130 为第 2 轴运转命令设定 · SR1230 为第 3 轴运转命令设定 · SR1330 为第 4 轴运转命令设定 · 第 5 ~ 16 轴运转命令与 SR 装置关系以此类推 · 说明如下 :

bit#	轴运转命令	bit#	轴运转命令
0	软件 STOP	8	单段速定位运动模式启动
1	-	9	单段速插入定位运动模式启动
2	JOG+运转	10	两段速定位运动模式启动
3	JOG-运转	11	两段速插入定位运动模式启动
4	变速度运转模式启动	12	OX 设定 0 : Stop ; 1 : Start
5	手摇轮输入操作	13	-
6	原点回归模式启动	14	-
7	-	15	-

手摇轮 (MPG) 输入 : X0.8+/X0.8- · X0.9+/X0.9-

3

3

轴工作模式
设定

SRmn31
(SR1031、
SR1131、
....SR2531)

SR1031 为第 1 轴工作模式设定，SR1131 为第 2 轴工作模式设定，SR1231 为第 3 轴工作模式设定，SR1331 为第 4 轴工作模式设定，第 5~16 轴工作模式与 SR 装置关系以此类推。说明如下：

bit#	轴工作模式	bit#	轴工作模式
0	-	8	-
1	-	9	-
2	CLR 信号输出模式	10	-
3	-	11	-
4	-	12	-
5	-	13	-
6	-	14	-
7	-	15	回归出厂设定

bit#	说明
2	当 b[2]=0 时，输出信号 CLR 为原点回归完成时输出 130ms 给 Servo 的信号，做为伺服内部位置寄存器的清除信号。 当 b[2]=1 时，CLR 输出点做为一般输出点，状态由 b[3]控制 ON/OFF
15	当 b[15]=1 时，所有参数恢复出厂值设定。

轴执行状态

SRmn40
(SR1040、
SR1140、
....SR2540)

SR1040 为第 1 轴执行状态设定，SR1140 为第 2 轴执行状态设定，SR1240 为第 3 轴执行状态设定，SR1340 为第 4 轴执行状态设定，第 5~16 轴执行状态与 SR 装置关系以此类推。说明如下：

bit#	轴执行状态
0	正向脉冲输出中
1	反向脉冲输出中
2	行程动作中
3	错误产生
4	行程暂停中
5	-
6	-
7	-

3.14 运动模式特殊寄存器[SR]

运动模式特殊寄存器 (SR) · 它的种类及功能如下所示。详细说明可参考第 7 章的功能说明 · 使用者可参考各特殊寄存器之使用说明 · 进一步设定参数值以及了解系统信息。

SR 编号*1		特殊寄存器名称	设定范围	出厂值
HW	LW			
-	mn00	参数设定	b0~b15	16#0
-	mn01	背隙补偿	依需求设定对应值	16#0
mn03	mn02	电机转一圈所需脉冲数 (A)	1~+2,147,483,647 PLS/REV	K2,000
mn05	mn04	电机转一圈之移动距离 (B)	1~+2,147,483,647*2	K1,000
mn07	mn06	最高速度 V_{MAX}	0~+2,147,483,647*3	K500,000
mn09	mn08	启动速度 V_{BIAS}	0~+2,147,483,647*3	K0
mn11	mn10	寸动 JOG 速度 V_{JOG}	0~+2,147,483,647*3	K5,000
mn13	mn12	原点回归速度 V_{RT}	0~+2,147,483,647*3	K50,000
mn15	mn14	原点回归减速速度 V_{CR}	0~+2,147,483,647*3	K1,000
-	mn16	零点 (PG0) 信号数 N	0~+32,767 PLS	K0
-	mn17	脉冲信号数 P	-32,768~+32,767 PLS	K0
mn19	mn18	原点位置定义 HP	0~±999,999	K0
-	mn20	加速时间 T_{ACC}	10~+32,767 ms	K100
-	mn21	减速时间 T_{DEC}	10~+32,767 ms	K100
mn23	mn22	目标位置 (I) P (I)	-2,147,483,648~+2,147,483,647	K0
mn25	mn24	运转速度 (I) V (I)	0~+2,147,483,647	K1000
mn27	mn26	目标位置 (II) P (II)	-2,147,483,648~+2,147,483,647	K0
mn29	mn28	运转速度 (II) V (II)	0~+2,147,483,647*2	K2,000
-	mn30	运转命令	b0~b15	16#0
-	mn31	工作模式	b0~b15	16#0
mn33	mn32	当前位置 CP (PLS)	-2,147,483,648~+2,147,483,647	K0
mn35	mn34	当前速度 CS (PPS)	0~+2,147,483,647 PPS	K0
mn37	mn36	当前位置 CP (unit*3)	-2,147,483,648~+2,147,483,647	K0
mn39	mn38	当前速度 CS (unit*3)	0~+2,147,483,647 PPS	K0
-	mn40	执行状态	b0~b15	16#0
-	mn41	错误编号	请参考错误码表	16#0
-	mn42	电子齿轮 (分子)	1~+32,767	K1
-	mn43	电子齿轮 (分母)	1~+32,767	K1

3

SR 编号*1		特殊寄存器名称	设定范围	出厂值
HW	LW			
mn45	mn44	手摇轮输入频率	由手摇轮输入的脉冲频率	K0
mn47	mn46	累计手摇轮输入脉冲个数	计数由手摇轮输入的脉冲个数	K0
-	mn48	手摇轮输入响应速度	手摇轮输入响应速度	K5
-	mn49	停止模式 (OX0-99)	依需求设定对应值	K0
mn51	mn50	电子原点	依需求设定对应值	K0
-	mn52	程序号码指定方式	依需求设定对应值	K0
-	mn53	Ox 错误 STEP 位置	依需求设定对应值	K0
mn69	mn68	DMCNET 伺服编码器当前位置*4	依台达 A2 伺服设定显示对应值	K0
-	mn72	DMCNET 伺服控制命令*4	依需求设定对应值	K0
-	mn73	DMCNET 伺服状态*4	依需求设定对应值	K0
mn75	mn74	DMCNET 伺服错误码*4	依需求设定对应值	K0
mn77	mn76	DMCNET 伺服参数读写 Data*4	依需求设定对应值	K0
-	mn78	DMCNET 伺服参数位置*4	依需求设定对应值	K0

*1. HW : High Word ; LW : Low Word ; mn= 10 (第 1 轴) ~25 (第 16 轴)。

*2. 单位可选择 um/rev、m deg/rev 及 10⁻⁴ inch/rev。

*3. 设定单位依照 SRmn00“参数设定”之 b0、b1 单位系设定而变化。

*4. 仅支持 AH20MC-5A、不支持 AH05PM/15PM-5A/10PM-5A。

运动模式特殊寄存器功能说明：

有关运动模式的 SR 装置详细功能以及使用方式说明参考第 7 章。

AH20MC-5A 机种，有关 DMCNET 的 SR 装置详细功能介绍以及使用方式说明参考第 14 章。

4

第4章 基本顺序指令

目录

4.1	基本指令一览表	4-2
4.2	基本指令说明	4-4

4.1 基本指令一览表

一般指令


指令码	功能	操作数	执行速度 (us)				STEP	页码
			20MC ^{*1}	10PM ^{*1}	05PM ^{*1}	15PM ^{*1}		
LD	载入 A 接点	Xn.n、Yn.n、M、S、T、C	0.22	0.22	0.22	0.11	3	4-4
LDI	载入 B 接点	Xn.n、Yn.n、M、S、T、C	0.22	0.22	0.22	0.11	3	4-4
AND	串联 A 接点	Xn.n、Yn.n、M、S、T、C	0.22	0.22	0.22	0.11	3	4-5
ANI	串联 B 接点	Xn.n、Yn.n、M、S、T、C	0.22	0.22	0.22	0.11	3	4-5
OR	并联 A 接点	Xn.n、Yn.n、M、S、T、C	0.22	0.22	0.22	0.11	3	4-6
ORI	并联 B 接点	Xn.n、Yn.n、M、S、T、C	0.22	0.22	0.22	0.11	3	4-6
ANB	串联回路方块	无	0.12	0.12	0.12	0.25	3	4-7
ORB	并联回路方块	无	0.12	0.12	0.12	0.25	3	4-7

输出指令

指令码	功能	操作数	执行速度 (us)				STEP	页码
			20MC ^{*1}	10PM ^{*1}	05PM ^{*1}	15PM ^{*1}		
OUT	驱动线圈	Yn.n、M、S	0.17	0.17	0.17	0.11	3	4-8
SET	动作保持 (ON)	Yn.n、M、S	0.39	0.39	0.39	0.23	3	4-8
RST	接点或寄存器清除	Yn.n、M、S、T、C、D、V、Z、W	0.47	0.47	0.47	0.32	3	4-9

定时器、计数器


指令码	功能	操作数	执行速度 (us)				STEP	页码
			20MC ^{*1}	10PM ^{*1}	05PM ^{*1}	15PM ^{*1}		
TMR	16 位定时器	T-K、T-D 或 T-W	5.6	5.6	5.6	5.2	5	4-9
CNT	16 位计数器	C-K、C-D 或 C-W (16 位)	3.2	3.2	3.2	2.8	5	4-10
DCNT	32 位计数器	C-K、C-D 或 C-W (32 位)	3.8	3.8	3.8	2.8	6	4-10

 接点上升沿/下降沿检出指令

指令码	功能	操作数	执行速度 (us)				STEP	页码
			20MC ^{*1}	10PM ^{*1}	05PM ^{*1}	15PM ^{*1}		
LDP	上升沿检出动作开始	Xn.n、Yn.n、M、S、T、C	0.39	0.39	0.39	0.23	3	4-11
LDF	下降沿检出动作开始	Xn.n、Yn.n、M、S、T、C	0.39	0.39	0.39	0.23	3	4-11
ANDP	上升沿检出串联连接	Xn.n、Yn.n、M、S、T、C	0.39	0.39	0.39	0.23	3	4-12
ANDF	下降沿检出串联连接	Xn.n、Yn.n、M、S、T、C	0.39	0.39	0.39	0.23	3	4-12
ORP	上升沿检出并联连接	Xn.n、Yn.n、M、S、T、C	0.37	0.37	0.37	0.04	3	4-13
ORF	下降沿检出并联连接	Xn.n、Yn.n、M、S、T、C	0.37	0.37	0.37	0.04	3	4-13

 上下微分输出指令

指令码	功能	操作数	执行速度 (us)				STEP	页码
			20MC ^{*1}	10PM ^{*1}	05PM ^{*1}	15PM ^{*1}		
PLS	上微分输出	Yn.n、M	0.37	0.37	0.37	0.23	3	4-14
PLF	下微分输出	Yn.n、M	0.37	0.37	0.37	0.23	3	4-15

 其它指令

指令码	功能	操作数	执行速度 (us)				STEP	页码
			20MC ^{*1}	10PM ^{*1}	05PM ^{*1}	15PM ^{*1}		
P	指针	P0~P255	-	-	-	-	1	4-15

*1. 05PM=AH05PM-5A ; 15PM=AH15PM-5A ; 10PM=AH10PM-5A ; 20MC=AH20MC-5A。

4.2 基本指令说明

指令码		操作数										功能	
LD		S										载入	

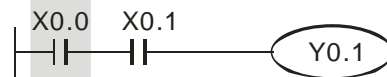
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●								●	●		

指令说明

- ◆ LD 指令用于左母线开始的 A 接点或一个接点回路块开始的 A 接点，它的作用是把当前内容保存，同时把取来的接点状态存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
AND X0.1 串联 X0.1 之 A 接点
OUT Y0.1 驱动 Y0.1 线圈

指令码		操作数										功能	
LDI		S										载入 B 接点	

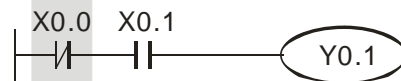
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●								●	●		

指令说明

- ◆ LDI 指令用于左母线开始的 B 接点或一个接点回路块开始的 B 接点，它的作用是把当前内容保存，同时把取来的接点状态存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

LDI X0.0 载入 X0.0 之 B 接点
AND X0.1 串联 X0.1 之 A 接点
OUT Y0.1 驱动 Y0.1 线圈

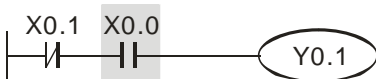
指令码		操作数										功能				
AND		S										串联 A 接点				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S	●	●	●	●								●	●			

指令说明

◆ AND 指令用于 A 接点的串联连接，先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“及”(AND)的运算，并将结果存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

```
LDI  X0.1  载入 X0.1 之 B 接点
AND  X0.0  串联 X0.0 之 A 接点
OUT  Y0.1  驱动 Y0.1 线圈
```

指令码		操作数										功能				
ANI		S										串联 B 接点				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S	●	●	●	●								●	●			

指令说明

◆ ANI 指令用于 B 接点的串联连接，它的作用是先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“及”(AND)的运算，并将结果存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

```
LD  X0.1  载入 X0.1 之 A 接点
ANI  X0.0  串联 X0.0 之 B 接点
OUT  Y0.1  驱动 Y0.1 线圈
```



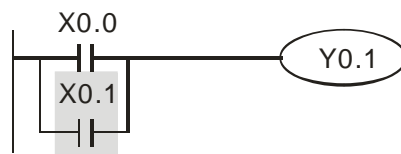
指令码		操作数										功能			
OR		S										并联 A 接点			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●								●	●		

指令说明

- ◆ OR 指令用于 A 接点的并联连接，它的作用是先读取目前所指定并联接点的状态再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
OR X0.1 并联 X0.1 之 A 接点
 OUT Y0.1 驱动 Y0.1 线圈

指令码		操作数										功能			
ORI		S										并联 B 接点			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●								●	●		

指令说明

- ◆ ORI 指令用于 B 接点的并联连接，它的作用是先读取目前所指定并联接点的状态再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
ORI X0.1 并联 X0.1 之 B 接点
 OUT Y0.1 驱动 Y0.1 线圈

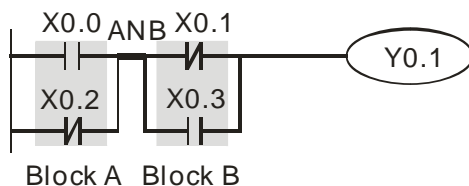
指令码	操作数	功能
ANB	-	串联回路方块

指令说明

◆ ANB 是将前一保存的逻辑结果与目前累积寄存器的内容作“及” (AND) 的运算。

程序范例

梯形图：



指令码： 说明：

- LD X0.0 载入 X0.0 之 A 接点
- ORI X0.2 并联 X0.2 之 B 接点
- LDI X0.1 载入 X0.1 之 B 接点
- OR X0.3 并联 X0.3 之 A 接点
- ANB** 串联回路方块
- OUT Y0.1 驱动 Y0.1 线圈

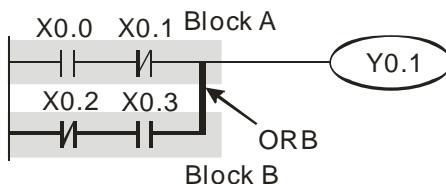
指令码	操作数	功能
ORB	-	并联回路方块

指令说明

◆ ORB 是将前一保存的逻辑结果与目前累积寄存器的内容作“或” (OR) 的运算。

程序范例

梯形图：



指令码： 说明：

- LD X0.0 载入 X0.0 之 A 接点
- ANI X0.1 串联 X0.1 之 B 接点
- LDI X0.2 载入 X0.2 之 B 接点
- AND X0.3 串联 X0.3 之 A 接点
- ORB** 并联回路方块
- OUT Y0.1 驱动 Y0.1 线圈

指令码				操作数							功能					
OUT				S							驱动线圈					
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S		●	●	●												

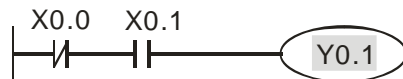
指令说明

- ◆ 将 OUT 指令之前的逻辑运算结果输出至指定的组件。
- ◆ 线圈接点动作：

运算结果	OUT 指令		
	线圈	接点	
		A 接点 (常开)	B 接点 (常闭)
FALSE	OFF	不导通	导通
TRUE	ON	导通	不导通

程序范例

梯形图：



指令码： 说明：

```
LDI  X0.0  载入 X0.0 之 B 接点
AND  X0.1  串联 X0.1 之 A 接点
OUT  Y0.1  驱动 Y0.1 线圈
```

指令码				操作数							功能					
SET				S							动作保持 (ON)					
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S		●	●	●												

指令说明

- ◆ 当 SET 指令被驱动，其指定的组件被设定为 ON，且被设定的组件会维持 ON，不管 SET 指令是否仍被驱动。可利用 RST 指令将该组件设为 OFF。

程序范例

梯形图：



指令码： 说明：

```
LD  X0.0  载入 X0.0 之 A 接点
ANI Y0.0  串联 Y0.0 之 B 接点
SET Y0.1  Y0.1 动作保持 ( ON )
```

指令码		操作数										功能				
RST		S										接点或寄存器清除				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S		●	●	●						●	●	●	●	●	●	

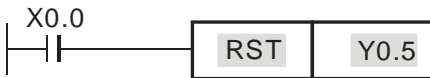
指令说明

◆ 当 RST 指令被驱动，其指定的组件的动作如下：

组件	状态
S · Yn.n · M	线圈及接点都会被设定为 OFF。
T · C	目前计时或计数值会被设为 0，且线圈及接点都会被设定为 OFF。
D · V · Z	内容值会被设为 0。

◆ 若 RST 指令没有被执行，其指定组件的状态保持不变。

程序范例

梯形图： 指令码： LD X0.0 RST Y0.5 说明： 载入 X0.0 之 A 接点 Y0.5 接点清除



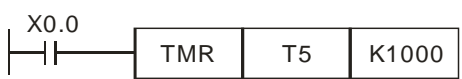
指令码		操作数										功能				
TMR		S1,S2										16 位定时器				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S1												●				
S2					●					●	●					

指令说明

◆ 当 TMR 指令执行时，其所指定的定时器线圈受电，定时器开始计时，当到达所指定的定时值（计时值>=设定值），其接点动作如下：

NO (Normally Open) 接点	闭合
NC (Normally Closed) 接点	开路

程序范例

梯形图： 指令码： LD X0.0 TMR T5 K1000 说明： 载入 X0.0 之 A 接点 T5 定时器 设定值为 K1000

补充说明

◆ 定时器操作数 T 使用范围请参考各系列机种功能规格表。

指令码		操作数										功能			
CNT		S1,S2										16 位计数器			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S1													●		
S2					●					●	●				

指令说明

- ◆ 当 CNT 指令由 OFF→ON 执行，表示所指定的计数器线圈由失电→受电，则该计数器计数值加 1，当计数到达所指定的定数值（计数值=设定值），其接点动作如下：

NO (Normally Open) 接点	闭合
NC (Normally Closed) 接点	开路

- ◆ 当计数到达之后，若再有计数脉冲输入，其接点及计数值均保持不变，若要重新计数或作清除的动作，请利用 RST 指令。

程序范例

梯形图：

指令码：

说明：



LD X0.0 载入 X0.0 之 A 接点

CNT C20 K100 C20 计数器设定值为 K100

指令码		操作数										功能			
DCNT		S1,S2										32 位计数器			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S1													●		
S2					●					●	●				

指令说明

- ◆ DCNT 为 32 位计数器 C200 至 C255 之启动指令。
- ◆ C200~C220 为特殊计数器，详见第十章高速计数与计时功能。
- ◆ 一般用加减算计数器 C221~C255，当 DCNT 指令由 OFF→ON 时，计数器之当前值将执行上数(加一)的动作或下数(减一)的动作，依 SM221~SM255 的设定模式。

程序范例

梯形图：

指令码：

说明：



LD M0 载入 M0 之 A 接点

DCNT C254 K1000 C254 计数器设定值为 K1000

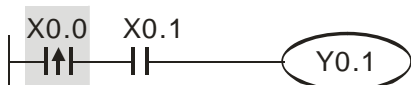
指令码		操作数										功能				
LDP		S										上升沿检出动作开始				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S	●	●	●	●								●	●			

指令说明

- ◆ LDP 指令用法上与 LD 相同，但动作不同，它的作用是指当前内容保存，同时把取来的接点上升沿检出状态存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

LDP X0.0 X0.0 上升沿检出动作开始
 AND X0.1 串联 X0.1 之 A 接点
 OUT Y0.1 驱动 Y0.1 线圈

补充说明

- ◆ 各操作数使用范围请参考各系列机种功能规格表。
- ◆ 若 AH500 运动模块电源开启前，指定上升沿接点的状态为 ON，则电源开启后该上升沿接点为 TRUE。

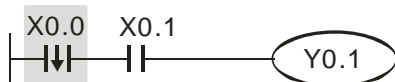
指令码		操作数										功能				
LDF		S										下降沿检出动作开始				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S	●	●	●	●								●	●			

指令说明

- ◆ LDF 指令用法上与 LD 相同，但动作不同，它的作用是指当前内容保存，同时把取来的接点下降沿检出状态存入累积寄存器内。

程序范例

梯形图：



指令码： 说明：

LDF X0.0 X0.0 下降沿检出动作开始
 AND X0.1 串联 X0.1 之 A 接点
 OUT Y0.1 驱动 Y0.1 线圈



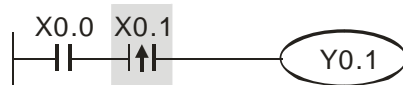
指令码		操作数										功能			
ANDP		S										上升沿检出串联连接			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●								●	●		

指令说明

◆ ANDP 指令用于接点上升沿检出的串联连接。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
ANDP X0.1 X0.1 上升沿检出串联连接
 OUT Y0.1 驱动 Y0.1 线圈

4

指令码		操作数										功能			
ANDF		S										下降沿检出串联连接			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●								●	●		

指令说明

◆ ANDF 指令用于接点下降沿检出的串联连接。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
ANDF X0.1 X0.1 下降沿检出串联连接
 OUT Y0.1 驱动 Y0.1 线圈

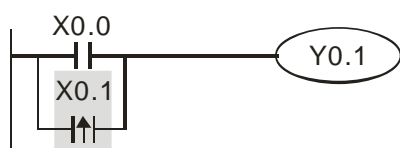
指令码		操作数											功能			
ORP		S											上升沿检出并联连接			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S	●	●	●	●								●	●			

指令说明

◆ ORP 指令用于接点上升沿检出的并联连接。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
ORP X0.1 X0.1 上升沿检出并联连接
 OUT Y0.1 驱动 Y0.1 线圈

4

指令码		操作数											功能			
ORF		S											下降沿检出并联连接			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S	●	●	●	●								●	●			

指令说明

◆ ORF 指令用于接点下降沿检出的并联连接。

程序范例

梯形图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点
ORF X0.1 X0.1 下降沿检出并联连接
 OUT Y0.1 驱动 Y0.1 线圈

指令码		操作数										功能			
PLS		S										上微分输出			

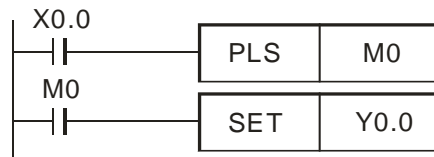
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S		●	●												

指令说明

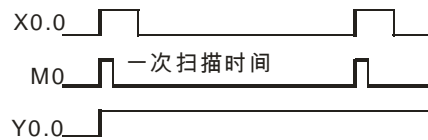
◆ 上微分输出指令。当 X0.0=OFF→ON(上升沿触发)时 PLS 指令被执行，M0 送出一脉冲，脉冲长度为一次扫描时间。

程序范例

梯形图：



时序图：



指令码： 说明：

LD X0.0 载入 X0.0 之 A 接点

PLS M0 M0 上微分输出

LD M0 载入 M0 之 A 接点

SET Y0.0 Y0.0 动作保持 (ON)



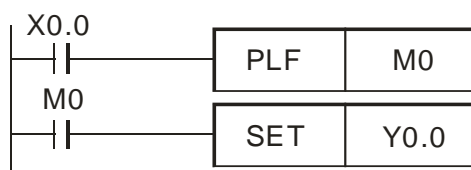
指令码		操作数										功能			
PLF		S										下微分输出			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S		●	●												

指令说明

◆ 下微分输出指令。当 X0.0=ON→OFF (下降沿触发) 时 PLF 指令被执行, M0 送出一脉冲, 脉冲长度为一次扫描时间。

程序范例

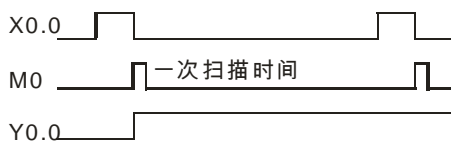
梯形图:



指令码: 说明:

LD X0.0 载入 X0.0 之 A 接点
PLF M0 M0 下微分输出
 LD M0 载入 M0 之 A 接点
 SET Y0.0 Y0.0 动作保持 (ON)

时序图:



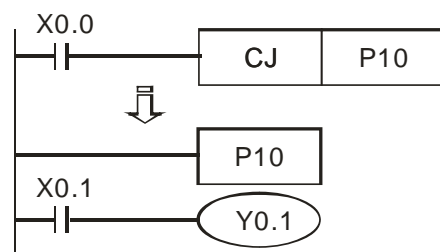
指令码	操作数	功能
P	-	指针

指令说明

1. 操作数 P 为指定子程序指针编号, 范围为 P0~P255。
2. 指针 P 用于 API 00 CJ、API 01 CALL、API 256 CJN、API 257 JMP 使用, 不需从编号 0 开始, 但是编号不能重复使用, 否则会发生不可预期的错误。

程序范例

梯形图:



指令码: 说明:

LD X0.0 载入 X0.0 之 A 接点
 CJ P10 跳转指令 CJ 到 P10
 :
P10 指针 P10
 LD X0.1 载入 X0.1 之 A 接点
 OUT Y0.1 驱动 Y0.1 线圈



MEMO

4

5

第5章 应用指令分类及基本使用

目录

5.1 应用指令一览表	5-4
5.2 应用指令的组成	5-8
5.3 应用指令对数值的处理方式	5-12
5.4 使用变址寄存器 V、Z 来修饰操作数	5-15
5.5 指令索引	5-16
5.6 应用指令说明	5-20
5.7 运动功能块一览表	5-129
5.8 运动功能块引脚介绍	5-131
5.8.1 输入/输出引脚功能定义	5-131
5.8.2 输入/输出引脚时序图	5-132
5.8.3 PMSOft 软件操作介绍	5-133
5.9 台达定义参数一览表	5-136
5.10 单轴运动功能块说明	5-139
5.10.1 绝对单段速运动	5-139
5.10.2 相对单段速运动	5-142
5.10.3 绝对两段速运动	5-145
5.10.4 相对两段速运动	5-148
5.10.5 插入单段速运动	5-150
5.10.6 插入两段速运动	5-154
5.10.7 寸动运动	5-156
5.10.8 手摇轮模式运动	5-159
5.10.9 原点回归运动	5-160
5.10.10 单轴运动停止	5-162
5.10.11 运动轴参数设定 I	5-164
5.10.12 运动轴参数设定 II	5-165

5.10.13 读取当前位置/速度	5-167
5.10.14 轴状态信息.....	5-168
5.10.15 设定当前位置	5-170
5.10.16 输入极性设定	5-171
5.10.17 电子齿轮运动	5-174
5.10.18 凸轮啮合	5-176
5.10.19 凸轮数据点读取	5-178
5.10.20 凸轮数据点写入	5-179
5.10.21 凸轮同步速率计算	5-181
5.10.22 凸轮曲线规划	5-182
5.10.23 凸轮曲线更新	5-184
5.11 多轴运动功能块说明	5-186
5.11.1 G 码参数设定	5-186
5.11.2 G 码运行	5-187
5.11.3 G 码停止	5-189
5.11.4 M 码读取	5-191
5.11.5 多轴绝对直线插补	5-193
5.11.6 多轴相对直线插补	5-194
5.11.7 多轴直线插补停止	5-196
5.12 网络功能块说明.....	5-198
5.12.1 伺服启动/停止.....	5-198
5.12.2 伺服重置	5-199
5.12.3 伺服参数写入	5-200
5.12.4 伺服参数读取	5-202
5.12.5 伺服回原点命令	5-204
5.12.6 伺服 DMCNET 通讯初始化	5-206
5.12.7 伺服捕捉功能	5-207
5.12.8 Ethernet IP Address 设定	5-209
5.13 其它功能块说明.....	5-210
5.13.1 SD 主程序备份	5-210
5.13.2 SD 装置备份	5-211
5.13.3 SD 装置回存	5-212

5.13.4	高速计数器	5-213
5.13.5	高速定时器	5-214
5.13.6	高速比较设定	5-216
5.13.7	高速比较重置	5-218
5.13.8	高速捕捉设定	5-219
5.13.9	高速捕捉遮蔽	5-222
5.13.10	中断设定	5-223
5.13.11	绝对型编码器	5-224

5.1 应用指令一览表

分类	API	指令码		P 指令	功能	支持机种			STEPS		页码
		16 位	32 位			20 MC	10PM/ 15PM	05 PM	16 位	32 位	
回路 控制	00	CJ	-	✓	条件跳转	✓	✓	✓	3	-	5-20
	01	CALL	-	✓	调用子程序	✓	✓	✓	3	-	5-23
	02	SRET	-	-	子程序结束	✓	✓	✓	1	-	5-24
	07	WDT	-	✓	逾时监视定时器	✓	✓	✓	1	-	5-26
	08	RPT	-	-	循环回路起始 (只能一层)	✓	✓	✓	3	-	5-27
	09	RPE	-	-	循环回路结束	✓	✓	✓	1	-	5-28
传 送 比 较	10	CMP	DCMP	✓	比较设定输出	✓	✓	✓	7	9	5-29
	11	ZCP	DZCP	✓	区域比较	✓	✓	✓	9	12	5-30
	12	MOV	DMOV	✓	数据移动	✓	✓	✓	5	6	5-31
	14	CML	DCML	✓	反转传送	✓	✓	✓	5	6	5-32
	15	BMOV	-	✓	全部传送	✓	✓	✓	7	-	5-33
	16	FMOV	DFMOV	✓	多点移动	✓	✓	✓	7	8	5-35
	17	XCH	DXCH	✓	数据的交换	✓	✓	✓	5	9	5-36
	18	BCD	DBCD	✓	BIN→BCD 变换	✓	✓	✓	5	5	5-37
19	BIN	DBIN	✓	BCD→BIN 变换	✓	✓	✓	5	5	5-38	
四 则 运 算	20	ADD	DADD	✓	BIN 加法	✓	✓	✓	7	9	5-39
	21	SUB	DSUB	✓	BIN 减法	✓	✓	✓	7	9	5-41
	22	MUL	DMUL	✓	BIN 乘法	✓	✓	✓	7	9	5-42
	23	DIV	DDIV	✓	BIN 除法	✓	✓	✓	7	9	5-43
	24	INC	DINC	✓	BIN 加一	✓	✓	✓	3	3	5-44
	25	DEC	DDEC	✓	BIN 减一	✓	✓	✓	3	3	5-45
	26	WAND	DWAND	✓	逻辑及 (AND) 运算	✓	✓	✓	7	9	5-46
	27	WOR	DWOR	✓	逻辑或 (OR) 运算	✓	✓	✓	7	9	5-47
	28	WXOR	DWXOR	✓	逻辑异或 (XOR) 运算	✓	✓	✓	7	9	5-48
29	NEG	DNEG	✓	取负数 (取 2 的补码)	✓	✓	✓	3	3	5-49	
旋 转 位 移	30	ROR	DROR	✓	右旋转	✓	✓	✓	5	6	5-51
	31	ROL	DROL	✓	左旋转	✓	✓	✓	5	6	5-52
	32	RCR	DRCR	✓	附进位标志右旋转	✓	✓	✓	5	6	5-53
	33	RCL	DRCL	✓	附进位标志左旋转	✓	✓	✓	5	6	5-54
	34	SFTR	-	✓	位右移	✓	✓	✓	9	-	5-55
	35	SFTL	-	✓	位左移	✓	✓	✓	9	-	5-56
	36	WSFR	-	✓	寄存器右移	✓	✓	✓	9	-	5-57
	37	WSFL	-	✓	寄存器左移	✓	✓	✓	9	-	5-58

分类	API	指令码		P 指令	功能	支持机种			STEPS		页码
		16 位	32 位			20 MC	10PM/ 15PM	05 PM	16 位	32 位	
旋转 位移	38	SFWR	-	✓	位移写入	✓	✓	✓	7	-	5-59
	39	SFRD	-	✓	位移读出	✓	✓	✓	7	-	5-60
数据	40	ZRST	-	✓	区域清除	✓	✓	✓	5	-	5-61
	41	DECO	-	✓	译码器	✓	✓	✓	7	-	5-62
	42	ENCO	-	✓	编码器	✓	✓	✓	7	-	5-64
	43	SUM	DSUM	✓	ON 位数量	✓	✓	✓	5	5	5-66
	44	BON	DBON	✓	ON 位判定	✓	✓	✓	7	8	5-67
	45	MEAN	DMEAN	✓	平均值	✓	✓	✓	7	8	5-68
	46	ANS	-	-	警报点输出	✓	✓	✓	7	-	5-69
	47	ANR	-	✓	警报点复位	✓	✓	✓	1	-	5-70
	48	SQR	DSQR	✓	BIN 开平方根	✓	✓	✓	5	6	5-71
49	-	DFLT	✓	BIN 整数→二进浮点数变换	✓	✓	✓	-	6	5-72	
高速 处理	50	REF	-	✓	I/O 更新处理	✓	✓	✓	5	-	5-73
便利	61	SER	DSER	✓	多点比较	✓	✓	✓	9	11	5-74
	66	ALT	-	✓	ON/OFF 交替	✓	✓	✓	3	-	5-75
I/O	78	FROM	DFROM	✓	I/O 模块 CR 数据读出	✓	✓	✓	9	12	5-76
	79	TO	DTO	✓	I/O 模块 CR 数据写入	✓	✓	✓	9	13	5-77
	87	ABS	DABS	✓	绝对值	✓	✓	✓	3	3	5-78
浮点 运算	110	-	DECMP	✓	二进浮点数比较	✓	✓	✓	-	9	5-79
	111	-	DEZCP	✓	二进浮点数区域比较	✓	✓	✓	-	12	5-80
	112	-	DMOV	✓	浮点数值数据移动	✓	✓	✓	-	6	5-81
	116	-	DRAD	✓	角度→弧度	✓	✓	✓	-	6	5-82
	117	-	DDEG	✓	弧度→角度	✓	✓	✓	-	6	5-83
	120	-	DEADD	✓	二进浮点数加法	✓	✓	✓	-	9	5-84
	121	-	DESUB	✓	二进浮点数减法	✓	✓	✓	-	9	5-85
	122	-	DEMUL	✓	二进浮点数乘法	✓	✓	✓	-	9	5-86
	123	-	DEDIV	✓	二进浮点数除法	✓	✓	✓	-	9	5-87
	124	-	DEXP	✓	二进浮点数取指数	✓	✓	✓	-	6	5-88
125	-	DLN	✓	二进浮点数取自然对数	✓	✓	✓	-	6	5-89	

分类	API	指令码		P 指令	功能	支持机种			STEPS		页码
		16 位	32 位			20 MC	10PM/ 15PM	05 PM	16 位	32 位	
浮 点 运 算	126	-	DLOG	✓	二进浮点数取对数	✓	✓	✓	-	9	5-90
	127	-	DESQR	✓	二进浮点数开平方根	✓	✓	✓	-	6	5-91
	128	-	DPOW	✓	浮点数权值指令	✓	✓	✓	-	9	5-92
	129	-	DINT	✓	二进浮点数→BIN 整数变换	✓	✓	✓	-	5	5-93
	130	-	DSIN	✓	二进浮点数→SIN 运算	✓	✓	✓	-	6	5-94
	131	-	DCOS	✓	二进浮点数→COS 运算	✓	✓	✓	-	6	5-96
	132	-	DTAN	✓	二进浮点数→TAN 运算	✓	✓	✓	-	6	5-98
	133	-	DASIN	✓	二进浮点数→ASIN 运算	✓	✓	✓	-	6	5-100
	134	-	DACOS	✓	二进浮点数→ACOS 运算	✓	✓	✓	-	6	5-101
	135	-	DATAN	✓	二进浮点数→ATAN 运算	✓	✓	✓	-	6	5-102
	136	-	DSINH	✓	二进浮点数→SINH 运算	✓	✓	✓	-	6	5-103
	137	-	DCOSH	✓	二进浮点数→COSH 运算	✓	✓	✓	-	6	5-104
	138	-	DTANH	✓	二进浮点数→TANH 运算	✓	✓	✓	-	6	5-105
	172	-	DADDR	✓	浮点数值加法	✓	✓	✓	-	9	5-106
	173	-	DSUBR	✓	浮点数值减法	✓	✓	✓	-	9	5-107
174	-	DMULR	✓	浮点数值乘法	✓	✓	✓	-	9	5-108	
175	-	DDIVR	✓	浮点数值除法	✓	✓	✓	-	9	5-109	
接 点 型 态 逻 辑 运 算	215	LD&	DLD&	-	S1&S2	✓	✓	✓	5	7	5-110
	216	LD	DLD	-	S1 S2	✓	✓	✓	5	7	5-110
	217	LD^	DLD^	-	S1^S2	✓	✓	✓	5	7	5-110
	218	AND&	DAND&	-	S1&S2	✓	✓	✓	5	7	5-111
	219	AND	DAND	-	S1 S2	✓	✓	✓	5	7	5-111
	220	AND^	DAND^	-	S1^S2	✓	✓	✓	5	7	5-111
	221	OR&	DOR&	-	S1&S2	✓	✓	✓	5	7	5-112
	222	OR	DOR	-	S1 S2	✓	✓	✓	5	7	5-112
	223	OR^	DOR^	-	S1^S2	✓	✓	✓	5	7	5-112
接 点 型 态 比 较 指 令	224	LD=	DLD=	-	S1 = S2	✓	✓	✓	5	7	5-113
	225	LD>	DLD>	-	S1 > S2	✓	✓	✓	5	7	5-113
	226	LD<	DLD<	-	S1 < S2	✓	✓	✓	5	7	5-113
	228	LD<>	DLD<>	-	S1≠S2	✓	✓	✓	5	7	5-113
	229	LD≤	DLD≤	-	S1≤S2	✓	✓	✓	5	7	5-113
	230	LD≥	DLD≥	-	S1≥S2	✓	✓	✓	5	7	5-113
	232	AND=	DAND=	-	S1 = S2	✓	✓	✓	5	7	5-114
	233	AND>	DAND>	-	S1 > S2	✓	✓	✓	5	7	5-114
接	234	AND<	DAND<	-	S1 < S2	✓	✓	✓	5	7	5-114
	236	AND<>	DAND<>	-	S1≠S2	✓	✓	✓	5	7	5-114

分类	API	指令码		P 指令	功能	支持机种			STEPS		页码
		16 位	32 位			20 MC	10PM/ 15PM	05 PM	16 位	32 位	
点 型 态 比 较 指 令	237	AND<=	DAND<=	-	$S1 \leq S2$	✓	✓	✓	5	7	5-114
	238	AND>=	DAND>=	-	$S1 \geq S2$	✓	✓	✓	5	7	5-114
	240	OR=	DOR=	-	$S1 = S2$	✓	✓	✓	5	7	5-115
	241	OR>	DOR>	-	$S1 > S2$	✓	✓	✓	5	7	5-115
	242	OR<	DOR<	-	$S1 < S2$	✓	✓	✓	5	7	5-115
	244	OR<>	DOR<>	-	$S1 \neq S2$	✓	✓	✓	5	7	5-115
	245	OR<=	DOR<=	-	$S1 \leq S2$	✓	✓	✓	5	7	5-115
	246	OR>=	DOR>=	-	$S1 \geq S2$	✓	✓	✓	5	7	5-115
其 它 指 令	152	SWAP	DSWAP	✓	上/下字节交换	✓	✓	✓	3	3	5-116
	154	RAND	DRAND	✓	随机数值产生	✓	✓	✓	7	9	5-117
	202	SCAL	-	✓	比例值运算	✓	✓	✓	7	-	5-118
	203	SCLP	DSCLP	✓	参数型比例值运算	✓	✓	✓	7	9	5-120
	256	CJN	-	✓	反条件跳转	✓	✓	✓	3	-	5-124
	257	JMP	-	-	无条件跳转	✓	✓	✓	3	-	5-125
	258	BRET	-	-	回母线	✓	✓	✓	1	-	5-126
	259	MMOV	-	✓	16→32 位数值转换	✓	✓	✓	6	-	5-127
260	RMOV	-	✓	32→16 位数值转换	✓	✓	✓	6	-	5-128	

补充说明：05PM=AH05PM-5A · 15PM=AH15PM-5A · 10PM=AH10PM-5A · 20MC=AH20MC-5A

5.2 应用指令的组成

- ◆ 应用指令的结构可分为两部份：指令名及操作数

指令名： 表示指令执行功能

操作数： 表示该指令运算处理的装置

应用指令的指令部份通常占 1 个地址 (Step)，而 1 个操作数会根据 16 位指令或 32 位指令的不同占 2 或 3 个地址，32 位指令的 K、16#和 F 都是占 3 个地址，其余都是 2 个地址。

- ◆ 应用指令的格式说明

1. PLC 指令提供一个特定的指令码及 API 编号，以便存储。下面的表格中指令的 API 编号为 012。指令码为 MOV，MOV 的功能描述是“数据移动”。

API	指令码			操作数	功能
012	D	MOV	P	S · D	数据移动

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

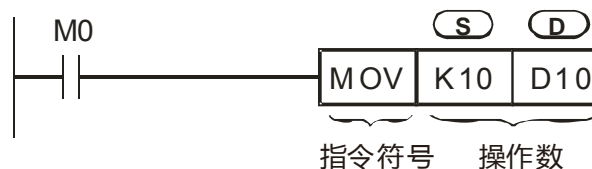
脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

2. 操作数区域会列出各种指令所用到的装置。S · D · n · m 会根据它的功能使用于操作数，如果不只一个操作数并且功能相同时，会附加编号使用，如 S₁ · S₂...
3. 脉冲执行型指令要求在指令码后面加上“P”，而 32 位指令要求在指令码前面加上“D”，如“D***P”，“***”是指令码。
4. 操作数区域中，32 位浮点数表示为 F。
5. 上表中含●的装置支持 V 及 Z 修饰，含○的装置不支持 V 及 Z 修饰。例如，操作数 S₁ 支持 D 装置的 V 及 Z 修饰。
6. 指令中若有 V 及 Z 装置，16 位指令只能用 V 装置，而 32 位指令只能用 Z 装置。
7. 表格的右下角，“✓”表有支持 AH05PM/15PM/10PM-5A 及 AH20MC-5A；“-”表不支持。详细的指令变化情况，可以对照着表格下方确认指令是否支持脉冲执行型、16/32 位指令。

应用指令的输入

应用指令中有些指令仅有指令部份 (指令名) 构成，例如：BRET...或 SRET 等等，但是大部份都是指令部份再加上好几个操作数所组合而成。

模块的应用指令是以指令号码 API 00~API 260 来指定的，同时每个指令均有其专用的名称符号，例如：API 12 的指令名称符号为 MOV (数据搬运)。若利用梯形图编辑软件 (PMSoft) 作该指令的输入，只需要直接打入该指令的名称“MOV”即可。而应用指令都会有不同的操作数指定，以 MOV 指令而言：



此指令是将 **S** 指定的操作数之值搬移至 **D** 所指定的目的操作数。其中：

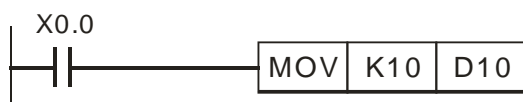
S	来源操作数；若来源操作数有一个以上，那么则以 S₁ 、 S₂ ... 分别表示。
D	目的操作数；若目的操作数有一个以上，那么则以 D₁ 、 D₂ ... 分别表示。
若操作数指定为常数时，则以 m 、 m₁ 、 m₂ 、 n 、 n₁ 、 n₂ 表示。	

◆ 操作数长度（16 位指令或 32 位指令）

操作数的数值内容，其长度可分为 16 位及 32 位，因此部份指令处理不同长度的数据则分为 16 位及 32 位的指令，用以区分 32 位的指令只需要在 16 位指令前加上“D”来表示即可。

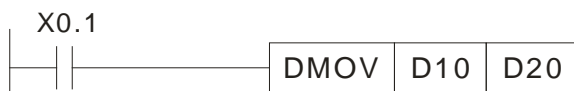
16 位 MOV 指令

当 X0.0=ON 时，K10 被传送至 D10



32 位 DMOV 指令

当 X0.1=ON 时，(D11, D10) 的内容被传送到 (D21, D20)

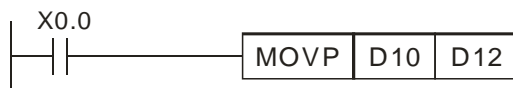


◆ 连续执行型脉冲执行型

以指令的执行方式来说亦可分成「连续执行型」及「脉冲执行型」2 种。由于指令不被执行时，所需的执行时间比较短，因此程序中尽可能的使用脉冲执行型指令，可减少扫描周期。在指令后面加上“P”记号的指令，即为脉冲执行型指令。有些指令大部份的应用上都是使用脉冲执行型方式，如 INC、DEC 及位移相关等指令。

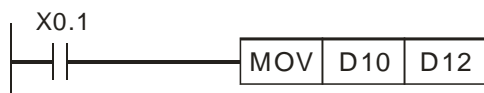
脉冲执行型

当 X0.0 由 OFF→ON 变化时，MOVP 指令被执行一次，该次扫描指令不再被执行，因此称之为脉冲执行型指令



连续执行型

于 X0.1=ON 的每次扫描周期，MOV 指令均被执行一次，因此称之为连续执行型指令



上图的两个条件接点 X0.0、X0.1=OFF 时，指令不被执行，目的地操作数 D 的内容没有变化

◆ 操作数的指定对象

1. M、S 等位装置也可以组合成字符装置使用，在应用指令里以 KnM、KnS 的型态来存放数值数据作运算。
2. 数据寄存器 D、定时器 T、计数器 C、变址寄存器 V、Z、都是一般操作数所指定的对象。
3. 数据寄存器一般为 16 位长度，也就是 1 个 D 寄存器，若指定 32 位长度的数据寄存器时，是指定连续号码的 2 个 D 寄存器。
4. 若 32 位指令的操作数指定 D0，则 (D1、D0) 所组成的 32 位数据寄存器被占用，D1 为上 16 位，而 D0 为下 16 位。定时器 T、16 位计数器 C0~C199 被使用的规则亦相同。
5. 32 位计数器 C240~C255 若是当数据寄存器来使用时，可指定为 32 位指令的操作数。

◆ SM/SR 装置与 M/D 相同

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

由于 SM 与 SR 装置义同 M 及 D 装置，因此可一并参考 M 及 D 字段。

◆ X/Y 装置单位

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

Xn.n (X0.0~X15.15) 与 Yn.n (Y0.0~Y15.15) 装置为 bit。

当指令有支持 D 装置时，亦即支持 X、Y 字符装置。X、Y 字符装置的操作方法等同 D 装置。操作数数据格式

1. 装置 X、Y、M 及 S 只能做为单点的 ON/OFF，我们将之定义为位装置 (Bit device)。
2. 16 位 (或 32 位) 装置 T、C、D 及 V、Z 等寄存器，我们将之定义为字符装置 (Word device)。
3. 利用 Kn (其中 n=1 表示 4 个位，所以 16 位可由 K1~K4，32 位可由 K1~K8) 加在位装置 M 及 S 前，可将其定义为字符装置，因此可作字符装置的运算，例如 K2M0 即表示 8 位 M0~M7。

5



◆ 位装置组合成字符装置的数值数据处理

16 位指令		32 位指令	
16 位所指定的数值为： K-32,768~K32,767		32 位所指定的数值为： K-2,147,483,648~K2,147,483,647	
指定位数 (K1~K4) 的数值为：		指定位数 (K1~K8) 的数值为：	
K1 (4 个位)	0~15	K1 (4 个位)	0~15
K2 (8 个位)	0~255	K2 (8 个位)	0~255
K3 (12 个位)	0~4,095	K3 (12 个位)	0~4,095
K4 (16 个位)	-32,768~+32,767	K4 (16 个位)	0~65,535
		K5 (20 个位)	0~1,048,575
		K6 (24 个位)	0~167,772,165
		K7 (28 个位)	0~268,435,455
		K8 (32 个位)	-2,147,483,648~+2,147,483,647

◆ 一般的标志信号

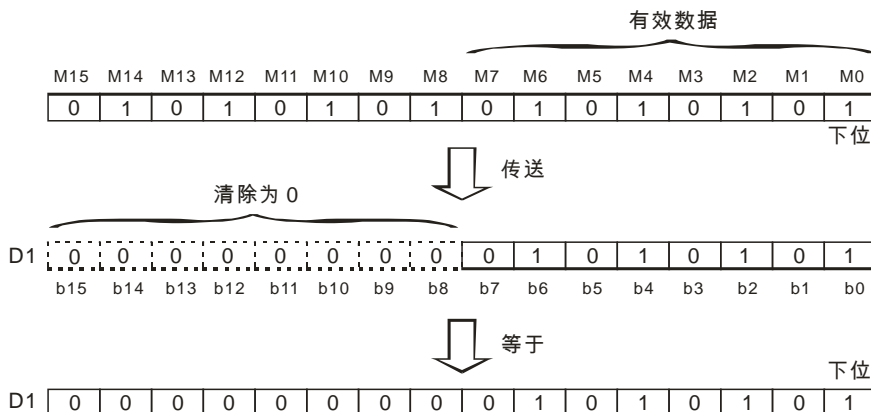
对应着应用指令运算结果，模块有下列的标志信号 (Flag)。

例：SM968：零标志 SM969：借位标志 SM970：进位标志

无论那一个标志信号都会在指令被执行时，随着指令的运算结果作 ON 或 OFF 的变化，例如：于 O100~M102 的主控制程序区段中，如果使用 ADD/SUB/MUL/DIV 等数值运算指令，执行结果会影响 SM968~SM970 等标志信号的状态。但是当指令不被执行时，标志信号的 ON/OFF 状态被保持住。请注意上述标志信号的动作，会与许多指令有关，请参阅个别指令说明。

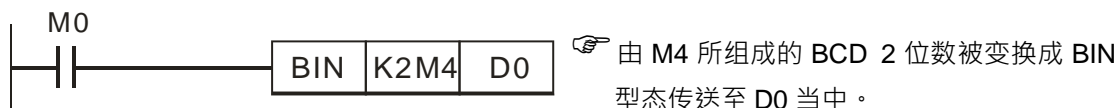
5.3 应用指令对数值的处理方式

- ◆ X、Y、M、S 等只有 ON/OFF 变化的装置称之为位装置 (Bit Device)，而 T、C、D、V、Z 等专门用来存放数值的装置称之为字符装置 (Word Device)。虽然说位装置只能作 ON/OFF 变化，但是 M、S 装置加上特定的宣告位装置也可以数值的型态被使用于应用指令的操作数当中，所谓的宣告是在位装置的前面加上位数，它是以 Kn 来表现。
- ◆ 16 位的数值可使用 K1~K4 而 32 位的数值则可使用 K1~K8。例如：K2M0 是由 M0~M7 所组成的 8 位数值。



5

- ◆ 将 K1M0、K2M0、K3M0 传送至 16 位的寄存器当中，不足的上位数据补 0。将 K1M0、K2M0、K3M0、K4M0、K5M0、K6M0、K7M0 传送至 32 位的寄存器也一样，不足的上位数据补 0。
- ◆ 16 位 (或 32 位) 的运算动作中，操作数的内容若是指定 K1~K3 (或 K4~K7) 的位装置时，不足的上位数据被视为 0，因此一般都是被认定为正数的运算。



- ◆ 位装置的编号可自由指定，M 及 S 的起始编号尽可能的指定为 16 的倍数。
- ◆ 连续号码的指定

以数据寄存器 D 为例，D 的连续号码为 D0、D1、D2、D3、D4...

对于指定位数的位装置而言，连续号码以下所示。

K1M0	K1M4	K1M8	K1M12.....
K2M0	K2M8	K2M16	K2M24.....
K3M0	K3M12	K3M24	K3M36.....
K4M0	K4M16	K4M32	K4M48.....

因此位装置号码如上，请勿跳号以免造成混乱。此外，如果将 K4M0 使用于 32 位的运算当中，上位 16 位被视为 0。32 位的数据请使用 K8M0。

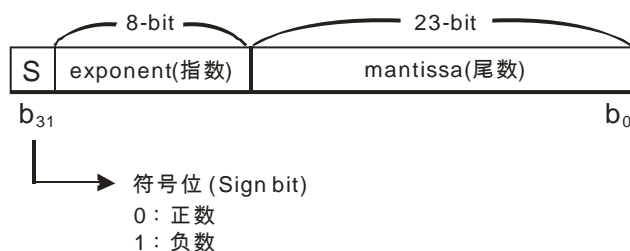
运动控制模块的内部数值运算一般是以 BIN 整数值为准。整数执行除算时，例： $40 \div 3 = 13$ ，余数为 1。整数执行开平方动作时，小数点会被舍弃掉。但若使用小数点运算指令则可求出小数点。

与小数点有关的应用指令如下所示。

API 110 (D ECMP)	API 111 (D EZCP)	API 116 (D RAD)	API 117 (D DEG)
API 120 (D EADD)	API 121 (D ESUB)	API 122 (D EMUL)	API 123 (D EDIV)
API 124 (D EXP)	API 125 (D LN)	API 126 (D LOG)	API 127 (D ESQR)
API 128 (D POW)	API 129 (D INT)	API 130 (D SIN)	API 131 (D COS)
API 132 (D TAN)	API 133 (D ASIN)	API 134 (D ACOS)	API 135 (D ATAN)
API 136 (D SINH)	API 137 (D COSH)	API 138 (D TANH)	

二进浮点数表示法

运动控制模块以 32 位的长度表示浮点数，而表示法系采用 IEEE754 的标准，格式如下：



可表达的大小为：

$$(-1)^S \times 2^{E-B} \times 1.M \quad \text{其中 } B = 127$$

因此 32 位的浮点数的数目范围为 $\pm 2^{-126}$ 到 $\pm 2^{+128}$ 相当于 $\pm 1.1755 \times 10^{-38}$ 到 $\pm 3.4028 \times 10^{+38}$ 。

范例一：以 32 位的浮点数表示 23

步骤一：将 23 转换成二进制： $23.0 = 10111$

步骤二：将二进制正规化： $10111 = 1.0111 \times 2^4$ ，其中 0111 为尾数，4 为指数。

步骤三：求出指数部份的储存值

$$\because E-B=4 \rightarrow E-127=4 \therefore E=131=10000011_2$$

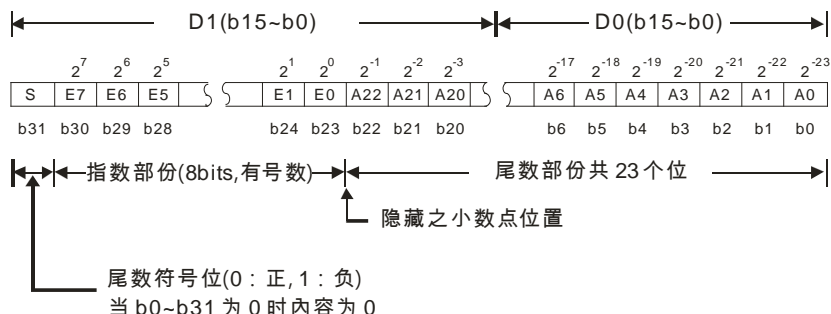
步骤四：组合符号位、指数、尾数成为浮点数。

$$0 \ 10000011 \ 0111000000000000000000_2 = 41B80000_{16}$$

范例二：以 32 位的浮点数表示-23.0

-23.0 浮点格式与 23.0 的转换步骤完全相同，只需将符号位改为 1 即可。

运动控制模块使用 2 个连续号码的寄存器组成 32 位的浮点数，我们以寄存器 (D1、D0) 来存放一个二进浮点数为例，如下所示：



十进浮点数

- ◆ 二进浮点数的内容比较无法被人所接受，因此二进浮点数可转换成十进浮点数来供人作判断。但是运动控制模块对小数点的运算仍旧是使用二进浮点数。
- ◆ 十进浮点数是使用 2 个连续号码的寄存器来表现，较小编号的寄存器号码存放常数部份、较大编号的寄存器号码存放指数部份。

就以寄存器 (D1、D0) 来存放一个十进浮点数为例，如下所示。

$$\text{十进浮点数} = [\text{常数 } D0] \times 10^{[\text{指数 } D1]}$$

底数 $D0 = \pm 1,000 \sim \pm 9,999$

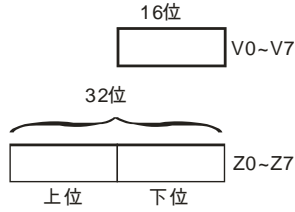
指数 $D1 = -41 \sim +35$

此外，底数 100 不存在于 D0 的内容，因为，100 是以 $1,000 \times 10^{-1}$ 来表现。十进浮点数的范围为 $\pm 1,175 \times 10^{-41}$ 到 $\pm 3,402 \times 10^{+35}$ 。

- ◆ 当 O100 ~ M102 的主控制程序区段中，如果使用 ADD/SUB/MUL/DIV 等数值运算指令，执行结果会影响 SM968~SM970 等标志信号的状态。使用浮点运算指令，执行结果也会影响零标志信号 (SM968)、借位标志信号 (SM969) 及进位标志信号 (SM970)。说明如下：
 - 零标志信号：运算结果为 0 时，SM968=ON。
 - 借位标志信号：运算结果绝对值超出最小处理单位时，SM969=ON。
 - 进位标志信号：运算结果绝对值超出使用范围时，SM970=ON。

5.4 使用变址寄存器V、Z来修饰操作数

变址寄存器 V、Z 为 16 位及 32 位寄存器，V0~V5，Z0~Z7 共计 14 点。



V 是 16 位的数据寄存器，它们可以自由的被写入及读出。但要使用 32 位长度时，必须指定 Z。

可修饰之装置为：X、Y、M、S、KnM、KnS、T、C、D、SM、SR。

可使用 V、Z 作修饰的各部装置如上所示。但是 V、Z 不可修饰本身、不可修饰常数 (K4@Z0 无效)，也不可以修饰 Kn。(K4M0@Z0 有效、K0@Z0M0 无效)。于个别应用指令说明中，凡是于操作数表格中加入 ● 之操作数都可使用 V、Z 作修饰。

5.5 指令索引

- 指令以字母排列

分类	API	指令码		P 指令	功能	STEPS		页码
		16 位	32 位			16 位	32 位	
A	87	ABS	DABS	✓	绝对值	3	3	5-78
	20	ADD	DADD	✓	BIN 加法	7	9	5-39
	134	-	DACOS	✓	二进浮点数→ACOS 运算	-	6	5-101
	172	-	DADDR	✓	浮点数值加法	-	9	5-106
	66	ALT	-	✓	ON/OFF 交替	3	-	5-75
	218	AND&	DAND&	-	S1&S2	5	7	5-111
	220	AND^	DAND^	-	S1^S2	5	7	5-111
	219	AND	DAND	-	S1 S2	5	7	5-111
	234	AND<	DAND<	-	S1 < S2	5	7	5-114
	237	AND<=	DAND<=	-	S1≤S2	5	7	5-114
	236	AND<>	DAND<>	-	S1≠S2	5	7	5-114
	232	AND=	DAND=	-	S1 = S2	5	7	5-114
	233	AND>	DAND>	-	S1 > S2	5	7	5-114
	238	AND>=	DAND>=	-	S1≥S2	5	7	5-114
	47	ANR	-	✓	警报点复位	1	-	5-70
	46	ANS	-	-	警报点输出	7	-	5-69
133	-	DASIN	✓	二进浮点数→ASIN 运算	-	6	5-100	
135	-	DATAN	✓	二进浮点数→ATAN 运算	-	6	5-102	
B	18	BCD	DBCD	✓	BIN→BCD 变换	5	5	5-37
	19	BIN	DBIN	✓	BCD→BIN 变换	5	5	5-38
	15	BMOV	-	✓	全部传送	7	-	5-33
	44	BON	DBON	✓	ON 位判定	7	8	5-67
	258	BRET	-	-	回母线	1	-	5-126
C	01	CALL	-	✓	调用子程序	3	-	5-23
	131	-	DCOS	✓	二进浮点数→COS 运算	-	6	5-96
	137	-	DCOSH	✓	二进浮点数→COSH 运算	-	6	5-104
	00	CJ	-	✓	条件跳转	3	-	5-20
	256	CJN	-	✓	反条件跳转	3	-	5-124
	14	CML	DCML	✓	反转传送	5	6	5-32
	10	CMP	DCMP	✓	比较设定输出	7	9	5-29

分类	API	指令码		P 指令	功能	STEPS		页码
		16 位	32 位			16 位	32 位	
D	25	DEC	DDEC	✓	BIN 减一	3	3	5-45
	41	DECO	-	✓	译码器	7	-	5-62
	117	-	DDEG	✓	弧度→角度	-	6	5-83
	23	DIV	DDIV	✓	BIN 除法	7	9	5-43
	175	-	DDIVR	✓	浮点数值除法	-	9	5-109
	120	-	DEADD	✓	二进浮点数加法	-	9	5-84
	110	-	DECMP	✓	二进浮点数比较	-	9	5-79
	123	-	DEDIV	✓	二进浮点数除法	-	9	5-87
	122	-	DEMUL	✓	二进浮点数乘法	-	9	5-86
E	42	ENCO	-	✓	编码器	7	-	5-64
	127	-	DESQR	✓	二进浮点数开平方根	-	6	5-91
	121	-	DESUB	✓	二进浮点数减法	-	9	5-85
	124	-	DEXP	✓	二进浮点数取指数	-	6	5-88
	111	-	DEZCP	✓	二进浮点数区域比较	-	12	5-80
F	49	-	DFLT	✓	BIN 整数→二进浮点数变换	-	6	5-72
	16	FMOV	DFMOV	✓	多点移动	7	8	5-35
	78	FROM	DFROM	✓	I/O 模块 CR 数据读出	9	12	5-76
I	24	INC	DINC	✓	BIN 加一	3	3	5-44
	129	-	DINT	✓	二进浮点数→BIN 整数变换	-	5	5-93
J	257	JMP	-	-	无条件跳转	3	-	5-125
L	215	LD&	DLD&	-	S1&S2	5	7	5-110
	217	LD^	DLD^	-	S1^S2	5	7	5-110
	216	LD	DLD	-	S1 S2	5	7	5-110
	226	LD<	DLD<	-	S1 < S2	5	7	5-113
	229	LD<=	DLD<=	-	S1 ≤ S2	5	7	5-113
	228	LD<>	DLD<>	-	S1 ≠ S2	5	7	5-113
	224	LD=	DLD=	-	S1 = S2	5	7	5-113
	225	LD>	DLD>	-	S1 > S2	5	7	5-113
	230	LD>=	DLD>=	-	S1 ≥ S2	5	7	5-113
	125	-	DLN	✓	二进浮点数取自然对数	-	6	5-89
	126	-	DLOG	✓	二进浮点数取对数	-	9	5-90
M	45	MEAN	DMEAN	✓	平均值	7	8	5-68

分类	API	指令码		P 指令	功能	STEPS		页码
		16 位	32 位			16 位	32 位	
	259	MMOV	-	✓	16→32 位数值转换	6	-	5-127
	12	MOV	DMOV	✓	数据移动	5	6	5-31
	112	-	DMOV R	✓	浮点数值数据移动	-	6	5-81
	22	MUL	DMUL	✓	BIN 乘法	7	9	5-42
	174	-	DMUL R	✓	浮点数值乘法	-	9	5-108
N	29	NEG	DNEG	✓	取负数 (取 2 的补码)	3	3	5-49
O	221	OR&	DOR&	-	S1&S2	5	7	5-112
	223	OR^	DOR^	-	S1^S2	5	7	5-112
	222	OR	DOR	-	S1 S2	5	7	5-112
	242	OR<	DOR<	-	S1 < S2	5	7	5-115
	245	OR<=	DOR<=	-	S1 ≤ S2	5	7	5-115
	244	OR<>	DOR<>	-	S1 ≠ S2	5	7	5-115
	240	OR=	DOR=	-	S1 = S2	5	7	5-115
	241	OR>	DOR>	-	S1 > S2	5	7	5-115
	246	OR>=	DOR>=	-	S1 ≥ S2	5	7	5-115
P	128	-	DPOW	✓	浮点数权值指令	-	9	5-92
R	116	-	DRAD	✓	角度→弧度	-	6	5-82
	154	RAND	DRAND	✓	随机数值产生	7	9	5-117
	33	RCL	DRCL	✓	附进位标志左旋转	5	6	5-54
	32	RCR	DRCR	✓	附进位标志右旋转	5	6	5-53
	50	REF	-	✓	I/O 更新处理	5	-	5-73
	260	RMOV	-	✓	32→16 位数值转换	6	-	5-128
	31	ROL	DROL	✓	左旋转	5	6	5-52
	30	ROR	DROR	✓	右旋转	5	6	5-51
	09	RPE	-	-	循环回路结束	1	-	5-28
	08	RPT	-	-	循环回路起始 (只能一层)	3	-	5-27
S	202	SCAL	-	✓	比例值运算	7	-	5-118
	203	SCLP	DSCLP	✓	参数型比例值运算	7	9	5-120
	61	SER	DSER	✓	多点比较	9	11	5-74
	39	SFRD	-	✓	位移读出	7	-	5-60
	35	SFTL	-	✓	位左移	9	-	5-56

分类	API	指令码		P 指令	功能	STEPS		页码
		16 位	32 位			16 位	32 位	
S	34	SFTR	—	✓	位右移	9	—	5-55
	38	SFWR	—	✓	位移写入	7	—	5-59
	130	—	DSIN	✓	二进浮点数→SIN 运算	—	6	5-94
	136	—	DSINH	✓	二进浮点数→SINH 运算	—	6	5-103
	173	—	DSUBR	✓	浮点数值减法	—	9	5-107
	48	SQR	DSQR	✓	BIN 开平方根	5	6	5-71
	2	SRET	—	—	子程序结束	1	—	5-24
	21	SUB	DSUB	✓	BIN 减法	7	9	5-41
	43	SUM	DSUM	✓	ON 位数量	5	5	5-66
	152	SWAP	DSWAP	✓	上/下字节交换	3	3	5-116
T	132	—	DTAN	✓	二进浮点数→TAN 运算	—	6	5-98
	138	—	DTANH	✓	二进浮点数→TANH 运算	—	6	5-105
	79	TO	DTO	✓	I/O 模块 CR 数据写入	9	13	5-77
W	26	WAND	DWAND	✓	逻辑及 (AND) 运算	7	9	5-46
	07	WDT	—	✓	逾时监视定时器	1	—	5-26
	27	WOR	DWOR	✓	逻辑或 (OR) 运算	7	9	5-47
	37	WSFL	—	✓	寄存器左移	9	—	5-58
	36	WSFR	—	✓	寄存器右移	9	—	5-57
	28	WXOR	DWXOR	✓	逻辑异或 (XOR) 运算	7	9	5-48
X	17	XCH	DXCH	✓	数据的交换	5	9	5-36
Z	11	ZCP	DZCP	✓	区域比较	9	12	5-30
	40	ZRST	—	✓	区域清除	5	—	5-61

5.6 应用指令说明

API	指令码		操作数				功能			
00	CJ	P	S				条件跳转			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S															

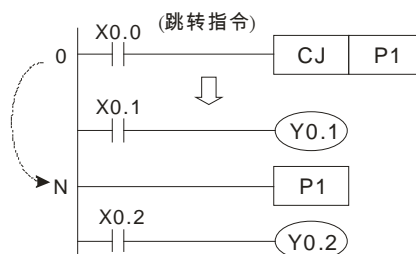
脉冲执行型	16 位指令 (3 steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：条件跳转之目的指针。
- ◆ P 指针范围是 P0~P255。
- ◆ 当使用者希望 O100 主程序中的某一部份不需要执行时，以缩短扫描时间，以及使用于双重输出时，可使用 CJ 或 CJP 指令。
- ◆ 指针 P 所指之程序若在 CJ 指令之前，需注意会发生 WDT 逾时之错误，O100 主程序停止运转，请注意使用。
- ◆ CJ 指令可重复指定同一指针 P，但 CJ 与 CALL 不可指定同一指针 P，否则会产生错误。
- ◆ 跳转执行中各种装置动作情形说明：
 - ◆ Y、M、S 保持跳转发生前之状态。
 - ◆ 执行计时中之 10ms 定时器会暂停计时。
 - ◆ 一般计数器会停止计数，以及一般应用指令不会被执行。
 - ◆ 定时器之清除指令若在跳转前被驱动，则在跳转执行中该装置仍处于清除状态。

程序范例
(一)

- ◆ 当 X0.0=ON 时，程序自动从地址 0 跳转至地址 N (即指定之标签 P1) 继续执行，中间地址跳过不执行。
- ◆ 当 X0.0=OFF 时，程序如同一般程序由地址 0 继续往下执行，此时 CJ 指令不被执行。



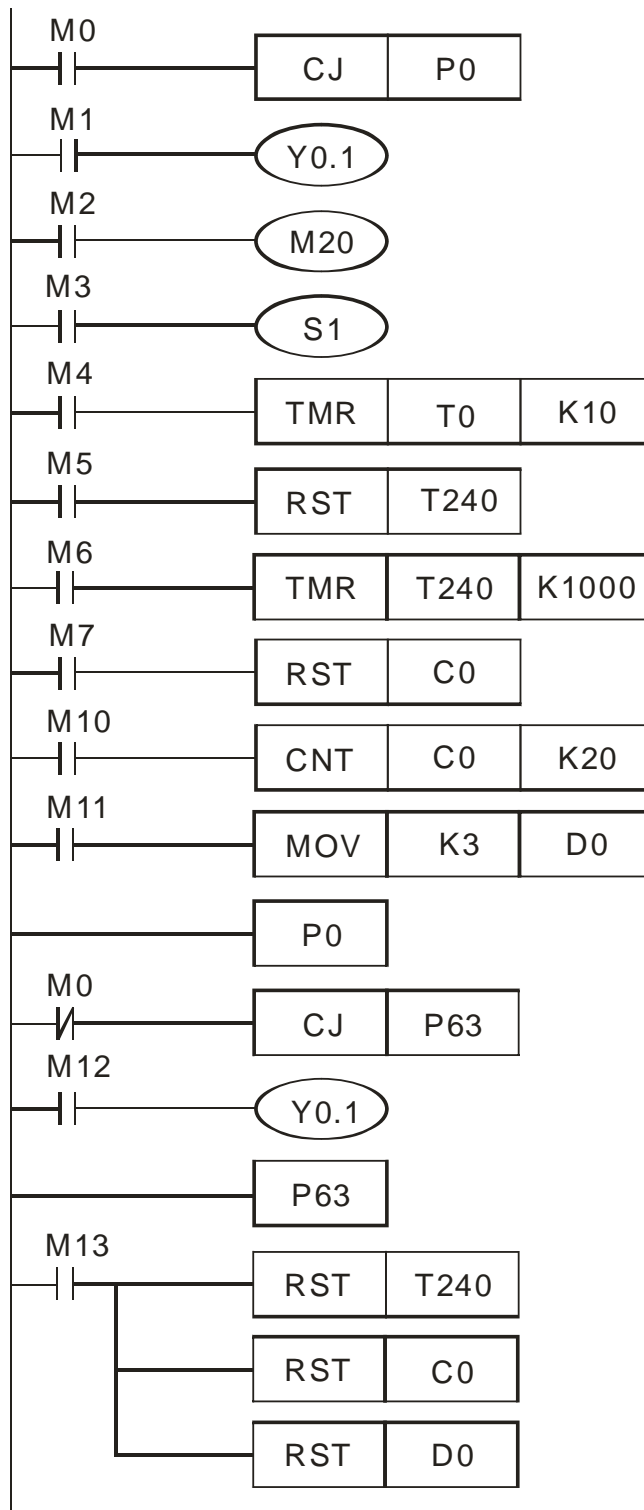
程序范例
(二)

◆ 底下表格为下列程序中，各个装置状态变化。

装置	CJ 执行前 接点状态	CJ 执行中 接点状态	CJ 执行中 输出线圈状态
Y、M、S	M1、M2、M3 OFF	M1、M2、M3 由 OFF→ON	Y0.1 ^{注一} 、M20、 S1 OFF
	M1、M2、M3 ON	M1、M2、M3 由 ON→OFF	Y0.1 ^{注一} 、M20、 S1 ON
10ms 定时器	M4 OFF	M4 由 OFF→ON	定时器 T0 不作计 时动作
	M4 ON	M4 由 ON→OFF	定时器 T0 立即停 止计时并保持·M0 由 ON→OFF·T0 被清为 0
	M6 OFF	M6 由 OFF→ON	定时器 T240 不作 计时动作
	M6 ON	M6 由 ON→OFF	定时器 T240 立即 停止计时并保持· M0 由 ON→OFF· T240 被清为 0
C0~C234	M7、M10 OFF	M10 ON/OFF 触 发	计数器 C0 不计数
	M7 OFF、 M10 ON/OFF 触 发	M10 ON/OFF 触 发	计数器 C0 停止计 数并保持·M0 OFF 后·C0 继续 计数
应用指令	M11 OFF	M11 由 OFF→ON	应用指令不执行
	M11 ON	M11 由 ON→OFF	被跳过之应用指令 不执行，但是 API 53~59、API 157~159 继续动作

注一：Y0.1 为双重输出，M0 为 OFF 时，由 M1 控制，M0 为 ON 时，由 M12 控制。

- ◆ Y0.1 为双重输出，M0=OFF 时，由 M1 来控制，M0=ON 时，由 M12 来控制。



5

API	指令码			操作数							功能				
01		CALL	P	S							调用子程序				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S															
		脉冲执行型			16 位指令 (3 steps)			32 位指令							
		✓			✓			—							

指令说明

- ◆ **S**：调用子程序之指针。
- ◆ **P** 指针范围是 P0~P255。
- ◆ 指针所指定的子程序请于 M102、M2 和 SRET 指令后编写。
- ◆ 指针 **P** 之号码在被 CALL 使用时，不可与 CJ、CJN、JMP 指令指定相同之号码。
- ◆ 若仅使用 CALL 指令则可不限次数调用同一指针号码之子程序。

API	指令码	操作数	功能
02	SRET	—	子程序结束

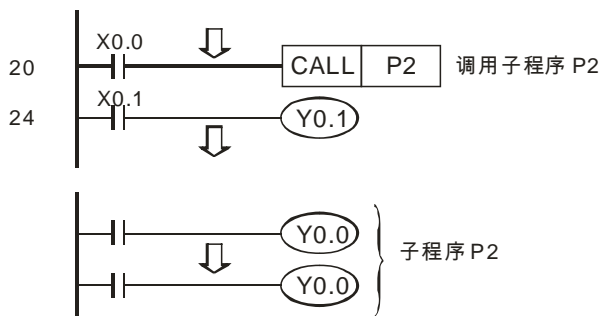
脉冲执行型	16 位指令 (1 step)	32 位指令
—	✓	—

指令说明

- ◆ 表示子程序结束。子程序执行结束由 SRET 返回 O100 主程序，执行原调用该子程序 CALL 指令的下一个指令。

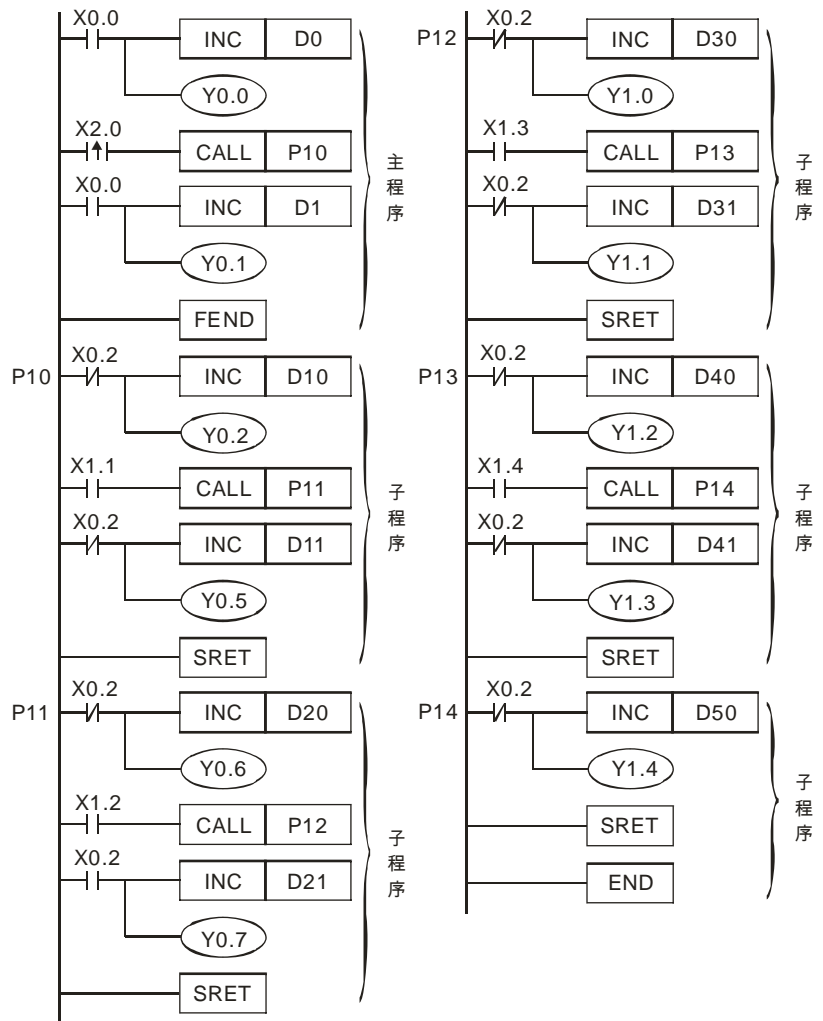
程序范例
(一)

- ◆ 当 X0.0 为 ON 时，则执行 CALL 指令，跳转到 P2 执行所指定的子程序，当执行 SRET 指令时，则回到地址 24，继续往下执行。



程序范例
(二)

- ◆ 当 X2.0 为由 OFF 到 ON 之上升沿触发执行 CALL P10 指令，跳转到 P10 执行所指定的子程序。
- ◆ 当 X1.1 为 ON 时，则执行 CALL P11，跳转到 P11 执行所指定的子程序。
- ◆ 当 X1.2 为 ON 时，则执行 CALL P12，跳转到 P12 执行所指定的子程序。
- ◆ 当 X1.3 为 ON 时，则执行 CALL P13，跳转到 P13 执行所指定的子程序。
- ◆ 当 X1.4 为 ON 时，则执行 CALL P14，跳转到 P14 执行所指定的子程序，当执行到 SRET 指令时，则回到前一个 P×子程序继续往下执行。
- ◆ 在 P10 子程序中执行到 SRET 指令后回到主程序。



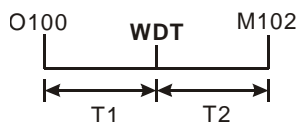
API	指令码		操作数	功能
07	WDT	P	—	逾时监视定时器

脉冲执行型	16 位指令 (1 step)	32 位指令
✓	✓	-

指令说明

- ◆ WDT 指令可用来清除运动控制模块中的监控定时器的计时时间。当扫描 (由地址 0 至 END 或 FEND 指令执行时间) 超过 200ms 时，ERROR 的指示灯会亮，使用者必须将电源 OFF 再 ON，运动控制模块会依据 MANU/AUTO 开关来判断 RUN/STOP 状态，若无 MANU/AUTO 开关，则会自动回到 STOP 状态。
- ◆ 令逾时监视定时器动作的时机：
 - ◆ 系统发生异常。
 - ◆ 程序执行时间太长，造成扫描周期大于 SR0 的内容值。可以下列 2 种方法来改善：

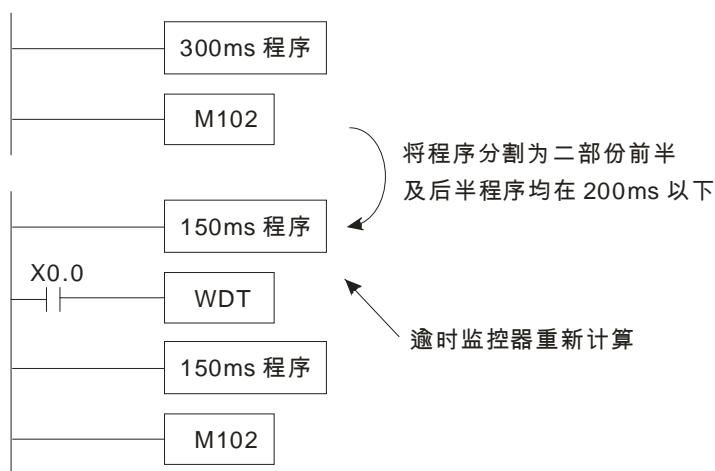
使用 WDT 指令



可由 SR0 (出厂设定值为 200ms) 的设定值改变逾时监视时间。

程序范例

- ◆ 若程序扫描时间为 300ms，此时，可将程序分割为 2 部份，并在中间放入 WDT 指令，使得前半及后半程序都在 200ms 以下。



补充说明

- ◆ WDT 指令用于输入条件成立时的程序扫描。可编写程序强制 WDT 指令只在一个周期内执行。用户可以使用 WDT 指令的脉冲型指令 WDTTP。
- ◆ 逾时监视定时器的预设设定值是 200ms。此时间限制可由用户自行定制，透过编辑数据寄存器 SR0 即可。

API	指令码			操作数								功能	
08		RPT		S								循环回路起始	

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	

脉冲执行型	16 位指令 (3 steps)	32 位指令
-	✓	-

指令说明

- ◆ S：回路重复执行的次数。
- ◆ 循环式 RPT~RPE 回路只能使用一层，当回路层数超过一层，就会显示错误产生。

PI	指令码		操作数	功能
09		RPE	-	循环回路结束

脉冲执行型	16 位指令 (1step)	32 位指令
-	✓	-

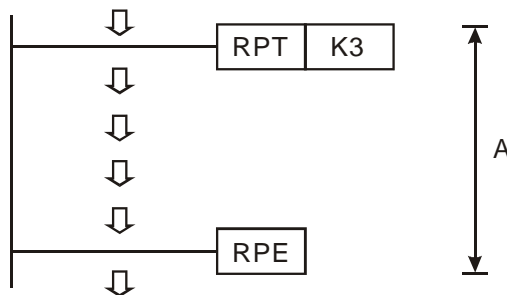
指令说明

- ◆ 由 RPT 指令指定 RPT~RPE 循环来回执行 N 次后跳出 RPT~RPE 循环往下继续执行。
- ◆ 指定次数范围 $N=K1\sim K32,767$ ，当指定次数范围 $N\leq K1$ 时，都视为是 K1。
- ◆ 当不执行 RPT~RPE 回路时，可使用 CJ 指令来跳出回路。
- ◆ 下列情形会产生错误：
 - ◆ RPE 指令在 RPT 指令之前。
 - ◆ 有 RPT 指令没有 RPE 指令。
 - ◆ RPT~RPE 指令个数不同时。
- ◆ 循环式 RPT~RPE 回路只能使用一层，当回路次数超过一次，就会显示错误产生。

5

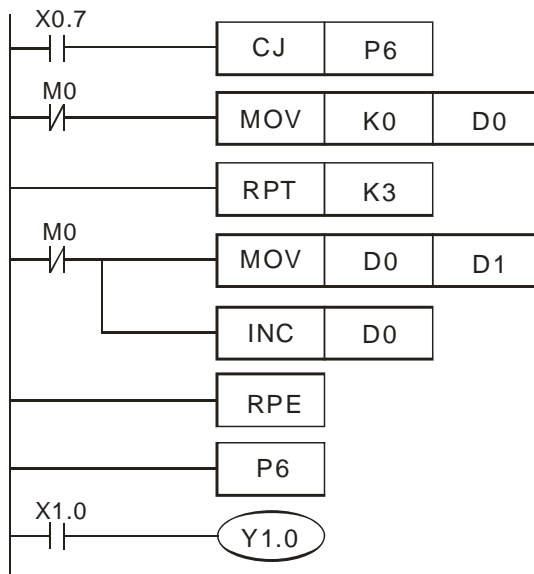
程序范例 (一)

- ◆ 如果想使 A 区段程序执行 3 次，可利用 RPT~RPE 之写法如下所示：



程序范例 (二)

- ◆ 当 X0.7=OFF 时，会执行 RPT~RPE 之间的程序，当 X0.7=ON 时，CJ 指令执行跳转至 P6 处，RPT~RPE 之间的程序跳过不执行。



API	指令码			操作数									功能		
10	D	CMP	P	$S_1 \cdot S_2 \cdot D$									比较设定输出		
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D		○	○	○											

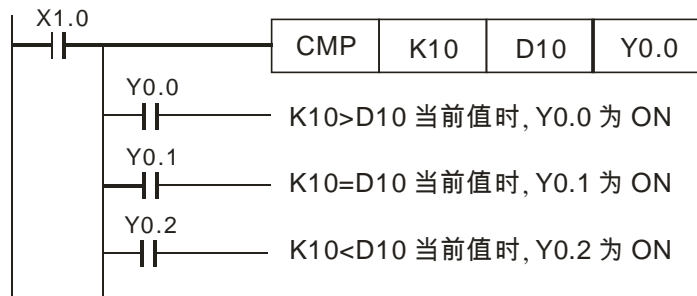
脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 比较值 1。 S_2 : 比较值 2。 D : 比较结果。
- ◆ 比较值 S_1 与 S_2 内容作比较，其比较结果在 D 作表示。
- ◆ D 操作数会占用连续 3 个点。

程序范例

- ◆ 指定装置为 $Y0.0$ ，则自动占有 $Y0.0$ 、 $Y0.1$ 及 $Y0.2$ 。
- ◆ 当 $X1.0=ON$ 时， CMP 指令执行， $Y0.0$ 、 $Y0.1$ 及 $Y0.2$ 其中之一会 ON ，当 $X1.0=OFF$ 时， CMP 指令不执行， $Y0.0$ 、 $Y0.1$ 及 $Y0.2$ 状态保持在 $X1.0=OFF$ 之前的状态。
- ◆ 若需要得到 \geq 之结果时，可将 $Y0.0$ 和 $Y0.1$ 串联即可取得。若需要得到 \leq 之结果时，可将 $Y0.1$ 和 $Y0.2$ 串联即可取得。若需要得到 \neq 之结果时，可将 $Y0.0 \sim Y0.2$ 串联即可取得。



API	指令码			操作数								功能			
11	D	ZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$								区域比较			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
S					○	○		●	●	●	●	●	●	○	○
D		○	○	○											

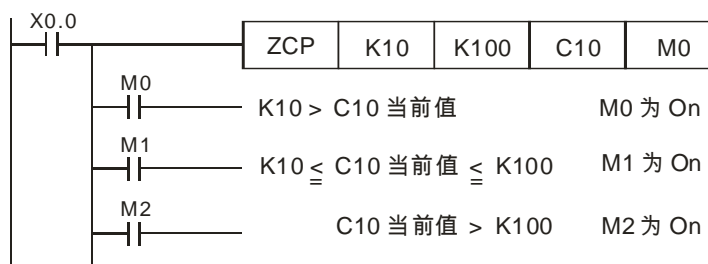
脉冲执行型	16 位指令 (9 steps)	32 位指令 (12 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 区域比较之下限值。 S_2 : 区域比较之上限值。 S : 比较值。 D : 比较结果。
- ◆ 比较值 S 与下限 S_1 及上限 S_2 作比较，其比较结果在 D 作表示。
- ◆ S_2 必须比 S_1 大。
- ◆ D 操作数会占用连续 3 个点。

程序范例

- ◆ 指定装置为 $M0$ ，则自动占有 $M0$ 、 $M1$ 及 $M2$ 。
- ◆ 当 $X0.0=ON$ 时， ZCP 指令执行， $M0$ 、 $M1$ 及 $M2$ 其中之一会 ON ，当 $X0.0=OFF$ 时， ZCP 指令不执行， $M0$ 、 $M1$ 及 $M2$ 状态保持在 $X0.0=OFF$ 之前的状态。



5

API	指令码			操作数									功能		
12	D	MOV	P	S · D									数据移动		
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

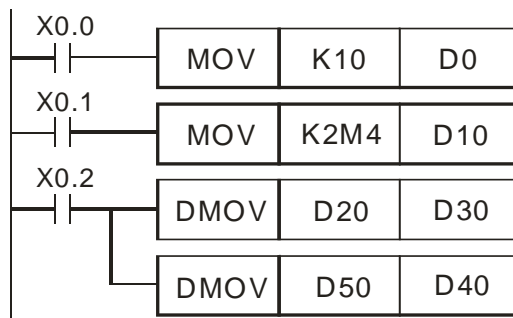
指令说明

- ◆ S：数据之来源。D：数据之搬移目的地。
- ◆ 当该指令执行时，将 S 的内容直接搬移至 D 内。当指令不执行时，D 内容不会变化。
- ◆ 若运算结果为 32 位输出时，(如应用指令 MUL 等) 数据搬动则必须要用 DMOV 指令。

程序范例

- ◆ 16 位数据搬移，须使用 MOV 指令。
 1. 当 X0.0=OFF 时，D0 内容没有变化，若 X0.0=ON 时，将数值 K10 传送至 D0 数据寄存器内。
 2. 当 X0.1=OFF 时，D10 内容没有变化，若 X0.1=ON 时，将 K2M4 当前值传送至 D10 数据寄存器内。
- ◆ 32 位数据搬移，须使用 DMOV 指令。

当 X0.2=OFF 时 (D31、D30) (D41、D40) 内容没有变化，若 X0.2=ON 时，将 (D21、D20) 当前值传送至 (D31、D30) 数据寄存器内。同时，将 (D51、D50) 当前值传送至 (D41、D40) 数据寄存器内。



API	指令码			操作数								功能			
14	D	CML	P	S · D								反转传送			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

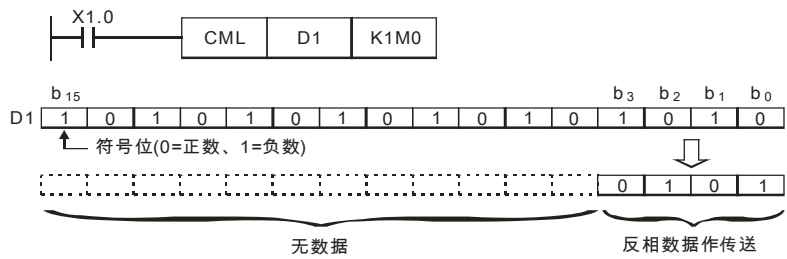
脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

指令说明

- ◆ S：传送数据来源。D：传送的目的地装置。
- ◆ 将 S 的内容全部取反 (0→1 · 1→0) 传送到 D 当中。

程序范例
(一)

- ◆ 当 X1.0=ON 时，将 D1 的 b0~b3 内容反相后传送到 M0~M3。



API	指令码			操作数								功能
15		BMOV	P	S · D · n								全部传送

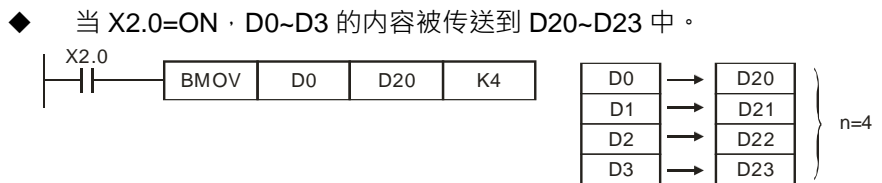
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●		
D								●	●	●	●	●	●		
n					○	○									

脉冲执行型	16 位指令 (7steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：来源装置起始。D：目的地装置起始。n：传送区块长度。
- ◆ 此指令用于传送多笔数据到新的寄存器。S 所指定的装置起始号码开始算 n 个寄存器的内容被传送至 D 所指定的装置起始号码开始算 n 个寄存器当中。如果 n 所指定点数超过该装置的使用范围时，只有有效范围被传送。
- ◆ n 的范围 1~512。

程序范例 (一)



程序范例 (二)



5

程序范例
(三)

◆ 为了防止两个操作数所指定传送的号码重叠时，所造成的混乱，请注意两个操作数所指定号码大小的安排。

1. 当 $S > D$ ，BMOV 指令以 ①→②→③的顺序传送



2. 当 $S < D$ ，the BMOV 指令以 ③→②→①的顺序传送，D11~D13 内容数值都是 D10 的内容。



API	指令码			操作数								功能
16		FMOV	P	S · D · n								多点传送

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●	○	
D								●	●	●	●	●	●		
n					○	○									

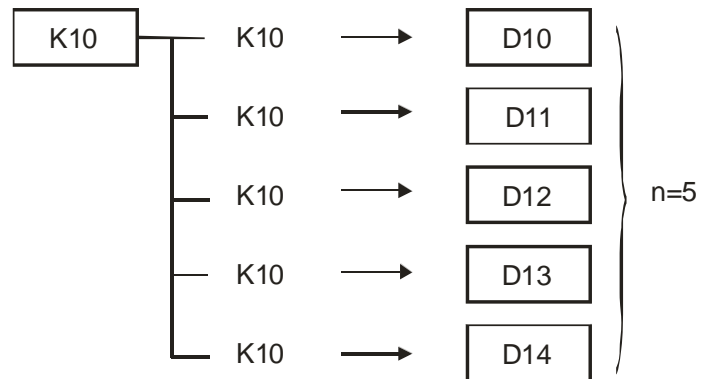
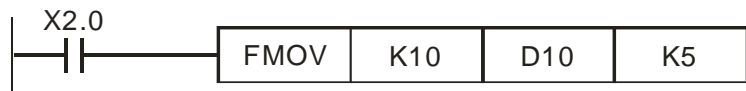
脉冲执行型	16 位指令 (7 steps)	32 位指令 (8 steps)
✓	✓	✓

指令说明

- ◆ **S** : 数据的来源。**D** : 目的地装置起始。**n** : 传送区块长度。
- ◆ **S** 的内容被传送至 **D** 所指定的装置起始号码开始算 **n** 个寄存器当中，如果 **n** 所指定点数超过该装置的使用范围时，只有有效范围被传送
- ◆ **n** 的范围 1~512。

程序范例

- ◆ 当 X2.0=ON 时 K10 被传送到由 D10 开始的连续 5 个寄存器(D10~D14) 中。



API	指令码			操作数								功能			
17	D	XCH	P	D ₁ · D ₂								数据的交换			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D ₁								●	●	●	●	●	●	○	○
D ₂								●	●	●	●	●	●	○	○

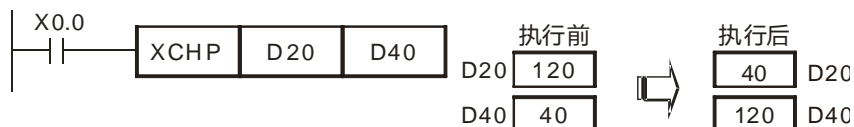
脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
✓	✓	✓

指令说明

- ◆ D₁ : 互相交换的数据 1 · D₂ : 互相交换的数据 2 ·
- ◆ 将 D₁ 与 D₂ 的内容互换 ·
- ◆ 此指令最好以脉冲的方式执行 (XCHP) ·

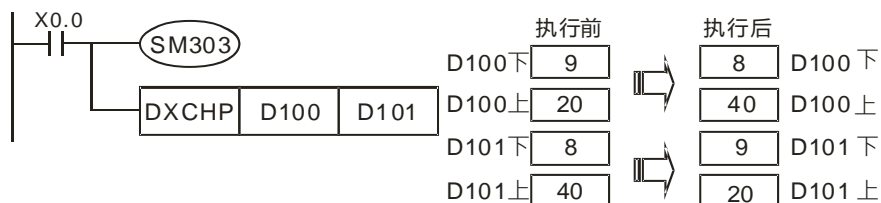
程序范例

- ◆ 当 X0.0=OFF→ON · D20 与 D40 的内容互换 ·



补充说明

- ◆ 16 位指令当 D₁ 及 D₂ 所指定的装置相同时 · 且 SM303=ON · 则该装置的上下 8 位内容互相交换 ·
- ◆ 32 位指令当 D₁ 及 D₂ 所指定的装置相同时 · 且 SM303=ON · 则该 32 位装置个别上下 16 位内容互相交换 ·
- ◆ 当 X0.0=ON 时 · 且 SM303=ON · D100 的 16 位内容与 D101 的 16 位内容互相交换 ·



API	指令码			操作数										功能	
18	D	BCD	P	S · D										BCD 转换	
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

指令说明

- ◆ 2 进制 S 的内容 (BIN 值) 被转换成 BCD 码数据 , 然后传送到 D 中。
- ◆ 如果 BCD 数据转换范围在 0 到 9,999 之外 , 则 BCD 指令不会执行。如果 BCD 转换数据范围是在 0 到 99,999,999 之外 , 则 DBCD 指令不会执行。
- ◆ BCD 指令把定位单元中的二进制数据转成 BCD 码数据 , 再输出到外部装置中 (如 7 段显示器等等)。
- ◆ 运算错误标志 , OX 是 SM1049 , O100 是 SM953 。

程序范例

- ◆ 当 X0.0=ON 时 , D10 之 BIN 值被转换成 BCD 值后 , 将结果的个位数存于 K1M0 (M0~M3) 四个 bit 组件。



若 D10=001E(Hex)=0030(十进制) , 则执行结果 M0~M3=0000(BIN)。

API	指令码			操作数							功能				
19	D	BIN	P	S · D							BIN 转换				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

