版本变更

版本	变更说明	修改者	日期
1.0	正式发行初版	林良腾	2014/01/30
2.0	修正【错误代码】(W576) 修正不同类型总线定义 新增【伺服参数同步错误】(R22)、【伺服参数同步错误码】(W22) 新增【手轮控制开关】(R144)、【手轮转速】(W80)、【手轮转速取样率】(W82)、【手轮转速平滑常数】(W83) 新增【循环记录关闭】(R498) 新增【记录输出档案启动】(R499) 新增【当前记录断电保持启动】(R500) 新增【伺服参数同步异常动作模式】(W14) 新增【手轮反向软件极限】(W690)、【手轮正向软件极限】(W692) 新增检视 HMC 命令历史记录功能（章节 6.3) 新增系统装置数据撷取功能（章节 7.4) 新增程序设计错误范例（章节 8)	林良腾	2015/04/01

内容:

1. 前言

1.1、 HMC控制器概述

现今产业朝向高度自动化的趋势发展，自动化设备追求高精,高速与更高的性价比，因此台达提出了分布式运动控制架构. 在此分布式架构中将逻辑控制器与运动计算控制器区隔开来，借着将原本集中式的大量计算负载分散开来，因而能使用多个较低阶处理器(整体而言较低价)来达成原本需要高阶处理器计算的运动效能与逻辑控制效果.

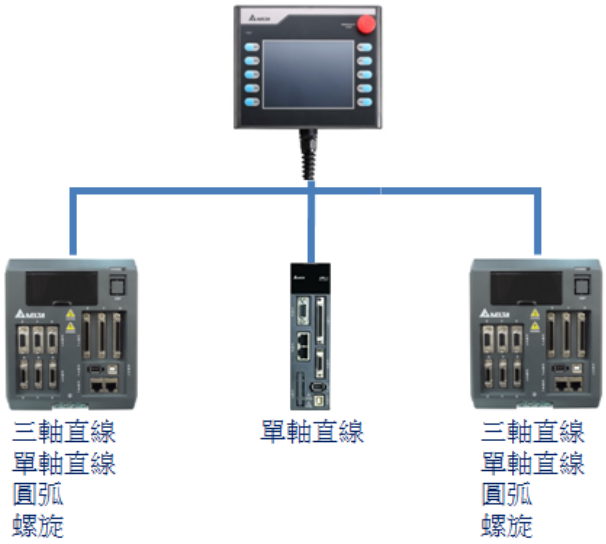
在台达提出的分布式架构中，我们将逻辑思考控制与人机操作整合工作交由 HMC 负责，将运动计算的控制工作交由 ASDA 驱动器负责，两者间透过 DMCNet 高速通讯可完美结合 HMC 与伺服驱动器. 这样的分散计算概念能实现多轴精准的运动控制，同时也能使整体设备的建置成本大幅降低.

台达 HMC (Human Machine Interface & Control)整合了人机操作接口(HMI)与逻辑计算(控制器)的功能，实现操作接口与逻辑控制的高效能整合. 台达 HMC 在现今工业设备系统与分散控制架构中将担任举足轻重之角色，带给用户更加强大功能与缩短开发工时等效益.

1.2、 分布式运动控制概述

在台达的分布式运动架构中，将程序逻辑等判断交给 HMC 处理，其中包含运动命令参数的计算与下达. HMC 将目标运动参数下达给 ASDA 驱动器并触发运动就不需再理会运动过程中路径补间计算工作，这一切与运动过程有关的计算将完全交由 ASDA 驱动器来直接处理，如此也能得到更精准的运动补间路径. 在运动过程中 HMC 上位控制只需定时与 ASDA 驱动器交换所需的数值信息(如当前位置，当前速度等)与旗标状态(伺服异常，命令完成等)即可.

由于运动过程中的补间计算是交由 ASDA 驱动器负责的，而非传统 PLC 在运动过程中需频繁地计算运动路径并下达给驱动器，因此路径将会更精准，运动曲线更平滑. 但由于多轴的补间运动(如多轴直线同动，圆弧或螺旋)是直接下达给 ASDA 驱动器并由驱动器自行运算执行的，因此这类运动补间计算是无法跨越驱动器的，这点使用时需要特别注意. 但对于跨越驱动器间的多轴直线同动运动，HMC 使用调配速度的方式来克服此限制，这在之后会有更详细的运动方式说明. 在分散架构下运动执行示意图如下.


三軸直線
單軸直線
圓弧
螺旋
單軸直線
三軸直線
單軸直線
圓弧
螺旋

2. 控制器介绍

2.1、 控制器架构

HMC 控制器使用的阶梯图工作原理为分时多任务处理与传统 PLC 阶梯图不同.

传统 PLC 只有一个处理器且执行单一个阶梯图程序, PLC 工作原理为每次执行周期的开始会先读取输入设备的状态, 接着逐一执行每行指令, 并且在执行周期终了会将演算结果送到输出装置, 如此周而复始地执行【读取输入状态】➔【演算】➔【改变输出状态】的动作循环. 然而单一的阶梯图程序面对愈来愈复杂庞大的控制程序与运算将面对程序开发与维护上的困难.

HMC 控制器使用分时多任务的处理架构, 能够只执行单一个主程序到最多同时执行四个主程序(Cyclic Task)运行的设计, 提供用户更大的程序开发弹性. 当 HMC 同时执行四个主程序时, 可视为有独立的四个小型 PLC 来规划, 将有利于复杂项目的程序开发. HMC 的每一个主程序都有各自的扫描运行时间, 透过软件设定让用户决定处理器执行每个主程序的时间比重, 进而达到调整每个程序的扫描周期时间. 分配更多比例的处理器运行时间给重要的主程序, 如此就能缩短重要主程序的扫描周期.

HMC 控制器也采用呼叫子程序的概念, 将使用到的功能规划为子程序, 当需要时只需呼叫该功能子程序, 如此就能简化程序的开发工作. HMC 控制器对于运动控制需求也提供了运动程序(Motion Program), 以提供给使用者更多元化的控制方式.

HMC 控制器的程序型态共有初始程序(Initial Task), 主程序(Cyclic Task), 定时程序(Timer Task), 子程序(Sub Program)和运动程序(Motion Program)等类型, 以下针对各类型程序将有更详尽的说明介绍.

阶梯图程序

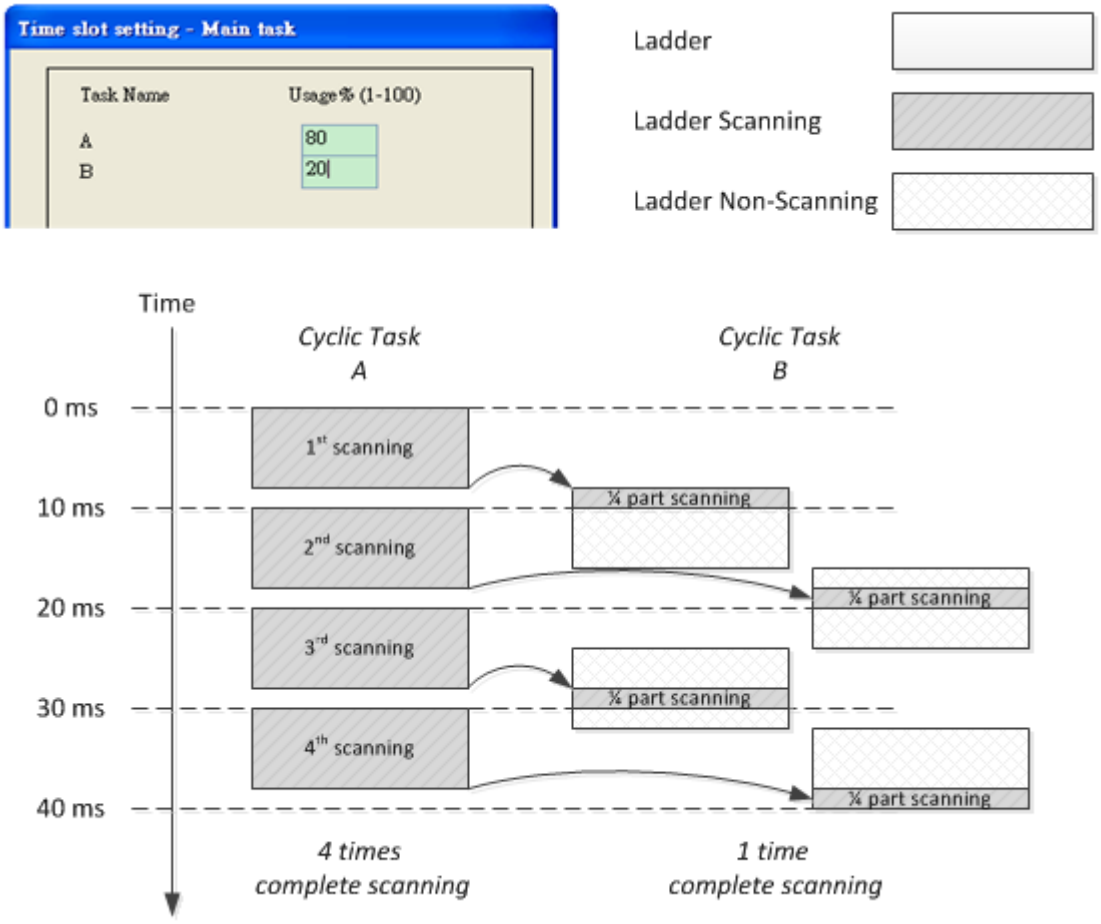
- 初始程序(INITIAL TASK):

在整个项目中只存在一个初始程序, HMC 启动后首先执行且只执行一次的程序, 因此可将系统的初始设定规画在初始程序中. 在初始程序执行时由于 DMCNet 通讯尚未建立完成, 因此在初始程序使用读写伺服参数动作指令, 例如【WSVP】,【RSVP】指令, 或读写对应伺服参数的 W 地址将会失效.

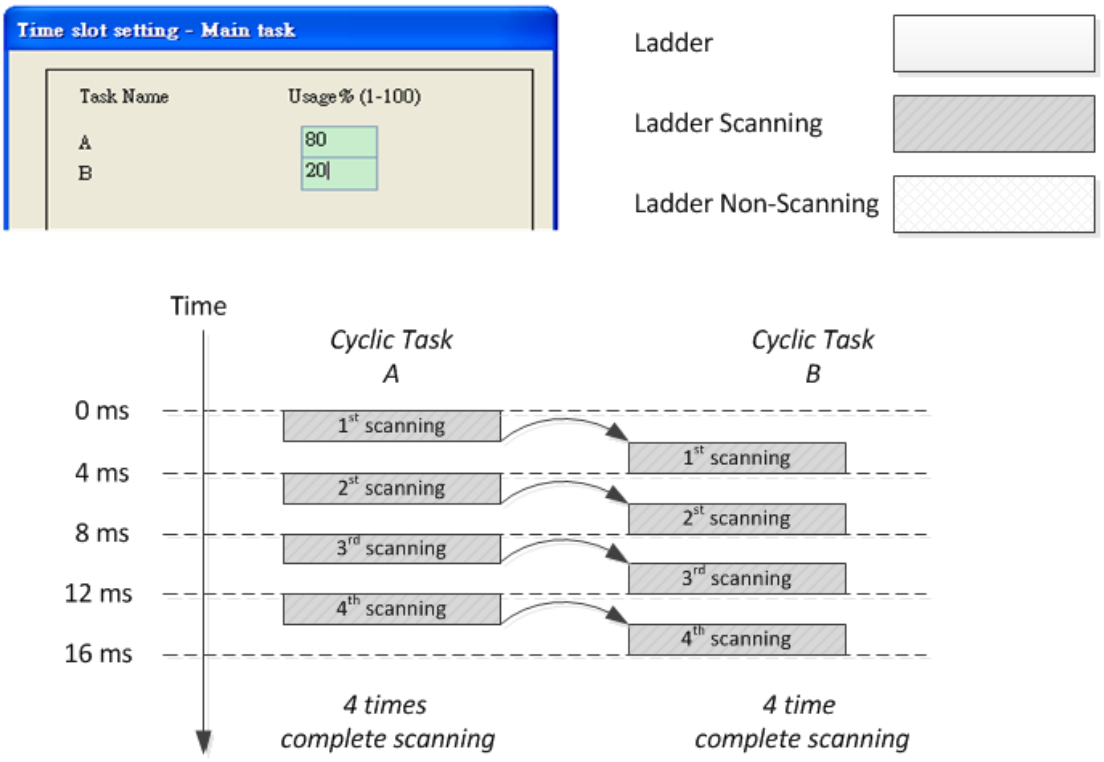
- 主程序(CYCLIC TASK):

在项目中至少存在一个主程序, 而最多可以增加至四个主程序, HMC 以多任务方式来同时执行这些主程序. HMC 控制器执行主程序的方式为透过设定各个主程序的使用率(Usage%)来决定处理器执行主程序的运算时间分配.

如下图为例，若项目中有 A 与 B 两个主程序，两者有相同的阶梯图程序，皆需要 8 ms 的扫描运算时间才能处理完毕，而两者使用率设定分别为 80%与 20%，因此处理器会将每周期的处理时间 10ms 分配 8 ms 给 A 主程序执行扫描与 2 ms 给 B 主程序执行扫描，执行两个主程序的行为将如以下示意图. 在控制器的第一次周期中, A 主程序会分配到 8 ms 的时间执行，因此完成了一次完整的程序扫描; B 主程序只分配到 2 ms 的时间执行，因此仅执行四分之一的程序. 在第二次周期中, A 主程序又完成了一次完整的程序扫描; B 主程序接着执行下一个四分之一的程序. 最后，当经过四次扫描周期后, B 主程序才执行完一次完整程序扫描，但 A 主程序已经执行完成四次了，也就是在这个例子中, A 主程序的扫描周期将会是 B 程序的四分之一. 因此，我们可以针对不同重要性的主程序，设定予不同的使用率，对于重要或需实时性处理的主程序可分配较高的使用率，如此将能缩短该主程序的周期扫描时间.



在下一个例子中，使用率的分配设定对于主程序的扫描时间影响将会不同于上个例子. 以下图为例，若项目中有 A 与 B 两个主程序，两者也都有相同的阶梯图程序，皆需要 2 ms 的扫描运算时间才能处理完毕，而两者使用率设定同样为 80%与 20%，因此处理器会将每周期的处理时间 10ms 分配 8 ms 给 A 主程序执行扫描与 2 ms 给 B 主程序执行扫描，执行两个主程序的行为将如以下示意图. 在控制器的第一次周期中, A 主程序会分配到 8 ms 的时间执行，由于只需花费 2 ms 时间即可完成整个程序的扫描，因此当完成后会立即切换至 B 主程序执行；而 B 主程序分配到 2 ms 的时间执行，也同样完成整个程序的扫描执行. 因此在这个例子中, A 主程序的扫描周期与 B 程序是相同的.

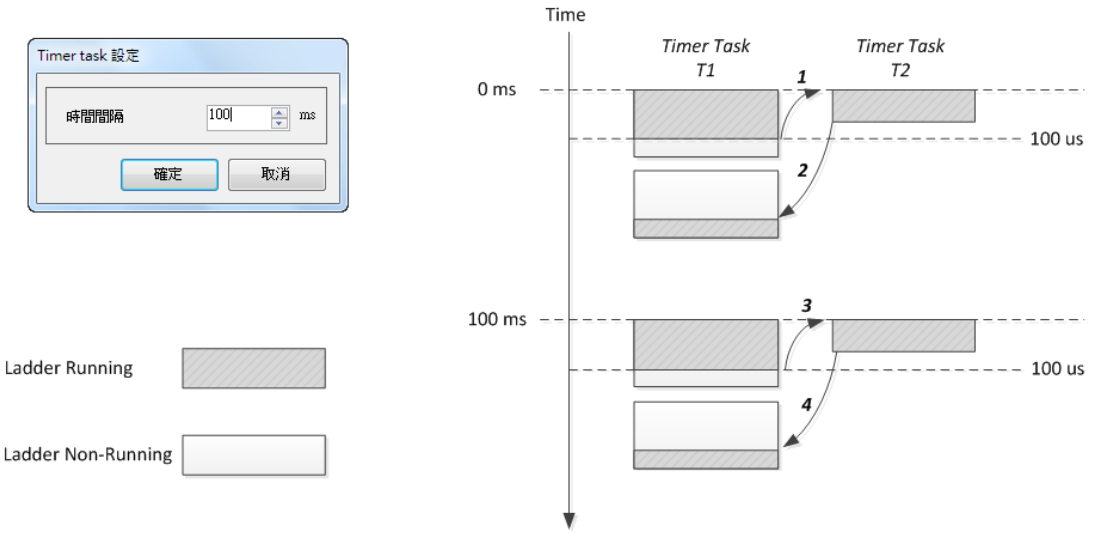


一般控制器的阶梯图程序在程序执行一开始会先将外部输入信号 On/Off 状态读进输入信号装置记忆区，接着程序执行时会将运算结果存入各装置记忆区内，当执行到 END 指令时，装置记忆区内输出 On/Off 状态将会送到输出装置记忆区以改变外部实际输出. 由于 HMC 控制器能同时存在多个主程序执行，在每个主程序开始时都会读取一次外部输入信号，同样的任一个主程序执行到 END 指令时，会立即将装置记忆区内运算结果的 On/Off 状态输出至外部输出装置中.

➢ 定时程序(TIMER TASK)：

在整个项目中最多可增至 8 个定时程序执行，每个定时程序都能设定各自的执行【时间间隔】(单位 ms)，且定时程序有最高的执行优先权以确保能被定时的执行. 由于能多个定时程序同时执行，因此对单一定时程序不可连续执行过久，以免造成其它定时程序无法实时执行，在系统中可设定【最长执行切换时间】(预设 50us)，以允许在多个定时程序间切换执行，达成同时执行多个定时程序的效果.

如下图为例，若项目中有 T1 与 T2 两个定时程序且设定【最长执行切换时间】为 100us, T1 为较大的阶梯图程序，完整扫描时间大于 100 us; T2 为较小程序且完整扫描时间少于 100 us，两者设定【时间间隔】皆为 100 ms. 如下图所示, T1 每隔 100 ms 即执行一次，但在 100 us 内无法完成整个程序扫描，当扫描执行 100us 后即停止 T1 的执行，处理器切换至 T2 执行; T2 同样为间隔 100 ms 执行一次，但 T2 程序完成扫描运行时间不会超过 100 us，因此 T2 会在完成完整扫描后，控制器立即切回 T1 程序执行直到 T1 完成.



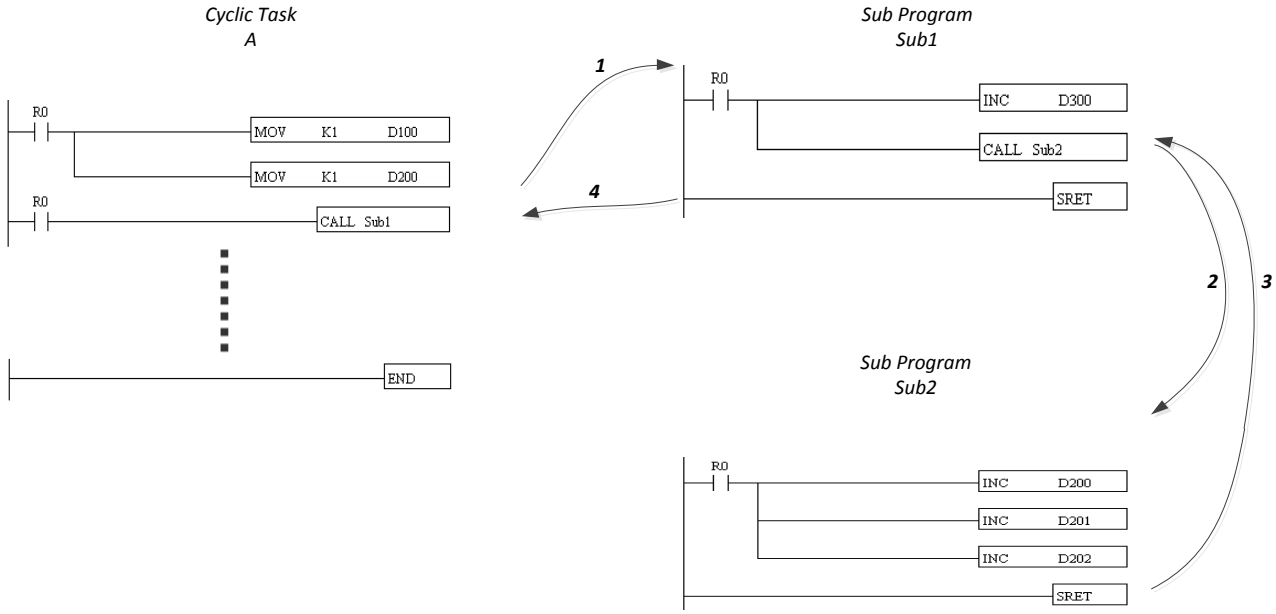
使用这类程序需特别注意，定时程序在所有程序种类中有最高的优先执行特性，因此若过于频繁地执行庞大定时程序将会影响到主程序的扫描周期，甚至将无法正常地运行.

- 子程序(SUB PROGRAM)：

在一个项目中最多可使用 256 个子程序，可将不同功能区块的程序放在不同的子程序中，而所有类型程序都可以重复地呼叫这些子程序，达到程序模块化目的，以利维护与提高可读性.

使用【CALL】子程序名称方式实现呼叫子程序功能，呼叫子程序后，处理器执行被呼叫的子程序，当执行到子程序中的【SRET】指令时，表示结束此子程序扫描并离开回到原本呼叫此子程序指令处继续执行.

如下图为例，在主程序 A 中以【CALL Sub1】指令呼叫子程序 Sub1 后，接着会执行子程序 Sub1. 在子程序 Sub1 中又再以【CALL Sub2】指令呼叫子程序 Sub2，因此切换到子程序 Sub2 执行. 当在子程序 Sub2 中执行到【SRET】指令，则子程序 Sub2 执行完毕并返回至子程序 Sub1 中的【CALL Sub2】指令处往下继续执行. 相同地，执行子程序 Sub1 中遇到【SRET】指令，也代表子程序 Sub1 执行完毕，接着返回主程序 A 并从【CALL Sub1】指令处往下继续执行，直到主程序的 END 指令处，则主程序扫描完毕.



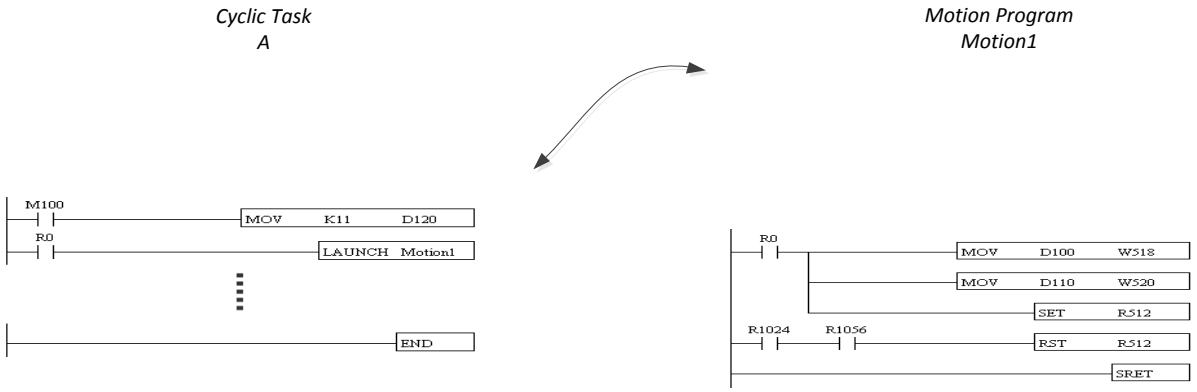
由于子程序中也允许呼叫子程序，请注意最多允许 8 层的子程序呼叫，若超过此限制则系统的【文法错误】(R18)状态会变为 On，同时【文法错误代码】(W18)显示为 6 表示.

- 运动程序(MOTION PROGRAM)：

在一个项目中最多可使用至 256 个运动程序. 运动程序特性类似子程序，不同的是当呼叫运动程序后原本执行中的程序还是会继续扫描执行，此时被呼叫的运动程序将启动且与原执行程序一同执行. 一般此类型程序特性多运用在运动控制中，为了能让主程序触发运动后继续执行原有的逻辑控制扫描，并不会受到触发运动动作而影响到原本的逻辑流程.

使用【LAUNCH】加上运动程序名称方式以启动运动程序执行，当运动程序中执行至【SRET】指令时即表示此运动程序结束. 同时间内只能有一个运动程序被执行，若有多个运动程序被同时呼叫，则这些运动程序将依序一个接一个地执行. 呼叫一次运动程序时该运动程序只会执行一次扫描，这点是要特别注意的.

如下图为例，在执行主程序 A 过程中以【LAUNCH Motion1】指令呼叫运动程序 Motion1 执行，主程序 A 中执行【LAUNCH Motion1】指令后并不会中断原本的程序扫描动作，会紧接着执行下一行指令；此时若没有其它运动程序在执行中，则会开始执行 Motion1 的程序指令直到【SRET】指令表试运动程序结束.



特别注意，运动程序中并无法再呼叫运动程序. HMC 最多只能暂存 256 个等待执行的运动程序，也就是未执行的运动程序不可超过 256 个. 若超过此限制则系统的【文法错误】(R18)状态会变为 On, 同时【文法错误代码】(W18)显示为 12 表示.

2.2、 使用装置

HMC 控制器使用的装置与范围一览表.

类别	装置	项目		装置范围		内容值范围
继电器(位)	X	输入继电器		0 ~ 511	共 512 点	
	Y	输出继电器		0 ~ 511	共 512 点	
	DX	DMCNet 输入继电器		1.0 ~ 12.63	共 768 点	
	DY	DMCNet 输出继电器		1.0 ~ 12.63	共 768 点	
	M	辅助继电器	一般用	0 ~ 511 1024 ~ 4095	共 4096 点	
			停电保持用	512 ~ 1023 (W10,W11 调整)		
	T	定时器	100ms	0 ~ 199	共 256 点	
			10ms	200~255		
	C	计数器	16 位	0 ~ 199	共 256 点	
			32 位	200 ~ 255		
	R	特殊继电器		0 ~ 1535	共 1536 点	
暂存器(字符)	T	定时器现在值	16 位	0 ~ 255	共 256 点	0 ~ 65535
	C	计数器现在值	16 位	0 ~ 199	共 256 点	0 ~ 65535
			32 位	200 ~ 255		-2147,483,648~2147,483,647
	D	数据缓存器	16 位 一般用	0 ~ 2999 4000 ~ 65535	共 65536 点	-32,768 ~ 32,767
			停电	3000~3999		

				保持用	(W12,W13 调整)		
	V	间接指定缓存器	16 位		0 ~ 127	共 128 点	-32,768~32,767
	Z	间接指定缓存器	32 位		0 ~ 127	共 128 点	-2147,483,648~2147,483,647
	W	特殊缓存器	16 位		0 ~ 4095	共 4096 点	
指标	N	主控回路指标			0 ~ 7	共 8 点	
	P	跳跃指标			0 ~ 255	共 256 点	
常数	K	10 进制常数					
浮点数	F	浮点数					

➢ 输入继电器(X)/ 输出继电器(Y)

输入/输出继电器以 10 进制编号，分别对应至 Remote I/O 模块的输入点及输出点，对应地址如下表:

装置	Remote I/O				
	第 1 站	第 2 站	第 3 站	~	第 16 站
输入 X	X0 ~ X31	X32 ~ X63	X64 ~ X95	~	X480~X511
输出 Y	Y0 ~ Y31	Y32 ~ Y63	Y64 ~ Y95	~	Y480~Y511

注一：一个 Remote I/O 模块有 32 输入点与 32 输出点.

注二：Remote I/O 模块最多可串接 16 站.

注三：Remote I/O 模块使用站号是以模块使用数量搭配模块上实体站号旋钮设定来决定. 例如 HMC 若供连接三个 Remote I/O 模块使用，则实体设定站号最小者为第一站，实体设定站号次之为第二站，实体设定站号最大为第三站.

■ 输入继电器(X)

与输入设备连接读取输入讯号. 每一输入继电器的 A 或 B 接点于程序中使用次数没有限制. 输入继电器 X 之 On/Off 只会跟随外部输入设备的 On/Off 做变化.

■ 输出继电器(Y)

送出 On/Off 信号以驱动连接输出接点 Y 的负载. 每一输出继电器的 A 或 B 接点于程序中使用次数没有限制.

➢ DMCNET输入继电器(DX)/ DMCNET输出继电器(DY)

DMCNet 输入/输出继电器以 10 进制编号, DMCNet 输入继电器 DX 及 DMCNet 输出继电器 DY 分别对应 DMCNet RM(MN\NT\PT)模块或 HMC-RIO3232RT5 模块的输入点及输出点，对应地址如下表:

装置	DMC-RMxx(MN\NT\PT), HMC-RIO3232RT5			
	第 1 站	第 2 站	~	第 12 站
输入 DX	DX1.0 ~ DX1.63	DX2.0 ~ DX2.63	~	DX12.0 ~ DX12.63
输出 DY	DY1.0 ~ DY1.63	DY2.0 ~ DY2.63	~	DY12.0 ~ DY12.63

■ DMCNET 输入继电器(DX)

以 DMCNet 连接方式透过 DMC-RM 模块读取输入讯号. 每一输入继电器的 A 或 B 接点于程序中使用次数没有限制. 输入继电器 DX 之 On/Off 只会跟随外部输入设备的 On/Off 做变化.

■ DMCNET 输出继电器(DY)

送出 On/Off 信号并透过 DMC-RM 模块驱动输出接点 DY 的负载. 每一输出继电器的 A 或 B 接点于程序中使用次数没有限制.

➢ 辅助继电器(M)

辅助继电器 M 与输出继电器 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. 依其性质可区分为下列两种:

辅助继电器 M	一般用	M0~M511 M1024~M4095	合计 4096 点
	停电保持用	M512~M1023，预设 512 点为停电保持区域 以 W10, W11 调整此区域范围	

■ 一般用辅助继电器

遇到停电，其状态将全部被复归为 Off，再送电时其状态仍为 Off.

■ 停电保持用辅助继电器

遇到停电，其状态将全部被保持，再送电时其状态为停电前状态.

➢ 定时器(T)

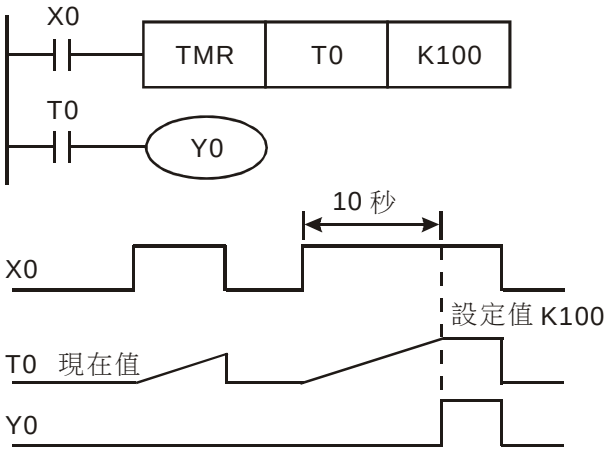
定时器以 10 进制编号，范围: T0 ~T255.

定时器 T	100ms 一般用	T0~T199, 200 点	合计 256 点
	10ms 一般用	T200~T255, 56 点	

定时器是以 10ms 或 100ms 为一个计时单位，计时方式采上数计时．当【定时器现在值=设定值】时，输出线圈导通．设定值为 10 进制 K 值，亦可使用数据缓存器 D 当成设定值．

定时器之实际设定时间 = 计时单位 * 设定值．

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．



当 X0=On 时，定时器 T0 之现在值以 100ms 采上数计时，当定时器现在值=设定值 K100 (10 秒)时，输出线圈 T0=On.
当 X0=Off 或停电时，定时器 T0 之现在值清为 0，输出线圈 T0 变为 Off.

➢ 计数器(C)

计数器以 10 进制编号，范围: C0 ~C255.

计数器 C	16 位上数一般用	C0~C199, 200 点	合计 256 点
	32 位上下数一般用	C200~C255,56 点.可使用 R32~ R87 设定变更成下数.	

计数器特点：

项目	16 位计数器	32 位计数器
类型	一般型	一般型
计数方向	上数	上，下数
设定值	0 ~ 65,535	-2,147,483,648 ~ 2,147,483,647
设定值的指定	常数 K 或数据缓存器 D	常数 K 或数据缓存器 D(指定 2 个)
现在值的变化	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归成 Off
复归动作	RST 指令被执行时现在值归零，接点被复归成 Off.	

计数器之计数条件信号由 Off ➔ On 时，计数器会累加 1．当计数器现在值等于设定值时，输出线圈导通．计数器设定值为 10 进制 K 值(当设定 K0 或 K1 时动作相同，在第一次计数时输出接点马上导通)，亦可使用数据缓存器 D 当成设定值．

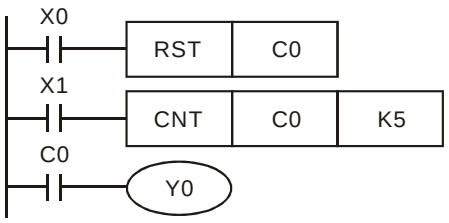
16 位计数器 C0~C199:

16 位计数器的设定范围: K0~K65,535.

计数器之设定值可使用常数 K 直接设定或使用缓存器 D 中之数值作间接设定.

范例:

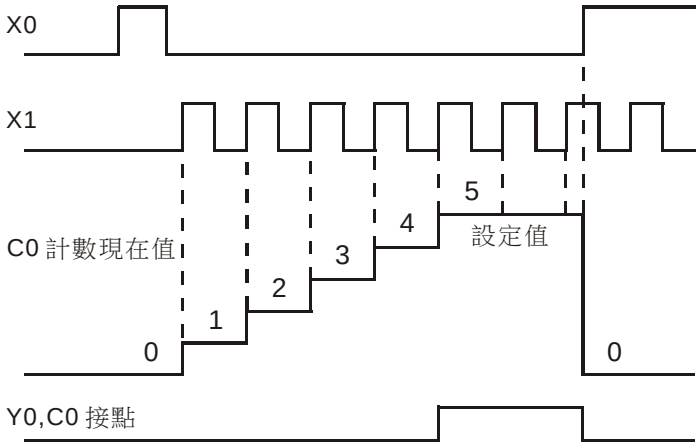
```
LD      X0
RST     C0
LD      X1
CNT     C0   K5
LD      C0
OUT     Y0
```



当 X0=On 时 RST 指令被执行, C0 的现在值归零, 输出接点被复归为 Off.

当 X1 由 Off ➔ On 时, 计数器之现在值将执行上数（加一）的动作.

当计数器 C0 计数到达设定值 K5 时, C0 接点导通, C0 现在值＝设定值=K5. 之后的 X1 触发信号 C0 完全不接受, C0 现在值保持在 K5 处.



32 位一般用加/减算计数器 C200~C255:

32 位一般用计数器的设定范围: K-2,147,483,648 ~ K2,147,483,647.

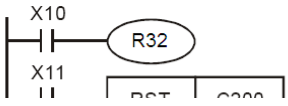
32 位一般用加减算计数器切换上下数用特殊辅助继电器: 由 R32~R87 来决定, 例如 R32=Off 时决定 C200 为加算, R32=On 时决定 C200 为减算, 其余类推.

设定值可使用常数 K 或使用数据缓存器 D 作为设定值, 可以是正负数, 若使用数据缓存器 D 则一个设定值占用两个连续的数据缓存器.

计数器现在值由 2,147,483,647 再往上累计时则变为-2,147,483,648. 同理计数器现在值由 -2,147,483,648 再往下递减时, 则变为 2,147,483,647.

范例：

```
LD      X10
OUT     R32
```



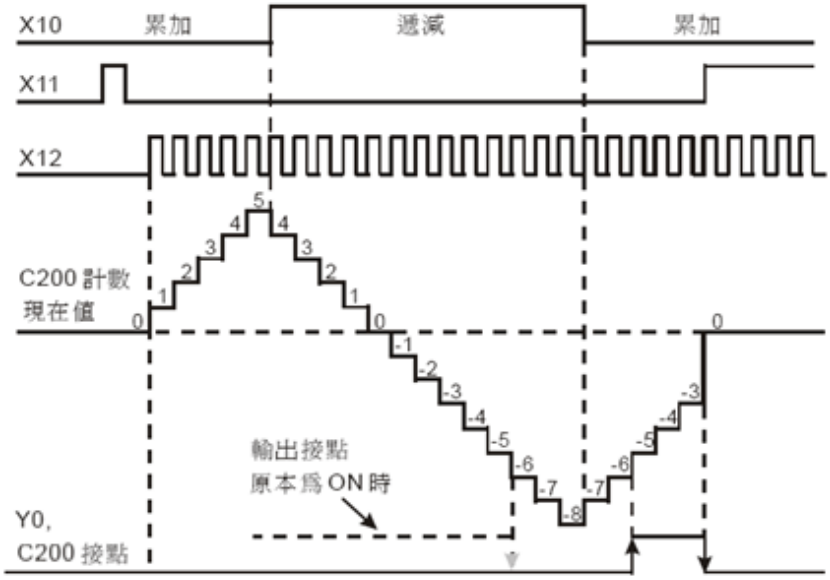
```
LD      X11
RST     C200
LD      X12
DCNT    C200   K-5
LD      C200
OUT     Y0
```

X10 驱动 R32 来决定 C200 为加算或减算.

当 X11 由 Off➔On 时, RST 指令执行, C200 之现在值被清为 0, 且接点变为 Off.

当 X12 由 Off➔On 时, 计数器之现在值将执行上数（加一）的动作或下数（减一）的动作.

当计数器 C200 之现在值从 K-6➔K-5(上数)变化时, C200 接点由 Off➔On. 当计数器 C200 之现在值从 K-5➔K-6(下数)变化时, C200 接点由 On➔Off.



➢ 数据缓存器(D)

用于储存数值数据, 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 位缓存器合并成一个 32 位缓存器(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 之数值资料.

数据缓存器 D	一般用	D0~D2999 D4000~D65535	合计 65536 点
	停电保持用	D3000~D3999, 预设 1000 点为停电保持区域, 以 W12, W13 调整此区域范围	

■ 一般用缓存器

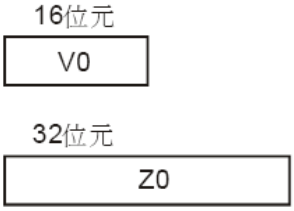
遇到断电时其值会被清除为 0.

■ 停电保持用缓存器

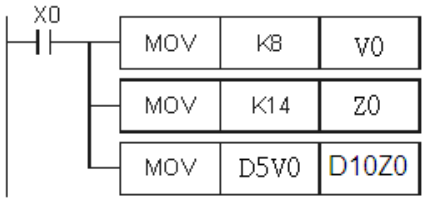
遇到断电时其值会保持.

➢ 间接指定用缓存器(V)/ (Z)

间接指定缓存器 V 为 16 位缓存器，而 Z 为 32 位缓存器. 范围 V0~V127 与 Z0~Z127 各 128 点.



V 与一般的数据缓存器一样的都是 16 位的数据缓存器，它可以自由的被写入及读出，若当一般缓存器用，仅能使用在 16 位的指令里.
Z 为 32 位的数据缓存器，若当一般缓存器用，仅能使用在 32 位的指令里.



当 X0=On 时, V0=8, Z0=14,
D5V0 = D(5+8) = D13,
D10Z0 = D(10+14) = D24,
此时会将 D13 的内容搬移至 D24 内.

➢ 指标(N)/ 指标(P)

指标	N	主控回路用	N0~N7, 8 点	主控回路控制点
	P	CJ 指令用	P0~P255, 256 点	CJ 的位置指针

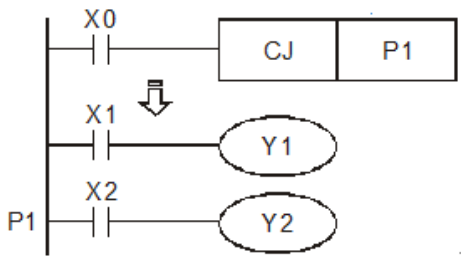
■ 指标 N

搭配指令 MC/MCR 使用, MC 为主控起始指令，当 MC 指令执行时，位于 MC 与 MCR 指令之间的指令照常执行.

■ 指标 P

搭配应用指令 CJ 使用.

范例如下:



当 X0=On 时程序自动从地址 0 跳跃至地址 N（即指定之标签 P1）继续执行，中间地址跳过不执行.
当 X0=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.

➢ 特殊继电器(R)/ 特殊缓存器(W)

在特殊装置章节中有更详细使用说明.

分类	范围
PLC 系统特殊继电器	R0 ~ R511
PLC 系统特殊缓存器	W0 ~ W511
运动模式特殊继电器	R512 ~ R1535
运动模式特殊缓存器	W512 ~ W4095

➢ 常数(K)/ 浮点数(F)

共使用 2 种数值类型执行运算的工作，各种数值的任务及功能如下说明. 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 bit0～bit3）可用以表示一个位数之 10 进制数字 0～15 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, bit0～bit7). 可表示 16 进制之 00～FF.
字符组 (Word)	是由连续之两个字节所组成(亦即 16 位, bit0~bit15)可表示 16 进制之 4 个位数值 0000~FFFF.
双字符组 (Double Word)	是由连续之两个字符组所组成(亦即 32 位, bit0~bit31)，可表示 16 进制之 8 个位数值 00000000~FFFFFFFF.

■ 常数 K

十进制数值通常会在数值前面冠以一【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 X, Y, M 可组合成为位数，字节，字符组或双字符组形式的数据.
例: K2Y10, 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.

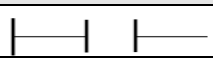
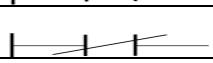
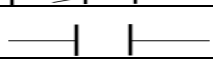
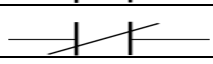
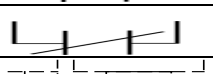
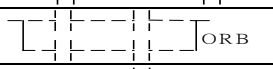
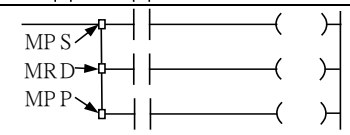
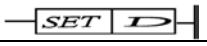
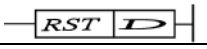
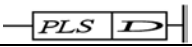
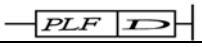
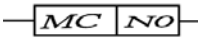
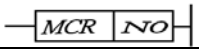
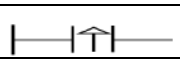
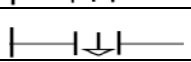
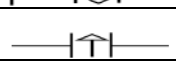
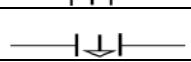
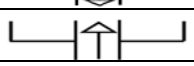
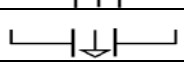
■ 常数 F：

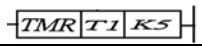
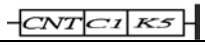
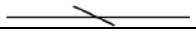
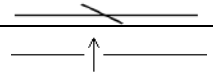
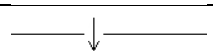
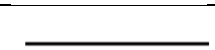
浮点数在应用指令中作为操作数使用，例:【FADD F12.3 F0 D0】. (F 浮点常数)

2.3、 使用指令一览表

以下为 HMC 控制器提供的指令.

基本命令

分类	命令记号	符号
接点命令	LD	
	LDI	
	AND	
	ANI	
	OR	
	ORI	
结合命令	ANB	
	ORB	
	MPS	
	MRD	
	MPP	
输出命令	OUT	
	SET	
	RST	
	PLS	
	P F	
主控命令	MC	
	MCR	
上下缘检出	LDP	
	LDF	
	ANDP	
	ANDF	
	ORP	
	ORF	

分类	命令记号	符号
计时	TMR	
计数	CNT	
	DCNT	
主程序结束	END	
定时程序结束	IRET	
子程序结束	SRET	
反相	INV	
上升缘	NP	
下降缘	PN	
无动作	NOP	

应用命令

分类	API	脚本		功能	STEPS
		16 位	32 位		
数据比较	001	LD※	DLD※	接点型态比较	5
	002	AND※	DAND※	接点型态比较	5
	003	OR※	DOR※	接点型态比较	5
数据传送	004	MOV	DMOV	数据移动	5
	005	BMOV	-	全部传送	11

	与比较	006	CML	DCML	反转传送	5
		007	BCD	DNCD	BIN→BCD 变换	5
		008	BIN	DBIN	BCD→BIN 变换	5
		009	-	FCMP	浮点数比较	7
		050	FMOV	DFMOV	多点移动	11
	I/O	010	REF	-	I/O 更新	2
	旋转位移	011	ROR	DROR	右旋转	3
		012	ROL	DROL	左旋转	3
	回路控制	013	CJ	-	条件跳跃	2
		014	CALL	-	呼叫子程序	2
		015	LAUNCH	-	启动运动程序	2
		016	FOR	-	巢串回路起始	3
		017	NEXT	-	巢串回路结束	1
	四则运算	018	ADD	DADD	BIN 加法	7
		019	SUB	DSUB	BIN 减法	7
		020	MUL	DMUL	BIN 乘法	7
		021	DIV	DDIV	BIN 除法	7
		022	INC	DINC	BIN 加一	3
		023	DEC	DDEC	BIN 减一	3
	逻辑运算	024	WAND	DWAND	AND 运算	7
		025	WOR	DWOR	OR 运算	7
		026	WXOR	DWXOR	XOR 运算	7
		027	NEG	DNEG	2 的补码	3
	浮点运算与转换	028	-	FADD	浮点数加法	7
		029	-	FSUB	浮点数减法	7
		030	-	FMUL	浮点数乘法	7
		031	-	FDIV	浮点数除法	7
		032	-	FINT	浮点数→整数	5
		033	-	FDOT	整数→浮点数	5
		034		FRAD	角度→径度	5
		035		FDEG	径度→角度	5
		036		FSIN	浮点数 SIN 运算	5
		037		FCOS	浮点数 COS 运算	5
		038		FTAN	浮点数 TAN 运算	5
		039		FASIN	浮点数 ASIN 运算	5
		040		FACOS	浮点数 ACOS 运算	5

		041		FATAN	浮点数 ATAN 运算	5	
		042		FSQR	浮点数开平方根运算	5	
	数据处理	043	ZRST	-	区域清除	4	
		044	DECO	-	译码器	11	
		045	ENCO	-	编码器	11	
		046	BON	DBON	ON 位判定	5	
		047	ALT	-	ON/OFF 交替	2	
	其它	048	RSVP	-	读取驱动器参数	13	
		049	WSVP	-	写入驱动器参数	13	
		051	CKFZ		禁区检查	5	

3. 特殊装置

3.1、 特殊装置一览表

特殊装置共有特殊继电器(R)与特殊缓存器(W)两类型，针对控制器系统与伺服运动控制又可分为【PLC系统】与【运动模式】两大类. 使用HMC的特殊装置即可实现系统的运动控制与监控功能. 请特别注意在HMC中对这类特殊缓存器不能使用区块搬移指令(例如BMOV指令)执行批次性改变内容值动作.

分类	范围
PLC 系统特殊继电器	R0 ~ R511
运动模式特殊继电器	R512 ~ R1535
PLC 系统特殊缓存器	W0 ~ W511
运动模式特殊缓存器	W512 ~ W4095

3.2、 PLC系统特殊继电器

此类继电器可用以取得目前系统之状态，包含计算结果，错误异常监控，外围装置联机与实体按键触发等各种状态.

分类	编号	功能	说明	属性	停电保持
运转旗标	R0	常闭接点	B 接点	R	No
	R1	常开接点	A 接点	R	No
	R4	错误总旗标	On 为发生异常, Off 为正常	R/W	Yes
	R7	运动控制重置	On 为重置执行，自动清除	R/W	No
	R8	零旗标	On 计算结果为 0	R	No
	R9	借位旗标	On 计算结果发生借位	R	No
	R10	进位旗标	On 计算结果发生进位	R	No
	R13	数据交换加速旗标	On 启动加速, Off 关闭加速	R/W	No
	R14	运动控制功能就绪	On 为就绪, Off 为未就绪	R	No
	R15	运动控制功能启动	On 为启动, Off 为未启动	R/W	No
错误类型旗标	R16	Remote IO 错误	On 为无法建立联机, Off 为联机建立	R	No
	R17	DMCNet 通讯错误	On 为无法建立联机, Off 为联机建立	R	No
	R18	文法错误	On 为运转发生文法错误需自行清除	R/W	No

	R19	运动控制错误	On 为运动控制异常, Off 为正常	R	No
	R20	指令错误	On 为运转发生指令错误, 需自行清除	R/W	No
	R22	伺服参数同步错误	On 为同步伺服重要参数时发生错误, 需自行清除	R/W	No
32 位计数模式设定	R32	C200 计数模式设定	On 为下数, Off 为上数	R/W	Yes
	R33	C201 计数模式设定	On 为下数, Off 为上数	R/W	Yes
	~	~	~	~	~
	R86	C254 计数模式设定	On 为下数, Off 为上数	R/W	Yes
	R87	C255 计数模式设定	On 为下数, Off 为上数	R/W	Yes
Remote IO 模块联机状态	R96	0 号站联机状态	On 为在线, Off 为脱机	R	No
	R97	1 号站联机状态	On 为在线, Off 为脱机	R	No
	~	~	~	~	~
	R126	30 号站联机状态	On 为在线, Off 为脱机	R	No
	R127	31 号站联机状态	On 为在线, Off 为脱机	R	No
PLC 特殊旗标	R139	EMS 按钮状态	On 为按压, Off 为解除	R	No
	R140	限动开关状态	On 为按压, Off 为解除	R	No
	R144	手轮控制开关	On 为启动, Off 为解除	R/W	No
记录启动旗标	R498	循环记录关闭	On 为关闭循环记录动作, Off 为启动循环记录动作	R/W	No
	R499	记录输出档案启动	On 为启动输出档案动作, Off 为关闭输出档案动作	R/W	No
	R500	当前记录断电保持启动	On 为启动记录保持动作, Off 为关闭记录保持动作	R/W	No

- 常闭接点(R0)
  - 定义

在控制器 RUN 中常时 On 接点, 即常闭接点(B 接点/ NC).

- 常开接点(R1)
  - 定义

在控制器 RUN 中常时 Off 接点, 即常开接点(A 接点/ NO).

- 错误总旗标(R4)
  - 定义

    任一错误类型旗标启动均会导致此旗标启动(On)，异常解除后需自行清除此旗标.
  - 关联装置

    【Remote I/O 错误】(R16)，【DMCNet 通讯错误】(R17)，【文法错误】(R18)，【运动控制错误】(R19)或【指令错误】(R20)启动均导致此旗标 On.

- 运动控制重置(R7)
  - 定义

    可启动此旗标(On) 用以重置系统运动控制异常，当异常清除执行完成后此旗标自动清除为 Off.

- 零旗标(R8)
  - 定义

    当运算指令执行后，运算结果为 0 时，此旗标为 On.

- 借位旗标(R9)
  - 定义

    当运算指令执行后, 16 位指令运算结果小于-32,768 或 32 位指令运算结果小于 -2,147,483,648 时，此旗标为 On.

- 进位旗标(R10)
  - 定义

    当运算指令执行后, 16 位指令运算结果大于 32,767 或 32 位指令运算结果大于 2,147,483,647 时，此旗标为 On.

- 数据交换加速旗标(R13)
  - 定义

    此旗标设 On 时代表系统会以更多运行时间处理与人机部分的通讯，而这将使人机能更实时地显示伺服状态与加速人机接口与系统的数据交换动作. 当此旗标设为 Off 为关闭此加速功能.

- 运动控制功能就绪(R14)
  - 定义

    DMCNet 网络中节点的联机建立完成与否, On 代表此网络中各节点联机已建立完成.

- 运动控制功能启动(R15)
  - 定义

    DMCNet 网络节点间底层建立完成与否, On 代表各节点间网络底层联机建立完成. 可重置(Off)以使网络重新建立.

- REMOTE IO错误(R16)
  - 定义

    HMC 连接之 Remote IO 联机是否正常, On 代表联机发生错误.
  - 关联装置

    此旗标启动时异常码显示在【Remote IO 错误代码】(W16). 联机恢复正常后此旗标自动清除.

- DMCNET通讯错误(R17)
  - 定义

    DMCNet 联机是否正常, On 代表联机发生错误. 联机恢复正常后此旗标自动清除.
  - 关联装置

    此旗标启动时异常码显示在【DMCNet 错误代码】(W17).

- 文法错误(R18)
  - 定义

    当执行指令时发生文法异常此旗标 On, 则程序扫描无法继续往下执行, 程序跳离原本异常指令并重头开始扫描. 异常解除后需自行清除此旗标.
  - 关联装置

    此旗标启动时异常码显示在【文法错误代码】(W18).

- 运动控制错误(R19)
  - 定义

    On 代表系统运动控制运算时发生错误. 此异常旗标需以启动【运动控制重置】(R7)清除.
  - 关联装置

    此旗标发生时异常码显示在【运动控制错误代码】(W19).

- 指令错误(R20)
  - 定义

    当指令执行发生异常时此旗标 On, 则程序扫描不会继续往下执行, 程序跳离原本异常指令并从程序开头重新开始扫描. 异常解除后需自行清除此旗标.

- 关联装置

此旗标启动时异常码显示在【指令错误代码】(W20).

- 伺服参数同步错误(R22)
  - 定义

当 DMCNet 联机建立后, HMC 将会执行与伺服参数的初始同步动作. 此外, 每当在 HMC 改变伺服重要参数时也会执行此同步动作. 当发生参数同步动作失败时, 此错误旗标为 On. 此错误旗标需使用者确认后手动清除, 在清除旗标同时, 【伺服参数同步错误码】(W22)也会重置为 0.

  - 关联装置

此旗标发生时的对应异常码显示在【伺服参数同步错误码】(W22).

- C200~ C255 计数模式设定旗标(R32~ R87)
  - 定义

旗标 Off 时决定对应计数器为上数, 旗标 On 时决定对应计数器为下数.

- REMOTE I/O联机状态旗标(R96~ R127)
  - 定义

旗标 On 代表与该站联机正常, 旗标 Off 代表与该站脱机. R96 为第一站联机状态, R97 为第二站联机状态, 以此类推.

- EMS按钮状态(R139)
  - 定义

旗标 On 代表 EMS 按钮被压下, 旗标 Off 代表 EMS 按钮解除.

- 限动开关状态(R140)
  - 定义

旗标 On 代表限动开关处于至能状态, 旗标 Off 代表限动开关处于解除状态.

- 手轮控制开关(R144)
  - 定义

旗标ON代表将以外部手轮转动脉波速度控制当前运动的速度, 在一般运动命令下达前或已经进行的运动中皆可启动此旗标, 启动后运动速度将依据手轮转动速度改变. 旗标OFF代表关闭此功能, 回复正常运动.

  - 关联装置

启动此功能后将根据【手轮转速】(W80)控制进行中运动的速度.

- 循环记录关闭(R498)
  - 定义

旗标 On 代表关闭自动记录功能，旗标 Off 代表开启自动记录功能，开启自动记录功能后，系统将会持续记录当前最新的 50 笔伺服命令与 50 笔伺服状态变化.

用户可透过编辑软件的【检视 HMC 命令历史记录】功能来取得这些记录信息以了解系统的状态履历.

- 记录输出档案启动(R499)
  - 定义

旗标 On 代表系统将每隔 10 秒把当前的历史记录输出成人机记录档案(.des)，最多可记录至 20 个人机记录档案. 旗标 Off 代表关闭此系统功能.

用户可透过人机系统内的【系统设定】➔【档案管理】➔【复制档案】功能将这些人机记录档案导出至外部装置(USB 碟或 SD 卡)后，再使用编辑软件内的【检视 HMC 命令历史记录】功能开启这些人机记录档案(.des)以了解系统的状态履历.

- 当前记录断电保持启动(R500)
  - 定义

旗标 On 代表系统会将当前的 50 笔伺服命令与 50 笔伺服状态变化记录于断电保持数据区中，且此时系统断电重开才能正常运行. 旗标 Off 代表无启动此功能.

用户可透过编辑软件的【装置数据表】功能取得这些断电保持记录信息，并导出成人机记录档案(.dep)后，再使用【检视 HMC 命令历史记录】功能开启这些人机记录档案(.dep)，以了解当时系统的状态履历.

