

ISPSoft 软件使用手册



ISPSoft 软件使用手册



中达电通 股份有限公司

公司网址: www.deltagreentech.com.cn

北京: 010-8225-3225	重庆: 023-8806-0306	济南: 0531-8690-6277	武汉: 027-8544-8475
太原: 0351-4039-485	广州: 020-3879-2175	南昌: 0791-6255-010	西安: 029-8669-0810
长春: 0431-8859-6018	哈尔滨: 0451-53665568	南京: 025-8334-6585	厦门: 0592-5313-601
长沙: 0731-8827-7881	杭州: 0571-8882-0610	上海: 021-6301-2827	郑州: 0371-6384-2772
成都: 028-8434-2072	合肥: 0551-2816-777	沈阳: 024-2334-1159	南宁: 0771-5875-699
天津: 022-2301-5082	乌鲁木齐: 0991-611-8160		

*本使用手册内容若有变更, 恕不另行通知

DVP-4959610-10

I SPSoft 软件使用手册

版本修订一览表

版本	变更内容	发行日期
第一版	第一版发行	2012/01/06
第二版	第二版发行	2012/06/30
第三版	第三版发行	2012/10/31
第四版	增加附录 D 及附录 E	2015/03/18
第五版	<ol style="list-style-type: none">1. 第 3 章加入 AHxxEMC 系列与 AS300 系列机种相关内容。2. 新增章节：第 8 章自定义数据类型、第 9 章轴、第 15 章连续功能图 CFC、第 21 章 G-Code 编辑器与 E-CAM 编辑器及第 22 章向导工具。3. 附录 B 加入 AHxxEMC 系列与 AS300 系列机种相关内容。4. 删除附录 D 及附录 E。5. 更新所有章节软件画面。	2016/05/18
第六版	<ol style="list-style-type: none">1. 第 2 章加入 AS300 仿真器功能介绍。2. 第 3 章 AHxxEMC 系列与 AS300 系列增加及修改功能。3. 第 9 章轴参数增加及修改功能。4. 第 14 章 SFC 增加下拉 Action 列表功能。5. 第 16 章增加 DVP 转 AS 功能及装置备注新增功能。6. 第 21 章 ECAM 表新增文件模式等多项功能及新增 Polynomial 函数功能。7. 第 22 章定位规画表轴设定画面变更及各种轴输出表格项目修改；数据示波器、数据记录器接口修改及新增双垂直量测线。8. 附录 A_USB 联机新增 WINDOWS7、WINDOWS8 及 WINDOWS10 操作系统 USB 联机方式。9. 附录 B 各系列机种注意事项修改 AS300 系列 S 装置范围。	2017/09/15
第七版	<ol style="list-style-type: none">1. 第 2 章自动储存内容增加备份文件数量设定及修改 AH 仿真器启动数量与限制内容。2. 第 3 章增加 AH500 冗余系列机种的 HWCONFIG 相	2017/12/29

版本	变更内容	发行日期
	<p>关操作介绍及新增系统配置区冗余扩展背板相关操作介绍；AH500 机种之 CPU 基本参数 - 名称/注释内容修改；AHxxEMC 机种 HWCONFIG 的 PLC 主机参数设定内，多增加本体 IO 滤波输入。</p> <ol style="list-style-type: none"> 3. 第 5 章新增 AH500 冗余系列机种的工作 (Task) 管理工具内容。 4. 第 6 章 AS 系列机种支持数据类型内容修改；AH 系列机种 S 装置支持内容修改。 5. 第 7 章新增 AS 系列机种功能块内部区域符号新增不支持的数据类型；功能块内存配置内容新增 AH500 冗余系列机种；功能块的基本规格内容修改。 6. 第 8 章新增结构项宣告结构及列举数量限制；新增结构改变列举及列举改变结构内容。 7. 第 9 章新增轴项目右侧周期时间的显示；新增「与 ECAT Builder」同步选项介绍；轴参数设定之输出装置 ID 设定内容修改。 8. 第 10 章 MPS 指令介绍；梯形图的允许与失效内容修改。 9. 第 13 章新增 AH/AS 系列机种 ST 语言不支持的指令列表及 AS 系列机种之 ST 语言有支持的指令但有所限制的规范事项。 10. 第 14 章修改 STEP 符号占用 PLC 内部的 S 装置内容；增加 AH500 冗余系列机种执行「Final Scan」内容。 11. 第 16 章修改转换机种可能导致部分程序遗失需存盘之提示字样。 12. 第 17 章新增 DVP 机种在线编辑介绍；PLC 系统信息及 PLC 系统记录内容修改增加 AH500 冗余机种相关显示信息及操作内容。 13. 第 20 章新增 CARD Utility 对 AH500 冗余系列机种相关内容。 14. 第 21 章叙述本版不支持 G-CODE 功能；ECAM 文件模式新增图表检视功能。 15. 第 22 章定位规画表之定位表格内，地址模式新增相对模式 (符号/地址)、绝对模式 (符号/地址) 相关设定内容；定位规画表新增符号装置仿真数值设定内容。 16. 附录 B 新增 AHCPU560-EN 机种内容。 	

版本	变更内容	发行日期
第八版	<ol style="list-style-type: none"> 1. 修改第 1 章 · 新增 DVP15MC 机种内容 · 以及安装 / 删除 ISPSOFT 内容修改。 2. 修改第 2 章 · 新增 DVP15MC 机种内容； DVP 仿真器内容及 AS 仿真器启动数量限制等内容修改；「封包绕送模式」功能内容修改。 3. 修改第 3 章 · 新增 DVP15MC 机种内容 · DVP15MC 机种 HWCONFIG 环境介绍及 PLC 主机参数设定介绍等新增内容；硬件规划工具 - HWCONFIG 内容修改；AS 系列机种的 PLC 主机参数设定内容修改；HWCONFIG 硬件参数管理及在线诊断内容修改。 4. 修改第 4 章 · (允许/失效) 区段相关内容修改。 5. 修改第 5 章 · 新增 DVP15MC 机种工作管理及 DVP15MC 机种工作管理工具操作等相关内容修改；POU 的管理操作相关内容修改；新增功能块 POU 锁定 (永久) 功能介绍；修改范例实作内容。 6. 修改第 6 章 · 新增 DVP15MC 机种内容；变数符号的数据类型内容修改 · 新增 DVP15MC 机种数据类型内容、数据类型说明表格及相关内容修改；变数符号的地址配置与初始值内容修改；变数符号的变址寄存器修饰内容修改；新增变数符号 · 对于字串、群组类型符号使用原则等内容修改；下载符号的初始值内容修改 · 新增 DVP15MC 机种相关内容；符号表的导入内容修改；范例实作内容修改。 7. 修改第 7 章 · 新增 DVP15MC 机种内容 · 功能块的输入与输出针对 DVP15MC 机种等相关内容修改；范例与实作内容修改；DVP15MC 机种对函示库内容新增。 8. 修改第 8 章 · 新增 DVP15MC 机种内容等相关内容。 9. 修改第 9 章 · 新增 DVP15MC 机种内容 · 轴操作及相关内容修改；AHxxEMC 建立轴相关内容修改；轴参数设定及监控与调测相关内容修改。 10. 修改第 10 章 · 新增 DVP15MC 机种内容；设置标签 (Label) 内容修改。 11. 修改第 11 章 · 设置标签 (Label) 内容修改。 12. 修改第 14 章 · 新增 SFC 的系统结构内容。 13. 修改第 16 章 · 新增 DVP15MC 机种内容；修改 DVP 转 AS 内容；修改项目下载与上传介绍内容；修改 DVP、AH/AS 机种的装置注释表相关内容；修改装置使用状况内容。 	2018/05/31

版本	变更内容	发行日期
	<p>14. 修改第 17 章 · 新增 DVP15MC 机种内容；程序监控的改变现在值内容修改；DVP 仿真器内容修改；除错模式内容修改；PLC 系统信息 · AH500 冗余系列「Redundant」分页等内容修改；轴记录内容修改。</p> <p>15. 修改第 18 章 · 新增 DVP15MC 机种内容。</p> <p>16. 修改第 19 章 · NWCONFIG 简介内容修改。</p> <p>17. 修改第 20 章 · AS 系列机种备份档名称、路径内容修改。</p> <p>18. 修改第 21 章 · ECAM 编辑器使用内容修改。</p> <p>19. 修改第 22 章 · 定位规划表支持机种及相关内容修改；数据示波器内容修改；数据记录器触发取样等内容修改；高速计数向导内容修改。</p> <p>20. 修改附录 A · 新增 AH560EN2 机种的驱动程序路径内容。</p> <p>21. 修改附录 B · 新增 DVP15MC 机种内容；各机种数据交换及 NWCONFIG 支持状况内容新增修改；编译与上下载时间内容修改；AHCPU501-EN、AHCPU501-RS2、AHCPU521-RS2、AHCPU531-RS2、AS200 机种内容新增；AHCPU511-EN/ AHCPU511-RS2 及 AHCPU531-EN 等机种定址范围修改。</p>	
第九版	<p>1.修改第 1 章 · 新增 DVPxxMC 机种叙述；安装 ISPSOft 内容及移除 ISPSOft 内容修改；安装 COMMGR 内容及移除 COMMGR 内容修改；新增 ISPSOft PLC 机种支持的语言支持列表及功能支持列表。</p> <p>2.修改第 2 章 · 进入 ISPSOft 的第一步及窗口标题与状态栏内容修改；功能工具栏与图标工具栏内容修改；项目管理区与讯息显示区及编辑工作区内容修改；ISPSOft 的项目架构 · 单机项目与群组项目、整合运动模块项目内容修改；系统及环境设定内容修改；通讯管理员 COMMGR 的启动与关闭及驱动管理、建立联机通道内容修改；建立 ISPSOft 与 COMMGR 之间的连接与连接主机与通讯端口内容修改。</p> <p>3.修改第 3 章 · DVPxxMC 机种 PLC 主机参数设定内容修改；AH560 冗余机种参数设定与相关说明内容修改；硬件规划工具 - HWCONFIG 内容修改。</p> <p>4.修改第 4 章 · 新增项目及硬件组态与模块配置等相</p>	2019/4/22

版本	变更内容	发行日期
	<p>关内容修改 ;主机与模块参数设定及建立程序内容修改 ; 基本编辑-梯形图区段选取与操作功能增加 ; 机本编辑-插入 API 应用指令、程序的检查与编译内容修改 ; 测试与除错建立联机、下载程序与组态参数、联机测试范例实作内容修改。</p> <p>5.修改第 5 章 · ISPSOft 的 POU 架构与 ISPSOft 中的工作(TASK)管理内容修改 ; 项目管理区的工作(TASK)项目与工作(TASK)中的 POU 执行顺序内容修改 ;POU 的建立与属性设定及 POU 的启动状态、POU 的删除与复制、POU 密码的设定与解除、导出程序 POU、导入程序 POU、导出功能块 POU、导入功能块 POU 等内容修改 ; 工作(TASK)管理的操作及工作(TASK)属性与中断条件的设定、POU 的配置、POU 的排序的内容修改 ; 范例实作 POU 与 TASK 的编程范例、DVP 机种的中断程示范例、AH500 机种的中断程序范例等内容修改。</p> <p>6.修改第 6 章 · 变量符号的类别与变量符号的数据类型及变量符号的位置配置与初始值内容修改 ; 变量符号表与新增变量符号及数组与字符串类型符号的使用原则、变量符号的修改与符号表编辑、下载变量符号的初始值、装置资源分配设定内容修改。</p> <p>7.修改第 7 章 · 功能块的变量符号与功能块的输出与输入内容修改 ; 功能块调用功能块的巢式结构与功能块的内存配置及功能块的基本规格内容修改 ; 建立范例程序内容修改 ; 新增使用者自定义函数库及引用使用者自定义函数库、引用台达函数库内容修改。</p> <p>8.修改第 8 章 · 建立结构(STRUCTURE)内容修改。</p> <p>9.修改第 9 章 · AHxxEMC 机种的建立轴内容修改 ; DVPxxMC 机种的建立轴与轴基本设定内容修改 ; 全局符号表的主符号表与轴符号表内容修改 ; AHxxEMC 机种的轴参数设定与 DVPxxEMC 机种的轴参数设定内容修改 ; AHxxEMC 机种的监控与调测与 DVPxxEMC 机种的监控与调测内容修改。</p> <p>10.修改第 10 章 · ISPSOft 的 LD 编辑环境内容修改 ; 于 LD 中使用装置变量符号及常数、插入 API 应用指令与功能块、编辑区块逻辑指令、符号模式与位置模式显示切换内容修改。</p> <p>11.修改第 11 章 · ISPSOft 的 FBD 编辑环境及于 FBD</p>	

版本	变更内容	发行日期
	<p>中使用装置变量符号及常数、API 应用指令比较接点与功能块内容修改；符号模式与地址模式的显示切换内容修改。</p> <p>12.修改第 12 章·编辑 IL 指令、插入 API 应用指令与功能块内容修改。</p> <p>13.修改第 13 章·ISPSOft 的 ST 编辑环境、编辑 ST 叙述、插入 API 应用指令与功能块、建立范例程序内容修改。</p> <p>14.修改第 14 章·步进点与 Action、转换点 (Transition)、跳跃点、Action 的修饰条件、条件定义、初始步进点等内容修改；内部属性、Step 属性参数、Action 与 Transition 属性参数内容修改；ISPSOft 的 SFC 编辑环境、Action 与 Transition 的建立与管理、指定步进点与转换点符号、配置 Action/折迭 Action 列表内容修改；建立范例程序内容修改。</p> <p>15.修改第 15 章·指令与功能块内容。</p> <p>16.修改第 16 章·新增变更 PLC 机种列表·新增项目的下载与上载列表·DVP 机种的装置注释表及装置使用状况、程序比对内容修改。</p> <p>17.修改第 17 章·DVP 机种的除错模式、AH500 机种的仿真器内容修改；PLC 系统信息、PLC 系统纪录内容修改。</p> <p>18.修改第 18 章·ISPSOft 的密码架构、POU 密码、子程序密码、其他密码的数据保护功能内容修改。</p> <p>19.修改第 19 章·NWCONFIG 简介、相关基础介绍、设定通讯参数、工作流程内容修改；部署节点、连接网络、节点与网络属性的设定、装置或网络的隐藏/显示、合法的网络架构、下载绕送路径表、绕送路径测试内容修改；选择主站装置、设定通讯参数、数据交换表简介、设定区块、数据交换表的机种同步性、数据交换表的管理操作、PLC Link 的监控功能、设置 PLC Link 的注意事项内容修改；Ether Link 简介、开启 Ether Link 的设置窗口、数据交换表的建立与管理、装置列表与图标的操作、Ether Link 的运作模式、下载 Ether Link 设置、上载 Ether Link 设置、删除异步组件、在线启动/停止 Ether Link、整体下载、ISPSOft 中的绕送应用内容修改。</p>	

版本	变更内容	发行日期
	<p>20.修改第 20 章，数据备份储存卡操作、PLC 永久备份设定、储存卡工具 CARD Utility 软件介绍、备份操作、还原操作内容修改。</p> <p>21.修改第 21 章，删除 G-Code 编辑器内容；E-CAM 编辑器、关于 E-CAM(电子凸轮)、电子凸轮意义说明、使用 E-CAM 编辑器内容修改。</p> <p>22.修改第 22 章，设定定位规划表、定位规划表的模拟、上载与下载定位规划表内容修改；轴控模式-单轴点对点模式、单轴多段输出模式、双轴直线插补模式、双轴圆弧插补模式内容修改；关于数据示波器、开启数据示波器、设定取样参数、取样模式、量测内容修改；关于数据纪录器、开启数据纪录器、设定取样参数、监看与录制模式、量测内容修改；关于高数计数器向导内容修改。</p> <p>23.修改附录 A，在 Windows10 环境下安装 USB 驱动程序内容。</p> <p>24.修改附录 B，DVPxxMC 机种的装置种类、DVPxxMC 机种之装置的寻址格式、DVPxxMC 机种之停电保持装置、DVPxxMC 系列机种内容修改。</p>	
第十版	<ol style="list-style-type: none"> 1.所有章节因增加第 16 章，内容提及第 16 章-第 22 章内容更新为调整后章节。 2.修改第 1 章功能支持列表更新。 3.修改第 2 章最近使用项目的显示方式变更，增加功能列 PLC 主机里的初始值选项。 4.修改第 3 章增加 HWCONFIG 4.0 的使用说明。 5.修改第 4 章 IEC61131-3 文字。 6.更新第 5.3.2 节 EI 程序图。 7.修改第 6 章全局变量表增加分类功能与宣告变量的初始值说明。 8.修改第 10 章书签定位功能会显示链接在项目树内。 9.修改第 13 章支持单行注解与/* */注解说明，输入可以自动带出结构。 10.增加第 16 章 C 语言。 11.修改第 17 章增加查找装置与查找变量说明。 12.修改第 18 章增加在线编辑可以暂时记住密码选项。 13.修改第 22 章 G-Code 支持。 	2019/12/20

版本	变更内容	发行日期
第十一版	<ol style="list-style-type: none"> 1.在各章节增加 AS516E-B、AS524C-B(简称为 AS5xx 机种) 相关资料。 2.修改第 1 章系统需求、语言支持列表、功能支持列表更新。 3.修改第 2 章工具>选项里面的 ISPSOft 预载项目内的功能项目，增加编辑器设定的 IL 及 ST 和 C 支持提示的选项设定，增加通信设置的 COMMGR 按钮说明，增加导出程序到文字格式。 4.第 3 章将第 3.1.2 节及第 3.1.3 节内容合并，增加第 3.1.2.8 节功能工具栏-常用，增加 IO-Link 及 HWCONFIG 符号设定说明。修改第 3 章 HWCONFIG 4.0 的章节编排顺序与使用说明，包含界面更新、HWCONFIG 符号功能、IO-Link 功能等。 5.增加第 5.2.1 节与第 5.4.1 节对函数的介绍。 6.增加第 6.1.2 节变量符号表增加 VAR_STATIC 与 VAR_STATIC_RETAIN 类别的说明，第 6.1.3 节增加函数可以使用的数据类型与指针 (Pointer) 及参考 (Reference) 数据类型说明，第 6.2.6 节停电保持符号初始值生效说明。 7.增加第 7 章函数的相关说明。 8.增加第 8 章联合 (UNION) 相关说明，修改结构、列举章节的图片及相关说明。 9.增加第 9 章轴支持列表，增加伺服参数备份还原章节，增加自动增益调整章节。 10.增加第 10.2.4 节在 DVPxxMC 与 AS5xx 机种的功能块引脚支持 ST 表达式的写法。 11.修改第 15.2.10 变更程序的执行顺序说明 12.修改第 17.1.1 变更 PLC 机种支持列表内容，修改程序上下载图片，变更查找替换章节说明内容与图片 13.增加第 21.3 节 CARD Utility 备份还原支持列表。 14.增加第 23.5 节 NTC 向导介绍。 15.增加附录 B 章的 AS5xx 相关说明。 	2020/7/10

版本	变更内容	发行日期
第十二版	<ol style="list-style-type: none"> 1.在各章节修改 AH 的旧 HWCONFIG3.0 相关图片及修改建立 POU 时密码字段的相关图片。 2.增加第 1.1.3 节安装 ISPSOft 时的流程说明，包含安装 USB 驱动、安装 HWCONFIG 及 Visual C++等。 3.变更第 1.3.1 节 ISPSOft 语言支持列表内容。 4.变更第 1.3.2 节 ISPSOft 功能支持列表内容。 5.增加第 2.4.3 节 COMMGR 通讯类别为 RS232/422/485 时，通讯协议多一个虚拟 COM 的字段可以使用。 6.增加 3.1 节 HWCONFIG 4.0 支持 AH 机种;增加 EtherNetIP 功能说明。 7.增加第 3.1.6 节 EtherNetIP 功能说明。 8.增加第 3.1.7 节 EtherCAT 功能说明。 9.增加第 5.4.9 节 POU 的目录管理。 10. 增加第 6.2.3 LD、CFC 数组索引使用变量。 11. 修改第 7.5.1 新增用户自定义函式库图片。 12. 增加第 8.1 节各机种的结构与联合所支持的数据型态表。 13. 修改第 9.1 节轴支持相关功能机种参考表格、增加 DVP50MC-04S、DV50MC-16S、AS532EST、AS564EST。 14. 修改第 17.1.3 项目上下载表支持 OPCUA。 15. 增加第 18.1.2 节 LD 监控数值显示位置切换。 16. 修改第 18.1.6 节在线编辑流程，暂时储存密码。 17. 修改第 19.1 节密码说明表格、增加各机种密码长度。 18. 增加第 21.3 节 CARD Utility 支持 AS5xx 及 DVP 机种。 19. 增加第 22.2.3 节 E-CAM 主从轴设定说明表格。 20. 增加第 23.2.6 节 DIA 示波器功能说明。 	2021/02/22

MEMO

ISPSoft 使用手册

目录

前言	I
第 1 章 软件简介	
1.1 ISPSoft 简介及系统需求	1-2
1.1.1 特色简介	1-2
1.1.2 系统需求	1-3
1.1.3 安装 ISPSoft	1-4
1.1.4 删除 ISPSoft	1-11
1.2 通讯管理员 COMMGR 简介	1-13
1.2.1 COMMGR 的工作模式	1-13
1.2.2 安装 COMMGR	1-14
1.2.3 删除 COMMGR	1-17
1.3 ISPSoft PLC 机种支持	1-18
1.3.1 ISPSoft 语言支持列表	1-18
1.3.2 ISPSoft 功能支持列表	1-18
第 2 章 启动与设定	
2.1 开始导览与环境介绍	2-3
2.1.1 进入 ISPSoft 的第一步	2-3
2.1.2 窗口标题与状态栏	2-7
2.1.3 功能工具栏	2-8
2.1.4 图标工具栏	2-11
2.1.5 项目管理区与信息显示区	2-12
2.1.6 编辑工作区	2-14
2.2 ISPSoft 的项目架构	2-17
2.2.1 单机项目与群组项目	2-17
2.2.2 整合运动模块项目	2-18
2.2.3 ISPSoft 的单机项目管理	2-19
2.2.4 ISPSoft 的群组项目管理	2-25
2.2.5 ISPSoft 的运动模块项目管理	2-32

2.3	ISPSOft 中的基本设定	2-37
2.3.1	系统及环境设定	2-37
2.3.2	用户编辑格式的导入与导出	2-43
2.3.3	导出程序到文字格式	2-44
2.4	通讯设定	2-45
2.4.1	通讯管理员 COMMGR 的启动与关闭	2-46
2.4.2	COMMGR 的驱动管理	2-47
2.4.3	建立联机通道 - 驱动的建立	2-48
2.4.3.1	RS232/422/485 通讯参数设置	2-50
2.4.3.2	USB (Virtual COM) 通讯参数设置	2-51
2.4.3.3	DirectLink (USB) 通讯参数设置	2-51
2.4.3.4	Ethernet 通讯参数设置	2-52
2.4.3.5	DirectLink (Ethernet) 通讯参数设置	2-53
2.4.3.6	DVP Simulator 通讯参数设置	2-54
2.4.3.7	AH5x0 与 AH5x1 Simulator 通讯参数设置	2-54
2.4.3.8	AS Simulator 通讯参数设置	2-55
2.4.4	建立联机通道 - 驱动的启动与停止	2-55
2.4.5	建立联机通道 - 驱动的修改与删除	2-57
2.4.6	建立联机通道 - 仿真器的面板操作	2-58
2.4.7	建立 ISPSOft 与 COMMGR 之间的连结	2-61
2.4.8	连接主机与通讯端口	2-63
2.4.9	实际联机测试	2-66
第 3 章 PLC 系统的配置与设定		
3.1	硬件规划工具 - HWCONFIG 4.0	3-5
3.1.1	进入 HWCONFIG 的第一步	3-6
3.1.2	HWCONFIG 的环境介绍	3-10
3.1.2.1	视窗标题与状态栏	3-10
3.1.2.2	菜单工具列-文件	3-11
3.1.2.3	菜单工具列-编辑	3-18
3.1.2.4	菜单工具列-检视	3-29
3.1.2.5	菜单工具列-通讯	3-29
3.1.2.6	菜单工具列-工具	3-39

3.1.2.7	菜单工具列-帮助	3-48
3.1.2.8	菜单工具列-常用	3-49
3.1.2.9	项目树	3-50
3.1.2.10	产品列表	3-52
3.1.2.11	输出视窗	3-53
3.1.2.12	硬件配置区	3-54
3.1.2.13	编辑区	3-55
3.1.3	进阶功能操作	3-57
3.1.3.1	新增模块/扩展背板	3-57
3.1.3.2	更换模块/背板	3-65
3.1.3.3	模块/扩展背板的移动	3-67
3.1.3.4	备注编辑	3-69
3.1.3.5	硬件配置区-重设配置	3-71
3.1.3.6	硬件配置区-显示模块详细信息	3-72
3.1.3.7	硬件配置区-输出/输入装置重新排序	3-73
3.1.3.8	硬件配置区-模块的地址配置	3-73
3.1.3.9	硬件配置区-更改远程装置地址	3-74
3.1.3.10	硬件配置区-重置大小	3-75
3.1.3.11	编辑区-模块内部参数的导出与导入	3-76
3.1.3.12	编辑区-模块内部参数的更新	3-77
3.1.3.13	显示/隐藏功能	3-77
3.1.3.14	信息检视与模块诊断	3-78
3.1.3.15	在线变更模块状态	3-79
3.1.3.16	远程模块配置	3-81
3.1.3.17	打开外部软件	3-83
3.1.3.18	IO-Link	3-84
3.1.3.19	变量设定	3-102
3.1.3.20	冗余功能	3-107
3.1.4	PLC 主机参数设定	3-108
3.1.4.1	系统设定-系统资讯	3-108
3.1.4.2	系统设定-系统参数	3-109
3.1.4.3	系统设定-装置元件范围设定	3-113
3.1.4.4	系统设定-输入点滤波时间	3-114

3.1.4.5	系统设定-定位轴参数	3-115
3.1.4.6	系统设定-日光节约时间.....	3-116
3.1.4.7	COM 通讯端口设定.....	3-117
3.1.4.8	以太网基本设定	3-118
3.1.4.9	以太网进阶设定-IP 过滤器	3-119
3.1.4.10	以太网进阶设定-网络时间校正.....	3-120
3.1.4.11	以太网进阶设定-邮件.....	3-121
3.1.4.12	以太网进阶设定-通讯接口	3-126
3.1.4.13	以太网进阶设定-RTU-EN01 对应.....	3-128
3.1.4.14	功能卡 1 设定	3-129
3.1.4.15	功能卡 2 设定	3-130
3.1.4.16	台达设备参数还原功能	3-134
3.1.4.17	数据交换.....	3-138
3.1.5	EtherNet/IP	3-141
3.1.5.1	开启 EIP.....	3-141
3.1.5.2	上载 EtherNet/IP	3-143
3.1.5.3	下载 EtherNet/IP	3-144
3.1.5.4	扫描从站	3-145
3.1.5.5	加入从站	3-146
3.1.5.6	设定主站\从站 IP	3-148
3.1.5.7	建立联机	3-149
3.1.5.8	删除联机	3-152
3.1.5.9	设定 RTU 模块配置.....	3-153
3.1.5.10	清空从站.....	3-154
3.1.5.11	显示 RTU 信息.....	3-155
3.1.5.12	显示数据交换	3-155
3.1.6	EtherCAT	3-156
3.1.6.1	开启 EtherCAT	3-157
3.1.6.2	上载 EtherCAT	3-158
3.1.6.3	下载 EtherCAT	3-159
3.1.6.4	扫描 EtherCAT 从站	3-160
3.1.6.5	EtherCAT 网络拓扑图操作.....	3-160
3.1.6.6	设定从站	3-167

3.1.6.7 设定主站	3-177
3.2 硬件规划工具 - HWCONFIG 3.0	3-179
3.2.1 HWCONFIG 的环境介绍	3-179
3.2.2 主机的版本管理	3-180
3.3 DVP 系列机种的参数设定	3-180
3.3.1 DVP 主机的系统管理工具	3-180
3.3.2 停电保持设定	3-181
3.3.3 联机信息	3-185
3.3.4 DVP-ES3 机种的主机参数设置	3-186
3.4 DVPxxMC 机种的 PLC 主机参数设定	3-188
3.4.1 开启 PLC 主机的参数设定页面	3-188
3.4.2 一般设定	3-189
3.4.2.1 一般设定-系统信息	3-189
3.4.2.2 一般设定-CANopen	3-190
3.4.2.3 一般设定-CANMotion	3-191
3.4.2.4 一般设定-RS232	3-191
3.4.2.5 一般设定- RS485	3-192
3.4.2.6 一般设定- 增量型编码器 1	3-192
3.4.2.7 一般设定- 增量型编码器 2	3-193
3.4.2.8 一般设定- 绝对型编码器	3-194
3.4.2.9 一般设定- 输入点过滤时间	3-195
3.4.2.10 一般设定- 运转 / 停止设定	3-196
3.4.2.11 一般设定- 以太网基本设定	3-197
3.4.3 上下载	3-198
3.5 万年历设定	3-199
3.6 PLC 内存设定 (清除程序&恢复出厂值)	3-200
第 4 章 快速入门	
4.1 快速入门	4-2
4.1.1 范例功能说明	4-2
4.1.2 硬件规划	4-2
4.1.3 程序规划	4-3
4.2 ISPSOft 的项目开发程序	4-3

4.3	新增项目	4-4
4.4	硬件组态	4-5
4.4.1	模块配置	4-5
4.4.2	主机与模块参数设置	4-7
4.5	建立程序	4-10
4.5.1	新增梯形图程序	4-10
4.5.2	基本编辑 - 建立接点与线圈	4-12
4.5.3	基本编辑 - 梯形图区段的新增/插入与指令编辑模式	4-15
4.5.4	基本编辑 - 梯形图区段的选取与操作	4-17
4.5.5	基本编辑 - 并联接点装置	4-20
4.5.6	基本编辑 - 编辑批注	4-21
4.5.7	基本编辑 - 插入 API 应用指令	4-22
4.5.8	基本编辑 - 比较接点的建立与常数的输入格式	4-24
4.5.9	完成范例程序	4-25
4.5.10	程序的检查与编译	4-26
4.6	测试与除错	4-27
4.6.1	建立联机	4-27
4.6.2	下载程序与组态参数	4-30
4.6.3	联机测试	4-33
第 5 章 POU 与 TASK		
5.1	关于程序组织单元 (POU) 与工作 (TASK)	5-2
5.2	认识程序组织单元 (POU)	5-3
5.2.1	POU 的编程架构与种类	5-3
5.2.2	ISPSOft 的 POU 架构	5-4
5.3	认识工作 (TASK)	5-6
5.3.1	ISPSOft 中的工作 (TASK) 管理	5-6
5.3.2	项目管理区的工作 (TASK) 项目	5-10
5.3.3	工作 (TASK) 中的 POU 执行顺序	5-13
5.4	POU 的管理操作	5-15
5.4.1	POU 的建立与属性设置	5-15
5.4.2	POU 的启动状态	5-20
5.4.3	POU 的删除与复制	5-21

5.4.4	POU 密码的设置与解除.....	5-23
5.4.5	导出程序 POU.....	5-24
5.4.6	导入程序 POU.....	5-25
5.4.7	导出功能块 POU.....	5-26
5.4.8	导入功能块 POU.....	5-27
5.4.9	POU 的目录管理.....	5-28
5.5	工作 (TASK) 管理的操作.....	5-29
5.5.1	工作 (TASK) 属性与中断条件的设置.....	5-31
5.5.2	POU 的配置.....	5-36
5.5.3	POU 的排序.....	5-38
5.6	范例实作.....	5-39
5.6.1	POU 与 TASK 的编程范例.....	5-39
5.6.2	DVP 机种的中断程序范例.....	5-43
5.6.3	AH5x0 机种的中断程序范例.....	5-48
第 6 章 变量符号		
6.1	认识变量符号.....	6-2
6.1.1	变量符号的作用范围与命名原则.....	6-2
6.1.2	变量符号的类型.....	6-3
6.1.3	变量符号的数据类型.....	6-4
6.1.4	变量符号的地址配置与初始值.....	6-11
6.1.5	变量符号的变址寄存器修饰.....	6-13
6.1.6	变量符号的位操作 (限定 AH/AS 机种).....	6-15
6.2	ISPSOFT 中的变量符号管理.....	6-17
6.2.1	变量符号表.....	6-17
6.2.2	新增变量符号.....	6-20
6.2.3	数组与字符串类型符号的使用原则.....	6-26
6.2.4	变量符号的修改与符号表编辑.....	6-29
6.2.5	清除变量符号的配置地址.....	6-31
6.2.6	下载变量符号的初始值.....	6-32
6.2.7	符号表的导出.....	6-34
6.2.8	符号表的导入.....	6-35
6.2.9	符号表的排序.....	6-38

6.2.10	变量符号的筛选	6-40
6.2.11	装置资源分配设置	6-41
6.3	范例实作	6-42
6.3.1	变量符号规划	6-42
6.3.2	建立范例程序	6-43
第 7 章	功能块与函数库	
7.1	认识功能块 (FUNCTION Block)	7-2
7.1.1	功能块简介	7-2
7.1.2	功能块的特性与优点	7-3
7.2	ISPSOFT 的功能块 (Function Block) 架构	7-4
7.2.1	功能块的使能 (En) 端点	7-4
7.2.2	功能块的变量符号	7-5
7.2.3	功能块的输出与输入	7-6
7.2.4	指针类型的变量符号	7-12
7.2.5	功能块的定义与实例	7-15
7.2.6	功能块调用功能块的嵌套结构	7-18
7.2.7	功能块的内存配置	7-20
7.3	功能块的使用	7-27
7.3.1	功能块的基本规格	7-27
7.3.2	功能块专用的脉冲型指令 (限定 AH/AS 机种)	7-28
7.3.3	功能块的程序监控	7-29
7.3.4	功能块的修改	7-30
7.4	范例实作	7-31
7.4.1	程序规划	7-31
7.4.2	建立范例程序	7-31
7.4.3	函数 (Function) 范例程序	7-41
7.5	认识函数库 (LIBRARY)	7-42
7.5.1	新增用户自定义函数库	7-42
7.5.2	引用用户自定义函数库	7-44
7.5.3	引用台达函数库	7-46

第 8 章 自定义数据类型

8.1 自定义数据类型	8-2
8.2 结构 (STRUCTURE)	8-3
8.2.1 结构 (STRUCTURE) 的意义	8-3
8.2.2 建立结构 (STRUCTURE)	8-3
8.2.3 使用结构 (STRUCTURE) 类型变量符号	8-5
8.2.4 结构 (STRUCTURE) 的常见用法	8-6
8.2.5 结构 (STRUCTURE) 改变为枚举 (ENUMERATION)	8-9
8.3 枚举 (ENUMERATION)	8-10
8.3.1 枚举 (ENUMERATION) 的意义	8-10
8.3.2 建立枚举 (ENUMERATION)	8-10
8.3.3 使用枚举 (ENUMERATION) 类型变量符号	8-12
8.3.4 枚举 (ENUMERATION) 的常见用法	8-13
8.3.5 枚举 (ENUMERATION) 改变为结构 (STRUCTURE)	8-15
8.4 联合 (UNION)	8-16
8.4.1 联合 (UNION) 的意义	8-16
8.4.2 建立联合 (UNION)	8-16
8.4.3 使用联合 (UNION) 类型变量符号	8-18
8.4.4 联合 (UNION) 的常见用法	8-20
8.4.5 内建联合的支持与名称	8-22

第 9 章 轴

9.1 轴	9-2
9.1.1 轴的意义	9-2
9.1.2 建立轴	9-2
9.1.2.1 AHxxEMS 机种的建立轴	9-2
9.1.2.2 DVPxxMC/AS5xx 机种建立轴与轴基本设定	9-5
9.1.3 全局符号表的主符号表与轴符号表	9-9
9.2 轴的参数设定、监控与调试	9-10
9.2.1 轴参数设定	9-10
9.2.1.1 AHxxEMC 机种的轴参数设定	9-10
9.2.1.2 DVPXXMC/AS5xx 机种的轴参数设定	9-13
9.2.2 监控与调试	9-20

9.2.2.1 AHxxEMC 机种的监控与调测	9-20
9.2.2.2 DVPxxMC/AS5xx 机种的监控与调测	9-25
9.2.3 伺服参数备份还原	9-33
9.2.4 自动增益调整	9-35
第 10 章 梯形图 LD	
10.1 梯形图 (LD) 简介	10-2
10.1.1 ISPSoft 的 LD 编辑环境	10-2
10.1.2 梯形图区段 (NETWORK)	10-4
10.1.3 选取对象	10-5
10.2 ISPSoft 中建立 LD 程序	10-6
10.2.1 建立接点 (Contact)、线圈 (Coil) 及 MPS	10-6
10.2.1.1 插入接点 (Contact) 与变更接点类型	10-6
10.2.1.2 插入线圈 (Coil) 与变更线圈类型	10-8
10.2.1.3 新增 MPS	10-9
10.2.2 LD 中使用装置、变量符号及常数	10-10
10.2.3 指令编辑模式	10-11
10.2.4 插入 API 应用指令与功能块	10-12
10.2.5 编辑比较接点	10-15
10.2.6 编辑区块逻辑指令	10-17
10.2.7 编辑多重输出	10-20
10.2.8 设置标签 (Label)	10-21
10.2.9 编辑批注	10-22
10.2.10 符号模式与地址模式的显示切换	10-24
10.2.11 书签功能	10-25
10.2.12 梯形图区段的使能与失效	10-26
第 11 章 功能块图 FBD	
11.1 功能块图 FBD 简介	11-2
11.1.1 关于功能块图 (FBD)	11-2
11.1.2 ISPSoft 的 FBD 编辑环境	11-2
11.1.3 功能块图区段 (NETWORK)	11-4
11.1.4 选取对象	11-5

11.2	ISPSOft 中建立 FBD 程序.....	11-6
11.2.1	输入节点与输出节点.....	11-7
11.2.2	FBD 中使用装置、变量符号及常数.....	11-10
11.2.3	AND 与 OR 方块.....	11-11
11.2.4	反相逻辑.....	11-14
11.2.5	上沿触发输入与下沿触发输入.....	11-16
11.2.6	置位 (SET) 输出与复位 (RESET) 输出.....	11-18
11.2.7	API 应用指令、比较接点与功能块.....	11-20
11.2.8	置位标签 (Label).....	11-24
11.2.9	批注与提示.....	11-25
11.2.10	符号模式与地址模式的显示切换.....	11-27
11.2.11	书签功能.....	11-28
11.2.12	功能块图区段的使能与失效.....	11-29
第 12 章	指令列表 IL	
12.1	指令列表 (IL) 简介.....	12-2
12.1.1	IL 程序的结构.....	12-2
12.1.2	IL 的功能块 (FB) 调用指令.....	12-4
12.1.3	IL 编程的注意事项.....	12-5
12.2	于 ISPSOft 中建立 IL 程序.....	12-6
12.2.1	ISPSOft 的 IL 编辑环境.....	12-6
12.2.2	编辑 IL 指令.....	12-7
12.2.3	插入 API 应用指令与功能块.....	12-8
12.2.4	书签功能.....	12-10
12.2.4.1	加入书签.....	12-10
12.2.4.2	取消书签.....	12-11
12.2.4.3	移至书签位置.....	12-12
第 13 章	结构化语言 ST	
13.1	认识结构化语言 ST.....	13-3
13.1.1	ST 程序的基本结构.....	13-3
13.1.2	叙述句 (Statement).....	13-4
13.1.3	表达式 (Expression).....	13-5

13.1.4	操作数 (Operand) 与运算符 (Operator)	13-6
13.1.5	关键词 (Keyword) 与批注 (Comment)	13-7
13.1.6	ST 程序中使用数组符号	13-9
13.1.7	ST 编程的注意事项	13-10
13.2	ISPSOft 中的 ST 叙述句结构	13-11
13.2.1	指定结构 :=	13-11
13.2.2	条件结构 IF	13-13
13.2.3	条件结构 CASE	13-15
13.2.4	循环结构 REPEAT	13-17
13.2.5	循环结构 WHILE	13-18
13.2.6	循环结构 FOR	13-19
13.2.7	API 应用指令结构	13-20
13.2.8	功能块 (FB) 调用结构	13-21
13.2.9	空白叙述	13-23
13.2.10	RETURN 叙述	13-24
13.2.11	EXIT 叙述	13-24
13.3	ISPSOft 中建立 ST 程序	13-26
13.3.1	ISPSOft 的 ST 编辑环境	13-26
13.3.2	编辑 ST 叙述	13-27
13.3.3	插入 API 应用指令与功能块	13-29
13.3.4	书签功能	13-30
13.4	ST 的程序范例	13-31
13.4.1	范例说明	13-31
13.4.2	硬件规划	13-31
13.4.3	程序规划	13-31
13.4.4	建立范例程序	13-32
第 14 章 循序功能图 SFC		
14.1	认识循序功能图 SFC	14-3
14.1.1	SFC 程序架构	14-3
14.1.2	SFC 动作原理	14-4
14.2	ISPSOft 中的 SFC 架构	14-5
14.2.1	步进点与 Action	14-5

14.2.2 转换点 (Transition)	14-8
14.2.3 共进分歧与选择分歧	14-10
14.2.4 共进合流与选择合流	14-11
14.2.5 跳跃点	14-12
14.2.6 Actions 的修饰条件	14-14
14.2.6.1 条件定义	14-15
14.2.6.2 设定修饰条件的注意事项	14-19
14.2.7 初始步进点	14-20
14.2.8 内部属性	14-21
14.2.8.1 Step 属性参数	14-21
14.2.8.2 Action 与 Transition 属性参数	14-25
14.3 于 ISPSOft 中建立 SFC 程序	14-32
14.3.1 ISPSOft 的 SFC 编辑环境	14-32
14.3.2 Actions 与 Transitions 的建立与管理	14-34
14.3.3 新增步进点结构	14-36
14.3.4 插入并联转换点	14-37
14.3.5 插入并联步进点	14-38
14.3.6 插入步进点结构 - 共进分歧&选择合流	14-39
14.3.7 插入步进点结构 - 选择分歧&共进合流	14-40
14.3.8 插入跳跃点	14-41
14.3.9 指定步进点与转换点符号	14-42
14.3.10 指定初始步进点	14-42
14.3.11 配置 Actions/折迭 Action 列表	14-43
14.4 SFC 的程序范例	14-46
14.4.1 范例说明	14-46
14.4.2 硬件规划	14-47
14.4.3 程序规划	14-47
14.4.4 建立范例程序	14-48
 第 15 章 连续功能图 CFC	
15.1 连续功能图 (CFC) 简介	15-2
15.1.1 关于连续功能图	15-2
15.1.2 CFC 编程的注意事项	15-2

15.1.3	ISPSoft 的 CFC 编辑环境	15-2
15.1.4	CFC 的图示工具栏	15-4
15.1.5	CFC 编辑工作区的快捷选单	15-5
15.2	ISPSoft 中建立 CFC 程序	15-6
15.2.1	选取对象	15-6
15.2.2	输入节点、输出节点与逻辑闸	15-6
15.2.2.1	插入节点或门	15-7
15.2.2.2	新增与删除接脚	15-8
15.2.3	改变接脚的形式	15-10
15.2.4	物件的联机与取消	15-10
15.2.5	指令与功能块	15-11
15.2.5.1	插入指令	15-12
15.2.5.2	插入功能块	15-13
15.2.6	删除对象	15-15
15.2.7	编辑装置或符号	15-16
15.2.8	物件的致能与失效	15-17
15.2.9	插入批注	15-17
15.2.10	变更程序的执行顺序	15-20
15.2.11	显示设备批注	15-20
第 16 章 C 语言		
16.1	C 语言编辑 (C Language)	16-2
16.1.1	C 语言之架构说明	16-2
16.1.2	C 语言编辑环境说明	16-2
16.1.3	C 语言编译与除错	16-9
16.1.4	内部函数库指令说明	16-10
16.1.5	书签与注解	16-11
16.1.6	范例操作说明	16-13
第 17 章 编辑辅助工具		
17.1	ISPSoft 的编辑工具与辅助功能	17-2
17.1.1	变更 PLC 机种	17-2
17.1.2	将 DVP 机种转换为 AS 机种	17-3

17.1.3	项目的下载与上载.....	17-14
17.1.4	LD、FBD 与 CFC 编辑环境的查找替换与定位到功能.....	17-16
17.1.5	IL 与 ST 编辑环境的查找替换功能.....	17-19
17.1.6	SFC 编辑环境的查找替换功能.....	17-22
17.1.7	符号表的查找功能.....	17-24
17.1.8	项目查找与替换.....	17-26
17.1.9	打印功能.....	17-27
17.2	装置与寄存器管理.....	17-30
17.2.1	DVP 机种的装置注解表.....	17-30
17.2.2	AH/AS 系列机种的装置注解表.....	17-34
17.2.3	装置使用状况.....	17-41
17.2.4	寄存器编辑.....	17-44
17.2.5	装置状态编辑.....	17-51
17.2.6	DVP 机种的文件寄存器编辑.....	17-56
17.3	步序定位.....	17-63
17.3.1	使用步序跳转.....	17-63
17.4	程序比对.....	17-64
17.4.1	程序比对简介.....	17-64
17.4.2	与文件比对.....	17-65
17.4.3	与 PLC 比对.....	17-67
 第 18 章 测试与除错工具		
18.1	在线监控.....	18-2
18.1.1	切换主机的运转/停止状态.....	18-2
18.1.2	在线监控的功能与环境介绍.....	18-2
18.1.3	在线模式下变更 X 装置的状态.....	18-8
18.1.4	程序监控的在线操作.....	18-9
18.1.5	装置监控表.....	18-15
18.1.6	在线编辑与在线更新.....	18-21
18.2	DVP 机种的除错模式.....	18-24
18.2.1	启动 DVP 的除错模式.....	18-24
18.2.2	设定与清除断点.....	18-25
18.2.3	除错模式下的程序执行.....	18-25

18.3	AH/AS 机种的除错模式.....	18-27
18.3.1	进入除错模式.....	18-27
18.3.2	断点操作.....	18-27
18.3.3	连续执行.....	18-28
18.3.4	单步执行.....	18-29
18.4	确认 PLC 状态.....	18-30
18.4.1	PLC 系统信息.....	18-30
18.4.2	PLC 系统记录.....	18-32
18.5	3D 坐标图.....	18-39
18.5.1	功能说明.....	18-39
18.5.2	建立 3D 坐标图.....	18-39
18.5.3	加入曲线.....	18-41
18.5.4	显示 3D 坐标图.....	18-43
第 19 章 密码管理与数据保护机制		
19.1	ISPSOft 的密码架构.....	19-2
19.2	程序识别码与 PLC 识别码.....	19-3
19.2.1	程序识别码的设定与解除.....	19-3
19.2.2	PLC 识别码的设置与解除.....	19-4
19.3	项目密码与 PLC 密码.....	19-5
19.3.1	项目密码的设置与解除.....	19-5
19.3.2	PLC 密码的设置与解除.....	19-7
19.3.3	项目密码与 PLC 密码同步.....	19-8
19.4	POU 密码.....	19-9
19.5	子程序密码.....	19-10
19.5.1	子程序密码简介.....	19-10
19.5.2	子程序密码的设定与解除.....	19-11
19.6	其它的密码与数据保护功能.....	19-11
19.6.1	程序上载失效.....	19-11
19.6.2	只读装置区域设定.....	19-12
19.6.3	TC-01 密码钥匙设定.....	19-13

第 20 章 网络规划与数据交换

20.1	网络规划工具 - NWCONFIG	20-3
20.1.1	NWCONFIG 简介.....	20-3
20.1.2	相关基础介绍	20-4
20.1.3	NWCONFIG 的通讯设定	20-6
20.1.3.1	NWCONFIG 中的联机机制.....	20-6
20.1.3.2	设定通讯参数	20-8
20.1.4	工作流程.....	20-9
20.2	网络架构的部署	20-13
20.2.1	部署节点.....	20-13
20.2.2	连结网络.....	20-17
20.2.3	调整或删除已部署的装置及网络	20-21
20.2.4	节点与网络属性的设定	20-24
20.2.5	装置或网络的隐藏/显示	20-28
20.2.6	合法的网络架构	20-31
20.2.7	下载绕送路径表 (Routing Table)	20-34
20.2.8	绕送路径测试	20-36
20.3	PLC Link 的建构	20-38
20.3.1	打开 PLC Link 的设定窗口	20-38
20.3.2	选择主站装置 (步骤 1)	20-40
20.3.3	设定通讯参数 (步骤 2)	20-41
20.3.4	设定数据交换表 (步骤 3)	20-43
20.3.4.1	数据交换表简介	20-43
20.3.4.2	设定区块.....	20-45
20.3.4.3	数据交换表的机种同步性	20-47
20.3.4.4	数据交换表的管理操作	20-49
20.3.5	PLC Link 的监控功能	20-50
20.3.6	设定 PLC Link 的注意事项	20-54
20.4	Ether Link 的建构.....	20-55
20.4.1	Ether Link 简介	20-55
20.4.2	开启 Ether Link 的设置窗口.....	20-56
20.4.3	数据交换表的建立与管理	20-58
20.4.4	装置清单与图标区的操作	20-61

20.4.5	Ether Link 的运作模式	20-63
20.4.6	下载 Ether Link 设定	20-65
20.4.7	上传 Ether Link 设定	20-67
20.4.8	删除异步装置	20-69
20.4.9	启动/关闭在线监控模式	20-70
20.4.9.1	启动监控	20-70
20.4.9.2	监控状态显示	20-72
20.4.9.3	关闭监控	20-72
20.4.10	在线启动/停止 Ether Link (标志模式)	20-74
20.4.10.1	启动 Ether Link	20-74
20.4.10.2	停止 Ether Link	20-75
20.4.11	监控表与错误记录	20-77
20.5	NWCONFIG 的管理与应用	20-78
20.5.1	保存与打印	20-78
20.5.2	整体下载	20-79
20.5.2.1	下载参数说明	20-79
20.5.2.2	下载操作说明	20-79
20.5.3	ISPSOft 中的绕送应用	20-81
 第 21 章 数据的备份与还原		
21.1	数据备份存储卡辅助工具	21-2
21.1.1	数据备份存储卡简介	21-2
21.1.2	数据备份存储卡的操作	21-3
21.2	PLC 永久备份设定	21-3
21.3	存储卡工具 - CARD UTILITY (AH/AS 机种限定)	21-4
21.3.1	软件介绍	21-6
21.3.2	备份操作	21-8
21.3.3	还原操作	21-15
21.3.4	命令列模式指令操作说明	21-19
 第 22 章 G-Code 编辑器与 E-CAM 编辑器		
22.1	G-Code 编辑器	22-2
22.1.1	关于 G-Code	22-2

22.1.2	使用 G-Code 编辑器	22-2
22.1.3	G-Code 的指令结构	22-7
22.1.4	G-Code 功能详细介绍	22-9
22.1.4.1	G0 (快速定位)	22-9
22.1.4.2	G1 (直线插补)	22-11
22.1.4.3	G2 (顺时针圆弧插补/螺旋插补)	22-13
22.1.4.4	G3 (逆时针圆弧插补/螺旋插补)	22-16
22.1.4.5	G4 (时间延时)	22-19
22.1.4.6	G17/G18/G19 (平面选择)	22-20
22.1.4.7	G50 (准确停止)	22-21
22.1.4.8	G51 (圆弧交接)	22-22
22.1.4.9	G52 (圆滑交接)	22-23
22.1.4.10	G90 (绝对模式)	22-25
22.1.4.11	G91 (相对模式)	22-26
22.1.4.12	M 码	22-27
22.1.5	G-Code 范例-三轴同动绘出 Delta LOGO	22-29
22.2	E-CAM 编辑器	22-30
22.2.1	关于 E-CAM (电子凸轮)	22-30
22.2.2	电子凸轮意义说明	22-30
22.2.3	使用 E-CAM 编辑器	22-31

第 23 章 向导工具

23.1	定位规划表	23-3
23.1.1	关于定位规划表	23-3
23.1.2	设定定位规划表	23-3
23.1.3	定位规划表的仿真	23-7
23.1.4	下载与上载定位规划表	23-9
23.1.5	轴控模式-单轴点对点模式	23-10
23.1.6	轴控模式-单轴多段输出模式	23-10
23.1.7	轴控模式-双轴直线插补模式	23-11
23.1.8	轴控模式-双轴圆弧插补模式	23-12
23.2	数据示波器	23-14
23.2.1	关于数据示波器	23-14

23.2.2	开启数据示波器	23-14
23.2.3	设定取样参数	23-15
23.2.4	取样模式	23-18
23.2.5	量测	23-20
23.2.6	DIA 示波器	23-24
23.3	数据记录器	23-34
23.3.1	关于数据记录器	23-34
23.3.2	开启数据记录器	23-34
23.3.3	设定取样参数	23-35
23.3.4	监看与录制模式	23-37
23.3.5	量测	23-40
23.4	高速计数器向导	23-44
23.4.1	关于高速计数器向导	23-44
23.4.2	使用高速计数器向导	23-44
23.5	NTC 向导	23-46
23.5.1	关于 NTC 向导	23-46
23.5.2	使用 NTC 向导	23-46
23.6	运动参数模拟	23-48
23.6.1	关于运动参数模拟	23-48

附录 A USB 联机

A.1	安装 PLC 的 USB 驱动程序	A-2
A.1.1	在 Windows XP SP3 环境下安装 USB 驱动程序	A-2
A.1.2	在 Windows 7 环境下安装 USB 驱动程序	A-7
A.1.3	在 Windows 8.1 环境下安装 USB 驱动程序	A-11
A.1.4	在 Windows 10 环境下安装 USB 驱动程序	A-15
A.2	于 COMMGR 中建立 USB 类型的 Driver	A-18
A.3	DVP-SX2 系列的 USB 设定	A-20

附录 B 各系列机种的相关注意事项

B.1	各系列机种的装置寻址	B-2
B.1.1	AH/AS 系列机种的装置种类	B-2
B.1.2	AH/AS 系列机种 X、Y、D、L 装置的寻址格式	B-3

B.1.3	AH/AS 系列机种 X/Y 接点状态的实时存取.....	B-4
B.1.4	DVPxxMC/AS5xx 机种的装置种类	B-5
B.1.5	DVPxxMC/AS5xx 机种装置的寻址格式.....	B-5
B.1.6	DVPxxMC/AS5xx 机种的停电保持装置.....	B-6
B.2	各系列机种的装置资源	B-6
B.2.1	AHCPU500-EN/AHCPU500-RS2 机种	B-6
B.2.2	AHCPU510-EN/AHCPU510-RS2 机种	B-7
B.2.3	AHCPU520-EN/AHCPU520-RS2 机种	B-7
B.2.4	AHCPU530-EN/AHCPU530-RS2 机种	B-8
B.2.5	AHCPU501-EN/AHCPU501-RS2 机种	B-8
B.2.6	AHCPU511-EN/AHCPU511-RS2 机种	B-9
B.2.7	AHCPU521-EN/AHCPU521-RS2 机种	B-9
B.2.8	AHCPU531-EN/AHCPU531-RS2 机种	B-10
B.2.9	AHCPU560-EN2 机种.....	B-10
B.2.10	AHxxEMC 系列机种.....	B-11
B.2.11	AS200/AS300 系列机种.....	B-11
B.2.12	DVPxxMC/AS5xx 系列机种.....	B-12
B.3	编译与上下载时间.....	B-12
附录 C 打印管理工具		
C.1	环境简介.....	C-2
C.2	操作功能说明	C-3

MEMO

前言

ISPSOft 为台达新一代的可编程控制器 (PLC) 开发工具，除了导入 IEC 61131-3 的编程架构外，更以多任务整合的方式来进行项目的管理，无论是单纯的小型应用，或是较为复杂的中大型控制系统，ISPSOft 都能提供用户一个高效且便利的开发环境。

关于 IEC 61131-3

早期当各家 PLC 开发商纷纷建立自己的编程语言时，其各自不同的编程方式，对于用户而言无疑是一个相当大的负担；有鉴于此，由国际组织 I.E.C. (International Electro-technical Commission) 所制定与推动的 IEC 61131-3 标准便由此应运而生。其主要的特色在于统整各家的编程特点，并考虑用户的背景与习惯，而规范了一个可让各家厂商与用户共同依循的标准，并可让符合 IEC61131-3 标准的程序代码具备相当高的可读性与兼容性。以下是关于 IEC 61131-3 的简述。

- 整合了各家的编程方式及用户习惯，提供了 7 种编程语言，包括包括**梯形图 (LD)**、**顺序功能图 (SFC)**、**功能区块图 (FBD)**、**指令列表 (IL)**、**结构化语言 (ST)** 以及**连续功能图 (CFC)** 与 **C 语言 (C)**。
- 使用变量的观念，让用户可自行定义变量符号来取代 PLC 的装置名称，除了大大地提高程序的可读性之外，更省去了用户费心配置装置地址的时间。
- 使用**程序组织单元 (POU)** 的架构，其特色在于将传统单一的程序结构，切割成若干个可独立开发的程序单元，并透过调用**函数 (Function)** 与**功能块 (Function Block)** 的方式，让程序架构更加多元化、模块化，易于用户使用及维护。
- 使用**工作 (Task)** 的观念来对各个开发的程序单元进行管理与组织，将程序开发的工作提升至项目管理的层次，让大规模的程序开发更便于管理。

手册导览

关于这本手册的编写方向，基本上是以软件功能的介绍与操作说明做为出发点，并会在其中加入部分相关的观念介绍，至于不同机种之间的差异性，若未造成软件操作上的严重障碍，原则上仅会稍作提示，而细节的部份则请自行参照各机种的相关操作手册或应用指令手册。

章节的编排主要以功能操作为导向，用户可轻易的由目录中找到与软件操作相关的各个主题，如设置通讯参数、在梯形图中插入接点、在线监控、启动仿真器等；若章节的主题有涉及相关的观念介绍时，原则上都会集中在该章节的前半部，而对于已有一定基础或仅需了解如何实作的用户，则可略过该部分的内容，并直接查阅与操作相关的节次即可。此外，部分的内容可能需要搭配其它章节的说明或操作，而此时于内容中亦会附注用户须参考的节次。以下则为手册中各章节内容的简介。

- **第 1 章**与**第 2 章**的内容主要在于软件系统的介绍，包括软件安装、环境介绍与基本设定。
- **第 3 章**的内容将介绍如何在 ISPSOft 中对 PLC 系统进行组态配置与参数设置。
- **第 4 章**为快速入门，在章节中将会透过一个简单的应用范例，而先以非 IEC6113-3 的编程架构来逐步带领用户迅速的得知如何在 ISPSOft 中建立一个传统的梯形图程序，并进行简单的测试。
- **第 5 章~第 8 章**将会正式介绍 IEC61131-3 的相关概念，并说明在 ISPSOft 当中的建构方式。

- 第 9 章将会介绍运动控制与轴的相关概念，并说明在 ISPSOft 当中的使用。
- 第 10 章~第 16 章将逐一介绍梯形图 (LD)、功能块图 (FBD)、指令列表 (IL)、结构化语言 (ST) 及顺序功能图 (SFC)、连续功能图 (CFC)、C 语言 (C) 等语言的编程概念，并说明如何于 ISPSOft 当中进行实作。
- 第 17 章的主要内容在介绍 ISPSOft 所提供的编辑辅助工具。
- 第 18 章将介绍如何在 ISPSOft 中进行监控、测试与除错，其中亦包括仿真器的操作。
- 第 19 章的内容主要在介绍 ISPSOft 所提供的密码与数据保护机制。
- 第 20 章的内容主要是介绍如何在 ISPSOft 当中部署一个网络架构，并藉以建构数据交换的机制。
- 第 21 章将介绍如何透过 ISPSOft 来对 PLC 或项目进行数据的备份与还原。
- 第 22 章将介绍运动控制中常使用到的几种工具，即在 ISPSOft 中的操作。
- 第 23 章将介绍 ISPSOft 中的几个辅助工具，不论编程或调试方面均大幅增加便利性。

图示说明

下表为手册当中搭配说明所使用的图示：

图示	说明
	点击鼠标左键
	点击鼠标右键
	快点两下鼠标左键
	按住鼠标左键后进行拖曳
	使用键盘输入
	操作顺序 (当说明中需要特别表达操作顺序时会配合此标注，例如  、 )
	配合图片说明的指示编号
	重点注意事项 (该事项可能涉及设备、财产或人体的损害)

商标声明

手册中所提及属于台达电子以外的产品与商标，其所有权分属于各公司。

1

第1章 软件简介

目录

1.1	ISPSoft 简介及系统需求	1-2
1.1.1	特色简介	1-2
1.1.2	系统需求	1-3
1.1.3	安装 ISPSoft	1-4
1.1.4	删除 ISPSoft	1-11
1.2	通讯管理员 COMMGR 简介	1-13
1.2.1	COMMGR 的工作模式	1-13
1.2.2	安装 COMMGR	1-14
1.2.3	删除 COMMGR	1-17
1.3	ISPSoft PLC 机种支持	1-18
1.3.1	ISPSoft 语言支持列表	1-18
1.3.2	ISPSoft 功能支持列表	1-18

1

1.1 ISPSOft 简介及系统需求

ISPSOft 为台达新一代的可编程控制器 (PLC) 开发工具。采用 IEC 61131-3 的编程架构，支持六种编程语言以及丰富的应用指令集。除了具备基本的程序编辑功能之外，ISPSOft 还包含了许多的辅助工具，加上多语系的环境与亲切的使用界面，提供给用户的将会是一个便利且高效的开发环境。

1.1.1 特色简介

- 采用国际标准 IEC 61131-3 的编程架构，并内建丰富的应用指令集。
- 支持六种语言，包括梯形图 (LD)、顺序功能图 (SFC)、功能块图 (FBD)、结构化语言 (ST)、以及连续功能图 (CFC) 与 C 语言 (C)，且在一个项目中并不限定只可使用一种语言。
- 包含繁体中文、简体中文、英文...等多语系支持。
- 提供便利的**查找/替换**功能，其范围可选择在目前的工作窗口或是包含整个项目。
- 开放用户自订的操作环境设置
- 项目管理采用清晰且易于操作的 Windows 阶层树状操作接口
- 支持项目群组的管理架构，可在同一的项目群组中，进行多个机种的项目开发。
- 提供许多的便利功能，如批注、书签、程序区段使能/失效、装置与符号管理、仿真器...等。
- 支持多种联机操作，如在线监控、在线编辑、符号表监控、除错模式、PLC 操作/设置...等。
- 支持多种项目对象的**导入/导出**功能
- 可直接开启 WPLSoft 的文件 (*.dvp)，并将其转换另存为 ISPSOft 格式的文件 (*.isp)。
- 提供多种密码及数据保护机制
- 支持新一代的通讯管理员 COMMGR
- 内建多种组态管理工具
 - 硬件规划工具 (HWCONFIG): 用于对系统上的硬件架构进行组态与参数管理。
 - 网络规划工具 (NWCONFIG): 用于管理及配置 PLC 系统的网络架构与数据交换机制。
 - 储存卡工具 (CARD Utility): 透过管理向导与储存卡的操作，可对系统进行备份与还原。
- 针对运动控制提供各种解决方案，包括 PLCopen MC 功能块、G-Code 编辑器、E-CAM 编辑器、定位规划表等工具。
- 本手册对 DVPxxMC 机种内容叙述，仅适用于 DVP15MC、DVP15MC-06、DVP50MC、DVP50MC-04S、DVP50MC-16S 以及 DVP50MC-06 机种。
- 本手册对 AS 机种之内容叙述，不包含 AS5xx 机种。
- 本手册对 AS5xx 机种之内容叙述，仅适用于 AS516E、AS524C、AS532EST 以及 AS564EST 机种。

1.1.2 系统需求

在开始使用 ISPSOft 之前，请务必确认操作系统是否符合以下的需求。

项目	系统需求	
操作系统	Windows XP / 7 / 8 / 10	
CPU	Pentium 1.5G 以上机种	
内存	1024MB 以上 (建议使用 2048MB 以上)	
磁盘驱动器	硬盘容量：至少须提供 5000MB 以上空间供 ISPSOft 使用	
光驱	用于透过光盘来进行软件安装の場合 (选配)	
显示器	分辨率：1024 × 768 以上 (建议显示设置：1920x 1080 / 96DPI)	
键盘/鼠标	一般的键盘鼠标或与 Windows 兼容的装置	
打印机	具 Windows 驱动程序的打印机 (选配 · 用于项目内容的打印)	
RS-232 端口	与 PLC 联机之用	三者择一即可。但仍须依据主机本身或搭配模块所提供的通讯接口 (*1)
USB	与 PLC 联机之用	
以太网	与 PLC 联机之用	
联机软件	计算机中须安装 通讯管理员 - COMMGR 的软件 (*2)	
支持机种	PLC：AH500 全系列 / DVP 全系列 (不含 DVP-PM 系列) (*3) 交流马达驱动器：VFD 系列内建 PLC 机种。 文本显示器：TP 系列内建 PLC 机种。	

*1. ISPSOft 支持多种与 PLC 的联机方式，进行联机前请先确认使用机种所提供的通讯端口及所支持的联机模式。

*2. 关于通讯管理员 COMMGR 的相关介绍请参阅第 1.2 节。

*3. AH10PM-5A 与 AH20MC-5A 模块的项目尚须搭配 PMSOft V2.05 版以上的软件来进行开发。

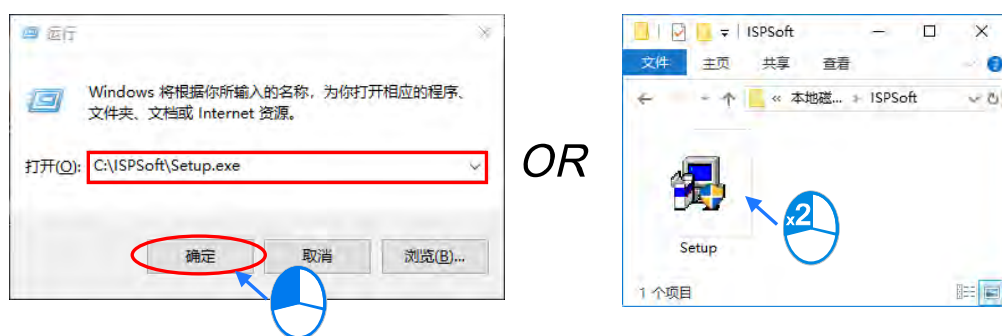
*4. 以上所提及的功能与规格仅适用于 ISPSOft V2.00 或更高的版本，先前的版本可能不具备完整功能。

1

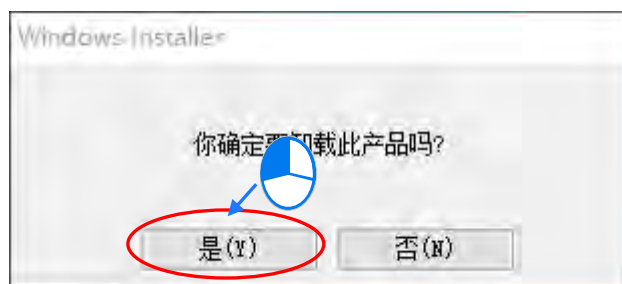
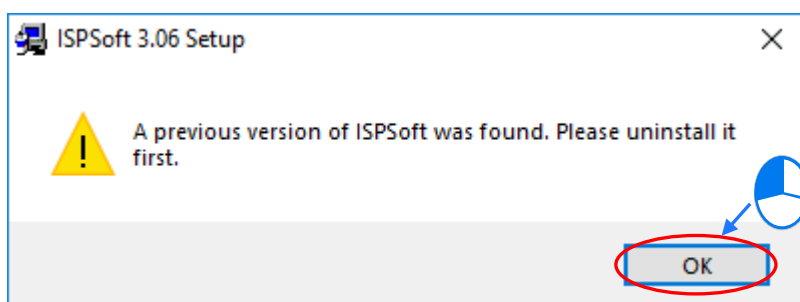
1.1.3 安装 ISPSOft

当计算机中已安装旧版的 ISPSOft 时，执行 ISPSOft 安装程序时，系统会自动检测有无安装旧的 ISPSOft 版本，并提示要移除旧的 ISPSOft，移除完成之后会自动继续安装新的 ISPSOft。

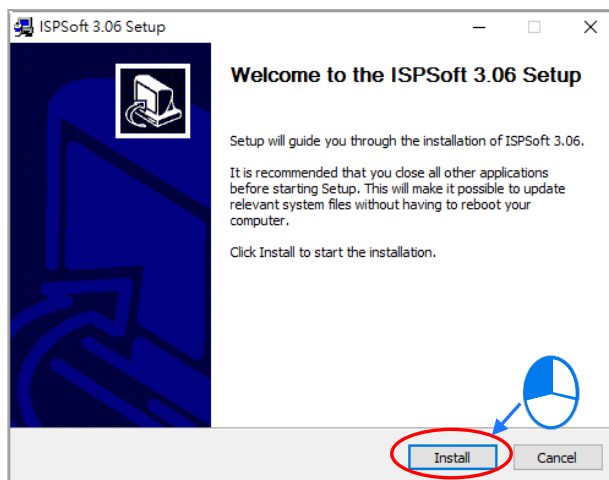
- (1) 启动计算机并进入操作系统，且须以具备计算机管理员的权限登入才可进行软件安装。
- (2) 将 ISPSOft 光盘放入光驱中或从台达官方网站 <http://www.delta.com.tw/ch/index.asp> 下载 ISPSOft 的安装程序（从网络下载的安装程序必须经过解压缩后才可进行安装。）。
- (3) 在「开始」→「运行」的窗口中指定安装文件路径后按下「确定」，或是直接于安装文件的图标上双击鼠标左键，以执行安装程序。



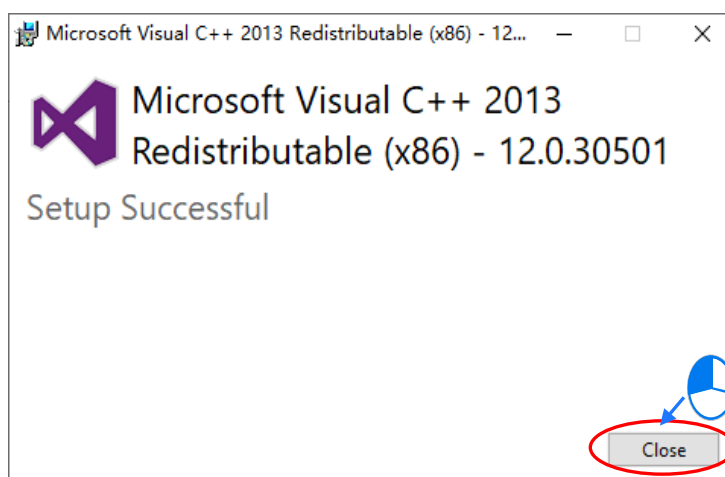
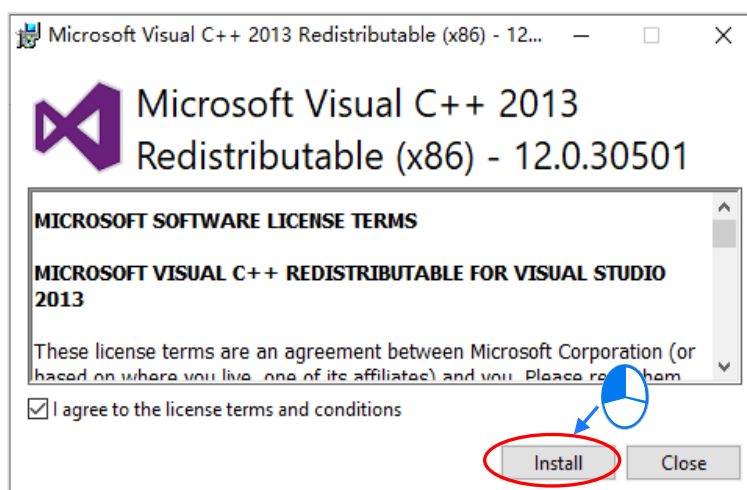
- (4) 如果系统内有安装旧版本 ISPSOft，出现此画面提示要移除旧版本的 ISPSOft，此时按下「OK」进行下一步，接着按「是 (Y)」，系统会开始移除旧版本 ISPSOft。



- (5) 待安装窗口出现后，按下「Install」进行下一步。



- (6) 安装程序会检测计算机有没有安装 Microsoft Visual C++ 2013 若没有则会跳出下面的安装画面，此时按下「Install」进行下一步，安装完成后按下「Close」。

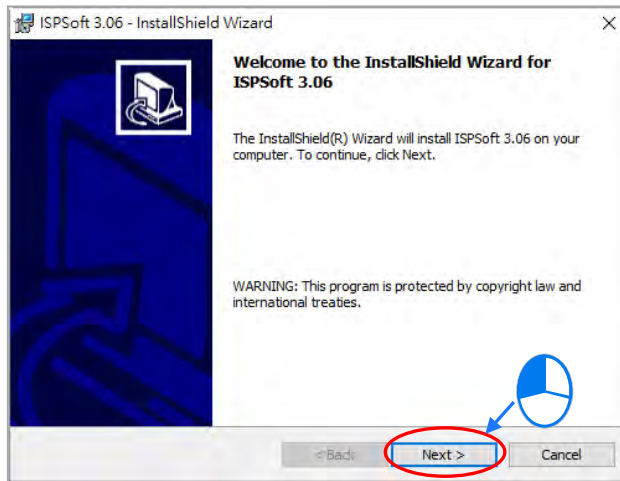


1

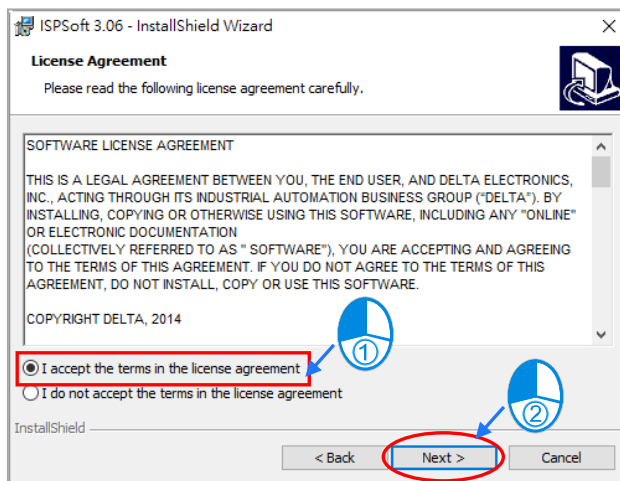
- (7) 若计算机没有 DELTA PLC USB 驱动，就会跳出这个页面并自动帮你安装 PLC USB 驱动，只须单击「安装」进行下一步。



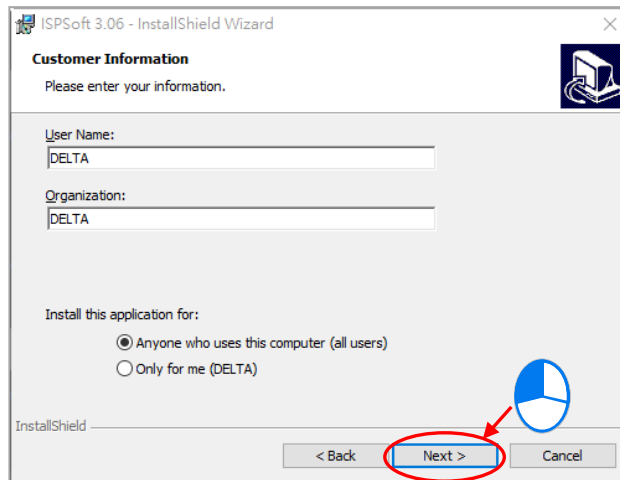
- (8) 按下「Next」进行下一步。



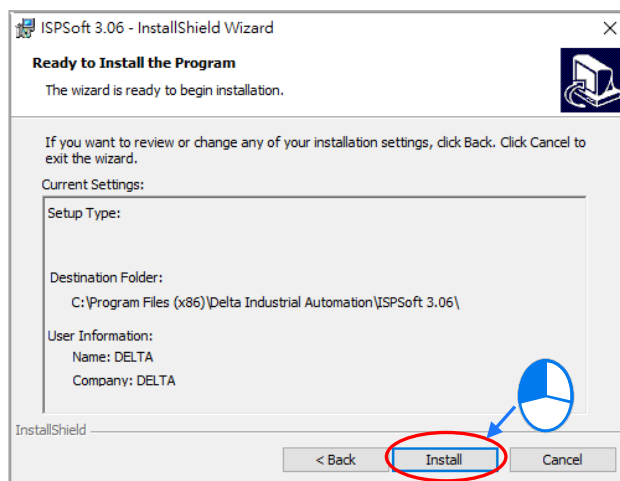
- (9) 选取「I accept the terms in the license agreement」同意安装，完成后按下「Next」进行下一步。



(10) 输入用户的相关信息后，按下「**Next**」进行后续的安装工作。



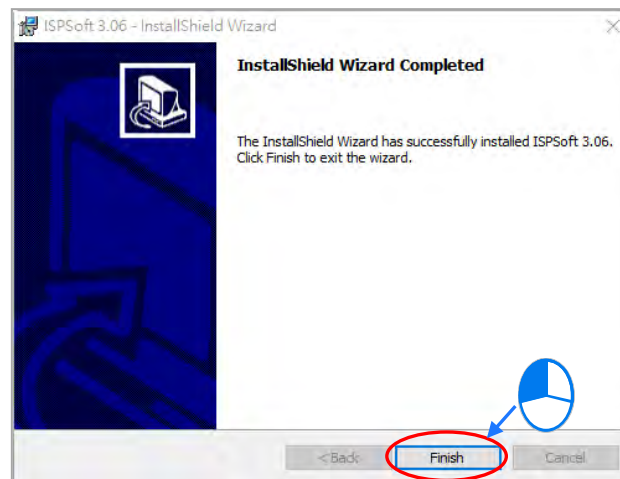
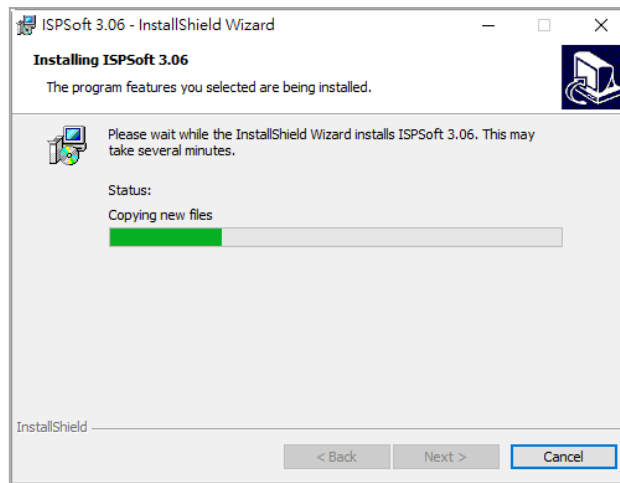
(11) 确认安装信息后，按下「**Install**」进行下一步。



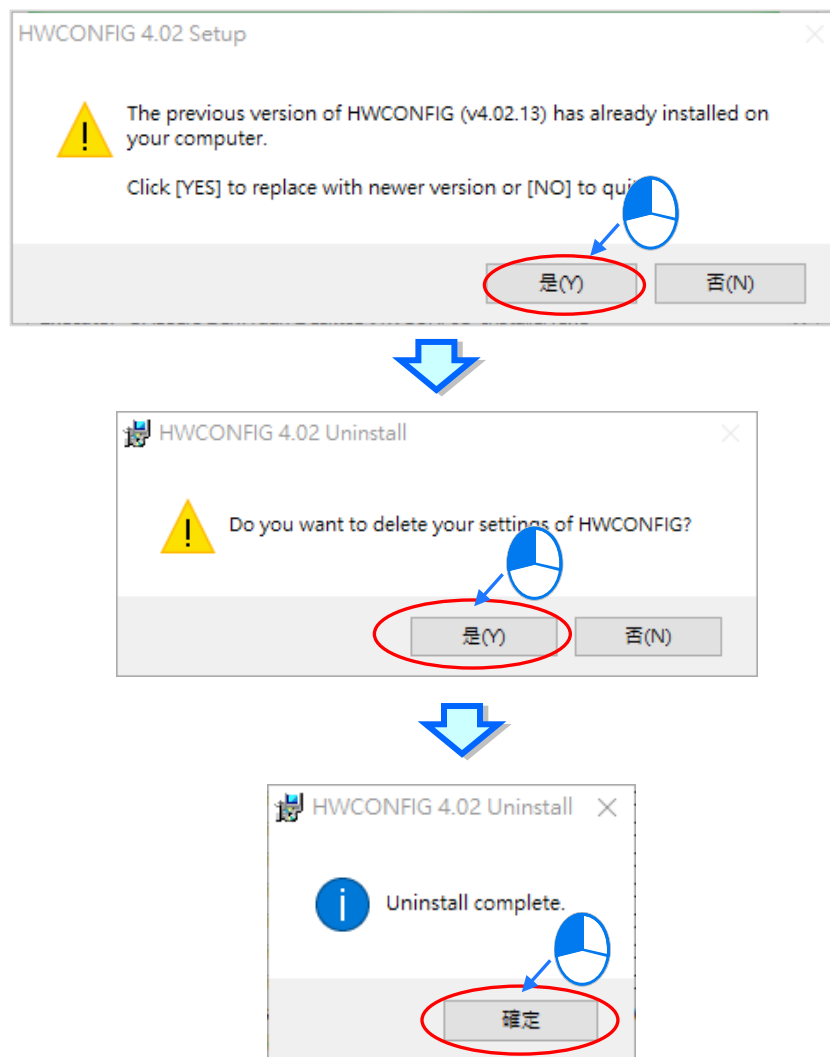
1

1

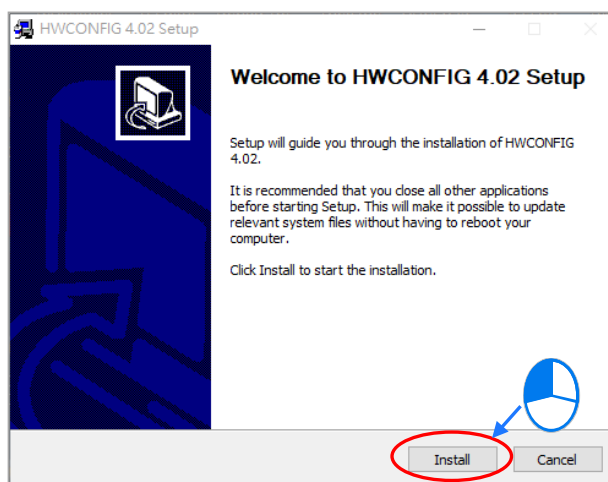
(12) 等待安装进度条完成时，按下「Finish」即可进行下一步。



(13) 接着会开始安装 HWCONFIG，如果系统内有安装旧版本 HWCONFIG，才会出现此画面提示要移除旧版本的 HWCONFIG，此时按下「是 (Y)」进行下一步，接着按下「是 (Y)」，系统会开始移除旧版本 HWCONFIG

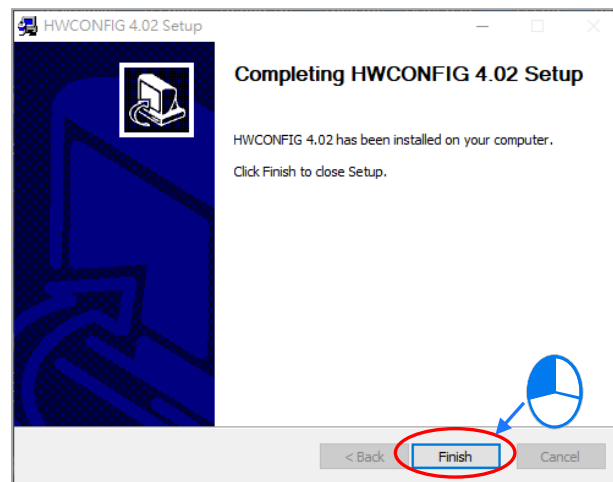


(14) 待安装窗口出现后，按下「Install」进行下一步。

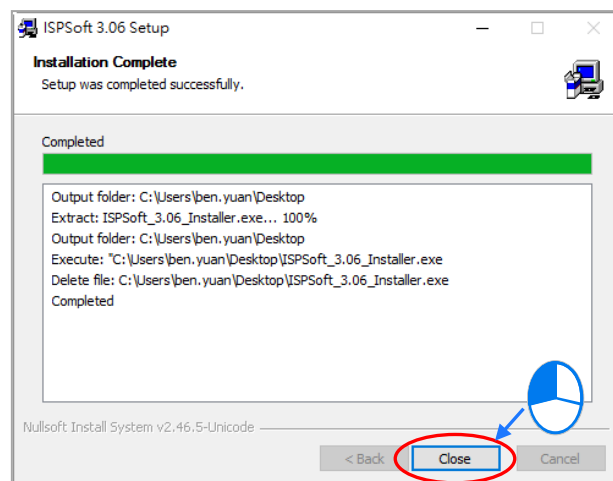


1

(15) HWCONFIG 安装完成后，按下「**Finish**」进行下一步。



(16) 完成安装后，桌面及开始菜单中会自动建立程序的执行快捷方式，按下「**Close**」即可结束安装。



1.1.4 删除 ISPSOft

- (1) 正常的情况下只要单击 ISPSOft Uninstall 或到控制面板的程序和功能项目 就可以移除 ISPSOft ; 在找不到 ISPSOft Uninstall 时 , 可以通过下面两种方式操作 :

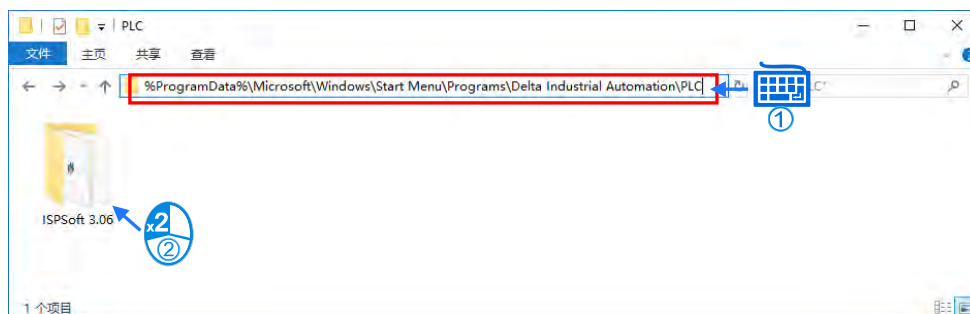
方法一 : 右击选单上的 ISPSOft x.xx > 更多 > 开启文件位置。

方法二 : 在文件管理窗口的地址栏输入

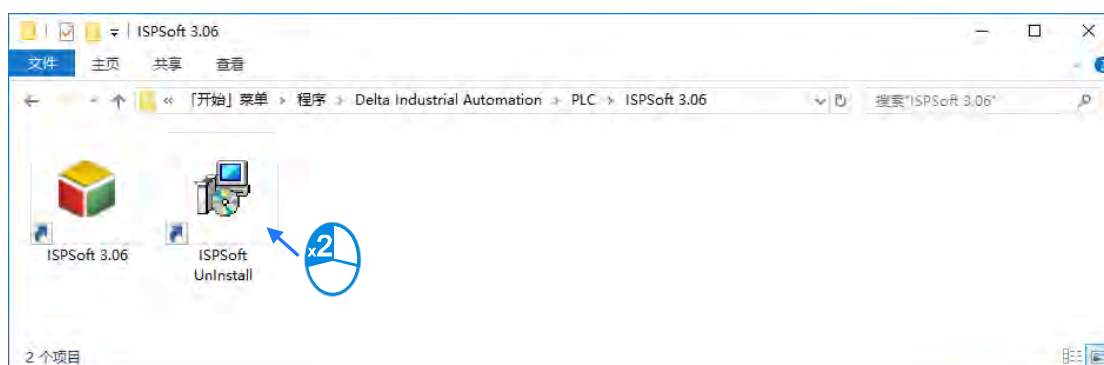
%ProgramData%\Microsoft\Windows\Start Menu\Programs\Delta Industrial Automation\PLC\
按下 **Enter** 后 , 双击 ISPSOft x.xx 文件夹。



OR

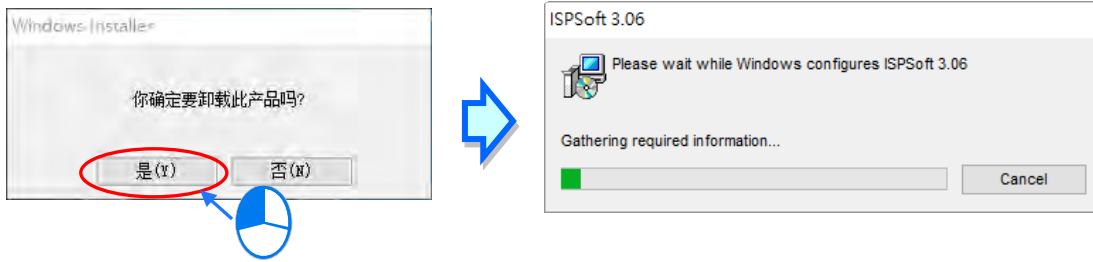


- (2) 在「ISPSOft UnInstall」双击鼠标左键 , 移除 ISPSOft 。



1

(3) 待 ISPSOft 移除程序窗口出现后点选「是」确认动作后即可开始进行移除，移除完成后窗口会自动关闭。

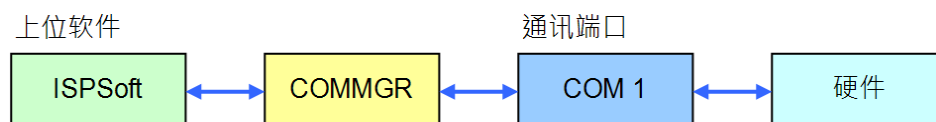


1.2 通讯管理员 COMMGR 简介

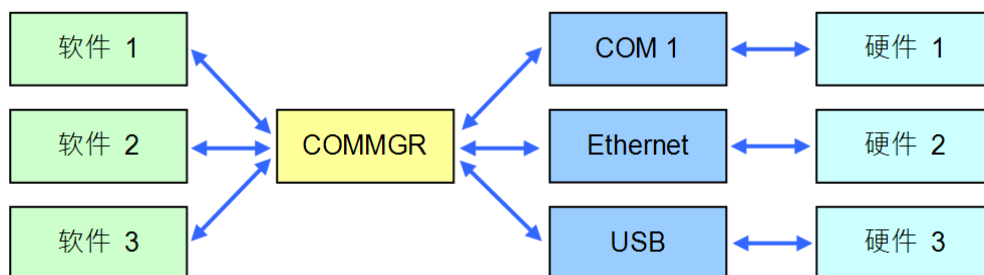
COMMGR 是台达电子的通讯管理工具，其主要的功能在于扮演台达软件与硬件之间的通讯桥梁，而通过 COMMGR 的管理，联机的工作亦将变得更为便利也更有效率。

1.2.1 COMMGR 的工作模式

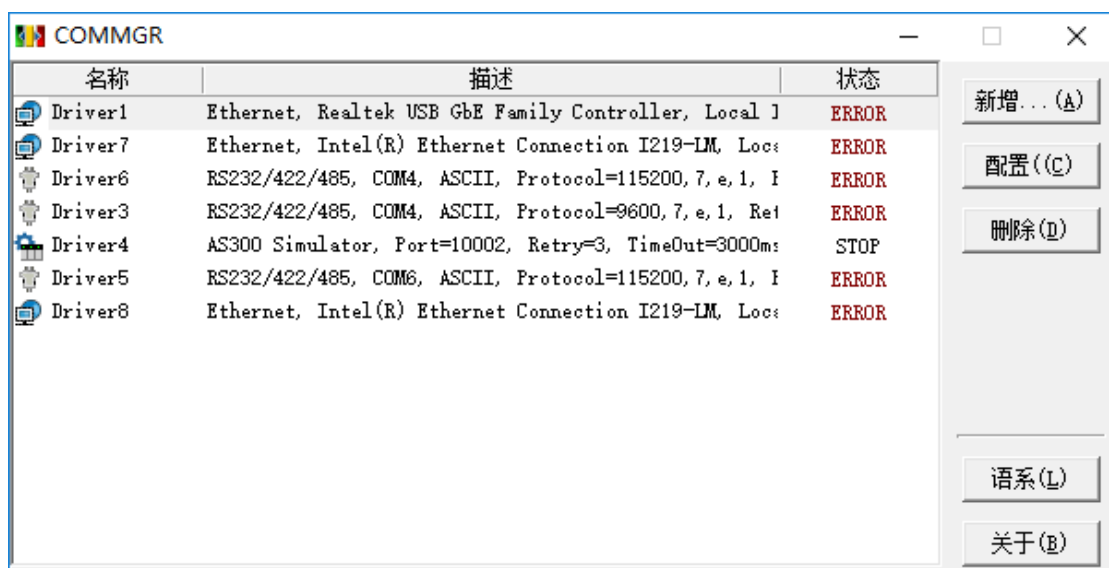
对于所有需要设定的联机参数，我们都可以事先将它们建立在 COMMGR 的管理列表中，而這些事先建立好的联机参数我们便称之为 **Driver**。依实际的使用状况，我们可以直接在 COMMGR 当中设置每组 **Driver** 的启动状态，而当该组 **Driver** 的状态被设定为「**START**」之后，实际的联机通道便会自动建立，接下来只须在 ISPSOft 中指定欲使用的 **Driver** 即可进行联机。



除了 ISPSOft 之外，其它通过 COMMGR 来进行联机的软件也可在同一时间一起运作，COMMGR 会主动管理所有的通讯命令，并可让各软件与硬件之间保持最佳的联机状态。



下图为 COMMGR 的主画面及管理列表，其中「名称」字段可由用户自行设定，「描述」字段则会显示各组 Driver 的相关参数，而「状态」则显示该 Driver 目前的状态。

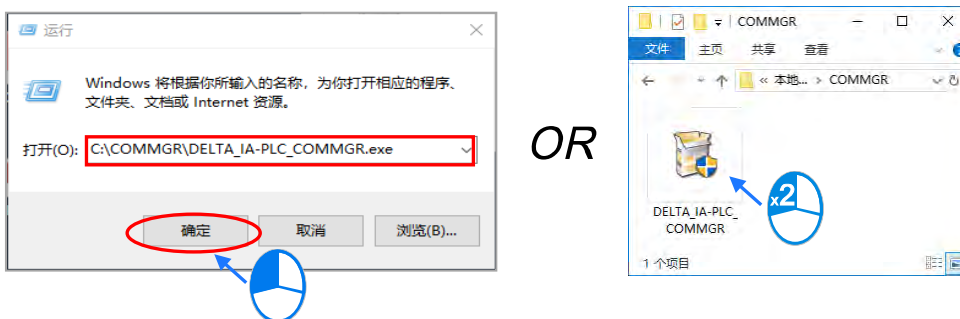


1

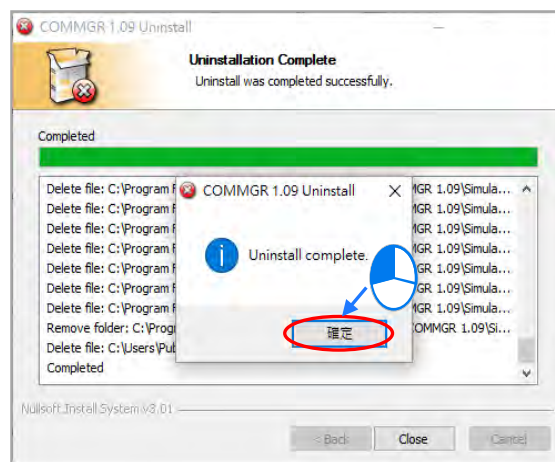
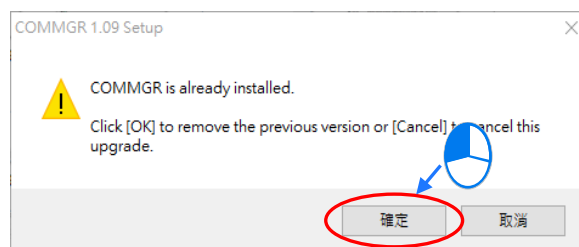
1.2.2 安装 COMMGR

COMMGR 为独立于 ISPSOft 的工具软件，因此用户必须另外对其进行安装，而当计算机中已安装旧版的 COMMGR 时，执行 COMMGR 安装程序，系统会自动检测有无安装旧的 COMMGR 版本，并提示要移除旧的 COMMGR，移除完成后会自动继续安装新的 COMMGR。

- (1) 启动计算机并进入操作系统，且须以具备计算机管理员的权限登入才可进行软件安装。
- (2) 将含 COMMGR 安装文件的光盘放入光驱或从台达网站 <http://www.delta.com.tw/ch/index.asp> 下载安装程序（从网络下载的安装程序必须经过解压缩后才可进行安装）。
- (3) 于「开始」→「运行」的窗口中指定安装文件路径后按下「确定」，或是直接于安装文件的图标上双击鼠标左键，以执行安装程序。



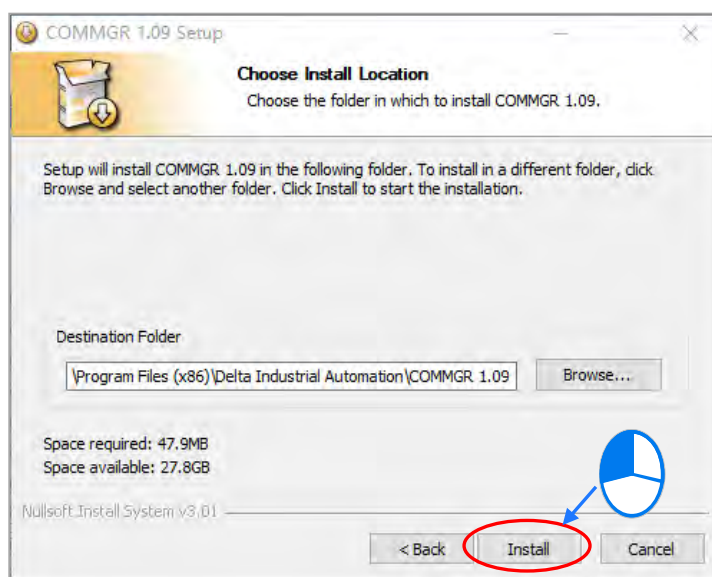
- (4) 如果系统内有安装旧版本 COMMGR，出现此画面提示要移除旧版本的 COMMGR，此时按下「确定」系统会开始移除旧版本 COMMGR，等待移除完成窗口出现后按下「确定」。



- (5) 待安装窗口出现后，按下「Next」。

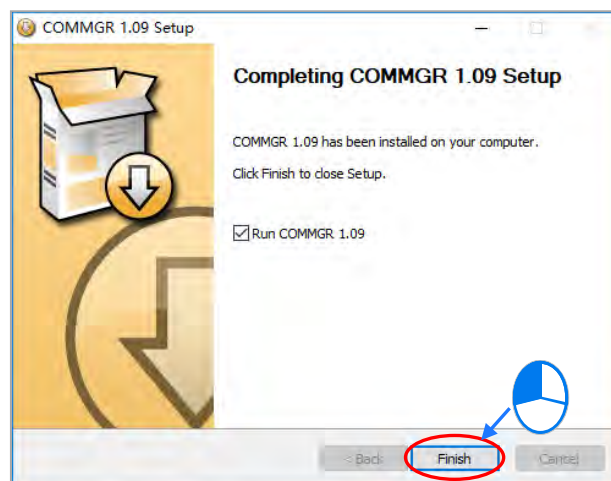
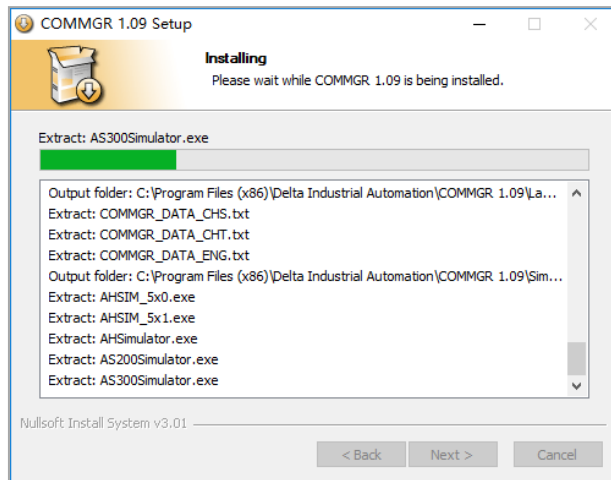


- (6) 建议使用默认路径安装，输入完安装路径后，按下「Install」进行后续的安装工作。



(7) 完成安装后，开始菜单中便会自动建立程序的执行快捷方式，而按下「**Finish**」后即可结束安装。

1



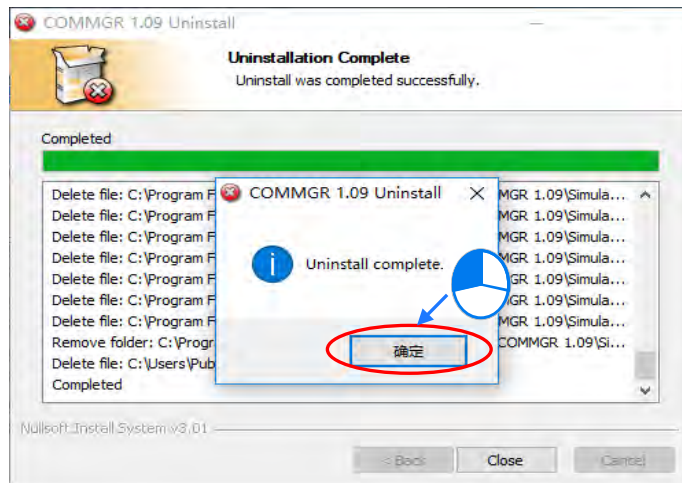
1

1.2.3 删除 COMMGR

- (1) 进入 Windows 设定中的「应用程序与功能」，选取「COMMGR x.xx」后按下「卸载」。



- (2) 按下「是 (Y)」系统会开始移除 COMMGR，等待移除完成窗口出现后按下「确定」，即可完成移除。



1

1.3 ISPSOft PLC 机种支持

ISPSOft 支持台达多款 PLC 设备，提供的功能也非常多，为了方便用户查询，这个章节将会介绍 ISPSOft 对于 AS、AH5x0、AH5x1、AH560、AHxxEMC、DVP、DVPxxMC、AS5xx，这八种系列 PLC 机种所支持的功能做一个整理。

1.3.1 ISPSOft 语言支持列表

ISPSOft 支持六种语言，包括梯形图 (LD)、循序功能图 (SFC)、功能块图 (FBD)、结构化语言 (ST) 以及连续功能图 (CFC) 与 C 语言 (C)，机种与语言支持列表如下所示。

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
梯形图 (LD)	V	V	V	V	V	V	V
功能块图 (FBD)		V	V	V			
结构化语言 (ST)	V	V	V	V	V	只支持 ES3	V
循序功能图 (SFC)	V	V	V	V	V	V	
连续功能图 (CFC)	V				V	只支持 ES3	
C 语言 (C)	V						

1.3.2 ISPSOft 功能支持列表

ISPSOft 支持的功能非常的多，下表针对较大的功能项目做机种支持的整理。

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
运动模块		V	V	V	V		
HWCONFIG	V	V	V	V	V	只支持 ES3	V
HWCONFIG 打印功能	V	V	V	V	V		
订购清单	V	V	V	V	V		
自定义函数库	V	V	V	V	V	V	
台达函数库	V	V	V	V	V	V	V

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
自定义	V	V	V	V	V	V	V
轴					V		V
变更 PLC 机种	V	V	V	V	V	V	V
装置批注表	V	V	V	V	V	V	
装置使用状况	V	V	V	V	V	V	
寄存器编辑	V	V	V	V	V	V	
装置状态编辑	V	V	V	V	V	V	
文件寄存器 编辑	V					只支持 SA/SX/SC/EH/EH2/ SV/EH2-L/ES2/ ES2-E/EX2/SX2/SA2 /EH3/SV2/EH3-L/ ES3	
步序定位	V	V	V	V	V	V	
程序比对	V	V	V	V	V	V	
在线编辑	V	V	V	V	V	只支持 EH3/EH3-L/SV2/ES2 /ES2-E/EX2/SA2/ SX2/MC/SS2	
除错工具栏	V	V	V	V	V	V	
系统纪录	V	V	V	V	V	只支持 ES3	V
3D 坐标图					V		
程序标识符	V	V	V	V	V	V	V
项目密码	V	V	V	V	V	V	V
POU 密码	V	V	V	V	V	V	V
子程序密码						不支持 ES3	
NWCONFIG		V				不支持 ES3	
数据备份存储 卡辅助工具						只支持 EH3/EH2	

1

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
PLC 永久 备份设定						只支持 EH2/EH2-L/EH3/ EH3-L/SV/SV2	
CARD Utility	V	V	V	V	V	V	V
G-Code 编辑器							V
E-CAM 编辑器	V				V		V
定位规划表	V					只支持 ES3	
数据示波器	V		V		V	只支持 ES3	
数据记录器	V		V			只支持 ES3	
高速计数器 向导	V					只支持 ES3	
AIO 向导						不支持 ES3	
温度监控						不支持 ES3	
称重模块						不支持 ES3	
扩充模块辅助 设计向导						不支持 ES3	
扩充模块 监控向导						不支持 ES3	
NTC 向导						只支持 SX、SE、SS2、SX2、 SA2、SV2	只支持 DVPxxMC
运动参数模拟							V

2

第2章 启动与设定

目录

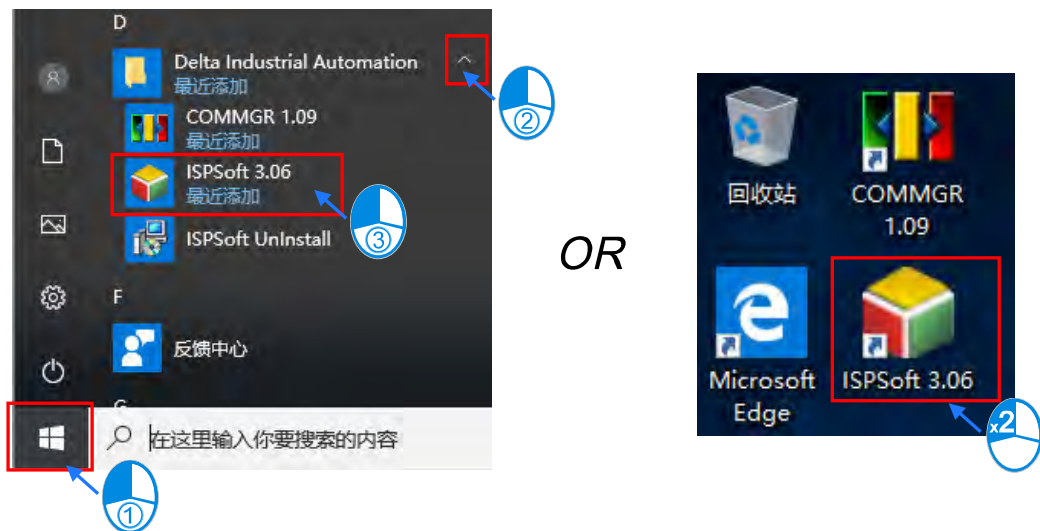
2.1 开始导览与环境介绍	2-3
2.1.1 进入 ISPSOft 的第一步	2-3
2.1.2 窗口标题与状态栏	2-7
2.1.3 功能工具栏	2-8
2.1.4 图标工具栏	2-11
2.1.5 项目管理区与信息显示区	2-12
2.1.6 编辑工作区	2-14
2.2 ISPSOft 的项目架构	2-17
2.2.1 单机项目与群组项目	2-17
2.2.2 整合运动模块项目	2-18
2.2.3 ISPSOft 的单机项目管理	2-19
2.2.4 ISPSOft 的群组项目管理	2-25
2.2.5 ISPSOft 的运动模块项目管理	2-32
2.3 ISPSOft 中的基本设定	2-37
2.3.1 系统及环境设定	2-37
2.3.2 用户编辑格式的导入与导出	2-43
2.3.3 导出程序到文字格式	2-44
2.4 通讯设定	2-45
2.4.1 通讯管理员 COMMGR 的启动与关闭	2-46
2.4.2 COMMGR 的驱动管理	2-47
2.4.3 建立联机通道 - 驱动的建立	2-48
2.4.3.1 RS232/422/485 通讯参数设置	2-50
2.4.3.2 USB (Virtual COM) 通讯参数设置	2-51
2.4.3.3 DirectLink (USB) 通讯参数设置	2-51
2.4.3.4 Ethernet 通讯参数设置	2-52
2.4.3.5 DirectLink (Ethernet) 通讯参数设置	2-53
2.4.3.6 DVP Simulator 通讯参数设置	2-54
2.4.3.7 AH5x0 与 AH5x1 Simulator 通讯参数设置	2-54
2.4.3.8 AS Simulator 通讯参数设置	2-55

2.4.4	建立联机通道 - 驱动的启动与停止	2-55
2.4.5	建立联机通道 - 驱动的修改与删除	2-57
2.4.6	建立联机通道 - 仿真器的面板操作	2-58
2.4.7	建立 ISPSOft 与 COMMGR 之间的连结	2-61
2.4.8	连接主机与通讯端口	2-63
2.4.9	实际联机测试	2-66

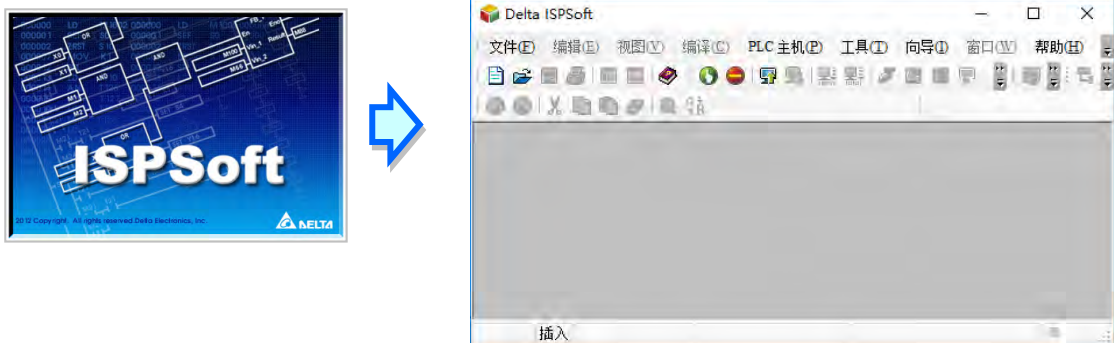
2.1 开始导览与环境介绍


2.1.1 进入 ISPSOft 的第一步

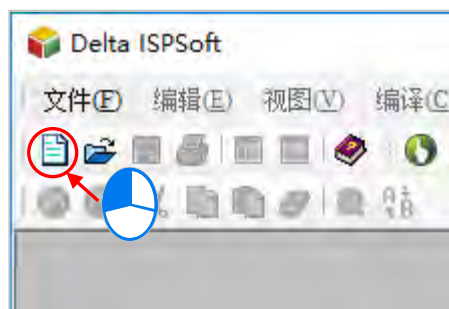
成功安装 ISPSOft 后，系统便会自动在桌面及程序菜单中建立快捷方式，点击该快捷方式即可启动 ISPSOft。ISPSOft 也可同时开启多个软件窗口，只要依据相同步骤，另外再启动 ISPSOft 即可。



待欢迎画面结束后便可正式进入 ISPSOft，而刚进入时，窗口中仅有一些基本的功能可供使用。

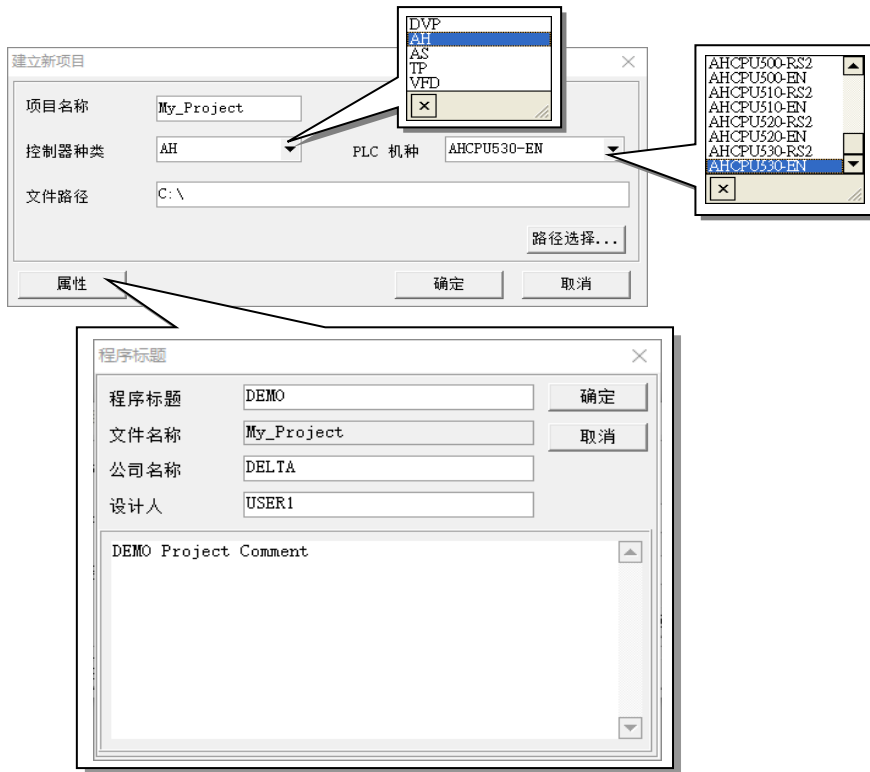




直接点击图标工具栏中的  图标即可建立一个新的项目。

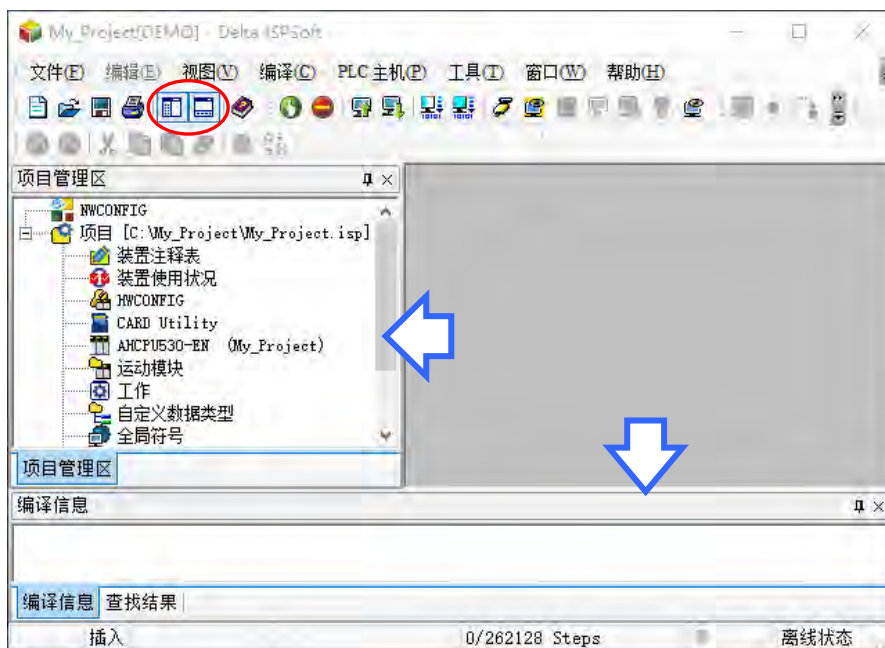


在设置窗口中依序输入「项目名称」及「文件路径」，之后在「控制器种类」及「PLC 机种」的下拉选单中选择机种型式；而点击「属性」则可为此项目加入相关的信息及描述。完成后按下「确定」即可。

2



成功建立项目后，画面左侧便会开启一个「项目管理区」，并以阶层树形图的方式列出所有对象；若未出现该区域，请点击功能工具栏的 **视图 (V) > 工作区 (W)** 或图标工具栏的  图标即可。若点击 **视图 (V) > 输出窗口 (M)** 或图标工具栏的  图标则可开启「信息显示区」。

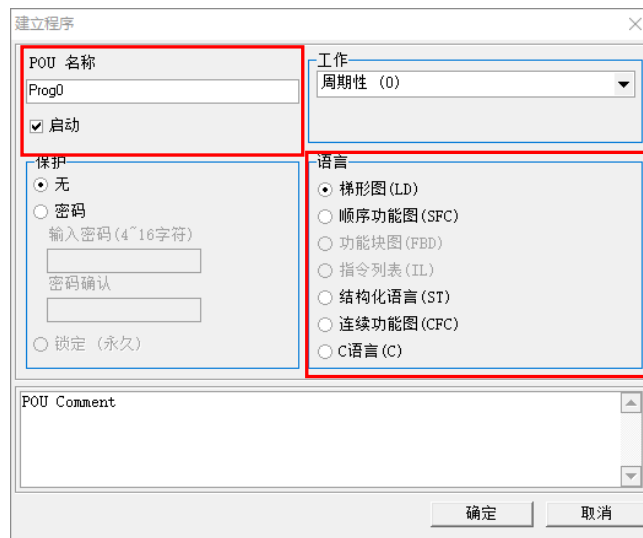


刚建立项目时，主要的工作区因尚未进行任何编辑而处于空白状态；若要开始建立 PLC 程序时，可在「项目管理区」的「程序」项目上点击鼠标右键，之后在选单中单击 **新增 (N)**。



2

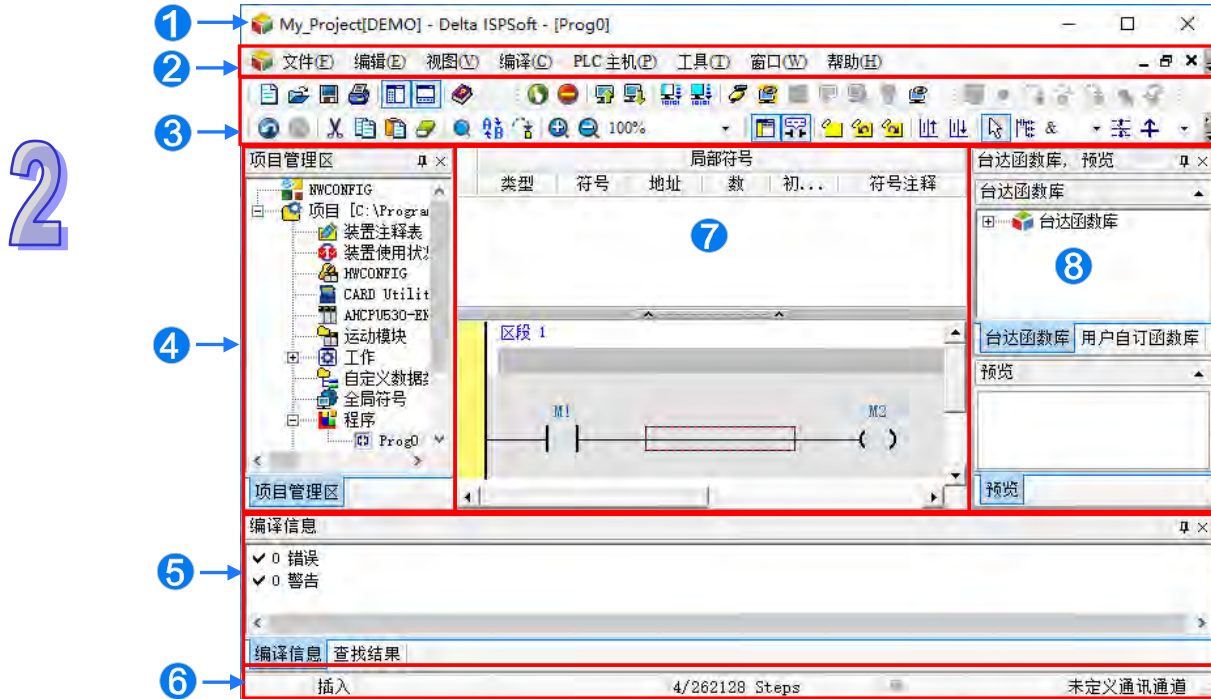
在属性窗口中，用户可在「POU 名称」字段中输入该程序的命名，之后再在「语言」字段中选择欲使用的编程语言；至于其它参数则可先沿用默认值，其意义将在第 5.4 节的内容当中另行介绍。



成功新增一个程序单元后，在「项目管理区」的「程序」项目下便会多出一个对象，而在主要的工作区中也会自动开启一个程序编辑窗口。

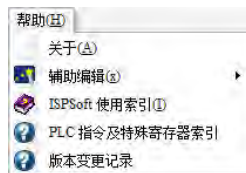



下图便是一个完整的 ISPSoft 工作画面。



- ❶ 窗口标题：显示目前所编辑的**项目名称与程序标题**。
- ❷ 功能工具栏：共有 8 类的功能操作选项。
- ❸ 图标工具栏：共有 5 类的图标工具栏可供操作。
- ❹ 项目管理区：阶层树状结构的项目对象管理接口。
- ❺ 信息显示区：显示编译结果及项目查找的相关信息。
- ❻ 状态栏：显示目前编辑或联机状态的相关信息。
- ❼ 编辑工作区：显示程序编辑区、符号表及监控表...等工作窗口。
- ❽ 函数库区：列出台达函数库与用户自定义函数库的内容。

在使用 ISPSoft 的过程中，用户可通过功能工具栏的「**帮助 (H)**」项目来启动各种的在线辅助。

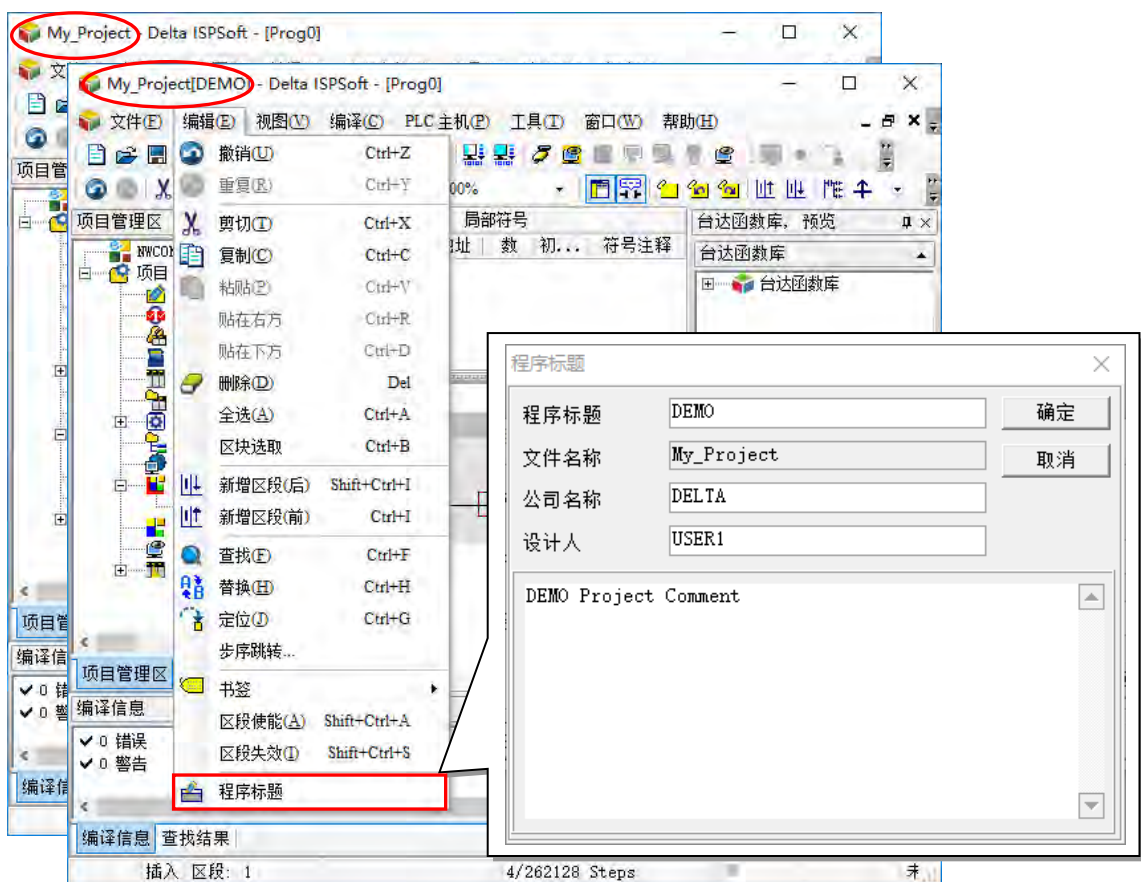


操作项	功能说明
关于	查询软件版本及发行日期等相关信息。
辅助编辑	辅助编辑工具如通讯格式<=>参数码。
ISPSoft 使用索引	查询软件的使用说明。(亦可通过图标工具栏的  图标来开启)

操作项	功能说明
PLC 指令及特殊寄存器索引	查询各种指令及寄存器的说明。而当在程序编辑区中单击某一指令后，按下键盘的【F1】键便可直接开启该指令的说明页面。
版本变更纪录	列出此版本软件的新增或修改功能。

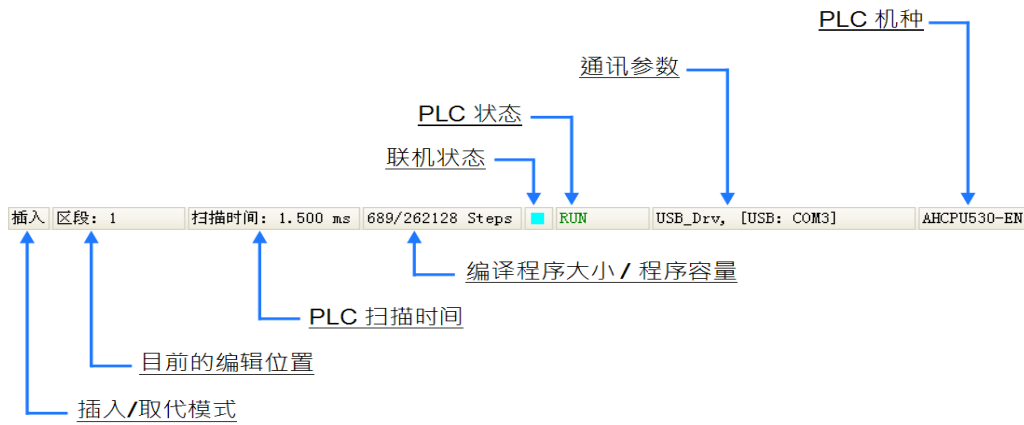
2.1.2 窗口标题与状态栏

成功建立项目之后，ISPSoft 的窗口标题便会显示该项目的名称，其格式为 **项目名称[程序标题]**，而当没有设置「程序标题」时，窗口标题则只会显示「项目名称」；当要查看「程序标题」等信息时，只要在功能工具栏上依序单击 **编辑 (E) > 程序标题** 即可。



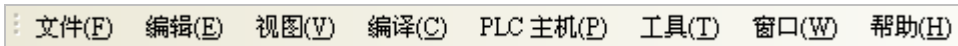
窗口下方的状态栏主要在于显示目前的各种工作状态，包括编辑模式透过键盘上的 **Insert** 按键可以切换（插入/取代）、目前编辑位置、PLC 扫描时间、程序大小/容量、联机状态、PLC 状态、通讯参数以及 PLC 机种等信息。

2



2.1.3 功能工具栏

ISPSOft 的功能工具栏共有 8 类功能项，且依据目前所进行的编辑工作与选用的机种，各类功能项的下层项目也会有所差异，在此先稍做浏览介绍，详细的操作方式则将在之后的章节当中陆续介绍。



- **文件**：主要功能为基本的项目存取操作。



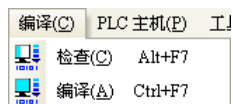
- **编辑**：主要提供进行编辑工作时所需的操作功能。



- **视图**：主要功能为查看项目中的信息以及设置工作环境的显示方式。



- **编译**：主要功能为检查程序语法或将项目程序编译为可让 PLC 运行的执行码。



2

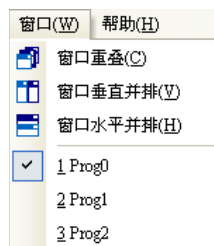
- **PLC 主机**：主要为通过 ISPSOft 对 PLC 主机进行的联机操作或设置功能。



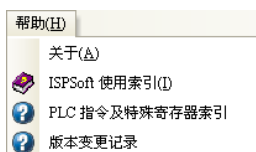
- **工具**：主要为 ISPSOft 所提供的一些便利功能。



- **窗口**：主要功能为管理编辑工作区中的各个窗口。

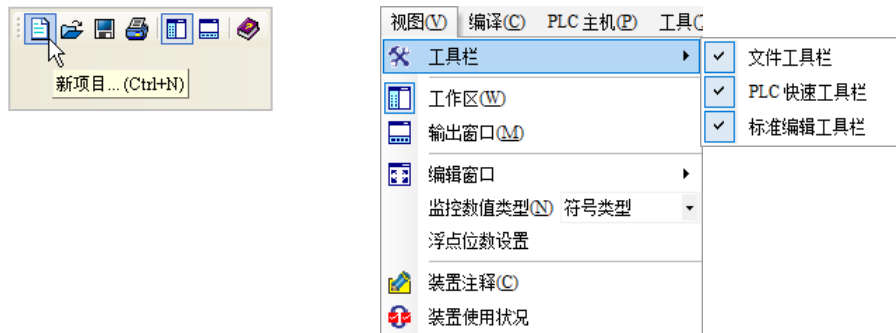


- **帮助**：主要为 ISPSOft 所提供的一些在线辅助功能。



2.1.4 图标工具栏

ISPSoft 中共有 5 类的图标工具栏，而且同样会依据目前的编辑内容与选择机种而有不同的显示。当鼠标移至图标的上方并稍作停留时，画面便会显示该图标的功能与快捷键提示；而在功能工具栏的视图 (V) > 工具栏 中，用户可自行勾选所要显示或隐藏的图标工具栏。



- **文件工具栏**：主要为项目操作的相关功能。



- **标准编辑工具栏**：主要为与编辑工作相关的操作选项。



- **PLC 快速工具栏**：主要为与 PLC 操作相关的功能。

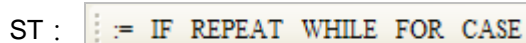



- **除错工具栏**：主要为除错功能的相关操作选项，且不同机种的操作图标也会有所差异。

*.目前 DVPxxMC/AS5xx 不支持除错模式。



- **编程工具栏**：主要为编辑程序时所需的各项功能，且不同的编辑语言会有不同的操作图标。





SFC : CFC : 



2.1.5 项目管理区与信息显示区

2



项目管理区的主要内容为与项目开发相关的一些对象，并以阶层树状结构做为其管理界面；内容包括由系统提供的组态管理工具、设备信息与应用指令，同时也包括用户自行建立的变量符号、程序项目、监控表等，且随着选用机种的不同，其所显示的项目，如工具、指令...等，亦会有所不同。

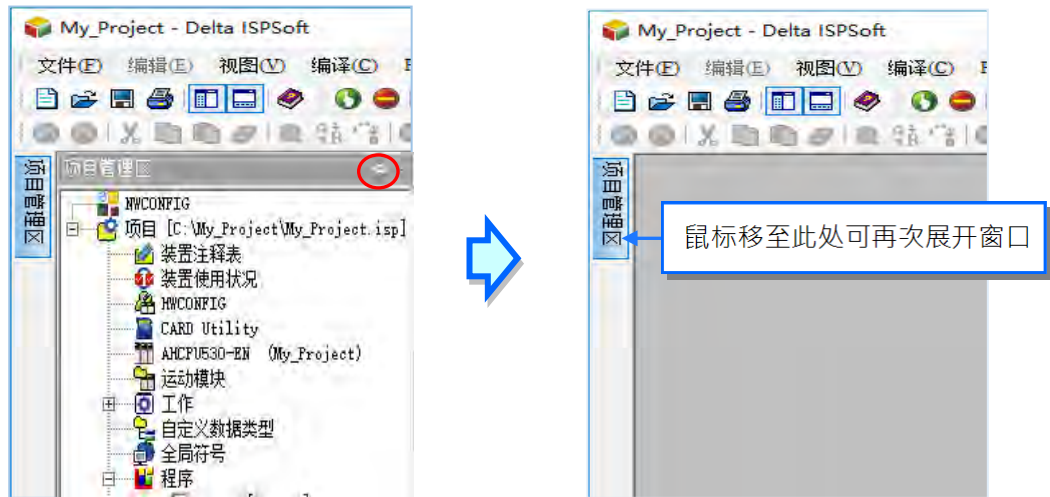
默认的项目管理区会固定在 ISPSOft 窗口的左侧，点击此区右上角的  图标即可将其关闭，而通过切换图标工具栏上的  图标或是功能工具栏的 **视图 (V) > 工作区 (W)** 项目则可开启或关闭项目管理区的显示。当在某个项目对象上按下鼠标右键时，画面便会出现对应的快捷选单，利用选单上的功能便可进行相关的对象操作。

信息显示区主要有编译信息与查找结果两个页面，而两者之间则可通过信息显示区下方的选项卡来进行切换。编译信息页面主要是显示进行程序编译或语法检查时的结果或提示信息，而查找结果页面则是用来条列执行项目查找后的查找结果。

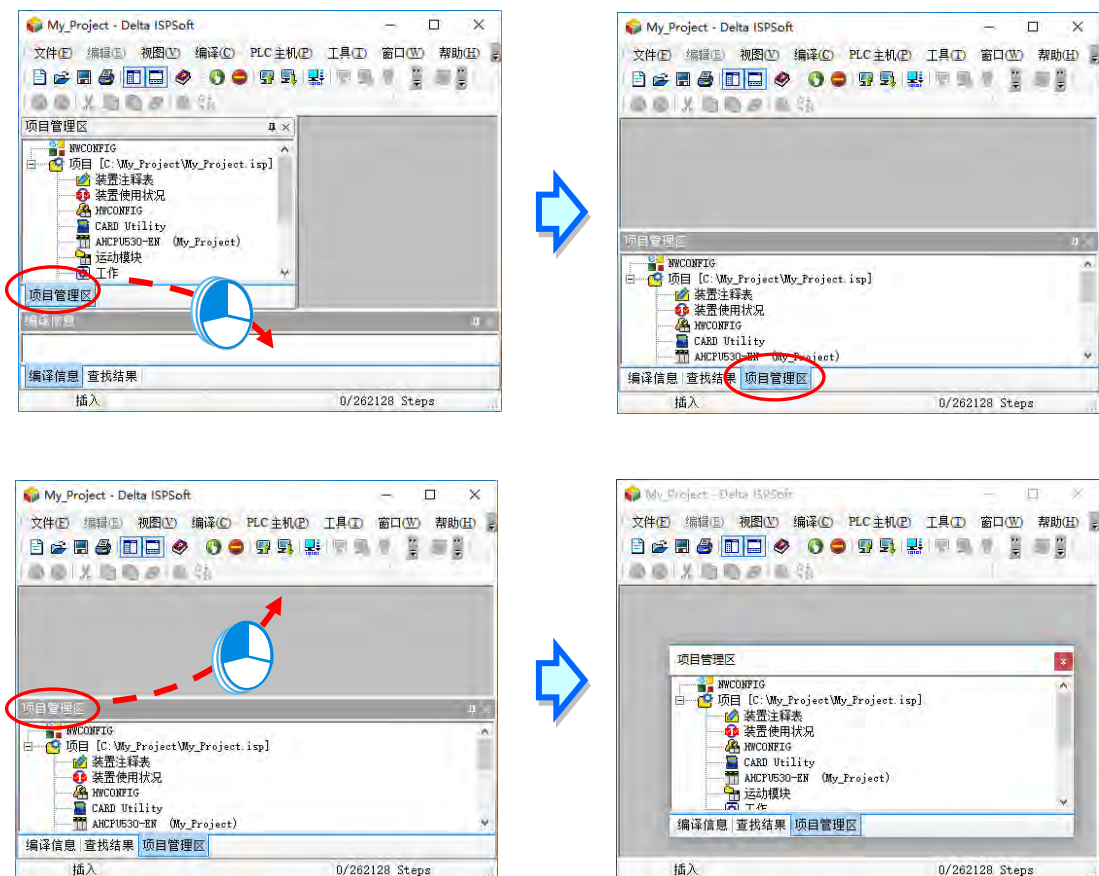
信息显示区的预设位置是固定在 ISPSOft 窗口的下方，而切换图标工具栏上的  图标，或是切换功能工具栏的 **视图 (V) > 输出窗口 (M)** 项目便可开启或关闭信息显示区的显示，而点击此区右上角的  图标亦可将其关闭。



不论是项目管理区或信息显示区，切换两者窗口右上角的  图标便可改变窗口的显示方式；当图标切换成  状态时便可将该窗口设为自动隐藏，只要鼠标离开该区域一段时间之后，该区域的窗口便会自动收藏至边缘，而当将鼠标移至收藏边缘上的选项卡时，鼠标所指的窗口便会再次展开。



当窗口未设置为自动隐藏的状态时，用户可在项目管理区或信息显示区的选项卡上按住鼠标左键，之后便可将该页面拖放至任何位置，甚至可拖放至另一区的位置上与其合并；而要拖放整个区域时，只要先按住该区上方的窗口标题之后再行拖放即可。

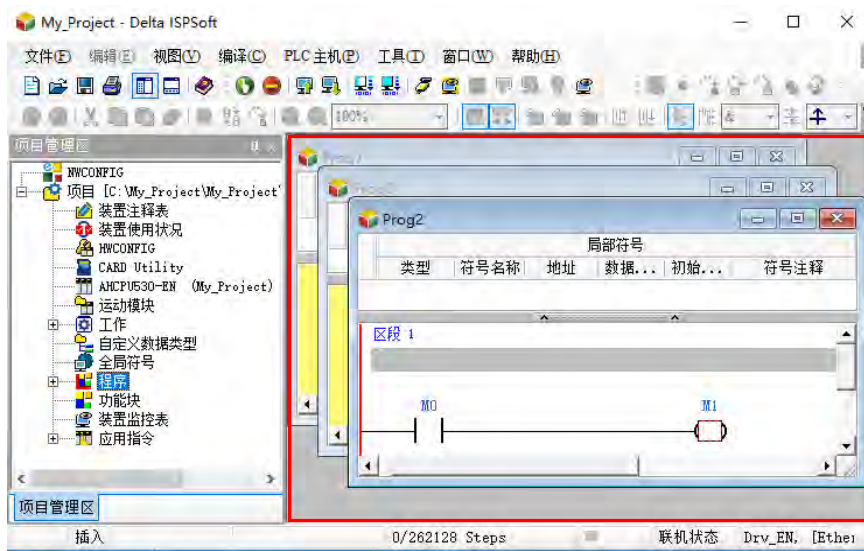


2.1.6 编辑工作区

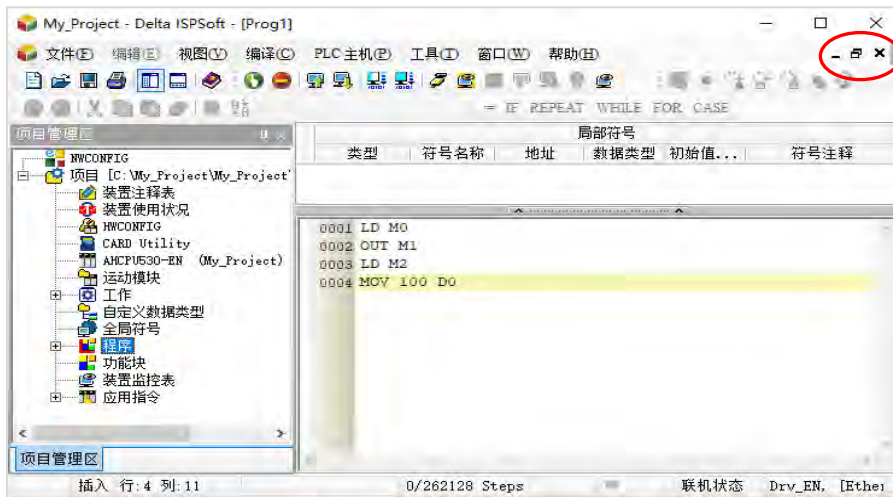
此区为各个工作窗口的所在位置，当将工作窗口设为最大化或最小化时，该窗口也会在此区域中进行缩放；而当工作窗口最大化时，该窗口的状态操作按钮则会显示在功能工具栏的右侧。

2

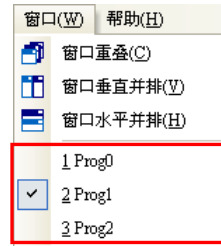
- 工作窗口陈列



- 工作窗口最大化



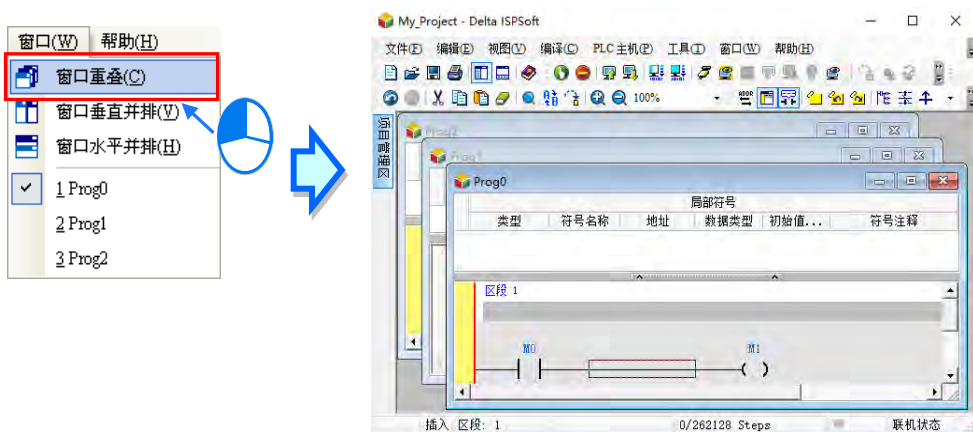
通过功能工具栏中的「窗口 (W)」功能项即可对这些工作窗口进行管理与切换。



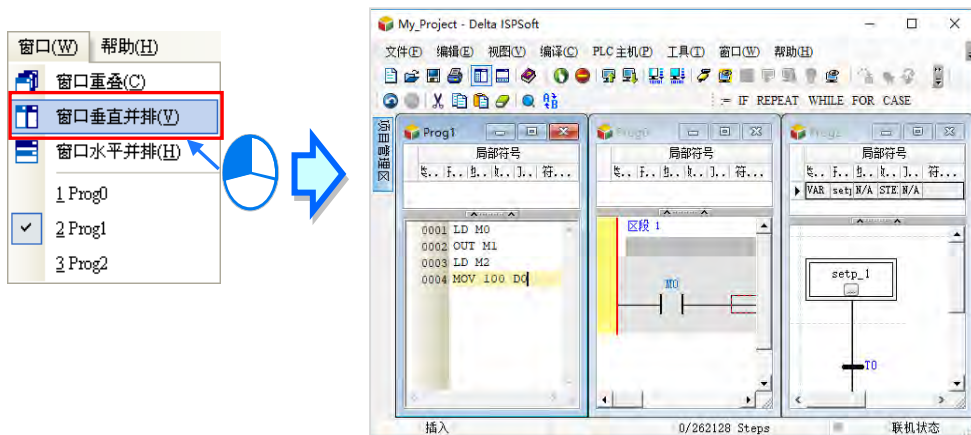
← 直接勾选以切换成目前的工作窗口

2

- **窗口重叠**：单击此项后，所有的窗口会以重叠的方式排列，其中目前的工作窗口会被排在最前方。

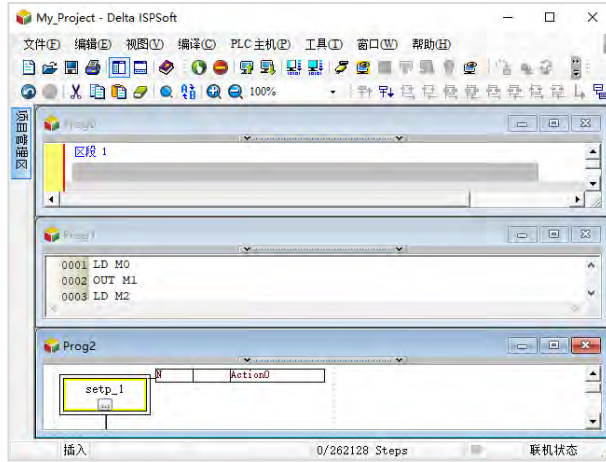
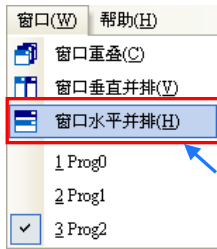


- **窗口垂直并排**：所有的窗口会以垂直对齐的方式并行排列，其中目前的工作窗口会被排在最左侧。



- 窗口水平并排：所有的窗口会以水平对齐的方式并行排列，其中目前的工作窗口会被排在最上方。

2



2.2 ISPSOft 的项目架构

2.2.1 单机项目与群组项目

ISPSOft 的项目架构共可分为单机项目与群组项目两种，说明如下。

● 单机项目

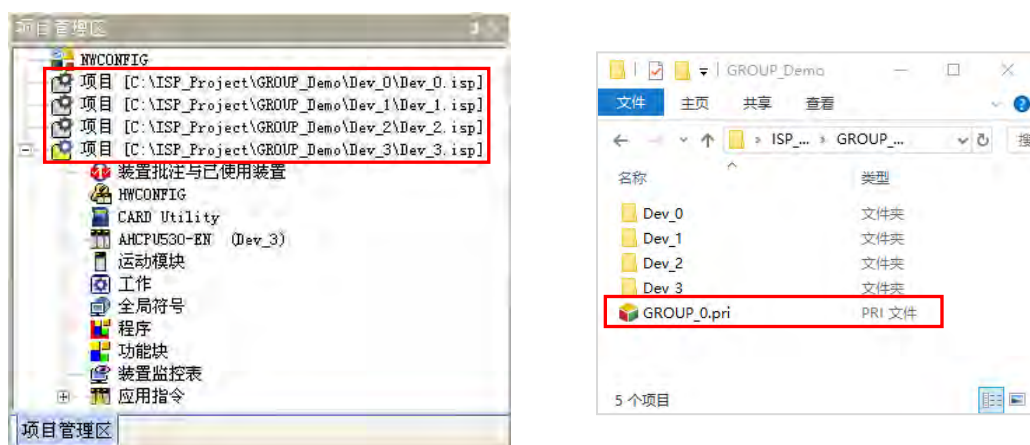
即最基本的项目类型，在每个项目中只可进行一个机种的开发，主要用于简易的单机应用，而项目文件的类型则为 *.isp。



● 群组项目

用于联网应用的多主机系统，在一个项目群组中可建立多个单机项目，且不限机种系列；一个群组中的项目数量并没有上限，用户可依照实际应用的情况加以规划，而通过 ISPSOft 中的网络规划工具 - NWCONFIG，即可简单且快速的对同一群组中的各个设备进行网络架构的规划。

群组项目的文件类型为 *.pri，而群组下的每个单机项目的文件类型则同样为 *.isp，且会放入与项目名称相同的文件夹中进行分类管理；对于先前已建立好的单机项目，后续亦可通过导入的方式来加入项目群组当中。

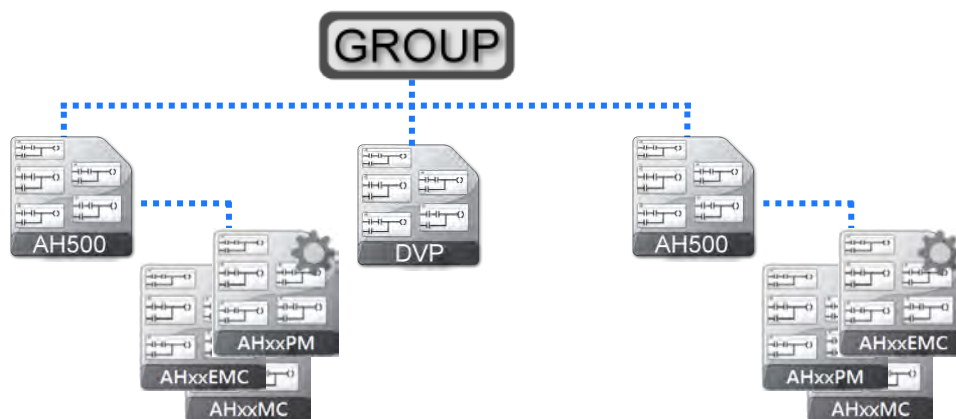


*.关于网络规划工具 - NWCONFIG 的相关说明将会在第 20 章的内容当中有详尽的介绍。

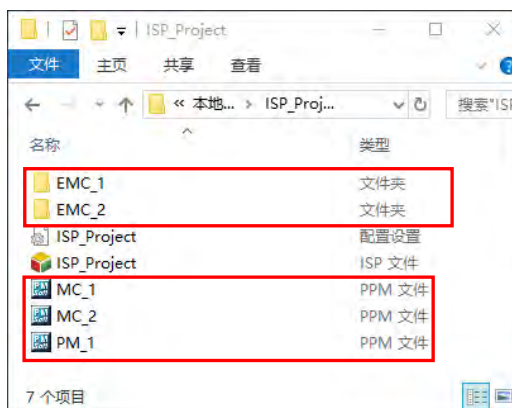
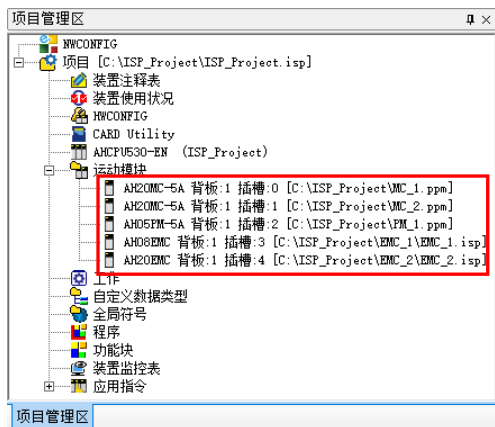
2.2.2 整合运动模块项目

AH系列系统提供AH05PM-5A、AH10PM-5A、AH15PM-5A与AH20MC-5A及AH08EMC、AH10EMC、AH20EMC等运动控制模块，且用户可于该模块内部布署独立的运动控制程序，而AHxxPM与AHxxMC其开发的软件为台达电子所推出的PMSOft，相关细节与使用方式请参考各机种与PMSOft的使用手册。AHxxEMC则是使用ISPSOft开发，可以当主机也可以当运动模块使用。

用户可直接在ISPSOft中建立运动模块项目，且可通过硬件规划工具 - HWCONFIG来进行相关的硬件配置与参数设置；而不分单机项目或是群组项目的架构，在每个AH500系列机种的项目中皆可建立一个以上的运动模块项目，还需要在HWCONFIG中的硬件配置增加对应的运动模块，相关细节将在第2.2.5节当中另行介绍。




AHxxPM与AHxxMC为PMSOft项目，文件类型为.ppm，而AHxxEMC则是ISPSOft项目，会有一个项目文件夹，文件类型为.isp，且会与AH机种的单机项目置于同一路径下，而对于先前已建立好的PMSOft项目或ISPSOft项目，则可通过导入的方式加入ISPSOft的项目中。

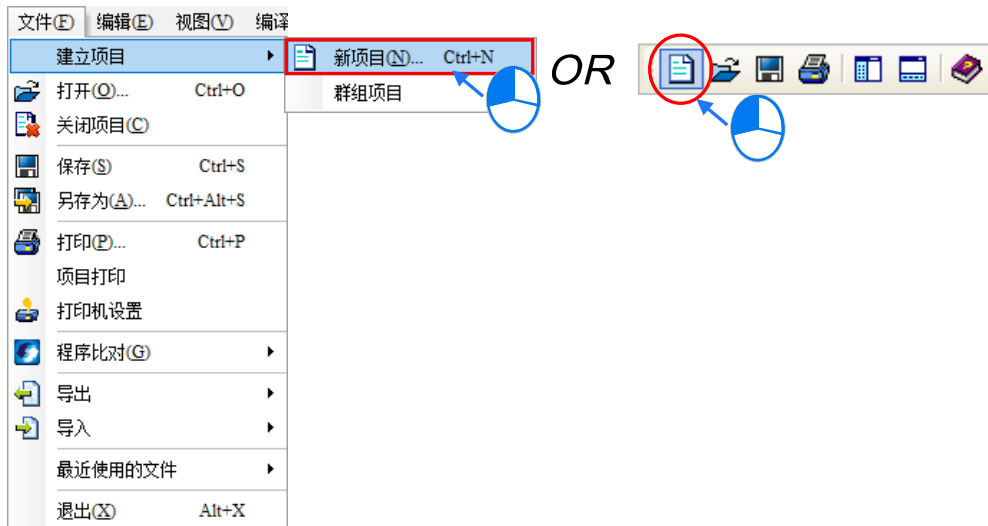


*1. 导入的PM项目必须为PMSOft V2.05版以上的文件格式，且不支持DVP-PM机种的项目。

2.2.3 ISPSOft 的单机项目管理

● 建立新单机项目

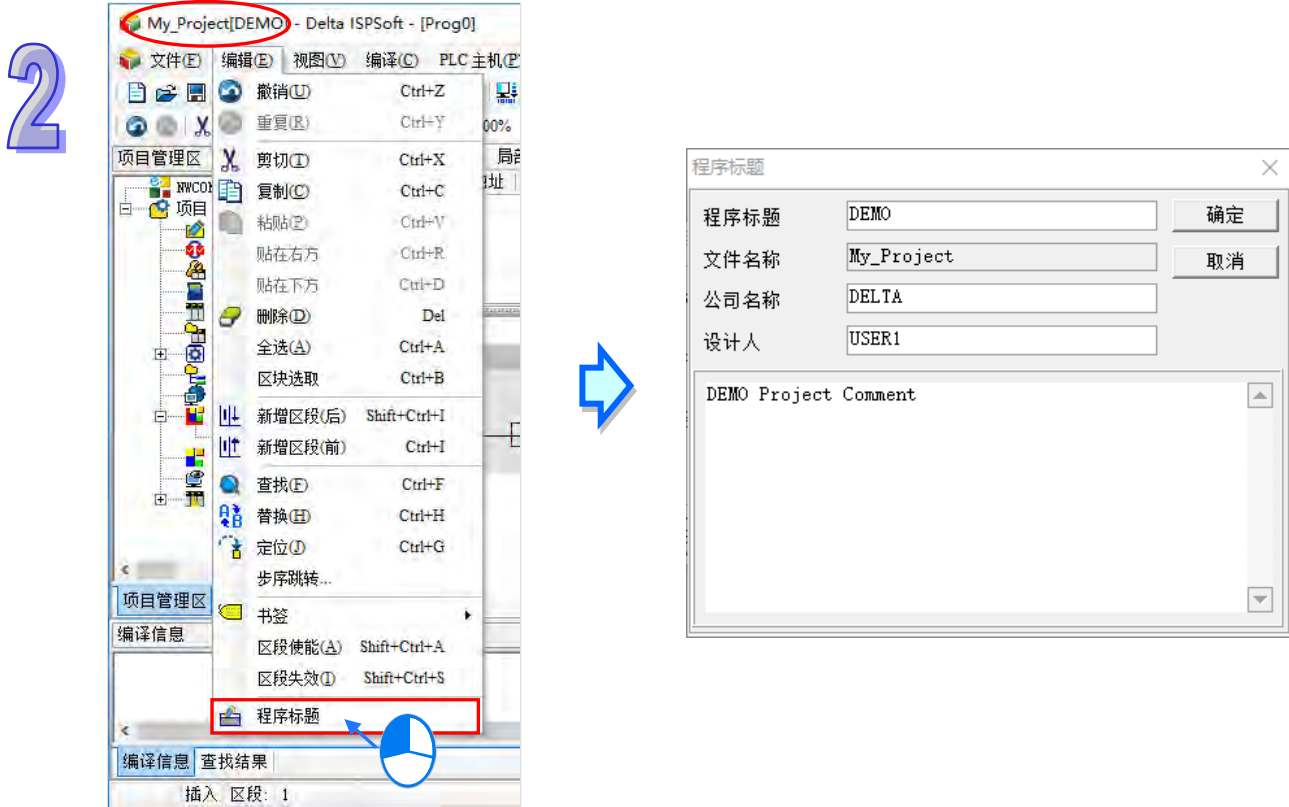
首先在功能工具栏中单击 **文件 (F) > 建立项目 > 新项目 (N)** 或点击图标工具栏的  图标。



接着在设置窗口中依序输入「项目名称」及「文件路径」，之后在「控制器种类」及「机种」的下拉选单中选择机种型式；而点击「属性」则可为此项目加入相关的信息及描述。完成后按下「确定」键即可。



完成新项目的建立后，在 ISPSOft 的窗口标题中便会显示项目的名称，格式为 **项目名称【程序标题】**，而当没有设置「程序标题」时，窗口标题则只会显示「项目名称」；当要查看「程序标题」等信息时，只要在功能工具栏上依序单击 **编辑 (E) > 程序标题** 即可。




在项目管理区中可见到项目文件的路径与所选的项目机种。而当所建立的为 AH 及 AS 系列的机种项目时，在机种类型的后方还会以括号显示该 PLC 的标签名称，且会以项目名称做为默认值，但之后仍可在 **硬件规划工具 - HWCONFIG** 中重新设置，详细说明将在第 3.1.4.1 节的内容当中另行介绍。



*. PLC 的标签名称仅有 AH、AS 系列、DVPxxMC/AS5xx 系列的机种才有，DVP 系列的机种则不会有此信息，而其主要的功能是用来作为设备识别，尤其是在联网系统的应用或是在进行联机操作时，此信息可用来确认实际的操作对象是否与我们所预期的相同。

● 保存项目

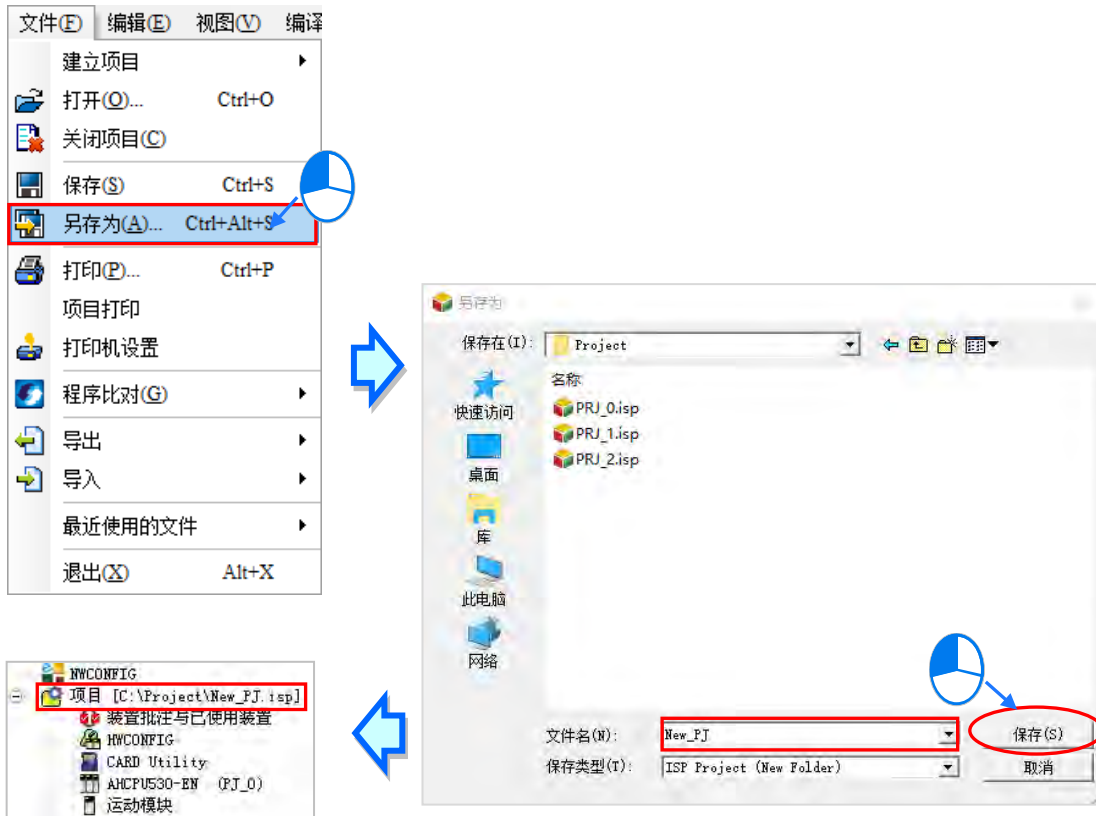
刚建立项目时，在指定的路径中并不会马上产生实际的项目文件，而必须在执行保存后才会真正的产生指定的文件。保存项目的操作方式如下。

在功能工具栏中单击 **文件 (E) > 保存 (S)** 或直接点击图标工具栏中的  图标即可。




若要以其它的名称或路径另外保存时，请确认 HWCONFIG、EthernetIP、EtherCAT 等相关编辑软件已经保存并关闭，接着于功能工具栏中单击 **文件 (E) > 另存为 (A)**，接着在文件窗口中设置新的路径及文件名，完成后按下「保存」按钮即可。

2



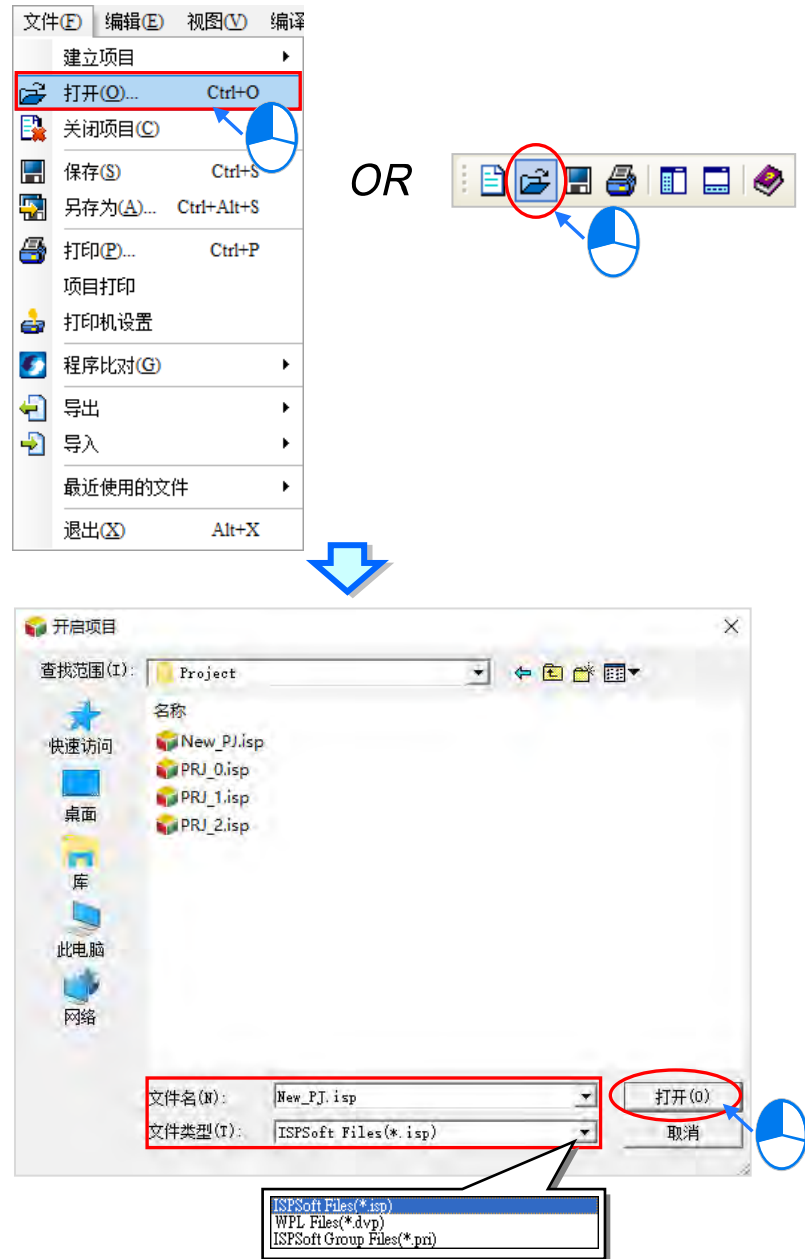
*.文件另存的操作并不会变更 PLC 的标签名称，因标签名称仅会在一开始建立新项目时以项目名称作为其默认值，但之后仍可在硬件规划工具 - HWCONFIG 中重新设置。

● 打开旧项目

若要在 ISPSOft 当中开启先前已建立的旧项目时，请在功能工具栏中单击 **文件(F)** > **打开(O)** 或直接点击图标工具栏中的  图标，接着在文件窗口的「文件类型」中下拉选择欲开启的文件类型，之后选取欲开启的文件后按下「**开启**」按钮即可。下表则为可开启的文件类型。

文件类型	说明
ISPSOft Files (*.isp)	此为 ISPSOft 的 单机项目文件 ，而若打开了旧格式的文件，则在保存项目时，ISPSOft 将会自动将其转换为新格式的项目文件。
WPL Files (*.dvp)	此为 WPLSoft 所建立的项目文件，而打开此类型的文件后，在保存项目时，ISPSOft 会自动将其转换为 .isp 的 ISPSOft 项目文件。
ISPSOft Group Files (*.pri)	此为 ISPSOft 的 群组项目文件 。

*.WPLSoft 亦为台达电子所推出的 PLC 编程软件，相关说明请参考 **WPLSoft 使用手册**。



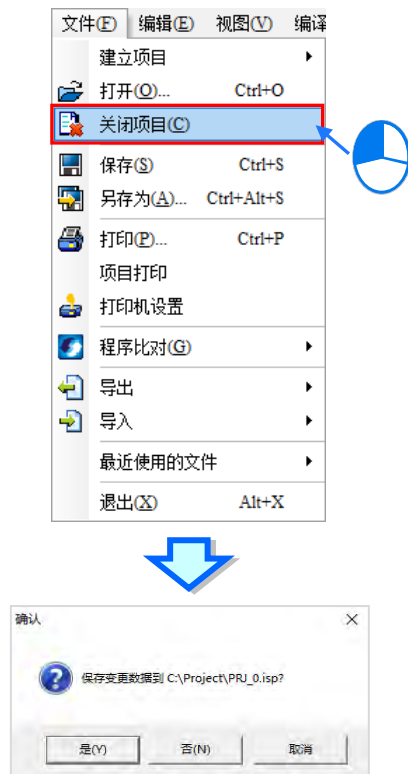
可在功能工具栏中直接点击「文件 (E)」项目下的项目清单中的「最近使用的文件」来开启最近编辑过的项目。

2



● 关闭项目

若已完成项目的编辑时，请在功能工具栏中单击 **文件 (F) > 关闭项目 (C)** 即可将项目关闭，而若尚未完成保存时，系统便会出现提示窗口询问是否要进行保存。



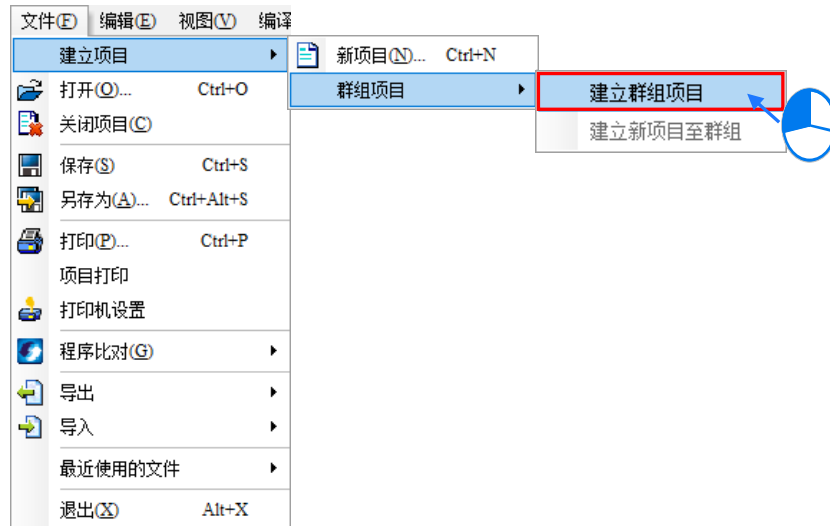
补充说明

若离开 ISPSOft 之前未先将项目关闭，则下次再启动 ISPSOft 时，系统默认将自动打开上次所编辑的项目。(自动打开前次项目的功能可由用户自行设置启动与否，相关细节请参考第 2.3.1 节。)

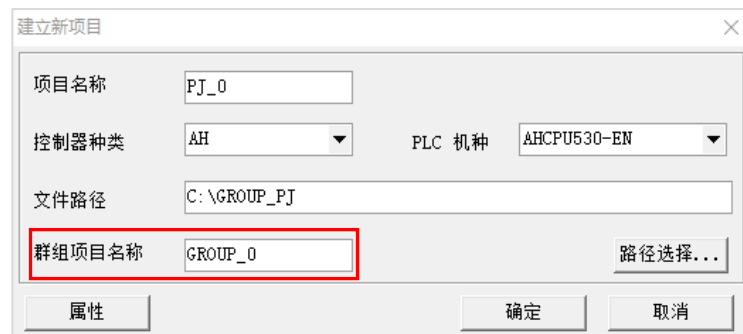
2.2.4 ISPSoft 的群组项目管理

● 建立新群组项目

首先在功能工具栏中单击 **文件 (F) > 建立项目 > 群组项目 > 建立群组项目**。

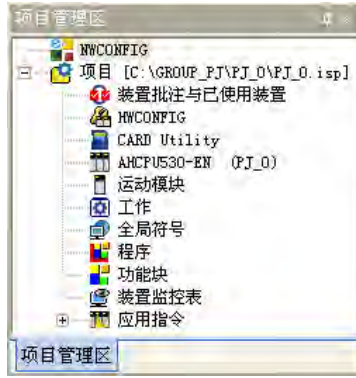


而建立新群组时必须连带建立该群组下的第一个单机项目，其设置方式则与建立**单机项目**时相同，分别在设置窗口中依序输入「项目名称」及「文件路径」，在「控制器种类」、「PLC 机种」的下拉选单中选择机种型式，而点击「属性」则可为此项目加入相关的信息及描述；除此之外，还必须在「**群组项目名称**」的字段中输入群组名称，完成后按下「**确定**」即可。



完成群组项目的建立后，在**项目管理区**中便会显示连带建立的单机项目，且因一开始建立时，群组中仅有一个单机项目，所以其显示与单独建立**单机项目**时相同，而所有相关的文件操作也都与在**单机项目**架构下的方式无异，唯一不同的是，在**群组项目**的架构下无法执行**另存文件**的功能。

2

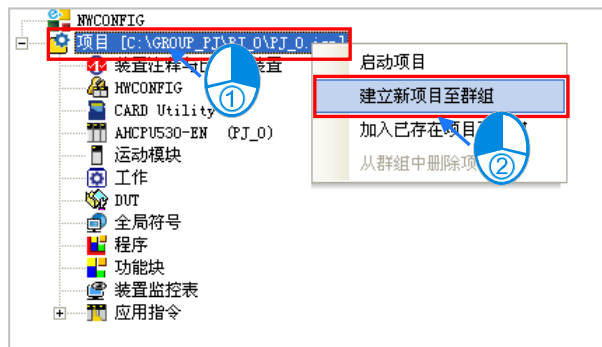


● 新增单机项目至群组

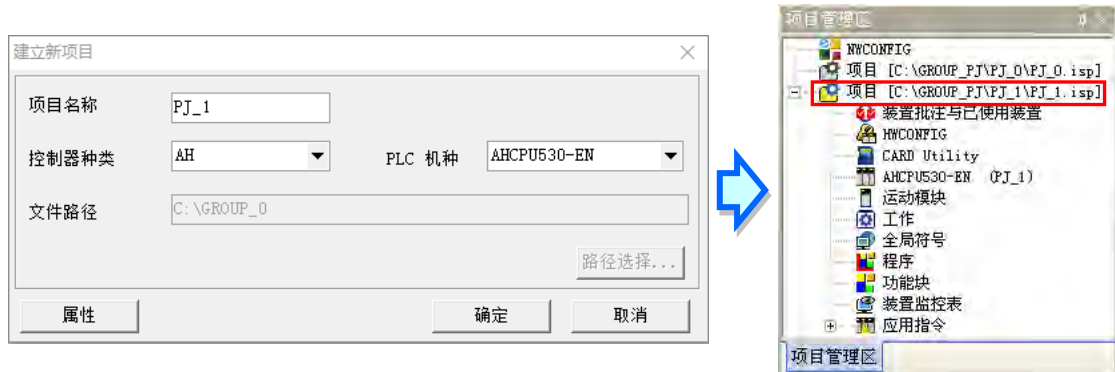
欲新增其它单机项目至群组中时，请在功能工具栏中依序单击 **文件(F)** > **建立项目** > **群组项目** > **建立新项目至群组**，亦或是在项目管理区中单击任一项目后按下鼠标右键，之后在快捷选单中单击**建立新项目至群组**。



OR

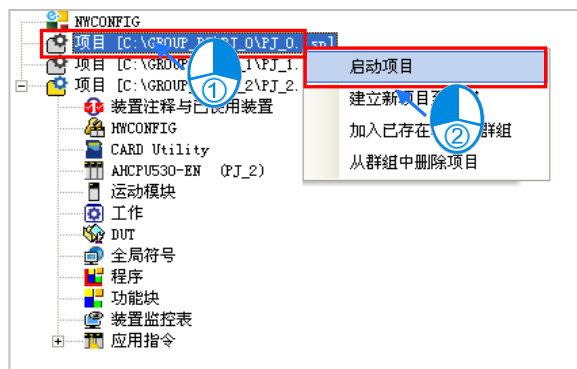


因在群组项目的架构下，同时间只能有一个单机项目处于编辑状态，因此当要新增一个项目至群组中时，原来正在编辑的项目便会被要求关闭，而此时若该项目尚未保存时，系统便会出现确认信息。新项目的设置窗口与一般建立新单机项目时相同，唯一不同的是无法设置文件路径。

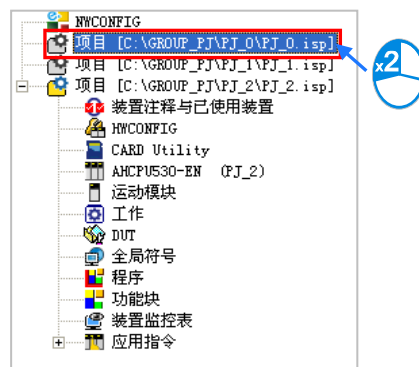


● 启动项目

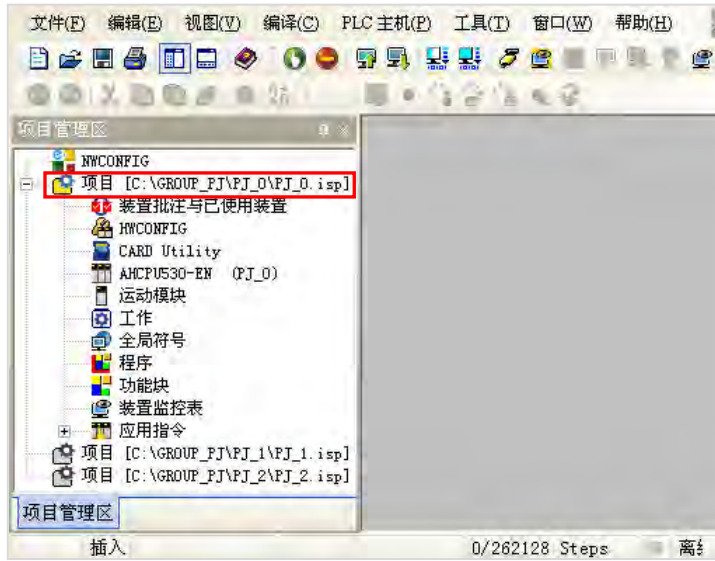
在群组项目的架构下，同时间只能有一个单机项目处于编辑状态，而要切换编辑的项目时，只要在项目对象上点击鼠标右键，之后在快捷选中单击 **启动项目**，或是直接在项目对象上双击鼠标左键即可。



OR



切换项目后，项目管理区中可编辑的项目会变成刚刚选取的项目，其余处于关闭状态的项目则会以灰阶的图标显示。

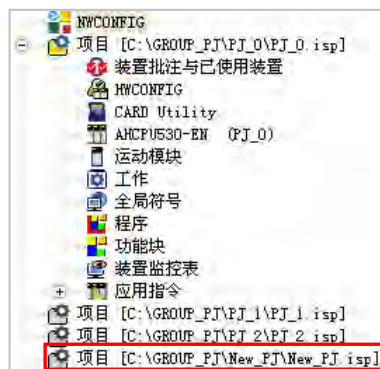
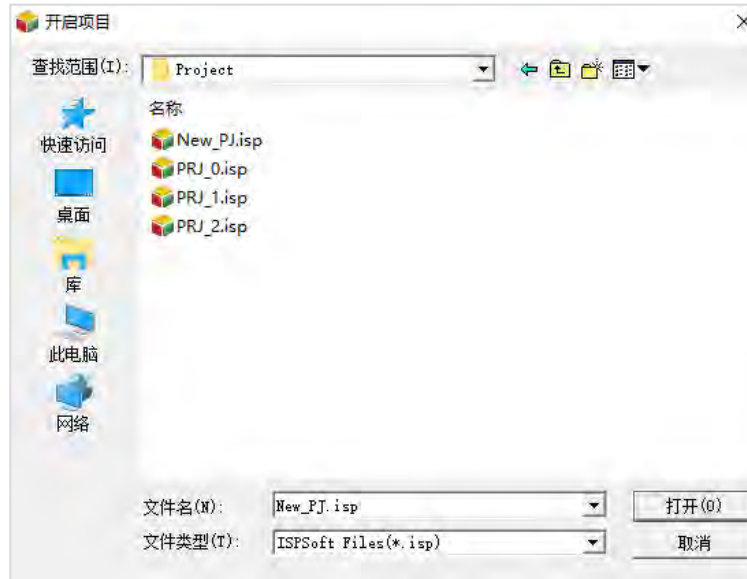


● 导入单机项目文件至群组

若有某个单机项目文件 (*.isp) 是在非群组项目的架构下所建立的，通过导入的方式亦可将其加入项目群组当中；欲进行导入时，请在项目管理区中单击任一项目后按下鼠标右键，之后在快捷选单中单击 **加入已存在项目至群组**，接着便可在文件窗口中选取欲导入的项目文件。



允许导入的文件类型只可为 *.isp 的单机项目文件，而导入项目文件时，系统仅会将该单机项目文件复制至项目分组的目录下，而原单机项目文件则仍会被保留在原来的路径中。

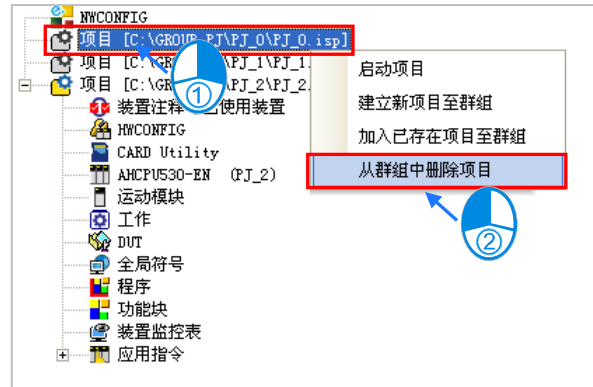


2

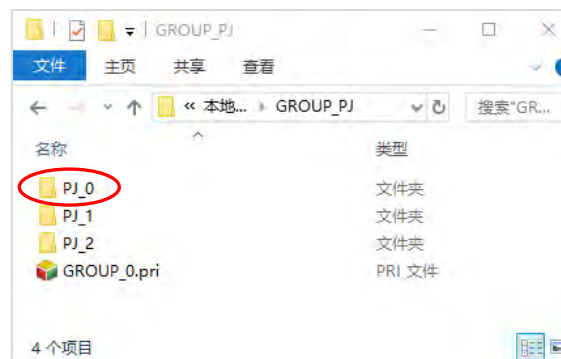
● 删除群组中的单机项目

若要将某个单机项目从群组中删除时，请在**项目管理区**中单击欲删除的项目后按下鼠标右键，之后在快捷选单中单击**从群组中删除项目**；不过若该群组中仅剩一个项目时便无法删除。


2

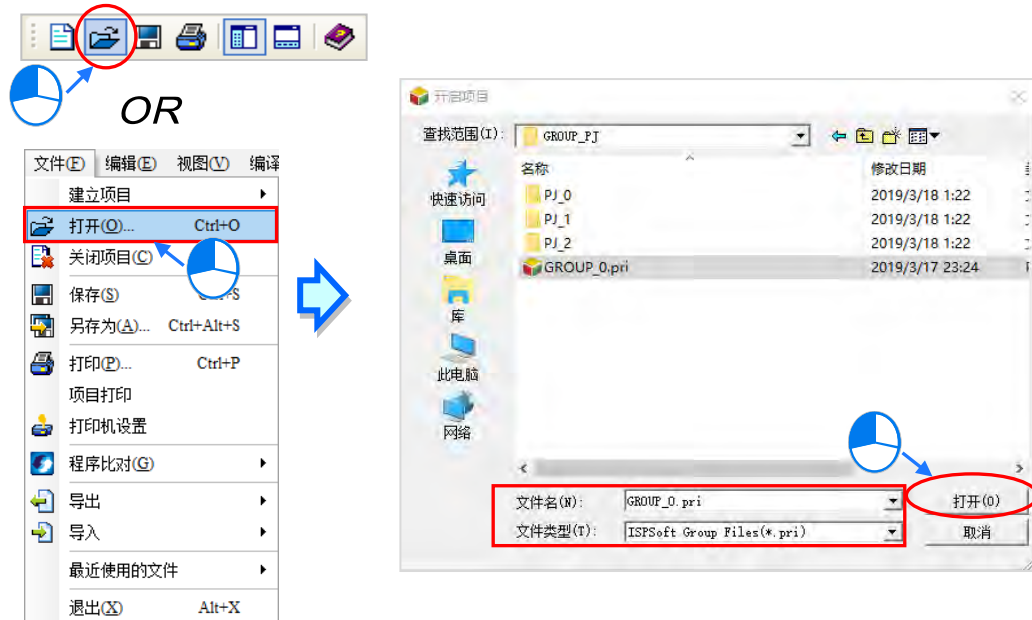


此操作仅会将该项目在群组项目的管理列表中删除而已，实际存放在原路径的**单机项目文件**并不会被删除，而之后若要重新将该项目加入群组时，只要通过导入的方式重新将其加入即可。

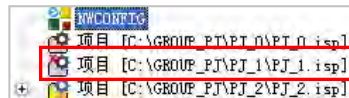


● 开启群组项目

若要开启之前建立的群组项目时，请在功能工具栏中单击 **文件 (F) > 打开 (O)** 或直接点击图标工具栏中的  图标，接着再在文件窗口的「文件类型」下拉选择 **ISPSoft Group Files (*.pri)**，之后选取欲开启的文件后按下「打开」按钮即可。



而若打开的群组项目中有出现红色叉号的图标时，代表位于此群组路径下的该**单机项目文件**出现异常，有可能已被删除、变更路径或是变更名称。



● 关闭群组项目

在功能工具栏中单击 **文件 (F) > 关闭项目 (C)** 可将目前正在编辑的单机项目关闭，但此时仍可再启动其它处于关闭状态的单机项目，而若要关闭整个群组项目时，请单击 **关闭全部项目**。



2.2.5 ISPSOft 的运动模块项目管理

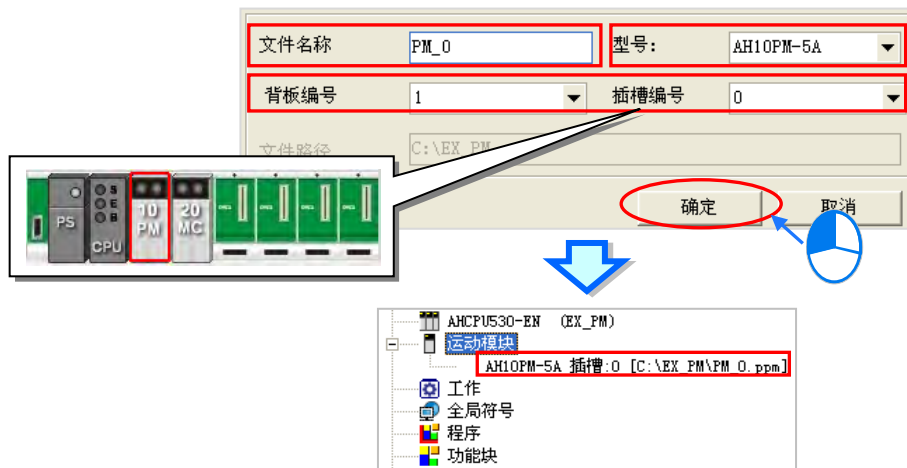
在 ISPSOft 的项目管理区中，针对 AH 系列机种的项目会多出一个「运动模块」的目录，若用户要增加 AHxxPM 与 AHxxMC 运动模块项目，则必须要安装 PMSOft V2.05 版以上的软件。

- 建立新 AHxxPM 与 AHxxMC 运动模块项目

要在 ISPSOft 中建立 PMSOft 项目时，请在项目管理区中单击「运动模块」后按下鼠标右键，之后在快捷选单中单击 建立运动模块。



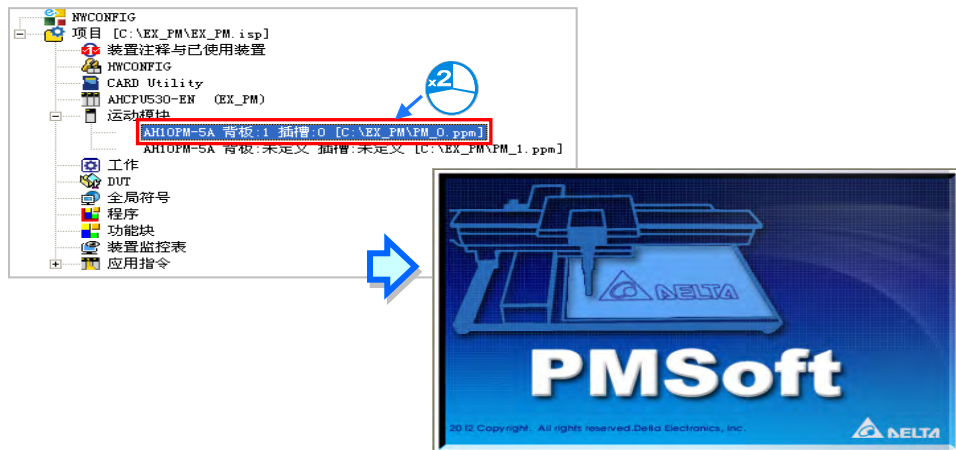
接着请在设置窗口的「文件名称」中输入运动模块项目的名称，在「型号」字段下拉选择模块型号，之后依据 HWCONFIG 中的配置，在「背板编号」及「插槽编号」字段中下拉选择该模块对应的插槽编号，且每个插槽编号仅可对应一个运动模块项目，而若尚未确定插槽编号亦或是之后要重新调整插槽时，可先在下拉选单中选择「未定义」。完成设置之后按下「确定」即可。



2

● 开启 AHxxPM 与 AHxxMC 运动模块项目

当要对已建立好的 AHxxPM 与 AHxxMC 运动模块项目进行编辑时，请直接在该项目对象上双击鼠标左键，之后系统便会自动开启 PMSoft，而用户后续即可直接在 PMSoft 当中进行该运动模块的程序开发。

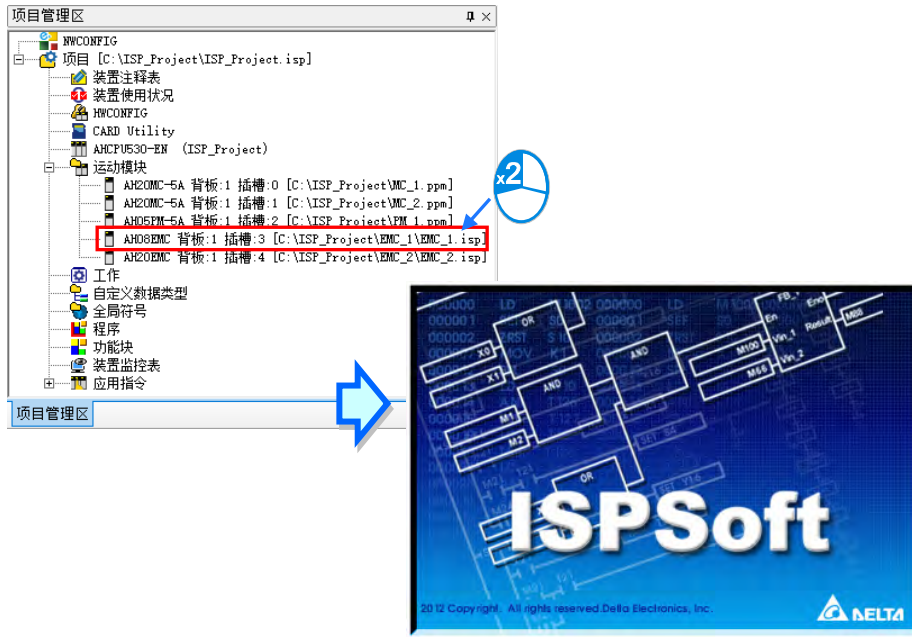


*.关于 PMSoft 的操作请参考 PMSoft 使用手册，而通过 ISPSOft 启动的 PMSoft，只可以针对该运动模块编辑，且编辑完成后仍必须于 PMSoft 中执行「保存」功能才可完成运动模块编辑。

● 开启 AHxxEMC 运动模块项目

当要对已建立好的 AHxxEMC 运动模块项目进行编辑时，请直接于该项目对象上双击鼠标左键，之后系统便会自动开启 ISPSOft，而用户后续即可直接于 ISPSOft 中进行该运动模块的程序开发。

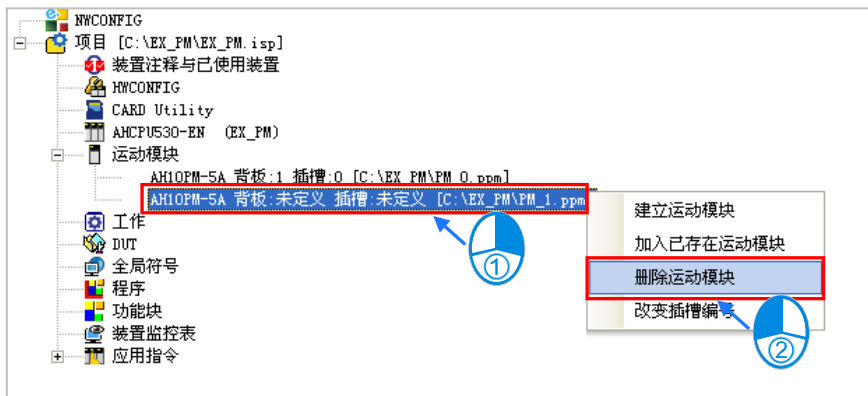
2



*.通过 ISPSOft 启动的 ISPSOft，只可以针对该运动模块编辑，且编辑完成后仍必须于 ISPSOft 中执行「保存」功能才可完成运动模块编辑。

● 删除运动模块项目

欲删除已建立好的运动模块项目时，请先单击欲删除的项目后按下鼠标右键，之后在快捷选单中单击删除运动模块项目；而删除运动模块项目时，系统仅会将该项目由项目管理清单中删除，实际存放的运动模块项目文件 (*.ppm) 或 (*.isp) 则仍会保留在原来的路径中。



● 导入运动模块项目

要导入先前所建立的运动模块项目文件 (*.ppm) 或 (*.isp) 时，请先在项目管理区中单击「运动模块」项目后按下鼠标右键，之后在快捷选单中单击 加入已存在运动模块。



2

接着请在文件窗口中选取欲导入的运动模块项目文件，完成后按下「打开」即可。而导入的运动模块项目，默认的插槽编号将为未定义。

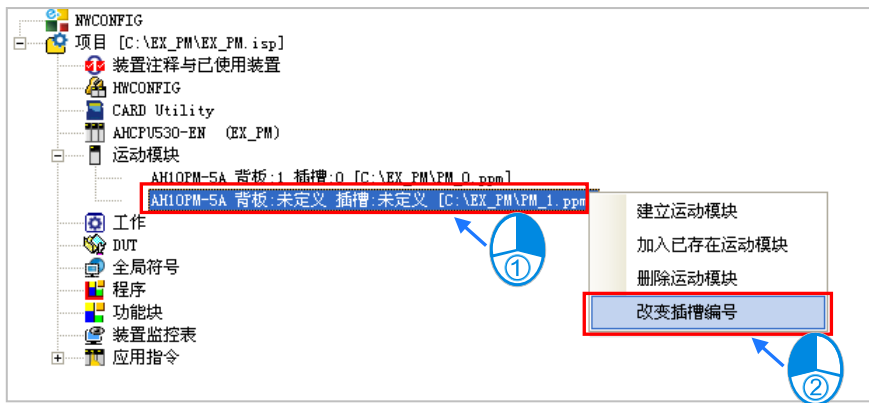


导入 PM 项目时，系统仅会将该 **PMSoft** 项目文件复制至 ISPSoft 的项目路径下，而原始的 **PMSoft** 项目文件则仍会被保留在原来的路径中；导入的 **PMSoft** 项目文件必须为 PMSoft V2.05 版以上的文件格式，且不支持 DVP-PM 机种的项目。导入 ISP 项目文件时，系统会将该 **ISPSoft** 项目档复制至 ISPSoft 的项目路径下并建立一个文件夹。

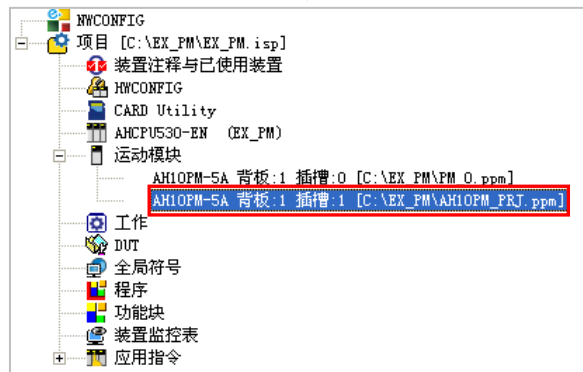
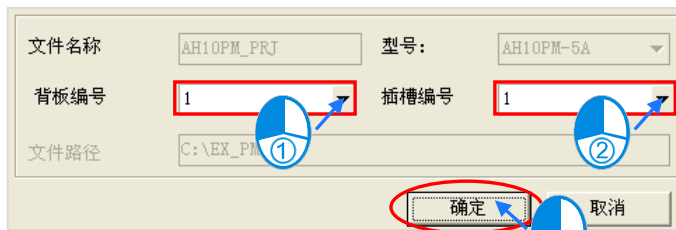
● 改变插槽编号

欲变更运动模块项目所对应的插槽编号时，请先单击该模块的项目后按下鼠标右键，之后在快捷选单中单击 **改变插槽编号**。

2



接着请在设定视窗的「背板编号」与「插槽编号」栏位下拉选择欲变更的编号，而选择时请注意不可选用已被占用的插槽；完成后按下「确定」即可。



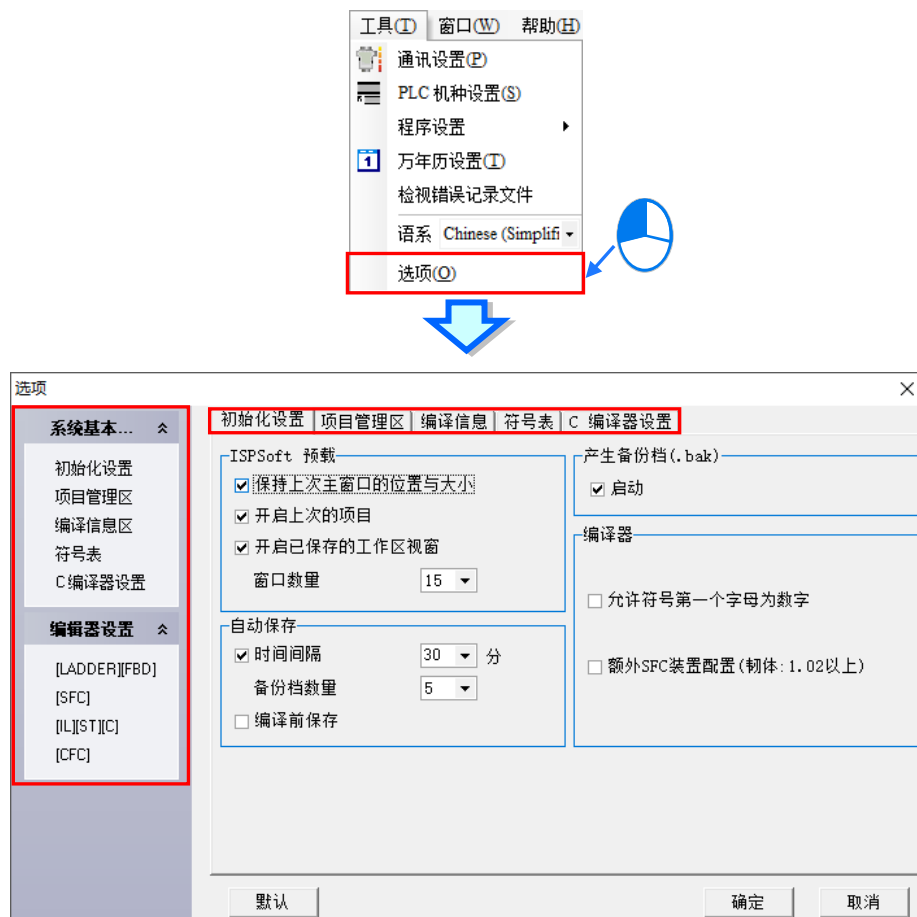
2.3 ISPSOft 中的基本设定

2.3.1 系统及环境设定

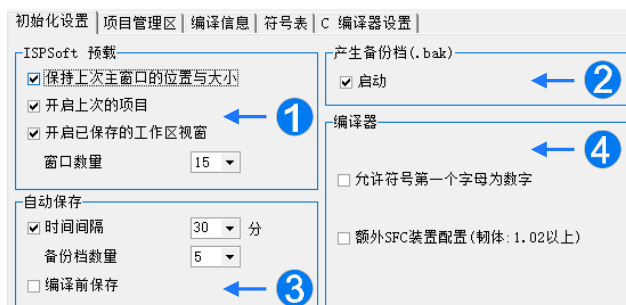
在功能工具栏中单击 **工具 (I)** > **语系**，之后在下拉选单中便可选择欲使用的操作语系。



在功能工具栏单击 **工具 (I)** > **选项 (O)** 之后便会出现操作环境的设置窗口，而通过窗口上方或左侧的选项卡便可切换各个页面。

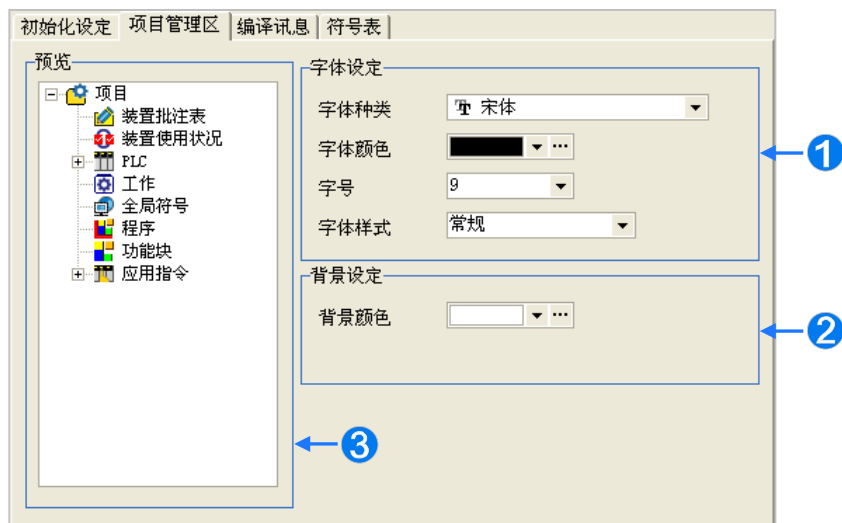


● 系统基本设定 - 初始化设定



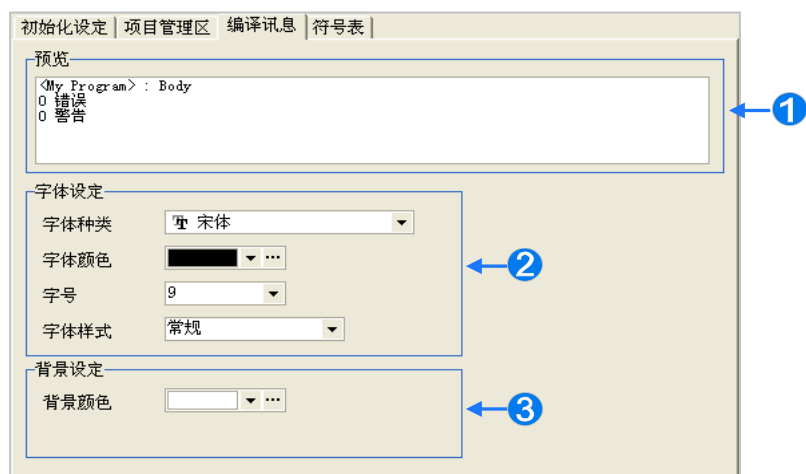
- ❶ 保持上次主窗口的位罝及大小**：启动 ISPSOft 时自动开启最后一次编辑的项目。
- 开启上次的项目**：自动存储最后一次编辑的主窗口位罝及大小，使用双屏幕编辑的用户可以取消此项目，以使主窗口每次都会出现在主屏幕，可以避免无第二个屏幕时，主窗口还停留在第二个屏幕的问题。
- 开启已储存的工作区窗口**：启动 ISPSOft 时自动开启最后一次存盘未关闭之工作区窗口，也可透过**窗口数量**选取要显示的窗口数量，最多为 15 个。
- ❷ 产生备份档启动**：存档时会自动产生一个备份文件。
- ❸ 时间间隔**：依照右侧的字段设定的时间，系统便会自动将目前所编辑的项目按照系统的时间存成项目档，用户可由「备份档数量」设定需要最大的备份数量，若自动存档超过最大数量，将依序覆盖最早的存档；若 ISPSOft 发生不正常关闭，则下次再开启该项目文件时，系统便会询问用户是否要加载最后备份的编辑内容。
- 编译前保存**：每次编译前会自动保存目前编辑的项目档。
- ❹ 允许符号表中地址字段的装置名称重复**：允许不同的变量符号指定到同一个装置地址。
- 允许符号第一个字母为数字**：允许变量符号名称为数字开头。
- 旧版变数阵列编译**：支持旧版变量数组编译。
- *此项目可能随着机种不同会有不同。

- 系统基本设定 - 项目管理区



- ① 设置项目管理区中，各对象的显示字型。
- ② 设置项目管理区的背景底色。
- ③ 此处显示的为变更后的预览画面。

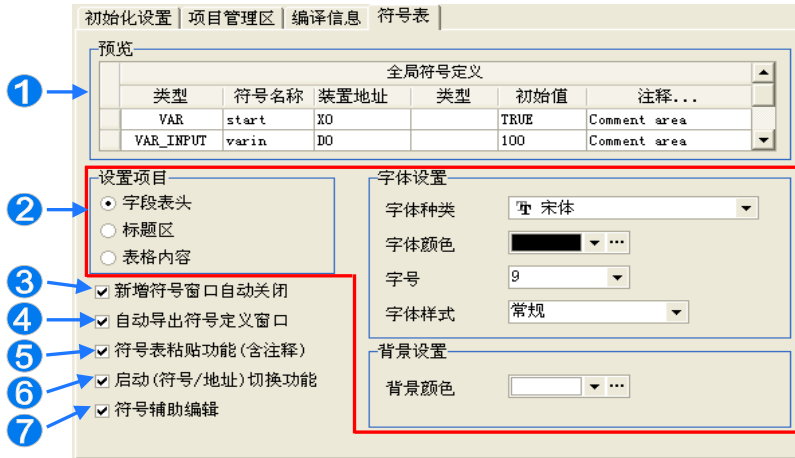
- 系统基本设定 - 编译信息




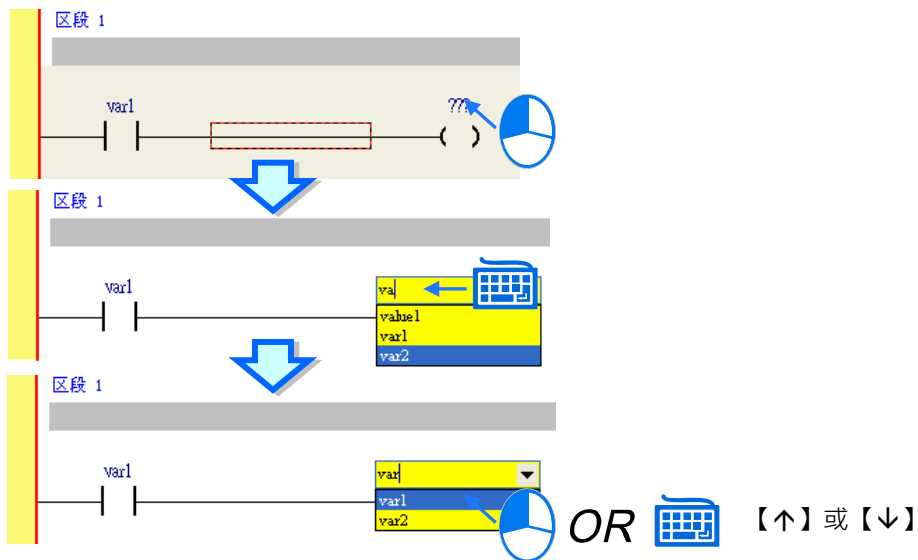
- ① 此处显示的为变更后的预览画面。
- ② 设置信息的显示字型。
- ③ 设置信息显示区的背景底色。

● 系统基本设定 - 符号表

2



- ① 此处显示的为变更后的预览画面。
- ② 先在左侧选择设置项目后，再在右侧字段中设置该项目的显示样式。
- ③ 符号宣告窗口在完成宣告后的状态会自动关闭。
- ④ 在 LD、FBD 或 SFC 等环境中输入未定义的变量符号，并在输入完毕后直接按下【Enter】键，系统将会自动导出符号宣告窗口。
- ⑤ 符号表之间进行复制时，粘贴的内容是否要包含变量符号的批注。
- ⑥ 启动图标工具栏的地址模式按钮  才可以切换地址模式或位址模式。
- ⑦ 当在程序中输入变量符号时，会自动列出所有符合的已建立变量符号项目，用户可直接单击鼠标左键或使用键盘上下键由列表中挑选，如下图。

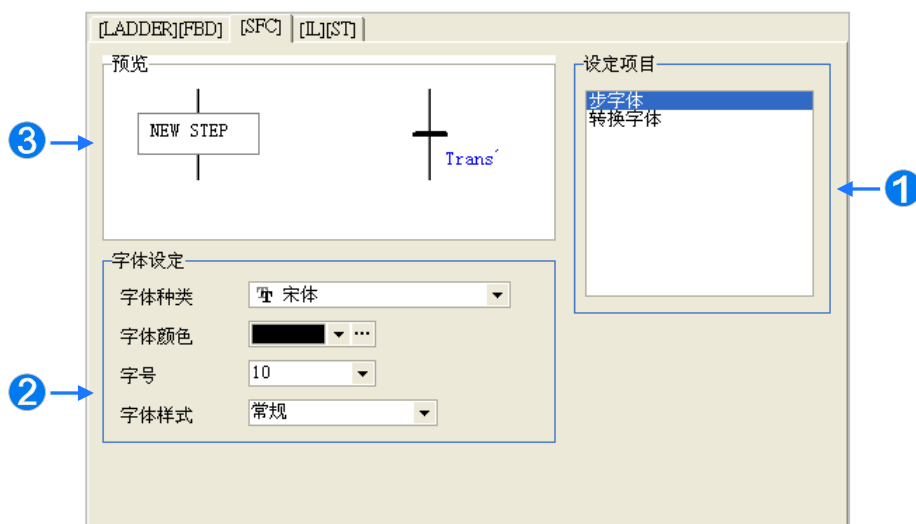


● 编辑器设定 - LADDER 及 FBD



- ① 先在此处选择设置对象。
- ② 此处为变更后的预览画面
- ③ 在此处设置所选对象的显示样式。
- ④ 在 LD 及 FBD 的程序编辑区中，当鼠标停留在某个装置或变量符号上时，画面将会显示该装置或变量符号的提示，而在此便可勾选将会显示的提示内容。
- ⑤ 在此处设置 LD 及 FBD 中的组件显示宽度，而此设置将会影响编程环境中，功能块图形内部可显示的字数长短。

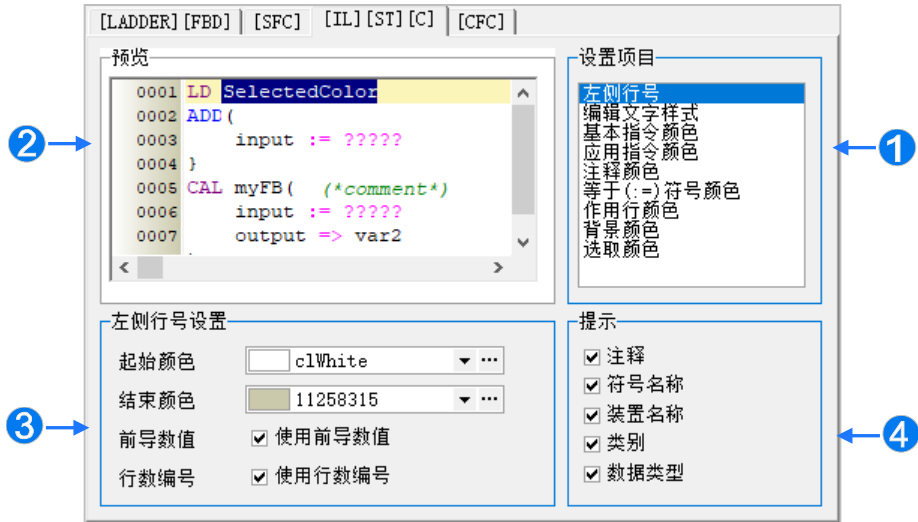
● 编辑器设定 - SFC



- ① 先在此处选择设置对象。
- ② 在此处设置所选对象的显示样式。
- ③ 此处显示的为变更后的预览画面。

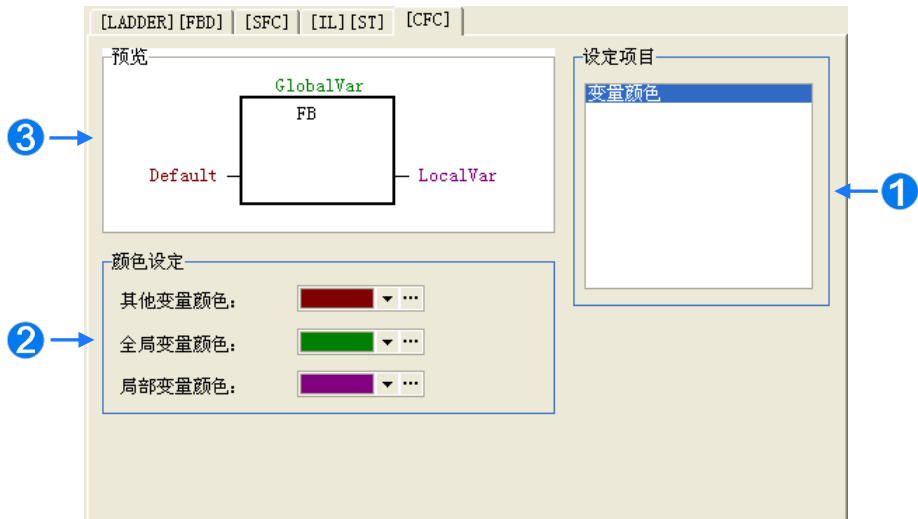
● 编辑器设定 - IL 及 ST 和 C

2



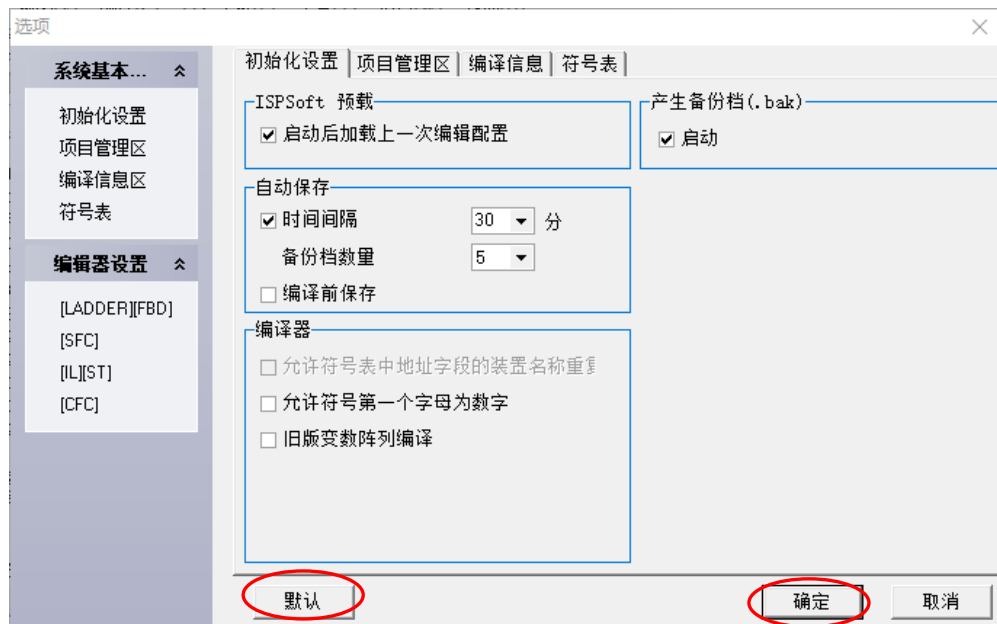
- ① 先在此处选择设置对象。
- ② 此处显示的为变更后的预览画面。
- ③ 于此设定所选项目的显示样式。
- ④ 在 IL 及 ST 的程序编辑区中，当鼠标停留在某个装置或变量符号上时，画面将会显示该装置或变量符号的提示，而于此便可勾选将会显示的提示项目。

● 编辑器设定 - CFC



- ① 先在此处选择设定项目。
- ② 在此处设定所选项目的显示样式。
- ③ 此处显示的为变更后的预览画面。

当按下「默认」按钮时，所有的设置值便可恢复为系统最初的默认值，而完成所有的设置之后，按下「确定」按钮即可应用所做的变更。



2

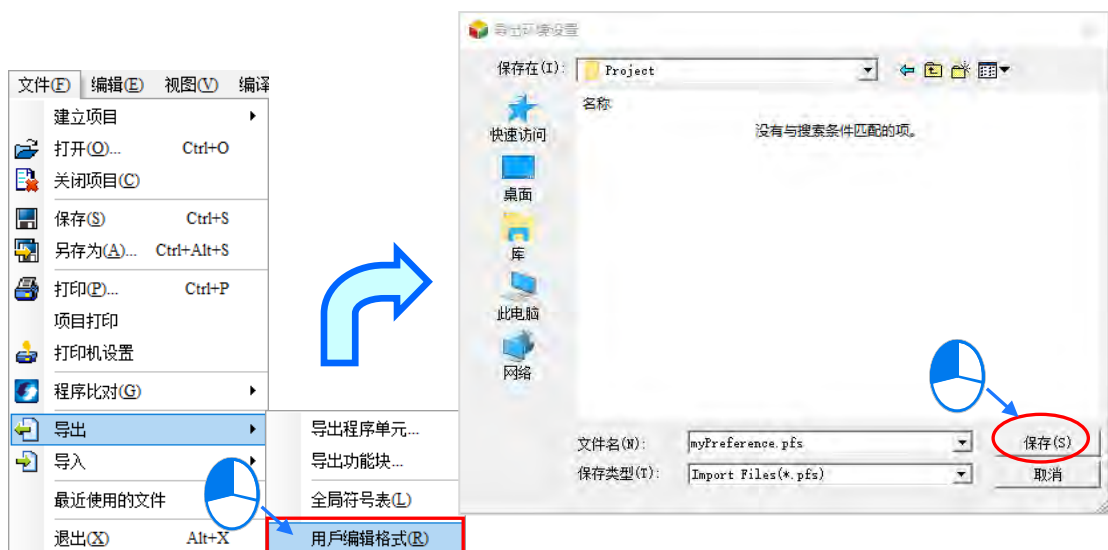
若同时开启多个 ISPSOFT 窗口时，不同的 ISPSOFT 窗口可做不同的设置，不过系统只会记忆最后一个关闭窗口的环境设置，因此当下次重新启动 ISPSOFT 时，系统便会自动应用最后一个关闭窗口中的设置值。

2.3.2 用户编辑格式的导入与导出

通过 ISPSOFT 所提供的功能，用户可将自己设置好的环境设置进行导出与导入的动作。

● 导出用户编辑格式

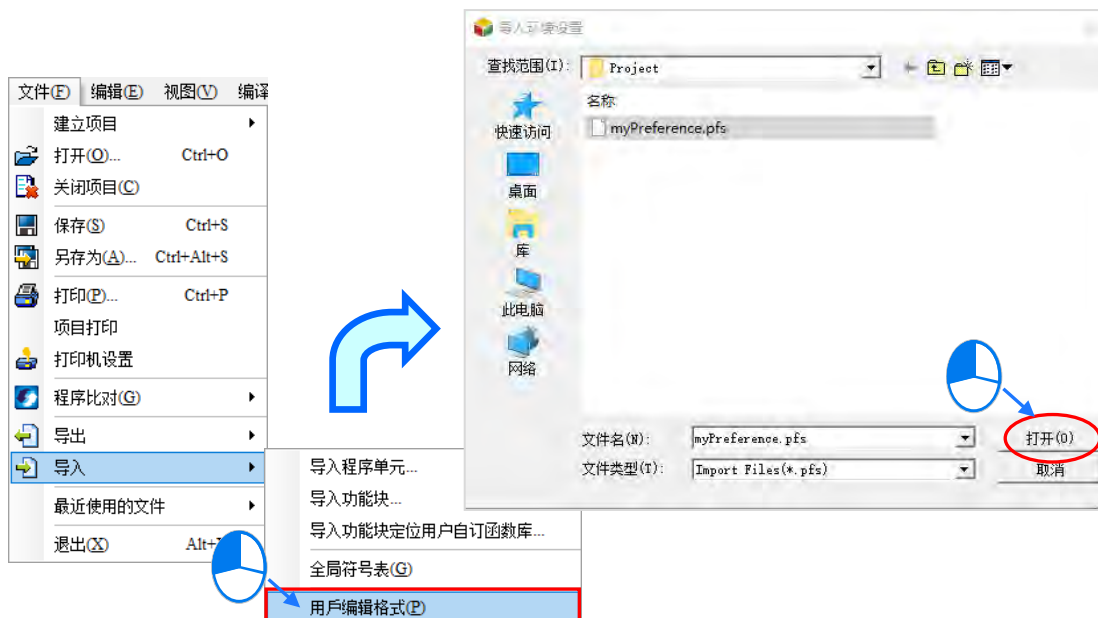
在功能工具栏中单击 **工具 (I) > 导出 > 用户编辑格式 (R)**，之后在文件窗口中输入欲导出的文件名后按下「保存」即可。



● 导入用户编辑格式

在功能工具栏中单击 **工具 (I)** > **导入** > **用户编辑格式 (P)**，之后在文件窗口中选择欲导入的文件后按下「开启」即可。

2



补充说明

不同版本的 ISPSOft，因其环境参数可能会有所不同，所以导入时不见得可以完全套用，因此若不定所导入的**用户编辑格式档 (*.pfs)**是由哪个版本的 ISPSOft 所汇出的，导入后请再次确认所导入的内容是否与预期相符。

2.3.3 导出程序到文字格式

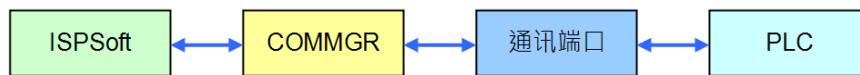
透过 ISPSOft 所提供的功能，用户可将程序导出到文字格式以便于申请专利，导出的文字格式无法再转回程序，于功能工具栏中点选 **工具 (I)** > **导出程序到文字格式**，之后于档案窗口中输入欲导出的档名后按下「存档」即可，如下图。



2

2.4 通讯设定

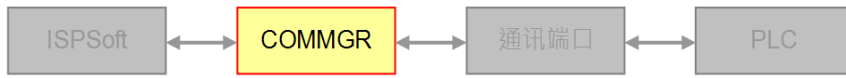
下图为 ISPSOft 与台达 PLC 之间的通讯架构，有别于以往的联机方式，ISPSOft 则是统一通过**通讯管理员 - COMMGR**来做为通讯的接口。而在此节的内容中，我们将会介绍如何在 ISPSOft 与 PLC 之间逐步地建立起正常的联机通讯，并完成基本的测试。



*1. 关于通讯管理员 - COMMGR 的简介与安装方式，请参考手册第 1.2 节的相关说明。

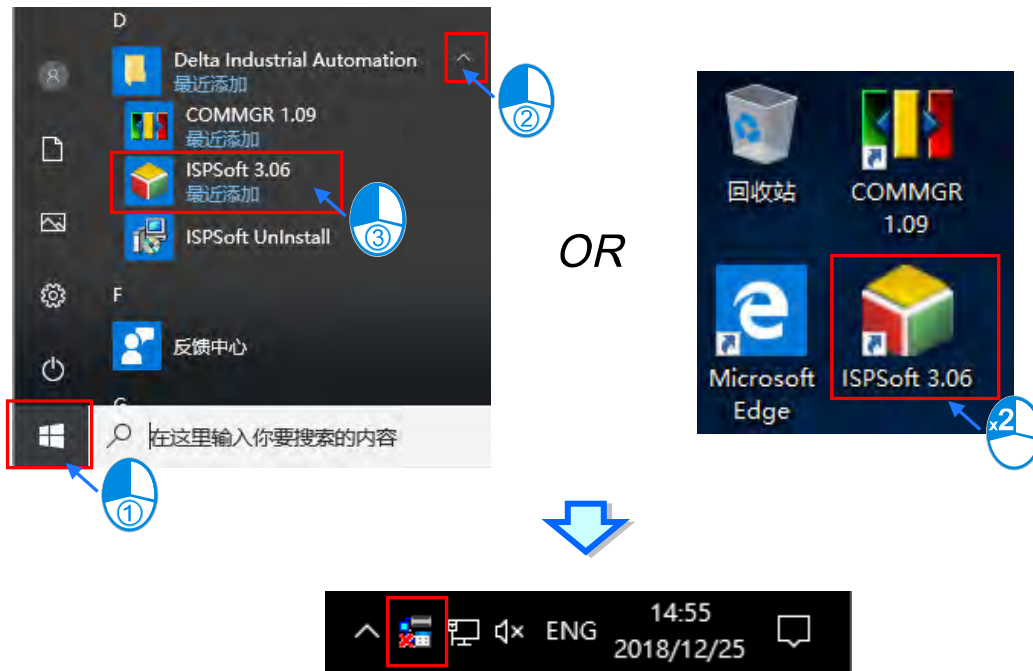
*2. ISPSOft 在 V2.0 版以上才开始采用 COMMGR 的通讯架构，较早期的版本则仍使用传统的联机方式。

2.4.1 通讯管理员 COMMGR 的启动与关闭



2

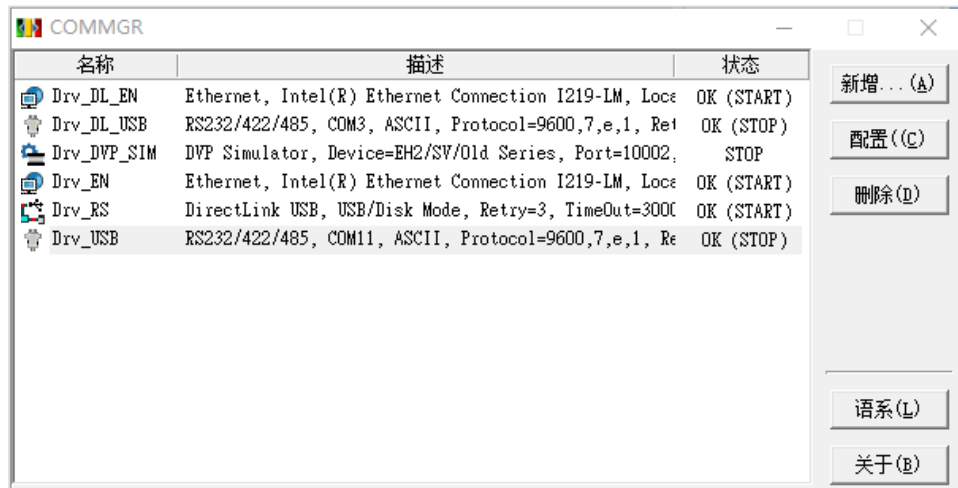
顺利完成 **COMMGR** 的安装后，系统便会自动在程序菜单当中建立一个执行快捷方式，直接点击该快捷方式即可启动 **COMMGR**；而除了首次安装完成后，用户需自行将其启动外，之后只要每次计算机重新开机，系统便会自动启动 **COMMGR**，并将其常驻在 Windows 的右下角；但若 Windows 的右下角没有出现该图标时，请直接由程序菜单当中自行启动 **COMMGR**。



当顺利启动 **COMMGR** 之后，在 Windows 的右下角当中便可发现 **COMMGR** 的图标，在该图标上双击鼠标左键，或是点击鼠标右键后单击「开启」即可开启 **COMMGR** 的管理窗口。

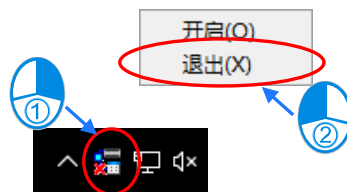


下图为 **COMMGR** 的操作画面，窗口的中间部份会列出所有已经建立好的通讯驱动，ISPSoft 便是通过指定其中的某个驱动来与 **PLC** 之间建立连结；而窗口右侧则是管理这些驱动的操作项，详细操作方式将在下一节的内容当中另行介绍。



2

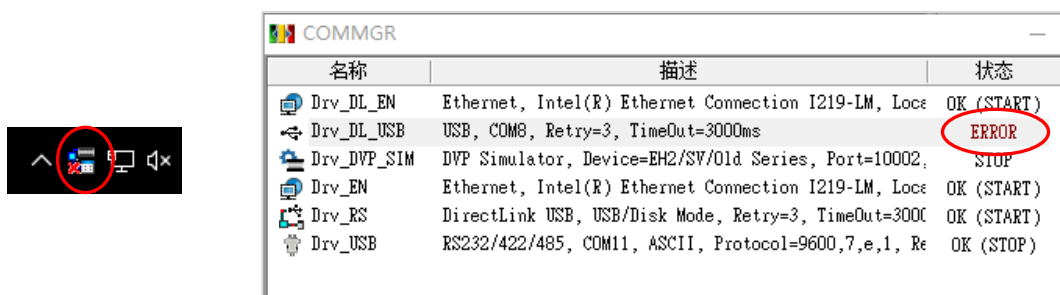
当用户按下窗口右上角的 键或是 键时都可将操作窗口关闭，不过 **COMMGR** 仍会继续常驻在 Windows 的右下角当中；若要完全关闭 **COMMGR** 时，请在 Windows 右下角的 **COMMGR** 图标上点击鼠标右键，之后选择「退出」即可将其关闭。



2.4.2 COMMGR 的驱动管理

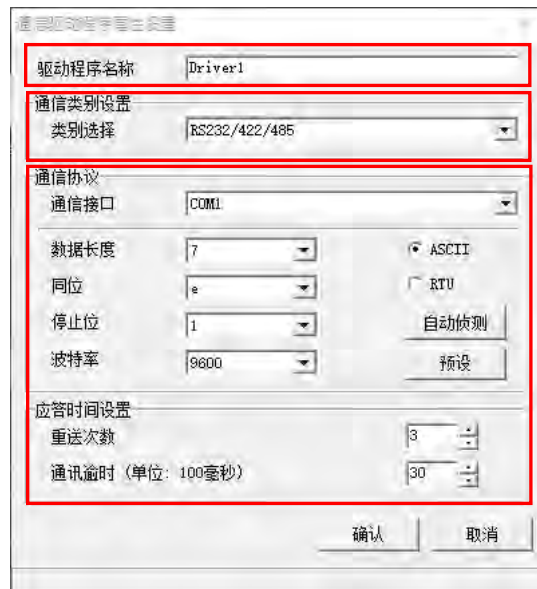
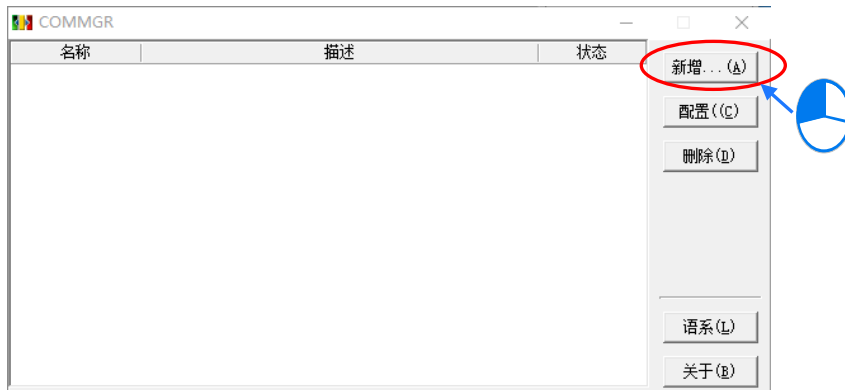


COMMGR 中的驱动所代表的意义为软件与计算机通讯端口之间的联机通道。当用户新增了一个驱动且设置好通讯参数，**COMMGR** 便会与该驱动中所指定的通讯端口建立起一个联机通道，且之后只要计算机重新启动时，**COMMGR** 便会将该驱动自动启动；而当所建立的驱动发生联机通道无法正常使用的情况，例如原本存在的网卡或是 USB 连接线被移除，**COMMGR** 便会自动将该驱动停止并标示为 **ERROR** 的状态，而此时位于 Windows 系统列上的 **COMMGR** 图标也会出现一个红色的 X 号；而一旦联机恢复正常，驱动的状态也将恢复为 **OK**。



2.4.3 建立联机通道 - 驱动的建立

在 **COMMGR** 右侧的「新增」键上点击鼠标左键即可开启通讯参数的设置窗口。



设置的步骤简述如下。

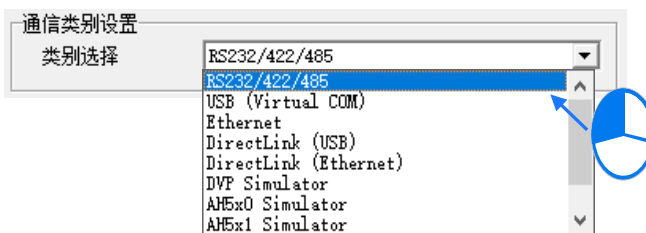
(1) 设置驱动名称

在「驱动程序名称」字段中输入此驱动机的识别名称，至多可输入 31 个字符，而输入的内容除了底线之外，不可使用其它如*、#、?、\、%、@ ...等特殊符号。



(2) 设置联机类别

在「通讯类别设定」的「类别选择」字段上点击鼠标左键，之后在下拉选单中选择欲使用的联机类型即可。下列则是目前 **COMMGR** 所支持的几种联机类型。



➤ RS232/422/485

通过计算机的 COM Port 与 PLC 主机进行通讯。

➤ USB (Virtual COM)

部分 PLC 主机本身有提供 USB 的通讯端口，因此计算机与 PLC 主机之间可直接通过 USB 连接。使用此种联机方式之前，请确认计算机中已安装该主机的 USB 驱动程序，安装方式请参考 附录 A 的相关说明或各机种的使用手册。

➤ Ethernet

通过计算机的以太网端口与 PLC 主机进行通讯。

➤ DirectLink (USB) & DirectLink (Ethernet)

此为台达 HMI 所提供的 DirectLink (穿透功能)。当 PLC 与 HMI 在正常联机的状态下，计算机便可通过 USB 或以太网络与 HMI 进行联机后，再间接地与 PLC 主机进行通讯。详细的联机设置方式以及使用上的注意事项，请参考台达 HMI 的相关使用手册。

➤ DVP Simulator & AH5x0 Simulator & AH5x1 Simulator & AS300 Simulator & AS200 Simulator

各机种仿真器的虚拟通道必须在 ISPSOft 中使用相应的机种才能进行连接。

(3) 设置通讯参数

针对所选的联机类型来设置其通讯参数，而不同的联机类型，其可设置的通讯参数也会不同。关于各种类型的参数设置方式将会在接下来的几个小节当中陆续介绍。

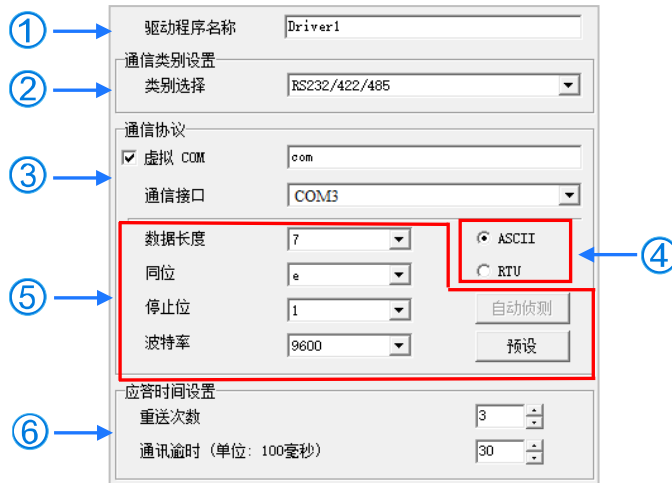
通信协议	
<input checked="" type="checkbox"/> 虚拟 COM	com
通信接口	COM3
数据长度	7
同位	e
停止位	1
波特率	9600
	<input checked="" type="radio"/> ASCII <input type="radio"/> RTU
	<input type="button" value="自动侦测"/> <input type="button" value="预设"/>
应答时间设置	
重送次数	3
通讯逾时 (单位: 100毫秒)	30

2.4.3.1 RS232/422/485 通讯参数设置

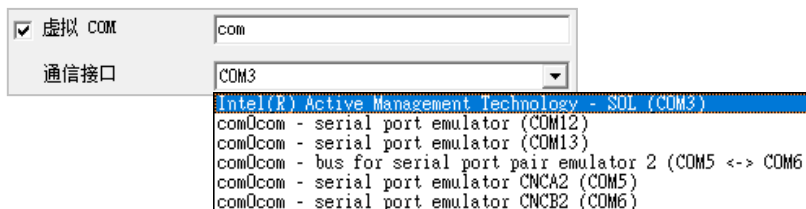
欲建立 RS232/422/485 类型的驱动时，在开启驱动设定窗口前请先确认下列两点事项：

- (a) 计算机中已成功安装相关驱动程序。
- (b) 计算机与 PLC 主机之间目前已连接在一起，且已正常运作。

2



- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」字段下拉选择「RS232/422/485」的联机类型。
- (3) 在「通信接口」字段下拉选择欲建立联机通道的通讯端口，而下拉选单中将会显示完整的 COM Port 编号与设备名称，显示的内容则会与 Windows 设备管理器中的内容一致，因此，需先确认上述 (a)(b) 两点事项之后才能选择到此 COM Port；亦可以勾选虚拟 COM 之后，输入实际连接的硬件装置名称，则可显示更多设备供选择。



- (4) 选择通讯的数据格式为 ASCII 或 RTU。
- (5) 设置该端口的通讯协议，且所设置的通讯协议并须与连接的设备一致，而按下「预设」键则可将通讯协议参数恢复为系统默认值。

若不确定联机设备的通讯协议时，可在该装置与计算机联机时按下「自动侦测」键即可自动侦测联机装置的参数，当侦测成功时，该联机装置的通讯协议参数会被回填在窗口中的对应字段；不过进行自动侦测时，系统将不会侦测 COM Port 与 ASCII / RTU 这两个参数，因此在执行自动侦测前，必须先确定这两个参数才可进行操作。

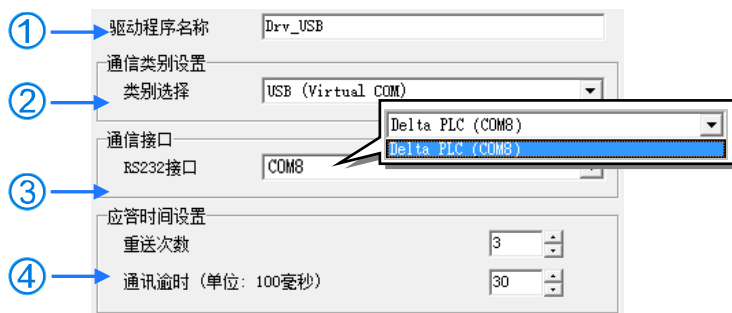
- (6) 设置通讯响应的相关参数。其中「重送次数」参数是设置联机错误时的重试次数，而「通讯超时」参数则是设置重试的时间间隔。

*.若有变更计算机连接的硬件装置，则必须关掉通讯驱动程序属性设定页面，再重新开启才能取得新的硬件通讯装置。

2.4.3.2 USB (Virtual COM) 通讯参数设置

欲建立 USB 类型的驱动时，在开启驱动设置窗口前请先确认下列两点事项：

- (a) 计算机中已成功安装该主机的 USB 驱动程序。
- (b) 计算机与 PLC 主机之间目前已通过 USB 连接线连接在一起，且已正常运作。



- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」字段下拉选择「USB (Virtual COM)」联机类型。
- (3) 在「通信接口」字段下拉选择欲建立联机通道的通讯端口，而若上述 (a)(b) 两点事项已确认无误，则此时在下拉选中便会显示已联机的主机设备名称与其 COM Port 编号。
- (4) 设定通讯响应的相关参数。其中「重送次数」参数是设定联机错误时的重试次数，而「通讯超时」参数则是设定重试的时间间隔。

*.关于 PLC 主机的 USB 驱动程序，其相关说明请参考手册中的 附录 A 或各主机的操作手册。

2.4.3.3 DirectLink (USB) 通讯参数设置



- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」字段下拉选择「DirectLink (USB)」的联机类别。
- (3) 选择 HMI 的 USB 传输模式。若选择「CDC Mode」时，在「COM Port」字段下拉选择欲建立联机通道的通讯端口。详细的联机设定方式及使用上的注意事项，请参考台达 HMI 的相关使用手册。

- (4) 设置通讯响应的相关参数。其中「重送次数」参数是设置联机错误时的重试次数，而「通讯超时」参数则是设置重试的时间间隔。

2.4.3.4 Ethernet 通讯参数设置

2

① 驱动程序名称: Drv_EN

② 通信类别设置: 类别选择: Ethernet

③ 以太网卡: 描述: Intel(R) Ethernet Connection I219-LM
192.168.2.95

④ IP 地址设置

IP地址	接口号	名称	机种
192.168.2.56	502	Main_Controller	AHCPU530-EN
192.168.2.40	502	CMC-EIP01	CMC-EIP01
192.168.2.211	502	DVP32ES2-E	ES2-E
192.168.2.210	502	DVP12SE	SE

⑤ 应答时间设置: 重送次数: 3, 通讯超时 (单位: 100毫秒): 30

- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」字段下拉选择「Ethernet」的联机类型。
- (3) 在「以太网卡」的字段下拉选择欲建立联机通道的网卡，字段左下角会显示目前的 IP 地址。
- (4) 于「IP 地址设置」区域中，建立通讯装置的 IP 地址与名称。
 - 按下「搜寻」后便会开始搜寻连接在相同网域的装置，并将结果显示于列表中。
 - 按下「增加」按钮可于下方的列表中新增一组 IP 地址，之后便可于各字段中输入相关信息。
 - ❶ IP 地址：装置的 IP 地址。
 - ❷ 接口号：通讯时使用的端口号，预设为 502。
 - ❸ 名称：装置的名称。
 - ❹ 机种：装置的机种，只能由搜寻功能显示。
 - 选取某组 IP 后按下「删除」按钮或直接按下键盘的【DEL】键即可将该组 IP 在列表中删除。
- (5) 设定通讯响应的相关参数。其中「重送次数」参数是设定联机错误时的重试次数，而「通讯超时」参数则是设定重试的时间间隔。

*.若有变更计算机连接的网络装置，则必须关掉通讯驱动程序属性设定页面，再重新开启才能取得新的网络装置。

2.4.3.5 DirectLink (Ethernet) 通讯参数设置

① 驱动程序名称: Drv_DL_EN

② 通信类别设置: 类别选择: DirectLink (Ethernet)

③ 以太网卡: 描述: Intel(R) Ethernet Connection I219-LM
192.168.2.95

④ IP 地址设置:

IP地址	接口号	名称	机种
192.168.2.220	502	HMI_1	

⑤ 应答时间设置: 重送次数: 3
通讯逾时 (单位: 100毫秒): 30

- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」字段下拉选择「DirectLink (Ethernet)」联机类型。
- (3) 在「以太网卡」的字段下拉选择欲建立联机通道的网卡，而此字段左下角的数字即是所选网卡目前被配置的 IP 地址。
- (4) 于「IP 地址设置」的区域中，建立通讯装置的 IP 地址与名称。
 - 按下「搜寻」后便会开始搜寻连接在相同网域的装置，并将结果显示于列表中。
 - 按下「增加」按钮可于下方的列表中新增一组 IP 地址，之后便可于各字段中输入相关信息。
 - ① IP 地址：装置的 IP 地址。
 - ② 接口号：通讯时使用的端口号，默认为 502。
 - ③ 名称：装置的名称。
 - ④ 机种：装置的机种，只能由搜寻功能显示。
 - 选取某组 IP 后按下「删除」按钮或直接按下键盘的【DEL】键即可将该组 IP 于列表中删除。
- (5) 设定通讯响应的相关参数。其中「重送次数」参数是设定联机错误时的重试次数，而「通讯逾时」参数则是设定重试的时间间隔。

*.若有变更计算机连接的网络装置，则必须关掉通讯驱动程序属性设定页面，再重新开启才能取得新的网络装置。

2.4.3.6 DVP Simulator 通讯参数设置

- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设定」的「类别选择」栏位下拉选择「**DVP Simulator**」类型。
- (3) 在此输入指定的通讯接口号。
- (4) 设定 DVP 仿真器的种类，可分为「**EH2/SV/Old Series**」、「**EH3/ EH3-L/SV2**」、「**ES2/ EX2/SA2/SX2/MC**」、「**SE**」及「**SS2**」。
- (5) 设定通讯回应的相关参数。其中「**重送次数**」参数是设定连线错误时的重试次数，而「**通讯超时**」参数则是设定重试的时间间隔。

*. **ES**、**SS**、**EX**、**VFD**、**EC** 机种项目，下载到 DVP 仿真器，上载回来时会变为 **EH** 机种项目。

*. **TP** 机种项目请选择 **SS2** 机种仿真器。

2.4.3.7 AH5x0 与 AH5x1 Simulator 通讯参数设置

- (1) 在「驱动程序名称」字段输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」字段下拉选择「**AH5x0 Simulator**」或「**AH5x1 Simulator**」的联机机种。
- (3) 在此输入指定的通讯接口号。

- (4) 设定通讯响应的相关参数。其中「重送次数」参数是设定联机错误时的重试次数，而「通讯超时」参数则是设定重试的时间间隔。

2.4.3.8 AS Simulator 通讯参数设置



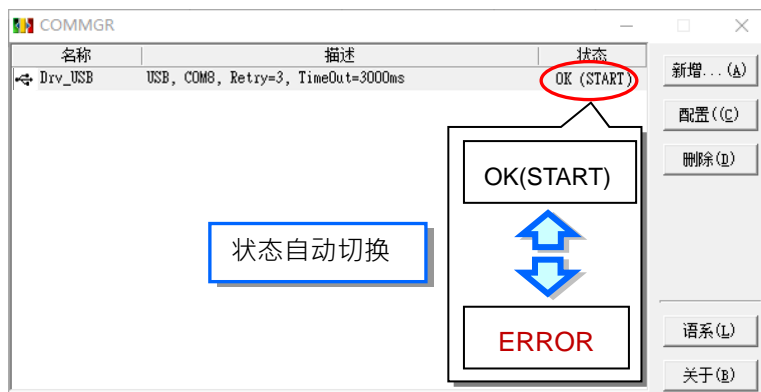
- (1) 在「驱动程序名称」栏位输入识别名称，而输入的内容除了底线之外，不可使用其它特殊符号。
- (2) 在「通讯类别设置」的「类别选择」栏位下拉选择「AS200 Simulator」或「AS300 Simulator」的机种。
- (3) 于此输入指定的通讯接口号。
- (4) 设定通讯响应的相关参数。其中「重送次数」参数是设定联机错误时的重试次数，而「通讯超时」参数则是设定重试的时间间隔。

补充说明

- (a) 仿真器的功能建议搭配 ISPSOft V3.05 (含) 以上版本使用。
- (b) 仿真器不支持完整的功能跟指令，使用上需要注意。

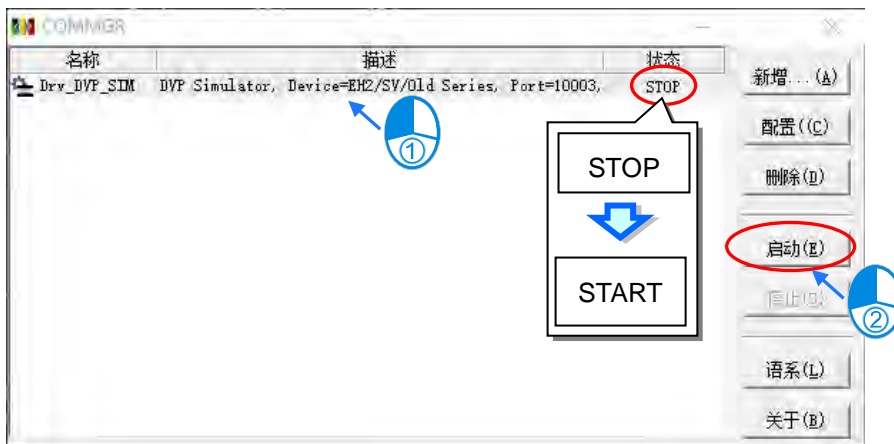
2.4.4 建立联机通道 - 驱动的启动与停止

对于 USB 连接的驱动，只要装置与计算机的联机正常，COMMGR 即会自动侦测联机，使用者无须任何操作。

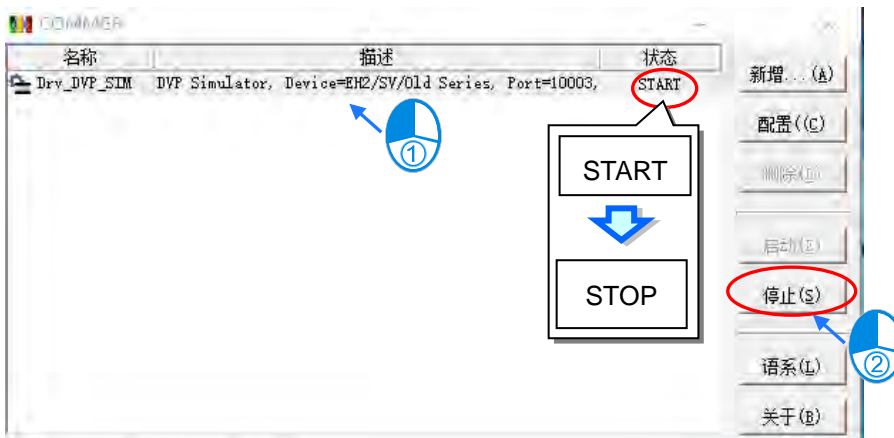


仿真器驱动必须先启动才能使用，需要先在 COMMGR 的管理列表上点选该仿真器驱动，之后再于右侧的「启动 (E)」点击鼠标左键，仿真器驱动启动成功会显示 **START**。

2

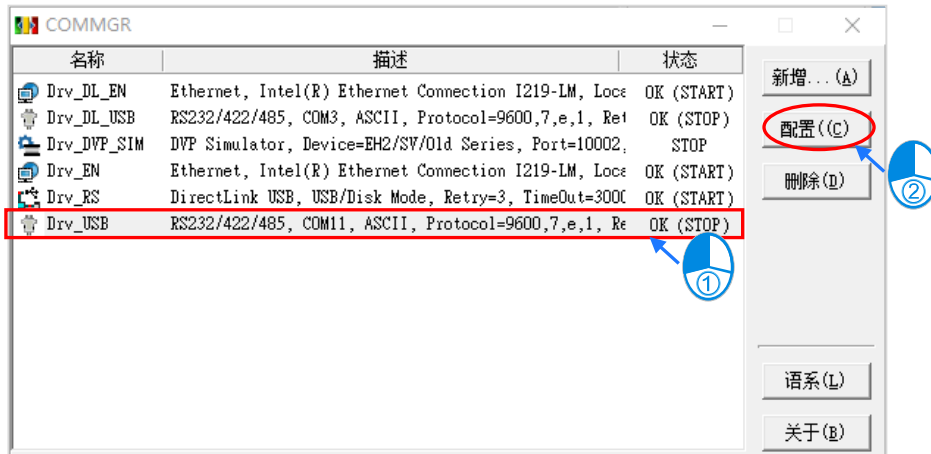


而欲停止仿真器驱动时，只要先在 COMMGR 的管理列表上单击该驱动，之后再在右侧的「Stop」键上点击鼠标左键，待该驱动的状态显示为 **STOP** 后即代表已经停止。

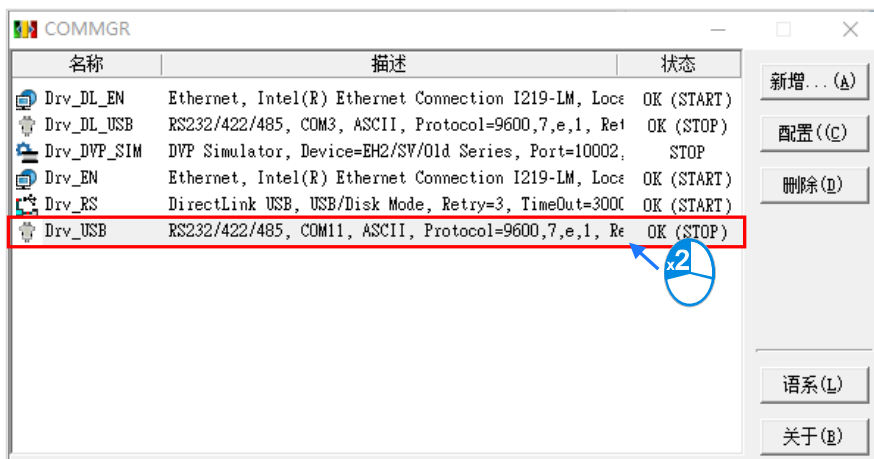


2.4.5 建立联机通道 - 驱动力的修改与删除

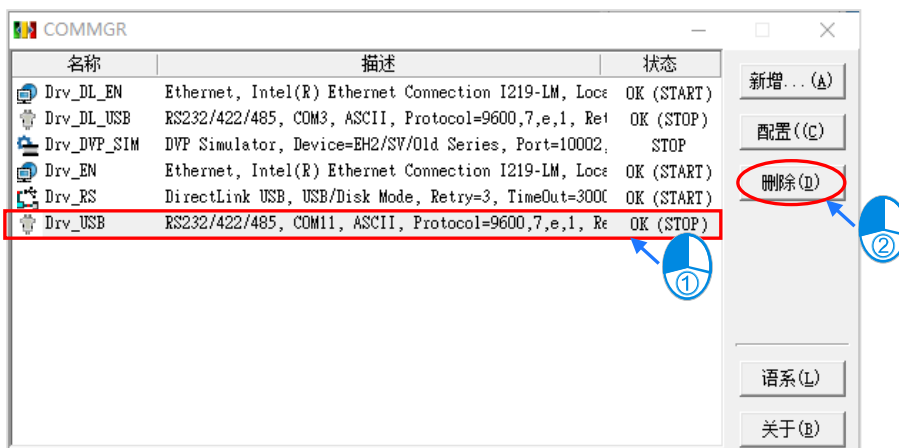
欲修改某个驱动力的参数时，请先在列表上单击该驱动后点击右侧的「配置」按钮，或是直接在该驱动上双击鼠标左键，之后便会开启设置窗口，即可修改此笔数据。



OR



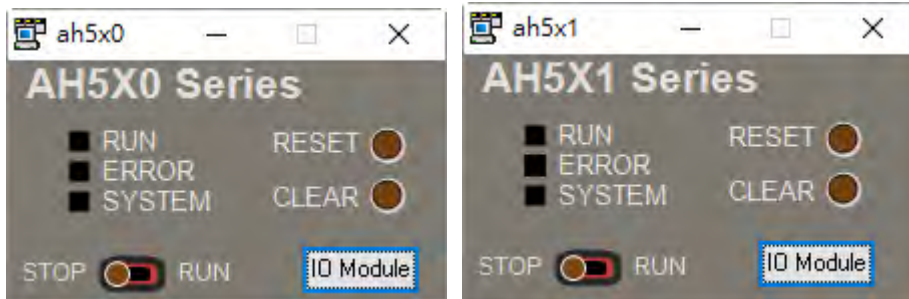
欲删除某个驱动时，请先在列表上单击该驱动后点击右侧「删除」按钮或直接按下键盘的【DEL】键即可将其删除。



2.4.6 建立联机通道 - 仿真器的面板操作

当仿真器的状态显示为 **START** 时，AH5x0 与 AH5x1 仿真器会弹出一个操作面板，用来查看目前仿真器的状态灯号，并且可以通过 **STOP** 和 **RUN** 来控制仿真器的停止与运行，如下图。

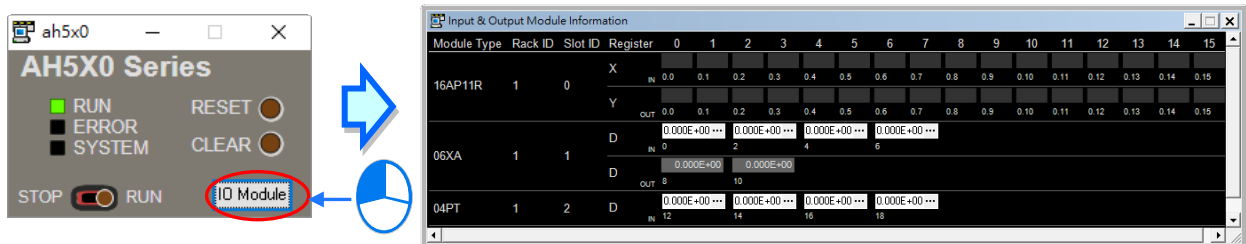
2



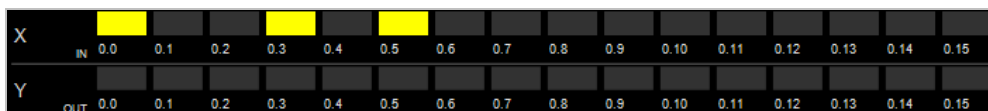
仿真器的「IO Module」按钮则可开启模块 IO 面板，使用此功能之前，必须先在 HWCONFIG 中配置模块并参考第 3.1.2.5 节将参数下载至主机，如下图。



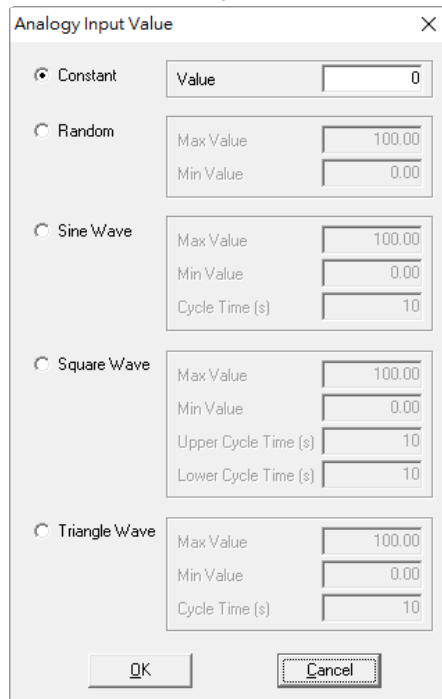
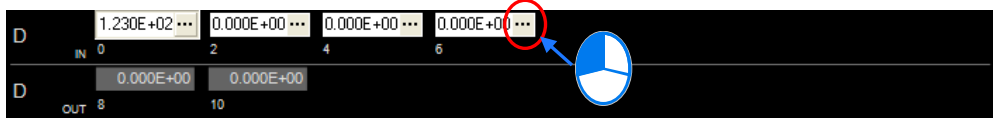
再将仿真器切换至 **RUN** 运行，接着就可以在模块 IO 面板看到跟 HWCONFIG 一样配置的模块与其状态，如下图。



下图是一个数字 IO 的缓存器操作区域，其中 Y 装置的部分仅可用来检视输出状况，但无法进行任何操作；而 X 装置的部份则可直接用鼠标点击各个接点而变更其 ON/OFF 状态，以模拟外部的输入。



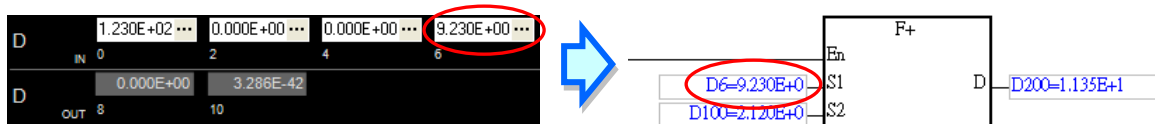
下图是一个模拟 IO 的缓存器操作区域，其中的每个字段皆是 32 位浮点数的格式，并对实际模块的各个通道；下方的 D8 与 D10 是对应模拟输出通道的 D 装置，输出仅可用来检视；上方的 D0、D2、D4 与 D6 则是对应模拟输入通道的 D 装置，可于此处点击 按钮，开启设定窗口输入数值，如下图。



2

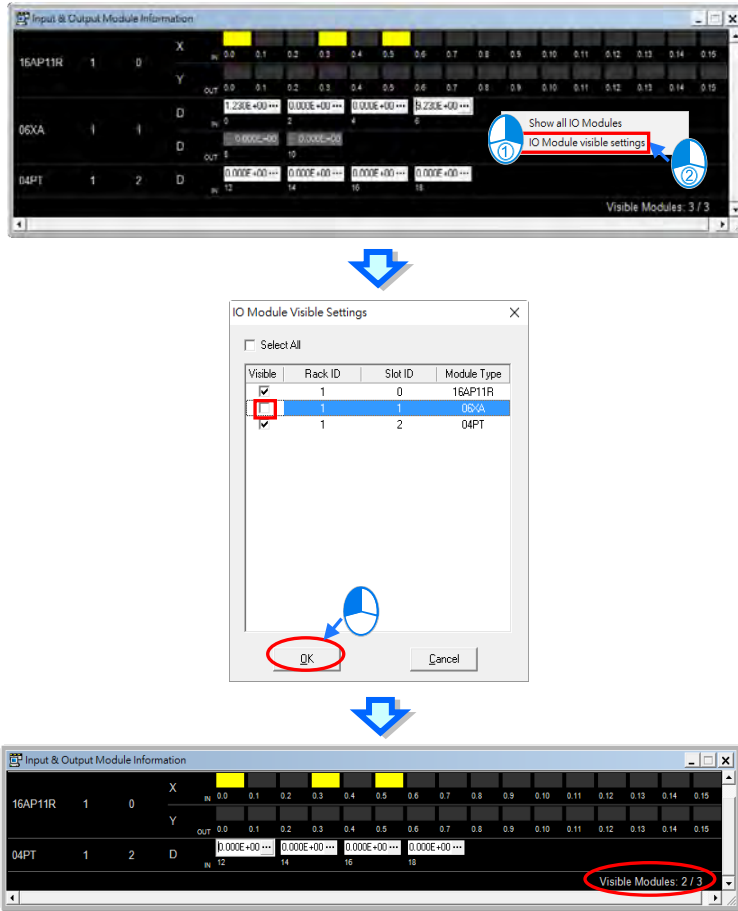
信号种类	说明
Constant	为一个固定数值的常数信号。
Random	可指定最大值与最小值的随机信号。
Sine Wave	可指定最大值、最小值与信号周期的正弦波信号。
Square Wave	可指定最大值、最小值、ON 时间与 OFF 时间的周期性矩形波。
Triangle Wave	可指定最大值、最小值与信号周期的三角波信号。

设定完毕后，该字段（通道）便会依据所设定的信号模式持续送出数据给 COMMGR，而 ISPSOft 便可通过 COMMGR 来撷取到仿真器的信号，并可藉此进行相关的测试。

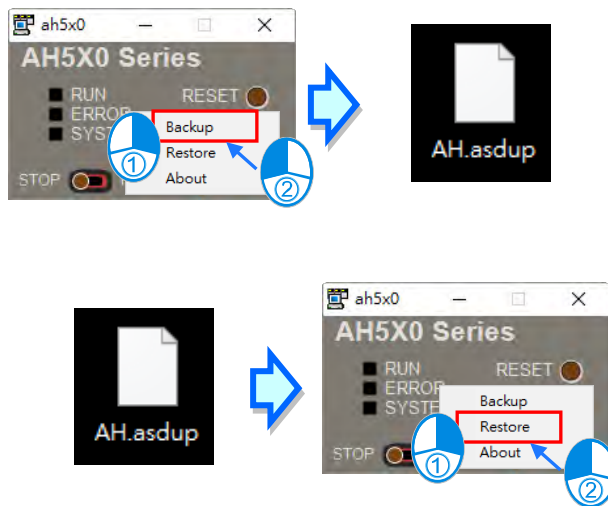


于模块 IO 面板按下鼠标右键，选取「**Show All IO Module**」可以显示所有模块，若要隐藏模块就点选「**IO Module Visible settings**」，接着于设定窗口中取消勾选欲隐藏的模块项目，完成后按下「**OK**」即可由窗口的右下角，得知目前的显示状况，如下图。