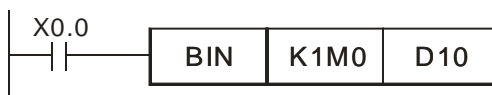
脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
✓	✓	✓

指令说明

- ◆ S：数据之来源。D：变换之结果数据。
- ◆ 2 进制 S 的内容 (BCD 值) 被转换成 BIN 码数据，然后传送到 D 中。
- ◆ 数据来源 S 的内容有效数值范围 BCD(0~9,999)、DBC(0~99,999,999)。
- ◆ 常数 K、16#会自动转换成 BIN，故不需运用此指令。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

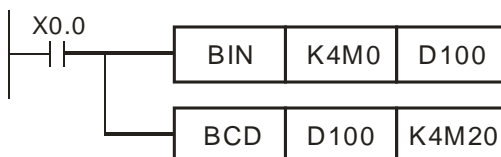
程序范例
(一)

- ◆ 当 X0.0=ON 时，K1M0 之 BCD 值被转换成 BIN 值后，将结果存于 D10 中。



补充说明

- ◆ BCD 与 BIN 指令应用说明：
 1. 当运动控制模块要从外界读取一个 BCD 型态指拨开关时，就必须使用 BIN 指令先将读取到的数据转换成 BIN 值再储存在运动控制模块内。
 2. 当运动控制模块要将内部储存的数据经由外界一个 BCD 型态的 7 段显示器显示出来时，就必须使用 BCD 指令先将要显示的内部数据转换成 BCD 值再送到 7 段显示器。
 3. 当 X0.0=ON 时，将 K4M0 BCD 值转换成 BIN 值传送到 D100，再将 D100 之 BIN 值转换成 BCD 值传送到 K4M20。



API	指令码			操作数									功能				
	D	ADD	P	S ₁ · S ₂ · D									BIN 加法				
20																	
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z		
S ₁					○	○		●	●	●	●	●	●	○	○		
S ₂					○	○		●	●	●	●	●	●	○	○		
D								●	●	●	●	●	●	○	○		
							脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)								
							✓	✓	✓								

指令说明

- ◆ S₁：被加数。S₂：加数。D：和。
- ◆ 将两个数据源：S₁ 及 S₂ 以 BIN 方式相加的结果存于 D。
- ◆ 各数据的最高位为符号位 0 表 (正) 1 表 (负)，因此可做代数加法运算。(例如：3+ (-9) =-6)。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。
- ◆ 加法相关标志变化。

16 位 BIN 加法：

1. 运算结果为 0 时，零标志 (Zero flag) 为 ON。
2. 运算结果小于-32,768 时，借位标志 (Borrow flag) 为 ON。
3. 运算结果大于 32,767 时，进位标志 (Carry flag) 为 ON。

32 位 BIN 加法：

1. 运算结果为 0 时，零标志 (Zero flag) 为 ON。
2. 运算结果小于-2,147,483,648 时，借位标志 (Borrow flag) 为 ON。
3. 运算结果大于 2,147,483,647 时，进位标志 (Carry flag) 为 ON。

程序范例 (一)

- ◆ 16 位 BIN 加法：当 X0.0=ON 时，被加数 D0 内容加上加数 D10 之内容将结果存在 D20 之内容当中。



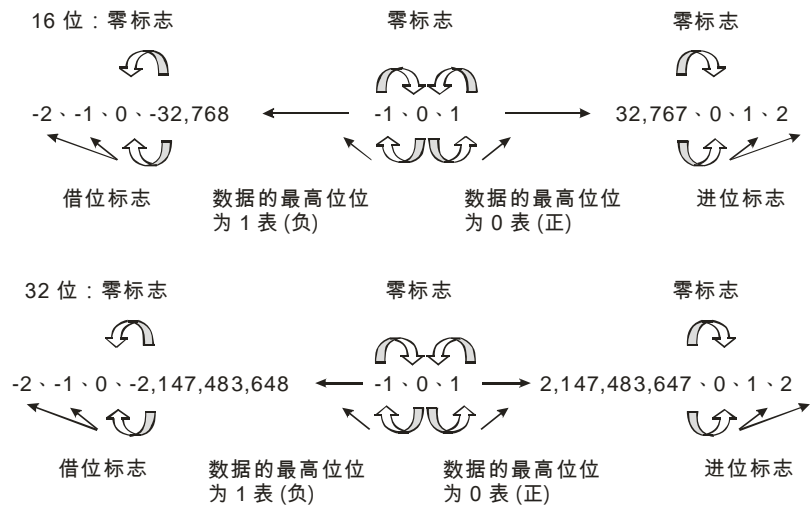
程序范例 (二)

- ◆ 32 位 BIN 加法：当 X0.1=ON 时，被加数(D31 · D30)内容加上加数(D41 · D40) 之内容将结果存在 (D51 · D50) 之中。(其中 D30 · D40 · D50 为低 16 位数据，D31 · D41 · D51 为高 16 位数据)



补充说明

◆ 标志动作与数值的正负关系：



API	指令码			操作数									功能		
21	D	SUB	P	$S_1 \cdot S_2 \cdot D$									BIN 减法		
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

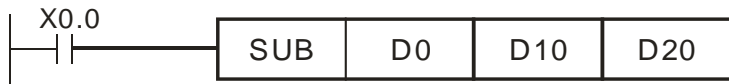
脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 被减数。 S_2 : 减数。 D : 差。
 - ◆ 将两个数据源： S_1 及 S_2 以 BIN 方式相减的结果存于 D 。
 - ◆ 各数据的最高位为符号位 0 表 (正) 1 表 (负)，因此可做代数减法运算。
 - ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
 - ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
 - ◆ 进位标志，OX 是 SM1066，O100 是 SM970。
 - ◆ 减法相关标志变化。
- 16 位 BIN 减法：
1. 运算结果为 0 时，零标志 (Zero flag) 为 ON。
 2. 运算结果小于 -32,768 时，借位标志 (Borrow flag) 为 ON。
 3. 运算结果大于 32,767 时，进位标志 (Carry flag) 为 ON。
- 32 位 BIN 减法：
1. 运算结果为 0 时，零标志 (Zero flag) 为 ON。
 2. 运算结果小于 -2,147,483,648 时，借位标志 (Borrow flag) 为 ON。
 3. 运算结果大于 2,147,483,647 时，进位标志 (Carry flag) 为 ON。
- ◆ 标志动作与数值的正负关系参考标志动作与数值的正负关系请参考上页指令 ADD 之补充说明。

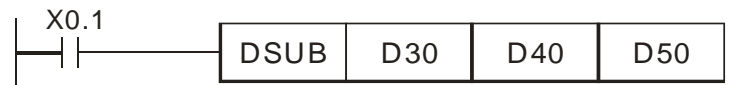
程序范例 (一)

- ◆ 16 位 BIN 减法 : 当 X0.0=ON 时，将 D0 内容减掉 D10 内容将差存在 D20 之内容中。



程序范例 (二)

- ◆ 32 位 BIN 减法 : 当 X0.1=ON 时，(D31、D30) 内容减掉 (D41、D40) 之内容后，将差存在 (D51、D50) 之中。(其中 D30、D40、D50 为低 16 位数据，D31、D41、D51 为高 16 位数据)



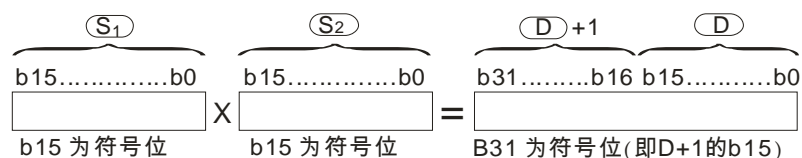
API	指令码			操作数								功能
22	D	MUL	P	$S_1 \cdot S_2 \cdot D$								BIN 乘法

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D										●	●	●	●		

脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

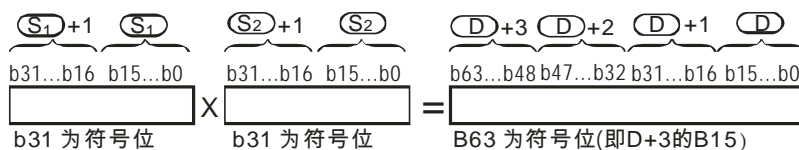
指令说明

- ◆ S_1 : 被乘数。 S_2 : 乘数。 D : 积。
- ◆ 将两个数据源： S_1 及 S_2 以有符号二进制方式相乘后的积存于 D 。 必须注意 16 位及 32 位运算时， S_1 、 S_2 及 D 的正负号位。
- ◆ 16 位 BIN 乘法运算：



符号位=0 为正数，符号位=1 为负数。

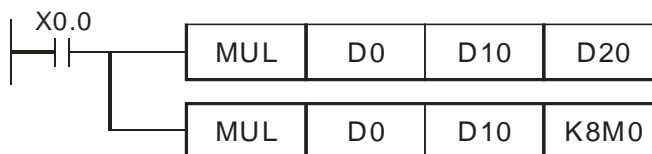
- ◆ 32 位 BIN 乘法运算：



符号位=0 为正数，符号位=1 为负数。

程序范例

- ◆ 16 位 D0 乘上 16 位 D10 其结果是 32 位之积，上 16 位存于 D21，下 16 位存于 D20 内，结果之正负由最左边位之 OFF/ON 来代表正或负值。



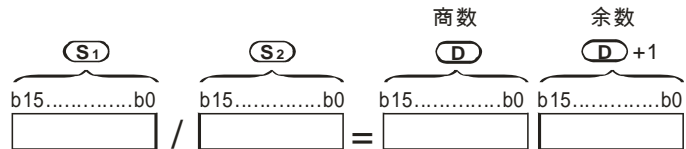
API	指令码			操作数										功能
23	D	DIV	P	$S_1 \cdot S_2 \cdot D$										BIN 除法

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D										●	●	●	●		

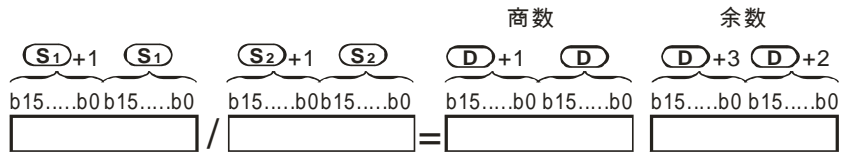
脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 被除数。 S_2 : 除数。 D : 商及余数。
- ◆ 将两个数据源： S_1 及 S_2 以有号数二进制方式相除后的商及余数存于 D 。必须注意 16 位及 32 位运算时， S_1 、 S_2 及 D 的正负号位。
- ◆ 除数为 0 时，指令不执行。
- ◆ 16 位 BIN 除法运算：

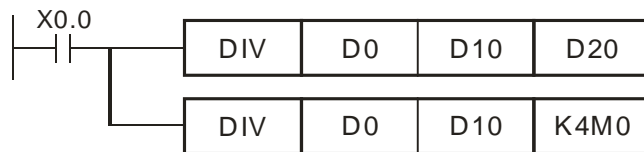


- ◆ 32 位 BIN 除法运算：



程序范例

- ◆ 当 X0.0=ON 时，被除数 D0 除以除数 D10 而结果商被指定放于 D20，余数指定放于 D21 内。所得结果之正负由最高位位之 OFF/ON 来代表正或负值。



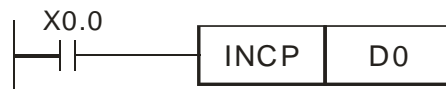
API	指令码			操作数								功能			
24	D	INC	P	D								BIN 加一			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
		脉冲执行型		16 位指令 (3 steps)				32 位指令 (3 steps)							
		✓		✓				✓							

指令说明

- ◆ D：目的地装置。
- ◆ 若指令不是脉冲执行型，则当指令执行时，程序每次扫描周期被指定的装置 D 内容都会加 1。
- ◆ 本指令一般都是使用脉冲执行型指令 (INCP、DINCP)。
- ◆ 16 位运算时 -32,767 再加 1 则变为 -32,768；32 位运算时 -2,147,483,647 再加 1 则变为 -2,147,483,648。

程序范例

- ◆ 当 X0.0=OFF→ON 时，D0 内容自动加 1。



API	指令码			操作数										功能	
25	D	DEC	P	D										BIN 减一	
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
								脉冲执行型	16 位指令 (3 steps)	32 位指令 (3 steps)					
								✓	✓	✓					

指令说明

- ◆ D：目的地装置。
- ◆ 若指令不是脉冲执行型，当指令执行时，程序每次扫描周期被指定的装置 D 内容都会减 1。
- ◆ 本指令一般都是使用脉冲执行型指令 (DECP、DDECP)。
- ◆ 16 位运算时 -32,768 再减 1 则变为 32,767 32 位运算时 -2,147,483,648 再减 1 则变为 2,147,483,647。

程序范例

- ◆ 当 X0.0=OFF→ON 时，D0 内容自动减 1。



API	指令码			操作数								功能
26	D	WAND	P	$S_1 \cdot S_2 \cdot D$								逻辑及 (AND) 运算

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

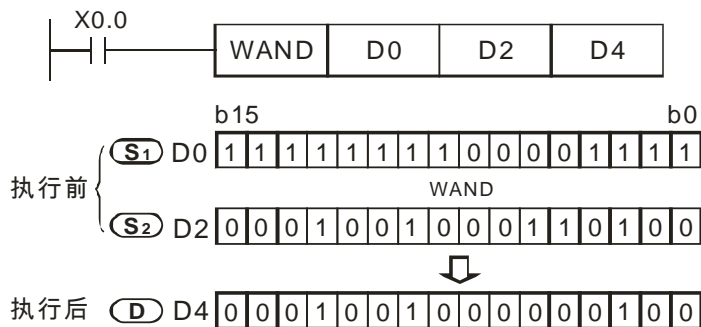
脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。 D : 运算结果。
- ◆ 两个数据源： S_1 及 S_2 作逻辑的'及' (AND) 运算并将结果存于 D 。
- ◆ 逻辑的'及' (AND) 运算之规则为任一为 0 结果为 0。

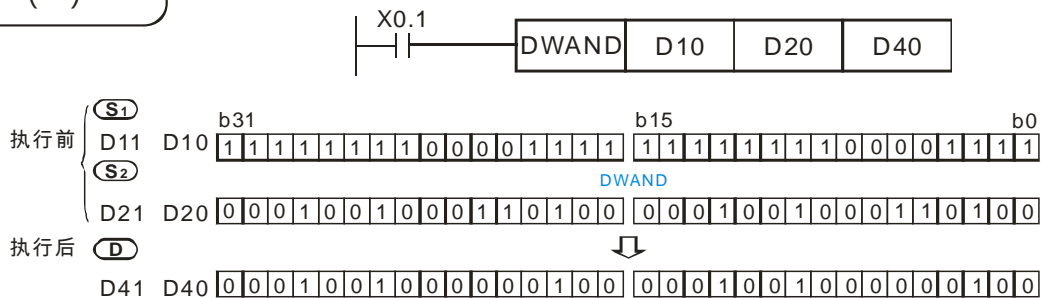
程序范例 (一)

- ◆ 当 X0.0=ON 时，16 位 D0 与 D2 作 WAND，逻辑及 (AND) 运算，将结果存于 D4 中。



程序范例 (二)

- ◆ 当 X0.1=ON 时，32 位 (D11、D10) 与 (D21、D20) 作 DWAND，逻辑及 (AND) 运算，将结果存于 (D41、D40) 中。



5

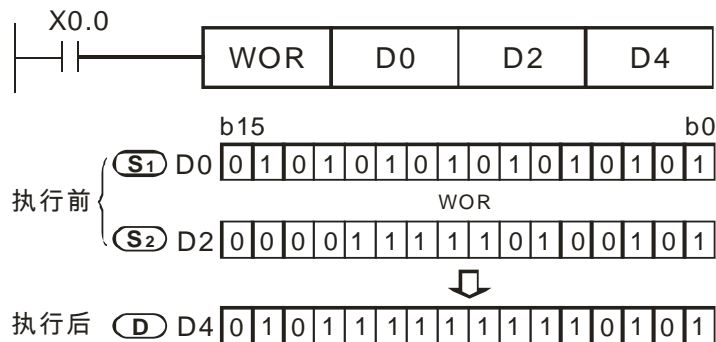
API	指令码			操作数									功能		
	D	WOR	P	$S_1 \cdot S_2 \cdot D$									逻辑或 (OR) 运算		
27															
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○
脉冲执行型							16 位指令 (7 steps)			32 位指令 (9 steps)					
✓							✓			✓					

指令说明

- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。 D : 运算结果。
- ◆ 两个数据源： S_1 及 S_2 作逻辑的‘或’ (OR) 运算结果存于 D 。
- ◆ 逻辑的‘或’ (OR) 运算之规则为任一为 1 结果为 1。

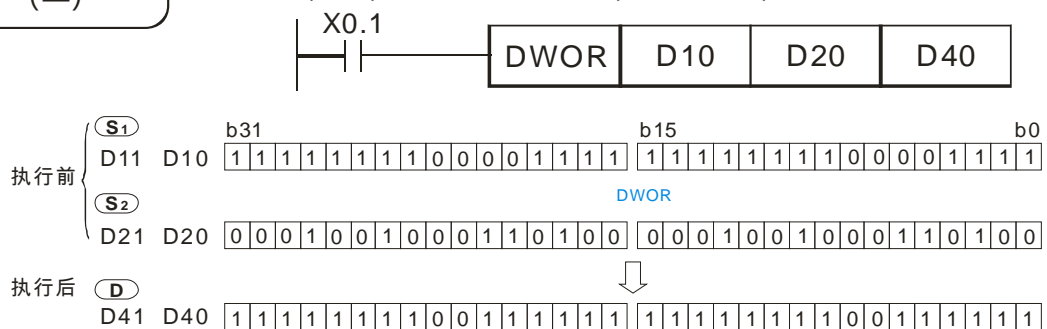
程序范例 (一)

- ◆ 当 X0.0=ON 时，16 位 D0 与 D2 作 WOR，逻辑或 (OR) 运算，将结果存于 D4 中。



程序范例 (二)

- ◆ 当 X0.1=ON 时，32 位 (D11、D10) 与 (D21、D20) 作 DWOR，逻辑或 (OR) 运算，将结果存于 (D41、D40) 中。



5

API	指令码			操作数							功能				
28	D	WXOR	P	$S_1 \cdot S_2 \cdot D$							逻辑异或 (XOR) 运算				

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

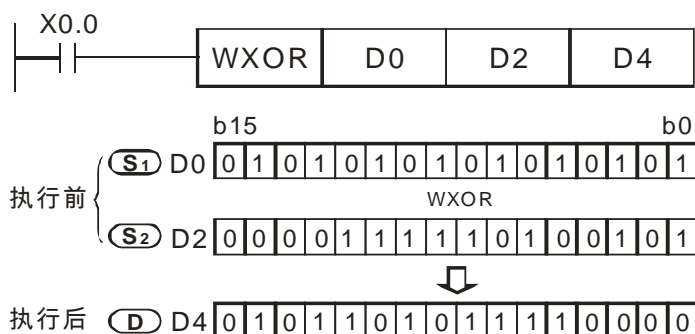
脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。 D : 运算结果。
- ◆ 两个数据源： S_1 及 S_2 作逻辑的'异或' (XOR) 运算结果存于 D 。
- ◆ 逻辑的'异或' (XOR) 运算之规则为两者相同结果为 0，两者不同结果为 1。

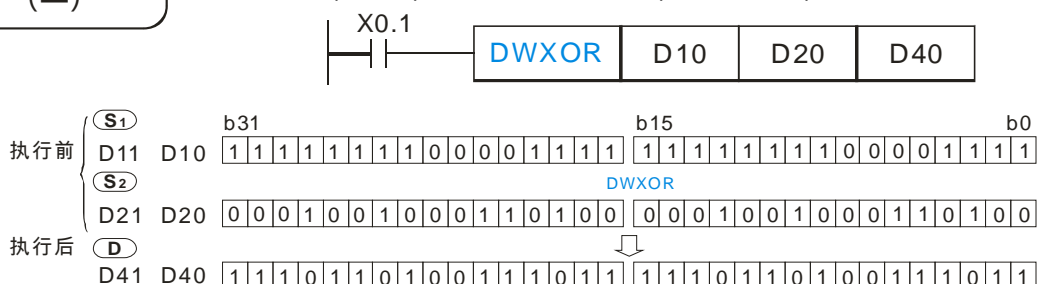
程序范例 (一)

- ◆ 当 X0.0=ON 时，16 位 D0 与 D2 作 WXOR，逻辑异或 (XOR) 运算，将结果存于 D4 中。



程序范例 (二)

- ◆ 当 X0.1=ON 时，32 位 (D11、D10) 与 (D21、D20) 作 DXOR，逻辑异或 (XOR) 运算，将结果存于 (D41、D40) 中。



5

API	指令码			操作数								功能			
29	D	NEG	P	D								取负数 (取 2 的补数)			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
脉冲执行型		16 位指令 (3 steps)				32 位指令 (3 steps)									
✓		✓				✓									

指令说明

- ◆ D：欲取 2 的补码之装置。本指令可将负数的 BIN 值转换成绝对值。
- ◆ 本指令一般都是使用脉冲执行型指令 (NEGP、DNEGP)。

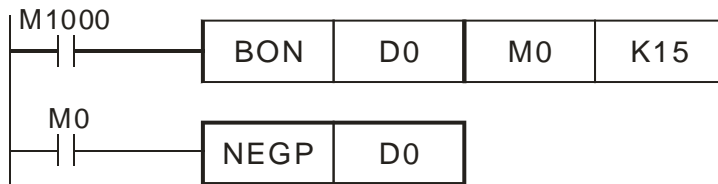
程序范例 (一)

- ◆ 当 X0.0=OFF→ON 时，D10 内容的各位全部反相 (0→1、1→0) 后再加 1 存放于原寄存器 D10 当中。



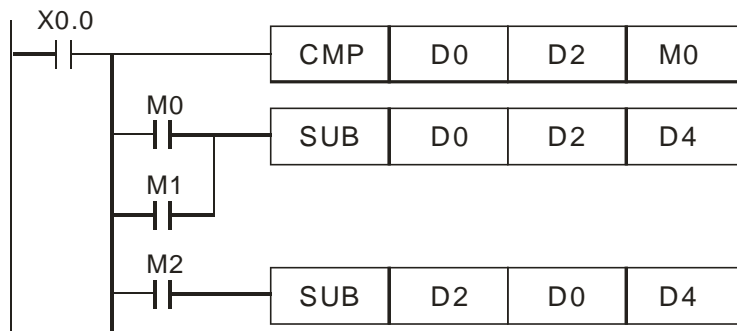
程序范例 (二)

- ◆ 求负数的绝对值：
 1. 当 D0 的 b15 为“1”时，M0=ON。(D0 表示为负数)
 2. M0=ON 时，用 NEG 指令将 D0 取 2 的补码可得到其绝对值。



程序范例 (三)

- ◆ 减法运算之差取绝对值，当 X0.0=ON 时：
 1. 若 D0>D2 时，M0=ON。
 2. 若 D0=D2 时，M1=ON。
 3. 若 D0<D2 时，M2=ON。
 4. 此可得 D4 保持为正值。



补充说明

◆ 负数的表现及绝对值

1. 正负数是以寄存器最上位（最左边）的位内容来表现，为“0”时，为正数、为“1”时，为负数。
2. 遇到负数时，可使用 **NEG** 指令将它转成绝对值。

(D0)=2
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

(D0)=1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D0)=0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(D0)=-1 → $(\overline{D0})+1=1$
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 → 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D0)=-2 → $(\overline{D0})+1=2$
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 → 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

(D0)=-3 → $(\overline{D0})+1=3$
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 → 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

(D0)=-4 → $(\overline{D0})+1=4$
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 → 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

(D0)=-5 → $(\overline{D0})+1=5$
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 → 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1

⋮

(D0)=-32,765 → $(\overline{D0})+1=32,765$
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 → 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1

(D0)=-32,766 → $(\overline{D0})+1=32,766$
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 → 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0

(D0)=-32,767 → $(\overline{D0})+1=32,767$
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 → 0 1

(D0)=-32,768 → $(\overline{D0})+1=-32,768$
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 → 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

↑
绝对值最大只可到 32,767

API	指令码			操作数								功能			
30	D	ROR	P	D · n								右旋转			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
n					○	○									

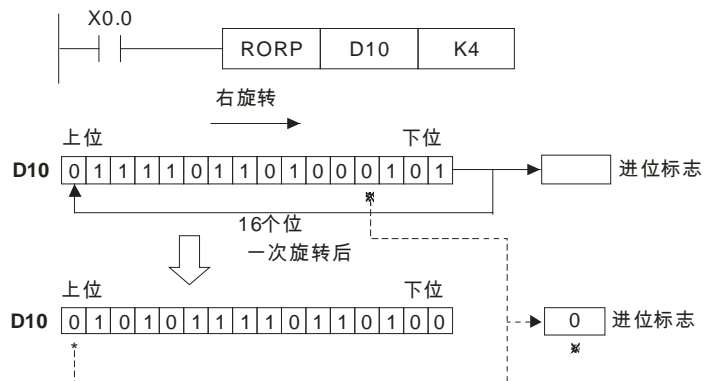
脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

指令说明

- ◆ D：欲旋转之装置。n：一次旋转之位数。
- ◆ 将 D 所指定的装置内容一次向右旋转 n 个位。
- ◆ 最后旋转的一位的状态被传送到进位标志。
- ◆ 本指令一般都是用脉冲执行型指令 (RORP · DRORP)。
- ◆ 如果操作数 D 指定为 KnM · KnS 时，只有 K4 (16 位) 及 K8 (32 位) 有效。
- ◆ n 操作数有效范围：1≤n≤16 (16 位) · 1≤n≤32 (32 位)。
- ◆ 进位标志，OX 是 SM1066 · O100 是 SM970。

程序范例

- ◆ 当 X0.0 从 OFF→ON 变化时，D10 的 16 个位以 4 个位为一组往右旋转，如下图所示标明*的位内容被传送至进位标志信号内。



API	指令码			操作数								功能
31	D	ROL	P	D · n								左旋转

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
n					○	○									

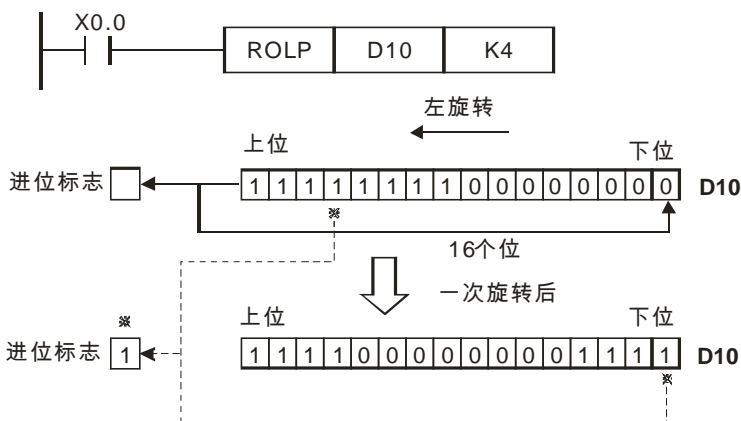
脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

指令说明

- ◆ D：欲旋转之装置。n：一次旋转之位数。
- ◆ 将 D 所指定的装置内容一次向左旋转 n 个位。
- ◆ 最后旋转的一位的状态被传送到进位标志 (进位标志)。
- ◆ 本指令一般都是用脉冲执行型指令 (ROLP · DROLP)。
- ◆ 如果操作数 D 指定为 KnM · KnS 时，只有 K4 (16 位) 及 K8 (32 位) 有效。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0 · K4S16 (十进制)。
- ◆ n 操作数有效范围：1 ≤ n ≤ 16 (16 位) · 1 ≤ n ≤ 32 (32 位)。
- ◆ 进位标志，OX 是 SM1066 · O100 是 SM970。

程序范例

- ◆ 当 X0.0 从 OFF→ON 变化时，D10 的 16 个位以 4 个位一组往左旋转，如下图所示标明※的位内容被传送到进位标志信号内。



API	指令码			操作数								功能			
32	D	RCR	P	D · n								附进位标志右旋转			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
n					○	○									

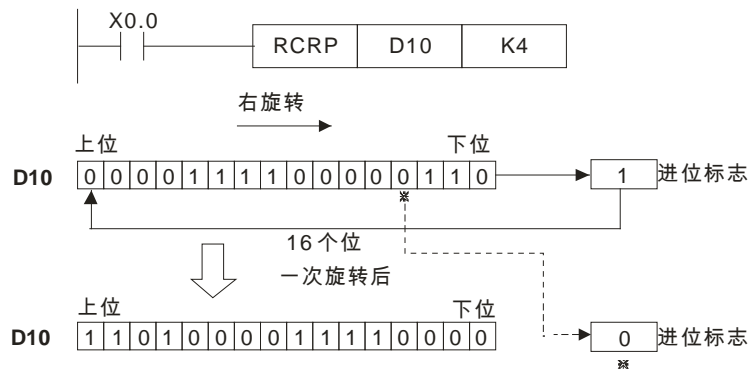
脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

指令说明

- ◆ D：欲旋转之装置。n：一次旋转之位数。
- ◆ 将 D 所指定的装置内容连同进位标志，一次向右旋转 n 个位。
- ◆ 最后旋转的一位的状态被传送到进位标志（进位标志）。在下面的指令操作时，进位标志第一个被传送到目的的设备。
- ◆ 本指令一般都是用脉冲执行型指令（RCRP · DRCRP）。
- ◆ 如果操作数 D 指定为 KnM · KnS 时，只有 K4（16 位）及 K8（32 位）有效。
- ◆ 指定 KnM（S）等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0 · K4S16（十进制）。
- ◆ n 操作数有效范围：1 ≤ n ≤ 16（16 位），1 ≤ n ≤ 32（32 位）。
- ◆ 进位标志 SM22。

程序范例

- ◆ 当 X0.0 从 OFF→ON 变化时，D10 的 16 个位连同进位标志共 17 个位以 4 个位为一组往右旋转，如下图所示标明※的位内容被传送到进位标志信号内。



API	指令码			操作数							功能
33	D	RCL	P	D · n							附进位标志左旋转

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
n					○	○									

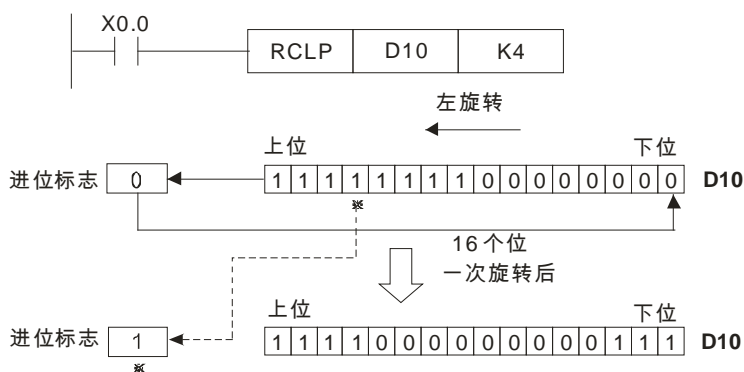
脉冲执行型	16 位指令 (5 steps)	32 位指令 (6 steps)
✓	✓	✓

指令说明

- ◆ D：欲旋转之装置。n：一次旋转之位数。
- ◆ 将 D 所指定的装置内容连同进位标志，一次向左旋转 n 个位。
- ◆ 最后旋转的一位的状态被传送到进位旗。在下面的指令操作时，进位标志第一个被传送到目的设备。
- ◆ 本指令一般都是用脉冲执行型指令 (RCLP · DRCLP)。
- ◆ 如果操作数 D 指定为 KnM · KnS 时，只有 K4 (16 位) 及 K8 (32 位) 有效。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0 · K4S16 (十进制)。
- ◆ n 操作数有效范围：1 ≤ n ≤ 16 (16 位) · 1 ≤ n ≤ 32 (32 位)。
- ◆ 进位标志 SM22。

程序范例

- ◆ 当 X0.0 从 OFF → ON 变化时，D10 的 16 个位连同进位标志共 17 个位以 4 个位一组往左旋转，如下图所示标明※的位内容被传送到进位标志信号内。



API	指令码			操作数								功能
34		SFTR	P	S · D · n ₁ · n ₂								位右移

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●											
D		●	●	●											
n ₁					○	○									
n ₂					○	○									

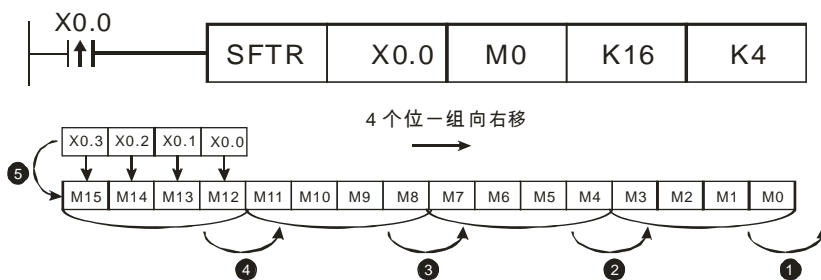
脉冲执行型	16 位指令 (9 steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：移位装置之起始编号。D：欲移位装置之起始编号。n₁：欲移位之数据长度。n₂：一次移位的位数。
- ◆ 将 D 开始之起始编号，具有 n₁ 个数字元（位移寄存器长度）的位装置，以 n₂ 位个数来右移。而 S 开始起始编号以 n₂ 位个数移入 D 中来填补位空位。
- ◆ 本指令一般都是使用脉冲执行型指令（SFTRP）。
- ◆ n₁ · n₂ 操作数有效范围：1 ≤ n₂ ≤ n₁ ≤ 1024。

程序范例

- ◆ 当 X0.0 OFF → ON 时，由 M0~M15 组成 16 位，以 4 位作右移。
- ◆ 扫描一次的位左移依照下列编号 ①~⑤ 动作。
 - ① M3~M0 → 进位
 - ② M7~M4 → M3~M0
 - ③ M11~M8 → M7~M4
 - ④ M15~M12 → M11~M8
 - ⑤ X0.3~X0.0 → M15~M12 完成



API	指令码			操作数								功能	
35		SFTL	P	S · D · n₁ · n₂								位左移	

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●											
D		●	●	●											
n₁					○	○									
n₂					○	○									

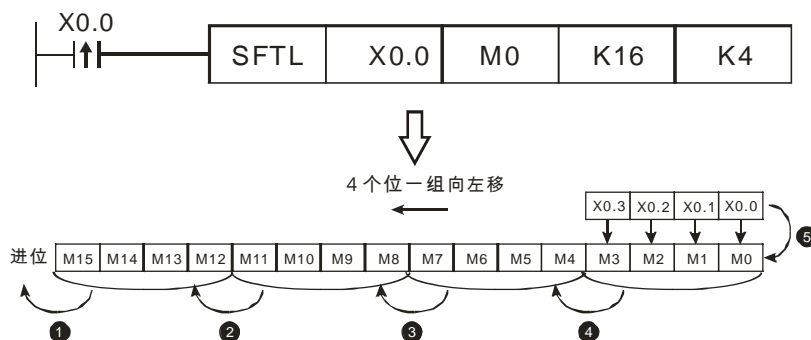
脉冲执行型	16 位指令 (9 steps)	32 位指令
✓	✓	-

指令说明

- ◆ **S** : 移位装置之起始编号。**D** : 欲移位装置之起始编号。**n₁** : 欲移位之数据长度。**n₂** : 一次移位的位数。
- ◆ 将 **D** 开始之起始编号, 具有 **n₁** 个数字元 (位移寄存器长度) 的位装置, 以 **n₂** 位个数来左移。而 **S** 开始起始编号以 **n₂** 位个数移入 **D** 中来填补位空位。
- ◆ 本指令一般都是使用脉冲执行型指令 (SFTRP)。
- ◆ **n₁ · n₂** 操作数有效范围 : $1 \leq n_2 \leq n_1 \leq 1024$ 。

程序范例

- ◆ 当 X0.0 OFF→ON 时, 由 M0~M15 组成 16 位, 以 4 位作左移。
- ◆ 扫描一次的位左移依照下列编号 ①~⑤ 动作。
 - ① M15~M12 → 进位
 - ② M11~M8 → M15~M12
 - ③ M7~M4 → M11~M8
 - ④ M3~M0 → M7~M4
 - ⑤ X0.3~X0.0 → M3~M0 完成



5

API	指令码			操作数								功能
36		WSFR	P	$S \cdot D \cdot n_1 \cdot n_2$								寄存器右移

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●		
D								●	●	●	●	●	●		
n ₁					○	○									
n ₂					○	○									

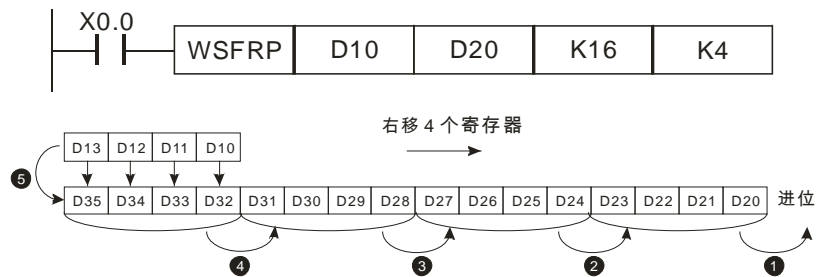
脉冲执行型	16 位指令 (9 steps)	32 位指令
✓	✓	-

指令说明

- ◆ **S** : 移位装置之起始编号。**D** : 欲移位装置之起始编号。**n₁** : 欲移位之数据长度。**n₂** : 一次移位之字符数。
- ◆ 将 **D** 开始之起始编号, 具有 **n₁** 个字符长度的数据串行, 以 **n₂** 个字符来右移。而 **S** 开始起始编号以 **n₂** 字符个数移入 **D** 中来填补字符空位。
- ◆ 本指令一般都是使用脉冲执行型指令 (WSFRP)。
- ◆ 指定 KnM (S) 等装置的位装置时, 以 10 进制中 16 的倍数为起始编号。如: K4M0 · K4S16 (十进制)。
- ◆ **n₁ · n₂** 操作数有效范围: $1 \leq n_2 \leq n_1 \leq 512$ 。

程序范例

- ◆ 当 X0.0 从 OFF→ON 时, 由 D20~D35 所组成的 16 个寄存器数据串行为位移区域, 以 4 个寄存器来右移。
- ◆ 扫描一次的字右移行作依照下列编号 ①~⑤ 动作。
 - ① D23~D20 → 进位
 - ② D27~D24 → D23~D20
 - ③ D31~D28 → D27~D24
 - ④ D35~D32 → D31~D28
 - ⑤ D13~D10 → D35~D32 完成



API	指令码			操作数								功能
37		WSFL	P	S · D · n₁ · n₂								寄存器左移

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●		
D								●	●	●	●	●	●		
n₁					○	○									
n₂					○	○									

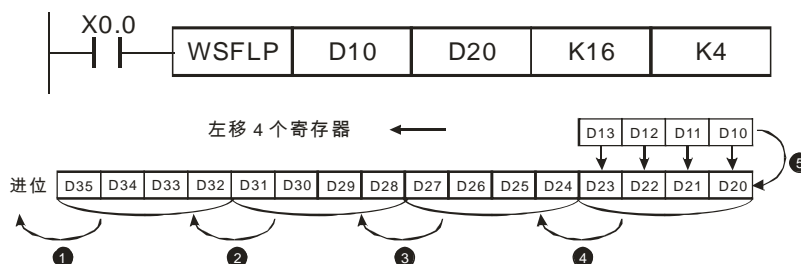
脉冲执行型	16 位指令 (9 steps)	32 位指令
✓	✓	-

指令说明

- ◆ **S** : 移位装置之起始编号。**D** : 欲移位装置之起始编号。**n₁** : 欲移位之数据长度。**n₂** : 一次移位之字符数。
- ◆ 将 **D** 开始之起始编号, 具有 **n₁** 个字符长度的数据串行, 以 **n₂** 个字符来左移。而 **S** 开始起始编号以 **n₂** 字符个数移入 **D** 中来填补字符空位。
- ◆ 本指令一般都是使用脉冲执行型指令 (WSFLP)。
- ◆ 指定 KnM (S) 等装置的位装置时, 以 10 进制中 16 的倍数为起始编号。如: K4M0 · K4S16 (十进制)。
- ◆ **n₁ · n₂** 操作数有效范围: $1 \leq n_2 \leq n_1 \leq 512$ 。

程序范例

- ◆ 当 X0.0 从 OFF→ON 时, 由 D20~D35 所组成的 16 个寄存器数据串行为位移区域, 以 4 个寄存器来左移。
- ◆ 扫描一次的字左移行作依照下列编号 ①~⑤ 动作。
 - ① D35~D32 → 进位
 - ② D31~D28 → D35~D32
 - ③ D27~D24 → D31~D28
 - ④ D23~D20 → D27~D24
 - ⑤ D13~D10 → D23~D20 完成



API	指令码				操作数								功能
38		SFWR	P		S · D · n								位移写入

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●		
n					○	○									

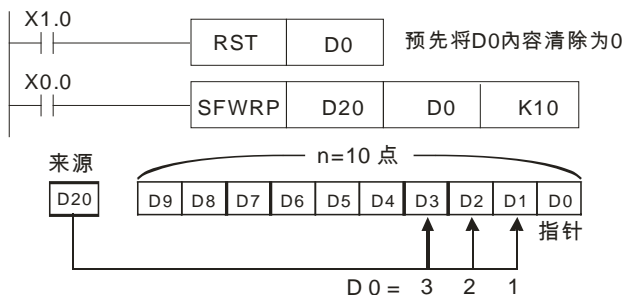
脉冲执行型	16 位指令 (9 steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：位移写入数据串行之装置。D：数据串行之起始编号。n：数据串行之长度。
- ◆ 将 D 起始编号开始 n 个字符装置的数据串行定义为先入先出数据串行，以第一个编号装置做为指针，当指令执行时，指针内容值先加 1，之后 S 所指定的装置其内容值会写入先入先出数据串行中由指针所指定的位置，当指针的内容超过 n-1 时，本指令不再处理写入的新值，进位标志信号 ON。
- ◆ 当指针的内容超过 n-1 时，本指令不再处理写入的新值，进位标志 SM22。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0，K4S16 (十进制)。
- ◆ 本指令一般都是使用脉冲执行型指令 (SFWRP)。
- ◆ n 操作数有效范围：2≤n≤512
- ◆ 零标志 SM20。

程序范例

- ◆ 开始先将指针 D0 清除为 0，当 X0.0=OFF→ON 变化时，D20 的内容被传送至 D1 当中，指针 D0 内容变成 1。变更 D20 的内容后，将 X0.0 再 OFF→ON 一次，则 D20 的内容被传送至 D2 当中，D0 内容变成 2。
- ◆ 指令执行一次位移写入动作依照下列编号①~②动作。
 - ①D20 的内容被传送至 D1 当中。
 - ②指针 D0 内容变成 1。



补充说明

- ◆ 本指令 API38 SFWR 与 API39 SFRD 可搭配使用，执行先入先出数据串行的写入读出控制。

API	指令码			操作数								功能	
39		SFRD	P	S · D · n								位移读出	

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●		
D								●	●	●	●	●	●	○	○
n					○	○									

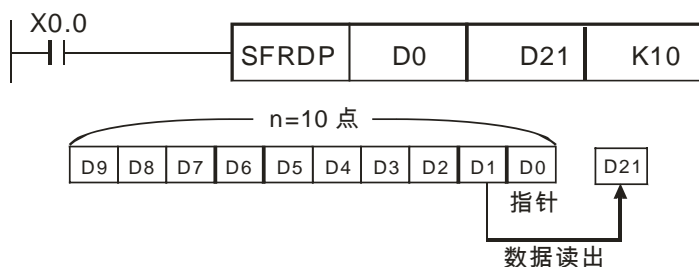
脉冲执行型	16 位指令 (7 steps)	32 位指令
✓	✓	-

指令说明

- ◆ **S** : 数据串行之起始编号。 **D** : 数据串行位移读出之装置。 **n** : 数据串行之长度。
- ◆ 将 **S** 起始编号开始 **n** 个字符装置的数据串行定义为先入先出数据串行，以第一个编号装置作为指针，当指令执行时，**S** 内容值先减 1，之后 (**S+1**) 所指定的装置其内容值会写入 **D** 所指定的位置，(**S+n-1**) ~ (**S+2**) 全部右移一个寄存器，(**S+n-1**) 的内容不变，当 **S** 的内容等于 0 时，本指令不再处理数据读出的动作，零标志信号 **SM20=ON**。
- ◆ 指定 KnM (**S**) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K1M0 · K4S16 (十进制)。
- ◆ 本指令一般都是使用脉冲执行型指令 (SFRDP)。
- ◆ **n** 操作数有效范围：2≤n≤512。

程序范例

- ◆ 当 X0.0 从 OFF→ON 变化时，D1 的内容被传送至 D21 内，D9~D2 全部往右位移一个寄存器 (D9 内容保持不变)，指针 D0 内容减 1。
- ◆ 执行指令一次位移读出动作依照下列编号 ❶~❸ 动作。
 - ❶ D1 的内容被读出传送至 D21 当中。
 - ❷ D9~D2 全部往右位移一个寄存器。
 - ❸ 指针 D0 内容减 1。



API	指令码				操作数								功能		
40		ZRST	P		$D_1 \cdot D_2$								区域清除		
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D_1		○	○	○						○	○	○	○		
D_2		○	○	○						○	○	○	○		

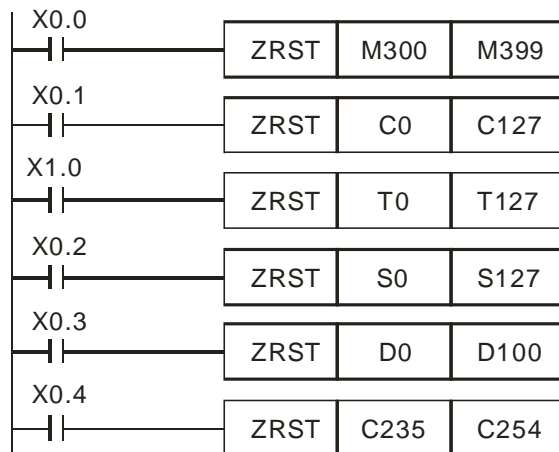
脉冲执行型	16 位指令 (5 steps)	32 位指令
✓	✓	-

指令说明

- ◆ D_1 : 区域清除起始装置。 D_2 : 区域清除结束装置。
- ◆ 16 位计数器与 32 位计数器可混在一起使用 ZRST 指令。
- ◆ 当 D_1 操作数编号 > D_2 操作数编号时，只有 D_2 指定之操作数被清除。

程序范例

- ◆ 当 X0.0 为 ON 时，辅助继电器 M300~M399 被清除成 OFF。
- ◆ 当 X0.1 为 ON 时，16 位计数器 C0~C127 全部清除。(写入 0，并将接点及线圈清除成 OFF)。
- ◆ 当 X1.0 为 ON 时，定时器 T0~T127 全部清除。(写入 0，并将接点及线圈清除成 OFF)。
- ◆ 当 X0.2 为 ON 时，步进点 S0~S127 被清除成 OFF。
- ◆ 当 X0.3 为 ON 时，数据寄存器 D0~D100 数据被清除为 0。
- ◆ 当 X0.4 为 ON 时，32 位计数器 C235~C254 全部清除。(写入 0，并将接点及线圈清除成 OFF)。



补充说明

- ◆ 装置可以单独使用清除指令 (RST)，如位装置 Y、M、S 和字符装置 T、C、D。

API	指令码			操作数								功能	
41		DECO	P	S · D · n								译码器	

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●						●	●	●	●	○	○
D		●	●	●						●	●	●	●	○	○
n					○	○									

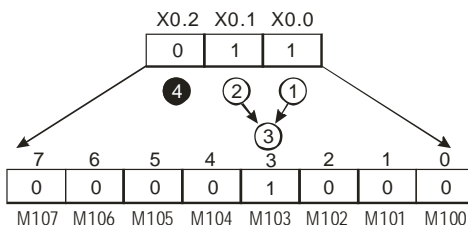
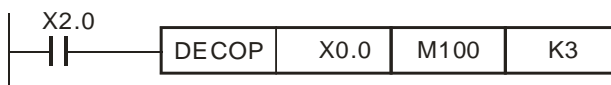
脉冲执行型	16 位指令 (7 steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：译码来源装置。D：存放译码结果之装置。n：译码位长度。
- ◆ 来源装置 S 的下位“n”位作译码，并将其“2ⁿ”位长度存于 D。
- ◆ 本指令一般都是使用脉冲执行型指令 (DECOP)。
- ◆ D 的范围为 1~8。

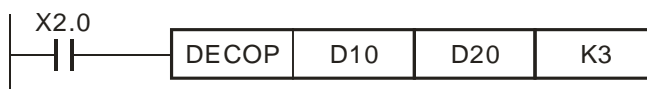
程序范例 (一)

- ◆ 当 D 是位装置时，n 有效范围：0<n≤8。但是如果 n=0 or n>8，会发生错误。
- ◆ 当 n=8 时，可做最大解码 2⁸=256 点。
- ◆ 当 X2.0 从 OFF→ON 时 指令将 X0.0~X0.2 内容值译码到 M100~M107。
- ◆ 如果 S=3，M103 (从 M100 开始算第 3 个位) =ON。
- ◆ 当指令执行后，X2.0 变为 OFF。已经做解码输出者照常动作。

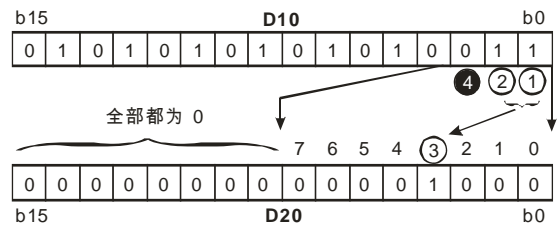


程序范例 (二)

- ◆ 当 D 为字符装置时，n 有效范围：0<n≤8，如果 n=0 or n>8，会发生错误。
- ◆ 当 n=8 时，可做最大解码 2⁸=256 点。
- ◆ 当 X2.0 从 OFF→ON 时，指令将 D10 中 (b2~b0) 的内容值解码到 D20 的 (b7~b0)。D20 中未被使用的位 (b15~b8) 全部变为 0。
- ◆ D10 的下位 3 位作译码存放于 D20 的下位 8 位，上 8 位皆为 0。
- ◆ 当指令执行后，X2.0 变为 OFF 后，已经做解码输出者照常动作。



5



API	指令码				操作数								功能
42		ENCO	P		S · D · n								译码器

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S	●	●	●	●						●	●	●	●	○	○
D										●	●	●	●	○	○
n					○	○									

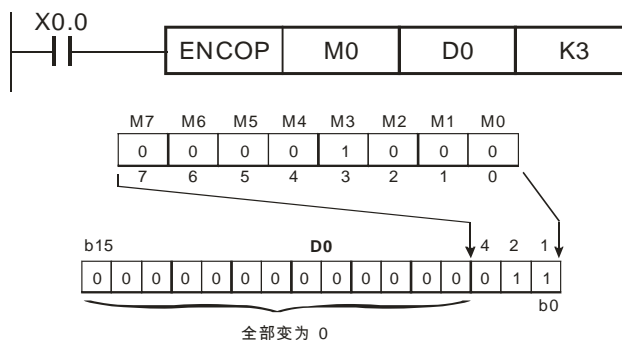
脉冲执行型	16 位指令 (7 steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：编码来源装置。D：存放编码结果的装置。n：编码位长度。
- ◆ 来源装置 S 的下位“2ⁿ”位长度的数据作编码，并将结果存于 D。
- ◆ 如果数据来源装置 S 有多数位为 1 时，则处理由高位往低位的第 1 个为 1 的位。
- ◆ 本指令一般都是使用脉冲执行型指令 (ENCOP)。
- ◆ 此指令有支持 V、Z 装置 (当 16 位指令时不能使用 Z 装置；当 32 位指令时不能使用 V 装置)
- ◆ S 为位装置时，n=1~8，当 S 为字符装置时，n=1~4。

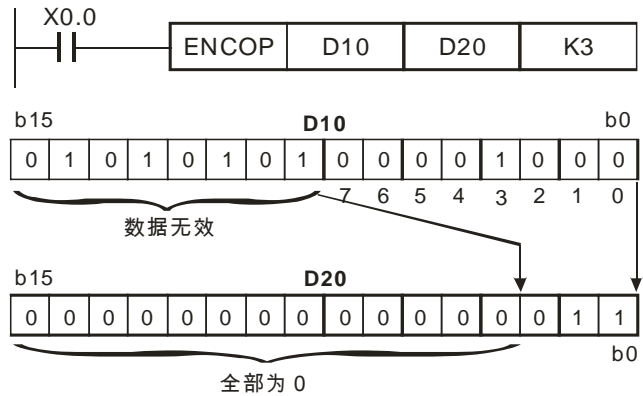
程序范例
(一)

- ◆ 当 S 为位装置时，n 有效范围：0<n≤8。如果 n=0 或者 n>8，会发生错误。
- ◆ 当 n=8 时，可做最大编码 2⁸=256 点。
- ◆ 当 X0.0 从 OFF→ON 时，指令将 2³ 位数据 (M0 to M7) 编码存放于 D0 的下位 3 位 (b2~b0) 内。D0 中未被使用的位 (b15~b3) 全部变为 0。
- ◆ 指令执行后，X0.0 变为 OFF 后，D 内数据不变。



程序范例
(二)

- ◆ 当 **S** 为字符装置时，**n** 有效范围： $0 < n \leq 4$ 。如果 $n=0$ 或者 $n>4$ 时，会发生错误。
- ◆ 当 $n=4$ 时，可做 $2^4=16$ 点编码。
- ◆ 当 X0.0 从 OFF→ON 时，D10 内 2^3 位数据 (b0~b7) 编码存放于 D20 之下位 3 位 (b2~b0) 内，D20 中未被使用之位 (b15~b3) 全部变为 0。(D10 内 b8~b15 为无效数据)。
- ◆ 当指令执行后，X0.0 变为 OFF，D 内数据不变。



API	指令码			操作数								功能			
43	D	SUM	P	S · D								ON 位数量			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●	○	○
D										●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
✓	✓	✓

指令说明

- ◆ S：来源装置。D：存放计数值的目的地装置。
- ◆ 在 S 中，所有位内容为“1”的总数将被储存于 D。
- ◆ 如果来源装置 S 的 16 个位全部为“0”时，零标志信号 ON。
- ◆ 当使用 32 位指令的时候，D 会占用两个寄存器。
- ◆ KnM (S) 等装置的位装置起始编号限定为 16 的倍数 (包含 0)，例如：K1M0 · K4M16。
- ◆ 零点标志，OX 是 SM1064 · O100 是 SM968。

5

程序范例



API	指令码			操作数								功能			
44	D	BON	P	S · D · n								ON 位判定			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D		●	●	●											
n										●	●	●	●	○	○

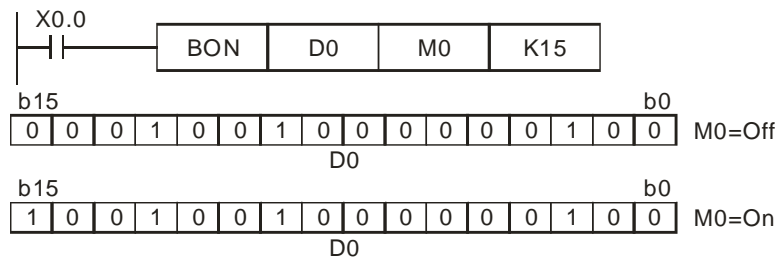
脉冲执行型	16 位指令 (7 steps)	32 位指令 (8 steps)
✓	✓	✓

指令说明

- ◆ **S** : 来源装置。 **D** : 存放判定结果的装置。 **n** : 指定判定的位 (自 0 开始编号)。
- ◆ 来源装置特定位的状态被表示在目标位置。
- ◆ 操作数 **n** 的有效范围 : **n=0~15** (16 位) · **n=0~31** (32 位)。
- ◆ **KnM** (**S**) 等装置的位装置起始编号限定为 **16** 的倍数 (包含 0) · 例如 : **K1M0** · **K4M16**。

程序范例

- ◆ 当 **X0.0=ON** 时 , 若是 **D0** 的第 15 个位为“1”时 , **M0=ON** , 为“0”时 , **M0=OFF**。
- ◆ **X0.0** 变成 **OFF** , **M0** 仍保持之前的状态。



API	指令码			操作数							功能
45	D	MEAN	P	S · D · n							平均值

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D		●	●	●											
n										●	●	●	●	○	○

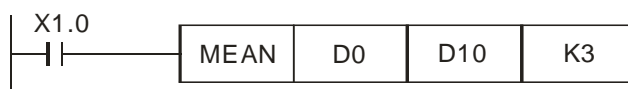
脉冲执行型	16 位指令 (7 steps)	32 位指令 (8 steps)
✓	✓	✓

指令说明

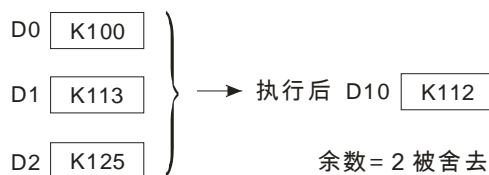
- ◆ S：欲取平均值之起始装置。D：存放平均值的装置。n：取平均值的装置个数。
- ◆ 将 S 起始的 n 个装置内容相加后取平均值存入 D 中。
- ◆ 如果计算中出现余数时，余数会被舍去。
- ◆ 如果 S 没有在有效范围内，只有正常范围内的装置编号被处理。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0 · K4S16 (十进制)。
- ◆ n 如果是 1~64 以外的数值时，PM 认定为“指令运算错误”。
- ◆ 操作数 n 的有效范围：n=1~64。

程序范例

- ◆ 当 X1.0=ON 时，D0 开始算的 3 个 (n=3) 寄存器的内容全部相加，相加之后再除以 3 以求得平均值并存于指定的 D10 当中，余数被舍去。



$(D0+D1+D2)/3 \rightarrow D10$



5

API	指令码			操作数								功能
46		ANS	P	S · m · D								报警点输出

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
m		●	●	●											
D										●	●	●	●	○	○

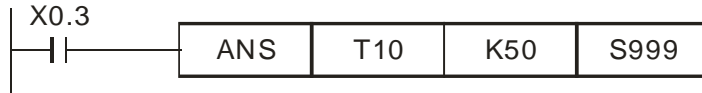
脉冲执行型	16 位指令 (7steps)	32 位指令
✓	✓	-

指令说明

- ◆ S：侦测警报定时器。m：计时时间设定。n：警报点装置。
- ◆ ANS 指令是用来驱动警报点输出的专用指令。
- ◆ 操作数 S 有效范围：T0~T183
操作数 m 有效范围：K1~K32,767 单位 100 ms。
操作数 D 有效范围：S912~S1023
见 ANR 更多的信息
- ◆ 此指令有支持 V、Z 装置 (当 16 位指令时不能使用 Z 装置；当 32 位指令时不能使用 V 装置)。

程序范例

- ◆ X0.3=ON 超过 5 秒钟时 警报点 S999=ON 之后就算是 X0.3 变成 OFF，S999 会继续保持 ON。(但是 T10 会复位成 OFF、当前值=0)。



API	指令码		操作数	功能
47	ANR	P	—	警报点复位

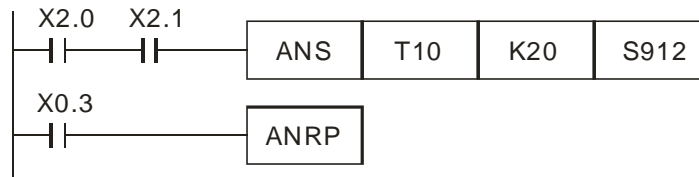
脉冲执行型	16 位指令 (1 step)	32 位指令
✓	✓	—

指令说明

- ◆ ANR 指令是用来复位警报点的专用指令。
- ◆ 复数个警报点同时 ON 的时候，较小号码的警报点被复位。
- ◆ 本指令一般都是使用脉冲执行型指令 (ANRP)。

程序范例

- ◆ X2.0 与 X2.1 同时 ON 超过 2 秒钟时，警报点 S912=ON 如 X2.0 或 X2.1 变成 OFF，警报点 S912 会继续保持 ON。T10 会被复位成 OFF，当前值为 0。
- ◆ X2.0 与 X2.1 同时 ON 未满足 2 秒钟时，T10 的当前值被复位成 0。
- ◆ X0.3 从 OFF→ON 时，动作中的警报点 S912~S1023 被复位。
- ◆ X0.3 再次从 OFF→ON 时，次小号码警报点被复位。



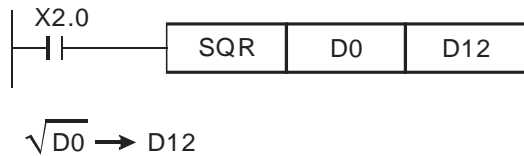
API	指令码			操作数								功能					
48	D	SQR	P	S · D								开平方根					
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z		
S					○	○				●	●						
D										●	●						
										脉冲执行型		16 位指令 (5 steps)			32 位指令 (6 steps)		
										✓		✓			✓		

指令说明

- ◆ **S**：欲开平方根之来源装置。**D**：存放结果的装置。
- ◆ 将 **S** 所指定的装置内容值开平方根后，存放于 **D** 所指定装置。
- ◆ **S** 只可以指定正数，若指定负数时，**PM** 视为错误，指令不执行。
- ◆ 运算结果 **D** 只求整数，小数点被舍弃。有小数点被舍弃时，借位标志信号 **ON**。
- ◆ 运算结果 **D** 为 0 时，零标志信号 **ON**。
- ◆ 零点标志，**OX** 是 **SM1064**，**O100** 是 **SM968**。
- ◆ 借位标志，**OX** 是 **SM1065**，**O100** 是 **SM969**。
- ◆ 进位标志，**OX** 是 **SM1066**，**O100** 是 **SM970**。

程序范例

- ◆ 当 **X2.0=ON**，将 **D0** 内容值开平方根后，存放于 **D12**。



API	指令码			操作数							功能				
49	D	FLT	P	S · D							BIN 整数→2 进小数点变换				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○				●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：变换来源装置。D：存放变换结果之装置。
- ◆ 将 BIN 整数变换成 2 进小数点值。
 1. 若转换结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
 2. 若转换结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
 3. 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

程序范例

- ◆ 当 X1.1=ON 时，将 D1、D0 (内为 BIN 整数) 变换成 D21、D20 (2 进小数点值)。
- ◆ 若 32-bit 寄存器 D0(D1)=K100000 则 X1.1=ON 转换后浮点数之 32-bit 数值为 16#4735000，存于 32-bit 寄存器 D20 (D21) 内。



5

API	指令码				操作数							功能			
50		REF	P		D · n							I/O 更新处理			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D	○	○													
n					○	○									

脉冲执行型	16 位指令 (5 steps)	32 位指令
✓	✓	-

指令说明

- ◆ D：欲 I/O 更新处理起始装置。n：I/O 更新处理数目。
- ◆ PM 的输入/出端子的状态全部为程序扫描至 END 后，才作状态的更新，其中输入点的状态是在程序开始扫描时，自外部输入点的状态读入存在输入点内存中，而输出端子在 END 指令后，才将输出点内存内容送至输出装置。因此在运算过程中需要最新的输入/出数据，则可利用本指令。
- ◆ D 操作数必须指定 X0.0、X1.0、Y0.0、Y1.0... 等小数点后第一位为 0 之编号。数字扩充机的 I/O 点无法使用此指令做立即更新处理。
- ◆ D 操作数指定的输入点及输出点仅限于主机的 I/O 点：
 - ◆ D 指定 X0.0 且 n<=8，则只有 X0.0~X0.7 会被更新，如果 n>8，则主机上所有输入及输出点都会被更新。
 - ◆ D 指定 Y0.0 且 n<=8，则只有 Y0.0~Y0.7 会被更新，如果 n>8，则主机上所有输入及输出点都会被更新。
- ◆ n 操作数范围 n=4~主机的 I/O 点，且为 4 之倍数。
- ◆ 当 X0.0=ON 时，AH500 运动模块可以读到 X0.0~X0.7 输入点的状态，输入信号更新，并没有输入延迟。

程序范例 (一)



程序范例 (二)

- ◆ 当 X0.0=ON 时，Y0.0~Y0.3 的 4 点输出信号将实时被送至输出端，不必到 END 指令才输出。



API	指令码			操作数								功能			
61	D	SER	P	$S_1 \cdot S_2 \cdot D \cdot n$								多点比较			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1								●	●	●	●	●	●		
S_2					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●		
n					○	○				○	○				

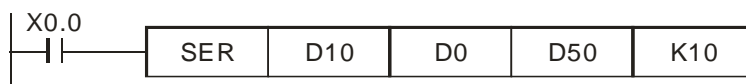
脉冲执行型	16 位指令 (9 steps)	32 位指令 (11 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 多点比较的数据区块的起始装置。 S_2 : 欲比较的数值数据。 D : 存放比较结果的起始装置 (占用 5 个连续的装置)。 n : 被比较的数据区块长度。
- ◆ S_1 指定被比较寄存器区域的号码, n 指定被比较的笔数, 该多笔被比较寄存器的内容与 S_2 所指定的数据作比较, 比较结果被存放于 D 所指定的数个寄存器当中。
- ◆ 使用 32 位指令时若指定寄存器, $S_1 \cdot S_2 \cdot D \cdot n$ 会指定 32 位寄存器。
- ◆ n 操作数范围: $n=1\sim 256$ (16 位指令); $n=1\sim 128$ (32 位指令)。
- ◆ 指定 KnM (S) 等装置的位装置时, 以 10 进制中 16 的倍数为起始编号。如: K4M0, K4S16 (十进制)。

程序范例

- ◆ 当 X0.0=ON 时, 由 D10~D19 组成之数据区块与 D0 作比较, 结果存放在 D50~D54 中, 当相等值不存在时, D50~D52 的内容全部为 0。
- ◆ 大小比较以代数型态进行。($-10 < 2$)。
- ◆ 所有比较数据之最小值编号记录在 D53, 最大值编号记录在 D54。当最小值最大值不只一个时, 会记录编号大者。



S_1	内容值	比较数据	数据编号	比较结果	D	内容值	说明
D10	88	S_2 D0=K100	0		D50	4	相等值的数据个数
D11	100		1	相等	D51	1	第一个相等值的编号
D12	110		2		D52	8	最后一个相等值的编号
D13	150		3		D53	7	最小值的编号
D14	100		4	相等	D54	9	最大值的编号
D15	300		5				
D16	100		6	相等			
D17	5		7	最小			
D18	100		8	相等			
D19	500		9	最大			

5

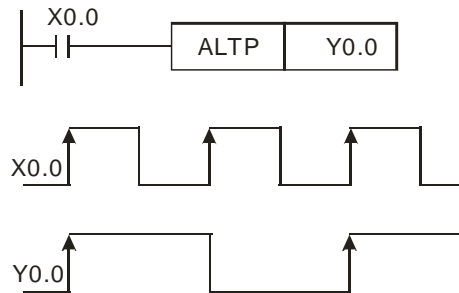
API	指令码			操作数								功能			
66		ALT	P	D								ON/OFF 交替			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D	●	●	●												
脉冲执行型								16 位指令 (3 steps)			32 位指令				
✓								✓			-				

指令说明

- ◆ D：目的地装置。
- ◆ ALT 指令执行时，D ON/OFF 交替。
- ◆ 本指令一般都是使用脉冲执行型指令 (ALTP)。

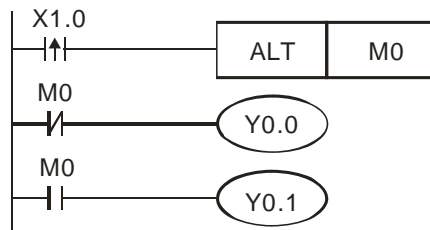
程序范例 (一)

- ◆ 当第一次 X0.0 从 OFF→ON 时，Y0.0=ON。第二次 X0.0 从 OFF→ON 时，Y0.0=OFF。



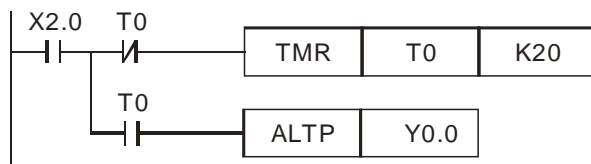
程序范例 (二)

- ◆ 使用单一开关控制启动与停止。一开始时，M0=OFF 故 Y0.0=ON、Y0.1=OFF，当 X1.0 作第一次 ON/OFF 时，M0=ON 故 Y0.1=ON、Y0.0=OFF，第二次 ON/OFF 时，M0=OFF 故 Y0.0=ON 而 Y0.1=OFF。



程序范例 (三)

- ◆ 如下例所示，Y0.0 会产生闪烁的动作。当 X2.0=ON 时，T0 每隔 2 秒产生一个脉冲，Y0.0 输出会依 T0 脉冲做 ON/OFF 交替变化。



API	指令码			操作数								功能	
78	D	FROM	P	$m_1 \cdot m_2 \cdot D \cdot n$								特殊模块 CR 数据读出	

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
m_1					○	○				●	●	●	●	○	○
m_2					○	○				●	●	●	●	○	○
D										●	●	●	●		
n					○	○				●	●	●	●	○	○

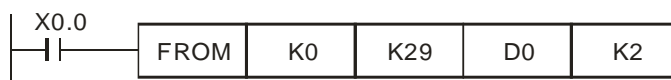
脉冲执行型	16 位指令 (9 steps)	32 位指令 (12 steps)
✓	✓	✓

指令说明

- ◆ m_1 : 特殊模块所在之编号 · 使用范围 (16 及 32 位指令): 0~255 · m_2 : 欲读取特殊模块之 CR (Controlled Register) 编号 · 使用范围 (16 及 32 位指令): 0~499 · D : 存放读取数据的位置 · n : 一次读取之数据笔数 · 16 位指令使用范围 : 1~ (500- m_2) · 而 32 位指令使用范围 : 1~ (500- m_2) /2 ·
- ◆ 运动控制模块利用此指令读取内部特殊模块之 CR 数据 ·

程序范例

- ◆ 将编号为 0 特殊模块的 CR#29 的内容读出至运动控制模块的 D0 当中 · CR#30 的内容读出至运动控制模块的 D1 当中 · 一次读取二笔 (n=2) ·
- ◆ X0.0=ON 的时候指令被执行 · X0.0 变成 OFF 时 · 指令不被执行 · 之前读出的数据其内容没有变化 ·



5

API	指令码			操作数							功能		
79	D	TO	P	$m_1 \cdot m_2 \cdot S \cdot n$							特殊模块 CR 数据写入		

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
m_1					○	○				●	●	●	●	○	○
m_2					○	○				●	●	●	●	○	○
S					○	○				●	●	●	●	○	○
n					○	○				●	●	●	●	○	○

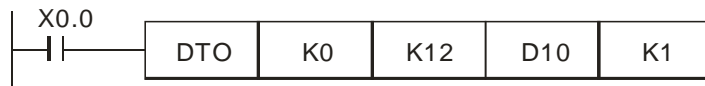
脉冲执行型	16 位指令 (9 steps)	32 位指令 (13 steps)
✓	✓	✓

指令说明

- ◆ m_1 : 特殊模块所在之编号，使用范围 (16 及 32 位指令) : 0~255。 m_2 : 欲写入特殊模块之 CR (Controlled Register) 编号，使用范围 (16 及 32 位指令) : 0~499。 S : 写入 CR 的数据。 n : 一次写入之数据笔数，16 位指令使用范围 : 1~ (500- m_2)，而 32 位指令使用范围 : 1~ (500- m_2) / 2。
- ◆ 运动控制模块利用此指令将数据写入内部特殊模块之 CR 内。

程序范例

- ◆ 使用 32 位指令 DTO，程序的动作是将 D11、D10 的内容写入编号为 0 之特殊模块的 CR#13、#12 当中，一次只写入一笔 ($n=1$)。
- ◆ X0.0=ON 时，指令被执行，X0.0 变成 OFF 时，指令不执行，写入的数据没有变化。

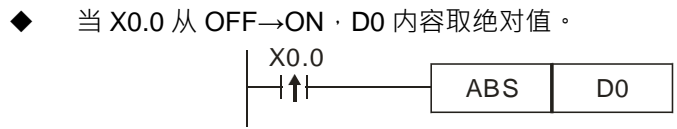


API	指令码			操作数											功能
87	D	ABS	P	D											绝对值
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
D								●	●	●	●	●	●	○	○
脉冲执行型								16 位指令 (3 steps)				32 位指令 (3 steps)			
✓								✓				✓			

指令说明

- ◆ **D** : 欲取绝对值之装置。
- ◆ 当 ABS 指令执行，被指定的组件 **D** 取绝对值。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。
如：K4M0 · K4S16 (十进制)。
- ◆ 本指令一般都是使用脉冲执行型指令 (ABSP · DABSP)。

程序范例



5

API	指令码			操作数								功能
110	D	ECMP	P	$S_1 \cdot S_2 \cdot D$								二进制浮点数比较

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D		●	●	●											

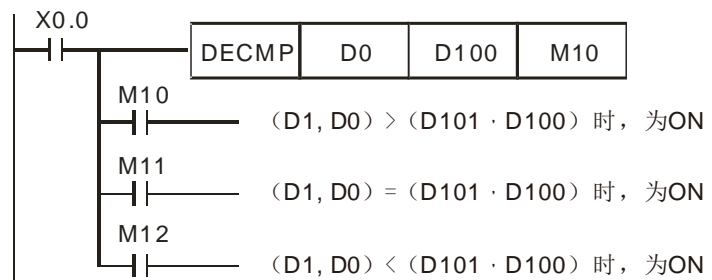
脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 二进制浮点数比较值 1。 S_2 : 二进制浮点数比较值 2。 D : 比较结果，占用连续 3 点。
- ◆ 二进制浮点数值 1 与二进制浮点数比较值 2 作比较，比较的结果 (> 、 = 、 <) 在 D 作表示。
- ◆ S_1 或 S_2 来源操作数若是指定浮点数 F 的话，指令会以该二进制浮点数值来作比较。
- ◆ F 表示浮点数输入，使用时要有小数点。

程序范例

- ◆ 指定装置为 M10，则自动占有 M10~M12。
- ◆ 当 X0.0=ON 时，DECMP 指令执行，M10~M12 其中之一会 ON，当 X0.0=OFF 时，DECMP 指令不执行，M10~M12 状态保持在 X0.0 =OFF 之前的状态。
- ◆ 若需要得到 \geq 之结果时，可将 M10 和 M11 串联即可取得。若需要得到 \leq 之结果时，可将 M11 和 M12 串联即可取得。若需要得到 \neq 之结果时，可将 M10~M12 串联即可取得。
- ◆ 若要清除其结果请使用 RST 或 ZRST 指令。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
111	D	EZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$								二进浮点数区域比较			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
S							○			●	●				
D		●	●	●											

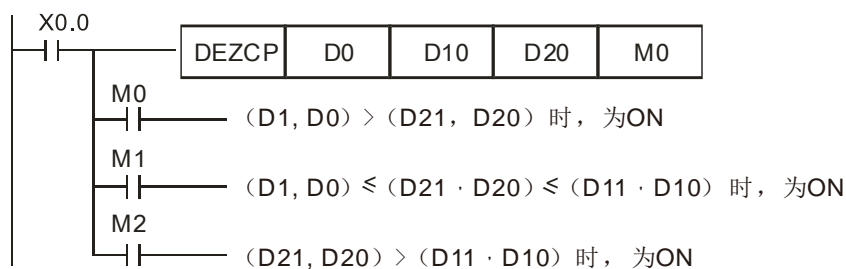
脉冲执行型	16 位指令	32 位指令 (12 steps)
✓	-	✓

指令说明

- ◆ S_1 : 区域比较之二进浮点数下限值。 S_2 : 区域比较之二进浮点数上限值。 S : 二进浮点数比较值。 D : 比较结果，占用连续 3 点。
- ◆ 二进浮点数比较值 S 与二进浮点数下限值 S_1 及二进浮点数上限值 S_2 作比较，其比较结果在 D 作表示。
- ◆ S_1 或 S_2 来源操作数若是指定浮点数 F 的话，指令会以该二进浮点数值来作比较。
- ◆ 当二进浮点数下限值 S_1 大于二进浮点数上限值 S_2 时，则指令以二进浮点数下限值 S_1 做为上下限值进行比较。

程序范例

- ◆ 指定装置为 M0，则自动占有 M0~M2。
- ◆ 当 X0.0=ON 时，DEZCP 指令执行，M0~M2 其中之一会 ON，当 X0.0=OFF 时，EZCP 指令不执行，M0~M2 状态保持在 X0.0=OFF 之前的状态。
- ◆ 若要清除其结果请使用 RST 或 ZRST 指令。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
112	D	MOVR	P	S · D								浮点数值数据移动			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○								
D										●	●	●	●		
										脉冲执行型		16 位指令		32 位指令 (6 steps)	
										✓		-		✓	

指令说明

- ◆ S：浮点数值数据来源。D：数据之搬移目的地。
- ◆ 该指令可直接在 S 操作数输入浮点数值。
- ◆ 当该指令执行时，将 S 的内容直接搬移至 D，当指令不执行时，D 内容不会变化。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0 · K4S16 (十进制)。

程序范例

- ◆ 当 X0.0=OFF 时，D10 · D11 内容没有变化。当 X0.0=ON 时，将 F1.2 浮点数当前值传送至 D10 · D11 数据寄存器内。



API	指令码			操作数							功能				
116	D	RAD	P	S · D							角度→弧度				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

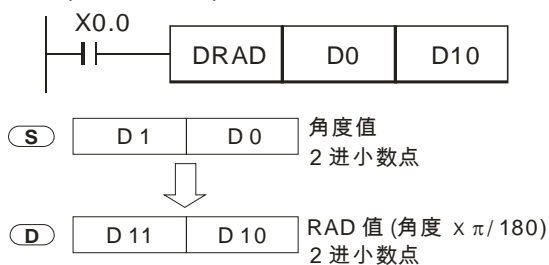
脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：数据来源（角度）。D：变换之结果（弧度）。
- ◆ 使用下列公式将角度转换成弧度。
弧度 = 角度 × (π/180)
- ◆ 若转换结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若转换结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

程序范例

- ◆ 当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 之角度值，将角度转换成弧度值后存于 (D11 · D10) 当中，内容为二进浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

5

API	指令码			操作数								功能			
117	D	DEG	P	S · D								弧度→角度			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				
										脉冲执行型		16 位指令		32 位指令 (6 steps)	
										✓		-		✓	

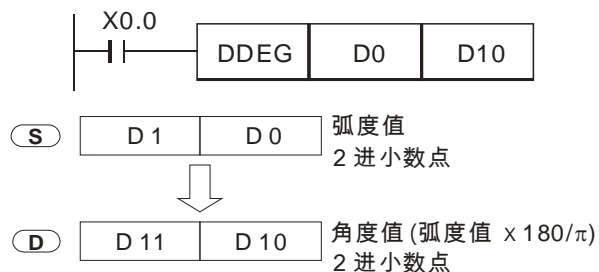
指令说明

- ◆ S：数据来源（弧度）。D：变换之结果（角度）。
- ◆ 使用下列公式将弧度转换成角度。

$$\text{角度} = \text{弧度} \times (180/\pi)$$
- ◆ 若转换结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若转换结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

程序范例

- ◆ 当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 之弧度值，将弧度值转换成角度后存于 (D11 · D10) 当中，内容为二进浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数							功能		
120	D	EADD	P	$S_1 \cdot S_2 \cdot D$							二进浮点数加算		

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

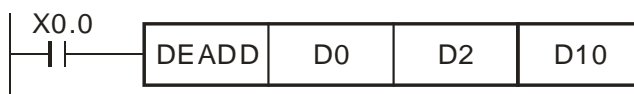
脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 被加数。 S_2 : 加数。 D : 和。
- ◆ S_1 所指定的寄存器内容加上 S_2 所指定的寄存器内容，和被存放至 D 所指定的寄存器当中，加算的动作全部以二进浮点数型态进行。
- ◆ S_1 或 S_2 来源操作数若是指定浮点数 F 的话，指令会以该二进浮点数值来作加算。
- ◆ S_1 及 S_2 可指定相同的寄存器编号，此种情况下若是使用“连续执行”型态的指令时，在条件接点 ON 的期间，该寄存器于每一次扫描时，均会被加算一次，一般的情况下都是使用脉冲执行型指令 (DEADDP)。
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

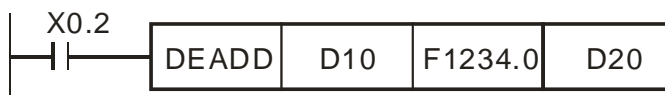
程序范例 (一)

- ◆ 当 X0.0=ON 时，将二进浮点数 (D1 · D0) + 二进浮点数 (D3 · D2)，结果存放在 (D11 · D10) 中。



程序范例 (二)

- ◆ 当 X0.2=ON 时，将二进浮点数 (D11 · D10) + F1234.0 (自动变换为二进浮点数)，结果存放在 (D21 · D20) 中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

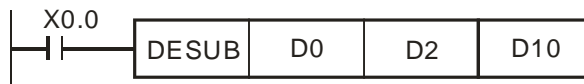
API	指令码			操作数								功能			
121	D	ESUB	P	$S_1 \cdot S_2 \cdot D$								二进制浮点数减算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				
							脉冲执行型	16 位指令		32 位指令 (9 steps)					
							✓	-		✓					

指令说明

- ◆ S_1 : 被减数。 S_2 : 减数。 D : 差。
- ◆ S_1 所指定的寄存器内容减掉 S_2 所指定的寄存器内容，差被存放至 D 所指定的寄存器当中，减算的动作全部以二进制浮点数型态进行。
- ◆ S_1 或 S_2 来源操作数若是指定浮点数 F 的话，指令会以该二进制浮点数值来作减算。
- ◆ S_1 及 S_2 可指定相同的寄存器编号，此种情况下若是使用“连续执行”型态的指令时，在条件接点 ON 的期间，该寄存器于每一次扫描时，均会被减算一次，一般的情况下都是使用脉冲执行型指令 (DESUBP)。
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

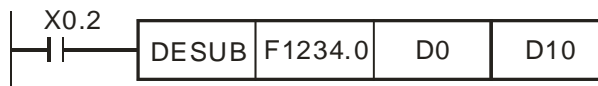
程序范例 (一)

- ◆ 当 X0.0=ON 时，将二进制浮点数 (D11 · D10) - 二进制浮点数 (D3 · D2)，结果存放在 (D11 · D10) 中。



程序范例 (二)

- ◆ 当 X0.2=ON 时，将 F1234.0 (自动变换为二进制浮点数) - 二进制浮点数 (D11 · D10)，结果存放在 (D11 · D10) 中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

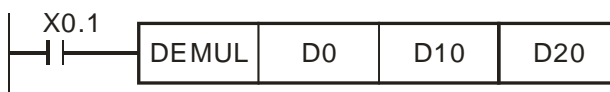
API	指令码			操作数								功能			
122	D	EMUL	P	$S_1 \cdot S_2 \cdot D$								二进浮点数乘算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				
										脉冲执行型		16 位指令		32 位指令 (9 steps)	
										✓		-		✓	

指令说明

- ◆ S_1 : 被乘数。 S_2 : 乘数。 D : 积。
- ◆ S_1 所指定的寄存器内容乘上 S_2 所指定的寄存器内容，积被存放至 D 所指定的寄存器当中，乘算的动作全部以二进浮点数型态进行。
- ◆ S_1 或 S_2 来源操作数若是指定浮点数 F 的话，指令会以该二进浮点数值来作乘算。
- ◆ S_1 及 S_2 可指定相同的寄存器编号，此种情况下若是使用“连续执行”型态的指令时，在条件接点 ON 的期间，该寄存器于每一次扫描时，均会被乘算一次，一般的情况下都是使用使用脉冲执行型指令 (**DEMULP**)。
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

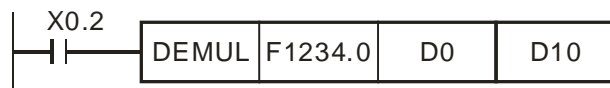
程序范例
(一)

- ◆ 当 X0.1=ON 时，将二进浮点数 (D1 · D0) 乘上二进浮点数 (D11 · D10) 将积存放至 (D21 · D20) 所指定的寄存器当中。



程序范例
(二)

- ◆ 当 X0.2=ON 时，将 F1234.0 (自动变换为二进浮点数) × 二进浮点数 (D1 · D0)，结果存放在 (D11 · D10) 中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
123	D	EDIV	P	$S_1 \cdot S_2 \cdot D$								二进浮点数除算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

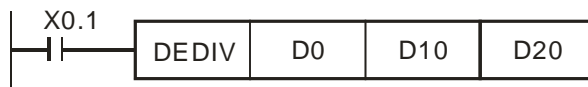
脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 被除数。 S_2 : 除数。 D : 商及余数。
- ◆ S_1 所指定的寄存器内容除以 S_2 所指定的寄存器内容，商被存放至 D 所指定的寄存器当中，除算的动作全部以二进浮点型态进行。
- ◆ S_1 或 S_2 来源操作数若是指定浮点数 F 的话，指令会以该二进浮点数值来作除算。
- ◆ 除数 S_2 的内容若为 0 即被认定为“运算错误”，指令不执行，运算错误标志 ON，并记录错误码 16#0E19。
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

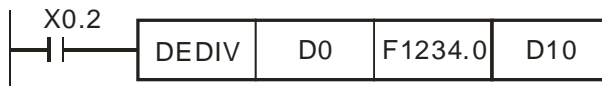
程序范例 (一)

- ◆ 当 X0.1=ON 时，将二进浮点数 (D1 · D0) 除以二进浮点数 (D11 · D10) 将商存放至 (D21 · D20) 所指定的寄存器当中。



程序范例 (二)

- ◆ 当 X0.2=ON 时，将二进浮点数 (D1 · D0) ÷ F1234.0 (自动变换为二进浮点数)，结果存放在 (D11 · D10) 中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
124	D	EXP	P	S · D								二进制浮点数取指数			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

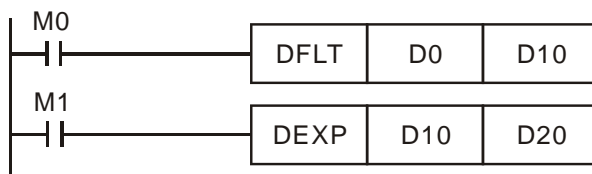
脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：运算来源装置。D：运算结果装置。
- ◆ $EXP^{[D+1 \cdot D]} = [S+1 \cdot S]$ ，以 $e=2.71828$ 为底数，S 为指数做 EXP 运算。
- ◆ S 内容正负数都有效，指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S 需为浮点数值。
- ◆ D 操作数内容值= e^S ； $e=2.71828$ ，S 为指定的来源数据
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

程序范例

- ◆ 当 M0 为 ON 时，将 (D1 · D0) 值转成二进制浮点数存于 (D11 · D10) 寄存器中。
- ◆ 当 M1 为 ON 时，(D11 · D10) 为指数做 EXP 运算，其值为二进制浮点数值并存放于 (D21 · D20) 寄存器中。
- ◆ 当 M2 为 ON 时，将 (D21 · D20) 二进制浮点数值转成十进浮点数值并存放于 (D31 · D30) 寄存器中。(此时 D31 为表示 D30 的 10 次幂方)



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
125	D	LN	P	S · D								二进浮点数取自然对数			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

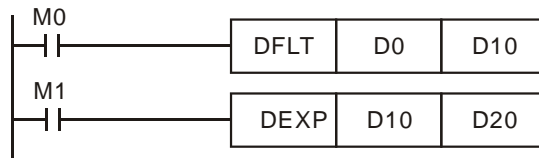
脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：运算来源装置。D：运算结果装置。
- ◆ 以 S 为操作数做自然对数 ln 运算。
LN[S+1 · S]=[D+1 · D]
- ◆ S 内容只有正数有效，指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S 需为浮点数值。
- ◆ S 内容若不为正数即被认定为“运算错误”，指令不执行，运算错误标志 ON，并记录错误码 16#0E19。
- ◆ $e^D=S \rightarrow D$ 操作数内容值=lnS；S 为指定的来源数据
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

程序范例

- ◆ 当 M0 为 ON 时，将 (D1 · D0) 值转成二进浮点数存于 (D11 · D10) 寄存器中。
- ◆ 当 M1 为 ON 时，将 (D11 · D10) 寄存器为真数做 ln 运算，其值为二进浮点数并存放于 (D21 · D20) 寄存器中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数							功能		
126	D	LOG	P	$S_1 \cdot S_2 \cdot D$							二进浮点数取对数		

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

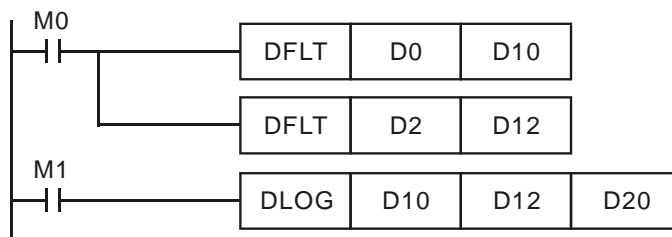
脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 运算底数装置。 S_2 : 运算来源装置。 D : 运算结果装置。
- ◆ 将 S_1 内容及 S_2 内容为操作数做 log 运算，结果存放于 D 。
- ◆ S_1 、 S_2 内容值只有正数有效，指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S_1 、 S_2 需为浮点数值。
- ◆ $S_1^D = S_2$ ，求 D 值 $\rightarrow \text{Log}_{S_1} S_2 = D$
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

程序范例

- ◆ 当 M0 为 ON 时，将 (D11 · D10) 内容及 (D13 · D12) 内容转成二进浮点数分别存于 (D11 · D10) 及 (D13 · D12) 32 位寄存器中。
- ◆ 当 M1 为 ON 时，将 (D11 · D10) 及 (D13 · D12) 32 位寄存器二进浮点数值做 log 运算并将结果存于 (D21 · D20) 32 位寄存器中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

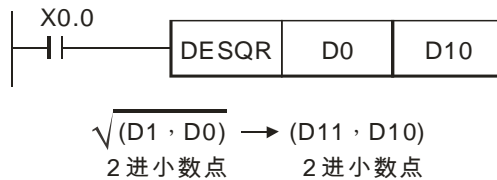
API	指令码			操作数								功能			
127	D	ESQR	P	S · D								二进浮点数开平方根			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				
										脉冲执行型		16 位指令		32 位指令 (6 steps)	
										✓		-		✓	

指令说明

- ◆ S：欲开平方根来源装置。D：开平方根之结果。
- ◆ S 所指定的寄存器内容被开平方，所得的结果暂存于 D 所指定的寄存器内容，开平方的动作全部以二进浮点数型态进行。
- ◆ S 来源操作数若是指定浮点数 F 的话，指令会将该浮点数转为二进浮点数值来作运算。
- ◆ 若开平方根之结果为 0 时，零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。
- ◆ 来源运算只有正数有效，负数时，视为“运算错误”，指令不执行，运算错误标志 ON，并记录错误码 16#0E19。

程序范例 (一)

- ◆ 当 X0.0=ON 时，将二进浮点数 (D1 · D0) 取开平方根，将结果存放至 (D11 · D10) 所指定的寄存器当中。



程序范例 (二)

- ◆ 当 X0.2=ON 时，将 F1234.0 (自动变换为二进浮点数) 取开平方根，结果存放在 (D11 · D10) 中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
128	D	POW	P	$S_1 \cdot S_2 \cdot D$								浮点数权值指令			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

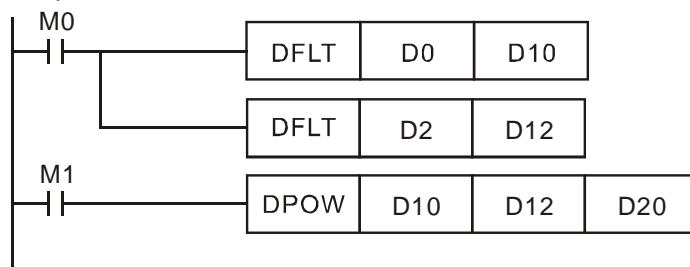
脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 底数装置。 S_2 : 次幂数装置。 D : 运算结果装置。
- ◆ 将二进浮点数据 S_1 及 S_2 以次幂数相乘后存放于 D 。
 $D = POW[S_1+1 \cdot S_1][S_2+1 \cdot S_2]$
- ◆ S_1 内容值只有正数有效， S_2 内容值正负值都有效。指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S_1 、 S_2 需为浮点数值。
- ◆ 若 S_1 及 S_2 的内容值无效，视为“运算错误”，指令不执行，运算错误标志 ON，并记录错误码 16#0E19。
- ◆ 若运算结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若运算结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

程序范例

- ◆ 当 M0 为 ON 时，将 (D11，D10) 内容及 (D3，D2) 内容转成二进浮点数分别存于 (D11，D10) 及 (D13，D12) 32 位寄存器中。
- ◆ 当 M1 为 ON 时，将 (D11，D10) 及 (D13，D12) 32 位寄存器二进浮点数做 pow 运算并将结果存于 (D21，D20) 32 位寄存器中。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数							功能				
129	D	INT	P	S · D							二进浮点数→BIN 整数变换				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S										●	●				
D										●	●				

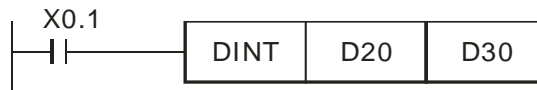
脉冲执行型	16 位指令	32 位指令 (5 steps)
✓	-	✓

指令说明

- ◆ S：欲变换之来源装置。D：变换之结果。
 - ◆ S 所指定的寄存器内容以二进浮点数型态被变换成 BIN 整数暂存于 D 所指定的寄存器当中，BIN 整数浮点数被舍弃。
 - ◆ 本指令的动作与 API 49 DFLT 指令刚好相反。
 - ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
 - ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
 - ◆ 进位标志，OX 是 SM1066，O100 是 SM970。
 - ◆ 变换结果若为 0 时，零标志=ON。
- 变换结果有浮点数被舍弃时，借位标志=ON。
- 变换结果若超出下列范围时（进位），进位标志=ON。
- 32 位指令：-2,147,483,648~2,147,483,647。

程序范例

- ◆ 当 X0.1=ON 时，将二进浮点数（D21，D20）变换成 BIN 整数将结果存放至（D31，D30）当中，BIN 整数浮点数被舍弃。



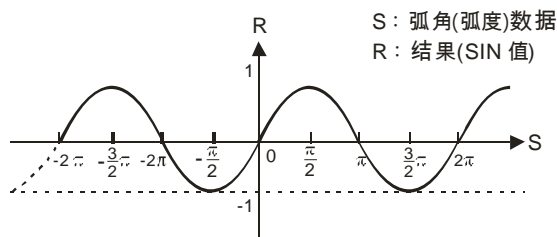
API	指令码			操作数							功能				
130	D	SIN	P	S · D							二进浮点数 SIN 运算				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源值。D：取 SIN 值结果。
- ◆ S 所指定的来源可指定为弧度或角度，由弧度/角度使用标志决定。
- ◆ 当弧度/角度使用标志=OFF 时，指定为弧度模式，弧度 (RAD) 值等于 (角度×π/180)。
- ◆ 当弧度/角度使用标志=ON 时，指定为角度模式，角度范围：0°≤角度值 < 360°。
- ◆ 当计算结果若为 0 时，零标志=ON。
- ◆ 将 S 所指定的来源值，求取 SIN 值后存于 D 所指定的寄存器当中。

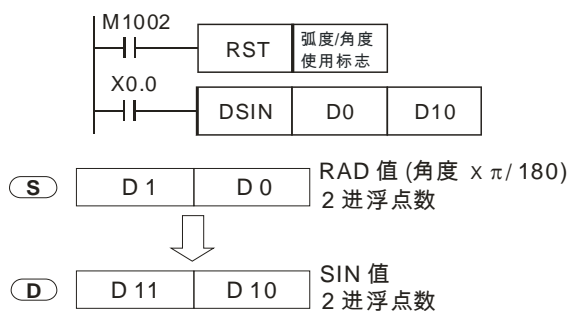
下图显示弧角与结果的关系：



- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

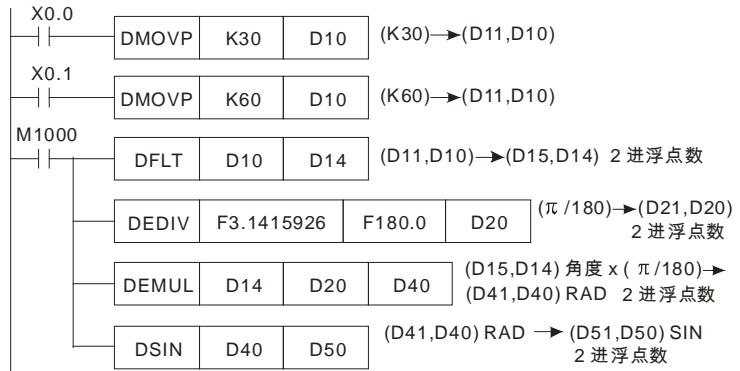
程序范例
(一)

- ◆ 弧度/角度使用标志=OFF，指定为弧度模式，当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 之弧度 (RAD) 值求取 SIN 值后存于 (D11 · D10) 当中，内容为二进浮点数。