3.3、 PLC系统特殊缓存器

此类缓存器可用以得知系统状态与相关设定，包含版本与控制器系统信息，错误异常代码与周边装置信息等.

分类	编号	功能	说明	属性	停电保持
控制器系统信息	W0	模块编号	DW	R	No
	W2	DSP 韧体编号	DW	R	No
	W4	程序格式版本	DW	R	No
	W7	程序容量	单位：Step	R	No
	W8	执行错误地址	DW，单位：Step	R/W	No

	W10	M 停电保持起始地址	出厂值：512, 设定值需为 16 的倍数.	R/W	Yes
	W11	M 停电保持数量	出厂值：512, 设定值需为 16 的倍数.	R/W	Yes
	W12	D 停电保持起始地址	出厂值：3000	R/W	Yes
	W13	D 停电保持数量	出厂值：1000	R/W	Yes
	W14	伺服参数同步异常动作模式	出厂值：0	R/W	No
错误代码	W16	Remote I/O 错误代码		R	No
	W17	DMCNet 错误代码		R	No
	W18	文法错误代码		R	No
	W19	运动控制错误代码		R	No
	W20	指令错误代码		R	No
	W22	伺服参数同步错误码		R	No
Remote IO 模块版本编号	W32	0 站模块版本编号		R	No
	~	~	~	~	No
	W63	31 站模块版本编号		R	No
FPGA 控制器信息	W66	FPGA 韧体版本		R	No
	W67	FPGA PCB 版本		R	No
其它	W68	时间戳	DW, 单位：0.1 ms	R	No
	W72	指令 Retry 次数	出厂值：0	R/W	Yes
	W73	连续写入伺服参数的时间间隔	出厂值：0	R/W	Yes
	W74	手轮倍率		R/W	Yes
	W75	DMCNet Mask	命令中对 DMCNet 各站的屏蔽设定. 预设为 FFFF, 但若使用 ASDA-M 四轴同动特殊驱动器需设为 FCFF.	R/W	Yes
	W76	手轮计数	DW, 外接式手轮脉波累计计数器	R/W	No
	W80	手轮转速	DW	R	No
	W82	手轮转速取样率	出厂值：10, 单位：1 ms 范围 ：10 ~ 65535	R/W	No

	W83	手轮转速平滑常数	出厂值：10, 范围 ：1 ~ 100	R/W	No

- 模块编号(W0)
  - 定义

    控制器的模块编号，此为 DW 格式.

- DSP韧体编号(W2)
  - 定义

    控制器的 DSP 韧体编号，此为 DW 格式.

- CWP格式版本(W4)
  - 定义

    控制器中程序的格式版本编号，此为 DW 格式.

- 程序容量(W7)
  - 定义

    控制器中程序的 Step 数.

- 执行错误地址(W8)
  - 定义

    当执行发生错误时，程序执行发生错误之 Step 地址.
  - 关联装置

    伴随【文法错误】(R18)或【指令错误】(R20)发生.

- M停电保持起始地址(W10)
  - 定义

    控制器中继电器 M 停电保持开始地址设定，此设定值需为 16 的倍数.

- M停电保持数量(W11)
  - 定义

    控制器中继电器 M 停电保持数量设定，此设定值需为 16 的倍数.

- D停电保持起始地址(W12)
  - 定义

控制器中缓存器 D 停电保持开始地址设定.

- D停电保持数量(W13)
  - 定义

控制器中缓存器 D 停电保持数量设定.

- 伺服参数同步异常动作模式(W14)
  - 定义

当发生伺服参数同步异常时，【伺服参数同步错误】(R22)将会置 On，且发生此错误的参数地址将会显示在【伺服参数同步错误码】(W22)中. 一旦系统发生此错误，将会依照此动作模式设定采取对应的异常处置动作. 模式代码对应：

模式码	定义
00	系统不执行任何处置. 仅以【伺服参数同步错误】(R22)旗标通知.
01	使用者未确认异常前(需手动清除【伺服参数同步错误】(R22)旗标，置为 Off)，系统将禁止执行任何运动动作，包含寸动与手轮操作.
02	系统将执行 DMCNet 重新联机动作，此动作将重复执行直到参数同步动作成功为止. 当执行 DMCNet 重新联机时在伺服面板上将会看到【Co-Ld】字样显示. 请注意，当 HMC 中的伺服参数写入范围错误而导致同步失败状况时，是无法在此模式下以人机画面组件或在线监控(OnLine Monitoring)方式改变错误参数的，需先将模式切回 0 或 1 以改变错误值.

- REMOTE I/O错误代码(W16)
  - 定义

与 Remote I/O 联机异常时错误代码. 异常码对应：

异常码	定义
01	未侦测到任何 Remote I/O 装置
03	已侦测到的 Remote I/O 装置断线

  - 关联装置

伴随【Remote IO 错误】(R16)启动.

- DMCNET错误代码(W17)
  - 定义

    DMCNet 底层联机发生异常时错误代码. 异常码对应：

异常码	定义
01	无法初使化 DMCNet 硬件.
02	无法启动 DMCNet 硬件.
03	无法初使化 DMCNet 联机.
04	DMCNet 通讯底层异常.
05	DMCNet 装置通讯异常

  - 关联装置

    伴随【DMCNet 通讯错误】(R17)启动.

- 文法错误代码(W18)
  - 定义

    运转时发生程序文法执行异常时错误代码. 异常码对应：

异常码	定义
05	FOR ~ NEXT 回路层数超过 5 层
06	CALL 子程序层数超过 8 层
07	浮点数格式出现错误
10	装置使用超过范围
11	RSVP, WSVP 指令执行结果异常
12	待执行 LAUNCH 运动程序超过 255 个

  - 关联装置

    伴随【文法错误】(R18)启动.

- 运动控制错误代码(W19)
  - 定义

    系统运动控制发生异常时错误代码. 异常码对应：

异常码	定义
03	系统发生严重运动错误，例如伺服发生异常撞机(AL. 30)，失速(AL. 918). 或是启动 HMC 命令数据记录功能
09	系统处理运动指令发生异常

- 关联装置

伴随【运动控制错误】(R19)启动.

- 指令错误代码(W20)
  - 定义

运转时发生指令执行异常时错误代码. 异常码对应：

异常码	定义
01	BCD 指令变换结果超过 0~9,999 或 DBCD 指令变换结果超过 0~99,999,999
02	DIV, DDI, FDIV 指令除数为 0.
03	ENCO 指令数据源没有位为 1
04	BIN/DBIN 指令数据源位数数值超过 0~9
10	FASIN 指令数据源正弦值超出-1.0 ~ 1.0 范围
11	FACOS 指令数据源余弦值超出-1.0 ~ 1.0 范围

- 关联装置

伴随【指令错误】(R20)启动.

- 伺服参数同步错误码(W22)
  - 定义

执行与伺服参数同步动作失败时错误代码，异常码对应：

异常码	定义
1XXXX	DMCNet 联机建立初始阶段, HMC 由伺服将参数读回动作失败. XXXX 代表动作失败的参数编号，例如值若为 0670，则表示为 W670 读回失败

2XXXX	HMC 将参数写入伺服时动作失败，失败动作也包含写入参数值超出伺服可接受范围. XXXX 代表动作失败的参数编号，例如值若为 0670，则表示为 W670 写入失败

- 关联装置

伴随【伺服参数同步错误】(R22)启动，清除此异常旗标则【伺服参数同步错误码】将同时被置为 0.

- REMOTE IO模块版本编号(W32~ W63)
  - 定义

对应 Remote IO 各站号之模块版本编号.

- FPGA韧体版本(W66)
  - 定义

HMC 之 FPGA 使用韧体版本编号.

- FPGA PCB版本(W67)
  - 定义

HMC 之 FPGA PCB 版本编号.

- 时间戳(W68)
  - 定义

系统时间戳，单位为 0.1ms，为 DW 格式.

- 指令RETRY次数(W72)
  - 定义

HMC 对伺服写入指令参数时通讯次数设定. 默认值为 0 时，系统设定每一动作指令参数均对伺服写入 2 次，但使用者可增加此写入次数设定以更加提升通讯质量. 若假设此值设定为 3，则动作指令参数均会对伺服写入 5 次，以此类推.

- 伺服参数RETRY次数(W73)
  - 定义

HMC 对属性为 Remote 之运动模式特殊继电器写入伺服时的通讯次数设定. 默认值为 0 时，系统设定此类参数对伺服会写入 2 次，但使用者可增加此写入次数设定以更加确保参数写入正确性. 若假设此值设定为 3，则此类参数会对伺服写入 5 次，以此类推.

- 手轮倍率(W74)
  - 定义

    实体手轮接收脉波与实际转换给驱动器之放大倍率设定. 透过此倍率设定可将手轮脉波放大以符合实际使用需求.

    例如, 若使用手轮转动一圈共送出 100 个脉波数, 以 ASDA-M 伺服而言马达旋转一圈需要 1,280,000 个脉波数, 因此若将手轮倍率设定为 12,800 (1,280,000/ 100 = 12,800)则实际上手轮转一圈马达也会跟着转动一圈.

  - 关联装置

    各轴手轮启用旗标分别为轴 1(R608), 轴 2(R609), ..., 轴 12(R619). 每次仅能启动一轴的手轮功能, 同时启动多轴手轮将导致命令错误发生. 此放大倍率设定会影响到【手轮计数】(W76)的累加计算.

- DMCNET MASK(W75)
  - 定义

    控制器与 DMCNet 中各站沟通时的屏蔽设定. 搭配一般的伺服驱动器使用时此值设定为 FFFF, 但在特殊的架构下会有不同的设定值, 如对应 ASDA-M 可支持四轴同动之驱动器时, 此值需设为 FCFF 才可正常运作.

- 手轮计数(W76)
  - 定义

    将实体手轮接收脉波数经倍率放大后转换给驱动器的计数累计值.

  - 关联装置

    手轮接收脉波数经【手轮倍率】(W74)设定值放大后, 【手轮计数】(W76)会一直将这些脉波放大值累加.

- 手轮转速(W80)
  - 定义

    当开启手轮控制功能后, 系统将以此手轮转速度控制实际的伺服运动速度.

  - 关联装置

    将手轮接收脉波先经【手轮倍率】(W74)放大脉波计数后, 再根据【手轮转速取样率】(W82)进行取样, 并参考【手轮转速平滑常数】(W83)最后得出【手轮转速】(W80).

- 手轮转速取样率(W82)
  - 定义

此设定值决定手轮转速计算时的脉波取样周期，设定值越小则对手轮转动越敏感，但容易有速度为 0 状况出现.

■ 关联装置

将手轮接收脉波先经【手轮倍率】(W74)放大脉波计数后，再根据【手轮转速取样率】(W82)进行取样，并参考【手轮转速平滑常数】(W83)最后得出【手轮转速】(W80).

➢ 手轮转速平滑常数(W83)

■ 定义

此设定值决定手轮转速控制时的速度平滑程度，设定值越大动作将会越平滑.

■ 关联装置

将手轮接收脉波先经【手轮倍率】(W74)放大脉波计数后，再根据【手轮转速取样率】(W82)进行取样，并参考【手轮转速平滑常数】(W83)最后得出【手轮转速】(W80).

3.4、 运动模式特殊继电器

HMC 最多同时可在 DMCNet 中控制 12 轴的伺服轴运动，各轴地址对应如下表：

功能	地址对应							
	轴 1	轴 2	轴 3	轴 4	轴 5	轴 6	~	轴 12
伺服轴控制								
命令执行	R512	R513	R514	R515	R516	R517	~	R523
实时停止	R528	R529	R530	R531	R532	R533	~	R539
正向寸动	R544	R545	R546	R547	R548	R549	~	R555
反向寸动	R560	R561	R562	R563	R564	R565	~	R571
Servo On	R576	R577	R578	R579	R580	R581	~	R587
Fault Reset	R592	R593	R594	R595	R596	R597	~	R603
启用手轮	R608	R609	R610	R611	R612	R613	~	R619
命令预载	R624	R625	R626	R627	R628	R629	~	R635
取消预载	R640	R641	R642	R643	R644	R645	~	R651
Feed Rate 执行	R656	R657	R658	R659	R660	R661	~	R667
动作暂停	R672	R673	R674	R675	R676	R677	~	R683
伺服轴								
命令错误	R1024	R1025	R1026	R1027	R1028	R1029	~	R1035
命令就绪	R1040	R1041	R1042	R1043	R1044	R1045	~	R1051
命令完成	R1056	R1057	R1058	R1059	R1060	R1061	~	R1067
SERVO ON	R1072	R1073	R1074	R1075	R1076	R1077	~	R1083
伺服 Quick	R1088	R1089	R1090	R1091	R1092	R1093	~	R1099

Stop 解除								
伺服 Fault	R1104	R1105	R1106	R1107	R1108	R1109	~	R1115
伺服 Warning	R1120	R1121	R1122	R1123	R1124	R1125	~	R1131
伺服就绪	R1136	R1137	R1138	R1139	R1140	R1141	~	R1147

运动模式控制继电器

下表为运动模式控制继电器，这些继电器能达成运动启动或错误旗标清除等功能. 以下以第一轴为例说明:

功能	编号	说明	属性	停电保持
命令执行	R512	On 运动命令执行	R/W	No
实时停止	R528	On 为伺服急停, Off 为解除	R/W	No
正向寸动	R544	On 为正向寸动启动, Off 为解除	R/W	No
反向寸动	R560	On 为反向寸动启动, Off 为解除	R/W	No
Servo On	R576	On 为 Servo On, Off 为 Servo Off	R/W	No
Fault Reset	R592	On 为伺服错误旗标清除启动	R/W	No
启用手轮	R608	On 为启动手轮, Off 为关闭手轮	R/W	No
命令预载	R624	On 加载连续运动指令并开始执行	R/W	No
取消预载	R640	On 取消连续运动指令	R/W	No
Feed Rate 执行	R656	On 为改变运动中速度，完成后自动清除 Off	R/W	No
动作暂停	R672	On 为运动暂停, Off 为运动回复	R/W	No

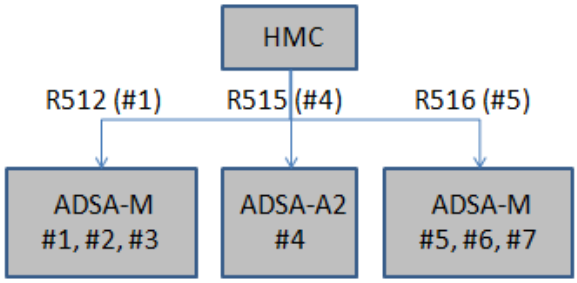
- 命令执行(R512)
  - 定义

    伺服轴命令执行启动旗标. 将【命令执行】旗标由 Off 触发为 On 后, HMC 开始将相关运动参数写至相关运动伺服中，并且在完成写入参数后即开始执行运动.

    当 HMC 控制多台驱动器时，仅需下命令给运动的最前轴. 例如若 HMC 控制一个五轴伺服设备时，其中五轴分别为轴 1，轴 2，轴 3，轴 4 与轴 5. 若执行五轴的直线同动运动时，仅需对最前面的运动轴下达命令与触发命令执行即可，也就是触发轴 1 的【命令执行】(R512)由 Off 触发为 On，同时透过设定【命令选项】(W513 设为 31，其中 W513 的 Bit0 On 代表轴 1 运动, Bit1 On 代表轴 2 运动，以此类推...)来决定运动之相关轴即可.

    又例如当要执行轴 2，轴 3，轴 4 的三轴直线同动运动时，仅需下达命令与触发命令执行给第 2 轴即可，也就是触发轴 2 的【命令执行】(R513)由 Off 触发为 On，并设定【命令选项】(W769 设为 7，其中 W769 的 Bit0 On 代表轴 2 运动, Bit1 On 代表轴 3 运动, Bit3 On 代表轴 4 运动)即可.

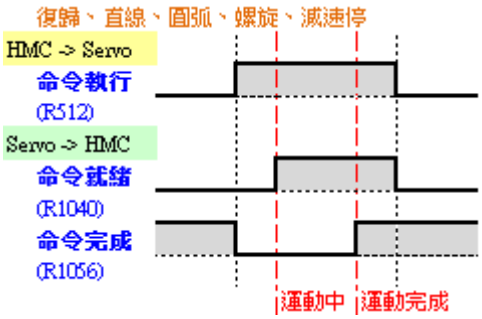
HMC 除了多轴直线运动外，若要执行圆弧或螺旋等运动，需对 ASDA-M 驱动器的第一轴下达命令才能达成. 如下图的架构中, HMC 仅能启动轴 1 的【命令执行】(R512)来执行轴 1，轴 2 与轴 3 间的圆弧或螺旋运动，启动轴 5 的【命令执行】(R516)来执行轴 5，轴 6 与轴 7 间的圆弧或螺旋运动.



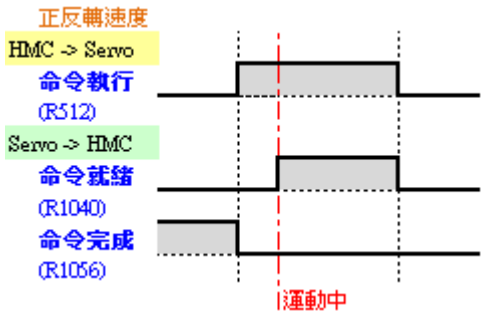
■ 关联装置

在不同运动中，当运动结束后【命令完成】旗标会有不同状态的现象，因此建议程序流程控制可遵照以下的交握方式来实现命令下达控制.

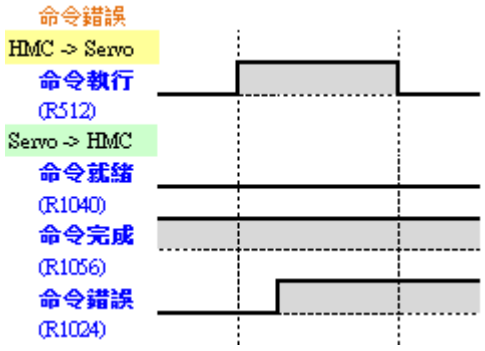
以轴 1 为例，在复归，直线，寸动等运动中，启动命令需先填入运动命令与正确的对应运动参数，如【命令代码】(W512)，【命令选项】(W513)，【速度设定】(W518)，【目标位置】(W520)等设定正确后，接着将【命令执行】(R512)由 Off 触发为 On. 当命令已写入伺服中则【命令就绪】(R1040)状态会成为 On，接着即开始执行运动命令. 最后当运动结束后【命令完成】(R1056)变为 On 代表此运动命令完成. 此类命令动作程序流程控制可遵照下图的交握方式来实现命令下达控制.



在速度命令中，下达【命令执行】后接着【命令就绪】会成为 On 代表命令已下达至驱动器完成并执行，但驱动器执行此类速度命令并不会使【命令完成】变为 On，因此不需等待【命令完成】旗标. 此类命令动作程序流程控制可遵照下图的交握方式来实现命令下达控制.



当下达命令参数错误或伺服状态异常无法接受命令时，将导致【命令执行】(R512)触发失效同时【命令错误】旗标也会变为 On.



■ 异常状况

以下情形可能导致【命令执行】启动后发生【命令错误】并导致无动作，对应的错误码会显示在各轴的【错误代码】(W576, ...)中. 错误码对应如下:

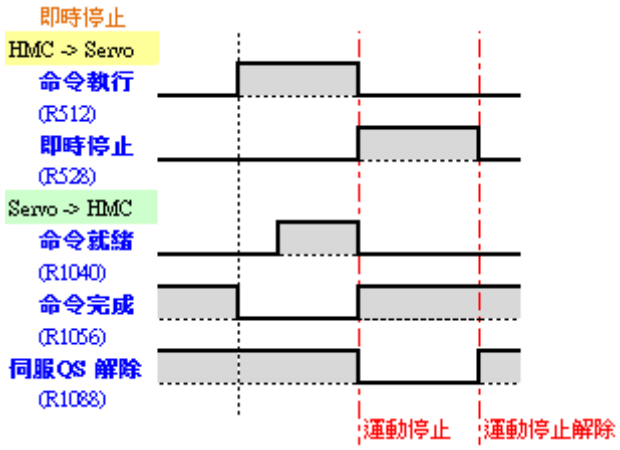
异常码	定义
01	运动速度参数为零
02	服务器紧急停止
03	服务器 Servo-Off
04	命令执行中
05	连续路径运动指令错误
06	螺旋或圆弧运动指令错误
07	命令代码错误
08	下达连续命令时超过最大连续指令数目
09	下达连续命令超时
10	运动命令无法使用于连续路径动作中
11	指定轴速错误

12	多轴同动暂停时间过长
13	多轴同动暂停时间为零
14	运动模式切换错误
15	更新伺服参数错误
16	伺服参数同步错误

- 实时停止(R528)
  - 定义

    伺服轴【实时停止】旗标由 Off 切至 On，若该轴在运动中则终止该动作并急停.
  - 关联装置

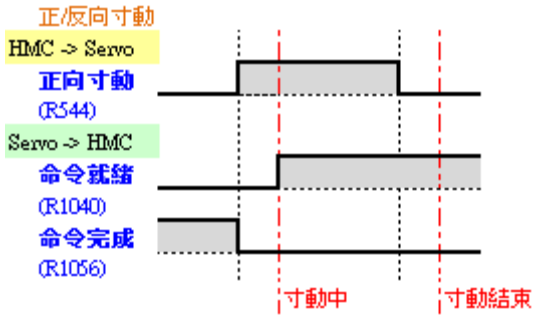
    以轴 1 为例，【实时停止】(R528)启动 On 后，伺服若在运动中，则会根据【Quick Stop 减速时间】(W670)执行急停运动，当急停完成后【伺服 Quick Stop 解除】(R1088)状态会由 On 变为 Off 代表此轴目前状态为 Quick Stop. 当【实时停止】(R528)清除为 Off 后，【伺服 Quick Stop 解除】(R1088)状态会由 Off 变为 On 代表此轴目前已解除实时停止状态.



- 正向寸动(R544)
  - 定义

    将伺服轴【正向寸动】旗标 On，该轴执行正向寸动，将旗标 Off 后寸动动作停止.
  - 关联装置

    以轴 1 为例，【正向寸动】(R544)切为 On 后，该轴依据【寸动加速时间】(W680)曲线加速至【寸动速度】(W678)，并以【寸动速度】(W678)维持正向等速运动. 当【正向寸动】(R544)切换至 Off 时，该轴依据【寸动减速时间】(W681)曲线减速至停止. 另外可搭配设定【寸动扭力限制】(W682)实现寸动运动的扭力保护功能.



■ 异常状况

以下情形可能导致启动旗标后却无动作：

a. 伺服轴不在 Servo On 状态.

b. 寸动速度(W678)大于最大速度限制(W660)设定.

c. 伺服轴处于 Quick Stop 状态.

d. 手轮功能已被启动.

e. 寸动速度(W678)设 0.

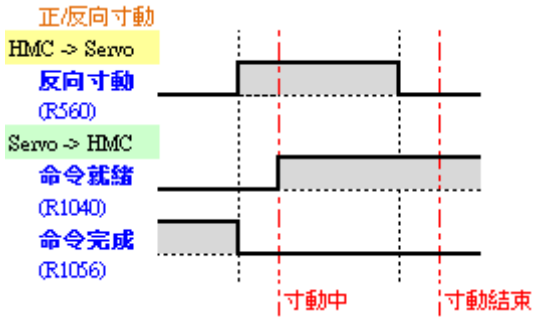
➢ 反向寸动(R560)

■ 定义

将伺服轴【反向寸动】旗标 On，该轴执行反向寸动，将旗标 Off 后寸动动作停止.

■ 关联装置

以轴 1 为例，【反向寸动】(R560)切为 On 后，该轴依据【寸动加速时间】(W680)曲线加速至【寸动速度】(W678)，并以【寸动速度】(W678)维持反向等速运动. 当【反向寸动】(R560)切换至 Off 时，该轴依据【寸动减速时间】(W681)曲线减速至停止. 另可搭配设定【寸动扭力限制】(W682)实现寸动运动的扭力保护功能.



■ 异常状况

以下情形可能导致启动旗标后伺服却无动作：

a. 伺服轴不在 Servo On 状态.

b. 寸动速度(W678)大于最大速度限制(W660)设定.

c. 伺服轴处于 Quick Stop 状态.

d. 手轮功能已被启动.

e. 寸动速度(W678)设 0.

- SERVO ON(R576)
  - 定义

将伺服轴【Servo On】旗标设为 On，则该轴即 Servo On，将旗标 Off 后该轴即 Servo Off.

  - 关联装置

以轴 1 为例，将【Servo On】旗标(R576)切为 On 后，该轴即执行 Servo On 动作，对应【SERVO ON】(R1072)同时会显示 On. 将【Servo On】旗标(R576)切为 Off 后，该轴即执行 Servo Off 动作，对应【SERVO ON】(R1072)同时显示 Off 状态.

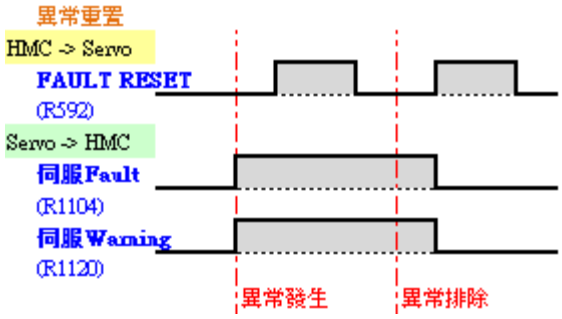
  - 异常状况

以下情形可能导致启动旗标后伺服却无正确动作：

a. DMCNet 联机异常，可由 DMCNet 通讯错误(R17)或运动控制错误(R19)确认.

- FAULT RESET旗标(R592)
  - 定义

伺服轴发生错误时，将旗标由 Off 切换至 On，即启动伺服执行错误清除动作. 使用者须自行清除.

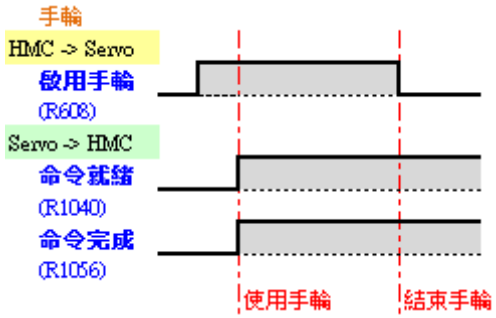


- 启用手轮(R608)
  - 定义

伺服轴将该旗标 On 后即可启动手轮. 同时间仅能启动一轴的手轮功能，启动多轴手轮将导致发生命令错误.

  - 关联装置

启动手轮功能后，控制器会根据【手轮倍率】(W74)将实际手轮送来的脉波数乘上此倍率转换后下达给伺服转动目标命令，此目标位置将会限定在【手轮反向软件极限】(W690)与【手轮正向软件极限】(W692)间.



- 异常状况

以下情形可能导致启动旗标后却无动作：

a. 伺服轴不在 Servo On 状态.

b. 其它轴手轮功能已被启动.

➢ 命令预载(R624)

- 定义

【命令预载】为连续运动命令的加载与运动执行启动旗标. 将【命令预载】旗标由 Off 切至 On, HMC 开始将相关运动参数写入至伺服(预载命令)，当写入两笔运动数据完成后 HMC 即开始执行连续运动. 在当前的连续运动尚未结束前(至少还剩两笔以上运动仍在执行)，可紧接着再加载其它运动命令，这些运动命令会持续写入伺服，并且会依序执行，如此能达成连续不间断的运动行为.

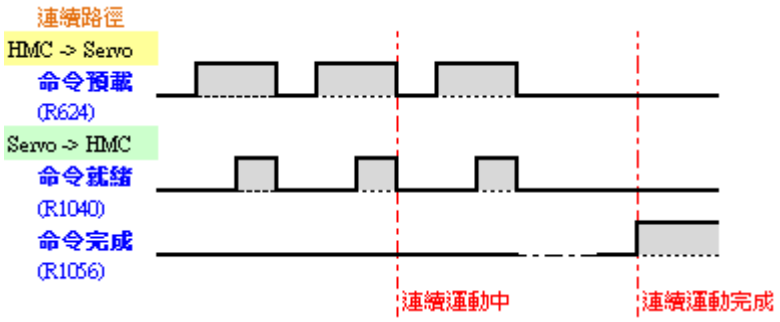
在连续运动中能下达多轴直线同动或圆弧等不同的运动，只需写入不同命令代码与参数即可，且透过【Overlap】设定可以开启路径间的迭合功能，当两段路径执行迭合动作时，能够使运动路径更加地平滑快速.

连续运动中每一段运动命令下达方式与单动命令相同，差别在于单动是触发【命令执行】(R512)且需等待【命令完成】(R1056)后才能再执行下一个运动命令. 在连续运动中，命令的下达与执行都是触发【命令预载】(R624)，在启动【命令预载】后需等待【命令就绪】(R1040)成为 On，才代表命令已成功加载伺服，接着才能再下达下一个动作命令. 当所有连续路径动作都完成后【命令完成】(R1056)才会成为 On，代表连续动作完成并结束.

需特别注意的是在连续路径运动中，当已经执行至最后一段路径时，就无法再接受新的命令加载动作了.

- 关联装置

以轴 1 为例，下达连续命令动作时，首先写入运动命令与参数，如【命令代码】(W512)，【命令选项】(W513)，【速度设定】(W518)，【目标位置】(W520)，【Overlap】(W525)等，接着触发【命令预载】(R624)为 On 以将命令参数写入伺服，并需等待【命令就绪】(R1040)状态变为 On 才代表命令已成功写入伺服中. 接着须再载入第二笔运动命令完成后，此时才会开始运动. 在运动尚未执行至最后一段动作时，可继续不断地加载新的运动命令，而这些加载的运动将会依序地执行. 当所有载入的运动都完成后【命令完成】(R1056)将变为 On 表示连续运动结束，相关旗标交握方式如下:



HMC 最多只能先预载 8 笔运动路径命令，可经由【路径剩余数】(W594)得知当前尚有几个命令预载于伺服中. 当命令预载空间为满时(【路径剩余数】(W594)等于 8)，这时再下达新的【命令预载】，【命令就绪】是不会成为 On 的，也就是这笔新的运动命令尚无法写入至伺服. 必须现有路径已完成，而此时【路径剩余数】(W594)会少于 8，才能再继续接受新的【命令预载】触发并写入新的命令至伺服中.

当连续路径执行中，若剩下最后一个运动在执行(W594 = 1)，则此时无法再接受新的预载命令，需等到目前这一段的连续运动完成或中止后才能重新下达新的运动命令. 当预载错误的命令时，则预载命令动作失败，此命令不会写至伺服中且【路径剩余数】也不会增加.

■ 异常状况

以下情形可能导致命令错误:

异常码	定义
01	运动速度参数为零
02	服务器紧急停止
03	服务器 Servo-Off
04	命令执行中
05	连续路径运动指令错误
06	螺旋或圆弧运动指令错误
07	命令代码错误

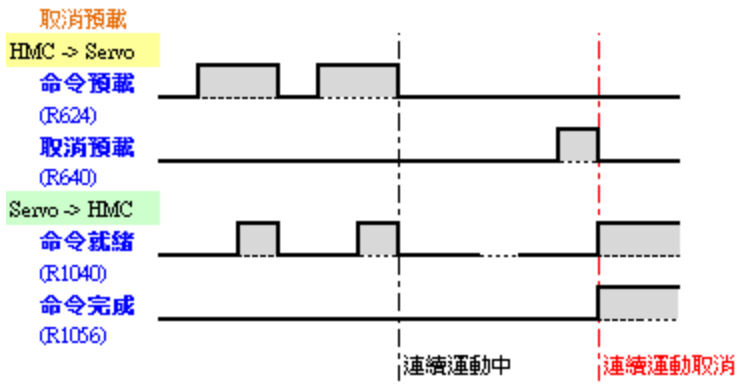
08	下达连续命令时超过最大连续指令数目
09	下达连续命令超时
10	运动命令无法使用于连续路径动作中
11	指定轴速错误
12	多轴同动暂停时间过长
13	多轴同动暂停时间为零
14	运动模式切换错误
15	更新伺服参数错误
16	伺服参数同步错误

- 取消预载(R640)
  - 定义

当执行连续路径动作时，触发【取消预载】(R640)成为 On 后，会取消所有预载中的连续运动路径数据，并且立即停止运动.

  - 关联装置

以轴 1 为例，当执行连续路径运动中，若触发【命令取消】为 On，则运动会停止并结束此连续运动. 当运动停止后【命令就绪】(R1040)与【命令完成】(R1056)会变为 On，相关旗标交握方式如下:



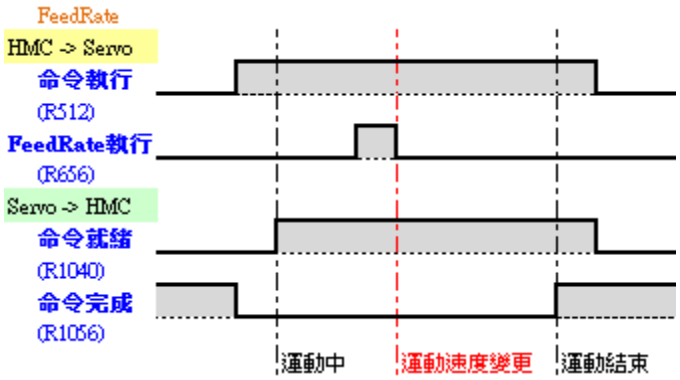
- FEED RATE执行(R656)
  - 定义

运动中将【Feed Rate 执行】触发为 On 后，会根据触发轴的【Feed Rate 速度】，【Feed Rate 加速时间】与【Feed Rate 减速时间】等设定改变目前运动中的速度相关设定. 当改变 Feed Rate 相关速度动作完成后此旗标会自动回复为 Off.

在连续路径运动中，若改变 Feed Rate 则只会改变当前的运动，当此段运动结束并进入下一路径运动时，会依据原本路径设定速度执行运动，也就是说 Feed Rate 改变速度只对当前运动路径有效.

- 关联装置

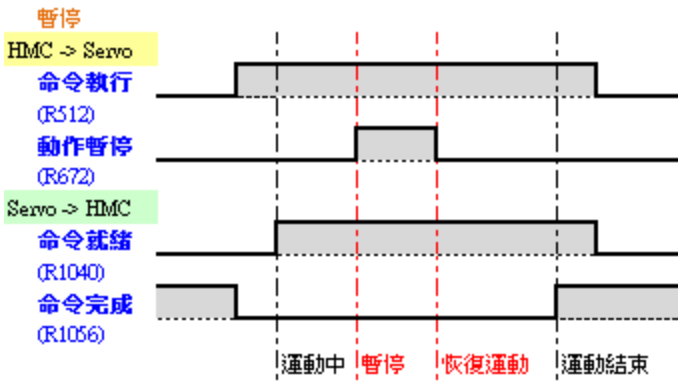
以轴 1 为例，若触发【Feed Rate 执行】(R656)为 On，则会将目前运动之速度改变为【Feed Rate 速度】(W684)，加速度时间改变为【Feed Rate 加速度时间】(W686)，减速度时间改变为【Feed Rate 减速度时间】(W687). 改变完成后【Feed Rate 执行】(R656)自动清除，同时此动作不会影响原本命令下达的时序.



➢ 动作暂停(R672)

- 定义

运动中将【动作暂停】设为 On，则当前动作停止. 运动停止时再将【动作暂停】设为 Off，则回复原本运动继续动作.



运动模式状态继电器

运动模式状态继电器，反应目前伺服异常与运动功能状态. 以下以第一轴为例说明:

功能	编号	说明	属性	停电保持
命令错误	R1024	On 为下达运动指令时异常 使用者需自行清除	R/W	No
命令就绪	R1040	On 为动作命令已下达至伺服	R	No
命令完成	R1056	On 为伺服已完成命令动作	R	No

SERVO ON	R1072	On 为伺服在 Servo On 状态	R	No
伺服 Quick Stop 解除	R1088	On 为伺服解除 Quick Stop 状态 Off 为伺服处于 Quick Stop 状态	R	No
伺服 Fault	R1104	On 为伺服发生错误	R	No
伺服 Warning	R1120	On 为伺服发生警告	R	No
伺服就绪	R1136	On 为与伺服间 DMCNet 建立完成	R	No

- 命令错误(R1024)
  - 定义

当对伺服轴下达命令时，若命令参数错误或伺服状态异常，导致下达命令无效时，该旗标会变为 On. 一旦命令发生错误，此旗标被设为 On 后，需使用者自行将此旗标清除为 Off. 此旗标仅反应曾发生的命令错误状态，若没有清除并不影响至下一个命令的下达.

  - 关联装置

以轴 1 为例，下达【命令执行】(R512)或【命令预载】(R624)，当命令错误发生时【命令错误】(R1024)成为 On，同时【错误代码】(W576)会伴随显示错误原因，错误代码如下:

异常码	定义
01	运动速度参数为零
02	服务器紧急停止
03	服务器 Servo-Off
04	命令执行中
05	连续路径运动指令错误
06	螺旋或圆弧运动指令错误
07	命令代码错误
08	下达连续命令时超过最大连续指令数目
09	下达连续命令超时
10	运动命令无法使用于连续路径动作中
11	指定轴速错误
12	多轴同动暂停时间过长
13	多轴同动暂停时间为零

14	运动模式切换错误
15	更新伺服参数错误
16	伺服参数同步错误

- 命令就绪(R1040)
  - 定义

    对伺服轴下达动作命令时，控制器将命令参数透过 DMCNET 通讯写至对应的驱动器中，并在参数写入完成后【命令就绪】会变为 On.

    若命令错误发生时，代表命令下达失败，则【命令就绪】并不会变为 On,【命令就绪】是命令下达成功与否的重要参考旗标.
  - 关联装置

    以轴 1 为例，当【命令执行】(R512)触发为 On 时，则【命令就绪】(R1040)先自动清除. 而当【命令执行】(R512)清除为 Off 时，【命令就绪】(R1040)也会跟着自动清除. 详细使用的时序图请参考【运动模式控制继电器】章节中【命令执行】说明.
- 命令完成(R1056)
  - 定义

    对伺服轴下达动作命令后，【命令完成】会先清除为 Off. 当动作完成结束后，【命令完成】即成为 On.
  - 关联装置

    以轴 1 为例，当【命令执行】(R512)设为 On 时，轴 1 的【命令完成】(R1056)会自动清除为 Off. 详细使用的时序图请参考【运动模式控制继电器】章节中【命令执行】说明.
- SERVO ON(R1072)
  - 定义

    驱动器的伺服状态，当此旗标为 On 时，表示该轴目前为 Servo On 状态. 当此旗标为 Off 时，表示该轴目前为 Servo Off 状态.
  - 关联装置

    以轴 1 为例，当【Servo On】(R576)设为 On 时，代表启动 SERVO ON. 当伺服轴 1 状态确实成为 Servo On 后，【SERVO ON】(R1072)即为 On.
- 伺服QUICK STOP解除(R1088)
  - 定义

伺服轴是否处于急停状态，当此旗标为On代表该轴目前已解除Quick Stop状态，处于可以接受运动命令状态. 但若此旗标为 Off 代表该轴目前尚在急停(Quick Stop)状态中，无法接受命令运动.

- 关联装置

以轴 1 为例，若将【实时停止】(R528)设为 On 时，表示 HMC 下达急停命令给伺服，当伺服急停完成后，【伺服 Quick Stop 解除】(R1088)旗标成为 Off.

将【实时停止】(R528)设为 Off 时，表示 HMC 下达解除急停命令给伺服，若伺服解除急停状态成功，【伺服 Quick Stop 解除】(R1088)旗标成为 On.

另外，若 ASDA 伺服参数中将【数字输入接脚 DI 功能规划】设为马达紧急停止(EMGS)时，会导致驱动器将以 DI 讯号控制伺服急停状态，导致 HMC 控制失效.

- 伺服FAULT(R1104)
  - 定义

对应伺服轴发生 Alarm 异常时此旗标为 On; 当伺服轴 Alarm 解除后，此旗标清除为 Off.

  - 关联装置

以轴 1 为例，伺服轴发生 Alarm 异常可藉由【伺服异常代码】(W585)来得知异常内容. 并可由【Fault Reset】(R592)旗标控制(设 On) 对伺服执行异常清除，重置等动作.

- 伺服WARNING(R1120)
  - 定义

对应伺服轴发生 Warning 异常时此旗标为 On，当 Warning 解除此旗标可自动清除为 Off.

  - 关联装置

以轴 1 为例，伺服轴发生 Warning 异常时可藉由【伺服异常代码】(W585)来得知异常内容. 并可由【Fault Reset】(R592)旗标控制(设 On) 对伺服执行异常清除，重置等动作. .

- 伺服就绪(R1136)
  - 定义

当 HMC 与伺服间完成 DMCNet 联机建立后，对应之伺服轴会将此旗标设为 On 代表联机建立.

3. 5、 运动模式特殊缓存器

在【运动模式特殊缓存器】的属性共有停电保持，可读(R)，可写(W)与 Remote 等，当特殊缓存器具有【停电保持】且【Remote】的属性时，代表当 HMC 与驱动器建立 DMCNet 联机成功后, HMC 会执行一次将这类型的参数设定值写入驱动器的动作. 以【电子齿轮比分子/分母】

为例，当 HMC 与驱动器联机成功后, HMC 立即会将【电子齿轮比分子】(W640)与【电子齿轮比分母】(W642)的设定值写入至驱动器的 P1-44(GR1)与 P1-45(GR2)中. 因此透过如此的运动参数的设定方式, HMC 能达成系统参数的一致性.

HMC 最多可控制 12 轴的伺服轴运动，功能与 DMCNet 各轴的地址对应如下表：

功能	地址对应							
	轴 1	轴 2	轴 3	轴 4	轴 5	轴 6	~	轴 12
命令								
命令代码	W512	W768	W1024	W1280	W1536	W1792	~	W3328
命令选项	W513	W769	W1025	W1281	W1537	W1793	~	W3329
命令模式	W514	W770	W1026	W1282	W1538	W1794	~	W3330
Delay 时间	W515	W771	W1027	W1283	W1539	W1795	~	W3331
加速时间	W516	W772	W1028	W1284	W1540	W1796	~	W3332
减速时间	W517	W773	W1029	W1285	W1541	W1797	~	W3333
速度设定(DW)	W518	W774	W1030	W1286	W1542	W1798	~	W3334
目标位置(DW)	W520	W776	W1032	W1288	W1544	W1800	~	W3336
速度比例	W522	W778	W1034	W1290	W1546	W1802	~	W3338
参数起始地址	W524	W780	W1036	W1292	W1548	W1804	~	W3340
Overlap	W525	W781	W1037	W1293	W1549	W1805	~	W3341
速度选项	W526	W782	W1038	W1294	W1550	W1806	~	W3342
状态								
错误代码	W576	W832	W1088	W1344	W1600	W1856	~	W3392
当前位置(DW)	W578	W834	W1090	W1346	W1602	W1858	~	W3394
平均转矩(DW)	W580	W836	W1092	W1348	W1604	W1860	~	W3396
当前速度(DW)	W582	W838	W1094	W1350	W1606	W1862	~	W3398
伺服异常代码	W585	W841	W1097	W1353	W1609	W1865	~	W3401
监控项目 1(DW)	W586	W842	W1098	W1354	W1610	W1866	~	W3402
监控项目 2(DW)	W588	W844	W1100	W1356	W1612	W1868	~	W3404
监控项目 3(DW)	W590	W846	W1102	W1358	W1614	W1870	~	W3406
监控项目 4(DW)	W592	W848	W1104	W1360	W1616	W1872	~	W3408
路径剩余数	W594	W850	W1106	W1362	W1618	W1874	~	W3410
DMCNet 通讯错误数 A	W596	W852	W1108	W1364	W1620	W1876	~	W3412
DMCNet 通讯错误数 B	W598	W854	W1110	W1366	W1622	W1878	~	W3414
高速监控项目(DW)	W600	W856	W1112	W1368	W1624	W1880	~	W3416
当前转速	W602	W858	W1114	W1370	W1626	W1882	~	W3418
运动参数								
电子齿轮比分子	W640	W896	W1152	W1408	W1664	W1920	~	W3456

(DW)								
电子齿轮比分母 (DW)	W642	W898	W1154	W1410	W1666	W1922	~	W3458
单位显示	W644	W900	W1156	W1412	W1668	W1924	~	W3460
加减速曲线	W645	W901	W1157	W1413	W1669	W1925	~	W3461
加速时间	W646	W902	W1158	W1414	W1670	W1926	~	W3462
减速时间	W647	W903	W1159	W1415	W1671	W1927	~	W3463
原点复归速度 1(DW)	W648	W904	W1160	W1416	W1672	W1928	~	W3464
原点复归速度 2(DW)	W650	W906	W1162	W1418	W1674	W1930	~	W3466
原点复归模式	W652	W908	W1164	W1420	W1676	W1932	~	W3468
原点复归加减速时间	W653	W909	W1165	W1421	W1677	W1933	~	W3469
原点复归偏移值 (DW)	W654	W910	W1166	W1422	W1678	W1934	~	W3470
正向软件极限(DW)	W656	W912	W1168	W1424	W1680	W1936	~	W3472
反向软件极限(DW)	W658	W914	W1170	W1426	W1682	W1938	~	W3474
最大速度限制(DW)	W660	W916	W1172	W1428	W1684	W1940	~	W3476
监控项目索引 1	W666	W922	W1178	W1434	W1690	W1946	~	W3482
监控项目索引 2	W667	W923	W1179	W1435	W1691	W1947	~	W3483
监控项目索引 3	W668	W924	W1180	W1436	W1692	W1948	~	W3484
监控项目索引 4	W669	W925	W1181	W1437	W1693	W1949	~	W3485
Quick Stop 减速时间	W670	W926	W1182	W1438	W1694	W1950	~	W3486
停止命令减速时间	W671	W927	W1183	W1439	W1695	W1951	~	W3487
通讯错误减速时间	W672	W928	W1184	W1440	W1696	W1952	~	W3488
马达过载减速时间	W673	W929	W1185	W1441	W1697	W1953	~	W3489
反向软件极限减速时间	W674	W930	W1186	W1442	W1698	W1954	~	W3490
正向软件极限减速时间	W675	W931	W1187	W1443	W1699	W1955	~	W3491
反向硬件极限减速时间	W676	W932	W1188	W1444	W1700	W1956	~	W3492
正向硬件极限减速时间	W677	W933	W1189	W1445	W1701	W1957	~	W3493
寸动速度(DW)	W678	W934	W1190	W1446	W1702	W1958	~	W3494
寸动加速时间	W680	W936	W1192	W1448	W1704	W1960	~	W3496
寸动减速时间	W681	W937	W1193	W1449	W1705	W1961	~	W3497
寸动扭力限制	W682	W938	W1194	W1450	W1706	W1962	~	W3498
FeedRate 速度(DW)	W684	W940	W1196	W1452	W1708	W1964	~	W3500

FeedRate 加速时间	W686	W942	W1198	W1454	W1710	W1966	~	W3502
FeedRate 减速时间	W687	W943	W1199	W1455	W1711	W1967	~	W3503
高速监控项目索引	W688	W944	W1200	W1456	W1712	W1968	~	W3504
最大转速限制	W689	W945	W1201	W1457	W1713	W1969	~	W3505
手轮反向软件极限(DW)	W690	W946	W1202	W1458	W1714	W1970	~	W3506
手轮正向软件极限(DW)	W692	W948	W1204	W1460	W1716	W1972	~	W3508
伺服参数								
自动低频抑振模式设定	W704	W960	W1216	W1472	W1728	W1984	~	W3520
对伺服的负载惯量比与负载重量比	W705	W961	W1217	W1473	W1729	W1985	~	W3521
位置控制比例增益	W706	W962	W1218	W1474	W1730	W1986	~	W3522
位置控制前馈增益	W707	W963	W1219	W1475	W1731	W1987	~	W3523
速度控制增益	W708	W964	W1220	W1476	W1732	W1988	~	W3524
速度积分补偿	W709	W965	W1221	W1477	W1733	W1989	~	W3525
共振抑制低通滤波	W710	W966	W1222	W1478	W1734	W1990	~	W3526
外部干扰抵抗增益	W711	W967	W1223	W1479	W1735	W1991	~	W3527
速度检测滤波及微振抑制	W712	W968	W1224	W1480	W1736	W1992	~	W3528
位置控制误差过大条件(DW)	W724	W980	W1236	W1492	W1748	W2004	~	W3540
电子凸轮的曲线表格倍率设定	W726	W982	W1238	W1494	W1750	W2006	~	W3542
E－CAM：Master 齿轮比设定 P	W728	W984	W1240	W1496	W1752	W2008	~	W3544
E－CAM：凸轮启动控制	W730	W986	W1242	W1498	W1754	W2010	~	W3546
E－CAM：脱离时机数据	W732	W988	W1244	W1500	W1756	W2012	~	W3548

命令缓存器

下达伺服运动命令功能. 以下以第一轴为例说明:

功能	编号	说明	属性	停电保持	出厂值
命令代码	W512	运动命令类型	R/W	No	0
命令选项	W513	命令代码所需的额外信息	R/W	No	0
命令模式	W514	定位命令数据模式	R/W	No	0
Delay 时间	W515	定位完成停留时间, 单位为 ms	R/W	No	0
加速时间	W516	伺服轴运动的加速时间	R/W	No	0

减速时间	W517	伺服轴运动的减速时间	R/W	No	0
速度设定(DW)	W518	单位为 PUU/s	R/W	No	0
目标位置(DW)	W520	单位为 PUU	R/W	No	0
速度比例	W522	实际运动之速度百分比	R/W	No	0
参数起始地址	W524	运动参数数据之读取起始 D 装置地址	R/W	No	0
Overlap	W525	路径重迭百分率	R/W	No	0
速度选项	W526	直线运动时速度使用模式	R/W	No	0

- 命令代码(W512)
  - 定义

依据不同动作需求，下达对应的运动命令代码给驱动器，目前支持以下的动作代码：

0：无动作

1：直线同动

4：正转速度

5：反转速度

6：减速停止

8：复归动作

10：圆弧: 半径&角度模式

11：圆弧: 中点&终点模式

12：圆弧: 圆心&终点模式

13：圆弧: 终点&半径模式

14：圆弧: 圆心&角度模式

24：四轴直线同动 (特殊型驱动器)

30：螺旋

31：螺旋 W

  - 关联装置

HMC 下达运动命令给驱动器时，需将动作命令写在触发轴的【命令代码】上. 例如 ASDA-M 三轴驱动器分别为轴 1，轴 2 与轴 3，若要执行三轴的运动，则需写至轴 1 的【命令代码】(W512)，【命令选项】(W513)，【速度设定】(W518)与【目标位置】(W520)等相关参数，最

后再将轴 1 的【命令执行】(R512) 触发为 On 以启动伺服开始运动. 在第五章的运动运行范例中会有更详细介绍.

- 命令选项(W513)
  - 定义

不同的【命令代码】中有对应的【命令选项】设定方式, 需在触发轴填入此【命令选项】设定.

在直线运动与速度运动中, 【命令选项】为对应启动的轴选项, 由低位至高位依序代表以触发轴为最前轴, 依序的各轴是否需启动运动. 例如若触发轴 1【命令执行】(R512), 则【命令选项】(W513)的位 0 为 On 代表轴 1 伺服需启动运动, 位 1 为 On 代表轴 2 伺服需启动运动, 以此类推. 但若触发轴 3【命令执行】(R514), 则第三轴的【命令选项】(W1025)的位 0 为 On 代表轴 3 伺服需启动运动, 位 1 为 On 代表轴 4 伺服启动运动.

在圆弧运动中, 【命令选项】为圆弧运动的轴选项. 若 ASDA-M 的三轴依序为轴 1, 轴 2 与轴 3, 0 代表在 1, 2 轴平面执行圆弧运动, 1 代表在 2, 3 轴平面执行圆弧运动, 2 代表在 1, 3 轴平面执行圆弧运动.

在螺旋运动中, 【命令选项】为运动中执行螺旋移动的轴选项. 若 ASDA-M 的三轴依序为轴 1, 轴 2 与轴 3, 0 代表在 1, 2 轴平面执行圆弧运动, 轴 3 为螺旋的高度移动运动; 1 代表在 2, 3 轴平面执行圆弧运动, 轴 1 为螺旋的高度移动运动; 2 代表在 1, 3 轴平面执行圆弧运动, 轴 2 为螺旋的高度移动运动.

  - 关联装置

当需要动作的轴包含轴 1 时, 此运动需触发轴 1【命令执行】(R512), 此时使用参数为轴 1 的【命令代码】(W512), 【命令选项】(W513)与相关参数如速度, 位置等. 在第五章的运动运行范例中会有更详细介绍.

- 命令模式(W514)
  - 定义

直线运动使用的命令数据模式, 目前支持以下:

0: 绝对寻址命令, 位置命令终点, 直接指定为 DATA.

1: 相对定位命令, 位置命令终点由目前位置回授, 加上增加量 DATA.

2: 增量定位命令, 位置命令终点由上一次命令终点, 加上增加量 DATA.

- DELAY时间(W515)
  - 定义

单次运动中位置到达后的 Delay 时间设定.

- 加速时间(W516)
  - 定义

    单次运动中的加速时间设定，若无设定则参考系统的【加速时间】(W646).

- 减速时间(W517)
  - 定义

    单次运动中的减速时间设定，若无设定则参考系统的【减速时间】(W647).

- 速度设定(W518)
  - 定义

    在多轴同动运动中，由于多轴间运动必须同时启动且同时结束，因此需依据各轴的运动速度设定与移动行程距离间进行实际运动速度的调整. 目前同动运动中驱动器默认是以多轴中最长运动距离的轴速度设定为分配的基准，也就是最长运动行程的轴运动速度会依照设定速度运行，而其它较短行程的轴将会以此基准速度进行速度分配，以达成多轴同动运动.
  - 异常状况

    以轴 1 为例，若速度设定为 0，则会导致命令错误发生，【命令错误】(R1024)变为 On. 而若速度设定超过【最大速度限制】(W660)设定值，仅会以【最大速度限制】之速度运动. 在多轴同动运动中，当某一轴运行速度设定大于【最大速度限制】，则实际上该轴仅会以【最大速度限制】来运动，因此整体来说多轴同动运动会被每一轴的【最大速度限制】限制住而可能导致速度下降.

- 目标位置(W520)
  - 定义

    命令到达位置设定.
  - 关联装置

    实际运动目标位置会受到【命令模式】(W514)影响，会因绝对，相对或增量等模式而造成最终运动命令的目标位置点不同.

- 速度比例(W522)
  - 定义

    轴运动时的速度比率设定. 轴实际的运动速度为【速度设定】乘上【速度比例】百分比后的速度动作，若【速度设定】为 10000，速度比例设为 20，则实际下达运动速度为 10000 x 20% = 2000，且各轴以触发轴之速度比例影响所有同动运动速度.

此设定值合理范围为1~100, 当超过此合理范围时则速度比例设定等同为100, 也就是直接以【速度设定】进行运动.

- 参数起始地址(W524)
  - 定义

在圆弧, 螺旋等运动命令下达时需要将相关参数数据写入一区连续的 D 缓存器中, 而【参数起始地址】就是设定此路径数据的起始位置. 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则运动命令下达时将会从 D1000 地址开始抓取路径数据并下达给伺服.