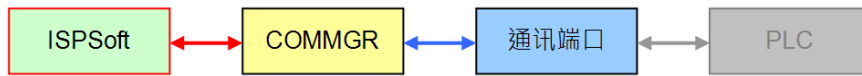
2



在 AH5x0 与 AH5x1 仿真器，还可以通过仿真器本身所提供的备份功能，将参数与程序备份成一个.asdup 档案，再通过这个文件还原到其他相同的仿真器如下图。



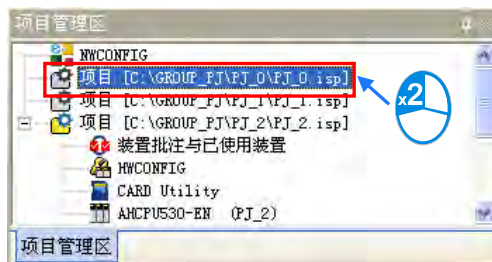
2.4.7 建立 ISPSOft 与 COMMGR 之间的连结



在 **COMMGR** 中成功建立驱动并启动后，接下来便可在 **ISPSOft** 中指定欲使用的驱动，而完成设置后，在 **ISPSOft** 与 **COMMGR** 之间便建立了连结。设置时，每个**单机项目**都必须单独指定其通讯方式，因此在**群组项目**的架构下，用户必须逐一开启每个单机项目并完成各自的设置。

2

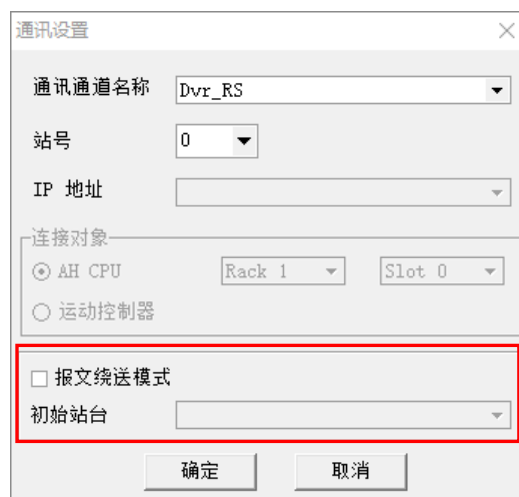
(1) 首先请在群组中将欲设置通讯方式的项目启动，但若是在**单机项目**的架构下则可略过此步骤。



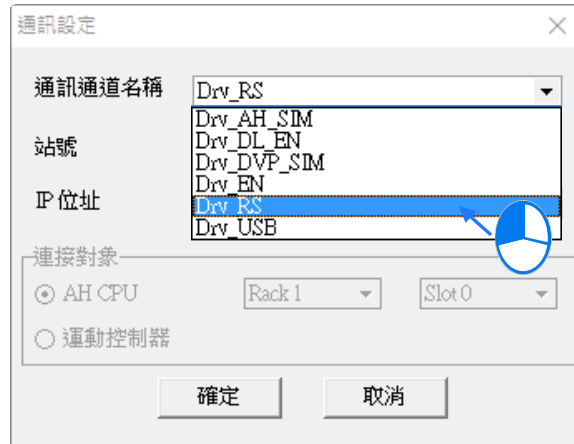
(2) 在 ISPSOft 中点击功能工具栏的 **工具 (I) > 通讯设置 (P)**，之后便会出现设置窗口。



(3) 当所设定的对象为**群组项目**下的 AH5x0 系列机种时，会增加「**封装绕送模式**」的项目，该选项需搭配 NWCONFIG 使用，相关操作与说明请参考第 20 章。



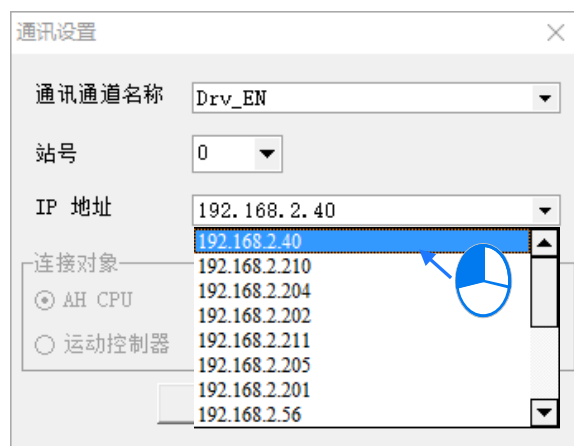
- (4) 在通信设置窗口中的「通道名称」，下拉选择欲连接的驱动。



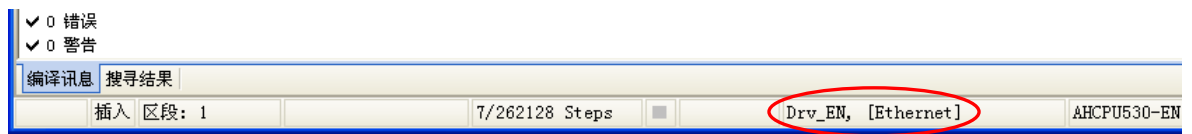
- (5) 于「站号」下拉选择与计算机实际联机的 PLC 主机站号，也可直接使用站号 0 (广播站号)。



- (6) 若驱动类型为 **Ethernet** 或 **DirectLink (Ethernet)** 时，于「IP 地址」下拉选择欲连接 PLC 主机的 IP 地址。

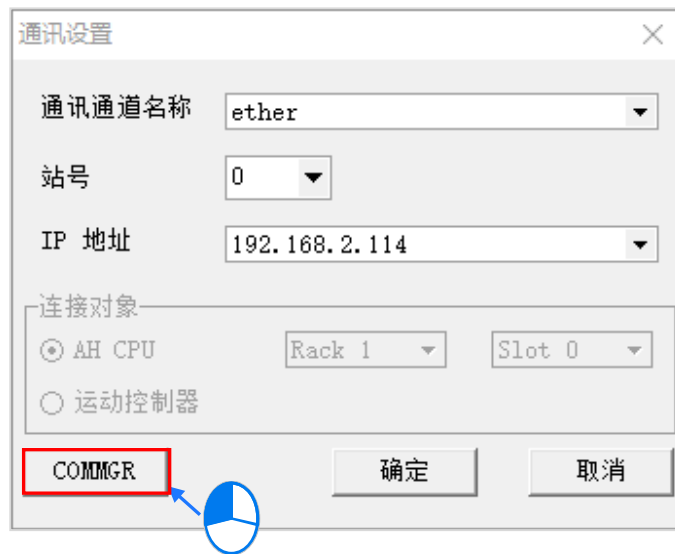


(7) 完成所有设置后按下「确定」按钮即可。在 ISPSOft 的状态栏中便会显示目前所连结的驱动信息。



补充说明

如果 COMMGR 设定有变更，则 ISPSOft 的通信设置，必须要重新选择一次，才能正常连接，也可以透过 **COMMGR** 按钮来调用 COMMGR 设定窗口，COMMGR 版本须提升至 V1.09.05 以上，才能搭配此便利功能。



2.4.8 连接主机与通讯端口

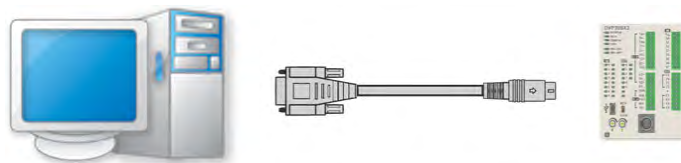


接下来便可通过通讯连接线将主机与指定的通讯端口接连接在一起。下述将列举一些常用的连接方式与提醒事项，而更进一步的接线与设置方式则请自行参考各机种的操作手册。

1. DVP 系列主机 (RS232)

直接以台达专用的 **PLC 通讯连接线** 与主机连接，而 COMMGR 驱动的种类则请选择 RS232。

注意事项：联机前请确认 COMMGR 驱动中所设置的 RS232 参数与 PLC 主机的设置一致。



2. DVP - SX2 主机 (USB)

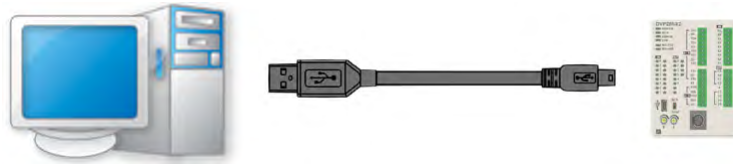
DVP 系列的 SX2 主机本身有提供 Mini USB (B-Type) 的通讯接口，用户可直接以 USB 连接线将主机与计算机进行接续，但因 SX2 机种的 USB 架构是将 USB 转 RS232 的电路内建在主机中，因

此其内部运作仍是采用 RS232 的模式，所以 COMMGR 驱动的种类请选择 RS232。

注意事项：

- (a) 请确认计算机中已正确安装 SX2 机种的 USB 驱动程序，相关说明请参考 附录 A。
- (b) 联机前请确认 COMMGR 驱动中所设置的 RS232 参数与 PLC 主机的设置一致。

2



3. AH/AS 系列主机 (USB)

直接通过 USB 连接线与主机进行接续，而其中主机的 USB 端口为 B-Type 的 Mini USB 型式；COMMGR 驱动的联机类型则请选择 USB (Virtual COM)。

注意事项：请确认计算机中已正确安装 AH500 机种的 USB 驱动程序，相关说明请参考 附录 A。

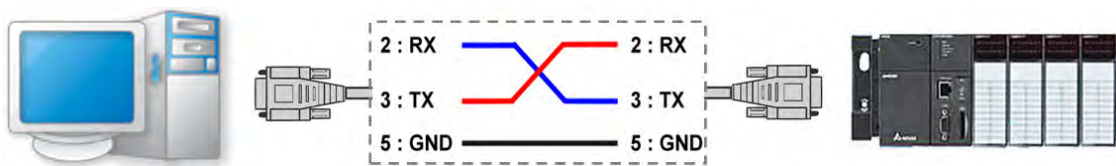


4. AH 系列主机 (RS232/422/485)

AH 系列主机的 COM 端口为标准型式，因此请使用 RX 与 TX 跳线的 RS232 连接线，或通过具跳线功能的转换器来与计算机进行连接，而 COMMGR 驱动的联机类型则请选择 RS232/422/485。

注意事项：

- (a) 联机前请确认 COMMGR 驱动中所设置的 RS232 参数与 PLC 主机的设置一致。
- (b) AH 系列机种的 COM 接口可规划为 RS232 / RS485 / RS422 等多种模式，联机前请确认该接口目前所规划的模式是否正确。(请参考后续的补充说明)

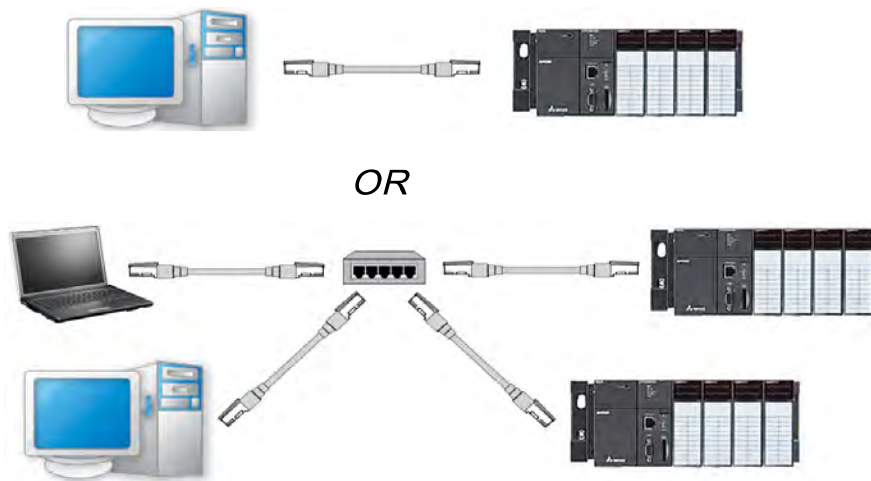


5. AH/AS 系列主机、DVPxxMC、AS5xx 与 DVP-SE (Ethernet)

AHCPU5xx-EN 系列、AHxxEMC 系列、AS 系列、DVPxxMC、AS5xx 及 DVP-SE 主机有内建 Ethernet 的 RJ - 45 端口，并可通过以太网络与计算机进行连接，可以通过 HUB 将计算机与 PLC 主机连接至同一网域上，也可以用网络线直接将计算机与 PLC 主机直接对接。至于 COMMGR 驱动类型则请选择 Ethernet。

注意事项：

- (a) 联机前请确认所架构的网络处于正常工作的状态。
- (b) 确认 CPU 主机本身的以太网参数是否正确，请参考后续的补充说明。



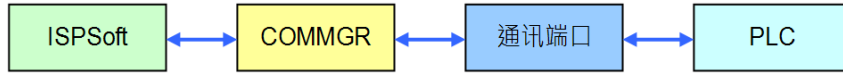
2

补充说明

各机种通讯参数的出厂默认值,请参阅各机种操作手册。

2.4.9 实际联机测试

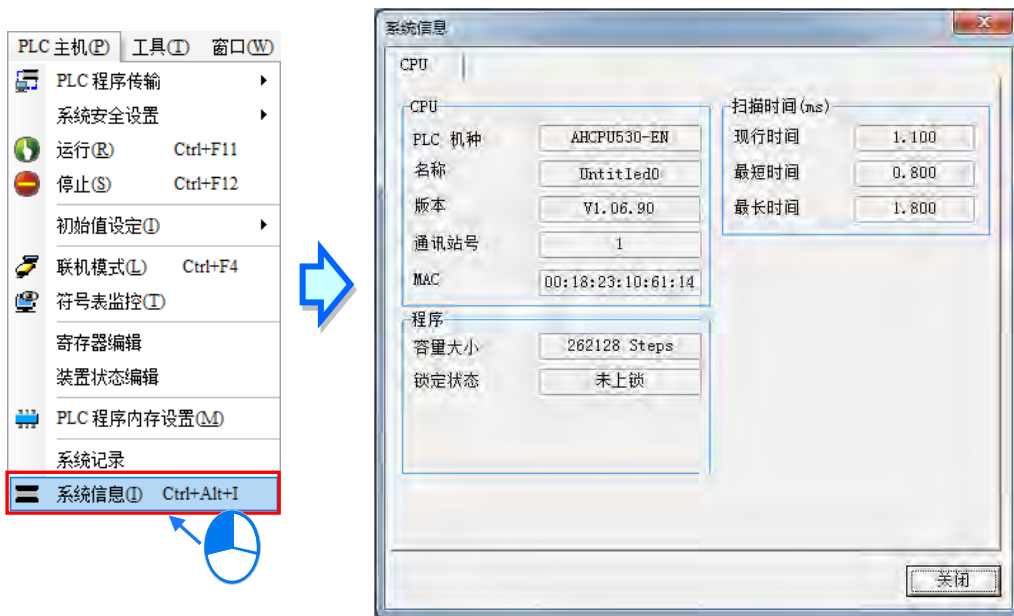
若已完成上述的所有步骤，接着便可通过简单的测试来验证是否可与主机进行正常联机。



2 联机之前请再次确认下列事项。

- (a) PLC 主机与计算机已正确通过通讯连接线连接，主机也已送电并处于正常运作的状态。
- (b) 提供通讯的装置，如计算机的网卡、网络 HUB、串行端口...等，已处于正常工作的状态。
- (c) 针对欲使用的 COMMGR 驱动，请确认相关参数的设定正确无误动，且驱动处于 **OK** 状态。
- (d) 确认在 ISPSOft 的「**通讯设置**」中，其所指定的驱动、站号与 IP 地址等设定皆正确无误。

确认完上述事项后，接着请于 ISPSOft 的功能工具栏中，点选 **PLC 主机 (P) > 系统信息 (!)**，此时便会弹出系统信息，若无弹出系统信息表示连接失败，请检查(a)~(d)项目后再试。



3

第3章 PLC 系统的配置与设置

目录

3.1 硬件规划工具 - HWCONFIG 4.0	3-5
3.1.1 进入 HWCONFIG 的第一步	3-6
3.1.2 HWCONFIG 的环境介绍	3-10
3.1.2.1 视窗标题与状态栏	3-10
3.1.2.2 菜单工具列-文件	3-11
3.1.2.3 菜单工具列-编辑	3-18
3.1.2.4 菜单工具列-检视	3-29
3.1.2.5 菜单工具列-通讯	3-29
3.1.2.6 菜单工具列-工具	3-39
3.1.2.7 菜单工具列-帮助	3-48
3.1.2.8 菜单工具列-常用	3-49
3.1.2.9 项目树	3-50
3.1.2.10 产品列表	3-52
3.1.2.11 输出视窗	3-53
3.1.2.12 硬件配置区	3-54
3.1.2.13 编辑区	3-55
3.1.3 进阶功能操作	3-57
3.1.3.1 新增模块/扩展背板	3-57
3.1.3.2 更换模块/背板	3-65
3.1.3.3 模块/扩展背板的移动	3-67
3.1.3.4 备注编辑	3-69
3.1.3.5 硬件配置区-重设配置	3-71
3.1.3.6 硬件配置区-显示模块详细信息	3-72
3.1.3.7 硬件配置区-输出/输入装置重新排序	3-73
3.1.3.8 硬件配置区-模块的地址配置	3-73
3.1.3.9 硬件配置区-更改远程装置地址	3-74
3.1.3.10 硬件配置区-重置大小	3-75
3.1.3.11 编辑区-模块内部参数的导出与导入	3-76

3.1.3.12	编辑区-模块内部参数的更新.....	3-77
3.1.3.13	显示/隐藏功能	3-77
3.1.3.14	信息检视与模块诊断	3-78
3.1.3.15	在线变更模块状态	3-79
3.1.3.16	远程模块配置	3-81
3.1.3.17	打开外部软件	3-83
3.1.3.18	IO-Link.....	3-84
3.1.3.19	变量设定	3-102
3.1.3.20	冗余功能	3-107
3.1.4	PLC 主机参数设定.....	3-108
3.1.4.1	系统设定-系统资讯.....	3-108
3.1.4.2	系统设定-系统参数.....	3-109
3.1.4.3	系统设定-装置元件范围设定.....	3-113
3.1.4.4	系统设定-输入点滤波时间	3-114
3.1.4.5	系统设定-定位轴参数	3-115
3.1.4.6	系统设定-日光节约时间	3-116
3.1.4.7	COM 通讯端口设定	3-117
3.1.4.8	以太网网络基本设定	3-118
3.1.4.9	以太网进阶设定-IP 过滤器.....	3-119
3.1.4.10	以太网进阶设定-网络时间校正.....	3-120
3.1.4.11	以太网进阶设定-邮件	3-121
3.1.4.12	以太网进阶设定-通讯接口	3-126
3.1.4.13	以太网进阶设定-RTU-EN01 对应.....	3-128
3.1.4.14	功能卡 1 设定	3-129
3.1.4.15	功能卡 2 设定	3-130
3.1.4.16	台达设备参数还原功能	3-134
3.1.4.17	数据交换	3-138
3.1.5	EtherNet/IP	3-141
3.1.5.1	开启 EIP	3-141
3.1.5.2	上载 EtherNet/IP	3-143
3.1.5.3	下载 EtherNet/IP	3-144

3.1.5.4	扫描从站	3-145
3.1.5.5	加入从站	3-146
3.1.5.6	设定主站\从站 IP.....	3-148
3.1.5.7	建立联机	3-149
3.1.5.8	删除联机	3-152
3.1.5.9	设定 RTU 模块配置.....	3-153
3.1.5.10	清空从站	3-154
3.1.5.11	显示 RTU 信息	3-155
3.1.5.12	显示数据交换.....	3-155
3.1.6	EtherCAT	3-156
3.1.6.1	开启 EtherCAT	3-157
3.1.6.2	上载 EtherCAT	3-158
3.1.6.3	下载 EtherCAT	3-159
3.1.6.4	扫描 EtherCAT 从站.....	3-160
3.1.6.5	EtherCAT 网络拓扑图操作	3-160
3.1.6.6	设定从站	3-167
3.1.6.7	设定主站	3-177
3.2	硬件规划工具 - HWCONFIG 3.0	3-179
3.2.1	HWCONFIG 的环境介绍	3-179
3.2.2	主机的版本管理	3-180
3.3	DVP 系列机种的参数设定	3-180
3.3.1	DVP 主机的系统管理工具	3-180
3.3.2	停电保持设定	3-181
3.3.3	联机信息	3-185
3.3.4	DVP-ES3 机种的主机参数设置	3-186
3.4	DVPxxMC 机种的 PLC 主机参数设定	3-188
3.4.1	开启 PLC 主机的参数设定页面	3-188
3.4.2	一般设定	3-189
3.4.2.1	一般设定-系统信息	3-189
3.4.2.2	一般设定-CANopen	3-190
3.4.2.3	一般设定-CANMotion.....	3-191

3.4.2.4	一般设定-RS232.....	3-191
3.4.2.5	一般设定- RS485.....	3-192
3.4.2.6	一般设定- 增量型编码器 1	3-192
3.4.2.7	一般设定- 增量型编码器 2	3-193
3.4.2.8	一般设定- 绝对型编码器.....	3-194
3.4.2.9	一般设定- 输入点过滤时间	3-195
3.4.2.10	一般设定- 运转 / 停止设定	3-196
3.4.2.11	一般设定- 以太网基本设定	3-197
3.4.3	上下载	3-198
3.5	万年历设定	3-199
3.6	PLC 内存设定 (清除程序&恢复出厂值)	3-200

在 ISPSOft 中，DVP 系列主机可通过内建的功能或工具来对 PLC 主机或整个系统进行组态配置与参数设置；而其它系列的主机，则使用硬件规划工具 HWCONFIG 来对 PLC 主机与模块作进行设定。基于 PLC 主机产品种类众多，旧 HWCONFIG 的接口都没有统一，新一代的 HWCONFIG 将提供全新的整合操作接口，让用户更快速方便的完成操作设定，新一代的 HWCONFIG 将会持续增加支持更多的机种，目前由 ISPSOft 开启的 HWCONFIG 将依照下列表格自动开出相对应的版本，接着我们将逐一介绍关于系统组态与参数设置等功能的操作步骤与设置方式。

机种 软件	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
HWCONFIG 4.0	V	V	V	V	V		只支持 AS5xx
HWCONFIG 3.0						只支持 ES3	只支持 DVPxxMC

3

3.1 硬件规划工具 - HWCONFIG 4.0

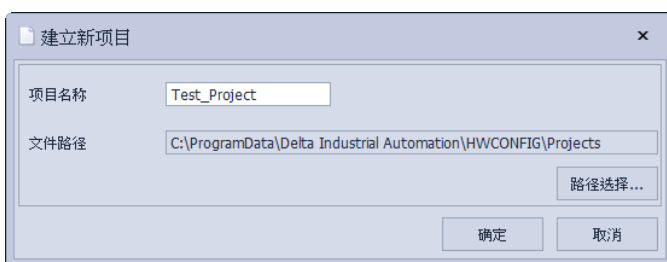
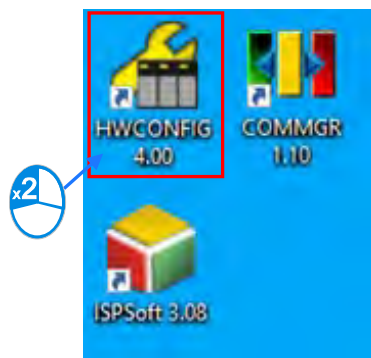
HWCONFIG 是 ISPSOft 的硬件规划工具之一，其功能包括 CPU 主机参数的设定、模块背板的配置、模块参数的设定、硬件参数上下载及在线检测与诊断功能。**HWCONFIG4.0** 提供用户自行选择显示的主题风格，默认为浅色系，本章节使用浅色系做介绍，详细设定方式，可以参考第 3.1.2.6 节环境设定。

⚠ 请注意！所有硬件规划的参数于设定完毕后，必须被下载至主机中才可生效。

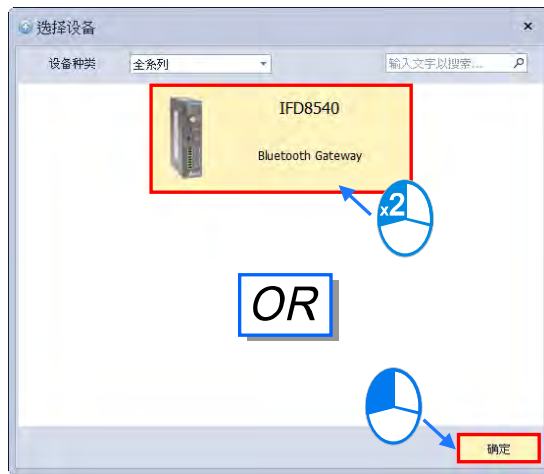
3.1.1 进入 HWCONFIG 的第一步

新的 HWCONFIG 支持独立开启，在安装完成 ISPSoft 之后，便会自动在桌面中建立快捷方式，单击该快捷方式即可启动单机版 HWCONFIG 如下图。开启单机版 HWCONFIG，必须先建立一个项目，系统会自动跳出建立新项目的页面，须于设定窗口中依序输入「项目名称」及「文件路径」，完成后单击「确定」。

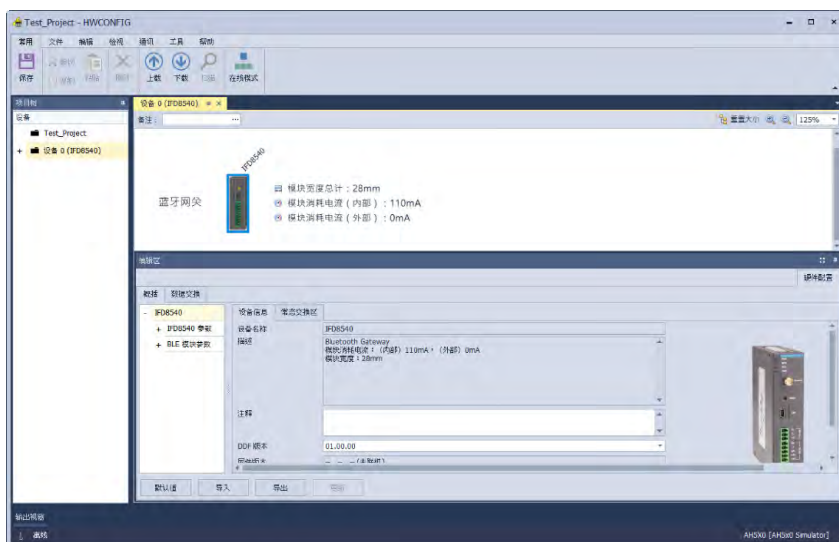
3



接着在选择装置窗口，单击「装置类型」的下拉选项并选择机种类型，再于要建立的 PLC 机种上双击鼠标右键，也可以在选取 PLC 机种之后单击「确定」，即可进入操作画面，如下图。

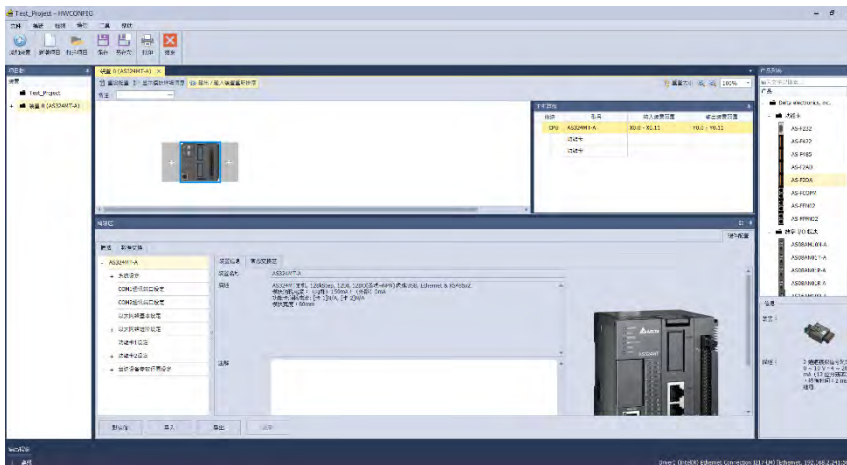
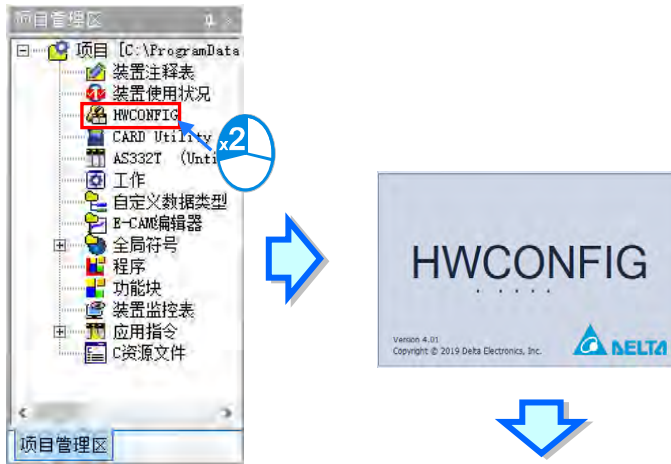


3

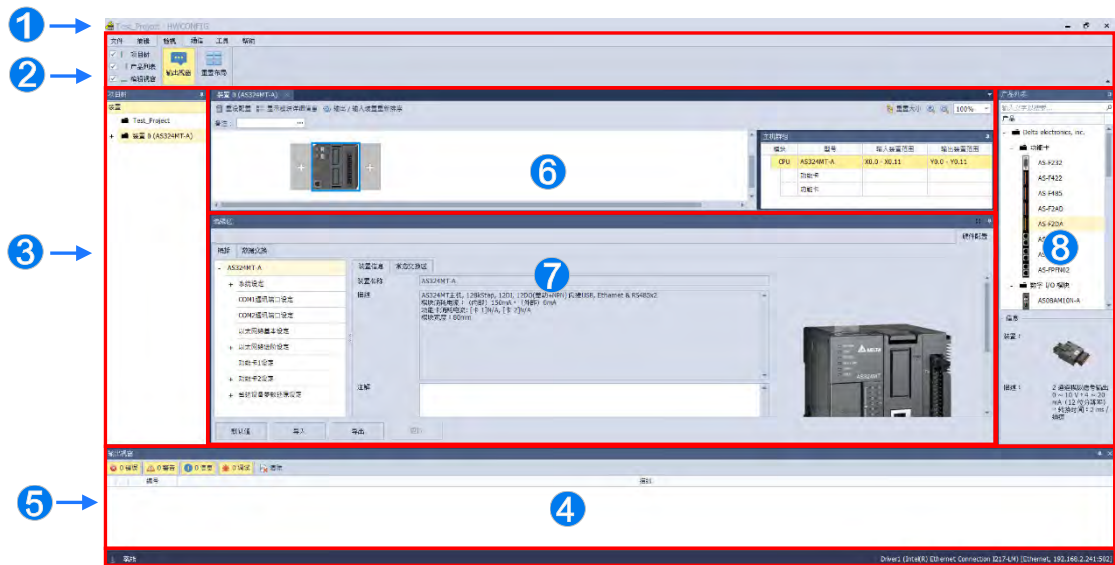


而在 ISPSoft 软件的项目中，双击鼠标左键开启 HWCONFIG，则项目机种与通讯设置，将会自动带入 HWCONFIG 中。

3



下图是一个完整的 HWCONFIG 工作画面。



- ❶ 窗口标题：显示目前所编辑的项目名称与程序标题。
- ❷ 菜单工具栏：共有 6 类的功能操作选项。
- ❸ 项目树：阶层树状结构的项目对象管理接口。
- ❹ 输出视窗：显示项目的相关信息。
- ❺ 状态栏：显示目前联机状态的相关信息。
- ❻ 硬件配置区：用户可于此进行硬件的配置。
- ❼ 编辑区：用户可于此进行主机与各模块的参数设定。
- ❽ 产品列表：阶层树状结构的方式呈现所有可用的硬件资源。

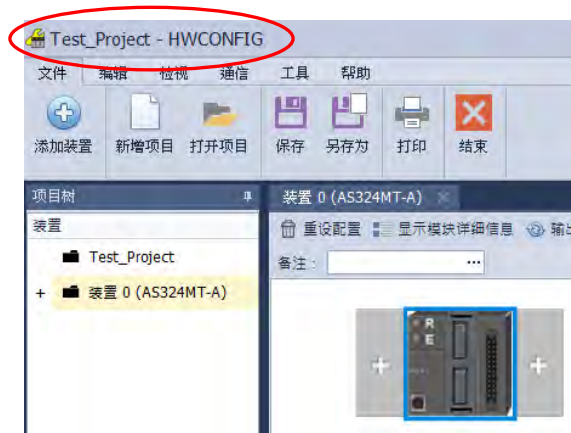
3.1.2 HWCONFIG 的环境介绍

HWCONFIG 的环境有分窗口标题与状态栏及菜单工具栏的 6 类菜单项等，依据所选用的机种，各类菜单项的下层项目也会有所差异，在此先稍做浏览介绍，详细的操作方式则将于下方的章节中陆续介绍。

3.1.2.1 视窗标题与状态栏

成功建立项目之后，HWCONFIG 的窗口标题便会显示该项目的名称，其格式为 项目名称[程序标题]。

3



窗口下方的状态栏主要在于显示目前的各种工作状态，包括联机状态以及通讯设置参数。



3.1.2.2 菜单工具列-文件

文件：主要菜单为基本的项目存取操作。



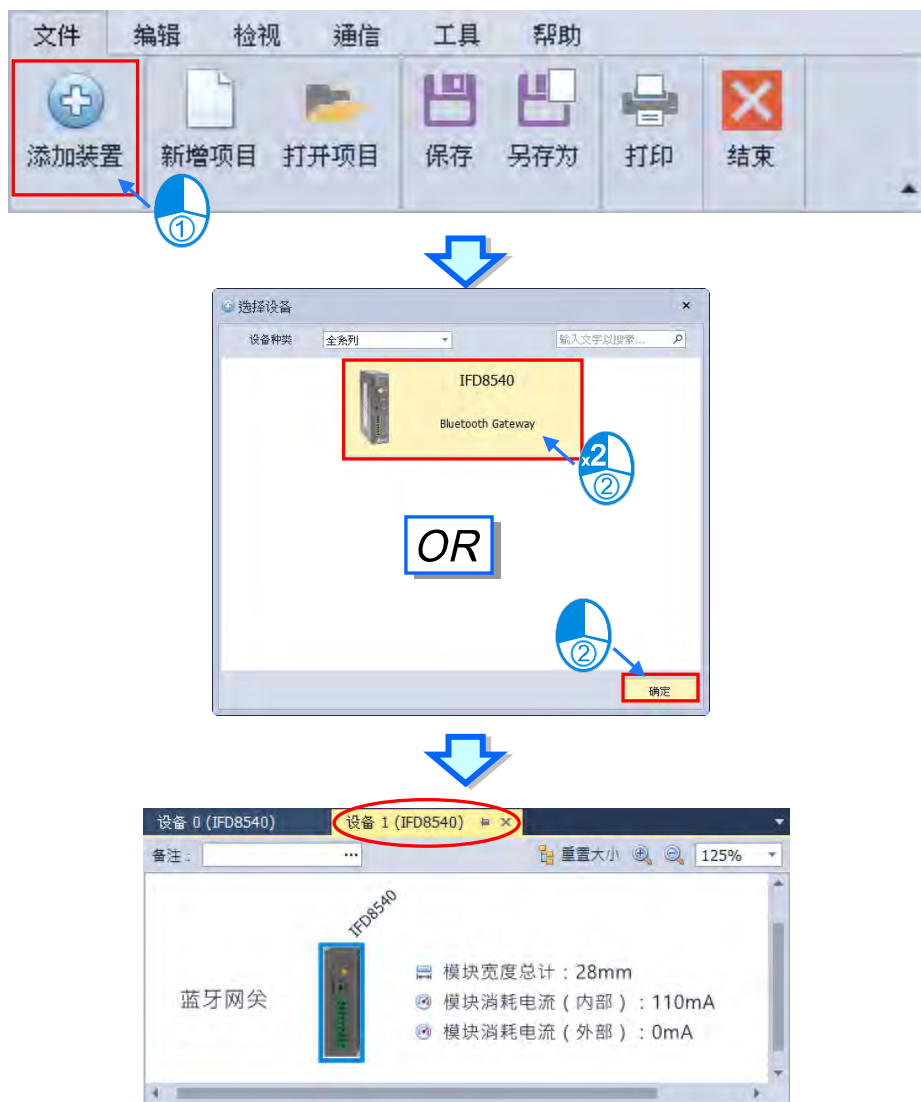
操作项	功能说明
添加装置	添加新的 PLC 主机至项目中。
新增项目	开启新项目。
打开项目	开启旧项目。
保存	保存项目。
另存为	另存项目在新的位置。
打印	打印项目硬件配置图。
结束	关闭 HWCONFIG 软件。

● 添加装置

添加装置只支持单机版。添加装置可在项目中增加另一个主机。

在「文件」内单击「添加装置」，会跳出选择装置窗口，选定好装置后可以按双击鼠标左键或单击「确定」键，即可在项目增加该装置，在硬件配置区会产生新装置的分页。

3

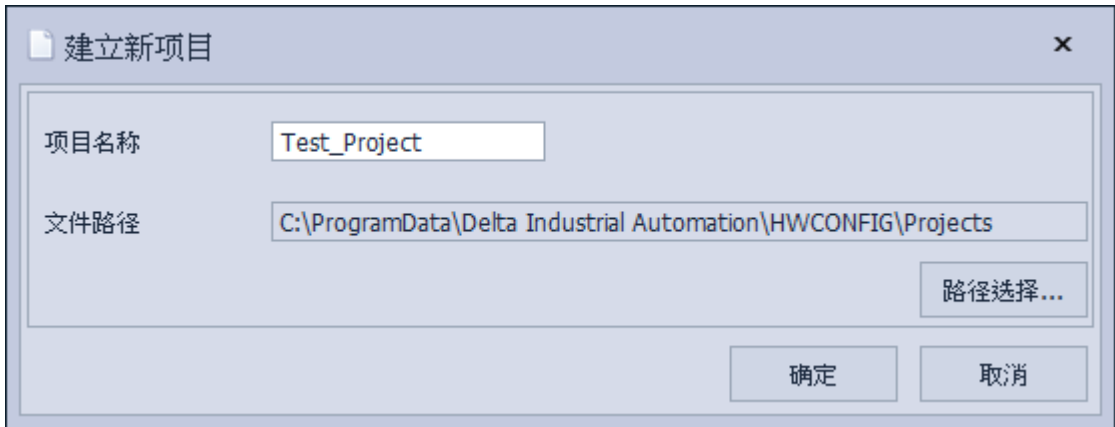


● 新增项目

新增项目只支持单机版。开启新项目。

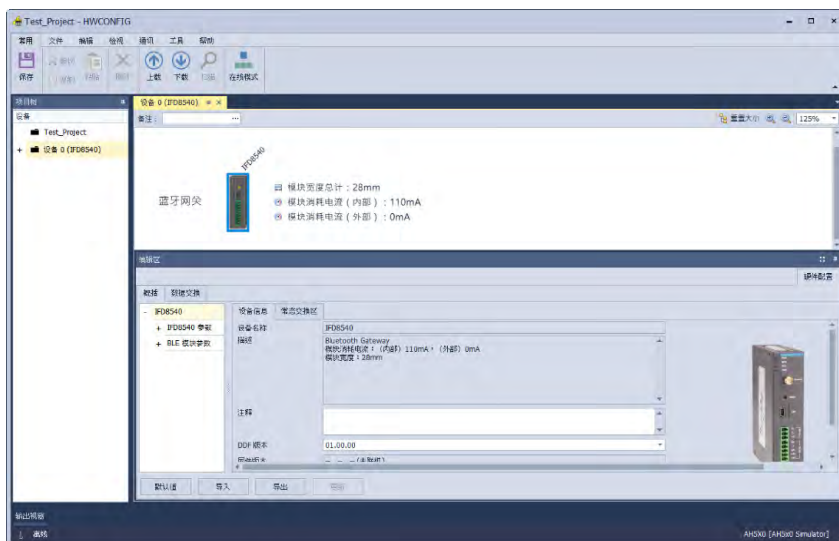
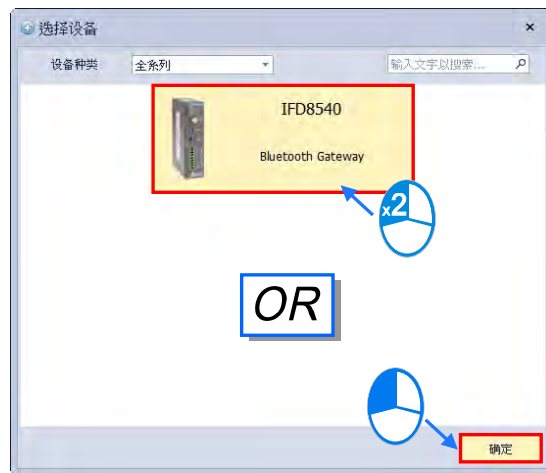
在单机版软件中单击「菜单工具栏」中的「文件」的「新增项目」，或使用快捷键 **Ctrl+N**，单击后会先询问是否要保存当前的项目「是 (Y)」为保存，「否 (N)」为不保存，「取消」为取消开启新项目。

待建立新项目的窗口出现，在设定窗口中依序输入「项目名称」及「文件路径」，完成后单击「确定」。



即会出现选择装置窗口，在「装置类型」的下拉菜单中选择机种类型后，在所选取的装置双击或单击 Enter 键，也可以在选定装置后单击「确定」，即可进入操作画面，如下图。

3

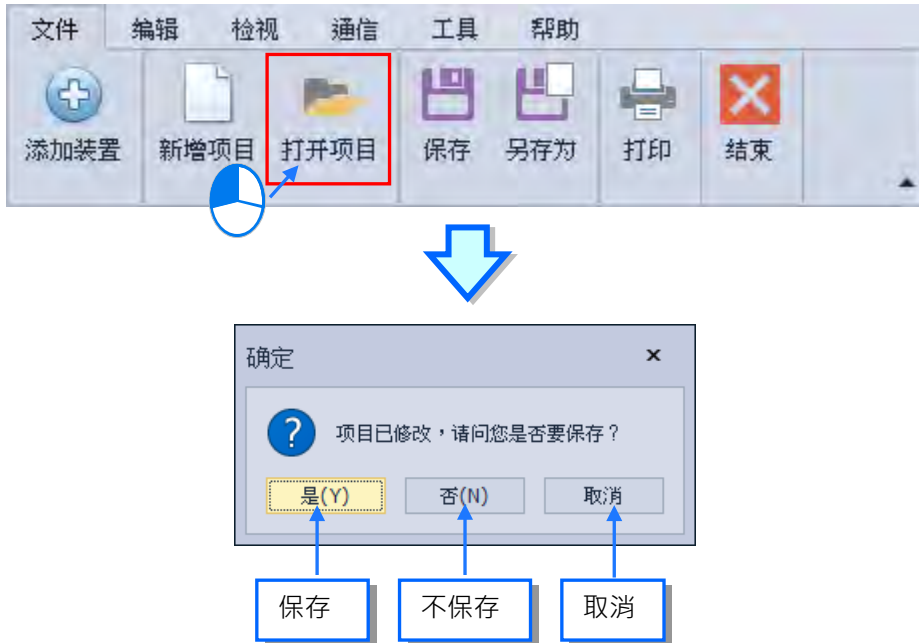


● 打开项目

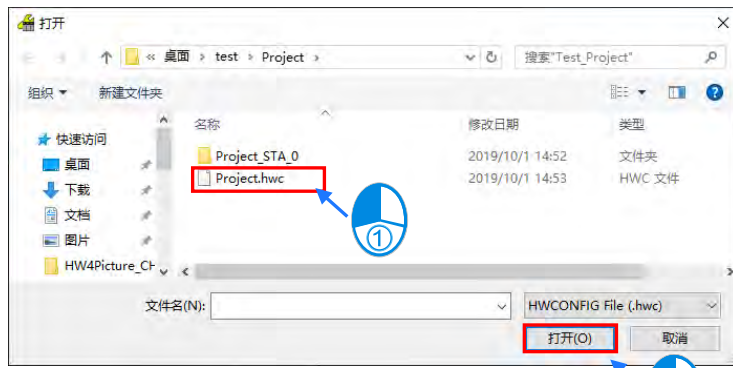
打开项目只支持单机版，打开旧项目。

在单机版软件中单击「菜单工具栏」中的「文件」的「打开」，或使用快捷键 **Ctrl+O**，单击后会先询问是否要保存当前的项目「是 (Y)」为保存，「否 (N)」为不保存，「取消」为取消打开旧项目。

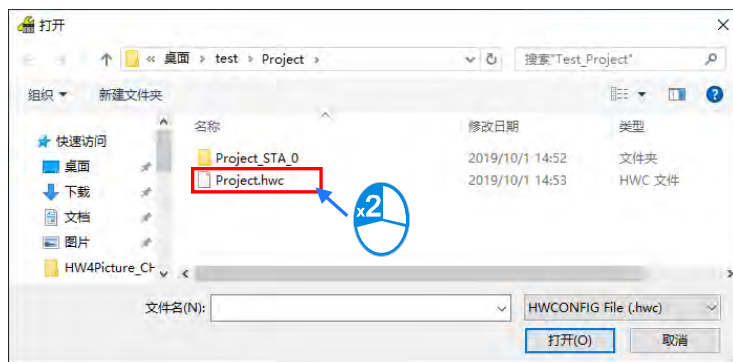
3



找到文件路径选择.hwc 文件后，单击确「打开」或是双击单击该文件，即可打开旧文件。



OR



- 保存

单击「菜单工具栏」中「文件」的「保存」，或使用快捷键 **Ctrl+S**。



保存成功后，下方输出窗口会显示项目保存成功以及保存目录。



- 另存为

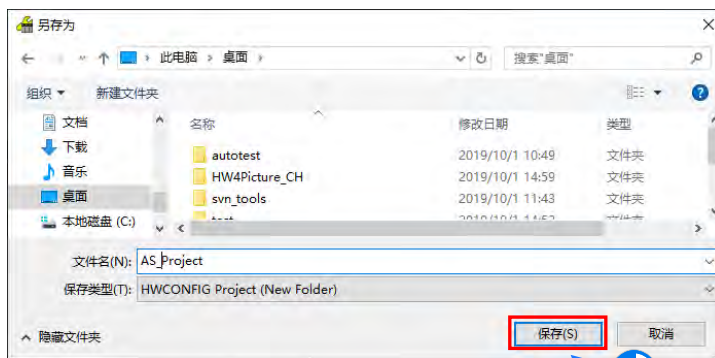
另存为只支持单机版。将项目保存在新的位置。

单击「菜单工具栏」中的「文件」的「另存为」，或使用快捷键 **Ctrl+Alt+S**。

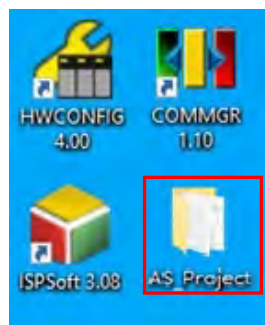


3

等待另存新文件画面跳出后，选择要保存的位置，在「文件名 (N)」内输入文件名后，单击「保存 (S)」，即将文件保存在新的文件夹。



3

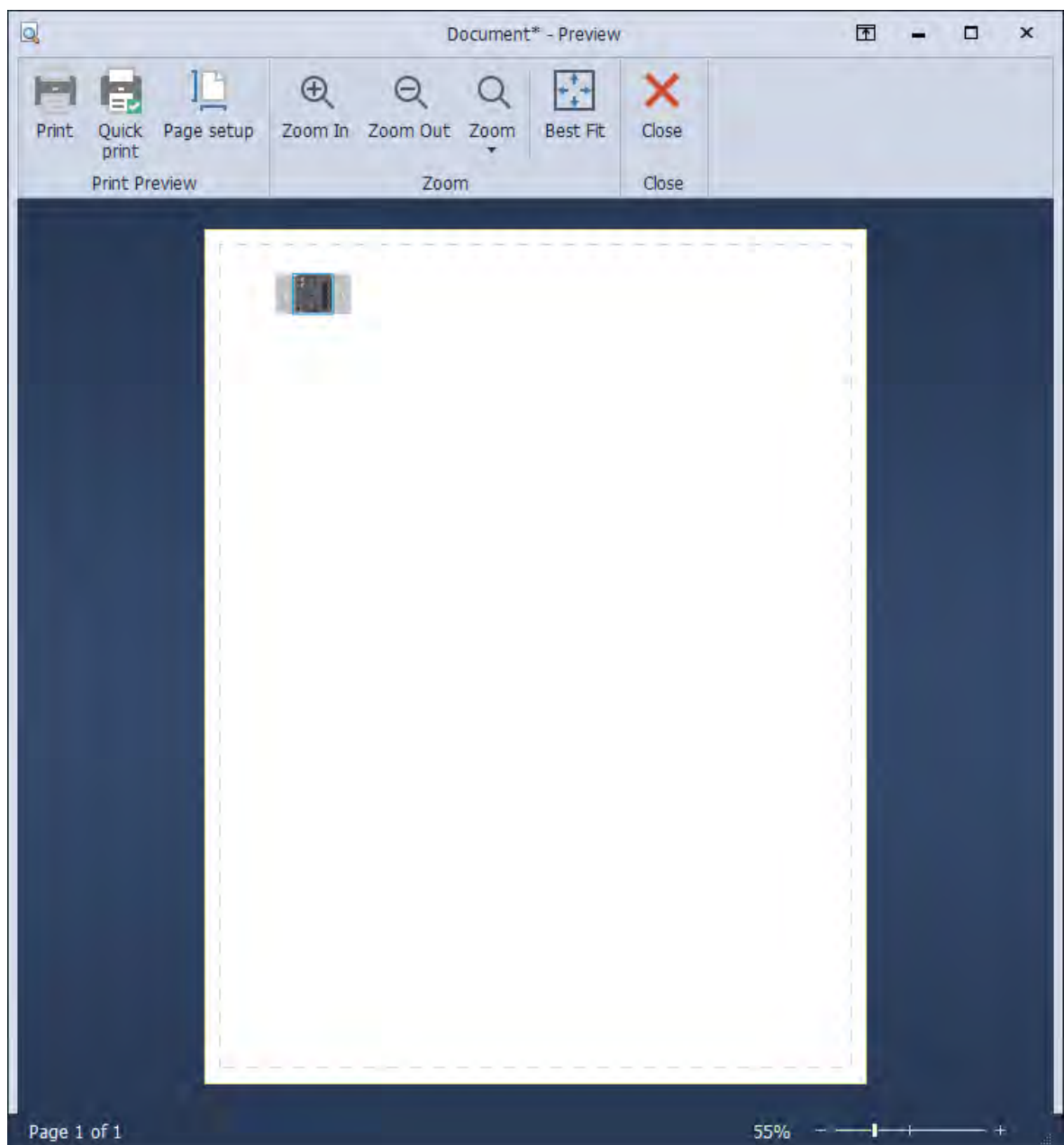


- 打印

打印项目硬件配置图。

单击「菜单工具栏」中的「文件」的「打印」，或使用快捷键 **Ctrl+P**，即可进入打印设定窗口。





3

➤ **Print Preview (打印设定)**

Print : 设定打印详细信息

Quick print : 快速打印

Page setup : 打印版面配置

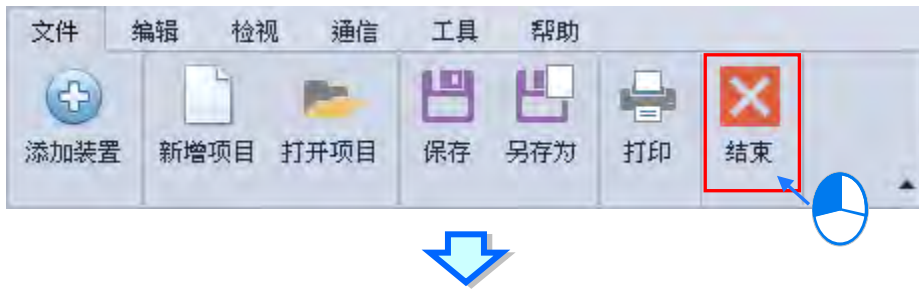
➤ **Zoom (检视显示大小)**

Zoom In : 放大显示

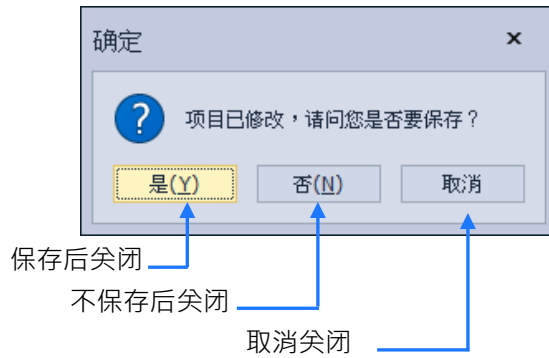
Zoom Out : 缩小显示

Zoom : 调整显示比例

Best Fit : 根据窗口页面调整显示大小



3



3.1.2.3 菜单工具列-编辑

编辑：主要提供编辑工作时所需的操作功能。



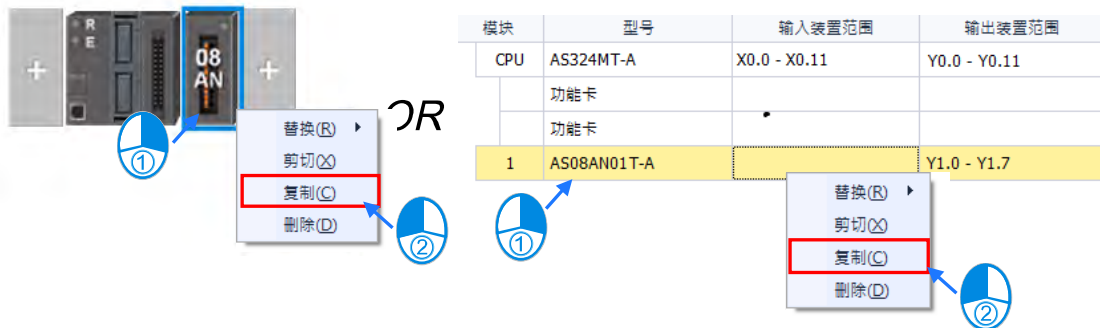
操作项	功能说明
粘贴	粘贴已复制或已剪切的模块与功能块。
剪切	剪切所选取的模块与功能块。
复制	复制所选取的模块与功能块。
删除	删除所选取的模块与功能块。

- **复制**

复制模块的方法有下列两种：

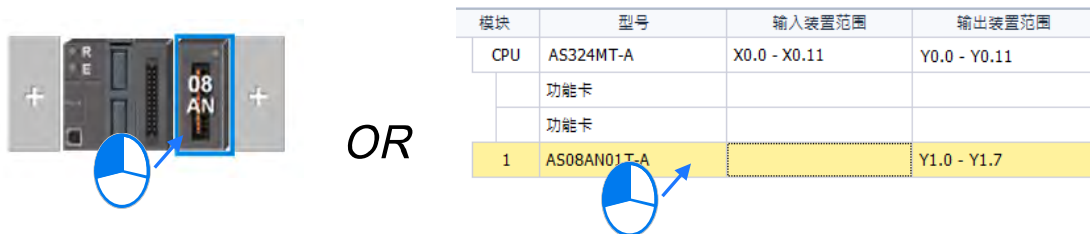
➤ 方法一

在硬件配置区的模块配置图或模块配置信息上选取要复制的模块后右击，之后在快捷菜单中单击「复制」。

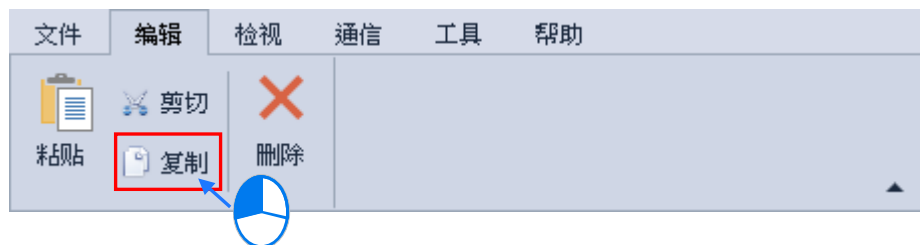


➤ 方法二

(1) 先在硬件配置区的模块配置图或模块配置信息上单击要复制的模块。



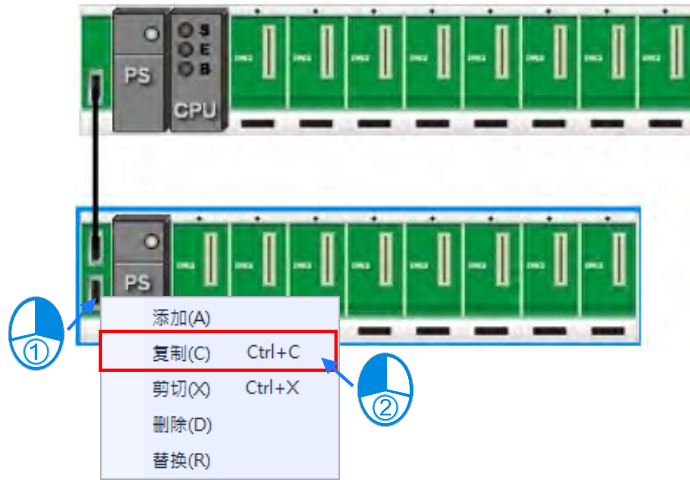
(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「复制」，或使用快捷键 **Ctrl+C**。



复制扩展背板的方法有下列两种：

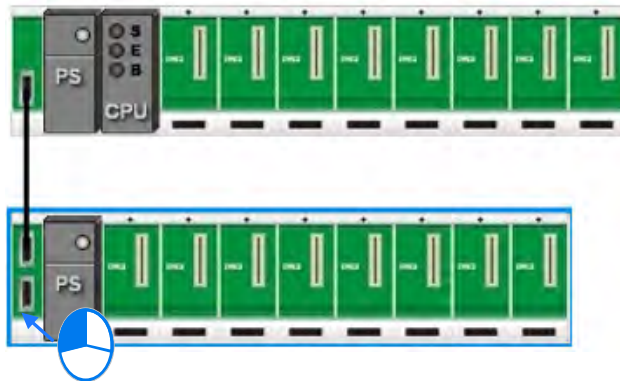
➤ 方法一

在硬件配置区的模块配置图中单击要复制的背板左侧后右击，之后在快捷选单中单击「复制」。



➤ 方法二

(1) 先在硬件配置区的模块配置图上单击要复制的扩展背板左侧。



(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「复制」，或使用快捷键 **Ctrl+C**。

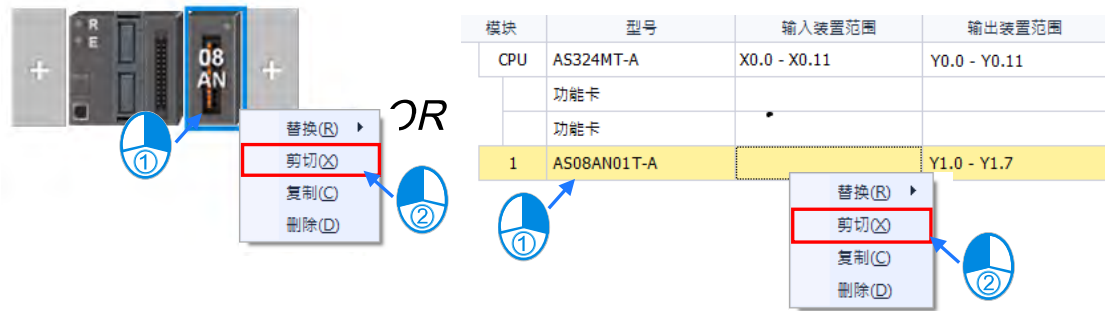


● 剪切

剪切模块的方法有下列两种：

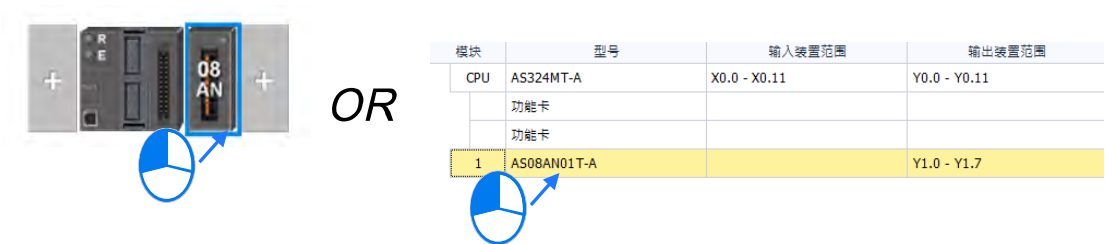
➤ 方法一

在硬件配置区的模块配置图或模块配置信息上单击要剪切的模块后右击，之后在快捷菜单中单击「剪切」。

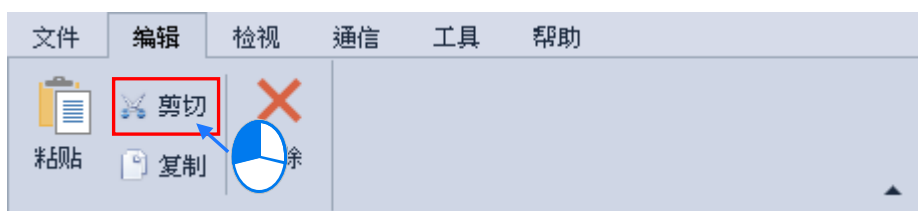


➤ 方法二

(1) 先在硬件配置区的模块配置图或模块配置信息上单击要剪切的模块。



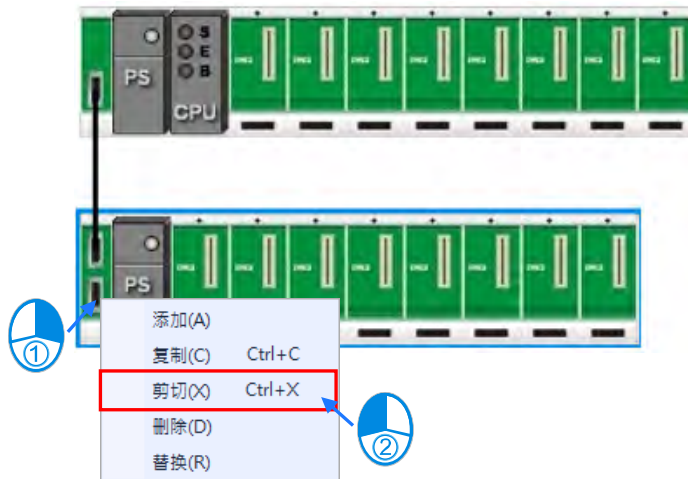
(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「剪切」，或使用快捷键 **Ctrl+X**。



剪切扩展背板的方法有下列两种：

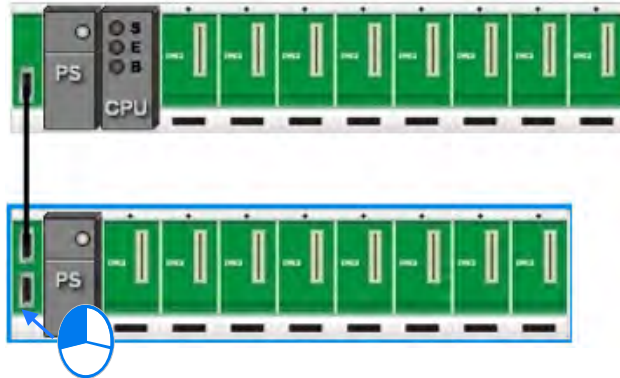
➤ 方法一

在硬件配置区的模块配置图中单击要剪切的背板左侧后右击，之后在快捷选单中单击「剪切」。



➤ 方法二

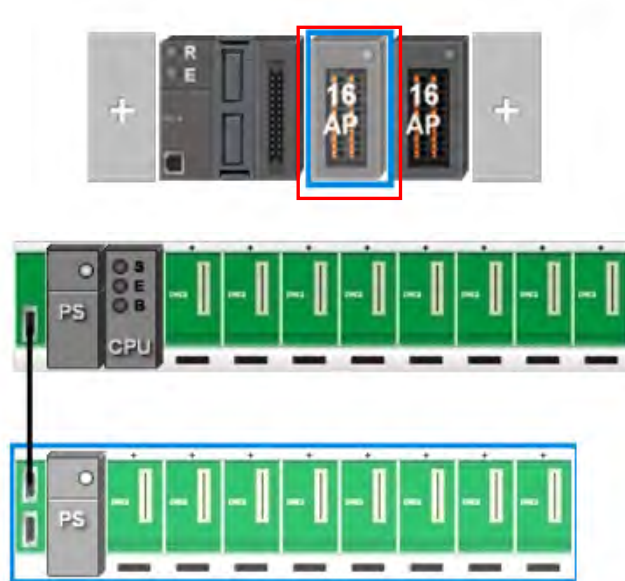
(1) 先在硬件配置区的模块配置图上单击要剪切的的扩展背板左侧。



(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「剪切」，或使用快捷键 **Ctrl+X**。



而被剪切的对象会呈现反灰的状态，若要解除反灰状态，则必须再对该对象使用复制、粘贴或删除；或选取其它对象使用剪切、复制或粘贴也可以解除反灰状态。



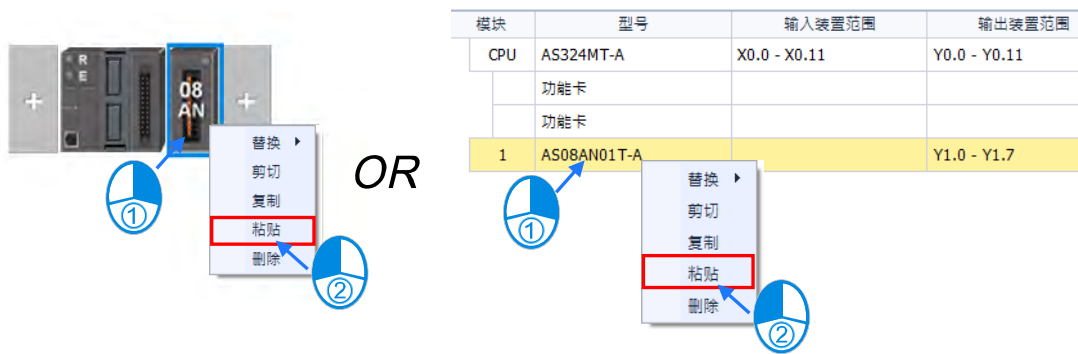
3

● 粘贴

粘贴模块的方法有下列两种：

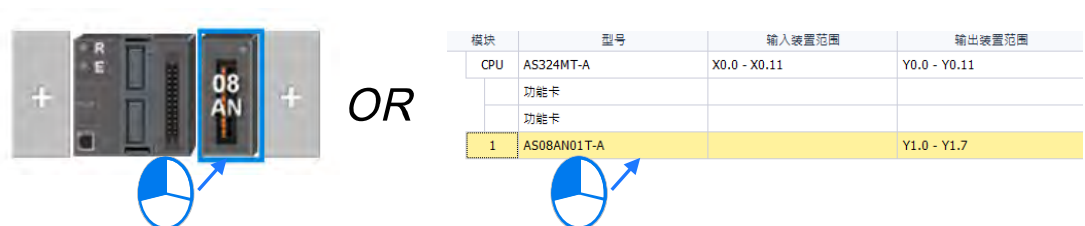
➤ 方法一

在硬件配置区的模块配置图或模块配置信息上单击要粘贴的插槽或位置后右击，再在快捷菜单中单击「粘贴」。



➤ 方法二

(1) 先在硬件配置区的模块配置图或模块配置信息上单击要粘贴的插槽。



(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「粘贴」或使用快捷键 **Ctrl+V**。

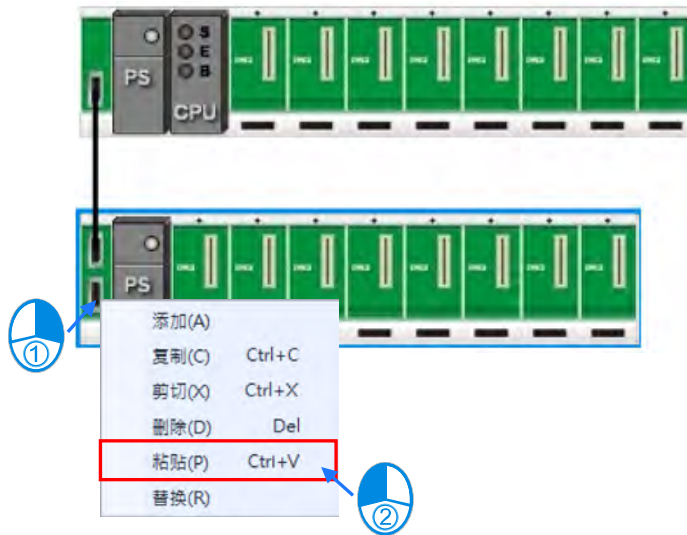


粘贴扩展背板的方法有下列两种：

3

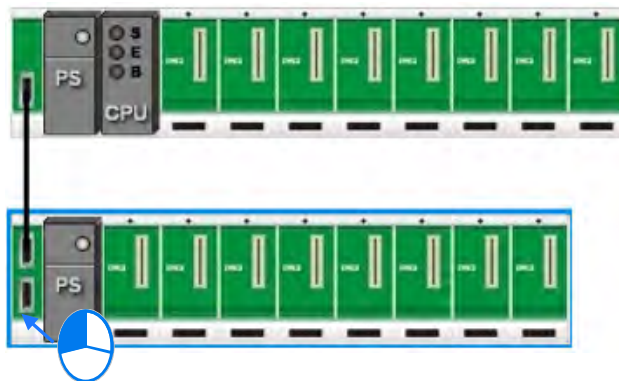
➤ 方法一

在硬件配置区的模块配置图上单击要插入的位置后右击，再在快捷选单中单击「粘贴」。



➤ 方法二

(1) 先在硬件配置区的模块配置图上单击要插入的位置。

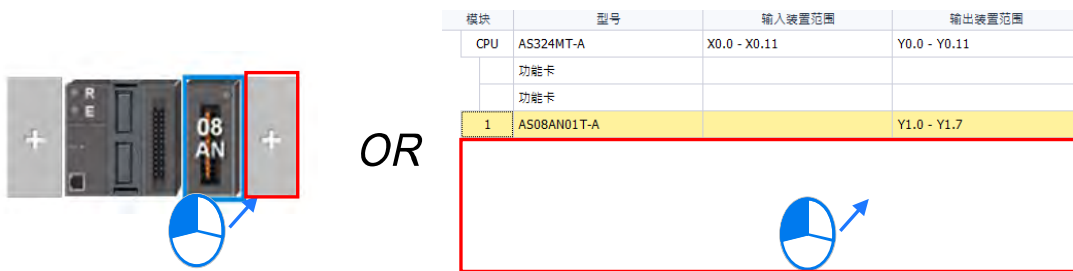


(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「粘贴」或使用快捷键 **Ctrl+V**。

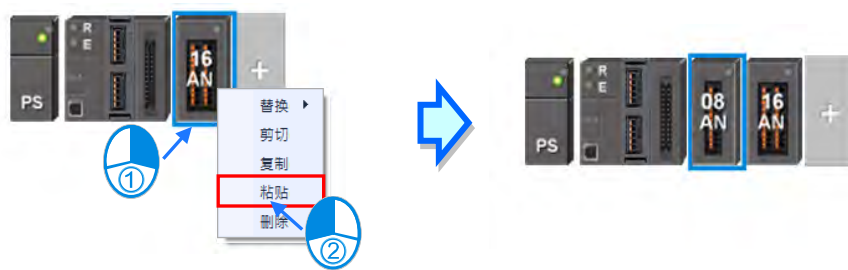


AS 系列机种为无背板设计，若不选定位置粘贴模块只能贴在最右方的位置，与模块配置区的最下方的空白列。

3

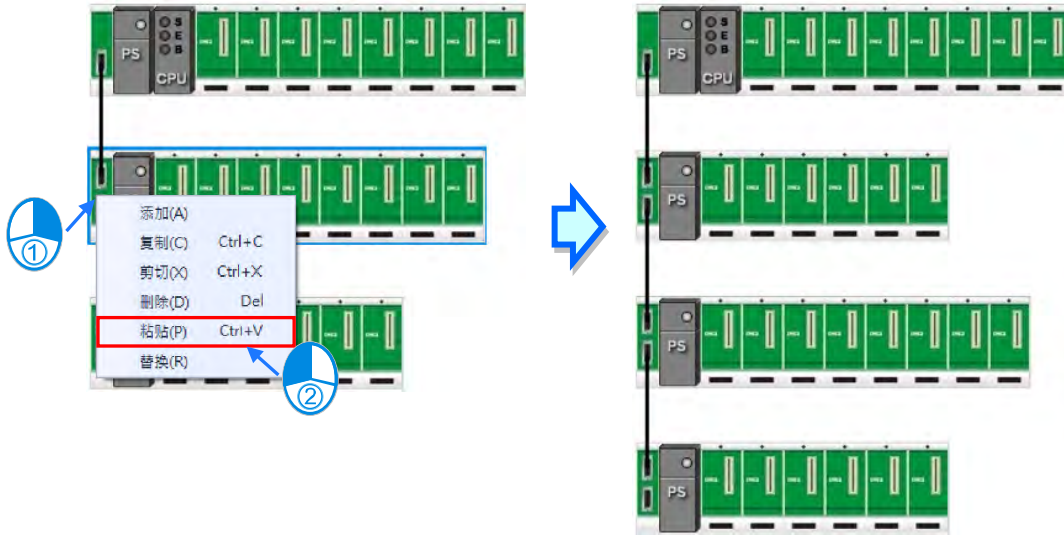


若选取的位置已经有模块，则会粘贴在选取位置的左侧，其余模块会往右侧移动，如下图，在 16AN 模块上单击快捷菜单的「粘贴」，将 08AN 模块粘贴至选取的位置并将 16AN 模块往右侧移动。



若选取的位置已经有扩展背板，则会粘贴在选取位置的上方，其余模块会往下方移动，如下图，在 8 个插槽的扩展背板上单击快捷选单的「粘贴」，将 6 个插槽的扩展背板粘贴至选取的位置并将 8 个插槽的扩展背板往下方移动。

3



补充说明

当模块进行粘贴的操作时，模块参数的处理规则如下。

- 输入/输出装置范围：模块的装置范围皆会由右侧第一个模块开始排序，并不会重复，在被粘贴的模块，会自动变更对应装置至可用的地址。

模块参数设定也可以被复制、粘贴如下图范例，被粘贴的模块参数会与复制的模块相同。

参数名称	数值
通道 1 模式设定	1V~+5V
通道 2 模式设定	0V~10V
通道 3 模式设定	-5V~+5V
通道 4 模式设定	0V~+5V

参数名称	数值
通道 1 模式设定	1V~+5V
通道 2 模式设定	0V~10V
通道 3 模式设定	-5V~+5V
通道 4 模式设定	0V~+5V

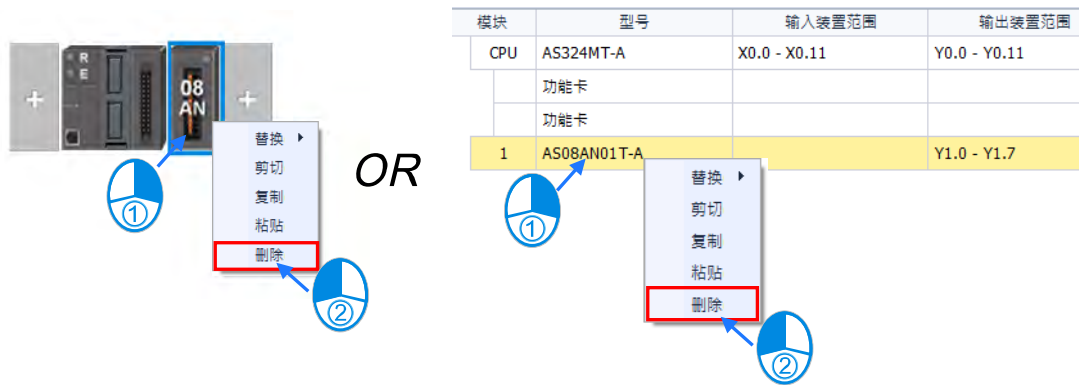
模块	型号	输入装置范围	输出装置范围
电源	AS-PS02A		
CPU	AS324MT-A	X0.0 - X0.11	Y0.0 - Y0.11
	AS-F422		
	功能卡		
1	AS04AD-A	D28000 - D28019	
2	AS04AD-A	D28020 - D28039	

● 删除

删除模块的方法有下列两种：

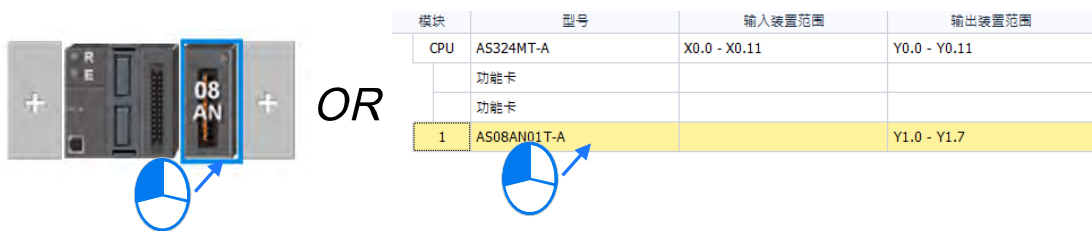
➤ 方法一

在硬件配置区的模块配置图或模块配置信息上单击要删除的模块后右击，之后在快捷菜单中单击「删除」。



➤ 方法二

(1) 先在硬件配置区的模块配置图或模块配置信息上单击要删除的模块。



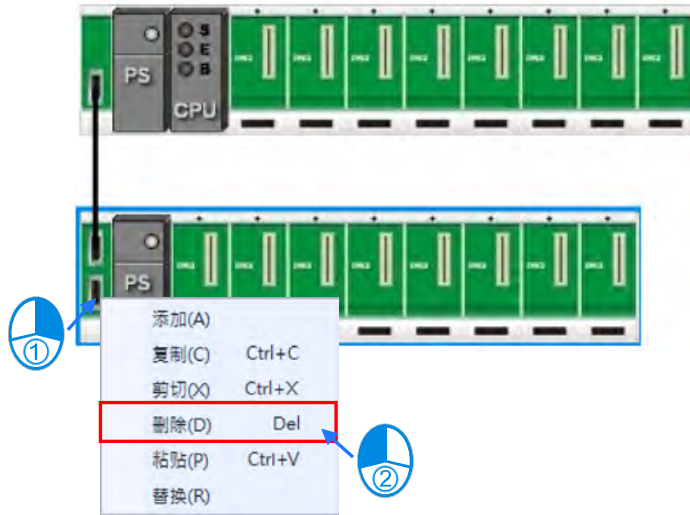
(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「删除」，或使用 **Delete** 键。



删除背板的方法有下列两种：

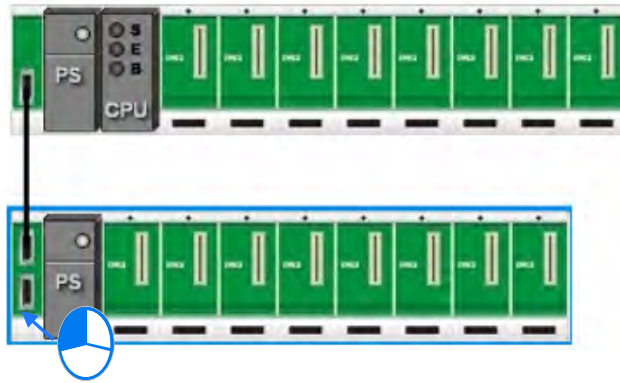
➤ 方法一

在硬件配置区的模块配置图上单击要插入的位置后右击，再在快捷选单中单击「删除」。



➤ 方法二

(1) 先在硬件配置区的模块配置图上单击要删除的扩展背板。

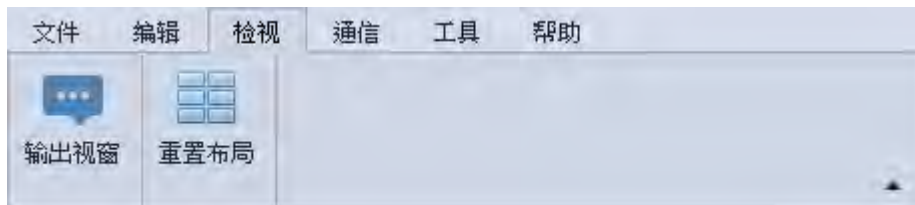


(2) 在 HWCONFIG 的菜单工具栏单击「编辑」的「删除」，或使用 **Delete** 键。



3.1.2.4 菜单工具列-检视

检视：主要功能为检视项目中的信息以及设定工作环境的显示方式。



操作项	功能说明
输出窗口	打开或关闭输出窗口。
重置布局	重置 HWCONFIG 版面配置。

3

3.1.2.5 菜单工具列-通讯

通讯：主要为通过 HWCONFIG 对 PLC 主机进行联机操作或设定的功能。



操作项	功能说明
上载	将 PLC 中的参数与模块设定传输至 HWCONFIG。
下载	将 HWCONFIG 项目中的参数与模块设定传输至 PLC。
扫描	扫描 PLC 目前的模块配置及设定。
在线模式	实时监控模块内的信息。
通信设置	设定通讯连接的方式。

● 上载/下载

在 HWCONFIG 当中所做的设定，必须被下载至主机中，才可正式生效。

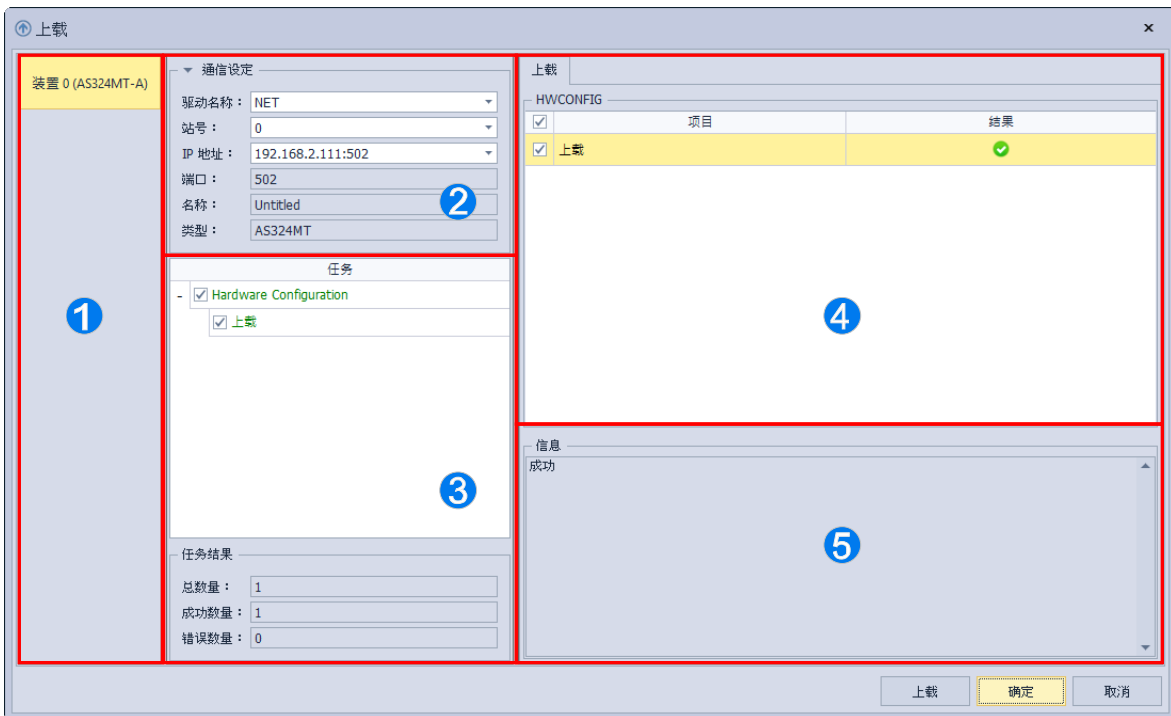
- (1) 单机版 HWCONFIG 在进行参数上下载之前，请先确认已将通信设置完成。
- (2) 若由 ISPSOFT 打开 HWCONFIG，HWCONFIG 会直接引用 ISPSOFT 的通信设置，因此进行参数上下载之前，请务必确认 ISPSOFT 与主机正常联机。执行时，若主机目前的卷标名称与参数设定的名称不符时，画面亦会出现提示信息。

- (3) 在 HWCONFIG 硬件配置上载/下载，内容不包含 EtherNet/IP 与 EtherCAT，若要设定 EtherNet/IP 与 EtherCAT，请先开启各分页，再单击上载/下载，详细设定请参考第 3.1.5 节及第 3.1.6 节。

➤ 上载系统硬件参数

HWCONFIG 进行下载时，先在菜单工具栏中单击「通讯」的「上载」，或使用快捷键 **Ctrl+F9**，后便会会出现上载窗口。

3



- ❶ 装置：显示目前所编辑的装置名称与装置型号。
- ❷ 通信设置：可以进行通信设置，以及显示目前所通讯的机型与 PLC 名称。
- ❸ 上载任务：为选取项目；为不选取项目。
- ❹ 上载任务之细节：可勾选要上载的细节项目，右方会显示上载结果。
- ❺ 信息：显示上载结果信息。

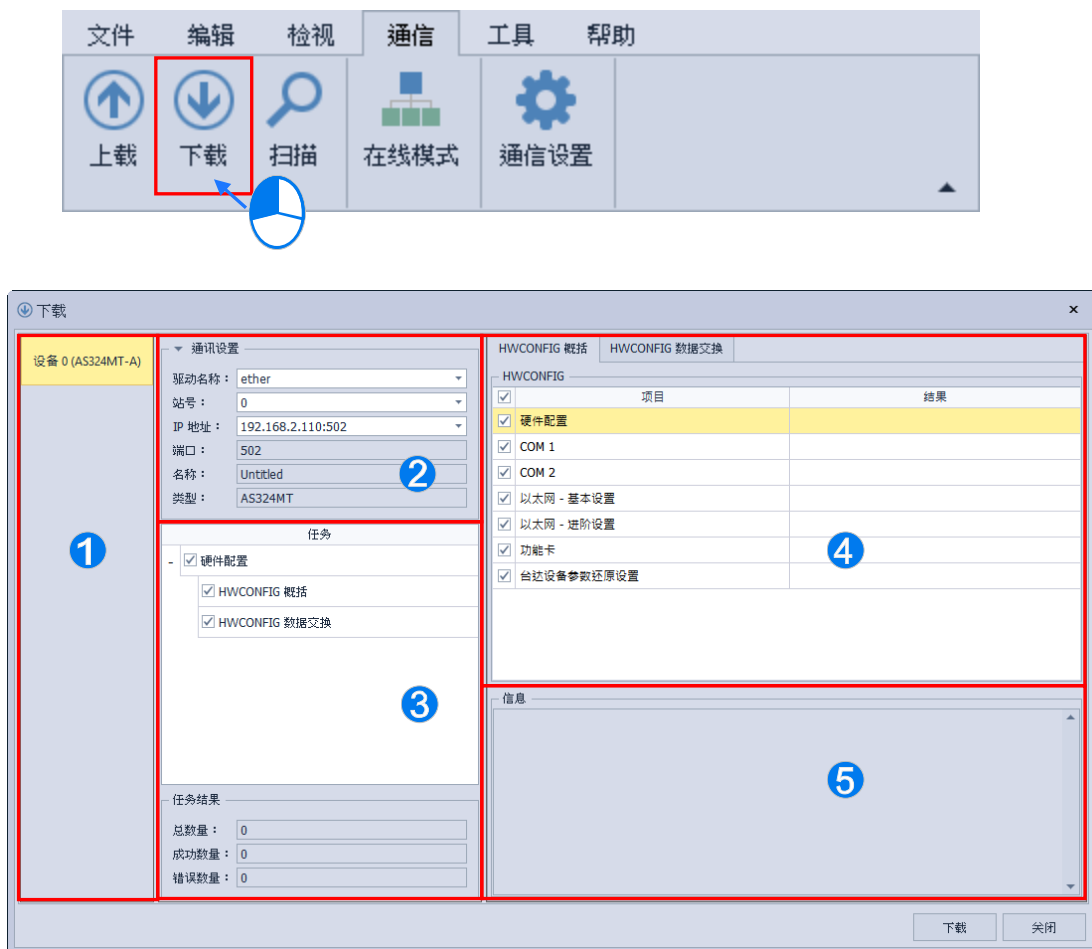
*1. 由 ISPSOft 软件所开启的 HWCONFIG，会直接引用 ISPSOft 的通信设置，故通信设置无法做变更，如果要更改通讯需先存盘关闭 HWCONFIG，在 ISPSOft 更改通信设置后，再次打开 HWCONFIG 即可。

上载窗口开启后，首先进行通信设置，接着单击「上载」按钮后待处理窗口结束后，便可将参数上载至主机中，上载结束后信息会出现成功，最后单击「确定」才会更新项目内的模块，单击「取消」则不会更改项目内的模块，其中上载项目包括模块配置、模块内部参数及 PLC 主机参数。

执行上传操作时，所有的主机参数将会全部被上传回来；而执行下载操作时，画面则会先出现勾选窗口，用户可在窗口中勾选想要下载的参数主类别，而无须将所有的设定全部下载至主机中。

● 下载系统硬件参数

HWCONFIG 进行下载时，先在菜单工具栏中单击「通信」的「下载」或使用快捷键 **Ctrl+F8**，之后便会出现下载窗口。



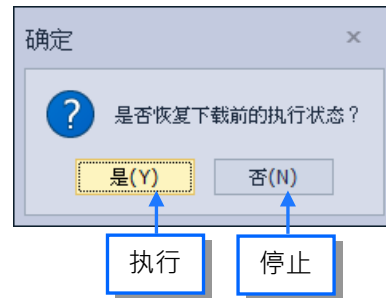
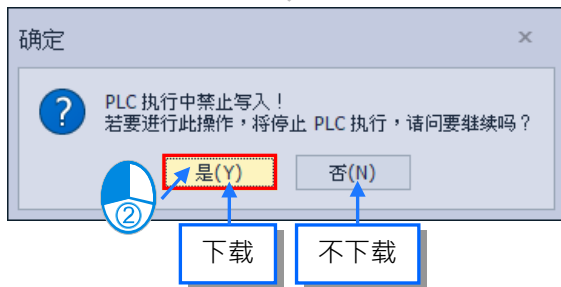
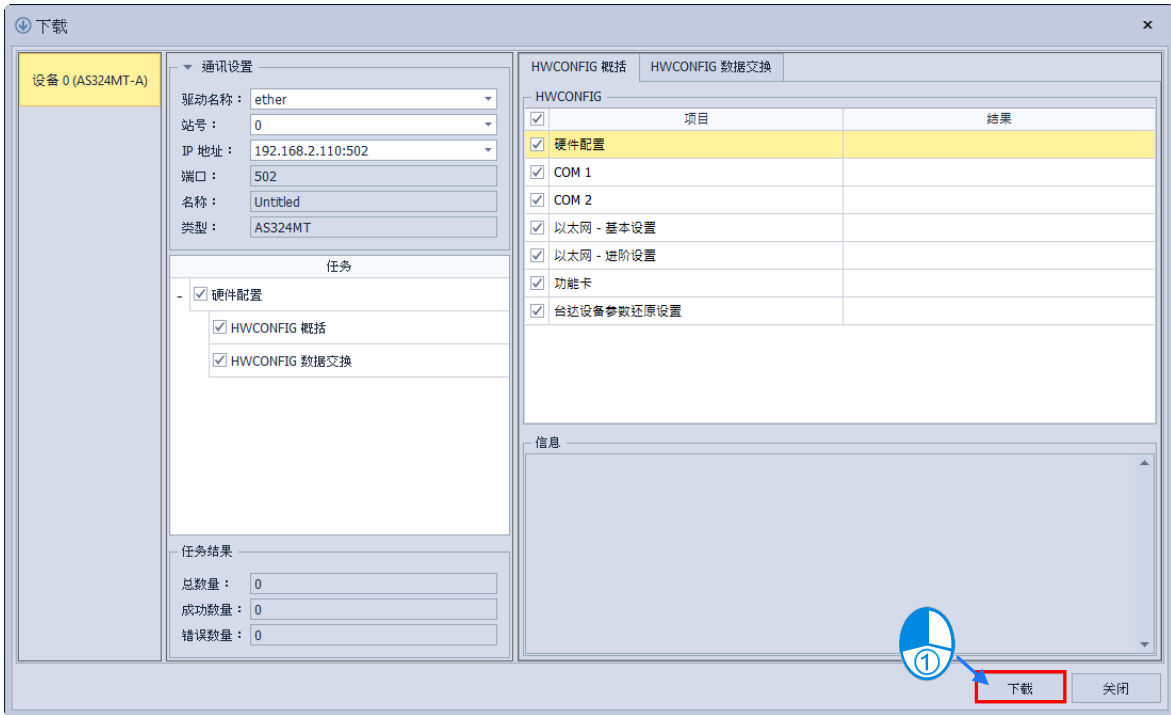
- ❶ 装置：显示目前所编辑的装置名称与装置型号。
- ❷ 通信设置：可以进行通信设置，以及显示目前所通讯的机型与 PLC 名称。
- ❸ 下载任务：为选取项目；为部分选取项目；为不选取项目。
- ❹ 下载任务之细节：可勾选装置系统设定以及数据交换的细节项目，右方会显示下载结果。
- ❺ 信息：显示下载结果的信息。

*1. 由 ISPSOFT 软件所打开的 HWCONFIG，会直接引用 ISPSOFT 的通信设置，故通信设置无法做变更，如果要更改通讯需先保存并关闭 HWCONFIG，再在 ISPSOFT 更改通信设置后，再次打开 HWCONFIG 即可。

下载窗口开启后，首先进行通信设置，接着可视需求勾选要下载的项目，单击「下载」按钮后待处理窗口结束后，便可将参数下载至主机中。

下图为下载窗口，下载时的主机状态必须为 STOP；若主机处于 RUN 的状态时，系统便会在开始下载前先显示提示信息，单击确定后，接着会询问是否恢复 PLC 的运行状态。

3



*1. 当主机目前的卷标名称与参数设定的名称不符时，在开始执行上传或下载之前会先显示提示信息。

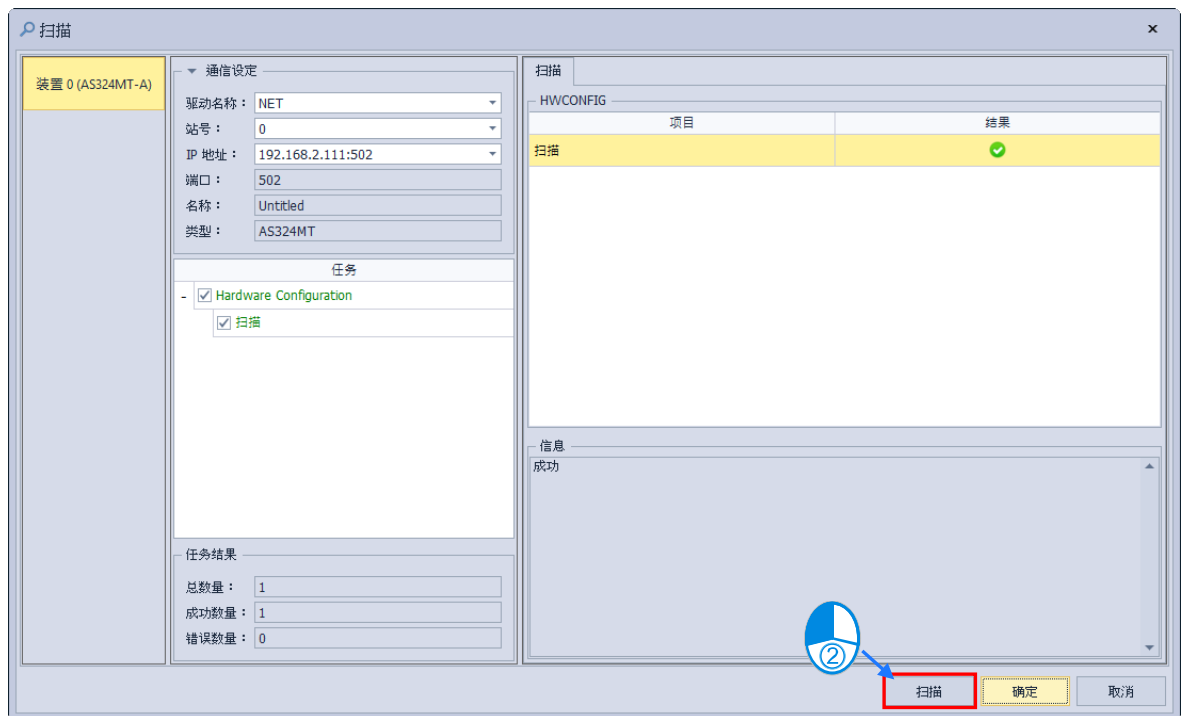
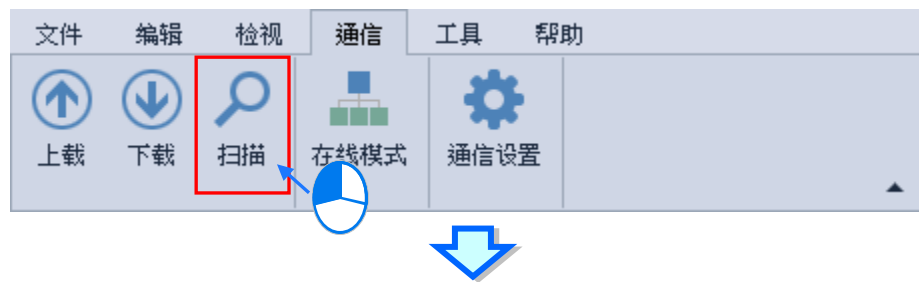
*2. HWCONFIG 会引用 ISPSOft 的通信设置，因此进行上下下载前请确认 ISPSOft 已与主机正常联机。

- 扫描



当用户已经完成硬设备配置并确认可以与主机正常通讯时，再利用 HWCONFIG 的「扫描」功能，将实际的模块配置状态扫描上来，如此便可省去逐一增加模块的时间；而进行操作之前，请确认 ISPSOFT 已经可以与主机正常联机。

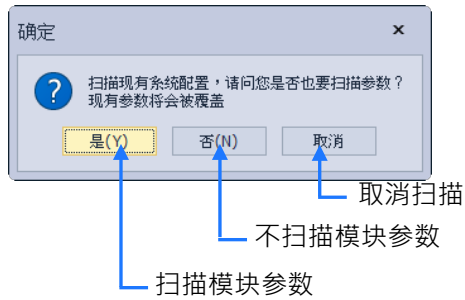
操作时请在 HWCONFIG 的菜单工具栏中单击「通信」的「扫描」或使用快捷键 **Ctrl+F7**；而检测的内容则包括模块型号、模块配置、模块固件版本，但不包含主机参数；且在执行扫描后，原本的模块配置与模块参数设定都将会被覆盖，而主机本身的参数设定，则不会受到影响。



部分机种可支持模块参数扫描，单击扫描后即会询问是否要扫描参数（没询问为不支持此功能），若选择「是」，则会将检测主机右侧模块（不包含主机内部参数*1），并检测包括模块型号、模块配置、背板配置、模块韧体版本，且覆盖目前软件的配置，当检测完成后，会出现完成窗口，单击确定后，即完成检测动作，此时实际的模块配置状态将会显示在硬件配置区，且会自动重新配置所有模块的输入/输出装置范围；若选择「否」，则模块的参数会为系统默认值。

*1.主机内部参数包括功能卡参数。

3



扫描结束后信息会出现成功，最后单击「确定」才会更新项目内的模块，单击「取消」则不会更改项目内的模块。

若是使用 AH560 冗余系列机种，在开启冗余功能后（冗余功能的开启，请参考 3.1.3.20 节），「扫描」结果会与未开启冗余功能不同，可参考下方范例图说明（此为范例说明，实际检测结果，请考虑现场联机状况）。





3

	上载	扫描
主机设定	V	
功能卡	O	V
功能卡设定	O	
模块配置	O	V
模块设定	O	扫描模块参数
模块韧体版本	O	V

*O：若主机内部模块配置与实际模块配置不同时，上载数据也会与实际模块配置不符。

接着我们将以一个实例来说明「扫描」与「上载」之间的差异。其中「扫描」会检测目前模块与功能卡的配置状况；而「上载」则会将从存放于主机内部的主机参数与模块配置重新读取回来，但读取回来的硬件配置则可能已与实际连接的配置状况不符。

范例：

首先将下图的模块配置下载至主机中。



接下来先移除 16AN 模块接着再执行「扫描」的功能，而检测后的结果则会与实际的模块配置相符。

3

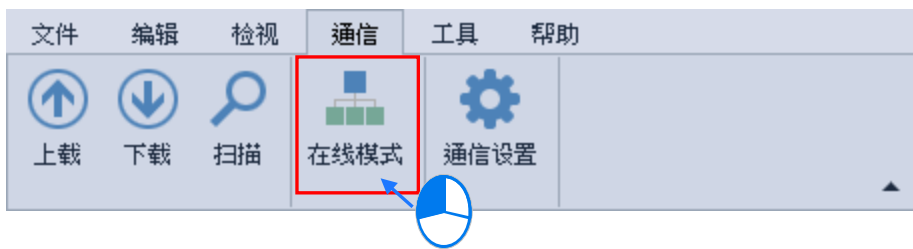


接着我们再执行「上载」的功能，而完成之后，我们会发现上载回来的模块配置与之前下载的相同。

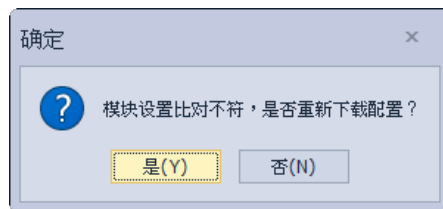


● 在线模式

(1) 在菜单工具栏中单击「通信」的「在线模式」或使用快捷键 **Ctrl+F4**，之后便可进入在线模式；而要离开在线模式时，只要再次单击「通信」的「在线模式」或使用快捷键 **Ctrl+F4** 即可结束在线模式。



(2) 在进入在线模式之前，软件会先检查存放于主机内部的硬件配置参数，并确认与模块配置区的模块设定是否一致，若两者不一致时，系统便会要求重新下载配置参数。



(3) 进入在线模式后，画面中的图示便会根据实际的模块状态而有所不同的显示；主机图标的灯号由上至下分别是 Run 与 Error。



3

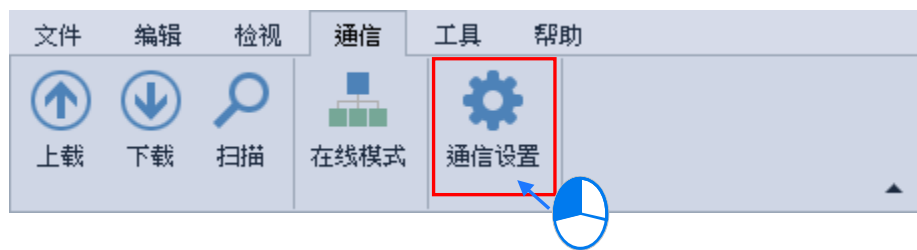
若是使用 AH560 冗余系列机种，在开启「冗余功能」，并启动在线模式后，还会显示 Master 和 Standby 状态。

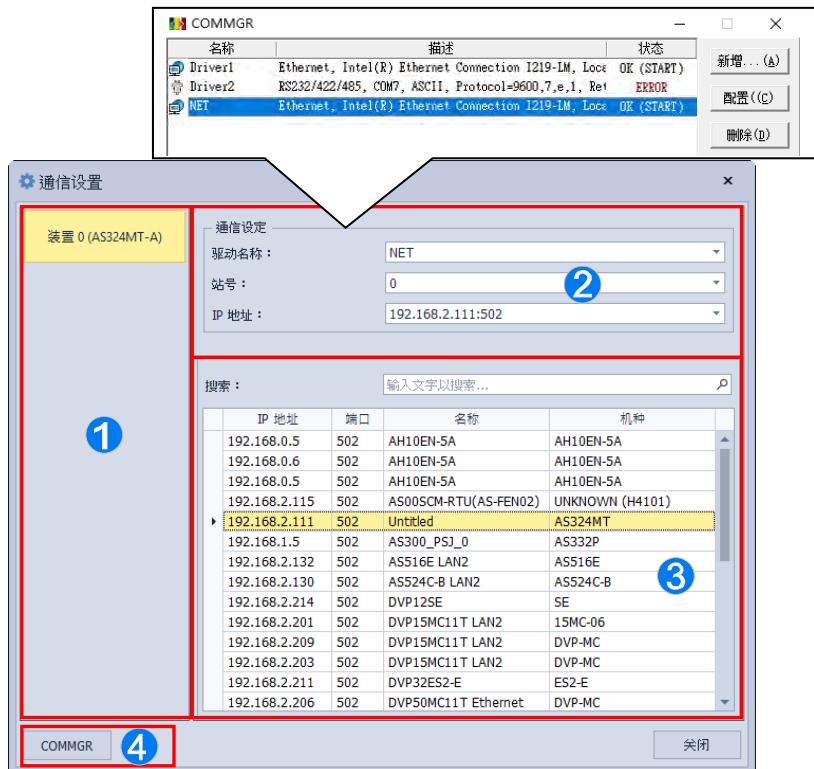


● 通信设置

通信设置只支持单机版。可与要连接的装置联机。

在菜单工具栏中单击「通信」的「通信设置」，可打开通信设置页面





- ❶ 装置：显示目前所编辑的装置名称与装置型号。
- ❷ 通信设置：可以选取 COMMGR 设定好的驱动。
- ❸ 选择联机装置：使用 Ethernet 时，可以选取要连接的主机。
- ❹ COMMGR：可打开 COMMGR 进行通信设置。

通讯窗口打开后可先打开 COMMGR 进行设定联机，接着在 ❷ 的通信设置中选择在 COMMGR 所设定的驱动，使用 Ethernet 时在 ❸ 会显示该驱动包含的主机，最后设定结束后单击「完成」即可。

3.1.2.6 菜单工具列-工具

工具：主要为 HWCONFIG 装置描述文件以及环境设定的功能。



操作项	功能说明
装置描述文件管理工具	导入 IODD、DDF 等装置描述文件。
环境设定	变更主题风格与显示语言。
订购清单	列出项目中配置所需的硬件及配件列表。

- **装置描述文件管理工具**

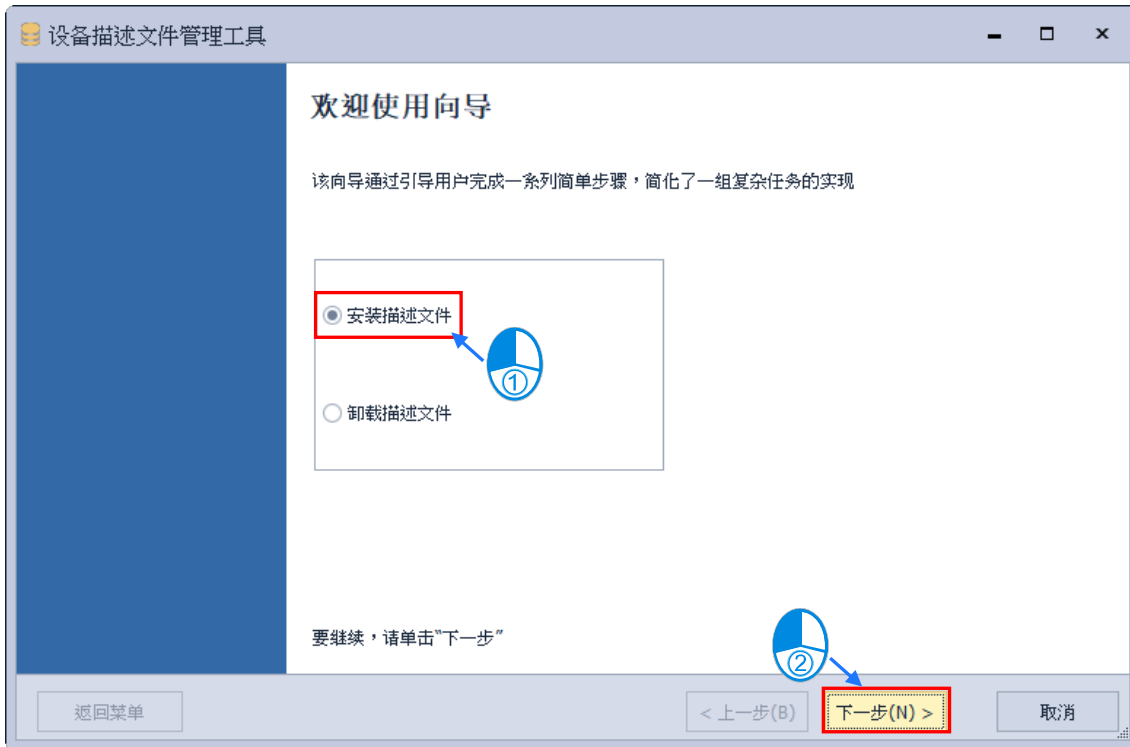
装置描述文件管理工具，可以新增或删除装置描述文件。

在菜单工具栏中单击「工具」的「装置描述文件管理工具」，即可开启装置描述文件管理页面。

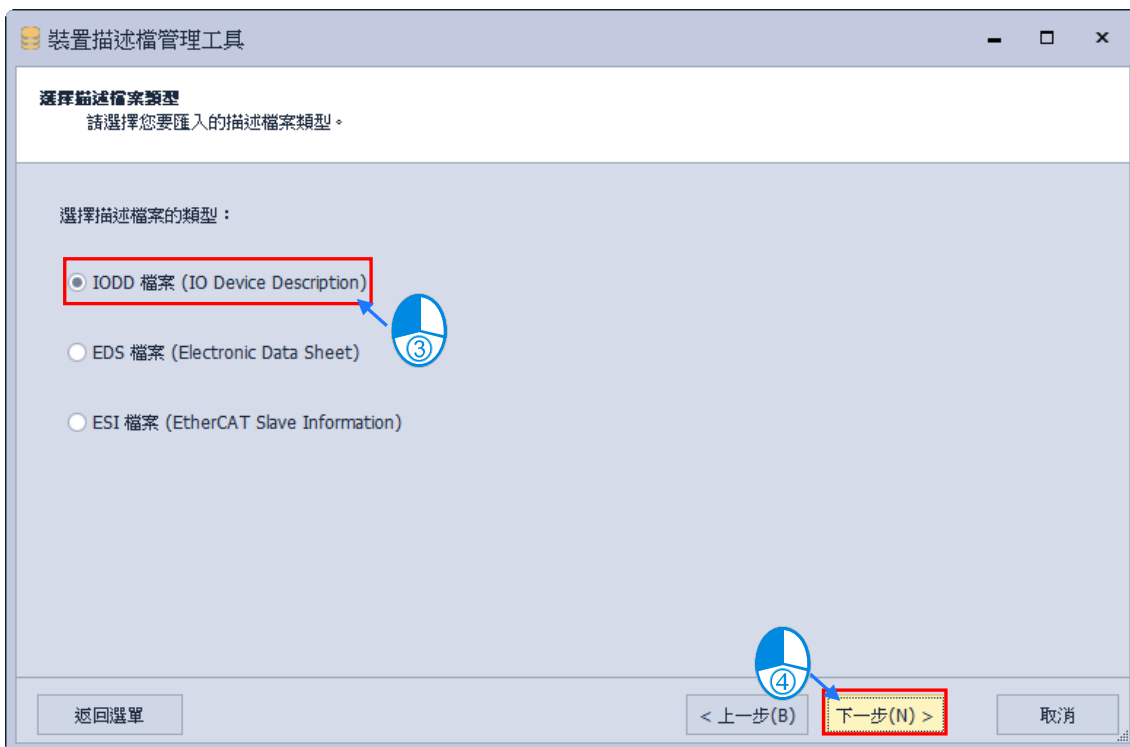


装置描述文件管理页面，可以选择「安装描述文件」或「卸载描述文件」，选好后单击「下一步」。

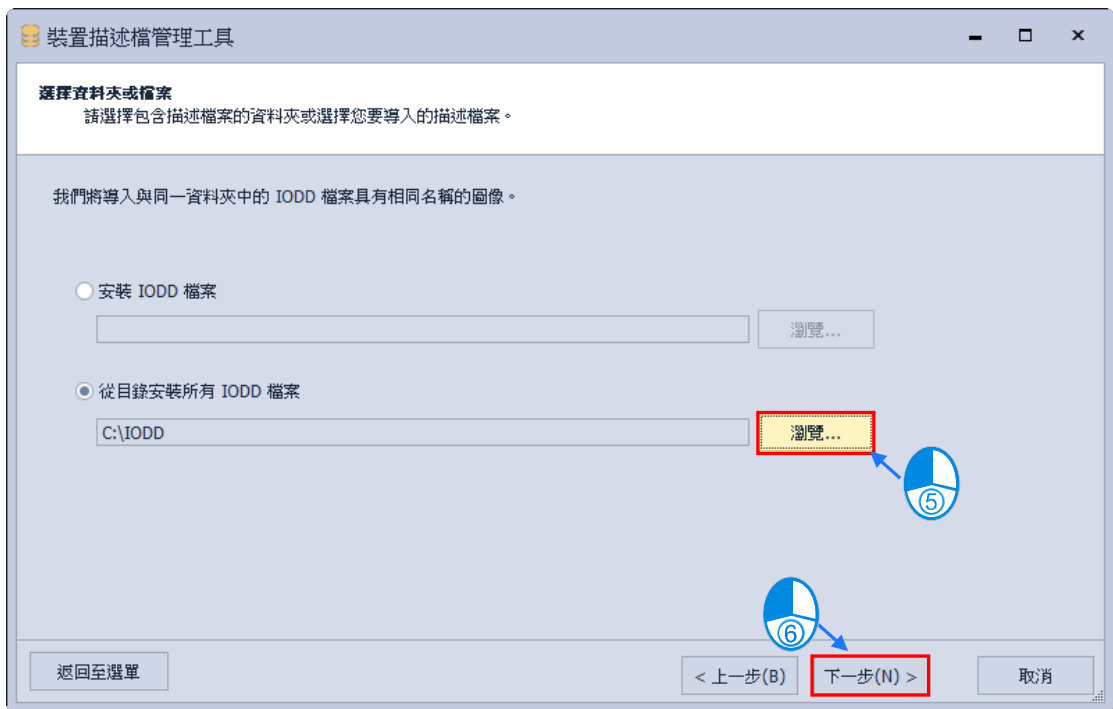
3



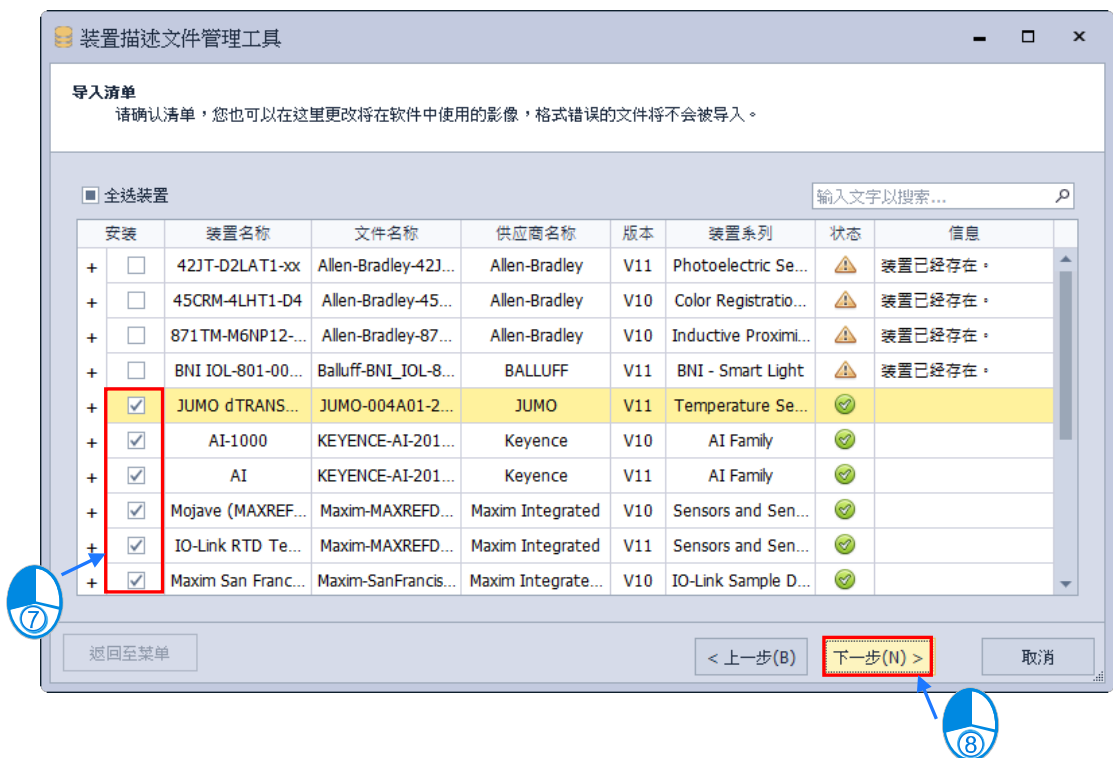
接着勾选描述文件的类型，选定好后单击「下一步」。



再来选择描述文件的路径，使用安装文件一次只能新增单一文件，选择从目录安装可以一次新增多个文件。

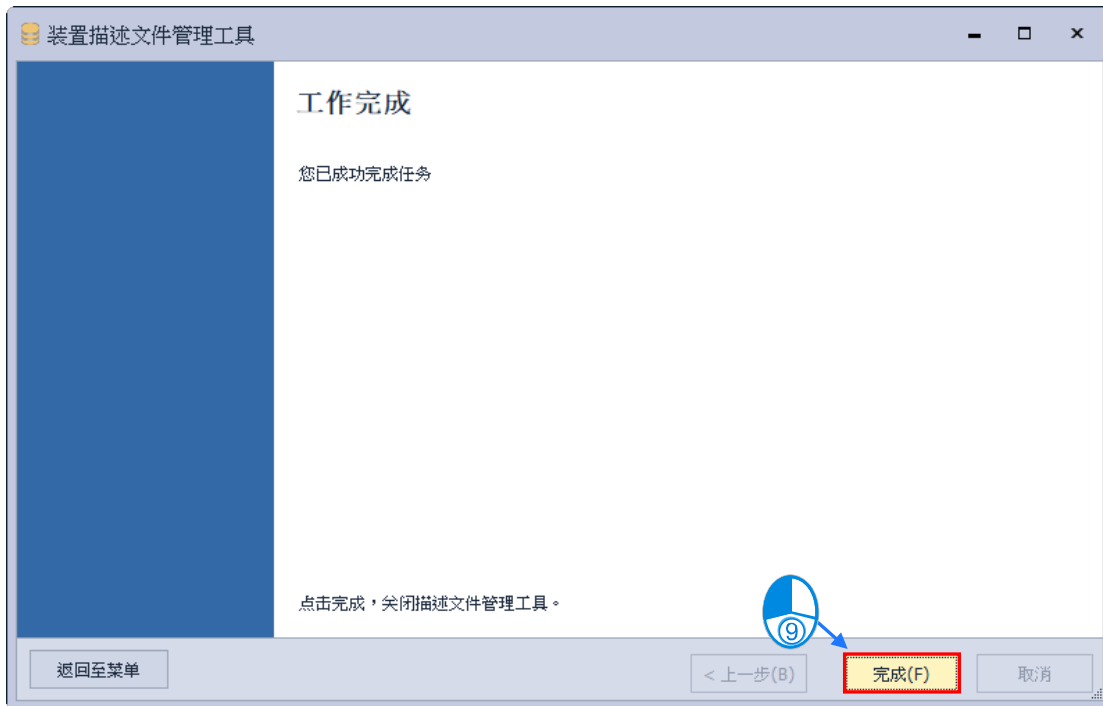


最后勾选所要加入的描述文件，单击「下一步」系统将会开始新增描述文件。



单击「完成」后则完成新增描述文件。

3

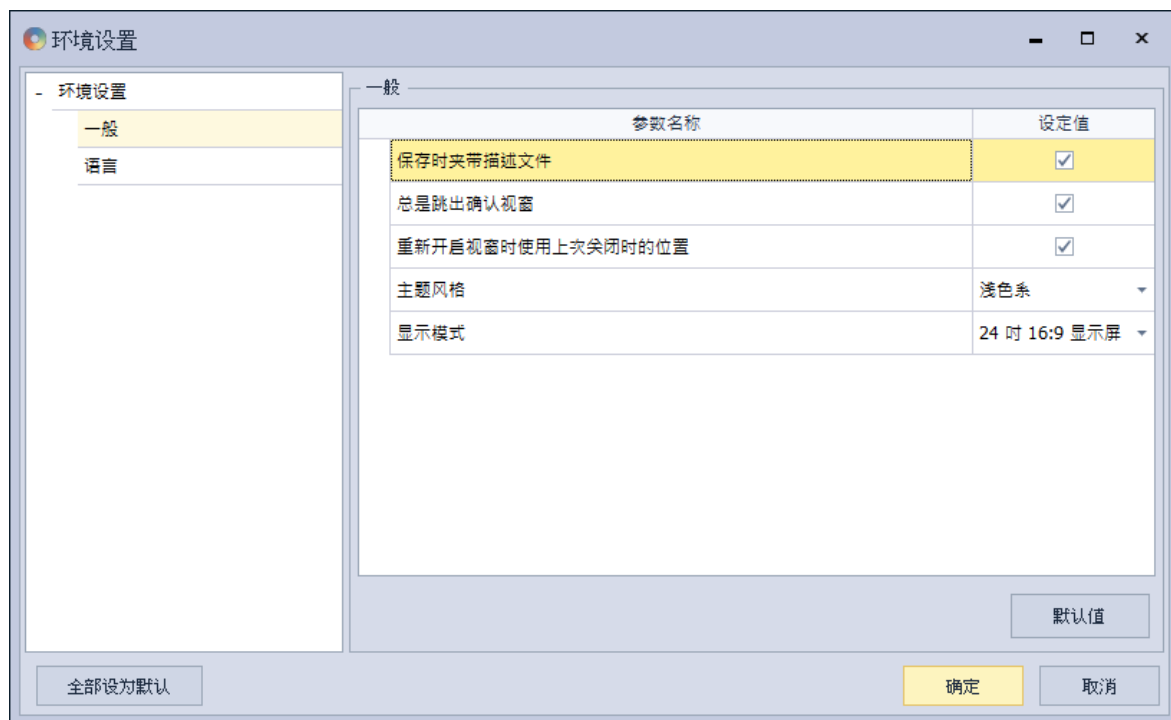


- 环境设定

在菜单工具栏中单击「工具」的「环境设定」，即可开启环境设定页面。



环境设定内有一般设定以及语言设定，下方会详细说明设定内容。



3

一般页面内的设定选项。

➤ **存档时夹带描述文件**

设定在存档时是否保存描述文件，勾选为保存描述文件；不勾选为不保存描述文件。若有文件保存描述文件，在没有该描述文件的 HWCONFIG 环境中，开启文件后描述文件会自动导入；若没有文件保存描述文件，则需要手动导入描述文件。

➤ **总是跳出确认窗口**

设定在操作变更时是否需跳出确认窗口，勾选为需跳出确认窗口；不勾选为不跳出确认窗口。

➤ **重新开启窗口时使用上次关闭时的位置及大小**

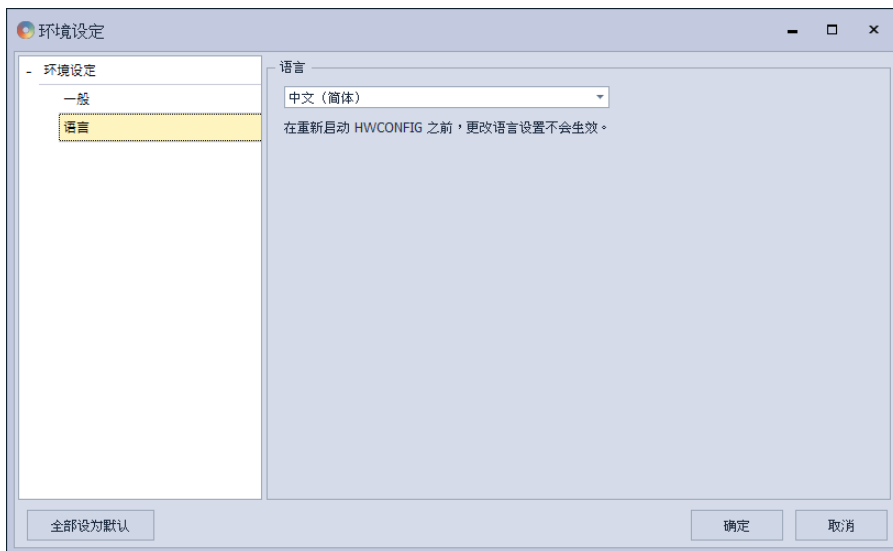
若勾选此选项，在关闭软件会记录窗口位置及大小配置；不勾选，则下次开启软件的窗口位置及大小为默认。

➤ **主题风格**

默认风格为浅色系，可更改软件风格浅色系或深色系两种。

显示模式

显示模式依照屏幕尺寸有两种选项可以做选择，第一种为 13 吋 16 : 9 屏幕，参考 HWCONFIG 3 操作模式，在设定主机及模块参数时会跳出新的窗口提供用户进行设定，建议笔电或屏幕较小时使用；第二种为 24 吋 16 : 9 屏幕，为 HWCONFIG 4 新操作模式，方便用户在同一个画面中显示更多信息。



语言页面内的设定选项。

语言

单机版可进行显示语言修改，修改完成后需重新启动 HWCONFIG 才会生效；由 ISPSOft 开启的 HWCONFIG 会与 ISPSOft 显示语言相同，若需更改语言请先关闭 HWCONFIG 至 ISPSOft 变更语言后再开启 HWCONFIG。

订购清单

将会列出项目中所配置的硬件及配件列表。

单击「菜单工具栏」中的「工具」的「订购清单」，或使用快捷键 **Ctrl+F6**。



待订购列表窗口出现后，上方会显示目前项目内所配置 PLC 主机、功能卡、以及模块信息及数量，下方会显示配件信息及数量。



配件可以在配件设定中调整配件规格及数量，单击 PLC 主机右方的配件设定。



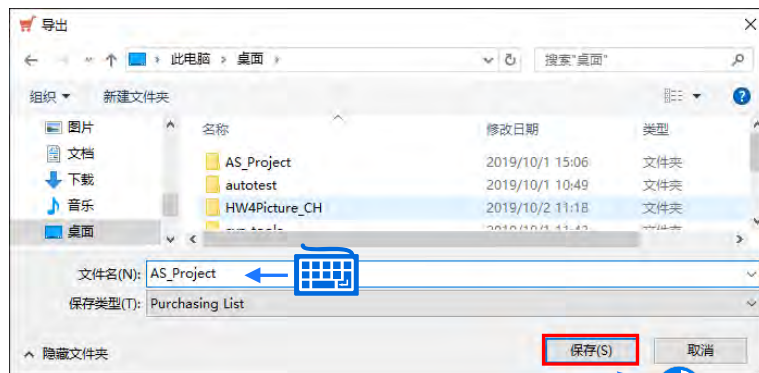
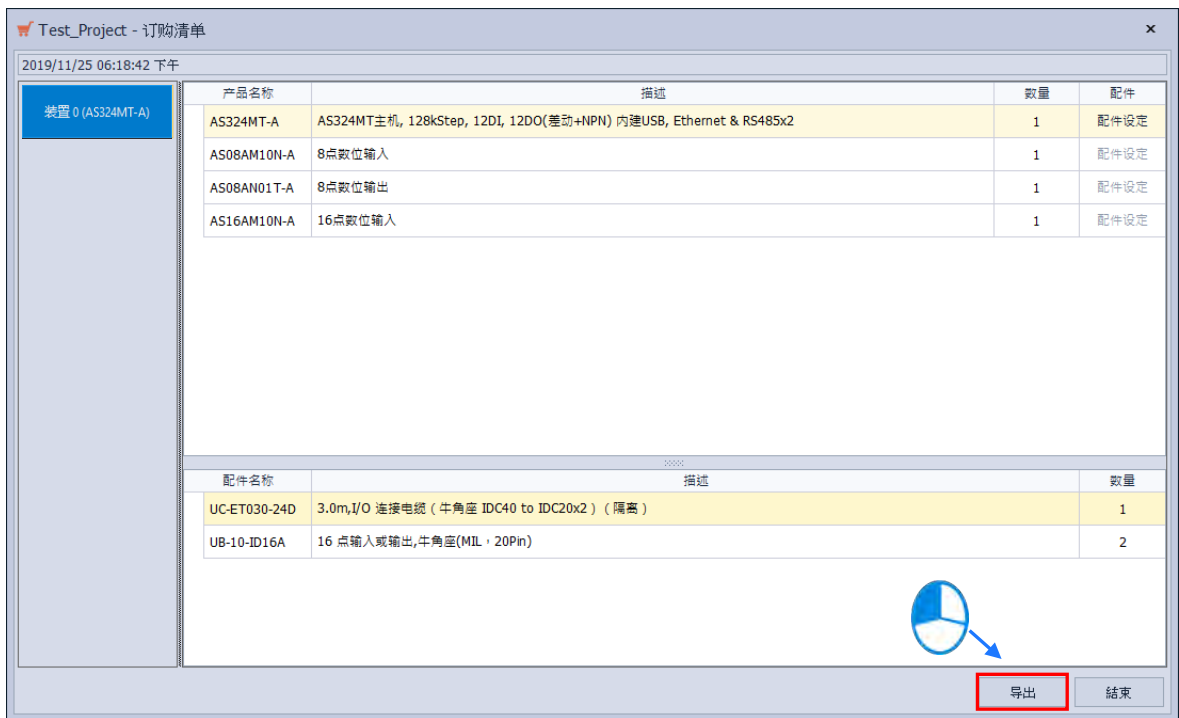
待配件设定窗口出现后，窗口会显示项目需要的转接线及转接板数量，可以更改配件型号及数量，其数量必须等于项目所需的最小数量，单击「默认值」会恢复默认配件及数量，填写完成后单击「确定」即可更改订购列表内配件数据。

配件名称	描述	数量
UC-ET010-24D	1.0m,I/O 连接电缆 (牛角座 IDC40 to IDC20x2) (隔离)	1
UB-10-ID16A	16 点输入或输出,牛角座(MIL, 20Pin)	2

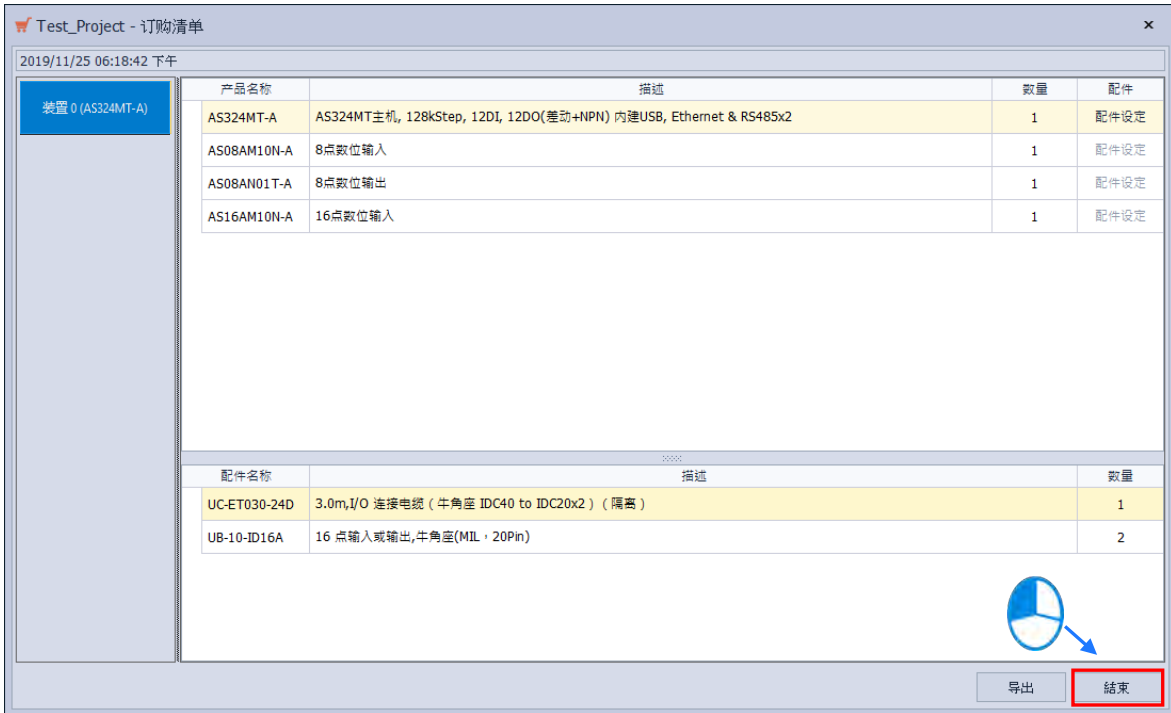
3

配件名称	描述	数量
UC-ET030-24D	3.0m,I/O 连接电缆 (牛角座 IDC40 to IDC20x2) (隔离)	1
UB-10-ID16A	16 点输入或输出,牛角座(MIL, 20Pin)	2

设定好配件设定后，单击「导出」即可将订购清单以 xlsx 文件的格式被导出，而之后便可通过 EXCEL 对导出的数据进行编辑或管理。

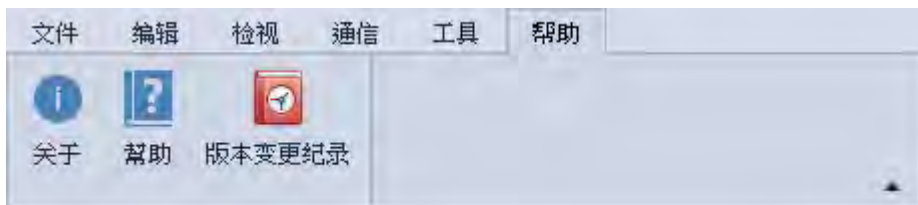


最后可单击「结束」来关闭订购列表窗口。



3.1.2.7 菜单工具列-帮助

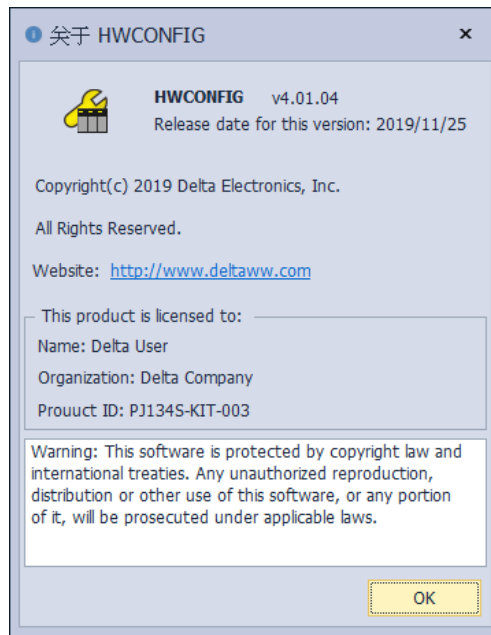
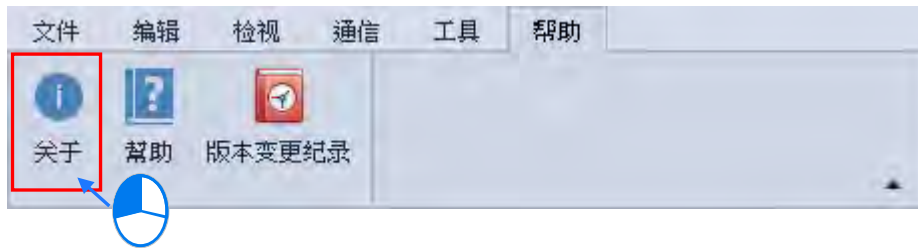
说明：主要为 HWCONFIG 所提供的一些辅助功能。



操作项	功能说明
关于	显示软件版本及发行日期等相关信息。
帮助	显示 HWCONFIG 的使用说明。
版本变更纪录	显示此版本的新增与修改项目。

- 关于

在「说明」内单击「关于」会显示 HWCONFIG 的软件版本、发行日期、以及软件信息。



3

3.1.2.8 菜单工具列-常用

说明：主要为 HWCONFIG 常用选项。

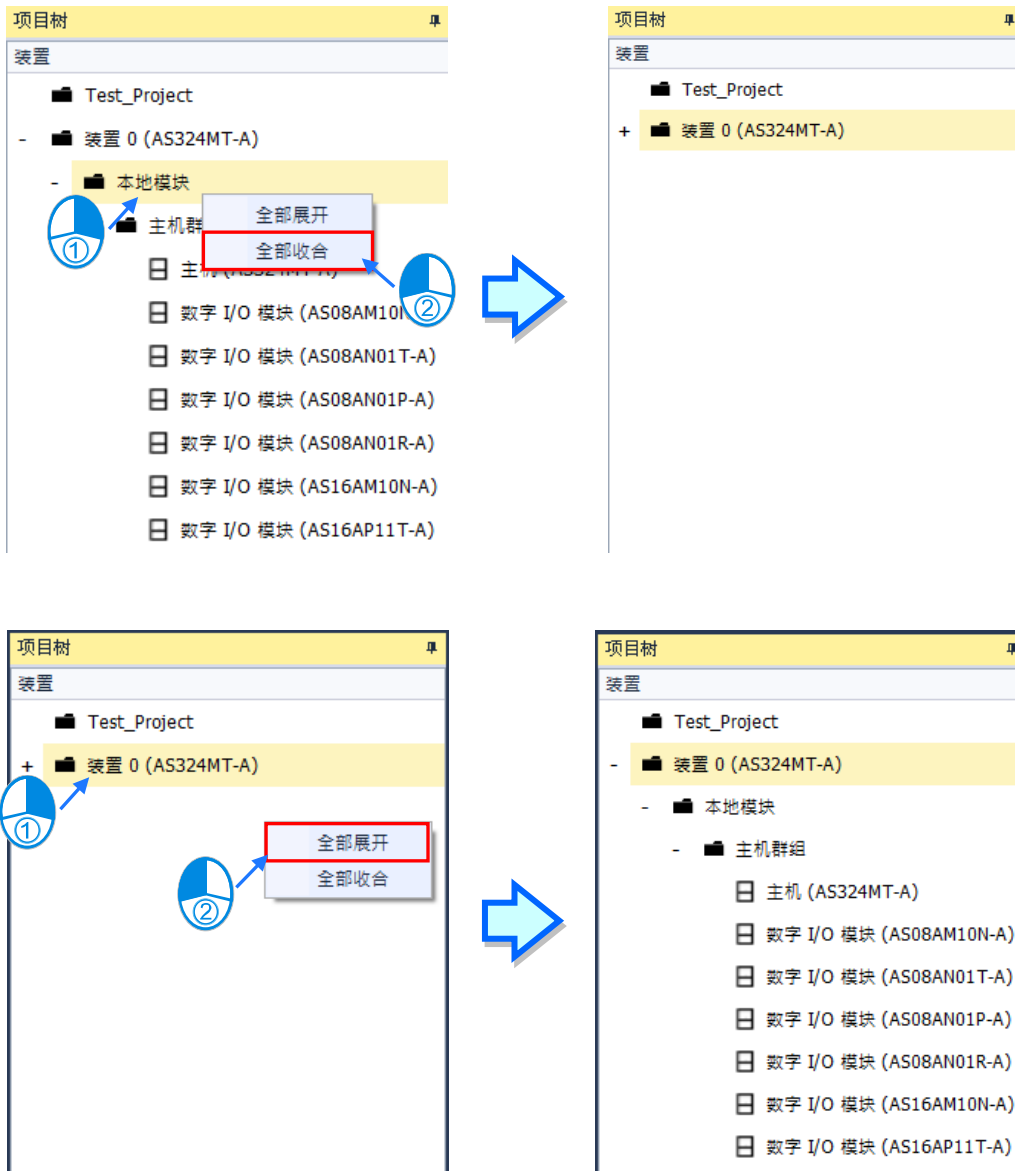


3.1.2.9 项目树

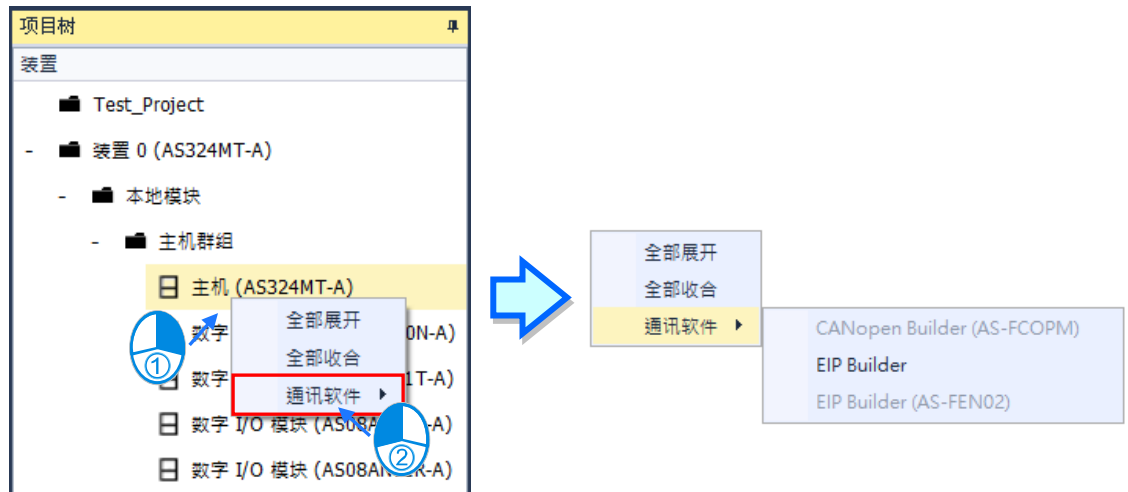
项目树的主要内容为目前项目所配置的装置，并以阶层树状结构做为其管理接口。

项目树窗口默认会在隐藏在 HWCONFIG 窗口的左侧，可以使用第 3.1.3.12 节的功能来固定或隐藏窗口，也可以从「检视」中的项目树选项来更改其显示方式，在项目树中某个对象上右击时，会出现「全部展开」以及「全部收合」2 种选项，「全部展开」为展开所有树形图的装置、「全部收合」为收起所有树形图的装置。

3



在项目树中的主机单击右键，有个通讯软件的项目，字体会呈现白色是指目前主机支持且可以打开的通讯软件，字体会呈现灰色是指目前主机有支持但需配置特定功能卡才可以使用的通讯软件。




3.1.2.10 产品列表

产品列表主要显示项目机种所支持的功能卡以及模块，并以阶层树状结构做为其管理接口，单击任一装置后，会在下方的信息窗口显示设备图片及描述数据。默认的产品列表会在固定在 HWCONFIG 窗口的右侧，可以使用第 3.1.3.12 节的功能来显示或隐藏产品列表。

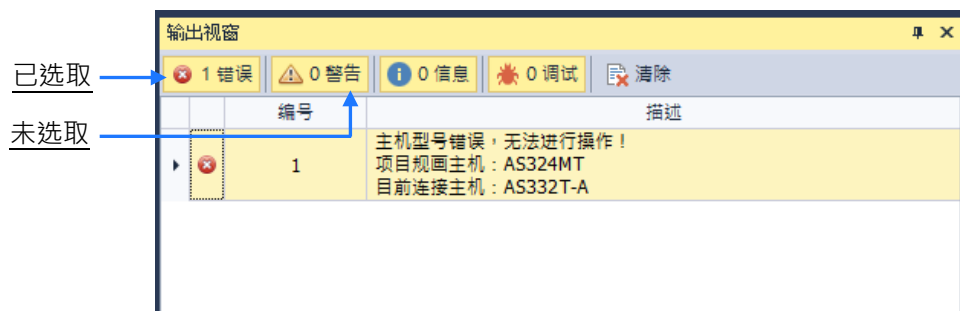
产品列表上方有搜寻功能，输入关键词可快速地找到装置，如下方右侧图。



3.1.2.11 输出视窗

输出视窗主要是显示进行硬件配置或联机模式时的提示信息。默认的项目树会在隐藏在 HWCONFIG 窗口的下方，可以使用第 3.1.3.12 节的功能来固定或隐藏输出视窗，若单击右上角的  图标可将其关闭，而通过菜单工具栏的「检视」中的「输出视窗」则可将其打开或关闭。


输出视窗下方有 5 个选项，选取错误、警告、信息、与调试时会在描述内显示信息，反之则不会显示，若单击清除时则会清除所有信息。



3

3.1.2.12 硬件配置区

硬件配置区主要是设定项目的模块与功能卡配置。

单机版项目中若有多个 PLC 装置，上方的分页标签可以让用户切换到不同 PLC 装置，也可以使用右上角  图示来切换分页；而通过 ISPSOft 打开的 HWCONFIG 项目无法新增 PLC 装置，所以只会出现一个分页。

硬件配置区上方有「清空模块」、「信息」、「重新排列地址」、「重置大小」、「放大」、「缩小」、「百分比」、「以及「备注」。

3

操作项	功能说明
清空模块	清空所有配置的模块，以及主机参数设定。
信息	显示或关闭主机及模块名称等信息。
重新排列地址	重新排列右边画面输入设备范围以及输出装置范围的地址。
重置大小	将模块配置图回到默认大小。
放大	单击以放大模块配置图。
缩小	单击以缩小模块配置图。
百分比	使用百分比来放大缩小模块配置图。
备注	可填入备注信息。

硬件配置区主要有两个区块，左边为项目的模块配置图；右边为项目的模块配置相关信息，包含模块、型号、输入设备范围、以及输出装置范围。



机种操作时，以鼠标单击某个模块图标即可将该模块选取起来。



AH 系列要选取背板时，请先将鼠标移至背板图标左侧的连接端口位置，接着单击鼠标左键即可。

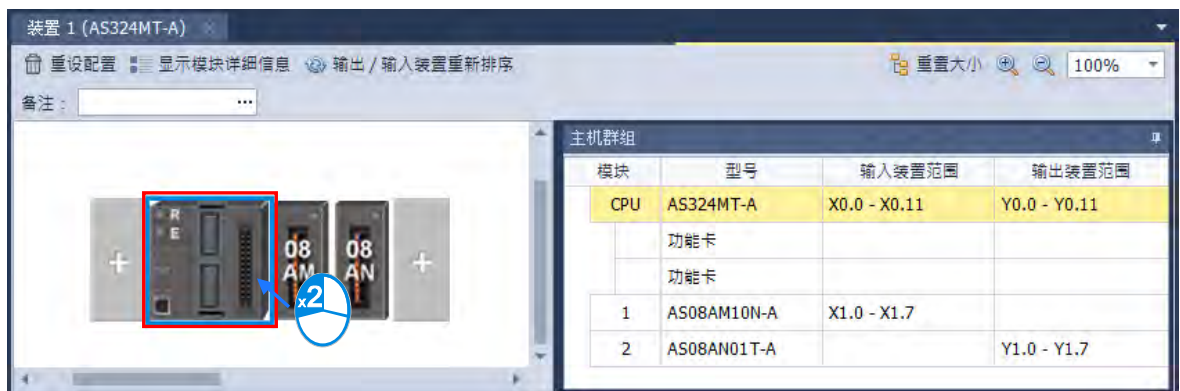




3

3.1.2.13 编辑区

编辑区主要是设定 PLC 与模块参数，PLC 包含系统设定、通讯端口设定、以太网网络设定、以及功能卡设定与模块的参数设定、数据交换的参数设定；模块参数设定请参照各模块操作手册。

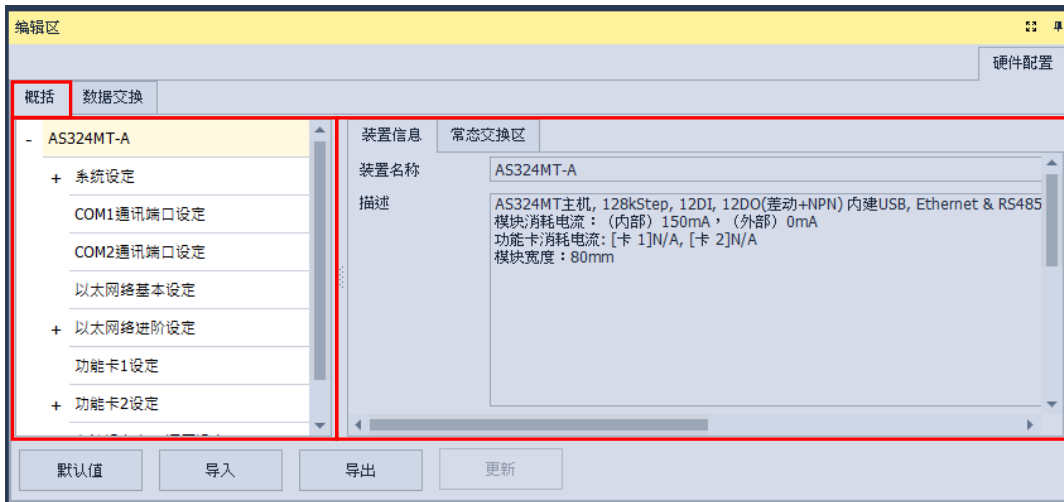
默认的编辑区不会在 HWCONFIG 中显示，必须要双击主机打开编辑区，如下图。



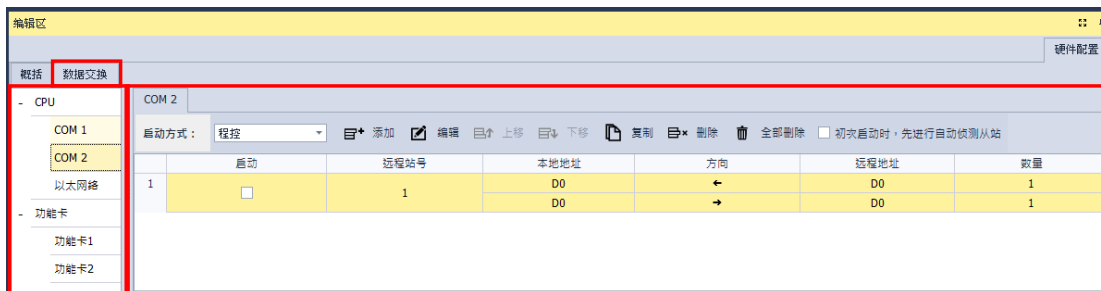
也可以使用参考第 3.1.2.6 节的环境设定将显示模式改变后，可以使用下方第 3.1.3.12 节的功能来显示或隐藏。单击编辑区右上角  图示会将编辑区放大，但会覆盖硬件配置区的窗口，若要显示硬件配置区只要再次单击  图示，或从「检视」选取「重置布局」也可将硬件配置区还原。

编辑区的「概括」页面左侧会显示主机、以及功能卡的参数设定选项；若在硬件配置区的模块配置图上选取任一模块时，则此页面会显示模块的参数设定选项，右侧则是显示设备信息与常态交换区信息。

3



「数据交换」页面可以设定不同通讯方式的数据交换设定，右侧为设定数据交换之内容。



3.1.3 进阶功能操作

本章节将会介绍 HWCONFIG 软件的进阶功能。

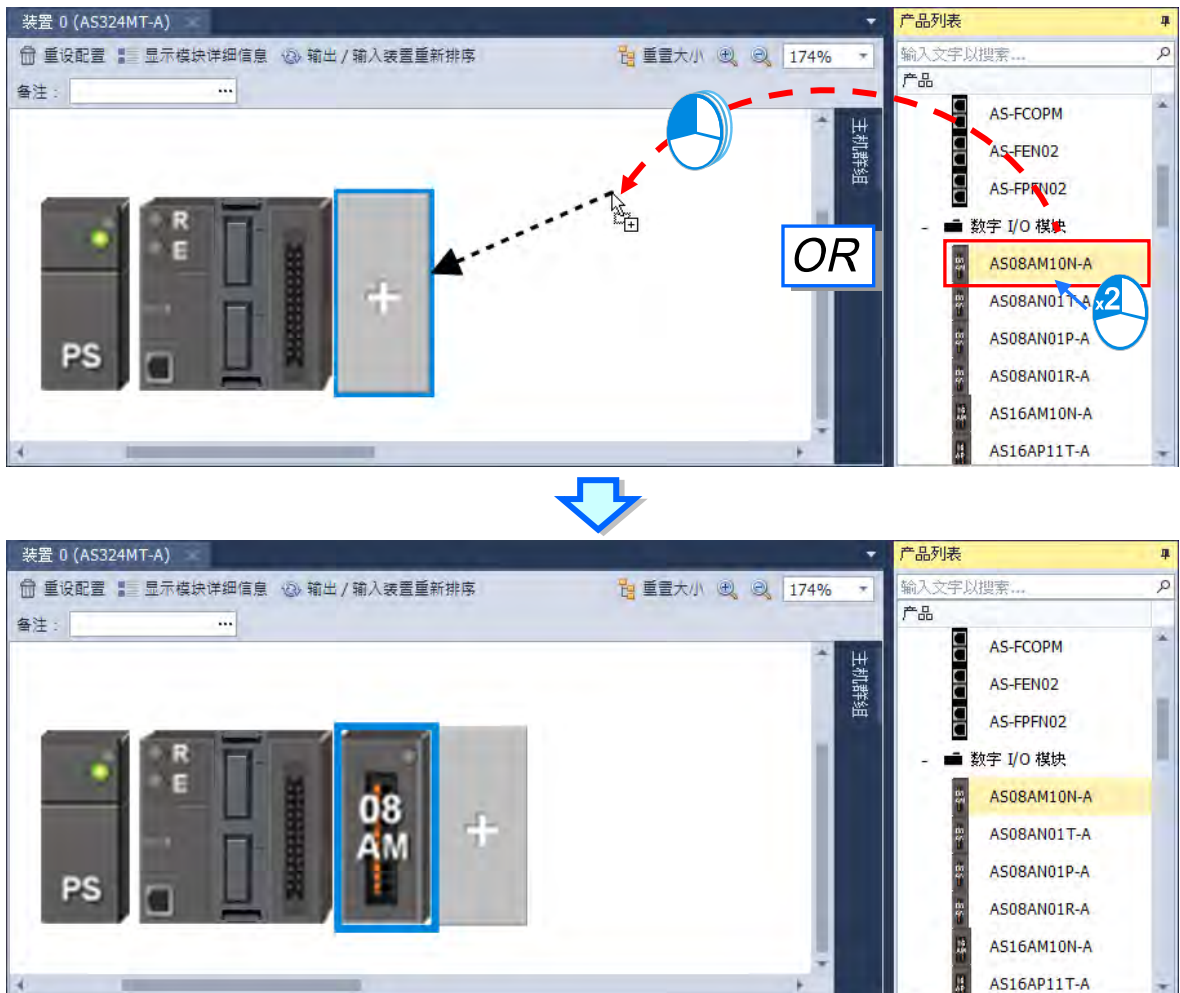
3.1.3.1 新增模块/扩展背板

将模块加入至硬件配置区有多种方式，以下将会介绍所有新增模块的方式。

- 方法一

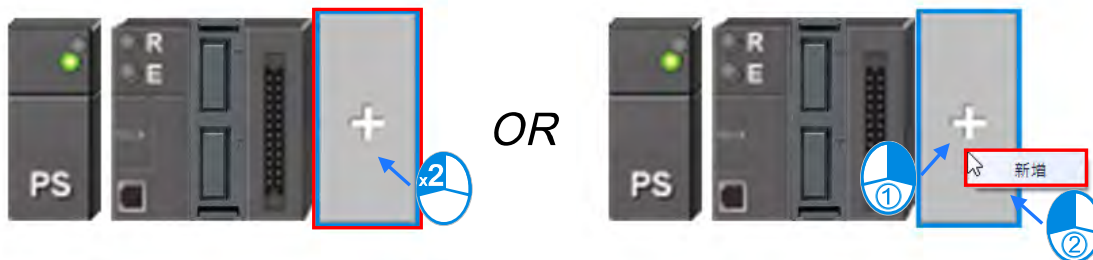
在右侧产品列表中选取要加入的模块后，双击鼠标左键，或将其拖曳至硬件配置区的空白处，会有虚线显示拖曳要放置的位置，接着再放开鼠标左键即可加入。

3



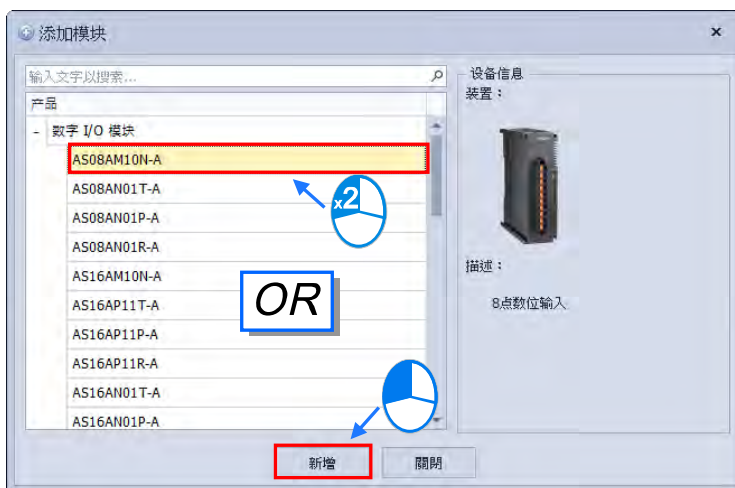
● 方法二

在模块配置图的灰色加号图标双击鼠标左键或是单击右键后选择「新增」。



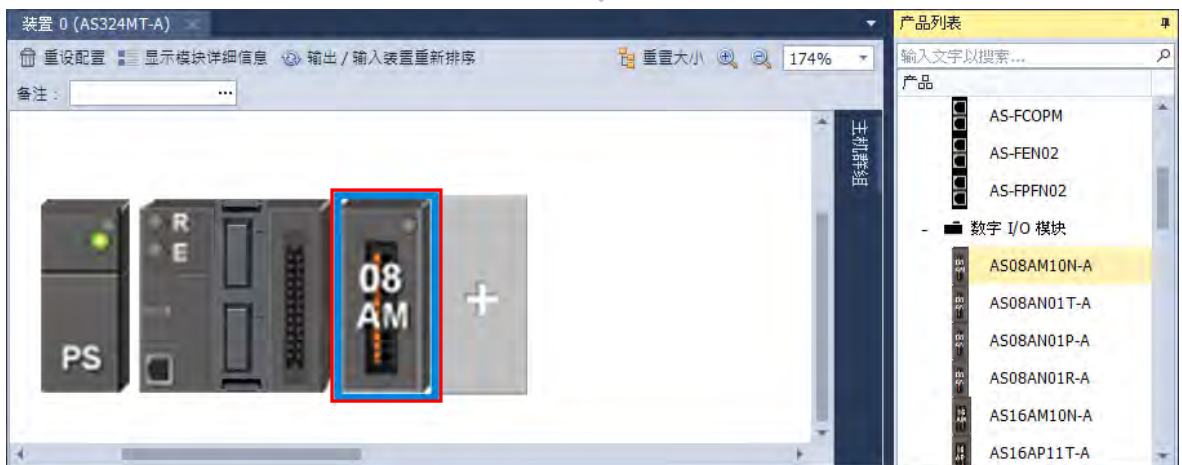
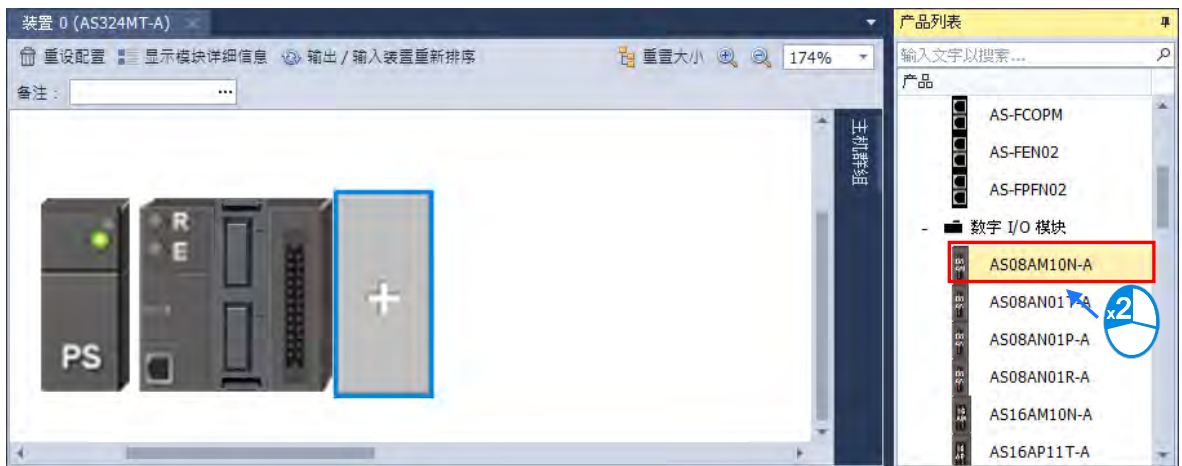
3

在要加入的模块上双击鼠标左键或单击「新增」就可以新增模块，可连续增加不同的模块，模块会依序向右侧增加，完成操作后单击「关闭」即可关闭此窗口。



● 方法三

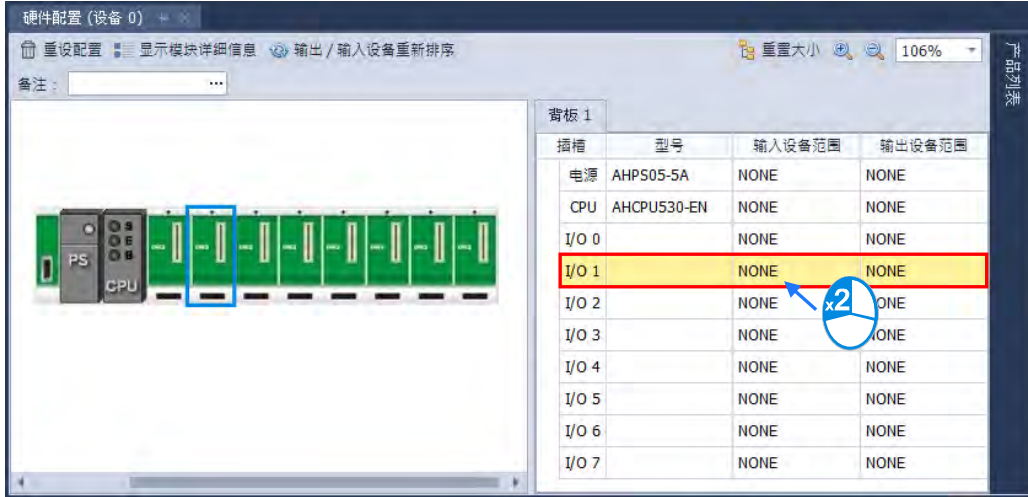
在产品列表中对要加入的装置双击鼠标左键或输入 Enter 键，都会将装置加入至硬件配置区。



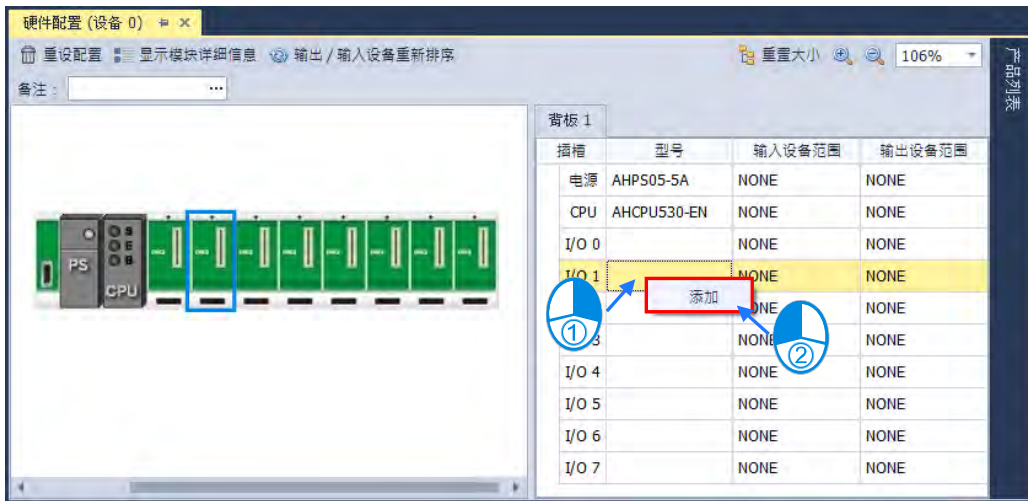
● 方法四

具有背板的 AH 模块，硬件配置区右侧信息字段中，在欲加入的插槽位置双击鼠标左键或是单击右键后选择「新增」，即会跳出新增模块窗口。

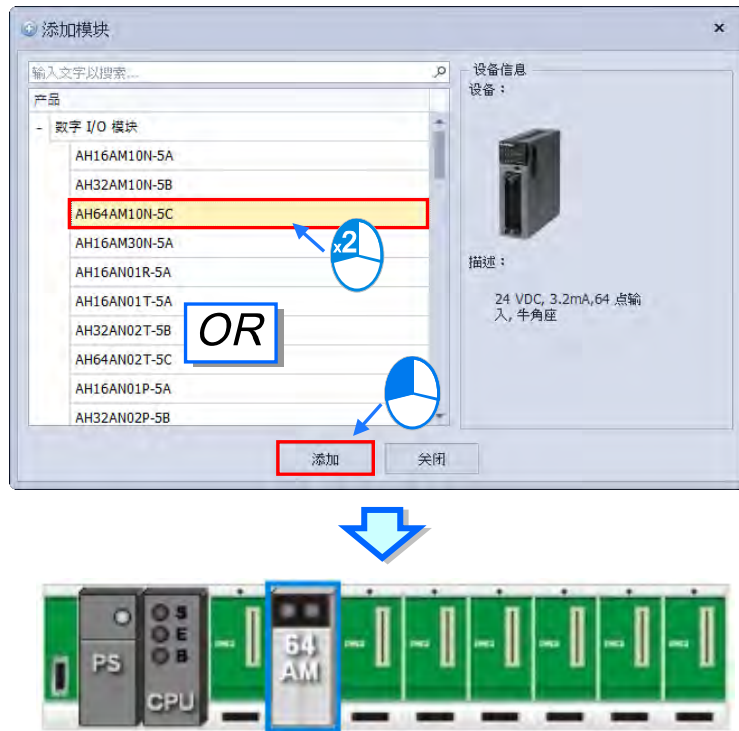
3



OR

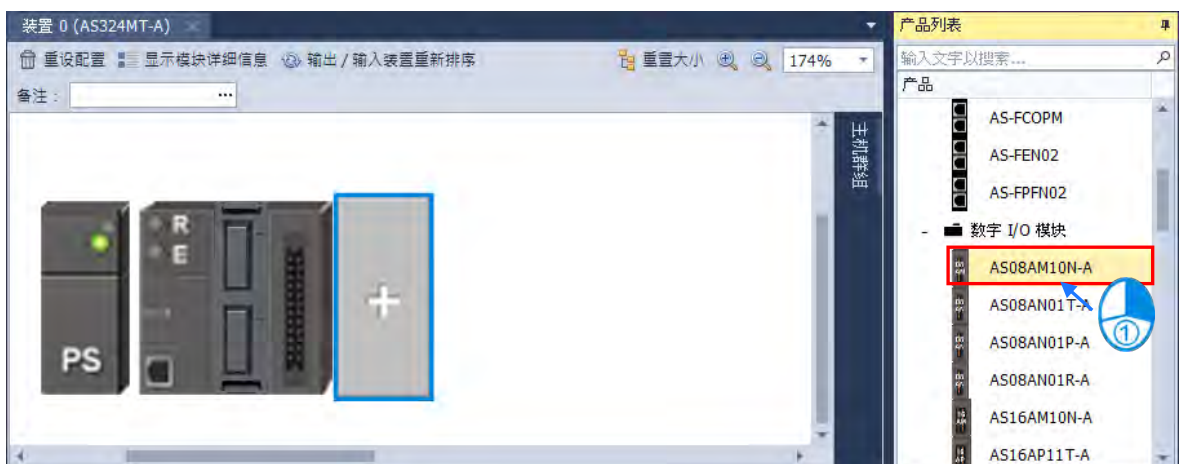


在要加入的模块上双击鼠标左键或单击「新增」就可以新增模块。

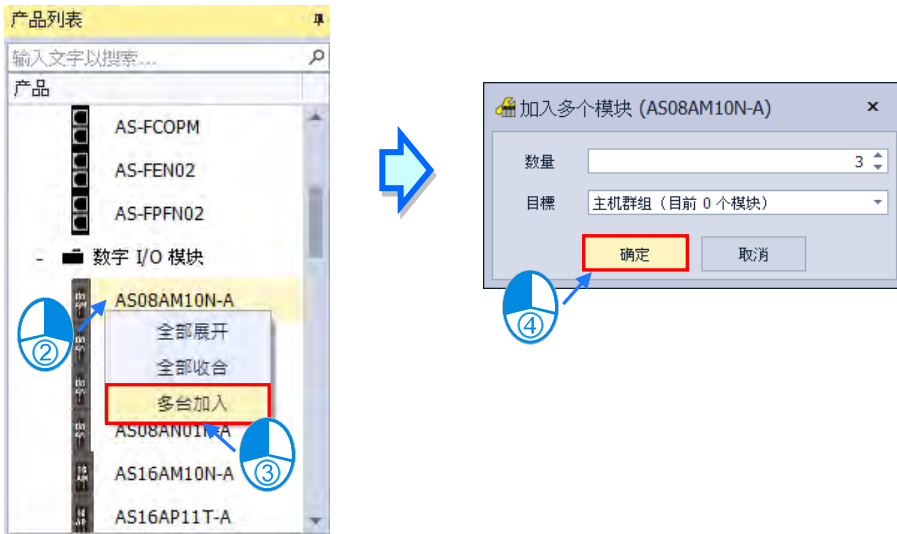


- 加入多台模块

右击时，画面便会出现「全部展开」、「全部收合」、以及「多台加入」等 3 种选项，「全部展开」为展开所有树形图的对象；「全部收合」为收起所有树形图的对象；「多台加入」为一次增加多台选定的模块至项目中，选择多台加入后会出现加入多个模块窗口，可以在「数量」中填入想要加入的模块数量，单击「确定」后即将模块加入。



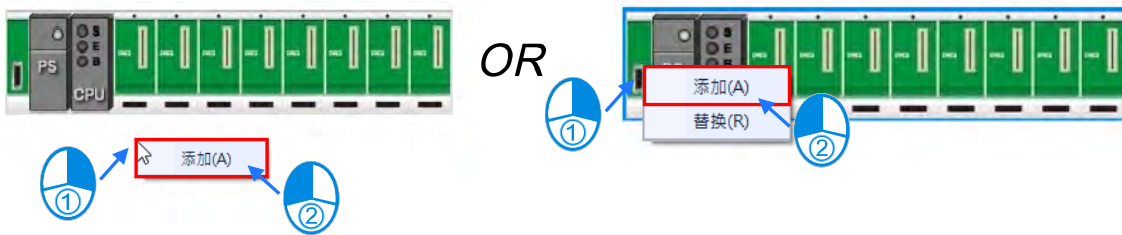
3



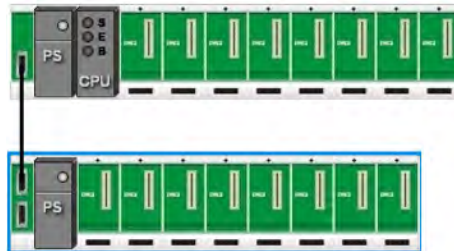
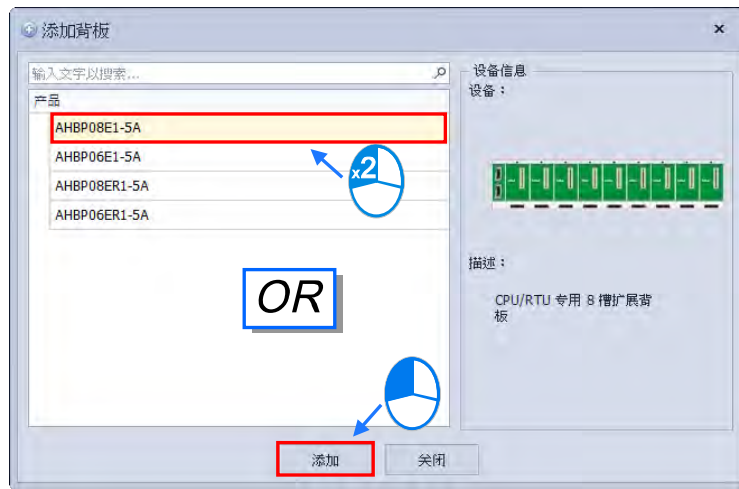
AH 系列机种具有背板，将扩展背板加入至硬件配置区有多种方式，以下将会介绍所有新增扩展背板的方式。

- 方法一

在模块配置图的空白区域或是在背板最左侧单击右键后选择「新增」。



在要加入的背板上双击鼠标左键或单击「新增」就可以新增扩展背板，可连续增加不同的背板，背板会依序向下方增加，完成操作后单击「关闭」即可关闭此窗口。

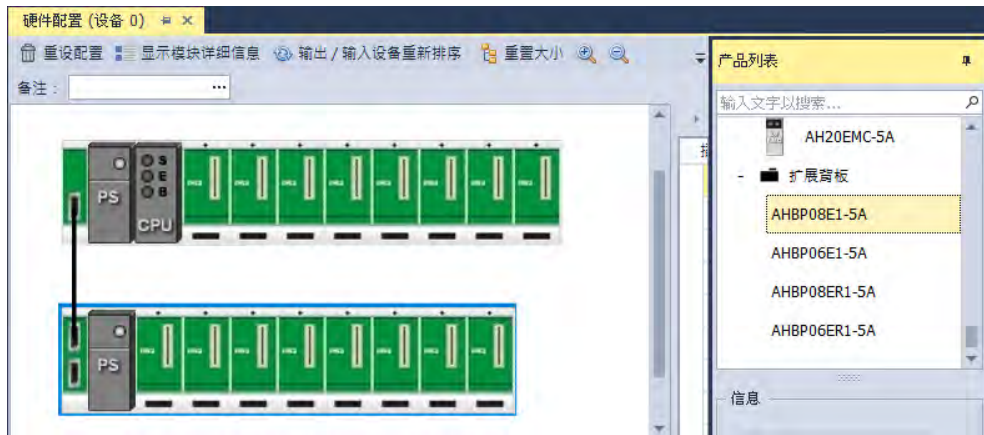
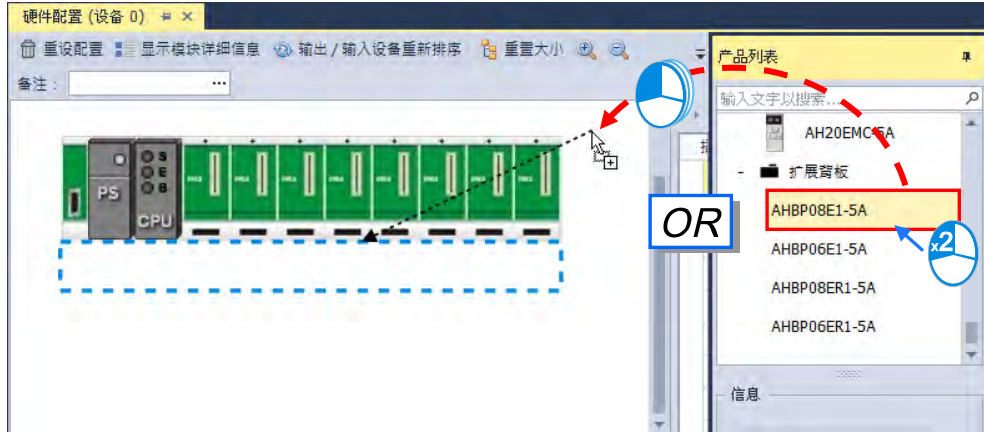


3

● 方法二

在右侧**产品列表**中选取要加入的背板后，双击鼠标左键，或直接将其拖曳至硬件配置区的空白处，会有虚线显示拖曳要放置的位置，接着再放开鼠标左键即可加入。

3

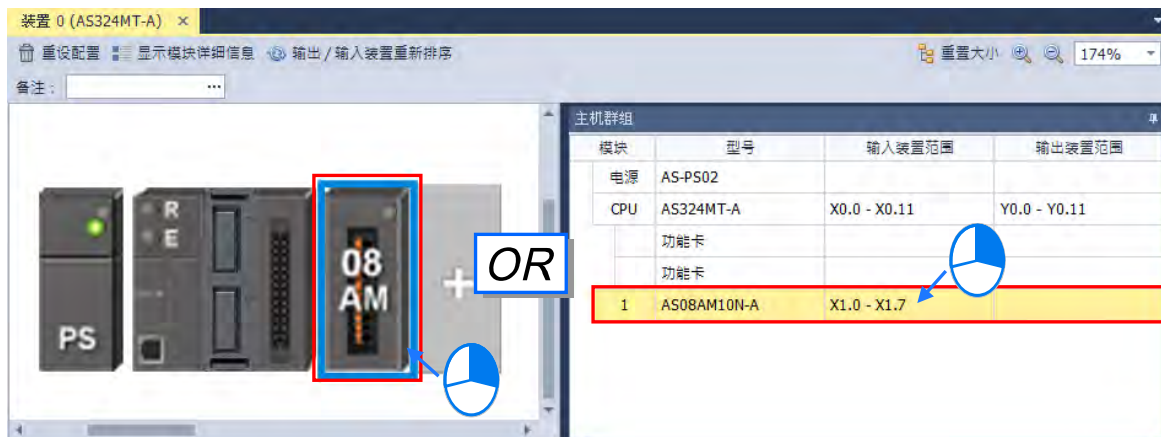


*冗余及一般的扩展备板无法同时存在，软件会自动保留最后选择的种类。

3.1.3.2 更换模块/背板

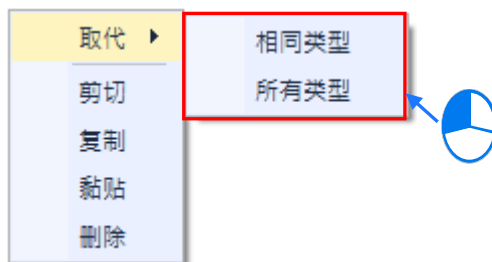
若是想要对硬件配置区的模块进行替换变更，可以依循以下步骤：

- (1) 在硬件配置区的模块配置区或模块配置信息上单击要变更的模块后右击。



3

- (2) 在右键快捷选项中单击「替换」，之后再选择替换的方式。关于两种替换的方式说明如下。



➤ 相同类型

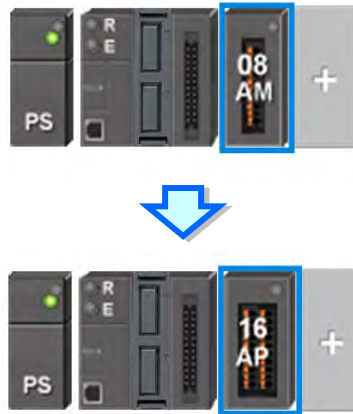
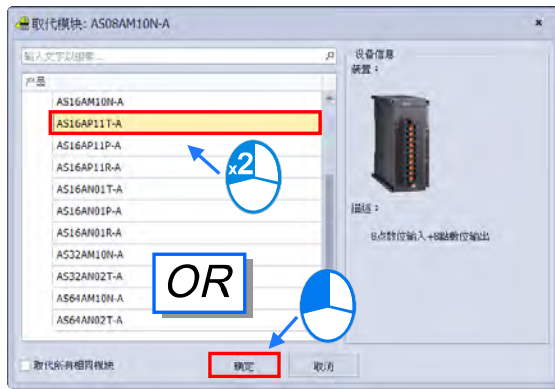
仅可以替换为所选取模块的相同类型之其它模块，而替换之后的模块参数设定若无法完全对应时将会被恢复为系统默认值。

➤ 所有类型

可将所选模块替换为所有类型的模块，而执行之后，其效果等同将原本的模块删除后，再重新插入另一模块，因此新模块的输入/输出装置范围将会重新配置，而模块的其它参数设定也将会重新恢复为系统默认值。

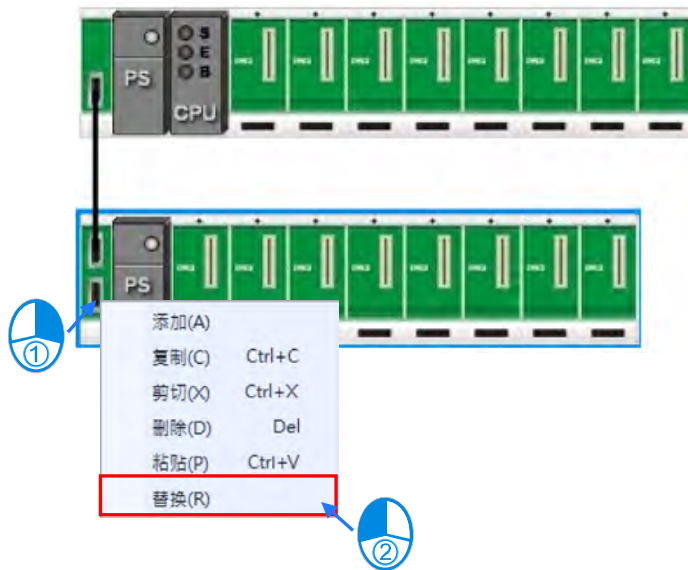
- (3) 单击要执行的替换方式后，画面便会出现模块选择窗口，窗口标题会显示被替换的模块，而窗口中可选择的模块项目，会依据所选的替换方式而有所不同；决定要替换的模块之后，在该模块项目上双击鼠标左键或单击「确定」键即可替换模块。

3

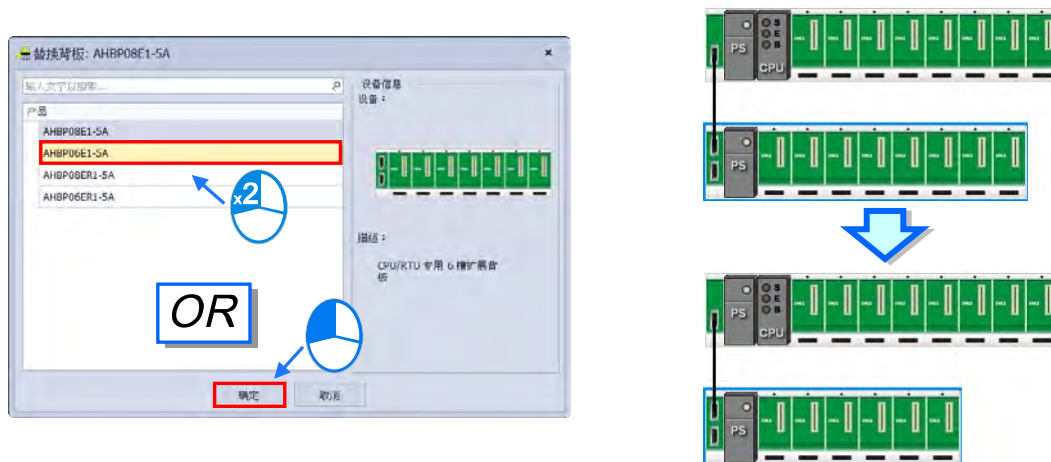


若是想要对硬件配置区的背板进行替换变更，可以依循以下步骤：

- (1) 在硬件配置区的模块配置图中单击要复制的背板左侧后右击，之后在快捷选中单击「替换」。



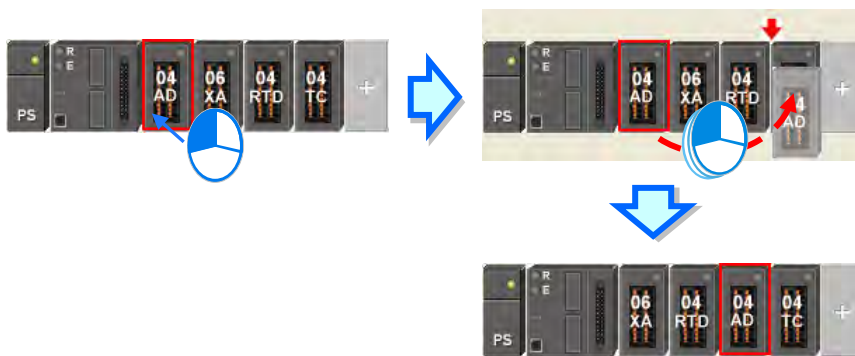
- (2) 单击替换后，画面便会出现背板选择窗口，窗口标题会显示被替换的背板，而窗口中可选择的背板，决定要替换的背板之后，在该背板项目上双击鼠标左键或单击「确定」键即可替换背板。



3.1.3.3 模块/扩展背板的移动

除了 CPU 之外，配置在**硬件配置区**的模块图标皆可直接使用鼠标拖曳来进行移动。

AS 系列机种为无背板设计，故拖曳某一模块至其它模块的位置上时，将插入到目标模块之前的位置，并且会在插入的位置出现一个红色的箭头标示。

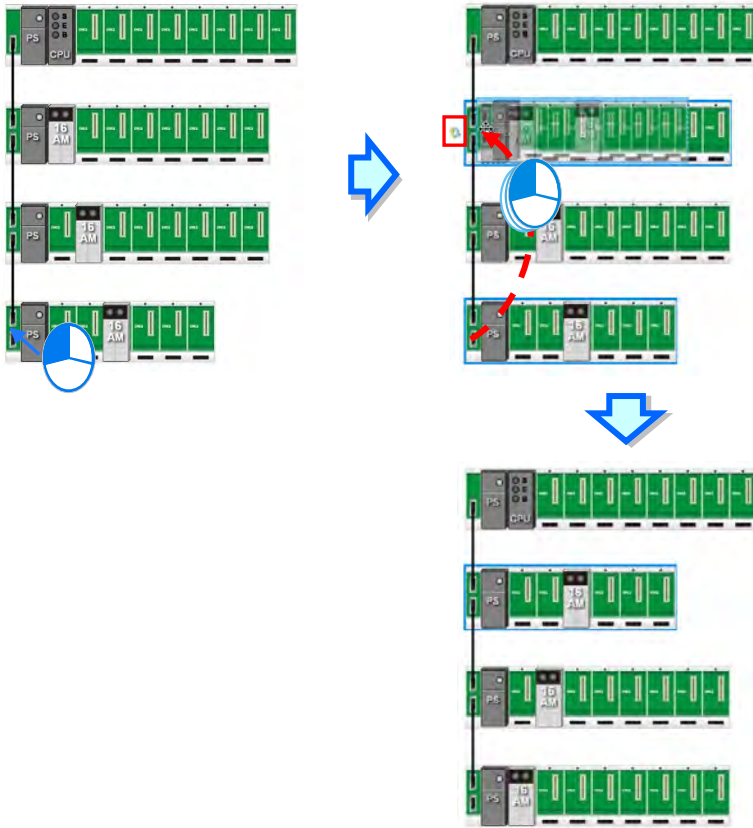


* 移动时，模块备注、模块内部参数、D 对应装置与智能型模块的进阶参数亦会跟随模块移动，且输入/输出装置范围也会依据模块摆放位置而自动重新排。

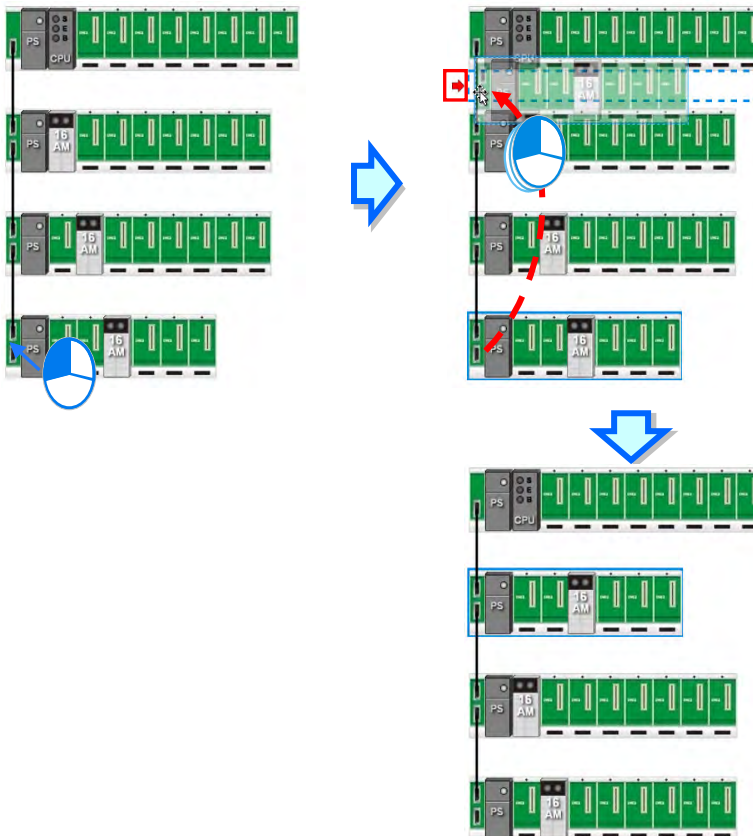
AH 系列机种具有背板，除了背板 1(主背板)之外，配置在**硬件配置区**的背板图标皆可直接使用鼠标拖曳来进行搬移。拖曳某一背板至其它背板的位置上时，将会出现交换的图示，即为两背板互相交换；拖曳某一背板至两个背板的中间时，将会出现插入的图示，即为将所拖曳的背板插入两背板间。

* 背板交换示意图

3



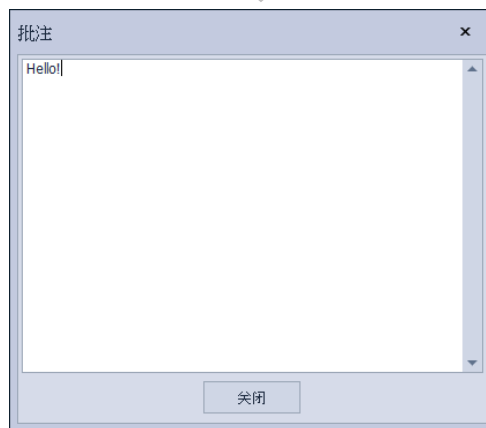
* 插入背板示意图



3.1.3.4 备注编辑

- 项目备注

鼠标单击**硬件配置区**上方的备注栏位后，即会跳出注释窗口，便可输入该项目的备注，而输入完毕后单击键盘【Enter】键或单击「关闭」选项即可进行保存。

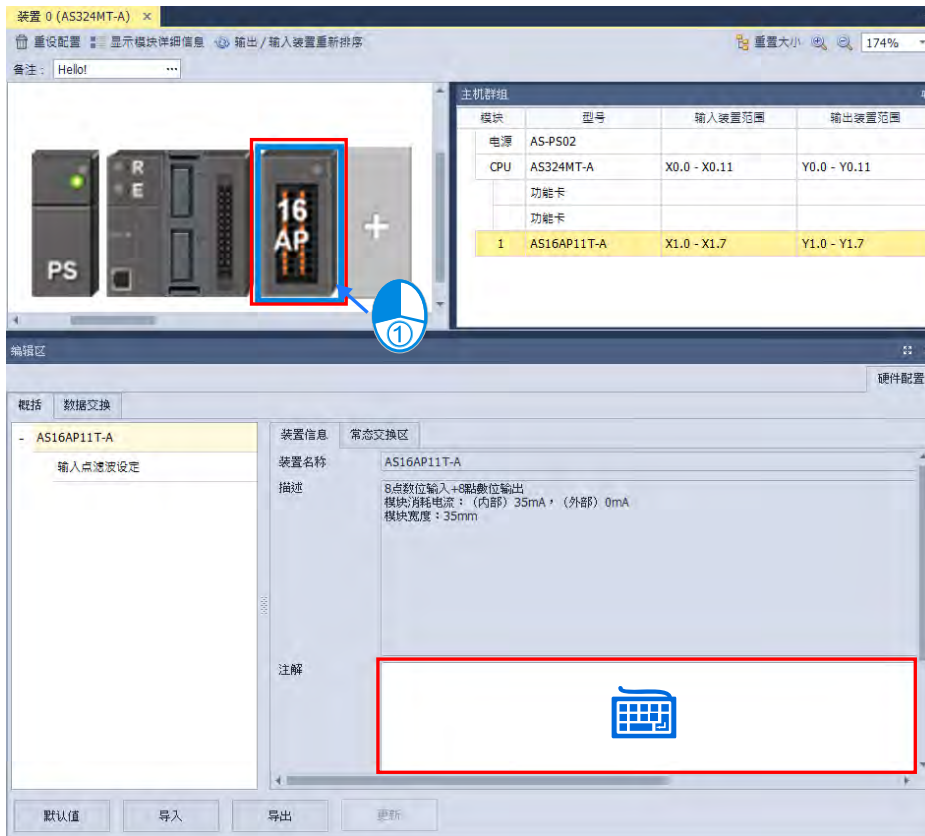


3

● PLC 主机与模块备注

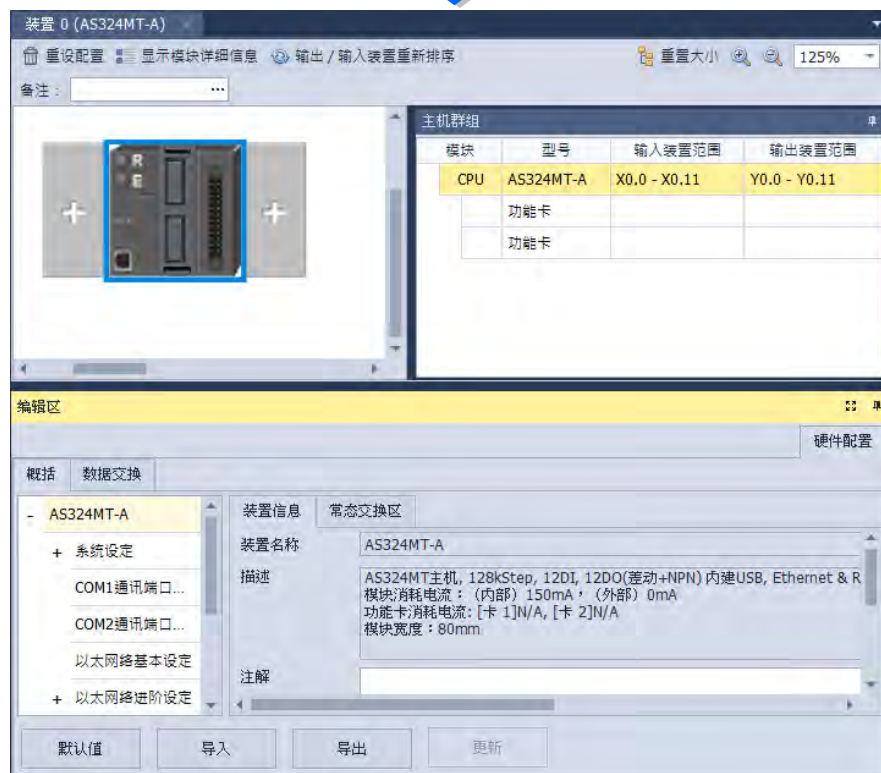
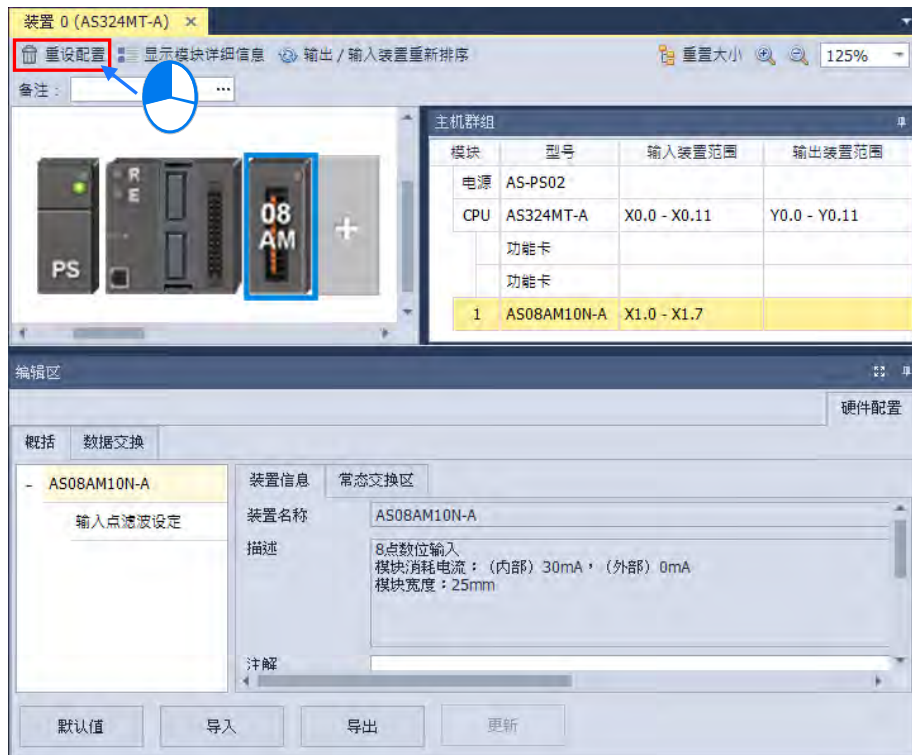
鼠标单击**硬件配置区**中的 PLC 主机或模块后，下方**编辑区**会出现其信息，而在装置信息内有备注字段，单击鼠标左键即可进行编辑。

3



3.1.3.5 硬件配置区-重设配置

使用鼠标单击**重设配置**，可清除所有模块以及 PLC 主机内部设定。



3.1.3.6 硬件配置区-显示模块详细信息

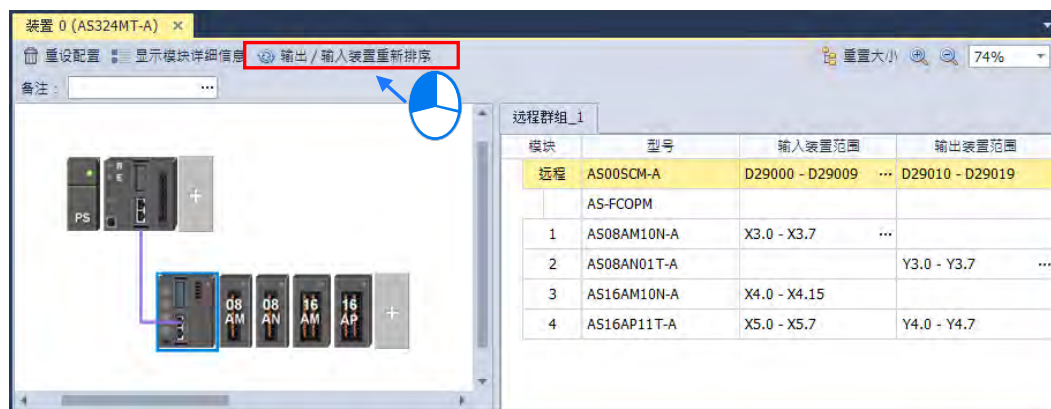
使用鼠标单击硬件配置区上方的显示模块详细信息后，在硬件配置区将会显示 PLC 及模块名称，再次单击就会隐藏显示信息。

3



3.1.3.7 硬件配置区-输出/输入装置重新排序

在进行模块配置的过程中，可能会经过许多的搬移或是复制/剪切/粘贴的操作，而此时各模块的输入/输出装置范围也可能已经被系统自动重新配置，而不是从可用的装置范围最前端分配，此时便可单击 HWCONFIG 硬件配置区上方的「输出/输入装置重新排序」选项，之后所有模块的输入/输出装置范围便会由可用的装置范围最前端的装置来分配。



模块	型号	输入装置范围	输出装置范围
远程	AS00SCM-A	D29000 - D29009 ...	D29010 - D29019
	AS-FCOPM		
1	AS08AM10N-A	X1.0 - X1.7	...
2	AS08AN01T-A		Y1.0 - Y1.7 ...
3	AS16AM10N-A	X2.0 - X2.15	
4	AS16AP11T-A	X3.0 - X3.7	Y2.0 - Y2.7

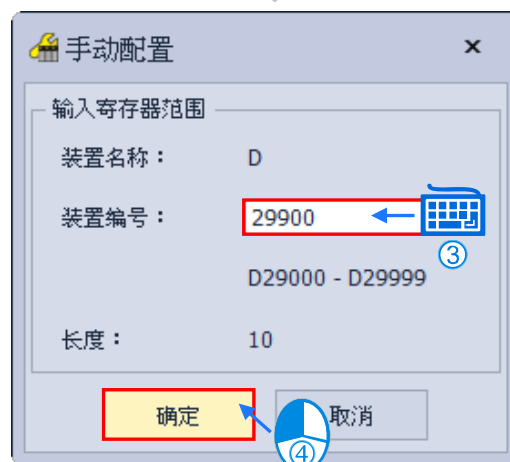
3.1.3.8 硬件配置区-模块的地址配置

模块有经常性的数据更新需求时，例如模拟输入模块会持续更新所接收到的模拟信号，并将其转换为主机可处理的数据，因此系统便会自动为其配置对应的装置地址用以存取数据，而配置的装置地址亦会显示在信息列表中的「输入设备范围」与「输出装置范围」两个字段。请参考下图范例。AS 系列機種主机群组不支持手动设定地址配置，依据模块的安装位置自动配置固定地址，而 RTU 远程群组仅能设置第一台 SCM 模块的起始装置地址，依序配置后续的模块，AS 系列機種远程模块配置请参考第 3.1.3.9 节。

3.1.3.9 硬件配置区-更改远程装置地址

在进行模块配置的过程中，用户可能希望能自行定义远程装置地址，而此时远程群组的「模块配置区」的输入/输出装置范围右方会出现 **...** 的图标，鼠标单击后即会出现地址设定窗口，输入设备编号后单击「确定」按钮，便会更新此远程模块的输入与输出装置范围，并且依序自动配置安装的模块输入与输出装置范围。

3



模块	型号	输入装置范围	输出装置范围
远程	AS00SCM-A	D29900 - D29909 ...	D29910 - D29919

3.1.3.10 硬件配置区-重置大小

在进行模块配置的过程中，可能会经过许多的搬移或是复制/剪切/粘贴的操作，硬件配置区的画面可能在任何位置，图像大小比例过大或过小，此时可以使用鼠标单击「重置大小」后，硬件配置区将会把 PLC 主机至于页面中，并将硬件配置大小设置成 100%。



3

3.1.3.11 编辑区-模块内部参数的导出与导入

所有模块内部的参数值皆可导出为 dep 文件另行备份，同时亦可将先前所导出的文件重新导入。要导出时，单击参数设定窗口的「导出」按钮，接着于文件窗口中指定要导出的路径与文件名后保存即可。要导入时，单击参数设定窗口的「导入」按钮，接着在文件窗口中指定要导入的文件后打开即可。

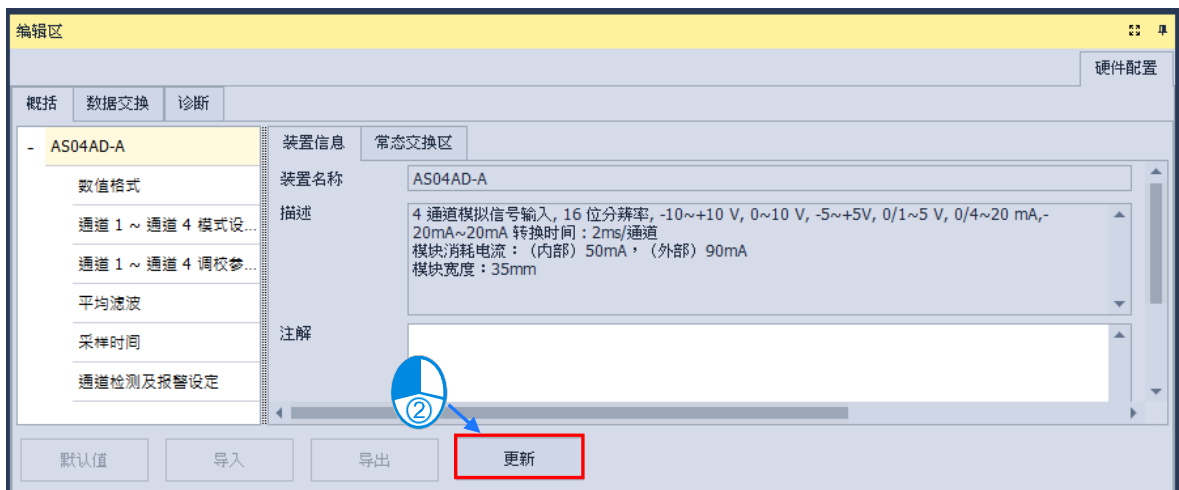
3







导入模块参数时系统会先检查文件格式及模块型号，当文件格式或模块型号不符时，系统将会禁止导入的操作，并显示警示信息。

3.1.3.12 编辑区-模块内部参数的更新

若要做模块内部参数的更新，首先，需进入在线模式，若要进入在线模式请参考第 3.1.2.5 节，进入在线模式后，单击想要更新参数的模块，接着单击编辑区下方的更新，此时模块内部参数即完成更新，此功能仅限于数字输入 I/O、模拟 I/O 及网络模块。

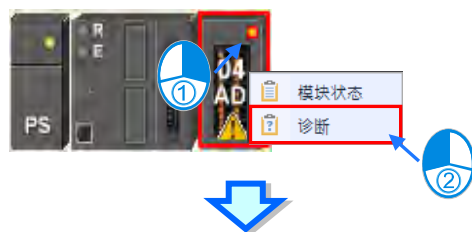




3.1.3.13 显示/隐藏功能

单击 HWCONFIG 的菜单工具栏  或  的图标，可以将工具栏功能隐藏。当工具栏功能隐藏后，将鼠标移动至滚动条时，窗口会自动展开，只要鼠标离开该区域一段时间之后，该区域的窗口便会自动隐藏，若要切换窗口的显示方式，单击  图标，当图标切换成  状态时窗口会固定在 HWCONFIG 窗口中。

3.1.3.14 信息检视与模块诊断

- (1) 在在线模式单击 CPU 或模块图标后右击，并在选项中单击「诊断」项目，下方编辑区将会出现诊断分页，显示该模块的错误记录。



- (2) 在诊断的分页中，「当前错误记录」字段会显示目前的错误状态。
- (3) 当错误状况已排除时，单击窗口中的清除图标  便可清除画面上与模块中保存的错误记录，并可解除模块的错误状态；单击更新图示  时，系统便会重新获取数据并将其更新在窗口中。

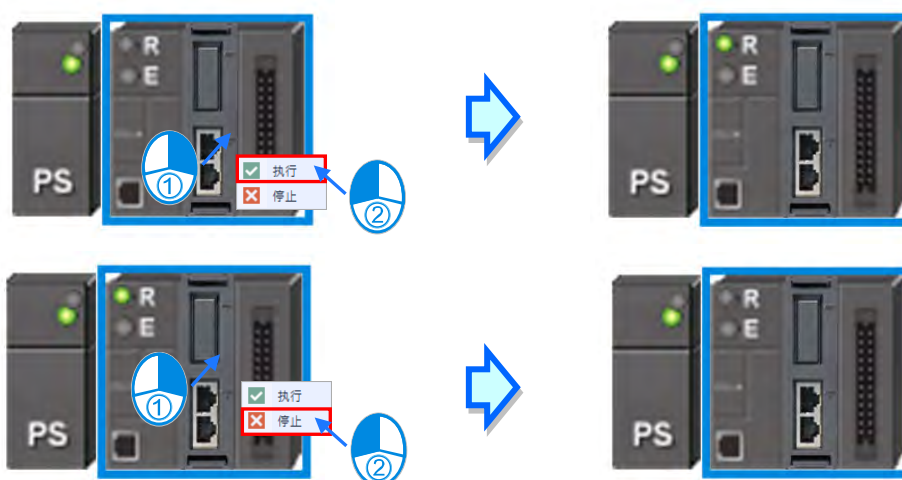
3.1.3.15 在线变更模块状态

HWCONFIG 支持于在线模式时变更装置本身的状态，用户可藉此进行简单的测试。

! 变更装置状态之前，请务必确认所做的操作不会造成系统或人员的伤害。

- 变更装置的运转状态

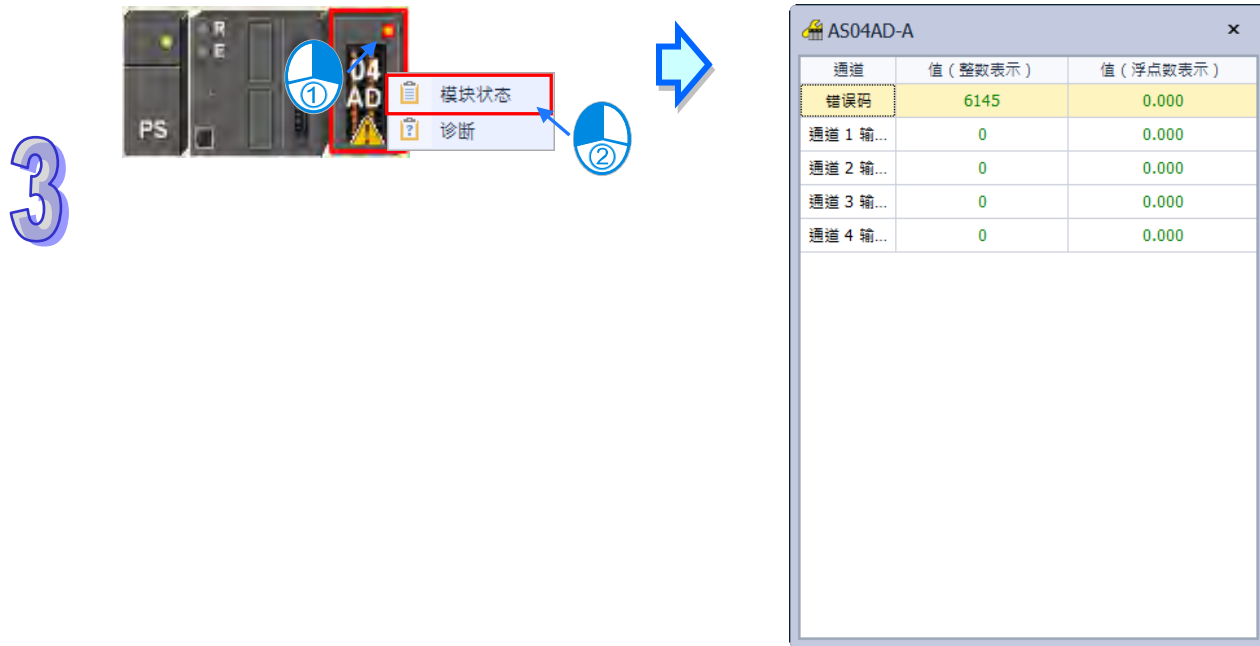
在在线模式时单击 CPU 主机图标后右击，接着在快捷选项中单击「执行」项目，之后主机与模块便开始执行，其图示亦会出现绿色的运转显示灯号；若单击「停止」项目，则此主机与模块便会恢复至停止状态。



3

● 变更模块的输入/输出状态

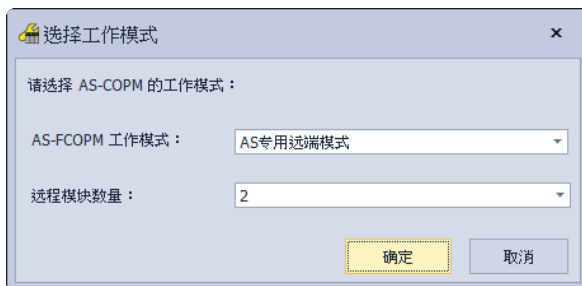
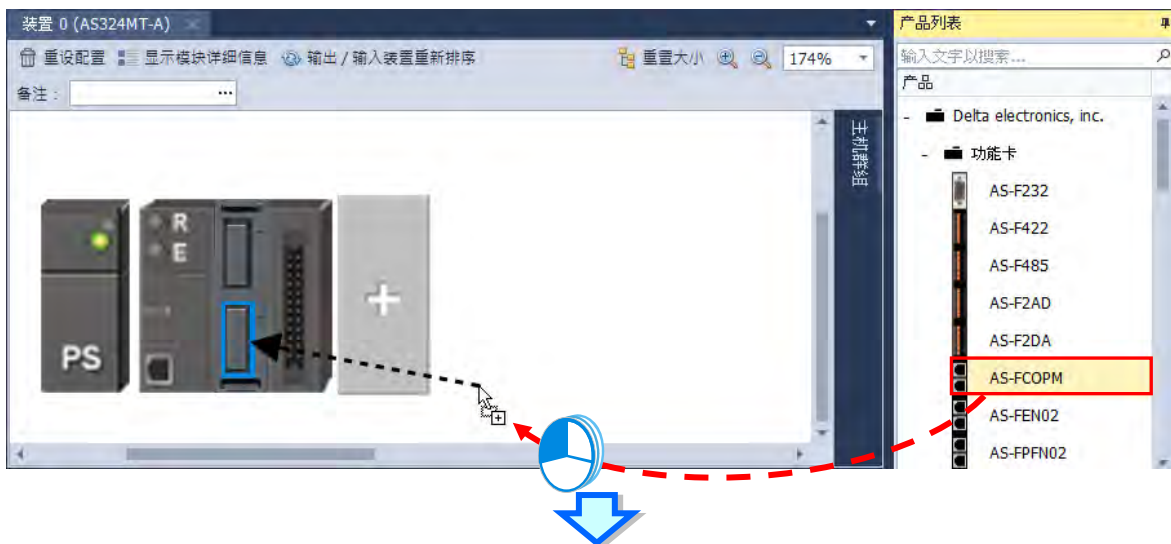
当硬件配置中包含**数字 IO**、**模拟 IO** 或**温度模块**时，在这些模块的右键快捷选项中单击「**模块状态**」项目，接着便可在状态窗口中监控这些 IO 模块的输入或输出状态，其中**数字 IO 模块**还可通过右键选项来对输入或输出通道进行强制 **ON/OFF** 的操作。不过当执行此功能时，CPU 与该模块则必须处于执行（**RUN**）的状态才会实际产生输出。



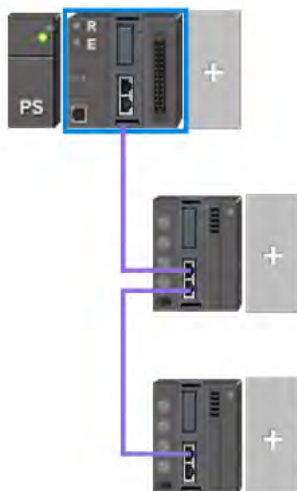
- **强制 ON**：将单击的通道状态强制设定为 **ON**。
- **强制 OFF**：将单击的通道状态强制设定为 **OFF**。
- **解除**：解除单击通道的强制状态。
- **全部解除**：解除全部通道的强制状态。

3.1.3.16 远程模块配置

在产品列表列表中，将 **AS-FCOPM** 模块拖曳至主机的功能卡 2 插槽，便可选择 **AS-FCOPM 工作模式** 将其设定为 AS 专用远程模式，再在 **远程模块数量** 输入连接的数量，如下图。

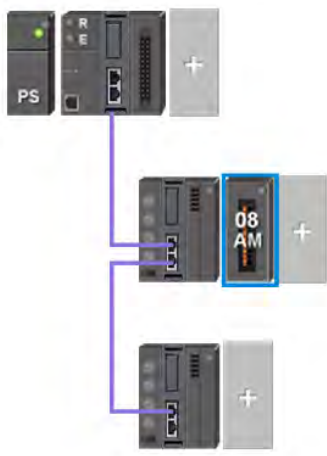


若**远程模块数量**设定为 2 台，单击「**确定**」按钮后，便会在系统配置区出现新增的 2 台远程模块。

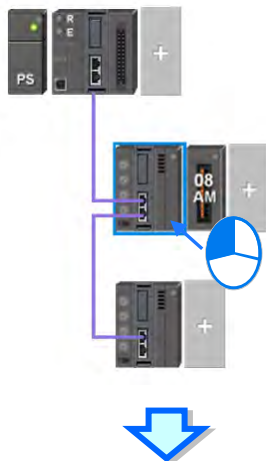


远程模块只能配置数字 I/O 模块与模拟 I/O 模块，其余模块皆无法配置。

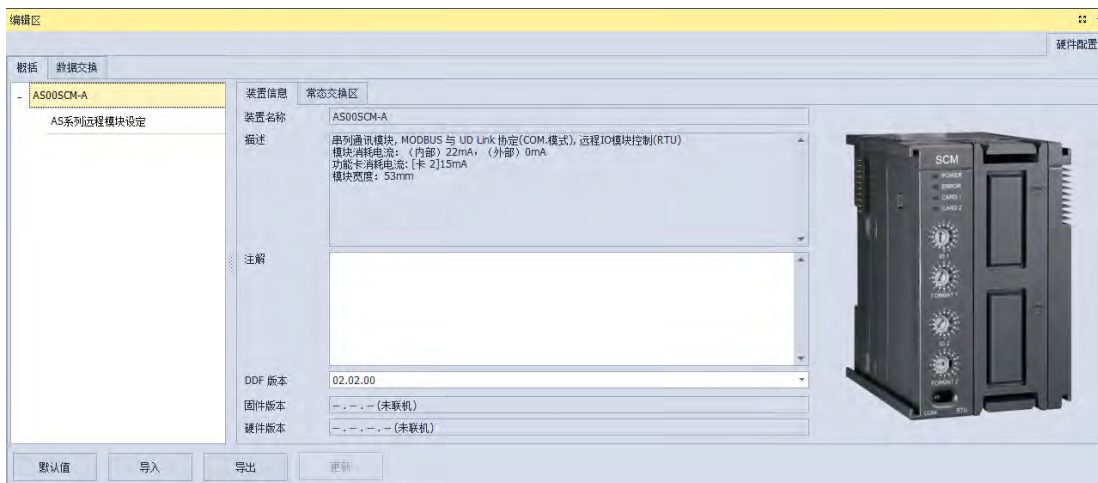
3



同样的，可以用鼠标单击远程模块及其扩展模块，在下方编辑区页面进行设定。



3



3.1.3.17 打开外部软件

在硬件配置区的主机上右击，接着单击「通讯软件」，就会出现可以打开的外部软件，部分软件需要搭配相关的功能卡，才可被使用。

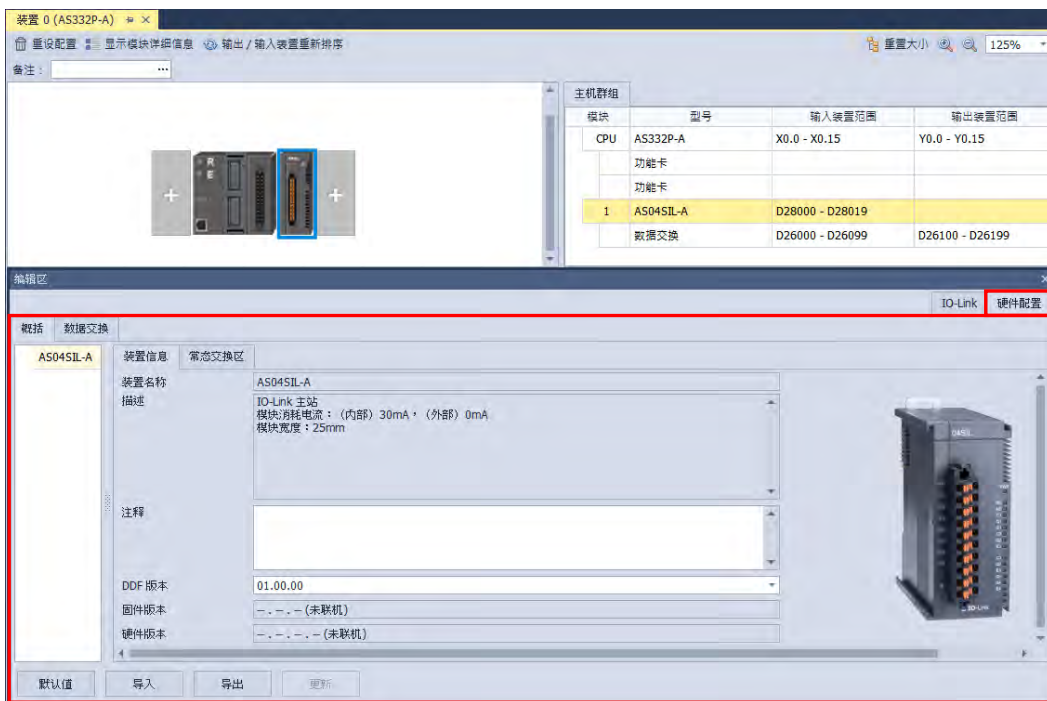


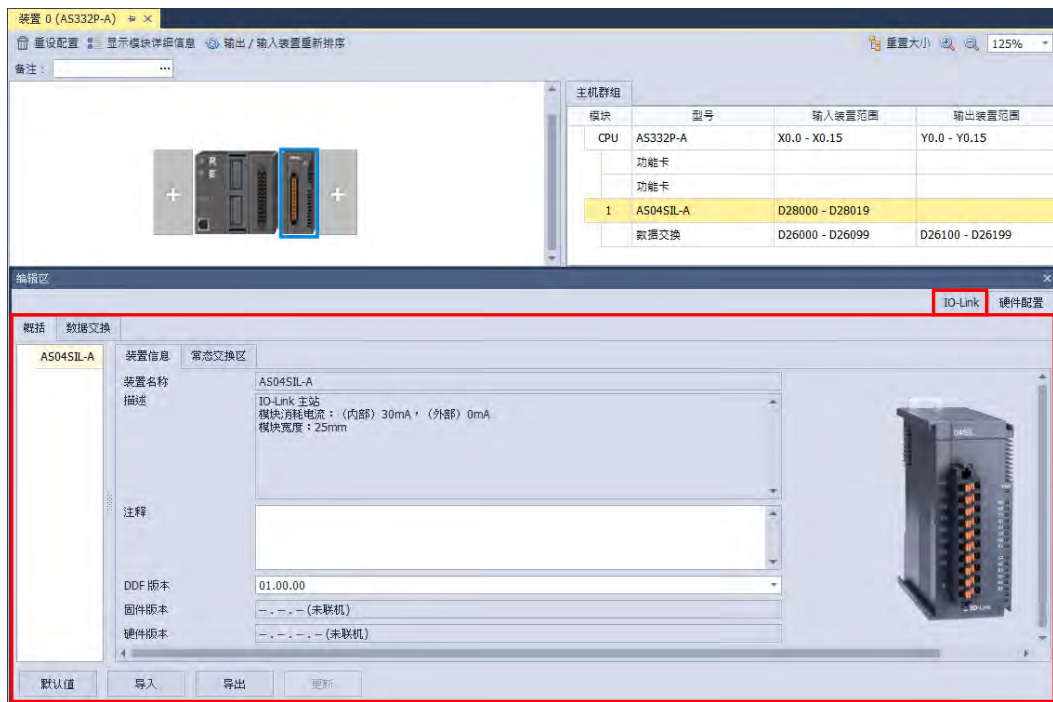
3.1.3.18 IO-Link

详细设定操作请参考 AS04SIL 模块手册。

AS04SIL-A 做为 IO-Link 主站支持 IO-Link 设备。主站与装置之间采用点对点连接，采用可靠的 3 线制技术，可通过非屏蔽标准电缆，连接智能传感器/执行器作为 IO-Link 装置。可向下兼容传统的数字传感器/执行器。电路状态和数据通道设计均基于可靠的 24VDC 技术。

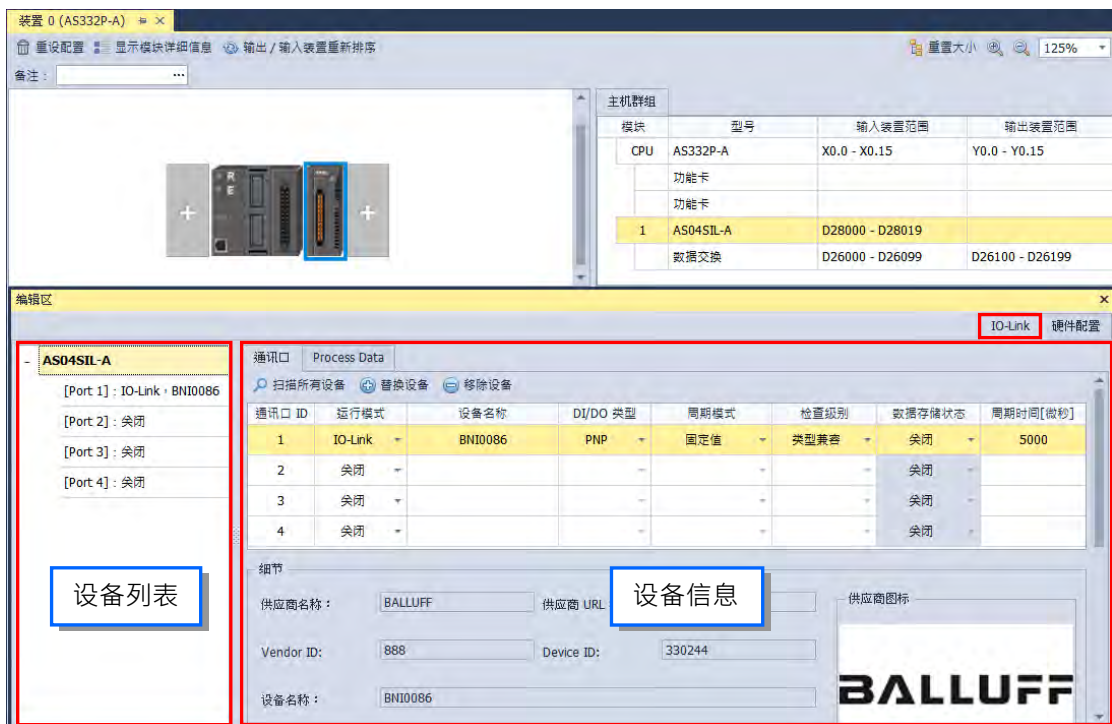
AS04SIL-A 编辑区接口有分为「硬件配置」分页为模块信息及模块相关设定、「IO-Link」分页为 IO-Link 信息及 IO-Link 相关设定，如下图所示。





3

单击 IO-Link 后编辑区接口左右分别是「设备列表」模块的设备配置、「设备信息」设备的详细信息。



「设备信息」的内容会因为「设备列表」选择「AS04SIL-A (模块)」或「设备」而显示不同信息，下列做说明。

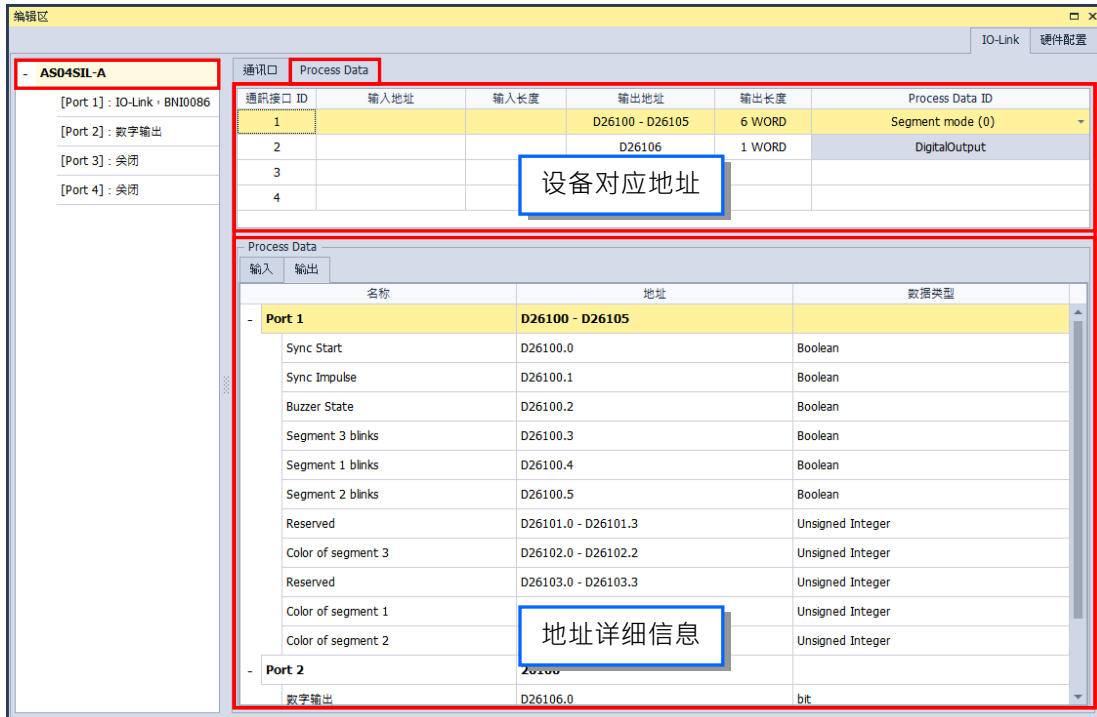
● AS04SIL-A (模块)

选择「AS04SIL-A (模块)」时，右侧「设备信息」内有「**通讯口**」与「**Process Data**」两个分页。

「**通讯口**」分页上方的「设备设定」主要功能为设备配置以及设备通信设置；「**通讯口**」分页下方的「设备信息」主要功能为显示设备设定中所选取的设备信息。



「**Process Data**」分页上方的「设备对应位置」主要功能为切换设备不同的工作模式 (参考 AS04SIL 模块手册) 以及显示所有设备对应的主机缓存器地址范围；「**Process Data**」分页下方的「地址详细信息」主要功能为显示所有设备地址所对应的详细意义。

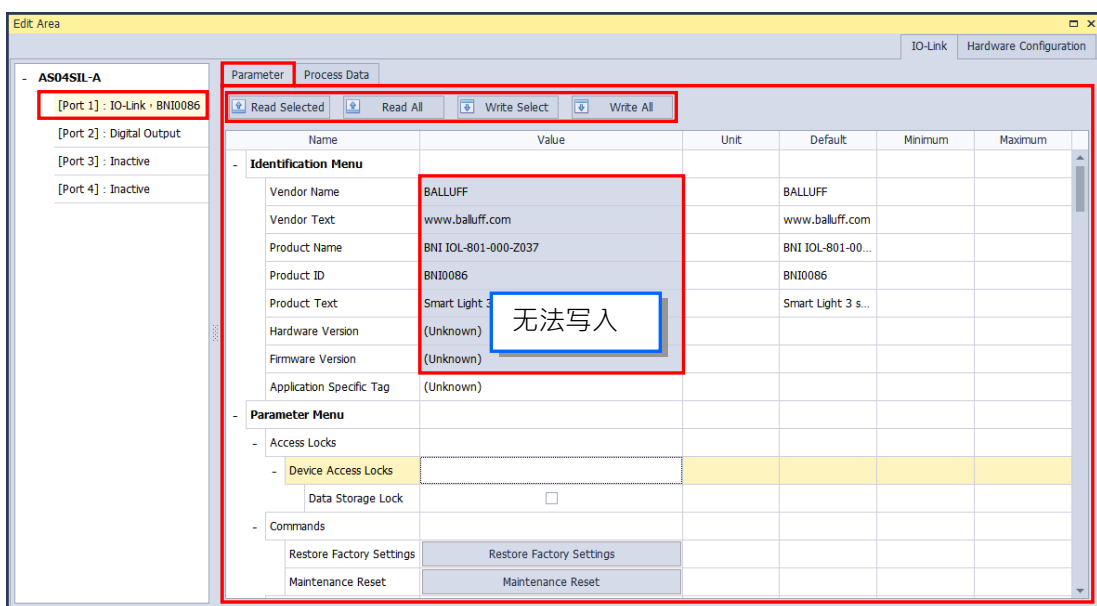


3

● 选择设备

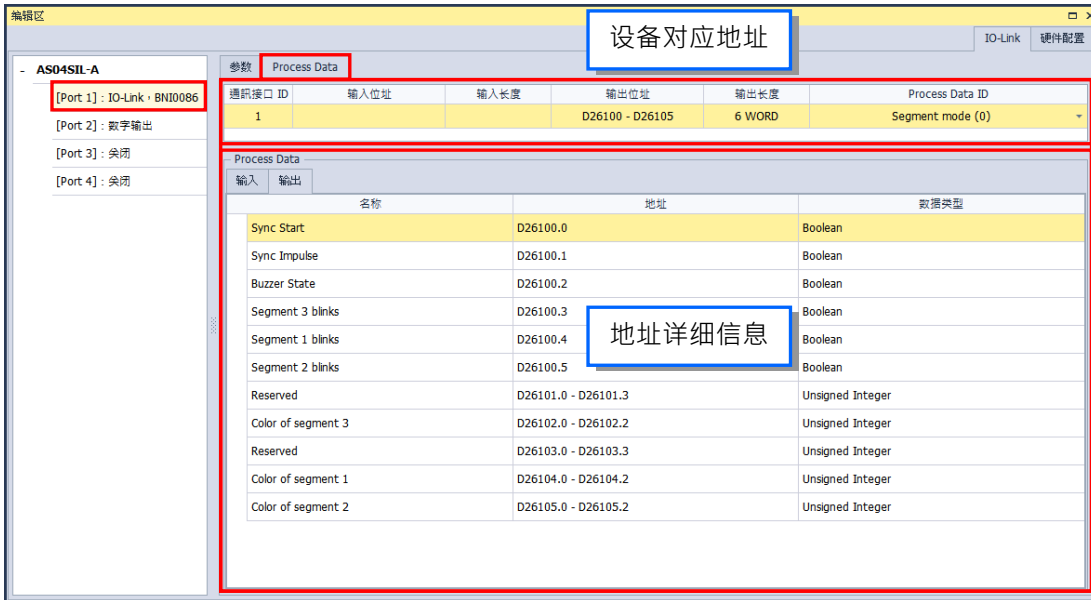
选择「模块」时，以[Port 1]为例，右侧「设备信息」内有「参数」与「Process Data」两个分页。

「参数」分页主要功能为设备参数设定，而页面上方提供包含针对单一项目或子集进行读取或写入，以及对所有参数进行读取或写入的四个选项可以使用；页面显示参数设定信息，包含参数名称、数值、单位、默认值、最小值以及最大值，若是数值字段为灰底表示该参数只能进行读取，无法写入，若为白底表示该参数可进行读取或写入。

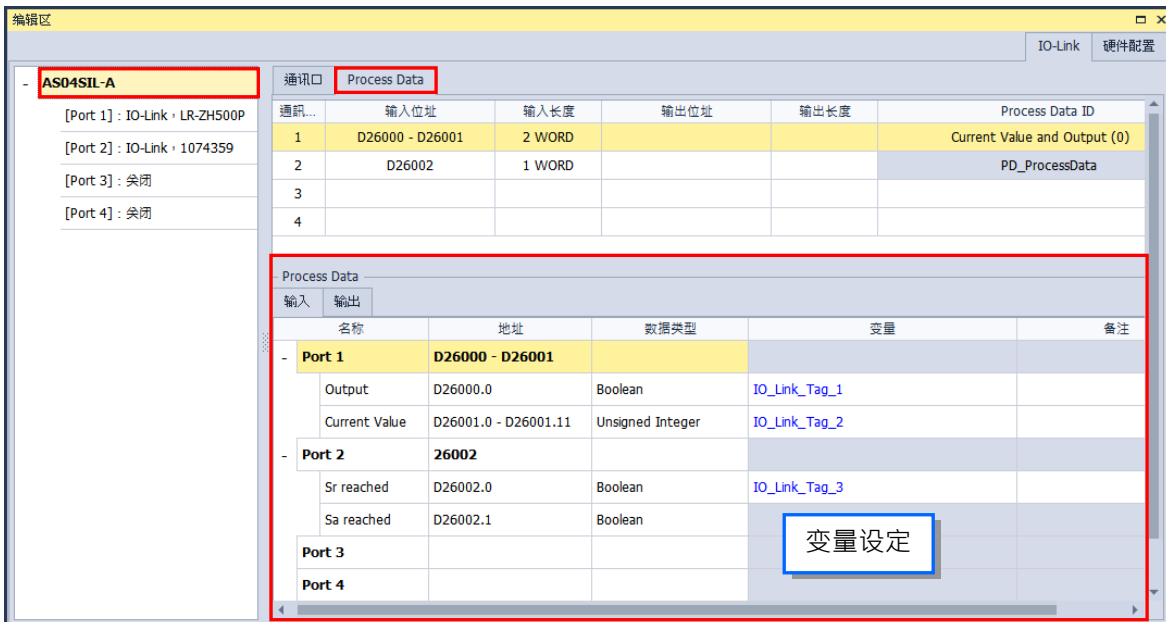


「**Process Data**」分页上方的「设备对应位置」主要功能为切换设备不同的工作模式（参考 AS04SIL 模块手册）以及显示[Port 1]设备对应的地址范围；「**Process Data**」分页下方的「位置详细信息」主要功能为显示[Port 1]设备地址所对应的详细意义。

3



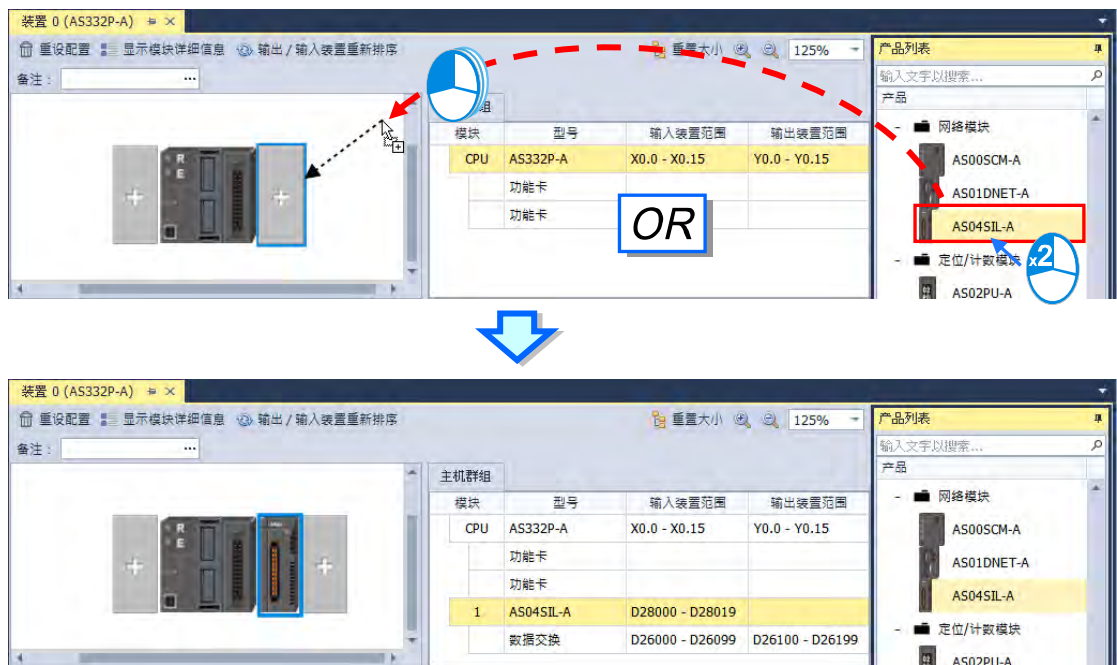
从 ISPSOft 开启 HWCONFIG 中「**Process Data**」会有符号设定功能（参考 3.1.3.18）章节，可在 ISPSOft 的全局符号中的 HWCONFIG 符号表中使用，即可在程序中存取对应的 IO-Link 设定参数。



IO-Link 应用范例

● 步骤一_加入 AS04SIL-A 模块

在软件右侧「产品列表」->「网络模块」->「AS04SIL-A」将模块加入至硬件配置区·如下图。



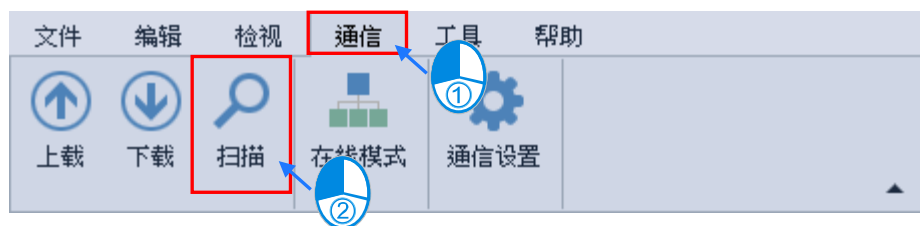
单击模块后·在下方编辑区右上角的「硬件配置」分页为模块信息及模块相关设定。

● 步骤二_加入设备

加入设备有三种方式

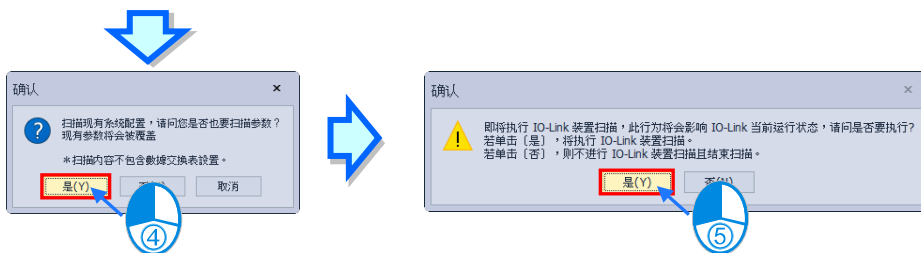
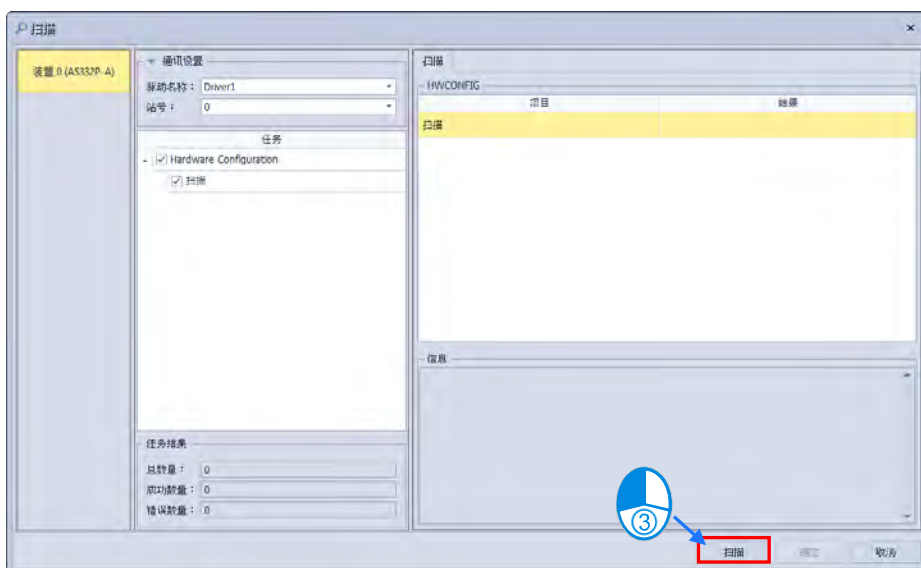
➢ 方法一_扫描所有模块的设备并加入

确认硬件都配置好之后·单击菜单工具栏「通信」->「扫描」

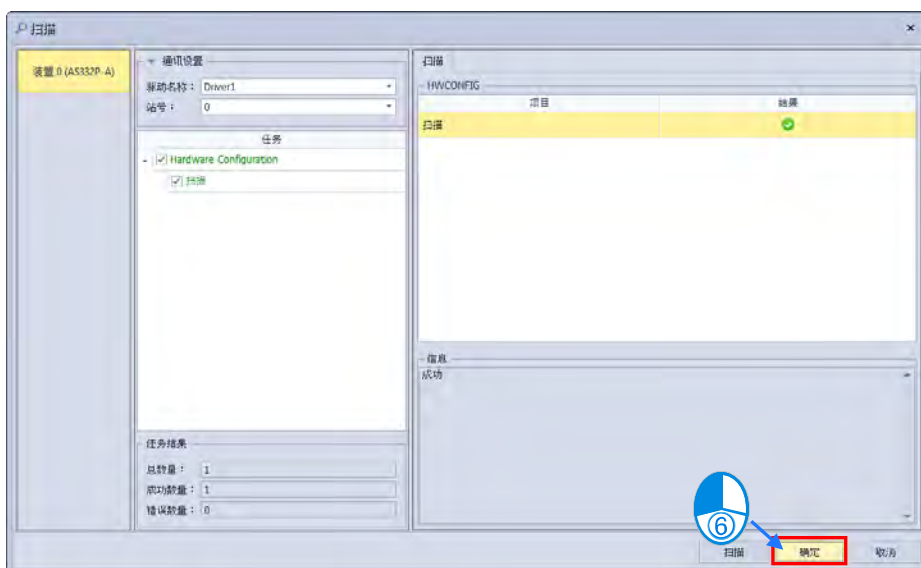


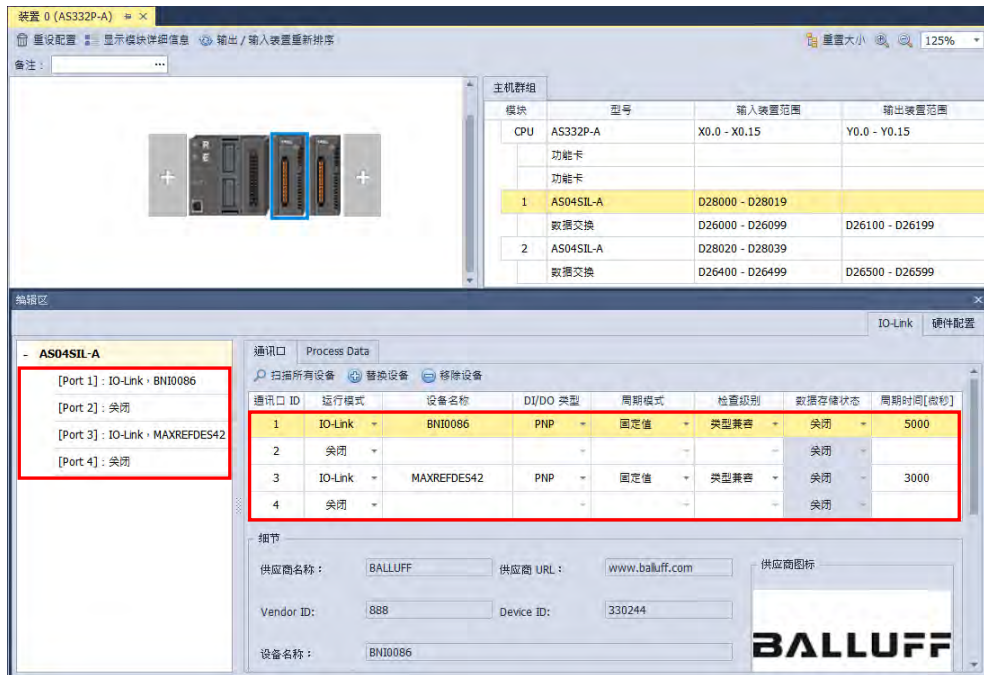
扫描窗口出现后单击「扫描」，接着单击「是」确认扫描，接下来单击「是」开始扫描 IO-Link 设备。

3



扫描完成后单击「确定」，IO-Link 设备即会自动加入至模块中。

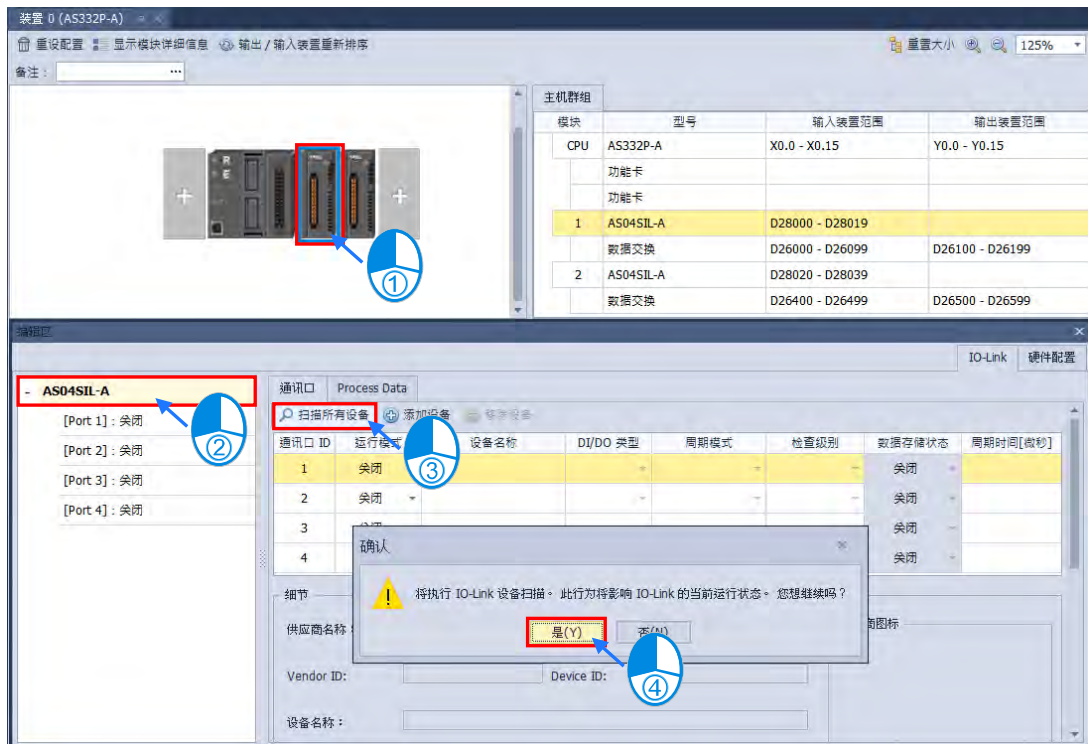




3

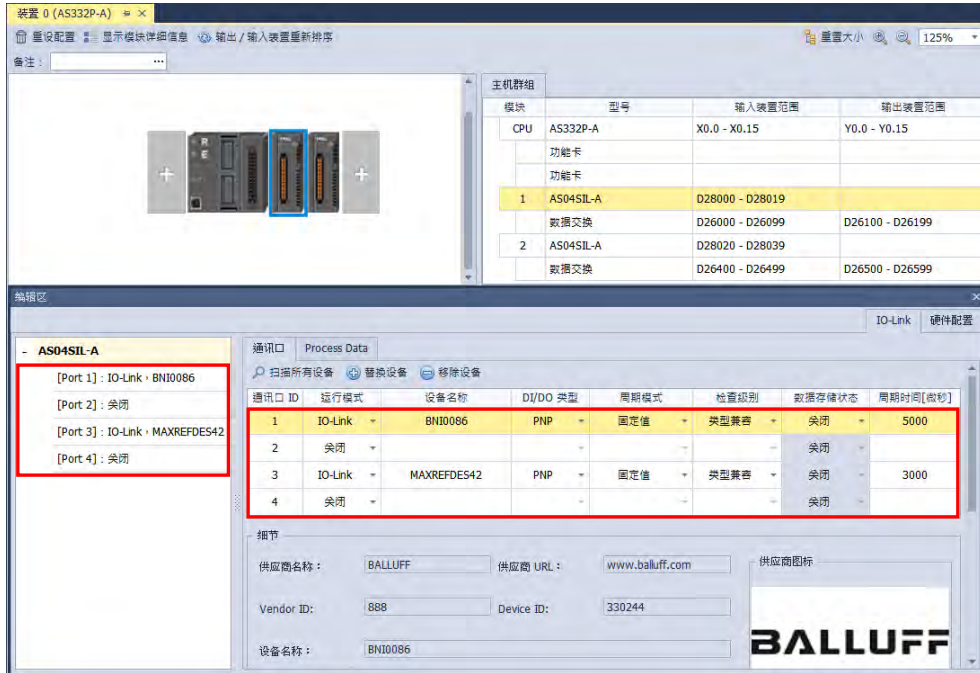
➤ 方法二_扫描单一模块的设备并加入

选择欲设定的模块，后单击编辑区左方设备列表的模块名称后，接着单击「扫描所有设备」，再确认信息单击「是」开始扫描。



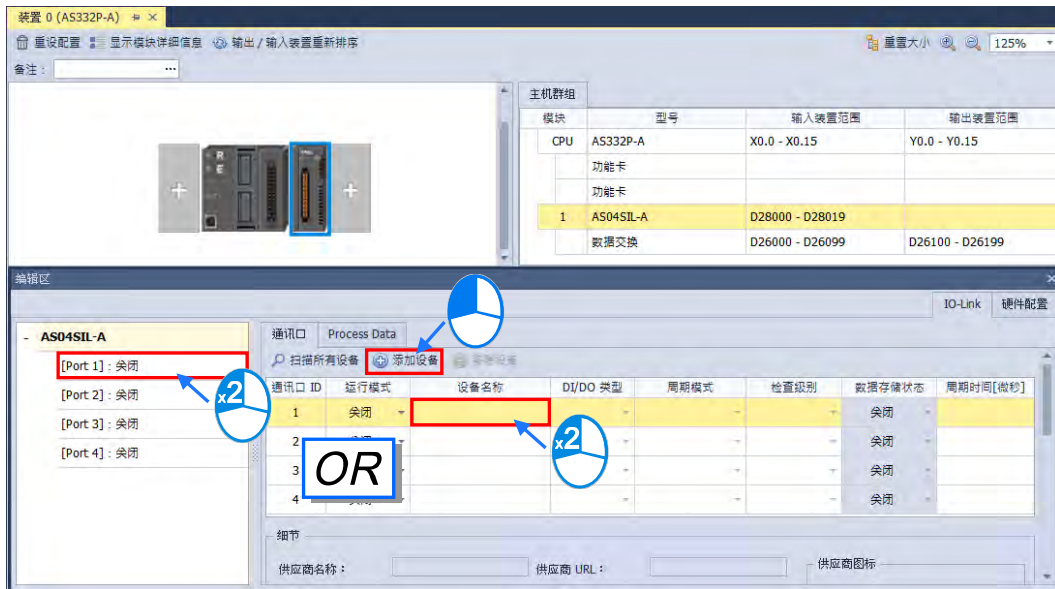
IO-Link 设备即会自动加入至模块中。

3

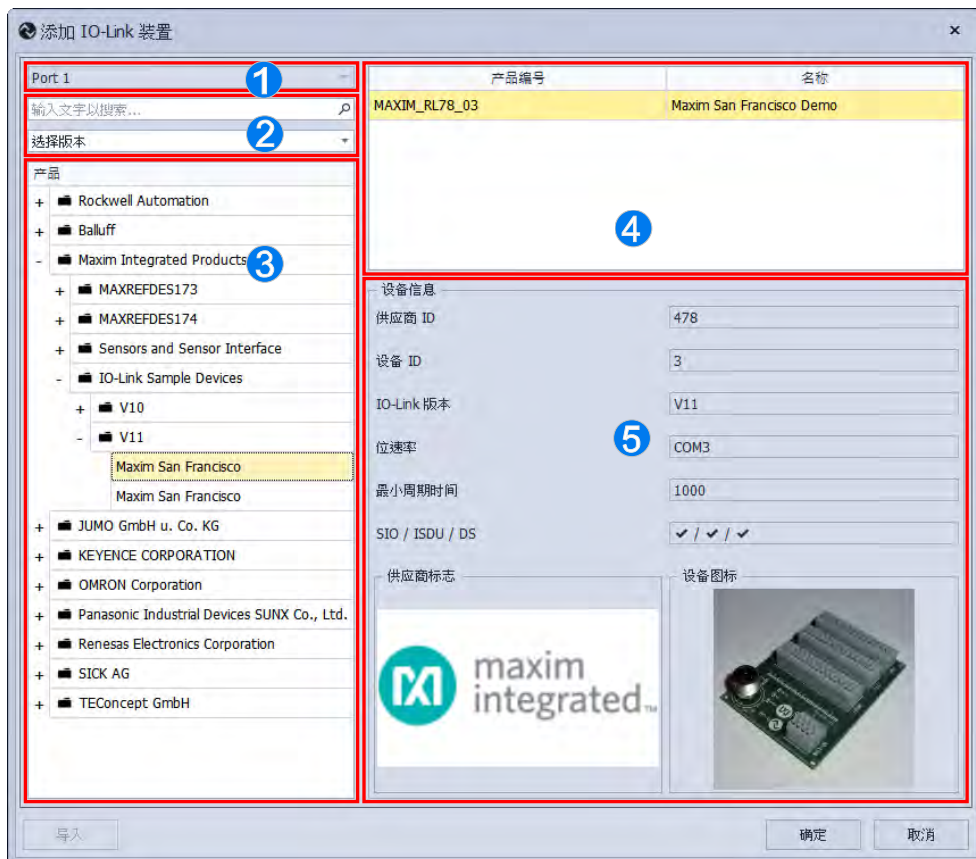


➤ 方法三_手动加入设备

若要手动在 Port 1 加入设备，先依照下方图标单击。



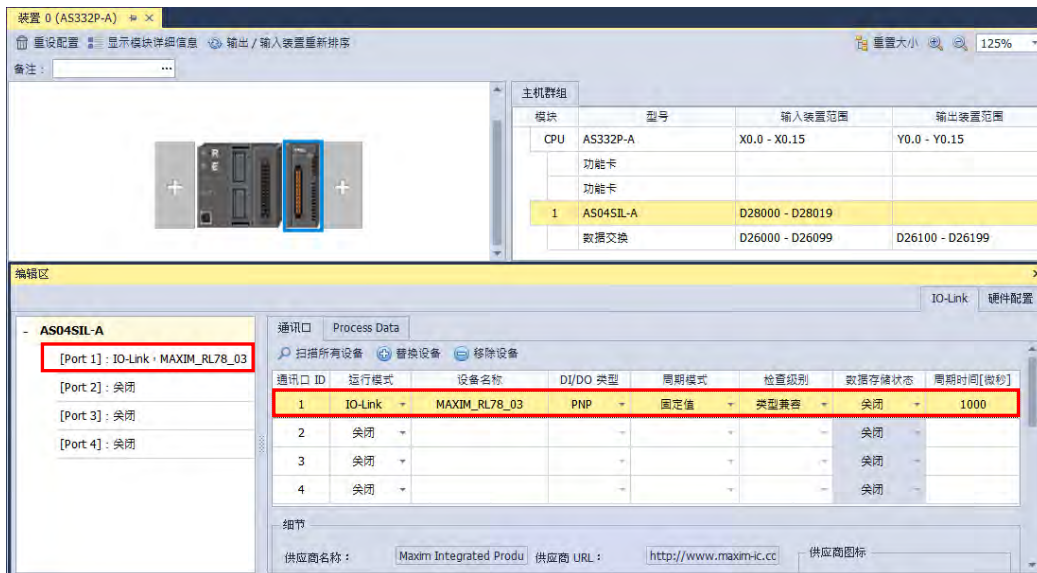
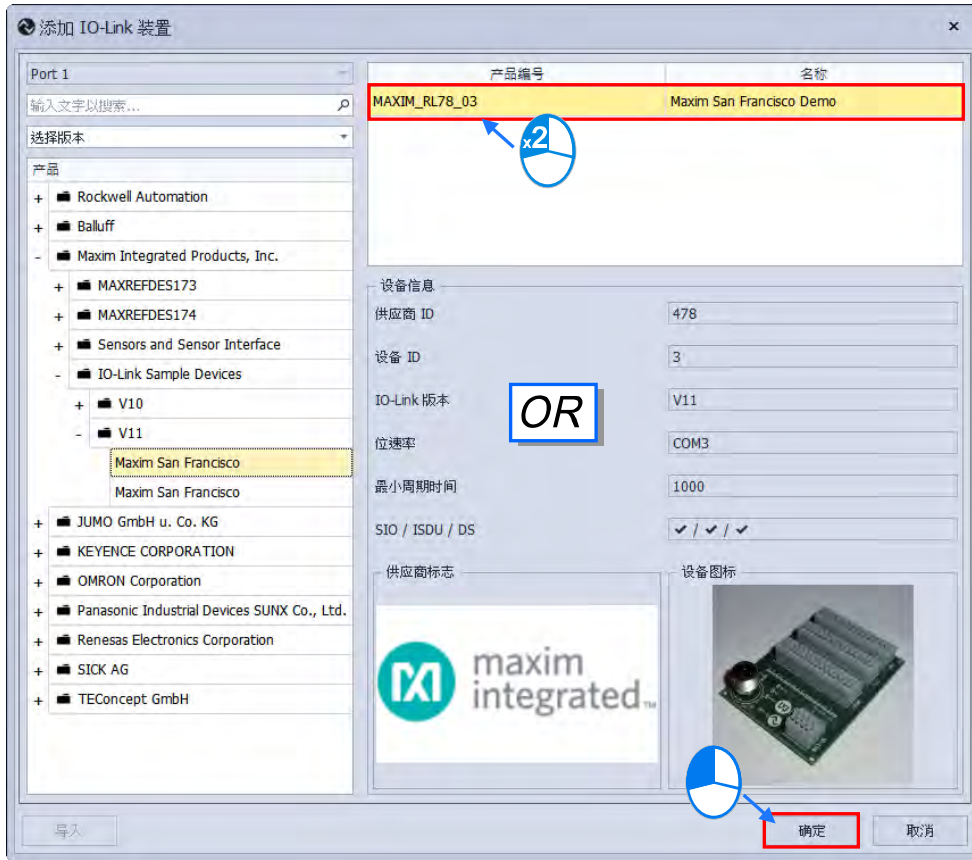
接着会开启新增 IO-Link 设备窗口，如下图



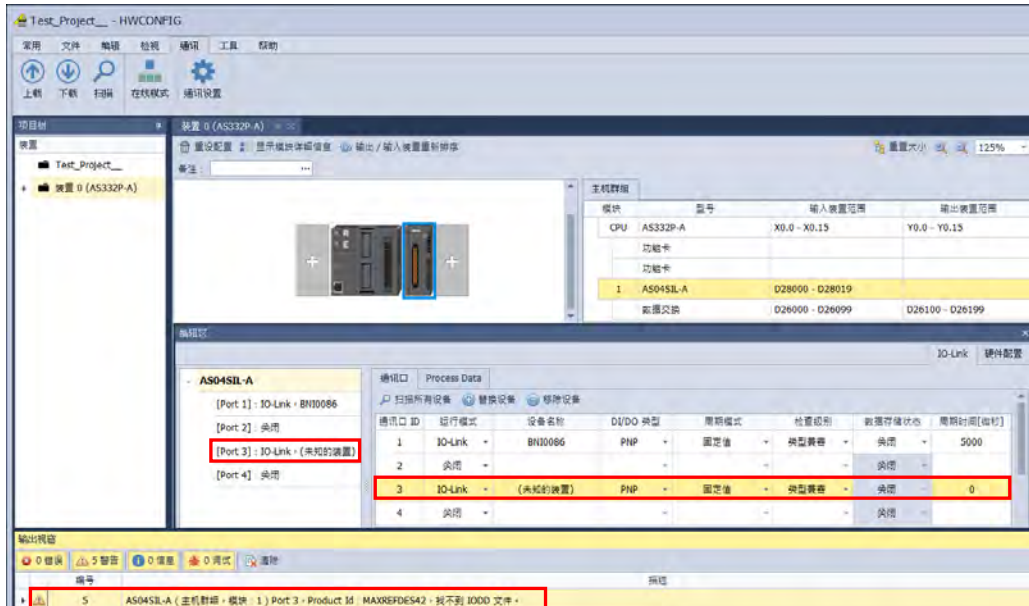
- ① 设备新增的位置。
- ② 过滤器可以输入关键词快速寻找设备。
- ③ 设备种类及版本。
- ④ 设备型号。
- ⑤ 设备详细信息。

选择好设备型号后单击确定 IO-Link 设备即会自动加入至模块中。

3



扫描后，若设备没有 IO-Link 描述文件，则会呈现未知的设备，请先重新导入描述文件后（参考 3.1.2.6），再重新扫描。

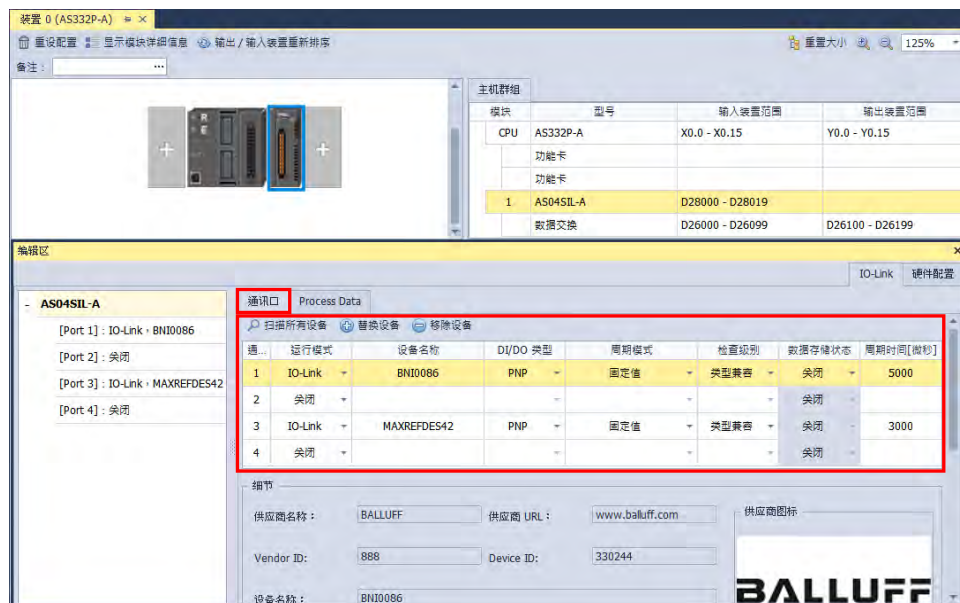


3

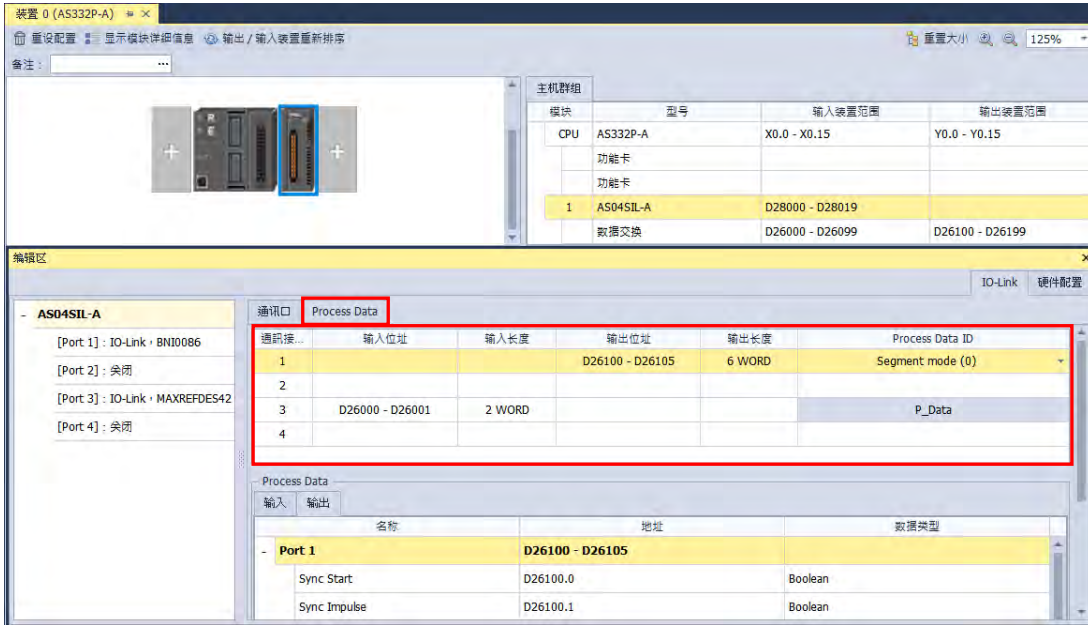
● 步骤三_设定设备配置（下载至 PLC）

(*下载并不会将设备参数下载至设备内)

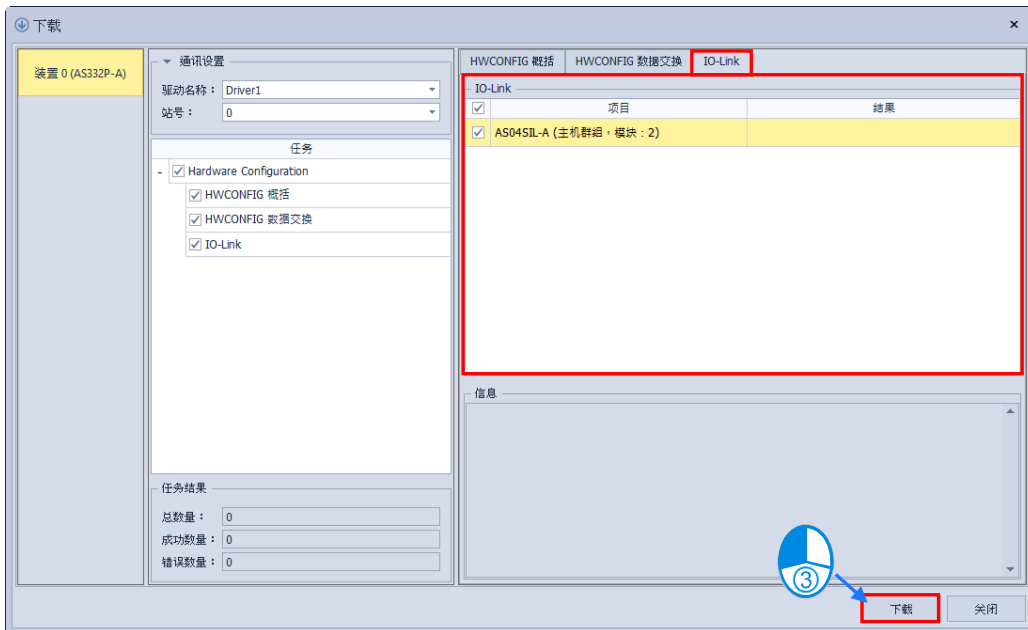
首先确认设定完所有模块的设备「通讯口」设定以及「Process Data」的工作模式后



3



单击菜单工具栏「通信」->「下载」, 下载窗口出现后, 先确认 IO-Link 项目有被勾选, 接着按「下载」将 IO-Link 设备配置下载至主机。



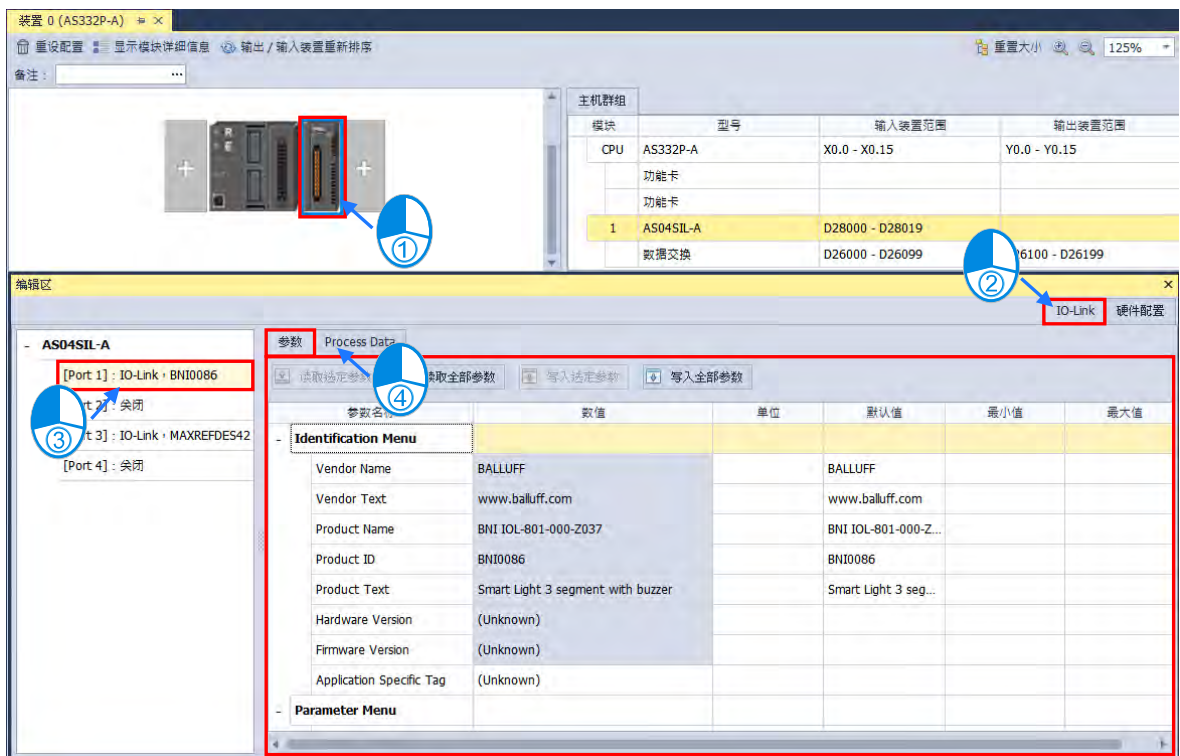
待成功画面出现, 即下载成功。

● 步骤四_设备参数设定

设备参数设定有两种，分别为「一般参数设定」以及「Process Data 参数设定」。

➢ 一般参数设定

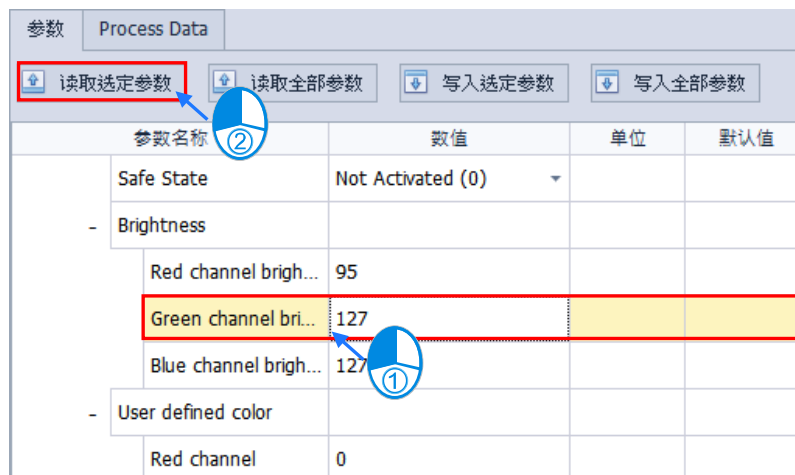
首先单击模块后，在下方编辑区 IO-Link 分页中的设备列表中选择要设定的设备，接着右侧单击「参数」页面，下方即可进行设定。



设定选项介绍（若数值为灰底的参数只可以进行读取）

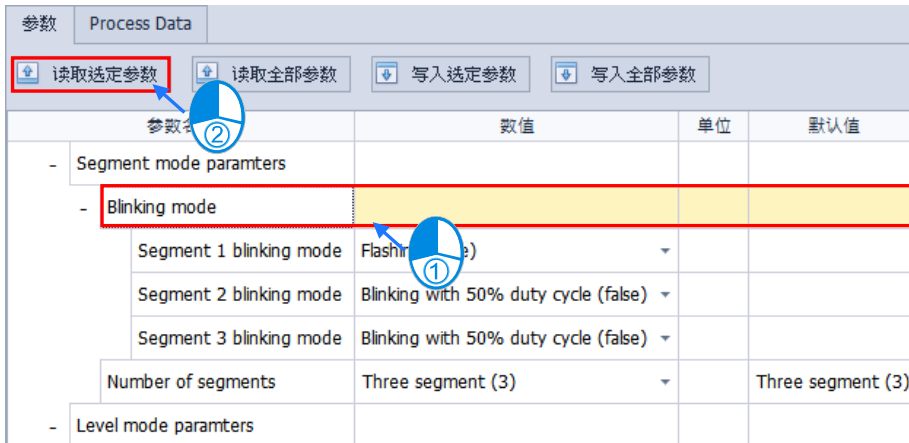
(1) 读取选定参数_针对单一项目或子集进行读取

➢ 选定想要读取的项目后，单击读取选定参数



➤ 选定想要读取的子集后，单击读取选定参数（有些单一选项无法直接读取）

3

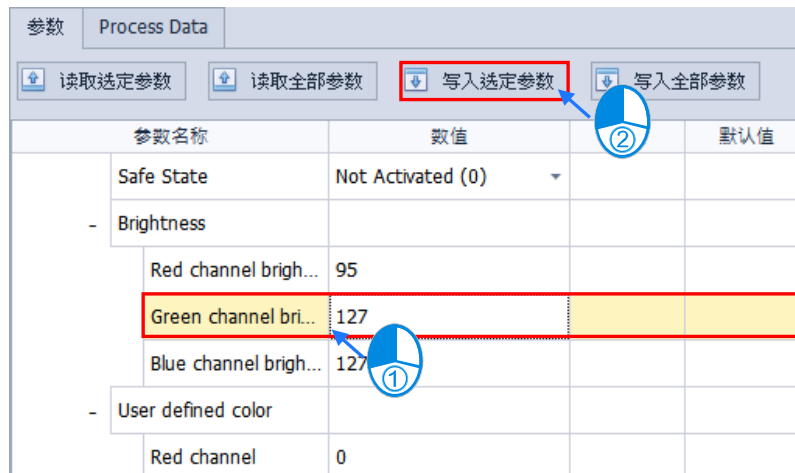


(2) 读取全部参数_针对所有项目进行读取
直接单击读取全部参数即可。



(3) 写入选定参数_针对单一项目或子集进行写入

- 选定想要写入的项目后，单击写入选定参数



3

- 选定想要写入的子集后，单击写入选定参数 (有些单一选项无法直接读取)



- (4) 写入全部参数_针对所有项目进行写入

直接单击写入全部参数即可。

3

参数			
Process Data			
<input type="button" value="读取选定参数"/> <input type="button" value="读取全部参数"/> <input type="button" value="写入选定参数"/> <input type="button" value="写入全部参数"/>			
参数名称	数值	单位	默认值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing (true)		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

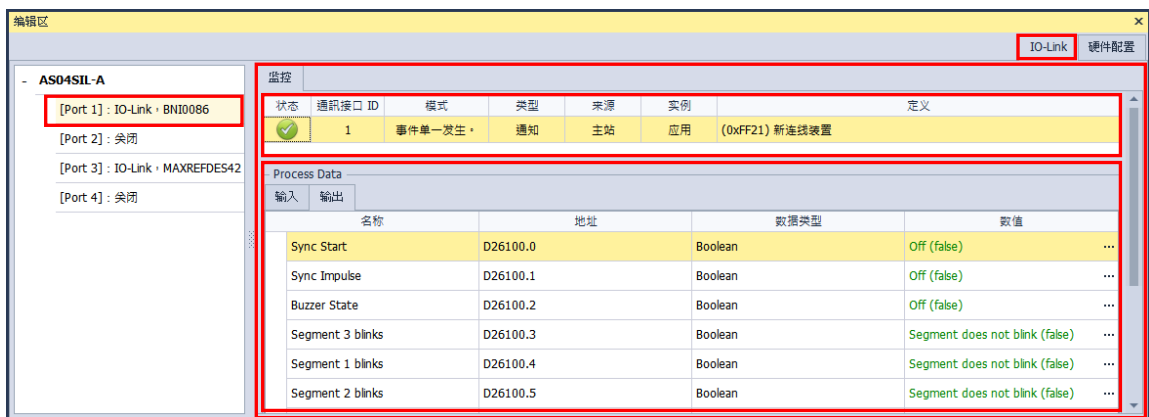
➤ Process Data 参数设定

首先单击在线模式后，单击模块，在下方编辑区 IO-Link 分页右上方的监控显示设备联机状况，右下方显示 Process Data 的参数数值。

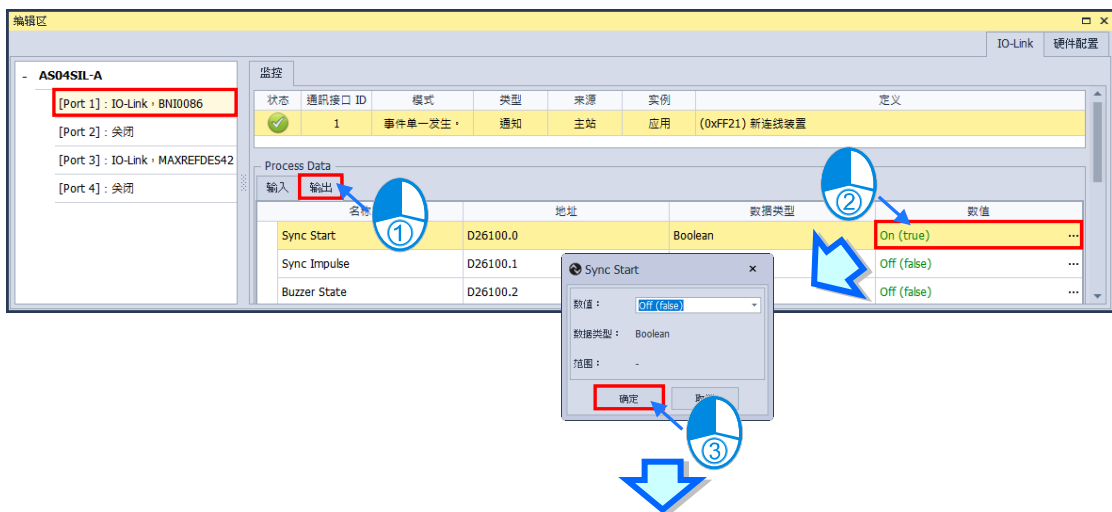
The screenshot shows the ISPSOft software interface. At the top, the '通信' (Communication) menu is highlighted with a red box and a blue arrow labeled '1'. Below it, the '在线模式' (Online Mode) button is highlighted with a red box and a blue arrow labeled '2'. A blue arrow labeled '3' points to the 'AS04SIL-A' module in the hardware configuration area. A blue arrow labeled '4' points to the 'IO-Link' tab in the 'AS04SIL-A' configuration page. The 'Process Data' tab is also visible, showing a table of parameters.