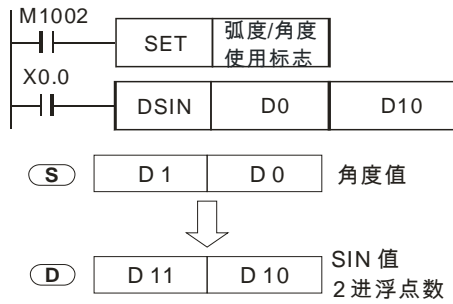
程序范例
(二)

- ◆ 弧度/角度使用标志=OFF，指定为弧度模式，由输入端 X0.0 及 X0.1 来选择角度，转成弧度 (RAD) 值后求取 SIN 值。



程序范例
(三)

- ◆ 弧度/角度使用标志=ON，指定为角度模式，当 X0.0=ON 时，指定 (D1, D0) 之角度值，角度范围： $0^{\circ} \leq \text{角度值} < 360^{\circ}$ 。求取 SIN 值后存于 (D11, D10) 当中，内容为二进浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数							功能		
131	D	COS	P	S · D							二进浮点数 COS 运算		

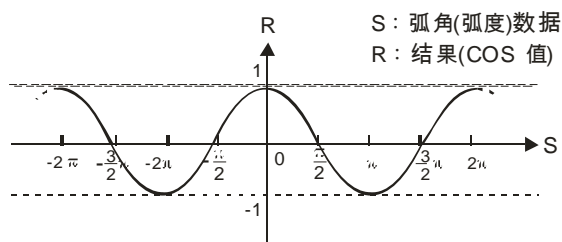
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源值。D：取 COS 值结果。
- ◆ S 所指定的来源可指定为弧度或角度，由标志弧度/角度使用标志决定。
- ◆ 当弧度/角度使用标志=OFF 时，指定为弧度模式，弧度 (RAD) 值等于 (角度 $\times\pi/180$)。
- ◆ 当弧度/角度使用标志=ON 时，指定为角度模式，角度范围： $0^{\circ}\leq$ 角度值 $<360^{\circ}$ 。
- ◆ 当计算结果若为 0 时，零标志=ON。
- ◆ 将 S 所指定的来源值，求取 COS 值后存于 D 所指定的寄存器当中。

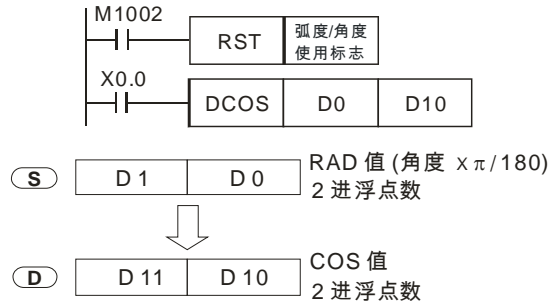
下图显示弧角与结果的关系：



- ◆ 标志弧度/角度使用标志弧度/角度切换：当弧度/角度使用标志=OFF 时，S 所指定的为弧度 (RAD) 值。当弧度/角度使用标志=ON 时，S 所指定的为角度值 ($0^{\circ}\sim 360^{\circ}$)。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

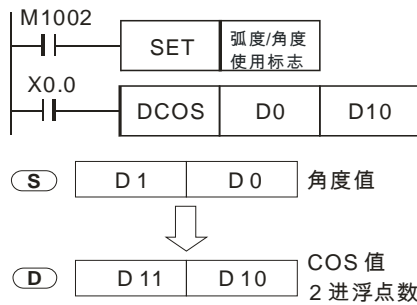
程序范例
(一)

- ◆ 弧度/角度使用标志=OFF，指定为弧度模式，当 X0.0=ON 时，指定二进制浮点数 (D1·D0) 之弧度值求取 COS 值后存于 (D11·D10) 当中，内容为二进制浮点数。



程序范例
(二)

- ◆ 弧度/角度使用标志=ON，指定为角度模式，当 X0.0=ON 时，指定 (D1·D0) 之角度值，角度范围： $0^{\circ} \leq \text{角度值} < 360^{\circ}$ 。求取 COS 值后存于 (D11·D10) 当中，内容为二进制浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

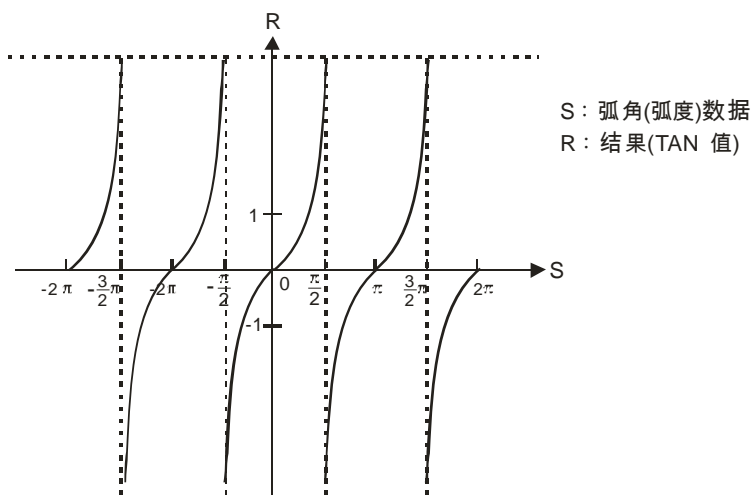
API	指令码			操作数								功能			
132	D	TAN	P	S · D								二进浮点数 TAN 运算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源值。D：取 TAN 值结果。
- ◆ S 所指定的来源可指定为弧度或角度，由标志弧度/角度使用标志决定。
- ◆ 当弧度/角度使用标志=OFF 时，指定为弧度模式，弧度 (RAD) 值等于 (角度×π/180)。
- ◆ 当弧度/角度使用标志=ON 时，指定为角度模式，角度范围：0°≤角度值 < 360°。
- ◆ 当计算结果若为 0 时，零标志=ON。
- ◆ 将 S 所指定的来源值，求取 TAN 值后存于 D 所指定的寄存器当中。

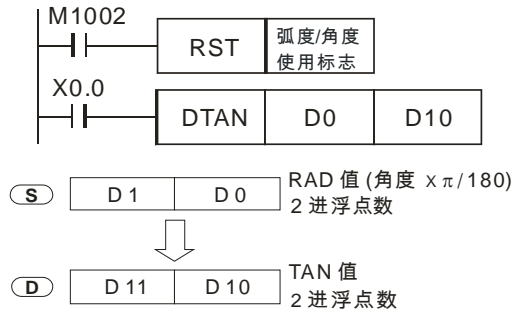
下图显示弧角与结果的关系：



- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

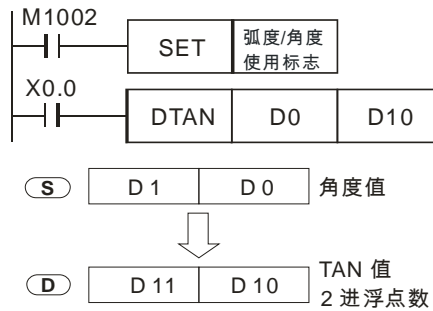
程序范例
(一)

- ◆ 弧度/角度使用标志=OFF，指定为弧度模式，当 X0.0=ON 时，指定二进制浮点数 (D1·D0) 之弧度 (RAD) 值求取 TAN 值后存于 (D11·D10) 当中，内容为二进制浮点数。



程序范例
(二)

- ◆ 弧度/角度使用标志=ON，指定为角度模式，当 X0.0=ON 时，指定 (D1·D0) 之角度值，角度范围： $0^{\circ} \leq \text{角度值} < 360^{\circ}$ 。求取 TAN 值后存于 (D11·D10) 当中，内容为二进制浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

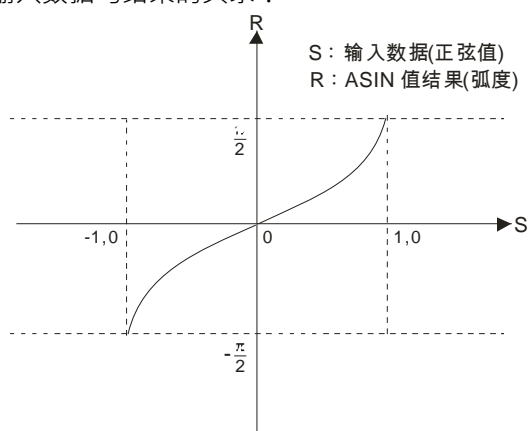
API	指令码			操作数								功能			
133	D	ASIN	P	S · D								二进浮点数 ASIN 运算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源（二进浮点数）。D：取 ASIN 值结果。
- ◆ $ASIN 值 = \sin^{-1}$

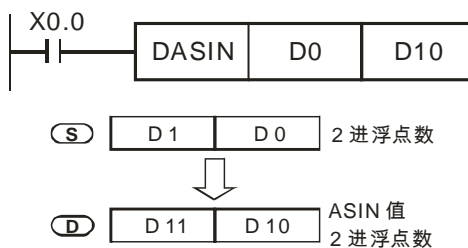
下图显示输入数据与结果的关系：



- ◆ S 操作数指定的正弦值数值之十进浮点值只能介于-1.0~+1.0 之间，若不在此范围内则指令不执行，运算错误标志 ON，并记录错误码 16#0E19。
- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

程序范例

- ◆ 当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 求取 ASIN 值后存于 (D11 · D10) 当中，内容为二进浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能
134	D	ACOS	P	S · D								二进浮点数 ACOS 运算

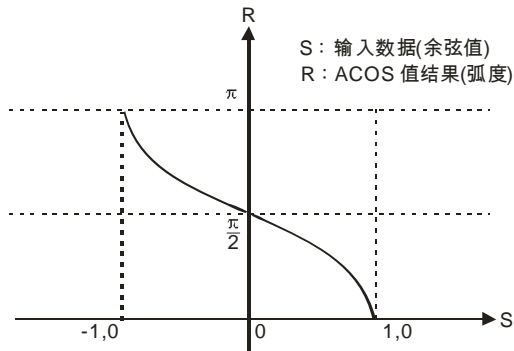
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源（二进浮点数）。D：取 ACOS 值结果。
- ◆ ACOS 值= \cos^{-1}

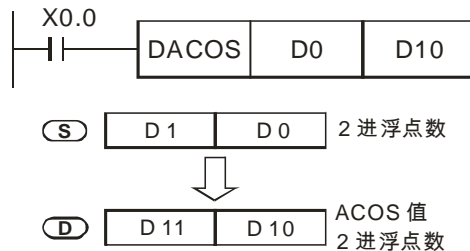
下图显示输入数据与结果的关系：



- ◆ S 操作数指定的余弦值数值之十进浮点值只能介于-1.0~1.0 之间，若不在此范围内则指令不执行，运算错误标志 ON，并记录错误码 16#0E19。
- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 运算错误标志，OX 是 SM1049，O100 是 SM953。

程序范例

- ◆ 当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 求取 ACOS 值后存于 (D11 · D10) 当中，内容为二进浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

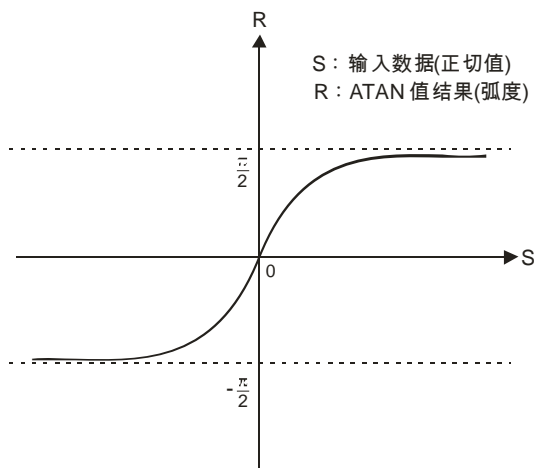
API	指令码			操作数								功能			
135	D	ATAN	P	S · D								二进制浮点数 ATAN 运算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源（二进制浮点数）。D：取 ATAN 值结果。
- ◆ ATAN 值 = \tan^{-1}

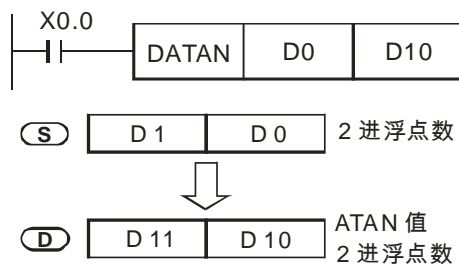
下图显示输入数据与结果的关系：



- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。

程序范例

- ◆ 当 X0.0=ON 时，指定二进制浮点数 (D1 · D0) 求取 TAN 值后存于 (D11 · D10) 当中，内容为二进制浮点数。



补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
136	D	SINH	P	S · D								二进浮点数 SINH 运算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

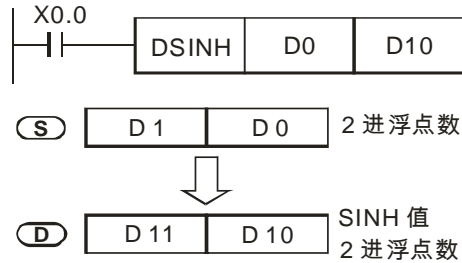
脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S：指定的来源（二进浮点数）。D：取 SINH 值结果。
- ◆ \sinh 值 = $(e^s - e^{-s}) / 2$

程序范例

- ◆ 当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 求取 SINH 值后存于 (D11 · D10) 当中，内容为二进浮点数。



- ◆ 若转换结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若转换结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志 · OX 是 SM1064 · O100 是 SM968。
- ◆ 借位标志 · OX 是 SM1065 · O100 是 SM969。
- ◆ 进位标志 · OX 是 SM1066 · O100 是 SM970。

补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数							功能
137	D	COSH	P	S · D							二进浮点数 COSH 运算

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				

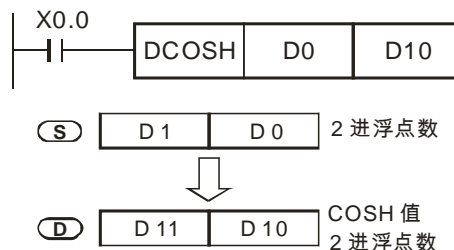
脉冲执行型	16 位指令	32 位指令 (6 steps)
✓	-	✓

指令说明

- ◆ S : 指定的来源 (二进浮点数) 。 D : 取 COSH 值结果。
- ◆ $\cosh \text{ 值} = (e^S + e^{-S}) / 2$

程序范例

- ◆ 当 X0.0=ON 时 , 指定二进浮点数 (D1 · D0) 求取 COSH 值后存于 (D11 · D10) 当中 , 内容为二进浮点数。



- ◆ 若转换结果的绝对值大于可表示之最大浮点值 , 则进位标志=ON。
- ◆ 若转换结果的绝对值小于可表示之最小浮点值 , 则借位标志=ON。
- ◆ 若转换结果为 0 , 则零标志=ON。
- ◆ 零点标志 , OX 是 SM1064 · O100 是 SM968。
- ◆ 借位标志 , OX 是 SM1065 · O100 是 SM969。
- ◆ 进位标志 , OX 是 SM1066 · O100 是 SM970。

补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

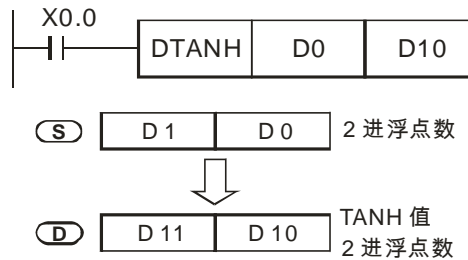
API	指令码			操作数								功能			
138	D	TANH	P	S · D								二进浮点数 TANH 运算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S							○			●	●				
D										●	●				
										脉冲执行型		16 位指令		32 位指令 (6 steps)	
										✓		-		✓	

指令说明

- ◆ S：指定的来源（二进浮点数）。D：取 TANH 值结果。
- ◆ $\tanh \text{ 值} = (e^s - e^{-s}) / (e^s + e^{-s})$

程序范例

- ◆ 当 X0.0=ON 时，指定二进浮点数 (D1 · D0) 求取 TANH 值后存于 (D11 · D10) 当中，内容为二进浮点数。



- ◆ 若转换结果的绝对值大于可表示之最大浮点值，则进位标志=ON。
- ◆ 若转换结果的绝对值小于可表示之最小浮点值，则借位标志=ON。
- ◆ 若转换结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

补充说明

- ◆ 浮点数的运算功能请参考第 5.3 章应用指令对数值的处理方式之说明。

API	指令码			操作数								功能			
172	D	ADDR	P	$S_1 \cdot S_2 \cdot D$								浮点数值加算			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 浮点数值被加数。 S_2 : 浮点数值被加数。 D : 和。
- ◆ $S_1 \cdot S_2$ 操作数可输入浮点数值。
- ◆ DADDR 指令可直接在 $S_1 \cdot S_2$ 操作数输入浮点数值 (例如 : F1.2) , 或以寄存器 D 存放浮点数值。
- ◆ 当 $S_1 \cdot S_2$ 操作数 , 以寄存器 D 存放浮点数值 , 其功能与 API 120 EADD 相同。
- ◆ 当 DADDR 指令执行时 , D 操作数将会存放浮点数值运算后的结果。
- ◆ $S_1 \cdot S_2$ 可指定相同的寄存器编号 , 此种情况下若是使用连续执行型态的指令时 , 在条件接点 ON 的期间 , 该寄存器于每一次扫描时 , 均会被加算一次 , 一般的情况下都是使用脉冲执行型指令 (DADDRP) 。
- ◆ 若运算结果的绝对值大于可表示的最大浮点数值 , 则进位标志=ON。若运算结果的绝对值小于可表示的最大浮点数值 , 则借位标志=ON。若运算结果为 0 , 则零标志=ON。
- ◆ 零点标志 , OX 是 SM1064 , O100 是 SM968。
- ◆ 借位标志 , OX 是 SM1065 , O100 是 SM969。
- ◆ 进位标志 , OX 是 SM1066 , O100 是 SM970。

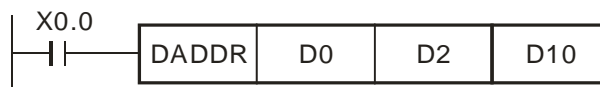
程序范例 (一)

- ◆ 当 X0.0=ON 时 , 将 F1.23456 浮点数值 , 加上 F1.0 浮点数值 , 其运算结果为 F2.23456 存放至 (D10 , D11) 数据寄存器内。



程序范例 (二)

- ◆ 当 X0.0=ON 时 , 将浮点数值 (D1 , D0) + 浮点数值 (D3 , D2) , 结果存放在 (D11 , D10) 中。



API	指令码			操作数								功能			
	173	D	SUBR	P	$S_1 \cdot S_2 \cdot D$								浮点数值减算		
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

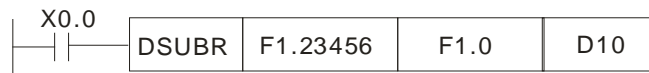
脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 浮点数值被减数。 S_2 : 浮点数值减数。 D : 差。
- ◆ $S_1 \cdot S_2$ 操作数可输入浮点数值。
- ◆ DSUBR 指令可直接在 $S_1 \cdot S_2$ 操作数输入浮点数值 (例如 : F1.2) , 或以寄存器 D 存放浮点数值。
- ◆ 当 $S_1 \cdot S_2$ 操作数 , 以寄存器 D 存放浮点数值 , 其功能与 API 121 ESUB 相同。
- ◆ 当 DSUBR 指令执行时 , D 操作数将会存放浮点数值运算后的结果。
- ◆ $S_1 \cdot S_2$ 可指定相同的寄存器编号 , 此种情况下若是使用连续执行型态的指令时 , 在条件接点 ON 的期间 , 该寄存器于每一次扫描时 , 均会被减算一次 , 一般的情况下都是使用脉冲执行型指令 (DSUBRP) 。
- ◆ 若运算结果的绝对值大于可表示的最大浮点数值 , 则进位标志=ON。若运算结果的绝对值小于可表示的最大浮点数值 , 则借位标志=ON。若运算结果为 0 , 则零标志=ON。
- ◆ 零点标志 , OX 是 SM1064 , O100 是 SM968。
- ◆ 借位标志 , OX 是 SM1065 , O100 是 SM969。
- ◆ 进位标志 , OX 是 SM1066 , O100 是 SM970。

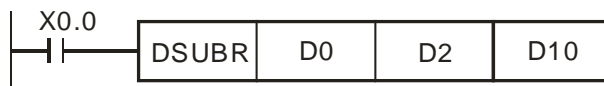
程序范例 (一)

- ◆ 当 X0.0=ON 时 , 将 F1.23456 浮点数值 , 减去 F1.0 浮点数值 , 其运算结果为 F0.23456 存放至 D10 , D11 数据寄存器内。



程序范例 (二)

- ◆ 当 X0.0=ON 时 , 将浮点数值 (D1 , D0) - 浮点数值 (D3 , D2) , 结果存放在 (D11 , D10) 中。



API	指令码			操作数							功能		
174	D	MULR	P	$S_1 \cdot S_2 \cdot D$							浮点数值乘算		

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 浮点数值被乘数。 S_2 : 浮点数值乘数。 D : 积。
- ◆ S_1 、 S_2 操作数可输入浮点数值。
- ◆ DMULR 指令可直接在 S_1 、 S_2 操作数输入浮点数值 (例如 : F1.2) , 或以寄存器 D 存放浮点数值。
- ◆ 当 S_1 、 S_2 操作数 , 以寄存器 D 存放浮点数值 , 其功能与 API 122 EMUL 相同。
- ◆ 当 DMULR 指令执行时 , D 操作数将会存放浮点数值运算后的结果。
- ◆ S_1 、 S_2 可指定相同的寄存器编号 , 此种情况下若是使用连续执行型态的指令时 , 在条件接点 ON 的期间 , 该寄存器于每一次扫描时 , 均会被乘算一次 , 一般的情况下都是使用脉冲执行型指令 (DMULRP) 。
- ◆ 若运算结果的绝对值大于可表示的最大浮点数值 , 则进位标志=ON。若运算结果的绝对值小于可表示的最大浮点数值 , 则借位标志=ON。若运算结果为 0 , 则零标志=ON。
- ◆ 零点标志 , OX 是 SM1064 , O100 是 SM968。
- ◆ 借位标志 , OX 是 SM1065 , O100 是 SM969。
- ◆ 进位标志 , OX 是 SM1066 , O100 是 SM970。

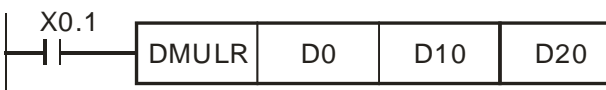
程序范例 (一)

- ◆ 当 X0.0=ON 时 , 将 F1.23456 浮点数值 , 乘以 F1.0 浮点数值 , 其运算结果为 F1.23456 存放至 (D10 , D11) 数据寄存器内。



程序范例 (二)

- ◆ 当 X0.1=ON 时 , 将浮点数值 (D1 , D0) 乘以浮点数值 (D11 , D10) 将积存放至 (D21 , D20) 数据寄存器内。



API	指令码			操作数								功能			
175	D	DIVR	P	$S_1 \cdot S_2 \cdot D$								浮点数值除算			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1							○			●	●				
S_2							○			●	●				
D										●	●				

脉冲执行型	16 位指令	32 位指令 (9 steps)
✓	-	✓

指令说明

- ◆ S_1 : 浮点数值被除数。 S_2 : 浮点数值除乘数。 D : 商。
- ◆ S_1 、 S_2 操作数可输入浮点数值。
- ◆ DDIVR 指令可直接在 S_1 、 S_2 操作数输入浮点数值 (例如 : F1.2)，或以寄存器 D 存放浮点数值。
- ◆ 当 S_1 、 S_2 操作数，以寄存器 D 存放浮点数值，其功能与 API 123 DEDIV 相同。
- ◆ 当 DDIVR 指令执行时，D 操作数将会存放浮点数值运算后的结果。
- ◆ S_1 、 S_2 可指定相同的寄存器编号，此种情况下若是使用连续执行型态的指令时，在条件接点 ON 的期间，该寄存器于每一次扫描时，均会被除算一次，一般的情况下都是使用脉冲执行型指令 (DDIVRP)。
- ◆ 若运算结果的绝对值大于可表示的最大浮点数值，则进位标志=ON。若运算结果的绝对值小于可表示的最大浮点数值，则借位标志=ON。若运算结果为 0，则零标志=ON。
- ◆ 零点标志，OX 是 SM1064，O100 是 SM968。
- ◆ 借位标志，OX 是 SM1065，O100 是 SM969。
- ◆ 进位标志，OX 是 SM1066，O100 是 SM970。

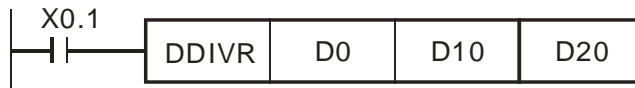
程序范例 (一)

- ◆ 当 X0.0=ON 时，将 F1.23456 浮点数值，除以 F1.0 浮点数值，其运算结果为 F1.23456 存放至 (D10，D11) 数据寄存器内。



程序范例 (二)

- ◆ 当 X0.1=ON 时，将浮点数值 (D1，D0) 除以浮点数值 (D11，D10) 将商存放至 (D21，D20) 数据寄存器内。



API	指令码			操作数								功能			
215~217	D	LD#		$S_1 \cdot S_2$								接点型态逻辑运算 LD #			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (7 steps)
-	✓	✓

指令说明

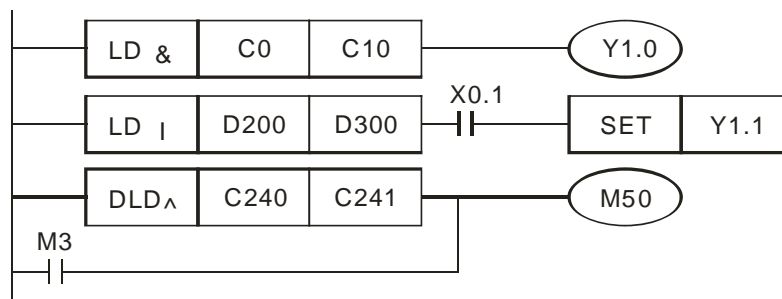
- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。
- ◆ S_1 与 S_2 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- ◆ LD # 的指令可直接与母线连接使用

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
215	LD&	DLD&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
216	LD	DLD	$S_1 S_2 \neq 0$	$S_1 S_2 = 0$
217	LD^	DLD^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- ◆ & : 逻辑的'及' (AND) 运算。
- ◆ | : 逻辑的'或' (OR) 运算。
- ◆ ^ : 逻辑的'异或' (XOR) 运算。
- ◆ 32 位计数器 (C200~C255) 以本指令作运算时，一定要使用 32 位指令 (DLD #)，若是使用 16 位指令 (LD #) 时，判定为"程序错误"，主机面板上 ERROR 指示灯闪烁。

程序范例

- ◆ C0 与 C10 的内容做逻辑的'及' (AND) 运算不等于 0 时，Y1.0=ON。
- ◆ D200 与 D300 的内容做逻辑的'或' (OR) 运算不等于 0 时，而且 X0.1=ON 的时候，Y1.1=ON 并保持住。
- ◆ C240 与 C241 的内容做逻辑的'异或' (XOR) 运算不等于 0 时或是 M3=ON 的时候，M50=ON。



5

API	指令码			操作数										功能	
218~220	D	AND#		$S_1 \cdot S_2$										接点型态逻辑运算 AND #	
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (7 steps)
-	✓	✓

指令说明

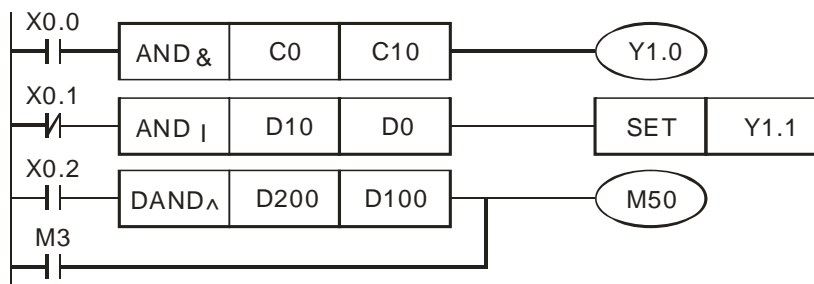
- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。
- ◆ S_1 与 S_2 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- ◆ AND # 的指令是与接点串接的运算指令。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
218	AND&	DAND&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
219	AND	DAND	$S_1 S_2 \neq 0$	$S_1 S_2 = 0$
220	AND^	DAND^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- ◆ & : 逻辑的'及' (AND) 运算。
- ◆ | : 逻辑的'或' (OR) 运算。
- ◆ ^ : 逻辑的'异或' (XOR) 运算。
- ◆ 32 位计数器 (C200~C255) 以本指令作运算时，一定要使用 32 位指令 (DAND #)，若是使用 16 位指令 (AND #) 时，判定为“程序错误”，主机面板上 ERROR 指示灯闪烁。

程序范例

- ◆ 当 X0.0=ON 时且 C0 与 C10 的内容做逻辑的'及' (AND) 运算不等于 0 时，Y1.0=ON。
- ◆ 当 X0.1=OFF 时且 D10 与 D0 的内容做逻辑的'或' (OR) 运算不等于 0 时，Y1.1=ON 并保持住。
- ◆ 当 X0.2=ON 时且 32 位寄存器 D200(D201)与 32 位寄存器 D100(D101) 的内容做逻辑的'异或' (XOR) 运算不等于 0 时或是 M3=ON 的时候，M50=ON。



API	指令码			操作数								功能			
221~223	D	OR#		$S_1 \cdot S_2$								接点型态逻辑运算 OR #			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (7 steps)
-	✓	✓

指令说明

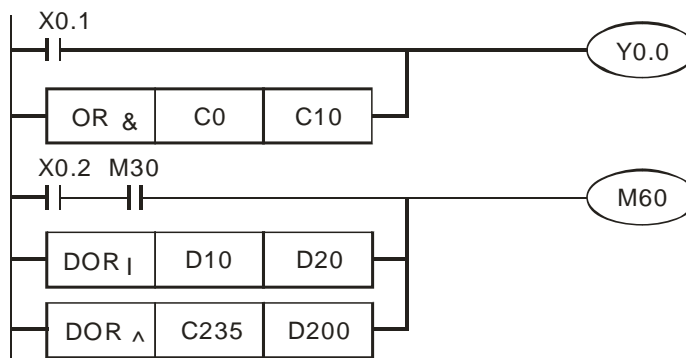
- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。
- ◆ S_1 与 S_2 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- ◆ OR # 的指令是与接点并接的运算指令。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
221	OR&	DOR&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
222	OR	DOR	$S_1 S_2 \neq 0$	$S_1 S_2 = 0$
223	OR^	DOR^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- ◆ & : 逻辑的'及' (AND) 运算。
- ◆ | : 逻辑的'或' (OR) 运算。
- ◆ ^ : 逻辑的'异或' (XOR) 运算。
- ◆ 32 位计数器 (C200~C255) 以本指令作运算时，一定要使用 32 位指令 (DOR #)，若是使用 16 位指令 (OR #) 时，判定为“程序错误”，主机面板上 ERROR 指示灯闪烁。

程序范例

- ◆ 当 X0.1=ON 时或 C0 与 C10 的内容做逻辑的'及' (AND) 运算不等于 0 时，Y0.0=ON。
- ◆ 当 X0.2 及 M30 都等于 ON 的时候，或者是 32-bit 寄存器 D10 (D11) 与 32 位寄存器 D20 (D21) 的内容做逻辑的'或' (OR) 运算不等于 0 时，或者是 32 位计数器 C235 与 32 位寄存器 D200 (D201) 的内容做逻辑的'异或' (XOR) 运算不等于 0 时，M60=ON。



API	指令码			操作数								功能			
224~230	D	LD※		$S_1 \cdot S_2$								接点型态比较 LD※			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
脉冲执行型								16 位指令 (5 steps)				32 位指令 (7 steps)			
-								✓				✓			

指令说明

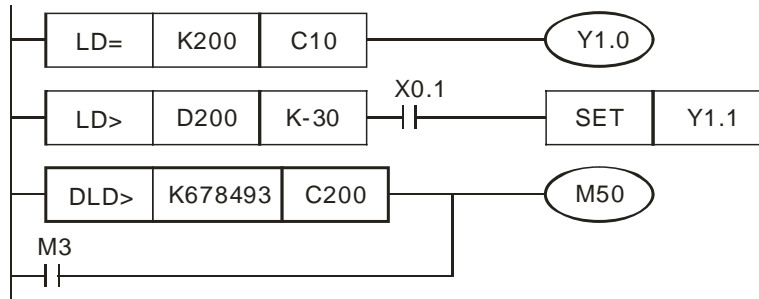
- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。
- ◆ S_1 与 S_2 之内容作比较的指令，以 API 224 LD=为例，比较结果为“等于”时，该指令导通，“不等于”时，该指令不导通。
- ◆ LD※的指令可直接与母线连接使用

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
224	LD =	DLD =	$S_1 = S_2$	$S_1 \neq S_2$
225	LD >	DLD >	$S_1 > S_2$	$S_1 \leq S_2$
226	LD <	DLD <	$S_1 < S_2$	$S_1 \geq S_2$
228	LD < >	DLD < >	$S_1 \neq S_2$	$S_1 = S_2$
229	LD < =	DLD < =	$S_1 \leq S_2$	$S_1 > S_2$
230	LD > =	DLD > =	$S_1 \geq S_2$	$S_1 < S_2$

- ◆ 32 位计数器 (C200~C255) 以本指令作比较时，一定要使用 32 位指令 (DLD※)，若是使用 16 位指令 (LD※) 时，判定为“程序错误”，主机面板上 ERROR 指示灯闪烁。

程序范例

- ◆ C10 的内容等于 K200 时，Y1.0=ON。
- ◆ 当 D200 的内容大于 K-30，而且 X0.1=ON 的时候，Y1.1=ON 并保持住。
- ◆ C200 的内容小于 K678,493 或者是 M3=ON 的时候，M50=ON。



API	指令码			操作数										功能	
232~238	D	AND※		$S_1 \cdot S_2$										接点型态比较 AND※	
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
脉冲执行型								16 位指令 (5 steps)				32 位指令 (7 steps)			
-								✓				✓			

指令说明

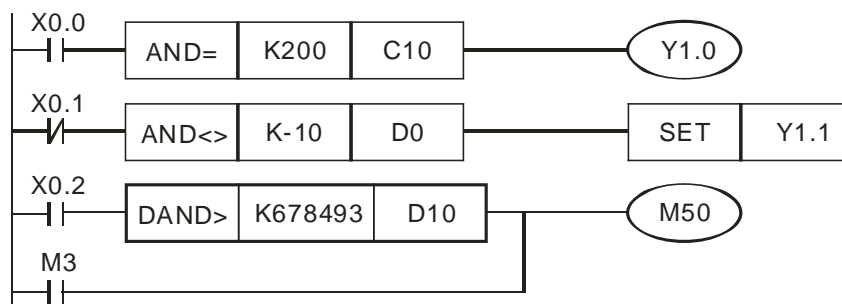
- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。
- ◆ S_1 与 S_2 之内容作比较的指令，以 API 232 AND= 为例，比较结果为等于时，该指令导通，不等于时，该指令不导通。
- ◆ AND※的指令是与接点串接的比较指令。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
232	AND =	DAND =	$S_1 = S_2$	$S_1 \neq S_2$
233	AND >	DAND >	$S_1 > S_2$	$S_1 \leq S_2$
234	AND <	DAND <	$S_1 < S_2$	$S_1 \geq S_2$
236	AND < >	DAND < >	$S_1 \neq S_2$	$S_1 = S_2$
237	AND < =	DAND < =	$S_1 \leq S_2$	$S_1 > S_2$
238	AND > =	DAND > =	$S_1 \geq S_2$	$S_1 < S_2$

- ◆ 32 位计数器 (C200~C255) 以本指令作比较时，一定要使用 32 位指令 (DAND※)，若是使用 16 位指令 (AND※) 时，判定为“程序错误”，主机面板上 ERROR 指示灯闪烁。

程序范例

- ◆ 当 X0.0=ON 时且 C10 的当前值又等于 K200 时，Y1.0=ON。
- ◆ 当 X0.1=OFF 而寄存器 D0 的内容又不等于 K-10 的时候，Y1.1=ON 并保持住。
- ◆ 当 X0.2=ON 而且 32 位寄存器 D0 (D11) 的内容又小于 678,493 的时候或 M3=ON 时，M50=ON。



API	指令码			操作数								功能			
240~246	D	OR※		$S_1 \cdot S_2$								接点型态比较 OR※			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○

脉冲执行型	16 位指令 (5 steps)	32 位指令 (7 steps)
-	✓	✓

指令说明

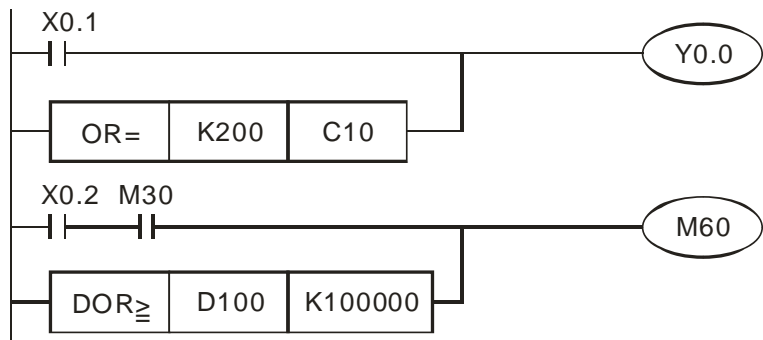
- ◆ S_1 : 数据来源装置 1。 S_2 : 数据来源装置 2。
- ◆ S_1 与 S_2 之内容作比较的指令。以 API 240 OR= 为例。比较结果为等于时，该指令导通，不等于时，该指令不导通。
- ◆ OR※的指令是与接点并接的比较指令。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
240	OR =	DOR =	$S_1 = S_2$	$S_1 \neq S_2$
241	OR >	DOR >	$S_1 > S_2$	$S_1 \leq S_2$
242	OR <	DOR <	$S_1 < S_2$	$S_1 \geq S_2$
244	OR < >	DOR < >	$S_1 \neq S_2$	$S_1 = S_2$
245	OR < =	DOR < =	$S_1 \leq S_2$	$S_1 > S_2$
246	OR > =	DOR > =	$S_1 \geq S_2$	$S_1 < S_2$

- ◆ 32 位计数器 (C200~C255) 以本指令作比较时，一定要使用 32 位指令 (DOR※)。若是使用 16 位指令 (OR※) 时，判定为“程序错误”，主机面板上 ERROR 指示灯闪烁。

程序范例

- ◆ 当 X0.1=ON 时，或者是 C10 的当前值等于 K200 时，Y0.0=ON。
- ◆ 当 X0.2 及 M30 都等于 ON 的时候，或者是 32 位寄存器 D100 (D101) 的内容大于或等于 K100000 时，M60=ON。



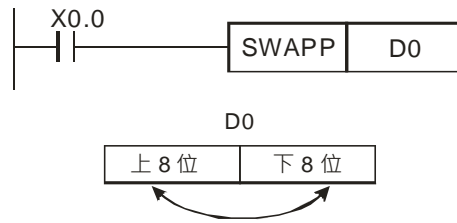
API	指令码			操作数										功能	
152	D	SWAP	P	S										上下 8 位互换	
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S								●	●	●	●	●	●	○	○
脉冲执行型		16 位指令 (3 steps)					32 位指令 (3 steps)								
✓		✓					✓								

指令说明

- ◆ S : 欲执行上下位 8 位互相交换之装置。
- ◆ 16 位指令时，上位 8 位与下位 8 位的内容互相交换。
- ◆ 32 位指令时，两个寄存器的上位 8 位与下位 8 位的内容互相交换。
- ◆ 本指令一般使用脉冲执行型指令 (SWAPP · DSWAPP)。

程序范例 (一)

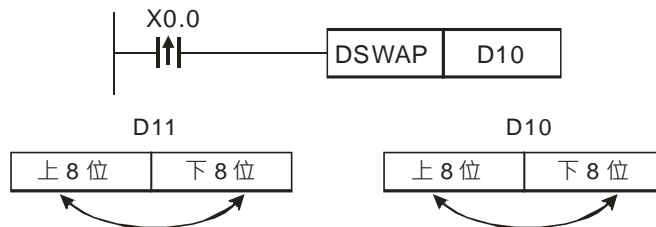
- ◆ 当 X0.0=ON 时，将 D0 的上 8 位与下位 8 位的内容互相交换。



5

程序范例 (二)

- ◆ 当 X0.0=ON 时，将 D11 的上位 8 位与下位 8 位的内容互相交换，D10 的上位 8 位与下位 8 位的内容互相交换。



API	指令码			操作数								功能			
154	D	RAND	P	$S_1 \cdot S_2 \cdot D$								随机数值			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○		●	●	●	●	●	●	○	○
S_2					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○

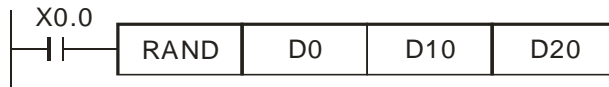
脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 随机数产生的范围下限。 S_2 : 随机数产生的范围上限。 D : 随机数产生的结果。
- ◆ 16 位操作数 S_1 、 S_2 使用范围： $K0 \leq S_1, S_2 \leq K32,767$ ，32 位操作数 S_1 、 S_2 使用范围： $K0 \leq S_1, S_2 \leq K2,147,483,647$ 。
- ◆ 操作数 $S_1 \leq$ 操作数 S_2 ，若使用者输入 $S_1 > S_2$ ，PM 判断为运算错误。
- ◆ 指定 KnM (S) 等装置的位装置时，以 10 进制中 16 的倍数为起始编号。如：K4M0、K4S16 (十进制)。

程序范例

- ◆ 当 X0.0=ON，RAND 指令产生介于范围下限 D0 与范围上限 D10 的随机数，将结果存放到 D20 内。



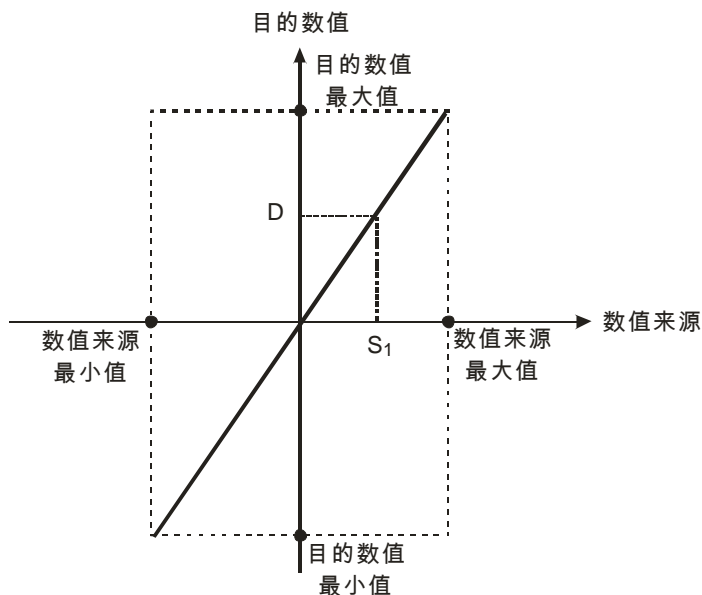
API	指令码			操作数								功能			
202		SCAL	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$								比例值运算			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○				●	●				
S_2					○	○				●	●				
S_3					○	○				●	●				
D										●	●				

脉冲执行型	16 位指令 (7 steps)	32 位指令
✓	✓	-

指令说明

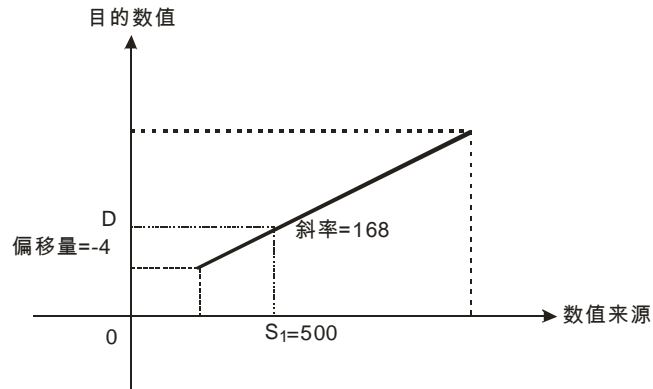
- ◆ S_1 : 来源数值数据。 S_2 : 斜率，单位为 0.001。 S_3 : 偏移量。 D : 目的地装置。
- ◆ 操作数范围 $S_1 \cdot S_2 \cdot S_3$ 是 -32767~32767
- ◆ 指令内部运算公式为： $D = (S_1 \times S_2) \div 1000 + S_3$
- ◆ S_2 和 S_3 的数值须由使用者依下列斜率与偏移量公式先行运行，然后将小数点 4 舍 5 入后，再取 16 位的整数数值输入。
- ◆ 斜率公式为： $S_2 = [(\text{目的数值最大值} - \text{目的数值最小值}) \div (\text{来源数值最大值} - \text{来源数值最小值})] \times 1000$
- ◆ 偏移量公式为： $S_3 = \text{目的数值最小值} - \text{来源数值最小值} \times S_2 \div 1000$
- ◆ 输出曲线如下图所示：



5

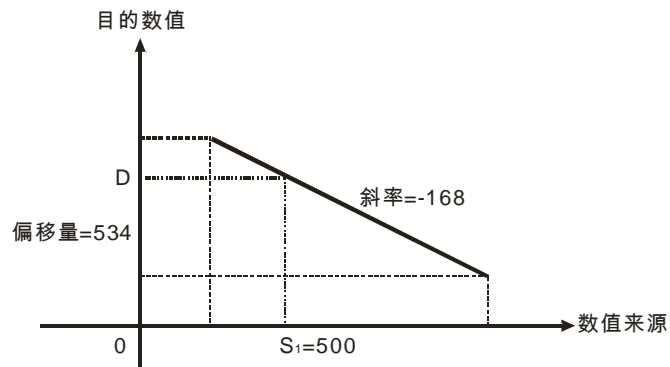
程序范例
(一)

- ◆ 已知 S_1 数值数据来源为 500 S_2 斜率为 168 S_3 偏移量为 -4。当 X0.0=ON 时，SCAL 指令执行，可在 D0 得到所要求的比例值。
- ◆ 运算方式： $D0 = (500 \times 168) \div 1000 + (-4) = 80$



程序范例
(二)

- ◆ 已知 S_1 数值数据来源为 500 S_2 斜率为 -168 S_3 偏移量 534。当 X1.0=ON 时，SCAL 指令执行，可在 D10 得到所要求的比例值。
- ◆ 运算方式： $D10 = (500 \times -168) \div 1000 + 534 = 450$



补充说明

- ◆ 此 SCAL 指令示用于已知斜率与偏移量，若不知斜率与偏移量建议使用 SCLP 指令来做运算。
- ◆ 输入参数 S_2 时，其输入数值必须为 -32,768~32,767 之间的数值(实际数值为 -32,768~32,767)。若是 S_2 实际数值超过范围时，请改用 SCLP 指令运算。
- ◆ 使用者运用斜率换算公式时，须注意来源数值最大值，必须大于来源数值最小值，而目的数值最大值，并不限制大于目的数值最小值。
- ◆ 若 $D > 32,767$ ，则 $D = 32,767$ 。若 $D < -32,768$ ，则 $D = -32,768$ 。

API	指令码			操作数								功能			
203	D	SCLP	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$								参数型比例运算			

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S_1					○	○				●	●				
S_2					○	○				●	●				
D										●	●				

脉冲执行型	16 位指令 (7 steps)	32 位指令 (9 steps)
✓	✓	✓

指令说明

- ◆ S_1 : 数值数据来源。 S_2 : 参数，单位为 0.001。 D : 目的地装置。
- ◆ 16 位指令 S_2 参数设定内容如下：

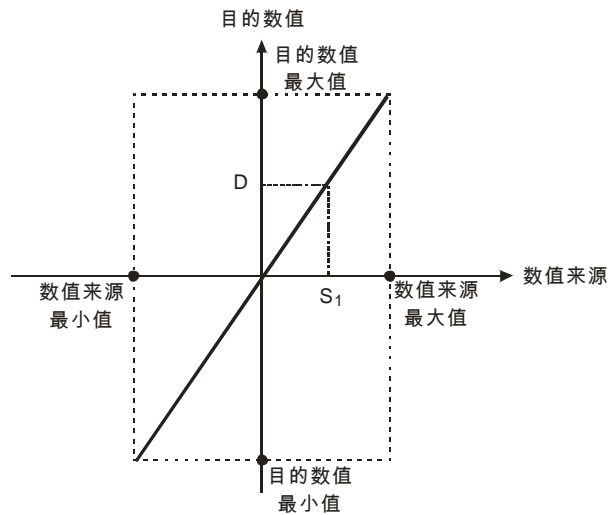
装置编号	参数名称与说明	设定范围
S_2	来源数值最大值	-32768~32767
S_{2+1}	来源数值最小值	-32768~32767
S_{2+2}	目的数值最大值	-32768~32767
S_{2+3}	目的数值最小值	-32768~32767

- ◆ 16 位指令 S_2 操作数将连续占用 4 个装置。
- ◆ 32 位指令 S_2 参数设定内容如下：

装置编号	参数名称与说明	设定范围	
		整数	浮点数
$S_2 \cdot S_{2+1}$	来源数值最大值	-2,147,483,648~ 2,147,483,647	32 位浮点数范围
$S_{2+2} \cdot 3$	来源数值最小值		
$S_{2+4} \cdot 5$	目的数值最大值		
$S_{2+6} \cdot 7$	目的数值最小值		

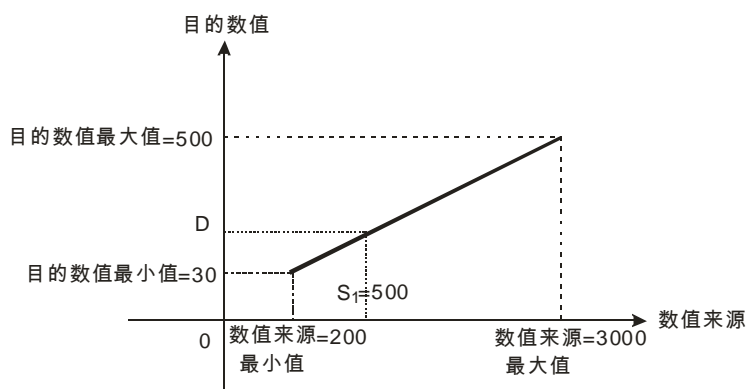
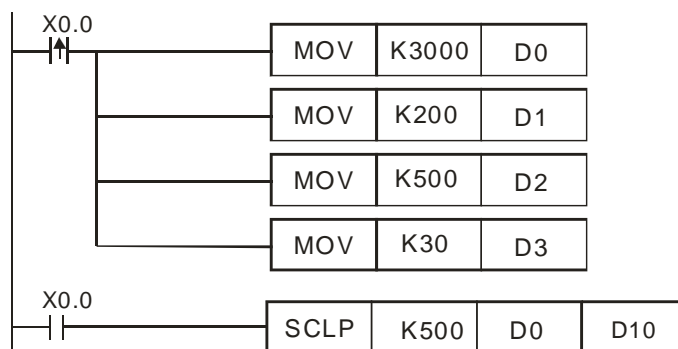
- ◆ 32 位指令 S_2 操作数将连续占用 8 个装置
- ◆ 标志信号：M1162 十进制整数与二进浮点数使用标志，ON 时表示二进浮点数。
- ◆ 指令内部运算公式为 $D=[(S_1-来源数值最小值) \times (目的数值最大值-目的数值最小值)] \div (来源数值最大值-来源数值最小值) + 目的数值最小值$

- ◆ 来源数值和目的数值运算关系：
 $y=kx+b$ · y =目的数值 (D) · k =斜率= $(\text{目的数值最大值}-\text{目的数值最小值}) \div (\text{来源数值最大值}-\text{来源数值最小值})$ · x =来源数值 (S_1) · b =偏移量= $\text{目的数值最小值}-\text{来源数值最小值} \times \text{斜率}$
- ◆ 将上面的各参数带入公式 $y=kx+b$ · 即可推导出指令内部运算公式：
 $y=kx+b=D=kS_1+b=\text{斜率} \times S_1 + \text{偏移量} = \text{斜率} \times S_1 + \text{目的数值最小值}-\text{来源数值最小值} \times \text{斜率} = \text{斜率} \times (S_1-\text{来源数值最小值}) + \text{目的数值最小值} = (\text{斜率} \times (S_1-\text{来源数值最小值}) \div (\text{来源数值最大值}-\text{来源数值最小值})) \times (\text{目的数值最大值}-\text{目的数值最小值}) + \text{目的数值最小值}$
- ◆ 假如 $S_1 > \text{来源数值最大值}$ · $S_1 = \text{来源数值最大值}$ · 假如 $S_1 < \text{来源数值最小值}$ · $S_1 = \text{来源数值最小值}$ · 当输入数值与参数设定完成后 · 则其输出曲线将如下图所示：



程序范例
(一)

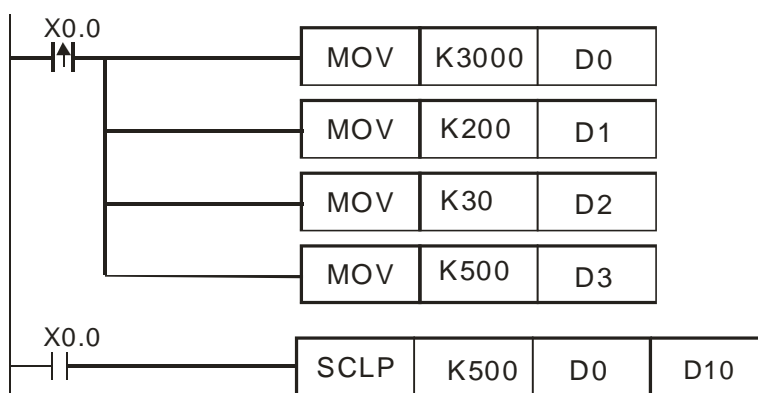
- ◆ 已知 S_1 数值数据来源为 500，来源数值最大值 $D0=3000$ ，来源数值最小值 $D1=200$ ，目的数值最大值 $D2=500$ ，目的数值最小值 $D3=30$ 。当 $X0.0=ON$ 时，SCLP 指令执行，可在 $D10$ 得到所要求的比例值。
- ◆ 运算方式： $D10=[(500-200) \times (500-30)] \div (3000-200) + 30 = 80.35$ 。
取整数， $D10=80$

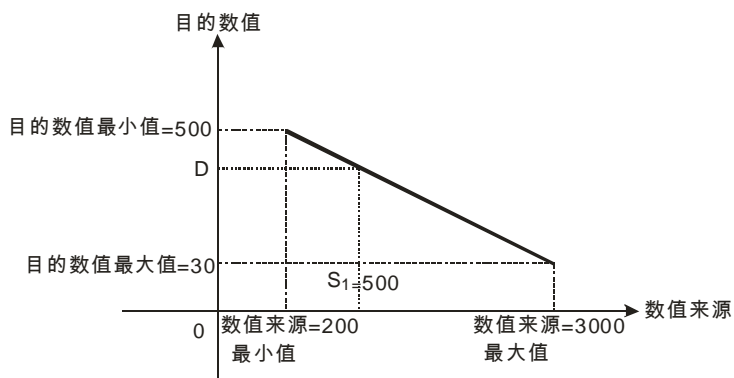


5

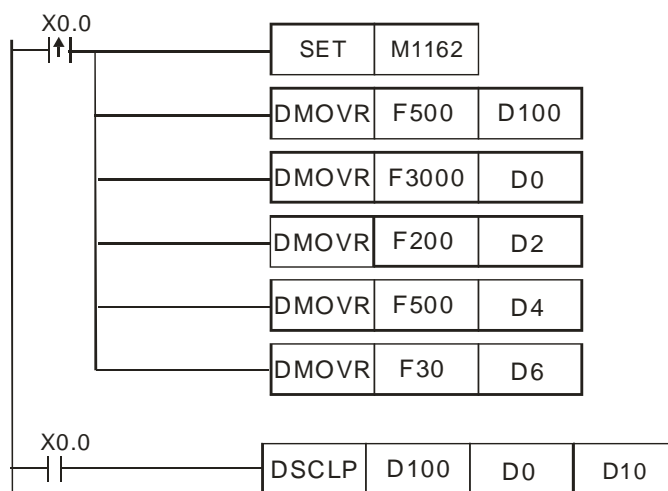
程序范例
(二)

- ◆ 已知 S_1 数值数据来源为 500，来源数值最大值 $D0=3000$ ，来源数值最小值 $D1=200$ ，目的数值最大值 $D2=30$ ，目的数值最小值 $D3=500$ 。当 $X0.0=ON$ ，SCLP 指令执行，可在 $D10$ 得到所要求的比例值。
- ◆ 运算方式： $D10=[(500-200) \times (30-500)] \div (3000-200) + 500 = 449.64$ 。
四舍五入取整数， $D10=450$





- ◆ 已知 S_1 数值数据来源 $D100=F500$ ，来源数值最大值 $D0=F3000$ ，来源数值最小值 $D2=F200$ ，目的数值最大值 $D4=F500$ ，目的数值最小值 $D6=F30$ 。当 $X0.0 = ON$ 时，SET M1162，使用浮点数运算且 DSCLP 指令执行。可在 D10 得到所要求比例值。
- ◆ 运算方式： $D10 = [(F500 - F200) \times (F500 - F30)] \div (F3000 - F200) + F30 = F80.35$ 。取整数， $D10 = F80$ 。



补充说明

- ◆ 16 位 S_1 操作数数值设定范围：来源数值最大值 $\geq S_1 \geq$ 来源数值最小值，-32768~32767。如果超出边界值以边界值运算。
- ◆ 32 位 S_1 整数操作数数值设定范围：来源数值最大值 $\geq S_1 \geq$ 来源数值最小值，-2,147,483,648~2,147,483,647。如果超出边界值以边界值运算。
- ◆ 32 位 S_1 浮点数操作数数值设定范围：来源数值最大值 $\geq S_1 \geq$ 来源数值最小值，依 32 位浮点数范围。如果超出边界值以边界值运算。
- ◆ 使用者运用时，须注意来源数值最大值，必须大于来源数值最小值，而目的数值最大值，并不限制大于目的数值最小值。

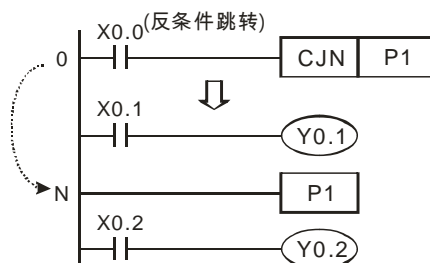
API	指令码			操作数							功能				
256		CJN	P	S							反条件跳转				
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
P															
					脉冲执行型			16 位指令 (3 steps)			32 位指令				
					✓			✓			-				

指令说明

- ◆ **S**：反条件跳转之目的指针，操作数指定 **P**，操作数可指定 **P0~P255**，但 **P** 装置不支持 **V**、**Z** 修饰。
- ◆ 当 **CJN** 指令前的条件接点为导通状态时，则至下一行程序动作；反之，当条件接点为不导通状态时，则跳转至所指定的 **P** 指针处动作。
- ◆ 当使用者希望 **O100** 程序中的某一部份不需要执行时，以缩短扫描时间，以及使用于双重输出时，可使用 **CJN** 或 **CJNP** 指令。
- ◆ 指针 **P** 所指之程序若在 **CJN** 指令之前，需注意会发生 **WDT** 逾时之错误，**O100** 停止运转，请注意使用。
- ◆ **CJN** 指令可重复指定同一指针 **P**，但 **CJN** 与 **CALL** 不可指定同一指针 **P**，否则会产生错误。
- ◆ 跳转执行中各种装置动作情形说明：
 - ◆ **Y**、**M**、**S** 保持跳转发生前之状态。
 - ◆ 执行计时中之 **10ms** 定时器会暂停计时。
 - ◆ 一般计数器会停止计数；而一般应用指令不会被执行。
 - ◆ 定时器之清除指令若在跳转前被驱动，则在跳转执行中该装置仍处于清除状态。

程序范例

- ◆ 当 **X0.0=OFF** 时，程序自动从地址 **0** 跳转至地址 **N**（即指定之标签 **P1**）继续执行，中间地址跳过不执行。
- ◆ 当 **X0.0=ON** 时，程序如同一般程序由地址 **0** 继续往下执行，此时 **CJN** 指令不被执行。



API	指令码			操作数								功能
257		JMP	P	S								无条件跳转

装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
P															

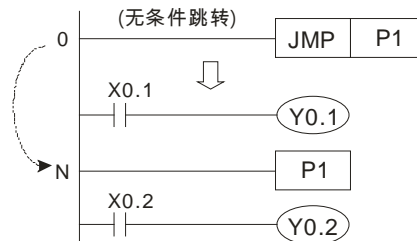
脉冲执行型	16 位指令 (3 steps)	32 位指令
	✓	-

指令说明

- ◆ JMP 指令功能与 CJ 指令雷同，不同之处只差在 CJ 指令前需要接点驱动，而 JMP 指令前是不须接点驱动的指令。
- ◆ JMP 指令不支持 JMPP 脉冲执行型。

程序范例

- ◆ 当程序扫描至第 0 行程序时，无论 JMP 指令前是否有无条件接点（也不考虑条件接点的状态的情况），程序皆从地址 0 跳转至地址 N（即指定之标签 P1）继续执行，中间地址跳过不执行。



API	指令码		操作数	功能
258		BRET	-	回母线

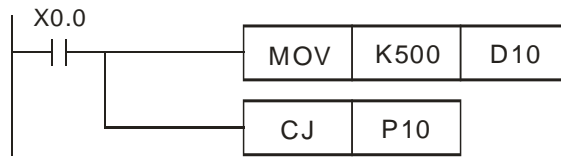
脉冲执行型	16 位指令 (1 step)	32 位指令
-	✓	-

指令说明

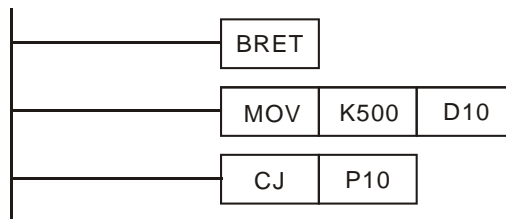
- ◆ BRET 指令无须接点驱动。
- ◆ 执行 BRET 指令之后，原本须条件接点驱动指令相当于已连接上母线，可以直接执行这些指令。

程序范例

- ◆ 一般程序，需要 X0.0 为 ON 时，才会执行条件接点后的指令。如下所示：



- ◆ 当加入 BRET 指令之后，原本须条件接点驱动指令，就像已经连接上母线，而可直接执行动作。如下所示：



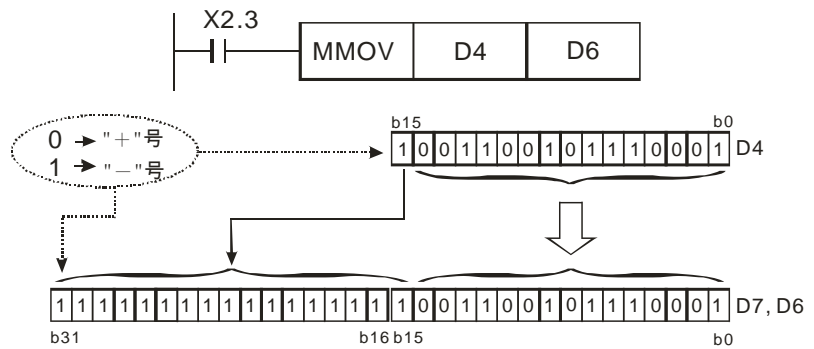
API	指令码				操作数								功能		
259		MMOV	P		S · D								16→32 位数值转换		
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z
S					○	○		●	●	●	●	●	●	○	○
D								●	●	●	●	●	●	○	○
脉冲执行型								16 位指令 (1 step)				32 位指令			
✓								✓				-			

指令说明

- ◆ S : 数据之来源 (16 位) 。 D : 数据之搬移目的地 (32 位) 。
- ◆ 将 16 位装置 S 中的数据传送到 32 位的装置 D 中 , 其中指定的符号位被重复的复制存放在目的地 。

程序范例

- ◆ 当 X2.3 为 0N 时 , D4 的数据传送到 D6 和 D7 。



在以上的例子中 , D4 的 b15 位数据传送到 (D7/D6) 的 b15 到 b31 位 , 变成负数 (和 D4 的一样) 。

API	指令码				操作数								功能			
260		RMOV	P		S · D								32→16 位数值转换			
装置	Xn.n	Yn.n	M	S	K	16#	F	KnM	KnS	D	W	T	C	V	Z	
S					○	○		●	●	●	●	●	●	○	○	
D								●	●	●	●	●	●	○	○	

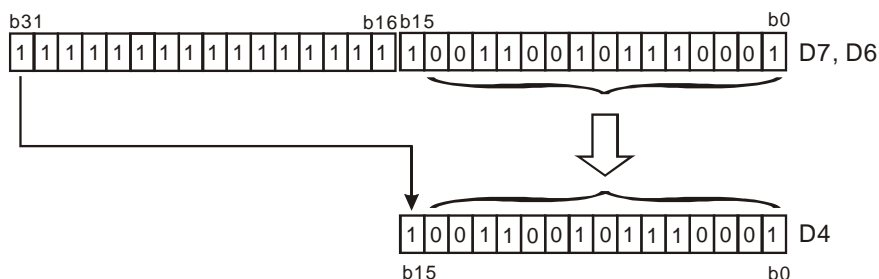
脉冲执行型	16 位指令 (6 steps)	32 位指令
✓	✓	-

指令说明

- ◆ S : 数据之来源 (32 位) 。 D : 数据之搬移目的地 (16 位) 。
- ◆ 将 32 位装置 S 中的数据传送到 16 位的装置 D 中 , 其中符号位被保留 。

程序范例

- ◆ 当 X2.4 为 0N 时 , D6 和 D7 的数据传送到 D4 。



当 X2.4 为 0N 时 , (S) 中最高位 (D7 : b31) 被传送到 (D) 中最高位 (D4 : b15) 中 , 其它位 (b0~b14) 则对应传送 , 而 b15~b30 被忽略未被传送 。

5.7 运动功能块一览表

分类	功能块名称	功能叙述	支持机种			页码
			20MC	10PM/ 15PM	05PM	
单轴运动功能	绝对单段速运动	轴运行至指定的绝对位置	✓	✓	✓	5-139
	相对单段速运动	轴运行至与当前的相对位置	✓	✓	✓	5-142
	绝对两段速运动	轴以两段速度运行至指定的绝对位置	✓	✓	✓	5-145
	相对两段速运动	轴以两段速度运行至与当前的绝对位置	✓	✓	✓	5-148
	插入单段速运动	轴以外部触发点当单段速运动起始点	✓	✓	✓	5-150
	插入两段速运动	轴以外部触发点当两段速运动起始点	✓	✓	✓	5-154
	寸动运动	轴以指定速度连续运动	✓	✓	✓	5-156
	手摇轮模式运动	轴跟随手轮运动	✓	✓	✓	5-159
	原点回归运动	轴运行原点回归动作	-	✓	✓	5-160
	单轴运动停止	轴运动停止	✓	✓	✓	5-162
	参数设定 1	轴参数设定 1	✓	✓	✓	5-164
	参数设定 2	轴参数设定 2	✓	✓	✓	5-165
	读取当前位置/速度	显示轴当前位置与速度	✓	✓	✓	5-167
	轴状态信息	显示轴目前的运动状态以及错误状态	✓	✓	✓	5-168
	设定当前位置	设定轴当前位置	✓	✓	✓	5-170
	输入极性设定	设定接点极性	✓	✓	✓	5-171
	电子齿轮运动	轴跟随电子齿轮功能运动	✓	✓	✓	5-174
	凸轮啮合	电子凸轮启动与脱离	✓	✓	✓	5-176
	凸轮数据点读取	电子凸轮曲线的数据点读取	✓	✓	✓	5-178
	凸轮数据点写入	电子凸轮曲线的数据点写入	✓	✓	✓	5-179
凸轮同步速率计算	计算飞剪同步区速率比例	✓	✓	✓	5-181	
凸轮曲线规划	生成飞剪凸轮曲线	✓	✓	✓	5-182	
凸轮曲线更新	动态更新飞剪凸轮曲线	✓	✓	✓	5-184	
多轴功能	G 码设定	设定运行 G-code 相关设定	✓	✓	✓	5-186
	G 码运行	开始运行指定 G-code	✓	✓	✓	5-187
	G 码停止	停止运行 G-code	✓	✓	✓	5-189
	M 码读取	对 M 码操作	✓	✓	✓	5-191
	多轴绝对直线插补	执行六轴绝对位置的同动插补	✓	✓	✓	5-193
	多轴相对直线插补	执行六轴相对位置的同动插补	✓	✓	✓	5-194
	插补停止	停止六轴同动插补功能	✓	✓	✓	5-196
网络功能	伺服启动/停止	启动/停止 DMC 伺服	✓	-	-	5-198
	伺服重置	重置 DMC 伺服	✓	-	-	5-199
	伺服参数写入	对 DMC 上的伺服作参数写入	✓	-	-	5-200
	伺服参数读取	对 DMC 上的伺服作参数读出	✓	-	-	5-202

分类	功能块名称	功能叙述	支持機種			页码
			20MC	10PM/ 15PM	05PM	
其它	伺服回原点命令	对 DMC 伺服下达原点回归动作	✓	-	-	5-203
	伺服 DMCNET 通讯初始化	对 DMC 伺服作通讯初始化动作	✓	-	-	5-205
	伺服捕捉设定	对 DMC 伺服作捕捉设定动作	✓	-	-	5-207
	Ethernet IP 设定	设定模块的 Ethernet IP Address	✓	✓	✓	5-209
	主程序备份	备份主程序到 SD 卡	✓	✓	✓	5-210
	装置备份	备份装置数据到 SD 卡	✓	✓	✓	5-211
	装置回存	将 SD 卡数据回存至模块中	✓	✓	✓	5-212
	高速计数器	高速计数器设定及启动	✓	✓	✓	5-213
	高速定时器	高速定时器设定及启动	✓	✓	✓	5-214
	高速比较设定	高速比较器功能设定	✓	✓	✓	5-216
	高速比较重置	重置高速比较器设定	✓	✓	✓	5-218
	高速捕捉设定	高速捕捉功能设定	✓	✓	✓	5-219
	高速捕捉遮蔽	遮蔽功能设定	✓	✓	✓	5-221
中断设定	设定中断程序功能	✓	✓	✓	5-223	
绝对型编码器	设定与启动绝对型编码器读取的功能	✓	✓	✓	5-224	

5.8 运动功能块引脚介绍

5.8.1 输入/输出引脚功能定义

下表为运动功能块的常见输入与输出引脚列表，单一运动功能块并非会有下表的所有引脚，例如以输入引脚来说，一个功能块只会有 **Execute** 或 **Enable** 其中之一。

输入引脚			
名称	说明	格式	设定值
Execute	启动功能块	BOOL	TRUE / FALSE
Enable	启动功能块	BOOL	TRUE / FALSE
输出引脚			
名称	说明	格式	设定值
Done	功能块动作完成	BOOL	功能块完成时
Valid	输出值有效	BOOL	Enable 上升沿时
Busy	功能块动作执行中	BOOL	Execute 上升沿时
Aborted	功能块被其它命令中断	BOOL	执行中被命令停止
Error	功能块产生错误	BOOL	执行中发生错误

在一个运动功能块中必定包含 **Execute** 引脚或是 **Enable** 引脚，目的在启动此运动功能块。而为了显示运动功能块的执行状态一般会包含 **Busy** 和 **Done** 两种引脚，如果此功能块会被其它运动功能块中断时，则会增加 **Aborted**，另外 **Error** 引脚主要目的在显示此运动功能块在启动过程中发生错误。

单一运动功能块的输入引脚除上列的 **Execute** 和 **Enable** 外，会有其它提供给使用者作运动数据输入的引脚，这些数据/状态引脚会有以下特性：

- 输入数据取用时机：
 - 功能块启动为 **Execute** 时：数据在 **Execute** 上升沿触发被取用，要让变更数据再次生效需在修改后输入参数后再次让 **Execute** 上升沿触发。
 - 功能块启动为 **Enable** 时：参数会在 **Enable** 上升沿触发被取用，但与搭配 **Execute** 使用相比会比较常被设计为执行中持续更新的形式。
- 输入数据超过范围：

运动功能块在伴随着超出允许范围的数据输入时被启动，会造成输入的数据被限制或运动功能块产生错误。若此时产生轴的错误结果，此乃为运动功能块的应用错误所导致，需使用者在应用程序中避免错误的输入产生。
- 输出状态的互斥性：
 - 功能块启动为 **Execute** 时：‘**Busy**’，‘**Done**’，‘**Aborted**’，‘**Error**’彼此互斥，在一个运动功能块中同时只能有一个为真，当 **Execute** 为真时，里面其中一个必须为真，且‘**Busy**’，‘**Done**’，‘**Aborted**’，‘**Error**’同时间只有一个会被设置。
 - 功能块启动为 **Enable** 时：输出‘**Valid**’ and ‘**Error**’彼此互斥，同时间只有一个能被设置。

- 输出数据/状态有效时机：
 - 功能块启动为 Execute 时：'Done'、'Error'、'Aborted'以及数据输出会在'Execute'下降沿时被重置，然而'Execute'下降沿不会停止甚至也不影响 FB 实际的执行。即使在运动功能块完成前 Execute 就被重置，相对应的输出状态仍会产生并保持一个周期，如果运动功能块在完成之前又接收到新的 execute，运动功能块将不会对之前动作的'Done'、'Aborted'有任何反馈，且可能会产生运动功能块错误。
 - 功能块启动为 Enable 时：'Valid'、'Busy'、'Error'输出会尽快跟着'Enable'下降沿被重置。
- Done 输出特性：

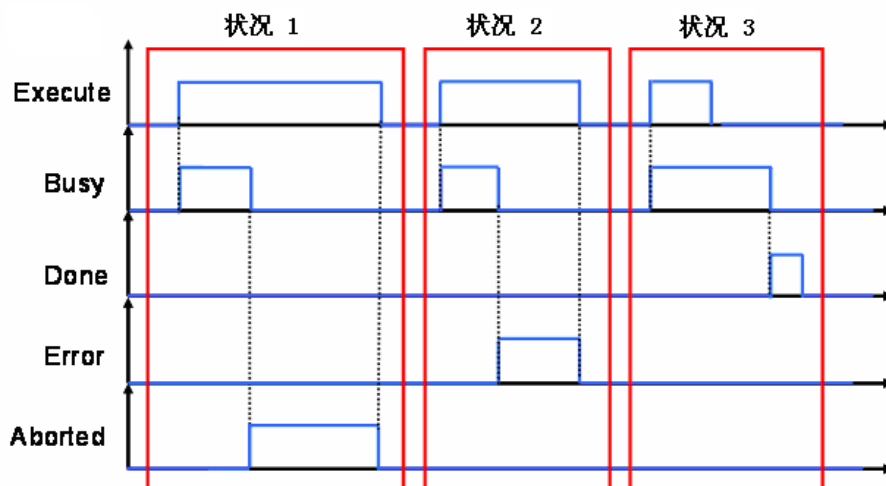
'Done' 输出会在被命令的运动成功完成时被设置
- Busy 输出特性：
 - 功能块启动为 Execute 时：每个运动功能块会有一个 Busy 输出用来反应运动功能块尚未完成，并且新的输出状态（值）可预期会产生。Busy 在'Execute'上升沿被设置，在 Done'、'Aborted'、'Error' 被设置时会被重置。
 - 功能块启动为 Enable 时：每个运动功能块会有一个 Busy 输出用来反应运动功能块尚未完成，并且新的输出状态（值）可预期会产生。Busy 在'Enable'上升沿被设置，且只要运动功能块还在执行动作就会保持住。同时，对应的输出仍会变化。
- Aborted 输出特性：

'Aborted'在运动尚未完成就被命令停止时会被设置。
- 输入 Enable 与输出 Valid 关系：

搭配Enable时若运动功能块有输出状态或数值的引脚就会有Valid输出引脚来表示这些输出否有效。Valid输出只在Enable为真及输出有效时为真，若运动功能块有错误会让输出不有效，Valid会为假，直到错误状况消失，输出值重新有效'Valid'才会再次被设置。

5

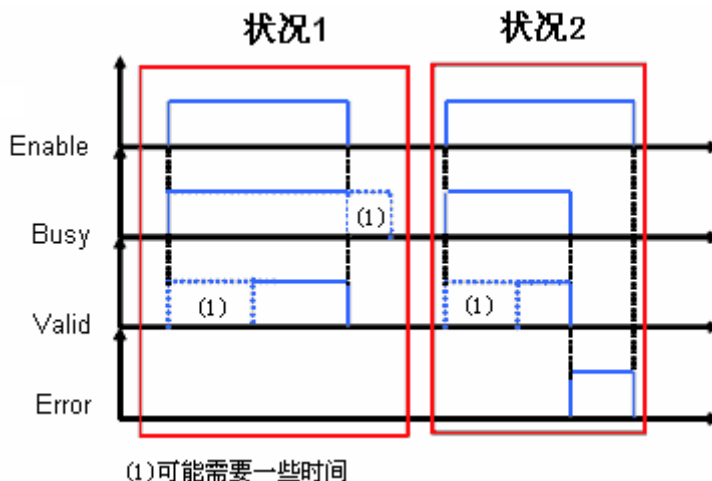
5.8.2 输入/输出引脚时序图



状况 1：运动功能块发生被强制中断时，输入 Execute 与输出 Busy、Done、Error、Aborted 时序图

状况 2：运动功能块发生被错误时，输入 Execute 与输出 Busy、Done、Error、Aborted 时序图

状况 3：运动功能块完成正常动作时，输入 Execute 与输出 Busy、Done、Error、Aborted 时序图

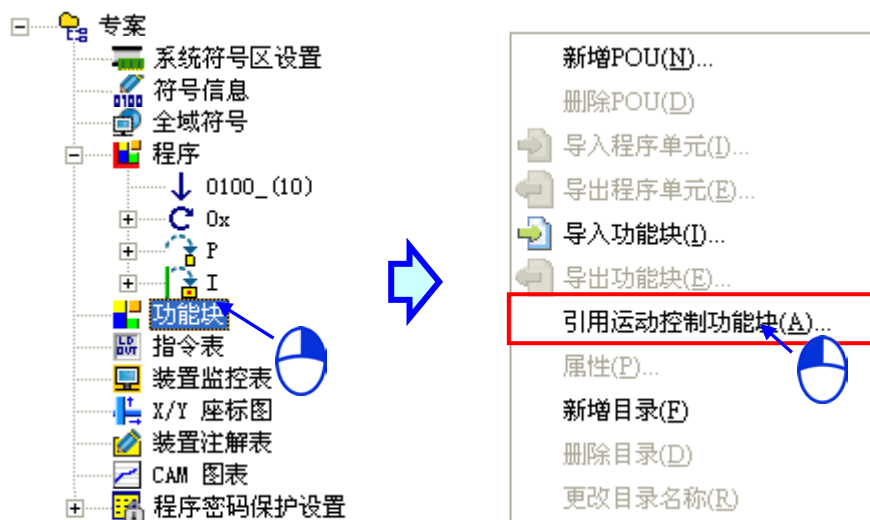


状况 1：运动功能块正常动作时，输入 Enable 与输出 Busy、Valid、Error 时序图
 状况 2：运动功能块发生被错误时，输入 Enable 与输出 Busy、Valid、Error 时序图

5.8.3 PMSoft软件操作介绍

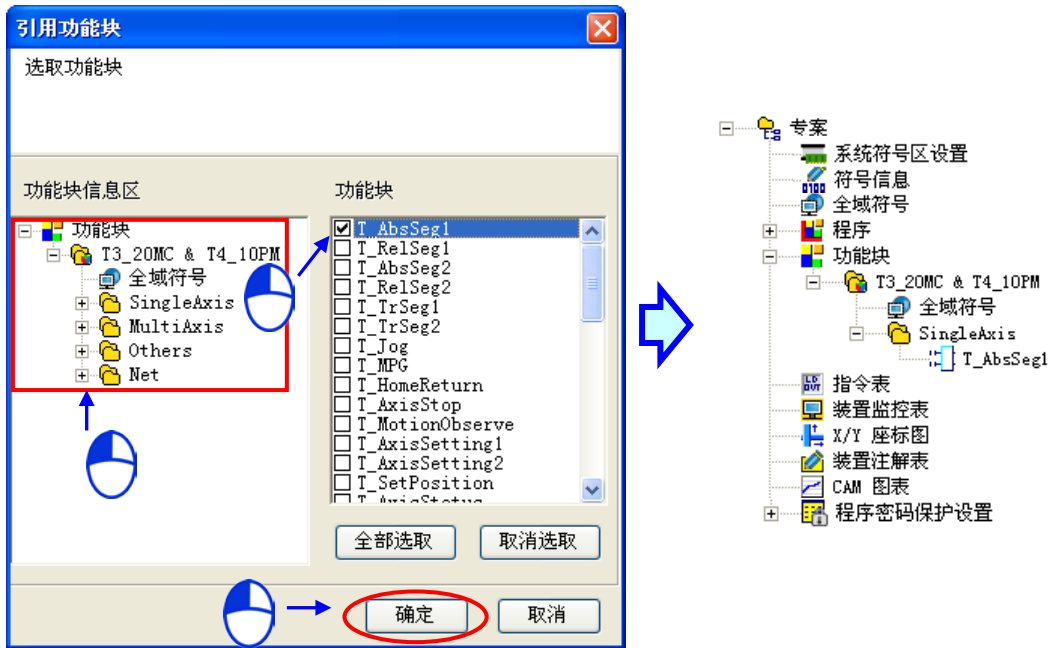
以下介绍台达运动功能块在 PMSoft 软件上面的操作方式。

(1) 使用 PMSoft 开启新文件后，在功能块图标上利用鼠标右键调出操作选单。



在选单中选择引用运动控制功能块。

(2) 单击引用功能块后，会跳出以下窗口



在窗口中可以个别选择要引用的功能块，或选择全部选取，会引用全部的功能块，在选择完成后单击确定，即可开始引用台达运动控制功能块。

5

(3) 在引用功能块页面单击确定，系统信息中功能块会自动增加台达运动控制功能块。

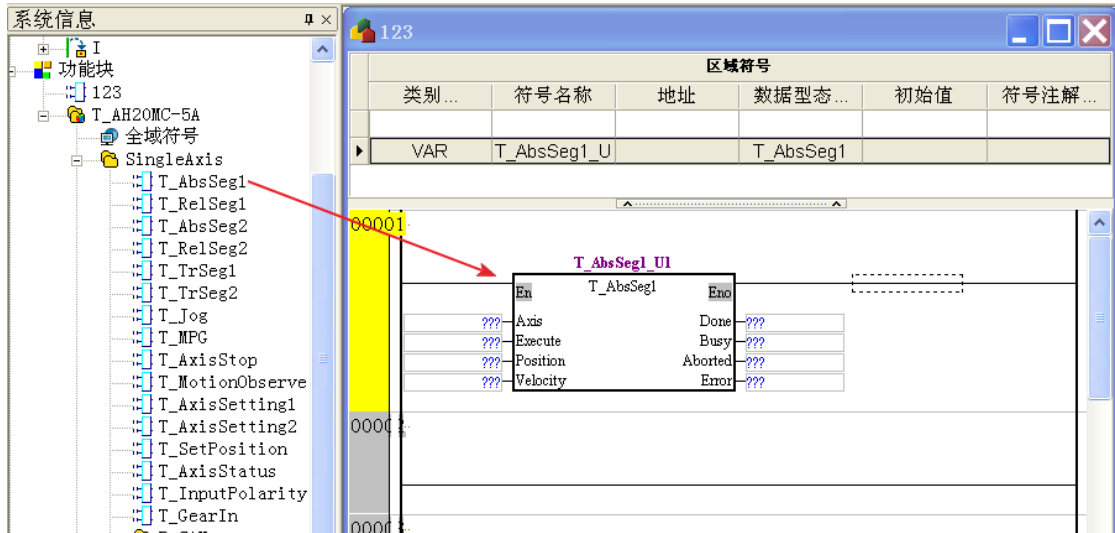


- 数据夹的架构如下：



- 各数据夹定义如下：
 - SingleAxis 为单轴运动，包含单轴点对点运动，电子齿轮跟随，电子凸轮跟随。
 - MultiAxis 为多轴运动，包含 G 码启动操作，多轴直线插补。
 - Net 为通讯功能，包含 DMCNET 以及 Ethernet。
 - Others 为其它功能，包含扩充存储卡操作，计数计时，高速比较捕捉，中断设定。

(4) 使用台达运动控制功能块，仅需从功能块数据夹内拖拉欲使用的功能块，即可开始操作。



5.9 台达定义参数一览表

台达定义参数用于台达运动控制功能块的输入引脚，便于使用者在操作功能块时不需另外寻找功能块输入脚的说明，直接透过参数的操作即可设定想要的操作，参数列表如下：

参数名称	参数型态	数值	使用功能块	功能叙述
TRUE	BOOL	TRUE	所有功能块	输入引脚，写入 TRUE
FALSE	BOOL	FALSE		输入引脚，写入 FALSE
mcRising	BOOL	TRUE	插入两段速定位、 插入单段速定位、原点回归	DOG 信号上升沿
mcFalling	BOOL	FALSE		DOG 信号下降沿
mcPositive	BOOL	TRUE	原点回归	原点回归正转启动
mcNegative	BOOL	FALSE		原点回归反转启动
mcSCurve	BOOL	TRUE	参数设定 2	速度曲线为 S 曲线
mcTrapezoid	BOOL	FALSE		速度曲线为梯形曲线
mcNC	BOOL	TRUE	输入极性设定	常闭接点
mcNO	BOOL	FALSE		常开接点
mc32bits	BOOL	TRUE	伺服参数写入	写入参数数据类型为 32bits
mc16bits	BOOL	FALSE		写入参数数据类型为 16bits
mcUp_Up	BOOL	TRUE	高速计时	触发方式从上升沿到上升沿
mcUp_Down	BOOL	FALSE		触发方式从上升沿到下降沿
mcCmpSet	BOOL	TRUE	高速比较	比较成立时启动输出
mcCmpRst	BOOL	FALSE		比较成立时重置输出
mcMotor	WORD	0	参数设定 2	单位系为电机单位
mcMachine	WORD	1		单位系为机械单位
mcComp	WORD	2		单位系为复合单位
mcUD	WORD	0	参数设定 2、 高速计数	脉冲型态为正逆转脉冲
mcPD	WORD	1		脉冲型态为脉冲+方向
mcAB	WORD	2		脉冲型态为 AB 相脉冲
mc4AB	WORD	3		脉冲型态为 AB 相脉冲 4 倍频
mcSD_M	WORD	0	装置回存	选择操作 M 装置
mcSD_D	WORD	5		选择操作 D 装置
mcSD_W	WORD	6		选择操作 W 装置

参数名称	参数型态	数值	使用功能块	功能叙述
IntTimer	WORD	0	中断设定	中断信号来源为时间触发
IntX8	WORD	1		中断信号来源为外部 X0.8
IntX9	WORD	2		中断信号来源为外部 X0.9
IntX10	WORD	3		中断信号来源为外部 X0.10
IntX11	WORD	4		中断信号来源为外部 X0.11
IntX12	WORD	5		中断信号来源为外部 X0.12
IntX13	WORD	6		中断信号来源为外部 X0.13
IntX14	WORD	7		中断信号来源为外部 X0.14
IntX15	WORD	8		中断信号来源为外部 X0.15
mcCmpAxis1	WORD	0		高速比较
mcCmpAxis2	WORD	1	比较来源为第 2 轴目前位置	
mcCmpAxis3	WORD	2	比较来源为第 3 轴目前位置	
mcCmpAxis4	WORD	3	比较来源为第 4 轴目前位置	
mcCmpC200	WORD	4	比较来源为 C200 计数值	
mcCmpC204	WORD	5	比较来源为 C204 计数值	
mcCmpC208	WORD	6	比较来源为 C208 计数值	
mcCmpC212	WORD	7	比较来源为 C212 计数值	
mcCmpY8	WORD	0	高速比较	比较时，操作装置为 Y0.8
mcCmpY9	WORD	1		比较时，操作装置为 Y0.9
mcCmpY10	WORD	2		比较时，操作装置为 Y0.10
mcCmpY11	WORD	3		比较时，操作装置为 Y0.11
mcCmpRstC200	WORD	4		比较时，操作装置为 C200
mcCmpRstC204	WORD	5		比较时，操作装置为 C204
mcCmpRstC208	WORD	6		比较时，操作装置为 C208
mcCmpRstC212	WORD	7		比较时，操作装置为 C212

参数名称	参数型态	数值	使用功能块	功能叙述
mcCapAxis1	WORD	1	高速捕捉	捕捉来源为第 1 轴目前位置
mcCapAxis2	WORD	2		捕捉来源为第 2 轴目前位置
mcCapAxis3	WORD	3		捕捉来源为第 3 轴目前位置
mcCapAxis4	WORD	4		捕捉来源为第 4 轴目前位置
mcCapC200	WORD	7		捕捉来源为 C200 计数值
mcCapC204	WORD	8		捕捉来源为 C204 计数值
mcCapC208	WORD	9		捕捉来源为 C208 计数值
mcCapC212	WORD	10		捕捉来源为 C212 计数值
mcCapX0	WORD	0	高速捕捉	捕捉触发信号来源 X0.0
mcCapX1	WORD	1		捕捉触发信号来源 X0.1
mcCapX2	WORD	2		捕捉触发信号来源 X0.2
mcCapX3	WORD	3		捕捉触发信号来源 X0.3
mcCapX8	WORD	8		捕捉触发信号来源 X0.8
mcCapX9	WORD	9		捕捉触发信号来源 X0.9
mcCapX10	WORD	10		捕捉触发信号来源 X0.10
mcCapX11	WORD	11		捕捉触发信号来源 X0.11
mcCapX12	WORD	12		捕捉触发信号来源 X0.12
mcCapX13	WORD	13		捕捉触发信号来源 X0.13
mcCapX14	WORD	14		捕捉触发信号来源 X0.14
mcCapX15	WORD	15		捕捉触发信号来源 X0.15

5.10 单轴运动功能块说明

5.10.1 绝对单段速运动

En	T_AbsSeg1	Eno
Axis		Done
Execute		Busy
Position		Aborted
Velocity		Error

1. 功能块说明

此运动功能块，旨在启动一段速度的绝对位置运动，利用 Axis 的输入数据指定某一轴，Velocity 的输入值做为单段速运动的运行速度，Position 的输入值做为单段速的目标位置，并且此目标位置为绝对位置，来运行单段速运动。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
Position	绝对位置	DWORD	K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿时
Velocity	目标速度	DWORD	K1 ~ K2,147,483,647	执行中持续更新
输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 定位完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

其中输入引脚 Position 和 Velocity 单位预设为脉冲数以及每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。

3. 错误产生及排除

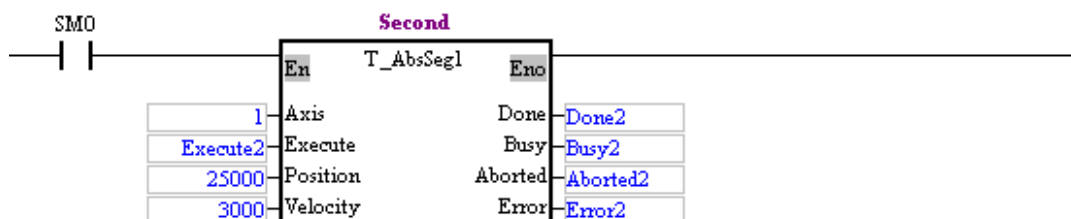
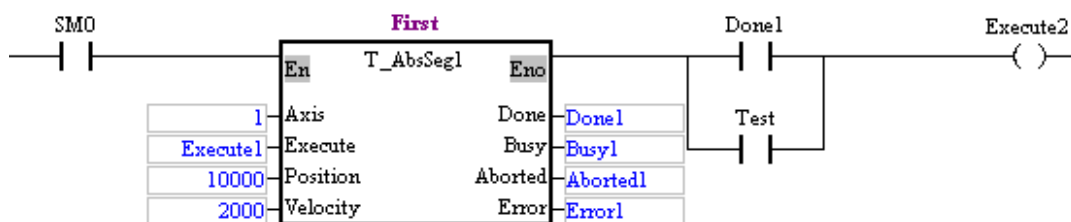
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块未启动或已完成

4. 范例程序

此范例的目的

- 第一个单段速运动完成后，接连执行第二个单段速运动。
- 第一个单段速运动尚未完成前，先执行第二个单段速运动。

首先设定运动功能块 **FIRST** 为设定第 1 轴以每秒 2000 脉冲的速度，运行至 10000 脉冲的位置，运动功能块 **SECOND** 为设定第 1 轴，以每秒 3000 脉冲的速度，运行至 25000 脉冲的位置。



- 第一个单段速运动完成后，接连执行第二个单段速运动。

操作步骤：

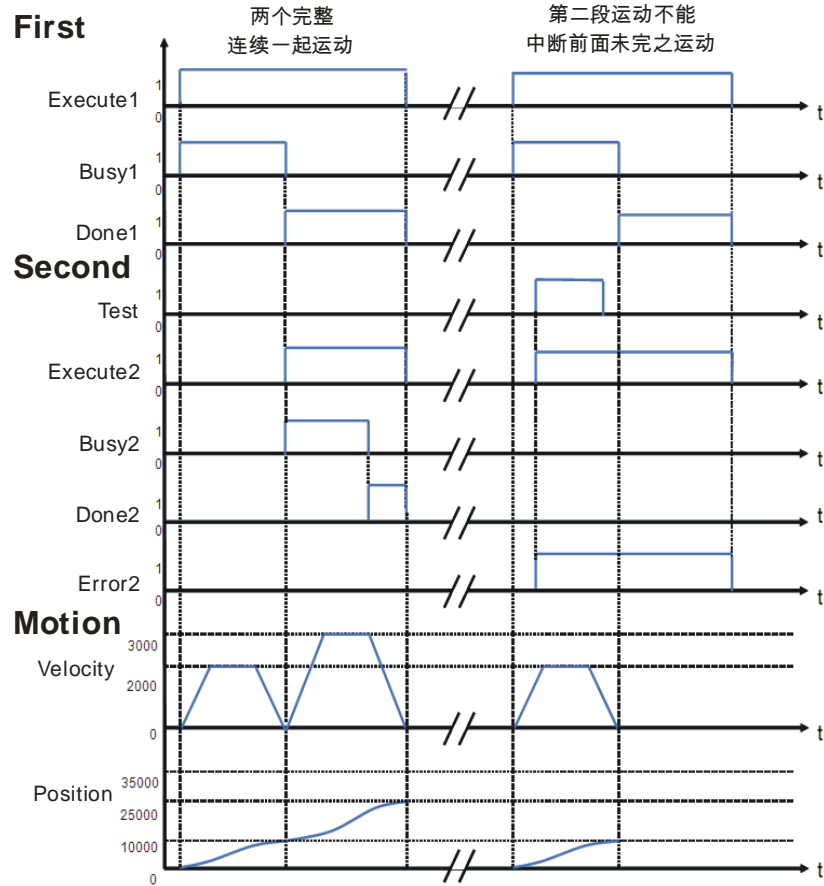
- 启动 Execute1
- 等待 Done2 或 Error2 上升沿产生

- 第一个单段速运动尚未完成前，预先执行第二个单段速运动。

操作步骤：

- 启动 Execute1
- 在 Busy1 为 True 时，启动 Test
- 等待 Done2 或 Error2 上升沿产生

两种操作结果其时序图与第 1 轴运动结果如下：



- 第一个单段速运动完成后，接连执行第二个单段速运动。
可以看到运动功能块 **FIRST** 动作完成后，轴执行运动功能块 **SECOND**，而最后轴的运动位置为 25000。
- 第一个单段速运动尚未完成前，先执行第二个单段速运动。
可看到在 **Error2** 出现后，轴的最后位置为 10000，运动功能块 **SECOND** 的命令无效。

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.2 相对单段速运动

En	T_RelSeg1	Eno
Axis		Done
Execute		Busy
Distance		Aborted
Velocity		Error

1. 功能块说明

此运动功能块，旨在启动一段速度的相对距离运动，利用 Axis 的输入数据指定某一轴，利用 Velocity 的输入值做为单段速运动的运行速度，Distance 的输入值做为单段速的移动位移量，并且此位移量为相对于当前位置的移动长度，来运行单段速运动。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
Distance	相对位移	DWORD	K-2,147,483,646 ~ K2,147,483,646	Execute 上升沿时
Velocity	目标速度	DWORD	K1 ~ K2,147,483,647	执行中持续更新

输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 定位完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

其中输入引脚 Distance 和 Velocity 单位预设为脉冲数以及每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。