在圆弧: 半径&角度模式运动(命令代码为 10)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达圆弧运动时会从 D1000 开始将参数数据下达给伺服执行圆弧运动. 参数使用 6 个连续字符, 因此须将 6 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1005 使用. 假设【参数起始地址】设定为 n, 使用圆弧运动时此区数据内容定义如下:

定义	参数地址定义
半径 (DW)	Dn
起始角度 (DW, 单位 0.5 度)	Dn+2
运动角度 (DW, 单位 0.5 度)	Dn+4

在圆弧: 中点&终点模式运动(命令代码为 11)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达运动命令时会从 D1000 开始将参数数据下达给伺服. 参数使用 8 个连续字符, 因此须将 8 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1007 使用. 假设【参数起始地址】设定为 n, 此区数据内容定义如下:

定义	参数地址定义
中点坐标 1 (DW)	Dn
中点坐标 2 (DW)	Dn+2
终点坐标 1 (DW)	Dn+4
终点坐标 2 (DW)	Dn+6

在圆弧: 圆心&终点模式运动(命令代码为 12)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达运动命令时会从 D1000 开始将参数数据下达给伺服. 参数使用 10 个连续字符, 因此须将 10 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1009 使用. 假设【参数起始地址】设定为 n, 此区数据内容定义如下:

定义	参数地址定义
圆心坐标 1 (DW)	Dn
圆心坐标 2 (DW)	Dn+2
终点坐标 1 (DW)	Dn+4
终点坐标 2 (DW)	Dn+6

正反转 (DW, 0 顺时, 1 逆时)	Dn+8

在圆弧:终点&半径模式运动(命令代码为 13)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达运动命令时会从 D1000 开始将参数数据下达给伺服. 参数使用 8 个连续字符, 因此须将 8 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1007 使用. 假设【参数起始地址】设定为 n, 此区数据内容定义如下:

定义	参数地址定义
终点坐标 1 (DW)	Dn
终点坐标 2 (DW)	Dn+2
半径 (DW)	Dn+4
正反转 (DW, 0 顺时, 1 逆时)	Dn+6

在圆弧:圆心&角度模式运动(命令代码为 14)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达运动命令时会从 D1000 开始将参数数据下达给伺服. 参数使用 6 个连续字符, 因此须将 6 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1005 使用. 假设【参数起始地址】设定为 n, 此区数据内容定义如下:

定义	参数地址定义
圆心坐标 1 (DW)	Dn
圆心坐标 2 (DW)	Dn+2
运动角度 (DW, 单位 0.5 度)	Dn+4

在螺旋运动(命令代码为 30)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达螺旋运动时会从 D1000 开始将参数数据下达给伺服执行螺旋运动. 参数使用 8 个连续字符, 因此须将 8 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1007 使用. 假设【参数起始地址】设定为 n, 使用螺旋运动时此区数据内容定义如下:

定义	参数地址定义
半径 (DW)	Dn
起始角度 (DW, 单位 0.5 度)	Dn+2
运动角度 (DW, 单位 0.5 度)	Dn+4
高度 (DW)	Dn+6

在螺旋 W 运动(命令代码为 31)中, 以轴 1 为例, 若【参数起始地址】(W524)设为 1000, 则当下达螺旋 W 运动时会从 D1000 开始将参数数据下达给伺服执行螺旋 W 运动. 参数使用 10 个连续字符, 因此须将 10 个连续字符长度保留, 如 W524 设为 1000, 则程序规划时须保留 D1000 ~ D1009 使用. 假设【参数起始地址】设定为 n, 使用螺旋 W 运动时此区数据内容定义如下:

定义	参数地址定义
圆心坐标 1 (DW)	Dn
圆心坐标 2 (DW)	Dn+2

一圈高度 (DW)	Dn+4
总圈数 (DW)	Dn+6
偏移角度 (DW, 单位 0.5 度)	Dn+8

■ 关联装置

以轴 1 执行圆弧运动为例, 在【命令代码】(W512)设定圆弧命令, 【命令选项】(W513)设定轴选项后, 再将【命令执行】(R512)触发命令下达与运动开始, 则 HMC 会从【参数起始地址】(W524) 设定值开始的 D 缓存器地址开始读取圆弧运动数据并下达参数给伺服.

➢ OVERLAP(W525)

■ 定义

当执行连续路径运动时, 【Overlap】为设定当前运动与下一个运动间的路径迭合设定值, 透过此设定能实现两路径间的补间运动.

提供 2 种 Overlap 方式:

1. 加减速时间重迭:

当前路径的减速时间与下一路径的加速时间的时间重迭百分比, 此时驱动器参数【重迭路径设定】(P1-78)需设定为 0, 【Overlap】设定值范围与定义如下:

級距	7	6	5	4	3	2	1	0
百分比	45%	40%	35%	30%	25%	20%	10%	0%
級距	F	E	D	C	B	A	9	8
百分比	100%	90%	80%	75%	70%	65%	55%	50%

2. 路径距离重迭:

当前路径减速区段距离占整个路径之百分比, 此时驱动器参数【重迭路径设定】(P1-78)需设定为 1,【Overlap】设定值范围与定义如下:

0 ~ F 索引	對應說明
0	1%
1	2%
2	4%
3	6%
4	8%
5	10%
6	12%
7	14%
8	16%
9	18%
A	20%
B	參考【P1-79 重疊路徑距離百分比設定】
C	參考【P1-80 重疊路徑距離設定】
D	Reserved
E	Reserved
F	Reserved

其中驱动器【P1-79】设定范围为 1 ~ 30, 【P1-80】设定单位为 PUU, 范围为 1000~2147483647.

■ 关联装置

当执行连续路径运动时, 以轴 1 为例, 当使用【命令预载】(R624)加载与触发命令时, 会将此【Overlap】写至伺服以设定两路径间的迭合补间动作.

➢ 速度选项(W526)

■ 定义

透过【速度选项】设定可改变直线运动时的使用速度型态, 提供以下设定:

1. 0: 最长行程速度.
2. 1 ~ 12: 指定轴速度, 设定值代表多轴直线运动中需维持指定轴的速度设定. 例如在轴 1, 轴 2, 轴 3 的三轴直线运动中, 此设定值若为 2, 代表运动时将维持轴 2 的速度设定(W774)并调整轴 1 与轴 3 的运动速度以达成三轴直线同动.
3. 255: 向量速度, 将触发轴的【速度设定】作为多轴直线运动的向量速度. 例如在轴 1, 轴 2, 轴 3 的三轴直线运动中, 此设定值若为 255, 代表运动时将以触发轴的速度设定(W518)作为运动的向量速度.

■ 关联装置

当执行多轴直线同动运动且包含轴 1 时, 需设定轴 1 的【命令代码】(W512)为 1(直线), 【命令选项】(W513)为相关运动轴选项, 【速度设定】(W518, W774, W1030, ...)与【目标位置】

(W520, W776, W1032, ...)等，并参考【速度选项】(W526)决定使用的速度模式，接着触发轴 1 的【命令执行】(R512)即可开始运动

状态缓存器

目前伺服异常与运动功能状态. 以第一轴为例说明:

功能	编号	说明	属性	停电保持	出厂值
错误代码	W576	命令下达失败的错误代码	R	No	0
当前位置(DW)	W578	当前伺服轴位置，单位为 PUU	R	No	0
平均转矩(DW)	W580	当前伺服轴平均转矩，单位为%	R	No	0
当前速度(DW)	W582	当前伺服轴速度，单位为 PUU/s	R	No	0
伺服异常代码	W585	驱动器发生异常代码，为 BCD 格式	R	No	0
监控项目 1(DW)	W586	监控项目索引 1 指定的伺服信息	R	No	0
监控项目 2(DW)	W588	监控项目索引 2 指定的伺服信息	R	No	0
监控项目 3(DW)	W590	监控项目索引 3 指定的伺服信息	R	No	0
监控项目 4(DW)	W592	监控项目索引 4 指定的伺服信息	R	No	0
路径剩余数	W594	当前写入伺服中的运动路径数	R	No	0
DMCNet 通讯错误数 A(DW)	W596	DMCNet 通讯之 A 信道遗失封包数	R	No	0
DMCNet 通讯错误数 B(DW)	W598	DMCNet 通讯之 B 信道遗失封包数	R	No	0
高速监控项目(DW)	W600	高速监控项目索引指定的伺服信息	R	No	0
当前转速	W602	当前伺服轴转速，单位为 RPM	R	No	0

- 错误代码(W576)
  - 定义

命令下达错误时，伴随【命令错误】发生，显示当前命令错误代码. 代码如下:

异常码	定义
01	运动速度参数为零: 1. 速度设定为零 2. 经电子齿轮比转换后转速为零
02	服务器紧急停止: 下达命令之相关轴为紧急停止状态

03	服务器 Servo-Off: 下达命令之相关轴为 Servo Off 状态
04	命令执行中: 下达新的非连续运动命令时相关各轴中尚有动作执行，不能接收此新命令
05	连续路径运动指令错误: 1. 在连续运动中的每个运动都必须是相同的伺服轴所组成，若途中的新运动命令改变了运动轴选项，将导致此错误 2. 运动命令中的相关轴不存在(未联机)
06	螺旋或圆弧运动指令错误: 1. 下达的运动位置参数错误，例如三点成弧的三点坐标为一直线，将导致此错误 2. 下达的运动平面选择参数错误 3. 运动命令中的相关轴不存在(未联机)
07	命令代码错误: 使用未定义的命令代码
08	下达连续命令时超过最大连续指令数目: 当执行连续路径时是可连续下达多笔运动命令，但系统仍有最多记忆 8 笔的个数限制. 当已达到记忆 8 笔运动命令状态，此时再下达新的连续命令触发将无法成功，发生此错误
09	下达连续命令超时: 执行连续路径时，当连续路径运动已执行至最后一笔运动，亦即连续运动将结束，此时再下达新的连续命令触发将无法成功，发生此错误
10	运动命令无法使用于连续路径动作中: 连续路径目前只支持直线、圆弧与螺旋，下达其它运动命令将发生此错误
11	指定轴速错误: 当执行指定轴速度的多轴直线运动时，但指定速度的轴移动距离为 0
12	多轴同动暂停时间过长: 多轴同动运动中，不移动轴的停止时间暂停时间是依该路径运动时间计算得出，当超过伺服可接受的暂停时间，则发生此错误. 可透过加快此路径运动

	速度解决.
13	多轴同动暂停时间为零: 多轴同动运动中，不移动轴的停止时间暂停时间是依该路径运动时间计算得出，当暂停时间计算为不正常的 0，则发生此错误.
14	运动模式切换错误: 触发【命令执行】或【命令预载】下达命令时该轴处于其它运动模式，前一寸动或手轮操作动作未完成导致.
15	更新伺服参数错误: 当处于不可填伺服参数模式时(寸动或手轮启动模式下)，执行变更伺服参数 W 动作时将导致此错误. 通常发生此错误原因可能为程序中存在持续写伺服参数指令导致.
16	伺服参数同步错误: 当系统已发生伺服参数同步失败错误时，且【伺服参数同步异常动作模式】(W14)设定为 1 或 2 模式，此时安全保护机制启动，在未异常确认前将无法下达任何运动命令.

- 当前位置(W578)
  - 定义

    当前伺服轴的位置，单位为 PUU.

- 平均转矩(W580)
  - 定义

    当前伺服轴的平均转矩，单位为%.

- 当前速度(W582)
  - 定义

    当前伺服轴的速度，单位为 PUU/s.

- 伺服异常代码(W585)
  - 定义

    驱动器发生异常时的代码，此代码为 BCD 格式.

    代码定义请参考伺服使用手册.

- 监控项目 1/2/3/4 (W586/W588/W590/W592)
  - 定义

    对应驱动器的状态监控缓存器.

  - 关联装置

    【监控项目索引】决定【监控项目】显示内容. 以轴 1 为例, 【监控项目 1】(W586)会根据【监控项目索引 1】(W666)设定而变更显示内容.

    当变更【监控项目索引】后 HMC 会将此改变值下达给驱动器, 因此需等 1ms 的通讯时间后对应的【监控项目】才能正确显示变更后参数内容.

- 路径剩余数(W594)
  - 定义

    已写入驱动器并等待执行的运动命令数. 目前 HMC 最多可先预先加载 8 个运动命令至驱动器中, 因此范围为 0~8, 而 0 代表目前无运动路径执行中.

    在连续运动中, 当【路径剩余数】为 8 代表已经预载 8 个运动命令至驱动器中了, 若此时再触发【命令预载】(R624)将无法失败, 也就是【命令就绪】(R1040)并不会变为 On, 需等到当前运动完成后, 使得【路径剩余数】低于 8 个, 新的命令才能再写入至驱动器中, 也就是此时在下达【命令预载】(R624), 【命令就绪】(R1040)才会变为 On.

- DMCNET通讯错误数A/B (W596/W598)
  - 定义

    显示在 DMCNet 信道 A/B 上发生的封包遗失累积数.

- 高速监控项目 (W600)
  - 定义

    高速更新对应驱动器的状态监控缓存器.

  - 关联装置

    【高速监控项目索引】(W688)决定【高速监控项目】(W600)显示之内容.

    当变更【高速监控项目索引】后 HMC 会将此改变值下达给驱动器, 因此需等 1ms 的通讯时间后对应的【高速监控项目】才能正确显示变更后参数内容.

- 当前转速(W602)
  - 定义

    当前伺服轴的速度, 单位为 RPM.

运动参数缓存器

下达伺服运动控制相关参数. 以下以第一轴为例说明:

功能	编号	说明	属性	停电保持	出厂值
电子齿轮比分子(DW)	W640	电子齿轮比分子	Remote	Yes	1
电子齿轮比分母(DW)	W642	电子齿轮比分母	Remote	Yes	1
单位显示	W644	单位设定, 0 为 PUU	R/W	Yes	0
加减速曲线	W645	S 形加减速平滑常数	Remote	Yes	0
加速时间	W646	加速时间	R/W	Yes	200
减速时间	W647	减速时间	R/W	Yes	200
原点复归速度 1(DW)	W648	第一段原点复归速度	R/W	Yes	2133333
原点复归速度 2(DW)	W650	第二段原点复归速度	R/W	Yes	426666
原点复归模式	W652	原点复归模式选择	R/W	Yes	1
原点复归加减速时间	W653	原点复归动作的加减速时间	R/W	Yes	200
原点复归偏移值(DW)	W654	原点与定位点的偏移值	R/W	Yes	0
正向软件极限(DW)	W656	正向软件极限位置	Remote	Yes	0x7FFFFFFF
反向软件极限(DW)	W658	反向软件极限位置	Remote	Yes	0x80000000
最大速度限制(DW)	W660	最大可运转速度，单位为 PUU/s	Remote	Yes	64000000
监控项目索引 1	W666	设定监控项目 1 内容	R/W	Yes	1
监控项目索引 2	W667	设定监控项目 2 内容	R/W	Yes	13
监控项目索引 3	W668	设定监控项目 3 内容	R/W	Yes	39
监控项目索引 4	W669	设定监控项目 4 内容	R/W	Yes	40
Quick Stop 减速时间	W670	Quick Stop 时减速时间	R/W	Yes	200
停止命令减速时间	W671	减速停止时减速时间	Remote	Yes	30
通讯错误减速时间	W672	通讯错误时减速时间	Remote	Yes	30
马达过载减速时间	W673	马达过载时减速时间	Remote	Yes	30
反向软件极限减速时间	W674	遇反向软件极限的减速停止时间	Remote	Yes	30
正向软件极限减速时间	W675	遇正向软件极限的减速停止时间	Remote	Yes	30
反向硬件极限减速时间	W676	遇反向硬件极限的减速停止时间	Remote	Yes	30
正向硬件极限减速时间	W677	遇正向硬件极限的减速停止时间	Remote	Yes	30
寸动速度(DW)	W678	寸动运动速度	Remote	Yes	426666
寸动加速时间	W680	寸动运动加速度曲线	R/W	Yes	200

寸动减速时间	W681	寸动运动减速度曲线	R/W	Yes	200
寸动扭力限制	W682	寸动运动时扭力限制设定，单位为 0.1%	R/W	No	0
FeedRate 速度(DW)	W684	FeedRate 速度设定	R/W	No	0
FeedRate 加速时间	W686	FeedRate 加速时间设定	R/W	Yes	200
FeedRate 减速时间	W687	FeedRate 减速时间设定	R/W	Yes	200
高速监控项目索引	W688	设定高速监控项目内容	R	No	0
最大转速限制	W689	最大可运转转速，单位为 RPM	R	No	0
手轮反向软件极限(DW)	W690	手轮动作时反向软件极限位置	R/W	Yes	0x80000000
手轮正向软件极限(DW)	W692	手轮动作时正向软件极限位置	R/W	Yes	0x7FFFFFFF

- 电子齿轮比分子(W640)
  - 定义

    伺服轴之电子齿轮比分子设定，必须在伺服 Servo Off 状态下才能设定.

    透过【电子齿轮比分子】与【电子齿轮比分母】的设定将脉波命令(Pulse)转换成位置指令(PUU):

    指令脉波输入: f1 ***位置指令: f2***

    电子齿轮比分子: N ***电子齿轮比分母: M***

    f2 = f1 x (N/M)
  - 关联装置

    根据【电子齿轮比分母】转换出伺服轴的用户单位 PUU. 电子齿轮比范围 1/50 ~ 25600.
- 电子齿轮比分母(W642)
  - 定义

    伺服轴之电子齿轮比分母设定，必须在伺服 Servo Off 状态下才能设定.

    透过【电子齿轮比分子】与【电子齿轮比分母】的设定将脉波命令(Pulse)转换成位置指令(PUU):

    指令脉波输入: f1 ***位置指令: f2***

    电子齿轮比分子: N ***电子齿轮比分母: M***

    f2 = f1 x (N/M)
  - 关联装置

    根据【电子齿轮比分子】转换出伺服轴的用户单位 PUU. 电子齿轮比范围 1/50 ~ 25600.

- 单位显示(W644)
  - 定义

    HMC 与驱动器间使用之单位设定，目前仅支持默认值 0，即 PUU.

- 加减速曲线(W645)
  - 定义

    运动时的系统 S 型加减速平滑常数设定，对应伺服轴的【P1-36 S 形平滑曲线中的加减速平滑常数】设定.

- 加速时间(W646)
  - 定义

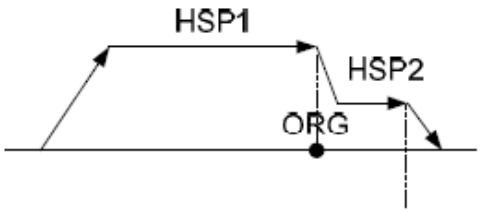
    运动时的系统加速时间设定.

- 减速时间(W647)
  - 定义

    运动时的系统减速时间设定.

- 原点复归速度 1(W648)
  - 定义

    第一段高速原点复归速度设定，当执行复归动作时，开始的复归运动速度，对应伺服参数【P5-05 HSP1】. 此伺服参数可接受设定值范围 0.1 ~ 2000.0 (RPM).

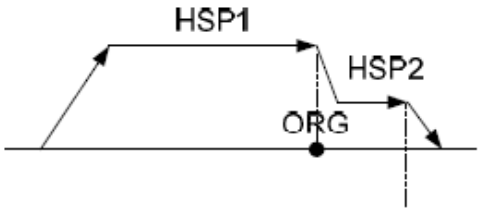


  - 关联装置

    执行原点复归运动时，会先以【原点复归速度 1】执行复归动作，当遇到原点参考点后切换至【原点复归速度 2】的速度寻找马达的 Z 脉波后才停止复归动作.

- 原点复归速度 2(W650)
  - 定义

    第二段低速原点复归速度设定，当执行复归动作时，遇到原点参考点后的复归运动速度，对应伺服参数【P5-06 HSP2】. 此伺服参数可接受设定值范围 1 ~ 500.0 (RPM).



■ 关联装置

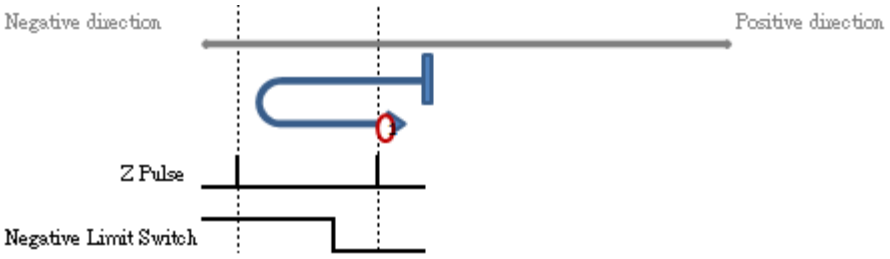
执行原点复归运动时，会先以【原点复归速度 1】执行复归动作，当遇到原点参考点后切换至【原点复归速度 2】的速度寻找马达的 Z 脉波后才停止复归动作.

➢ 原点复归模式(W652)

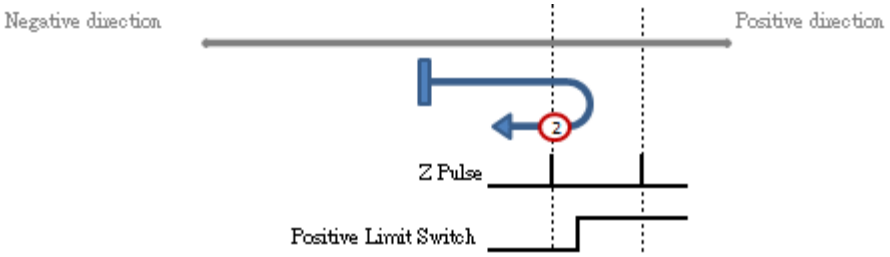
■ 定义

原点复归运动时使用的复归模式设定，支持的动作模式代码如下:

1: 反转找反向硬件极限，遇反向硬件极限后正转，找到第一个 Z 脉波为原点.



2: 正转找正向硬件极限，遇正向硬件极限后反转，找到第一个 Z 脉波为原点.



3: 根据 Home Switch 状态决定正转或反转.

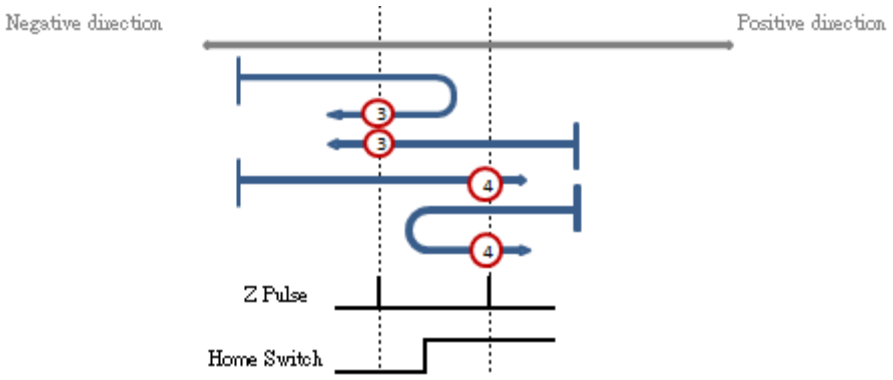
若执行复归动作，一开始在 Home Switch 上则反转，直到脱离 Home Switch 后，遇到第一个 Z 脉波为原点.

若执行复归动作，一开始不在 Home Switch 上则正转，寻找 Home Switch，接着反转直到脱离 Home Switch 后，遇到第一个 Z 脉波为原点.

4: 根据 Home Switch 状态决定正转或反转.

若执行复归动作，一开始在 Home Switch 上则反转，至脱离 Home Switch，接着正转遇到第一个 Z 脉波为原点.

若执行复归动作，一开始不在 Home Switch 上则正转，寻找 Home Switch 后，遇到第一个 Z 脉波为原点.



5: 根据 Home Switch 状态决定正转或反转.

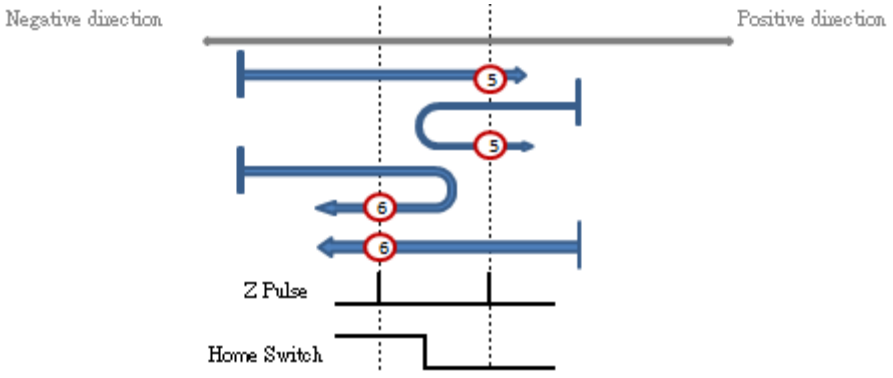
若执行复归动作一开始在 Home Switch 上则正转，直到脱离 Home Switch 后，遇到第一个 Z 脉波为原点.

若执行复归动作一开始不在 Home Switch 上则反转，寻找 Home Switch，接着反转直到脱离 Home Switch 后，遇到第一个 Z 脉波为原点.

6: 根据 Home Switch 状态决定正转或反转.

若复归动作一开始在 Home Switch 上则正转，至脱离 Home Switch，接着反转遇到第一个 Z 脉波为原点.

若复归动作一开始不在 Home Switch 上则反转，寻找 Home Switch 后，遇到第一个 Z 脉波为原点.



7: 根据 Home Switch 状态决定正转或反转.

若复归动作一开始不在 Home Switch 上，则正转至遇到 Home Switch 后，接着反转，至脱离 Home Switch 的第一个 Z 脉波为原点，若正转方向没有遇到 Home Switch 而遇到了正向硬件极限，先反转离开正向硬件极限，直到遇到 Home Switch 后又脱离 Home Switch 的第一个 Z 脉波为原点.

若复归动作一开始在 Home Switch 上则反转，至脱离 Home Switch 后的第一个 Z 脉波为原点.

简单的说就是寻找 Home Switch 下降缘讯号.

8: 根据 Home Switch 状态决定正转或反转.

若复归动作一开始不在 Home Switch 上则正转，至遇到 Home Switch 后的第一个 Z 脉波为原点，若正转方向没有遇到Home Switch，而遇到了正向硬件极限，先反转，离开正向硬件极限，接着继续反转，直到脱离 Home Switch，接着正转，当遇到 Home Switch 后的第一个 Z 脉波即为原点.

若复归动作一开始在Home Switch 上先反转，至脱离 Home Switch 后，接着正转，当遇到 Home Switch 的第一个 Z 脉波即为原点.

简单的说就是寻找 Home Switch 上升缘讯号.

9: 根据 Home Switch 状态决定正转或反转.

若复归动作一开始不在 Home Switch 上先正转，若遇到 Home Switch 则继续正转，至脱离 Home Switch 后，接着反转，至遇到 Home Switch 的第一个 Z 脉波即为原点. 若正转没有遇到 Home Switch，而是先遇到了正向硬件极限，则反转，直到遇到 Home Switch 后的第一个 Z 脉波为原点.

若复归动作一开始在 Home Switch 上先正转，至脱离 Home Switch 后，接着反转，遇到 Home Switch 的第一个 Z 脉波即为原点.

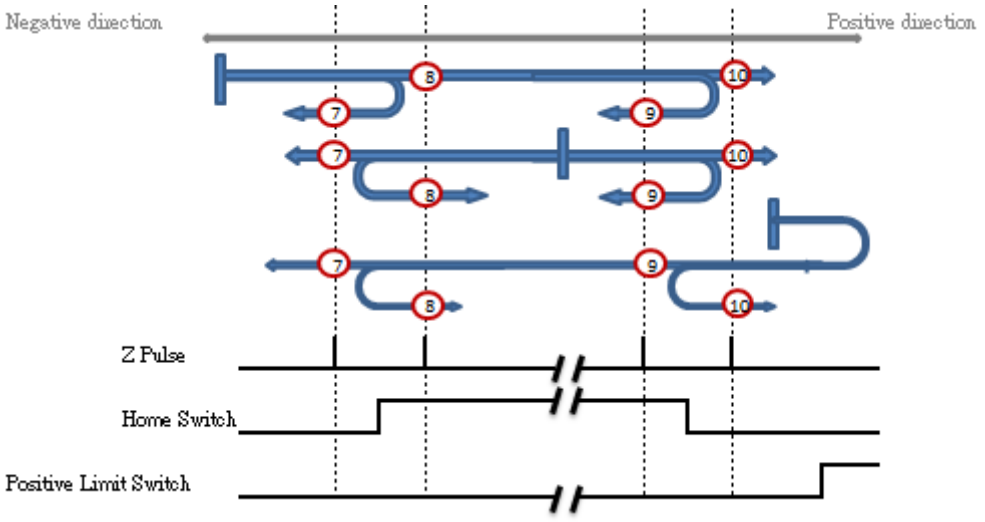
简单的说就是寻找 Home Switch 上升缘讯号.

10: 根据 Home Switch 状态决定正转或反转.

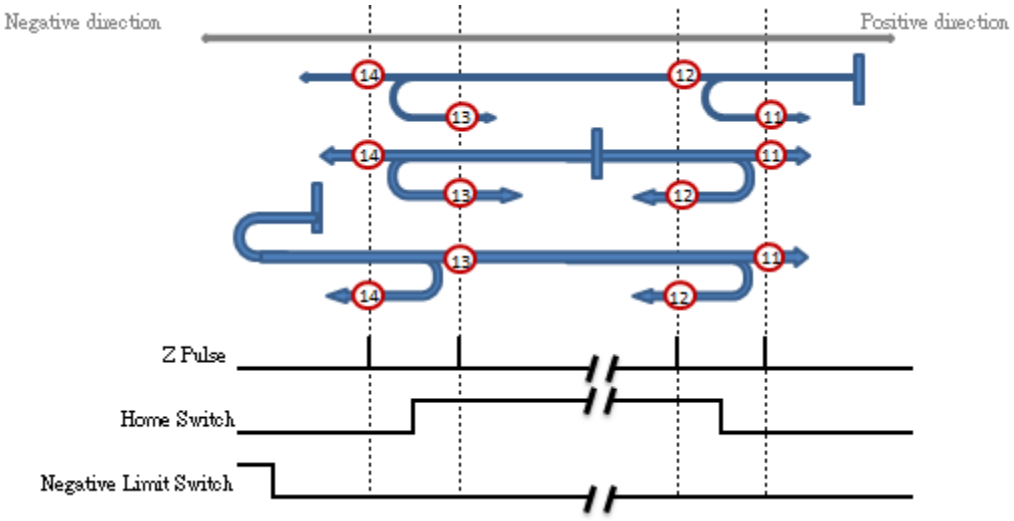
若复归动作一开始不在 Home Switch 上先正转，若先遇到 Home Switch 则继续正转，至脱离 Home Switch 后的第一个 Z 脉波即为原点. 若正转方向没有遇到 Home Switch，而先遇到了正向硬件极限，先反转，脱离正向硬件极限，继续反转，直到遇到 Home Switch，接着正转，脱离 Home Switch 后的第一个 Z 脉波即为原点.

若复归动作一开始在 Home Switch 上则正转，至脱离 Home Switch 后的第一个 Z 脉波为原点.

简单的说就是寻找 Home Switch 下降缘讯号.



11~14: 复归方法与 7~10 对应，差别在于一开始的正反转相反..



17~30: 复归方法与 1~14 相似，差别在于不需再寻 Z 脉波.

33: 反转寻找第一个 Z 脉波位置为原点.

34: 正转寻找第一个 Z 脉波位置为原点.

35: 当前位置为原点.

■ 关联装置

执行原点复归动作时，【命令代码】须设定为 8 代表执行原点复归动作且将【命令选项】写入相关运动轴后再触发【命令执行】，此时各轴会根据【原点复归模式】,【原点复归速度 1】,【原点复归速度 2】,【原点复归加减速时间】与【原点复归偏移植】等参数执行原点复归动作.

➢ 原点复归加减速时间(W653)

■ 定义

伺服原点复归运动时的加减速时间设定，单位为 ms.

- 原点复归偏移值(W654)
  - 定义

    执行原点复归动作完成后将找到的位置点设定为与原点相距的偏移值. 如将此设定值设为 1000, 当执行原点复归后找到的位置会定位为-1000 的坐标位置.

- 正向软件极限(W656)
  - 定义

    设定伺服轴的正向软件极限位置.
  - 关联装置

    寸动或原点复归运动时, 遇到软件极限并不会使动作停止, 但该伺服轴会发生【伺服 Warning】状态变为 On. 执行其它运动时若遇到软件极限, 会使动作停止且该轴会发生【伺服 Warning】为 On 与【伺服 Quick Stop 解除】变为 Off 状态.

    须启动【Fault Reset】为 On 使伺服重置异常状态, 才能正确执行接下来运动命令.

- 反向软件极限(W658)
  - 定义

    设定伺服轴的反向软件极限位置.
  - 关联装置

    寸动或原点复归运动时, 遇到软件极限并不会使动作停止, 但该伺服轴会发生【伺服 Warning】状态变为 On. 执行其它运动时若遇到软件极限, 会使动作停止且该轴会发生【伺服 Warning】为 On 与【伺服 Quick Stop 解除】变为 Off 状态.

    须启动【Fault Reset】为 On 使伺服重置异常状态, 才能正确执行接下来运动命令.

- 最大速度限制(W660)
  - 定义

    在寸动运动时速度设定超过此设定值, 则该伺服轴无法寸动. 其它运动命令中, 若运动速度命令大于该设定值, 则会以【最大速度限制】之速度运动.

    若变更电子齿轮比后, 导致超过驱动器最大限制则会自动变更【最大速度限制】设定值. 假设 HMC 搭配 ASDA 驱动器为 1280000 Pulse 转一圈, 最高转速 5000rpm. 透过【电子齿轮比分子】与【电子齿轮比分母】将脉波命令(Pulse)转换成位置指令(PUU), 以下为最大速度限制的计算方式,:

    指令脉波输入: f1 ***位置指令: f2***

    电子齿轮比分子: N ***电子齿轮比分母: M***

f2 = f1 x (N/M)

=> f1 = f2 x (M/N)

=> 1 Pulse = (M/N) PUU

=> 1280000 Pulse = 1280000 x (M/N) PUU

=> 1 r = 1280000 x (M/N) PUU

=> 1 rps = 1280000 x (M/N) PUU/s

=> 1 rpm = 1280000 / 60 x (M/N) puu/s

伺服最大转速 5000rmp

=> 5000 rmp = 5000 x 1280000 /60 x (M/N) puu/s

- 监控项目索引 1/2/3/4 (W666/W667/W668/W669)
  - 定义

设定【监控项目 1/2/3/4】的显示内容. 设定内容与【驱动器状态显示】参数功能相同, 参数定义如下:

00: 马达回授脉波数

01: 脉波命令输入脉波数

02: 控制命令脉波与回授脉波误差数

03: 马达回授脉波数

04: 脉波命令输入脉波数

05: 误差脉波数

06: 脉波命令输入频率

07: 马达转速

08: 速度输入命令

09: 速度输入命令

10: 扭矩输入命令

11: 扭矩输入命令

12: 平均转矩

13: 峰值转矩

14: 主回路电压

15: 负载/马达惯性比

16: IGBT 温度

17: 共振频率

18: 相对于编码器 Z 相的绝对脉波数

39: DI 状态

40: DO 状态

- 关联装置

【监控项目索引】决定【监控项目】的显示内容. 以轴 1 为例, 【监控项目 1】(W586)会根据【监控项目索引 1】(W666)设定而变更显示参数内容.

➢ QUICK STOP减速时间(W670)

- 定义

伺服 Quick Stop 动作时的减速时间设定.

- 关联装置

以轴 1 为例, 在运动中将【实时停止】(R528)启动, 则会以【Quick Stop 减速时间】(W670)执行停止运动, 停止动作完成后【伺服 Quick Stop 解除】(R1088)为 Off 表示伺服处于即停状态中.

➢ 停止命令减速时间(W671)

- 定义

伺服执行减速停止命令动作时的减速时间设定.

- 关联装置

以轴 1 为例, 要执行减速停止运动时, 须将【命令代码】(W512)设为 6, 接着触发【命令执行】(R512)才能执行停止运动, 当伺服速度为 0 停止后【命令完成】(R1056)状态变为 On, 表示动作完成.

➢ 通讯错误减速时间(W672)

- 定义

DMCNet 通讯发生错误时, 伺服会以【通讯错误减速时间】减速停止并且伺服 Servo Off.

- 马达过载减速时间(W673)
  - 定义

    当伺服连续输出负载高于设定比例时，伺服会以【马达过载减速时间】减速停止.

- 反向软件极限减速时间(W674)
  - 定义

    伺服运动遇到反向软件极限时的减速停止时间设定.

- 正向软件极限减速时间(W675)
  - 定义

    伺服运动遇到正向软件极限时的减速停止时间设定.

- 反向硬件极限减速时间(W676)
  - 定义

    伺服运动遇到反向硬件极限时的减速停止时间设定.

- 正向硬件极限减速时间(W677)
  - 定义

    伺服运动遇到正向硬件极限时的减速停止时间设定.

- 寸动速度(W678)
  - 定义

    伺服寸动运动时的速度设定.
  - 关联装置

    若【寸动速度】设定值大于【最大速度限制】，则无法执行寸动.

- 寸动加速时间(W680)
  - 定义

    伺服寸动运动时的加速度时间曲线设定.

- 寸动减速时间(W681)
  - 定义

    伺服寸动运动时的减速度时间曲线设定

- 寸动扭力限制(W682)
  - 定义

    伺服寸动运动时的最大输出扭力限制设定，设定单位为 0.1%. 当此值设为 500 时,代表寸动运动时最高的扭力输出不会超过 50%.

- FEEDRATE速度(W684)
  - 定义

    改变当前运动之速度设定值.
  - 关联装置

    当触发轴的【Feed Rate 执行】为 On 后, 会将目前运动速度改变为【Feed Rate 速度】设定值, 改变完成后【Feed Rate 执行】自动清除.

- FEEDRATE加速时间(W686)
  - 定义

    改变当前运动之加速时间设定值.
  - 关联装置

    当触发轴的【Feed Rate 执行】为 On 后, 会将目前运动之加速时间改变为【Feed Rate 加速度时间】, 改变完成后【Feed Rate 执行】自动清除.

- FEEDRATE减速时间(W687)
  - 定义

    改变当前运动之减速时间设定值.
  - 关联装置

    当触发轴的【Feed Rate 执行】为 On 后, 会将目前运动之减速时间改变为【Feed Rate 减速度时间】, 改变完成后【Feed Rate 执行】自动清除.

- 快速监控项目索引 (W688)
  - 定义

    设定【快速监控项目】的显示内容. 设定内容与【驱动器状态显示】参数功能相同, 参数定义如下:

    00: 马达回授脉波数

    01: 脉波命令输入脉波数

    02: 控制命令脉波与回授脉波误差数

03: 马达回授脉波数

04: 脉波命令输入脉波数

05: 误差脉波数

06: 脉波命令输入频率

07: 马达转速

08: 速度输入命令

09: 速度输入命令

10: 扭矩输入命令

11: 扭矩输入命令

12: 平均转矩

13: 峰值转矩

14: 主回路电压

15: 负载/马达惯性比

16: IGBT 温度

17: 共振频率

18: 相对于编码器 Z 相的绝对脉波数

39: DI 状态

40: DO 状态

- 关联装置

【快速监控项目索引】(W688)决定【快速监控项目】(W600)的显示内容.

➢ 最大转速限制(W689)

- 定义

最大可运动转速限制，此设定值会将【最大速度限制】(W660)同步改变. 在寸动运动时速度设定超过此设定值，则该伺服轴无法寸动. 其它运动命令中，若运动速度命令大于该设定值，则会以【最大速度限制】之速度运动.

➢ 手轮反向软件极限(W690)

- 定义

设定启用手轮动作时伺服轴的反向软件极限位置.

■ 关联装置

开启【手轮启用】(R608)后，手轮控制模式下的反向位置极限.

➢ 手轮正向软件极限(W692)

■ 定义

设定启用手轮动作时伺服轴的正向软件极限位置.

■ 关联装置

开启【手轮启用】(R608)后，手轮控制模式下的正向位置极限.

伺服参数缓存器

下达伺服控制相关参数. 以下以第一轴为例说明:

功能	编号	属性	停电保持	出厂值
自动低频抑振模式设定	W704	Remote	No	0
对伺服的负载惯量比与负载重量比	W705	Remote	No	10
位置控制比例增益	W706	Remote	No	35
位置控制前馈增益	W707	Remote	No	50
速度控制增益	W708	Remote	No	500
速度积分补偿	W709	Remote	No	100
共振抑制低通滤波	W710	Remote	No	20
外部干扰抵抗增益	W711	Remote	No	0
速度检测滤波及微振抑制	W712	Remote	No	0
位置控制误差过大条件(DW)	W724	Remote	Yes	3840000
电子凸轮的曲线表格倍率设定	W726	Remote	No	1000000
E－CAM：Master 齿轮比设定 P	W728	Remote	No	3600
E－CAM：凸轮启动控制	W730	Remote	No	0
E－CAM：脱离时机数据	W732	Remote	No	0

➢ 自动低频抑振模式设定(W704)

■ 定义

对应驱动器的【P1-29 AVSM 自动低频抑振模式设定】，设定为 0 代表固定频率，设定为 1 代表抑振后自动固定.

➢ 对伺服的负载惯量比与负载重量比(W705)

■ 定义

对应驱动器的【P1-37 GDR 对伺服的负载惯量比与负载重量比】，

旋转式马达: (J_load / J_motor)

其中 J_motor: 伺服马达本体的转动惯量,

J_load: 外部机械负载的总体等效转动惯量

- 位置控制比例增益(W706)
  - 定义

对应驱动器的【P2-00 KPP 位置控制比例增益】，位置控制增益值加大时，可提升位置应答性及缩小位置控制误差量. 但若设定太大时易产生振动及噪音.

- 位置控制前馈增益(W707)
  - 定义

对应驱动器的【P2-02 PFG 位置控制前馈增益】，位置控制命令平滑变动时，增益值加大可改善位置跟随误差量. 若位置控制命令不平滑变动时，降低增益值可降低机构的运转振动现象.

- 速度控制增益(W708)
  - 定义

对应驱动器的【P2-04 KVP 速度控制增益】，速度控制增益值加大时，可提升速度应答性. 但若设定太大时易产生振动及噪音.

- 速度积分补偿(W709)
  - 定义

对应驱动器的【P2-06 KVI 速度积分补偿】，速度控制积分值加大时，可提升速度应答性及缩小速度控制误差量. 但若设定太大时易产生振动及噪音.

- 共振抑制低通滤波(W710)
  - 定义

对应驱动器的【P2-25 NLP 共振抑制低通滤波】，设定共振抑制低通率波时间常数. 设为 0 时关闭低通滤波功能.

- 外部干扰抵抗增益(W711)
  - 定义

对应驱动器的【P2-26 DST 外部干扰抵抗增益】，调大此参数会增加速度回路的阻尼.

- 速度检测滤波及微振抑制(W712)
  - 定义

对应驱动器的【P2-49 SJIT 速度检测滤波及微振抑制】，设定速度估测滤波.

- 位置控制误差过大条件(W724)
  - 定义

对应驱动器的【P2-35 PDEV 位置控制误差过大警告条件】，驱动器位置控制误差过大警告条件之设定.

- 电子凸轮的曲线表格倍率设定(W726)
  - 定义

对应驱动器的【P5-19 TBS 电子凸轮的曲线表格倍率设定】，在不改变电子凸轮曲线表格的内容下，改变本参数，相当于对表格数据作放大 / 缩小.

- E－CAM：MASTER 齿轮比设定P (W728)
  - 定义

对应驱动器的【P5-84 ECMP E－CAM：Master 齿轮比设定 P】，收到 Master 脉波数 P，凸轮转轴旋转 M 周，即凸轮表格 M 周.

- E－CAM：凸轮启动控制(W730)
  - 定义

对应驱动器的【P5-88 ECON E－CAM：凸轮启动控制】，凸轮启动/命令来源/嚙合之控制.

- E－CAM：脱离时机数据(W732)
  - 定义

对应驱动器的【P5-89 ECRD E－CAM：脱离时机数据】，凸轮脱离之控制.

4. 指令介绍

4.1、 基本指令

➢ LD

指令	功能	使用地址数
LD	载入 A 接点	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

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缓存器内.

程式範例

阶梯图：

脚本：		说明：
D	X0	载入 X0 之 A 接点
AND	X1	串联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈

➢ LDI

指令	功能	使用地址数
LDI	载入 B 接点	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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缓存器内.

程式範例

阶梯图：

脚本：		说明：
LDI	X0	载入 X0 之 B 接点
AND	X1	串联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈

➢ AND

指令	功能	使用地址数
AND	串联 A 接点	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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，并将结果存入累积缓存器内.

程式範例

阶梯图：

脚本：

```
LDI   X1
AND   X0
OUT   Y1
```

说明：

载入 X1 之 B 接点
串联 X0 之 A 接点
驱动 Y1 线圈

➢ ANI

指令	功能	使用地址数
ANI	串联 B 接点	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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，并将结果存入累积缓存器内.

程式範例

阶梯图：

脚本：

```
LD    X1
ANI   X0
OUT   Y1
```

说明：

载入 X1 之 A 接点
串联 X0 之 B 接点
驱动 Y1 线圈

➢ OR

指令	功能	使用地址数
OR	并联 A 接点	1 Step

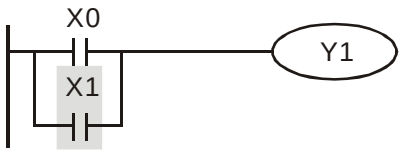
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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串联接点的状态再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累积缓存器内.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
OR	X1	并联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈

➢ ORI

指令	功能	使用地址数
ORI	并联 B 接点	1 Step

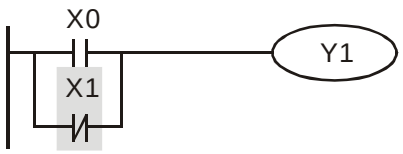
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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串联接点的状态再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累积缓存器内.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
ORI	X1	并联 X1 之 B 接点
OUT	Y1	

➢ ANB

指令	功能	使用地址数
ANB	串联回路方块	1 Step

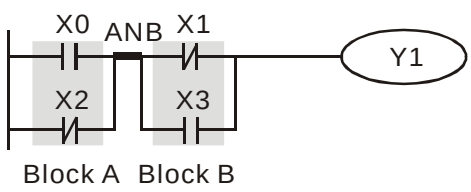
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

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

程式範例

阶梯图：



脚本：		说明：
LD	X	载入 X0 之 A 接点
ORI	X2	并联 X2 之 B 接点
LDI	X1	载入 X1 之 B 接点
OR	X3	并联 X3 之 A 接点
ANB		串联回路方块
OUT	Y1	驱动 Y1 线圈

➢ ORB

指令	功能	使用地址数
ORB	并联回路方块	1 Step

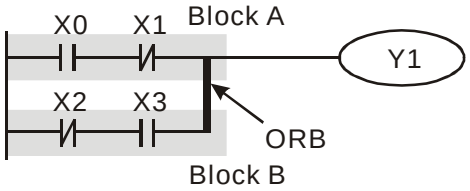
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

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

程式範例

阶梯图：



脚本：

		说明：
LD	X0	载入 X0 之 A 接点
ANI	1	串联 X1 之 B 接点
LDI	X2	载入 X2 之 B 接点
AND	X3	串联 X3 之 A 接点
ORB		并联回路方块
OUT	Y1	驱动 Y1 线圈

➢ MPS

指令	功能	使用地址数
MPS	存入堆栈	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

将目前累积缓存器的内容存入堆栈. （堆栈指针加一）

➢ MDR

指令	功能	使用地址数
MDR	读出堆栈（指针不动）	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

将目前累积缓存器的内容存入堆栈. （堆栈指针加一）

➢ MPP

指令	功能	使用地址数
MPP	读出堆栈	1 Step

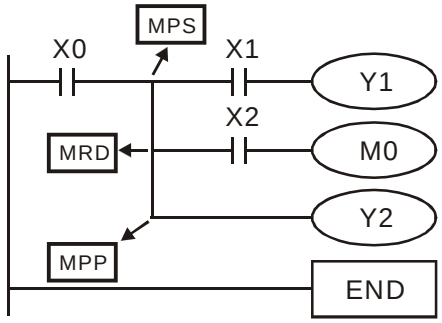
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

自堆栈取回前一保存的逻辑运算结果, 存入累积缓存器. （堆栈指针减一）

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
MPS		存入堆栈
AND	X1	串联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈
MRD		读出堆栈（指针不动）
AND	X2	串联 X2 之 A 接点
OUT	M0	驱动 M0 线圈
MPP		读出堆栈
OUT	Y2	驱动 Y2 线圈
END		程序结束

➢ OUT

指令	功能	使用地址数
OUT	驱动线圈	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

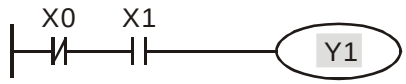
将 OUT 指令之前的逻辑运算结果输出至指定的组件.

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

程式範例

阶梯图：



脚本：		说明：
LDI	X0	载入 X0 之 B 接点
AND	X1	串联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈

➢ SET

指令	功能	使用地址数
SET	动作保持（ON）	1 Step

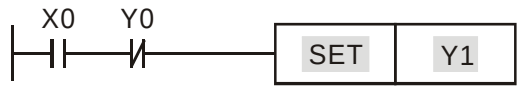
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
ANI	Y0	串联 Y0 之 B 接点
SET	Y1	Y1 动作保持（ON）

➢ RST

指令	功能	使用地址数
RST	接点或缓存器清除	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

当 RST 指令被驱动，其指定的组件的动作如下：

元件	状态
S, Y, 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, E, F	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.

程式範例

阶梯图：

脚本：		说明：
LD	X0	载入 X0 之 A 接点
RST	Y5	Y5 接点清除

➢ PLS

指令	功能	使用地址数
PLS	上微分输出	1 Step

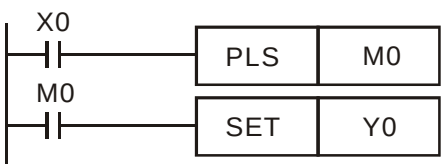
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

上微分输出指令. 当条件接点由 Off 到 On（正缘触发）时, PLS 指令被执行, S 送出一次脉波，脉波长度为一次扫描时间.

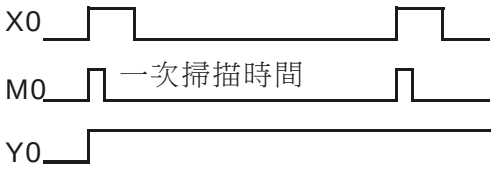
程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
PLS	M0	M0 上微分输出
LD	M0	载入 M0 之 A 接点
SET	Y0	Y0 动作保持(ON)

时序图：



➢ PLF

指令	功能	使用地址数
PLF	下微分输出	1 Step

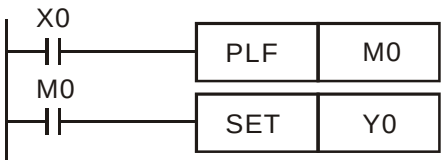
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

下微分输出指令. 当条件接点由 On 到 Off（负缘触发）时, PLF 指令被执行, S 送出一次脉波，脉波长度为一次扫描时间.

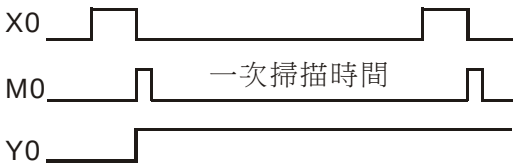
程式範例

阶梯图：



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

时序图：



➢ MC/MCR

指令	功能	使用地址数
MC/MCR	共通串联接点之连结 / 解除	1 Step

操作数	N0 ~N7

指令說明

MC 为主控起始指令，当 MC 指令执行时，位于 MC 与 MCR 指令之间的指令照常执行. 当 MC 指令 Off 时，位于 MC 与 MCR 指令之间的指令动作如下所示：

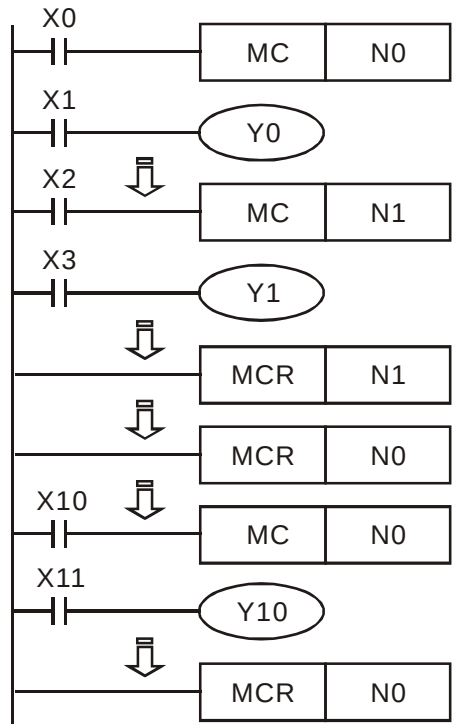
指令区分	说明
一般定时器	计时值归零，线圈失电，接点不动作
积算型定时器	线圈失电，计时值及接点保持目前状态
计数器	线圈失电，计数值及接点保持目前状态
OUT 指令驱动的线圈	全部不受电
SET, RST 指令驱动的组件	保持目前状态
应用指令	全部不动作，但 FOR-NEXT 巢串回路仍会来回执行 N 次，但 FOR-NEXT 间的任何指令依 MC-MCR 之间其它指令相同动作

MCR 为主控结束指令，置于主控程序最后，在 MCR 指令之前不可有接点指令.

MC-MCR 主控程序指令支持巢状程序结构，最多可 8 层， 使用时依 N0~N7 的顺序，请参考如下程序所示：

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
MC	N0	N0 共通串联接点之连结
LD	X1	载入 X1 之 A 接点
OUT	Y0	驱动 Y0 线圈
:		
LD	X2	载入 X2 之 A 接点
MC	N1	N1 共通串联接点之连结
LD	X3	载入 X3 之 A 接点
OUT	Y1	驱动 Y1 线圈
:		
MCR	N1	N1 共通串联接点之解除
:		
MCR	N0	N0 共通串联接点之解除
:		
LD	X10	载入 X10 之 A 接点
MC	N0	N0 共通串联接点之连结
LD	11	载入 X11 之 A 接点
OUT	Y10	驱动 Y10 线圈
:		
MCR	N0	N0 共通串联接点之解除

➢ LDP

指令	功能	使用地址数
LDP	正缘检出动作开始	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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缘检出状态存入累积缓存器内.

程式範例

阶梯图：

脚本：		说明：
L P	X0	X0 正缘检出动作开始
AND	X1	串联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈

➢ LDF

指令	功能	使用地址数
LDF	负缘检出动作开始	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符

	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

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缘检出状态存入累积缓存器内.

程式範例

阶梯图：

脚本：		说明：
LDF	X0	X0 负缘检出动作开始
AND	X1	串联 X1 之 A 接点
OUT	Y1	驱动 Y1 线圈

➢ ANDP

指令	功能	使用地址数
ANDP	正缘检出串联连接	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

ANDP 指令用于接点上升缘检出的串联连接.

程式範例

阶梯图：

脚本：		说明：
LD	X0	载入 X0 之 A 接点
ANDP	X1	X1 正缘检出串联连接
OUT	Y1	驱动 Y1 线圈

➢ ANDF

指令	功能	使用地址数
ANDF	负缘检出串联连接	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

ANDF 指令用于接点下降缘检出的串联连接.

程式範例

阶梯图：

脚本：		说明：
LD	X0	载入 X0 之 A 接点
ANDF	X1	X1 负缘检出串联连接
OUT	Y1	驱动 Y1 线圈

➢ ORP

指令	功能	使用地址数
ORP	正缘检出并联连接	1 Step

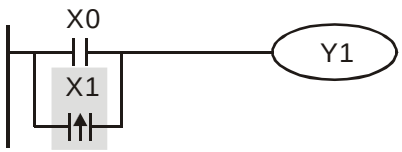
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

ORP 指令用于接点上升缘检出的并联连接.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
ORP	X1	X1 正缘检出并联连接
OUT	Y1	驱动 Y1 线圈

➢ ORF

指令	功能	使用地址数
ORF	负缘检出并联连接	1 Step

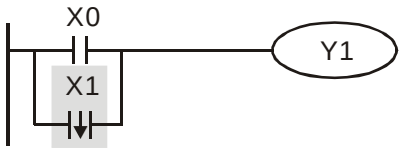
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	O	O	O	O	O	O	-	-	-	-	-	-	-	-	-	O	-

指令說明

ORP 指令用于接点下降缘检出的并联连接.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
ORF	X1	X1 负缘检出并联连接
OUT	Y1	驱动 Y1 线圈

➢ TMR

指令	功能	使用地址数
TMR	16 位定时器	2 Step

操作数	T-K	T0~T255, K0~K32,767
	T-D	T0~T255, D0~D65,535

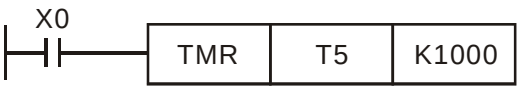
指令說明

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	开路
NC(Normally Close) 接点	闭合

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
TMR	T5 K1000	T5 定时器 设定值为 K1000

➢ CNT

指令	功能	使用地址数
CNT	16 位计数器	2 Step

操作数	C-K	C0~C199, K0~K32,767
	C-D	C0~C199, D0~D65,535

指令說明

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	开路
NC(Normally Close) 接点	闭合

当计数到达之后，若再有计数脉波输入，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.