名称	地址	数据类型	数值
Port 1			
Port 2			
- Port 3	D26000 - D26001		
Ambient temperature switch...	D26000.0	Boolean	OFF (false)
Ambient temperature value	D26001.0 - D26001.15	Unsigned Integer	9064
Port 4			

可以从编辑区左侧选择单一设备进行监控



在右下方的「Process Data」选择输出分页，接着单击欲修改项目的数值字段，会跳出设定窗口，设定完成后单击确定即修改完成。



在 HWConfig 的 IO-Link 中所显示的参数值皆由电压信号经过转换后具有物理意义，经由 ISP 读取设备地址皆为电压信号，使用前请先自行确认。

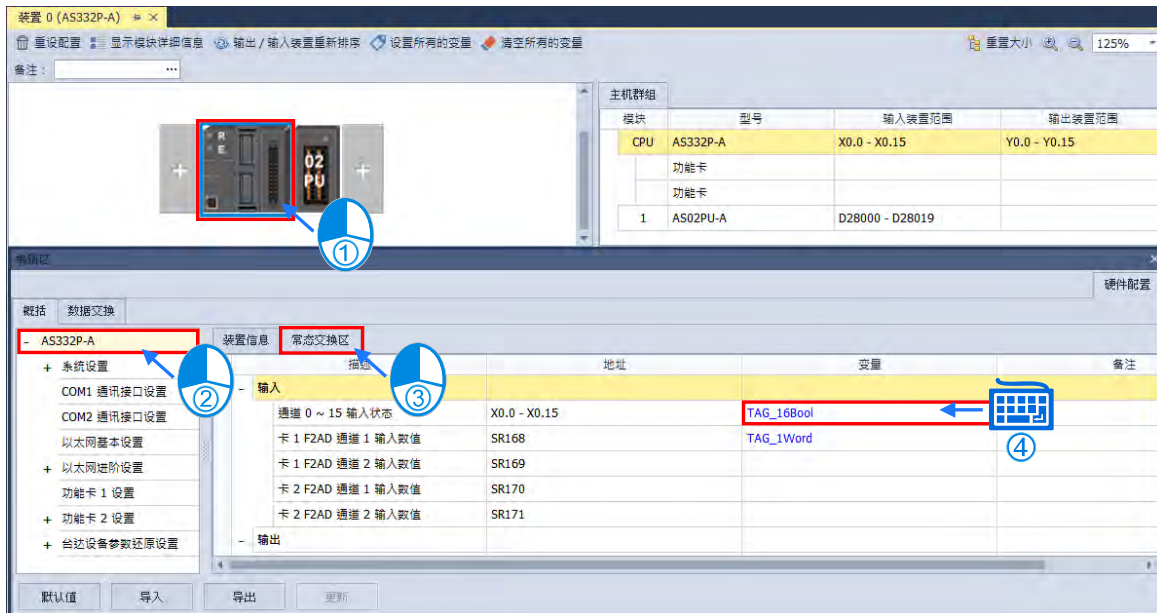
3.1.3.19 变量设定

从 ISP 开启 HWCONFIG 才支持变量设定功能，针对主机及模块的常态数据交换区 HWConfig 提供用户设定变量，之后便可在 ISPSOft 的全局变量中的 HWCONFIG 变量表中使用，即可在程序中存取对应的 HWConfig 主机及模块参数。

建立变量

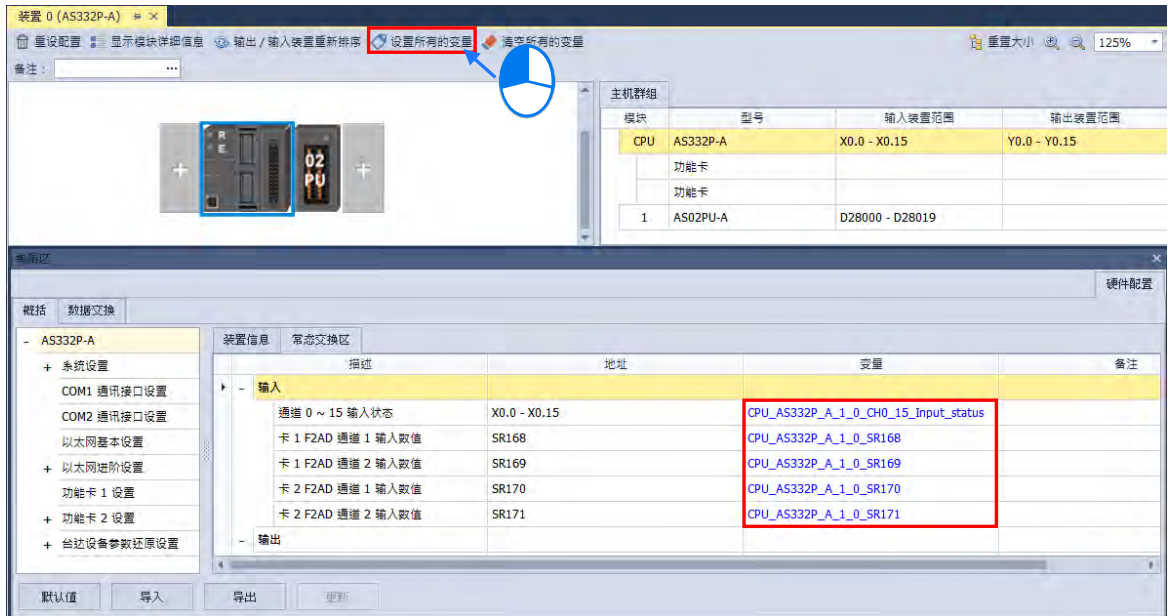
首先单击主机或模块，在下方编辑区左侧单击主机名，接着右侧单击「常态交换区」，下方即可设定变量名称及注释。

*变量命名规则：建立的变量名称最大长度，不可超过 40 个字符，变量名称不可使用特殊字符，例如 *、#、?、\、%、@ ... 等，且开头不能为数字；变量名称命名不可以重复。



也可以单击硬件配置区上方的「设定所有的变量」，一次性的建立所有变量，也可以使用「清除所有的变量」来清空变量。

*设定所有变量命名规则：模块类型_模块型号_群组_模块顺序_描述，范例：PIO_AS02PU_A_1_1_Error_code。



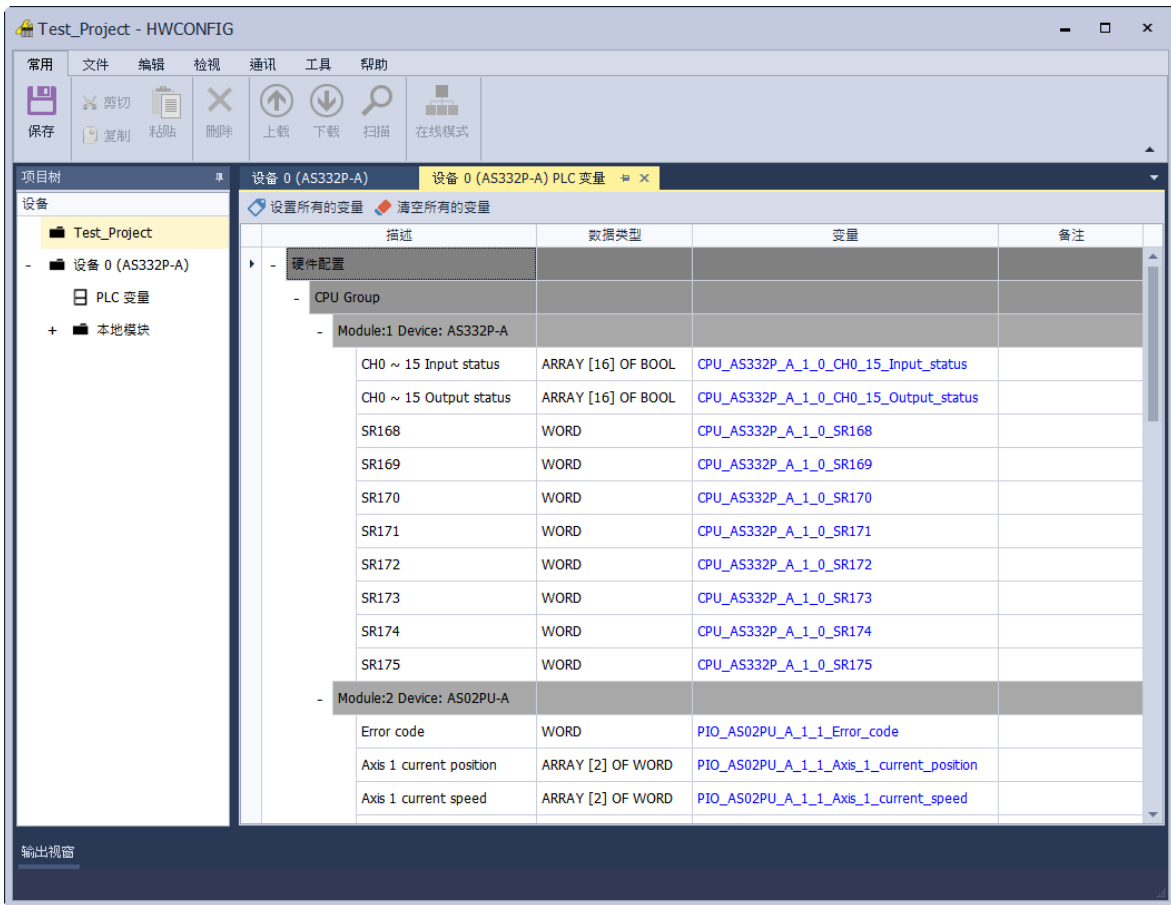
3

变量设定完成后单击菜单工具栏的「文件」>「保存」。



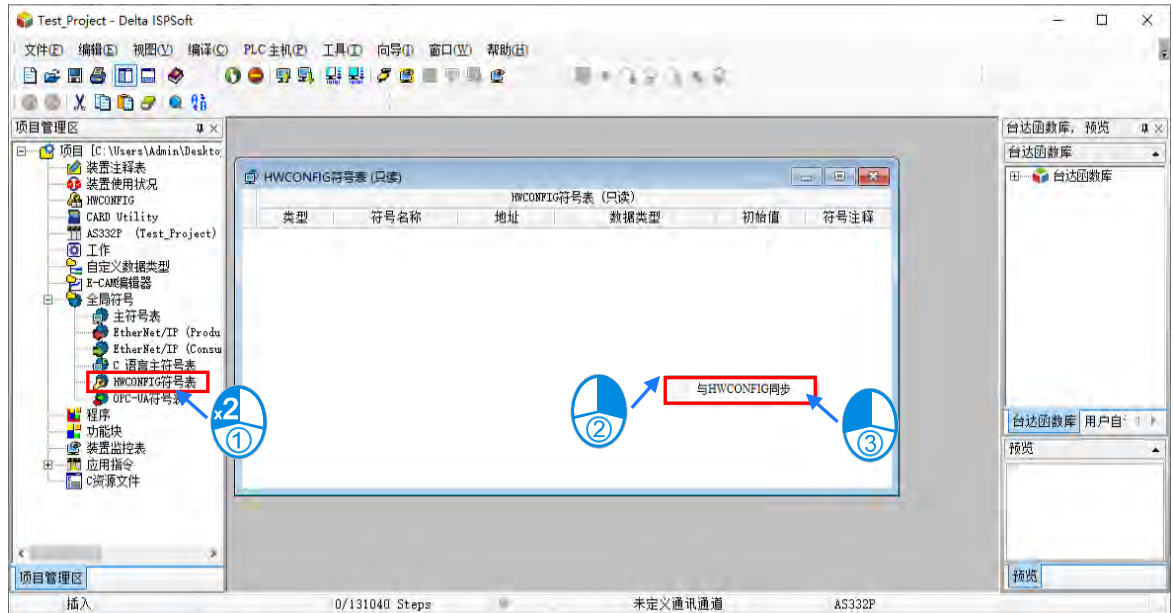
单击项目树中的「PLC 变量」，会出现变量总揽，「PLC 变量」也提供编辑变量的功能，如下图所示。

3



ISPSOft 加载 HWCONFIG 变量

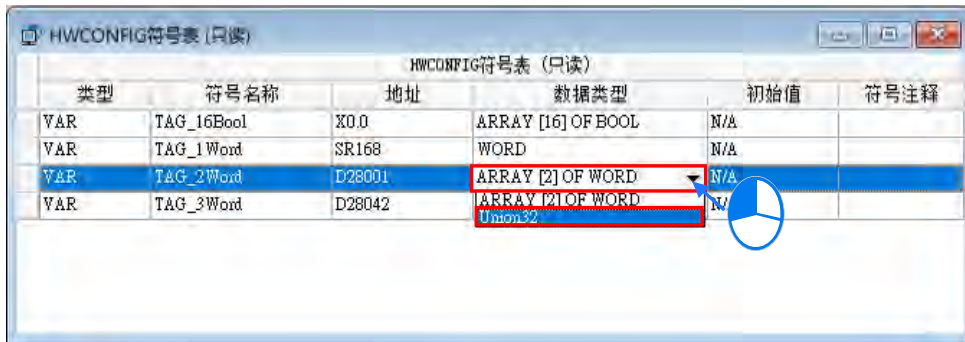
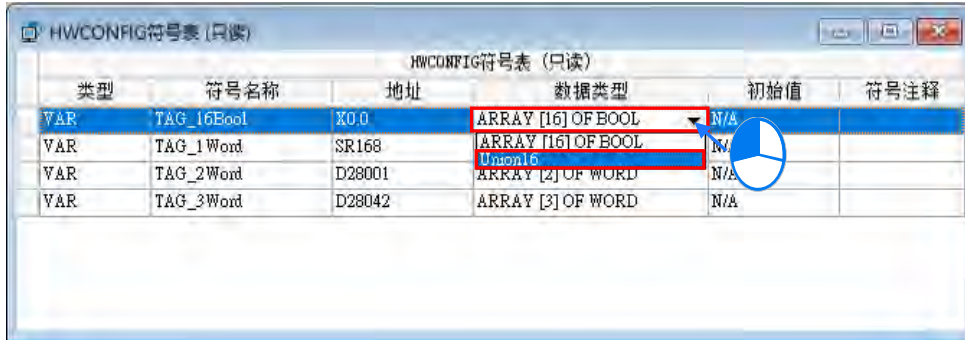
在 ISPSOft 项目树单击「全局变量」 > 「HWCONFIG 变量表」，接着在 HWCONFIG 变量表页面单击右键中的「与 HWCONFIG 同步」，就会更新 HWCONIG 变量表页面，若关闭 HWCONFIG，也会自动完成 HWCONFIG 变量表更新，之后便可在 ISPSOft 的程序中使用此变量符号。



HWCONFIG 符号表 (只读)						
类型	符号名称	地址	数据类型	初始值	符号注释	
VAR	TAG_16Bool	X0.0	ARRAY [16] OF BOOL	N/A		
VAR	TAG_1Word	SR168	WORD	N/A		
VAR	TAG_2Word	D28001	ARRAY [2] OF WORD	N/A		
VAR	TAG_3Word	D28042	ARRAY [3] OF WORD	N/A		

同步后的变量符号长度为 16 Bits (ARRAY [16] OF BOOL 或 WORD) · 可以在 HWCONFIG 变量的数据类型字段使用下拉选项将数据类型选为 Union16 ; 32 Bits (ARRAY [2] OF WORD) · 可以选为 Union32 ; 64Bits (ARRAY[4] OF WORD) · 可以选为 Union64 以方便用户以不同数据类型来使用 · 详细 Union 使用方式请参考第 8.4 节。

3



3.1.3.20 冗余功能

单击编辑区「系统信息」>「系统」>「冗余功能」单击「启动」，即可开启冗余功能。

此功能启动时，此时「硬件配置区」会出现两个冗余主背板，如下图，分别为「左侧冗余主背板」及「右侧冗余主背板」，而「右侧冗余主背板」为「左侧冗余主背板」的镜像映射，是无法进行新增、删除、变更模块等动作，若需进行此类动作，需在「左侧冗余主背板」进行。在启动冗余功能之后，「系统配置区」内所有扩展背板将自动改为冗余扩展背板；冗余主背板可配置的模块及冗余功能详细操作方式请参阅 **AH560 冗余系列机种** 该产品之相关手册。（目前仅 **AH560 冗余系列机种** 支持启动冗余功能）。



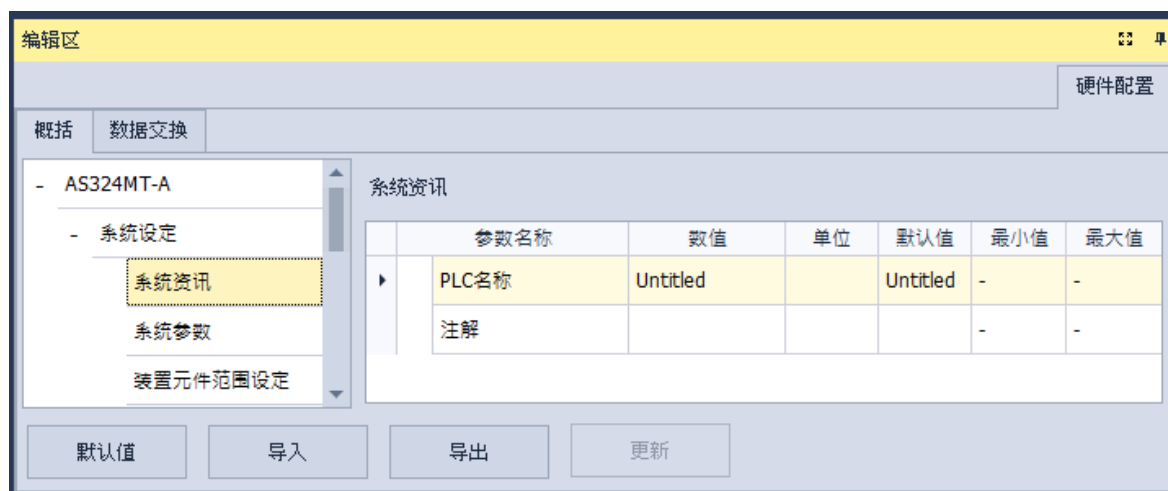
3.1.4 PLC 主机参数设定

PLC 主机参设定页面，里面的项目会随着所选取的 PLC 主机而有所不同，本章节只是初略简介项目，详细的参数设定及使用说明，还需参考各 PLC 主机的操作手册。

3.1.4.1 系统设定-系统资讯

「系统资讯」页面只有「PLC 名称」与「注解」两个字段，其中「PLC 名称」栏最多可输入 16 字符；「注解」栏最多可输入 32 个字符。所输入的字符并没有特别的限制，可输入包括特殊字符及空格键，不过须注意一个中文字将会占用两个字符。

3



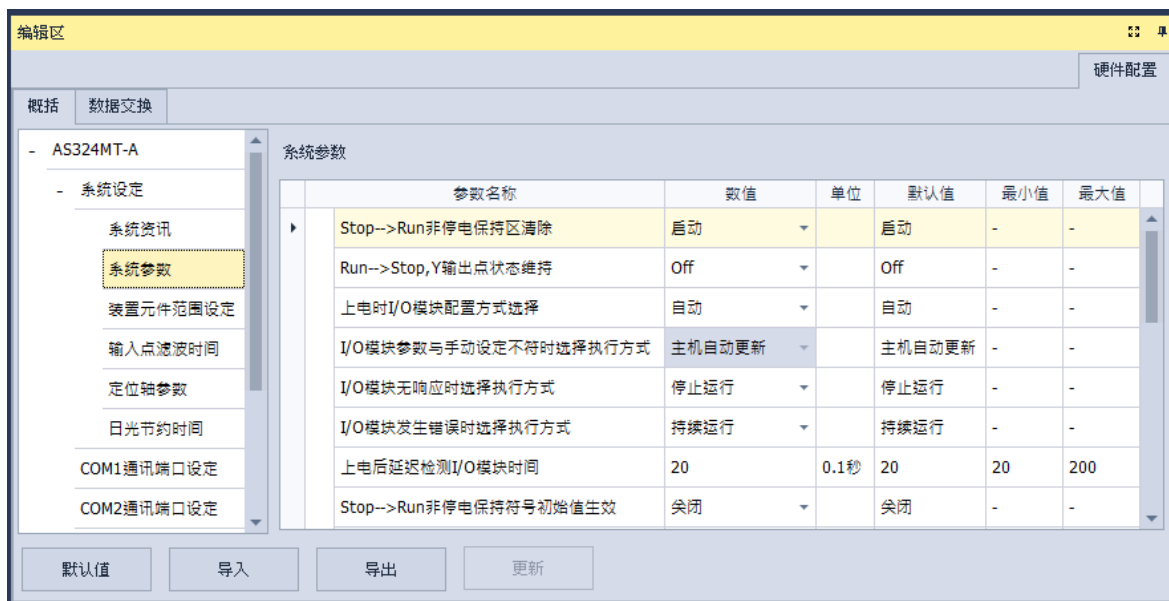
当新增一个 ISPSOft 项目时，系统便会以项目的名称作为主机的默认名称，且会显示在项目管理区的机种名称后方，而后续则可在此处重新定义。



PLC 名称的功能主要是用来做为设备识别，尤其是在多台 PLC 进行联机操作时，藉由确认 PLC 名称的方式便可防止用户的误操作；当用户要对主机进行数据的上下载或其它的联机操作时，若发现操作对象的主机名与项目中的名称不符时，系统便会提醒用户做再次的确认，以避免对其它的主机造成非预期的影响。

3.1.4.2 系统设定-系统参数

「系统参数」页面设定各项参数如下，可藉由下拉选项或在输入字段填入适当的数值。



- **Stop → Run 非停电保持区清除**

设定当 PLC 主机的状态由 Stop 切换至 Run 时，非停电保持区的所有装置状态及数值是否被清除。

- 关闭：装置状态与数值保持现状。
- 启动：装置状态与数值清除为默认值。

- **Run → Stop 时 Y 输出处理方式**

设定主机状态在 RUN 与 STOP 状态间切换时的 Y 装置状态。

- Off：将所有 Y 装置的输出都设为 Off。
- 保持：保持现状输出。

- **上电时 I/O 模块配置方式选择**

设定主机上电时自动读取模块实际配置，或由用户在 HWCONFIG 选取手动设定的配置。

- 自动：用户不须设定任何模块与参数，依据实际连接的模块运行。
- 手动：依据用户由 HWCONFIG 定义。

- 手动+标志：主机右侧第 1~32 台模块启动标志（V1.06.00 版（含）以后支持）详细操作方式需参考 AS 操作应用手册。

- **I/O 模块参数与手动设定不符时选择执行方式**

若前述上电时 I/O 模块配置方式选择设为手动时，此项目才可设定当实际模块配置与手动设定不符时的处理

- 主机自动更新：主机依据实际模块配置更新存放在主机内部的配置信息。
- 主机报警：主机回报错误。

- **I/O 模块无响应时选择执行方式**

设定当连接的扩展模块其中有通讯断线无响应的模块时，主机与其它正常的模块是否持续运行。

- 停止运行：主机停止且回报错误。
- 持续运行：主机与其它正常模块继续运行。

- **I/O 模块发生错误时选择执行方式**

设定当扩展模块出现轻微错误时的处理方式。

- 停止运行：主机停止且回报错误。
- 持续运行：主机继续运行，但会记录警告信息。

- **上电后延时检测 I/O 模块时间**

设定主机上电后，延时多久的时间再对联机的 I/O 模块与远程 I/O 检测。

- **Stop → Run 非停电保持符号初始值生效**

选取启动时，则 PLC 由停止状态变为运行状态时，非停电保持符号的值会恢复为初始值。

- **Stop → Run 停电保持符号初始值生效**

选取启动时，则 PLC 由停止状态变为运行状态时，停电保持符号的值会恢复为初始值。

- **指定主机内建 X 输入点控制 Run/Stop**

设定是否由某个 X 输入点控制主机的 Run/Stop 状态。

- 关闭：功能关闭，状态由主机上的指拨开关切换。
- 启动：使用指定的 X 输入点控制，但主机上的指拨开关操作仍有效。

- **选择主机内建 X 输入点**

若前述指定主机内建 X 输入点控制 Run/Stop 设为启动时，由下拉选项选择一个 X 输入点作为 Run/Stop 切换用。

- **启动固定扫描周期时间**

设定主机在最短在固定的时间完成一次扫描周期。



- 关闭：功能关闭。
 - 启动：则当实际的程序扫描时间小于设定值时，主机便会等待时间到达设定值，才会开始下一轮的扫描；而若实际的扫描时间大于此设定值，则主机便不理睬此设定值，而根据实际的扫描时间进行运作。
- **输入固定时间**

若前述**启动固定扫描周期时间**设为启动时，在此字段输入设定的周期时间。须注意若设定的扫描时间大于**设定扫描逾时时间**的设定值时，一旦主机开始运行后，会发生超时错误。
 - **设定扫描逾时时间**

设定的扫描逾时时间，若程序执行超过此时间，主机会回报错误。
 - **显示电池低电压警报**

当纪录时间的锂电池处于低电压状态或未安装时，可以设定是否显示警报。

 - 关闭：功能关闭。
 - 启动：锂电池处于低电压状态或未安装时，显示警报。
 - **错误历史数据存放位置**

设定选择错误记录的存放位置。

 - PLC：存放在 PLC。存满二十笔数据后，旧的纪录会被新的纪录覆盖。
 - PLC & SD Card：存放在 PLC 与 SD 卡。PLC 存满二十笔数据后，会被搬移至 SD 卡存放。
 - **COM 口通讯错误记录**

设定 COM 口发生错误时是否开启错误记录。

 - 关闭：功能关闭。
 - 启动：启动记录功能。
 - **24VDC 电源不稳定处理机制**

可选择 24VDC 电源发生跳电状况的处理机制。

 - 稳定后持续运行：当 24VDC 供应电源发生不稳定状况时，自动等待电源恢复稳定之后，再持续运行。
 - 进入快闪错误灯状态：当 24VDC 供应电源发生不稳定状况时，强制 PLC 进入快闪 ERROR 灯，并且 STOP 的状态。
 - **通讯模块刷新机制**
 - 扫描速度优先：自动分批次更新数据异动的部分，以不增加太多扫描时间为优先。
 - 数据同步优先：每一次扫描周期更新所有数据异动的部分，依数据内容同步为优先。

● **AS 远程模块刷新机制**

- PLC 程序扫描优先：使用 PLC 程序扫描时间为优先。
- 单台远程同步优先：使用单台远程同步为优先。
- 全部远程同步优先：使用全部远程同步为优先。

➤ **I601 时间中断设定**

设定 I601 时间中断的时间，搭配工作管理中的定时中断 0 使用。

➤ **I602 时间中断设定**

设定 I602 时间中断的时间，搭配工作管理中的定时中断 1 使用。

➤ **I603 时间中断设定**

设定 I603 时间中断的时间，搭配工作管理中的定时中断 2 使用。

➤ **I604 时间中断设定**

设定 I604 时间中断的时间，搭配工作管理中的定时中断 3 使用。




3.1.4.3 系统设定-装置元件范围设定

「装置元件范围设定」页面设定各项参数如下，供用户设定各种装置的停电保持区范围及子程序元件范围。



参数名称	数值	单位	默认值	最小值	最大值
停电保持D元件起始位址	20000	...	20000	0	29999
停电保持D元件结束位址	23999		23999	0	29999
停电保持16位元C元件起始位址	448	...	448	0	511
停电保持16位元C元件结束位址	511		511	0	511
停电保持32位元C元件起始位址	128	...	128	0	255
停电保持32位元C元件结束位址	255		255	0	255
停电保持M元件起始位址	6000	...	6000	0	8191
停电保持M元件结束位址	8191		8191	0	8191
停电保持S元件起始位址	512	...	512	0	2047
停电保持S元件结束位址	1023		1023	0	2047
副程式型T元件起始位址	412	...	412	0	511
副程式型T元件结束位址	511		511	0	511

单击  打开设定窗口，可设定装置的起始与结束地址。单击「默认值」按钮恢复默认值，单击「清除」按钮将字段清除为 0，完成设定后单击「确定」以保存设定；或不保存设定单击「取消」离开页面。



参数设置

起始地址: 20000

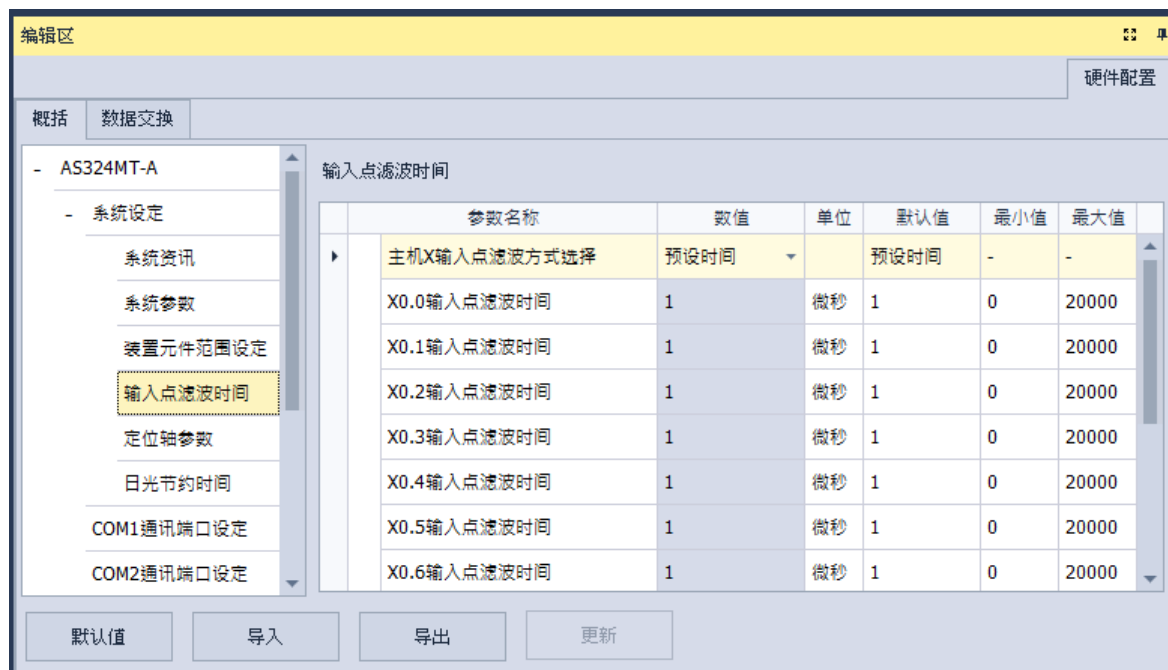
结束地址: 23999

默认值 清除

确定 取消

3.1.4.4 系统设定-输入点滤波时间

「输入点滤波时间」页面用于设定主机上每个输入点的滤波时间，若输入点接收到的信号状态持续时间小于滤波时间，则会被视为噪声将其过滤，用户可根据现场应用环境的干扰状况设定适当的滤波时间。



➤ 主机 X 输入点滤波方式选择

设定是否设定各输入点的滤波时间或使用默认值

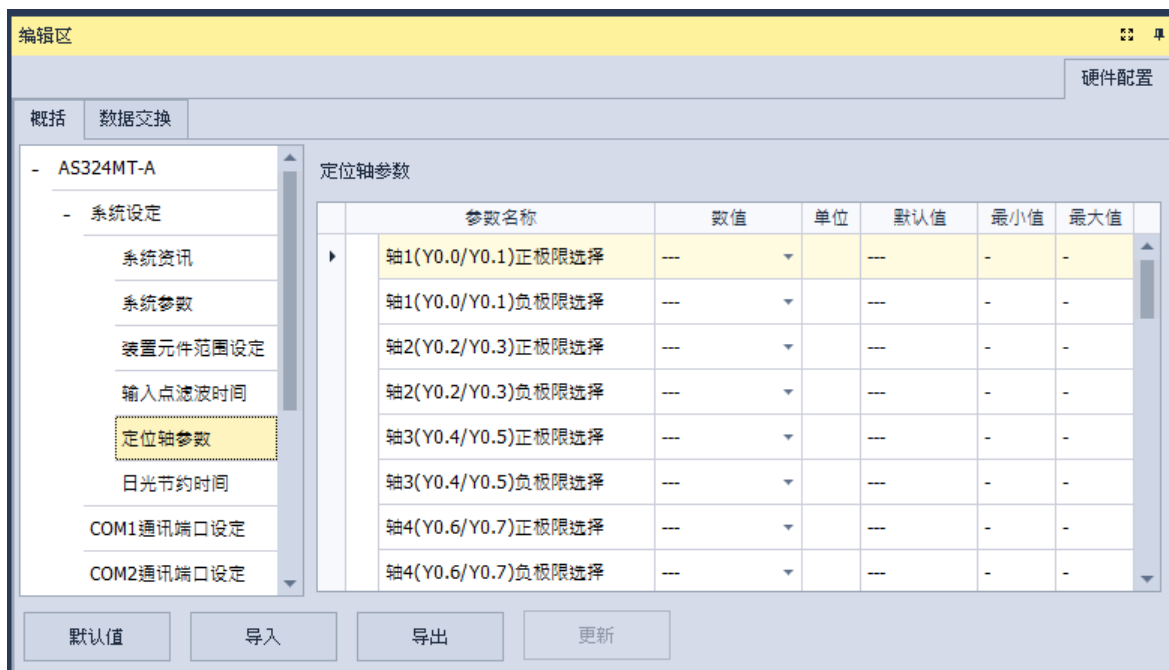
- 默认时间：使用默认的滤波时间。
- 自定义时间：由用户自行设定每个 X 输入点的滤波时间。

➤ X0.0~X0.15 输入点滤波时间

若前述主机 X 输入点滤波方式选择设为自定义时间时，可在此输入 X0.0~X0.15 各自的滤波时间。

3.1.4.5 系统设定-定位轴参数

「定位轴参数」页面用于设定指定输入点作为轴 1~轴 6 各通道的正极限与负极限，可设定共 12 个极限点；并可设定指定输入点作为轴 1~轴 6 的 Z 相触发点、原点回归后清除输出单击择；以及对正、负向软件极限位置做设定。



- 轴 1 (Y0.0/Y0.1) 正、负极限选择 ~ 轴 6 (Y0.10/Y0.11) 正、负极限选择
由下拉选项选择上升沿或下降沿触发及输入的 X 接点。
- 轴 1 (Y0.0/Y0.1) Z 相触发选择 ~ 轴 6 (Y0.10/Y0.11) Z 相触发选择
由下拉选项选择上升沿或下降沿触发及输入的 X 接点。
- 轴 1 (Y0.0/Y0.1) 原点回归后清除输出单击择 ~ 轴 6 (Y0.10/Y0.11) 原点回归后清除输出单击择
由下拉选项选择输出接点。
- 轴 1 (Y0.0/Y0.1) 正、负向软件极限位置设定 ~ 轴 6 (Y0.10/Y0.11) 正、负向软件极限位置设定
可在字段设定用户需求的脉冲数当成轴 1 ~ 轴 6 的正、负向软件极限位置，设定范围为 -2147483648 ~ 2147483647。

3.1.4.6 系统设定-日光节约时间

「日光节约时间」页面用于打开或关闭日光节约时间功能及设定进行日光节约时间的日期。打开日光节约时间时，设定的时间区段内时钟将自动提早设定的节约时间。

3

编辑区

硬件配置

概括 数据交换

- AS324MT-A

系统设定

系统资讯

系统参数

装置元件范围设定

输入点滤波时间

定位轴参数

日光节约时间

COM1通讯端口设定

日光节约时间

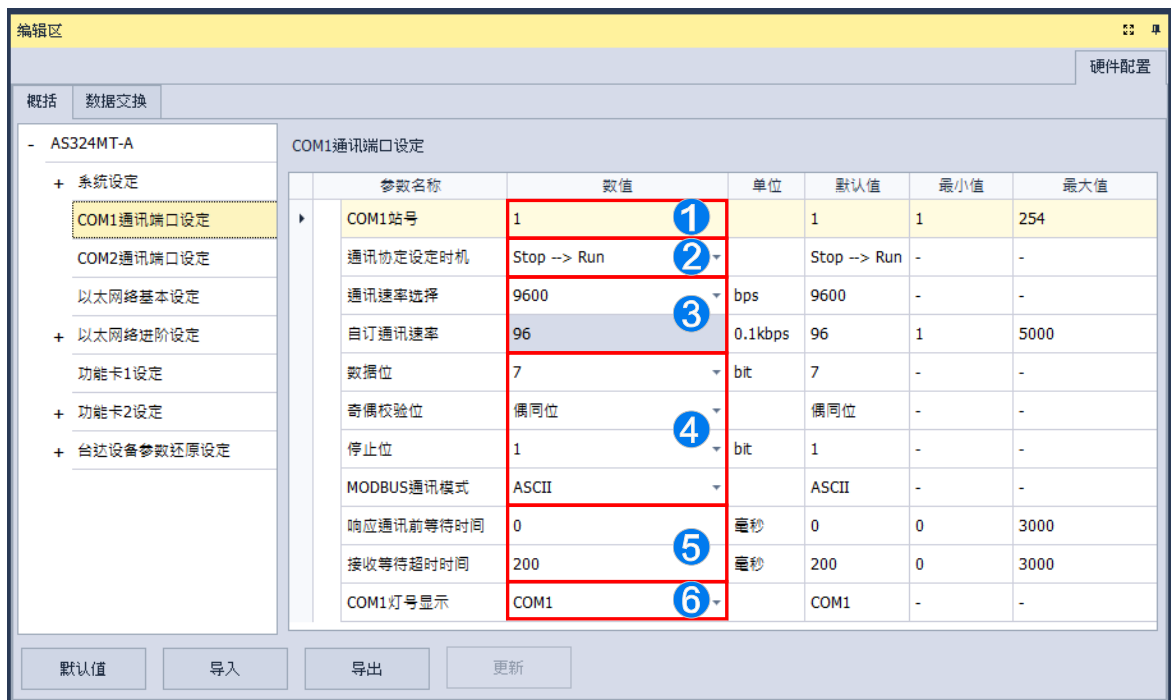
参数名称	数值	单位	默认值	最小值	最大值
日光节约时间关闭/开启	关闭		关闭	-	-
开始日期-月	4		4	3	5
开始日期-日	1		1	1	31
结束日期-月	10		10	9	11
结束日期-日	1		1	1	31
节约时间	60	分钟	60	1	180

默认值 导入 导出 更新

- **日光节约时间关闭/打开**
设定关闭或打开日光节约时间。
- **开始日期-月**
设定开始进行日光节约时间的月份。
- **开始日期-日**
设定开始进行日光节约时间的日期。
- **结束日期-月**
设定结束进行日光节约时间的月份。
- **结束日期-日**
设定结束进行日光节约时间的日期。
- **节约时间**
设定调整提早的日光节时间，单位为分钟。

3.1.4.7 COM 通讯端口设定

「COM 通讯端口设定」在各系列機種依据型号不同有具备 1 或 2 个 COM 端口，单击项目列表中的 **COM1 通讯端口设定** 或 **COM2 通讯端口设定**，可分别设定 COM1 与 COM2 参数，两 COM 端口设定项目皆相同。



- ❶ 设定此通讯端口的通讯站号，做为通讯上的装置识别，且连接在同一条线路上的站号不允许重复。
- ❷ 设定此通讯端口的设定生效的时机，选择 **Stop → Run** 为主机状态由 Stop 切换为 Run 时生效，选择**上电时**则为主机上电时生效。
- ❸ 设定此通讯端口所采用的通讯速率 (Baud Rate)。可由下拉选项选择预先定义的通讯速率之一，或选择**自定义**，输入自定义通讯速率。
- ❹ 设定通讯相关设定。
- ❺ 设定**响应通讯前等待时间**，当 AS 主机接收到通讯命令后等待设定的时间再进行回复，配合远程设备模式切换的时间。设定**接收等待超时时间**，当 AS 主机作为主站时送出通讯命令，等待设定的时间后未收到回复则视为超时。
- ❻ 设定 COM1 灯号显示，若下拉选项选择为 **COM1**，则通讯灯号显示的为 COM1 的 RS485 通讯状态，若选择为 **Card1**，则通讯灯号显示的为扩展卡 **CARD1** 的通讯状态。
设定 COM2 灯号显示，若下拉选项选择为 **COM2**，则通讯灯号显示的为 COM2 的 RS485 通讯状态，若选择为 **Card2**，则通讯灯号显示的为扩展卡 **CARD2** 的通讯状态。

3.1.4.8 以太网基本设定

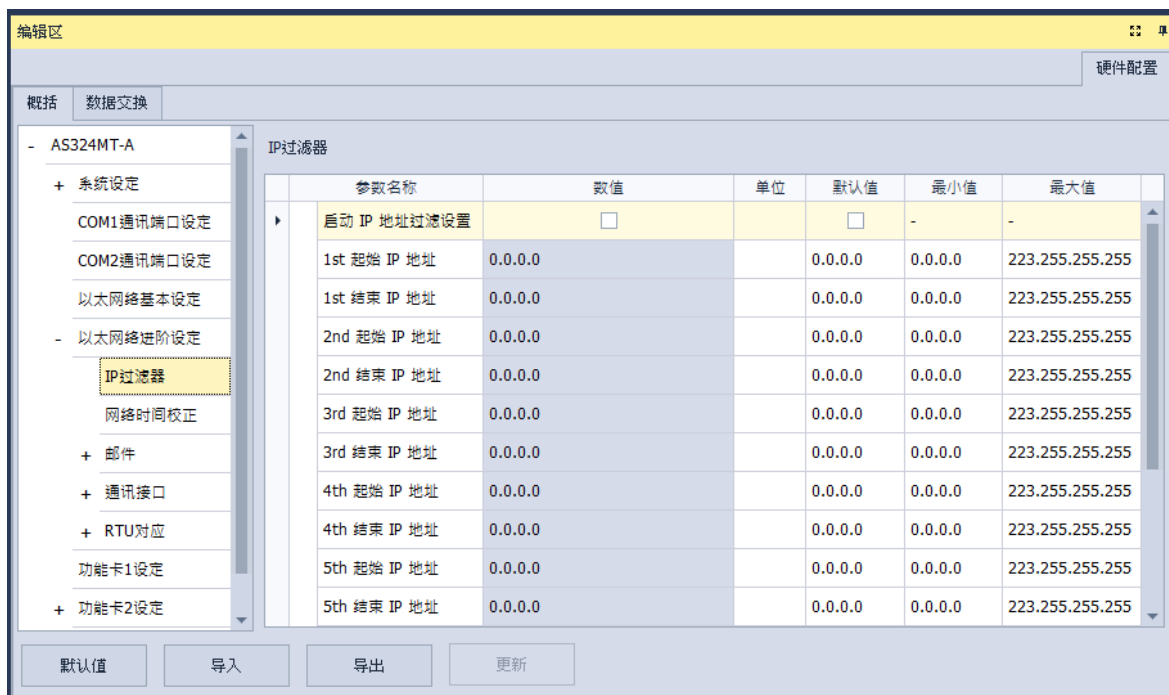
在项目列表中单击「以太网基本设定」项目即可进入。此处主要用于设定主机以太网（Ethernet）端口的通讯参数，关于通过 Ethernet 来与主机进行联机。



设定主机 IP 的寻址模式。若设为「静态」，代表 IP 地址等参数将由用户直接指定。但若设为「DHCP」或「BOOTP」，则代表 IP 地址将经由 DHCP/BOOTP 来动态分配。

3.1.4.9 以太网进阶设定-IP 过滤器

「IP 过滤器」页面提供用户设定网络装置的过滤功能，通过此设定便可多加一层的确认机制，以确保通讯的对象为我们所允许的装置。此功能可用以限定通讯对象。当此功能被启用后，只有 IP 地址列于列表范围中的装置才允许跟主机进行通讯。此功能的设定步骤如下，而清单中最多可设定 8 组的地址范围。

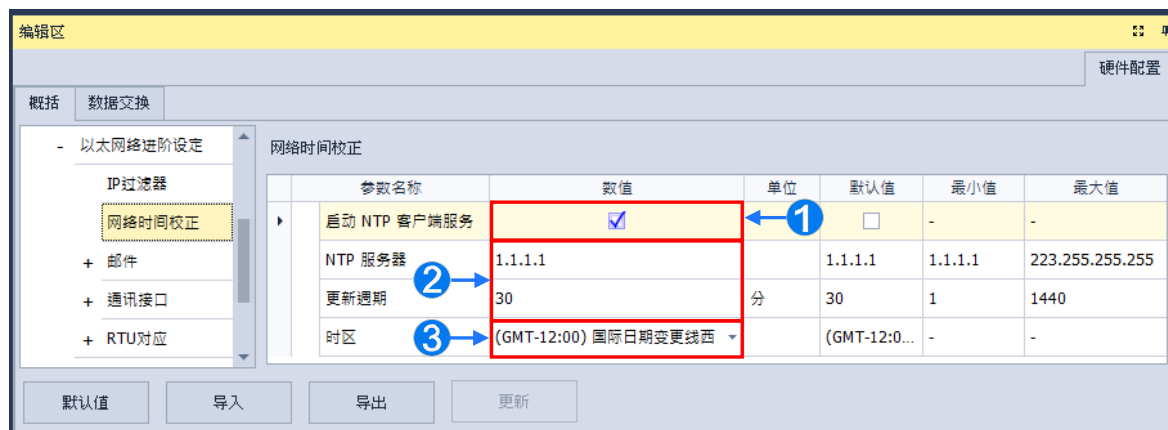


要启动 IP 过滤器功能，请勾选启动 IP 地址过滤设置，便可输入以下 1~8 组的 IP 过滤范围的起始与结束地址。

3.1.4.10 以太网进阶设定-网络时间校正

「网络时间校正」让主机可以通过 NTP 服务器来进行时间校正的功能，并可于此处进行相关设定。

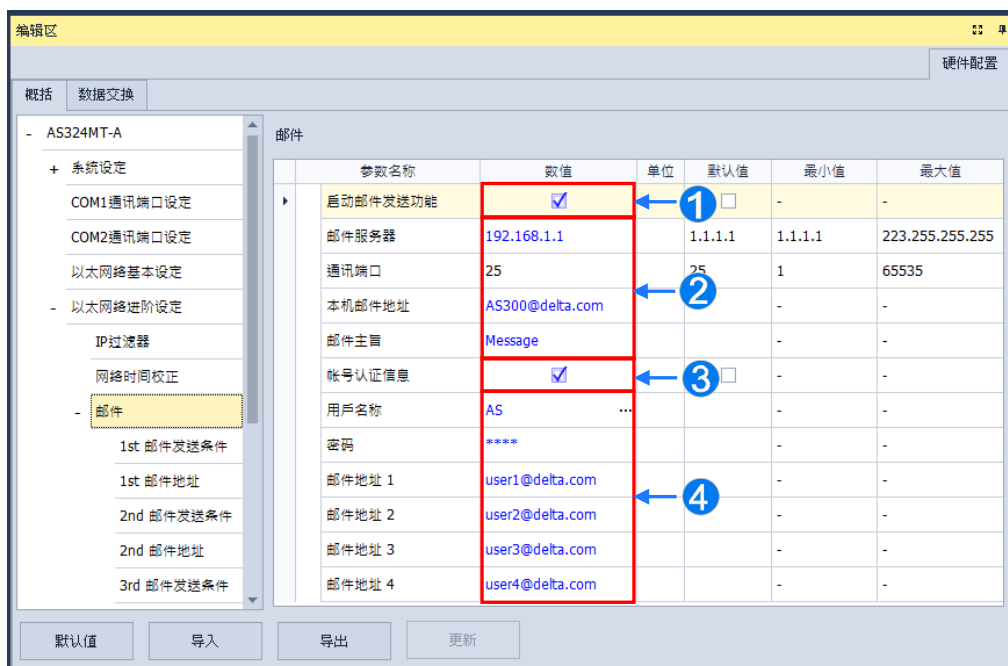
3



- ❶ 勾选**启动 NTP 客户端服务**开启功能。
- ❷ 设定 **NTP 服务器**的地址与定期更新的**更新周期**。以上图为例，主机将会每隔 30 分钟便联机至 NTP 服务器。
- ❸ 设定采用时间校正的时区。

3.1.4.11 以太网进阶设定-邮件

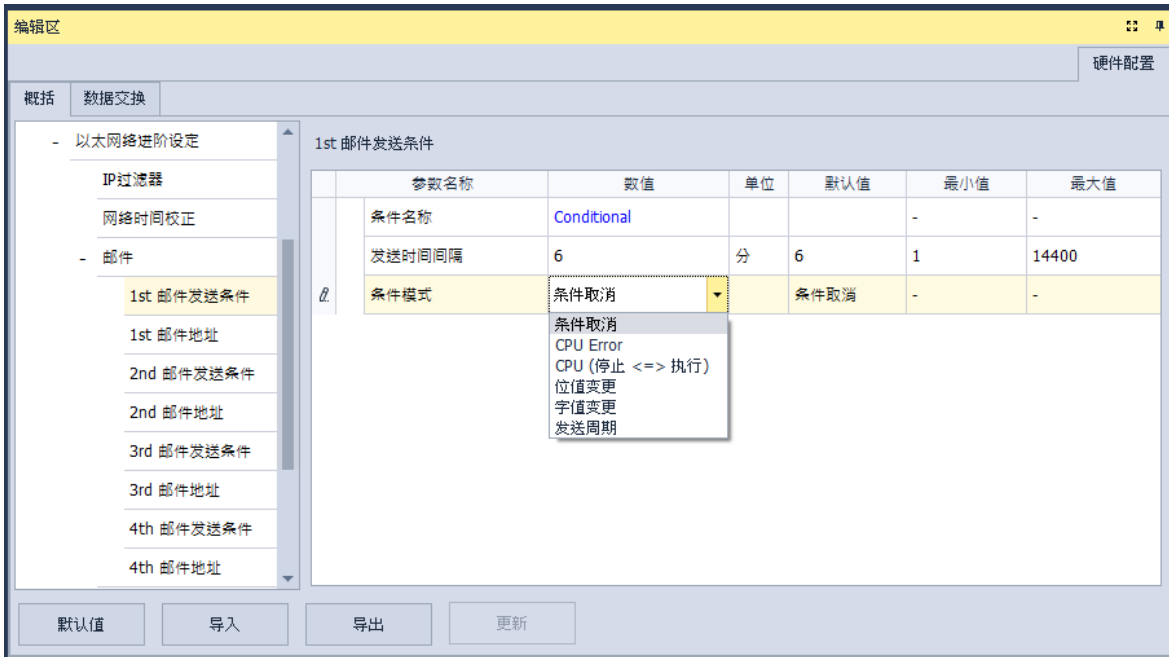
「邮件」页面提供用户设定邮件相关功能。启用当触发设定的条件模式时，将寄送邮件至设定的邮件地址的相关设定。共可设定四组邮件发送条件与四组邮件地址。



- ❶ 勾选启动邮件发送功能开启功能，便可设定下方字段。
- ❷ 由邮件服务器设定外送邮件服务器的 IP 地址，在通讯端口输入外送邮件服务器的通讯端口，在本机邮件地址设定发送电子邮件时所使用的寄件信箱，输入邮件主旨作为每封邮件的主旨开头。
- ❸ 勾选帐号认证开启功能，依据外送邮件服务器的状况，在此处设定登入前是否须先验证账号及密码。
- ❹ 输入要传送的目标邮件地址。

在「邮件发送条件」项目输入「条件名称」与「发送时间间隔」，再在「条件模式」的下拉选项选择邮件发送的触发条件。当发送条件持续成立时，系统便会每隔一段发送时间间隔就发出一封电子邮件；而在时间间隔内，针对同样的发送条件，邮件则不会再重复发送。

3



「条件模式」可设为以下

➤ CPU Error

当主机发生错误时，发送条件便会成立，而关于主机错误状态的相关说明，请参考各机种的操作手册。当选择此条件时，请再在右侧的下拉选项中挑选错误的类别。

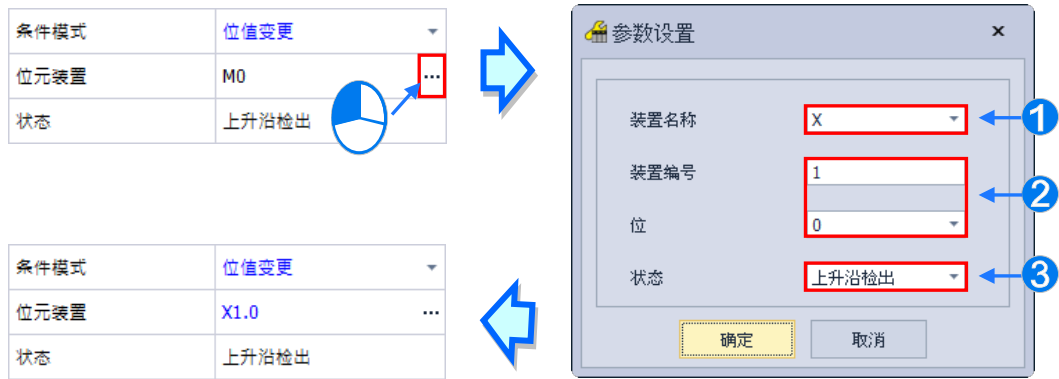
- (a) **致命错误**：当主机发生被定义为**致命错误**的状况时才会触发邮件发送。
- (b) **所有错误**：只要主机有错误产生时，邮件发送条件就会成立。

➤ CPU (停止 <=> 执行)

当主机的运行状态由 STOP 切换至 RUN，或由 RUN 切换至 STOP 时，发送条件便会成立。

➤ 位值变更

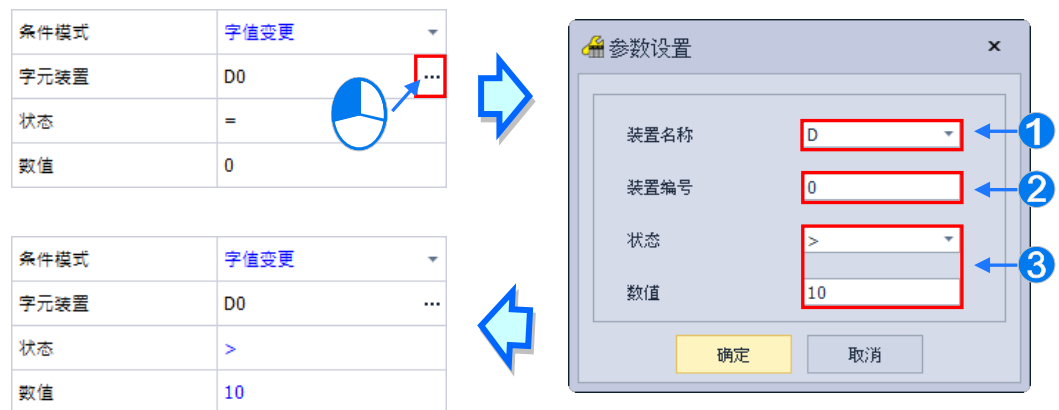
当指定装置的位状态变化符合设定条件时便会触发邮件发送。要设定条件时，单击表格右方 **...** 图标即可打开窗口进行设定。以下图为例，当 X1.0 由 OFF 变为 ON 时，邮件发送条件就会成立。



- ① 装置名称：在此下拉选择装置的类别。
- ② 装置编号与位：在此输入设备的地址，且当所选装置为 X/Y 时须一并指定位编号。
- ③ 状态：在此下拉选择触发的条件为上升沿触发或下降沿触发。

➤ 字符值变更

当指定装置的内容数值符合设定条件时便会触发邮件发送。单击表格右方 **...** 图标即可开启窗口进行设定。以下图为例，当 D0 大于 10 时，邮件发送条件就会成立。

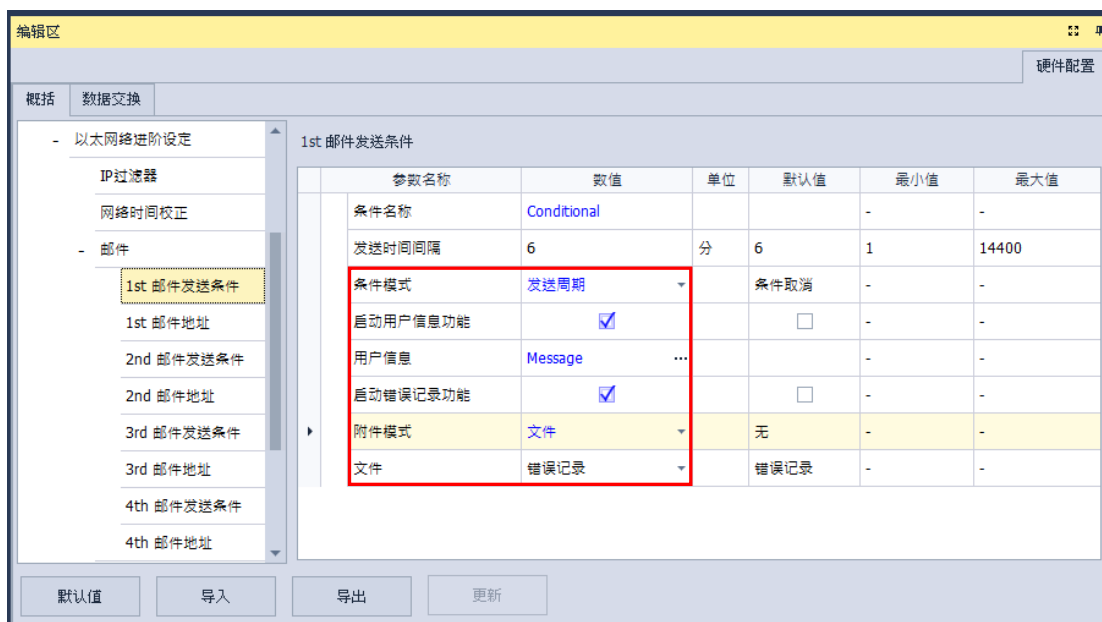


- ① 装置名称：在此下拉选择装置的类别。
- ② 装置编号：在此输入设备的地址。
- ③ 状态与数值：在此设定触发的条件。

➤ 发送周期

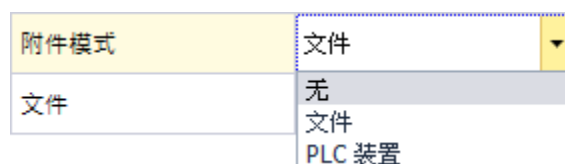
定期发出电子邮件。而发送时间则根据上方「发送时间间隔」来决定。

当有设定任何条件模式时，会出现用户信息与错误记录的相关设定。



勾选**启动用户信息功能**再单击**用户信息**右方 **...** 图标，便可在弹出窗口中输入要做为邮件本文的内容；若勾选**启动错误记录功能**，便可在邮件内容中自动添加错误记录。

接着在「**附件模式**」下拉选项选择是否要在邮件中加入附件。而要设定附件时，请务必确认各机种所允许的最大邮件文件大小，相关信息请参考各机种的操作手册。



(1) 无

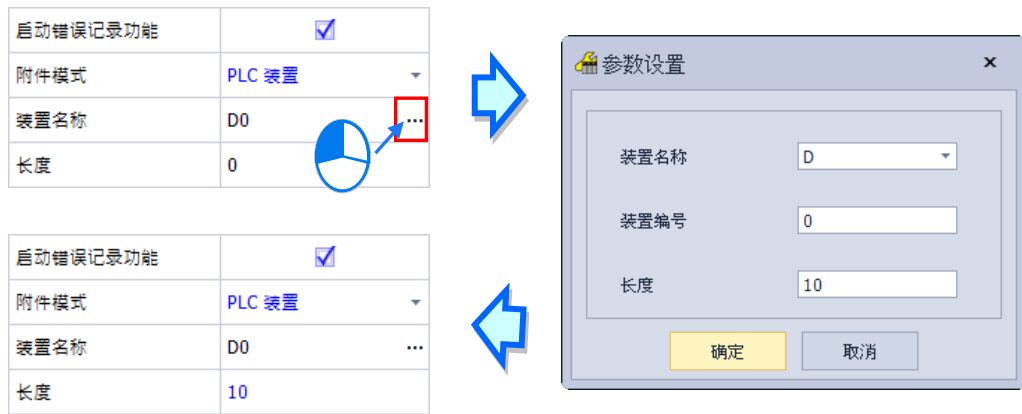
选择此项代表无需附加文件。

(2) 文件

可选择存储卡中的**错误记录**或**系统备份文件**做为邮件附件。

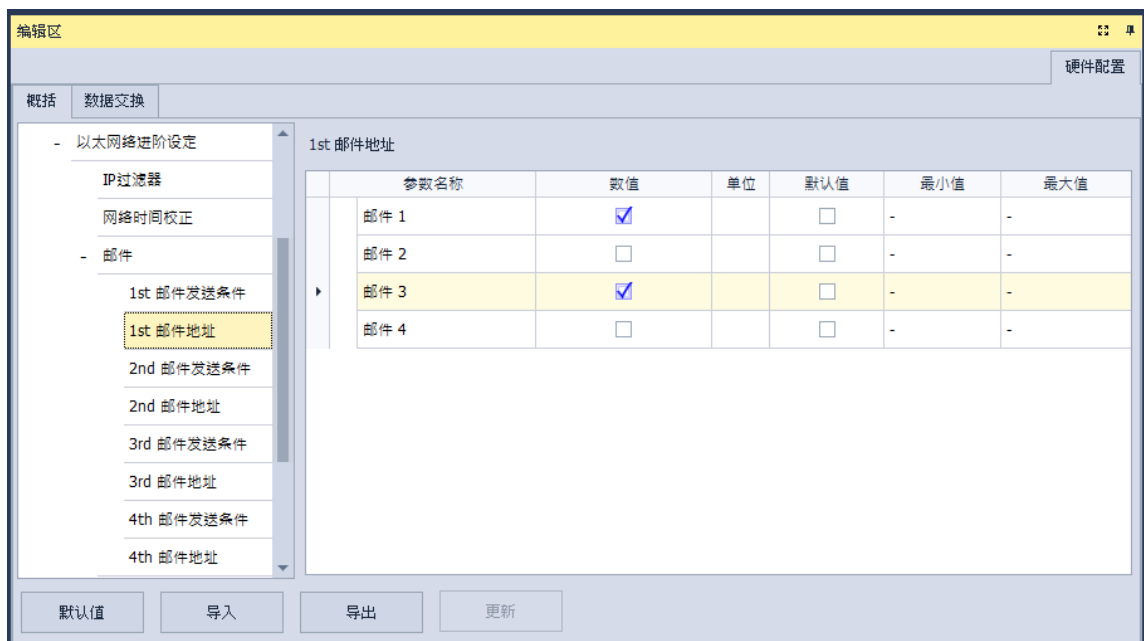
(3) PLC 装置

若选择此项，则当发送电子邮件时，系统会自动获取下方设定范围中的装置状态或数值来做为附加文件；而在选择此项之后，在下方的装置名称单击右方 **...** 图标，便会出现设定窗口。以下图为例，当此组条件成立时，装置 D0~D9 的内容值便会被当作附件而一并发送出去。



3

在**邮件地址**项目，勾选当邮件发送条件达成时要发送的目标邮件地址。邮件地址在邮件项目中设定。



3.1.4.12 以太网进阶设定-通讯接口

「通讯接口」页面中，主要在于设定通过以太网络来进行数据传递的通讯接口参数；不过此功能仍需搭配特定的 API 应用指令才可进行操作。

主机与其它主机之间，经由通讯接口(Socket)的建立来进行数据传递，且同时支持 TCP 与 UDP 两种传输协议。TCP 与 UDP 协议的参数几乎相同，差别只在于 UDP 没有「联机保持时间」的参数，参数说明如下。

3

The screenshot displays the configuration interface for the communication interface settings. It is divided into two main sections: '1st TCP 设置' and '1st UDP 设置'.

1st TCP 设置

参数名称	数值	单位	默认值	最小值	最大值
远端 IP 地址	192.168.1.1		0.0.0.0	0.0.0.0	223.255.255.255
远端通讯端口	65500		0	0	65535
本机通讯端口	65501		0	0	65535
传送数据地址	D0	...	D0	-	-
传送数据长度	10		0	0	512
接收数据地址	D100	...	D0	-	-
接收数据长度	20		0	0	512
联机保持时间	60	秒	30	0	65535

1st UDP 设置

参数名称	数值	单位	默认值	最小值	最大值
远端 IP 地址	192.168.1.1		0.0.0.0	0.0.0.0	223.255.255.255
远端通讯端口	65500		0	0	65535
本机通讯端口	65501		0	0	65535
传送数据地址	D0	...	D0	-	-
传送数据长度	10		0	0	512
接收数据地址	D100	...	D0	-	-
接收数据长度	20		0	0	512

- 远程 IP 地址：设定远程装置的 IP 地址。
- 远程通讯口：设定在这组联机中，远程装置所使用的通讯端口编号，输入范围为 0 ~ 65535。
- 本机通讯端口：设定在这组联机中，本地主机所使用的通讯端口编号，输入范围为 0 ~ 65535。

- **传送数据地址**：设定在本地主机中，存放传送数据的起始地址。
- **传送数据长度**：设定本地主机要传送的数据长度，输入范围为 0 ~ 512。
- **接收数据地址**：设定在本地主机中，存放接收数据的起始地址。
- **接收数据长度**：设定本地主机要接收的数据长度，输入范围为 0 ~ 512。
- **联机保持时间**：设定联机的最大空闲时间。当建立的联机超过此时间无数据传输时，主机便会自动中断联机。

设定时，本机与远程的通讯端口编号不可相同，且传送与接收的装置地址范围不可重迭。而以上图为例，远程装置的 IP 地址为 192.168.1.1，且开启一个编号为 65500 的通讯接口，而本地主机也开启一个编号为 65501 的通讯接口，两者之间便将通过这组联机通道来进行数据的传递。

当本地主机要传送数据给远程装置时，本地主机会先将数据存放在 D0~D9 共 10 个 WORD 的空间暂存之后，再将数据传送出去；而当接收到来自远程装置的数据时，主机则会将接收的数据存放在装置 D100 ~ D119 共 20 个 WORD 的空间中。

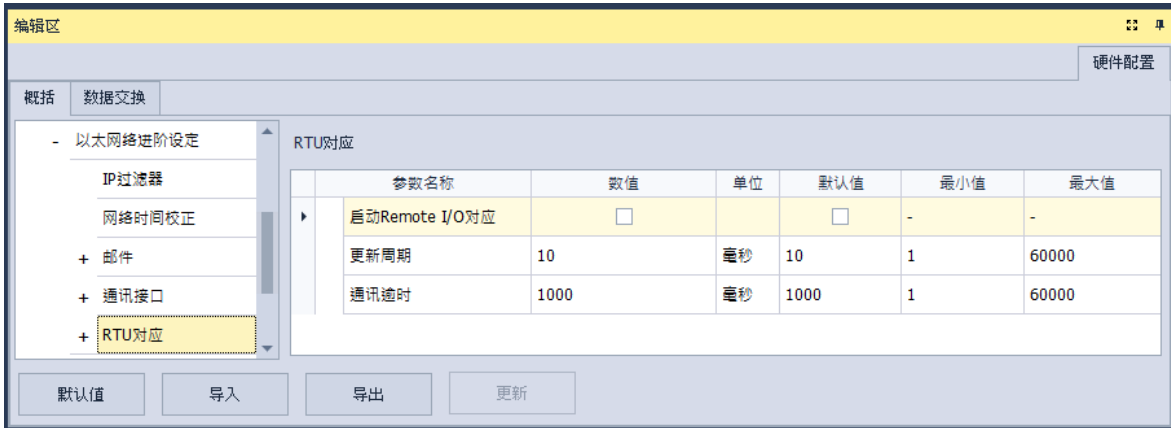
若收到的数据量大于所设定的长度时，主机会根据设定的长度，将最前面的 20 个数据存放在 D100 ~ D119 中，超出的部份则会直接将其丢弃；同样的，若收到的数据量小于设定长度时，主机则会从设定的起始位置 D100 依序存放数据，而未被存放新数据的位置则会保持原本的内容值。

当 TCP 通道联机已超过 60 秒未有任何的数据传输时，主机便会主动将主机端的通讯接口 (Socket) 关闭，并结束通道联机。

3.1.4.13 以太网进阶设定-RTU-EN01 对应

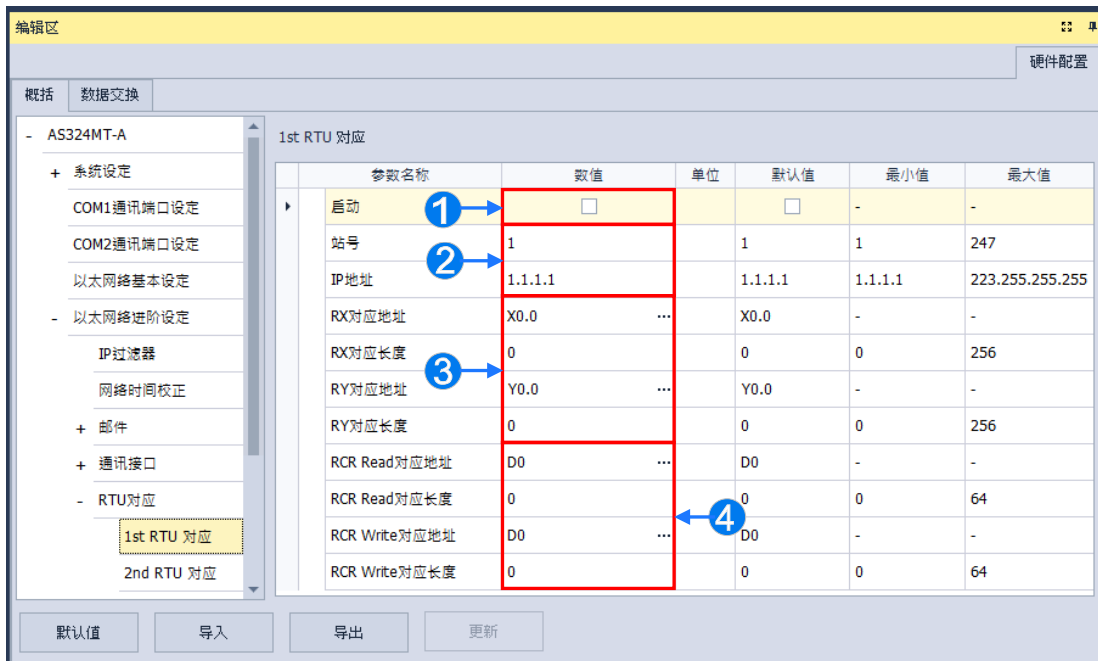
「RTU」对应页面用于设定 PLC 主机与台达 RTU-EN01 从站，可经由以太网络操作远程设备。关于 RTU-EN01 的操作与设定，请参考该产品手册。

3



勾选**启动 Remote I/O 对应**，开启 RTU 模块与 PLC 主机的联机功能，设定**更新周期**，对 RTU 依此时间周期性更新。设定**通讯超时**，当 RTU 超过此时间未响应则为超时。

RTU 对应页面用于设定每组 RTU-EN01 与 PLC 主机的数据对应。而 RTU-EN01 与其安装的 I/O 模块需通过 DCISoft 设定，关于 DCISoft 请参考其操作手册。



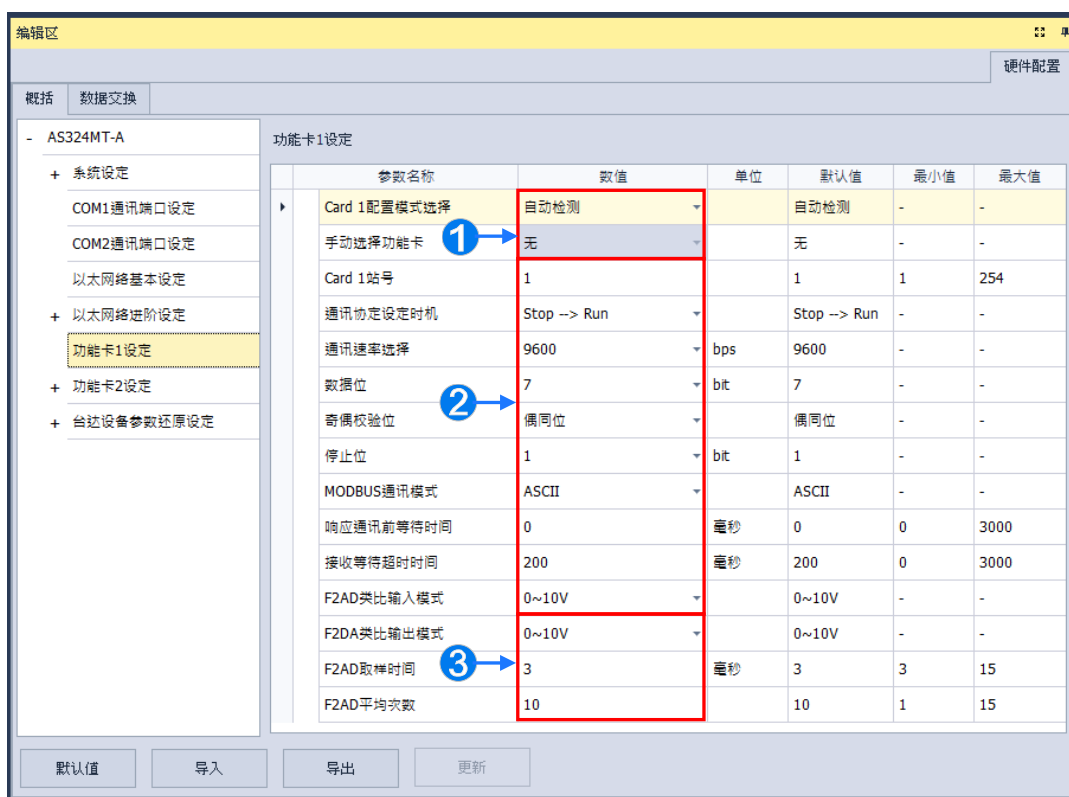
- ❶ 勾选**启动**开启此站 RTU 功能，各站可单独设定。
- ❷ 设定**站号**与 **IP 地址**。

- ③ 设定 **RX 对应地址**与 **RX 对应长度**，用于将连接在 RTU 的 DI 模块上的数字输入点，对应到 X/M 装置与长度。而设定 **RY 对应地址**与 **RY 对应长度**，用于将连接在 RTU 的 DO 模块上的数字输出点，对应到 Y/M 装置与长度。
- ④ 设定 **RCR Read 对应地址**与 **RCR Read 对应长度**，用于将连接在 RTU 的 AI 模块上的模拟输入点，对应到 D/SR 装置与长度。而设定 **RCR Write 对应地址**与 **RCR Write 对应长度**，用于将连接在 RTU 的 AO 模块上的模拟输出点，对应到 D/SR 装置与长度。

3.1.4.14 功能卡 1 设定

功能卡 1 设定页面，用于设定可安装功能卡的功能参数设定。

3

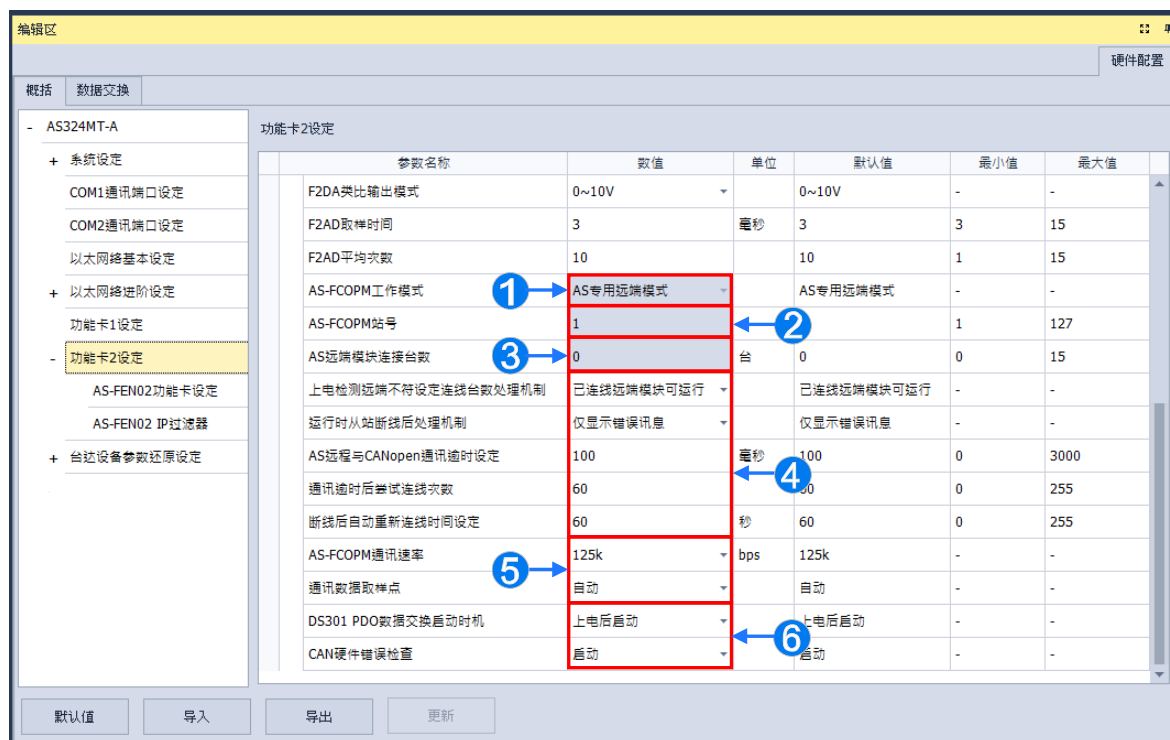


- ① **Card1 配置模式选择**字段，可选择**自动检测**检测实际卡片型号与设置，或选择**手动选择**，便可在**手动选择功能卡**字段中选择要配置的 AS--F232 通讯卡/F422 通讯卡/F485 通讯卡/F2AD 卡/F2DA 卡。
- ② 设定 AS--F232 通讯卡/F422 通讯卡/F485 通讯卡的通讯参数。
- ③ 而 **F2AD 模拟输入模式**设定 F2AD 卡接受 0~10V 或 4~20mA 信号；**F2DA 模拟输出模式**设定 F2DA 卡送出 0~10V 或 4~20mA 信号。**F2AD 取样时间**与 **F2AD 平均次数**分别用来设定 F2AD 卡取样的间隔时间与平均次数。

3.1.4.15 功能卡 2 设定

功能卡 2 设定页面除了本项目本身设定还可设定此项目内 **AS-FEN02 功能卡设定** 及 **IP 过滤器**。功能卡 2 与功能卡 1 差异为**手动选择功能卡**字段多了 **AS-FCOPM 卡**、**AS-FEN02 卡** 选项，设定为此功能卡后，便可设定以下参数。

3



- ① **AS-FCOPM 工作模式** 字段，可选择 **AS 专用远程模式/台达专属驱动器及 AS 远程模式/CANopen DS301 协议/台达专属远程与 CANopen DS301 混和模式** 共四种。**AS 专用远程模式** 为用于连接 **AS 系列机种远程 I/O**，**台达专属驱动器及 AS 远程模式** 用于对台达伺服及变频器产品的专用通讯协议以及用于连接 **AS 系列机种远程 I/O**，**CANopen DS301** 用于 **DS301 标准通讯协议** 的场合。
- ② 当 **AS-FCOPM 工作模式** 字段设为 **CANopen DS301** 时，可设定此站的站号。
- ③ 当 **AS-FCOPM 工作模式** 字段设为 **AS 专用远程模式** 或 **台达专属驱动器及 AS 远程模式** 时，用户可以在 **AS 远程模块连接台数** 输入连接的台数。
- ④ **上电检测远程不符设定联机台数处理机制** 为当设定的连接台数与实际联机检测到的数量不同时，**已联机的远程模块可运行或所有远程模块皆停止**。
运行时从站断线后处理机制 为当运行时远程 I/O 断线的处理方式为 **仅显示错误信息** 或 **停止运行并报警**。
AS 远程与 CANopen 通讯超时 设定为当远程 I/O 超过设定时间未响应时，视为超时。
通讯超时后尝试联机次数 及 **断线后自动重新连接时间** 设定为通讯重传次数与发生逾时候重新建立联机的时间。

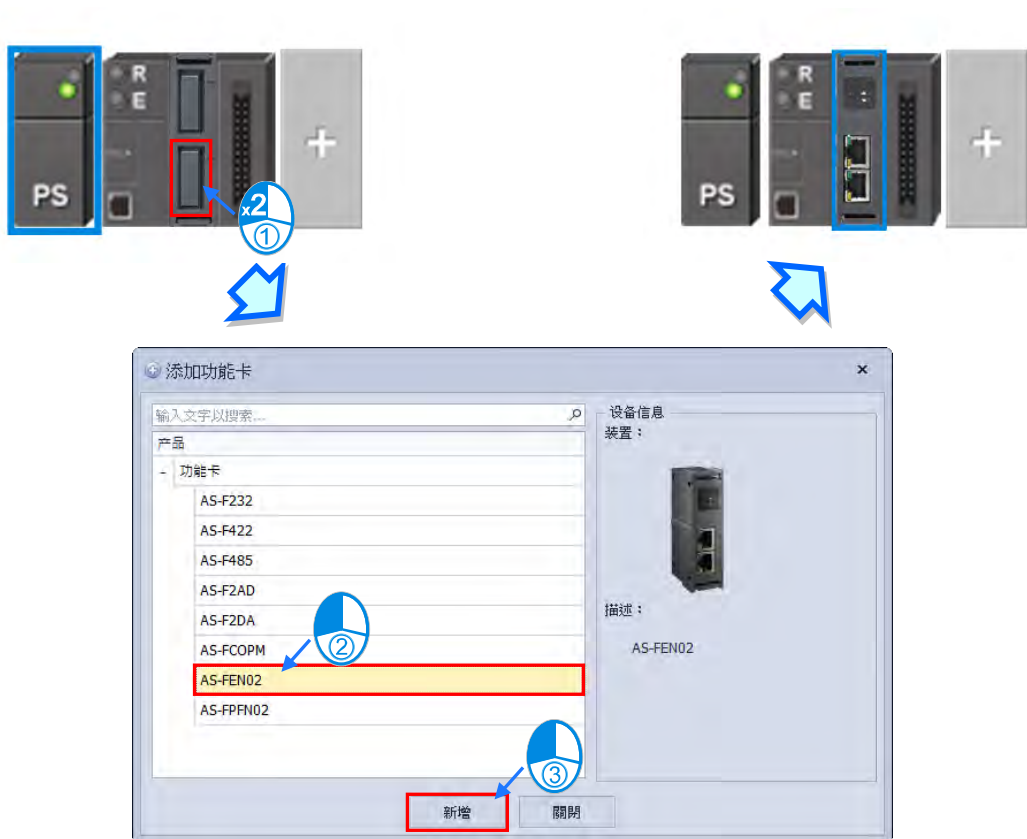
⑤ AS-FCOPM 通讯速率可设定通讯速率设定。

通讯数据取样点则可设定数据取样点的百分率。

⑥ DS301 PDO 数据交换启动时机为 CANopen DS301 协议时，功能才可生效，决定 DS301 PDO 数据交换启动时机，可分为上电后启动及 PLC RUN 启动。

CAN 硬件检查错误可以设定启动或关闭。

而 AS-FEN02 功能卡设定，则可以在硬件配置区的功能卡插槽，双击鼠标左键接着选取 AS-FEN02 功能卡这样就配置完成了，如下图。



3

也可在下方编辑区的功能卡 2 设定分页，在 **Card 2 配置模式**选择手动选择并将手动选择功能卡设定为 AS-FEN02 卡后，则功能卡 1 设定，将无法做设定，接着便可设定「AS-FEN02 功能卡设定」之以太网 (Ethernet) 端口的通讯参数，如下图。

3

編輯區

硬體配置

AS324MT-A

系統設定

COM1通訊埠設定

COM2通訊埠設定

乙太網路基本設定

乙太網路進階設定

功能卡1設定

功能卡2設定

AS-FEN02功能卡設定

AS-FEN02 IP過濾器

台達設備參數還原設定

功能卡2設定

名稱	數值	單位	預設值	最小值	最大值
Card 2配置模式選擇	手動選擇		自動偵測	-	-
手動選擇功能卡	AS-FEN02卡		無	-	-
Card 2站號	1		1	1	254
通訊協定設定時機	Stop --> Run		Stop -->...	-	-
通訊速率選擇	9600	bps	9600	-	-
資料位元	7	bit	7	-	-
同位元	偶同位		偶同位	-	-
停止位元	1	bit	1	-	-
MODBUS通訊模式	ASCII		ASCII	-	-

預設 匯入 匯出 更新



編輯區

硬體配置

AS324MT-A

系統設定

COM1通訊埠設定

COM2通訊埠設定

乙太網路基本設定

乙太網路進階設定

功能卡1設定

功能卡2設定

AS-FEN02功能卡設定

AS-FEN02 IP過濾器

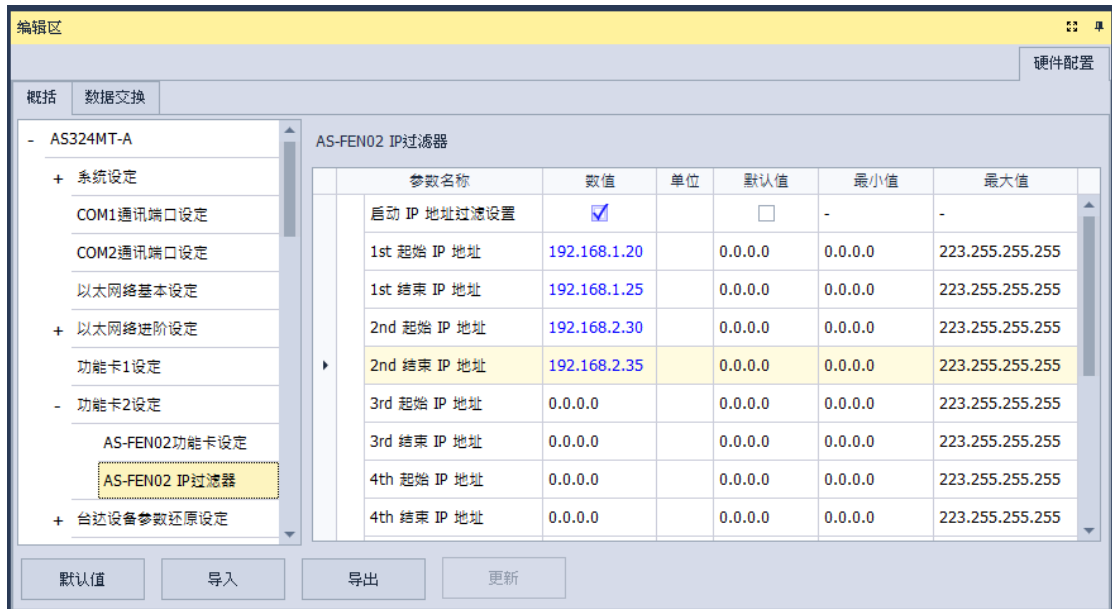
台達設備參數還原設定

AS-FEN02功能卡設定

名稱	數值	單位	預設值	最小值	最大值
IP位址	192.168.1.5		192.168.1.5	1.1.1.1	223.255.255.255
子網路遮罩	255.255.255.0		255.255.2...	0.0.0.0	255.255.255.255
閘道位址	192.168.1.1		192.168.1.1	1.1.1.1	223.255.255.255
TCP連線保持時間	30	秒	30	1	65535
IP定址模式	靜態		靜態	-	-

預設 匯入 匯出 更新

IP 过滤器是当 Card 2 使用 AS-FEN02 卡后，也可设定「IP 过滤器」页面，此页面为设定网络装置的过滤功能，只有 IP 地址列于列表范围中的装置才允许跟 AS-FEN02 卡进行通讯，此功能最多可设定 8 组的地址范围，如下图。



要启动 IP 过滤器功能，请勾选启动 IP 地址过滤设置，便可输入以下 1~8 组的 IP 过滤范围的起始与结束地址。

3.1.4.16 台达设备参数还原功能

*操作软件：必须使用 ISPSOft V3.10 (含) 以上，搭配 EIP Builder V1.70 (含) 以上。

*PLC 主机：AS 主机固件必须更新到 V1.08.00 (含) 以上。

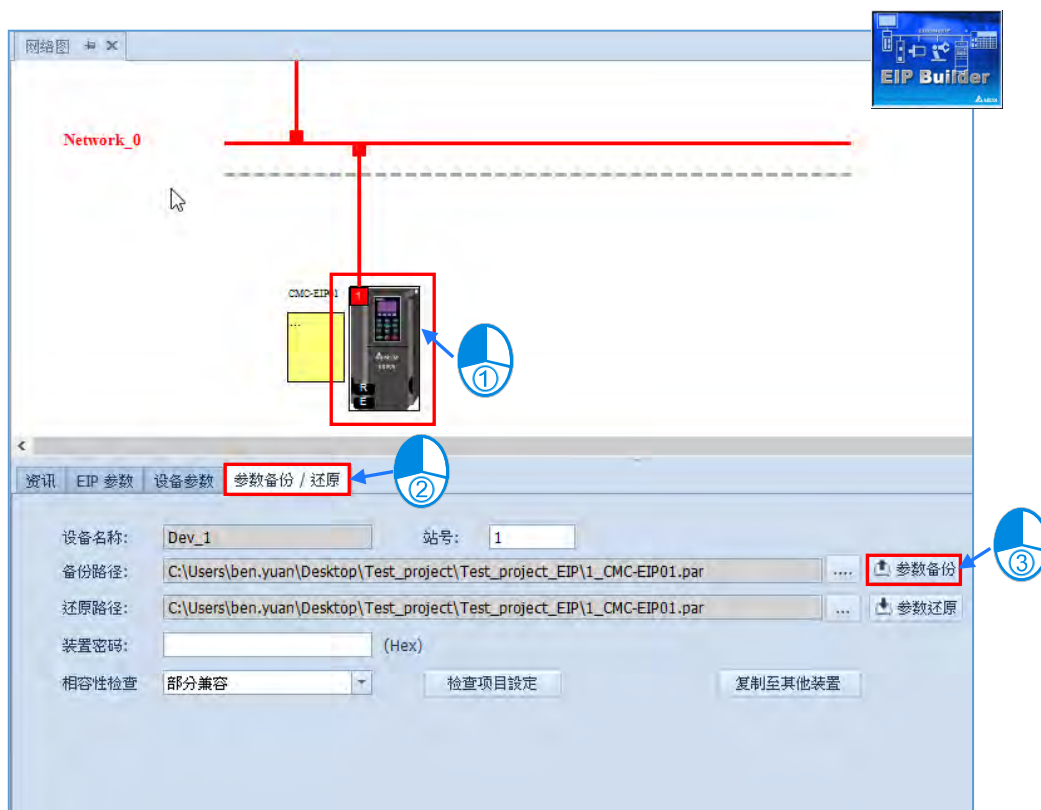
*驱动器：目前只支持还原 Delta C2000 变频器参数版本: V2.06 (含) 以上。

● 目的

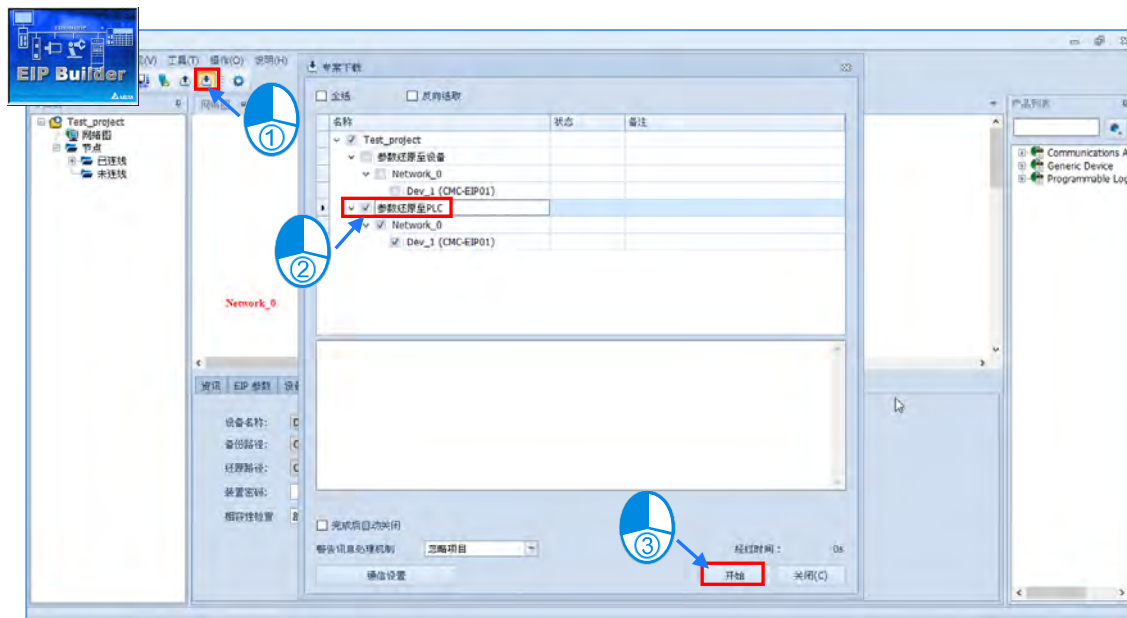
可使用 EIP Builder 软件进行备份参数动作并将备份结果下载到 AS 主机的 SD 卡上 (需插入 SD 卡)，如果已经做好备份的则可以略过「将设备参数备份到 PLC 的 SD 卡上」的步骤，直接跳到「主机还原的相关设定」步骤，接着设定「台达设备参数还原设定」并下载到 AS 主机，就可以通过设定 PLC 的 SR 装置，让 AS 主机依需求进行台达设备的参数还原动作，目前 AS 主机最高可以支持 32 台台达设备，关于详细的使用方式请参考 AS 系列操作手册。

● 将设备参数备份到 PLC 的 SD 卡上

请参考第 3.1.3.16 节开启 EIP Builder 并且完成项目的网络规画后，先单击要备份的 C2000 变频器「CMC-EIP01」设备，接着单击「参数备份 / 还原」分页，再单击「参数备份」直接通过网络对台达驱动器进行参数备份 (保存在计算机 扩展名.par) 如下图，多台设备时可使用不同的站号，文件命名会自动随着站号而改变，以区分各站号的参数备份文件，亦可直接单击「参数还原」通过网络对台达驱动器进行参数还原。



接着先确认 SD Card 已经放进 AS 主机内，并且确认 PLC 处于 STOP 状况，就可以执行下载动作，并且选择参数还原至 PLC，再单击开始按钮后就会进行选定设备的参数备份动作，完成后就表示数据已经存放到 PLC 的 SD 卡内部了。



3

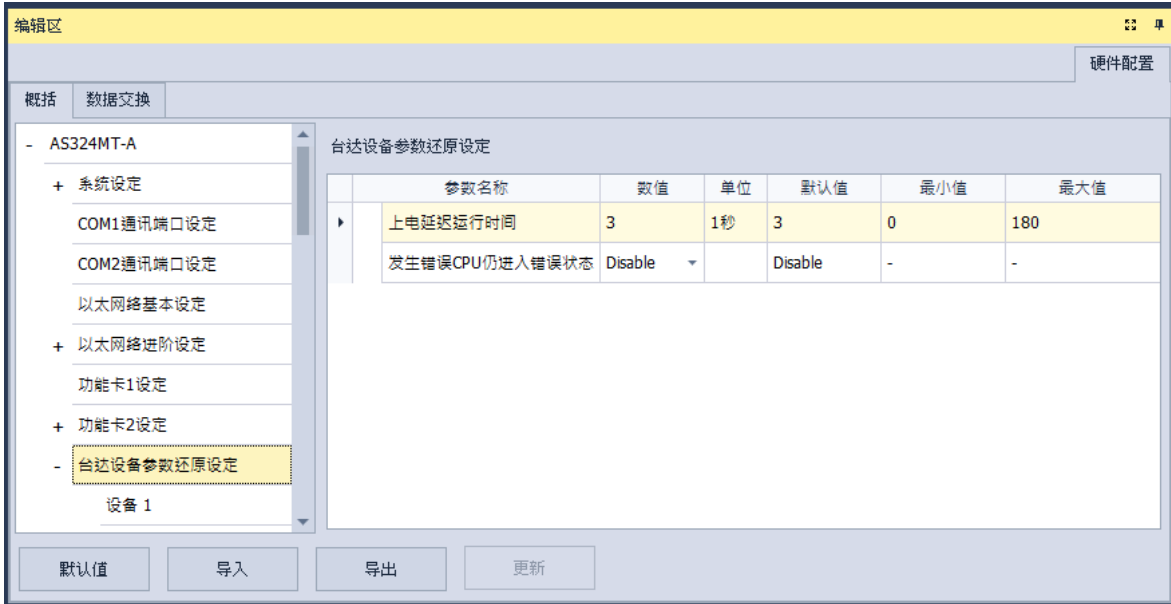
补充说明

- ◆ 默认保存路径
 - AS300: \SDCard\PLC CARD\AS300\DevPara
 - AS200: \SDCard\PLC CARD\AS200\DevPara
 - ◆ 文件名格式：mmmm_nn_tt_pppppppppppppppp.par
 - mmmm: 设备或模块代码 (PLC 主机: 0000)
 - nn: 设备套接字编号 (PLC Ethernet 通讯口编号: 03)
 - tt: 通讯协议类型 (固定代码: 01)
 - pppppppppppppppp: 驱动器设备信息 (IP + ID + 补 0)
- Ex. IP: 192.168.1.9, ID: 9 · 文件名为: C0A80109 (IP) + 09 (ID) + 000000 (补 0) = C0A8010909000000.par

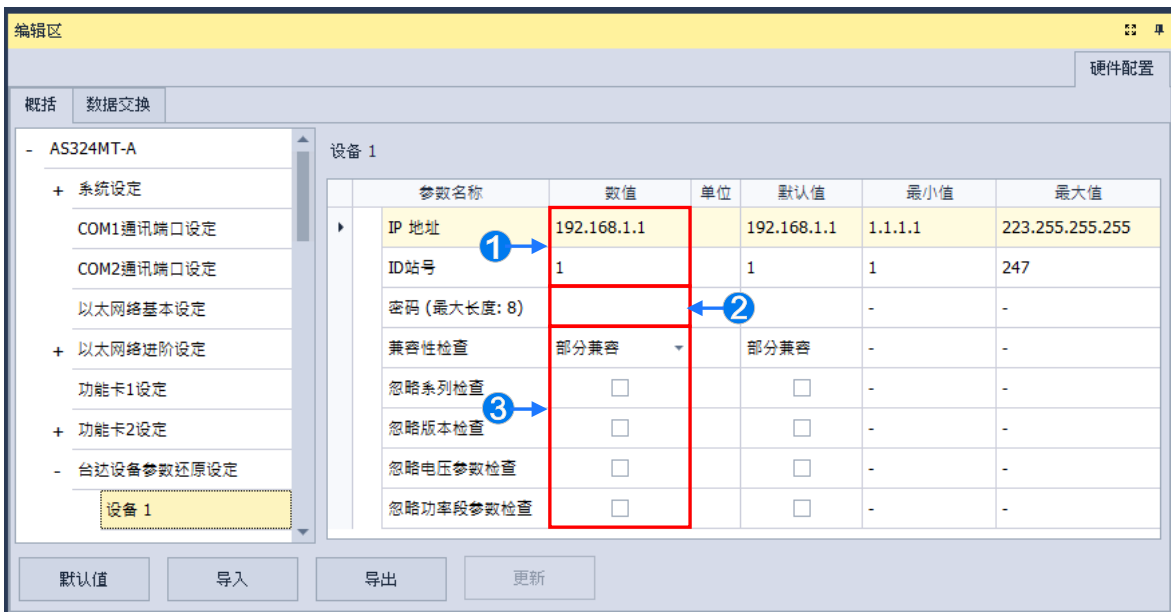
● 主机还原的相关设定

在主机 HWCONFIG 的「台达设备参数还原设定」的页面可以设定上电延时执行的时间默认为 3 秒钟，也可以设定发生错误时 CPU 是否要进入 STOP 状态。

3



在「设备 1~32」的页面可以设定每一台台达设备的连接信息与还原条件，如下图。



- ❶ 输入目标设备的 IP 地址与 ID 站号。
- ❷ 输入目标设备的密码，最大长度为 8 个字符，格式可选择。
- ❸ 兼容性检查字段是设定由目标设备决定检查条件是否可以忽略，设为部分兼容的时候可以勾选下面四个还原条件。

尝试忽略系列检查：设备的系列不相同的时候依然可以**尝试**还原。

尝试忽略版本检查：设备的版本不相同的时候依然可以**尝试**还原。

尝试忽略电压参数检查：设备的电压参数不相同的时候依然可以**尝试**还原。

尝试忽略功率段参数检查：设备的功率段参数不相同的时候依然可以**尝试**还原。

当**兼容性检查**字段设为**完全兼容**的时候则必须所有条件兼容时才执行还原。

当**兼容性检查**字段设为**强制写入**的时候则会**尝试**直接强制执行还原。

若目标设备不支持**兼容性检查**的条件，则即使设定了忽略条件，还是有可能无法还原。

关于目标设备支持那些条件，则需要参考所连接的目标设备使用手册。

接着必须参照下表的 **SR 装置说明**，依据需求去设定对应的数值到 **SR 装置** (停电保持)，可以使用 **PLC 编辑软件**、**HMI** 或是 **SCADA** 等方式来进行设定，完成后下次主机重新启动时就会依设定进行对应的参数还原动作。

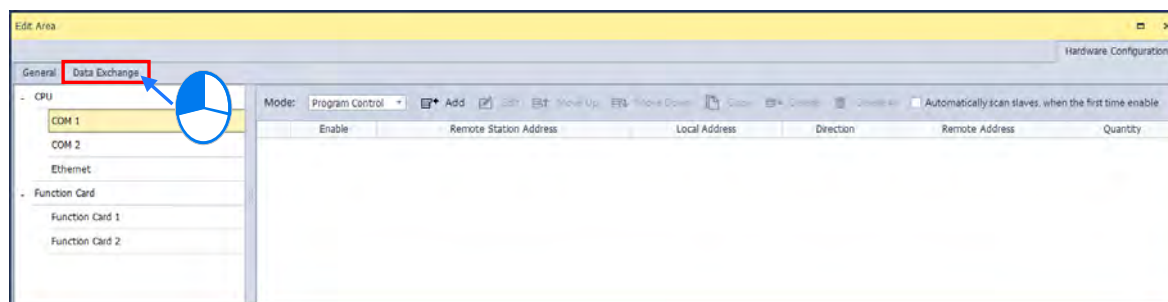
3

SM / SR 编号	功能名称	说明
SR1600	参数还原执行代码	还原动作代码如下数值与说明： 0：不执行。(默认值) 1：当 PLC 上电或 Stop→Run 时，执行参数还原动作，直到执行完成时才清除为 0
SR1601.0~SR1601.15 ⋮ SR1602.0~SR1602.15	指定参数还原的目标驱动器编号	SR1601.0 对应到目标驱动器编号 1 SR1601.1 对应到目标驱动器编号 2 ⋮ SR1602.15 对应到目标驱动器编号 32 Off：不执行。(默认值) On：被指定执行参数还原之动作

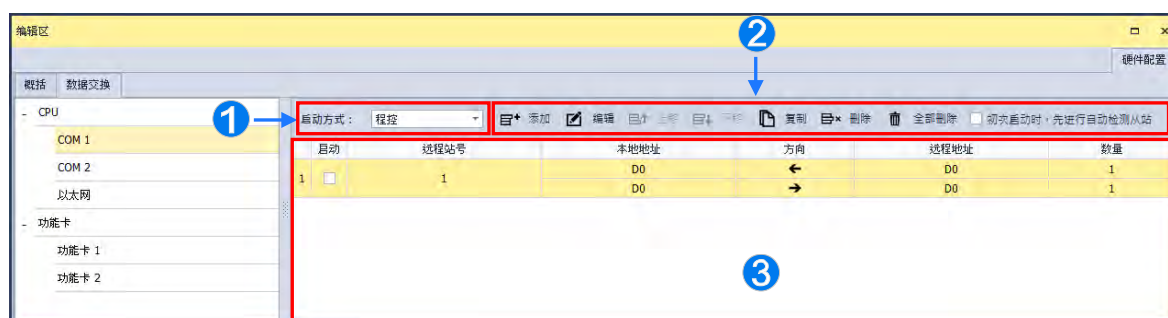
3.1.4.17 数据交换

CPU 模块本身内建了基于标准 MODBUS TCP 的数据交换功能，此功能位于「编辑区」的「数据交换」分面中，功能的设定方式如下。

3



CPU 依据机种型号，有提供两个 COM 通讯口，或是一个 COM 端口与一个以太网网络接口的机种，这些参数分别位于「数据交换-COM1」与「数据交换-COM2」或「数据交换-COM」与「数据交换-Ethernet」分页。主站，对于作为从站的远程设备进行数据交换，COM1 与 COM2 通讯口通过 MODBUS 协议；而以太网网络端口口则通过 MODBUS TCP 协议。设定字段三分页皆相同，如下图。








❶ 设定启动方式为





- A. 程控时：由 PLC 的程控是否执行设定的数据交换的标志；其各通讯口的控制标志，请参考程序篇手册的特殊标志 SM 说明。
- B. PLC 执行：当 PLC 为 RUN 的状态时，会自动执行设定的数据交换；为 STOP 时会停止通讯。
- C. 永远执行：当 PLC 上电时，便会不断执行数据交换。

此工具栏功能介绍如下：

图示	说明
	会在区域❸的数据交换表区域新增一列新的数据交换表。
	可对目前选择的数据交换进入数据交换设定页面。

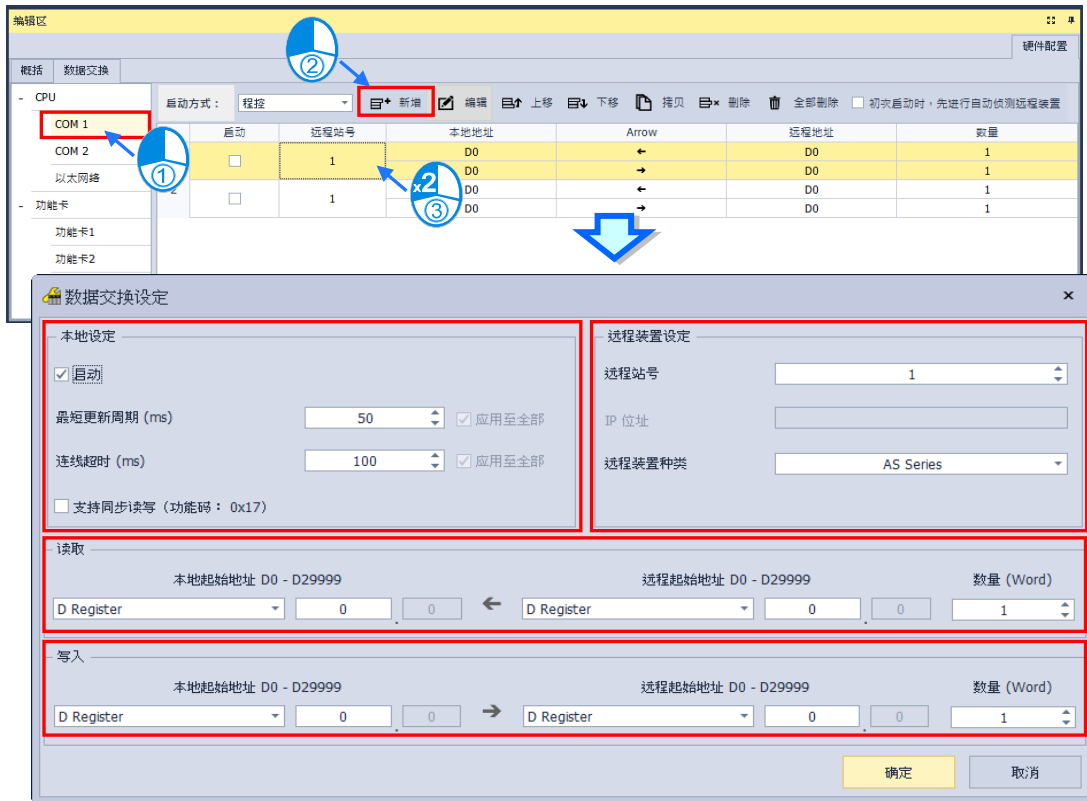
图示	说明
 上移  下移	以对目前选择的数据交换表往上一列或下一列搬移。
 拷贝	可将目前选择的数据交换表复制后贴到最下方一列。
 删除	可删除目前选择的数据交换表。
 全部删除	可删除目前所有的数据交换表。
<input checked="" type="checkbox"/> 初次启动时，先进行自动侦测远程装置	<p>当初次启动数据交换前，将会检测数据交换表已指定启动的从站是否存在？若未检测到从站，则会认定该从站不需要继续进行通讯联机，以节省通讯效率。</p> <p>建议：此功能使用在有确定从站都会比主站快上电等待通讯之应用，若无法保证时，请勿勾选此功能。</p>

③ 列出建立的数据交换表。说明如下表。

表格字段	说明
启动	表示进行数据交换时，是否启动此组数据交换表。
远程站号	注明此组数据交换表所属的从站地址，同一个地址可规划为多组数据交换表。 COM1 与 COM2 分页注明为站号，以太网网络分页则注明为 IP。
本地地址	注明主站在此组数据交换表中所使用的装置地址范围。
 	<p>：输入，表示此列所显示的是主站读取从站的数据区块</p> <p>：输出，表示此列所显示的是主站写入从站的数据区块。</p>
远程地址	注明从站在此组数据交换表中所使用的装置地址范围。而当此组数据交换表的从站为自行定义的「MODBUS Device」装置时，此范围会以 16 进制的地址编号来显示。
数量	注明此组交换表的长度，基本上与装置范围所计算的结果一致。

数据交换设定页面

以双击欲设定的数据交换表，开启数据交换表的设定项目，如下图。



● 「本地设定」的数据区块

勾选**启动**代表要启动此组数据交换表，并且随着前述的**启动方式**启动。**最短更新周期 (ms)**为设定此组数据交换表进行数据交换的时间周期。**连线超时 (ms)**为设定若目标设备超过此设定时间未响应，则为超时。勾选**支持同步读写 (功能码 0x17)**将利用特定的 MODBUS 功能码，主站便可在一次的命令当中完成**读与写**的动作，以提高数据交换的效率；不过设定时，请确认参与数据交换的所有装置都有支持同步读写的 MODBUS 功能码，否则便可能在接收到主站的命令后，会因无法识别该功能码而造成读写失败。

● 「远程设定」的数据区块

远程站号用于进行数据交换的目标从站设备站号。

IP 地址为目标从站设备 IP 地址 (只支持以太网网络)。

远程装置种类为目标从站设备的机种，包括台达 PLC 与标准 MODBUS 设备。

● 「读取」的数据区块

设定当 PLC 主机自远程设备读取数据时，PLC 主机用以存放数据的**装置类别**、**装置起始地址**，接着指定要读取的远程设备**装置类别**及**起始位置**。设定输入的**数据长度**。

● 「写入」的数据区块

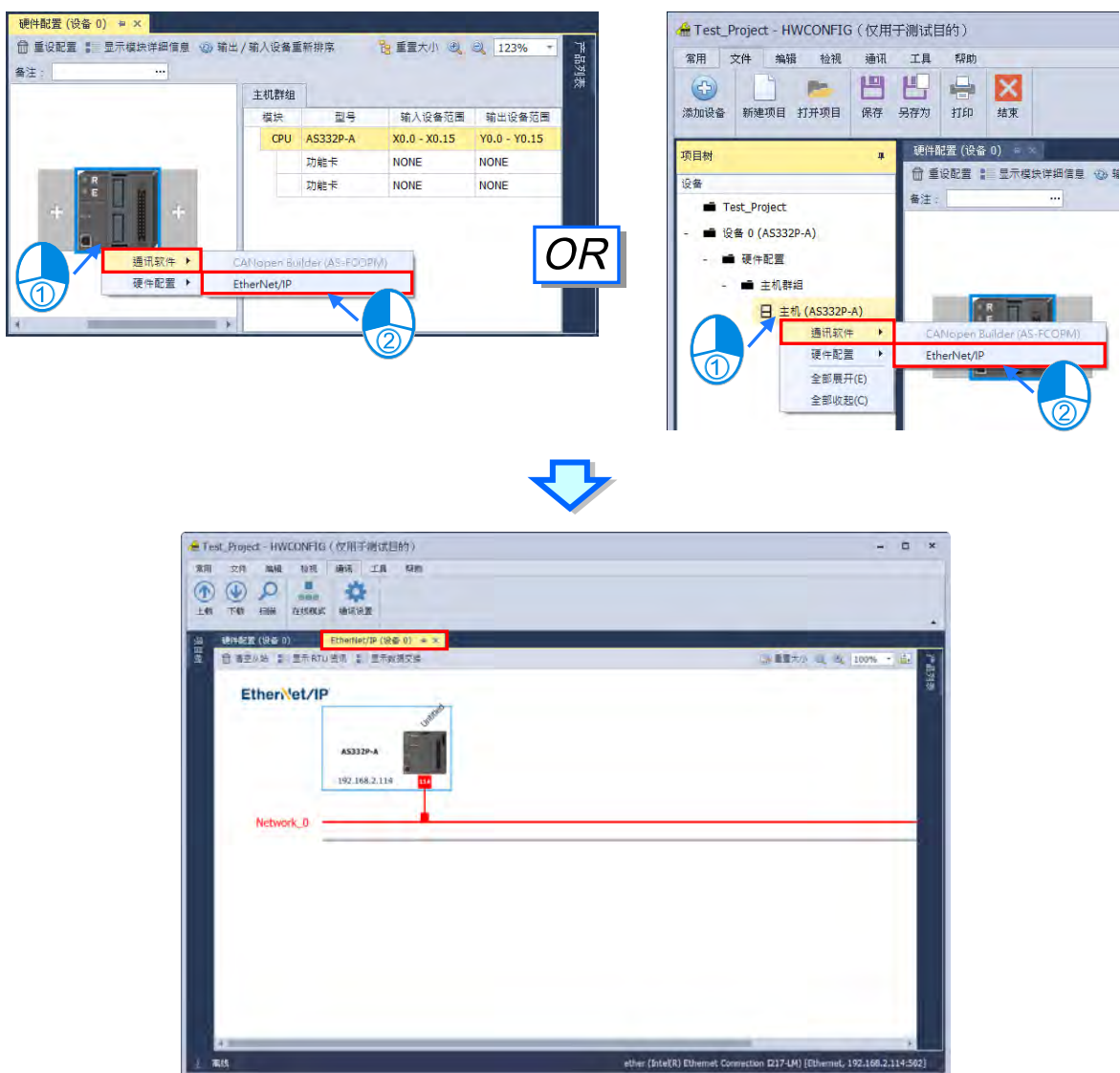
设定当 PLC 主机写入数据至远程设备时 PLC 主机源数据的**装置类别**、**起始地址**，接着指定被写入数据的远程设备**装置类别**及**起始位置**。设定输出的**数据长度**。

3.1.5 EtherNet/IP

EtherNet/IP 为由 ODVA 协会 (Open DeviceNet Vendors Association) 管理的工业以太网通讯协议，其名称中 IP 是工业协议 (Industrial Protocol) 的缩写。EtherNet/IP 架构于 TCP/IP 通讯协议上，可兼容于一般 IT 网络，提供工厂自动化 (FA)、楼宇自动化 (BA) 和程序自动化 (PA) 等应用高速稳定的应用。台达 EtherNet/IP 产品涵盖控制类与驱动类产品，包括可程序逻辑控制器、变频器、人机、伺服和交换器，支持产品一览表请参考产品手册第 9 章。此外，可通过 EDS 文件与其它厂牌 EtherNet/IP 设备连接。

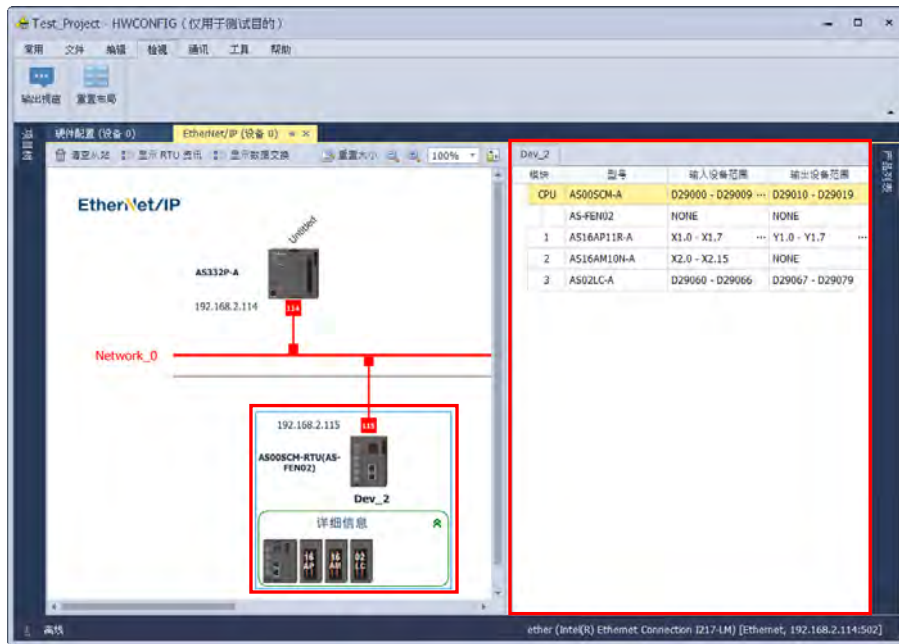
3.1.5.1 开启 EIP

在硬件配置区的主机或项目树的主机鼠标右击后，于快捷选项单击「通讯软件」>「EtherNet/IP」，即可开启 EIP 设定页面。



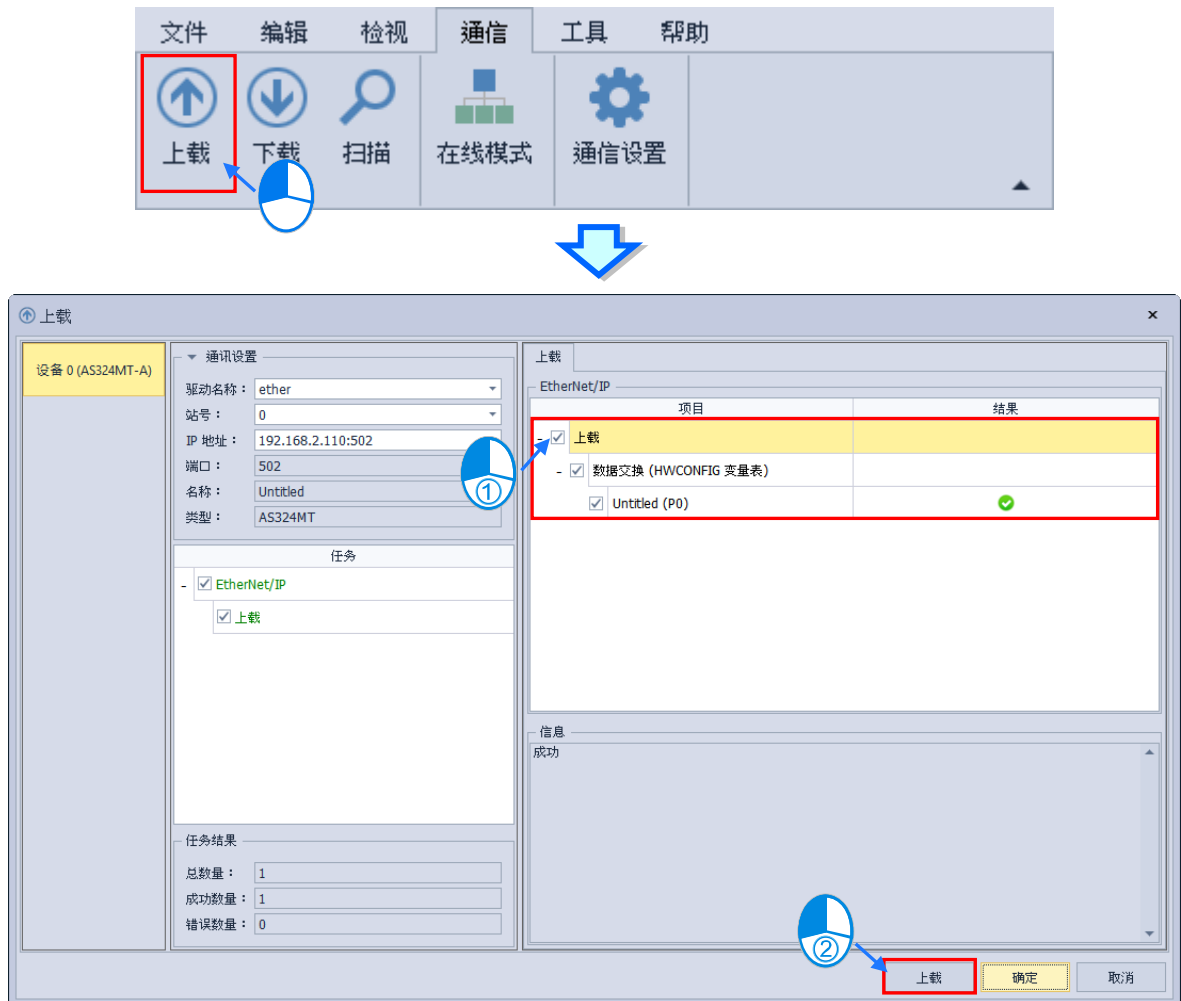
EIP 页面左侧为 EIP 配置图，若单击 RTU 从站，右侧会显示 RTU 内部的详细模块配置，如下图。

3



3.1.5.2 上载 EtherNet/IP

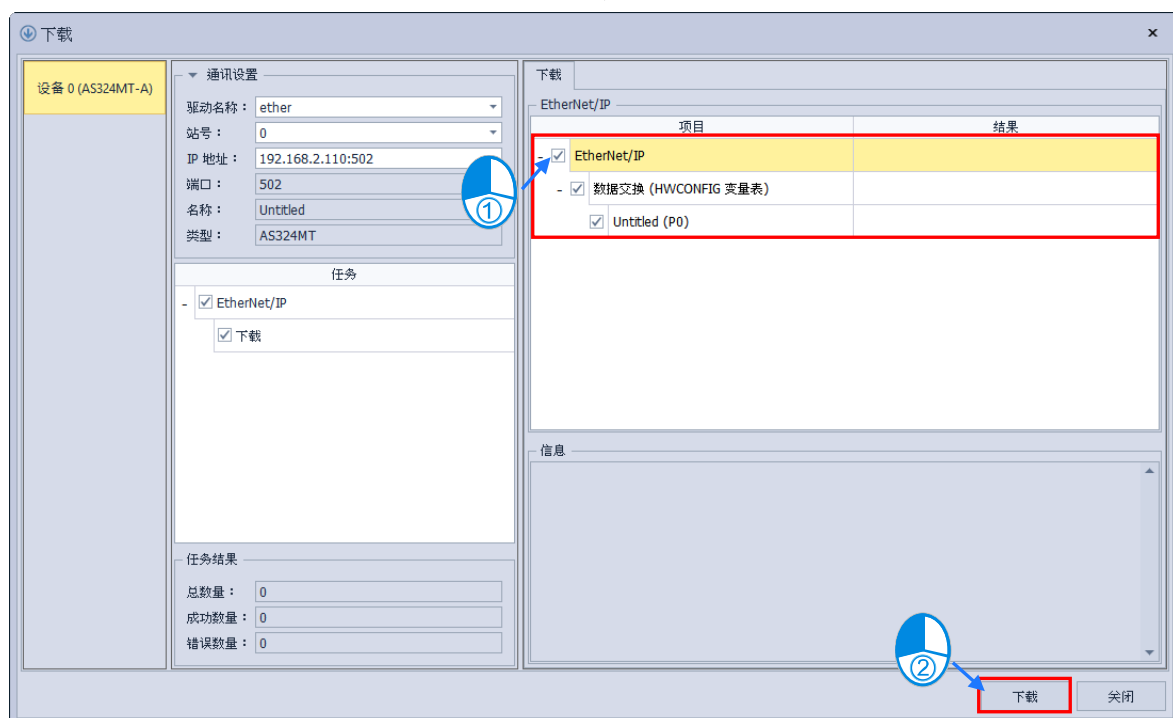
开启 EIP 分页，单击菜单工具栏「通讯」>「上载」后，将会跳出上载窗口，单击「上载」按钮后，等待信息窗口出现成功后，单击「确定」即可更新软件 EIP 设定，若单击「取消」则不会更新软件 EIP 设定，如下图所示。



3.1.5.3 下载 EtherNet/IP

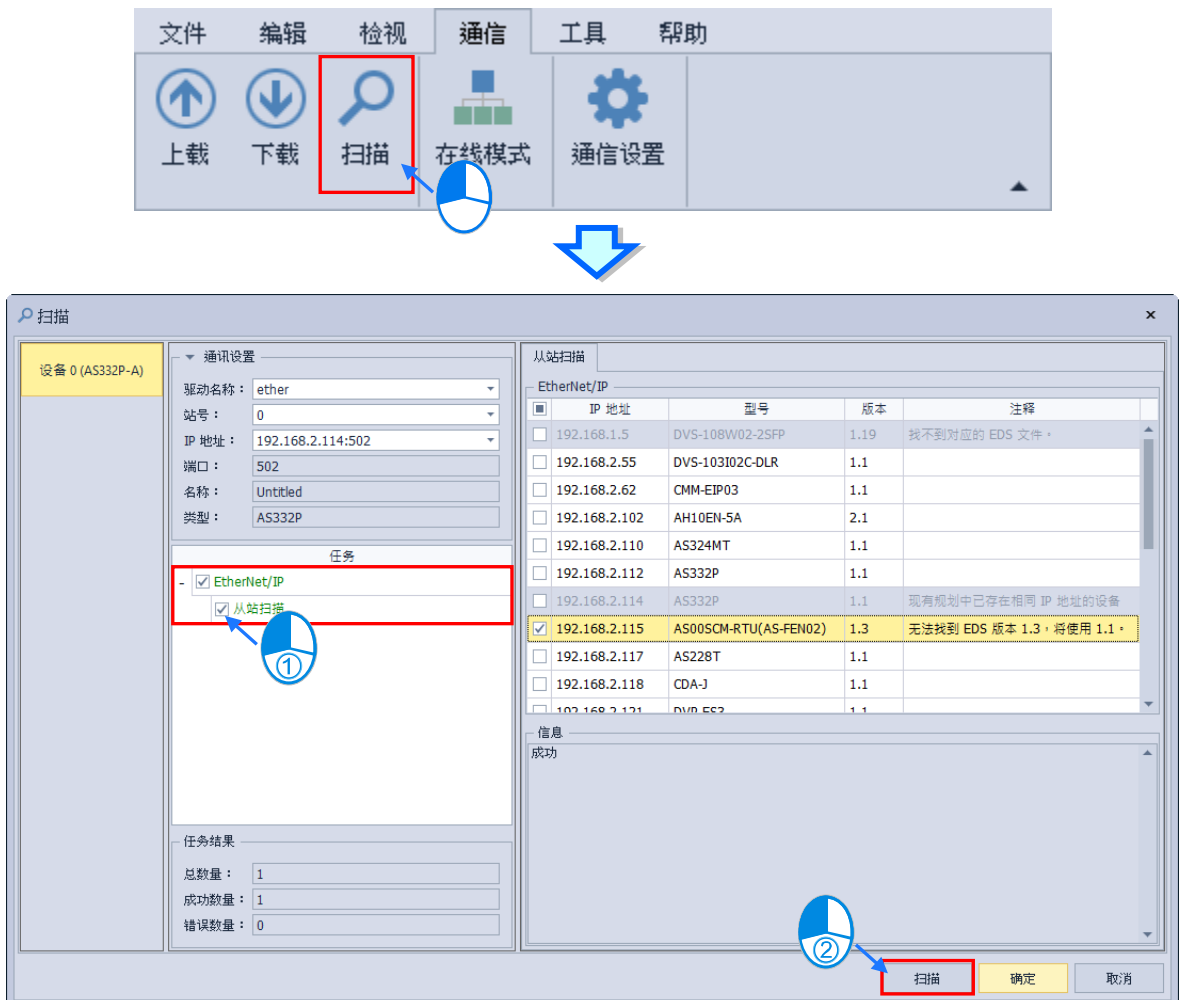
开启 EIP 分页，单击菜单工具栏「通信」>「下载」后，将会跳出下载窗口，确认下载选项后，单击「下载」，确认完成后即完成下载，如下图所示。

3



3.1.5.4 扫描从站

开启 EIP 分页，单击菜单工具栏「通讯」>「扫描」后，将会跳出扫描窗口，确认从站扫描勾选后，单击「扫描」后在窗口右侧即会出现可通讯的从站，如下图所示。

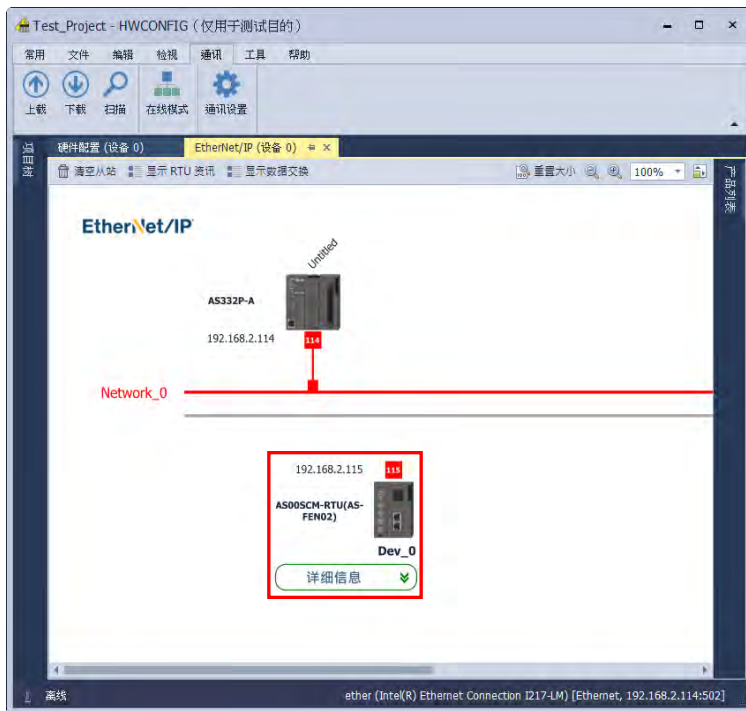
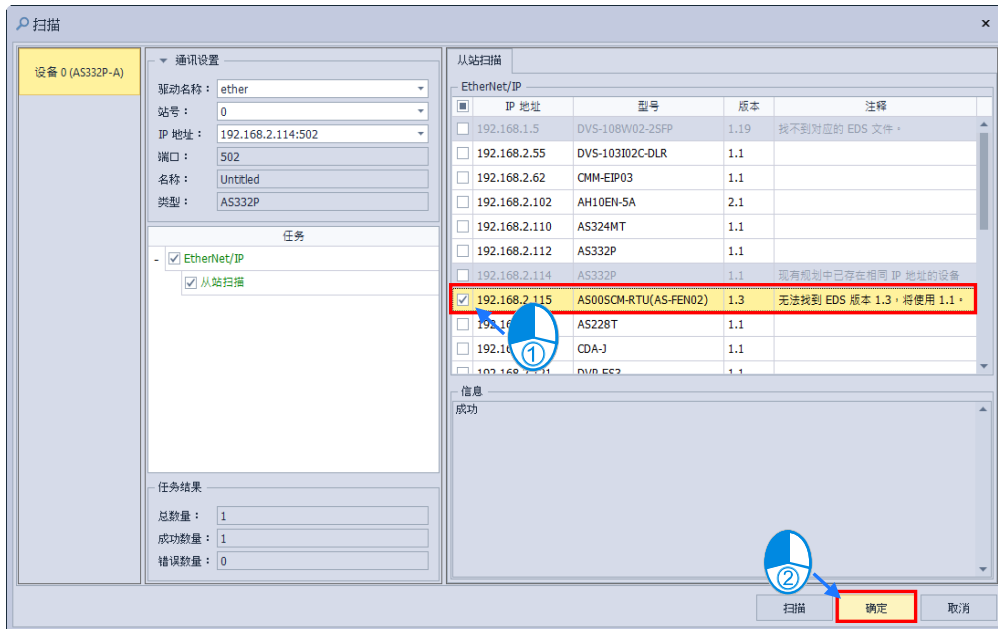


3.1.5.5 加入从站

➤ 方法一

从扫描完成后的窗口勾选要加入的从站后，单击「确定」即可将从站加入画面中。

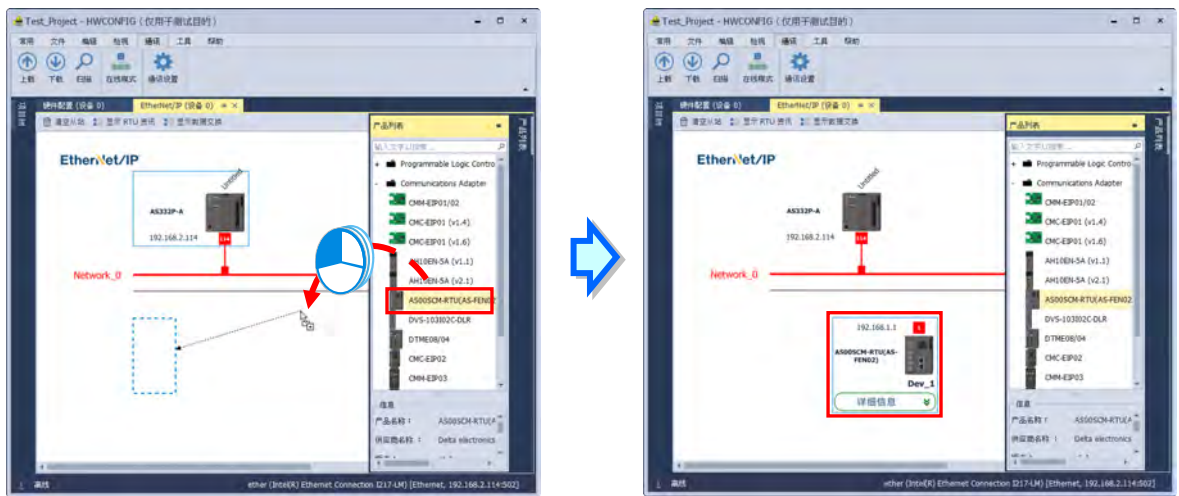
3



*扫描加入的从站会将从站信息以及 RTU 后端模块带入项目中。

➤ 方法二

从产品列表拖曳欲加入的从站至画面中



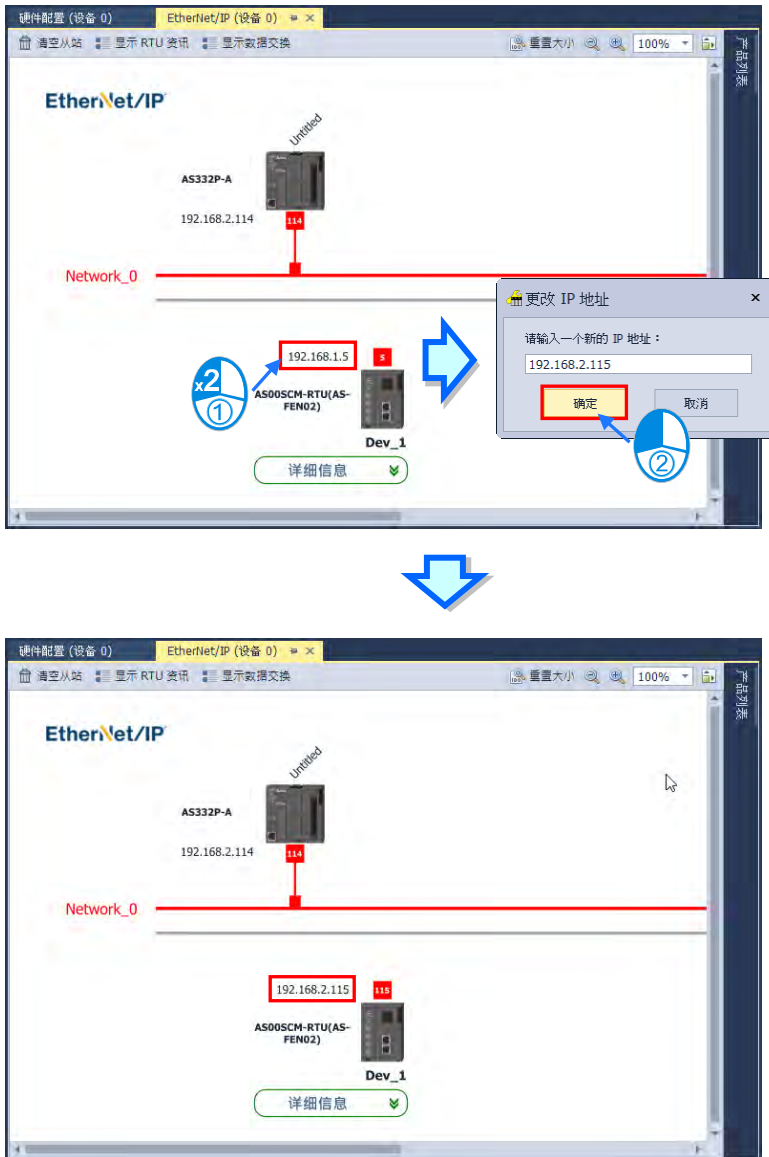
3.1.5.6 设定主站\从站 IP

➤ 设定主站 IP

主站 IP 是从主机的以太网络的 IP 地址带入至 EIP 分页，故无法从 EIP 页面修改，若要修改主站 IP 请参考第 3.1.4.8 章。

➤ 设定从站 IP

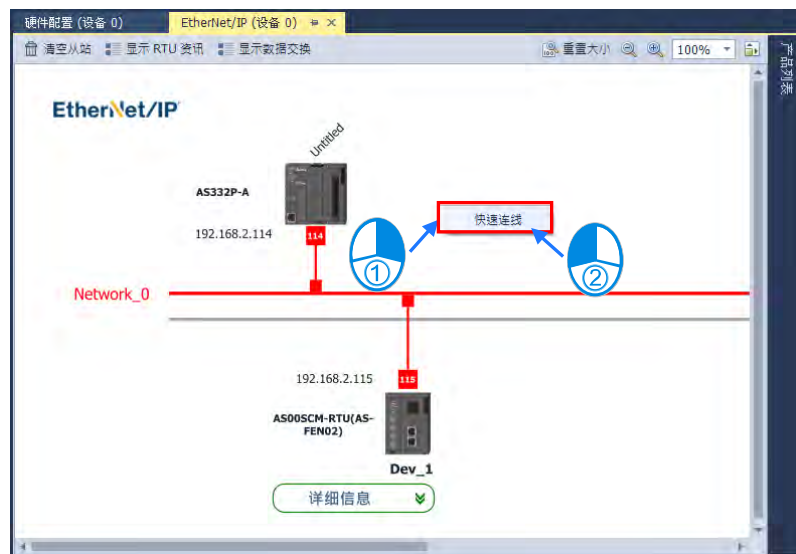
在 EIP 分页中双击从站的 IP，将会跳出编辑窗口，修改完成后关闭编辑窗口即可设定完成。



3.1.5.7 建立联机

➤ 方法一

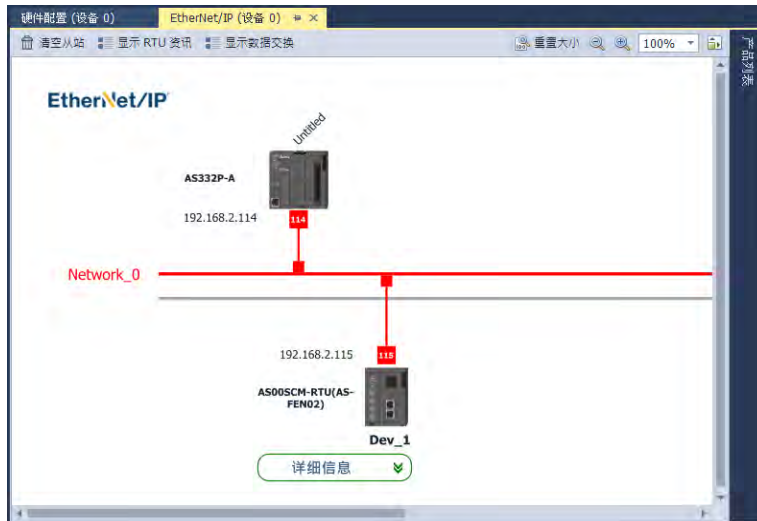
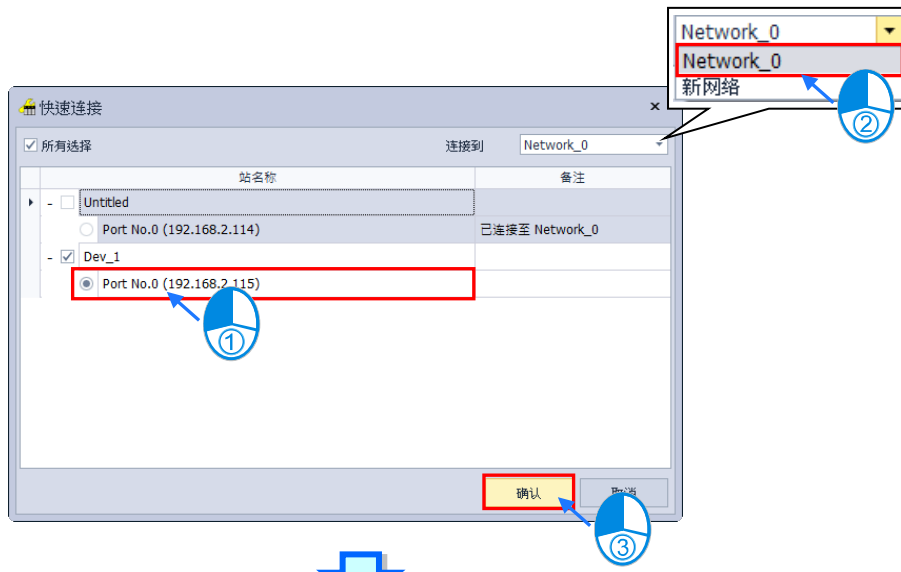
在画面空白处单击右键的快捷选单中单击「快速联机」，将会跳出快速联机窗口。



快速联机窗口，首先可以勾选需要连接的从站，接着在右上角可以选择要联机的网络，设定完成后单击「确认」按键即可完成设定联机。

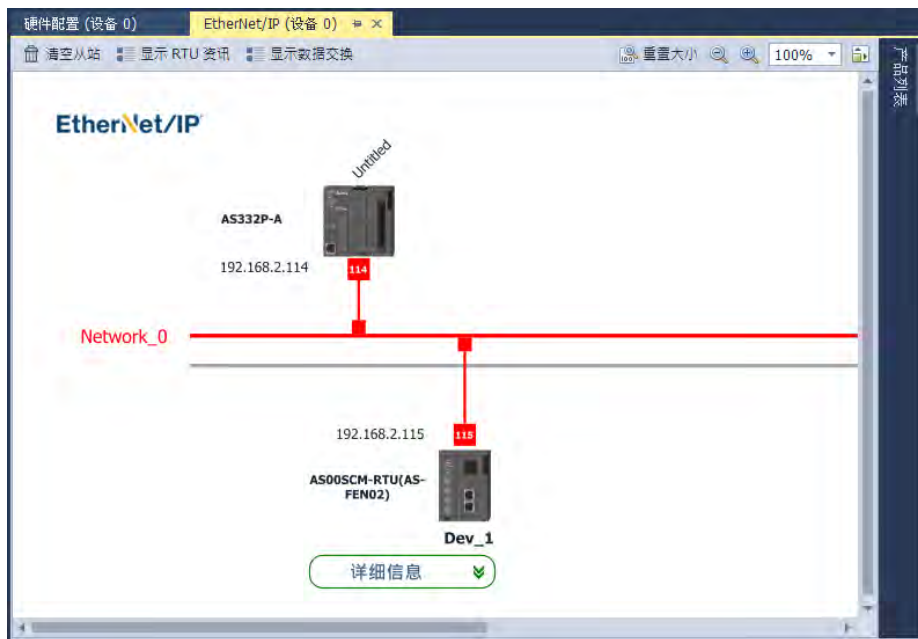
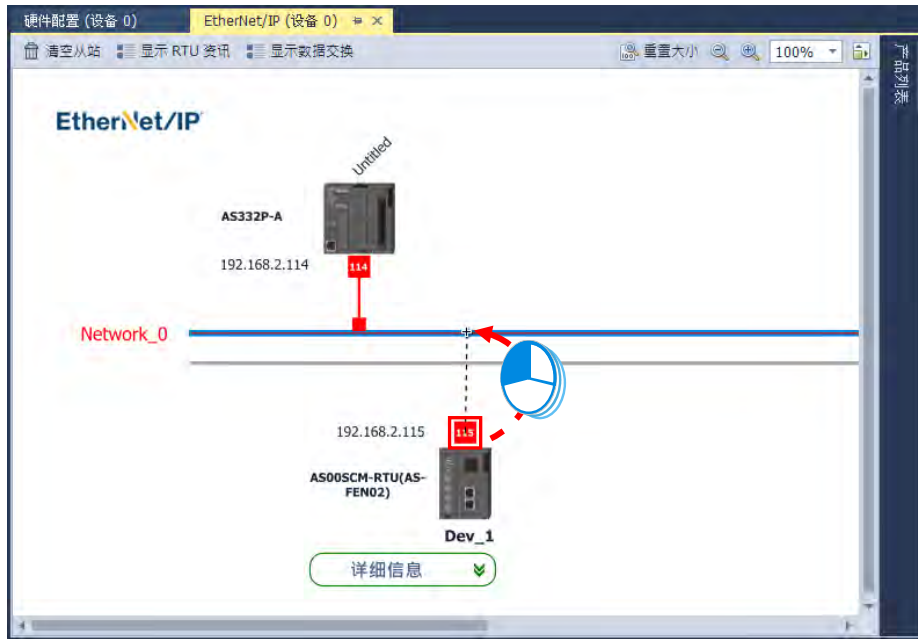
3

3



➤ 方法二

鼠标左键拖曳从站上方的红色方块至想要加入的网络后会呈现蓝色，接着放开鼠标左键，即可建立联机。

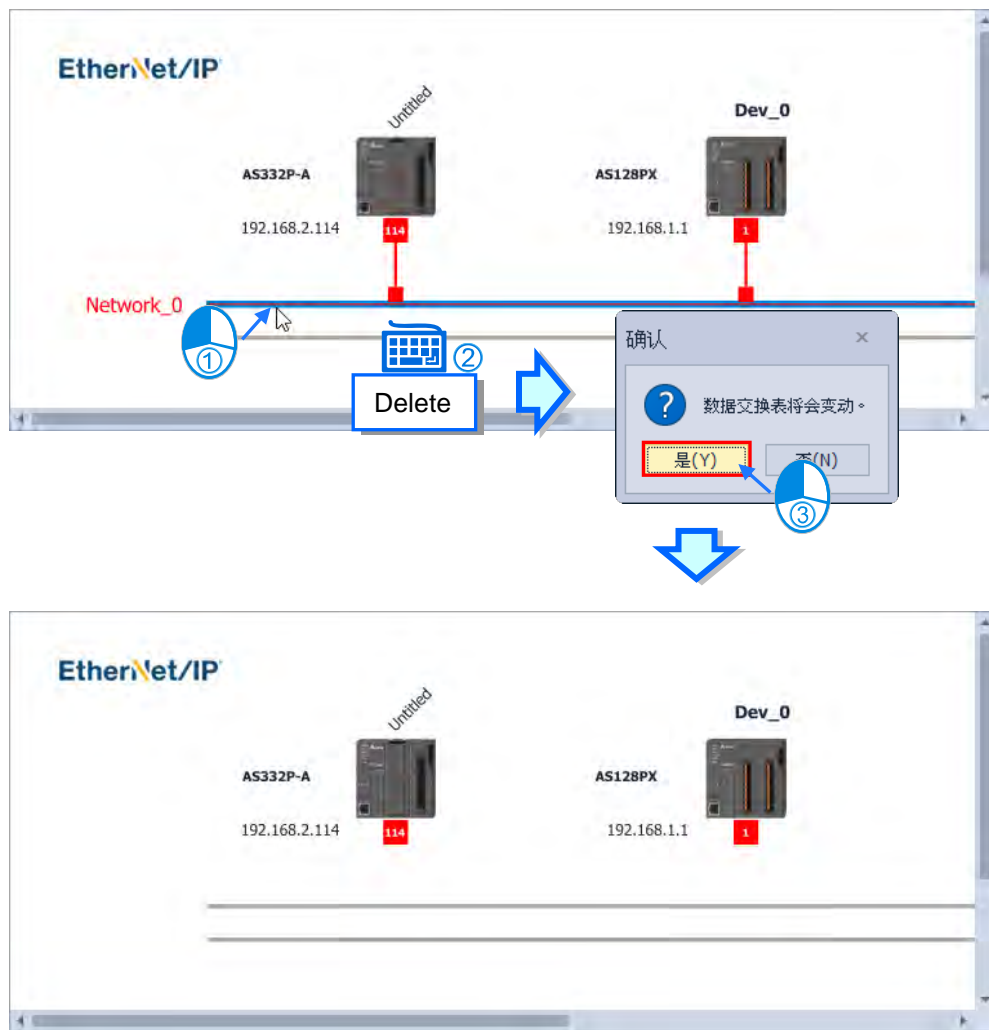


3

3.1.5.8 删除联机

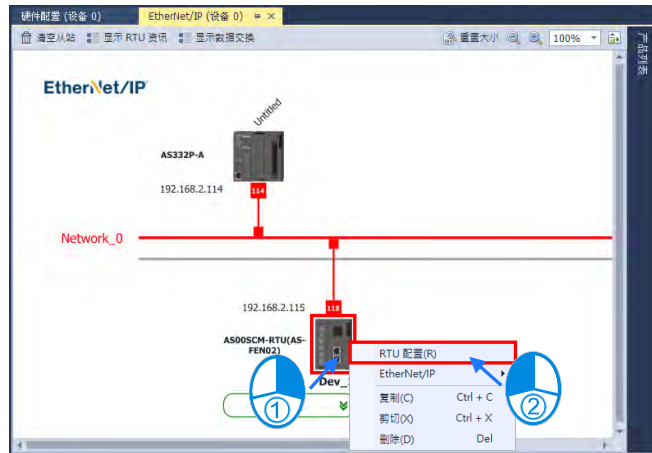
删除联机不仅可以对整个 Network 进行删除，可以针对单一台从站联机进行删除。已删除整个 Network 为例，鼠标单击到欲删除的联机后，被选择的线会呈现蓝色，接着单击键盘的 Delete 键即可删除联机。

3



3.1.5.9 设定 RTU 模块配置

要设定 RTU 模块配置前必须先建立联机，在 RTU 模块图标上鼠标右击，接着单击「RTU 模块配置(R)」，即可开启 RTU 模块配置分页，接着可以从产品量表加入模块至配置区中，加入模块方式请参考第 3.1.3.1 章节。

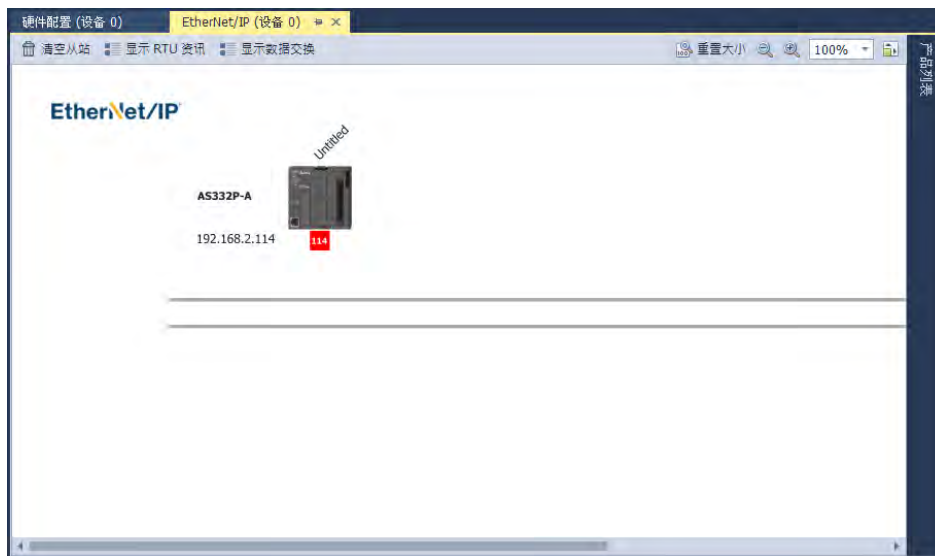
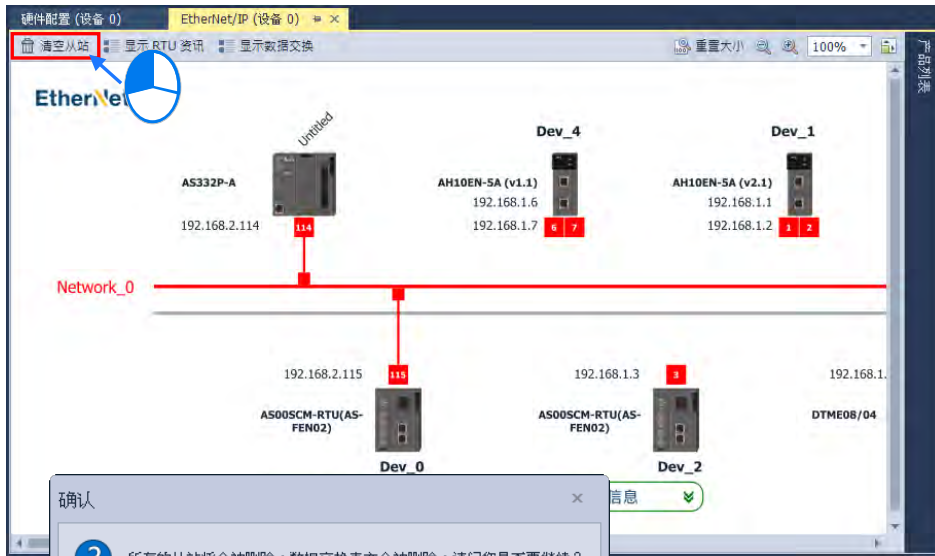


3

3.1.5.10 清空从站

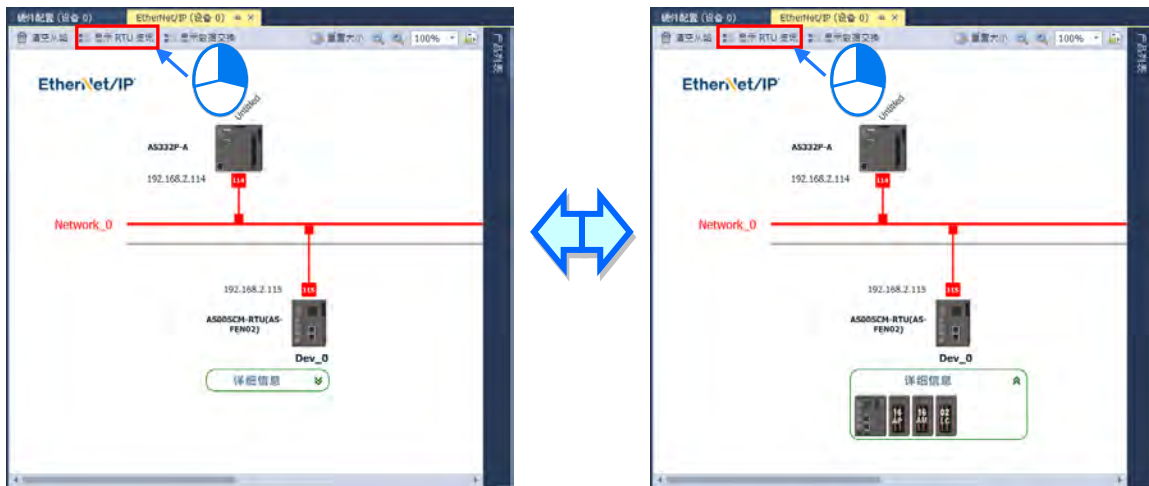
单击 EIP 分页上方的「清空从站」后，单击确认即可清除所有从站，如下图。

3



3.1.5.11 显示 RTU 信息

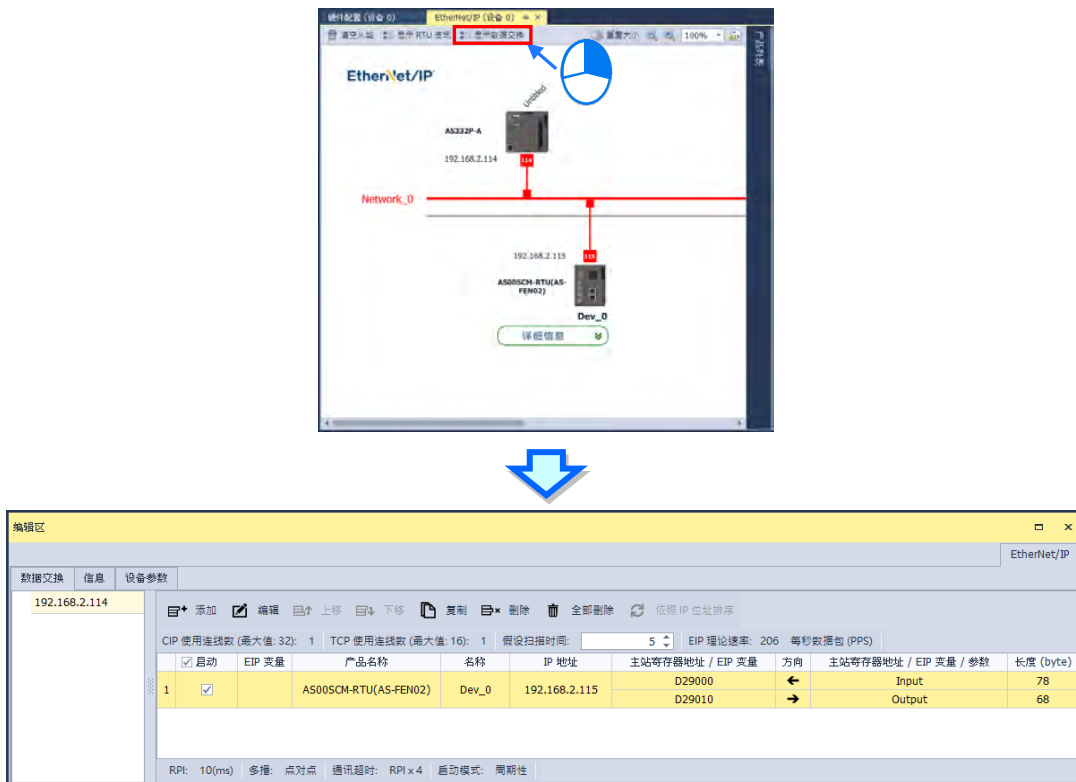
单击 EIP 分页上方的「显示 RTU 信息」后，即可显示或隐藏所有 RTU 模块设定，如下图。



3

3.1.5.12 显示数据交换

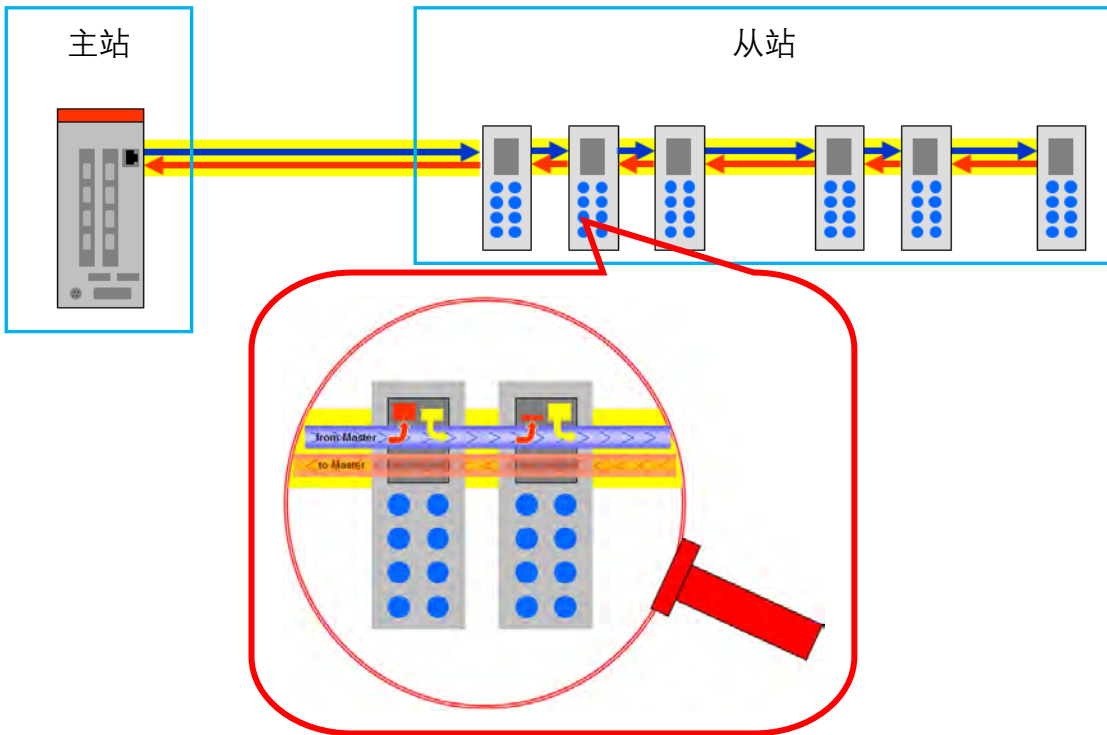
单击 EIP 分页上方的「显示数据交换」后，将会跳出数据交换设定窗口，如下图，详细数据交换设定请参考第 3.1.4.17 章。



3.1.6 EtherCAT

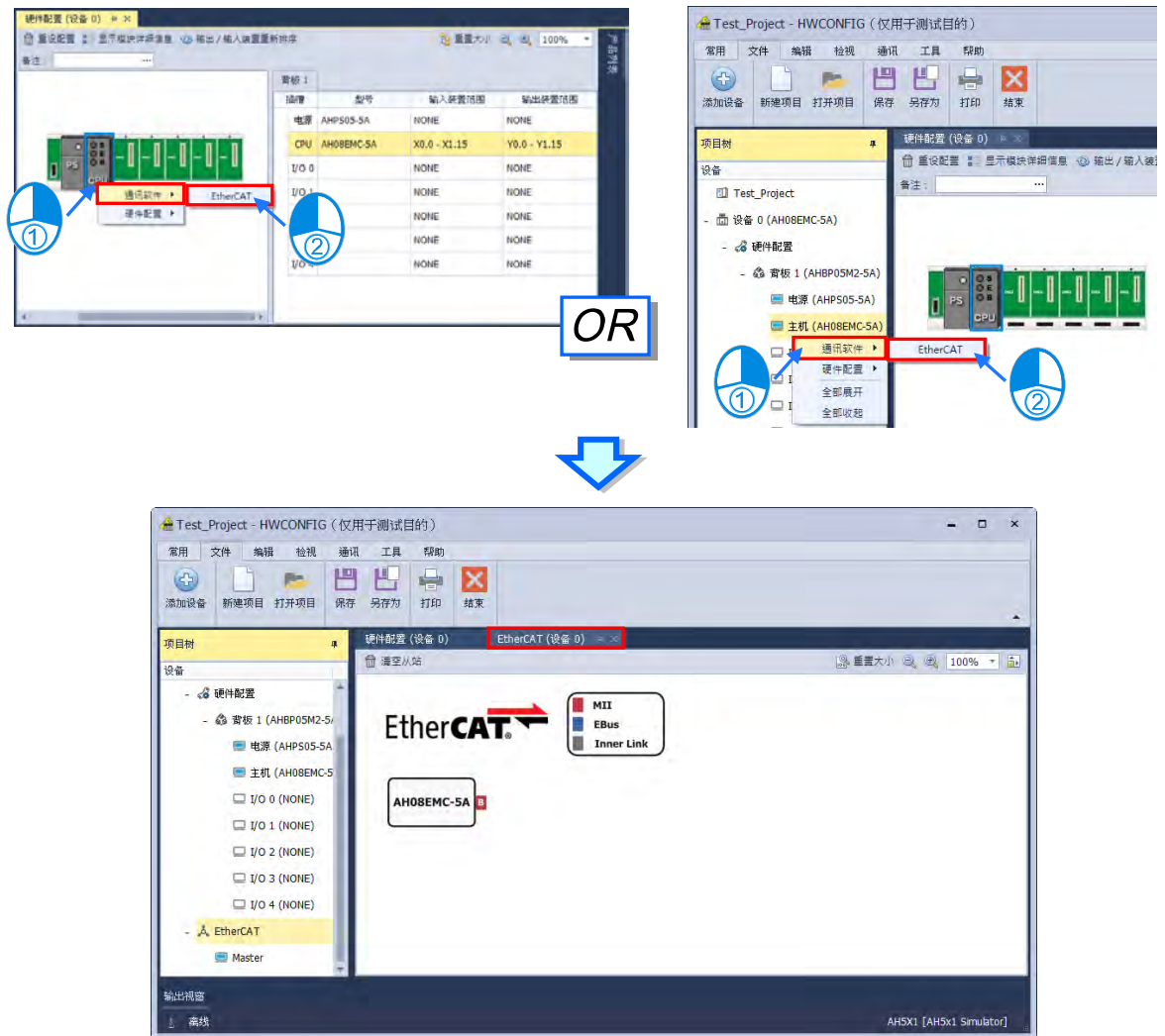
EtherCAT 为新一代的工业网络协议，常用于运动控制的应用。EtherCAT 的特色为主站将同一数据包依序传送到各从站，各从站可以直接读写数据包的内容后，传递到下一个从站，最后一站再将数据包传送回主站，完成数据的传递，如下图，整个数据包在各站之间无须译码，并且各站可由同步时钟进行同步，因此传输同步性可大幅提高，以达成多轴同动的运动控制。并采用 CoE 应用层，因此支持 CANopen 协议的 PDO 数据交换及各种参数与运动模式。

3



3.1.6.1 开启 EtherCAT

在硬件配置区的主机或项目树的主机鼠标右击后，于快捷选项单击「通讯软件」>「EtherCAT」，即可开启 EtherCAT 网络拓扑图。

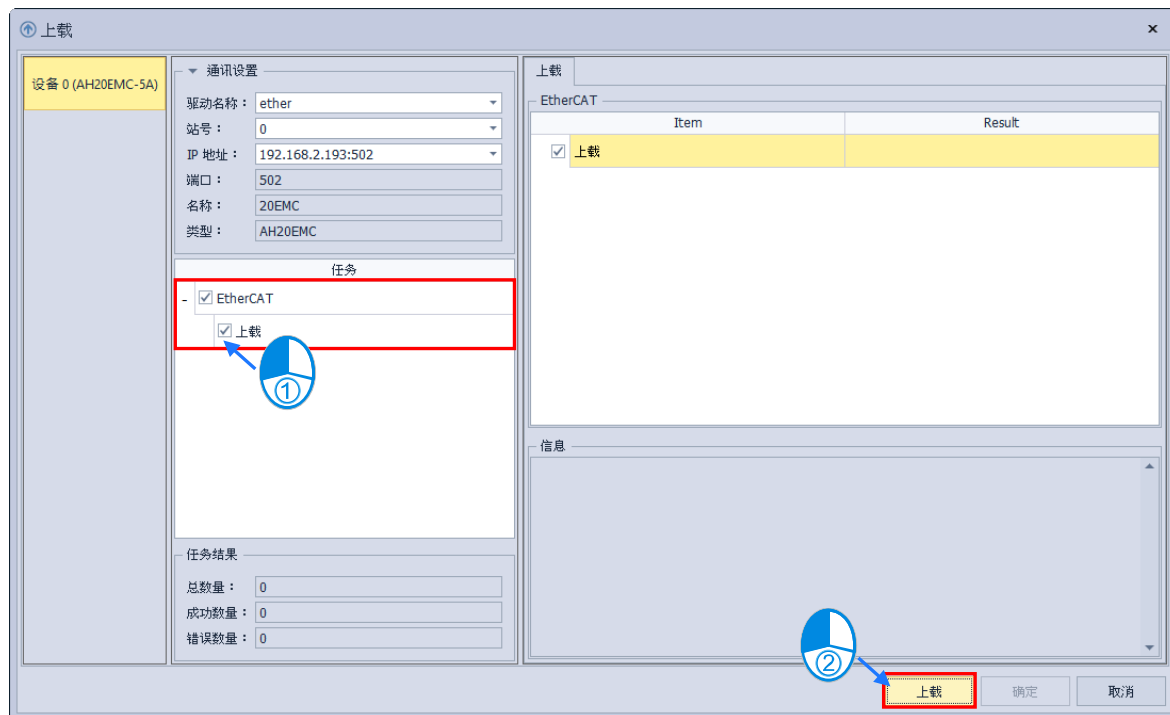


3

3.1.6.2 上载 EtherCAT

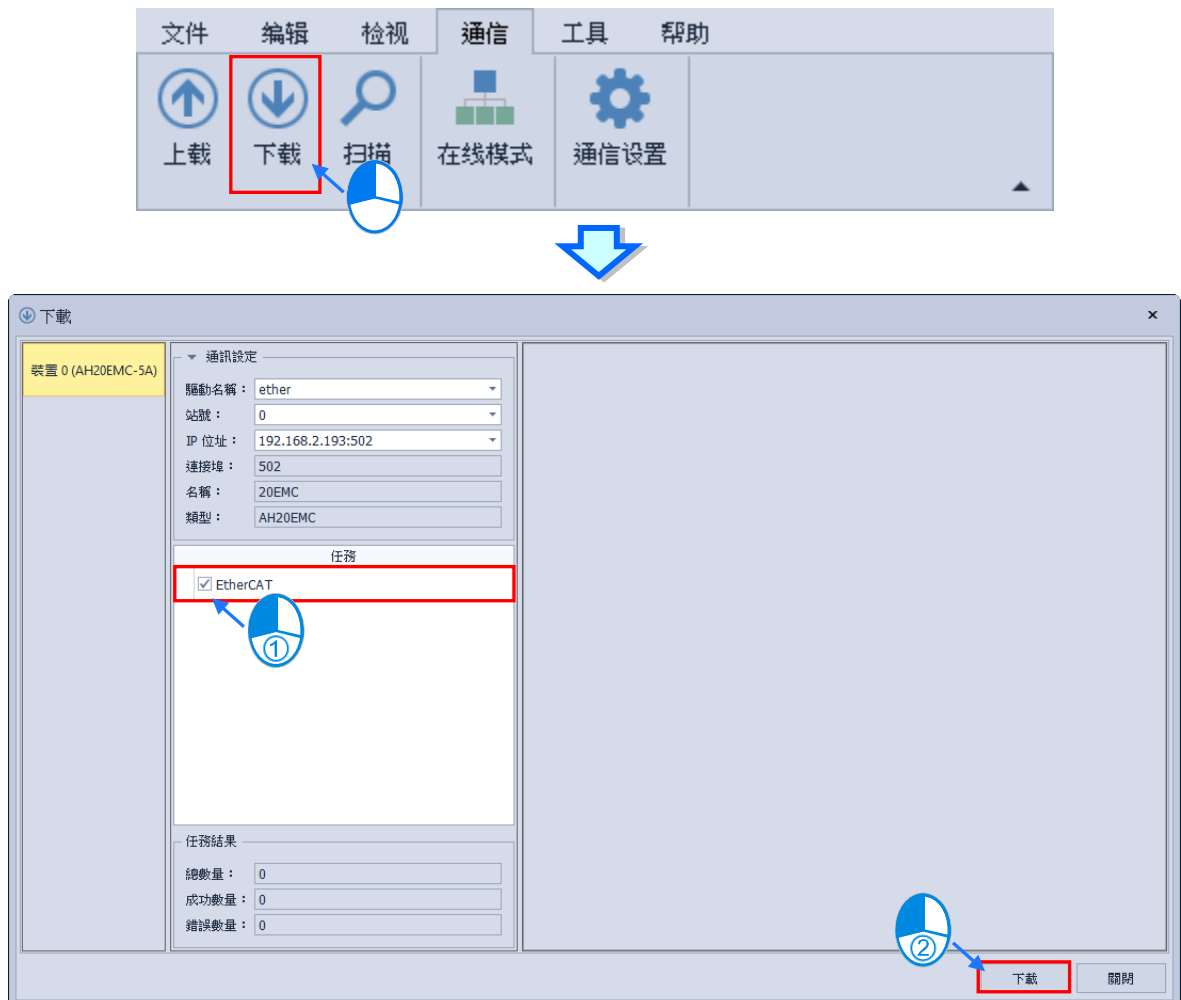
开启 EtherCAT 分页，单击菜单工具栏「通讯」>「上载」后，将会跳出上载窗口，单击「上载」按钮后，等待信息窗口出现成功后，单击「确定」即可更新软件 EtherCAT 设定，若单击「取消」则不会更新软件 EtherCAT 设定，如下图所示。

3



3.1.6.3 下载 EtherCAT

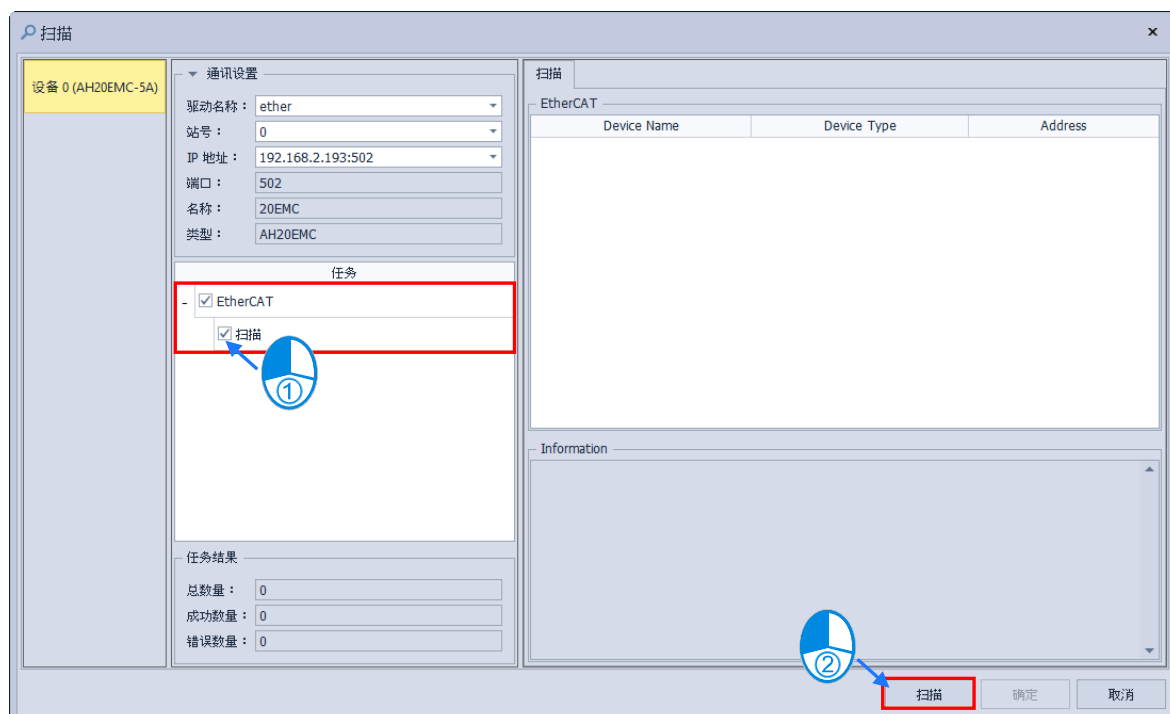
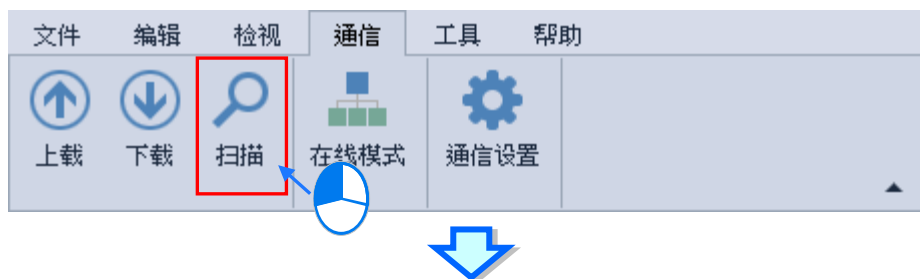
开启 EtherCAT 分页，单击菜单工具栏「通讯」>「下载」后，将会跳出下载窗口，确认下载选项后，单击「下载」，确认完成后即完成下载，如下图所示。



3.1.6.4 扫描 EtherCAT 从站

开启 EtherCAT 分页，单击菜单工具栏「通讯」>「扫描」后，将会跳出扫描窗口，确认从站扫描勾选后，单击「扫描」后在窗口右侧即会出现可通讯的从站，如下图所示。

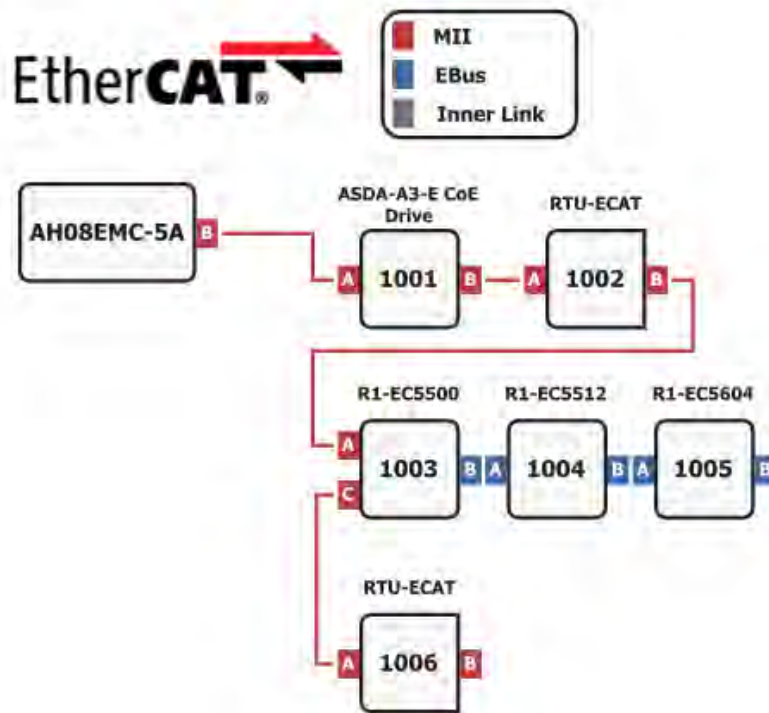
3



3.1.6.5 EtherCAT 网络拓扑图操作

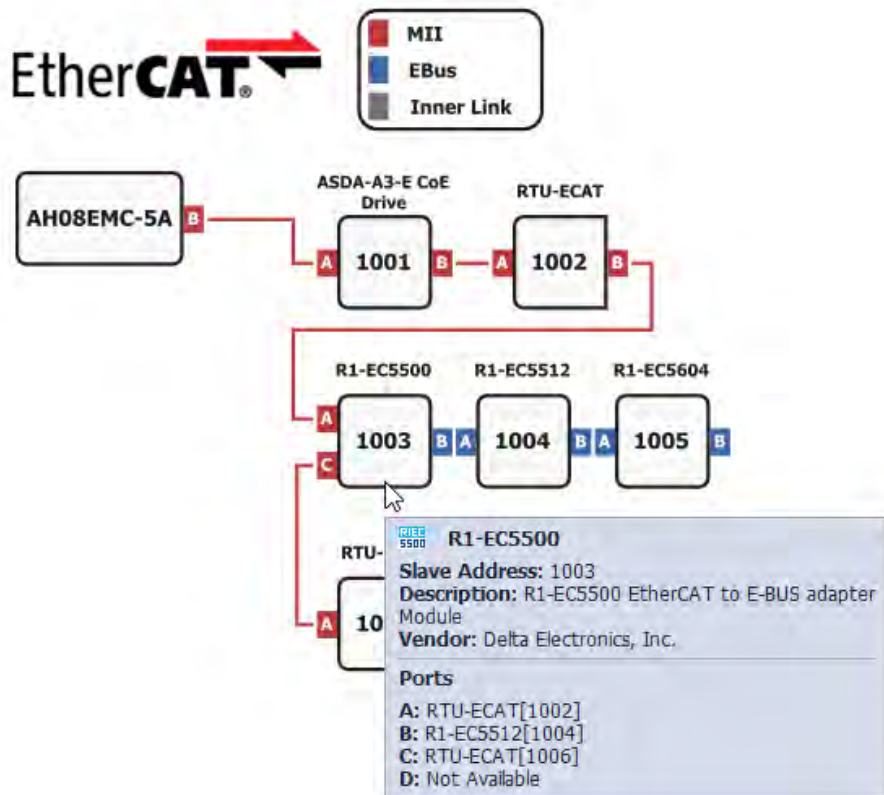
在介绍操作前，本篇会先对于从站的图示作个简易的介绍。

如下图所示主站 / 从站 A 为输入接口，其余皆为输出接口，输入接口的通讯方式必须要与输出接口相同才能连接，EtherCAT 接口有分为 3 种通讯模式分别为 MII、EBus 与 Inner Link。



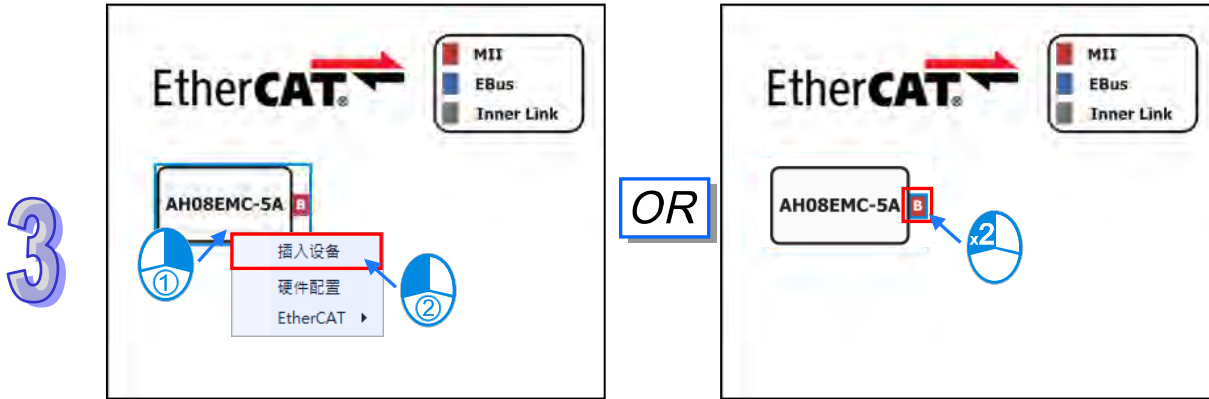
3

将鼠标移动到从站上方将会显示从站详细信息以及连接状态，每个从站最多会有 4 个接口，若从站没有此接口会显示 Not Available，如下图所示。

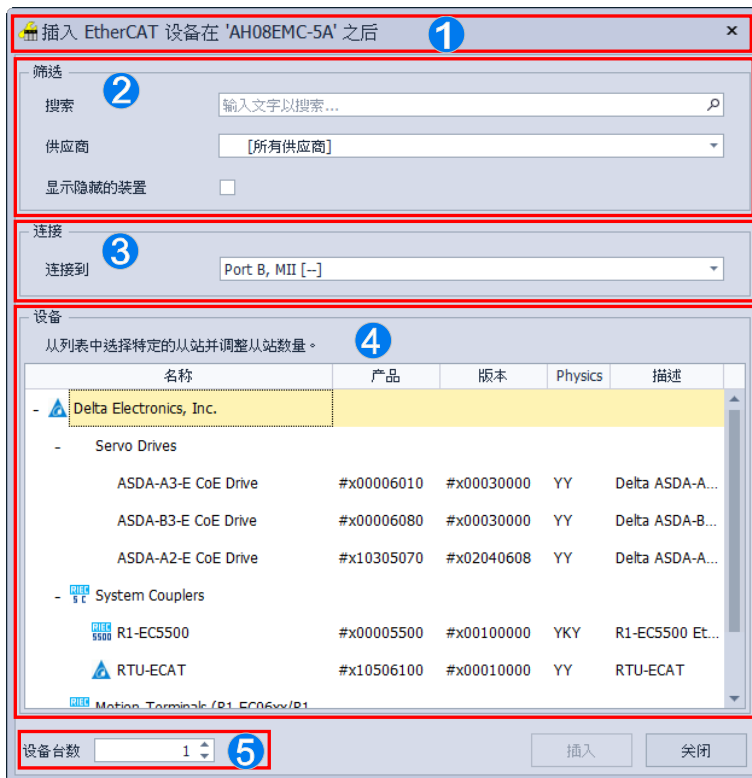


● 新增从站

鼠标右击主站 / 从站，选择下拉选项「插入装置」，或是左键双击主站 / 从站的接口，即会跳出新增从站窗口。



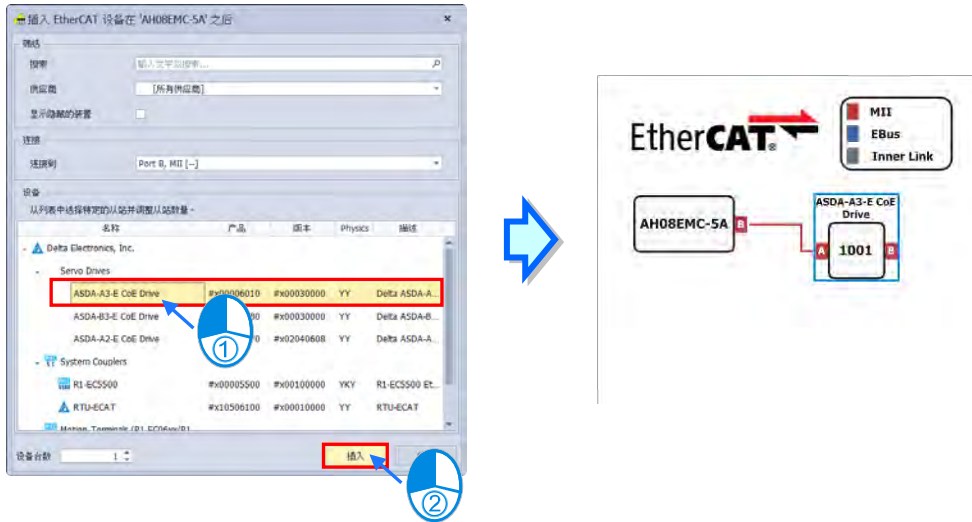
下图为插入从站窗口



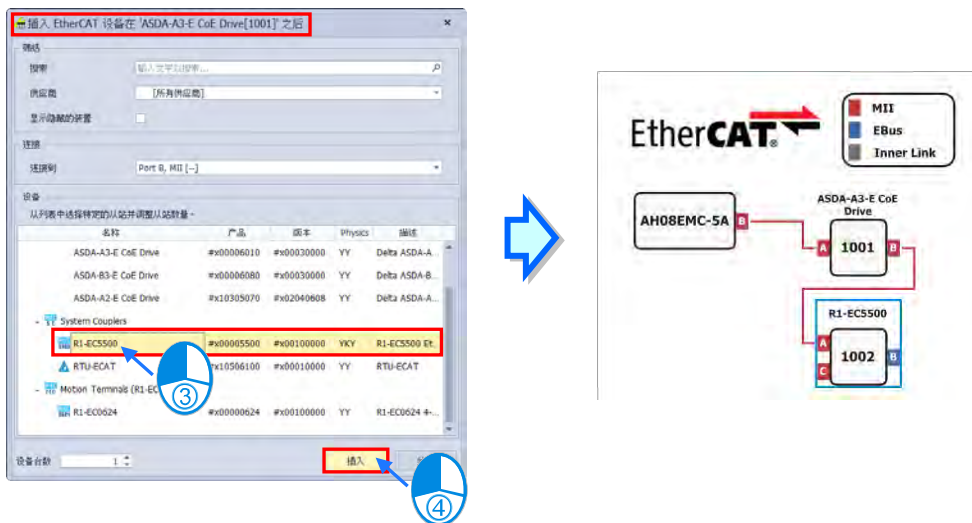
- ❶ 窗口标题：显示从站即将加入的位置。
- ❷ 筛选：可以让用户快速地过滤从站，筛选结果将会显示于下方装置区。
- ❸ 连接：选择欲连接接口，装置区会出现符合选择接口的装置。

- ④ 装置：显示符合接口以及筛选后的装置结果。
- ⑤ 装置台数：设定欲加入的从站数量。

确认选择好从站后单击「插入」即可将从站串接至后方，如下图。



若不关闭窗口，窗口标题会显示目前选取的从站，若继续插入就会往后串接，如下图，完成新增从站后即可关闭插入窗口。



● 复制 / 粘贴从站

鼠标右击欲复制的从站，选择下拉选项中的「复制」或是单击从站后使用快捷键「Ctrl+C」，接着在想要加入的位置上单击右键，选择下拉选项中的「粘贴」或是单击从站后使用快捷键「Ctrl+V」，即可将从站插在所选择从站的后方。

*若欲粘贴的从站 A 接口不符合前面从站输出接口，将无法粘贴。

● 删除从站

鼠标右击欲删除的从站，选择下拉选项中的「删除(D)」或是单击从站后使用快捷键「Delete」，

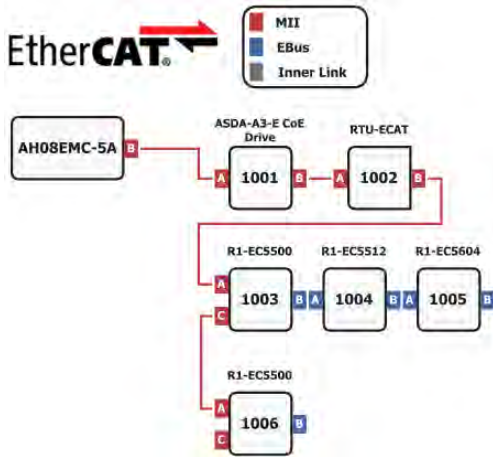
即会跳出要删除的相关从站，确认后单击「是(Y)」，即可删除从站。

*若欲删除的从站 A 接口与后方从站 A 接口通讯模式不同，将会一并被删除。

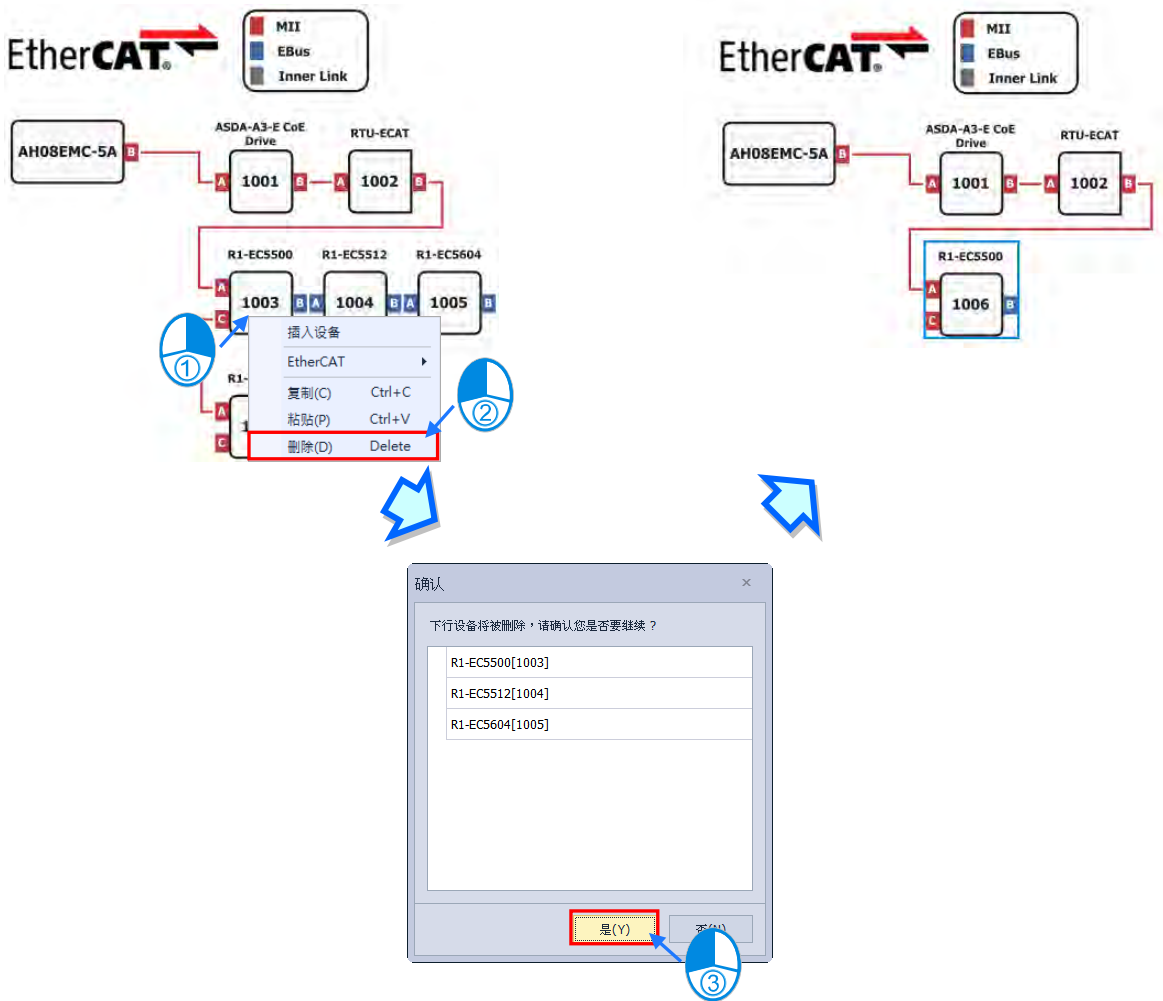
*删除从站后，后方所连接的从站将会依序连接。

以下方案例作解释。

3



如果将从站 1003 移除，从站 1002 因接口不合，故无法与从站 1004、从站 1005 接上，所以删除从站 1003 时会一并将从站 1004、从站 1005 删除，而从站 1006 将会接在从站 1002 后面，如下图。

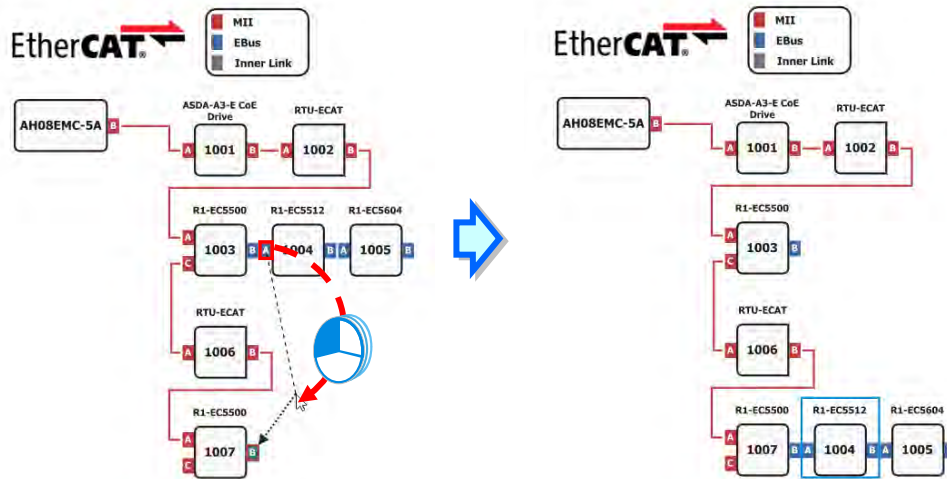


● 删除连接线

鼠标右击欲删除的连接线，选择下拉选项中的「删除(D)」或是单击从站后使用快捷键「Delete」，即会跳出要删除的相关从站，确认后单击「是(Y)」，即可删除连接线后方所有从站。

● 移动从站

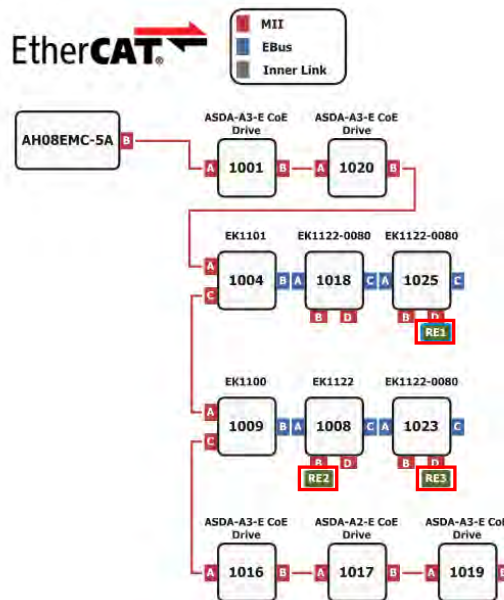
鼠标单击欲移动的从站 A 接口，拖拉至目标位置放开鼠标完成搬移，移动只能搬移至未串接从站的接口，不能中途插入。



3

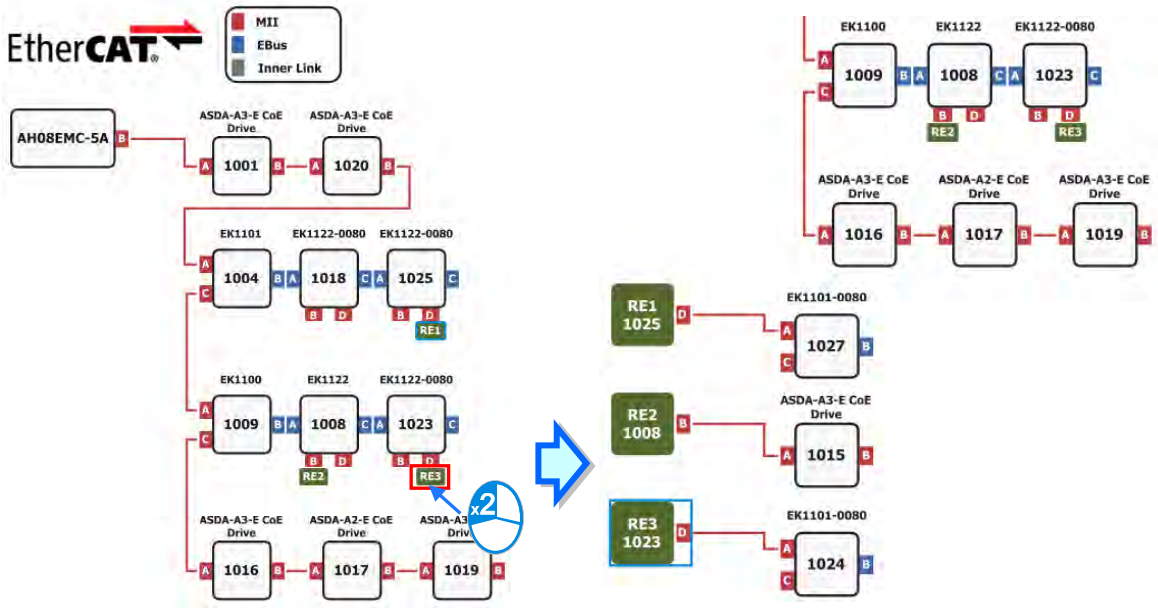
● 快速跳转

当 EtherCAT 拓扑图较为复杂时，软件会自动使用 reference 来协助用户进行编辑，如下图。

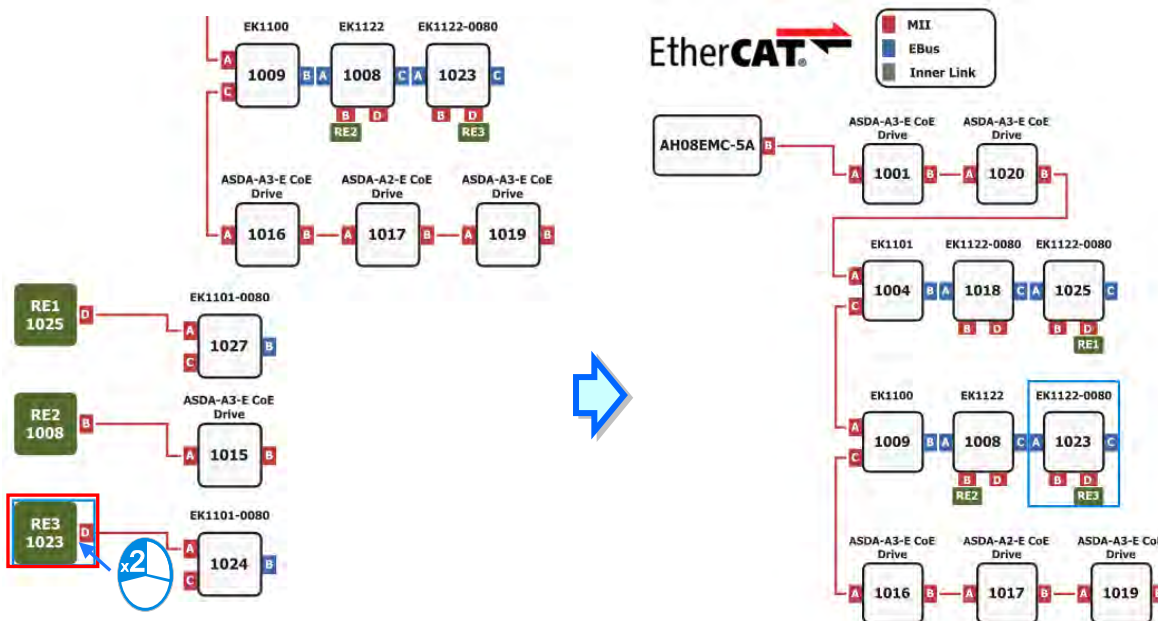


如下图所示，用户可以通过双击 RE3 快速跳至站号 1023 的从站 D 接口继续编辑。

3



如下图所示，用户也可以通过双击 RE3-1023 快速跳至站号 1023 的从站。



3.1.6.6 设定从站

双击欲设定的主站，即会跳出编辑窗口，如下图所示。

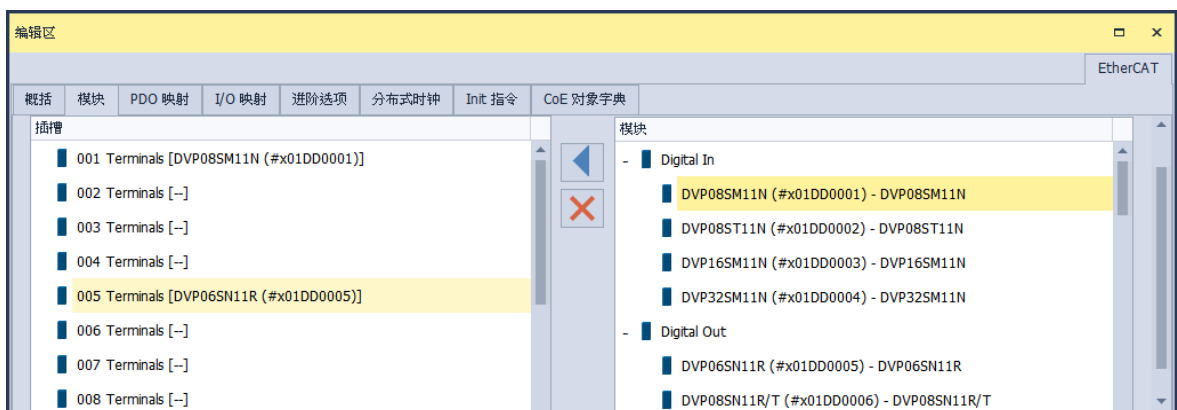
从站的设定有一般、模块、PDO 映像、I/O 映像、进阶选项、分布式时钟、Init 指令以及参数，下方会依序说明，设定项目会依据不同从站而有所不同。



● 一般



- ❶ ESI 信息：设备名称、供货商、产品代码、版本号、描述。
- ❷ 从站地址：设定从站站号(不可重复)。
- ❸ Ports：显示目前所连接的状态(Port A 为输入 Port，Port D、Port B、Port C 为输出 Port)。
- ❹ 注释：输入从站注释。

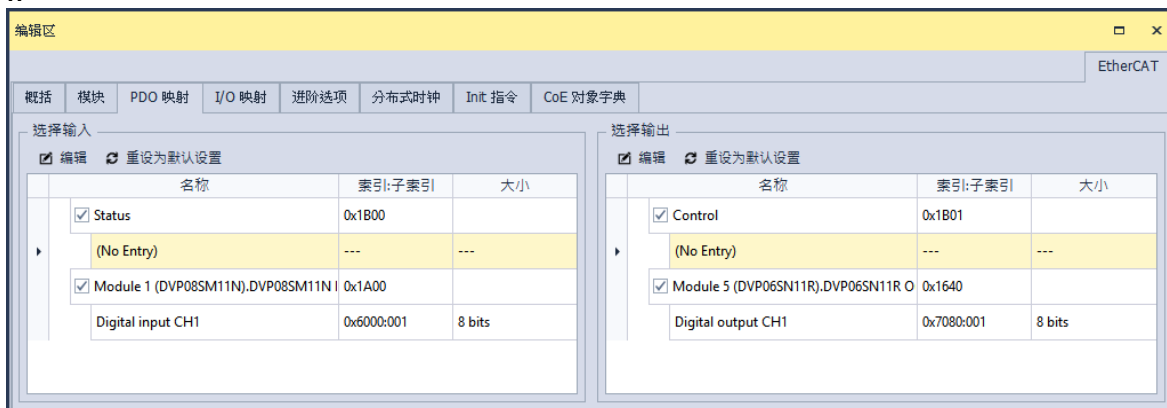
● 模块



左侧为模块插槽，右侧是从站可支持的模块，双击左侧模块或单击  即可将模块放至于插槽内；双击插槽内的模块或单击  即可将所选的模块删除。

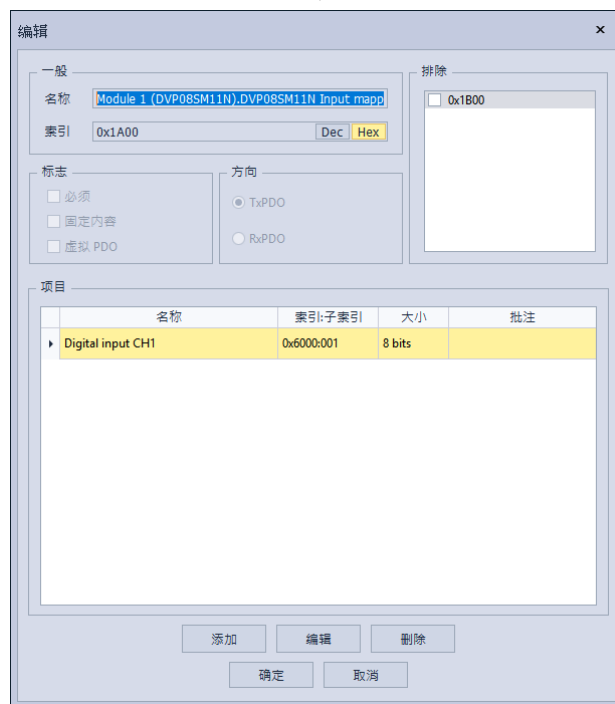
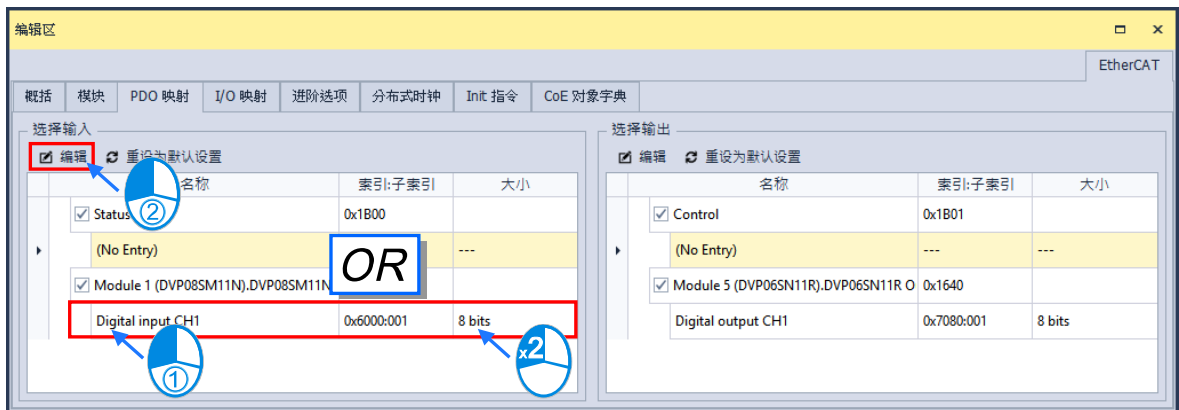
● PDO 映射

3



EtherCAT 网络的数据交换为 CoE 的 PDO 映像，是一种主站与从站架构的周期性数据交换；从站要送出到主站的数据将包入 TxPDO 中，从站要从主站读取的数据则包入 RxPDO 中，在「PDO 映像」页面的输入设定与输出设定字段便列出可供数据交换的 PDO，并且可对 PDO 内容进行编辑。于设备的 ESI 文件中，已经定义了默认勾选的 PDO 与 PDO 的内容，并依据 ESI 的定义，有些设备的 PDO 内容允许用户自行编辑。

用户可直接勾选 要交换的 PDO。欲编辑 PDO 的内容时，可单击欲编辑的 PDO 的项目后，单击「编辑」按钮或双击欲修改项目后便会开启编辑 PDO 窗口，也可以使用「还原为默认」将全部设定初始化选项。



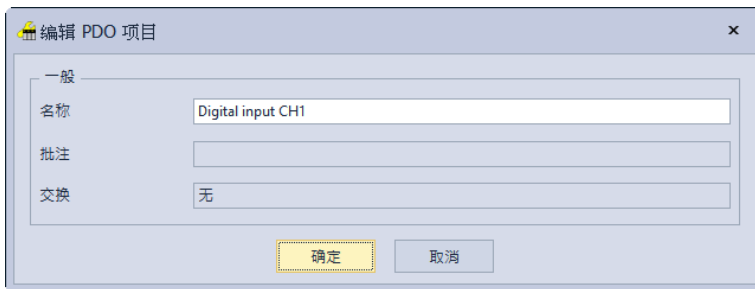
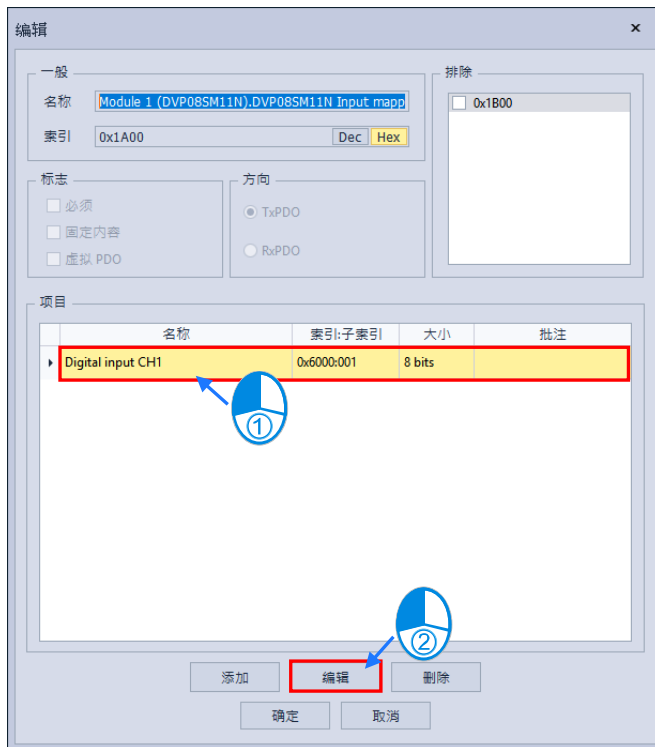
PDO 编辑窗口内容说明如下：

- i. **一般**：可设定 PDO 的名称；及显示此 PDO 的索引值，可切换 Dec 或 Hex 以十进制或十六进制显示索引值。
- ii. **标志**：若勾选为强制性，则此 PDO 不可删除；若勾选为固定内容，则此 PDO 的内容不可被变更；若勾选为虚拟 PDO，则此 PDO 不配置内容。
- iii. **方向**：说明此 PDO 的类别为 TxPDO 或 RxPDO。
- iv. **排除**：设定排除的 PDO，即为当此 PDO 在「PDO 映射」页面被勾选时，设定不可同时被勾选的其它 PDO 的索引值。
- v. **项目**：列出此 PDO 包含的内容。

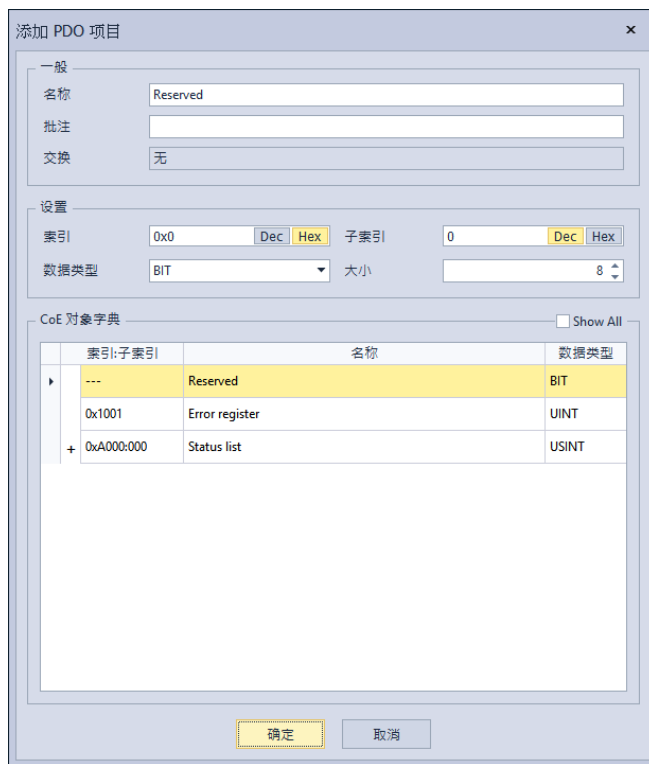
先以鼠标单击项目后，可单击下方的「删除」按钮删除此项目，或单击「编辑」按钮，将出现编辑 PDO 条目窗口显示此项目的内容，并且可编辑项目的名称，如下图，单击「确认」按钮完成编辑。

3

3



单击编辑 PDO 窗口中的「新增」按钮后，便会出现新增 PDO 项目窗口，如下图



在此窗口中从 CoE Object 目录中选择要加入 PDO 的内容项目，并且可设定项目的各项字段。新增 PDO 项目窗口内容说明如下：

- vi. 一般：可设定 PDO 项目的名称及注释。
- vii. 设定：显示索引、子索引、数据类型及 Bit 长度。
- viii. CoE 对象字典：显示 CoE 对象字典内的参数。

单击「确认」按钮后便会将此条目新增到 PDO 的项目内容之中。

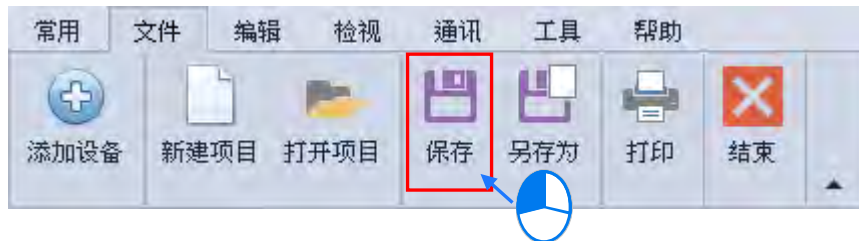
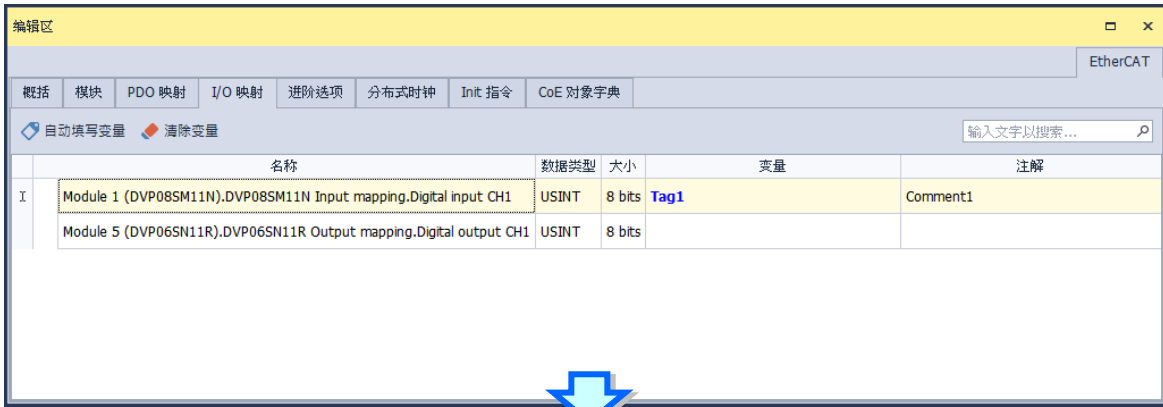
● I/O 映射



「I/O 映像」分页将会列出由「PDO 映像」分页所设定的 PDO 内容，并且供用户设定变量符号，用户也可以单击「自动填入符号」使软件自行填写，之后便可在 ISPSOFT 中导入在此定义的变量符号，即可在程序中存取对应的 EtherCAT 从站参数。在数值列表中欲建立变量符号的符号字段输入符号名

称及注释(建立的符号名称最大长度,不可超过 40 个字符,符号名称不可使用特殊字符,例如 *、#、?、\、%、@ ...等,且开头不能为数字;符号名称命名不可以重复),完成后单击菜单工具栏的 文件 > 储存,储存后才能将 I/O 映像参数带入 ISPSOft 中。

3

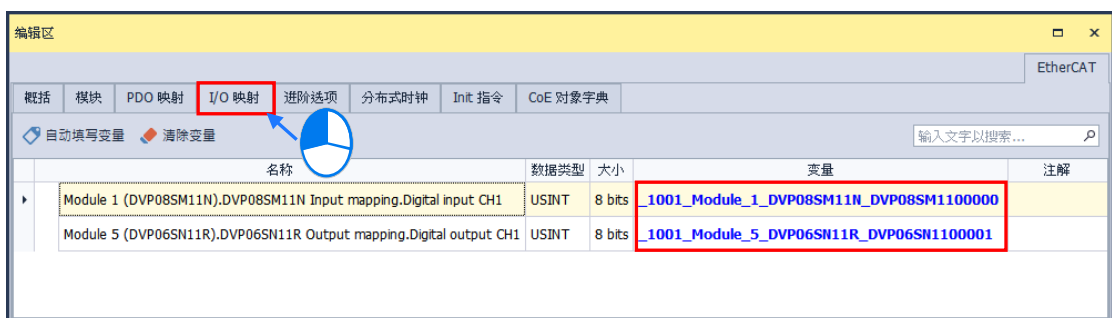


接着开启 ISPSOft 中的项目树的全局符号 >EtherCAT 符号表项目,在 EtherCAT 符号表窗口中单击鼠标右键,再单击选单中的「与 ECAT Builder 同步」,EtherCAT 符号表便会出现加入到 ECAT Builder 中的变量符号,若关闭 HWCONFIG,也会自动完成 EtherCAT 符号表更新,之后便可在 ISPSOft 的程序中使用此变量符号。

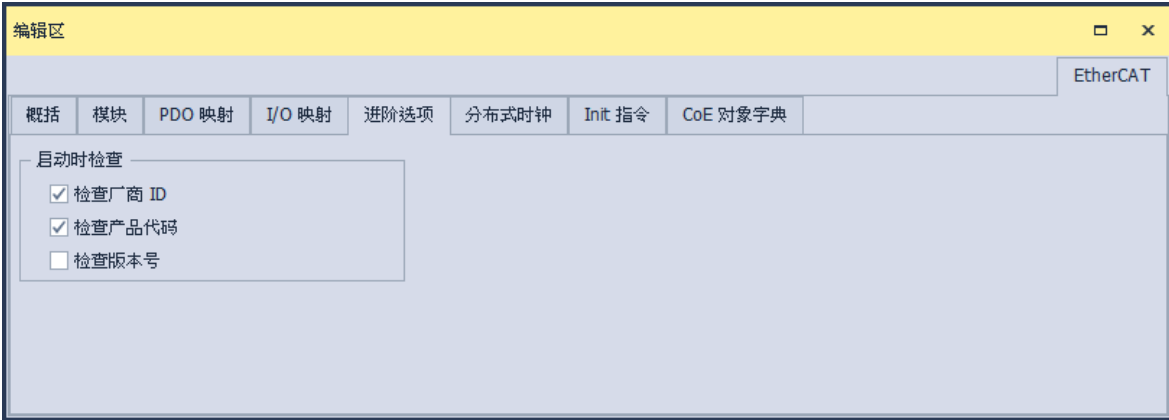


3

在「I/O 映射」分页，若单击「自动填写符号」按钮，则未定义的符号字段将会自动产生符号，自动产生的符号名称由站号与 PDO 内容所组成（产生的符号字符长度不得超过 40 个字符，超过后的内容将自动被删除，其倒数三码会变成 000~999 的流水号）。如下图。若单击「清除符号」按钮，则此站所有符号皆被清除。



- 进阶选项



3

用户可设定在 PLC 运行时，会确认 EtherCAT 的从站信息，若设定项目匹配错误 PLC 将会回报错误码并停止运行。

● 分布式时钟



让所有 EtherCAT 设备使用相同的系统时间，从而控制各个设备任务的同步执行。分布式时钟有 Free run、DC 同步与 SM 同步三种模式，但会依据从站支持的同步模式而显示不同的设定选项。

● Init 指令

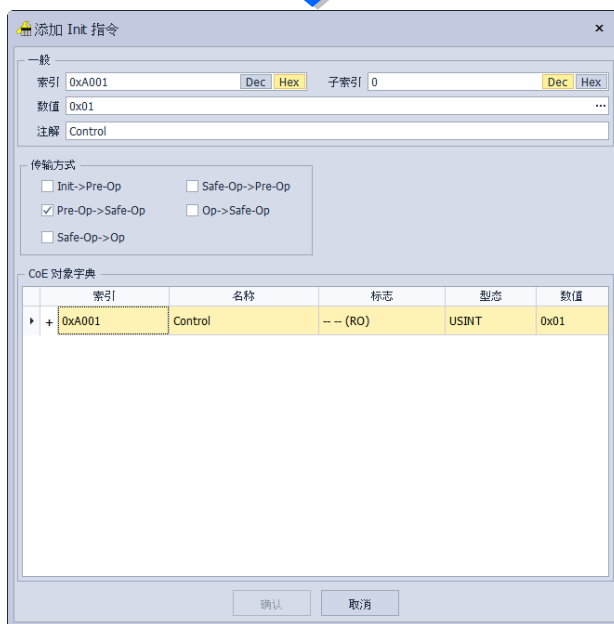


「Init 指令」分页列出的指令为在 EtherCAT 网络运行的特定的状态时，主站将对从站读写数值，列表会带出 ESI 文件已经默认定义的指令，用户可视需求增减或修改。

单击「新增」按钮后，将出现「新增 Init 指令」窗口，如下图。先从 CoE Object 目录中单击要读写的参数，在上面的区块进行编辑。单击确认后便会加入到 Init 指令列表中，「新增 Init 指令」窗口功能说明如下。



3

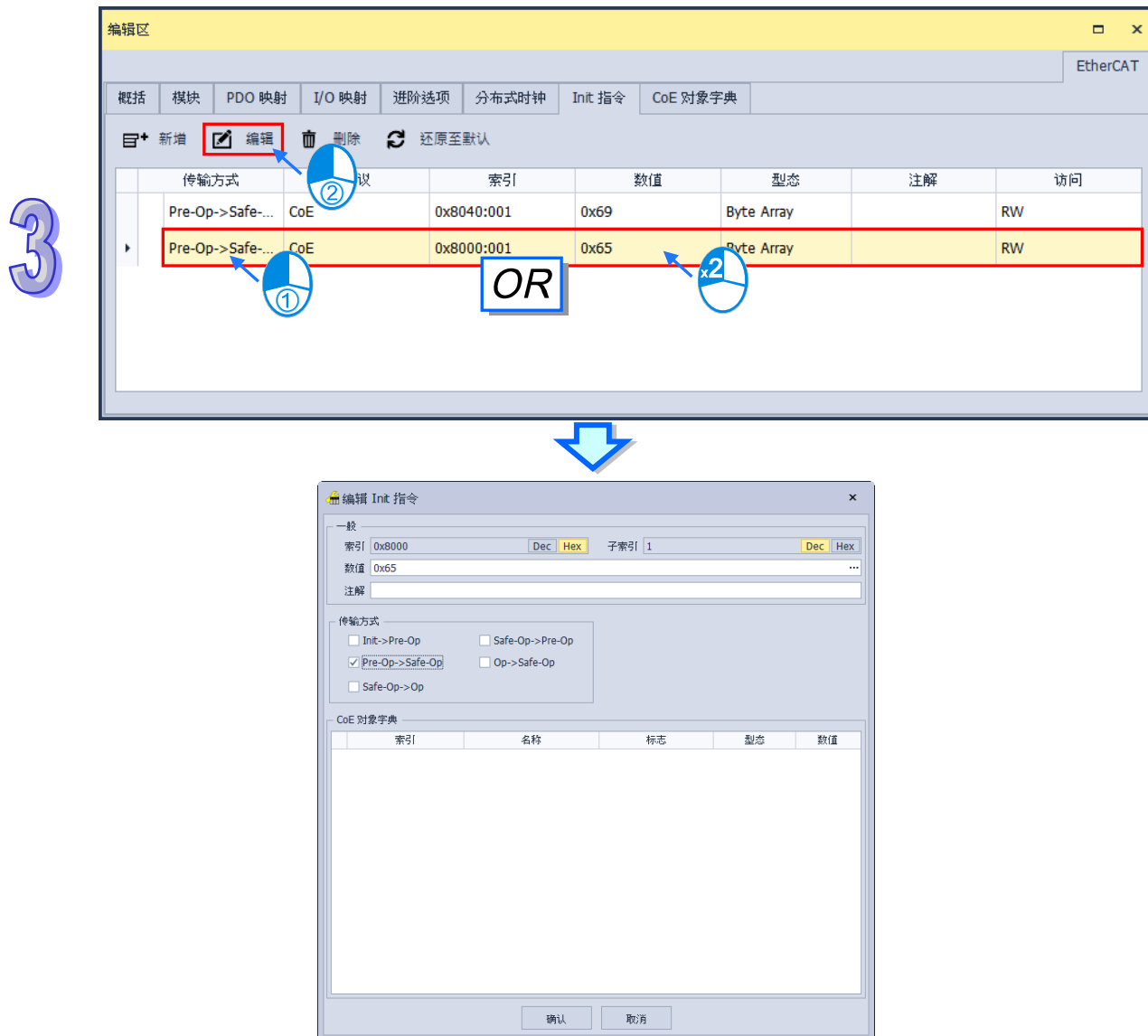


Init 指令编辑窗口内容说明如下：

- ix. 一般：可对选择的参数设定数值或注释，数值可切换为十进制或十六进制显示。
- x. 传输方式：勾选设定 EtherCAT 网络在何种状态转换时触发此指令。
- xi. CoE 对象字典：列出此站支持的所有参数。

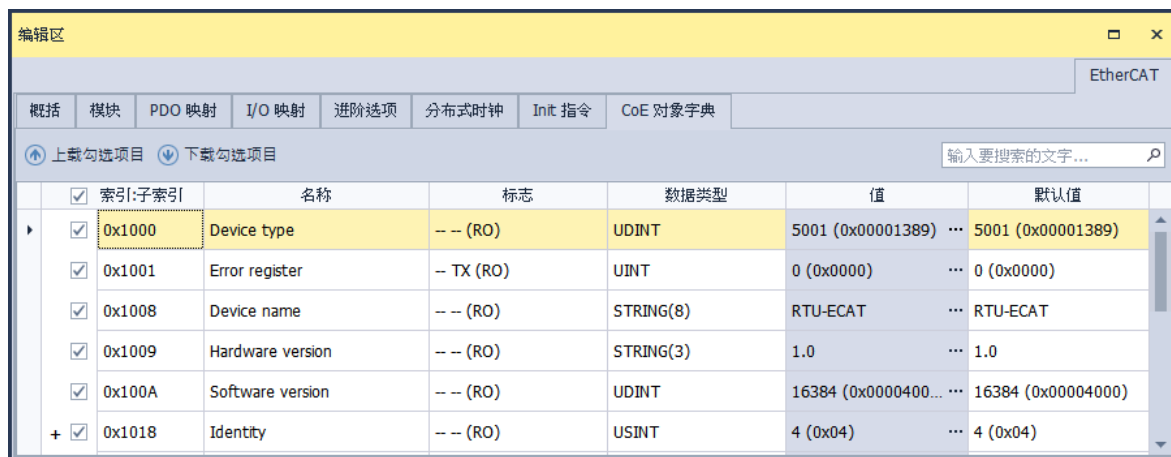
先以鼠标单击项目后，单击「确认」按钮完成编辑。

欲编辑 Init 指令的内容时，可单击欲编辑的 Init 指令的项目后，单击「编辑」按钮或双击欲修改项目后便会开启编辑 Init 指令窗口，可以使用「删除」将 Init 指令移除；也使用「还原为默认」将全部设定初始化选项。



● CoE 对象字典

「CoE 对象字典」分页用于读写从站参数的数值或状态。列出供选择的参数与相关信息。标志字段说明参数可为 RO (只读) 或 RW (可读可写)。若要更新数据可以单击分页中的「上载勾选项目」，若要将参数写入从站中可以单击「下载勾选项目」。



索引:子索引	名称	标志	数据类型	值	默认值
0x1000	Device type	-- (RO)	UDINT	5001 (0x00001389) ...	5001 (0x00001389)
0x1001	Error register	-- TX (RO)	UINT	0 (0x0000) ...	0 (0x0000)
0x1008	Device name	-- (RO)	STRING(8)	RTU-ECAT ...	RTU-ECAT
0x1009	Hardware version	-- (RO)	STRING(3)	1.0 ...	1.0
0x100A	Software version	-- (RO)	UDINT	16384 (0x00004000) ...	16384 (0x00004000)
0x1018	Identity	-- (RO)	USINT	4 (0x04) ...	4 (0x04)

使用此功能前，必须确认软件设定要与 PLC 内部 EtherCAT 设定一致。

3.1.6.7 设定主站

双击欲设定的主站，即会跳出编辑窗口，如下图所示。



编辑区 EtherCAT

主站名称: Delta_ECAT_Master 停用所有从站

模块名称: AH20EMC-5A **注释**

周期时间(微秒): 2000

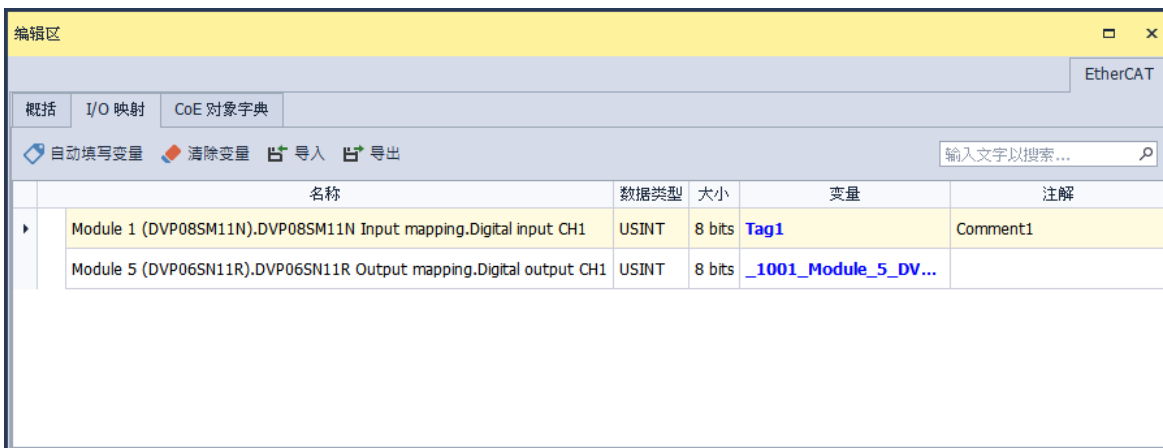
- 概括

「概括」分页列出主站相关信息

- ① 主站名称：机种默认，用户无法改变。
- ② 周期时间(微秒)：设定主站发送数据交换通讯命令的周期时间。
- ③ 模块名称：主站型号。
- ④ 停用所有从站：停用 EtherCAT 功能。
- ⑤ 注释：设定主站注释。

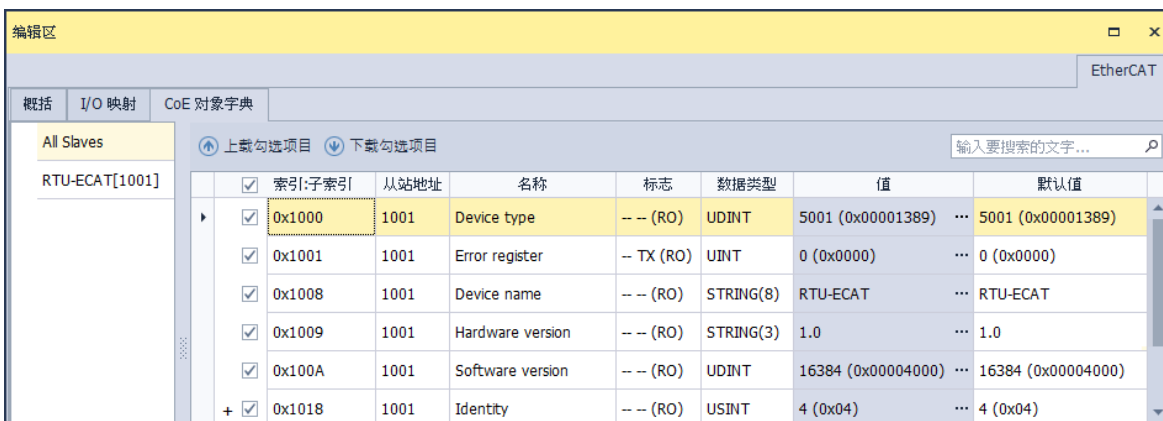
● I/O 映射

3



「I/O 映像」分页列出所有从站个别的「I/O 映像」分页内容，供用户参考此网络所有的数据交换与变量符号信息，在此分页中可编辑符号名称或注释，也可以进行符号的导入、导出功能。

● CoE 对象字典



「CoE 对象字典」分页将列出所有从站的参数分页中所加入的参数，如第 3.1.6.6 节的 CoE 对象字典说明，在从站地址字段将列出参数的来源从站 ID。单击「上载勾选项目」按钮，便会将所勾选的从站参数数值读回并显示到数值字段。对于可写入的参数，用户可直接在数值字段修改数值后，单击「下载勾选项目」，此数值便会被写入到此参数中。下载过的参数值将随项目一同保存，再次开启项目后也可单击「上载勾选项目」按钮，将所有「CoE 对象字典」分页的参数值一次下载到所有目标设备中。

3.2 硬件规划工具 - HWCONFIG 3.0

HWCONFIG 是 ISPSOFT 的硬件规划工具之一，其功能包括模块背板的配置、CPU 主机参数的设置、模块参数设置、硬件参数上下载及简易的在线检测与诊断功能。本章节会先就各机种皆通用的功能先行叙述，再指出某机种及其特有的功能。

请注意！所有硬件规划的参数在设置完毕后，必须下载至主机中才可生效。

3.2.1 HWCONFIG 的环境介绍

在项目管理区的「HWCONFIG」上双击鼠标左键即可打开硬件规划工具的工作窗口。若是使用 DVPxxMC 机种，硬件规划工具的工作窗口如下图显示。



- ① 机种主机图样。
- ② 描述：描述机种主机的功能。
- ③ 注释：主机的注释。
- ④ 版本：主机的 DDF 版本。
- ⑤ 参数：单击「参数设定」钮，即可进入主机参数设定。
- ⑥ CANopen：单击「CANopen Builder」钮，即可开启「CANopen Builder」软件。

3.2.2 主机的版本管理

随着主机与模块版本的更新，可使用的功能或参数可能也会有所不同，而在 HWCONFIG 中，提供了定义功能与参数的设备描述文件；当在安装 ISPSOft 时，最新版的设备描述文件便会一并被安装至系统中。

因设备描述文件的版本会向下包含所有固件版本的信息，因此当我们在 HWCONFIG 当中进行配置时，便可依据实际设备的固件版本而选择适合的配置，且依据所配置的版本，可设定的参数也会有所不同；不过当发生在 HWCONFIG 中所设定的固件版本与实际设备不符时，两者之间的兼容性与下载之后的状态请参考下表的叙述。

3

版本差异	兼容性说明
设定版本较实际固件版本旧	设备仍可正常运作，而未在 HWCONFIG 中定义的模块功能或参数则会以系统默认的状态参与运行。
设定版本较实际固件版本新	设备无法运作并进入错误状态，此时用户必须先先在 HWCONFIG 中重新选择符合实际模块的固件版本，并在重新确认参数之后再次执行下载的程序。

* 1. 上述的兼容性机制仅限 V1.0 版本以后的模块，V1.0 版以前的版本则无法与 V1.0 版或以上的版本兼容。

* 2. 依据机种的不同，设备描述文件可能称为「DDF」或「MDS」文件。

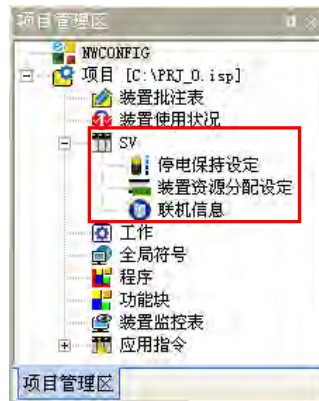
3.3 DVP 系列机种的参数设定

旧型的 DVP 系列机种没有支持 HWCONFIG 工具，新一代的 DVP-ES3 机种有支持 HWCONFIG，DVP-ES3 机种的参数设定可参见第 3.3.4 节。

3.3.1 DVP 主机的系统管理工具

在 DVP 系列机种的项目中，将项目管理区的主机名称展开后，在其下层便可发现与该主机相关的系统管理工具，而部分的功能则须与主机进行联机后方可操作。各对象的功能简述如下：

- **停电保持设定：**设置主机停电保持装置的起止范围。
- **装置资源分配设定：**设置自动配置变量符号时所使用的装置范围。
- **联机信息：**查看目前主机的系统设置与参数。

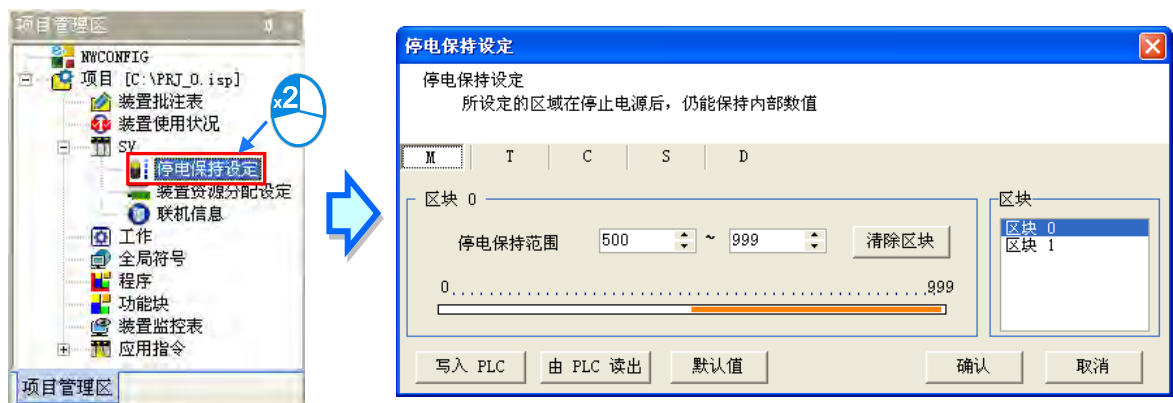


上述功能除了「装置资源分配设定」将会在第 6.2.11 节的内容当中另行介绍之外，「停电保持设定」与「联机信息」的功能将会在接下来的两个小节当中逐一说明。

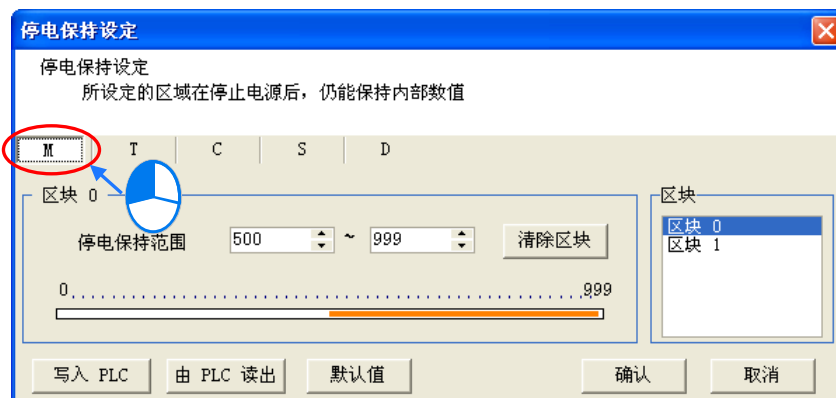
3

3.3.2 停电保持设定

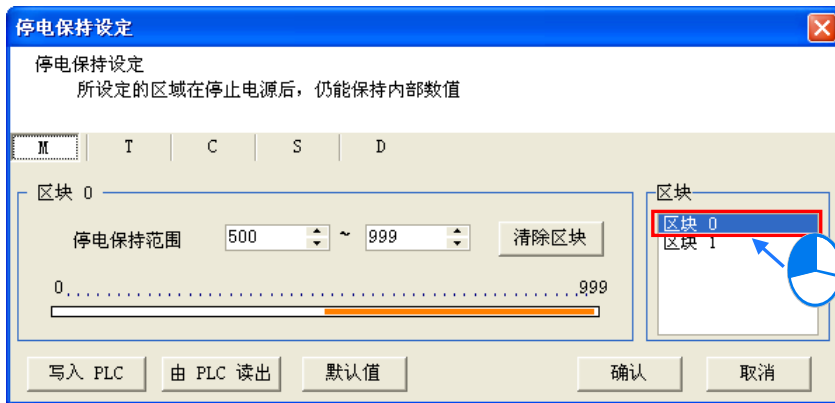
(1) 在「停电保持设定」的对象上双击鼠标左键之后，画面便会出现该机种的停电保持设定窗口。