3. 错误产生及排除

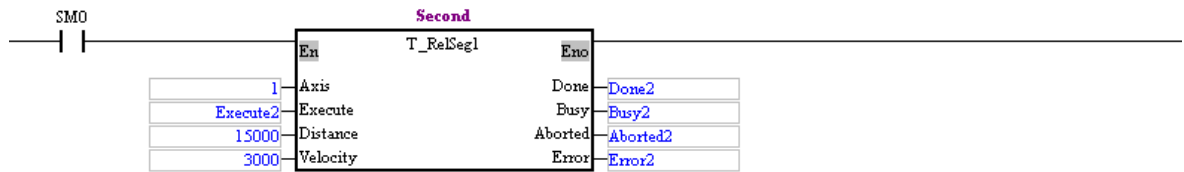
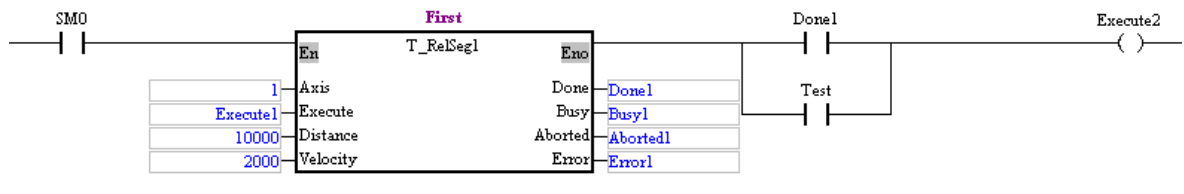
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 范例程序

此范例的目的

- 第一个单段速运动完成后，接连执行第二个单段速运动。
- 第一个单段速运动尚未完成前，先执行第二个单段速运动。

首先设定运动功能块 **FIRST** 为第 1 轴，以每秒 2000 脉冲的速度，运行 10000 个脉冲的距离，运动功能块 **SECOND** 为设定第 1 轴，以每秒 3000 脉冲的速度，运行 15000 个脉冲的距离。



- 第一个单段速运动完成后，接连执行第二个单段速运动。

操作步骤：

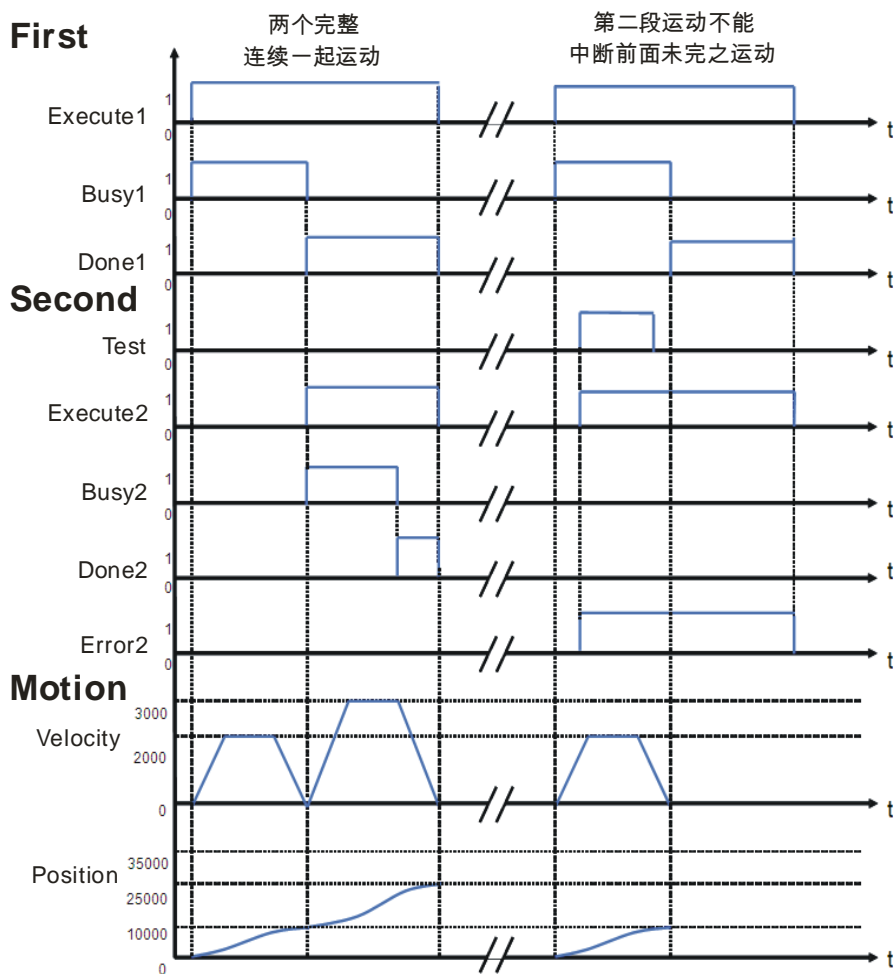
- 启动 Execute1
- 等待 Done2 或 Error2 上升沿产生

- 第一个单段速运动尚未完成前，预先执行第二个单段速运动。

操作步骤：

- 启动 Execute1
- 在 Busy1 为 True 时，启动 Test。
- 等待 Done2 或 Error2 上升沿产生

两种操作结果其时序图与第 1 轴运动结果如下：

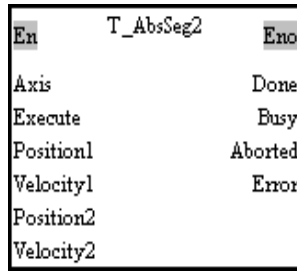


- 第一个单段速运动完成后，接连执行第二个单段速运动。
可以看到运动功能块 **FIRST** 动作完成后，轴移动到 10000，而后轴执行运动功能块 **SECOND**，轴以 10000 无开始移动 15000 的距离，其最后位置为 25000。
- 第一个单段速运动尚未完成前，先执行第二个单段速运动。
可看到在 **Error2** 出现后，轴的最后位置为 10000，运动功能块 **SECOND** 的命令无效。

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.3 绝对两段速运动



1. 功能块说明

此运动功能块，旨在启动两段速度的绝对位置运动，利用 Axis 的输入数据指定某一轴，Velocity1 的输入值做为第一段运动的运行速度，Position1 的输入值做为第一段的目标位置，在第一段运动运行到达目标位置时，利用 Velocity2 的输入值做为第二段运动的运行速度，Position2 的输入值做为第二段的目标位置，接续运行第二段运动，并且目标位置 Position1 和 Position2 为绝对位置。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
Position1	第一段绝对位置	DWORD	K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿时
Velocity1	第一段目标速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
Position2	第二段绝对位置	DWORD	K-2,147,483,648 ~ K2,147,483,647 (Position1 > 0→Position2 ≥Position1; Position1 ≤ 0→Position2 ≤Position1)	Execute 上升沿时
Velocity2	第二段目标速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时

输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 定位完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

其中输入引脚 Position1 和 Position2 单位预设为脉冲数，而 Velocity1 和 Velocity2 为每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。

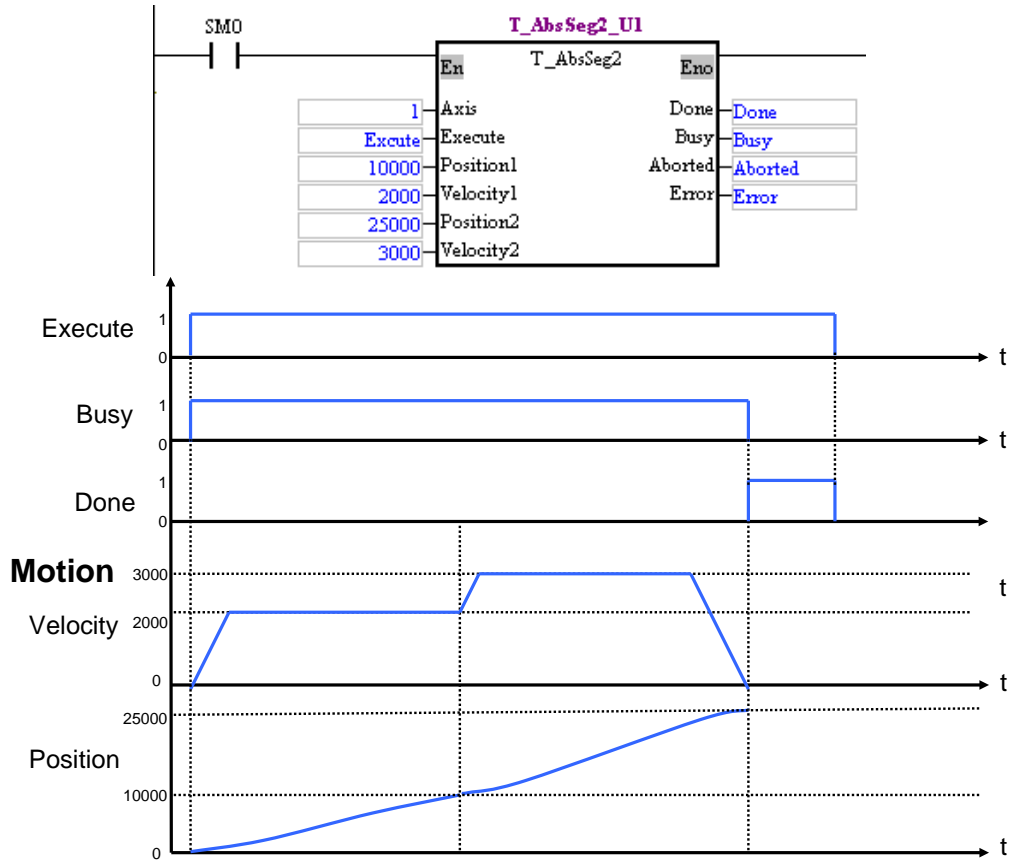
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 范例程序

此范例的目的

- 利用绝对两段速功能块，操作指定的运动轴作绝对位置的两段速运动。
 首先设定第一段运行的速度为每秒 2000 脉冲，运行的目标位置距为 10000 个脉冲，而第二段绝对位置的运行参数为每秒 3000 脉冲的速度，运行至 25000 个脉冲的目标位置。

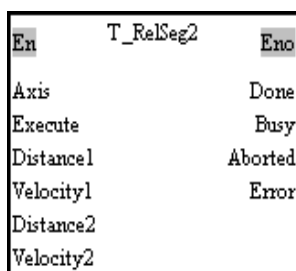


在运动功能块启动后，第一轴先以目标速度每秒 2000 脉冲运行到目标位置 10000 处，而后再以第二段目标速度每秒 3000 脉冲运行到目标位置 25000 处。

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.4 相对两段速运动



1. 功能块说明

此运动功能块，旨在启动两段速度的相对距离运动，利用 Axis 的输入数据指定某一轴，Velocity1 的输入值做为第一段运动的运行速度，Distance1 的输入值做为第一段的移动距离，在第一段运动运行达到所需的距离后，利用 Velocity2 的输入值做为第二段运动的运行速度，Distance2 的输入值做为第二段的移动距离，接续运行第二段运动，并且相对位移 Distance1 和 Distance2 为相对于第一段和第二段运动开始的位移量。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时 启动运动	BOOL	TRUE / FALSE	-
Distance1	第一段相对 位移	DWORD	K-2,147,483,646 ~ K2,147,483,646	Execute 上升沿时
Velocity1	第一段目标 速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
Distance2	第二段相对 位移	DWORD	K-2,147,483,646 ~ K2,147,483,646 且需 与 Distance1 同号	Execute 上升沿时
Velocity2	第二段目标 速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时

输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 定位完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

其中输入引脚 Distance1 和 Distance 2 单位预设 为脉冲数，而 Velocity1 和 Velocity2 为每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。

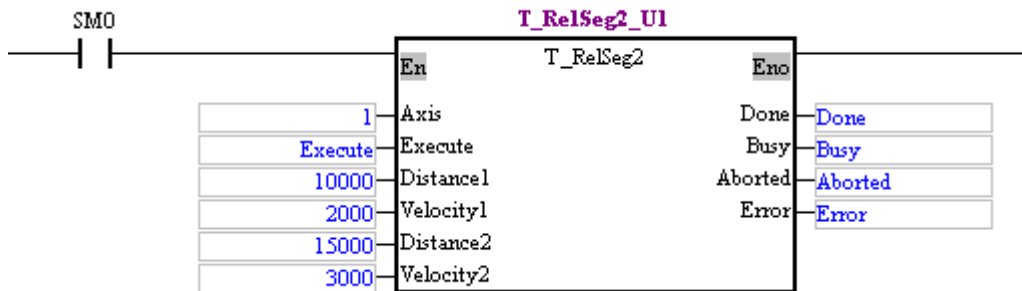
3. 错误产生及排除

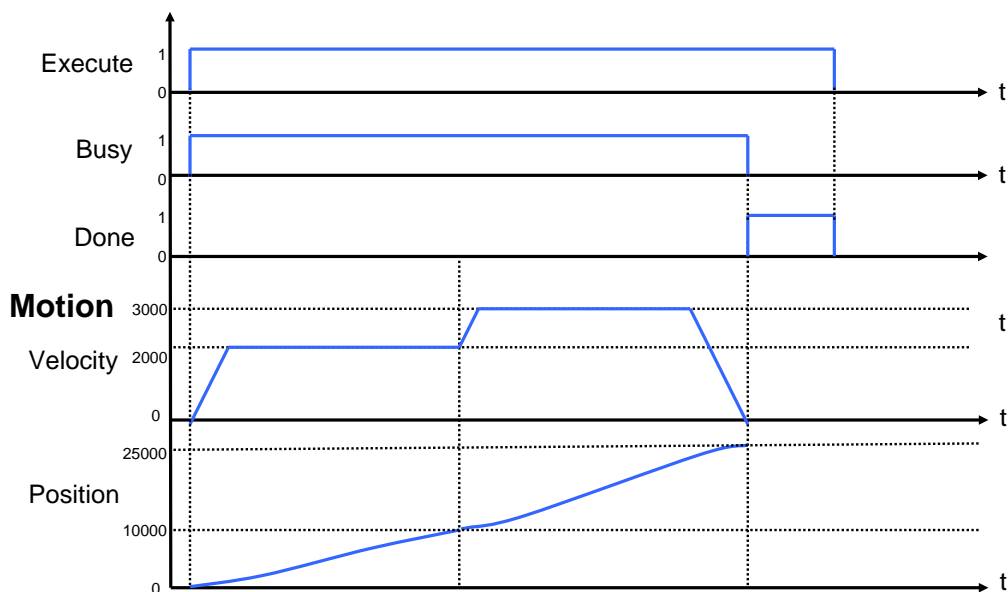
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 范例程序

此范例的目的

- 利用相对两段速功能块，操作指定的运动轴作相对位移的两段速运动。
首先设定第一段运行的速度为每秒 2000 脉冲，运行的距离为 10000 个脉冲，而第二段绝对位置的运行参数为每秒 3000 脉冲的速度，运行的距离为 15000 个脉冲。





在运动功能块启动后，第一轴先以目标速度每秒 2000 脉冲运行 10000 脉冲的距离，而后再以第二段目标速度每秒 3000 脉冲运行 25000 个脉冲的距离。

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

5.10.5 插入单段速运动

En	T_TrSeg1	Eno
Axis		Done
Execute		Busy
DogEdge		Aborted
Distance		Error
Velocity		

1. 功能块说明

此运动功能块，旨在启动插入单段速度运动，利用 Axis 的输入数据指定某一轴，在启动后利用 Velocity 的输入值做为运动的运行速度，而后依照 DogEdge 的输入值做为运动启动信号的设定，可设定运动启动为 DOG 的上升沿或 DOG 的下降沿做为位移距离的开始信号，而位移距离为 Distance 的输入值，并且相对位移 Distance 为相对于启动点的位移量。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时 启动运动	BOOL	TRUE / FALSE	-
DogEdge	近点触发方 向	BOOL	mcRising (TRUE) / mcFalling (FALSE)	Execute 上升沿时
Distance	DOG 触发 后插入的位 移	DWORD	K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿时
Velocity	目标速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时

输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 定位完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其 它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生 错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

其中输入引脚 **Distance** 单位预设脉冲数，而 **Velocity** 为每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。而 **DogEdge** 可利用变量 **mcRising** 和 **mcFalling** 来设定启动信号为 **DOG** 的上升沿或下降沿。

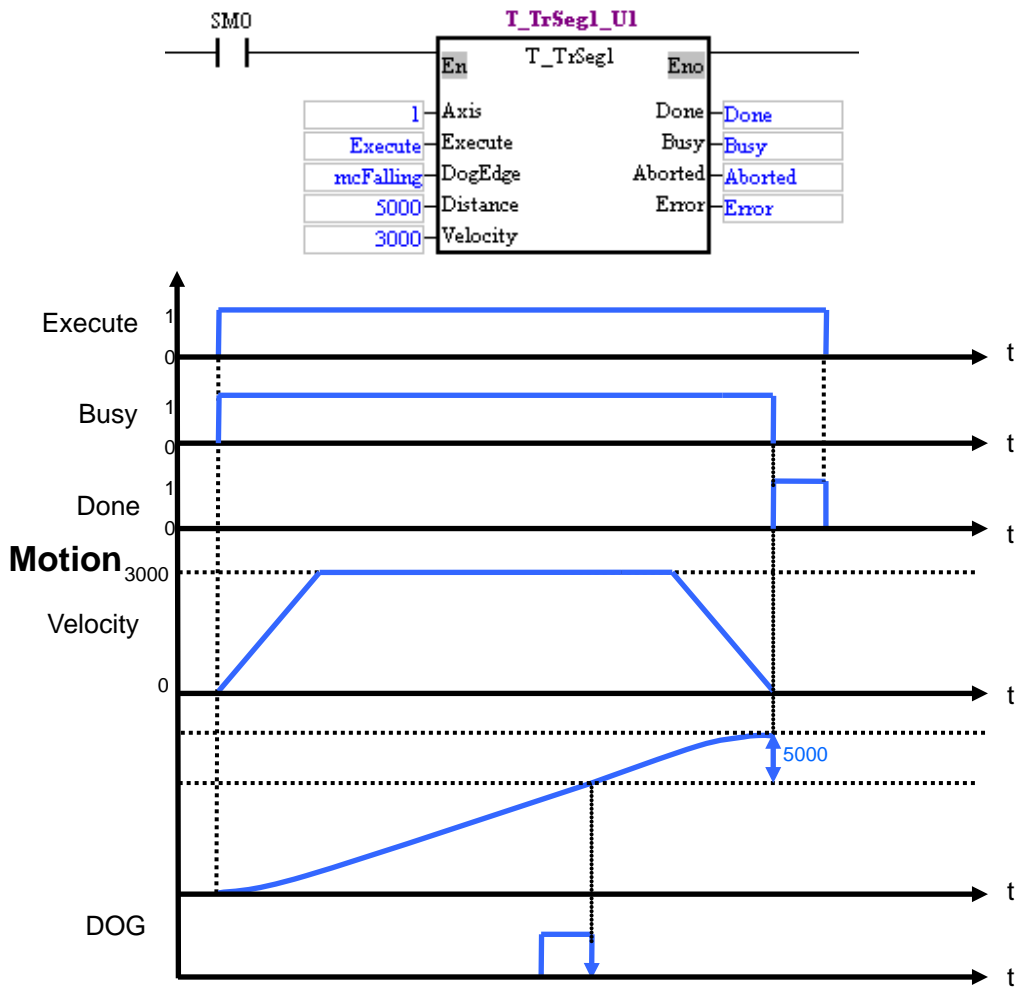
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 范例程序

范例一：

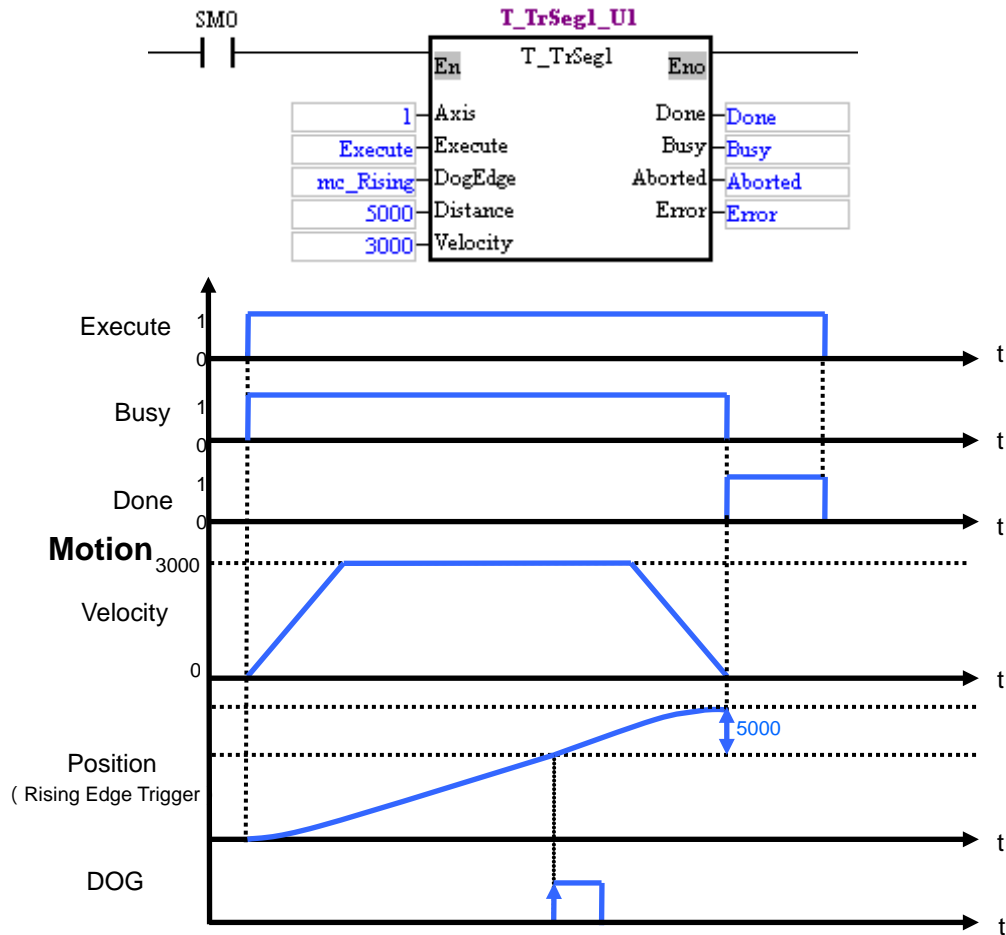
- 利用插入单段速功能块，操作指定的运动轴利用 DOG 下降沿作单段速运动的起始信号。首先设运行的速度为每秒 3000 脉冲，而遇到 DOG 的下降沿开始运行的距离为 5000 个脉冲，可看到运动轨迹再遇到 DOG 下降沿后，运行 5000 个脉冲，等到 5000 个脉冲的距离完成后，Done 信号被设定。



5

范例二：

- 利用插入单段速功能块，操作指定的运动轴利用 DOG 上升沿作单段速运动的切换信号。首先设运行的速度为每秒 3000 脉冲，而遇到 DOG 的上升沿开始运行的距离为 5000 个脉冲，可看到运动轨迹再遇到 DOG 上升沿后，运行 5000 个脉冲，等到 5000 个脉冲的距离完成后，Done 信号被设定。



5

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.6 插入两段速运动

En	T_TrSeg2	Eno
Axis		Done
Execute		Busy
DogEdge		Aborted
Velocity1		Error
Distance		
Velocity2		

1. 功能块说明

此运动功能块，旨在启动插入两段速度运动，利用 Axis 的输入数据指定某一轴，在启动后利用 Velocity1 的输入值做为运动的运行速度，而后依照 DogEdge 的输入值做为运动启动信号的设定，可设定运动启动为 DOG 的上升沿或 DOG 的下降沿做为位移距离的开始信号，其运动的速度为 Velocity2 的输入值而运动的位移距离为 Distance 的输入值，并且相对位移 Distance 为相对于启动点的位移量。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时 启动运动	BOOL	TRUE / FALSE	-
DogEdge	近点触发方向	BOOL	mcRising (TRUE) / mcFalling (FALSE)	Execute 上升沿时
Velocity1	DOG 触发前之 目标速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
Distance	DOG 触发后插 入的位移	DWORD	K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿时
Velocity2	DOG 触发后之 目标速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时

输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 定位完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

其中输入引脚 Distance 单位预设 为脉冲数，而 Velocity1 和 Velocity2 为每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。而 DogEdge 可利用变量 mcRising 和 mcFalling 来设定启动信号为 DOG 的上升沿或下降沿。

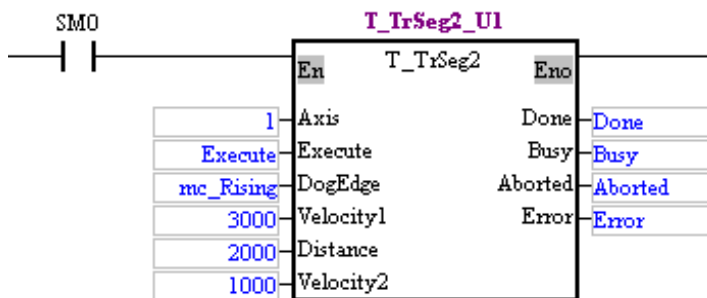
5

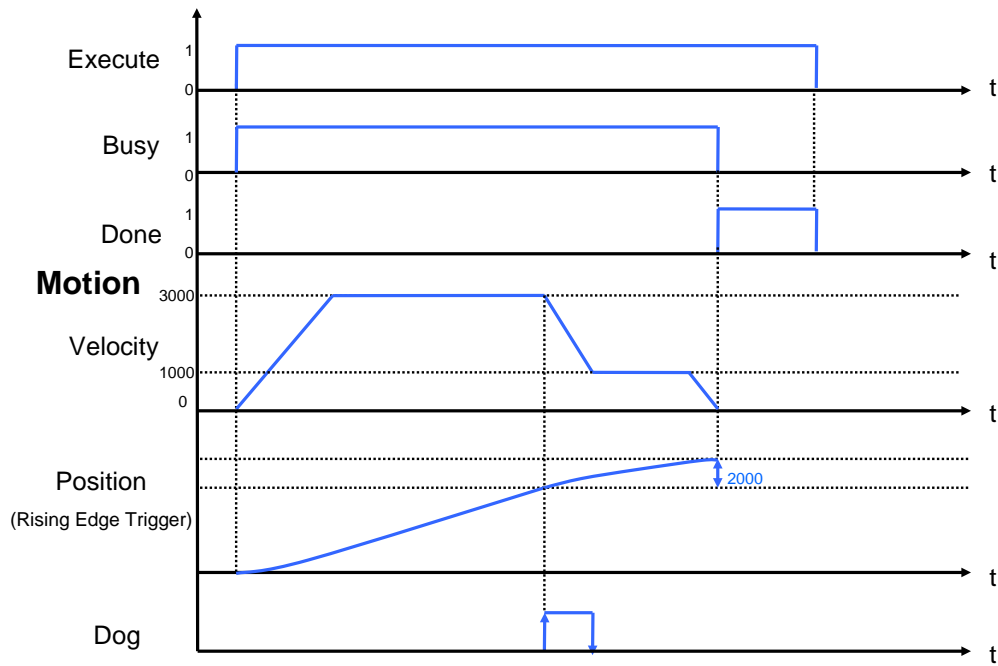
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 范例程序

利用插入两段速功能块，操作指定的运动轴利用 DOG 上升沿作两段速运动的切换信号。首先设运行的速度为每秒 3000 脉冲，而遇到 DOG 的上升沿开始以每秒 1000 脉冲的速度，运行 5000 个脉冲的距离。





可看到运动轨迹再遇到 DOG 上升沿前，先以每秒 3000 脉冲的速度运转，而后遇到 DOG 上升沿，以每秒 1000 个脉冲，运行 2000 个脉冲的距离，等到距离到达后，Done 信号被设定。

5

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.7 寸动运动

En	T Jog	Eno
Axis		Busy
PositiveEnable		Aborted
NegativeEnable		Error
Velocity		

1. 功能块说明

此运动功能块，旨在启动寸动运动，利用 Axis 的输入数据指定某一轴，利用 Velocity 的输入值做为寸动的运行速度，而后依照 PositiveEnable 和 NegativeEnable 的输入值上升沿做为运动启动设定，其中 PositiveEnable 为正向的寸动运动，NegativeEnable 为反向的寸动运动。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
PositiveEnable	正向寸动运动允许	BOOL	TRUE / FALSE	<ul style="list-style-type: none"> ● 正反向同时上升沿触发时 正向优先于反向，同时反向允许会被 Reset 到 FALSE。 ● 仅 PositiveEnable 上升沿触发时，若 NegativeEnable 为 TRUE 则 NegativeEnable 会被 Reset 到 FALSE 同时由反向运动减速到停止后再转成正向运动。
NegativeEnable	反向寸动运动允许	BOOL	TRUE / FALSE	<ul style="list-style-type: none"> ● 正反向同时上升沿触发时 正向优先于反向，同时反向允许会被 Reset 到 FALSE。 ● 仅 NegativeEnable 上升沿触发时，若 PositiveEnable 为 TRUE 则 PositiveEnable 会被 Reset 到 FALSE，同时由正向运动减速到停止后再转成反向运动。
Velocity	目标速度	DWORD	K1 ~ K2,147,483,647	执行中持续更新

输出引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● PositiveEnable 上升沿或 NegativeEnable 上升沿时 	<ul style="list-style-type: none"> ● 运动停止时 ● Error 上升沿时 ● Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> ● 执行中被命令停止 	<ul style="list-style-type: none"> ● PositiveEnable 下降沿或 NegativeEnable 下降沿时 ● 被中断时，若 PositiveEnable 和 NegativeEnable 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 ● 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> ● PositiveEnable 下降沿或 NegativeEnable 下降沿时

其中输入引脚 Velocity 为每秒脉冲数，若需修改单位系需利用参数设定的运动功能块修改。

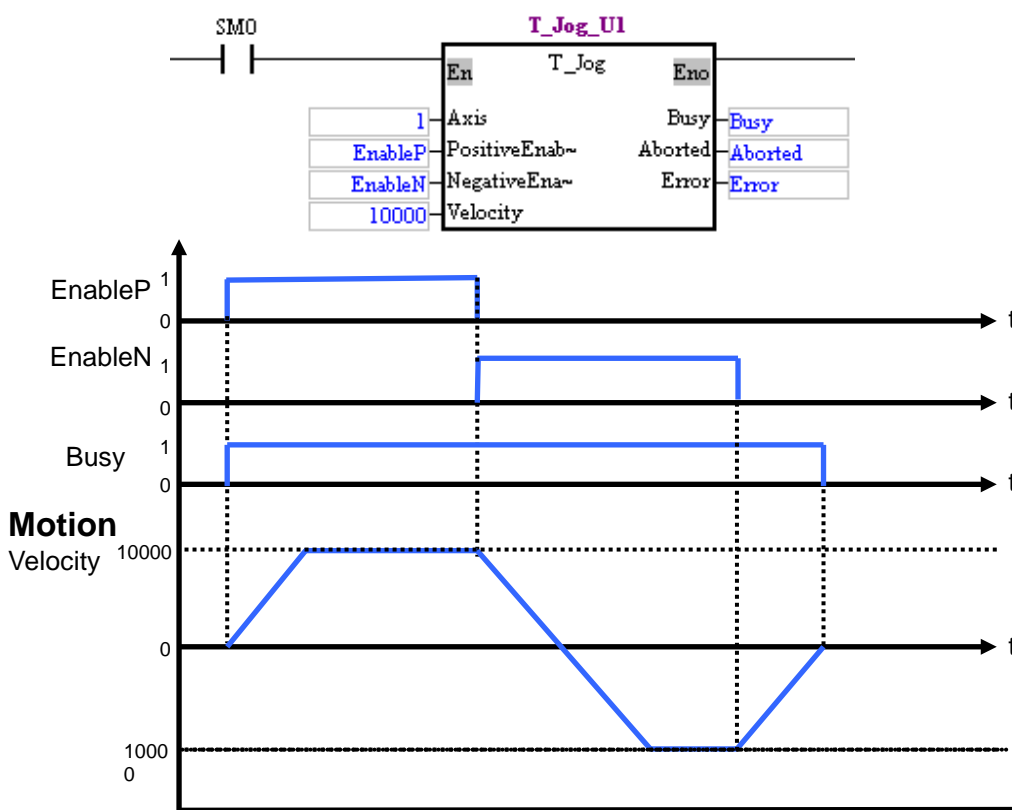
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 范例程序

利用寸动功能块，操作指定的运动轴，利用 EnableP 操作正向寸动，利用 EnableN 操作反向寸动。

首先设运行的速度为每秒 10000 脉冲，而利用操作 EnableP 和 EnableN 为 1 来控制第 1 轴作正向寸动与反向寸动。



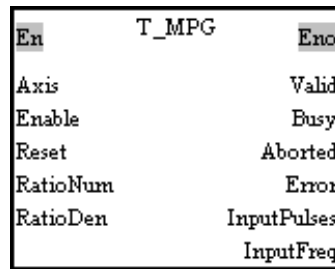
当 EnableP 为 1 时，第一轴以每秒 10000 脉冲的速度运转，作正向寸动，而当 EnableN 为 1 时，第一轴以每秒 10000 个脉冲作反向寸动，当两者均非 1 时第一轴停止运动。

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

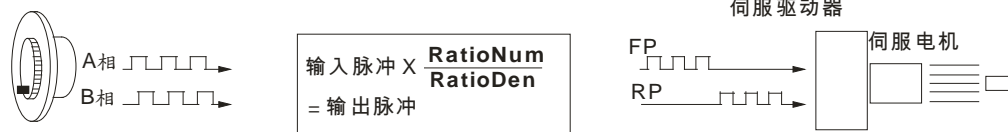
5.10.8 手摇轮模式运动



1. 功能块说明

此运动功能块，旨在启动手摇轮模式运动，利用 Axis 的输入数据指定某一轴，此轴的运动为跟随外部手摇轮动作，而轴位置与外部手轮输入的比例关系，由 RatioNum 和 RatioDen 决定。

伺服驱动器



手摇轮硬件配线请参阅“第二章 硬件规格及配线方式”说明。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Enable 上升沿时
Enable	手摇轮模式运动允许	BOOL	TRUE / FALSE	-
Reset	重置手摇轮计数量	BOOL	TRUE / FALSE	Enable 上升沿时
RatioNum	电子齿轮比之分子	WORD	K0 ~ K32,767	执行中持续更新
RatioDen	电子齿轮比之分母	WORD	K1 ~ K32,767	执行中持续更新

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输入之数值有效	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> 运动停止时 Error 上升沿时 Aborted 上升沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Enable 下降沿时 被中断时，若 Enable 为 FALSE，下个周期 Aborted 会变成 FALSE
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Enable 下降沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Enable下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出范围	更新型式
InputPulses	累计手摇轮输入脉冲个数	DWORD	K-2,147,483,648 ~ K2,147,483,647	执行中持续更新
InputFreq	手摇轮输入频率	DWORD	K 0 ~ K 2,147,483,647	执行中持续更新

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.9 原点回归运动

En	T_HomeReturn	End
Axis		Done
Execute		Busy
Direction		Aborted
DogEdge		Error
HomePosition		
VRT		
VCR		
Signal_N		
Distance_P		

1. 功能块说明

此运动功能块，目的在启动原点回归运动，利用 Axis 的输入数据指定某一轴，依照 Direction 决定原点回归方向，VRT 决定原点回归的速度，而 DogEdge 决定遇到原点以上升沿或下降沿为触发信号，再遇到原点以后以 VCR 做为原点到零点的速度，零点信号的个数以 Signal_N 决定，若有补充的脉冲数需增加则利用 Distance_P 大小决定，原点回归动作完成后以 Home Position 的数值做为该轴的当前位置。原点回归的模式为「正常模式」可参考第 7.6 节说明。

位置 (4)：目前不支持。

2. 输入/输出说明：

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K6	Execute 上升沿时
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
Direction	原点回归方向切换	BOOL	mcNegative (FALSE) / mcPositive (TRUE)	Execute 上升沿时
DogEdge	DOG 近点触发方向	BOOL	mcFalling (FALSE) / mcRising (TRUE)	Execute 上升沿时
Home Position	原点位置定义	DWORD	K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿时
VRT	原点回归速度	DWORD	K1 ~ K1000000 Vbias < VRT ≤ Vmax	Execute 上升沿时
VCR	原点回归减速速度	DWORD	K1 ~ VRT	Execute 上升沿时
Signal_N	零点信号数	WORD	K0 ~ K32,767 (仅适用于轴 1~轴 4)	Execute 上升沿时
Distance_P	补充脉冲数	WORD	K-32768 ~ K32,767	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	<ul style="list-style-type: none"> 原点回归完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

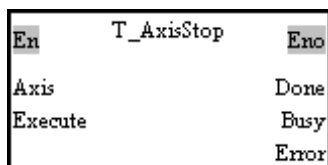
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A 及 AH15PM-5A 此三种 AH500 机种之运动控制模块。

5.10.10 单轴运动停止



1. 功能块说明

此运动功能块，目的在停止运动，利用 Axis 的输入数据指定某一轴，将其运动动作强制停止。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时 运动命令停止	BOOL	TRUE / FALSE	-

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	停止完成	BOOL	<ul style="list-style-type: none"> 执行完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE。
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 该轴目前在非单轴、齿轮及凸轮之类别执行运动 	<ul style="list-style-type: none"> Execute 下降沿时

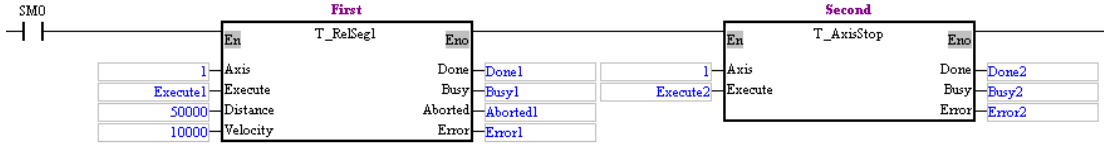
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
功能块动作冲突	确认功能块在启动时，其它单轴运动功能块为未启动或已完成

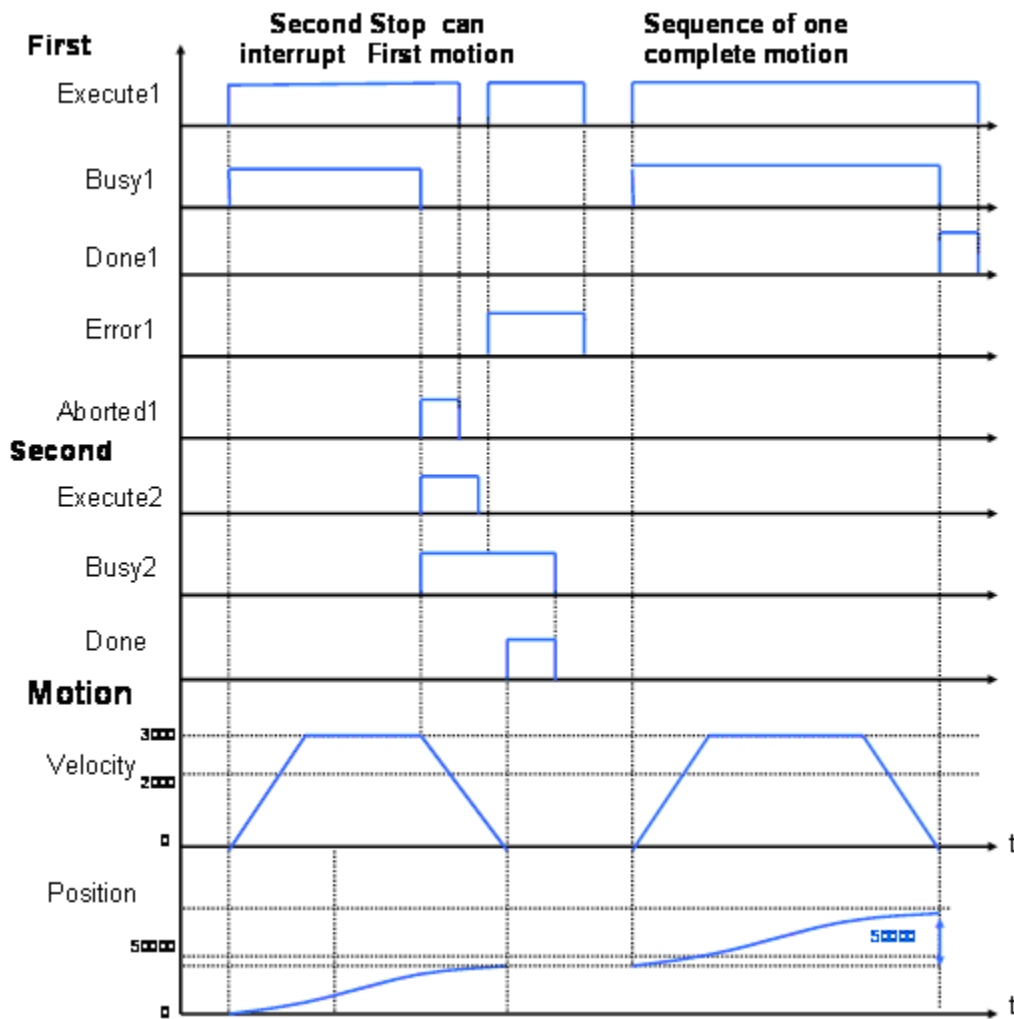
4. 范例程序

指定特定轴作相对单段速运动，再利用运动停止功能块将动作强制停止。

FIRST 为一个相对单段速运动之功能块，其中设定第 1 轴以每秒 10000 脉冲的速度运行 50000 脉冲的距离，SECOND 为运动停止功能块，所指定的运动轴为第 1 轴。



启动 FIRST 运动功能块，在 Done1 尚未为 1 时，利用 Execute2 启动 SECOND。



启动 FIRST 后 第 1 轴以目标速度每秒 10000 脉冲运行 在 SECOND 启动后 FIRST 的 Aborted1 信号会被设置为 1，此时 Busy1 为 0，第 1 轴同时开始停止动作。再停止动作的过程中，不接受任何运动的启动，若启动其它运动动作均会发生错误信息。

5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

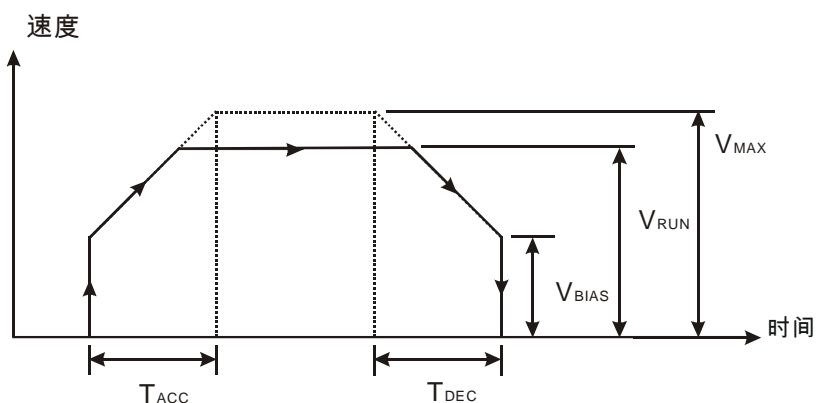
5.10.11 运动轴参数设定I

En	T_AxisSettingI	Eno
Axis		Done
Execute		Busy
Vmax		Error
Vbias		
Tacc		
Tdec		

1. 功能块说明

此运动功能块，旨在设定运动相关参数，利用 Axis 的输入数据指定某一轴，设定此轴在运动过程中可以达到的最高速度，启动速度，加速时间及减速时间。

下图为加减速时间、启动速度、最大速度与运行速度关系：



V_{RUN} 表示为执行速度。执行中的速度会依据加减速时间、启动速度、最大速度的设定作运行。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时 写入参数	BOOL	TRUE / FALSE	-
Vmax	最高速度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
Vbias	启动速度	DWORD	K0 ~ K2,147,483,647	Execute 上升沿时
Tacc	加速时间 (单位:ms)	WORD	K0 ~ K32,767	Execute 上升沿时
Tdec	减速时间 (单位:ms)	WORD	K0 ~ K32,767	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	写入完成	BOOL	<ul style="list-style-type: none"> 执行完成时 	<ul style="list-style-type: none"> Execute 下降沿时 定位完成时若Execute为FALSE，下个周期Done会变成FALSE。
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error上升沿时 Aborted 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 	<ul style="list-style-type: none"> Execute下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

5.10.12 运动轴参数设定II

En	T_AxisSetting2	Eno
Axis		Done
Execute		Busy
Vcurve		Error
OutputType		
Unit		
PulseRev		
DistanceRev		

1. 功能块说明

此运动功能块，目的在设定运动相关参数，利用 Axis 的输入数据指定某一轴，设定此轴在运动过程中的速度曲线，控制脉冲的输出型式，以及目前运动过程所在的位置与速度的单位系设定，并且单位系的设定需同时有电机转一圈所需的脉冲数及电机转一圈的移动距离。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时	BOOL	TRUE / FALSE	-

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
	写入参数			
Vcurve	速度曲线	BOOL	mcTrapezoid:FALSE mcSCurve:TRUE	Execute 上升沿时
OutputType (不支持 20MC)	输出脉冲型式	WORD	mcUD:0 mcPD:1 mcAB:2 mc4AB:3	Execute 上升沿时
Unit	单位系设定	WORD	mcMotor:0 mcMachine:1 mcComp:2	Execute 上升沿时
PulseRev	电机转一圈所需脉冲数	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
DistanceRev	电机转一圈之移动距离	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	写入完成	BOOL	• 执行完成时	<ul style="list-style-type: none"> • Execute 下降沿时 • 定位完成时若Execute为FALSE，下个周期Done会变成FALSE。
Busy	执行中	BOOL	• Execute 上升沿时	<ul style="list-style-type: none"> • Done 上升沿时 • Error上升沿时 • Aborted 上升沿时
Error	功能块产生错误	BOOL	• 输入参数错误	• Execute下降沿时

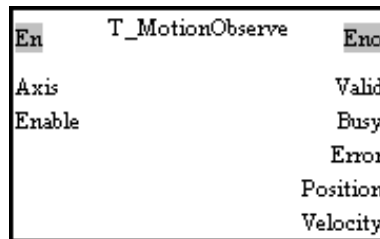
3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.13 读取当前位置/速度



1. 功能块说明

此运动功能块，目的在读取轴当前命令的位置与速度，利用 Axis 的输入数据指定某一轴，而在启动此功能块后，可透过 Position 得知此轴的当前命令位置，Velocity 得知此轴的当前命令速度。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Enable 上升沿时
Enable	手摇轮模式 运动允许	BOOL	TRUE / FALSE	-

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输入之数值有效	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> Enable 下降沿时 Error 上升沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> Enable 下降沿时 Error 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 	<ul style="list-style-type: none"> Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出范围	更新型式
Position	当前位置 (Pulse Unit)	DWORD	K -2,147,483,648 ~ K 2,147,483,647	执行中持续更新
Velocity	当前速度 (Pulse Unit)	DWORD	K 0 ~ K 2,147,483,647	执行中持续更新

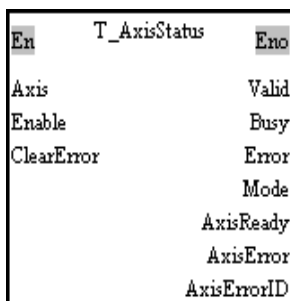
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.14 轴状态信息



1. 功能块说明

此运动功能块，目的为读取以及清除轴的错误状态，利用 Axis 的输入数据指定某一轴，透过 ClearError 清除轴的错误状态，轴目前的错误状态则透过 AxisErrorID 显示。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Enable 上升沿时
Enable	轴状态显示允许	BOOL	TRUE / FALSE	-
ClearError	上升沿触发清除错误状态	BOOL	TRUE / FALSE	执行中

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> Enable 下降沿时 Error 上升沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> Enable 下降沿时 Error 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 	<ul style="list-style-type: none"> Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出范围	更新型式
Mode	运动模式	WORD	16#0~16#32x (*1)	执行中持续更新
AxisReady	运动轴待命标志	BOOL	TRUE / FALSE	执行中持续更新

输出数值引脚				
引脚名称	功能说明	数据格式	输出范围	更新型式
AxisError	运动轴错误标志	BOOL	TRUE / FALSE	执行中持续更新
AxisErrorID	错误代码	WORD	16#0002~16#C4FF、 16#8001~16#8380 (*2)	执行中持续更新

*1：Mode 输入值定义如下

输入值	定义
16#0	闲置
16#100	单轴运动停止中
16#101	绝对单段速定位运动
16#102	相对单段速定位运动
16#103	绝对连续两段速定位运动
16#104	相对连续两段速定位运动
16#105	插入单速段定位运动
16#106	插入两段速定位运动
16#107	寸动运动
16#108	手摇轮模式运动
16#109	原点回归运动
16#10A	电子齿轮运动
16#10B	电子凸轮运动
16#200	G 码运行停止中
16#201	G 码运行
16#300	多轴插补停止中
16#31x	多轴绝对直线插补运动
16#32x	多轴相对直线插补运动

*2：AxisErrorID 输入值定义如下

输入值	定义
16#0002~16#C4FF	AH 模块错误
16#8001~16#8380	ASD-A2-F 发生错误

- AH500 模块错误码定义请参照附录 16 章
- ASD-A2-F 错误码定义如下：
ASD-A2-F 的 AL 码+16#8000 为 AxisErrorID

5

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.15 设定当前位置

En	T_SetPosition	Enc
Axis		Done
Execute		Busy
Position		Error

1. 功能块说明

此运动功能块·目的在设定运动轴当前命令位置·利用 Axis 的输入数据指定某一轴·透过 Position 设定此轴的当前命令位置。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K16	Execute 上升沿时
Execute	上升沿时 写入当前位置	BOOL	TRUE / FALSE	-
Position	当前位置	DWORD	K -2,147,483,648 ~ K 2,147,483,647	Execute 上升沿时
输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	运动完成	BOOL	• 位置写入完成	<ul style="list-style-type: none"> • Execute 下降沿时 • 定位完成时若 Execute 为 FALSE·下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	• Execute 上升沿时	<ul style="list-style-type: none"> • Done 上升沿时 • Error 上升沿时
Error	功能块产生 错误	BOOL	• 输入参数错误	• Execute 下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值·是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

注意:请避免用于设置同步运动(凸轮、齿轮)主轴之当前位置·避免运动出现异常。

5.10.16 输入极性设定

20MC/10PM

En	T_InputPolarity	Eno
Enable		Valid
X0_Pg0	Pg0_X0	
X1_Pg1	Pg1_X1	
X2_Pg2	Pg2_X2	
X3_Pg3	Pg3_X3	
X8_mpgA	mpgA_X8	
X9_mpgB	mpgB_X9	
X10_Dog4	Dog4_X10	
X11_Dog5	Dog5_X11	
X12_Dog0	Dog0_X12	
X13_Dog1	Dog1_X13	
X14_Dog2	Dog2_X14	
X15_Dog3	Dog3_X15	
		Busy

15PM

En	T_InputPolarity	Eno
Enable		Valid
X00_Pg0	Pg0_X00	
X01_Pg1	Pg1_X01	
X02_Pg2	Pg2_X02	
X03_Pg3	Pg3_X03	
X04_Dog0	Dog0_X04	
X05_Dog1	Dog1_X05	
X06_Dog2	Dog2_X06	
X07_Dog3	Dog3_X07	
X08_mpgA	mpgA_X08	
X09_mpgB	mpgB_X09	
X0A_LSP0	LSP0_X0A	
X0B_LSN0	LSN0_X0B	
X0C_LSP1	LSP1_X0C	
X0D_LSN1	LSN1_X0D	
X0E_LSP2	LSP2_X0E	
X0F_LSN2	LSN2_X0F	
X10_LSP3	LSP3_X10	
X11_LSN3	LSN3_X11	
X12_CHG0	CHG0_X12	
X13_CHG1	CHG1_X13	
X14_CHG2	CHG2_X14	
X15_CHG3	CHG3_X15	
		Busy

05PM

En	T_InputPolarity	Eno
Enable		Valid
X0_Pg0	Pg0_X0	
X1_Pg1	Pg1_X1	
X8_mpgA	mpgA_X8	
X9_mpgB	mpgB_X9	
X12_Dog0	Dog0_X12	
X13_Dog1	Dog1_X13	
		Busy

1. 功能块说明

此功能块，旨在设定与输入点极性与读取输入点状态。输入端分别可设定为常开接点或是常闭接点；输出端可读取当前端子状态。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Enable	端子极性设定与状态显示允许	BOOL	TRUE / FALSE	-
X0_Pg0	端子极性	BOOL	mcNO:FALSE mcNC:TRUE	执行中持续更新
X1_Pg1	端子极性	BOOL		
X2_Pg2	端子极性	BOOL		
X3_Pg3	端子极性	BOOL		
X4_Dog0 ^{*1}	端子极性	BOOL		
X5_Dog1 ^{*1}	端子极性	BOOL		
X6_Dog2 ^{*1}	端子极性	BOOL		
X7_Dog3 ^{*1}	端子极性	BOOL		
X8_mpgA	端子极性	BOOL		
X9_mpgB	端子极性	BOOL		



输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
X0A_LSP0 ^{*1}	端子极性	BOOL		
X0B_LSN0 ^{*1}	端子极性	BOOL		
X0C_LSP1 ^{*1}	端子极性	BOOL		
X0D_LSN1 ^{*1}	端子极性	BOOL		
X0E_LSP2 ^{*1}	端子极性	BOOL		
X0F_LSN2 ^{*1}	端子极性	BOOL		
X10_Dog4 /X10_LSP3 ^{*1}	端子极性	BOOL		
X11_Dog5 /X11_LSN3 ^{*1}	端子极性	BOOL		
X12_Dog0 /X12_CHG0 ^{*1}	端子极性	BOOL		
X13_Dog1 /X12_CHG1 ^{*1}	端子极性	BOOL		
X14_Dog2 /X12_CHG2 ^{*1}	端子极性	BOOL		
X15_Dog3 /X12_CHG3 ^{*1}	端子极性	BOOL		

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输入之数值有效	BOOL	<ul style="list-style-type: none"> • Enable 上升沿时 	<ul style="list-style-type: none"> • Enable 下降沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> • Enable 上升沿时 	<ul style="list-style-type: none"> • Enable 下降沿时
Pg0_X0	端子极性	BOOL	<ul style="list-style-type: none"> • 执行中输入引脚为 ON 且外部信号为 OFF • 执行中输入引脚为 OFF 且外部信号为 ON 	<ul style="list-style-type: none"> • 执行中输入引脚为 ON 且外部信号为 ON • 执行中输入引脚为 OFF 且外部信号为 OFF • Enable 下降沿时
Pg1_X1	端子极性	BOOL		
Pg2_X2	端子极性	BOOL		
Pg3_X3	端子极性	BOOL		
Dog0_X4 ^{*1}	端子极性	BOOL		
Dog1_X5 ^{*1}	端子极性	BOOL		
Dog2_X6 ^{*1}	端子极性	BOOL		
Dog3_X7 ^{*1}	端子极性	BOOL		
mpgA_X8	端子极性	BOOL		
mpgB_X9	端子极性	BOOL		
LSP0_X0A ^{*1}	端子极性	BOOL		
LSN0_X0B ^{*1}	端子极性	BOOL		
LSP1_X0C ^{*1}	端子极性	BOOL		
LSN1_X0D ^{*1}	端子极性	BOOL		
LSP2_X0E ^{*1}	端子极性	BOOL		
LSN2_X0F ^{*1}	端子极性	BOOL		
Dog4_X10 /LSP3_X10 ^{*1}	端子极性	BOOL		
Dog5_X11 /LSN3_X11 ^{*1}	端子极性	BOOL		
Dog0_X12 /CHG0_X12 ^{*1}	端子极性	BOOL		
Dog1_X13 /CHG1_X13 ^{*1}	端子极性	BOOL		
Dog2_X14 /CHG2_X14 ^{*1}	端子极性	BOOL		
Dog3_X15 /CHG3_X15 ^{*1}	端子极性	BOOL		

*1 : 此为 AH15PM-5A 引脚编号

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块：

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.17 电子齿轮运动

En	T_GearIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
RatioNum		InputPulses
RatioDen		InputFreq

1. 功能块说明

此运动功能块，旨在启动电子齿轮运动，利用 **Master** 输入决定主轴的来源，**Slave** 输入指定某一轴为从轴，从轴将会做跟随主轴运动。在此动作过程中，**RatioNum** 和 **RatioDen** 决定从轴跟随主轴齿轮比的分子与分母，而 **Reset** 信号做为主轴累计输入脉冲数的清除信号。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Master	主轴编号	WORD	0~16、200、204、208、212、216、220 (*1)	Enable 上升沿时
Slave	从轴编号	WORD	1~16	Enable 上升沿时
Enable	齿轮啮合运动允许	BOOL	TRUE / FALSE	-
Reset	允许时重置 InputPulses	BOOL	TRUE / FALSE	Enable 上升沿时
RatioNum	电子齿轮比之分子	DWORD	K-32,767 ~ K32,767	执行中持续更新
RatioDen	电子齿轮比之分母	DWORD	K1 ~ K32,767	执行中持续更新

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> 运动停止时 Error 上升沿时 Aborted 上升沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Enable 下降沿时 被中断时，若 Enable 为 FALSE，下个周期 Aborted 会变成 FALSE
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Enable 下降沿时
Error	功能块产生	BOOL	<ul style="list-style-type: none"> 输入参数错误 	<ul style="list-style-type: none"> Enable 下降沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
	错误		<ul style="list-style-type: none"> 启动时设定轴已在运动状态 	

输出数值引脚				
引脚名称	功能说明	数据格式	输出范围	更新型式
InputPulses	累计主轴输入脉冲个数	DWORD	K-2,147,483,648 ~ K2,147,483,647	执行中持续更新
InputFreq	主轴输入脉冲频率	DWORD	K 0 ~ K 2,147,483,647	执行中持续更新

(*1) Master 输入值定义如下

输入值	定义
0	手摇轮
1~16	运动轴 1~16
200	计数器 C200
204	计数器 C204
208	计数器 C208
212	计数器 C212
216	计数器 C216
220	计数器 C220

3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.18 凸轮啮合

En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop		CycleStartFlag
MasterOffset		Index
MasterScaling		InputPulses
SlaveScaling		InputFreq

1. 功能块说明

此运动功能块，旨在启动凸轮运动，利用 **Master** 输入决定主轴的来源，**Slave** 输入指定某一轴为从轴，从轴将会做跟随主轴作凸轮运动。在此动作过程中，**MasterOffset** 决定从轴凸轮运动对于主轴的起始角度，**Reset** 信号做为主轴累计输入脉冲数的清除信号，运动过程中，当输入 **CamOut** 信号时，从轴脱离凸轮运动；**Enable** 重置时若有输入 **CycleStop** 信号，凸轮执行完整个周期时才会停止运动。细部说明可参考第 8.2.2 节。

2. 输入/输出说明

5

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Master	主轴编号	WORD	0~16、200、204、 208、212、216、 220 (*1)	Enable 上升沿时
Slave	从轴编号	WORD	1~16	Enable 上升沿时
Enable	电子凸轮运动允许	BOOL	TRUE / FALSE	-
Reset	允许时重置 InputPulses	BOOL	TRUE / FALSE	Enable 上升沿时
CamOut	脱离主轴咬合命令	BOOL	TRUE / FALSE	执行中持续取用
CycleStop	整周期停止命令	BOOL	TRUE / FALSE	Enable 下降沿时
MasterOffset	凸轮表主轴起始角度 (单位 pulses)	DWORD	K 0 ~ K 2,147,483,647	Enable 上升沿时
MasterScaling	主轴缩放比例	FLOAT	0.~650.00 (小数点第 2 位)	Enable 上升沿时
SlaveScaling	从轴缩放比例	FLOAT	0.~650.00 (小数点第 2 位)	Enable 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	<ul style="list-style-type: none"> Enable 上升沿时 	<ul style="list-style-type: none"> 运动停止时 Error 上升沿时 Aborted 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Busy	执行中	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Enable 下降沿时 被中断时·若 Enable 为 FALSE·下个周期 Aborted 会变成 FALSE
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Enable 下降沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 凸轮表错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Enable 下降沿时
CycleStartF lag	周期开始标志 (保持一个循环)	BOOL	<ul style="list-style-type: none"> 凸轮周期开始 	<ul style="list-style-type: none"> 凸轮周期开始的下个程序周期
InCam	凸轮啮合标志	BOOL	<ul style="list-style-type: none"> 执行中CamOut上升沿 	<ul style="list-style-type: none"> 执行中CamOut下降沿

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	更新型式
Index	凸轮点指针	DWORD	K1 ~ K2047	执行中持续更新
InputPulses	累计主轴输入脉冲个数	DWORD	K-2,147,483,648 ~ K2,147,483,647	执行中持续更新
InputFreq	主轴输入脉冲频率	DWORD	K 0 ~ K 2,147,483,647	执行中持续更新

(*1) Master 输入值定义如下

输入值	定义
0	手摇轮
1~16	运动轴 1~16
200	计数器 C200
204	计数器 C204
208	计数器 C208
212	计数器 C212
216	计数器 C216
220	计数器 C220

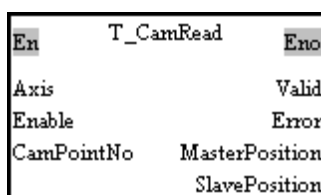
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.19 凸轮数据点读取



1. 功能块说明

此运动功能块，目的在读取凸轮曲线的指定数据点，利用 Axis 输入决定凸轮曲线属于的运动轴，CamPointNo 的输入值决定凸轮的数据点编号，而输出的 MasterPosition 显示此数据点的主轴位置，SlavePosition 显示此数据点的从轴位置。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~16	Enable 上升沿时
Enable	上升沿时启动读取	BOOL	TRUE / FALSE	-
CamPointNo	凸轮数据点位置	DWORD	K0~2046	执行中持续更新

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	● Enable 上升沿时	● Enable 下降沿时 ● Error 上升沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Enable 下降沿时 ● Error 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
MasterPosition	主轴位置	DWORD	K -2,147,483,647 ~ K 2,147,483,647	执行中持续更新
SlavePosition	从轴位置	DWORD	K -2,147,483,647 ~ K 2,147,483,647	执行中持续更新

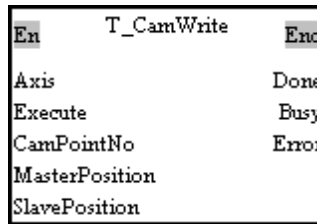
3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.20 凸轮数据点写入



1. 功能块说明

此运动功能块，目的在修正凸轮曲线的指定数据点，利用 Axis 输入决定凸轮曲线属于的运动轴，CamPointNo 的输入值决定凸轮的数据点编号，而输入的 MasterPosition 显示此数据点修改的主轴位置，SlavePosition 显示此数据点修改的从轴位置。

注意事项：若要修正凸轮曲线的所有数据点，请于最后一笔数据结束后再加上 (0·0)。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~16	Execute 上升沿时
Execute	上升沿时 启动更新	BOOL	TRUE / FALSE	-
CamPointNo	凸轮数据点编号	DWORD	K0~2046	Execute 上升沿时
MasterPosition	主轴位置	DWORD	K -2,147,483,647 ~ K 2,147,483,647	Execute 上升沿时
SlavePosition	从轴位置	DWORD	K -2,147,483,647 ~ K 2,147,483,647	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	● 写入完成时	● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	● Error 上升沿时 ● Done 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 ● 启动时加工轴已在凸轮状态 	<ul style="list-style-type: none"> ● Execute 下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.21 凸轮同步速率计算

En	T_CamSyncRatio	Eno
Execute		Done
M360Length		Busy
M360Pulse		Error
S360Length		MRatio
S360Pulse		SRatio
		SyncRatio

1. 功能块说明

此运动功能块，目的在计算主轴同步比例，利用输入 M360Length(主轴物理量)和 M360Pulse(主轴脉冲数)与输入 S360Length(从轴的物理量)和 S360Pulse(从轴脉冲数)。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	同步速率计算允许	BOOL	TRUE / FALSE	-
M360Length	主轴周期长度	DWORD	K1~ K2,147,483,647	Execute 上升沿时
M360Pulse	主轴周期脉冲数	DWORD	K1~ K2,147,483,647	Execute 上升沿时
S360Length	从轴周期长度	DWORD	K1~ K2,147,483,647	Execute 上升沿时
S360Pulse	从轴周期脉冲数	DWORD	K1~ K2,147,483,647	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	计算完成	BOOL	<ul style="list-style-type: none"> ● 执行完成时 	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Execute 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 ● 启动时加工轴已在运动状态 	<ul style="list-style-type: none"> ● Execute 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
MRatio	主轴长度与脉冲之比率	DWORD	K -2,147,483,647 ~ K 2,147,483,647	Done 上升沿时
SRatio	从轴长度与脉冲之比率	DWORD	K -2,147,483,647 ~ K 2,147,483,647	Done 上升沿时
SyncRatio	主从轴同步比例	DWORD	K -2,147,483,647 ~ K 2,147,483,647	Done 上升沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

5.10.22 凸轮曲线规划

En	T_CamCurve	Eno
Axis		Done
Execute		Busy
MLength		Error
SLength		ErrNo
SSyncLength		SyncBegin
SSyncRatio		SyncEnd
SMaxRatio		
AccCurve		
eCamCurve		
Concatenate		

1. 功能块说明

此运动功能块，旨在规划凸轮曲线，利用 Axis 输入决定规划凸轮曲线属于的运动轴，在规划过程中 MLength、SLength、SSyncRatio、SMaxRatio，分别为产生凸轮曲线所需的物理量，AccCurve、eCamCurve 决定生成凸轮曲线的种类。详细信息可参考第 8.4.2 节说明。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~16	Execute 上升沿时
Execute	上升沿时启动更新	BOOL	TRUE / FALSE	
MLength	主轴长度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
				时
SLength	从轴长度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
SSyncLength	从轴同步区长度	DWORD	K1 ~ K2,147,483,647	Execute 上升沿时
SSyncRatio	从轴同步比率	Float	$1.1755 \times 10^{-38} \sim 3.4028 \times 10^{+38}$	Execute 上升沿时
SMaxRatio	从轴最高比率限制	Float	$1.1755 \times 10^{-38} \sim 3.4028 \times 10^{+38}$	
AccCurve	加速曲线	WORD	0~3 (*1)	Execute 上升沿时
eCamCurve	CAM 曲线	WORD	0~5 (*2)	
Concatenate	接续生成	BOOL	TRUE / FALSE	Execute 上升沿时

5

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	曲线规划完成	BOOL	● 执行完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE · 下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Done 上升沿时 ● Error 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
ErrNo	错误码	WORD	0~2	执行中持续更新
SyncBegin	同步起点	DWORD	K 0 ~ K 2,147,483,647	执行中持续更新
SyncEnd	同步终点	DWORD	K 0 ~ K 2,147,483,647	执行中持续更新

(*1) AccCurve 输入值定义如下

输入值	定义
0	等速度曲线
1	等加速度曲线
2	SingleHypot 曲线
3	摆线

(*2) eCamCurve 输入值定义如下

输入值	定义
0	leftCAM
1	midCAMall
2	midCAMbegin
3	midCAMend
5	rightCAM

3. 错误产生及排除

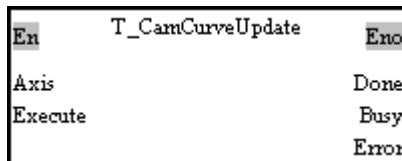
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.10.23 凸轮曲线更新

5



1. 功能块说明

此运动功能块，目的在让下一个凸轮周期时将凸轮表更新成 CamCurve 规划的凸轮曲线，利用 Axis 输入决定更新凸轮曲线属于的运动轴。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~16	Execute 上升沿时
Execute	上升沿时启动更新	BOOL	TRUE / FALSE	

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	完成动态变更凸轮曲线数据	BOOL	● 执行完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE。

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Execute 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 	<ul style="list-style-type: none"> ● Execute 下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块：

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.11 多轴运动功能块说明

5.11.1 G码参数设定

En	T_GcodeSetting	Eno
Execute		Done
ContIP		Busy
VelPercentage		Error

1. 功能块说明

此运动功能块，旨在设定 G 码行程中的参数值，利用输入 ContIP 配置连续插补的最低减速速度，G 码连续行程中当速度低于此值，以此数值运行，VelPercentage 设定运转速度比率。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时设定 Gcode 参数	BOOL	TRUE / FALSE	-
ContIP	设定最大减速速度	DWORD	k0 ~ k500000	Execute 上升沿时
VelPercentage	运行速度百分比设定	WORD	k0~k65,535	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	参数设定完成	BOOL	<ul style="list-style-type: none"> ● 执行完成时 	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Execute 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 	<ul style="list-style-type: none"> ● Execute 下降沿时

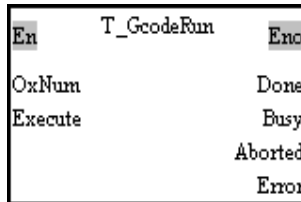
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.11.2 G码运行



1. 功能块说明

此运动功能块，目的在设定 OXn 并执行其运动副程序，利用输入 Oxn 设定执行的编号。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
OxNum	输入 OX 运动程序编号	WORD	OX0~OX99:0~99 SD Card: 100~199	Execute 上升沿时
Execute	上升沿时执行 OX 运动子程序	BOOL	TRUE / FALSE	-

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	● 运动完成时	● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	● Error 上升沿时 ● Done 上升沿时 ● Aborted 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误 ● 启动时加工轴已在运动状态	● Execute 下降沿时
Aborted	功能块被其它命令中断	BOOL	● 执行中被命令停止	● Execute 下降沿时

3. 错误产生及排除

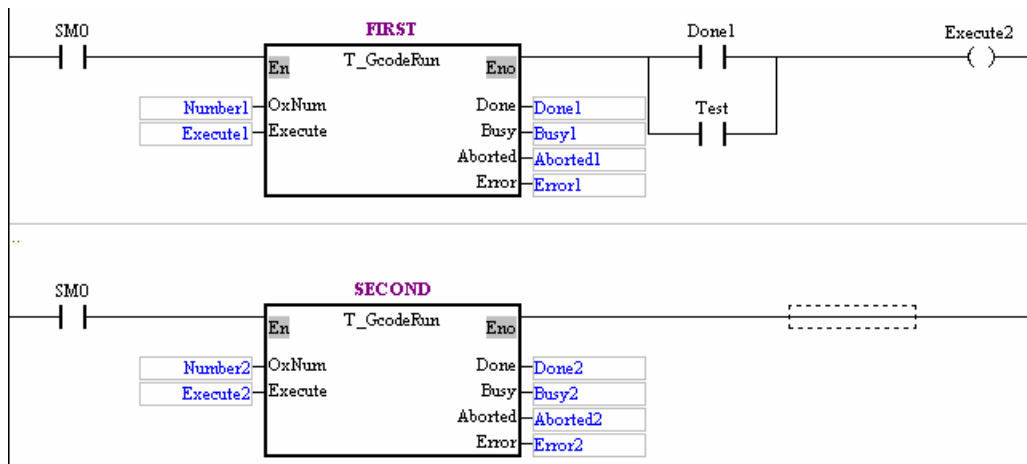
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 范例程序

此范例的目的在

- 第一个 G 码运动完成后，接连执行第二个 G 码运动。
- 第一个 G 码运动尚未完成前，先执行第二个 G 码运动。

首先设定运动功能块 FIRST 和运动功能块 SECOND 为启动两个不同的 Ox 作 G 码运动。

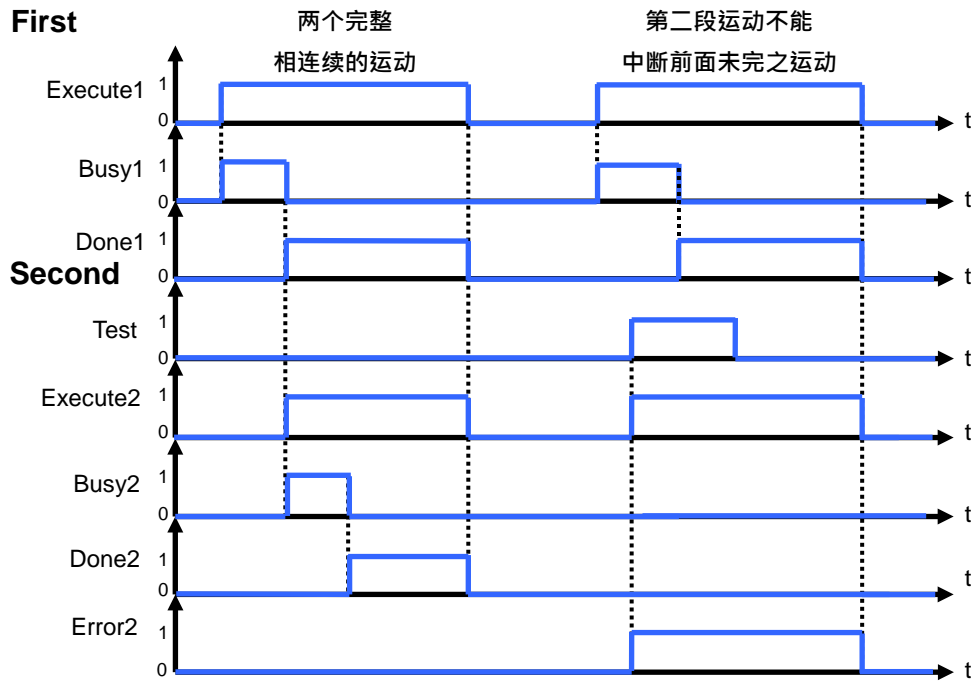


- 第一个 G 码运动完成后，接连执行第二个 G 码运动。
操作步骤：
(a) 启动 Execute1
(b) 等待 Done2 或 Error2 上升沿产生
- 第一个 G 码运动尚未完成前，预先执行第二个 G 码运动。
操作步骤：
(a) 启动 Execute1
(b) 在 Busy1 为 True 时，启动 Test。
(c) 等待 Done2 或 Error2 上升沿产生

两种操作结果其时序图与第 1 轴运动结果如下：

5

Number1 = Number2

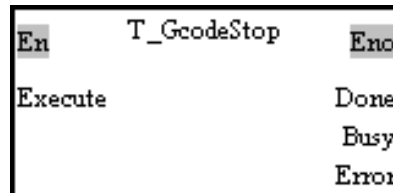


5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

5.11.3 G码停止



1. 功能块说明

此运动功能块，目的在停止 OXn 运动副程序。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时执行停止 G 码	BOOL	TRUE / FALSE	-

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	G 码运行停止完成	BOOL	● 执行完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

3. 错误产生及排除

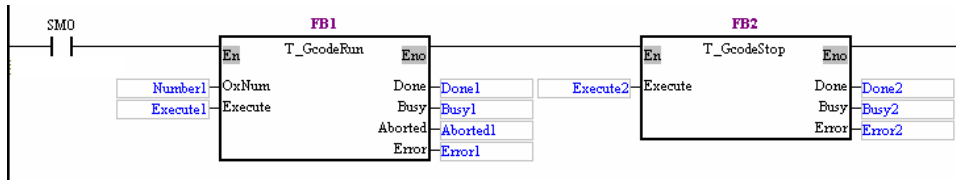
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 范例程序

此范例的目的在

- 第一个 G 码运动开始运行后，强制停止 G 码运动。

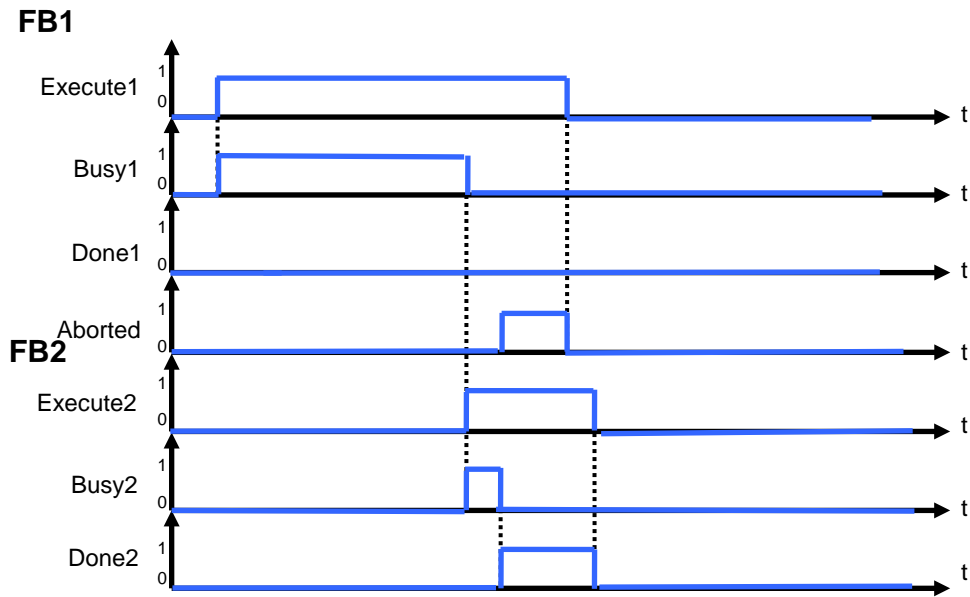
首先设定运动功能块 FB1 启动 Ox 作 G 码运动，运动功能块 FB2 为停止 Ox 作 G 码运动。



操作步骤：

- 利用 Execute1 启动 FB1
- 运行指定的 Ox 内的 G 码
- 在 G 码尚未完成，启动 FB2 的 Execute2
- 强制停止 G 码动作，同时 FB1 的 Aborted 会被设置为 ON

引脚关系图如下：



5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.11.4 M码读取

5

En	T_Mcode	Eno
Enable		Valid
CLRMcoding		Busy
		Error
		Value

1. 功能块说明

此运动功能块，旨在读取 M 码并且可清除指定 M 码编号，利用输入 CLR M 码清除指定的 M 码编号。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Enable	M 码读取允许	WORD	TRUE/FALSE	-
CLR M coding	在 Enable=On 状态下上升沿时清除目前 M code	BOOL	TRUE/FALSE	执行中

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	<ul style="list-style-type: none"> M code 启动时 	<ul style="list-style-type: none"> Enable 下降沿 Valid=ON 时 CLR M 码上升沿
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Error 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 	<ul style="list-style-type: none"> Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
Value	当 Valid=ON 时，显示目前使用中的 M code	WORD	K0 ~ 4096	当 Valid=ON 时持续更新数值

5

3. 错误产生及排除：

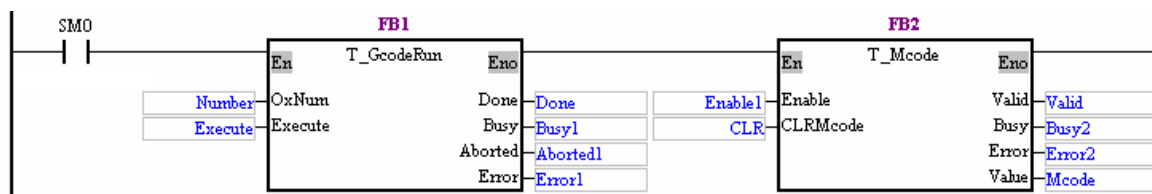
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 范例程序：

此范例的目的在

- 第一个 G 码运动开始运行后，利用 M 码运动功能块，检查 M 码运行状态，若运行至 M 码部分，则利用功能块输入清除 M 码。

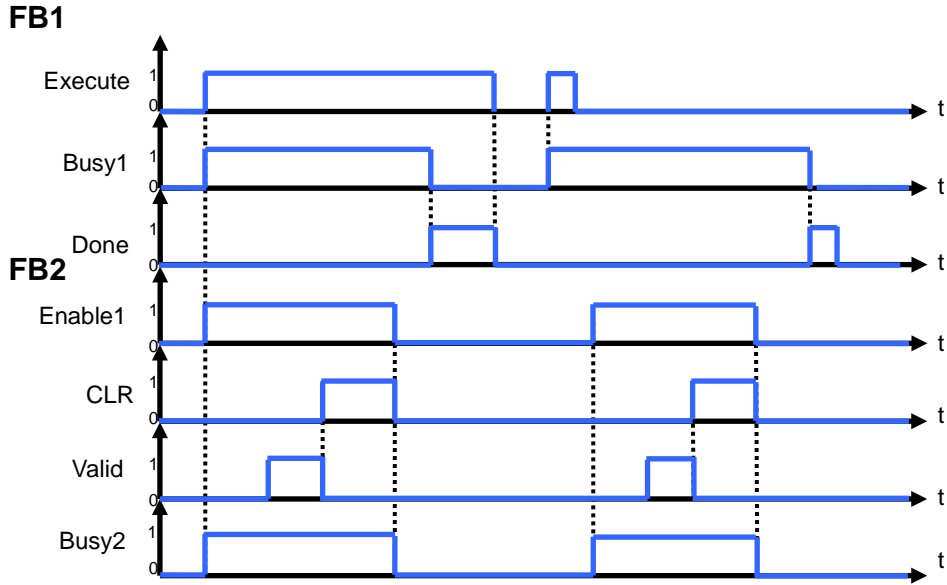
首先设定运动功能块 FB1 启动 Ox 作 G 码运动，运动功能块 FB2 为检查当前 M 码状态。



操作步骤：

- 利用 Execute 启动 FB1
- 运行指定的 Ox 内的 G 码
- 在 G 码尚未完成后，启动 FB2 的 Enable1
- 观察目前 M 码状态
- 在执行 G 码过程中运行至 M 码，则 FB2 的 Valid 为 ON
- 此时可利用 CLR 清除 M 码

引脚关系图如下：



5. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

5.11.5 多轴绝对直线插补

En	T_AbsMoveLi-	Eno
AxisGroup		Done
Execute		Busy
Position		Error
Velocity		Aborted

1. 功能块说明

此运动功能块，目的在启动指定多轴的绝对位置插补群组，利用输入 **AxisGroup** 设定执行的编号，**Position** 设定群组中每一轴目标的绝对位置，**Velocity** 设定此群组的插补速度。

输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
AxesGroup	插补轴群组	WORD[6]	[, , , , , ,] 0:不设定 n:加入第 n(1~16)轴 (第一栏须设定)	Execute 上升沿
Position	目标位置群组	DWORD[6]	[, , , , , ,] K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿
Velocity	插补速度设定	DWORD	K1~ K2,147,483,647	Execute 上升沿
输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	● 执行完成时	● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE · 下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时
Aborted	功能块被其它命令中断	BOOL	● 执行中被命令停止	● Execute 下降沿时

2. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

3. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.11.6 多轴相对直线插补

En	T_RelMoveLi~	Eno
AxesGroup		Done
Execute		Busy
Distance		Error
Velocity		Aborted

1. 功能块说明

此运动功能块，目的在启动指定多轴的相对距离插补群组，利用输入 **AxesGroup** 设定执行的编号，**Distance** 设定群组中每一轴相对当前的相对距离，**Velocity** 设定此群组的插补速度。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
AxesGroup	插补轴群组	WORD[6]	[, , , , , ,] 0:不设定 n:加入第 n(1~16)轴 (第一栏须设定)	Execute 上升沿
Distance	目标位移群组	DWORD[6]	[, , , , , ,] K-2,147,483,648 ~ K2,147,483,647	Execute 上升沿
Velocity	插补速度设定	DWORD	K1~ K2,147,483,647	Execute 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	● 执行完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时
Aborted	功能块被其它命令中断	BOOL	● 执行中被命令停止	● Execute 下降沿时

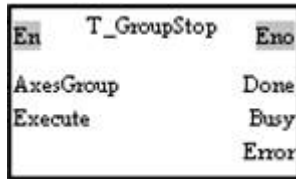
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.11.7 多轴直线插补停止



1. 功能块说明

此运动功能块，目的在停止指定多轴的插补群组，利用输入 AxesGroup 设定执行的编号。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时命令直线插补停止	BOOL	TRUE / FALSE	-
AxesGroup	插补轴停止群组（一般与启动群组相同）	WORD[6]	[-, -, -, -, -, -] 0:不设定 n:加入第 n(1~16)轴 (第一栏须设定)	Execute 上升沿

5

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	<ul style="list-style-type: none"> ● 执行完成时 	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Execute 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 	<ul style="list-style-type: none"> ● Execute 下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

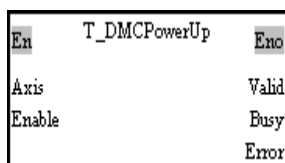
4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5

5.12 网络功能块说明

5.12.1 伺服启动/停止



1. 功能块说明

此运动功能块，目的在 DMCNET 网络下启动或停止指定的伺服，利用输入 Axis 设定执行的轴编号。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~12	Execute 上升沿
Enable	上升沿时 Servo ON, 下降沿时 Servo Off	BOOL	TRUE / FALSE	-

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之状态有效	BOOL	● Enable 上升沿时	● Error 上升沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Enable 下降沿时 ● Error 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误时	● Enable 下降沿时

3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH20MC-5A 此种 AH500 机种之运动控制模块。

5.12.2 伺服重置



1. 功能块说明

此功能块仅能用于网络出现异常时。在使用「伺服重置」功能块将网络重置后，重置完成时需再使用「伺服 DMCNET 通讯初始化」功能块让控制器与伺服回到联机模式。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~12	Execute 上升沿
Execute	上升沿时开始重置	BOOL	TRUE / FALSE	-

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	伺服重置完成	BOOL	<ul style="list-style-type: none"> ● 执行完成时 	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Execute 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 	<ul style="list-style-type: none"> ● Execute 下降沿时

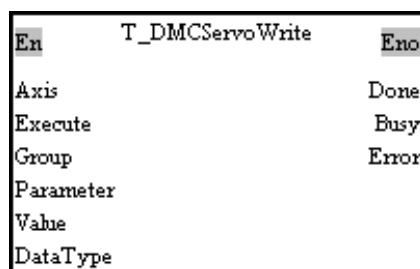
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH20MC-5A 此种 AH500 机种之运动控制模块。

5.12.3 伺服参数写入



1. 功能块说明

此运动功能块，目的在 DMCNET 网络下对指定的伺服写入指定的参数，利用输入 Axis 设定执行的轴编号，Group 为伺服参数的群组代码，Parameter 为指定参数的参数代码，DataType 为此参数的数据类型，Value 为写入值。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~12	Execute 上升沿
Execute	上升沿时开始设定	BOOL	TRUE / FALSE	-
Group	A2 伺服参数-群组代码 K0~K9(参考 A2 手册)	WORD	0~9	Execute 上升沿
Parameter	A2 伺服参数-参数代码 K0~K99 (参考 A2 手册)	WORD	0~99	Execute 上升沿
Value	设定参数内容 (参考 A2 手册)	DWORD	K-2,147,483,647~ 2,147,483,647	Execute 上升沿
DataType	设定数据类型	BOOL	mc16bits: FALSE mc32bits: TRUE	Execute 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	设定完成	BOOL	● 写入完成时	● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

3. 错误产生及排除

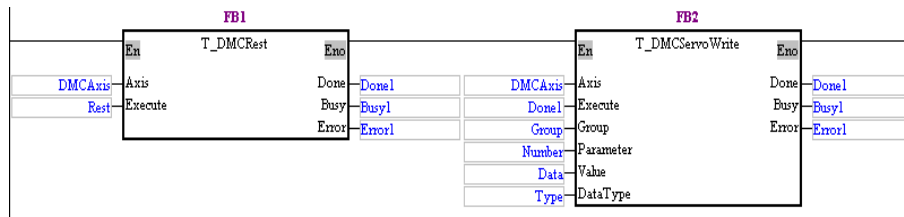
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 范例程序

此范例的目的在

- 利用运动功能块 **DMCRest** 对指定伺服作 **DMC** 通讯重置后，使用伺服参数写入功能块将伺服指定参数写入设定值。

利用运动功能块 **FB1** 指定伺服作重置动作，运动功能块 **FB2** 的 **Group** 写入伺服参数群组，**Parameter** 写入伺服参数代码，**Value** 写入参数数值。

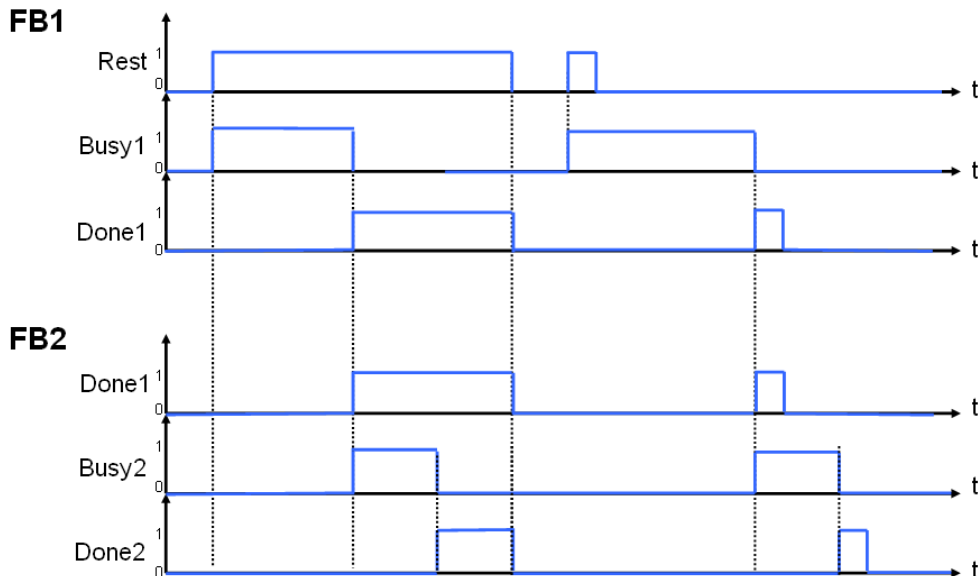


启动 **FB1** 将指定的伺服作重置动作，在重置动作完成后，将自动执行 **FB2**，在 **FB2** 完成后 **Done2** 自动为 **ON**，此时设定值 **Data** 已被写入伺服内部，其信号关系图如下：

操作步骤：

- 启动 **FB1** 将指定的伺服作重置动作
- 重置动作完成后，将自动执行 **FB2**
- FB2** 完成后 **Done1** 为 **ON**，此时设定值 **Data** 已被写入伺服内部

其信号引脚关系图如下：



5. 支持模块

此运动功能块支持 **AH20MC-5A** 此种 **AH500** 机种之运动控制模块。

5.12.4 伺服参数读取

En	T_DMCServoRead	Eno
Axis		Done
Execute		Busy
Group		Error
Parameter		Value

1. 功能块说明

此运动功能块，目的在 DMCNET 网络下对指定的伺服读出指定的参数内容，利用输入 Axis 设定执行的轴编号，Group 为伺服参数的群组代码，Parameter 为指定参数的参数代码，Value 为读出值。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~12	Execute 上升沿
Execute	上升沿时开始读取	BOOL	TRUE / FALSE	-
Group	伺服参数群组代码	WORD	0~9	Execute 上升沿
Parameter	伺服参数参数代码	WORD	0~99	Execute 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	读取完成	BOOL	● 读取完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
Value	参数内容	DWORD	K-2,147,483,647~2,147,483,647	Done 上升沿时

3. 错误产生及排除

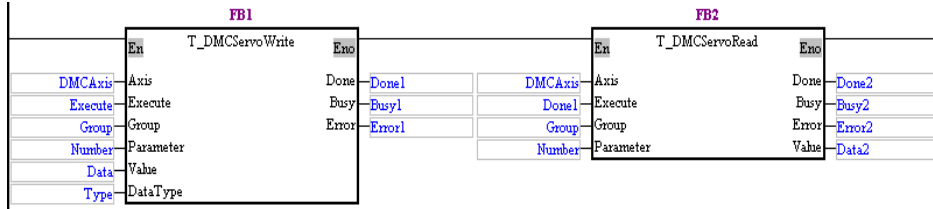
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 范例程序

此范例的目的在

- 利用功能块参数写入对指定伺服作参数写入后，使用伺服参数读出功能块将写入的参数读回其内容。

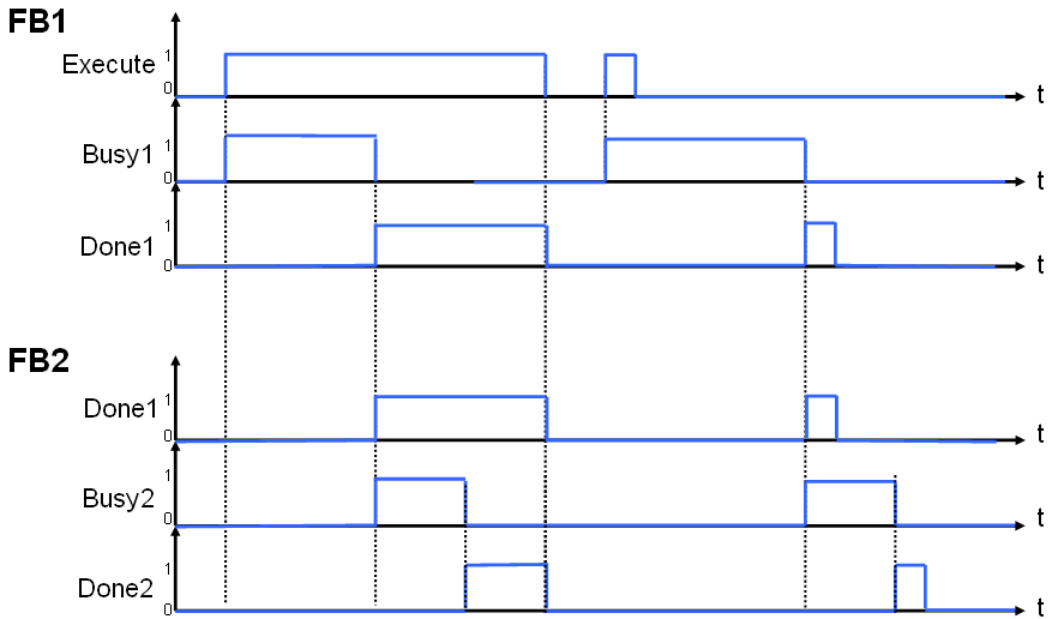
运动功能块 **FB1** 指定伺服作参数写入后，利用运动功能块 **FB2** 读回该参数，**Group** 和 **Number** 分别为参数群组 and 参数代码



操作步骤：

- 启动 **FB1** 将指定的伺服作参数写入
- 写入动作完成后，将自动执行 **FB2**
- FB2** 完成后 **Done2** 为 ON，此时 **Data2** 为读回的参数数值
- Data2** 和 **Data** 应为相同数值

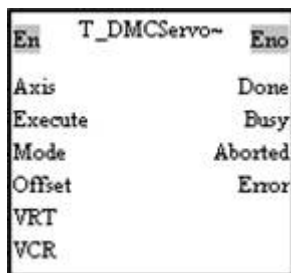
其信号关系图如下：



5. 支持模块

此运动功能块支持 AH20MC-5A 此种 AH500 机种之运动控制模块。

5.12.5 伺服回原点命令



1. 功能块说明

此运动功能块，目的在 DMCNET 网络下，对指定的伺服下达原点回归运动，利用输入 Axis 设定执行的轴编号，Mode 为原点回归方式，Offset 为原点回归完成后的当前位置设定值。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~12	Execute 上升沿
Execute	上升沿时启动运动	BOOL	TRUE / FALSE	-
Mode	回原点模式	WORD	1~35	Execute 上升沿
Offset	回原点偏移量	WORD	K-32,767 ~ 32,767	Execute 上升沿
VRT	原点回归速度 单位：RPM	DWORD	K1 ~ K2000	Execute 上升沿 时
VCR	原点回归减速速度 单位：RPM	DWORD	K1 ~ 500	Execute 上升沿 时
输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	● 执行完成时	● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	● Error 上升沿时 ● Done 上升沿时
Aborted	功能块被其他命令中断	BOOL	● 执行中被命令停止	● Execute 下降沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

3. 错误产生及排除

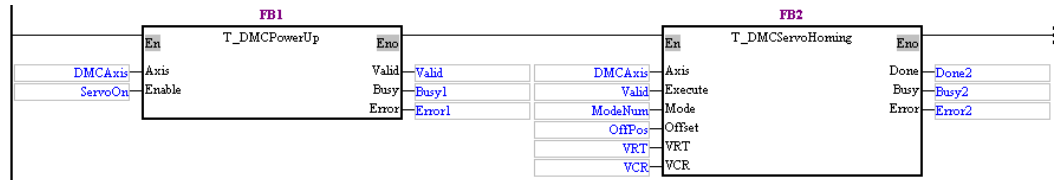
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 范例程序

此范例的目的在

- 利用运动功能块 **DMCPowerUp** 对指定伺服 **Servo ON** 动作后，使用原点回归功能块将伺服以指定方式启动原点回归。

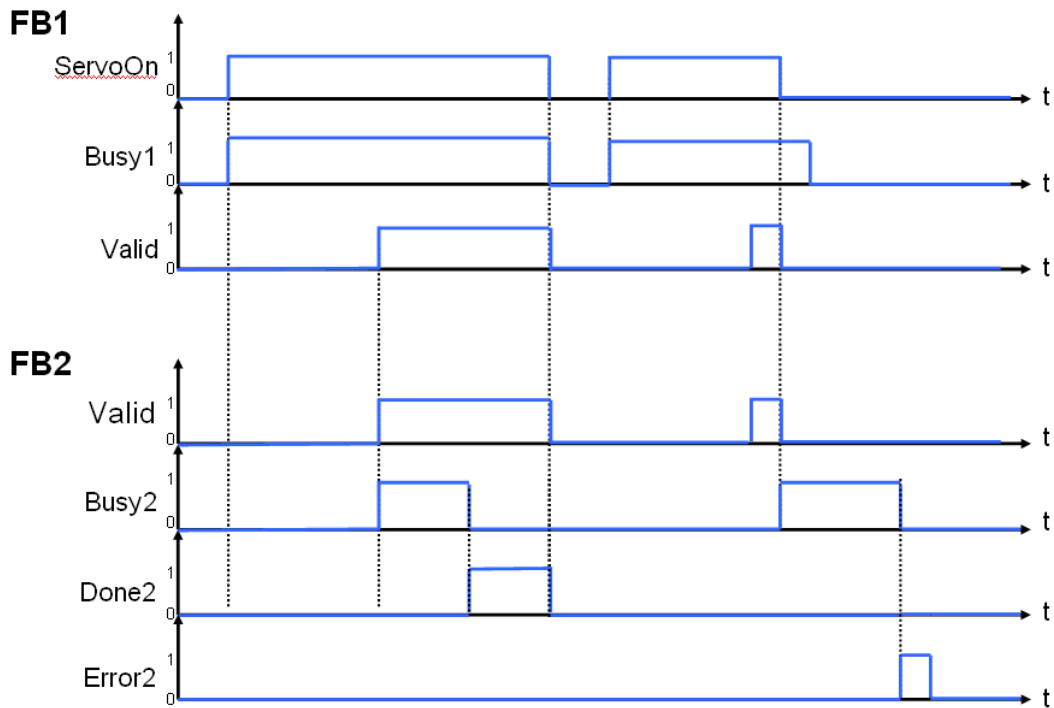
运动功能块 **FB1** 指定伺服 **Servo ON** 后，利用运动功能块 **FB2** 的输入原点回归的方式，命令伺服作原点回归的动作。