程式範例

阶梯图：

脚本：		说明：
LD	X0	载入 X0 之 A 接点
CNT	C20 K100	C20 计数器设定值为 K100

➢ DCNT

指令	功能	使用地址数
DCNT	32 位计数器	3 Step

操作数	C-K	C200~C255, K-2,147,483,648~K2,147,483,647
	C-D	C200~C255, D0~D65,535

指令說明

DCNT 为 32 位计数器 C200 至 C255 之启动指令.

一般用加减算计数器 C200~C255，当 DCNT 指令由 Off→On 时，计数器之现在值将执行上数（加一）的动作或下数（减一）的动作，依特 R32~R87 的设定模式.

当 DCNT 指令 Off 时，该计数器停止计数，但原有计数值不会被清除，可使用指令 RST C2XX 清除计数值及其接点.

程式範例

阶梯图：

脚本：		说明：
LD	M0	载入 M0 之 A 接点
DCNT	C254 K1000	C254 计数器 设定值为 K1000

➢ END

指令	功能	使用地址数
END	主程序结束	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

在主程序最后必须存在 END 指令. PLC 由地址 0 扫描到 END 指令，执行之后，返回到地址 0 重新作扫描执行.

编译后软件程序会自动加上.

➢ IRET

指令	功能	使用地址数
IRET	定时程序结束	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

在定时程序最后必须存在 IRET 指令. 定时程序中 PLC 由地址 0 扫描到 IRET 指令，执行之后该定时程序结束.

编译后软件程序会自动加上.

➢ SRET

指令	功能	使用地址数
SRET	子程序/运动程序结束	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

在子程序/运动程序最后必须存在 SRET 指令. 子程序/运动程序中 PLC 由地址 0 扫描到 SRET 指令，执行之后就结束该子程序/运动程序扫描.

编译后软件程序会自动加上.

➢ INV

指令	功能	使用地址数
INV	反相	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

将 INV 指令之前的逻辑运算结果反相存入累积缓存器内.

程式範例

阶梯图:

脚本:

		说明:
LD	X0	载入 X0 之 A 接点
INV		运算结果反相
OUT	Y1	驱动 Y1 线圈

➢ NP

指令	功能	使用地址数
NP	上升缘	1 Step

操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

将 NP 指令之前的逻辑运算结果取得上升缘检出状态并存入累积缓存器内.

程式範例

阶梯图:

脚本:

		说明:
LD	X0	载入 X0 之 A 接点
LD	M1	载入 M1 之 A 接点
NP		运算结果上升缘
OUT	Y1	驱动 Y1 线圈

➢ PN

指令	功能	使用地址数
PN	下降缘	1 Step

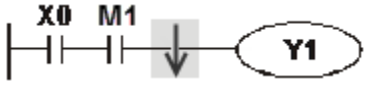
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

将 PN 指令之前的逻辑运算结果取得下降缘检出状态并存入累积缓存器内.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 A 接点
LD	M1	载入 M1 之 A 接点
PN		运算结果下降缘
OUT	Y1	驱动 Y1 线圈

➢ NOP

指令	功能	使用地址数
NOP	无动作	1 Step

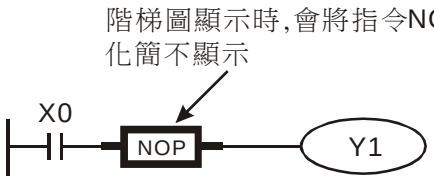
操作数	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
	-																

指令說明

指令 NOP 在程序不做任何运算，因此执行后仍会保持原逻辑运算结果，使用时机如下：想要删除某一指令，而又不想改变程序长度，则可以 NOP 指令取代.

程式範例

阶梯图：



脚本：		说明：
LD	X0	载入 X0 之 B 接点
NOP		无动作
OUT	Y1	驱动 Y1 线圈

4.2、 应用指令

➢ LD※

API					
001	D	LD※		S1 S2	接点型态比较 LD※

	位装置						字符装置									外接装置		
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符	
S1							O	O	O	O	O	O	O	O	O		O	16 位指令 (5 STEP) LD※
S2							O	O	O	O	O	O	O	O	O		O	
操作数使用注意：※：=、>、<、<>、≦、≧																		32 位指令 (5STEP) DLD※
																		旗标信号：无

指令說明

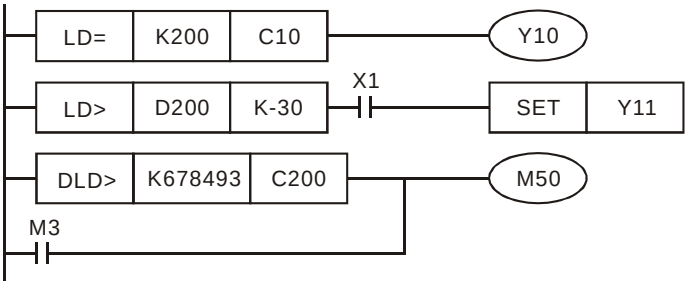
S1：数据源装置 1. S2：数据源装置 2.
S1 与 S2 之内容作比较的指令，比较结果成立时，该指令导通，不成立时，该指令不导通.
LD※的指令可直接与母线连接使用

16-bit 指令	32-bit 指令	导通条件	非导通条件
LD＝	DLD＝	S1 ＝ S2	S1 ≠ S2
LD＞	DLD＞	S1 ＞ S2	S1 ≦S2
LD＜	DLD＜	S1 ＜ S2	S1 ≧S2
LD＜＞	DLD＜＞	S1 ≠ S2	S1 ＝ S2
LD＜＝	DLD＜＝	S1 ≦S2	S1 ＞ S2
LD＞＝	DLD＞＝	S1 ≧S2	S1 ＜ S2

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※).

程式範例

C10 的内容等于 K200 时, Y10=On.
当 D200 的内容大于 K-30, 而且 X1=On 的时候, Y11=On 并保持住.
C200 的内容小于 K678,493 或者是 M3=On 的时候, M50=On.



➢ AND※

API					
002	D	AND※		(S1) (S2)	接点型态比较 AND※

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O

操作数使用注意：※：=、>、<、<>、≦、≧

16 位指令 (5 STEP)
AND
※

32 位指令 (5STEP)
DAND
※

旗标信号：无

指令說明

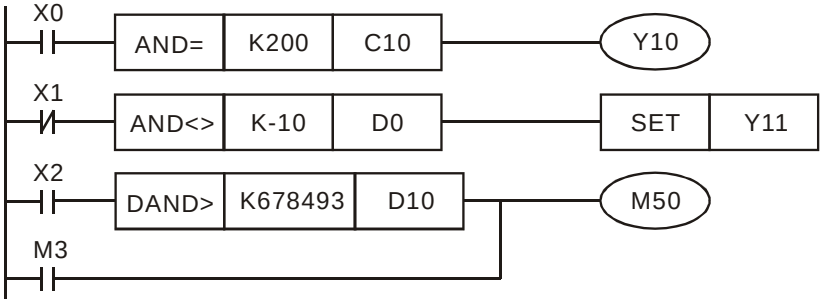
S1：数据源装置 1. S2：数据源装置 2.
S1 与 S2 之内容作比较的指令，比较结果成立时，该指令导通，不成立时，该指令不导通.
AND※的指令是与接点串接的比较指令.

16-bit 指令	32-bit 指令	导通条件	非导通条件
AND＝	DAND＝	S1 ＝ S2	S1 ≠S2
AND＞	DAND＞	S1 ＞ S2	S1 ≦S2
AND＜	DAND＜	S1 ＜ S2	S1 ≧S2
AND＜＞	DAND＜＞	S1 ≠ S2	S1 ＝ S2
AND＜＝	DAND＜＝	S1 ≦S2	S1 ＞ S2
AND＞＝	DAND＞＝	S1 ≧S2	S1 ＜ S2

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※).

程式範例

当 X0=On 时且 C10 的现在值又等于 K200 时, Y10=On.
当 X1=Off 而缓存器 D0 的内容又不等于 K-10 的时候, Y11=On 并保持住.
当 X2=On 而且 32 位缓存器 D0(D11)的内容又小于 678,493 的时候或 M3=On 时, M50=On.



➢ OR※

API		OR※		S1 S2	接点型态比较 OR※
003	D				

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O

操作数使用注意：※：=、>、<、<>、≦、≧

16 位指令 (5 STEP)
OR※

32 位指令 (5STEP)
DOR※

旗标信号：无

指令說明

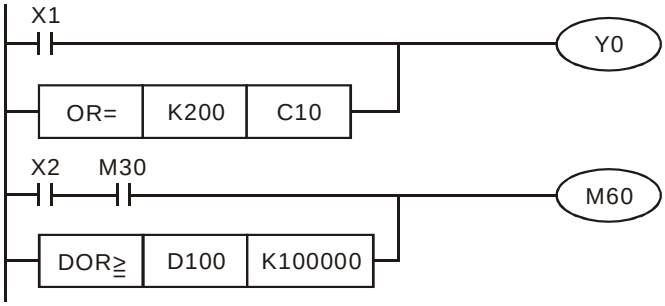
S1：数据源装置 1. S2：数据源装置 2.
S1 与 S2 之内容作比较的指令，比较结果成立时，该指令导通，不成立时，该指令不导通.
OR※的指令是与接点并接的比较指令.

16-bit 指令	32-bit 指令	导通条件	非导通条件
OR＝	DOR＝	S1 ＝ S2	S1 ≠ S2
OR＞	DOR＞	S1 ＞ S2	S1 ≦S2
OR＜	DOR＜	S1 ＜ S2	S1 ≧S2
OR＜＞	DOR＜＞	S1 ≠ S2	S1 ＝ S2
OR＜＝	DOR＜＝	S1 ≦S2	S1 ＞ S2
OR＞＝	DOR＞＝	S1 ≧S2	S1 ＜ S2

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※).

程式範例

当 X1=On 时，或者是 C10 的现在值等于 K200 时，Y0=On.
当 X2 及 M30 都等于 On 的时候，或者是 32 位缓存器 D100(D101)的内容大于或等于 K100,000 时，M60=On.



➢ MOV

API					
004	D	MOV		S D	数据移动

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置、
D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (5 STEP)
MOV

32 位指令 (5STEP)
DMOV
※

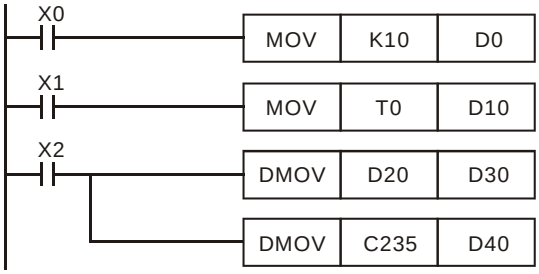
旗标信号：无

指令說明

S：资料之来源. 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.
若演算结果为 32 位输出时，（如应用指令 MUL 等）和 32 位装置高速计数器的现在值数据搬动则必须要用 DMOV 指令.

程式範例

16 位数据搬移，须使用 MOV 指令.
当 X0=Off 时, D10 内容没有变化，若 X0=On 时，将数值 K10 传送至 D10 数据缓存器内.
当 X1=Off 时, D10 内容没有变化，若 X1=On 时，将 T0 现在值传送至 D10 数据缓存器内.
32 位数据搬移，须使用 DMOV 指令.
当 X2=Off 时, (D31、D30)、(D41、D40)内容没有变化，若 X2=On 时，将(D21、D20)现在值传送至(D31、D30)数据缓存器内. 同时，将 C235 现在值传送至(D41、D40)数据缓存器内.



➢ BMOV

API		BMOV		S D n	全部传送
005					

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S										O	O	O			O		O
D										O	O	O					O
n												O			O		

操作数使用注意：S 操作数可使用外部 DAI、DAO 装置、
D 操作数可使用外部 DAO 装置、
n 操作数可使用 K 装置

16 位指令 (11 STEP)
BMOV

32 位指令

旗标信号：无

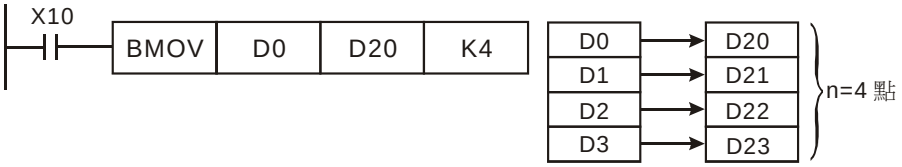
指令說明

S：来源装置起始. D：目的地装置起始. n：传送区块长度.

S 所指定的装置起始号码开始算 n 个缓存器的内容被传送至 D 所指定的装置起始号码开始算 n 个缓存器当中，如果 n 所指定点数超过该装置的使用范围时，则指令不执行.

程式範例

当 X10=On 时, D0~D3，共 4 个缓存器的内容被传送至 D20~D23 的 4 个缓存器内.



➢ CML

API					
006	D	CML		(S) (D)	反转传送

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置、
D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (5 STEP)
CML

32 位指令 (5STEP)
DCML

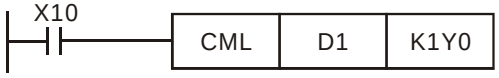
旗标信号：无

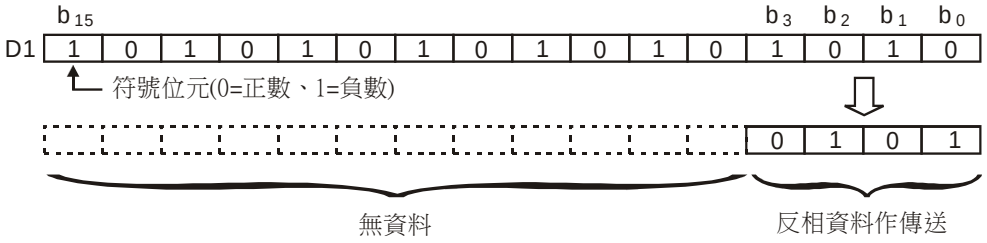
指令說明

S：传送之数据源. D：传送之目的地装置.
希望作反相输出时，使用本指令. 将 S 的内容全部反相（0→1、1→0）传送至 D 当中. 如果内容为 K 常数时，此 K 常数自动被转换成 BIN 值.

程式範例（一）

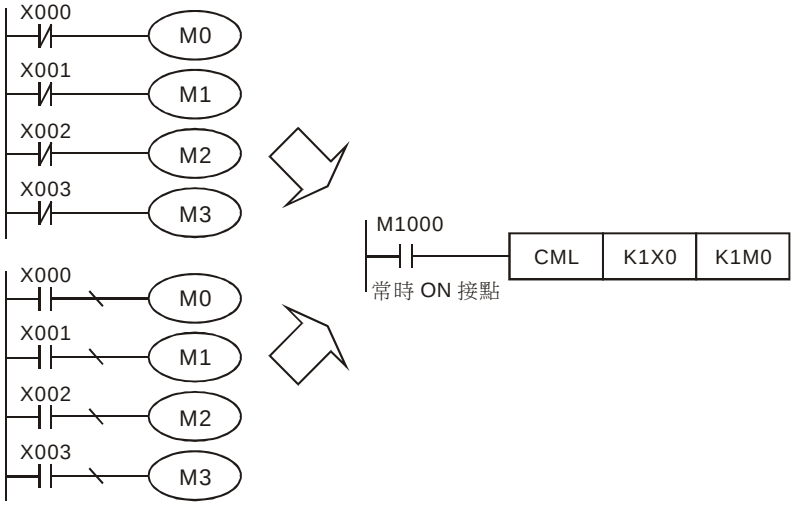
当 X10=On 时，将 D1 之 b0~b3 内容反相后传送到 Y0~Y3.





程式範例（二）

下图左边的回路也可以使用 CML 指令来表现，如下图右所示



➢ BCD

API					
007	D	BCD		(S) (D)	BIN→BCD 变换

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O			O
D												O	O	O			O

操作数使用注意：S 操作数可使用外部 DAI、DAO 装置、
D 操作数可使用外部 DAO 装置

16 位指令 (5 STEP)
BCD

32 位指令 (5STEP)
DBCD

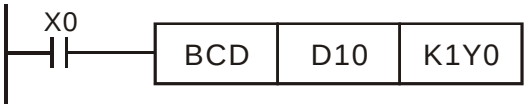
旗标信号：R20

指令說明

S：数据源. D：变换之结果.
数据源 S 的内容（BIN 值）作 BCD 的转换，存于 D.
在 BCD 变换结果若超过 0~9,999，指令错误旗标 R20=On，指令错误待码为 01.（BCD 值以 Hex 表示有任一位数不在 0~9 的范围内）
在 DBCD 转换结果若超过 0~99,999,999，指令错误旗标 R20=On，指令错误代码 W20 为 01.
PLC 内的四则运算、用及 INC、DEC 指令都是以 BIN 方式来执行. 所以在应用方面，当要看到 10 进制数值的显示器时，用 BCD 转换即可将 BIN 值变为 BCD 值输出.

程式範例

当 X0=On 时, D10 之 BIN 值被转换成 BCD 值后，将结果的个位数存于 K1Y0（Y0~Y3）四个 bit 组件.



若 D10=001E (Hex)=0030(十进制)，则执行结果 Y0~Y3=0000(BIN).

➢ BIN

API				
008	D	BIN	(S) (D)	BCD→BIN 变换

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O			O
D												O	O	O			O

操作数使用注意：S 操作数可使用外部 DAI、DAO 装置、
D 操作数可使用外部 DAO 装置

16 位指令 (5 STEP)
BIN

32 位指令 (5STEP)
DBIN

旗标信号：R20

指令說明

S：数据源. D：变换之结果.
数据源 S 的内容（BCD：0~9,999）作 BIN 的转换，存于 D.
数据源 S 的内容有效数值范围：BCD（0~9,999）, DBCD（0~99,999,999）.
当 S 的资料内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误，指令错误旗标 R20=On，指令错误代码 W20 为 04.
常数 K 会自动转换成 BIN，故不需运用此指令.

程式範例

当 X0=On 时, K1M0 之 BCD 值被转换成 BIN 值后，将结果存于 D10 中.

➢ FCMP

API					
009		FCMP		(S1) (S2) (D)	浮点数比较

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O	O		
S2												O	O	O	O		
D		O	O														

操作数使用注意：S1 操作数可使用 F 装置
S2 操作数可使用 F 装置

16 位指令

32 位指令 (7 STEP)
FCMP

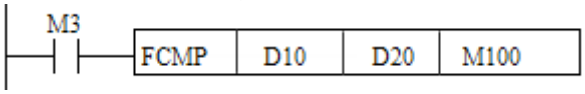
旗标信号：无

指令說明

S1：浮点数比较值 1. S2：浮点数比较值 2. D：比较结果.
浮点数比较值 1 与浮点数比较值 2 作比较, 比较的结果（＞、＝、＜）在 D 作表示. 当比较结果＞成立时, D 的第一个位 On；比较结果＝成立时, D 的第二个位 On；比较结果＜成立时, D 的第三个位 On.

程式範例

当 M3=On 时, D10 与 D20 的缓存器内容以浮点数格式做比较.
当 D10 大于 D20 时, M100=On.
当 D10 等于 D20 时, M101=On.
当 D10 小于 D20 时, M102=On.



➢ FMOV

API					
050		FMOV		(S) (D) (n)	多点移动

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S										O	O	O			O		O
D										O	O	O					O
N												O			O		O

操作数使用注意：S 操作数可使用 K 与外部 DAI、DAO 装置、
D 操作数可使用外部 DAO 装置、
n 操作数可使用 K 与外部 DAI、DAO 装置

16 位指令(11 STEP)
FMOV

32 位指令 (11 STEP)
DFMOV

旗标信号：无

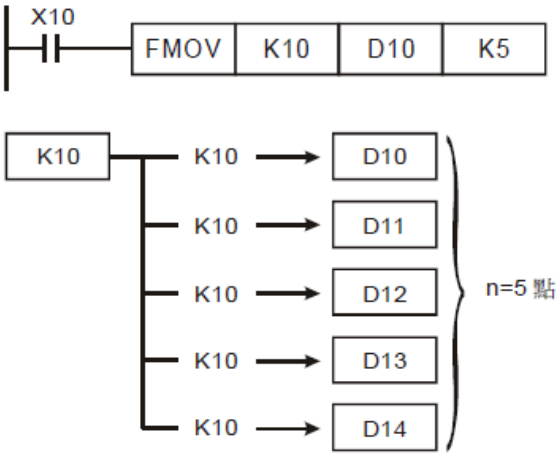
指令說明

S：数据源. 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 个缓存器当中, 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.

程式範例

当 X10=On 时, K10 被传送到由 D10 开始的连续 5 个缓存器中.



➢ REF

API					
010		REF		D n	I/O 更新

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D		O	O													O	
n																	

操作数使用注意：D 操作数可使用外部 DX、DY 装置、装置编号需为 16 的倍数
n 操作数范围 16~512，且为 16 之倍数

16 位指令 (2 STEP)
REF

32 位指令

旗标信号：无

指令說明

D：欲 I/O 更新处理之起始装置. n：I/O 更新处理之数目.
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. 因此在演算过程中需要最新的输入/出数据，则可利用本指令.

程式範例

当 X0=On 时，会立即读取 X0~X17 之输入点状态，输入信号更新，并没有输入延迟.

➢ ROR

API					
011	D	ROR		D n	右旋转

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D								O	O	O	O	O	O	O		O	
n																	

操作数使用注意：D 操作数可使用外部 KnDY、DAO 装置
n 操作数范围 n=K1~K16（16 位）, n=K1~K32（32 位）

16 位指令 (3 STEP)
ROR

32 位指令 (3 STEP)
DROR

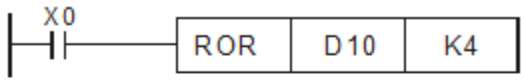
旗标信号：R10

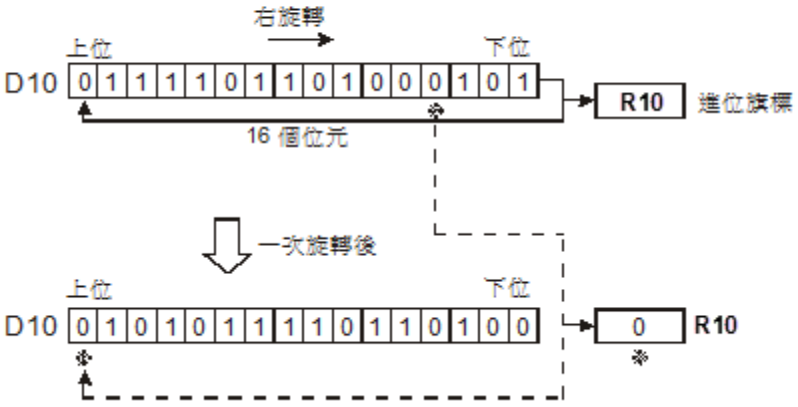
指令說明

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.

程式範例

当 X0 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旗标信号 R10 内.





➢ ROL

API					
012	D	ROL		(D) (n)	左旋转

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D								O	O	O	O	O	O	O		O	
n																	

操作数使用注意：D 操作数可使用外部 KnDY、DAO 装置
n 操作数范围 n=K1~K16（16 位）, n=K1~K32（32 位）

16 位指令 (3 STEP)
ROL

32 位指令 (3 STEP)
DROL

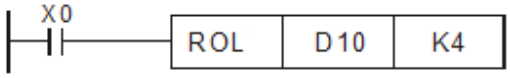
旗标信号：R10

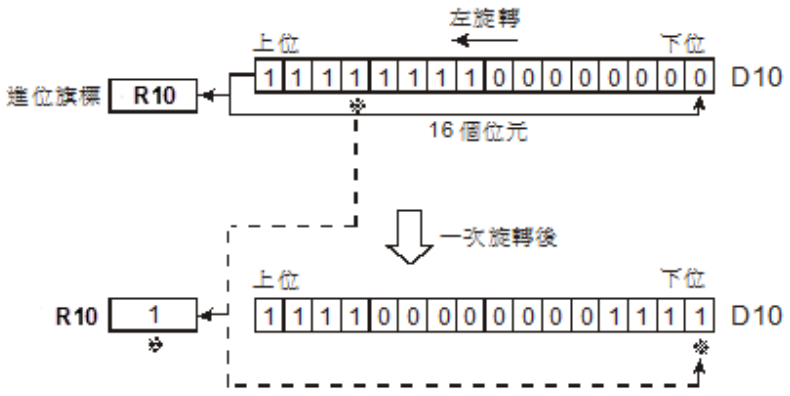
指令說明

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.

程式範例

当 X0 从 Off→On 变化时, D10 的 16 个位以 4 个位为一组往左旋转, 如下图所示标明※的位内容被传送至进位旗标信号 R10 内.





➢ CJ

API		CJ		(S)	条件跳跃
013					

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
操作数使用注意：S 操作数可指定 P0~P255																	

16 位指令 (2 STEP)
CJ

32 位指令

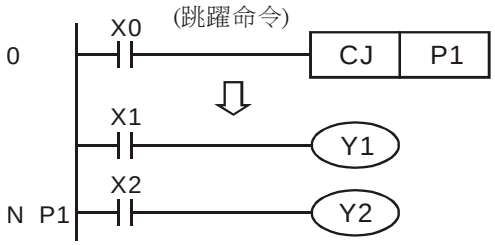
旗标信号：无

指令說明

S：条件跳跃之目的指标.
当用户希望 PLC 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 指令.
指针 P 所指之程序若在 CJ 指令之前, 需注意会发生程序扫描无法完成, 请注意使用.
CJ 指令可重复指定同一指标 P.
跳跃执行中各种装置动作情形说明：
Y、M、S 保持跳跃发生前之状态.
执行计时中之 10ms、100ms 定时器会暂停计时.
执行子程序用定时器 T192~T199 会继续计时, 且输出接点正常动作.
计数器会停止计数.
定时器之清除指令若在跳跃前被驱动, 则在跳跃执行中该装置仍处于清除状态.
应用指令不会被执行.

程式範例

当 X0=On 时, 程序自动从地址 0 跳跃至地址 N（即指定之标签 P1）继续执行, 中间地址跳过不执行.
当 X0=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.



➢ CALL

API					
014		CALL		S	呼叫子程序

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符

操作数使用注意：S 操作数为子程序名称

16 位指令 (2 STEP)
CALL

32 位指令

旗标信号：R18

指令說明

S：呼叫之子程序名称，此子程序需已存在.

使用 CALL 指令则可不限次数呼叫同一子程序.

子程序中再使用 CALL 指令呼叫其他子程序，包括本身最多可八层. 若进入第九层则该程序不执行，并会产生文法错误，文法错误旗标 R18=On，文法错误代码 W18 为 06.

➢ LAUNCH

API					
015		LAUNCH		S	启动运动程序

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符

操作数使用注意：S 操作数为运动程序名称

16 位指令 (2 STEP)
LAUNCH

32 位指令

旗标信号：R18

指令說明

S：呼叫之运动程序名称，此运动程序需已存在.
使用 LAUNCH 指令则可不限次数呼叫同一运动程序.
运动程序中不能使用 LAUNCH 指令.
在主程序或子程序中可不限次数使用 LAUNCH 指令呼叫运动程序，待执行的运动程序最多可暂存至 256 个. 若超过 256 个待执行运动程序，则之后再呼叫运动程序的指令不被执行，并会产生文法错误，文法错误旗标 R18=On，文法错误代码 W18 为 12.

➢ FOR

API					
016		FOR		S	巢串回路起始

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O			O

操作数使用注意：S 操作数可使用外部 DAO 装置

16 位指令 (3 STEP)
FOR

32 位指令

旗标信号：R18

指令說明

S：回路重复执行的次数.

➢ NEXT

API					
017		NEXT		-	巢串回路结束

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符

操作数使用注意：无操作数，不须接点驱动的指令

16 位指令 (1 STEP)
NEXT

32 位指令

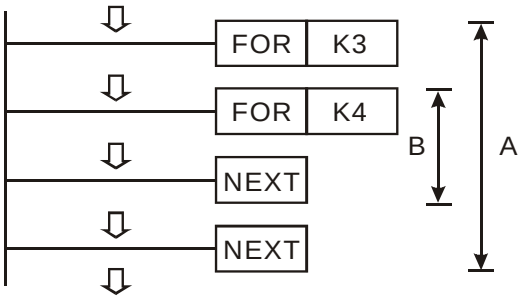
旗标信号：R18

指令說明

由 FOR 指令指定 FOR ~ NEXT 循环来回执行 N 次后跳出 FOR ~ NEXT 循环往下继续执行.
指定次数范围 N = K1 ~ K32,767，当指定次数范围 N ≦ K1 时，都视为是 K1.
当不执行 FOR ~ NEXT 回路时，可使用 CJ 指令来跳出回路.
下列情形会产生错误：
NEXT 指令在 FOR 指令之前.
有 FOR 指令没有 NEXT 指令.
END、SRET 或 IRET 指令之后有 NEXT 指令时.
FOR ~ NEXT 指令个数不同时.
巢串式 FOR ~ NEXT 回路最多可使用 5 层，回路次数超过时，将导致文法错误产生，文法错误旗标 R18=On，文法错误代码 W18 为 05.

程式範例

A 程序执行 3 次后在到 NEXT 指令以后的程序继续执行. 而 A 程序每执行一次 B 程序会执行四次，所以 B 程序合计共执行 3 × 4 ＝ 12 次.



➢ ADD

API					
018	D	ADD		S1 S2 D	BIN 加法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (7 STEP)
ADD

32 位指令 (7 STEP)
DADD

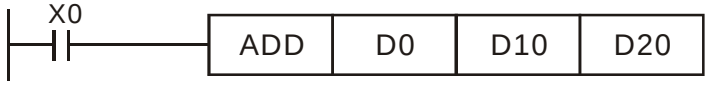
旗标信号：R8
R9
R10

指令說明

S1：被加数. S2：加数. D：和.
将两个资料源：S1 及 S2 以 BIN 方式相加的结果存于 D.
各数据的最高位位为符号位 0 表（正）1 表（负），因此可做代数加法运算. （例如：3+(-9)=-6）
加法相关旗标变化.
16 位 BIN 加法：
演算结果为 0 时，零旗标（Zero flag）R8 为 On.
演算结果小于 –32,768 时，借位旗标（Borrow flag）R9 为 On.
演算结果大于 32,767 时，进位旗标（Carry flag）R10 为 On.
32 位 BIN 加法：
1. 演算结果为 0 时，零旗标（Zero flag）R8 为 On.
2. 演算结果小于 –2,147,483,648 时，借位旗标（Borrow flag）R9 为 On.
3. 演算结果大于 2,147,483,647 时，进位旗标（Carry flag）R10 为 On.

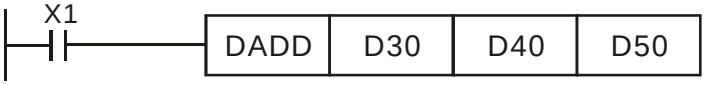
程式範例
(一)

16 位 BIN 加法：当 X0=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.



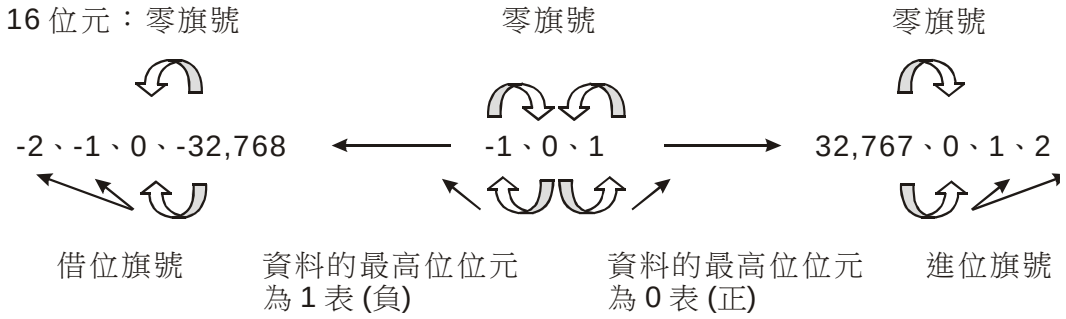
程式範例（二）

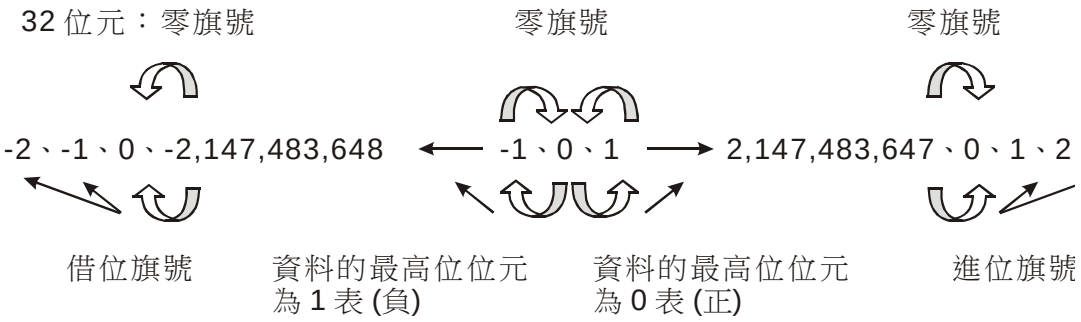
32 位 BIN 加法：当 X1=On 时，被加数(D31、D30)内容加上加数(D41、D40)之内容将结果存在(D51、D50)之中.（其中 D30、D40、D50 为低 16 位数据, D31、D41、D51 为高 16 位数据）



補充說明

旗标动作与数值的正负关系：





➢ SUB

API					
019	D	SUB		S1 S2 D	BIN 减法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (7 STEP)
SUB

32 位指令 (7 STEP)
DSUB

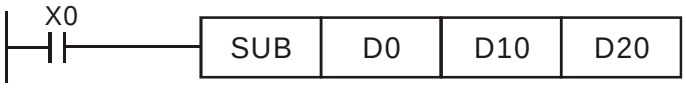
旗标信号：R8
R9
R10

指令說明

S1：被减数. S2：减数. D：减.
将两个资料源：S1 及 S2 以 BIN 方式相减的结果存于 D.
各数据的最高位位为符号位 0 表（正）1 表（负），因此可做代数减法运算. （例如：3+(-9)=-6）
减法相关旗标变化.
16 位 BIN 减法：
演算结果为 0 时，零旗标（Zero flag）R8 为 On.
演算结果小于 –32,768 时，借位旗标（Borrow flag）R9 为 On.
演算结果大于 32,767 时，进位旗标（Carry flag）R10 为 On.
32 位 BIN 减法：
1. 演算结果为 0 时，零旗标（Zero flag）R8 为 On.
2. 演算结果小于 –2,147,483,648 时，借位旗标（Borrow flag）R9 为 On.
3. 演算结果大于 2,147,483,647 时，进位旗标（Carry flag）R10 为 On.

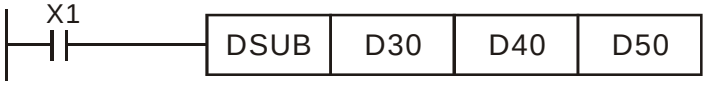
程式範例（一）

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



程式範例（二）

32 位 BIN 减法：当 X1=On 时, (D31、D30)内容减掉(D41、D40)之内容将差存在(D51、D50)之中. （其中 D30、D40、D50 为低 16 位数据, D31、D41、D51 为高 16 位数据）



➢ MUL

API					
020	D	MUL		S1 S2 D	BIN 乘法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
D 操作数可使用外部 KnDY、DAO 装置
16 位指令 D 操作数会占用连续 2 点
32 位指令 D 操作数会占用连续 4 点

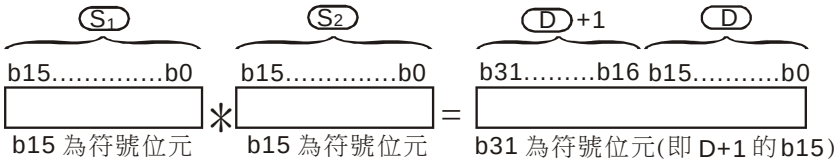
16 位指令 (7 STEP)
MUL

32 位指令 (7 STEP)
DMUL

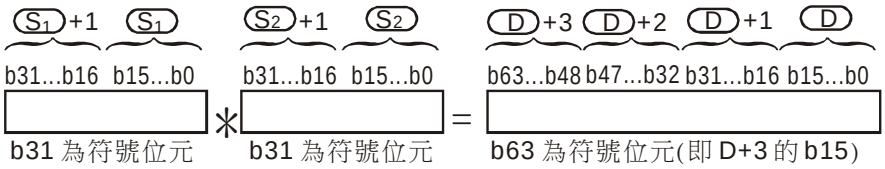
旗标信号：无

指令說明

S1：被乘数. S2：乘数. D：积.
将两个资料源：S1 及 S2 以有号数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，S1、 S2 及 D 的符号位.
16 位 BIN 乘法运算：



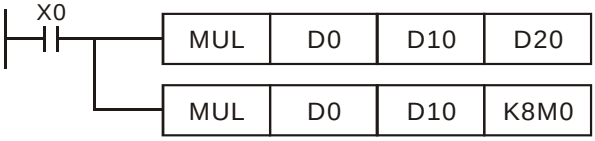
符号位=0 为正数，符号位=1 为负数.
32 位 BIN 乘法运算：：



符号位=0 为正数，符号位=1 为负数.

程式範例

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.



➢ DIV

API		DIV		S1 S2 D	BIN 除法
021	D				

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
D 操作数可使用外部 KnDY、DAO 装置
16 位指令 D 操作数会占用连续 2 点
32 位指令 D 操作数会占用连续 4 点

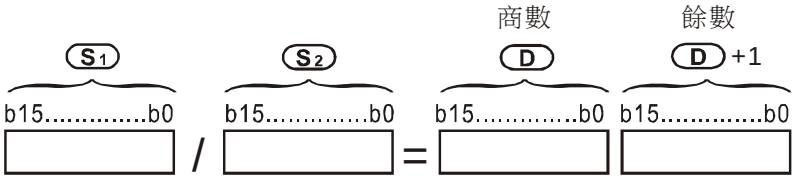
16 位指令 (7 STEP)
DIV

32 位指令 (7 STEP)
DDIV

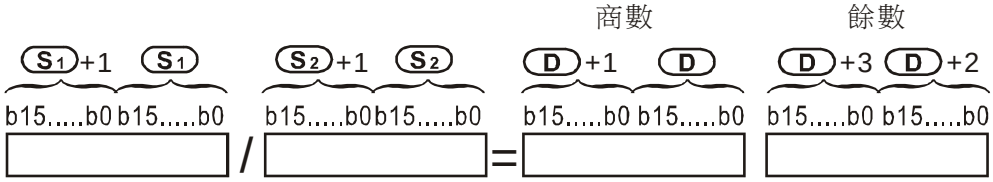
旗标信号：R20

指令說明

S1：被除数. S2：除数. D：商及余数.
将两个资料源：S1 及 S2 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, S1、 S2 及 D 的符号位.
除数为 0 时，指令不执行，指令错误 R20=On，指令记录错误代码 W20 为 02.
16 位 BIN 除法运算：

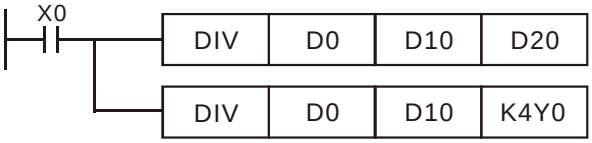


32 位 BIN 除法运算：



程式範例

当 X0=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.



➢ INC

API		INC		D	BIN 加一
022	D				

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D								O	O	O	O	O	O	O	O		O

操作数使用注意：D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (3 STEP)
INC

32 位指令 (3 STEP)
DINC

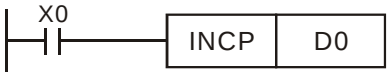
旗标信号：无

指令說明

D：目的地装置.
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.
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.

程式範例

当 X0=Off→On 时, D0 内容自动加 1.



➢ DEC

API		DEC		D	BIN 减一

023	D				

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D								O	O	O	O	O	O	O	O		O
操作数使用注意：D 操作数可使用外部 KnDY、DAO 装置																	

16 位指令 (3 STEP)
DEC

32 位指令 (3 STEP)
DDEC

旗标信号：无

指令說明

D：目的地装置.
则当指令执行时，程序每次扫描周期被指定的装置 D 内容都会减 1.
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.

程式範例

当 X0=Off→On 时, D0 内容自动减 1.

➢ WAND

API 024	D	WAND		S1 S2 D	AND 运算

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O
操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置 S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置 D 操作数可使用外部 KnDY、DAO 装置																	

16 位指令 (7 STEP)
WAND

32 位指令 (7 STEP)
DWAN
D

旗标信号：无

指令說明

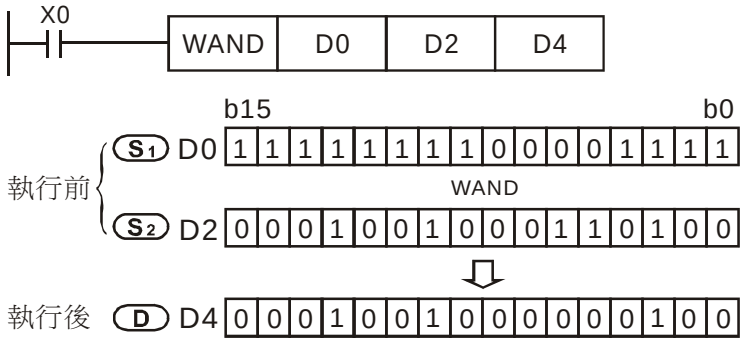
S1：数据源装置 1. S2：数据源装置 2. D：运算结果.

两个资料源：S1 及 S2 作 AND 运算并将结果存于 D.

AND 运算之规则为任一为 0 结果为 0.

程式範例

当 X0=On 时, 16 位 D0 与 D2 作 WAND, AND 运算后将结果存于 D4 中.



➢ WOR

API				
025	D	WOR	S1 S2 D	OR 运算

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O
D								O	O	O	O	O	O	O	O		O

操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (7 STEP)
WOR

32 位指令 (7 STEP)
DWOR

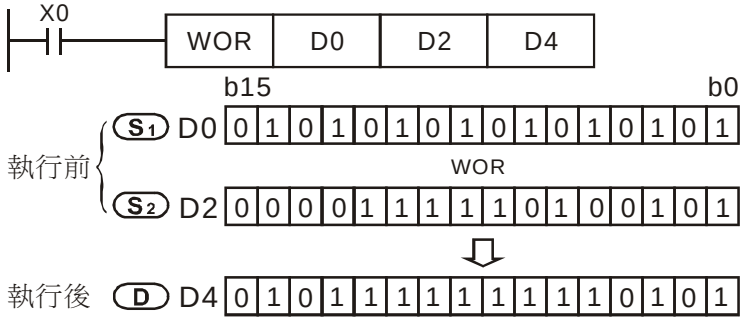
旗标信号：无

指令說明

S1：数据源装置 1. S2：数据源装置 2. D：运算结果.
两个资料源：S1 及 S2 作 OR 运算并将结果存于 D.
OR 运算之规则为任一为 1 结果为 1.

程式範例

当 X0=On 时, 16 位 D0 与 D2 作 WOR, OR 运算后将结果存于 D4 中.



➢ WXOR

API				
026	D	WXOR	S1 S2 D	XOR 运算

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1							O	O	O	O	O	O	O	O	O		O
S2							O	O	O	O	O	O	O	O	O		O

16 位指令 (7 STEP)
WXOR

D								0	0	0	0	0	0	0	0		0

操作数使用注意：S1 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
S2 操作数可使用外部 KnDX、KnDY、DAI、DAO 装置与 K 装置
D 操作数可使用外部 KnDY、DAO 装置

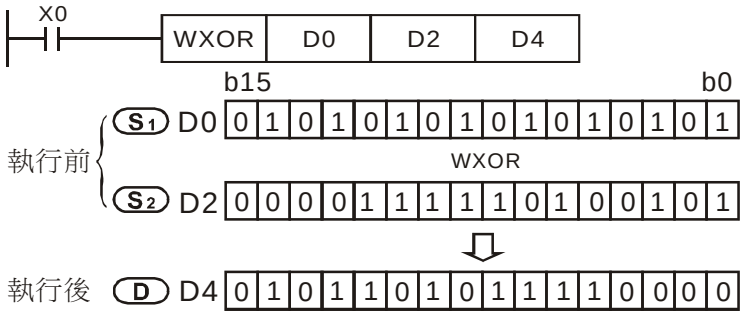
32 位指令 (7 STEP)
DWXOR
旗标信号：无

指令說明

S1：数据源装置 1. S2：数据源装置 2. D：运算结果.
两个资料源：S1 及 S2 作 XOR 运算并将结果存于 D.
OR 运算之规则为两者相同结果为 0, 两者不同结果为 1.

程式範例

当 X0=On 时, 16 位 D0 与 D2 作 WXOR, XOR 运算后将结果存于 D4 中.



➢ NEG

API				
027	D	NEG	D	2 的补码

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D								0	0	0	0	0	0	0	0		0

操作数使用注意：D 操作数可使用外部 KnDY、DAO 装置

16 位指令 (3 STEP)
NEG

32 位指令 (3 STEP)
DNEG

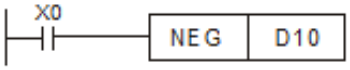
旗标信号：无

指令說明

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.

程式範例

当 X0=On 时, D10 内容的各位全部反相（0→1、1→0）后再加 1 存放于原缓存器 D10 当中.



補充說明

负数的表现及绝对值
正负数是以缓存器最上位（最左边）的位内容来表现, 为"0"时, 为正数、为"1"时, 为负数.
遇到负数时, 可使用 NEG 指令将它转成绝对值.

➢ FADD

API				
028		FADD	S1 S2 D	浮点数加法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O	O		O
S2												O	O	O	O		O
D												O	O	O	O		O

操作数使用注意：S1 操作数可使用 F 装置
S2 操作数可使用 F 装置

16 位指令

32 位指令 (7 STEP)
FADD

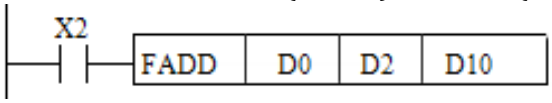
旗标信号：R8
R9
R10

指令說明

S1：被加数. S2：加数. D：和.
S1 所指定的缓存器内容加上 S2 所指定的缓存器内容，和被存放至 D 所指定的缓存器当中，加算的动作全部以浮点数型态进行.
若运算结果的绝对值大于可表示之最大浮点值，则进位旗标 R10=On.
若运算结果的绝对值小于可表示之最小浮点值，则借位旗标 R9=On.
若运算结果为 0，则零旗标 R8=On.

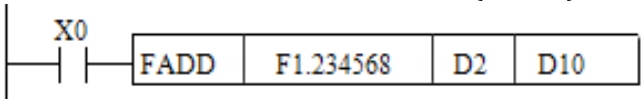
程式範例 (一)

当 X2=On 时，将浮点数(D1, D0) + 浮点数(D3, D2)，结果存放在(D11, D10)中.



程式範例 (二)

当 X0=On 时，将 F1.234568 ＋浮点数(D3, D2)，结果存放在(D11, D10)中.



➢ FSUB

API					
029		FSUB		S1 S2 D	浮点数减法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O	O		O
S2												O	O	O	O		O
D												O	O	O	O		O

操作数使用注意：S1 操作数可使用 F 装置
S2 操作数可使用 F 装置

16 位指令

32 位指令 (7 STEP)
FSUB

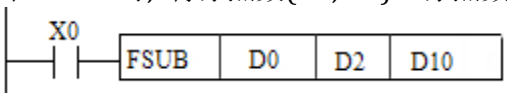
旗标信号：R8
R9
R10

指令說明

S1：被减数. S2：减数. D：差.
S1 所指定的缓存器内容减掉 S2 所指定的缓存器内容，差被存放至 D 所指定的缓存器当中，减算的动作全部以浮点数型态进行.
若运算结果的绝对值大于可表示之最大浮点值，则进位旗标 R10=On.
若运算结果的绝对值小于可表示之最小浮点值，则借位旗标 R9=On.
若运算结果为 0，则零旗标 R8=On.

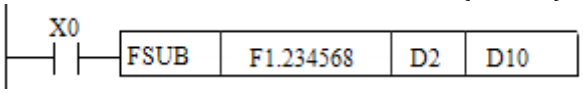
程式範例（一）

当 X0=On 时，将浮点数(D1, D0) - 浮点数(D3, D2)，结果存放在(D11, D10)中.



程式範例（二）

当 X0=On 时，将 F1.234568 - 浮点数(D3, D2)，结果存放在(D11, D10)中.



➢ FMUL

API 030		FMUL		S1 S2 D	浮点数乘法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O	O		O
S2												O	O	O	O		O
D												O	O	O	O		O

操作数使用注意：S1 操作数可使用 F 装置
S2 操作数可使用 F 装置

16 位指令

32 位指令 (7 STEP)
FMUL

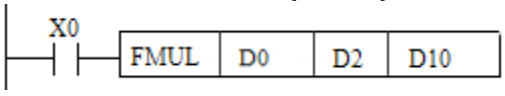
旗标信号：R8
R9
R10

指令說明

S1：被乘数. S2：乘数. D：积.
S1 所指定的缓存器内容乘上 S2 所指定的缓存器内容，积被存放至 D 所指定的缓存器当中，乘算的动作全部以浮点数型态进行.
若运算结果的绝对值大于可表示之最大浮点值，则进位旗标 R10=On.
若运算结果的绝对值小于可表示之最小浮点值，则借位旗标 R9=On.
若运算结果为 0，则零旗标 R8=On.

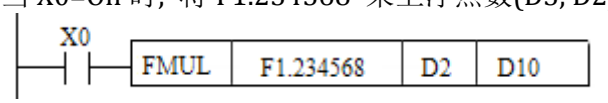
程式範例（一）

当 X0=On 时，将浮点数(D1, D0) 乘上浮点数(D3, D2)，结果存放在(D11, D10)中.



程式範例（二）

当 X0=On 时，将 F1.234568 乘上浮点数(D3, D2)，结果存放在(D11, D10)中.



➢ FDIV

API 031		FDIV		S1 S2 D	浮点数除法

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O	O		O
S2												O	O	O	O		O
D												O	O	O	O		O

操作数使用注意：S1 操作数可使用 F 装置
S2 操作数可使用 F 装置

16 位指令

32 位指令 (7 STEP)
FDIV

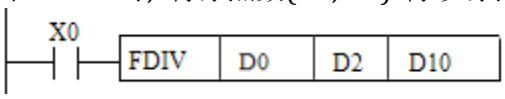
旗标信号：R8
R9
R10
R20

指令說明

S1：被除数. S2：除数. D：商.
S1 所指定的缓存器内容乘上 S2 所指定的缓存器内容，积被存放至 D 所指定的缓存器当中，乘算的动作全部以浮点数型态进行.
除数 S2 的内容若为 0 即被认定为"数值运算错误"，指令不执行，数值运算错误 R20=On，运算错误代码为 02.
若运算结果的绝对值大于可表示之最大浮点值，则进位旗标 R10=On.
若运算结果的绝对值小于可表示之最小浮点值，则借位旗标 R9=On.
若运算结果为 0，则零旗标 R8=On.

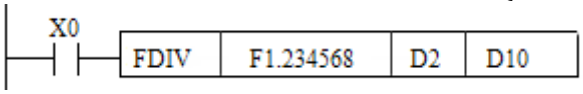
程式範例 (一)

当 X0=On 时，将浮点数(D1, D0) 除以浮点数(D3, D2)，结果存放在(D11, D10)中.



程式範例 (二)

当 X0=On 时，将 F1.234568 除以浮点数(D3, D2)，结果存放在(D11, D10)中.



➢ FINT

API 032		FINT		S D	浮点数→整数

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

16 位指令

32 位指令 (5STEP)
FINT

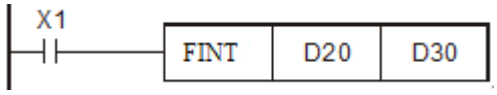
旗标信号：R8

指令說明

S：欲变换之来源装置. D：变换之结果.
S 所指定的缓存器内容以浮点数型态被变换成 BIN 整数暂存于 D 所指定的缓存器当中, BIN 整数浮点数被舍弃.
变换结果若为 0 时，零旗标 R8=On.

程式範例

当 X1=On 时，将浮点数(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.



➢ FDOT

API					
033		FDOT		(S) (D)	整数→浮点数

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 K 装置
D 操作数占用连续 2 点

16 位指令

32 位指令 (5STEP)
FDOT

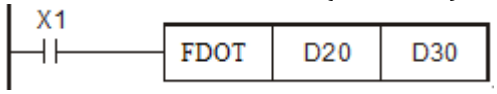
旗标信号：R8

指令說明

S：欲变换之来源装置. D：变换之结果.
S 所指定的缓存器内容以 BIN 整数型态被变换成浮点数暂存于 D 所指定的缓存器当中.
变换结果若为 0 时，零旗标 R8=On.

程式範例

当 X1=On 时，将 BIN 整数(D21, D20) 变换成浮点数将结果存放至(D31, D30)当中.



➢ FRAD

API					
034		FRAD		(S) (D)	角度→径度

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

16 位指令

32 位指令 (5STEP)
FRAD

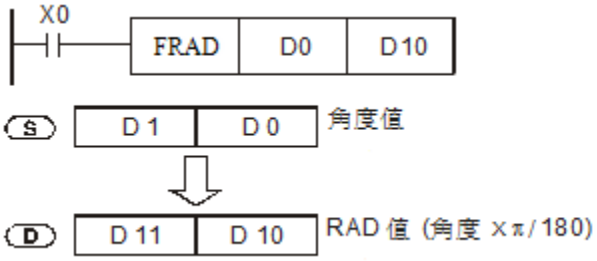
旗标信号：R8

指令說明

S：数据源(角度). D：变换之结果(径度).
使用下列公式将角度转换成径度.
径度＝角度 ×(π/180)
S 所指定的缓存器内容以浮点数格式角度转换为以浮点数格式径度暂存于 D 所指定的缓存器当中.
若转换结果为 0，则零旗标 R8=On.

程式範例

当 X0=On 时，指定浮点数(D1, D0)之角度值，将角度转换成径度值后存于(D11, D10) 当中，内容为浮点数.



➢ FDEG

API		FDEG		S D	径度→角度
035					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		
操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置																	

16 位指令 (5 STEP)
BIN

32 位指令 (5STEP)
DBIN

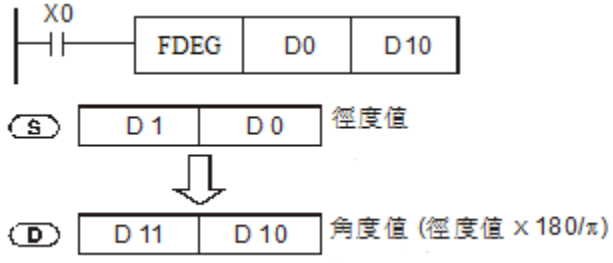
旗标信号：R20

指令說明

S：数据源(径度). D：变换之结果(角度).
使用下列公式将径度转换成角度.
角度＝径度 ×(180/π)
S 所指定的缓存器内容以浮点数格式径度转换为以浮点数格式角度暂存于 D 所指定的缓存器当中.
若转换结果为 0，则零旗标 R8=On.

程式範例

当 X0=On 时，指定二进浮点数(D1, D0)之角度值，将径度值转换成角度后存于(D11, D10) 当中，内容为二进浮点数.