(2) 首先用鼠标单击欲设置的装置类别。



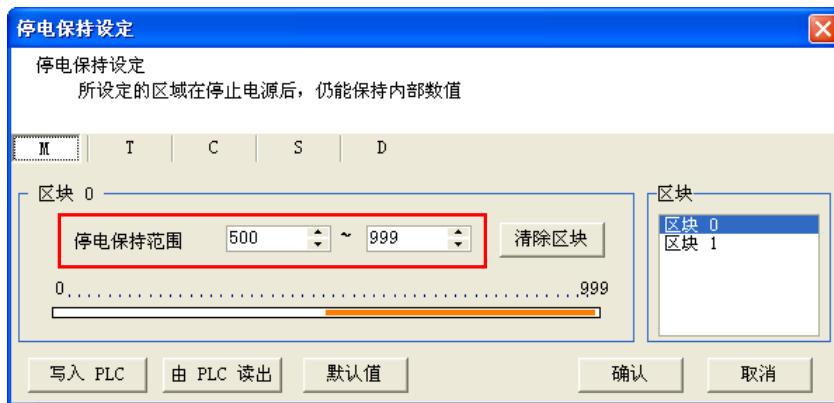
- (3) 因主机的停电保持区可能是一个不连续的范围，因此在设置前请选择欲设置的范围区块。



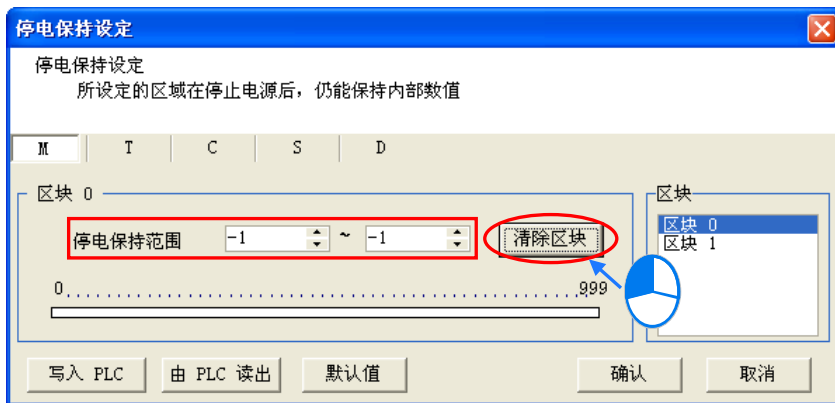
3

- (4) 在设置区域的下方会显示一个条状图，而条状图两侧的数值则代表可设置的范围极限，至于内部的橙色部份则代表已设为停电保持的范围；但若下方的设置区域显示为灰阶时，便代表目前的区块范围在所选机种中无法设为停电保持。

设置时请直接在「停电保持范围」的两个栏位中分别输入范围的起始地址，其中左侧栏位为范围起点，而右侧则为范围终点；输入设置值时，左侧起点栏位的数值必须小于右侧的终点栏位，且不可超出条状图所显示的范围极限。



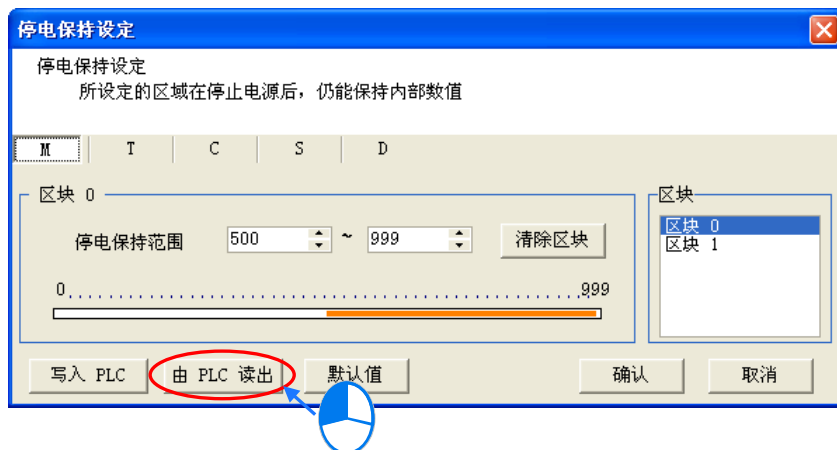
- (5) 若想让该区块的所有装置都不具备停电保持的功能时，只要单击设置区域中的「清除区块」按钮即可，而此时的范围栏位则会显示 -1 ~ -1。



- (6) 若想让设置恢复为系统默认值时，请单击「默认值」按钮即可，而此时所有装置的停电保持设置都会恢复为系统默认值。



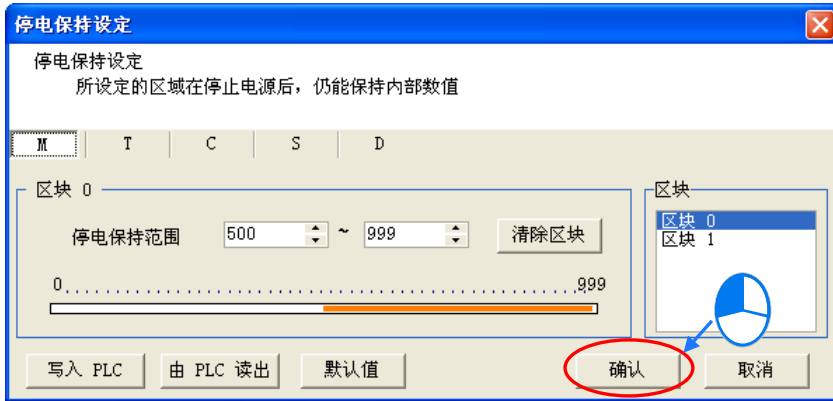
- (7) 在 ISPSOft 可与 PLC 主机正常联机的状况下，单击「由 PLC 读出」按钮便可将主机中所有装置的停电保持设置读取回来。



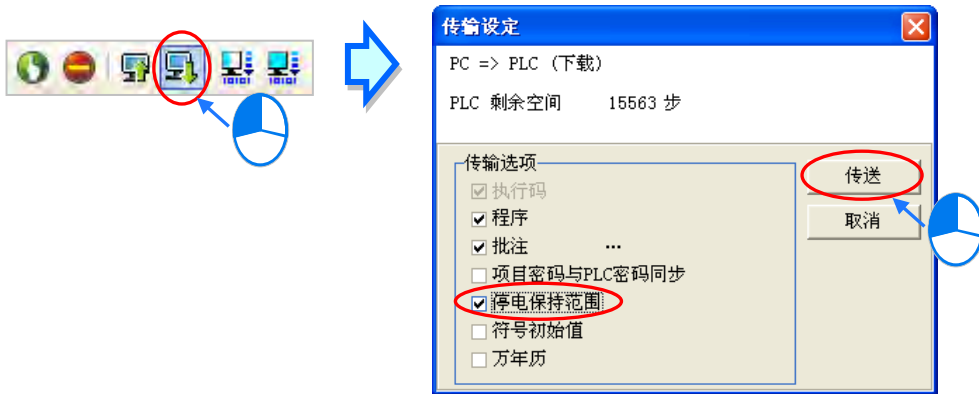
- (8) 完成所有的设置后，在 ISPSOft 可与 PLC 主机正常联机的状况下，单击「写入 PLC」按钮便可将所有装置的停电保持设置立即写入主机当中，而此时系统便会自动将设置窗口关闭并保存所有的设置值，之后当保存 ISPSOft 的项目时，此处的设置值亦会一并被保存。



- (9) 若暂时不须要将设置值写入 PLC 主机时，单击「确认」键便可应用所有的设置，而之后当保存 ISPSOft 的项目时，此处的设置值亦会一并被保存。

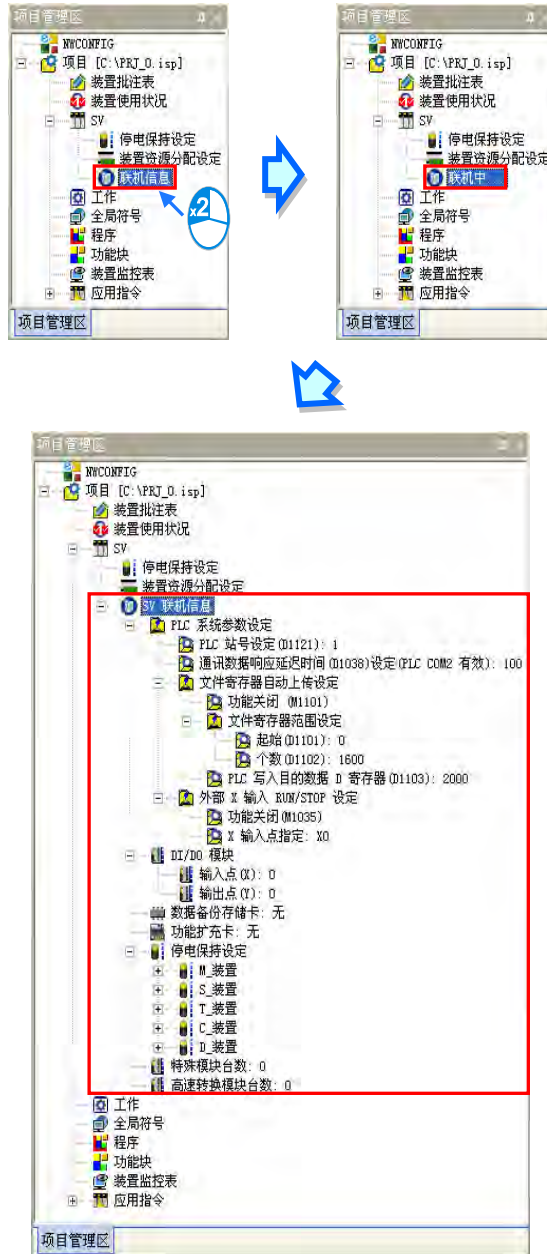


- (10) 除了可通过设置窗口的功能将设置值写入主机之外，亦可在下载项目程序时，将该项目中所设置的停电保持范围一并写入主机当中，操作时仅须将传输选项中的「停电保持范围」勾选起来，之后单击「传送」便可将数据下载至主机中。关于下载的详细操作将在第 17.1.3 节中另行介绍。



3.3.3 联机信息

在 ISPSOft 可与 PLC 主机正常联机的状况下，在项目管理区的「联机信息」项目上双击鼠标左键，之后会将 PLC 主机中的系统参数获取回来，并显示在「联机信息」项目下，此处的信息仅可检视。



3

3.3.4 DVP-ES3 机种的主机参数设置

双击项目管理区的「HWCONFIG」之后，在「参数设定」单击鼠标左键，接着系统便会自动开启 CPU 主机参数设定页面。

! 设定主机参数前，请务必查阅各机种的操作手册，并确认所设定的参数对主机本身以及整体系统的影响，以免造成系统或人员的伤害。

*.参数设定完毕后，必须下载至主机当中才可生效，下载的方式请参考第 3.4.3 节。

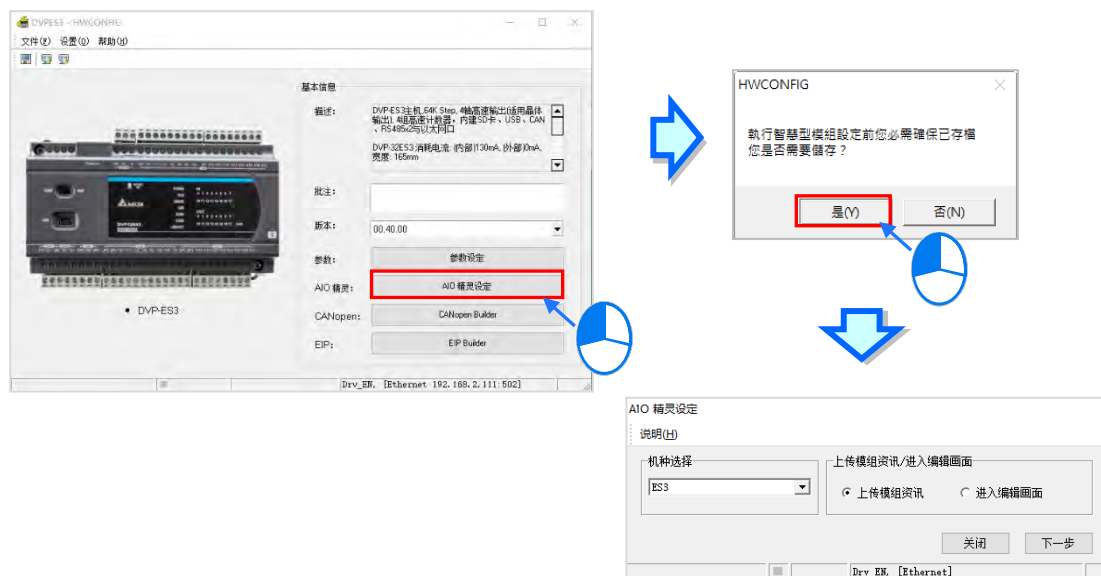
3



装置设定页面中亦根据参数类别而区分为许多分页，操作时可配合窗口上方的**主类别滚动条**以及位于每个主类别页面中页面左方的项目阶层切换不同的设定页面。页面右方标题栏依序列出**参数名称**，及供用户填入**设定值**字段，设定值的**单位**及设定值的**默认值**、**最大值**与**最小值**，各项目的详细操作方式，可以参考第 3.5.2 节。



DVP-ES3 机种需通过 AIO 精灵设定，来配置右侧模块，而这些参数亦将决定实际运作时，各个模块所表现出来的功能与特性。进行模块参数的设定时，请在「**AIO 精灵设定**」单击鼠标左键，之后系统便会开启 AIO 精灵设定，接着可以单击「**上传模块信息**」将配置由 PLC 上载回来，或选取「**进入编辑画面**」手动建立模块。如下图。



3.4 DVPxxMC 机种的 PLC 主机参数设定

3.4.1 开启 PLC 主机的参数设定页面

启动 HWCONFIG 之后，在「参数设定」单击鼠标左键，接着系统便会自动开启 CPU 主机的参数设定页面，且随着主机型号的不同，其设定的参数与显示的画面也会有所差异。

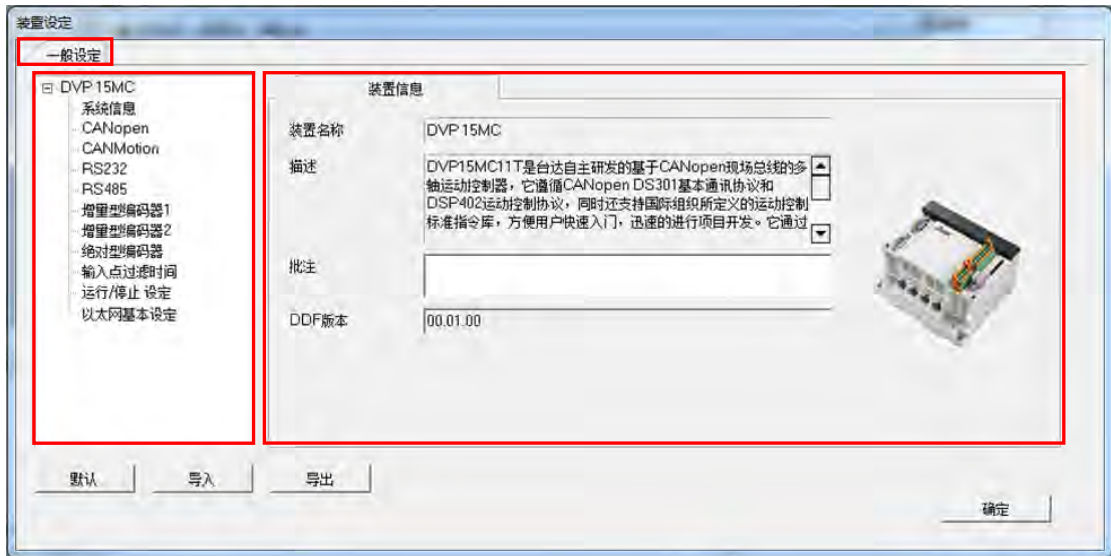
⚠ 设定主机参数前，请务必查阅各机种的操作手册，并确认所设定的参数对主机本身以及整体系统的影响，以免造成系统或人员的伤害。

3

*. 参数设定完毕之后，仍必须被下载至主机当中才可生效，下载的方式请参考第 3.4.3 节。



参数设定页面中亦根据参数类别而区分为许多的分页，操作时可配合窗口上方的主类别标签以及位于每个主类别页面中的项目阶层切换不同的设定页面。页面右方标题列依序列出**参数名称**，及供用户填入的**设定值**栏位，设定值的**单位**及设定值的**默认值**、**最大值**与**最小值**。

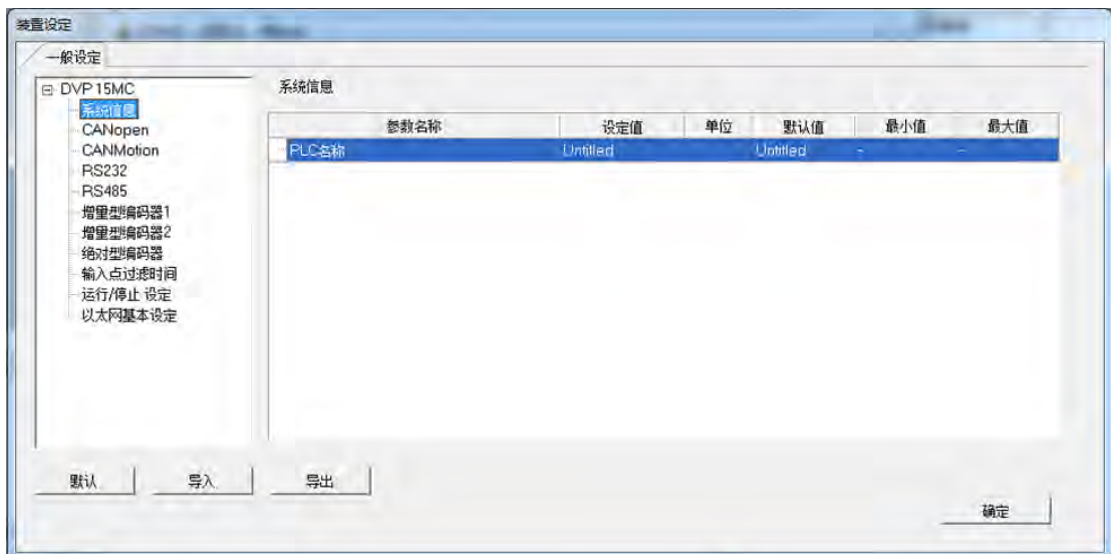


3

3.4.2 一般设定

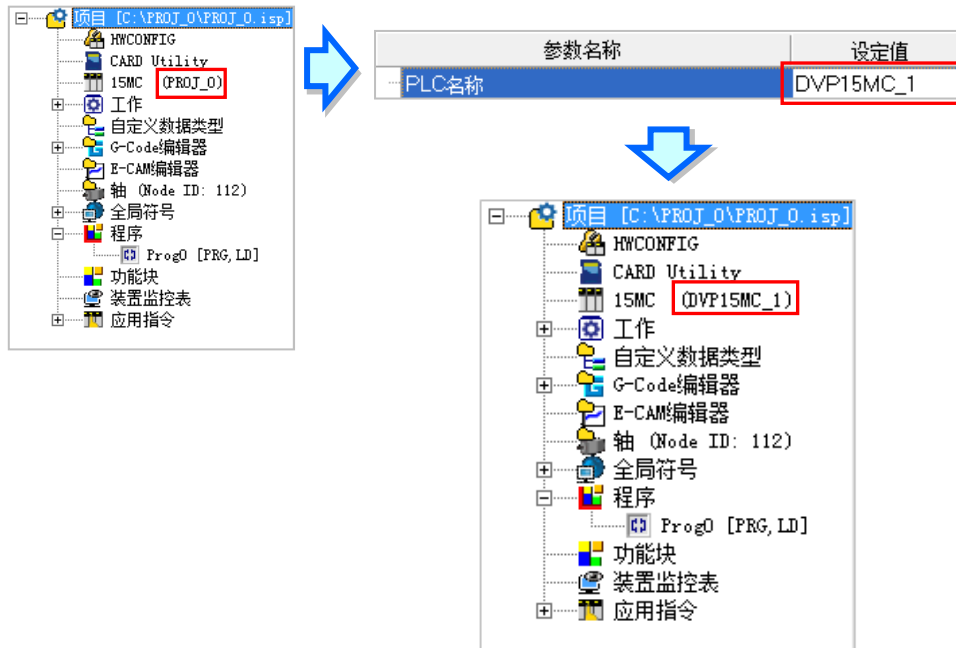
3.4.2.1 一般设定-系统信息

此类参数位于「一般设定」分页的「系统信息」项目中，此页面内只有「PLC名称」，而栏最多可输入 15 字符。且所输入的字符并没有特别的限制，包括特殊字符及空白键皆可输入，不过须注意一个中文字将会占用两个字符。



当新增一个 ISPSOft 项目时，系统便会以项目的名称作为主机的默认名称，且会显示在项目管理区的机种名称后方，而后续则可在此处重新定义。

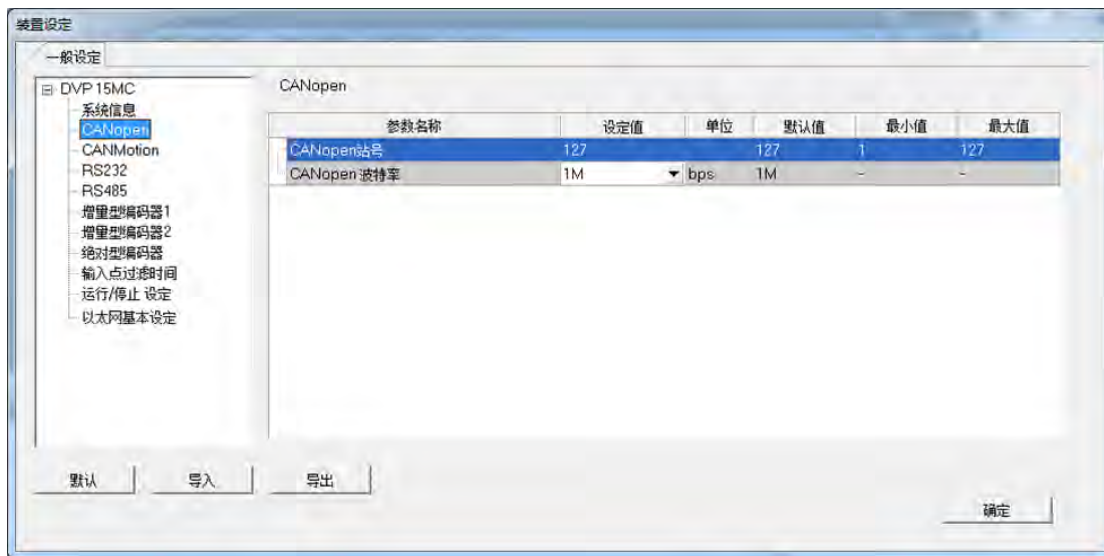
3



PLC 标签名称的功能主要是用来做为装置识别，尤其是在联网系统的应用或是在进行连线操作时，藉由确认 PLC 标签名称的方式便可防止用户的误操作；当用户要对主机进行数据的上下载或其它的连线操作时，若发现操作对象的主机标签名称与项目中的标签名称不符时，系统便会提醒用户做再次的确认，以避免对其它的主机造成非预期的影响。

3.4.2.2 一般设定-CANopen

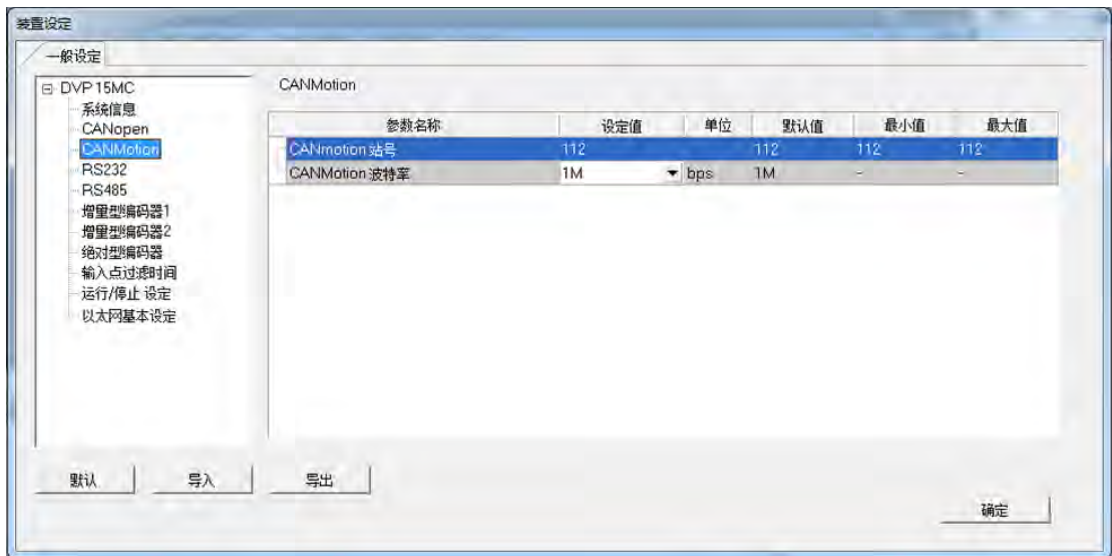
「CANopen」页面设定各项参数如下，可藉由下拉选项或输入框设定适当的参数值。



此处用作设定主机的 CANopen 通讯口站号和通讯口通讯波特率，通讯波特率可选择 20k bps、50k bps、125k bps、250k bps、500k bps、800k bps 及 1Mbps。

3.4.2.3 一般设定-CANMotion

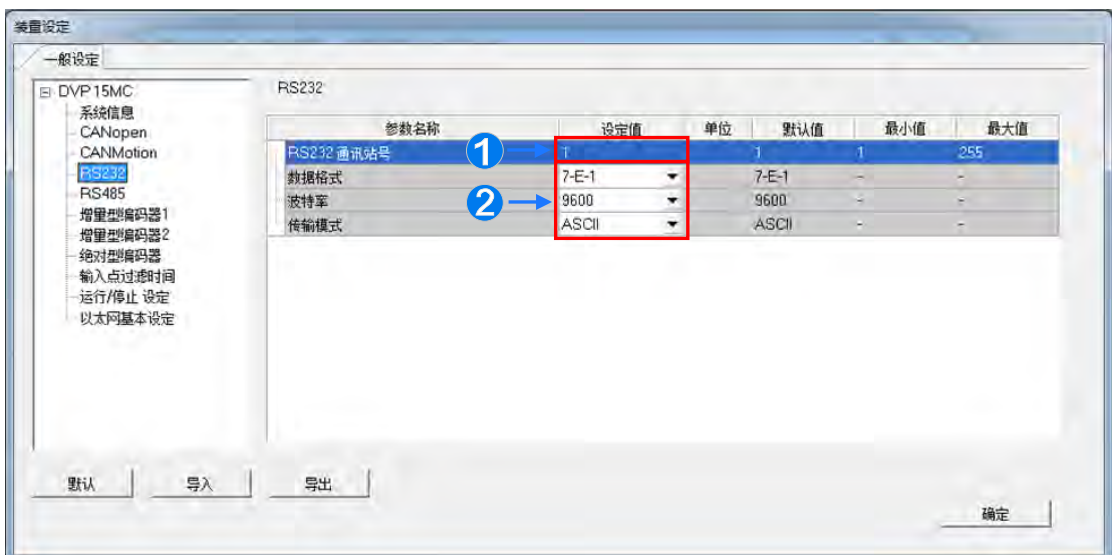
「CANMotion」页面设定各项参数如下，可藉由下拉选项或输入框设定适当的参数值。



此处用作主机的 Motion 通讯口站号，默认为 112，不可更改；但可做通讯口通讯速率设定，通讯速率可选择 1Mbps 或 500kbps。

3.4.2.4 一般设定-RS232

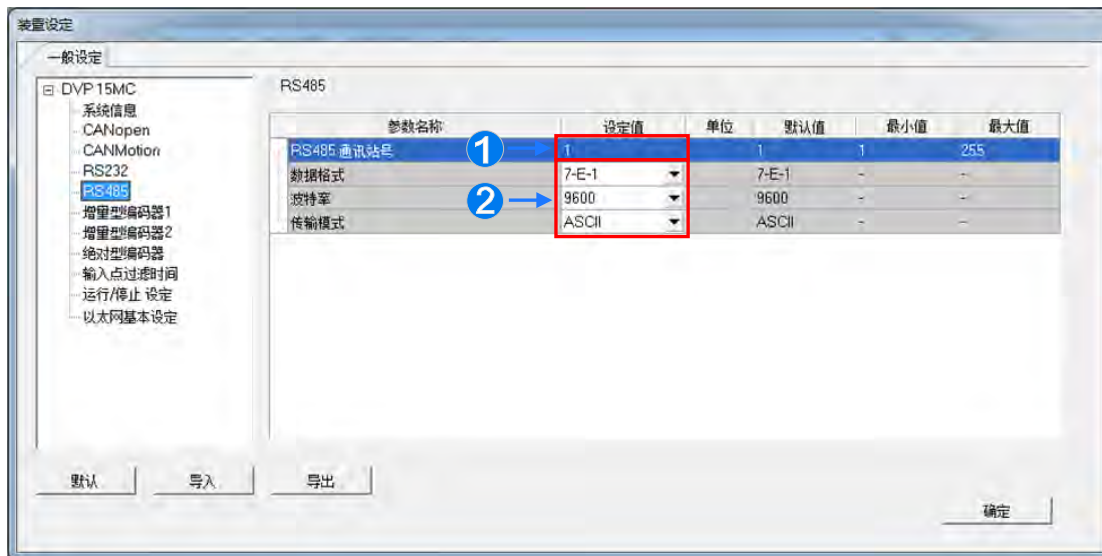
「RS232」此页面用作主机 RS232 通讯端口设定。



- ❶ 此处用作设定 RS232 通讯站号。
- ❷ 此处用作设定 RS232 数据格式、通讯速率及传输模式，而通讯速率有 9600、19200、38400、57600 及 115200 可选择。

3.4.2.5 一般设定- RS485

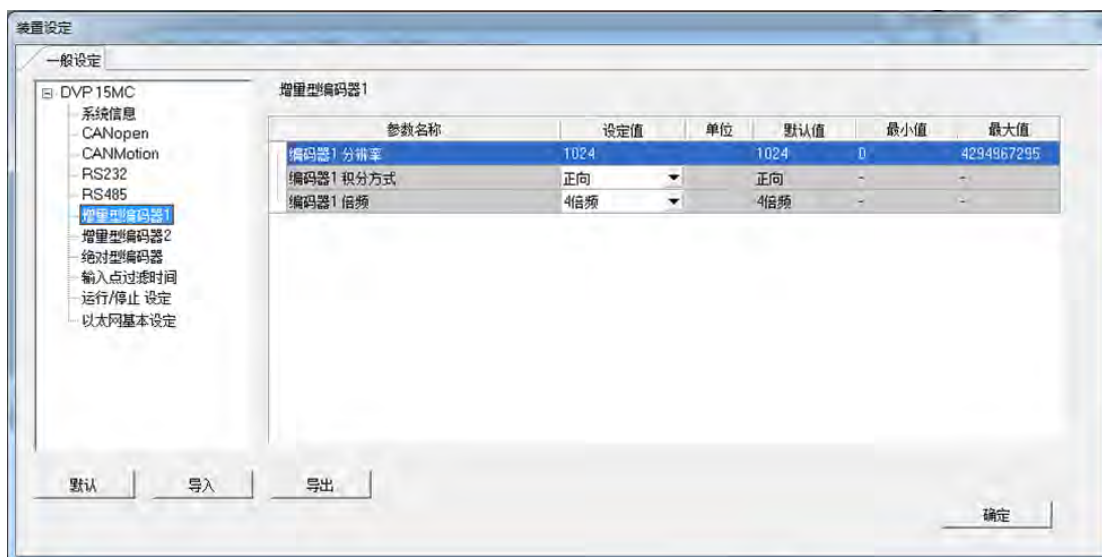
「RS485」此页面用作主机 RS485 通讯端口设定。



- ① 此处用作设定 RS485 通讯站号。
- ② 此处用作设定 RS485 数据格式、通讯速率及传输模式，而通讯速率有 9600、19200、38400、57600 及 115200 可选择。

3.4.2.6 一般设定- 增量型编码器 1

「增量型编码器 1」此页面用作连接第一组增量型编码器的设定。



此处用作设定增量型编码器 1 解析度、积分方式及倍频。

➤ 增量型编码器 1 解析度

此处用作「增量型编码器 1 解析度」之设定，默认为 1024。

➤ 增量型编码器 1 积分方式

此处提供「反向」及「正向」之选择。

➤ 增量型编码器 1 倍频

此处提供「1 倍频」、「2 倍频」及「4 倍频」之选择。

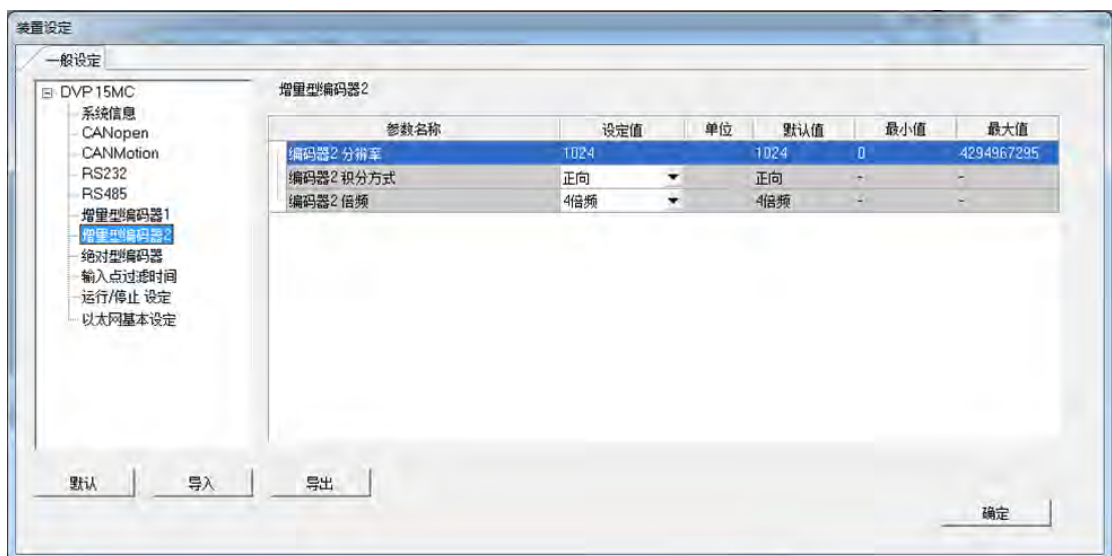
而增量型编码器的积分方式及倍频具体说明如下表。

积分方式	含义	计数方式
正向 1 倍频	检测到 B 相上升沿时计数	增计数
正向 2 倍频	检测到 B 相上升沿时和下降沿时均计数	增计数
正向 4 倍频	检测到 A 相和 B 相上升沿时和下降沿时均计数	增计数
反向 1 倍频	检测到 B 相上升沿时计数	减计数
反向 2 倍频	检测到 B 相上升沿时和下降沿时均计数	减计数
反向 4 倍频	检测到 A 相和 B 相上升沿时和下降沿时均计数	减计数

3

3.4.2.7 一般设定- 增量型编码器 2

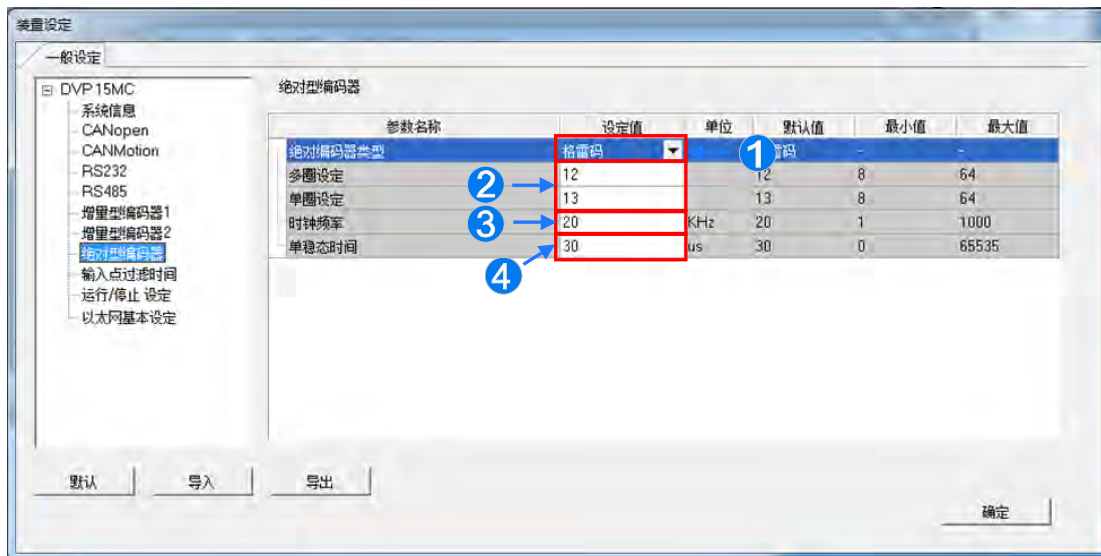
「增量型编码器 2」此页面用作连接第二组增量型编码器的设定。



此处用作设定增量型编码器 2 解析度、积分方式及倍频，其设定方式同增量型编码器 1，可参考第 3.4.2.6 节。

3.4.2.8 一般设定- 绝对型编码器

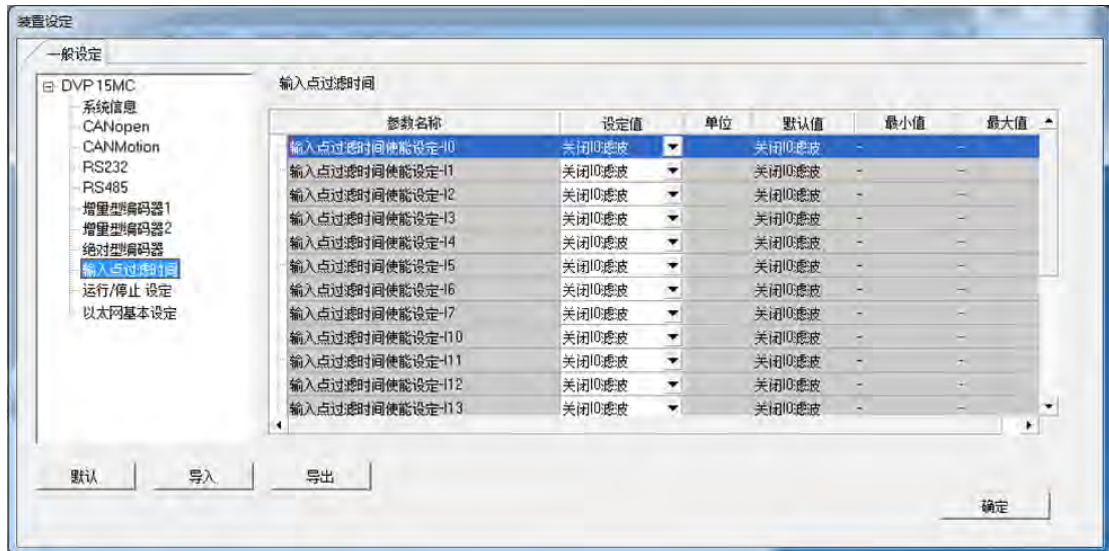
「绝对型编码器」此页面用作连接绝对型编码器的设定。



- ❶ 此项目为选择绝对型编码器的输出为「二进制」或「格雷码」，此参数由编码器决定。
- ❷ 此项目的「多圈设定」为用作设定多圈解析度，绝对型编码器脉冲数超过允许的最大圈后重新计数，若设定为 N 则表示可以转的圈数为 2 的 N 次方，此参数由编码器决定；而「单圈设定」项目用作设定单圈解析度，绝对型编码器转一圈产生的脉冲数，若设定为 N 则表示单圈产生的脉冲数为 2 的 N 次方，此参数由编码器决定。
- ❸ 「时间频率」项目为设定数据传输速率。
- ❹ 「单稳态时间」项目，单稳态时间由时钟序列最后一个脉冲下降沿开始计时，它决定了最低传输频率

3.4.2.9 一般设定- 输入点过滤时间

「输入点过滤时间」此页面用于设定 DVPxxMC 输入点过滤设定和 IO 刷新跟随 Task 的机制。



3

➤ 输入点过滤时间使能设定-I0~I7

此项滤波器为 I0~I7 这些点对短脉冲信号的过滤，从而降低输入干扰信号的影响，通过增大滤波值，可减少输入信号的抖动或外部干扰的影响。可由下拉选项，选择开启或关闭 IO 滤波。

➤ 输入点过滤时间使能设定-I10~I17

此项滤波器为 I10~I17 这些点对短脉冲信号的过滤，从而降低输入干扰信号的影响，通过增大滤波值，可减少输入信号的抖动或外部干扰的影响。可由下拉选项，选择开启或关闭 IO 滤波。

➤ I0~I7 过滤时间因数

此处为设定 I0~I7 过滤时间因数 0~255，过滤时间： $t=31\mu s * (\text{过滤时间因数})$ 。因此过滤时间是 31us 的倍数，过滤时间因数默认值为 0。

➤ I10~I13 过滤时间因数

此处为设定 I10~I13 过滤时间因数 0~255，过滤时间： $t=31\mu s * (\text{过滤时间因数})$ 。因此过滤时间是 31us 的倍数，过滤时间因数默认值为 0。

➤ I14~I17 过滤时间因数

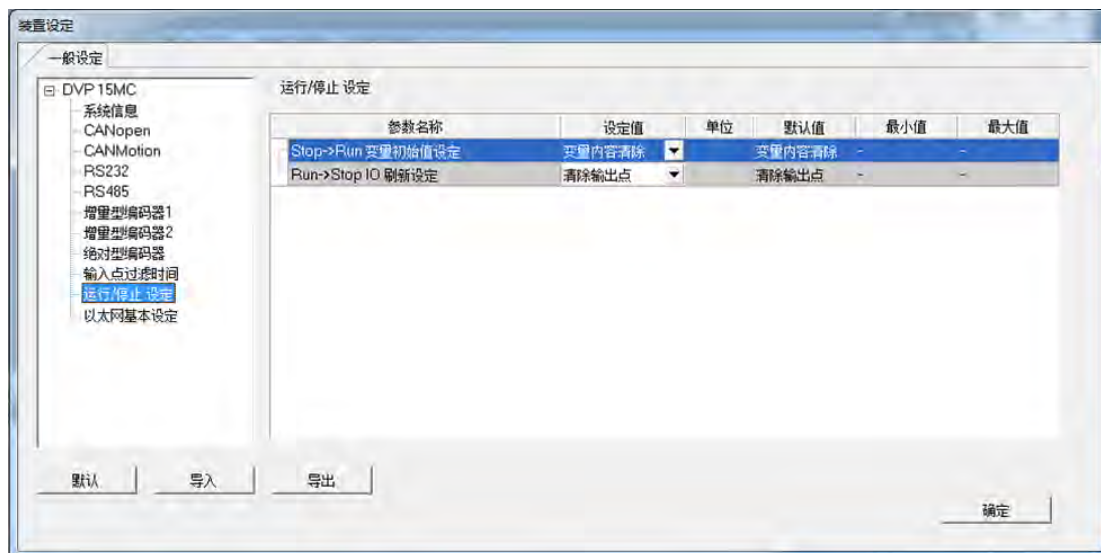
此处为设定 I14~I17 过滤时间因数 0~255，过滤时间： $t=31\mu s * (\text{过滤时间因数})$ 。因此过滤时间是 31us 的倍数，过滤时间因数默认值为 0。

➤ IO 刷新跟随 Task

I/O 包括 DVPxxMC 本身 I/O 点和左右扩展模组数据、CANopen 数据。选择「FF」则 IO 数据会在系统处理中刷新；选择其它指定 Task 则 IO 数据在该指定 Task 执行前更新。

3.4.2.10 一般设定- 运转 / 停止设定

「运转 / 停止」此页面用于设定 IO 是否刷新以及变数装置是否恢复初始值。



➤ Stop → Run 变数初始值设定

此项用可选择「变数内容清除」或「变数内容保持」。

- (1) 当选择「变数内容清除」，由停止到运行时，非停电保持型变数和非停电保持装置的值会恢复为初始值或默认值，停电保持型的变数 (VAR_RETAIN 型) 和停电保持区的装置并不会恢复初始值。

非停电保持型的变数绑定停电保持装置时，若该变数设定初始值，则当由停止状态变为运行状态时或重上电后，变数值会恢复为初始值；若该变数未设定初始值，则由停止状态变为运行状态时或重上电后，该变数的值会变为绑定装置中存储的值。

- (2) 当选择「变数内容保持」，由停止到运行时，变数和装置会保持停止前的值。软体重新下载时，由停止到运行时，非停电保持型变数和非停电保持区装置的值会恢复为初始值或默认值；停电保持型变数 (VAR_RETAIN 型) 和保持区装置不会初始化。

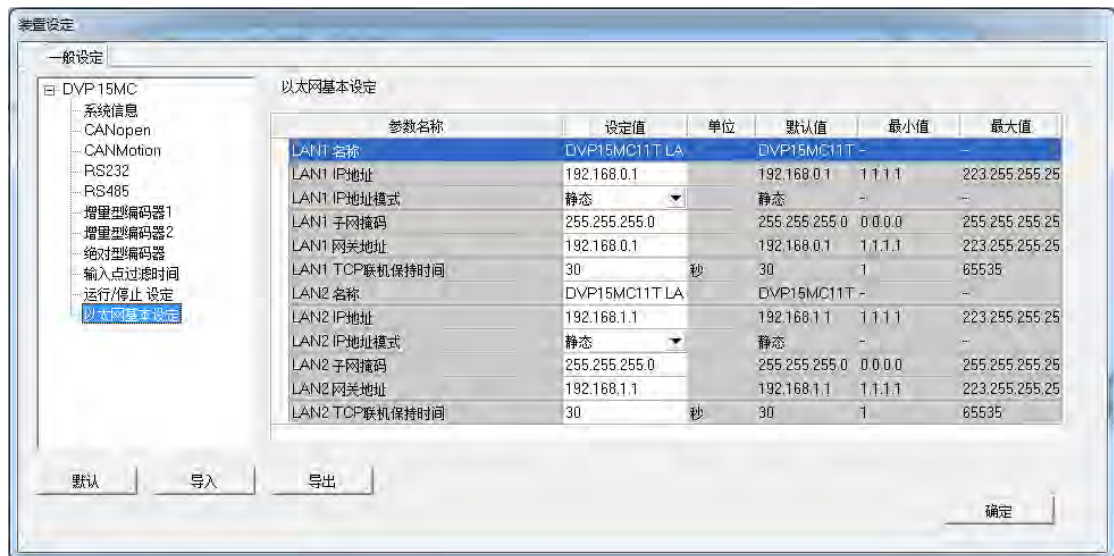
➤ Run → Stop IO 刷新设定

此项用作主机在从运行到停止时，IO 刷新的设定，可选择「清除输出点」或「保持输出点」。

- (1) 当选择「清除输出点」，当处于停止状态时，会删除当前 IO 数据，并且不在刷新 IO 数据。
- (2) 当选择「保持输出点」，由运行到停止时，不会删除当前 IO 数据，并且在处于停止状态时仍然保持 IO 刷新。

3.4.2.11 一般设定- 以太网基本设定

于项目列表中单击「以太网基本设定」项目即可进入。主要用于设定主机的两个以太网网络 (Ethernet) 端口的通讯参数。15MC 支持 LAN1/LAN2，50MC 只有 LAN1 另一个 Port 为 EtherCAT。




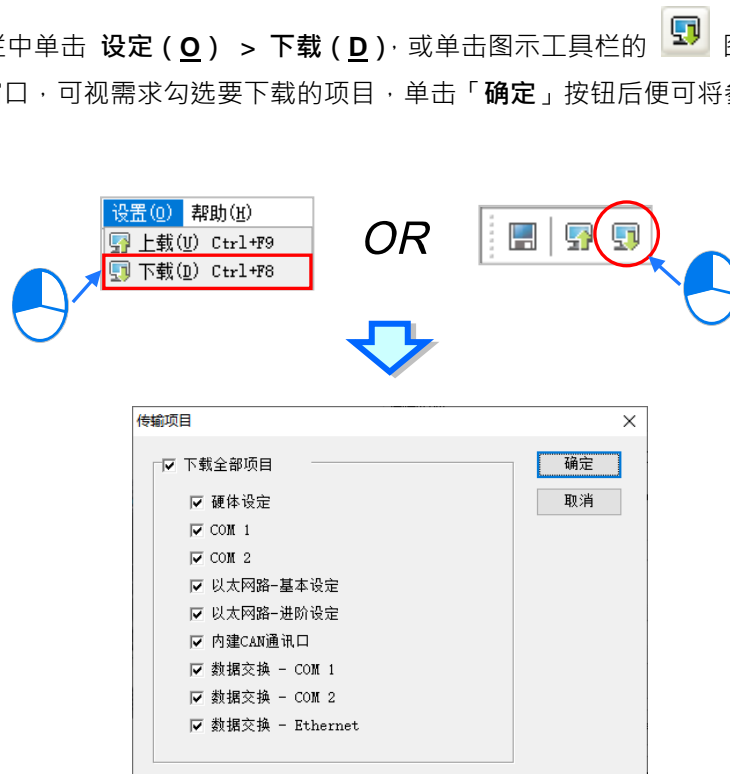
3

3.4.3 上下载


于 HWCONFIG 当中所做的设定，必须被下载至主机中才可正式生效，且因 HWCONFIG 会直接引用 ISPSOft 的通信设置，因此进行参数上下载之前，请务必确认 ISPSOft 与主机正常联机。执行时，若主机目前的卷标名称与参数设定的名称不符时，画面亦会出现提示信息。

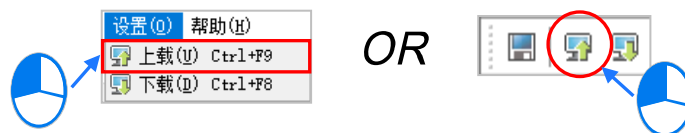
➤ 下载系统硬件参数

于菜单工具栏中单击 **设定 (O) > 下载 (D)**，或单击图示工具栏的  图标，之后便会出现下载项目的确认窗口，可视需求勾选要下载的项目，单击「确定」按钮后便可将参数下载至主机中。



➤ 上传系统硬件参数

于菜单工具栏中单击 **设定 (O) > 上传 (U)**，或单击图示工具栏的  图标，之后即可将主机中 PLC 主机参数上传回 HWCONFIG 的设定画面中。



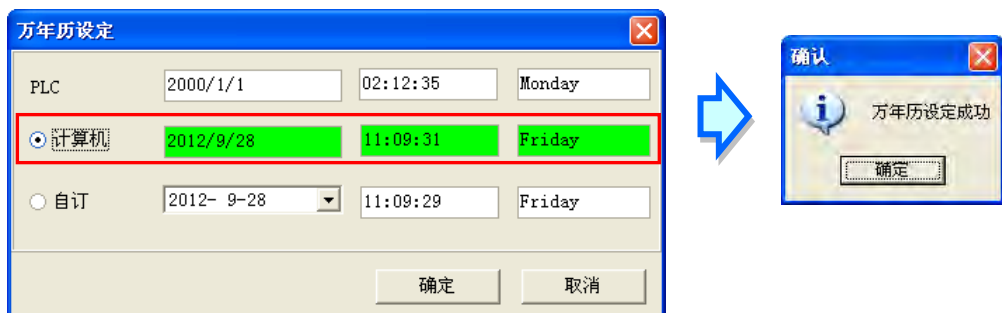
3.5 万年历设定

PLC 主机中的万年历信息可通过 ISPSOft 所提供的工具来进行设置，而设置之前，同样必须先确认 ISPSOft 已可与主机正常联机，详细的设置与操作方式请参考第 2.4 节当中的内容。

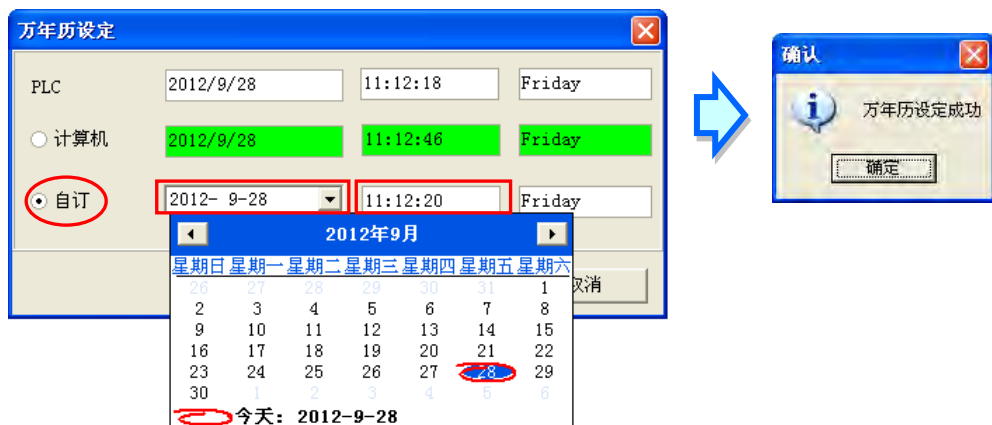
- (1) 在菜单工具栏中单击 **工具(I)** > **万年历设定**，之后系统便会出现万年历的设置窗口，其中最上方的日期时间为窗口打开时由 PLC 内部所获取出来的信息。



- (2) 当要使用计算机目前的时间做为万年历的设置值时，选择「计算机」选项后单击「确定」键即可。



- (3) 若要自订万年历的时间，则请先选择「自订」选项后，分别在日期与时间栏位中输入欲设置的值，而其中单击日期栏位右侧的按钮便会出现辅助工具；完成设置后单击「确定」键即可。

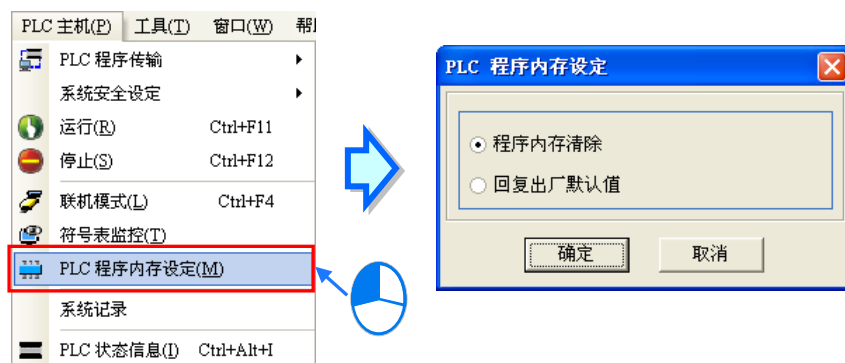


3.6 PLC 内存设定 (清除程序&恢复出厂值)

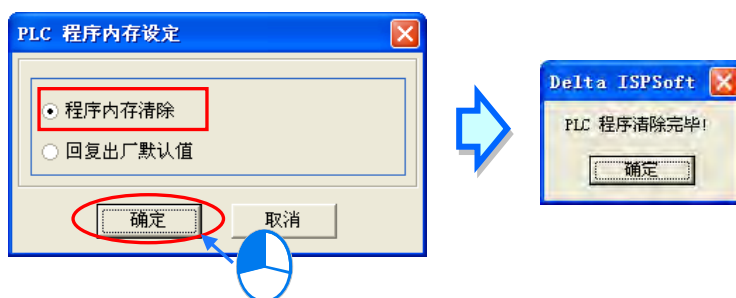
通过 ISPSOft，用户可将 PLC 主机中的程序内存清除，或是直接将 PLC 恢复为出厂默认值，而同样的，操作前请先确认 ISPSOft 已可与主机正常联机，详细的设置与操作方式请参考第 2.4 节。

(1) 在菜单工具栏中单击 **PLC 主机 (P)** > **PLC 程序内存设定 (M)**，之后便会出现操作窗口。

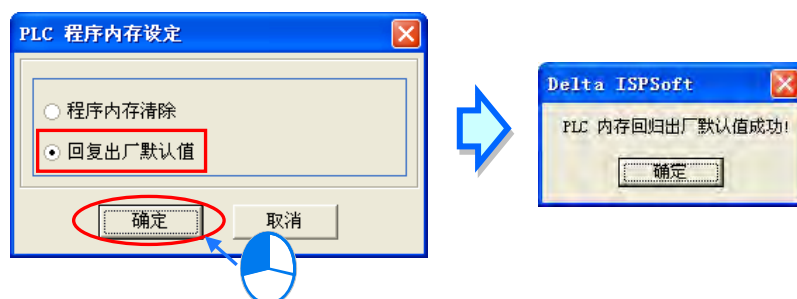
3



(2) 选择「程序内存清除」后单击「确定」，系统便会将主机内部的程序内存数据全部清除。



(3) 若选择「回复出厂默认值」，则主机内部的所有数据与状态都会在确认执行之后被恢复为出厂默认值。而依据机种的不同，完成后可能必须将 PLC 重新上电才可继续之后的操作。



若连接的是 AH 系列机种，则必须先将 PLC 主机参数中的「远端重置」功能启动并完成参数的下载，之后才允许通过 ISPSOft 将主机恢复为出厂默认值。详细说明请参考第 3.3 与 3.4 节当中的内容。

4

第4章 快速入门

目录

4.1 快速入门	4-2
4.1.1 范例功能说明	4-2
4.1.2 硬件规划	4-2
4.1.3 程序规划	4-3
4.2 ISPSOft 的项目开发程序	4-3
4.3 新增项目	4-4
4.4 硬件组态	4-5
4.4.1 模块配置	4-5
4.4.2 主机与模块参数设置	4-7
4.5 建立程序	4-10
4.5.1 新增梯形图程序	4-10
4.5.2 基本编辑 - 建立接点与线圈	4-12
4.5.3 基本编辑 - 梯形图区段的新增/插入与指令编辑模式	4-15
4.5.4 基本编辑 - 梯形图区段的选取与操作	4-17
4.5.5 基本编辑 - 并联接点装置	4-20
4.5.6 基本编辑 - 编辑批注	4-21
4.5.7 基本编辑 - 插入 API 应用指令	4-22
4.5.8 基本编辑 - 比较接点的建立与常数的输入格式	4-24
4.5.9 完成范例程序	4-25
4.5.10 程序的检查与编译	4-26
4.6 测试与除错	4-27
4.6.1 建立联机	4-27
4.6.2 下载程序与组态参数	4-30
4.6.3 联机测试	4-33

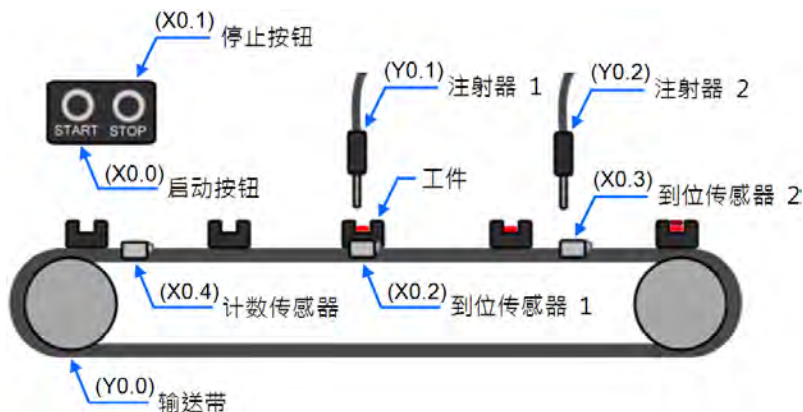
4.1 快速入门

本章将以一个简单的范例，带领用户于 ISPSOft 的开发环境中，快速建立一个传统的梯形图程序；不过在本章中尚不会导入 IEC61131-3 的相关编程观念，如 POU、功能块 (FB)、变量符号...等，目的是为了尚不熟悉 IEC61131-3 的用户同样能快速地了解 ISPSOft 所提供的功能，并得以了解如何建立传统的梯形图程序。至于 IEC61131-3 的相关观念亦将于后续的章节当中陆续介绍。

4.1.1 范例功能说明

设备开始运转时，工件会在输送带上持续地由左至右传送，当传感器检测到工件位于注射器的下方时，PLC 便会送出一触发信号给注射装置，此时注射器便会开始注射胶体，而注射时间将会由外部进行设置，在 PLC 的程序中则无须加以控制，但设计上必须能让触发信号恢复至 OFF 状态，以便进行下次的触发。输送带上共有两站的点胶动作，且两站的程序完全相同。

另外，在输送带的左侧会安置一传感器，当工件通过时，计数值便会加 1，当计数值到达 100 时，内部的完成标志便会被设置为 ON，而该标志的状态则可于后续供其它程序运用，本范例将暂不提及。



4.1.2 硬件规划

在本范例中，我们将选用 AH5x 系列主机 **AHCPU530-EN**，数字 IO 模块则选用 8 点输入/8 点输出的 **AH16AP11R-5A**，而主背板则选用 4 槽的 **AHBP04M1-5A**。下表即为本范例的 IO 配置表。

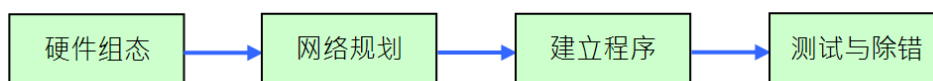
型式	装置地址	说明
DI	X0.0	启动按钮
DI	X0.1	停止按钮
DI	X0.2	到位传感器 1
DI	X0.3	到位传感器 2
DI	X0.4	计数传感器
DO	Y0.0	输送带
DO	Y0.1	注射器 1 触发信号
DO	Y0.2	注射器 2 触发信号

4.1.3 程序规划

- (1) 当**启动按钮** (X0.0) 由 OFF 变为 ON 时，将内部的运转标志设为 ON，并让**输送带** (Y0.0) 开始持续运转；而当**停止按钮** (X0.1) 由 OFF 变为 ON 或侦测到错误发生 (错误标志为 ON) 时，则将运转标志重置为 OFF，且让所有的动作停止。
- (2) 当**到位传感器 1** (X0.2) 为 ON 时，便将**注射器 1 触发信号** (Y0.1) 也设置为 ON；而当传感器变为 OFF 时，便将该信号重置回 OFF。
- (3) 当**到位传感器 2** (X0.3) 为 ON 时，便将**注射器 2 触发信号** (Y0.2) 也设置为 ON；而当传感器变为 OFF 时，便将该信号重置回 OFF。
- (4) 每当**计数传感器** (X0.4) 由 OFF 变为 ON 时，便将内部的计数值加 1；而当计数值大于或等于 100 时，则将内部的完成标志设置为 ON。

4.2 ISPSoft 的项目开发程序

下图为一般在 ISPSoft 中进行项目开发时的建议程序，不过用户仍可依据实际的应用情形与本身的使用习惯加以调整，相关说明如下。



● 硬件组态

设置 PLC 主机的相关参数，如停电保持区、通讯端口站号...等，而 AH500 系列的机种还须针对所搭载的模块进行组态配置与参数设置等工作。

● 网络规划

当所应用的系统有运用网络架构或是装置之间的数据交换时，透过 ISPSoft 本身所提供的网络规划工具 - **NWCONFIG**，用户便可轻易进行网络部署、PLC Link、Ether Link...等网络组态的规划。

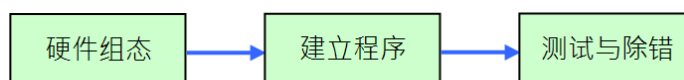
● 建立程序

于 ISPSoft 的程序编辑器上撰写控制程序，并在撰写完成之后进行编译的工作；而当编译产生错误时，利用**编译讯息区**的引导功能，用户便可快速移动至产生错误的位置以进行程序代码的确认。


● 测试与除错

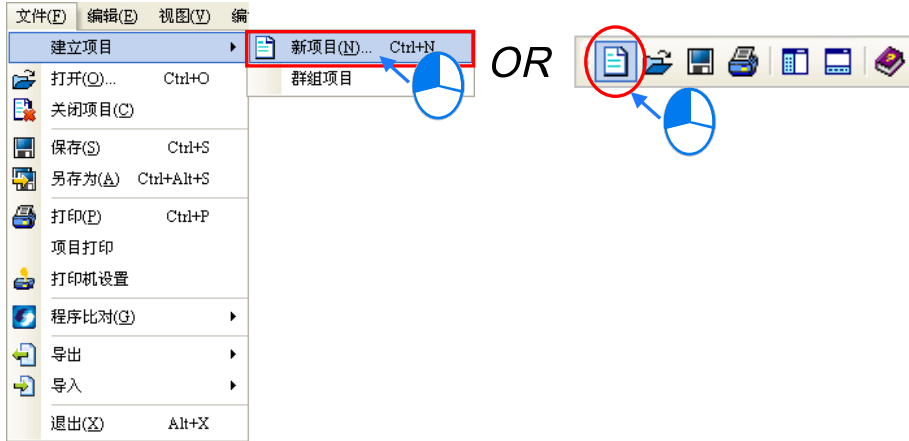
将编译完成的程序、硬件与网络的组态参数下载至 PLC，并利用 ISPSoft 所提供的各种在线监控功能来进行测试与除错的工作。

针对本章所介绍的范例，因不包含网络架构的部份，因此仅须进行如下的工作，而各工作的建构方式与操作步骤，我们亦将于后续的几个小节当中逐一说明。



4.3 新增项目


启动 ISPSOft 后，于工具栏中点选 **文件 (F) > 建立项目 > 新项目 (N)** 或直接单击  图标。

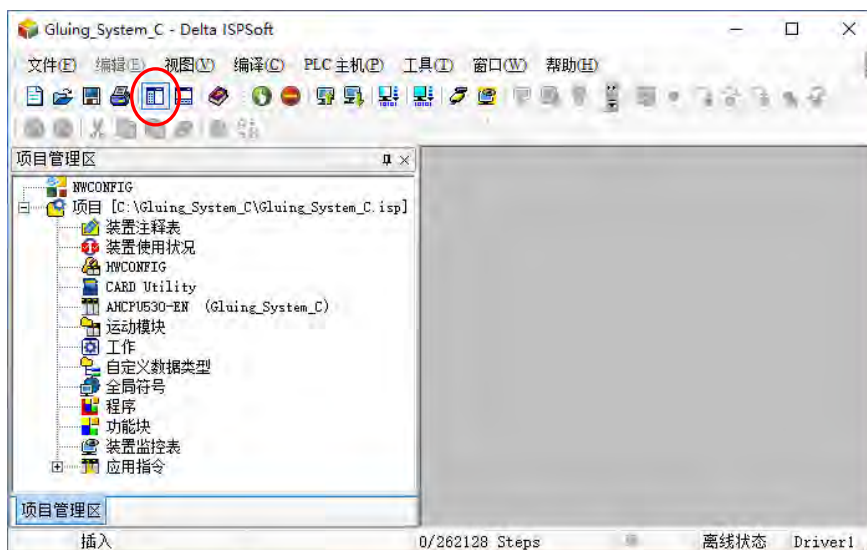


4

接着于项目建立窗口中依序输入「项目名称」及「文件路径」，并于「控制器种类」与「机种」的下拉选单中选择正确的机种型式，完成后按下「确定」。(本范例中的机种型式为 AHCPU530-EN)

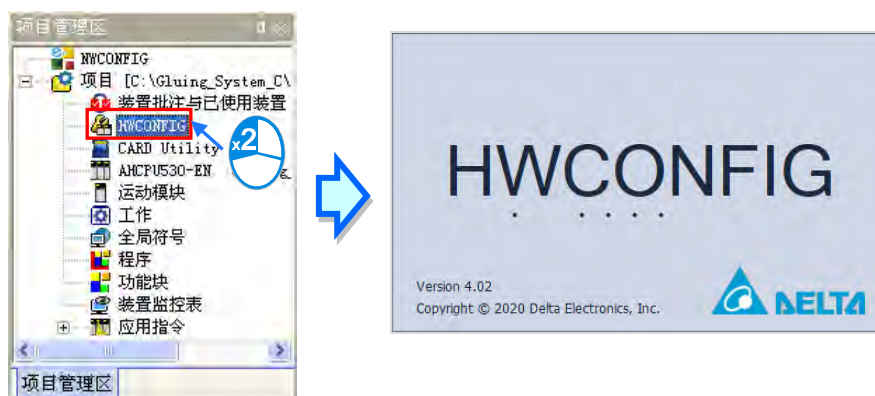


成功建立项目后，画面的左侧便会打开一个「项目管理区」，并以阶层树形图的方式列出所有对象；若未出现该区域时，请单击菜单工具栏中的 **视图 (V) > 工作区 (W)** 或单击  图标即可。



4.4 硬件组态

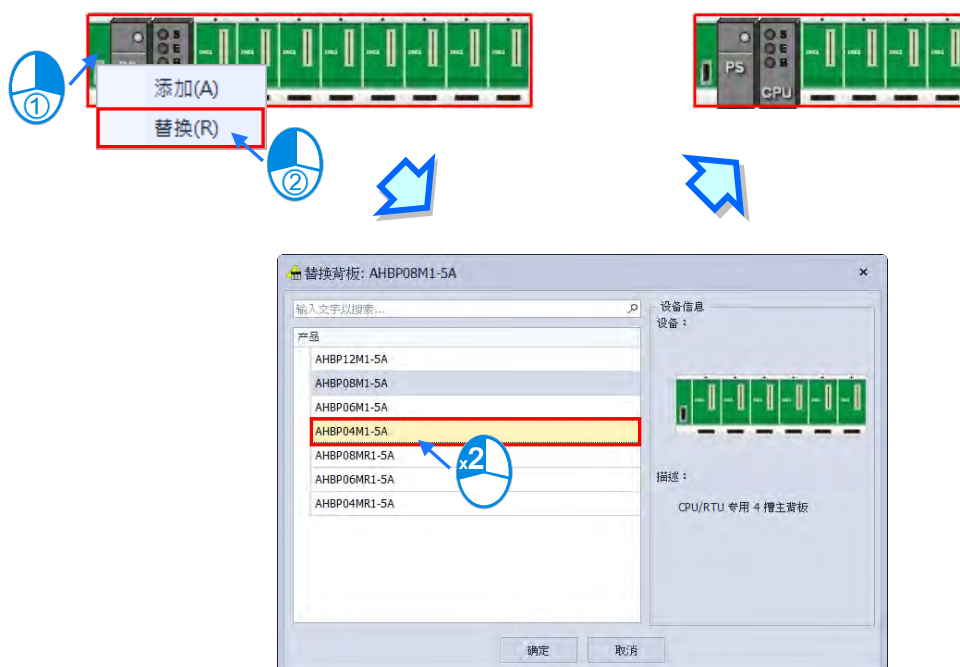
项目管理区中的「HWCONFIG」上双击鼠标左键即可启动硬件规划工具。



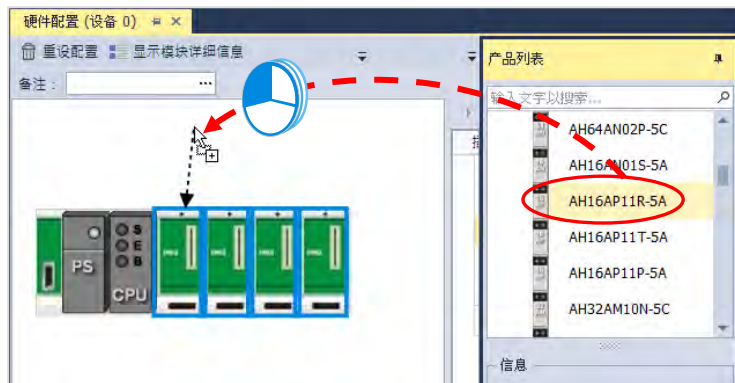
4.4.1 模块配置

初次进入 HWCONFIG 的操作画面时，默认的配置只有搭载 CPU 与电源模块的一个 8 槽背板。但在本范例的规划中，背板应为 4 槽的 **AHBP04M1-5A**，并须搭配 **AH16AP11R-5A** 的数字 IO 模块。

欲变更背板时，请先于系统配置区的背板图标左侧单击鼠标右键，并于快捷选单中点选「替换 (R)」，之后再于背板选择窗口中的「**AHBP04M1-5A**」项目上双击鼠标左键，如此便可完成背板的变更。



接着请展开左侧「产品列表」的「数字 IO 模块」，并于该目录下找到 **AH16AP11R-5A** 的项目，之后直接将该项目拖曳至系统配置区的背板插槽上即可。而成功新增一模块后，于下方表格的对应位置中便会出现该模块的项目，并列出相关的信息与地址配置。

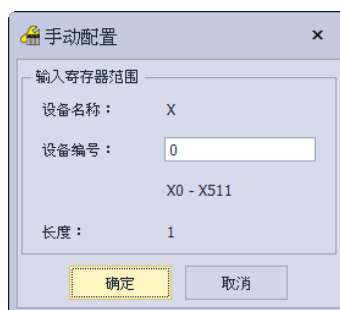


4



插槽	型号	输入设备范围	输出设备范围
电源	AHPS05-5A	NONE	NONE
CPU	AHCPU530-EN	NONE	NONE
I/O 0	AH16AP11R-5A	X0.0 - X0.7 ...	Y0.0 - Y0.7 ...
I/O 1		NONE	NONE
I/O 2		NONE	NONE
I/O 3		NONE	NONE

HWCONFIG 会自动为每个加入的模块配置对应的装置地址，而当配置的地址与规划不符时，于表格中对应该模块的地址字段上单击鼠标左键，之后再单击字段旁的 **...** 键即可自行指定地址。



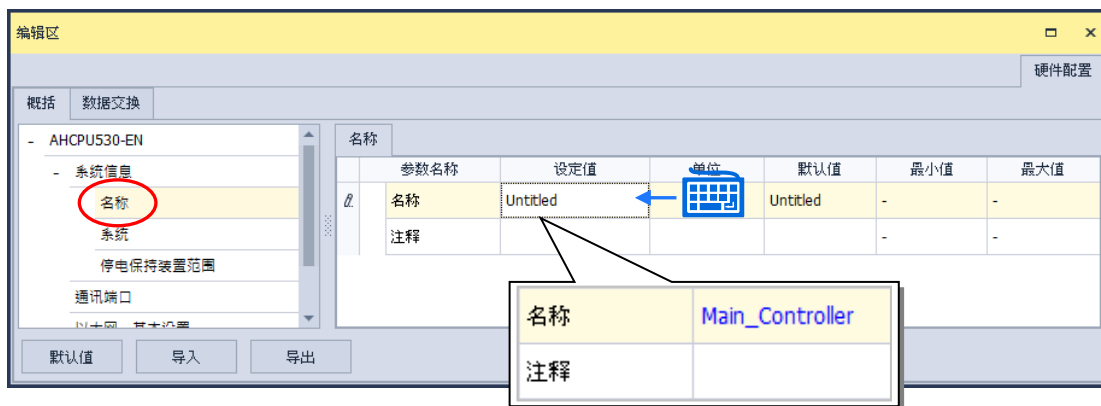
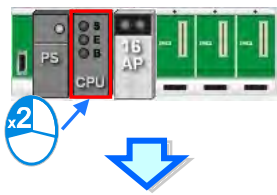
4

4.4.2 主机与模块参数设置

完成模块配置后，接着便可进一步针对 PLC 主机与扩充模块进行各别的参数设置，操作时仅须于欲设置参数的主机或模块上双击鼠标左键即可打开对应的设置窗口。

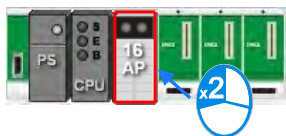
首先于 CPU 的图标上双击鼠标左键，之后即可打开主机的参数设置窗口。于设置窗口中，我们可以透过左侧树状标识切换各个设置页，并可在各个页面中设置 PLC 主机的相关参数；而在本范例中，我们仅须重新定义 PLC 主机的标签名称即可。

PLC 标签名称的参数位于「系统信息」中的「名称」副类别页面中，且通常在第一次打开 PLC 主机的参数设置窗口时便会进入该页面；进入该页面后，我们可发现在「名称」的字段中已存在一个与项目名称相同的默认名称，接着请直接将其修改为「Main_Controller」后，按下「确定」键离开即可。

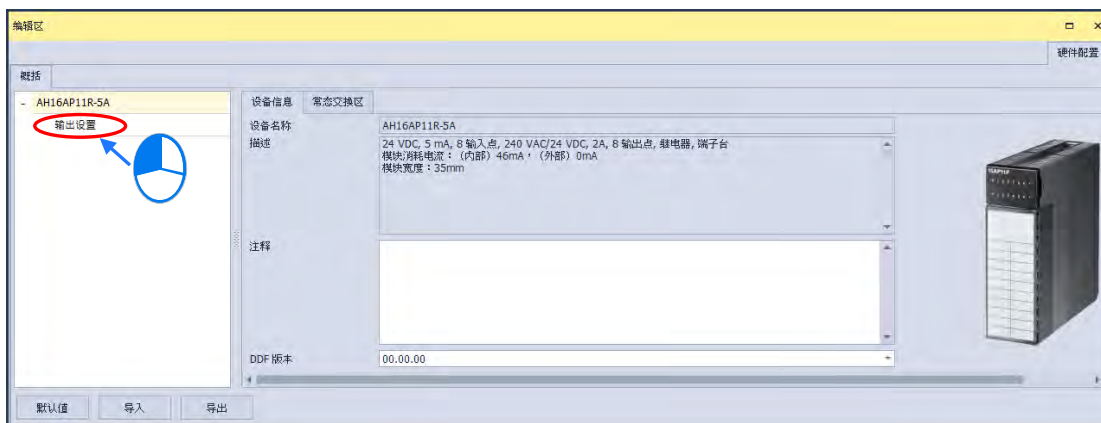


4

接下来请于配置的 IO 模块图标上双击鼠标左键，之后即可打开设置窗口。



进入模块的参数设置窗口后，主画面中会显示该模块的相关信息，而要开始设置参数时，请先于窗口左侧的列表中单击欲设置的参数类别，之后即可于参数表格中设置相关的参数。在本范例中，我们只须沿用默认值而无须做任何变更，所以此步骤可直接略过。



4

完成主机及模块的参数设置后，第一阶段的硬件规划工作便已完成，不过后续仍须将所做的配置与设置下载至主机后才可生效；在此我们可先将其进行保存，而待之后下载项目程序时再一并下载即可。欲进行保存时，请直接于菜单工具栏中单击 **文件 (E) > 保存 (S)**，或使用快捷键 **Ctrl+S**。完成之后便可关闭 **HWCONFIG** 的窗口。



*.更详细的 **HWCONFIG** 操作方式，请参考第 3 章的相关说明。

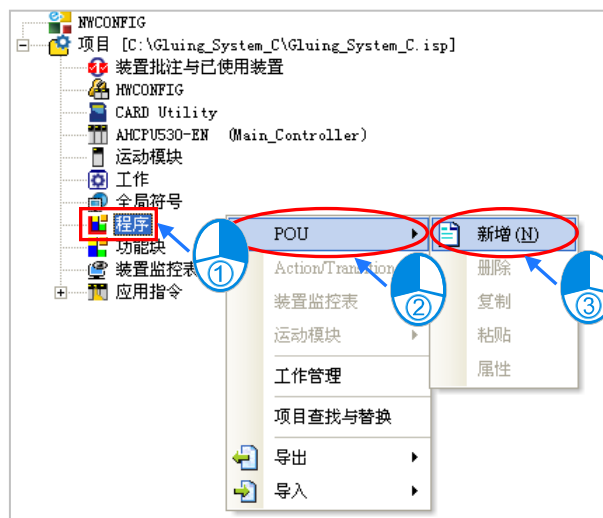
4.5 建立程序

在接下来的内容中，我们将逐步示范如何在 ISPSOft 当中完成一个传统的梯形图程序，内容包括建立程序对象、梯形图的基本编辑、程序编译...等，并期待可让用户快速上手且具备基本的操作能力。

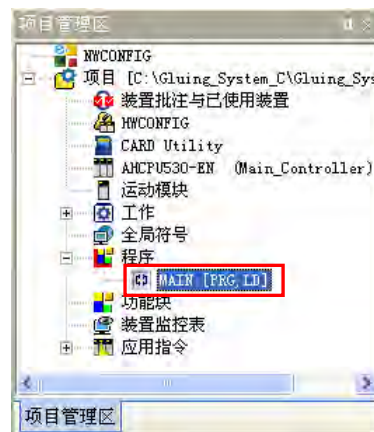
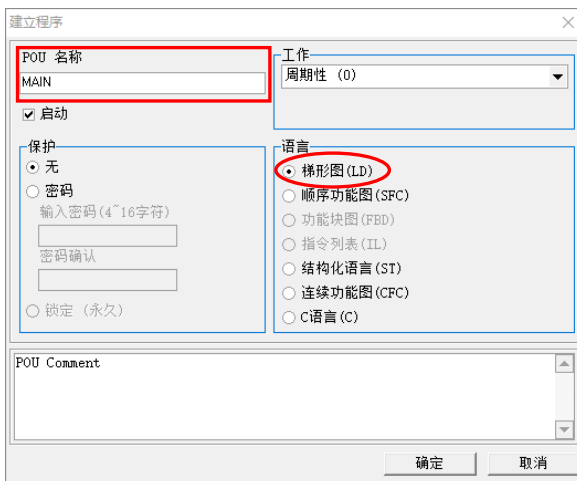
4.5.1 新增梯形图程序

- (1) 于项目管理区的「程序」上单击鼠标右键，接着于选单中依次点选 **新增 (N)**。

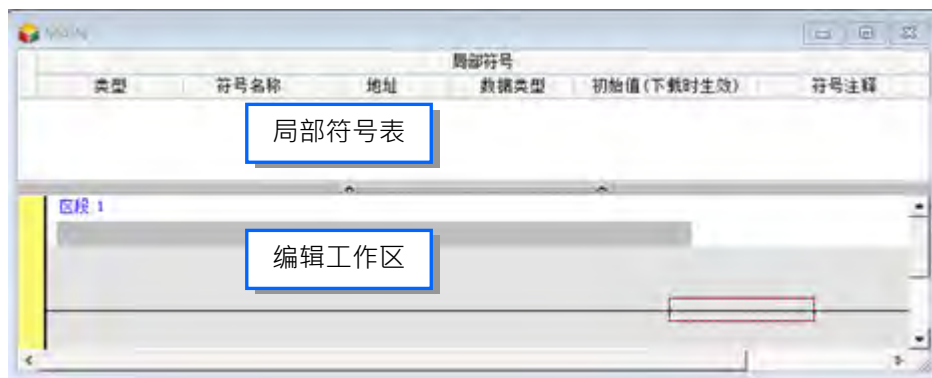
4



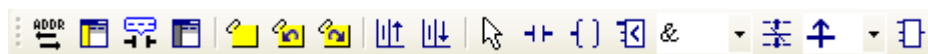
- (2) 于「POU 名称」字段中输入程序的名称，并在「语言」字段中选择**梯形图 (LD)**，而其它的设置则请保持与下图相同的默认值即可；设置完成之后，请按下「确定」键，此时于项目管理区的「程序」项目下便会多出一个对象，而该对象在 ISPSOft 当中便称之为**程序组织单元 - POU**。



(3) 完成 POU 的新增后，在 ISPSOft 的编辑工作区中便会自动打开一个程序编辑窗口。




而进入 LD 的编辑环境后，于 ISPSOft 的窗口中亦会出现对应的编程图标工具栏，其功能简介如下。



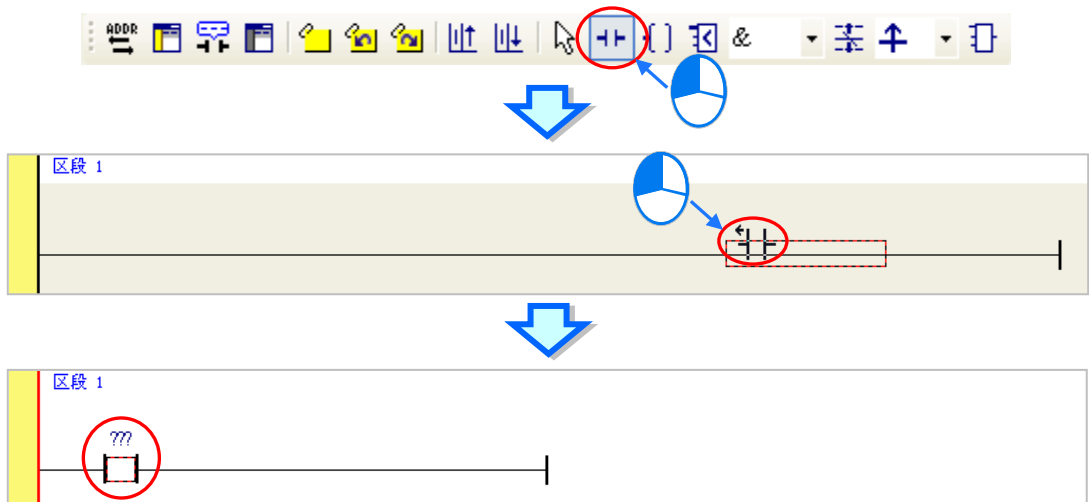
图标	键盘 (快捷键)	功能说明
	无	切换为地址模式
	Shift + Ctrl + C	显示/隐藏区段批注
	无	显示/隐藏装置批注与提示
	Shift + Ctrl + A	将目前选取的梯形图区段设为使能或失效状态
	Shift + Ctrl + B	在目前选取的梯形图区段上加入或取消书签标记
	Shift + Ctrl + P	移至前一个书签的标记位置
	Shift + Ctrl + N	移至下一个书签的标记位置
	Ctrl + I	于目前选取的梯形图区段之前插入一个新区段
	Shift + Ctrl + I	于目前选取的梯形图区段之后新增一个新区段
	ESC	切换为选取工具
	指令编辑模式	插入接点装置
	指令编辑模式	插入线圈装置
	指令编辑模式	插入比较接点
	指令编辑模式	选择比较接点的类型
	指令编辑模式	插入区块逻辑指令 (NP/PN/INV/FB_NP/FB_PN)
	指令编辑模式	选择区块逻辑指令的类型 (NP/PN/INV/FB_NP/FB_PN)
	Shift + Ctrl + U	插入指令或功能块


*.关于指令编辑模式请参考第 4.5.3 节的相关说明。

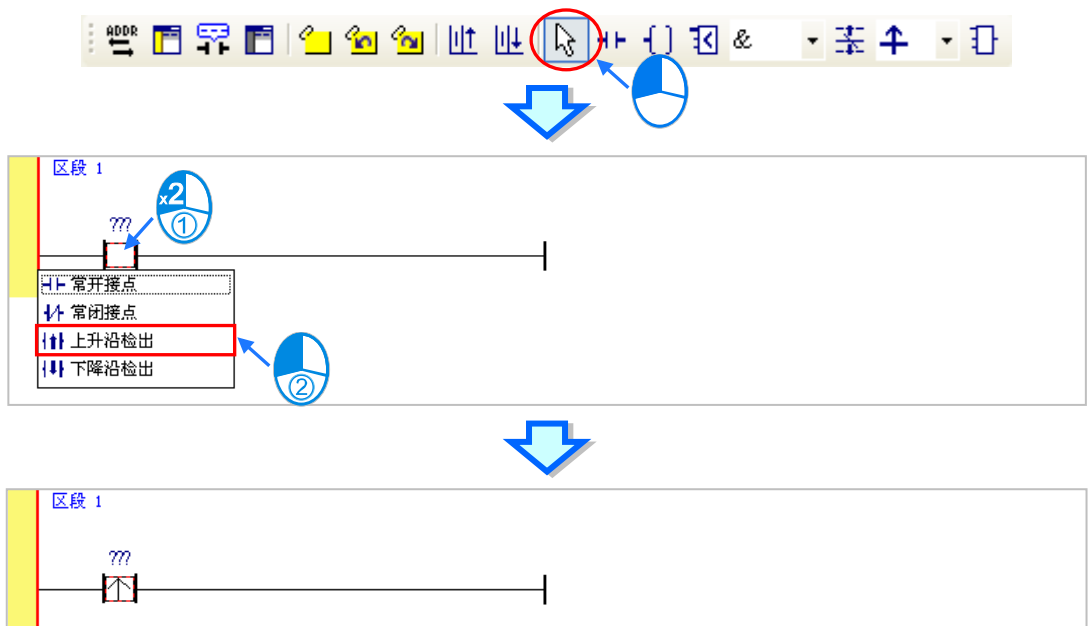
4.5.2 基本编辑 - 建立接点与线圈


- (1) 于图标工具栏中点选  图标以切换为**接点工具**，接着将鼠标移至梯形图区段的红色框线处，而此时鼠标光标便会变为接点的图标；当鼠标往红色框线的左右或下方偏移时，鼠标的图标也会随之变化，依此便可决定**接点**的插入方式；而编辑梯形图程序时，必须将鼠标靠近编辑位置才可进行操作，且插入的组件亦会由系统自动排列，用户将无法任意搬移组件的位置。
- 此处尚无须考虑接点的插入方式，因此请直接将鼠标靠近红色框线处按下鼠标左键即可。

4

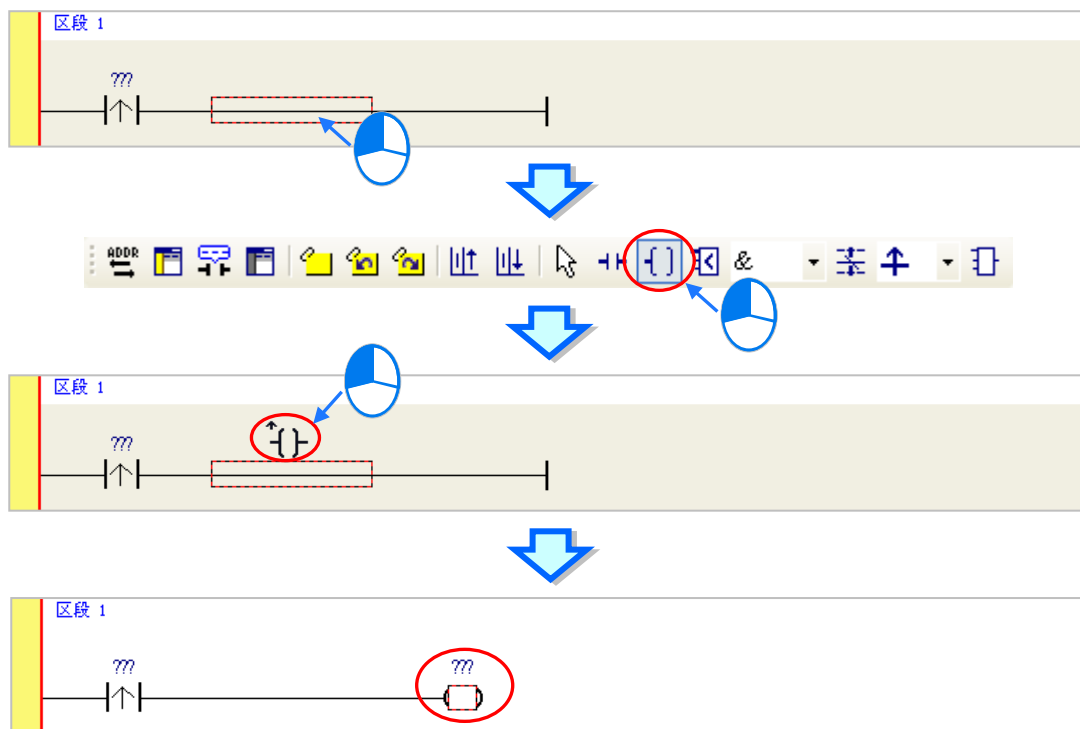



- (2) 接着请点选  图标或按下键盘的【ESC】键将鼠标光标切回**选取**状态，并将鼠标移至接点的图标上双击鼠标左键，此时便会出现接点型式的下拉选单，其中包括「常开接点」、「常闭接点」、「上升沿触发」及「下降沿触发」。在本范例中，此接点请选择「上升沿触发」的型式。

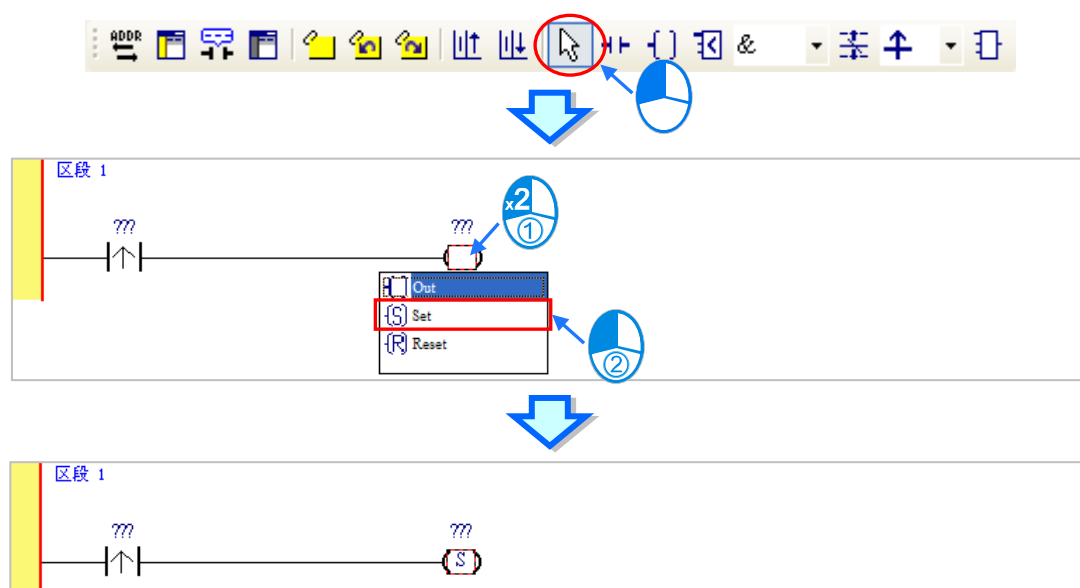


- (3) 接下来请将鼠标移至右方的线路上单击一下左键以改变编辑位置，之后再于图标工具栏中点选  图标以切换为线圈工具，接着再将鼠标移至红色框线处；同样的，当鼠标往红色框线的上方或下方偏移时，鼠标的图标也会跟着变动，依此便可决定线圈的插入方式。

此处同样无须考虑线圈的插入方式，因此请将鼠标靠近红色框线处并按下鼠标左键即可。

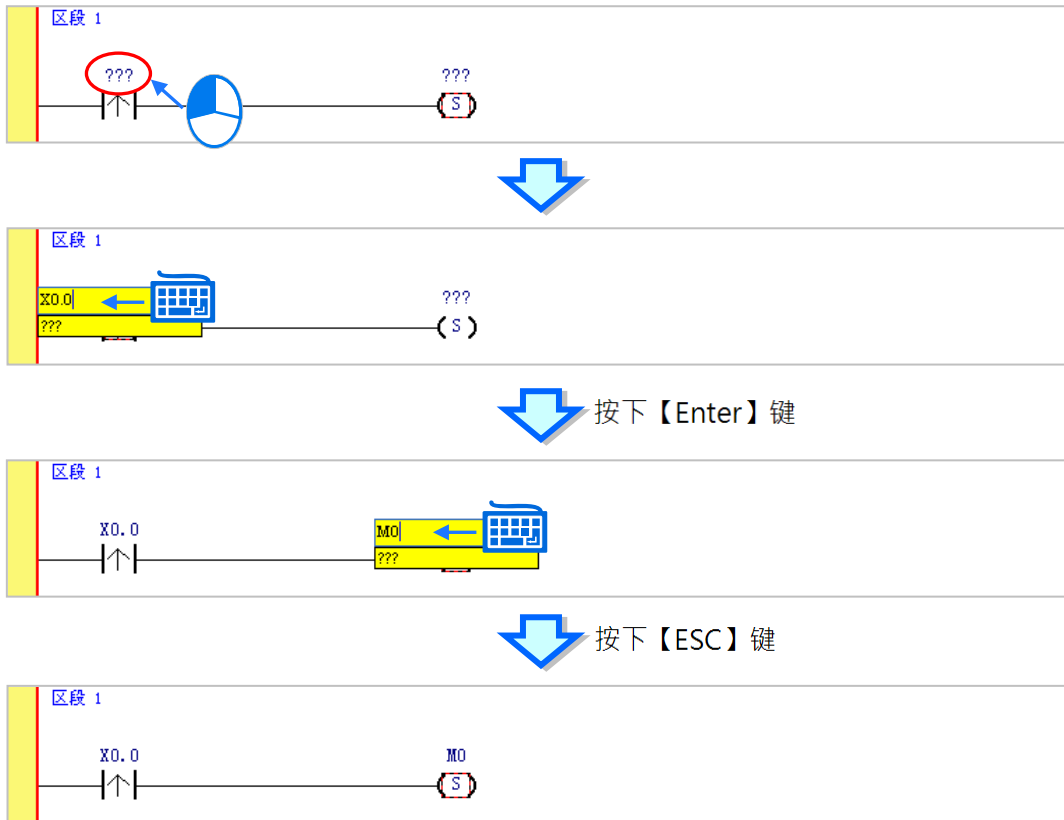


- (4) 接着请点选  图标或按下键盘的【ESC】键将鼠标切回选取状态，并将鼠标移至线圈的图标上双击鼠标左键，此时便会出现线圈型式下拉选单，其中包括「Out」、「Set」及「Reset」。在本范例中，此线圈请选择「Set」的型式。



- (5) 最后请于接点或线圈的图标上方单击 ??? 字段，此时该字段便会进入编辑状态，而之后便可在该字段中输入欲配置给此组件的装置地址；输入完成后，按下键盘的【Enter】键即可自动跳至该区段中的下一个编辑字段，而完成所有字段的编辑后按下【Esc】键即可结束编辑。

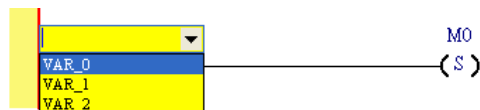
在本范例中，请于接点处输入 X0.0，而线圈的部份请输入 M0。





补充说明

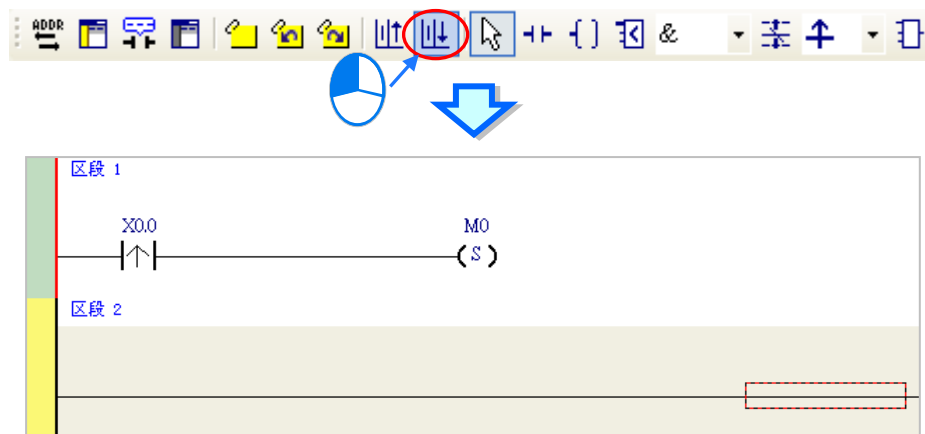
当用鼠标单击某个区段后按下【Enter】键，接着系统便会自动打开编辑字段，利用键盘的【Enter】键便可在同一个区段中反复移动编辑字段，而利用【Tab】键则可跨区段移动，但到了最后一个编辑字段之后便会停止；编辑时，用户可直接于编辑字段中使用键盘输入，而完成编辑后，按下【Enter】键便会自动再切换至该区段中的下一个编辑位置；当要结束编辑时，只要按下【Esc】键即可。

若用户有宣告变量符号，则在装置地址字段呈现编辑状态时，先清除装置输入字段后便会自动出现符合该字段型态变量符号的下拉选单供使用者挑选，此时直接使用鼠标点选或利用键盘的上下键便可进行选择。关于变量符号的相关说明将于第 6 章的内容当中有详尽的介绍，此处将不再多加叙述。



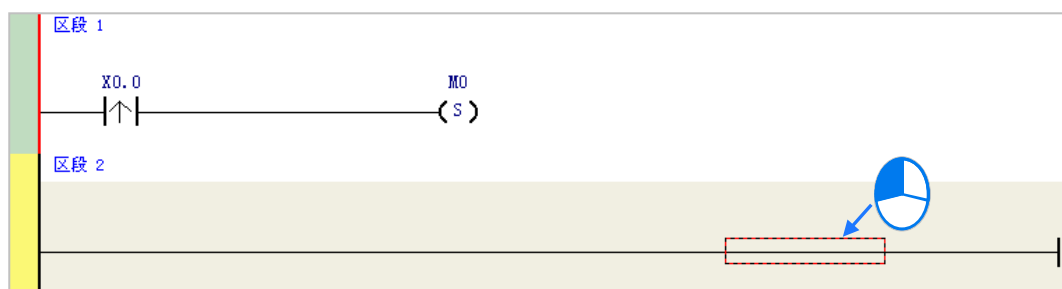
4.5.3 基本编辑 - 梯形图区段的新增/插入与指令编辑模式

当于图标工具栏中点选  图标便可于目前选取的区段下方新增一个空白区段；而点选  图标则会于目前选取的区段上方插入一个空白区段。此处我们先在区段 1 的下方新增一个空白区段。

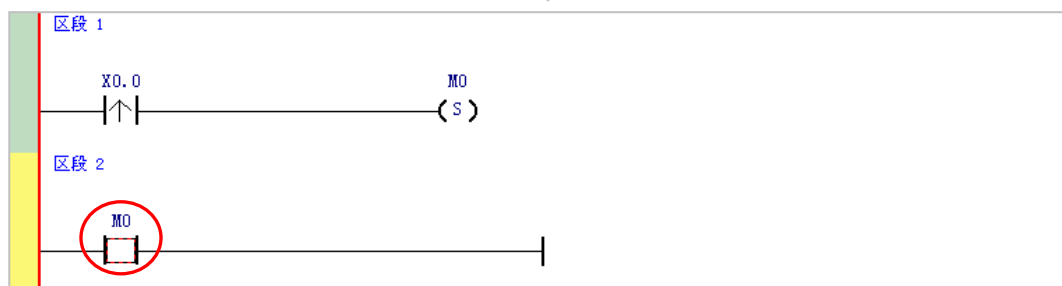
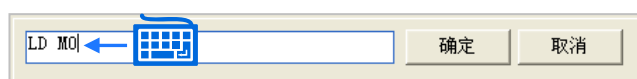


建立接点与线圈的方式除了利用前一节所介绍的图标工具之外，尚可利用指令编辑模式来达成。

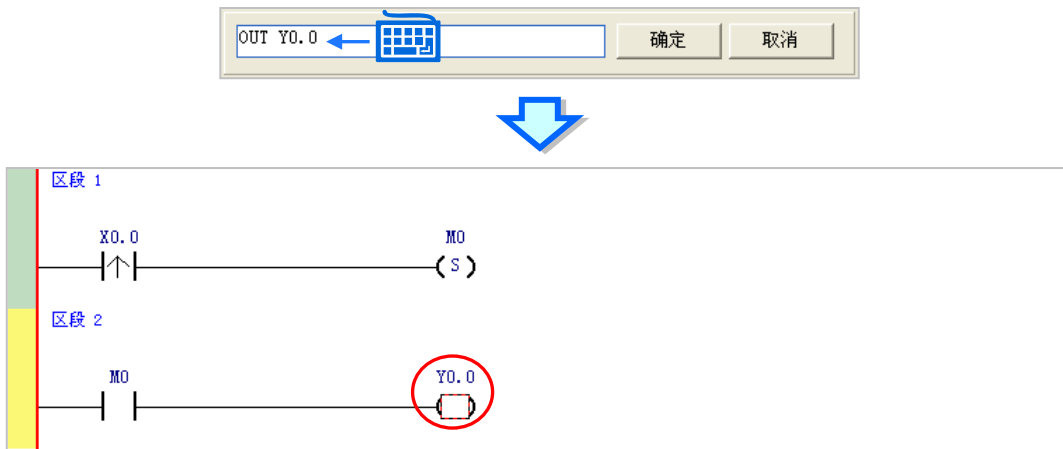
(1) 首先请用鼠标单击区段 2 的线路以决定编辑位置。



(2) 接着请直接用键盘输入 IL 指令（无须分大小写） - "LD M0"，且一按下键盘之后，画面便会自动出现编辑窗口，待输入完毕后按下键盘的【Enter】键或以鼠标单击窗口的「确定」钮即可。



(3) 接着请再输入 IL 指令 - " **OUT Y0.0** "，并完成如下的程序。

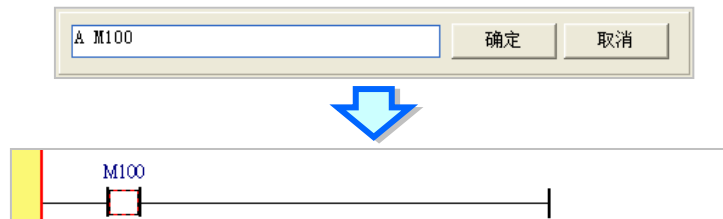


4

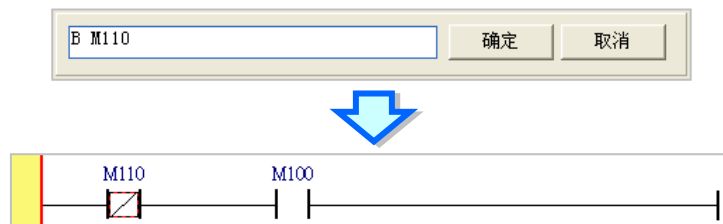
补充说明

针对接点与线圈，系统亦提供简易输入的功能，请参考下述。（输入的指令无须分大小写）

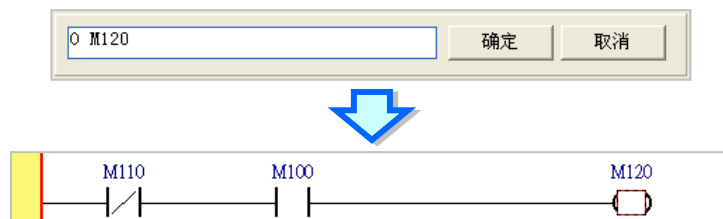
- 插入常开接点（A 接点）指令：“A 装置地址”



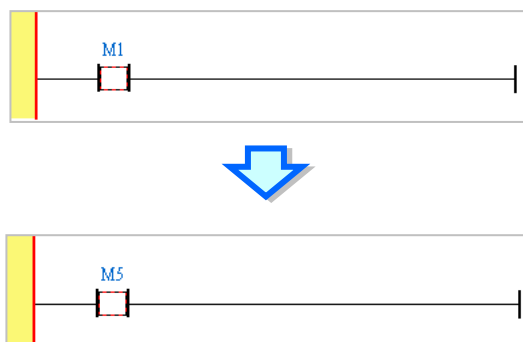
- 插入常闭接点（B 接点）指令：“B 装置地址”




- 插入输出线圈（OUT）指令：“O 装置地址”

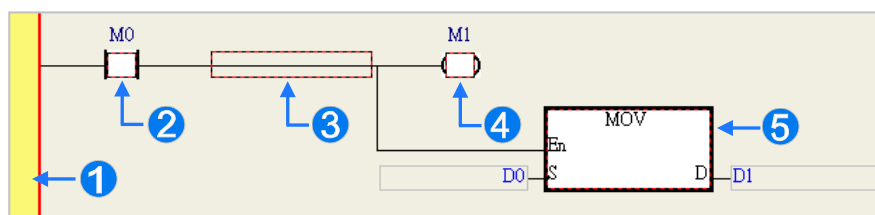


当加入装置项目时，切换键盘的【Insert】键便可选择使用插入或取代模式，切换键盘的【Insert】键并在 M1 位置输入 LD M5，原本的 M1 就会被 M5 取代，如下图。



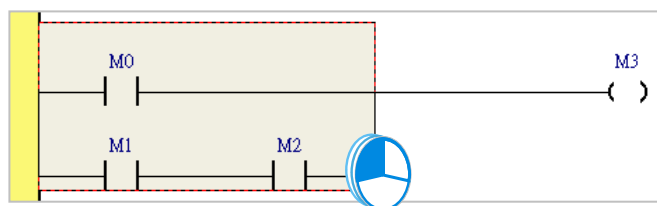
4.5.4 基本编辑 - 梯形图区段的选取与操作

要在梯形图区段中进行选取时，请先按下键盘的【ESC】键，或于图标工具栏中单击  图标，待鼠标切换为选取状态后，直接于欲选取的对象上单击鼠标左键即可。下列即为基本的选取操作。

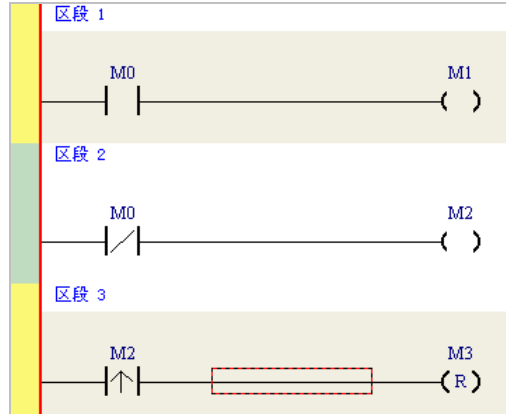


- ❶ 选取整个区段。
- ❷ 选取输入接点。
- ❸ 选取整个区段。
- ❹ 选取输出线圈。
- ❺ 选取方块。

要选取区块时，可先点选任一装置后按住鼠标左键，接着拖曳鼠标便可将区块框选起来；或是先点选区块范围的第一个对象后，按下键盘的【Ctrl】+【B】，接着再点选区块范围的最后一个对象，并再次按下【Ctrl】+【B】键便可将其框选；而框选时，该区块必须位于同一个梯形图区段中，且必须是连续位置的对象，而梯形图区段中的输入装置与输出装置则无法被框选为同一个区块。



当要同时选取多个梯形图区段时，请先按住键盘的【Ctrl】键，接着便可用鼠标分别点选多个不连续位置的区段；另外也可先按住【Shift】键后，再分别用鼠标点选欲选取范围的第一个与最后一个梯形图区段，之后便可将整个范围内的区段全部选取起来。



当选取某个对象后按下鼠标右键，之后便可利用快捷选单执行一些基本的操作。

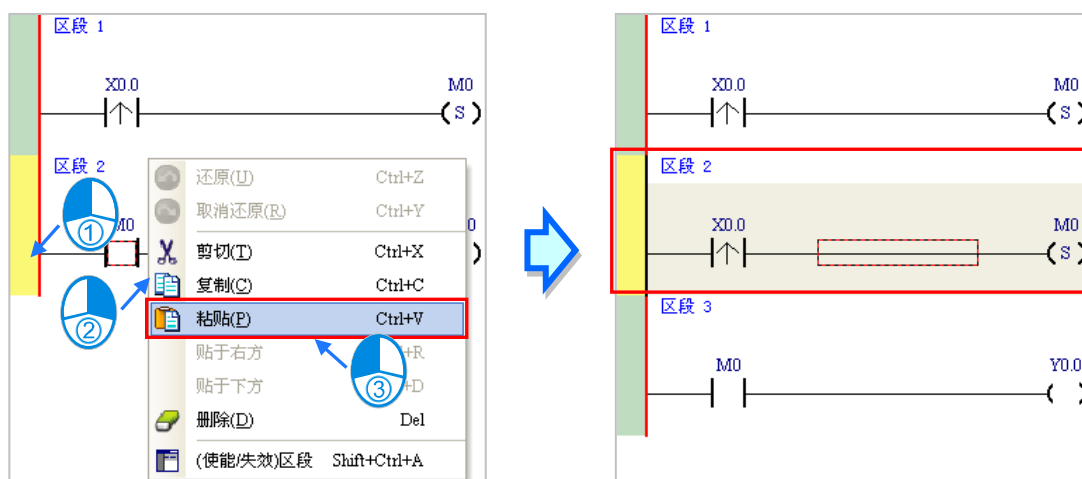
操作项目	功能说明
复原	复原方才的编辑动作（最多可复原至 20 个步骤）
取消复原	取消复原而重复方才的动作
剪切	剪切选取的装置、区块或梯形图区段
复制	复制选取的装置、区块或梯形图区段
粘贴	将方才复制或剪切的对象贴在目前的编辑位置
贴于右方	将对象贴在目前选取的编辑区右方（代表与选取区块串联）
贴于下方	将对象贴在目前选取的编辑区下方（代表与选取区块并联）
删除	删除选取的装置、区块或梯形图区段
区段使能	将选取的区段状态设为使能
区段失能	将选取的区段状态设为失效（失效区段则会于编译时被略过）
自动生成变量	于功能块的接脚上使用，可自动生成变量
新增至装置监控表	于所选取的接脚上使用，将使装置快速新增至监控表

接下来则继续本章范例的操作。


- (1) 首先选取**区段 1** 的整个梯形图区段后按下鼠标右键，之后于快捷选单中点选「复制」。



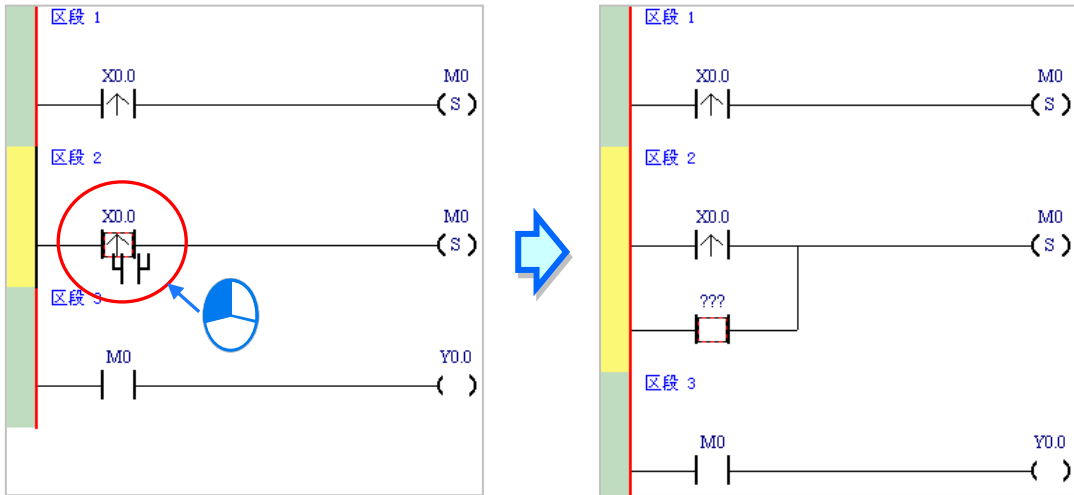
- (2) 接着选取**区段 2**，并在按下右键后于快捷选单中点选「粘贴」，此时方才所复制的梯形图区段便会被插在目前选取的**区段 2** 上方，而原本的**区段 2** 则将自动递增编号为**区段 3**。



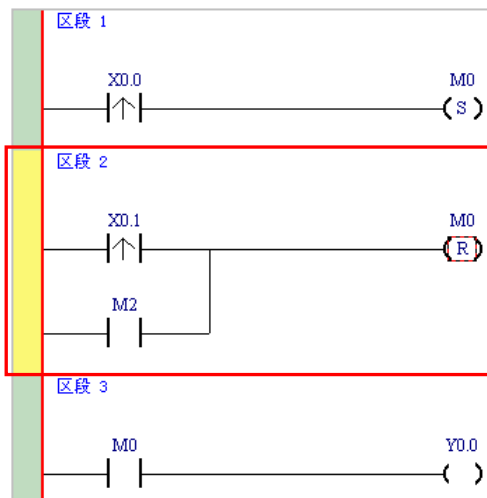
4.5.5 基本编辑 - 并联接点装置

- (1) 首先请于图标工具栏中点选  图标以将鼠标切换为**接点工具**，接着再将鼠标移至目前**区段 2** 当中的输入接点，而此时鼠标亦会变为接点的形状；刻意将鼠标稍微往下方偏移，待鼠标符号变为并联的形状后按下鼠标左键，如此便可完成一个并联类型的区块。

4

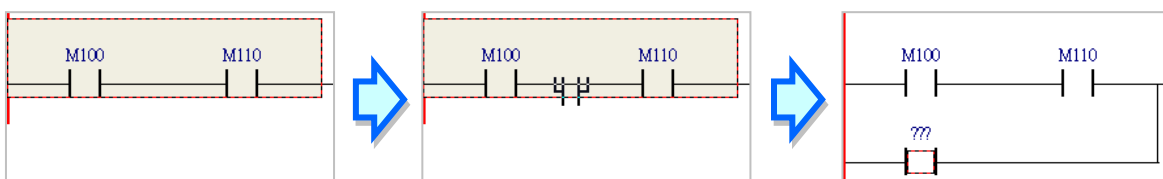


- (2) 接着请参考前述的方法将**区段 2** 修改成如下图的程序。



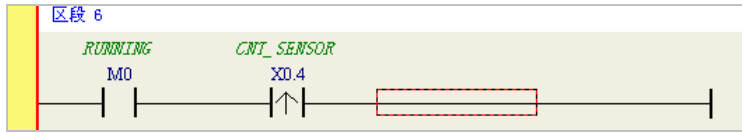
补充说明

当要与某个区块进行并联时，只要先做区块选取之后，再比照上述的方式操作即可。



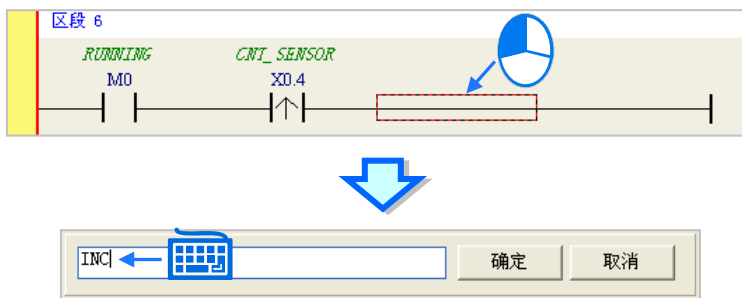
4.5.7 基本编辑 - 插入 API 应用指令

首先请于上一节完成的五个区段下方新增区段 6，并编写如下的程序。之后由下列三种方式选择其中一种来插入 API 应用指令。



● 方法一

先点选欲插入指令的位置，接着直接以键盘输入指令名称（此范例为 INC）后按下【Enter】键即可。

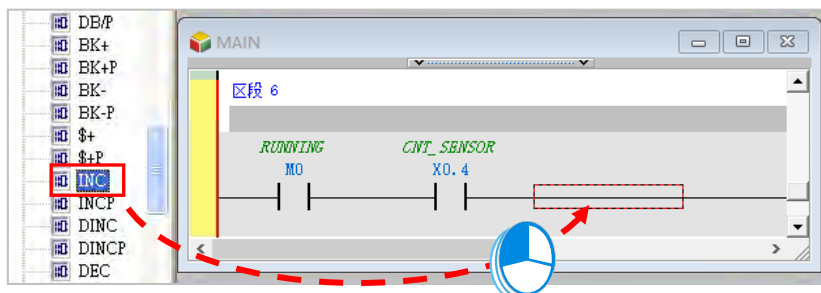


● 方法二


展开项目管理区的「应用指令」，并在找到欲插入指令的类别后将其展开。

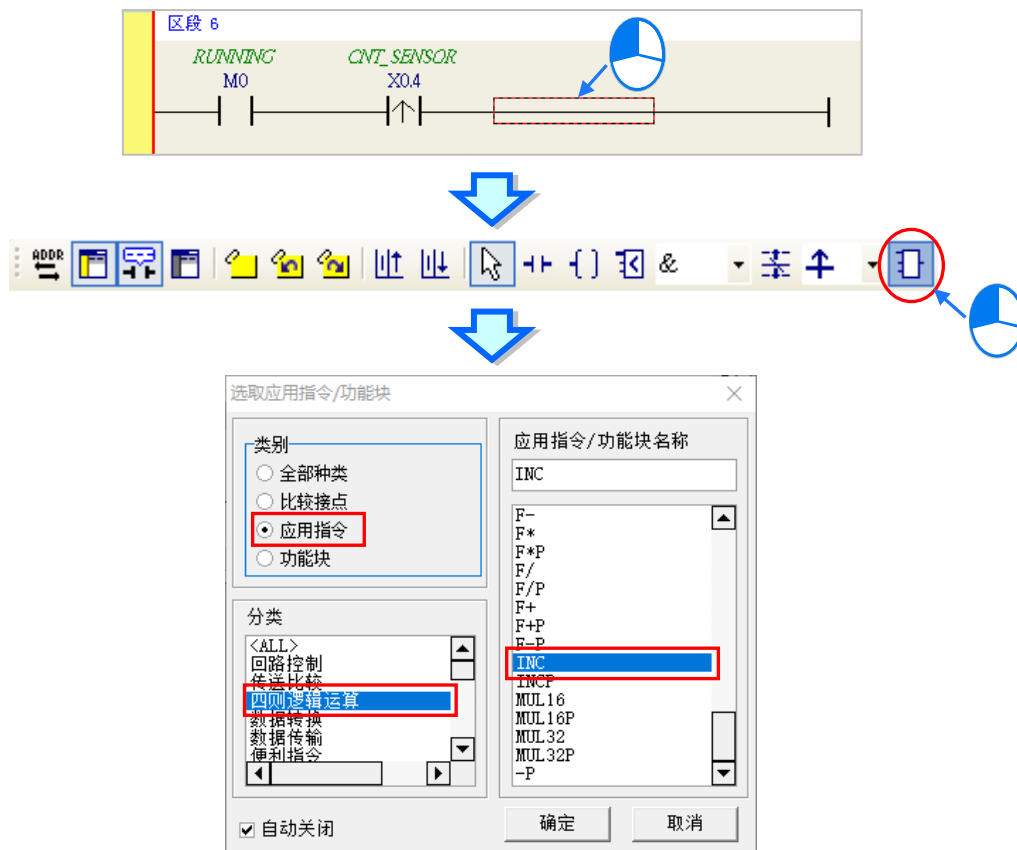


点选欲插入的指令（此范例为 INC）后按住鼠标左键，接着将其拖曳至插入位置后放开即可。

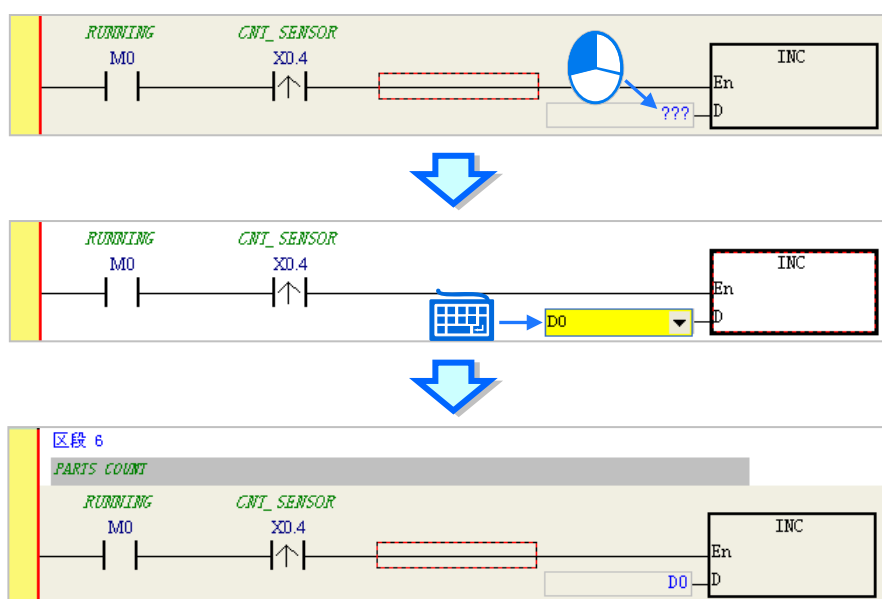


- 方法三

先于欲插入指令的位置上单击鼠标左键，之后点选  图标，接着再于出现的选取窗口中点选要加入的 API 应用指令（此范例为 INC），完成后按下「确定」键即可。

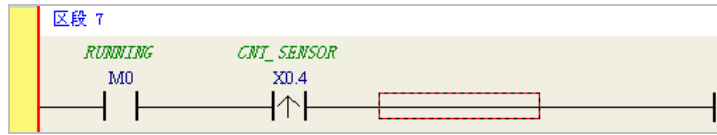


成功插入 API 应用指令后，请参考前述的方式修改指令的操作数，并完成如下图的梯形图程序。

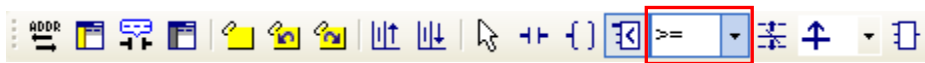


4.5.8 基本编辑 - 比较接点的建立与常数的输入格式

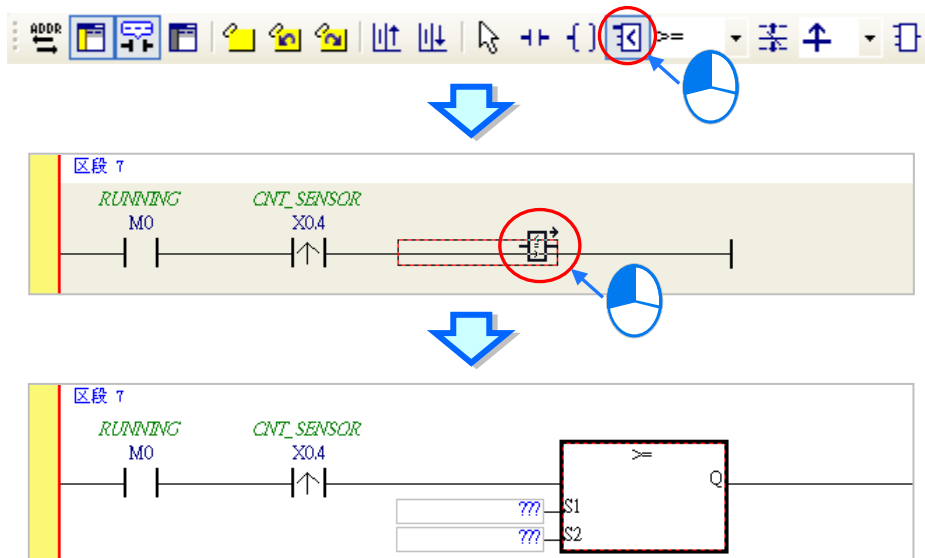
插入比较接点的方式，除了第 4.5.7 节所介绍的三种方法之外，还可以依据下列的步骤。在此之前，我们先在所有区段的下方新增一个区段 7，并编写如下的程序。



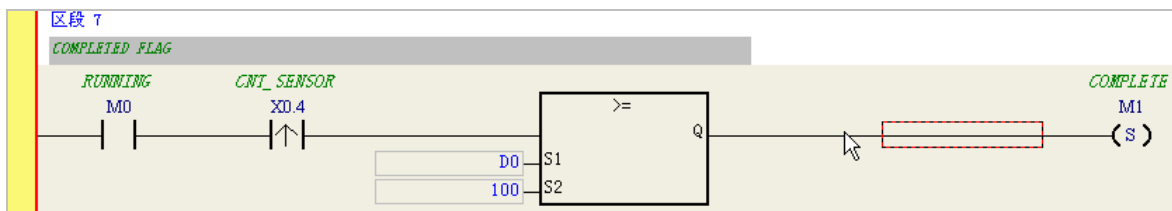
- (1) 先点选工具栏图标 并下拉选择比较接点的类别。(此处请选择 >=)



- (2) 点选工具栏图标 ，并将鼠标移至欲插入比较接点的位置，此时鼠标的偏移同样会改变鼠标的图标而决定插入的位置。在确定位置之后按下鼠标左键即可插入比较接点。

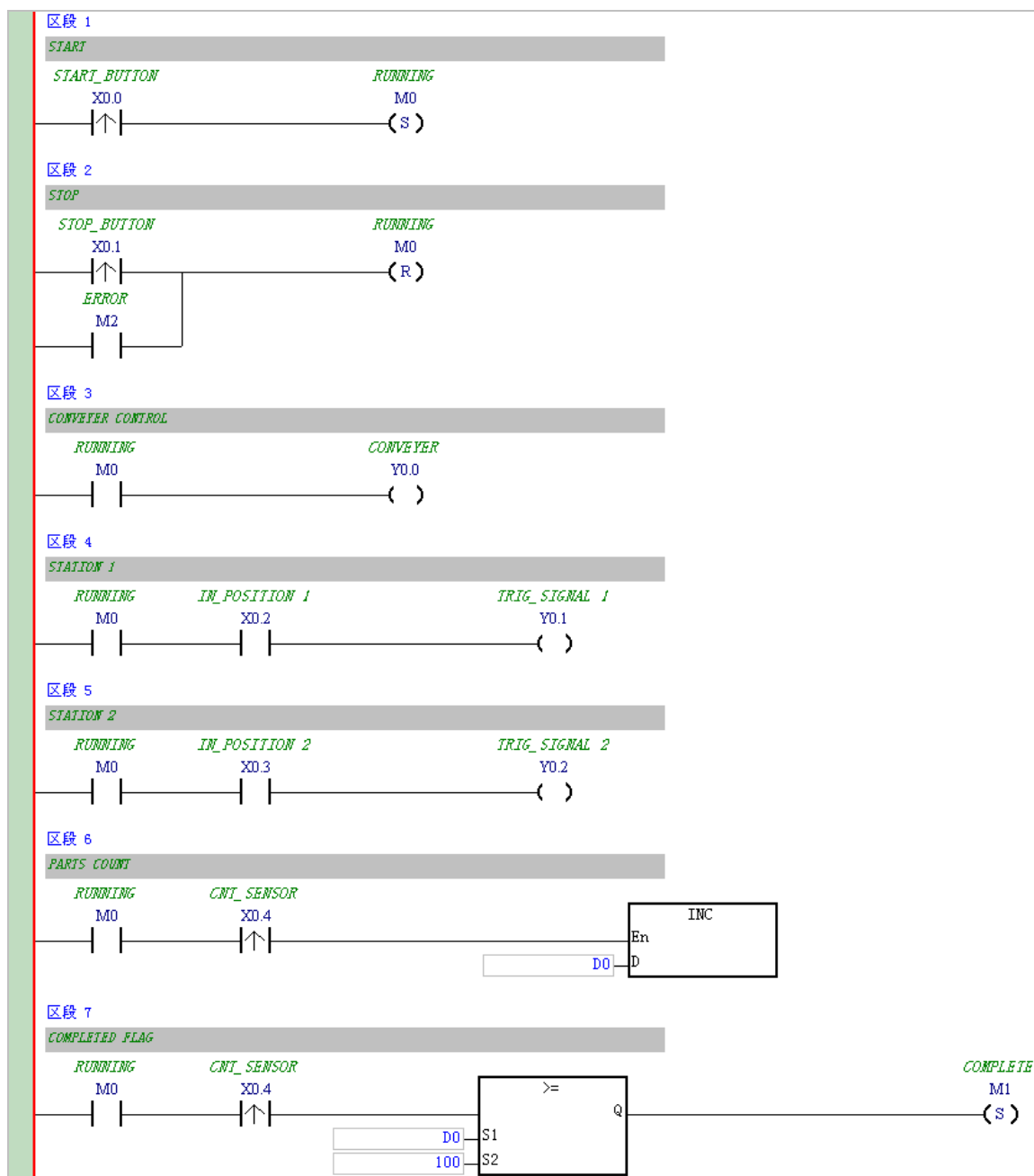


接着请参考前述的方式完成如下的梯形图程序。而有别于早期在 WPLSoft 中使用「K」与「H」来表达常数的方式，当要在 ISPSOft 中输入 10 进制常数时，仅需直接输入数字即可，而 16 进制常数则必须于前方加上「16#」，如 16#7FFF；亦可使用「8#」与「2#」来表达 8 进制与 2 进制常数。



4.5.9 完成范例程序

梯形图编程的基础操作方式都已大致介绍完毕，而依据前面各节的说明，我们便可逐步完成如下图的范例程序；在编辑的过程中，我们会发现左侧的母线一直是呈现红色的状态，这代表此程序尚未完成编译，而在接下来的章节中，我们将继续介绍如何对完成的范例程序进行编译，并将编译后的程序下载至主机当中进行基本的测试。




- *1. 上图的范例程序亦会放置在 ISPSoft 安装目录下的 `..\Project\Example\Gluing_System_CI` 资料夹中。
- *2. 更详尽的梯形图编程说明，请参考第 10 章的相关介绍。

4.5.10 程序的检查与编译


当完成一个程序的编辑之后，接着便可对其进行语法检查或编译的工作。其中「检查」只会对目前工作窗口的内容进行语法与结构的确认，而「编译」则会针对整个项目进行确认，并在检查无误后自动产生可让 PLC 运行的执行码，且编译成功后，梯形图左侧的母线也会转变为黑色的状态。

● 检查

于菜单工具栏中点选 **编译 (C) > 检查 (C)** 或直接于图标工具栏中单击  图标即可。

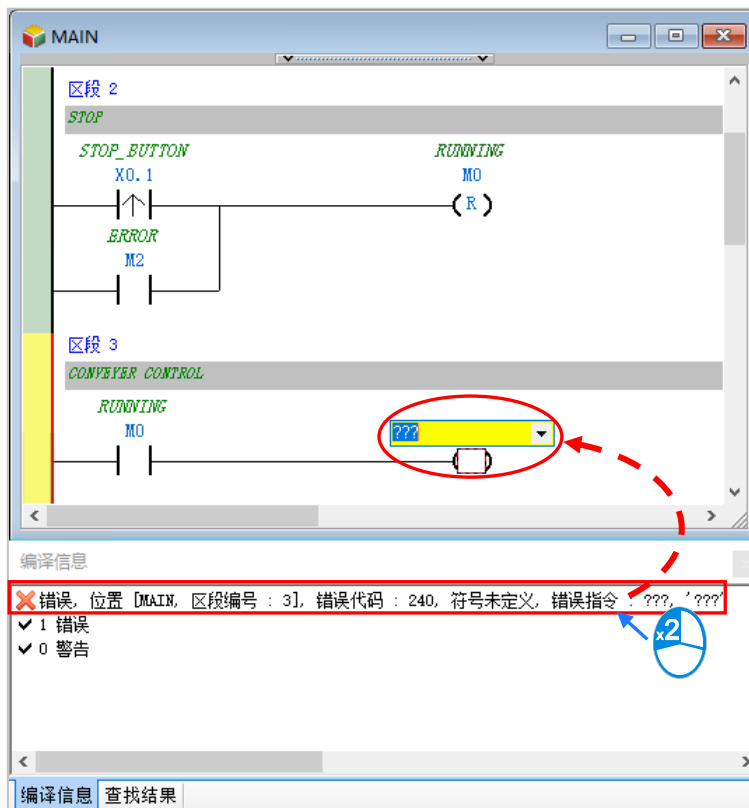


● 编译

于菜单工具栏中点选 **编译 (C) > 编译 (A)** 或直接于图标工具栏中单击  图标即可。



执行完毕后，于 ISPSOft 的「编译讯息」窗口中将会显示确认的结果，而当发现错误时，该窗口便会列出相关的讯息，直接于该讯息上单击两下鼠标左键，系统便会自动引导用户移动至产生该错误的位置，而接着便可进行修正的工作；当修正完毕后，只要再次执行「检查」或「编译」的动作即可。

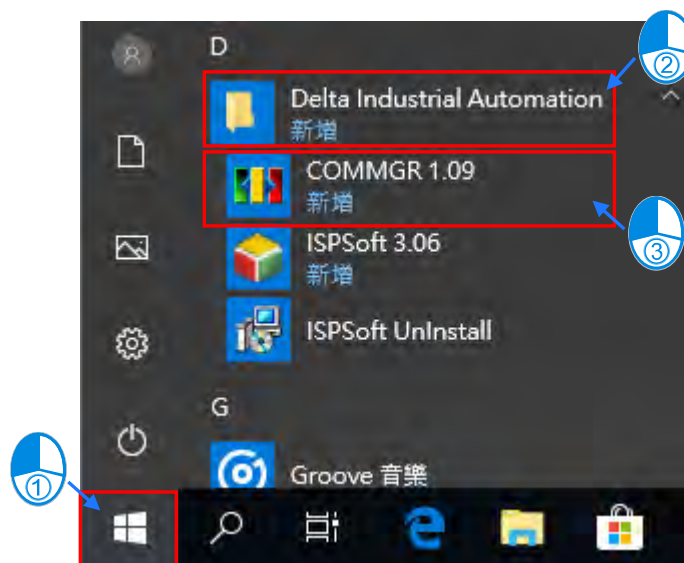


4.6 测试与除错

4.6.1 建立联机

要将程序与参数下载至主机或是进行在线监控之前，必须先在 ISPSOft 与主机之间建立联机，以下针对本章范例所选用的主机 **AHCPU530-EN**，并透过 **USB** 的联机方式来稍做说明，若已经与主机建立联机可直接跳至第 4.6.2 节 下载程序与组态参数。

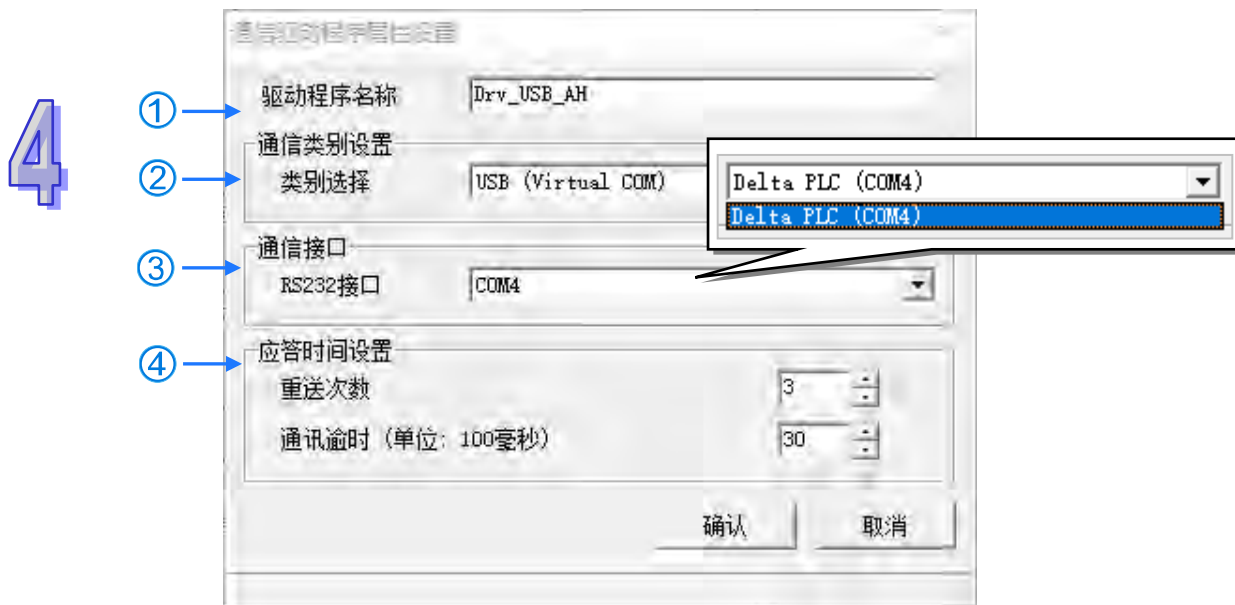
- (1) 将模块正确安装至主背板上，且安装的插槽编号与 HWCONFIG 中的配置必须一致；接着确认接线无误后，将主机上电。
- (2) 将主机与计算机透过 USB 通讯电缆接续在一起，若计算机中已安装 AH500 系列的 USB 驱动程序，此时于 Windows 的设备管理器中便会出现「Delta PLC」的项目，并且会分配到一个 COM Port 的编号。关于安装 USB 驱动程序的相关说明与注意事项请参考 附录 A 当中的内容。
- (3) 确定**通讯管理员 (COMMGR)** 已启动并常驻在 Windows 的系统列中，如果没有启动请至「程序集」中依下列路径至「Delta Industrial Automation」→「COMMGR」目录中单击「COMMGR」的程序执行快捷方式。



- (4) 于画面右下角的 COMMGR 图标上双击鼠标左键，之后便会打开 COMMGR 的管理窗口；接着请于 COMMGR 窗口的右侧单击「新增... (A)」键以新增一组 Driver。

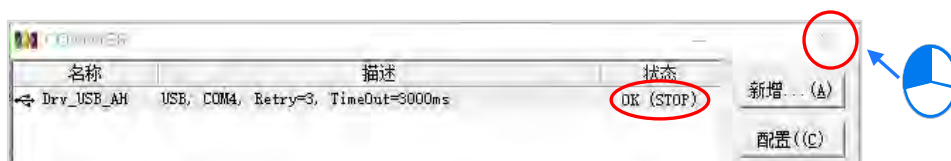


(5) 于 Driver 设置窗口中依序完成各字段的参数设置，完成后按下「确认」按钮即可。



- ① 于「驱动程序名称」字段输入此 **Driver** 的识别名称。
- ② 于「通讯类别设定」的「类别选择」字段下拉选择「**USB (Virtual COM)**」的联机类型。
- ③ 于「通讯端口」字段下拉选择欲建立联机通道的通讯端口，若前述 (1)(2) 两个步骤已正确完成操作，则此时于下拉选单中便会显示已联机的主机装置名称与其 COM Port 编号。
- ④ 设定通讯响应的相关参数。其中「重送次数」参数是设定联机错误时的重试次数，而「通讯逾时」参数则是设定重试的间隔时间。

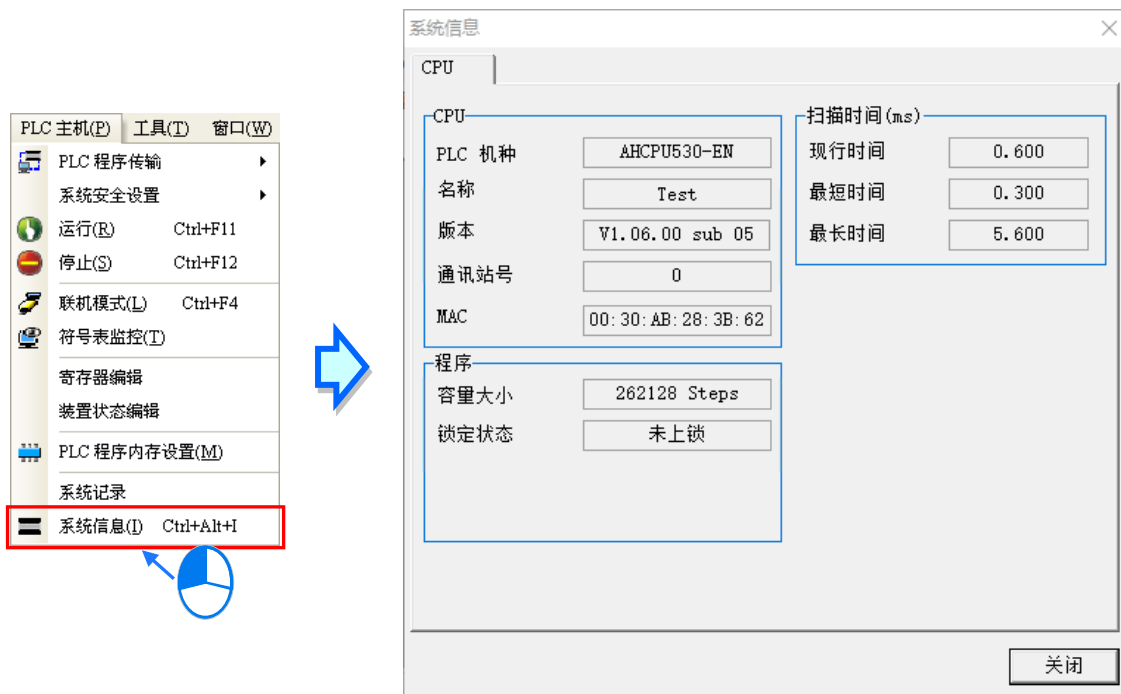
(6) 确认 Driver 的 Status 字段为 **OK** 便可关闭窗口。而此时 COMMGR 仍会继续常驻在 Windows 的系统列中。



启动 ISPSOft 之后，于菜单工具栏中依序单击 **工具 (I)** > **通讯设定 (P)**，接着请于通讯设定窗口的「**通讯信道名称**」字段处下拉选择刚才所建立的 **COMMGR Driver**，而站号字段则保持**站号 0**的设置即可。完成后按下「**确定**」键离开，此时 ISPSOft 的状态栏便会显示目前的设置。



- (7) 于工具栏中点选 **PLC 主机 (P)** > **PLC 状态信息 (I)**，若已可顺利与主机进行通讯时，ISPSOft 便会从主机中获取相关信息，并将其显示于画面中。

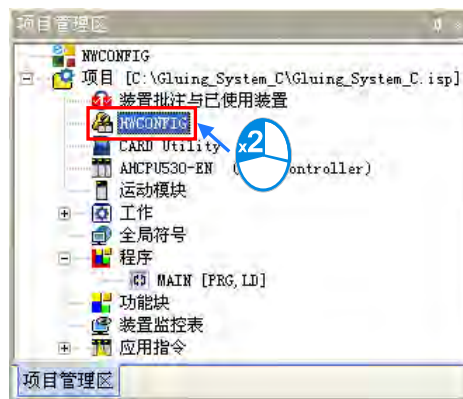


4.6.2 下载程序与组态参数

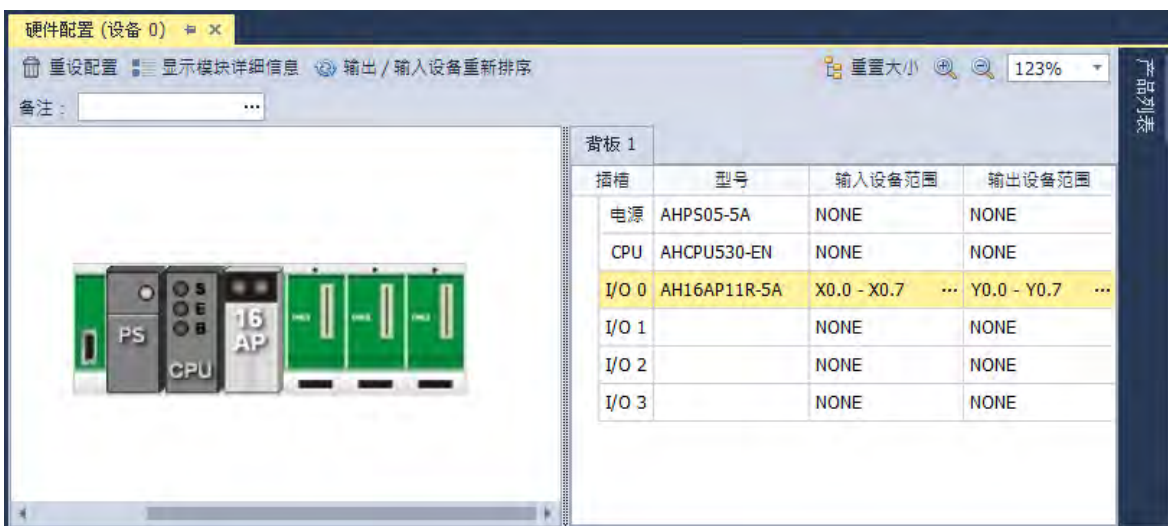
当已确定 ISPSOft 可与主机正常进行联机之后，接着便可将项目中的参数与程序下载至主机当中。首先请启动 ISPSOft 并打开于前几节内容中所建立好的项目，而此范例中需要下载的参数共有 2 项，分别是硬件规划参数与项目程序。

● 下载硬件规划参数

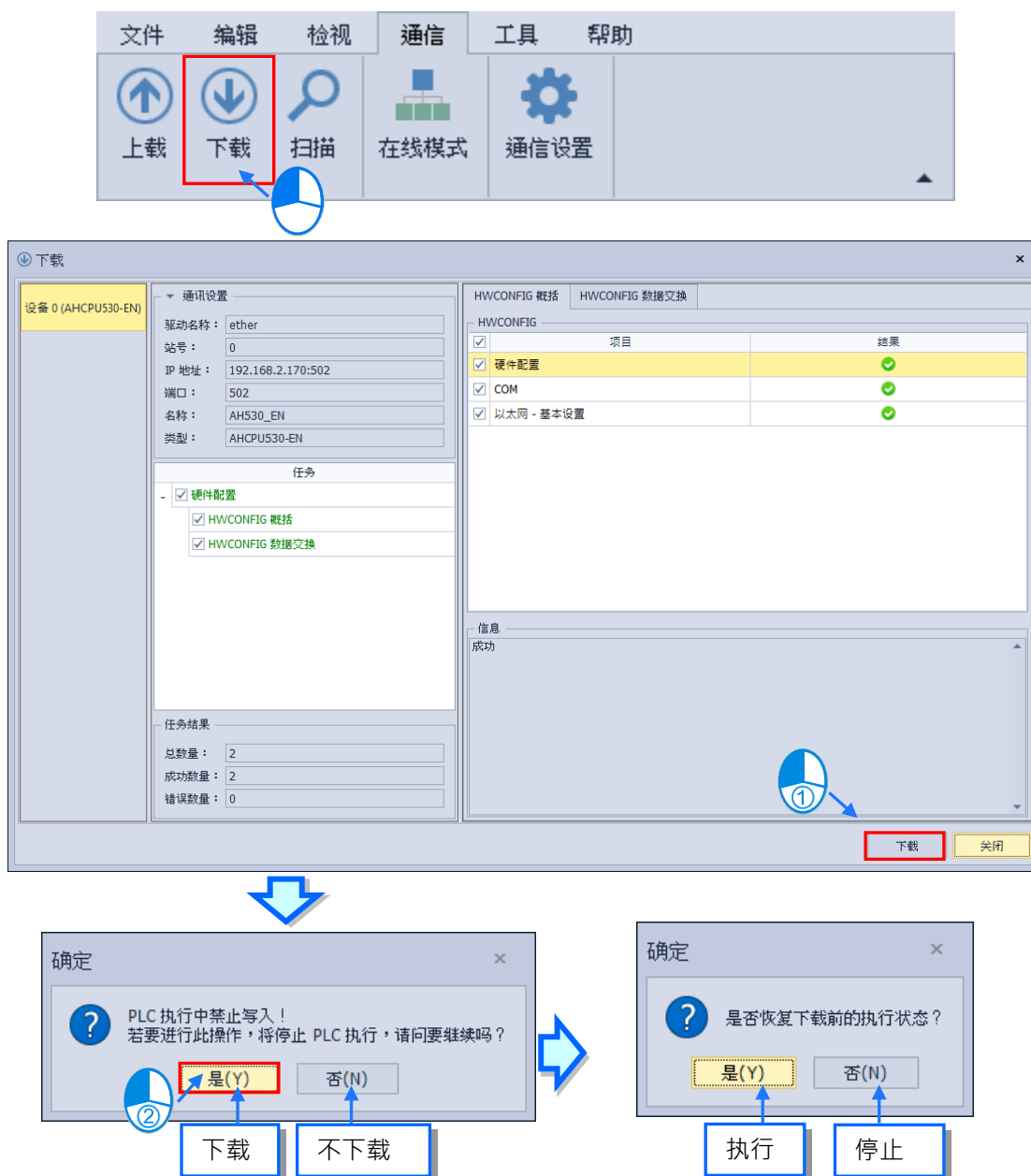
(1) 于项目管理区的「HWCONFIG」上双击鼠标左键，以打开硬件规划工具。



(2) 进入 HWCONFIG 的操作环境后，画面中便会显示先前所规划的硬件配置，而在下载之前请再次确认实际硬件的配置与此处的规划是否相同。




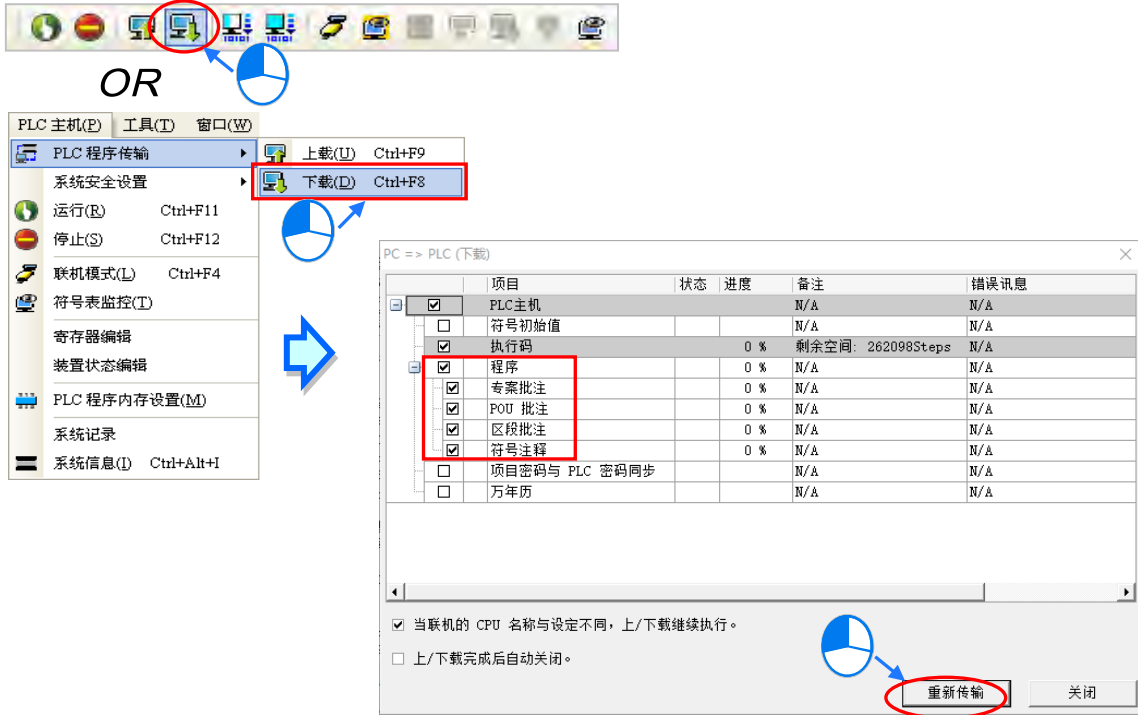
- (3) 于 **HWCONFIG** 的菜单工具栏单击 **通讯> 下载**，接着画面便会出现下载项目的提示窗口，勾选要下载至 PLC 参数后，按下「**下载**」按钮之后即可开始进行下载。



- (4) 成功下载后，主机本体上的 **BUS FAULT** 灯号应要呈现熄灭的状态，此时便可将 **HWCONFIG** 的窗口关闭。但若 **BUS FAULT** 灯号仍呈现恒亮或闪烁的状态时，代表 PLC 本身处于异常状态，此时请再次确认 **HWCONFIG** 中的规划与实际的硬件配置是否一致，或参考该机种的操作手册进行错误排除。更进一步的 **HWCONFIG** 操作说明请参考第 3 章的相关内容。

● 下载项目程序



确认程序编译无误后，于 ISPSOft 的菜单工具栏点选 **PLC 主机 (P) > PLC 程序传输 > 下载 (D)** 或单击图标工具栏的  图标，接着便会出现下载项目的勾选窗口，此时可将项目中的「程序」与相关批注项目勾选起来，以便日后可由 PLC 中将程序重新上传，完成后按下「重新传输」即可。




4

4.6.3 联机测试

当程序下载至 PLC 主机后，我们便可经由 ISPSOft 来对 PLC 的运行状况进行监控。而于 ISPSOft 中，监控功能又可分为「装置监控」与「程序监控」。

监控模式	说明
 装置监控	可透过监控表来实时监控主机目前的装置状态，且因在此模式下，ISPSOft 仅需更新装置状态，因此 ISPSOft 目前打开的程序与主机内部的程序无须一致。
 程序监控	在此模式下，系统会实时将程序的运作状况显示于程序画面中，也因此系统会要求 ISPSOft 目前打开的程序必须与主机内部的程序一致。

* 装置监控模式可单独启动，而程序监控模式则必须伴随装置监控模式一起启动。

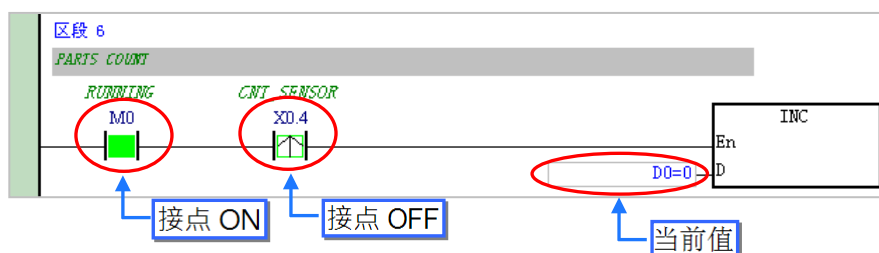
直接于菜单工具栏中点选 **PLC 主机 (P)** > **联机模式 (L)**，或按下图标工具栏的  图标便可进入在线监控，而此时系统将会同时启动「装置监控」与「程序监控」两种模式：





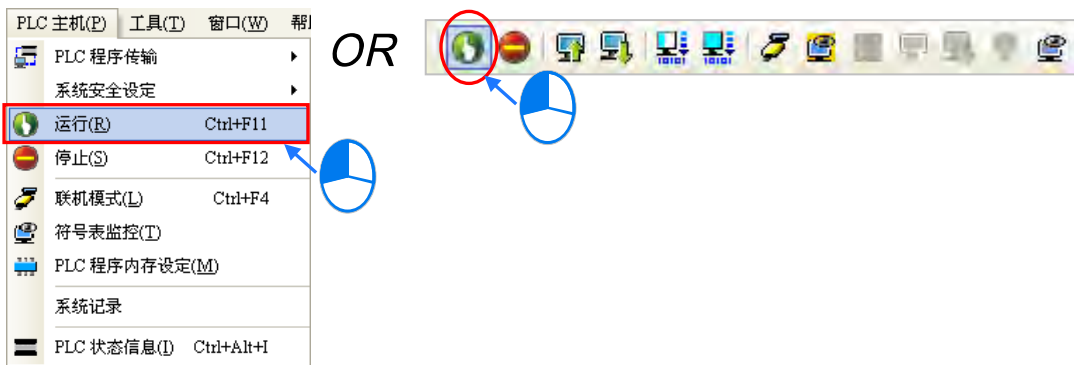
进入在线模式时，由 ISPSOft 的状态栏，便可查看目前的扫描时间、通讯状况与主机状态。



且启动程序监控的模式后，于原本编辑的程序画面中便会实时显示目前的装置状态。



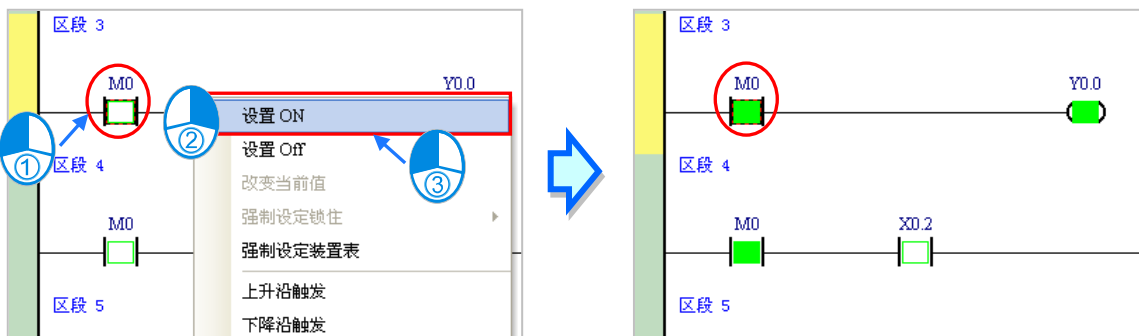
除了可藉由切换主机本体的 RUN/STOP 开关来改变主机的运转状态之外，于 ISPSOft 中亦提供相同的功能供用户进行操作；当于 ISPSOft 的菜单工具栏中点选 **PLC 主机 (P) > 运行 (R)** 或按下图标工具栏的  图标时，PLC 主机便会切换至运行 (RUN) 状态；而当要将 PLC 主机切换回停止 (STOP) 状态时，只要再次于菜单工具栏中点选 **PLC 主机 (P) > 停止 (S)** 或直接按下图标工具栏的  图标即可。



4

在线监控的模式下，先点选任一装置后按下鼠标右键，之后于快捷选单中便会有许多的控制选项可供使用，利用这些功能便可于测试过程中暂时改变某个装置的状态或内容值。

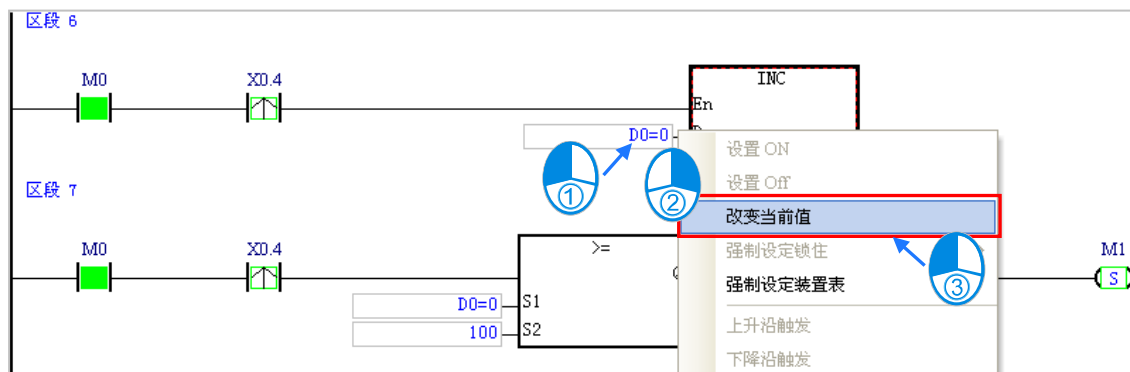
⚠ 变更装置状态前，请务必确认所做的操作不会影响系统运作，或造成系统与人员的伤害。



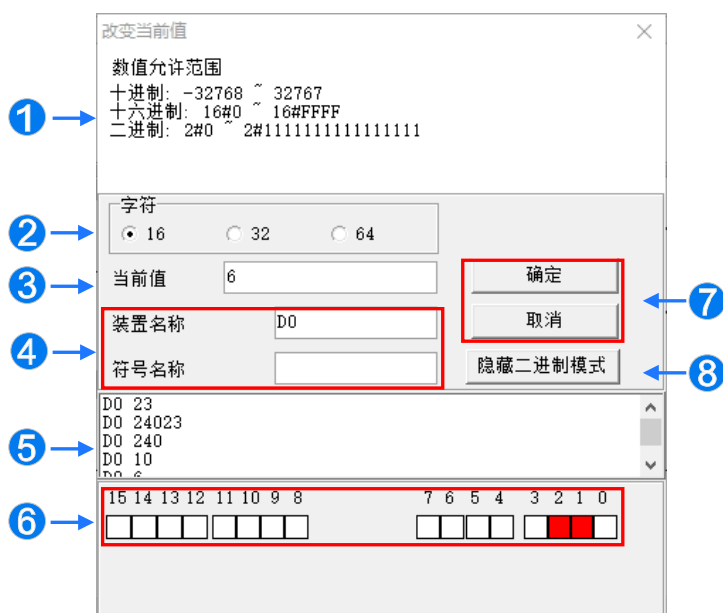
下表则为各操作项的说明简述，而其中**强制设定锁住**的功能只可用于 X 与 Y 接点。

操作项目	说明
设定 ON	将点选的接点状态设置为 ON。
设定 OFF	将点选的接点状态设置为 OFF。
上升沿触发	不管被点选的接点状态为何，系统皆会将该接点先设为 OFF 之后，再设为 ON。
下降沿触发	不管被点选的接点状态为何，系统皆会将该接点先设为 ON 之后，再设为 OFF。
强制设定锁住	将 X 或 Y 接点强制锁定在 ON 或 OFF 的状态。
强制设定装置表	利用表格的方式，批次对 X 或 Y 接点的强制锁定状态进行管理。

当要改变某个装置的内容值时，请先点选该装置后按下鼠标右键，接着于快捷选中点选「改变当前值」，之后便可由数值设置窗口中设置该操作数的内容值。



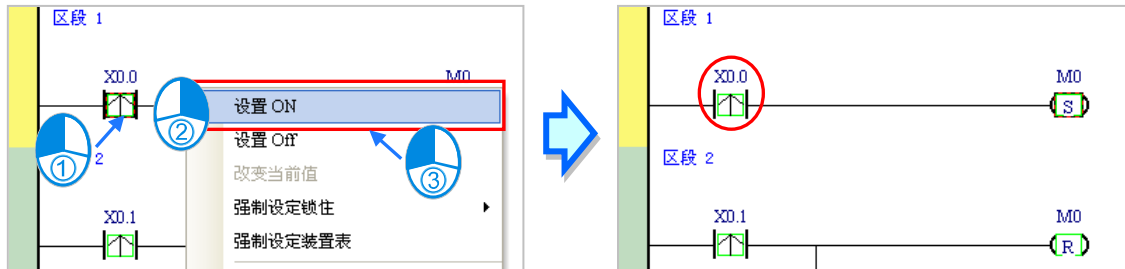
下列即为数值设置窗口的相关说明。



- ① 提示讯息。
- ② 于此选择输入的数值格式为 16 位或 32 位或 64 位。
- ③ 于此输入欲变更的数值。
- ④ 欲变更现在值的装置名称或变量符号名称。
- ⑤ 数值变更记录。(显示格式：装置名称 变更数值)。
- ⑥ 当开启**二进制模式**时，用鼠标便可直接于此处设定各个 Bit 的 ON/OFF 状态。
- ⑦ 按下「**确定**」键可套用设定值，按下「**取消**」键则可关闭此设定窗口。
- ⑧ 单击此键可开启或隐藏二进制模式。

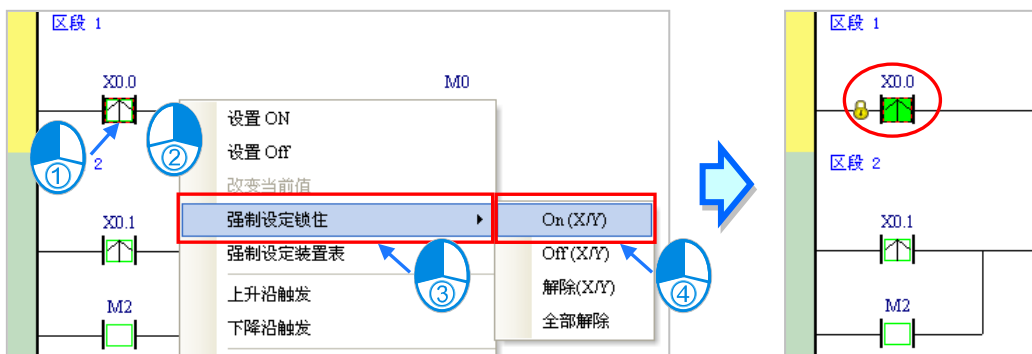
4

在本范例中，X0.0~X0.15 与 Y0.0~Y0.15 为配置给数字 IO 模块 **AH16AP11R-5A** 的输入/输出装置范围，因此当硬件参数下载至 PLC 后，装置 X0.0~X0.15 的状态便会与实际模块的输入状态保持同步。此时若尝试于程序监控画面中，将 X0.0~X0.15 的接点状态直接设为 **ON** 或设为 **OFF**，该接点的状态仍旧会依据外部实际输入的讯号而被重复刷新。



4

不过在进行测试的过程中，还是可藉由**强制设定**的功能来将 X 接点锁定在固定的状态。设置时，请先点选欲设置的 X 或 Y 接点，接着按下鼠标右键并于快捷选单中点选「**强制设定锁住**」，之后便可透过下阶的项目来进行相关的操作。而处于锁定状态的 X 或 Y 接点，其图标旁便会会出现一个大锁的符号。



强制设定锁住	说明
On (X/Y)	将点选的 X 或 Y 接点锁定在 ON 的状态。
Off (X/Y)	将点选的 X 或 Y 接点锁定在 OFF 的状态。
解除 (X/Y)	解除点选接点的锁定状态。
全部解除	解除所有接点的锁定状态。


当程序中的 Y 接点处于**强制设定**状态时，代表该接点的输出状态将不会受到程序执行结果的影响。

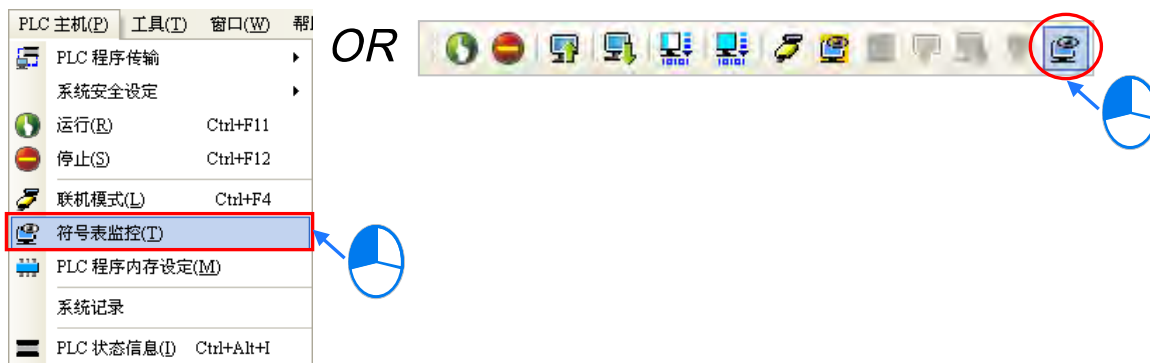


*.离开在线监控时并不会自动解除主机内部的强制设定状态，因此结束测试后，请确认是否须将强制设定的状态全部解除。

下述为建立监控表的方式，且不论是离线或联机的状态下皆可进行监控表的建立。

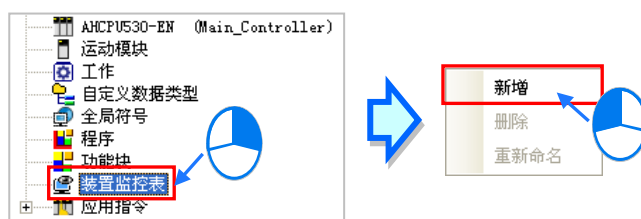
● 方法一

于菜单工具栏中点选 **PLC 主机 (P)** > **符号表监控 (I)**，或于图标工具栏中点选  图标。

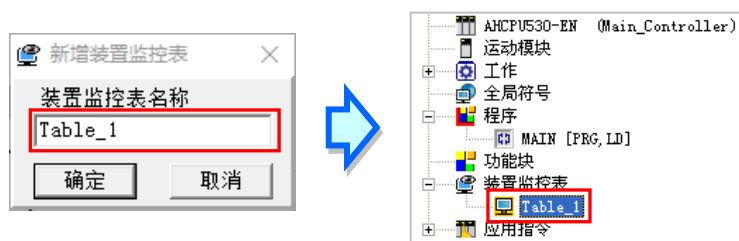


● 方法二

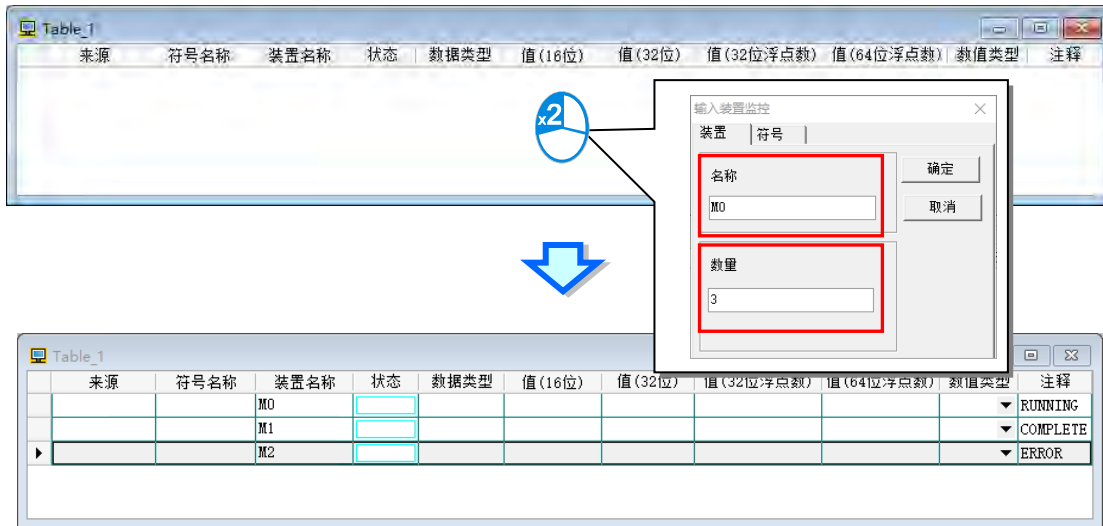
在项目管理区点选「装置监控表」后按下鼠标右键，快捷选单中点选 **新增**。



依据上述方式操作之后，接着在弹出的窗口中输入监控表的名称，完成后按下「确定」。项目管理区的「装置监控表」下便会产生一个对象，当后续要打开监控表时，只要于该对象上双击鼠标左键即可；一个项目中可建立多个监控表，而所建立的监控表亦将随着项目一起被保存。



当新增一个监控表时，画面便会自动打开该监控表的管理窗口，此时便可直接于该窗口中建立监控对象。欲建立对象时，请于**监控表**的空白处双击鼠标左键，或直接输入装置名称，接着便可于弹出的窗口中输入启始的装置地址与欲加入的数量，不过须注意每个监控表最多只可建立 100 个监控对象。



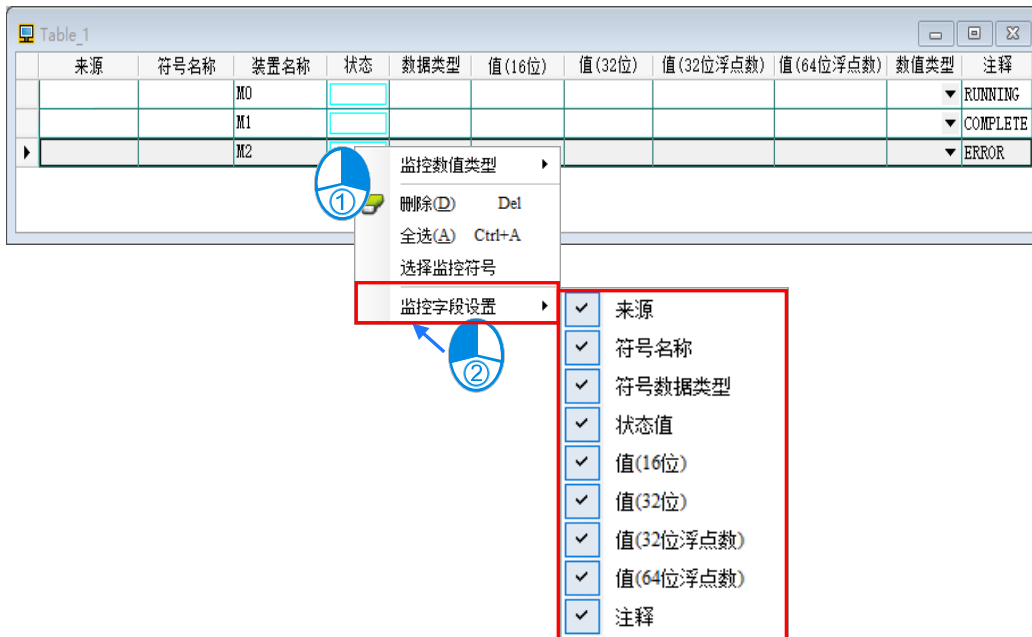
4

当加入装置项目时，切换键盘的【Insert】键便可选择使用插入或取代模式；ISPSOft 窗口的状态栏中便得知目前所选择的状态。

当选择插入模式时，新增的监控对象将会插于目前监控表中被选取的对象上方；而选择取代模式时，新增的监控对象便会直接将目前所选取的监控对象取代掉。



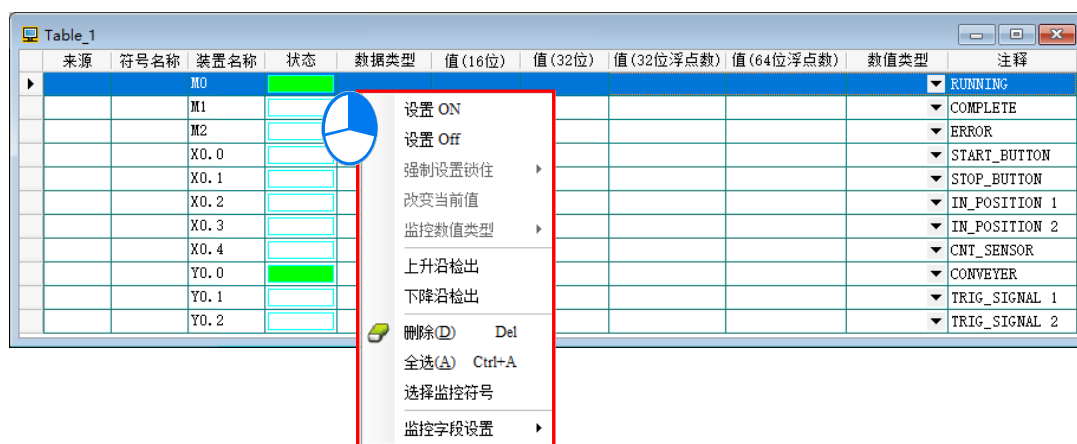
当要隐藏某些监控表的字段时，请于监控表中按下鼠标右键，接着于选单中点选「监控字段设定」，之后便可透过其下阶的项目来进行设置，而当取消勾选时，该字段便会被隐藏。



下表则为监控表中各字段的相关说明。

监控表字段	说明
来源	显示变量符号的来源。
符号名称	显示变量符号的名称。
装置名称	显示监控的装置名称。
状态	当监控的对象为位或接点时，此字段会显示其 ON/OFF 状态。
数据类型	若监控的对象为变量符号，则此字段会显示该变量符号的数据类型。
值 (16 位)	实际联机监控时，此字段会以 16 位的类型来显示监控值。
值 (32 位)	实际联机监控时，此字段会以 32 位的类型来显示监控值。
值 (32 位浮点数)	实际联机监控时，此字段会以 32 位的浮点数类型来显示监控值。
值 (64 位浮点数)	实际联机监控时，此字段会以 64 位的浮点数类型来显示监控值。
数值类型	于此下拉可选择联机监控时所显示的数值格式。
批注	显示监控装置的 装置批注 或监控符号的 符号批注 。

当建立好一个监控表后，可于在线模式下透过监控表来监控表格中的各个对象；而当处于在线模式时，于**监控表**中点选某个对象后下鼠标右键，之后于快捷选单中便会出现与程序监控画面相同的操作项，利用这些功能便可对监控对象的状态或内容值进行变更。



依据本节所介绍的内容，对于我们所建立的范例程序便可进行简易的测试与除错，进一步的测试与侦错功能，**第 18 章**的内容当中有更详尽的介绍。

MEMO

4

5

第5章 POU 与 TASK

目录

5.1	关于程序组织单元 (POU) 与工作 (TASK)	5-2
5.2	认识程序组织单元 (POU)	5-3
5.2.1	POU 的编程架构与种类	5-3
5.2.2	ISPSOft 的 POU 架构	5-4
5.3	认识工作 (TASK)	5-6
5.3.1	ISPSOft 中的工作 (TASK) 管理	5-6
5.3.2	项目管理区的工作 (TASK) 项目	5-10
5.3.3	工作 (TASK) 中的 POU 执行顺序	5-13
5.4	POU 的管理操作	5-15
5.4.1	POU 的建立与属性设置	5-15
5.4.2	POU 的启动状态	5-20
5.4.3	POU 的删除与复制	5-21
5.4.4	POU 密码的设置与解除	5-23
5.4.5	导出程序 POU	5-24
5.4.6	导入程序 POU	5-25
5.4.7	导出功能块 POU	5-26
5.4.8	导入功能块 POU	5-27
5.4.9	POU 的目录管理	5-28
5.5	工作 (TASK) 管理的操作	5-29
5.5.1	工作 (TASK) 属性与中断条件的设置	5-31
5.5.2	POU 的配置	5-36
5.5.3	POU 的排序	5-38
5.6	范例实作	5-39
5.6.1	POU 与 TASK 的编程范例	5-39
5.6.2	DVP 机种的中断程序范例	5-43
5.6.3	AH5x0 机种的中断程序范例	5-48

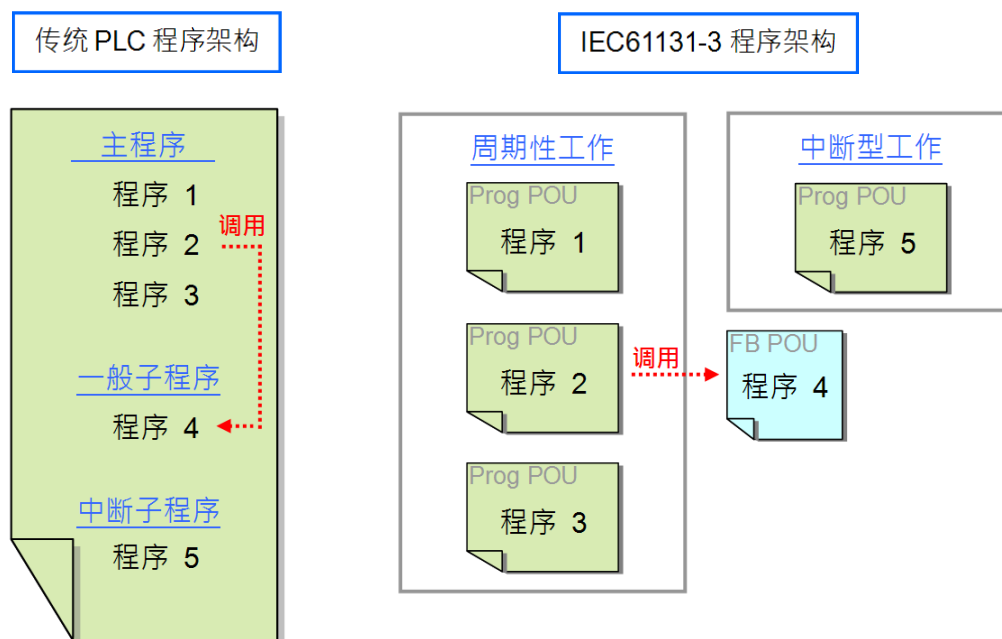
5.1 关于程序组织单元 (POU) 与工作 (TASK)

程序组织单元 (Program Organization Unit, POU) 与工作 (TASK) 是 IEC61131-3 中相当重要的编程观念。此二者的最大意义在于让 PLC 的编程由传统的程序撰写提升至项目管理的层次。下表即为传统梯形图与 IEC61131-3 在程序架构上的分别。

传统 PLC 程序架构	IEC61131-3 程序架构
主程序	1. 将单一的程序切割为许多独立的程序单元 (Prog POU)。 2. 将建立的程序单元指定为周期性的工作 (TASK)。
一般子程序	1. 将子程序建立为功能块 (FB POU)。 2. 在其它的程序单元中对建立好的功能块 (FB) 进行调用。
中断子程序	1. 为不同的中断程序, 分别建立独立的程序单元 (Prog POU)。 2. 将建立好的程序单元指定至对应的中断型工作 (TASK) 中。

下列即为两种程序架构的示意图, 很明显的可看出传统的 PLC 编程必须从头到尾的将程序撰写在同一份原始码中, 而 IEC61131-3 的程序架构则是将程序切割为若干个程序单元, 且每个程序单元皆可独立开发, 之后再透过工作管理的方式将各个程序单元分配到适当的执行位置。

5



当控制系统到达一定的规模时, 其内部的控制程序必定也会有相当的复杂度, 且程序与程序之间的互动也势必更加紧密, 此时若仍采用传统的 PLC 程序架构, 不论是开发过程或是日后的维护, 必定会付出相当大的时间与人力; 而在 IEC61131-3 的架构中, 藉由 POU 的概念便可让所开发的程序具有高重用性与高模块化的特性, 且透过 TASK 的管理更可以让程序单元之间轻易的达到整合。而在接下来的几个小节中, 我们将陆续介绍 POU 与 TASK 的相关观念, 并说明如何在 ISPSOft 当中进行实作。

5.2 认识程序组织单元 (POU)

5.2.1 POU 的编程架构与种类

程序组织单元 (POU) 是建构 PLC 程序的基本元素, 有别于传统的 PLC 程序, 由 IEC61131-3 所导入的程序架构, 其特色在于将原本一大段的程序分割成若干个小单元, 而这些小单元便称之为 POU。如上一节的内容, 传统的 PLC 程序是将所有的程序, 包括子程序都写在同一段程序代码当中, 当程序愈来愈庞大时, 对于程序的维护与除错都会是相当大的负担; 而 POU 的编程方式则是将程序依功能或特性, 分割成若干个小单元, 如此对于程序的开发与后续的维护都可以带来相当大的便利; 除此之外, 因为其模块化的特性, 不同的 POU 亦可交由不同的设计者来进行开发, 对于人力的分配与项目开发的进行也都能带来显著的效益。

在 ISPSOft 中的 POU, 依功能特性共可分为程序 (PROG)、功能块 (FB) 与函数 (FC) 三种。

- 程序 (Program, PROG)

程序 POU 可依据其指定的工作 (TASK) 类型而决定其扮演的角色; 若被指定至周期性的工作时, 该程序 POU 扮演的便是主程序的角色; 而若被指定至中断型的工作时, 该程序 POU 便扮演中断子程序的角色; 此外, 在程序 POU 当中可以对功能块 (FB) 进行调用。

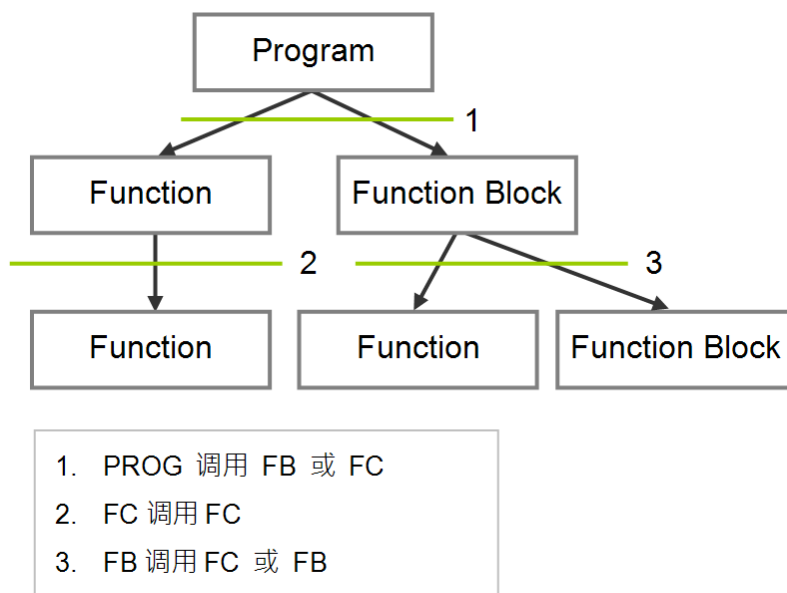
- 功能块 (Function Block, FB)

功能块 (FB) 内部可以宣告静态型 (有记忆性的) 的变量符号, 因此变量符号的值在运算之后可以被保持住, 即使输入相同的参数值, 响应的数值仍会依据功能块目前记忆的数值而与输入的数值一起运算后再输出, 所以输入相同的参数值, 响应的数值不一定是一样的。此外, 功能块的内部可以再调用其它功能块。

- 函数 (Function, FC)

每一个函数都是一个独立封装的程序单元, 可以为其定义输入参数, 用于值操作。相较于程序和功能块, 函数没有固定的内存空间, 所以不会储存上一次的运算结果, 也就使得同一函数在输入参数相同的情况下, 产生的结果相同, 例如加法运算, 目前只有 AS5xx 与 DVPxxMC 机种提供此功能。

下图为各 POU 的调用对应关系。

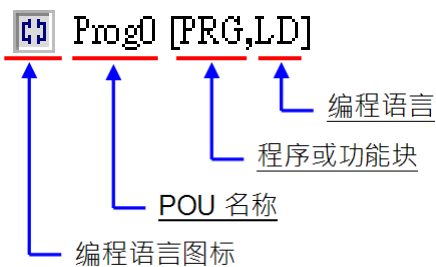
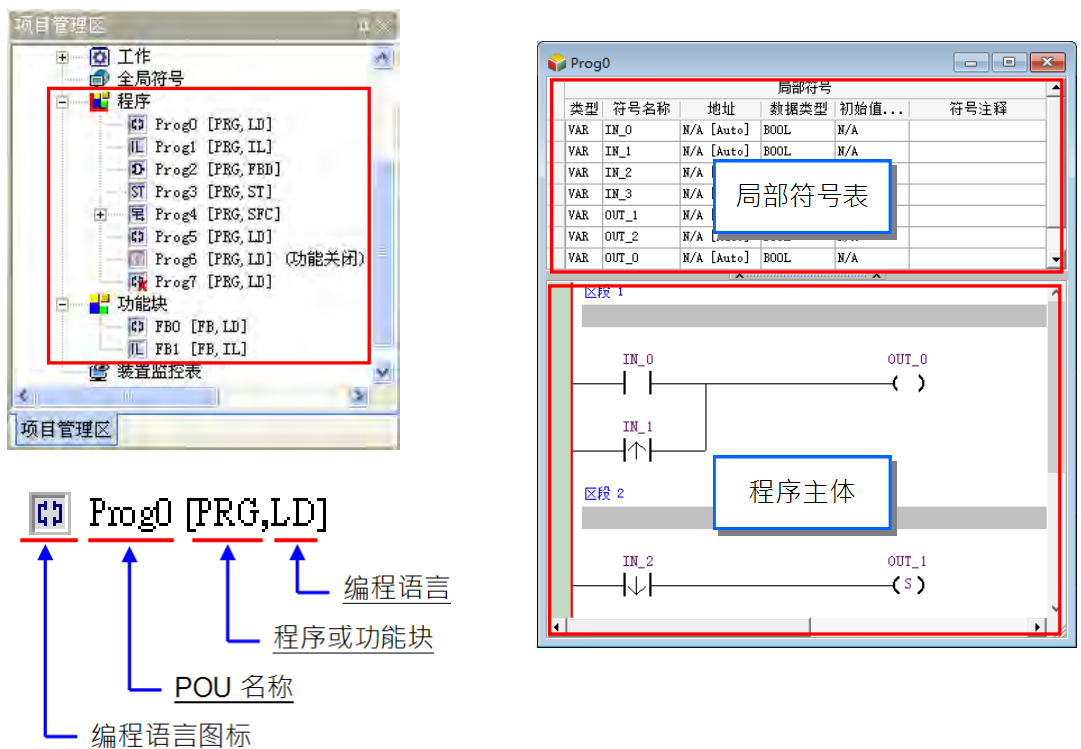


5.2.2 ISPSOft 的 POU 架构

5

所有用户所建立的 POU 对象皆会在项目管理区中被列出，而其中程序类型与功能块类型的 POU 会被分开管理；此外，根据该 POU 对象的编程语言及状态，所显示的图标也会有所不同，而在每个 POU 名称的后方亦会一并带出相关的信息。

在项目管理区的 POU 对象上双击鼠标左键即可打开该 POU 的编辑窗口，而窗口主要分为上下两个部份，上半部为此 POU 的局部符号表，而下半部则为程序的主体，且随着编程语言的不同，下半部的编辑环境也会随之不同。关于变量符号表与程序编辑的相关说明，将于后续章节当中另行介绍。



下表则为各种 POU 图标的相关说明。

图标	说明
	代表此对象为使用 梯形图 (LD) 所开发的 POU
	代表此对象为使用 指令列表 (IL) 所开发的 POU
	代表此对象为使用 功能块图 (FBD) 所开发的 POU
	代表此对象为使用 结构化语言 (ST) 所开发的 POU
	代表此对象为使用 顺序功能图 (SFC) 所开发的 POU
	代表此对象为使用 连续功能图 (CFC) 所开发的 POU
	代表此对象为使用 C 语言 (C) 所开发的 POU
	当图标呈现灰阶代表此 POU 的状态为「 关闭 」, 而被设为关闭的 POU 将在编译的过程中被略过, 当然也不会被执行。
	当图标出现红色叉号代表此 POU 尚未被指定工作 (TASK), 而未被指定工作 (TASK) 的 POU 将在编译的过程中被略过, 当然也不会被执行。

*.关于 POU 的启动状态将于第 5.4.2 节中另行介绍, 而工作 (TASK) 的指定则请参考下一节的说明。

5.3 认识工作 (TASK)

工作 (TASK) 的意义在于赋予各个程序 POU 一个明确的执行任务，并指定每个程序 POU 之间的执行顺序或是启动方式，而在同样的应用场合，工作 (TASK) 也有被称作任务 (TASK)。

换句话说，把各个程序 POU 比喻为球员的话，那么工作 (TASK) 便类似于球员所担任的位置或是球场上的阵型，而决定着球员之间的配合模式。

5.3.1 ISPSOft 中的工作 (TASK) 管理

在 ISPSOft 中，一个项目中的程序 POU 并非都会参与实际的执行，而是必须在指派 TASK 之后才可确定该 POU 是否执行以及如何执行；当 POU 未被指派 TASK 时，该 POU 仅会被当做一般的原始码保存在项目，本身并不会被编译为 PLC 的执行码。只有程序 POU 需要被指派 TASK，至于功能块 (FB) 的执行方式，则是依据调用它们的上位 POU 程序而定。

IEC61131-3 程序架构的最大特色便是将程序切割成若干个独立开发的 POU，而最后编译时则会将所有 POU 程序重新排列组合为一个可逐步扫描的执行码，其排列组合的依据便是 TASK 的配置状态。

在 ISPSOft 中，一个程序 POU 只能指派给一个 TASK，只有 AS5xx 与 DVPxxMC 机种可以一个程序 POU 指派给多个 TASK，但同一个 TASK 却可配置一个以上的 POU，且所配置的 POU 也可在该 TASK 中指定执行的顺序；当用户新增一个程序类型的 POU 时，就必须为其指派一个预设的 TASK。

5

适用机种	AS/AH/DVP	AS5xx/DVPxxMC
工作管理	一个程序 POU 只能指派给一个 TASK	一个程序 POU 可以指派给多个 TASK
运行方式	周期性、定时中断以及条件中断	自由运行、周期以及事件

AS、AH 以及 DVP 机种的 TASK 运行方式大致可分为周期性、定时中断以及条件中断等 3 种。

● 周期性

被指派至周期性 TASK 的程序 POU，其执行方式便是单纯的来回扫描。在 DVP 系列的机种中，因其系统设置只有一个周期性的 TASK，因此所有周期性扫描的 POU 皆须被分配至此 TASK；而 AH/AS 系列的机种则有 32 个周期性的 TASK 可供选择，其编号为 0~31，扫描顺序则以编号小的为优先，且 AH/AS 系列机种尚有支持控制 TASK 的 API 指令 (TKON 与 TKOFF)，用户可利用该指令于 PLC 的运行过程中任意启动或暂停某个 TASK 的执行。

以 SFC 语言编辑的 POU 只能被配置于周期性的 TASK 中，而不可配置于中断型的 TASK。

● 定时中断

被指派至定时中断 TASK 的程序 POU，其功能类似定时中断子程序，当定时中断的时间到达后，被分配至该 TASK 的所有 POU 便会依照排列顺序执行一遍。定时中断 TASK 的数量会依据选用的 PLC 机种而定，该机种支持几组定时中断源，系统便会有几组定时中断的 TASK。

● 条件中断

条件中断亦可区分为多种类型，如外部中断、I/O 中断、计数中断...等，而不同机种所提供的中断服务与数量皆会有所不同，因此规划前必须先确认所选用的项目机种支持哪些中断模式。当程序 POU 被分配至条件中断 TASK 时，其功能便类似中断子程序，一旦中断的条件成立，如计数到达设定值，则被分配至该 TASK 的所有 POU 便会依照排列顺序执行一遍。

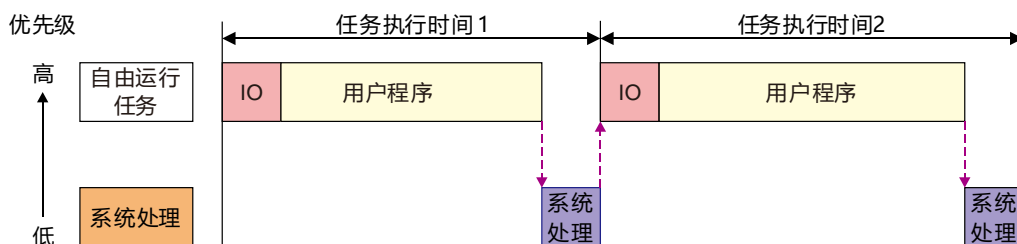
针对 AS5xx 与 DVPxxMC 机种，一个程序 POU 可以指派给多个 TASK，且同一个程序 POU 可以重复指派给同一个 Task，而一个 TASK 可配置一个以上的 POU，且所配置的 POU 也可在该 TASK 中指定执行的顺序，而 AS5xx 与 DVPxxMC 机种 TASK 运行方式，分为自由运行、周期以及事件 3 种，接下来就 TASK 运行方式及 TASK 执行优先级做介绍。

● TASK 运行方式

(1) 自由运行

程序一开始运行工作 (TASK) 就被处理，一个运行周期结束后，工作 (TASK) 将在下一个周期中被重新启动。

▶自由运行工作 (TASK) 执行方式如下：

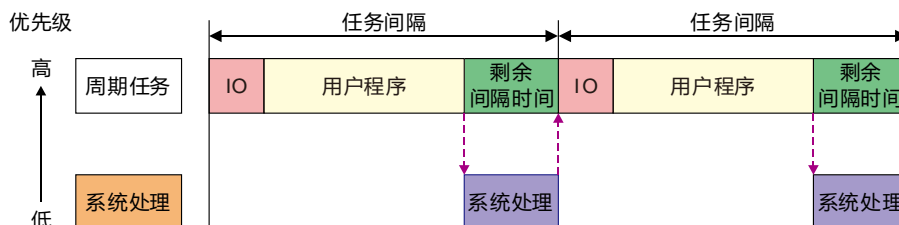


注意：自由运行工作没有固定持续时间，所以上图中的任务执行时间 1 和任务执行时间 2 可能不相同。

(2) 周期

按照工作 (TASK) 管理工具内的工作属性的「间隔时间」所设定的时间执行。

▶周期工作 (TASK) 执行方式如下：



IO：表示 I/O 更新。I/O 包括本机 I/O 点和左右侧扩充模块数据、CANopen 数据。此数据可以指定在设定的工作 (Task) 执行前更新；未指定时，此数据在系统处理中更新。

用户程序：表示用户程序的执行，按照工作 (Task) 中程序分配循序执行。

剩余间隔时间：控制器执行系统处理；如果有低优先顺序的工作 (Task)，则会先执行低优先顺序的工作 (Task)，然后再执行系统处理。

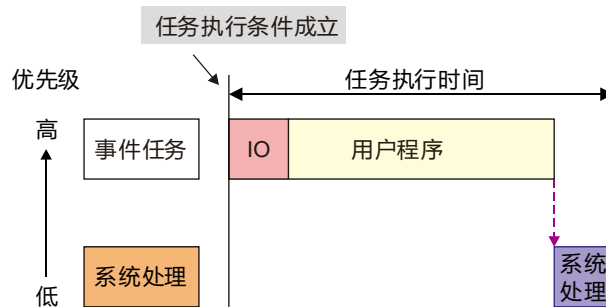
系统处理：控制器执行完所有工作 (Task) 请求后执行系统处理；系统处理包括 Ethernet 通讯、RS232、RS485 通讯处理等。

注意：如果为某个周期工作 (Task) 定义的周期过短，则该工作 (Task) 会在执行完使用者程序后立即重复执行，而不会执行其它低优先顺序的工作 (Task) 或任何系统处理，这将会影响所有工作 (Task) 的执行。如果为该工作 (Task) 设定了程序扫描超时定时器，则会导致程序扫描超时定时器超时，控制器会进入 **Error** 状态，并且停止执行用户程序；如果没有为工作 (Task) 设定程序扫描超时定时器，控制器将无法执行系统处理，出现通讯超时等问题。

(3) 事件

事件工作 (Task) 是指仅在指定的事件发生时执行一次。事件工作 (Task) 的执行时机取决于事件发生的时机和事件工作 (Task) 的优先顺序。

➤事件工作 (TASK) 执行方式如下：



➤可供选择的事件包含以下几种触发方式：

- 运动事件 (运动控制工作 (Task))；
- 本机输入点 (I0~I7、I10~I17) 的上升沿或下降沿；
- CANopen 网络的同步信号；
- 增量型编码器 1 或增量型编码器 2 的 Z 脉冲信号的上升沿；

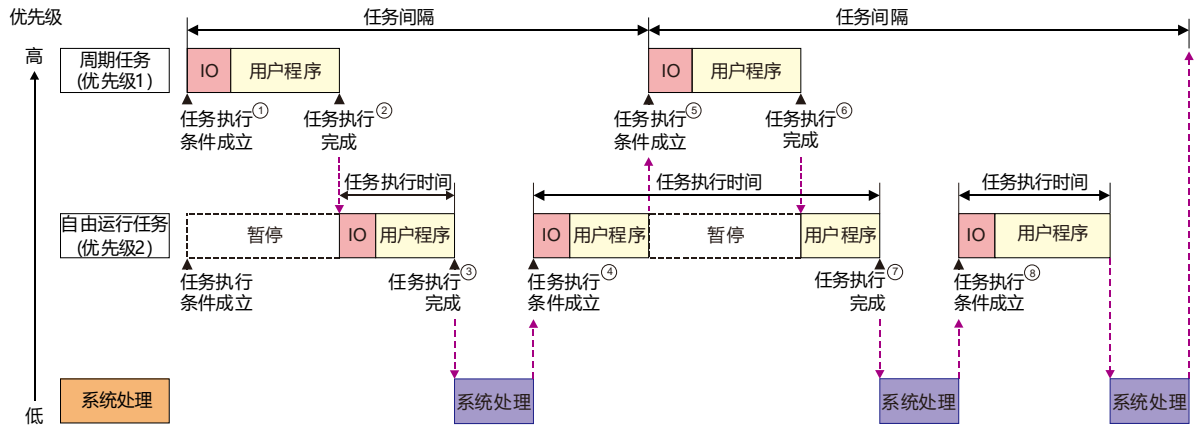
事件工作 (Task) 完成前，该事件工作 (Task) 的执行条件再次成立时，忽略第 2 次的执行条件成立。事件工作 (Task) 完成前是指正在执行的事件工作 (Task) 或正在等待执行的事件工作 (Task)。

● TASK 执行优先顺序

控制器无法同时执行多个工作 (Task)，每个工作 (Task) 须分配一个优先顺序，各工作 (Task) 按照设定的优先顺序执行。优先顺序可以设定为 1 到 24 (1 表示最高优先顺序，24 表示最低优先顺序)，每个工作 (Task) 的优先顺序必须唯一。优先执行优先顺序较高的工作 (Task)，高优先顺序的工作 (Task) 可以中断低优先顺序的工作 (Task)。

建议将高即时要求的工作 (Task) 设定为高优先顺序，将低即时要求的工作 (Task) 设定为低优先顺序。在 ISPSOft 中默认建立的运动控制工作 (Task) 预设被配置为优先顺序 1。

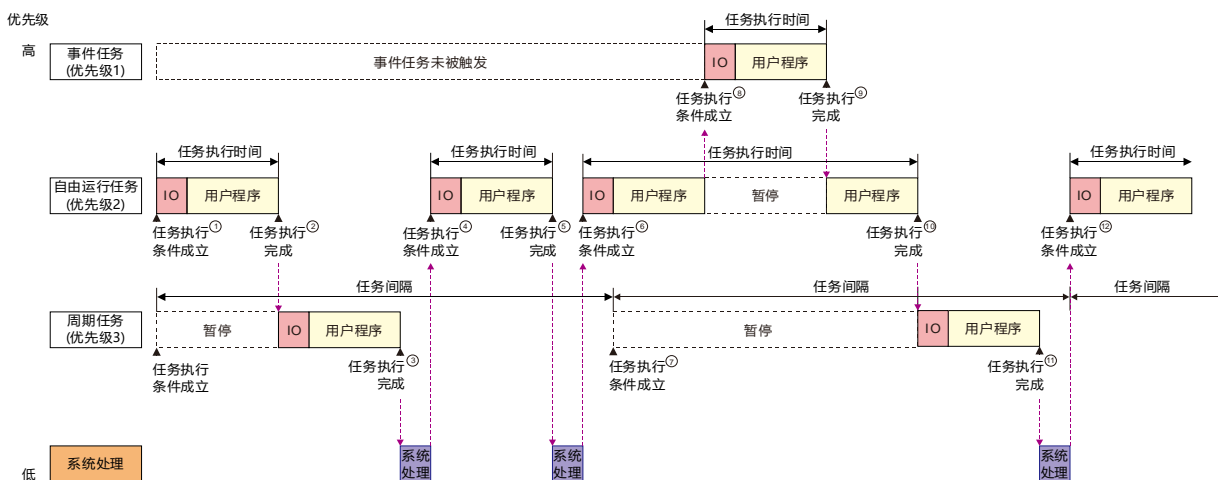
►两个工作 (Task) 同时执行时 (周期和自由运行)



- ① 周期工作 (Task) 和自由运行工作 (Task) 执行条件同时成立，因为周期工作 (Task) 的优先顺序高，所以先执行周期工作 (Task)；
- ② 周期工作 (Task) 执行完成，自由运行工作 (Task) 开始执行；
- ③ 自由运行工作 (Task) 执行完成，由于没有其它工作 (Task) 请求，控制器执行系统处理；
- ④ 因为高优先顺序的周期工作 (Task) 请求还没有到来，所以继续执行自由运行工作 (Task)；
- ⑤ 自由运行工作 (Task) 执行过程中，高优先顺序的周期工作 (Task) 请求到来，周期工作 (Task) 打断自由运行工作 (Task)，控制器执行周期工作 (Task)；
- ⑥ 周期工作 (Task) 执行完成，控制器继续执行低优先顺序的自由运行工作 (Task) 未执行的部分；
- ⑦ 自由运行工作 (Task) 执行完成，由于没有其它的工作 (Task) 请求，控制器执行系统处理；
- ⑧ 系统处理完成，因为没有高优先顺序的周期工作 (Task) 请求，所以继续执行自由运行工作 (Task)。

5

►三个工作 (Task) 混合执行时 (事件、自由运行、周期)



- ① 自由运行工作 (Task) 和周期工作 (Task) 执行条件同时成立，因为自由运行工作 (Task) 的优先顺序高，所以先执行自由运行工作 (Task)；

- ② 自由运行工作 (Task) 执行完成，周期工作 (Task) 开始执行；
- ③ 周期工作 (Task) 执行完成，由于没有其它的工作 (Task) 请求，控制器执行系统处理；
- ④ 系统处理完成，执行自由运行工作 (Task)；
- ⑤ 自由运行工作 (Task) 执行完成，由于没有其它的工作 (Task) 请求，控制器执行系统处理；
- ⑥ 系统处理完成，执行自由运行工作 (Task)；
- ⑦ 周期工作 (Task) 执行条件成立，因为周期工作 (Task) 的优先顺序低于自由运行工作 (Task)，所以自由运行工作 (Task) 继续执行，周期工作 (Task) 等待；
- ⑧ 事件工作 (Task) 执行条件成立，因为事件工作 (Task) 的优先顺序最高，所以事件工作 (Task) 中断自由运行工作 (Task)；
- ⑨ 事件工作 (Task) 执行完毕，控制器继续执行低优先顺序的自由运行工作 (Task) 未执行的部分；
- ⑩ 自由运行工作 (Task) 执行完毕，⑦ 的周期工作 (Task) 请求还未回应，控制器执行周期工作 (Task)；
- ⑪ 周期工作 (Task) 执行完毕由于没有其它的工作 (Task) 请求，控制器执行系统处理。

5.3.2 项目管理区的工作 (TASK) 项目

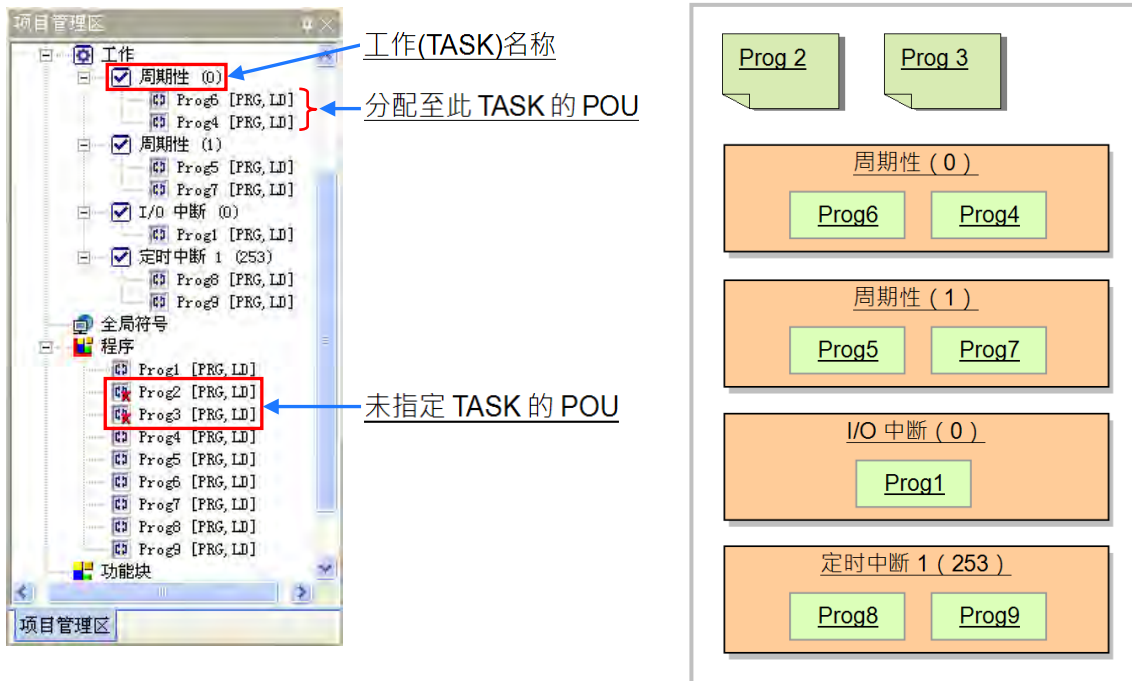
5

在项目管理区的「工作」项目下会列出所有 POU 的配置状况，而未被分配任何 POU 的 TASK 项目则不会被列出；每个 TASK 项目下的 POU 排列顺序则代表了各 POU 被执行的顺序。

以下图的 AH 机种项目为例，项目中总共建立了 9 个程序 POU，且其 TASK 的配置状况如下。

工作 (TASK) 名称	POU	执行方式
未分配	Prog2、Prog3	因未被指派 TASK，因此这两个 POU 并不会被执行。
周期性 (0)	Prog6、Prog4	因扫描顺序会以编号小的周期性 TASK 为优先，因此当程序执行时便会以 Prog6 → Prog4 → Prog5 → Prog7 的顺序来回扫描。
周期性 (1)	Prog5、Prog7	
I/O 中断 (0)	Prog1	当该中断条件成立时，Prog1 就会被执行一次。
定时中断 1 (253)	Prog8、Prog9	定时中断 1 会每隔一段固定时间就被触发一次，而分配至此 TASK 的 Prog8 → Prog9 也会依序执行一次。

*.关于中断条件的设置将在第 5.5 节中稍做介绍，而更详细的说明请参考各机种的使用手册。



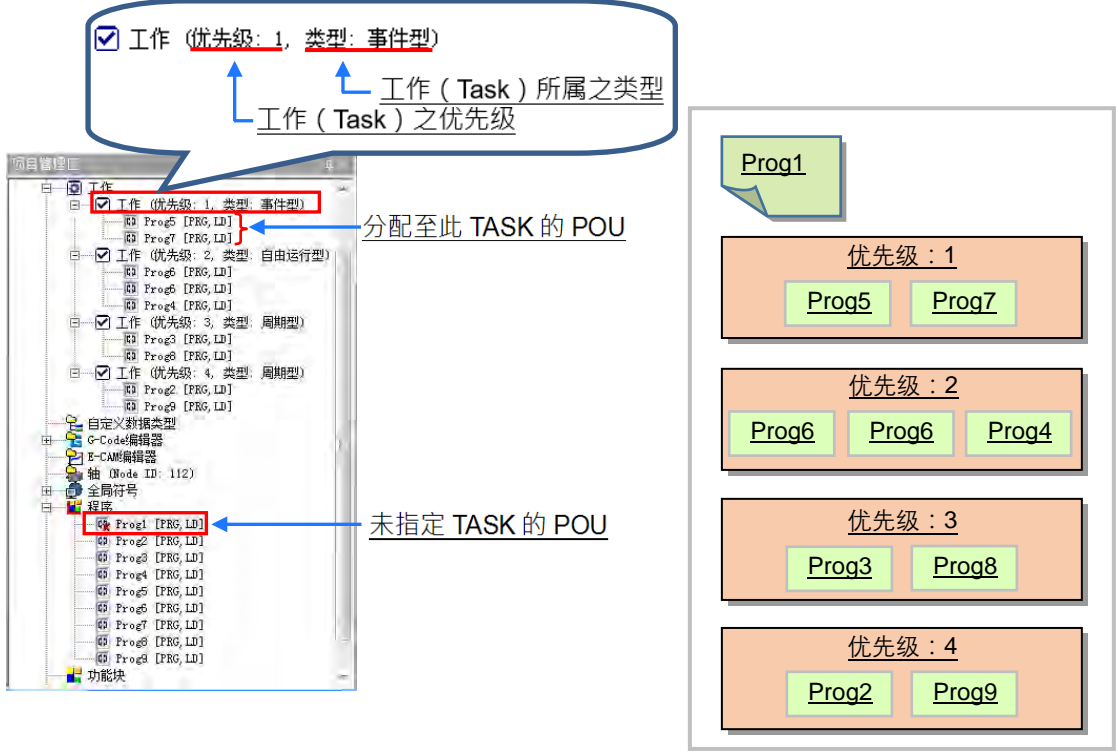
欲启动中断功能时，仍必须于周期性的程序中插入 API 应用指令「EI」方可正常执行中断程序。



5

而 AS5xx 与 DVPxxMC 机种在工作 (TASK) 的名称显示上稍有不同，增加了优先级与类型两种，下面项目中总共建立了 9 个程序 POU，且其 TASK 的配置状况如下。而工作 (TASK) 优先级顺序及工作 (TASK) 型态共同决定工作 (TASK) 执行的顺序。

工作 (TASK) 优先级	工作 (TASK) 类型	POU	执行方式
未分配		Prog1	因未被指派 TASK，因此这个 POU 并不会被执行。
优先级：1	事件型	Prog5、Prog7	当程序执行时便会以 Prog5 → Prog7 的顺序来回扫描。
优先级：2	自由运行型	Prog6、Prog6、Prog4	当程序执行时便会以 Prog6 → Prog6 → Prog4 的顺序来回扫描。
优先级：3	周期型	Prog3、Prog8	当程序执行时便会以 Prog3 → Prog8 的顺序来回扫描。
优先级：4	周期型	Prog2、Prog9	当程序执行时便会以 Prog2 → Prog9 的顺序来回扫描。



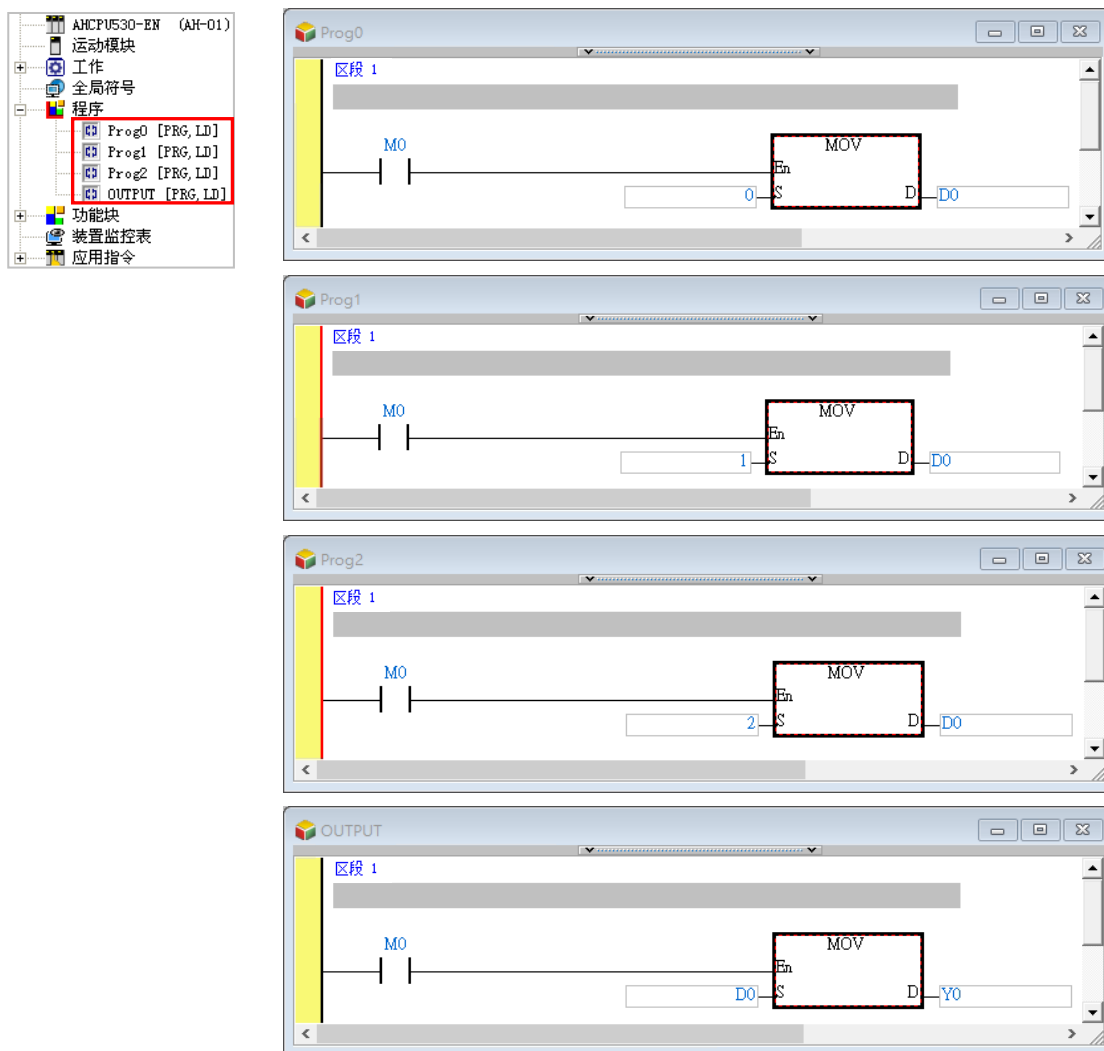
5

5.3.3 工作 (TASK) 中的 POU 执行顺序

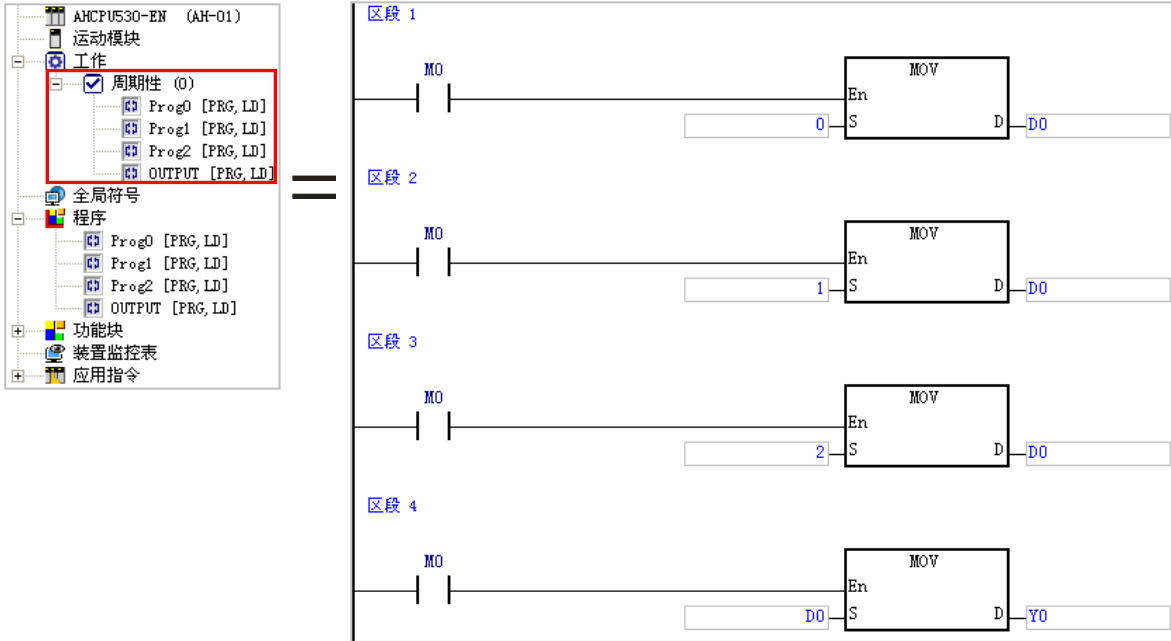
同一个工作 (TASK) 项目下可以指定一个以上的 POU，而这些 POU 的排列顺序则会决定实际执行时的顺序。请参考下列范例，该项目共有 4 个 POU，而各 POU 的功能则简述于下表。

POU 名称	功能说明
Prog0	当 M0 为 ON 时，将装置 D0 的值设置为 0。
Prog1	当 M0 为 ON 时，将装置 D0 的值设置为 1。
Prog2	当 M0 为 ON 时，将装置 D0 的值设置为 2。
OUTPUT	当 M0 为 ON 时，依据目前 D0 的值决定 Y0 的输出。

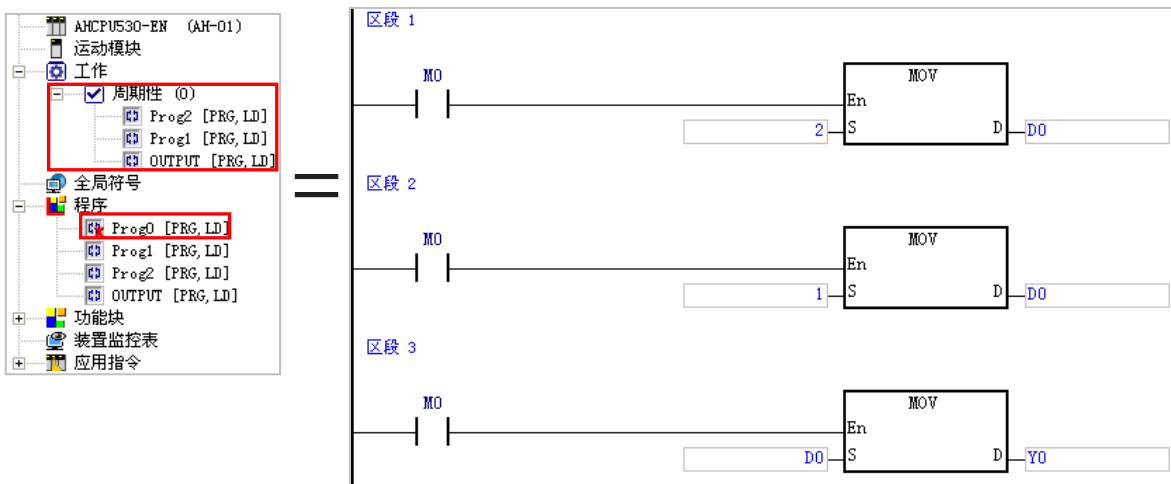
各个 POU 内的程序则请参考下图。



假设 4 个程序 POU 都被指派至周期性 (O) 的工作 (TASK) 中，且各 POU 的排序如下图所示，则其执行顺序便为 Prog0 → Prog1 → Prog2 → OUTPUT；而这也表示执行至「OUTPUT」时的 DO 值将决定于「Prog2」的程序，因此当 M0 为 ON 时，执行至「OUTPUT」时的 DO 值便会等于 2，而将 DO 的值搬移给 Y0 后，Y0 的值也会为 2，而这也代表在一个扫描周期结束后，实际输出的 Y0 通道中只有 Y0.1 为 ON。下图即为其等效的程序。



但若我们将「Prog2」移至最上方，且将「Prog0」由工作 (TASK) 清单中删除，则「Prog0」中的程序便不会被编译为执行码，而执行的顺序则会变为 Prog2 → Prog1 → OUTPUT，因此程序执行至「OUTPUT」时的 DO 值将会决定于「Prog1」的程序，而当 M0 为 ON 时，Y0 的值便会等于 1，亦即最后输出的 Y0 通道中只有 Y0.0 为 ON。



5.4 POU 的管理操作

ISPSOft 所支持的 POU 种类分为程序、功能块及函数，其中函数的用法大致上与功能块相同可参考功能块相关介绍。下列表格是各机种所支持的 POU 种类。

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx/ DVPxxMC
程序	V	V	V	V	V	V	V
功能块	V	V	V	V	V	V	V
函数							V

5.4.1 POU 的建立与属性设置

(1) 项目管理区中单击要建立的项目，若要建立程序类型的 POU 请单击「程序」，若要建立功能块类型的 POU 则单击「功能块」。单击后鼠标右击并单击 **新增 (N)**。

● 新增程序 POU



● 新增功能块 POU



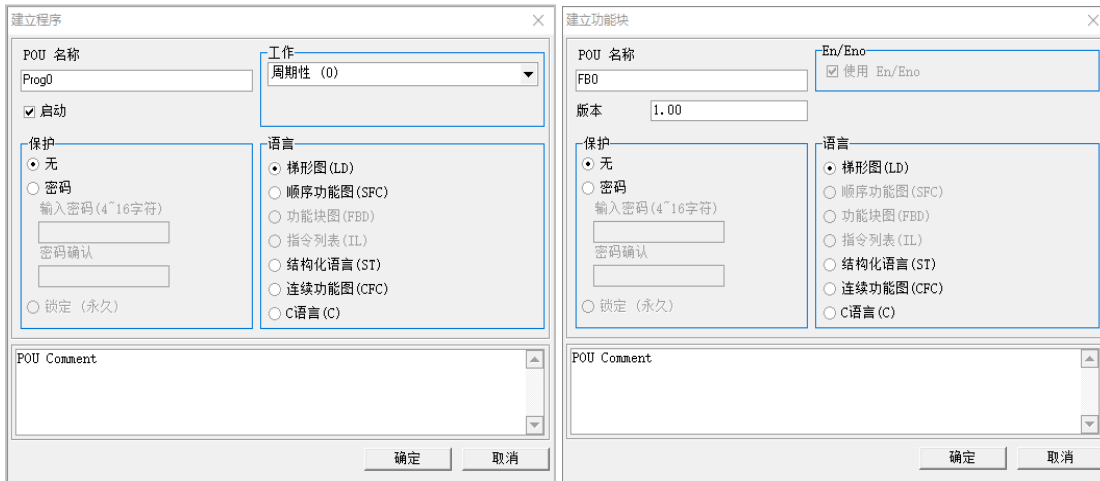
● 新增函数 POU



(2) 随后出现的窗口中设置该 POU 的属性，而虽然程序类型与功能块类型的 POU 属性有些许差异，但大致上设置是相同的，在程序 POU 的属性内，工作 (Task) 栏位的下拉式选项内容随选择不同机种项目可能会有所不同。

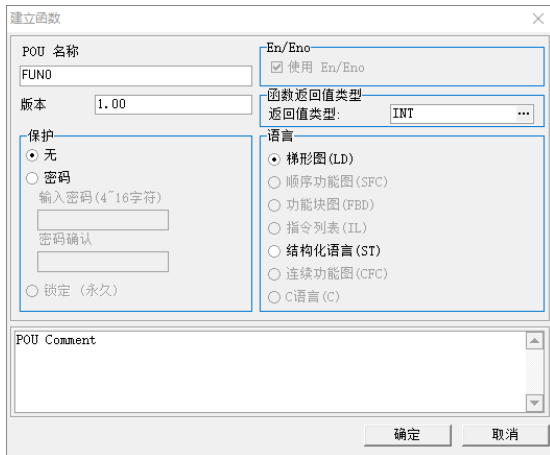
● 程序 POU

● 功能块 POU



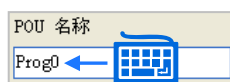
5


● 函数 POU

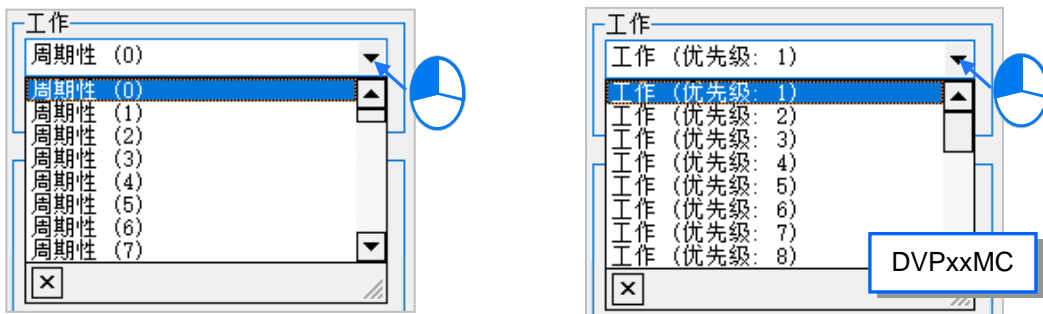


● **POU 名称**：在此处输入 POU 的名称，而其命名的规则如下。

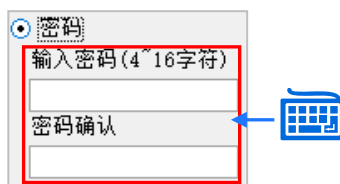
- (a) 不可使用特殊字符或空白，例如 *、#、?、\、%、@ ... 等。
- (b) 最多只可输入 30 个字符，且须注意一个中文字将会占用两个字符。
- (c) 名称当中可使用下划线，但不可连续使用，或是置于结尾，例如 **"POU_1"** 是合法的名称，但 **"POU__1"** 与 **"POU_1_"** 则是不合法的名称。



- **函数返回值类型**：建立的为函数类型的 POU，则需要设定函数定义返回值的数据类型，也可通过  按钮进行选择，若设为 VOID，表示该函数没有返回值，即是没有输出参数。
- **工作**：若建立的为程序类型的 POU，则可于此处预先指定此 POU 的工作 (TASK) 类型。
下图左侧为一般机种，如果项目机种为 DVPxxMC 或 AS5xx 在显示的内容会有所不同，如下图右侧。



- **密码**：要设置密码时，请先后在此二字段中输入相同的密码，输入的字符可为英文字母、数字或符号。

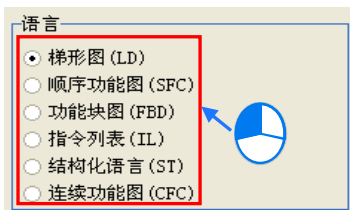


5

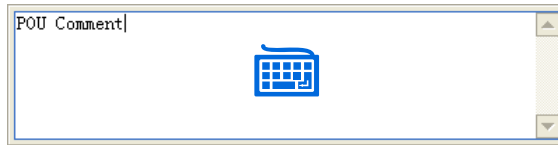
- **锁定 (永久)**：要将功能块永久锁定时，可使用此选项，锁定后将无法开启功能块，建议使用此选项前先将项目做备份；此选项要在新增完功能块 POU 后，在该 POU 项目上点击鼠标右键并点击属性，才可选择。



- **语言**：此选择开发此 POU 的编程语言。(编程语言选择内容，可能随机种有所变化)



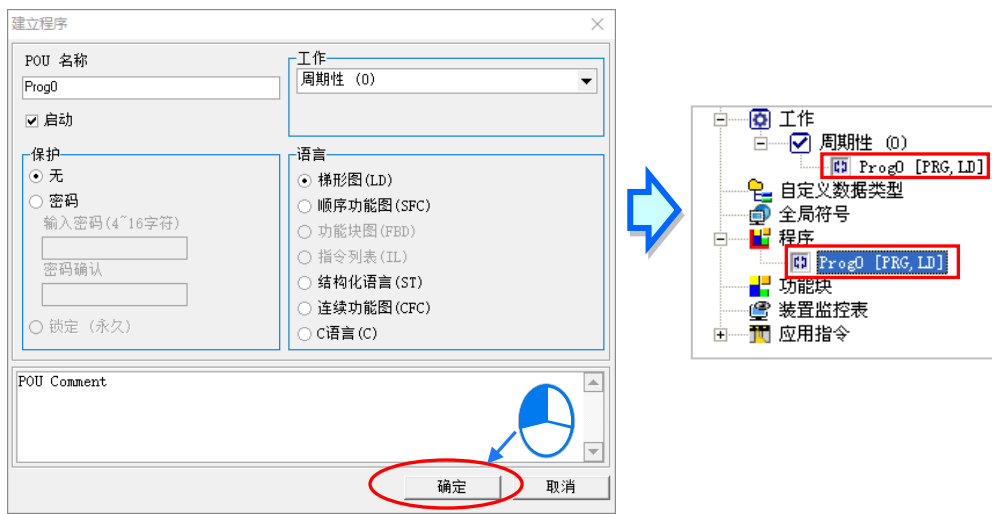
- **批注**：要加入批注时，请直接于下方空白处输入批注即可。



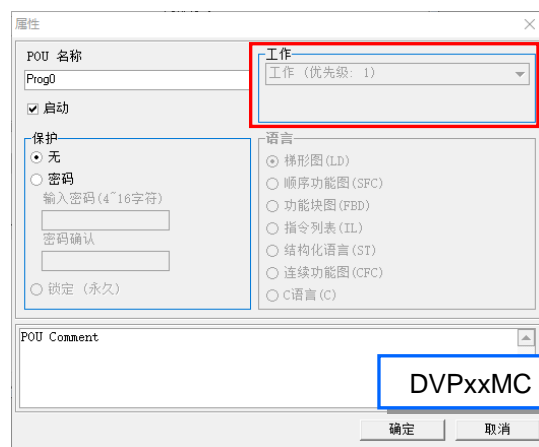
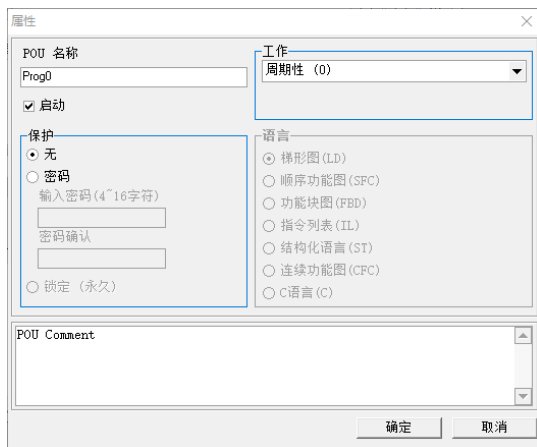
*.若建立的为程序类型的 POU，则在窗口中还会有一个「启动」的选项，关于此功能将于第 5.4.2 节的内容当中另行介绍。

- (3) 完成设置并按下「确定」后，项目管理区的「程序」或「功能块」下便会列出所建的 POU；而若建立的是程序类型的 POU 时，因已为其指定默认的工作 (TASK) 类型，因此于「工作」下亦会将该 POU 列出。

5



- (4) 欲修改已建立的 POU 属性时，于该 POU 项目上点击鼠标右键并点选 **属性**，之后便可参考前面的叙述进行设定，使用属性编辑是无法变更编程的语言，如果项目机种为 AS5xx 或 DVPxxMC 机种，如下图右侧使用属性编辑，是无法从此处变更工作 (Task)。



5

5.4.2 POU 的启动状态

ISPSOft 提供可将某个 POU 暂时关闭的功能，当 POU 被设为关闭状态时，该 POU 便会在编译的过程中被略过，且其内部的程序也不会被执行。不过此功能仅支持程序类型的 POU。

此功能可被用于进行调测或除错的过程，当在过程中想先略过某些程序，则在操作上便可暂时将该 POU 设为失效状态，待调测完毕后再将其恢复为启动状态即可。

如下图范例，该设备共有五个工作站，且其工作程序被分为 5 个程序 POU 来进行开发。

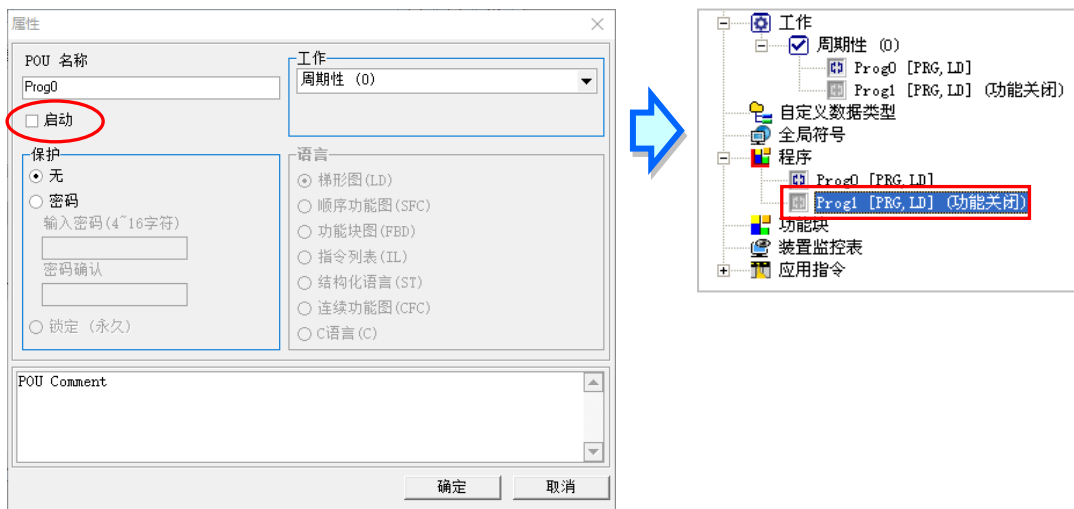


假设该设备已组立完毕，但测试仪器尚未完成架设，此时即可将「测试 A」与「测试 B」两个 POU 设为「未启动」状态，如此便可略过此两站的动作，而仅针对其它的部份进行调测。



5

欲设置 POU 的启动状态时，可在 POU 属性窗口中的「启动」项目进行勾选，取消勾选便代表将该 POU 的功能设为关闭，而处于关闭状态的 POU，在项目管理区中亦会以不同的图标来显示。

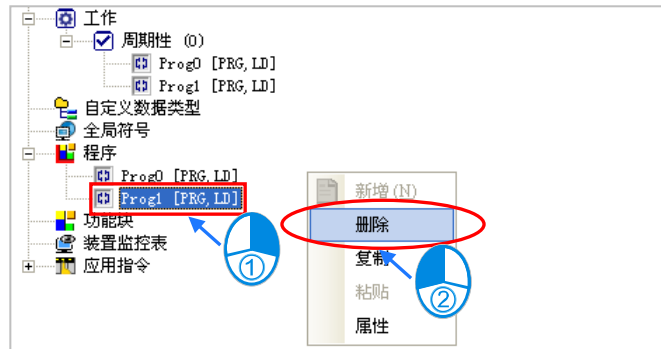


* 1. 虽然将 POU 由原本指定的 TASK 清单中删除也可达到关闭该 POU 功能的效果，但若之后要重新执行该 POU 的功能时，除了须再次将其分配至原本的 TASK 之外，还必须对该 TASK 清单中的 POU 重新进行排序，因此针对暂时性的 POU 功能关闭，利用此节介绍的功能来进行操作会相对的较为便利。

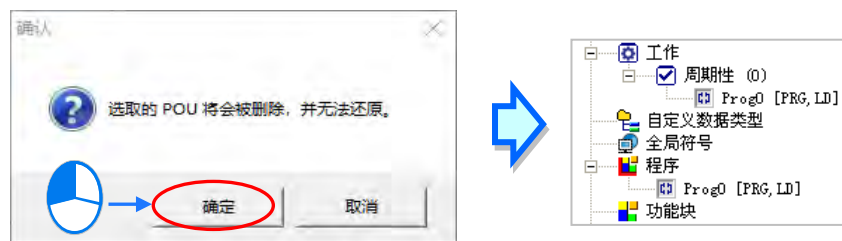
* 2. 除了可透过 POU 属性窗口来设置启动状态之外，藉由工作管理工具同样可进行设置，详细操作请参考第 5.5 节的介绍。

5.4.3 POU 的删除与复制

欲删除 POU 时，请在欲删除的对象上点击鼠标右键，接着在快捷选项中单击 **删除**。



POU 一旦被删除之后便无法再还原，确认之后按下「**确定**」键即可将所选的 POU 自项目中删除；而若删除的是程序类型的 POU，则系统亦会自动将其于工作 (TASK) 清单中剔除。



POU 也支持批次删除，请于项目管理区的**程序**或**功能块**项目上点击鼠标右键，接着快捷选项中单击「**POU 批量删除**」。选择窗口勾选欲删除的 POU 后按下「**确定**」按钮，也可使用「**全部选取**」、「**反向选取**」、「**全部取消**」按钮。再确认窗口按下「**确定**」按钮后完成 POU 的批次删除。



5

欲复制 POU 时，请于欲复制的对象上点击鼠标右键，并在快捷选项中单击 **复制**。



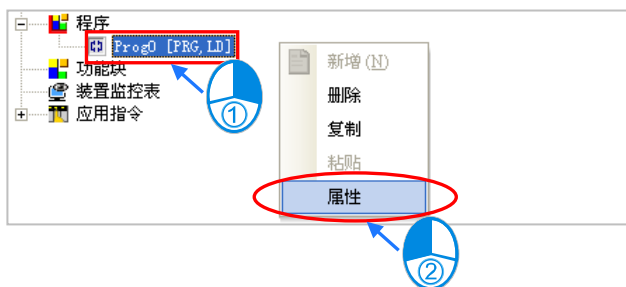
复制的目的位置可为同一个项目或其它项目，但不可将程序类型的 POU 复制至「功能块」下，而功能块类型的 POU 亦不可复制至「程序」下。欲贴上所复制的 POU 时，请于目的位置的「程序」或「功能块」上点击鼠标右键，并在快捷选项中单击 **粘贴** 即可；而若复制的 POU 为程序类型时，粘贴的 POU 将默认为未指派工作 (TASK) 的状态。

*.若粘贴的位置已存在相同名称的 POU 时，系统会自动在粘贴的 POU 名称后面加上 CopyOf_n (n 为流水号) 做为新名称。

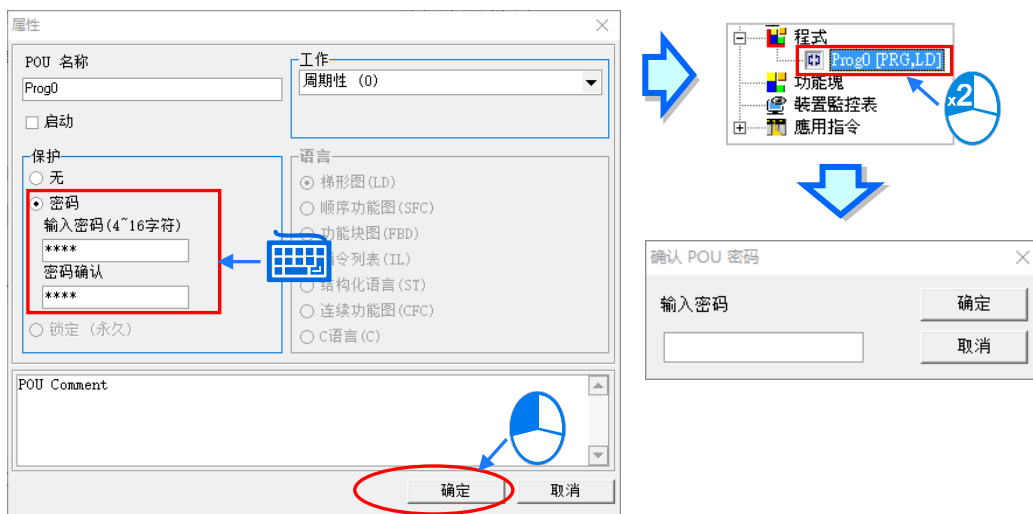
5.4.4 POU 密码的设置与解除

在 ISPSOft 中支持单独对各个 POU 设置密码的功能，而设置与解除的方式如下。

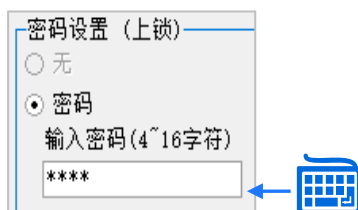
- (1) POU 的密码可在一开始新增该 POU 时便直接设置，但若一开始并未设置密码，而是在后续才要加上密码保护时，请先至项目管理区中，在欲设置密码的 POU 上鼠标右击，之后在快捷选项中单击 **属性** 以打开 POU 的属性窗口。



- (2) 在「密码设定」区先后输入两次密码，两次输入的密码必须一致，而输入的字符可为英数字或符号，完成之后按下「确定」键即可。一旦 POU 加上密码保护之后，每次打开该 POU 的窗口时，系统便会要求先输入密码以进行确认。



- (3) 欲解除密码时，请再次打开 POU 的属性窗口，之后再在「密码设定」区输入正确的密码，完成后按下「确定」键即可。



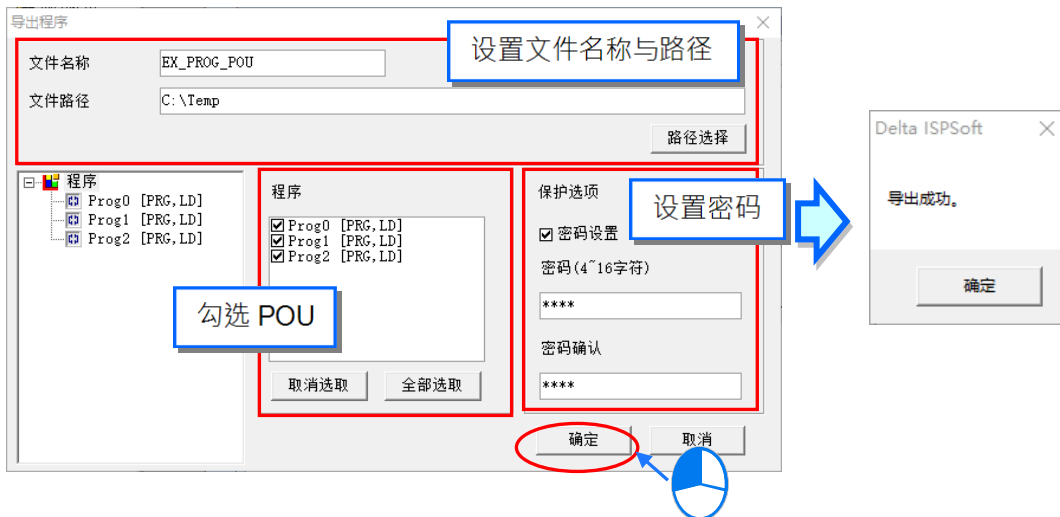
5.4.5 导出程序 POU

ISPSOft 支持将 POU 进行「导出」与「导入」的功能，不过在群组项目的架构下，每次则只能针对目前正在编辑的项目进行操作。

在菜单工具栏单击 **文件 (F) > 导出 > 导出程序单元**，之后便可直接对目前正在编辑的项目进行程序单元的导出操作；另外也可于**项目管理区**中单击目前正在编辑项目中的**程序**项目，之后鼠标右击并于快捷选项中单击 **导出程序单元**。



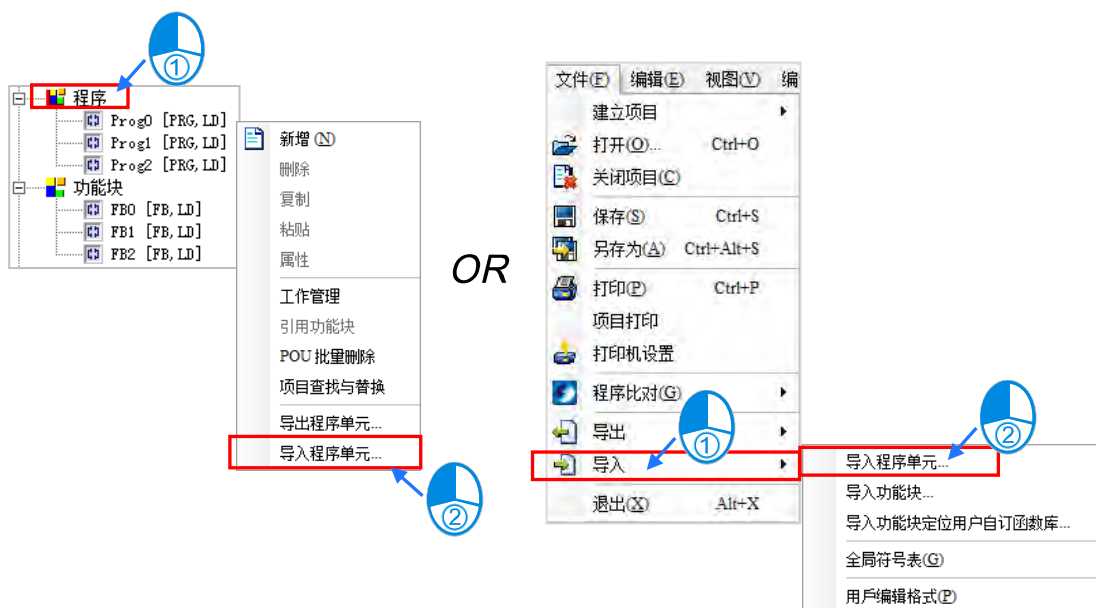
首先设置窗口中设置导出的文件名称及路径，并在「程序」清单中勾选欲导出的程序 POU，而若要为导出的文件加上密码保护时，请先勾选「密码设定」后，于两个密码字段中输入密码，且两个密码必须一致以确认输入无误，而输入的字符可为英数字或空白，完成后按下「确定」键即可。



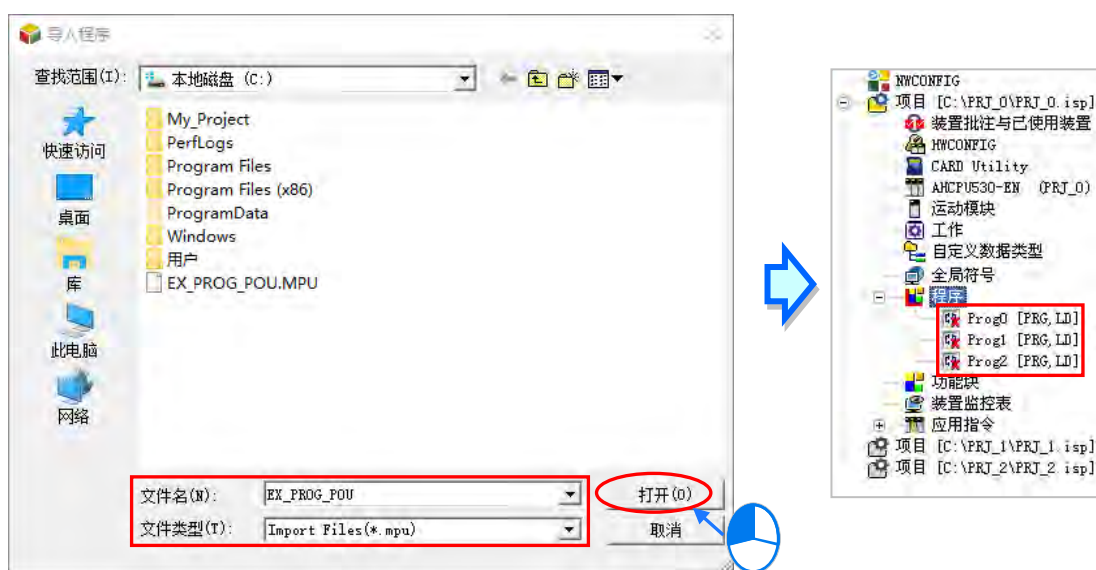
5.4.6 导入程序 POU

与「导出」的操作相同，进行「导入」时亦只能针对目前正在编辑的项目进行操作。此外，执行导入时，不可将导出的功能块 POU，重新导入至「程序」下。

于菜单工具栏单击 **文件 (F) > 导入 > 程序单元**，之后便可直接对目前正在编辑的项目进行程序单元的导入操作；另外也可于项目管理区中单击目前正在编辑项目中的**程序**项目，之后鼠标右击并于快捷选项中单击 **导入程序单元**。



选择要导入的文件后按下「打开」，而若该文件有设置密码保护时，请在出现的窗口中输入正确的密码后按下「确定」键即可。导入的 POU 将不会被指派任何的工作 (TASK)，且项目中有相同名称的 POU 时，系统将不允许重复名称的 POU 被导入，而其它的 POU 则可正常导入。



5.4.7 导出功能块 POU

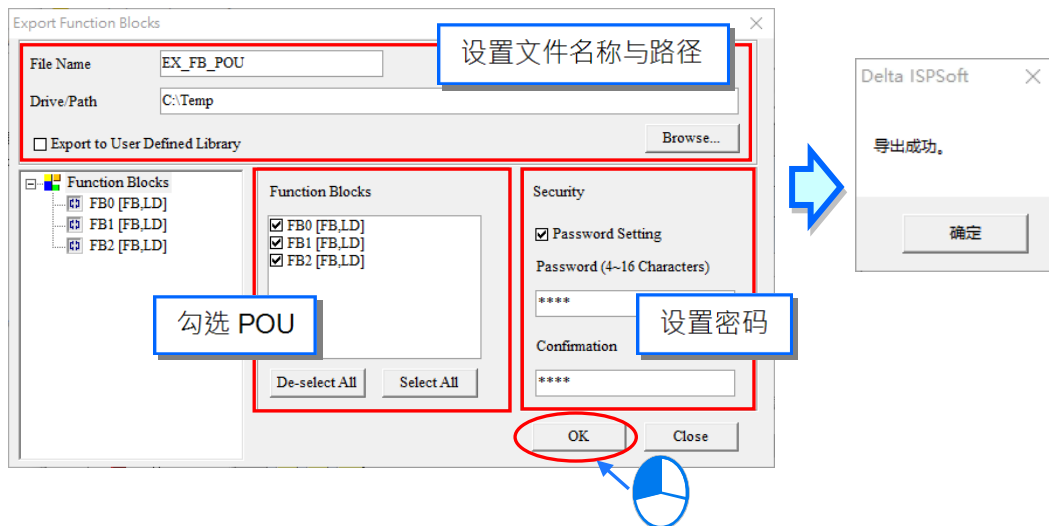
ISPSOft 支持将 POU 进行「导出」与「导入」的功能，不过在群组项目的架构下，每次则只能针对目前正在编辑的项目进行操作。

菜单工具栏单击 **文件 (E) > 导出 > 功能块**，之后便可直接对目前正在编辑的项目进行功能块的导出操作；另外也可在**项目管理区**中单击目前正在编辑项目中的功能块项目，之后鼠标右击并在快捷选项中单击 **导出功能块**。



5

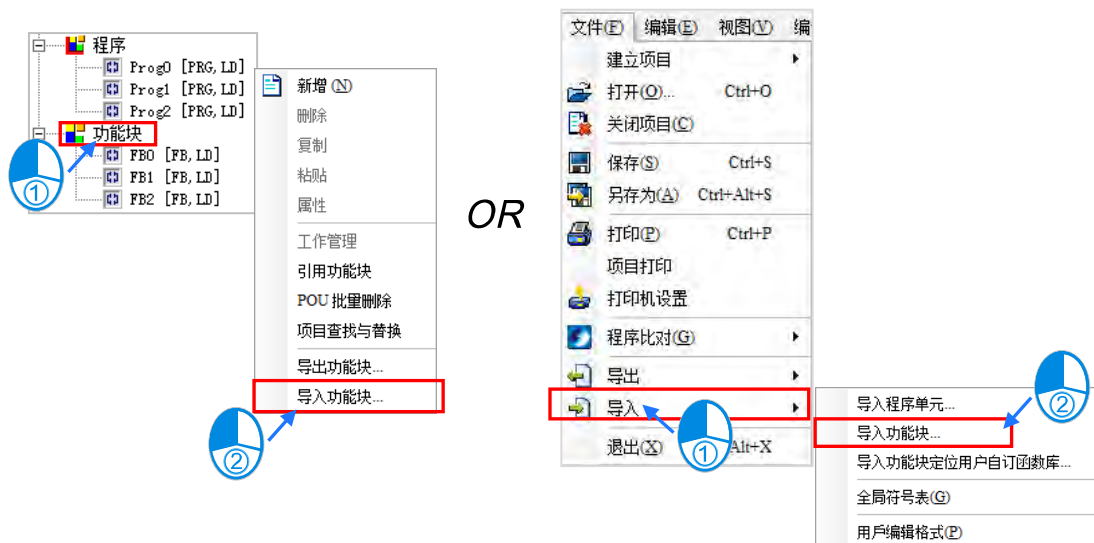
首先在设置窗口中设置导出的文件名称及路径，并于「功能块」清单中勾选欲导出的功能块 POU，而若要为导出的文件加上密码保护时，请先勾选「密码设定」后，在两个密码字段中输入密码，且两个密码必须一致以确认输入无误，而输入的字符可为英数字或空白，完成后按下「确定」键即可。



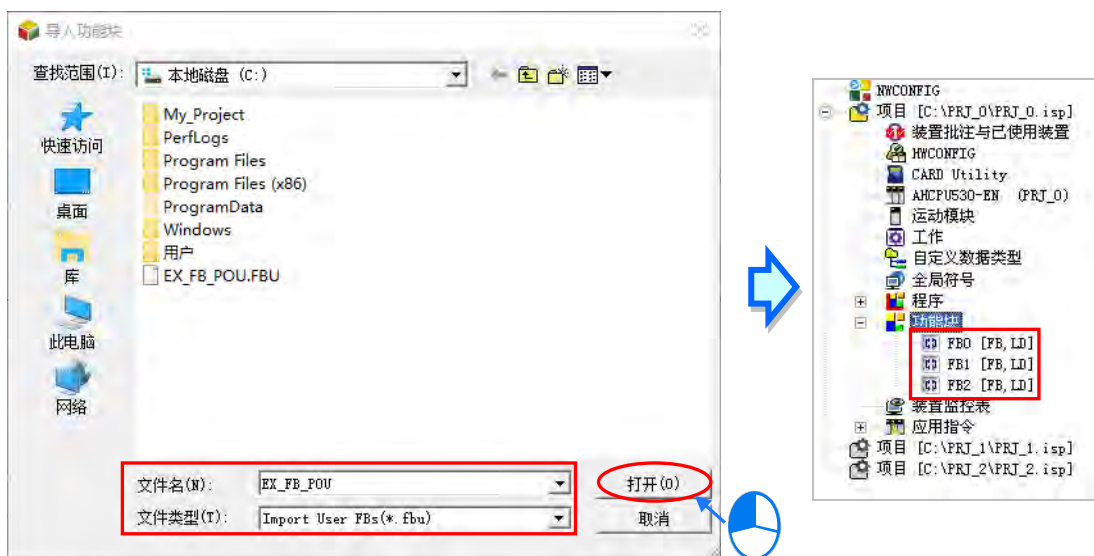
5.4.8 导入功能块 POU

与「导出」操作相同，进行「导入」时亦只能针对目前正在编辑的项目进行操作。此外，亦不可将导出的程序单元 POU，重新导入至「功能块」下。

在菜单工具栏单击 **文件 (F) > 导入 > 功能块**，之后便可直接对目前正在编辑的项目进行功能块的导入操作；另外也可于项目管理区中单击目前正在编辑项目中的功能块项目，之后鼠标右击并于快捷选项中单击 **导入功能块**。