操作步骤：

- 功能块 **FB1** 完成后，伺服为 **Servo ON** 状态
- 将自动执行 **FB2**
- 以 **ModeNum** 决定原点回归方式

其信号引脚关系图如下：



5. 支持模块

此运动功能块支持 AH20MC-5A 此种 AH500 机种之运动控制模块。

5.12.6 伺服DMCNET通讯初始化

En	T_DMCControlInit	Eno
Axis		Done
Execute		Busy
DMC_RatioNum		Error
DMC_RatioDen		

1. 功能块说明

此运动功能块，目的在 DMCNET 网络下对指定的伺服下达通讯初始化，利用输入 Axis 设定执行的轴编号，DMC-RatioNum 和 DMC-RatioDen 为设定伺服参数中的电子齿轮比。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	1~12	Execute 上升沿
Execute	上升沿时开始初始化	BOOL	TRUE / FALSE	-
DMC_RatioNum	伺服电子齿轮比分子	WORD	K1 ~32,767	Execute 上升沿
DMC_RatioDen	伺服电子齿轮比分母	WORD	K1 ~ 32,767	Execute 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	执行完成	BOOL	● 执行完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

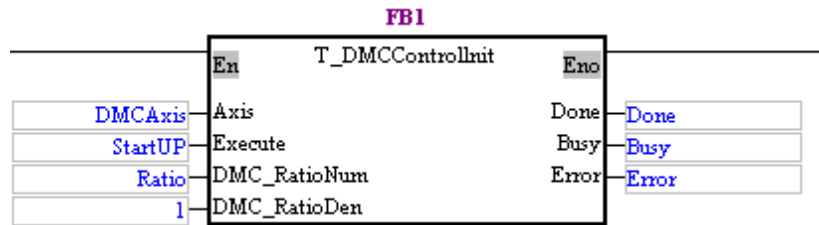
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

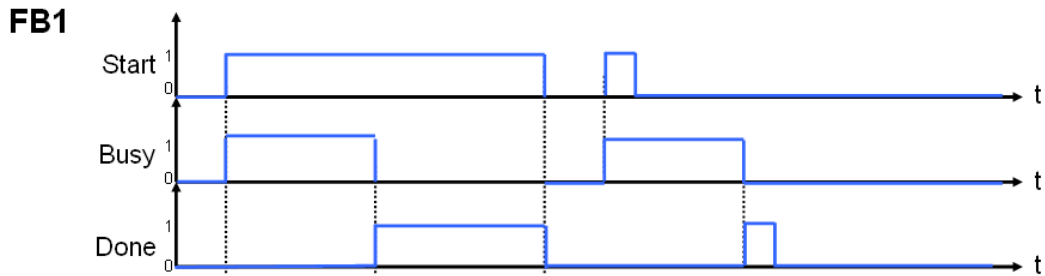
4. 范例程序

此范例的目的

- 利用运动功能块 DMCNET 通讯初始化，对伺服作完整的通讯初始化动作，包含伺服电子齿轮比参数的设定。



利用功能块可设定伺服的电子齿轮比参数，并且在功能块完成后可透过单轴或多轴的运动功能块开始命令伺服运动，此范例的信号图如下：

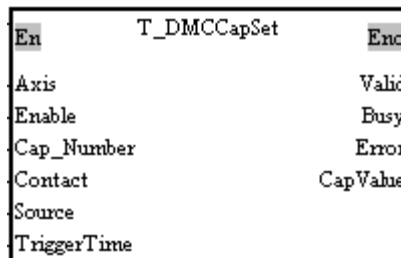


5. 支持模块

此运动功能块支持 AH20MC-5A 此种 AH500 机种之运动控制模块。

5

5.12.7 伺服捕捉功能



1. 功能块说明

此运动功能块，目的在 DMCNET 网络下对指定的伺服启动捕捉功能，利用输入 Axis 设定执行的轴编号 CAP_Number contact source triggerTime 为设定伺服捕捉功能参数 输出 CapValue 为捕捉到的数值。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Axis	运动轴编号	WORD	K1~K12	Enable 上升沿
Enable	上升沿时开始启动捕捉	BOOL	TRUE / FALSE	-
Cap_Number	捕捉数量设定	WORD	K1~K400	Enable 上升沿
Contact	捕捉信号设定	BOOL	TRUE / FALSE	Enable 上升沿
Source	捕捉值来源	WORD	K0~K3 (*1)	Enable 上升沿
TriggerTime	触发最小时间(msec)	WORD	K0~K1000	Enable 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	● Enable 上升沿时	● Enable 下降沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错误	BOOL	● 输入参数错误 ● 指定来源已被占用	● Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
CapValue	捕捉值	DWORD	K0~2,147,483,647	当 Valid=ON 时持续更新数值

(*1) source

输入值定义如下

输入值	定义
0	不作用
1	光学尺
2	运动命令脉冲
3	主编码器

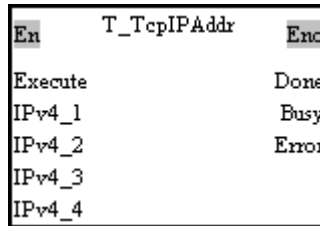
3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH20MC-5A 此种 AH500 机种之运动控制模块。

5.12.8 Ethernet IP Address 设定



1. 功能块说明

此运动功能块，目的为设定模块的 Ethernet IP Address，利用输入 IPv4_1、IPv4_2、IPv4_3、IPv4_4，决定 IP 的四码。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时写入参数	BOOL	TRUE / FALSE	-
IPv4_1	IP Address 第一码字符	WORD	K0~255	Execute 上升沿时
IPv4_2	IP Address 第二码字符	WORD	K0~255	Execute 上升沿时
IPv4_3	IP Address 第三码字符	WORD	K0~255	Execute 上升沿时
IPv4_4	IP Address 第四码字符	WORD	K0~255	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	写入完成	BOOL	● 写入完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

3. 错误产生及排除：

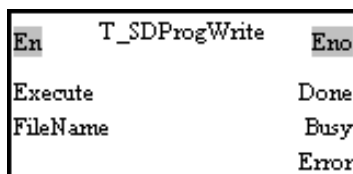
错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH10PM-5A、AH15PM-5A、AH20MC-5A 此三种 AH500 机种之运动控制模块。

5.13 其它功能块说明

5.13.1 SD主程序备份



1. 功能块说明

此运动功能块，目的在将主程序备份到 SD 卡中，利用输入 FileName 设定存放的文件名称。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时发启动主程序备份	BOOL	TRUE / FALSE	-
FileName	设定主程序备份至 SD Card 中文件名称	WORD	K0~4095	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	写入完成	BOOL	● 写入完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

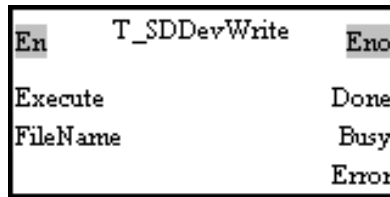
3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH10PM-5A、AH15PM-5A、AH20MC-5A 此三种 AH500 机种之运动控制模块。

5.13.2 SD装置备份



1. 功能块说明

此运动功能块，目的在将装置备份到 SD 卡中，利用输入 FileName 设定存放的文件名称

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时启动装置备份	BOOL	TRUE / FALSE	-
FileName	设定装置备份至 SD Card 中文件名称	WORD	K0~4095	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	备份完成	BOOL	● 动作完成时	● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	● Error 上升沿时 ● Done 上升沿时
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH10PM-5A、AH15PM-5A、AH20MC-5A 此三种 AH500 机种之运动控制模块。

5.13.3 SD装置回存

En	T_SDevRead	Eno
Execute		Done
FileName		Busy
Device		Error
Begin		
End		

1. 功能块说明

此运动功能块，旨在将指定装置从 SD 卡中指定的文件中读出，利用输入 **FileName** 设定文件的名称，**Device** 设定装置种类，**Begin** 和 **End** 为此装置的开始与结束位置。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Execute	上升沿时启动装置回存	BOOL	TRUE / FALSE	-
FileName	指定 SD Card 中文文件名称	WORD	K0~4095	Execute 上升沿时
Device	选择回存装置	WORD	mcSD_M:M 装置:0 mcSD_D:D 装置:5 mcSD_W:W 装置:6	Execute 上升沿时
Begin	装置回存开始范围	WORD	M: K0~4,096 D: K0~9,999 W: K0~65,535	Execute 上升沿时
End	装置回存结束范围	WORD	M:K 0~4,096 D: K0~9,999 W: K0~65,535	Execute 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	动作完成	BOOL	● 动作完成时	<ul style="list-style-type: none"> ● Execute 下降沿时 ● 定位完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	● Execute 上升沿时	<ul style="list-style-type: none"> ● Error 上升沿时 ● Done 上升沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Error	功能块产生错误	BOOL	● 输入参数错误	● Execute 下降沿时

3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH10PM-5A、AH15PM-5A、AH20MC-5A 此三种 AH500 机种之运动控制模块。

5.13.4 高速计数器

En	T_HCnt	Eno
Channel		Valid
Enable		Busy
ExtRstEN		Error
InputType		CountValue
InitialValue		

1. 功能块说明

此运动功能块，目的为启动高速计数器，利用输入 Channel 决定计数器，InputType、ExtRstEN、InitialValue 为计数器的参数设定，输出 CountValue 则为计数中所得的计数值。

2. 输入/输出说明：

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Channel	设定计数器组别	WORD	0~5 (*1)	Enable 上升沿
Enable	计数允许	BOOL	TRUE / FALSE	-
ExRstEN	外部重置允许	BOOL	TRUE / FALSE	Enable 上升沿
InputType	输入脉冲型式	WORD	mcUD:0 mcPD:1 mcAB:2 mc4AB:3	执行中持续更新
ExternalRst	外部重置开关	BOOL	TRUE / FALSE	Enable 上升沿
InitialValue	计数器初始值设定	DWORD	K0~2,147,483,647	Enable 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	● Enable 上升沿时	● Enable 下降沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Error 上升沿时 ● Enable 下降沿时

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 ● 指定来源已被占用 	<ul style="list-style-type: none"> ● Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
CountValue	计数值	DWORD	K0~2,147,483,647	当 Valid=ON 时持续更新数值

(*1) Channel 输入值定义如下

输入值	定义	硬件引脚
0	C200	X0.8, X0.9
1	C204	X0.10, X0.11
2	C208	X0.12, X0.13
3	C212	X0.14, X0.15
4	C216	X0.12, X0.13
5	C220	X0.14, X0.15

5

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A、AH20MC-5A 此四种 AH500 机种之运动控制模块。

5.13.5 高速定时器

En	T_HTimer	Eno
Channel		Valid
Enable		Busy
TriggerMode		Error
		TimerValue

1. 功能块说明

此运动功能块，目的为启动高速定时器，利用输入 Channel 决定定时器，TriggerMode 为定时器的参数设定，输出 TimerValue 则为计时中所得的计时值

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Channel	设定定时器组别	WORD	0~3 (*1)	Enable 上升沿
Enable	计时允许	BOOL	TRUE / FALSE	-
TriggerMode	计时触发模式	BOOL	mcUp_Down: FALSE mcUp_Up: TRUE	执行中持续更新

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	● Enable 上升沿时	● Enable 下降沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错误	BOOL	● 输入参数错误 ● 指定来源已被占用	● Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
TimerValue	计时值	DWORD	K0~2,147,483,647	执行中持续更新数值·若无触发事件·计时值会维持上次计时值

(*1) Channel 输入值定义如下

输入值	定义	硬體引脚
0	C200	X0.0
1	C204	X0.1
2	C208	X0.2
3	C212	X0.3

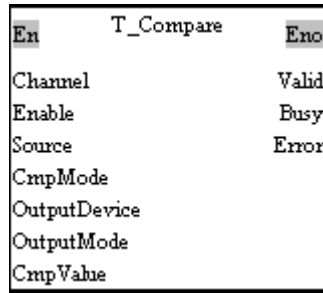
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值·是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 此四种 AH500 机种之运动控制模块。

5.13.6 高速比较设定



1. 功能块说明

此运动功能块，目的为启动高速比较功能，利用输入 Channel 决定比较器组别，Source 为比较器来源，CmpMode、OutputDevie 为比较器的参数设定，比较成立时输出 Status

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Channel	设定比较器组别	WORD	0~7	Enable 上升沿时
Enable	上升沿触发执行此功能块	BOOL	TRUE / FALSE	-
Source	设定比较来源	WORD	mcCmpAxis1 (0) : 轴 1 当前位置 mcCmpAxis2(1): 轴 2 当前位置 mcCmpAxis3(2): 轴 3 当前位置 mcCmpAxis4(3): 轴 4 当前位置 mcCmpC200 (4):C200 当前值 mcCmpC204 (5) : C204 当前值 mcCmpC208 (6) : C208 当前值 mcCmpC212 (7) : C212 当前值	Enable 上升沿时
CmpMode	设定比较方式	WORD	0: = 1: ≥ 2: ≤	Enable 上升沿时
OutputDevice	当比较条件成立时输出装置	WORD	mcCmpY8(0): Y0.8 mcCmpY9 (1) : Y0.9	Enable 上升沿时

5

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
			mcCmpY10 (2) : Y0.10 mcCmpY11 (3) : Y0.11 mcCmpRstC200(4) : C200 mcCmpRstC204(5) : C204 mcCmpRstC208(6) : C208 mcCmpRstC212(7) : C212	
OutputMode	设定输出装置输出方式	BOOL	mcCmpSet:TRUE mcCmpRst:FALSE	Enable 上升沿
CmpValue	设定比较值	DWORD	K-2,147,483,647~K2,147,483,647	Enable 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	<ul style="list-style-type: none"> ● Enable 上升沿时 	<ul style="list-style-type: none"> ● Enable 下降沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Enable 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 ● 指定来源已被占用 	<ul style="list-style-type: none"> ● Enable 下降沿时

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围
此组别已被使用	更改使用组别

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.13.7 高速比较重置

En	T_CmpOutRst	Eno
Enable		Valid
CLR_Y08		CMP_Y08
CLR_Y09		CMP_Y09
CLR_Y010		CMP_Y010
CLR_Y011		CMP_Y011
CLR_C200Rst		CMP_C200Rst
CLR_C204Rst		CMP_C204Rst
CLR_C208Rst		CMP_C208Rst
CLR_C212Rst		CMP_C212Rst
		Busy

1. 功能块说明

此运动功能块·目的为重置高速比较功能与查看当前比较条件状态·利用 CLR_Y08、CLR_Y09、CLR_Y010、CLR_Y011、CLR_C200Rst、CLR_C204Rst、CLR_C208Rst、CLR_C212Rst 决定清除的输出。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Enable	上升沿触发执行此功能块	BOOL	TRUE / FALSE	-
CLR_Y08	清除比较输出	BOOL	TRUE / FALSE	执行中持续更新
CLR_Y09				
CLR_Y010				
CLR_Y011				
CLR_C200Rst				
CLR_C204Rst				
CLR_C208Rst				
CLR_C212Rst				

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	<ul style="list-style-type: none"> ● Enable 上升沿时 	<ul style="list-style-type: none"> ● Enable 下降沿时
Busy	执行中	BOOL	<ul style="list-style-type: none"> ● Enable 上升沿时 	<ul style="list-style-type: none"> ● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> ● 输入参数错误 ● 指定来源已被占用 	<ul style="list-style-type: none"> ● Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
CMP_Y08	比较输出状态	BOOL	TRUE / FALSE	当 Valid=ON 时持续更新
CMP_Y09				
CMP_Y010				
CMP_Y011				
CMP_C200Rst				
CMP_C204Rst				
CMP_C208Rst				
CMP_C212Rst				

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 四种 AH500 机种之运动控制模块。

5.13.8 高速捕捉设定

En	T_Capture	Eno
Channel		Valid
Enable		Busy
Source		Error
TriggerDevice		CapValue
InitialValue		

1. 功能块说明

此运动功能块，目的为启动高速捕捉功能，利用输入 Channel 决定捕捉器组别，Source 为捕捉信号来源，TriggerDevice、InitialValue 为捕捉的参数设定，捕捉触发时输出 CapValue。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Channel	设定捕捉器组别	WORD	0~7	Enable 上升沿
Enable	上升沿触发执行此功能块	BOOL	TRUE / FALSE	-
Source	设定捕捉来源	WORD	mcCmpAxis1 (0): 轴 1 当前位置 mcCmpAxis2 (1): 轴 2 当前位置 mcCmpAxis3 (2): 轴 3 当	Enable 上升沿

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
			前位置 mcCmpAxis4 (3) : 轴 4 当前位置 前位置 mcCmpC200 (4) : C200 当前值 前值 mcCmpC204 (5) : C204 当前值 前值 mcCmpC208 (6) : C208 当前值 前值 mcCmpC212 (7) : C212 当前值 前值	
TriggerDevice	捕捉触发装置	WORD	mcCapX0 (0) : X0.0 mcCapX1 (1) : X0.1 mcCapX2 (2) : X0.2 mcCapX3 (3) : X0.3 mcCapX8 (8) : X0.8 mcCapX9 (9) : X0.9 mcCapX10 (10) : X0.10 mcCapX11 (11) : X0.11 mcCapX12 (12) : X0.12 mcCapX13 (13) : X0.13 mcCapX14 (14) : X0.14 mcCapX15 (15) : X0.15	Enable 上升沿
InitialValue	捕捉初始值	DWORD	K-2,147,483,648 ~ K2,147,483,647	Enable 上升沿

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有效	BOOL	● Enable 上升沿时	● Enable 下降沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错误	BOOL	● 输入参数错误 ● 指定来源已被占用	● Enable 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	数据有效时机
CapValue	捕捉值	DWORD	K-2,147,483,648 ~ K2,147,483,647	执行中持续更新数值·若无 触发事件·捕捉值会维持上 一次捕捉到的值

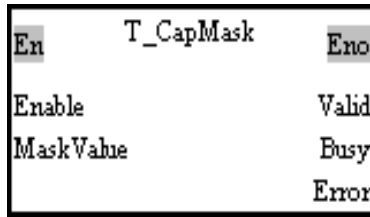
3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值·是否超过可允许范围
此组别已被使用	更改使用组别

4. 支持模块

此运动功能块支持 AH05PM-5A、AH15PM-5A、AH10PM-5A、AH20MC-5A 此四种 AH500 机种之运动控制模块。

5.13.9 高速捕捉遮蔽



1. 功能块说明

此运动功能块，目的为启动高速捕捉遮蔽，利用输入 MaskValue 决定捕捉遮蔽值

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Enable	上升沿触发 执行此功能 块	BOOL	TRUE / FALSE	-
MaskValue	设定捕捉遮 蔽值范围	DWORD	K1~2,147,483,647	执行中持续更新

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	输出之数值有 效	BOOL	● Enable 上升沿时	● Enable 下降沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错 误	BOOL	● 输入参数错误 ● 指定来源已被占 用	● Enable 下降沿时

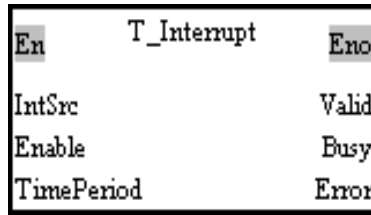
3. 错误产生及排除：

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A、AH20MC-5A 此四种 AH500 机种之运动控制模块。

5.13.10 中断设定



1. 功能块说明

此运动功能块，目的为设定中断程序的触发来源，利用输入 IntSrc 决定中断程序的触发来源，当为时间中断时 TimePeriod 为时间中断的周期。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
IntSrc	设定中断组别	WORD	IntTimer:0 IntX8:1 IntX9:2 IntX10:3 IntX11:4 IntX12:5 IntX13:6 IntX14:7 IntX15:8	Enable 上升沿时
Enable	上升沿触发执行此功能块	BOOL	TRUE / FALSE	-
TimePeriod	设定时间中断周期[ms] (端子中断不使用)	WORD	K1~65,535	执行中持续更新

输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Valid	中断允许	BOOL	● 设定完成并开启中断功能	● Enable 下降沿时
Busy	执行中	BOOL	● Enable 上升沿时	● Error 上升沿时 ● Enable 下降沿时
Error	功能块产生错误	BOOL	● 输入参数错误 ● 指定来源已被占用	● Enable 下降沿时

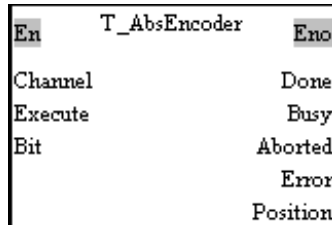
3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块支持 AH05PM-5A、AH10PM-5A、AH15PM-5A 及 AH20MC-5A 此四种 AH500 机种之运动控制模块。

5.13.11 绝对型编码器



1. 功能块说明

此运动功能块，目的为设定与启动绝对型编码器读取的功能。

2. 输入/输出说明

输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	数据有效时机
Channel	组别设定	WORD	K1~4(*1)	Enable 上升沿时
Execute	上升沿时启动读取	BOOL	TRUE / FALSE	-
Bit	编码器分辨率设定	DWORD	K1~32(*2)	执行中持续更新
输出状态引脚				
引脚名称	功能说明	数据格式	产生上升沿时机	产生下降沿时机
Done	设定完成	BOOL	<ul style="list-style-type: none"> 执行完成时 	<ul style="list-style-type: none"> Execute 下降沿时 完成时若 Execute 为 FALSE，下个周期 Done 会变成 FALSE
Busy	执行中	BOOL	<ul style="list-style-type: none"> Execute 上升沿时 	<ul style="list-style-type: none"> Done 上升沿时 Error 上升沿时 Aborted 上升沿时
Aborted	功能块被其它命令中断	BOOL	<ul style="list-style-type: none"> 执行中被命令停止 	<ul style="list-style-type: none"> Execute 下降沿时 被中断时，若 Execute 为 FALSE，下个周期 Aborted 会变成 FALSE
Error	功能块产生错误	BOOL	<ul style="list-style-type: none"> 输入参数错误 启动时设定轴已在运动状态 	<ul style="list-style-type: none"> Execute 下降沿时

输出数值引脚				
引脚名称	功能说明	数据格式	输出范围	更新型式
Position	目前编码器位置	DWORD	K 0 ~ K 2,147,483,647	Done 上升沿时

(*1) 硬件配线方式

组别	T+	T-	D+	D-
1	Y0.0+	Y0.0-	X0.0+	X0.0-
2	Y0.2+	Y0.2-	X0.1+	X0.1-
3	Y0.4+	Y0.4-	X0.2+	X0.2-
4	Y0.6+	Y0.6-	X0.3+	X0.3-

(*2) 编码器分辨率设定方式：

若 SSI 编码器规格如下：

内容	规格
每圈分辨率	8192 (13bit)
连续圈数	4096 (12bit)

则编码器分辨率设定值为：每圈分辨率+连续圈数+1 = 13+12+1 = 26

3. 错误产生及排除

错误状况	排除方式
输入数值错误	检查输入引脚输入数值，是否超过可允许范围

4. 支持模块

此运动功能块仅支持 AH10PM-5A 此种 AH500 机种之运动控制模块。

MEMO

5

6

第6章 模块数据传递

目录

6.1	功能简介.....	6-2
6.2	参数介绍.....	6-2
6.3	功能使用.....	6-5

6.1 功能简介

AH500CPU 模块可透过设定数据交换方式，达成 AH500 运动控制模块与 AH500CPU 模块数据交换。可支援：

1. CPU 模块写入 AH500 运动控制模块 400 个 word
2. AH500 运动控制模块写入 CPU 模块 400 个 word
3. CPU 模块写入 AH500 运动控制模块 400 个 bit
4. AH500 运动控制模块写入 CPU 模块 400 个 bit

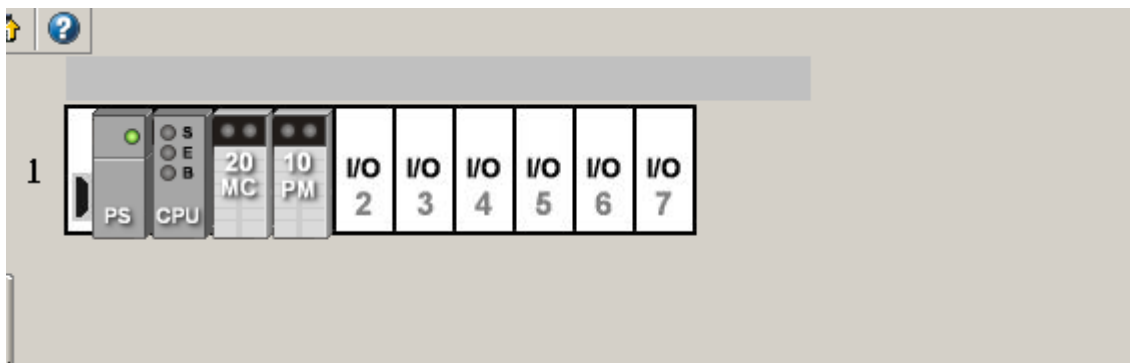
而使用者可利用此 400 个 word 与 400 个 bit 来做模块的控制或监控之用。例如可使用 CPU 模块写入的数值，透过 AH500 运动控制模块内部已写入的 PMSoft 程序，将此数值填入已定义功能的寄存器内，藉以启动或修改数值，并且也可透过程序将已定义功能的寄存器写入数据交换的区域，用以让 CPU 模块获得目前模块的运行状况。

以下章节以 AH20MC-5A 模块做为介绍范例。

6.2 参数介绍

ISPSOft 中的硬件规划工具 (HWCONFIG) 画面如下：

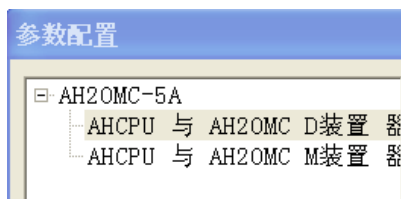
6



在上图为 HWCONFIG 中显示硬件配置画面，在此画面下，对 AH20MC-5A 利用鼠标左键连点两下，会出现以下页面：



在此页面中可分为两部分操作。在页面的左侧，显示此模块的参数功能列表，以 AH20MC-5A 来说一共分为主机与 AH20MC-5A 的 D 装置器通讯设定和主机与 AH20MC-5A 的 M 装置器通讯设定两种：



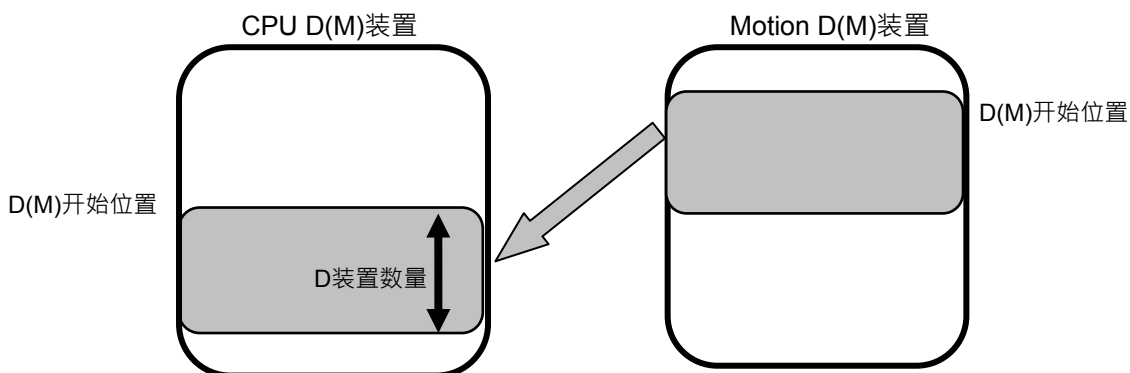
而在页面的右侧为 - 此功能下的细部功能设定参数：

AHCPU 与 AH20MC D装置器通讯设定						
	描述	地址	监控	设置	注	
▶	AHCPU << AH20MC -AHCPU D装置 起始编号		<input type="checkbox"/>	0	...	AHCPU << AH
	AHCPU << AH20MC -AH20MC D装置 起始编号		<input type="checkbox"/>	0		AHCPU << AH
	AHCPU << AH20MC -D装置 长度		<input type="checkbox"/>	0		AHCPU << AH
	AHCPU >> AH20MC -AHCPU D装置 起始编号		<input type="checkbox"/>	0	...	AHCPU >> AH
	AHCPU >> AH20MC -AH20MC D装置 起始编号		<input type="checkbox"/>	0		AHCPU >> AH
	AHCPU >> AH20MC -D装置 长度		<input type="checkbox"/>	0		AHCPU >> AH

细部参数功能定义如下：

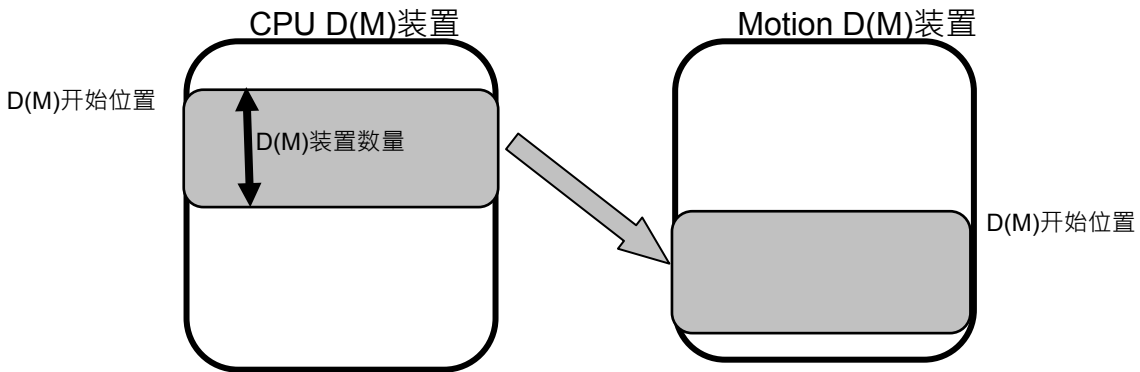
▶	AHCPU << AH20MC -AHCPU D装置 起始编号		<input type="checkbox"/>	0	...
	AHCPU << AH20MC -AH20MC D装置 起始编号		<input type="checkbox"/>	0	
	AHCPU << AH20MC -D装置 长度		<input type="checkbox"/>	0	

- ◆ AHCPU <<AH20MC- AHCPU D(M)装置起始编号，为主机读取 AH20MC-5A 模块的 D(M)装置时，主机存放数据的起始装置编号。
- ◆ AHCPU <<AH20MC-AH20MC D(M)装置起始编号，为主机读取 AH20MC-5A 模块的 D(M)装置时，AH20MC-5A 模块被读取的起始装置编号。
- ◆ AHCPU <<AH20MC- D(M)装置长度，为主机读取 AH20MC-5A 模块的 D(M)装置时，装置编号的总数量。



▶	AHCPU << AH20MC -AHCPU D装置 起始编号		<input type="checkbox"/>	0	...
	AHCPU << AH20MC -AH20MC D装置 起始编号		<input type="checkbox"/>	0	
	AHCPU << AH20MC -D装置 长度		<input type="checkbox"/>	0	

- ◆ AHCPU >>AH20MC- AHCPU D(M)装置起始编号，为主机写入 AH20MC-5A 模块的 D(M)装置时，主机写入的起始装置编号。
- ◆ AHCPU >> AH20MC --AH20MC D(M)装置起始编号，为主机写入 AH20MC-5A 模块的 D(M)装置时，AH20MC-5A 模块被写入的起始装置编号。
- ◆ AHCPU >> AH20MC --AH20MC D(M)装置长度，为主机写入 AH20MC-5A 模块的 D(M)装置时，装置编号的总数量。

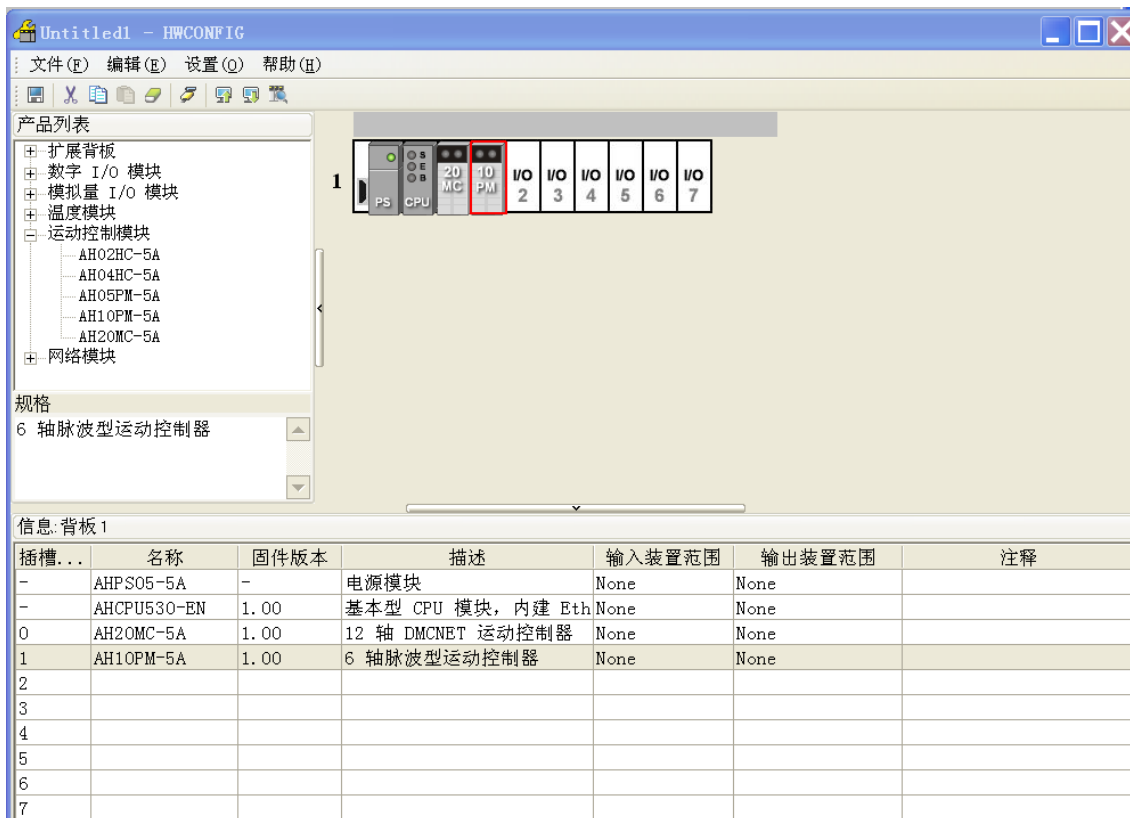


6

6.3 功能使用

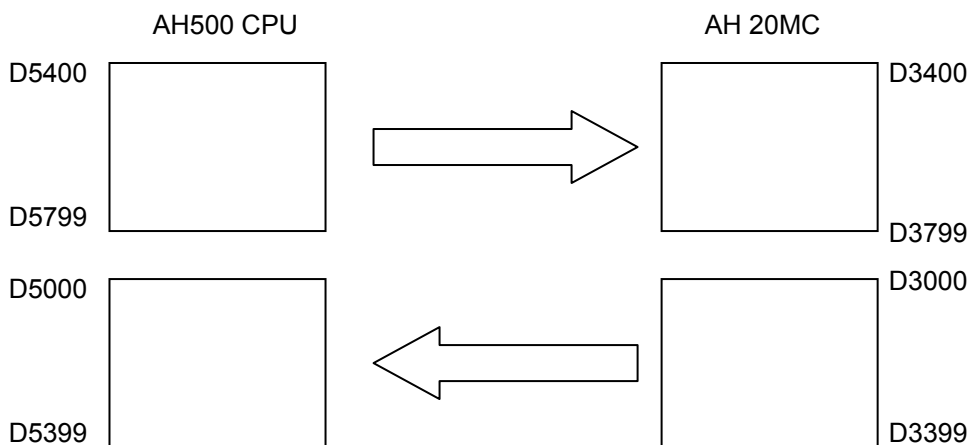
ISPSOft 设定使用步骤

1. 先利用 HWCONFIG 做模块扫描，扫描完成后可看到如以下画面：



2. 规划 AH500 系统主机与 AH20MC-5A 的数据交换的长度以及数据摆放的位置。

假设分别为




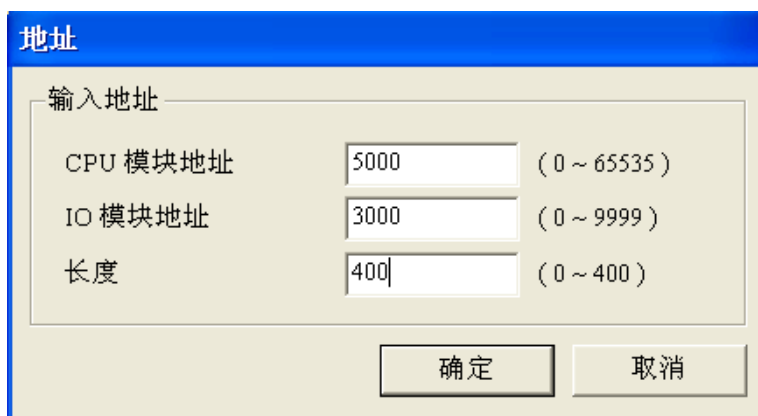
代表 AH500 系统主机每次写入 AH20MC-5A 模块 400 个 D 装置，并且同时读出 400 个 D 装置的位置，而写入的 D 装置为从 AH500 系统主机的 D5400 到 D5799 写入 AH20MC-5A 模块中 D3400 到 D3799，并将 AH20MC-5A 模块中 D3000 到 D3399 读出且摆放到 AH500 系统主机的 D5000 到 D5399。

3. HWCONFIG 设定

在 HWCONFIG 中利用模块参数设定方式，调用出设定参数画面。




单击 AHCPU<<AH20MC-AHCPUD 装置起始编号，初始值的 ，跳出以下窗口，在此窗口中设定 CPU 模块位置 5000，IO 模块位置 3000，长度 400。

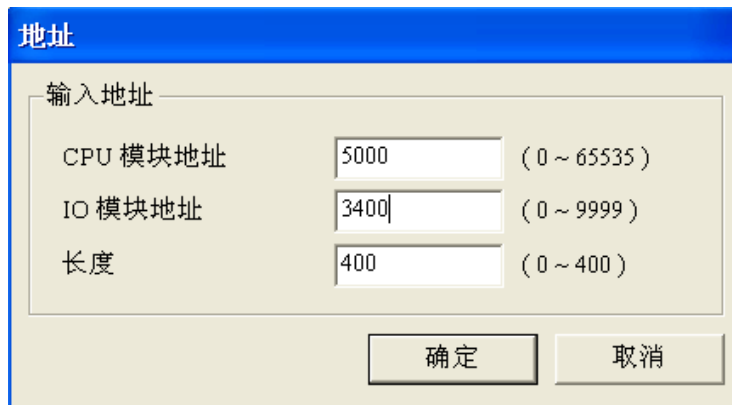


按下确定后

AHCPU << AH20MC -AHCPU D装置 起始编号	<input type="checkbox"/>	5000	...
AHCPU << AH20MC -AH20MC D装置 起始编号	<input type="checkbox"/>	3000	
AHCPU << AH20MC -D装置 长度	<input type="checkbox"/>	400	

6

单击在 AHCPU>>AH20MC-AHCPUD 装置起始编号，初始值的 ，跳出以下窗口，在此窗口中设定 CPU 模块位置 5400，IO 模块位置 3400，长度 400。



地址

输入地址

CPU 模块地址 (0 ~ 65535)

IO 模块地址 (0 ~ 9999)

长度 (0 ~ 400)

确定 取消

按下确定后

AHCPU << AH20MC -AHCPU D装置 起始编号	<input type="checkbox"/>	5000	...
AHCPU << AH20MC -AH20MC D装置 起始编号	<input type="checkbox"/>	3400	
AHCPU << AH20MC -D装置 长度	<input type="checkbox"/>	400	

4. PMSoft 程序

AH20MC-5A 模块可利用与主机的数据交换来获得主机的设定值，而后再利用使用者所写的 PMSoft 程序，将主机传送来的数据填入模块中的 SR 寄存器。例如单段速运动：



将 D3401 和 D3402 写入 SR1023 和 SR1024 中，做为第 1 轴单段速的目标位置，同时将 D3403 写入 SR1030 做为第 1 轴的启动命令。另外设定 SM1048 的 ON/OFF 来切换 D3000 的数值，AH500 系统主机即可以此做为单段速是否完成的判断依据。

MEMO

6



第7章 单轴运动功能

目录

7.1	单轴运动功能	7-2
7.2	单轴运动功能简介	7-13
7.3	JOG 模式使用介绍	7-14
7.3.1	专属对应特殊寄存器	7-14
7.3.2	运动操作说明	7-15
7.4	变速度模式使用介绍	7-16
7.4.1	专属对应特殊寄存器	7-16
7.4.2	运动操作说明	7-17
7.5	手摇轮操作模式使用介绍	7-18
7.5.1	专属对应的特殊寄存器	7-18
7.5.2	运动操作说明	7-19
7.6	原点回归模式使用介绍	7-20
7.6.1	专属对应的特殊寄存器	7-20
7.6.2	运动操作说明	7-22
7.7	单段速定位模式使用介绍	7-26
7.7.1	专属对应的特殊寄存器	7-26
7.7.2	运动操作说明	7-27
7.8	插入单段速定位模式使用介绍	7-28
7.8.1	专属对应的特殊寄存器	7-28
7.8.2	运动操作说明	7-29
7.9	连续两段速定位模式使用介绍	7-30
7.9.1	专属对应的特殊寄存器	7-30
7.9.2	运动操作说明	7-32
7.10	插入两段速定位使用介绍	7-33
7.10.1	专属对应的特殊寄存器	7-33
7.10.2	运动操作说明	7-34
7.11	状态标志与寄存器	7-35

7.1 单轴运动功能

运动模式特殊寄存器 (SR) · 它的种类及功能如下所示。详细说明可参考本节的功能说明。

SR 编号		特殊寄存器名称	设定范围	出厂值
第 (1+N) 轴 ^{*4}				
HW ^{*1}	LW ^{*1}			
-	SR1000+100*N	参数设定	b0~b15	16#0
-	SR1001+100*N	背隙补偿	依需求设定对应值	16#0
SR1003+100*N	SR1002+100*N	电机转一圈所需脉冲数 (A)	1~+2,147,483,647 PLS/REV	K2,000
SR1005+100*N	SR1004+100*N	电机转一圈之移动距离 (B)	1~+2,147,483,647 ^{*2}	K1,000
SR1007+100*N	SR1006+100*N	最高速度 V_{MAX}	0~+2,147,483,647 ^{*3}	K10500,000
SR1009+100*N	SR1008+100*N	启动速度 V_{BIAS}	0~+2,147,483,647 ^{*3}	K0
SR1011+100*N	SR1010+100*N	寸动 JOG 速度 V_{JOG}	0~+2,147,483,647 ^{*3}	K5,000
SR1013+100*N	SR1012+100*N	原点回归速度 V_{RT}	0~+2,147,483,647 ^{*3}	K50,000
SR1015+100*N	SR1014+100*N	原点回归减速速度 V_{CR}	0~+2,147,483,647 ^{*3}	K1,000
-	SR1016+100*N	零点 (PG0) 信号数 N	0~+32,767 PLS	K0
-	SR1017+100*N	脉冲信号数 P	-32,768~+32,767 PLS	K0
SR1019+100*N	SR1018+100*N	原点位置定义 HP	0~±999,999 ^{*1}	K0
-	SR1020+100*N	加速时间 T_{ACC}	10~+32,767 ms	K5100
-	SR1021+100*N	减速时间 T_{DEC}	10~+32,767 ms	K5100
SR1023+100*N	SR1022+100*N	目标位置 (I) P (I)	-2,147,483,648~+2,147,483,647 ^{*1}	K0
SR1025+100*N	SR1024+100*N	运转速度 (I) V (I)	0~+2,147,483,647 ^{*1}	K1000
SR1027+100*N	SR1026+100*N	目标位置 (II) P (II)	-2,147,483,648~+2,147,483,647 ^{*1}	K0
SR1029+100*N	SR1028+100*N	运转速度 (II) V (II)	0~+2,147,483,647 ^{*2}	K2,000
-	SR1030+100*N	运转命令	b0~b15	16#0

7

SR 编号		特殊寄存器名称	设定范围	出厂值
第 (1+N) 轴 ^{*4}				
HW ^{*1}	LW ^{*1}			
-	SR1031+100*N	工作模式	b0~b15	16#0
SR1033+100*N	SR1032+100*N	当前命令位置 CP (PLS)	-2,147,483,648~+2,147,483,647 ^{*1}	K0
SR1035+100*N	SR1034+100*N	当前命令速度 CS (PPS)	0~+2,147,483,647 PPS	K0
SR1037+100*N	SR1036+100*N	当前命令位置 CP (unit*3)	-2,147,483,648~+2,147,483,647 ^{*1}	K0
SR1039+100*N	SR1038+100*N	当前命令速度 CS (unit*3)	0~+2,147,483,647 PPS	K0
-	SR1040+100*N	执行状态	b0~b15	16#0
-	SR1041+100*N	错误编号	请参考错误码表	16#0
-	SR1042+100*N	电子齿轮 (分子)	1~+32,767	K1
-	SR1043+100*N	电子齿轮 (分母)	1~+32,767	K1
SR1045+100*N	SR1044+100*N	手摇轮输入频率	由手摇轮输入的脉冲频率	K0
SR1047+100*N	SR1046+100*N	累计手摇轮输入脉冲个数	计数由手摇轮输入的脉冲个数	K0
-	SR1048+100*N	手摇轮输入响应速度	手摇轮输入响应速度	K5
-	SR1049+100*N	停止模式 (OX0-99)	依需求设定对应值	K0
SR1051+100*N	SR1050+100*N	电子原点	依需求设定对应值	K0
-	SR1052+100*N	程序号码指定方式	依需求设定对应值	K0
-	SR1053+100*N	OX 错误 STEP 位置	依需求设定对应值	K0
SR1069+100*N	SR1068+100*N	DMCNET 伺服编码器当前位置 ^{*5}	依台达 A2 伺服设定显示对应值	K0

*1. HW : High Word ; LW : Low Word 。

*2. 单位可选择 um/rev 、 m deg/rev 及 10⁻⁴ inch/rev 。

7

- *3. 设定单位依照“SR1000+100*N”参数设定”之 b0 · b1 单位系设定而变化。
- *4. N 轴设定值为 0~15。
- *5. 仅支持 AH20MC-5A。

不同的单轴运动对应的寄存器一览表如下，操作模式对应的寄存器也会有不同。

运动模式对应使用寄存器		参数名称代码	操作模式								
第 (1+N) 轴			寸动 JOG	原点回归	单段速度定位	插入单段速度定位	单段速外部触发	连续两段速度定位	插入二段速度定位	变速度	手摇轮 (MPG) 输入
HW	LW										
-	SR1000+100*N	参数设定	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
SR1003+100*N	SR1002+100*N	电机转一圈所需脉冲数 A	若单位系选择为电机单位，则不须设定 若为机械单位或复合单位系，则必须作设定								
SR1005+100*N	SR1004+100*N	电机转一圈移动距离 B									
SR1007+100*N	SR1006+100*N	最高速度 V _{MAX}	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
SR1009+100*N	SR1008+100*N	启动速度 V _{BIAS}	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
SR1011+100*N	SR1010+100*N	寸动 JOG 速度 V _{JOG}	⊙	-	-	-	-	-	-	-	-
SR1013+100*N	SR1012+100*N	原点回归速度 V _{RT}									
SR1015+100*N	SR1014+100*N	原点回归减速速度 V _{CR}									
-	SR1016+100*N	原点回归之零点 (PGO) 信号数 N	-	⊙	-	-	-	-	-	-	-
-	SR1017+100*N	原点回归之脉冲信号数 P									
SR1019+100*N	SR1018+100*N	原点位置定义 HP	-	⊙	-	-	-	-	-	-	-
-	SR1020+100*N	加速时间 T _{ACC}	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	-
-	SR1021+100*N	减速时间 T _{DEC}	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	-
SR1023+100*N	SR1022+100*N	目标位置 (I) P (I)	-	-	⊙	⊙	⊙	⊙	⊙	-	⊙
SR1025+100*N	SR1024+100*N	运转速度 (I) V (I)	-	-	⊙	⊙	⊙	⊙	⊙	⊙	-
SR1027+100*N	SR1026+100*N	目标位置 (II) P (II)	-	-	-	-	-	⊙	⊙	-	⊙
SR1029+100*N	SR1028+100*N	运转速度 (II) V (II)	-	-	-	-	-	⊙	⊙	-	-
-	SR1030+100*N	运转命令	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
-	SR1031+100*N	工作模式	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙



运动模式对应使用寄存器		参数名称代码	操作模式								
第 (1+N) 轴			手摇 JOG	原点 回归	单段 速度定 位	插入 单段速 定位	单段 速外部 触发	连续 两段速 定位	插入 二段速 定位	变速度	手摇 轮 (MPG) 输入
HW	LW										
SR1033+100*N	SR1032+100*N	当前命令位置 CP (PLS)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
SR1035+100*N	SR1034+100*N	当前速命令度 CS (PPS)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
SR1037+100*N	SR1036+100*N	当前命令位置 CP (unit)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
SR1039+100*N	SR1038+100*N	当前命令速度 CS (unit)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
-	SR1042+100*N	电子齿轮分子	-	-	-	-	-	-	-	-	⊙
-	SR1043+100*N	电子齿轮分母	-	-	-	-	-	-	-	-	⊙
SR1045+100*N	SR1044+100*N	手摇轮输入频率	-	-	-	-	-	-	-	-	⊙
SR1047+100*N	SR1046+100*N	累计手摇轮输入脉冲 数	-	-	-	-	-	-	-	-	-
-	SR1048+100*N	MPG 响应速度	-	-	-	-	-	-	-	-	-
-	SR1049+100*N	停止模式 (OX0-99)	-	-	-	-	-	-	-	-	-
SR1051+100*N	SR1050+100*N	电子原点	-	-	-	-	-	-	-	-	-
-	SR1052+100*N	程序号码指定方式	-	-	-	-	-	-	-	-	-
-	SR1053+100*N	OX 错误 STEP 位置	-	-	-	-	-	-	-	-	-
SR1069+100*N	SR1068+100*N	DMCNET 伺服编码器 当前位置*1	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙

*1. 仅支持 AH20MC-5A。

常用特殊寄存器做功能详细说明：

1. 参数设定

第 1+N 轴	
HW	LW
-	SR1000+100*N

7

[说明]

特殊寄存器名称															
SR1000+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	加减速曲线选择	DOG 触发启动模式	绝对/相对坐标设定	旋转方向	原点回归 DOG 下降沿检测	原点回归模式	原点回归方向	'	'	脉冲输出型式 (正逻辑)			'	设定单位系	

bit 0~bit15 各位所代表的意义如下所示：

- SR1000+100*N 之 b0 · b1：设定单位系

b1	b0	单位系	说明
0	0	电机单位	以脉冲为单位
0	1	机械单位	以长度·角度为单位
1	0	复合单位	位置以长度·角度(机械单位)为单位·速度则以脉冲(电机单位)为单位
1	1		

	电机单位	复合单位	机械单位
位 置	pulse	um	
	pulse	m deg	
	pulse	10-4inch	
速 度	pulse/sec		cm/min
	pulse/sec		10deg/min
	pulse/sec		inch/min

- 位置数据：原点位置【HP】、目标位置(I)【P(I)】、目标位置(II)【P(II)】、当前命令位置【CP】。
- 速度数据：最高速度【V_{MAX}】、启动速度【V_{BIAS}】、寸动 JOG 速度【V_{JOG}】、原点回归速度【V_{RT}】、原点减速速度【V_{CR}】、运转速度(I)【V(I)】、运转速度(II)【V(II)】。
- 范例一：

电机单位 b[1:0]=00⇒单位系：位置数据：Pulse·速度数据：Pulse/sec (PPS)。设定：目标位置 P(I)：10,000 (Pulse)·运转速度 V(I)：10K (PPS)。

位置控制器只要送出 10,000 个 Pulse (频率为 10kPPS)·可移动至目标位置·每个 Pulse 所移动之距离·由使用者依据设备参数计算每一个脉冲可移动多少距离。

7

■ 范例二：

机械单位 $b[1:0]=01 \Rightarrow$ 单位系：位置数据： μm · 速度数据： cm/min 。若设定轴为 $N=0$ · 则 $(SR1003 \cdot SR1002) = 1,000 (\text{Pulse}/\text{REV}) \cdot (SR1005 \cdot SR1004) = 100 (\mu\text{m}/\text{REV}) \cdot$ 且目标位置 $P(I) = 10,000 (\mu\text{m}) \cdot$ 运转速度 $V(I) = 6 (\text{cm}/\text{min}) \cdot$ 位置控制器的脉冲指令的脉冲个数及其频率为：

$$\text{距离} = \frac{\text{距离}}{\text{圈}} \times \frac{\text{圈}}{\text{脉冲数}} \times \text{脉冲数}$$

$$\text{由位置控制器计算要跑到 } P(I) \text{ 所须之脉冲数} = \frac{P(I)\mu\text{m}}{B/A} = P(I) \times \frac{A}{B} = 100,000 \text{ Pulse}$$

$$\text{运转速度 } V(I) : 6 (\text{cm}/\text{min}) = 60,000/60 (\mu\text{m}/\text{sec})$$

$$\text{速度} = \frac{\text{距离}}{\text{时间}} = \frac{\text{距离}}{\text{圈}} \times \frac{\text{圈}}{\text{脉冲数}} \times \frac{\text{脉冲数}}{\text{时间}}$$







$$\text{由位置控制器计算脉冲频率 (PPS)} = V(I) \times \frac{10^4}{60} \times \frac{A}{B} = \frac{60,000}{60} \times \frac{1,000}{100} = 10,000 \text{ PPS}$$

■ 范例三：

复合单位 $b[1:0]=10 \cdot 11 \Rightarrow$ 单位系：位置数据： μm · 速度数据： $\text{pulse}/\text{sec} (\text{PPS})$ · 若设定轴为 $N=0$ · 则 $(SR1003 \cdot SR1002) = 2,000 (\text{Pulse}/\text{REV}) \cdot (SR1005 \cdot SR1004) = 100 (\mu\text{m}/\text{REV}) \cdot$ 且目标位置 $P(I) : 10,000 (\mu\text{m}) \cdot$ 运转速度 $V(I) : 10\text{K} (\text{PPS}) \cdot$ 此时位置控制器的脉冲指令的脉冲个数为：由位置控制器计算要跑到 $P(I)$ 所须之脉冲数

$$= \frac{P(I)\mu\text{m}}{B/A} = P(I) \times \frac{A}{B} = 200,000 \text{ PULSE}$$

● SR1000+100*N 之 b4 · b5：脉冲输出方式

b5	b4	脉冲输出型式 (正逻辑)	说明
0	0	FP 正转脉冲  RP 反转脉冲 	双脉冲
0	1	FP 脉冲  RP 方向(DIR) 	单脉冲
1	0	FP A相脉冲 	A/B 相脉冲
1	1	RP B相脉冲  正转 反转	4*A/B 相脉冲

● SR1000+100*N 之 b8：原点回归方向

● SR1000+100*N 之 b9：原点回归模式



- SR1000+100*N 之 b10：原点回归 DOG 下降沿检测
b8、b9、b10 详见第 7.6 章原点回归。
- SR1000+100*N 之 b11：旋转方向
 - (1) b[11]=0 时，当正转运动时，当前命令位置 (CP) 内容值递增。
 - (2) b[11]=1 时，当正转运动时，当前命令位置 (CP) 内容值递减。
- SR1000+100*N 之 b12：绝对/相对坐标设定
 - (1) 当 b[12]=0 时，设定为绝对坐标定位。
 - (2) 当 b[12]=1 时，设定为相对坐标定位。
- SR1000+100*N 之 b13：DOG 触发启动模式
详见第 7.1 章 (插入单段速定位运动模式、插入二段速定位运动模式有效)。
- SR1000+100*N 之 b14：加减速曲线选择
 - (1) b[14]=0 时，为梯形加减速曲线。
 - (2) b[14]=1 时，为 S 加减速曲线。

2. 电机转一圈所需脉冲数 (A)

第 1+N 轴	
HW	LW
SR1003+100*N	SR1002+100*N

[说明]

- 由于伺服驱动器内具有电子齿轮比例设定功能，因此伺服电机转一圈所需之脉冲数并不一定等于编码器旋转一圈产生脉冲数，因此两者需搭配如下：
电机转一圈所需指令脉冲数 A_x 电子齿轮(CMX/CDV)=编码器旋转一圈产生脉冲数
EX：若电机转一圈所需指令脉冲数 $A=20,000$ ，搭配台达伺服 ASDA-A2 系列的分辨率为 1,280,000，则设定电子齿轮为 128/2。
- 设定单位依照 SR1000+100*N 参数设定之 b0、b1 单位系设定来变化，当系统单位系设定为机械单位或复合单位时，参数 A 设定有效；若系统单位设定为电机单位，则参数 A 设定不作用。

3. 电机转一圈之移动距离 (B)

第 1+N 轴	
HW	LW
SR1005+100*N	SR1004+100*N

[说明]

- 电机转一圈所带动之移动量，单位有三种可供选择： um/Rev 、 $mdeg/Rev$ 、 $10^{-4} inch/Rev$ ，由 SR1000+100*N 之 b0、b1 作设定： $B=1\sim+2,147,483,647$
- 设定单位依照 SR1000+100*N 之 b0、b1 参数设定单位系设定来变化。当系统单位系设定为机械单位或复合单位时，参数 B 设定有效；若系统单位设定为电机单位，则参数 B 设定不作用。



4. 最高速度 V_{MAX}

第 1+N 轴	
HW	LW
SR1007+100*N	SR1006+100*N

[说明]

- 各种操作模式的速度上限。设定范围 0~+2,147,483,647 (单位系由 SR1000+100*N 之 b0、b1 设定)。
- 对应运动指令 10~1000kPPS。若大于 1000k，则以 1000k 输出。若小于 10，则以 10 输出。

5. 启动速度 V_{BIAS}

第 1+N 轴	
HW	LW
SR1009+100*N	SR1008+100*N

[说明]

- 脉冲输出启动速度。设定范围 0~+2,147,483,647 (单位由 SR1000+100*N 之 b0、b1 设定)。
- 对应运动指令 0~1000kPPS。若大于 1000k，则以 1000k 输出。若小于 0，则以 0 输出。
- 若为步进驱动系统，请注意步进电机的共振区频率，将启动速度 V_{BIAS} 设定在共振区频率之上，作安全启动。

6. 加速时间 T_{ACC}

第 1+N 轴	
HW	LW
-	SR1020+100*N

[说明]

- 从 SR1008+100*N 启动速度 V_{BIAS} 加速到 SR1006+100*N 最高速度 V_{MAX} 所需的时间，设定值范围 0~32767，单位为 ms。
- 当设定值<10 时，则视为 10。设定值>32,767 时，则视为 32,767。
- S 加速曲线控制时，若要完整之 S 加速曲线控制，请将最高速度 V_{MAX} 设定成与运转速度相同。

7. 减速时间 T_{DEC}

第 1+N 轴	
HW	LW
-	SR1021+100*N

[说明]

- 从 SR1006+100*N 最高速度 V_{MAX} 减速到 SR1008+100*N 启动速度 V_{BIAS} 所需的时间，设定值范围 0~32767，单位为 ms。
- 当设定值<10 时，则视为 10。设定值>32,767 时，则视为 32,767。
- S 加速曲线控制时，若要完整之 S 加速曲线控制，请将最高速度 V_{MAX} 设定成与运转速度相同。

8. 当前命令位置 CP (PLS)

第 1+N 轴	
HW	LW
SR1033+100*N	SR1032+100*N

[说明]

- 显示范围：-2,147,483,648~+2,147,483,647。
- 当前命令位置，以脉冲值 (PLS) 表示，由 SR1000+100*N 之 b0 · b1 作设定。原点回归完成时，SR1018+100*N 原点位置定义 HP，会填入当前命令位置 CP (PLS) 数据中。

9. 当前命令速度 CS (PPS)

第 1+N 轴	
HW	LW
-	SR1035+100*N

[说明]

- 显示范围：0~+2,147,483,647。
- 当前速度，以 PPS 表示。

10. 当前命令位置 CP (unit)

第 1+N 轴	
HW	LW
SR1037+100*N	SR1036+100*N

[说明]

- 显示范围：-2,147,483,648~+2,147,483,647。
- 设定单位依照 SR1000+100*N 之 b0 · b1 单位系设定变化。当原点回归完成时，SR1018+100*N 原点位置定义 HP 填入 SR1036+100*N 当前命令位置数据中。

7

11. 当前命令速度 CS (unit)

第 1+N 轴	
HW	LW
SR1039+100*N	SR1038+100*N

[说明]

- 显示范围：0~+2,147,483,647。
- 设定单位依照 SR1000+100*N 之 b0 · b1 单位系设定来变化。

12. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2		b0
'	'	'	Jog 启动	两段速 插入定 位模式 启动	两段速 定位模 式启动	单段速 插入定 位模式 启动	单段速 定位运 动模式 启动	'	原点回 归模式 启动	手摇轮 输入操 作	变速度 运转模 式启动	JOG- 运转	JOG+ 运转		软件 Stop

[说明]

- 动作时机：当轴的 Ready 标志(SM1048+100*N)为 ON 时，bit 上升沿触发 (0→1) 启动运动命令。
- 软件 Stop：bit[0]上升沿触发 (0→1) 运动减速停止。
- 各运动模式详见第 7.2~7.10 节。

13. 工作模式

第 1+N 轴	
HW	LW
-	SR1031+100*N

[说明]

特殊寄存器名称															
SR1031+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
回 归 出 厂 设 定	'	'	'	'				'					CLR 信号 输出 模式	'	'

SR1031+100*N 之 bit0~bit15 各位所代表的意义如下所示：

- b2：CLR 信号输出模式。
 - (1) 当 b[2]=0 时，输出信号 CLR 为原点回归完成时，输出给伺服驱动器信号（时间长度约为 130ms），做为伺服内部偏差计数器的清除信号。

7

(2) 当 $b[2]=1$ 时，CLR 输出点做为一般输出点。

● b15 : 回归出厂设定

(1) $b[15]=1$: 所有参数回归出厂值设定。



7.2 单轴运动功能简介

1. AH500 运动定位模块共有 8 种运动模式：

1. 机械原点回归操作
2. 寸动运动模式操作
3. 单段速定位运动模式
4. 插入单段速定位运动模式
5. 连续两段速定位运动模式
6. 插入二段速定位运动模式
7. 变速度运动模式
8. 手摇轮 (MPG) 输入操作

2. 当多个工作模式同时被启动时，其处理优先级如下：

1. STOP 停止
2. 机械原点回归操作
3. 寸动 (JOG+) 运动模式操作
4. 寸动 (JOG-) 运动模式操作
5. 手摇轮 (MPG) 输入操作
6. 变速度运动模式
7. 单段速定位运动模式
8. 插入单段速定位运动模式
9. 连续两段速定位运动模式
10. 插入二段速定位运动模式

当同一轴其中一个工作模式执行中，而另一工作模式又被启动时，AH500 运动定位模块仍维持原来工作模式操作。

3. 单轴运动功能由各轴“运转命令”的 SR1030+100*N 来控制，将各轴的相关参数设定好后启动。

第 1+N 轴		运转命令
HW	LW	
-	SR1030+100*N	

[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2		b0
'	'	'	OX 启动	两段速 插入定 位模式 启动	两段速 定位模 式启动	单段速 插入定 位模式 启动	单段速 定位运 动模式 启动	'	原点回 归模式 启动	手摇轮 输入操 作	变速度 运转模 式启动	JOG+ 运转	JOG+ 运转		软件 Stop

● b0：软件 STOP

- (1) 动作时机：b[0]上升沿触发 (0→1)。
- (2) 定位控制器均会做减速停止。

7

7.3 JOG模式使用介绍

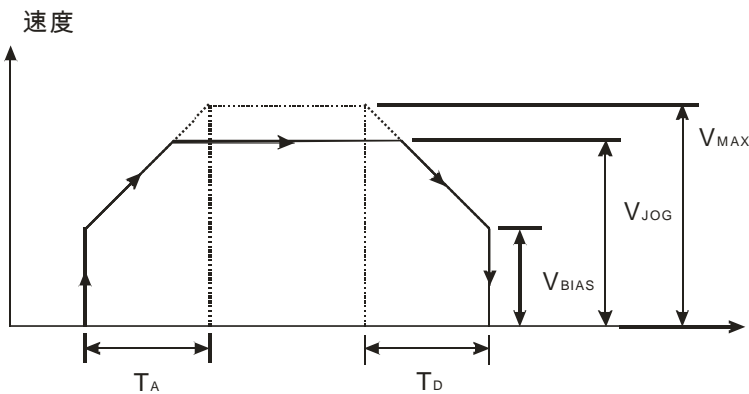
7.3.1 专属对应特殊寄存器

1. 寸动 JOG 速度 V_{JOG}

第 1+N 轴	
HW	LW
SR1011+100*N	SR1010+100*N

[说明]

- 寸动指令速度。设定范围 0~+2,147,483,647 (单位由 SR1000 (SR1100 · SR1200...) 之 b0 · b1 设定)。
 - 对应的脉冲指令 10~1000kPPS。若大于 1000k，则以 1000k 输出，若小于 10，则以 10 输出。
 - 设定范围限制 $V_{MAX} > V_{JOG} > V_{BIAS}$ 。
- 若 V_{JOG} 设定 $> V_{MAX}$ 设定，则 V_{JOG} 输出 = V_{MAX} 。
- 若 V_{JOG} 设定 $< V_{BIAS}$ 设定，则跳 ERROR
- JOG 速度可于运行间修改。若速度为 0 时运动停止，需重新启动；速度设为 0 时启动会显示错误。



2. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

7

[说明]

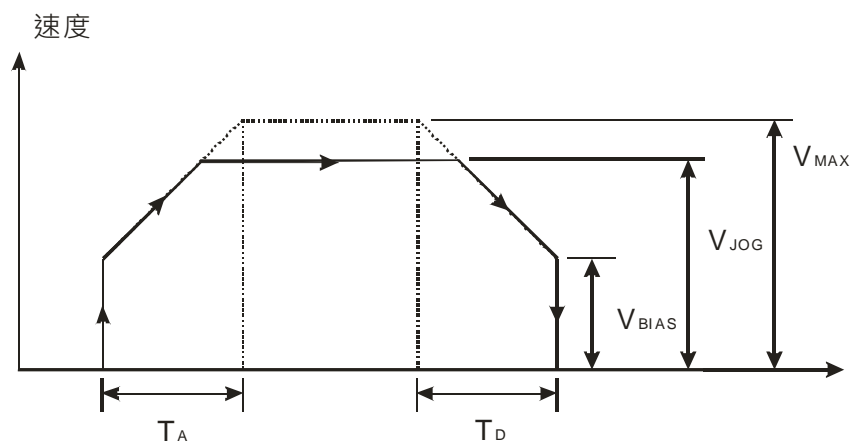
特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段 速插入 定位模 式启动	两段 速定位 模式启 动	单段 速插入 定位模 式启动	单段 速定位 运动模 式启动	'	原 点回 归模 式启 动	手 摇 轮 输 入 操 作	变 速 度 运 转 模 式 启 动	JOG- 运 转	JOG+ 运 转	'	软 件 S t o p

JOG+运转需设定 bit 2 为 1

JOG-运转需设定 bit 3 为 1。

7.3.2 运动操作说明

- SR1030+100*N 之 b2 : JOG+运转。当 b[2]=1 时，JOG+运转发送依所设定之 JOG 速度正方向脉冲 (CW)。
- SR1030+100*N 之 b3 : JOG-运转。当 b[3]=1 时，JOG-运转发送反方向脉冲 (CCW)。
- JOG 模式启动后，这时定位控制器开始执 JOG 运转，而 JOG 速度可于运行间修改。若速度为 0 时运动停止，需重新启动；启动时速度设为 0 时会错误。



7

7.4 变速度模式使用介绍

7.4.1 专属对应特殊寄存器

1. 运转速度 (I) V (I)

第 1+N 轴	
HW	LW
SR1025+100*N	SR1024+100*N

[说明]

- 设定范围 0~+2,147,483,647 (单位由 SR1000 (SR1100 · SR1200...) 之 b0 · b1 定义)。
- 对应的脉冲指令 10~1000kPPS。若大于 1000k，则以 1000k 输出，若小于 10，则以 10 输出。
设定范围限制： $V_{MAX} > V(I) > V_{BIAS}$ 。
- 当变速度运转时 (SR1030 (SR1130 · SR1230...) 之 b4=1)，运转速度 (I) V (I) 在运转中可改变。

2. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段速插入定位模式启动	两段速定位模式启动	单段速插入定位模式启动	单段速定位运动模式启动	'	原点回归模式启动	手摇轮输入操作	变速度运转模式启动	JOG- 运转	JOG+ 运转	'	软件 Stop

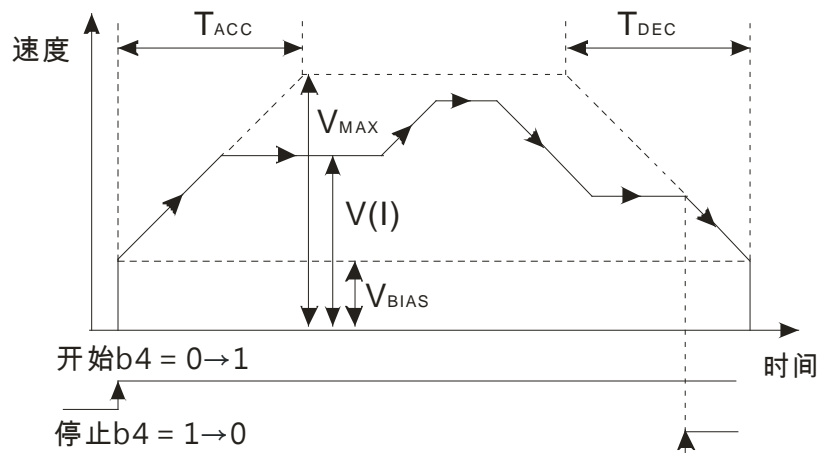
变速度运转模式启动需设定 bit 4 为 1。

7

7.4.2 运动操作说明

- SR1030+100*N 设定 bit 4=1 启动变速度运动，这时定位控制器开始执行变速度 $V(I)$ 运转，开始由 AH10PM-5A、AH15PM-5A 及 AH05PM-5A 脉冲产生单元送出脉冲，AH20MC-5A 则由 DMCNET 发送位置给伺服运行。
- 动作说明：运转时，由启动速度 V_{BIAS} 开始加速运转速度 $V(I)$ 后稳态运行，在运转期间，使用者可以任意的更改运转速度 $V(I)$ ，AH500 运动定位模块会依照运转速度 $V(I)$ 的设定值加速 / 减速。

动作示意图：



7.5 手摇轮操作模式使用介绍

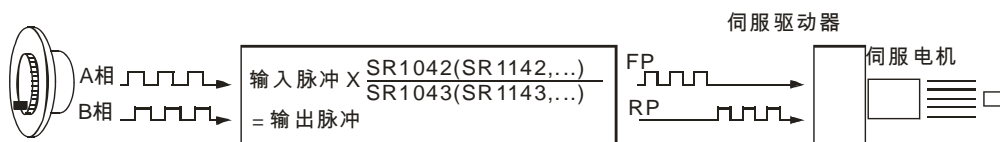
7.5.1 专属对应的特殊寄存器

1. 电子齿轮比

第 1+N 轴		电子齿轮比
HW	LW	
-	SR1042+100*N	电子齿轮 (分子)
-	SR1043+100*N	电子齿轮 (分母)

[说明]

- 当运转命令 SR1030+100*N 之 b5 设为 ON，表示为手摇轮脉冲输入工作模式启动。
- 以手摇轮产生 A/B 相脉冲输入至脉冲输入端，FP/RP 输出与输入脉冲关系如下图所示：



- 输出运行速度为手摇轮产生的脉冲输入频率与电子齿轮 (SR1042 (SR1142 · SR1242...)) · SR1043 (SR1143 · SR1243...)) 成比例的关系。

2. 手摇轮输入频率

第 1+N 轴	
HW	LW
SR1045+100*N	SR1044+100*N

[说明]

显示手摇轮输入的脉冲频率，频率值不受电子齿轮比 (SR1042+100*N · SR1043+100*N) 设定影响。

7

3. 累计手摇轮输入脉冲个数

第 1+N 轴	
HW	LW
SR1047+100*N	SR1046+100*N

[说明]

- 计数由手摇轮输入的脉冲个数，正转脉冲输入，该计数值为“加”动作，若为反转脉冲输入，则该计数值为“减”动作。
- 计数值不受电子齿轮比设定影响。

4. 手摇轮输入响应速度

第 1+N 轴	
HW	LW
-	SR1048+100*N

[说明]

- 响应速度设定愈快，表示命令脉冲输出与手摇轮脉冲输入时序越同步。
- 响应速度设定愈慢，表示命令脉冲输出反应落后于手摇轮脉冲输入时序。

设定值	响应速度
≥5	4ms (初始值)
4	32 ms
3	108 ms
2	256 ms
1 或 0	500 ms

5. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

[说明]

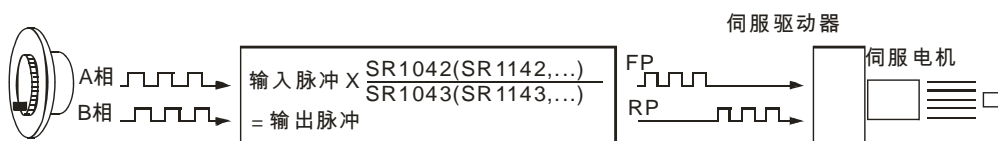
特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段速 插入定 位模式 启动	两段速 定位模 式启动	单段速 插入定 位模式 启动	单段速 定位运 动模式 启动	'	原点回 归模式 启动	手摇轮 输入操 作	变速度 运转模 式启动	JOG 运转	JOG+ 运转	'	软件 Stop

手摇轮输入操作模式启动需设定 bit 5 为 1。

7

7.5.2 运动操作说明

- SR1030+100*N 之 b 5=1 时，启动手摇轮脉冲输入功能。
- 可由 SR1046+100*N 获得手摇轮累计个数



7.6 原点回归模式使用介绍

7.6.1 专属对应的特殊寄存器

1. 原点回归速度 V_{RT}

第 1+N 轴	
HW	LW
SR1013+100*N	SR1012+100*N

[说明]

- 回到机械原点的速度。设定范围 1~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 设定)。
- 对应的脉冲指令 10~1000kPPS。若大于 1000k，则以 1000k 输出，若小于 10，则以 10 输出。
- 设定范围限制 $V_{MAX} > V_{RT} > V_{BIAS}$ 。
- 原点回归速度 V_{RT} 在执行中不允许改变。

2. 原点回归减速速度 V_{CR}

第 1+N 轴	
HW	LW
SR1015+100*N	SR1014+100*N

[说明]

- 设定范围 1~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 设定)。
- 对应的脉冲指令 10~1000kPPS。若大于 1000k，则以 1000k 输出，若小于 10，则以 10 输出。
- 执行原点回归时，电机以原点回归速度 V_{RT} 运转，当碰到 DOG 信号时，电机减速到原点回归减速速度 V_{CR} 运转。
- 为了精准的定位在零点，建议将原点回归减速速度 V_{CR} 设定为低速运行。
- 原点回归减速速度 V_{CR} 在执行中不允许改变。

3. 原点回归之零点 (PG0) 信号数 N

第 1+N 轴	
HW	LW
-	SR1016+100*N

[说明]

- 设定范围 0~65,535 (PULSE)。设定为正方向的零点信号数 N。
- 电机减速停止之参考信号，请参考 SR1000+100*N 之 b9 · b10 原点回归模式之说明。

4. 原点回归之脉冲信号数 P

第 1+N 轴	
HW	LW
-	SR1017+100*N

7

[说明]

- 设定范围-32,768~32,767 (PULSE)。正值设定为同原点回归方向的脉冲信号数 P，负值设定为反原点回归方向的脉冲信号数 P。
- 电机减速停止之参考信号，请参考 SR1000+100*N 之 b9，b10 原点回归模式之说明。

5. 原点位置定义 HP

第 1+N 轴	
HW	LW
SR1019+100*N	SR1018+100*N

[说明]

- 设定范围 0~±999,999 (单位由 SR1000+100*N 之 b0，b1 定义)。
- 原点回归完成时，当前命令位置 CP 会被更新为原点位置 HP。

6. 工作模式

第 1+N 轴	
HW	LW
-	SR1031+100*N

[说明]

- SR1031+100*N 之 b2：CLR 信号输出模式。
 - 当 b[2]=0 时，输出信号 CLR 为原点回归完成时，输出给伺服驱动器信号 (时间长度约为 130ms)，做为伺服内部偏差计数器的清除信号。
 - 当 b[2]=1 时，CLR 输出点做为一般输出点，状态由 b[3]控制 ON/OFF。

7. 参数设定

第 1+N 轴	
HW	LW
-	SR1000+100*N

[说明]

在原点回归模式下：

- SR1000+100*N 之 b8：原点回归方向
 - 当 b[8]=0 时，执行原点回归方向，往当前命令位置 (CP) 内容值反 (递减) 方向执行。
 - 当 b[8]=1 时，执行原点回归方向，往当前命令位置 (CP) 内容值正 (递增) 方向执行。
- SR1000+100*N 之 b9：原点回归模式
 - 当 b[9]=0 时，正常模式。即为检测近点 (DOG) 信号产生后，经过 N 个原点回归零点讯号后，再经过 P 个原点回归脉冲讯号后，电机立刻停止。
 - 当 b[9]=1 时，复写模式。即为检测近点 (DOG) 信号产生后，当零点信号数 N 脉冲及脉冲信号数 P 脉冲其中一项先到达时，电机立刻停止。

7

- SR1000+100*N 之 b10 : 原点回归 DOG 下降沿检测
 - 当 b[10]=0 时 , 原点回归 DOG 下降沿检测 ON , 即为 DOG 下降沿检测。
 - 当 b[10]=1 时 , 原点回归 DOG 下降沿检测 OFF , 即为 DOG 上升沿检测。

8. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

[说明]

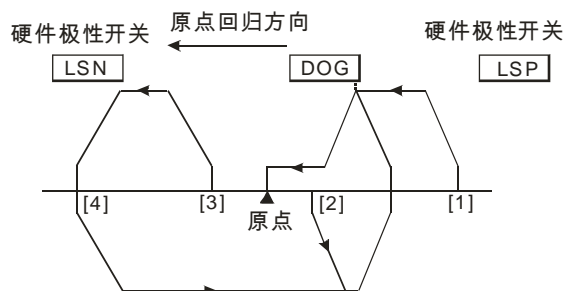
特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段 速度插入 定位模式 启动	两段 速度定位 模式启动	单段 速度插入 定位模式 启动	单段 速度定位 运动模式 启动	'	原点 回归模式 启动	手摇 轮输入 操作	变速度 运转模式 启动	JOG- 运转	JOG+ 运转	'	软件 Stop

原点回归模式启动需设定 bit 6 为 1。

7.6.2 运动操作说明

7

- SR1030+100*N 之 bit 6 由 0→1 变化时 , 开始执行原点回归动作 , 由于目前位置 (CP) 所处的位置不同 , 执行原点回归动作也有所不同 , 可分为两种情况 :



- 位置 (1) : 起始位置[1] , 在原点及 DOG 信号的右边 , DOG 信号=OFF。
- 位置 (2) : 起始位置[2] , 在原点的右边 , DOG 信号=ON。
- 位置 (3) : 目前不支持。
- 位置 (4) : 目前不支持。

● 参数设定

减速停止可分为四种模式：

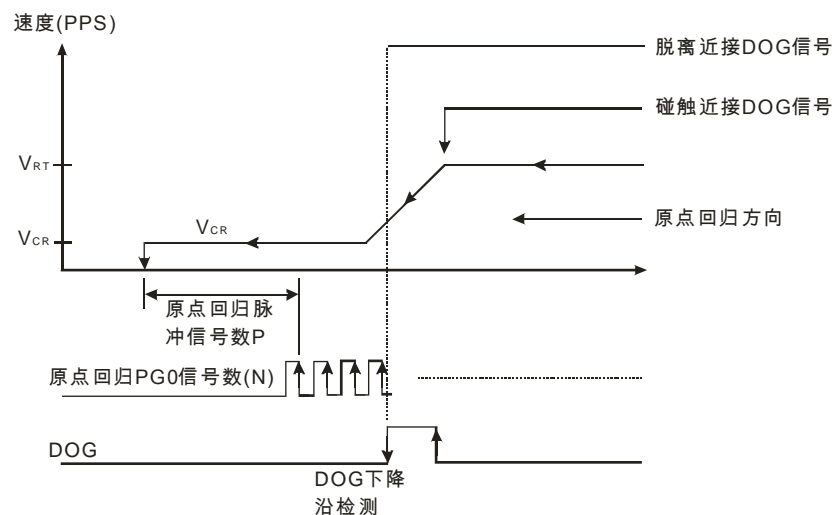
- SR1000+100*N 之 b[9 : 10]=00⇒正常模式，原点回归 DOG 下降沿检测 ON

原点回归动作步骤：

1. 电机以原点回归速度 V_{RT} 运转，
2. 当碰到近点 (DOG) 信号时，电机减速到以原点回归减速速度 V_{CR} 运转，
3. 检测近点 (DOG) 信号下降沿产生后，经过 N 个原点回归之零点信号脉冲后，再经过 P 个原点回归之脉冲信号脉冲后电机停止。

若设定原点回归之 PG0 零点信号数 N 或原点回归之脉冲信号数 P 太小时，碰到近点 (DOG) 信号时，电机开始向原点回归减速速度 V_{CR} 做减速动作，再检测近点 (DOG) 信号下降沿产生后，若达到指定之 N 个 PG0 零点信号，再经过 P 个原点回归之脉冲信号立即停止 (不论是否已到达原点回归减速速度 V_{CR})。

假设原点回归之信号数 N 设定为 0，原点回归之脉冲信号数 P 设定为 0，则碰到近点 (DOG) 信号后再检测近点 (DOG) 信号下降沿产生后立即停止。



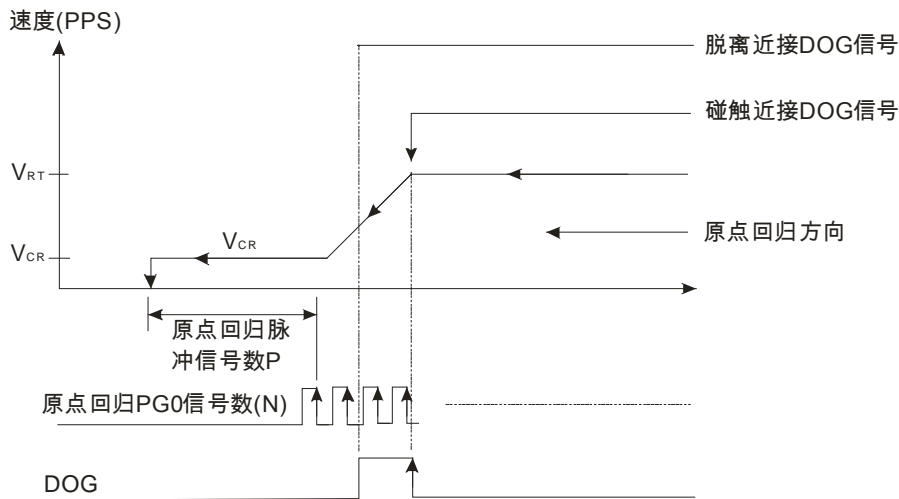
- SR1000+100*N 之 b[9 : 10]=01⇒正常模式，原点回归 DOG 下降沿检测 OFF

原点回归动作步骤：

1. 电机以原点回归速度 V_{RT} 运转
2. 当碰到近点 (DOG) 信号时，电机开始向原点回归减速速度 V_{CR} 做减速动作，经过 N 个原点回归之零点信号脉冲后，再经过 P 个原点回归之脉冲信号脉冲后电机停止。

若设定原点回归之 PG0 零点信号数 N 或原点回归之脉冲信号数 P 太小时，碰到近点 (DOG) 信号时，电机开始向原点回归减速速度 V_{CR} 做减速动作，若达到 N 个指定之 PG0 零点信号数，再经过 P 个原点回归之脉冲信号立即停止 (不论是否已到达原点回归减速速度 V_{CR})。

假设原点回归之信号数 N 设定为 0，原点回归之脉冲信号数 P 设定为 0，则碰到近点 (DOG) 信号立即停止。



■ SR1000+100*N 之 b[9 : 10]=10⇒复写模式，原点回归 DOG 下降沿检测 ON

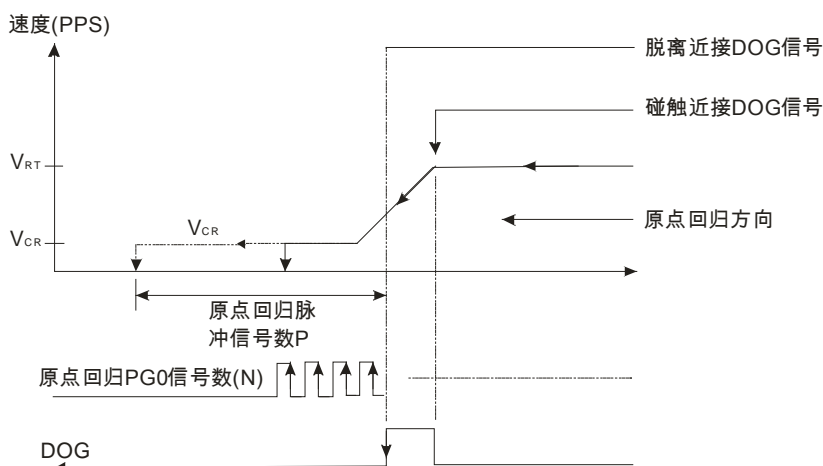
原点回归动作步骤：

1. 电机以原点回归速度 V_{RT} 运转
2. 当碰到近点 (DOG) 信号时，电机开始向原点回归减速速度 V_{CR} 做减速动作
3. 检测近点 (DOG) 信号下降沿产生后，此时经过 N 个原点回归之零点信号脉冲后，或经过 P 个原点回归之脉冲信号脉冲后电机停止。

若设定原点回归之 PG0 零点信号数 N 或原点回归之脉冲信号数 P 太小时，碰到近点 (DOG) 信号时，电机减速到原点回归减速速度 V_{CR} 运转，再检测近点 (DOG) 信号下降沿产生后，若达到 N 个指定之 PG0 零点信号，或达到 P 个原点回归之脉冲信号立即停止 (不论是否已到达原点回归减速速度 V_{CR})。

假设原点回归之信号数 N 设定为 0，原点回归之脉冲信号数 P 设定为 0，则电机检测近点 (DOG) 信号下降沿产生后立即停止。

7



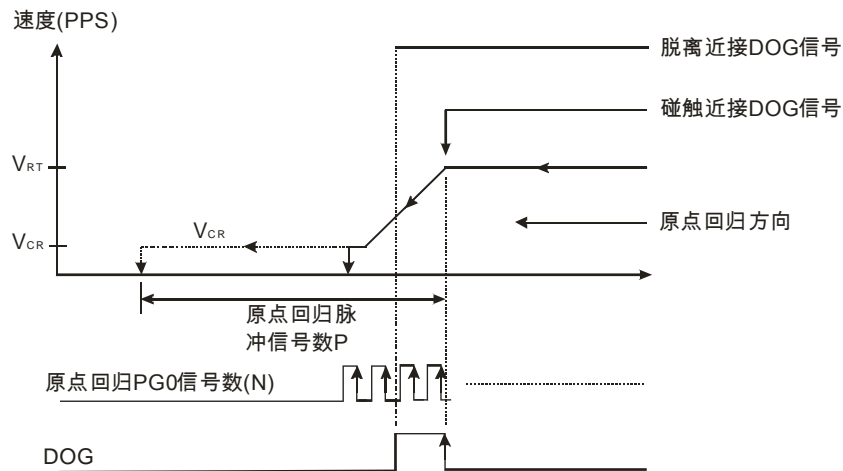
■ SR1000+100*N 之 b[9 : 10]=11⇒复写模式，原点回归 DOG 下降沿检测 OFF

原点回归动作步骤：

1. 电机以原点回归速度 V_{RT} 运转
2. 当碰到近点 (DOG) 信号时，电机开始向原点回归减速速度 V_{CR} 做减速动作，此时经过 N 个原点回归之零点信号脉冲后，或经过 P 个原点回归之脉冲信号后电机停止。

若设定原点回归之 $PG0$ 零点信号数 N 或原点回归之脉冲信号数 P 太小时，碰到近点 (DOG) 信号时，电机开始向原点回归减速速度 V_{CR} 做减速动作，若达到 N 个指定之 $PG0$ 零点信号，或达到 P 个原点回归之脉冲信号数立即停止（不论是否已到达原点回归减速速度 V_{CR} ）。

假设原点回归之信号数 N 设定为 0，原点回归之脉冲信号数 P 设定为 0，或电机碰到近点 (DOG) 信号后立即停止。



7.7 单段速定位模式使用介绍

7.7.1 专属对应的特殊寄存器

1. 目标位置 (I) P (I)

第 1+N 轴	
HW	LW
SR1023+100*N	SR1022+100*N

[说明]

- 设定范围-2,147,483,648~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 定义)。
- 目标位置 P (I) 数据属性：
 - (1) 绝对坐标 (SR1000+100*N 之 b12=0)
代表从 0 开始之位置，当目标位置 (I) P (I) 大于当前命令位置 (SR1032+100*N) 时，电机正转，小于当前命令位置时，电机反转。
 - (2) 相对坐标 (SR1000+100*N 之 b12=1)
代表从当前命令位置 (SR1032+100*N) 开始算电机行走之距离，当相对坐标为正值时，电机正转，相对坐标为负值时，电机反转。

2. 运转速度 (I) V (I)

第 1+N 轴	
HW	LW
SR1025+100*N	SR1024+100*N

[说明]

- 设定范围 0~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 定义)。
- 对应的脉冲指令 10~1000kPPS。若大于 1000k，则以 1000k 输出，若小于 10，则以 10 输出。
- 设定范围限制： $V_{MAX} > V (I) > V_{BIAS}$ 。
- 运转速度 (I) V (I) 在运转中可改变，若速度为 0 时运动停止，需重新启动。

3. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

7

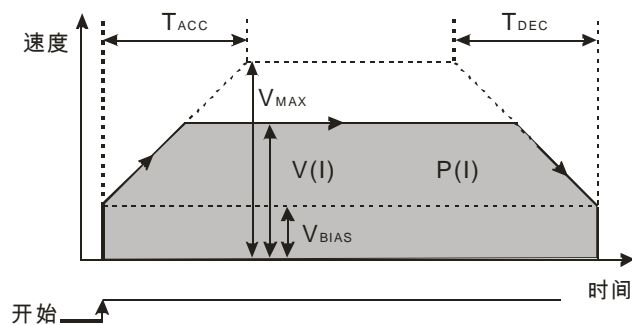
[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段速 插入定 位模式 启动	两段速 定位模 式启动	单段速 插入定 位模式 启动	单段速 定位运 动模式 启动	'	原点回 归模式 启动	手摇轮 输入操 作	变速度 运转模 式启动	JOG- 运转	JOG+ 运转	'	软件 Stop

其中单段速定位模式启动需设定 bit 8 为 1。

7.7.2 运动操作说明

- SR1030+100*N 之 bit 8=1 启动单段速定位模式，开始执行第 1 段定位程序，定位的步进数及运动速度根据【P(I)】和【V(I)】决定，开始由脉冲产生单元送出脉冲。
- 在相对坐标定位模式，运转方向由设定位置寄存器 P(I) 内容值之符号位决定；运转时，由启动速度 V_{BIAS} 加速至运转速度 V(I) 稳态运行，直到输出脉冲数接近目标位置 P(I) 时减速至启动速度 V_{BIAS} 后停止，其间总共发送产生出 P(I) 个脉冲数。
- 在绝对坐标定位模式，目标位置 P(I) 大于当前命令位置时正转，小于时反转；运转时，由启动速度 V_{BIAS} 加速至运转速度 V(I) 稳态运行，当当前命令位置接近目标位置 P(I) 时减速至启动速度 V_{BIAS} 停止于目标位置 P(I)。



7

7.8 插入单段速定位模式使用介绍

7.8.1 专属对应的特殊寄存器

1. 目标位置 (I) P (I)

第 1+N 轴	
HW	LW
SR1023+100*N	SR1022+100*N

[说明]

- 设定范围-2,147,483,648~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 定义)。
- 目标位置 P (I) 数据属性：
代表从 DOG 触发时的当前命令位置 (SR1032+100*N) 开始算电机行走之距离，当相对坐标为正值时，电机正转，相对坐标为负值时，电机反转。

2. 运转速度 (I) V (I)

第 1+N 轴	
HW	LW
SR1025+100*N	SR1024+100*N

[说明]

1. 运转速度 (I) V (I) 在运转中无法改变
2. 详细请参照第 7.7 章单段速定位模式。

3. 参数设定

第 1+N 轴	
HW	LW
-	SR1000+100*N

[说明]

1. 详细内容请参照本章节开头之常用特殊寄存器做功能详细说明。
- 插入单段速定位运动模式下：
 - Bit 13=0 时，DOG 信号上升沿触发。
 - Bit 13=1 时，DOG 信号下降沿触发。

4. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

7

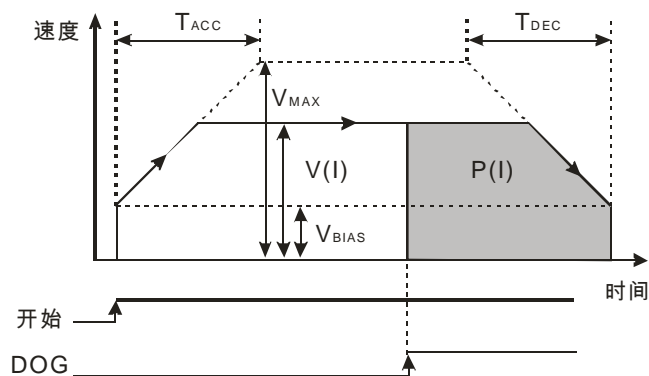
[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段速 插入定位 模式启动	两段速 定位模式 启动	单段速 插入定位 模式启动	单段速 定位运动 模式启动	'	原点回 归模式 启动	手摇轮 输入操 作	变速度 运转模 式启动	JOG- 运转	JOG+ 运转	'	软件 Stop

插入单段速定位运动模式启动需设定 bit 9 为 1。

7.8.2 运动操作说明

- SR1030+100*N 之 bit9=1 执行插入单段速运动；由脉冲产生单元送出脉冲，当外部近点 DOG 信号动作时，则以相对坐标的方式对目标位置 (I)【P(I)】进行定位，P(I)=0 启动时会错误。
- 运转方向：目标位置 P(I) 为正时时正转，为负时反转
- 操作步骤：
 1. 运转速度会由启动速度 V_{BIAS} 开始加速至运转速度 $V(I)$ 稳态运行
 2. 于脉冲送出期间遇到 DOG 近点信号触发时，脉冲产生单元会再送出位置寄存器 P(I) 内容值之步进数，同样地此时和单段速定位运动模式一样，产生的脉冲个数直到快到达位置寄存器 P(I) 之内容值时，减速至启动速度而后停。



7

7.9 连续两段速定位模式使用介绍

7.9.1 专属对应的特殊寄存器

1. 目标位置 (I) P (I)

第 1+N 轴	
HW	LW
SR1023+100*N	SR1022+100*N

[说明]

详细请参照第 7.7 章单段速定位模式。

2. 运转速度 (I) V (I)

第 1+N 轴	
HW	LW
SR1025+100*N	SR1024+100*N

[说明]

- 运转速度 (I) V (I) 在运转中无法改变。
- 详细请参照第 7.7 章单段速定位模式。

3. 目标位置 (II) P (II)

第 1+N 轴	
HW	LW
SR1027+100*N	SR1026+100*N

[说明]

- 设定范围-2,147,483,648~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 定义)。
- 目标位置 P (II) 数据属性：
 - 绝对坐标 (SR1000+100*N 之 b12=0) : P (II) 要在当前命令位置到 P (I) 延伸的位置。
 - 相对坐标 (SR1000+100*N 之 b12=1) : P (I) P (II) 要同号同方向

4. 运转速度 (II) V (II)

第 1+N 轴	
HW	LW
SR1029+100*N	SR1028+100*N

[说明]