➢ FSIN

API		FSIN		(S) (D)	浮点数 SIN 运算
036					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

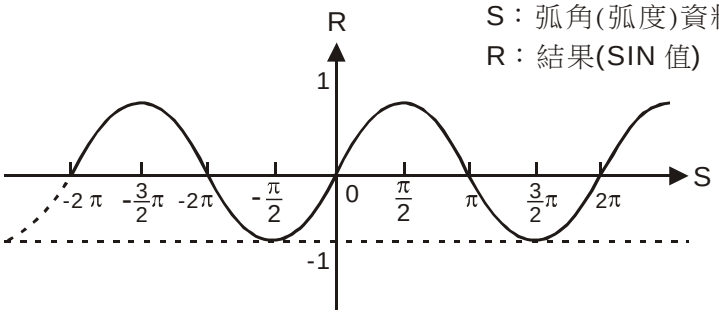
16 位指令

32 位指令 (5STEP)
FSIN

旗标信号：R8

指令說明

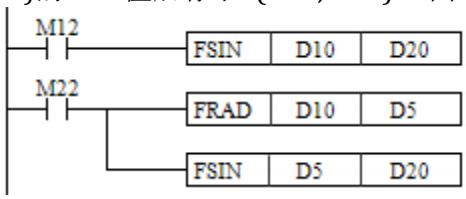
S：指定的来源值(浮点数). D：取 SIN 值结果(浮点数).
将 S 所指定的径度来源值, 求取 SIN 值后存于 D 所指定的缓存器当中.
下图显示弧角与结果的关系：



若转换结果为 0, 则零旗标 R8=On.

程式範例

当 M12=On 时, (D11, D10)之径度(RAD)值求取 SIN 值后存于 (D21, D20) 当中, 内容为浮点数.
当 M22=On 时, (D11, D10)之角度先转为径度(RAD)值暂存于 (D6, D5), 再求取 (D6, D5)的 SIN 值后存于 (D21, D20) 当中, 内容为浮点数.



➢ FCOS

API		FCOS		(S) (D)	浮点数 COS 运算
037					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

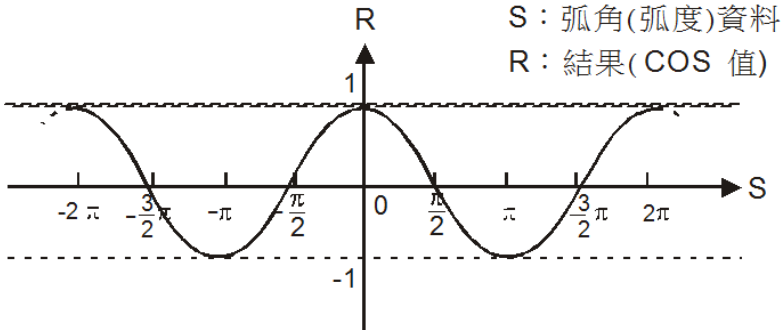
16 位指令

32 位指令 (5STEP)
FCOS

旗标信号：无

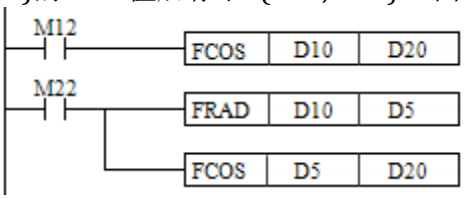
指令說明

S：指定的来源值(浮点数). D：取 COS 值结果(浮点数).
将 S 所指定的径度来源值，求取 COS 值后存于 D 所指定的缓存器当中.
下图显示弧角与结果的关系：



程式範例

当 M12=On 时，(D11, D10)之径度(RAD)值求取 COS 值后存于 (D21, D20) 当中，内容为浮点数.
当 M22=On 时，(D11, D10)之角度先转为径度(RAD)值暂存于 (D6, D5)，再求取 (D6, D5)的 COS 值后存于 (D21, D20) 当中，内容为浮点数.



➢ FTAN

API 038		FTAN		(S) (D)	浮点数 TAN 运算

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

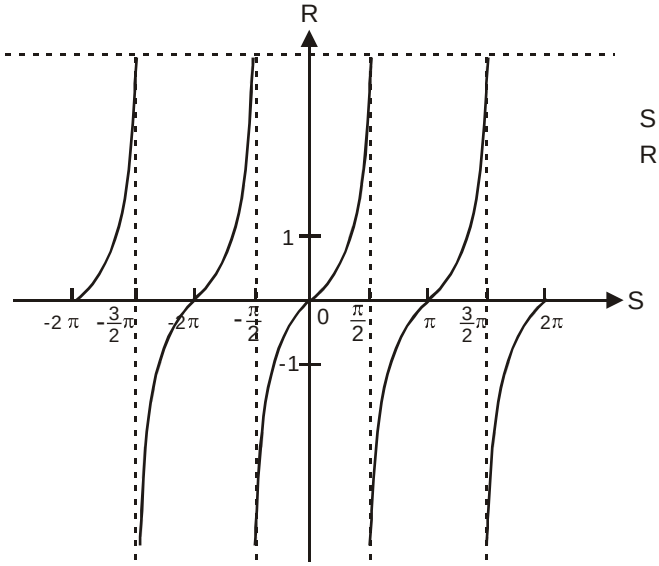
16 位指令

32 位指令 (5STEP)
FTAN

旗标信号：R8

指令說明

S：指定的来源值(浮点数). D：取 TAN 值结果(浮点数).
将 S 所指定的径度来源值，求取 TAN 值后存于 D 所指定的缓存器当中.
下图显示弧角与结果的关系：

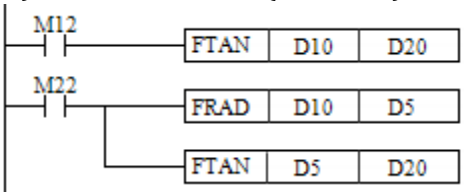


S：弧角(弧度)資料
R：結果(TAN 值)

若转换结果为 0，则零旗标 R8=On.

程式範例

当 M12=On 时, (D11, D10)之径度(RAD)值求取 TAN 值后存于 (D21, D20) 当中，内容为浮点数.
当 M22=On 时, (D11, D10)之角度先转为径度(RAD)值暂存于 (D6, D5)，再求取 (D6, D5)的 TAN 值后存于 (D21, D20) 当中，内容为浮点数.



➢ FASIN

API 039		FASIN		(S) (D)	浮点数 ASIN 运算

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

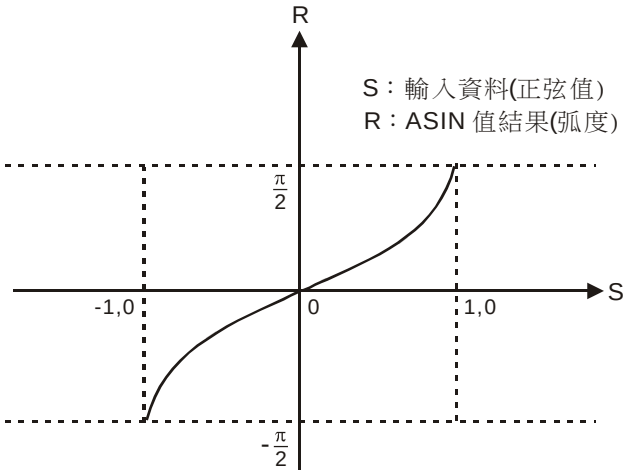
16 位指令

32 位指令 (5STEP)
FASIN

旗标信号：R8
R20

指令說明

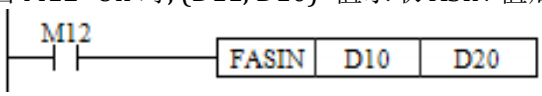
S：指定的正弦值来源(浮点数). D：取 ASIN 值径度结果(浮点数).
ASIN 值=sin-1
下图显示输入数据与结果的关系：



S 操作数指定的正弦值数值只能介于 –1.0 ~ +1.0 之间，若不在此范围内则数值运算错误旗标 R20=On，数值运算错误代码 W20 为 10.
若转换结果为 0，则零旗标 R8=On.

程式範例

当 M12=On 时，(D11, D10) 值求取 ASIN 值后存于 (D21, D20) 当中，内容为浮点数.



➢ FACOS

API		FACOS		(S) (D)	浮点数 ACOS 运算
040					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

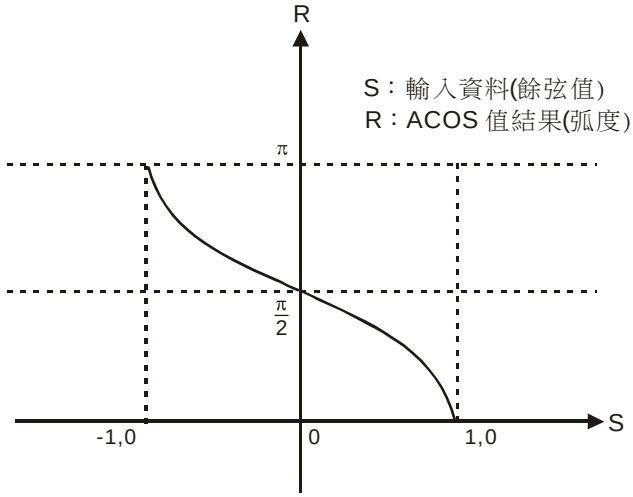
16 位指令

32 位指令 (5STEP)
FACOS

旗标信号：R8
R20

指令說明

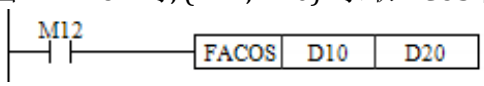
S：指定的余弦值来源(浮点数). D：取 ACOS 值径度结果(浮点数).
ACOS 值=cos-1
下图显示输入数据与结果的关系：



S 操作数指定的余弦值数值只能介于 –1.0 ~ +1.0 之间, 若不在此范围内则数值运算错误旗标 R20=On, 数值运算错误代码 W20 为 11.
若转换结果为 0, 则零旗标 R8=On.

程式範例

当 M12=On 时, (D11, D10) 求取 ACOS 值后存于 (D21, D20) 当中, 内容为浮点数.



➢ FATAN

API		FATAN		(S) (D)	浮点数 ATAN 运算
041					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		
操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置 D 操作数占用连续 2 点																	

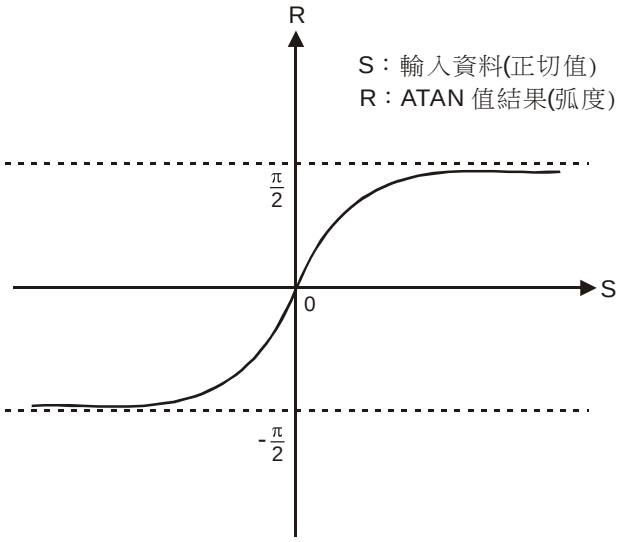
16 位指令

32 位指令 (5STEP)
FATAN

旗标信号：R8

指令說明

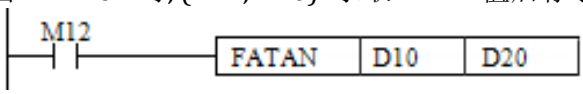
S：指定的正切值来源(浮点数). D：取 ATAN 值径度结果(浮点数).
ATAN 值=tan-1
下图显示输入数据与结果的关系：



若转换结果为 0，则零旗标 R8=On.

程式範例

当 M12=On 时，(D11, D10) 求取 ATAN 值后存于 (D21, D20) 当中，内容为浮点数.



➢ FSQR

API		FSQR		(S) (D)	浮点数开平方根运算
042					

	位装置						字符装置								外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O	O	O	O		
D												O	O	O	O		

操作数使用注意：S 操作数占用连续 2 点、可使用 F 装置
D 操作数占用连续 2 点

16 位指令

32 位指令 (5STEP)
FSQR

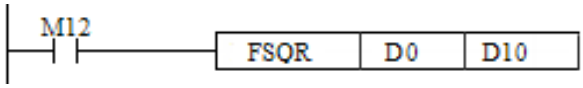
旗标信号：R8

指令說明

S：欲开平方根来源装置(浮点数). D：开平方根之结果(浮点数).
S 所指定的缓存器内容被开平方，所得的结果暂存于 D 所指定的缓存器内容，开平方的动作全部以浮点数型态进行.
S 来源操作数若是指定常数 K 或 H 的话，指令会将该常数变换成浮点数来作运算.
来源运算只有正数有效，负数时，视为"运算错误"，指令不执行，数值运算错误旗标 R20=On，数值运算错误代码 W20 为 12.
若开平方根之结果为 0 时，零旗标 R8=On.

程式範例

当 X0=On 时，将浮点数(D1, D0)取开平方根，将结果存放至(D11, D10)所指定的缓存器当中.



➢ ZRST

API 043		ZRST		D1 D2	区域清除

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
D1		O	O	O	O					O	O	O	O	O			
D2		O	O	O	O					O	O	O	O	O			

操作数使用注意：D1 操作数编号≦ D2 操作数编号
D1、D2 操作数必须指定相同类型装置

16 位指令 (4 STEP)
ZRST

32 位指令

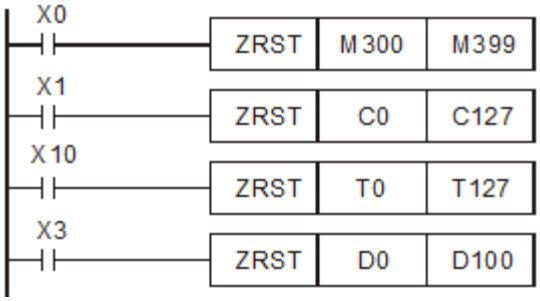
旗标信号：无

指令說明

D1：区域清除起始装置. D2：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 D1 操作数编号＞D2 操作数编号时，只有 D2 指定之操作数被清除.

程式範例

当 X0 为 On 时，辅助继电器 M300 ~ M399 被清除成 Off.
当 X1 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 X10 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 X3 为 On 时，数据缓存器 D0 ~ D100 数据被清除为 0.



➢ DECO

API 044		DECO		S D n	译码器

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S	O	O	O							O	O	O	O	O		O	O
D		O	O							O	O	O	O	O		O	O
n																	

操作数使用注意：S 操作数可使用外部 DX、DY、DAI、DAO 装置与 K 装置、
D 操作数可使用外部 DY、DAO 装置、
n 操作数可使用 K 装置

16 位指令 (5 STEP)
DECO

32 位指令

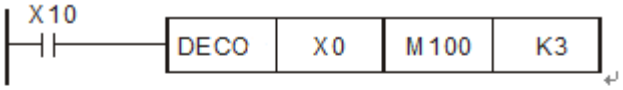
旗标信号：无

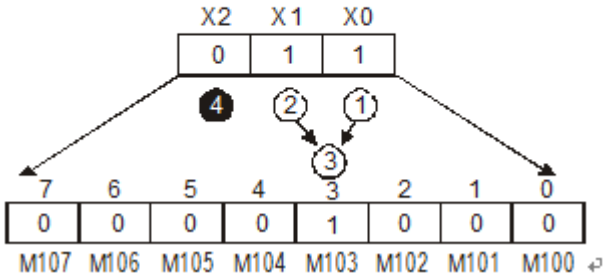
指令說明

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“2 n”位长度的结果存于 D.

程式範例

当 D 为位装置时, n＝1~8, 若 n＝0 或 n＞8 时, 会发生错误.
当 n=8 时, 可做最大解碼 2 8= 256 点. (须注意译码后的装置储存范围, 勿重复使用)
X10＝On 时, DECO 指令将 X0~X2 的内容值译码到 M100~M107.
当数据源为 1+2＝3 时, 从 M100 开始算第 3 个位 M103 设定为 1.
当 DECO 指令执行过后, 而 X10 变为 Off, 已经做解碼输出者照常动作.





➢ ENCO

API		ENCO		S D n	编码器
045					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S	O	O	O							O	O	O	O	O		O	O
D										O	O	O	O	O			O
n																	

操作数使用注意：S 操作数可使用 DX、DY、DAI、DAO 装置与 K 装置、
D 操作数可使用外部 DAO 装置、
n 操作数可使用 K 装置

16 位指令 (11 STEP)
ENCO

32 位指令

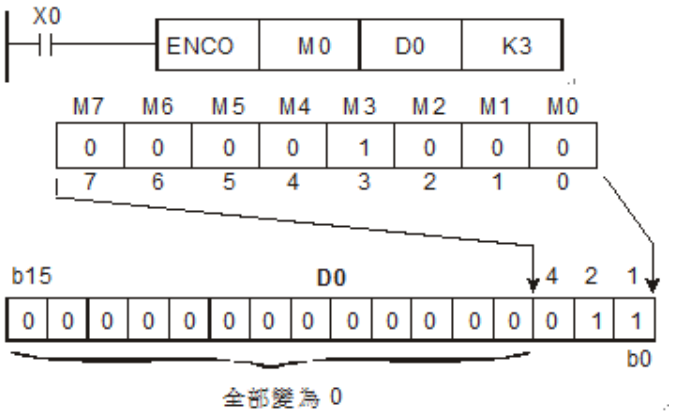
旗标信号：R20

指令說明

S：编码来源装置. D：存放编码结果之装置. n：编码位长度.
来源装置 S 的下位 “2 n” 位长度的数据作编码, 并将结果存于 D .
若数据源 S 有多数位为 1 时, 则处理由高位往低位的第 1 个为 1 之位.
若数据源 S 都没有位为 1 时, 则指令错误旗标 R20=On, 指令错误代码 W20 为 03.

程式範例

S 为位装置时, n＝1~8, 若 n＝0 或 n＞8 时, 会发生错误.
当 n=8 时, 可做 2 8= 256 点编码.
当 X0＝On 时, ENCO 指令将 2 3 位数据(M0~ M7)编码存放于 D0 之下位 3 位(b2~b0) 内, D0 中未被使用之位(b15~b3)全部变为 0.
当 ENCO 指令执行过后, 而 X0 变为 Off 后, D 内资料不变.



➢ BON

API					
046	D	BON		(S) (D) (n)	ON 位判定

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S										O	O	O			O		O
D										O	O	O					O
n												O			O		

操作数使用注意：S 操作数可使用外部 DAI、DAO 装置、
D 操作数可使用外部 DAO 装置、
n 操作数可使用 K 装置

16 位指令 (5 STEP)
BON

32 位指令 (5 STEP)
DBON

旗标信号：无

指令說明

S：来源装置. D：存放判定结果之装置. n：指定判定之位(自 0 开始编号).

程式範例

当 X0=On 时, 若是 D0 的第 15 个位为 "1" 时, M0=On, 为 "0" 时, M0=Off.

X0 变成 Off 时, M0 仍保持之前的状态.

➢ ALT

API				
047		ALT	(D)	ON/OFF 交替

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符

16 位指令 (2 STEP)

D		O	O			O										O		

操作数使用注意：D 操作数可使用外部 DY 装置

ALT

32 位指令

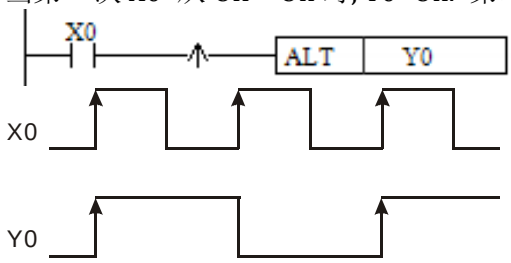
旗标信号：无

指令說明

D：目的地装置.
ALT 指令执行时, D On/Off 交换.

程式範例

当第一次 X0 从 Off→On 时, Y0=On. 第二次 X0 从 Off→On 时, Y0=Off.



➢ RSVP

API					
048		RSVP		S1 S2 D	读取驱动器参数

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O			
S2												O	O	O			
D												O	O	O			

操作数使用注意：S1 操作数可使用 K 装置
S2 操作数可使用 K 装置
S3 操作数占用连续 2 点

16 位指令 (13 STEP)
RSVP

32 位指令

旗标信号：R18

指令說明

S1：读取参数之驱动器轴编号. S2：读取参数编号. D：读取结果.
读取驱动器轴编号 S1 内的伺服参数 ，假设读取伺服参数 P3-21 则 S2 指定值为 0321(十进制), 读取之参数内容存放至 D 所指定的缓存器当中.
S2 格式对应驱动器参数:

S2	AB CD
驅動器參數	PAB - CD

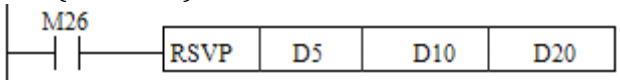
S2 例子

S2	03 21
驅動器參數	P03 21

若联机中断或读取驱动器参数值不正确, 将导致读取参数动作失败, 则文法错误旗标 R18=On, 文法错误码 W18 为 11.

程式範例

当 M26=On 时, 会至十进制(D5) 指定的驱动器轴中读取驱动器参数(D10), 并将结果存放在(D21, D20)中.



➢ WSVP

API		WSVP		(S1) (S2) (D)	写入驱动器参数
049					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S1												O	O	O			
S2												O	O	O			
D												O	O	O			

操作数使用注意：S1 操作数可使用 K 装置
S2 操作数可使用 K 装置
S3 操作数占用连续 2 点，可使用 K 装置

16 位指令 (13 STEP)
WSVP

32 位指令

旗标信号：R18

指令說明

S1：写入参数之驱动器轴编号. S2：写入参数编号. D：写入数据源.
写入驱动器轴编号为 S1，假设写入伺服参数 P3-21 则 S2 设定值为 0321(十进制)，将 D 所指定的缓存器内容写入驱动器参数中.
S2 格式对应驱动器参数:

S2	AB CD
驅動器參數	PAB - CD

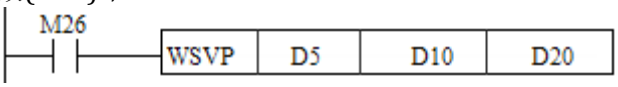
S2 例子

S2	03 21
驅動器參數	P03 21

若联机中断或写入驱动器参数值不正确，将导致写入参数动作失败，则文法错误旗标 R18=On，文法错误码 W18 为 11.

程式範例

当 M26=On 时，至十进制(D5) 指定的驱动器轴中将(D21, D20) 的内容值写入驱动器参数(D10)中.



➢ CKFZ

API		CKFZ		(S) (D)	禁区检查
051					

	位装置						字符装置									外接装置	
	X	Y	M	T	C	R	KnX	KnY	KnM	T	C	D	V	Z	W	位	字符
S												O					
D		O	O													O	

操作数使用注意：D 操作数可使用外部 DY 装置

16 位指令 (5 STEP)
CKFZ

32 位指令

旗标信号：无

指令說明

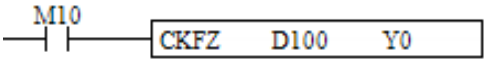
S：禁区坐标与线段坐标数据开始地址. D：判断结果.
由 S 地址开始的数据定义为:

S	定義
Dn	禁區起始X座標
D(n+2)	禁區起始Y座標
D(n+4)	禁區起始Z座標
D(n+6)	禁區結束X座標
D(n+8)	禁區結束Y座標
D(n+10)	禁區結束Z座標
D(n+12)	線段起始X座標
D(n+14)	線段起始Y座標
D(n+16)	線段起始Z座標
D(n+18)	線段結束X座標
D(n+20)	線段結束Y座標
D(n+22)	線段結束Z座標

若禁区数据与线段数据有交集，则 D 输出结果为 ON；若没有交集，则 D 输出结果为 OFF.

程式範例

当 M10=On 时，执行确认设定于 D100 开始数据区块中的空间禁区是否与线段交集. 当交集发生时输出点 Y0 被设为 ON，当没有交集发生时输出点 Y0 被设为 OFF.



5. 动作运行范例

5.1、 运动前准备

- 驱动器站号设定 P3-00

  在 DMCNet 环境中站号 1 必须存在。

- 驱动器控制模式设定 P1-01

  设为 x00b 表示 DMCNet 通讯使用。

- 驱动器通信设置 P3-01

  一般驱动器设为 0203。

  四轴同动驱动器设为 5203。

- 驱动器 CANOPEN 协议设定 P3-10

  设为 11 表示支持完整 CANOpen DS402 协议与通讯错误时马达 Servo OFF。

➢ 驱动器异常解除

- 驱动器异常重置【FAULT RESET】(R592, R593, R594, ...)设为 ON.
- 确认【伺服 FAULT】(R1104, R1105, R1106, ...)解除为 OFF.
- 确认【伺服 WARNING】(R1120, R1121, R1122, ...)解除为 OFF.

➢ 伺服启动

- 将执行运动的伺服轴须先启动, 【SERVO ON】(R576, R577, R578, ...)设为 ON.
- 确认驱动器伺服状态启动, 【SERVO ON】(R1072, R1073, R1074, ...)为 ON.

➢ 驱动器急停解除

- 执行运动的伺服轴【实时停止】(R528, R529, R530, ...)解除为 OFF.
- 确认驱动器伺服急停状态解除,【伺服 QUICK STOP 解除】(R1088, R1089, R1090, ...)为 ON.
- 若驱动器急停状态仍无法解除, 请检查驱动器上 DI 设定.

➢ 其它

- 非执行手轮功能, 确认【启用手轮】(R608, R609, R610, ...)解除为 OFF.

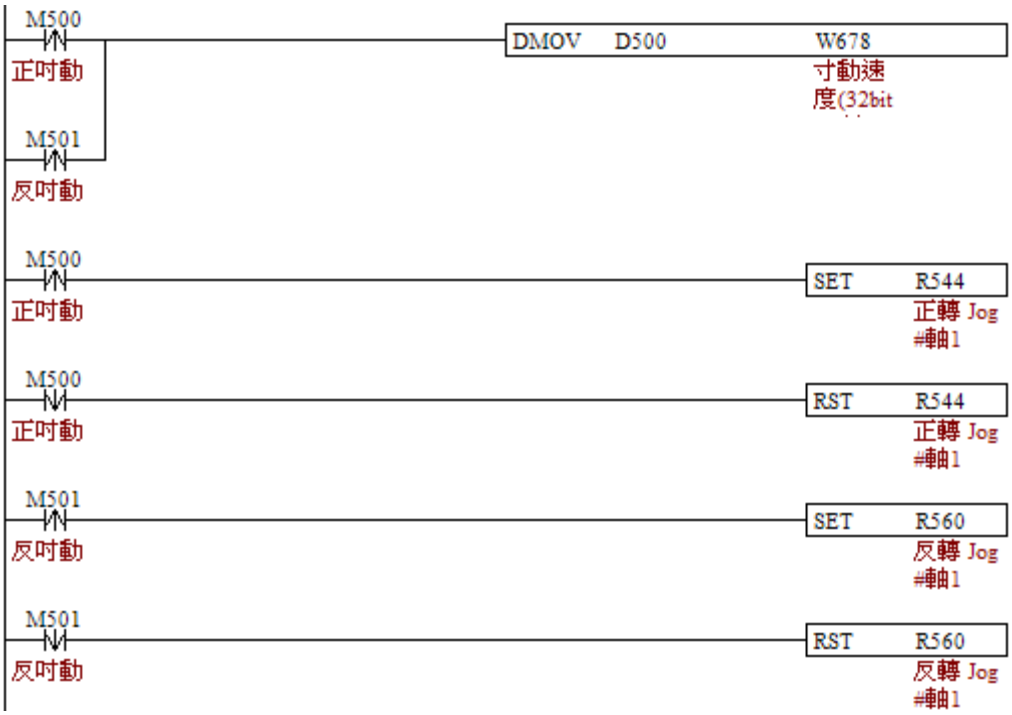
5.2、 寸动

➢ 范例说明

- 使用 M500 作为第一轴正转寸动控制位, M501 为第一轴反转寸动控制位.

➢ 范例程序

- 主程序



a. 当 M500 或 M501 由 Off 状态变为 On 时，将 D500 设定值写至【寸动速度】W678(第一轴寸动速度).

b. M500 变为 On 时，设定【正向寸动】R544 为 On，则第一轴执行正转寸动运动.

c. M500 变为 Off 时，设定【正向寸动】R544 为 Off，则第一轴正转寸动停止.

d. M501 变为 On 时，设定【反向寸动】R560 为 On，则第一轴执行反转寸动运动.

e. M501 变为 Off 时，设定【反向寸动】R560 为 Off，则第一轴反转寸动停止.

➢ 备注

- 寸动速度不能大于最大速度限制，否则触发后驱动器将无反应.
- 同一轴中若同时启动正转寸动与反转寸动则会以先启动者为主.
- 设定【寸动扭力限制】(W682, ...)可实现寸动时的扭力限制保护功能.

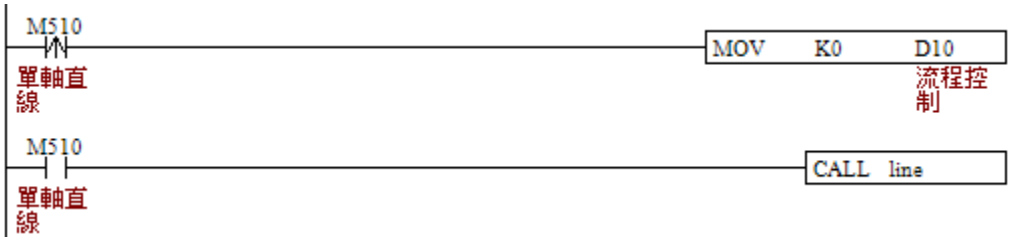
5.3、 单轴直线

➢ 范例说明

- 以 M510 为触发与执行单轴直线运动的致能条件. 启动 M510 后将会开始执行单轴直线运动的参数与命令下达动作.

➢ 范例程序

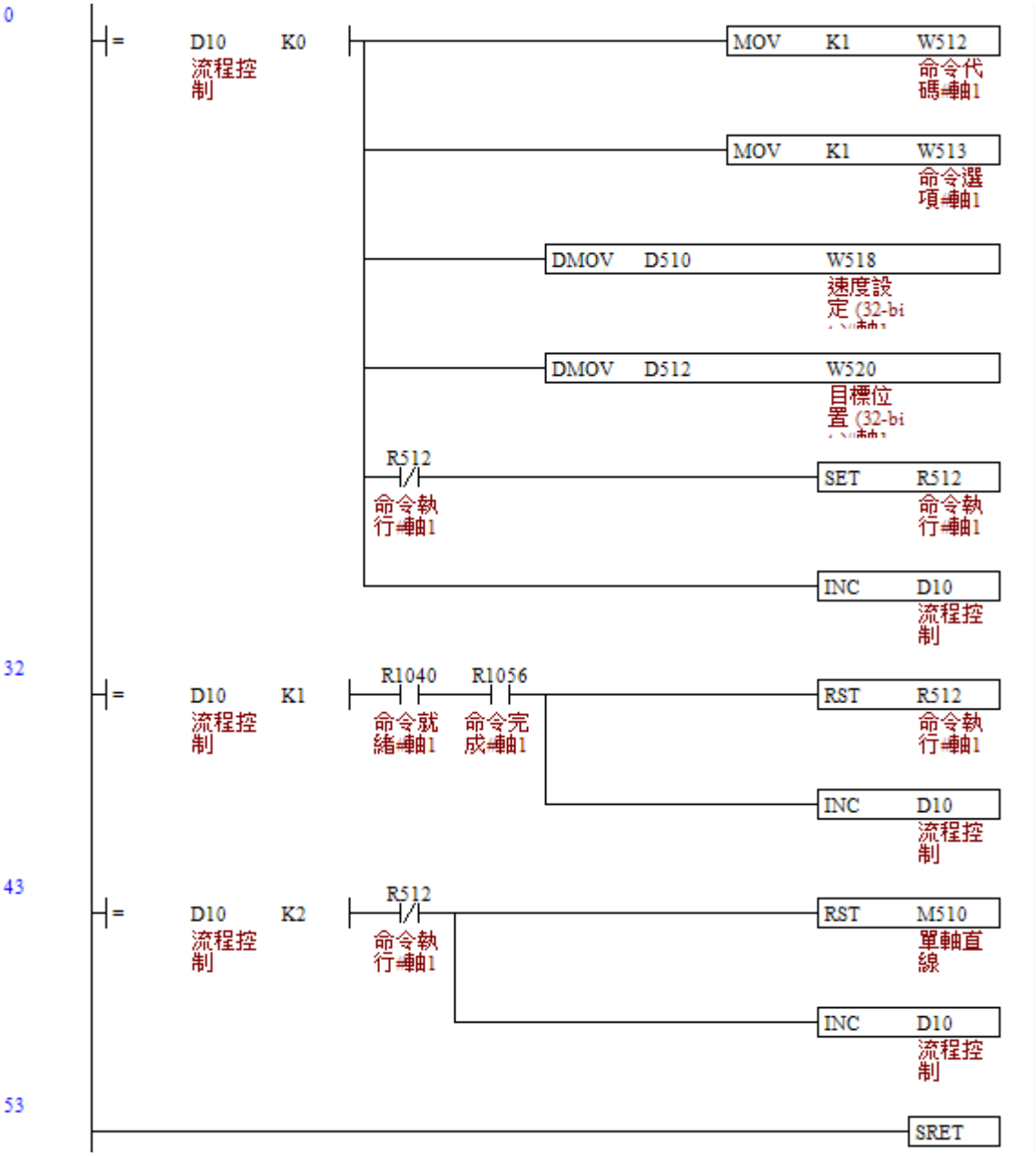
- 主程序



a. 当 M510 由 Off 状态变为 On 时，初始控制状态(D10=0)

b. 当 M510 为 On 时，进入子程序 line 执行单轴直线运动．单轴直线动作完成后 M510 清除.

■ LINE 子程序



a. 在状态 0 时(D10=0)，开始下达动作参数，【命令代码】(W512)设为 1 代表进行直线运动，【命令选项】(W513)设为 1 代表执行第 1 轴的单轴速度选项, D510 写至第一轴【速度设定】(W518), D512 写至第一轴【目标位置】(W520)．最后触发【命令执行】(R512)为 On 并进入状态 1.

b. 在状态 1 时(D10=1)，【命令就绪】(R1040)为 On 代表运动已进行．等待【命令完成】(R1056)为 On 代表运动已完成结束．接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D10=2)，运动已结束．此时确认【命令执行】(R512)已确实清除为 Off 后清除执行旗标 M510 为 Off，控制流程结束.

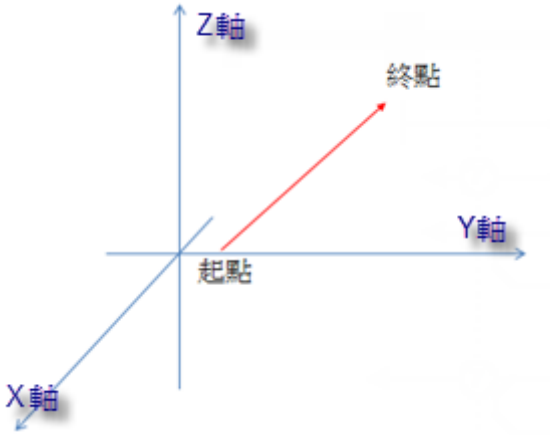
➢ 备注

■ 运动速度若大于【最大速度限制】，将以限制的最大速度运动.

5.4、 三轴同动直线

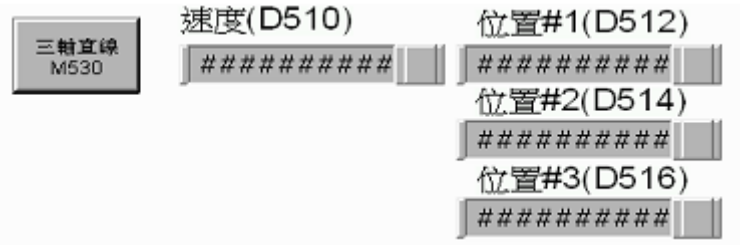
➢ 范例说明

■ 以 M530 为触发与执行三轴同动运动的致能条件. 启动 M530 后将会开始执行三轴直线同动运动的参数与命令下达动作. 其三轴直线运动路径如下图所示:

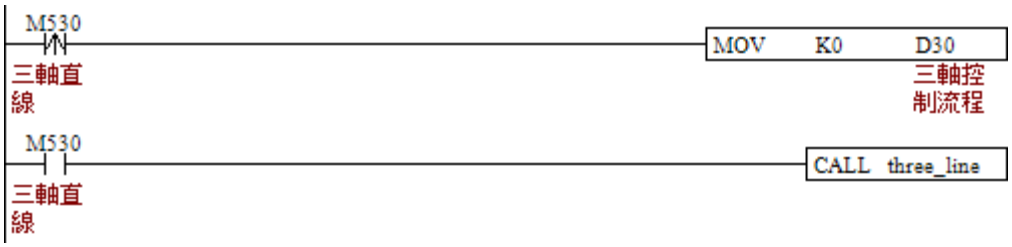


➢ 范例程序

■ 人机画面



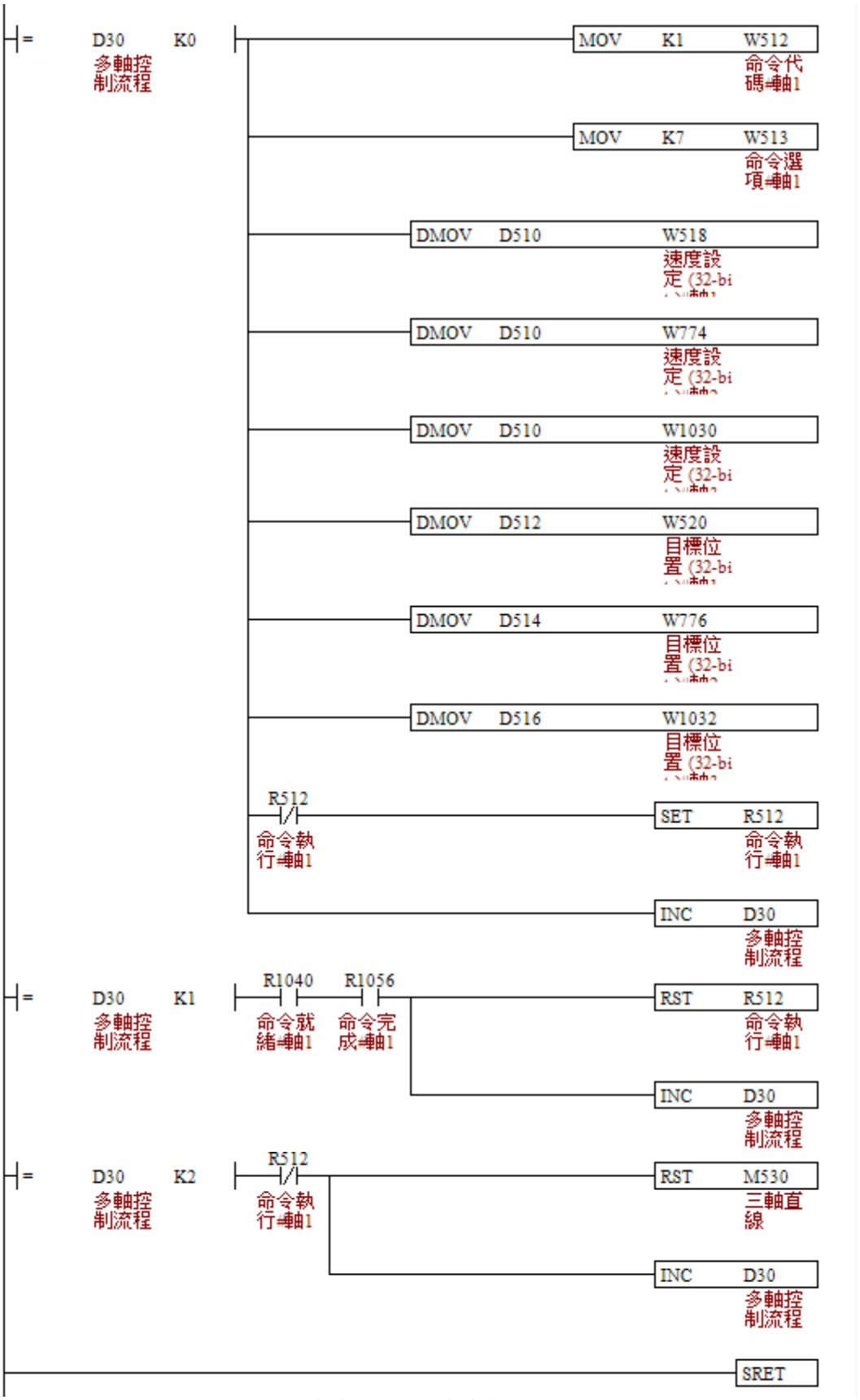
■ 主程序



a. 当 M530 由 Off 状态变为 On 时，初始控制状态(D30=0)

b. 当 M530 为 On 时，进入子程序 three_line 执行三轴同动运动. 动作完成后 M530 自动清除.

■ THREE_LINE 子程序



a. 在状态 0 时(D30=0), 开始下达命令参数, 【命令代码】(W512)设为 1 代表进行直线运动, 【命令选项】(W513)设为 7 代表执行轴 1, 轴 2, 轴 3 的三轴运动, D510 写至各轴【速度设定】(W518, W744, W1030), D512 为轴 1【目标位置】(W520), D514 为轴 2【目标位置】(W776), D516 为轴 3【目标位置】(W1032). 最后触发【命令执行】(R512)为 On 并进入状态 1.

b. 在状态 1 时(D30=1)，【命令就绪】(R1040)为 On 代表三轴同动运动已进行，并等待【命令完成】(R1056)为 On 代表运动已完成结束．接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D30=2)，运动结束．此时确认【命令执行】(R512)已确实清除为 Off 后清除执行旗标 M530 为 Off，控制流程结束．

➢ 备注

- 运动速度若大于【最大速度限制】，将以限制的最大速度运动．
- 在非向量速度或指定轴速度的多轴直线运动中，驱动器会依照最长行程轴为速度基准，并调整其它轴速度以达成直线同动运动．

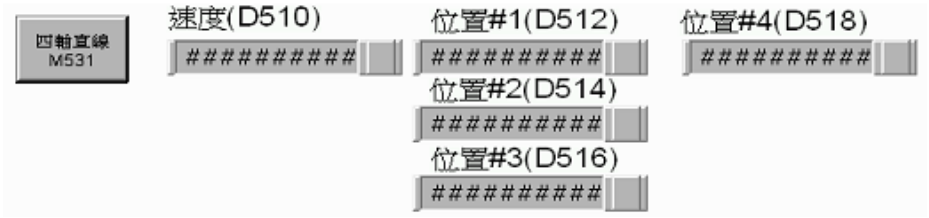
5.5、 四轴同动直线 (特殊型)

➢ 范例说明

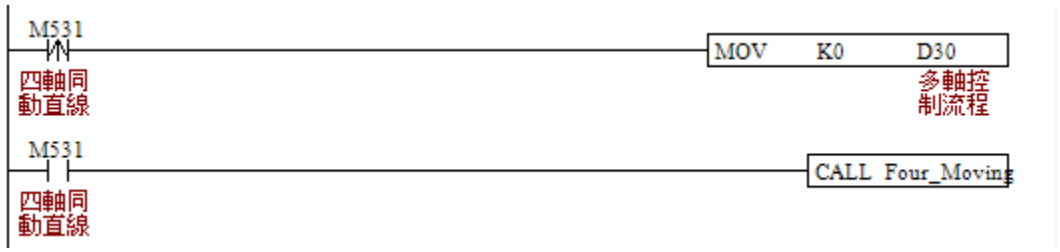
- 以 M531 为触发与执行四轴同动直线运动的致能条件．启动 M531 后将会开始执行四轴同动直线运动的参数与命令下达动作．

➢ 范例程序

- 人机画面



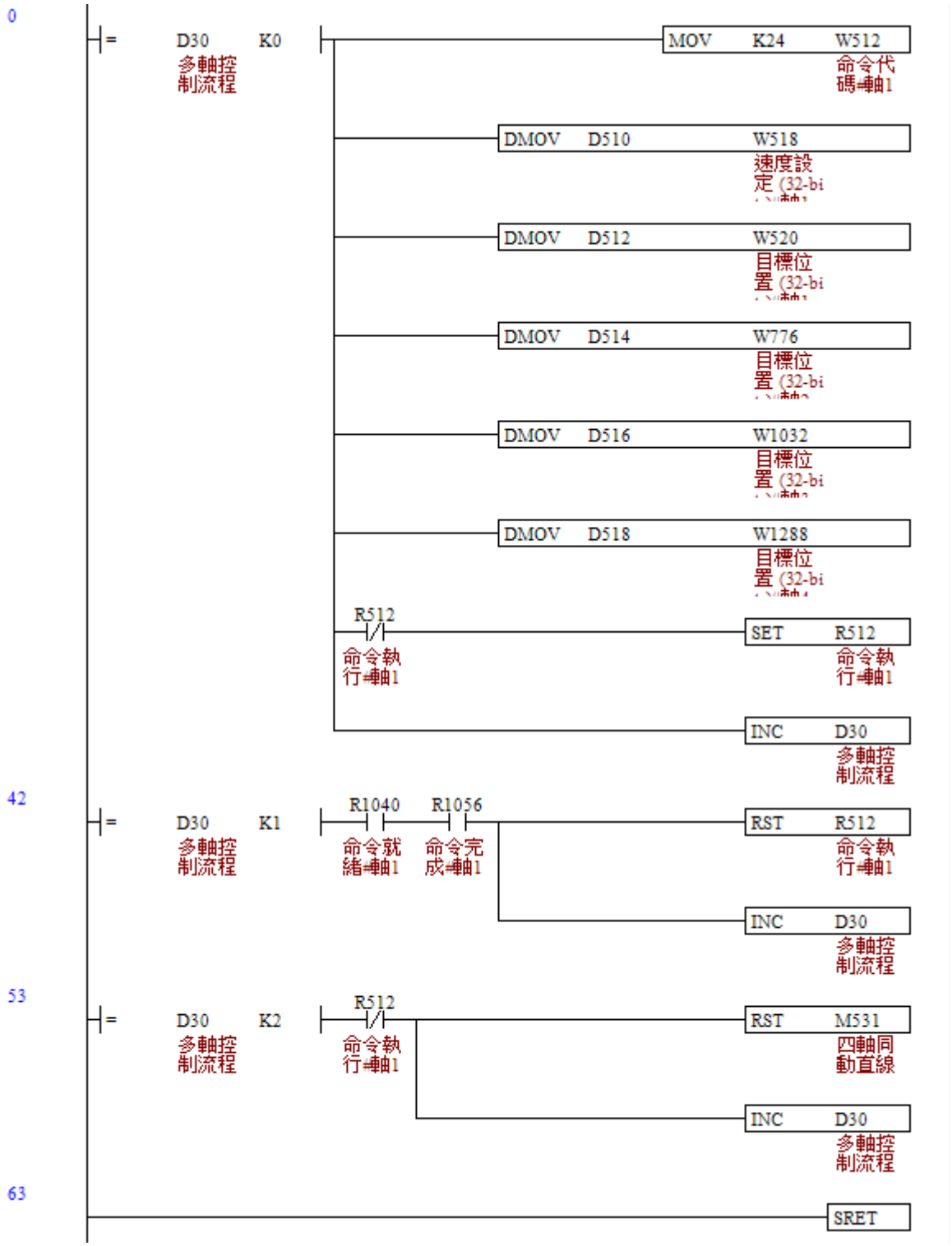
- 主程序



a. 当 M531 由 Off 状态变为 On 时，初始控制状态(D30=0)

b. 当 M531 为 On 时，进入子程序 Four_Moving 执行四轴同动运动．四轴同动动作完成后 M531 自动清除．

- FOUR_MOVING 子程序



a. 在状态 0 时(D30=0), 开始下达动作参数, 【命令代码】(W512)设为 24 代表进行四轴同动直线运动, D510 只需要写至轴 1【速度设定】(W518), D512 写至轴 1【目标位置】(W520), D514 写至轴 2【目标位置】(W776), D516 写至轴 3【目标位置】(W1032), D518 写至轴 4【目标位置】(W1288). 最后触发【命令执行】(R512)为 On 并进入状态 1.

b. 在状态 1 时(D30=1), 若【命令就绪】(R1040)为 On 代表四轴同动直线运动已进行. 等待【命令完成】(R1056)为 On 代表运动已完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D30=2), 运动已结束. 此时确认【命令执行】(R512)已确实清除为 Off 后清除执行旗标 M531 为 Off, 控制流程结束.

➢ 备注

- 四轴同动直线补间为搭配特殊 ASDA-M 驱动器功能，将命令下达给驱动器后，四轴同动驱动器即执行四轴间的同动差补计算，使用架构与设定参照附录 C.
- 运动速度若大于【最大速度限制】，将以限制的最大速度运动.
- 命令速度下达仅需给轴 1 运动速度.

5.6、 正转速度

➢ 范例说明

- 以 M400 为触发与执行正转速度运动的致能条件. 启动 M400 后将会执行正转速度运动的参数与命令下达动作.

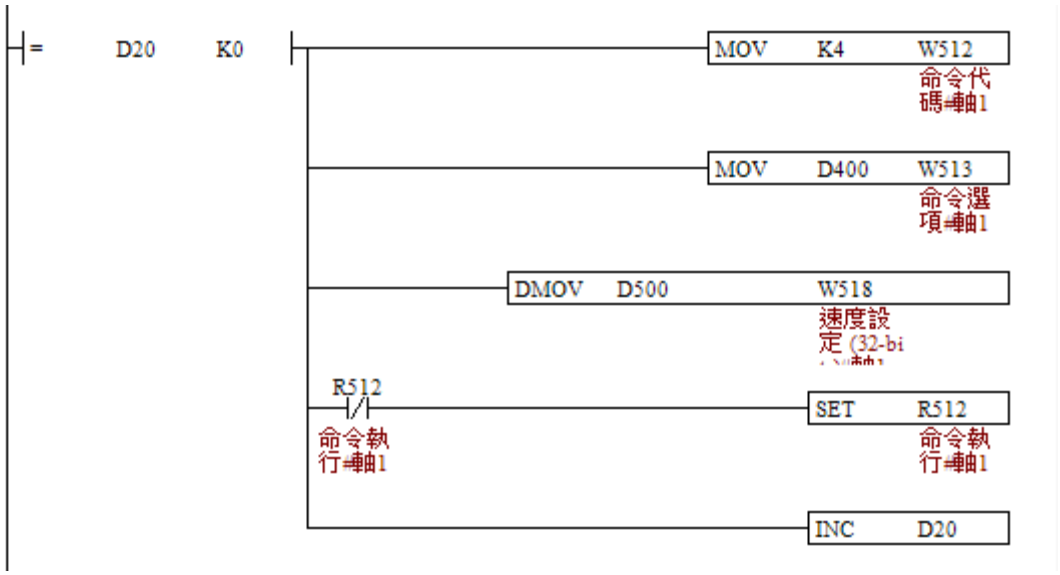
➢ 范例程序

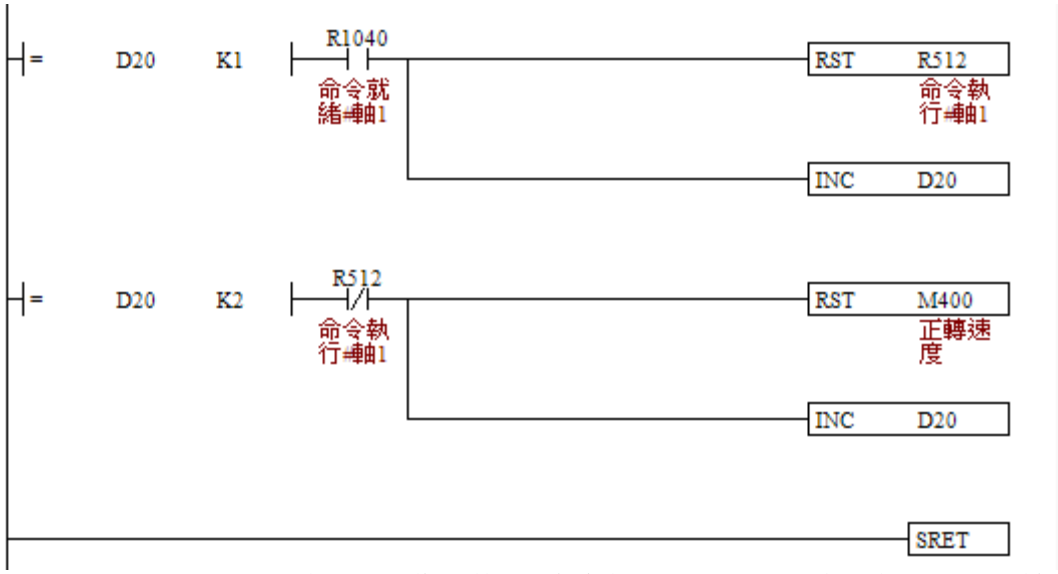
- 主程序

a. 当 M400 由 Off 状态变为 On 时，初始控制状态(D20=0)，同时重置【命令控制】(R512).

b. 当 M400 为 On 时，进入子程序 P_Speed 执行多轴正转速度运动. 正转速度动作完成后 M400 自动清除.

- P_SPEED 子程序





a. 在状态 0 时(D20=0), 开始下达动作参数, 【命令代码】(W512)设为 4 代表执行正转速度运动, 并将 D400 写至【命令选项】(W513)选择启动正转速度轴(Bit 0 为 On 代表轴 1 启动, Bit 1 为 On 代表轴 2 启动, 以此类推), D500 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 并进入状态 1.

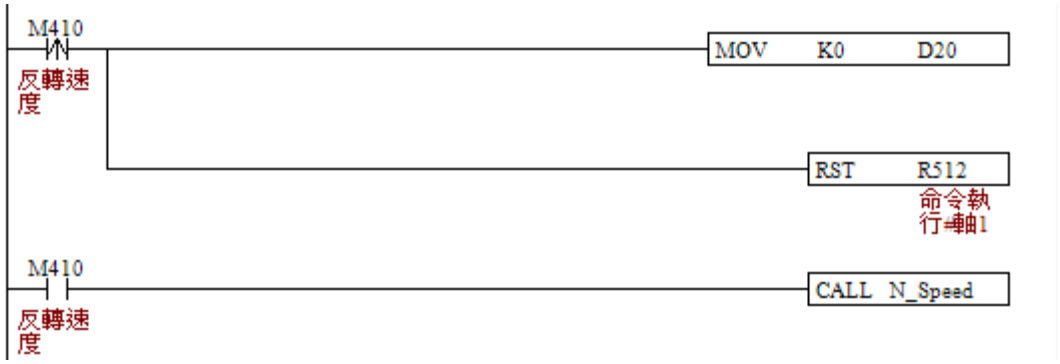
b. 在状态 1 时(D20=1), 【命令就绪】(R1040)为 On 代表正转运动进行. 由于并无【命令完成】旗标变为 On 的状态, 因此可直接清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D20=2), 确认【命令执行】(R512)已确实清除为 Off 后清除执行旗标 M400 为 Off, 控制流程结束.

- 备注
  - 运动速度若大于【最大速度限制】, 将以限制的最大速度运动.
  - 需以减速停止运动指令或【实时停止】旗标停止正转速度运动.
  - 若已经执行正转速度或反转速度运动, 再下达运动命令会造成命令错误.