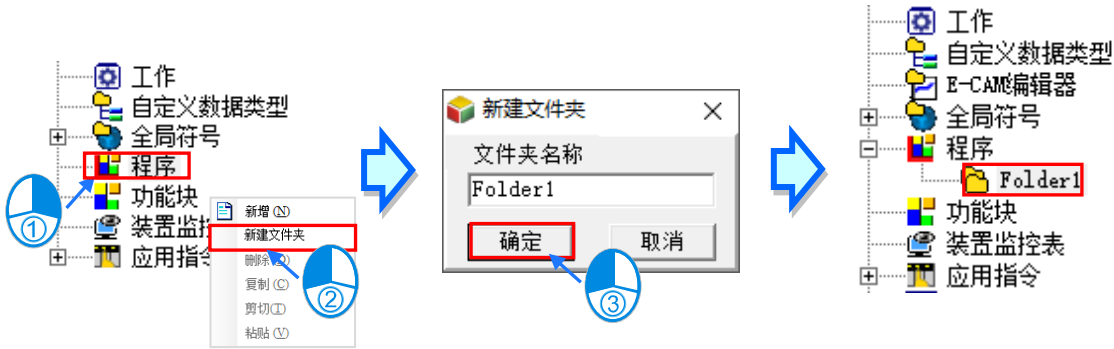


选择要导入的文件后按下「打开」，而若该文件有设置密码保护时，请在出现的窗口中输入正确的密码后按下「确定」键即可。当项目中有相同名称的 POU 时，系统将不允许重复名称的 POU 被导入，而其它的 POU 则可正常导入。



5.4.9 POU 的目录管理

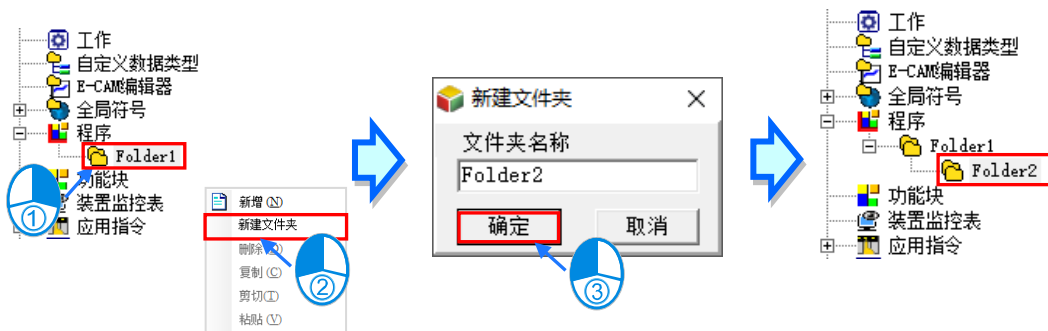
在项目管理区中的程序、功能块或函式项目，鼠标右击并点选**新增文件夹**，接着输入文件夹名称，完成后即可增加文件夹。



文件夹命名的规则如下。

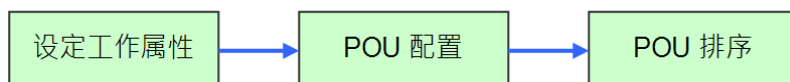
- (a) 不可使用特殊字符或空白，例如 *、#、?、\、%、@ ...等。
- (b) 至多只可输入 40 个字符，且须注意一个中文字将会占用两个字符。
- (c) 起始位不得为数字。
- (d) 名称当中可使用底线，但不可连续使用，或是置于结尾，例如 **"POU_1"** 是合法的名称，但 **"POU_1"** 与 **"POU_1_"** 则是不合法的名称。

文件夹可以用阶层的方式建立，如下图，亦可以将建立好的程序 POU、功能块 POU、文件夹拖拉移动到不同阶层，提供更方便的 POU 分类管理。



5.5 工作 (TASK) 管理的操作

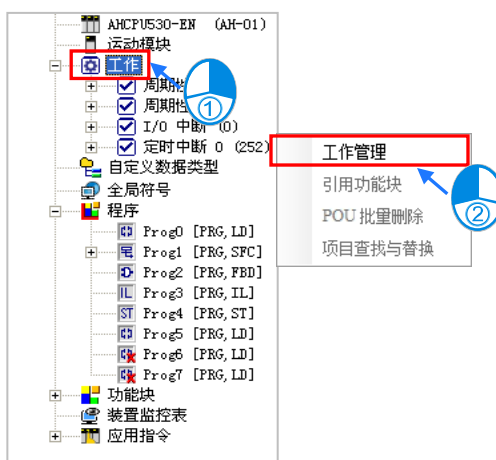
下图为在 ISPSOft 中进行工作管理的操作步骤，详细的操作方式将于后续的几个小节当中逐一介绍。



进行工作管理之前必须先打开 ISPSOft 中的**工作管理工具**，而其打开的方式有以下三种。

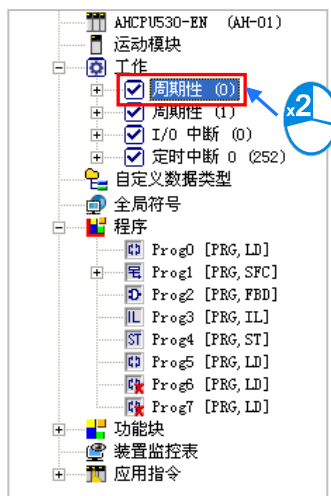
● 方法一

在项目管理区的「工作」上点击鼠标右键，接着在快捷选项中单击「工作管理」，而之后系统便会直接打开管理工具。



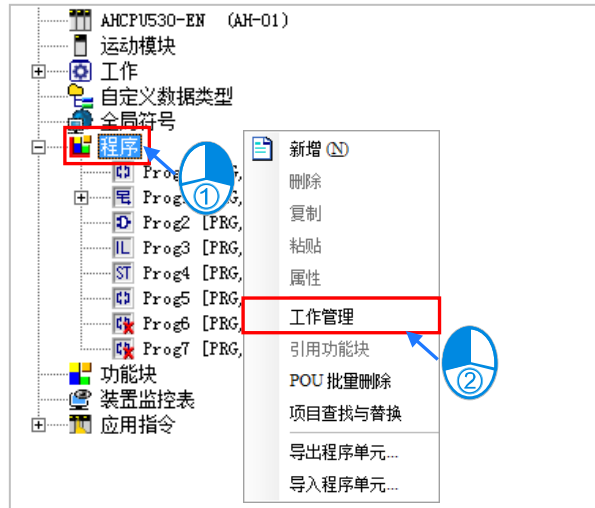
● 方法二

展开项目管理区的「工作」目录，并在要设置的工作 (TASK) 上双击鼠标左键，而之后系统便会打开管理工具，并直接显示该 TASK 的设置页面。



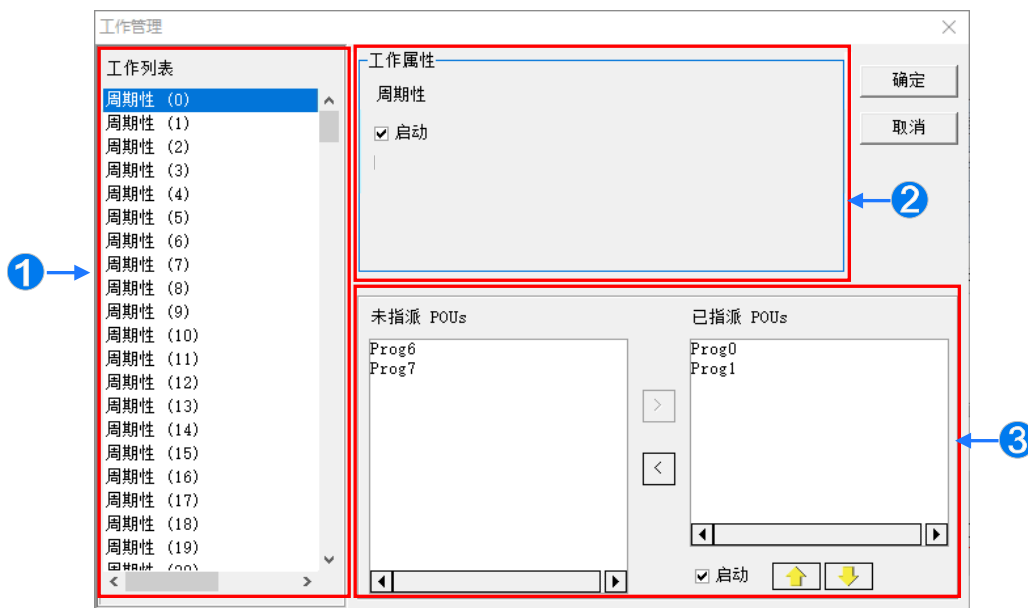
● 方法三

于项目管理区点选「程序」项目中，接着点击鼠标右键并于快捷菜单中点选「工作管理」，而之后系统便会自动开启管理工具。



下图便是工作 (TASK) 管理工具的操作界面。

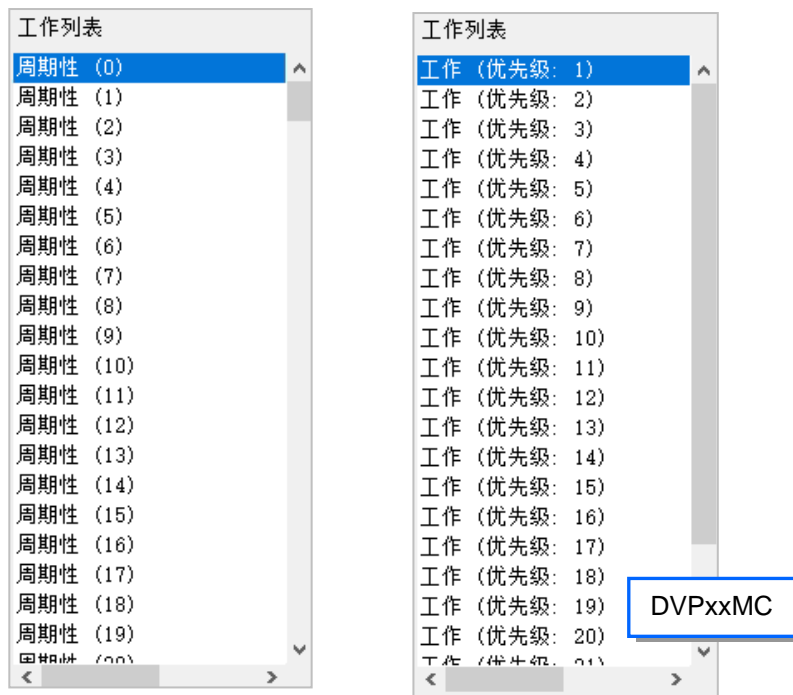
5



- ① 工作 (TASK) 列表：此处会列出所有可规划的工作 (TASK) 项目。
- ② 工作 (TASK) 属性：此处会显示所选 (TASK) 的属性设置及说明。
- ③ 工作 (TASK) 管理区：此处为对 POU 进行配置与排序的操作区。

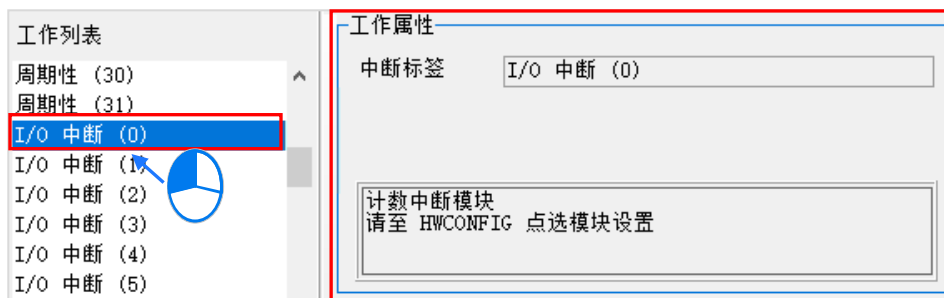
5.5.1 工作 (TASK) 属性与中断条件的设置

在「任务列表」中的 (TASK) 项目，会因为不同机种，所提供的 TASK 项目而有所不同，如下图左侧为一般机种；若为 DVPxxMC 或 AS5xx 机种，则显示为下图右侧。

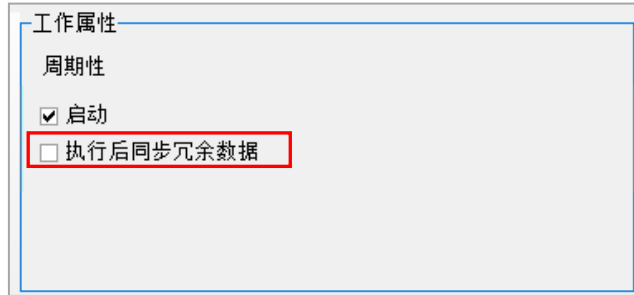


5

在左侧列表中选定工作 (TASK) 对象后，「工作属性」的区域便会显示对应该 TASK 的设置与提示，而此处的功能主要在于设置该 TASK 的执行条件；项目所提供的 TASK 与所选机种的功能特性有关，因此每个 TASK 的属性内容都会有所差异，不过基本的设置方式都是相同的。





而「工作属性」的区域所显示的项目，也会随着项目机种而有所不同，如下图，如果项目机种选取 AH560 冗余机种，才会有「执行后同步冗余数据」的选项，而这个功能的相关使用内容，可以参阅该产品相关手册。



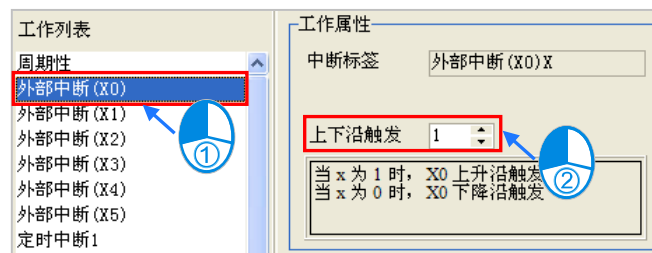
先以几个简单的例子来说明如何对工作属性进行设置，而其它工作属性的设置方式则请参考下述的范例，再配合对应机种的操作手册或相关说明文档来进行设置。

EX1：假设要设置 SV 机种项目的中断 TASK 属性，首先必须先确认 SV 机种所提供的中断指标，而下表即为 **DVP-PLC 应用技术手册（程序篇）** 中注明的 SV 机种中断指标。

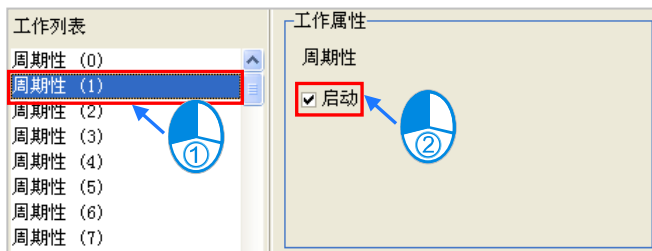
外部中断插入	I00□(X0), I10□(X1), I20□(X2), I30□(X3) I40□(X4), I50□(X5), 6 点 (□=1, 上升沿触发  □=0, 下降沿触发 )
定时中断插入	I6□□, I7□□, 2 点(□□=1~99, 时基=1ms) I8□□, 1 点(□□=1~99, 时基=0.1ms)
高速计数到达中断插入	I010、I020、I030、I040、I050、I060, 6 点
脉冲中断插入	I110、I120、I130、I140, 4 点
通讯中断插入	I150、I160、I170, 3 点

5

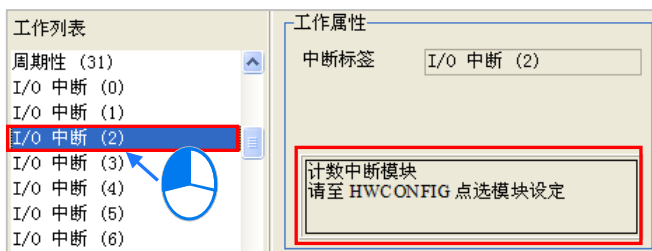
下图则为 SV 项目的 TASK 列表。若要设置与中断指标 I001 (即 X0 的上升沿触发中断) 对应的 TASK 时，请先于工作列表中选择「外部中断 (X0)」，之后于右侧的属性设置区域便会显示与其相关的设置与提示，而于提示的内容可得知，当上下沿触发的字段设置为 1 时便代表为上升沿触发，因此接着只要再将「上下沿触发」的字段设置为 1 即可。



EX2：下图为 AHCPU530-EN 机种的 TASK 属性设置画面，且单击的 TASK 页面为周期性 (1)；而因 AH 系列机种可透过 **TKON** 与 **TKOFF** 的 API 应用指令来启动或关闭周期性 TASK 的功能，因此于此处可设置该 TASK 的初始状态；当「启动」项目未被勾选时，该 TASK 于程序执行时便会先处于关闭的状态，而之后便必须等待其它 TASK 中的 POU 对其使用 **TKON** 指令后，该 TASK 才会开始执行。关于 **TKON** 与 **TKOFF** 指令的相关说明请参考相关程序手册。



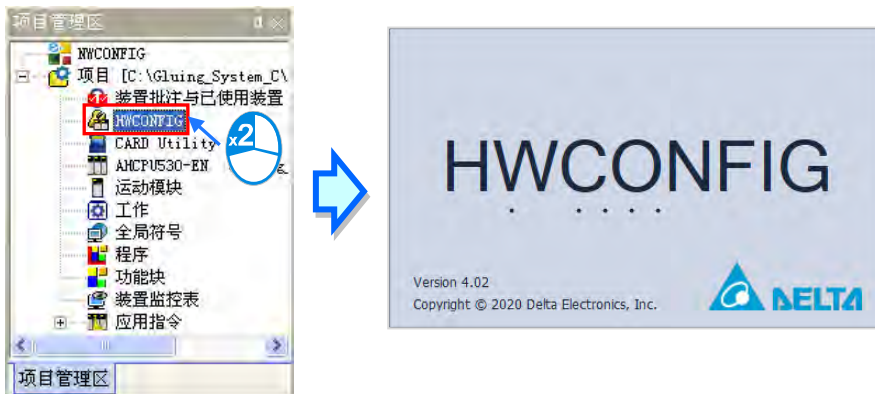
EX3：下图为 AHCPU530-EN 机种的 TASK 属性设置画面，且单击的 TASK 页面为 I/O 中断 (2)；
 而因 AH 机种的 I/O 中断条件可由用户自行规划，因此提示文字中便会注明至 HWCONFIG
 中进行设置。下列我们将继续示范如何于 HWCONFIG 中规划中断条件。



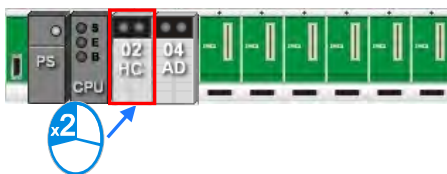
此处我们假设指定 I/O 中断 (2) 的中断条件是要来自计数器模块 AH02HC-5A 的通道 0，而下列
 则为透过 HWCONFIG 设置 IO 中断条件的范例，更详细的操作步骤请参考手册第 3 章的说明内容，
 而关于 AH 系列机种的中断功能，则请参考 AH 机种的操作手册或相关技术文件。

5

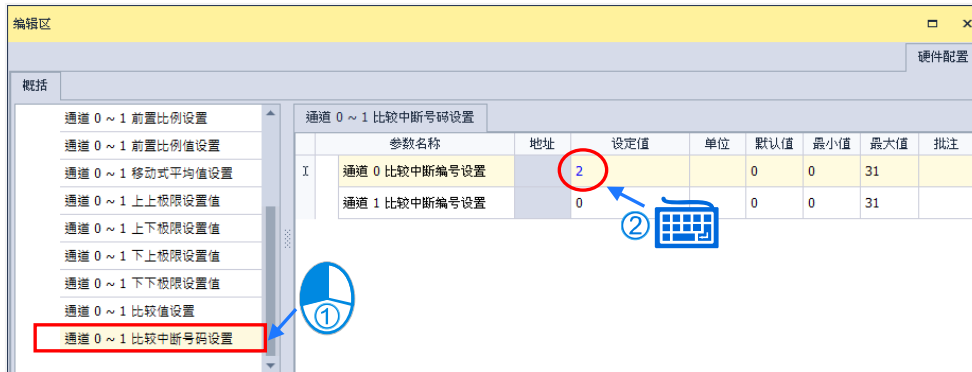
(1) 打开 ISPSOft 中的 HWCONFIG。



(2) 双击欲产生中断信号的计数器模块打开模块参数的设置窗口。



- (3) 打开模块设置窗口后，请先切换到中断编号的设置页面，并将「通道 0 比较中断编号设置」中的「初始值」字段设置为预计的中断编号 2。



*.通道 1 与通道 2 的中断编号可相同，而这样的设置代表当有任何一个通道的条件成立时都会触发同一个 I/O 中断。

- (4) 接着切换至比较值的设置页面，并于「通道 0 比较值设置」的「初始值」字段中设置触发中断条件的计数值。

5



- (5) 而完成设置后请务必记得进行保存，并在开始执行前将所设置的参数下载至主机中才可生效。

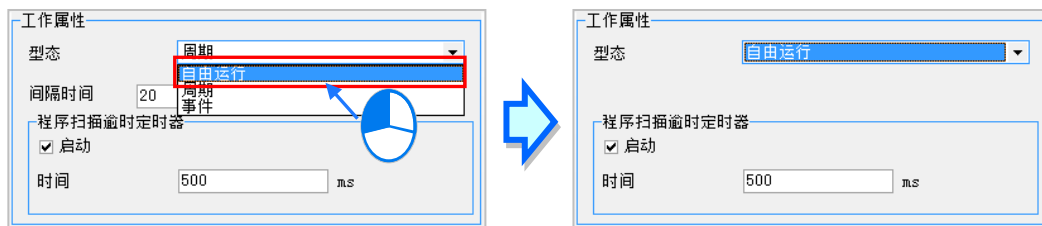


若项目使用 DVPxxMC 或 AS5xx 机种，那么任务栏表的选项会有，优先级 1 ~ 24 (1 代表最高优先级，24 代表最低优先级)。选取完优先级之后，可以在工作属性项目里，选择适当的「型态」及设定「程序扫描逾时定时器」。接下来就「型态」以及「程序扫描逾时定时器」来作介绍。

● 型态

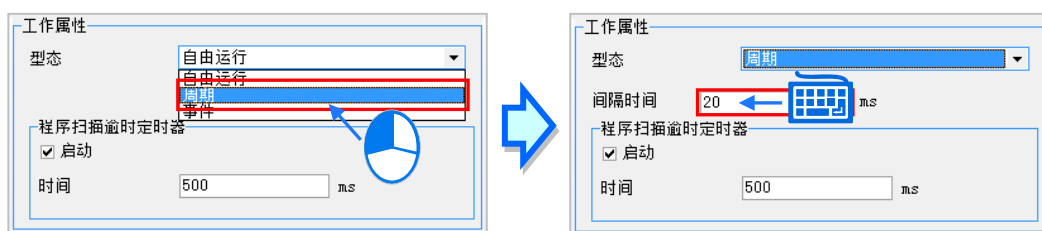
用户可选择的「型态」，共可分为三种型态，分别为「自由运行」、「周期」以及「事件」。

(1) 自由运行



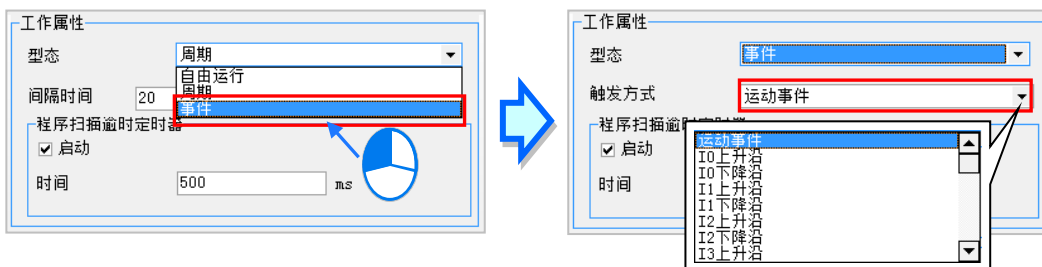
于「型态」字段下拉式选单中，选择「自由运行」，此项目为程序一开始运行工作（TASK）就被处理，一个运行周期结束后，工作（TASK）将在下一个周期中被重新启动。

(2) 周期



于「型态」字段下拉式选单中，选择「周期」，并可在「间隔时间」字段输入欲设定时间；周期工作（Task）运作将按照工作（TASK）管理工具内工作属性的「间隔时间」设定的时间周期执行。

(3) 事件



于「型态」字段下拉式选单中，选择「事件」，并可在「触发方式」字段下拉式选单选择欲触发的方式。

►可供选择的事件包含以下几种触发方式：

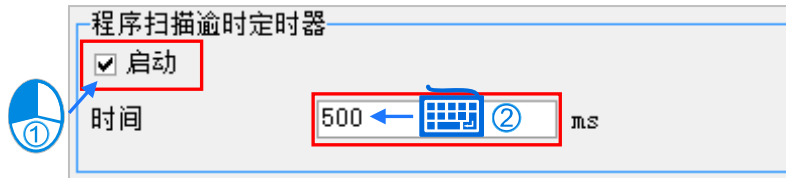
- 运动事件（运动控制工作（Task））；
- 本机输入点（I0~I7，I10~I17）的上升沿或下降沿；
- CANopen 网络的同步信号；
- 增量型编码器 1 或增量型编码器 2 的 Z 脉冲信号的上升沿；

事件工作（Task）是指仅在指定的事件发生时执行一次。事件工作（Task）的执行时机取决于事件发生的时机和事件工作（Task）的优先级。

● 程序扫描逾时定时器

属性项目里的「程序扫描逾时定时器」，用于监控工作 (TASK) 单次运行时间是否超出设定的超时时间，若超出，则会产生相应的报错信息。每个工作 (TASK) 均可设置其超时时间以及选择是否开启「程序扫描逾时定时器」功能。

用户若欲设定「程序扫描逾时定时器」可勾选「启动」项目后，再输入欲设定的时间。



5.5.2 POU 的配置

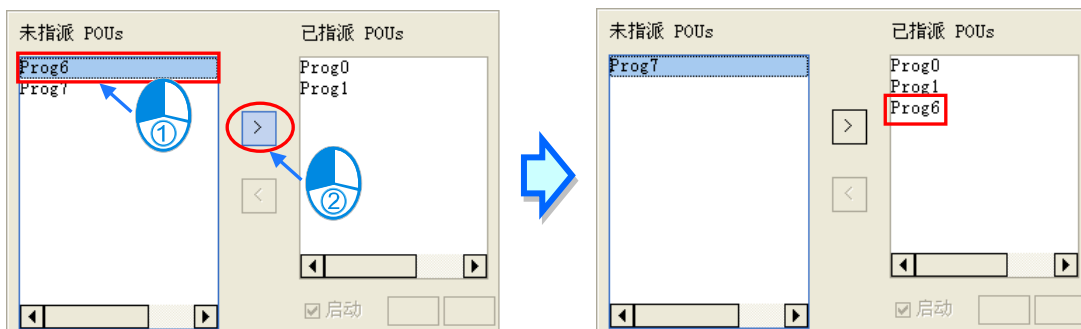
完成 TASK 的属性设置后，接下来便要将该 TASK 会执行的 POU 程序配置于该 TASK 的清单中，配置时请注意，一个程序 POU 只能指派给一个 TASK，而且 SFC 编辑的 POU 只能被配置于周期性的 TASK 中。


- (1) 先于左侧工作列表中单击要设置的 TASK 项目，之后管理区中便会显示该 TASK 的配置状况，其中左侧为尚未指派 TASK 的 POU，而右侧则为已分配给此 TASK 的 POU。

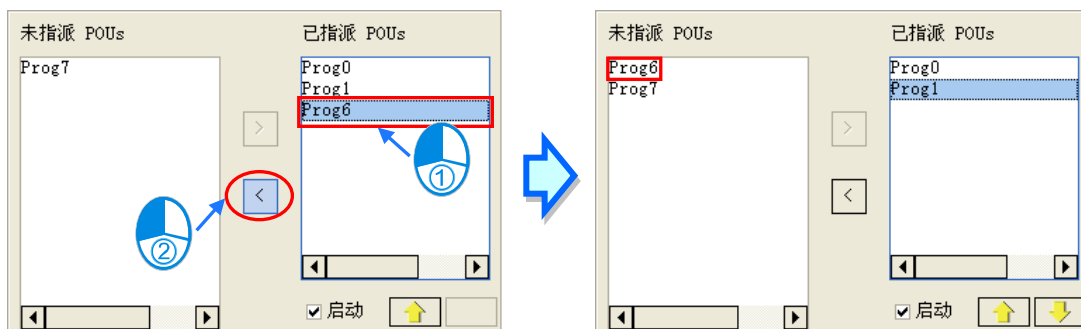
5



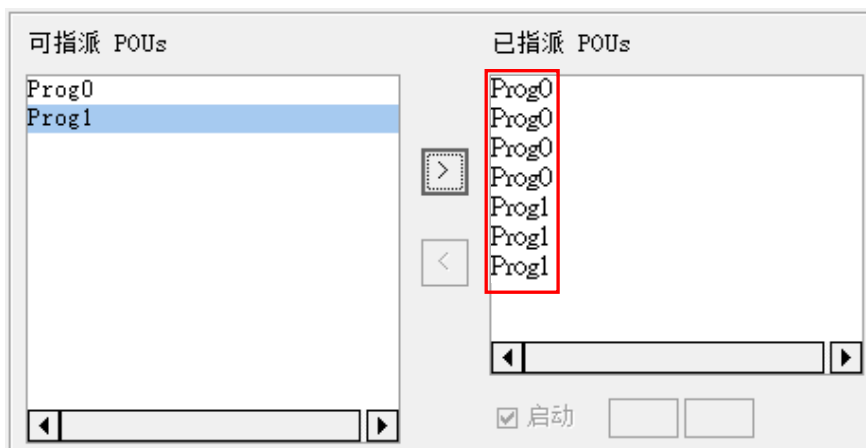
- (2) 在工作 (TASK) 管理区的左侧列表中，单击欲分配至此 TASK 的 POU 后，按下 > 图标即可将该 POU 分配给此 TASK。



- (3) 在工作 (TASK) 管理区的右侧列表中, 单击欲取消指派的 POU 后, 按下  图标即可将该 POU 由此 TASK 清单中删除。




若项目使用 DVPxxMC 或 AS5xx 机种, 则同一个 POU 可以重复指派给多个不同的 Task, 也可以在同一个 Task 内重复指派多次, 如下图。

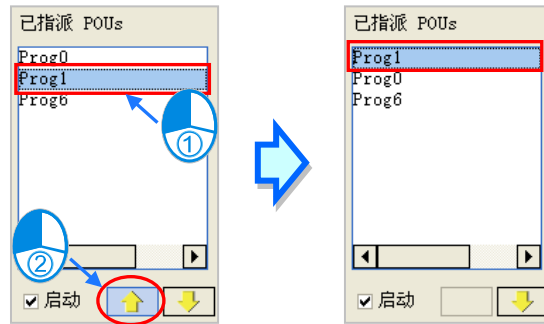



5

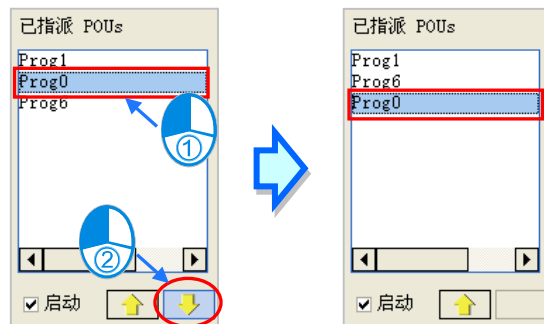
5.5.3 POU 的排序

完成 TASK 清单中的 POU 配置后，接着便要对这些 POU 进行排序，以决定实际执行时的顺序。

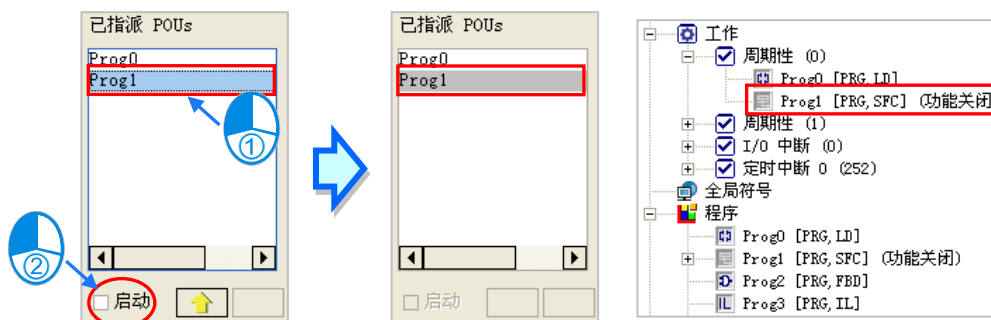
- (1) 在工作 (TASK) 管理区的「已指派 POUs」清单中，先单击欲变更执行顺序的 POU 之后，按下  图标即可将该 POU 的执行顺序往上移动一位，代表其执行顺序提前。



- (2) 在工作 (TASK) 管理区的「已指派 POUs」清单中，先单击欲变更执行顺序的 POU 之后，按下  图标即可将该 POU 的执行顺序往下移动一位，代表其执行顺序延后。



- (3) 在工作 (TASK) 管理区的「已指派 POUs」清单中单击某一 POU 后，勾选或取消「启动」选项也可改变该 POU 的启动状态，而设为未启动状态的 POU 于列表中亦会以灰阶显示。



*.关于 POU 的启动状态请参考第 5.4.2 节的相关说明。

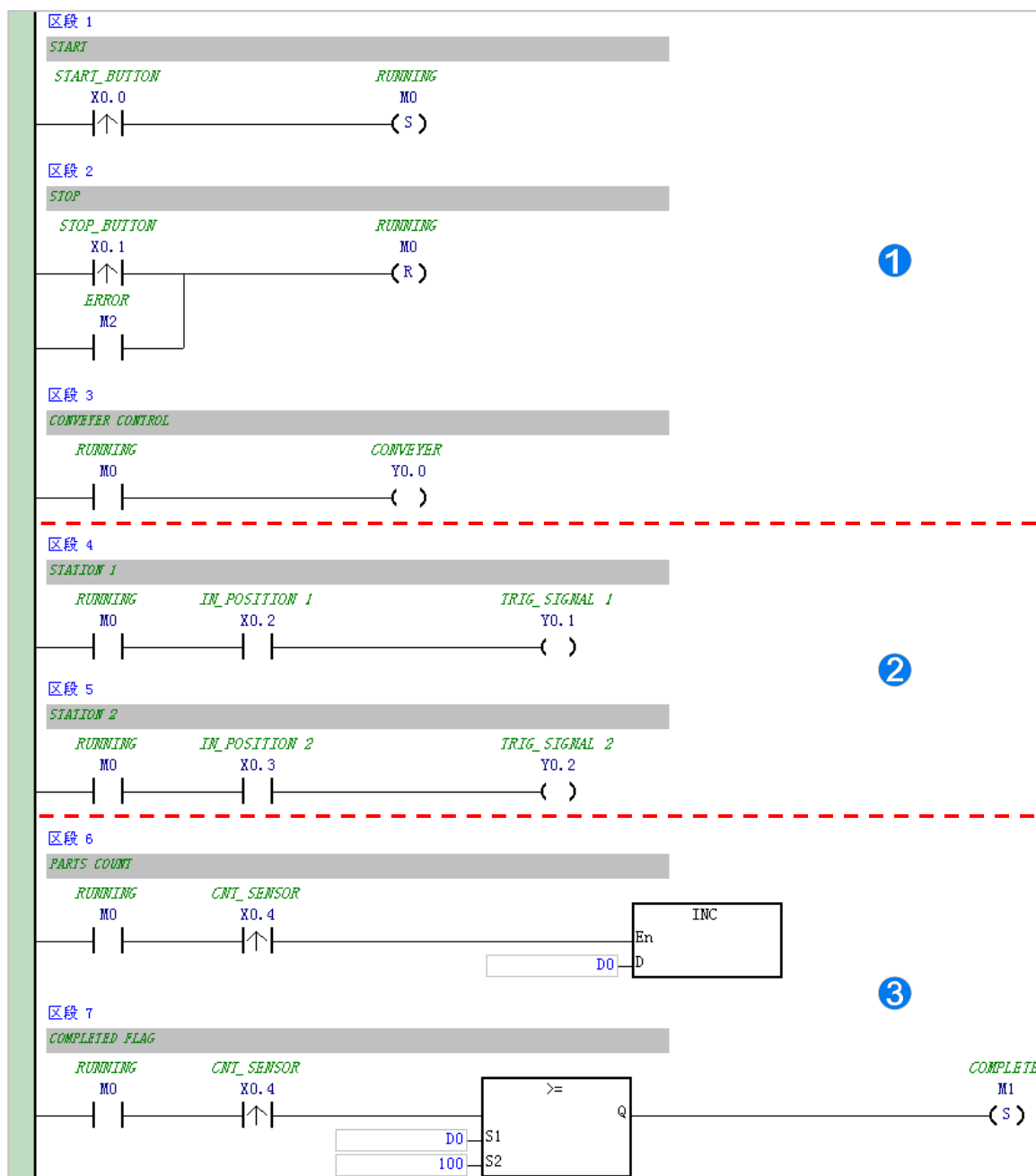
- (4) 完成所有工作 (TASK) 的设置后，按下工作管理工具的「确定」键即可。

5

5.6 范例实作

5.6.1 POU 与 TASK 的编程范例

延续第 4 章内容中的范例，接着我们将在此节中以 POU 与 TASK 的编程架构重新进行规划。下图即为我们在第 4 章中所建立的程序，且在此我们将依据程序的功能与作用，而将其重新规划为三个部份。

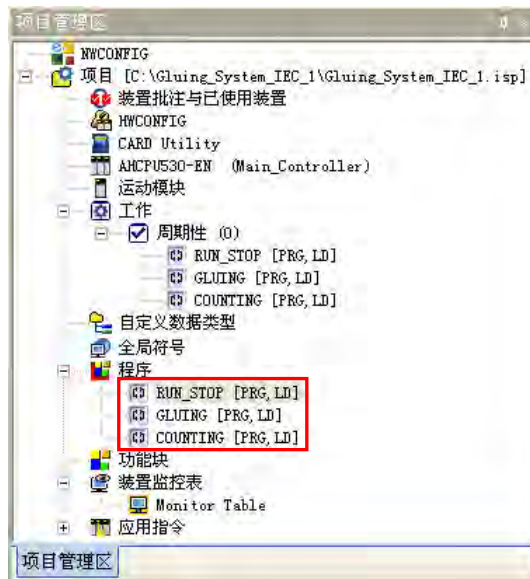


- ❶ 此部分的主要功能在于控制系统的启动或停止。
- ❷ 此部分的程序主要在于点胶的操作。

③ 此部分的作用在于计数已送入的工件数量。

此节的内容将着重于程序架构的部份，而关于**硬件组态**、**参数设置**与**除错测试**的操作则与第 4 章的说明内容相同，请试着自行完成相关的部分。

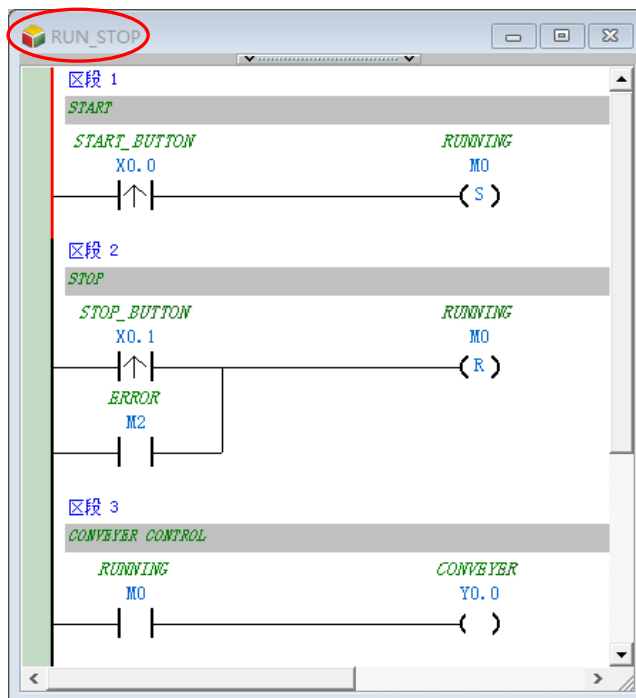
首先请新增一项目，并参考第 4 章的内容完成硬件组态的工作，接着依据上述的规划在「程序」下分别建立如下的 3 个 POU，建立的方式请参考第 5.4.1 节的相关说明；而建立 POU 时，属性窗口中的「工作」字段请先保持「周期性 (O)」的默认值即可。



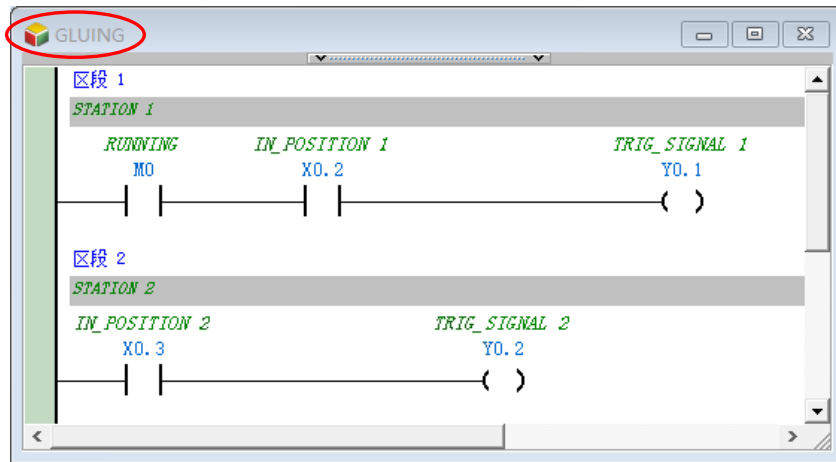
5

接着请参考第 4 章所介绍的梯形图编辑方式，在各个 POU 窗口中完成以下的程序。

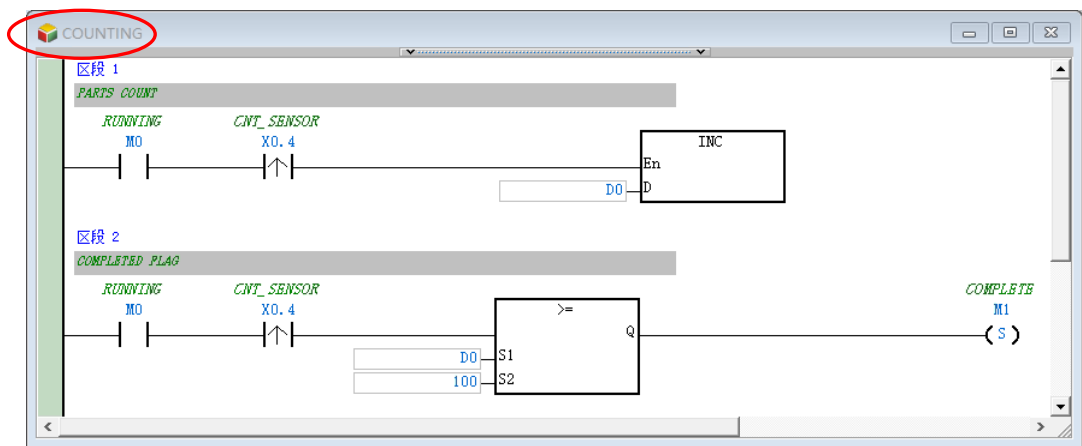
首先是控制系统启动/停止的 POU – 「**RUN_STOP**」。



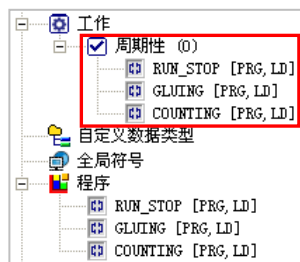
再来是控制点胶动作的 POU – 「GLUING」。



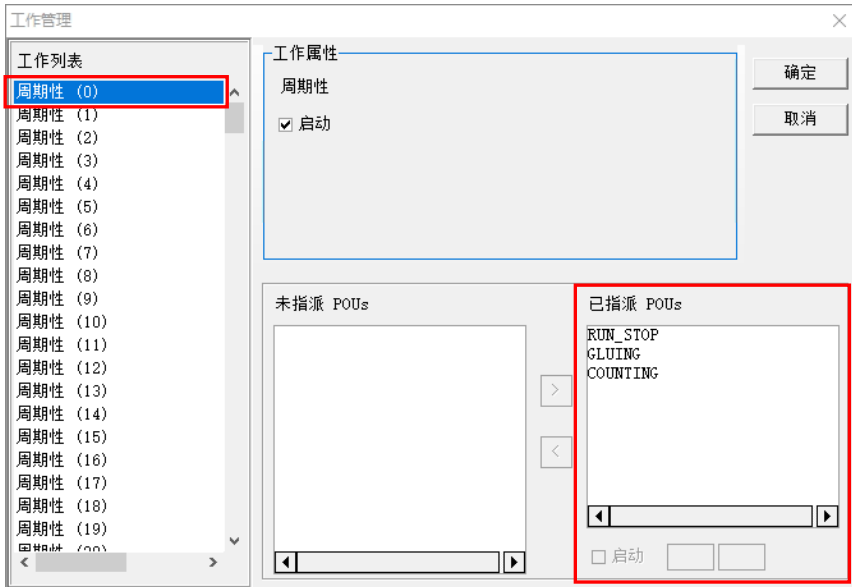
最后完成负责计数功能的 POU – 「COUNTING」。



完成程序建立后，请确认「工作」下的 POU 顺序与下图一致，若否，则请重新进行工作规划。

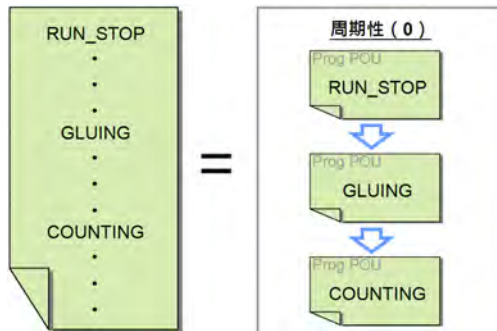


若要重新规划各 POU 的工作配置时，请于项目管理区的「工作」上点击鼠标右键，之后在快捷选项中单击「工作管理」便可打开工作管理工具，将 POU 加入到周期性(0)的已指派 POU 内。



5

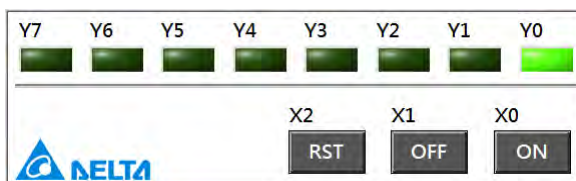
若已依据上述的顺序完成 POU 的规划，则之后此程序的执行结果便会与第 4 章的传统梯形图程序相同。接着请自行参考第 4 章的内容将程序完成编译，并下载至机器中进行后续测试。



5.6.2 DVP 机种的中断程序范例

• 范例说明

当「ON」的按钮 (X0) 被按下时, Y0~Y7 便会由 Y0 开始轮流扫描; 当「OFF」的按钮 (X1) 被按下时, 输出的动作便会停止, 而灯号则会停留在目前的状态; 当「RST」的按钮 (X2) 被按下时, 输出便会被重置, 且让扫描灯号重新由 Y0 开始。



• 程序规划

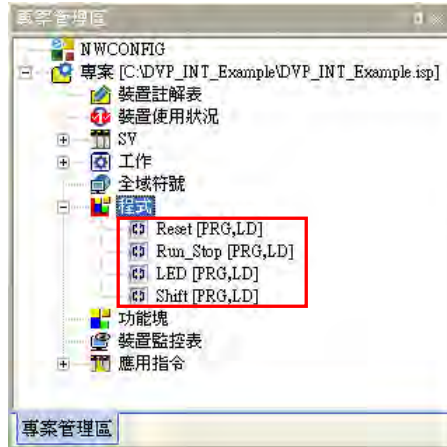
此范例所选用的机种为 SV, 且在此我们将程序规划为 4 个程序 POU, 而各 POU 的功能请见下表。

POU 名称	功能说明
Run_Stop	1. 初始化运转标志 (MO) 状态与输出数据寄存器 (DO) 的值, 并使能中断。 2. 当 X0 由 OFF 变为 ON 时, 将运转标志 (MO) 设置为 ON。
LED	依据输出数据寄存器 (DO) 的值更新 Y0~Y7 的输出状态。
Shift	每隔 0.5 秒, 将输出数据寄存器 (DO) 的内容偏移一个位, 并判断当数值超过 128 (即偏移超过 8 个位) 时, 将输出数据寄存器 (DO) 的值重设为 1 (代表重新由 Bit 0 开始进行偏移)。
Reset	将输出数据寄存器 (DO) 的值重设为 1, 亦即将输出的位重置为 Bit 0。 且此 POU 预计会由外部中断 I201 (X2 上升沿) 来触发执行。

*. 关于 SV 机种的中断指标及相关的 API 应用指令请参考 DVP-PLC 应用技术手册 (程序篇)。

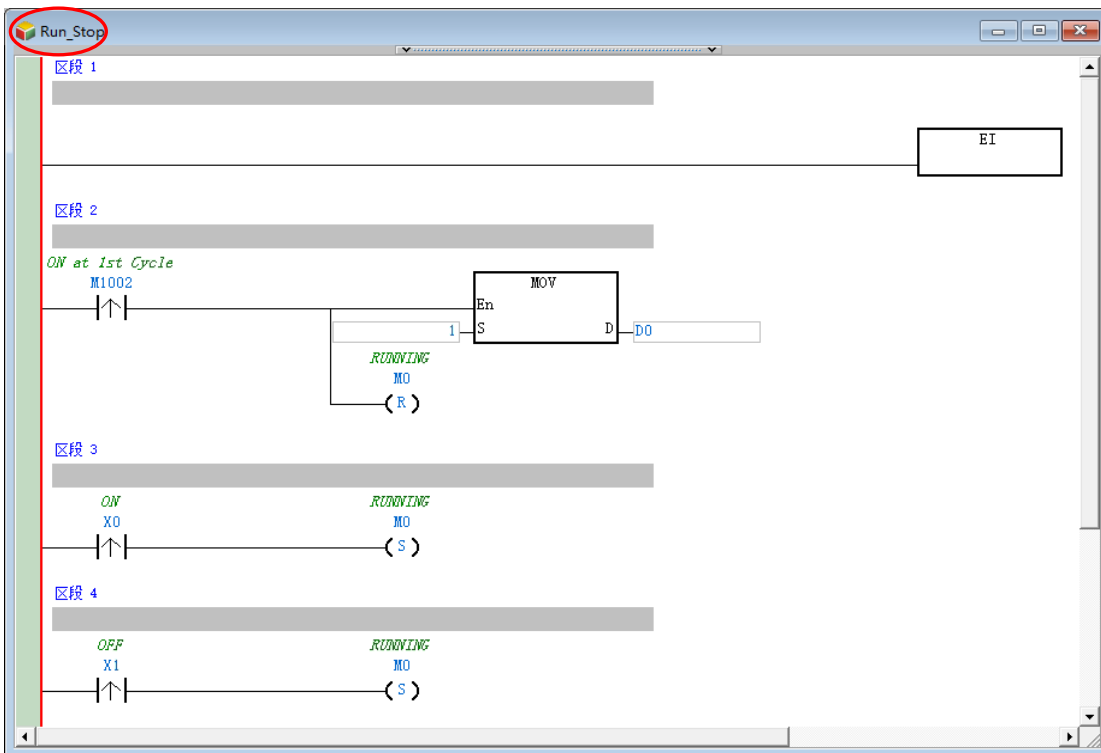
• 建立程序

首先请建立一个 SV 机种的项目, 接着请于项目管理区的「程序」项目下分别建立如下的 4 个 POU, 且 POU 属性的「工作」字段可先保持「周期性」的默认值即可。

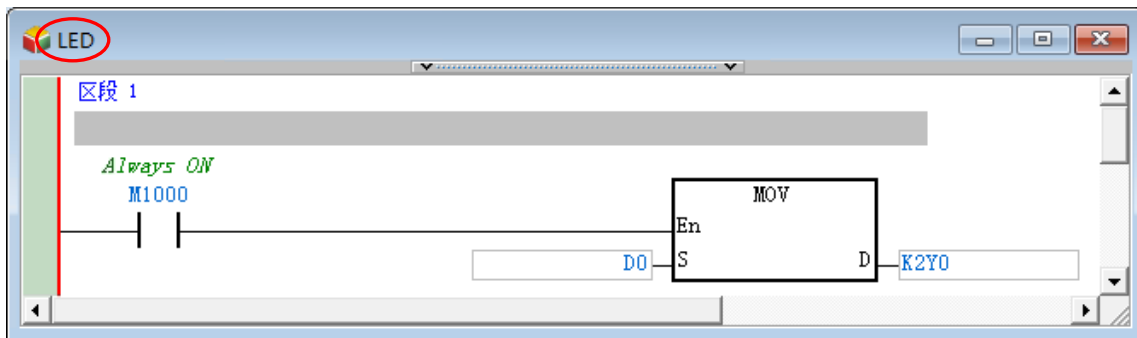


接着請在各 POU 的窗口中完成下列的程序。

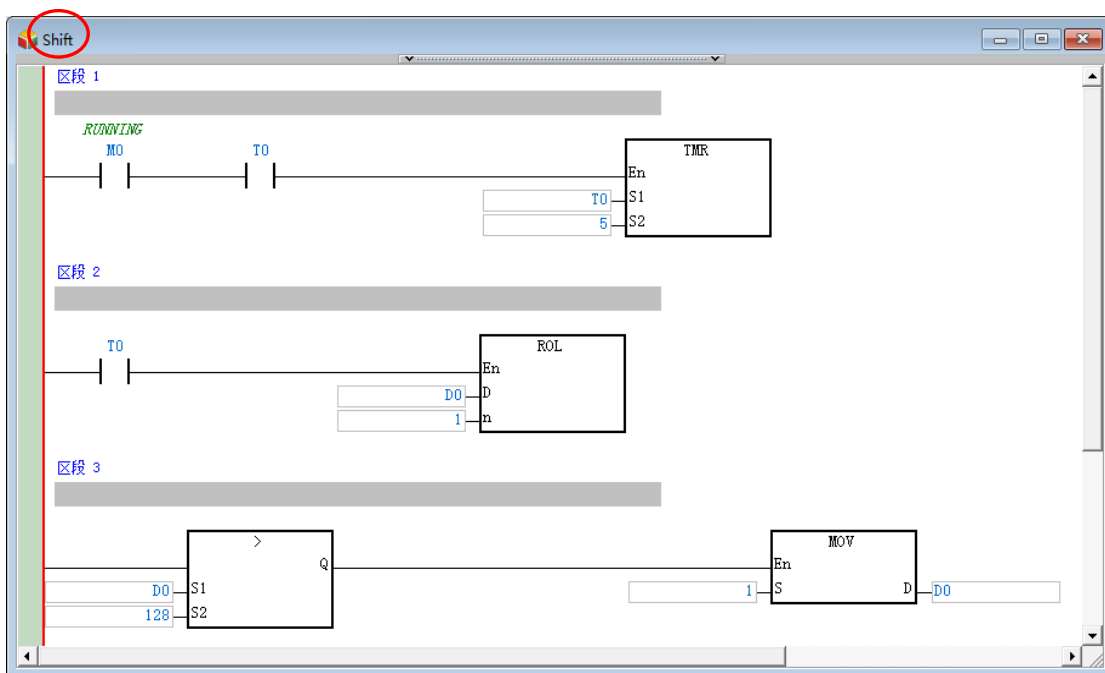
5



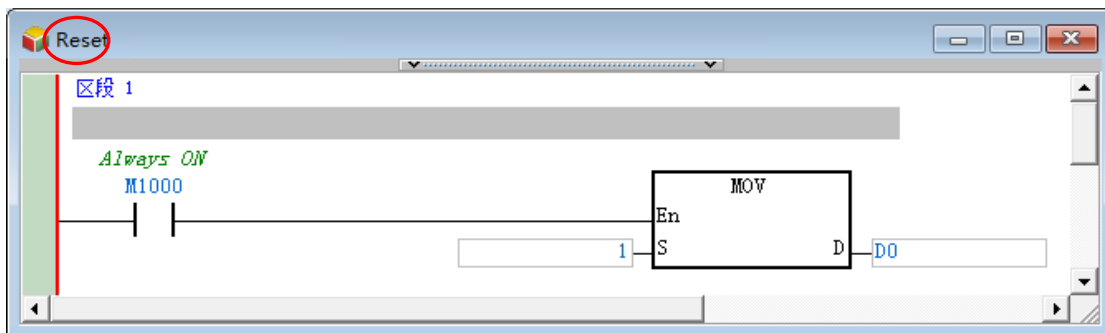
*.MOV 指令屬於 API 應用指令的「傳送比較」類別，而 EI 指令則屬於「回路控制」類別。



*.MOV 指令属于 API 应用指令的「传送比较」类别。



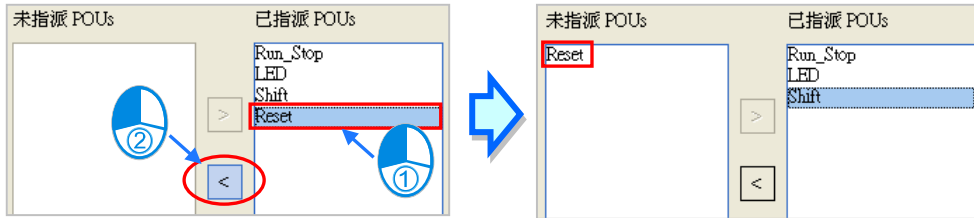
*.TMR 指令属于 API 应用指令的「基本指令」类别·ROL 指令属于「旋转位移」类别·MOV 指令属于「传送比较」分类·而比较触点则可由项目管理区的「应用指令」下的「触点类型逻辑运算」分类中找到。



*.MOV 指令属于 API 应用指令的「传送比较」类别。

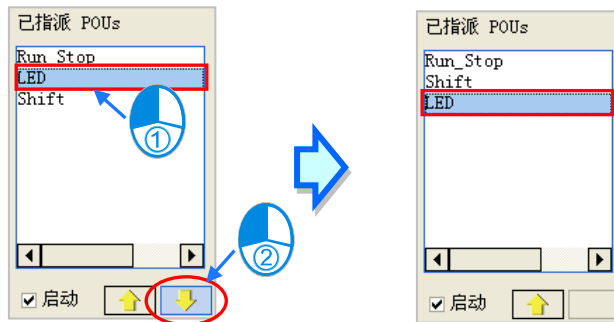
5

完成程序编辑后，请打开工作管理工具，而因建立 POU 时，其「工作」属性都沿用「周期性」的默认值，因此开启工作管理工具后，我们会发现所有的 POU 都配置在「周期性」的 TASK 列表中。但「Reset」的功能将由外部中断来触发，所以请将「Reset」由「周期性」的 TASK 列表中移除。

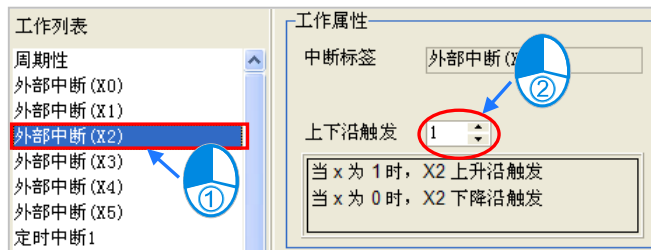


确定「周期性」的 TASK 清单后，接着便要对清单中的 POU 进行排序，而因为我们希望将初始化与输入的功能放在扫描周期的最前面，而输出的功能则放在扫描周期的最后，因此便必须将「周期性」的 TASK 清单依照「Run_Stop」→「Shift」→「LED」的顺序重新排序。

5



接着请在工作列表中选择「外部中断 (X2)」，并将属性设置区域的「上下沿触发」字段设置为 1，代表此 TASK 的中断条件为 X2 的上升沿触发。



完成 TASK 的属性设置后，接着请将「Reset」POU 配置至「外部中断 (X2)」的 TASK 清单中，且因清单中只有一个 POU，因此无须对清单中的 POU 进行排序。



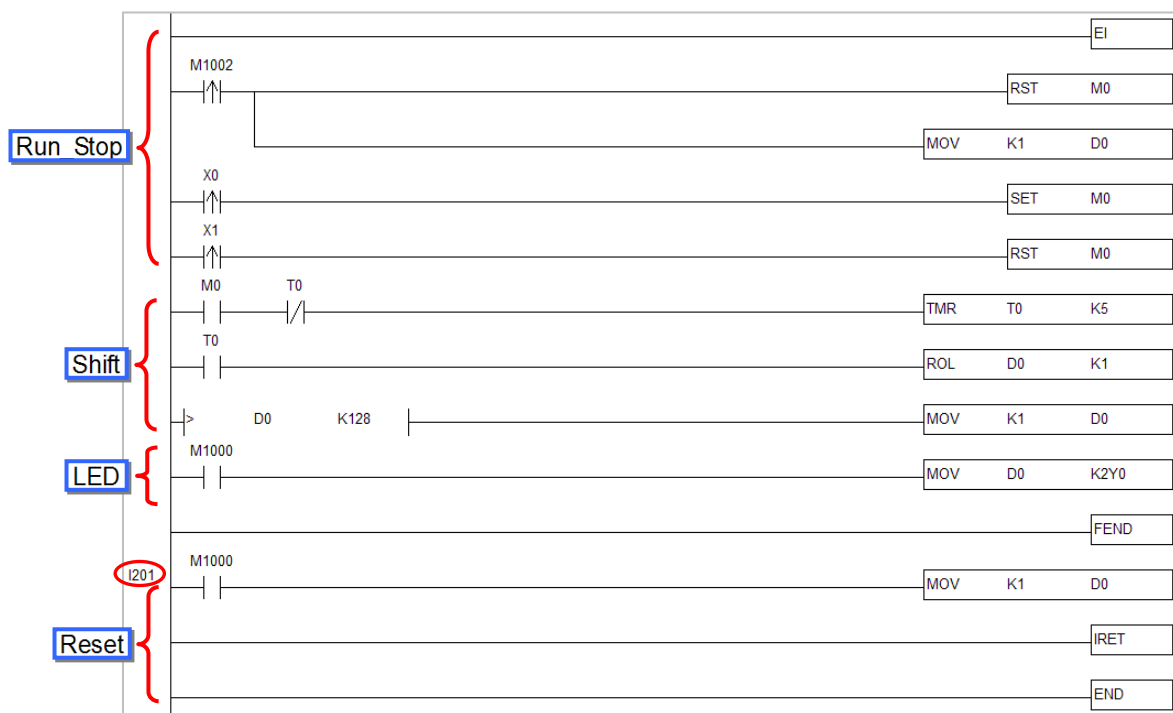
完成所有操作后请按下「确定」键应用于工作管理工具中所做的配置与设置，而其配置的结果如下。此范例程序到目前为止已编辑完毕，接着请自行编译并试着下载至主机中进行后续的测试。



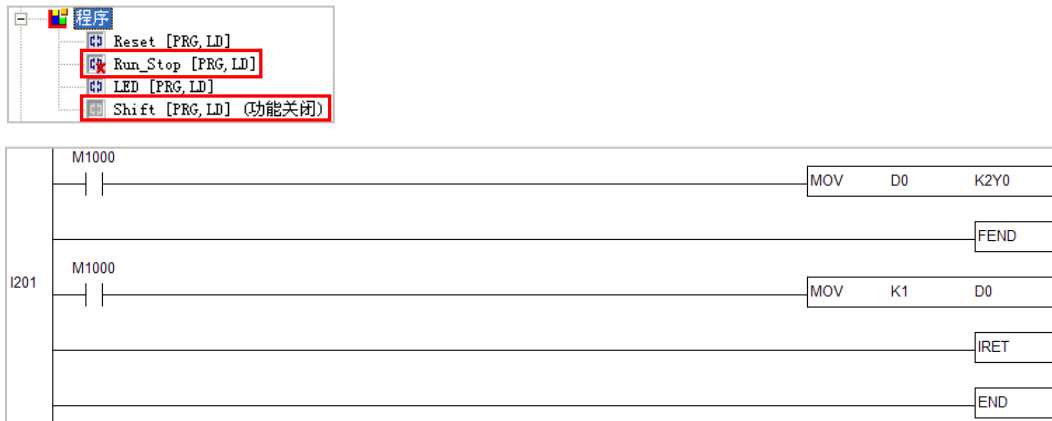
补充说明

若上述的范例程序已成功编译并下载至主机时，接着我们便可以试着透过 WPLSoft 软件将 PLC 内部的程序代码再上传回来进行比对，下列即是上传回来的程序代码。

由下图的程序可发现，主程序的部分即是由 3 个「周期性」TASK 的 POU 所组成，且组成的顺序为「Run_Stop」→「Shift」→「LED」，而中断子程序 I201 的内容则与「Reset」POU 相同。



接下来我们可以试着将「Run_Stop」由 TASK 清单中删除，并将「Shift」的 POU 状态设为关闭，接着完成编译并下载至主机，而之后再透过 WPLSoft 软件将程序代码上传，此时我们便可发现主程序的部份已不再包含这两个 POU 的程序。

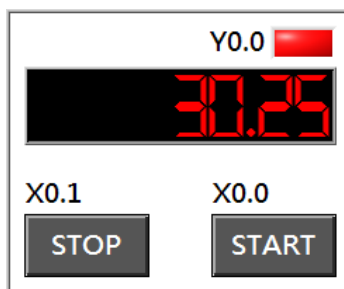


*.WPLSoft 亦为台达的 PLC 编程软件，而其程序架构则属于传统的 PLC 编程架构，相关操作请参考 WPLSoft 的使用手册。

5.6.3 AH5x0 机种的中断程序范例

• 范例说明

按下「START」键 (X0.0) 时系统便会开始计时，之后必须在 30 秒内按下「STOP」键 (X0.1)，否则「警报器」(Y0.0) 便会启动。

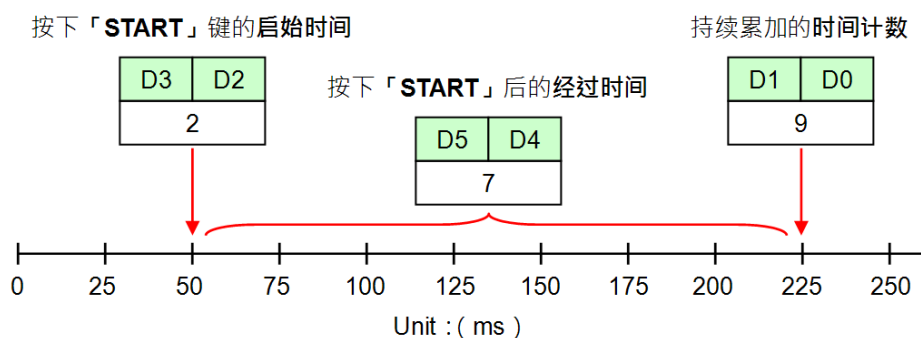


• 硬件规划

本节的范例将选用 AHCPU530-EN 主机，数字 IO 模块则为 8 点输入/8 点输出的 AH16AP11R-5A，而主背板则选用 4 槽的 AHBPO4M1-5A。(此规划与第 4 章的范例相同)

• 程序规划

此范例中，我们将利用定时中断的特性来产生一个固定 25ms 时基的计数，并利用程序的规划来达到计时的功能；如下图，定时中断 0 的 TASK 会每隔 25ms 就触发一次，而每次触发中断时便会将 32Bits 的时间计数寄存器 (D0,D1) 的值加 1，且只要 PLC 维持 RUN 的状态，此动作便会持续进行，藉此便可产生一个固定时基的计数；之后只要读取时间计数寄存器 (D0,D1) 的值便可知 PLC 已经运转的时间，例如时间计数寄存器 (D0,D1) 的值为 10，则 PLC 运转的时间便是 250ms (10x25ms)。当「START」键按下时，程序会将时间计数寄存器 (D0,D1) 的当前值传给起始时间寄存器 (D2,D3)，之后时间计数寄存器 (D0,D1) 会继续累加，而起始时间寄存器 (D2,D3) 则会保持在「START」键被按下的瞬间；因此后续只要将时间计数寄存器 (D0,D1) 的值减去起始时间寄存器 (D2,D3) 的值，便可知「START」键被按下之后的经过时间，我们会将这个值放到经过时间寄存器 (D4,D5) 中。



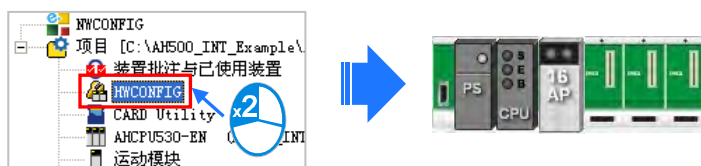
此节的范例除了会示范 AH5x0 机种的定时中断之外，还将示范 **TKON** 与 **TKOFF** 这两个 TASK 的控制指令。在此我们总共将程序规划为 5 个程序 POU，其功能简述如下。

POU 名称	功能说明
INT_Timer	每 25ms 便会将 32Bits 的时间计数寄存器 (D0,D1) 的值加 1。 (此 POU 将规划于定时中断 0 的 TASK 中)
Initialize	初始化标志 (M0,M1) 及所有相关的寄存器 (D0~D7)，并使能中断。 此 POU 会在第一个扫描周期执行时，将其它周期性 POU 所属的周期性 (0) TASK 功能启动，接着再将自己本身所属的周期性 (1) TASK 功能关闭，所以此 POU 仅会在第一个扫描周期时被执行。
Control	当 X0.0 由 OFF 变为 ON 时，便启动运转标志 (M0) 并将目前时间计数寄存器 (D0,D1) 的值传送给起始时间寄存器 (D2,D3)；当 X0.1 由 OFF 变为 ON 或计时产生溢位时，便关闭运转标志 (M0)。
Time_CHK	将时间计数寄存器 (D0,D1) 的值减去起始时间寄存器 (D2,D3) 的值以求出「START」键被按下后的经过时间，并将其存放于经过时间寄存器 (D4,D5) 中；且当计算出来的经过时间小于 0 时，便将溢位标志 (M1)
Signal	当经过时间超出 30 秒 (25ms x 1200) 或溢位标志 (M1) 为 ON 时，将警报器 (Y0.0) 设置为 ON

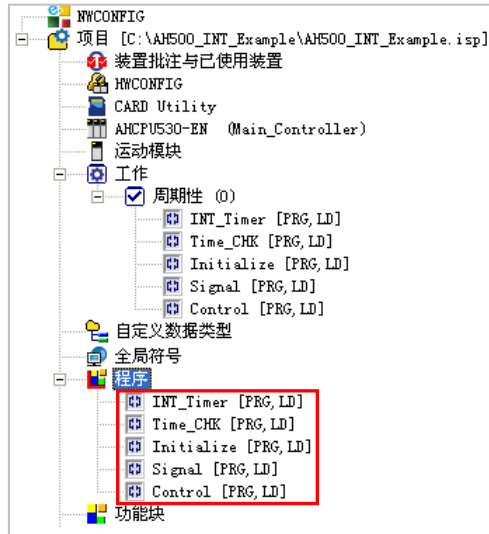
*. 关于 AH5x0 机种的中断指标及相关的 API 应用指令请参考 AH500 操作手册及 AH500 程序手册。

• 建立程序

首先请建立一 **AHCPU530-EN** 机种的项目，接着进到 HWCONFIG 增加一台 16AP 模块并下载硬件组态及参数设定到主机内。

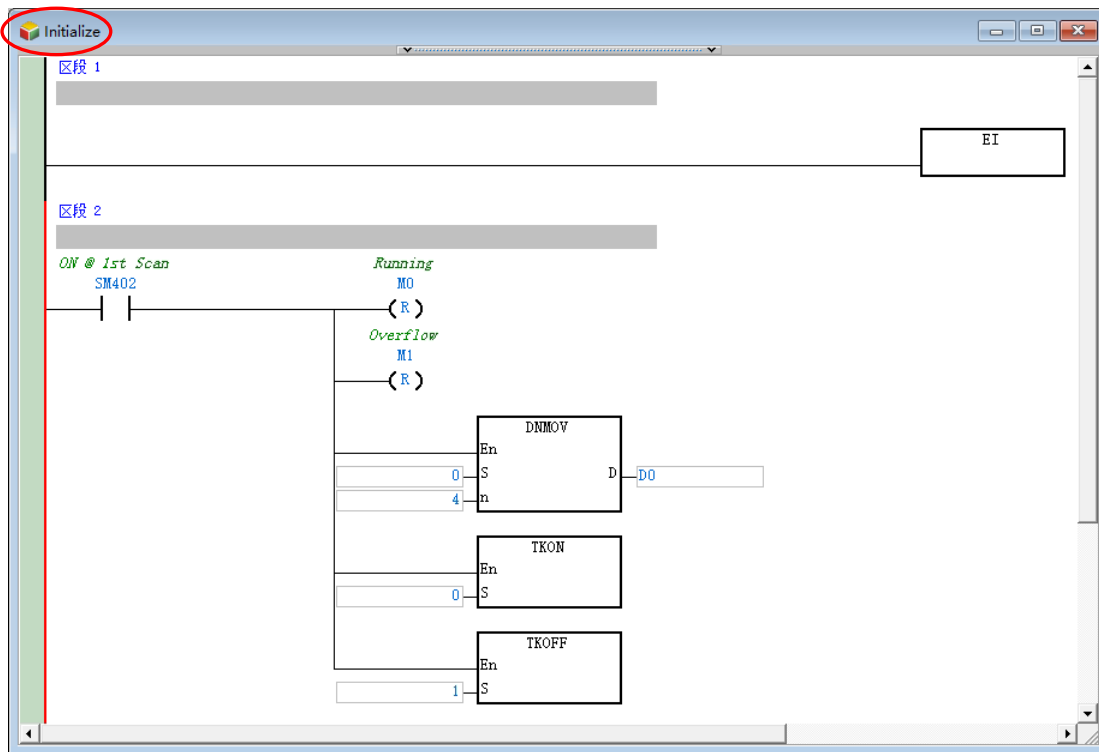


请在项目管理区的「程序」下分别建立如下的 5 个 POU，且 POU 属性的「工作」字段可先保持「周期性 (O)」的默认值即可。

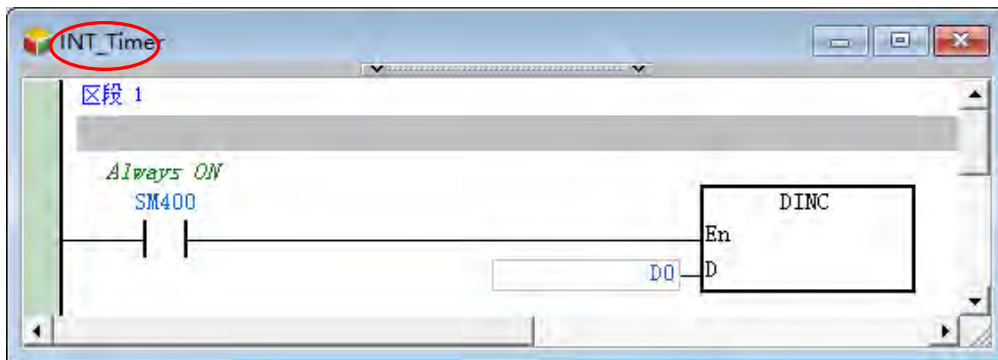


接着请在各 POU 的窗口中完成下列的程序。

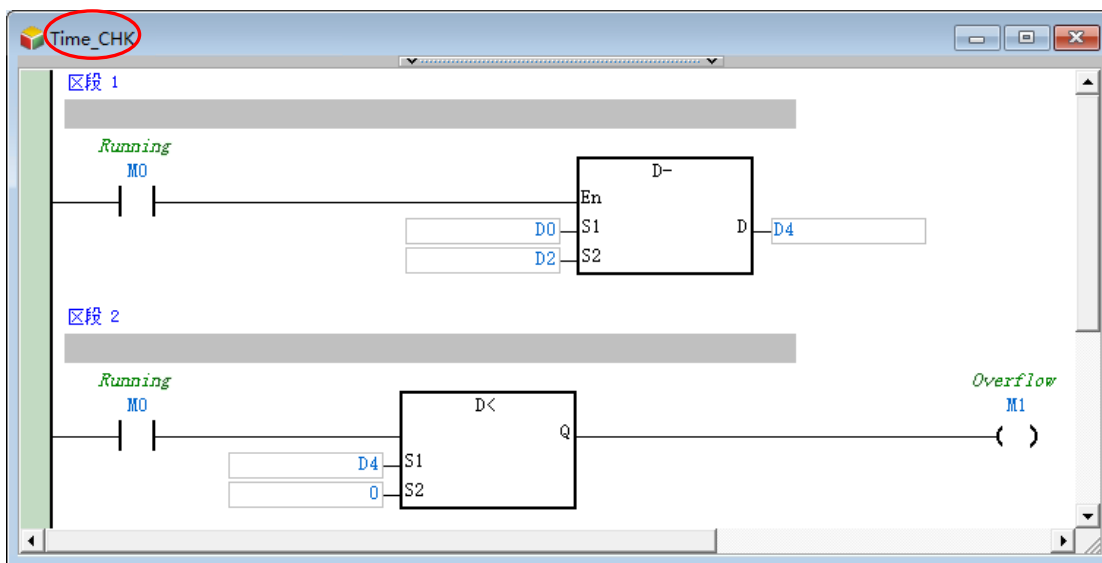
5



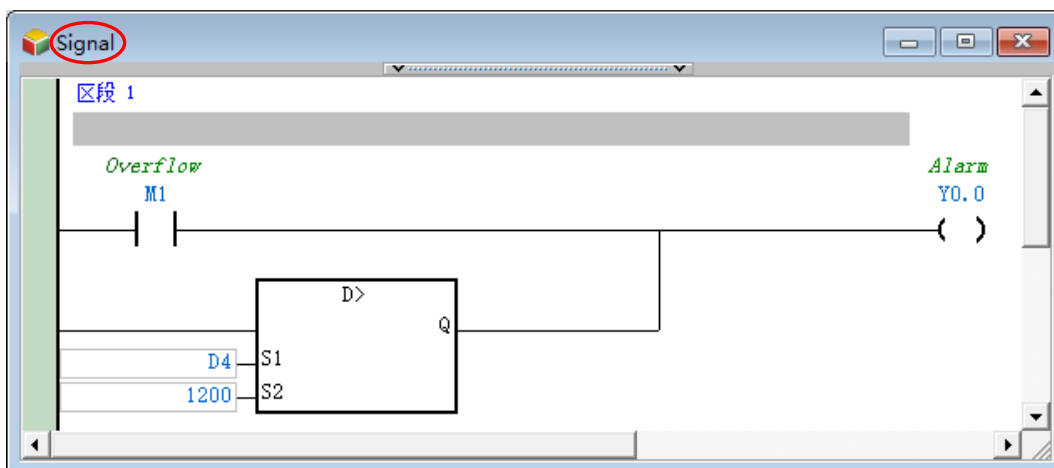
*.EI 指令属于 API 应用指令下的「回路控制」类别，DNMOV 指令则属于「数据传输」类别，而 TKON 指令与 TKOFF 指令则属于「工作管理」类别。



*.DINC 指令属于 API 应用指令下的「四则逻辑运算」类别。

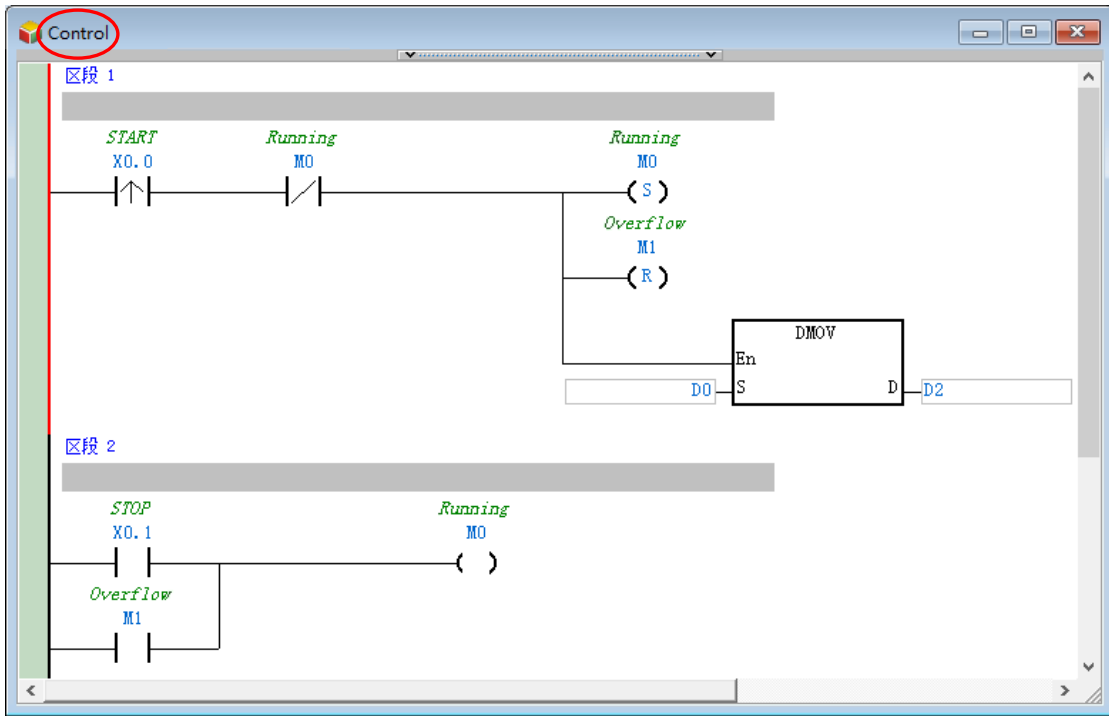


*.D- 指令属于 API 应用指令下的「四则逻辑运算」类别，比较接点 D< 则位于「接点类型逻辑运算」的分类中。



*.比较接点 D> 可由项目管理区的「应用指令」下的「接点类型逻辑运算」分类中找到。

5



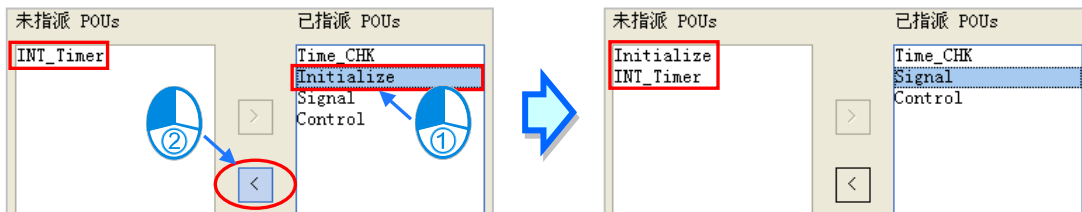
*.DMOV 指令属于 API 应用指令下的「数据传输」类别。

5

完成程序编辑后请打开工作管理工具。首先设置「周期性 (0)」的 TASK 属性，而因在程序的规划中，「周期性 (0)」TASK 的功能将会由「Initialize」这个 POU 来启动，因此我们必须先取消勾选「周期性 (0)」的「启动」项目。



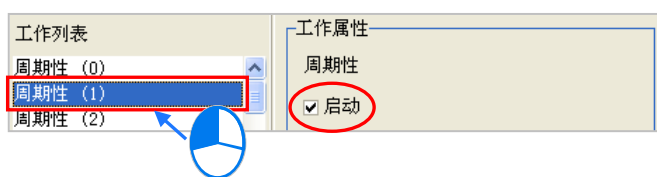
因「INT_Timer」须配置于「定时中断 0」的 TASK，而「Initialize」则须配置在「周期性 (1)」的 TASK 中，所以在此先将这两个 POU 由「周期性 (0)」的 TASK 清单中删除。



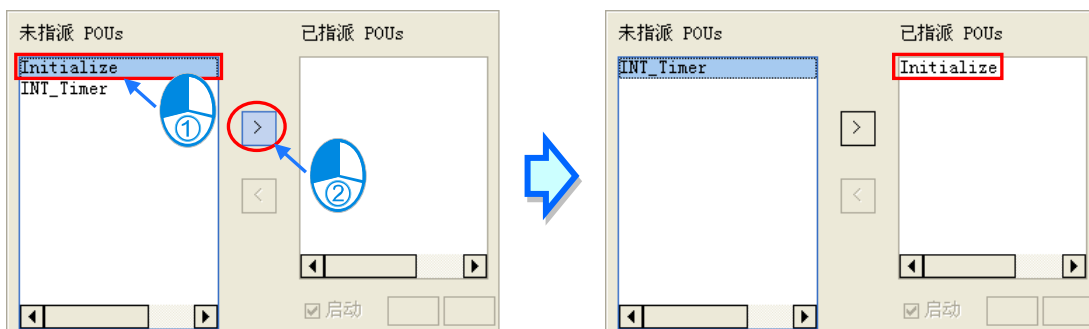
确定「周期性 (0)」的 TASK 清单后，接着便要对清单中的 POU 进行排序，而因为我们希望将与输入相关的功能放在扫描周期的最前面，而与输出相关的功能则放在扫描周期的最后，因此将 TASK 清单中的 POU 依照「Control」→「Time_CHK」→「Signal」的顺序重新排序。



接着我们选择「周期性 (1)」的 TASK，且因「周期性 (1)」的 TASK 必须先在一个扫描周期之后才会关闭，因此请直接保留其「启动」项目的勾选状态。

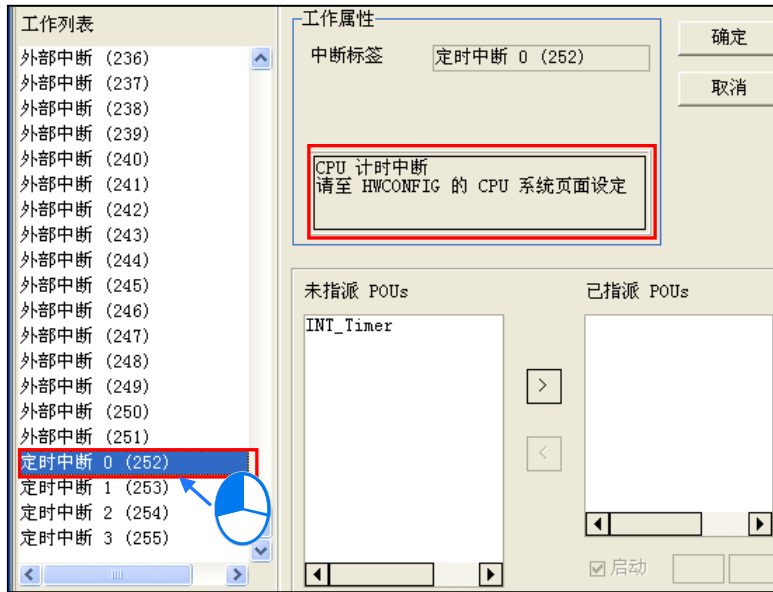


之后将「Initialize」这个 POU 配置到「周期性 (1)」的 TASK 中即可，且因 TASK 中只有一个 POU 所以无须进行排序。



5

接着我们选择「定时中断 0 (252)」的 TASK，而由工作属性区的提示得知，此 TASK 的属性必须在 HWCONFIG 中进行设置，因此我们可在此先完成 TASK 的配置，后续再另外进行工作属性的设置。

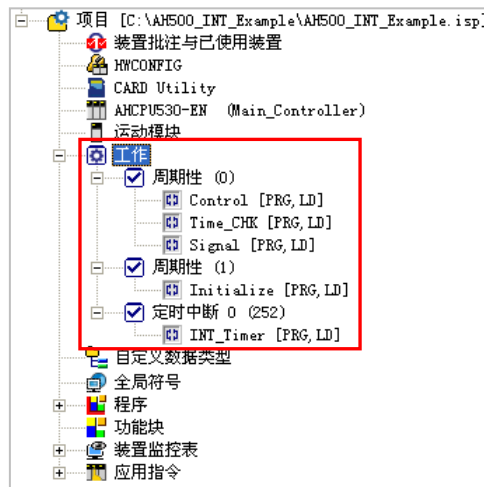


5

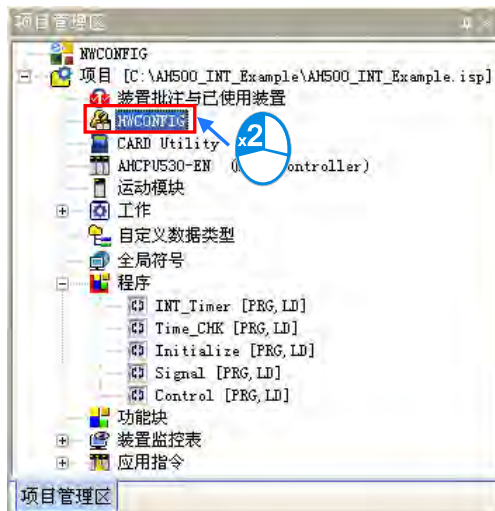
将「INT_Timer」这个 POU 配置到「定时中断 0 (252)」的 TASK 中。



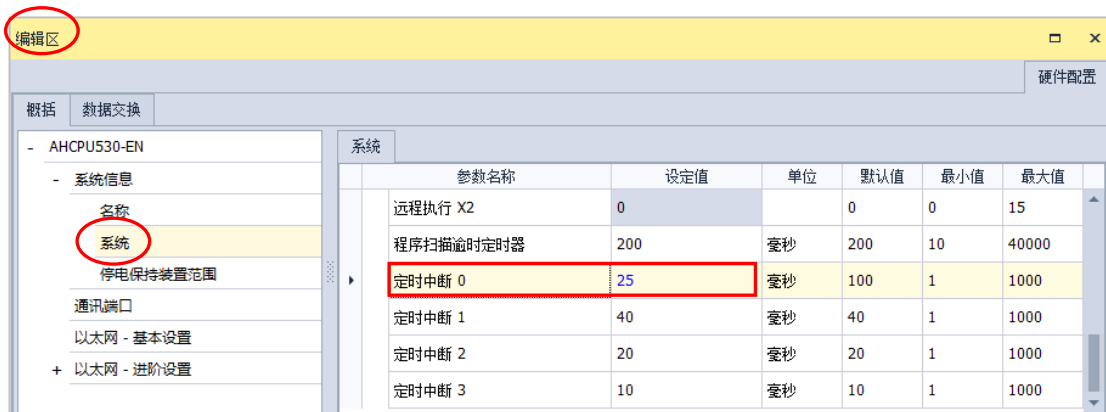
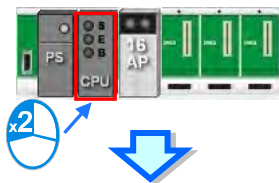
完成所有操作后请按下「确定」键应用在工作管理工具中所做的配置与设置，而配置的结果如下。



接着便要继续设置「定时中断 0」的工作属性。请先打开 **HWCONFIG** 工具。



打开 **HWCONFIG** 工具后，请于 CPU 模块的图标上双击鼠标左键以打开 CPU 的属性设置窗口。接着切换至「编辑区」主类别下的「系统」副类别，在「系统」副类别的页面中可见到「定时中断时间设定」的区域；将「定时中断 0」字段的值设定为 25ms 后，关闭窗口即可完成设定。



完成后请将 **HWCONFIG** 中的设置进行保存，并将此处的设置值一并下载至主机中才可正式生效。接着请将程序编译，并试着下载至主机进行测试。

MEMO

5

6

第6章 变量符号

目录

6.1	认识变量符号	6-2
6.1.1	变量符号的作用范围与命名原则	6-2
6.1.2	变量符号的类型	6-3
6.1.3	变量符号的数据类型	6-4
6.1.4	变量符号的地址配置与初始值	6-11
6.1.5	变量符号的变址寄存器修饰	6-13
6.1.6	变量符号的位操作 (限定 AH/AS 机种)	6-15
6.2	ISPSOft 中的变量符号管理	6-17
6.2.1	变量符号表	6-17
6.2.2	新增变量符号	6-20
6.2.3	数组与字符串类型符号的使用原则	6-26
6.2.4	变量符号的修改与符号表编辑	6-29
6.2.5	清除变量符号的配置地址	6-31
6.2.6	下载变量符号的初始值	6-32
6.2.7	符号表的导出	6-34
6.2.8	符号表的导入	6-35
6.2.9	符号表的排序	6-38
6.2.10	变量符号的筛选	6-40
6.2.11	装置资源分配设置	6-41
6.3	范例实作	6-42
6.3.1	变量符号规划	6-42
6.3.2	建立范例程序	6-43

6.1 认识变量符号

在传统的 PLC 程序开发过程中，往往需要花费很多的时间在管理装置地址，而在面对大型的项目时，不论在管理或除错上都会是很大的负担。有鉴于此，在 IEC 61131-3 中便导入了一般高级语言中的变量观念，除了可自行定义变量符号名称来替换 PLC 的装置编号之外，更可利用自动配置地址的功能来节省配置装置的时间，同时也提高了程序的可读性及开发效率。

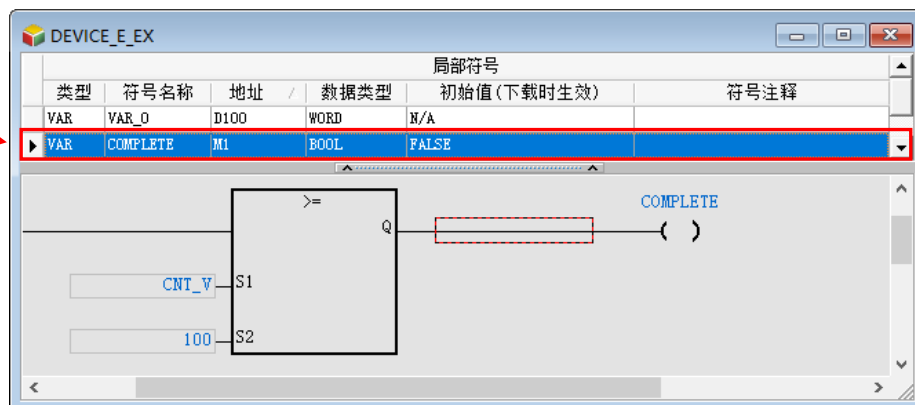
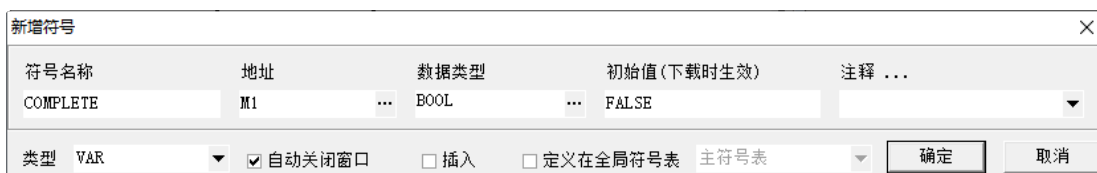
*变量在 ISPSOft 中被称之为符号 (Symbol)，因此在手册之中，变量与符号代表的将会是相同的意义。

6.1.1 变量符号的作用范围与命名原则

变量符号在使用前必须经过宣告的程序 (如下图)，且依据宣告位置的不同又可分为**全局符号**与**局部符号**；**全局符号**可让项目中的所有 POU 共享，而**局部符号**则只能在宣告该符号的 POU 当中使用。此外，在不同 POU 当中的局部符号名称是可以重复的，但若宣告的局部符号与某个全局符号的名称重复时，则在宣告该局部符号的 POU 当中，系统会自动将此名称的变量符号视为局部符号。

变量符号在命名时有其一定的规则与限制，说明如下。

- 变量符号的名称不可超过 30 个字符，且须注意一个中文字会占用两个字符。
- 不可使用系统保留的关键词，如指令名称、装置名称，或其它在系统中已被赋予特殊意义的保留字，但若只是被包含在名称当中则是合法的，如 “M0” 是不合法的名称，但 “_M0” 则是合法的。
- 变量符号的名称不可与列举名称重复，列举的用法请参阅第 8 章。
- 变量符号的名称当中不可有空白，例如 “INPUT CH0” 便是不合法的名称。
- 名称当中可使用下划线，但不可连续使用，或是置于结尾，例如 “ INPUT_CH0” 是合法的名称，但 “INPUT_CH0_” 与 “INPUT_CH0_” 则是不合法的名称。
- 变量符号的名称不可使用特殊字符，例如 *、#、?、\、%、@ ...等。



*. 变量符号请避免将第一个字命名为数字，以及 DFB_ (系统内定的函数库开头)；变量符号的地址建议使用自动配置以达分配的优化。

6.1.2 变量符号的类型

变量符号的类别依功能可分为下列 7 种，而在全局变量支持的符号类别与程序（局部变量）相同，程序（局部变量）与功能块（局部变量）及函数（局部变量）各种类别的特性与使用说明如下表所示。

变量符号 类别	局部变量						
	程序			功能块			函数
	AH/AS	DVP	AS5xx/ DVPxxMC	AH/AS	DVP	AS5xx/ DVPxxMC	AS5xx/ DVPxxMC
VAR	✓	✓	✓	不可指定 位置	只支持 ES3 且不可指 定位置	不可指定 位置	不可指定 位置
VAR_RETAIN	不可指定 位置 (*1)	只支持 ES3 且不可指 定位置 (*1)	✓			不可指定 位置	不可指定 位置
VAR_STATIC			✓			不可指定 位置	不可指定 位置
VAR_STATIC_ RETAIN			✓			不可指定 位置	不可指定 位置
VAR_INPUT				不可指定 位置	不可指定 位置	不可指定 位置	不可指定 位置
VAR_OUTPUT				不可指定 位置	不可指定 位置	不可指定 位置	
VAR_IN_OUT				不可指定 位置	不可指定 位置		

*1. 其数据类型亦不可为 **TIMER**、**COUNTER** 或 **STEP**。

● VAR - 一般变量符号

当变量符号被宣告成此类型时，代表该变量符号仅供一般的运算之用，而其所代表的意义则必须视宣告时，该变量符号被指定的数据类型或是对应的装置名称。

● VAR_RETAIN - 停电保持型的一般变量符号

当变量符号被宣告成此类别时，会自动被配置在断电保持的区域，因此在 PLC 主机断电后，该变量符号仍会保有最后的状态值。

● VAR_STATIC - 静态型的变量符号

具有静态属性，STATIC 类型的变量只能在定义该变量的 POU 中使用。当变量是 STATIC 类型时，系统会给该变量分配一个固定的地址，使得该变量的值在控制器运行过程于相同 POU 的不同 Instance 之间能够共享变量。

● VAR_STATIC_RETAIN - 静态型的断电保持变量符号

当变量符号被宣告成此类别时，会变成具有静态属性且有断电保持，因此在 PLC 主机断电后，该变量符号仍会保有最后的状态值。

● VAR_INPUT - 功能块输入端点的变量符号

此类型的变量符号主要用做功能块的输入端点，当功能块接受调用时，便可利用此类型的变量符号来接收调用方的输入值。在 LD/FBD 等编程环境中，此类型的变量符号会被排列在功能块图标的左侧，并会配置一个端点用以接受调用方的输入值。

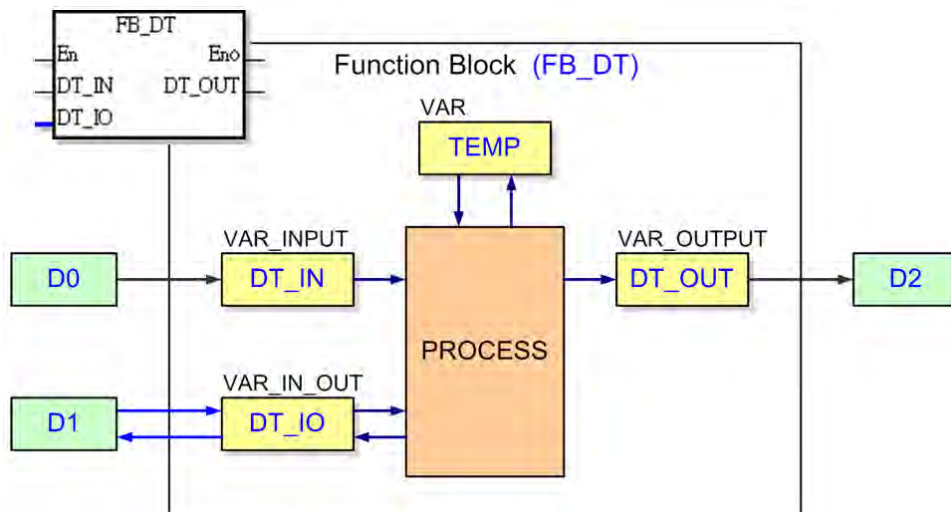
● VAR_OUTPUT - 功能块输出端点的变量符号

此类型的变量符号主要用做功能块的输出端点，当功能块执行完毕之后，其运算的结果便可经由此类型的变量符号来回传给调用方。在 LD/FBD 等编程环境中，此类型的变量符号会被排列在功能块图标的右侧，并会配置一个端点用以输出结果。

● VAR_IN_OUT - 功能块反馈型端点的变量符号

此类的变量符号主要用做功能块的反馈型端点，其功能请参考下例。

当功能块被调用时，调用方会先将 D1 的状态值输入 VAR_IN_OUT 类型的变量符号 DT_IO，而运算结束后，DT_IO 便会再将自己最后的状态值反馈输出至 D1。在 LD/FBD 等编程环境中，此类型的变量符号会被排列在功能块图标的左侧，且该端点会以蓝色的粗线来与调用方进行连结。



6.1.3 变量符号的数据类型

变量符号的数据类型会决定其内容值所代表的意义，例如有两个符号 VAR_1 与 VAR_2，其中 VAR_1 的数据类型为 BOOL，而 VAR_2 则是 WORD 类型；当两者在程序当中被使用时，VAR_1 便可用来代

表一个接点装置，而 VAR_2 则用来存放 16 Bits 大小的数值，且可进行四则运算或数据搬移。

下表为 ISPSOft 所支持的数据类型，而除了不同机种所支持的数据类型会有所不同之外，在全局变量与程序（局部变量）大致相同，但与功能块（局部变量）及函数（局部变量）可宣告的变量符号有所不同，请参考下表。

数据类型	说明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程序	功能块	程序	功能块	程序	功能块	函数
BOOL	布尔格式，使用 0、1 或 TRUE or FALSE 其内容值可代表一个接点装置的状态；AS5xx/DVPxxMC 则只支持 TRUE or FALSE，默认值为 FALSE。	✓	✓	✓	✓	✓	✓	✓
BYTE	8 Bits 数值格式，有效范围为 16#00 ~ FF 可存放 8 个 Bit 大小的数据，默认值为 16#00。					✓	✓	✓
WORD	16 Bits 数值格式，有效范围为 16#0000 ~ FFFF 可存放 16 个 Bit 大小的数据，默认值为 16#0000。	✓	✓	✓	✓	✓	✓	✓
DWORD	32 Bits 数值格式，有效范围为 16#00000000 ~ FFFFFFFF 可存放 32 个 Bit 大小的数据，默认值为 16#00000000。	✓	✓	✓	✓	✓	✓	✓
LWORD	64 Bits 数值格式，有效范围为 16#0000000000000000 ~ FFFFFFFFFFFFFFFF 可存放 64 个 Bit 大小的数据，默认值为 16#0000000000000000。	只支持 AH	只支持 AH			✓	✓	✓
SINT	8 Bits 的带符号整数格式，有效范围为-128 ~ 127 其最高位代表正负号，默认值为 0。					✓	✓	✓
INT	16 Bits 的带符号整数格式，有效范围为-32768 ~ 32767 其最高位代表正负号，默认值为 0。	✓	✓	只支持 ES3	只支持 ES3	✓	✓	✓

数据类型	说明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程序	功能块	程序	功能块	程序	功能块	函数
DINT	32 Bits 的带符号整数格式，有效范围为-2147483648 ~ 2147483647 其最高位代表正负号，默认值为 0。	✓	✓	只支持 ES3	只支持 ES3	✓	✓	✓
LINT	64 Bits 的带符号整数格式，有效范围为-9223372036854775808 ~ 9223372036854775807 其最高位代表正负号，默认值为 0。	只支持 AH	只支持 AH			✓	✓	✓
USINT	8 Bits 的无符号整数格式，有效范围为 0 ~ 255，默认值为 0。					✓	✓	✓
UINT	16 Bits 的无符号整数格式，有效范围为 0 ~ 65535，默认值为 0。					✓	✓	✓
UDINT	32 Bits 的无符号整数格式，有效范围为 0 ~ 4294967295，默认值为 0。					✓	✓	✓
ULINT	64 Bits 的无符号整数格式，有效范围为 0 ~ 18446744073709551615，默认值为 0。					✓	✓	✓
REAL	32 Bits 的浮点数格式，有效范围为-3.402823e+38 ~ -1.175495e-38、0 及 1.175495e-38 ~ 3.402823e+38，其内容为利用 32 Bits 的数据来表达一个具有小数点的数值，默认值为 0.0。	✓	✓	✓	✓	✓	✓	✓

数据类型	说明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程序	功能块	程序	功能块	程序	功能块	函数
LREAL	64 Bits 的浮点数格式，有效范围为-1.79769313486231e+308 ~ -2.22507385850721e-308、0 及 2.22507385850721e-308 ~ 1.79769313486231e+308，其内容为利用 64 Bits 的数据来表达一个具有小数点的数值，默认值为 0.0。	只支持 AH	只支持 AH			✓	✓	✓
ARRAY	数组格式，而数组长度与存放的数据类型则必须于宣告时一并指定。（最大的数组长度为 512）	✓	✓	✓	✓	✓	✓	✓
STRING	字符串格式，以 8 个 Bit 代表 1 个 ASCII 字符，AH/AS 进行宣告时须一并指定字符串长度，最多 128 个字符；AS5xx/DVPxxMC 的字符串格式在头尾，必须加上单引号，如 'abcd'，初始值最大长度可为 32000 个字符，若不设定初始值则字符默认长度为 80 个字	✓	✓	只支持 ES3	只支持 ES3	✓	✓	✓
STEP	用来作为步进点状态的识别旗标。（*1）（*2）	✓		✓	不支持 ES3			
功能块	代表一个功能块实例的名称（*2）。	✓	✓	✓	只支持 ES3	✓	✓	✓
COUNTER	计数器格式，用来代表计数器装置。	✓		✓	不支持 ES3			
TIMER	定时器格式，用来代表定时器装置。	✓		✓	不支持 ES3			
POINTER	类别选取 VAR_IN_OUT 才能选用 WORD 装置的指针类型（*2）。		✓		只支持 ES3			
T_POINTER	类别选取 VAR_IN_OUT 才能选用，定时器专用的指针类型（*2）。		✓		只支持 ES3			

数据类型	说明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程序	功能块	程序	功能块	程序	功能块	函数
C_ POINTER	类别选取 VAR_IN_OUT 才能选用，计数器专用的指针类型变量 (*2)。		✓		只支持 ES3			
HC_ POINTER	类别选取 VAR_IN_OUT 才能选用，高速计数器专用的指针类型变量 (*2)。		✓		只支持 ES3			
TIME	时间格式、可以使用 TIME#XXXXXXdXXhXXmXXsX XX.XXXms，或 T#XXXXXXdXXhXXmXXsXXX. XXXms 输入，以 ms 为单位。有效范围为： T#0ms~T#213503d23h34m33s7 09.551ms 数值可以为小数但不能出现负数，数值可以超出时间的有效范围，如 T#25h 与 T#1d1h； d、h、m、s、ms 单位可以减少，但至少保留一个，且不能任意排列；数值必须为 10 进制，默认值为 T#0ms。					✓	✓	✓
DATE	时间格式、可以使用 DATE#年- 月-日，或 D#年-月-日。有效范围为： D#1970-01-01~D#2106-02-07。 以日为单位。年、月、日均不能缺少，且不能超出有效范围；数值前可以新增多个 0；数值必须为 10 进制，默认值为 D#1970-01-01。					✓	✓	✓

数据类型	说明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程序	功能块	程序	功能块	程序	功能块	函数
TOD	<p>时间格式、可以使用 TIME_OF_DAY#时:分:秒.毫秒 或 TOD#时:分:秒.毫秒。有效 范围为：</p> <p>TOD#00:00:00~23:59:59.999 以 ms 为单位。数值为 0 时显示 TOD#00:00:00，数值为 1 时显示 TOD#00:00:00.001，数值为 86399999 时显示 TOD#23:59:59.999，数值为 86400000 时显示 TOD#00:00:00，数值为 4294967295 时显示 TOD#17:2:47.295，默认值为 TOD#00:00:00。</p>					✓	✓	✓
DT	<p>时间格式、可以使用 DT#年-月- 日-时-分-秒输入。有效范围为： DT#1970-01-01-0:0:0~DT#2106 -02-07-6:28:15。以 s 为单位，默 认值为 DT#1970-01-01-0:0:0。</p>					✓	✓	✓
Pointer	<p>指针变量存储的是一个内存的地 址，程序中使用函数 ADR 可得到 某个变量的地址；对符号使用「^」 可取得地址的内容数值。值得注意 的是，使用指标前必须先给指标 值，否则会造成运行异常！</p>					✓	✓	✓

数据类型	说明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程序	功能块	程序	功能块	程序	功能块	函数
Reference	需要设定初始值并且设定为相同数据类型的符号。表示为此符号的别名 (Alias)。举例，在符号表宣告一个数据类型为 REFERENCE TO INT 的符号 (例如：myref)，初始值字段填入被参考的符号(例如：mydata)，使用 myref 等同于使用 mydata。					✓	✓	✓

1. 关于步进点的说明请参考 第 14 章 - 顺序功能图 SFC。

*2. 功能块类型与指针类型的变量符号有其特殊意义，相关说明请参考 第 7 章-功能块 FB 与函数库。

6

6.1.4 变量符号的地址配置与初始值

每个变量符号都会依据其数据类型而配置一个对应的装置地址，并且可为其设置一个**初始值**，当下载项目程序时便可选择是否将**初始值**一并写入该变量符号所对应的装置地址中，**初始值**的相关操作请参考本章第 6.2.6 节的说明。

下表为变量符号的地址配置原则，而不同的机种，其配置原则也不相同。

机种	配置原则
AH/AS	<ul style="list-style-type: none"> ● 全局符号与程序 POU 的局部符号可自行指定对应的装置地址或交由系统自动配置。 ● 功能块的局部符号仅可交由系统自动配置，而无法自行指定。 ● 自动配置的地址会配置在预留的内存区块，而不会占用 D 或 M 装置；但针对 STEP、TIMER 与 COUNTER 几种类型，自动配置时仍旧会为其配置 S、T 与 C 装置。
DVP	<ul style="list-style-type: none"> ● 所有的全局符号与局部符号皆可自行指定对应的装置地址或交由系统自动配置。 ● 系统自动配置的地址皆会占用可用的装置地址。(用户可设置自动配置的装置范围)
AS5xx/ DVPxx MC	<ul style="list-style-type: none"> ● 全局符号与程序 POU 的区域符号可自行指定对应的装置地址或交由系统自动配置。 ● 功能块的区域符号仅可交由系统自动配置，无法自行指定。 ● 自动配置的地址会配置在主机预留的内存区块，不会占用 I/Q/M 装置。

下表则为各种数据类型可指定或配置的装置种类。

数据类型	AH/AS		DVP	
	自行指定	自动配置	自行指定	自动配置
BOOL	M / SM 接点及位操作的 D / L / X / Y 装置	内部存储器	M / X / Y 接点装置	M 接点装置
WORD	D / L / X / Y / E / SR 装置	内部存储器	D / E / F 装置	D 装置
DWORD	D / L / X / Y / E / SR 装置	内部存储器	D / E / F 装置	D 装置
LWORD	D / L / X / Y / E / SR 装置 (只支持 AH)	内部存储器	无此类型	
INT	D / L / X / Y 装置	内部存储器	无此类型	
DINT	D / L / X / Y 装置	内部存储器	无此类型	
LINT	D / L / X / Y 装置 (只支持 AH)	内部存储器	无此类型	
REAL	D / L / X / Y 装置	内部存储器	D 装置	D 装置
LREAL	D / L / X / Y 装置 (只支持 AH)	内部存储器	无此类型	
STRING	D / L / X / Y 装置	内部存储器	无此类型	
STEP	S 装置 (只支持 AH5x0)	S 装置	S 装置	S 装置
COUNTER	C / HC 装置	C 装置	C 装置	C 装置

数据类型	AH/AS		DVP	
	自行指定	自动配置	自行指定	自动配置
TIMER	T 装置	T 装置	T 装置	T 装置
ARRAY	ARRAY 类型的变量符号会根据指定的数组数据类型决定所配置的装置类型，且会由自行指定或自动配置的起始地址开始，连续配置符合数组长度的装置数量。 ARRAY 所指定的装置不可为 SR / SM / E / F。			

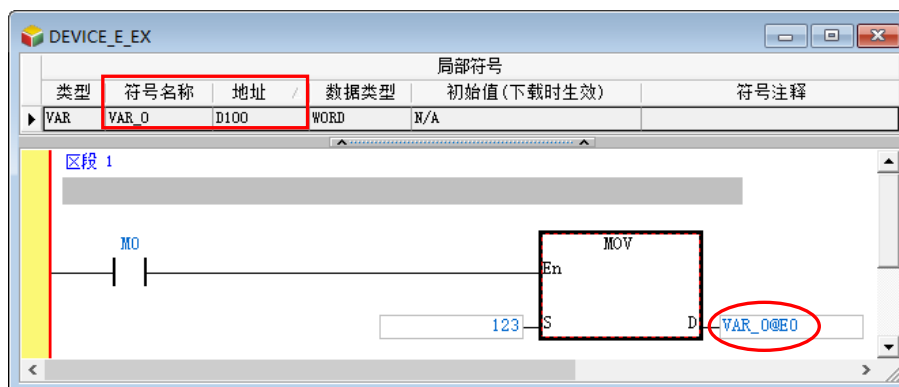
数据类型	AS5xx/DVPxxMC	
	自行指定	自动配置
BOOL	I/Q/M 的 B 与 X	内部存储器
BYTE	I/Q/M 的 B 装置	内部存储器
WORD	I/Q/M 的 W 装置	内部存储器
DWORD	I/Q/M 的 D 装置	内部存储器
LWORD	I/Q/M 的 L 装置	内部存储器
USINT	I/Q/M 的 B 装置	内部存储器
SINT	I/Q/M 的 B 装置	内部存储器
UINT	I/Q/M 的 W 装置	内部存储器
INT	I/Q/M 的 W 装置	内部存储器
UDINT	I/Q/M 的 D 装置	内部存储器
DINT	I/Q/M 的 D 装置	内部存储器
LINT	I/Q/M 的 L 装置	内部存储器
ULINT	I/Q/M 的 L 装置	内部存储器
REAL	I/Q/M 的 D 装置	内部存储器
LREAL	I/Q/M 的 L 装置	内部存储器
STRING	I/Q/M 的 B 装置	内部存储器
TIME	I/Q/M 的 L 装置	内部存储器
DATE	I/Q/M 的 D 装置	内部存储器
TOD	I/Q/M 的 D 装置	内部存储器
DT	I/Q/M 的 D 装置	内部存储器
ARRAY	I/Q/M 装置，根据类型决定为何种装置(B、W、D、L)。WORD Array 不支持 WORD[ArrayIndex].Bit 的指定地址写法	内部存储器

6

- *1. AH/AS 机种中，当变量符号的对应装置地址被配置于内部存储器时，用户无法得知实际的地址值。
- *2. 关于 ARRAY 与 STRING 符号的相关注意事项，请参考本章 第 6.2.3 节 的相关说明。
- *3. 关于 DVP 机种自动配置范围的设置方式与限制，请参考本章 第 6.2.11 节 的相关说明。
- *4. 功能块类型的变量符号具有特殊意义，相关说明将于 第 7 章 - 功能块 FB 与函数库当中另行介绍。
- *5. 位操作指的是以 X0.0、Y0.1...等格式来表达字中的某个位，详细说明请参考 附录 B 的相关内容。

6.1.5 变量符号的变址寄存器修饰

ISPSOft 允许使用变址寄存器来修饰变量符号，其格式为 **符号名称@变址寄存器**，其中变址寄存器可以是一个 E 或 F 装置的编号，也可以是一个指定 E、F 装置地址的变量符号。



变址寄存器内部储存的数据，其意义代表被修饰对象的偏移值；以上图为例，当变址寄存器 E0 的值为 2 时，则 VAR_0@E0 代表的便是 VAR_0 所指定的装置 (D100) 再偏移 2 个地址，即装置 D102，而当 M0 为 ON 时，被变更内容的对象便会是 D102。

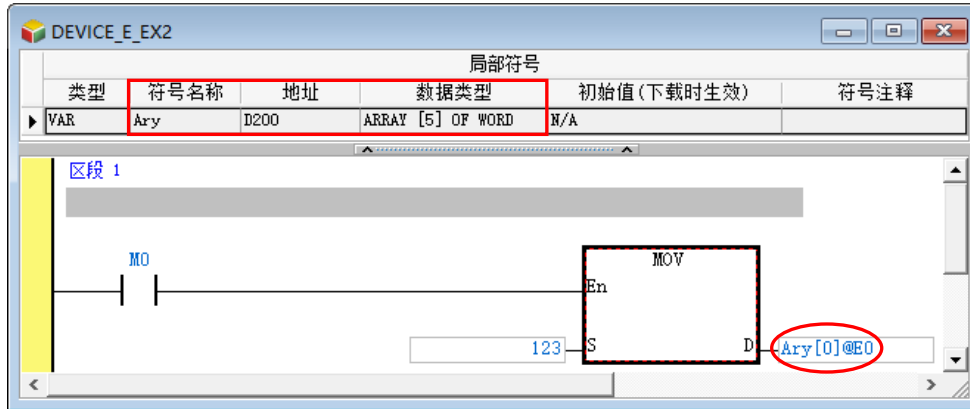
装置名称	状态	值(16位)	数值类型
E0		2	有号数十进制
M0			
D100		0	有号数十进制
D101		0	有号数十进制
D102		0	有号数十进制
D103		0	有号数十进制
D104		0	有号数十进制

➔

装置名称	状态	值(16位)	数值类型
E0		2	有号数十进制
M0			
D100		0	有号数十进制
D101		0	有号数十进制
D102		123	有号数十进制
D103		0	有号数十进制
D104		0	有号数十进制

此规则亦可应用于数组类型的变量符号中，其格式为 **数组符号名称[索引]@变址寄存器**，其中索引值必须为常数格式，若索引值为变量符号的格式时，将不再允许使用变址寄存器来修饰。

如下图，Ary 为一含 5 个元素的数组，且其指定的起始地址为 D200，而系统在编译时便会自动为其配置 D200~D204 这个区间的装置；因此若变址寄存器 E0 的值为 2，则 Ary[0]@E0 的操作对象便会是 Ary[0] 的地址 D200 再偏移 2 个地址，即装置 D202，同时也是数组元素 Ary[2] 的地址。但若再将变址寄存器 E0 的值为变更为 6，则 Ary[0]@E0 的操作对象便会是 D206，很明显已经超出此数组所涵盖的范围，不过这样的使用仍是合法的，因为变址寄存器只负责忠实的进行地址偏移而已，所以若拿来作为数组索引的用途时请特别小心。关于数组的使用将于第 6.2.3 节另行介绍。



关于利用**变址寄存器**来修饰操作数的用法，在不同系列的机种上亦会有所差异，请参考下表。

应用场合	范例	AH/AS	DVP	AS5xx/DVPxxMC
修饰指令操作数	MOV 100 VWD@E0	依指令规则而定	依指令规则而定	不支持
修饰接点或线圈	LD VBL_0@E0 OUT VBL_1@E1	支持	依机种而定	不支持
修饰数组类型的变量符	ARY[0]@E0	支持	支持	不支持

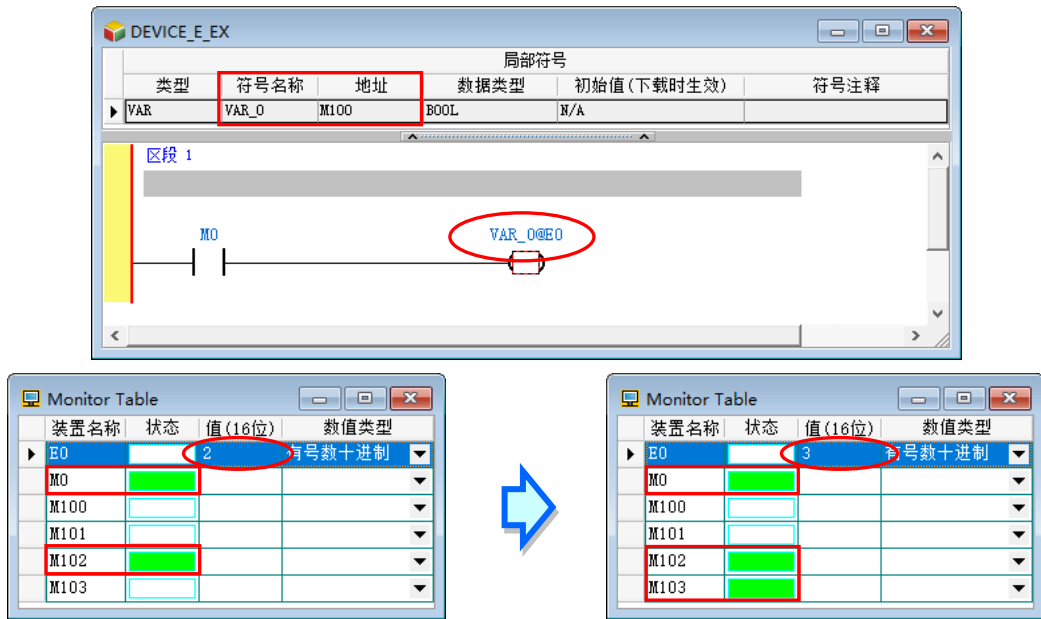
*1. AH/AS 系列的机种当中，变址寄存器只有 E 装置，而 DVP 系列机种除了 E 装置之外尚有 F 装置可搭配使用，相关说明请参考各机种的操作手册或技术文档。

*2. 变址寄存器不可再拿变址寄存器来修饰，例如 E0@E1，因此若拿变址寄存器来修饰变量符号时，请确认该符号的指定地址不可为 E 装置或 F 装置。

6

补充说明

- 变址寄存器**的使用是针对某个装置进行寻址的偏移，但若所修饰的变量符号，其装置地址是采用自动配置时，因用户无法得知该变量符号会被随机配置到哪个地址，因此贸然使用**变址寄存器**将可能造成程序执行的错乱。
- 当经**变址寄存器**修饰后的装置地址超过可用范围时，PLC 的运作将可能造成错误，因此在使用上请务必小心；以 AHCPU530-EN 机种为例，使用 D65500@E0 为操作数，当 E0 的值超过 35 时，便会造成寻址的范围超过 D 装置的可用范围 D65535，而使得 PLC 进入错误状态。
- 当**变址寄存器**的值被变更时，实际操作的装置便会随即变更，因此若原本操作的装置没有在程序的其它位置被使用到的话，该装置的值便会保留在最后的状况；以 AHCPU530-EN 机种为例，如下图程序，当 E0 的值为 2 时，M0 的状态会输出给 M102，因此 M102 的状态为 ON；但当 E0 的值由 2 变成 3 时，实际操作的装置便会变更成 M103，此时 M103 便会为 ON，而因 M102 在程序中已经没有其它程序对其进行操作，因此 M102 便会继续保持在 ON 的状态。

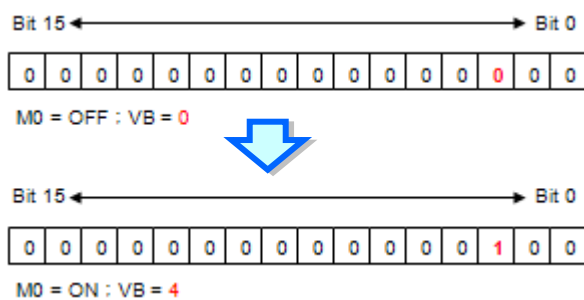
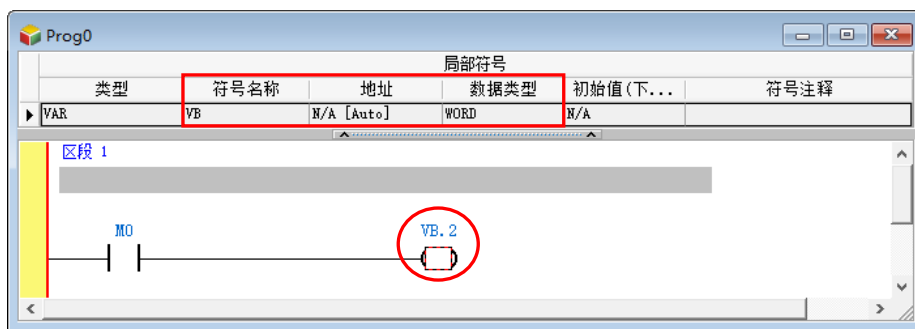


6.1.6 变量符号的位操作 (限定 AH/AS 機種)

AH/AS 系列機種的字裝置可支持位操作，其格式為 字裝置.位編號，例如 D0.2 便代表操作對象為裝置 D0 中的 Bit 2；而此項規則亦可應用至變量符號，格式為 符號名稱.位編號，如 VB.2；同時也可應用至陣列類型的變量符號，其格式為 陣列符號名稱[索引].位編號，例如 Ary[0].1。

位編號只可使用 10 進制常數，且不論所修飾的變量符號為何種數據類型，**位編號**的使用範圍只能為 0~15；且亦不可用於 BOOL、STEP、COUNTER、TIMER、功能塊、HC_POINTER、C_POINTER 及 T_POINTER...等類型的變量符號。

參考下图程序，變量 VB 為 WORD 格式，且假設其目前的值為 0，則當 M0 為 ON 時，VB 的 Bit2 就會為 1，此時 VB 的值便會變成 4。



6

在 AH/AS 系列机种中，变址寄存器与**位操作**可以混合使用，而当两者混合使用时，「@」符号会有较高的优先权。请参考下列的范例，其中变量 VB 为 WORD 格式。

EX1 : VB.1@E0 (VB 指定为 D100 ; E0=3)

解读方式为先解读 1@E0，因此 VB.1@E0 等同 VB.4，而实际的操作对象便会是 D100 的 Bit 4。

而 AH 系列机种另外还支持，以下的修饰方式。

EX1 : VB@E0.1 (VB 指定为 D100 ; E0=3)

先解读 VB@E0，因此内容会解读为 D103.1，而实际的操作对象便会是 D103 的 Bit 1。

EX2 : VB@E0.1@E1 (VB 指定为 D100 ; E0=3 ; E1=2)

解读方式为将 VB@E0，与 1@E1 分开解读。在此 VB@E0 等同于 D103，而 1@E1 则等于 3，因此整个叙述可解读为 D103.3，所以实际的操作对象便会是 D103 的 Bit 3。

- *1. 使用位操作时，系统仅会忠实的取得数据中的某个位状态，至于该位状态是否具有意义，则须由用户自行判断。
- *2. 在现阶段的 ISPSoft 中，可使用的位编号范围只有 0~15，不过在将来新版的软件中，系统将视变量符号的数据长度，而允许使用 0~31 或 0~63 的位编号范围。

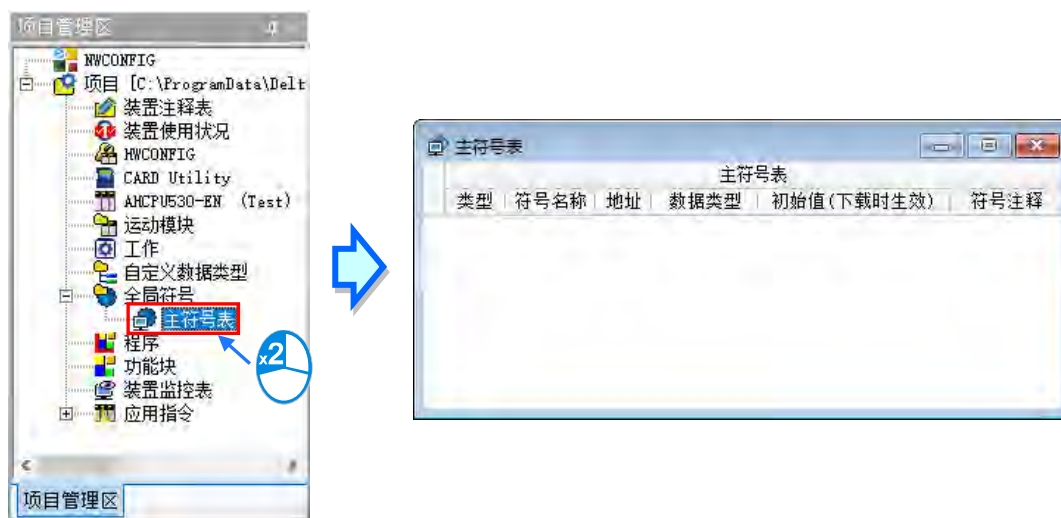
6

6.2 ISPSOft 中的变量符号管理

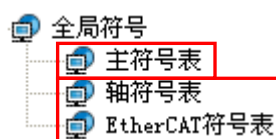
6.2.1 变量符号表

● 全局符号表

在项目管理区的「全局符号」节点中「主符号表」项目上双击鼠标左键，双击鼠标左键主符号表便可开启主符号表。



部分机种的全局符号表又依照功能的不同而分类了数个子项目，例如 AHxxEMC 系列机种的全局符号表可展开如下图，主符号即表为一般的全局符号表，其它的符号表则为对应不同功能所产生的全局符号表，这些符号表中的符号在程序中也可如全局符号般被使用，但建立符号的方式请参阅各相关功能的说明章节内容。



6

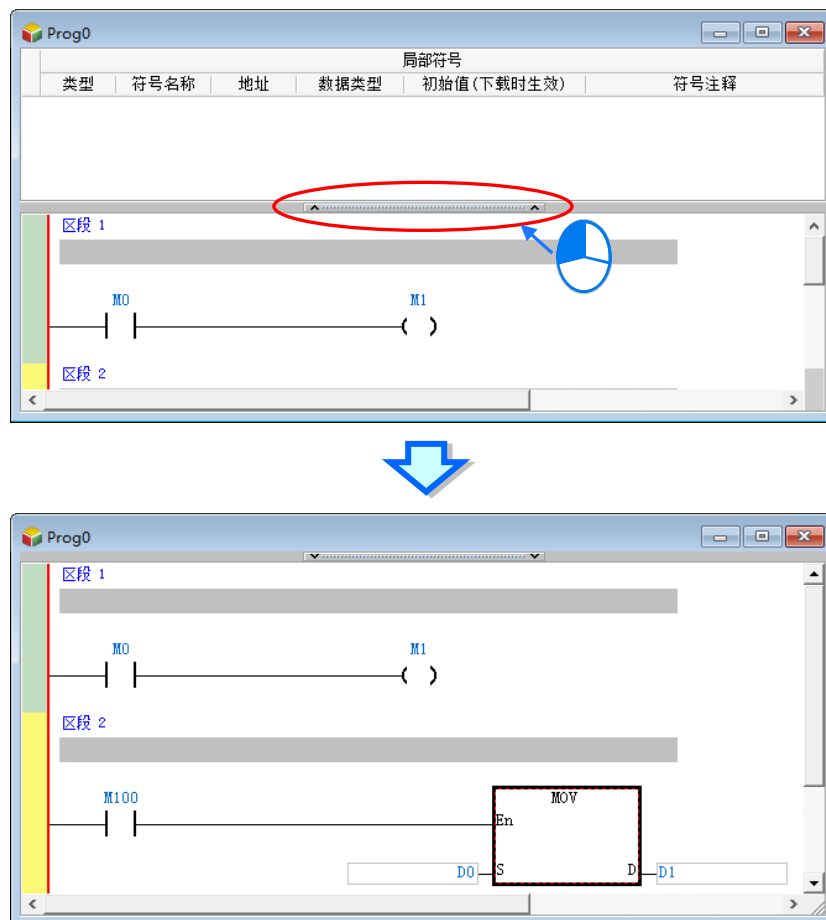
在项目管理区的「全局符号」项目上双击鼠标右键，接着单击「新增(N)」及可新增自定义变量符号表，以使用户方便管理全局变量，而新增完成后便会现于「全局符号」下方节点。



在项目管理区的「全局符号」下方节点中所新增的自定义变量符号表上单击鼠标右键，接着单击「删除」及可将其删除，而删除完成后便会消失于「全局符号」下方节点。

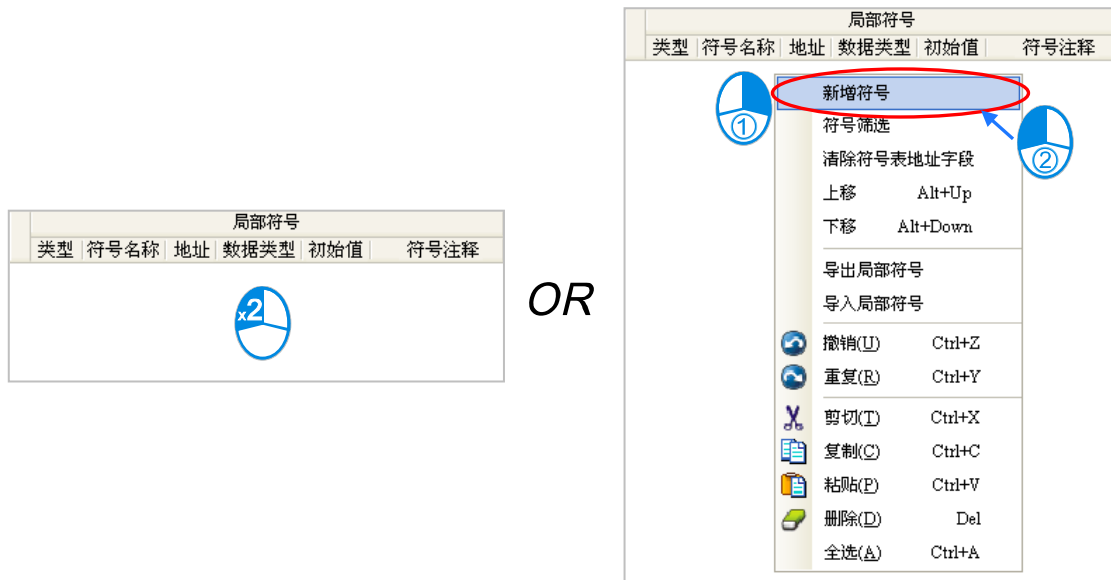
- 局部符号表

在 ISPSOft 中，局部符号表被安排在每个 POU 窗口的上方，而当用鼠标单击局部符号表下方的按钮时，便可将局部符号表隐藏起来，再单击一次便可重新开启。



6.2.2 新增变量符号

- (1) 直接在符号表的空白处双击鼠标左键，或于空白处单击鼠标右键后，于选项中单击「新增符号」。



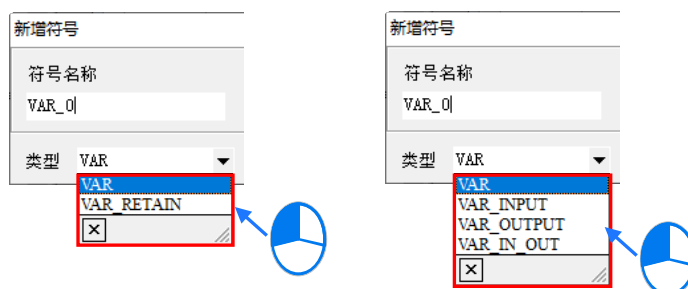
- (2) 执行新增符号后，画面便会出现变量符号的宣告窗口。请先在「符号名称」字段输入欲使用的变量符号名称，而命名的规则请参考本章第 6.1.1 节的相关说明。

6



- *1. 随着不同的宣告位置，如不同机种、全局/局部，出现的窗口亦会有所差异，不过基本的字段与设置方式都是相同的。
- *2. 在各种编程语言的环境中都有自动导出宣告窗口的方法，详细说明请参考手册第 8 章~第 12 章的相关内容。

- (3) 在「类型」字段的下拉选项中选择变量符号的类型，而随着变量符号宣告的位置不同，该字段可选择的项目亦会有所不同。相关说明请参考本章第 6.1.2 节的内容。



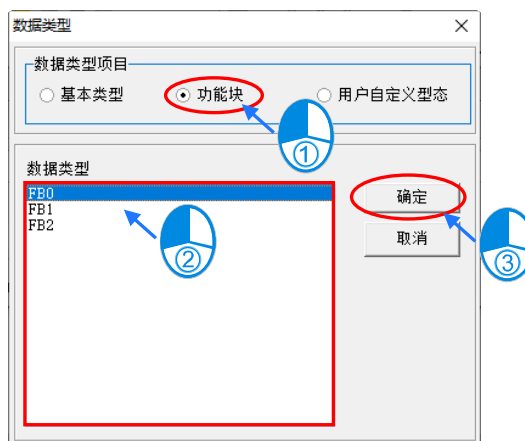
- (4) 按下「数据类型」字段右侧的按钮，并于随后弹出的窗口中设置变量符号的数据类型，完成后按下「确定」键即可。关于各种数据类型的说明请参考本章第 6.1.3 节的内容。



关于功能块、用户自定义类型、STRING 与 ARRAY 等几种数据类型的设置，补充说明如下。

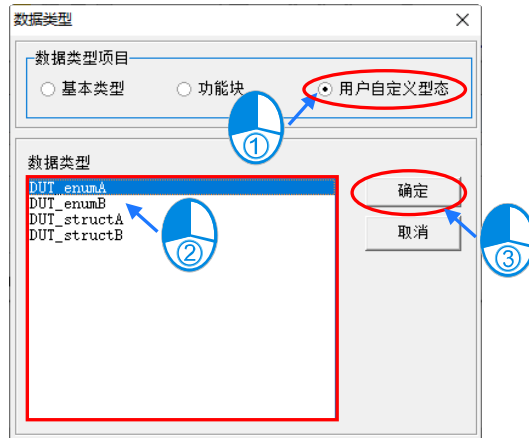
● 功能块

在「数据类型项目」中选择「功能块」，之后再在清单中指定对应的功能块定义。宣告为此数据类型的变量符号便代表宣告一个所选功能块的实例，详细说明请参考手册第 7 章的相关内容。



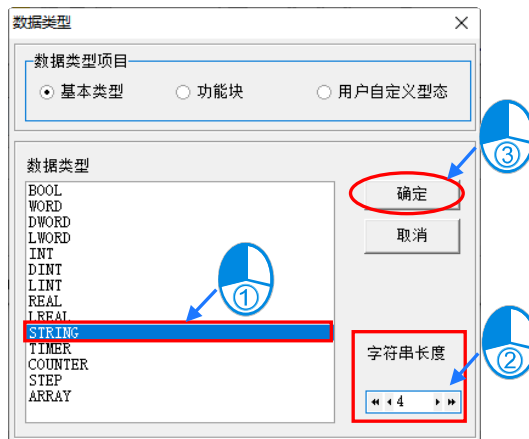
● 用户自定义类型

「数据类型项目」中选择「用户自定义类型」，之后再在清单中指定已宣告的 DUT。宣告为此数据类型的变量符号，详细说明请参考手册第 8 章的相关内容。



● STRING

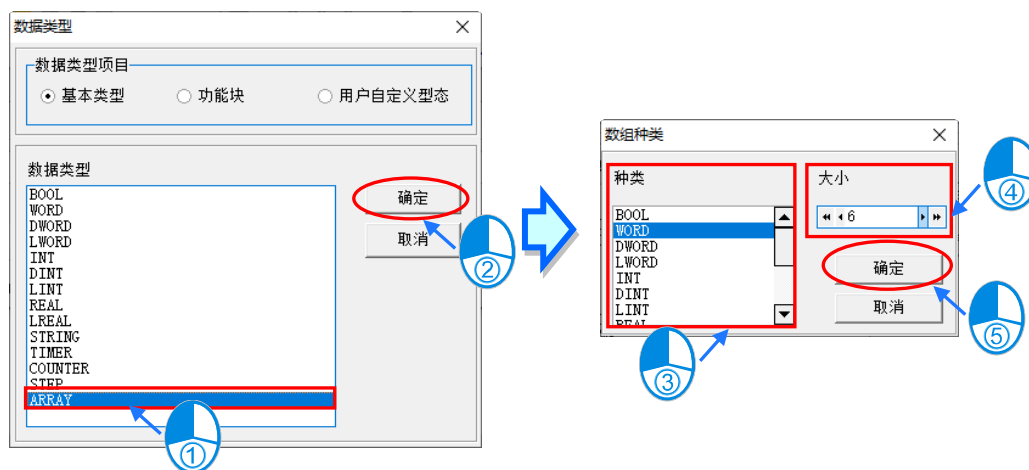
若选择的数据类型为 STRING，则请一并在窗口的右下角设置字符串的最大长度，其范围为 1~128，而所设置的值则代表此字符串符号最大可容纳的字符数。设置时以鼠标单击 ◀ 或 ▶ 按钮便可递减或递增设置值的个位数，而单击 ◀◀ 或 ▶▶ 按钮便可递减或递增设置值的十位数。



6

● ARRAY

若选择的数据类型为 ARRAY，则按下确定后便会出现 ARRAY 的设置窗口，请在此窗口中一并设置 ARRAY 的种类与大小；而其中数组大小的范围为 1~512，设置时以鼠标单击 ◀ 或 ▶ 按钮便可递减或递增设置值的个位数，而单击 ◀◀ 或 ▶▶ 按钮便可递减或递增设置值的十位数。



- (5) 若「地址」字段保持空白，则代表此变量符号的装置地址将交由系统自动配置；而若要指定装置地址时，可直接于「地址」字段输入装置地址或单击该字段右侧的按钮，并于随后弹出的窗口中指定使用装置。相关说明请参考本章第 6.1.4 节的内容。



OR



*1. 当选择的数据类型为功能块时，「地址」字段将无法进行设置。

*2. 若是支持可使用位操作机种，而此装置设定窗口便会有「位编号」的字段，填入编号可以表示，如 X0.0、D0.1...

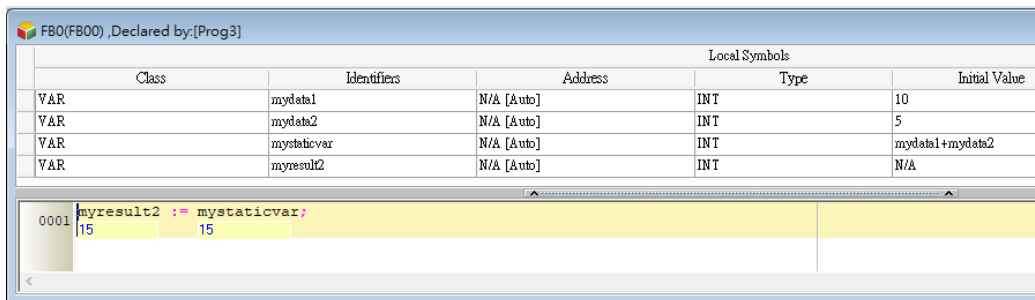
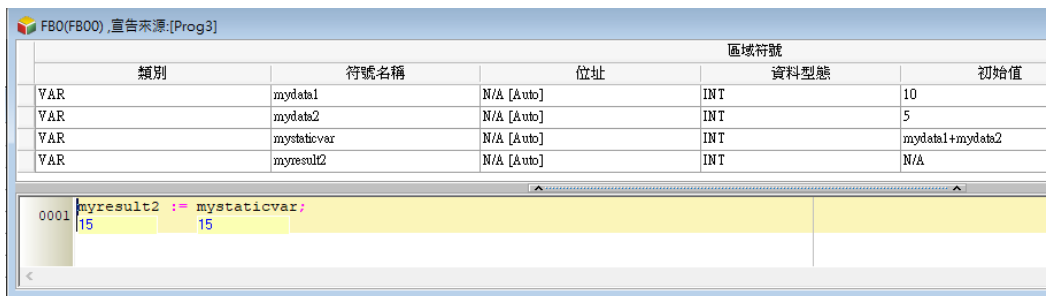
- (6) 若须为此变量符号设置初始值时，请在「初始值」字段中直接输入即可，而所设置的值必须符合变量符号的数据类型及大小；且当变量符号的数据类型为 **ARRAY** 时，单击「初始值」字段后会出现一个初始值的设置窗口，之后便可于该窗口中逐一设置各个数组元素的初始值。

BOOL 符号的初始值为 TRUE 或 FALSE，且在输入时亦可简略的输入 T 或 F，或者直接输入 1 或 0，系统将会在输入完毕之后自动进行转换；而在输入 **字符串符号** 的初始值时，其字符数不得超过所设置的字符串长度，且亦无须为该字符串加上双引号，否则加上的双引号将会被视为字符串内容的一部分，若是选择 **AS5xx/DVPxxMC** 机种，在输入 **字符串符号** 的初始值时，需要对该字符串的头尾加上单引号的符号；若符号的数据类型为功能块或使用者自定义形态时，「初始值」字段将不可设置。



在 DVPxxMC 与 AS5xx 的初始值支持 ST 表达式的写法，如下图。

6



- (7) 欲加入注释时，请先在「注释」字段单击鼠标左键，接着便可于出现的下拉页面中输入注释。



- (8) 宣告窗口中尚有三个可勾选的选项，其功能请参考下述。而确认所有的设置后，按下「确定」键即可完成一个变量符号的新增。



● 自动关闭窗口

若取消此项的勾选，则按下确定后，宣告窗口便不会自动关闭，如此即可方便用户连续宣告变量符号。而其默认值的设置方式请参考第 2.3.1 节中的「系统基本设定 - 符号表」项目。

- 插入

当勾选此项时，右方下拉选项会出现**主符号表**及**自定义全局符号表**，此处宣告的变量符号便会自动被宣告在所选择的主符号表或自定义全局符号表中；而若一开始开启宣告窗口的位置就是在主符号表或自定义全局符号表时，此选项便不会出现。

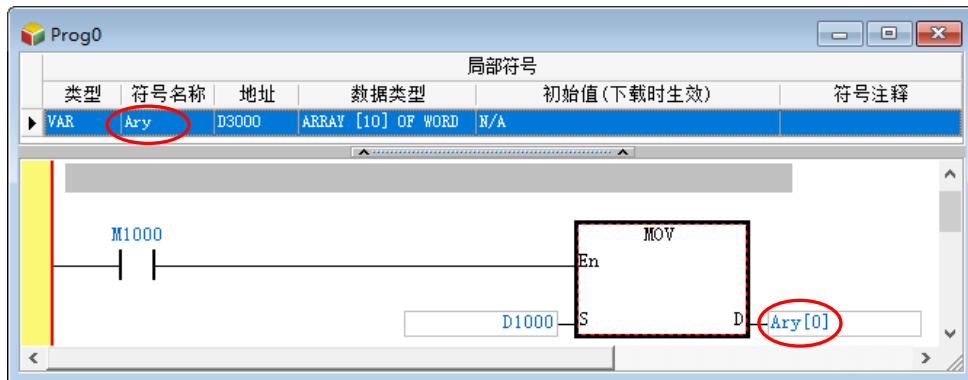
- 宣告在全局符号表

当勾选此项时，此处宣告的变量符号便会自动被宣告在全局符号表中；而若一开始开启宣告窗口的位置就是在全局符号表时，此选项便不会出现。

6.2.3 数组与字符串类型符号的使用原则

- 数组类型的变量符号

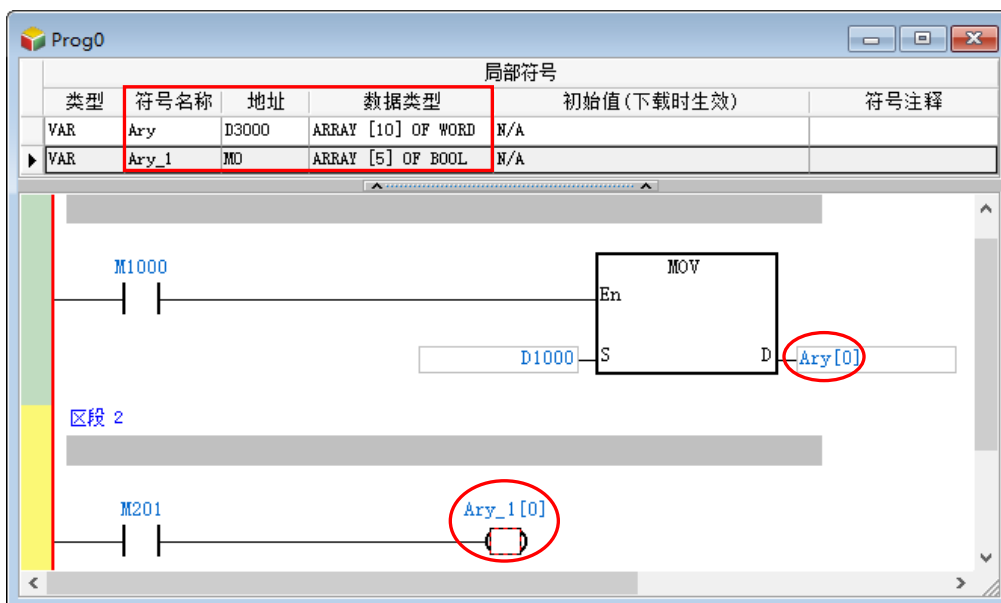
如下图范例，当要在程序当中存取数组类型的变量符号时，其表达的格式为 **符号名称[索引]**，而在一般的状况下，**索引值**只可为 10 进制常数，但若是 AH/AS 系列机种的项目，且选择的为 ST 语言时，**索引值**的部份则可为变量符号；而不论是常数或变量，**索引值**的内容必须由 0 开始，且不可大于或等于数组的大小，例如大小为 10 的数组符号，其**索引值**的范围则必须为 0~9；此外，当使用变量做为**索引值**时，请小心避免变量的内容值超出**索引值**的合法范围，否则将可能在运作时造成数据的错乱。



宣告数组类型的变量符号时，除了须指定数组元素的数据类型之外，亦须一并指定数组的大小，范围则为 1~2048；且在自行指定数组符号的起始地址时，须根据数组的数据类型而指定对应的装置，但须注意不可指定为 SR、SM、E、F...等装置，关于各种数据类型所对应的装置类别请参考第 6.1.4 节的相关说明；而在配置数组符号所使用的地址区间时，不论是自行指定或自动配置，皆会由起始的地址开始，连续配置符合数组长度的装置数量。

如下图，其中 **Ary_0** 的「数据类型」字段为 **ARRAY[3] OF DWORD**，代表该数组的大小为 3，亦即包含 3 个元素，且其元素的数据类型为 DWORD，而由「地址」字段可知所指定的起始地址为 D0，因此该数组将会占用 D0~D5 共 6 个 WORD (3 个 DWORD) 的装置地址。

接着再看 **Ary_1** 的数组符号，其「数据类型」字段为 **ARRAY[5] OF BOOL**，代表该数组的大小为 5，且元素的数据类型为 BOOL，而由「地址」字段可知所指定的起始地址为 M0，因此该数组将会占用 M0~M4 共 5 个 M 装置；若是使用 DVPxxMC 与 AS5xx 机种，「数据类型」栏位为 **ARRAY[10] OF BOOL**，地址指定 %IB0，则代表该阵列会占用到 %IB0 ~ %IB9 装置，共 10 个 %IB 装置。



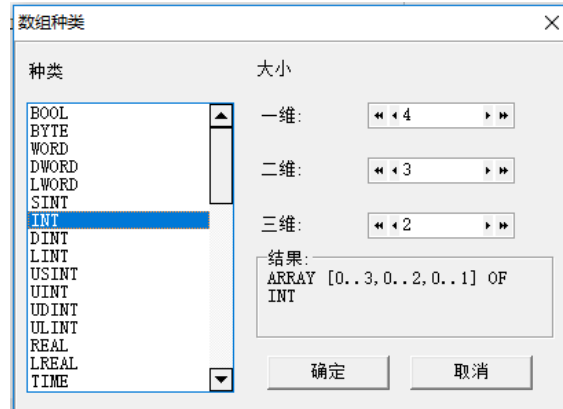
在宣告数组类型的变量符号时可各别为每个元素定义其初始值，请参考下列范例。

A_Ary 的「初始值」字段显示为 [1,2,3,4,5]，代表数组元素 A_Ary[0] 的初始值为 1，A_Ary[1] 的初始值为 2，其它依此类推；而 B_Ary 的「初始值」字段则显示为 [1,3(0),5]，其中的第二个初始值为 3(0)，代表由第二个元素开始，有连续 3 个元素的初始值皆为 0。

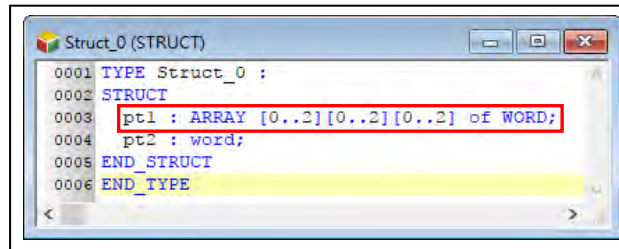
局部符号						
类型	符号名称	地址	数据类型	初始值	符号批注	
VAR	A_Ary	D7000	ARRAY [5] OF WORD	[1, 2, 3, 4, 5]		
VAR	B_Ary	D7005	ARRAY [5] OF WORD	[1, 3(0), 5]		

A_Ary		B_Ary	
元素	初始值	元素	初始值
A_Ary[0]	1	B_Ary[0]	1
A_Ary[1]	2	B_Ary[1]	0
A_Ary[2]	3	B_Ary[2]	0
A_Ary[3]	4	B_Ary[3]	0
A_Ary[4]	5	B_Ary[4]	5

在 DVPxxMC 与 AS5xx 的机种支持多维数组数据类型，可定义最多三维数组，于宣告的时候可以在数组种类窗口选取数组大小输入，如下图。



另外，当自定义数据类型中具有二维或三维数组的元素时，此类型的变量符号将可使用 $[n,n]$ 或 $[n,n,n]$ 加以指定。如下图自定义数据类型中的元素 `pt1` 为三维数组，则程序中的变量符号 `vardut` 可以 `vardut.pt1[1,1,1]` 的方式指定数组中的第 $[1,1,1]$ 个数组元素。

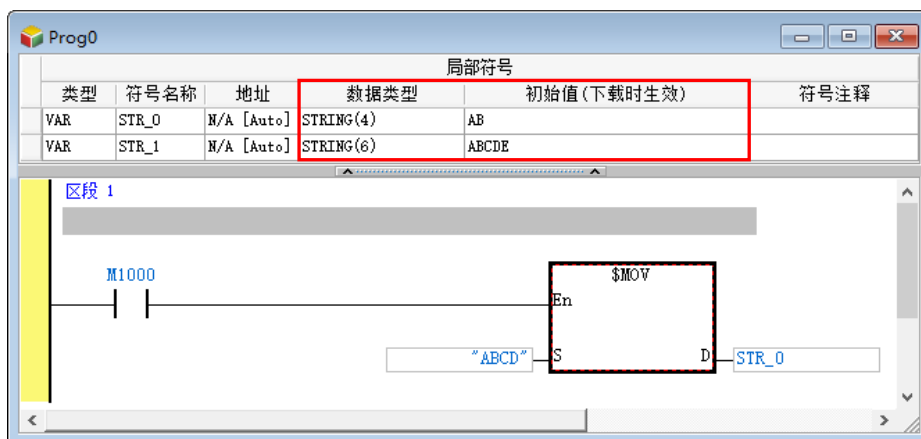


6

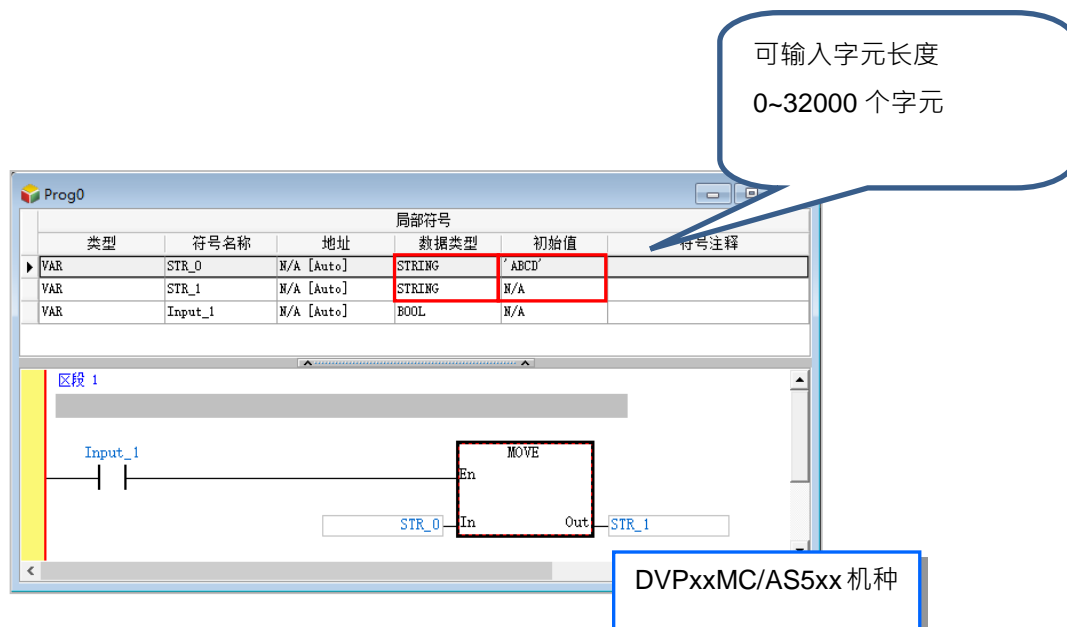
● 字符串类型的变量符号

ISPSOft 中的字符串是以 ASCII 格式来表达，而字符串常数的表达格式则是加上双引号，例如 "ABCD"，而 DVPxxMC 与 AS5xx 机种则是以单引号，例如 'ABCD'；且字符串类型的变量符号通常会搭配 API 应用指令来进行操作。相关的使用规则请参考各机种的应用指令手册或相关技术文件。用户在宣告字符串类型的符号时便须一并指定字符串的最大长度，指定的范围为 1~128，其中 1 个字符就会占用 1 个 Byte 的空间；而为其配置地址时，除了配置符合最大长度的空间之外，尚须多配置一个 Byte 的空间以储存字符串的结束字符，且若配置的最后一段空间未满 1 个 WORD 时，便须补足 1 个 WORD；例如 3 个字符的字符串须配置 2 个 WORD 的空间 (4 Bytes)，而 4 或 5 个字符的字符串则须配置 3 个 WORD (6 Bytes)。

请参考下图的字符串符号，其中在数据类型 STRING 后的括号中会显示字符串的最大长度；指定字符串符号的初始值时无须加上双引号，且初始值的字符串长度必须小于或等于宣告的最大长度。



若用户选择的是 DVPxxMC 与 AS5xx 机种，宣告字符串时长度由软件配置，用户无需调整，只有在指定初始值时，初始值字符长度为 0~32000 个字符，在宣告完后，其数据类型 STRING 后的括号中不会显示字符串的最大长度。



6

6.2.4 变量符号的修改与符号表编辑

若要修改变量符号本身的属性时，请直接于欲修改的变量符号上双击鼠标左键，之后便可再次开启该变量符号的宣告窗口，接着参考本章第 6.2.2 节的说明修改各字段的设置值即可。

局部符号					
类型	符号名称	地址	数据类型	初始值(下载时生效)	符号注释
VAR	VAR_0	N/A [Auto]	BOOL	TRUE	
VAR	VAR_1	N/A [Auto]	WORD	0	

修改符号

符号名称	地址	数据类型	初始值(下载时生效)	注释...
VAR_0	...	BOOL	...	TRUE

类型: VAR 自动关闭窗口 定义在全局符号表 主符号表

当要对符号表中的项目进行编辑时，用户可通过菜单工具栏中的「编辑」功能项或符号表的右键快捷选项来进行操作。而其中各种基本操作的相关说明请参考下表。

6

全局符号	
类型	符号名称
VAR	VAR_1
VAR	VAR_0

新增符号

符号筛选

清除符号表地址字段

上移 Alt+Up

下移 Alt+Down

导出全局符号

导入全局符号

撤销(U) Ctrl+Z

重复(R) Ctrl+Y

剪切(T) Ctrl+X

复制(C) Ctrl+C

粘贴(P) Ctrl+V

删除(D) Del

全选(A) Ctrl+A

OR

编辑(E)	视图(V)	编译(C)	F
还原(U)	Ctrl+Z		
取消还原(B)	Ctrl+Z		
剪切(T)	Ctrl+X		
复制(C)	Ctrl+C		
粘贴(P)	Ctrl+V		
删除(D)	Del		
全部选取(A)	Ctrl+A		
查找(F)	Ctrl+F		
替换(H)	Ctrl+H		

操作项目	功能说明
还原	还原方才的编辑动作 (*最多可复原 20 个步骤)
取消还原	取消还原而重复方才的动作
剪切	剪切选取的变量符号
复制	复制选取的变量符号
粘贴	将方才复制或剪切的变量符号贴在目前编辑的符号表中
删除	删除选取的变量符号
全部选取	选取符号表中所有的变量符号

- *1. 在不同机种或 POU 之间进行变量符号的 复制/剪切/粘贴 操作时，请务必确认粘贴后的变量符号属性是否正确。
- *2. AH/AS 系列机种与 DVP 系列机种与 DVPxxMC 与 AS5xx 机种，三者之间的变量符号都不可互相进行 复制/剪切/粘贴的操作。
- *3. 若粘贴的符号表中已有相同名称的符号时，系统会自动在粘贴的符号名称后方加上 CopyOf_n (n 为流水号) 做为新名称。

6.2.5 清除变量符号的配置地址


不论是用户自行指定或是系统自动配置的装置地址，执行此功能便可将符号表中的所有装置地址全部清除，而之后再次执行编译时，系统便会自动为其重新配置。另外，此功能仅会清除目前单击的符号表中的装置地址，而不会影响其它的符号表。

执行时请先在符号表中单击鼠标右键后，接着在快捷选项中单击「清除符号表地址字段」即可。



6.2.6 下载变量符号的初始值

每个变量符号在宣告时都可为其设置一个**初始值**，而在下载项目程序时便可选择是否将**初始值**一并写入该变量符号所对应的装置地址中。不过初始值的写入动作仅会在每次下载时才执行一次，而后续如主机断电或是 RUN/STOP 切换后，再次进入 RUN 模式的这个阶段，变量符号的初始值无法恢复原设置；因此对于须确保每次进入执行状态必须初始化的变量符号，建议还是于程序中加入初始值的初始化程序。执行在线更新的动作时，系统也不会重新下载变量符号的初始值。

欲执行时，请于菜单工具栏中单击 **PLC 程序传输 > 下载 (D)** 或单击图标工具栏的  图标，接着勾选传输选项的「**符号初始值 (下载时生效)**」后，按下「**传送**」即可将符号初始值写入 PLC 当中。



OR

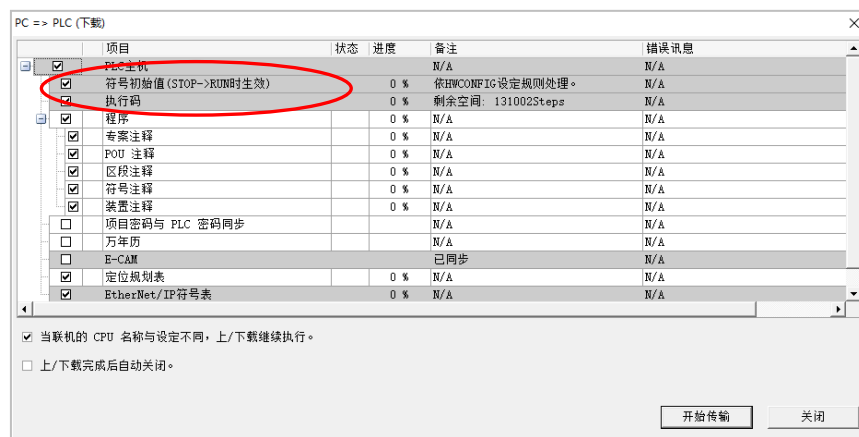


*.进行下载前请先确认 ISPSOft 已与主机正常进行联机。

部分机种的新初体支持新的初始值方式请参考下表，新的初始值方式提供 **Stop → Run 非停电保持符号初始值生效**、**Stop → Run 停电保持符号初始值生效**与**停电保持符号初始值生效**三种功能，使用 **Stop → Run 停电保持**与非**停电保持初始值生效**功能，请于下载项目程序时先参考第 3.1.4 节在 HWCONFIG 设定 **Stop → Run 停电保持**与非**停电保持初始值生效**的设定并下载至 PLC 主机。

机种	韧体版本	初始值表最大容量
AH5x1-RS2	1.05.00	80K Bytes
AH5x1-EN	2.03.00	80K Bytes
AH560-EN2	1.02.00	80K Bytes
AS	1.06.60	20K Bytes
ES3	0.40.10	20K Bytes
AHxxEMC	2.03	80K Bytes
AS 仿真器	1.06.60	20K Bytes

ISPSoft 也会自动检测连接装置是否支持新的初始值方式，若连接的为支持新的初始值机种，则在下载选单中「传输选项」的「符号初始值 (Stop → Run 时生效)」项目会变成无法取消勾选，如下图。



6

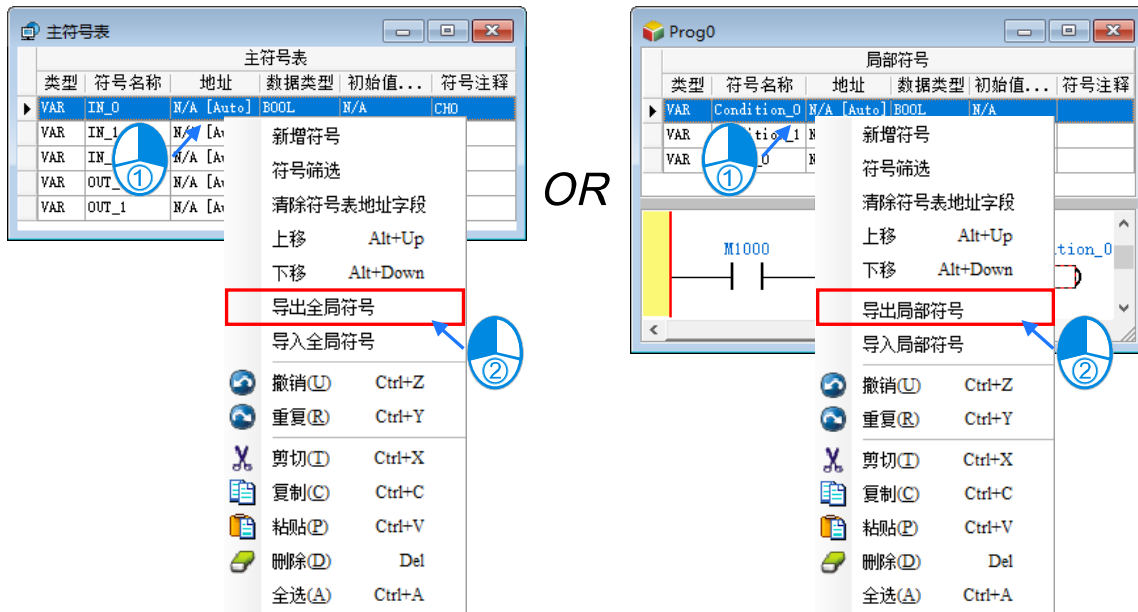
下载完成初始值表之后点选 **PLC 主机 (P)** > **初始值设定 (I)** > **停电保持符号初始值生效 (R)**，就可以使用单次停电保持符号初始值生效的动作，必须确认 ISPSoft 已可与主机正常进行联机，如下图。



6.2.7 符号表的导出

项目中所建立的变量符号亦可被导出为 CSV 文件，而之后用户便可直接通过 EXCEL 来开启，并对导出的数据进行后续的编辑与管理。

欲导出变量符号时，请直接在符号表中按下鼠标右键，接着在快捷选项中单击「导出全局符号」或「导出局部符号」即可。



6

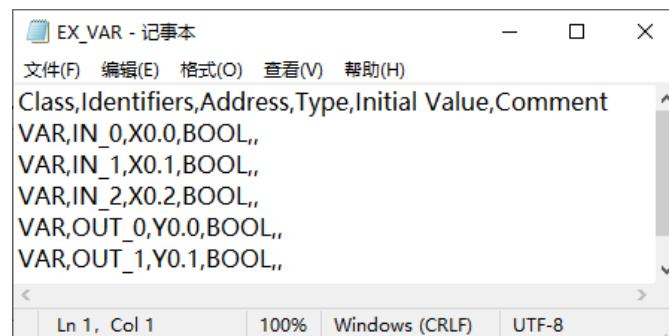
而除了通过右键的快捷选项之外，在导出全局符号表时，亦可由菜单工具栏单击**文件(F) > 导出 > 全局符号表(L)**或在项目管理区选取要导出的全局符号表单击右键导出，每次则只能针对目前选取的全局符号表进行操作。

6.2.8 符号表的导入

建立变量符号时，除了可在 ISPSOft 的符号表当中逐一进行宣告之外，用户亦可先将符号表导出，接着再通过 EXCEL 直接开启，并依据导出的格式进行编辑，而编辑完毕之后，只要再将其重新导入 ISPSOft 的符号表中即可，如此便可轻松地建立大量的变量符号。下列便是导出后的符号表格式。



类型	符号名称	地址	数据类型	初始值(下载时生效)	符号注释
VAR	IN_0	X0.0	BOOL	N/A	
VAR	IN_1	X0.1	BOOL	N/A	
VAR	IN_2	X0.2	BOOL	N/A	
VAR	OUT_0	Y0.0	BOOL	N/A	
VAR	OUT_1	Y0.1	BOOL	N/A	

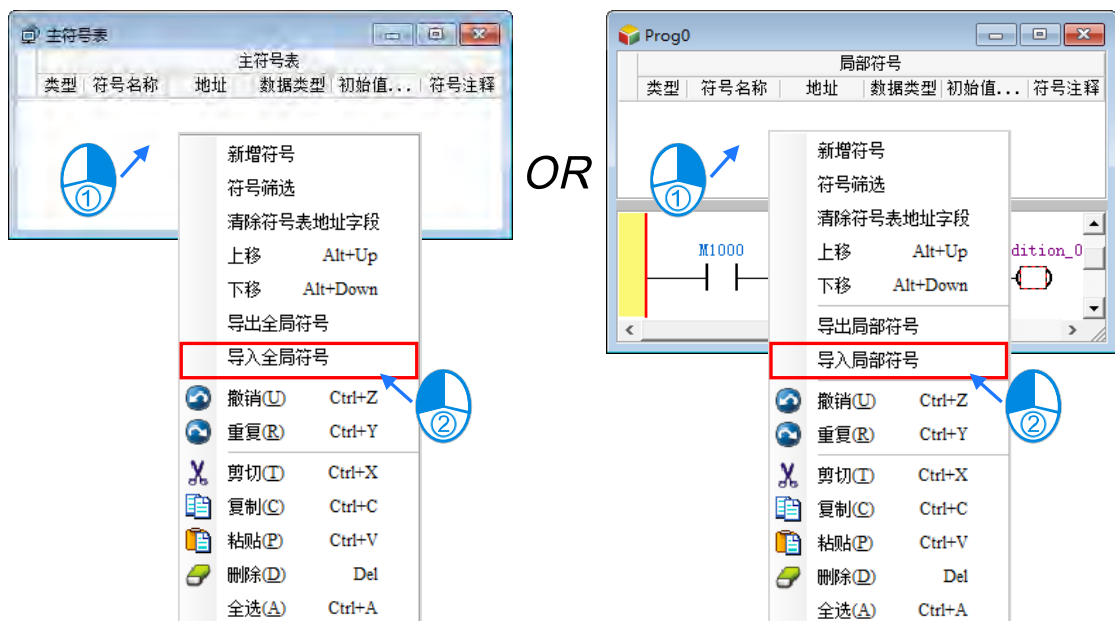



```

Class,Identifiers,Address,Type,Initial Value,Comment
VAR,IN_0,X0.0,BOOL,,
VAR,IN_1,X0.1,BOOL,,
VAR,IN_2,X0.2,BOOL,,
VAR,OUT_0,Y0.0,BOOL,,
VAR,OUT_1,Y0.1,BOOL,,
  
```

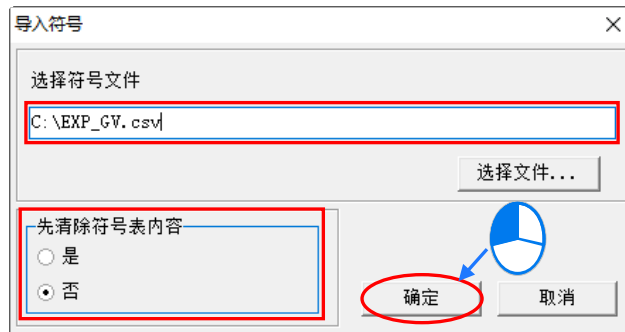
6

欲导入变量符号时，请直接于符号表中按下鼠标右键，接着于快捷选项中单击「导入全局符号」或「导入局部符号」的项目即可。



而除了通过右键的快捷选项之外，在导入全局符号表时，亦可由菜单工具栏单击**文件 (F) > 导入 > 全局符号表 (L)** 或在项目管理区选取要导出的全局符号表单击右键导入，每次则只能针对目前选取的全局符号表进行操作。

依据上述的方式执行之后，接着请于弹出的窗口中选择欲导入的文件，并在左下角的字段中设置在导入之前，是否要先清除该符号表中所有已存在的符号，完成后按下「**确定**」键即可。



补充说明

变量符号可在不同的机种或 POU 之间进行导出与导入，但目前 DVPxxMC 与 AS5xx 机种不支持与其它机种的变量符号做导出与导入动作，而若准备导入的文件中有存在以下状态的变量符号时，该变量符号将被自动排除或转换，而其它的变量符号则可正常被导入；当导入的过程中有出现异常时，于项目的路径下便会产生一个记录文件，其文件名为「**项目名称_ErrorLog**」，查看该文档便可得知导入过程中所发生的问题。

- 6
- (a) 当接受导入的符号表中已存在相同名称的变量符号时，文件中重复名称的变量符号将自动被排除，但若选择在导入前先清除符号表内容则不会有此问题。
 - (b) 当文件中的变量符号，其数据类型为被导入机种所不支持的类型时，该变量符号将被排除。
 - (c) 文件中的的变量符号有指定装置地址，但该装置为被导入机种不支持的装置种类，或装置地址超过被导入机种的装置寻址范围时，该变量符号将被排除。
 - (d) 在不同 POU 或全局符号之间进行导出导入的操作时，若欲导入的符号类型，如 VAR_INPUT、VAR_OUTPUT...，与被导入的符号表不兼容时，该项目将被排除。
 - (e) 将 AH/AS 系列机种的全局符号表导入 DVP 系列机种时，若变量符号的类型为 VAR_RETAIN，则导入后，所有 VAR_RETAIN 的类型将会被视为 VAR 类型。

下列为将 AHCPU530-EN 项目的全局符号表导入至 SV 机种项目的范例，其中符号 Var_A、Var_B、Var_C 皆可被导入，但符号 Var_C 的 VAR_RETAIN 类别将被自动视为 VAR 类别；而变量符号 Var_D 则因为 SV 机种不支持 L 装置而被排除；另外，变量符号 Var_E 则是因为 D65535 已超过 SV 机种的寻址范围，而变量符号 Var_F 则是因为 SV 机种不支持 STRING 类型的变量符号，因此也都会被自动排除。

类型	符号名称	地址	数据类型	初始值...	符号注释
VAR	Var_A	N/A [Auto]	WORD	N/A	
VAR	Var_B	MD	BOOL	N/A	
VAR	Var_C	N/A [Auto]	WORD	N/A	
VAR	Var_D	LD	WORD	N/A	
VAR	Var_E	D65536	WORD	N/A	
VAR	Var_F	N/A [Auto]	STRING(4)	N/A	

类型	符号名称	地址	数据类型	初始值...	符号注释
VAR	Var_A	N/A [Auto]	WORD	N/A	
VAR	Var_B	MD	BOOL	N/A	
VAR	Var_C	N/A [Auto]	WORD	N/A	

除了须注意符号表之间的数据格式之外，在符号批注的部份亦须注意下列事项。

下图是分别以 EXCEL 与记事本来开启的同一个文件，而由当中可发现，实际 CSV 文件的格式是以逗号来区隔每个字段，因此当我们在符号表的批注字段中使用逗号时，系统便会自动把逗号之后的内容视为其它字段，如下图红框中的 C2 与 C3，而重新导入时，逗号之后的内容便会因此而遗失。

	A	B	C	D	E	F	G	H
1	Class	Identifiers	Address	Type	Initial Value	Comment		
2	VAR	IN_0	X0.0	BOOL	FALSE	C1	C2	C3
3	VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3		

```

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Class, Identifiers, Address, Type, Initial Value, Comment
VAR, IN_0, X0.0, BOOL, FALSE, C1, C2, C3
VAR, IN_1, X0.1, BOOL, FALSE, C1 / C2 / C3
  
```

● 导出前

全局符号					
类型	符号名称	地址	数据类型	初始值	符号批注
VAR	IN_0	X0.0	BOOL	FALSE	C1, C2, C3
VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3

● 重新导入后

全局符号					
类型	符号名称	地址	数据类型	初始值	符号批注
VAR	IN_0	X0.0	BOOL	FALSE	C1
VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3

6.2.9 符号表的排序

不论是全局或区域符号表，当要调整符号的排序时，可以直接在符号表上用右键单击要移动的符号，再于快捷选项上单击**上移**或**下移**，移动指定的符号。



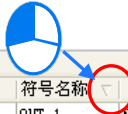
也可在符号表中单击欲调整的变量符号，之后利用键盘的【Alt】+【↑】或【Alt】+【↓】键调整该变量符号的上下位置。

在功能块的局部符号表中，VAR_INPUT、VAR_IN_OUT及VAR_OUTPUT等类型的变量符号，其排列顺序将会影响功能块端点的操作数顺序。

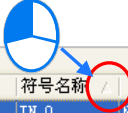
6



在全局符号表及程序 POU 的局部符号表中，在「符号名称」或「地址」的字段名称上单击鼠标左键后，符号表中的变量符号便会以该字段为参考值而重新进行排序；而此时该字段名称上也会出现排序方向的符号，再次单击字段名称则可改变排序的方向。而如同上述，**功能块局部符号**的排列将会影响功能块端点的操作数次序，因此**功能块的局部符号表**将无法使用此种方式进行排序。



局部符号						
类型	符号名称	地址	数据类型	初始值	符号批注	
VAR	OUT_1	N/A [Auto]	BOOL	N/A		
VAR	OUT_0	N/A [Auto]	BOOL	N/A		
VAR	IN_2	N/A [Auto]	BOOL	N/A		
VAR	IN_1	N/A [Auto]	BOOL	N/A		
▶ VAR	IN_0	N/A [Auto]	BOOL	N/A		



局部符号						
类型	符号名称	地址	数据类型	初始值	符号批注	
▶ VAR	IN_0	N/A [Auto]	BOOL	N/A		
VAR	IN_1	N/A [Auto]	BOOL	N/A		
VAR	IN_2	N/A [Auto]	BOOL	N/A		
VAR	OUT_0	N/A [Auto]	BOOL	N/A		
VAR	OUT_1	N/A [Auto]	BOOL	N/A		

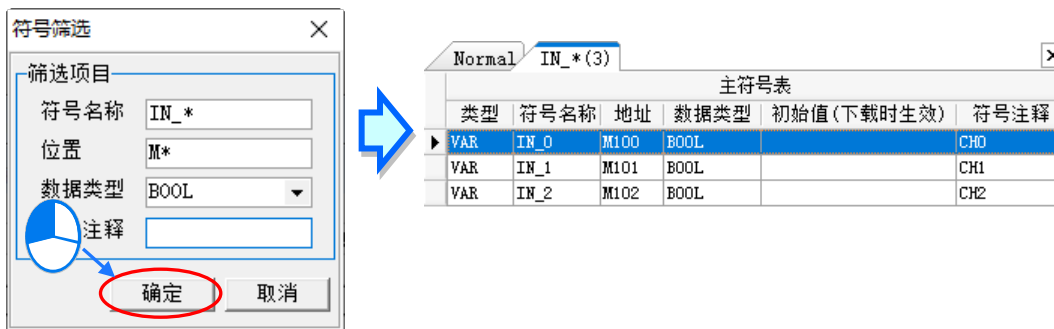
6.2.10 变量符号的筛选

此功能可用于查看具有共同条件的变量符号。

- (1) 符号表处按下鼠标右键，之后单击「符号筛选」。



- (2) 出现的窗口中设置筛选条件，而输入的内容亦可使用万用字符「*」来代表任何字符串，且当系统进行筛选时，符号表中的变量符号必须同时符合每个设置的条件才会被筛选出来，而若希望略过某个条件时，只要让该条件的字段保持空白即可。设置完成后按下「确定」键即可。



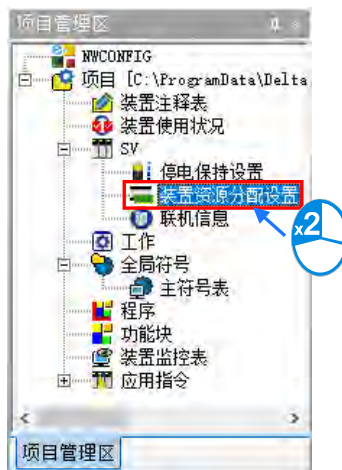
- (3) 被筛选出来的变量符号会显示于符号表中另外开启的显示分页中，单击符号表上方的选项卡即可进行切换，当筛选的组别过多时，利用页面右上方的 ◀ 与 ▶ 键亦可对选项卡进行切换，而单击页面右上方的 ✕ 键即可关闭该分页的显示。此外，这些另外开启的显示分页仅供查看之用，本身无法进行任何编辑，若要对符号表进行编辑时，请先切回原符号表，即选项卡名称为 **Normal** 的显示分页，接着再进行后续的操作。



6.2.11 装置资源分配设置

部分的 DVP 机种中，自动配置的地址都会使用原本的装置资源，而为了便于管理，用户可事先在项目中设置用来进行自动配置的装置范围，且不同的项目皆可有其专属的设置；当进行项目编译时，系统只会使用此范围内的装置来进行配置，但若变量符号的数量超过所设置的范围时，编译的结果便会产生装置资源不足的错误信息。关于装置地址配置的相关说明请参考本章第 6.1.4 节的内容。

设置时先展开项目管理区的机种项目，之后再于其下层的「装置资源分配设定」项目上双击鼠标左键。



接着便可于出现的窗口中设置各种装置的配置范围，且各种装置都有其各自的范围限制，例如 SV 机种的 D 装置只能从 D2000 开始，M 装置只能从 M2000 开始，C 装置只能配置一般用的 16 Bits 计数器，T 装置则只可设置时基为 100ms 的定时器，而当输入不允许的地址时，系统便会自动进行修正。完成后按下「确定」键即可，而此处的设置亦会随着项目一起被储存。

6



6.3 范例实作

在第 4 章的内容中，我们曾以一个简单的范例来示范如何建立一个梯形图程序，并在第 5.6.1 节中以 POU 与 TASK 的程序架构重新规划；而在接下来的内容中，我们将延续于 5.6.1 节中所建立的范例，并在程序中以变量符号来替换原本的装置名称。

6.3.1 变量符号规划

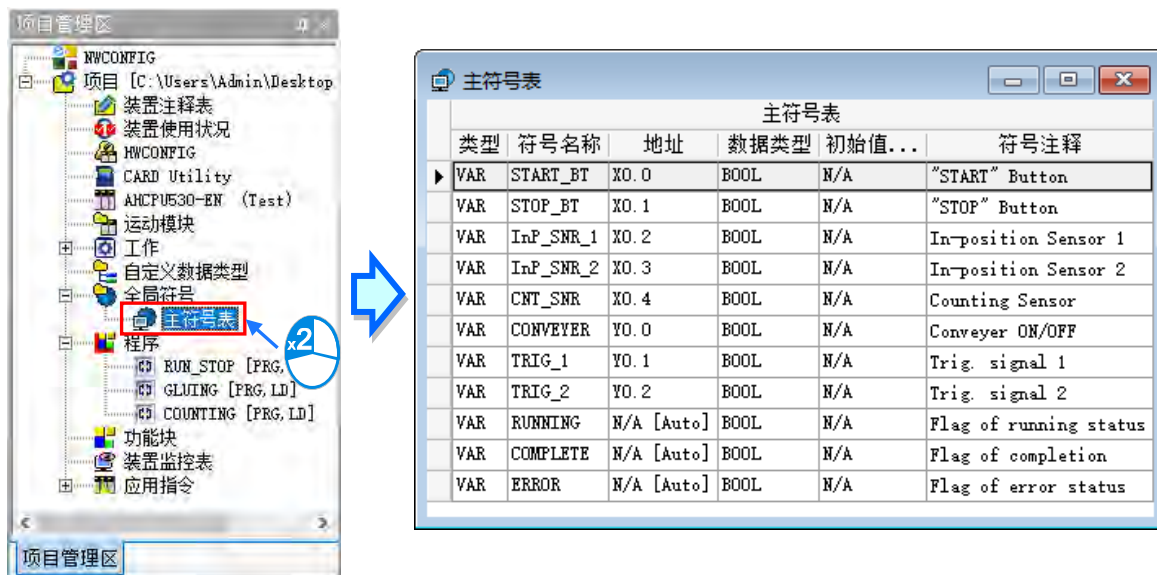
下表为重新规划的装置符号对照表，而其规划的原则与建议请参考下述。

原装置名称	符号名称	指定地址	宣告位置	功能说明
X0.0	START_BT	X0.0	全局符号表	启动按钮
X0.1	STOP_BT	X0.1	全局符号表	停止按钮
X0.2	InP_SNR_1	X0.2	全局符号表	到位传感器 1
X0.3	InP_SNR_2	X0.3	全局符号表	到位传感器 2
X0.4	CNT_SNR	X0.4	全局符号表	计数传感器
Y0.0	CONVEYER	Y0.0	全局符号表	输送带
Y0.1	TRIG_1	Y0.1	全局符号表	注射器 1 触发信号
Y0.2	TRIG_2	Y0.2	全局符号表	注射器 2 触发信号
M0	RUNNING	自动配置	全局符号表	运转标志
M1	COMPLETE	自动配置	全局符号表	完成标志
M2	ERROR	自动配置	全局符号表	错误标志
D0	CNT_DT	自动配置	局部符号表	入料计数值

- X 与 Y 装置为与 PLC 外部连接的接点，且其装置地址与设备的接线有关，所以必须自行指定正确的装置地址；此外，虽然部分的 X 与 Y 装置仅会于某个 POU 当中被使用，但仍建议将其建立于全局符号表中，这对于后续的维护会较为便利，例如变更 IO 接线，或是要于后续新增的程序中存取这些接点时；不过这并非必然的，用户仍可依据自己的习惯来进行规划。
- 原本的 M0、M1 与 M2 这 3 个标志，因为会被不同的 POU 进行存取，所以将这三个变量符号宣告于全局符号表中，且在操作上无须刻意指定装置地址，因此将其规划为自动配置。
- 原本的 D0 寄存器，其功能仅做为计数值的寄存之用，且只会用于「COUNTING」的 POU 当中，因此在规划上，其对应的变量符号仅须直接宣告在「COUNTING」POU 的局部符号表当中，且其装置地址亦只要规划为自动配置即可。

6.3.2 建立范例程序

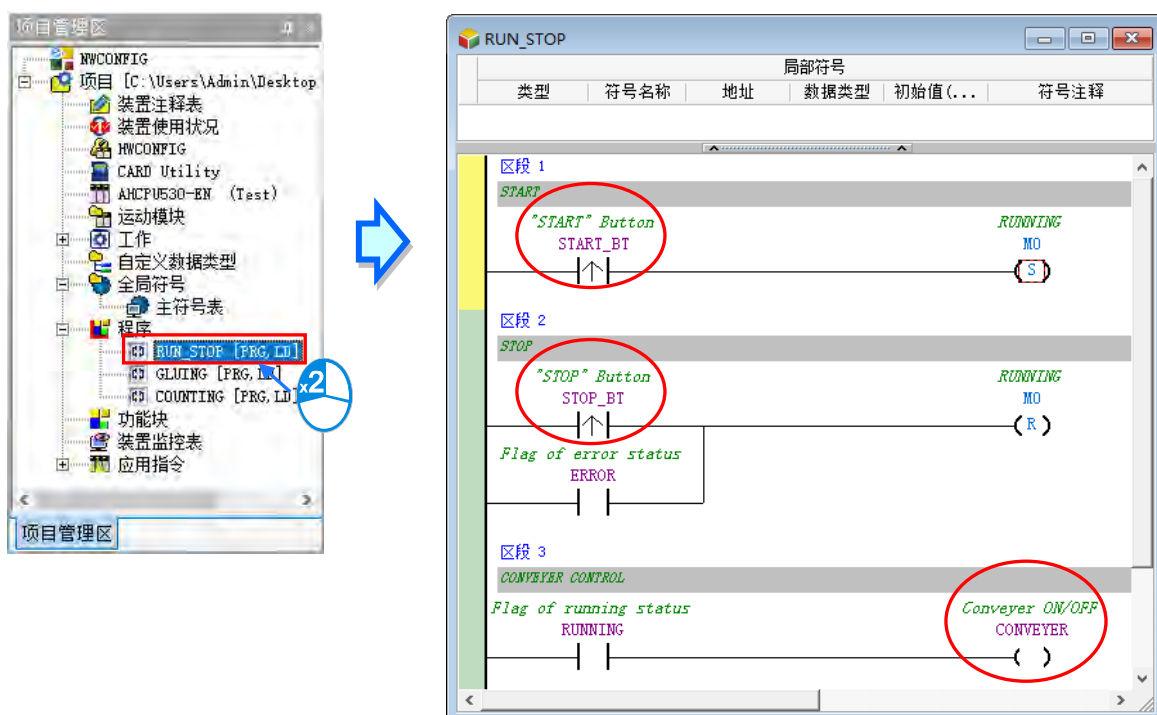
接下来可直接开启第 5.6.1 节当中的范例来进行修改或是重新建立程序。若要重新建立程序时，请先新增一个项目，之后参考第 4 章与第 5.6.1 节的相关介绍，并配合接下来的说明逐一完成此节的范例。首先请开启全局符号表，并参考本章第 6.2 节的内容于全局符号表中建立如下的变量符号。



主符号表					
类型	符号名称	地址	数据类型	初始值...	符号注释
VAR	START_BT	X0.0	BOOL	N/A	"START" Button
VAR	STOP_BT	X0.1	BOOL	N/A	"STOP" Button
VAR	InP_SNR_1	X0.2	BOOL	N/A	In-position Sensor 1
VAR	InP_SNR_2	X0.3	BOOL	N/A	In-position Sensor 2
VAR	CNT_SNR	X0.4	BOOL	N/A	Counting Sensor
VAR	CONVEYER	Y0.0	BOOL	N/A	Conveyer ON/OFF
VAR	TRIG_1	Y0.1	BOOL	N/A	Trig. signal 1
VAR	TRIG_2	Y0.2	BOOL	N/A	Trig. signal 2
VAR	RUNNING	N/A [Auto]	BOOL	N/A	Flag of running status
VAR	COMPLETE	N/A [Auto]	BOOL	N/A	Flag of completion
VAR	ERROR	N/A [Auto]	BOOL	N/A	Flag of error status

完成全局符号表的建立后，请开启「RUN_STOP」这个程序 POU，而若是直接开启第 5.6.1 节中的范例来进行修改的用户将可发现，自行指定地址的装置都已变更为对应的符号名称，且原本的装置批注也已显示为符号批注；若未有如下图的显示时，请参考后续的内容进行显示的切换。

6





局部符号

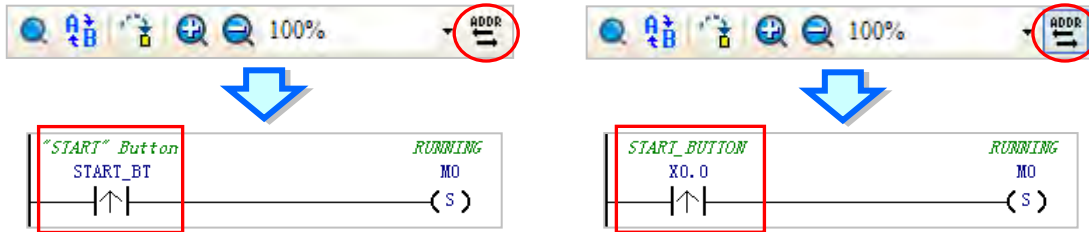
类型	符号名称	地址	数据类型	初始值(...)	符号注释
	"START" Button	START_BT			
	"STOP" Button	STOP_BT			
	Flag of error status	ERROR			
	Flag of running status	RUNNING			
	Conveyer ON/OFF	CONVEYER			

图 1 展示了 RUN_STOP 程序的梯形图逻辑。图中包含三个逻辑段：

- 图段 1 (START): 包含常开触点 START_BT 驱动线圈 RUNNING MO (S)。
- 图段 2 (STOP): 包含常开触点 STOP_BT 驱动线圈 RUNNING MO (R)。
- 图段 3 (CONVEYER CONTROL): 包含常开触点 RUNNING 驱动线圈 CONVEYER ()。

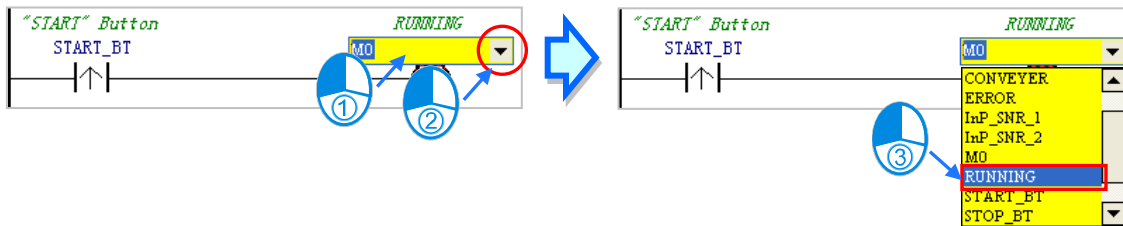
 图中还显示了 ERROR 符号的常开触点。所有符号名称和地址均与全局符号表一致。

对于有指定装置地址的变量符号，于 LD 的环境中，用户可自行选择该变量符号的呈现方式；当图标工具栏的  图标为按下的状态时，程序画面便会以指定的装置地址与装置批注来显示这些变量符号；而若  图标为未按下的状态时，程序画面便会以符号名称与符号批注来显示。



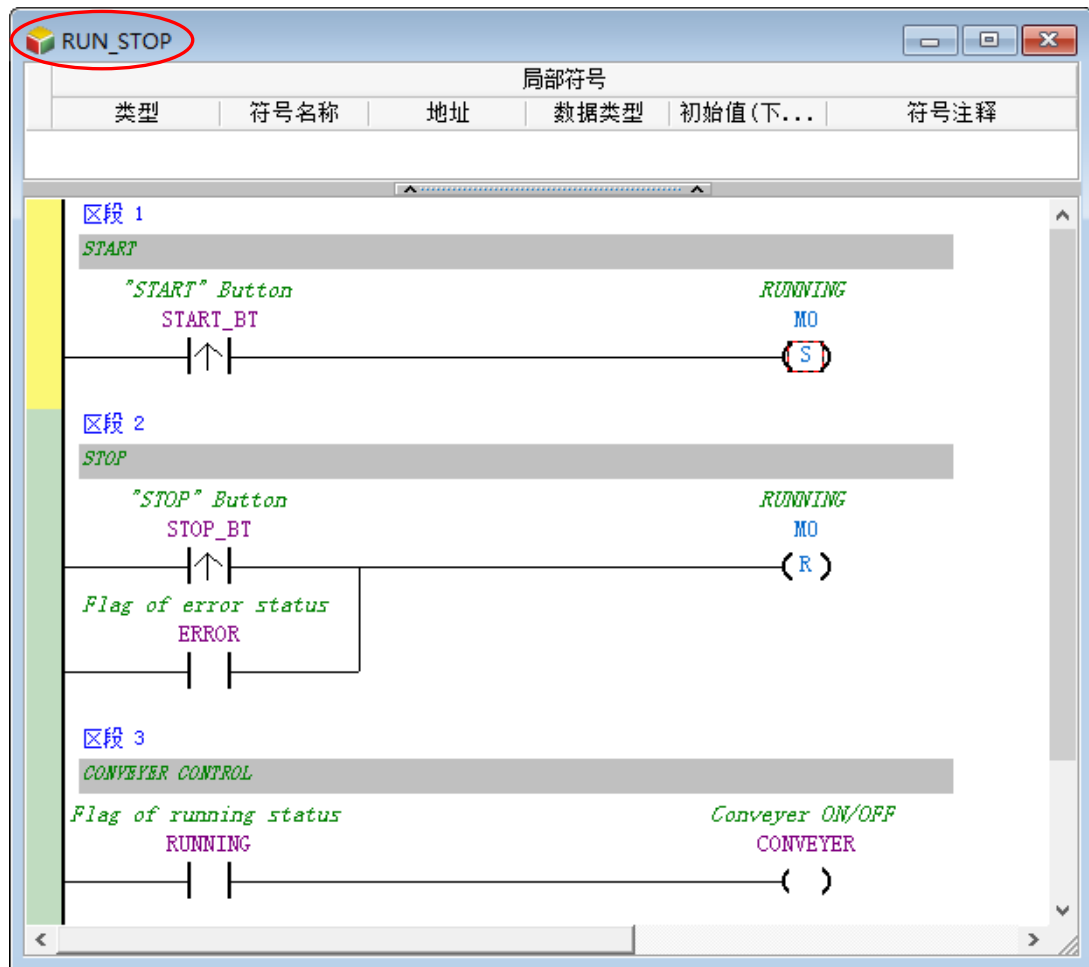
*.上图的 M0 因并未配置给任何的变量符号，所以切换时仍会一直保持原状；而同样的，若变量符号未配置装置地址，或是由系统自动配置至内部存储器时，该变量符号的显示也会保持原状。

接着可将程序中的运转标志 M0 变更为变量符号「RUNNING」，而变更的方式除了依据第 4 章的说明直接输入之外，先清除装置输入字段后单击鼠标左键，之后自动出现符合该字段类型变量符号的下拉选项供使用者挑选。而若是使用键盘时，亦可在该字段进入编辑状态时，配合【↑】与【↓】键便可进行选取，完成后按下【Enter】键即可。

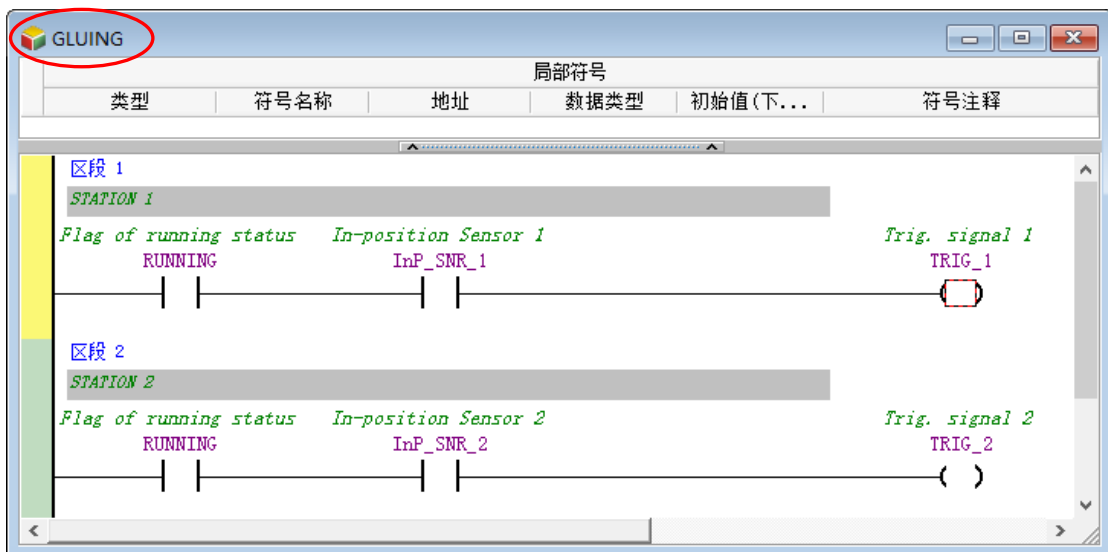


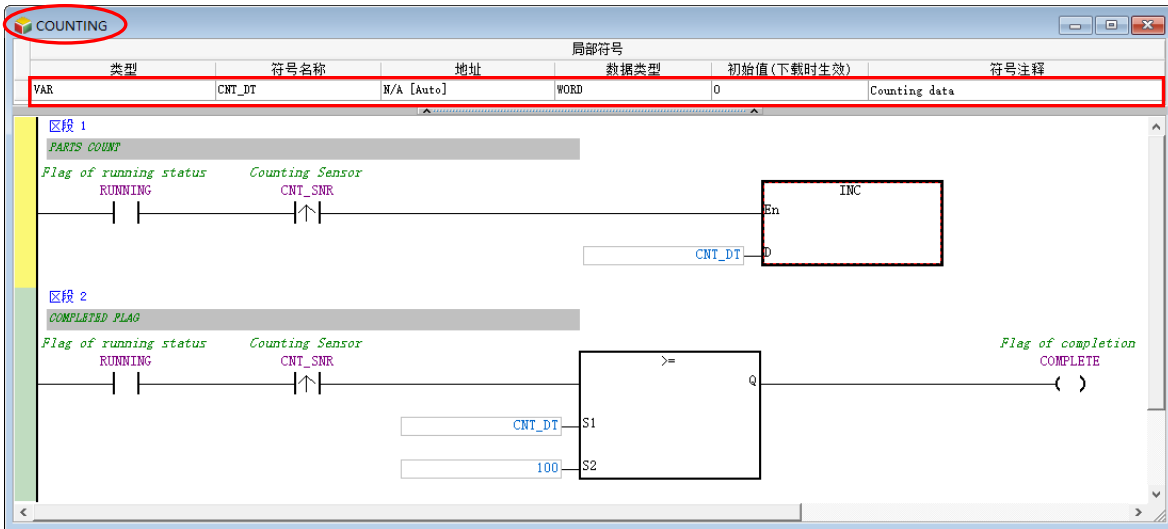
6

接下来请参考之前的说明逐一完成如下的程序 POU。



6





若是重新建立程序的用户，请依据第 4 章与第 5.6.1 节的相关说明完成硬件组态、硬件参数设置、工作 (TASK) 管理... 等操作；而完成范例程序后，请自行编译项目程序，并将硬件参数与程序下载至主机当中进行测试；而下载程序时，请一并勾选「符号初始值」以便将「COUNTING」POU 中的局部符号「CNT_DT」的初始值一并写入主机中。

6





第7章 功能块与函数库

目录

7.1 认识功能块 (Function Block)	7-2
7.1.1 功能块简介	7-2
7.1.2 功能块的特性与优点	7-3
7.2 ISPSOft 的功能块 (Function Block) 架构	7-4
7.2.1 功能块的使能 (En) 端点	7-4
7.2.2 功能块的变量符号	7-5
7.2.3 功能块的输出与输入	7-6
7.2.4 指针类型的变量符号	7-12
7.2.5 功能块的定义与实例	7-15
7.2.6 功能块调用功能块的嵌套结构	7-18
7.2.7 功能块的内存配置	7-20
7.3 功能块的使用	7-27
7.3.1 功能块的基本规格	7-27
7.3.2 功能块专用的脉冲型指令 (限定 AH/AS 机种)	7-28
7.3.3 功能块的程序监控	7-29
7.3.4 功能块的修改	7-30
7.4 范例实作	7-31
7.4.1 程序规划	7-31
7.4.2 建立范例程序	7-31
7.4.3 函数 (Function) 范例程序	7-41
7.5 认识函数库 (Library)	7-42
7.5.1 新增用户自定义函数库	7-42
7.5.2 引用用户自定义函数库	7-44
7.5.3 引用台达函数库	7-46

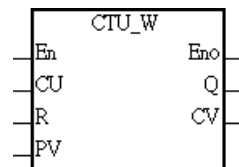
7.1 认识功能块 (Function Block)

功能块 (**Function Block** · 简称 **FB**) 与函数 (**Function** · 简称 **FC**) 是在 PLC 编程中相当重要的一个环节 · 同时也是 IEC61131-3 所规范的重点项目之一 · 而因其本身所具有的特性及优点 · 所以在 ISPSOft 中同时也针对功能块的特性提供了相当多的支持与功能。在 ISPSOft 中 · 提供了用户自行定义的 FB POU 与系统定义的 FB POU ; 目前只有 DVPxxMC 与 AS5xx 机种支持用户自行定义的 FC POU · 且提供了 API 应用指令做为系统的 FC 供用户调用。

7.1.1 功能块简介

功能块是一种具有运算功能的程序组件 · 本身也属于 POU 的一种 · 但它无法自行运作 · 而是必须透过程序 POU 对它进行调用 · 并传递相关参数之后 · 才能执行功能块所定义的功能 ; 而功能块在执行完毕后 · 也同样可将内部的运算结果再回传给上层 POU (调用者) 所指定的装置或变量符号中。

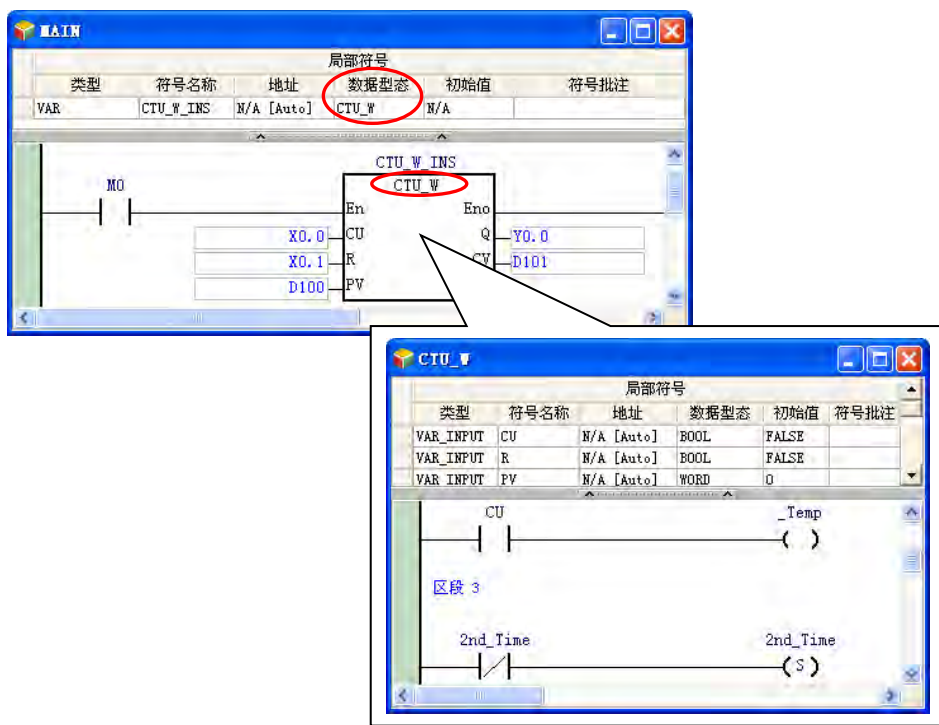
右图即为功能块的图标 · 外观与 API 应用指令相当类似 · 于左右两侧皆有输入与输出的端点 · 而使用的方法也十分相近 · 较大的差异在于上层 POU (调用者) 在调用功能块时 · 用户必须为其宣告一个功能块类型的变量符号 · 而且在功能块内部也允许再对其他的功能块进行调用。



功能块是含记忆功能的单位 · 每一个功能块都会被配置一组实体的内存区块 · 用来存放内部变量符号的状态值 · 且因功能块内部的变量符号状态会在执行结束后被保留下来 · 所以每次功能块的执行结果都将受到上次执行状态的影响 · 因此实际上有可能在同样的输入条件下 · 而得到不同的输出结果。

下图便是个一功能块的应用范例 · 其中可看到上层 POU (调用方) 的局部符号表中有一个功能块类型的变量符号 · 且该数据类型与功能块的名称相同。关于功能块的使用将于接下来的章节中陆续介绍。

7



7.1.2 功能块的特性与优点

功能块之所以被 IEC 61131-3 纳入标准，并且受到众多用户的喜爱，主要是因为它具有许多传统 PLC 编程方式所无法达到的特性与优点。

- **模块化设计**

将一个庞大的程序分成许多不同的子程序，并将各个子程序以功能块的型式加以建立，使用时只要在程序 POU 中加以调用并排列组合即可。

- **高独立性**

不同功能块可依据及其本身的功能特性或开发人员的习惯而使用不同的编程语言来建立，如 LD、IL、ST、FBD...等；而在不同编程语言的 POU 中调用功能块时，也完全无需考虑该功能块是以何种语言所建立的；例如在 LD 的程序中，若遇到一段须进行复杂运算的程序时，用户便可将这段程序以 ST 语言建立一个功能块，之后再于原来的 LD 程序中进行调用即可。

- **可重用性**

功能块一旦建立之后，只要符合使用规则及限制，原则上可一再的被重复使用。

- **高可移植性**

功能块开发完成后，不只是在原来的项目中才可使用，利用导出/导入的功能，在别的项目程序中亦可使用这些开发好的功能块，因此用户可逐渐地建立属于自己的功能块库。

- **维护便利**

因功能块已将其所属的程序封装成一个独立的模块，因此当发现功能块内部的程序有错误或是所定义的功能与实际需求不符时，用户仅须集中焦点在修改功能块内部所定义的内容即可，而无须在整个庞大的程序当中进行除错或修改。

- **增加程序的可读性**

用户可将较为复杂或重复性高的程序封装成一个功能块，接着再于原本的程序当中调用使用，如此便可精简原本程序的架构，并提高其可读性。

- **高保密性**

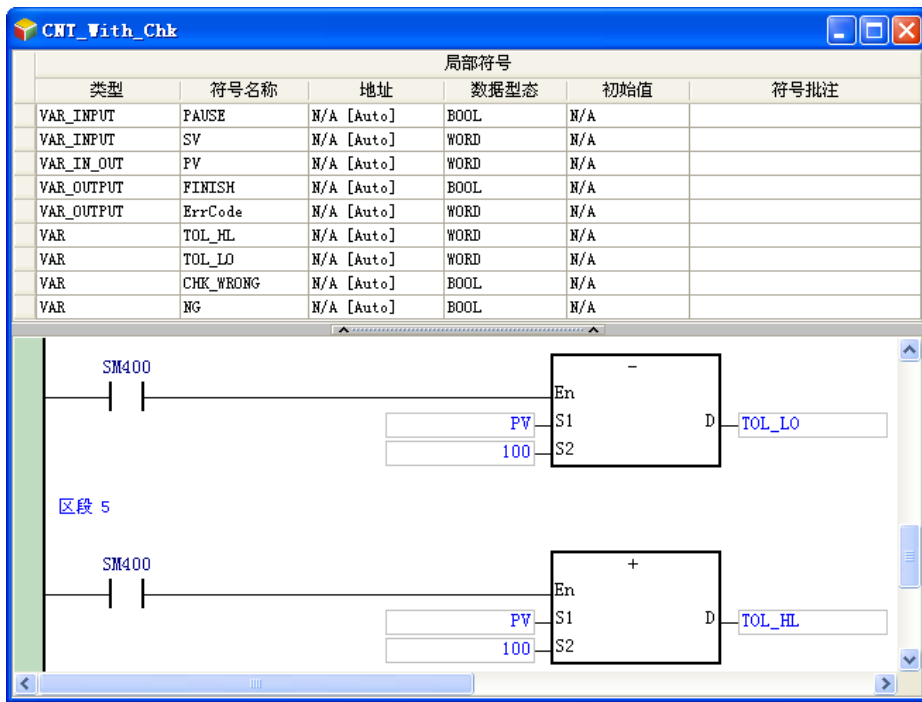
所建立的功能块可单独设置密码，用户可放心的将自己的核心技术建立在功能块中并提供给他人使用；除此之外，系统商或独立程序开发者亦可在不公开原始码的情况下，只提供特定功能的功能块给有需求的客户。

- **高开发效率**

依据上述所提及的种种特点，以项目开发的角度来看，功能块的应用可以提供一个高效率的开发环境，且因功能块的模块化特性，更可让不同的开发人员甚至外包厂商共同参与整个项目的开发，对于人力资源的利用，亦可提供相当大的弹性。

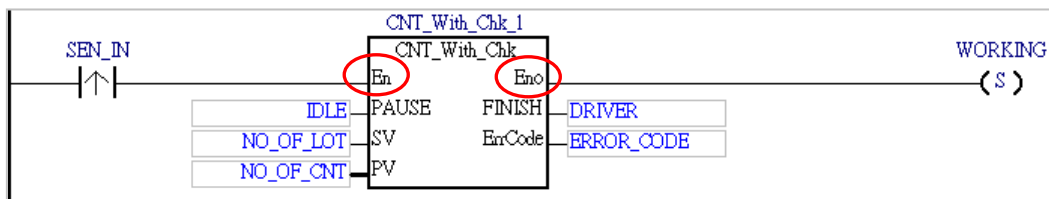
7.2 ISPSOft 的功能块 (Function Block) 架构

请参考下图，功能块及函数内部的结构与一般的程序 POU 相似，分为局部符号表与程序主体，且在编辑的方式上也与一般的程序 POU 几乎相同。在 ISPSOft 中可用来建立功能块的编程语言分别有 LD、IL、ST 及 FBD，但不包含 SFC，而在这些语言的程序当中也同样可以调用功能块，其中包括在 SFC 的 Action 中同样可以对功能块进行调用。此外，功能块内部的程序也允许再对其他的功能块进行调用，而调用功能块时，完全无需考虑该功能块的程序是以何种语言所编写的。



7.2.1 功能块的使能 (En) 端点

上层 POU (调用方) 当中调用功能块及函数的方式与执行 API 应用指令的方式类似，同样以输入 En 端点的逻辑状态来决定该功能块是否被执行；当连结至 En 端点的状态为 ON 时，功能块才会被执行，反之当连结至 En 端点的状态为 OFF 时，该功能块的动作就会被略过；至于功能块的 Eno 端点，其功能则仅是把 En 端点的逻辑状态再串接输出给后方的区块。此外，当于 ST 语言中使用功能块时则不会带入 En 接点的状态，详细说明将于第 13 章的内容当中另行介绍。



7.2.2 功能块的变量符号

与一般的程序 POU 相同，功能块的内部可以建立属于自己的局部符号表，而且也可以在功能块当中使用全局符号，只是在功能块内部使用全局符号时，便可能失去其可移植性，因为当我们利用导出与导入的方式将功能块复制到另一个项目当中使用时，该项目不见得会有相同的全局符号。

下列为与功能块相关的变量符号说明摘要，各机种在功能块使用变量符号类别的支持状态，如下表。

- 功能块的变量符号类型

类别	说明	AH/AS	DVP	AS5xx DVPxxMC
VAR	仅供功能块内部运算用，且在运算结束后，变量符号的状态会被保留下来。	✓	✓	✓
VAR_INPUT	用做功能块的输入端点。可接收上层 POU（调用方）所指定的操作数，并在执行功能块时将该操作数的内容值复制进功能块。	✓	✓	✓
VAR_OUTPUT	用做功能块的输出端点。可在功能块执行完毕后，将运算的结果透过此类变量符号输出至上层 POU（调用方）所指定的操作数。	✓	✓	✓
VAR_IN_OUT	功能块的反馈型端点。执行功能块时，先透过此端点将上层 POU（调用方）所指定的操作数内容传进功能块，待功能块执行完毕后，再将运算的结果反馈输出至上层 POU（调用方）所指定的操作数。	✓	✓	
VAR_RETAIN	RETAIN 类型的变量具有断电保持的属性，断电保持的变量最多可以定义 15999 个。			✓
VAR_STATIC	系统会给该变量分配一个固定的地址，使得该变量的值在控制器运行过程于相同功能块的不同 Instance 之间能够共享变量。			✓
VAR_STATIC_RETAIN	变量为 STATIC 类型且具有断电保持属性。			✓

- 功能块内部局部符号所支持的数据类型

机种	说明
AH/AS	BOOL、WORD、DWORD、LWORD (*注 2)、INT、DINT、LINT (*注 2)、REAL、LREAL (*注 2)、ARRAY、STRING、POINTER (*注 1)、T_POINTER (*注 1)、C_POINTER (*注 1)、HC_POINTER (*注 1)、功能块类型、用户自定义类型
DVP	BOOL、WORD、DWORD、LWORD、REAL、ARRAY、STEP、TIMER、COUNTER
AS5xx/ DVPxxMC	BOOL、BYTE、WORD、DWORD、LWORD、SINT、INT、DINT、LINT、USINT、UINT、UDINT、ULINT、REAL、LREAL、ARRAY、STRING、TIME、DATE、TOD、DT、功能块类型、用户自定义类型

*1. POINTER、T_POINTER、C_POINTER 与 HC_POINTER 等指针类型的相关说明将于第 7.2.4 节当中另行介绍。

*2. AS 系列机种在使用功能块内部区域符号不支持 LWORD、LINT 及 LREAL。

● 功能块的内存配置

机种	说明
AH/AS	宣告为功能块类型的变量符号 (功能块实例) 与建立于功能块内部的局部符号, 宣告时皆无法自行指定装置地址, 系统将会自动为其配置一个主机内部预留的内存区块。
DVP	每个宣告为功能块类型的变量符号 (功能块实例) 都会占用一个 P 装置; 而建立于功能块内部的局部符号, 配置时则会视其数据类型, 自主机目前可用的装置资源中, 为每个功能块实例的局部符号配置一个对应的装置地址。
AS5xx/ DVPxxMC	宣告为功能块类型的变量符号 (功能块实例) 与建立于功能块内部的区域符号, 宣告时皆无法自行指定装置地址, 系统将会自动为其配置一个主机内部预留的内存区块。

*.关于功能块实例与内存配置的详细说明将于第 7.2.5 节~第 7.2.7 节的内容当中另行介绍。

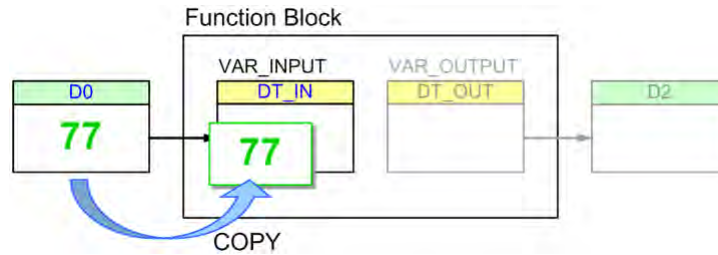
7.2.3 功能块的输出与输入

一般功能块的输出与输入都是采用传值的方式, 即不论是调用方将某个装置指定给功能块的输入端, 或是功能块的运算结果经由输出端点回传给调用方的装置, 彼此只是把本身的值复制给对方而已, 单个功能块支持多个输入与输出, 而函数仅有一个输出参数也就是其返回值。

在此以下列的图例来加以说明。

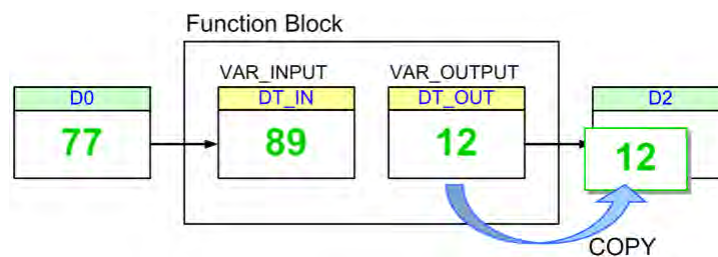
● 功能块执行前

调用方将装置 D0 指定给功能块的输入端点 DT_IN (VAR_INPUT 类型), 因此当功能块被调用时, 系统便会将装置 D0 的当前值复制给功能块的变量符号 DT_IN。

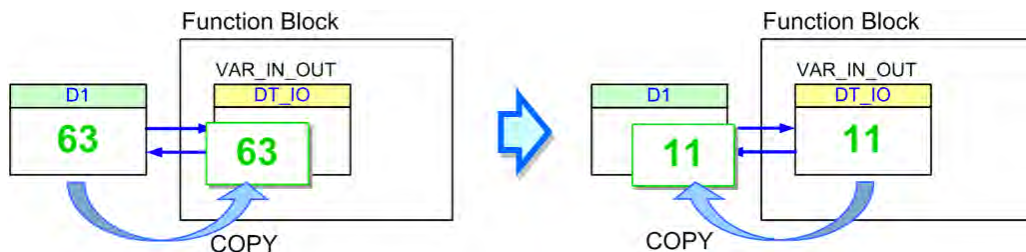


● 功能块执行后

运算结束后，变量符号 DT_IN 的内容值变为 89，不过此改变并不会影响调用方的装置 D0；此外，功能块的 DT_OUT (VAR_OUT 类型) 会将最后的运算结果复制给调用方指定的装置 D2，而之后即使 D2 的内容在程序执行的过程中被改写了，这个结果依旧不会影响功能块内部的 DT_OUT。

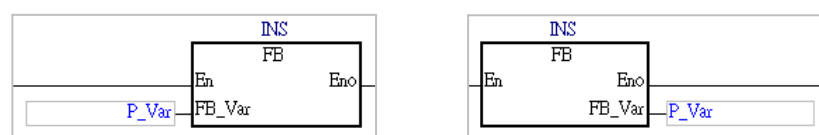


至于变量符号类型为 VAR_IN_OUT 的情况，虽然此类型的变量符号会在功能块执行结束后，立即的更新调用方指定给该端点的装置内容，但本质上还是先将调用方装置的当前值复制进来后，再将最后的结果复制回去给调用方装置。



*.在 AH/AS 系列的机种中，功能块的 VAR_IN_OUT 类型端点可宣告为指针类型的变量符号，而有别于上述的传值方式，透过指针类型的变量符号，调用方可以把外部装置的地址指针传进功能块，详细说明请参考下一节的内容。

接着请参考下列两个功能块的图例，其中 FB_Var 为功能块的端点变量符号，而变量符号 P_Var 则为上层 POU (调用方) 指定给功能块端点的操作数。基本上 P_VAR 与 FB_VAR 两者之间的数据类型必须符合下述的基本原则，但为了编程上的便利，ISPSOft 尚提供几个特例原则，只要使用上仍符合这几个特例原则中的其中一个，即使不符合基本原则，系统仍会将其视为合法的使用方式，而 DVPxxMC 与 AS5xx 机种则遵守其数据类型的原则。



7

● 一般数据类型的基本原则

调用方的变量符号 (P_VAR) 与功能块端点 (FB_VAR) 的数据类型须完全一致。

● 一般数据类型的特例原则 (参考下表)

1. 当有任何一方为 WORD 系列的数据类型时，另一方便允许是 WORD、INT、REAL 任一系列的数据类型，而满足此条件时，P_VAR 的数据长度便允许大于或等于 FB_VAR。
2. 若 P_VAR (调用方) 的数据类型为 STEP，则 FB_VAR (功能块) 的数据类型可为 BOOL，此操作代表调用方的步进点状态将会在功能块的内部使用。

适用机种		AS/AH/DVP								
数据类型		FB_Var (功能块)								
		WORD	DWORD	LWORD	INT	DINT	LINT	REAL	LREAL	BOOL
P_VAR (调用方)	WORD	✓			✓					
	DWORD	✓	✓		✓	✓		✓		
	LWORD	✓	✓	✓	✓	✓	✓	✓	✓	
	INT	✓			✓					
	DINT	✓	✓			✓				
	LINT	✓	✓	✓			✓			
	REAL	✓	✓					✓		
	LREAL	✓	✓	✓					✓	
	BOOL									✓
	STEP									✓

除了上述单纯数据类型的原则之外，针对 ARRAY 型式的变量符号，其使用原则说明如下。

● ARRAY 操作的基本原则

1. 两方 ARRAY 元素的数据类型必须相同。
2. 数据类型相同时，P_VAR (调用方) 的 ARRAY 长度允许大于 FB_VAR (功能块) 的长度。

● ARRAY 操作的特例原则

1. 比较 ARRAY 的数据类型时，前述一般数据类型的特例原则亦可适用。
2. 当一方为 ARRAY 型式，而另一方为非 ARRAY 型式的变量符号时，系统会将非 ARRAY 型式的变量符号视为数组长度为 1 的 ARRAY。
3. 当有一方的 ARRAY 类型为 WORD 系列时，另一方可为 WORD、INT、REAL 任一系列，且两方在比较数据长度时，可以采用加总的结果来进行判断。例如 DWORD 相容于长度为 2 的 INT 阵列，而长度为 4 的 WORD 阵列亦相容于一个 LINT 的变数符号；不过长度为 4 的 INT 阵列与一个 LINT 的变数符号并不相容，因两方并没有 WORD 类型的数据格式，因此不符合此原则。

下列则为上述几项使用原则的说明范例。

✓ EX1 : P_VAR (调用方) 为 INT，FB_VAR (功能块) 为 INT。

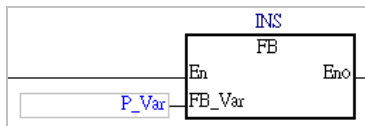
➔ 符合一般数据类型的基本原则，毫无疑问的是合法使用。



- ✓ EX2 : P_VAR (调用方) 为 WORD · FB_VAR (功能块) 为 INT 。
 - ➔ 符合一般数据类型的特例原则 1 ，因此是合法使用。
- ✓ EX3 : P_VAR (调用方) 为 LWORD · FB_VAR (功能块) 为 DINT 。
 - ➔ 符合一般数据类型的特例原则 1 ，因此是合法使用。
- ✗ EX4 : P_VAR (调用方) 为 WORD · FB_VAR (功能块) 为 LINT 。
 - ➔ 虽然有一方为 WORD 类型 ，但 P_VAR 的数据长度小于 FB_VAR ，因此是不合法的。
- ✗ EX5 : P_VAR (调用方) 为 LINT · FB_VAR (功能块) 为 DINT 。
 - ➔ 两方数据类型不一致 ，因此是不合法的。
- ✓ EX6 : P_VAR (调用方) 为长度 4 的 INT 数组 · FB_VAR (功能块) 为长度 3 的 INT 数组 。
 - ➔ 符合 ARRAY 操作的基本原则 ，因此是合法使用。
- ✓ EX7 : P_VAR 为长度 2 的 WORD 数组 · FB_VAR 则为一般的 WORD 类型 (非数组格式) 。
 - ➔ 符合 ARRAY 操作的特例原则 2 ，因此是合法使用。
- ✓ EX8 : P_VAR 为长度 4 的 WORD 数组 · FB_VAR 则为一般的 DWORD 类型 (非数组格式) 。
 - ➔ 符合 ARRAY 操作的特例原则 3 ，因此是合法使用。
- ✗ EX9 : P_VAR 为长度 4 的 DINT 数组 · FB_VAR 为长度 2 的 INT 数组 。
 - ➔ 两方数据类型不一致 ，因此是不合法的。
- ✓ EX10 : P_VAR 为一般的 DWORD 类型 (非数组格式) · FB_VAR 则为长度 2 的 WORD 数组 。
 - ➔ 符合 ARRAY 操作的特例原则 3 ，因此是合法使用。

● DVPxxMC/AS5xx 数据类型的原则

举下图为例



1. 此处一般情况调用方的变数符号 (P_VAR) 与功能块端点 (FB_VAR) 的数据类型须完全一致 ，但在 ISPSOft 也允许部分情况 P_VAR (调用方) 的数据类型长度允许小于或等于 FB_VAR (功能块) 的数据类型长度 ，此详细说明请参考下表。
2. 若 FB_VAR (功能块) 的变数数据类型为 ARRAY ，则 P_VAR (调用方) 数据类型必须为 ARRAY ，数据类型内容及长度大小需与 FB_VAR (功能块) 相同。

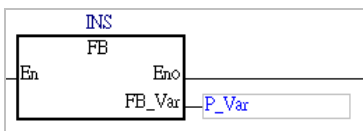
适用机种		DVPxxMC/AS5xx														
数据类型		FB_Var (功能块)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (调用方)	BOOL	✓														
	BYTE		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	WORD			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
	DWORD				✓	✓			✓	✓			✓	✓	✓	✓
	LWORD					✓				✓				✓	✓	✓
	USINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓



适用機種		DVPxxMC/AS5xx														
数据类型		FB_Var (功能块)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
数据类型	UINT			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
	UDINT				✓	✓			✓	✓			✓	✓	✓	✓
	ULINT					✓				✓			✓	✓	✓	✓
	SINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INT			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
	DINT				✓	✓			✓	✓			✓	✓	✓	✓
	LINT					✓				✓			✓	✓	✓	✓
	REAL														✓	✓
	LREAL														✓	✓

适用機種		DVPxxMC/AS5xx				
数据类型		FB_Var (功能块)				
		TIME	DATE	TOD	DT	STRING
P_VAR (调用方)	TIME	✓				
	DATE		✓			
	TOD			✓		
	DT				✓	
	STRING					✓

再举下图为例



1. 此处一般情况调用方的变数符号 (P_VAR) 与功能块端点 (FB_VAR) 的数据类型须完全一致，但在 ISPSOft 也允许部分情况 P_VAR (调用方) 的数据类型长度允许大于或等于 FB_VAR (功能块) 的数据类型长度，此详细说明请参考下表。
2. 若 FB_VAR (功能块) 的变数数据类型为 ARRAY，则 P_VAR (调用方) 数据类型必须为 ARRAY，数据类型内容及长度大小需与 FB_VAR (功能块) 相同。

适用機種		DVPxxMC/AS5xx														
数据类型		FB_Var (功能块)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (调用方)	BOOL	✓														
	BYTE		✓				✓				✓					
	WORD		✓	✓			✓	✓			✓	✓				

适用机种		DVPxxMC/AS5xx														
数据类型		FB_Var (功能块)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (调用方)	DWORD		✓	✓	✓		✓	✓	✓		✓	✓	✓			
	LWORD		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
	USINT		✓				✓				✓					
	UINT		✓	✓			✓	✓			✓	✓				
	UDINT		✓	✓	✓		✓	✓	✓		✓	✓	✓			
	ULINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	SINT		✓				✓				✓					
	INT		✓	✓			✓	✓			✓	✓				
	DINT		✓	✓	✓		✓	✓	✓		✓	✓	✓			
	LINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	REAL		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	LREAL		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

适用机种		DVPxxMC/AS5xx				
数据类型		FB_Var (功能块)				
		TIME	DATE	TOD	DT	STRING
P_VAR (调用方)	TIME	✓				
	DATE		✓			
	TOD			✓		
	DT				✓	
	STRING					✓

补充说明

(a) 在符合上述原则的情况下，当 P_VAR (调用方) 的数据长度大于 FB_VAR (功能块) 时，若是将 P_VAR 输入 FB_VAR 的场合，系统便会由 P_VAR 的低位开始，只取符合 FB_VAR 长度的数据；若是由 FB_VAR 输出给 P_VAR 的场合，系统则会由低位开始，只改写 P_VAR 中符合 FB_VAR 长度的数据，超过长度的部份则会保持原状。使用时请小心数据的正确性。

(b) 在 AH500 的功能块变量符号中尚有指针类型的特殊用法，详细说明请参考第 7.2.4 节的内容。

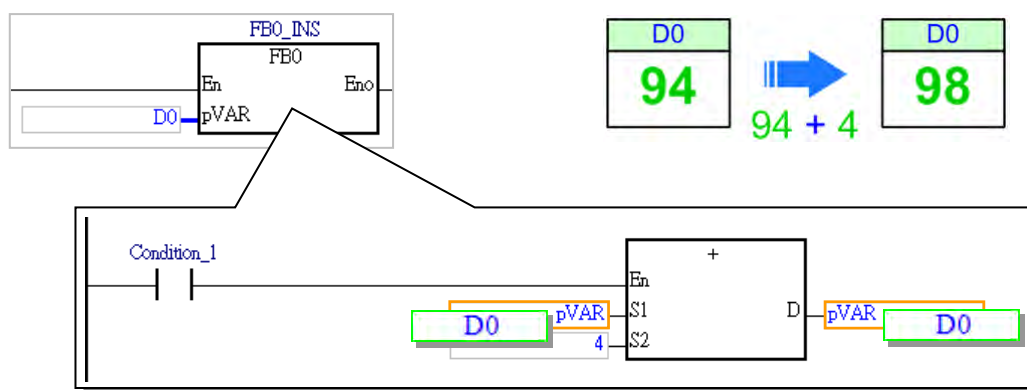
(c) 在 AS5xx/DVPxxMC 机种，当 P_VAR (调用方) 的数据长度小于 FB_VAR (功能块) 时，若是将 P_VAR 输入 FB_VAR 的场合，系统便会由 P_VAR 的低位元开始，只取符合 P_VAR 长度的数据输入给 FB_VAR；若是由 FB_VAR 输出给 P_VAR 的场合，系统则会由低位元开始，只改写 P_VAR 中符合 FB_VAR 长度的数据，超过长度的高位元部份则会被写成 0。使用时请小心数据的正确性。

7

7.2.4 指针类型的变量符号

指针类型的变量符号主要用于将调用方指定的变量符号或装置指针传进功能块内部，其使用方式上与一般的数据类型有相当大的不同。一般数据类型的变量符号会使用复制的方式来传递两方目前的内容值；但若功能块端点被宣告为指针类型时，传进功能块的数据则会是指定给该端点的变量符号或装置本身的指针，因此在功能块内部进行处理的便会被指定的对象本身。

如下图范例，调用方将装置 D0 指定给功能块端点「pVAR」，而「pVAR」在功能块内部则被宣告为指针类型的变量符号，因此当功能块被调用并开始执行时，功能块内部便会把「pVAR」视为装置 D0 的本身，而不是单纯的将 D0 的内容值复制进来而已。



关于指针类型的变量符号，其使用原则如下。

1. 只适用于 AH/AS 以及 DVP 的 ES3 機種系列的机种。
2. 只可在功能块内部宣告，且必须宣告为 VAR_IN_OUT 类型的功能块端点。
3. 各种指针类型的变量符号及使用限制请参考下表。

7

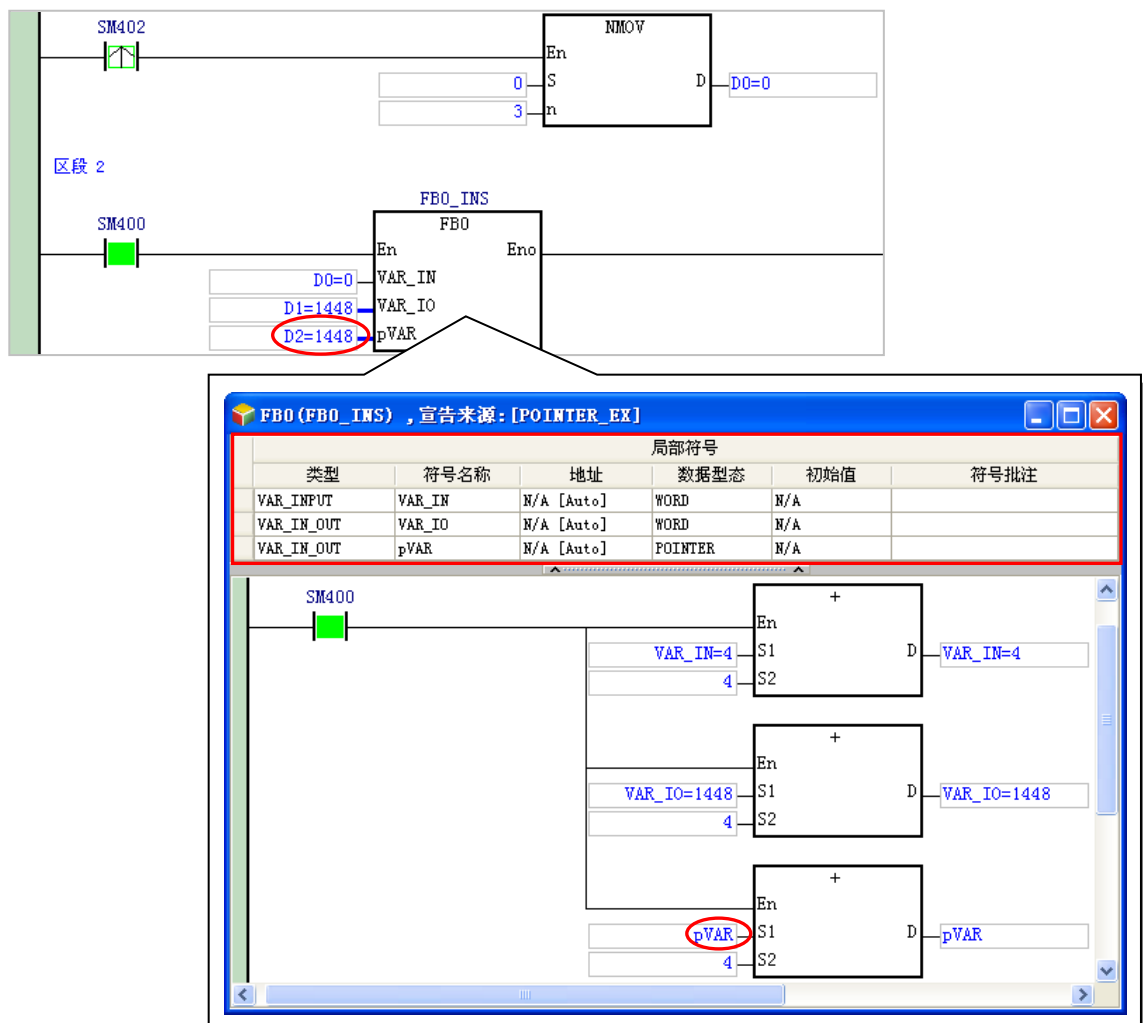
指针类型	使用限制	
一般指针 POINTER	数量限制	每个功能块定义中，最多可宣告 16 个
	可传指针	WORD/DWORD/LWORD/INT/DINT/LINT 等类型的变量符号或 D 装置、L 装置或 X、Y 等字装置 (如 X0、Y0)。
定时器专用指针 T_POINTER	数量限制	每个功能块定义中，最多可宣告 8 个
	可传指标	T 装置或 TIMER 类型的变量符号
计数器专用指针 C_POINTER	数量限制	每个功能块定义中，最多可宣告 8 个
	可传指针	C 装置或 COUNTER 类型的变量符号
高速计数器专用指针 HC_POINTER	数量限制	每个功能块定义中，最多可宣告 8 个
	可传指针	HC 装置或指定配置 HC 装置的 COUNTER 类型符号

● POINTER 类型符号的使用范例

功能块 FB0 分别有三个输入端点，其数据类型与类型请参考下图红框部份的局部符号表，而调用方的 POU 则会分别把 D0、D1 与 D2（初始值皆为 0）指定给这三个端点。下述则为程序执行的状况。

- D0 指定给功能块端点 VAR_IN，且因该符号为 VAR_INPUT 类型的 WORD 类型，因此每次调用功能块时都会先将装置 D0 的当前值复制给 VAR_IN，而经过功能块内部的运算之后，VAR_IN 的值会被加上 4，但这个结果并不会影响调用方指定的装置 D0，因此 D0 都会固定保持为初始值 0。
- D1 指定给功能块端点 VAR_IO，且因该符号为 VAR_IN_OUT 类型的 WORD 类型，因此每次调用功能块时都会先将装置 D1 的当前值复制给 VAR_IO，而经运算之后，VAR_IO 的值会被加上 4，且基于 VAR_IN_OUT 类型的特性，功能块执行完毕后将 VAR_IO 的值再度复制回去给装置 D1，因此每次 D1 传进 VAR_IO 的值都是累加后的结果，所以 D1 的内容会在执行的过程中持续累加。
- D2 指定给指针类型的功能块端点 pVAR，而基于指针型符号的特性，当功能块在执行时会直接将现在 D2 的值加上 4 之后，再直接把结果写回 D2，因此在执行过程中，D2 的内容会持续累加。

虽然 D1 与 D2 的执行结果相同，但实际执行的过程却有所差异。此外，因 pVAR 的数据内容是指定装置 D2 的指针，因此在进行在线监控时并不会如同变量符号 VAR_IO 一般显示目前的数值。



*.SM400 为 AH/AS 机种的特殊继电器，其特性为主机通电后会保持为 ON，而 SM402 则是启动正向（RUN 的瞬间为“ON”）

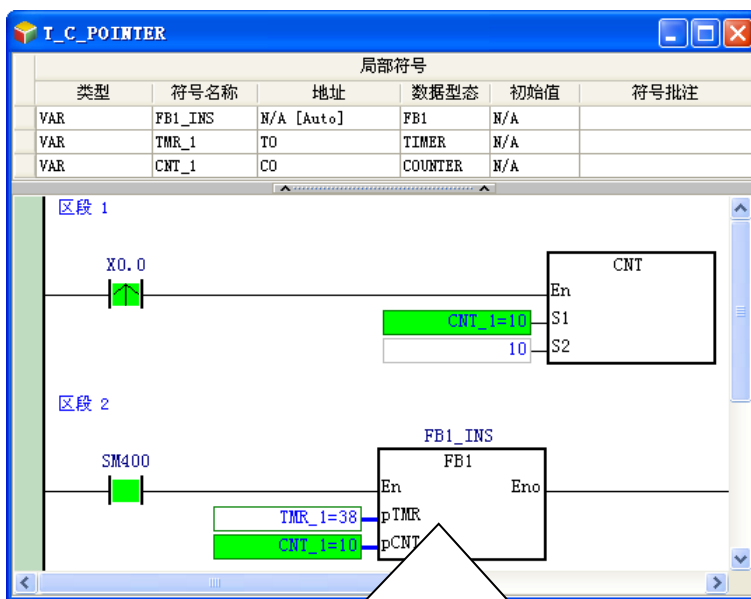
● T_POINTER 与 C_POINTER 类型符号的使用范例

因 AH/AS 系列主机的功能块内部无法宣告 TIMER 与 COUNTER 类型的变量符号，因此我们可利用 T_POINTER 与 C_POINTER 类型的变量符号将计时/计数器的装置指针传进功能块内部进行运作。

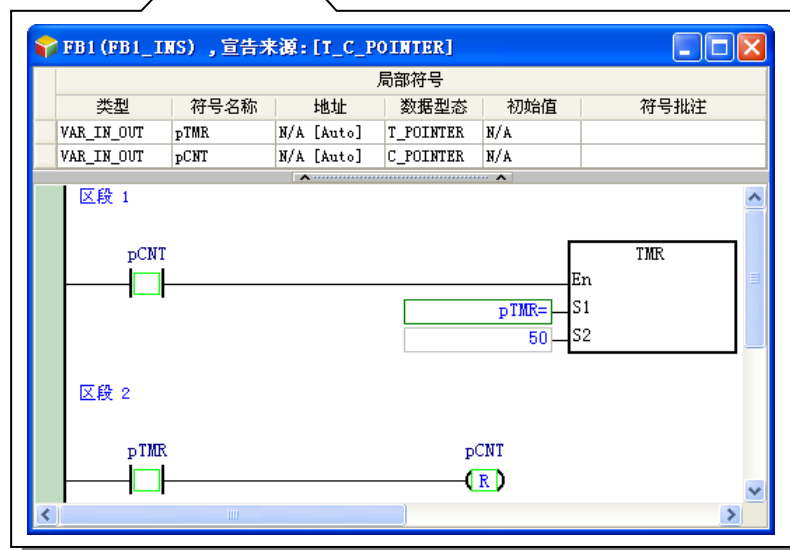
功能块 FB1 共有 2 个输入端点，分别是 T_POINTER 类型的 pTMR 及 C_POINTER 类型的 pCNT；而调用方则分别把变量符号 TMR_1 (TIMER 类型) 与 CNT_1 (COUNTER 类型) 指定给两端点。

在调用方的程序当中，当 X0.0 由 OFF→ON 时 CNT_1 的计数值便会加 1，一直到 CNT_1 的计数值到达 10 之后，CNT_1 的接点状态便会为 ON；而功能块内部因接收了 TMR_1 与 CNT_1 的指标，因此执行时会直接判断 CNT_1 (pCNT) 的接点状态，当 CNT_1 (pCNT) 为 ON 时便会启动 TMR_1 (pTMR) 的计时；当计时到达 5 秒后，CNT_1 (pCNT) 的状态便会被重置。

同样的，指针型符号于在线监控时不会显示对应装置的当前值。

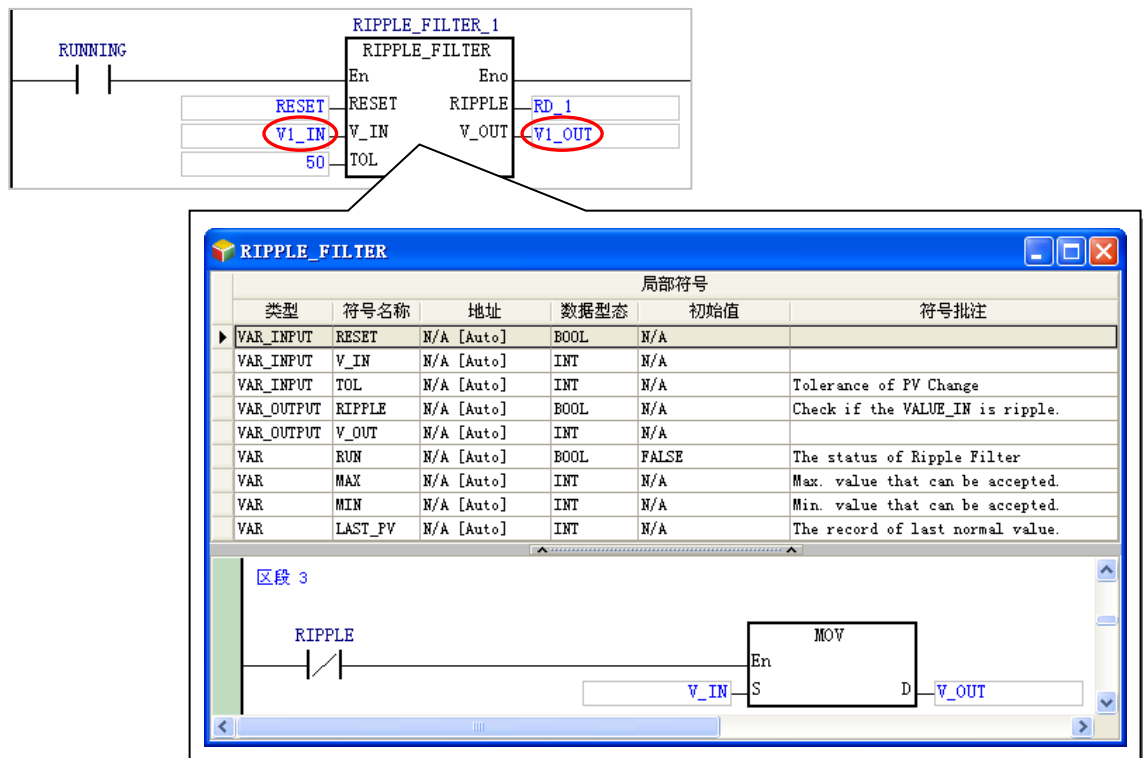


7

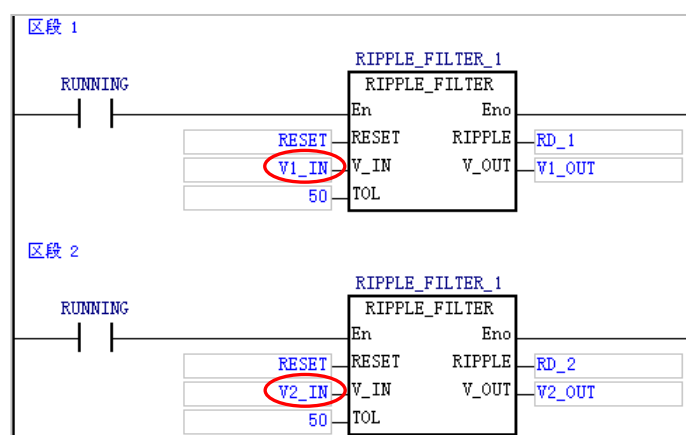


7.2.5 功能块的定义与实例

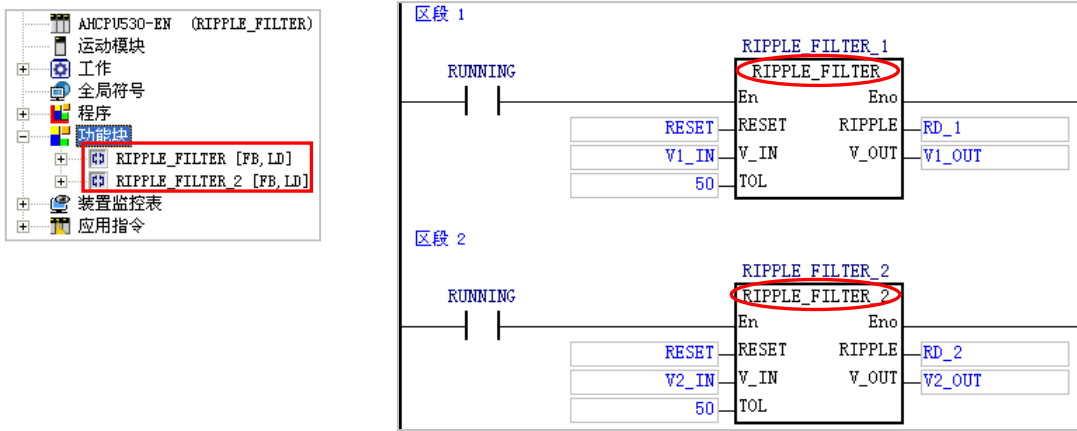
功能块的定义与实例是一个相当重要的概念，请参考下列的范例。上层 POU (调用方) 会在每次执行时将变量符号 **V1_IN** 的值输入给功能块 **RIPPLE_FILTER**，接着功能块内部便会将此次输入的值与上次记录的值进行比较后，再将最后的运算结果输出给上层 POU 的 **V1_OUT**；而为了记录每次运算的结果，功能块内部便必须宣告可以储存运算结果的局部符号，因此进行编译时，系统便必须为这些局部符号配置一个实体的内存区块。



接下来考虑一个应用状况，若上层 POU 有好几个数据都须要进行相同的处理程序，而各个数据之间亦必须彼此独立，则若是如下图的程序，其执行的结果便会是错误的，因每次 **V2_IN** 输入功能块时，在功能块内部进行比较的基准却是上次运算 **V1_IN** 的结果，所以两者的数据将会彼此干扰。



而若是重新建立一个内容完全相同的功能块 POU，虽然编译时会为两个功能块 POU 的局部符号配置不同的内存区块，而执行的结果也会是正确的，然而在编程的程序上便会产生许多重复的工作。



有鉴于此，在 IEC61131-3 中便导入了功能块实例的观念。

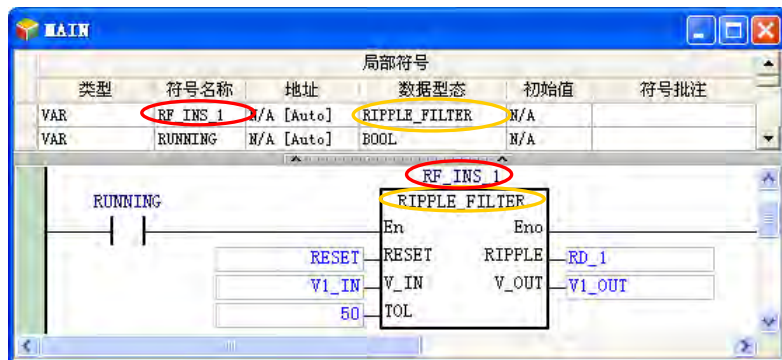
当我们在项目管理区中新增了一个功能块 POU，并在此 POU 当中进行局部符号的宣告以及程序的编辑，而完成之后所产生的成品便称之为「功能块定义」，但其意义仅如同一份文档而已，本身并不会参与实际的运算，同时也不会占用 PLC 运作的任何资源。



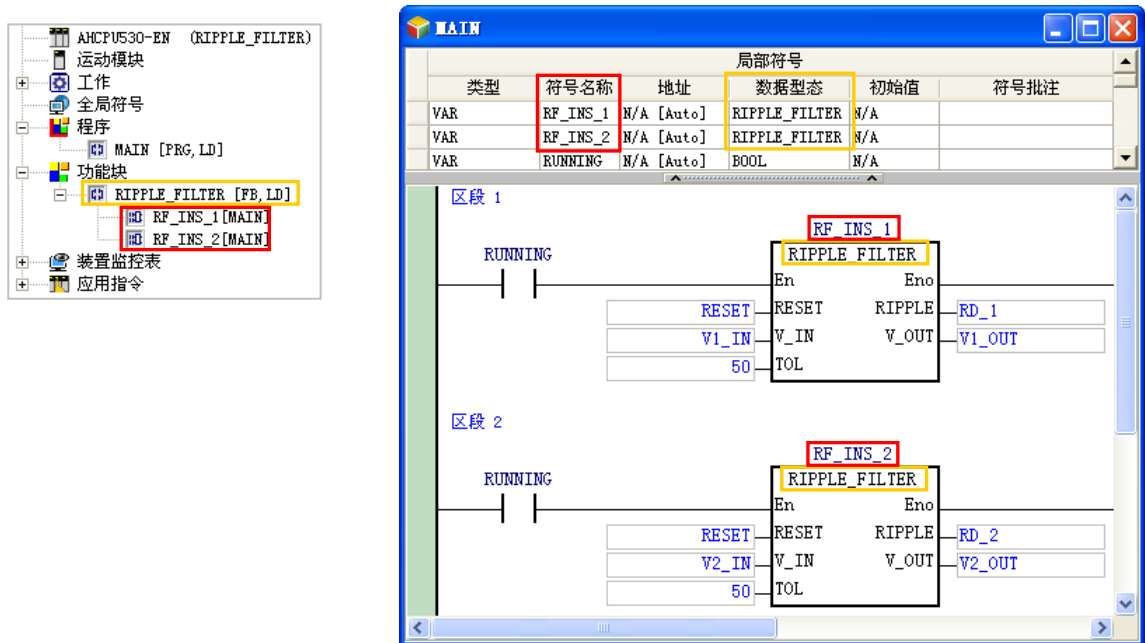
当要在某个程序 POU 当中调用该功能块时，用户便必须先宣告该功能块类型的变量符号，此时便代表要依据「功能块定义」来制造一个会参与实际运作的对象，其所产出的成品便称之为「功能块实例」，而其中功能块类型的符号名称即是「功能块实例」的名称。

当进行编译时，系统便会依据定义而为功能块实例与其内部的局部符号配置一个实体的内存区块。

7



再重新考虑之前的情况，当运作上必须要有相同的运算程序，但又需要彼此独立的内存区块，操作上便可利用功能块实例的特性，只要在调用功能块时，为需要不同内存区块的功能块，另外宣告一个不同名称的功能块符号即可，系统便会将不同的功能块符号名称视为不同的功能块实例，而在编译时便会为其配置独立的内存区块。



如下图，项目程序在编译前仅会有功能块定义的项目，而编译后，该定义的下层便会产生实际参与程序运作的功能块实例，其显示格式为 **实例名称[调用方 POU 名称]**。



与一般的变量符号相同，只要在符号表中宣告了一个功能块类型的变量符号，之后即使在编辑的程序当中没有使用到，编译时仍会为其产生对应的功能块实例；此外，在 AH5X0 系列机种中，被程序 POU 所调用的「功能块实例」数量合计不得超过 1024 个。（未产生实例的功能块定义则可不计），但其他机种系列则无此限制，直到内存被配置完毕为止。

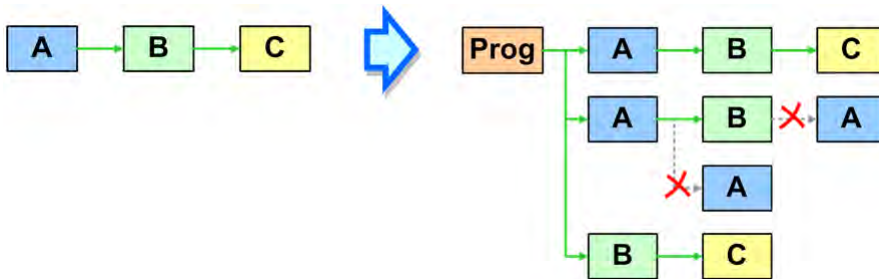
综合上述，功能块的操作包含了「定义」与「实例」两种观念。「定义」属于规划的层面，而「实例」则偏向执行的层面；以产品为例，「功能块定义」相当于工程人员针对开发产品所订定的产品规格，而「功能块实例」则相当于依据此规格所生产出来的产品。

7.2.6 功能块调用功能块的嵌套结构

在 ISPSOft 的功能块规格中，功能块可以调用功能块，在 AH 系列机种最多可允许调用至 32 层，在 AS/DVP 系列机种最多可允许调用至 16 层，而 DVPxxMC 与 AS5xx 机种则没有此限制，而在认定上，由程序 POU 所调用的功能块实例会被定义为第一层。



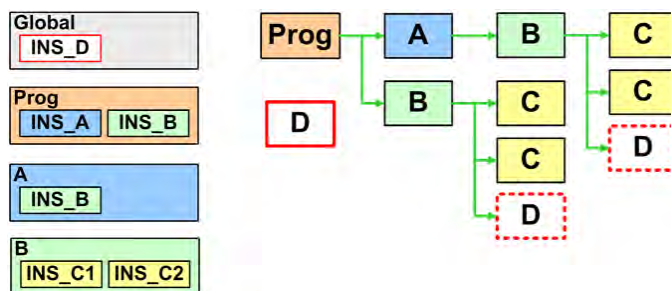
如下图的结构，在功能块 A 的程序中调用了功能块 B，而在功能块 B 的程序中也调用了功能块 C，此时功能块 A 便是功能块 B 与 C 的上层功能块，而功能块 B 本身也是 C 的上层功能块；且因 ISPSOft 不支持自己调用自己的结构，所以一旦功能块 A 是功能块 B 的上层，则在功能块 B 以下的所有下层就不允许再调用功能块 A，但所有下层功能块都可以在最上层的程序 POU 当中被调用。



当功能块 A、B、C 皆未被任何程序 POU 所调用时，即使功能块 B 被 A 所调用，而 C 也被 B 所调用，但因三者皆不会参与程序的实际运作，所以编译时都不会产生**功能块实例**；直到在某个程序 POU 中或在全局符号表中宣告了功能块 A 的实例后，系统才会在编译时依据三者的调用关系而产生对应的**功能块实例**。下列将以一范例说明在功能块调用功能块的结构下，系统的功能块实例将如何产生。

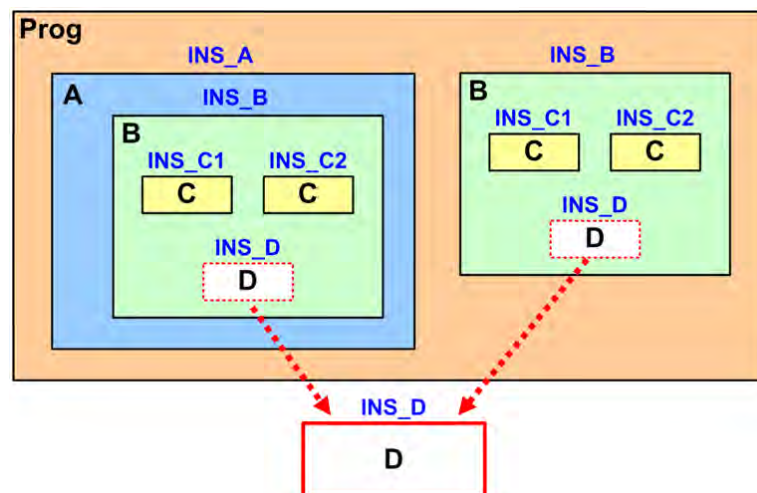
请参考下列功能块实例的宣告状况及各功能块之间的调用关系。其中除了功能块 D 的实例被宣告于全局符号表之外，其余的功能块实例皆被宣告于上层 POU（调用方）的局部符号表中。

7

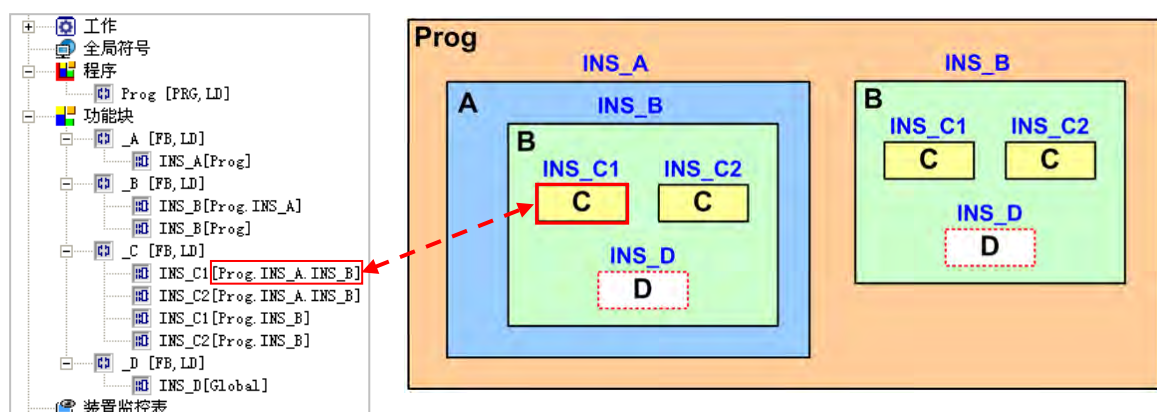


宣告位置		宣告项目
全局符号		1 个功能块 D 实例：INS_D
局部符号	程序 Prog	1 个功能块 A 实例：INS_A 1 个功能块 B 实例：INS_B
	功能块 A	1 个功能块 B 实例：INS_B
	功能块 B	2 个功能块 C 实例：INS_C1 及 INS_C2