1. 设定范围 0~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 定义)。
2. 对应的脉冲指令 10~1000kPPS。若大于 1000k，则以 1000k 输出，若小于 10，则以 10 输出。
3. 设定范围限制： $V_{MAX} > V (II) > V_{BIAS}$ 。
4. 运转速度 (II) V (II) 在运转中无法改变。

7

5. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

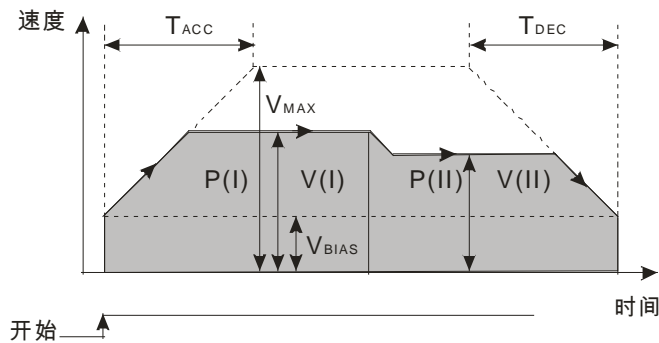
[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	XO 启动	两段速插入定位模式启动	两段速定位模式启动	单段速插入定位模式启动	单段速定位运动模式启动	'	原点回归模式启动	手摇轮输入操作	变速度运转模式启动	JOG- 运转	JOG+ 运转	'	软件 Stop

其中连续两段速定位运动模式启动需设定 bit 10 为 1。

7.9.2 运动操作说明

1. SR1030+100*N 之 bit 10=1 启动连续两段速度定位运动；以运转速度 $V(I)$ 运行，到达第 1 段目标位置 $P(I)$ 后，立即以运转速度 $V(II)$ 运行至第 2 段目标位置 $P(II)$ 。
2. 相对坐标定位模式下，运转方向由目标位置 $P(I)$ 内容值之符号位决定，设定正数则正转，设定负数则反转 $P(I)$ $P(II)$ 要同号；运转时，由启动速度 V_{BIAS} 加速至运转速度 $V(I)$ 稳态运行，直到输出目标位置 $P(I)$ 的脉冲数后，此时会由运转速度 $V(I)$ 加速/减速至运转速度 $V(II)$ 稳态运行，当第二段输出脉冲数接近目标位置 $P(II)$ 时减速至启动速度 V_{BIAS} 停止，其间总共发送产生出 $P(I) + P(II)$ 个脉冲数。
3. 绝对坐标定位模式下，目标位置 $P(I)$ 大于当前位置时正转，小于时反转， $P(I)$ 要在当前位置与 $P(II)$ 间；运转时，由启动速度 V_{BIAS} 加速至运转速度 $V(I)$ 稳态运行，直到当前位置等于目标位置 $P(I)$ ，此时会由运转速度 $V(I)$ 加速/减速至运转速度 $V(II)$ 稳态运行，当当前位置接近目标位置 $P(II)$ 时减速至启动速度 V_{BIAS} 停止于目标位置 $P(II)$ 。



7.10 插入两段速定位使用介绍

7.10.1 专属对应的特殊寄存器

1. 运转速度 (I) V (I)

第 1+N 轴	
HW	LW
SR1025+100*N	SR1024+100*N

[说明]

1. 运转速度 (I) V (I) 在运转中无法改变
2. 详细请参照第 7.7 章单段速定位模式。

2. 运转速度 (II) V (II)

第 1+N 轴	
HW	LW
SR1029+100*N	SR1028+100*N

[说明]

详细请参照第 7.9 章连续两段速定位模式。

3. 目标位置 (II) P (II)

第 1+N 轴	
HW	LW
SR1027+100*N	SR1026+100*N

[说明]

1. 设定范围-2,147,483,648~+2,147,483,647 (单位由 SR1000+100*N 之 b0 · b1 定义)。
2. 目标位置 P (II) 数据属性：
代表从 DOG 触发时的当前命令位置 (SR1032+100*N) 开始算电机行走之距离，当相对坐标为正值时，电机正转，相对坐标为负值时，电机反转。

4. 运转命令

第 1+N 轴	
HW	LW
-	SR1030+100*N

7

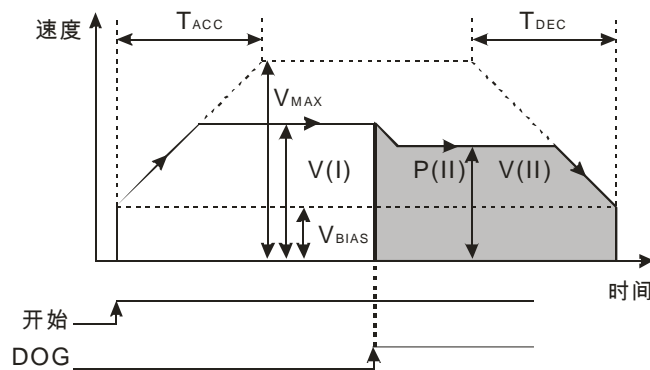
[说明]

特殊寄存器名称															
SR1030+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	'	'	OX 启动	两段速 插入定 位模式 启动	两段速 定位模 式启动	单段速 插入定 位模式 启动	单段速 定位运 动模式 启动	'	原点回 归模式 启动	手摇轮 输入操 作	变速度 运转模 式启动	JOG- 运转	JOG+ 运转	'	软件 Stop

插入两段速定位运动模式启动需设定 bit 11 为 1。

7.10.2 运动操作说明

1. SR1030+100*N 之 b11=1 启动外部两段速定位运动；第 1 段位置定位期间以运转速度 V (I) 运行，当外部 DOG 信号触发，会立即以运转速度 V (II) 以相对坐标的方式对目标位置 (II)【P (II)】进行定位。在 DOG 触发前变更 V (I) DOG 触发时运动会停止，在 DOG 触发后变更 V (I)，V (I) 会取代 V (II) 成为定位运动的速度。
2. 目标位置 P (II) 为正时正转，为负时反转。
3. 运转中，由启动速度 V_{BIAS} 开始加速至运转速度 V (I) 后稳态运行，当外部 DOG 信号触发时，立即加速/减速至运转速度 V (II) 后稳态运行至 P (II) 结束。



7

7.11 状态标志与寄存器

1. Ready 标志

第 1+N 轴	
HW	LW
-	SM1048+100*N

[说明]

1. 各轴各有 Ready 标志，第 1 轴：SM1048，第 2 轴：SM1148，第 3 轴：SM1248...。可用来判断行程是不是还在运转。
2. 以下为第 1 轴 Ready 标志动作说明：第 1 轴在运转前 SM1048 为 ON，开始运转后 SM1048 由 ON→OFF，运转完后则由 OFF→ON。

2. 运动错误标志

第 1+N 轴	
HW	LW
-	SM1017+100*N

[说明]

各轴发生运动错误时，会将错误信息分别存放至各个相对应的寄存器之中。

欲进行除错时，请清除错误信息寄存器与错误标志。

3. 执行状态

第 1+N 轴	
HW	LW
-	SR1040+100*N

[说明]

特殊寄存器装置															
SR1040+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
.			行程暂停中	错误产生	行程动作中	反向脉冲输出中	正向脉冲输出中

4. 错误编号

第 1+N 轴	
HW	LW
-	SR1041+100*N

[说明]

请参照附录。

7

MEMO

7

8

第8章 电子凸轮功能

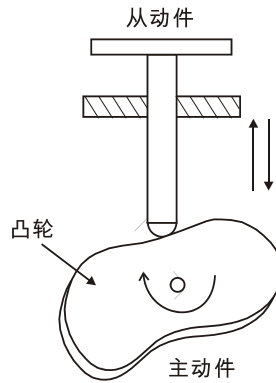
目录

8.1	电子凸轮 (E-CAM) 简介	8-2
8.2	电子凸轮的实现方式	8-3
8.2.1	初始设定	8-3
8.2.1.1	建立电子凸轮数据	8-3
8.2.1.2	输入/出脉冲形式设定	8-4
8.2.2	主从轴设定及操作	8-5
8.2.2.1	主轴来源设定	8-6
8.2.2.2	主轴启动角度设定	8-6
8.2.2.3	从轴输出设定	8-7
8.2.3	周期式电子凸轮启动/停止	8-8
8.2.3.1	周期式电子凸轮启动	8-8
8.2.3.2	周期式电子凸轮停止	8-9
8.3	电子凸轮数据 (电子凸轮数据) 建立	8-11
8.3.1	PMSOFT CAM Chart 绘制电子凸轮数据	8-11
8.3.1.1	根据主从轴的标准函数关系绘制	8-11
8.3.1.2	主从轴根据实际工作的点对点关系绘制	8-15
8.3.1.3	单笔电子凸轮数据 (电子凸轮数据) 建立/变更	8-17
8.4	电子凸轮应用-飞剪应用	8-19
8.4.1	飞剪电子凸轮数据建立	8-21
8.4.2	功能块-凸轮飞剪曲线生成操作	8-21
8.4.3	功能块-凸轮动态变更操作	8-26
8.4.4	飞剪电子凸轮范例	8-26

8.1 电子凸轮 (E-CAM) 简介

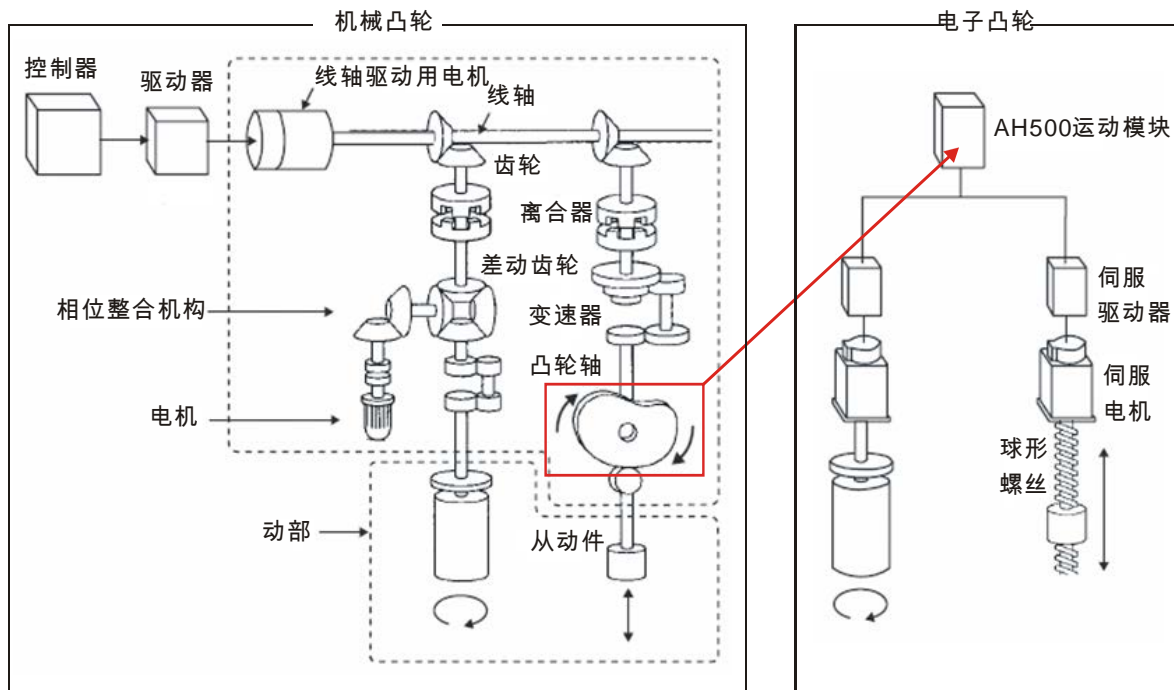
传统的机械凸轮机构，是由凸轮、从动件以及机架等三部份组成。

1. 机械凸轮是一种不规则形状的机件，一般为等转速的输入件，可经由直接接触传递运动到从动件，使从动件依设定的规律运动。
2. 从动件为机械凸轮所驱动的被动物件，一般为产生不等速、不连续、不规则运动的输出件。
3. 机架则是用来支持机械凸轮与从动件的机件。

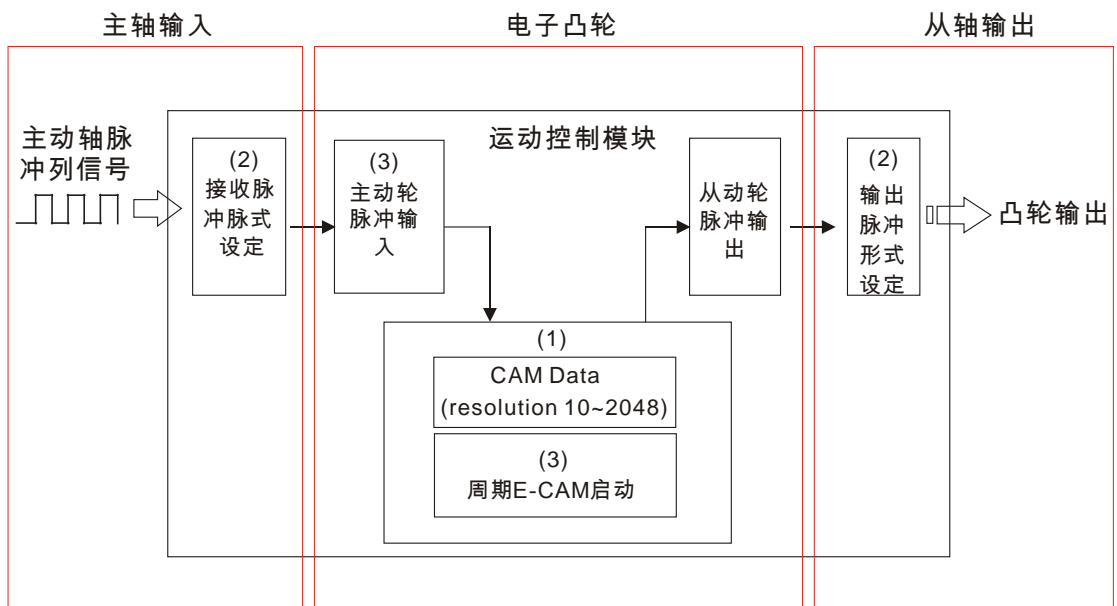


相对传统的方式，使用电子凸轮有着以下的好处：

1. 较友善的用户接口
2. 不同的产品需要不同的凸轮曲线，可利用软件修改电子凸轮表内电子凸轮数据，不用修改机构
3. 可以有较高的加速
4. 运行可以较为平顺



8.2 电子凸轮的实现方式



步骤一	步骤二	步骤三
初始设定	主从轴信号设定	电子凸轮启动/停止
建立电子凸轮数据 (1) 输入/出脉冲形式设定 (2)	主轴来源设定 (3) 主轴启动角度设定 (3) 从轴输出设定 (3)	周期性凸轮启动/停止 (3)

8.2.1 初始设定

8.2.1.1 建立电子凸轮数据

电子凸轮的曲线数据建立可透过以下两种途径绘制：

途径 1：根据主从轴的标准函数关系。

途径 2：根据实际工作中测量到的点与点之间的对应关系。

建立的详细步骤参照第 8.3 章

8.2.1.2 输入/出脉冲形式设定

1. 输入脉冲形式设定

电子凸轮的主轴来源可分为 MPG0，以及模块本体的运动轴 1 到运动轴 16，或为计数器 C200、C204、C208、C212、C216、C220。如用户利用计数器做为主轴来源时则需设定输入脉冲形式，此设定可透过计数器功能块，针对指定的计数器设定输入的脉冲形式来源。

En	T_HCnt	Eno
Channel		Valid
Enable		Busy
ExtRstEN		Error
InputType		CountValue
InitialValue		

2. 输入引脚 Input Type 输入值定义如下：

输入值	输入脉冲型式 (正逻辑)	说明
mcUD(0)	<p>FP正转脉冲 RP反转脉冲</p>	双脉冲 (FP/RP)
mcPD(1)	<p>Fp脉冲 RP方向(DIR) 正转 反转</p>	单脉冲 (P/D)
mcAB(2)	<p>FP A相脉冲 RP B相脉冲 正转 反转</p>	A/B 相脉冲 (1 倍频)
mc4AB(3)	<p>FP A相脉冲 RP B相脉冲 正转 反转</p>	A/B 相脉冲 (4 倍频)



3. 输出脉冲形式设定

凸轮脉冲输出形式，功能仅需脉冲型运动控制器设定，如 AH10PM 系列，通讯型运动控制器则不需作此设定。设定方式为透过功能块可设定指定的运动轴其控制脉冲形式。

En	T_AxisSetting2	Eno
Axis		Done
Execute		Busy
Vcurve		Error
OutputType		
Unit		
PulseRev		
DistanceRev		

4. 输入引脚 OutputType 输入值定义如下：

输入值	脉冲输出型式 (正逻辑)	说明
mcUD(0)	<p>FP正转脉冲</p> <p>Rp正转脉冲</p>	双脉冲 (FP/RP)
mcPD(1)	<p>FP脉冲</p> <p>RP方向(DIR)</p> <p>正转</p> <p>反转</p>	单脉冲 (P/D)
mcAB(2)	<p>FP A相脉冲</p> <p>RP B相脉冲</p>	A/B 相脉冲 (AB)
mc4AB(3)	<p>正转</p> <p>反转</p>	

8.2.2 主从轴设定及操作

电子凸轮的主从轴设定与电子凸轮的操作，仅需透过 CamIn 功能块即可达成。

En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop		CycleStartFlag
MasterOffset		Index
MasterScaling		InputPulses
SlaveScaling		InputFreq

在输入引脚设定完成后，透过 Enable = TRUE，则设定的功能即开始运作，运作状态提供输出引脚作监控标志使用。

8.2.2.1 主轴来源设定

透过 CamIn 功能块中，透过输入引脚 Master 决定主轴来源。

En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop	CycleStartFlag	
MasterOffset		Index
MasterScaling		InputPulses
SlaveScaling		InputFreq

输入引脚 Master 输入值定义如下：

输入值	定义	说明
0	MPG	外部手轮为主轴信号
1~16	运动轴	以模块内部运动轴 1 ~ 16 做为主轴信号来源
200	计数器	以计数器 C200 做为主轴信号来源
204	计数器	以计数器 C204 做为主轴信号来源
208	计数器	以计数器 C208 做为主轴信号来源
212	计数器	以计数器 C212 做为主轴信号来源
216	计数器	以计数器 C216 做为主轴信号来源
220	计数器	以计数器 C220 做为主轴信号来源

8.2.2.2 主轴启动角度设定

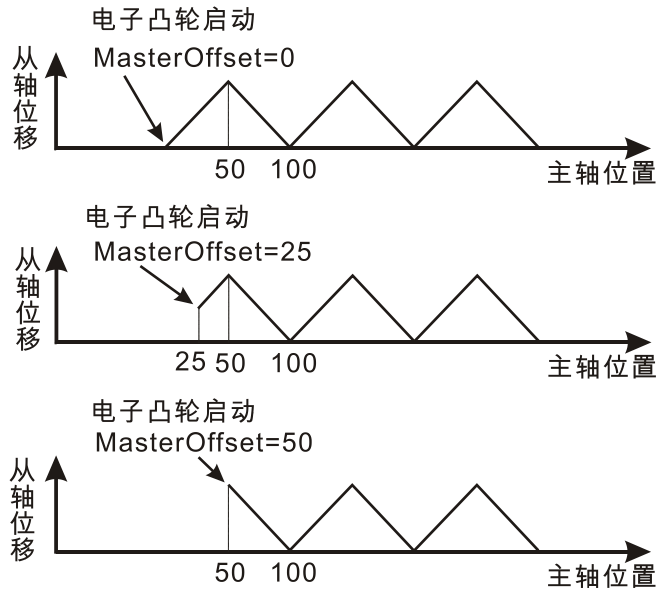
透过 CamIn 功能块中，透过输入引脚 MasterOffset 设定凸轮运动的主轴起始角度。



En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop	CycleStartFlag	
MasterOffset		Index
MasterScaling		InputPulses
SlaveScaling		InputFreq

输入引脚 Master 输入值定义如下：

输入值	定义	说明
K0~ K2,147,483,647	起始角度	此设定值单位 pulse，以此设定值的主轴位置与从轴位置，为电子凸轮运动曲线的起始点。



8.2.2.3 从轴输出设定

透过 CamIn 功能块中，透过输入引脚 Slave 决定从轴。

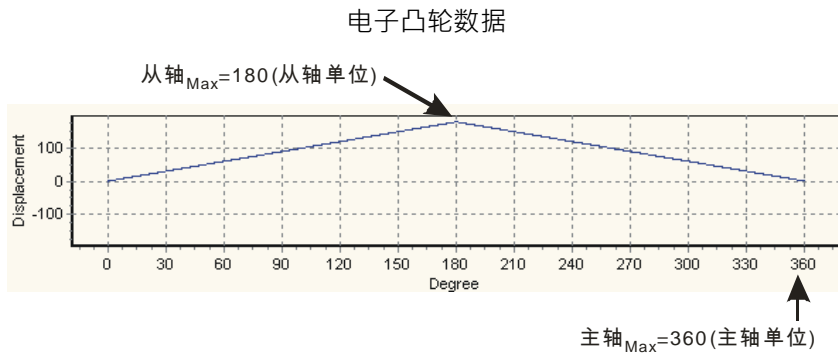
En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop		CycleStartFlag
MasterOffset		Index
MasterScaling		InputPulses
SlaveScaling		InputFreq

输入引脚 Slave 输入值定义如下：

输入值	定义	说明
1~16	运动轴	以模块内部运动轴 1 ~ 16 做为从轴输出

8.2.3 周期式电子凸轮启动/停止

周期式电子凸轮指的是在主轴不断前进的同时，凸轮轴会依电子凸轮数据（电子凸轮数据）实现相对应的位置，但电子凸轮数据只定义了一个周期的数据，所以在此模式下主/从轴的位置关系就是电子凸轮数据的不断重复延伸。



8.2.3.1 周期式电子凸轮启动

CamIn 功能块在利用输入脚设定电子凸轮的主从轴，以及启动角度后，设定输入脚 Enable 为 1 后，开始启动周期式电子凸轮。

En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop	CycleStartFlag	
MasterOffset		Index
MasterScaling		InputPulses
SlaveScaling		InputFreq

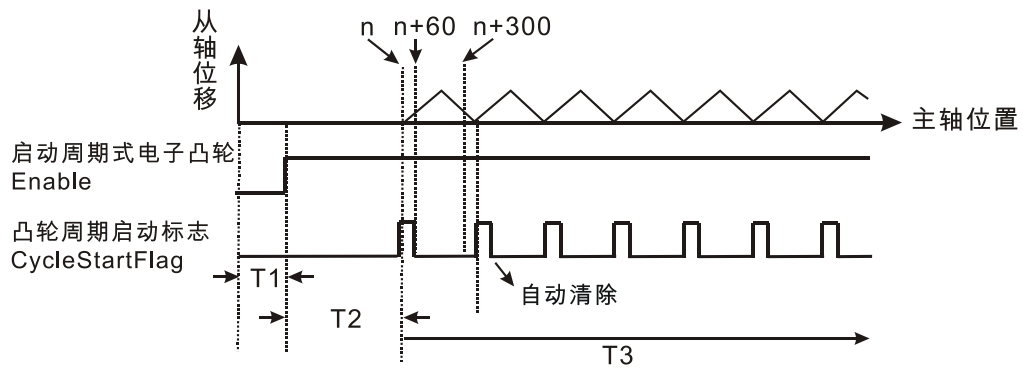
输出引脚 CycleStartFlag 定义如下：

输出值	定义	说明
TRUE/FALSE	周期启动标志	此标志表示凸轮该周期启动，并且标志 ON 后在程序的下一个周期时自动清除

以下周期式电子凸轮启动顺序。

- 在时间 T1 时，Enable= TRUE，启动周期式电子凸轮。
- 经过时间 T2 后，主动设定 CycleStartFlag（电子凸轮周期启动标志），此标志在程序执行一个周期后自动清除。
- 在时间 T3 期间，电子凸轮初始化完成开始周期性动作。从轴开始依照电子凸轮数据的位置关系跟随主轴的动作。





8.2.3.2 周期式电子凸轮停止

电子凸轮停止，透过操作 CamIn 功能块的输入脚 Enable、CamOut。

En	T_CamIn	Eno
Master		Valid
Slave		Busy
Enable		Aborted
Reset		Error
CamOut		InCam
CycleStop	CycleStartFlag	
MasterOffset		Index
MasterScaling	InputPulses	
SlaveScaling	InputFreq	

1. 输入 CamOut 引脚定义如下：

输入值	定义	说明
TRUE/FALSE	凸轮脱离	CamOut 表示该运动轴脱离电子凸轮模式

2. 输入 CycleStop 引脚定义如下：

输出值	定义	说明
TRUE/FALSE	整周期停止	当 Enable 为 FALSE 时，凸轮运动在周期结束后才停止动作

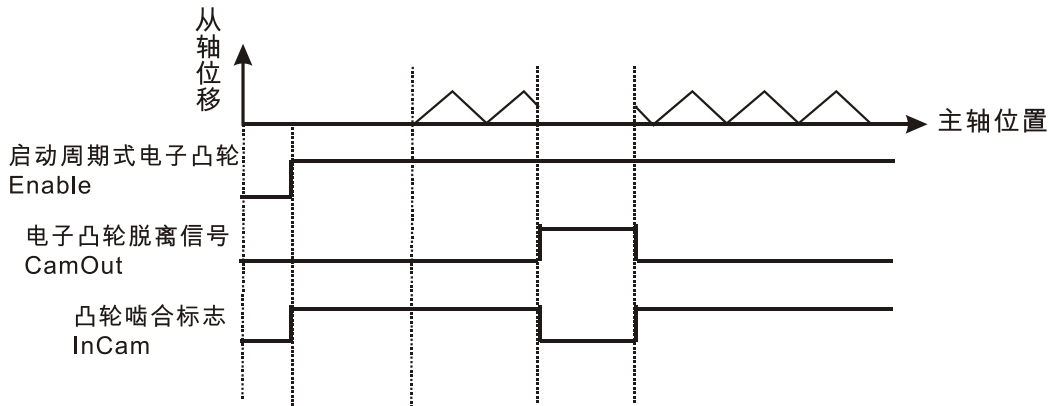
3. 输出 InCam 引脚定义如下：

输出值	定义	说明
TRUE/FALSE	凸轮啮合标志	此标志表示该运动轴目前为电子凸轮模式



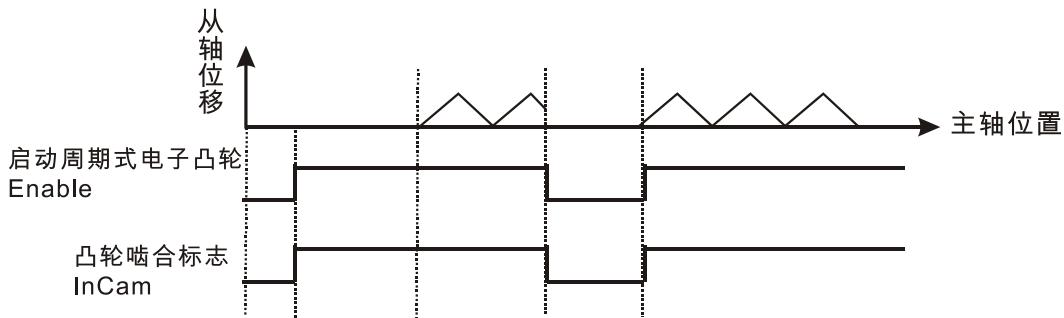
4. CamOut 停止电子凸轮动作

在 Enable 信号为 TRUE 时，设定 CamIn 功能块的 CamOut 信号 TRUE，此时从轴脱离电子凸轮模式，并且此时 InCam 标志为 FALSE，重新设定 CamOut 信号为 FALSE 时，从轴接续先前凸轮动作持续运行，同时 InCam 标志为 TRUE。

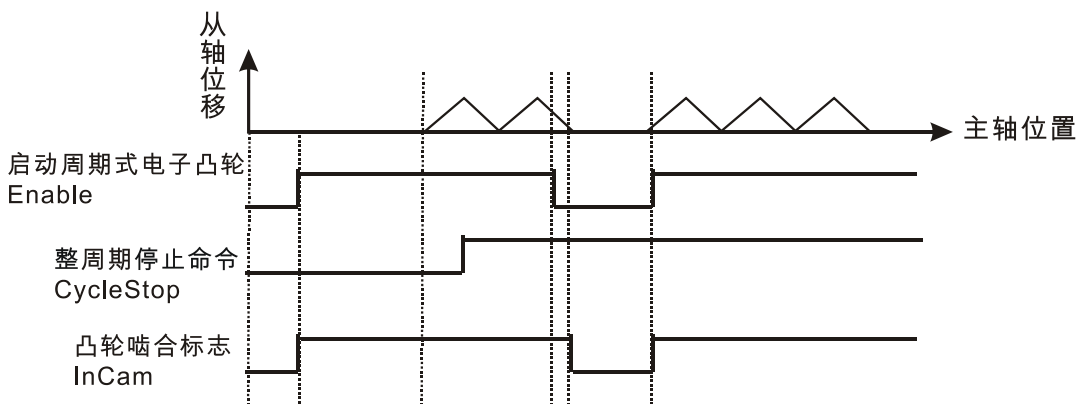


5. Enable 停止电子凸轮动作

在 Enable 信号为 TRUE 时，电子凸轮启动，设定 Enable 为 FALSE 电子凸轮停止动作，同时 InCam 信号为 FALSE，再重新设定 Enable 为 TRUE，电子凸轮重新由最初状态启动。



在 Enable 信号为 TRUE 时，电子凸轮启动，设定 CycleStop 为 TRUE 后将 Enable 设定为 FALSE，此时电子凸轮将在一个周期完成后停止动作，此时 InCam 信号为 FALSE，再重新设定 Enable 为 TRUE，电子凸轮重新由最初状态启动。



8.3 电子凸轮数据 (电子凸轮数据) 建立

定义主轴/从轴 (凸轮轴) 之间的关系，称之为电子凸轮数据 (电子凸轮数据)。

1. 使用 PMSoft CAM Chart 绘制，需要得到主从轴的关系，关系数据可透过以下两种途径获得：

- 途径 1：根据主从轴的标准函数关系。
- 途径 2：根据实际工作中测量到的点与点之间的对应关系。

E-CAM 主从轴关系确定后，根据主轴的位置，就能得到从轴的位置

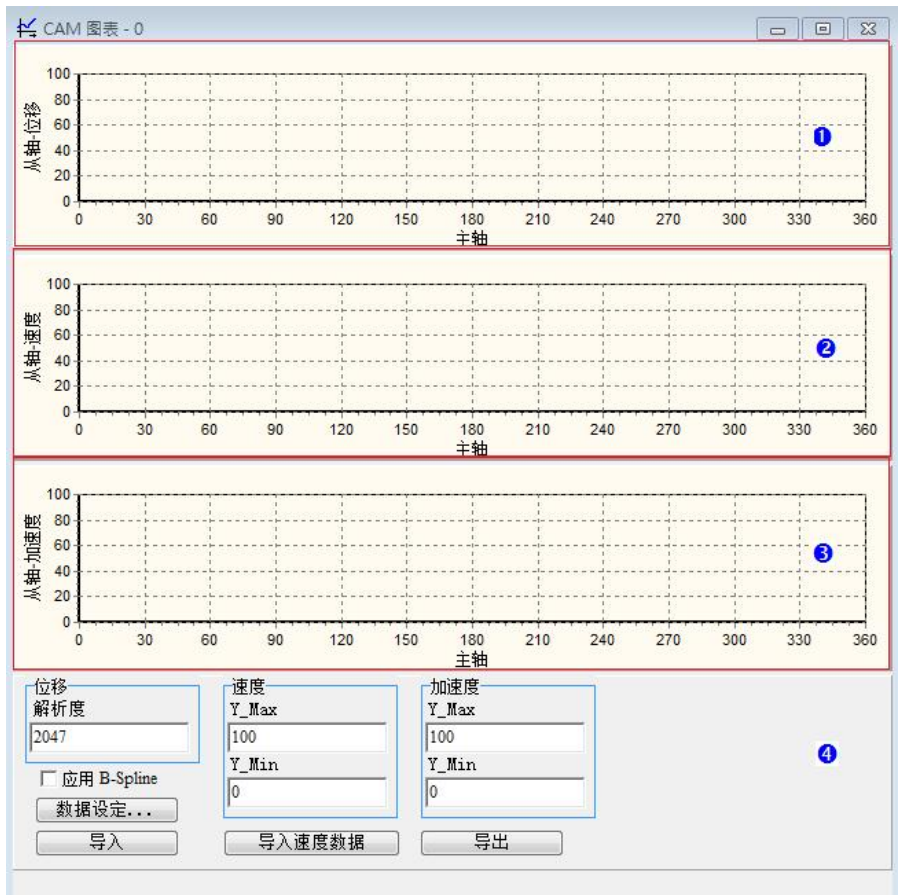
8.3.1 PMSoft CAM Chart 绘制电子凸轮数据

8.3.1.1 根据主从轴的标准函数关系绘制

建立专案后，右击鼠标单击系统信息区中的 **CAM 图表** 项目，开启快捷选单选择加入 **CAM 图表 (C)**，窗口会自动开启一空白的 CAM 图表供用户编辑。



CAM 图表如下图，说明如下。



- ❶ 位移关系图：主轴与从轴的相对位移数据。
- ❷ 速度关系图：主轴与从轴的相对速度数据。
- ❸ 加速度关系图：主轴与从轴的相对加速度数据。
- ❹ 数据设定区：
 - 位移解析度：供用户设定此电子凸轮总共所占用的数据点数，可设定范围从 10~2046 点。
 - 速度：此字段会显示速度坐标的最高与最低值，其数值则会根据位移数据，由系统自动计算。用户也可自行更改。
 - 加速度：此字段会显示加速度坐标的最高与最低值，其数值则会根据位移数据，由系统自动计算。用户也可自行更改。
 - 数据设定：开启**数据设定**窗口，用于描述主轴与从轴的位移变化。开启数据设定窗口时会带入数据设定区的位移解析度。当在数据设定区勾选**套用 B-Spline**时开启行程设定窗口将自动设为 **B-spline**。
 - 导入：自存文件导入主轴与从轴的位移变化关系。
 - 导出：将主轴与从轴的位移变化关系导出自存文件。
 - 导入速度数据：自文件导入主轴与从轴的速度变化关系。

单击数据设定按钮，开启数据设定窗口。数据设定的窗口是由数个区段所组成，每个区段提供用户设定一段凸轮曲线，再由数个区段组成整个凸轮曲线。用户可建立数个区段（最多 360 个），所有已建立的区段则构成一个电子凸轮周期，说明如下。



① 用户在此处输入各个区段的主轴与从轴关系。

- 主轴：主轴位移，单位为脉冲数，主轴为必为大于 0 的数值且为递增。
- 从轴：从轴位移，单位为脉冲数，从轴可为正或为负。
- CAM 曲线：目前区段所使用的函数，包括 Const Speed、Const Acc.、Single Hypot.、Cycloid 与 B-Spline。若用户于 CAM 图表窗口勾选**套用 B-Spline**时，开启行程设定窗口将在此处全部自动设为 B-spline。
- 解析度：目前区段所使用的数据点数，整个电子凸轮可设定范围共 10~2047 点。如果没有设定则 PMSoft 会由剩余的总点数均分。解析度设定依设备需求来设定，解析度越高设备运作较平滑，但电子凸轮数据占的容量越大。

② 完成区段建立后的各项功能，或加载储存的区段数据。

- 保存：将目前区段的设定储存
- 载入：加载储存的区段数据
- 清除：清除所有的区段数据
- 绘图：将区段的数据编译产生电子凸轮数据，且绘制在 CAM Chart 上。
- 确定：将区段的数据编译产生电子凸轮数据，且绘制在 CAM Chart 上，并关闭行程设定窗口。
- 取消：关闭行程设定窗口

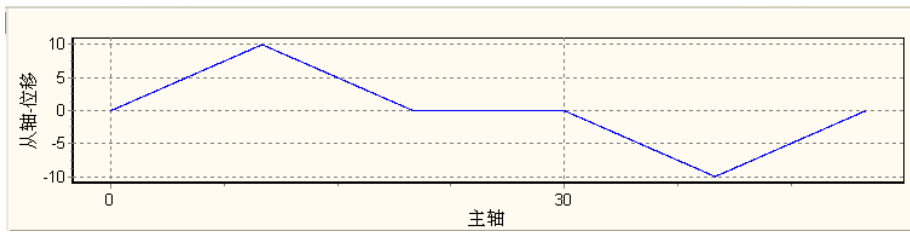
按下“导出”后，PMSoft 会将目前在 CAM Chart 的位移关系图、速度关系图及加速度关系图中的电子凸轮数据储存到 PMSoft 安装目录下 CAMData 的子目录，里面一共有 3 个文件分别为：

- <PMSoft 安装目录>\CAMData\Data_S.txt：点对点的位移数据
- <PMSoft 安装目录>\CAMData\Data_V.txt：点对点的速度数据
- <PMSoft 安装目录>\CAMData\Data_A.txt：点对点的加速度数据

Data_S.txt、Data_V.txt、Data_A.txt，如下图所示。

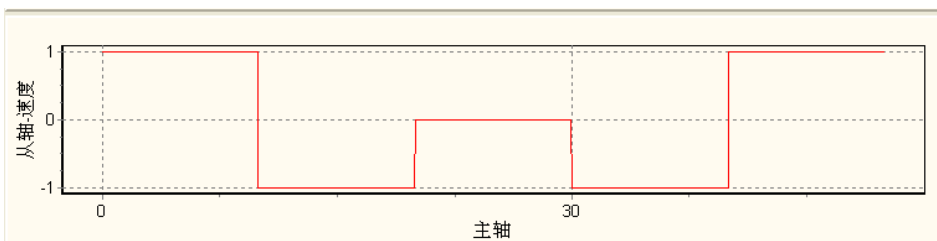
位移关系图

储存位置：<PMSoft 安装目录>\CAMData\Data_S.txt



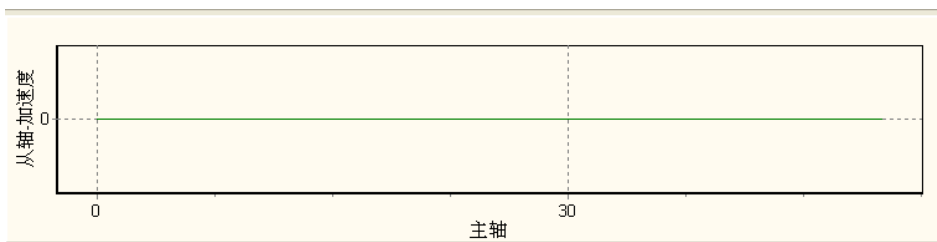
速度关系图

储存位置：<PMSoft 安装目录>\CAMData\Data_V.txt



加速度关系图

储存位置：<PMSoft 安装目录>\CAMData\Data_A.txt



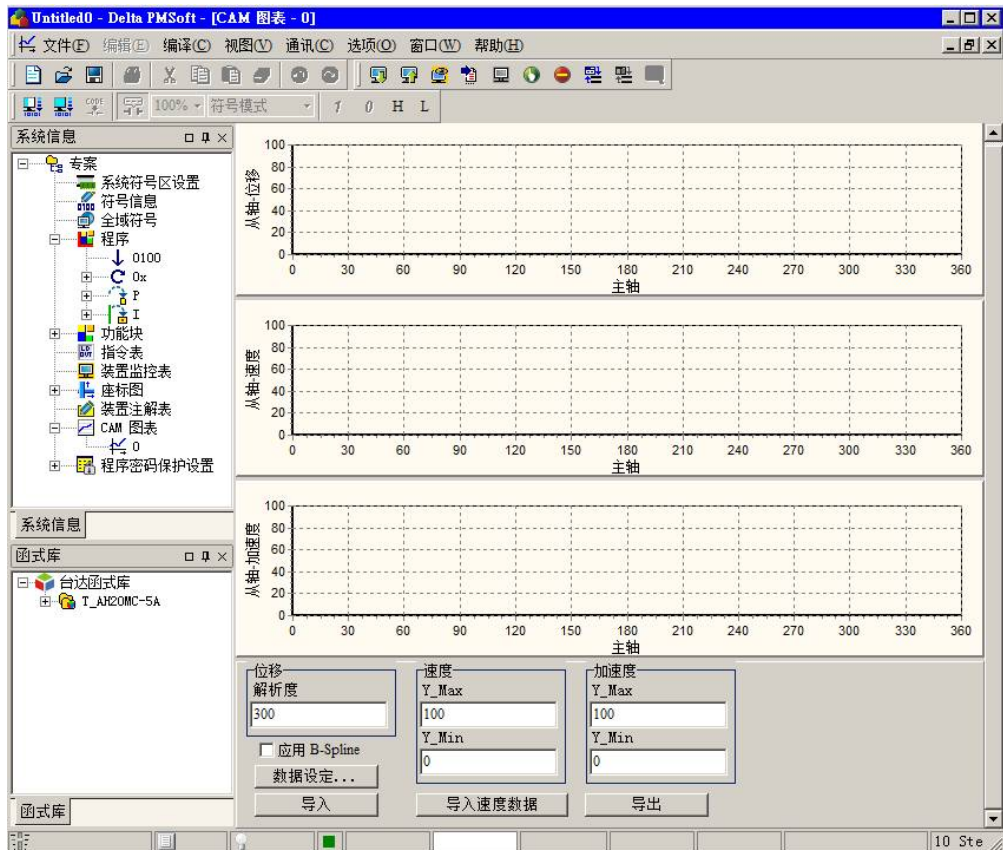
8.3.1.2 主从轴根据实际工作的点对点关系绘制

用户可将点对点数据输入至 PMSoft 的 CAMData 文件中，再将点对点的数据导入至 PMSoft CAM Chart 当中用图表表示。操作步骤如下：

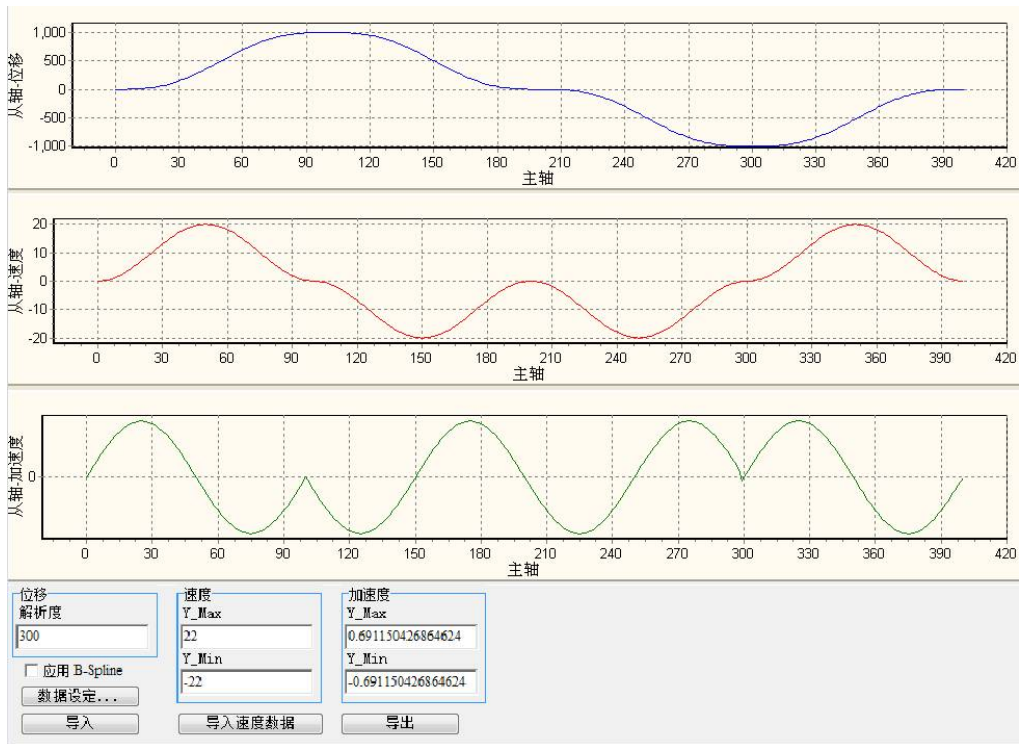
1. 在 PMSoft 安装目录下 CAMData 的子目录中的 Data_S.txt 输入点对点位移数据：



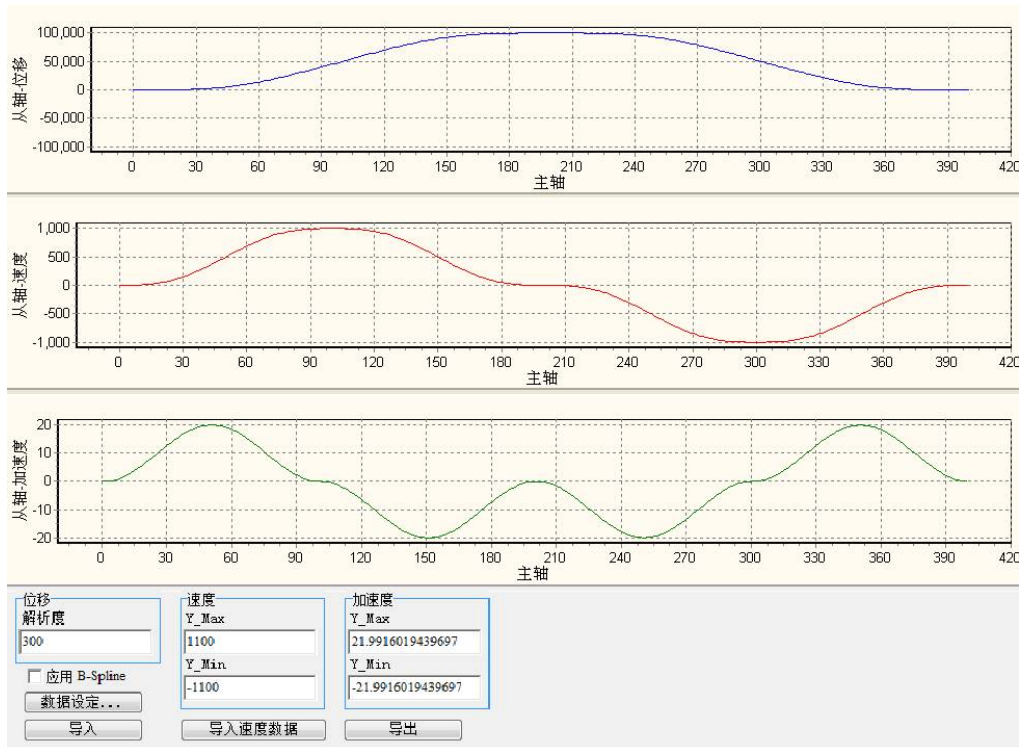
2. 在 PMSoft 的 CAM Chart 新增一个 Chart，填入解析度：



3. 按下“导入”后 PMSoft 会读取已存在 Data_S.txt 的内容，并绘制在 E-CAM Chart 的位移关系图上，同时速度关系图及加速度关系图会以 Data_S.txt 的数据计算出来一并绘制：



4. 按下“导入速度数据”后，PMSoft 会读取已存在 Data_S.txt 的内容，并绘制在 CAM Chart 的速度关系图上，同时位移关系图及加速度关系图会以 Data_S.txt 的数据计算出来一并绘制：



8

8.3.1.3 单笔电子凸轮数据 (电子凸轮数据) 建立/变更

使用 PMSoft CAM Chart 建立电子凸轮数据后，电子凸轮数据就跟着用户程序一起下载到 AH500 运动控制模块里，如要修改电子凸轮图表内的电子凸轮数据，则要在 PMSoft 上修改完后再次下载。除此方式外用户若要在程序中修改电子凸轮数据，可透过功能块变更单组电子凸轮数据。

单笔电子凸轮数据的操作方式，提供读取单笔数据功能块 **CamRead** 和写入单笔数据功能块 **CamWrite**

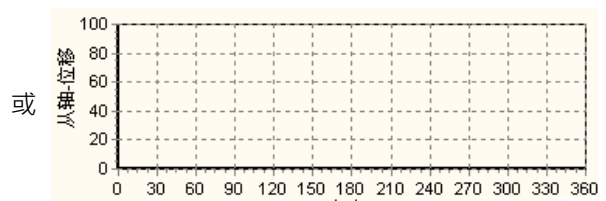
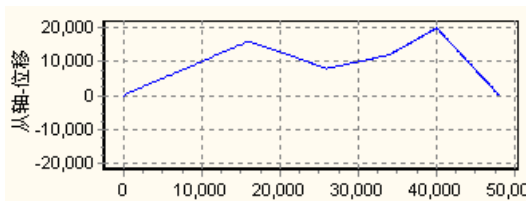
注意事项：若要修正凸轮曲线的所有数据点，请于最后一笔数据结束后再加上 (0, 0)。

CamRead	CamWrite																														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">En</td> <td style="width: 33%; text-align: center;">T_CamRead</td> <td style="width: 33%;">Eno</td> </tr> <tr> <td>Axis</td> <td></td> <td>Valid</td> </tr> <tr> <td>Enable</td> <td></td> <td>Error</td> </tr> <tr> <td>CamPointNo</td> <td>MasterPosition</td> <td>SlavePosition</td> </tr> </table>	En	T_CamRead	Eno	Axis		Valid	Enable		Error	CamPointNo	MasterPosition	SlavePosition	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">En</td> <td style="width: 33%; text-align: center;">T_CamWrite</td> <td style="width: 33%;">Eno</td> </tr> <tr> <td>Axis</td> <td></td> <td>Done</td> </tr> <tr> <td>Execute</td> <td></td> <td>Busy</td> </tr> <tr> <td>CamPointNo</td> <td></td> <td>Error</td> </tr> <tr> <td>MasterPosition</td> <td></td> <td></td> </tr> <tr> <td>SlavePosition</td> <td></td> <td></td> </tr> </table>	En	T_CamWrite	Eno	Axis		Done	Execute		Busy	CamPointNo		Error	MasterPosition			SlavePosition		
En	T_CamRead	Eno																													
Axis		Valid																													
Enable		Error																													
CamPointNo	MasterPosition	SlavePosition																													
En	T_CamWrite	Eno																													
Axis		Done																													
Execute		Busy																													
CamPointNo		Error																													
MasterPosition																															
SlavePosition																															

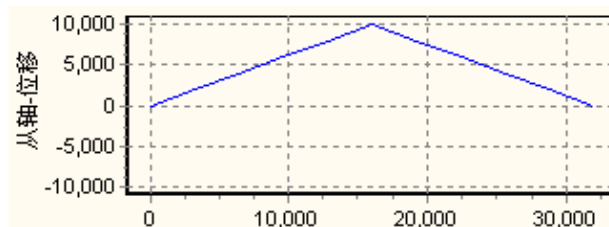
1. 范例说明

【功能说明】

使用单笔数据修改方式，变更原有 CAM Chart0 内容数据，如下图所示。图 (a) 表示原有的 CAM Chart0 内容，欲修改的 CAM Chart 当中可能已包含数据或是尚无数据；经单笔数据修改三点主从坐标点为 (16000,10000) 和 (32000,0)，如下图所示，最后结束时加上 (0,0) 结束凸轮执行，写入的数据点数为 3 点。



(a) 原有的 E-CAM Chart 内容

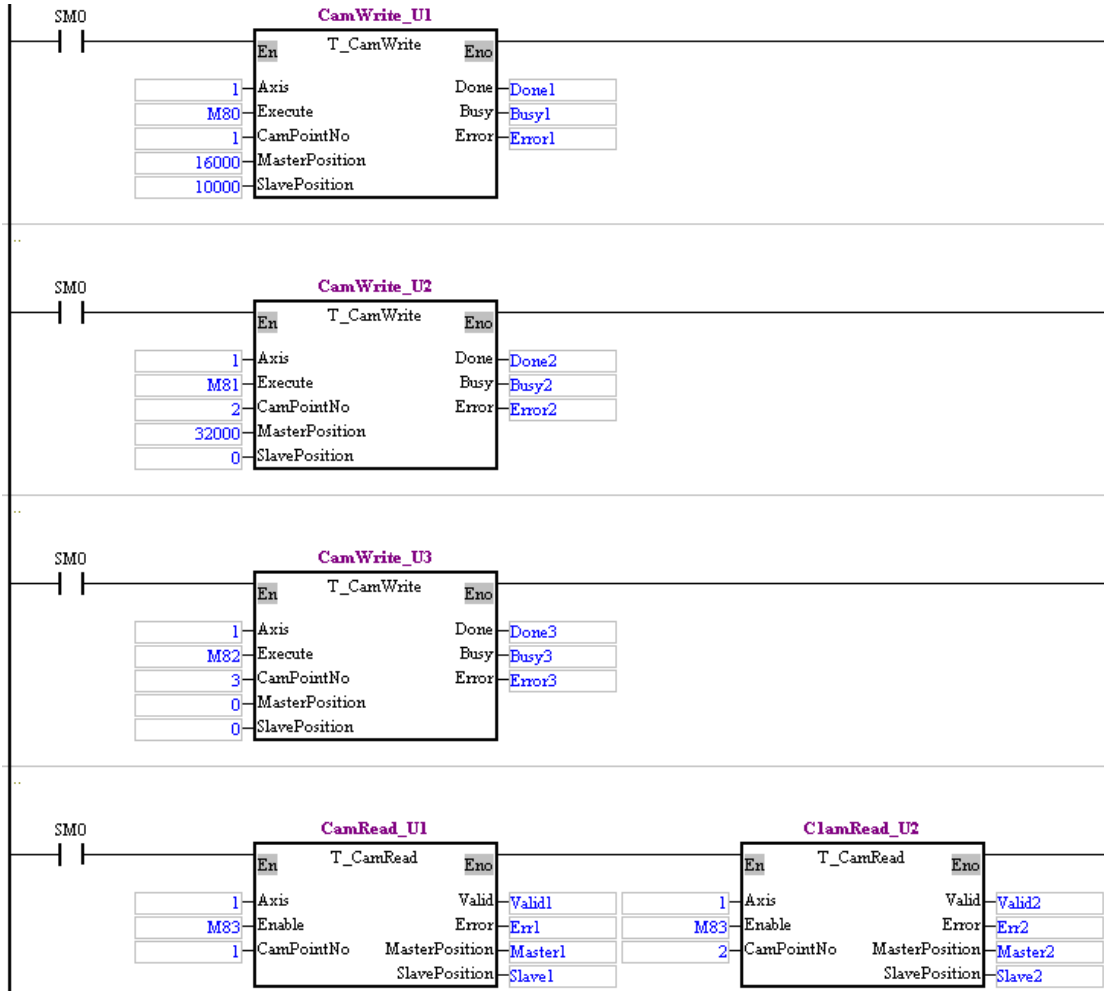


(b) 三点坐标组成的 E-CAM Chart

【操作步骤】

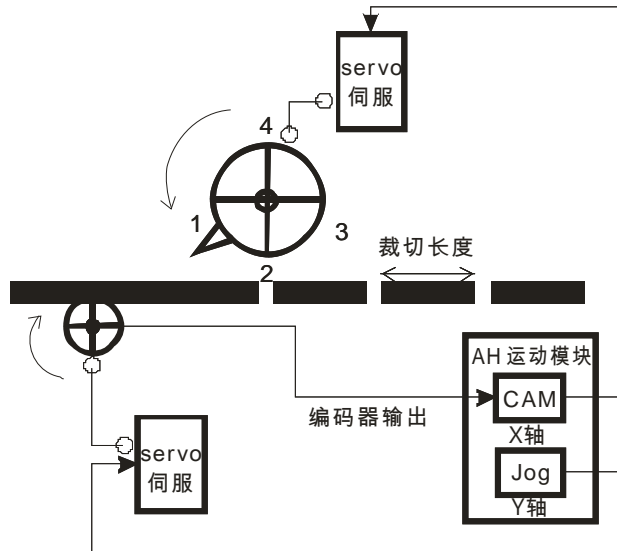
1. Set M80，将主从轴位置 (16000,10000) 写入 CamPointNo 1。
2. Set M81，将主从轴位置 (32000,0) 写入 CamPointNo 2。

3. Set M82 · 将主从轴位置 (0,0) 写入 CamPointNo 3 。
4. Set M83 · 将电子凸轮数据内的 CamPointNo 1 和 CamPointNo 2 读回 · 比较与写入数值是否相同 。



8.4 电子凸轮应用-飞剪应用

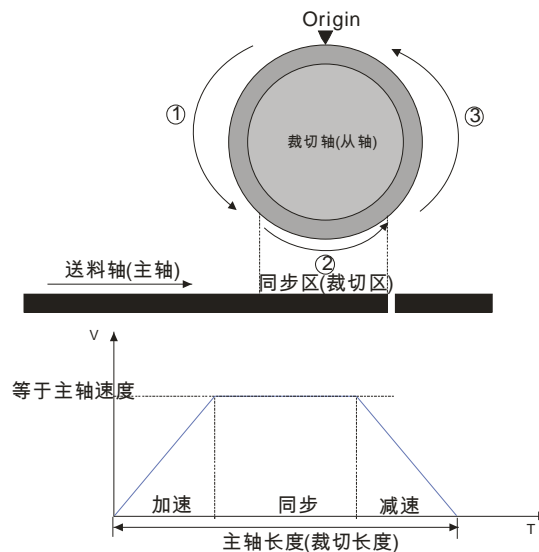
在送料裁切应用上，传统的做法是使用走停式，送料轴先走到固定的长度，随后剪料轴再动作，之后不断重复“送料停”及“剪料停”的过程。这种方法的缺点在送料轴走停的过程中，所需要的加减速使生产效益无法提高，因此新的做法是采用送料不停的方式来达成，一般有二种送料裁切的方法分别为飞剪跟追剪，两者的差异为追剪是往返运动，而飞剪是为同向运动，所设定的 CAM 表曲线也不同。以下将分别对飞剪应用详细说明：



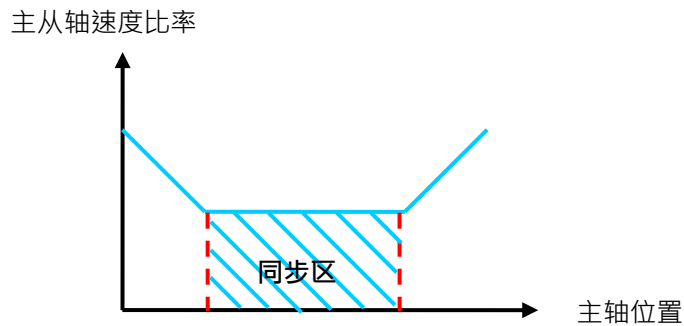
【观念说明】

1. 飞剪控制中裁切轴往同一方向旋转，利用刀具接触物料的时间点进行裁切，期间送料轴可不间断的持续等速送料。飞剪控制的动作与输出行程如下图所示：

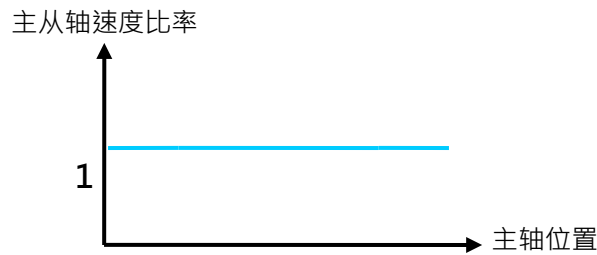
- 从轴一开始先加速移动至同步区
- 待离开同步区从轴便减速移回原点，完成一个周期裁切。得知行程后即可画成速度关系。



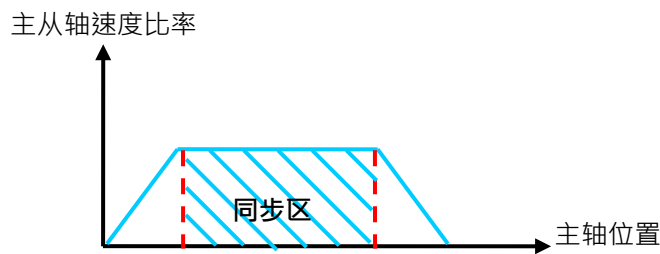
2. 在旋切过程中，最重要的是速度同步，比如在切刀接触到物料时一定要与物料速度同步，如果接触时切刀速度大于同步速度，出现对物料一个向前牵扯的力，会照成物料切面不平，如果速度低于物料速度，会出现堵料的现象。
3. 同步区的规划会影响到实际设备的运行，一个裁切周期中若同步区越大，加减速的时间就越小，表示设备需要在短时间内进行加减速，对于电机、机台、切刀的冲击都很大，而且容易导致伺服过流报警，设备无法正常运行。
4. 裁切长度与切刀周长的关系：
 - 裁切长度 < 切刀周长：在同步区切刀线速度与送料速度同步，过了同步区后，为了赶上下一次裁切，则裁切轴加速，如下图所示。



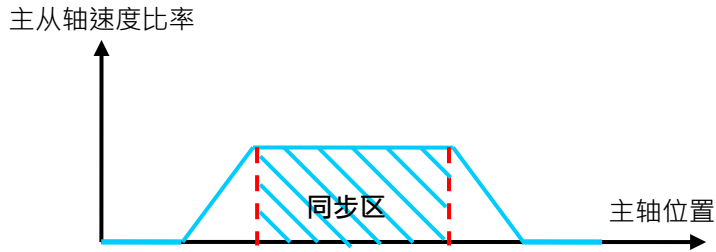
- 裁切长度 = 切刀周长：裁切轴匀速运动



- 1 倍切刀周长 < 裁切长度 < 2 倍切刀周长：于同步区裁切动作完成后，裁切轴减速，然后再加速到同步进行下次裁切，如下图所示。



- 裁切长度 > 2 倍切刀周长：在剪断长大于 2 倍刀周长情况下（这也是最常见的一种情况），在一个周期中，刀刃在同步区剪断完成后，减速到停止，等待一定长度过去后，启动下次裁切。

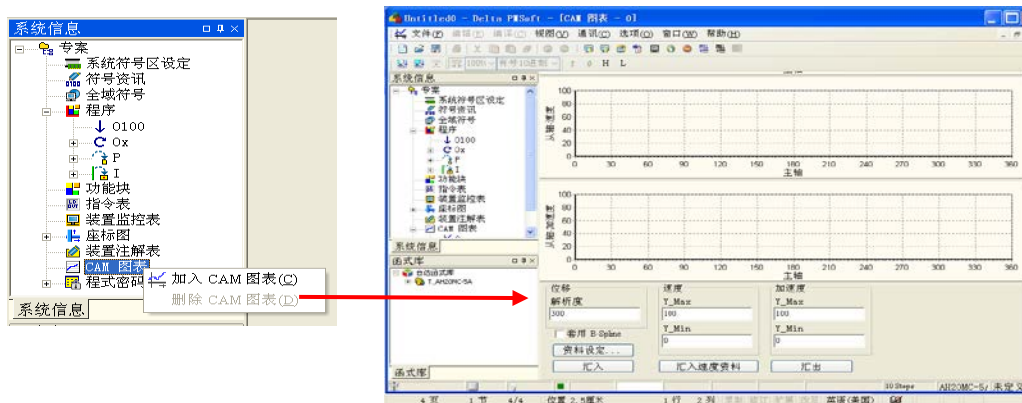


8.4.1 飞剪电子凸轮数据建立

飞剪曲线的生成除利用第 8.3 章的方式建立电子凸轮数据外，AH500 运动模块提供一运动功能块凸轮飞剪曲线生成，专门针对飞剪的凸轮数据作自动生成之用。

8.4.2 功能块-凸轮飞剪曲线生成操作

1. 开启一组 CAM Chart：在 PMSoft 开启一组 CAM Chart，解析度设定则依照生成的飞剪曲线数量而定，一个飞剪曲线需要 300 个解析度；再将这一组空白的 CAM Chart 下载至 AH500 模块当中，供执行中储存电子凸轮数据。



2. 设定功能块飞剪曲线生成：将实际应用飞剪机具参数、包括欲裁切的长度、同步长度以及同步比率输入至输入引脚，执行后生成飞剪曲线。

En	T_CamCurve	Eno
Axis		Done
Execute		Busy
MLength		Error
SLength		ErrNo
SSyncLength		SyncBegin
SSyncRatio		SyncEnd
SMaxRatio		
AccCurve		
eCamCurve		
Concatenate		

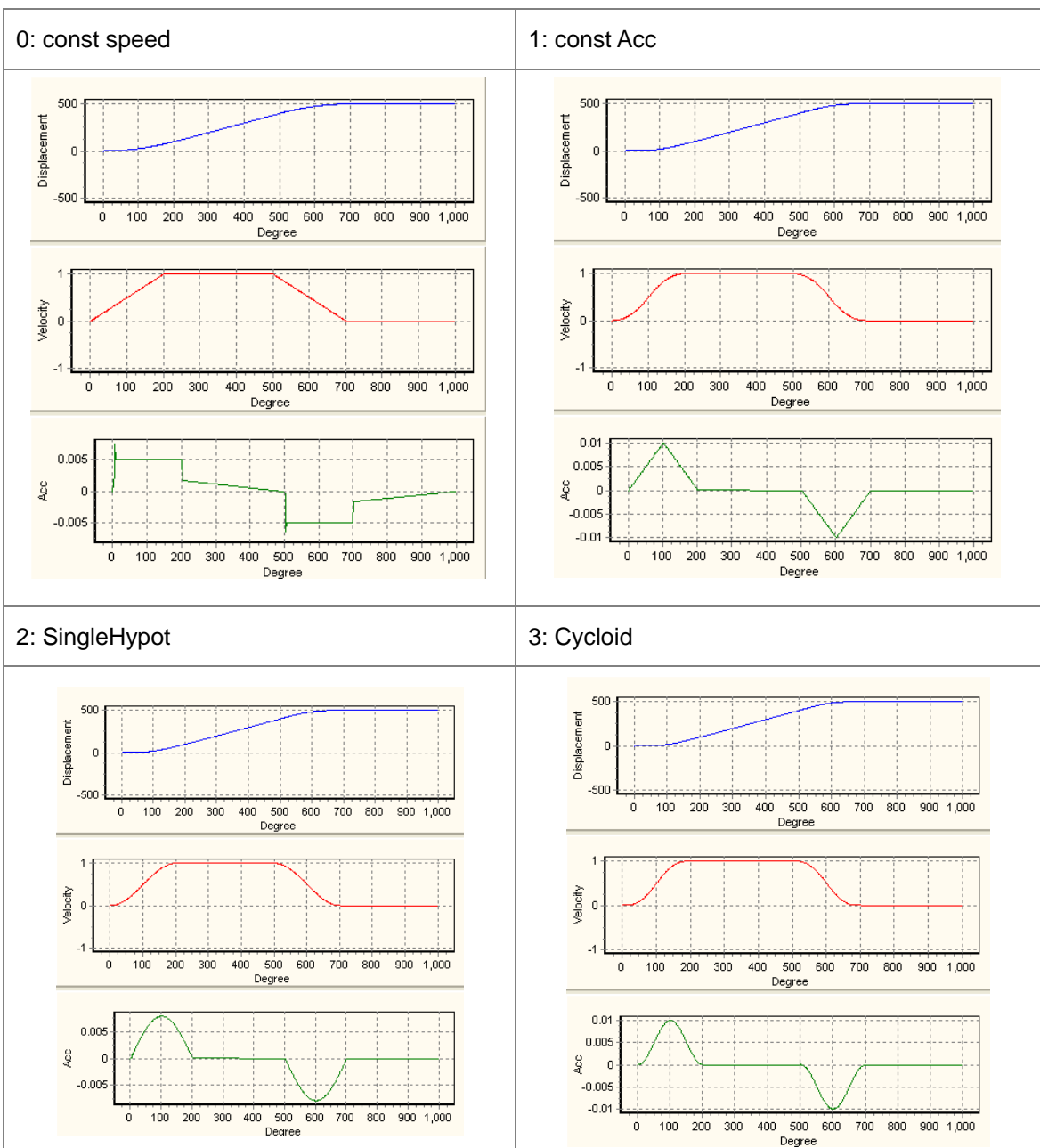
输入引脚				
引脚名称	功能说明	数据格式	允许范围设定值	说明
Axis	运动轴编号	WORD	1~16	电子凸轮从轴
Execute	上升沿时启动更新	BOOL	TRUE / FALSE	启动凸轮飞剪曲线生成
MLength	主轴长度	DWORD	K1 ~ K2,147,483,647	电子凸轮主轴长度 (PPS)
SLength	从轴长度	DWORD	K1 ~ K2,147,483,647	电子凸轮从轴长度 (PPS)
SSyncLength	从轴同步区长度	DWORD	K1 ~ K2,147,483,647	从轴同步区长度 (PPS)
SSyncRatio	从轴同步比率	Float	1.1755 x 10-38~ 3.4028 x 10+38	主从轴同步区速度比率 (从轴速度/主轴速度)
SMaxRatio	从轴最高比率限制	Float	1.1755 x 10-38~ 3.4028 x 10+38	主从轴同步区速度比率最高限制
AccCurve	加速曲线	WORD	0~3 (*1)	飞剪凸轮加速曲线
eCamCurve	CAM 曲线	WORD	0~5 (*2)	飞剪凸轮速度曲线
Concatenate	接续生成	BOOL	TRUE / FALSE	接续先前凸轮曲线

输出状态引脚				
引脚名称	功能说明	数据格式	输出数值范围	说明
Done	运动完成	BOOL	TRUE / FALSE	曲线生成完成
Busy	执行中	BOOL	TRUE / FALSE	曲线生成中
Error	功能块产生错误	BOOL	TRUE / FALSE	曲线生成错误

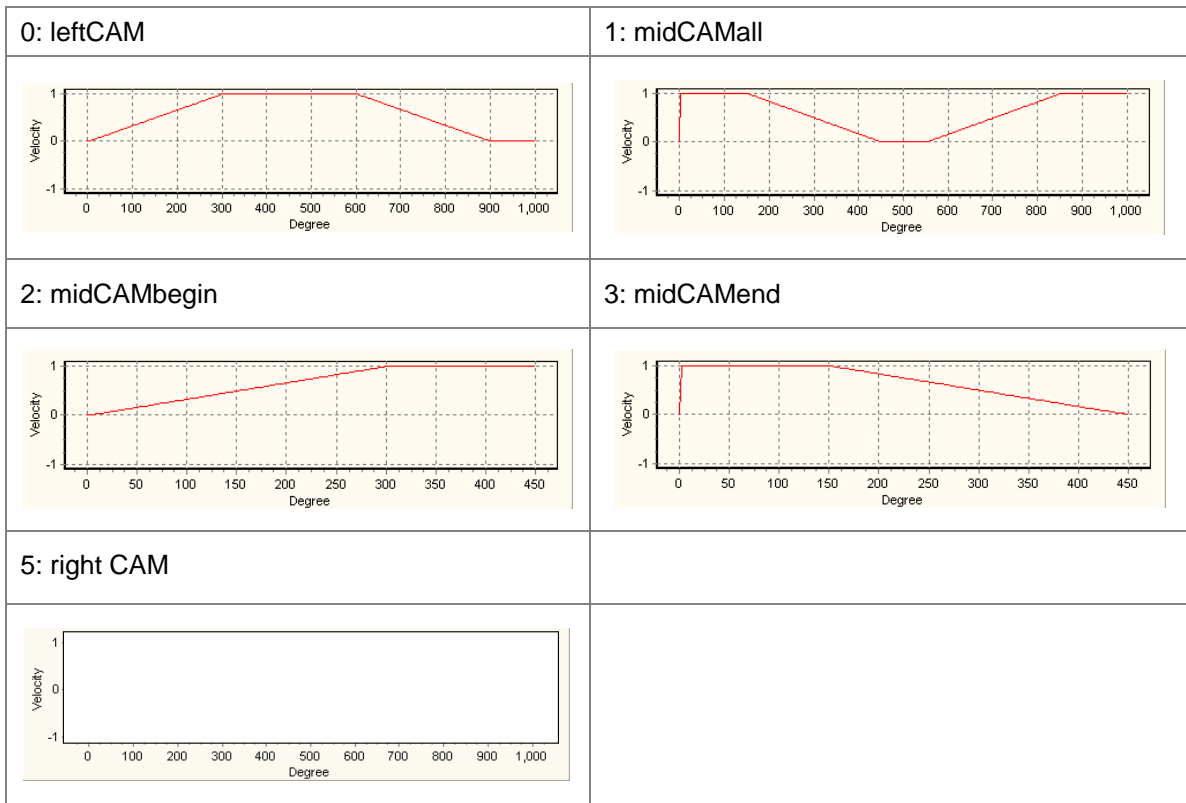


输出数值引脚				
引脚名称	功能说明	数据格式	输出数值范围	说明
ErrNo	错误码	WORD	0~2	曲线错误时错误码
SyncBegin	同步起点	DWORD	K 0 ~ K 2,147,483,647	曲线同步区起点
SyncEnd	同步终点	DWORD	K 0 ~ K 2,147,483,647	曲线同步区终点

(*1) accCurve 输入值定义如下：



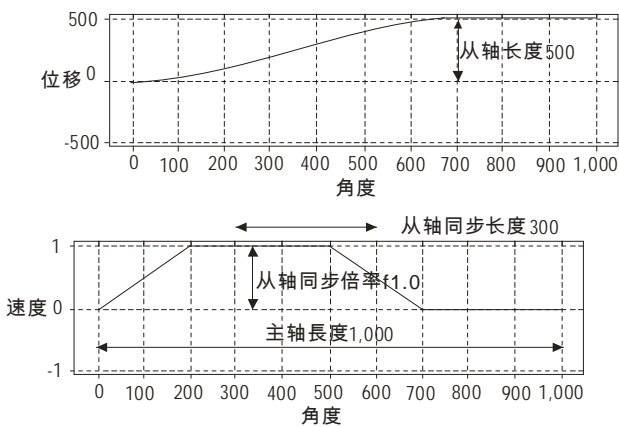
(*2) eCamCurve 输入值定义如下：



● 范例说明

【功能说明】

本范例说明使用飞剪曲线自动生成功能块建立飞剪曲线的步骤，其中飞剪设定参数为主轴长度=1000、从轴长度=500、同步长度=300、同步比率=1.0；。下图为此范例生成之飞剪曲线与飞剪参数设定值说明：

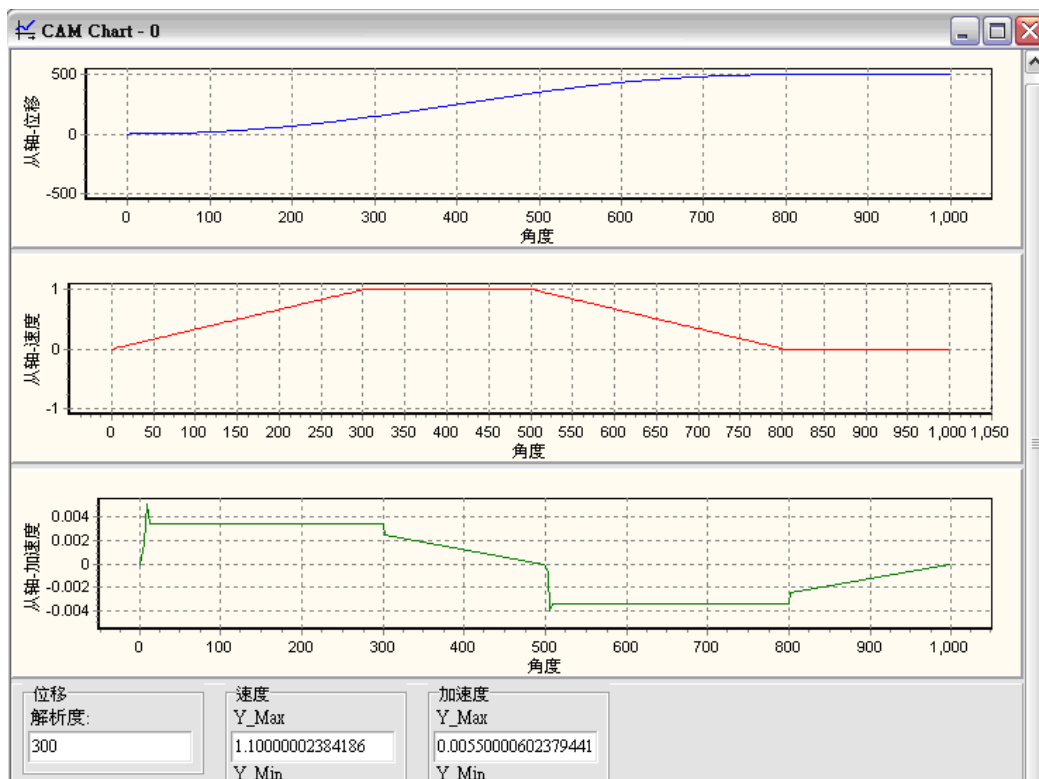
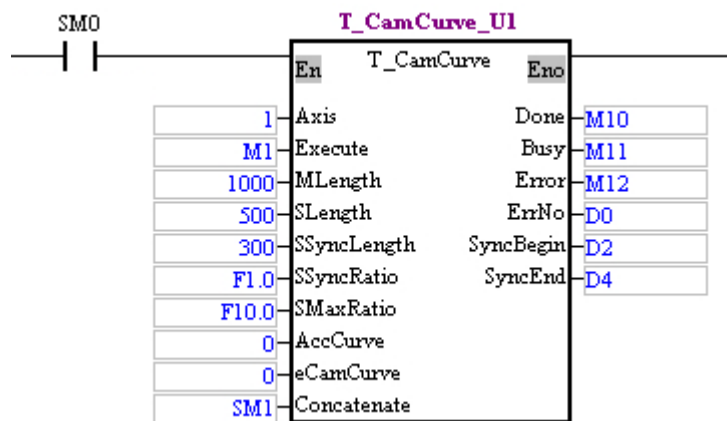


参数	设定值
主轴长度	1000
从轴长度	500
同步长度	300
同步比率	1.0
同步比率限制	10.0
加速曲线	0
CAM 曲线	0
生成设定	仅允许单笔动态变更
生成结果	0

【操作步骤】

- 在 PMSoft 中开启一组 CAM Chart，解析度设定为 300。
- 将程序下载至 AH500 运动控制模块后执行
- Set M1 启动运动功能块飞剪曲线生成
- 停止执行 AH500 运动模块，将程序上载。
- 检视上载后的 CAM Chart0 曲线，为自动生成的飞剪曲线。

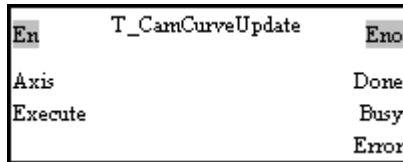
【PM 程序】



8

8.4.3 功能块-凸轮动态变更操作

功能块凸轮动态变更为搭配凸轮飞剪曲线生成之用。目的为当凸轮飞剪需要修改时，利用凸轮飞剪曲线生成来产出新的飞剪曲线，而后利用凸轮动态变更来更新飞剪曲线数据。

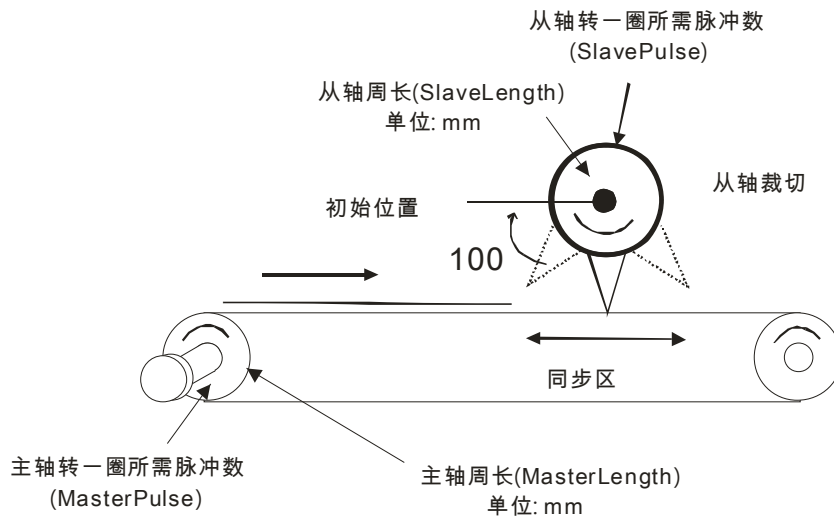


输入引脚				
引脚名称	功能说明	数据格式	设定值范围	说明
Axis	运动轴编号	WORD	1~16	电子凸轮从轴编号
Execute	上升沿时启动更新	BOOL	TRUE / FALSE	启动动态变更

输出状态引脚				
引脚名称	功能说明	数据格式	输出值范围	说明
Done	凸轮曲线更新完成	BOOL	TRUE / FALSE	完成动态变更
Busy	执行中	BOOL	TRUE / FALSE	动态变更中
Error	功能块产生错误	BOOL	TRUE / FALSE	错误发生

8.4.4 飞剪电子凸轮范例

【范例】



基本飞剪应用架构，利用第 1 轴为电子凸轮从轴，第 2 轴为电子凸轮主轴

【控制需求】

1. 使用飞剪曲线自动生成建立凸轮曲线。
2. 裁切轴与送料轴所搭配的设备，其伺服=10,000 pulse/rev。
3. 相关参数
 - ◆ 裁切料长度为 500 mm
 - ◆ 裁切轴周长为 60π mm
 - ◆ 送料轴周长为 100π mm
 - ◆ 送料轴速度为 1,000 Hz
4. 运动轴规划
 - 第 1 轴：电子凸轮从轴
 - 第 2 轴：电子凸轮主轴
5. 功能块使用列表

功能块名称	功能块外观	功能说明
凸轮同步速率计算		用于飞剪曲线生成时协助计算输入脚 SyncRatio 数值
凸轮飞剪曲线生成		自动生成飞剪凸轮曲线
电子凸轮运动		启动/停止电子凸轮

【组件说明】

PLC 装置		说 明
PMSoft 软件接点	M0	计算同步比率比率
	M10	生成飞剪曲线
	M50	飞剪曲线生成标志
	M70	启动/关闭电子凸轮功能
	M72	凸轮脱离开关
	M200	进入凸轮模式

【控制程序】

1. 利用飞剪曲线自动生成建立电子凸轮曲线

- 需输入主轴裁切料长度，由于裁切料长度为 500mm $\xrightarrow{\text{转换}} 1,000 \times \frac{1,000}{100\pi} = 3,183$

pulse。

- 从轴周长即从轴转一圈需多少脉冲数，此范例为 10,000 pulse
- 从轴同步长度设定约为从轴周长的百分之 30 为 $10,000 \times 30\% = 3,000$ pulse。
- 同步比率为计算
 - ◆ 主轴一周长度 314mm
 - ◆ 主轴一周脉冲数 10,000 pulse
 - ◆ 从轴一周长度 188mm
 - ◆ 从轴一周脉冲数 10,000 pulse

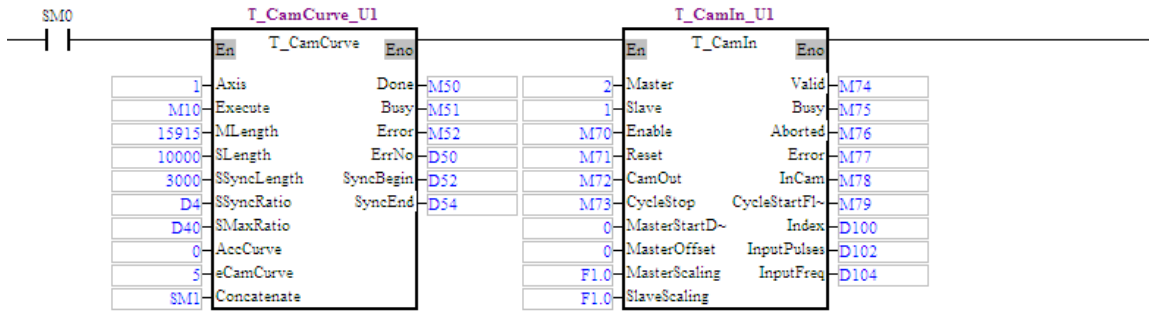
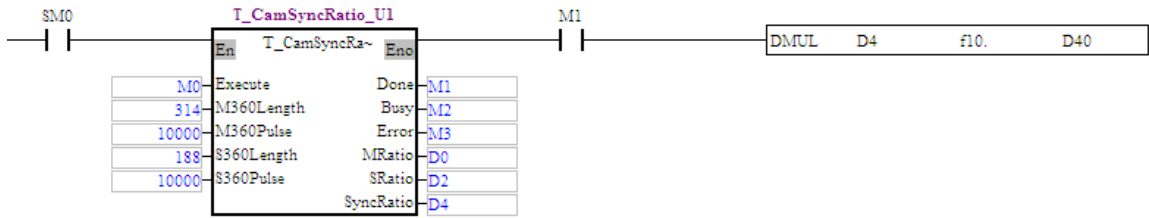
可由 CamSyncRatio 的输出引脚 SyncRatio 得到同步比率比率。

- 最高比率限制为：设定为同步比率的 10 倍。
- 速度曲线选择为：Constant Speed
- 飞剪曲线选择为：曲线为 RightCAM
- 选择不接续生成

依据上述的参数设定透过飞剪曲线功能块，产生出 CAM 表。



2. PMSoft 主程序

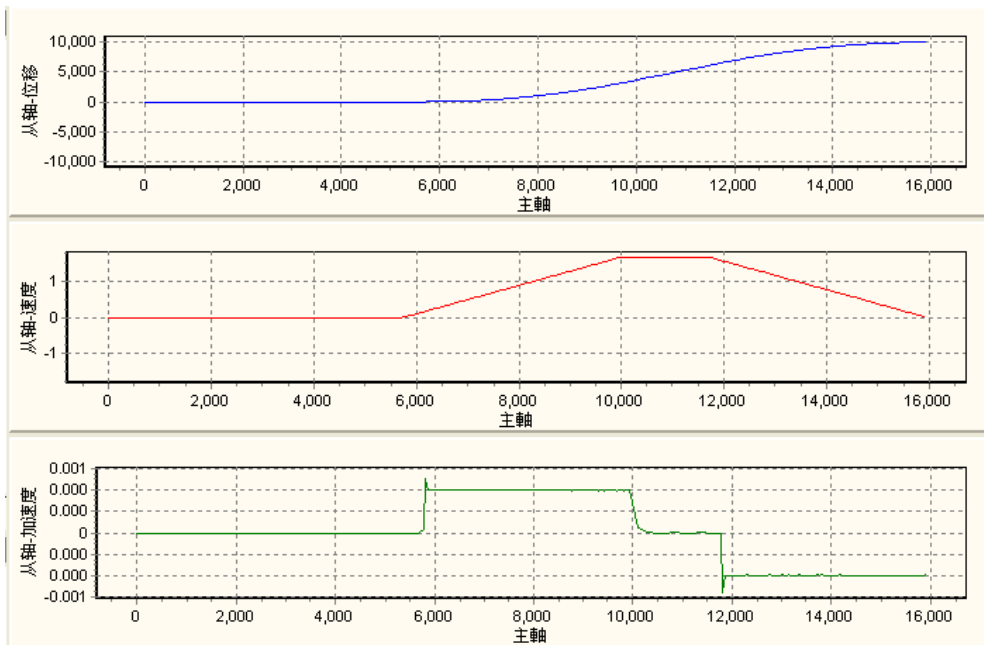


【操作步骤】

1. 新增 1 个 CAM Chart，解析度设定为 300。
2. 将程序下载至 AH500 运动模块中当中执行
3. Set M0 计算同步比率
4. Set M10 后将建立出凸轮曲线表、可得到设定同步区的上下限分别为 9,927 和 11,723。

5. 将模块内的 CAM 表数据读回，如下图所示，查看所建立的 CAM chart 是否符合。

- 由下图如得知主轴长度与所设定的长度 15,915 pulse 相同；
- 于速度图可得知主从轴之间的比率与 CamSyncRatio 计算出的 SyncRatio 相同；
- 同步区所设定的大小为 3,000pulse 。
- 由速度图发现同步区由 9,927~11,723，其区间为 1,796 代表主轴的同步区长度，而非从轴同步区长度。
- 从轴同步区长度=主轴同步区范围 x 同步比率，即 $1,796 \times 1.6702 = 3000$ 。



- Set M70 启动电子凸轮模式
- Set M72 将会脱离电子凸轮模式

9

第9章 多轴插补运动

目录

9.1	多轴插补功能简介.....	9-2
9.2	O 指针/M 码指令及 G 码指令一览表.....	9-2
9.3	G 码指令的组成.....	9-3
9.4	G 码指令说明.....	9-6
9.5	O 指针/M 码指令.....	9-23
9.6	TO 指令使用说明.....	9-26

9.1 多轴插补功能简介

AH500 运动定位模块支持多轴插补。而在插补的操作使用上可分为两类：

- ◆ 运动子程序 OX0~OX99 中写入 G 码来实现简易 CNC 加工运动
- ◆ 利用 To 指令在程序中下达实现插补命令

9.2 O指针/M码指令及G码指令一览表

指令码	功能	说明	适用機種			
			20MC	10PM	15PM	05PM
O	程序指针	主程序 O100 · 运动子程序 OX0~OX99	✓	✓	✓	✓
M	M 码指令	M0~M65535 M102 : O100 主程序结束指令 M2 : OX0~OX99 运动子程序结束指令	✓	✓	✓	✓

📖 G 码指令

G 码	功能	适用機種			
		20MC	10PM	15PM	05PM
00	三轴高速定位	✓	✓	✓	✓
01	双轴同动直线插补移动	✓	✓	✓	✓
01	三轴同动直线插补移动	✓	✓	✓	✓
02	顺时针圆弧移动 (设定圆心位置)	✓	✓	✓	✓
02	顺时针螺旋移动 (设定圆心位置)	✓	✓	✓	✓
02	顺时针圆弧移动 (设定半径长度)	✓	✓	✓	✓
02	顺时针螺旋移动 (设定半径长度)	✓	✓	✓	✓
03	逆时针圆弧移动 (设定圆心位置)	✓	✓	✓	✓
03	逆时针螺旋移动 (设定圆心位置)	✓	✓	✓	✓
03	逆时针圆弧移动 (设定半径长度)	✓	✓	✓	✓
03	逆时针螺旋移动 (设定半径长度)	✓	✓	✓	✓
04	延迟时间	✓	✓	✓	✓
17	XY 平面设定	✓	✓	✓	✓
18	XZ 平面设定	✓	✓	✓	✓
19	YZ 平面设定	✓	✓	✓	✓
90	绝对坐标系统设定	✓	✓	✓	✓
91	相对坐标系统设定	✓	✓	✓	✓

补充说明：05PM=AH05PM-5A · 10PM=AH10PM-5A · 15PM=AH15PM-5A · 20MC=AH20MC-5A

9.3 G 码指令的组成

◆ G 码指令的结构可分为两部份：指令名及操作数

	指令名	表示指令执行功能
操作数	菜单示	参数标记 (X · Y · Z · I · J · K · R · F)
	参数设定	设定参数数值大小

- 操作数菜单示之参数标记必须填写。
- 操作数参数常数可输入型式：阿拉伯数字整数，都以 32 位计算。
例 1：G00 X100 Y100
例 2：G00 X100.0 Y100.0。
- 操作数若为浮点常数，则为常数为 1,000 倍，但浮点数只支持小数点以下 3 位。
例如：G00 X100.123 Y100.45678 位移等于 G00 X100123 Y100456。
- AH500 运动定位模块操作数参数除上述形式设定，还可输入 16 位寄存器 (D · W) 与 32 位寄存器 (DD · WW)。例如下面所示使用 D 寄存器使用说明：
G0 XD11 YDD20 ZWW25；
G01 XDD30 YD40 ZW10 F400；
G1 X100.0 Y25.0 FD50；
G02 XD60 Y50.0 ID100 JDD80；
G03 YDD90 RD70 F300.0；

◆ 计算 G 码指令所占程序空间

- G00、G01、G02、和 G03 等指令名占程序空间 2 个 Step，其余占 1 个 Step。
- 操作数参数，输入阿拉伯数字之操作数，1 个操作数占 3 个 Step，若为 D 或 W 装置占 1 个 Step，而 DD 和 WW 装置都占 2 个 Step。

◆ G 码指令的格式说明

①	②	③					
④	G 码	指令码	运算元				功能
	0000	G00	X (P ₁)	Y (P ₂)	Z (P ₃)	高速定位	
			A (P ₄)	B (P ₅)	C (P ₆)		
	装置	K	16#	F	D	DD	WW
	P ₁	●	●		●	●	●
	P ₂	●	●		●	●	●
	P ₃	●	●		●	●	●
	P ₄	●	●		●	●	●
	P ₅	●	●		●	●	●
	P ₆	●	●		●	●	●

- ① G 码指令编号号码
- ② G 码操作数参数标记
- ③ G 码操作数参数数值大小
- ④ G 码操作数参数可使用的装置。

◆ G 码指令的输入

G 码中有些指令仅有指令部份 (指令名) 构成，例如：G90、G91。但是大部份都是指令部份再加上好几个操作数所组合而成。G 码指令前不须条件接点。

◆ G 码使用说明

- 同一行程序可摆放多种功能
例如：G91G01 X100.0 Y300.0 F500.0 M8 G04 X4.5 ;
- 同一群族摆放在同一行程序以最后一个指令为主
例如：
G02 G00 G03 G01 X100.0 Y300.0 F500.0 ;=>G01 X100.0 Y300.0 F500.0 ;
G02 G00 X100.0 G03 G01 Y300.0 F500.0 ;=>G01 Y300.0 F50 0.0 ;
- 高速定位指令 (G00) 不需设定移动速度
例如：G00 X100.2 Y50 0.0 ;
其中速度为控制器内部参数设定中的最高移动速度 (V_{MAX}) 移动。
- 高速定位 (G00) 与直线插补 (G01) 指令有延续功能。
N0000 G00 X500.0 Y125.0 ;
N0001 X-40 0.0 Y-500.0 ; =>G00 X-4 00.0 Y-500.0 ;
N0002 G01 X100.0 Y25.0 F200.0 ;
N0003 X-20 0.0 Y50.0 ; =>G01 X-2 00.0 Y50.0 F200.0 ;

9

- G01 G02 G03 速度参数 F 有延续功能。(第一行指令必需指定参数 F)


```
N0000 G01 X 500.0 Y 125.0 F 200.0 ;
N0001 G03 X -40.0 Y -50.0 R 100.0 ;           =>G03 X-40.0 Y-50.0 R 100.0
                                                F200.0 ;
N0002 G02 X 100.0 Y 25.0 I 400.5 F 200.0 ;
N0003 G01 X -200.0 Y 50.0 ;                   =>G01 X-200.0 Y 50.0 F 200.0 ;
```
- G90 (绝对坐标) 与 G91 (相对坐标) 有最高优先权


```
G90 G 01 X100.0 Y 300.0 F 500.0 ;           =>G90 G 01 X 100.0 Y 300.0 F 500.0 ;
G01G90 X 100.0 Y 300.0 F 500.0 ;           =>G90 G 01 X 100.0 Y 300.0 F 500.0 ;
```
- 程序代码有空格或没空格都能辨识


```
G01G91X500.0 Y 125.0F200.0 ;               =>G01 G 91 X 500.0 Y 125.0 F 200.0 ;
```
- 坐标值与速度值转码时都变为 32 位


```
G01 X -125.5 F 200.0 ;                      =>G01 X -125500 F 200000 ;
```
- 坐标值与速度值转码时有小数点 (.) 都乘以 1000 倍。


```
G01 X 100 Y -125.5 F 200.0 ;               =>G01 X 100 Y -125500 F 200000 ;
```
- G04 参数 X : 设定延迟单位为 1 秒 ; 参数 P : 设定延迟单位为 1 毫秒) · 如下所示 P2509 中的 9 · 系统将自动忽略 :


```
G04 X 4.5 (延迟 4.5 秒)
G04 X 5 (延迟 5 秒)
G04 P 4500 (延迟 4.5 秒)
G04 P 2509 (延迟 2.5 秒)
```
- 对不支持的 G 码都能忽略不读取。


```
G21G54G01 X -125.5 F 200.0 ;               =>G01 X -125500 F 200000 ;
G43G87G96 X 250.5 F 200.0 ;               =>G01 X 250500 F 200000 ;
```
- 编写指令方式为更符合一般 G 码 · 操作数参数标记的顺序可不用按照 X、Y、Z、I、J、K、R 等顺序编写 · 可任意编写顺序 · 如下所示都是合法 :


```
G0 X4.5 Z40.0 Y30.5 F200.2 ;               =>G00 X4.500 Y30.500 Z40.000 F200.200
      Z100.5Y400.0X300.0 ;                   =>G00 X300.000 Y400.000 Z100.500
G1xd100zd300y200.45 fd400                 =>G01 XD100 Y200.450 ZD300 FD400
G3 ZD100 I200.0F50.60XD300 m80           =>G03 XD300 ZD100 I200.000 F50.600
      M80
G03 yD100 x9999.9Z200.0r777.7 Fd800     =>G03 X9999.900 YD100 Z200.000
      R777.700 FD800
```

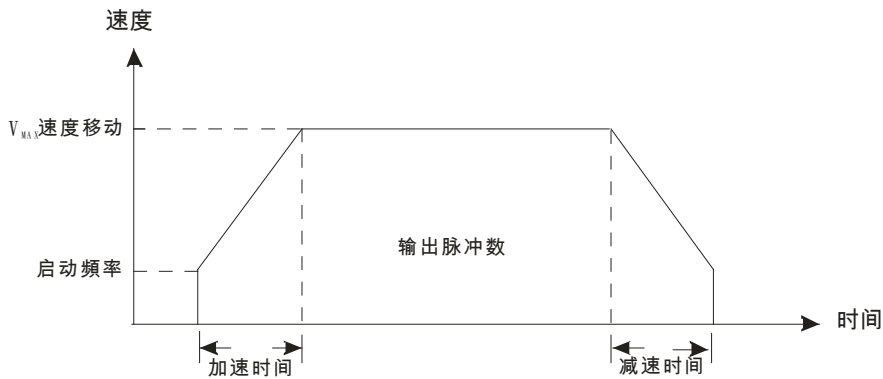
9.4 G码指令说明

G 码	指令码		操作数				功能	
0000		G00	X (P ₁)	Y (P ₂)	Z (P ₃)	A (P ₄)	高速定位	
			B (P ₅)	C (P ₆)				