5.7、 反转速度

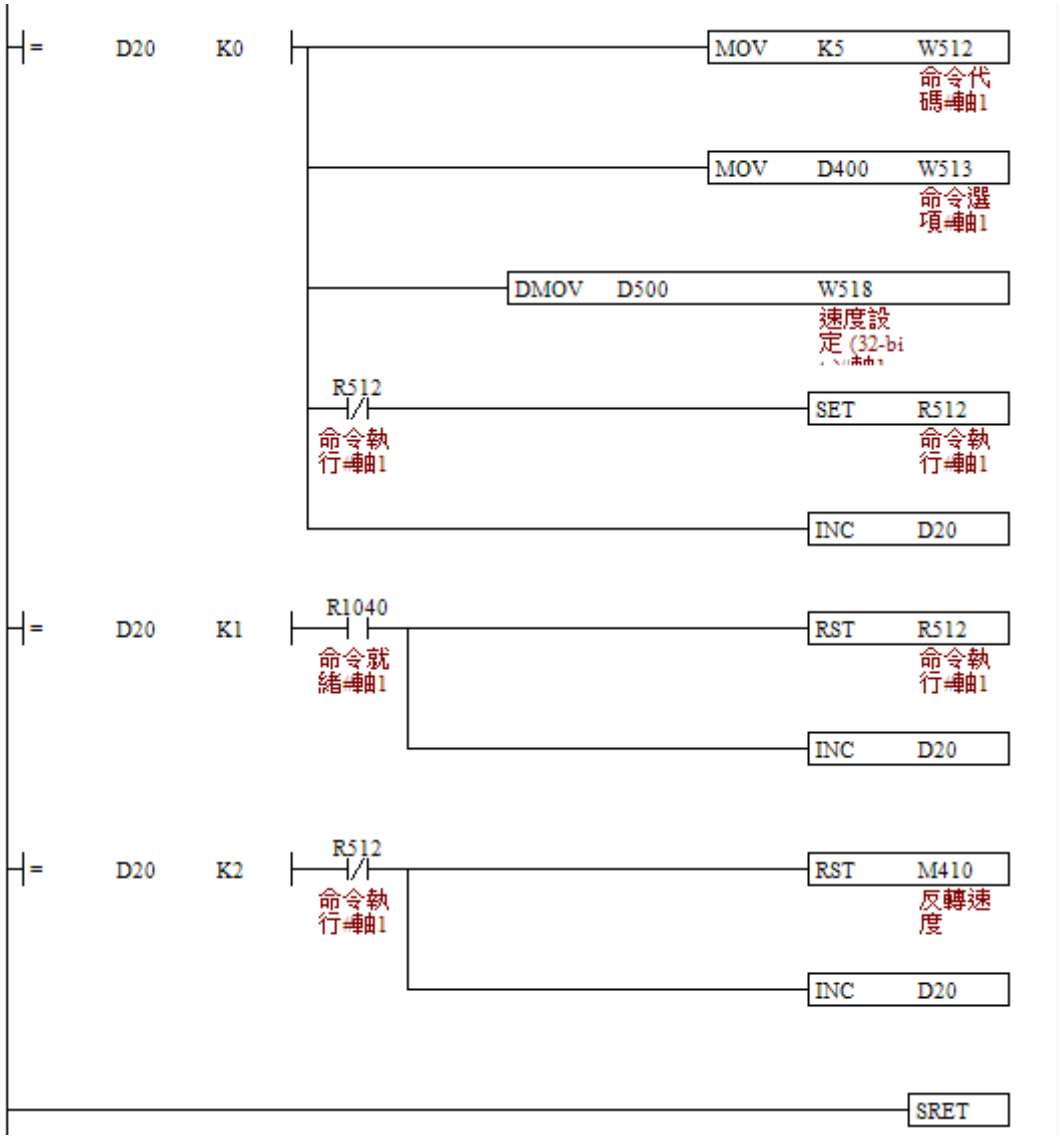
- 范例说明
  - 以 M410 为触发与执行反转速度运动的致能条件. 启动 M410 后将会执行反转速度运动的参数与命令下达动作.
- 范例程序
  - 主程序



a. 当 M410 由 Off 状态变为 On 时，初始控制状态(D20=0)，同时重置【命令控制】(R512).

b. 当 M410 为 On 时，进入子程序 N_Speed 执行多轴反转速度运动. 反转速度动作完成后 M410 自动清除.

■ N_SPEED 子程序



a. 在状态 0 时(D20=0)，开始下达动作参数，【命令代码】(W512)设为 5 代表执行反转速度运动，并将 D400 写至【命令选项】(W513)选择启动反转速度轴(Bit 0 为 On 代表轴 1 启动, Bit 1 为 On 代表轴 2 启动，以此类推), D500 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 并进入状态 1.

b. 在状态 1 时(D20=1)，【命令就绪】(R1040)为 On 代表反转运动进行．由于并无【命令完成】旗标变为 On 的状态，因此可直接清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D20=2)，确认【命令执行】(R512)已确实清除为 Off 后清除执行旗标 M410 为 Off，控制流程结束．

➢ 备注

- 运动速度若大于【最大速度限制】，将以限制的最大速度运动．
- 需以减速停止运动指令或【实时停止】旗标停止反转速度运动．
- 若已经执行正转速度或反转速度运动，再下达运动会造成命令错误．

5.8、 减速停止

➢ 范例说明

- 以 M420 为触发与执行减速停止运动的致能条件．启动 M420 后将会执行减速停止运动的参数与命令下达动作．

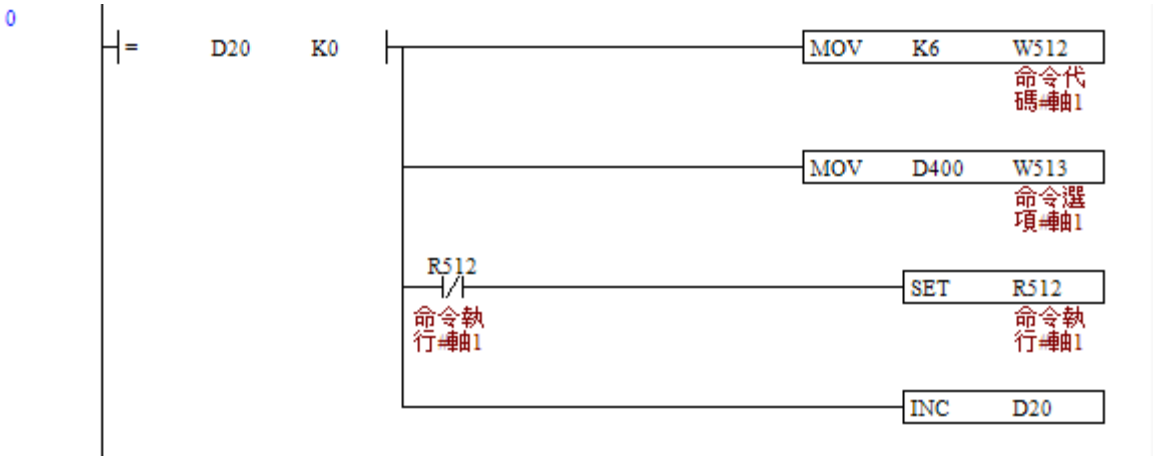
➢ 范例程序

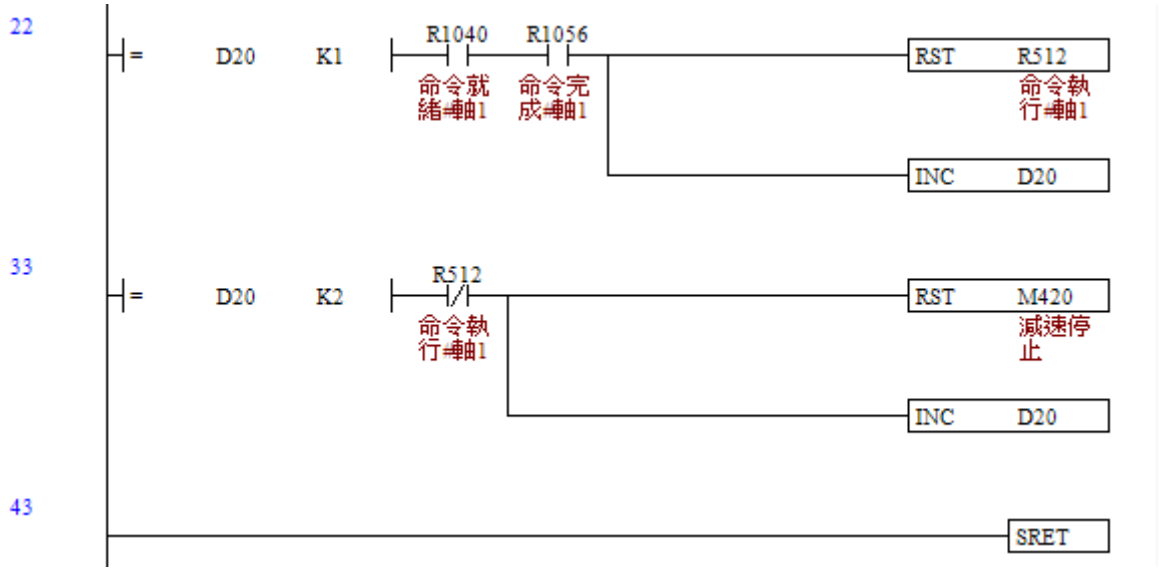
- 主程序

a. 当 M420 由 Off 状态变为 On 时，初始控制状态(D20=0)，同时重置【命令控制】(R512)．

b. 当 M420 为 On 时，进入子程序 STOP_Speed 执行减速停止运动．减速停止动作完成后 M420 自动清除．

- STOP_SPEED 子程序





a. 在状态 0 时(D20=0)，开始下达动作参数，【命令代码】(W512)设为 6 代表减速停止运动，并透过 D400 设定【命令选项】(W513)以选择停止轴(Bit 0 为 On 代表轴 1 启动，Bit 1 为 On 代表轴 2 启动，以此类推)．最后触发【命令执行】(R512)为 On 并进入状态 1.

b. 在状态 1 时(D20=1)，【命令就绪】(R1040)为 On 代表开始减速停止运动，当【命令完成】(R1056)为 On 时则运动已停止．接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D20=2)，确认【命令执行】(R512)清除为 Off 后，清除执行旗标 M420 为 Off，控制流程结束.

➢ 备注

■ 减速停止运动使用【停止命令减速时间】(W670, ...)做为减速时间依据.

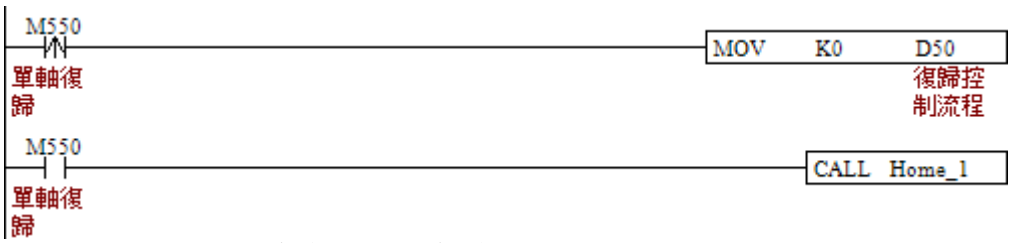
5.9、 原点复归

➢ 范例说明

■ 以 M550 为触发与执行原点复归运动的致能条件．启动 M550 后开始执行复归运动的参数与命令下达动作.

➢ 范例程序

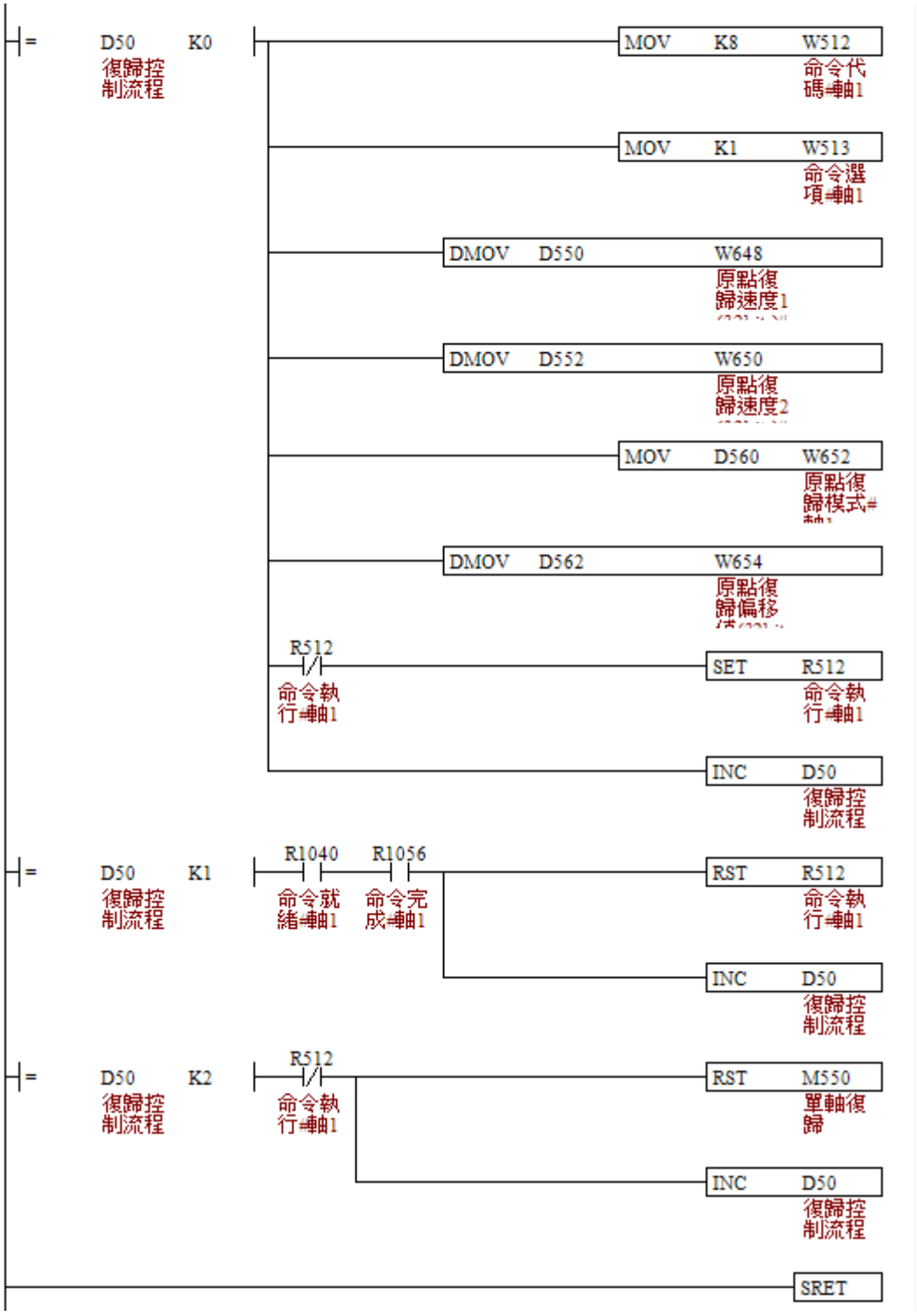
■ 主程序



a. 当 M550 由 Off 状态变为 On 时，初始控制状态(D50=0)

b. 当 M550 为 On 时，进入子程序 Home_1 执行单轴复归运动．复归动作完成后 M550 自动清除.

■ HOME_1 子程序



a. 在状态 0 时(D50=0)，开始下达命令参数，将【命令代码】(W512)设为 8 代表进行复归运动,【命令选项】(W513)设为 1 代表执行轴 1 运动，运动速度 D550 设定值写至【原点复归第一段速度】(W648)，运动速度 D552 设定值写至【原点复归第二段速度】(W650)，复归模式设定 D560 写至【原点复归模式】(W652)，复归偏移量设定 D562 写至【原点复归偏移量】(W654). 最后触发【命令执行】(R512)为 On 并进入状态 1.

b. 在状态 1 时(D50=1)，【命令就绪】(R1040)为 On 代表运动已进行. 等待【命令完成】(R1056)为 On 代表运动完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D50=2)，运动结束．此时确认【命令执行】(R512)已确实清除为 Off 后，清除执行旗标 M550 为 Off，控制流程结束．

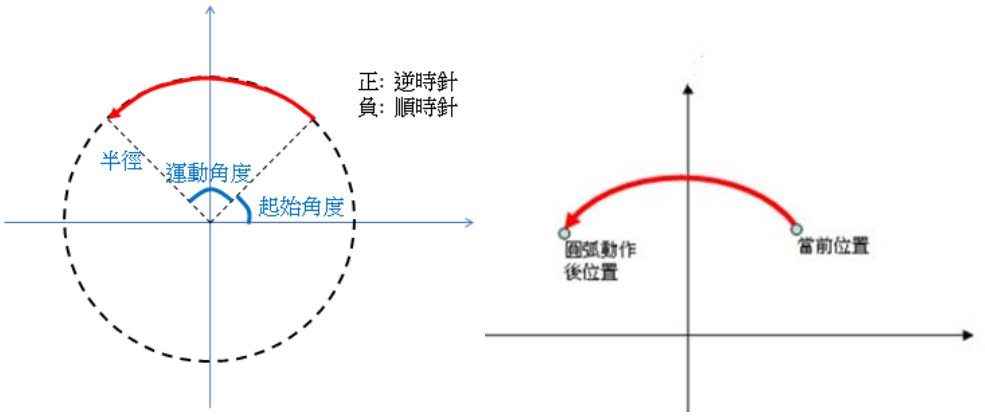
➢ 备注

- 运动速度若大于【最大速度限制】，将以限制的最大速度运动．
- 可透过【命令选项】设定对应的位以启动多轴的复归动作执行．

5.10、圆弧:半径角度

➢ 范例说明

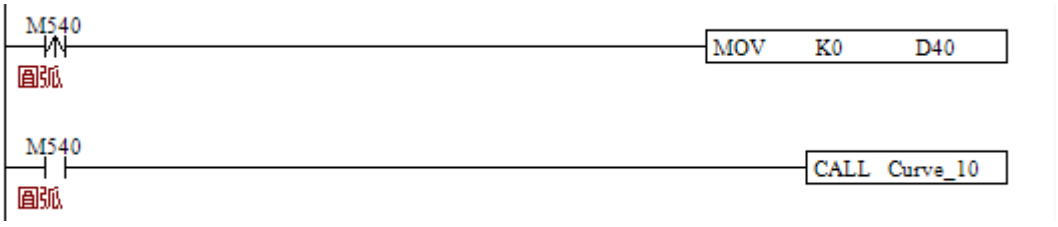
- 以 M540 为触发与执行圆弧运动的致能条件．启动 M540 后开始执行圆弧运动的参数与命令下达动作．
- 圆弧运动需要下达【半径】，【初始角度】与【运动角度】三个参数．若数据参数起始地址为 D1000，则 D1000 代表圆弧【半径】(PUU)，D1002 代表【初始角度】，D1004 代表【运动角度】．需特别注意角度单位为 0.5 度，也就是设定值为 180 代表 90 度角．根据参数设定其运动路径如下图所示:



➢ 范例程序

- 人机画面

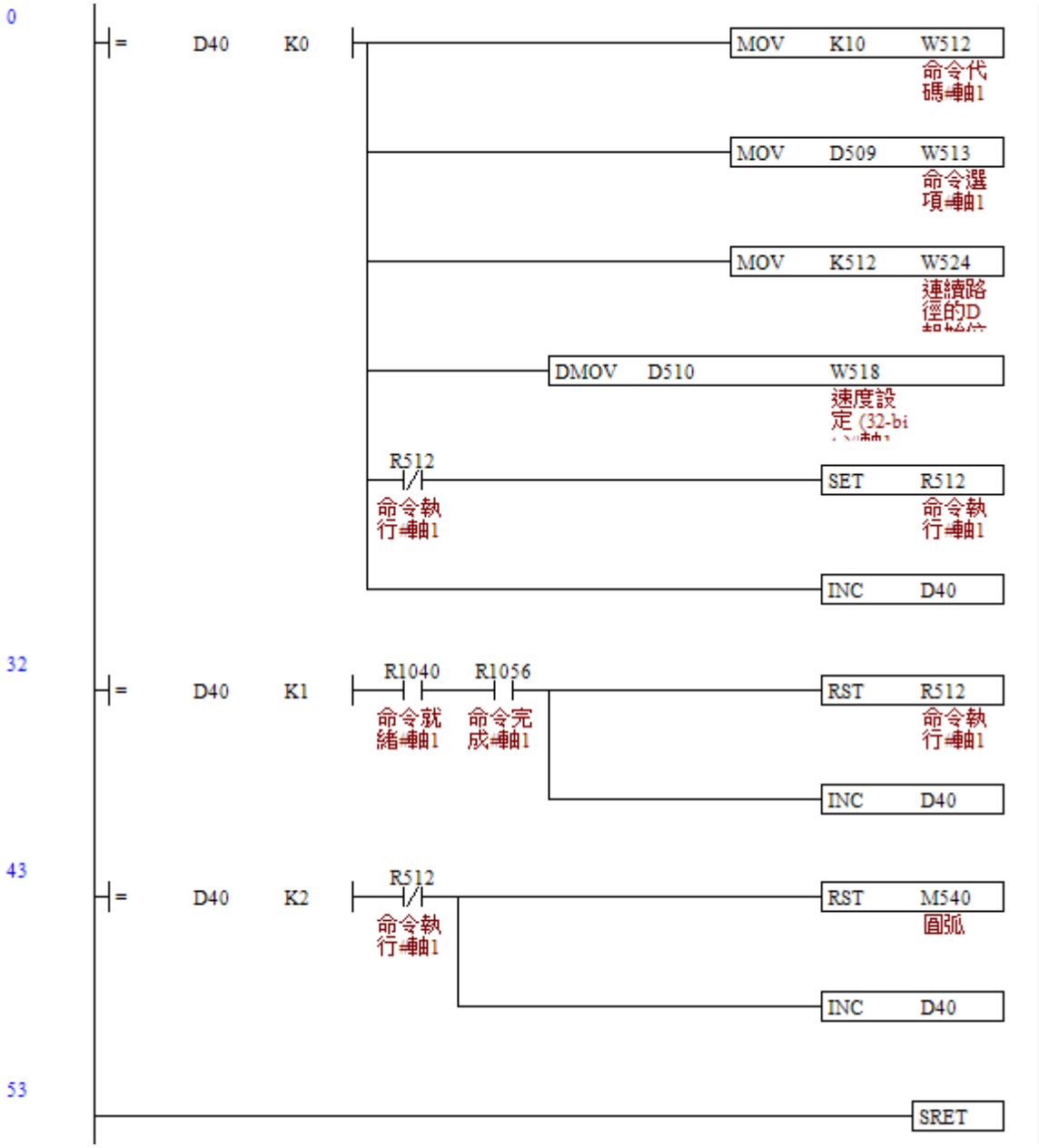
- 主程序



a. 当 M540 由 Off 状态变为 On 时，初始控制状态(D40=0)

b. 当 M540 为 On 时，进入子程序 Curve_10 执行圆弧动作. 圆弧动作完成后 M540 清除.

■ CURVE_10 子程序



a. 在状态 0 时(D40=0)，开始下达动作参数，【命令代码】(W512)设为 10 代表进行圆弧运动, D509 写至【命令选项】(W513)代表执行圆弧的轴选项，其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补. 【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至轴 1【速度设定】(W518). 最后触发【命令执行】(R512)为 On 后将会把运动参数与圆弧数据:【半径】D512，【起始角度】D514 与【运动角度】D516 写入驱动器并进入状态 1.

b. 在状态 1 时(D40=1)，【命令就绪】(R1040)为 On 代表圆弧运动进行. 等待【命令完成】(R1056)为 On 代表运动完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D40=2)，运动已结束. 此时确认【命令执行】(R512)已清除为 Off 后，清除执行圆弧旗标 M540 为 Off，控制流程结束.

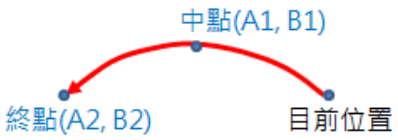
➢ 备注

- 圆弧运动是一次下达三轴的动作命令，因此只能下达给 ASDA_M 驱动器执行三轴的圆弧补间运动.
- 三轴驱动器中当有两轴在执行圆弧运动时，另一轴无法再执行其它的运动命令.
- 参数中角度设为正值，代表逆时针方向；相反地，角度设为负值，代表顺时针方向

5.11、 圆弧:中点终点

➢ 范例说明

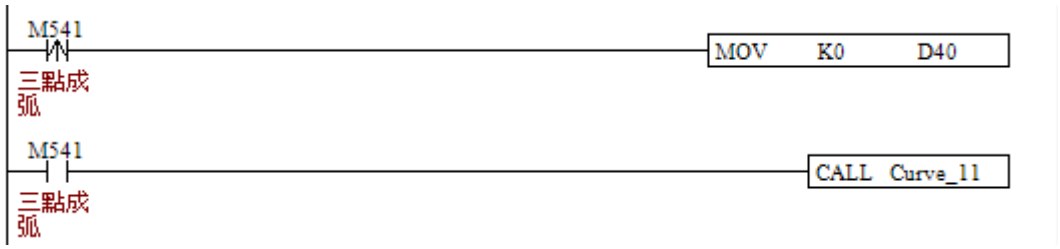
- 以 M541 为触发与执行圆弧运动的致能条件. 启动 M541 后开始执行圆弧运动的参数与命令下达动作.
- 圆弧：中点终点运动需下达【中点坐标 1】(A1)，【中点坐标 2】(B1)，【终点坐标 1】(A2)与【终点坐标 2】(B2)等四个参数. 若数据参数起始地址为 D1000，则 D1000 代表【中点坐标 1】(PUU), D1002 代表【中点坐标 2】(PUU), D1004 代表【终点坐标 1】(PUU), D1006 代表【终点坐标 2】(PUU). 根据参数其运动路径如下图所示:



➢ 范例程序

- 人机画面

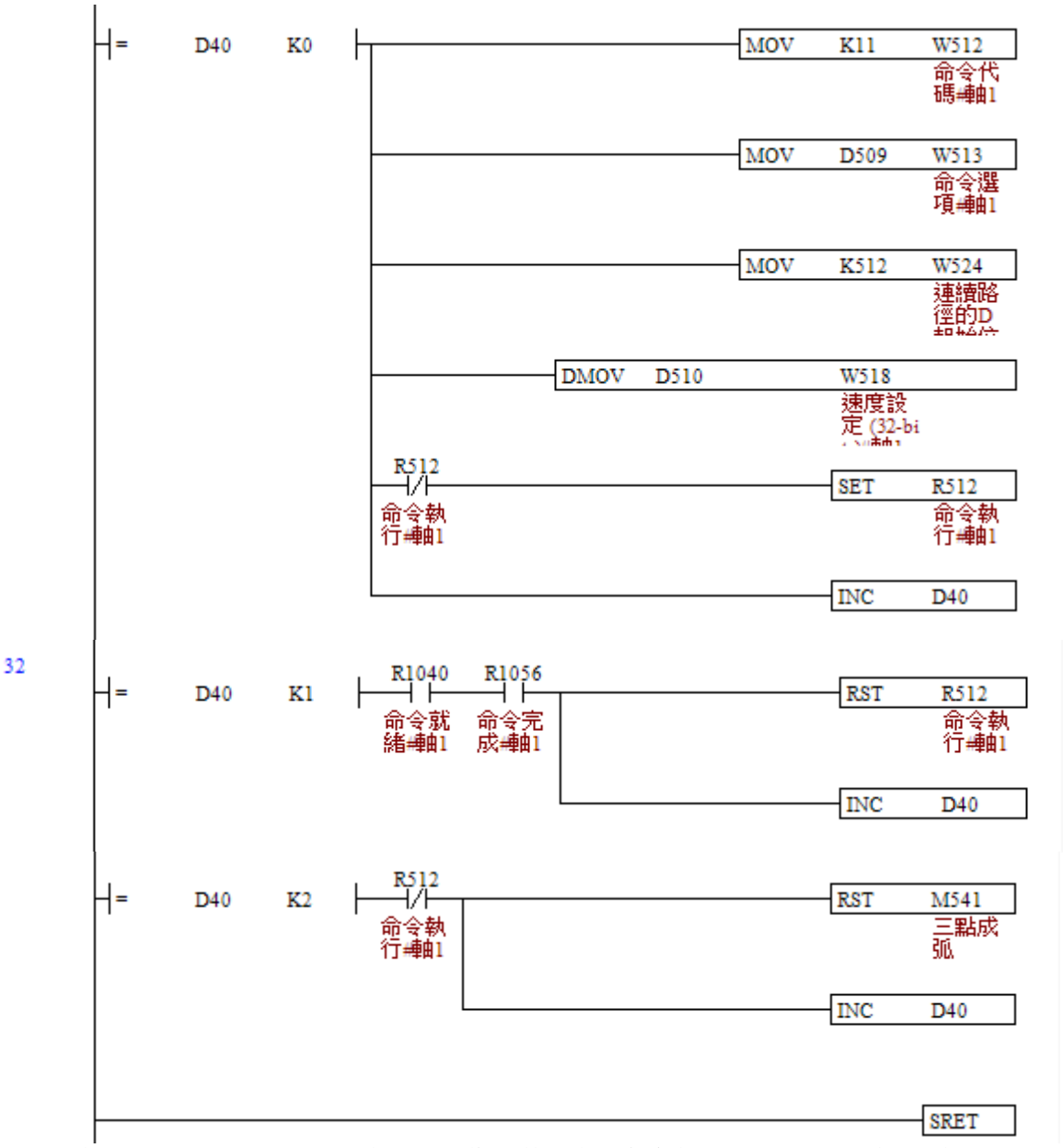
- 主程序



a. 当 M541 由 Off 状态变为 On 时，初始控制状态(D40=0)

b. 当 M541 为 On 时，进入子程序 Curve_11 执行三点成弧动作. 动作完成后 M541 清除.

- CURVE_11 子程序



a. 在状态 0 时(D40=0), 开始下达动作参数, 将【命令代码】(W512)设为 11 代表进行圆弧: 中点&终点模式运动, D509 写至【命令选项】(W513)代表执行圆弧轴选项, 其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补. 【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 后开始将运动参数: 【中点坐标 1】(D512), 【中点坐标 2】(D514), 【终点坐标 1】(D516)与【终点坐标 2】(D518)写入驱动器并进入状态 1.

b. 在状态 1 时(D40=1),【命令就绪】(R1040)为 On 代表圆弧运动进行中. 等待【命令完成】(R1056)为 On 代表运动完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D40=2), 运动已结束. 此时确认【命令执行】(R512)已清除为 Off 后, 清除执行圆弧旗标 M541 为 Off, 控制流程结束.

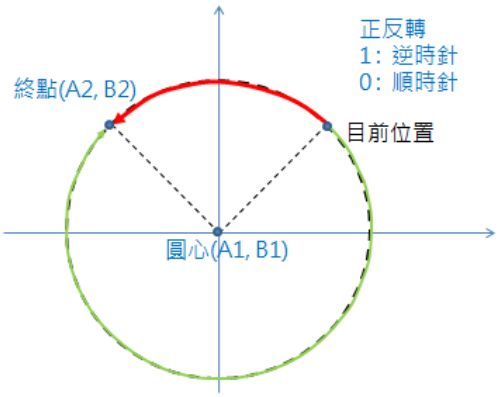
➢ 备注

- 圆弧运动是一次下达三轴的动作命令, 因此只能下达给 ASDA_M 驱动器执行三轴的补间运动.
- 三轴驱动器中当有两轴在执行圆弧运动时, 另一轴无法再执行其它的运动命令.

5.12、 圆弧:圆心终点

➢ 范例说明

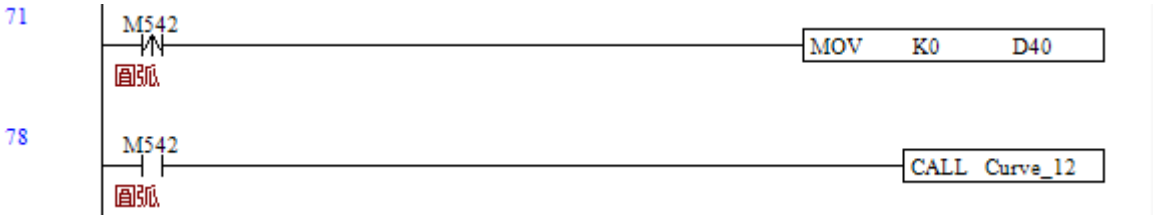
- 以 M542 为触发与执行圆弧运动的致能条件. 启动 M542 后开始执行圆弧运动的参数与命令下达动作.
- 圆弧: 圆心终点运动需下达【圆心坐标 1】(A1), 【圆心坐标 2】(B1), 【终点坐标 1】(A2), 【终点坐标 2】(B2)与【正反转】等五个参数. 若数据参数起始地址为 D1000, 则 D1000 代表【圆心坐标 1】(PUU), D1002 代表【圆心坐标 2】(PUU), D1004 代表【终点坐标 1】 (PUU), D1006 代表【终点坐标 2】 (PUU)与 D1008 代表【正反转】. 根据参数其运动路径如下图所示:



➢ 范例程序

- 人机画面

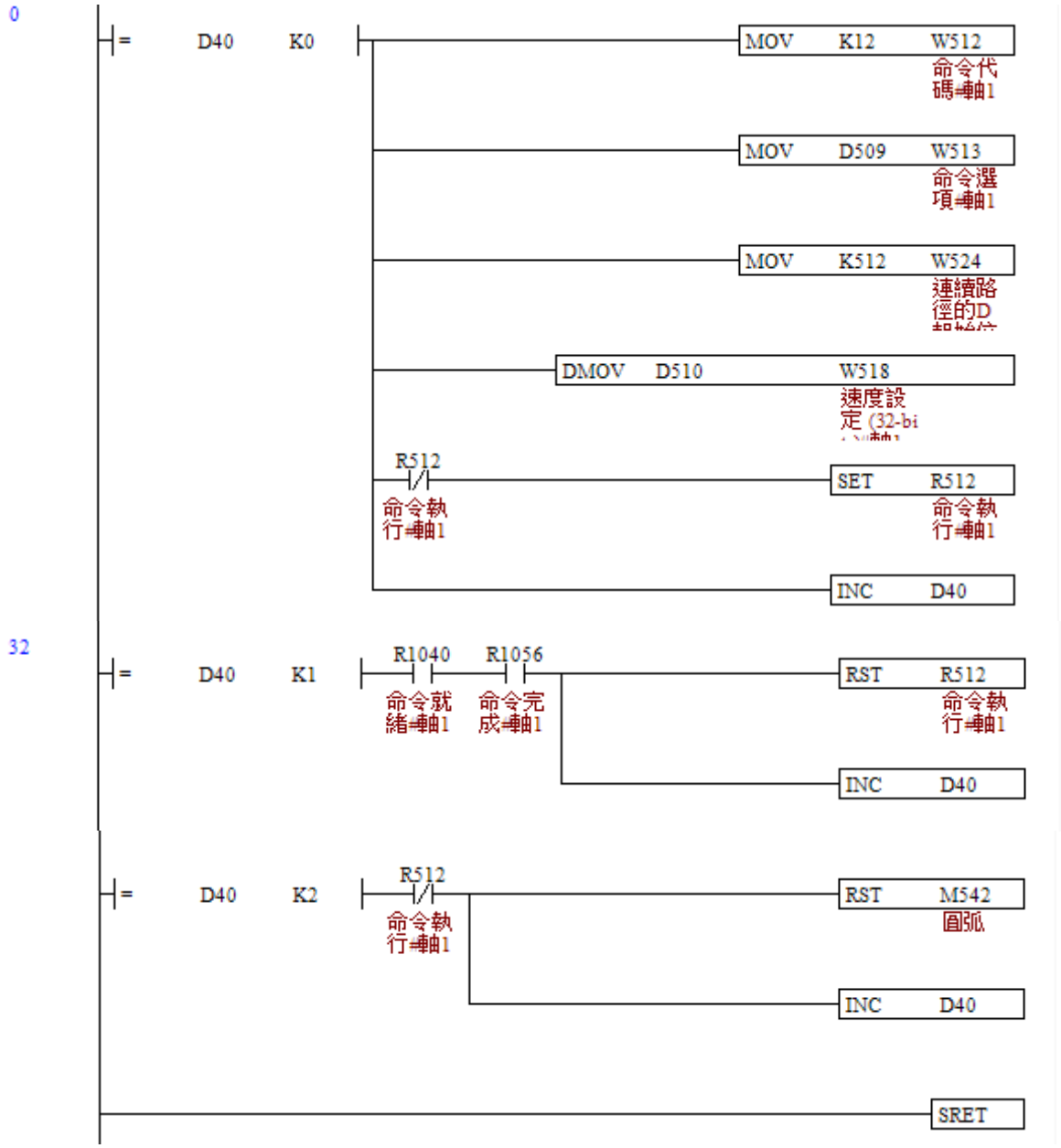
- 主程序



a. 当 M542 由 Off 状态变为 On 时, 初始控制状态(D40=0)

b. 当 M542 为 On 时, 进入子程序 Curve_12 执行圆弧动作. 动作完成后 M542 清除.

- CURVE_12 子程序



a. 在状态 0 时(D40=0), 开始下达动作参数, 将【命令代码】(W512)设为 12 代表进行圆弧: 圆心&终点模式运动, D509 写至【命令选项】(W513)代表执行圆弧轴选项, 其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补. 【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 后则将运动参数: 【圆心坐标 1】(D512), 【圆心坐标 2】(D514), 【终点坐标 1】(D516), 【终点坐标 2】(D518)与【正反转】(D520)写入驱动器中并进入状态 1.

b. 在状态 1 时(D40=1), 【命令就绪】(R1040)为 On 此时圆弧运动已进行中. 等待【命令完成】(R1056)为 On 代表运动完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D40=2), 运动已结束. 此时确认【命令执行】(R512)已清除为 Off 后, 清除执行圆弧旗标 M542 为 Off, 控制流程结束.

➢ 备注

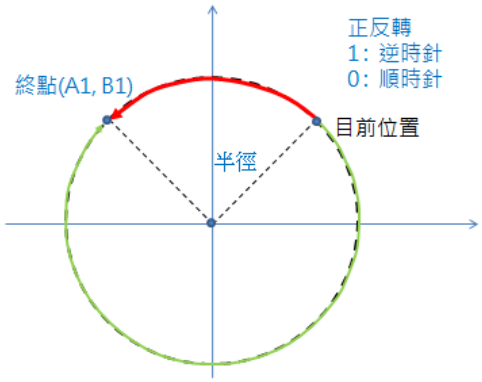
- 圆弧运动是一次下达三轴的动作命令, 因此只能下达给 ASDA_M 驱动器执行三轴的补间运动.
- 三轴驱动器中当有两轴在执行圆弧运动时, 另一轴无法再执行其它的运动命令.

- 参数中正反转若设为 1，代表逆时针方向；相反地，设为 0，代表顺时针方向

5.13、 圆弧:终点半径

➢ 范例说明

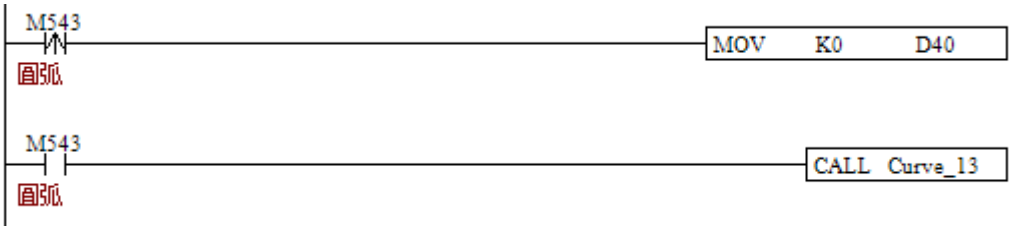
- 以 M543 为触发与执行圆弧运动的致能条件. 启动 M543 后开始执行圆弧运动的参数与命令下达动作.
- 圆弧：终点半径运动需下达【终点坐标 1】(A1)，【终点坐标 2】(B1)，【半径】与【正反转】等四个参数. 若数据参数起始地址为 D1000，则 D1000 代表【终点坐标 1】(PUU), D1002 代表【终点坐标 2】(PUU), D1004 代表【半径】(PUU), D1006 代表【正反转】. 根据参数其运动路径如下图所示:



➢ 范例程序

- 人机画面

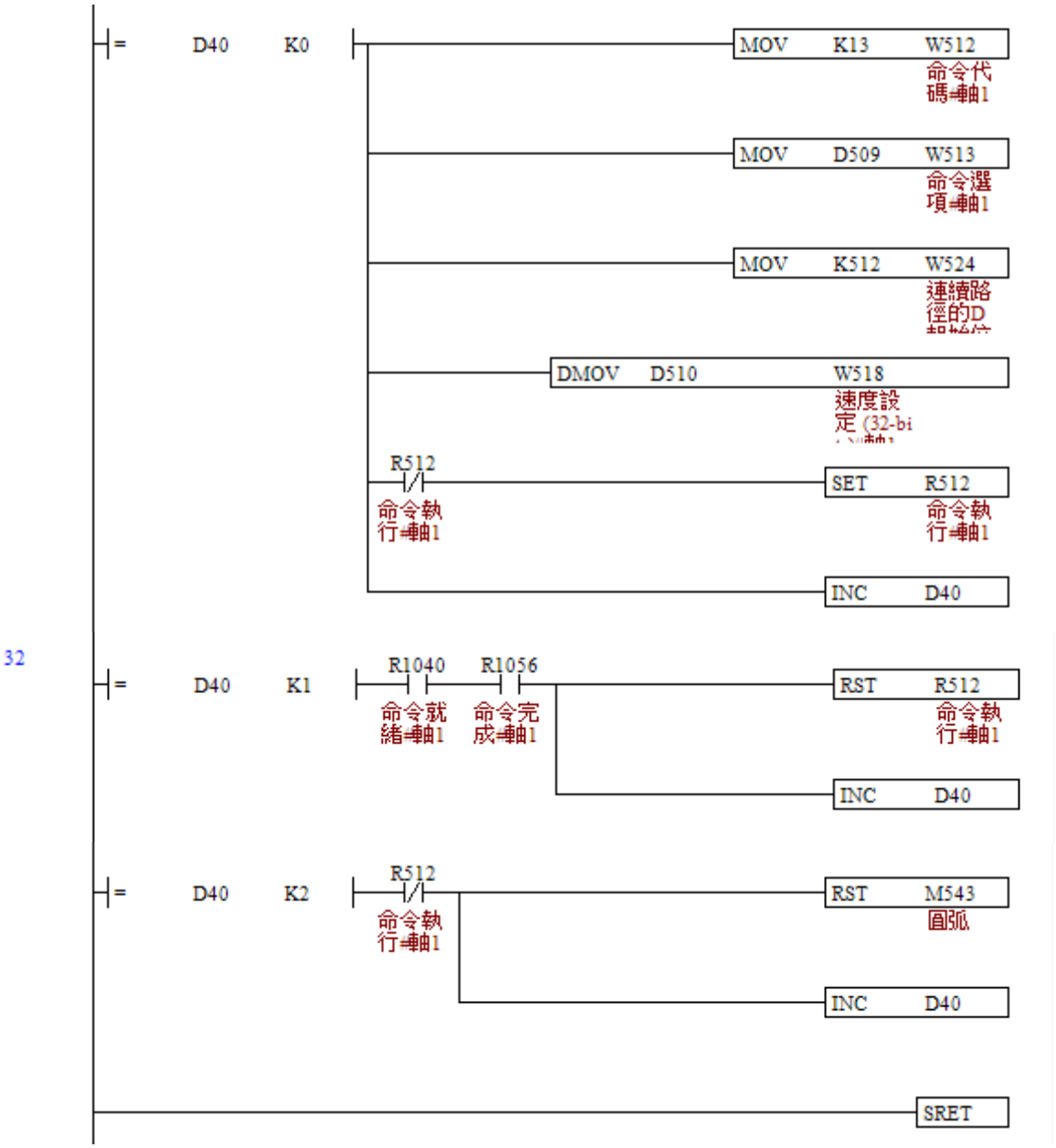
- 主程序



a. 当 M543 由 Off 状态变为 On 时，初始控制状态(D40=0)

b. 当 M543 为 On 时，进入子程序 Curve_13 执行圆弧动作. 动作完成后 M543 清除.

- CURVE_13 子程序



a. 在状态 0 时(D40=0), 开始下达动作参数, 将【命令代码】(W512)设为 13 代表进行圆弧: 终点&半径模式运动, D509 写至【命令选项】(W513)代表执行圆弧轴选项, 其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补. 【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 后开始将运动参数圆弧数据:【终点坐标 1】(D512),【终点坐标 2】(D514),【半径】(D516)与【正反转】(D518)写入驱动器中并进入状态 1.

b. 在状态 1 时(D40=1),【命令就绪】(R1040)为 On 代表圆弧运动进行中. 等待【命令完成】(R1056)为 On 代表运动已完成. 接着清除【命令执行】(R512)并进入状态 2.

c. 在状态 2 时(D40=2), 运动已结束. 此时确认【命令执行】(R512)已清除为 Off 后, 清除执行圆弧旗标 M543 为 Off, 控制流程结束.

➢ 备注

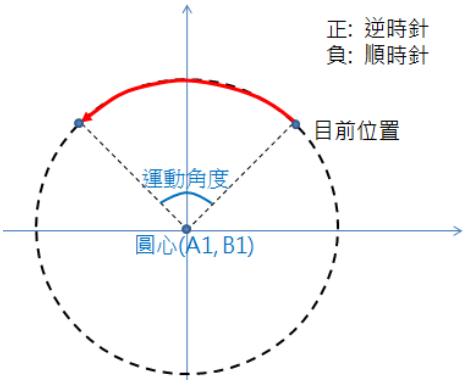
■ 圆弧运动是一次下达三轴的动作命令, 因此只能下达给 ASDA_M 驱动器执行三轴的补间运动.

- 三轴驱动器中当有两轴在执行圆弧运动时，另一轴无法再执行其它的运动命令.
- 参数中正反转若设为 1，代表逆时针方向；相反地，设为 0，代表顺时针方向

5.14、 圆弧:圆心角度

➢ 范例说明

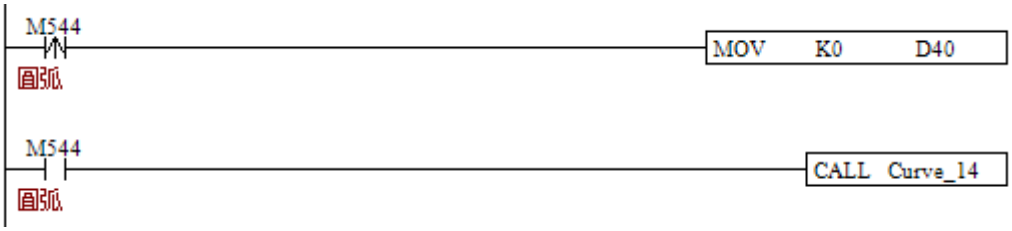
- 以 M544 为触发与执行圆弧运动的致能条件. 启动 M544 后开始执行圆弧运动的参数与命令下达动作.
- 圆弧：圆心角度运动需下达【圆心坐标 1】(A1)，【圆心坐标 2】(B1)与【运动角度】等三个参数. 若数据参数起始地址为 D1000，则 D1000 代表【圆心坐标 1】(PUU), D1002 代表【圆心坐标 2】(PUU), D1004 代表【运动角度】设定. 需特别注意角度单位为 0.5 度，也就是值为 180 代表 90 度角. 根据参数其运动路径如下图所示:



➢ 范例程序

- 人机画面

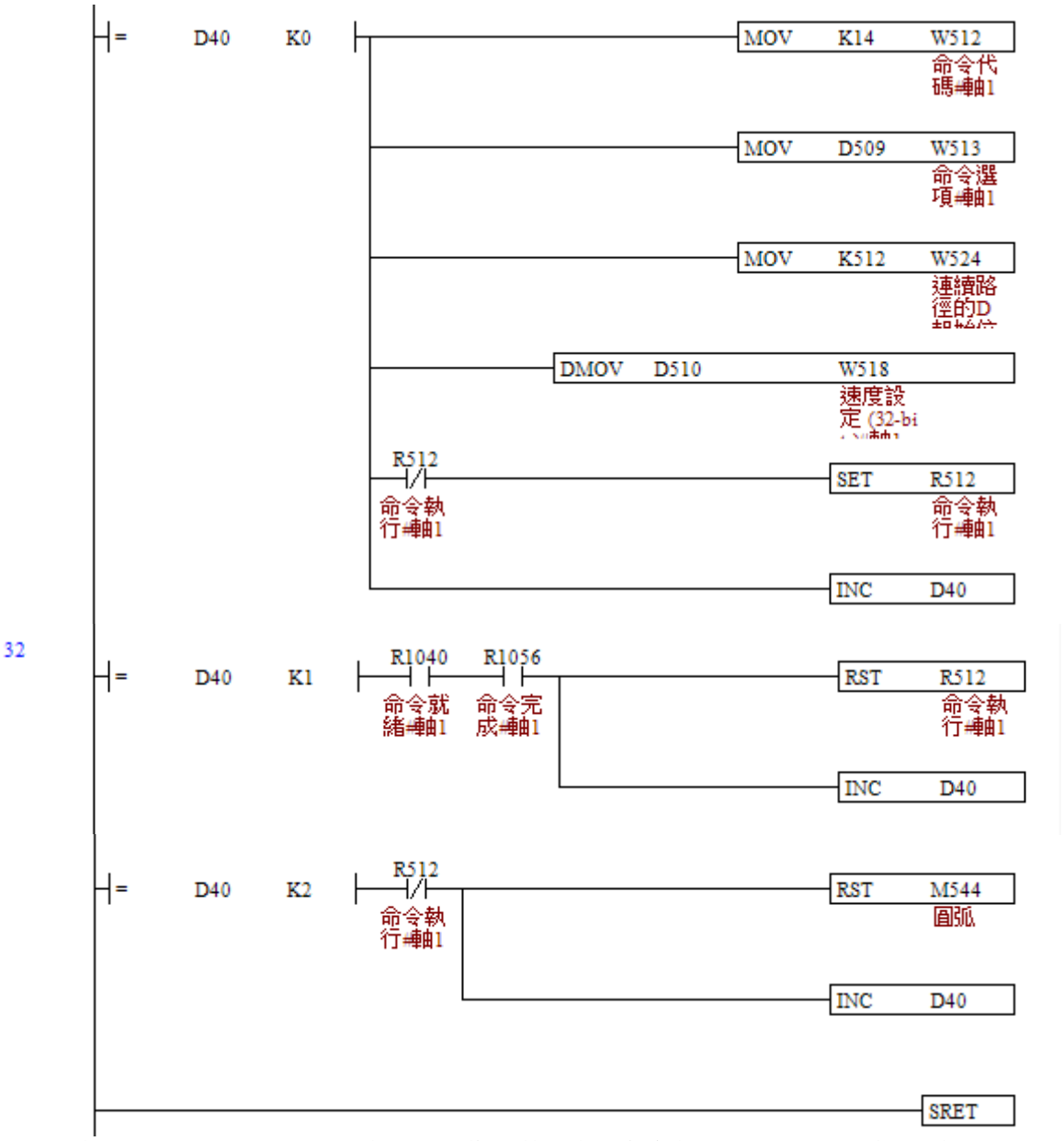
- 主程序



a. 当 M544 由 Off 状态变为 On 时，初始控制状态(D40=0)

b. 当 M544 为 On 时，进入子程序 Curve_14 执行圆弧动作. 动作完成后 M544 清除.

- CURVE_14 子程序



a. 在状态 0 时(D40=0), 开始下达动作参数, 将【命令代码】(W512)设为 14 代表进行圆弧: 圆心&角度模式运动, D509 写至【命令选项】(W513)代表执行圆弧轴选项, 其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补. 【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 后开始将运动参数圆弧数据:【圆心坐标 1】(D512),【圆心坐标 2】(D514)与【运动角度】(D516)写入驱动器中并进入状态 1.

b. 在状态 1 时(D40=1),【命令就绪】(R1040)为 On 代表圆弧运动进行中. 等待【命令完成】(R1056)为 On 代表运动已完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D40=2), 运动已结束. 此时确认【命令执行】(R512)已清除为 Off 后, 清除执行圆弧旗标 M544 为 Off, 控制流程结束.

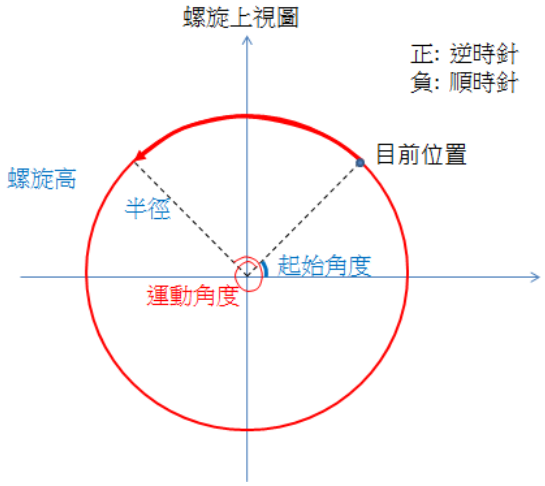
- 备注
  - 圆弧运动是一次下达三轴的动作命令, 因此只能下达给 ASDA_M 驱动器执行三轴的补间运动.
  - 三轴驱动器中当有两轴在执行圆弧运动时, 另一轴无法再执行其它的运动命令.