下列示意图为系统依据上述的调用关系而在编译时所产生的功能块实例。且在下列的图标中可发现，因功能块 C 的两个实例被宣告于功能块 B 的局部符号表中，因此当有不同的功能块 B 实例被产生时，就如同细胞分裂一般，宣告于其局部符号表的功能块 C 实例也会等比的被产生出来。因为功能块 D 的实例被宣告于全局符号表内，所以该实例的功能与其内部储存的数据状态便会让所有的 POU 一起共享，因此系统只会产生一个功能块 D 的实例，且虽然功能块 D 的实例会在两个地方被调用，但实际在两处程序当中运作的都会是同一个功能块 D 的实例。



下图则为实际进行编译后，在项目管理区中所显示的项目，其中在每个功能块实例后方的中括号内，其文字的内容会由最上层的程序 POU 开始，依序往下列出各层调用方的 POU 名称。



在 DVP 机种中，功能块实例不可宣告在功能块的局部符号表中，而当要在功能块内部调用其它功能块时，下层功能块实例便必须宣告在全局符号表，因此操作时请小心避免运算数据的错乱，若需要独立的内存区块时，请于全局符号表中宣告不同的功能块实例。

在 AH500 的机种中，功能块实例允许宣告在功能块内部的局部符号表中，因此若非所调用的下层功能块实例也必须与其它 POU 共享，否则可直接将该下层功能块的实例宣告在局部符号表中即可。且在 AH500 系列的机种中，功能块实例的宣告位置将影响系统配置内存区块时所占用的资源。

7.2.7 功能块的内存配置

当功能块内部有宣告为自动配置地址的局部符号时，在编译及产生实例的过程中，系统便会为这些局部符号配置一个独立的内存区块；若功能块内部宣告的局部符号有自行指定装置地址，则该变量符号便会被视为绝对地址，而在编译时便不会另外为其配置内存区块，之后即使在不同的功能块实例当中，只要与该变量符号有关的程序，系统都会以用户自行指定的装置内容来参与运算。

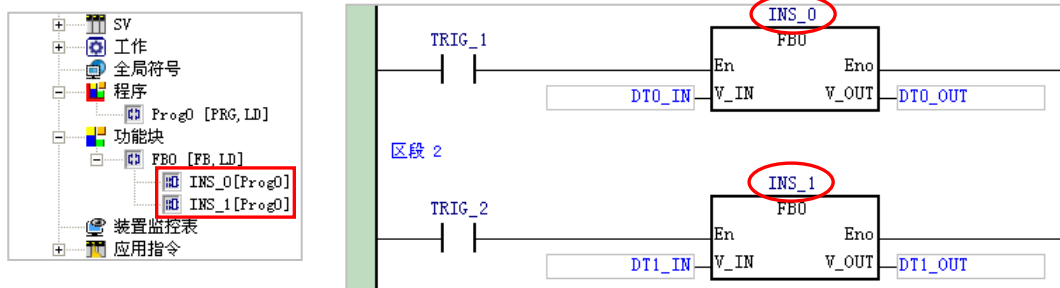
当系统在配置资源时，DVP 系列、DVPxxMC 系列、AS5xx 系列、AH5X0 系列、AH5X1 系列、AH560 冗余系列、AHxxEMC 系列与 AS 系列机种所采用的机制并不相同，请参考下述说明。

● DVP 系列机种

DVP 机种在配置内存区块时，每个实例都会被自动配置一个 P 装置，但若该 POU 的名称是以“P0_”或“P1_”做为开头，如“P0_FB”，则编译后，该实例将被强制配置 P0 或 P1 装置，其它依此类推；而后续若该功能块被宣告了一个以上的实例，则编译时便会因为重复使用相同的 P 装置而造成错误。

针对功能块内部所宣告的局部符号，系统将会依据每个变量符号的数据类型而为其配置对应的装置地址，例如 WORD 类型会配置一个 D 装置，而 BOOL 类型则会配置一个 M 装置，因此 DVP 机种的实例大小会依其所宣告的局部符号而定，且不同的实例会占用不同的装置地址。下图便是一个 DVP 机种所产生的两个实例，而在开启功能块实例的窗口后，于其各自的局部符号表中便可看到系统会为不同的实例配置不同的装置地址。





FBO (INS_0) 宣告来源: [Prog0]

局部符号						
类型	符号名称	地址	数据类型	初始值	符号批注	
VAR_INPUT	V_IN	D7000	WORD	0		
VAR_OUTPUT	V_OUT	D7001	WORD	0		
VAR	TEMP	M3003	BOOL	FALSE		

FBO (INS_1) 宣告来源: [Prog0]

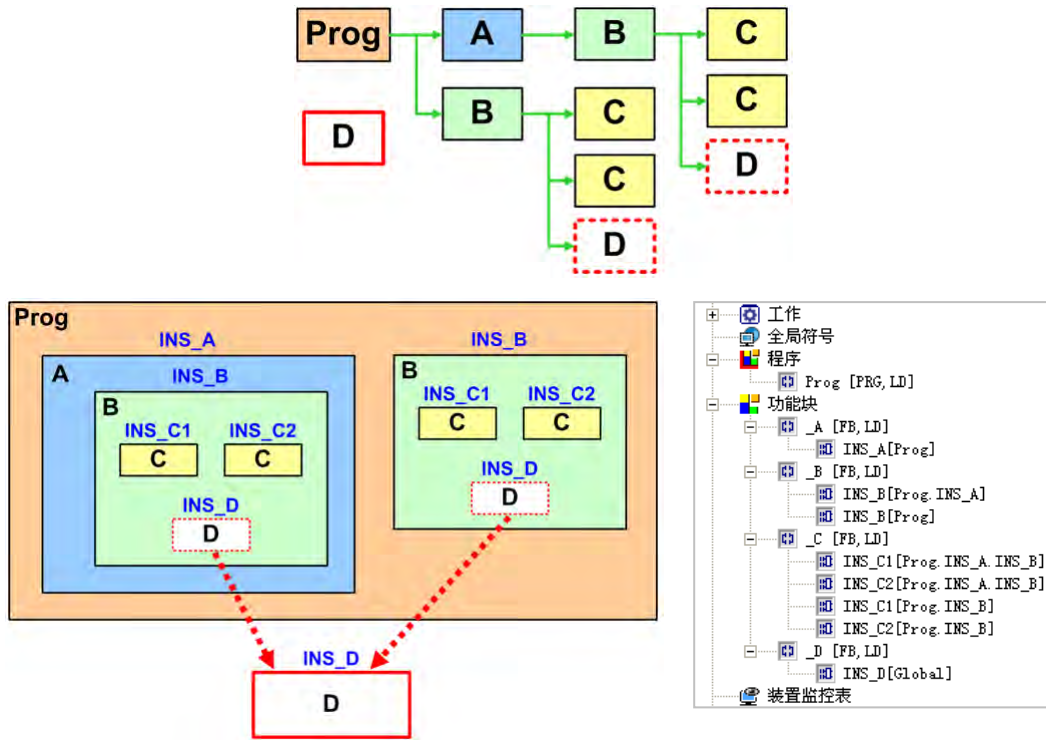
局部符号						
类型	符号名称	地址	数据类型	初始值	符号批注	
VAR_INPUT	V_IN	D7006	WORD	0		
VAR_OUTPUT	V_OUT	D7007	WORD	0		
VAR	TEMP	M3007	BOOL	FALSE		

● AH5X0 系列机种

在 AH5X0 系列机种中，系统将会在内部的存储器中预留固定数量的「实例内存区块」供功能块实例使用，而数量的大小将依机种而定，每个有产生实例的功能块定义会占用一个 P 装置；其中每个区块的容量大小都会固定为 4096 Words，且因配置的位置是主机预留的内存，因此用户无法藉由开启功能块实例的窗口而得知系统配置的结果。

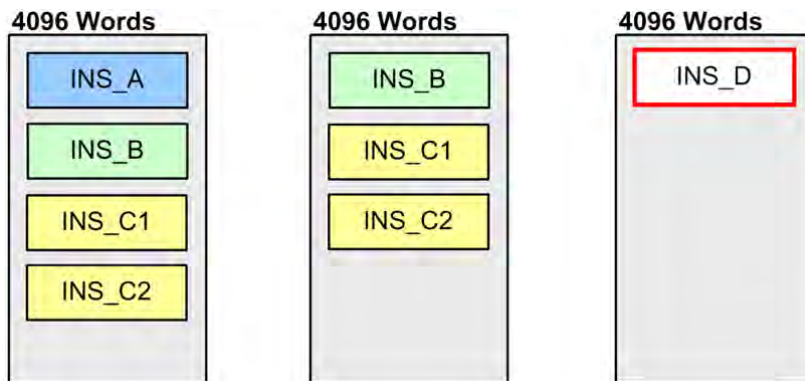
下列为第 7.2.6 节范例中的功能块调用结构。

7



以 AH5X0 机种来说，对于在程序 POU 的局部符号表中所宣告的第一层功能块实例，系统会单独配置一个 4096 Words 大小的「实例内存区块」，之后包括第一层的功能块实例与其下层调用的所有功能块实例，其所需的所有内存都会被分配在同一个 4096 Words 的区块中；另外，针对宣告于全局符号表的功能块实例，系统亦会单独为其配置一个相同大小的「实例内存区块」。

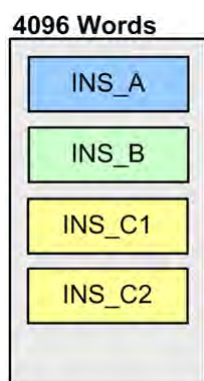
而根据上述的范例，系统总共会配置如下 3 个 4096 Words 大小的「实例内存区块」。



AH5X0 机种为功能块实例所配置的「实例内存区块」会存放在主机另外预留的内部存储器中，而配置时并不会占用用户可用的装置资源，且不论功能块实例的大小，所配置的「实例内存区块」都会固定为 4096 Words，因此在编程时请小心避免超过可用的区块大小。计算的方式如下。

- (1) 每个功能块实例都会固定占用最基本的 2 Words 大小。
- (2) 在每个实例的局部符号表中，单独针对每个变量符号，依据其数据大小计算每个变量符号所占用的 Word 数，未满 1 个 Word 以 1 个 Word 计算。至于功能块类型的变量符号（即下层的功能块实例）则可在上述（1）中一起统计，此处可先略过不计。

请参考下列的范例，其中各个功能块定义所宣告的局部符号亦统计于下列表格中。



功能块定义	局部符号数量
功能块 A	DWORD x 1 个、LREAL x 1 个、下层功能块实例 x 1 个
功能块 B	WORD x 1 个、REAL x 2 个、下层功能块实例 x 2 个
功能块 C	BOOL x 2 个、长度为 12 的 BOOL 数组 x 1 个

下列则为此范例的计算过程。

- (1) 每个功能块的实例会固定占用 2 Words 的大小，而本范例总共产生 4 个功能块实例，因此共计占用 8 Words。
- (2) 接着计算各个功能块实例的局部符号表中，所有非功能块类型的变量符号所占用的内存大小。

➤ 功能块实例 **INS_A**

➔ 2 Words (DWORD x 1 个) + 4 Words (LREAL x 1 个)，总计为 $2 + 4 = 6$ (Words)。

➤ 功能块实例 **INS_B**

➔ 1 Word (WORD x 1 个) + 4 Words (REAL x 2 个)，总计为 $1 + 4 = 5$ (Words)。

➤ 功能块实例 **INS_C1 / INS_C2**

因每个变量符号必须单独计算，且未满 1 个 Word 必须计为 1 个 Word，所以计算每个 BOOL 类型的变量符号时都必须以 1 个 Word 来计算；而长度为 12 的 BOOL 数组，因长度超过 1 个 Word，但未满 2 个 Word，因此必须计为 2 个 Word。计算如下。

➔ 2 Words (BOOL x 2 个) + 2 Words (长度为 12 的 BOOL 数组 x 1 个)，总计为 $2 + 2 = 4$ ；

而功能块 C 总共产生了 2 个实例，所以两个功能块实例合计为 $2 \times 4 = 8$ (Words)。

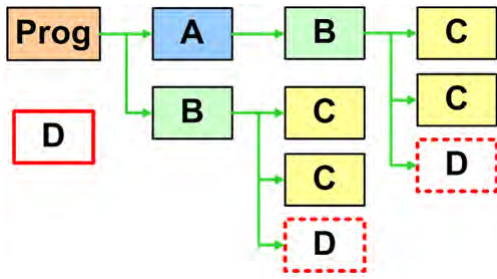
统计总和之后，可得知此区块对应的功能块实例所占用的内存为 $8 + 6 + 5 + 8 = 27$ (Words)。

● **AH5X1 系列、AH560 冗余系列、AHxxEMC 系列与 AS300 系列机种**

AH5X1 系列、AH560 冗余系列、AHxxEMC 系列与 AS300 系列机种功能块的内存，则会存放在主机另外预留的内部存储器中，而配置时并不会占用用户可用的装置资源，与 AH5X0 系列的不同之处为 AH5X1 系列、AHxxEMC 系列与 AS300 系列机种不会限制实例的大小或个数，而是将此内部存储器配置完为止，每个有产生实例的功能块定义会占用一个 P 装置。

下列使用第 7.2.6 节范例，其中各个功能块定义所宣告的区域与全局符号数据类型数量亦统计于下列表格中。

7



功能块定义	区域/全局符号数量
功能块 A	DWORD x 1 个、LREAL x 1 个 下层功能块实例 x 1 个
功能块 B	WORD x 1 个、REAL x 2 个、 层功能块实例 x 2 个
功能块 C	BOOL x 2 个、长度为 22 的 BOOL 数组 x 1 个
功能块 D	WORD x 1 个

因此内部存储器占用的方式将如下图所示



7

下列则为此范例的计算过程。

(1) 所有的实例依宣告的顺序占用内部存储器，首先加入宣告在全局符号表中的功能块实例 INS_D。

➤ 功能块实例 **INS_D**

➔ 1 Words，总计为 1 (Words)。

(2) 接着依照调用的顺序计算各个区域符号表中，所有非功能块类型的变量符号所占用的内存大小。

➤ 功能块实例 **INS_A**

➔ 2 Words (DWORD x 1 个) + 4 Words (LREAL x 1 个)，总计为 2 + 4 = 6 (Words)。

➤ 功能块实例 **INS_B**

➔ 1 Word (WORD x 1 个) + 4 Words (REAL x 2 个)，总计为 1 + 4 = 5 (Words)。

➤ 功能块实例 INS_C1 / INS_C2

因每个变量符号必须单独计算，且未满足 1 个 Word 必须计为 1 个 Word，所以计算每个 BOOL 类型的变量符号时都必须以 1 个 Word 来计算；而长度为 22 的 BOOL 数组，因长度超过 1 个 Word，但未满足 2 个 Word，因此必须计为 2 个 Word。计算如下。

→ 2 Words (BOOL x 2 个) + 2 Words (长度为 22 的 BOOL 数组 x 1 个)，总计为 2 + 2 = 4；而功能块 C 总共产生了 2 个实例，所以两个功能块实例合计为 2 x 4 = 8 (Words)。

➤ 功能块实例 INS_B / INS_C1 / INS_C2

→ 如同前述 INS_B 占用 5 (Words)；INS_C1 / INS_C2 占用 8 (Words)。所以三个功能块实例合计为 5 + 8 = 13 (Words)。

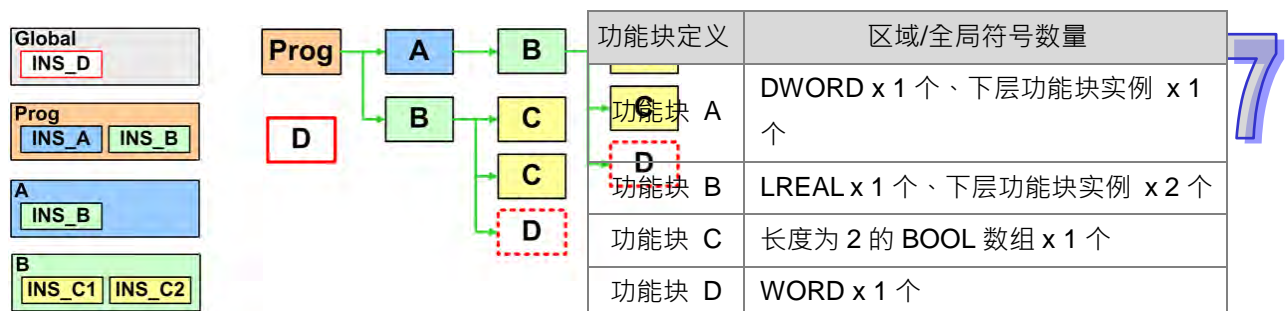
统计总和之后，可得知此区块对应的功能块实例所占用的内存为 1 + 6 + 5 + 8 + 13 = 33 (Words)。

● DVPxxMC 系列与 AS5xx 机种

DVPxxMC 与 AS5xx 机种在配置内存区块时，实例会分配在内存的某一区块，直到此区块占满 2×10^{13} Byte 的内存空间为止，但用户须注意，内存区块仍会被 POU 数量所影响，当用户定义 FB 数量越多，相对地，在引用每一定义的 FB 所宣告的实例数将会有所限制。

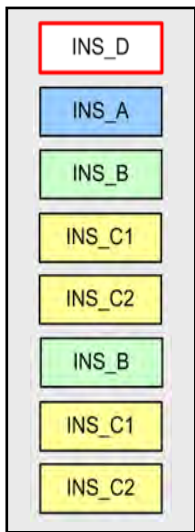
在每个实例的区域符号表中，单独针对每个变量符号，依据其数据大小计算每个变量符号所占用的 Byte 数，在未满足 1 个 Byte 的情况，若没有绑定装置以 1 个 Byte 计算；若变量符号数据类型为 BOOL 且有绑定装置，装置类型为 Bit 则以 Bit 计算，装置类型为 Byte 则以 Byte 计算。至于功能块类型的变量符号（即下层的功能块实例）

下列为第 7.2.6 节范例中的功能块调用结构，详细内容请参考该节的相关说明。其中各个功能块定义所宣告的区域与全局符号数据类型数量亦统计于下列表格中。



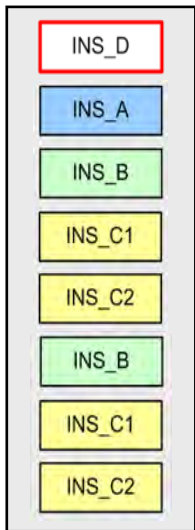
因此内部存储器占用的方式将如下图所示

2x10¹³Bytes



实例	符号资料型態	佔用記憶體															
INS_D	WORD																
INS_A	DWORD																
INS_B	LREAL																
INS_C1	ARRAY[2] of BOOL																
INS_C2	ARRAY[2] of BOOL																
INS_B	LREAL																
INS_C1	ARRAY[2] of BOOL																
INS_C2	ARRAY[2] of BOOL																

2x10¹³Bytes



实例	符号资料型態	佔用記憶體															
INS_D	WORD																
INS_A	DWORD																
INS_B	LREAL																
INS_C1	ARRAY[2] of BOOL																
INS_C2	ARRAY[2] of BOOL																
INS_B	LREAL																
INS_C1	ARRAY[2] of BOOL																
INS_C2	ARRAY[2] of BOOL																

7

下列则为此范例的计算过程。

➢ 功能块实例 **INS_A**

➔ 4 Bytes (DWORDx 1 个)，总计为 5 (Bytes)。

➢ 功能块实例 **INS_B**

➔ 8 Bytes (LREAL x 1 个)，总计为 8 (Bytes)。

➢ 功能块实例 **INS_C1 / INS_C2**

BOOL 数组，每个 ARRAY 的 BOOL 元素将固定占掉 1 个 Byte。计算如下。

➔ 2 Bytes (长度为 2 的 BOOL 数组 x 1 个)，总计为 2；而功能块 C 总共产生了 2 个实例，所以两个功能块实例合计为 2 x 2 = 4 (Bytes)。

统计总和之后，可得知此区块对应的功能块实例所占用的内存为 5 + 8 + 4 = 17 (Bytes)。

7.3 功能块的使用

7.3.1 功能块的基本规格

下列表格为 ISPSOft 的功能块规格，用户在使用功能块时，请务必确认所有的操作都确实符合下表所规范的内容，否则便有可能造成编译或执行上的错误。

● 功能块规格

功能块定义数	全机种：无限制。
功能块实例数	AH5X1/ AH560 冗余系列/AHxxEMC/AS：决定于可用控制器装置资源。 AH5X0：最大的「实例内存区块」数量会依机种而定（请参考附录 B）；决定于可用控制器装置资源。 DVP：决定于可用控制器装置资源。
功能块容量	AH5X1/ AH560 冗余系列/AS：无限制（直到内部存储器被配置完毕为止） AH5X0：每一个「实例内存区块」所提供的内存容量固定为 4096 Words DVP：决定于可配置的装置资源
巢状结构层数	AH 系列机种：功能块调用功能块的巢状结构，最大允许至 32 层、 AS/DVP 系列机种：最大允许至 16 层、 DVPxxMC 与 AS5xx 机种：则没有限制。

● 注意事项

- 所有的功能块端点都必须被指定一个操作数，该操作数可为装置地址或变量符号，而当端点的符号类型为 VAR_INPUT 时，所指定的操作数亦可为常数。
- 不可在下层功能块中调用上层功能块，或是自己调用自己。
- 导出功能块时仅包含该功能块的定义，下层功能块的定义与使用的全局符号则不会一并导出。
- 除了 ST 语言之外，使用功能块时的 En 接点前方必须连接至有逻辑状态的接点或区块。
- 支持建立功能块的语言有 LD、IL、ST、FBD 及 CFC，但不包含 SFC。
- 可调用功能块的语言有 LD、IL、ST、FBD 及 CFC，而在 SFC 的 Action 程序中也可调用功能块。
- 用户只能修改功能块定义而无法对功能块实例进行修改。（请参考第 7.3.4 节）
- 对功能块定义进行在线编辑时，若修改的结果造成上层 POU 的功能块图标异常，此时必须先离线并将该图标删除后再重新加入，接着才可重新编译并下载程序。（请参考第 7.3.4 节）
- 使用跳跃指令如 CJ、JMP...时，其目的地仅限同层功能块的内部，而不可跳出功能块外部或跳至下层功能块的内部。
- 在 AH/AS 系列机种的功能块内部，用户不可宣告 TIMER 与 COUNTER 类型的变量符号，但可以使用指针类型的变量符号。
- AH/AS 机种的功能块内部不可使用脉冲型的指令（如 MOVP），而其它不可使用的指令尚有 LDP/ANDP/ORP、LDF/ANDF/ORF、PLS、PLF、NP、PN、MC、MCR、GOEND。
- AH/AS 机种本身有提供在功能块内部专用的脉冲型指令，详细内容请参考下节的相关说明。

- *1. 新增功能块 POU 的方法请参考第 5.4.1 节的介绍；而宣告功能块实例的方法则请参考第 6.2.2 节的说明。
- *2. 调用功能块的方法请参考第 10 章~第 15 章中各编程语言的相关介绍。

7.3.2 功能块专用的脉冲型指令（限定 AH/AS 机种）

因 AH/AS 系列机种的功能块内部无法使用脉冲型的指令，如 LDP、LDF、NP、PN 等，因此在应用上可考虑搭配下列的指令来达到相同的效果。关于各指令的使用说明及限制请参考 **AH500 程序手册**。

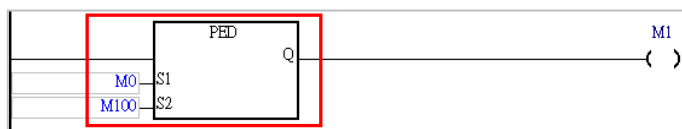
● NED 及 PED

其功能类似 LDP 及 LDF，唯 NED 与 PED 需另外指定一个 BOOL 格式的寄存装置，且被用来做为寄存功能的装置在整个项目中都不可再被使用于其它地方，否则便可能产生运算错乱。

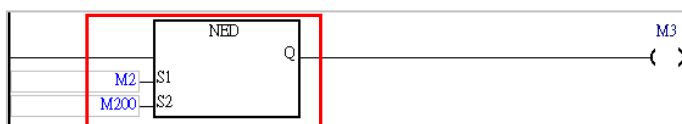
- 程序 POU 中可使用 LDP 指令（上升沿触发）与 LDF 指令（下降沿触发）。



- 功能块内部可使用 PED 来替换 LDP，但必须多指定一个寄存用的装置，如下图的 M100。



- 功能块内部可使用 NED 来替换 LDF，但必须多指定一个寄存用的装置，如下图的 M200。

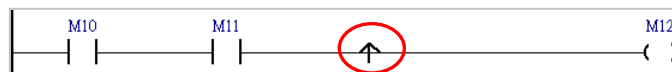


7

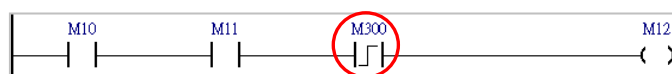
● FB_NP 及 FB_PN

其功能类似 NP 及 PN，唯 FB_NP 与 FB_PN 需另外指定一个 BOOL 格式的寄存装置，且被用来做为寄存功能的装置在整个项目中都不可再被使用于其它地方，否则便可能产生运算错乱。

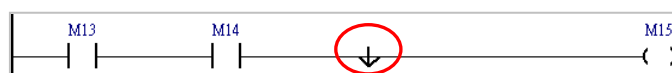
- 程序 POU 中可使用 NP 指令。



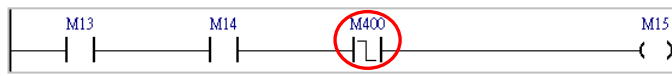
- 功能块内部可使用 FB_NP 来替换 NP，但必须多指定一个寄存用的装置，如下图的 M300。



- 程序 POU 中可使用 PN 指令。

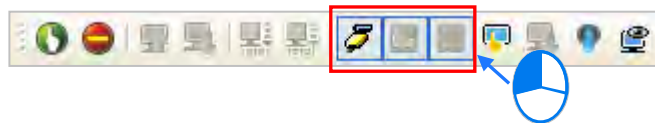


► 功能块内部可使用 FB_PN 来替换 PN，但必须多指定一个寄存用的装置，如下图的 M400。

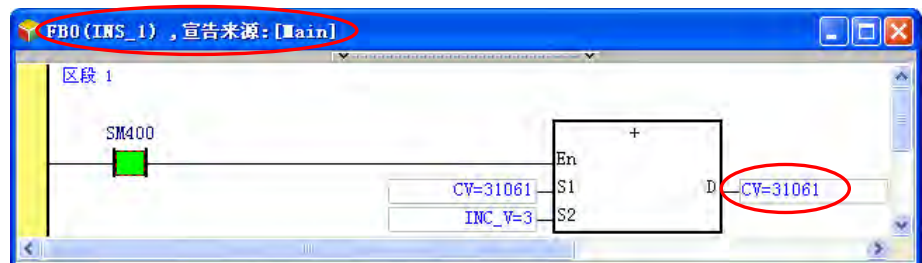
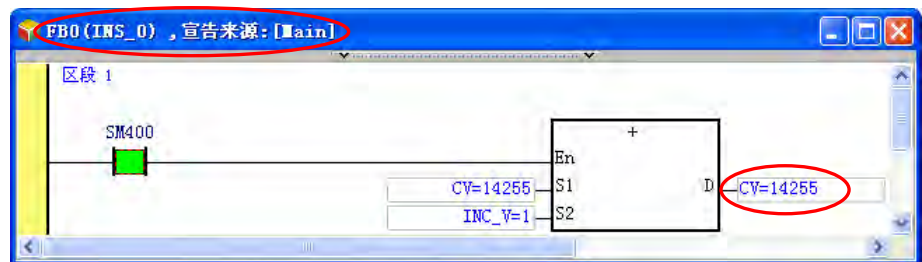


7.3.3 功能块的程序监控

因为在程序执行时，实际参与运作的是「功能块实例」，所以当进行在线的程序监控时，必须开启「功能块实例」的窗口才可正常进行监控。

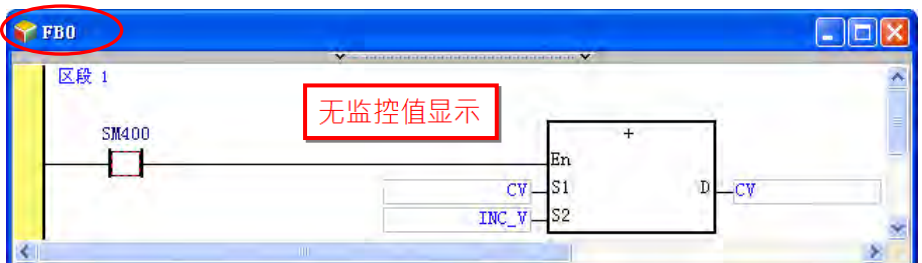


如下图，当进行程序监控时，必须开启「功能块实例」的窗口才可进行监控；且由下图可发现，不同的功能块实例，其运作过程是彼此独立的。



7

若开启「功能块定义」的窗口时，该窗口中将不会有任何监控值的显示，因此无法进行程序监控。



*.进行在线编辑时，则必须于「功能块定义」的窗口中才可以修改功能块的内容，相关细节请参考第 7.3.4 节的说明。

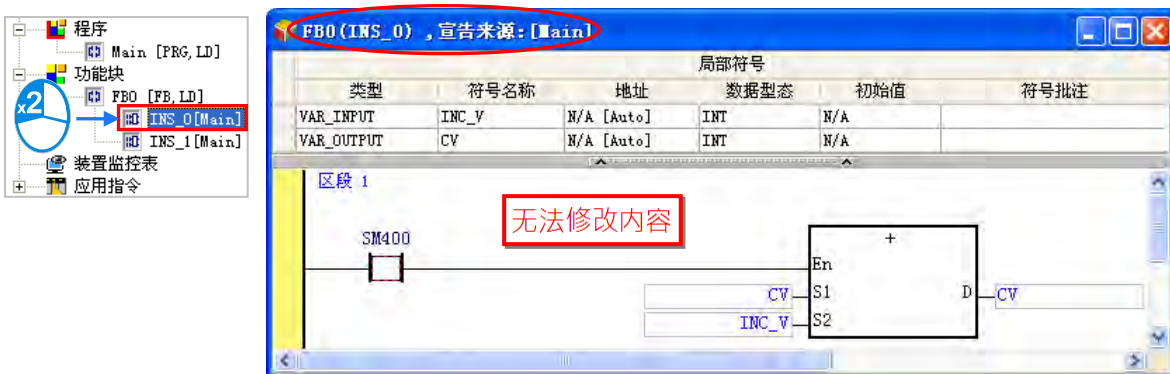
7.3.4 功能块的修改

因「功能块实例」是依据「功能块定义」所产生的对象，因此「功能块实例」中的内容将无法直接修改，而必须在「功能块定义」的窗口中才可以修改内容；当然，所修改的内容将会应用在所有由该功能块定义所产生的实例中，且在修改之后，仍必须经过编译方可产生新的「功能块实例」。同样的，进行在线编辑时，亦必须开启「功能块定义」的窗口才可修改内容。

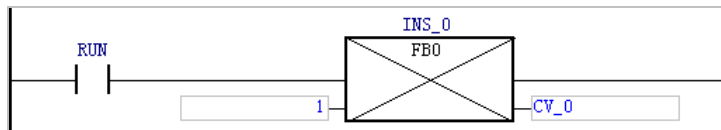
如下图，「功能块定义」处双击鼠标左键，之后便可在「功能块定义」的窗口中修改程序内容。



若开启的是「功能块实例」窗口，则用户将无法在此处修改程序内容。



当某个功能块已经被其它的 POU 所调用时，则删除该功能块便可能造成上层 POU 中的功能块图标产生异常（如下图），此时只要将该图标删除即可。



7.4 范例实作

在第 4 章的内容中，我们曾以一个简单的范例来示范如何建立传统的梯形图程序，并在第 5.6.1 节中以 IEC61131-3 的程序架构重新规划，之后又在第 6.3 节中使用变量符号来替换原本的装置编号；而在接下来的内容中，我们将延续第 6.3 节中所建立的范例，并在范例中示范如何规划并使用功能块。

7.4.1 程序规划

下表为第 6.3 节的范例程序中，各 POU 的功能简介。

POU 名称	功能说明
RUN_STOP	当按下 启动按钮 时让设备开始运转，而按下 停止按钮 或侦测到错误发生时，设备便会停止运转。
GLUING	共包含两个相同的工作站程序，内容为当 光电传感器 为 ON 时，便将 注射器的触发信号 设置为 ON；而当 光电传感器 变为 OFF 时，便将该信号重置回 OFF。
COUNTING	计数工件的入料数量，当入料数量到达 100 时，便将 完成标志 设置为 ON。

根据各 POU 的程序特性，我们预计规划如下的两个功能块。

- 在「**GLUING**」POU 中共包含了两段相同的程序，因此我们可以将该程序建立为一个功能块，之后便可重复使用；而所完成的功能块将会于「**GLUING**」POU 当中被调用，且因两个工作站之间的运作应该彼此独立，因此我们会为其宣告两个独立的功能块实例。
- 在「**COUNTING**」POU 中的程序，除了**运转标志**与**完成标志**之外，本身的程序并不会与其它程序有太多的联结，因此独立性相当的高，所以我们可以将其建立为功能块以增加程序的可读性。

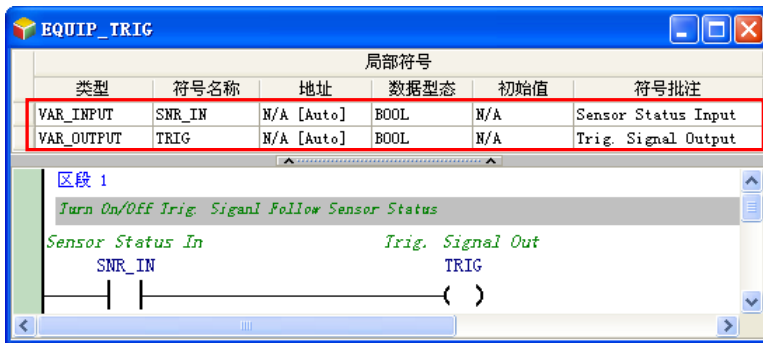
7.4.2 建立范例程序

接下来可直接开启第 6.3 节当中的范例来进行修改或是重新建立程序。若要重新建立程序时，请先新增一个项目，之后参考第 4 章至第 6 章的范例介绍，并配合接下来的说明逐一完成此节的范例。

首先请参考第 5.4.1 节的说明，于项目管理区的「功能块」下新增一个 POU - 「**EQUIP_TRIG**」。



功能块「**EQUIP_TRIG**」主要是负责点胶的功能。接着请参考第 6.2.2 节的说明，局部符号表中建立如下的 2 个端点符号，并参考第 4 章至第 6 章的范例介绍，完成下图的程序。

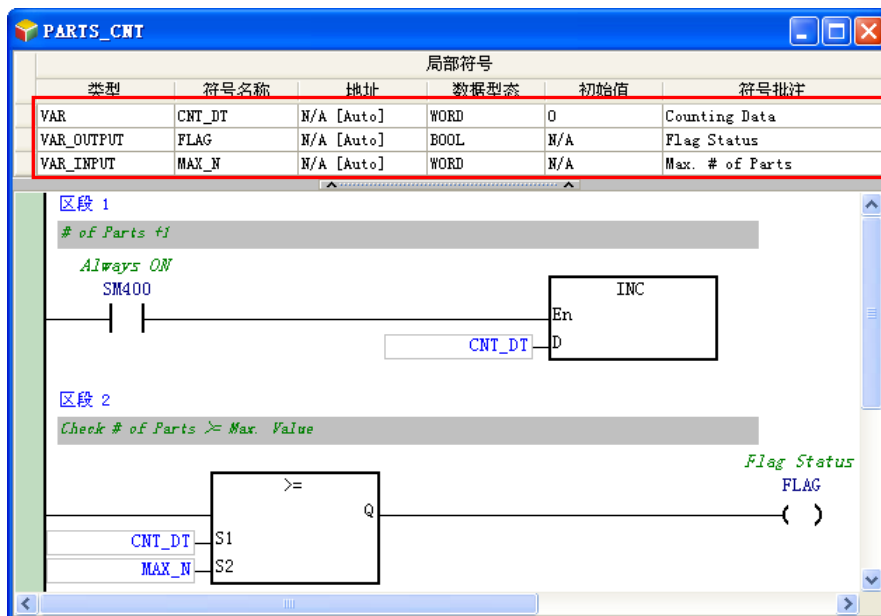


再来请在项目管理区的「功能块」下新增另一 POU - 「**PARTS_CNT**」。



功能块「**PARTS_CNT**」的功能则是负责计数入料工件的数量。接着请参考第 6.2.2 节的说明，于局部符号表中建立如下的 3 个符号，而其中 CNT_DT 的初始值请将其设置为 0；之后请参考第 4 章至第 6 章的范例介绍，完成下图的程序。

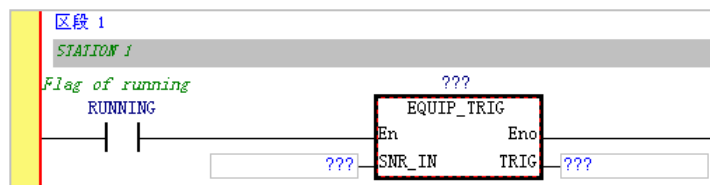
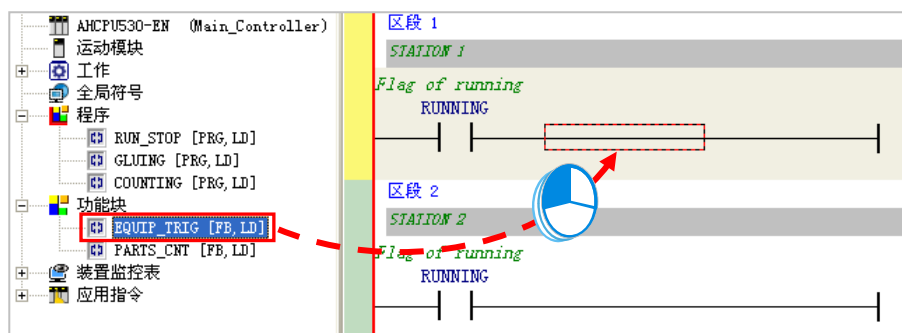
7



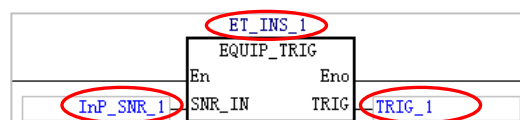
再来我们便要在「GLUING」POU 当中调用功能块「EQUIP_TRIG」。首先请参考第 6.2.2 节的说明在「GLUING」POU 的局部符号表中宣告如下图所示的 2 个功能块实例，并将程序修改如下。



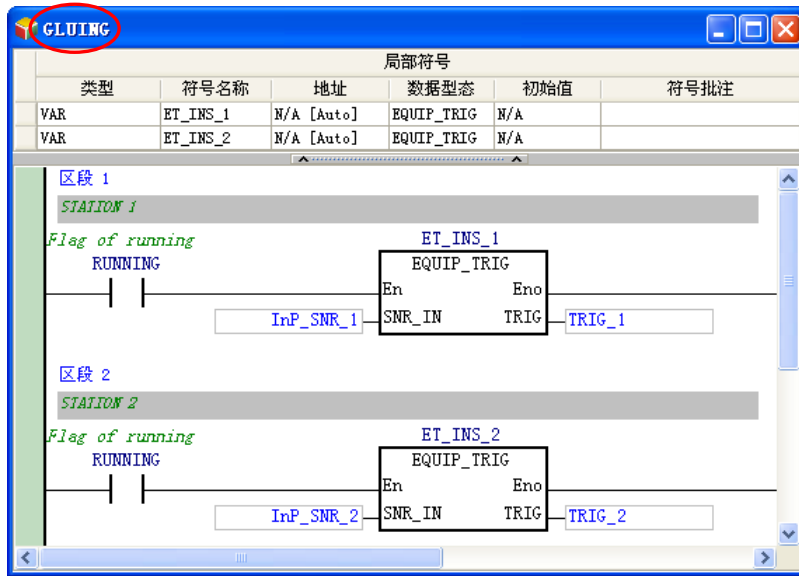
要在程序当中插入功能块的方式与插入应用指令 API 的方式类似，直接由项目管理区中，将准备插入的功能块「EQUIP_TRIG」拖曳至欲插入的位置即可。而若该功能块的下层已产生实例时，拖曳时请注意必须拖曳功能块的定义，而不可拖曳其下层的功能块实例项目。



插入功能块后，紧接着请逐一指定功能块实例与操作数。



接下来请依照上述的说明完成「GLUING」POU 的修改，而完成的程序如下。

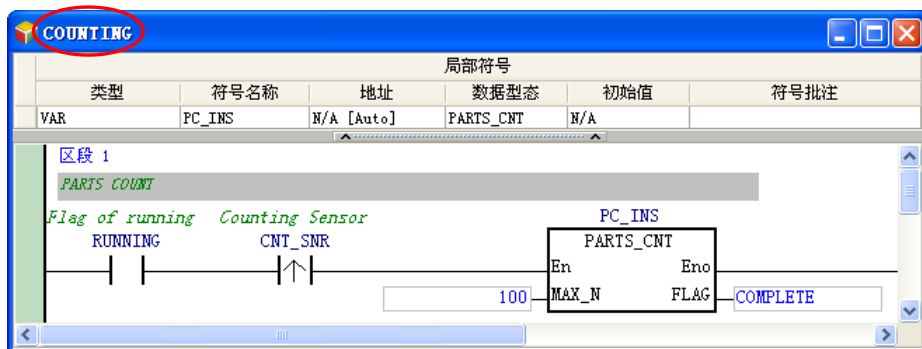


我们接着继续在「COUNTING」POU 当中调用功能块「PARTS_CNT」。首先在「COUNTING」POU 的局部符号表中宣告如下的功能块实例，而原本存在的局部符号「CNT_DT」则已没有作用，所以请将其删除，之后请将「COUNTING」POU 的程序修改如下。



7

接着请参考前述的说明插入功能块「PARTS_CNT」，并完成「COUNTING」POU 的修改。



到目前为止，此节的范例程序已编辑完成，接着请再次比对各位所编辑的程序与下列各项是否相同。



类型	符号名称	地址	数据类型	初始值	符号批注
VAR	START_BT	XO.0	BOOL	N/A	"START" Button
VAR	STOP_BT	XO.1	BOOL	N/A	"STOP" Button
VAR	InP_SNR_1	XO.2	BOOL	N/A	In-position Sensor 1
VAR	InP_SNR_2	XO.3	BOOL	N/A	In-position Sensor 2
VAR	CNT_SNR	XO.4	BOOL	N/A	Counting Sensor
VAR	CONVEYER	YO.0	BOOL	N/A	Conveyor ON/OFF
VAR	TRIG_1	YO.1	BOOL	N/A	Trig. signal 1
VAR	TRIG_2	YO.2	BOOL	N/A	Trig. signal 2
VAR	RUNNING	N/A [Auto]	BOOL	N/A	Flag of running status
VAR	COMPLETE	N/A [Auto]	BOOL	N/A	Flag of completion
VAR	ERROR	N/A [Auto]	BOOL	N/A	Flag of error status

类型	符号名称	地址	数据类型	初始值	符号批注
VAR_INPUT	SNR_IN	N/A [Auto]	BOOL	N/A	Sensor Status Input
VAR_OUTPUT	TRIG	N/A [Auto]	BOOL	N/A	Trig. Signal Output

区段 1
 Turn On/OFF Trig. Signal Follow Sensor Status

```

    Sensor Status In          Trig. Signal Out
      SNR_IN                   TRIG
    | |                         ( )
    ---|-----|-----
    
```

类型	符号名称	地址	数据类型	初始值	符号批注
VAR	CNT_DT	N/A [Auto]	WORD	0	Counting Data
VAR_OUTPUT	FLAG	N/A [Auto]	BOOL	N/A	Flag Status
VAR_INPUT	MAX_N	N/A [Auto]	WORD	N/A	Max. # of Parts

区段 1
 # of Parts fj

```

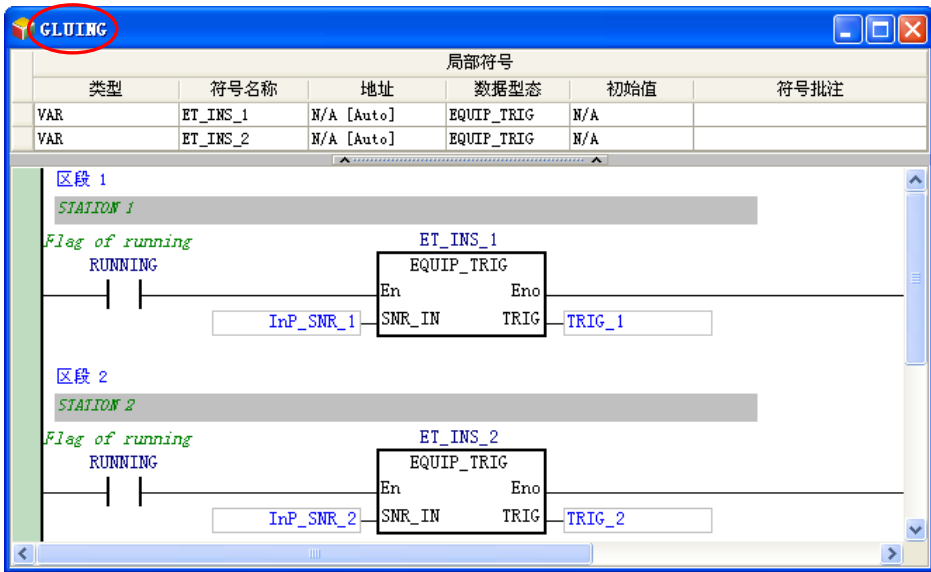
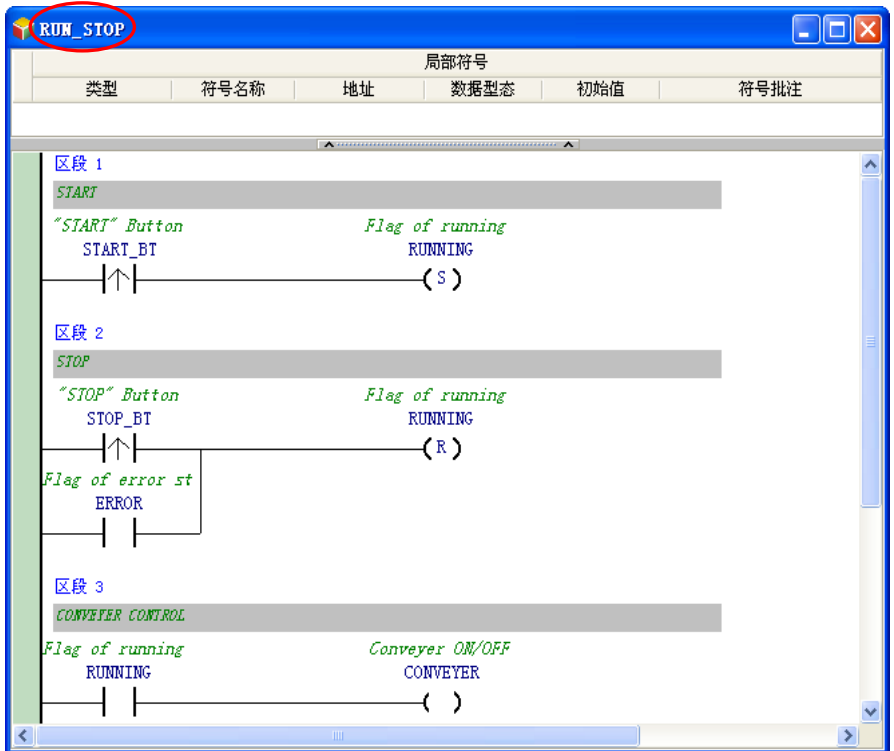
    Always ON
    SM400
    | |-----|-----> En
    | |-----|-----> D
    | |-----|-----> CNT_DT
    | |-----|-----> INC
    
```

区段 2
 Check # of Parts >= Max. Value

```

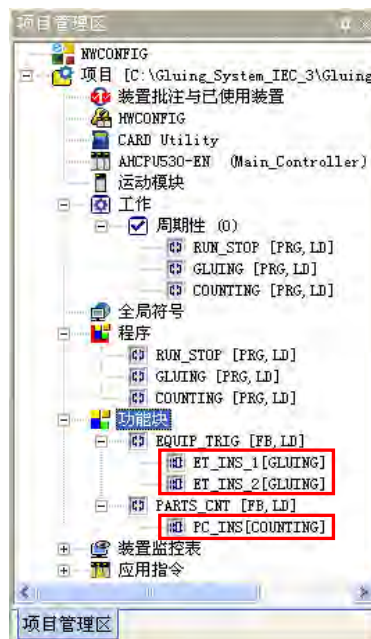
    CNT_DT S1
    MAX_N S2
    | |-----|-----> Q
    | |-----|-----> FLAG
    | |-----|-----> ( )
    
```

7



7

在确认编辑的程序与本节所列的范例相同后，请试着先执行编译，而编译完成后便可发现在项目管理区中会多出功能块实例的对象。



接着请确认已完成硬件组态、硬件参数设置、工作 (TASK) 管理...等工作，若否，则请依据第 4 章至第 6 章的范例，逐一完成编辑与设置；完成后便可重新执行编译，并将硬件参数与程序下载至主机当中进行测试。下载程序时，请一并勾选「符号初始值」，以便将功能块「PARTS_CNT」中的局部符号「CNT_DT」的初始值一并写入主机中。



完成下载之后，我们试着进入程序监控的状态，并将主机状态设为「RUN」，接着请依据下列的步骤来进行功能块特性的验证。

7

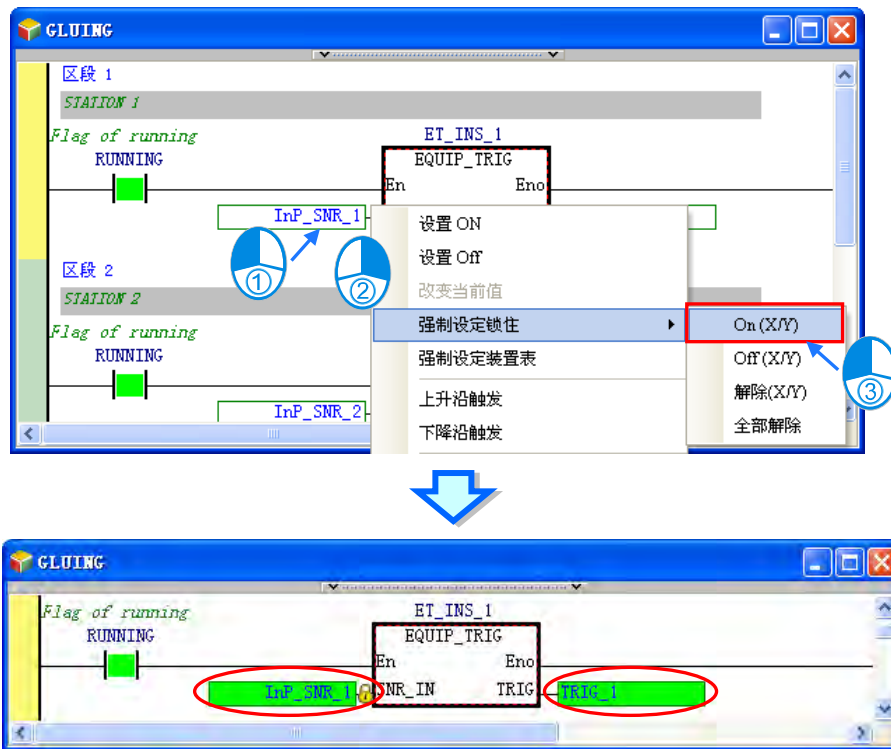


进入程序监控画面并启动 PLC 后，请先开启「**RUN_STOP**」POU 的窗口，并将「**START_BT**」的状态强制设定为 ON，此时「**RUNNING**」标志便会为 ON，代表设备已经开始运转。

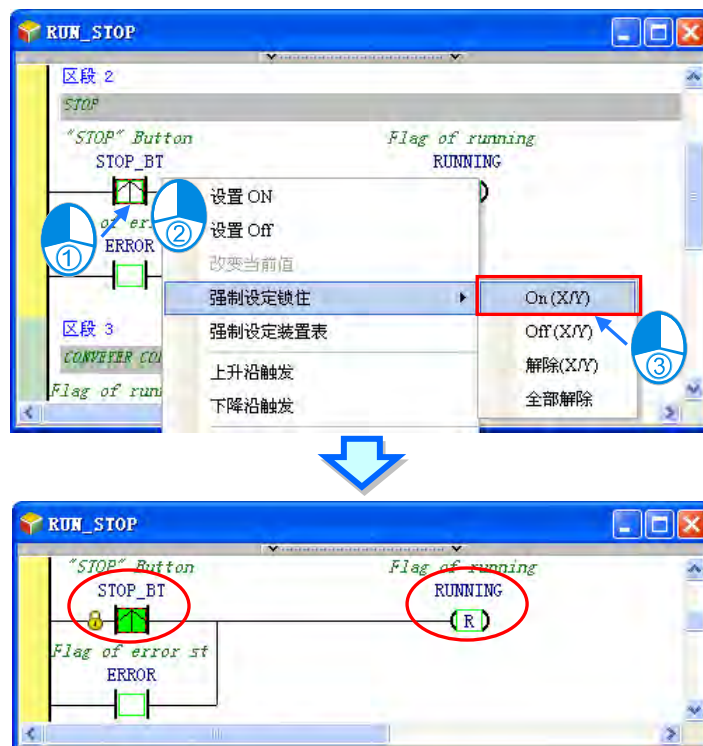


接着请再开启「**GLUING**」POU 的窗口，并将「**InP_SNR_1**」的状态也强制设定为 ON，而此时的「**TRIG_1**」状态也会跟着变为 ON，代表传感器已感测到工件，并将触发信号送出。

7



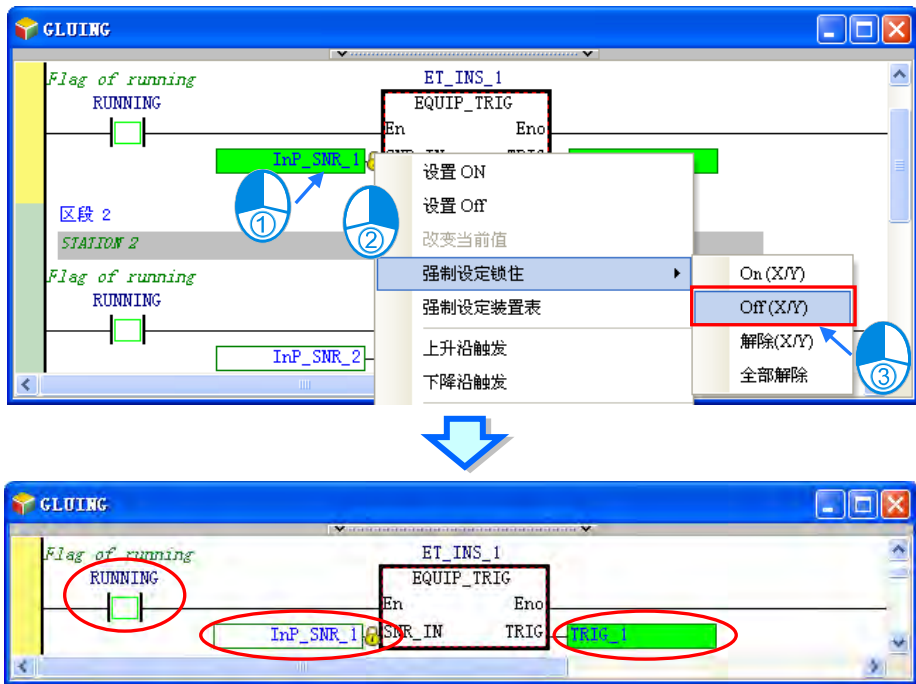
再来请至「RUN_STOP」POU 的窗口中，并将「STOP_BT」的状态强制设定为 ON，而此时标志「RUNNING」的状态便会恢复为 OFF，代表设备已停止运转。



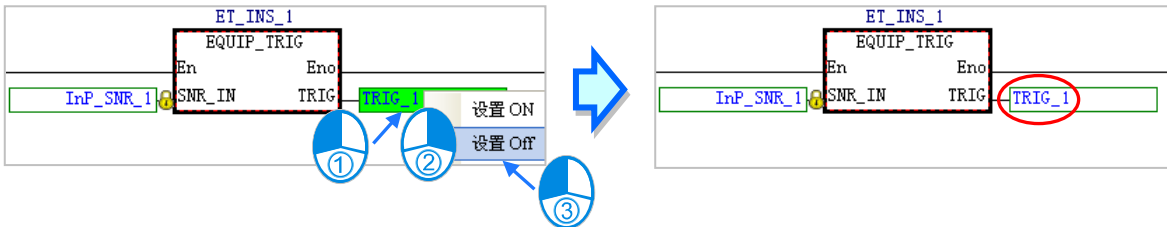
接着请再回至「GLUING」POU 的窗口，并将「InP_SNR_1」的状态强制设定为 OFF，但此时输出「TRIG_1」的状态却仍会维持为 ON；而此状况则是正确的，主要是因为连接至功能块 EN 接点的标

7

志「**RUNNING**」已是 OFF 状态，而这也代表功能块将不会被执行，因此功能块并不会将其「**TRIG**」端点的值输出给变量符号「**TRIG_1**」，所以「**TRIG_1**」的状态会维持原状。

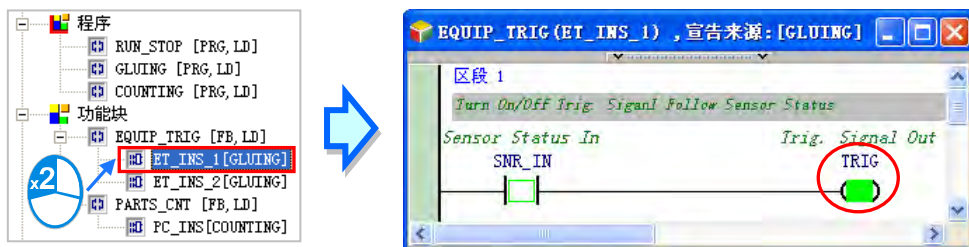


接着我们直接将「**TRIG_1**」的状态设置为 OFF。



7

然后开启功能块实例「**ET_INS_1**」的窗口。而由功能块实例的监控画面可发现，「**TRIG**」端点的状态仍会维持 ON，这代表当功能块实例不被执行时，其内部的变量符号仍会保持在最后执行的状态。



接着我们再开启功能块实例「**ET_INS_2**」的窗口并与「**ET_INS_1**」进行比对，可发现两者的状态并不相同，而这也代表不同功能块实例之间的运作是彼此独立的。



依据上述的操作，我们可验证下述三点。

1. 当 EN 接点的状态为 OFF 时，功能块内部的程序将不会被执行。
- 2 功能块内部的变量符号状态会在执行结束后被保留下来。
3. 不同功能块实例之间的运作是彼此独立的。

不过很明显的，目前程序的执行状况已与原本不同，在第 6.3 节的范例中，当「RUNNING」标志为 OFF 时，「TRIG_1」的状态会被设置为 OFF；而在本节的范例中，当「RUNNING」标志为 OFF 时，「TRIG_1」会保持在最后的状态；不过这也是事先就已预期且可接受的状态，主要是因为设备的规划中，将「TRIG_1」设置为 OFF 的目的是为了能进行下次的触发，因此即使在设备不运转的状况下，「TRIG_1」的状态保持原状也不会影响设备原本的功能性。关于设备功能的规划，请参考第 4.1 节的相关说明。

7.4.3 函数 (Function) 范例程序

宣告一个 FUN0 的函数，返回值为 INT 类型。其中 FUN0 定义两个输入 INT 类型的符号相加后再加 1，并将返回值填入 FUN0。用法如下图：

The screenshot displays two windows from the SIMATIC Manager software. The top window, titled 'FUN0', shows the function definition. It includes a table for local symbols and a single line of code: `0001 FUN0 := input1 + input2 + 1;`. The bottom window, titled 'Prog0', shows the function being called within a program. It includes a table for local symbols and three lines of code: `0001 mydata1 := 2;`, `0002 mydata2 := 3;`, and `0003 myresult := FUN0(mydata1, mydata2);`. The code in the bottom window is highlighted in yellow.

局部符号						
类型	符号名称	地址	数据类型	初始值	符号注释	
VAR_INPUT	input1	N/A [Auto]	INT	N/A		
VAR_INPUT	input2	N/A [Auto]	INT	N/A		

```

0001 FUN0 := input1 + input2 + 1;

```

局部符号						
类型	符号名称	地址	数据类型	初始值	符号注释	
VAR	mydata1	N/A [Auto]	INT	N/A		
VAR	mydata2	N/A [Auto]	INT	N/A		
VAR	myresult	N/A [Auto]	INT	N/A		

```

0001 mydata1 := 2;
0002 mydata2 := 3;
0003 myresult := FUN0(mydata1, mydata2);

```

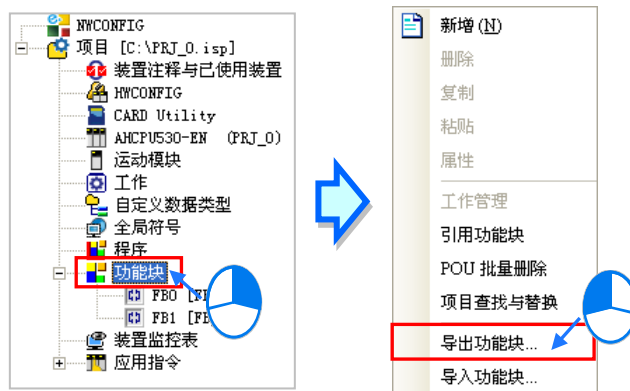
7

7.5 认识函数库 (Library)

函数库(Library)包括了用户自定义函数库(User Defined Library)与台达函数库(Delta Library)。有了前述章节功能块的概念之后，对于常用的功能块，用户可以将其导出成为用户自定义函数库(User Defined Library)，以便在不同的项目中重复使用这些功能块。而台达也对于各种应用场合提供一系列的功能块，存放在台达函数库 (Delta Library) 中，供用户直接运用。

7.5.1 新增用户自定义函数库

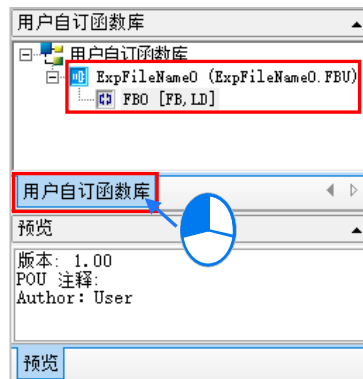
当项目中建立了功能块，希望在不同的项目中重复使用时，便可利用导出到用户自定义函数库的功能。选择项目管理区的功能块项目，点击鼠标右键开启快捷选单，点选「导出功能块」。



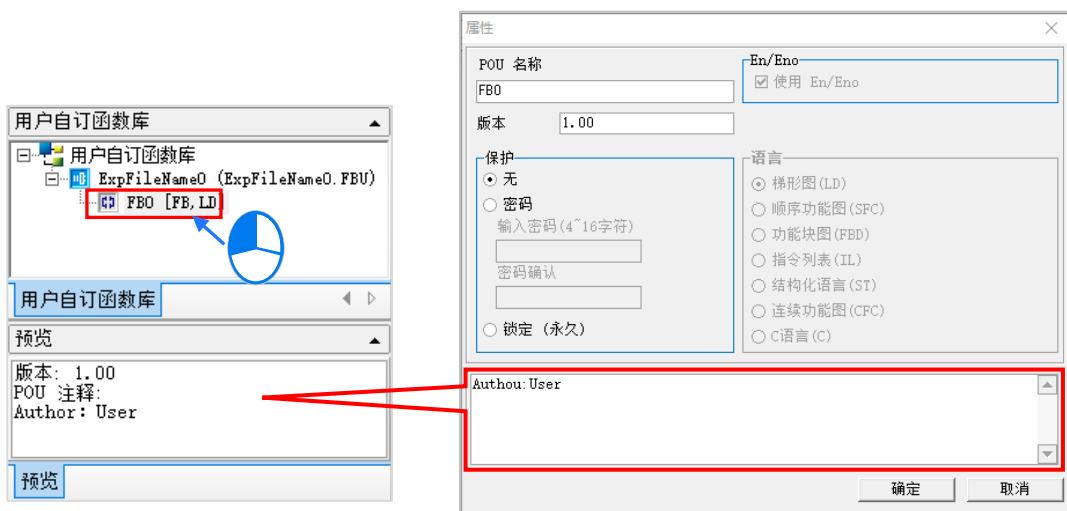
将出现导出功能块窗口，依序输入文件名、勾选导出至用户自定义函数库及要导出的功能块；文件名为用于函数库的分类，同一文件名可导出多个功能块函数库。完成设定后按下「确定」按钮，再于导出成功窗口按下「确定」按钮完成。



在函数库区点选用户自定义函数库分页，便会在用户自定义函数库区中找到刚才导出的函数。



此时点选此函数，在预览区将会显示此函数功能块属性设定的批注内容。

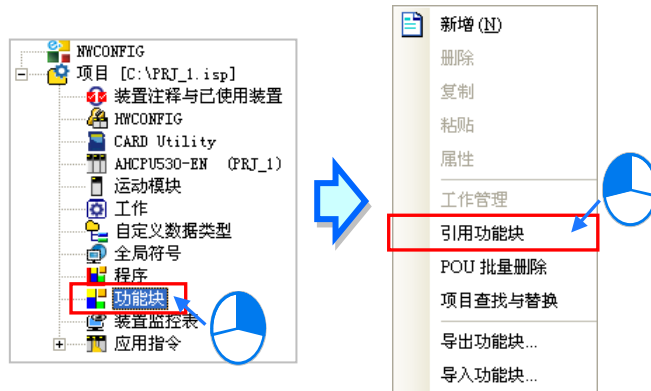


7.5.2 引用用户自定义函数库

如同前述章节，一旦导出功能块到用户自定义函数库后，同一台计算机即便开启新的项目，在用户自定义函数库中，也会出现已导出功能块的函数，此时便可以下列方式将函数引用到功能块区中。

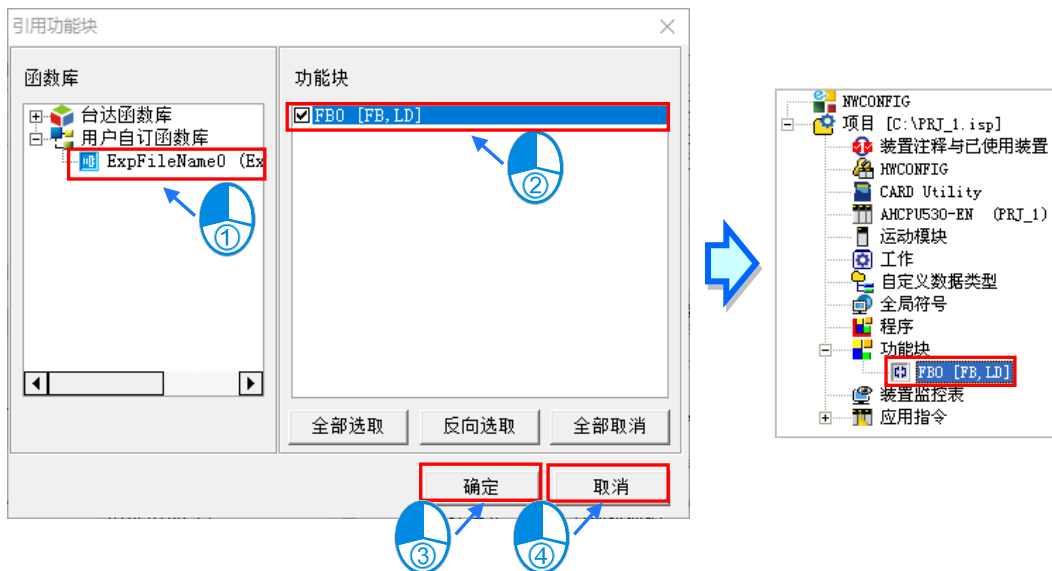
- 方法一

点选功能块项目，按下鼠标右键开启快捷选单，点选引用功能块。



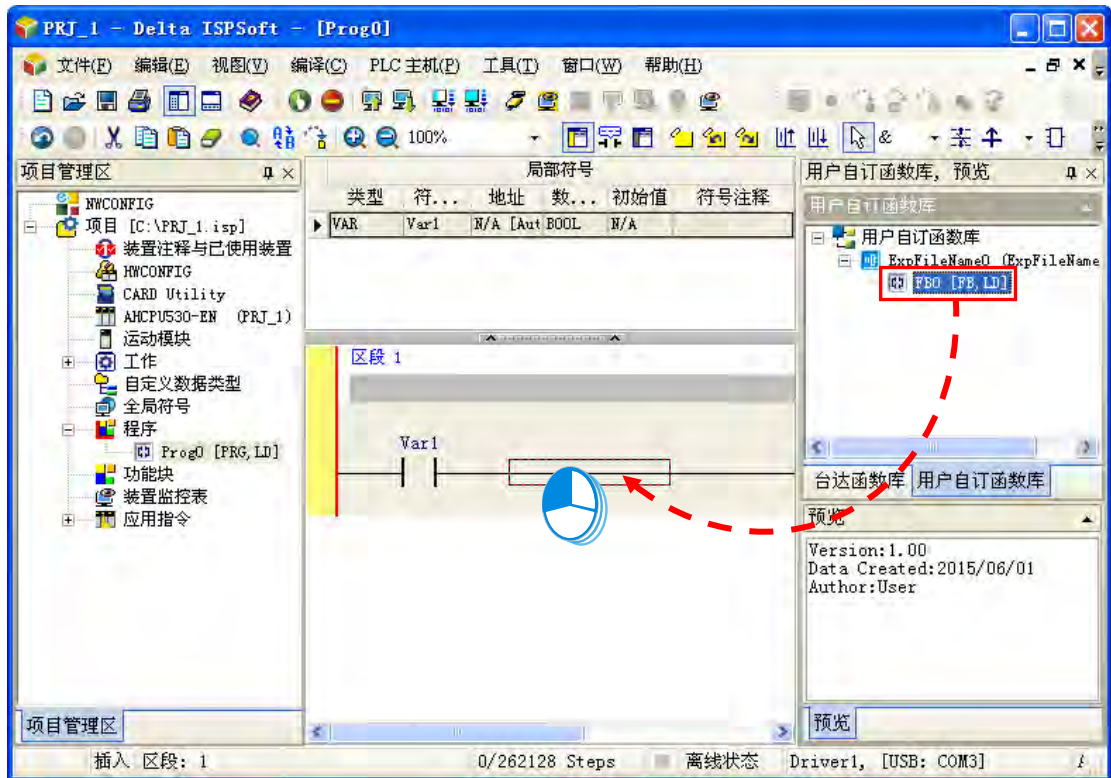
此时会出现引用功能块窗口，点选函数库的类别，从中选择要引用的函数，点选「确定」按钮。最后完成所有函数的引用后，按下「取消」按钮，关闭窗口，在项目管理区的功能块项目下，便会新增引用的函数库功能块，功能块的操作则与一般的功能块相同。

7

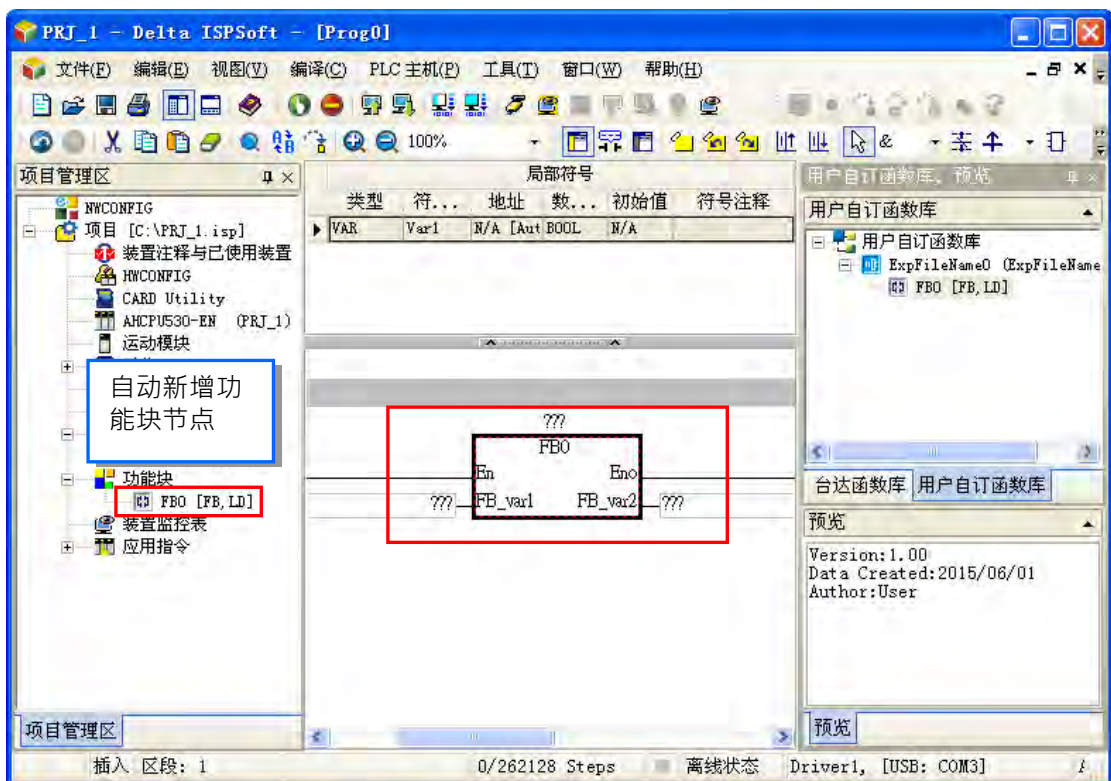


- 方法二

直接在用户自定义函数库中点选要使用的函数，按住鼠标后拖曳到 POU 中要调用的位置，便会加入到 POU 中。



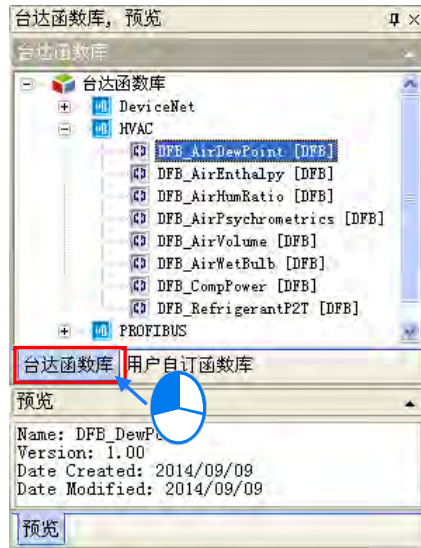
加入到 POU 中后，在项目管理区的功能块项目下，也会自动新增出此函数的功能块节点，功能块的操作则与一般的功能块相同。



7

7.5.3 引用台达函数库

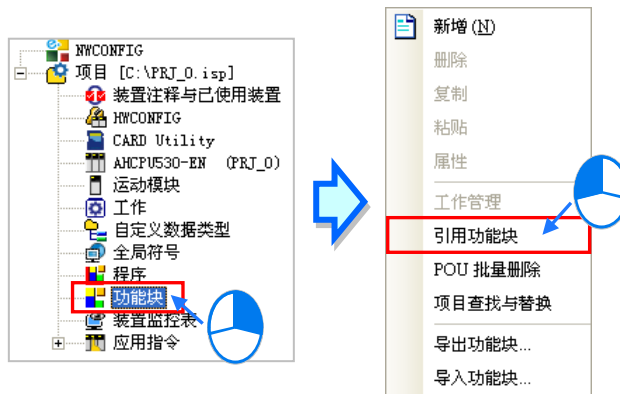
对于各种产业的应用场合，台达提供了一系列的函数库供用户引用，点选函数库区的台达函数库分页，便会分类列出目前版本所支持的台达函数库。



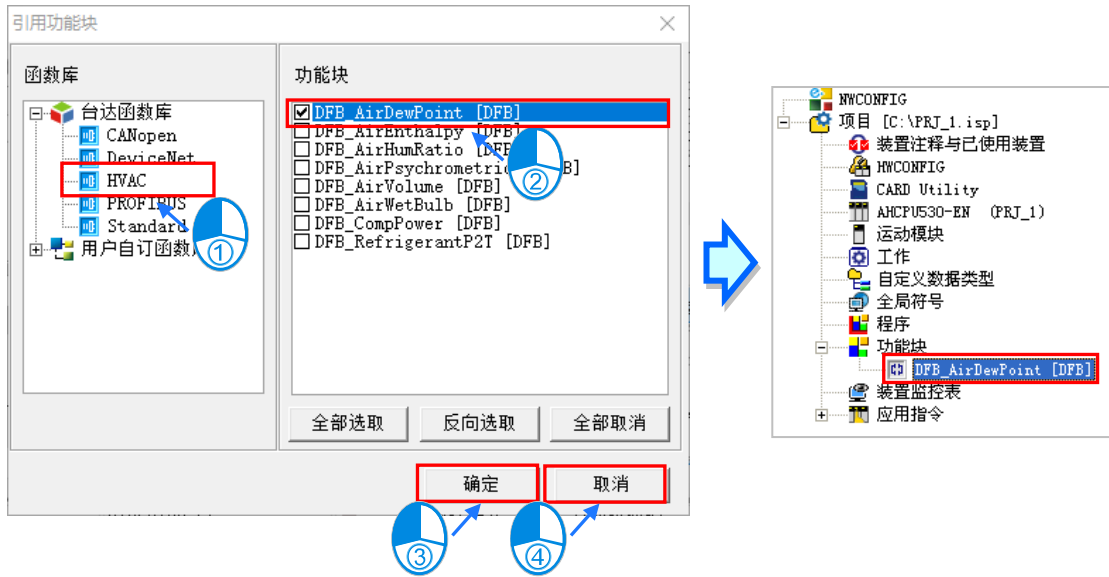
同样的，对于台达函数库有以下两种方法引用到功能区中。

- 方法一

点选功能块项目，按下鼠标右键开启快捷选单，点选「引用功能块」。

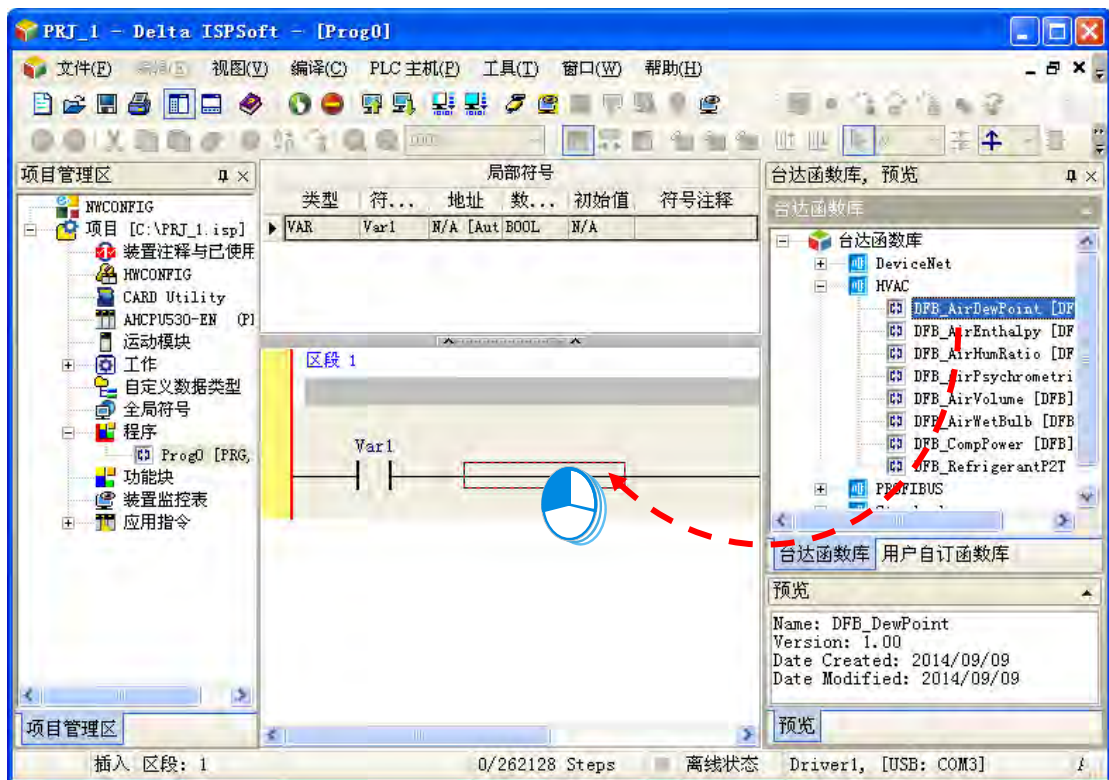


此时会出现引用功能块窗口，点选函数库的类别，从中选择要引用的函数，点选「确定」按钮。最后完成所有函数的引用后，按下「取消」按钮，关闭窗口，在项目管理区的功能块项目下，便会新增引用的函数库功能块，功能块的操作则与一般的功能块相同。

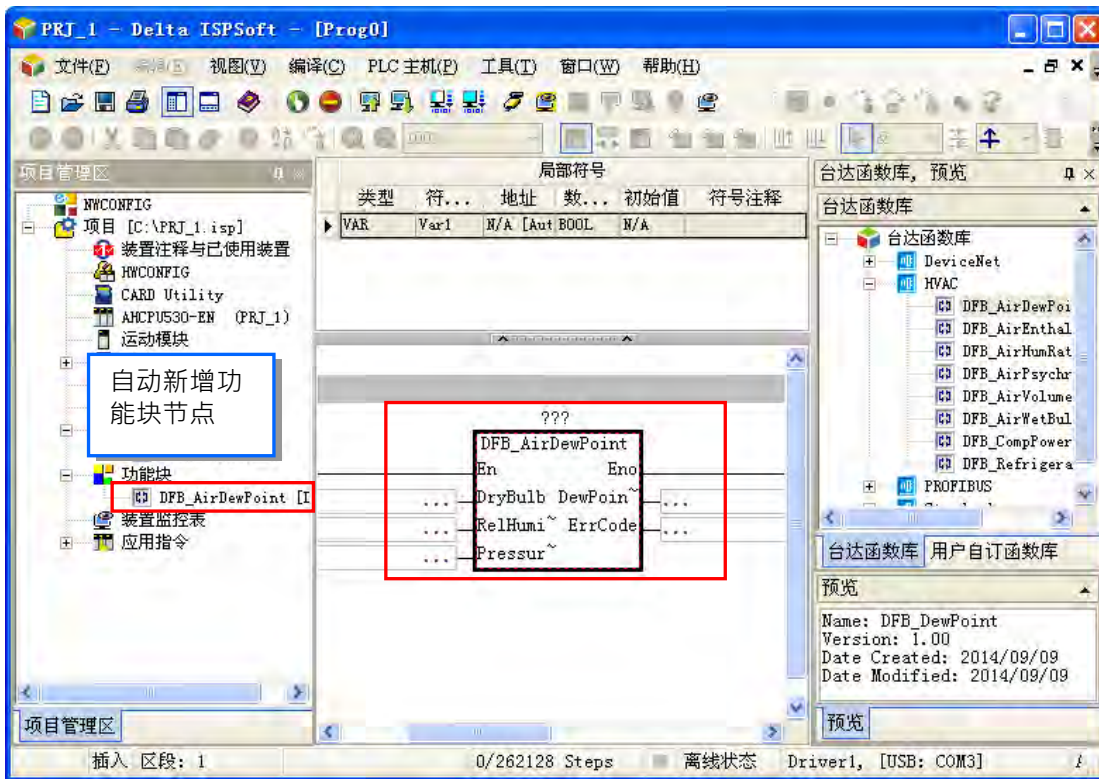


● 方法二

直接在台达函数库中点选要使用的函数，按住鼠标后拖曳到 POU 中要调用的位置，便会加入到 POU 中。



加入到 POU 中后，在项目管理区的功能块项目下，也会自动新增出此函数的功能块节点，功能块的操作则与一般的功能块相同。



*1. 台达函数库的功能块定义的程序内容，用户无法开启检视或编辑。



第8章 自定义数据类型

目录

8.1	自定义数据类型	8-2
8.2	结构 (STRUCTURE)	8-3
8.2.1	结构 (STRUCTURE) 的意义	8-3
8.2.2	建立结构 (STRUCTURE)	8-3
8.2.3	使用结构 (STRUCTURE) 类型变量符号	8-5
8.2.4	结构 (STRUCTURE) 的常见用法	8-6
8.2.5	结构 (STRUCTURE) 改变为枚举 (ENUMERATION)	8-9
8.3	枚举 (ENUMERATION)	8-10
8.3.1	枚举 (ENUMERATION) 的意义	8-10
8.3.2	建立枚举 (ENUMERATION)	8-10
8.3.3	使用枚举 (ENUMERATION) 类型变量符号	8-12
8.3.4	枚举 (ENUMERATION) 的常见用法	8-13
8.3.5	枚举 (ENUMERATION) 改变为结构 (STRUCTURE)	8-15
8.4	联合 (UNION)	8-16
8.4.1	联合 (UNION) 的意义	8-16
8.4.2	建立联合 (UNION)	8-16
8.4.3	使用联合 (UNION) 类型变量符号	8-18
8.4.4	联合 (UNION) 的常见用法	8-20
8.4.5	内建联合的支持与名称	8-22

8.1 自定义数据类型

自定义数据类型 (又称 DUT (Data Unit Type) 或 UDT (User-defined Data Type)) · 为供用户使用现有基本数据类型组合而成的数据类型。在 PLC 程序开发过程中，有时基本的数据类型仍不能满足用户程序中变量符号的使用需要。在 IEC 61131-3 中便导入了一般高级语言中的自定义数据类型观念，ISPSOft 支持下列三种自定义数据类型，同时也提高了程序的可读性及开发效率。

类型	说明
结构	结构由多个成员构成，这些成员可以有不同的数据类型。
枚举	枚举类型由多个枚举值构成，每一个枚举值都对应一个 INT 类型的常数。
联合	联合与结构相似，不同之处为联合的所有成员共享一块内存，改变一个成员的值后，所有成员的值都会改变。

各机种的结构与联合所支持的数据类型可以参考下表。

数据类型	AH/AS	AS5xx/DVPxxMC	DVP
BOOL	✓	✓	只支持 ES3
BYTE		✓	
WORD	✓	✓	只支持 ES3
DWORD	✓	✓	只支持 ES3
LWORD	只支持 AH	✓	
SINT		✓	
INT	✓	✓	只支持 ES3
DINT	✓	✓	只支持 ES3
LINT	只支持 AH	✓	
USINT		✓	
UINT		✓	
UDINT		✓	
ULINT		✓	
REAL	✓	✓	只支持 ES3
LREAL	只支持 AH	✓	
STRING	不支持 ARRAY	✓	
STEP			
COUNTER			
TIMER			
TIME		✓	
DATE		✓	
TOD		✓	
DT		✓	

8.2 结构 (STRUCTURE)

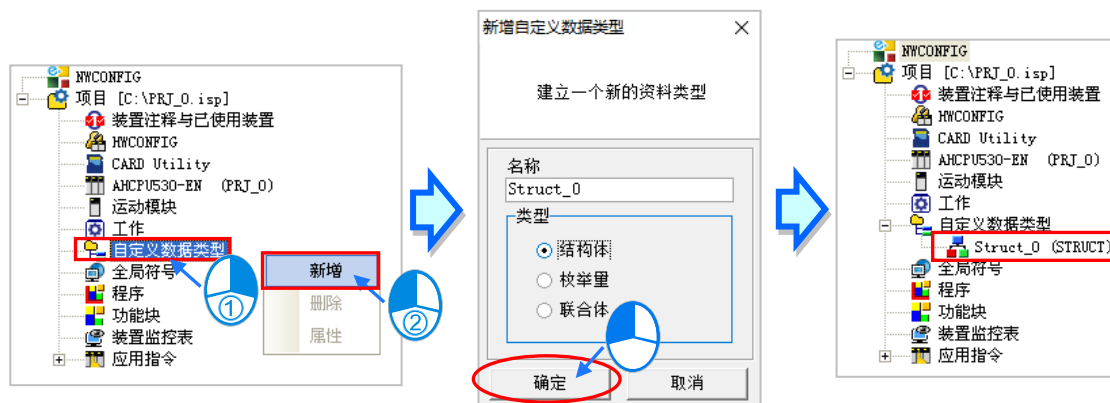
8.2.1 结构 (STRUCTURE) 的意义

结构 (STRUCTURE) 是由多个各种数据类型的成员组成，如下图，是由四个成员组成的结构所占用的内存空间大小共为 7 个 WORD。而结构可以做为另一结构的成员。结构可用于定义分组数据以及作为一个数据单元来传送数值。

STRUCTURE		数值内容																															
成员名称	数据类型	WORD 1														WORD 0																	
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Str1	ARRAY [0..31] OF BOOL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1
Str2	WORD															12,039 (16#2F07)																	
Str3	ARRAY [0..1] OF WORD	32,769(16#8001)														12,039 (16#2F07)																	
Str4	DWORD	2,147,561,223(16#80012F07)																															

8.2.2 建立结构 (STRUCTURE)

在项目管理区的「自定义数据类型」项目点击鼠标右键，单击「新增」，出现新增自定义数据类型窗口，输入「名称」，选择类型为「结构」后，按下「确定」按钮，即会于自定义数据类型项目下新增此项目。



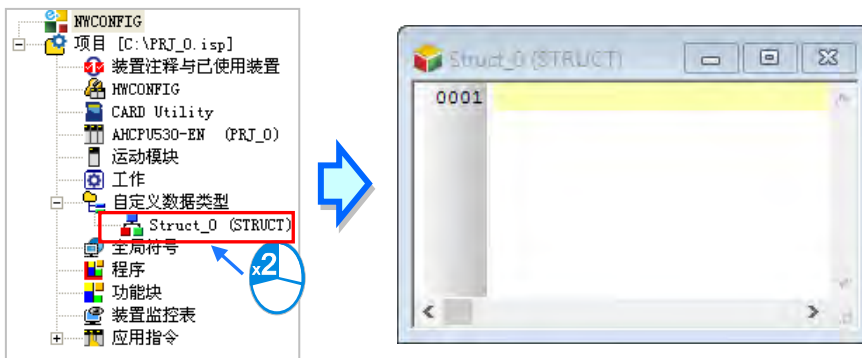
* 结构的名称命名限制请参阅下方结构名称限制。

一个结构项目，只能宣告出一个结构 (STRUCTURE)，新增完成后会随即在编辑工作区开启结构化编辑窗口，并自动带出注释与模板供用户参考修改。

```

Structure_0 (STRUCT)
0001 *****
0002 ** Example1:
0003 **
0004 ** TYPE DUT_StructA :
0005 ** STRUCT
0006 **   Element1 : BOOL;
0007 **   Element2 : REAL;
0008 **
0009 **   (Array notation: ARRAY [lowerbound_index..upperbound_index] OF DataType)
0010 **   (Hint: the max dimension of ARRAY is 3)
0011 **   Element3 : ARRAY [0..1][0..3][0..5] OF INT;
0012 ** END_STRUCT
0013 ** END_TYPE
0014 **
0015 ** Example2:
0016 **
0017 ** TYPE DUT_StructB :
0018 ** STRUCT
0019 **   Element1 : DUT_EnumA; (Hint: using Enum in Struct declaration)
0020 **   Element2 : ARRAY [0..7] OF DUT_StructA;
0021 ** END_STRUCT
0022 ** END_TYPE
0023 *****
0024
0025 TYPE Structure_0 :
0026 STRUCT
0027   Element1 : BOOL;
0028 END_STRUCT
0029 END_TYPE
    
```

之后也可在项目管理区的此结构项目上双击鼠标左键再次开启编辑窗口。



结构的格式如下，蓝色字体为系统关键词，黑色字体为用户自定义的名称与成员名称。

TYPE <自定义名称>:

STRUCT

<成员 1 名称>: <数据类型>;

<成员 2 名称>: <数据类型>;

...

<成员 n 名称>: <数据类型>;

END_STRUCT

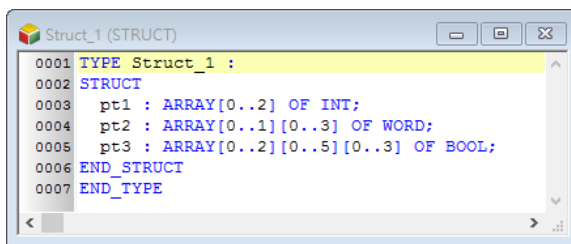
END_TYPE

```

Struct_0 (STRUCT)
0001 TYPE Struct_0 :
0002 STRUCT
0003   pt1 : BOOL;
0004   pt2 : WORD;
0005 END_STRUCT
0006 END_TYPE
    
```

- **自定义名称**：预设带入的自定义名称，不能修改为其它名称否则编译会报错。
- **成员名称**：定义成员的称。
- **数据类型**：定义成员的数据类型，可为 **BOOL**、**WORD**、**DWORD**、**LWORD**、**INT**、**DINT**、**LINT**、**REAL**、**LREAL**、**STRING**、**ARRAY** 或另一 **STRUCTURE** 或 **ENUMERATION**。若选择 **DVPxxMC**

或 AS5xx 机种，则可支持更多数据类型如 BYTE、SINT、USINT、UINT、UDINT、ULINT、TIME、DATE、TOD、DT、STRING。当成员数据类型为 ARRAY 时格式为 **ARRAY [0..n] OF <数据类型>;**，n 最大为 65535，若选择 DVPxxMC 或 AS5xx 机种，则最大为 2147483647。若成员数据类型为 STRUCTURE 时，直接写入另一 STRUCTURE 的名称。ARRAY 中的数据类型无法再输入 ARRAY，若要定义多维数组，则最多支持三维数组，三维数组格式为 **ARRAY [0..n] [0..n] [0..n] OF <数据类型>;**，二维数组格式为 **ARRAY [0..n] [0..n] OF <数据类型>;**，如下图。

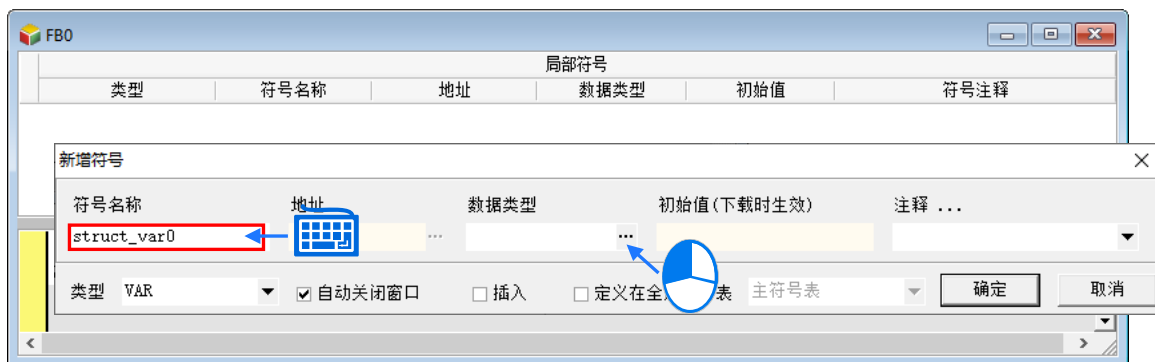


自定义名称与成员名称使用时有其一定的规则与限制，说明如下。

- 自定义名称不可与其它自定义数据类型名称重复。
- 同一个自定义中不允许重复的成员名称。
- 自定义名称与成员名称不可超过 30 个字符，且须注意一个中文字会占用两个字符。
- 不可使用系统保留的关键词，如指令名称、装置名称，或是其它在系统中已被赋予特殊意义的保留字，但若只是被包含在名称当中则是合法的，如“MO”是不合法的名称，但“_MO”则是合法的。
- 自定义名称与成员名称当中不可有空白，例如“INPUT CH0”便是不合法的名称。
- 自定义名称与成员名称当中可使用底线，但不可连续使用，或是置于结尾，例如“INPUT_CH0”是合法的名称，但“INPUT__CH0”与“INPUT_CH0_”则是不合法的名称。
- 自定义名称与成员名称不可使用特殊字符，例如 *、#、?、\、%、@ ...等。

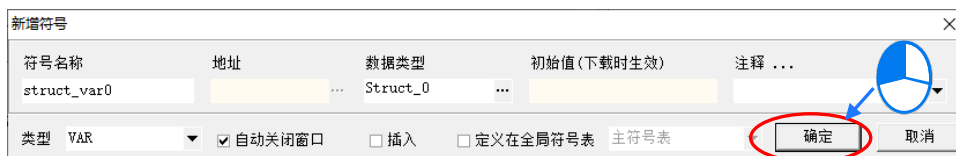
8.2.3 使用结构 (STRUCTURE) 类型变量符号

完成结构类型的建立，再执行检查或编译后，在符号表中便可宣告此结构类型的变量，在全局或区域符号表开启新增符号窗口，输入符号名称后单击数据类型右侧...按钮，如下图。



8

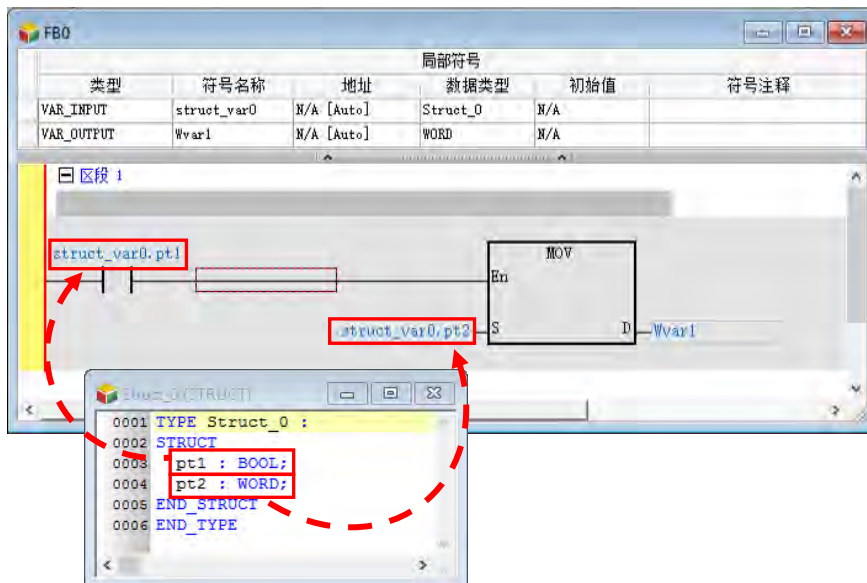
在数据类型窗口单击「用户自定义类型」，在下方字段单击已宣告的结构数据类型。按下「确定」后完成新建符号。



8

8.2.4 结构 (STRUCTURE) 的常见用法

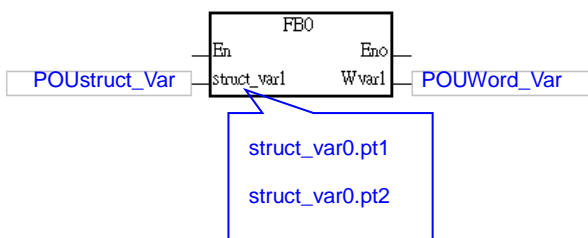
在完成宣告结构与变量符号后，便可于 POU 中使用。如下图，功能块中变量符号 **struct_var0** 的数据类型为 **Struct_0**。在程序中使用此结构变量符号的成员，可用“.”的方式，例如下图中 **struct_var0.pt1** 为使用此结构变量符号的第一个成员 **pt1** 作为此接点的 **BOOL** 数据类型操作数；而 **struct_var0.pt2** 为使用此结构变量符号的第二个成员 **pt2** 作为此指令接脚的 **WORD** 数据类型操作数。



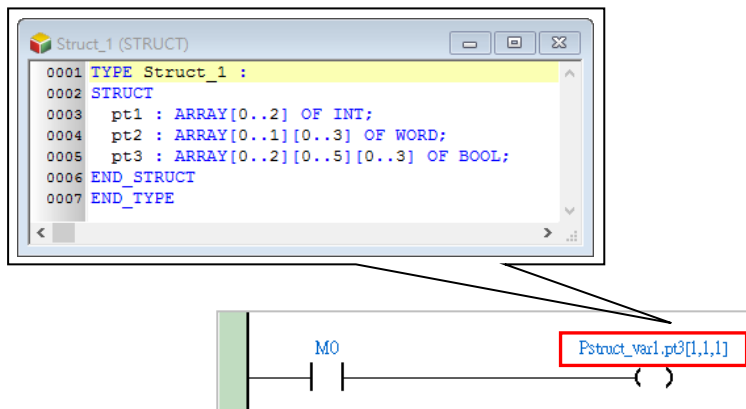
若在 POU 中使用上述功能块，为了对应功能块中 Struct_0 数据类型的输入与输出接脚，也必须在此 POU 中宣告 Struct_0 数据类型的变量符号作为操作数，如下图 POUstruct_var，同样结构的成员也可用”.”的方式使用。



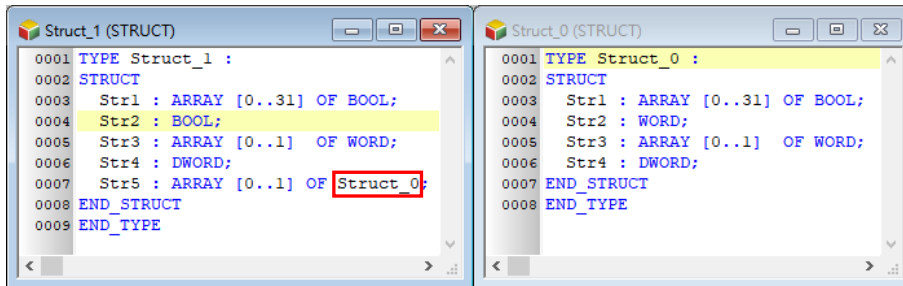
由上述说明可知，当结构变量符号 POUstruct_Var 输入到 FBO 时，便同时传递了此变量符号内所有的成员，供 FBO 中的 struct_var0.pt1 与 struct_var0.pt2 运算。因此用户在开发功能块时，可将结构变量定义为功能块输入或输出接脚，便可同时传递此变量内的所有成员，藉此节省了一一定义接脚的动作，同时也让程序更为简洁



当结构的内容包含数组成员时，使用此类结构的变量符号可以以下形式书写，如下图中将 M0 的状态写入自定义数据类型变量符号 Pstruct_var1 中指定成员 pt3 中三维数组的第[1,1,1]个。



另一个例子为结构可以当成另一个结构的成员，如下图，结构 Struct_1 内包含 Struct_0 的成员。



其中 Struct_1 的所占用的内存大小如下表所示

8

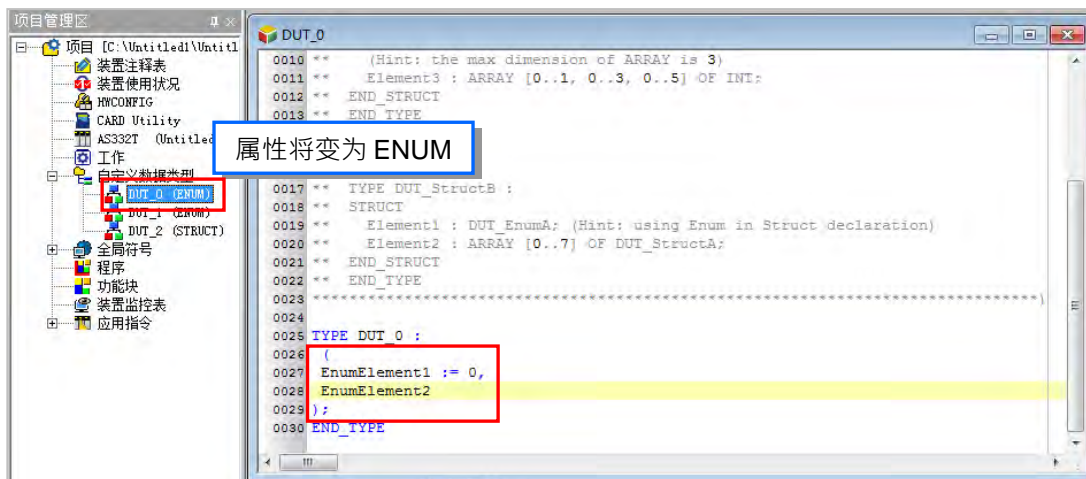
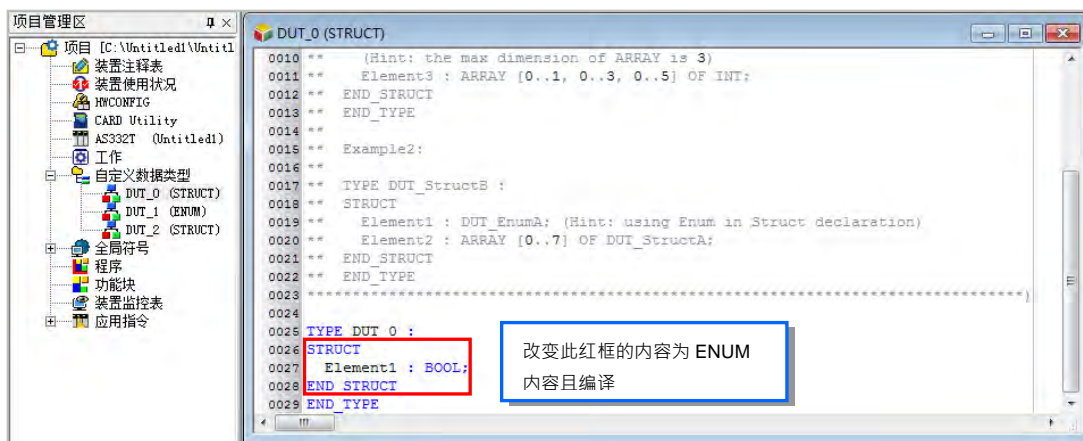
自定义名称	成员名称	数据类型	占用内存(Bit)
Struct_1	Str1	ARRAY [0..31] OF BOOL	32
	Str2	BOOL	1
	Str3	ARRAY [0..1] OF WORD	4
	Str4	DWORD	32
	Str5	ARRAY [0..1] OF Struct_0	8

而上图红色框内为一个 Struct_0 成员所占的内存大小为 7 个 WORD，如下图。

		Struct_0																																
成员名称	数据类型	数值内容																																
		WORD 1																WORD 0																
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Str1	ARRAY [0..31] OF BOOL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	1	1
Str2	WORD																	12,039 (16#2F07)																
Str3	ARRAY [0..1] OF WORD	32,769(16#8001)																12,039 (16#2F07)																
Str4	DWORD	2,147,561,223(16#80012F07)																																

8.2.5 结构 (STRUCTURE) 改变为枚举 (ENUMERATION)

当用户想将自定义数据类型「结构」改成「枚举」，可以将 STRUCT 与 END_STRUCT 改成“(”以及“);” 此时刻再将成员内容改成 ENUM 类型 在执行编译 则自定义数据类型的属性将由(STRUCT) 变为 (ENUM)。ENUM 的设定内容请参考第 8.3 节。



8

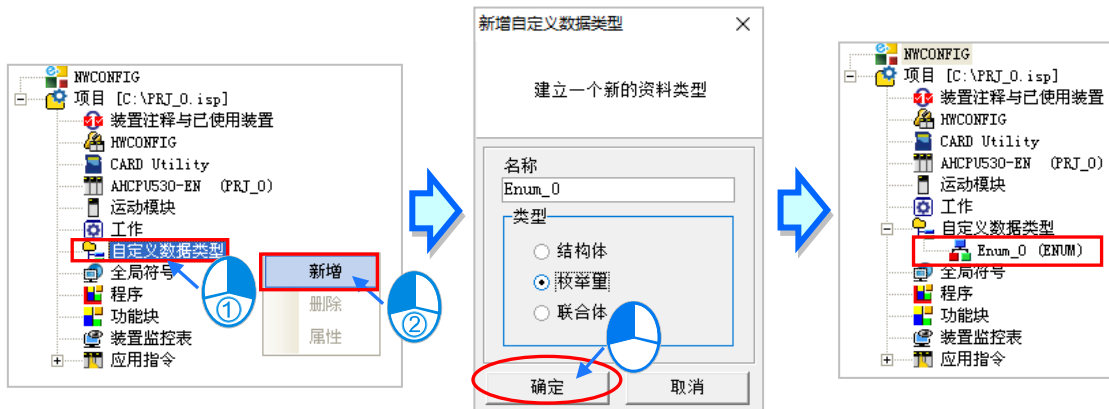
8.3 列举 (ENUMERATION)

8.3.1 列举 (ENUMERATION) 的意义

列举 (ENUMERATION) 用于定义一组字符串别名以及对应的数值，便于帮助用户记忆。

8.3.2 建立列举 (ENUMERATION)

项目管理区的「自定义数据类型」项目点击鼠标右键，单击「新增」，出现新增自定义数据类型窗口，输入名称，选择类型为列举后，按下「确定」按钮，即会于自定义数据类型项目下新增此项目。



*.列举的名称命名限制请参阅下方列举名称限制。

一个列举项，只能宣告出一个列举 (ENUMERATION) 新增完成后会随即在编辑工作区开启列举编辑窗口，并自动带出注释与模板供用户参考修改。

```

Enum_0 (ENUM)
0001 |-----
0002 ** Example1:
0003 **
0004 ** TYPE DUT_EnumA :
0005 ** (
0006 **   Red := 0,
0007 **   Blue := 2,
0008 **   Green := 3
0009 ** );
0010 ** END_TYPE
0011 **
0012 ** Example2:
0013 **
0014 ** TYPE DUT_EnumB :
0015 ** (
0016 **   Red := 0, (Hint: allowable to set or not to set the Enum value)
0017 **   Blue,
0018 **   Green
0019 ** );
0020 ** END_TYPE
0021 |-----
0022
0023 TYPE Enum_0 :
0024 (
0025   EnumElement1 := 0,
0026   EnumElement2
0027 );
0028 END_TYPE

```

之后也可在项目管理区的此列举项目上双击鼠标左键再次开启编辑窗口。

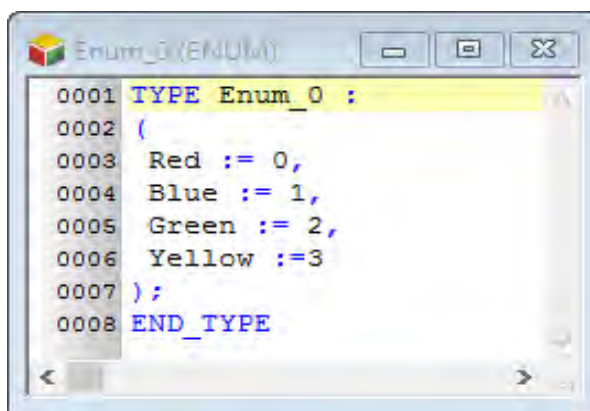


列举的格式如下，蓝色字体为系统关键词，黑色字体为用户定义自定义的名称、成员名称及数值。

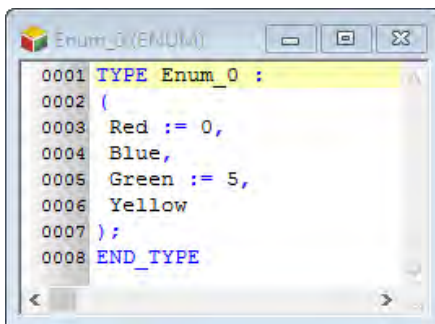
TYPE <自定义名称>:

```
(
  <成员 1 名称> := <数值 1>,
  <成员 2 名称> := <数值 2>,
  ...
  <成员 n 名称> := <数值 n>
```

);
END_TYPE



- **自定义名称**：预设带入的自定义名称，不能修改为其它名称否则编译会报错。
- **成员名称**：定义成员的名称。
- **数值**：定义每个成员的数值，范围可为-32768~32767，也可空白不输入，系统将会根据前一成员的值依序自行带入默认值，且避开已使用的值。如下图，系统将会自动设定成员 **Blue** 的值为 1，而 **Yellow** 的值为 6。



自定义名称与成员名称使用时有其一定的规则与限制，说明如下。

- 自定义名称不可与其它自定义数据类型名称重复。
- 同一个自定义中不允许重复的成员名称。
- 自定义名称与成员名称不可超过 30 个字符，且须注意一个中文字会占用两个字符。

- 不可使用系统保留的关键词，如指令名称、装置名称，或是其它在系统中已被赋予特殊意义的保留字，但若只是被包含在名称当中则是合法的，如“MO”是不合法的名称，但“_MO”则是合法的。
- 自定义名称与成员名称当中不可有空白，例如“INPUT CH0”便是不合法的名称。
- 自定义名称与成员名称当中可使用底线，但不可连续使用，或是置于结尾，例如“INPUT_CH0”是合法的名称，但“INPUT__CH0”与“INPUT_CH0_”则是不合法的名称。
- 自定义名称与成员名称不可使用特殊字符，例如 *、#、?、\、%、@ ...等。
- 列举数值为 INT 的数据类型。

8.3.3 使用列举 (ENUMERATION) 类型变量符号

完成列举类型的建立，再执行**检查**或**编译**后，在符号表中便可宣告此列举类型的变量，如下图于全局或区域符号表开启新建符号窗口，输入**符号名称**后单击数据类型右侧...按钮。



数据类型窗口单击**用户自定义类型**，再于下方字段单击已宣告的列举数据类型。按下「**确定**」后完成新建符号。

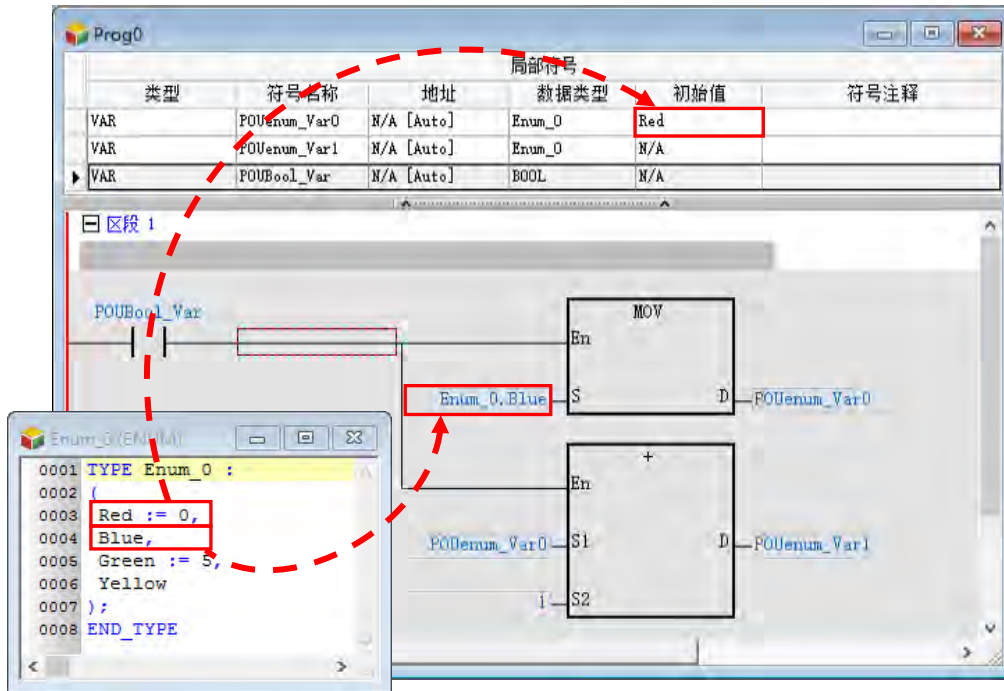


8.3.4 列举 (ENUMERATION) 的常见用法

在完成宣告列举与变量符号后，便可于 POU 中使用。如下图，变量符号 **POUenum_Var0** 与 **POUenum_Var1** 的数据类型为 Enum_0，在符号表中可以成员名称或数值定义初始值，如下图的 **Red**。

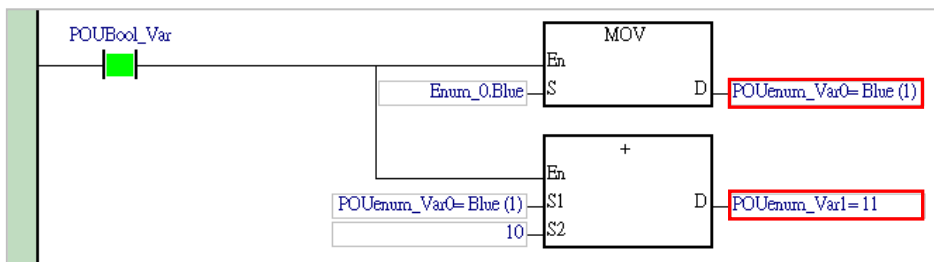
程序中可用“.”的方式指定已定义的列举中某一成员，如下图的 **Enum_0.Blue**，并将其指定到一个 Enum_0 类型变量 **POUenum_Var0** 中 (以 MOV 指令)，相当在将定义的成员数值写入列举变量中，因此 MOV 指令的结果便会使 **POUenum_Var0** 的值为 1。

8



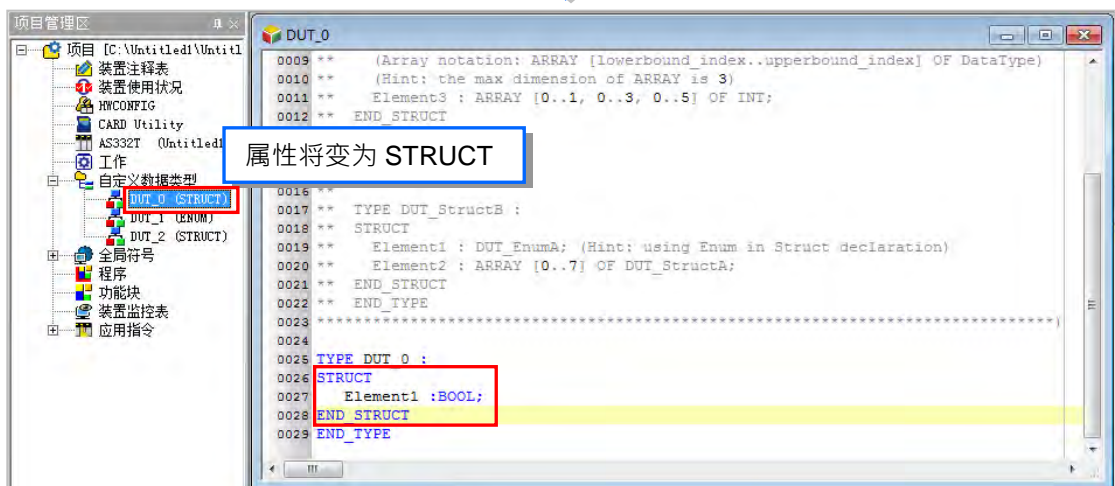
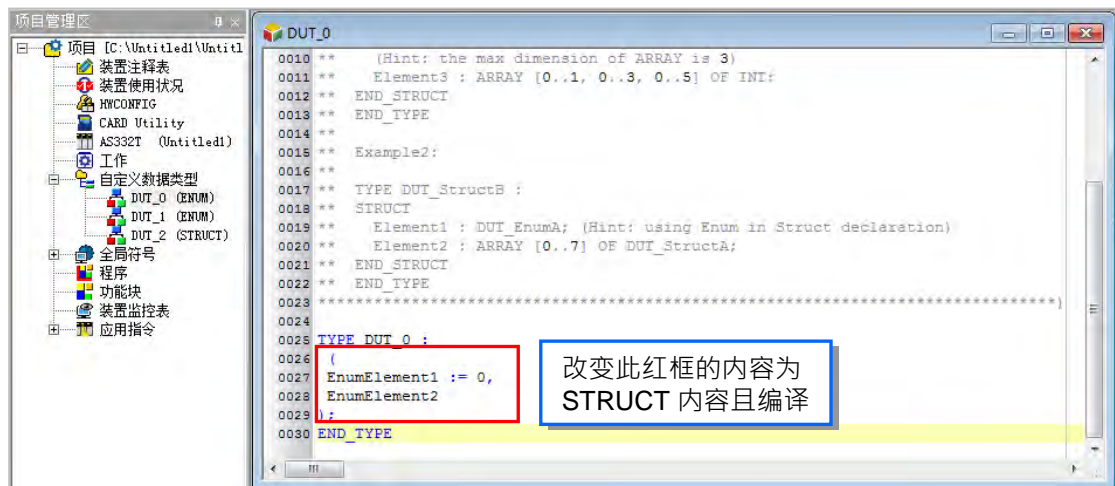
另外也可以对枚举变量符号进行数值运算，如上图以+指令对 **POUenum_Var0** 的数值进行+1，并将结果写入到 **POUenum_Var1**，因为 **POUenum_Var0** 的数值为 1，故 **POUenum_Var1** 的执行结果将为 2，即为定义的成员 **Green**。

因此在程序实际运行并在线监控时，所有的枚举变量符号便会显示当前数值所定义的成员名称，如下图的 **POUenum_Var0=Blue(1)**。需注意的是，若枚举变量符号当前的数值未在定义的成员范围内，则会直接显示此数值，如下图的 **POUenum_Var1=11** 因为数值 11 未在枚举中定义。



8.3.5 枚举 (ENUMERATION) 改变为结构 (STRUCTURE)

当用户想将自定义数据类型「枚举」改成「结构」，可以将“ (”与“);”改成 **STRUCT** 以及 **END_STRUCT**，此时刻再将成员内容改成 **STRUCT** 类型，再执行编译，则自定义数据类型的属性将由 (EUM) 变为 (STRUCT)，STRUCT 的设定内容请参考第 8.2 节。



8.4 联合 (UNION)

8.4.1 联合 (UNION) 的意义

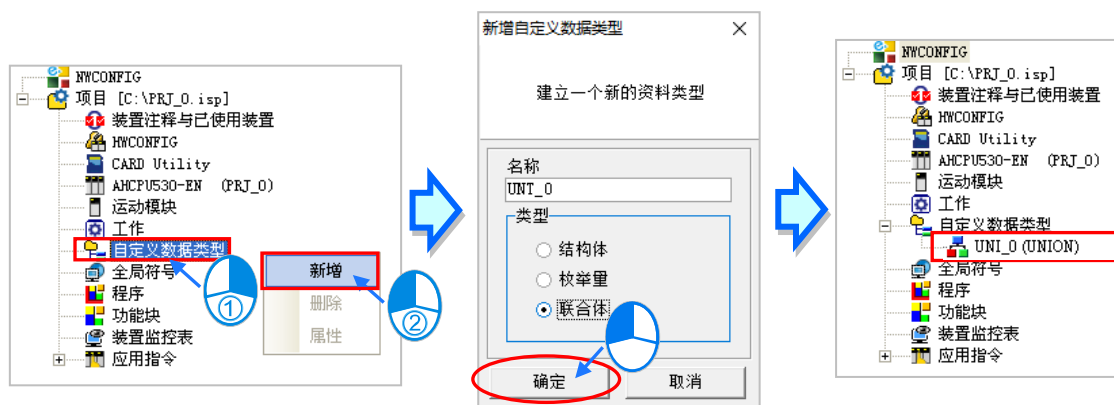
联合与结构相似，不同之处为联合的所有成员共享一块内存且一个自定义联合的最大值，取决于联合中的最大成员，如下图联合是由四个成员组成其所占用的内存为 32 个 Bit (2 个 Word)，改变一个成员的数值后，所有成员的数值都会跟着改变。

		UNION																																
成员名称	数据类型	数值内容																																
		WORD 1																WORD 0																
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
uni1	ARRAY [0..31] OF BOOL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1	1	1
uni2	WORD																	12,039 (16#2F07)																
uni3	ARRAY [0..1] OF WORD	32,769(16#8001)																12,039 (16#2F07)																
uni4	DWORD	2,147,561,223(16#80012F07)																																

8.4.2 建立联合 (UNION)

于项目管理区的「自定义数据类型」项目点击鼠标右键，点选「新增」，出现新增自定义数据类型窗口，输入「名称」，选择类型为「联合」后，按下「确定」按钮，即会于自定义数据类型项目下新增此项目。

8



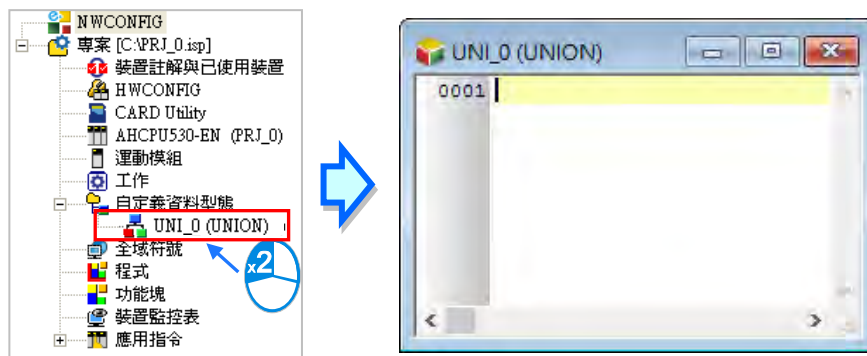
*.联合的名称命名限制请参阅下方联合名称限制。

一个联合项目，只能宣告出一个联合 (UNION)，新增完成后会随即在编辑工作区开启联合编辑窗口，并自动带出注释与模板供用户参考修改。

```

UNI_0 (UNION)
0001 *****
0002 ** Example1:
0003 **
0004 ** TYPE DUT_UnionA :
0005 ** UNION
0006 **   Element1 : BOOL;
0007 **   Element2 : REAL;
0008 **
0009 **   (Array notation: ARRAY [lowerbound_index..upperbound_index] OF DataType)
0010 **   (Hint: the max dimension of ARRAY is 3)
0011 **   Element3 : ARRAY [0..1, 0..3, 0..5] OF INT;
0012 ** END_UNION
0013 ** END_TYPE
0014 **
0015 ** Example2:
0016 **
0017 ** TYPE DUT_UnionB :
0018 ** UNION
0019 **   Element1 : DUT EnumA; (Hint: using Enum in Union declaration)
0020 **   Element2 : ARRAY [0..7] OF DUT_UnionA;
0021 ** END_UNION
0022 ** END_TYPE
0023 *****
0024
0025 TYPE UNI_0 :
0026 UNION
0027   Element1 : BOOL;
0028 END_UNION
0029 END_TYPE
    
```

之后也可于项目管理区的此联合项目上双击鼠标左键再次开启编辑窗口。



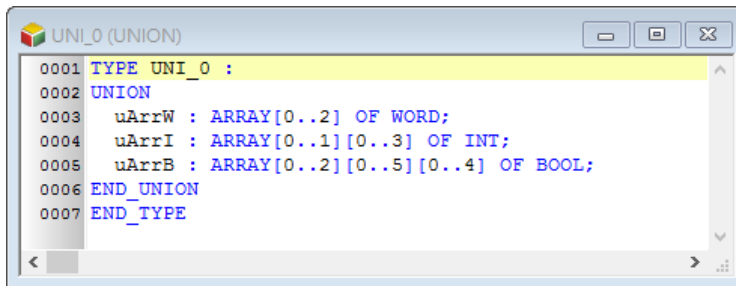
联合的格式如下，蓝色字体为系统关键词，黑色字体为用户自定义的名称、成员名称及数值。

TYPE <自定义名称>:
UNION
 <成员 1 名称> := <数据类型>;
 <成员 2 名称> := <数据类型>;
 ...
 <成员 n 名称> := <数据类型>;
END_UNION
END_TYPE

```

UNI_0 (UNION)
0001 TYPE UNI_0 :
0002 UNION
0003   uArrB : ARRAY[0..15] OF BOOL;
0004   uW    : WORD;
0005   uI    : INT;
0006 END_UNION
0007 END_TYPE
    
```


- **自定义名称**：预设带入的自定义名称，不能修改为其它名称否则编译会报错。
- **成员名称**：定义成员的名称。
- **数据类型**：定义成员的数据类型，可为 `BOOL`、`WORD`、`DWORD`、`LWORD`、`INT`、`DINT`、`LINT`、`REAL`、`LREAL`、`ARRAY` 或另一 `STRUCTURE` 或 `ENUMERATION`。若选择 `DVPxxMC` 或 `AS5xx` 机种，则可支持更多数据类型如 `BYTE`、`SINT`、`USINT`、`UINT`、`UDINT`、`ULINT`、`TIME`、`DATE`、`TOD`、`DT`、`STRING`。当成员数据类型为 `ARRAY` 时格式为 **`ARRAY [0..n] OF <数据类型>`**；`n` 最大为 65535，若选择 `DVPxxMC` 或 `AS5xx` 机种，则最大为 2147483647。若成员数据类型为 `STRUCTURE` 时，直接写入另一 `STRUCTURE` 的名称。`ARRAY` 中的数据类型无法再输入 `ARRAY`，若要定义多维数组，则最多支持三维数组，三维数组格式为 **`ARRAY [0..n] [0..n] [0..n] OF <数据类型>`**；，二维数组格式为 **`ARRAY [0..n] [0..n] OF <数据类型>`**；，如下图。



```

0001 TYPE UNI_0 :
0002 UNION
0003   uArrW : ARRAY[0..2] OF WORD;
0004   uArrI : ARRAY[0..1][0..3] OF INT;
0005   uArrB : ARRAY[0..2][0..5][0..4] OF BOOL;
0006 END_UNION
0007 END_TYPE

```

自定义名称与成员名称使用时有其一定的规则与限制，说明如下。

- 自定义名称不可与其它自定义数据类型名称重复。
- 同一个自定义中不允许重复的成员名称。
- 自定义名称与成员名称不可超过 30 个字符，且须注意一个中文字会占用两个字符。
- 不可使用系统保留的关键词，如指令名称、装置名称，或是其它在系统中已被赋予特殊意义的保留字，但若只是被包含在名称当中则是合法的，如 **“MO”** 是不合法的名称，但 **“_MO”** 则是合法的。
- 自定义名称与成员名称当中不可有空白，例如 **“INPUT CH0”** 便是不合法的名称。
- 自定义名称与成员名称当中可使用底线，但不可连续使用，或是置于结尾，例如 **“INPUT_CH0”** 是合法的名称，但 **“INPUT__CH0”** 与 **“INPUT_CH0_”** 则是不合法的名称。
- 自定义名称与成员名称不可使用特殊字符，例如 `*`、`#`、`?`、`\`、`%`、`@` ... 等。

8

8.4.3 使用联合 (UNION) 类型变量符号

完成联合类型的建立，再执行**检查**或**编译**后，在符号表中便可宣告此联合类型的变量，于全局或区域符号表开启新增符号窗口，输入**符号名称**后点选数据类型右侧 **...** 按钮，如下图。



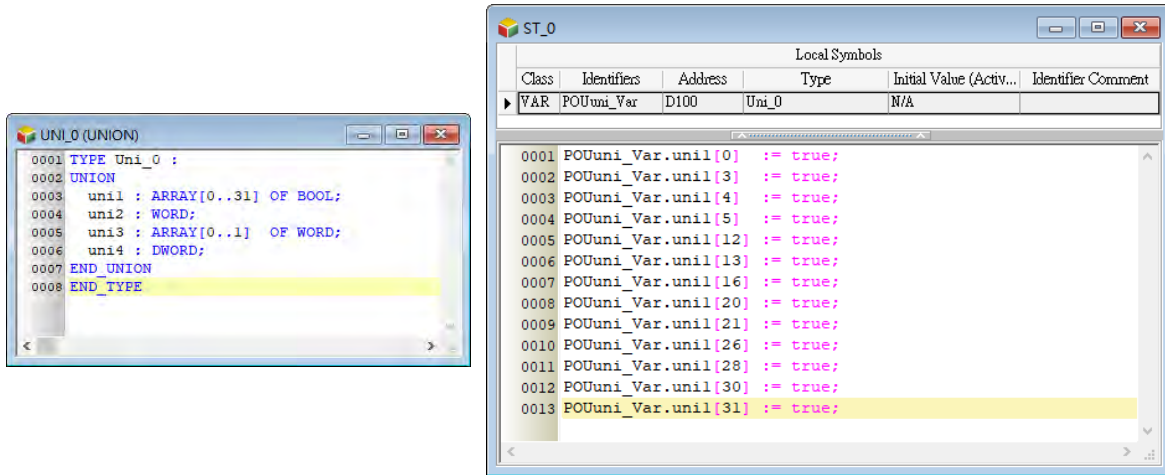
于数据类型窗口点选**用户自定义类型**，再于下方字段点选已宣告的枚举数据类型。按下「**确定**」后完成新建符号。



8

8.4.4 联合 (UNION) 的常见用法

下列范例建立一个联合里面包含 uni1 到 uni4 四个成员，接着将变量符号 **POUuni_Var** 数据类型宣告为 Uni_0，然后使用程序改变 **POUuni_Var.uni1** 的数值，如下图。



再透过监控表查看 **POUuni_Var.uni2** 与其它成员的数值是否改变，如下图。

来源	符号名称	装置名称	状态	数据类型	值 (16位)	值 (32位)	数值类型
ST_0	POUuni_Var.uni2	D100	<input type="checkbox"/>	WORD	12345	3559993401	自动检测
ST_0	POUuni_Var.uni3[0]	D100	<input type="checkbox"/>	WORD	12345	3559993401	自动检测
ST_0	POUuni_Var.uni3[1]	D101	<input type="checkbox"/>	WORD	54321	54321	自动检测
I ST_0	POUuni_Var.uni4	D100	<input type="checkbox"/>	DWORD	12345	3559993401	自动检测

由下表数值可以看出成员 uni1 到 uni4 所对应的数值。

		POUuni_Var																																			
成员名称	数据类型	数值内容																																			
		WORD 1																WORD 0																			
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0				
uni1	ARRAY [0..31] OF BOOL	1	1	0	1	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1
uni2	WORD																	12,345																			
uni3	ARRAY [0..1] OF WORD	54,321																12,345																			
uni4	DWORD	3,559,993,401																																			

8

第二个范例改变 **POUuni_Var.uni4** 的数值，如下图。



再从监控表观察其它成员的数值，所以改变一个成员的数值内容后，所有成员的数值都会随之改变。

来源	符号名称	装置名称	状态	数据类型	值 (16位)	值 (32位)	数值类型
ST_1	POUuni_Var.uni1[0]	D100.0	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[1]	D100.1	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[2]	D100.2	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[3]	D100.3	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[4]	D100.4	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[5]	D100.5	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[6]	D100.6	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[7]	D100.7	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[8]	D100.8	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[9]	D100.9	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[10]	D100.10	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[11]	D100.11	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[12]	D100.12	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[13]	D100.13	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[14]	D100.14	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[15]	D100.15	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[16]	D101.0	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[17]	D101.1	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[18]	D101.2	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[19]	D101.3	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[20]	D101.4	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[21]	D101.5	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[22]	D101.6	<input checked="" type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[23]	D101.7	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[24]	D101.8	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[25]	D101.9	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[26]	D101.10	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[27]	D101.11	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[28]	D101.12	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[29]	D101.13	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[30]	D101.14	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni1[31]	D101.15	<input type="checkbox"/>	BOOL			
ST_1	POUuni_Var.uni2	D100	<input type="checkbox"/>	WORD	23970	5201314	自动检测
ST_1	POUuni_Var.uni3[0]	D100	<input type="checkbox"/>	WORD	23970	5201314	自动检测
ST_1	POUuni_Var.uni3[1]	D101	<input type="checkbox"/>	WORD	79	79	自动检测

由下表数值可以看出成员 uni1 到 uni4 所对应的数值。

POUni_Var																																	
成员名称	数据类型	数值内容																															
		WORD 1													WORD 0																		
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
uni1	ARRAY [0..31] OF BOOL	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1	0
uni2	WORD														23,970																		
uni3	ARRAY [0..1] OF WORD	79													23,970																		
uni4	DWORD	5,201,314																															

8.4.5 内建联合的支持与名称

部分机种在 ISPSOft 内部有支持内建的联合供用户使用，请参考下表。

机种 联合	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
Union16	V	V	V	V	V		
Union32	V	V	V	V	V		
Union64		V	V	V	V		

下表为内建的联合所包含的成员及数据类型，用户可于宣告符号时，在用户自定义数据类型选取。

联合 成员	Union16	Union32	Union64
uArrB	ARRAY [0..15] OF BOOL	ARRAY [0..31] OF BOOL	ARRAY [0..63] OF BOOL
uW	WORD		
uI	INT		
uArrW		ARRAY [0..1] OF WORD	ARRAY [0..3] OF WORD
uArrI		ARRAY [0..1] OF INT	ARRAY [0..3] OF INT
uDW		DWORD	
uDI		DINT	
uR		REAL	
uArrDW			ARRAY [0..1] OF DWORD
uArrDI			ARRAY [0..1] OF DINT
uArrR			ARRAY [0..1] OF REAL
uLW			LWORD
uLI			LINT
uLR			LREAL



第9章 轴

目录

9.1 轴.....	9-2
9.1.1 轴的意义.....	9-2
9.1.2 建立轴.....	9-2
9.1.2.1 AHxxEMS 机种的建立轴.....	9-2
9.1.2.2 DVPxxMC/AS5xx 机种建立轴与轴基本设定.....	9-5
9.1.3 全局符号表的主符号表与轴符号表	9-9
9.2 轴的参数设定、监控与调试	9-10
9.2.1 轴参数设定	9-10
9.2.1.1 AHxxEMC 机种的轴参数设定	9-10
9.2.1.2 DVPXXMC/AS5xx 机种的轴参数设定.....	9-13
9.2.2 监控与调试	9-20
9.2.2.1 AHxxEMC 机种的监控与调测	9-20
9.2.2.2 DVPxxMC/AS5xx 机种的监控与调测	9-25
9.2.3 伺服参数备份还原.....	9-33
9.2.4 自动增益调整	9-35

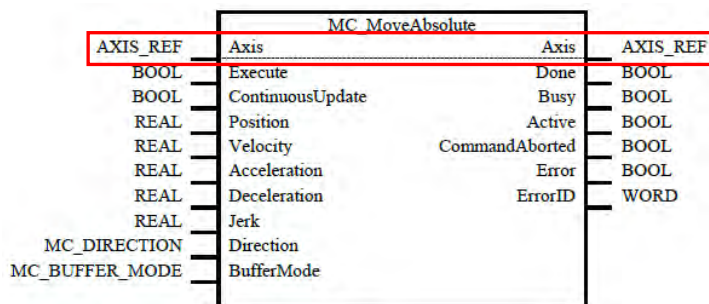
9.1 轴

支持轴相关功能的机种请参考下列表格。

功能 \ 机种	AHxxEMC	DVPxxMC/AS5xx
轴参数设定	V	V
监控与调测	V	V
伺服参数备份还原	V	
自动增益调整	V	只支持 DVP50MC/DVP50MC-06/DVP50MC-04S/DV50MC-16S /AS516E-B/AS532EST/AS564EST

9.1.1 轴的意义

轴在软件上的定义，依据 PLCopen MC 规范定义，「轴」为用在运动控制功能块中的 STRUCTURE 数据类型 VAR_IN_OUT 参数。轴提供了程序中的运动控制到实际输出设备中的一个接口，轴中纪录了各项运动控制相关信息，例如位置、速度、... 等等。如下图为 PLCopen MC 规范定义的功能块，而 AXIS_REF 便为规范定义的轴。



在实际应用上，便是利用各种运动功能块控制轴，再输出到目标马达电机驱动器上，达到运动控制。

*.DVPxxMC/AS5xx 机种，目前因开发延续性内容的关系，并不遵守「轴」用于运动控制功能块中的 STRUCTURE 数据类型 VAR_IN_OUT 原则，而采用 STRUCTURE 数据类型 VAR_INPUT。

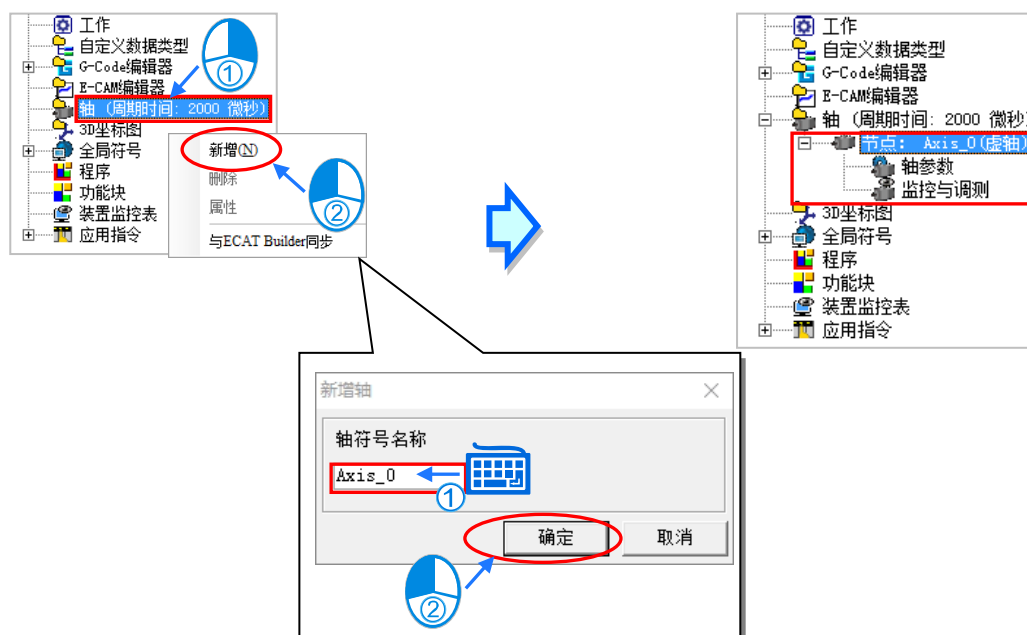
9

9.1.2 建立轴

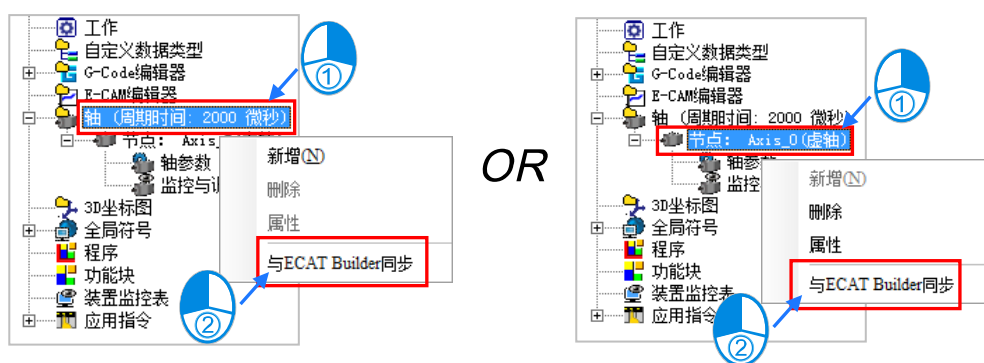
9.1.2.1 AHxxEMS 机种的建立轴

在项目管理区的轴项目单击鼠标右键，于快捷选单点击「新增」，出现新增轴窗口，输入轴符号名称，按下「确定」按钮，即会于轴项目下新增此节点项目。须注意轴符号名称不得与其它全局符号或区域

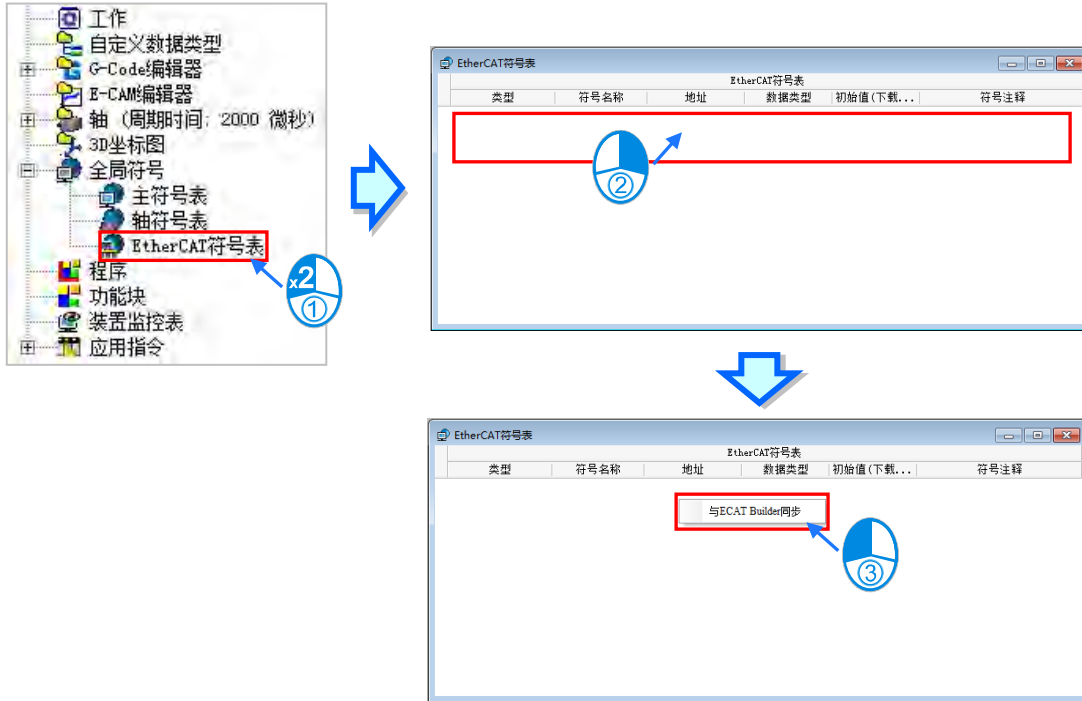
符号重复；至于轴项目右侧的「周期时间」，为 ECAT Builder 内主站的「周期时间」，「周期时间」的详细内容请参阅 ECAT Builder 使用手册第 3.2 节「主站设定」。



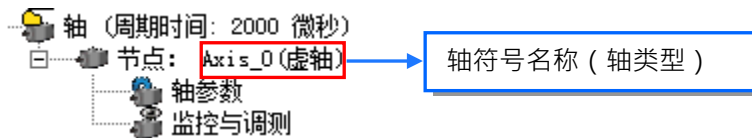
若用户欲将 ECAT Builder 的信息与「轴」的信息（「轴」的信息包含周期时间、节点信息）同步，可先在 ECAT Builder 内做保存的动作，接着在项目管理区的轴项目单击鼠标右键，快捷选单单击「与 ECAT Builder 同步」，此时刻 ECAT Builder 的信息将与「轴」符号表内容同步。



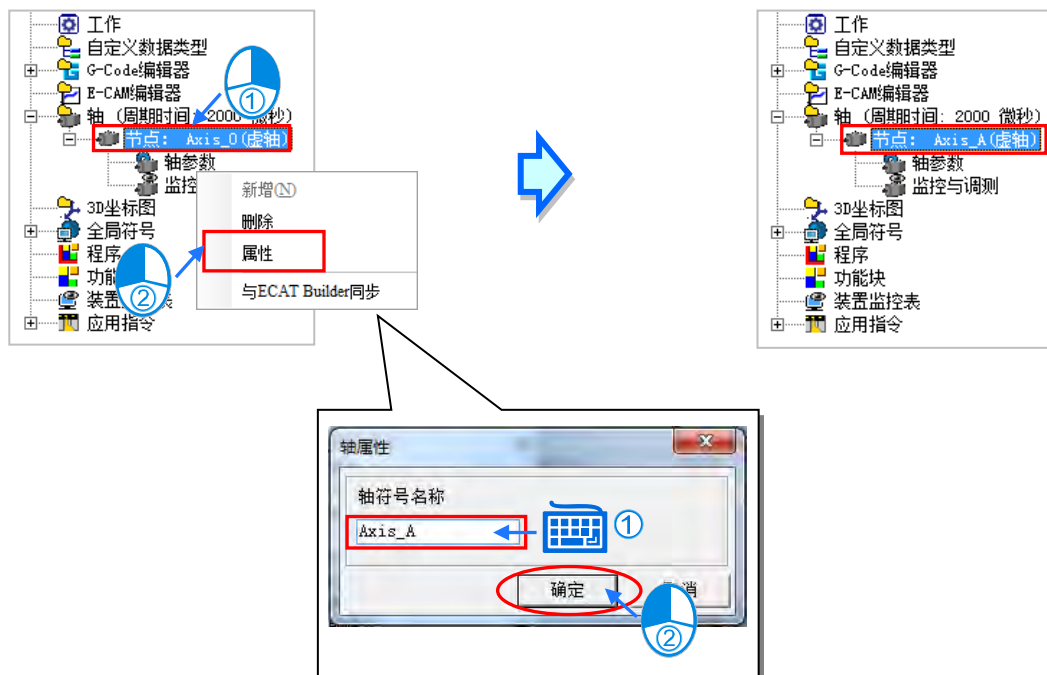
除了利用轴项目做同步更新，用户也可利用在 EtherCAT 符号表单击鼠标右键，快捷选单单击「与 ECAT Builder 同步」，将 ECAT Builder 的信息与「轴」的信息、EtherCAT 符号表的内容做同步。



轴节点项目名称列出前述的轴符号名称与轴类型，轴类型为依据参数设定中的轴类型设定，可为虚轴或实轴的 EtherCAT 从站名称与站号。每个轴项目下有轴参数设定和监控与调测两个子项目，请参阅后续章节。

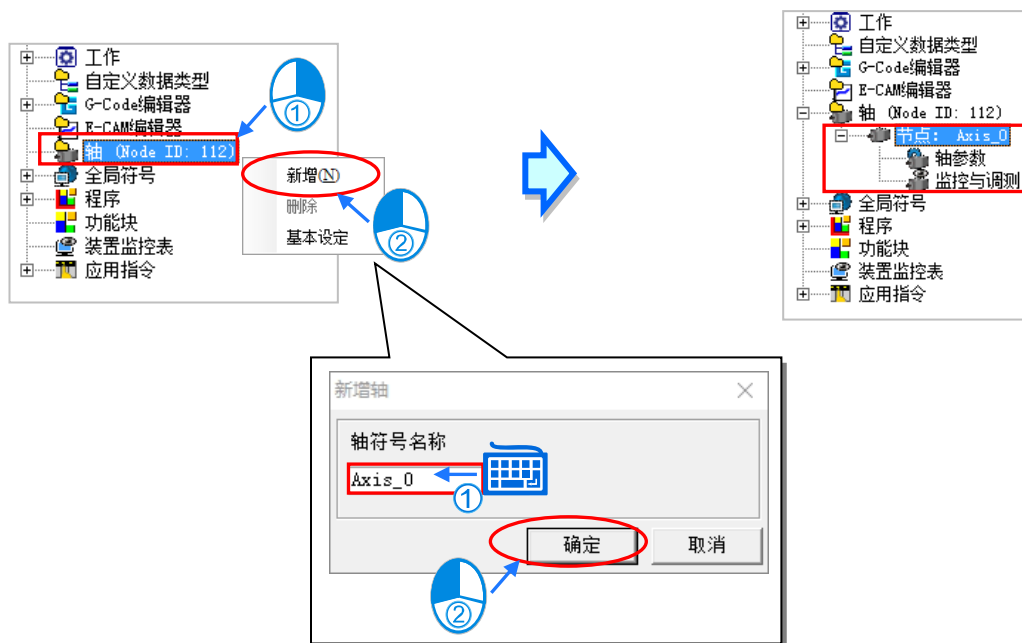


在已建立的轴上单击鼠标右键，可于快捷选单单击「删除」，删除此轴；或是单击「属性」，重新设定轴符号名称。



9.1.2.2 DVPxxMC/AS5xx 机种建立轴与轴基本设定

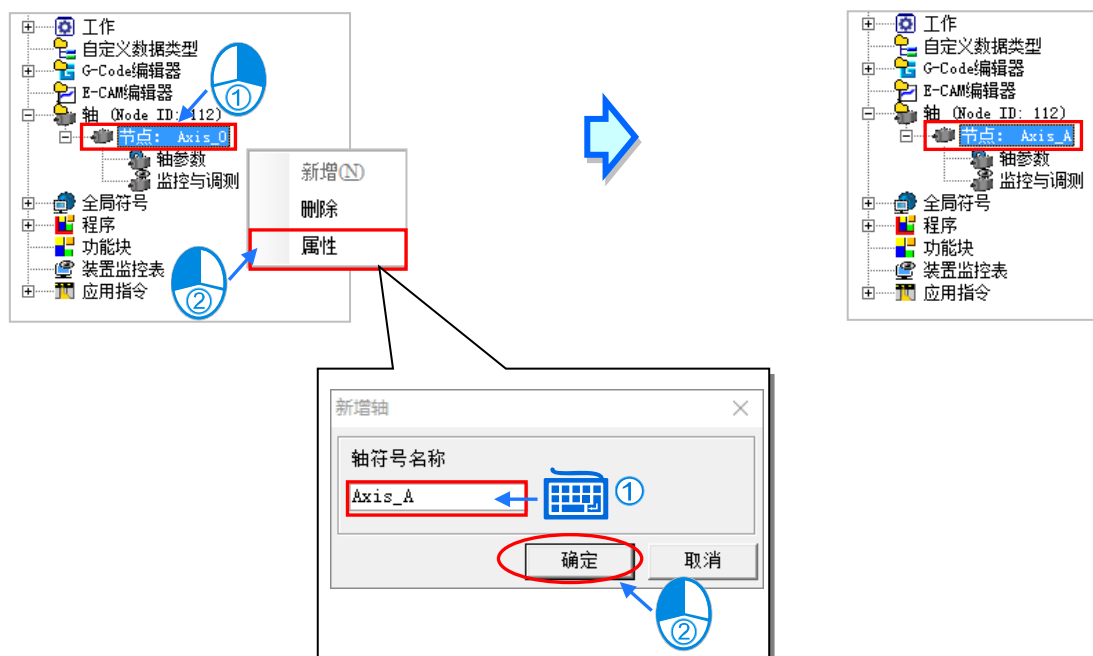
在项目管理区的轴项目点击鼠标右键，在快捷选单点击「新增」，出现新增轴窗口，输入轴符号名称，按下「确定」按钮，即会于轴项目下新增此节点项目，目前 DVPxxMC 与 AS5xx 机种建立轴的最大数目为 32 轴，但实轴最大仅支持 24 轴，而关于实轴、虚轴等的轴类型相关设定，可参考第 9.2.1.2 节。须注意轴符号名称不得与其他全域符号或区域符号重复；至于轴项目右侧的「NodeID:112」，为控制器 Motion 站号，预设是 112，不可更改。



轴节点项目会列出前述的**轴符号名称**。每个轴项目下有**轴参数**和**监控与调测**两个子项目，请参阅后续章节。



在已建立的轴上点击鼠标右键，可于快捷选单点击「**删除**」，删除此轴；或是点击「**属性**」，重新设定轴符号名称。



若用户需要调整轴的基本设定，于项目管理区的轴项目点击鼠标右键，于快捷选单点击「**基本设定**」，即可调整轴的基本设定。



而轴的基本设定可分成「常规」与「同步对象 (SYNC)」两个项目，接下来就这两个项目作介绍。

● 常规



- ❶ 设备名称：预设为用户使用的设备名称，不可修改。
- ❷ 站号：预设为用户 112，不可修改。
- ❸ 通讯速率：此处会显示 HWCONFIG 设置的 CANMotion 通讯口之速率，此处 CANMotion 通讯速率设定可参考第 3.4.2 节。

● 同步对象 (SYNC)



同步周期：

此处设定同步周期，主站按照同步周期设定时间周期性发送同步封包。在总线运动控制方式中，同步周期是一个非常重要的参数。因此如果同步周期的设定不当，可能会导致马达在通讯中出现 AL303/AL302/ AL301 报警，或者一些马达动作异常现象。

首先我们说明一下同步周期的构成。每个同步周期开始，会先扫描运动控制程序，扫描结束后将运算所得的控制资讯发送给所有轴。因此同步周期可以理解为由运动控制程序执行时间 + DVPXXMC 和所有马达的通讯时间。

运动控制程序执行时间可以参考系统变量中最大程序执行时间的数值，单位为微秒。1000 微秒为 1 毫秒。在实际应用中，建议向上取整数。例如最大程序执行时间数值为 2567 微秒=2.5 毫秒，我们可以取 3 毫秒作为程序执行时间。

DVPXXMC 和马达通讯的时间可以大概预估为每轴 0.5 毫秒。在时间应用中，建议向上取整数，例如应用中配置 5 个马达。通讯时间为 5×0.5 毫秒 = 2.5 毫秒。我们可以取 3 毫秒作为通讯时间。因此可以得到计算公式：同步周期时间 (毫秒) = 最大程序执行时间数值向上取整数 (毫秒) + DVPXXMC 和所有马达的通讯时间 (毫秒) + 1 (程序变更预留时间) (毫秒)。

为了预防程序变更后，程序的执行时间增大较多，导致之前设定的同步时间不适用，因此在设定的时候尽量做 1 到 2 毫秒的预留时间。

例如：最大程序执行时间为 1634 微秒，应用中共有 5 个马达。程序变更预留时间 1 毫秒。

同步周期 = 2 毫秒(最大程序执行时间 1634 微秒向上取整数) + 3 毫秒(5×0.5 向上取整数) + 1 毫秒 (程序预留时间) = 6 毫秒。

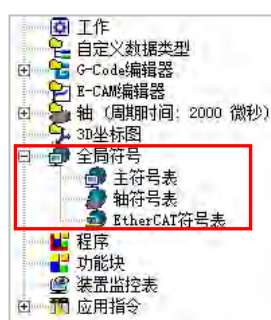
DVPxxMC 与 AS5xx 机种，目前有提供轴的相关系统结构，可参考下表。

成员变量		数据类型	功能描述
CmdPos		LREAL	命令位置 (单位：单位)
CmdVel		LREAL	命令速度 (单位：单位/秒)
CmdAcc		LREAL	命令加速度 (单位：单位/秒 ²)
CmdTrq		INT	命令扭矩 (额定扭矩的千分比)
ActPos		LREAL	实际位置 (单位：单位)
ActVel		LREAL	实际速度 (单位：单位/秒)

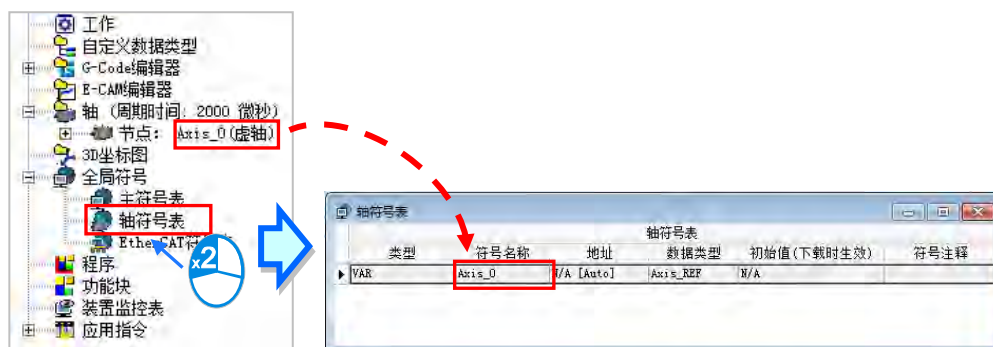
成员变量		数据类型	功能描述
ActTrq		INT	实际扭矩 (额定扭矩的千分比)
ActCur		INT	实际电流

9.1.3 全局符号表的主符号表与轴符号表

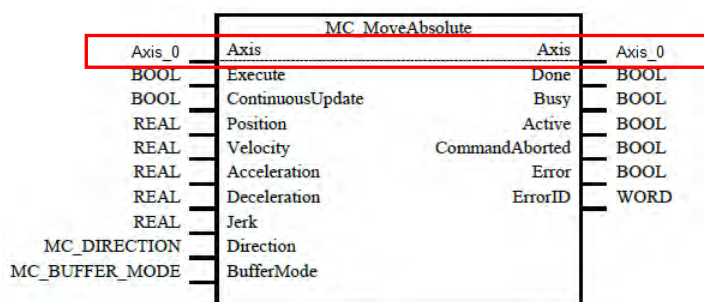
支持轴项目标机种，为了便于使用上的分类管理，全局符号表将包含多个子项目，其中包括主符号表、轴符号表，与其它网络符号表。主符号表相当于其它机种的全局符号表，用于编辑全局符号。而轴符号表则列出所有已建立的轴符号名称。其它网络符号表为此机种所支持的总线网络，由其它网络配置软件 (如 ECAT Builder) (所带入的符号数据，关于其它网络配置软件，请参阅各软件操作手册。



完成轴的建立后，在全局符号表中的轴符号表便会自动产生此轴符号名称的变量符号，所以轴符号表内容无法编辑，如下图以鼠标左键双击轴符号表项目，开启轴符号表。



轴变量符号可用于 PLCopen MC 功能块与台达功能块的接脚。如下图。

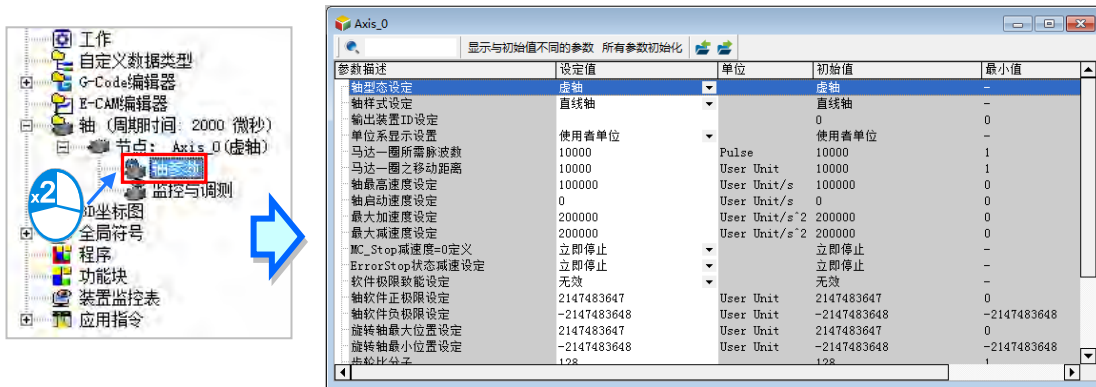


9.2 轴的参数设定、监控与调试

9.2.1 轴参数设定

9.2.1.1 AHxxEMC 机种的轴参数设定

在项目管理区的轴建立一个轴节点项目后，便会自动开启此轴的轴参数设定窗口。关闭后用户也可自行双击此轴节点项目下的轴参数设定项目再次开启参数设定窗口，此窗口列出此轴所有关于运动控制的设定，设定项目说明如下，详细意义可参照各机种手册。完成设定后可与项目一同下载至主机。



轴类型设定

用于设定有实际输出到驱动组件的**实轴**，或仅在控制器中运算的**虚轴**。

轴样式设定

对于不同的应用场合可设定轴为**直线轴**或**旋转轴**，直线轴为朝正方向（或负方向）旋转时位置会不断递增（递减），而旋转轴为朝正方向（或负方向）旋转到达极限值后；位置值会从最小值（最大值）开始重新计算。

输出装置 ID 设定

其中当轴类型设定为**实轴**时，**输出装置 ID 设定**必须由 ECAT Builder 中带入的马达名称与站号选择，代表此轴为实际输出到设定的 EtherCAT 马达驱动器。如下图，输出装置 ID 设定选择为名称 ASDA-A2-E CoE Drive 站号 5 的 EtherCAT 从站。用户须注意**输出装置 ID 设定**的下拉选单内容，是在 ECAT Builder 保存后 关闭 ECAT Builder 且关闭 HWCONFIG 时会更新，或在 ECAT Builder 保存后，有将「轴」的信息与 ECAT Builder 同步时，会更新，「轴」的信息与 ECAT Builder 同步方式可参阅第 9.1.2 节。



单位显示设置

选择系统单位；以用户单位为单位。

电机一圈所需脉冲数

设定控制器送出多少脉冲将使马达电机旋转一圈。

电机一圈移动距离

设定马达电机旋转一圈将使实际机构移动多少距离。

轴最高速度设定

设定轴允许的速度上限。

轴启动速度设定

轴开始运动时的启动速度。

最大加速度设定

设定轴允许的加速度上限。

最大减速度设定

设定轴允许的减速度上限。

MC_Stop 减速度=0 定义

当以指令或是发生错误使轴停止时，设定轴停止的方式，可设定**立即停止**或以目前的减速度**减速停止**。

ErrorStop 状态减速设定

当轴运动超越软极限设定而使轴停止时，设定轴停止的方式，可设定**立即停止**或以目前的减速度**减速停止**。

软件极限设定

设定是否开启软件极限，极限值由下两参数设定。

轴软件正极限设定

设定轴软件正极限值，只适用于直线轴。

轴软件负极限设定

设定轴软件负极限值，只适用于直线轴。

旋转轴最大位置设定

设定旋转轴朝正方向旋转的最大位置，旋转到达极限值后；位置值会从最小值开始重新计算。

旋转轴最小位置设定

设定旋转轴朝负方向旋转的最小位置。旋转到达极限值后；位置值会从最大值开始重新计算。

齿轮比分子

设定电子齿轮比的分子值。

齿轮比分母

设定电子齿轮比分母值。

原点复归加速度

原点复归加速度。执行原点复归时的加速度。

原点回归速度

原点回归速度。执行原点回归时的速度。

原点回归接近速度

执行原点回归时，电机先以原点复归速度运转，当触发近原点信号时，电机开始减速到原点回归接近速度运转。

位置偏差容许值

此值为实际位置与命令位置误差容许值。

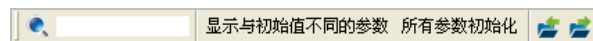
单轴运动立即启动

此处可以选择「关闭」或「启动」。若选择「启动」，当单轴运动启动后，不会受到 PLC 主机周期时间影响而马上执行运动，详细内容请见该机种相关手册。

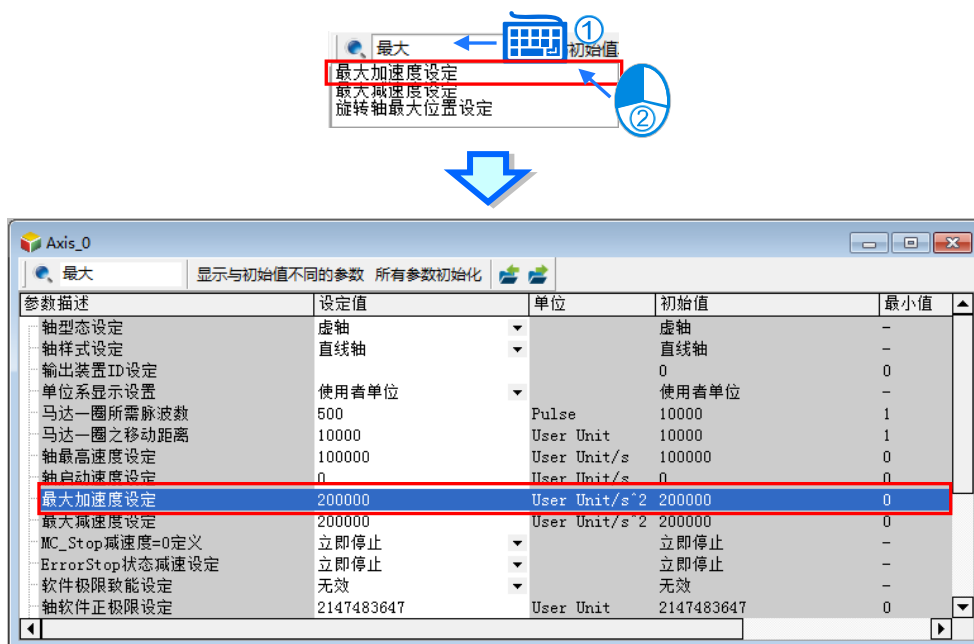
MC_CamIn 主轴方向

此处可以选择「正向」或「不指定」。当选择「正向」时，当主轴反向运动时，从轴不会动作；当选择「不指定」时，从轴不受主轴正反向限制。

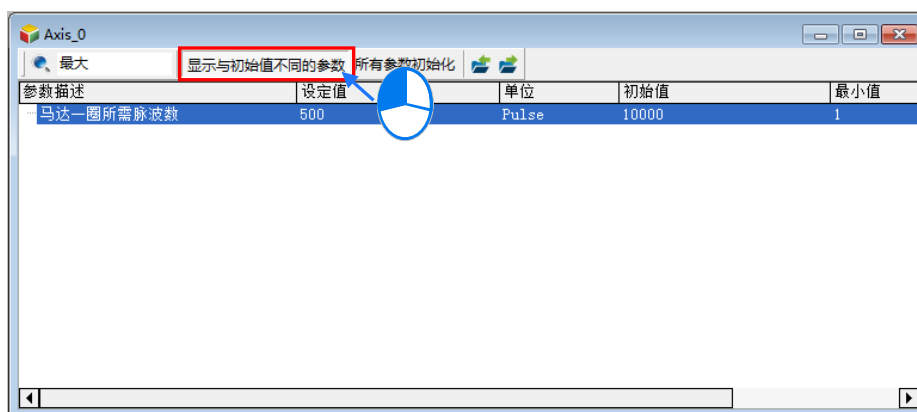
轴参数设定窗口提供了如下图的工具栏，便于编辑此轴参数表格。



搜寻工具，输入字符串后会列出相关的字符串，点击后表格会自动锁定到此字段。



点击 **显示与初始值不同的参数** 按钮后，表格将过滤出与初始设定不同的参数字段。



点击 **所有参数初始化** 按钮，将会把所有设定字段恢复为初始值。点击 **汇出** 按钮，可将目前的参数以 .CSV 格式导出到指定路径；点击 **汇入** 按钮可将指定路径的配置文件案汇入。

9.2.1.2 DVPXXMC/AS5xx 机种的轴参数设定

若用户选择的是 DVPXXMC 或 AS5xx 机种，一旦建立了一个轴节点项目后，便会自动开启此轴的轴参数窗口。关闭后用户也可自行双击此轴节点项目下的轴参数项目再次开启参数设定窗口。



而轴参数窗口列出此轴所有关于运动控制的设定，设定项目可分为两种，「参数设定」和「周期性通讯数据」，「参数设定」的参数内容详细意义可参照各机种手册。完成设定后便可与项目一同下载至主机。

● 参数设定



参数设定的内容如下：

主站站号

此项目为主站站号，目前固定为站号 112。

轴号

此处设定轴号，取值范围 1 ~ 32。

轴类型

用以选择轴的类型，分为「直线轴」和「旋转轴」。

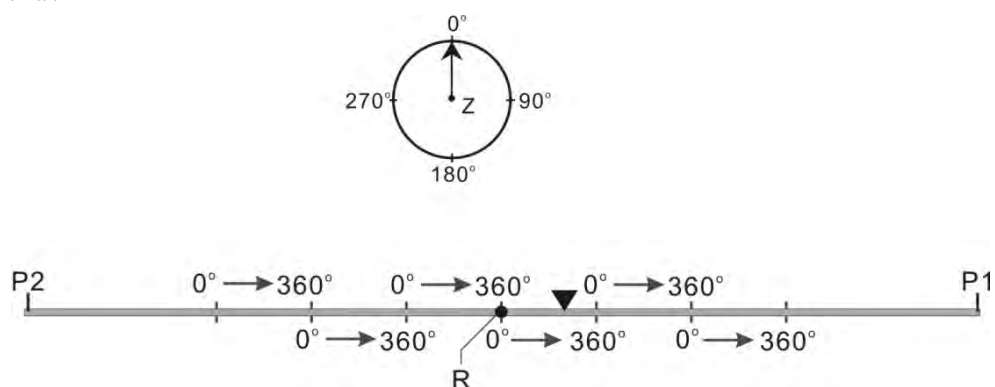
➤ 直线轴



直线轴模型批注：

P1	正极限
P2	反极限
▼	马达所在的位置

➤ 旋转轴



旋转轴模型批注：

P1	正极限
P2	反极限
▼	马达所在的位置
R	原点位置
Z	马达马达的轴心

模

用以平分终端执行机构实际位置的周期。

电子齿轮比分子

和电子齿轮比分母一起设定马达转一圈需要的脉波数。

电子齿轮比分母

和电子齿轮比分子一起设定马达转一圈需要的脉波数。

参考速度

此参数功能暂时保留，对控制器功能无影响。

轴最高速度设定

设定轴连接机构的最大速度。

最大加速度设定

此参数功能暂时保留，对控制器功能无影响。

最大减速度设定

此参数功能暂时保留，对控制器功能无影响。

原点复归模式

设定马达驱动器原点复归模式，取值范围 1~35。详细内容请参考该机种相关手册。

原点复归第一段速

为原点复归开始找到原点开关的速度。

原点复归第二段速

为找到原点开关后到机械原点的速度。

软极限开关

设定是否开启软体极限。当选项开启后，轴终端执行机构位置到达设定的「正向极限」或「反向极限」时，运动指令报错并且停止。

正向极限

允许轴的正向极限。

反向极限

允许轴的反向极限。

偏差错误开关

设定控制器是否检测轴的命令位置和回馈位置的差值。该参数设定为开启时，控制器检测命令位置和实际位置误差的值；该参数设定为关闭时，控制器不检测命令位置和实际位置误差的值。

位置偏差允许值

允许轴的命令位置和回馈位置之间差值的最大值。

位置偏差持续时间

允许轴的最大位置误差值的持续时间。

齿轮箱输入圈数

与齿轮箱输入圈数一起确定机构齿轮比。

齿轮箱输出圈数

与齿轮箱输出圈数一起确定机构齿轮比。



机构导程

齿轮箱输出端转一圈对应终端执行机构移动的单元数目。

轴模式


用于设定有实际输出到驱动元件的**实轴**、仅在控制器中运算的**虚轴**以及配合编码器应用的**编码器轴**。

编码器轴模式

可选择的模式分为增量型编码器 1、增量型编码器 2、绝对型编码器及数据来源，此项目需在上一项目「轴模式」，选择在「编码器轴」时，才可使用。

编码器轴模式选项	用途
增量型编码器 1	编码器轴的数据来源为增量型编码器 1
增量型编码器 2	编码器轴的数据来源为增量型编码器 2
绝对型编码器	编码器轴的数据来源为绝对型编码器
数据来源	编码器轴的数据来源为指定装置中的数据

编码器轴数据来源

此项目需在上一项目「编码器轴模式」，选择在「数据来源」时，才可使用。当用户选择「数据来源」后，可对「编码器轴数据来源」的设定值栏位此按键点击鼠标左键，此时将出现「编码器轴数据来源」窗口，如下图。



- ① 此处下拉选单可以选择装置名称，可选择%IL、%QL及%ML。
- ② 此处可设定装置的编号。

而选择在「数据来源」时，编码器轴的位置变化量为指定装置值每个扫描周期的变化量。

EtherCAT 节点号

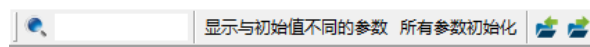
DVP50MC、DVP50MC-06 以及 AS516E-B 的机种支持此项目，节点编号是由 1001 开始一直到 1024。

● 周期性通讯数据



勾选「速度」、「扭矩」、「电流」或者「用户定义参数」后，则马达驱动器在每个同步周期后往总在线会发出对数据资讯，周期通讯数据最多选择 8 个位元组的数据。周期通讯数据对应的数据勾选后，才可以在轴结构体中监控到对应的参数。

轴参数设定窗口提供了如下图的工具列，便于编辑此轴参数表格。

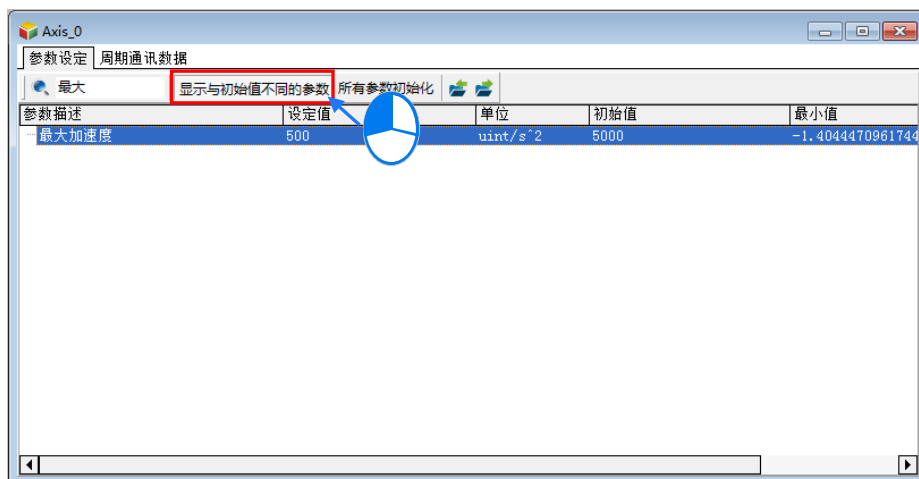


搜寻工具，输入字串后会列出相关的字串，点击后表格会自动锁定到此栏位。



参数描述	设定值	单位	初始值	最小值
轴号	1		-	-
轴类型	直线轴		直线轴	-
模	360		360	-1.4044470961
电子齿轮比分子	128		128	1
电子齿轮比分母	1		1	1
参考速度	10000	uint/s	10000	-1.4044470961
最大速度	10000	uint/s	10000	-1.4044470961
最大加速度	5000	uint/s²	5000	-1.4044470961
最大减速度	5000	uint/s ²	5000	-1.4044470961
原点回归模式	1		1	1
原点回归第一段速	20	rpm	20	1
原点回归第二段速	10	rpm	10	1
软极限开关	关闭		关闭	-
正向极限	0	uint	0	-1.4044470961
反向极限	0	uint	0	-1.4044470961
误差错误开关	关闭		关闭	-
位置误差允许值	500	uint	500	-1.4044470961

点击 **显示与初始值不同的参数** 按钮后，表格将过滤出与初始设定不同的参数栏位。

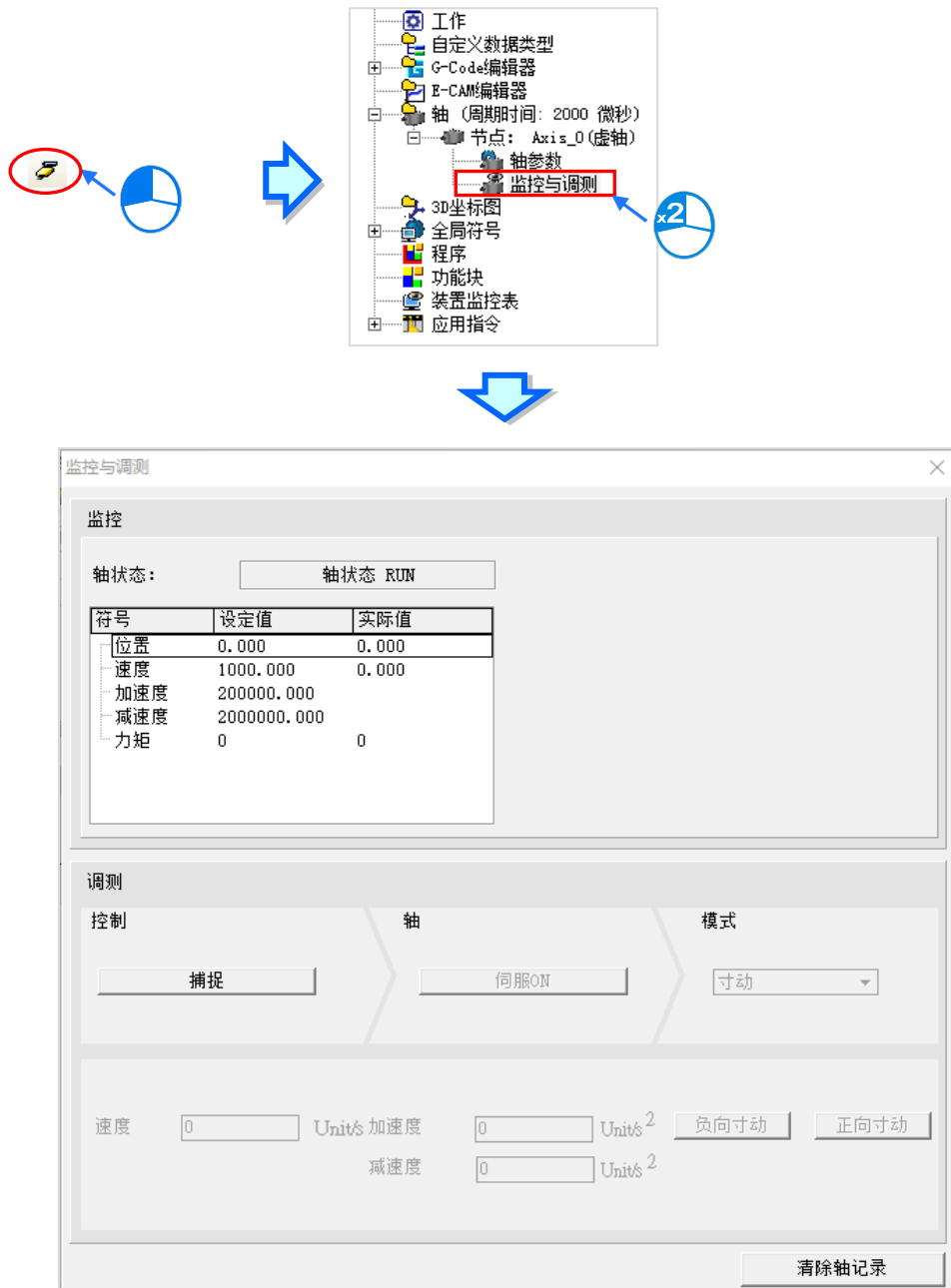


点击 **所有参数初始化** 按钮，将会把所有设定栏位恢复为初始值。点击 **汇出** 按钮，可将目前的参数以 .CSV 格式汇出到指定路径；点击 **汇入** 按钮可将指定路径的设定档案汇入。

9.2.2 监控与调试

9.2.2.1 AHxxEMC 机种的监控与调测

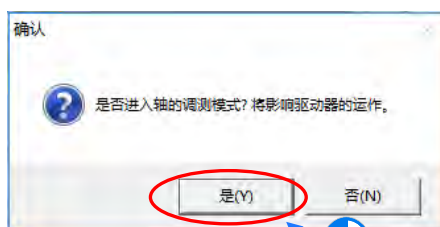
当 ISPSOft 进入在线模式后，以鼠标左键双击轴项目下的**监控与调测**项目，开启监控与调测窗。如下图。



窗口的上半部为监控窗口，用于实时显示轴的状态与轴的物理量设定值与实际值，例如位置、速度等等。



窗口的下半部为调测窗口，用于在测试阶段连接硬件后，对此轴实时下达调测的命令，如下图，使用时按下「捕捉」的按钮，将会出现询问窗口，确认取得对此轴的控制。请注意，对轴进行调测前，须确认现场的配置与人员的位置，在电机运作时不会造成人员的伤害或物体的损坏。



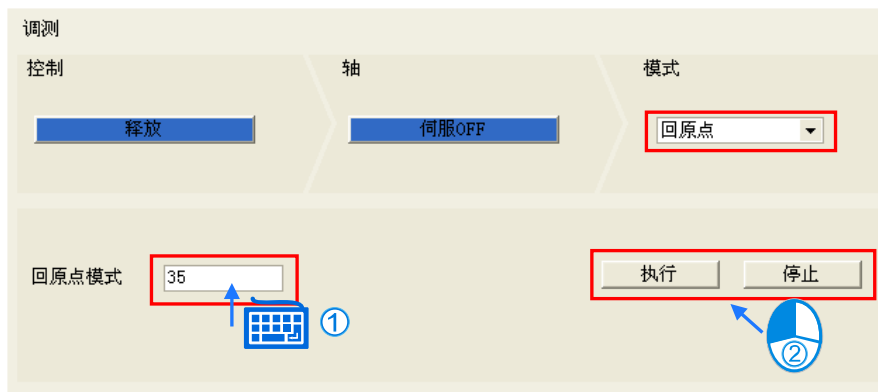
欲关闭调测模式时，点击「释放」按钮，便释放此轴的控制权。

取得轴控制后，接着可按下「马达 ON」或「马达 OFF」开关马达驱动器。



当在**马达 ON** 时，于**模式**的字段便可选择各种对轴调测的模式，不同的模式下方也会出现不同的执行功能项目供用户输入。对应的功能如下说明。

回原点：依据轴参数的设定执行原点回归，依据 CANopen 规范的 35 种回原点模式中，输入**回原点模式**字段，按下「执行」按钮，便会开始进行原点复归动作；原点复归时按下「停止」按钮，便会中止动作。



寸动：依据轴参数设定的寸动速度执行寸动，输入寸动**速度**、寸动**加速度**、寸动**减速度**后，按下按钮执行「正向寸动」或「负向寸动」。



速度：输入**速度**、**加速度时间**、**减速度时间**值。当用户按下「正向」或「负向」按钮时，驱动器以设定的加速度时间加速到设定速度维持正向或负向运转，按下「停止」后以设定减速度时间减速停止。



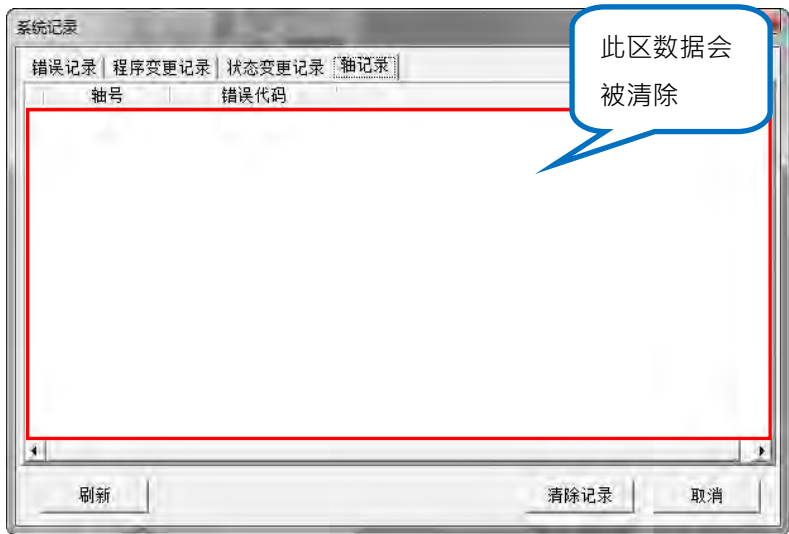
相对位置：输入**相对位置**（正或负表示）、**速度**、**加速度时间**、**减速度时间**值。当用户按下「启动」按钮时，驱动器以设定的加速度时间加速到设定速度维持运转，依减速度时间减速停止后到达相对位置。运转时按下「停止」按钮，便会中止动作。



绝对位置：输入**位置**、**速度**、**加速度时间**、**减速度时间**值。当用户按下「启动」按钮时，驱动器以设定的加速度时间加速到设定速度维持运转，依减速度时间减速停止后到达绝对位置。运转时按下「停止」按钮，便会中止动作。



若单击「清除轴记录」按钮，此时系统内的轴记录将被清除。

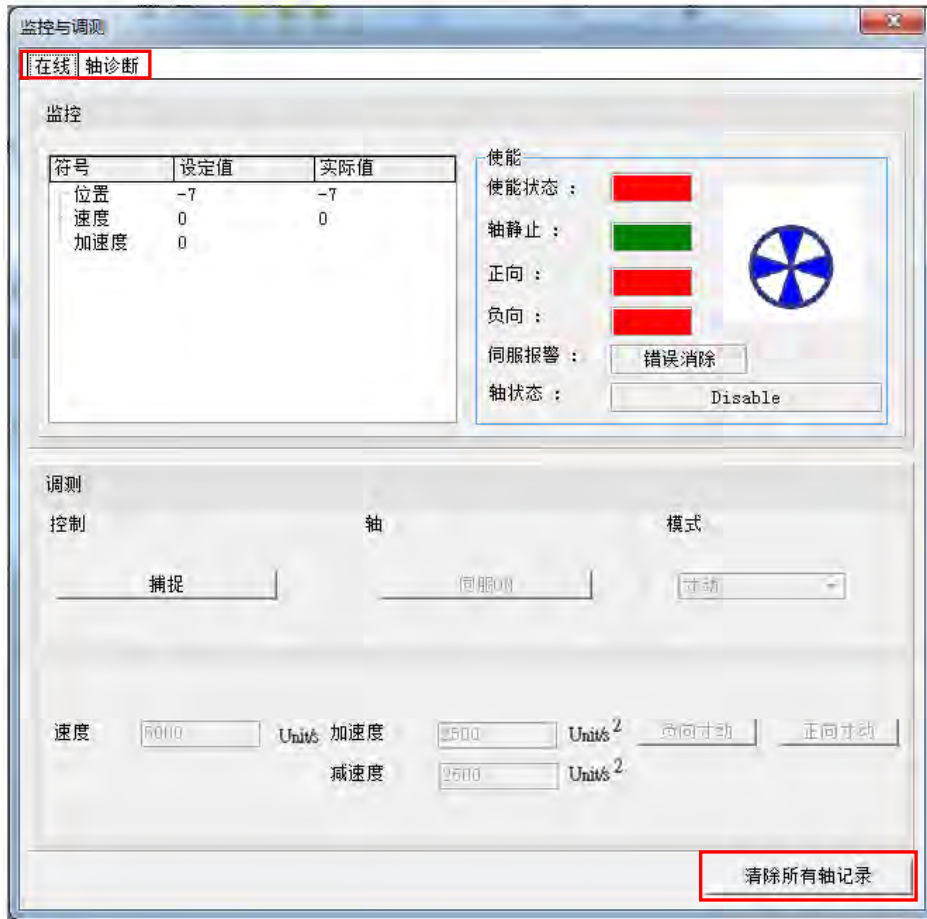


9.2.2.2 DVPxxMC/AS5xx 机种的监控与调测

若用户选择的是 DVPxxMC 或 AS5xx 机种，当 ISPSOft 进入在线模式后，以鼠标左键双击轴项目下的**监控与调测**项目，开启监控与调测窗口。如下图。



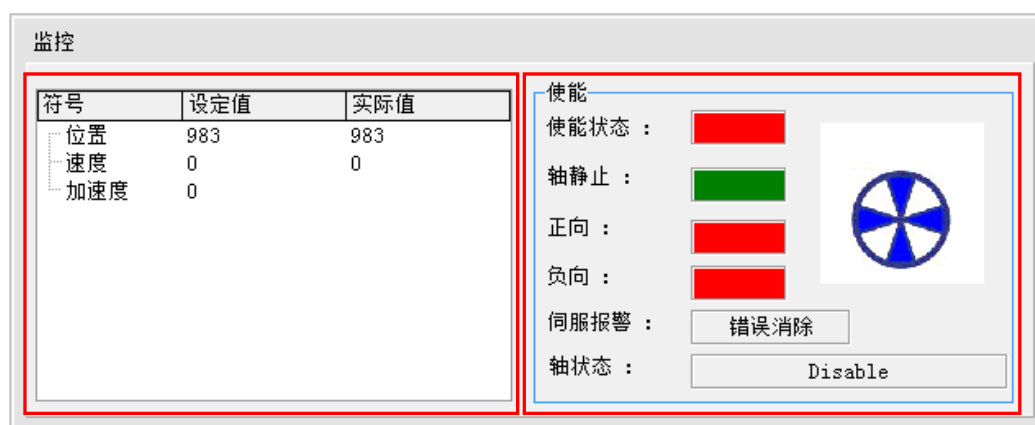
监控与调测窗口，将可见两个分页「在线」及「轴诊断」，来做进一步监控调测使用，而底部的「清除所有轴记录」按钮，可用来清除项目管理区的所有轴之记录。



- 在线



「在线」分页的上半部为监控窗口，左侧用于即时显示轴的状态与轴的物理量设定值与实际值，例如位置、速度等等，右侧「使能」区则会显示轴的相关状态及警报，此区域可配合「在线」分页的下半部之调测窗口做使用。



「在线」分页的窗口下半部为调测窗口，用于在测试阶段连接硬件后，对此轴即时下达调测的命令，如下图，使用时按下「捕捉」的按钮，将会出现询问窗口，确认取得对此轴的控制。请注意，对轴进行调测前，须确认现场的配置与人员的位置，在马达运作时不会造成人员的伤害或物体的损坏。



欲关闭调测模式时，点击「释放」按钮，便释放此轴的控制权。

取得轴控制后，接着可按下「马达 ON」或「马达 OFF」开关马达驱动器。

*.若用户选择的轴模式为「虚轴」则取得轴控制后，将会自动将此轴维持在马达为 ON 的状态。轴模式的设定可参阅第 9.2.1.2 节。



控制器 Motion 通讯口在连接马达的前提下，点击「马达 ON」，软体会提示寸动时需注意以下事项。

- (a) 确认周期数据交换中已勾选速度。
- (b) 程序中请勿对该轴使用 MC_Power 指令。
- (c) 如果再寸动过程中，马达通讯断线，马达轴会在 1 秒后自动停止。

而当马达由 OFF 变为 ON 时，「使能」区的「使能状态」项目将由红灯变为绿灯，且「轴状态」项目将由「Disable」变为显示「standstill」。

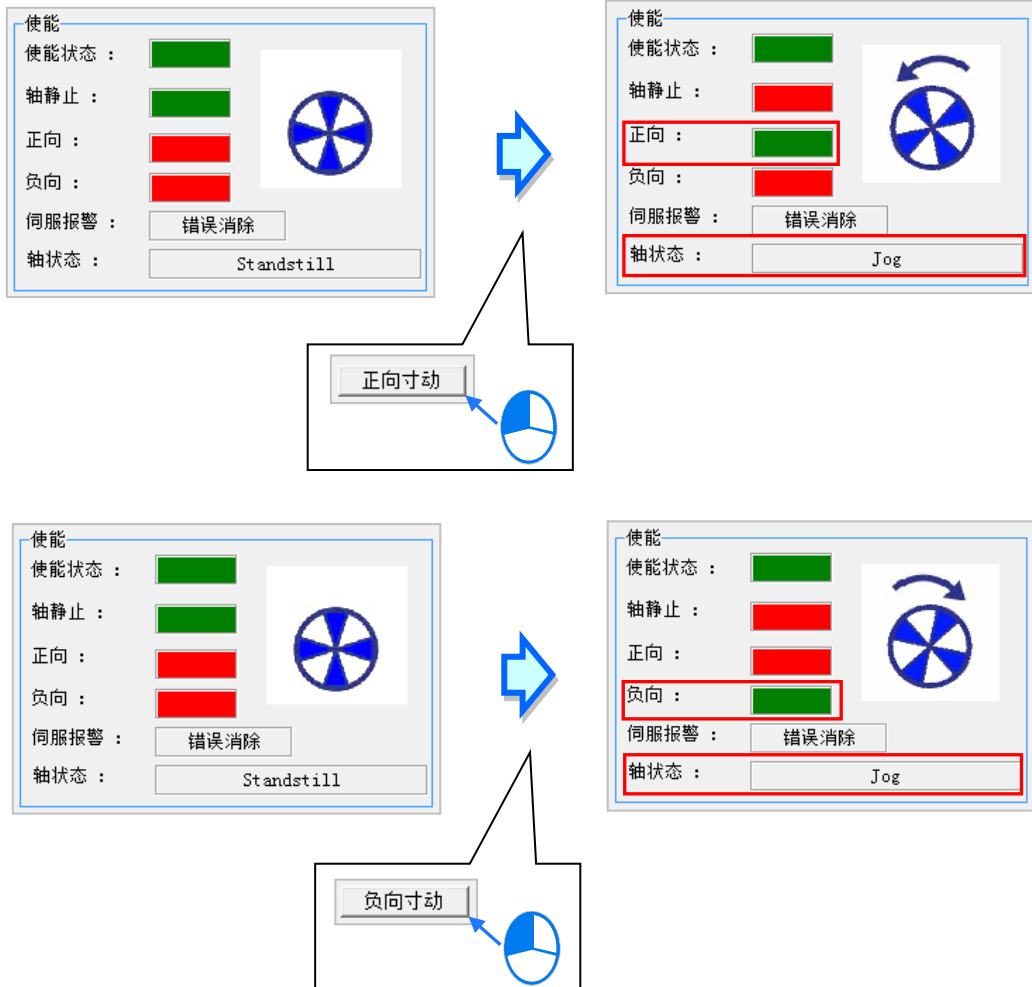


当于马达 ON 时，在「模式」的栏位便可选择各种对轴调测的模式，目前仅提供「寸动」供用户选择。而其功能如下说明。

寸动：依据轴参数设定的寸动速度执行寸动，输入寸动速度、寸动加速度、寸动减速度后，按下按钮执行「正向寸动」或「负向寸动」。



当在按下按钮执行「正向寸动」/「负向寸动」时，「使能」区的「正向」/「负向」项目将由红灯变为绿灯，且「使能」区之「轴状态」项目将显示「Jog」。(轴的寸动功能需配合控制器为运行时，才可做使用)

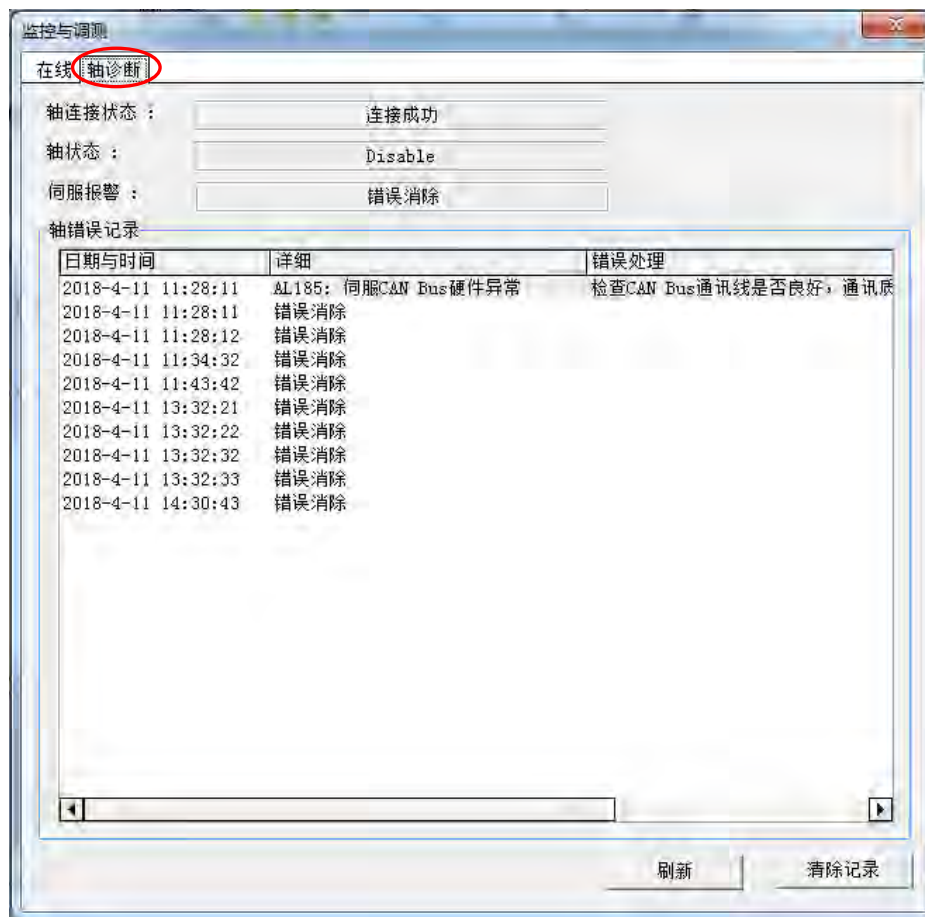


下表为「使能」区各项参数的含义。

参数名称	说明
使能状态	绿色表示使能，红色表示未使能
轴静止	绿色表示静止，红色表示轴寸动
正向	绿色表示轴正转，红色表示轴反转或静止
负向	绿色表示轴反转，红色表示轴正转或静止
马达警报	显示马达对应错误信息
轴状态	分为 Disable、Standstill、ErrorStop、Stopping、Homing、Discrete Motion、Continuous Motion、Synchronized Motion 及 Jog

● 轴诊断

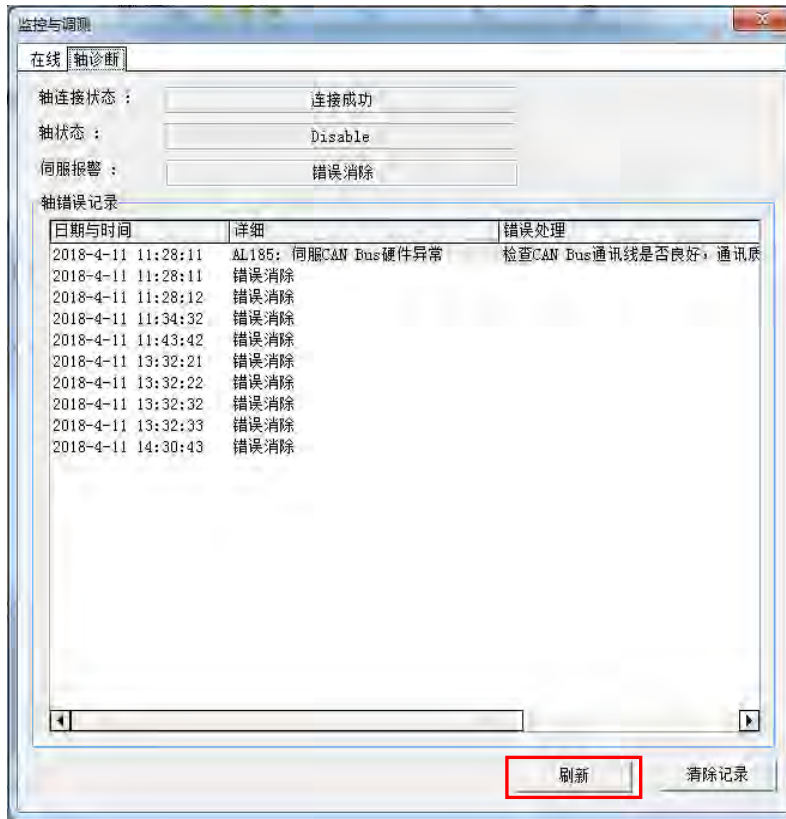
此处用于诊断控制器和轴之间连接是否正常及轴是否警报，会记录轴连接状态、轴状态及马达警报，而「轴错误记录」区会记录错误原因及错误处理。



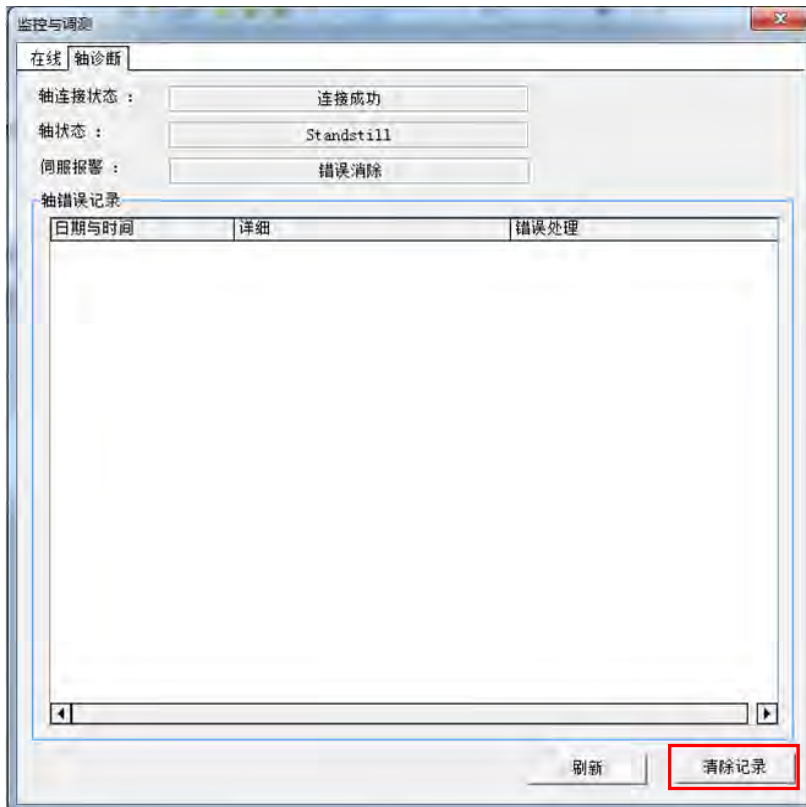
上图各项参数的含义如下表所示。

参数名称	说明
轴连接状态	控制器和伺服驱动器是否成功连接
轴状态	轴对应状态机（状态机介绍可参阅该機種相关手册）
伺服报警	显示伺服报警内容
轴错误记录	记录伺服报警信息，包括警报发生的时间、可能的原因和错误处理方法

点击「重新整理」，可以更新轴诊断页面的。

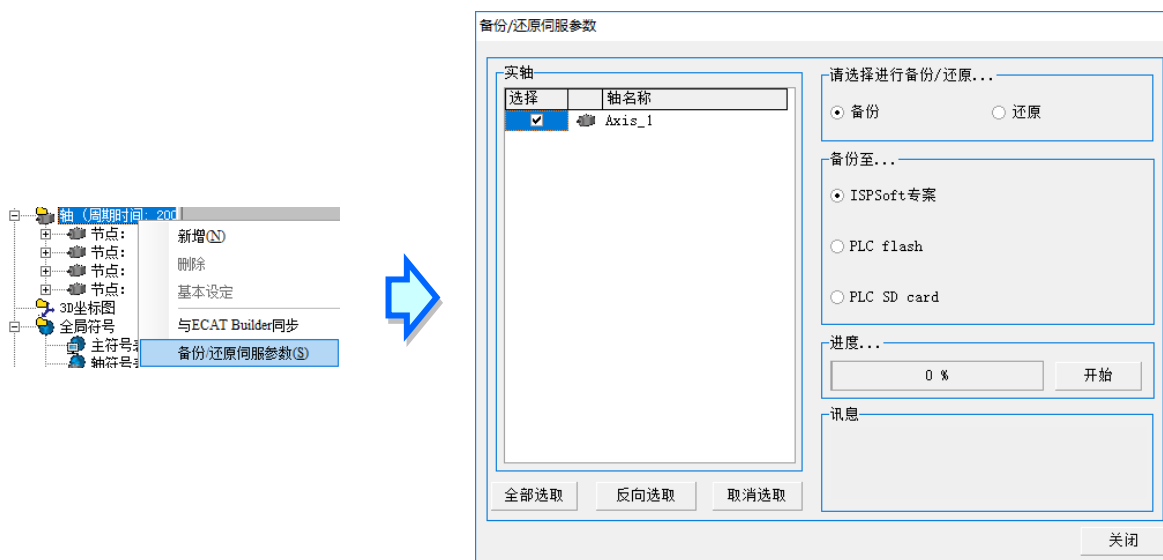


点击「清除纪录」清除当前轴错误信息纪录。



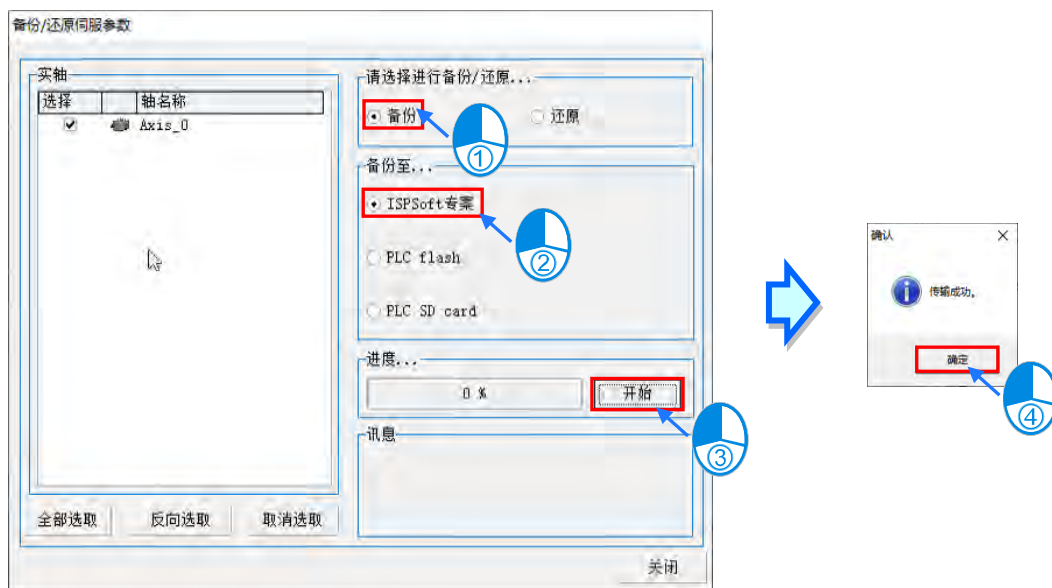
9.2.3 伺服参数备份还原

在「轴」节点点击鼠标右键，接着点击「备份/还原伺服参数」，透过备份/还原伺服参数页面的内容，可以将所连接的实轴设定，备份至 ISPSOft 项目、PLC Flash、PLC SD Card 这三个地方，亦可将备份好的配置还原至 ISPSOft 项目、PLC Flash、PLC SD Card 这三个地方，选好要操作的动作点击开始即可。



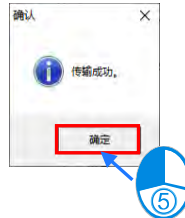
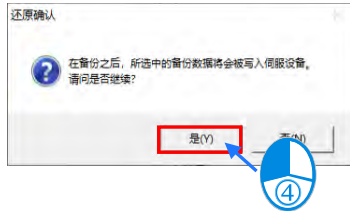
范例将实轴设定备份至 ISPSOft 项目，再从 ISPSOft 项目还原至新项目中。

在「备份/还原伺服参数」页面中单击「备份」接着再点选备份至...中的「ISPSOft 项目」，最后按下「开始」等待进度条完成后跳出完成窗口，即完成备份。



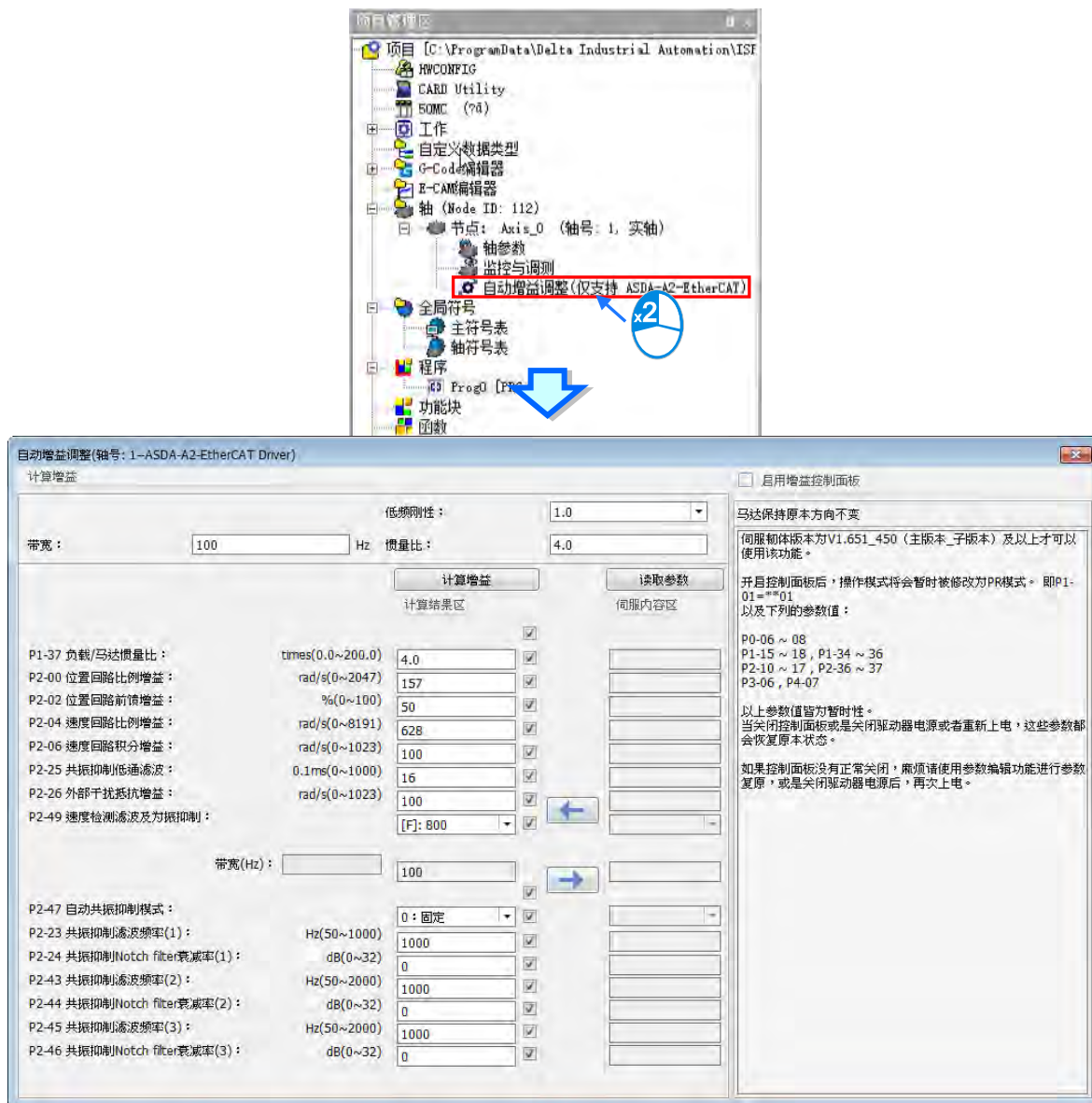
范例将实轴设定从 ISP 项目还原至新项目中。

在「备份/还原伺服参数」页面中单击「还原」接着再点选备份至...中的「ISPSOft 项目」，最后按下「开始」，跳出还原确认窗口，确定将实轴设定写入伺服设备后单击「是」，等待进度条完成后跳出确认窗口即完成还原。

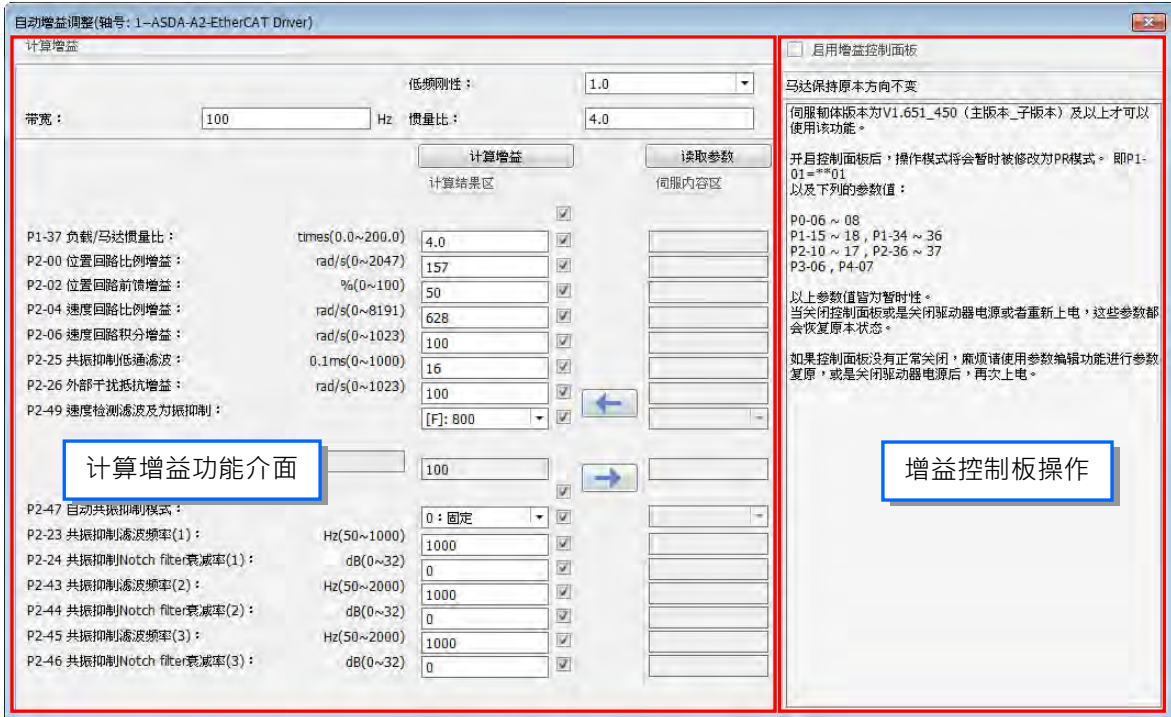


9.2.4 自动增益调整

在项目树中的「轴」节点下拉选单中，点击「自动增益调整」打开自动增益调整接口。

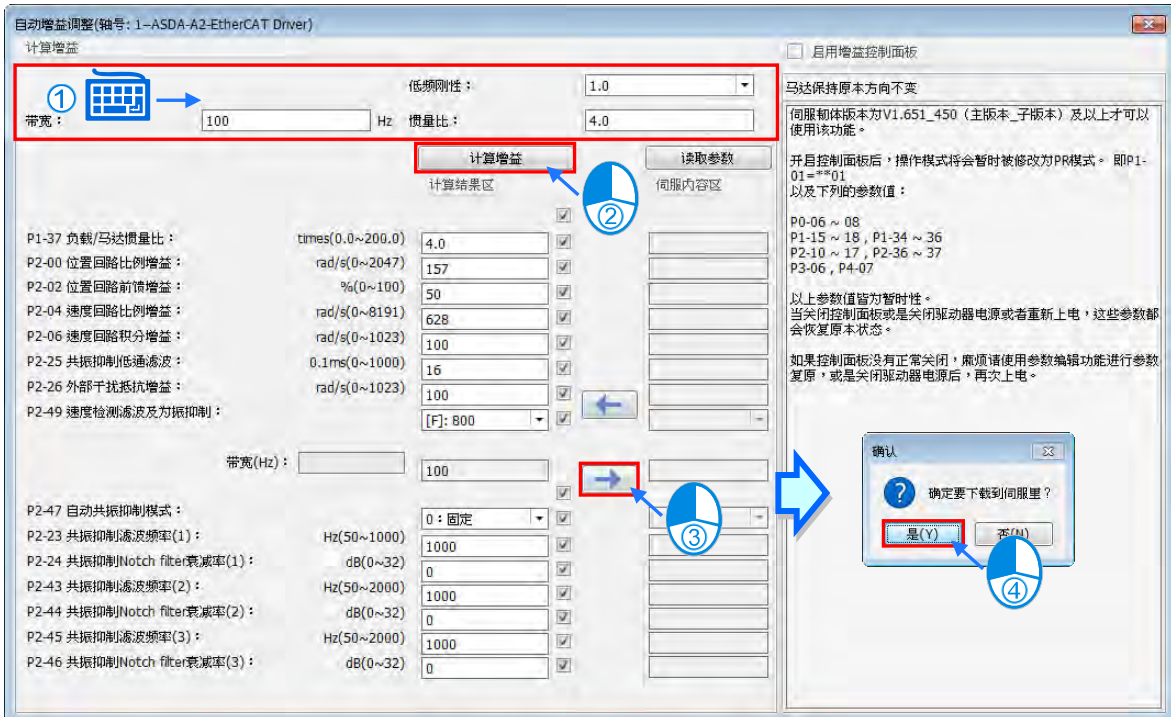


自动增益调整接口中，左边为计算增益功能接口，右边为增益控制板操作接口。



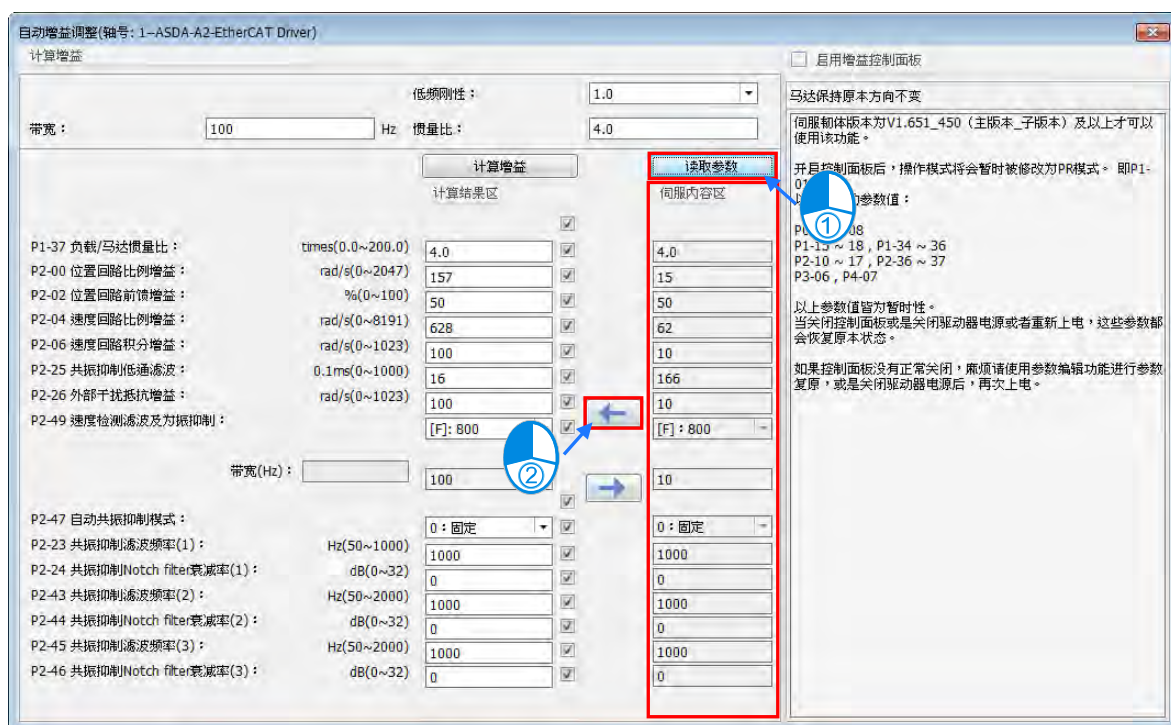
● 计算增益功能接口

计算增益：用户可在下图红色方框处输入适当的低频刚性、带宽和惯量比后，按下「计算增益」按钮，即可算出相关的增益参数值。再按下“→”按钮，可以将计算得到的结果写入到伺服中。写入到伺服时，可勾选是否下载该参数，有勾选表示下载该参数。



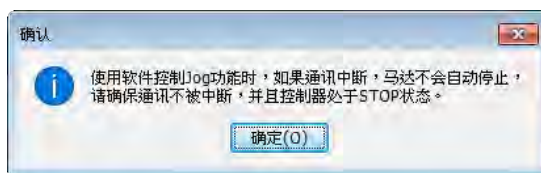
读取参数：按下「读取参数」按钮后，会读取伺服中参数的现在值并显示在接口的“伺服内容区”

中，以供用户检查是否合适。参数读取成功后，用户可以使用“←”按钮，将“伺服内容区”中的值写入“计算结果区”，然后修改相关参数后再写入到伺服。

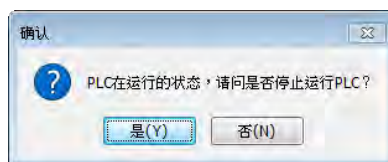


● 增益控制板操作

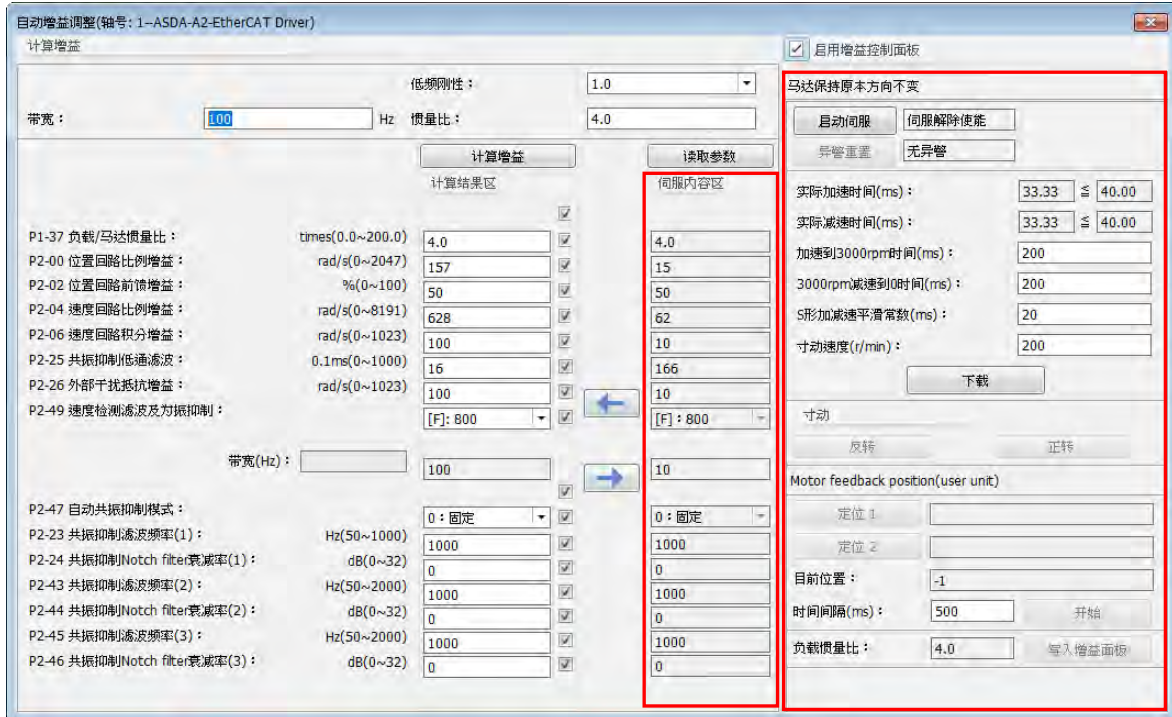
使用增益控制板前，需要用户勾选右侧「增益控制板操作」中的「启用增益控制板」，勾选之后软件会提示相关信息如下图所示，提醒用户在使用控制伺服点动过程中，保持通讯正常。



点击「确定」后，若控制器处于运行 (RUN) 状态，则软件会弹出窗口让控制器由运行 (RUN) 状态切换到停止 (STOP) 状态，如下图所示。



点击上图中「是」按钮后，软件会打开增益控制板，同时自动读取参数并填入「伺服内容区」。启动增益控制板时，软件会控制伺服进入 Pr 模式，伺服参数 P1-01、P0-10~12、P0-18~20、P1-15~18、P5-20~21、P1-36、P2-10~17、P5-60、P5-61、P3-06、P4-07 会暂时被修改，其中 P1-01 的输出方向设定不会被修改。当关闭增益控制板时，这些参数会恢复启动增益控制板前的设定值。



该接口中相关功能介绍如下（详细设定请参考产品手册）：

1. 启动伺服：按下「启动伺服」以启动伺服使能，若此时伺服有异警发生，则无法继续操作，必须先排除异警后，才可以继续操作。
2. 设定点动相关参数：设定加减速时间，S形加减速平滑常数与点动速度。点动速度代表调整时电机转动速度。设定完成后，点击「下载」按钮，将设定写入伺服，或者在按下点动按钮时，这些参数也会写入伺服。
3. Jog：依方向不同选择正向或者反向点动按钮。按住按钮时，马达运转，松开按钮时，马达停止。马达回授位置会显示在“目前位置”一栏中。用户可以在正向点动完成后，点击「定位 1」按钮，可以将马达回授位置显示到「定位 1」一栏中，在反向点动完成后，点击「定位 2」按钮，将马达回授位置显示到「定位 2」一栏中。
4. 定位：点击「定位 1」或「定位 2」按钮，软件会读取当前的马达回授位置值，并显示到「定位 1」或「定位 2」一栏中。
5. 时间间隔：定位 1 和定位 2 执行完后所需等待的时间，如定位 1 执行完后等待“时间间隔”设定的时间后再执行定位 2，定位 2 执行完后等待“时间间隔”设定的时间后再执行定位。
6. 开始调试：点击「开始」按钮，开始执行调试。开始调试后，「开始」按钮会变为「停止」按钮，调试过程中，定位 1 和定位 2 交替执行，观察负载惯量比的变化，当趋近于稳定状态（变动幅度不大），用户可以按下“停止”按钮停止马达动作。
7. 负载惯量比：参考第 6 点说明部分。
8. 写入增益面板：按下「写入增益面板」，可以将估测好的惯量数值写入计算增益功能接口的「计算结果区」中。当用户设定好相关参数后，可以点击“→”按钮，将参数写入伺服。
9. 关闭增益控制板：结束动态计算后，请用户先按「启动伺服」按钮，将伺服设为“Servo Off”状态，再取消勾选“启动增益控制板”。

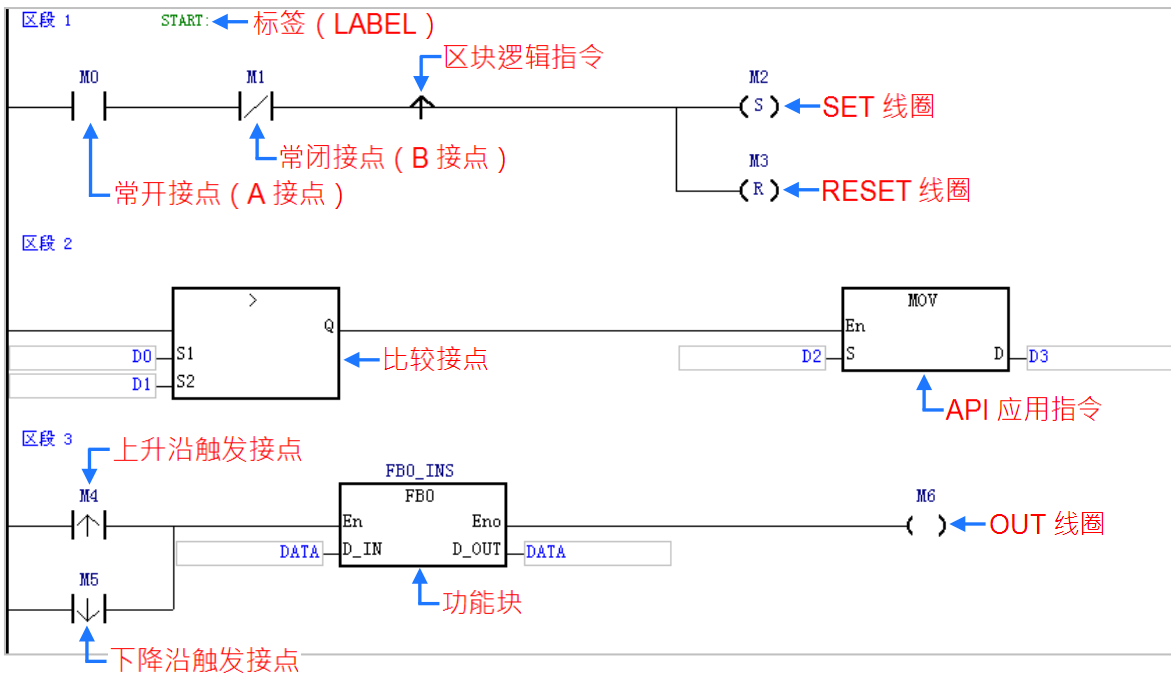
第10章 梯形图 LD

目录

10.1	梯形图 (LD) 简介.....	10-2
10.1.1	ISPSoft 的 LD 编辑环境.....	10-2
10.1.2	梯形图区段 (NETWORK)	10-4
10.1.3	选取对象.....	10-5
10.2	ISPSoft 中建立 LD 程序	10-6
10.2.1	建立接点 (Contact)、线圈 (Coil) 及 MPS	10-6
10.2.1.1	插入接点 (Contact) 与变更接点类型.....	10-6
10.2.1.2	插入线圈 (Coil) 与变更线圈类型	10-8
10.2.1.3	新增 MPS	10-9
10.2.2	LD 中使用装置、变量符号及常数	10-10
10.2.3	指令编辑模式	10-11
10.2.4	插入 API 应用指令与功能块	10-12
10.2.5	编辑比较接点	10-15
10.2.6	编辑区块逻辑指令.....	10-17
10.2.7	编辑多重输出	10-20
10.2.8	设置标签 (Label)	10-21
10.2.9	编辑批注.....	10-22
10.2.10	符号模式与地址模式的显示切换	10-24
10.2.11	书签功能	10-25
10.2.12	梯形图区段的使能与失效.....	10-26

10.1 梯形图 (LD) 简介

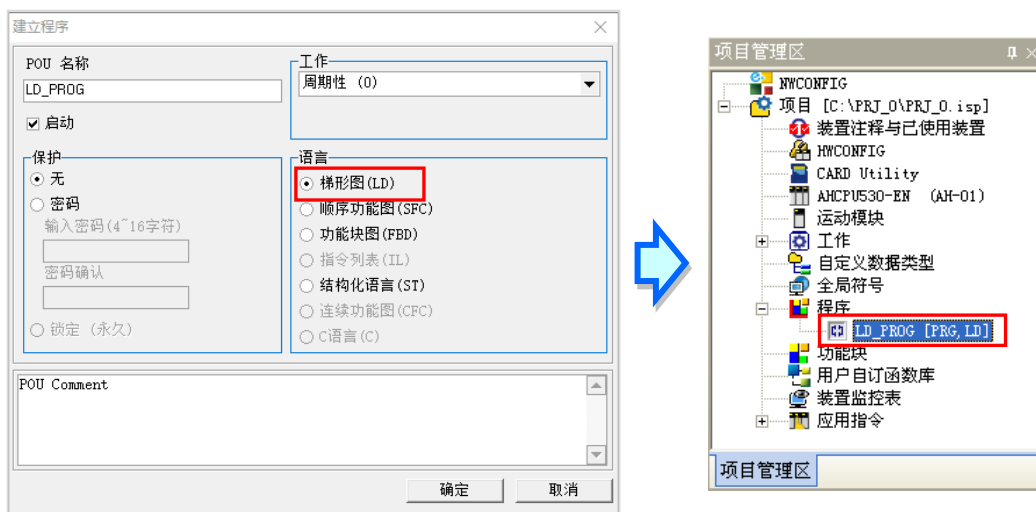
梯形图 (Ladder Diagram · LD) 为 IEC61131-3 所规范的 PLC 编程语言之一，也是最被广泛使用的 PLC 开发工具，其特色在于以控制回路图来表达程序的逻辑。下图即为 ISPSOft 中的梯形图程序。



*.关于梯形图的编程概念请参考台达 DVP 机种的应用技术手册 (程序篇)，本章将不再多加说明。

10.1.1 ISPSOft 的 LD 编辑环境

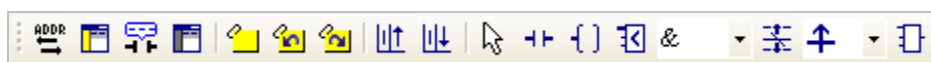
在建立 POU 时，于语言字段选择「梯形图 (LD)」便可建立梯形图程序，请参考第 5.4.1 节的说明。



下图即为 ISPSOft 当中的梯形图 (LD) 编辑环境，窗口的上方为区域符号表，而下方则是编辑工作区。



进入 LD 的编辑环境后，ISPSOft 的窗口中便会出现对应的编程图标工具栏，其功能简介如下。



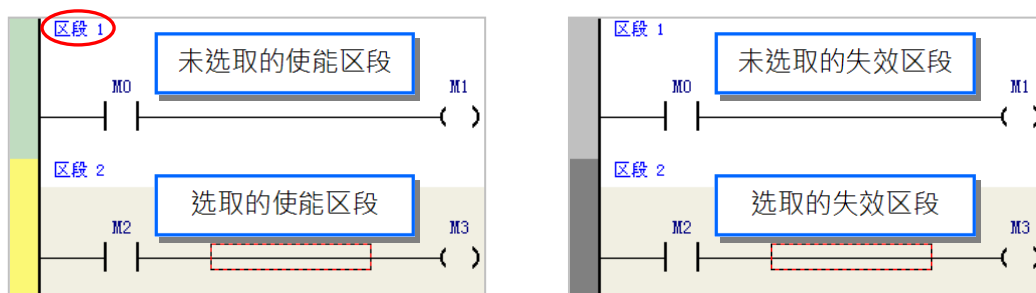
图示	键盘 (快捷键)	功能说明
	无	切换显示符号或地址
	Shift + Ctrl + C	显示/隐藏区段批注
	无	显示/隐藏装置批注与提示
	Shift + Ctrl + A	将目前选取的梯形图区段设为使能或失效状态
	Shift + Ctrl + B	在目前选取的梯形图区段上加入或取消书签标记
	Shift + Ctrl + P	移至前一个书签标记的位置
	Shift + Ctrl + N	移至下一个书签标记的位置
	Ctrl + I	于目前选取的梯形图区段之前插入一个新区段
	Shift + Ctrl + I	于目前选取的梯形图区段之后新增一个新区段
	ESC	切换为选取工具
	指令编辑模式	插入接点装置
	指令编辑模式	插入线圈装置
	指令编辑模式	新增 MPS
	指令编辑模式	插入比较接点
	指令编辑模式	选择比较接点的类型
	指令编辑模式	插入区块逻辑指令 (NP/PN/INV/FB_NP/FB_PN)
	指令编辑模式	选择区块逻辑指令的类型 (NP/PN/INV/FB_NP/FB_PN)
	Shift + Ctrl + U	插入指令或功能块


*.关于指令编辑模式请参考本章第 10.2.3 节的相关说明。

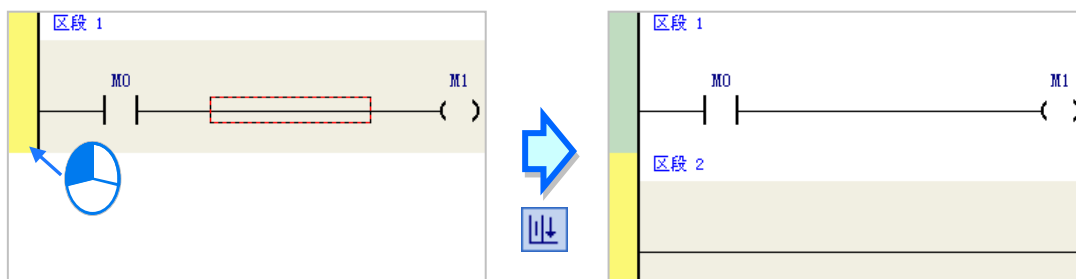
10.1.2 梯形图区段 (NETWORK)


梯形图 (LD) 程序是以区段 (NETWORK) 为编辑单位，每个梯形图区段都是一个独立的运算程序，而一个梯形图程序亦是由许多的区段所建构而成。此外，在 ISPSOft 中，每个梯形图区段可插入的组件数量并没有特别的限制，因此不会出现跳行的链接符号。

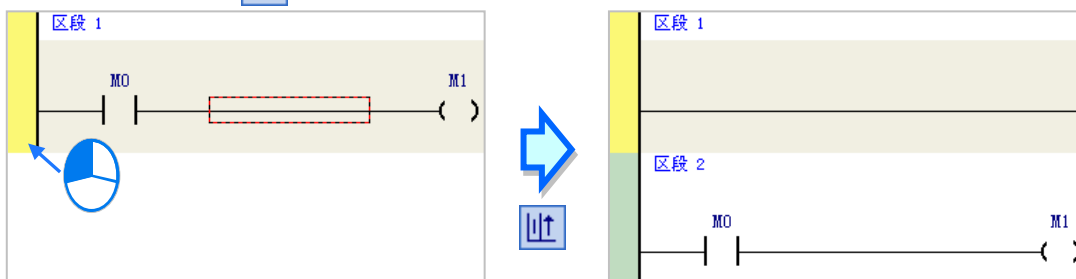
在每个梯形图区段的左上角都会标示该区段的区段编号，且由区段左侧的颜色便可得知该区段目前的状态，其状态的显示依功能可分为**使能与失效**，依编辑状态则可分为**选取与未选取**，而所显示的颜色则可于 ISPSOft 的「选项」设定中由用户自行定义，详细操作请参考第 2.3.1 节。




当新增一个梯形图程序时，系统会预先建立一个空白的梯形图区段，而要新增梯形图区段时，请先选取某个现有区段后，于图示工具栏中单击  图标即可于选取的梯形图区段下方新增一空白区段。

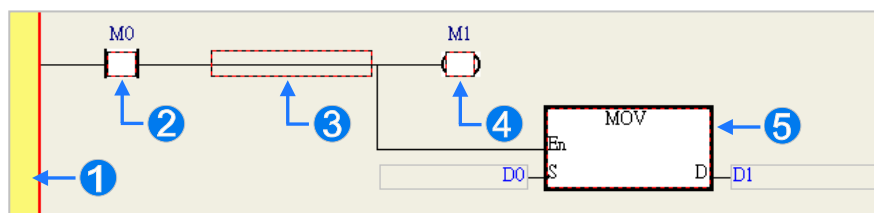


而于图示工具栏中单击  图示则可于选取的梯形图区段上方插入一空白区段。



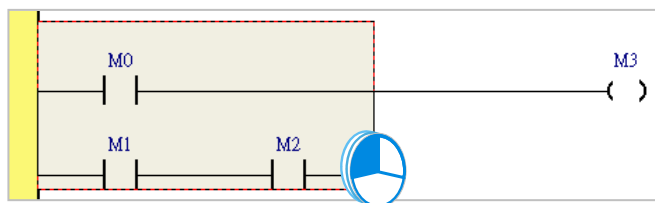
10.1.3 选取对象

要在梯形图区段中进行选取时，请先单击键盘的【ESC】键，或于图示工具栏中单击  图示，待鼠标切换为选取状态后，直接于欲选取的对象上单击鼠标左键即可。下列即为基本的选取操作。

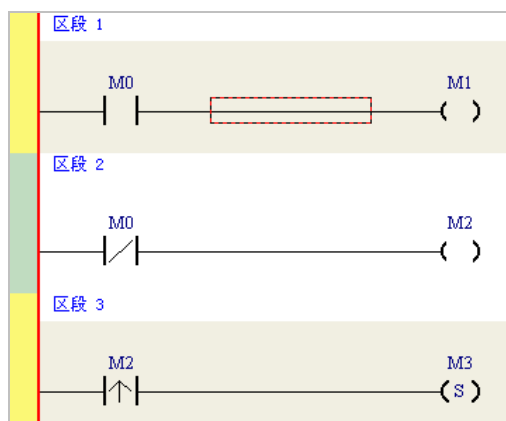


- ❶ 选取整个区段。
- ❷ 选取输入接点。
- ❸ 选取整个区段。
- ❹ 选取输出线圈。
- ❺ 选取方块。

要选取区块时，可先单击任一装置后按住鼠标左键，接着拖曳鼠标便可将区块框选起来；或是先单击区块范围的第一个对象后，单击键盘的【Ctrl】+【B】，接着再单击区块范围的最后一个对象，并再次单击【Ctrl】+【B】键便可将其框选；而框选时，该区块必须位于同一个梯形图区段中，且必须是连续位置的对象，而梯形图区段中的输入设备与输出装置则无法被框选为同一个区块。



当要同时选取多个梯形图区段时，请先按住键盘的【Ctrl】键，接着便可用鼠标分别单击多个不连续位置的区段；另外也可先按住【Shift】键后，再分别用鼠标单击欲选取范围的第一个与最后一个梯形图区段，之后便可将整个范围内的区段全部选取起来。



10.2 ISPSOft 中建立 LD 程序

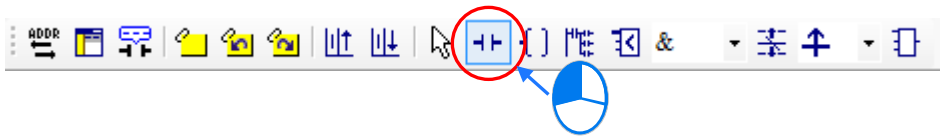
10.2.1 建立接点 (Contact)、线圈 (Coil) 及 MPS

下表即为 ISPSOft 中各种接点与线圈类型的图标。

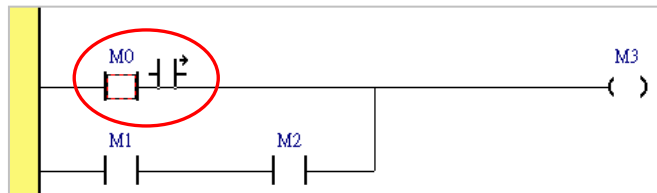
接点				线圈		
常开接点	常闭接点	上升沿触发	下降沿触发	OUT	SET	RESET

10.2.1.1 插入接点 (Contact) 与变更接点类型

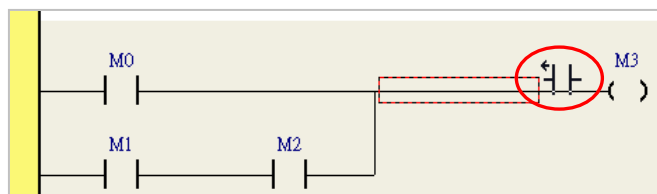
- 首先请选择欲插入接点 (Contact) 的位置，该位置可为线路、接点或区块，而所选的位置也决定了之后的串并联结构。当要与某个区块做串并联时，在执行前请先将该区块框选起来。确定插入位置之后，请于图标工具栏中单击 图示以将鼠标切换为接点工具。



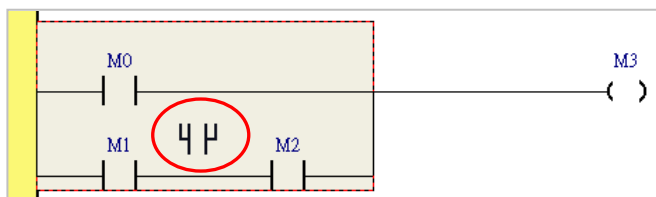
- 插入接点时，必须先将鼠标靠近欲插入的位置，而此时鼠标光标便会变为接点的图示。
 - 当鼠标靠近某个接点时，便代表要与该接点进行串并联的动作。



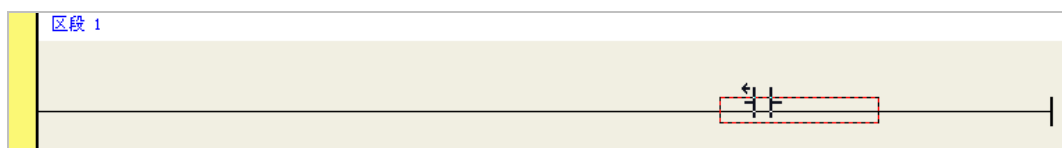
- 当鼠标靠近线路时，便代表要在该线路上插入接点。



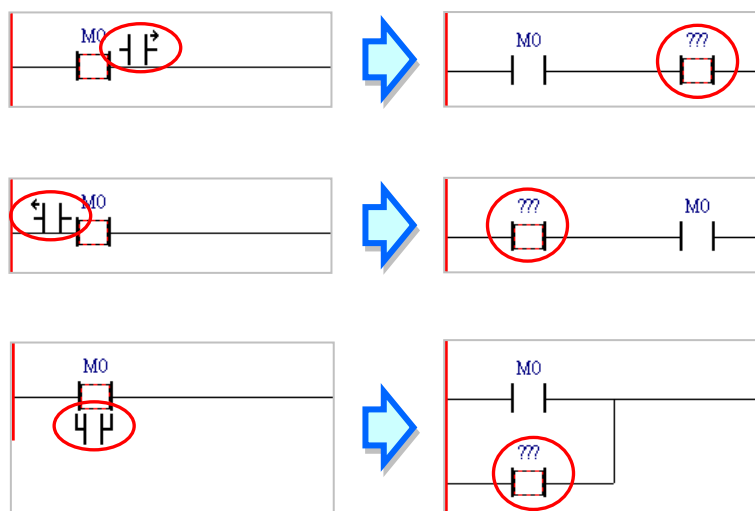
- 当鼠标靠近区块时，便代表要与该区块进行串并联。




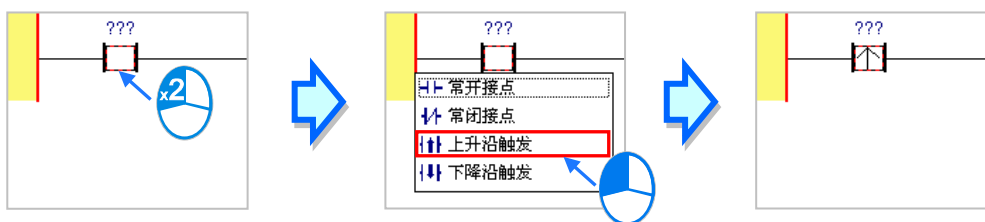
- 当要在新区段中插入接点时，请将切换为接点工具的鼠标靠近如下图线路的红框处即可。




- (3) 在欲插入的位置上刻意将鼠标往左右或下方偏移时，鼠目标图示也会跟着变化，而依据鼠目标图示便可决定新接点的插入方式。在确定插入的方式之后单击鼠标左键，接着系统便会插入一个接点，并将其自动排列在适当的位置上。



- (4) 插入接点后，单击  图标或单击键盘的【ESC】键将鼠标光标切回选取状态，接着于该接点的图标上双击鼠标左键，之后便可于下拉选单中选择接点的类型。



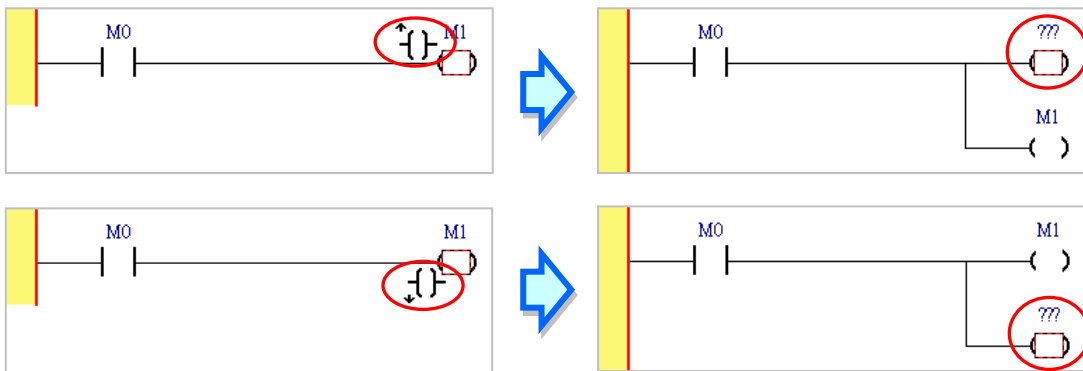
10.2.1.2 插入线圈 (Coil) 与变更线圈类型

- (1) 首先请选择欲插入线圈 (Coil) 的位置，该位置可为线路、线圈或 API 应用指令。确认插入位置后，请于图标工具栏中单击  图示以将鼠标切换为线圈工具。

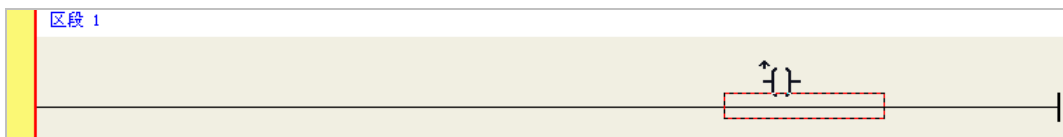



- (2) 插入线圈时，必须先将鼠标靠近欲插入的位置，而此时鼠标光标便会变为线圈的图示；当鼠标靠近某个线圈或 API 应用指令，并将鼠标往该对象的上方或下方偏移时，便代表线圈将插于该对象的上方或下方；而当鼠标靠近的位置是区段的线路上时，插入线圈之后，新线圈只会被排列在所有输出对象的最上方。

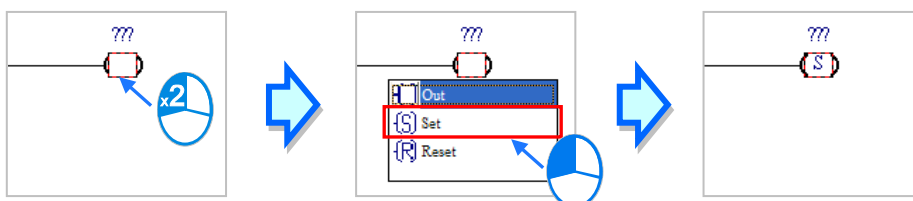
确定插入位置后单击鼠标左键，接着系统便会插入一个线圈，并将其自动排列在适当的位置上。




当区段中没有任何输出对象时，则必须将鼠标靠近选取线路的位置才可插入线圈。

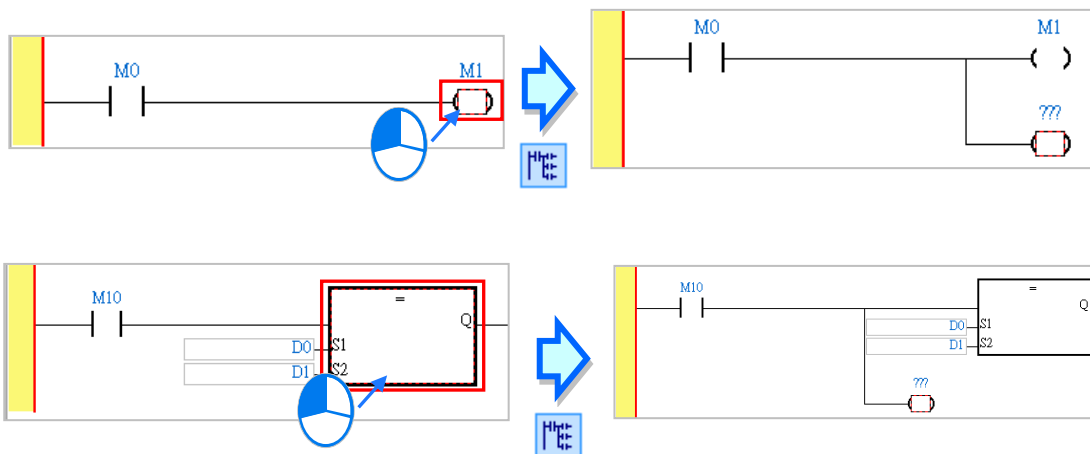


- (3) 插入线圈之后，单击  图标或单击键盘的【ESC】键将鼠标光标切回选取状态，接着于该线圈的图标上双击鼠标左键，之后便可于下拉选单中选择线圈的类型。

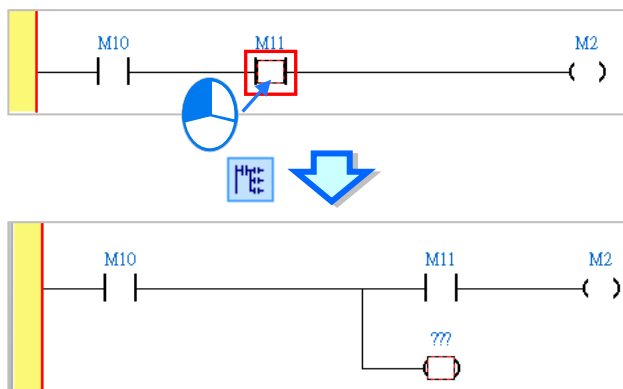


10.2.1.3 新增 MPS

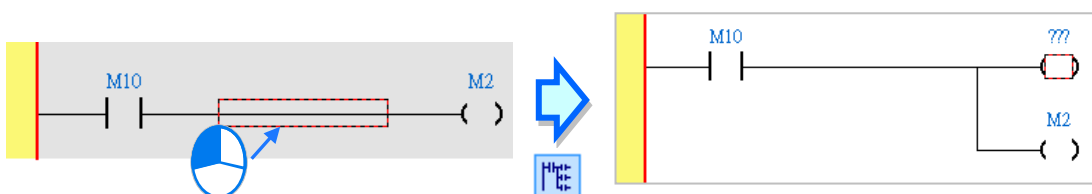
- (1) 首先请选择欲新增 MPS 的位置，该位置可为线路、线圈、比较或 API 应用指令。确认新增位置后，请在图标工具栏中单击  图示，即可新增 MPS，新增 MPS 功能为新增一个分支点。
- (2) 用户可单击线圈或比较指令，接着单击新增 MPS 钮，即可新增一个分支点。



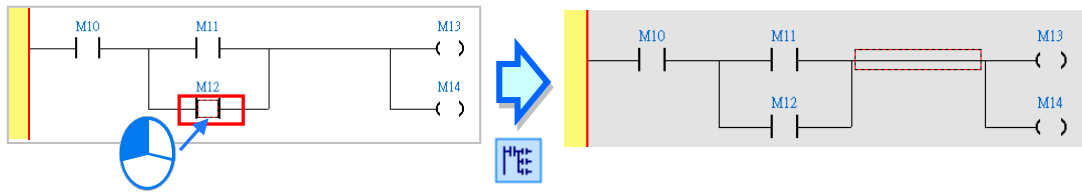
- (3) 用户可单击接点，接着单击新增 MPS 钮，即可新增一个分支点。



- (4) 用户也可单击线路，接着单击新增 MPS 钮，即可新增一个分支点。

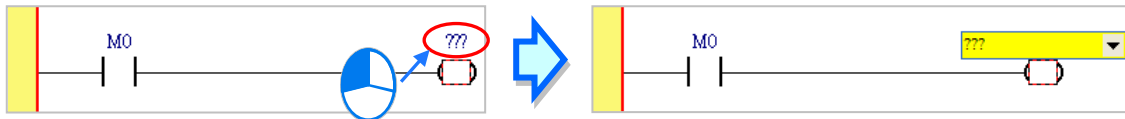


(5) 某些位置点无法新增 MPS，若单击新增 MPS 按钮，则会自动跳至一个可新增 MPS 的位置点。

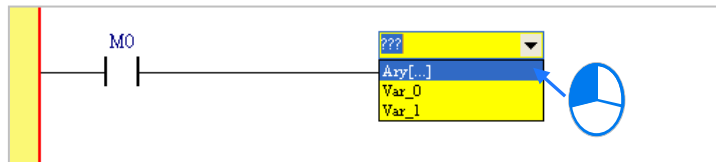


10.2.2 LD 中使用装置、变量符号及常数

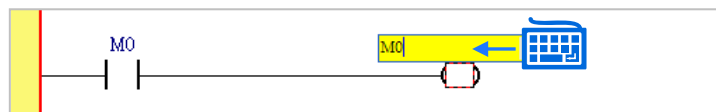
欲于插入的对象中编辑操作数时，请直接于操作数的位置上单击鼠标左键，之后该操作数便会呈现编辑状态，接着再依据下述方式进行操作即可。



直接单击编辑字段右侧的箭头按钮，接着便可于下拉选单中选择已宣告好的变量符号；而当有宣告数组符号时，于下拉选单中可挑选该数组，如下图的 Ary[...]，再于 [] 中输入数组的元素，多维数组的输入请见下述说明。



另外亦可直接于编辑字段以键盘输入设备名称或是变量符号，而当输入未经宣告的变量符号时，输入完毕后单击键盘【Enter】键，系统默认会自动导出符号宣告窗口。(设定方式请参考第 2.3.1 节)



当用鼠标单击某个区段后单击【Enter】键，接着系统便会自动开启编辑字段，而利用键盘的【Enter】键便可在同一个区段中反复移动编辑字段，利用【Tab】键则可跨区段移动，但到了最后一个编辑字段后便会停止；编辑时，用户可直接于编辑字段中使用键盘输入，该字段的下拉选单便会随之展开并过滤出符合的变量符号，利用键盘的上下键便可进行选择；完成编辑后，单击【Enter】键便会自动再切换至该区段中的下一个编辑位置，而要结束编辑时，只要单击【Esc】键即可。

补充说明

(a) 要在 ISPSOft 的梯形图 (LD) 中使用常数时，其格式的范例如下。

- 10 进制：23456 (未加任何标注的数字会被视为 10 进制)
- 16 进制：16#5BA0 (数字前方加上 16#)
- 8 进制：8#55640 (数字前方加上 8#)
- 2 进制：2#101101110100000 (数字前方加上 2#)
- 字符串：“XYZ” (以双引号包住字符串)；DVPxxMC/AS5xx 机种是用单引号包住字符串如：'XYZ'。
- BOOL：须使用常开/常闭接点 SM400/SM401 (AH/AS) 或 M1000/M1001 (DVP) 表达。

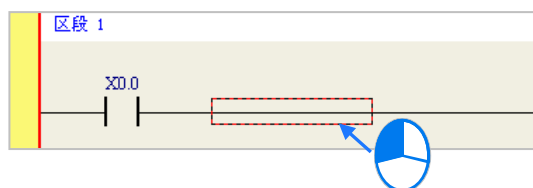
*.若用户选择 DVP15MC 机种项目，BOOL 常数以及其他的常数的格式，可参考附录 B。

- (b) 使用数组时，其输入格式为 **符号名称[索引]**，如上图下拉选单中的 Ary[...]，索引值可使用小于数组大小的 10 进制常数，且必须由 0 开始；数组最高支持三维数组，以“,”分隔，例如 Ary[1,2,3]，可用于搭配自定义数据类型使用，请参考第 8 章。

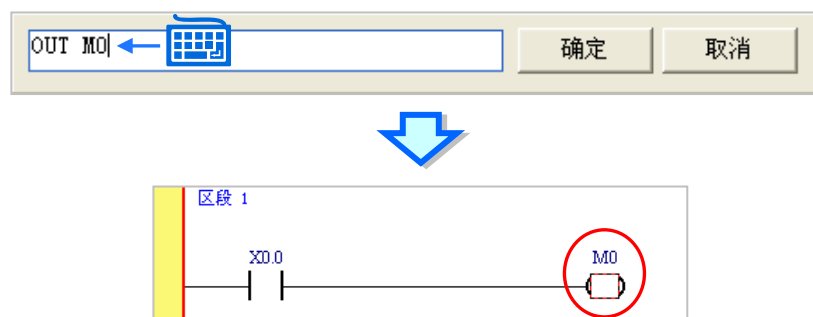
10.2.3 指令编辑模式

在 ISPSOft 的梯形图中，亦提供以直接输入指令的方式来新增装置、API 指令或功能块。

- (1) 首先于编辑工作区中单击欲插入对象的位置。

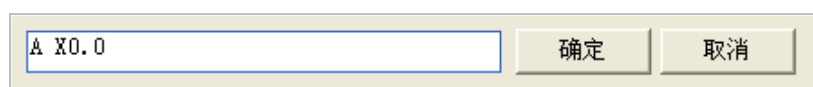


- (2) 接着直接用键盘输入指令，而一单击键盘之后，画面便会自动出现编辑窗口，待输入完毕后单击键盘的【Enter】键或以鼠标单击窗口的「确定」钮即可。(输入的指令无须分大小写)