装置	K	16#	F	D	DD	W	WW
P ₁	●	●		●	●	●	●
P ₂	●	●		●	●	●	●
P ₃	●	●		●	●	●	●
P ₄	●	●		●	●	●	●
P ₅	●	●		●	●	●	●
P ₆	●	●		●	●	●	●

指令说明：

- P₁：X 轴目标位置。P₂：Y 轴目标位置。P₃：Z 轴目标位置。
P₄：A 轴目标位置。P₅：B 轴目标位置。P₆：C 轴目标位置。
- 无输入小数点，参数设定范围：(-2,147,483,648~2,147,483,647)；有输入小数点，参数设定范围：(-2,147,483.648~2,147,483.647)。
- 此指令编写不需设定移动速度，每轴移动速度为各轴设定最高速度移动。
- 编写指令只要有一轴以上（含一轴）目标位置即可。
- 各轴运作图如下：

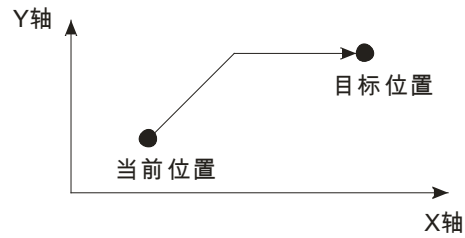


- 加减速时间与启动速度可由 SR 寄存器来设定。
- 加减速时间会依最高速度设定等比例增加或减少。

- 程序范例：**G00 X250.0 Y150**

此指令以当前位置 (50.0 · 50.0) 两轴快速直线移动到目标位置 (250.0 · 150.0) 。其目标位置有可能是绝对坐标或是相对坐标，这由之前最靠近这指令的定义坐标系指令 (G90 、 G91) 来决定，而 X 、 Y 轴移动速度以最高速度 V_{MAX} 运转速度输出。

移动路径：

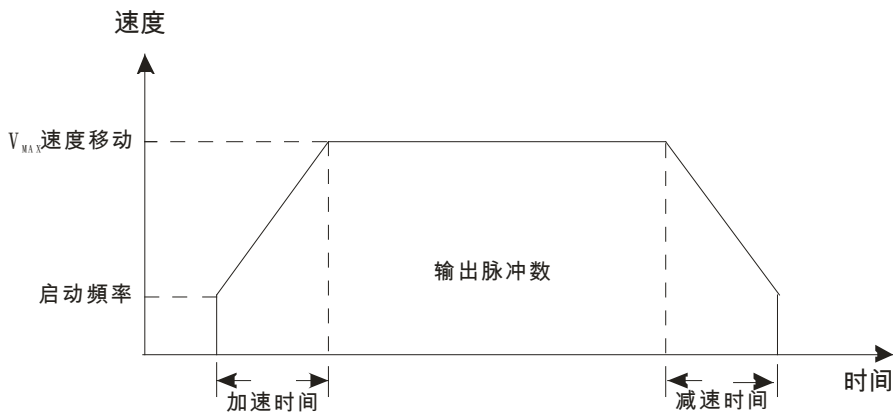


G 码	指令码		操作数			功能	
0001		G01	X (P ₁)	Y (P ₂)	Z (P ₃)	A (P ₄)	直线插补移动 (可考虑剩余距离)
			B (P ₅)	C (P ₆)	F (V)		

装置	K	16#	F	D	DD	W	WW
P ₁	●	●		●	●	●	●
P ₂	●	●		●	●	●	●
P ₃	●	●		●	●	●	●
P ₄	●	●		●	●	●	●
P ₅	●	●		●	●	●	●
P ₆	●	●		●	●	●	●
V	●	●		●	●	●	●

指令说明：

- P₁：X 轴目标位置。P₂：Y 轴目标位置。P₃：Z 轴目标位置。
P₄：A 轴目标位置。P₅：B 轴目标位置。P₆：C 轴目标位置。V：直线插补运转速度。
- 无输入小数点，P₁、P₂、P₃ 参数设定范围：(-2,147,483,648~2,147,483,647)；V 参数设定范围：(0~500,000)；有输入小数点，P₁、P₂、P₃ 参数设定范围：(-2,147,483.648~2,147,483.647)；V 参数设定范围：(0~500.0)。
- 此指令若有编写移动速度 V，则直线插补运转以 V 速度移动，若没有编写移动速度，则以之前 G01、G02 或 G03 指令设定速度移动。
 - V 运转速度设定最高 V_{MAX}
 - 编写指令只要有一轴以上（含一轴）目标位置即可。
 - 各轴运作图如下：

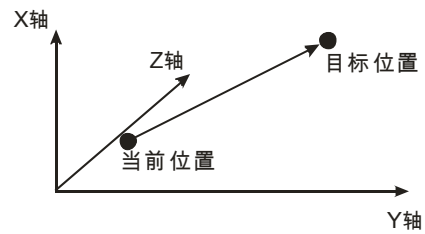


- 加减速时间与启动速度可由 SR 寄存器来设定。
- 加减速时间会依最高速度设定等比例增加或减少。

- 程序范例：**G01 X200.0 Y400.0 Z250.0 F400.0**

此指令以当前位置 (0 · 10.0 · 100.0) 三轴快速直线移动到目标位置 (200.0 · 400.0 · 250.0) ，其目标位置有可能是绝对坐标或是相对坐标，这由之前最靠近这指令的定义坐标系指令 (G90 、 G91) 来决定，而移动速度为 **400kHz** 输出。

移动路径：

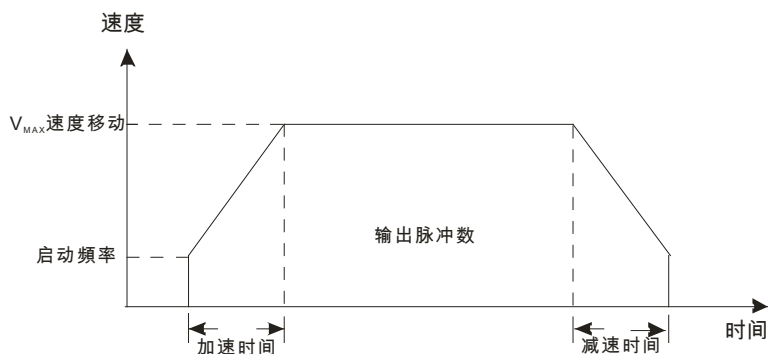


G 码	指令码		操作数	功能
0002 0003	G02 G03		X (P ₁) Y (P ₂) Z (P ₃) I (P ₄) J (P ₅) K (P ₆) F (V)	顺时针圆弧/螺旋移动 逆时针圆弧移动/螺旋移动 (设定圆心位置)

装置	K	16#	F	D	DD	W	WW
P ₁	●	●		●	●	●	●
P ₂	●	●		●	●	●	●
P ₃	●	●		●	●	●	●
P ₄	●	●		●	●	●	●
P ₅	●	●		●	●	●	●
P ₆	●	●		●	●	●	●
V	●	●		●	●	●	●

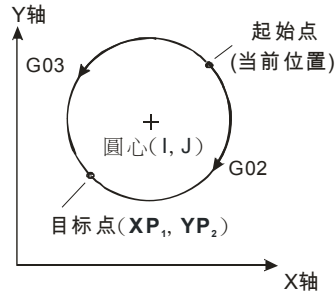
指令说明：

- P₁：X 轴目标位置。P₂：Y 轴目标位置。P₃：Z 轴目标位置。P₄：圆心相对于起始点的 X 轴距离。P₅：圆心相对于起始点的 Y 轴距离。P₆：圆心相对于起始点的 Z 轴距离。V：圆弧/螺旋插补运转速度。
- P₄、P₅ 和 P₆：以圆弧或螺旋圆心为基准点，相对于当前位置向量值，也就是圆心坐标减掉当前位置（圆弧起点）。
- 无输入小数点，P₁、P₂、P₃、P₄、P₅、P₆ 参数设定范围：(-2,147,483,648~2,147,483,647)，V 参数设定范围：(0~500,000)；有输入小数点，P₁、P₂、P₃、P₄、P₅、P₆ 参数设定范围：(-2,147,483.648~2,147,483.647)，V 参数设定范围：(0~500.0)。
- 此指令若有编写移动速度 V，则圆弧或螺旋插补运转以 V 速度移动，若没有编写移动速度，则以之前 G01、G02 或 G03 指令设定速度移动。
 - V 运转速度设定最高 VMAX
 - 各轴动作图：



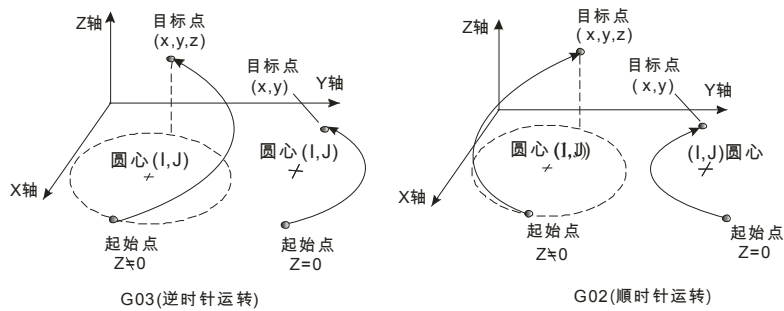
- 加减速时间与启动速度可由 SR 寄存器来设定。
- 加减速时间会依最高速度设定等比例增加或减少。

- 双轴圆弧插补运转：用于任两垂直轴圆弧插补运转，使用 G 码指令中的 G17 (XY 平面设定)、G18 (XZ 平面设定) 或 G19 (YZ 平面设定)，控制任两轴圆弧插补。下图所示为 XY 两轴圆弧插补。

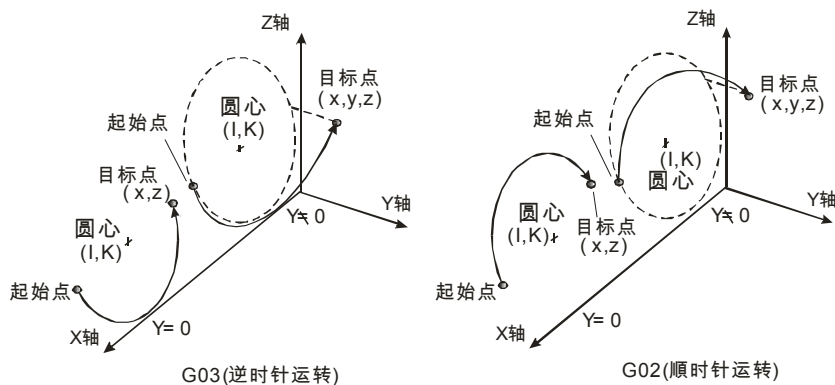


- 螺旋插补运转：此功能为两轴圆弧插补，再增加一轴垂直两轴圆弧平面的高度移动，而这三轴是同动运转，其为圆弧插补功能的延伸。当执行螺旋插补指令而高度值变化是零，就是圆弧功能。

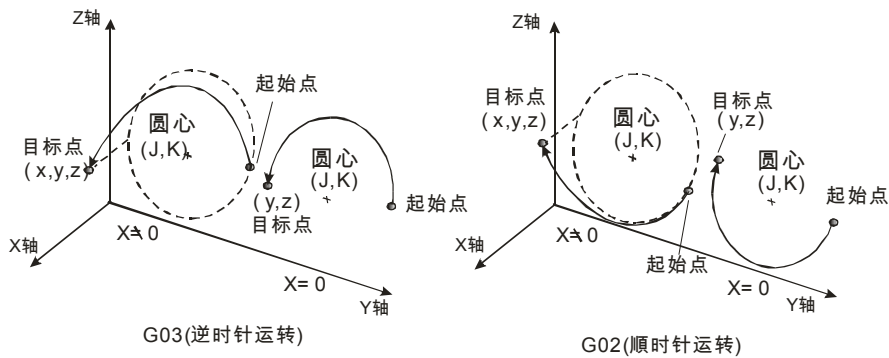
- 下图为螺旋插补在 G17 模式 (XY 平面) 下，假如 Z 轴无变化量 ($Z=0$)，螺旋插补运作同 XY 轴两轴圆弧插补。



- 下图为螺旋插补在 G18 模式 (XZ 平面) 下，假如 Y 轴无变化量 ($Y=0$)，螺旋插补运作同 XZ 轴两轴圆弧插补。



- 下图为螺旋插补在 G19 模式 (YZ 平面) 下，假如 X 轴无变化量 (X=0)，螺旋插补运作同 YZ 轴两轴圆弧插补。



- 编写指令原则：(1) 一定要有目标位置，及圆心位置，但不一定要设移动速度。(2) 目标位置任一轴与目前位置对应轴无变化量，可省略。(3) 圆心位置两轴任一轴与目前位置对应轴无变化量，可省略。按上述三种原则，编写圆弧/螺旋指令可有多种组合。

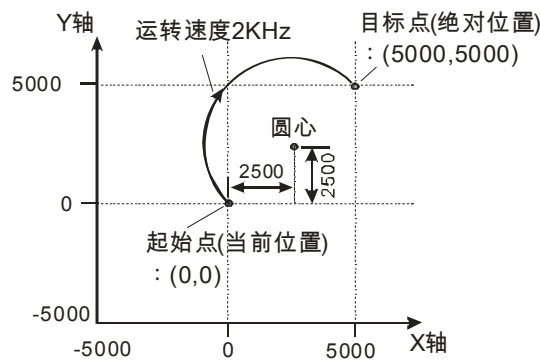
NO.	G 码	操作数参数组合	G17	G18	G19
1	G02/G03	X(P ₁) I(P ₃)	✓	✓	
2		X(P ₁) I(P ₃) F(V)	✓	✓	
3		X(P ₁) J(P ₄)	✓		
4		X(P ₁) J(P ₄) F(V)	✓		
5		X(P ₁) I(P ₃) J(P ₄)	✓		
6		X(P ₁) I(P ₃) J(P ₄) F(V)	✓		
7		Y(P ₂) I(P ₃)	✓		
8		Y(P ₂) I(P ₃) F(V)	✓		
9		Y(P ₂) J(P ₄)	✓		✓
10		Y(P ₂) J(P ₄) F(V)	✓		✓
11		Y(P ₂) I(P ₃) J(P ₄)	✓		
12		Y(P ₂) I(P ₃) J(P ₄) F(V)	✓		
13		X(P ₁) Y(P ₂) I(P ₃)	✓	✓	
14		X(P ₁) Y(P ₂) I(P ₃) F(V)	✓	✓	
15		X(P ₁) Y(P ₂) J(P ₄)	✓		✓
16		X(P ₁) Y(P ₂) J(P ₄) F(V)	✓		✓
17		X(P ₁) Y(P ₂) I(P ₃) J(P ₄)	✓		
18		X(P ₁) Y(P ₂) I(P ₃) J(P ₄) F(V)	✓		
19		X(P ₁) Z(P ₃) I(P ₃)	✓	✓	
20		X(P ₁) Z(P ₃) I(P ₃) F(V)	✓	✓	
21		X(P ₁) Z(P ₃) J(P ₄)	✓		✓
22		X(P ₁) Z(P ₃) J(P ₄) F(V)	✓		✓
23		X(P ₁) Z(P ₃) I(P ₃) J(P ₄)	✓		
24		X(P ₁) Z(P ₃) I(P ₃) J(P ₄) F(V)	✓		

NO.	G 码	操作数参数组合	G17	G18	G19
25	G02/G03	Y (P ₂) Z (P ₃) I (P ₃)	✓	✓	
26		Y (P ₂) Z (P ₃) I (P ₃) F (V)	✓	✓	
27		Y (P ₂) Z (P ₃) J (P ₄)	✓		✓
28		Y (P ₂) Z (P ₃) J (P ₄) F (V)	✓		✓
29		Y (P ₂) Z (P ₃) I (P ₃) J (P ₄)	✓		
30		Y (P ₂) Z (P ₃) I (P ₃) J (P ₄) F (V)	✓		
31		X (P ₁) Y (P ₂) Z (P ₃) I (P ₃)	✓	✓	
32		X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) F (V)	✓	✓	
33		X (P ₁) Y (P ₂) Z (P ₃) J (P ₄)	✓		✓
34		X (P ₁) Y (P ₂) Z (P ₃) J (P ₄) F (V)	✓		✓
35		X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) J (P ₄)	✓		
36		X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) J (P ₄) F (V)	✓		
37		X (P ₁) K (P ₆)			✓
38		X (P ₁) K (P ₆) F (V)			✓
39		X (P ₁) I (P ₃) K (P ₆)			✓
40		X (P ₁) I (P ₃) K (P ₆) F (V)			✓
41		Z (P ₃) I (P ₃)			✓
42		Z (P ₃) I (P ₃) F (V)			✓
43		Z (P ₃) K (P ₆)			✓
44		Z (P ₃) K (P ₆) F (V)			✓
45		Z (P ₃) I (P ₃) K (P ₆)			✓
46		Z (P ₃) I (P ₃) K (P ₆) F (V)			✓
47		X (P ₁) Z (P ₃) K (P ₆)			✓
48		X (P ₁) Z (P ₃) K (P ₆) F (V)			✓
49		X (P ₁) Z (P ₃) I (P ₃) K (P ₆)			✓
50		X (P ₁) Z (P ₃) I (P ₃) K (P ₆) F (V)			✓
51		X (P ₁) Y (P ₂) K (P ₆)			✓
52		X (P ₁) Y (P ₂) K (P ₆) F (V)			✓
53		X (P ₁) Y (P ₂) I (P ₃) K (P ₆)			✓
54		X (P ₁) Y (P ₂) I (P ₃) K (P ₆) F (V)			✓
55		Y (P ₂) Z (P ₃) K (P ₆)			✓
56		Y (P ₂) Z (P ₃) K (P ₆) F (V)			✓
57	Y (P ₂) Z (P ₃) I (P ₃) K (P ₆)			✓	
58	Y (P ₂) Z (P ₃) I (P ₃) K (P ₆) F (V)			✓	
59	X (P ₁) Y (P ₂) Z (P ₃) K (P ₆)			✓	
60	X (P ₁) Y (P ₂) Z (P ₃) K (P ₆) F (V)			✓	
61	X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) K (P ₆)			✓	
62	X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) K (P ₆) F (V)			✓	

NO.	G 码	操作数参数组合	G17	G18	G19
63	G02/G03	Y (P ₂) K (P ₆)			✓
64		Y (P ₂) K (P ₆) F (V)			✓
65		Y (P ₂) J (P ₄) K (P ₆)			✓
66		Y (P ₂) J (P ₄) K (P ₆) F (V)			✓
67		Z (P ₃) J (P ₄)			✓
68		Z (P ₃) J (P ₄) F (V)			✓
69		Z (P ₃) J (P ₄) K (P ₆)			✓
70		Z (P ₃) J (P ₄) K (P ₆) F (V)			✓
71		Y (P ₂) Z (P ₃) J (P ₄) K (P ₆)			✓
72		Y (P ₂) Z (P ₃) J (P ₄) K (P ₆) F (V)			✓
73		X (P ₁) Y (P ₂) J (P ₄) K (P ₆)			✓
74		X (P ₁) Y (P ₂) J (P ₄) K (P ₆) F (V)			✓
75		X (P ₁) Z (P ₃) J (P ₄) K (P ₆)			✓
76		X (P ₁) Z (P ₃) J (P ₄) K (P ₆) F (V)			✓
77		X (P ₁) Y (P ₂) Z (P ₃) J (P ₄) K (P ₆)			✓
78	X (P ₁) Y (P ₂) Z (P ₃) J (P ₄) K (P ₆) F (V)			✓	

- 圆弧移动路径或螺旋路径垂直投影轨迹可以全圆 360 度。
- 程序范例 1：

程序一开始为设定绝对坐标，CW 为顺时针圆弧指令，圆弧起点位置为 (0·0)，圆弧终点位置为 (5000·5000)，圆心在相对于圆弧起点位置 (2500·2500)，而 2000Hz 输出速度移动。

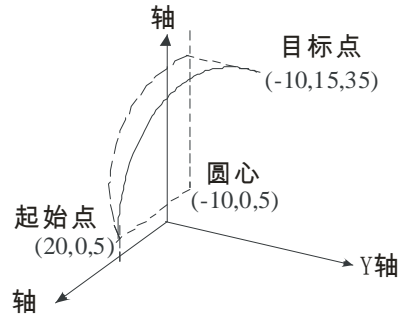


程序写法如下：

```
G90 ;
G02 X50.0 Y50.0 I2500 J2.5 F2000 ;
```

● 程序范例 2 :

程序一开始为设定绝对坐标。在 XZ 平面模式下，G02 为顺时针螺旋指令，螺旋终点位置为 $(-10, 15, 35)$ ，螺旋圆心为 $(-10, 0, 5)$ ，而 2000Hz 输出速度移动。



程序写法如下：

G90 ;

G18 ;

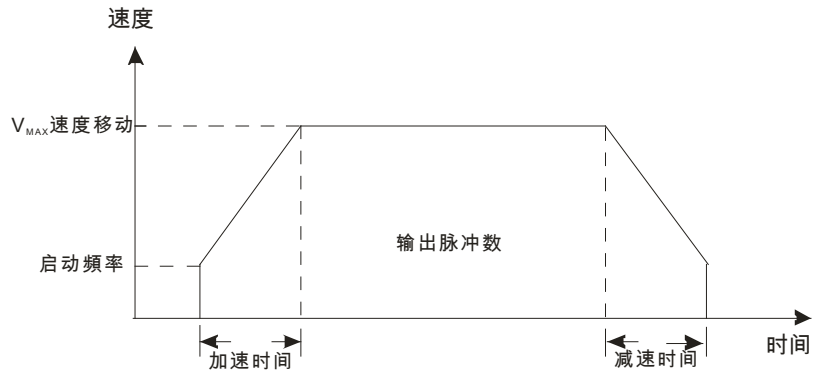
G02 X-10 Y15 Z35 I-30 J0 (可省略) K0 (可省略) F2000 ;

G 码	指令码		操作数	功能
0002 0003	G02 G03		X (P ₁) Y (P ₂) Z (P ₃) R (L) F (V)	顺时针圆弧/螺旋移动 逆时针圆弧移动/螺旋移动 (设定半径长度)

装置	K	16#	F	D	DD	W	WW
P ₁	●	●		●	●	●	●
P ₂	●	●		●	●	●	●
P ₃	●	●		●	●	●	●
L	●	●		●	●	●	●
V	●	●		●	●	●	●

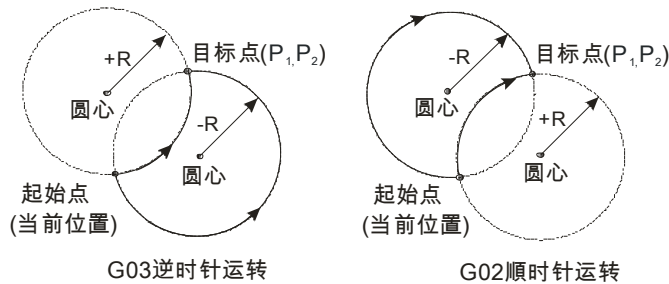
指令说明：

- P₁：圆弧 X 轴目标位置。P₂：圆弧 Y 轴目标位置。P₃：Z 轴目标位置。L：圆弧半径（弧角小于 180 度，R 为“+”。弧角等于大于 180 度，R 为“-”）。V：圆弧移动到目标位置速度。
- 无输入小数点，P₁、P₂、P₃、R 参数设定范围：(-2,147,483,648~2,147,483,647)；V 参数设定范围：(0~500,000)；有输入小数点，P₁、P₂、P₃、R 参数设定范围：(-2,147,483.648~2,147,483.647)；V 参数设定范围：(0~500.0)。
- 圆弧半径 L：弧角小于 180 度，R 为“+”。弧角等于大于 180 度，R 为“-”。
- 此指令若有编写移动速度 V，则圆弧或螺旋插补运转以 V 速度移动，若没有编写移动速度，则以前 G01、G02 或 G03 指令设定速度移动。
 - V 运转速度设定最高 VMAX
 - 各轴动作图：



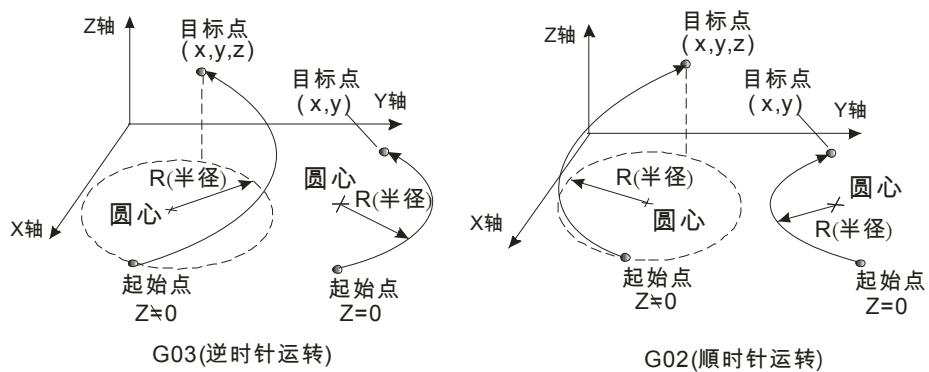
- 加减速时间与启动速度可由 SR 寄存器来设定。
- 加减速时间会依最高速度设定等比例增加或减少。

- 双轴圆弧插补运转：用于任两垂直轴圆弧插补运转，使用 G 码指令中的 G17 (XY 平面设定)、G18 (XZ 平面设定) 或 G19 (YZ 平面设定)，控制任两轴圆弧插补。下图所示为 XY 两轴圆弧插补。

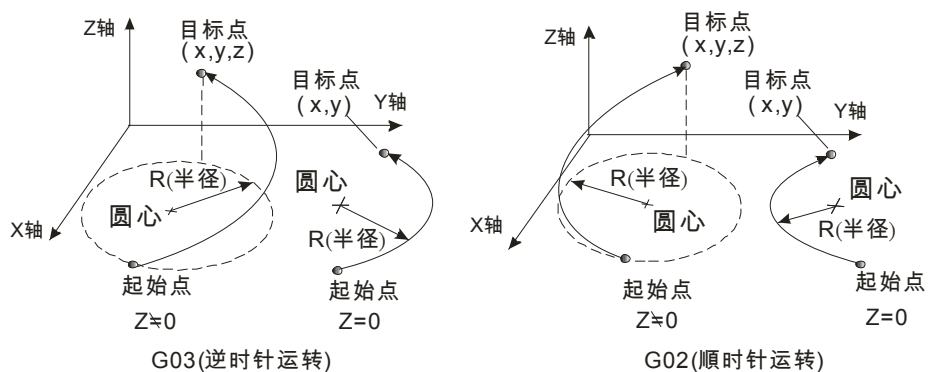


- 螺旋插补运转：此功能为两轴圆弧插补，再增加一轴垂直两轴圆弧平面的高度移动，而这三轴是同动运转，其为圆弧插补功能的延伸。当执行螺旋插补指令而高度值变化是零，就是圆弧功能。

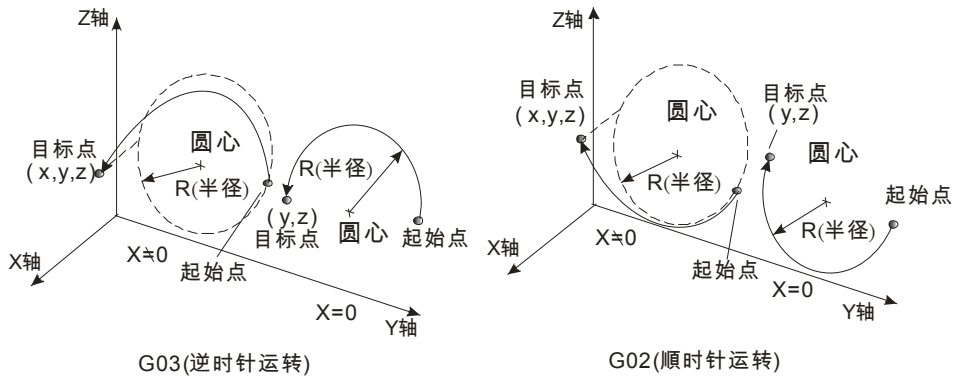
- 下图为螺旋插补在 G17 模式 (XY 平面) 下，假如 Z 轴无变化量 ($Z=0$)，螺旋插补运作同 XY 轴两轴圆弧插补。



- 下图为螺旋插补在 G18 模式 (XZ 平面) 下，假如 Y 轴无变化量 ($Y=0$)，螺旋插补运作同 XZ 轴两轴圆弧插补。



- 下图为螺旋插补在 G19 模式 (YZ 平面) 下，假如 X 轴无变化量 (X=0)，螺旋插补运作同 YZ 轴两轴圆弧插补。



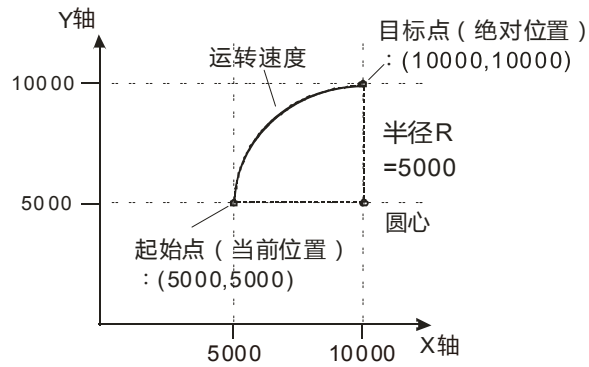
- 编写指令原则：(1) 一定要有目标位置，及圆心位置，但不一定要设移动速度。(2) 目标位置任一轴与目前位置对应轴无变化量，可省略。按上述两种原则，编写圆弧/螺旋指令可有如下表多种组合。

NO.	G 码	操作数参数组合	G17	G18	G19
1	G02/G03	X (P ₁) R (L)	✓	✓	
2		X (P ₁) R (L) F (V)	✓	✓	
3		Y (P ₂) R (L)	✓		✓
4		Y (P ₂) R (L) F (V)	✓		✓
5		X (P ₁) Y (P ₂) R (L)	✓	✓	✓
6		X (P ₁) Y (P ₂) R (L) F (V)	✓	✓	✓
7		X (P ₁) Z (P ₃) R (L)	✓	✓	✓
8		X (P ₁) Z (P ₃) R (L) F (V)	✓	✓	✓
9		Y (P ₂) Z (P ₃) R (L)	✓	✓	✓
10		Y (P ₂) Z (P ₃) R (L) F (V)	✓	✓	✓
11		X (P ₁) Z (P ₃) Y (P ₂) R (L)	✓	✓	✓
12		X (P ₁) Z (P ₃) Y (P ₂) R (L) F (V)	✓	✓	✓
13		Z (P ₃) R (L)		✓	✓
14		Z (P ₃) R (L) F (V)		✓	✓

- 圆弧移动路径或螺旋路径垂直投影轨迹无法全圆 360 度。

- 程序范例 1：

程序一开始设定绝对坐标，以 G02 顺时针圆弧指令，圆弧起点位置为 (5000 · 5000)，圆弧终点位置为 (10000 · 10000)，半径为 5000，而弧角小于 180 度，所以是正。而以每秒 1,000 个脉冲输出速度移动。



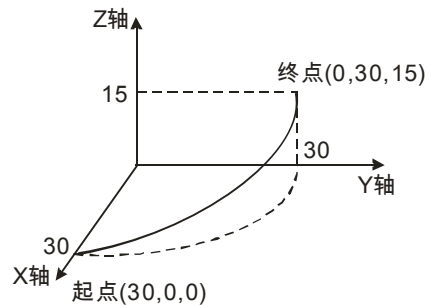
程序写法如下：

G90 ;

G02 X 10000 Y 10000 R 5000 F 1000 ;

- 程序范例 2：

程序一开始设定绝对坐标，以 G3 逆时针螺旋指令，螺旋起点位置为 (30 · 0 · 0)，螺旋终点位置为 (0 · 30 · 15)，半径为 30.0，而弧角小于 180 度，所以是正。而以每秒 1000 个脉冲输出速度移动。



程序写法如下：

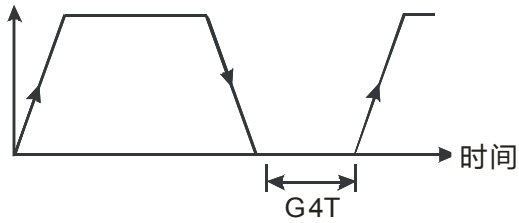
G90 ;

G3 X 10000 Y 10000 R 5000 F1000 ;

G 码	指令码		操作数				功能
0004		G04	X(T)/P(T)				延迟时间
装置	K	16#	F	D	DD	W	WW
T	●		●	●	●	●	●

指令说明：

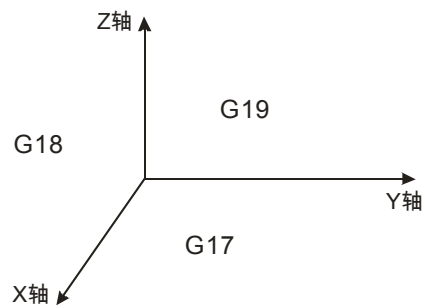
- T 装置设定暂停时间。
- 使用 X 操作数参数其单位为 1 秒，如 G4 X1 表示延迟 1 秒，G4 X2.5 表示延迟 2.5 秒。
- 使用 P 操作数参数其单位为 1 毫秒。G4 P100 表示延迟 0.1 秒，G4 P4500 表示延迟 4.5 秒，设定值以 10ms 为基数，若设定值小于 10ms 时，将以 0ms 为设定值，以此类推当设定值若为 23ms 时，则以 20ms 为设定值。
- 延迟时间是用来设定指令与指令间的等待时间。



G 码	指令码		操作数	功能
0017		G17	无操作数	选择 XY 平面
0018		G18	无操作数	选择 XZ 平面
0019		G19	无操作数	选择 YZ 平面

指令说明：

- 此三个指令动作内容，用于决定圆弧插补和螺旋插补作平面选择，对于直线插补无影响。
- 程序执行时，此三个工作平面可相互切换。程序中若无设定任何平面选择，系统初始状态为 XY 平面 (G17) 状态。
- 下图所示三种平面



G 码	指令码		操作数	功能
0090		G90	无操作数	绝对坐标系统设定
0091		G91	无操作数	相对坐标系统设定

指令说明：

- 执行指令 G90 绝对坐标：代表从 0 开始之位置，当目标位置大于当前位置时，电机正转，小于当前位置时，电机反转。
- 执行指令 G91 相对坐标：代表从当前位置开始算电机行走之距离，当相对坐标为正值时，电机正转，相对坐标为负值时，电机反转。
- 圆弧圆心坐标 (I · J · K)、半径 (R) 都增量距离，不受 G90 与 G91 影响。

9.5 O指针/M码指令

■ O 指标

下表所示为所有 O 指标，O100 指针为主程序，而 OX0~OX99 为一百组运动程序指针。

指令码	操作数	功能
O100	无操作数	主程序指针
OX0~OX99	无操作数	G 码运动程序指针

指令说明：

- O100 为一般主程序起始指针，而执行 M102 指令表示 O100 主程序结束。
- OX0~OX99 为 100 组 G 码运动子程序指针，可由使用者编写不同运动路径，使用 SR1052 寄存器 Low byte 储存 G 码运动子程序指针编号，但同时 SR1052 的 bit14 或 bit15 位要为 1，才能储存指标编号，最后以 SR1030 bit12=ON 控制启动指定的 G 码运动子程序，下面范例说明。

范例：要启动 OX98 运动子程序两个步骤：

(1) 设定编号：SR1052=16#8062 (或 16#4062、16#C062)。

(2) 启动 OX98：SR1030=16#1000。

- OX0~OX99 的运动子程序结束指令都是 M2。

程序范例：

行号 N0000~N0100 为 O100 主程序，N0102~N0304 为编号 OX50 运动子程序：

```

行号    程序
N0000  O 100
N0001  L D M1000
N0002  M OV 16#8063 D1868
N0007  M OV 16#1000 D1846
      :      :
N0099  O UT Y30
N0100  M 102
N0101  N OP
N0102  O X50
N0103  G 90 G00 X200.0 Y40.0
N0104  G 01 X500.0 F25.0
      :      :
N0304  M 2

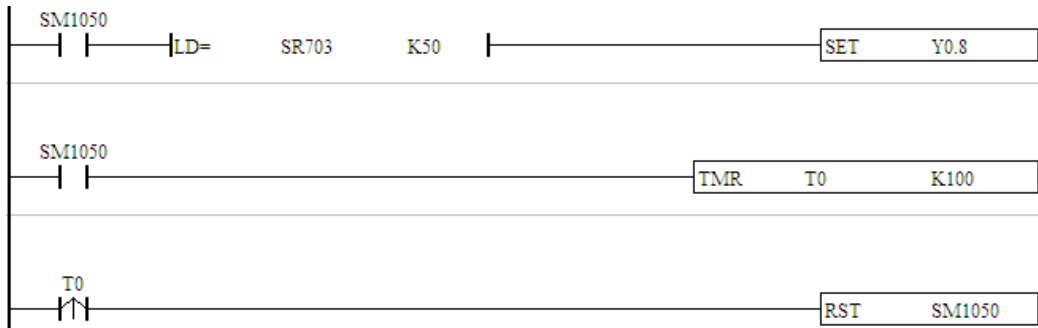
```


■ M 码

指令码	操作数	功能
M0~M65535	无操作数	M 码指令

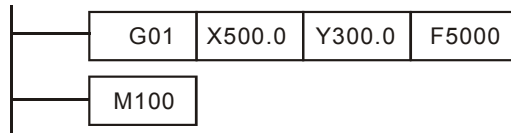
指令说明：

- M102 与 M02 指令分别为 O100 主程序和 OX0~OX99 运动程序结束，因此避免使用。
- M 码指令用于 Ox 运动子程序中，执行 M 码时，将 M 码编号存进 SR703，同时 SM1050 状态自动为 ON；若 SM1050 设为 OFF 状态，表示 M 码执行完毕。
- 执行 M 码时 SM1050 状态立即变为 ON 且 SR703 立即储存 M 码编号，可利用此特性在主程序作控制条件。如下范例说明：
 - 当执行到 M 码的 M50 时，立即设定 Y0.8 为 ON，过 1 秒之后，M50 执行结束，O100 主程序编辑程序如下：

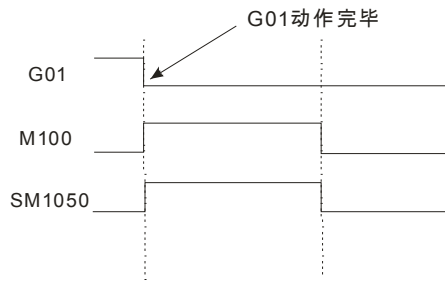


- M 码可分为两种模式：一行程序中只有 M 码指令，为 after 模式；若 M 码指令摆放于运动指令尾端，为 with 模式。如下范例说明两种模式的差异：

- after 模式：单独一行程序中，只有 M 码指令。如下所示：

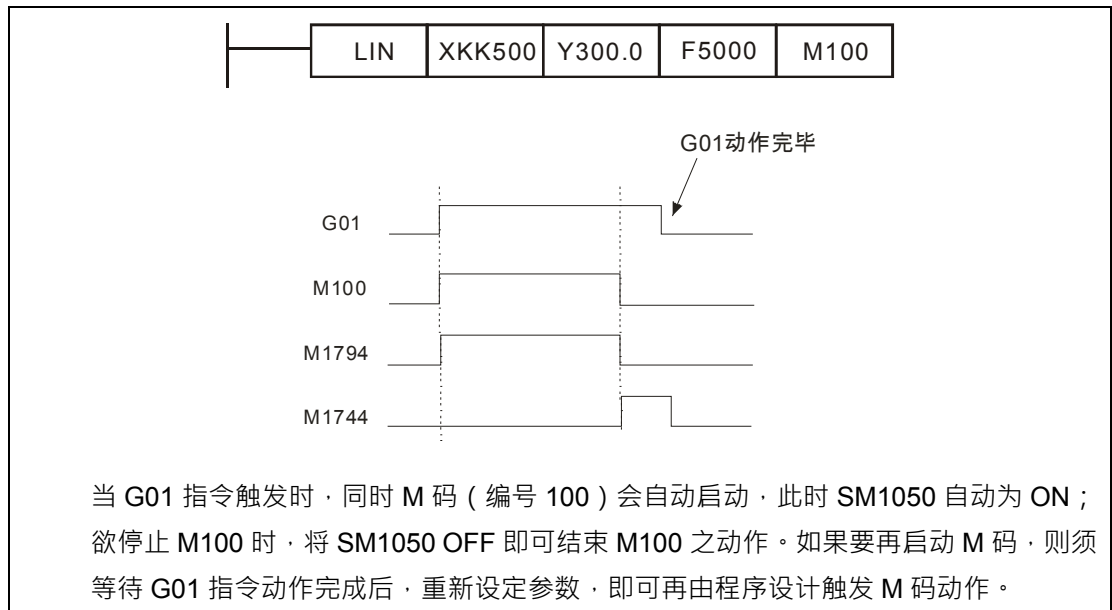


动作时序如下图：



当 G01 指令动作完毕后，M 码（编号 M100）将自动启动，此时 SM1050 也自动为 ON；欲停止 M100 时，将 SM1050 OFF 即可结束 M100 之动作。如果要再次启动 M 码，只须重新设定好，由程序中规划再次启动 M 码即可。

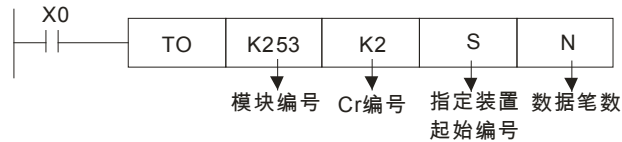
- with 模式：M 码指令摆放于运动指令尾端。如下所示：



9.6 TO指令使用说明

AH500 运动定位模块可透过 TO 指令设定各轴的直线插补功能开启以及关闭。以下说明如何使用 TO 指令来设定直线插补功能的状态。

■ CR#2 设定插补开启



- 各笔数据意义如下：

位置	设定控制
S · S ₊₁	插补轴选择
S ₊₂ · S ₊₃	插补速度
S ₊₄ · S ₊₅	轴 1 位置
S ₊₆ · S ₊₇	轴 2 位置
S ₊₈ · S ₊₉	轴 3 位置
:	:
S ₊₂₉ · S ₊₂₈	轴 15 位置
S ₊₃₁ · S ₊₃₀	轴 16 位置

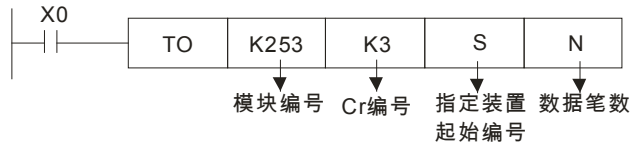
- 插补轴选择 (S · S₊₁) 定义如下：

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
轴号	轴 16	轴 15	轴 14	轴 13	轴 12	轴 11	轴 10	轴 9								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
轴号	轴 8	轴 7	轴 6	轴 5	轴 4	轴 3	轴 2	轴 1								

- 插补轴选择中，每一轴使用两个 bit 做为句柄，而句柄定义如下：

内容值	定义
0	不参与插补
1	直线插补开启
2	目前不使用
3	目前不使用

■ CR#3 设定插补停止



- 各笔数据意义如下：

位置	设定控制
S · S ₊₁	插补轴选择

- 插补轴选择 (S · S₊₁) 定义如下：

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
轴号	轴 16	轴 15	轴 14	轴 13	轴 12	轴 11	轴 10	轴 9	轴 8	轴 7	轴 6	轴 5	轴 4	轴 3	轴 2	轴 1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
轴号	轴 8	轴 7	轴 6	轴 5	轴 4	轴 3	轴 2	轴 1								

- 插补轴选择中，每一轴使用两个 bit 做为句柄，而句柄定义如下：

内容值	定义
0	不参与插补
1	直线插补停止
2	目前不使用
3	目前不使用

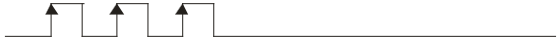




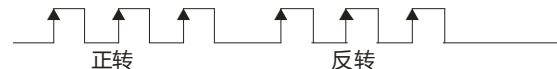
- 而各轴在运行直线插补时的相关设定参数可利用 SR1000 (SR1100 · SR1200...)，来做设定上的修改。相关参数设定定义如下：

特殊寄存器名称															
SR1000+100*N															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
'	加减速曲线选择 (注三)	DOG 触发启动模式	绝对/相对坐标设定	旋转方向	原点回归 DOG 下降沿检测	原点回归模式	原点回归方向	'	'	二)	脉冲输出型式 (正逻辑) (注	'			设定单位系 (注一)

注一：

	电机单位	复合单位	机械单位
位置	pulse		um
	pulse		m deg
	pulse		10-4inch
速度	pulse/sec		cm/min
	pulse/sec		10deg/min
	pulse/sec		inch/min

注二：

b5	b4	脉冲输出型式 (正逻辑)	说明
0	0	FP 正转脉冲  RP 反转脉冲 	双脉冲
0	1	FP 脉冲  RP 方向(DIR) 	单脉冲
1	0	FP A相脉冲 	A/B 相脉冲
1	1	RP B相脉冲 	4*A/B 相脉冲

注三：

bit#	说明
12	b[12]=0：设定为绝对坐标定位；b[12]=1：设定为相对坐标定位
14	b[14]=0：为加速度梯形加速曲线 b[14]=1：为加速度 S 型加速曲线

而各轴的直线插补动作是否完成可利用运动标志 SM1048 (SM1148、SM1248....) 来判断。

第10章 高速计数与计时功能

目录

10.1	高速计数器	10-2
10.2	高速定时器	10-6

AH500 运动定位模块配置高速计数器搭配虚拟高速计数器，由这高速计数器演化定时器应用，以下章节说明高速计数器与定时器功能。

10.1 高速计数器

1. 计数模式设定

下表为高速计数器使用设定：

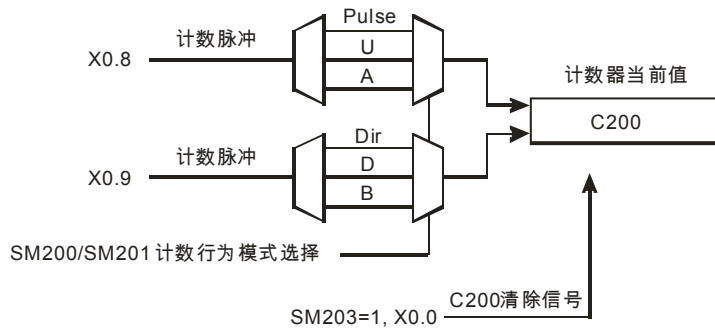
组别	计数装置	计数模式		清除信号设定	清除信号外部引脚	外部输入引脚 (*1) (*2)
		装置	设定值 (*3)			
0	C200	K1SM200	0 : U/D 1 : P/D 2 : A/B (1 倍频) 3 : 4A/B (4 倍频)	SM203	X0.0+、X0.0-	X0.8、X0.9、S/S
1	C204	K1SM204		SM207	X0.1+、X0.1-	X0.10、X0.11、S/S
2	C208	K1SM208		SM211	X0.2+、X0.2-	X0.12、X0.13、S/S
3	C212	K1SM212		SM215	X0.3+、X0.3-	X0.10、X0.11、S/S
4	C216	K1SM216		SM219	X0.2+、X0.2-	X0.12、X0.13、S/S
5	C220	K1SM220		SM223	X0.3+、X0.3-	X0.14、X0.15、S/S

*1. AH20MC-5A 机种输入端为差动线路，AH15PM-5A 机种之 X0.8 及 X0.9 输入端为差动线路，AH05PM/10PM-5A 集电极机种输入端为集电极线路，AH15PM-5A 机种之 X0.10~X0.15 为集电极线路。

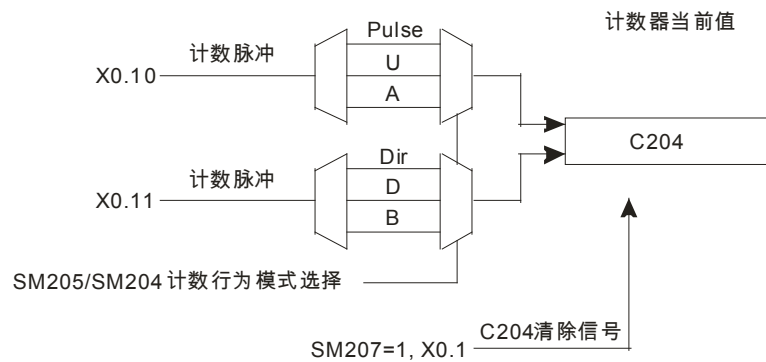
*2. 只有 AH05PM/10PM-5A 需接 S/S。AH15PM-5A 机种之 X0.10~X0.15 需接 S/S

*3. U/D：上数/下数，P/D：脉冲/方向，A/B：A 相/B 相。

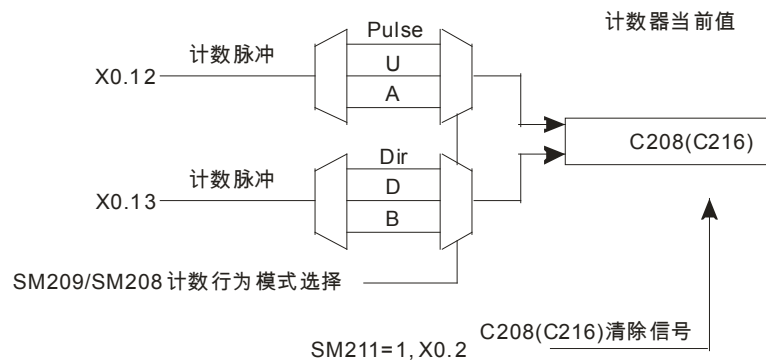
- C200 由 SM200/SM201 选择计数模式、脉冲由 X0.8、X0.9 输入；清除信号由 SM203 允许、由 X0.0 输入 (上升沿触发)。



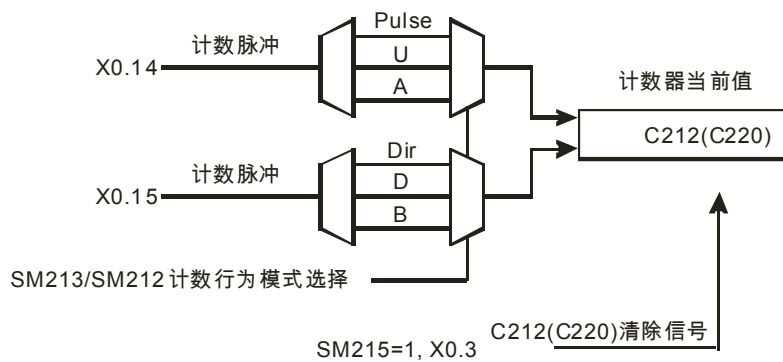
- C204 由 SM204/SM205 选择计数模式、脉冲由 X0.10、X0.11 输入；清除信号由 SM207 允许、由 X0.1 输入。



- C208 由 SM208/SM209 选择计数模式、脉冲由 X0.12、X0.13 输入；清除信号由 SM211 允许、由 X0.2 输入，其中第 4 组计数器 C216 也会跟着计数，此为第 1 组虚拟计数器。



- C212 由 SM212/SM213 选择计数模式、脉冲由 X0.14、X0.15 输入；清除信号由 SM215 允许、由 X0.3 输入，其中第 5 组计数器 C220 也会跟着计数，此为第 2 组虚拟计数器。



2. 计数器当前值即被清除，若为停电保持型（积分型）计数器，则会将停电前的当前值及计数器接点状态存储着，重新上电后会继续累计。
3. 计数器当前值由 2,147,483,647 再往上累计时则变为-2,147,483,648。同理，计数器当前值由 -2,147,483,648 再往下递减时，则变为 2,147,483,647。

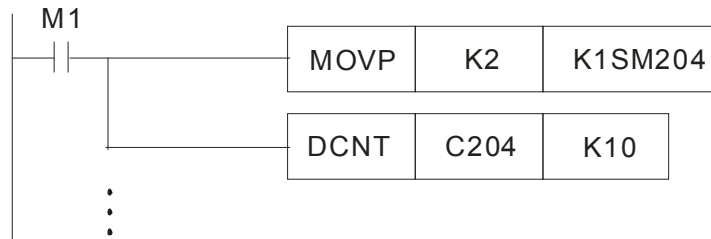
范例说明：

使用第 1 组高速计数器计数 A/B 相信号，其设定步骤如下：

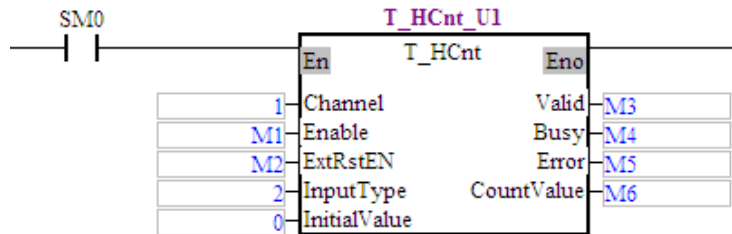
- (1) 首先设定第 1 组计数器模式为 A/B 相（一倍频）计数模式（K1SM204=K2）。
- (2) 启动 C204 开始计数。

上述步骤 1~2 之范例程序如下：

■ 梯形图形式



■ 功能块形式



- (3) 若设定由外部信号清除目前计数值，则模式设定为（K1SM204=16#A）。

■ 梯形图形式:K1SM204=16#A

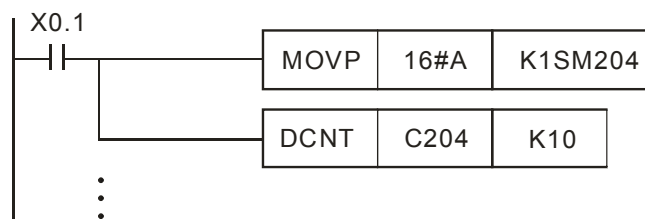
SM207	SM206	SM205	SM204
1	0	1	0

■ 功能块形式：将 ExRstEn 引脚设定为 ON

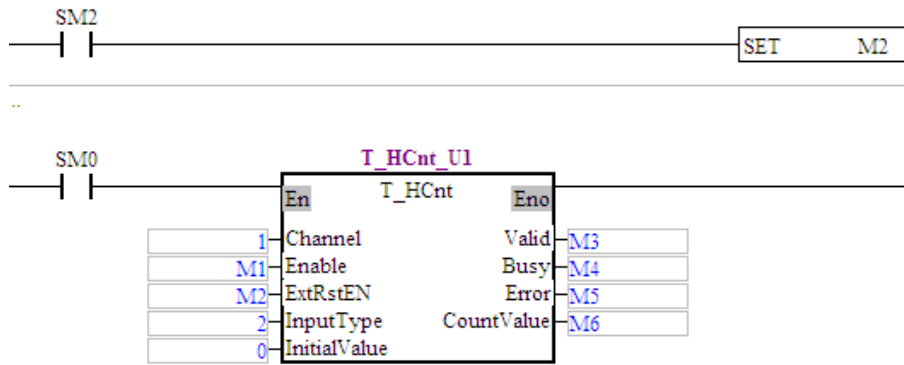
- (4) 启动 C204 开始计数，当接点 X0.1 为 ON，C204 立即被清除为零。

上述步骤 3~4 之范例程序如下：

■ 梯形图形式



■ 功能块形式



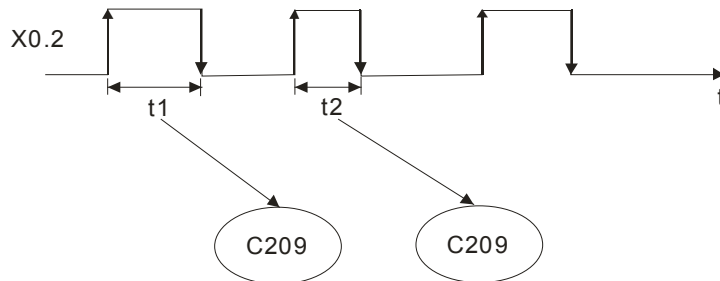
10.2 高速定时器

下表为高速定时器使用设定：

组别	计数装置	计时模式				启动计时外部信号	计时储存装置						
		装置	设定值										
0	C200	K1SM200	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr><tr><td>-</td><td>启动计时</td><td>-</td><td>模式设定</td></tr></table>	Bit3	Bit2	Bit1	Bit0	-	启动计时	-	模式设定	X0.0	C201
Bit3	Bit2	Bit1	Bit0										
-	启动计时	-	模式设定										
1	C204	K1SM204	Bit2：启动计时功能	X0.1	C205								
2	C208	K1SM208	Bit0：模式设定 b=0:一般模式·计时区间为上升沿至下降沿	X0.2	C209								
3	C212	K1SM212	b=1:周期模式·计时区间为上升沿至下一个上升沿	X0.3	C213								

范例说明一：使用第三组定时器为一般模式

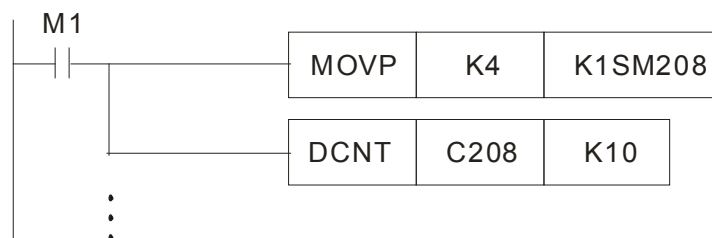
- 首先设定第三组定时器计时模式为一般模式 (Bit0=0)、启动计时功能 (Bit2=1)，则装置 K1SM208=K4。
- 启动 C208 开始计数，计算 X0.2 外部触发信号上升沿至信号的下降沿之间隔时间，将此值写入 C209，写入数值单位为 0.01us，如下图所示。



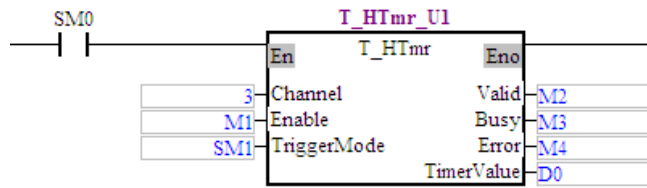
一般模式·单位 0.01μs

上述步骤之范例程序如下：

■ 梯形图形式

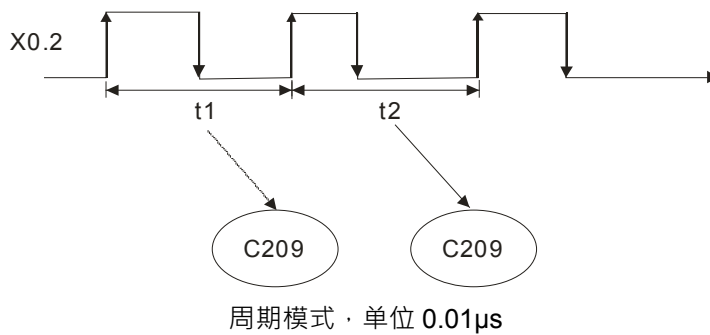


■ 功能块形式



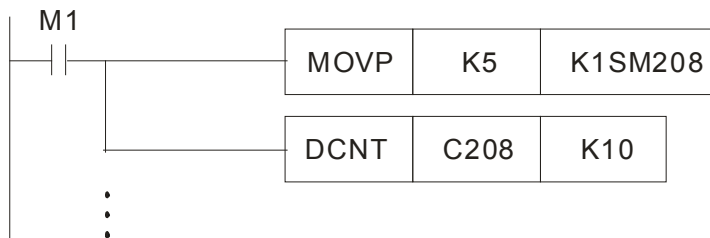
范例说明二：指定使用第 2 组定时器为周期模式

1. 若设定计时方式为周期模式 (Bit0=1)，则装置 K1SM208=K5。
2. 启动 C208 开始计数，计算 X0.2 外部触发信号上升沿至下个上升沿之间隔时间，将此值写入 C209，写入数值单位为 0.01us，如下图所示。

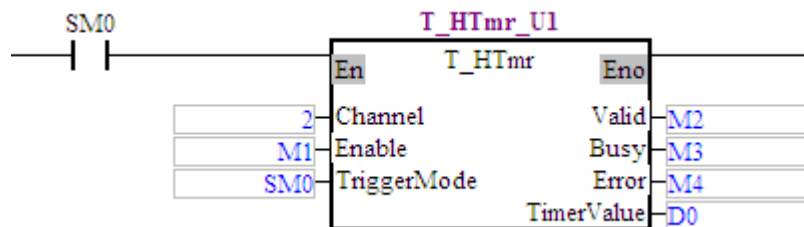


上述步骤之范例程序如下：

■ 梯形图形式

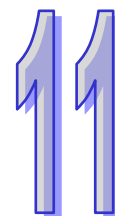


■ 功能块形式



3. 周期模式适用于测频应用

MEMO



第11章 高速捕捉与比较功能

目录

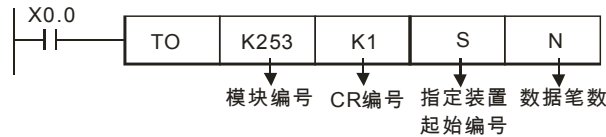
11.1	高速比较与捕捉功能使用格式	11-2
11.2	比较功能	11-2
11.3	清除输出功能	11-9
11.4	捕捉功能	11-10
11.5	屏蔽功能	11-15

11.1 高速比较与捕捉功能使用格式

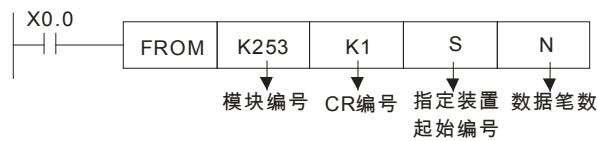


AH500 运动定位模块透过 FROM/TO 指令设定/读取控制高速比较或是捕捉功能状态与数据，以下说明如何使用 FROM/TO 指令来设定/读取高速比较与捕捉功能的状态。

■ 设定控制

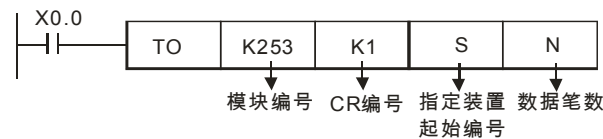


■ 读取状态



11.2 比较功能

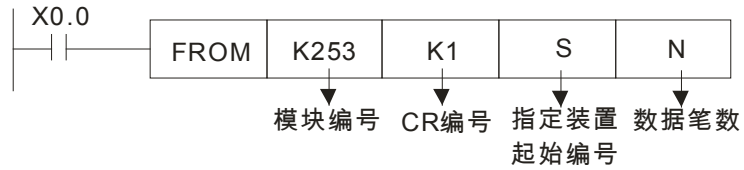
■ 设定控制



各笔数据定义如下：

位置	设定控制
S	起始组别 n (n=0~7)
S ₊₁	0
S ₊₂ · S ₊₃	组别 n 控制寄存器 CRn
S ₊₄ · S ₊₅	组别 n 数据寄存器 DRn
S ₊₆ · S ₊₇	组别 n+1 控制寄存器 CRn
S ₊₈ · S ₊₉	组别 n+1 数据寄存器 DRn
:	:
S ₊₃₀ · S ₊₃₁	组别 n+7 控制寄存器 CRn
S ₊₃₂ · S ₊₃₃	组别 n+7 数据寄存器 DRn
S ₊₅₀	数据长=2+m*4，m=使用组数，最多 8 组

■ 读取状态



各笔数据定义如下：

位置	读取各计数器状态
S	起始组别 n (n=0~7)
S ₊₁	0
S ₊₂ · S ₊₃	组别 n 控制寄存器 CRn
S ₊₄ · S ₊₅	组别 n 数据寄存器 DRn
S ₊₆ · S ₊₇	组别 n+1 控制寄存器 CRn
S ₊₈ · S ₊₉	组别 n+1 数据寄存器 DRn
:	:
S ₊₃₀ · S ₊₃₁	组别 n+7 控制寄存器 CRn
S ₊₃₂ · S ₊₃₃	组别 n+7 数据寄存器 DRn
S ₊₅₀	数据长=2+m*4 · m=使用组数 · 最多 8 组

■ 设定控制/读取

(1) 在各笔数据中的控制寄存器 CRn，设定为高速比较模式时格式如下：

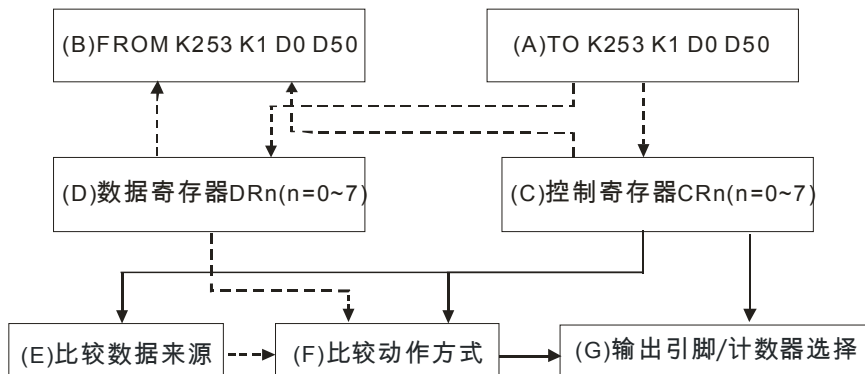
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
设定项目					比较结果输出				输出动作		比较条件		比较数据来源			

设定项目	Bit	设定值	AH20MC-5A	AH10PM-5A /AH15PM-5A	AH05PM-5A
比较数据来源	[3-0]	0	第 1 轴当前位置	第 1 轴当前位置	第 1 轴当前位置
		1	第 2 轴当前位置	第 2 轴当前位置	第 2 轴当前位置
		2	第 3 轴当前位置	第 3 轴当前位置	第 3 轴当前位置
		3	第 4 轴当前位置	第 4 轴当前位置	第 4 轴当前位置
		4	C200 计数值	C200 计数值	C200 计数值
		5	C204 计数值	C204 计数值	
		6	C208 计数值	C208 计数值	
		7	C212 计数值	C212 计数值	
比较条件	[5-4]	0	判断为“捕捉功能”，见第 11.4 章		
		1	等于 (=)	等于 (=)	等于 (=)
		2	大于等于 (≥)	大于等于 (≥)	大于等于 (≥)

设定项目	Bit	设定值	AH20MC-5A	AH10PM-5A /AH15PM-5A	AH05PM-5A
		3	小于等于 (\leq)	小于等于 (\leq)	小于等于 (\leq)
输出动作	[7-6]	0	触发 (Set)	触发 (Set)	触发 (Set)
		1	重置 (Rst)	重置 (Rst)	重置 (Rst)
		2、3	输出不动作	输出不动作	输出不动作
比较结果	[11-8]	0	Y0.8	Y0.8	Y0.8
		1	Y0.9	Y0.9	Y0.9
		2	Y0.10	Y0.10	
		3	Y0.11	Y0.11	
		4	清除 C200 计数值	清除 C200 计数值	清除 C200 计数值
		5	清除 C204 计数值	清除 C204 计数值	
		6	清除 C208 计数值	清除 C208 计数值	
		7	清除 C212 计数值	清除 C212 计数值	

数据寄存器 DCRn 设定格式：32BIT 的比较值

- (2) 高速比较的功能如下图所示，使用 FROM/TO 指令读取/写入设定值去控制运作方式及比较数据，下图说明高速比较功能的运作方式。



※ 虚线为数据流，实线为控制流程

方块图 (A)：使用 TO 指令写入数据至控制寄存器 (方块图 C) 及数据寄存器 (方块图 D)。

方块图 (B)：使用 FROM 指令读取控制寄存器 (方块图 C) 及数据寄存器 (方块图 D)。

方块图 (C)：控制数据寄存器依照其接收 TO 指令数据去设定比较数据来源 (方块图 E)、比较运作方式 (方块图 F) 及输出引脚设定 (方块图 G)。

方块图 (D)：储存 TO 指令写入的比较数据，用与方块图 (E) 所设定的数据来源，当比较运作时互相比。

方块图 (E)：比较数据来源有四轴当前位置，C200、C204、C208 和 C212 计数值等八种选择，其中 C200/C204/C208/C212 可参考前面高速计数器章节。

方块图 (F) : 比较方式有等于、大于等于和小于等于三种。当方块图 (D) 数据与方块图 (E) 所设定数据来源比较方式成立，而触发方式设定为 Set (Bit7-6=0)，即方块图 (G) 之设定引脚 (Y0.8、Y0.9、Y0.10 和 Y0.11) 设定 ON 或是计数器 (C200/C204/C208/C212) 启动重致 (清除计数器数值为零，且不能再计数)，若触发方式设定为 Rst (Bit7-6=1)，就对设定的引脚设定为 OFF 或是解除计数器重致状态 (计数器可以计数)。

方块图 (G) : 当比较成立，触发引脚选择设定或是计数器设定，依机种不同分别有 Y0.8、Y0.9、Y0.10、Y0.11、C200、C204、C208 与 C212 六种选择设定。

其高速比较功能的运作流程为：方块图 (A) TO 指令写入数据设定相关控制寄存器→方块图 (E) 设定的数据来源数据与方块图 (D) 数据寄存器数据比较符合方块图 (F) 设定→立即根据方块图 (G) 引脚状态输出，或是 C200/C204/C208/C212 清除为零停止计数或是可再计数。

■ 范例说明

【功能说明】

以下范例使用手轮装置，将脉冲传给 C204 高速计数，机种以 AH10PM-5A 为例，条件如下：

- 当计数值超过 100 时触发 Y0.9，
- 超过 300 时重置 Y0.9；

程序中使用两组比较器，一组用来触发 Y0.9，另外一组用于重置 Y0.9。在 Y0.9 触发时，AH500 运动定位模块外壳上并无显示灯号，但可透过外部线路得知 Y0.9 是否为 ON，故在外部接线中将 C1 端子接 24G，Y0.9 接 X0.2-，X0.2+接+24V，以及 X0.10 和 X0.11 外接一手摇轮。

【操作步骤】

1. 启动 O100 时先执行高速比较初始设定
 - D0=0→设定第一组比较起始组别 n=0
 - D1=0
 - D20=10→TO 指令写入 10 笔数据，有两组高速比较数据
 - D60=10→FROM 指令读取 10 笔数据，有两组高速比较数据
2. 设定 M1，设定两组高速比较数据
 - 第一组 控制数据 D3，D2=16#125→比较来源 C204 (Bit3-0=5)，大于等于设定 (Bit5-4=2)，触发设定 (Bit7-6=0)，触发引脚设定 Y0.9 (Bit11-8=1)
 - 第一组 暂存数据 D5，D4=K100 设定比较值大于等于 100 将触发
 - 第二组 控制数据 D7，D6=16#165→比较来源 C204 (Bit3-0=5)，大于等于设定 (Bit5-4=2)，触发清除 (Bit7-6=1)，触发引脚设定 Y0.9 (Bit11-8=1)
 - 第二组 暂存数据 D9，D8=K300 设定比较值大于等于 300 将触发
3. 设定 M2 启动两组高速比较功能

4. 设定 M3 读取两组高速比较功能设定状态，如下图所示。



装置	数值...	值	批注
C204	d32u	0	Y轴手轮计数值
D44	d32u	100	读取到n=0数据寄存器数据D5..D4
D48	d32u	300	读取到n=1数据寄存器数据D9..D8
D40	d16u	0	读取D0数据
D41	d16u	0	读取D1数据
D42	h32	0000124	读取到n=0控制寄存器数据D3..D2
D44	d32u	100	读取到n=0数据寄存器数据D5..D4
D46	h32	0000164	读取到n=1控制寄存器数据D7..D6
D48	d32s	300	读取到n=1数据寄存器数据D9..D8

5. 设定 M4 设定 C204 为 PD (脉冲+方向)，然后设定 M5，启动计数。
 6. 转动手摇轮，看 C204 是否有计数，如下图所示。

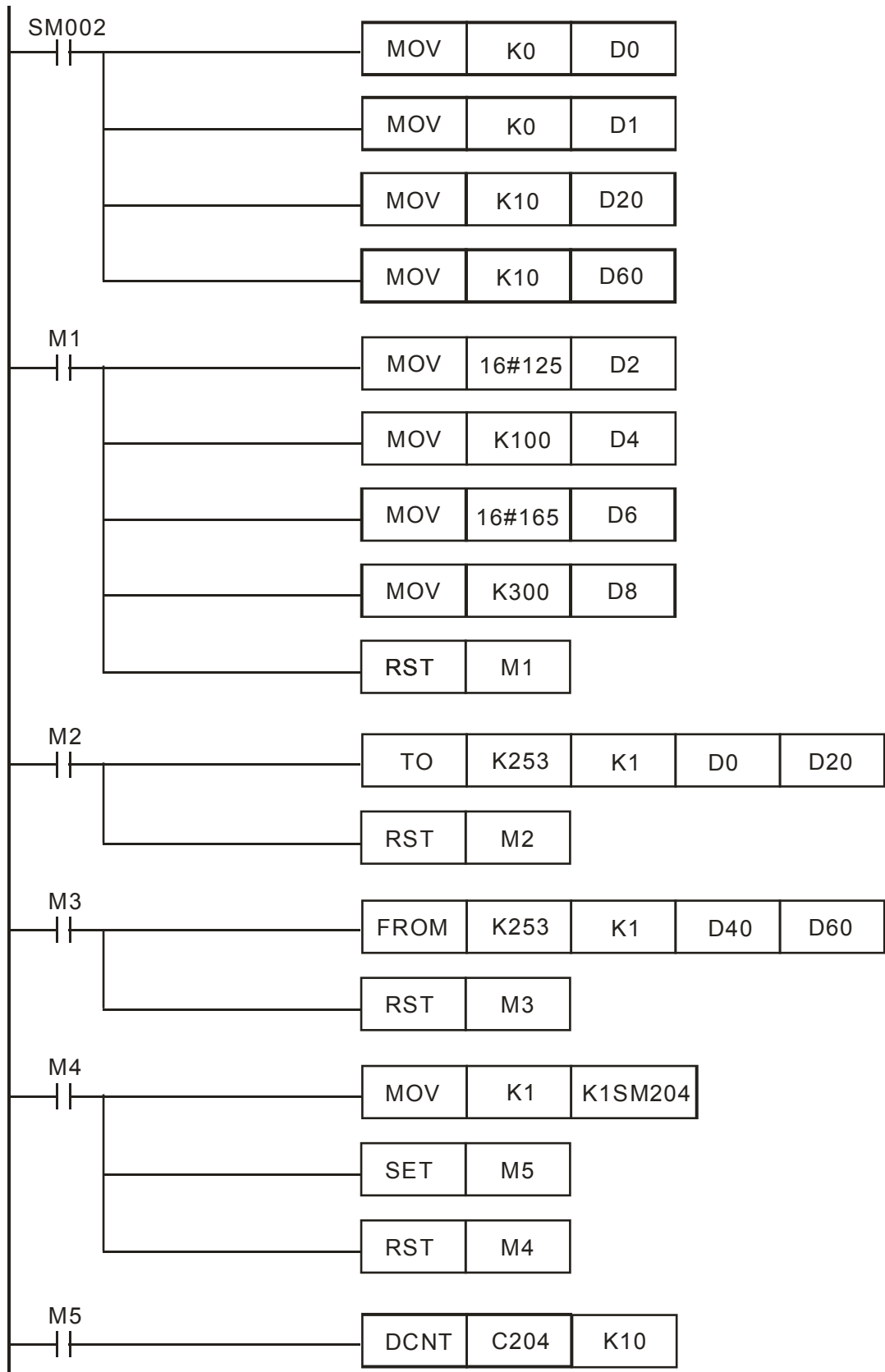


装置	数值...	值	批注
C204	d32u	95	Y轴手轮计数值
D44	d32u	100	读取到n=0数据寄存器数据D5..D4
D48	d32u	300	读取到n=1数据寄存器数据D9..D8

7. 转动手摇轮，当 C204 值大于 100，使用 PMSoft 观察 X0.2 是否为 ON。
 8. 手轮持续转动，当 C204 值大于 300，观察 X0.2 是否为 OFF。

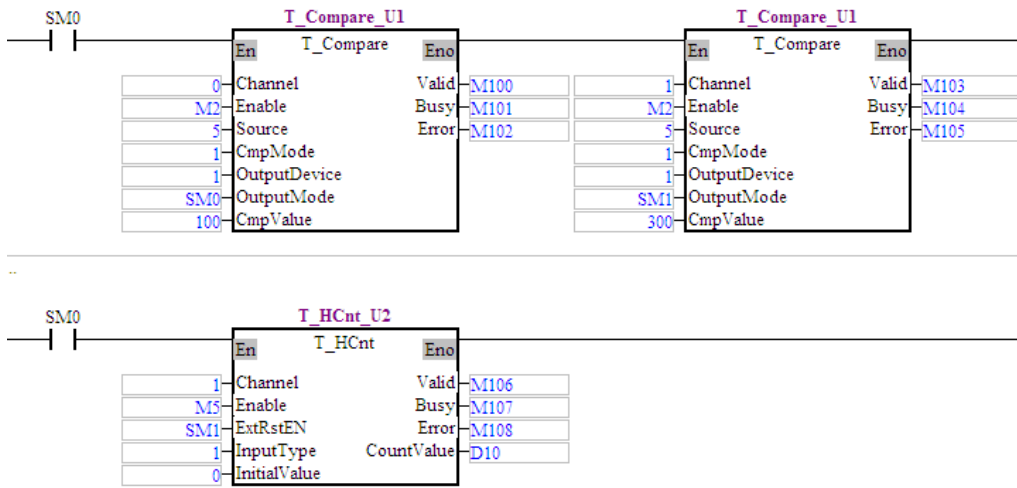
【PMSoft 程序】

■ 梯形图形式：



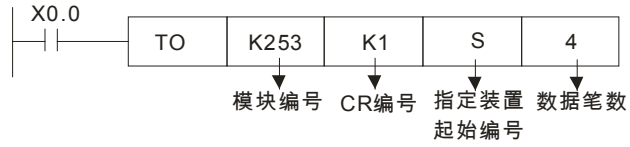
■ 功能块形式：

11



11.3 清除输出功能

清除输出功能用于高速比较时输出状态清除，下图说明清除功能的设定方式：



各笔数据定义如下：

位置	清除输出
S	0
S_{+1}	1
$S_{+2} \cdot S_{+3}$	输出状态

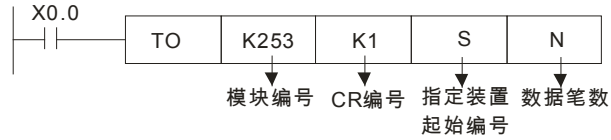
其中设定数据 S_{+2} 为设定高速比较输出对应的功能，下图所示为高速比较输出的设定值。

设定项目	Bit	AH20MC-5A	AH10PM-5A /AH15PM-5A	AH 05PM-5A
比较结果	0	Y0.8	Y0.8	Y0.8
	1	Y0.9	Y0.9	Y0.9
	2	Y0.10	Y0.10	-
	3	Y0.11	Y0.11	-
	4	清除 C200 计数值	清除 C200 计数值	清除 C200 计数值
	5	清除 C204 计数值	清除 C204 计数值	-
	6	清除 C208 计数值	清除 C208 计数值	-
	7	清除 C212 计数值	清除 C212 计数值	-

11.4 捕捉功能

11

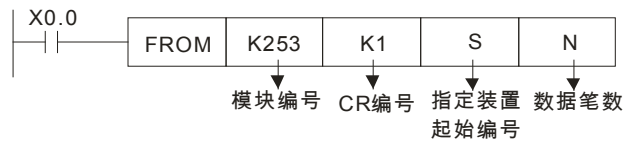
■ 设定控制



各笔数据定义如下

位置	设定控制
S	起始组别 n (n=0~7)
S ₊₁	0
S ₊₂ · S ₊₃	组别 n 控制寄存器 CRn
S ₊₄ · S ₊₅	组别 n 数据寄存器 DRn
S ₊₆ · S ₊₇	组别 n+1 控制寄存器 CRn
S ₊₈ · S ₊₉	组别 n+1 数据寄存器 DRn
:	:
S ₊₃₀ · S ₊₃₁	组别 n+7 控制寄存器 CRn
S ₊₃₂ · S ₊₃₃	组别 n+7 数据寄存器 DRn
S ₊₅₀	数据长=2+m*4 · m=使用组数 · 最多 8 组

■ 读取状态



各笔数据定义如下

位置	读取各计数器状态
S	起始组别 n (n=0~7)
S ₊₁	0
S ₊₂ · S ₊₃	组别 n 控制寄存器 CRn
S ₊₄ · S ₊₅	组别 n 数据寄存器 DRn
S ₊₆ · S ₊₇	组别 n+1 控制寄存器 CRn
S ₊₈ · S ₊₉	组别 n+1 数据寄存器 DRn
:	:
S ₊₃₀ · S ₊₃₁	组别 n+7 控制寄存器 CRn
S ₊₃₂ · S ₊₃₃	组别 n+7 数据寄存器 DRn
S ₊₅₀	数据长=2+m*4 · m=使用组数 · 最多 8 组

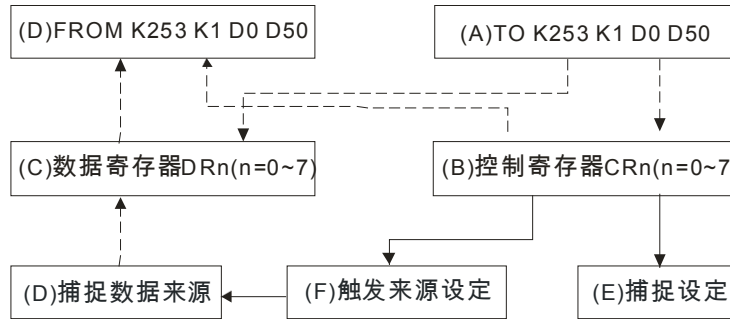
(1) 在各笔数据中控制寄存器 CRn，当为捕捉模式时设定格式如下：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
设定项目	触发来源设定								捕捉设定				捕捉数据来源			

设定项目	Bit	设定值	AH20MC-5A	AH10PM-5A /AH15PM-5A	AH 05PM-5A
捕捉资料来源	[3-0]	0	第 1 轴当前位置	第 1 轴当前位置	第 1 轴当前位置
		1	第 2 轴当前位置	第 2 轴当前位置	第 2 轴当前位置
		2	第 3 轴当前位置	第 3 轴当前位置	第 3 轴当前位置
		3	第 4 轴当前位置	第 4 轴当前位置	第 4 轴当前位置
		4	C200 计数值	C200 计数值	C200 计数值
		5	C204 计数值	C204 计数值	-
		6	C208 计数值	C208 计数值	-
		7	C212 计数值	C212 计数值	-
捕捉设定	[5-4]	0	捕捉设定为 0		
外部触发来源设定	[15-12]	0	X0.0	X0.0	X0.0
		1	X0.1	X0.1	X0.1
		2	X0.2	X0.2	-
		3	X0.3	X0.3	-
		4	-	-	-
		5	-	-	-
		6	-	-	-
		7	-	-	-
		8	X0.8	X0.8	X0.8
		9	X0.9	X0.9	X0.9
		10	X0.10	X0.10	-
		11	X0.11	X0.11	-
		12	X0.12	X0.12	X0.12
		13	X0.13	X0.13	X0.13
		14	X0.14	X0.14	-
		15	X0.15	X0.15	-

数据寄存器 DCRn 设定格式：32BIT 的捕捉值，可设定其初始值，触发后，捕捉值会被更新。

- (2) 一般程序扫描执行读取四轴位置数据或是 C200/C204/C208/C212 计数数据会有误差值，使用捕捉功能透过外部输入信号立即执行读取数据的动作，可以避免误差发生的情形。下图说明捕捉功能的运作方式：



方块图 (A)：使用 TO 指令写入数据至控制寄存器 (方块图 B) 及数据寄存器 (方块图 C)。

方块图 (B)：控制寄存器依照其接收 TO 指令数据去设定捕捉数据来源 (方块图 D)、捕捉设定 (方块图 E) 及触发来源引脚设定 (方块图 F)。

方块图 (C)：储存因外部硬件信号触发而捕捉到当下的数据来源数值(方块图 (D))。

方块图 (D)：捕捉的来源为：

1. 4 轴当前位置
2. C200/C204/C208/C212 等 4 组高速计数器。

方块图 (E)：设定捕捉功能。

方块图 (F)：触发捕捉信号来源。

方块图 (G)：使用 FROM 指令读取数据寄存器 (方块图 C) 及控制寄存器 (方块图 B)，其中数据寄存器存放数据为捕捉到的数据。

捕捉功能的运作流程：

TO 指令写入数据设定相关控制寄存器 (方块图 A) → 当外部触发信号产生 (方块图 F) → 立即捕捉方块图 (D) 所设定的数据来源，将此数据放入方块图 (C) 的数据寄存器 → 使用 FROM 指令读取捕捉到的数据。

■ 范例说明

【功能说明】

启动 C204 (高速计数功能，当外部信号 X0.1 触发时补捉 C204 计数值。

【操作步骤】

1. O100 运作 SM002 瞬间为 ON 捕捉初始设定。
 - D0=0 → 设定第一组捕捉起始组别 n=0。
 - D1=0。
 - D20=6 → TO 指令写入 6 笔数据，只有一组捕捉资料。
 - D60=6 → FROM 指令读取 6 笔数据，只有一组捕捉数据。

2. Set M1 设定捕捉
 - 控制数据 D3..D2=16#1004→捕捉来源 C204 (Bit3-0=5) · 捕捉模式设定 (Bit5-4=0) · 触发捕捉信号 X0.1 (Bit15-12=1)。
 - 暂存资料 D5..D4=K100 可任意值忽略。
3. Set M2 使用 TO 指令写入 · 捕捉功能启动。
4. Set M3 使用 FROM 指令读取第一组捕捉设定状态 · 如下为读取状态值。

装置	数值...	值	批注
C204	d32s	0	
D44	d32u	100	对映D5..D4数据寄存器
D40	d16s	0	对映D0
D41	d16s	0	对映D1
D42	h32	00005004	对映D3..D2数据寄存器
D44	d32u	100	对映D5..D4数据寄存器

5. Set M4 设定 C204 为 AB (脉冲+方向) · 设定 M5 · 启动计数。
6. 转动手轮 · 监看 C204 是否有计数 · 如下图所示。

装置	数值...	值	批注
C204	d32s	244	
D44	d32u	100	对映D5..D4数据寄存器
D40	d16s	0	对映D0
D41	d16s	0	对映D1
D42	h32	00005004	对映D3..D2数据寄存器
D44	d32u	100	对映D5..D4数据寄存器

7. 持续转动手摇轮 · 并使用外部信号触发 X0.1。
8. Set M3 读取第一组捕捉设定状态 · 677 为 X0.1 触发时捕捉到 C204 的计数值。

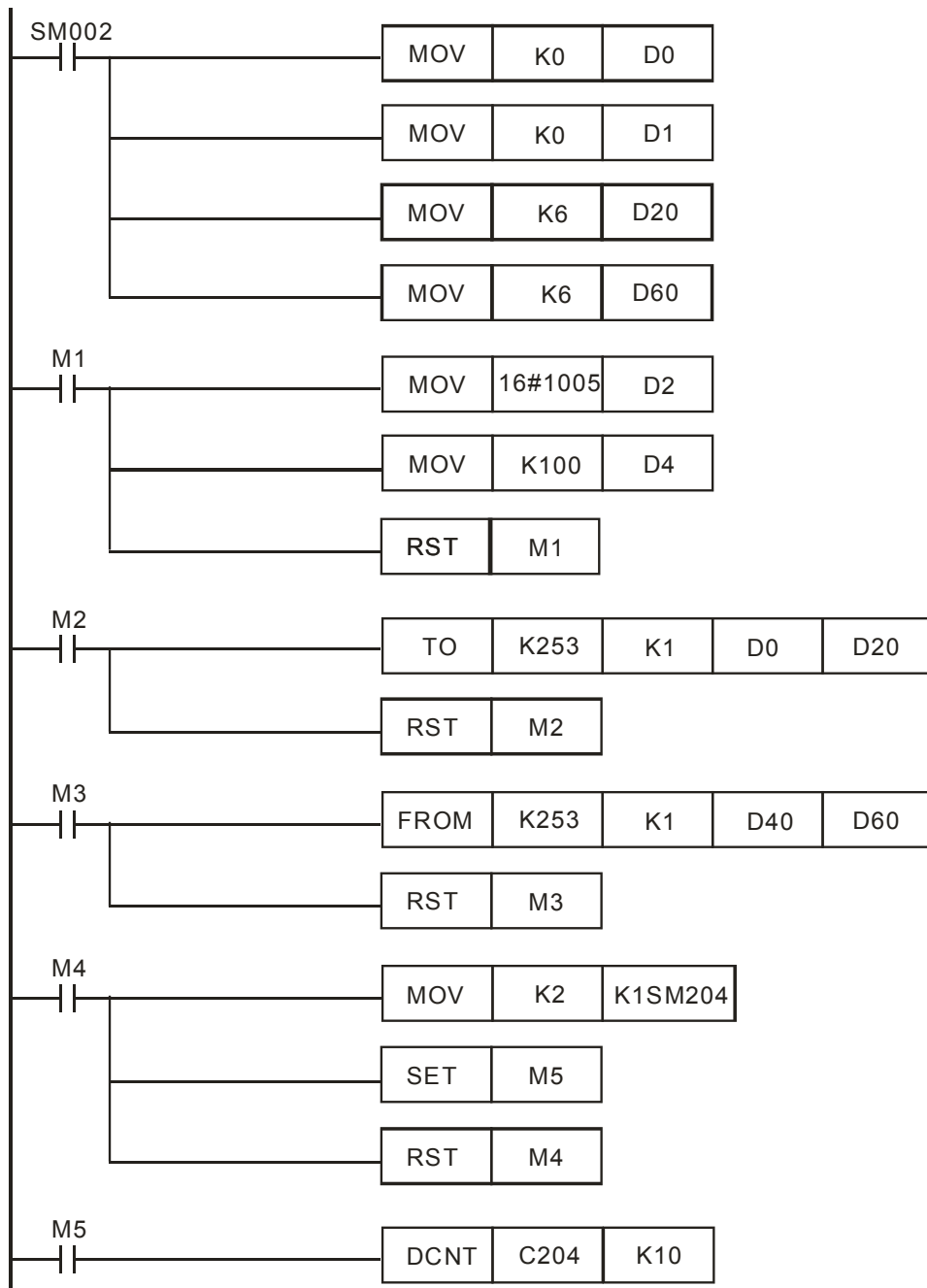
装置	数值...	值	批注
C204	d32s	726	
D44	d32u	677	对映D5..D4数据寄存器
D40	d16s	0	对映D0
D41	d16s	0	对映D1
D42	h32	00005004	对映D3..D2数据寄存器
D44	d32u	677	对映D5..D4数据寄存器

X0.1触发捕捉到 C204计数值

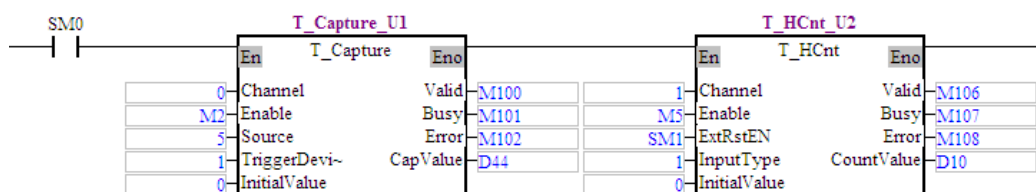
【PMSoft 程序】

- 梯形图型式：

11

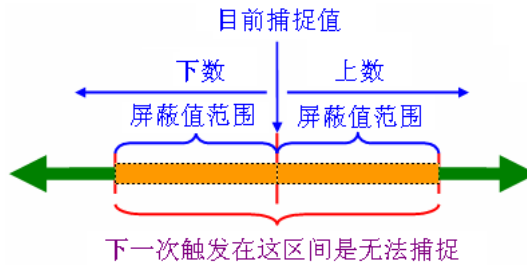


- 功能块形式：

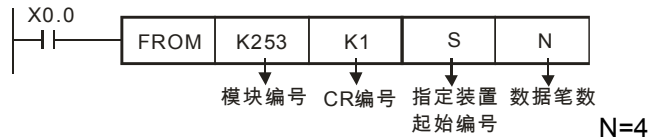


11.5 屏蔽功能

屏蔽功能用于高速捕捉模式下，当目前值与上一次捕捉值的相对值小于屏蔽值，捕捉信号是无效，如下所示为屏蔽概念：



下图说明屏蔽功能的设定方式：

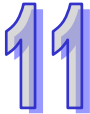


各笔数据定义如下：

位置	设定屏蔽
S	0
S ₊₁	2
S ₊₂ · S ₊₃	屏蔽值

当屏蔽功能启动之后，8 组捕捉功能都共享同一个屏蔽功能。

MEMO



第12章 以太网网络设置

目录

12.1	功能简介	12-2
12.2	功能规格	12-2
12.3	参数功能介绍	12-2
12.4	PMSoft 通讯功能说明	12-3
12.5	MODBUS 通讯功能说明	12-7
12.6	以太网网络错误说明及排除方式	12-9

12.1 功能简介

AH500 运动定位模块备有以太网通讯口，可使用一般网络通讯线做为网络装置的数据交换使用，其具备功能如下：

- ◆ 支持与 PC 端软件 PMSoft 做程序上下载、实时监控
- ◆ 支持标准 MODBUSTCP Slave 端功能

12

12.2 功能规格

- 以太网连接器

项目	规格
传输方式	以太网
电气隔离	500VDC
接头	可插拔式连接器 (5.08mm)
传输电缆	四条通讯线

- 通讯

项目	规格
数据类型	TCP/IP
串行传输速度	10M/100M (位/秒)
最大传输距离	100M (100 公尺)

12.3 参数功能介绍

SR808、SR809：以太网 IP 设置。

[说明]

设置模块以太网络的 IP，会占据到两个寄存器，初始值 IP 地址为 192.168.0.100。设置之格式请参考以下范例：IP：192.168.0.100

SR809		SR808	
H-byte	L-byte	H-byte	L-byte
192	168	0	100

12.4 PMSoft 通讯功能说明

■ 以太网网络 IP 设置

模块的以太网网络 IP 出厂默认值为 192.168.0.100，但可以透过设置寄存器 SR808、SR809 来修改模块的以太网网络 IP。模块的 IP 在每一次断电后，均不会保留之前的设置值，故每一次重新上电后将会自动恢复成 192.168.0.100。IP 地址另外可透过运动功能块设置，功能块外观如下图所示。

En	T_TcpIPAddr	Eno
Execute		Done
IPv4_1		Busy
IPv4_2		Error
IPv4_3		
IPv4_4		

输入引脚 IPv4_1 ~ IPv4_4 分别表示 IP 地址的格式，设置完毕后输出引脚 Done 显示为 ON。

■ 以太网网络与 PC 端 PMSoft 联机。

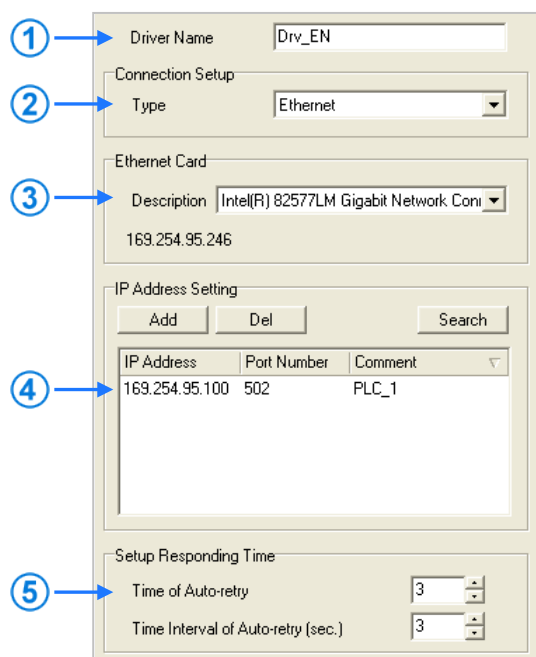
具有以太网网络接口的 AH500 运动定位模块配合 PM 编辑软件 PMSoft，可利用以太网网络方式做 PM 程序上下载以及实时监控，设置方式如下。

● 硬件配线

在硬件联机上利用一般网络线连接 AH500 运动定位模块的网络接口以及 PC 上的网络接口即可，当 AH500 运动定位模块与 PC 可联机时以太网网络指示灯会亮起，当指示灯未亮起，需检查硬件或 PC 设置是否有错误。

● COMMGR 设置

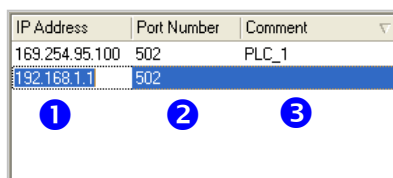
12



- ① 在 **Driver Name** 字段输入此 **Driver** 的识别名称。
- ② 在 **Connection Setup** 的 **Type** 字段下拉选择 **Ethernet** 的联机型式。
- ③ 在 **Ethernet Card** 的字段下拉选择欲建立联机信道的网络卡，而此字段左下角的数字即是所选网卡目前的 IP 地址。
- ④ 依据以太网络的特性，计算机可以与连接至同一个网域上的所有装置进行通讯，因此使用者可在 **IP Address Setting** 区域中，预先将可能透过此 **Driver** 进行通讯的装置 IP 地址建立起来。