- 参数中角度若设为正值，代表逆时针方向；相反地，角度设为负值，代表顺时针方向

5.15、螺旋

➢ 范例说明

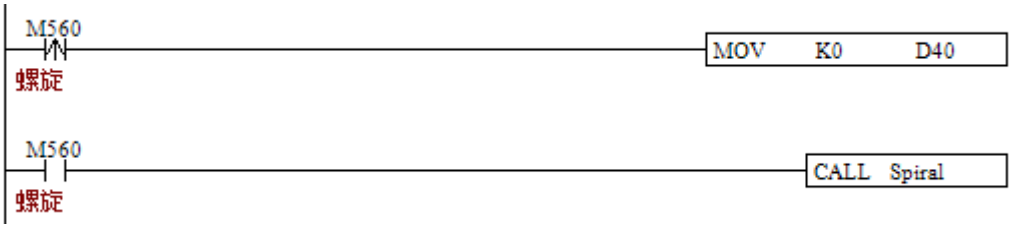
- 以 M560 为触发与执行螺旋运动的致能条件. 启动 M560 后开始执行螺旋运动的参数与命令下达动作.
- 螺旋运动为原本的圆弧运动加上另一轴的高度移动补间，需要下达【半径】，【初始角度】，【运动角度】与【运动高度】四个参数. 若数据参数起始地址为 D1000，则 D1000 代表【半径】(PUU), D1002 代表【初始角度】, D1004 代表【运动角度】, D1006 代表【运动高度】(PUU). 需特别注意角度单位为 0.5 度，也就是 180 代表 90 度角. 根据参数其运动路径上视图如下所示:



➢ 范例程序

- 人机画面

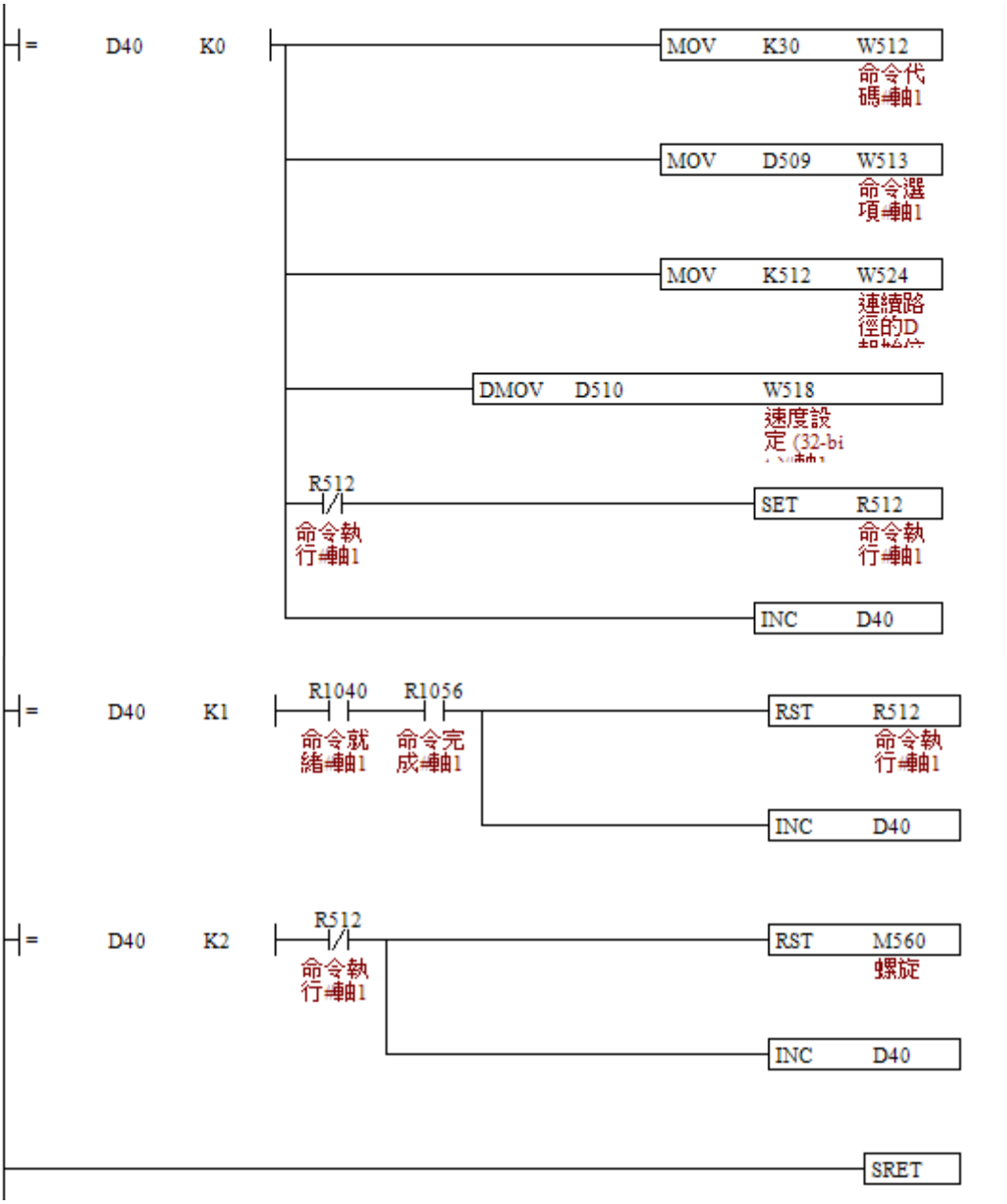
- 主程序



a. 当 M560 由 Off 状态变为 On 时，初始控制状态(D40=0)

b. 当 M560 为 On 时，进入子程序 Spiral 执行螺旋动作．螺旋动作完成后 M560 清除．

■ SPIRAL 子程序



a. 在状态 0 时(D40=0)，开始下达动作参数，将【命令代码】(W512)设为 30 代表进行螺旋运动, D509 写至【命令选项】(W513)代表执行螺旋的圆弧轴选项，其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补．【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至【速度设定】(W518)．最后触发【命令执行】(R512)为 On 后开始将运动参数与螺旋数据：【半径】(D512)，【初始角度】(D514)，【运动角度】(D516)与【运动高度】(D518)写入驱动器中并进入状态 1.

b. 在状态 1 时(D40=1)，【命令就绪】(R1040)为 On 代表螺旋运动进行．等待【命令完成】(R1056)为 On 代表运动已完成．接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D40=2)，运动已结束．此时确认【命令执行】(R512)已清除为 Off 后，清除执行螺旋旗标 M560 为 Off，控制流程结束．

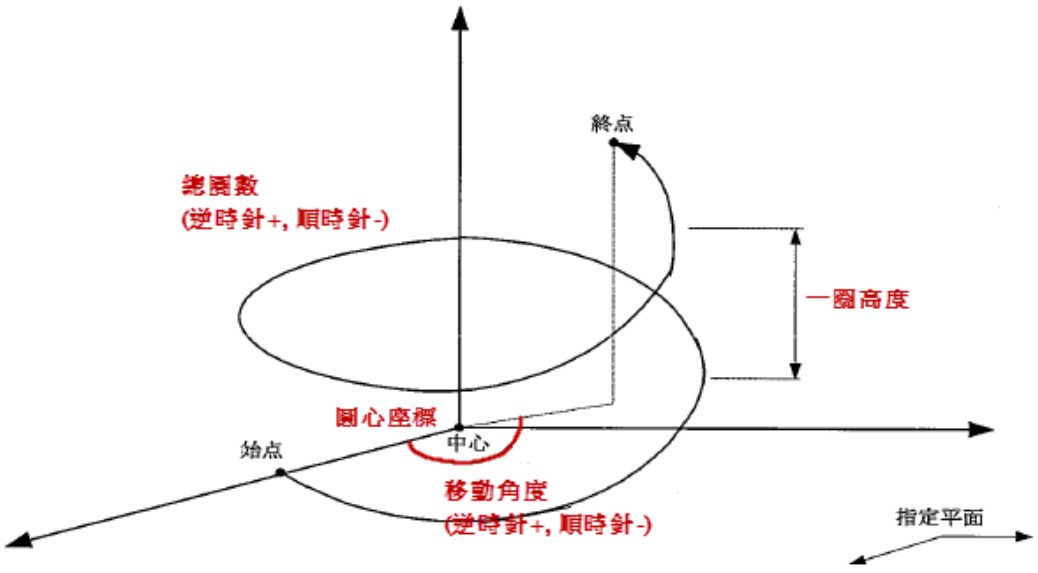
➢ 备注

- 螺旋运动是一次下达三轴的动作命令，因此只能下达给 ASDA_M 驱动器执行三轴补间运动.
- 螺旋参数中角度设为正值，代表逆时针方向；相反地，角度设为负值，代表顺时针方向

5.16、 螺旋W

➢ 范例说明

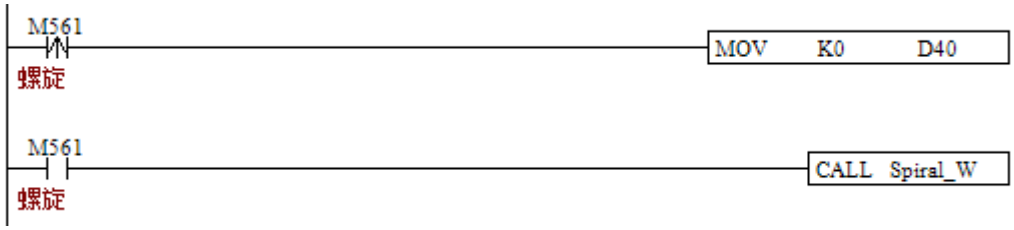
- 以 M561 为触发与执行螺旋 W 运动的致能条件. 启动 M561 后将会开始执行螺旋 W 运动的参数与命令下达动作.
- 螺旋 W 运动为参考现在位置与设定的圆心坐标得到螺旋半径后，再经由设定的每圈移动距离，总圈数与最终偏移角度等信息完成整个螺旋运动的参数设定. 因此螺旋 W 需要下达【圆心坐标 1】，【圆心坐标 2】，【螺距】，【总圈数】与【偏移角度】等五个参数. 若数据参数起始地址为 D1000，则 D1000 代表【圆心坐标 1】，D1002 代表【圆心坐标 2】，D1004 代表【螺距】，D1006 代表【总圈数】，D1008 代表【偏移角度】. 需特别注意角度单位为 0.5 度，也就是 180 代表 90 度角. 根据参数其运动路径如下图所示:



➢ 范例程序

- 人机画面

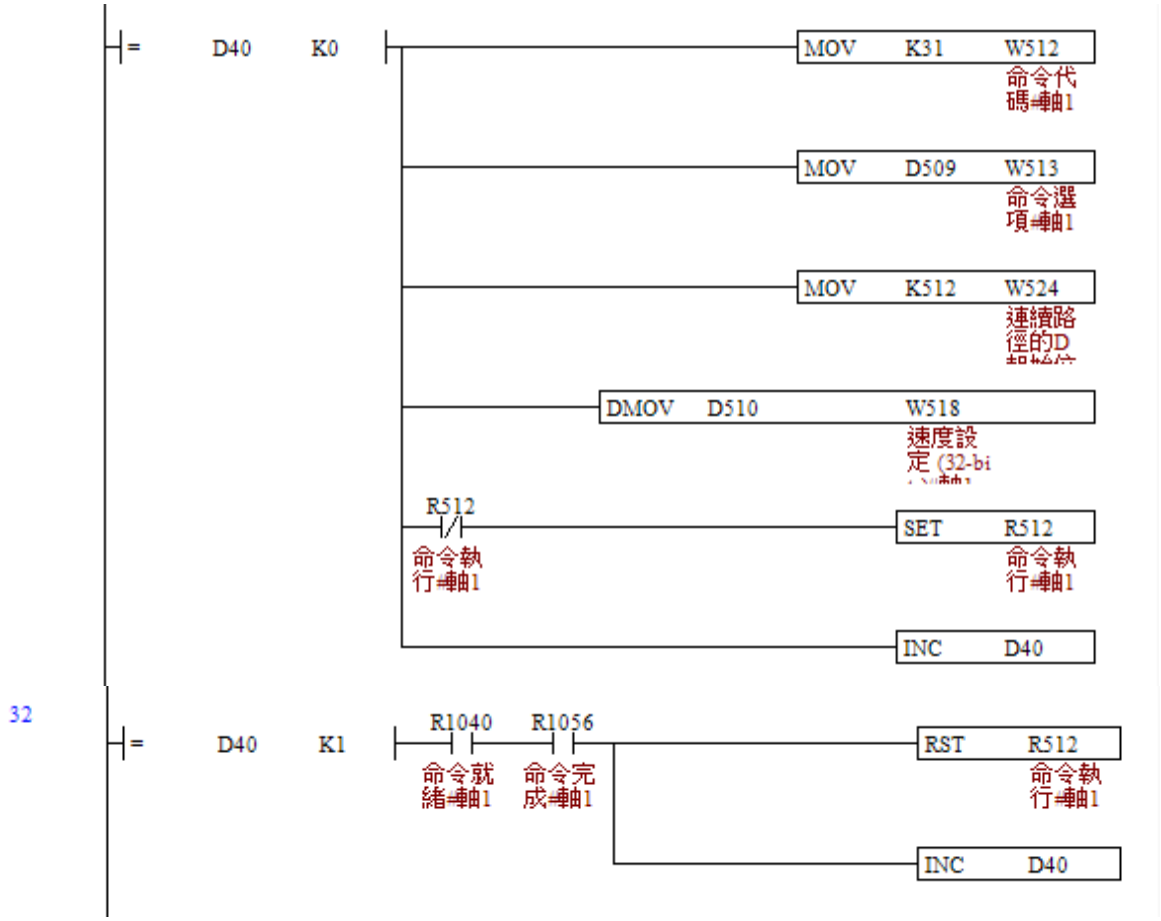
■ 主程序

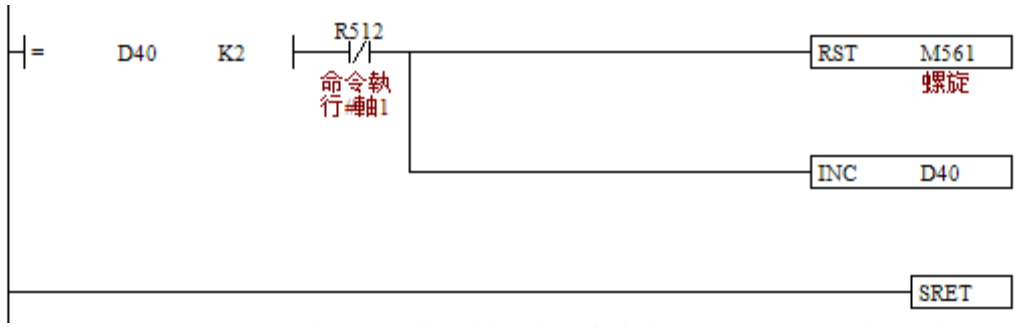


a. 当 M561 由 Off 状态变为 On 时，初始控制状态(D40=0).

b. 当 M561 为 On 时，进入子程序 Spiral_W 执行螺旋 W 动作．螺旋 W 动作完成后 M561 清除.

■ SCREW_W 子程序





a. 在状态 0 时(D40=0), 开始下达动作参数, 将【命令代码】(W512)设为 31 代表进行螺旋 W 运动, D509 写至【命令选项】(W513)代表执行螺旋 W 的圆弧轴选项, 其中 0 代表 XY 两轴圆弧差补, 1 代表 YZ 两轴圆弧差补, 2 代表 XZ 两轴圆弧差补. 【参数起始地址】(W524)设为 512 代表由 D512 开始读取路径数据参数, D510 写至【速度设定】(W518). 最后触发【命令执行】(R512)为 On 后开始将运动参数与螺旋 W 数据: 【圆心坐标 1】(D512), 【圆心坐标 2】(D514), 【螺距】(D516), 【总圈数】(D518)与【偏移角度】(D520)写入驱动器中并进入状态 1.

b. 在状态 1 时(D40=1), 【命令就绪】(R1040)为 On 代表螺旋 W 运动进行. 等待【命令完成】(R1056)为 On 代表运动已完成. 接着清除【命令执行】(R512)并进入状态 2

c. 在状态 2 时(D40=2), 运动已结束. 此时确认【命令执行】(R512)已清除为 Off 后, 清除执行螺旋 W 旗标 M561 为 Off, 控制流程结束.

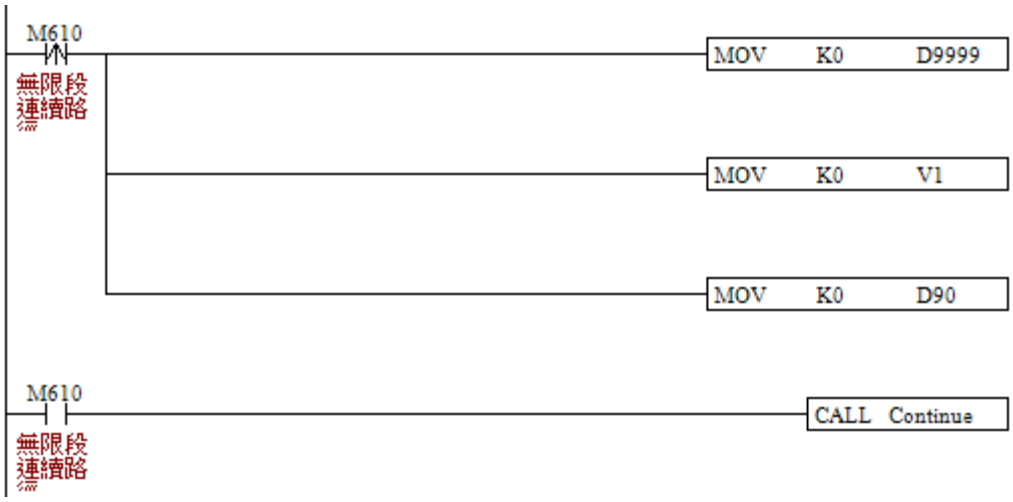
- 备注
  - 螺旋 W 运动是一次下达三轴的动作命令, 因此只能下达给 ASDA_M 驱动器执行三轴补间运动.
  - 螺旋 W 参数中总圈数与偏移角度若设定为正值, 代表逆时针方向; 相反地, 设定为负值, 代表顺时针方向

5.17、 连续路径

- 范例说明
  - 以 M610 为触发与执行连续运动的致能条件.
  - 启动 M610 后根据【连续路径数】(D2000)以设定连续运动路径个数(例子为最多 4 个路径)并将画面中 PATH#1 到 PATH#4 的路径数据加载到伺服中. 例如若【连续路径数】设为 3, 则只执行 PATH#1 到 PATH#3 的三个连续动作数据填入并启动连续运动.
- 范例程序
  - 人机画面

画面最多可输入四笔路径数据. 启动连续运动时, 根据【连续路径数】来加载设定数量的路径数据给驱动器执行. 以下为人机的设定输入画面.

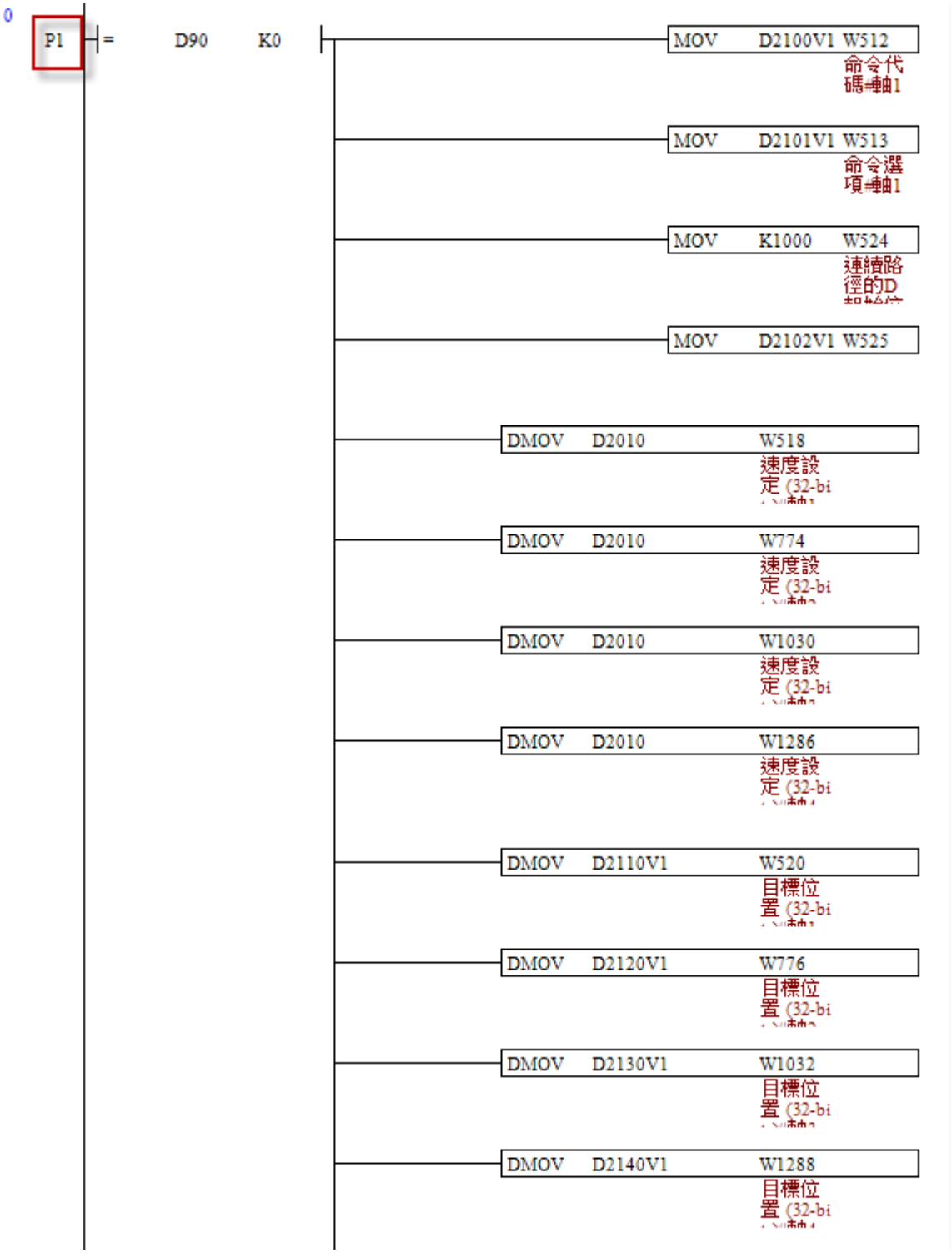
■ 主程序

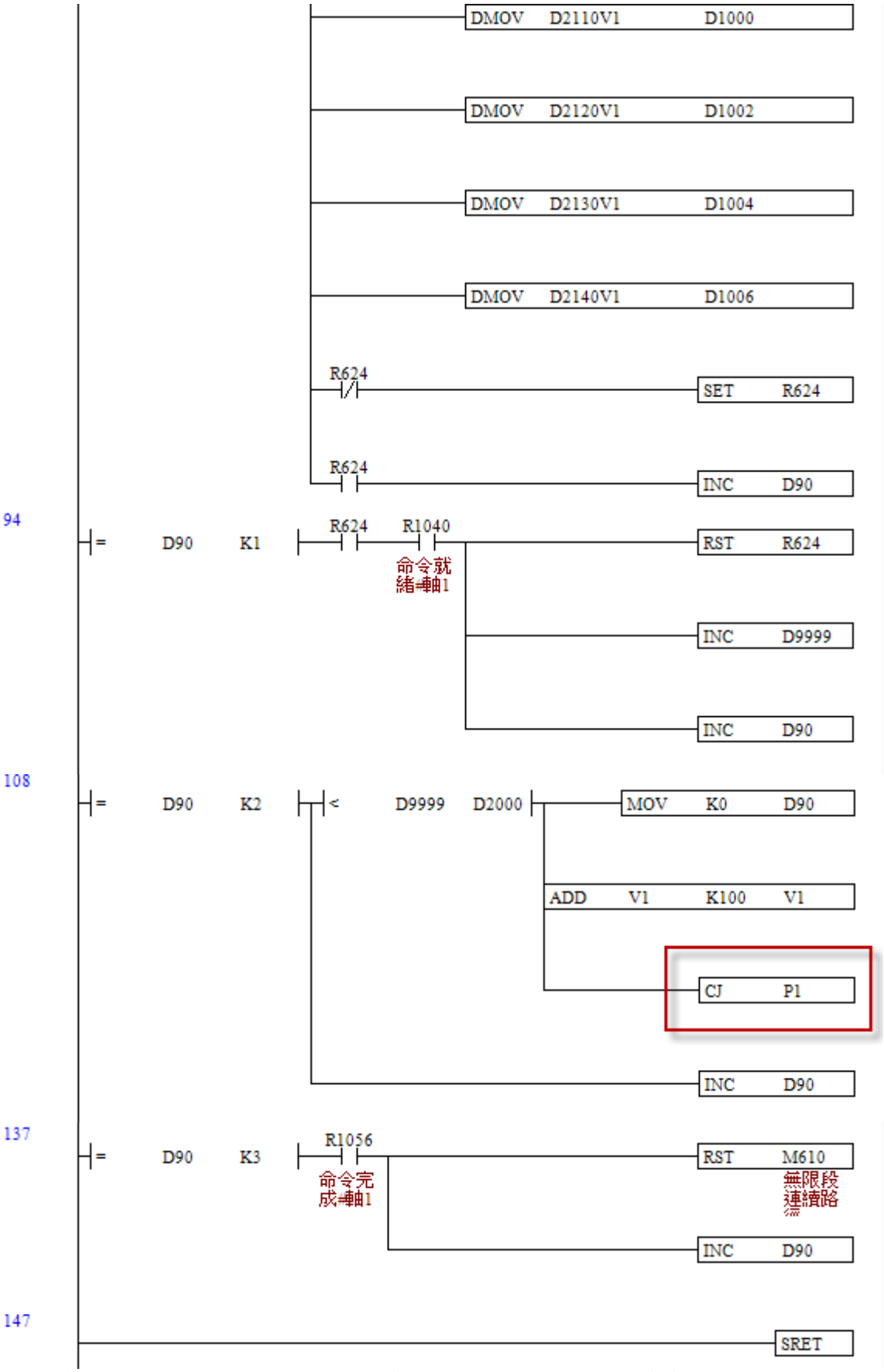


a. 当 M610 由 Off 状态变为 On 时, 初始控制状态 D90, 命令参数偏移值 V1 与已成功下达连续路径数 D9999 的初始动作.

b. 当 M610 为 On 时呼叫子程序 Continue, 执行连续路径命令下达与动作. 连续路径动作完成后 M610 清除.

■ CONTINUE 子程序


0
P1
=
D90
K0
MOV D2100V1 W512
命令代碼軸1
MOV D2101V1 W513
命令選項軸1
MOV K1000 W524
連續路徑的D
MOV D2102V1 W525
DMOV D2010 W518
速度設定 (32-bi
DMOV D2010 W774
速度設定 (32-bi
DMOV D2010 W1030
速度設定 (32-bi
DMOV D2010 W1286
速度設定 (32-bi
DMOV D2110V1 W520
目標位置 (32-bi
DMOV D2120V1 W776
目標位置 (32-bi
DMOV D2130V1 W1032
目標位置 (32-bi
DMOV D2140V1 W1288
目標位置 (32-bi



a. 在状态 0 时(D90=0)，开始下达动作参数，将 D2100 写至【命令代码】(W512), D2101 写至【命令选项】(W513),【参数起始地址】(W524)设为 1000, D2102 写至【Overlap】(W525), D2010 写至各轴【速度设定】(W518, …), D2110, D2120, D2130 与 D2140 写至各轴【目标

位置】(W520, …). 如果命令运动非直线，而是圆弧或螺旋等运动时，则需要由数据表读取参数的，因此也将 D2110, D2120, D2130 与 D2140 等写至参考数据区 D1000 开始的地址. 最后触发【命令预载】(R624)为 On 并确认后进入状态 1.

b. 在状态 1 时(D90=1)，触发【命令预载】(R624)须等待【命令就绪】(R1040)变为 On 才代表此一动作参数成功加载至驱动器中. 并在写入命令成功后将【已下达命令数】(D9999)加 1，并进入状态 2

c. 在状态 2 时(D90=2)，先判断下达命令数是否已到达设定(D2000 为下达命令数设定). 若未到达，需再执行加载下一个命令至驱动器. 首先，增加命令参数偏移值 V1(增加 100 为参考 Path#2，增加 200 为参考 Path#3，以此类推)，再回到状态 0 后跳跃至下达命令参数程序区段(CJ P1)；若下达命令数已到达，不需再加载其它命令，则进入状态 3 (D90=3).

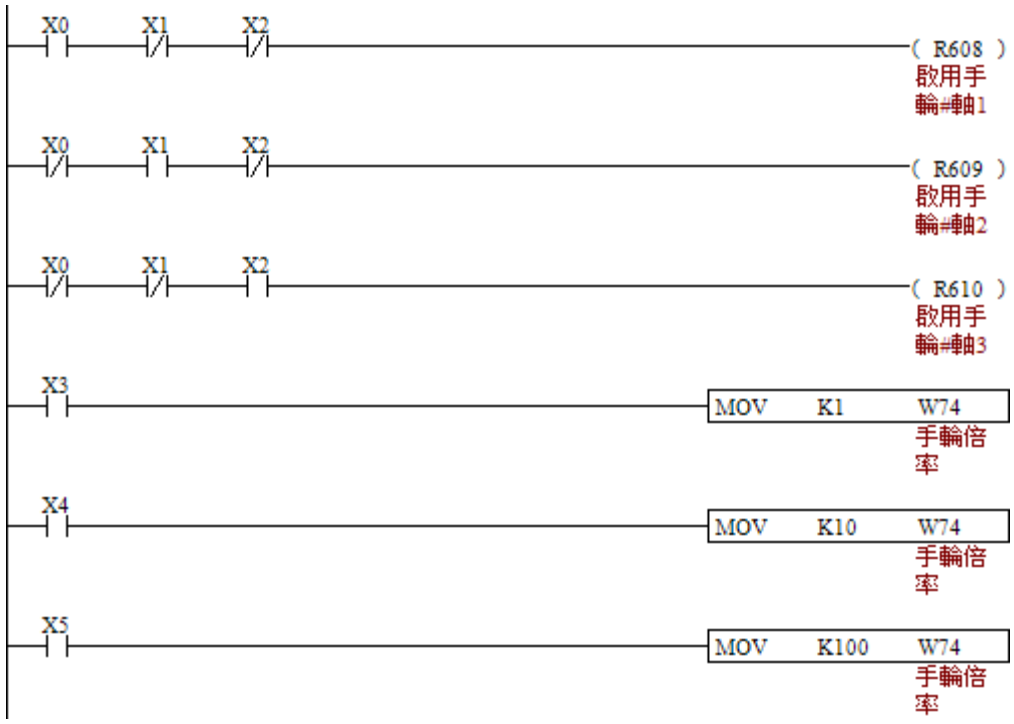
d. 第二次以后(*N* 次)进入到状态 0 时，下达动作动作参数，将 D(2100+100x(*N*-1))写至【命令代码】(W512), D(2101+100x(*N*-1))写至【命令选项】(W513), D(2102+100x(*N*-1))写至【Overlap】(W525), D2010+100x(*N*-1))写至各轴【速度设定】(W518, …), D(2110+100x(*N*-1))地址开始写至各轴【目标位置】(W520, …)，也将 D(2110+100x(***N***-1))开始之地址写入至路径数据区 D1000 开始之连续地址中. 写参数完成后，触发【命令预载】(R624)为 On 并确认完成后进入状态 1，重复步骤 b~d.

e. 在状态 3 时(D90=3)，等待连续动作的完成旗标，即【命令完成】(R1056)成为 On 代表所有连续运动已完成，才进入结束状态 4 并清除启动连续运动旗标 M610.

- 备注
  - 连续路径可下达无限制段的运动参数数据给驱动器，但一次最多可同时容纳 8 笔数据在驱动器内部，只要执行完成一个运动后即可紧接着再使用【命令预载】触发下一个运动命令下达.
  - 在连续路径运动中，需下达两笔路径数据后才会开始运动.
  - 在连续路径运动中，当执行至剩下最后一个路径运动时，此时已来不及再下达新命令了.
  - 使用【命令预载】下达命令时，可透过【命令就绪】来得知命令下达是否成功，并透过【路径剩余数】来得知当前驱动器内剩余未执行完成的路径数.

5.18、 手轮

- 范例说明
  - 将手轮装置接于 HMC08 上，并透过 I/O 来达到切换倍率与手轮控制轴的选择. 将手轮装置启动轴 1 输出接脚接于 X0，启动轴 2 输出接脚接于 X1，启动轴 3 输出接脚接于 X2, 1 倍率输出接脚接于 X3, 10 倍率输出接脚接于 X4, 100 倍率输出接脚接于 X5.
- 范例程序
  - 主程序



a. 当 X0 为 On, X1 为 Off, X2 为 Off 时，输出 R608 为 On, R609 为 Off, R610 为 Off. 即启动轴 1【启用手轮】为 On，此时启用轴 1 手轮功能运动.

b. 当 X1 为 On, X0 为 Off, X2 为 Off 时，输出 R608 为 Off, R609 为 On, R610 为 Off. 即启动轴 2【启用手轮】为 On，此时启用轴 2 手轮功能运动.

c. 当 X2 为 On, X0 为 Off, X1 为 Off 时，输出 R608 为 Off, R609 为 Off, R610 为 On. 即启动轴 3【启用手轮】为 On，此时启用轴 3 手轮功能运动.

d. 当 X3 为 On 时，将【手轮倍率】(W74)设为 1，手轮即以 1 倍速度运动.

e. 当 X4 为 On 时，将【手轮倍率】(W74)设为 10，手轮即以 10 倍速度运动.

f. 当 X5 为 On 时，将【手轮倍率】(W74)设为 100，手轮即以 100 倍速度运动.

➢ 备注

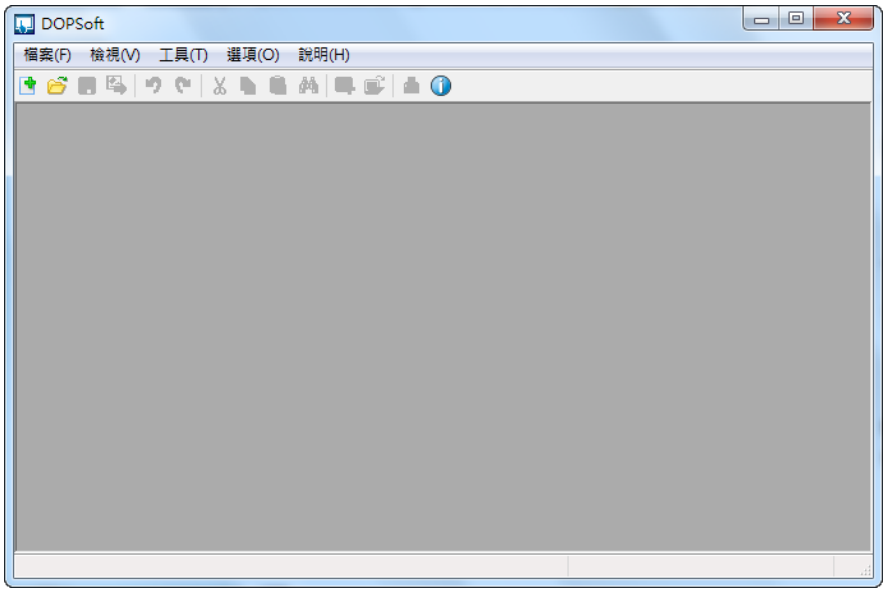
■ 同一时间内仅能启动一轴的手轮功能.

6. LADDER EDITOR软件

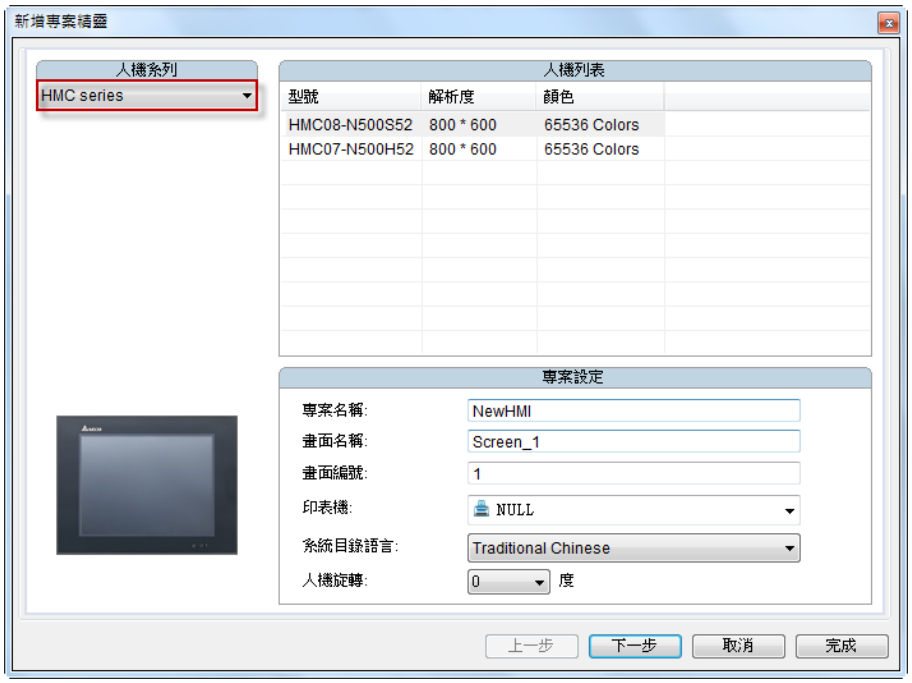
本章节主要是介绍如何使用 Ladder Editor 软件, Ladder Editor 已整合于 DOPSoft 中，安装 DOPSoft 软件方式与人机画面编辑各功能操作请参考- 【DOPSoft 软件使用手册】.

6.1、 LADDER EDITOR软件

- 开启DOPSOFT软件

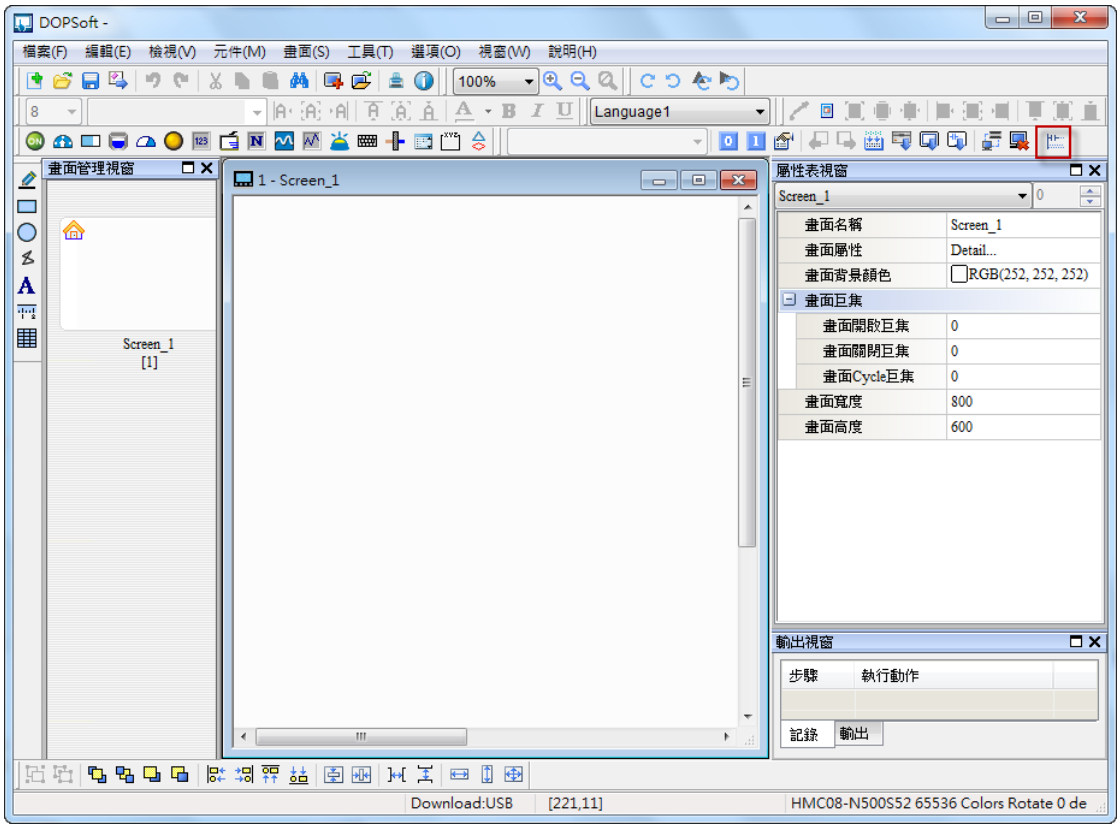


- 选择HMC机种:

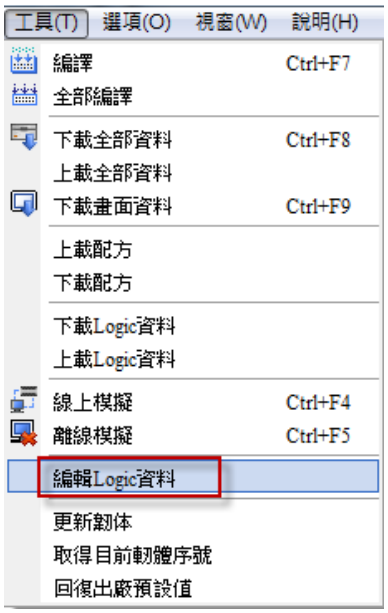


- 开启LADDER EDITOR

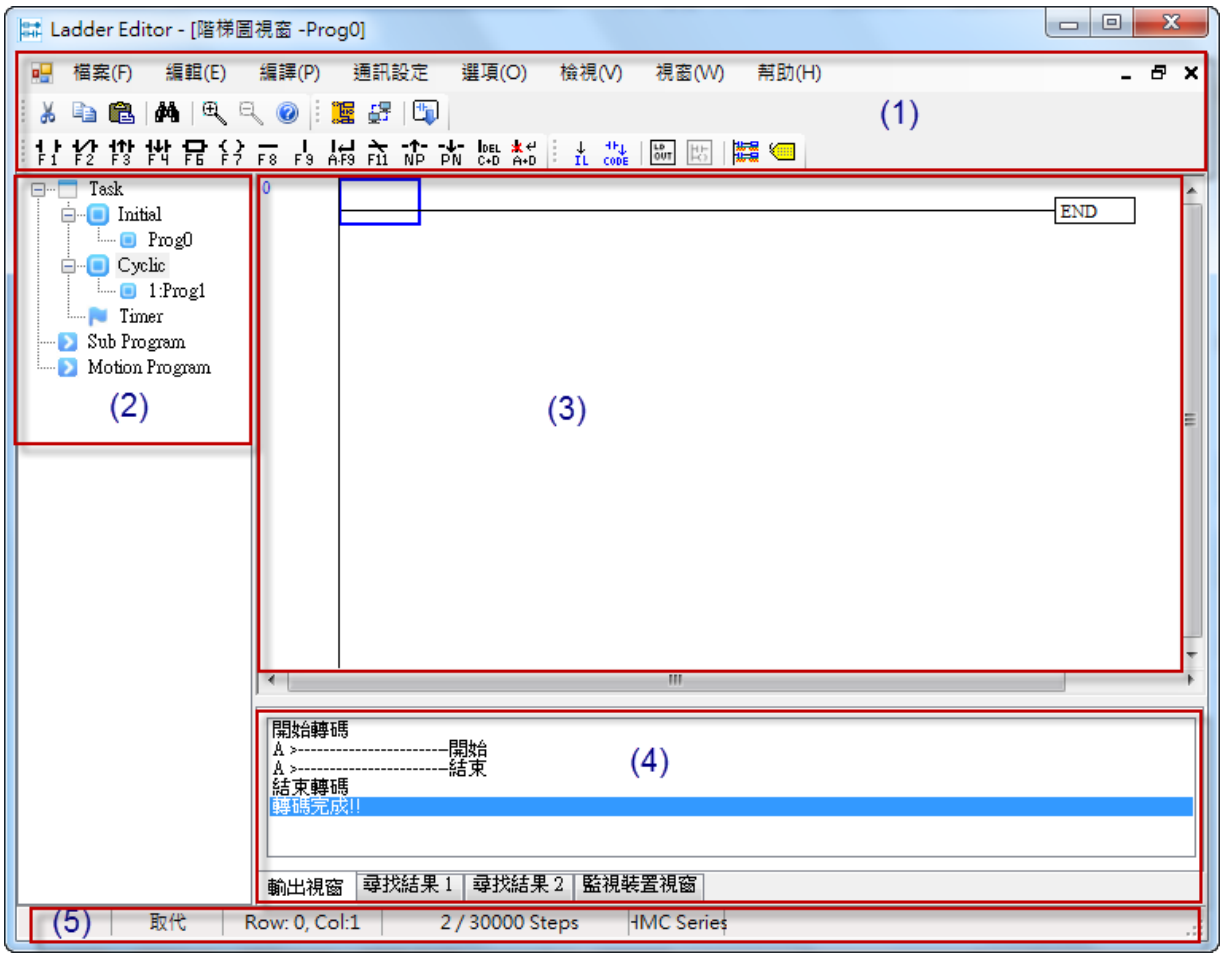
在工具栏中点下【编辑 Logic 数据】图标启动 Ladder Editor.



或在【工具】➔【编辑 Logic 数据】以启动 Ladder Editor.



➢ 开启LADDER EDITOR完成

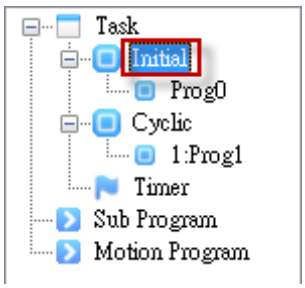


标记	注意项目	叙述内容
(1)	工具栏	档案，编辑，编译，通信设置等功能
(2)	程序树形图	目前项目内使用 Ladder 程序架构
(3)	程序编辑区	当前选择程序内容编辑
(4)	应用窗口区	包含输出窗口，寻找结果，监视装置窗口
(5)	编辑状态区	当前编辑状态，可切换【取代】或【插入】模式

6. 2、 LADDER程序新增与设定

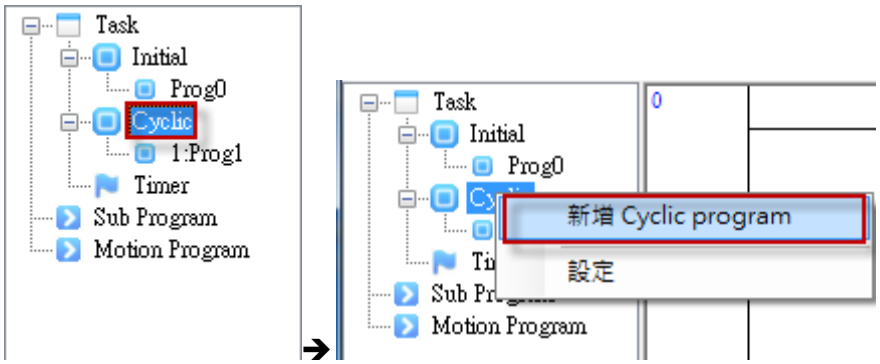
➢ 初始程序(INITIAL TASK)

Initial 程序只有唯一且无法新增，同时也无法改变其程序名称．可将系统的初始设定动作写在此初始程序中．

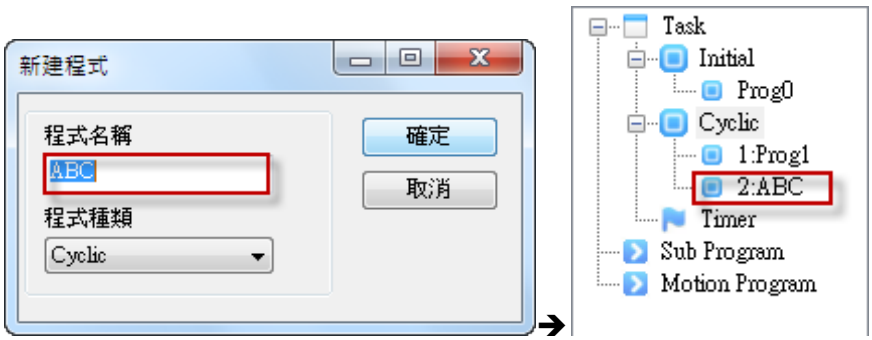


- 主程序(CYCLIC TASK)
  - 新增CYCLIC TASK

在 Cyclic 中按鼠标右键，接着点选【新增 Cyclic program】会出现【新建程序】窗口.

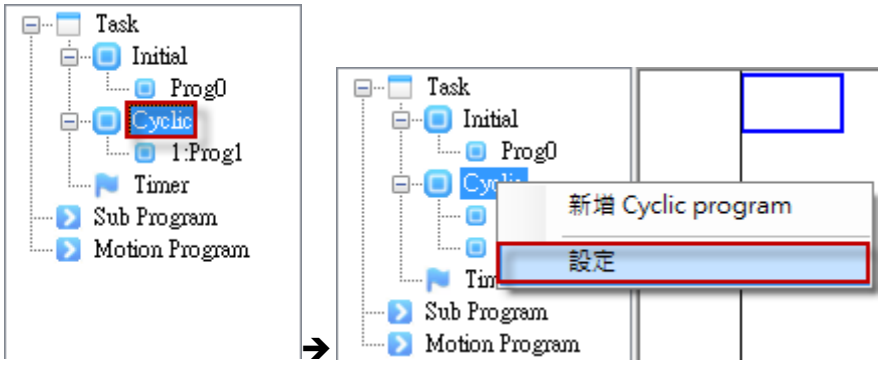


输入程序名称，最多不可超过 16 个字符限制，输入完成后按下【确定】即完成.



  - 设定CYCLIC TASK

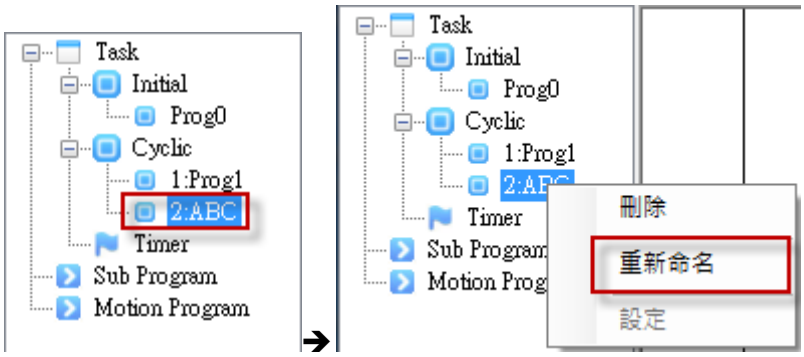
在 Cyclic 中按鼠标右键，接着点选【新增 Cyclic program】会出现【新建程序】窗口.



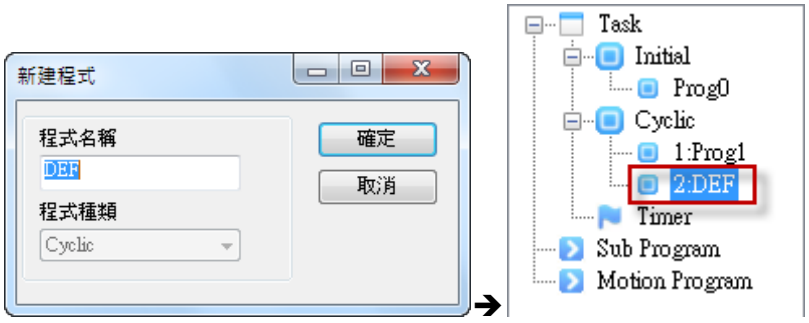
根据实际存在的 Task 显示于窗口上，输入每一个 Task 的使用比例%，所有 Task 使用比例总和需为 100，否则将出现【全部 Task 的使用比例总和必需等于 100】警告讯息. 也可直接使用按钮【平均分配到全部】与【平均分配到未指定】做快速设定.

■ 变更程序名称

在程序名称中按鼠标右键，接着点选【重新命名】会出现【新建程序】窗口.



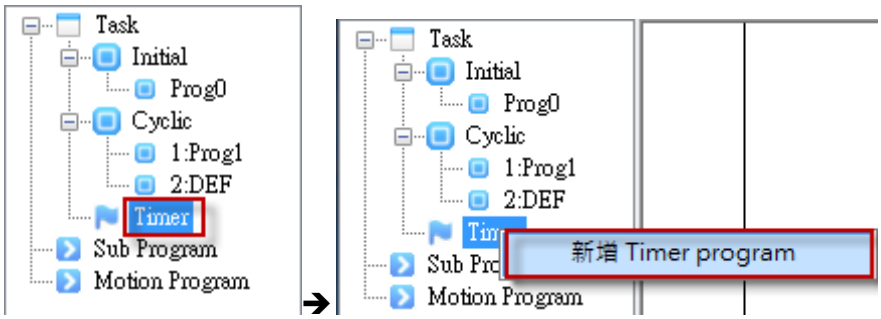
输入变更程序名称，输入完成后按下【确定】即变更完成.



➢ 定时程序(TIMER TASK)

■ 新增TIMER TASK

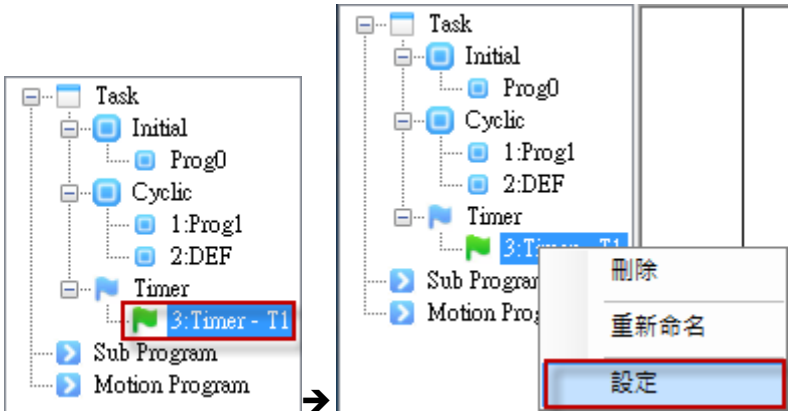
在 Timer 中按鼠标右键，接着点选【新增 Timer program】会出现【新建程序】窗口.



输入程序名称，最多不可超过 16 个字符限制，输入完成后按下【确定】即完成.

■ 设定TIMER TASK

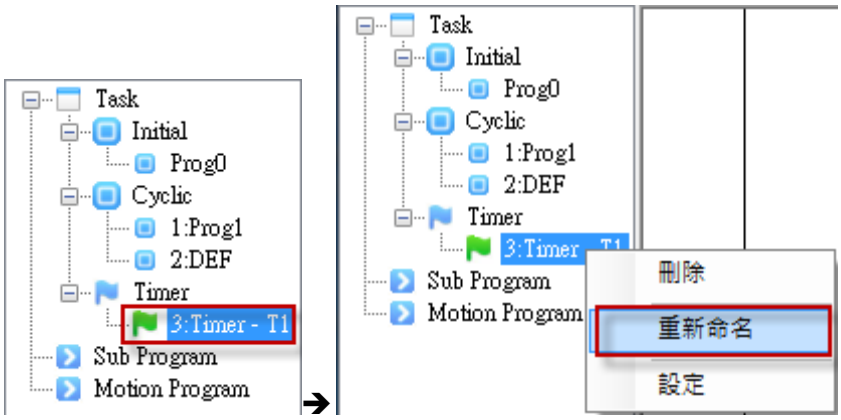
Timer Task 设定是每个 Timer Task 自行独立设定的，在程序名称中按鼠标右键，接着点选【设定】会出现【Timer task 设定】窗口.



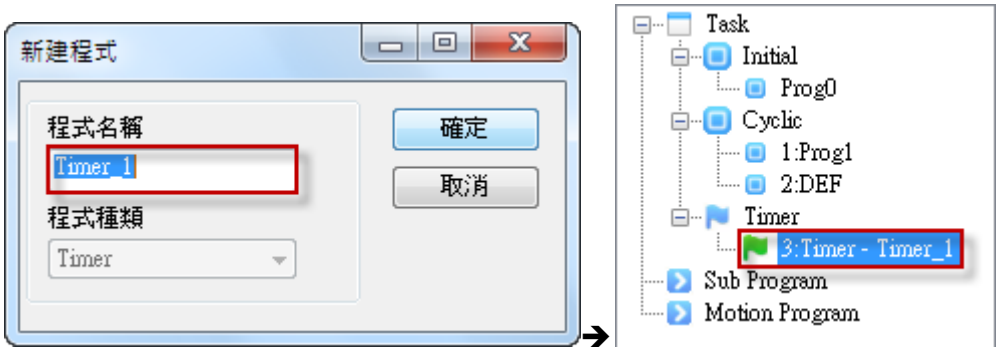
于窗口中输入该 Timer Task 的时间间隔，设定单位为 ms，输入范围最小为 1ms，最大为 30000ms.

■ 变更程序名称

在程序名称中按鼠标右键，接着点选【重新命名】会出现【新建程序】窗口.



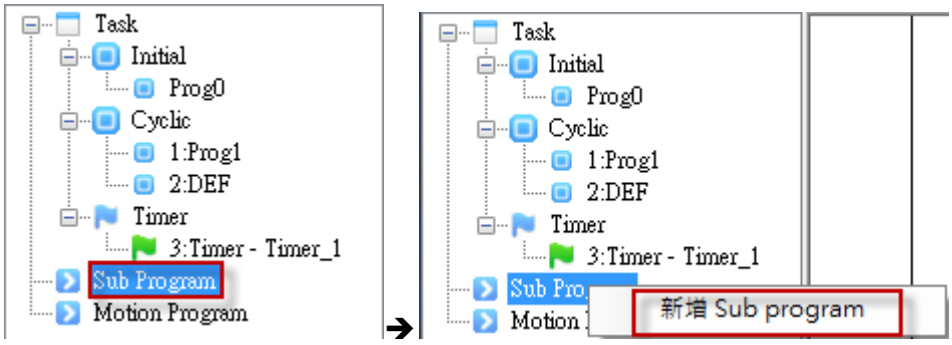
输入变更程序名称，输入完成后按下【确定】即变更完成.



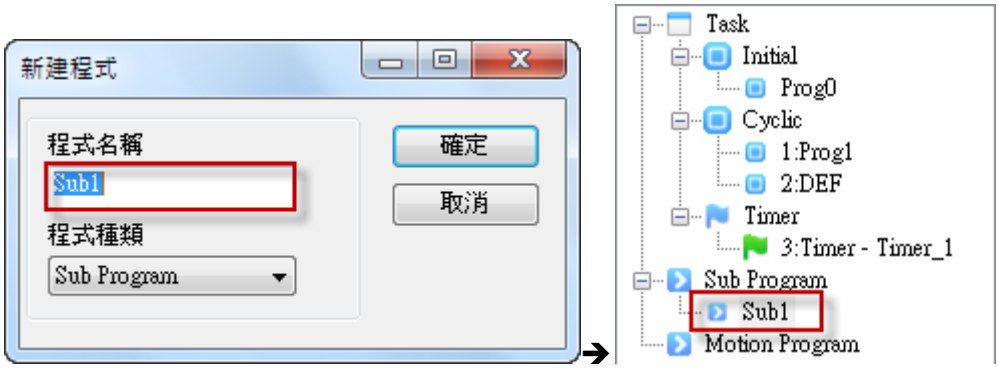
➢ 子程序(SUB PROGRAM)

■ 新增SUB PROGRAM

在 Sub Program 中按鼠标右键，接着点选【新增 Sub Program】会出现【新建程序】窗口.