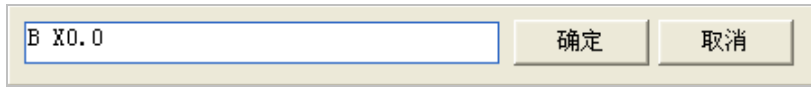


针对接点与线圈，系统亦提供简易输入的功能，请参考下图。(输入的指令无须分大小写)

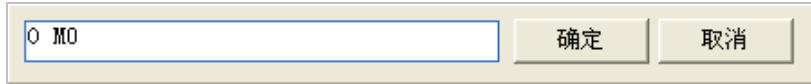
- 插入常开接点 (A 接点) 指令：“A 装置地址”



- 插入常闭接点 (B 接点) 指令 : " B 装置地址 "



- 插入输出线圈 (OUT) 指令 : " O 装置地址 "

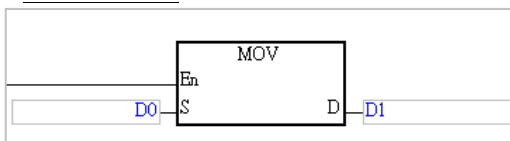


10.2.4 插入 API 应用指令与功能块

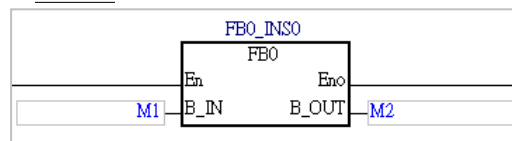
在 ISPSOft 的梯形图中 , API 应用指令与功能块是以方块的图示来呈现 , 而在方块中除了对应指令或功能块的操作数端点之外 , 尚包含一使能端点 En , 该端点的功能在于链接前方的逻辑状态 , 当前方的逻辑状态为 ON 时 , 该 API 指令或功能块才会被执行。

不过两者之间不同的是 , 功能块的输出会包含一个 Eno 端点 , 而 API 指令则没有 , 因此 API 指令在梯形图中只能被加在梯形图区段的末端 , 而无法如同功能块一般于后方串接其他组件 ; 此外 , 在功能块图标上方还会有一个 **功能块实例** 的字段 , 而关于功能块的详细介绍请参考第 7 章的内容。

- API 应用指令



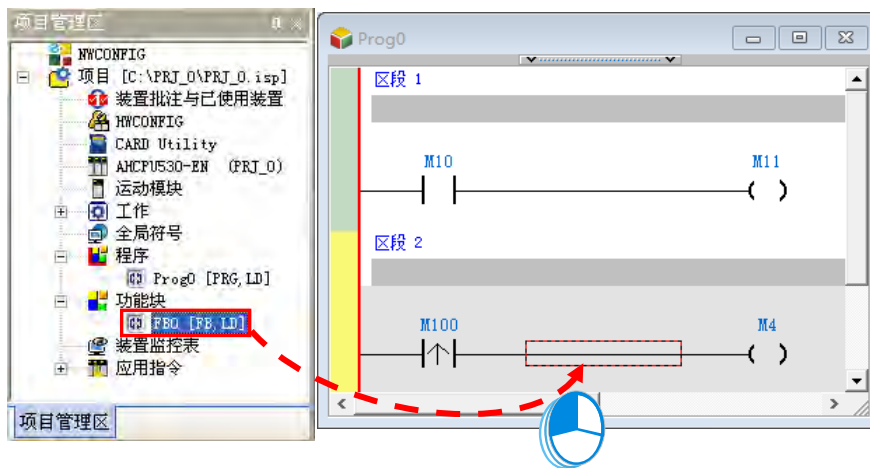
- 功能块



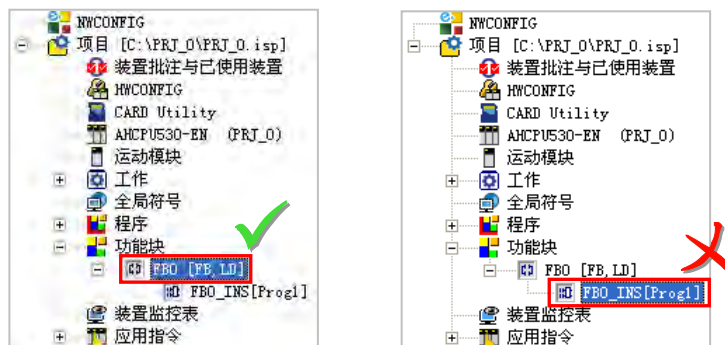
于 ISPSOft 的 LD 编辑环境中 , 针对插入 API 应用指令与功能块的操作提供了多种方式 , 说明如下。

- 方法一


于 **项目管理区** 中展开 **应用指令** 或 **功能块** 的目录 , 在其中找到欲插入的项目后 , 于该项目上按住鼠标左键 , 之后再将其拖曳至欲插入的位置后放开左键即可。

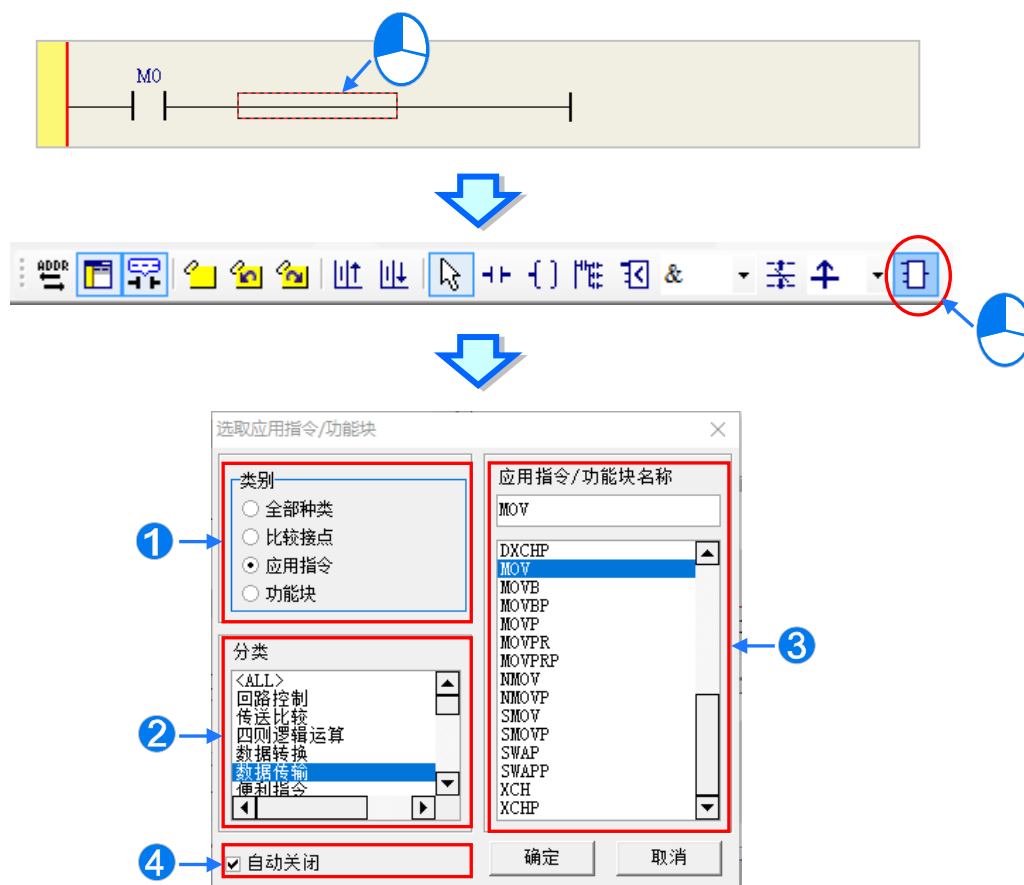


拖曳功能块时须注意只能拖曳功能块的定义，而不能拖曳其下层的功能块实例项目。



● 方法二

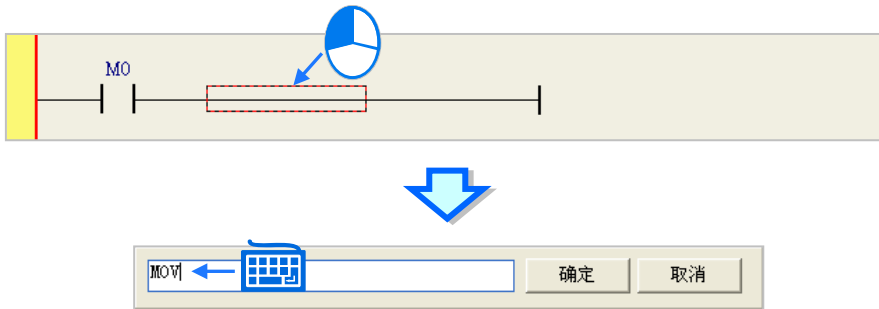
先于欲插入 API 应用指令或功能块的位置上单击鼠标左键，之后于图示工具栏中单击  图标，接着再于出现的选取窗口中单击要加入的项目，完成后单击「确定」键即可将其插入指定位置。



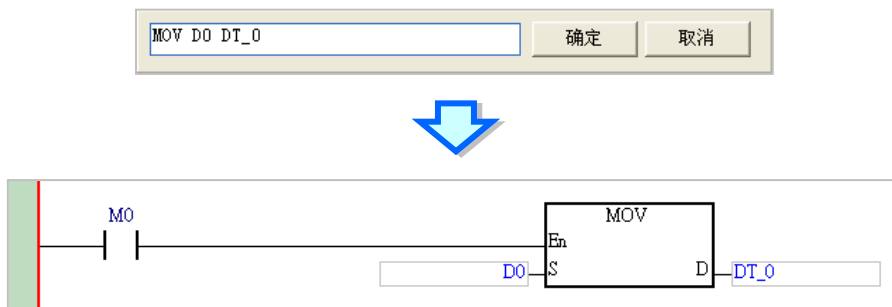
- ❶ 先选取欲加入的类别。
- ❷ 若欲加入的是 API 应用指令时，请再于此处选择指令分类。
- ❸ 于此选取 API 应用指令、比较接点或功能块。
- ❹ 若取消此处的勾选，之后便可利用此窗口连续加入指令或功能块。

● 方法三

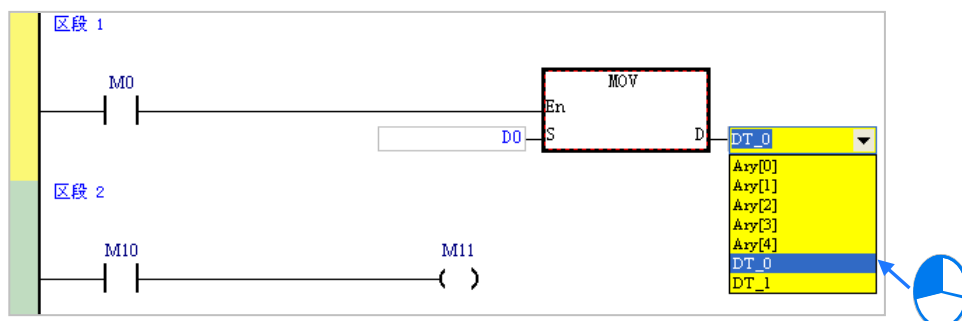
先于欲插入 API 应用指令或功能块的位置上单击鼠标左键，之后直接以键盘输入**功能块定义**或**API 应用指令**的名称，此时画面便会出现**指令编辑模式**的输入字段，输入完毕后单击键盘的【Enter】键或以鼠标单击窗口的「确定」按钮即可。(输入的指令或功能块的名称无须分大小写)



直接以**指令编辑模式**输入时，若输入的是**功能块**，则输入时只可输入功能块定义的名称，而不可加上功能块端点的操作数；但若输入的是**API 应用指令**，则输入时可依照该指令的 IL 语法一并输入指令的操作数，且输入的操作数数量无须完整，系统将会依序将其插入对应的操作数字段，并让缺少的操作数字段维持「???'的显示；此外，若输入的操作数包含未经宣告的变量符号时，确认输入之后，系统亦会自动导出符号宣告窗口。详细的指令语法则请参考各机种的指令手册。

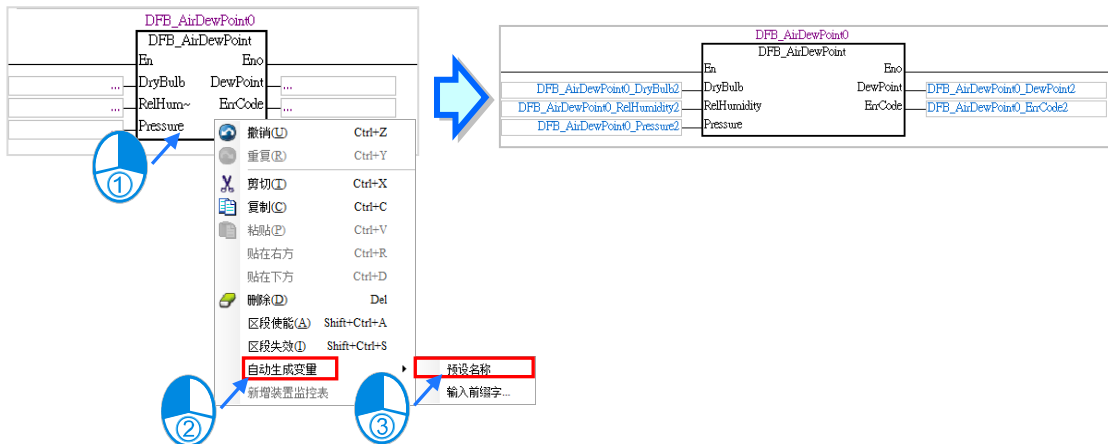


成功插入 API 应用指令或功能块之后，请再参考第 10.2.2 节的说明，逐一完成每个操作数的编辑。

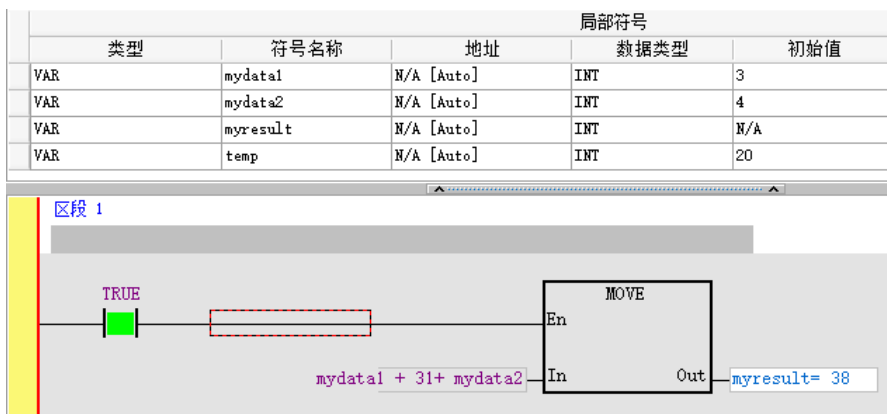


*. 自动导出符号宣告窗口的功能可于「选项」当中进行设定，设定方式请参考第 2.3.1 节的说明。

可以在功能块上方点击鼠标右键，选取自动生成变量的功能，选择默认名称，会以功能块的名称+引脚名称，产生变量并填入引脚位置，如下图；而选择输入前缀字，则是会以输入的文字，产生变量并填入引脚位置。

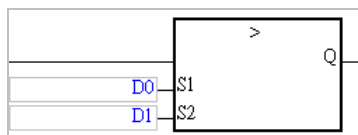


在 DVPxxMC 与 AS5xx 机种的功能块引脚支持 ST 表达式的写法，如下图。

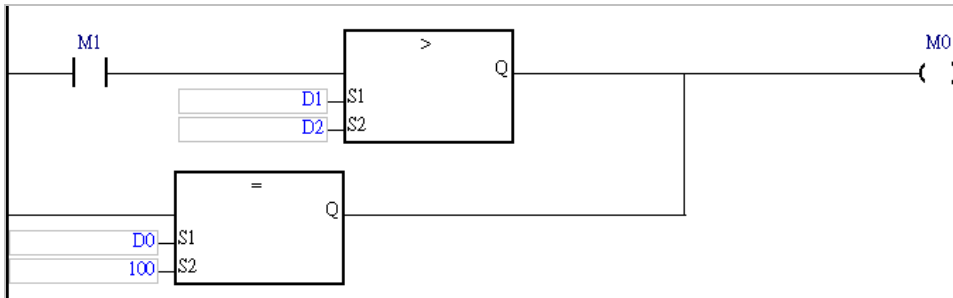


10.2.5 编辑比较接点


下图是 ISPSOft 中的比较接点图标，方块图标上方的「>」则是比较的类型，「S1」与「S2」端点则为比较的对象，其接受的数据格式决定于比较的类型，而输出「Q」则可直接趋动线圈或是串联接点。

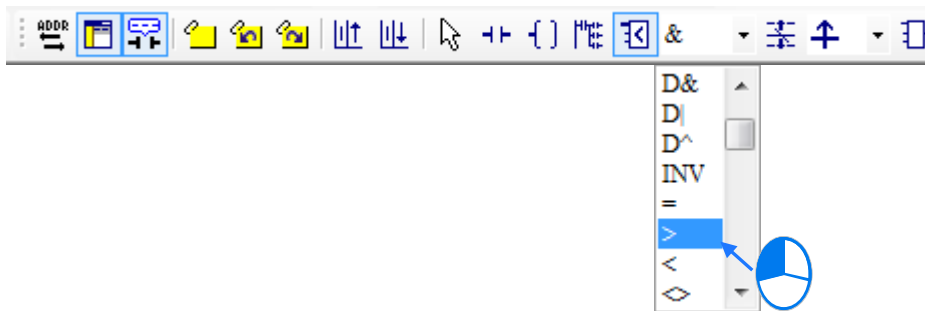


在 LD 的程序中，比较接点可如同一般的接点而与其它接点进行串并联的逻辑运算。




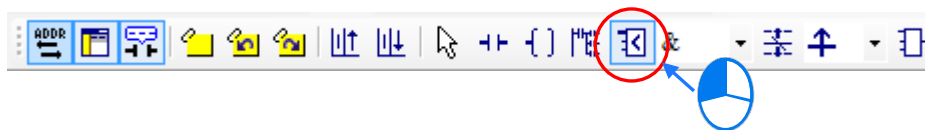
于 LD 程序插入比较接点的方式与插入 API 应用指令相同，操作方式请参考第 10.2.4 节的相关说明。而除此之外，亦可使用如同插入一般接点 (Contact) 的方式来插入比较接点，说明如下。

- (1) 首先请于图示工具栏的  字段下拉选择比较类型。

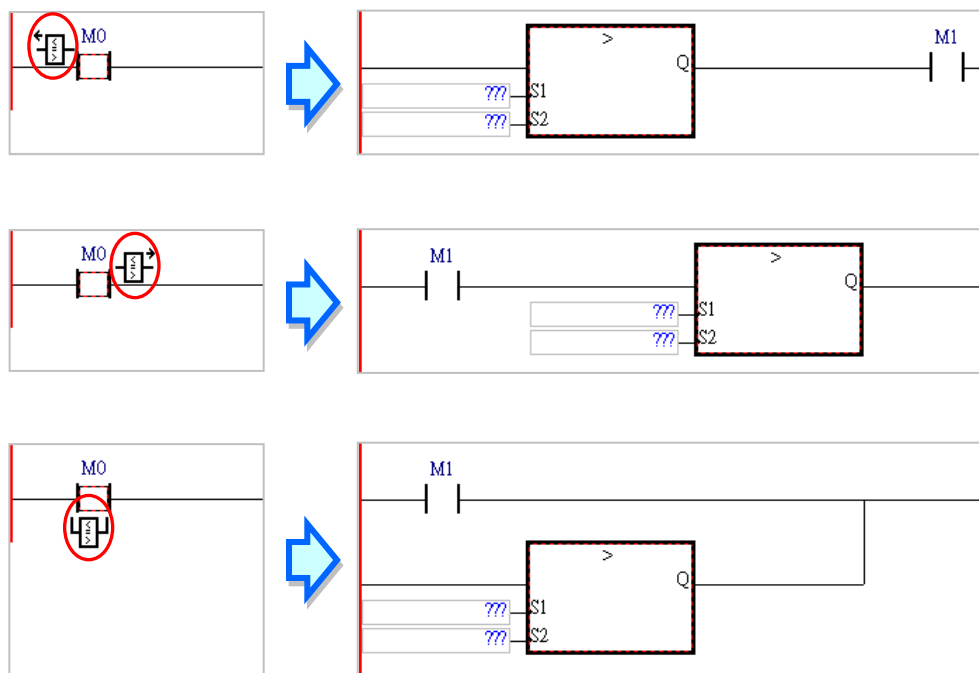


- (2) 接着请选择欲插入比较接点的位置，该位置可为线路、接点或区块，而所选的位置也决定了之后的串并联结构。当要与某个区块做串并联时，在执行前请先将该区块框选起来。

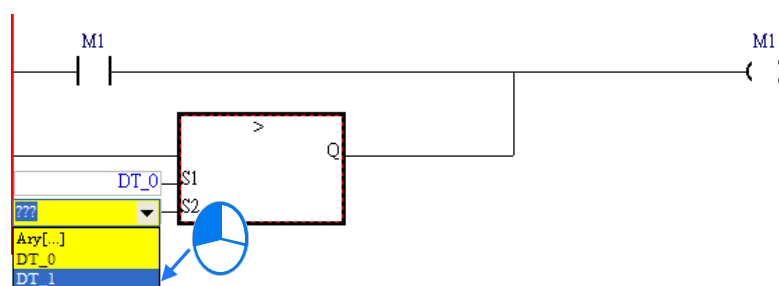
确定插入位置之后，请于图标工具栏中单击  图示以将鼠标切换为比较接点工具。



插入接点时，必须先将鼠标靠近欲插入的位置，而此时鼠标光标便会变为比较接点的图示；而在该位置上刻意将鼠标往左右或下方偏移时，鼠目标图示也会跟着变化，而依据鼠目标图示便可决定新接点的插入方式。在确定插入的方式之后单击鼠标左键，接着系统便会插入一个接点，并将其自动排列在适当的位置上。

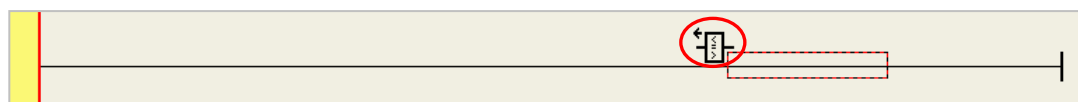


(3) 成功插入比较指令之后，请参考第 10.2.2 节的说明，逐一完成每个操作数的编辑。



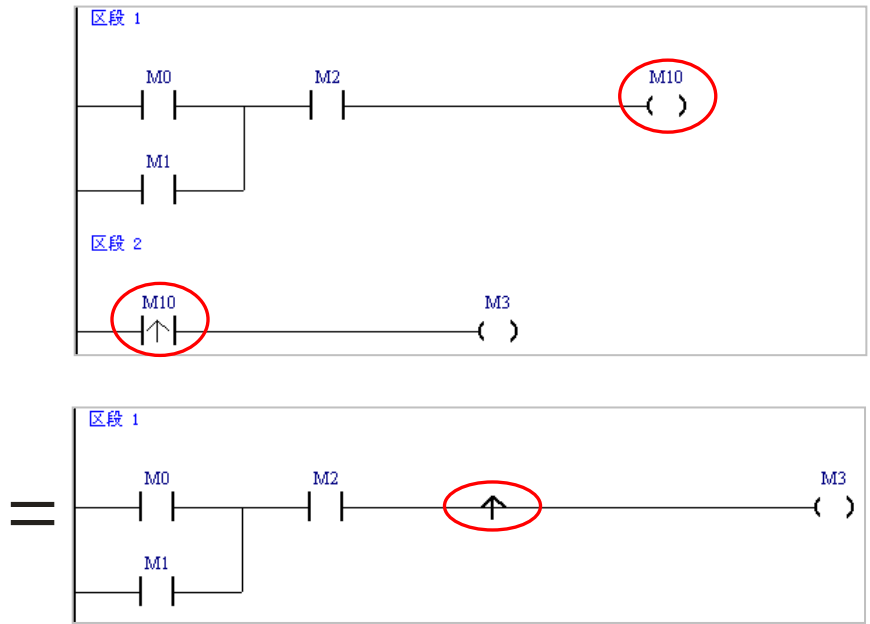
补充说明

当要在新区段中插入比较接点时，必须将切换为比较接点工具的鼠标靠近线路才可将其插入。



10.2.6 编辑区块逻辑指令

如下图，当要将某个区块的逻辑运算结果拿来作上升沿触发操作时，传统的编辑方式必须将其分为两个梯形图区段，而使用区块逻辑指令便可将其合并为同一个梯形图区段。



下表即为 ISPSOft 中所提供的区块逻辑指令及各机种可以使用指令在 POU 程序或 FB 功能块的支持列表。

梯形图标	指令名称	功能说明	AH/AS		DVP		AS5xx DVPxxMC	
			POU	FB	POU	FB	POU	FB
	NP	当前方区块的逻辑状态由 OFF 变为 ON 时才会输出 ON。	✓	✓	✓	✓		
	PN	当前方区块的逻辑状态由 ON 变为 OFF 时才会输出 ON。	✓	✓	✓	✓		
	INV	将前方区块的逻辑状态反相后输出。	✓	✓	✓	✓	✓	✓
	FB_NP	功能与 NP 类似，唯须额外指定一暂存装置或变量。		✓		只支持 ES3		
	FB_PN	功能与 PN 类似，唯须额外指定一暂存装置或变量。		✓		只支持 ES3		

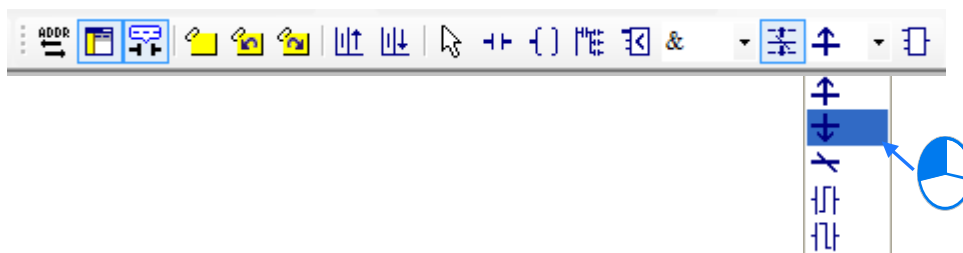
* 关于 FB_NP 与 FB_PN 指令请查阅各机种程序手册，或参考第 7.3.2 节当中的相关说明。




下列则为插入区块逻辑指令的方式。

● 方法一

- (1) 请先于图示工具栏的  字段下拉选择欲插入的指令类型。

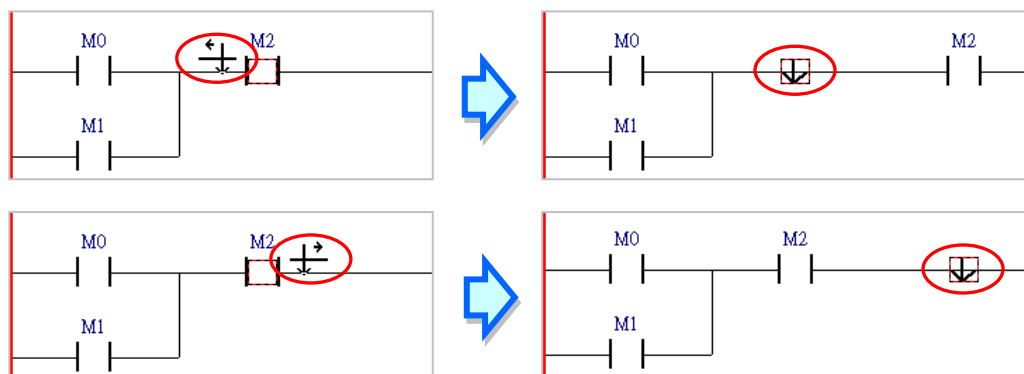


- (2) 单击工具栏图示  将鼠标光标切换为区块逻辑指令的状态。



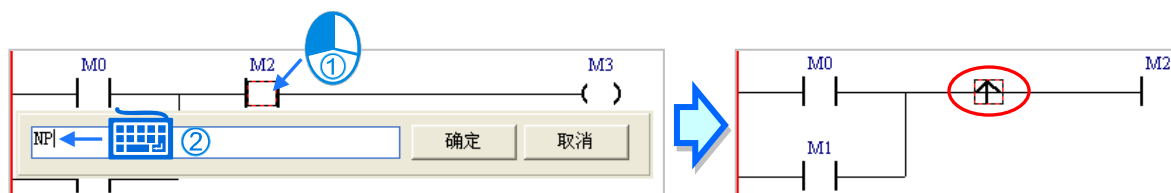
(3)

- (4) 当将鼠标光标靠近欲插入的位置时，鼠标光标便会变为区块逻辑指令的图标，而刻意将鼠标往该位置的左方或右方偏移时，鼠标便会随之变化以代表插入的方式。在确定插入的方式之后单击鼠标左键，接着系统便会插入一个区块逻辑指令，并将其自动排列在适当的位置上。



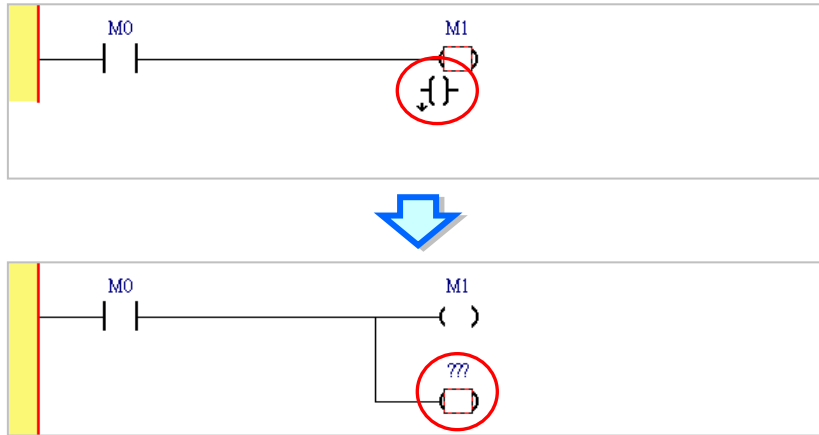
● 方法二

先单击欲插入的位置之后，直接以键盘输入区块逻辑指令的名称，此时画面便会出现指令编辑模式的输入字段，输入完毕后单击键盘的【Enter】键或以鼠标单击窗口的「确定」钮，此时系统便会于所选位置的前方插入输入的指令图标。(输入的指令无须分大小写)

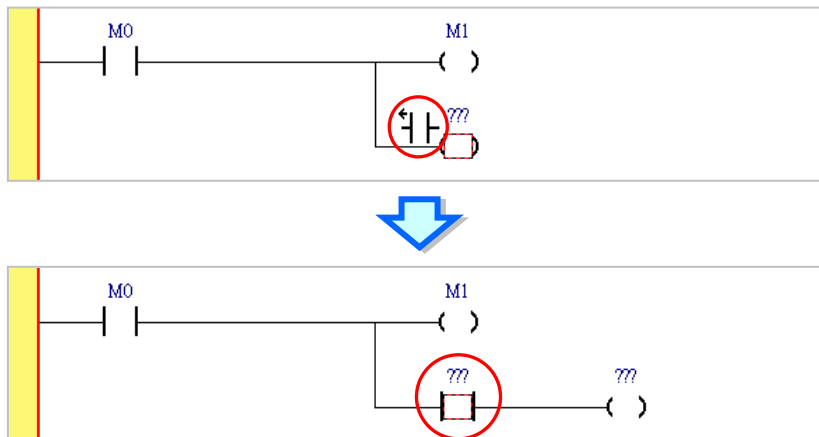


10.2.7 编辑多重输出

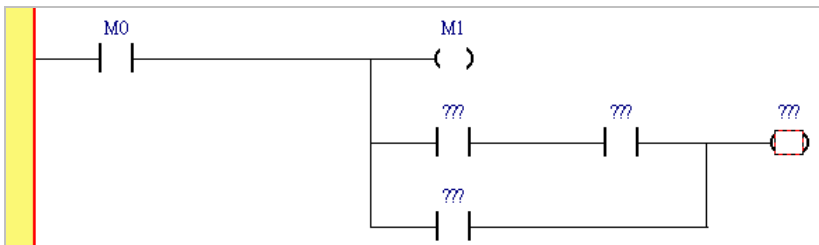
(1) 欲编辑多重输出时，请先于线路上完成输出的线圈或 API 应用程序。



(2) 接着请将鼠标切换为接点或比较接点工具，之后再将鼠标靠近输出对象后单击鼠标左键。



(3) 接下来便可依据前述的说明逐一完成多重输出的结构。

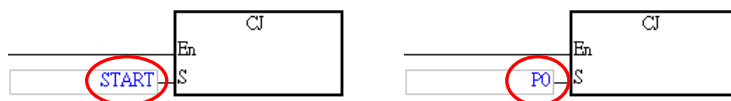


10.2.8 设置标签 (Label)

当要使用跳跃指令，如 CJ、JMP...时，便必须为跳跃的目标区段设置一个卷标 (Label)，而卷标的位置则位于每个区段编号的右侧。

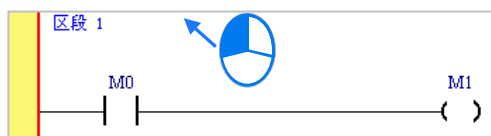


要在指令中使用卷标 (Label) 时，卷标符号无须再加上冒号『:』。

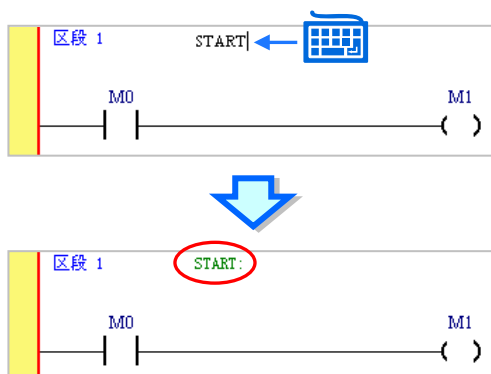


下列则为设置标签 (Label) 的操作方式。

- (1) 在欲设置标签的区段编号右侧单击鼠标左键。

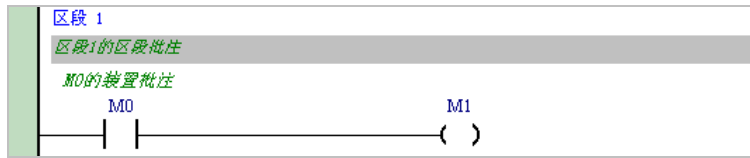


- (2) 请输入卷标符号名称；而输入时，标签后的冒号『:』可直接省略，系统之后会自行将其补上。完成之后单击【Enter】键即可。




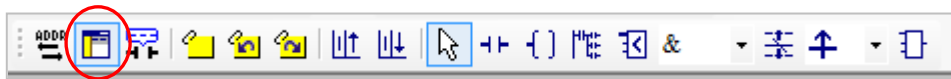
10.2.9 编辑批注

在 ISPSOft 的梯形图画面中，除了可直接在程序画面编辑**装置批注**外，用户亦可替每个梯形图区段加上**区段批注**，而区段批注则位于每个梯形图区段的编号下方。

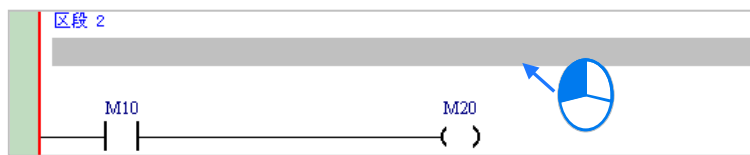


● 编辑区段批注

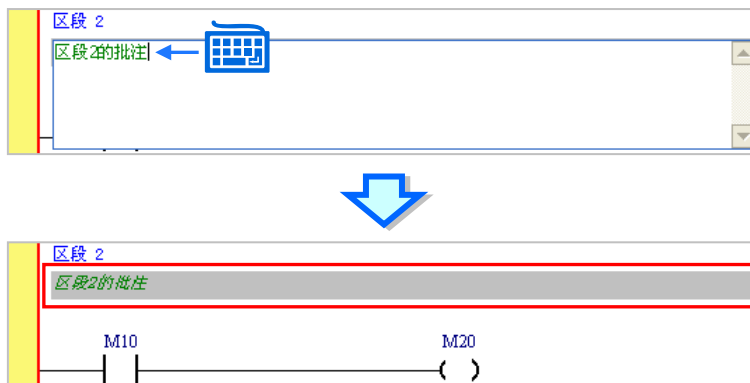
- (1) 当梯形图工具栏中的  图示为单击的状态时，于每个区段编号的下方便会显示该区段的**区段批注**；而同样的，当要隐藏**区段批注**时，请直接将该图示恢复为未单击的状态即可。




- (2) 于**区段批注**的字段上单击鼠标左键，之后便会出现**区段批注**的输入字段。

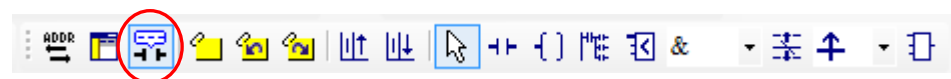


- (3) 接着请直接在出现的编辑字段中编辑批注，其中单击键盘的【Shift】+【Enter】即可换行输入；输入完毕后，直接单击键盘的【Enter】键或用鼠标单击他处即可。

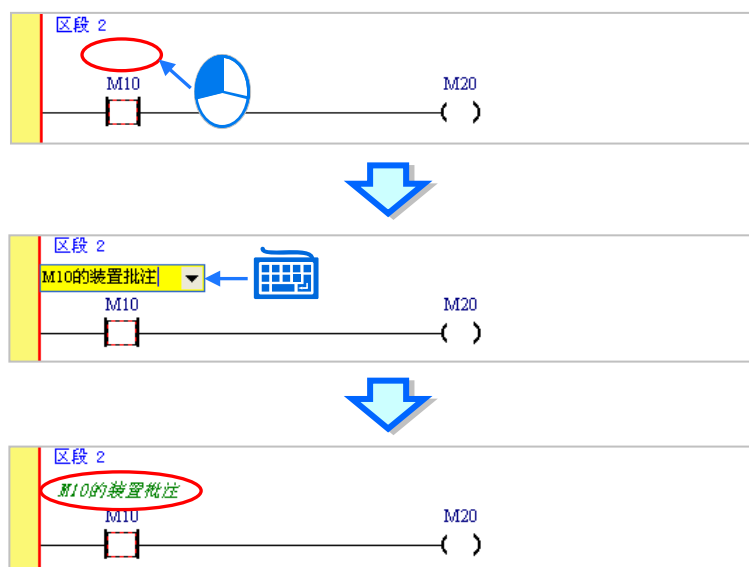


● 编辑装置批注


- (1) 当梯形图工具栏中的  图示为单击的状态时，在每个装置编号的上方便会显示**设备批注**；而同样的，当要隐藏**装置批注**时，请直接将该图示恢复为未单击的状态即可。




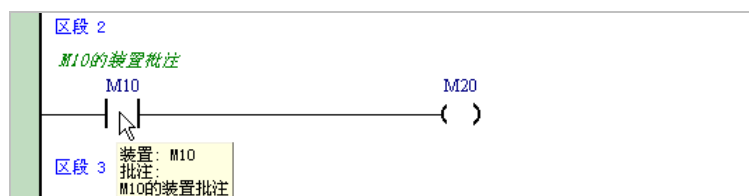
- (2) 于**装置批注**的字段上单击鼠标左键，接着便可直接于该字段中编辑**装置批注**，也可以对 API 的 PIN 脚编辑批注；输入完毕后，直接单击键盘的【Enter】键或用鼠标单击他处即可。




补充说明


若梯形图工具栏的  图标为单击的状态，当程序中的接点或线圈是使用符号名称，且该变量符号有编辑符号批注时，画面中亦会显示该符号的**符号批注**，但因**装置批注**与**符号批注**在 ISPSOft 中是属于不同的数据，因此用户将无法直接于程序画面中编辑符号批注；若要编辑符号批注时，用户仍必须至符号表中才可进行编辑，相关操作请参考第 6.2.4 节。

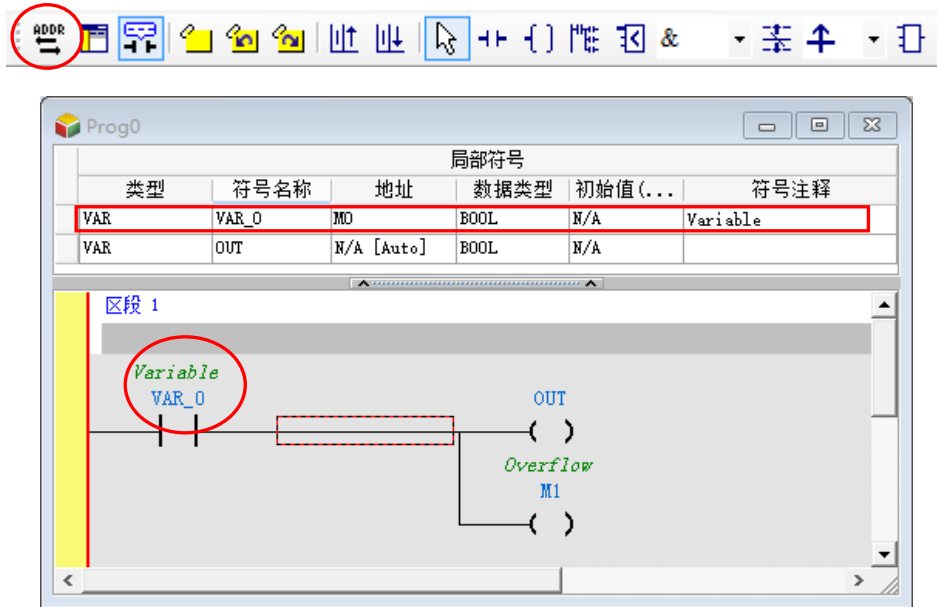
当梯形图工具栏的  图标为单击的状态时，若将鼠标移至某个装置的上方稍作停留，则此时画面便会出现该装置的相关提示，而提示的内容则可于 ISPSOft 的「选项」功能中进行设定，请参考手册中第 2.3.1 节的相关说明。




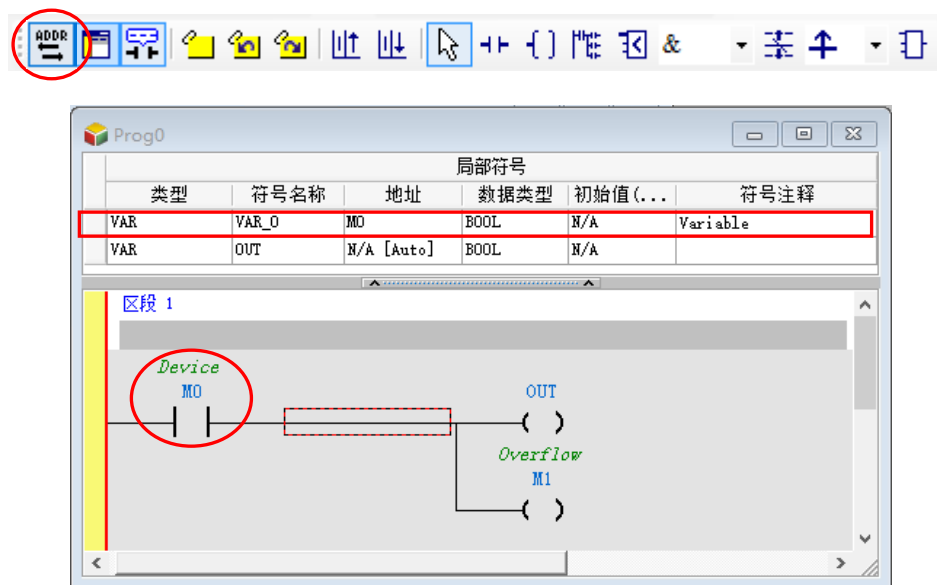
10.2.10 符号模式与地址模式的显示切换

在 LD 与 FBD 的编辑环境中，当某个变量符号在宣告时有一并指定装置地址时，藉由切换图示工具栏的  图标便可变更该变量符号的显示模式；但未指定装置地址的变量符号，或未被变量符号所指定的装置地址则会保持原状。

如下图，当  图示呈现未单击的状态时，代表目前的显示方式为**符号模式**，此时有指定装置地址的变量符号便会以**符号名称**来显示，而组件上方的批注亦会显示为**符号批注**。





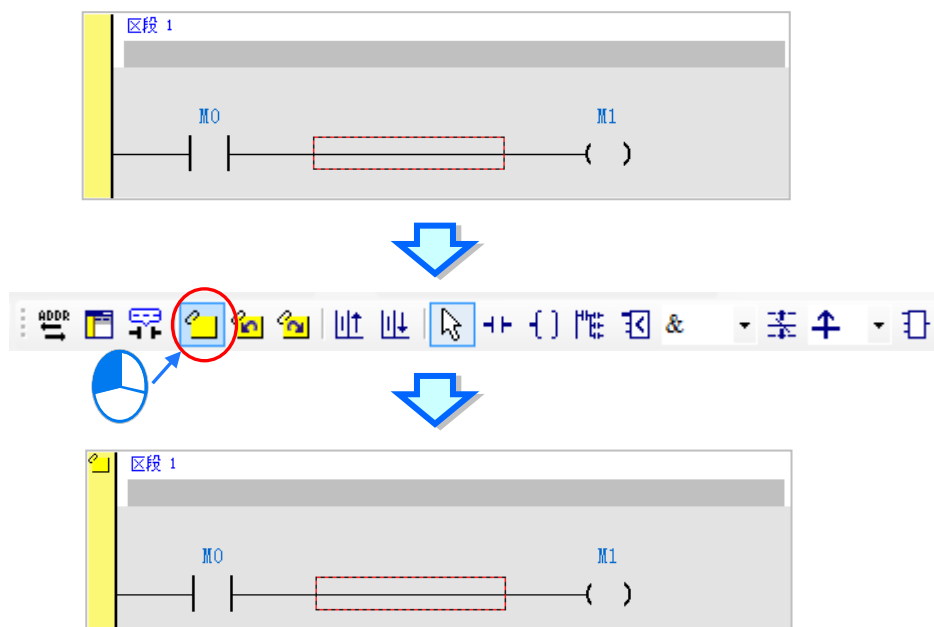
如下图，当  图标被单击之后，便代表目前的显示方式为**地址模式**，此时有指定装置地址的变量符号便会以**装置名称**来显示，而组件上方的批注亦会显示为**装置批注**。





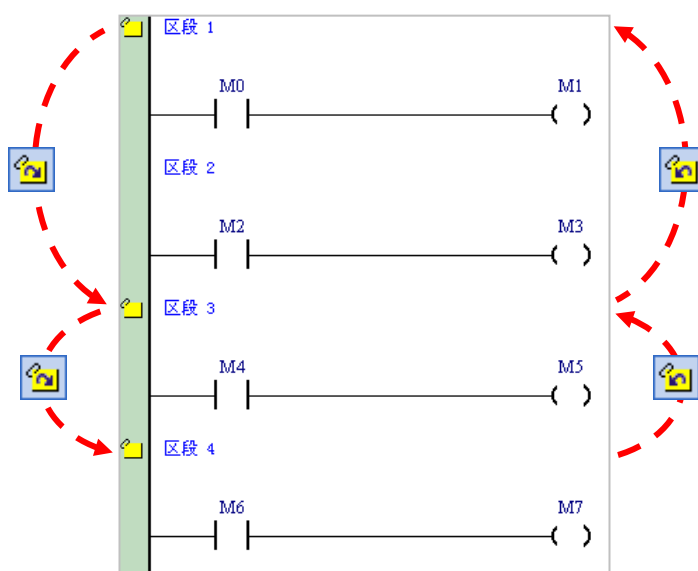
10.2.11 书签功能



书签的功能主要在于协助用户迅速搜寻或移动至加入书签标注的梯形图区段。

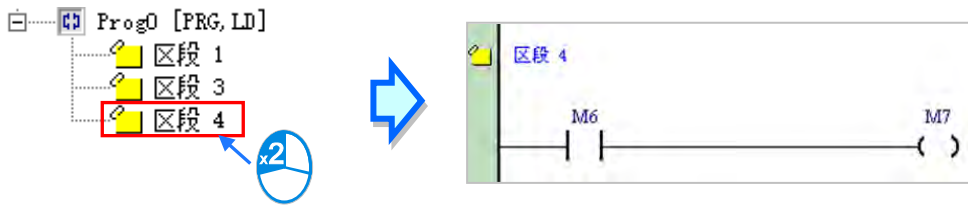
- (1) 单击图示工具栏的  图标便可于目前选取的区段中加入一书签；若要取消书签时，只要选取该区段后，再次单击  图标即可。



- (2) 当程序中有插入书签时，单击工具栏图标  即可移动至下一个书签的位置，而单击  图标则可移动至前一个书签的位置。



- (3) 在程序中的区段插入书签，会同时出现  图案在项目管理区相对应的程序下方，鼠标双击  区段编号，即可移动至书签所在的位置。



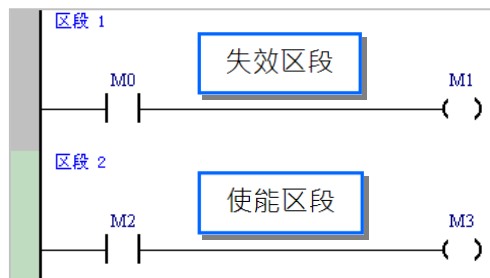
补充说明

于功能工具栏的 **编辑** > **书签** 项目中亦有相同的操作项可供使用，其中单击「**移除所有书签**」项目则可同时将所有的书签移除，不过此操作仅会针对目前正在编辑的窗口产生作用。

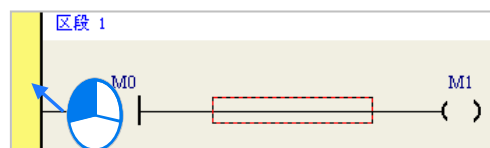
10.2.12 梯形图区段的使能与失效

若某个梯形图区段被设为失效状态，则程序的编译动作便会主动略过该区段，因此当用户在进行测试或除错的工作时，利用此功能便可暂时关闭程序中的某些程序。

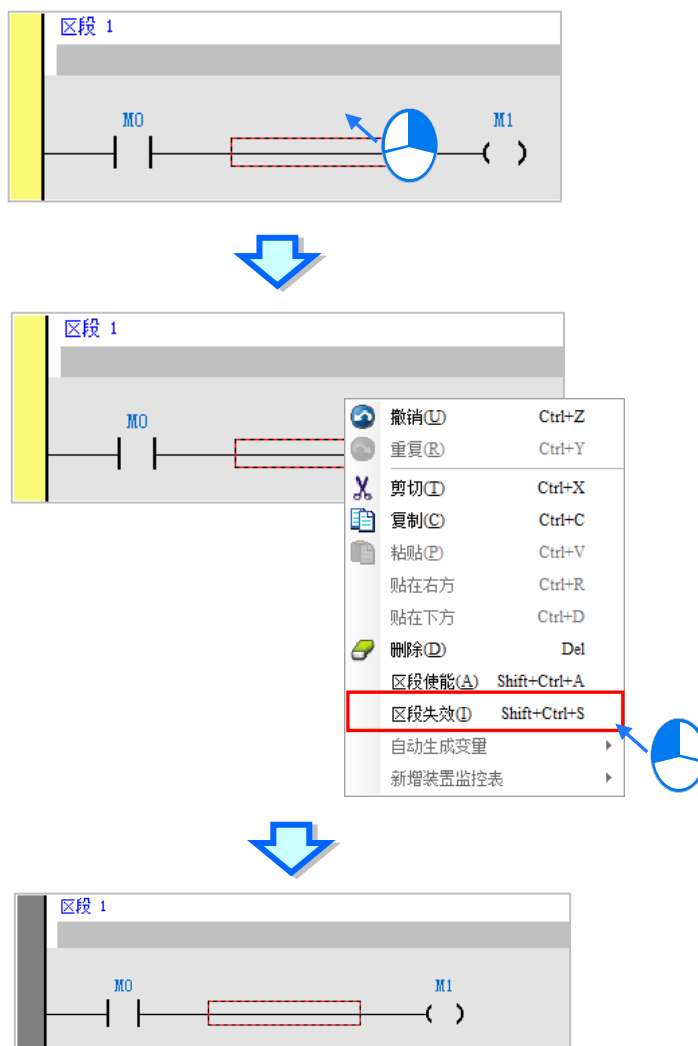
设为**使能**或**失效**的区段，在该区段的左侧会使用不同的颜色来表示，而显示的颜色则可于 ISPSOft 的「**选项**」功能中进行设定，请参考手册中**第 2.3.1 节**的相关说明。



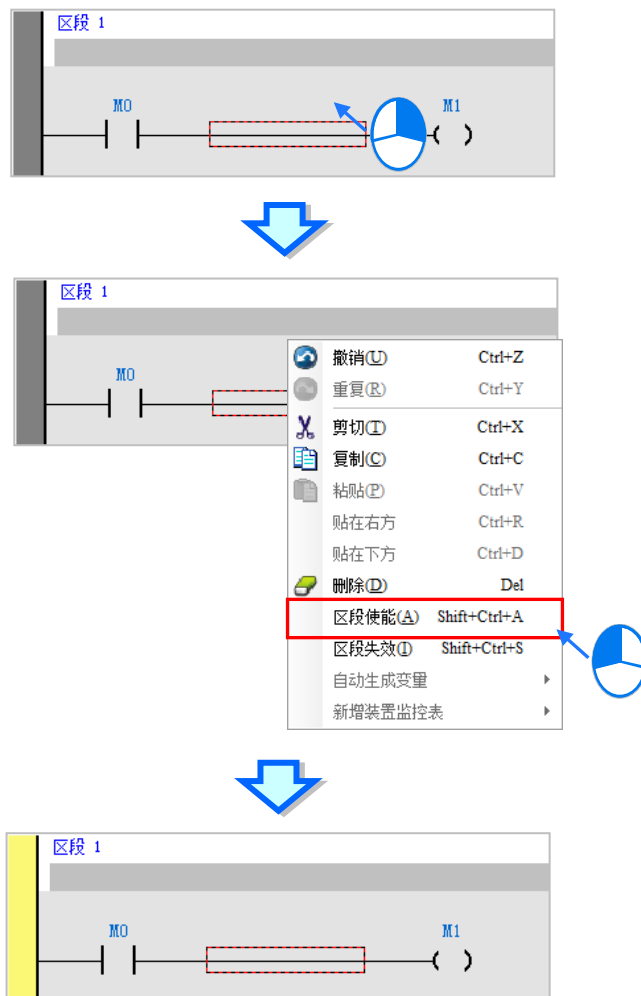
要变更区段的状态时，请先将该区段选取起来，而选取时亦可同时选取多个区段。



于功能工具栏中依序单击 **编辑 (E)** > **区段失效 (L)** 或是在选取的区段上单击鼠标右键，接着于快捷选单中选取「**区段失效 (L)**」，则此时所选区段则会为失效状态。



若再单击编辑 (E) > 区段使能 (A) 或是在选取的区段上单击鼠标右键，接着在快捷选单中选取「区段使能 (A)」，则会恢复使能状态。



第11章 功能块图 FBD

目录

11.1	功能块图 FBD 简介	11-2
11.1.1	关于功能块图 (FBD)	11-2
11.1.2	ISPSOft 的 FBD 编辑环境	11-2
11.1.3	功能块图区段 (NETWORK)	11-4
11.1.4	选取对象	11-5
11.2	ISPSOft 中建立 FBD 程序	11-6
11.2.1	输入节点与输出节点	11-7
11.2.2	FBD 中使用装置、变量符号及常数	11-10
11.2.3	AND 与 OR 方块	11-11
11.2.4	反相逻辑	11-14
11.2.5	上沿触发输入与下沿触发输入	11-16
11.2.6	置位 (SET) 输出与复位 (RESET) 输出	11-18
11.2.7	API 应用指令、比较接点与功能块	11-20
11.2.8	置位标签 (Label)	11-24
11.2.9	批注与提示	11-25
11.2.10	符号模式与地址模式的显示切换	11-27
11.2.11	书签功能	11-28
11.2.12	功能块图区段的使能与失效	11-29

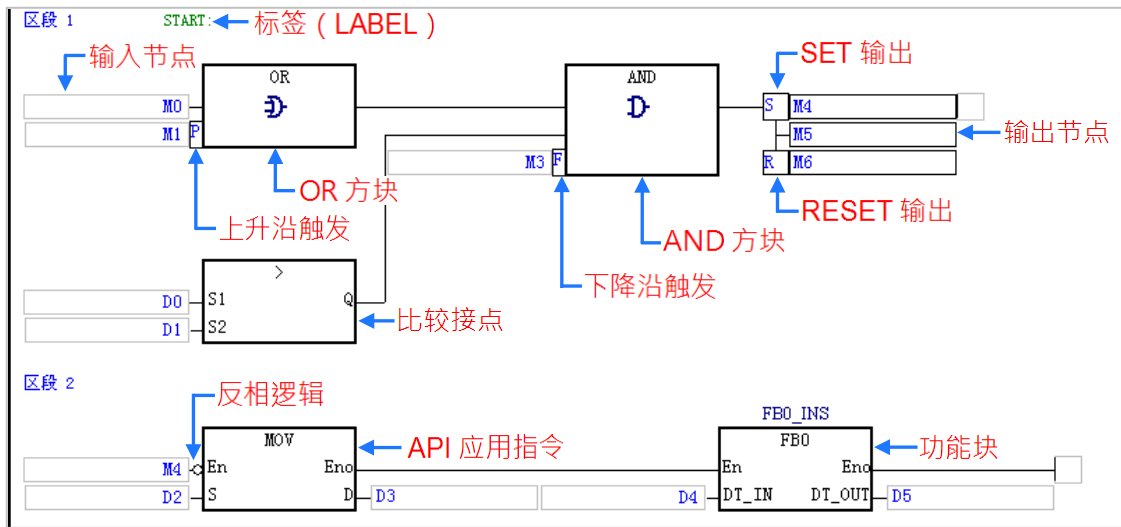
11.1 功能块图 FBD 简介



11.1.1 关于功能块图 (FBD)

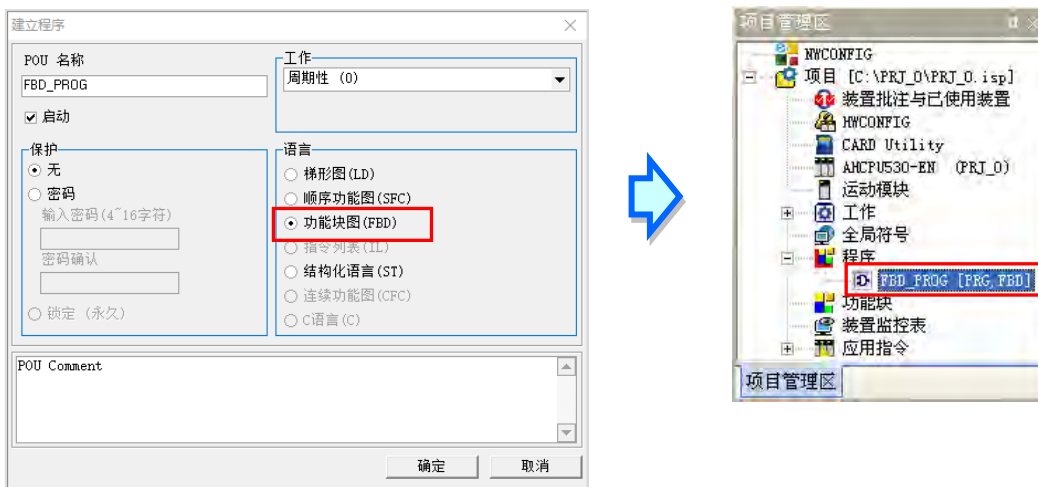
功能块图 (**Function Block Diagram, FBD**) 也是符合 IEC 61131-3 的编程语言，与梯形图一样属于图形化的编程方式，不同的是梯形图是以接点为主，类似传统的控制回路图，而 FBD 则是以功能块为主，呈现方式较接近于系统方块图。FBD 的特点在于着重功能块的模块化优势，编程时可先将整个程序有系统的切割成许多功能块之后，接着再于 FBD 程序中将各个功能块重新链接部署即可。

下图即是 ISPSOft 当中的 FBD 程序。



11.1.2 ISPSOft 的 FBD 编辑环境

在建立 POU 时，于语言字段选择「功能块图 (FBD)」便可建立功能块图程序，请参考第 5.4.1 节。



下图即为 ISPSOft 中的功能块图 (FBD) 编辑环境，窗口的上方为区域符号表，下方则是编辑工作区。



进入 FBD 的编辑环境后，ISPSOFT 的窗口中便会出现对应的编程图标工具栏，其功能简介如下。



图示	键盘 (快捷键)	功能说明
	无	切换显示符号或地址
	Shift + Ctrl + C	显示/隐藏区段批注
	无	开启/关闭提示功能
	Shift + Ctrl + A	将目前选取的功能块图区段设为使能或失效状态
	Shift + Ctrl + B	在目前选取的功能块图区段上加入或取消书签标记
	Shift + Ctrl + P	移至前一个书签的标记位置
	Shift + Ctrl + N	移至下一个书签的标记位置
	Ctrl + I	于目前选取的功能块图区段之前插入一个新区段
	Shift + Ctrl + I	于目前选取的功能块图区段之后新增一个新区段
	Shift + F1	插入 AND 方块
	Shift + F2	插入 OR 方块
	Shift + F3	插入输出节点
	Shift + F4	增加 AND/OR 方块的输入端点
	Shift + F5	删除 AND/OR 方块的输入端点
	Shift + F6	插入反相逻辑
	Shift + F7	插入置位(SET)/复位(RESET)输出
	Shift + F8	插入上沿/下沿触发
	Shift + Ctrl + U	插入指令或功能块

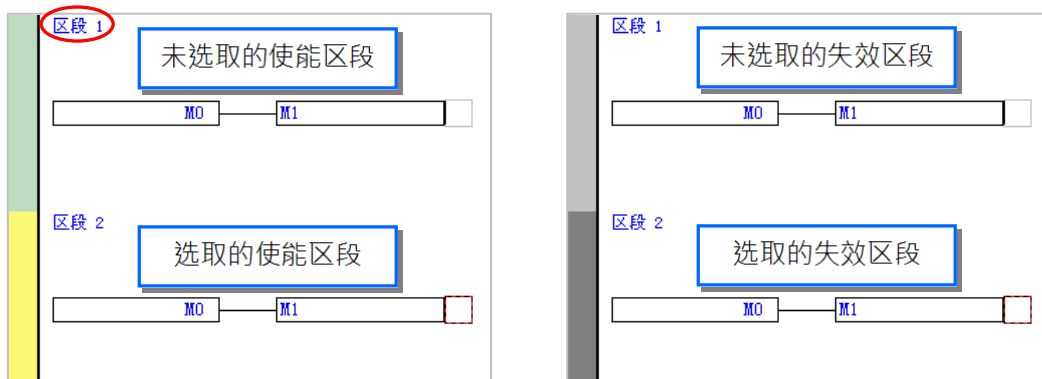
*.插入指令与功能块亦可使用键盘直接输入的方式，请参考本章第 11.2.7 节的相关说明。


11.1.3 功能块图区段 (NETWORK)

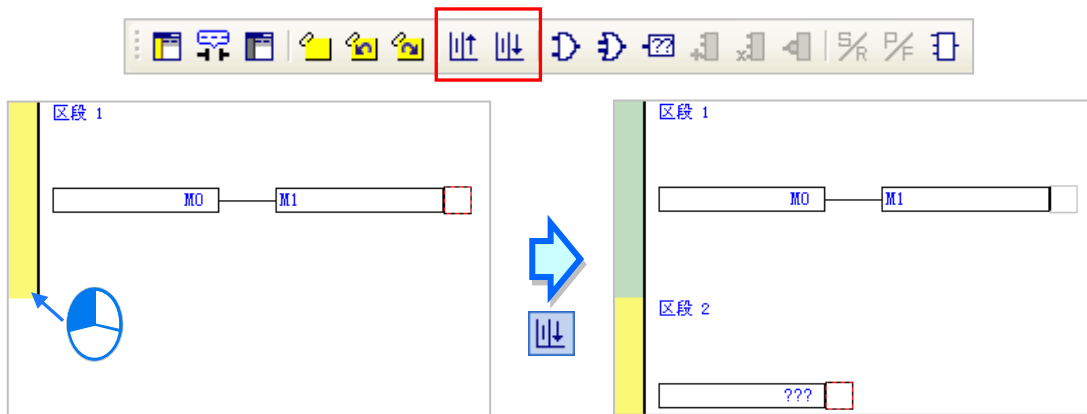



功能块图 (FBD) 程序与梯形图 (LD) 程序一样是以区段 (NETWORK) 为编辑单位，每个功能块图区段都是一个独立的运算程序，而一个功能块图程序亦是由许多的区段所建构而成。在 ISPSOft 中，每个功能块图区段可插入的组件数量并没有特别的限制。

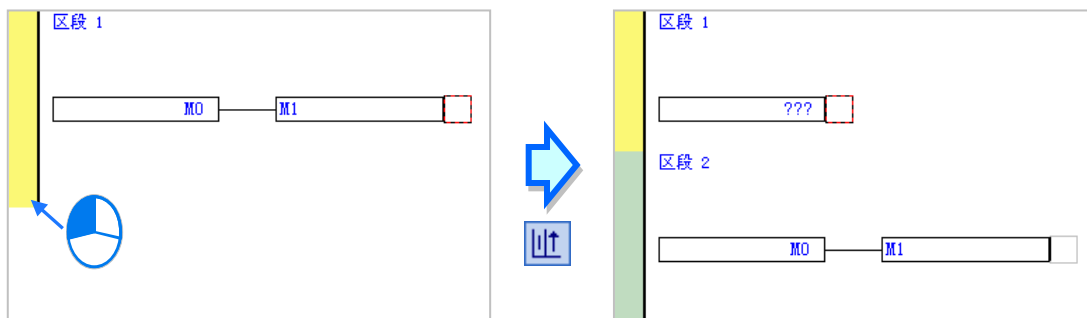
在每个功能块图区段的左上角都会标示该区段的区段编号，且由区段左侧的颜色便可得知该区段目前的状态，其状态的显示依功能可分为**使能与失效**，依编辑状态则可分为**选取与未选取**，而所显示的颜色则可于 ISPSOft 的「选项」设定中由用户自行定义，详细操作请参考第 2.3.1 节。



当新增一个 FBD 程序时，系统会预先建立一个空白的功能块图区段，而要新增区段时，请先选取某个现有区段后，于图示工具栏中点击  图标即可于选取的功能块图区段下方新增一空白区段。

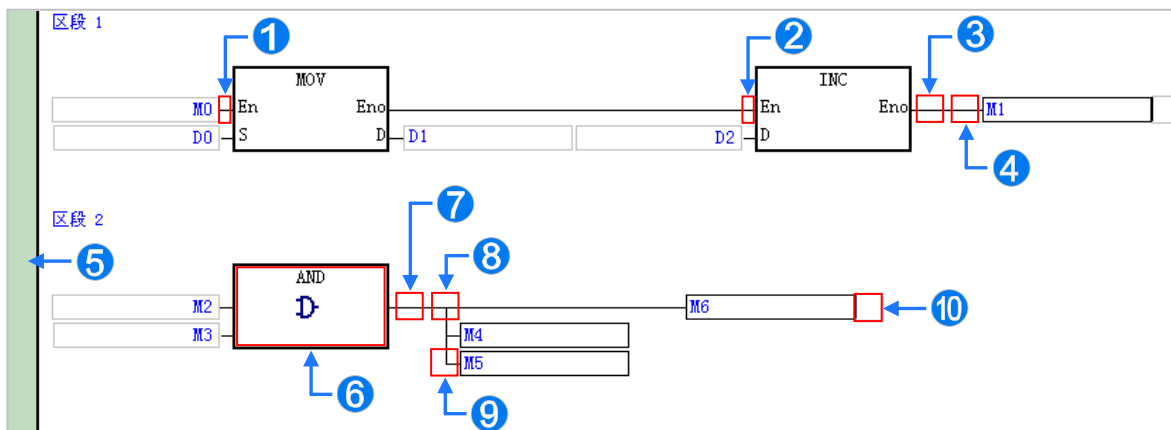


于图示工具栏中点击  图标则可于选取的功能块图区段上方插入一空白区段。



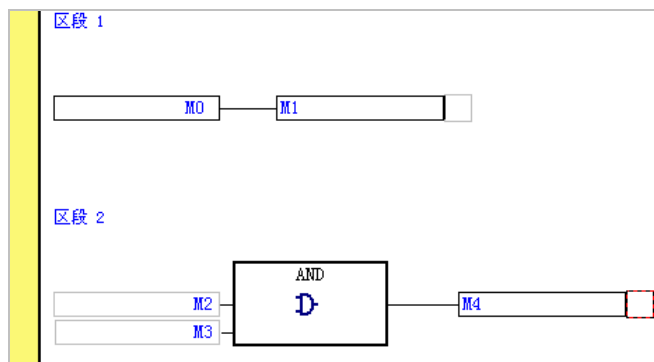
11.1.4 选取对象

在编辑工作区中的红色方框代表目前的编辑位置，也代表目前所选取的对象，而编辑时亦必须将鼠标移至红框的位置点击后才可进行选取；此外，根据选取位置的不同，编辑时可使用的功能也会有所差异，而当进行一般的复制/剪下/贴上/删除的操作时，不同的选取位置，其所包含的操作对象也会有很大的不同；下列即为不同选取位置所代表的意义。



- ① 选取此点左侧的输入节点，如上图的 M0。
- ② 选取此点之前的所有区块。
- ③ 选取此点之前的所有区块。
- ④ 选取右侧的输出节点，如上图的 M1。
- ⑤ 选取整个区段。
- ⑥ 选取方块。
- ⑦ 选取此点之前的所有区块。
- ⑧ 选取此点下方的所有输出节点。
- ⑨ 选取此点右侧的输出节点。
- ⑩ 选取此点之前的所有区块。

当要同时选取多个功能块图区段时，请先按住键盘的【Ctrl】键，接着便可用鼠标分别点选多个不连续位置的区段；另外也可先按住【Shift】键后，再分别用鼠标点选欲选取范围的第一个与最后一个功能块图区段，之后便可将整个范围内的区段全部选取起来。

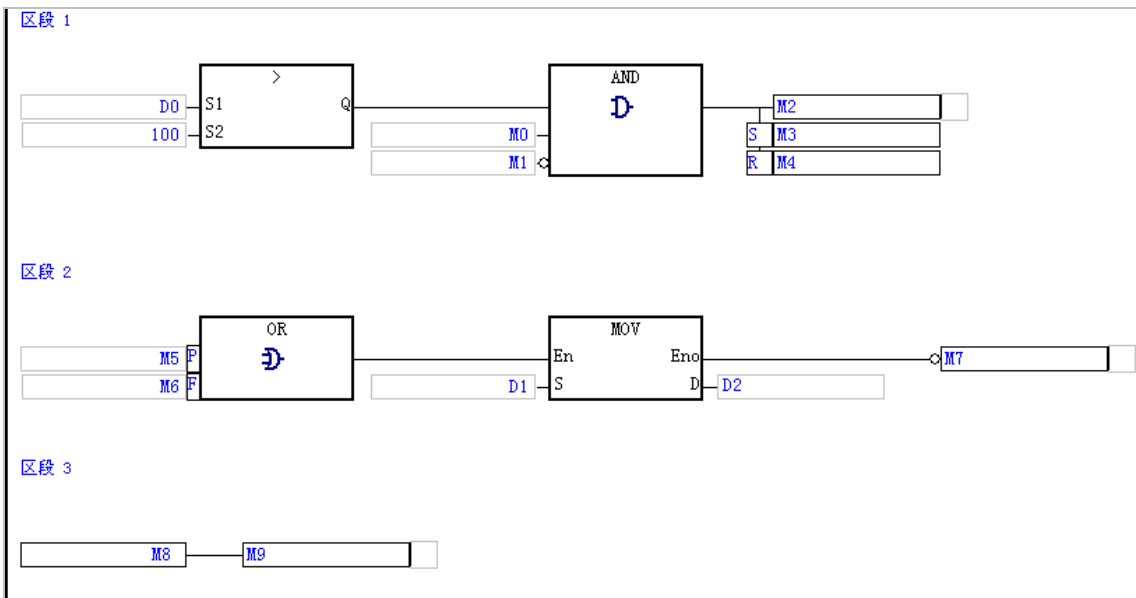


11.2 ISPSOft 中建立 FBD 程序

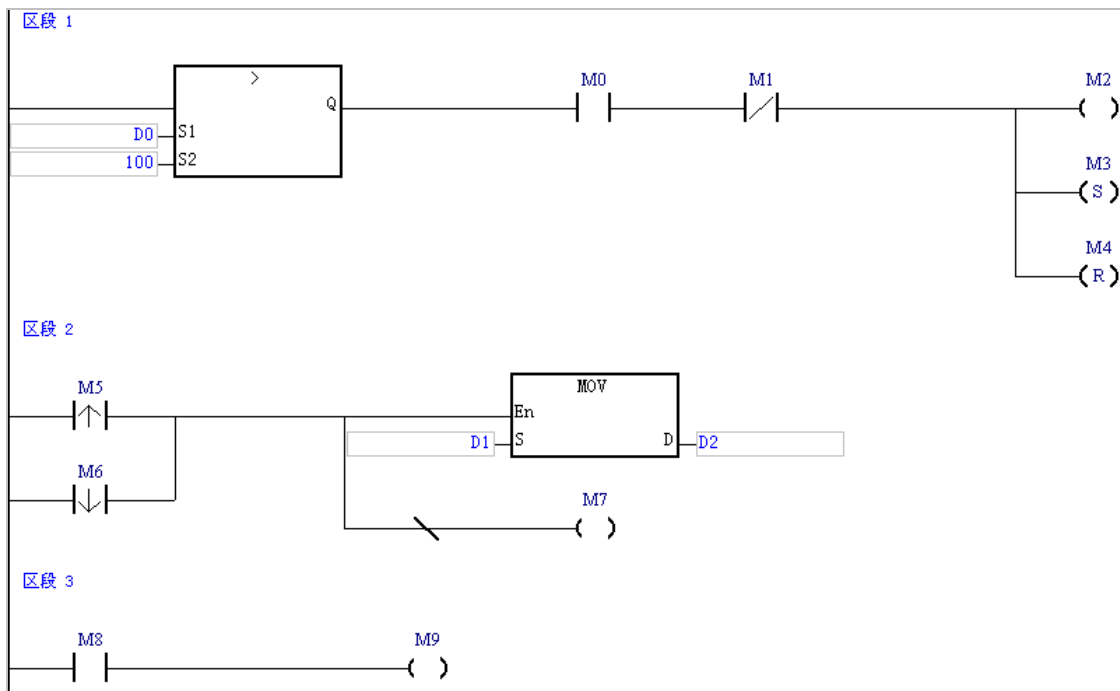


下列为基本的功能块图 (FBD) 程序范例与其等效的梯形图 (LD) 程序，而关于各种 FBD 图示所代表的意义与其建立的方式，我们也将在本节后续的内容当中逐一介绍。

● 功能块图 (FBD) 程序



● 等效梯形图 (LD) 程序



11.2.1 输入节点与输出节点

在 FBD 的图形中，装置与变量符号皆会以节点的图示来表达，而虽然所代表的节点图标并不会因数据类型态的不同而有所区分，但实际运作时，彼此仍必须符合下述原则才可互相传递数据。

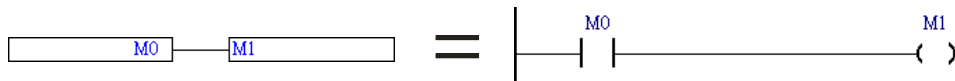
- (a) M、S、T、C、HC 装置与 BOOL、STEP、TIMER、COUNTER 等具备位型态的符号可互传。
- (b) D、L 装置可与所有 BOOL、STEP、REAL、STRING 以外所有型态的符号互传数据。
- (c) T、C 装置可与 D、L 装置及 WORD、INT、TIMER、COUNTER 等型态的符号互传数据。
- (d) D、L、HC 装置与指定 HC 装置的 COUNTER 符号可与 DWORD 与 DINT 型态的符号互传数据。
- (e) WORD 系列与 INT 系列可互传数据，但输出节点的数据长度必须大于或等于输入节点。
- (f) REAL 系列与 STRING 只可与相同型态且长度相同的符号互相传递数据。
- (g) 多重输出节点的结构中，每个输出节点的型态皆须符合上述规定，而每组多重输出至多只可增加至 32 个，但只要往右扩增输出节点之后，便可再建立一组多重输出。

● 图示

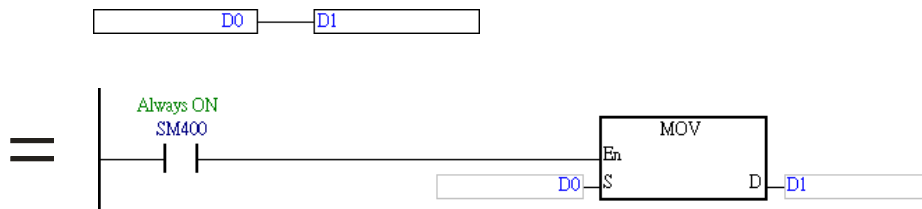


● 范例

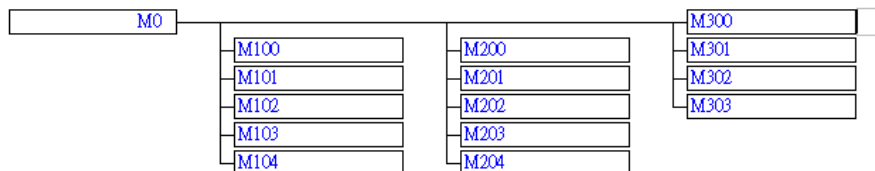
EX1：将 M0 接点的状态输出至 M1。



EX2：将装置 D0 的数据传递给装置 D1。



EX3：以 M100、M200 与 M300 为首，下图总共有三组多重输出，其中每组多重输出至多可再向下增加到 32 个输出节点，而向右则可继续扩增，且数量上并没有特别限制。

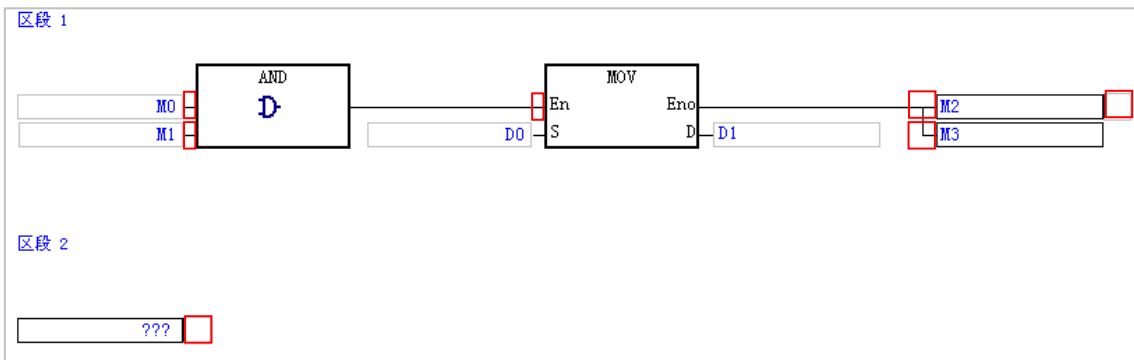


EX4：下表则为几个错误的范例。

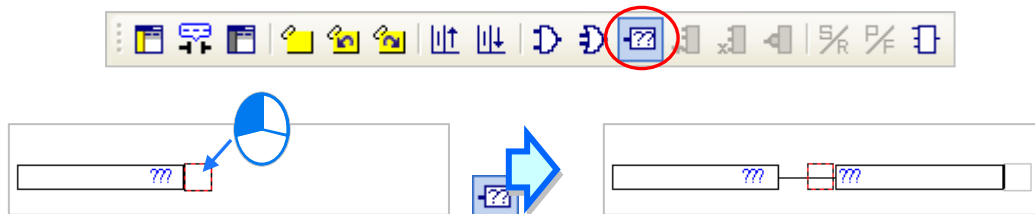
错误范例	说明
	输入节点与输出节点的数据型态不符。
	输出节点的数据长度不得小于输入节点。
	REAL 符号不可直接与 D 装置互传数据。
	REAL 系列的符号必须长度相同才可互传数据。
	STRING 符号必须长度相同才可互传数据。
	TIMER 符号不可输出给 DWORD 符号，但可传给 WORD。
	HC 装置不可输出给 WORD 符号，但可传给 DWORD。
	多重输出的每个输出节点皆需符合规则，而左例的错误则是其中一个输出节点的数据长度小于输入节点。

● 插入输出节点

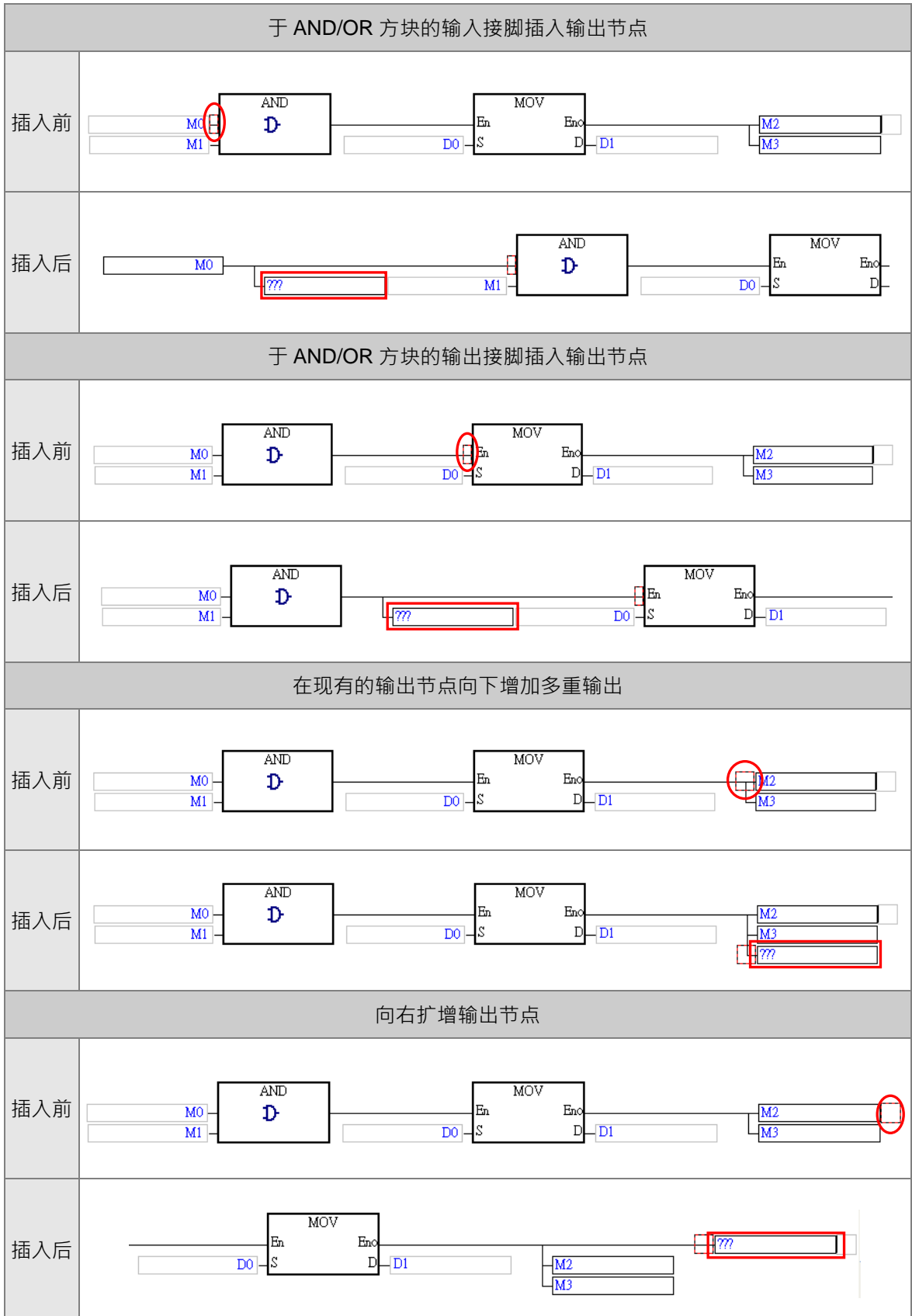
(1) 于编辑工作区中点选欲插入输出节点的位置，而下图红框的位置则都允许插入输出节点。



(2) 除了上图的例子之外，其他的结构也都可能允许插入输出节点，若所点选的位置允许加入输出节点时，图示工具栏中的 图标便会呈现可点选的状态，此时点击该图标便可插入输出节点。



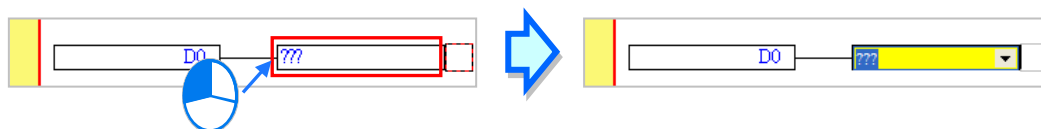
下表中则列举几个输出节点的结构。



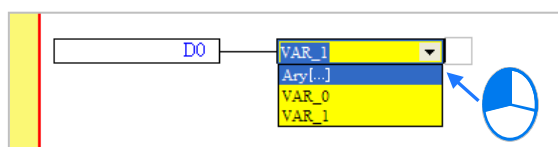
*.功能块或 API 应用指令本身的操作数端点不允许建立多重输出。

11.2.2 FBD 中使用装置、变量符号及常数

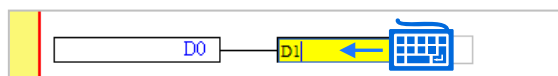
欲编辑节点或指令的操作数时，请直接于该组件的位置上点击鼠标左键，之后该组件便会呈现编辑状态，接着再依据下述方式进行操作即可。



直接点击编辑字段右侧的箭头按钮，接着便可于下拉选单中选择已宣告好的变量符号；而当有宣告数组符号时，于下拉选单中可便可挑选该数组的各个元素，如下图的 Ary[...]。



另外亦可直接于编辑字段以键盘输入设备名称或是变量符号，而当输入未经宣告的变量符号时，输入完毕后按下键盘【Enter】键，系统默认会自动导出符号宣告窗口。(设定方式请参考第 2.3.1 节)



当用鼠标点击某个区段后按下【Enter】键，接着系统便会自动开启编辑字段，而利用键盘的【Enter】键便可在同一个区段中反复移动编辑字段，利用【Tab】键则可跨区段移动，但到了最后一个编辑字段后便会停止；编辑时，使用者可直接于编辑字段中使用键盘输入，而若按下【Page Down】键时，该字段的下拉选单便会随之展开，利用键盘的上下键便可进行选择；完成编辑后，按下【Enter】键便会自动再切换至该区段中的下一个编辑位置，而要结束编辑时，只要按下【Esc】键即可。

补充说明

(a) 要在 ISPSOft 的功能块图 (FBD) 中使用常数时，其格式的范例如下。

- 10 进制：23456 (未加任何标注的数字会被视为 10 进制)
- 16 进制：16#5BA0 (数字前方加上 16#)
- 8 进制：8#55640 (数字前方加上 8#)
- 2 进制：2#101101110100000 (数字前方加上 2#)
- 字符串：“XYZ” (以双引号包住字符串)
- BOOL：须使用常开/常闭接点，可以使用 SM400/SM401(AH/AS 系列機種)或 M1000/M1001 (DVP 系列機種) 来表达。

(b) 使用数组时，其输入格式为 符号名称[索引]，如上图下拉选单中的 Ary[1]。

11.2.3 AND 与 OR 方块

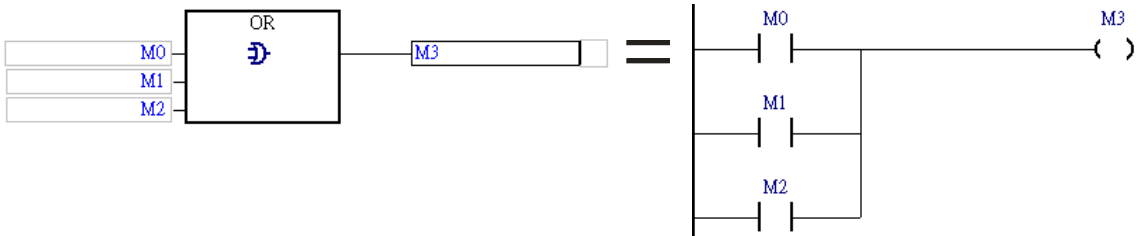
FBD 中的接点串并联皆是透过 AND 与 OR 的方块来表达，而其输入与输出节点则必须为布尔格式，且数量至多可增加至 32 个；此外，AND/OR 的输出至少须有一个输出节点或是再串接至下一组件。

- 图示

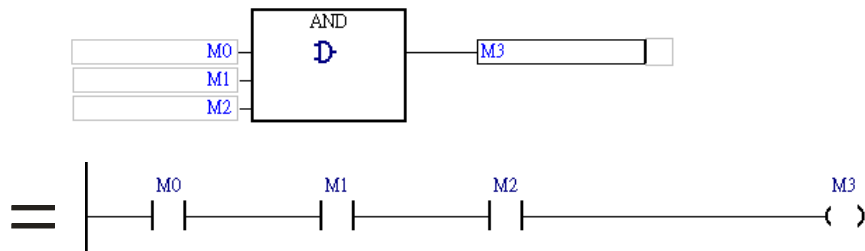


- 范例

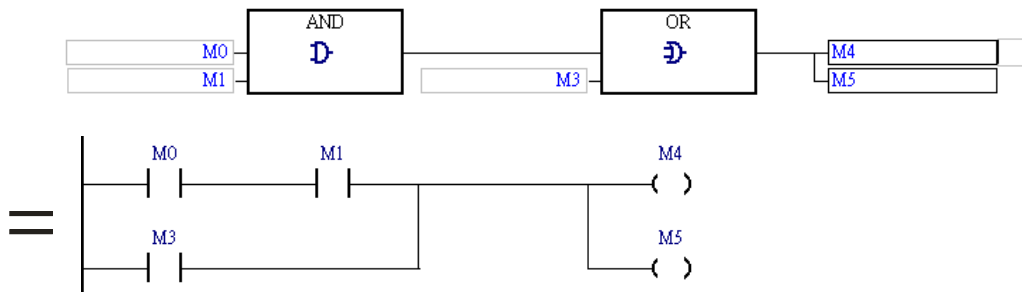
EX1：将 M0、M1、M2 三个接点输入 OR 方块进行运算，并将结果输出给 M3；其运算则相当于将 M0、M1、M2 三个接点并联之后，再将并联后的导通结果拿来驱动 M3 线圈。



EX2：将 M0、M1、M2 三个接点输入 AND 方块进行运算，并将结果输出给 M3；其运算则相当于将 M0、M1、M2 三个接点串联之后，再将串联后的导通结果拿来驱动 M3 线圈。



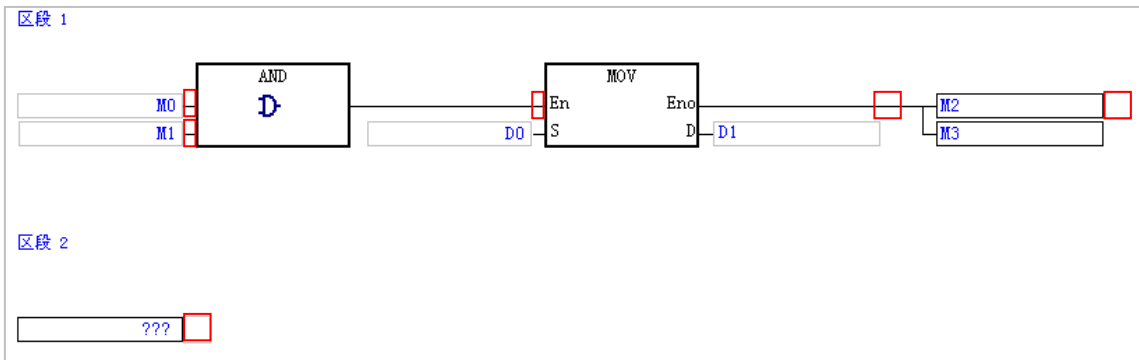
EX3：下图则为混合 AND/OR 方块所建立的逻辑运算，亦相当于梯形图 (LD) 中的串并联结构。





● 插入 AND/OR 方块

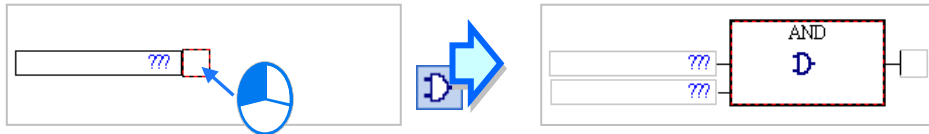
(1) 于编辑工作区中点选欲插入 AND/OR 方块的位置，而下图红框的位置则都允许插入方块。


11

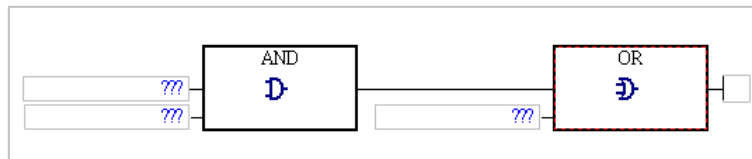
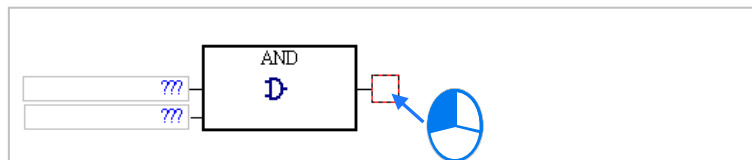


(2) 除了上图的例子之外，其他的结构也都可能允许插入方块，而若所点选的位置允许加入方块时，图示工具栏中的  与  图示便会呈现可点选的状态。


点击  图示时便可将 AND 方块插入所选的位置。

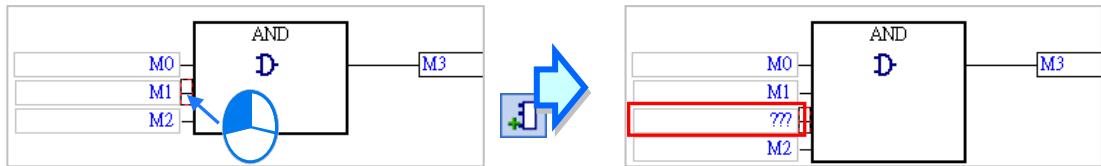


点击  图标则可于所选位置插入 OR 方块。

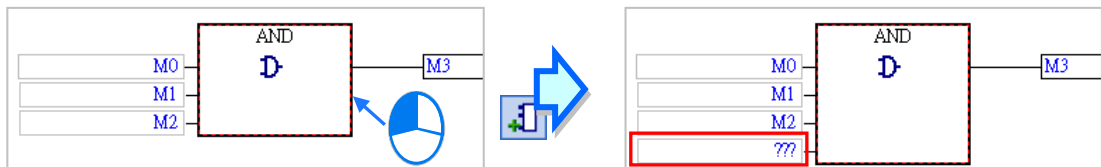


● 增加 AND/OR 方块的输入节点


首先点选某个 AND/OR 方块的输入节点，接着点击图示工具栏的  图示便会于所选节点的下方新增一节点；但须注意若点选的位置是位于最上方的输入节点，如下图的 M0，便无法新增输入节点。



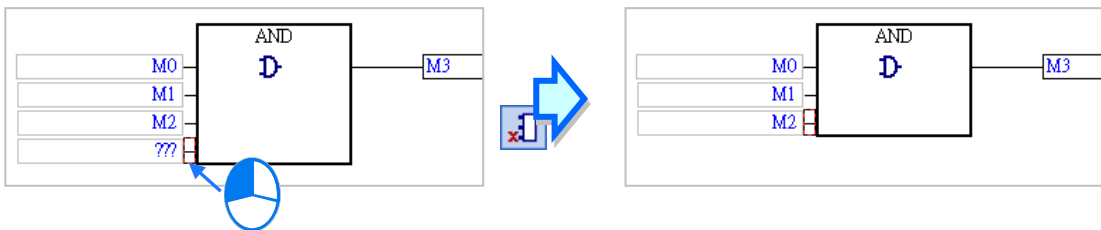
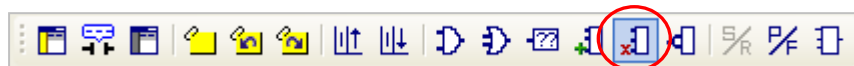
而若一开始选取的位置是在方块上，则输入节点会直接新增于方块的最下方。



● 删除 AND/OR 方块的输入节点

首先于 AND/OR 方块中点选欲删除的输入节点后，点击图示工具栏的  图示便可将其删除，但删除节点时须注意下列两点限制。

- (a) 位于最上方的输入节点，如下图的 M0，不可删除。
- (b) 只剩下两个输入节点时便不允许再删除输入节点。



11.2.4 反相逻辑

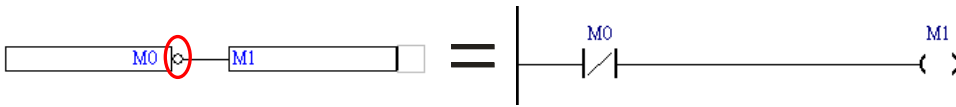
在 FBD 的图形中，对于具有布尔逻辑的位置，如 **BOOL** 型态的输出与输入节点、功能块或 API 应用指令的 **En** 接点，我们便可在该位置上插入反相逻辑的运算；不过须注意，功能块或 API 应用指令本身的操作数端点则不允许加上反相逻辑。

● 图示

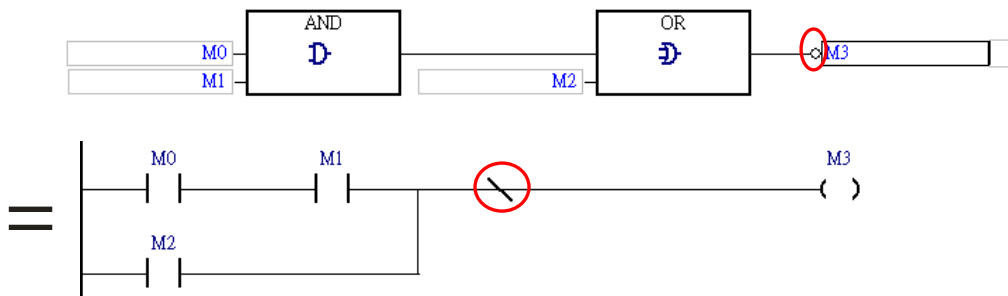


● 范例

EX1：将 M0 的状态反相后输出至 M1，其操作等同让 M0 的常闭接点直接驱动 M1 线圈。



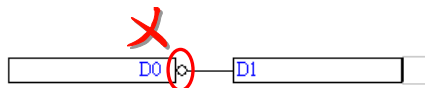
EX2：在输出节点加上反相逻辑便代表将前方区块的逻辑运算结果经过反相后再输出。



EX3：下图是一个二选一的程序执行架构。在同一个功能块图区段中，当 M0 为 ON 时，系统将会执行功能块 **BLACK**；而当 M0 为 OFF 时，系统则会执行功能块 **WHITE**。

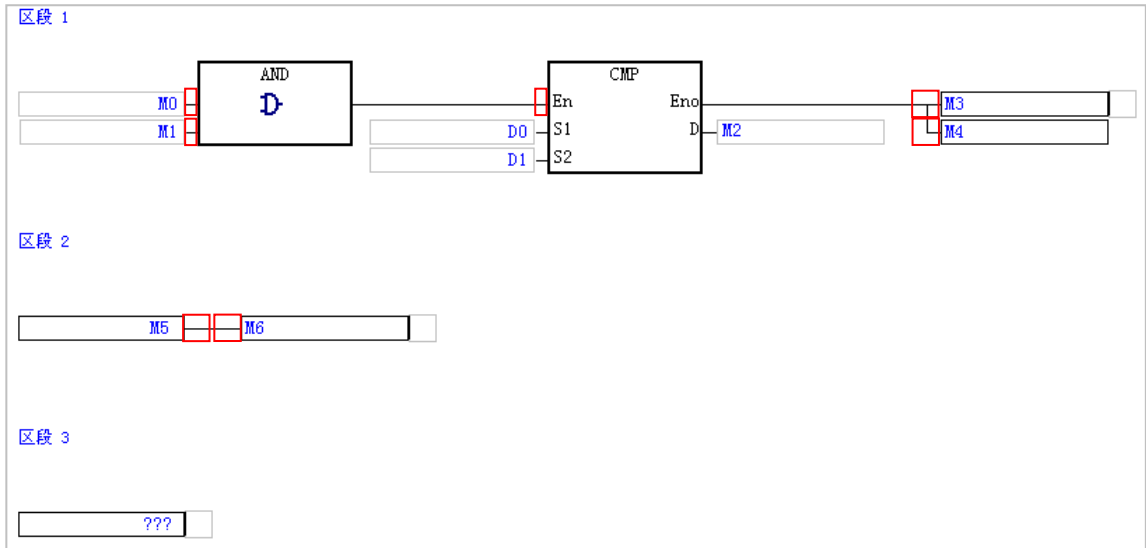



EX4：下图为错误的范例，反相逻辑不得放置于非布尔型态的节点上。




● 插入反相逻辑

- (1) 于编辑工作区中点选欲插入反相逻辑的位置，而下图红框的位置则都允许插入反相逻辑。其中 **CMP** 指令的输出虽然也是 **BOOL** 型态的操作数，不过在 **FBD** 的程序中，功能块或 **API** 应用指令本身的操作数端点并不允许进行附加的状态编辑，如增加多重输出或加上反相逻辑；此外，反相逻辑亦不允许加在没有输出节点或未继续串接其他组件的位置上，如区段 3 的输入节点。





- (2) 除了上图的例子之外，其他的结构也都有可能允许插入反相逻辑，若所点选的位置允许加入反相逻辑时，图标工具栏中的  图示便会呈现可点选的状态，点击该图示便可插入反相逻辑。



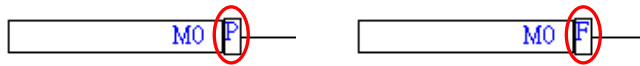
欲取消反相逻辑时，只要于相同位置再次点击图标工具栏中的  图示即可。



11.2.5 上沿触发输入与下沿触发输入

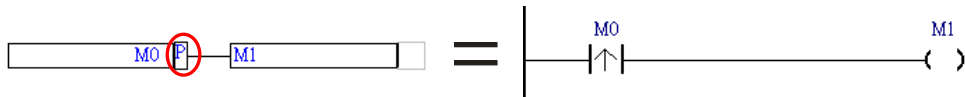
在 ISPSOft 的 FBD 图形中，输入节点上的  符号代表上沿触发，而  符号则代表下沿触发；在进行编辑时，边沿触发只能加在布尔型态的输入节点，或是 AND/OR 方块的输入端点，而功能块或 API 应用指令本身的操作数及 En 端点则不允许加上边沿触发。

● 图示

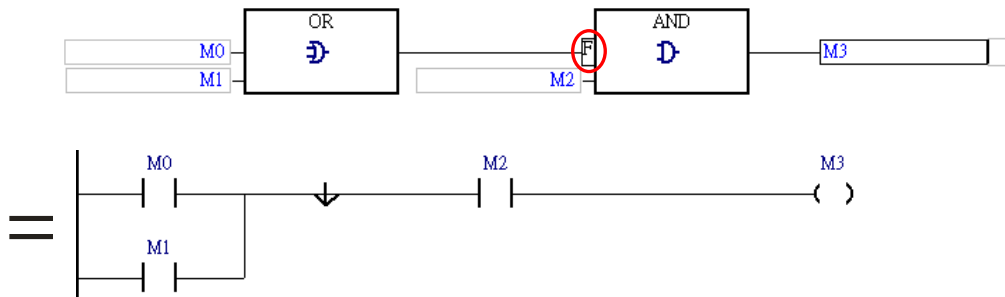


● 范例

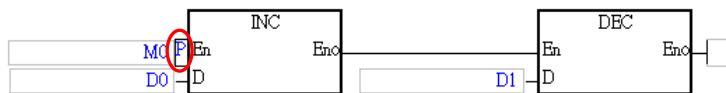
EX1：当接点 M0 的状态由 OFF 变为 ON 的周期，M1 的输出便会为 ON，而在下个周期时，M1 则会再度恢复为 OFF。



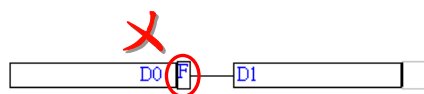
EX2：在 AND/OR 方块的输入端点处，虽然没有明确的输入节点，但仍允许加上边沿触发。



EX3：如下图，第一个 API 应用指令的 En 端点因直接连接到输入节点 M0，因此便可在 M0 的位置加上边沿触发，不过第二个 API 应用指令的 En 端点便不允许加上边沿触发的运算。

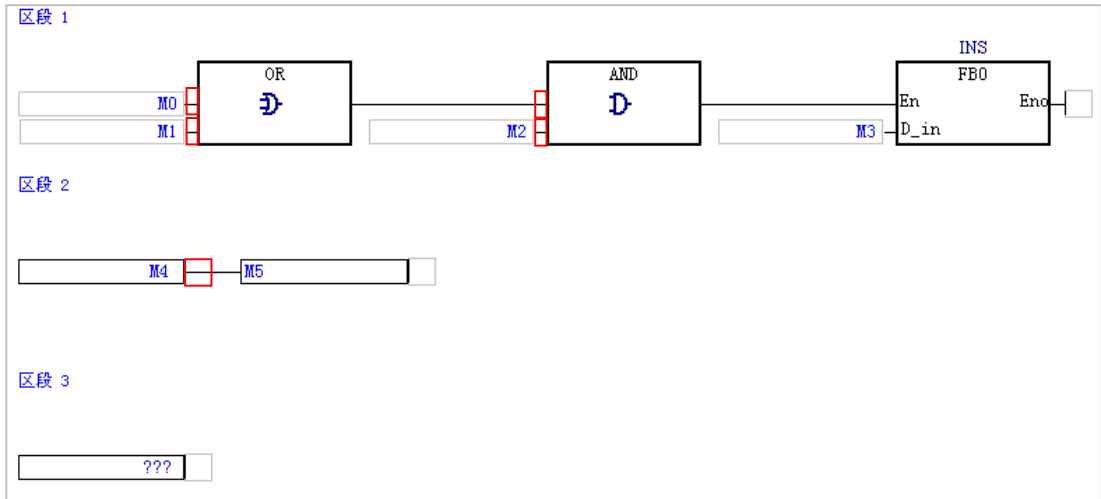



EX4：下图为错误的范例，边沿触发不能加在非布尔型态的输入节点上。

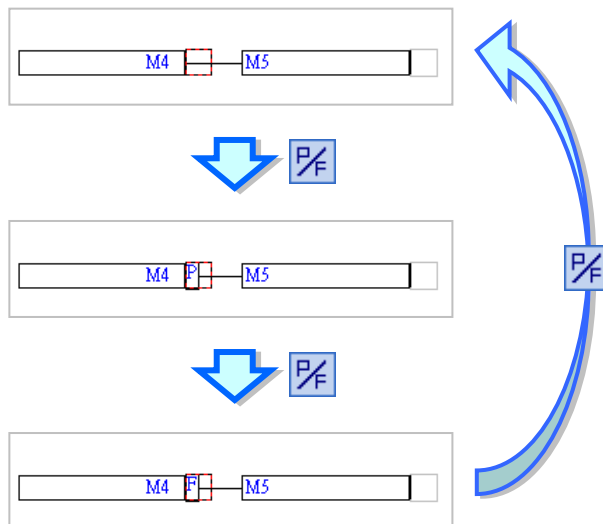


● 插入边沿触发

- (1) 于编辑工作区中点选欲插入边沿触发的位置，而下图红框的位置则都允许插入边沿触发。其中功能块 FBO 的输入端点虽然也是 BOOL 型态的操作数，不过在 FBD 的程序中，功能块或 API 应用指令本身的操作数端点并不允许进行附加的状态编辑，如加上反相逻辑或边沿触发；此外，边沿触发亦不允许加在没有输出节点或未继续串接其他组件的位置上，如区段 3 的输入节点。



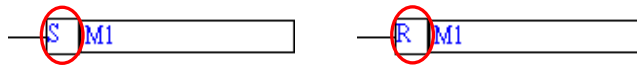
- (2) 除了上图的例子之外，其他结构也可能允许插入边沿触发，若所点选的位置允许加入边沿触发时，工具栏中的  图标便会呈现可点选的状态，反复点击该图标便可切换边沿触发的状态。



11.2.6 置位 (SET) 输出与复位 (RESET) 输出

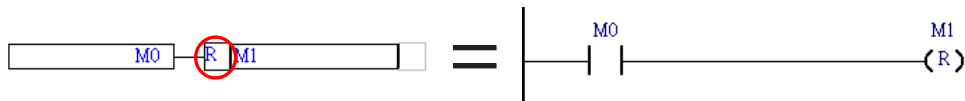
在 ISPSoft 的 FBD 图形中，若输出节点加上 **S** 符号便代表置位 (SET) 输出，而加上 **R** 符号则代表复位 (RESET) 输出；在进行编辑时，置位 (SET) 与复位 (RESET) 只能加在布尔型态的输出节点，而功能块或 API 应用指令本身的操作数则不允许加上置位 (SET) 或复位 (RESET) 的运算。

● 图标

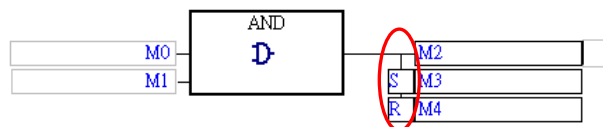


● 范例

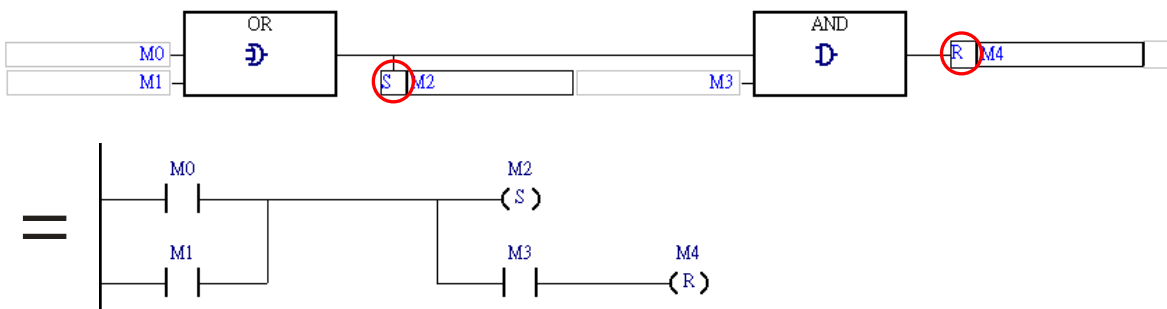
EX1：当 M0 为 ON 时，装置 M1 便会被复位 (RESET) 为 OFF。



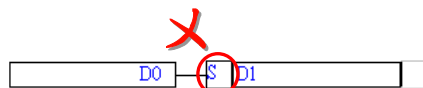
EX2：如下图，在多重输出的结构中，每个输出节点皆可独立编辑其输出型态。



EX3：当 M0 与 M1 任一个接点为 ON 时，M2 便会被置位 (SET) 为 ON；而当 M0 与 M1 任一个接点为 ON 且 M3 也为 ON 时，M4 便会被复位 (RESET) 为 OFF。

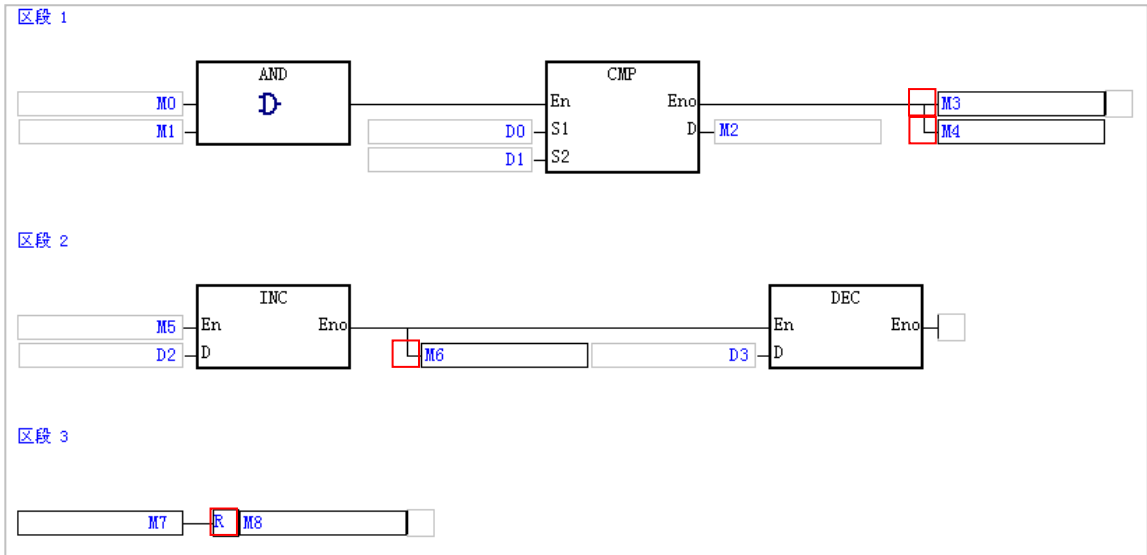



EX4：下图为错误的范例，置位 (SET) 或复位 (RESET) 输出不能加在非布尔型态的输出节点。

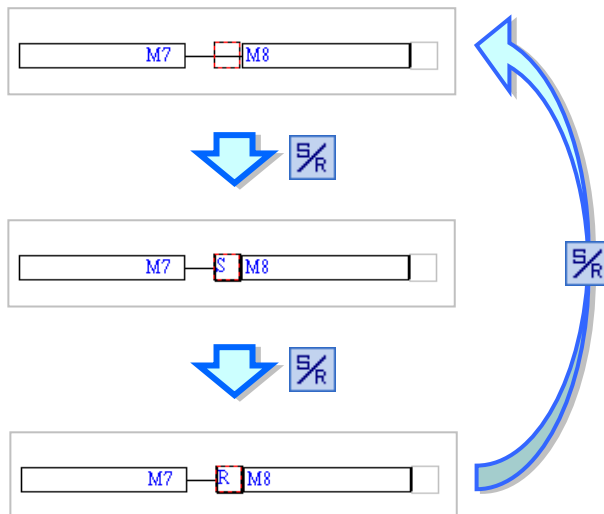


● 插入置位 (SET) 或复位 (RESET) 输出

(1) 于编辑工作区中点选欲插入置位 (SET) 或复位 (RESET) 输出的位置，而下图红框的位置则都允许插入置位 (SET) 或复位 (RESET) 输出。其中 CMP 指令的输出端点虽然也是 BOOL 型态的操作数，不过在 FBD 的程序中，功能块或 API 应用指令本身的操作数端点并不允许进行附加的状态编辑，如加上反相逻辑或置位 (SET) /复位 (RESET) 输出。



(2) 除了上图的例子之外，其他结构也可能允许插入置位 (SET) 或复位 (RESET) 输出，若所点选的位置允许加入置位 (SET) 或复位 (RESET) 输出时，工具栏中的  图示便会呈现可点选的状态，而反复点击该图示便可切换所选位置的输出状态。

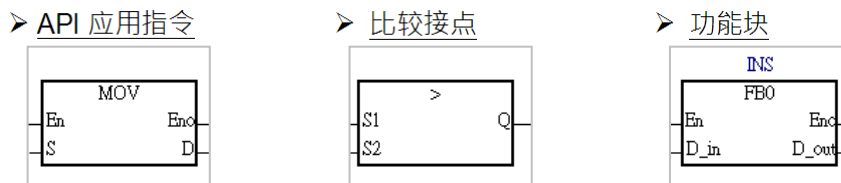


11.2.7 API 应用指令、比较接点与功能块

在 ISPSOft 的 FBD 图形中，API 应用指令、比较接点与功能块的图标与梯形图 (LD) 相同，皆是以方块的图示来呈现，且在功能块图标的上方同样会有一个**功能块实例**的字段。

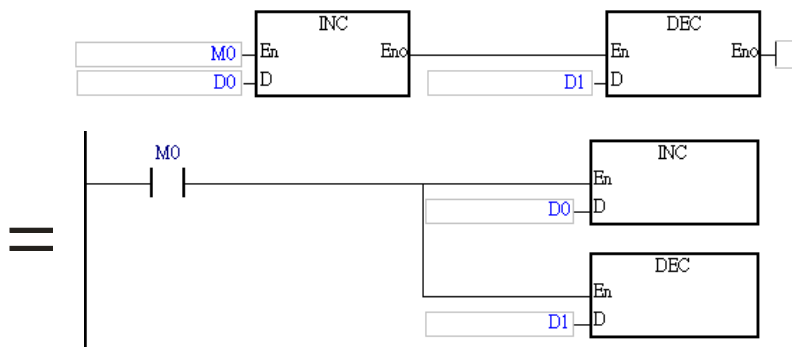
FBD 的 API 应用指令及功能块用法与 LD 相同，当连接至使能端点 En 的逻辑状态为 ON 时，该功能块或 API 应用指令才会被执行；不过基于 FBD 的特性，于同一区段中的功能块或 API 应用指令之间皆可再插入其它的逻辑运算，如反向逻辑或 AND/OR 方块，因此使用起来具有相当高的灵活度。

● 图标

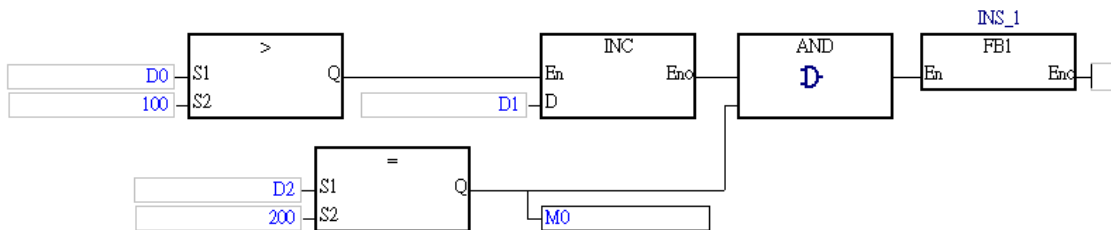


● 范例

EX1：FBD 中的 API 应用指令有包含 Eno 端点，因此在 FBD 的程序中，API 应用指令可于其后方继续串接其他指令或组件。



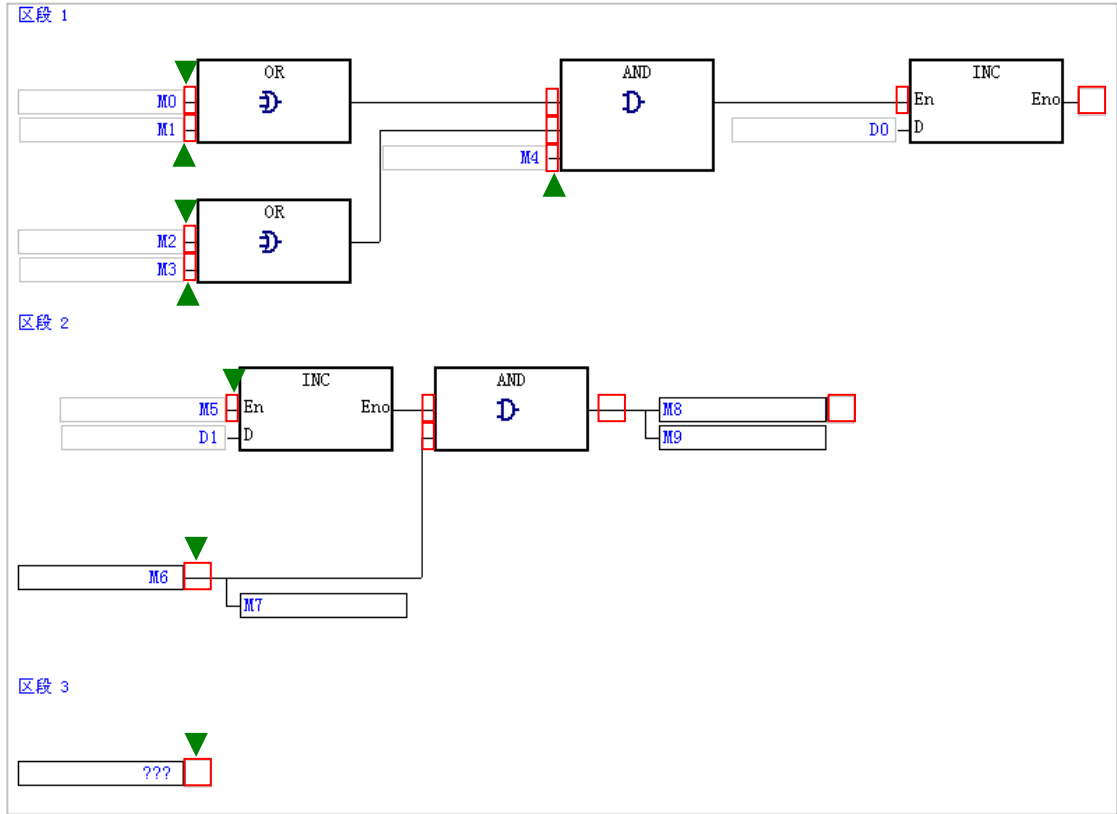
EX2：在 FBD 的程序中，比较接点的前方无法再串接其他组件，而其输出则可串接至 AND/OR 方块的输入端点或直接串接输出节点，亦或是连接至功能块或 API 应用指令的 En 端点。



*.关于功能块的详细介绍请参考第 7 章的内容。

● 插入 API 应用指令、比较接点与功能块

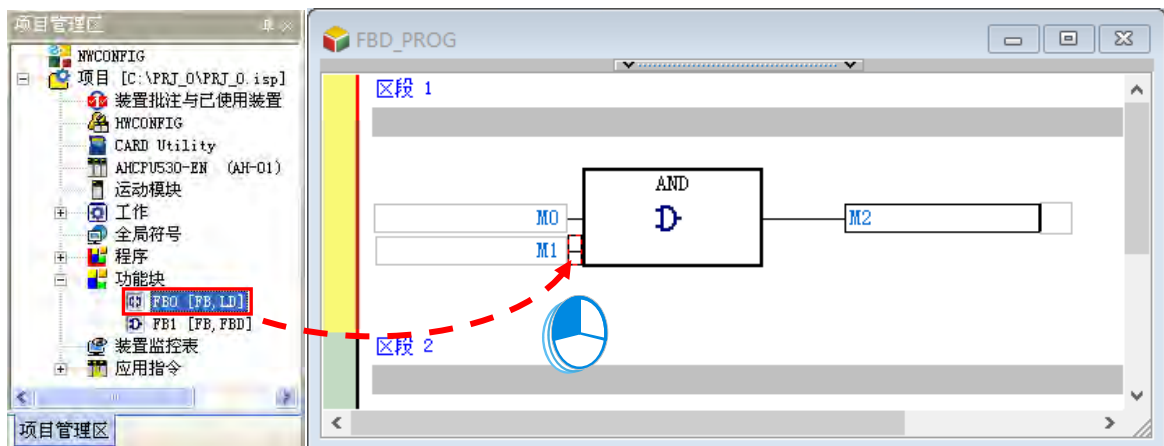
- (1) 首先确认欲插入 API 应用指令、比较接点与功能块的位置，而下图红框的位置则都允许插入 API 应用指令与功能块，但比较接点则只可插在下图中有绿色三角形标志的位置。



- (2) 在 FBD 的编辑环境中，针对插入 API 应用指令、比较接点与功能块的操作提供了多种方式。

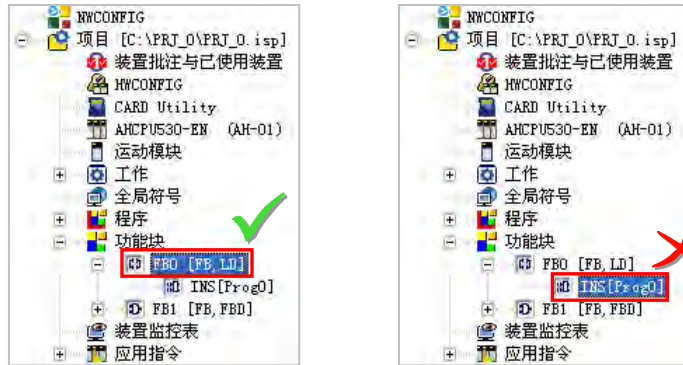
➢ 方法一

于项目管理区中展开应用指令或功能块的目录，在其中找到欲插入的项目后，于该项目上按住鼠标左键，之后再将其拖曳至欲插入的位置后放开左键即可。




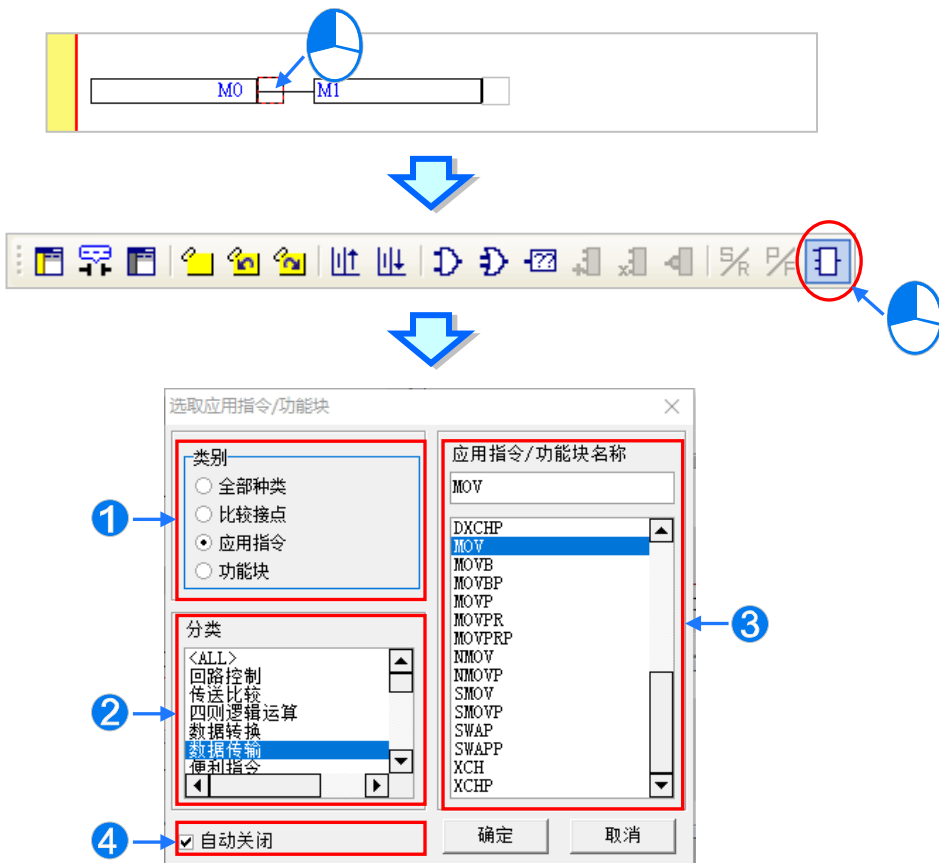
拖曳功能块时须注意只能拖曳功能块的定义，而不能拖曳其下层的功能块实例项目。

11



➤ 方法二

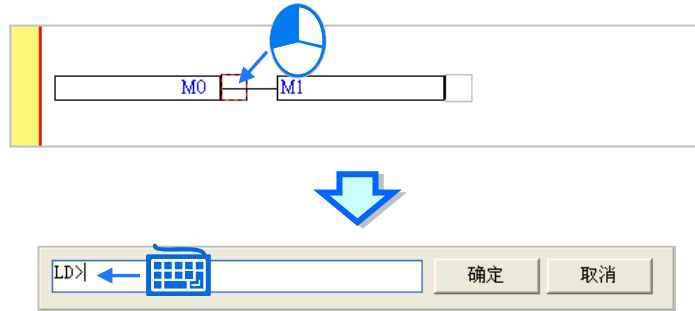
先于欲插入 API 应用指令、比较接点或功能块的位置上点击鼠标左键，之后于工具栏中点选  图标，接着再于出现的选取窗口中点选要加入的项目，完成后按下「确定」键即可。



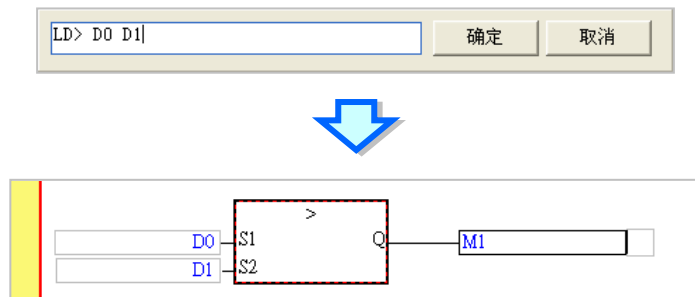
- ① 先选取欲加入的类别。
- ② 若欲加入的是 API 应用指令（函数库）时，请再于此选择指令分类。
- ③ 于此选取 API 应用指令、比较接点或功能块。
- ④ 若取消此处的勾选，之后便可利用此窗口连续加入指令或功能块。

方法三

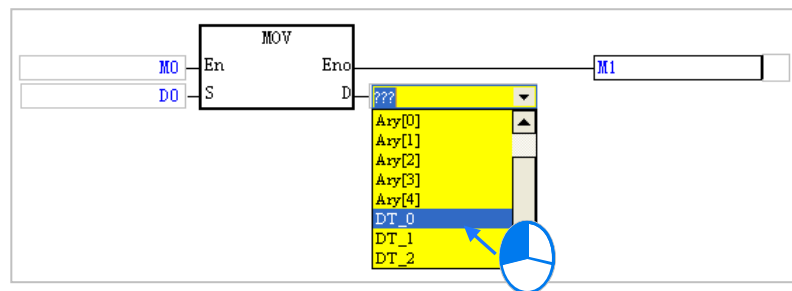
先于欲插入 API 应用指令、比较接点或功能块的位置上点击鼠标左键，之后直接以键盘输入功能块定义、比较接点指令或 API 应用指令的名称（无须分大小写），此时画面便会出现指令编辑模式的输入字段，输入完毕后按下键盘的【Enter】键或以鼠标点击窗口的「确定」按钮即可。



若欲插入的是功能块，则输入时只可输入功能块定义的名称，而不可加上功能块端点的操作数；但若欲插入的是比较接点或 API 应用指令，则输入时可依照该指令的 IL 语法一并输入指令的操作数，且输入的操作数数量无须完整，系统将会依序将其插入对应的操作数字段，并让缺少的操作数字段维持「???'」的显示；若输入的操作数包含未经宣告的变量符号时，确认输入之后，系统亦会自动导出符号宣告窗口。关于 IL 指令的语法请参考各机种的指令手册。



(3) 成功插入 API 应用指令、比较接点或功能块之后，请再逐一完成每个操作数的编辑。

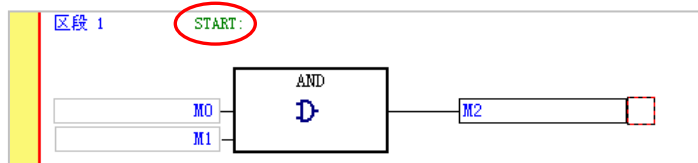


*.自动导出符号宣告窗口的功能可于「选项」当中进行设定，设定方式请参考第 2.3.1 节的说明。

11.2.8 置位标签 (Label)



当要使用跳跃指令，如 CJ、JMP...时，便必须为跳跃的目标区段置位一个卷标 (Label)，而卷标的位置则位于每个区段编号的右侧。若选择的项目机种为 DVP 系列机种或 AH/AS 系列机种时，因其 P 装置是由系统自行配置，因此用户不可直接使用 P 装置，而必须输入一个不重复的卷标符号名称，且卷标符号不需于符号表中宣告。关于 P 装置的用法与限制请参考各机种的使用手册。

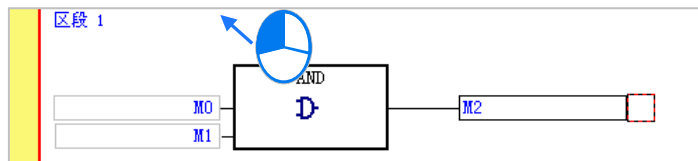


要在指令中使用卷标 (Label) 时，使用卷标符号无须再加上冒号『 :』。

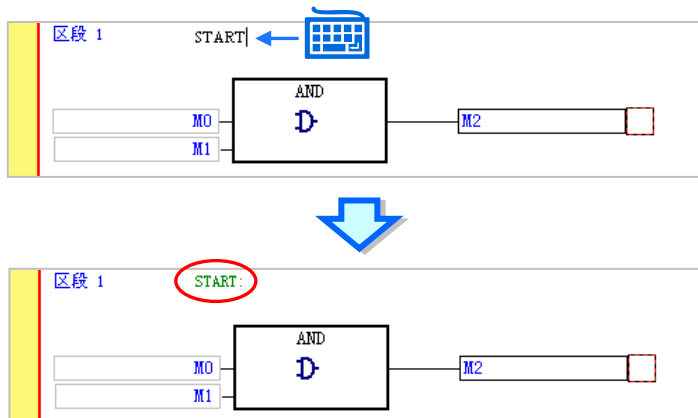


下列则为置位标签 (Label) 的操作方式。

- (1) 在欲置位标签的区段编号右侧点击鼠标左键。

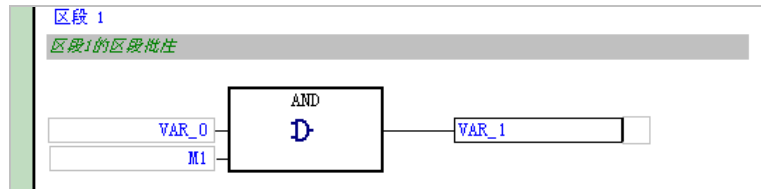


- (2) 若为 DVP 或 AH/AS 机种则请输入卷标符号名称；而输入时，标签后的冒号『 :』可直接省略，系统之后会自行将其补上。完成之后按下【Enter】键即可。




11.2.9 批注与提示

如同梯形图 (LD) 的编辑环境，于 FBD 中同样可替每个功能块图区段加上**区段批注**，而区段批注则位于每个功能块图区段的编号下方。

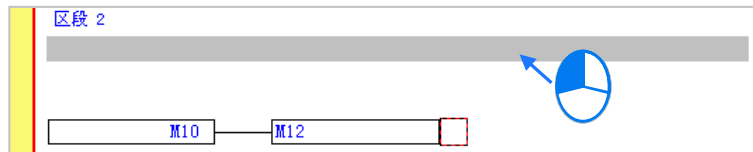


● 编辑区段批注

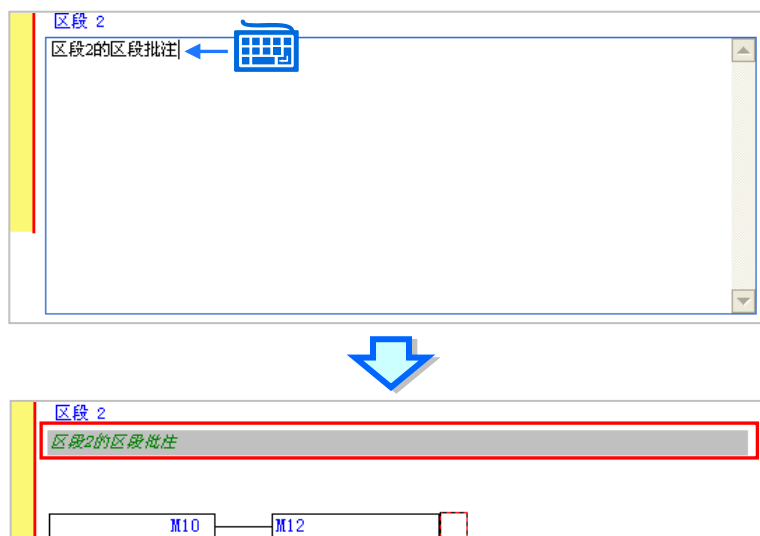
- (1) 当功能块图工具栏中的  图示为按下的状态时，于每个区段编号的下方便会显示该区段的**区段批注**；而同样的，当要隐藏**区段批注**时，请直接将该图示恢复为未按下的状态即可。




- (2) 于**区段批注**的字段上点击鼠标左键，之后便会出现**区段批注**的输入字段。

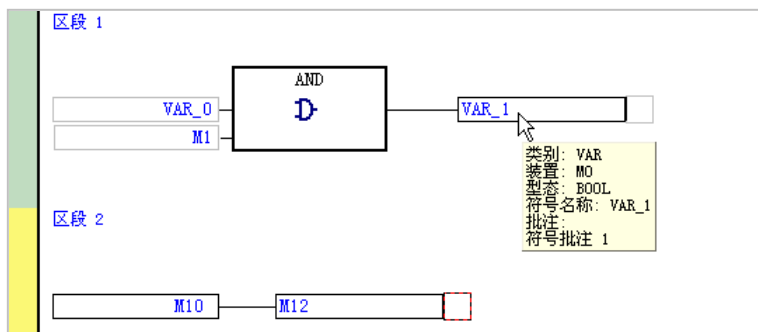


- (3) 接着请直接在出现的编辑字段中编辑批注，其中按下键盘的【Shift】+【Enter】即可换行输入；输入完毕后，直接按下键盘的【Enter】键或用鼠标点击他处即可。

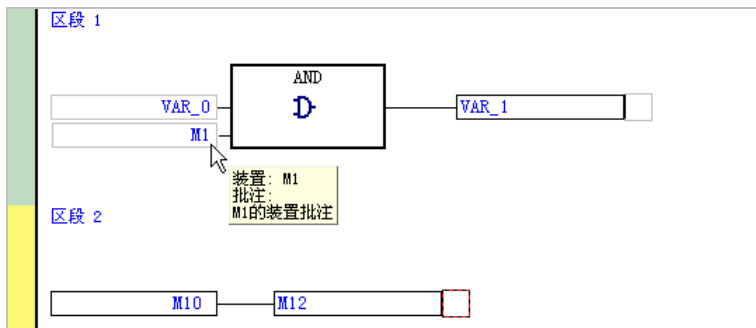


● FBD 的装置与符号提示

在 FBD 的图形中，装置批注与符号批注并不会直接显示于程序画面中，而要检视批注时，请确认图示工具栏中的  图标为按下的状态，之后再将鼠标移至欲检视批注的位置稍作停留，此时画面便会出现该装置或变量符号的相关提示，而提示的内容则可于 ISPSOft 的「选项」功能中进行设定，请参考第 2.3.1 节的相关说明；同样的，当要关闭提示时，请直接将该图示恢复为未按下的状态即可。




除此之外，因装置与变量符号在 ISPSOft 中是属于不同的对象，因此将鼠标指向变量符号与指向装置时所显示的提示与批注内容亦会有所不同。




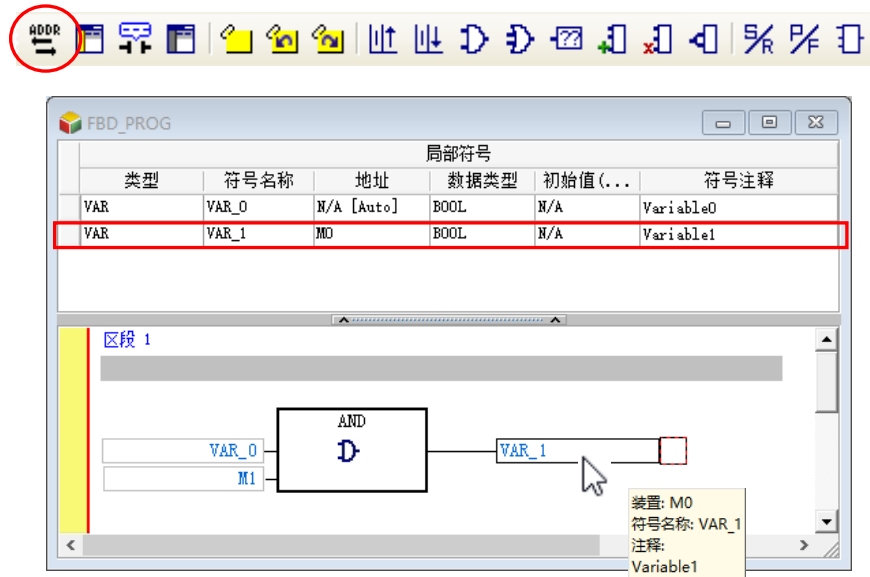
补充说明


当要编辑装置批注时，亦可利用 ISPSOft 所提供的装置批注管理工具；于该工具中，使用者可透过集中管理的方式对每个装置的批注进行编辑，详细的操作方式则将于第 17.2 节当中另行介绍。

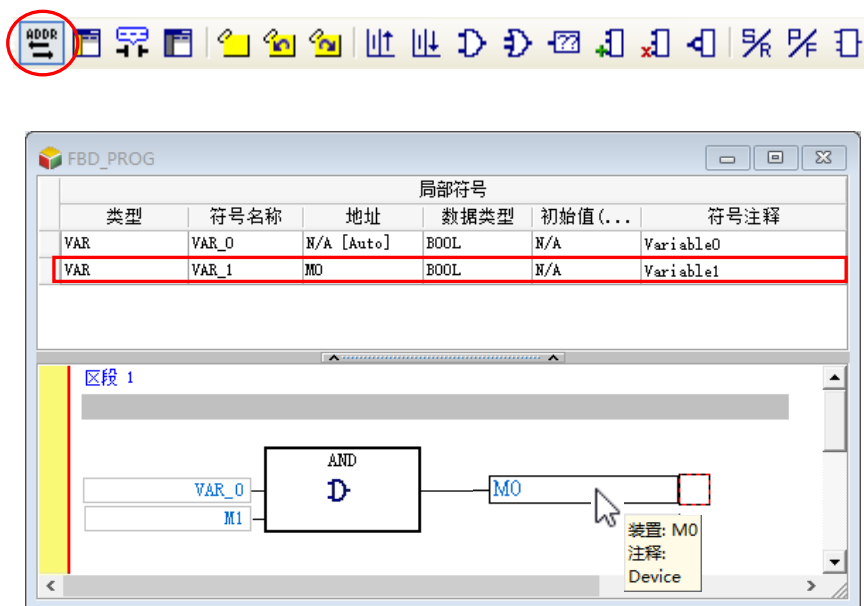
11.2.10 符号模式与地址模式的显示切换

在 LD 与 FBD 的编辑环境中，当某个变量符号在宣告时有一并指定装置地址时，藉由切换图示工具栏的  图标便可变更该变量符号的显示模式；但未指定装置地址的变量符号，或未被变量符号所指定的装置地址则会保持原状。

如下图，当  图标呈现未按下的状态时，代表目前的显示方式为**符号模式**，此时有指定装置地址的变量符号便会以**符号名称**来显示，而提示的内容亦会显示为变量符号的提示。





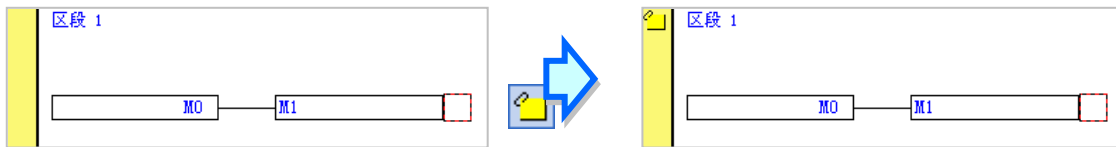
如下图，当  图标被按下之后，便代表目前的显示方式为**地址模式**，此时有指定装置地址的变量符号便会以**装置名称**来显示，而提示的内容亦会显示为装置的提示。





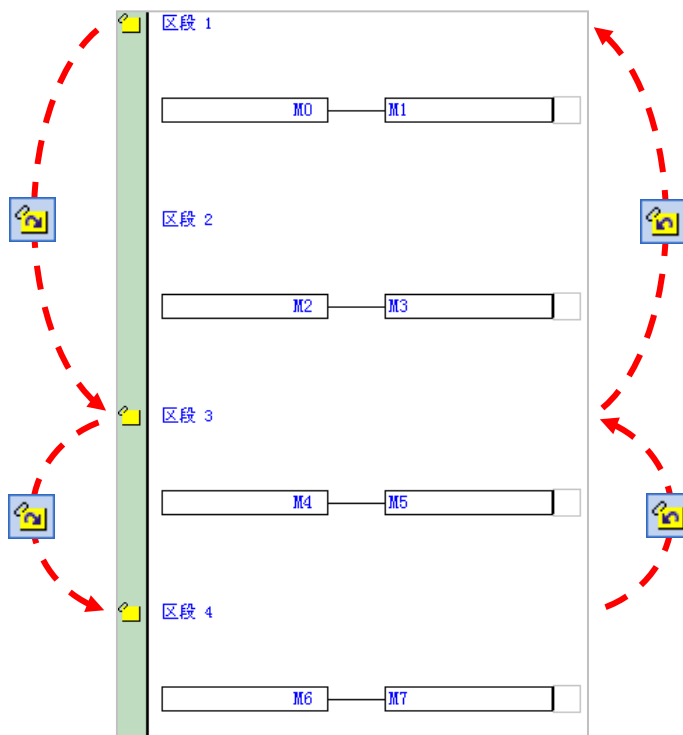
11.2.11 书签功能

书签的功能主要在于协助用户迅速搜寻或移动至加入书签标注的功能块图区段。

- (1) 点击图示工具栏的  图标便可于目前选取的区段中加入一书签；若要取消书签时，只要选取该区段后，再次按下  图标即可。



- (2) 当程序中有插入书签时，按下工具栏图标  即可移动至下一个书签的位置，而按下  图标则可移动至前一个书签的位置。



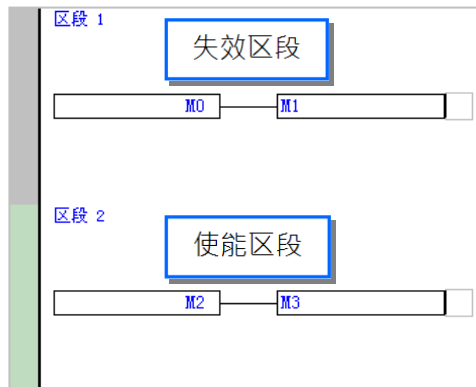
补充说明

于功能工具栏的 **编辑 > 书签** 项目中亦有相同的操作项可供使用，其中点选「**移除所有书签**」项目则可同时将所有的书签移除，不过此操作仅会针对目前正在编辑的窗口产生作用。

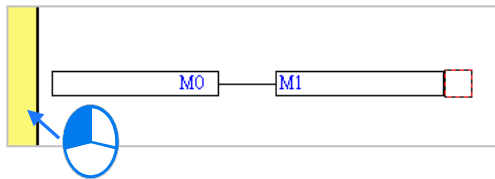
11.2.12 功能块图区段的使能与失效


若某个功能块图区段被设为失效状态，则程序的编译动作便会主动略过该区段，因此当使用者在进行测试或除错的工作时，利用此功能便可暂时关闭程序中的某些程序。

设为**使能**或**失效**的区段，在该区段的左侧会使用不同的颜色来表示，而显示的颜色则可于 ISPSOft 的「选项」功能中进行设定，请参考手册中第 2.3.1 节的相关说明。



要变更区段的状态时，请先将该区段选取起来，而选取时亦可同时选取多个区段。



于功能工具栏中依序点选 **编辑 (E)** > **(使能/失效) 区段**，或是在选取的区段上按下鼠标右键，接着于快捷选单中选取「**(使能/失效) 区段**」，另外也可直接于图示工具栏中点击  图示，之后便可切换被选取区段的使能或失效状态。



当要将所有的区段都恢复为使能状态时，请于功能工具栏中点选 **编辑 (E)** > **失效区段全部取消**，不过此操作仅会针对目前正在编辑的窗口产生作用。

MEMO



第12章 指令列表 IL

目录

12.1	指令列表 (IL) 简介.....	12-2
12.1.1	IL 程序的结构.....	12-2
12.1.2	IL 的功能块 (FB) 调用指令	12-4
12.1.3	IL 编程的注意事项	12-5
12.2	ISPSOFT 中建立 IL 程序	12-6
12.2.1	ISPSOFT 的 IL 编辑环境.....	12-6
12.2.2	编辑 IL 指令.....	12-7
12.2.3	插入 API 应用指令与功能块	12-8
12.2.4	书签功能.....	12-10
12.2.4.1	加入书签	12-10
12.2.4.2	取消书签	12-11
12.2.4.3	移至书签位置.....	12-12

12.1 指令列表 (IL) 简介

指令列表 (**Instruction List, IL**) 是一种类似汇编语言的编程方式，在早期 PC 尚未十分普及的年代，PLC 的编程方式便是利用一种称为**书写器**的装置将控制指令逐一的输入 PLC 当中，而当时所使用的语言便是**指令列表 (IL)**。

12.1.1 IL 程序的结构

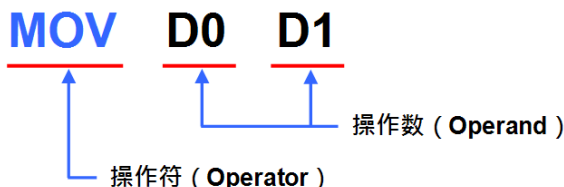
IL 程序是由一行一行的指令叙述建构而成，而其中的每一行指令便代表一个动作，因此要建构一个完整梯形图区段所能达成的任务时，以 IL 而言便可能须要编辑多行的程序指令才可达到。

下列即为一个简单的 IL 程序范例，内容中共有 5 行的程序代码，而右侧的 LD 示意图则用来表达每行 IL 指令对应于梯形图程序所代表的意义。

LD M0：加载 M0 的状态值	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
OR M1：加入 M1 的值做 OR 运算	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
AND M2：再加入 M2 的值做 AND 运算	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
OUT M3：将目前的运算结果输出至 M3	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
(*Note*)：此行为程序批注，因此编译时会自动略过。	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	<div style="border: 1px solid black; padding: 10px; text-align: center;">无动作</div>

IL 的指令结构可拆解为**操作符 (Operator)** 与**操作数 (Operand)** 的组合，其中操作符是代表所要执行的动作，而操作数则是参与执行动作的成员或对象，且其型态依指令格式可以是装置、变量符号或常数；此外，在 IL 语法中，各个操作数之间必须使用空白来区隔。

如下列范例，其中 **MOV** 代表要执行的是数据搬移的动作（操作符），而 **D0** 与 **D1** 则分别是参与搬移动作的来源与目的装置（操作数）。

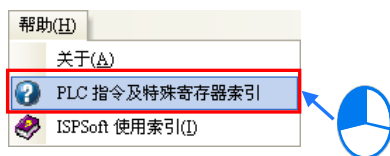


在 IL 语法中，被框在 (* 与 *) 符号之间的文字都会被视作批注，而其内部不论包含任何指令或关键词，当进行编译时，系统都会自动略过这段内容；此外，只要在不破坏指令格式的前提下，批注可被加在程序中的任何位置，但须留意程序的可读性。如下图，在下列范例中的所有批注都是合法的。

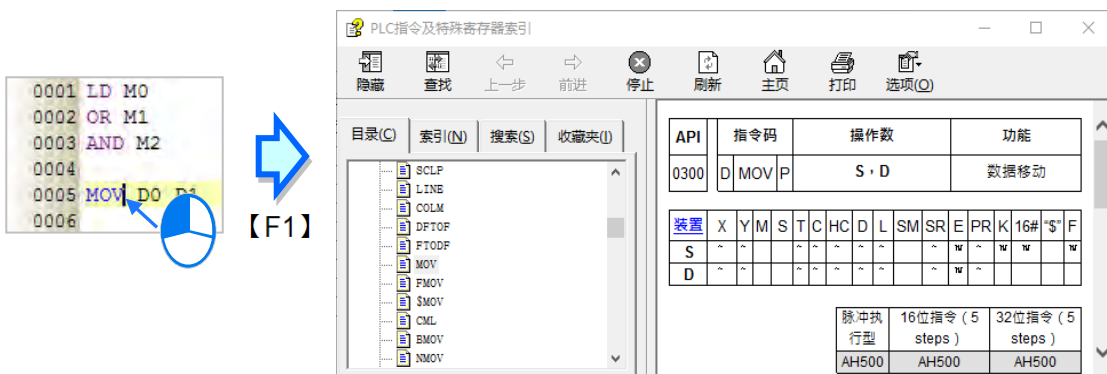
```

0001 LD M0 (*Note*)
0002 (*Note*) OR M1
0003 AND (*Note*) M2
0004 OUT M3
0005 (*Note 1
0006 Note 2
0007 Note 3*)
    
```

IL 的指令可分为基本指令与 API 应用指令，相关指令的格式与用法在各机种的指令手册中已有详尽的说明，本章将不再多加叙述；而在编辑过程中要确认指令的用法时，可于 ISPSOft 的功能工具栏中点选 **帮助 (H) > PLC 指令及特殊寄存器索引**，之后系统便会自动开启说明文档。



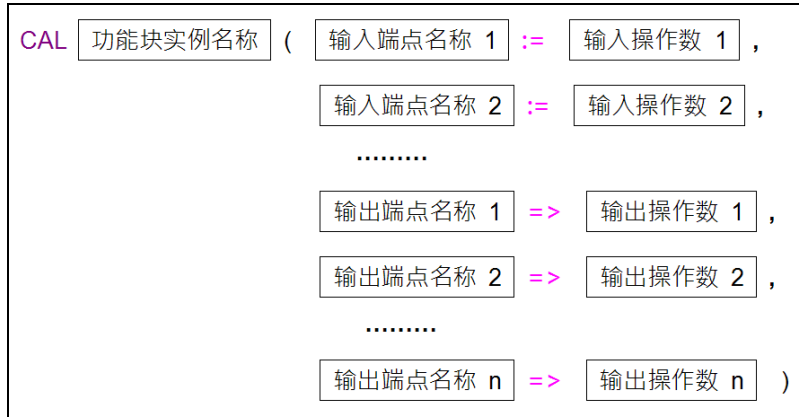
此外，若点选某个指令后再按下键盘的【F1】键，则系统便会在开启文档后直显示该指令的页面。



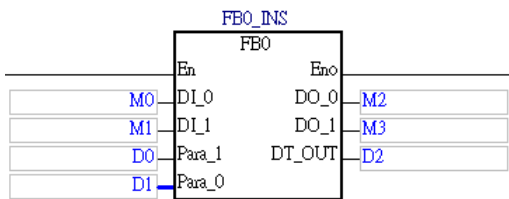
12.1.2 IL 的功能块 (FB) 调用指令

下列则为在 IL 程序中调用功能块的指令格式与注意事项。

12



- 除了 **CAL 指令**、**功能块实例名称**与**左括号**须位于同一行之外，其余部分皆可任意换行。
- 功能块的每个输入输出端点与其对应的操作数都必须被包含在括号之中，且每组端点与操作数之间须以逗号隔开，而可用的操作数型態则必须依据各功能块的定义来加以指定。
- **CAL 指令**之后的名称应为**功能块实例**而非**功能块定义**，亦即应输入被宣告成**功能块**型态的变量符号名称。关于**功能块定义**与**功能块实例**的详细说明请参考**第 7 章**的相关内容。
- 输入侧端点所使用的指定符号为「:=」，而输出侧端点所使用的指定符号则是「=>」。
- 结构内的端点与操作数顺序必须依照各功能块的定义，而不可任意变动，其顺序则为排除 **En/Eno** 端点后，先由上而下列出所有输入侧的端点，其中 **VAR_IN_OUT** 类型的端点则被归类为输入侧；接着再从上而下列出所有输出侧的端点。请参考下列功能块的梯形图图标与 **IL 指令**。



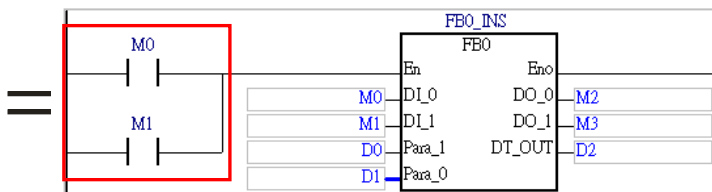
```

0003 CAL FBO_INS ( DI_0 := M0 ,
0004                DI_1 := M1 ,
0005                Para_1 := D0 ,
0006                Para_0 := D1 ,
0007                DO_0 => M2 ,
0008                DO_1 => M3 ,
0009                DT_OUT => D2 )
    
```

- 在功能块指令结构的内部并没有任何 **En** 端点的指定动作，因此当执行至该指令时，系统便会以执行至该指令之前的逻辑运算状态做为功能块使能与否的判断条件。

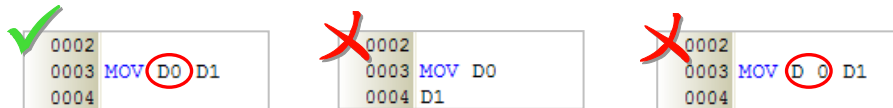
```

0001 LD M0
0002 OR M1
0003
0004 CAL FBO_INS ( DI_0 := M0 ,
0005                DI_1 := M1 ,
0006                Para_1 := D0 ,
0007                Para_0 := D1 ,
0008                DO_0 => M2 ,
0009                DO_1 => M3 ,
0010                DT_OUT => D2 )
0011
    
```

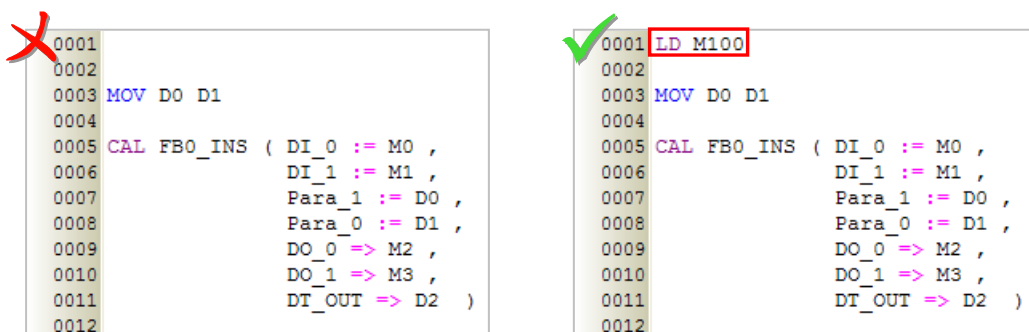


12.1.3 IL 编程的注意事项

- 除功能块指令之外，在 IL 程序中的每行文字都会被视为一个指令，因此操作数与操作符之间不可有换行，否则便会被视为两个不完整的指令。此外，于指令名称、功能块名称或操作数名称的字符串之间不可插入空白，否则进行编译时将无法识别。



- 在不破坏指令格式与完整性的前提下，任何的空白或换行都是被允许的。
- 全角与半角文字在 ISPSOft 中会被视为不同的字符，因此请尽量避免使用全角文字以免造成识别及解读上的误判。
- 在 IL 的语法中，英文无大小写的分别，亦即 OUT、Out 与 out 的意义是相同的。
- 要在 IL 程序当中使用常数时，其表达的格式如下。
 - 10 进制：23456（未加任何标注的数字会被视为 10 进制）
 - 16 进制：16#5BA0（数字前方加上 16#）
 - 8 进制：8#55640（数字前方加上 8#）
 - 2 进制：2#101101110100000（数字前方加上 2#）
 - 字符串：“XYZ”（以双引号包住字符串）
 - BOOL：须使用常开/常闭接点 SM400/SM401（AH/AS）或 M1000/M1001（DVP）来表达。
- 使用数组时，其输入格式为 符号名称[索引]，且因 IL 尚不支持变量型态的索引值，因此索引值只可使用小于数组大小的 10 进制常数，且必须由 0 开始。
- 当使用功能块或 API 应用指令时，若在下达指令之前没有任何 LD 指令或逻辑运算的区块，则执行编译时便会产生错误。

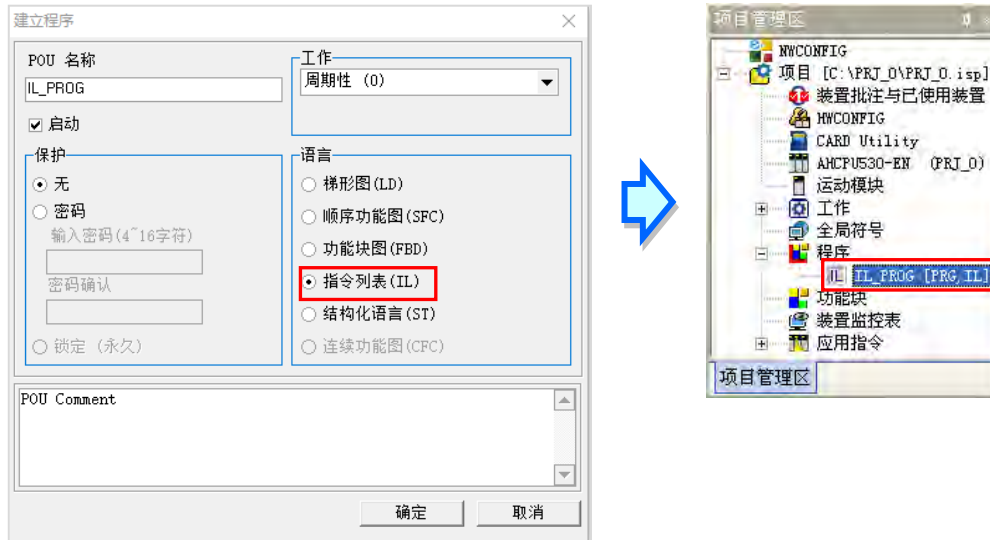


- IL 语言可编辑的行数并没有限制，但仍需考虑编译后的程序大小是否会超出主机的内存容量。
- ISPSOft 的 IL 编辑环境支持复制/剪切/粘贴的功能，因此用户可从其它的文字编辑器中将编辑好的内容复制到 ISPSOft 的 IL 编辑工作区中。

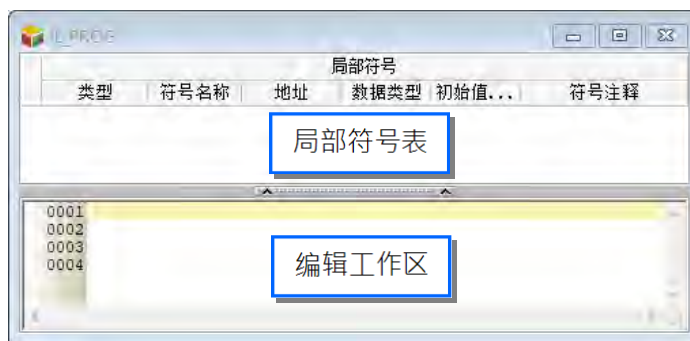
12.2 ISPSOft 中建立 IL 程序

12.2.1 ISPSOft 的 IL 编辑环境

在建立 POU 时，于语言字段选择「指令列表 (IL)」便可建立 IL 程序，请参考第 5.4.1 节的说明。



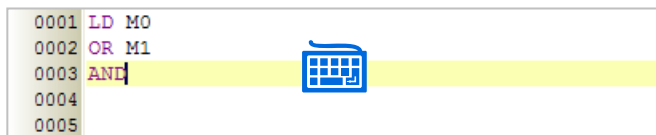
下图即为 ISPSOft 中的 IL 编辑环境，其窗口配置的上方为局部符号表，而下方则是编辑工作区，且在编辑工作区的左侧会有行号显示，用户可自行设置其显示的格式，详细操作请参考第 2.3.1 节的相关说明。在进入 IL 的编辑环境后，窗口中亦会出现对应的图标工具栏，其功能如下。



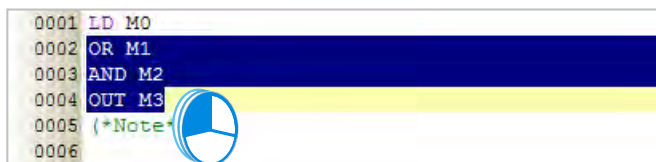
图标	键盘 (快捷键)	功能说明
	Shift + Ctrl + B	在目前选取的行列上加入或取消书签标记
	Shift + Ctrl + P	移至前一个书签的标记位置
	Shift + Ctrl + N	移至下一个书签的标记位置
	Shift + Ctrl + U	插入指令或功能块

12.2.2 编辑 IL 指令

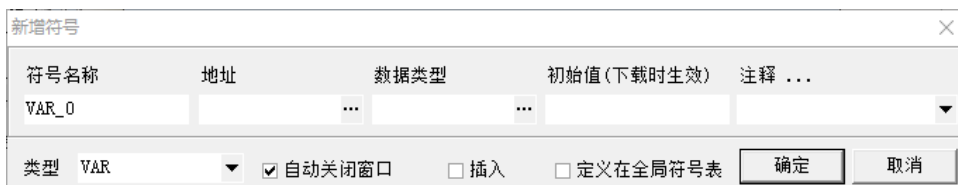
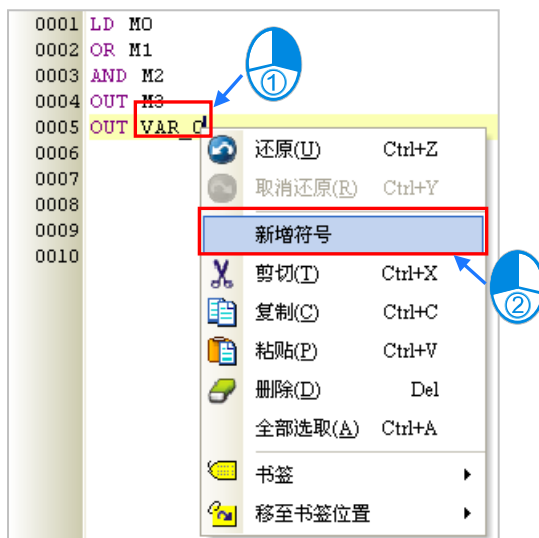
在 IL 的编辑工作区中，其操作方式与一般的文字编辑器类似，编辑时只要依据指令格式直接使用键盘在工作区上面输入或修改文字即可，且当按下键盘的【Enter】键时便可执行换行的动作。



选取文字的方法也与一般的文字编辑器相同。先将鼠标点在选取区块的起点并按住鼠标左键，之后往选取的方向移动便可框选文字；又或者是先用鼠标点选在某个区块的起点，接着按住键盘的【Shift】键，之后再用鼠标点选该区块的终点便可将整段区块选取起来。



当在编辑的过程中输入了未经宣告的变量符号时，输入完毕后用鼠标点击该文字并按下右键，之后再于快捷选单中点选「新增符号」便可自动导出变量符号的宣告窗口。



12.2.3 插入 API 应用指令与功能块

插入 API 应用指令与功能块的方法有下述三种。

● 方法一

依据应用指令与功能块的格式，直接于编辑工作区中利用键盘输入即可。

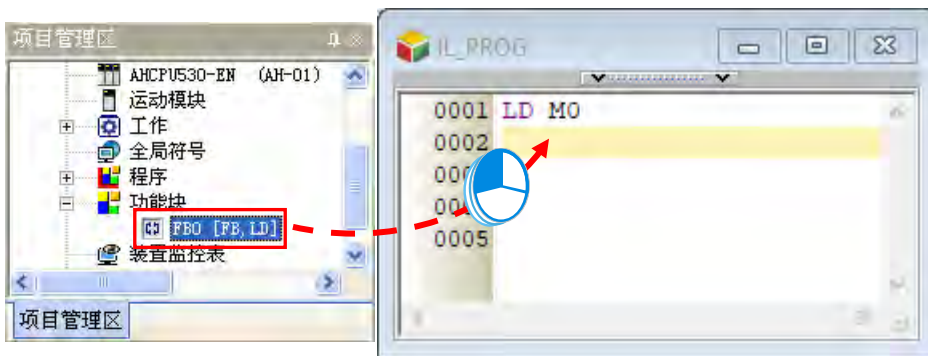
12

```

0001 LD MO
0002 CAL FBO_INS (
0003     TRIG := MO ,
0004     DT_IN := DO ,
0005     DT_OUT => D1
0006 )
0007
0008
    
```

● 方法二

于项目管理区中展开应用指令或功能块的目录，在其中找到欲插入的对象后，于该对象上按住鼠标左键，之后再将其拖曳至欲插入的位置后放开左键即可。




插入应用指令或功能块之后，请再依据其定义的格式，直接于插入的样板中编辑对应的操作数。

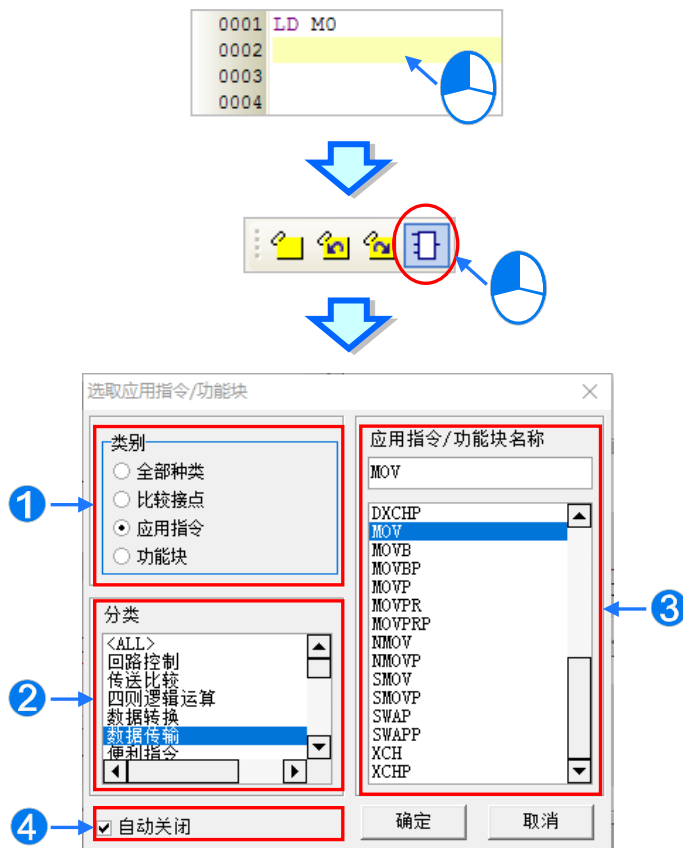
<pre> 0001 LD MO 0002 CAL ????? (0003 TRIG := ?????, 0004 DT_IN := ?????, 0005 DT_OUT => ????? 0006) 0007 </pre>		<pre> 0001 LD MO 0002 CAL FBO_INS (0003 TRIG := MO , 0004 DT_IN := DO , 0005 DT_OUT => D1 0006) 0007 </pre>
---	--	--

除此之外，拖曳功能块时须注意只能拖曳功能块的定义，而不能拖曳其下层的功能块实例。



● 方法三

先于欲插入 API 应用指令或功能块的位置上点击鼠标左键，之后于图标工具栏中点选  图标，接着再于出现的选取窗口中点选要加入的对象，完成后按下「确定」键即可将其插入指定位置。



- ① 先选取欲加入的类别。
- ② 若欲加入的是 API 应用指令（函数库）时，请再于此处选择指令分类。
- ③ 于此选取 API 应用指令、比较接点或功能块。
- ④ 若取消此处的勾选，之后便可利用此窗口连续加入指令或功能块。

接着请再依据应用指令与功能块的格式，直接于插入的样板中编辑对应的操作数。



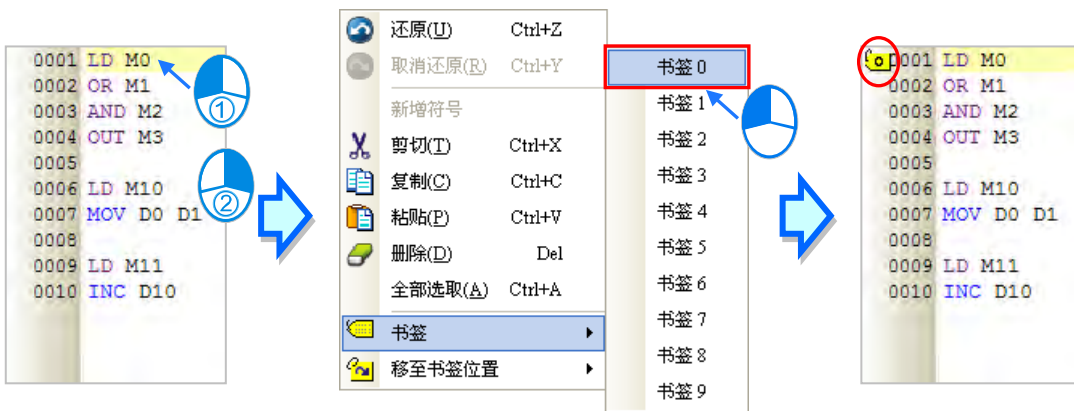
12.2.4 书签功能

书签的主要功能在于帮助用户迅速移动至加入书签的行列，而每个 IL 程序至多可加入 10 个书签。


12.2.4.1 加入书签

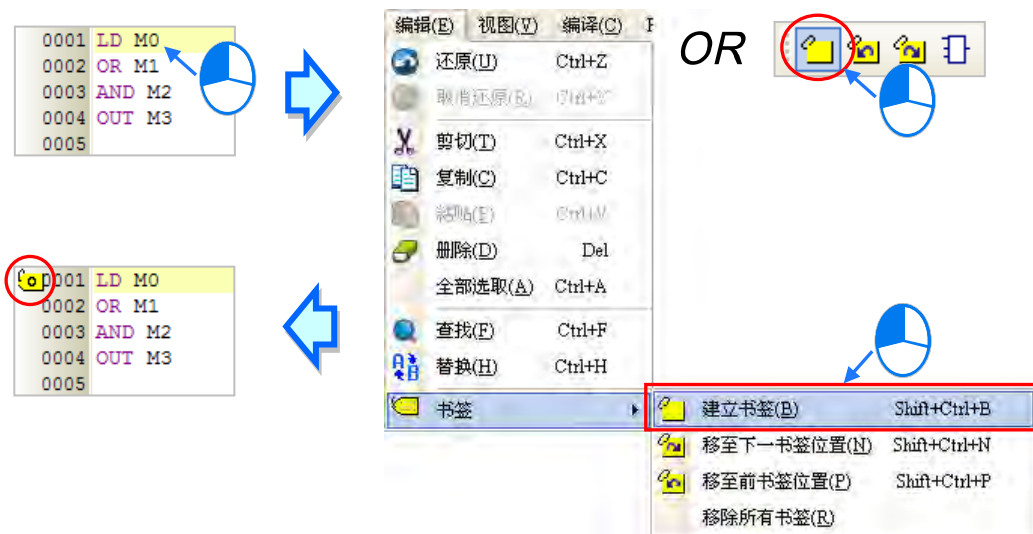
● 方法一

先用鼠标点选欲加入书签的行列后按下鼠标右键，接着于快捷选单中点选「书签」，之后再于其中选择欲加入的书签编号即可。不过若选择的编号已使用于其它行列，则该编号的书签仍会被搬移至目前选取的行列上。



● 方法二

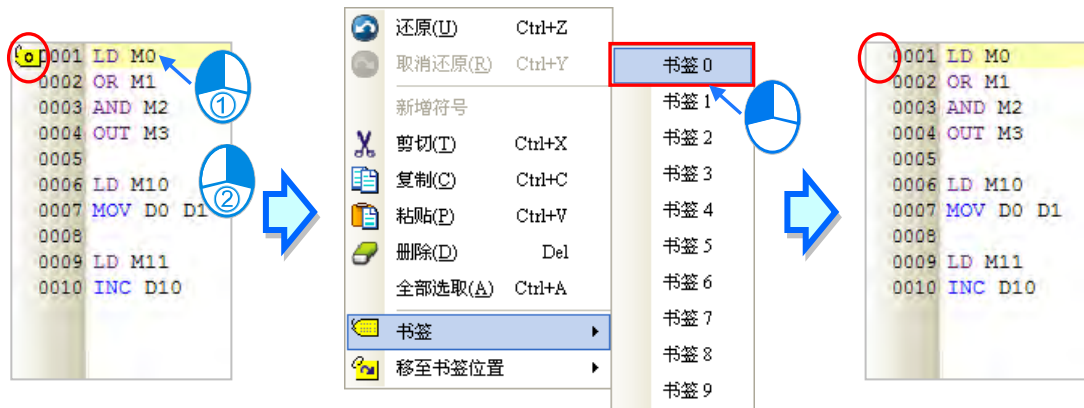
首先要在要加入书签的行列上点击一下鼠标左键，接着于图标工具栏中点击  图标，或于功能工具栏中依序点击 **编辑 (E)** > **书签** > **建立书签 (B)** 即可。不过使用此方法建立的书签编号只能按照顺序递增，用户无法自行指定。




12.2.4.2 取消书签

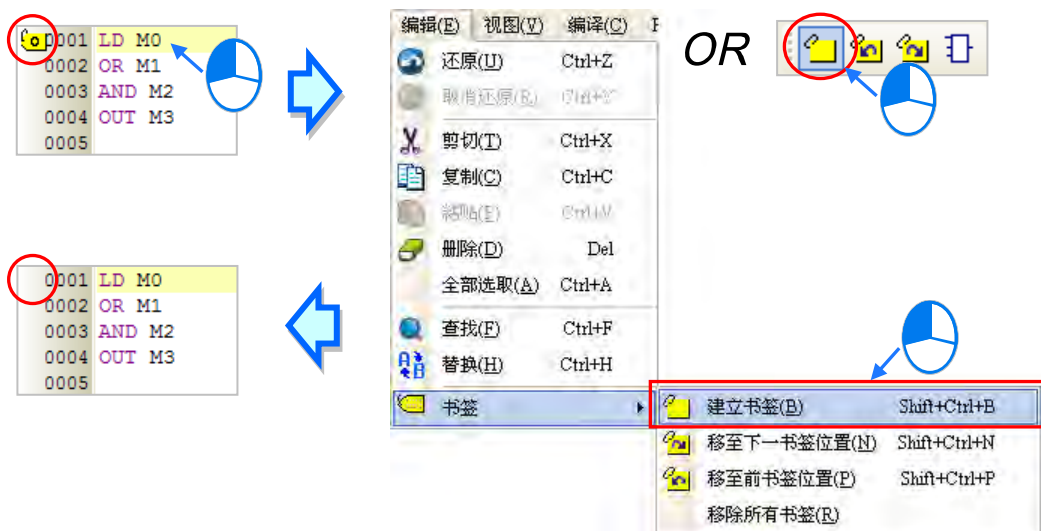
● 方法一

先用鼠标点选欲取消书签的行列后按下右键，接着于快捷选单中点选「书签」，之后再于其中点选与所选行列相同编号的书签即可。不过若选择的是其它编号时，所选行列的书签便会被变更为该编号，且若此编号的书签已使用于其它行列，则该书签仍会被搬移至目前所选取的行列上。



● 方法二

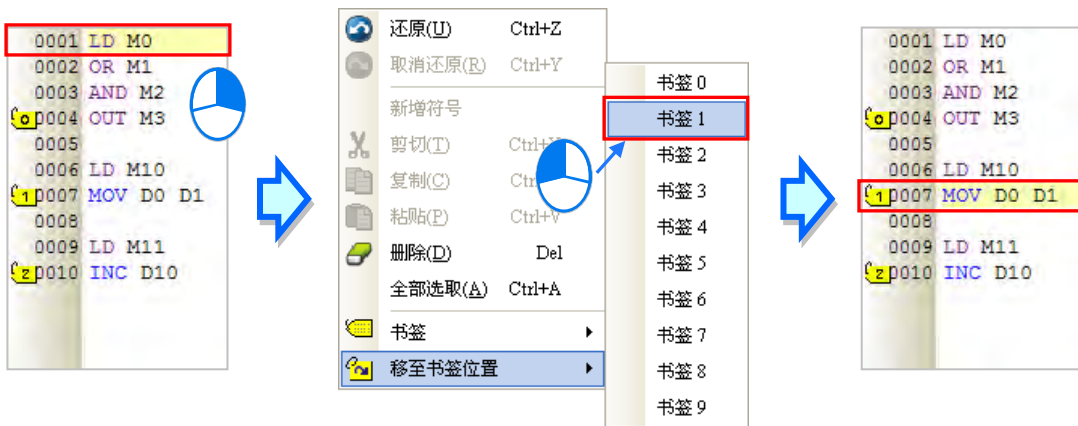
首先要在要取消书签的行列上点击一下鼠标左键，接着于图标工具栏中点击  图标，或于功能工具栏中再次点击 **编辑 (E) > 书签 > 建立书签 (B)** 即可。但若于功能工具栏中点击的是 **编辑 (E) > 书签 > 移除所有书签 (R)**，则目前编辑窗口的中的所有书签便会被全部取消。





12.2.4.3 移至书签位置

● 方法一

于编辑工作区中按下鼠标右键，接着于快捷选单中点选「移至书签位置」，之后在于目录中选择移动目标的书签编号即可。



● 方法二

当程序中有插入书签时，于功能工具栏的 **编辑 (E)** > **书签** 目录下，或于图标工具栏中按下  图标即可移动至下一个书签的位置，而按下  图标则可移动至前一个书签的位置。



第13章 结构化语言 ST

目录

13.1	认识结构化语言 ST	13-3
13.1.1	ST 程序的基本结构	13-3
13.1.2	叙述句 (Statement)	13-4
13.1.3	表达式 (Expression)	13-5
13.1.4	操作数 (Operand) 与运算符 (Operator)	13-6
13.1.5	关键词 (Keyword) 与批注 (Comment)	13-7
13.1.6	ST 程序中使用数组符号	13-9
13.1.7	ST 编程的注意事项	13-10
13.2	ISPSOft 中的 ST 叙述句结构	13-11
13.2.1	指定结构 :=	13-11
13.2.2	条件结构 IF	13-13
13.2.3	条件结构 CASE	13-15
13.2.4	循环结构 REPEAT	13-17
13.2.5	循环结构 WHILE	13-18
13.2.6	循环结构 FOR	13-19
13.2.7	API 应用指令结构	13-20
13.2.8	功能块 (FB) 调用结构	13-21
13.2.9	空白叙述	13-23
13.2.10	RETURN 叙述	13-24
13.2.11	EXIT 叙述	13-24
13.3	ISPSOft 中建立 ST 程序	13-26
13.3.1	ISPSOft 的 ST 编辑环境	13-26
13.3.2	编辑 ST 叙述	13-27
13.3.3	插入 API 应用指令与功能块	13-29
13.3.4	书签功能	13-30
13.4	ST 的程序范例	13-31
13.4.1	范例说明	13-31
13.4.2	硬件规划	13-31
13.4.3	程序规划	13-31

13.4.4	建立范例程序	13-32
--------	--------------	-------

13.1 认识结构化语言 ST

结构化语言 (Structured Text · ST) 是一种类似 C 语言或 PASCAL 等高级语言的编程方式，其最大的特色在于整个程序是由许多的文字叙述所建构而成，而相较于指令列表 (IL)，ST 则具有较高的可读性，且可透过类似数学方程式的文法来表达一个复杂的表达式，因此对于一个需要进行繁复运算的程序来说，ST 是较适合此种应用场合的开发语言。除此之外，对于习惯编写 C 语言等高级语言的程序设计者而言，ST 语言亦提供了一个较为熟悉的编程环境。

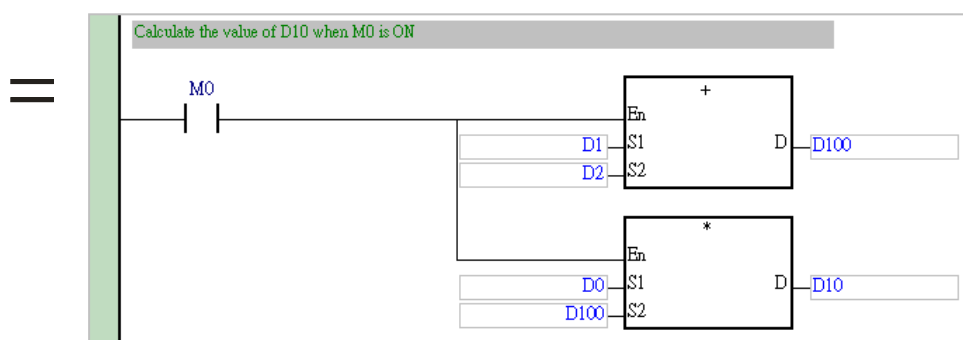
13.1.1 ST 程序的基本结构

下列便是一个很基本的 ST 程序范例，它与下方的梯形图具有同样的功能，但却能用更精简的方式来加以表达，且所展现的文字结构已经近乎人类所使用的语言文法；而在 ST 程序当中，这样具有执行意义的文字结构便称之为叙述句 (Statement)。

```

0001 (*Calculate the value of D10 when M0 is ON*)
0002 IF M0 THEN
0003     D10 := D0*(D1+D2) ;
0004 END_IF ;
0005

```



上述的 ST 程序范例可被解析为如下的结构，而一个 ST 程序便是由许多这样的基本结构所组成。

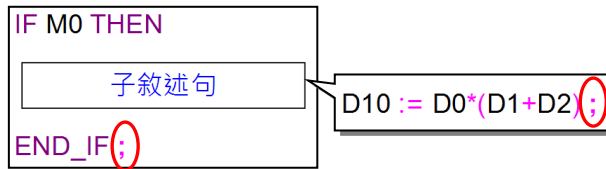
叙述句 (Statement)



* 在 ST 的语法中，『:=』符号代表指定的意义，而『=』符号则代表等于的意义，此点与 C 语言的表达方式不同。

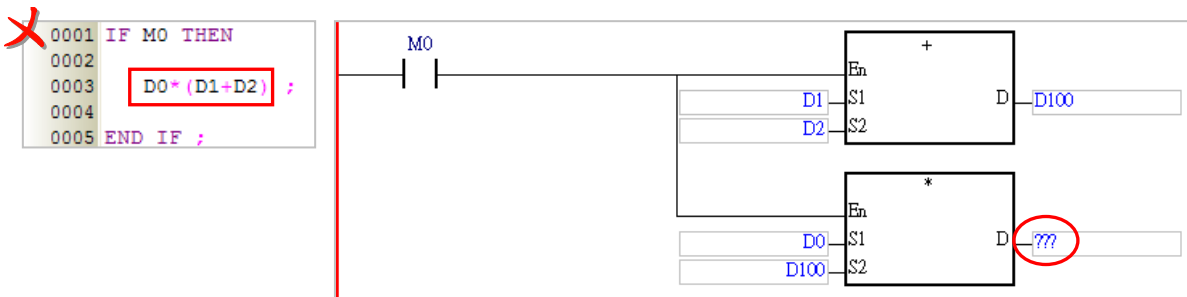
13.1.2 叙述句 (Statement)

叙述句是 ST 语言中最基本的执行单位，它代表的是一个可被完整执行的工作。而如同前述的范例，一个完整叙述句的呈现可能并不局限在同一行文字当中，但它必定是以分号『;』来做为该叙述句的结尾。另外，一个叙述句中也可能再包含多个或多层的子叙述句，且不论其所处的位置为何，只要是一个完整叙述句的结尾，最后必定会跟随着『;』符号。



13

一个完整的叙述句相当于一个具备完整功能的梯形图区段 (NETWORK)，它必须能够明确的表达一件工作。以上述程序中的子叙述句为例，它所执行的工作便是将各装置的内容值依照数学式表达的顺序 $D0 * (D1 + D2)$ 来进行运算，并将运算的结果指定给装置 D10；但如下图红色框线中的内容便不是一个合法的叙述句，它只是一个表达式 (Expression)，代表的只是一个数学运算的数值，却不是一件具体的工作，就如同右侧缺少输出操作数的梯形图程序一样，并不是一个可被完整执行的程序。



由前述可知，一个 ST 程序是由许多的叙述句所组成，而一个叙述句的结构也同时包含了许多元素；下图便是一个 ST 的程序范例，而其中每个红框的位置都代表一个完整的叙述句。

```

0001
0002 IF M0 THEN
0003     M2 := TRUE ;
0004 ELSIF M1 THEN
0005     M2 := FALSE ;
0006 END_IF ;
0007
0008 M20 := M10 ;
0009
0010 M21 := M11 ;
0011
0012 IF M2 THEN
0013     ROR (D0, 2) ;
0014 END_IF ;
0015
    
```

* 在 ST 的语法中，『:=』符号代表指定的意义，而『=』符号则代表等于的意义，此点与 C 语言的表达方式不同。

13.1.3 表达式 (Expression)

表达式在叙述句结构当中是一个相当重要的组成元素，它代表的是一个「值」，例如 TRUE 或 FALSE 的布尔值，亦或是 16 或 -3 的整数值；而表达式视使用的场合，可以是一个表达式或是一个常数，当然也可以是一个变量符号或装置。下列便是一些表达式的例子。

- $M0 \& M1$ (布尔值的表达式)

此表达式所代表的是 $M0$ 与 $M1$ 进行 $\&$ 运算后的布尔值。

- $M0 = FALSE$ (布尔值的表达式)

此表达式所代表的是「 $M0=FALSE$ 」这个条件是否成立。当 $M0$ 的值为 ON 时，此表达式所代表的布尔值便是 FALSE；但若 $M0$ 为 OFF 时，因条件的成立，表达式所代表的布尔值便会为 TRUE。

- $M0$ (布尔值的表达式)

此表达式则直接以 $M0$ 的值当作其代表的布尔值。当 $M0$ 的值为 ON 时，表达式所代表的布尔值便会为 TRUE，而当 $M0$ 为 OFF 时，表达式所代表的布尔值则为 FALSE。

- $D1 + D2$ (数值的表达式)

此表达式代表的数值为 $D1$ 与 $D2$ 相加后的运算结果。

- $D0$ (数值的表达式)

此表达式则直接以 $D0$ 的当前值做为其代表的数值。

- $D2 = D0 + D1$ (布尔值的表达式)

这是一个相当容易被混淆的布尔值表达式，它代表的是「 $D2=D0+D1$ 」这个条件是否成立。也就是说当 $D0$ 与 $D1$ 相加的结果等于 $D2$ 的当前值时，表达式代表的便是布尔值 TRUE；相反的，如果 $D0$ 与 $D1$ 相加的结果不等于 $D2$ 的当前值时，表达式所代表的即是布尔值 FALSE。

- $D2 := D0 + D1 ;$ (叙述句，非表达式)

此为一个完整的叙述句而非表达式，其代表的意义是「将 $D0$ 与 $D1$ 的相加结果指定给 $D2$ 」的一个工作；不过这个叙述句也是由「 $D2$ 」与「 $D0+D1$ 」这两个表达式所组成的。

请参考下列范例，此程序主要由两个 IF 结构的叙述句与其子叙述句所组成。当程序执行至第一个 IF 叙述句时，系统会先判断表达式「 $M0$ 」的值是否为 TRUE，若此时装置 $M0$ 的状态为 ON 时，系统便会执行「将 $D21+D22$ 的值指定给 $D20$ 」的叙述句工作；而执行至第二个 IF 叙述句时，系统会先判断 $D0 * (D1+D2)$ 的运算结果是否等于 $D20$ 的值，若两方相等，则代表表达式「 $D20 = D0 * (D1+D2)$ 」的布尔值为 TRUE，此时系统便会执行「将 $M1$ 的值设为 TRUE」的叙述句工作；但若两方不相等，系统则会执行「将 $M1$ 的值设为 FALSE」的叙述句工作。

```

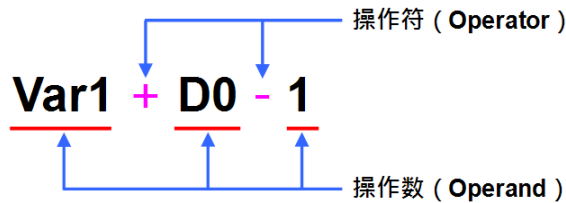
0001 IF M0 THEN
0002     D20 := D21 + D22 ;
0003 END_IF;
0004
0005 IF D20 = D0*(D1+D2) THEN
0006     M1 := TRUE ;
0007 ELSE
0008     M1 := FALSE ;
0009 END_IF;

```

13.1.4 操作数 (Operand) 与运算符 (Operator)

操作数与运算符为组成一个表达式的基本元素。操作数指的是参与运算的对象，而运算符则代表所执行的运算动作，例如表达式 $D0 + D1$ 中， $D0$ 与 $D1$ 都是操作数，而 $+$ 号则是运算符。

由上一节的各个例子可知，表达式可为一群操作数与运算符的组合，但也可单独的以一个操作数来呈现一个表达式，而其中做为操作数的可为装置、变量符号或是常数。



13

如同数学式一般，运算符本身也有执行运算的优先级，而当优先的等级相同时，进行运算的顺序便会是由左至右。下表即为 ISPSOft 中 ST 语法的运算符一览表。

符号	功能	数据格式		表达式范例		优先等级
		操作数	运算结果 (表达式的值)	表达式	值	
()	区块优先	不限	不限	$(D0 + 6) * 3$	33	↑ 最高
**	次方	浮点数	浮点数	$2.0 ** 5.0$	3.2E+1	
-	负号	带号数值	带号数值	$-(D0 + 3)$	-8	
NOT	逻辑反向	布尔	布尔	NOT M0	TRUE	
*	乘法	任何数值	任何数值	$D0 * 3$	15	
/	除法	任何数值	任何数值	$15 / D0$	3	
MOD	除法取余数	整数	整数	$D0 \text{ MOD } 3$	2	
+, -	加法、减法	任何数值	任何数值	$D0 + 3$	8	
<, >, <=, >=	数值比较	任何数值	布尔	$D0 > 2$	TRUE	
=, <>	等于、不等于	任何数值	布尔	$D0 <> 2$	TRUE	
		布尔		$M0 = \text{TRUE}$	FALSE	
AND, &	"及"运算	布尔	布尔	$M0 \& M1$	FALSE	
XOR	"互斥或"运算	布尔	布尔	$M0 \text{ XOR } M1$	TRUE	
OR	"或"运算	布尔	布尔	$M0 \text{ OR } M1$	TRUE	最低

*1. 在上述的表达式范例中，各装置的默认值为 $M0=\text{FALSE}$ ， $M1=\text{TRUE}$ ， $D0=5$ 。

*2. 使用次方运算符「**」时，其搭配的操作数必须使用常数或宣告为浮点格式的变量符号，而不可直接使用装置名称。

13.1.5 关键词 (Keyword) 与批注 (Comment)

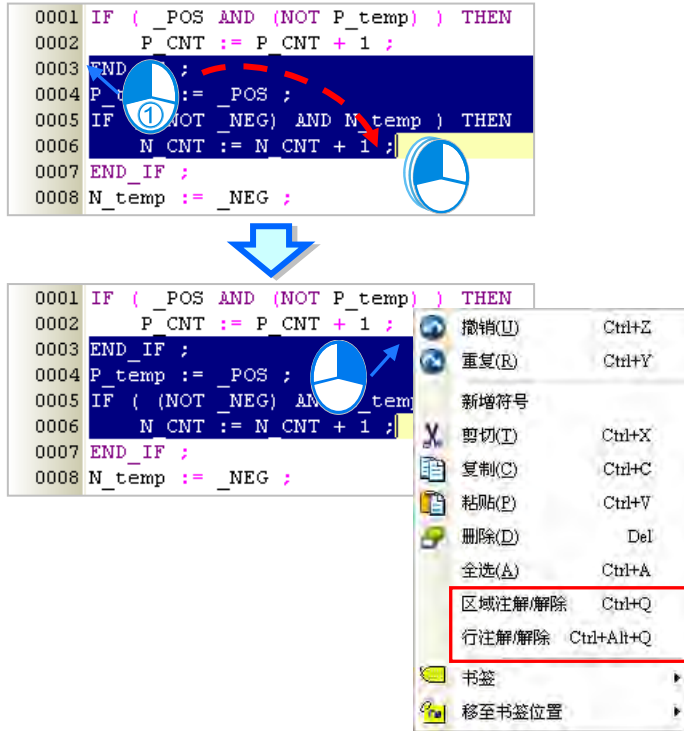
在程序中具有特殊意义的文字便称之为关键词，例如「TRUE」与「FALSE」在程序中代表的是一个布尔值，而前述范例中的「IF」则代表叙述句所要处理的工作；除此之外，包括 API 应用指令的名称也会被视为关键词，且为了避免程序的解读错误，在程序中被视为关键词的文字将无法被宣告为变量符号的名称，但若只是被包含在符号名称当中则是允许的，例如「FIFO」及「_IF」。

在 ST 语法中，被框在 (* 与 *) 和 /* 与 */ 符号之间的文字都会被视为批注，也可以使用 // 双斜线做单行批注而其内部不论是否包含叙述句、表达式，亦或是任何的关键词，当进行编译时，系统都会自动略过这段内容；除此之外，只要在不破坏文体结构的前提下，批注可被加在程序中的任何位置，但须留意程序的可读性。

请参考下列两个范例，其中编辑于左侧范例中的所有批注都是合法的，而右侧的批注则因为将关键词「END_IF」切开而破坏其完整性，与单行注解将必要的结尾符号；切断，因此编译时便会产生错误。

<pre> 0001 (*Note*) 0002 0003 IF (*Note*) M100 = TRUE THEN 0004 0005 D10 (*Note*) := D10 * (D1+D2); 0006 0007 END_IF (*Note*); (*Note*) 0008 0009 (*Note*) 0010 0011 //Note 0012 0013 IF /*Note*/ M100 = TRUE THEN 0014 0015 D10 /*Note*/ := D10 * (D1+D2); 0016 0017 END_IF/*Note*/;/*Note*/ 0018 0019 /*Note*/ </pre>	<pre> 0001 0002 IF M100 = TRUE THEN 0003 0004 D10 := D10 * (D1+D2); 0005 0006 END (*Note*)_IF; 0007 0008 0009 IF M100 = TRUE THEN 0010 0011 D10 := //D10 * (D1+D2); 0012 0013 END_/*Note*/_IF; 0014 </pre>
---	--

ISPSOft 中也提供了将程序转换为区域批注或行批注的快捷功能。先以鼠标左键，选取一段 ST 语言区块后，按下鼠标右键开启快捷选单。



若单击 **区域批注/解除** 后，所选区块前后便会自动加上批注符号，如下图左；若单击 **行批注/解除** 后，所选区块的每一行前后便加上批注符号，如下图右。

```

0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 (*P temp := POS ;
0005 IF ( (NOT NEG) AND N temp ) THEN
0006     N_CNT := N_CNT + 1 ;
0007 END_IF ;*)
0008 N_temp := _NEG ;
    
```

区域批注

```

0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 (*P temp := POS ;*)
0005 (*IF ( (NOT NEG) AND N temp ) THEN*)
0006     (* N_CNT := N_CNT + 1 ; *)
0007 (*END_IF ;*)
0008 N_temp := _NEG ;
    
```

行批注

13.1.6 ST 程序中使用数组符号

在 ISPSOft 的 ST 程序中使用数组符号时，其表达格式为 **符号名称[索引]**，其中索引的部份可为常数或变量符号，不过只有在 AH/AS 系列机种的项目中才可使用变量符号做为索引值；而不论是变量符号或常数，索引值的内容都必须由 0 开始，同时也不可大于或等于数组的大小，例如大小为 10 的数组符号，其索引值的范围则必须为 0~9。关于数组符号的使用说明请参考第 6.2.3 节的相关内容。

```
0001 IF M0 THEN
0002
0003 Ary_A[0] := Ary_B[IDX] ;
0004
0005 END_IF ;
```

除此之外，使用数组时还须注意下列几点事项。

- 使用数组符号时只可对单一的数组元素进行操作，而无法将代表整个数组的变量符号当作操作数。
- 目前 ISPSOft 的数组类型的变量符号只支持至一维数组；但自定义数据形态中的结构 (STRUCTURE) 内所定义的元素支持最多三维数组，此类变量符号可以下形式书写，如下图中将 D0 的值写入自定义数据类型变量符号 vardut 中指定元素 pt1 中三维数组的第[1,1,1]个。自定义数据类型请参考第 8 章。

```
0001 IF M0 THEN
0002
0003 vardut.pt1[1,1,1] := D0;
0004
0005 END_IF;
0006
```

- 当数组索引为变量符号时，若在操作的过程中不慎造成索引值超出合法范围，此时 PLC 仍会继续运作且不会有错误码产生，但实际参与运算的数据便可能已产生错乱。
- 使用常数索引值的数组符号允许使用**间接指定寄存器**来进行修饰，但若使用变量符号当作数组索引时，系统便不允许再加上**间接指定寄存器**。关于**间接指定寄存器**的说明请参考第 6.1.5 节。

13.1.7 ST 编程的注意事项

- 在不破坏关键词或叙述句结构的前提下，任何的换行或空白都是被允许的。
- 原则上全角与半角文字在 ISPSoft 中会被视为不同的字符，因此请尽量避免使用全角文字以免造成识别及解读上的误判。
- 在 ST 的语法中，英文无大小写的分别，亦即 IF、If 与 if 的意义是相同的。(此点与 C 语言不同)
- 在 ST 语言中使用常数时，其表达的格式如下。
 - 10 进制：23456（未加任何标注的数字会被视为 10 进制）
 - 16 进制：16#5BA0（数字前方加上 16#）
 - 8 进制：8#55640（数字前方加上 8#）
 - 2 进制：2#101101110100000（数字前方加上 2#）
 - 字符串：“XYZ”（以双引号包住字符串）
 - BOOL：TRUE 或 FALSE。
- 在 ISPSoft 的 ST 语法中，原则上一个表达式里的操作数必须为相同的数据类型才可以进行运算，但若操作数为 INT 与 WORD 类型的数值时，两者之间便可进行混合运算，且 DINT 与 DWORD 之间的混合运算也同样是合法的，不过若是不同数据长度的操作数则不允许进行混合运算。

```

0001 Var_0 := Var_INT + Var_WORD ; ✓
0002
0003 Var_1 := Var_DINT + Var_DWORD ; ✓
0004
0005 Var_2 := Var_INT + Var_DINT ; ✗

```

- 进行运算时，不论是 INT 系列或 WORD 系列的数据，系统都会将其视为带符号的整数来处理。
- ST 语言可编辑的行数并没有限制，但仍必需考虑主机的内存容量。
- ISPSoft 的 ST 编辑环境支持 **复制/剪下/贴上** 的功能，因此用户可从其他的文本编辑器中将编辑好的内容复制到 ISPSoft 的 ST 编辑工作区中。
- 下表则为 ST 语言不支持的指令。

AH/AS 系列機種
所有 IL 的专用指令皆不可使用，例如 LD、LDI、OUT、SET、RST、LD&、LDP...等。
所有脉冲型的 API 指令皆不支持，例如 ROLP、RORP、BCDP、TRDP...等。
所有四则逻辑运算、逻辑控制、接点类型逻辑运算与结构指令类别的指令皆不支持。另外也不支持 PLS、PLF、MOV、DMOV、DFMOV、\$MOV、MC、MCR、CJ、JMP、GOEND...等指令。
一般计数器编号搭配的 CNT、DCNT 指令，以及子程序型定时器编号搭配 TMR 指令

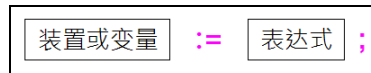
- 下表则为 ST 语言有支持的指令，但须有所限制使用的规范事项：

AS 系列機種
所有高速输出指令 (API 27xx)，如 DPLSY、DDRVI 等，使用前请参考 DDRVI 指令说明。
高速计数器搭配 DCNT 指令时，判断式需永远被执行，以便刷新当前计数值。
通讯类的指令，而且需等待设定完成其目标指令，须注意程序流程可让指令被扫描执行。

13.2 ISPSoft 中的 ST 叙述句结构

13.2.1 指定结构 :=

- 格式



- 功能说明

将右侧表达式所代表的值指定（传送）给左侧的装置或变量符号。

- 使用规则与限制

- 结构右侧的表达式可为常数、装置、变量符号或任何表达式，但左侧只能是装置或变量符号。
- 指定结构两侧的操作数必须符合下列规则才可传递数据。
 - (a) M、S、T、C、HC 装置与 BOOL、STEP、TIMER、COUNTER 等位类型的符号可互传。
 - (b) D、L 装置可与所有 BOOL、STEP、REAL、STRING 以外所有类型的符号互传数据。
 - (c) T、C 装置可与 D、L 装置及 WORD、INT、TIMER、COUNTER 等类型的符号互传数据。
 - (d) D、L、HC 装置与指定 HC 装置的 COUNTER 符号可与 DWORD 与 DINT 的符号互传数据。
 - (e) WORD 系列与 INT 系列可互传数据。
 - (f) REAL 系列与 STRING 只可与相同类型且长度相同的符号互相传递数据。
 - (g) WORD 与 DWORD 可以互传数据，但 DWORD 传到 WORD 数据可能会遗失。
 - (h) INT 与 DINT 可以互传数据，但 DINT 传到 INT 数据可能会遗失。

- 范例

EX1：将 M0 的值传递给变数 OUT_0。



EX2：将装置 M0 设置为 ON。



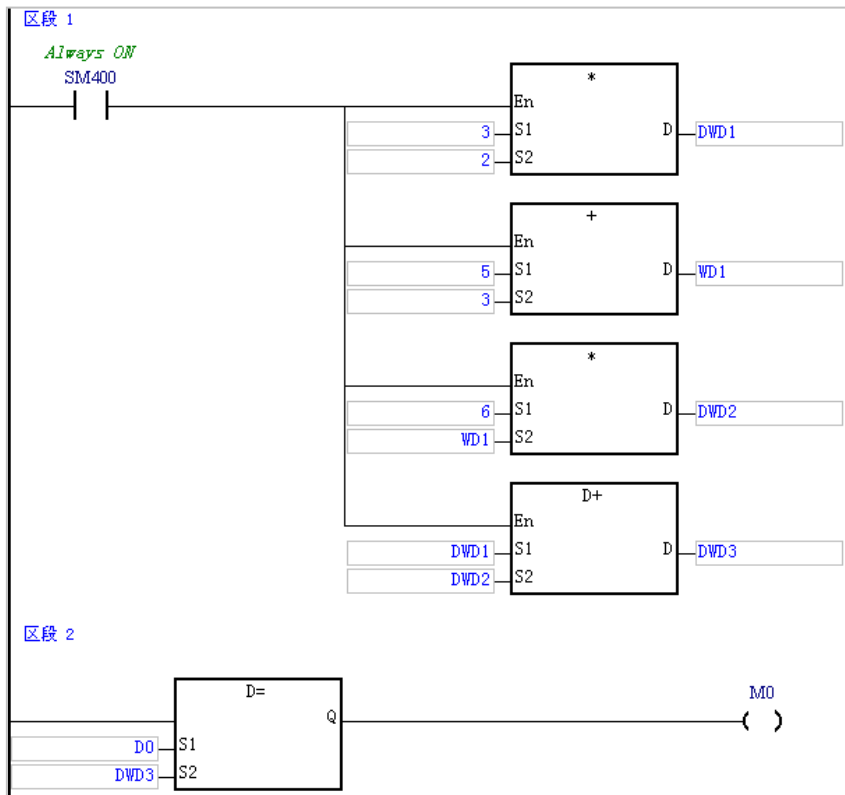
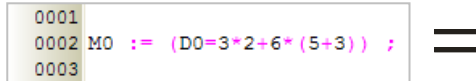
EX3：将 D1 与 D2 的内容值相加后传递给 D0。



EX4：将变量 DATA 的内容值指定为 3。



EX5：确认 D0 的内容值是否等于 $3*2+6*(5+3)$ ，并把判断结果传送给装置 M0。

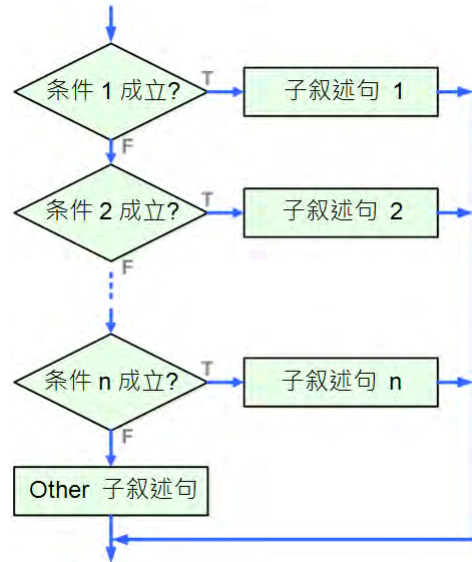


EX6：下表则为几个错误的范例。

ST 叙述句	说明
$D0 := M0 ;$	两侧的数据类型不符。
$D0 := V_REAL ;$	REAL 符号不可直接与 D 装置互传数据。
$V_LREAL := V_REAL ;$	REAL 系列的符号必须长度相同才可互传数据。
$V_DWORD := V_TIMER ;$	TIMER 符号不可传值给 DWORD 符号，但可传给 WORD。
$V_WORD := HC0 ;$	HC 装置不可传值给 WORD 符号，但可传给 DWORD。

13.2.2 条件结构 IF

● 格式



13

● 功能说明

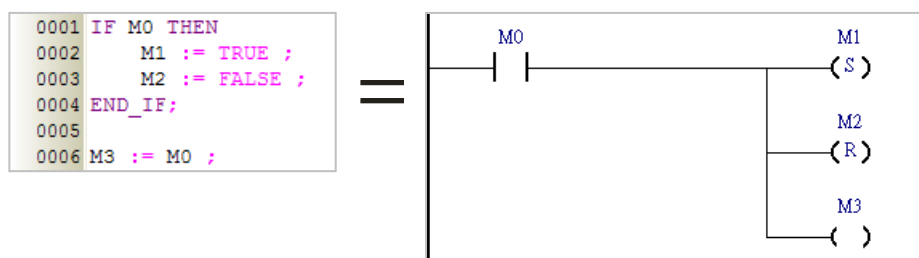
若「布尔值表达式 1」的值为真 (True) 时，程序便会执行「子叙述句 1」后离开；若为否 (False) 则往下判断，并依此类推...；当所有的判断值都为否时，最后便会执行「Other 子叙述句」后离开。

● 使用规则与限制

- 布尔值表达式可为装置、变量符号或是任何表达式，但不可为常数。
- 子叙述句的内容可以是任何合法的完整叙述句，当然也可以是另一个 IF 结构。
- 跟在关键词 THEN 或 ELSE 后面的子叙述句数量并没有限制。
- 结构中的 ELSIF 与 ELSE 部份均可省略，而当 ELSE 被省略时，若当中所有的布尔值表达式都为否 (FALSE) 时，该结构最后将不会执行任何子叙述句而直接离开。
- 结构中的 ELSIF 可随意增加，但 ELSE 的部份只能有一个，且必须加在最后一段。

● 范例

EX1：如果 M0 为 ON，则把 M1 设为 ON，M2 设为 OFF，否则就直接结束此步骤；且不论 M0 的值为何，其值将会永远传递给 M3。

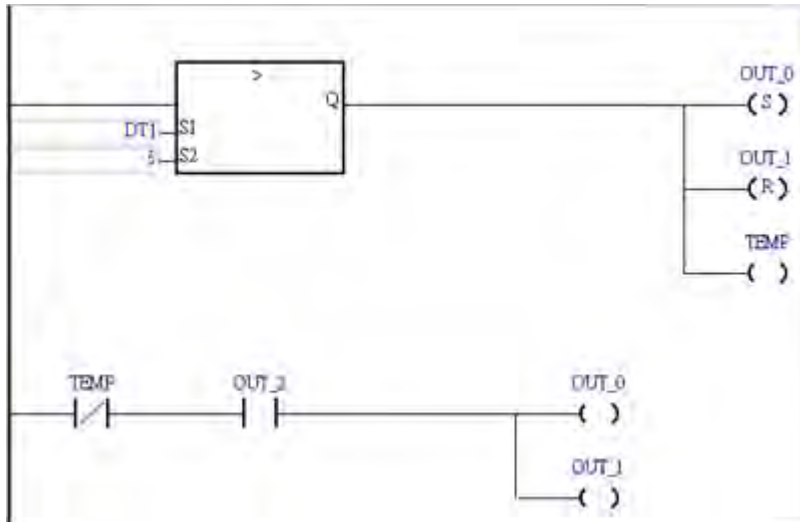


EX2 : 若 DT 的值大于 5 , 则将 OUT_0 设置为 ON , OUT_1 重置为 OFF , 否则 OUT_0 与 OUT_1 的状态都将依循 OUT_2 。

```

0001 IF DT>5 THEN
0002     OUT_0 := TRUE ;
0003     OUT_1 := FALSE ;
0004 ELSE
0005     OUT_0 := OUT_2 ;
0006     OUT_1 := OUT_2 ;
0007 END_IF;
    
```

=



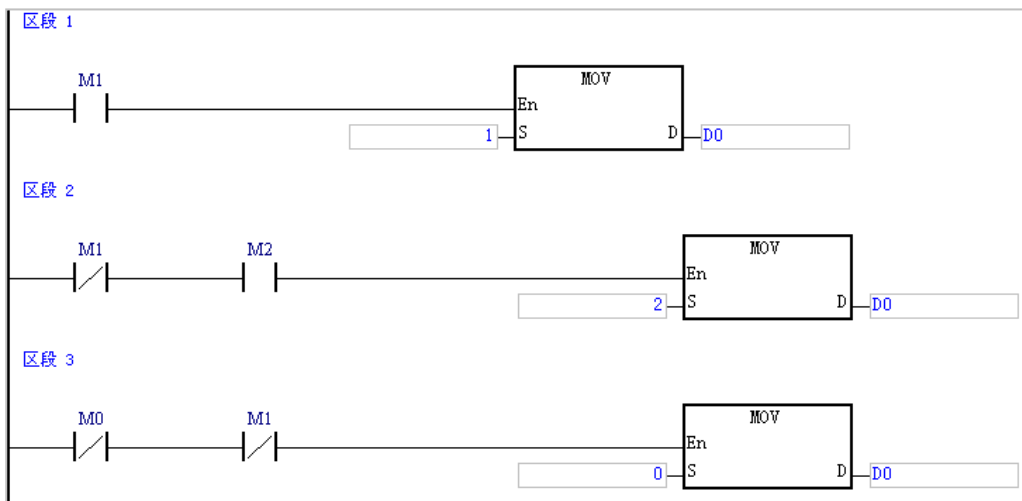
13

EX3 : 若 M1 为 ON , 则将 D0 的值设为 1 ; 但若 M1 不为 ON , 则继续判断 M2 是否为 ON , 若是 , 则将 D0 的值设为 2 ; 但若 M2 也不为 ON , 则 D0 的值将被设为 0 。

```

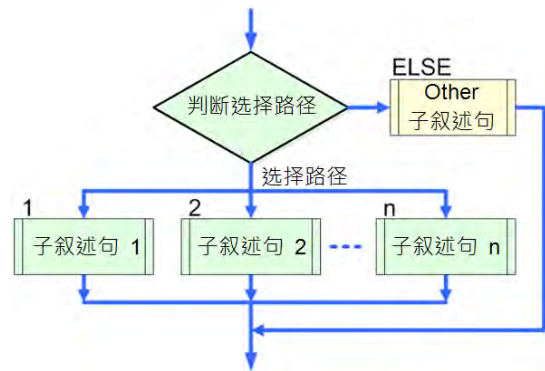
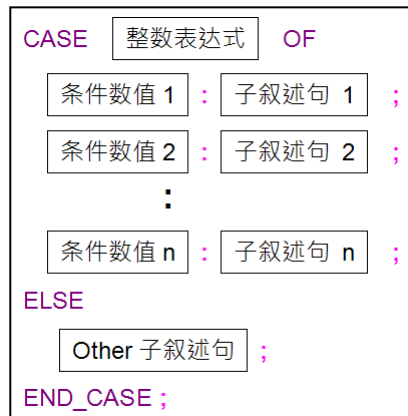
0001 IF M1 THEN
0002     D0 := 1 ;
0003 ELSIF M2 THEN
0004     D0 := 2 ;
0005 ELSE
0006     D0 := 0 ;
0007 END_IF;
    
```

=



13.2.3 条件结构 CASE

● 格式



● 功能说明

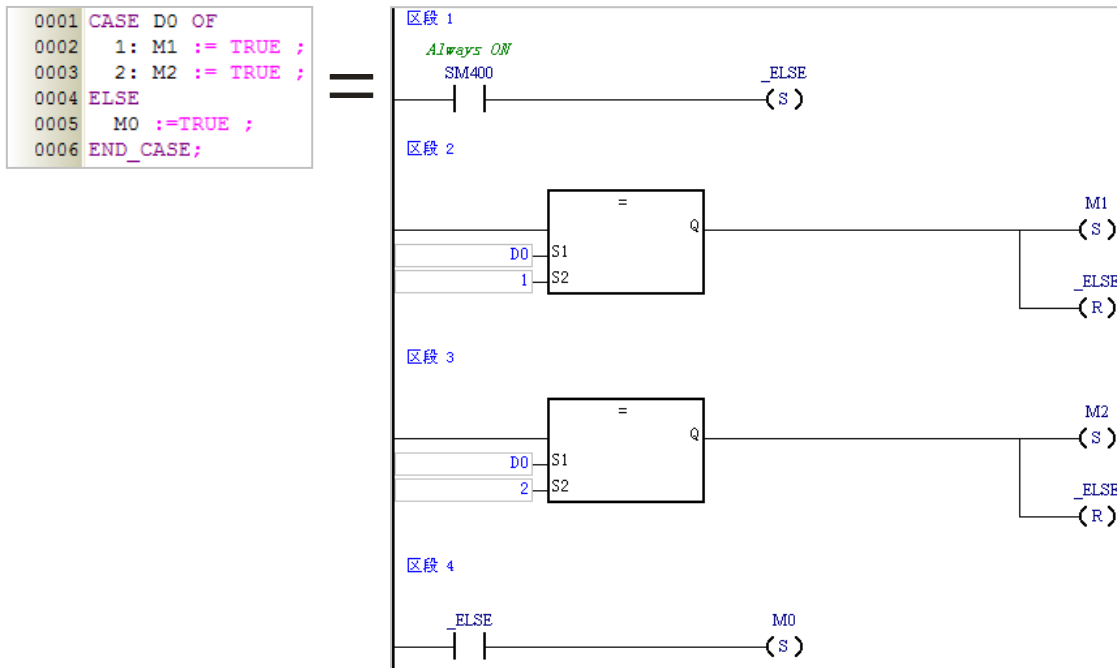
判断「**整数表达式**」的值是否在「**条件数值**」的列表中，若是，则执行该条件所对应的「**子叙述句**」后离开此结构；若否，则执行关键词「**ELSE**」之后的「**Other 子叙述句**」后离开。

● 使用规则与限制

- 上图结构中的**整数表达式**可为装置、变量符号或是任何表达式，但不可为常数，且其代表数值的数据类型必须为 INT (-32768 ~ 32767) 或 DINT (-2147483648 ~ 2147483647) 类型的整数，而若直接使用装置编号，如 D0，系统则会自动将该装置的内容值视为 INT 类型的数据；除此之外，若使用表达式且其结果不为整数时，程序将会自动略去小数而只取整数的部份。
- 列表中的条件数值不可有重复且必须为整数，其范围则须视 CASE 关键词后的整数表达式类型，即 INT 类型为 -32768 ~ 32767，而 DINT 则是 -2147483648 ~ 2147483647。
- 当有不同的条件数值却执行相同的叙述句时，编辑时便可将其合并为一列。
 - (a) 若各条件为不连续数值，则各数值间须以逗号隔开，例如 **1, 3, 5 : 子叙述句**，而此例则代表当**整数表达式**的数值满足 1、3、5 中的任何一个时，系统便会执行同一个子叙述句。
 - (b) 若各条件数值是某个连续的区块，则编辑时可使用『 .. 』符号来表示条件数值的范围，例如 **3..6 : 子叙述句** 便代表当**整数表达式**的数值落在 3 到 6 的范围内 (包含 3 与 6) 时，系统便会执行同一个子叙述句。且当用『 .. 』符号表示条件数值时，左侧数值必须小于右侧，如 **6..3** 的范围叙述便是不合法的。
- 子叙述句的内容可以是任何合法的完整叙述句，当然也可以是其他的结构。
- 在 **CASE** 结构中的每个条件数值都可对应多个子叙述句。
- 关键词 **ELSE** 与其对应的子叙述句可被省略，但实际操作时，若**整数表达式**的值没有在条件数值的列表当中，则该结构将会不执行任何子叙述句而直接离开。
- 在执行完对应条件的子叙述句后，程序便会直接跳离该 **CASE** 结构，而无需其他的跳脱指令。

● 范例

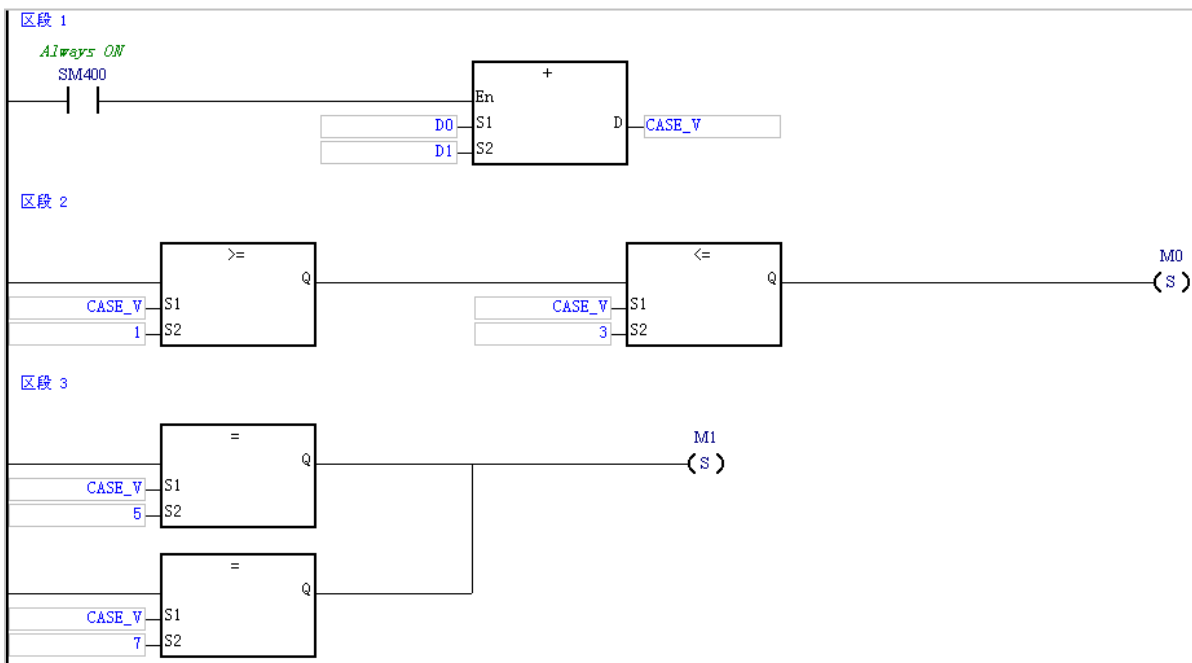
EX1：若 D0=1，则 M1 为 ON；若 D0=2，则 M2 为 ON；若 D0 为其他数值则 M0 为 ON。



EX2：先判断 D0+D1 的运算值，若该值的范围为 1~3，则将 M0 设置为 ON；但若该值为 5 或 7，则将 M1 设置为 ON，而当该值不在上述的范围时，程序将不做任何动作并直接离开此结构。

```

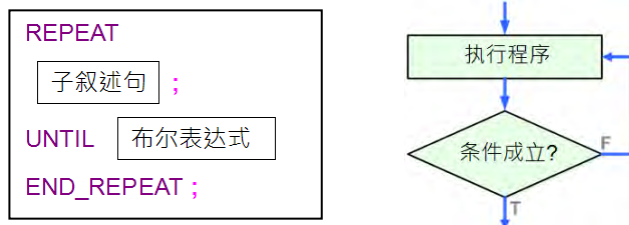
0001 CASE D0+D1 OF
0002 1..3: M0 := TRUE;
0003 5,7 : M1 := TRUE;
0004 END_CASE;
    
```



13

13.2.4 循环结构 REPEAT

● 格式



● 功能说明

先执行子叙述句后再判断「布尔表达式」的值是否为真 (True)，若其值为真，则程序会直接离开此结构；若为否 (False)，则结构内部的程序便会重复执行，直到该值为真 (True) 之后才会离开。

● 使用规则与限制

- 布尔表达式可为装置、变量或是任何表达式，但不可为常数。
- REPEAT 结构为先执行子叙述句后再判断是否离开，因此结构中的子叙述句至少会执行一次。
- 子叙述句的内容可以是任何合法的完整叙述句，当然也可以是其他的结构，且 REPEAT 结构中的子叙述句数量并没有限制，但若是循环内部还有循环的巢状结构时，最多则只能有 64 层。
- UNTIL 后的布尔表达式为离开循环的判断条件，因此请勿使用无法更新的装置或变量来当作该表达式的操作数，以避免造成无限循环。
- X 装置的值必须在程序的扫描周期完成后才会被更新，因此若要以 X 装置的状态做为离开循环的条件时，请务必在结构内部加上 IO 更新指令的叙述句，或是在中断程序中更新 X 装置的值。

● 范例

EX1：DT1 的初始值为 0，进入循环后，DT1 便会持续加 2 递增，直到 DT1 大于 100 时才会结束循环。

```

0001 DT1 := 0 ;
0002 REPEAT
0003     DT1 := DT1 + 2 ;
0004 UNTIL DT1 > 100
0005 END_REPEAT ;
  
```

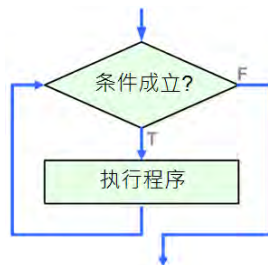
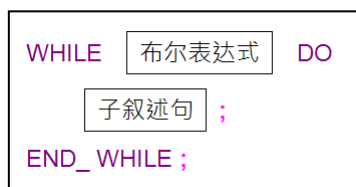
EX2：下列为一个阶层运算的范例。进入循环之前，D0 的值会先初始化为 1，而循环结束后，D0 的值便会等于 D10 内容值的阶层运算；若 D10 的值为 5，则最后的 D0 值便会等于 $5 \times 4 \times 3 \times 2 \times 1$ 。

```

0001 D10 := 5 ;
0002
0003 D0 := 1 ;
0004 REPEAT
0005     D0 := D0 * D10 ;
0006     D10 := D10 - 1 ;
0007 UNTIL D10 = 0
0008 END_REPEAT ;
  
```

13.2.5 循环结构 WHILE

● 格式



● 功能说明

先判断「布尔表达式」的值是否为真 (True)，若为真 (True)，则程序便会继续执行子叙述句，且只要该值持续为真 (True) 时，结构内部的程序便会重复执行，直到该值为否 (False) 之后才会离开。

● 使用规则与限制

- 布尔表达式可为装置、变量或是任何表达式，但不可为常数。
- **WHILE** 结构为先判断条件再决定是否执行循环中的叙述，因此内部的叙述句可能完全不执行。
- 子叙述句的内容可以是任何合法的完整叙述句，当然也可以是其他的结构，且 **WHILE** 结构中的子叙述句数量并没有限制，但若是循环内部还有循环的巢状结构时，最多则只能有 64 层。
- 跟在关键词 **WHILE** 之后的布尔表达式为离开循环的判断条件，因此请勿使用无法更新的装置或变量来当作该表达式的操作数，以避免造成无限循环。
- X 装置的值必须在程序的扫描周期完成后才会被更新，因此若要以 X 装置的状态做为离开循环的条件时，请务必在结构内部加上 **IO 更新指令** 的叙述句，或是在中断程序中更新 X 装置的值。

● 范例

EX1：DT1 的初始值为 0，进入循环后，DT1 便会持续加 2 递增，直到 DT1 大于 100 时才会结束循环。

```

0001 DT1 := 0 ;
0002
0003 WHILE DT1 <= 100 DO
0004     DT1 := DT1 + 2 ;
0005 END_WHILE ;
  
```

EX2：右侧为 **WHILE** 结构与 **REPEAT** 结构的比较，而在此例中，我们会先让离开循环的条件在进入之前就已成立。因 **REPEAT** 结构是先执行再判断，因此其结构内部的子叙述句会被执行一次，所以离开循环后的 DO 值便会等于 1；而 **WHILE** 结构则是先判断再执行，且因一开始所判断的离开条件就已成立，所以其结构内部的子叙述句完全没被执行过，因此结束后的 D1 值还是会保持为 0。

```

0001 MO := TRUE ;
0002 DO := 0 ;
0003 D1 := 0 ;
0004
0005 REPEAT
0006   DO := DO + 1 ;
0007 UNTIL MO
0008 END_REPEAT ;
0009
0010 WHILE (NOT MO) DO
0011   D1 := D1 + 1 ;
0012 END_WHILE ;

```

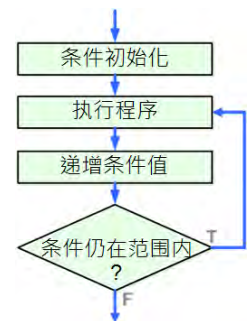
13.2.6 循环结构 FOR

● 格式

```

FOR 条件变量或装置 := 起点值表达式 TO 终点值表达式 BY 递增量表达式 DO
    子叙述句 ;
END_FOR ;

```



● 功能说明

先将「条件变量或装置」的内容设为「起点值」，之后每执行完一次循环内的程序就将「条件变量或装置」的内容值加上一次的「递增量」，接着再判断其值是否还在「起点值」至「终点值」的范围内，若是，则继续执行循环；若否，则程序便会直接离开 FOR 结构。

● 使用规则与限制

- 上述的起点值、终点值与递增量表达式可以是装置、变量符号、表达式或常数，且其代表数值的数据类型必须为 INT (-32768 ~ 32767) 或 DINT (-2147483648 ~ 2147483647) 类型的整数，而若直接使用装置编号，如 D0，系统则会自动将该装置的内容值视为 INT 类型的数据；除此之外，若使用表达式且其结果不为整数时，程序将会自动略去小数而只取整数的部份。
- 当起点值小于终点值时，所设定的递增量便必须为负数，否则将会造成无限循环。
- 条件变量的数据类型亦必须是 INT 或 DINT，且当条件变量或装置在循环的程序当中有参与运算时，请小心避免因变更到它的值而造成无限循环。
- 子叙述句的内容可以是任何合法的完整叙述句，当然也可以是其他的结构，且 FOR 结构中的子叙述句数量并没有限制，但若是循环内部还有循环的巢状结构时，最多则只能有 64 层。
- 实际运作时，系统会先将条件变量或装置的内容值加上递增量后，再判断是否可以离开循环，但若加上递增量之后的值会产生溢位时，程序便会因为离开的条件无法成立而造成无限循环。

● 范例

在循环的执行过程中，符号 INDEX 会由 1 开始每次递增 2，一直到第 5 次的循环结束后，INDEX 的值会再被加上 2 而等于 11，而此时系统便会判断已超过条件范围（1~9）而离开循环；且因符号 SU 的初始值为 0，因此离开循环后，SU 的值便会是 1~9 的奇数平方和，即 $1^2 + 3^2 + 5^2 + 7^2 + 9^2$ 。

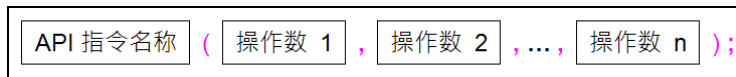
```

0001 SU := 0 ;
0002
0003 FOR INDEX := 1 TO 9 BY 2 DO
0004     SU := INDEX * INDEX + SU ;
0005 END_FOR ;
0006
    
```

13.2.7 API 应用指令结构

13

● 格式



● 功能说明

执行 API 应用指令的功能。

● 使用规则与限制

- ST 程序中执行 API 指令时，本身并不须要指定 En 端点的状态，只要程序执行到该叙述句时，对应的 API 指令就会被执行。不过应用上还是可以搭配 IF 结构来达到具有 En 端点的效果。
- 只要不破坏操作数与关键词的完整性，任何的换行都是被允许的。
- 在 ST 语法中的 API 应用指令也是一个叙述句，因此最后的结尾一样要加上分号『;』。
- 指令中的每个操作数都必须被加在括号中，而可用的操作数则请参考各机种的指令手册。
- 内部操作数的排列顺序不可变动，其顺序请参考各机种的指令手册。
- 在 ISPSOft 中的 API 应用指令并不支持回传值的功能，因此 `M0 := CMP (D0, D10, M0);` 的叙述句便是不合法的。
- 只要指令的结构不变，其文字的换行是允许的。

✓

```

0001 CMP (
0002     D0,
0003     D10,
0004     M0
0005 ) ;
    
```

✓

```

0001 CMP
0002 ( D0, D10, M0 ) ;
0003
0004
0005
    
```

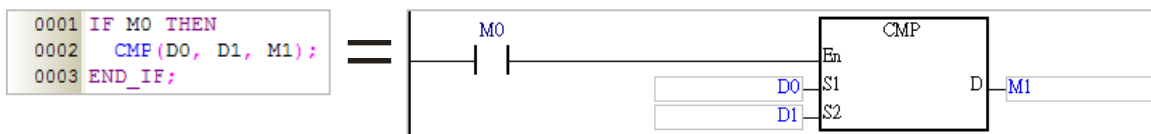
✗

```

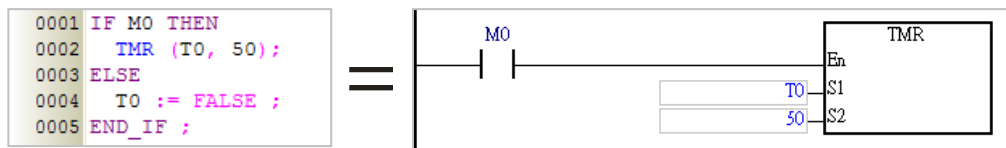
0001 CM
0002 P ( D0, D10, M0 ) ;
0003
0004
0005
    
```

● 范例

EX1：搭配 IF 结构使用 CMP 指令。

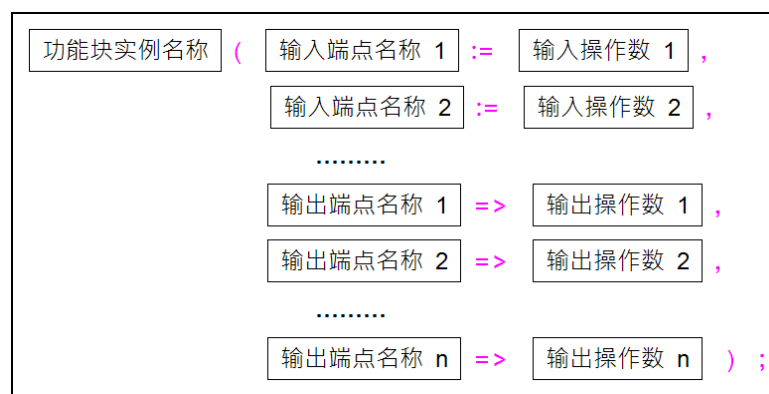


EX2：因在 ST 程序中的 API 应用指令无法操作 En 端点，因此若使用了定时器指令（TMR）时，请务必于程序中加入重置定时器的叙述句，如下例的行号 4，否则该定时器将无法归零。



13.2.8 功能块 (FB) 调用结构

● 格式

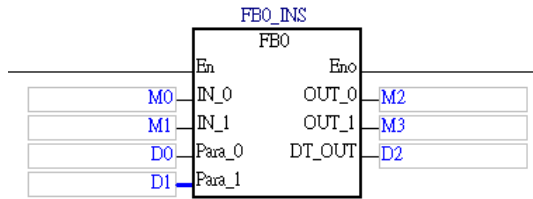


● 功能说明

执行功能块的功能。

● 使用规则与限制

- ST 程序中调用功能块时，本身并不须要指定 En 端点的状态，只要程序执行到该叙述句，被调用的功能块就会执行。不过应用上还是可以搭配 IF 结构来达到具有 En 端点的效果。
- 只要不破坏操作数与关键词的完整性，任何的换行都是被允许的。
- 功能块的结构也是一个叙述句，因此最后的结尾一样要加上分号『;』。
- 功能块的每个输出输入端点与其对应的操作数都必须被包含在括号之中，且每组端点与操作数之间须以逗号隔开，而可用的操作数类型则必须依据各功能块的定义来加以指定。
- 括号之前的名称应为**功能块实例**而非**功能块定义**，亦即应输入被宣告成功能块类型的变量符号名称。关于功能块定义与与功能块实例的详细说明请参考第 7 章的相关内容。
- 输入侧端点所使用的指定符号为「:=」，而输出侧端点所使用的指定符号则是「=>」。
- 结构内的端点与操作数顺序必须依照功能块的定义，而不可任意变动，其顺序则为排除 En/Eno 端点后，先由上而下列出所有输入侧端点，其中 VAR_IN_OUT 类别的端点则被归类为输入侧；接着再由上而下列出所有输出侧的端点。请参考下列功能块的梯形图图标与 ST 结构。

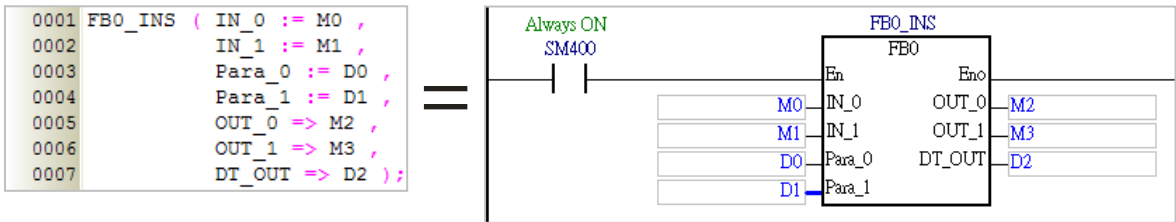


```

0001 FBO_INS ( IN_0 := M0 ,
0002           IN_1 := M1 ,
0003           Para_0 := D0 ,
0004           Para_1 := D1 ,
0005           OUT_0 => M2 ,
0006           OUT_1 => M3 ,
0007           DT_OUT => D2 );
    
```

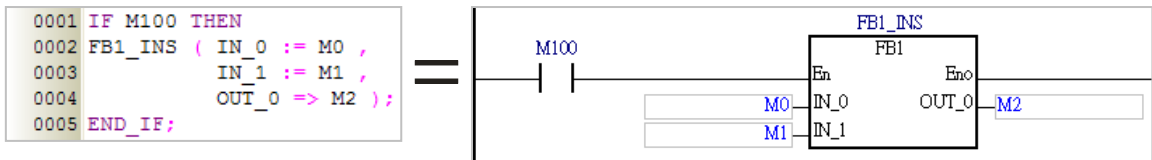
● 范例

EX1：在 ST 的程序中，只要执行到调用功能块的叙述句时，被调用的功能块就会执行。



13

EX2：搭配 IF 结构来执行功能块的功能。

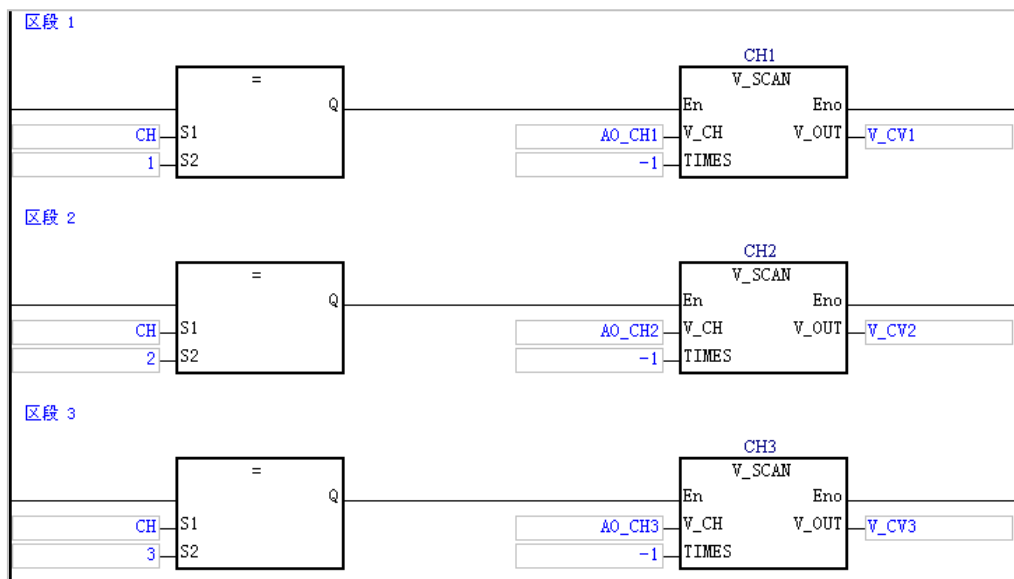


EX3：根据变量符号 CH 的值来决定执行哪一个功能块实例。

```

0001 CASE CH OF
0002   1: CH1 ( V_CH := AO_CH1,
0003           TIMES := -1,
0004           V_OUT => V_CV1 );
0005
0006   2: CH2 ( V_CH := AO_CH2,
0007           TIMES := -1,
0008           V_OUT => V_CV2 );
0009
0010   3: CH3 ( V_CH := AO_CH3,
0011           TIMES := -1,
0012           V_OUT => V_CV3 );
0013 END_CASE;
    
```

=



13.2.9 空白叙述

- 格式

```

;
    
```

- 功能说明

用于必须加入叙述句，但却又无须执行任何动作的应用场合。

- 范例

下图为一个不执行任何动作的 FOR 循环，此循环的功能在于将程序停滞在循环的处理中以达到延迟 (DELAY) 执行的效果，而调整循环次数便可调整延迟的时间。

```

0001 FOR D100:=1 TO 3000 BY 1 DO
0002   ;
0003 END_FOR;
    
```


13.2.10 RETURN 叙述

- 格式

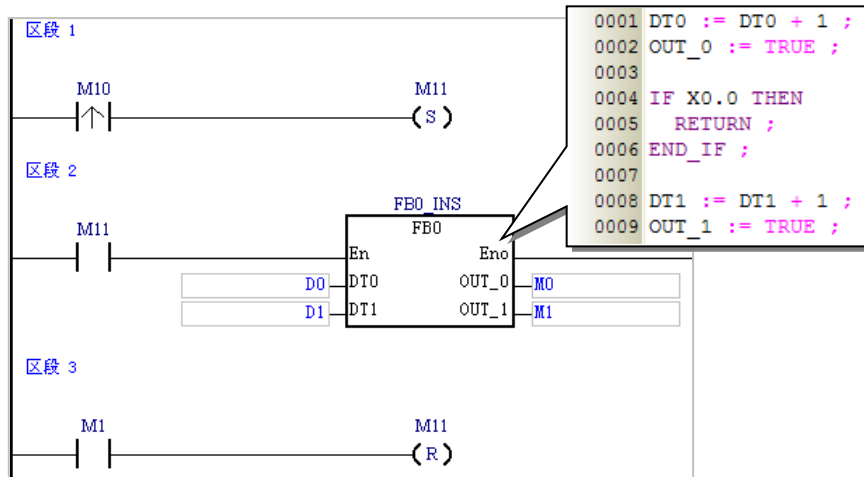
```
RETURN ;
```

- 功能说明

此叙述句，只可加在用 ST 语言撰写的功能块程序当中，其功能为强制离开目前正在执行的功能块。

- 范例

下例中的功能块 FB0 为使用 ST 语言所撰写的程序。当程序实际运作时，在功能块执行完上半段的程序后便会先判断 X0.0 的状态，若 X0.0 为 ON，则此功能块的动作便会立即结束；但若 X0.0 为 OFF，则该功能块便会执行完整段的程序后才会离开。



13.2.11 EXIT 叙述

- 格式

```
EXIT ;
```

- 功能说明

此叙述句只可用于循环结构的内部，包括 **REPEAT**、**WHILE** 及 **FOR**，其功能则为强制脱离循环。

- 范例

EX1：下例为一个 **REPEAT** 的循环结构，而离开的条件除了待 DT1 持续递减至 0 之外，循环中又加入了一个 **IF** 结构搭配 **EXIT** 的叙述句，且让 D0 由 0 开始每次递增 1，而当 D0 递增至 100 时，程序便会直接离开循环，所以该循环最多只能执行 100 次；在下列的范例中，因 DT1 的设定值为 110，因此当 DT1 的值递减至 10 的时候，循环便会因为已执行了 100 次而直接结束，所以最后 DT1 的值便会等于 10，而不会递减至 0。

```
0001 DT1 := 110 ;
0002 D0 := 0 ;
0003
0004 REPEAT
0005
0006     DT1 := DT1 - 1 ;
0007     D0 := D0 + 1 ;
0008
0009     IF D0=100 THEN
0010         EXIT ;
0011     END_IF ;
0012
0013 UNTIL DT1 = 0
0014 END_REPEAT ;
```

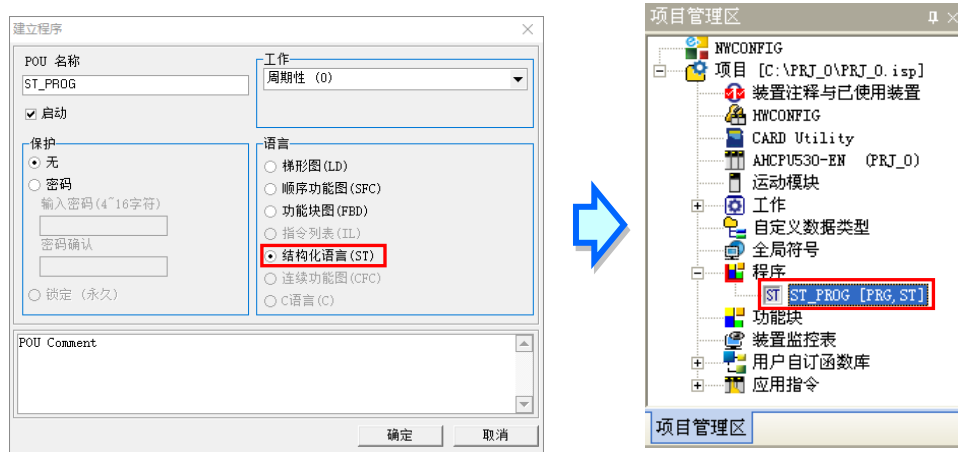
EX2：下例为一个 FOR 循环，原本是进行 DT1 内容值（初始值为 10）的阶层运算，并将最后的运算结果放置在装置 D0 当中，但其中加入了一个 IF 结构搭配 EXIT 叙述句后，只要 D0 大于 100 时，程序便会立即停止计算并强制离开循环；因此在下列的范例中，当 D10 递增至 5 时，D0 的值便会是 $1*2*3*4*5=120$ ，而循环则也会因此提前结束。

```
0001 DT1 := 10 ;
0002 D0 := 1 ;
0003
0004 FOR D10 := 1 TO DT BY 1 DO
0005
0006     D0 := D0 * D10 ;
0007
0008     IF D0>100 THEN
0009         EXIT ;
0010     END_IF ;
0011
0012 END_FOR ;
```

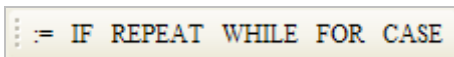
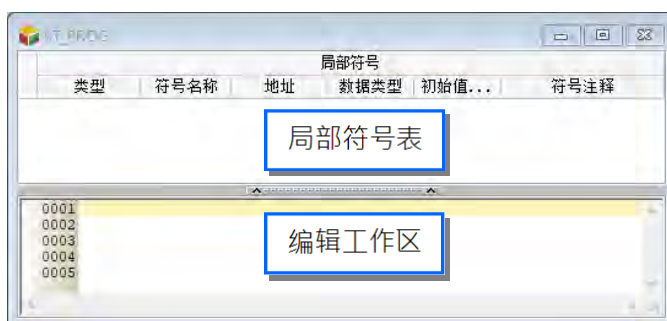
13.3 ISPSOft 中建立 ST 程序

13.3.1 ISPSOft 的 ST 编辑环境

建立 POU 时，于语言字段选择「结构化语言 (ST)」便可建立 ST 程序，请参考第 5.4.1 节的说明。



下图即为 ISPSOft 中的 ST 编辑环境，其窗口配置的上方为区域符号表，而下方则是编辑工作区，且在编辑工作区的左侧默认会有行号显示，用户可自行设定其显示的格式，详细操作请参考第 2.3.1 节的相关说明。在进入 ST 的编辑环境后，窗口中亦会出现对应的图标工具栏，其功能如下。



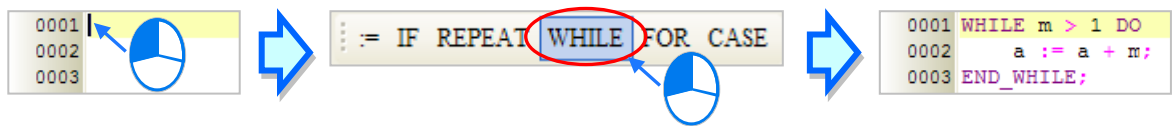
图示	功能说明
	插入指定结构 := 的范本
	插入条件结构 IF 的样板
	插入循环结构 REPEAT 的样板
	插入循环结构 WHILE 的样板
	插入循环结构 FOR 的样板
	插入条件结构 CASE 的样板

13.3.2 编辑 ST 叙述

在 ST 的编辑工作区中，其操作方式与一般的文本编辑器类似，编辑时只要依据 ST 语法的格式直接使用键盘在工作区上面输入或修改文字即可，且当按下键盘的【Enter】键时便可执行换行的动作。

```
0001 MO := TRUE ;
0002
0003
```

除此之外，亦可先单击欲编辑的位置后，直接透过图标工具栏来插入结构的样板并进行修改即可。



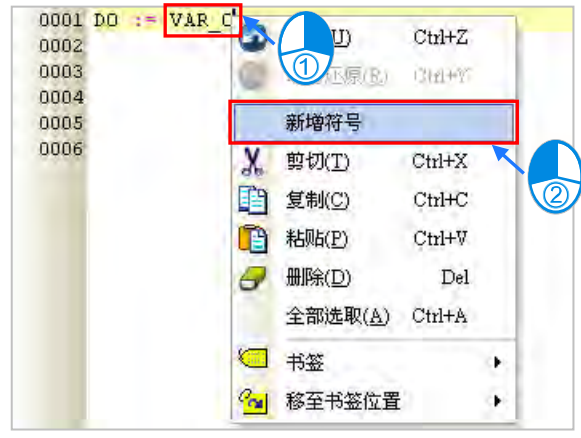
或者直接输入结构的第一个英文字母，接着单击自动带出的项目来插入结构，键盘的光标也会直接移动到输入条件位置，用户可更快速的编辑 ST。



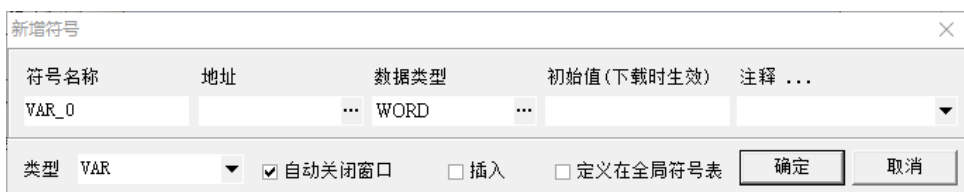
选取文字的方法也与一般的文本编辑器相同。先将鼠标点在选取区块的起点并按住鼠标左键，之后往选取的方向移动便可框选文字；又或者是先用鼠标单击在某个区块的起点，接着按住键盘的【Shift】键，之后再鼠标单击该区块的终点便可将整段区块选取起来。

```
0001 MO := FALSE ;
0002 MO := FALSE ;
0003 DO := 0 ;
0004 D1 := 0 ;
0005
```

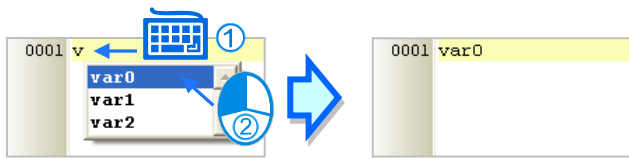
当在编辑的过程中输入了未经宣告的变量符号时，输入完毕后用鼠标单击该文字并按下右键，之后再于快捷选单中单击「新增符号」便可自动导出变量符号的宣告窗口。



13



程序输入的过程中，会自动从以建立的变量符号中筛选出符合的变量符号供用户挑选，可利用键盘的「↑」、「↓」键，或直接在选单中点击鼠标左键完成输入。



13.3.3 插入 API 应用指令与功能块

插入 API 应用指令与功能块的方法有下述两种。

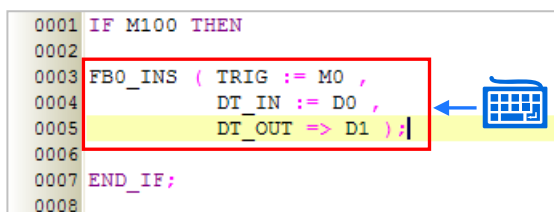
● 方法一

依据应用指令与功能块的格式，直接于编辑工作区中利用键盘输入即可。

```

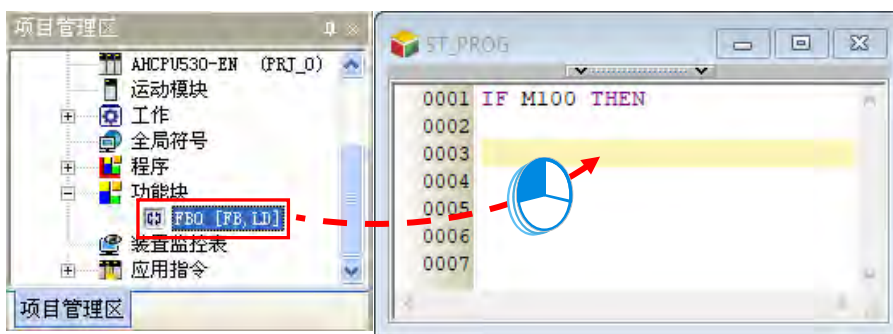
0001 IF M100 THEN
0002
0003 FBO_INS ( TRIG := M0 ,
0004           DT_IN := D0 ,
0005           DT_OUT => D1 );
0006
0007 END_IF;
0008

```



● 方法二

于项目管理区中展开应用指令或功能块的目录，在其中找到欲插入的项目后，于该项目上按住鼠标左键，之后再将其拖曳至欲插入的位置后放开左键即可。




插入应用指令或功能块之后，请再依据其定义的格式，直接于插入的样板中编辑对应的操作数。

```

0002
0003 ????? (
0004     TRIG := ?????,
0005     DT_IN := ?????,
0006     DT_OUT => ?????
0007 );
0008

```



```

0002
0003 FBO_INS (
0004     TRIG := M0 ,
0005     DT_IN := D0 ,
0006     DT_OUT => D1
0007 );
0008

```

拖曳功能块时须注意只能拖曳功能块的定义，而不能拖曳其下层的功能块实例项目。



13.3.4 书签功能

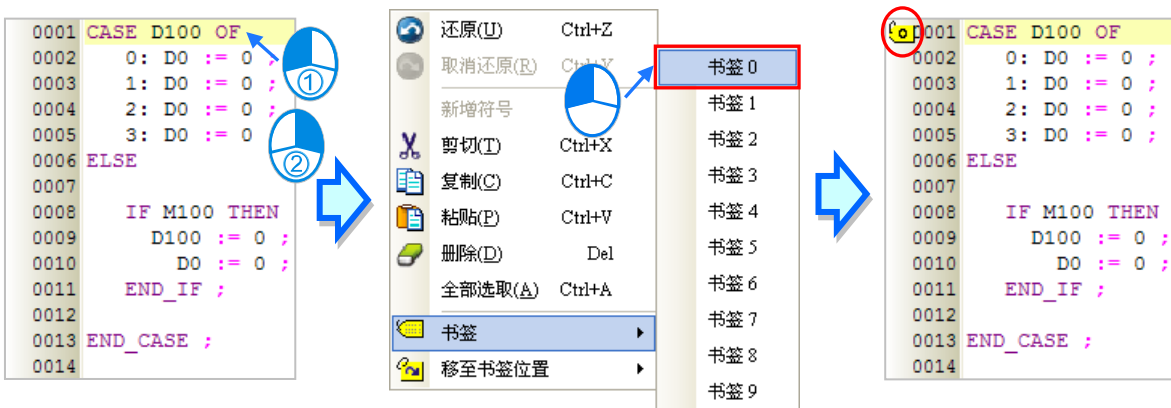
书签功能为方便搜寻或移动至特定编辑位置的便利工具。

● 加入/移除书签

先用鼠标单击欲加入书签的行列后按下鼠标右键，接着于快捷选单中单击「**书签**」项目，之后再于该项目中选择欲加入的书签编号即可。而要移除书签时，同样的请先用鼠标单击欲取消书签的行列后按下鼠标右键，接着于快捷选单中单击「**书签**」项目，之后再于该项目中单击与所选行列相同编号的书签即可；不过若选择的是其它编号时，所选行列的书签便会被变更为该编号。若于功能工具栏中点击**编辑 (E) > 书签 > 移除所有书签 (R)**，则目前编辑窗口的中的所有书签便会被全部取消。

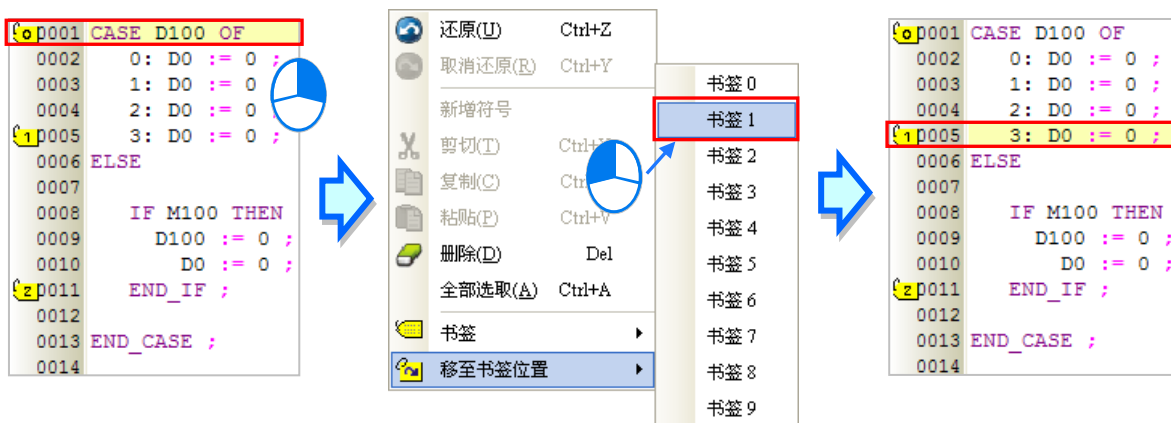
操作时若单击了一个已使用于其他行列的编号，则该编号的书签便会被搬移至目前选取的行列上。

13



● 移至书签位置

编辑工作区中按下鼠标右键，接着于快捷选单中单击「**移至书签位置**」项目，之后在目录中选择移动目标的书签编号即可。



13.4 ST 的程序范例

13.4.1 范例说明

此范例为自动控制应用中经常会使用到的塔灯控制，而此处选用的则是三色灯号加上蜂鸣器的配置。下表即为各种状态下的控制机制，其中当有一个以上的状态成立时，系统将以优先权较高的为主。



状态代码	设备状态	灯号显示				优先
		红灯	黄灯	绿灯	蜂鸣器	
4	设备异常	ON	OFF	OFF	Flash	最高 ↑ 最低
3	设备异常 (关闭蜂鸣器)	Flash	OFF	OFF	OFF	
2	运转中	OFF	OFF	ON	OFF	
1	工作完成	OFF	Flash	OFF	OFF	
0	闲置	OFF	ON	OFF	OFF	最低

13

13.4.2 硬件规划

在本范例中，我们将选用 AH 系列主机 **AHCPU530-EN**，数字 IO 模块则选用 8 点输入/8 点输出的 **AH16AP11R-5A**，而主背板则选用 4 槽的 **AHBP04M1-5A**。下表即为本范例的 IO 配置表。

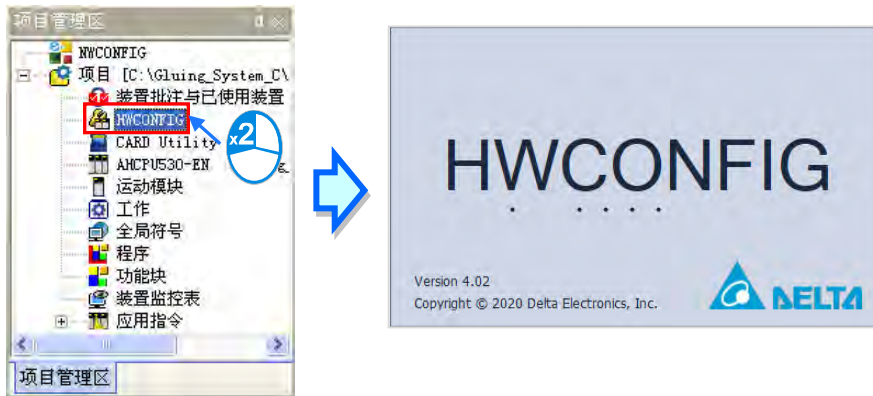
Type	ID	说明
DO	Y0.0	红灯输出
DO	Y0.1	黄灯输出
DO	Y0.2	绿灯输出
DO	Y0.3	蜂鸣器输出