➤ 按下 **Add** 按钮可在下方的列表中新增一组 IP 地址，之后请在各字段中输入相关信息。

- ❶ 在此输入联机装置的 IP 地址。
- ❷ 在此输入指定的通讯端口编号。
- ❸ 视使用者的需求在此字段输入此组 IP 的批注。



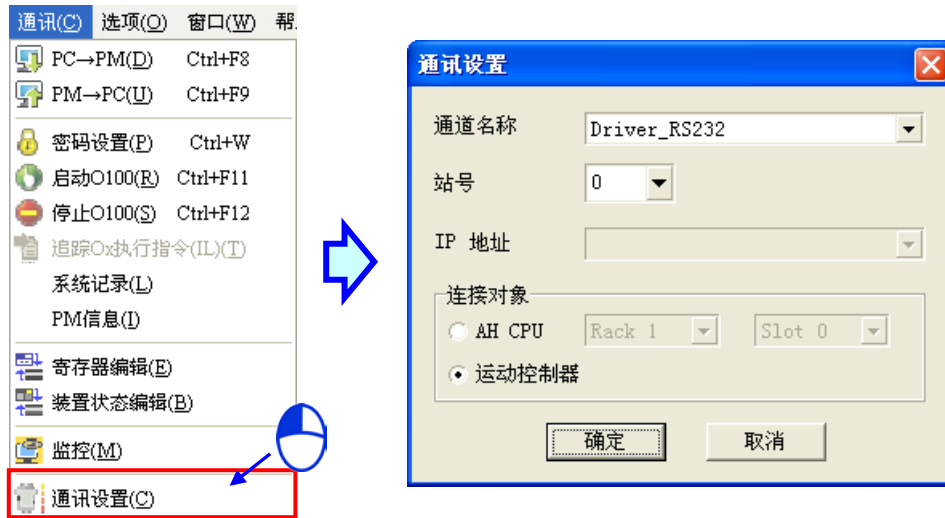
➤ 选取某组 IP 后按下 **Del** 按钮或直接按下键盘的 **Delete** 键即可将该组 IP 在列表中删除。

- ⑤ 设置通讯响应的相关参数。其中 **Time of Auto-retry** 参数是设置联机错误时的重试次数，而 **Time Interval of Auto-retry** 参数则是设置重试的时间间隔。

● PMSoft 设置流程

直接将计算机与运动主机上的 **Ethernet** 通讯口依前述完成连接与各项设置后，进行以下设置。

- (1) 开启 PMSOft 软件，在功能工具列中单击**通讯 (C)** > **通讯设置 (C)**。



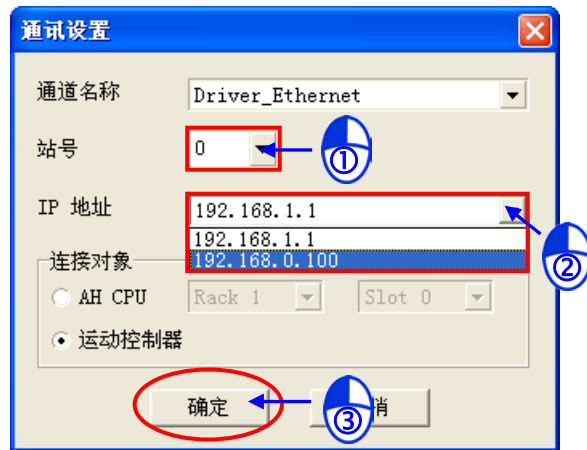
- (2) 在**通道名称**字段处，下拉选择欲使用的 **Driver**，且在开始联机之前，请务必确认该 **Driver** 已在 **COMMGR** 中被设为 **START** 状态，并且勾选连接对象为**运动控制器**，按下**确定**键完成设置。当选择不同的通道名称，下方的设置选项将随之改变，说明如下。



➤ Ethernet

在站号字段下拉选择欲联机的硬件站号，当站号设置为 0 时，则代表要进行广播通讯。之后再 IP 地址字段中，下拉选择事先在 COMMGR 中所建立好的 IP 地址。

12



12.5 MODBUS通讯功能说明

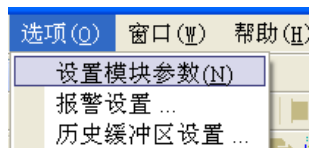
■ 以太网网络 IP 设置

模块的以太网网络 IP 出厂默认值为 192.168.0.100，但可以透过设置寄存器 SR808、SR809 或运动功能块来修改模块的以太网网络 IP。

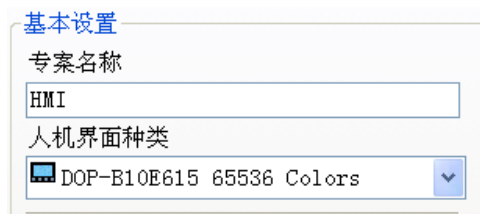
■ 利用以太网网络的 MODBUS TCP 功能联机。

在以太网网络联机功能下，可做为一般 MODBUS TCP Slave 使用。以搭配人机为例，人机设置方式如下：

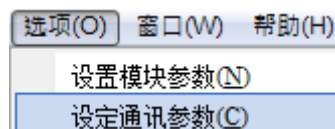
1. 在开启文件后在工具列中『选项』选择『设置模块参数』。



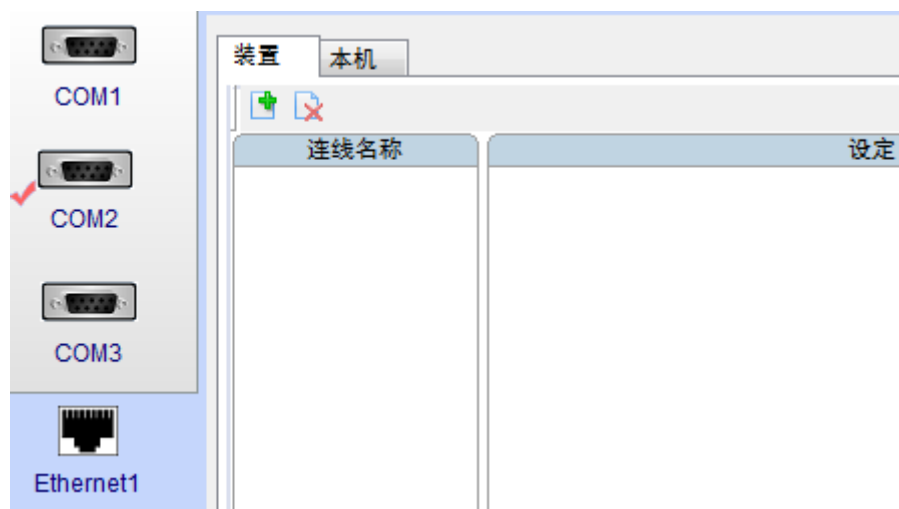
2. 在『一般』页面中，设置人机接口种类为 DOP-B10E615。



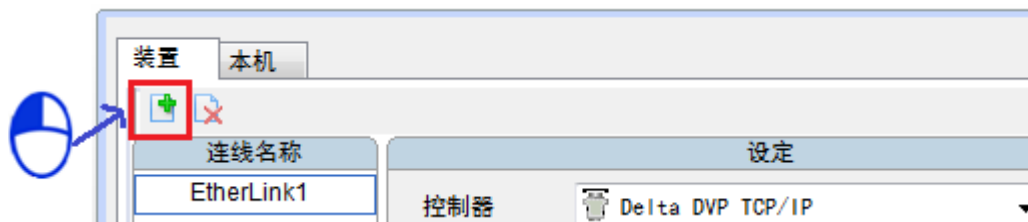
3. 在工具栏中『选项』选择『设定通讯参数』。



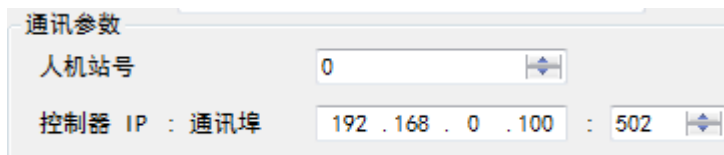
4. 选择『Ethernet』页面



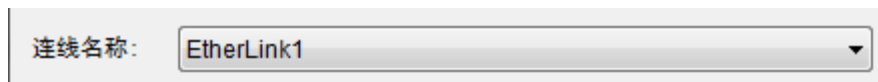
5. 新增通讯对象之联机，单击『新增』，设定连线名称，控制器选择 Delta DVP TCP/IP。



6. 设定通讯对象 IP，在通讯参数中 IP 位置，设定 AH500 运动定位模块的预定的 IP 位置。



7. 在使用的组件的输入页面中的联机选择先前定义的装置名称，即可利用以太网将此组件定义的内存位置做操作。



目前提供的联机个数最大为 12 点，可同时建立 12 条联机，超过此最大联机个数则无法在联机。

12.6 以太网网络错误说明及排除方式

错误状况	处理方法
以太网路灯号未亮起	检查以太网网络线是否有正常连接
PMSoft 联机无法建立	检查网络的 IP 设置是否正确，若不为同一个群组则无法正确建立联机
MODBUS TCP 联机无法建立	<ol style="list-style-type: none">1. 检查以太网络的 client 端的以太网 Server IP 设置是否正确，并且所有的 MODBUS TCP 连线各数不能大于 12 条。2. 确认站号设定值是否正确.不确定站号时,可先将站号设定为 0

MEMO

12

第13章 扩展保存装置

目录

13.1	功能简介	13-2
13.2	参数功能介绍	13-2
13.3	G 码读取与执行	13-4
13.4	装置备份及回存	13-5
13.4.1	备份	13-5
13.4.2	回存	13-8
13.5	程序备份及回存	13-9
13.5.1	备份	13-9
13.5.2	回存	13-9
13.6	韧体更新	13-10

13.1 功能简介

AH500 运动定位模块提供存储卡的外部的扩展保存装置接口，此接口可使用的存储卡的格式为 FAT16、FAT32，最大保存为 4GB。存储卡对 AH500 运动定位模块用于以下四种功能：

1. G 码读取与执行
2. 装置备份及回存
3. 程序备份及回存
4. 韧带更新

以上四种功能，在存储卡中指定其文件放于以下四种路径内

1. \AHMotion\Gcode\
2. \AHMotion\Device\
3. \AHMotion\Program\
4. \AHMotion\bin\

13.2 参数功能介绍

■ 参数列表

参数名称	功能描述
SR200	存储卡备份 M 辅助继电器开始地址
SR201	存储卡备份 M 辅助继电器结束地址
SR202	存储卡备份 T 定时器开始地址
SR203	存储卡备份 T 定时器结束地址
SR204	存储卡备份 C16 位计数器开始地址
SR205	存储卡备份 C16 位计数器结束地址
SR206	存储卡备份 C32 位计数器开始地址
SR207	存储卡备份 C32 位计数器结束地址
SR208	存储卡备份 S 步进点开始地址
SR209	存储卡备份 S 步进点结束地址
SR210	存储卡备份 D 寄存器开始地址
SR211	存储卡备份 D 寄存器结束地址
SR212	存储卡备份 W 寄存器开始地址
SR213	存储卡备份 W 寄存器结束地址
SR214	存储卡备份控制寄存器

■ 参数介绍

1. SR200、SR201 存储卡备份 M 辅助继电器开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 M 辅助继电器的开始回存地址，以及结束回存地址。若回存地址中包含特 SM 时，则 SM 辅助继电器将不会被作回存动作，但在回存范围内的其它 M 辅助继电器，仍旧会被作回存动作。

2. SR202、SR203 存储卡备份 T 定时器开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 T 定时器的开始回存地址，以及结束回存地址。

3. SR204、SR205 存储卡备份 C16 位计数器开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 C16 位计数器的开始回存地址，以及结束回存地址。

4. SR206、SR207 存储卡备份 C32 位计数器开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 C32 位计数器的开始回存地址，以及结束回存地址。

5. SR208、SR209 存储卡备份 S 步进点开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 S 步进点的开始回存地址，以及结束回存地址。

6. SR210、SR211 存储卡备份 D 寄存器开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 D 寄存器的开始回存地址，以及结束回存地址。若回存地址中包含 SR 时，则 SR 寄存器将不会被作回存动作，但在回存范围内的其它 D 寄存器，仍旧会被作回存动作。

7. SR212、SR213 存储卡备份 W 寄存器开始地址/结束地址

[说明]

在使用装置回存动作时，决定回存 W 寄存器的开始回存地址，以及结束回存地址。

8. SR214 存储卡备份控制寄存器

[说明]

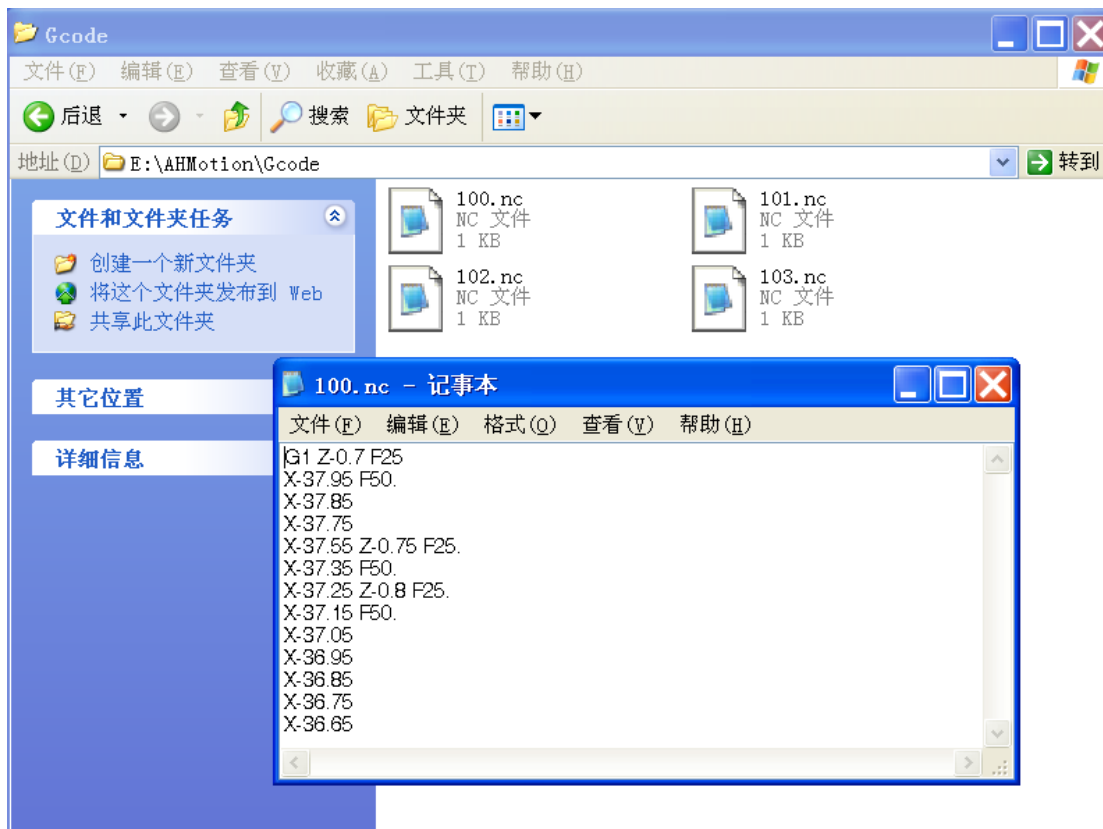
寄存器的使用定义如下：

Bit	名称	说明	Reset value
[12-0]	文件名名称	备份动作时为备份至存储卡上的文件名称。 回存动作时为从存储卡上读回的文件名称。	0x0000
13	存取控制命令	当此数值为 0 时表示执行装置或程序作回存动作。 当此数值为 1 时表示执行装置或程序作备份动作。 在备份装置时，所有的全装置均会备份。	0
14	程序存取	当此数值为 1 时，设备份或回存对象为程序，当操作完成后，此数值会自动归 0。	0
15	装置存取	当此数值为 1 时，设备份或回存对象为装置，当操作完成后，此数值会自动归 0。	0

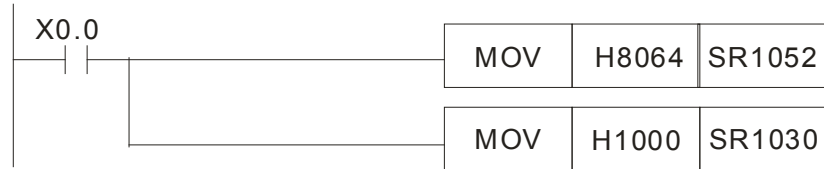
13

13.3 G码读取与执行

NC 文件的固定存放路径为存储卡根目录下\AHMotion\Gcode\数据夹内，文件名为 100~199，扩展名为 NC（不区分大小写），文件格式同一般文字文件，数据夹内可以摆放 100 个可供使用的 G 码文件。



这 100 个文件相当于运动子程序 ox100~199 (接续内部的 0x0~99)，檔名 100 为 ox100、檔名 101 为 ox101、...、依此类推，文件名 199 为 ox199。如下执行 ox100，主机会先开启文件 VAHMotion\Gcode\1.NC，读取文件后，并转码加工。



13.4 装置备份及回存

AH500 运动定位模块提供利用存储卡做为装置备份及回存的外部保存空间。

13.4.1 备份

关于装置备份的操作方式为：

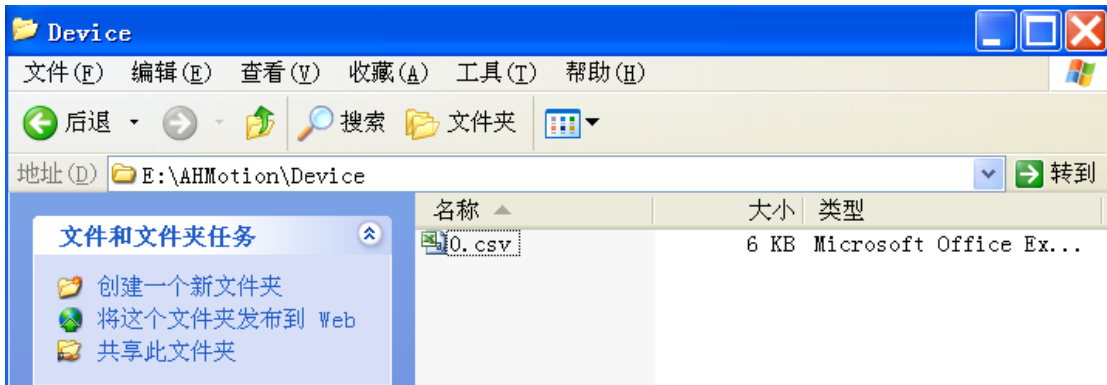
1. 用户利用 PMSoft 监控 AH20MC-5A、AH10PM-5A 或 AH15PM-5A，并且在监控状态下停止运行，或设定 SM072 为 0 停止 0100。



2. 利用监控表或外部通讯直接设定 SR214。
 - ◆ 设定 SR214 的 bit12 到 bit0，编号大小为备份文件名称。
 - ◆ 设定 SR214 的 bit13 为 1 表示作备份动作。
 - ◆ 设定 SR214 的 bit15 为 1 表示此次的备份对象为装置。

装置	数值型态	值	批注
SR214	b16	1010_0000_0000_0000	

例如设定 SR214 为 16#A000，表示对装置作备份，并且备份的文件名称为 0。因此在存储卡的 \AHMotion\Device\中可以看到 0.csv。



此 CSV 文件，包含所有装置的数据，并且在作此动作时无法选择备份装置范围，一律为备份所有装置。

关于此 CSV 可在一般窗口环境下作装置参数的修改，CSV 档格式如下：

	A	B	C	D	E	F	G	H	I	J
1	AHMotion SD Format V0.3									
2	PLC Type :									
3	PLC FWVer :									

文件开头为 AHMotion 的存储卡装置格式版本。

PLC Type：为表示为 AH500 运动定位模块的机种类别。

PLCFWVer：为此机种的韧体版本。

13

5	M									
6	0	0	1	0	1	1	0	1	1	0
7	0	0	1	0	1	1	0	1	1	0
8	0	0	1	...						
9										
10	T									
11	0	0	1	0	1	1	0	1	1	0
12	0	0	1	0	1	1	0	1	1	0
13	0	0	1	...						
14										
15	C16b									
16	0	0	1	0	1	1	0	1	1	0
17	0	0	1	0	1	1	0	1	1	0
18	0	0	1	...						
19										
20	C32b									
21	0	0	1	0	1	1	0	1	1	0
22	0	0	1	0	1	1	0	1	1	0
23	0	0	1	...						
24										
25	S									
26	0	0	1	0	1	1	0	1	1	0
27	0	0	1	0	1	1	0	1	1	0
28	0	0	1	...						
29										
30	C16W									
31	1234	0	0	0	0	0	0	0	0	0
32	0	0	4444	0	0	0	0	0	0	0
33	0	0	0	...						
34										
35	C32W									
36	0	0	0	44s	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0
38	0	0	0	...						
39										
40	D									
41	0	0	0	44s	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	...						
44										
45	W									
46	0	0	0	44s	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	...						

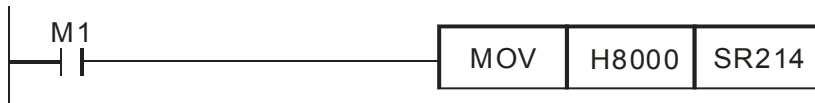
在叙述完此 CSV 所属的机种以及韧体版本后，为所有的装置数值区，若用户想修改装置内容，可直接利用此表格修改。在表格中 M 为 M 辅助继电器，T 为 T 定时器，C16b 和 C16W 为 C16 位计数器的状态以及计数量，C32b 和 C32W 为 C32 位计数器的状态以及计数量，D 为 D 寄存器，W 为 W 寄存器。

13.4.2 回存

用户若欲作装置的回存时，需注意在存储卡中的\AHMotion\Device\目录下，必须先存在欲回存的文件名称，关于装置备份的操作方式为：

1. 设定装置回存的范围。利用 SR200 到 SR213 决定此次装置对于，M、T、C16、C32、S、D、W 的回存范围。
2. 设定 SR214。
 - ◆ 设定 SR214 的 bit12 到 bit0，编号大小为备份文件名称。
 - ◆ 设定 SR214 的 bit13 为 0 表示作回存动作。
 - ◆ 设定 SR214 的 bit15 为 1 表示此次的备份对象为装置

13



例如设定 SR214 为 16#8000，表示将存储卡内的\AHMotion\Device\0.csv 作装置回存到 AH500 运动定位模块中。

13.5 程序备份及回存

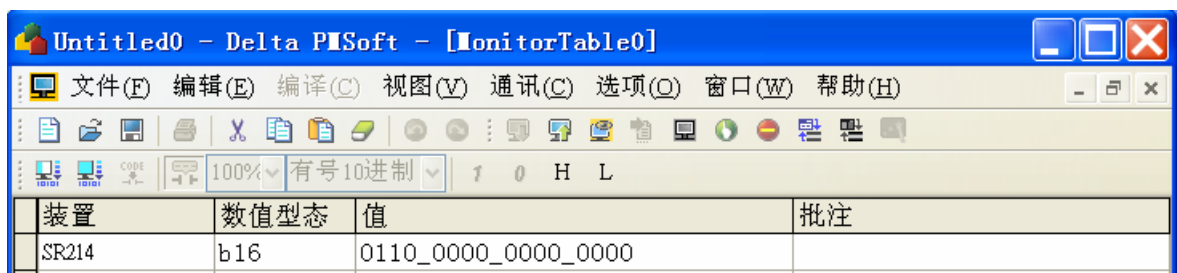
13.5.1 备份

在程序备份的使用方式如下：

1. 用户利用 PMSoft 监控 AH500 运动定位模块，并且在监控状态下停止运行，或设定 SM072 为 0 停止 0100。



2. 利用监控表或外部通讯直接设定 SR214。
 - ◆ 设定 SR214 的 bit12 到 bit0，编号大小为备份文件名称。
 - ◆ 设定 SR214 的 bit13 为 1 表示作备份动作。
 - ◆ 设定 SR214 的 bit14 为 1 表示此次的备份对象为程序

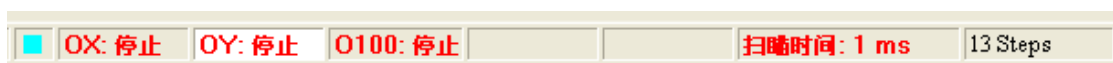


在程序备份完成后，会在存储卡中的\AHMotion\Program\产生一个 raw 文件，并且在文件中包含此备份程序的密码，以上图的例子此 raw 文件的名称为 0.raw。

13.5.2 回存

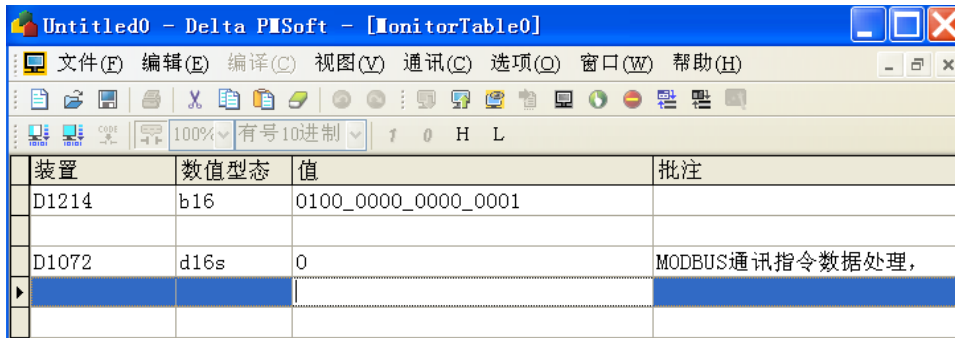
在 AH500 运动定位模块，对于程序的回存上，当上电时会自动检查存储卡中的\AHMotion\Program\是否存在 0.raw 文件，如果存在将自动把此 raw 文件回存至运动模块中。另外，可利用以下使用方式强制回存盘案：

1. 用户利用 PMSoft 监控 AH500 运动定位模块，并且在监控状态下停止运行，或设定 SM072 为 0 停止 0100。



2. 利用监控表或外部通讯直接设定 SR214。

- ◆ 设定 SR214 的 bit12 到 bit0，编号大小为备份文件名称。
- ◆ 设定 SR214 的 bit13 为 0 表示作回存动作。
- ◆ 设定 SR214 的 bit14 为 1 表示此次的备份对象为程序。



以上图为例表示将 1.raw 回存到运动模块中。

所有回存动作均会把程序的密码作回存，并且在回存过程中 AH500 运动定位模块的 run 灯会呈现闪烁状态，当闪烁结束表示回存动作完成。

13

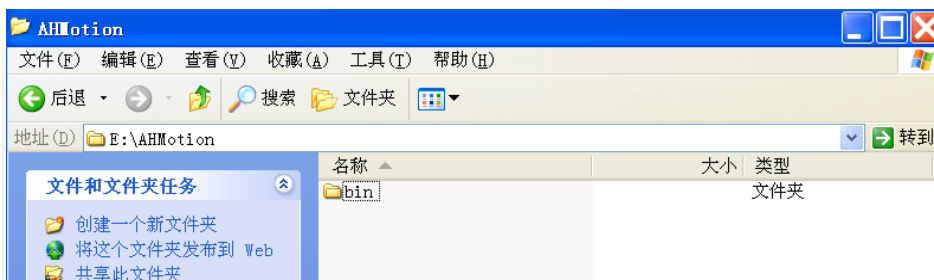
13.6 韧体更新

存储卡可做为 AH500 运动定位模块的韧体更新管道。韧体更新操作方式如下：

1. 先将更新启动开关切换至启动韧体更新模式。



2. 在存储卡的根目录下建立以下数据夹 AHMotion\bin。



3. 根据 AH500 运动定位模块在数据夹 AHMotion\bin 下保存不同的 bin 档。
 - i. AH20MC-5A 机种在数据夹下保存：C5A20MC.bin 以及 G5A20MC.bin
 - ii. AH10PM-5A 机种在数据夹下保存：C5A10PM.bin 以及 G5A10PM.bin
 - iii. AH15PM-5A 机种在数据夹下保存：C5A15PM.bin 以及 G5A15PM.bin
 - iv. AH05PM-5A 机种在数据夹下储存：C5A05PM.bin 以及 G5A05PM.bin

4. 将存储卡插入存储卡插槽中，并且将模块重新上电，当 RUN 灯闪烁时表示 FW 正在更新，更新完成后 RUN 灯常亮，若在更新过程中发生错误则 ERROR 会亮起。
5. 更新完成后，将更新启动开关扳回原来方向。



完成后需重新上电，再次启动模块。

MEMO

13

第14章 DMCNET

目录

14.1	功能简介	14-2
14.2	功能规格	14-2
14.3	参数功能	14-3
14.4	DMCNET 连线说明	14-8
14.5	伺服参数读写说明	14-11
14.6	DMCNET 运动控制说明	14-13
14.7	DMCNET 连线范例	14-18
14.7.1	增量型伺服连线范例	14-18
14.7.2	绝对型伺服连线范例	14-19
14.8	DMCNET 错误说明及排除方式	14-20

14.1 功能简介

AH20MC-5A 模块备有 DMCNET 通讯端口，DMCNET 为台达专用运动控制通讯方式，是一种实时系统 (Real Time System) 的功能，更新 12 轴的命令只需一毫秒 (1ms)、内建多种的回原点模式、12 轴可以同动，亦可分配为 4 组的 3 轴螺旋或直线插补，或分配为 6 组的 2 轴直线或圆弧插补：

- ◆ 支持 12 轴伺服台达 ASDA-A2 F 机种
- ◆ 支持可利用 DMCNET 对伺服做内部参数读写功能
- ◆ 支持 DMCNET 原点回归以及同步运动功能

14.2 功能规格

● 连接器

项目	规格
传输方式	DMCNET
电气隔离	500VDC
接头	可插拔式连接器 (5.08mm)
传输电缆	四条通讯线

● 通讯

项目	规格
数据类型	Static frame、Dynamic frame
串行传输速度	两通道每通道 10M (位/秒)
最大传输距离	20 公尺 (需 120 欧姆终端电阻)

14.3 参数功能

■ 参数列表

参数名称	轴 1	轴 2	轴 3	轴 4	轴 5	轴 6	轴 7	轴 8
DMCNET 伺服控制命令	SR1072	SR1172	SR1272	SR1372	SR1472	SR1572	SR1672	SR1772
DMCNET 伺服状态	SR1073	SR1173	SR1273	SR1373	SR1473	SR1573	SR1673	SR1773
DMCNET 伺服错误码*1	SR1074	SR1174	SR1274	SR1374	SR1474	SR1574	SR1674	SR1774
DMCNET 伺服错误码*2	SR1075	SR1175	SR1275	SR1375	SR1475	SR1575	SR1675	SR1775
DMCNET 伺服参数读写 Data*1	SR1076	SR1176	SR1276	SR1376	SR1476	SR1576	SR1676	SR1776
DMCNET 伺服参数读写 Data*2	SR1077	SR1177	SR1277	SR1377	SR1477	SR1577	SR1677	SR1777
DMCNET 伺服参数位置	SR1078	SR1178	SR1278	SR1378	SR1478	SR1578	SR1678	SR1778
DMCNET 回原点模式方式								
参数名称	轴 9	轴 10	轴 11	轴 12	轴 13	轴 14	轴 15	轴 16
DMCNET 伺服控制命令	SR1872	SR1972	SR2072	SR2172	SR2272	SR2372	SR2472	SR2572
DMCNET 伺服状态	SR1873	SR1973	SR2073	SR2173	SR2273	SR2373	SR2473	SR2573
DMCNET 伺服错误码*1	SR1874	SR1974	SR2074	SR2174	SR2274	SR2374	SR2474	SR2574
DMCNET 伺服错误码*2	SR1875	SR1975	SR2075	SR2175	SR2275	SR2375	SR2475	SR2575
DMCNET 伺服参数读写 Data*1	SR1876	SR1976	SR2076	SR2176	SR2276	SR2376	SR2476	SR2576
DMCNET 伺服参数读写 Data*2	SR1877	SR1977	SR2077	SR2177	SR2277	SR2377	SR2477	SR2577
DMCNET 伺服参数位置	SR1878	SR1978	SR2078	SR2178	SR2278	SR2378	SR2478	SR2578
DMCNET 回原点模式方式								

*1. 该参数的 Low word。

*2. 该参数的高 word。

■ 参数介绍

- ◆ SR1072、SR1172、SR1272、SR1372、SR1472、SR1572、SR1672、SR1772、SR1872、SR1972、SR2072、SR2172、SR2272、SR2372、SR2472、SR2572：DMCNET 控制命令。

[说明]

第 1 轴		第 2 轴		第 3 轴		第 4 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1072	-	SR1172	-	SR1272	-	SR1372
第 5 轴		第 6 轴		第 7 轴		第 8 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1472	-	SR1572	-	SR1672	-	SR1772
第 9 轴		第 10 轴		第 11 轴		第 12 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1872	-	SR1972	-	SR2072	-	SR2172
第 13 轴		第 14 轴		第 15 轴		第 16 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR2272	-	SR2372	-	SR2472	-	SR2572

寄存器的使用定义如下：

数值 (HEX)	DMCNET 控制命令	数值 (HEX)	DMCNET 控制命令
xxx1	Servo OFF	x1xx	16-bit 的伺服参数写入
xxx2	Servo ON	x2xx	32-bit 的伺服参数写入
xxx3	NMT Reset	x3xx	读出伺服参数
xxx4	Error Reset	x4xx	DMCNET 的回原点模式
		x5xx	DMCNET 的 DMC 运动模式

- Servo OFF/ON 的功能为设定伺服启动或是关闭，设定完毕后，可以在 DMCNET Servo Status 中反映目前伺服的状态。
- NMT Reset 为设定伺服做 DMCNET 通讯重置动作，此动作不论伺服是否已连线均可以操作。
- Error Reset 为伺服发生错误后，做错误清除的动作，此时伺服必须为已连线才有效。
- 16bit/32bit 的伺服参数写入，为写入伺服参数时的动作，16-bit 或 32-bit 则要看伺服对于该参数的定义大小。读出伺服参数动作则不需要考虑该参数的定义大小。
- DMCNET 的模式设定，目前支持回原点模式以及 DMC 运动模式。
 - ◆ 回原点模式为利用 DMCNET 的原点回归，需搭配设定回原点模式的方式。
 - ◆ 设定 DMC 运动模式后，对于运动轴的操作需利用单轴运动控制方式，来对伺服做控制。

- ◆ SR1073、SR1173、SR1273、SR1373、SR1473、SR1573、SR1673、SR1773、SR1873、SR1973、SR2073、SR2173、SR2273、SR2373、SR2473、SR2573：DMCNET 伺服状态。

[说明]

第 1 轴		第 2 轴		第 3 轴		第 4 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1073	-	SR1173	-	SR1273	-	SR1373
第 5 轴		第 6 轴		第 7 轴		第 8 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1473	-	SR1573	-	SR1673	-	SR1773
第 9 轴		第 10 轴		第 11 轴		第 12 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1873	-	SR1973	-	SR2073	-	SR2173
第 13 轴		第 14 轴		第 15 轴		第 16 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR2273	-	SR2373	-	SR2473	-	SR2573

寄存器的状态定义如下：

数值 (HEX)	DMCNET 伺服状态	数值 (HEX)	DMCNET 伺服状态
xxx0	disconnected	x16#x	Control Cmd Done
xxx1	Servo OFF	x1xx	Control Cmd Run
xxx2	Servo ON	x2xx	Control Cmd Error
xx1x	Servo Error		

- 伺服状态 disconnected 表示目前伺服尚未连线，硬件上若已连线但状态为 disconnected 时，应是目前模块无法确认伺服的通讯状态所致，可以下列动作重新设定：
 1. 此时需对伺服下达 **NMT Reset** 动作。
 2. 待伺服 **DMCNET** 网络功能重置完成后，伺服的状态将会为 **Servo OFF**。
- 伺服 Servo OFF/ON 的状态会随着控制命令下达完成后，而修正为 Servo OFF/ON 的状态。
- Servo Error 为伺服在运转过程中发生错误的标志 (Bit4)，清除的方式为在命令中下达 Error Reset 的动作，完成后会清除此错误标志。
- Control Cmd Done/Run/Error，为显示 SR1072 (SR1172、SR1272、SR1372...) 当下达 Bit11-8 的控制命令后，AH20MC-5A 对此命令运行的状态，用户可利用此状态来判读目前运行的结果。

- ◆ SR1075、SR1074、SR1175、SR1174、SR1275、SR1274、SR1375、SR1374、SR1475、SR1474、SR1575、SR1574、SR1675、SR1674、SR1775、SR1774、SR1875、SR1874、SR1975、SR1974、SR2075、SR2074、SR2175、SR2174、SR2275、SR2274、SR2375、SR2374、SR2475、SR2474、SR2575、SR2574：DMCNET 错误码。

[说明]

第 1 轴		第 2 轴		第 3 轴		第 4 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR1075	SR1074	SR1175	SR1174	SR1275	SR1274	SR1375	SR1374
第 5 轴		第 6 轴		第 7 轴		第 8 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR1475	SR1474	SR1575	SR1574	SR1675	SR1674	SR1775	SR1774
第 9 轴		第 10 轴		第 11 轴		第 12 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR1875	SR1874	SR1975	SR1974	SR2075	SR2074	SR2175	SR2174
第 13 轴		第 14 轴		第 15 轴		第 16 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR2275	SR2274	SR2375	SR2374	SR2475	SR2474	SR2575	SR2574

错误码为伺服的状态发生 Servo Error 后，伺服回传的错误码。其错误码的定义，可参照伺服的操作手册中，错误码定义以及故障排除来做为排除的方式。当错误标志清除后，错误码会自动清除为零。

- ◆ SR1077、SR1076、SR1177、SR1176、SR1277、SR1276、SR1377、SR1376、SR1477、SR1476、SR1577、SR1576、SR1677、SR1676、SR1777、SR1776、SR1877、SR1876、SR1977、SR1976、SR2077、SR2076、SR2177、SR2176、SR2277、SR2276、SR2377、SR2376、SR2477、SR2476、SR2577、SR2576：DMCNET 伺服参数读写数据。

[说明]

第 1 轴		第 2 轴		第 3 轴		第 4 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR1077	SR1076	SR1177	SR1176	SR1277	SR1276	SR1377	SR1376
第 5 轴		第 6 轴		第 7 轴		第 8 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR1477	SR1476	SR1577	SR1576	SR1677	SR1676	SR1777	SR1776
第 9 轴		第 10 轴		第 11 轴		第 12 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR1877	SR1876	SR1977	SR1976	SR2077	SR2076	SR2177	SR2176
第 13 轴		第 14 轴		第 15 轴		第 16 轴	
HW	LW	HW	LW	HW	LW	HW	LW
SR2277	SR2276	SR2377	SR2376	SR2477	SR2476	SR2577	SR2576

伺服参数读写数据：利用 DMCNET 对伺服做参数读写时，写入数据或为读出的数据，如果读写过程中发生错误，则会在此寄存器中存放错误代码。

- ◆ SR1078 \SR1178 \SR1278 \SR1378 \SR1478 \SR1578 \SR1678 \SR1778 \SR1878 \SR1978 \SR2078 \SR2178 \SR2278 \SR2378 \SR2478 \SR2578 :DMCNET 伺服参数读写位置 :DMCNET 回原点模式方式设定。

[说明]

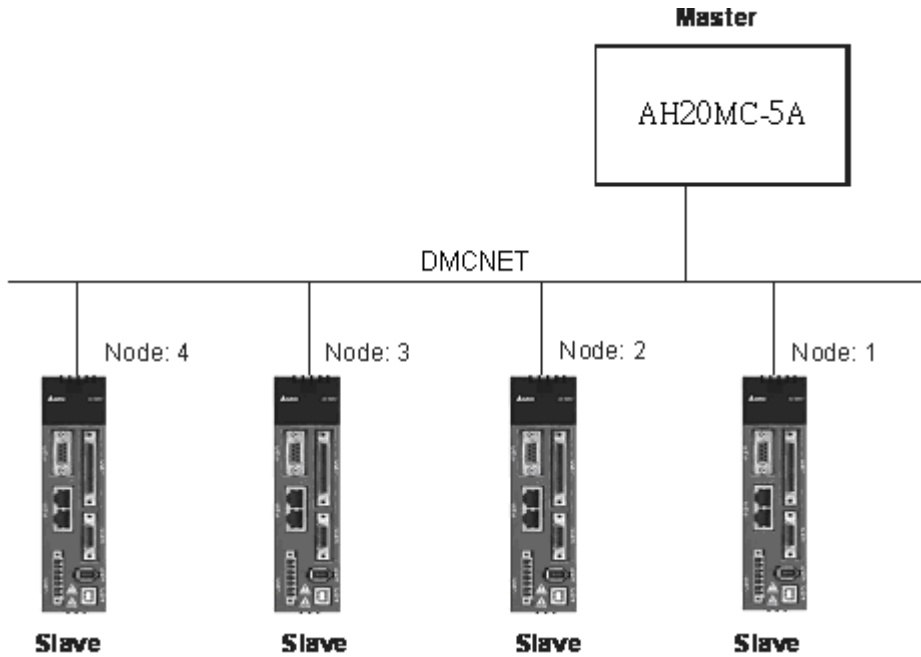
第 1 轴		第 2 轴		第 3 轴		第 4 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1078	-	SR1178	-	SR1278	-	SR1378
第 5 轴		第 6 轴		第 7 轴		第 8 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1478	-	SR1578	-	SR1678	-	SR1778
第 9 轴		第 10 轴		第 11 轴		第 12 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR1878	-	SR1978	-	SR2078	-	SR2178
第 13 轴		第 14 轴		第 15 轴		第 16 轴	
HW	LW	HW	LW	HW	LW	HW	LW
-	SR2278	-	SR2378	-	SR2478	-	SR2578

- 设定伺服读写参数功能时，设定参数的位置，其中 high byte 为伺服的群组字符，low byte 为伺服的参数字节。以 P1-44 为例，设定时 high byte 为 1 而 low byte 为 44，也就是 16#012C。
- 在伺服设定为回原点模式时，此参数为设定目前伺服的回原点方式，可设定值为 1~35。其各项回原点模式，可参考第 14.6 章。

14

14.4 DMCNET连线说明

■ 连线设定



● 硬件配置

◆ 硬件配线

DMCNET 的硬件接在线，需利用台达的 DMCNET 缆线，并且在整个系统最后加上台达 DMCNET 的终端电阻 ASD-TR-DM0008 (如下图所示)。整个系统才可以稳定通讯，并且整个系统最大连线长度不可超过 30 公尺。



ASD-TR-DM0008 DMCNET 终端电阻

◆ 伺服韧体版本确认

1. 利用面板设定 P0-00 查看是否为 1.744 以上版本，其中的 7 代表为 A2 伺服中的 F 型式机种，若非为 7 请更换伺服驱动器。
2. 若 P0-00 = 1.744，再查看 P5-00 是否为 873 以上版本。

◆ ASDA-A2 伺服设定

在建立 DMCNET 连线之前，首先设定伺服模式为 DMCNET 模式，其操作步骤如下：

1. 设定 A2 P1-01 为 16#0B (此设定为将模式设定为 DMCNET 模式)。
2. 设定 A2 P3-00，其数值代表节点编号，数值范围为 16#01~16#0C，驱动器的节点 ID 分别为 1 到 12，彼此间不能重复使用，其中节点 ID 为 1 代表第 1 轴，为 2 代表第 2 轴，为 3~12 则分别代表第 3~12 轴。需注意整个 DMCNET 网络上，一定需要存在一台节点 ID 为 1 的 A2 伺服，整体网络才能正常通讯，例如两台 A2，其中必须有一台节点 ID 为 1，另外一台节点 ID 可为 2~12。全网络只存在一台 A2 时，只能设定此 A2 节点 ID 为 1，其余设定 DMCNET 网络均会失效。
3. 设定 P3-01，设定通讯模式为 16#0203 代表台达 DMCNET 模式。
4. 设定 P0-02 = 16#120，检查连线状态。

● 连线状态检查

伺服是否已连线可利用 SR1073 (SR1173, SR1273...) 中 Bit0~3 来确认。当状态为 disconnected 表示模块并未搜寻到该伺服，此时可以下列步骤重新设定：

1. 利用 SR1072 (SR1172, SR1272...) 中的 Bit0~3 对该伺服下达网络重置命令。
2. 伺服面板会显示 OO-OO，表示伺服正在做通讯重置动作。
3. 当通讯重置完成后，可由 SR1073 (SR1173, SR1273...) 中 Bit0-3 来确认目前伺服是否为 Servo OFF 状态。

可透过设定 P0-02 = 16#120 查看伺服连线的状态。设定完成后查看伺服驱动器面板显示值，显示值说明如下表所示：

面板显示值	说明
16#06	伺服端 DMCNET 等待连线
16#80	伺服端 DMCNET 连线成功
16#111	上位机与伺服进入运行模式。

● 伺服启动/关闭

◆ 直接操作特殊寄存器 SR

伺服的连线状态为 Servo OFF 状态下，可对此伺服做启动与关闭的动作。动作流程如下：

1. 由 SR1072 (SR1172, SR1272...) 的 Bit 0-3 数值为 1 时表示下达伺服关闭，Bit 0-3 数值为 2 时表示下达伺服启动。
2. 下达命令完成后可利用 SR1073 (SR1173, SR1273...) 的 Bit0-3 来确认该伺服已经为启动状态。

◆ 利用运动功能块

可透过运动功能块伺服启动/停止来达成

En	T_DMCPowerUp	Eno
Axis		Valid
Enable		Busy
		Error

运动功能块输入以及输出引脚详细说明请参照第五章

14.5 伺服参数读写说明

伺服的参数可透过 DMCNET 方式，利用模块来修改或读取已连线伺服的参数，一次只能对一台伺服做设定，在设定完成后，才可利用同样的方式设定其余伺服。在做此动作前，需要先利用 SR1073 (SR1173·SR1273...) 的 Bit0-3 确认该伺服已经连线，才能利用其它的寄存器做读写参数的设定。

- 伺服参数写入设定步骤：

- ◆ 直接操作特殊寄存器 SR

1. 确认目前伺服的 ID，ID 为 1 时，则对第 0 轴操作，2 则对第 2 轴操作，若为 3 至 12 则为对第 3 至第 12 轴操作，每一次操作只能挑选一轴伺服。
2. 利用 SR1073 (SR1173·SR1273...) 的 Bit0-3 确认该伺服状态为 Servo OFF 或 Servo ON。
3. 参考伺服的参数说明，可得知该参数的群组字符和参数字元，并且得知该参数的数据大小，以及此参数是否在 Servo ON 能设定。以 P1-44 为例，群组字符为 1 而参数字元为 44 同时数据大小为 32bit。
4. 如伺服参数需在 Servo OFF 才可设定，将该伺服状态切换至 Servo OFF，若无限制可跳过此步骤。
5. 对 SR1078 (SR1178·SR1278...) 做写入参数的位置设定，以 P1-44 为例，群组字符为 1 则 High byte 为 16#01，参数字元为 44 则 Low byte 为 16#2C，故在 SR1078 (SR1178·SR1278...) 写入 16#012C。
6. 对 SR1077 (SR1177·SR1277...)、SR1076 (SR1176·SR1276...) 做写入参数的内容设定，可在此填入欲写入的数值。
7. 对 SR1072 (SR1172·SR1272...) 的 Bit11-8，下达写入参数的命令，其中为 1 时，代表写入数值的大小为 16bit，为 2 表示写入数值的大小为 32bit。以 P1-44 为例，查到写入数据大小为 32bit，故在此将 Bit11-8 填入 2。
8. 利用 SR1073 (SR1173·SR1273...) 的 Bit11-8，检查此次的参数写入动作是否正确，若发生错误，则在 SR1077 (SR1177·SR1277...)、SR1076 (SR1176·SR1276...) 可以查到错误码，若成功完成，则 SR1073 (SR1173·SR1273...) 的 Bit11-8 会显示 0。
9. 当欲设定的伺服已完成或显示发生错误后，才可对其他伺服做写入或是读出参数的动作。

- ◆ 利用运动功能块

可透过运动功能块伺服参数写入来达成伺服参数写入动作

En	T_DMCServoWrite	Eno
Axis		Done
Execute		Busy
Group		Error
Parameter		
Value		
DataType		

运动功能块输入以及输出引脚详细说明请参照第五章

● 伺服参数读出设定步骤：

◆ 直接操作特殊寄存器 SR

1. 确认目前伺服的 ID，ID 为 1 时，则对第 1 轴操作，2 则对第 2 轴操作，若为 3 至 12，则为对第 3 至第 12 轴操作，每一次操作只能挑选一轴伺服。
2. 参考伺服的参数说明，可得知该参数的群组字符和参数字元。以 P1-44 为例，群组字符为 1 而参数字元为 44。
3. 对 SR1078 (SR1178 · SR1278...) 做读出参数的位置设定，以 P1-44 为例，群组字符为 1 则 High byte 为 16#01，参数字元为 44 则 Low byte 为 16#2C，故在 SR1078 (SR1178 · SR1278...) 写入 16#012C。
4. 对 SR1072 (SR1172 · SR1272...) 的 Bit11-8，填入 3 下达读出参数的命令。
5. 利用 SR1073 (SR1173 · SR1273...) 的 Bit11-8，检查此次的参数读出动作是否正确，若发生错误，则在 SR1077 (SR1177 · SR1277...)、SR1076 (SR1176 · SR1276...) 可以查到错误码，若成功完成，则 SR1073 (SR1173 · SR1273...) 的 Bit11-8 会显示 0。
6. 若成功读出后，可利用 SR1077 (SR1177 · SR1277...)、SR1076 (SR1176 · SR1276...) 可知读出伺服传送回来的参数内容。
7. 当欲设定的伺服已完成或显示发生错误后，才可对其他伺服做写入或是读出参数的动作。

◆ 利用运动功能块

可透过运动功能块伺服参数读取来达成伺服参数读取动作

En	T_DMCServoRead	Eno
Axis		Done
Execute		Busy
Group		Error
Parameter		Value

运动功能块输入以及输出引脚详细说明请参照第五章

14.6 DMCNET运动控制说明

■ 回原点模式

◆ 直接操作特殊寄存器 SR

在执行 DMCNET 的回原点模式的状况下，先确认

1. 连线状态 SR1073 (SR1173 · SR1273...) Bit0-3 的数值为 2，表示此伺服已处于启动状态下。
2. 由于在 DMCNET 回原点状况下，由伺服决定遇到左右极限以及 DOG 的运作方式，故需要利用伺服的 CN1 I/O 将左右极限点及 DOG 点做连接和功能设定，其连接及设定方式请参考 ASDA-A2 的 3.4.2 CN1 I/O 连接器信号说明。

上述的环境设定已确认后，可开始做回原点模式的设定方式如下：

1. 对做原点回归的伺服设定 SR1072 (SR1172 · SR1272...) Bit11-8 为 4，表示设定此伺服为 DMCNET 回原点模式。
2. 对 SR1078 (SR1178 · SR1278...) 设定伺服回归原点方式，设定完成后伺服开始运作原点回归。
3. 当伺服原点回归动作运作中时 SR1073 (SR1173 · SR1273...) 的 Bit11-8 会显示 1，完成后 SR1073 (SR1173 · SR1273...) 的 Bit11-8 会显示 0 表示此伺服的原点回归完成。
4. 原点回归完成后，伺服的当前位置数值将会与轴参数中的当前位置 (SRmn33..mn32, mn=10~25)，其对应轴相同。

◆ 利用运动功能块

可透过运动功能块伺服回原点命令来达成伺服回原点动作

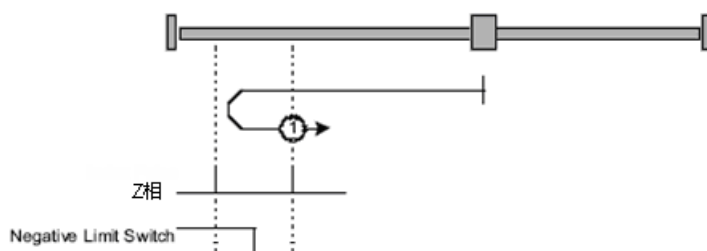
En	T_DMCServo~	Eno
Axis		Done
Execute		Busy
Mode		Aborted
Offset		Error
VRT		
VCR		

运动功能块输入以及输出引脚详细说明请参照第五章

目前 DMCNET 回原点模式提供多种不同的状态，针对各种模式叙述如下：

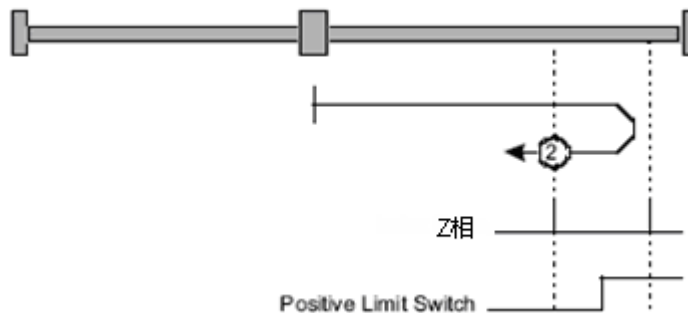
◆ 设定 SR1078 (SR1178 · SR1278...) 为 1 时：

电机将顺时针旋转到碰到反转静止极限，则电机逆时针旋转至出厂设定的 Z 相后停止。



- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 2 时 :

电机将**逆时针**旋转到碰到正转静止极限 · 则电机**顺时针**旋转至出厂出场设定的 Z 相后停止。



- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 3、4 时 :

电机将**逆时针**旋转到碰到 DOG 点时 · 当设定为 3 时电机则会顺时针旋转至出厂设定的 Z 相后停止 ; 当设定为 4 时电机则会逆时针旋转到出厂设定的 Z 相后停止 · 在过程中若先碰到静止极限时 · 电机则会产生错误码后停止。

- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 5、6 时 :

电机将**顺时针**旋转到碰到 DOG 点时 · 当设定为 5 时电机则会顺时针旋转至出厂设定的 Z 相后停止 · 当设定为 6 时电机则会逆时针旋转到出厂设定的 Z 相后停止 · 在过程中若先碰到静止极限时 · 电机则会产生错误码后停止。

- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 7、8、9、10 时 :

7、8、9、10 皆为正方向寻找 DOG · 将回原点分为以下三种情况 :

a. 非在 DOG 或 LSP 上时

模式 7 : 遇到 DOG 上升沿后 · 反方向脱离 DOG 寻找 Z 相后停止

模式 8 : 遇到 DOG 上升沿后 · 寻找 Z 相后停止

模式 9 : 遇到 DOG 下降沿后 · 反方向寻找 Z 相后停止

模式 10 : 遇到 DOG 下降沿后 · 脱离 DOG 寻找 Z 相后停止

b. 在 DOG 上

模式 7 : 反方向脱离 DOG 寻找 Z 相后停止

模式 8 : 反方向脱离 DOG 后 · 正方向寻找 DOG 上升沿后 · 寻找 Z 相后停止

模式 9 : 遇到 DOG 下降沿后 · 反方向寻找 Z 相后停止

模式 10 : 遇到 DOG 下降沿后 · 脱离 DOG 寻找 Z 相后停止

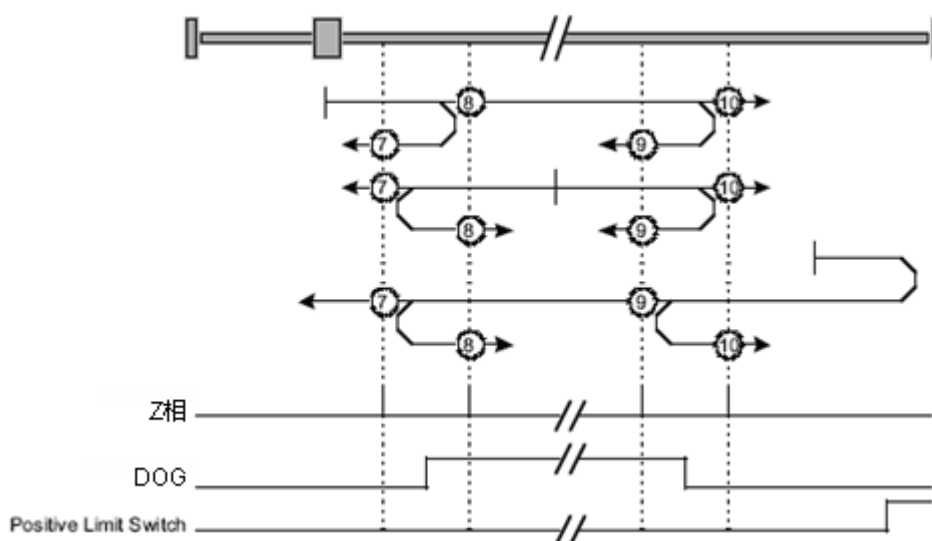
c. 遇到 LSP 后

模式 7 : 反方向脱离 DOG 寻找 Z 相后停止

模式 8 : 反方向脱离 DOG 后 · 正方向寻找 DOG 上升沿后 · 寻找 Z 相后停止

模式 9 : 反向遇到 DOG 上升沿后 · 寻找 Z 相后停止

模式 10 : 反向遇到 DOG 上升沿后 · 正方向脱离 DOG 寻找 Z 相后停止



◆ 设定 SR1078 (SR1178 · SR1278...) 为 11、12、13、14 时：

11、12、13、14 皆为反反方向寻找 DOG，将回原点分为以下三种情况：

a. 非在 DOG 或 LSN 上时

模式 11：遇到 DOG 上升沿后，正方向脱离 DOG 寻找 Z 相后停止

模式 12：遇到 DOG 上升沿后，寻找 Z 相后停止

模式 13：遇到 DOG 下降沿后，正方向寻找 Z 相后停止

模式 14：遇到 DOG 下降沿后，脱离 DOG 寻找 Z 相后停止

b. 在 DOG 上

模式 11：正方向脱离 DOG 寻找 Z 相后停止

模式 12：正方向脱离 DOG 后，反方向寻找 DOG 上升沿后，寻找 Z 相后停止

模式 13：遇到 DOG 下降沿后，正方向寻找 Z 相后停止

模式 14：遇到 DOG 下降沿后，脱离 DOG 寻找 Z 相后停止

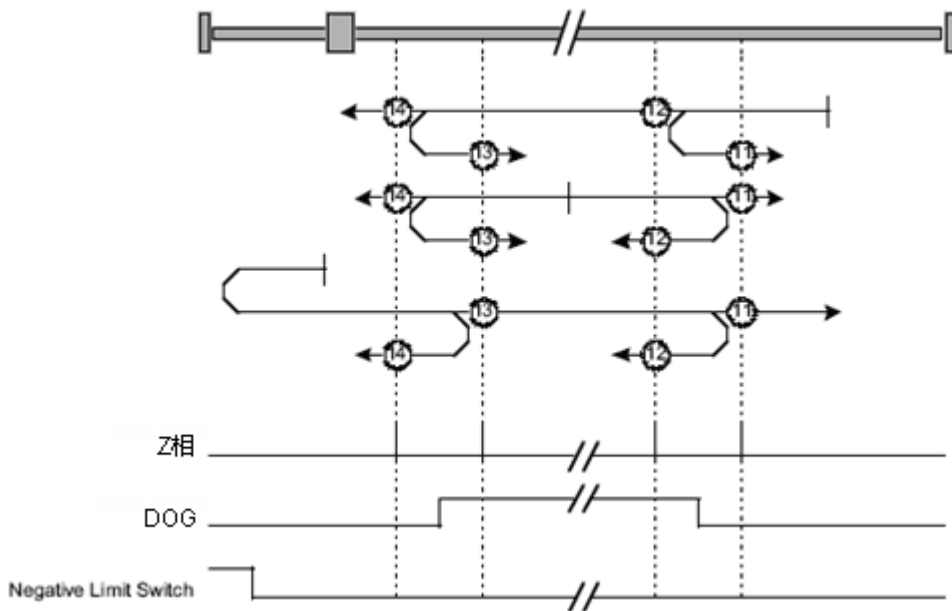
c. 遇到 LSN 后

模式 11：正方向脱离 DOG 寻找 Z 相后停止

模式 12：正方向脱离 DOG 后，反方向寻找 DOG 上升沿后，寻找 Z 相后停止

模式 13：正向遇到 DOG 上升沿后，寻找 Z 相后停止

模式 14：正向遇到 DOG 上升沿后，反方向脱离 DOG 寻找 Z 相后停止



- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 17~30 时：
电机运转方式雷同 1 to 14，但 1 to 14 过程中，电机以 Z 相为停止的原点，而 17~30 时，则以 DOG 点来取代 Z 相。
- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 33 时：
电机将顺时针旋转，碰到第一个 Z 相后停止。
- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 34 时：
电机将逆时针旋转，碰到第一个 Z 相后停止。
- ◆ 设定 SR1078 (SR1178 · SR1278...) 为 35 时：
电机将目前的位置的数值，更改为欲设定的位置值。

■ DMCNET 运动模式

AH20MC-5A DMCNET 运动模式可同时对 12 轴伺服做数据更新，且此更新周期为 1ms。只要用户将特殊寄存器设定为 DMCNET 运动模式，则对应的伺服就会被设定为 DMCNET 运动模式，12 轴间可以部分为回原点模式，部分为 DMCNET 运动模式。只要用户将伺服设定为 DMCNET 的运动模式，之后对于轴控动作的设定，均可利用单轴运动功能设定，并且支持多轴运动的命令。设定 DMCNET 运动模式方法如下：

1. 确认连线状态 SR1073 (SR1173 · SR1273...) Bit0-3 的数值为 2，表示此伺服已处于启动状态下。
2. 设定该伺服的 SR1072 (SR1172 · SR1272...) Bit11-8 的数值为 5，表示设定此伺服于 DMCNET 运动模式之下。
3. 检查 SR1073 (SR1173 · SR1273...) Bit11-8 的数值，此数值将会维持在 1，表示此伺服处于 DMCNET 运动模式运作之下。
4. 利用单轴运动功能决定该轴的运动方式，可为单段速运动、插入单段速、两段速...等等单轴运动功能，使用方式直接设定单轴运动的寄存器即可。
5. 做多轴运动时可利用 G code 做多轴之间的插补运动。
6. 切换至原点回归模式时，将 SR1072 (SR1172 · SR1272...) Bit11-8 的数值设定为 4，并且利用先前提到的原点回归使用方式即可。

各轴之间原点回归以及 DMCNET 运动模式可同时做设定，所有伺服可设定为不同的模式但设定为原点回归的伺服无法做单轴运动以及多轴 G 码动作，伺服一定要为 DMCNET 运动模式下才可做运动控制。

注意事项：当设定为 DMCNET 运动模式前，需先将伺服驱动器的当前位置与 AH20MC-5A 该轴所代表的当前位置一致 SRmn33..mn32 (mn=10~25)，此动作可透过 DMCNET 在设定 DMCNET 运动模式前，先利用 DMCCENT 回原点模式 35，将 SRmn33..mn32 写入电机当前位置。

除以上动作外，另外提供运动功能块伺服 DMCNET 通讯初始化

En	T_DMCControlInit	Eno
Axis		Done
Execute		Busy
DMC_RatioNum		Error
DMC_RatioDen		

此运动功能块，将 DMCCENT 上指定的轴号做出所有通讯上的初始化的动作，其动作包含伺服启动、将控制器与伺服的位置作同步化、伺服进入同步控制模式，此功能块的详细叙述请参照第五章

14.7 DMCNET 连线范例

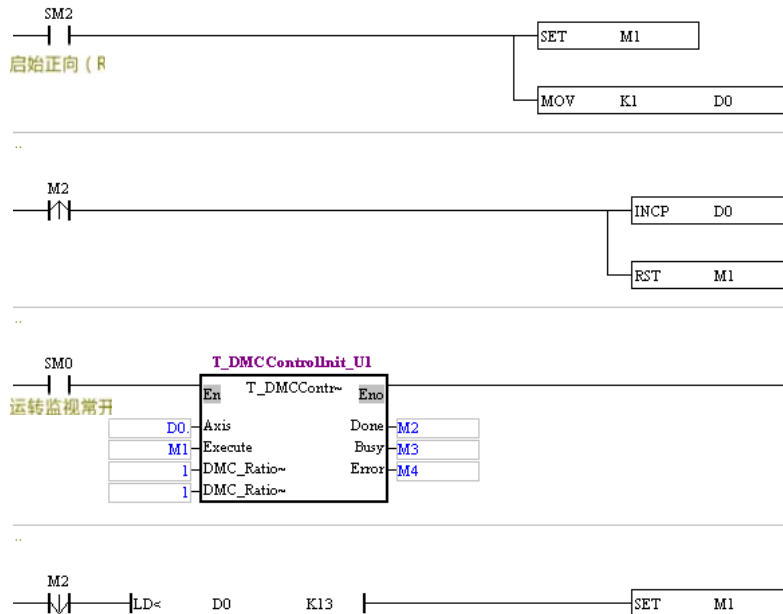
14.7.1 增量型伺服连线范例

■ 伺服设定：

- ◆ P2-08 执行 16#10 回归出厂设定
- ◆ 重新断电
- ◆ 将伺服上电
- ◆ 设定 P1-01 = 16#0B：设定为 DMCNET 模式
- ◆ 设定 P3-01 设定为 16#203：此为台达 DMCNET 专用通讯速率
- ◆ 设定 P2-15= 16#0000：删除逆向运转停止极限（用户可自行设定）
- ◆ 设定 P2-16= 16#0000：删除逆向运转停止极限（用户可自行设定）
- ◆ 设定 P2-17 = 16#0000：删除紧急停止（用户可自行设定）
- ◆ 设定 P3-00 设定伺服站号，可设置范围 16#01~16#0C。
- * 在一组 DMCNET 网络中，必需有其中 1 轴的伺服站号为 16#01。
- ◆ 伺服断电，重新上电。
- ◆ 设定 P0-02 = 16#120 查看面板显示值显示 16#80 表示硬件连线成功；若显示 16#06 请确认伺服中是否有站号设定为 1。

■ 程序编写：

在 20MC 程序中写入 T_DMCControllnit，如下图范例。



此范例当 PLC 开始执行后便会由站号 1 开始执行初始化动作，当初始化完成后可查看面板显示应为 16#111 表示进入可执行运行状态且伺服会为 SERVO ON 的状态。

* 在使用 DMCControllnit 功能块时需注意，一次只能针对一个站号进行初始化。

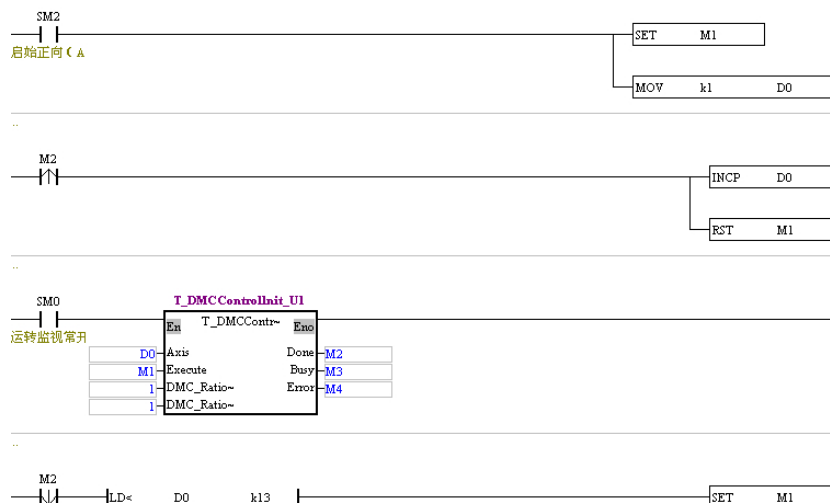
14.7.2 绝对型伺服连线范例

■ 伺服设定：

- ◆ P2-08 执行 16#10 回归出厂设定
- ◆ 重新断电
- ◆ 将伺服上电
- ◆ 设定 P1-01 = 16#0B：设定为 DMCNET 模式
- ◆ 设定 P3-01 设定为 16#203：此为台达 DMCNET 专用鲍率
- ◆ 设定 P2-69 = 16#0001：设定为绝对型编码器
- ◆ 设定 P2-15= 16#0000：删除逆向运转停止极限(用户可自行设定)
- ◆ 设定 P2-16= 16#0000：删除正向运转停止极限(用户可自行设定)
- ◆ 设定 P2-17 = 16#0000：删除紧急停止(用户可自行设定)
- ◆ 设定 P3-00 设定伺服站号，可设置范围 16#01~16#0C
 - * 在一组 DMCNET 网络中，必需有其中 1 轴的伺服站号为 16#01
- ◆ 设定 P3-12 = 16#100：启动伺服参数储存功能
- ◆ 伺服断电，重新上电。
- ◆ 设定 P0-02 = 16#120 查看面版显示值显示 16#80 表示硬件连线成功
若显示 16#06 请确认伺服中是否有站号设定为 1

■ 程序编写：

在 20MC 程序中写入 T_DMCControllnit，如下图范例。



此范例当 PLC 开始执行后会由站号 1 开始执行初始化动作，当初始化完成后可查看面板显示应为 16#111 表示进入可执行运行状态但伺服会为 SERVO ON 的状态。

* 在使用 DMCControllnit 功能块时需注意，一次只能针对一个站号进行初始化。

14.8 DMCNET错误说明及排除方式

错误状况	处理方法
DMCNET 灯号未亮起	检查整体网络线是否有发生断线的状况，并且终端电阻有正确连接。
连线状态缓存器无显示连线	利用的 DMCNET 网络重置动作，做连线重置动作。如重置动作后也无反应时，需检查硬件的接线是否确实连接，以及终端电阻是否有连
设定伺服参数错误	检查欲设定的伺服参数对于设定时伺服的启动或关闭状态的要求，或设定对象的数据型态是否符合规定。
设定伺服参数后回传逾时	检查欲设定的伺服是否可以正确连线
伺服 P0-02 = 16#120 时，伺服驱动面板显示 16#06	<ol style="list-style-type: none"> 1. 检查 DMCNET 所有伺服中是否有设定其中一轴的节点编号为 1 2. 确认硬件线路是否有松脱的情况

第15章 PMSoft软件-USB通讯设置

目录

15.1	功能简介	15-2
15.2	功能规格	15-2
15.3	PMSoft 通讯功能说明	15-2

15.1 功能简介

AH500 运动定位模块备有 Mini USB 通讯端口，用来与 PC 端软件 PMSoft 做通讯，PMSoft 可利用此通讯端口做以下操作：

- ◆ 程序上下载
- ◆ 实时监控，修改缓存器的数值

15.2 功能规格

- 通讯

项目	规格
数据型态	MODBUS ASCII
串行传输速度	9600、19200、38400、57600、115200 bit/s
最大传输距离	5 公尺

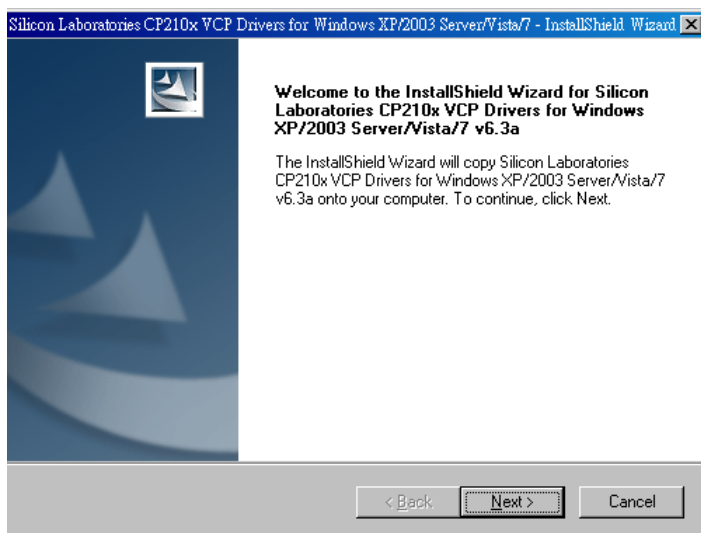
15.3 PMSoft通讯功能说明

- 联机环境设置

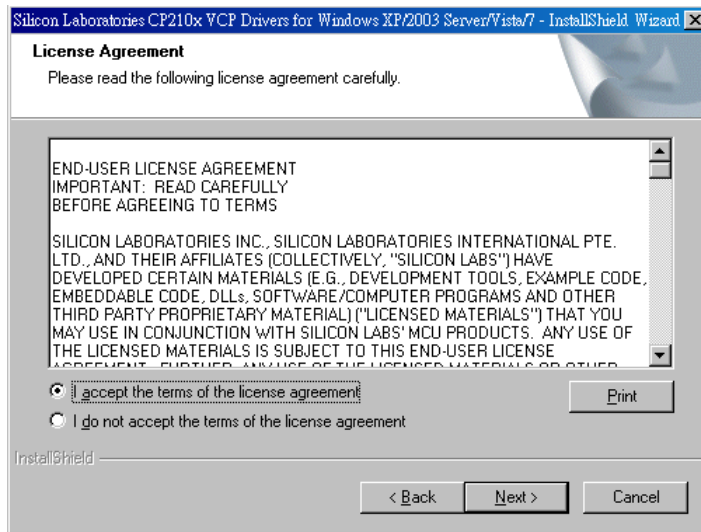
在使用 AH500 运动定位模块的 Mini USB 通讯端口时，需先在 PC 端安装驱动程序，可至 PMSoft 软件安装路径下（Delta Industrial Automation\PMSoft

2.08\drivers\CP210x_VCP_Win_XP_S2K3_Vista_7）安装驱动程序。若用户已安装过 CP210X 芯片的驱动程序，可不需再次安装，安装方法为：

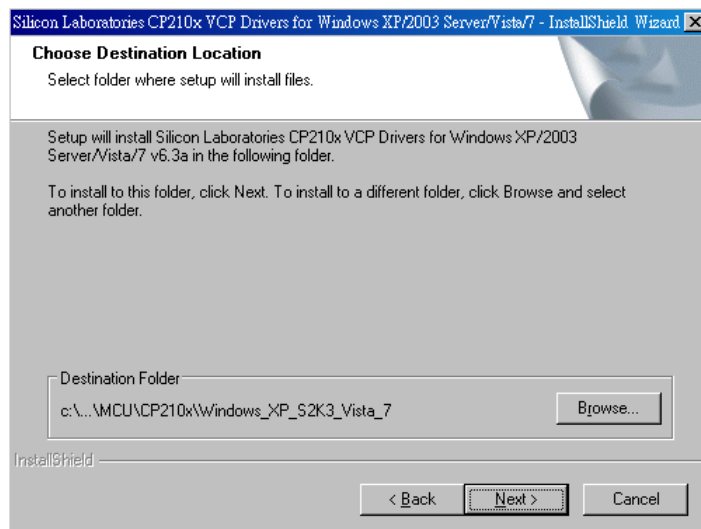
1. 点选驱动程序安装档。
2. 跳出以下页面，点击 Next。



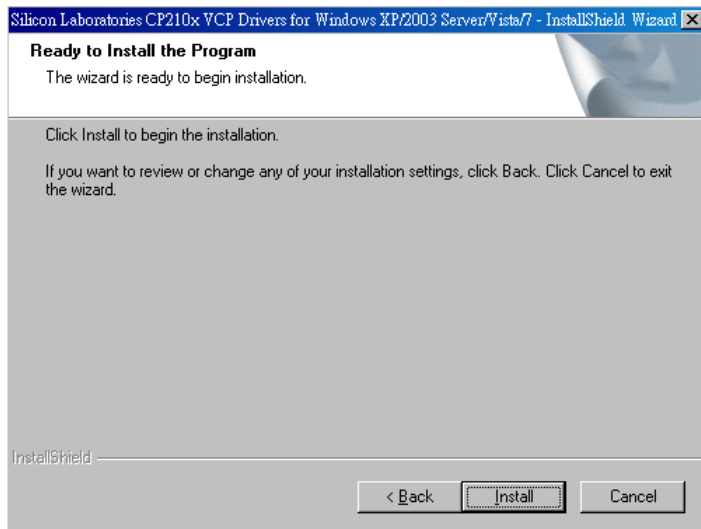
3. 跳出以下页面，选择「I accept the terms of the license agreement」，并且点击 Next。



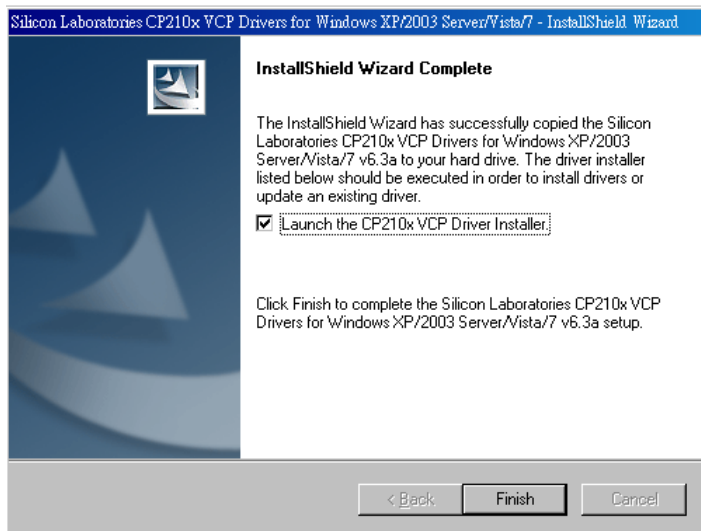
4. 跳出以下页面，可选择安装的路径，若不修改可直接点击 Next。



5. 跳出以下页面，点击 **Install**，即开始安装驱动程序。

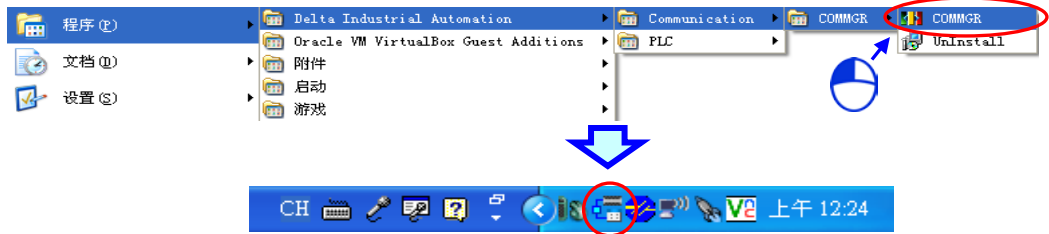


6. 安装完成后会跳出以下画面，点击 **Finish** 完成安装流程。



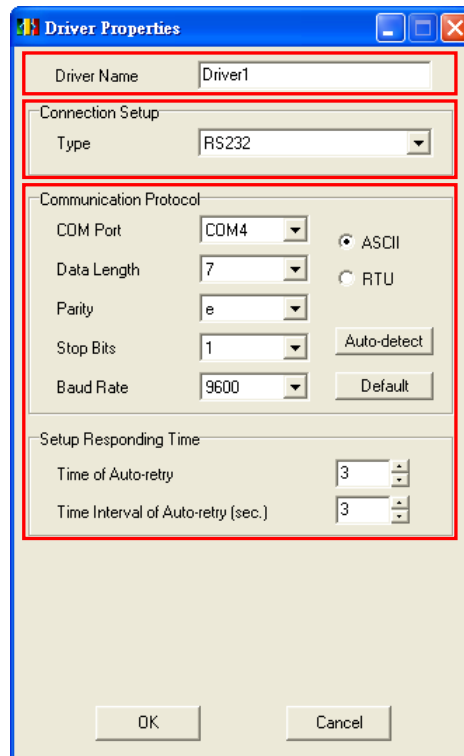
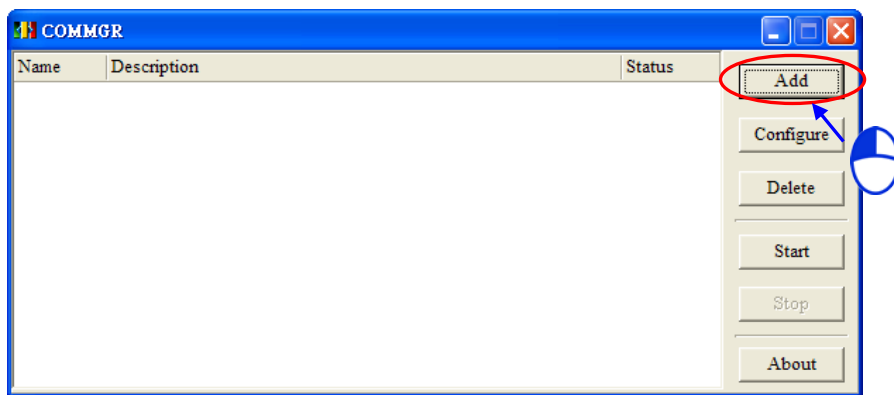
● PMSoft 联机设置

1. 启动通讯管理员 COMMGR。如在 Windows 开机后系统列中没有出现该图示时，请点选程序集>Delta Industrial Automation>Communication>COMMGR

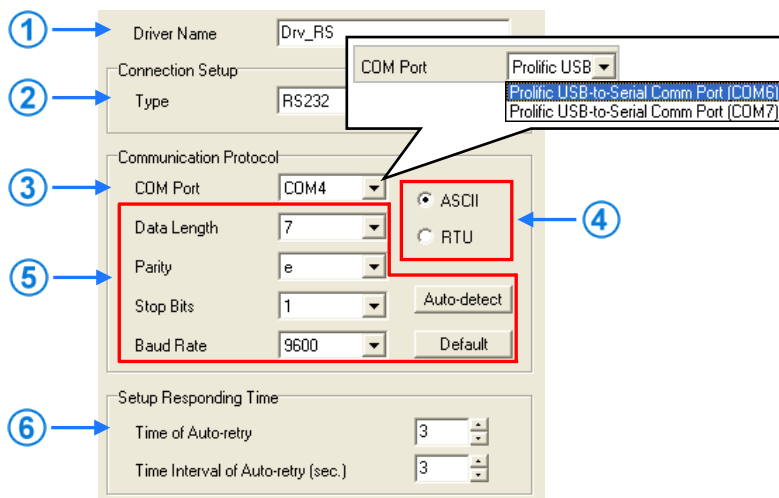


2. 在 COMMGR 中建立 Driver

于 COMMGR 右侧的 Add 键上点击鼠标左键即可开启通讯参数的设置窗口。



◆ RS-232 Driver 参数设置



- ① 于 **Driver Name** 字段输入此 **Driver** 的识别名称。
- ② 于 **Connection Setup** 的 **Type** 字段下拉选择 **RS-232** 的联机型式。
- ③ 于 **COM Port** 字段下拉选择欲建立联机通道的 **RS-232** 通讯端口口，而下拉选单中将会显示完整的 **COM Port** 编号与装置名称，显示的内容则会与 **Windows** 装置管理员中的内容一致。
- ④ 选择通讯的数据格式为 **ASCII** 或 **RTU**。
- ⑤ 设置该端口口的通讯协议，且所设置的通讯协议并须与连接的硬件装置一致，而按下 **Default** 键则可将通讯协议参数恢复为系统默认值。

若不确定硬件装置的通讯协议时，可先使用 **RS-232** 通讯电缆将该装置与所选的 **RS-232** 通讯端口口连接在一起，之后按下 **Auto-detect** 键即可自动检测联机装置参数，若检测成功时，该硬件装置的通讯协议参数便会被回填于窗口中的对应字段；不过进行自动检测时，系统将不会检测 **COM Port** 与 **ASCII / RTU** 这两个参数，因此执行自动检测前，必须先确定此二参数的设置值正确后方可进行操作。

- ⑥ 设置通讯响应的相关参数。其中 **Time of Auto-retry** 参数是设置联机错误时的重试次数，而 **Time Interval of Auto-retry** 参数则是设置重试的时间间隔。

◆ Ethernet Driver 参数设置

The screenshot shows the Ethernet Driver configuration window with the following sections and callouts:

- 1** Driver Name: Drv_EN
- 2** Connection Setup: Type: Ethernet
- 3** Ethernet Card: Description: Intel(R) 82577LM Gigabit Network Coni, IP: 169.254.95.246
- 4** IP Address Setting: Table with columns IP Address, Port Number, Comment. Row 1: 169.254.95.100, 502, PLC_1
- 5** Setup Responding Time: Time of Auto-retry: 3, Time Interval of Auto-retry [sec.]: 3

- ① 于 **Driver Name** 字段输入此 **Driver** 的识别名称。
- ② 于 **Connection Setup** 的 **Type** 字段下拉选择 **Ethernet** 的联机型式。
- ③ 于 **Ethernet Card** 的字段下拉选择欲建立联机信道的网络卡，而此字段左下角的数字即是所选网卡目前的 IP 地址。
- ④ 依据以太网络的特性，计算机可以与连接至同一个网域上的所有装置进行通讯，因此用户可在 **IP Address Setting** 区域中，预先将可能透过此 **Driver** 进行通讯的装置 IP 地址建立起来。

➢ 按下 **Add** 按钮可于下方的列表中新增一组 IP 地址，之后请于各字段中输入相关信息。

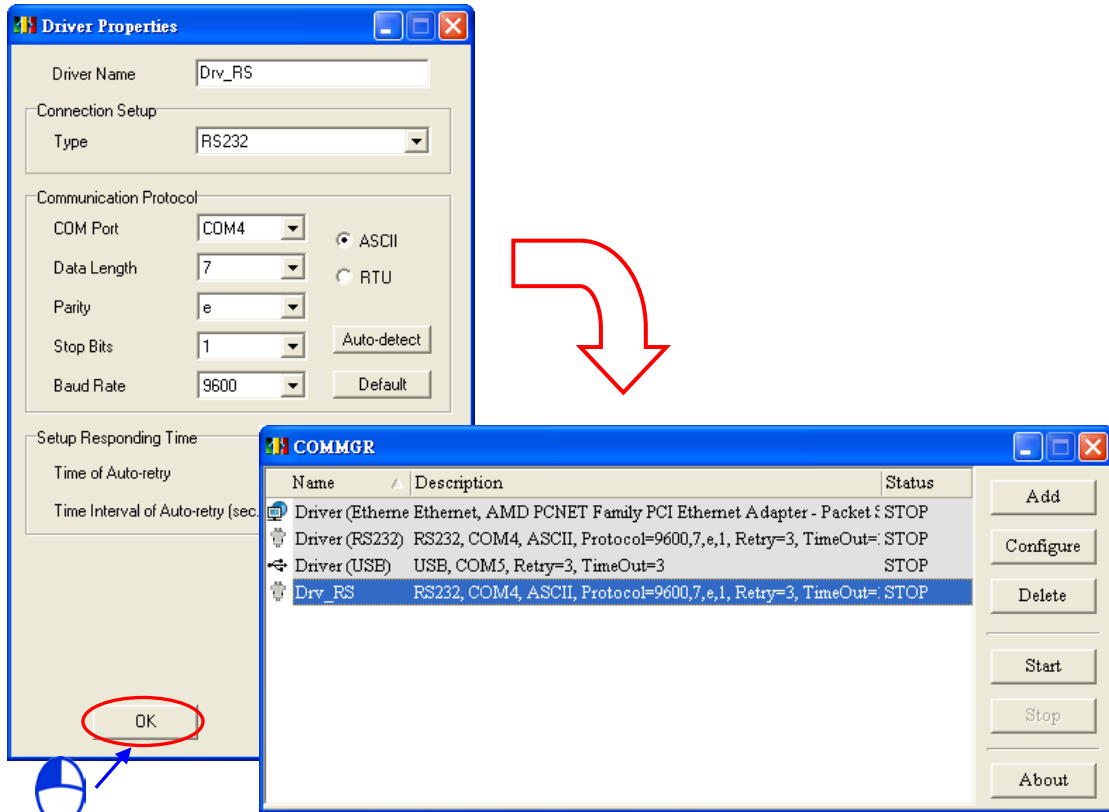
- ① 于此输入联机装置的 IP 地址。
- ② 于此输入指定的通讯端口编号。
- ③ 视用户的需求于此字段输入此组 IP 的批注。

IP Address	Port Number	Comment
169.254.95.100	502	PLC_1
192.168.1.1	502	

Callouts 1, 2, and 3 point to the IP Address, Port Number, and Comment columns of the second row respectively.

- 选取某组 IP 后按下 **Del** 按钮或直接按下键盘的 **Delete** 键即可将该组 IP 于列表中删除。
- ⑤ 设置通讯响应的相关参数。其中 **Time of Auto-retry** 参数是设置联机错误时的重试次数，而 **Time Interval of Auto-retry** 参数则是设置重试的时间间隔。

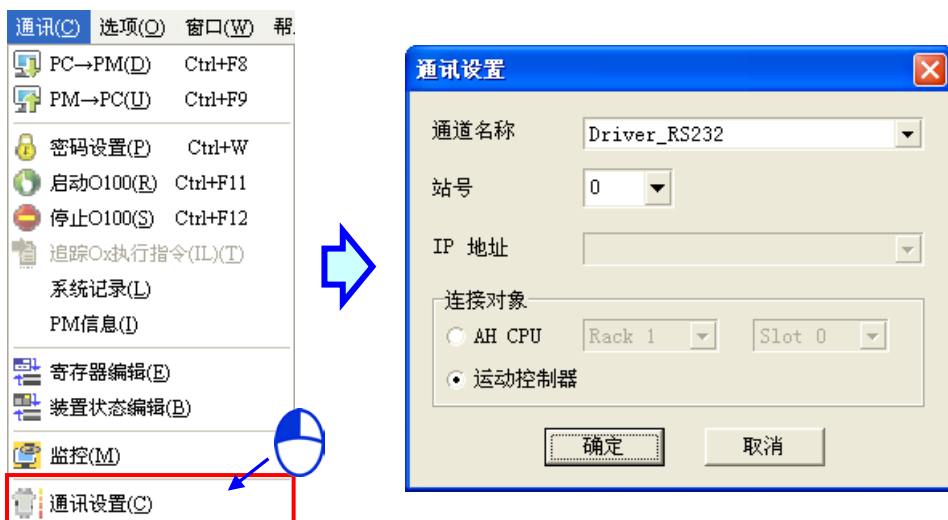
完成通讯参数的设置后按下 **OK** 键，之后于 **COMMGR** 主画面的列表中便会列出该组 Driver 的相关参数，而完成了一个 Driver 的建立便代表建立了一个联机通道，之后便可依据实际联机的需求而随时启动或关闭该组联机通道



3. 与 PMSoft 联机操作

◆ 与 PMSoft 直接联机

(1) 开启 PMSoft 软件，于功能工具列中点选 **通讯 (C)** > **通讯设置 (C)**。



- (2) 于**通道名称**字段处，下拉选择欲使用的 **Driver**，且在开始联机之前，请务必确认该 **Driver** 已于 **COMMGR** 中设为 **START** 状态，并且勾选连接对象为**运动控制器**，按下**确定**键完成设置。当选择不同的通道名称，下方的设置选项将随之改变，说明如下。



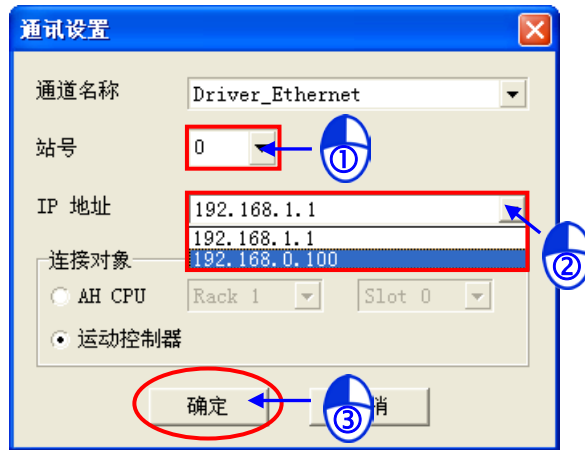
➤ RS232 及 USB

于站号字段下拉选择欲联机的硬件站号，当站号设置为 0 时，则代表要进行广播通讯。当无法通讯或不确定站号设定值时，可先将站号设定为 0 进行通讯。



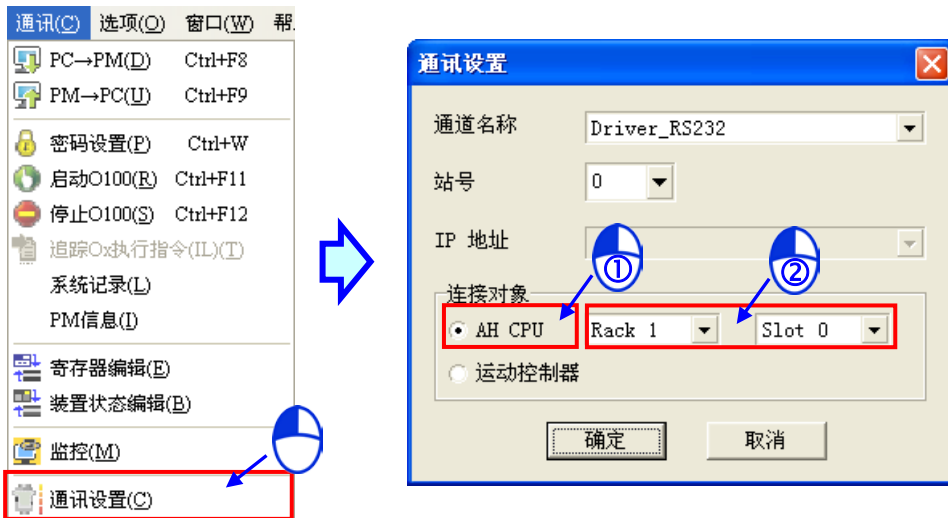
➤ Ethernet

于站号字段下拉选择欲联机的硬件站号，当站号设置为 0 时，则代表要进行广播通讯。之后再于 IP 地址字段中，下拉选择事先于 COMMGR 中所建立好的 IP 地址。



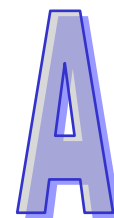
◆ 使用 ISPSOft 与 PMSOft 间接联机

于开启的 PMSOft 中点选功能工具列的 **通讯 (C)** > **通讯设置 (C)**，此时必须选择连接对象为 **AH500 CPU**，并且于右方选择正确的插槽编号。



更多关于 PMSOft 细部操作，请参照 PMSOft 软件手册

15



附录A 错误代码原因对照表

目录

A.1 错误代码原因对照表	A-2
---------------------	-----

A.1 错误代码原因对照表

将程序写入 AH500 运动控制模块内部后，在不同的程序区块 O100、OX 若发生 ERROR，错误指示灯闪烁，错误标志 On，原因可能是指令操作数（装置）使用不合法，程序语法回路错误，或运动参数设定不当，可根据错误寄存器的错误码（16 进制编码）并对照下表，以得知错误原因。

◆ O100/OX 对应之错误讯息表

错误类别	程序错误		运动错误
	程序区块		mn=10~25
	O100	OX	10 为第 1 轴，25 为第 16 轴
错误标志 (SM)	SM 953	SM 1049	SM mn49
错误寄存器 (SR)	SR 802	SR 1041	SR mn41
STEP 数	SR 803	SR 1053	

◆ 程序区及运动轴之错误码 (以 16 进制表示)

错误码	说明	错误码	说明
0002	使用子程序无内容	0031	正方向脉冲禁止
0003	CJ、CJN、JMP 缺少对应的 Pn	0032	反方向脉冲禁止
0004	主程序中有子程序标志	0033	到达左右极限
0005	缺少子程序	0040	装置组件使用范围错误
0006	同程序中指针重复	0044	V/Z 修饰错误
0007	子程序指针重复	0045	浮点数转换错误
0008	不同子程序中的转移指令指针重复	0047	读取 SD 卡中 OX 运动编号错误
0009	跳转指令与调用子程序指令使用相同标志	0E18	BCD 转换错误
000A	指针与子程序的指针一样	0E19	除法运算错误 (除数 = 0)
0011	目标位置(I)设定错误	C401	一般回路错误
0012	目标位置(II)设定错误	C402	LD/LDI 指令连续使用 9 次以上
0021	运转速度(I)设定错误	C404	RPT~RPE 超过 1 层以上
0022	运转速度(II)设定错误	C405	SRET 使用在 RPT~RPE 之间
0023	原点回归速度 (V _{RT}) 设定错误	C4EE	程序中没有结束指令
0024	原点回归减速速度 (V _{CR}) 设定错误	C4FF	无此指令/操作数格式或范围错误
0025	寸动 JOG 速度设定错误	8000	DMCNET 伺服通讯断线 (20MC 支持)



- ◆ 在 AH20MC-5A 机种中，在伺服已成功联机状态下，错误寄存器会包含部分伺服的错误代码，规则为伺服的 AL 代码+H8000 为错误码。
未列出之伺服错误码，可参考「ASDA-A2 系列 应用技术手册」第 10 章异警排除。

错误码	说明	错误码	说明
8001	伺服过电流	8014	伺服反向极限异常
8002	伺服过电压	8015	伺服正向极限异常
8003	伺服低电压	8016	伺服 IGBT 过热
8004	伺服电机匹配异常	8018	伺服检出器异常
8005	伺服回生异常	8030	伺服碰撞错误
8006	伺服过负载	8301	DMCNET 同步失效
8007	伺服过速度	8302	DMCNET 同步信号太快
8009	伺服位置控制误差过大	8303	DMCNET 同步信号超时
8011	伺服位置检出器异常		
8012	伺服校正异常		
8013	伺服紧急停止		



MEMO