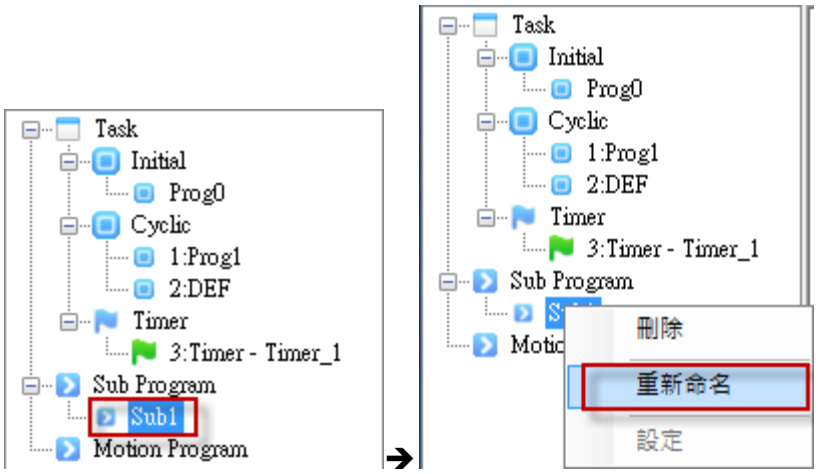


输入程序名称，最多不可超过 16 个字符限制，输入完成后按下【确定】即完成.

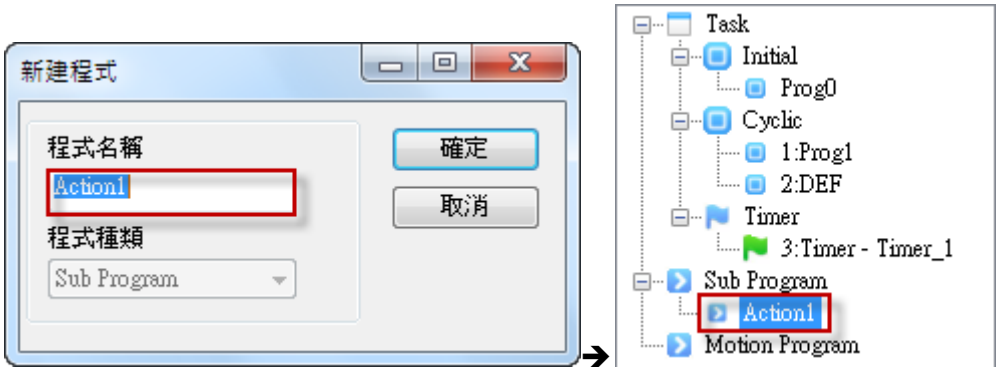


■ 变更程序名称

在程序名称中按鼠标右键，接着点选【重新命名】会出现【新建程序】窗口.



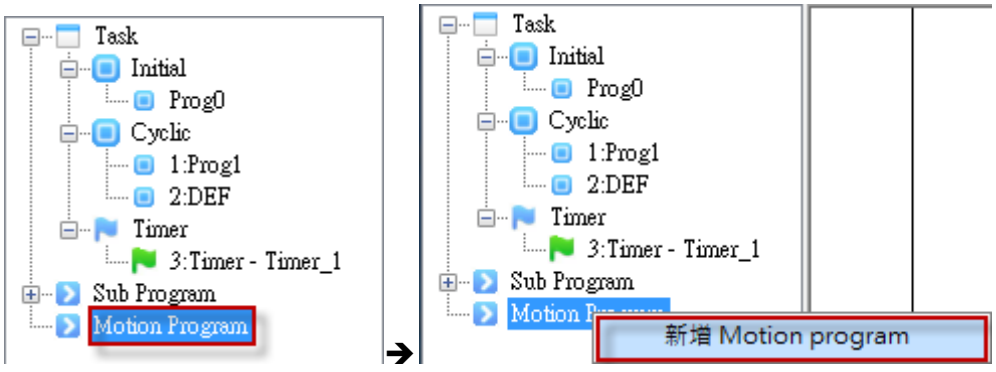
输入变更程序名称，输入完成后按下【确定】即变更完成，同时 Ladder 程序中若有 CALL 此子程序的指令都会自动变更呼叫子程序名称.



➢ 运动程序(MOTION PROGRAM)

■ 新增MOTION PROGRAM

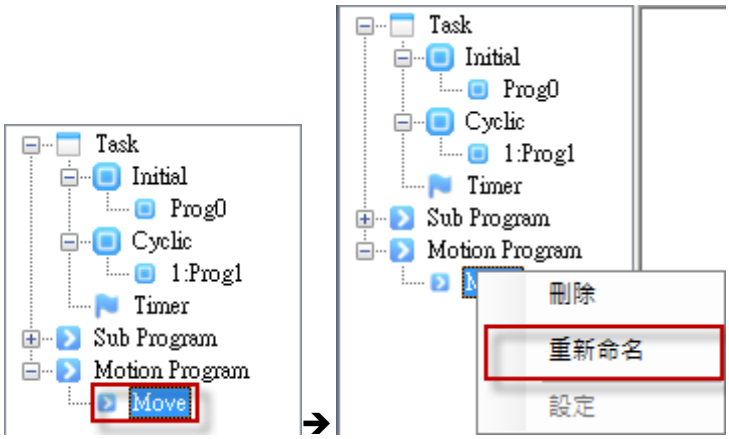
在 Motion Program 中按鼠标右键，接着点选【新增 Motion Program】会出现【新建程序】窗口.



输入程序名称，最多不可超过 16 个字符限制，输入完成后按下【确定】即完成.

■ 变更程序名称

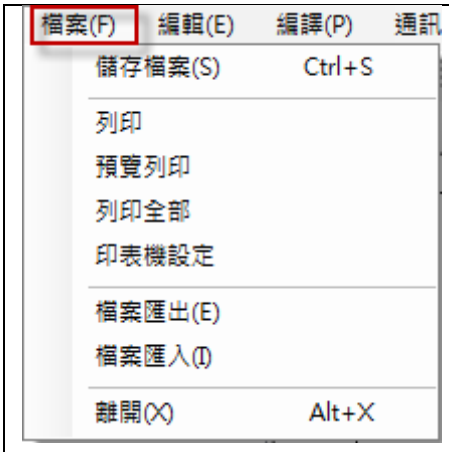
在程序名称中按鼠标右键，接着点选【重新命名】会出现【新建程序】窗口.



输入变更程序名称，输入完成后按下【确定】即变更完成，同时 Ladder 程序中若有 LAUNCH 此运动程序的指令都会自动变更呼叫运动程序名称.

6.3、 其它功能

➢ 档案功能

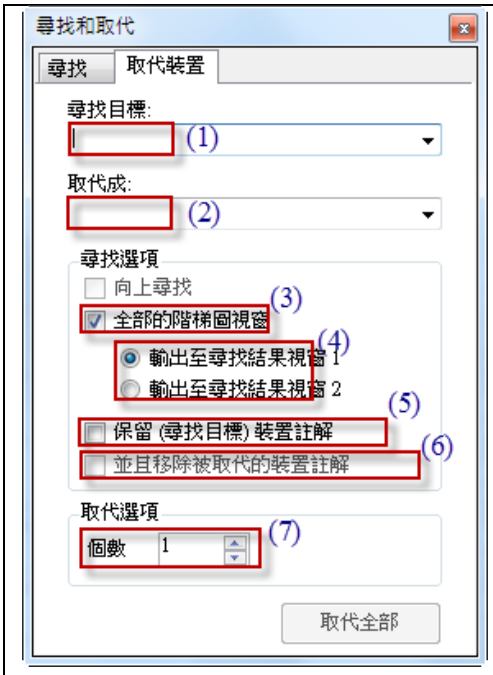


项目名称	叙述内容
储存档案	储存当前 Ladder 程序
打印	打印目前编辑的 Ladder 程序内容
打印预览	打印预览目前编辑的 Ladder 程序内容
打印全部	打印所有未加密的 Ladder 程序内容
打印机设定	设定打印格式，包含纸张大小，边界，方向等
档案汇出	导出 Ladder 程序（.cwp)
档案汇入	由外部汇入 Ladder 程序（.cwp)
离开	关闭 Ladder Editor

➢ 编辑功能

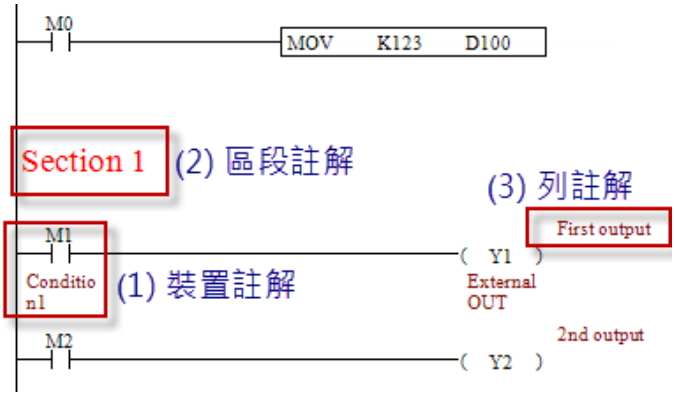
项目名称	叙述内容
选择全部	选择当前 Ladder 程序全部内容
删除	删除选择内容
剪下	剪下选择内容
复制	复制选择内容
贴上	贴上选择内容
寻找	于当前程序或全部程序中寻找目标
取代	于当前程序或全部程序中寻找目标并指定取代装置
跳到	跳到指定的 STEP 位置
跳到起始位置	跳到编辑程序中 STEP 0 的位置
跳到结束位置	跳到编辑程序中 END 指令的位置
装置批注	编辑装置批注
区段批注	编辑区段批注
列批注	编辑列批注
装置总表	开启装置总表窗口
符号总表	开启符号总表窗口

■ 取代

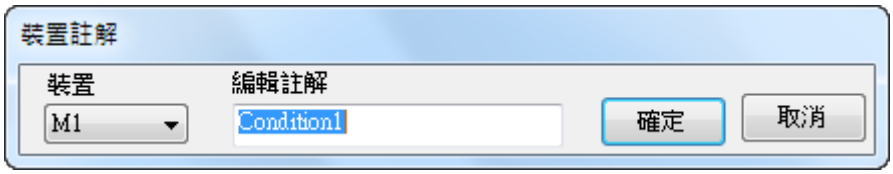


項目	叙述内容
(1)	寻找装置输入
(2)	取代装置输入
(3)	搜寻范围为【目前程序】或【全部程序】
(4)	选择输出结果至【结果窗口 1】或【结果窗口 2】
(5)	将【寻找装置批注】更新至【取代装置批注】
(6)	将【寻找装置批注】更新至【取代装置批注】，并移除【寻找装置批注】
(7)	取代的装置个数

■ 装置批注\区段批注\列批注

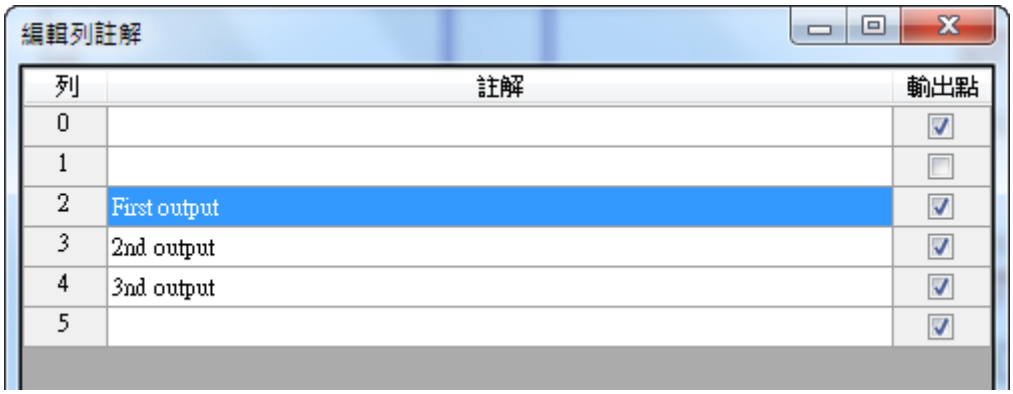


选取装置后，点选【编辑】➔【装置批注】开启编辑输入窗口.



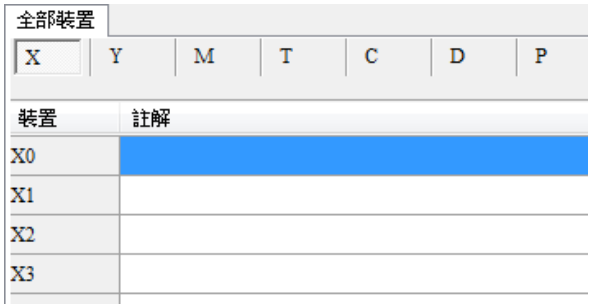
选取空白列后，点选【编辑】➔【区段批注】开启编辑输入窗口.

点选【编辑】➔【列批注】开启编辑列批注窗口.

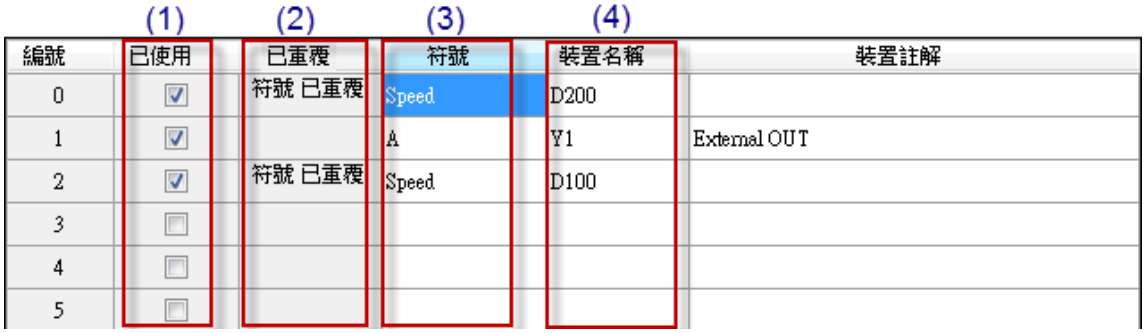


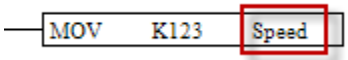
■ 装置总表

能显示所有装置的批注表，依据选择的装置种类显示，并可直接于此表中编辑.

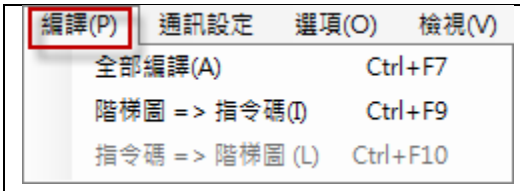


■ 符号总表



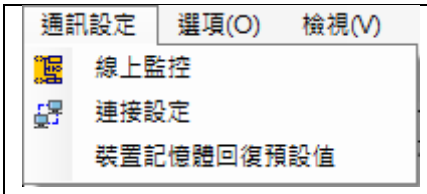
项目	叙述内容
(1)	若勾选代表此装置已在程序中使用
(2)	符号已重复➔不同装置间使用了相同符号 装置已重复➔同一装置使用了两个以上的符号
(3)	装置使用的符号；程序中将以符号取代装置 
(4)	选择使用符号的装置

➢ 编译功能



项目名称	叙述内容
全部编译	所有程序执行编译
阶梯图=>脚本	目前编辑程序执行阶梯图编译为脚本
脚本=>阶梯图	目前编辑程序执行脚本反编译为阶梯图

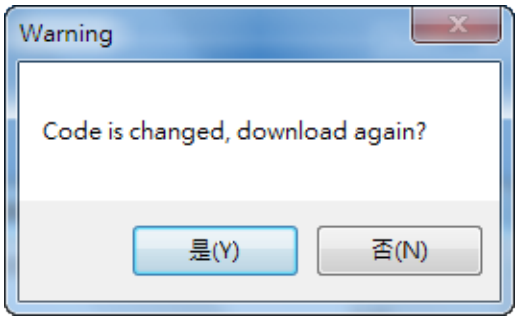
➢ 通信设置功能



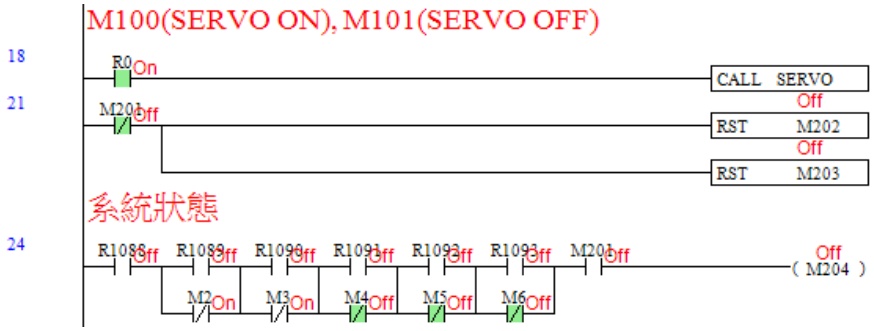
项目名称	叙述内容
在线监控	透过以太网络在线监控 HMC 阶梯图程序执行
连接设定	联机 HMC 之以太网络设定
装置内存回复默认值	将装置内容值皆回复至出厂预设

■ 在线监控

根据联机设定以连接指定 HMC，在线监控前 HMC 程序必须先编译完成，且会比对与 HMC 内部程序与编辑软件上是否一致，不同时会出现提示讯息.

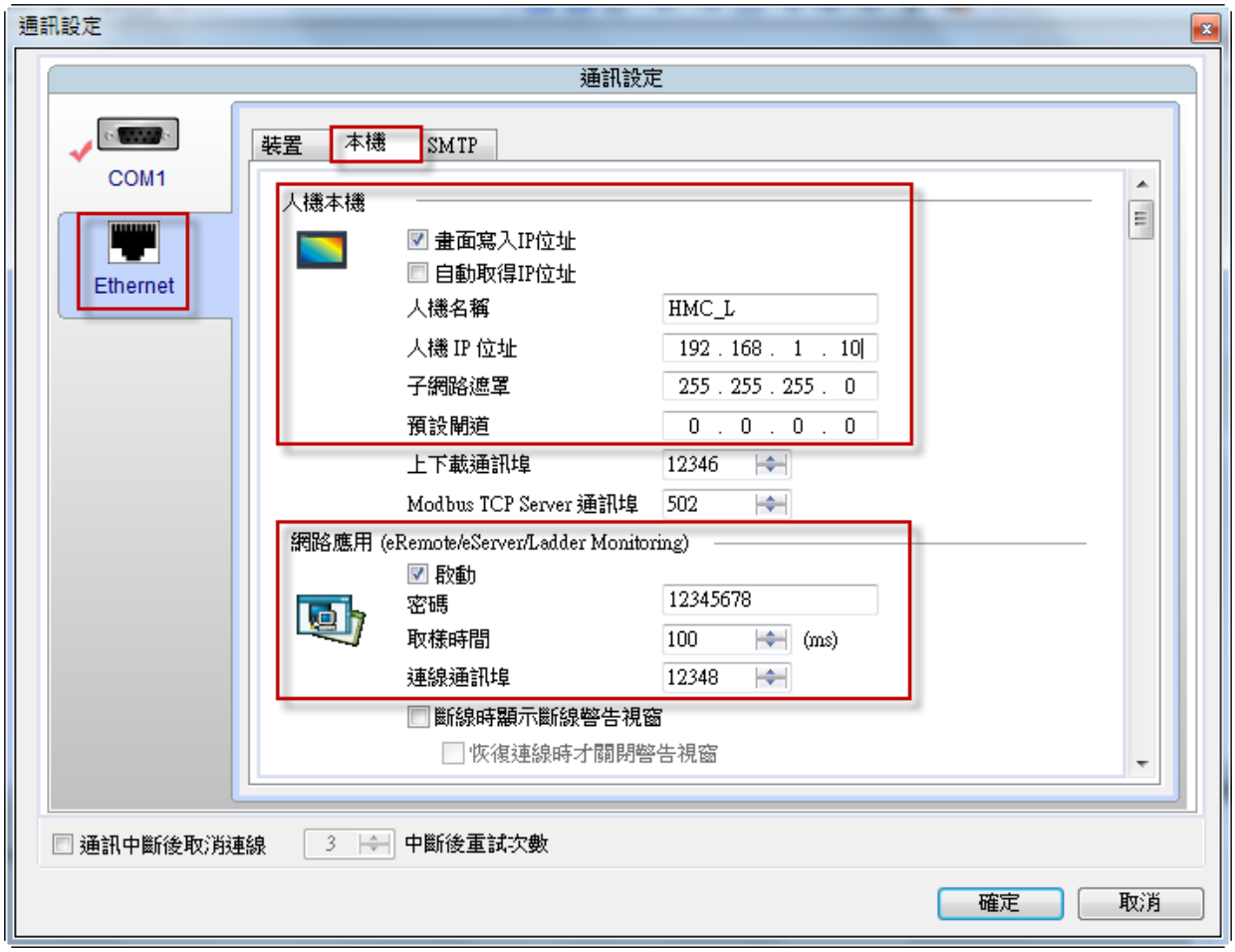


联机成功后，即可监控当前阶梯图执行状态.

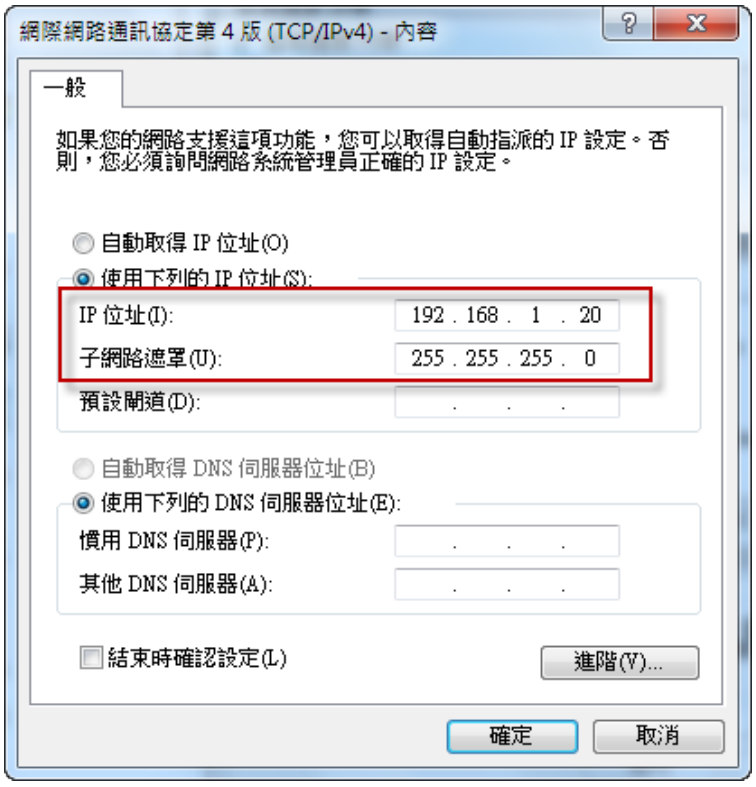


■ 连接设定

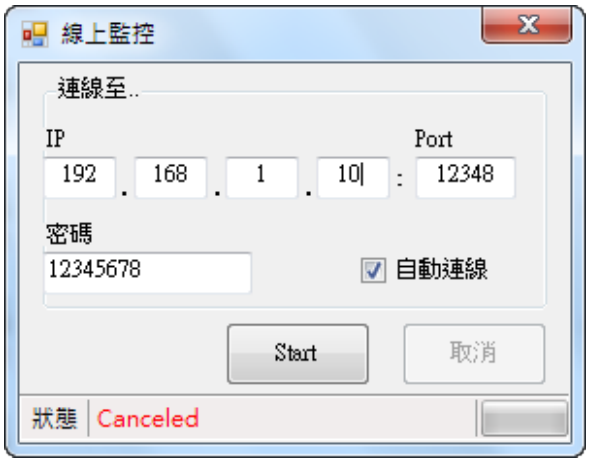
在 DOPSoft 软件中开启【选项】➔【设定通讯参数】中的通信设置窗口，将 IP 地址如下设定(如监控计算机在同一网域内)并勾选启动【网络应用】后将画面下载至人机中.



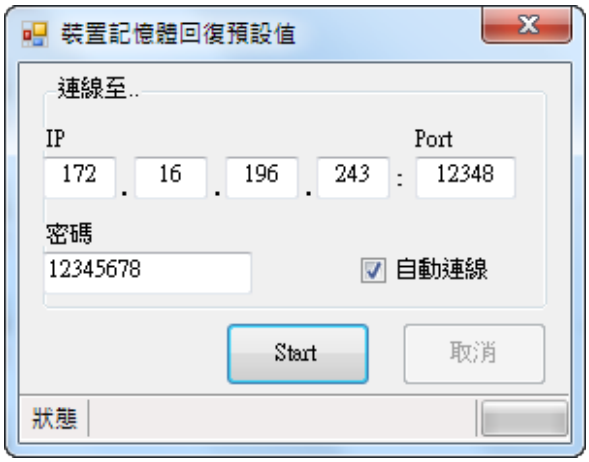
计算机 IP 设定需与 HMC 在同一网域中.



设定欲联机的 HMC IP, 使用 Port 与密码.



■ 装置内存回复默认值



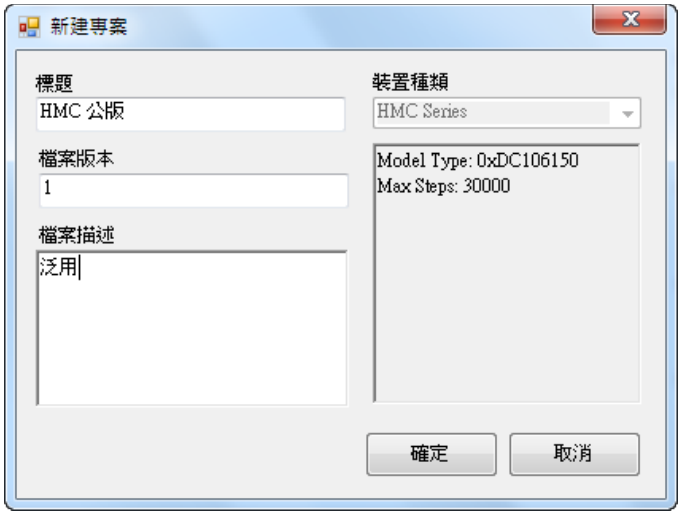
透过 Ethernet 将内存内装置回复至出厂默认值.

➢ 项目功能

项目名称	叙述内容
标题	设定项目版本等信息
设定	设定项目参数
Ladder 上锁	密码验证后，可将指定阶梯图上锁，上锁的程序无法开启与变更.
上锁密码变更	上锁密码变更
群组伺服设定	使用伺服组态设定

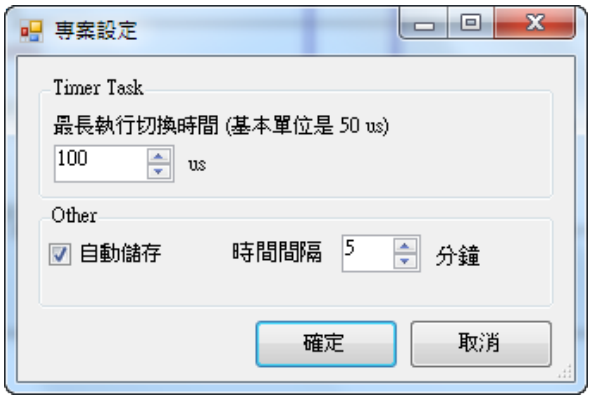
■ 标题

可输入项目标题，档案版本与档案描述.



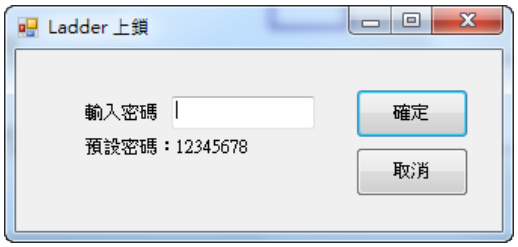
■ 设定

设定项目中定时程序的最长执行切换时间(单位为 us)，与自动储存阶梯图程序时间周期.

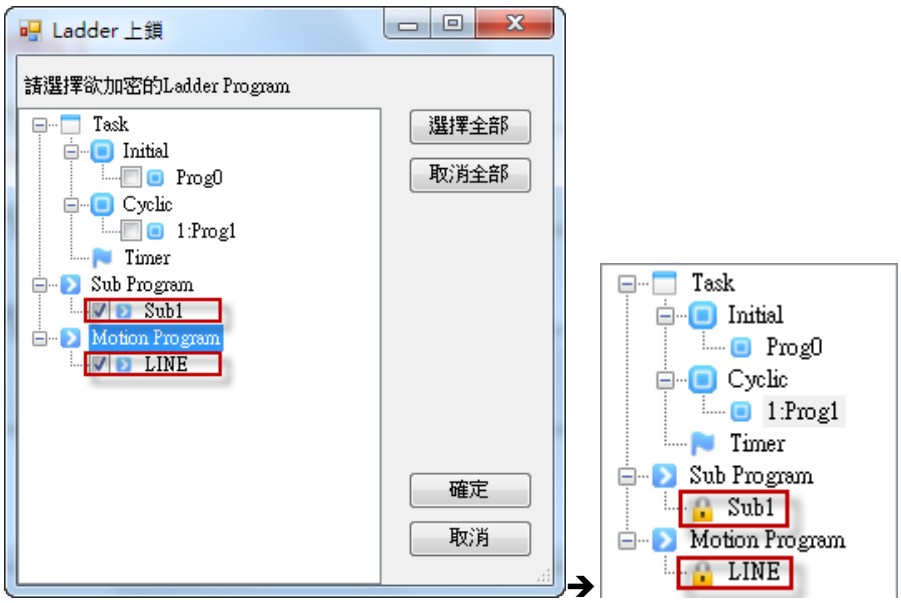


■ LADDER 上锁

需先经过密码验证.

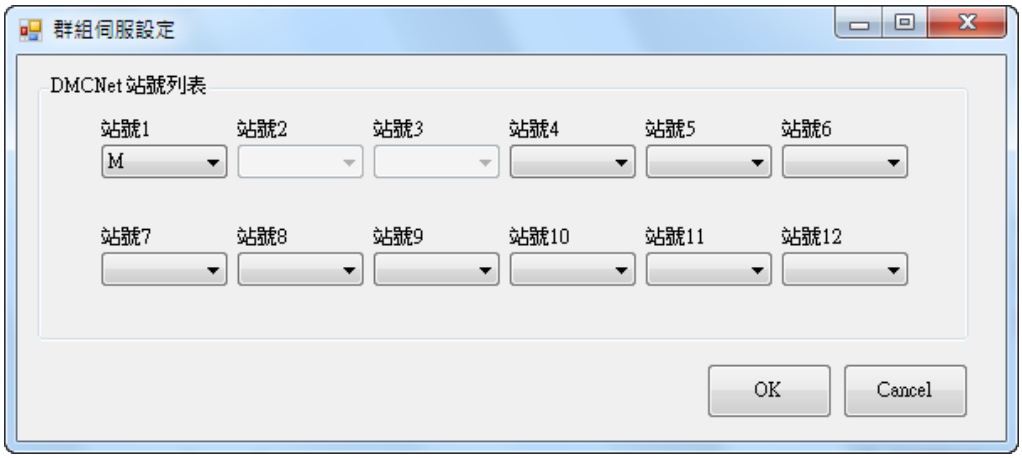


选择欲加密的阶梯图程序，勾选后按下确定．在程序编辑中无法开启或编辑被锁定的阶梯图程序了．

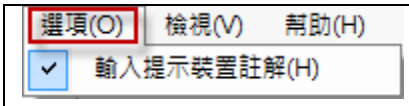


■ 群组伺服设定

设定实际使用的伺服组态，用以实现跨伺服的多轴运动命令下达设定．

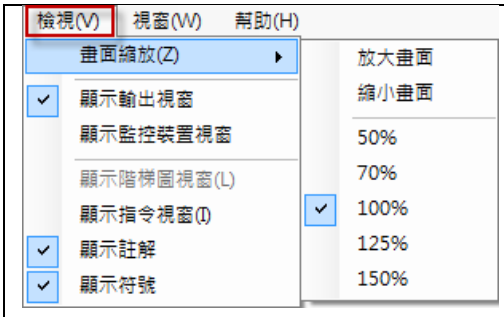


➢ 选项功能



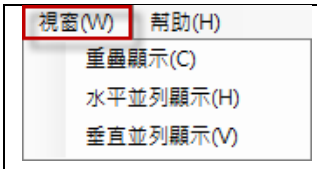
项目名称	叙述内容
输入提示装置批注	输入指令后自动检查装置批注是否存在，若装置无批注，则自动启动批注输入窗口.

➢ 检视功能



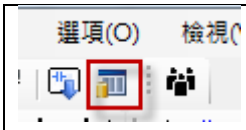
项目名称	叙述内容
画面缩放	编辑窗口可做 50%, 70%, 100%, 125%, 150%的大小缩放
显示输出窗口	显示输出窗口
显示监控装置窗口	显示监控装置窗口
显示阶梯图窗口	阶梯图程序显示
显示指令窗口	指令程序显示
显示批注	是否显示设备批注与列批注
显示符号	是否显示符号或装置

➢ 窗口功能



项目名称	叙述内容
重迭显示	多个阶梯图窗口重迭显示
水平并列显示	多个阶梯图窗口水平并列显示
垂直并列显示	多个阶梯图窗口垂直并列显示

➢ 检视HMC命令历史记录功能

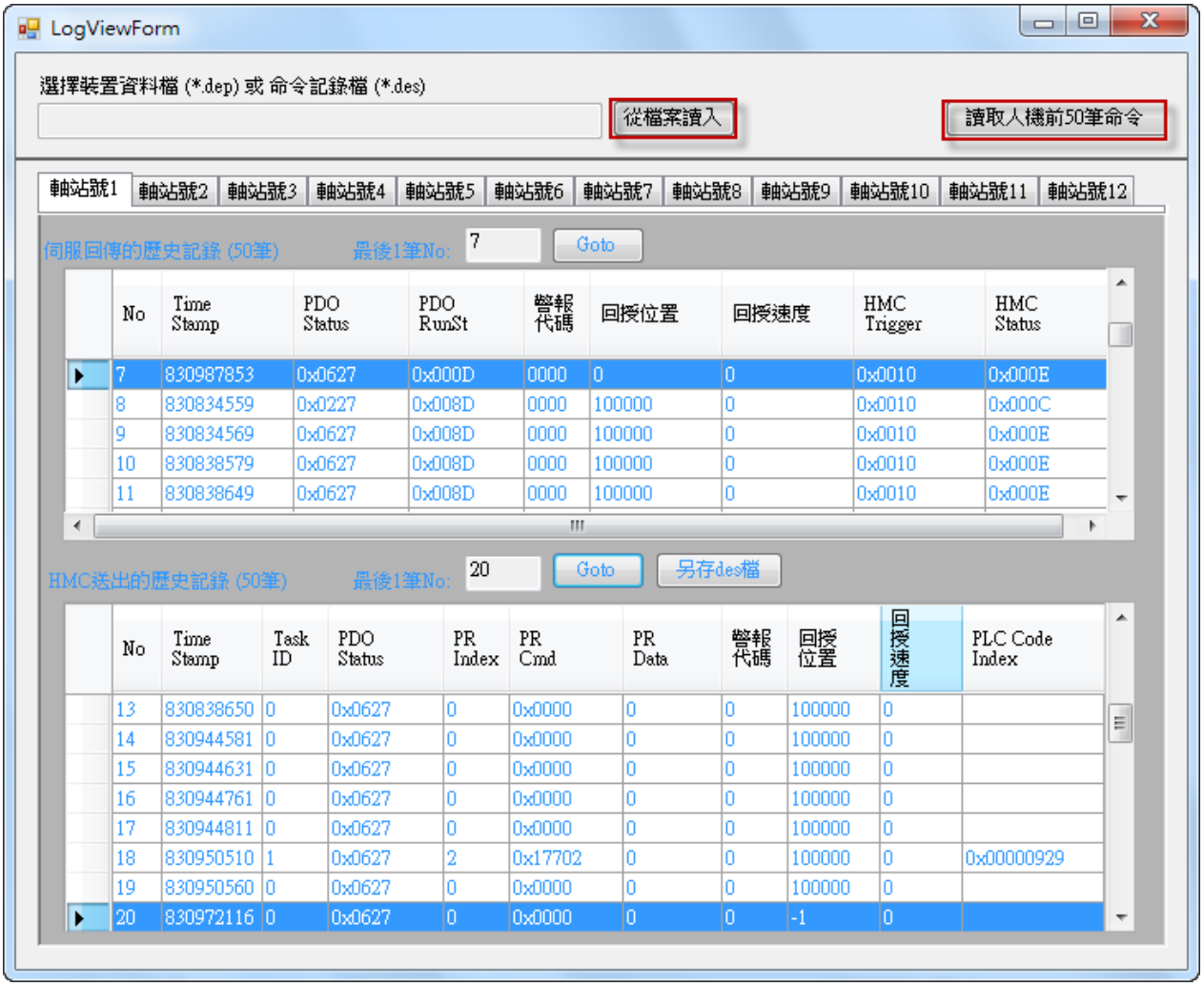


项目名称	叙述内容
检视 HMC 命令历史记录	取得目前 HMC 记录信息: 1.最近 50 笔下达伺服的运动命令记录 2.最近 50 笔运动相关状态变化

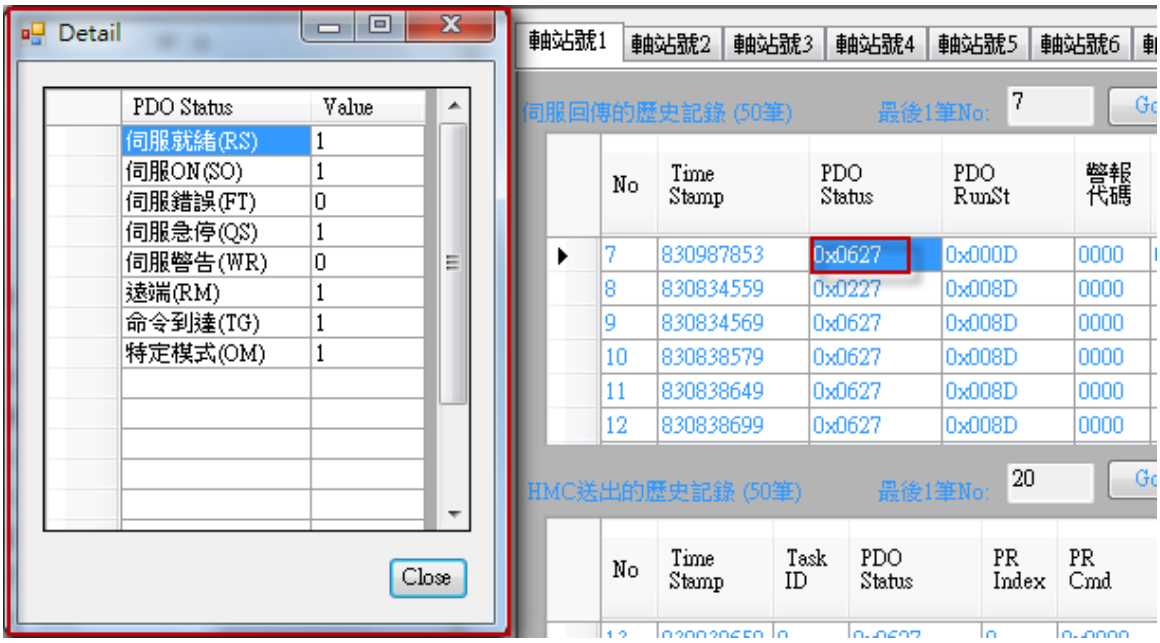
■ LOGVIEWFORM

鼠标左键点击【检视 HMC 命令历史记录】按钮将可打开【LogViewForm】工具窗口.

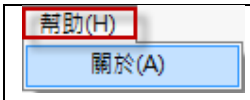
在该工具中可使用【从档案读入】方式, 汇入外部的纪录数据文件(.dep 或.des)以读取记录数据详细内容, 或透过联机 HMC 后以【读取人机前 50 笔命令】方式读取当前 HMC 内的纪录数据.



记录数据报含所有 12 轴数据, 可点选要查看的【轴站号】. 并以鼠标左键双击在要观看的数据字段方式, 打开该字段的详细说明窗口.



➢ 帮助功能



项目名称	叙述内容
关于	Ladder Editor 软件版本信息

7. 附录

7.1、 扩充接脚(含手轮安装)

接脚定义(如右图)	说明
1	24V A (FOR PHASE A、B)
2	PHASE A (手摇轮 PHASE A)
3	PHASE B (手摇轮 PHASE B)
4	GND A (FOR PHASE A、B)
5	GND B (FOR INTERRUPT 0~3)
6	INTERRUPT 0
7	INTERRUPT 1
8	INTERRUPT 2
9	INTERRUPT 3
10	24V B (FOR INTERRUPT 0~3)

注：手轮只需要接 1~4 接脚

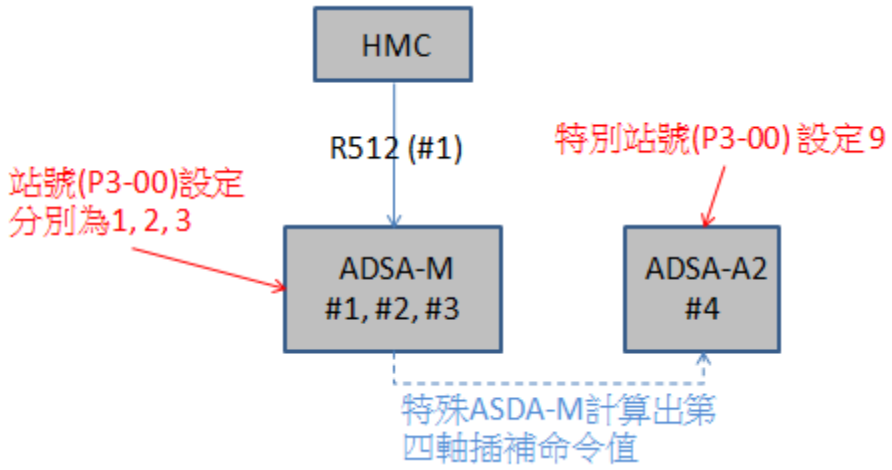
7.2、 总线接脚定义

线条名称	编号名称	说明	12PIN	16PIN	32PIN
白橘	URG_C	紧急开关 B 接点	●	●	●
白橘	URG_C	紧急开关 B 接点	●	●	●
白绿	URG_O	紧急开关 A 接点	●	●	●
白绿	URG_O	紧急开关 A 接点	●	●	●
红	Power	系统电源 24V+	●	●	●
小黑	PGND	系统电源接地	●	●	●
白	EGND	大地接地	●	●	●
黄	422_TX+	RS422: TX+, RS232: TX, RS485: T+/R+			●
白黄	422_TX-	RS422: TX-, RS485: T-/R-			●
黑白	CGND	通讯接地	●	●	●
黑白	CGND	通讯接地			●
黑白	CGND	通讯接地			●
白蓝	LIM_O	限动关关 A 接点			●

白蓝	LIM_O	限动关关 A 接点			●
紫	422_RX+	RS422: R+, RS232:RX			●
白紫	422_RX-	RS422: R-			●
黑橘	INT1	外部中断 1 输入(保留)			●
黑绿	INT0	外部中断 0 输入(保留)			●
红黑	I_GND	外部中断接地			●
白红	I_PW	外部中断电源 24V+			●
RJ45 蓝	DMC	DMCNet 接线	●	●	●
RJ45 黑	ETH	Ethernet 接线			●
RJ45 绿	RIO	RemoteIO 接线		●	●

7.3、 ASDA-M四轴同动驱动器设定与架构

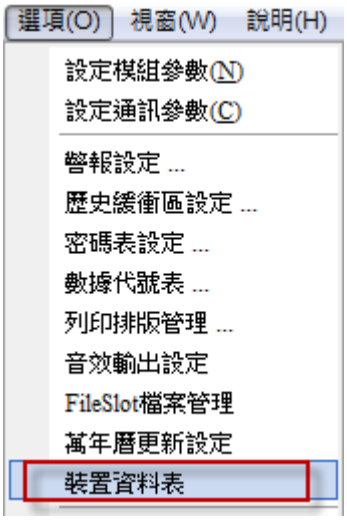
特殊四轴同动控制架构：



7.4、 系统装置数据撷取功能

在 DOPSoft 人机编辑软件中，提供撷取当前【装置数据表】功能. 用户透过此功能，可将 HMC 内的所有断电装置当前状态 ($M, D, W, M, R)与启动异常记录时的 HMC 命令历史记录数据 (!SYS)撷取回来，使用这些数据将可帮助用户实现项目数据复制或远程除错等功能.

在 DOPSoft 软件中开启【选项】➔【装置数据表】中的 Device Data 窗口.



在 Device Data 窗口中，包含以下功能:

a. 【WORD】显示:

每笔数据数值内容为 Word 组成.

b. 【DWORD】显示:

每笔数据数值内容为 DWord 组成.

c. 【10 进位】显示:

每笔数据数值显示格式为有号数十进制(Signed Decimal)表示.

d. 【16 进位】显示:

每笔数据数值显示格式为 16 进制(Hexadecimal)表示.

e. 【从人机上载回来】:

透过 PC 端与 HMC 联机建立完成后，将其 HMC 内部的装置数据撷取回 PC 端，使用的联机方式可以是 USB 联机或以太网络联机.

f. 【下载至人机】:

将当前 Device Data 窗口中的装置数据写入 HMC 中.

g. 【汇入】

将外部数据文件(.dep)以【汇入】方式打开检视.

h. 【汇出】

将当前 Device Data 窗口中的装置数据表储存成档案(.dep)，可使用【检视 HMC 命令历史记录】功能开启此档案即可检视系统的状态履历(!SYS)内容.

i. 下载全部数据时包含装置数据表:

当勾选启动【包含装置数据表】时，在下载画面时会一并将项目文件内的装置数据表下载至 HMC.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
$M0	6	6	0	0	10000	0	0	25	0	0
$M10	0	0	0	0	0	0	0	0	0	0
$M20	0	0	0	0	0	0	0	0	0	0
$M30	0	0	0	0	0	0	0	0	0	0
$M40	0	0	0	0	0	0	0	0	0	0
$M50	100	0	0	0	0	0	0	0	0	0
$M60	0	0	0	0	0	0	0	0	0	0
$M70	0	0	0	0	0	0	0	0	0	0
$M80	0	0	0	0	0	0	0	0	0	0
$M90	0	0	0	0	0	0	0	0	0	0
$M100	0	0	0	0	0	0	0	0	0	0
$M110	0	0	0	0	0	0	0	0	0	0
$M120	0	0	0	0	0	0	0	0	0	0
$M130	0	0	0	0	0	0	0	0	0	0

HMC 命令历史记录数据(!SYS)在以下三种状况将会自动更新为当前状态:

a. 伺服发生 AL.918 异常:

此为伺服过速度异常. 当运转伺服速度超过【最高保护转速 (P1-55)】时，伺服将自动 Servo Off 且输出 DO:BRKR. 发生此异常 HMC 系统将自动把目前命令历史记录数据记录于系统断电保持装置!SYS，以供后续分析.

b. 伺服发生 AL.30 异常:

此为伺服防撞保护发生. 当防撞保护功能到达启动条件时，伺服将自动 Servo Off 且输出 DO:BRKR. 发生此异常 HMC 系统将自动把目前命令历史记录数据记录于系统断电保持装置!SYS，以供后续分析.

c. 将【当前记录断电保持启动】(R500)置 ON:

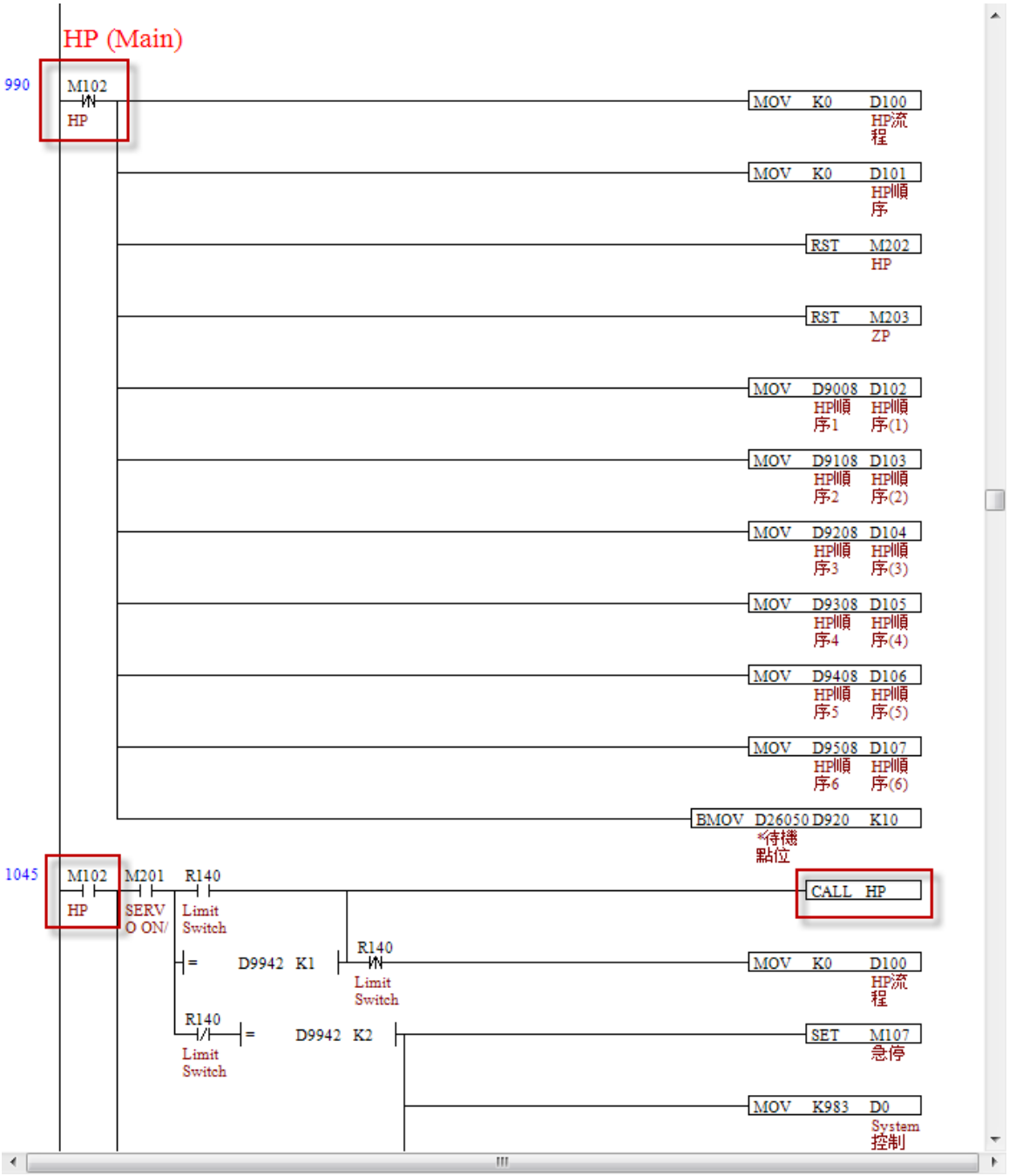
当启动断电保持记录旗标时, HMC 系统会将自动把目前命令历史记录数据记录于系统断电保持装置!SYS.

8. 程序设计错误范例

8.1、 旗标上升缘初始

a. 错误写法:

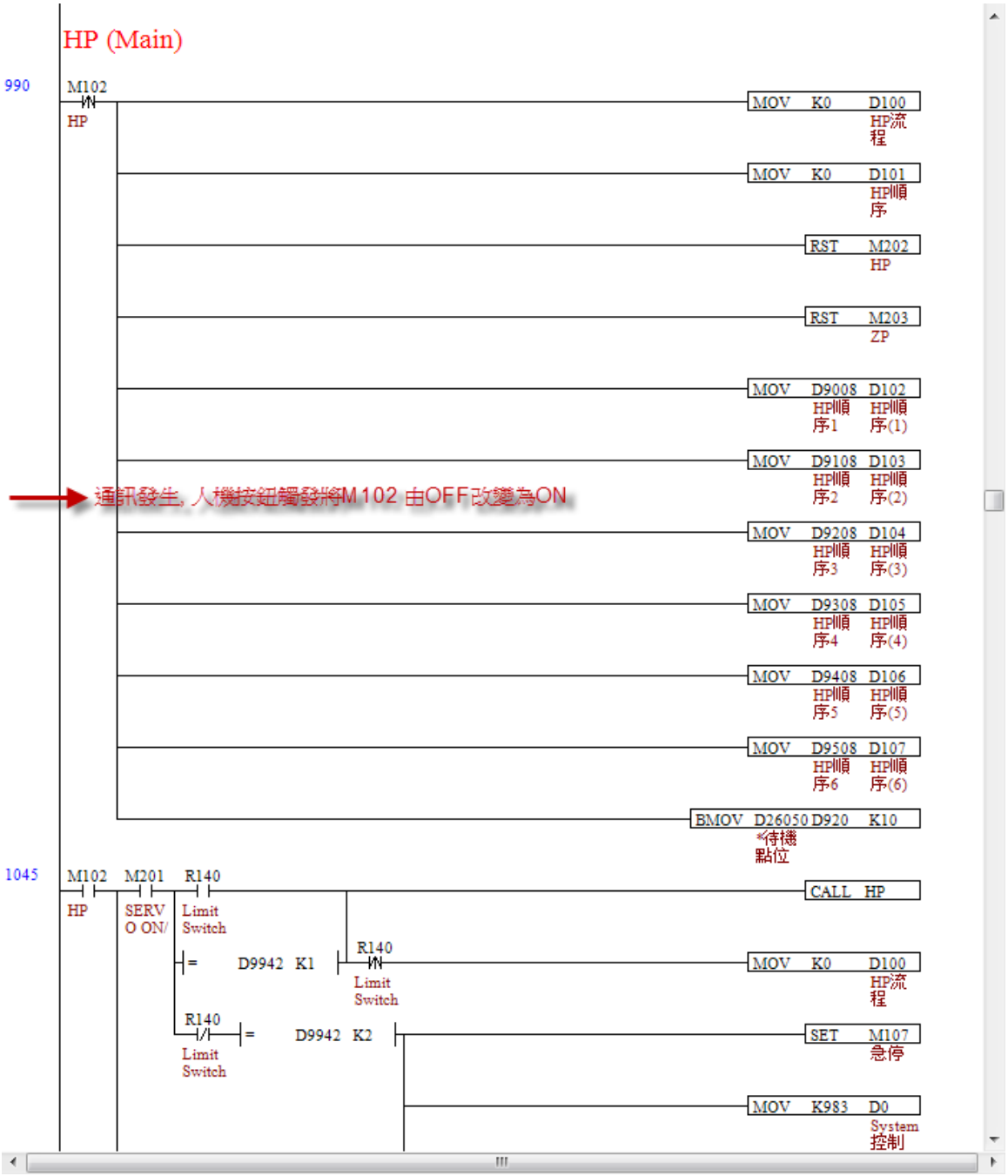
- M102 为人机按钮触发并设为 ON, 代表开始进行复归动作
- M102 上升缘为进行复归动作的初始设定.
- M102 为 ON, 进入子程序【HP】进行复归动作 (复归执行完毕后清除 M102).



b. 原因:

人机与控制器为时时通讯，若透过人机将 M102 设为 ON 的发生时机如下，将造成尚未初始设定前，即开始进行复归动作．发生时序过程如下:

- M102 上升缘判断不成立，不执行复归初始 (M102 尚为 OFF)
- M102 改变为 ON (发生时机点如下图)
- M102 为 ON 进入【HP】，开始复归动作，但因尚未初始可能为错误动作.
- 下一次扫描, M102 上升缘判断成立，但是动作已开始后才再执行复归初始，顺序错误.



c. 建议写法:

增加初始完成旗标，在此例中 HP 初始完成旗标为 M4070，以确保一定需完成初始设定后才能进入动作执行.

- M102 上升缘为进行复归动作的初始设定，初始完成后设立完成旗标.
- M102 为 ON，且需完成初始设定(M4070 为 ON) 才能进入子程序【HP】进行复归动作.
- 复归执行完成或不执行时(M102 为 OFF)，需清除完成旗标(RST M4070).