13.4.3 程序规划

- (1) 当标志 **ERROR** 为 **ON**，且标志 **BUZZER_OFF** 为 **OFF** 时，将状态代码 **SC** 设为 4。
- (2) 当标志 **ERROR** 为 **ON**，而标志 **BUZZER_OFF** 为 **ON** 时，将状态代码 **SC** 设为 3。
- (3) 当标志 **ERROR** 为 **OFF**，且标志 **RUNNING** 为 **ON** 时，将状态代码 **SC** 设为 2。
- (4) 当标志 **ERROR** 与 **RUNNING** 都为 **OFF**，但标志 **COMPLETE** 为 **ON** 时，将状态代码 **SC** 设为 1。
- (5) 当标志 **ERROR**、**RUNNING** 与 **COMPLETE** 都为 **OFF** 时，将状态代码 **SC** 设为 0。
- (6) 依据状态代码 **SC** 切换对应的灯号显示状态。而因灯号控制在许多的应用中都会使用，且灯号的显示可以有许许多多变化，因此考虑程序的扩充性与可再用性，我们会将此段程序规划为功能块。

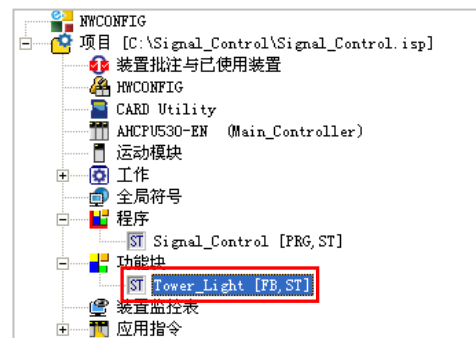
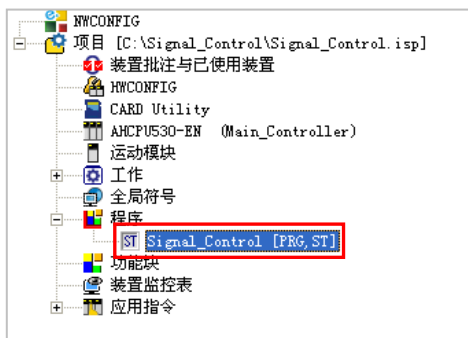
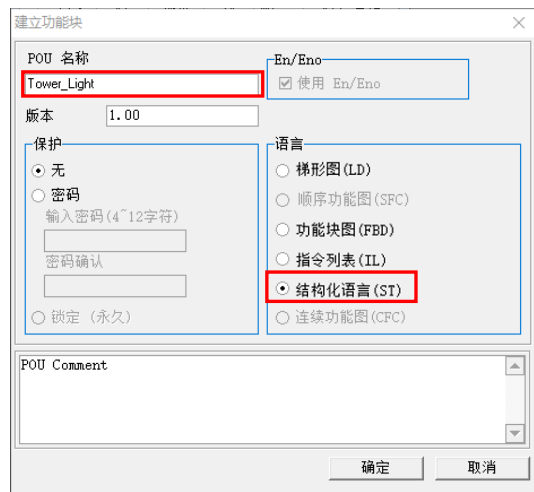
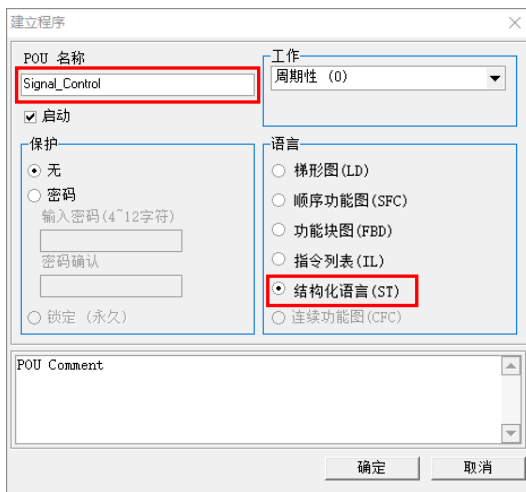
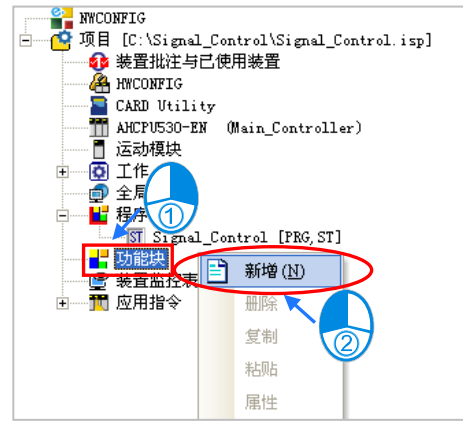
13.4.4 建立范例程序

首先请开启 ISPSOft 并建立一新项目，本范例的硬件规划与第 4 章相同，因此请参考第 4 章的内容，并自行完成硬件规划的工作。

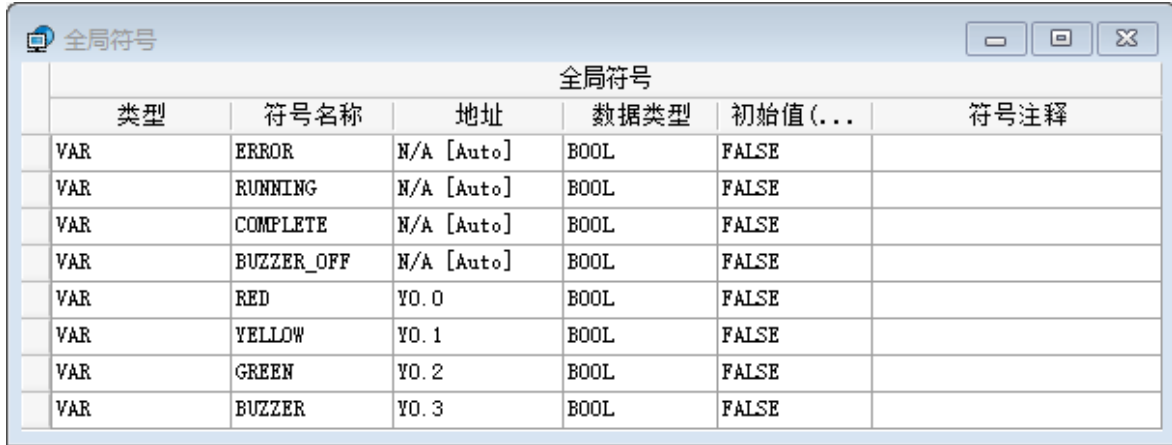


13

接下来请直接于项目管理区中分别建立主程序与功能块的 POU，且在建立时请都选择使用 ST 语言。



然后将已知的系统标志及 IO 配置建立在全局符号表中。



全局符号					
类型	符号名称	地址	数据类型	初始值(...)	符号注释
VAR	ERROR	N/A [Auto]	BOOL	FALSE	
VAR	RUNNING	N/A [Auto]	BOOL	FALSE	
VAR	COMPLETE	N/A [Auto]	BOOL	FALSE	
VAR	BUZZER_OFF	N/A [Auto]	BOOL	FALSE	
VAR	RED	YO.0	BOOL	FALSE	
VAR	YELLOW	YO.1	BOOL	FALSE	
VAR	GREEN	YO.2	BOOL	FALSE	
VAR	BUZZER	YO.3	BOOL	FALSE	

13

接着再编辑功能块 **POU** 的程序，而在规划上，我们须要建立一个输入端点用以接收主程序所指定的状态代码，且针对灯号与蜂鸣器的部份，我们亦将其状态的输出结果建立为功能块的输出端点。

因功能块的功用在接受主程序的状态代码，并在经过判断之后将对应的灯号状态输出，所以相当适合使用 **CASE** 的结构来设计；而当后续有不同的灯号显示模式时，我们只须要新增一个状态代码，接着再于此功能块的程序中插入一个 **CASE** 的条件即可，因此对于后续的扩充将会有相当大的弹性。编辑时可于编辑工作区中插入 **CASE** 结构的样板后再加以修改，或是直接输入下列的 **ST** 程序。

The screenshot shows the Tower_Light software interface. At the top, there is a window title bar with the name 'Tower_Light' and standard window controls. Below the title bar is a table titled '局部符号' (Local Symbols). The table has six columns: '类型' (Type), '符号名称' (Symbol Name), '地址' (Address), '数据类型' (Data Type), '初始值...' (Initial Value...), and '符号注释' (Symbol Comment). The table contains five rows of data:

类型	符号名称	地址	数据类型	初始值...	符号注释
VAR_INPUT	MODE	N/A [Auto]	INT	N/A	
VAR_OUTPUT	RL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	YL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	GL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	BR	N/A [Auto]	BOOL	FALSE	

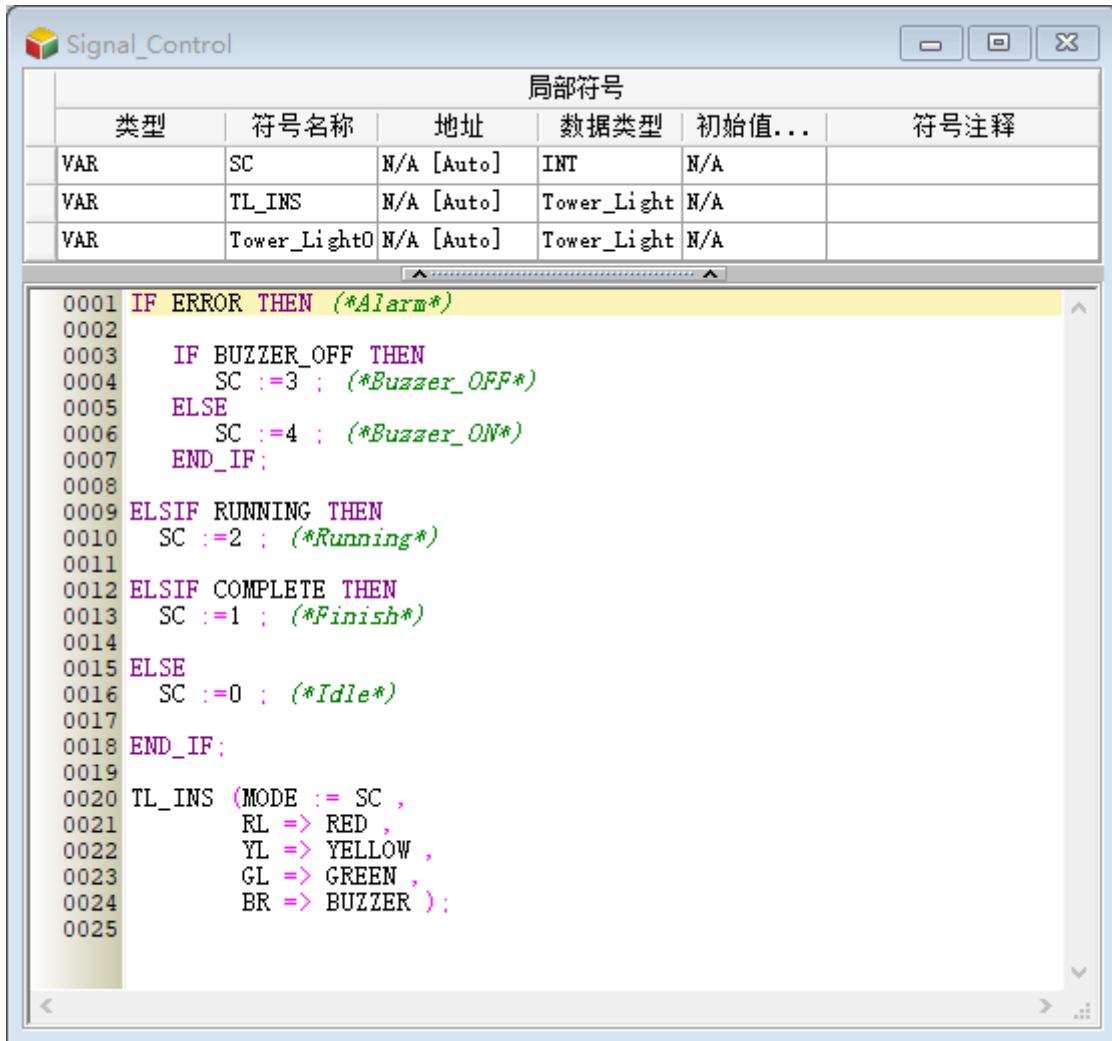
Below the table is a text editor showing a ladder logic program. The program starts with a comment: '(* SM407: 0.5S on/ 0.5S off *)'. The main logic is a CASE statement based on the MODE variable, with four cases (0, 1, 2, 3, 4) defining the states of RL, YL, GL, and BR. The program ends with 'END_CASE;'.

```

0001 CASE MODE OF
0002
0003 (* SM407: 0.5S on/ 0.5S off *)
0004
0005     0: RL :=FALSE ;
0006         YL :=TRUE ;
0007         GL :=FALSE ;
0008         BR :=FALSE ;
0009
0010     1: RL :=FALSE ;
0011         YL :=SM407 ;
0012         GL :=FALSE ;
0013         BR :=FALSE ;
0014
0015     2: RL :=FALSE ;
0016         YL :=FALSE ;
0017         GL :=TRUE ;
0018         BR :=FALSE ;
0019
0020     3: RL :=SM407 ;
0021         YL :=FALSE ;
0022         GL :=FALSE ;
0023         BR :=FALSE ;
0024
0025     4: RL :=TRUE ;
0026         YL :=FALSE ;
0027         GL :=FALSE ;
0028         BR :=SM407 ;
0029
0030 END_CASE ;

```

接下来便可继续编辑主程序的部份，其中我们必须先在主程序的区域符号表中宣告代表状态代码的变量符号 **SC** 以及功能块实例 **TL_INS**；而在程序的规划上，因为在灯号的控制中有优先权的机制，因此此程序较适合使用 **IF** 的结构来进行设计；在确定状态代码之后，我们还须调用方才所建立的功能块，并将状态代码提供给功能块进行运算，而最后所得到的结果便可再指定给实际输出的 **IO** 接点。最后完成的主程序如下。



完成以上的编辑后，接着请自行完成编译，并将硬件参数与程序下载至主机当中进行后续测试。

第14章 顺序功能图 SFC

目录

14.1	认识顺序功能图 SFC	14-3
14.1.1	SFC 程序架构	14-3
14.1.2	SFC 动作原理	14-4
14.2	ISPSOft 中的 SFC 架构	14-5
14.2.1	步进点与 Action	14-5
14.2.2	转换点 (Transition)	14-8
14.2.3	共进分支与选择分支	14-10
14.2.4	并行汇合与选择汇合	14-11
14.2.5	跳跃点	14-12
14.2.6	Actions 的修饰条件	14-14
14.2.6.1	条件定义	14-15
14.2.6.2	设置修饰条件的注意事项	14-19
14.2.7	初始步进点	14-20
14.2.8	内部属性	14-21
14.2.8.1	Step 属性参数	14-21
14.2.8.2	Action 与 Transition 属性参数	14-25
14.3	在 ISPSOft 中建立 SFC 程序	14-32
14.3.1	ISPSOft 的 SFC 编辑环境	14-32
14.3.2	Actions 与 Transitions 的建立与管理	14-34
14.3.3	新增步进点结构	14-36
14.3.4	插入并联转换点	14-37
14.3.5	插入并联步进点	14-38
14.3.6	插入步进点结构 - 共进分支&选择汇合	14-39
14.3.7	插入步进点结构 - 选择分支&并行汇合	14-40
14.3.8	插入跳跃点	14-41
14.3.9	指定步进点与转换点符号	14-42
14.3.10	指定初始步进点	14-42
14.3.11	配置 Actions/折叠 Action 列表	14-43
14.4	SFC 的程序范例	14-46

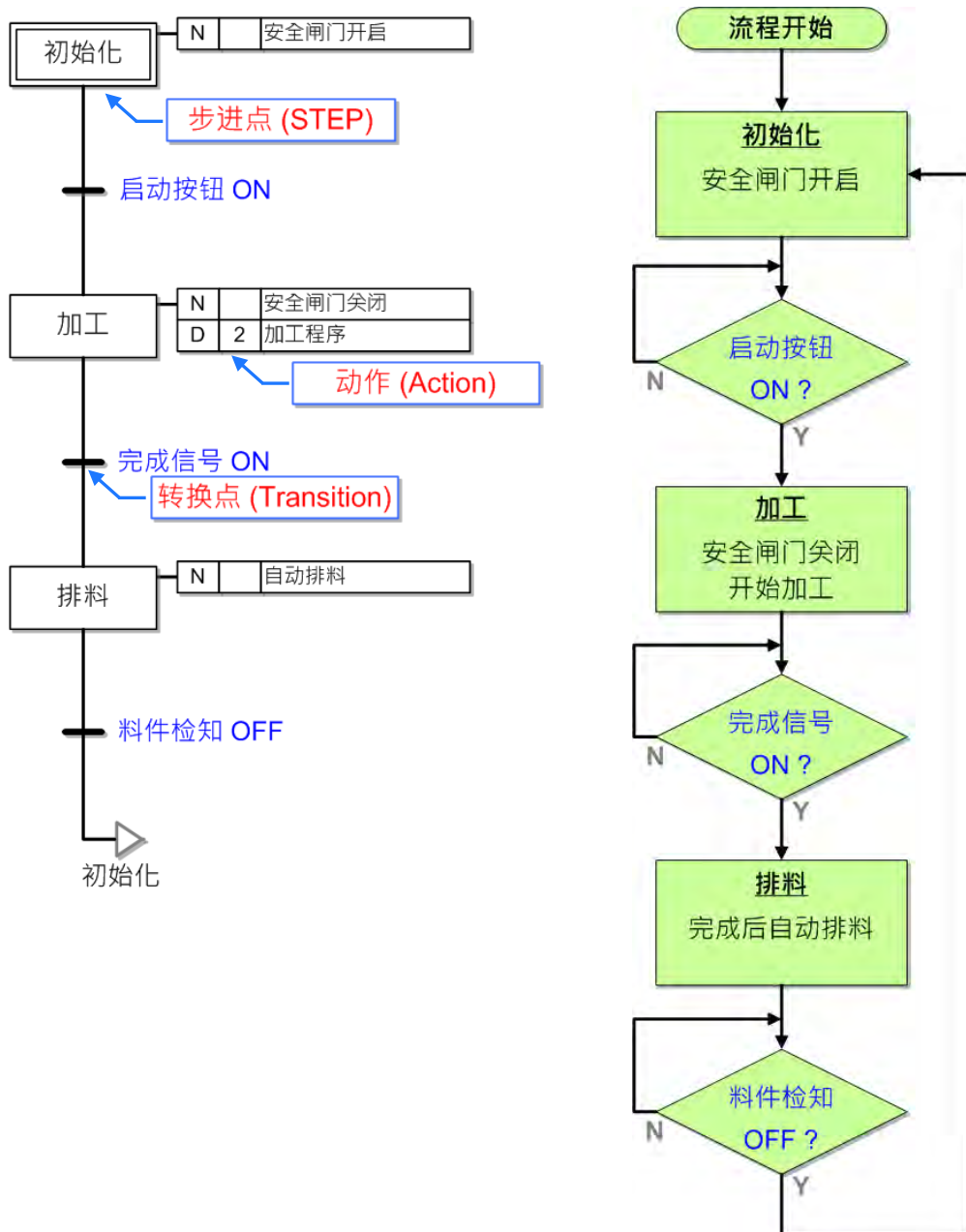
14.4.1	范例说明.....	14-46
14.4.2	硬件规划.....	14-47
14.4.3	程序规划.....	14-47
14.4.4	建立范例程序.....	14-48

14.1 认识顺序功能图 SFC

14.1.1 SFC 程序架构

顺序功能图 (**Sequential Function Chart, SFC**) 是 IEC 61131-3 所规范的 PLC 编程语言之一，其最大的特色在于以类似流程图的观念来规划程序，适用于着重流程控制及状态转移的控制程序。

在下图中，左侧为基本的 SFC 图形，而右侧则为其对应的程序流程图。由左侧的结构可看出 SFC 程序是由许多的步进点 (**STEP**)、转换点 (**Transition**) 以及每一个步进点当中的细部动作 (**Action**) 所构筑而成，其中步进点的功能如同流程图的执行程序，而转换点则如同流程图中的条件判断。

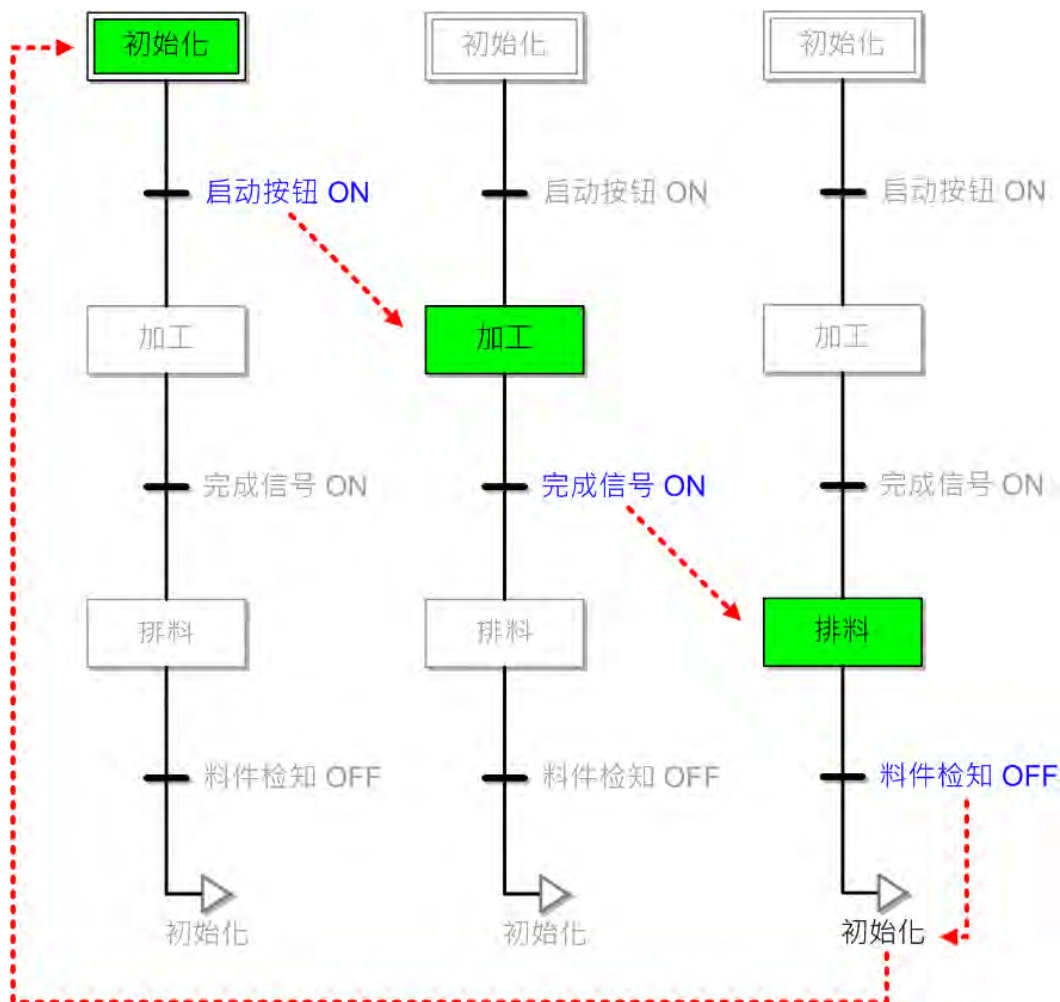


14.1.2 SFC 动作原理

SFC 程序的执行方式就如同接力赛一般，一个接着一个地传递执行权，当有某个步进点成为启动状态时，代表该步进点取得执行权，而系统便会执行该步进点所定义的程序，直到该步进点下方的转换点条件成立后，程序才会将执行权移交给下一个步进点，并将目前执行的步进点关闭，像这样执行权转移的程序也叫做**步进点转移**；而当依照这样的规则重复执行，直到最下方的转换点条件成立时，系统便会再将执行权转移至最后指定跳跃的步进点来继续整个程序的循环。

在执行过程中，系统只会关注拥有执行权的步进点，并执行其内部定义的动作程序，且也只会确认该步进点下方的转换条件是否成立；换句话说，对于未获得执行权的步进点而言，其下方的转换条件并不会影响程序的执行步序，而系统也不会主动执行这些步进点所指定的动作程序。

在 ISPSOft 的 SFC 架构中，步进点与转换点必须是交错排列的，因此在编绘 SFC 的图形时，连续两个**步进点**或连续两个**转换点**的结构将无法通过编译。



上例是单纯应用的 SFC 架构，而在 ISPSOft 中尚提供许多丰富且灵活的编程功能，包括分支与汇合的结构、模块化的动作（Action）与转换点（Transition）程序、可定义动作（Action）执行方式的修饰条件...等；而这些功能亦将在后续的章节中逐一介绍，并说明如何在 ISPSOft 当中进行实作。

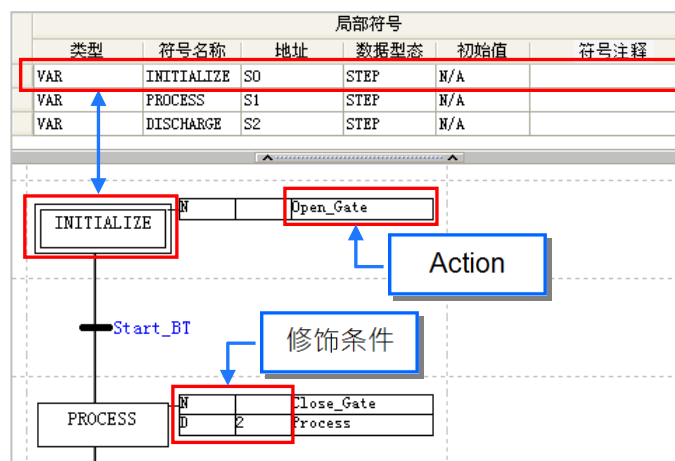
14.2 ISPSOft 中的 SFC 架构

若用户使用 SFC 编程，SFC 编程架构内容可能会随机种而不同，接下来本章节将对 SFC 架构做介绍。

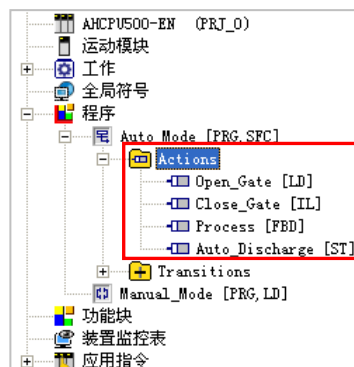
14.2.1 步进点与 Action

在 ISPSOft 中，每个步进点都必须配置一个 **STEP** 型态的变量符号来做为其启动与否的状态标志，而当对应的标志为 ON 时，便代表该步进点已进入启动状态，亦即取得执行权。AH5x0 机种及 DVP 机种每个 **STEP** 符号都将占用一个 PLC 内部的 S 装置，至于 S 装置的使用限制与相关说明，详细内容请参考各机种的操作手册。

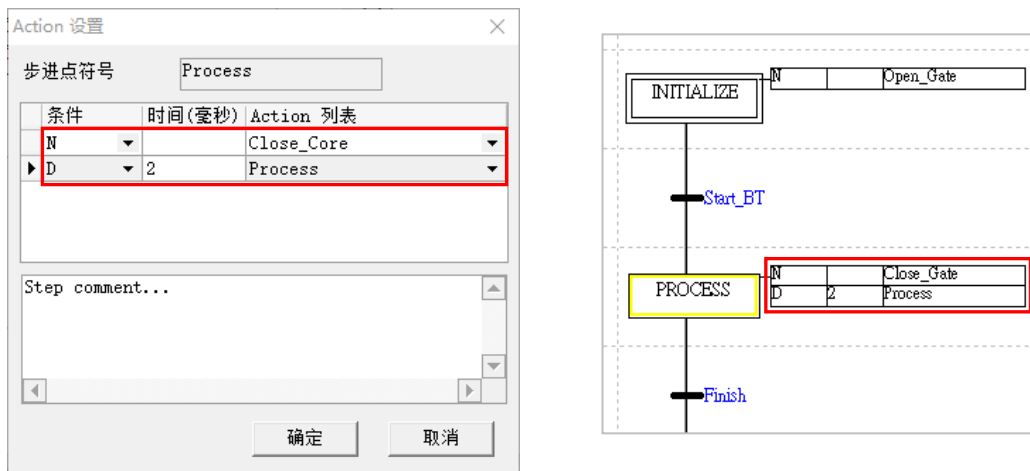
当步进点被启动后，其内部所指定的动作即会依据所定义的方式开始执行，直到该步进点将执行权移出去之后，所有对应的动作便会视其执行方式的定义来决定停止与否；而在 ISPSOft 中，这些动作程序亦可如同子程序一般的被模块化，并可重复在不同的步进点当中被指定，且同一个步进点亦可指定一个以上的动作模块，而这些动作模块在 ISPSOft 当中便称之为 **Action**；此外，每个 Action 皆可单独被定义其执行方式，至于决定 Action 执行方式的定义，在 ISPSOft 当中便称之为**修饰条件**。



在 ISPSOft 中，Action 本身也是一段程序代码，且该程序代码亦可用梯形图 (LD)、指令列表 (IL)、功能块图 (FBD) 或是结构化语言 (ST) 来进行开发，而所建立的 Action 程序亦会被列于项目管理区中；此外，Action 的编程窗口与各个语言的编程环境相同，唯一差异只有 Action 的编程窗口并没有区域符号表，这是因为在 ISPSOft 中，同一个 SFC 程序的 Action 与 Transition 会共享同一份区域符号表。



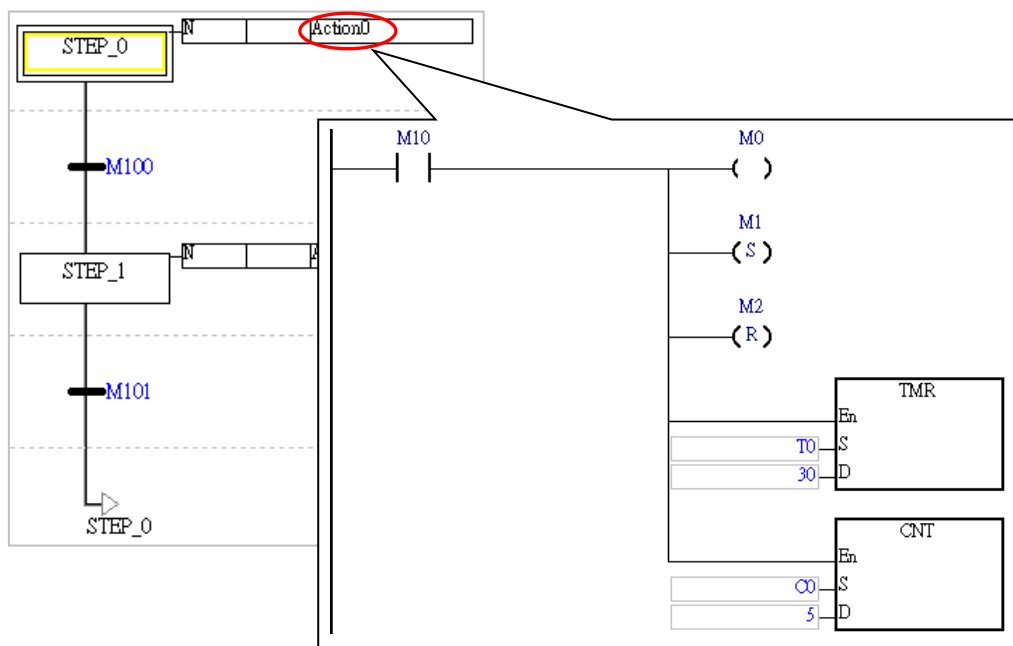
在编辑 SFC 的程序时，一般会先建立好 Action 的程序之后，再将其配置给各个步进点，并逐一定义每个 Action 的执行方式（修饰条件）；下图即为 Action 的设置画面与设置完成后的 Action 列表。



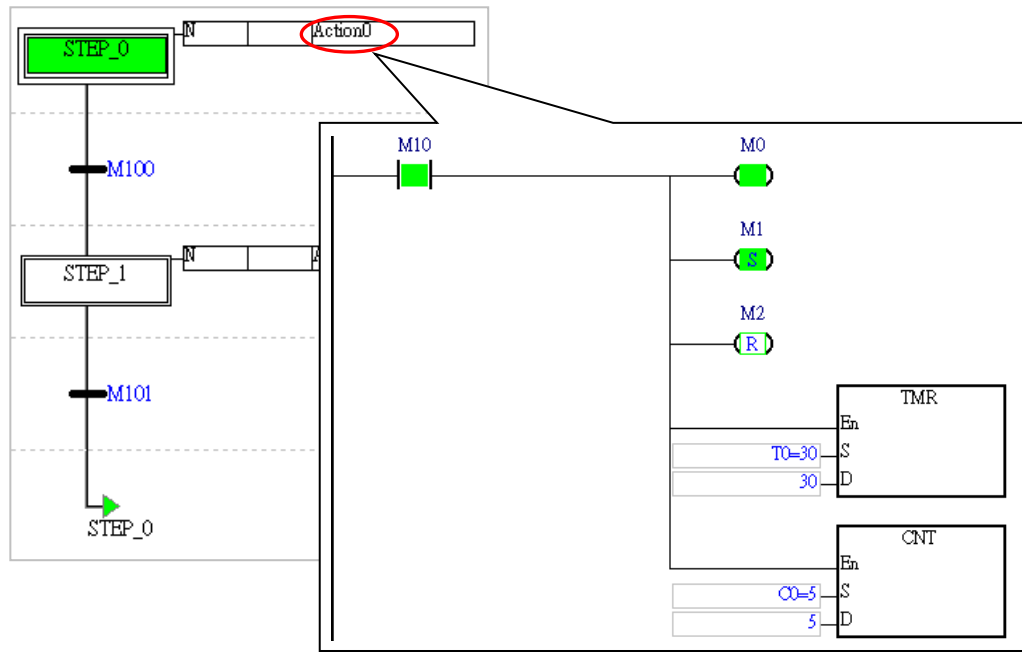
14

在 SFC 的执行过程中有一个相当重要的程序，称之为「Final Scan」，该程序一般是发生在 Action 停止执行时，系统会再自动执行一个程序以关闭该 Action 当中的所有输出动作；当执行「Final Scan」时，原本 Action 当中透过 OUT 指令驱动的线圈全部会关闭而呈现 OFF 状态，而所有于该 Action 当中执行的应用指令与功能块也不再执行，包括定时器也将被重置；不过若是透过 SET 或 RESET 指令而驱动的线圈，则因该指令的特性，所以仍会保持原状，且计数器也只会停止计数，而其接点的状态与计数值则同样会维持不变，而不会如同定时器一般被重置。

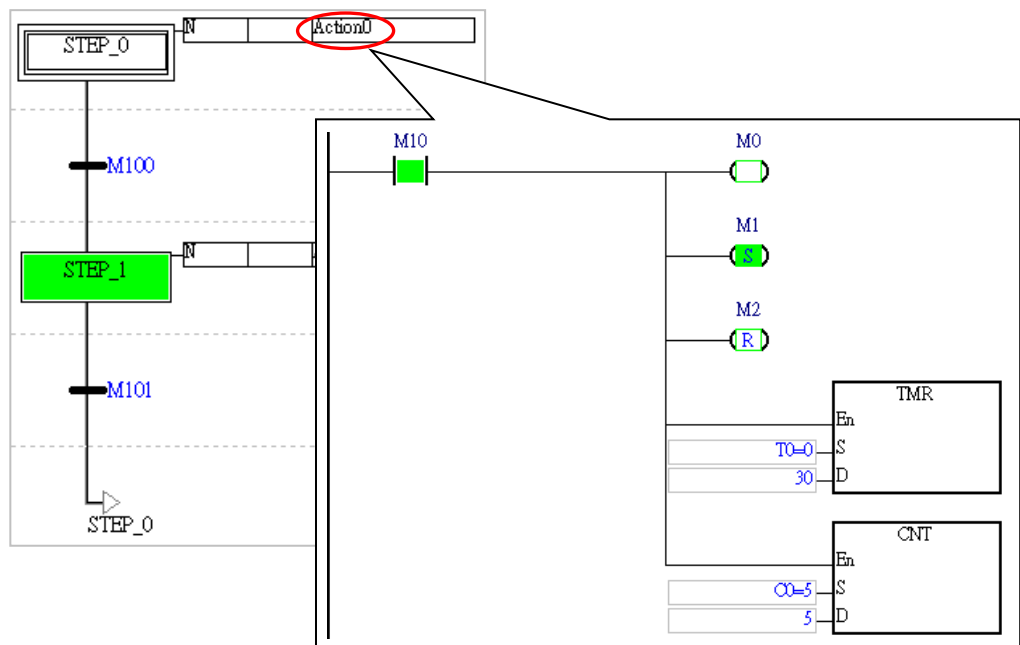
以下图为例，其中 Action0 为配置于 STEP_0 的 Action，而其修饰条件则为 N (Normal)，即一般的执行条件，因此当执行权由 STEP_0 转移至 STEP_1 时，系统便会再执行一个 Final Scan 的程序。



下图为步进点 STEP_0 为启动状态的监控画面，其中 Action0 的 M10 处于 ON 的状态，因此输出 M0 的状态也会为 ON，而 M1 则被设置 (SET) 为 ON，M2 也被重置 (RST) 为 OFF，至于定时器 T0 与计数器 C0 则都处于计时/计数到达的状态。

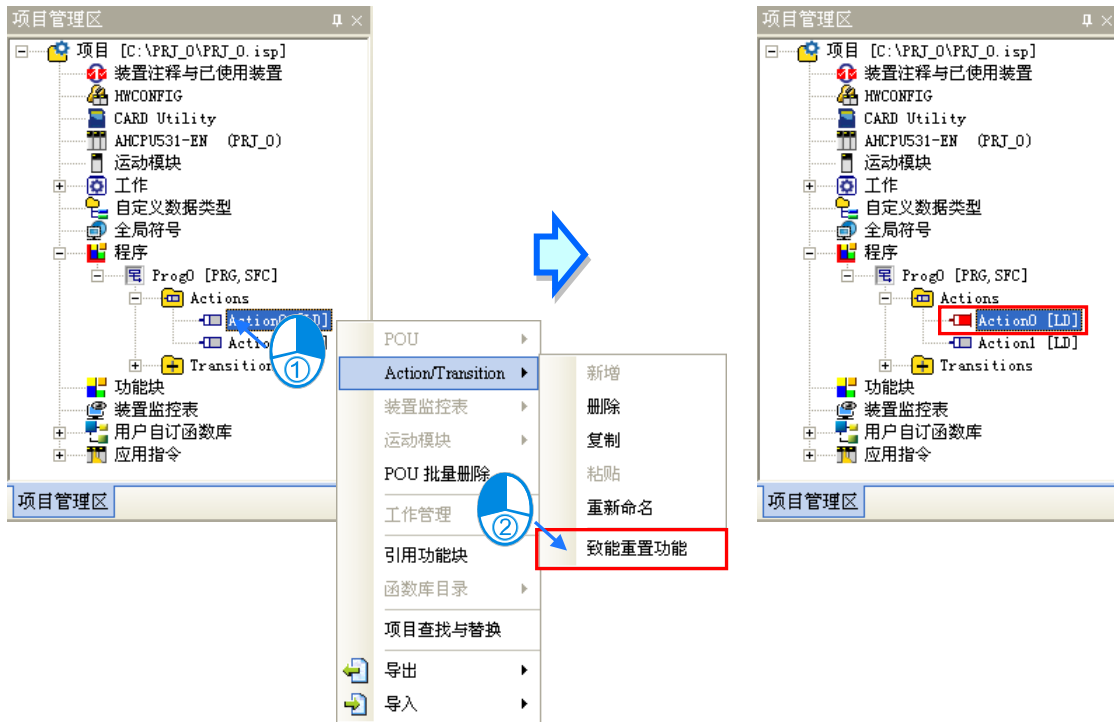


下图则为步进点转移至 STEP_1 后的监控画面，其中可发现，即使 M10 仍保持为 ON，输出的 M0 同样会被关闭，且定时器 T0 也会被重置；不过 M1、M2 与计数器 C0 则还是会保持在原本的状态。



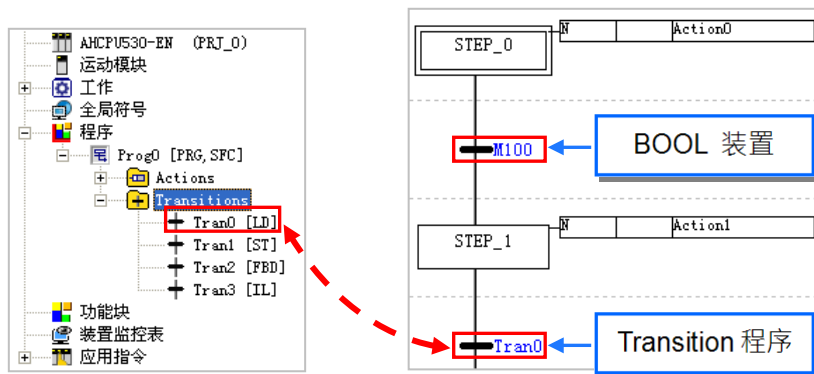
AH5x1、AH560 冗余系列机种、AHxxEMC 与 AS 系列机种可设置每一个 Action 是否执行「Final Scan」，ISPSOft 为预设执行上述的「Final Scan」，欲取消时，在不执行「Final Scan」的 Action 上按下鼠标右键，将 **Action/Transition > 致能重置功能** 取消选择，则此 Action 上的图示将以红色表示不执行「Final Scan」，因此，前述范例中的 STEP_0 在步进点转移后，Action0 中的状态仍将为转移前的状态。

14

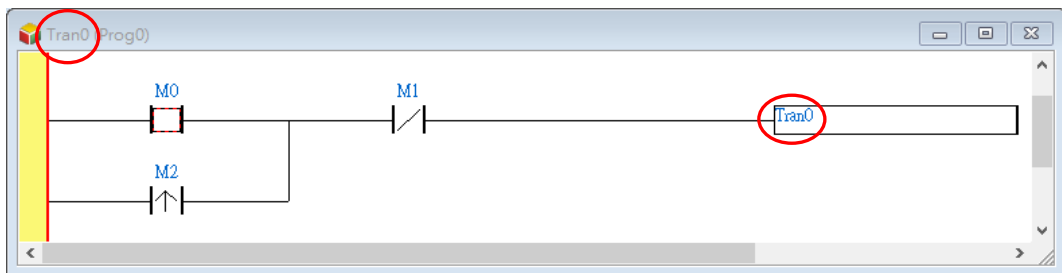


14.2.2 转换点 (Transition)

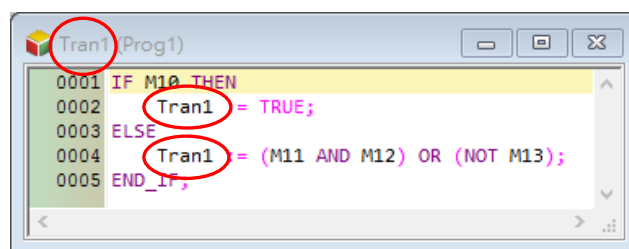
转换点是决定 SFC 执行时序的重点指标，它会决定目前启动中的步进点是否该将执行权移转到下一个步进点，且如同前述，只有位于目前启动中的步进点，其下方的转换点才能决定执行权的移转。在 ISPSOft 中，转换点可以是某个布尔格式的装置或变量，同时也可以具备逻辑运算的程序代码；若转换点是一段逻辑运算的输出，则在编辑时便可先将这段运算程序建立为 Transition 程序，之后再将其指定于对应的转换点即可，且所建立好的 Transition 程序也同样会被列于项目管理区中。



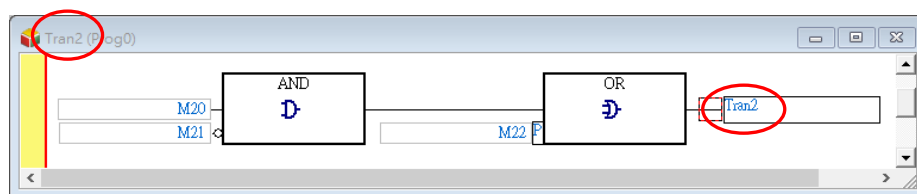
支持 Transition 程序的语言包括 LD、IL、FBD 与 ST，且虽然编辑的规则不同，其共通点则都是必须将某个布尔状态输出至与该 Transition 程序同名的符号，而该符号亦无须另外宣告；不过若在符号表中有宣告与某个 Transition 程序同名的变量符号时，其编译结果便会产生错误。除此之外，在 Transition 的程序中无法使用 API 应用指令与功能块，但可以使用比较接点与区块逻辑指令（NP、PN、INV）。当使用 LD 来开发 Transition 程序时，其程序代码只能有一个梯形图区段，而最后输出的接点也只能是与该 Transition 程序同名的符号；除此之外，在 LD 的 Transition 程序当中不得有多重输出。



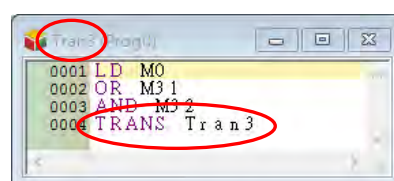
当使用 ST 来开发 Transition 程序时，其程序代码并没有行数的限制，只要在程序代码中有将某个布尔状态指定给与该 Transition 程序同名的符号即可。



当使用 FBD 来开发 Transition 程序时，其程序代码也只可有一个功能块图区段，而最后的输出节点也必须是与该 Transition 程序同名的符号，且不得有多重输出。



当使用 IL 来开发 Transition 程序时，其程序代码并没有行数的限制，但无法使用 OUT、SET、RST 与 API 应用指令，亦即程序中只可建立逻辑运算的部份，且最后输出的指令亦必须固定为「TRANS」，并搭配与该 Transition 程序同名的符号做为其操作数。



当建立一个 Transition 程序时，系统便会预先将与该 Transition 程序同名的符号加入程序代码中，用户仅需完成条件运算的部份即可；而因为同一个 SFC 程序的 Action 与 Transition 会共享同一份区域符号表，因此在 Transition 程序的编辑窗口中同样没有区域符号表。

由上述的规则可知，于 Transition 程序当中并无法建立过于复杂的程序，因此当转换条件必须经过复杂的运算时，建议可将该程序建立于步进点的 Action 程序当中，并于该 Action 程序中产生一个决定步进点转移的条件标志，而之后再将该标志指定于步进点下方的转换点位置即可。

虽然用户无须于符号表中宣告 Transition 程序的符号，但系统仍会为其配置一个内存的位置做为转换条件是否成立的判断标志，因此编辑 Transition 的程序时，必须确保该符号可以被指定一个确切的逻辑状态，否则便可能会产生执行上的错误。

以下图 ST 的 Transition 程序为例，在第一次执行的过程中，一旦 (M0 AND M1) 的结果为 ON 时，Tran4 也将被设为 ON，而此时条件便会成立并将执行权转移至下一个步进点；但当 SFC 的程序再度将执行权转移回此步进点时，下图的 Transition 程序便会再次运作，而此时若 (M0 AND M1) 的运算结果尚为 OFF，则下图的程序便不会执行「THEN」后面的叙述句，因此 Tran4 将无法被更新而保持在上次为 ON 的状态，而此时系统便会判断条件已成立，并立即将执行权转移至下一个步进点。

若以下图的程序而言，较佳的方式则是加上「ELSE」的描述，以确保 Transition 符号可以被指定一个确切的逻辑状态；而除了 IF 结构之外，使用 CASE 结构时亦必须留意相同的问题。

```
0001 IF (M0 AND M1) THEN
0002     Tran4 := TRUE;
0003 END_IF;
```

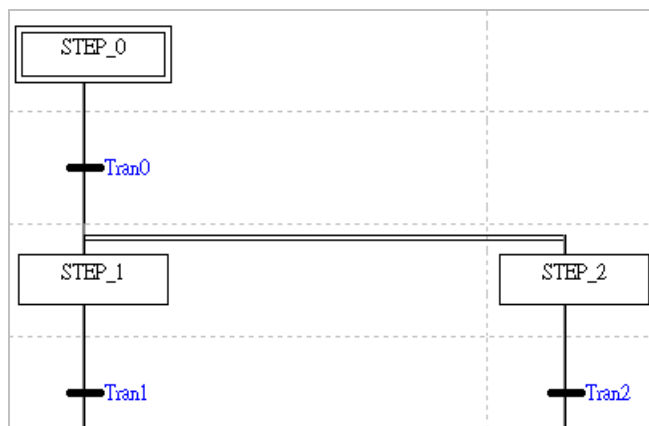
14.2.3 共进分支与选择分支

在 SFC 程序中可以有并行处理的程序，这样的结构我们便称之为分支，而分支结构依执行的机制可分为共进分支与选择分支。

● 共进分支

下图为共进分支的图例，其中分支的步进点会刻意用双线来进行连结；而共进分支的特征便是所有分支的步进点会共享同一个转换点，当该转换点的条件成立之后，下方的所有步进点就会同时被启动，亦即所有的平行路径都会执行。

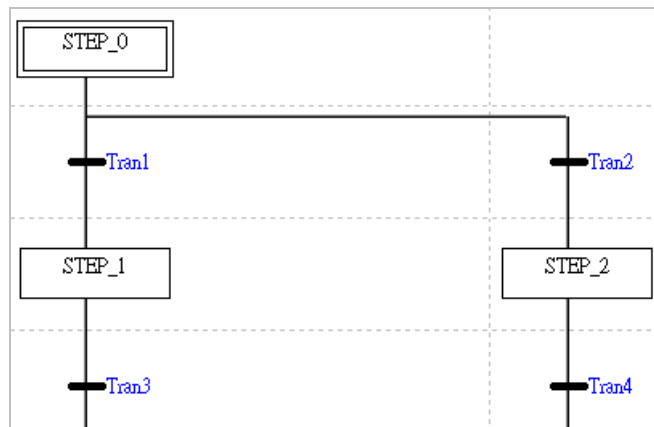
如下图，当转换点 Tran0 成立后，STEP_0 便会关闭，而 STEP_1 及 STEP_2 则会一起启动。



● 选择分支

下图为选择分支的图例，其特征为所有分支的步进点都会有各自的转换点，当其中某一个转换点的条件成立之后，执行权就会移转至该转换点下方的步进点，而其他的转换点则不再继续判断，因此在选择分支的平行路径中，同时间只有一条路径的步进点程序会被执行。因为转换点的判断顺序是由左至右，因此如果有多个转换点条件同时成立时，系统仍只会认定位于最左边的转换点。

以下图为例，当转换点 Tran2 的条件成立后，STEP_0 便会关闭并将 STEP_2 启动，而此时的 Tran1 则会停止判断并让 STEP_1 继续保持在不启动的状态。



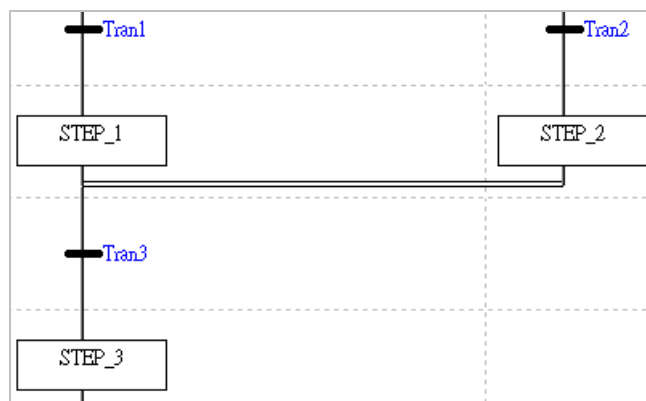
14

14.2.4 并行汇合与选择汇合

在 SFC 的分支结构中，最后仍须将所有分支的路径再重新汇整在一起，而这样的汇整程序便称之为汇合；如同分支结构一般，汇合依执行的机制亦可分为并行汇合与选择汇合。

● 并行汇合

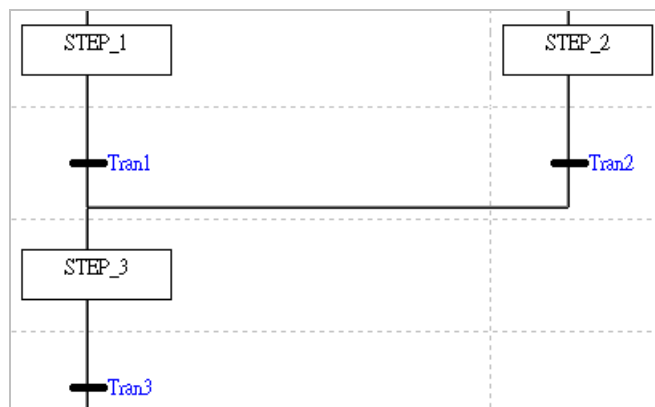
在图示中，并行汇合的步进点之间亦刻意用双线来链接，而其特征则为所有分支的步进点都会联结至同一个转换点，因此必须等待连接至该转换点上方的所有步进点都被启动后，转换点才可进行判断。如下图的范例，须先等待 STEP_1 及 STEP_2 都被启动后，转换点 Tran3 才会开始判断，而当 Tran3 的条件成立时，STEP_1 及 STEP_2 便会一起被关闭，并启动 STEP_3。



● 选择汇合

下图为选择汇合的图例，其特征为所有汇合的步进点都会有各自的转换点，而只要其中某一个转换点的条件成立之后，执行权就会先行移转至下一个的步进点；但若平行路径中仍有其他被启动的步进点时，则所有被启动的步进点仍会继续执行，一直到其汇合的转换点成立之后，便会再次启动汇合下方的步进点，因此选择汇合有可能造成汇合点下方的顺序流程同时有多个步进点被启动，使用上请务必确认是否会造成程序上的错乱。

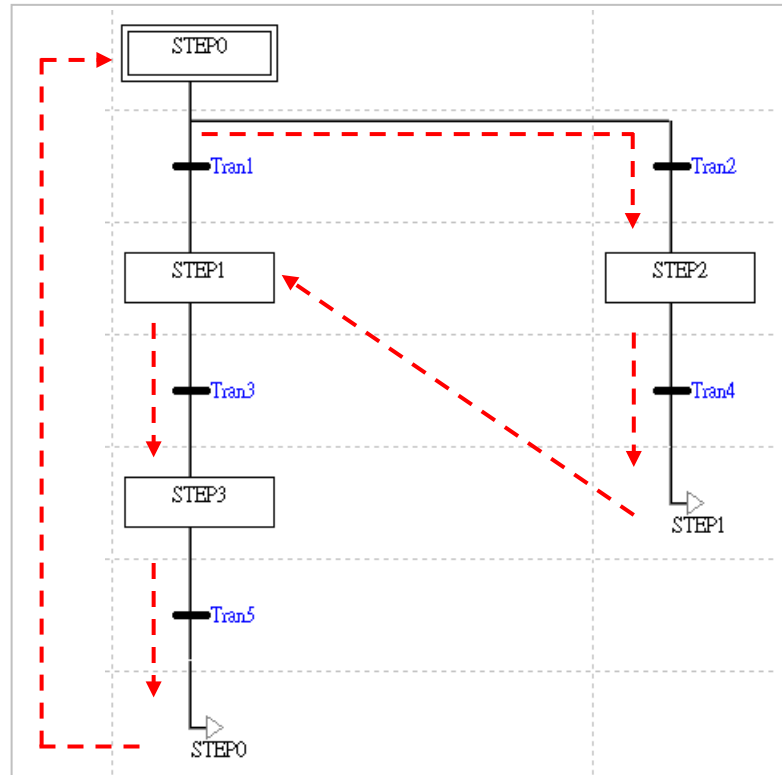
如下图，当转换点 Tran1 成立时，STEP_1 便会关闭并将 STEP_3 启动，而后续当 Tran2 的条件也成立时，STEP_2 便一样会被关闭，且不论 STEP_3 的状态为何，系统仍会再次启动 STEP_3。



14.2.5 跳跃点

SFC 允许使用跳跃的结构来将程序中的任一个步进点指定为执行权转移的对象，且在 SFC 的图形中至少会有一个跳跃点，也就是在图形最下方决定程序循环起点的位置。因为使用跳跃功能的自由度相当高，所以使用时请务必确认不会造成程序上的错乱。

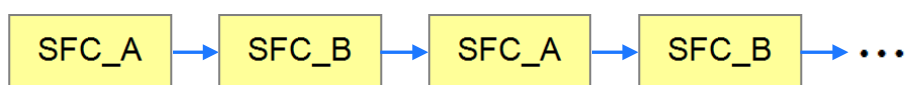
下图为一个包含选择分支结构的 SFC 程序，此处假设转换点 Tran2 的条件先行成立，因此执行的路径便会先转移至右侧，而当 Tran4 成立之后，程序便会经由跳跃功能转移至左侧的 STEP1；之后待程序执行到最下方且当 Tran5 的条件成立时，程序便会重新由 STEP0 开始继续执行。



使用步进点跳跃时须注意所指定的目标必须为同一 SFC 图形中有使用的步进点，若该步进点未于同一 SFC 程序中被使用时，即使该步进点已宣告于区域或全局符号表，编译时仍会产生错误。

不过若在系统的规划上需要从某个 SFC 跳跃至另一个 SFC 的程序当中继续执行时，在设计上可善用**步进点与转换点**的特性来达成，例如可将跳跃目标的步进点或转换点符号宣告于全局符号表，并在达成跳跃条件时将该符号设置为 ON，而此段程序则可写于某个 Action 的程序当中。

如下列范例，我们希望程序运作时会先执行 **SFC_A** 的程序，而当 **SFC_A** 的所有步序都完成后便跳跃至 **SFC_B** 的程序当中继续执行；而同样的，当 **SFC_B** 的所有步序都完成后便重新跳回 **SFC_A** 以继续下个循环；而在设计上我们则可透过**转换点**的特性来达成。

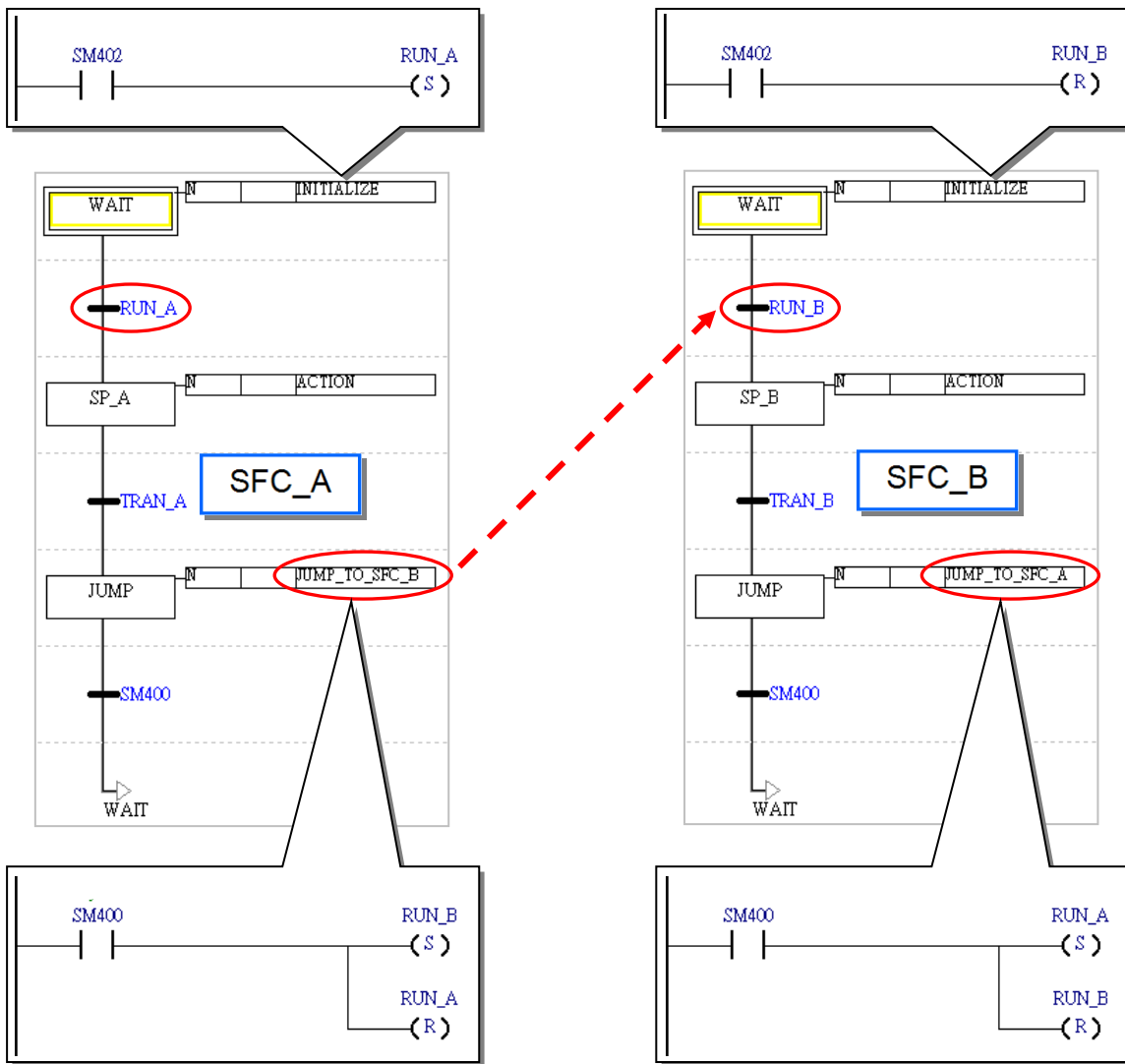


为了便于说明，下列的范例仅会用步进点「**SP_A**」与「**SP_B**」来表达原本要执行的 SFC 程序，且在其前后也都加上了控制用的步进点，并将转换点「**RUN_A**」与「**RUN_B**」宣告于全局符号表中。当程序执行时，两个 SFC 都会进入步进点「**WAIT**」，并执行「**INITIALIZE**」的 Action；且因 SFC_A 中的「**INITIALIZE**」会在第一个扫描周期中将转换点「**RUN_A**」设置为 ON，因此 SFC_A 的程序便会在随后转移至步进点「**SP_A**」中，并执行预定的程序；而 SFC_B 中的「**INITIALIZE**」则会在第一个扫描周期中将转换点「**RUN_B**」设置为 OFF，因此 SFC_B 便会停留在「**WAIT**」当中继续等待。

当 SFC_A 执行至步进点「JUMP」时，其 Action 会将转换点「RUN_B」设置为 ON，并将「RUN_A」重置为 OFF，而当转换点「RUN_B」设置为 ON 后，SFC_B 的程序便可转移至步进点「SP_B」中开始执行其预定的程序；除此之外，因为「JUMP」下方的转换点是保持为 ON 的特殊继电器，因此 SFC_A 的步序会立即跳回「WAIT」步进点，且因其转换点「RUN_A」已被重置为 OFF，因此 SFC_A 便会停留在「WAIT」步进点当中，并等待 SFC_B 的程序再次将「RUN_A」设置为 ON。

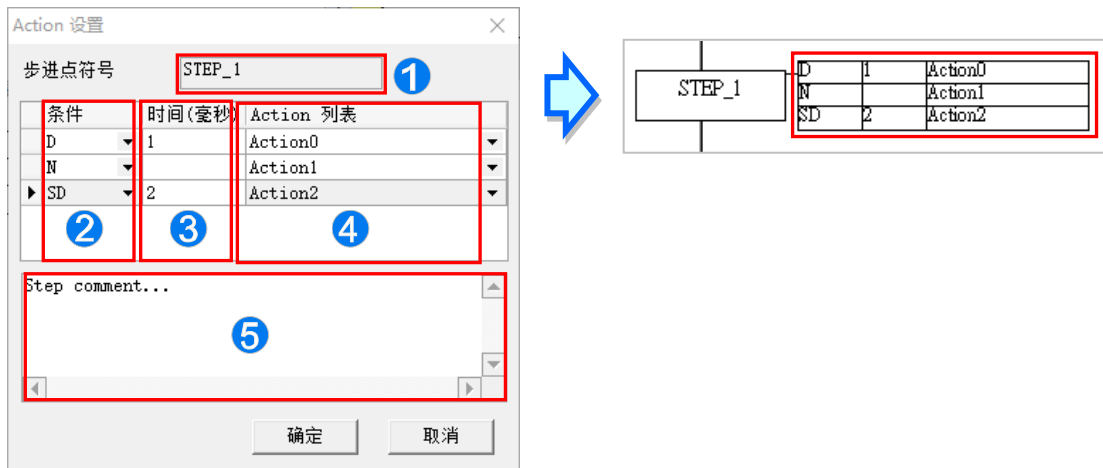
在这个架构中，步进点「WAIT」与「JUMP」的功能只用于切换彼此的转换点，其余的时间在实际意义上都是处于空转的状态；而当程序所应用的机种为 AH/AS 系列时，在规划上还可将两个步进点当中的 Action 指定为「P」或「P1」的修饰条件，相关细节将于第 14.2.6 节当中另行介绍。

14



14.2.6 Actions 的修饰条件

在 ISPSOft 中，用户可透过对 Action 的修饰，来对其执行方式加以定义，不过目前的 DVP 系列机种无法设置修饰条件，因此都将被视为条件 N，即一般的执行方式。



- ① 显示此组设置所属的步进点名称。
- ② 此选择欲指定的修饰条件。
- ③ 当选择与时间相关的修饰条件时，于此可设置时间属性。
- ④ 此选择被修饰的 Action。
- ⑤ 此字段可编辑此组设置的批注说明。

14.2.6.1 条件定义

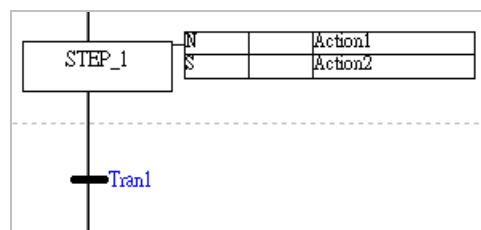
● N (Normal)

此为一般的执行方式，当步进点启动时，动作便会立即执行，而当步进点因转移而失效后，系统便会再执行一次 **Final Scan**，并将所有的动作与输出停止，不过对于使用类似 **SET** 指令的输出则会保持原状。目前 DVP 系列机种的所有 Action 都将指定为此种条件。

● S (Set)

当步进点启动时，动作便会立即执行，且一旦 Action 开始执行之后，即使步进点已因执行权转移而失效，该 Action 仍会保持执行状态。

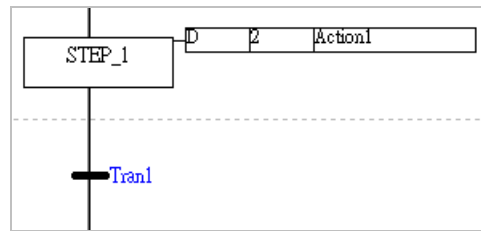
如下图，当 STEP_1 启动时，Action1 与 Action2 便会立即执行，但当 Tran1 条件成立而让 STEP_1 失效时，Action1 便会停止动作，并执行 **Final Scan**，而 Action2 则会继续维持执行状态。



● D (Delay)

当步进点启动时，须延迟一段时间后 Action 才会开始执行；但若转换条件在延迟时间到达之前即已成立，则此步进点将会直接关闭并转移至下一步进点，而 Action 也不会执行，且到了下次这个步进点再次启动时，延迟的时间将会重新开始计时。

如下图，当 STEP_1 启动后，Action1 并不会马上执行，而必须在延迟 2 秒之后，Action1 才会开始执行动作，但若 STEP_1 在延迟时间到达前即已失效，Action1 便不会执行动作。



● SD (Set Delay)

当步进点启动时，需延迟一段时间后 Action 才会开始执行，而一旦 Action 开始执行之后，不论步进点是否已经转移，该 Action 仍会保持执行状态。此外，即使转换条件在延迟时间到达之前即已成立，且完成了步进点的转移，然而延迟计时的动作仍会继续，并在计时到达之后开始执行 Action 的动作。

● DS (Delay Set)

此条件与 SD 类似，差别在于当转换条件在延迟时间到达之前就已成立时，设置为 DS 条件的延迟计时动作便会停止，且 Action 也不会执行，而到了下次这个步进点再次启动时，延迟的时间便会重新开始计时。不过一旦 Action 开始执行之后，不论步进点是否已经转移，该 Action 仍会保持执行状态。

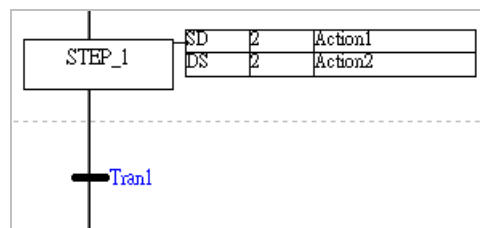
如下图，当 STEP_1 启动后，Action1 与 Action2 并不会马上执行，而必须在延迟 2 秒之后才会开始动作，不过依据 STEP_1 维持启动状态的时间，可分别考虑下列两种状况。

(a) STEP_1 维持启动状态的时间大于 2 秒

Action1 与 Action2 在时间到达 2 秒时便会开始执行，且之后即使步进点 STEP_1 因步进点转移而失效后，Action1 与 Action2 仍会维持执行状态。

(b) 步进点 STEP_1 维持启动状态的时间小于 2 秒

Action1 在步进点转移之后仍会继续计时，而在时间到达 2 秒时便会开始执行并维持执行状态；而 Action2 则会在步进点转移后停止计时，并且也不会执行 Action2 的动作。



● L (Limit)

Action 在步进点启动时便会立即执行，但当运行时间到达设置值时，即使步进点尚未转移，Action 还是会自动停止，并执行 **Final Scan**；而若步进点在时间到达之前即已转移，Action 便会随之停止。

● SL (Set Limit)

当步进点启动时，动作便会立即执行，但会固定运行时间。当 Action 的运行时间已到达设置值时，即使步进点尚未转移，Action 仍会自动停止，并执行一次 **Final Scan**；而当步进点在时间到达之前即已转移时，Action 还是会等待运行时间到达之后才会停止。

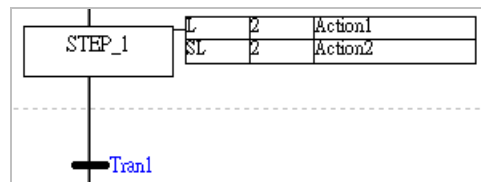
如下图，当 STEP_1 启动时，Action1 与 Action2 便会开始执行，此处再分别考虑下列两种状况。

(a) 步进点 STEP_1 维持启动状态的时间大于 2 秒

Action1 与 Action2 在时间到达 2 秒时便会自动停止动作，并执行 **Final Scan**。

(b) 步进点 STEP_1 维持启动状态的时间小于 2 秒

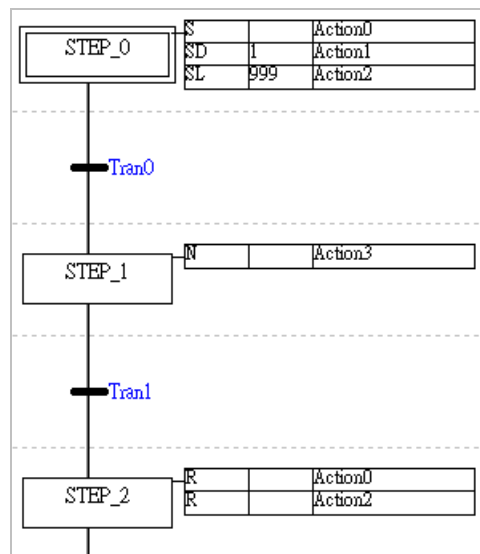
Action1 在步进点转移之后便会停止动作并执行 **Final Scan**，但 Action2 仍会继续等待时间到达 2 秒之后才会停止。



● R (Reset)

停止 Action 并执行 **Final Scan**，且即使是由 S、SD、DS 或 SL 条件来启动的 Action 都会被关闭。

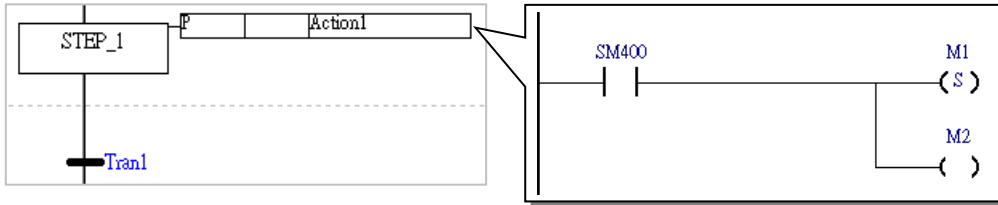
如下图，假设 Action0、Action1 与 Action2 在 STEP_0 为启动状态时即已顺利执行动作，并且在步进点转移至 STEP_2 之前都会维持执行状态；当步进点转移至 STEP_2 时，Action0 与 Action2 的动作都将被停止，并执行 **Final Scan**，而 Action1 则因未被重置，所以仍会保持执行状态。



● P (Pulse)

在步进点启动后的第一个扫描周期，动作便会立即执行，但到了第二个周期时，即使步进点仍在启动状态，所有的动作还是会被停止，并执行 **Final Scan**。

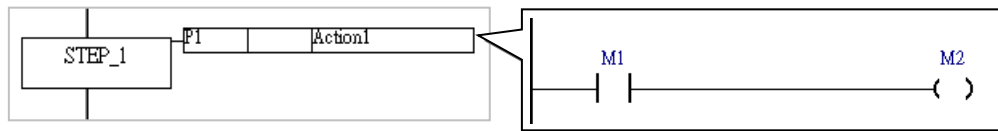
如下图，在 STEP_1 启动后的第一个扫描周期 Action1 便会开始执行，而因 SM400 的状态为 ON，因此 M1 与 M2 也都会为 ON；而到了第二个扫描周期时，系统便会将 Action1 的动作关闭，此时即使 SM400 还是为 ON，M2 都会变成 OFF，不过因为 M1 使用 SET 输出，所以仍会保持在 ON 的状态。



● P1 (Raising Pulse)

设置此条件的 Action 只会在步进点启动后的第一个扫描周期执行一次，且也不会执行 Final Scan 的程序；因此当 Action 的内部有使用 OUT (如下图) 或 TMR 等需要将动作关闭的指令时，若将其赋予 P1 的修饰条件，在使用上仍是允许的，但编译时会出现警告。

下图的 Action1 在 STEP_1 启动后的第一个扫描周期便会执行动作，此时若 M1 为 ON，则 M2 也会为 ON，但因使用 P1 修饰的 Action 只会在该步进点启动后的第一个扫描周期中执行一次，并且在之后也不会执行 Final Scan 的程序，因此后续不论 M1 的状态为何，M2 都将不再随之改变状态。

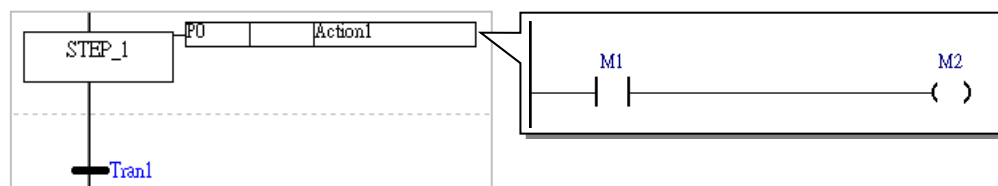


14

● P0 (Falling Pulse)

设置此条件的 Action 只会在步进点刚失效的扫描周期中执行一次，且之后也不会执行 Final Scan 的程序；因此当 Action 的内部有使用 OUT (如下图) 或 TMR 等需要将动作关闭的指令时，若将其赋予 P0 的修饰条件，在使用上仍是允许的，但编译时会出现警告。

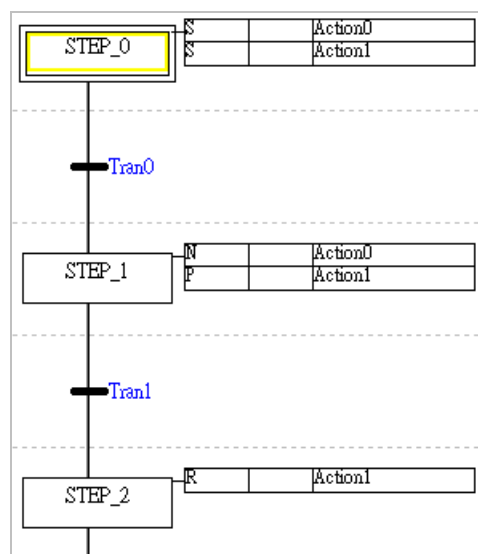
下图的 Action1 在 STEP_1 启动后仍不会执行动作，而必须等待 Tran1 的条件成立而使得 STEP_1 失效时才会执行一次，此时若 M1 为 ON，则 M2 也会为 ON，且因使用 P0 修饰的 Action 只会在该步进点失效的扫描周期中执行一次，并且在之后也不会执行 Final Scan 的程序，因此之后不论 M1 的状态为何，M2 都将不再随之改变状态。



14.2.6.2 设置修饰条件的注意事项

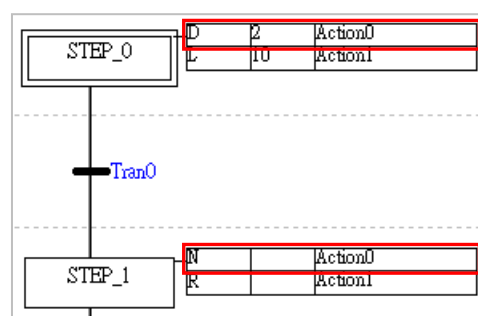
- 当 Action 一旦由 S 条件来启动之后，即使于他处将该 Action 设置为 N、P 或其它会将 Action 停止的条件时，该 Action 仍会维持在执行的状态，而唯有在被赋予条件 R 并顺利执行之后，该 Action 才会被重置。因此当程序中的 Action 曾被赋予 S 的条件属性时，请务必确认是否须配置条件 R 来将该 Action 重置。

如下图，Action0 与 Action1 在 STEP_0 启动时便会开始执行并维持其执行状态，而当步进点转移至 STEP_1 时，即使两者分别被赋予 N 与 P 的修饰条件，Action0 与 Action1 仍会继续执行；直到步进点转移至 STEP_2 时，Action1 会因修饰条件 R 而被重置并停止动作，但 Action0 则因没有任何 R 条件的修饰，以致在整个程序的执行过程中会一直保持在执行状态。



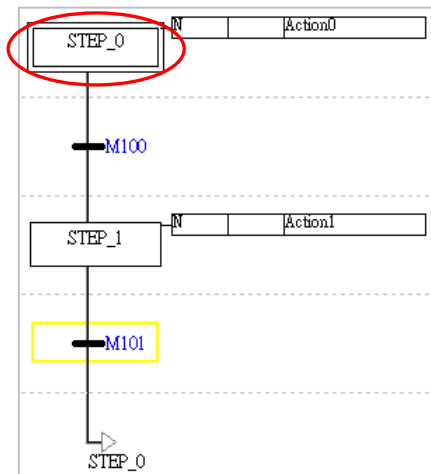
- 设置与时间相关的修饰条件的 Action，除了设置为 R 的修饰条件之外，只能在 SFC 程序中指定一次。也就是当某个 Action 被设置为 D、SD、DS、L 或 SL 的其中一个条件时，除了设置为 R 之外，该 Action 就不可再被使用。

如下图范例，Action0 与 Action1 在 STEP_0 的步序中都分别被赋予与时间相关的修饰条件，而在 STEP_1 的步序中，Action1 则被赋予条件 R，因此其用法是正确的；但 Action0 却又再度被赋予非 R 的修饰条件，因此 Action0 的部份在程序进行编译时便会产生错误。



14.2.7 初始步进点

在 SFC 的执行过程中，因每个步进点都会轮流地被启动且持续循环，因此必须定义一个唯一的步进点做为程序第一次执行的起始位置，而这个步进点便称之为**初始步进点**。在 ISPSOft 中，初始步进点会以双线的方框来表示，且每个 SFC 程序只能有一个初始步进点。



AH/AS 系列机种可指定 SFC 程序中的任一个步进点做为初始步进点，且在 PLC 启动运转后，当第一次执行至该 POU 时便会自动由指定的初始步进点开始执行。

而在 DVP 系列的机种中，其 SFC 的 POU 并无法自动执行，而必须藉由其他的 POU 对 SFC 中的某个步进点执行 SET 指令之后，该 SFC 才会开始运转；因此在 DVP 机种的 SFC 图形中无法指定初始步进点，且会固定在图形中的第一个位置；而实际运作时，真正的初始步进点则决定于在其他 POU 中被下达 SET 指令的步进点装置或符号。下列则为 DVP 机种的 SFC 启动范例。

首先，请决定该 SFC 的实际初始步进点，并将初始步进点的变量符号宣告在全局符号表，接着在其它的 POU 当中对该初始步进点使用 SET 指令；而在对步进点符号下达 SET 指令的程序代码中，须注意其前方的条件接点或逻辑输出，要避免在程序执行的过程中还会变为 ON 的状态，否则便可能再次将该步进点设置为 ON 而造成 SFC 的执行程序产生错乱；建议使用特殊继电器 M1002，而该继电器的特性则是只有在主机启动后的第一个扫描周期才会为 ON。



14.2.8 内部属性

为了让用户在编程时，能更容易使用 SFC 某些属性参数，内部属性供用户在各 POU 之间调用。这些属性参数可分为 Step、Action 及 Transition 三种类别，本章节将会对此三种类别内容逐一介绍。

14.2.8.1 Step 属性参数

若用户需要调用 SFC 的 POU 中，Step 的属性参数，目前属性参数的架构，是以结构(STRUCTURE)方式来呈现。

此结构由「Step 所在 SFC 之 POU 名称」、「Step 的名称」及「Step 的成员」组成，用户可以选择适当的 POU 及 Step 名称后，再决定所需使用 Step 的成员。



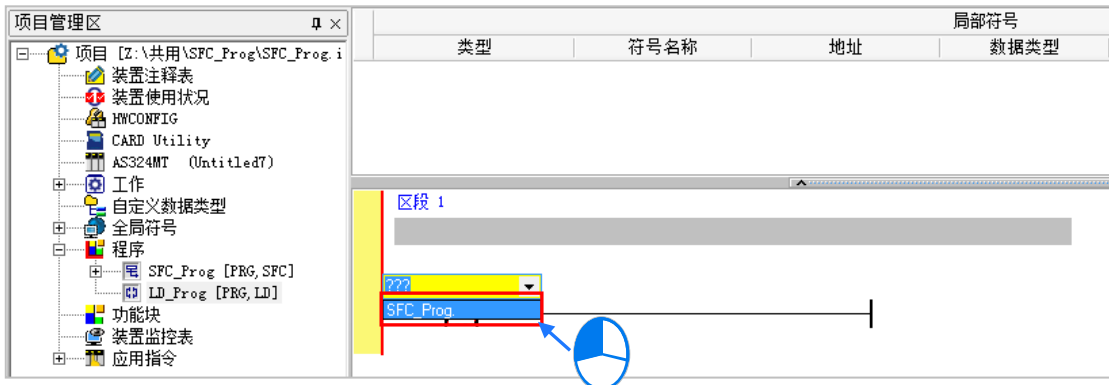
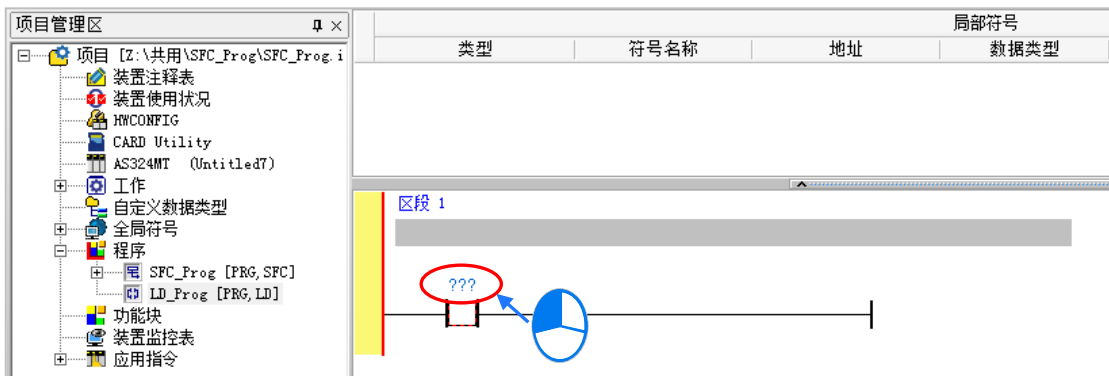
下表为 Step 的成员分类及作用一览表。

类型	成员	说明	读	写
Step	X	当 Step 正在执行，此 SFC 属性参数将为 ON；此 SFC 属性参数用法将取代过去 S 装置	V	X
	done	当 Step 之 T 属性(目前 Step 已运行时间) >= Step 之 timePreset 属性，此值将为 ON	V	X
	counter	当 Step 之 X 属性由 OFF 变为 ON，此属性参数计数加 1	V	X
	timePreset	决定 Step 之 done 属性由 OFF 到 ON 所需经过的时间(单位:ms，数据长度:DWORD)	V	X
	maxTime	记录 Step 之 T 属性，曾经计时最大时间(单位:ms，数据长度:DWORD)	V	X
	T	目前 Step 已执行多少时间(单位:ms，数据长度:DWORD)	V	X

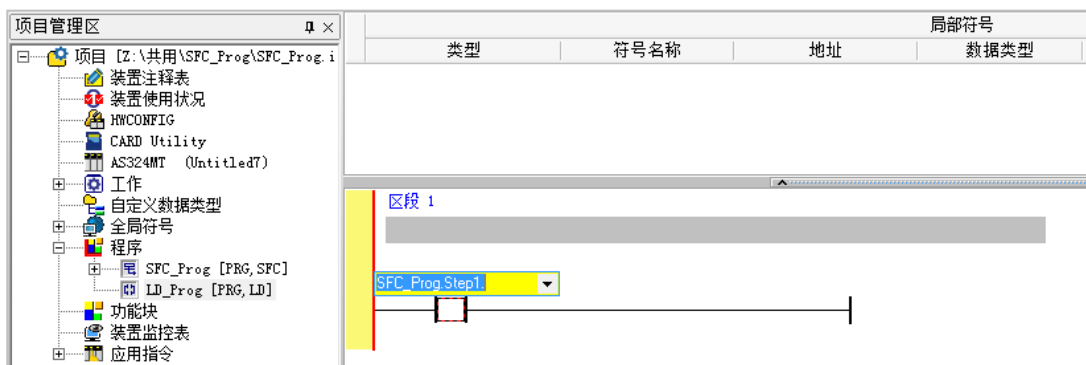
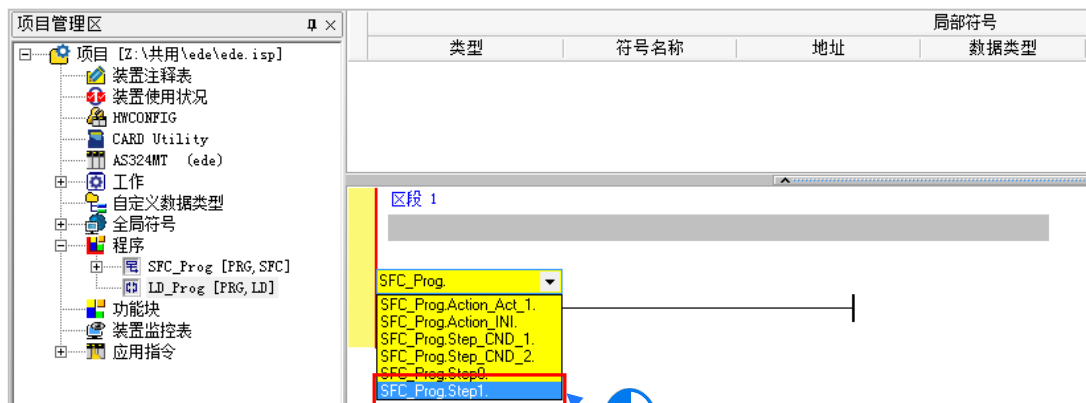
14

以 SFC 之 Step 的成员 X 的操作使用为例，若希望在所写的 SFC 程序 POU 的 Step 开始执行时，触发 Ladder 程序 POU 中的常开接点为 ON，程序书写方式说明如下。

可以在 Ladder 程序 POU 中使用一常开接点，并于常开接点位置上方点击鼠标左键，此时下拉选单将自动带出可选择的 SFC 程序 POU 的名称，接着点击 SFC 程序 POU 的名称。



点击 SFC 程序 POU 的名称后，再点击键盘右键，此时下拉选单将自动带出可选择的 SFC 程序 POU 含 Step 的名称，接着点击欲选择的 SFC 程序 POU 含 Step 的名称。

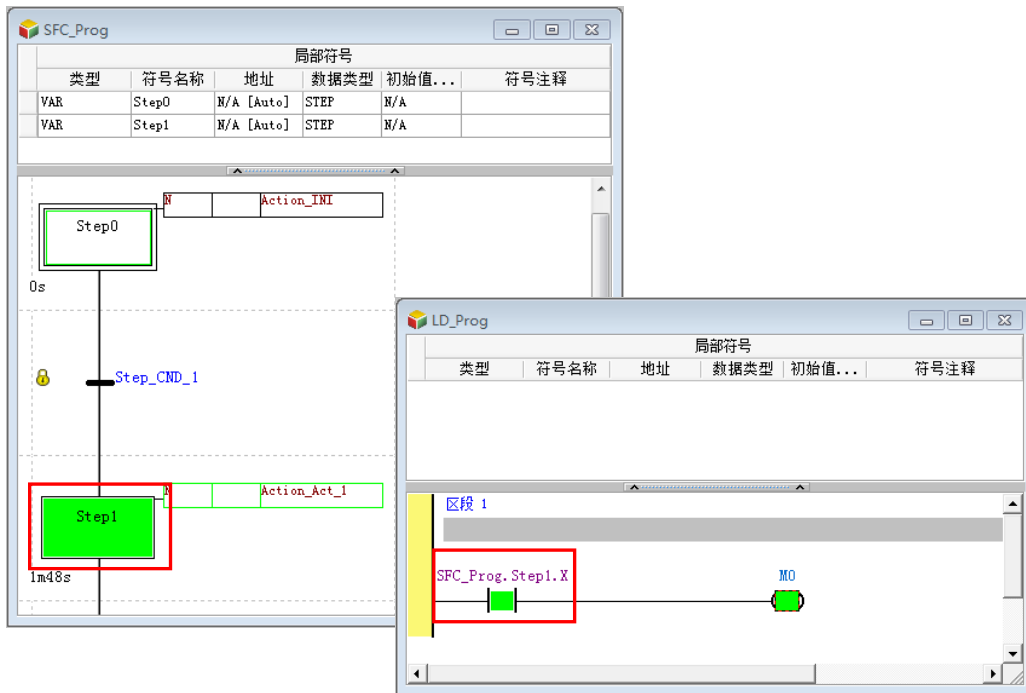


点击 SFC 程序 POU 含 Step 的名称后，再点击键盘右键，此时下拉选单将自动带出可选择的 SFC 程序 POU 含 Step 含成员的名称，接着点击欲选择的 SFC 程序 POU 含 Step 含成员的名称，即完成选取。



14

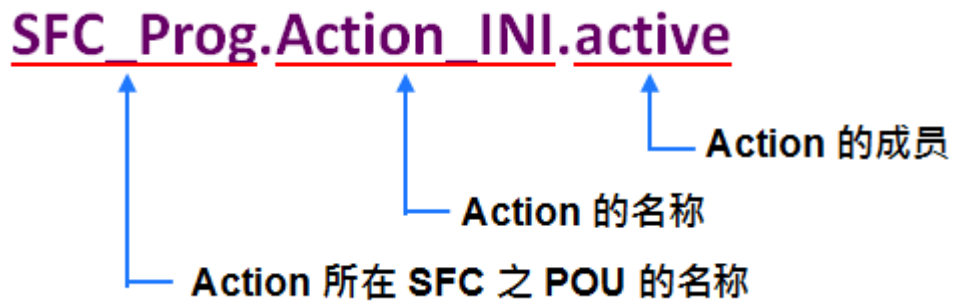
当在执行 SFC 程序 POU 的 Step 时，SFC_Prog.Step1.X 此常开接点将会 ON。



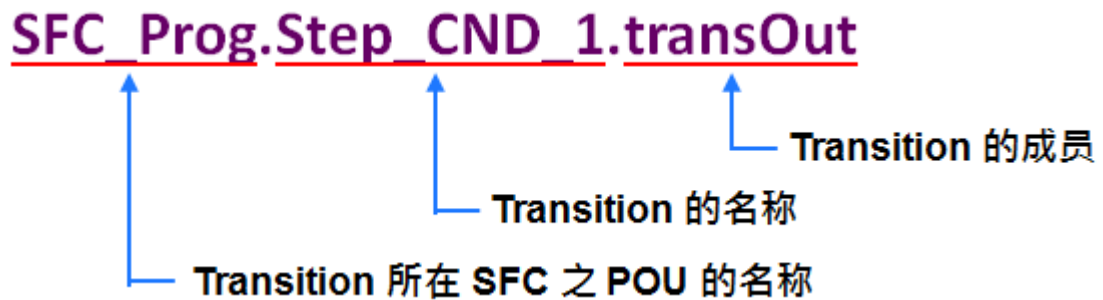
14.2.8.2 Action 与 Transition 属性参数

若用户需要调用 SFC 的 POU 中 Action 或 Transition 的属性参数，目前属性参数的架构是以结构 (STRUCTURE) 方式来呈现。

Action 属性参数结构由「Action 所在 SFC 之 POU 名称」、「Action 的名称」及「Action 的成员」组成，用户可以选择适当的 POU 及 Action 名称后，再决定所需使用 Action 的成员。



Transition 属性参数结构由「Transition 所在 SFC 之 POU 名称」、「Transition 的名称」及「Transition 的成员」组成，用户可以选择适当的 POU 及 Transition 名称后，再决定所需使用 Transition 的成员。

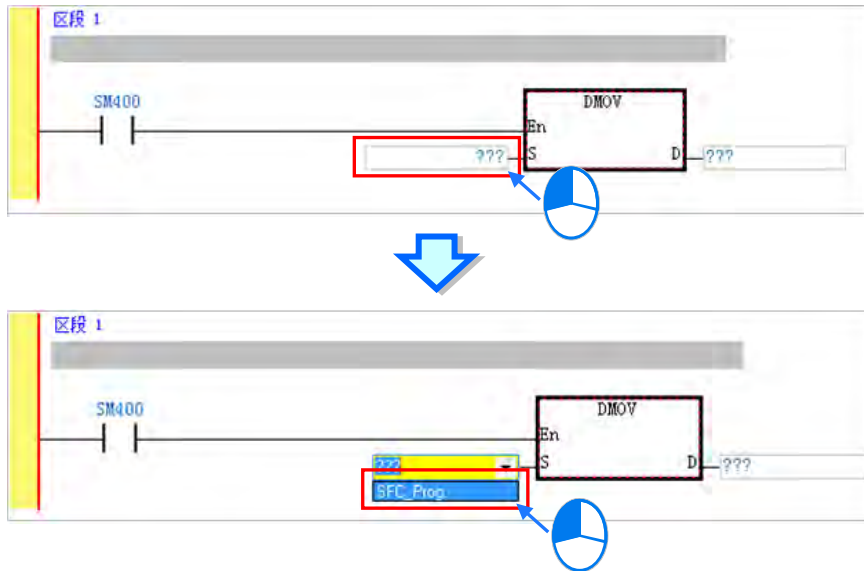


下表为 Action 及 Transition 的成员分类及作用一览表。

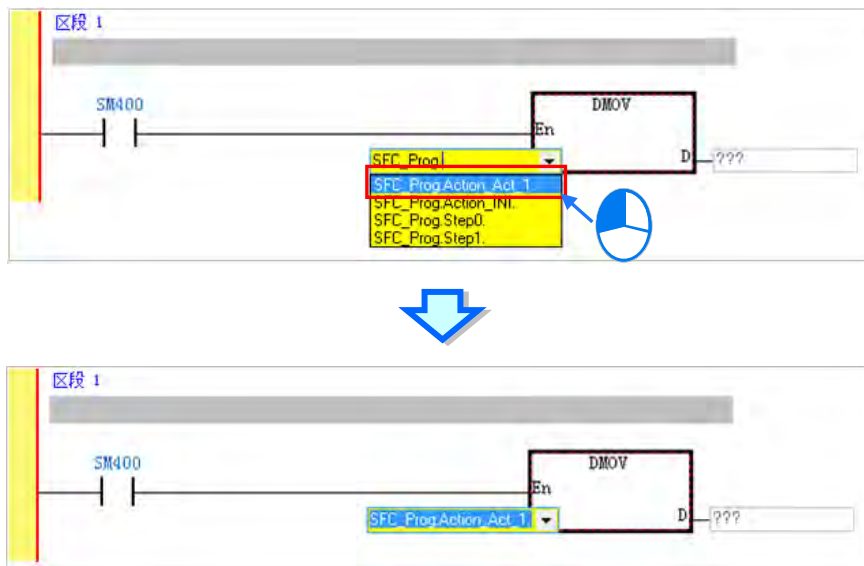
类型	成员	说明	读	写
Action	enabled	当 Action 的 qualifier 条件成立时，此 SFC 属性参数为 ON，可看出 Action 是否启用	V	X
	active	当完成第一次执行 Action 程序，此 SFC 属性参数将为 ON	V	X
	rstDisable	当 Final scan 的功能启用时，此值为 OFF 当 Final scan 的功能不启用时，此值为 ON	V	X
	counter	当 Action 的 active 属性由 OFF 变为 ON，此属性参数计数加 1	V	X
	setTimer	跟时间相关修饰 (qualifier) 的时间显示 (单位:ms，数据长度: DWORD)	V	X
	transOut	当 Transition 移转条件成立为 ON，或 Transition 被强制启动时为 ON	V	X

以 SFC 之 Action 的成员 counter 的操作使用为例，若希望监看 Action 的 active 属性参数，由 0 变为 1 的次数，并将此 active 属性参数值传至 D100 暂存器，程序书写方式说明如下。

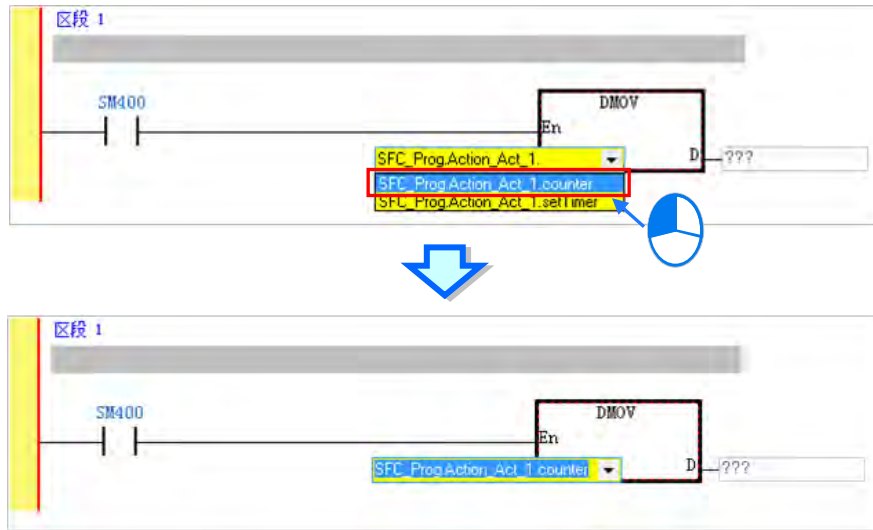
可以在 Ladder 程序 POU 中使用 SM400 常开接点及 DMOV 指令，对 DMOV 指令的 S 接脚，点击鼠标左键，此时下拉选单将自动带出可选择的 SFC 程序 POU 的名称，接着点击 SFC 程序 POU 的名称。



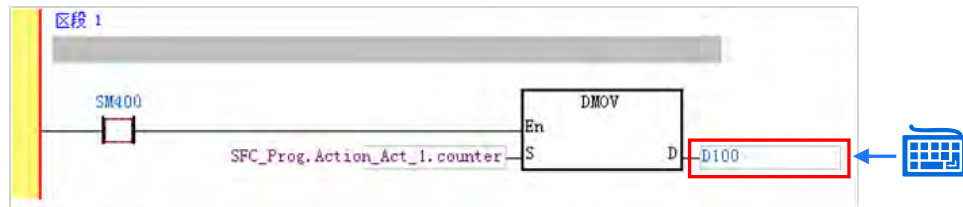
点击 SFC 程序 POU 的名称后，再点击键盘右键，此时下拉选单将自动带出可选择的 SFC 程序 POU 含 Action 的名称，接着点击欲选择的 SFC 程序 POU 含 Action 的名称。



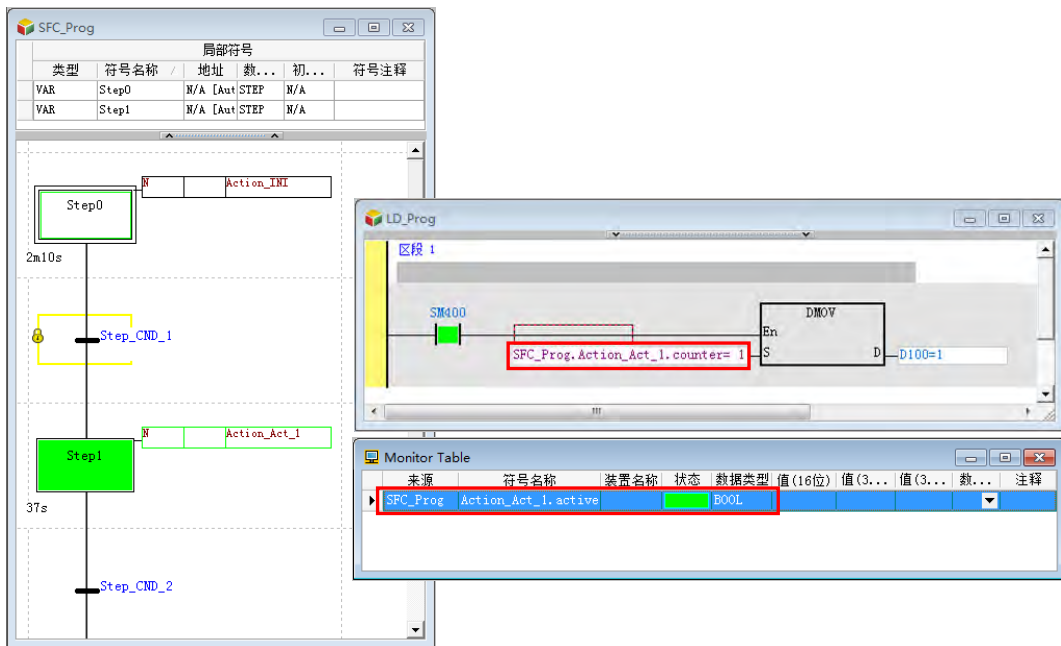
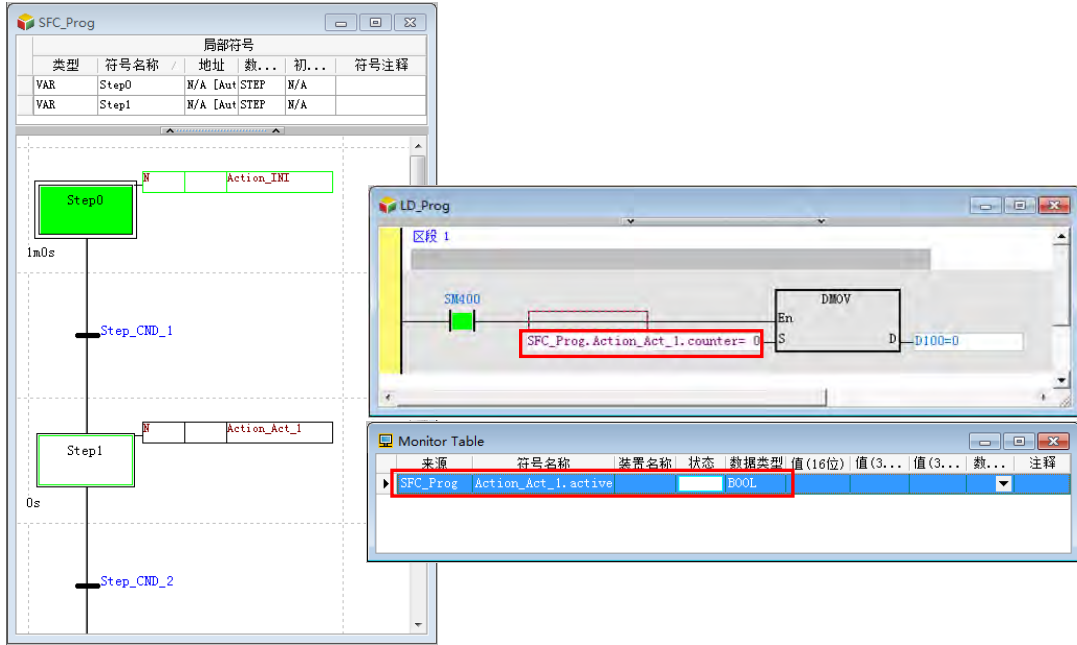
点击 SFC 程序 POU 含 Action 的名称后，再点击键盘右键，此时下拉选单将自动带出可选择的 SFC 程序 POU 含 Action 含成员的名称，接着点击欲选择的 SFC 程序 POU 含 Action 含成员的名称，即完成选取。



对 DMOV 指令的 D 接脚，输入 D100，完成撰写。



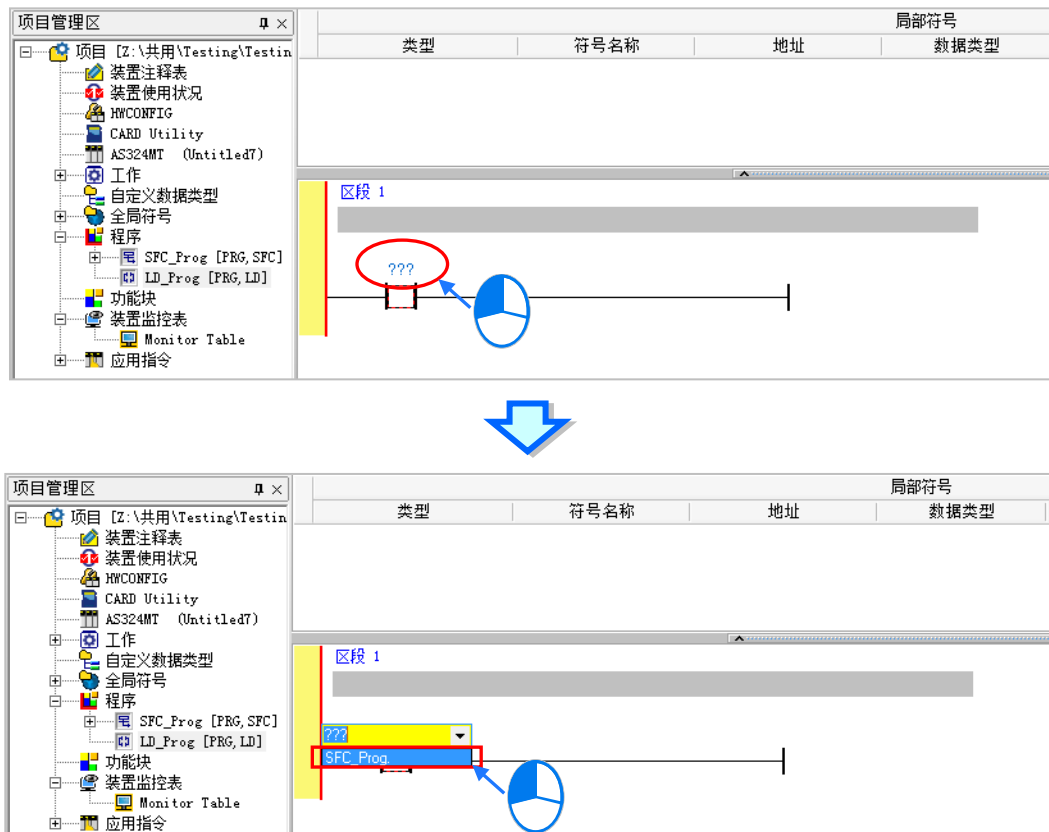
在 SFC_Prog.Action.Act_1.Active，由 0 变为 1 时，SFC_Prog.Action.Act_1.counter 计数将会加 1。



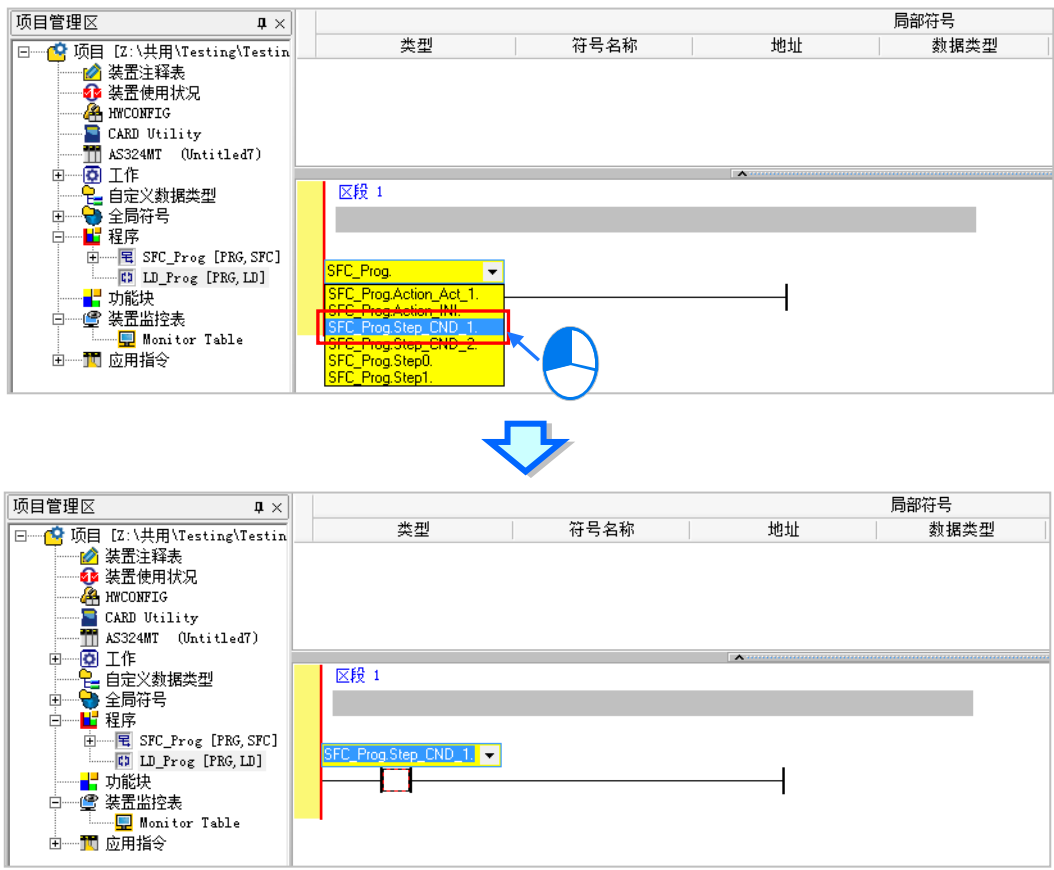
14

使用 SFC 之 Transition 的成员 transOut 的操作使用为例。若希望在撰写之 SFC 程序 POU 的 Transition 移转条件成立时，触发 Ladder 程序 POU 中的常开接点为 ON，程序撰写的方式，说明如下。

可以在 Ladder 程序 POU 中使用一常开接点，并于常开接点位置上方，点击鼠标左键，此时下拉选单将自动带出可选择的 SFC 程序 POU 的名称，接着点击 SFC 程序 POU 的名称。

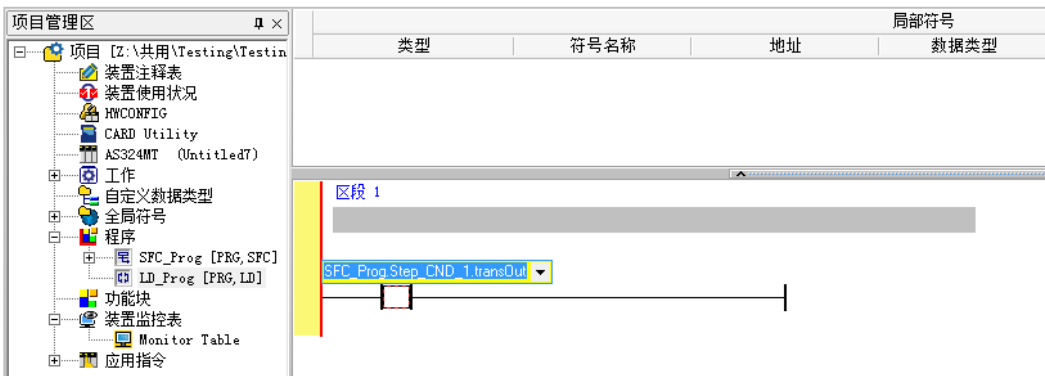


点击 SFC 程序 POU 的名称后，再点击键盘右键，此时下拉选单将自动带出可选择的 SFC 程序 POU 含 Transition 的名称，接着点击欲选择的 SFC 程序 POU 含 Transition 的名称。



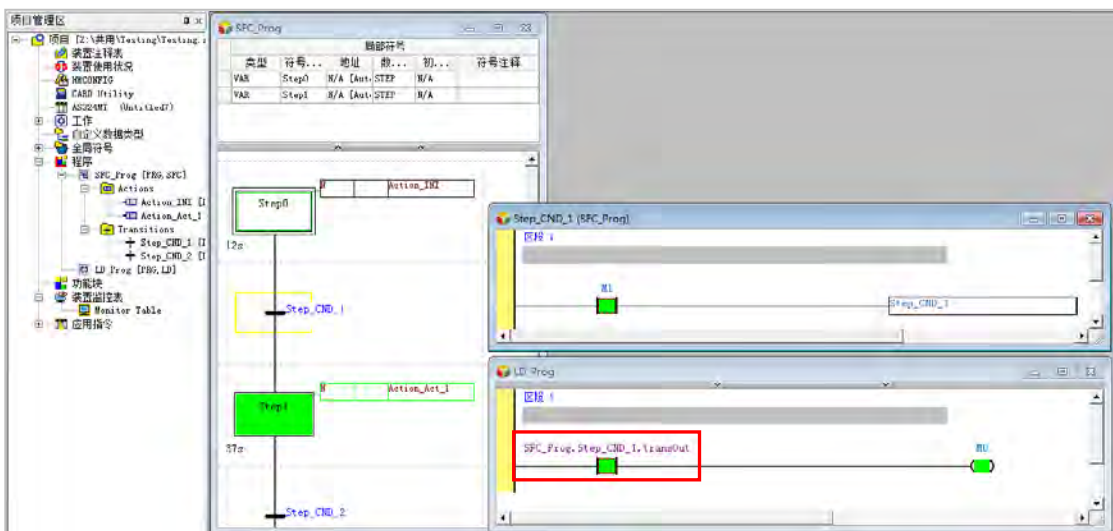
14

点击 SFC 程序 POU 含 Transition 的名称后，再点击键盘右键，此时下拉选单将自动带出可选择的 SFC 程序 POU 含 Transition 含成员的名称，接着点击欲选择的 SFC 程序 POU 含 Transition 含成员的名称，即完成选取。



14

当在 SFC 程序 POU 的此 Step_CND_1 此 Transition 条件成立时 SFC_Prog.Step_CND_1.transOut 此常开接点将会 ON。

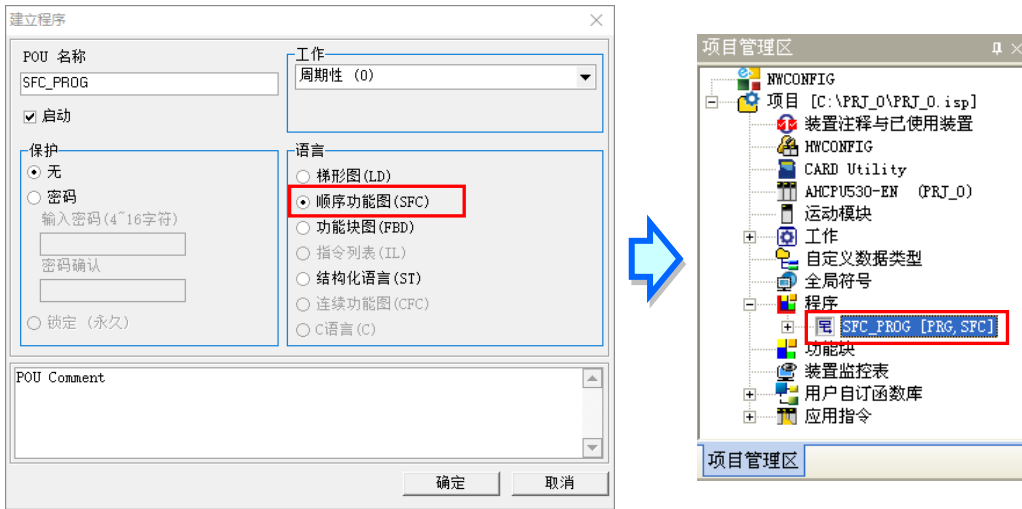


14.3 在 ISPSOft 中建立 SFC 程序

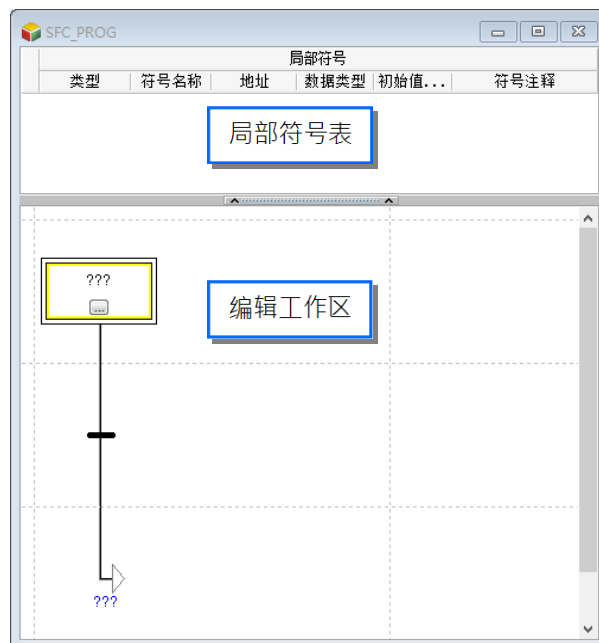
14.3.1 ISPSOft 的 SFC 编辑环境

在建立 POU 时，于语言字段选择「顺序功能图 (SFC)」便可建立 SFC 程序，更进一步的操作细节请参考第 5.4.1 节的说明。而除此之外，建立 SFC 程序时须再留意下列两点事项。

- (a) 建立功能块 POU 时不可选择 SFC 语言。
- (b) SFC 语言的程序 POU 不得指派周期性以外的工作 (TASK)。



下图为 ISPSOft 中的 SFC 编辑环境，基本的窗口配置与其他的语言相同，上方为区域符号表，下方则是 SFC 图形的编辑工作区；而 SFC 的程序图面则是以棋盘的方式来呈现，每个步进点或转换点都会依序被放在图面的方格中，当用鼠标点击程序图面上的对象或是该对象所处的方格时，该对象便会被选取，并以黄色的外框来表示已被选取的状态。



进入 SFC 的编辑环境后，ISPSoft 的窗口中便会出现对应的编程图标工具栏，其功能简介如下。



图示	键盘 (快捷键)	功能说明
	Shift + F1	新增步进结构于选取位置上方
	Shift + F2	新增步进结构于选取位置下方
	Shift + F3	选取的转换点右侧并联一新转换点
	Shift + F4	选取的转换点左侧并联一新转换点
	Shift + F5	选取的步进点右侧并联一新步进点
	Shift + F6	选取的步进点左侧并联一新步进点
	Shift + F7	选取的步进点右侧建立一共进分支&选择汇合的结构
	Shift + F8	选取的步进点左侧建立一共进分支&选择汇合的结构
	Shift + F9	选取的转换点右侧建立一选择分支&并行汇合的结构
	Shift + F10	选取的转换点左侧建立一选择分支&并行汇合的结构
	Shift + F11	插入一跳跃点

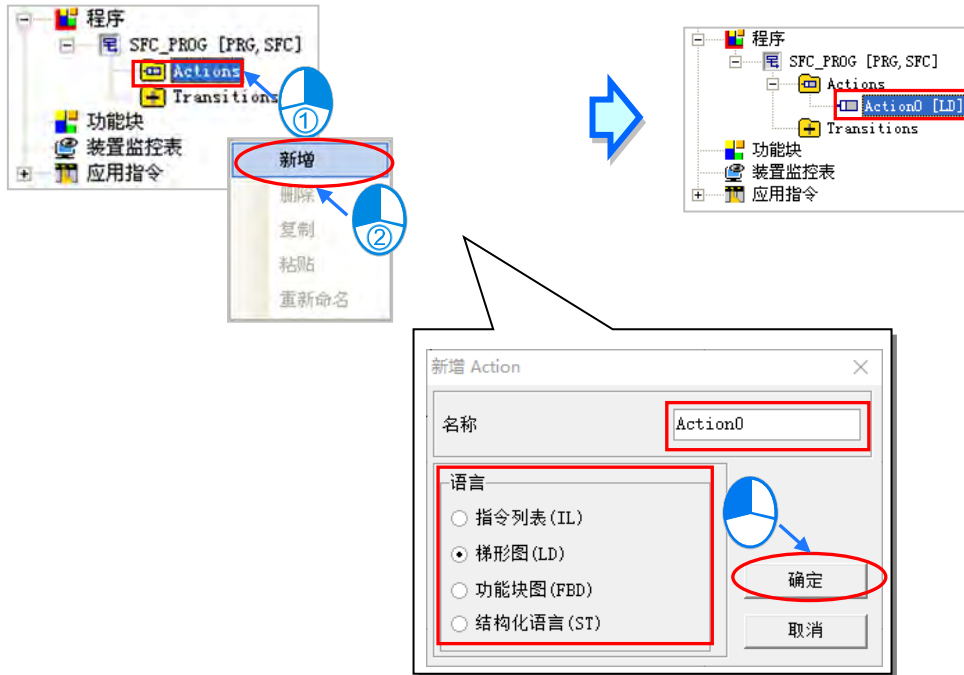
14

除了可通过图示工具栏来进行编辑之外，当点选编辑工作区中某个对象后按下鼠标右键，于快捷选单的「编辑项目」下一样会有相同的操作项可供使用，而为了确保图形的完整性与合法性，随着点选位置的不同，可使用的图形编辑功能也会跟着不同。



14.3.2 Actions 与 Transitions 的建立与管理

当要新增 Action 的项目时，请先于项目管理区中展开 SFC 的程序 POU，并点选其下方的「Actions」项目后按下鼠标右键，接着再于快捷选单中点选「新增」，最后再于弹出的设置窗口中设置 Action 的名称以及开发此 Action 时所使用的编程语言即可。

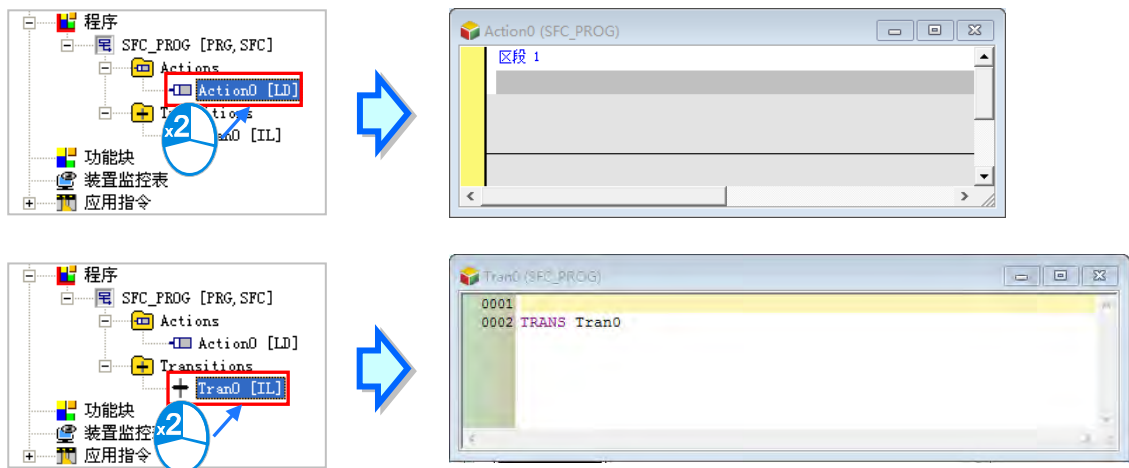


而要新增 Transition 的项目时，请同样先于项目管理区中展开 SFC 的程序 POU 后，在其下方点选「Transitions」项目后按下鼠标右键，接着再于快捷选单中点选「新增」，最后再弹出的设置窗口中设置 Transition 的名称以及开发此 Transition 时所使用的编程语言即可。



项目管理区中建立 Action 与 Transition 的项目后，直接于建立的项目上双击鼠标左键，之后画面便会出现对应所选语言的程序编辑窗口，而其中在 Transition 的编辑窗口中还会自动带出 Transition 符号的部份。

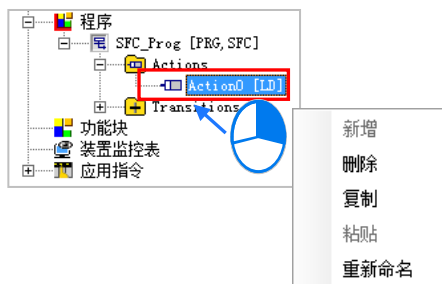
Action 与 Transition 的编辑方式与一般的 POU 相同，唯一不同的是在编辑窗口中没有区域符号表，因此要宣告区域符号时，请于 SFC 图形的编辑窗口中进行宣告，而同一个 SFC 程序的所有 Action 与 Transition 都将共用同一个区域符号表。



14

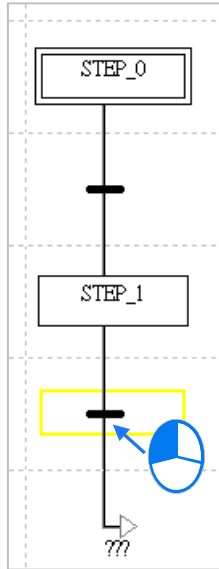
*. 若建立的 Transition 已指定给 SFC 图中的某个转换点时，于该转换点上双击鼠标左键亦可开启 Transition 的编辑窗口。



当在建立的 Action 或 Transition 项目上按下鼠标右键后，快捷选单中便有提供一些操作项可供使用，透过这些操作功能便可对所建立的 Action 或 Transition 项目进行管理。其中当使用复制/贴上功能时，须注意不可将 Actions 的项目复制至 Transitions 的项目下，当然也不可将 Transitions 的项目复制至的 Actions 项目下。














14.3.3 新增步进点结构












(1) 先在编辑工作区中点选欲插入步进点结构的位置。

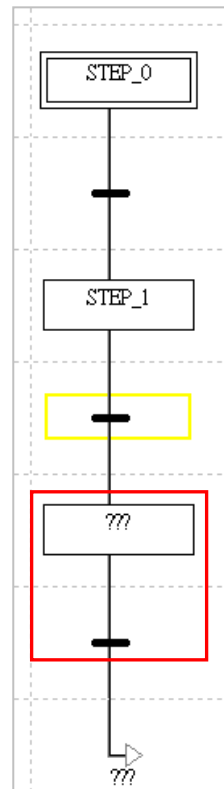


(2) 点击图示工具栏或右键快捷选单 > 编辑项目中的  图标便可将新的步进点结构插于选取位置的上方，而按下  图标则可将新结构插至选取位置的下方。

OR

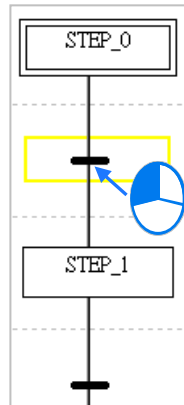
	在上方插入步进点	Shift+F1
	在下方插入步进点	Shift+F2
	在右方插入转换	Shift+F3
	在左方插入转换	Shift+F4
	在右方插入步进点	Shift+F5
	在左方插入步进点	Shift+F6
	在右方插入步进点(并行分支, 选择汇合)	Shift+F7
	在左方插入步进点(并行分支, 选择汇合)	Shift+F8
	在右方插入转换(选择分支, 并行汇合)	Shift+F9
	在左方插入转换(选择分支, 并行汇合)	Shift+F10
	插入跳跃	Shift+F11





14

14.3.4 插入并联转换点

(1) 先在编辑工作区中点选欲并联转换点的位置。



(2) 点击图示工具栏或右键快捷选单 > 编辑项目中的  图标便可将转换点并联至选取位置的右方，而按下  图标则可将转换点并联至选取位置的左方。

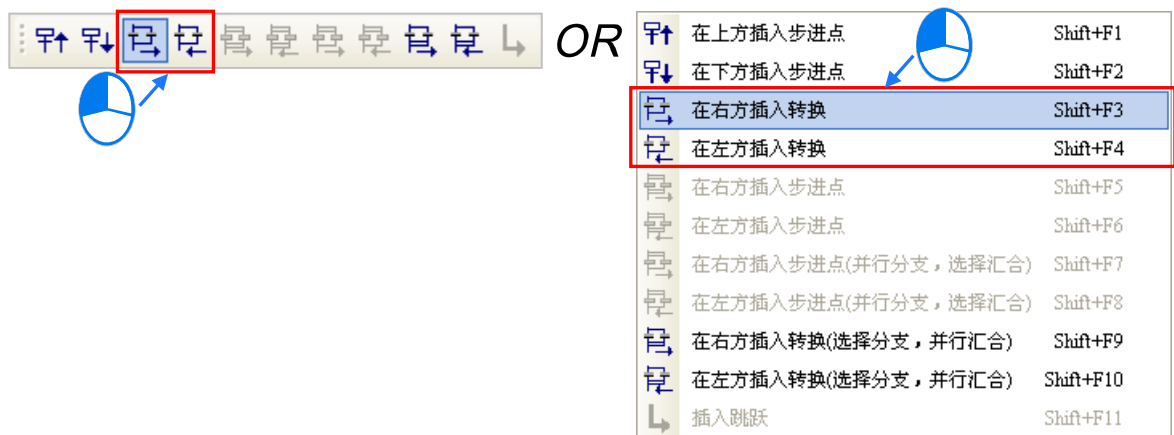











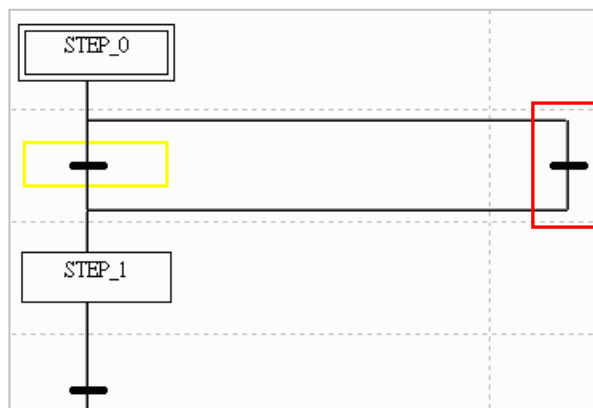


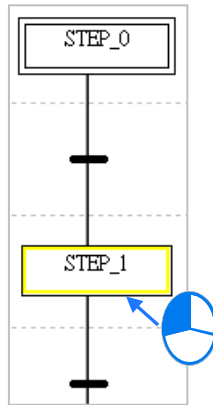
Diagram illustrating the toolbars and context menu for inserting parallel transition points. The toolbars show icons for inserting transition points to the right (right-click icon) and left (left-click icon). The context menu lists the following options:



	在上方插入步进点	Shift+F1
	在下方插入步进点	Shift+F2
	在右方插入转换	Shift+F3
	在左方插入转换	Shift+F4
	在右方插入步进点	Shift+F5
	在左方插入步进点	Shift+F6
	在右方插入步进点(并行分支, 选择汇合)	Shift+F7
	在左方插入步进点(并行分支, 选择汇合)	Shift+F8
	在右方插入转换(选择分支, 并行汇合)	Shift+F9
	在左方插入转换(选择分支, 并行汇合)	Shift+F10
	插入跳跃	Shift+F11



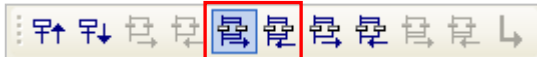
14.3.5 插入并联步进点












(1) 先在编辑工作区中点选欲并联步进点的位置 (图面最上方的步进点无法与其他的步进点并联)。

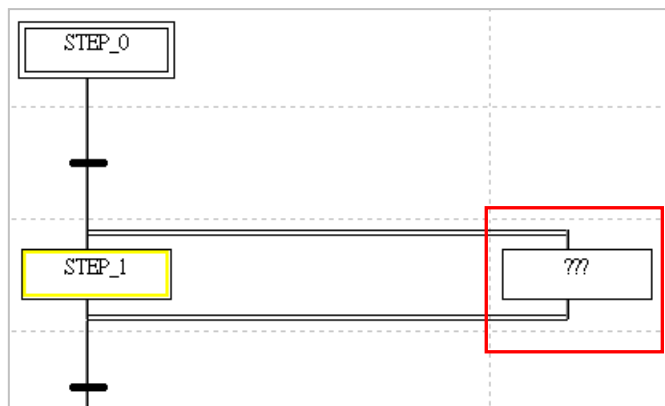


(2) 点击工具栏或右键快捷选单 > 编辑项目中的  图示便可将步进点并联至选取位置的右方，而按下图示  则可将其并联至选取位置的左方。

14

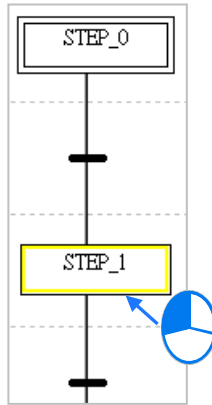
 OR



	在上方插入步进点	Shift+F1
	在下方插入步进点	Shift+F2
	在右方插入转换	Shift+F3
	在左方插入转换	Shift+F4
	在右方插入步进点	Shift+F5
	在左方插入步进点	Shift+F6
	在右方插入步进点(并行分支, 选择汇合)	Shift+F7
	在左方插入步进点(并行分支, 选择汇合)	Shift+F8
	在右方插入转换(选择分支, 并行汇合)	Shift+F9
	在左方插入转换(选择分支, 并行汇合)	Shift+F10
	插入跳跃	Shift+F11

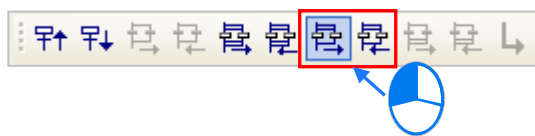


14.3.6 插入步进点结构 - 共进分支&选择汇合












- (1) 先在编辑工作区中点选欲插入结构的步进点位置，而点选时必须点在步进点上，且不能是 SFC 图面中最上方的步进点。

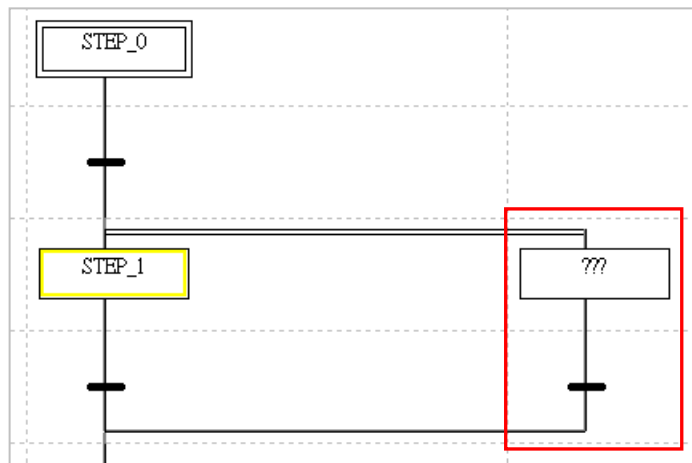


- (2) 点击工具栏或右键快捷选单 > 编辑项目中的  图标便可将结构并联至选取位置的右方，而按下  图标则可将结构并联至选取位置的左方。



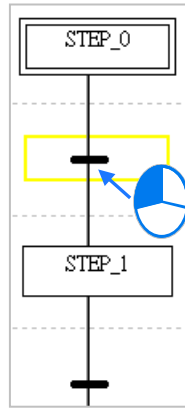
OR



	在上方插入步进点	Shift+F1
	在下方插入步进点	Shift+F2
	在右方插入转换	Shift+F3
	在左方插入转换	Shift+F4
	在右方插入步进点	Shift+F5
	在左方插入步进点	Shift+F6
	在右方插入步进点(并行分支, 选择汇合)	Shift+F7
	在左方插入步进点(并行分支, 选择汇合)	Shift+F8
	在右方插入转换(选择分支, 并行汇合)	Shift+F9
	在左方插入转换(选择分支, 并行汇合)	Shift+F10
	插入跳跃	Shift+F11



14.3.7 插入步进点结构 - 选择分支&并行汇合

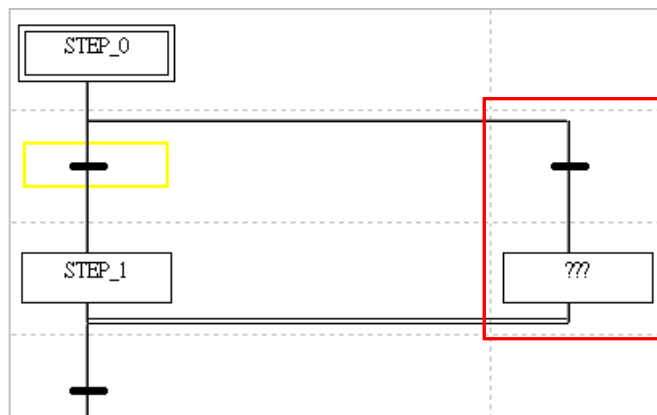
- (1) 先在编辑工作区中点选欲插入结构的转换点位置，而点选时必须点在转换点上，且不能是 SFC 图面中最下方的转换点。



- (2) 点击工具栏或右键快捷选单 > 编辑项目中的  图标便可将结构并联至选取位置的右方，而按下  图标则可将结构并联至选取位置的左方。

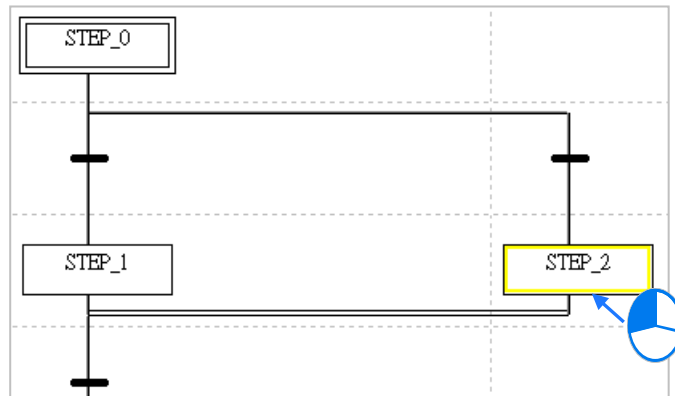
14


	在上方插入步进点	Shift+F1
	在下方插入步进点	Shift+F2
	在右方插入转换	Shift+F3
	在左方插入转换	Shift+F4
	在右方插入步进点	Shift+F5
	在左方插入步进点	Shift+F6
	在右方插入步进点(并行分支, 并行汇合)	Shift+F7
	在左方插入步进点(并行分支, 并行汇合)	Shift+F8
	在右方插入转换(选择分支, 并行汇合)	Shift+F9
	在左方插入转换(选择分支, 并行汇合)	Shift+F10
	插入跳跃	Shift+F11

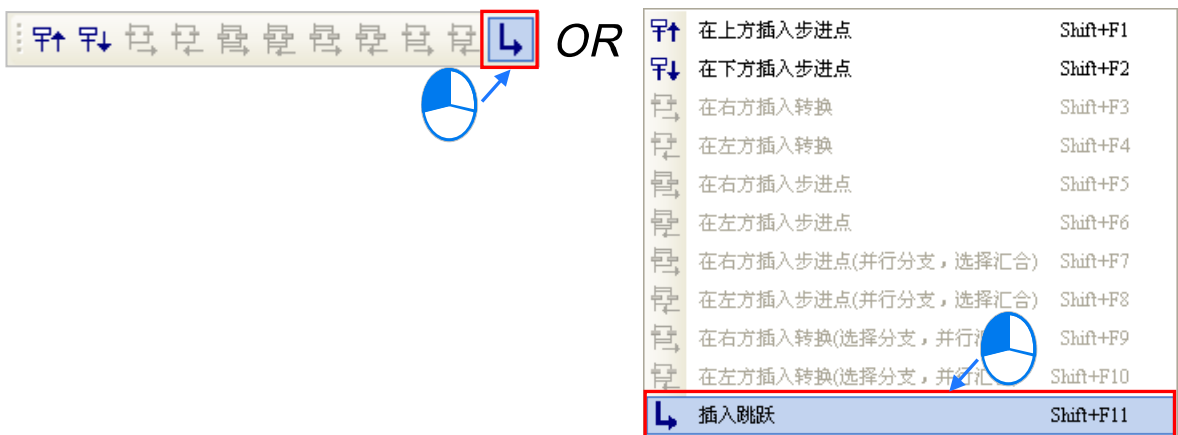


14.3.8 插入跳跃点

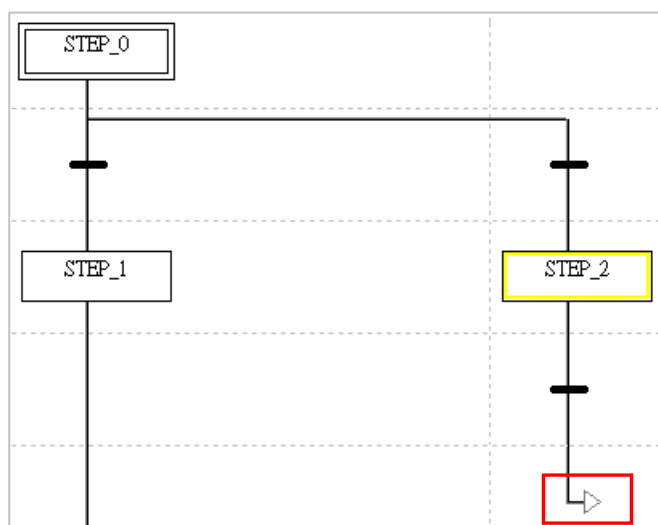
- (1) 先在编辑工作区中点选欲插入跳跃点的位置，而视汇合的结构，点选时必须点在分支路径中最下方的步进点或转换点。



- (2) 点击工具栏或右键快捷选单 > 编辑项目中的  图示即可插入跳跃点。

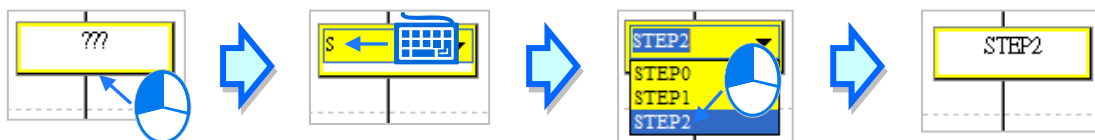


	在上方插入步进点	Shift+F1
	在下方插入步进点	Shift+F2
	在右方插入转换	Shift+F3
	在左方插入转换	Shift+F4
	在右方插入步进点	Shift+F5
	在左方插入步进点	Shift+F6
	在右方插入步进点(并行分支, 选择汇合)	Shift+F7
	在左方插入步进点(并行分支, 选择汇合)	Shift+F8
	在右方插入转换(选择分支, 并行汇)	Shift+F9
	在左方插入转换(选择分支, 并行汇)	Shift+F10
	插入跳跃	Shift+F11



14.3.9 指定步进点与转换点符号

当完成图形的部署后，接着便必须于图形中逐一指定每个位置所对应的**步进点**或**转换点**符号；编辑时请先点选欲编辑的**步进点**或**转换点**，待黄色方框出现后，于该位置再点击一次鼠标左键便会出现该**步进点**或**转换点**的编辑字段，输入部分文字后，便可从过滤出的下拉选单中选择已建立好的项目即可。



除了利用鼠标之外，用户亦可直接使用键盘来进行操作。先利用键盘的上下键移动画面中的黄色方框，在选定编辑位置后按下【Enter】键即会出现编辑字段，此时即可直接使用键盘来输入字符串，便会出现过滤出的下拉选单，利用键盘的上下键便可进行选择，完成编辑后，按下【Enter】键便会自动再切换至下一个编辑位置，当要结束编辑时，则只要按下【Esc】键即可。此外，当用键盘输入的**步进点**名称是尚未宣告的变量符号时，按下【Enter】键后，系统默认将会自动导出符号宣告窗口，但若输入的位置是**转换点**时，画面便不会出现符号宣告窗口。

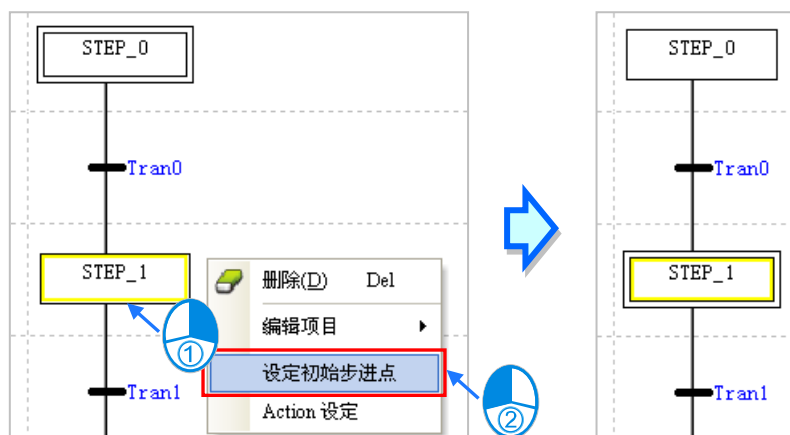


转换点可以指定为结构变量符号中的 **BOOL** 型态元素，关于结构的说明请参阅第 8 章。

*.自动导出符号宣告窗口的功能可由用户自行设置是否开启，设置方式则请参考第 2.3.1 节。

14.3.10 指定初始步进点

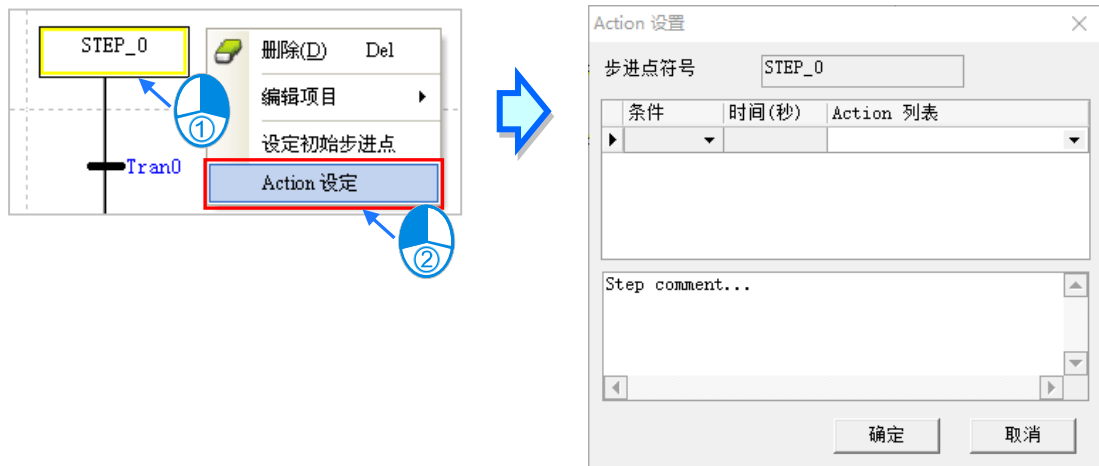
此功能仅限 AH/AS 机种，DVP 机种无法于画面中指定初始步进点。设置时请先在编辑工作区中点选要设为初始步进点的位置，之后按下鼠标右键，并于快捷选单中点选「设置初始步进点」即可。



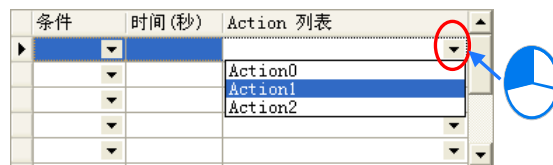
*.关于 DVP 机种的步进点启动请参考本章第 14.2.7 节的相关说明。

14.3.11 配置 Actions/折叠 Action 列表

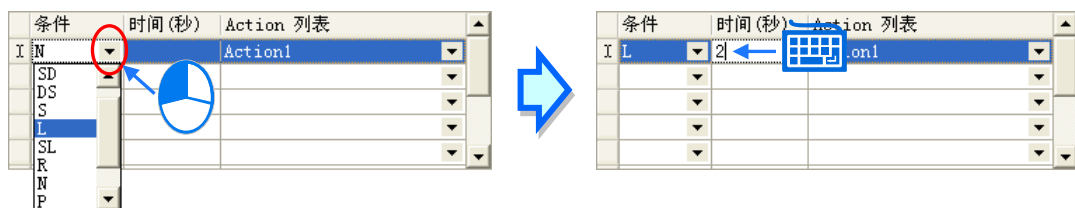
- (1) 首先点选欲配置 Action 的步进点后按下鼠标右键，接着快捷菜单中点选「Action 设定」项目便可开启设置窗口；但若该步进点的变量符号尚未宣告时，系统将不允许进行 Action 的配置。



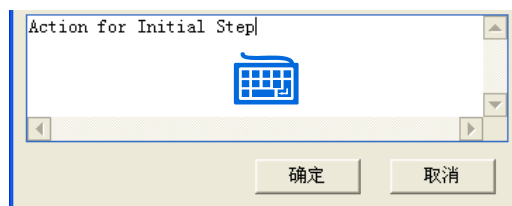
- (2) 开启设置窗口后，请先于「Action 列表」中下拉选择欲配置的 Action 项目。



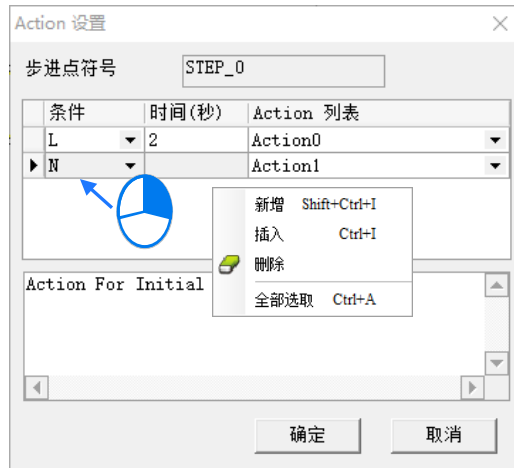
- (3) 选择好 Action 项目后，请再于「修饰条件」字段下拉选择对应的条件，且当设置的条件与时间相关时，必须再于「时间(秒)」字段中设置时间参数，而输入时必须以秒为单位，且只可输入整数（DVP 机种无法设置修饰条件）。



- (4) 若要输入批注时，请直接于最下方的空白字段中进行编辑即可。

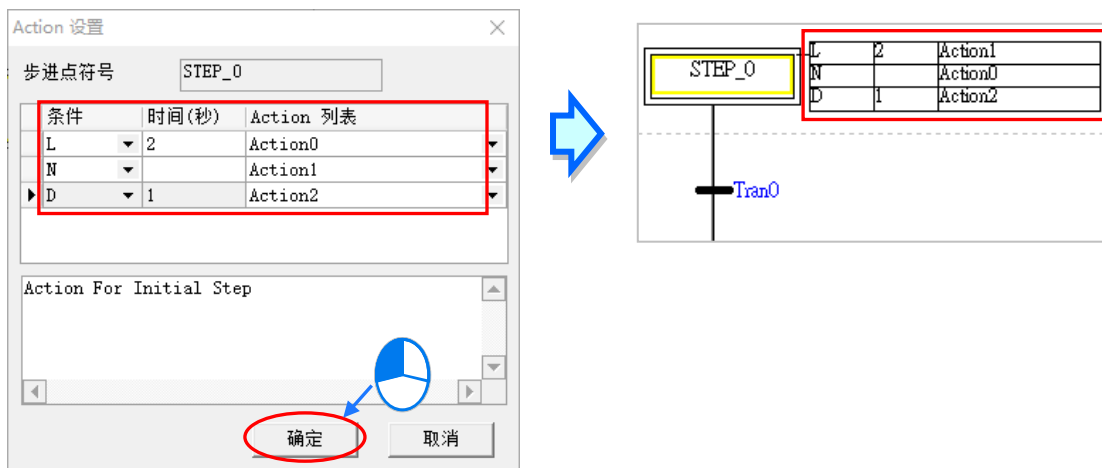


- (5) 当列表中选某一项后按下鼠标右键，之后便可透过快捷选单的操作项进行各种编辑；至于各操作项的功能则请参考下表。当要于 Action 设置窗口中同时选取多个行列时，可先按住键盘的【Ctrl】键，接着再用鼠标分别点选不连续位置的行列；另外也可先按住【Shift】键后，再先后用鼠标点选欲选取范围的第一个与最后一个行列，之后便可将整个范围内的行列全部选取起来。

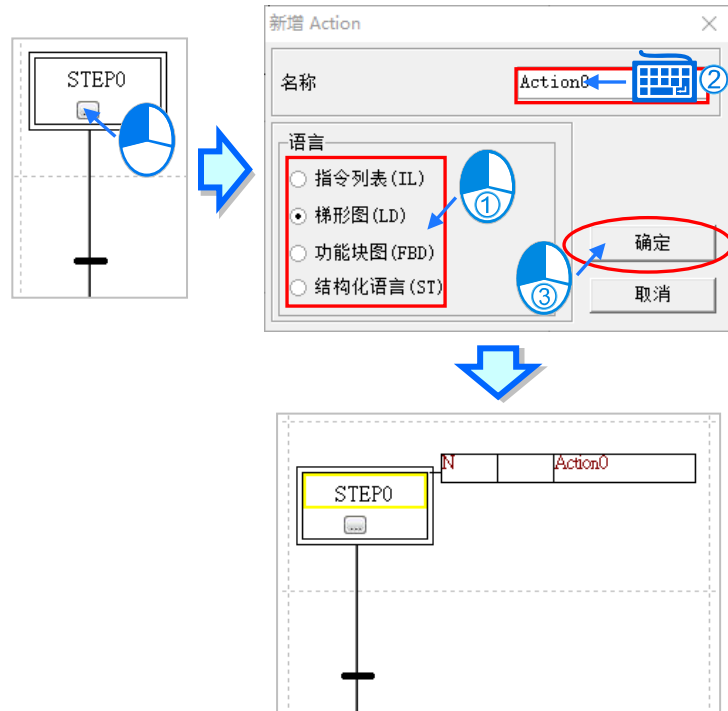


操作项	功能说明
新增	于所选项目的下方新增一新行列。
插入	于所选项目的上方插入一新行列。
删除	删除目前所选的项目。
全部选取	同时将所有项目选取起来。

- (6) 完成所有的设置后按下「确定」按钮即可，而此时于该步进点的右侧便会出现配置的列表。

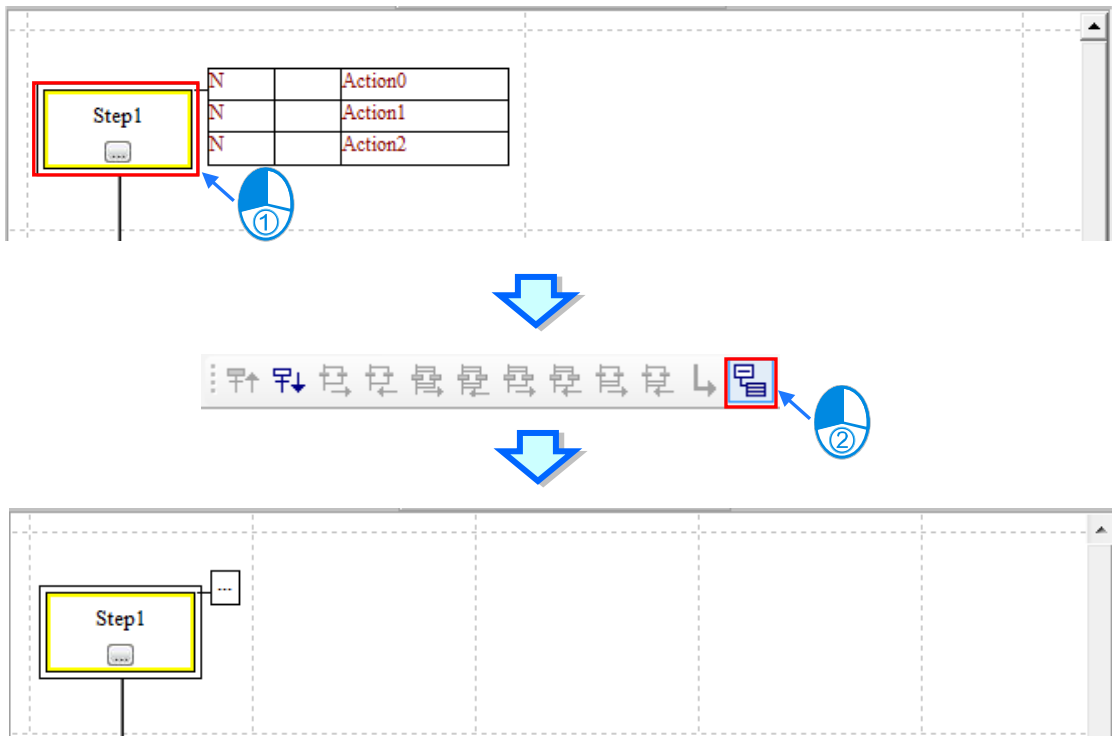


为便于用户快速的新增与配置 Action，ISPSOft 在 STEP 上提供了一个快捷按钮，点选此按钮后便会出现新增 Action 窗口，输入完成后按下「确定」按钮，便会同时建立 Action 与配置此 Action 到 STEP 中。



若用户觉得配置的 Action 会占用画面空间，可单击「折叠 Action 列表」按钮，来做折叠缩放的动作。

14



14.4 SFC 的程序范例

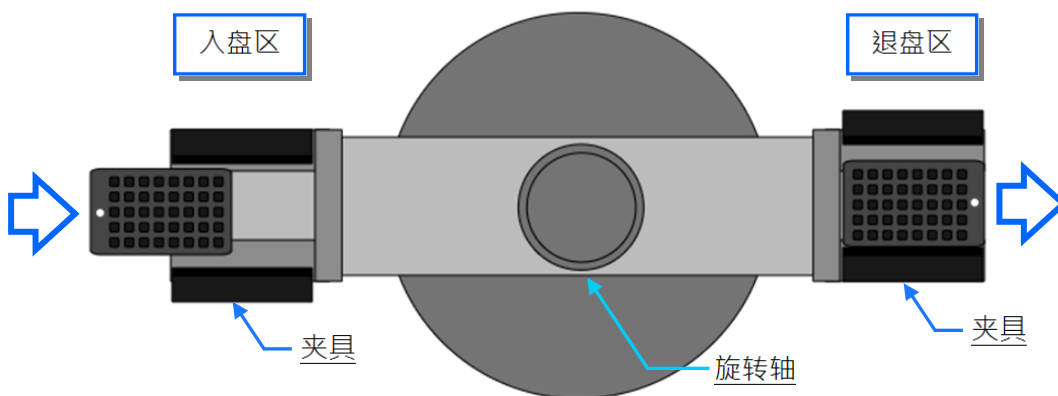
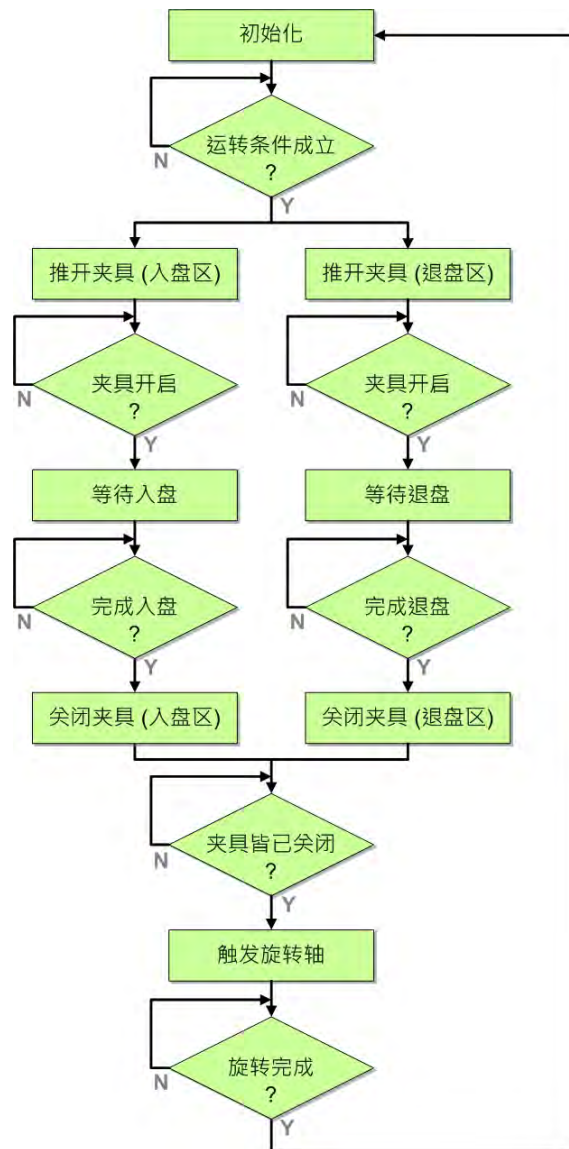
14.4.1 范例说明

此范例主要为两个工作站之间的载盘转送功能，右侧为其工作流程图，而下方则为实际设备的示意图。

如下图，平台的左侧为入盘区，而右侧则为退盘区，当入盘与退盘的工作完成后，连结平台中心的旋转轴便会旋转 180 度，之后于入盘区接收的载盘便可转送至退盘区，且完成退盘的空夹具亦可再次回到入盘区进行下次的入盘动作。在整个过程当中，除了旋转轴的动作之外，其余期间的入盘与退盘动作皆是彼此独立的。

当系统开始运转时，位于夹具下方的气缸会先把两侧的夹具推开，当夹具开启后，程序便可将入盘与退盘的 Ready 信号设置为 ON，并等待与此系统衔接的前后工作站各自完成其入盘与退盘的动作。

当完成入盘或退盘的动作后，系统便会收到回送的完成信号，此时便可将其对应的 Ready 信号设为 OFF，并将推开夹具的气缸退回以让夹具关闭；而当系统确认两侧的夹具皆已关闭后，系统便会送出旋转轴的触发信号，并在旋转完成且收到回送的完成信号后开始继续下一个循环。



14.4.2 硬件规划

在本范例中，我们将选用 AH5x0 系列主机 **AHCPU530-EN**，数字 IO 模块则选用 8 点输入/8 点输出的 **AH16AP11R-5A**，主背板则选用 4 槽的 **AHBP04M1-5A**。下表即为本范例的 IO 配置表。

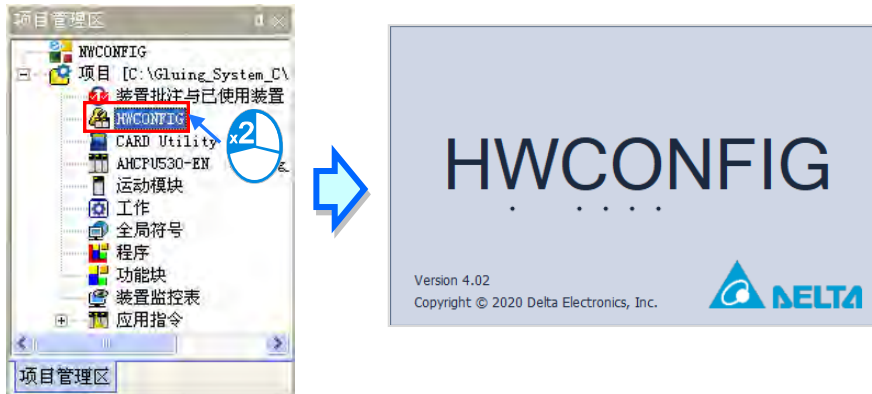
Type	ID	说明
DI	X0.0	入盘完成信号
DI	X0.1	退盘完成信号
DI	X0.2	旋转完成信号
DI	X0.3	(入盘区) 夹具气缸 - 开启 Sensor
DI	X0.4	(入盘区) 夹具气缸 - 关闭 Sensor
DI	X0.5	(退盘区) 夹具气缸 - 开启 Sensor
DI	X0.6	(退盘区) 夹具气缸 - 关闭 Sensor
DO	Y0.0	入盘 Ready 信号
DO	Y0.1	退盘 Ready 信号
DO	Y0.2	旋转轴触发信号
DO	Y0.3	(入盘区) 夹具气缸
DO	Y0.4	(退盘区) 夹具气缸

14.4.3 程序规划

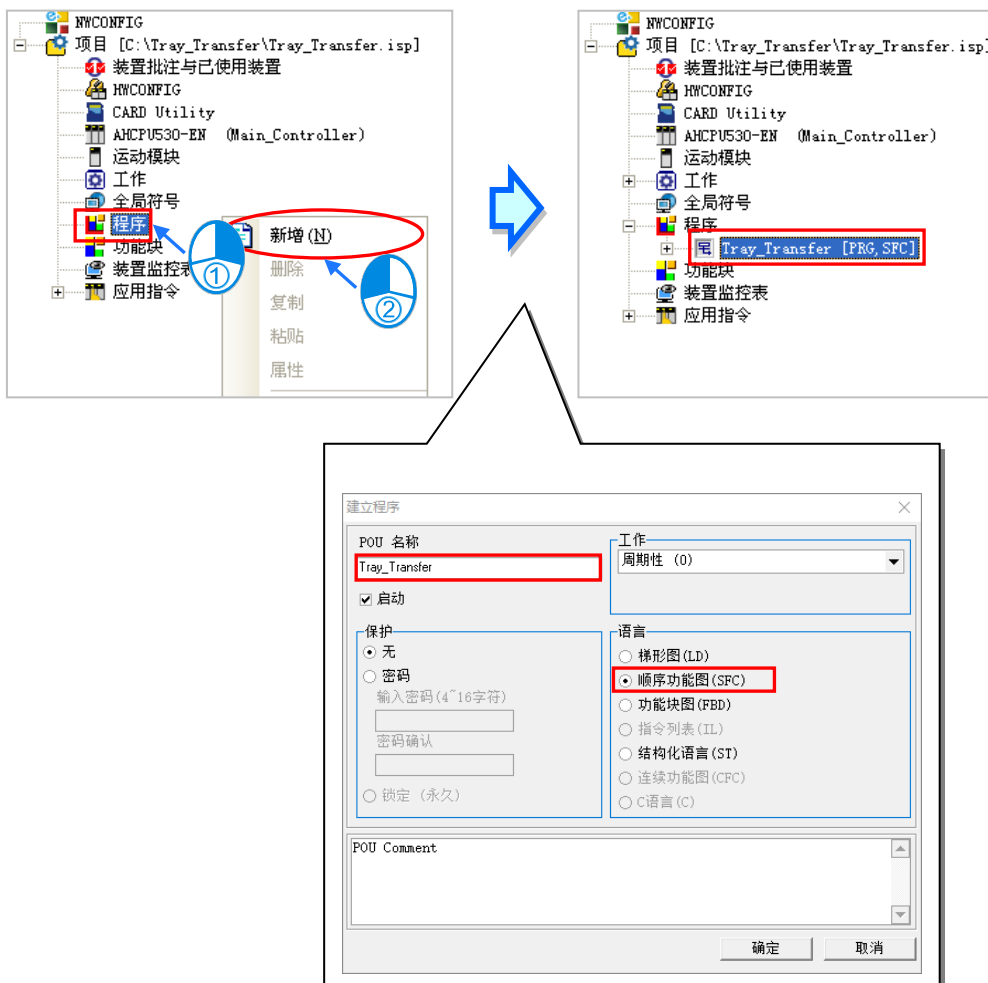
- (1) 系统的**初始程序**会先将所有的触发与 Ready 信号重置为 OFF，并等待系统的运转条件成立；而在本范例的规划中，当**运转标志**为 ON 且**错误标志**为 OFF 时，运转条件便会成立。
- (2) 一旦确定为运转状态后，系统会先将两侧的**夹具 (Y0.3 & Y0.4)** 推开，并确认两个夹具的**开启 Sensor (X0.3 & X0.5)** 为 ON。
- (3) 确认夹具已经开启后，系统会先延迟 1 秒，待状态稳定后便会分别送出入盘与退盘的 **Ready 信号 (Y0.0 & Y0.1)**，以通知与系统衔接的前后工作站。
- (4) 当入盘侧的动作完成后，系统便会收到**入盘完成信号 (X0.0)**，此时系统会先延迟 1 秒，待状态稳定后再将入盘区的**夹具 (Y0.3)** 关闭。
- (5) 当退盘侧的动作完成后，系统同样会收到**退盘完成信号 (X0.1)**，此时系统亦会先延迟 1 秒，待状态稳定后再将退盘区的**夹具 (Y0.4)** 关闭。
- (6) 当确认两侧夹具的**关闭 Sensor (X0.4 & X0.6)** 已全部为 ON 后，系统便会发出旋转轴的触发信号 (**Y0.2**) 以开始转送载盘；待旋转完毕后，系统便会收到旋转完成信号 (**X0.2**)，接着步序便会重新回到**初始程序**，而此时若运转条件仍然成立，则程序便会继续进行下一次的循环。

14.4.4 建立范例程序

首先请开启 ISPSOft 并建立一新项目，因本范例的硬件规划与第 4 章相同，因此请参考第 4 章的内容，并自行完成硬件规划的工作。



接下来请直接在项目管理区中新增一 SFC 的程序 POU。



再来请将已知的系统标志及 IO 配置建立在全局符号表中。

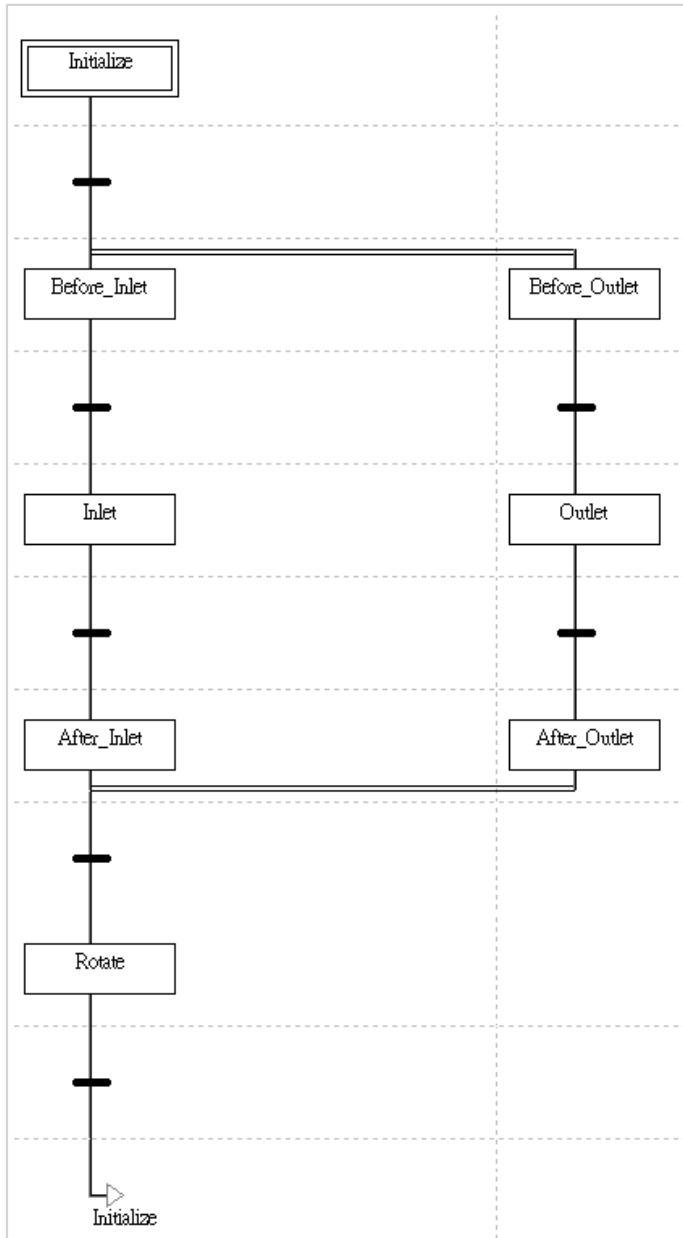
全局符号					
类型	符号名称	地址	数据类型	初始值	符号批注
VAR	IN_Done	X0.0	BOOL	FALSE	
VAR	OUT_Done	X0.1	BOOL	FALSE	
VAR	R_Done	X0.2	BOOL	FALSE	
VAR	Clip_1_OS	X0.3	BOOL	FALSE	
VAR	Clip_1_CS	X0.4	BOOL	FALSE	
VAR	Clip_2_OS	X0.5	BOOL	FALSE	
VAR	Clip_2_CS	X0.6	BOOL	FALSE	
VAR	Inlet_Rdy	Y0.0	BOOL	FALSE	
VAR	Outlet_Rdy	Y0.1	BOOL	FALSE	
VAR	Rotator_Trig	Y0.2	BOOL	FALSE	
VAR	Clip_1	Y0.3	BOOL	FALSE	
VAR	Clip_2	Y0.4	BOOL	FALSE	
VAR	RUN	N/A [Auto]	BOOL	FALSE	
VAR	ERROR	N/A [Auto]	BOOL	FALSE	

接着请在 SFC 程序的区域符号表中宣告各个步进点的变量符号。

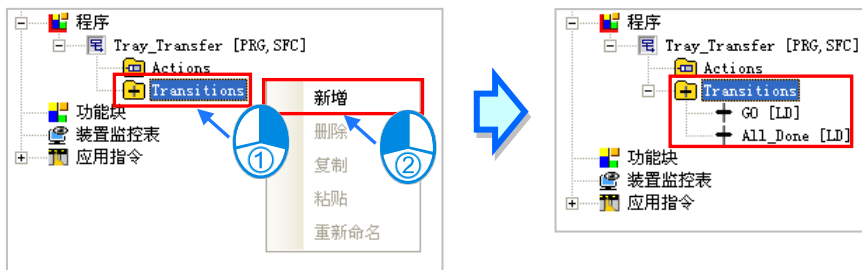
局部符号					
类型	符号名称	地址	数据类型	初始值	符号批注
VAR	Initialize	N/A [Auto]	STEP	FALSE	
VAR	Before_Inlet	N/A [Auto]	STEP	FALSE	
VAR	After_Inlet	N/A [Auto]	STEP	FALSE	
VAR	Before_Outlet	N/A [Auto]	STEP	FALSE	
VAR	After_Outlet	N/A [Auto]	STEP	FALSE	
VAR	Inlet	N/A [Auto]	STEP	FALSE	
VAR	Outlet	N/A [Auto]	STEP	FALSE	
VAR	Rotate	N/A [Auto]	STEP	FALSE	

接着参照下图建立 SFC 流程，而初始步进点的部份则请直接沿用预设的第一个步进点即可。

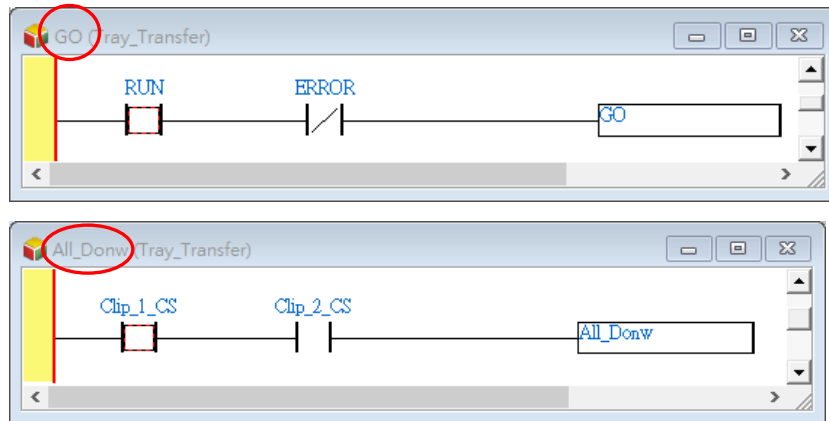
14



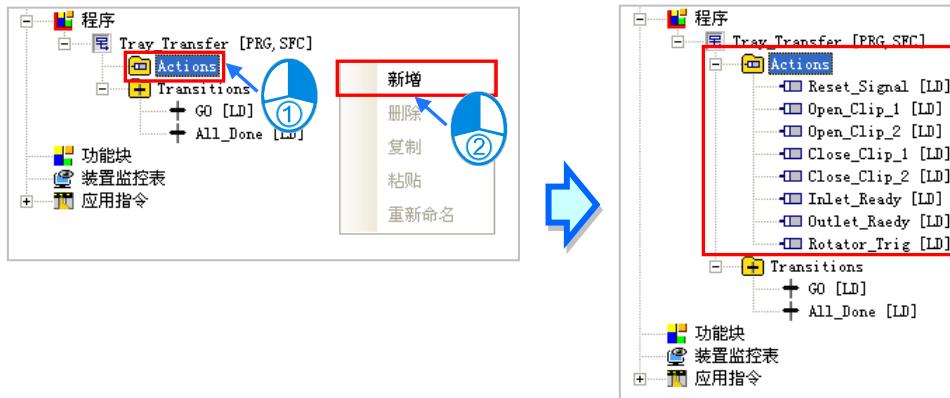
再来请至项目管理区中，「Transitions」项目下建立「GO」与「All_Done」两个 Transition。



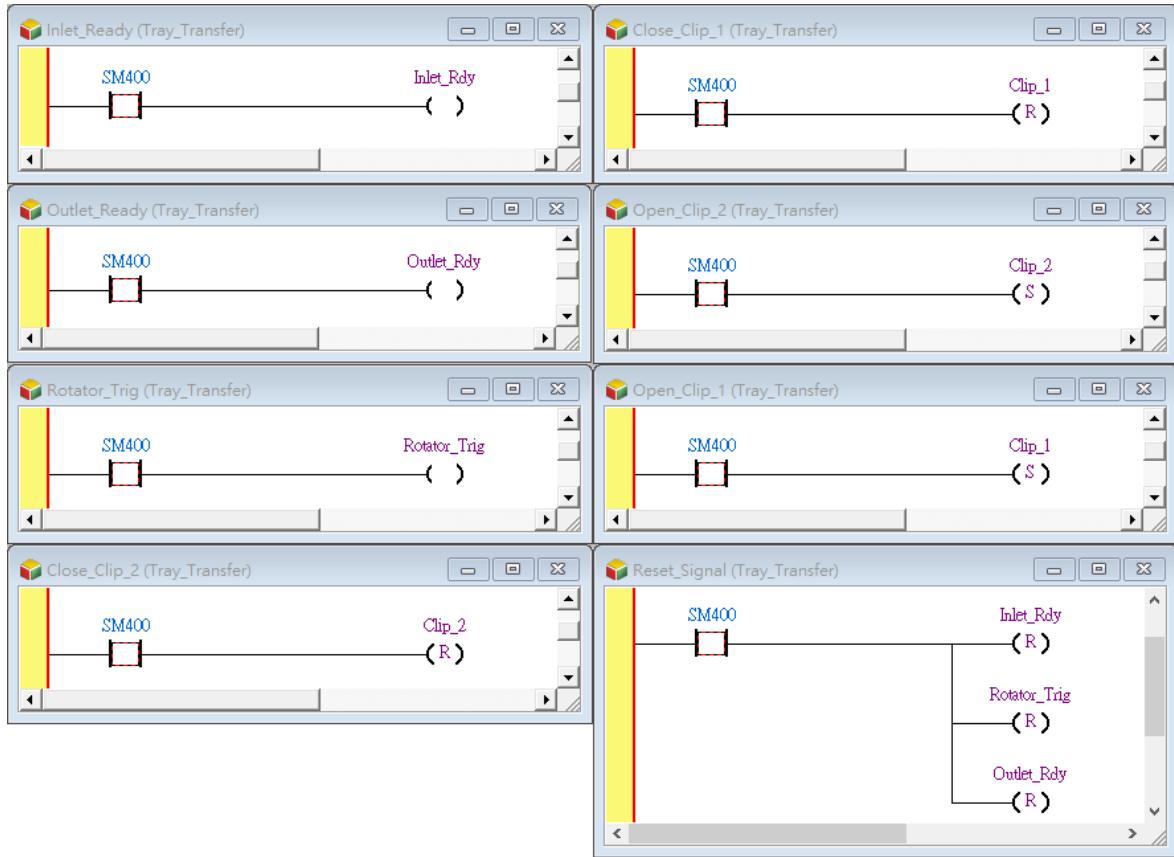
完成 **Transition** 项目的建立后，请逐一完成如下的 **Transition** 程序内容。



接着请再于 POU 的「**Actions**」项目下建立各个程序的 Action。

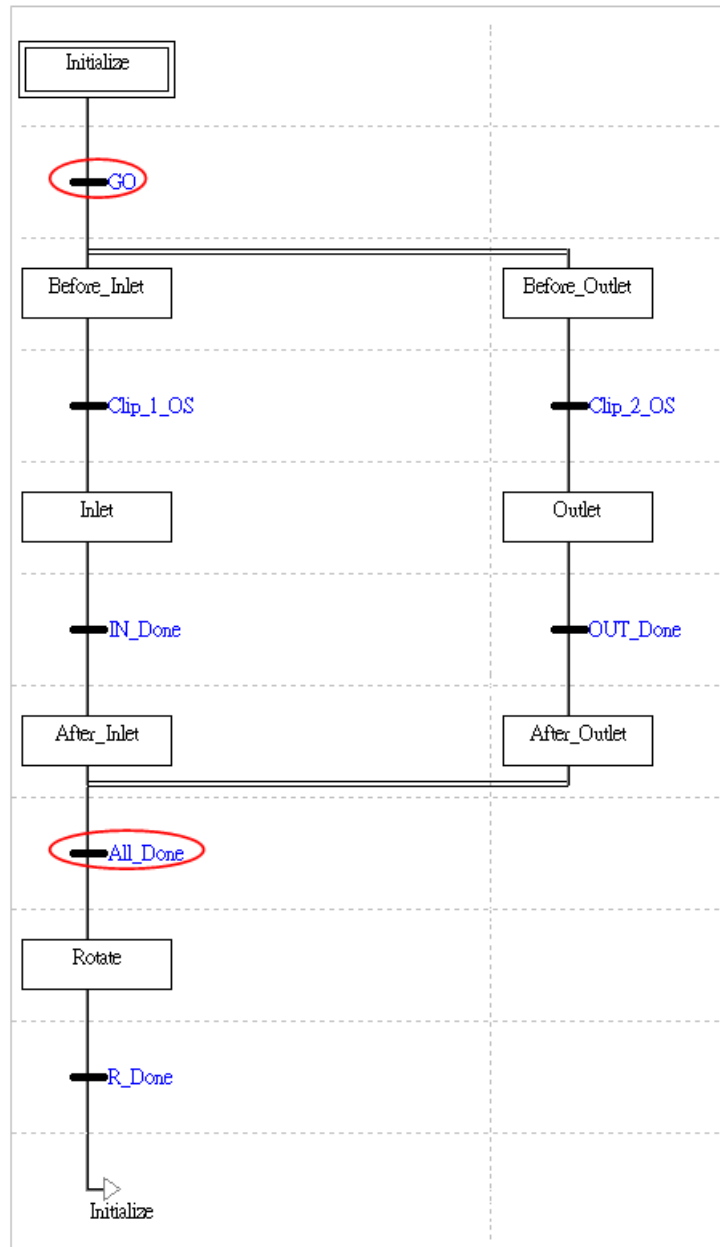


完成 **Action** 项目的建立后，请再逐一完成如下的 **Action** 程序内容。



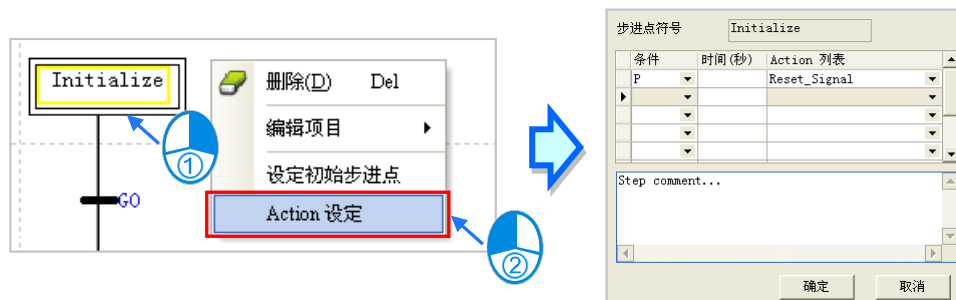
14

完成所有 **Action** 与 **Transition** 程序建立后，请于 **SFC** 的图形中逐一指定各个转换点，而其中除了「**GO**」与「**All_Done**」两个 **Transition** 之外，其余皆以之前建立的全局符号做为转换点的条件。其完成图形如下所示：

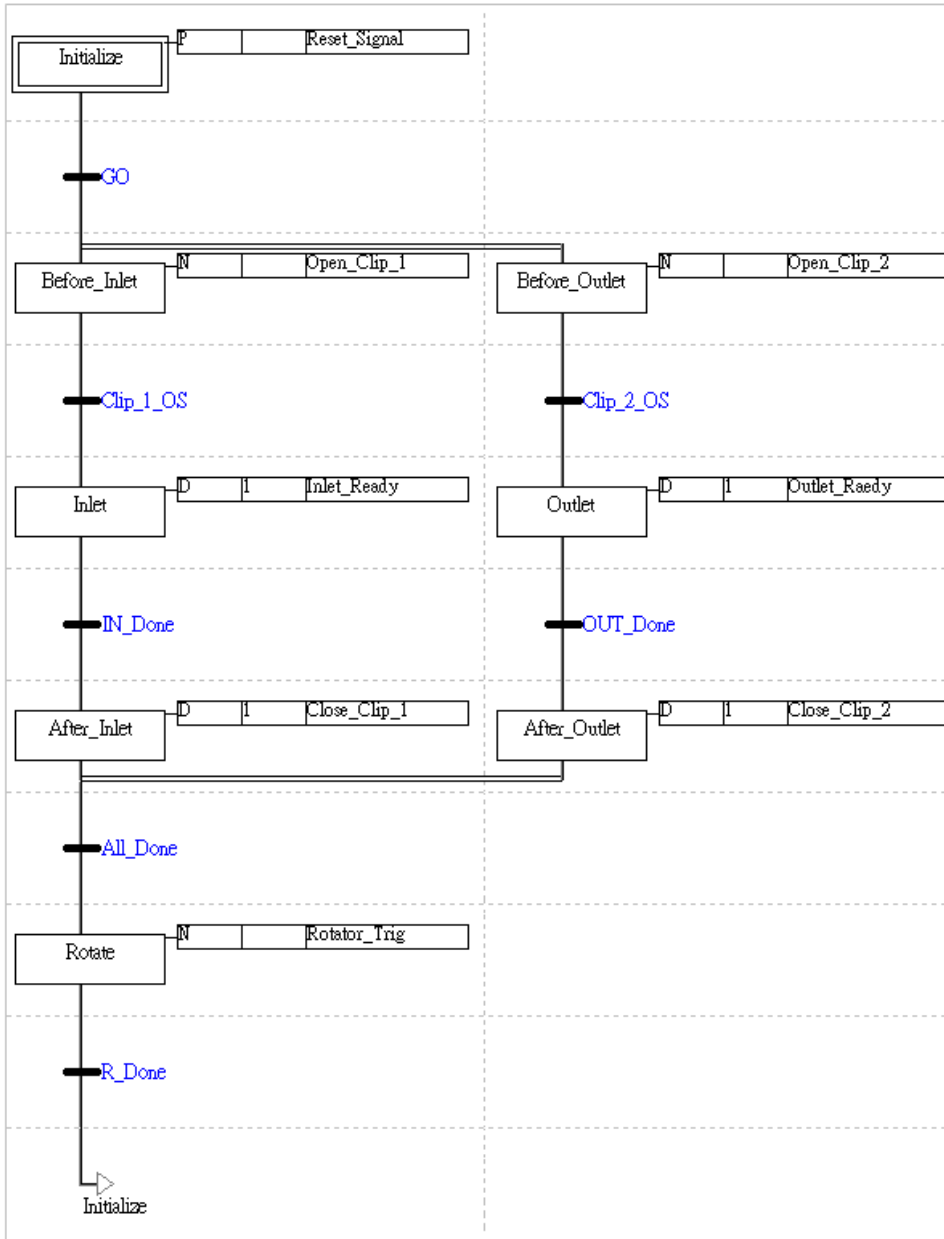


14

接下来请逐一于各个步进点中配置其对应的 Action，并指定各个 Action 的修饰条件。



最后完成的程序如下，接着请自行完成编译，并将硬件参数与程序下载至主机当中进行后续测试。



14

第15章 连续功能图 CFC

目录

15.1	连续功能图 (CFC) 简介	15-2
15.1.1	关于连续功能图	15-2
15.1.2	CFC 编程的注意事项	15-2
15.1.3	ISPSOft 的 CFC 编辑环境	15-2
15.1.4	CFC 的图示工具栏	15-4
15.1.5	CFC 编辑工作区的快捷选单	15-5
15.2	ISPSOft 中建立 CFC 程序.....	15-6
15.2.1	选取对象.....	15-6
15.2.2	输入节点、输出节点与逻辑闸.....	15-6
15.2.2.1	插入节点或门.....	15-7
15.2.2.2	新增与删除接脚.....	15-8
15.2.3	改变接脚的形式	15-10
15.2.4	物件的联机与取消.....	15-10
15.2.5	指令与功能块	15-11
15.2.5.1	插入指令	15-12
15.2.5.2	插入功能块	15-13
15.2.6	删除对象.....	15-15
15.2.7	编辑装置或符号	15-16
15.2.8	物件的致能与失效.....	15-17
15.2.9	插入批注.....	15-17
15.2.10	变更程序的执行顺序	15-20
15.2.11	显示设备批注	15-20

15.1 连续功能图 (CFC) 简介

15.1.1 关于连续功能图

连续功能图 (Continuous Function Chart, CFC) 为基于 IEC61131-3 所进一步延伸发展出来的 PLC 编程语言，连续功能图经常被使用在运动控制领域中，其最大的特色就是以类似逻辑设计图来表达程序的控制，相较于其他的 PLC 编程语言，连续功能图提供更明确的输入与输出的逻辑关系，及程序的执行顺序。

本章后续仅针对 ISPSOft 中的连续功能图操作与建立方式加以介绍，至于连续功能图的动作原理在此将不再多加叙述。

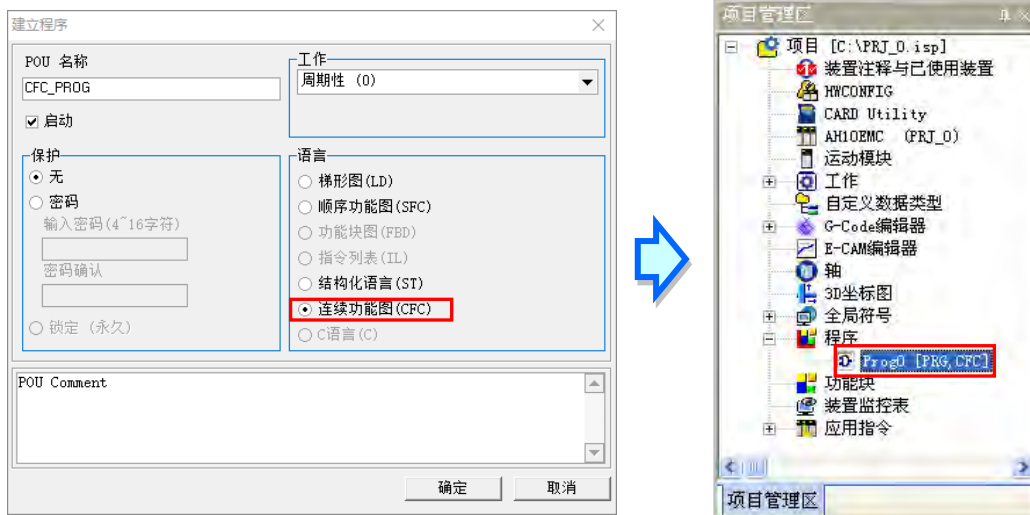
15.1.2 CFC 编程的注意事项

- 支持键盘输入，但不支持 LD、LDI、LDP、LDF 指令，可以直接输入寄存器或变量的名称，自动贴上 一个输入节点。
 - 键盘输入时，英文无大小写的分别，亦即 OUT、Out 与 out 的意义是相同的。
 - 常数的进制表示如下。
 - 10 进制：2345
 - 16 进制：16#5BA0
 - 浮点数：4.123
- *. 仍支持 K、H 表示法。

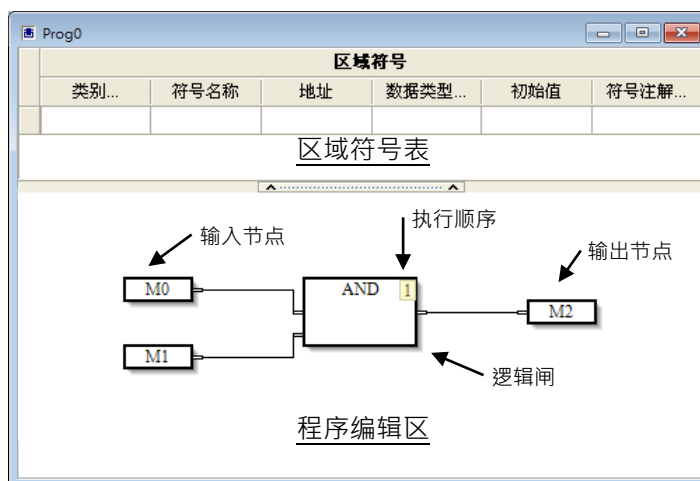
- CFC 语言可编辑的回路数并没有限制，但仍需考虑编译后的程序大小是否会超出主机的内存容量。
- 节点、逻辑闸、指令或功能块输入接脚只能来自一个来源，但输出接脚的目标可到多个。

15.1.3 ISPSOft 的 CFC 编辑环境

在建立 POU 时，于语言字段选择「连续功能图 (CFC)」便可建立连续功能图程序，请参考第 5.4.1 节的说明。



下图为 ISPSOft 中的 CFC 编辑工作区环境。其窗口的上方配置为区域符号表，用于宣告区域符号，而下方则是程序编辑区，用于编写 CFC 程序。连续功能图程序的对象包括了节点与逻辑闸等对象，对象右上角的编号代表程序的执行顺序。



15

15.1.4 CFC 的图示工具栏

进入 CFC 的编辑环境后，ISPSOft 的窗口中便会出现对应的编程图示工具栏，如下图，其功能简介如下表。



图示	键盘 (快捷键)	功能说明
	无	显示设备批注
	Shift + F9	输入节点
	Shift + F3	输出节点
	Shift + F10	批注
	Shift + F11	OUT 闸
	Shift + F1	AND 闸
	Shift + F2	OR 闸
	Shift + F12	开启/关闭功能块或指令的 EN 接脚
	Shift + F4	新增输入接脚
	Shift + F5	删除输入接脚
	Shift + F6	反相
	Shift + F8	正/负缘
	Shift + F7	设定/重置
	无	自动排列，将所有对象依据数据流排序执行顺序，再由上至下，由左至右依执行顺序排列。
	无	自动缩放，会将画面调整至看到所有对象的大小

15.1.5 CFC 编辑工作区的快捷选单

在程序编辑区中按下鼠标右键便会出现快捷选单如下图，相关功能如下表。



操作项目	功能说明
撤销	撤销刚才的编辑动作
重复	取消复原而重复刚才的动作
剪切	剪切选取的对象
复制	复制选取的对象
粘贴	将方才复制或剪下的对象贴在目前的编辑位置
删除	删除选取的对象
全选	全选程序编辑区所有对象
批注	在鼠标光标位置插入批注
执行顺序	设定逻辑闸指令或功能块的执行顺序；设为数据流，会依照对象之间的先后关系重新安排执行顺序；设为拓朴，会依照对象摆放的相对位置，由上至下，由左至右安排执行顺序。
致能/失效	切换对象是否被执行
En	开启/关闭功能块或指令的 EN 接脚
反向	切换接脚是否为反相
正/负缘	切换接脚的正/负缘状态
设定/重置	切换接脚的设定/重置状态

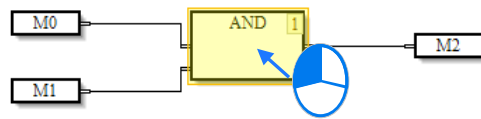
15.2 ISPSOft 中建立 CFC 程序

15.2.1 选取对象

CFC 语言的对象只要使用鼠标便可进行选取，选取后的对象便可进行如剪下、复制等各种编辑动作。

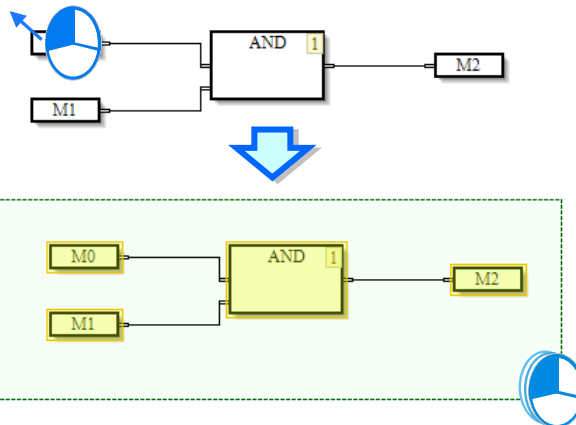
- 选取对象

将鼠标移至欲选取的对象上方后，点击鼠标左键即可。被选取对象的对象将会以橘色表示，未选取的对象则以白色表示。如下图。也可使用键盘的箭头键切换选取的对象。



- 多重选取

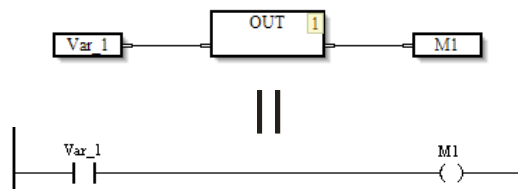
在空白处按住鼠标左键后，拖曳欲选取的范围，范围内的对象都将被选取，如下图。



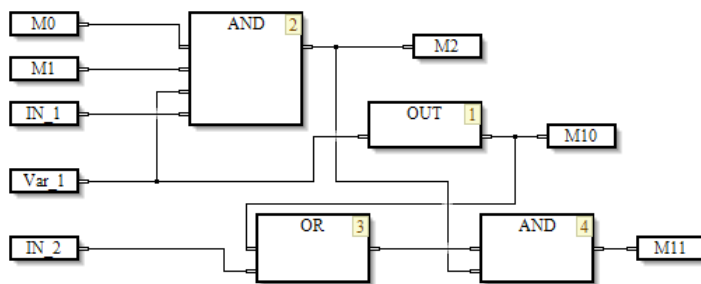
15

15.2.2 输入节点、输出节点与逻辑闸

CFC 图的原理是藉由输入节点输入来源的变量或寄存器的状态，藉由逻辑闸的运算后将状态输出到输出节点。但输入与输出节点不可直接相连，故若需要将状态进行搬移，必须使用 OUT 闸，如下图，为变量 Var_1 的状态搬移到寄存器 M1，及对应梯形图相同意义的程序。



其他的逻辑闸包括了 AND 闸与 OR 闸，藉由连接这些对象，便可设计各种复杂的回路，如下图。

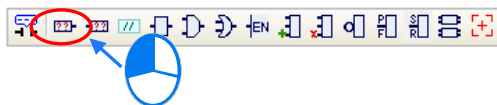


15.2.2.1 插入节点或门

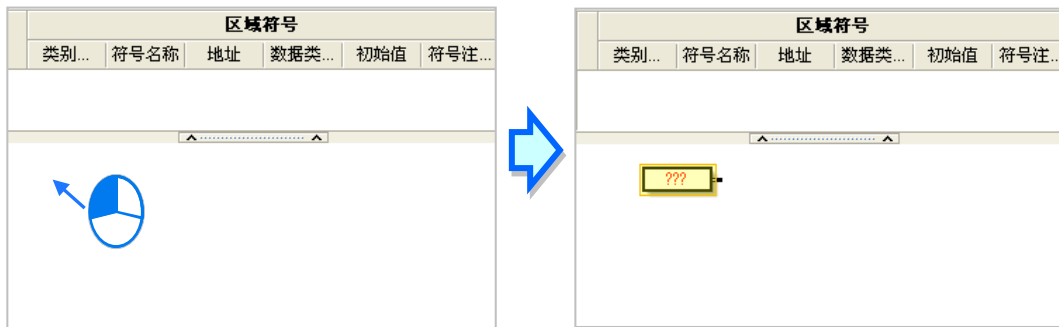
插入节点可用下列二种方式，请依使用习惯择一即可。

- 方法一：图示工具栏

(1) 点击图标工具栏上要插入的按钮。如下图。

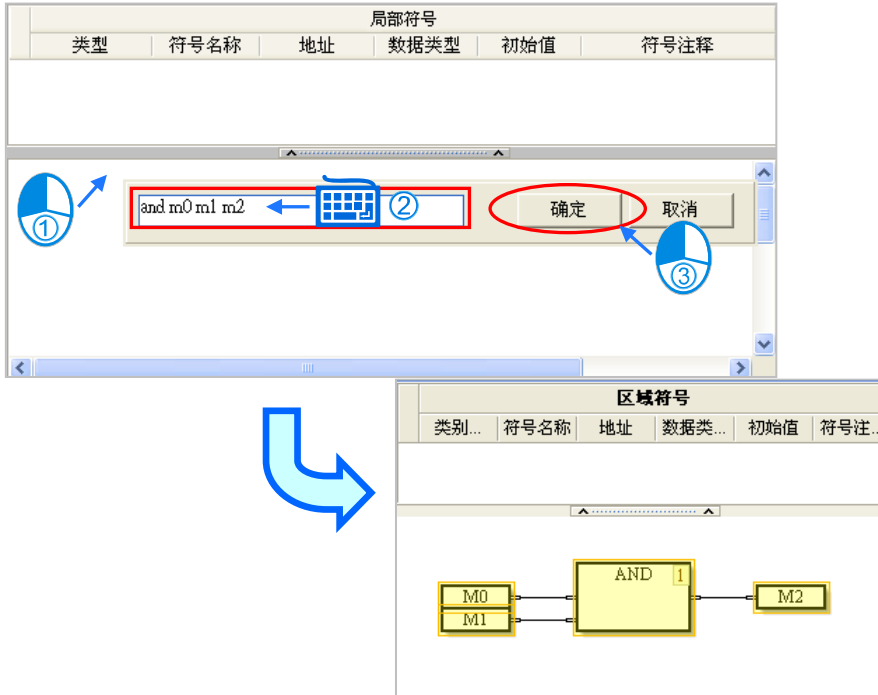


(2) 在程序编辑区要放置的位置点击鼠标左键，便完成对象的插入。如下图。



- 方法二：键盘输入

(1) 先用鼠标左键点选程序编辑区空白处，再直接用键盘输入指令后，画面便会自动出现指令编辑窗口，待输入完毕后按下键盘的「Enter」键，或以鼠标点击窗口的「确定」钮，此时移动鼠标光标，在程序编辑区需要插入指令的地方按下鼠标左键，便在此处完成指令的建立（所输入的指令无须分大小写，且若输入不正确的应用指令或对应装置，按确定键时会有错误讯息）。如下图。



键盘输入支持的指令如下表。

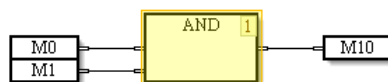
15

键盘输入指令	功能说明
OUT	产生一个 OUT 闸
AND	产生一个 AND 闸
OR	产生一个 OR 闸
寄存器名称	产生一个此寄存器的输入节点
指令名称	产生一个此指令的方块
功能块名称	产生一个此功能块的方块

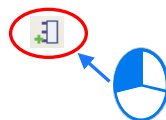
15.2.2.2 新增与删除接脚

AND 与 OR 闸可使用以下步骤插入及删除输入接脚，便于多输入条件时的设计。

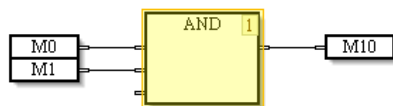
(1) 鼠标点选逻辑闸，使之成为被选取的状态。如下图。



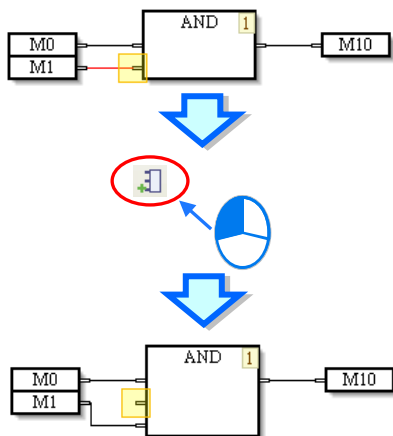
(2) 此时图示工具栏的「新增输入接脚」按钮便会亮起，使用鼠标左键点击。如下图。





(3) 此逻辑闸便会在最下方新增一个输入接脚，如下图。

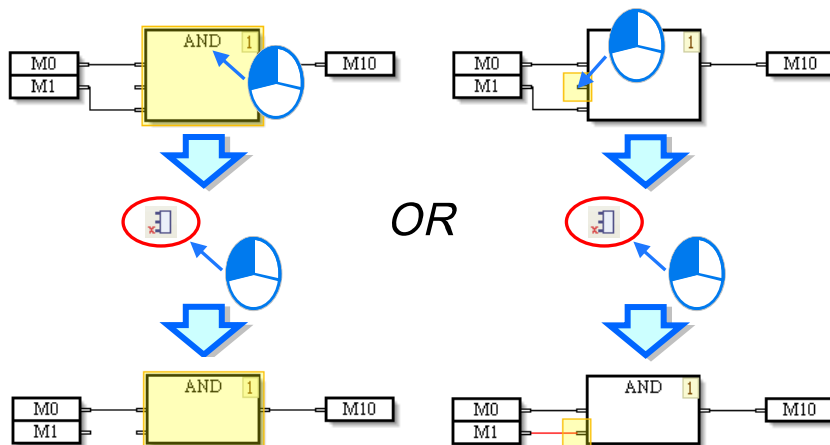


(4) 另一方面，若先选择某个输入接脚，再按下「新增输入接脚」按钮后，则会在选择的输入接脚上方插入一个输入接脚，如下图。




(5) 当需要删除接脚时，可以选择逻辑闸后按下图标工具栏的「删除输入接脚」按钮，此时会删除逻辑闸最下方的输入接脚；或着也可直接点选某个输入接脚后按下「删除输出接脚」按钮，删除该接脚。

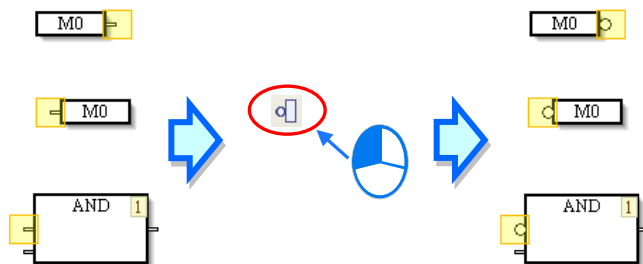
15




15.2.3 改变接脚的形式


可藉由以下的方式改变接脚的形式，以完成复杂的 CFC 程序设计。

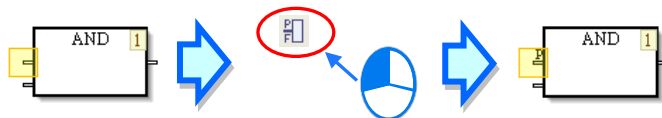
- (1) 若需要将输入或输出的状态取反相时，使用鼠标点选节点、逻辑闸、功能块、指令等等对象的接脚，使之成为被选取的状态，再按下图示工具栏或是右键快捷选单的「反相」按钮 ，便可使之进行状态的反相。如下图。若选取反相的接脚再次按下反相按钮后，便取消反相的设定。



- (2) 若需要对输出接脚的状态进行设定或重置的设计时，选取输出节点的接脚，再按下图示工具栏或是右键快捷选单的「设定/重置」按钮 ，便可将此接脚进行设定或重置。如下图。若在同一接脚多次按下此按钮，则会将此接脚依序在设定、重置、取消设定之间变换。



- (3) 需要对接脚的状态进行上升沿触发或下降沿触发的设计时，选取逻辑闸、功能块、指令等等对象的接脚，再按下图示工具栏或是右键快捷选单的「上升/下降沿」按钮 ，便可将此接脚进行正缘或负缘触发的设定。如下图。若在同一接脚多次按下此按钮，则会将此接脚依序在正缘、负缘、取消设定之间变换。



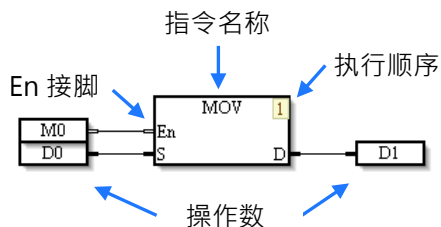
15.2.4 物件的联机与取消

欲连接已插入的对象，在其中一个对象的接脚按住鼠标左键不放，此时可以联机的目标接脚将会以灰色显示，鼠标移动到另一个组件的最接近的接脚会出现箭头符号，放开鼠标左键将连接此二接脚。如下图。反之若欲变更或取消对象的联机时，将线段拖曳到新的接脚或空白处即可。

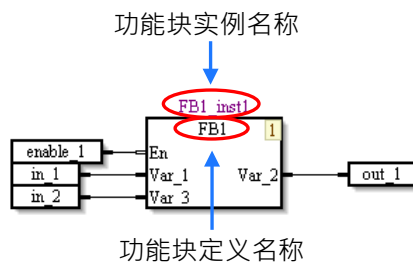


15.2.5 指令与功能块

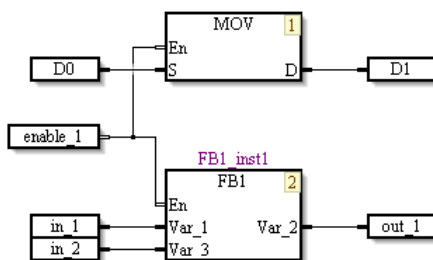
在 ISPSOft 的连续功能图中，各种指令以方块图标呈现，方块中列出指令名称及操作数，右上角为 CFC 程序执行顺序，另外透过 En 接脚链接前方的逻辑状态，当 En 接脚的逻辑状态为 ON 时，指令才会执行。如下图。




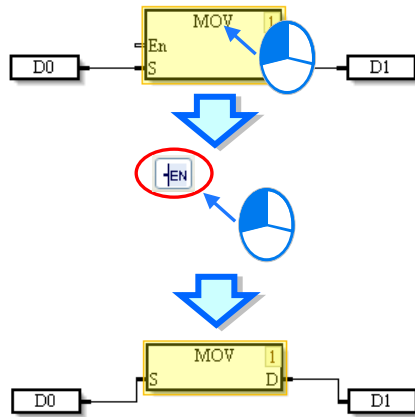
下图则是 ISPSOft 连续功能图中的功能块符号，同样透过 En 接脚链接前方的逻辑状态，当前方的逻辑状态为 ON 时，功能块才会执行。功能块符号上方的文字为该功能块的符号名称，也就是**功能块实例**的名称，关于功能块的详细介绍请参考手册中**第 7 章**的相关说明。如下图。



当需要透过同一个条件启动多个指令或功能块时，可以将此条件的输入节点连接到各个指令与功能块的 En 接脚，当此输入为 ON 所连接的对象都将依照执行顺序执行，如下图的 enable_1 输入，同时控制了图中的指令与功能块。



在不需要条件控制的情况，可以选取指令或功能块后按下图标工具栏或右键选单的「En」按钮 ，将 En 接脚关闭，此时 CFC 程序执行到此执行顺序时皆会执行此指令或功能块，如下图。若再次按下此 En 按钮，则会将此对象的 En 接脚再次开启。

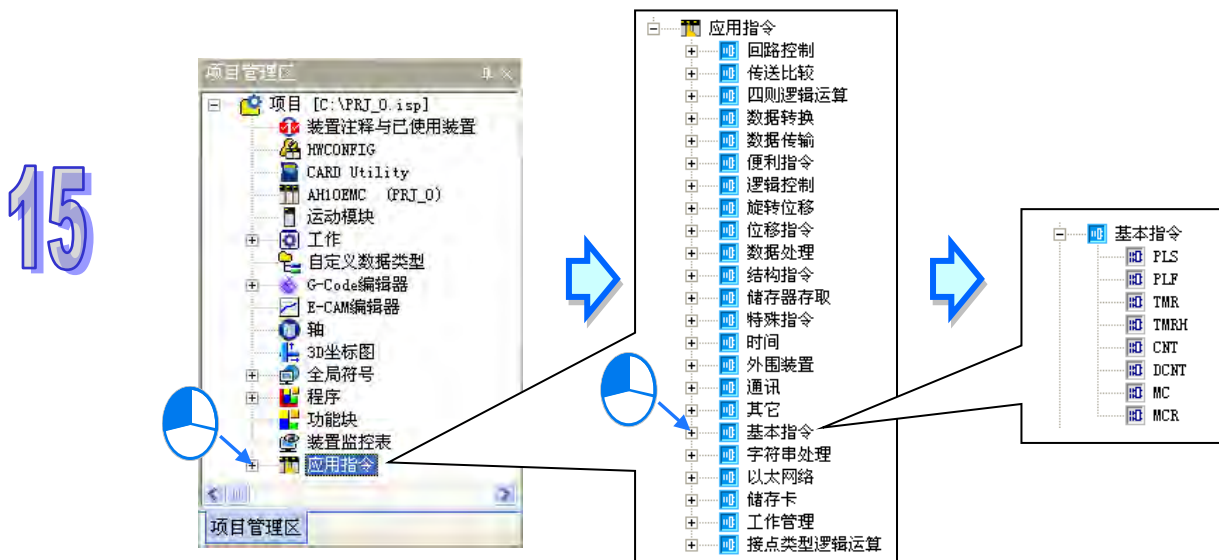


15.2.5.1 插入指令

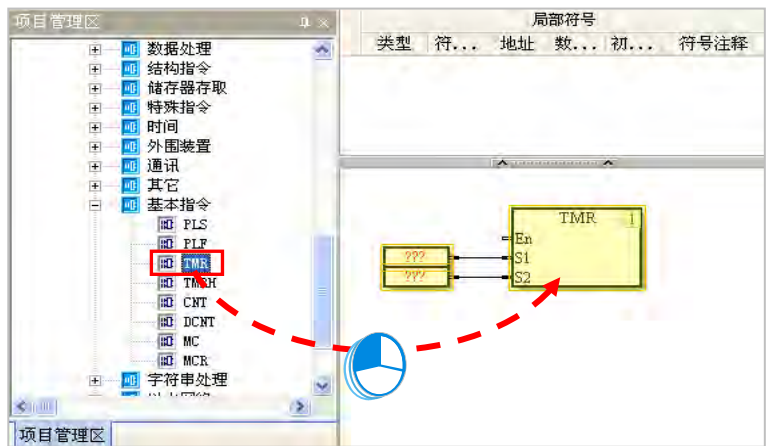
插入指令可用下列两种方式，请依使用习惯择一即可。

- 方法一：拖曳指令

(1) 展开应用指令项目，找到欲加入的指令。如下图。

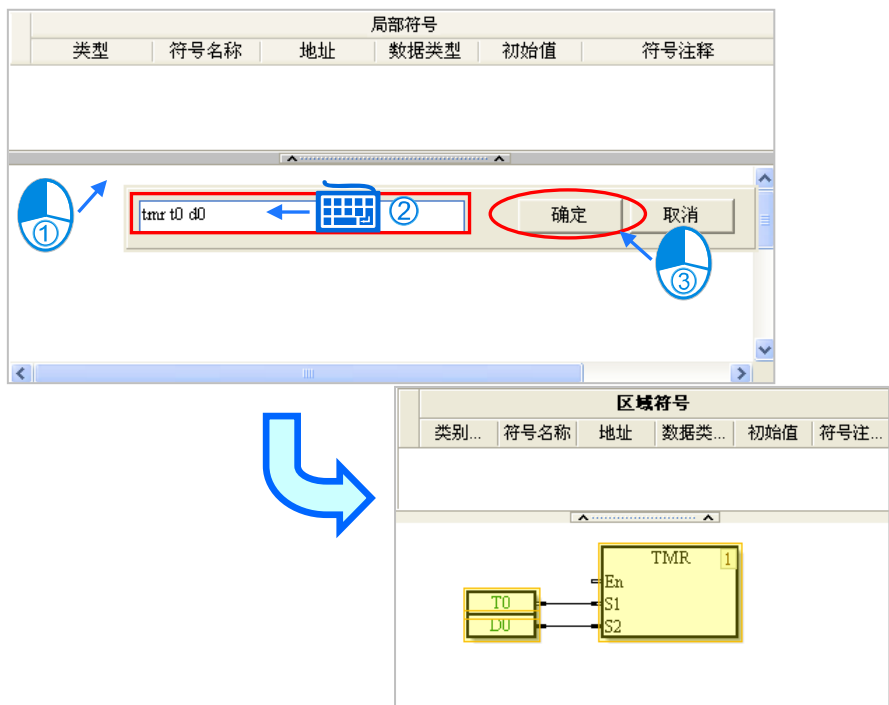


(2) 选择要加入的指令按住鼠标左键，拖曳到程序编辑区，在程序编辑区要插入指令的地方放开鼠标左键，便在此处完成指令的建立。如下图。



• 方法三：键盘输入

先用鼠标左键点选程序编辑区空白处，再直接用键盘输入该指令后，画面便会自动出现指令编辑窗口，待输入完毕后按下键盘的「Enter」键，或以鼠标点击窗口的「确定」钮，此时移动鼠标光标，在程序编辑区需要插入指令的地方按下鼠标左键，便在此处完成指令的建立(所输入的指令无须分大小写，且若输入不正确的应用指令或对应装置，按确定键时会有错误讯息)。如下图。



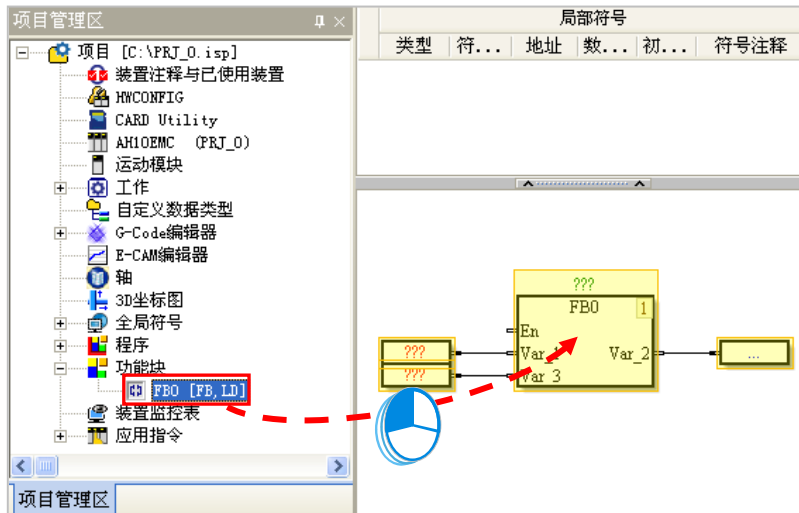
15

15.2.5.2 插入功能块

插入功能块的方法有二种，请依使用习惯择一即可。

• 方法一：拖曳新增

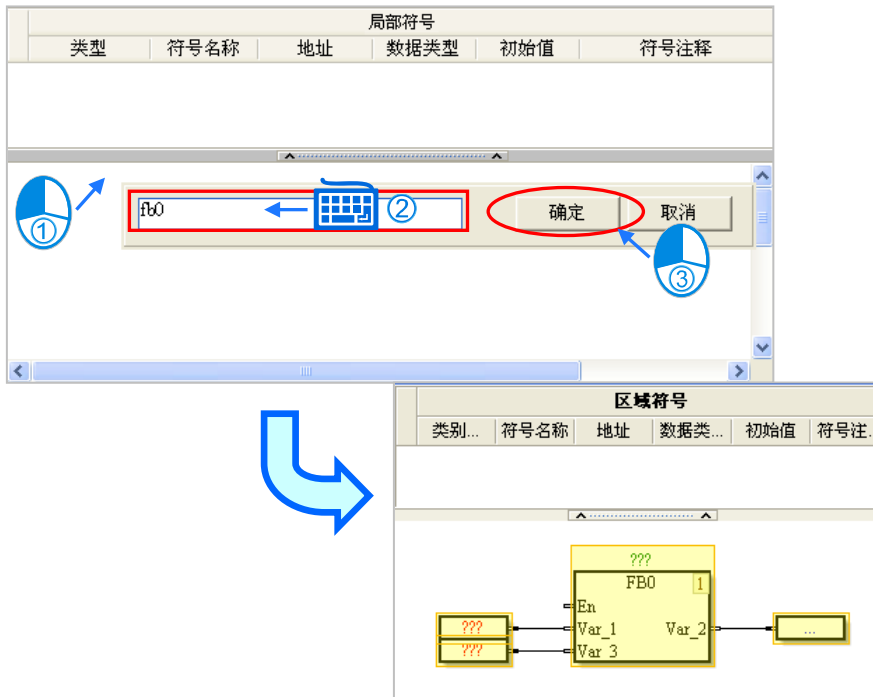
点选欲插入的功能块定义项目，按住鼠标左键并将它拖曳至程序编辑区中欲插入的位置后放开，便在此处完成功能块的建立，如下图。



• 方法二：手动输入


先用鼠标左键点选程序编辑区空白处，再直接用键盘输入该功能块定义的名称后，画面便会自动出现指令编辑窗口，待输入完毕后按下键盘的「Enter」键，或以鼠标点击窗口的「确定」钮，此时移动鼠标光标，在程序编辑区需要插入功能块的地方按下鼠标左键，便在此处完成功能块的建立（所输入的功能块定义名称无须分大小写，且若输入不正确的功能块定义名称或对应装置，按确定键时会有错误讯息）。如下图。

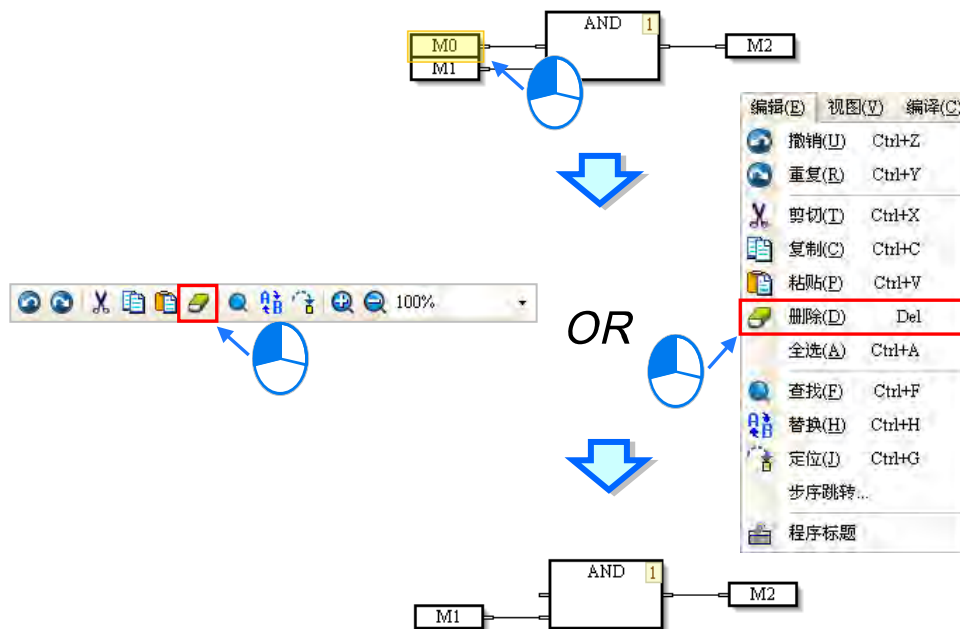
15



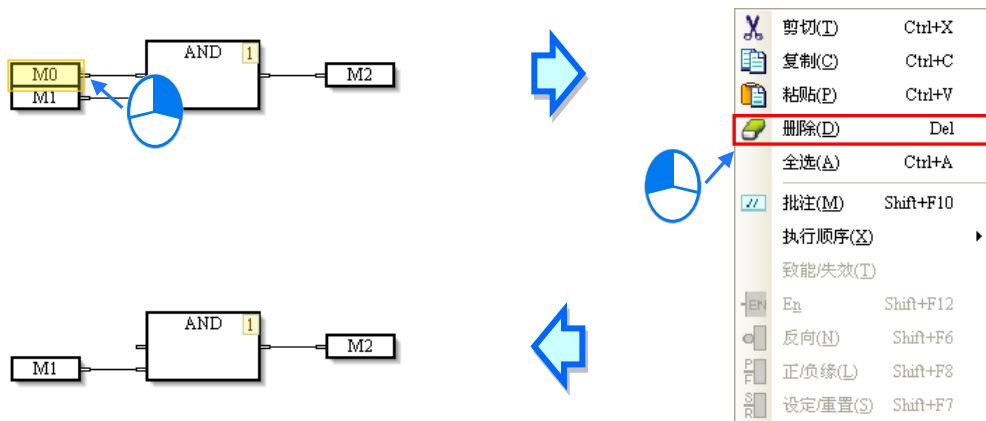
15.2.6 删除对象

当需要删除连续功能图上的单一或多个对象时，可依下列方式删除：

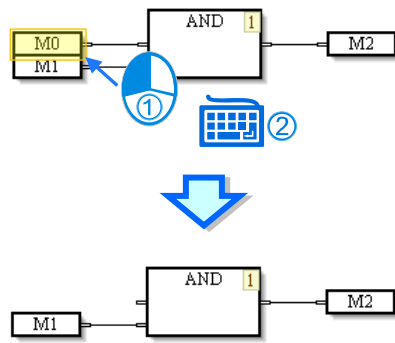
(1) 选取该对象，于功能工具栏中点选 **编辑 (E) > 删除 (D)** 或点击图示工具栏  按钮。如下图。



(2) 选取该对象，按下鼠标右键后，于快捷选单点选「**删除 (D)**」。如下图。



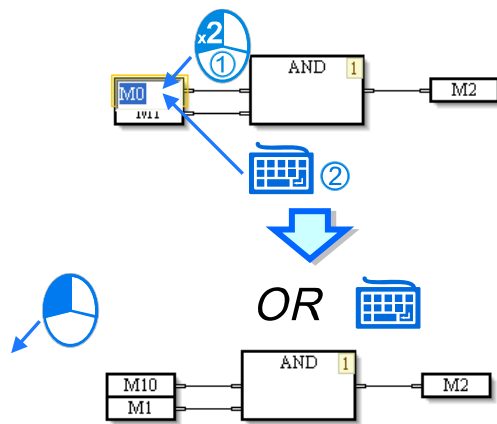
(3) 选取该对象，按下键盘「Delete」键，即可删除。如下图。



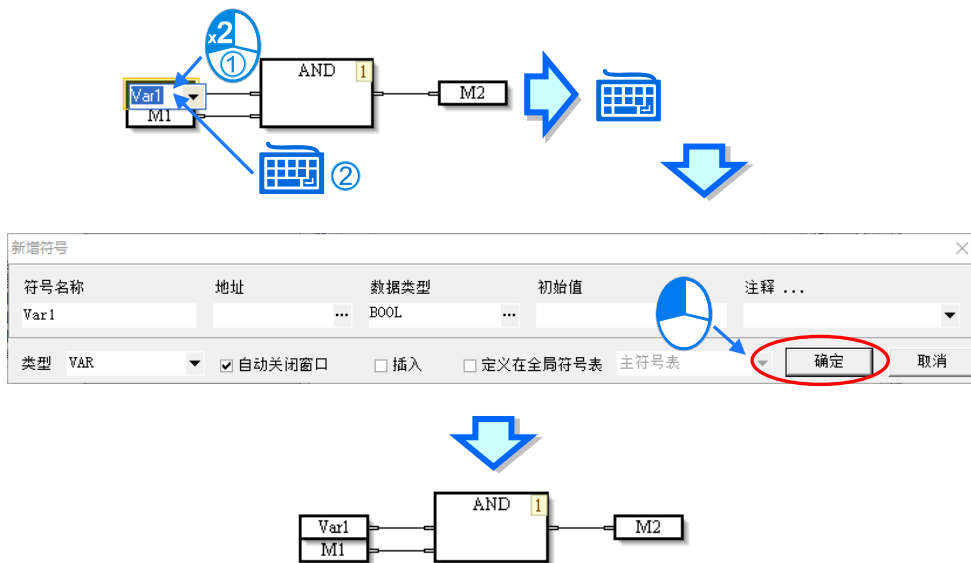
15.2.7 编辑装置或符号

以鼠标左键双击欲编辑的节点，直接输入欲修改的装置名称，再点选程序编辑区空白处或按下键盘「Enter」键即可，如下图。

15



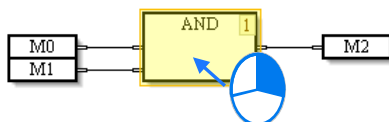
若于节点输入变量符号，按下键盘的「Enter」键后，将出现新增符号窗口，完成设定后按下「确定」，便完成变量符号的宣告，如下图，变量符号相关规定请参阅第6章。



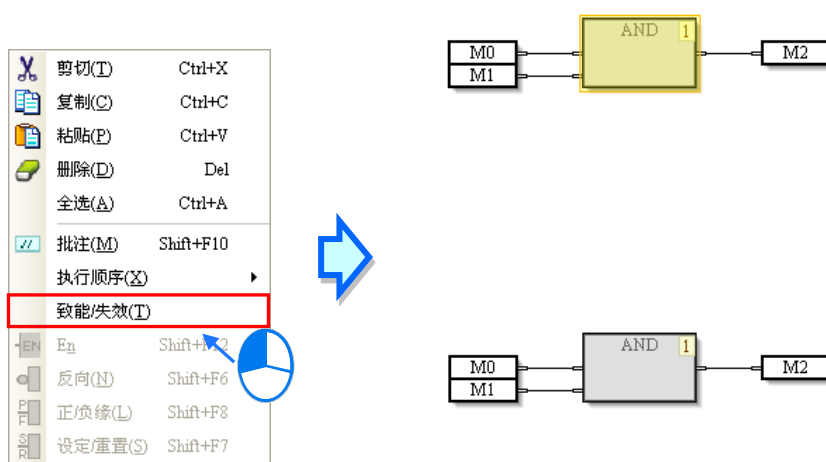
15.2.8 物件的致能与失效

若某个对象被设为失效状态，则当程序进行编译时，便会主动略过此失效对象的执行顺序，因而此功能可被用来暂时关闭程序中的某些程序。

(1) 在欲失效的对象上点选鼠标右键。如下图。



(2) 于跳出的快捷选单选择「致能/失效 (I)」，该对象会以暗色表示。如下图。



(3) 若须回复为致能对象，则重复上述步骤，于跳出的快捷选单选择「致能/失效 (I)」即可。

15.2.9 插入批注

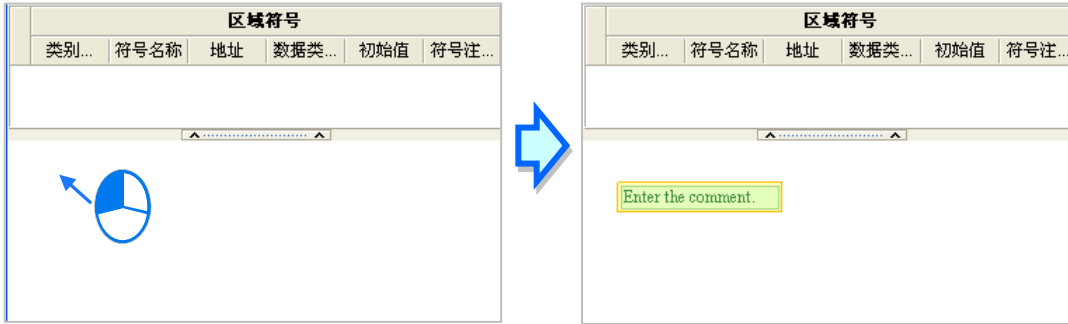
CFC 编辑器提供批注功能供用户使用。共有两种使用批注的方式说明如下。

• 方法一：在任意位置插入批注

(1) 点选图标工具栏的「批注」按钮。如下图。



(2) 在程序编辑区要放置的位置点击鼠标左键，便完成对象的插入。如下图。



(3) 双击插入的批注对象，便开启编辑窗口，于此输入批注的说明内容。如下图。若欲换行时请按下键盘的「Shift+Enter」键。

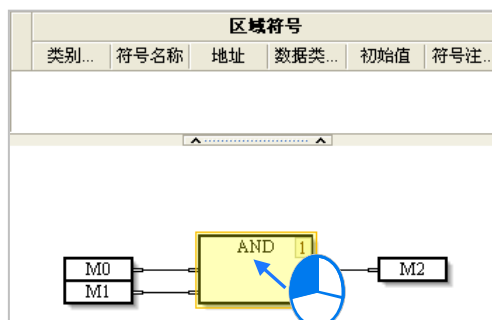


15 (4) 再点选画面空白处或按下键盘的「Enter」键，完成批注的输入。如下图。此类批注为独立于其他对象，不因其他对象的新增删除或移动而改变。

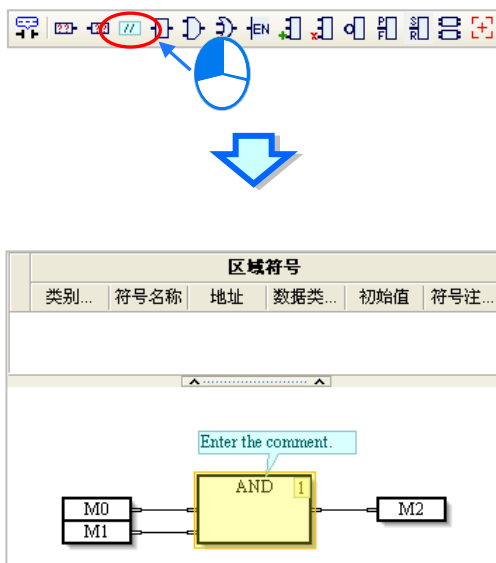


● 方法二：插入物件的批注

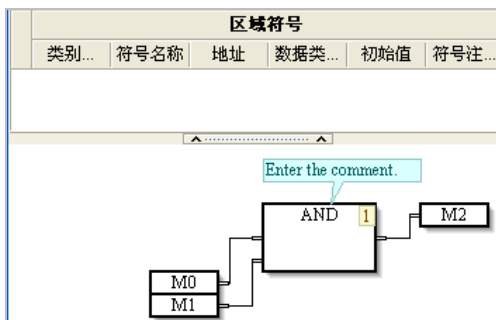
(1) 先选取要插入批注的对象。如下图。



- (2) 再点选图标工具栏的「批注」按钮，会出现一个连接到此对象的批注，如下图。同样的可以于此批注对象输入批注的说明内容。

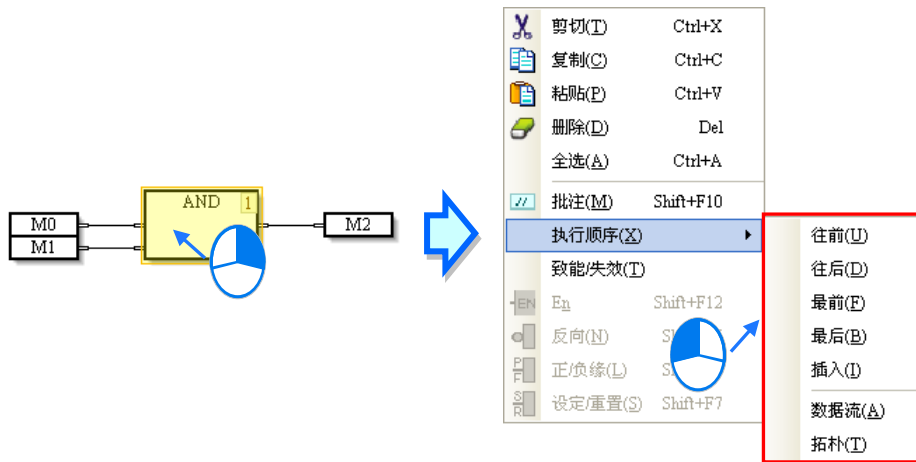



- (3) 此类的批注会随被选取对象的搬移而改变位置，当用户搬移了此对象，连接其上的批注也将随之移动。如下图。



15.2.10 变更程序的执行顺序


CFC 程序依据对象右上角的编号顺序执行，当需要变更某对象的执行顺序时，在此对象点击右键，开启快捷选单，展开「**执行顺序 (X)**」选单，设定对象的执行顺序，如下图；设为「**数据流**」时，会依照对象之间的先后关系重新安排执行顺序；设为「**拓朴**」时，会依照对象摆放的相对位置，由上至下，由左至右安排执行顺序。

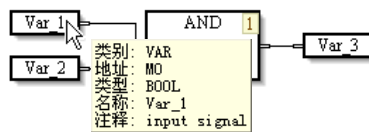


! 变更执行顺序会导致输出结果改变，如不了解顺序建议使用工具栏上的自动排列  按钮。

15

15.2.11 显示设备批注

点选「**显示设备批注**」按钮 ，成为按下的状态后，将鼠标移至装置或符号的文字上方便会显示该对象的相关信息。信息的内容包含地址与批注等。如下图。



第16章 C 语言

目录

16.1 C 语言编辑 (C LANGUAGE)	16-2
16.1.1 C 语言之架构说明	16-2
16.1.2 C 语言编辑环境说明	16-2
16.1.3 C 语言编译与除错	16-9
16.1.4 内部函数库指令说明	16-10
16.1.5 书签与注解	16-11
16.1.6 范例操作说明	16-13

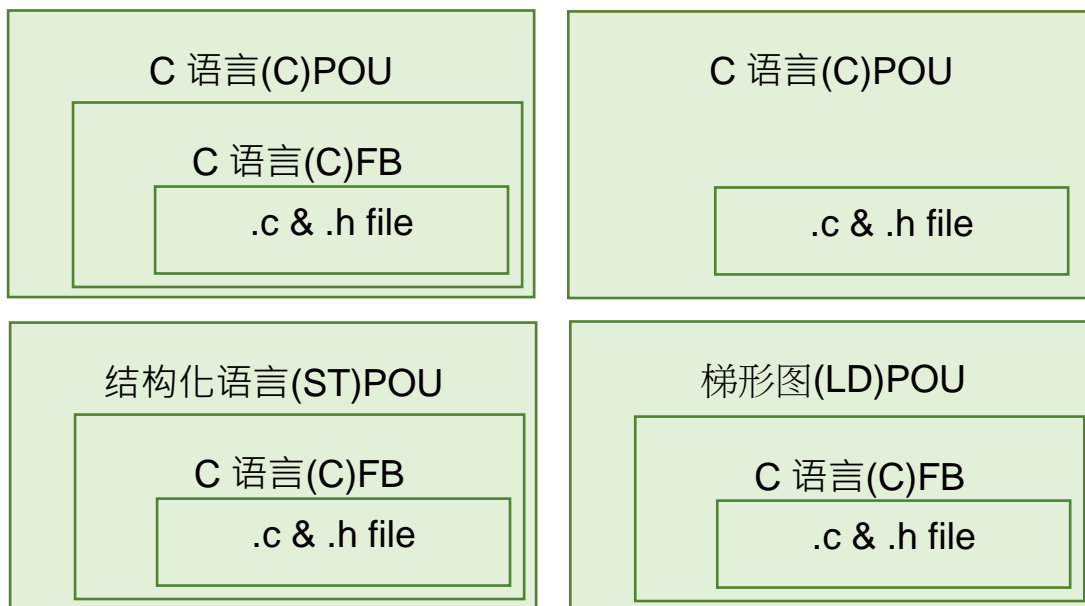
16.1 C 语言编辑 (C Language)

C 语言的特色在于整个程序是由许多的文字叙述所建构而成，而相较于指令列表 (IL)，C 语言则具有较高的可读性，且可通过类似数学方程式的文法来表达一个复杂的表达式，因此对于一个需要进行繁复运算的程序来说，C 语言是较适合此种应用场合的开发语言。

*下载 C 语言的程序到主机之前，必须确认 AS 主机韧体是否为 V1.06.00 (含) 以上。

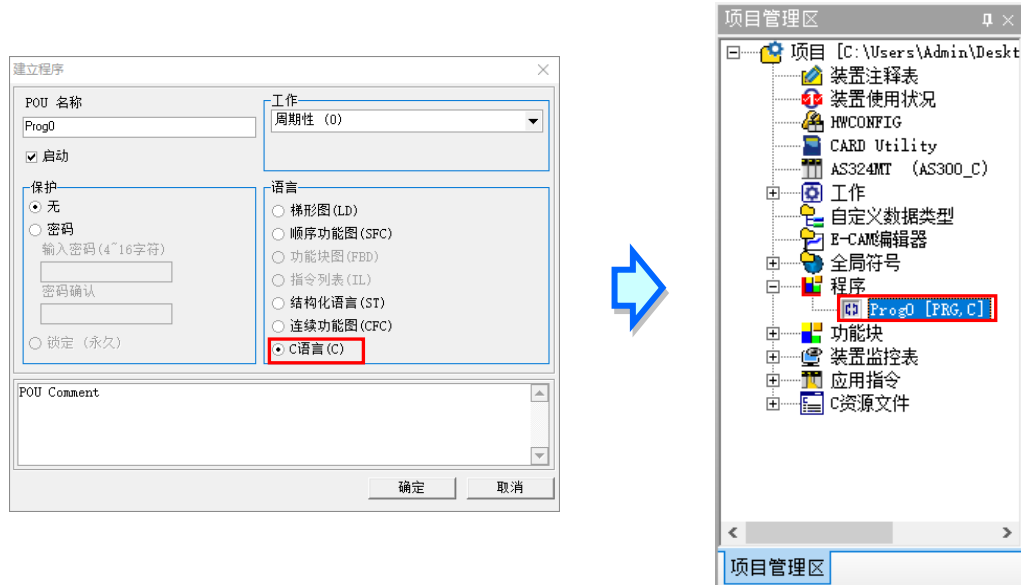
16.1.1 C 语言之架构说明

此编辑语言适合已熟悉 C 语言编辑之程序设计者，如欲开始采用 C 语言编辑前，请参考下图 C 语言所支持的编辑语言架构；用户可先使用市面上常见之 DEV-C、TurboC 以及 Visual Studio 等编辑软件进行编辑、编译与测试，之后在使用 ISPSOft 提供之导入功能，导入已经完成之 C 语言“.C”与“.h”文件并使用在 C 语言 (C) FB 或 C 语言 (C) POU，如下图，也可以在结构化语言 (ST) POU 或梯形图 (LD) POU 引用 C 语言 (C) FB。



16.1.2 C 语言编辑环境说明

AS 系列主机提供的 C Code 执行码储存空间 (ROM) 为 64K Bytes，以及变量宣告可用空间 (RAM) 为 8K Bytes。以上两个储存空间与现有的 PLC 程序使用空间皆各自独立，每次编译完成之后，建议从编译信息框内关注剩余空间。运用 ISPSOft 提供的 C 编辑画面，进行 C 语言程序编写，建立 POU 时，于语言字段选择「C 语言 (C)」便可建立 C 程序。

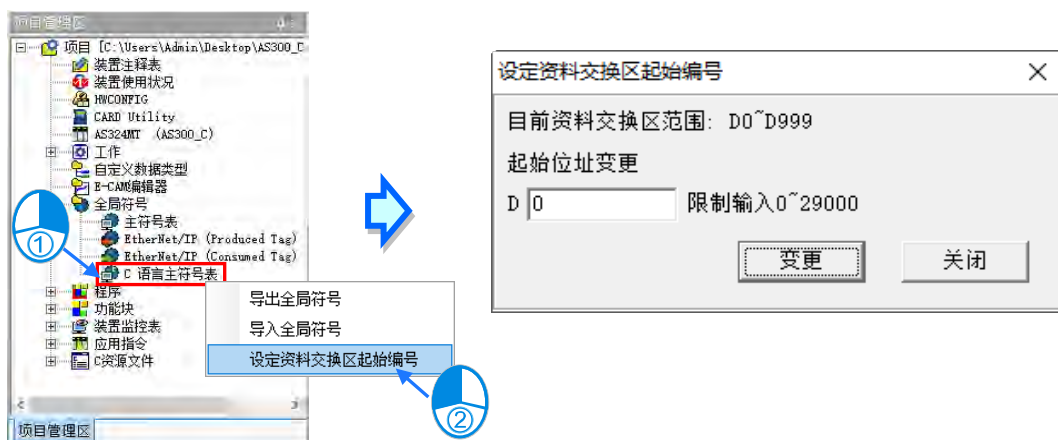


16

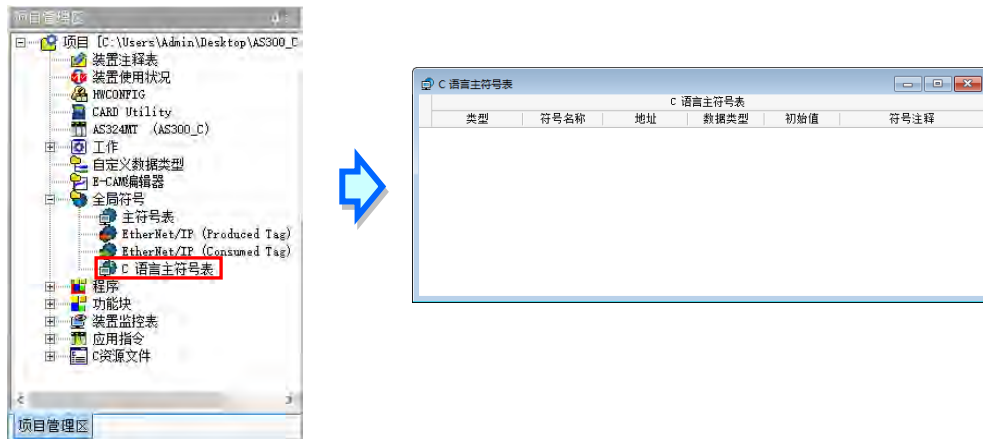
在 C 的编辑工作区中，只支持标准 C 语言写法，其它如：PLC 指令、C++或 Visual C...等高阶语法不支持。其操作方式与一般的文本编辑器类似，编辑时只要依据 C 语法的格式，直接使用键盘在工作区上面输入或修改文字即可，且当按下键盘的【Enter】键时便可执行换行的动作

```
0019 C_Var_0 = 123;
0020
0021
```

进行变量宣告时，可使用矩阵 (Array)，但须注意变量可宣告之内存有限，因此不可超出范围使用，否则严重可能会造成 PLC 当机之状况。必须在全局符号的 C 语言主符号表，新增变量符号，在 C 语言符号可使用的装置地址有 1000 个 D 装置，可以自行设定起始地址范围，在「C 语言主符号表」，单击鼠标右键之后再于快捷选项中单击「设定资料交换区起始编号」便可设定数据交换区的起始位置，起始地址可从 D0~D29000 填一个，系统会自动往后面设定连续 1000 个 D 装置，设定完成会显示在目前数据交换区范围。



接着双击「C 语言主符号表」会弹出「C 语言主符号表」窗口。

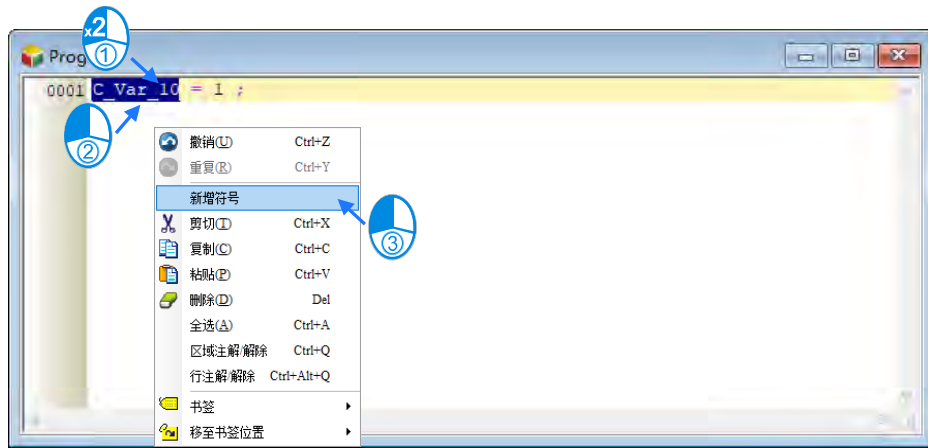


16

在 C 语言主符号表的空白处，使用鼠标单击右键，之后再于快捷选项中单击「新增符号」便可自动导出变量符号的宣告窗口。C 语言变量符号只支持 **BOOL**、**WORD**、**DWORD**、**REAL** 四种类型，地址起始地址为 C 语言数据交换区所设定的范围，下图为 D0 宣告完之后的变量符号可以直接在 C 语言的程序内使用，也相当于一个全局变量可以在其它程序语言被使用。



在编辑的过程中输入了未经宣告的变量符号时，输入完毕后用鼠标双击该文字并按下右键，之后再于快捷选项中单击「新增符号」便可自动导出变量符号的宣告窗口。

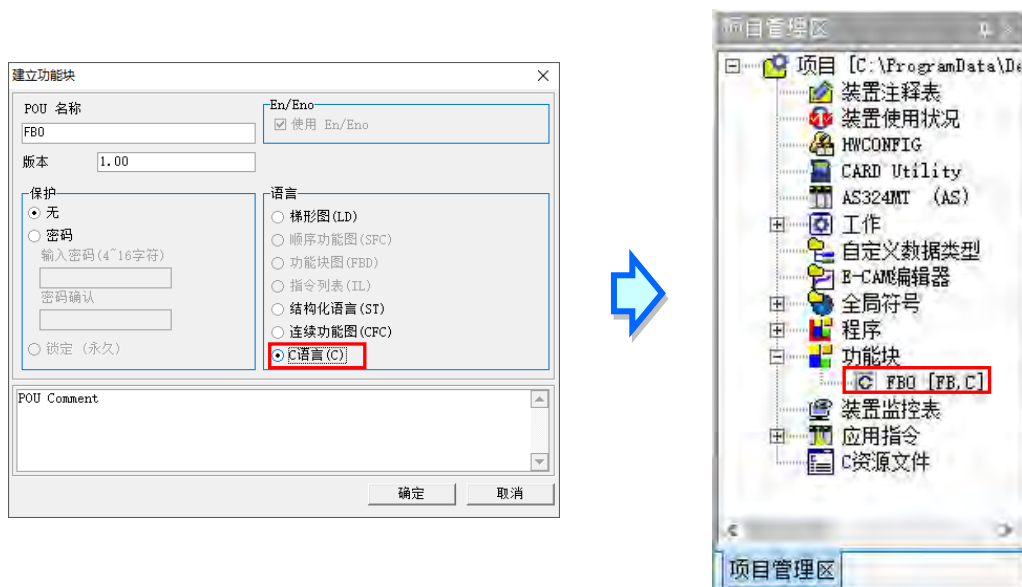


16

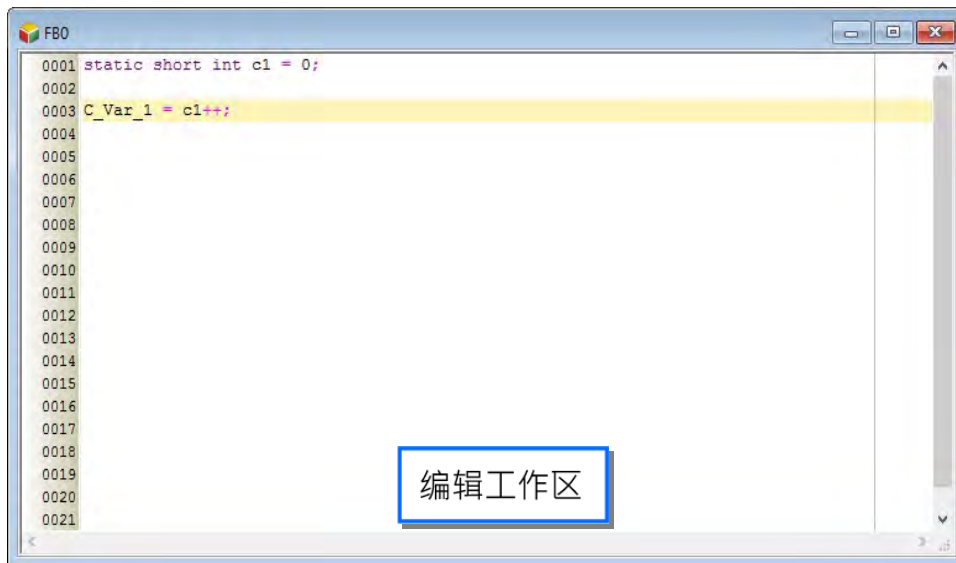
程序输入的过程中，会自动从以建立的变量符号中筛选出符合的变量符号供用户挑选，可利用键盘的「↑」、「↓」键，或直接在选项中单击鼠标左键完成输入。



建立功能块时，于语言字段选择「C 语言 (C)」便可建立 C 功能块。



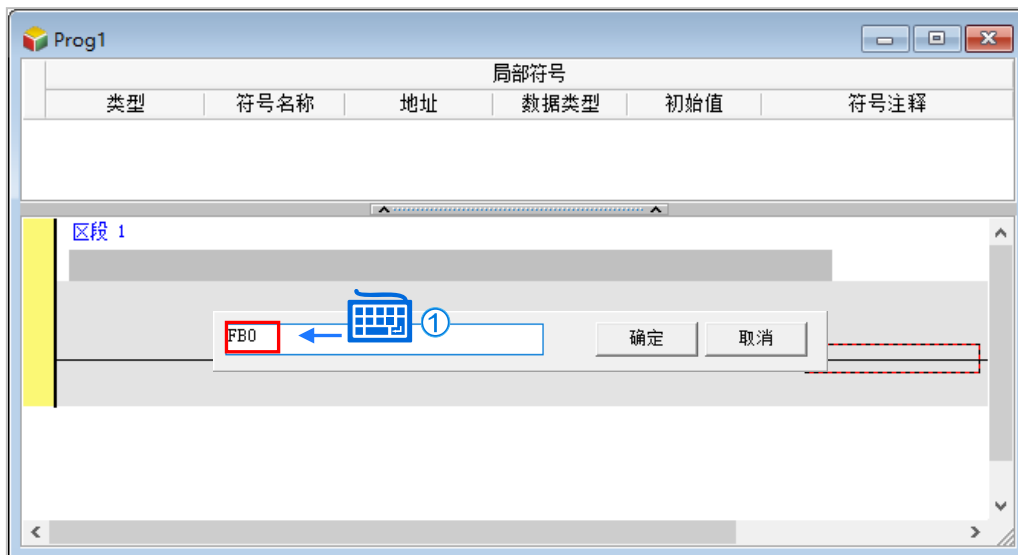
C 功能块的编辑方式与 C 程序相同，依据 C 语法的格式，直接使用键盘在工作区上面输入或修改文字即可。



插入 C 语言功能块到 LD 语言程序的方法有下述两种。

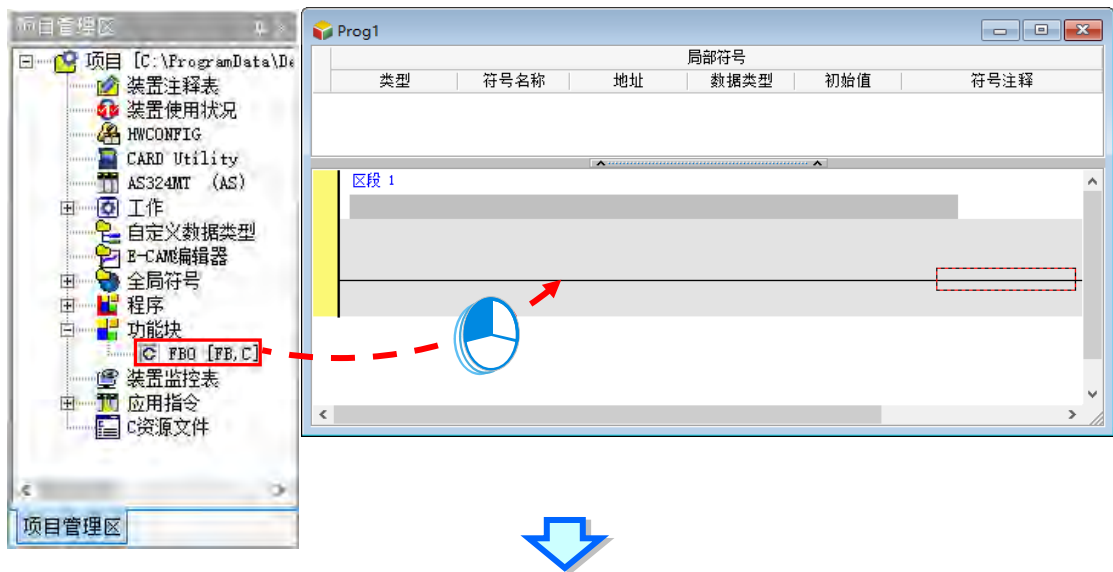
方法一

依据功能块名称，直接于 LD 语言的编辑工作区中利用键盘输入即可。



方法二

于项目管理区中展开功能块的目录，在其中找到欲插入的项目后，于该项目上按住鼠标左键，之后再将其拖曳至欲插入的位置后放开左键即可。

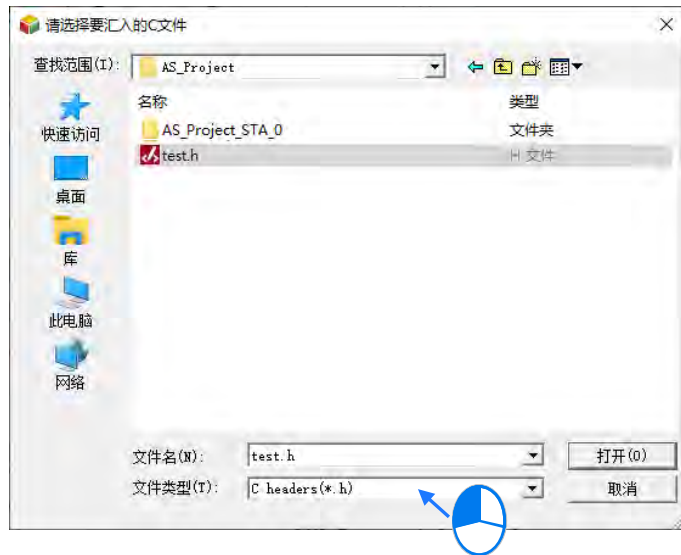
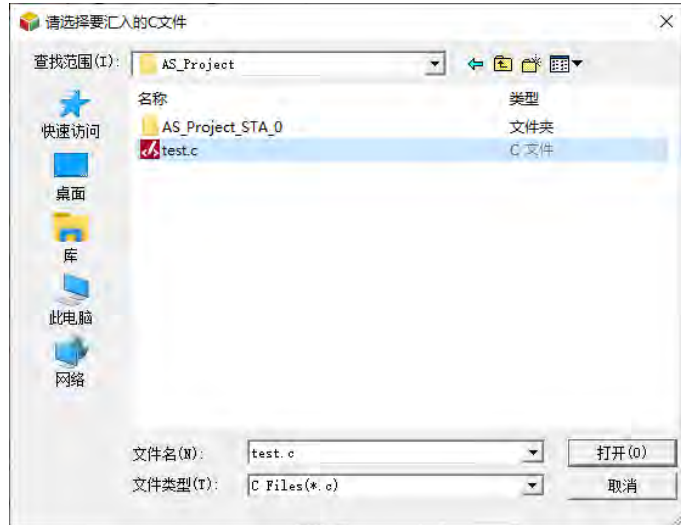


16



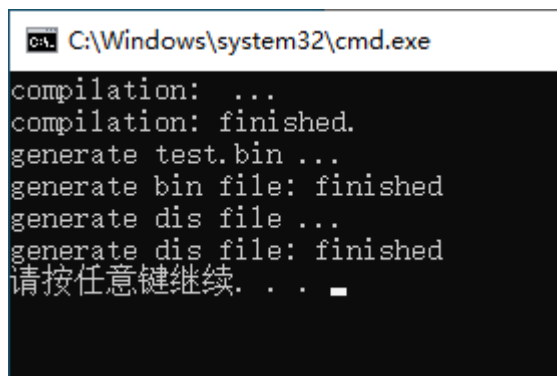
导入已经完成之 C 语言“.c”与“.h”文件，先于「C 资源文件」按下鼠标右键，接着于快捷选项中单击「新增」项目，之后在选择要导入的 C 文件窗口加，选取要导入的 C 文件，如下图。若要导入.H 文件则需要于「文件类型」项目中选取 C headers (*.h)。

16



16.1.3 C 语言编译与除错

ISPSoft 提供标准的 GCC 编译程序，以及 PLC 已内建的三角函数与指数类函式库，用户不用宣告就可以使用，但不提供市面上其它外挂的标准 C 函式库（如：math.h），以节省 PLC 内部提供的 C code 储存空间。



```
cmd C:\Windows\system32\cmd.exe
compilation: ...
compilation: finished.
generate test.bin ...
generate bin file: finished
generate dis file ...
generate dis file: finished
请按任意键继续. . .
```

编译信息	
✓ 程序 - 停电保持装置使用率	: 0 / 5000
✓ 功能块 & SFC - 装置使用率	: 2 / 16384
✓ C语言执行码使用率	: 352 / 65536 Bytes
✓ C语言变数使用率	: 20 / 8192 Bytes
✓ C语言与PLC共用变数使用率	: 4 / 1000 Words

ISPSoft 只显示 GCC 编译后的错误信息，但无法支持单击错误描述之后，直接跳到错误位置修改的功能，用户可在现有 PLC 程序与功能块进行在线监看，但无法对 C 语言编辑的程序进行一行一行的 C Code 进行监看。

16.1.4 内部函数库指令说明

一般常使用之整数类型或浮点数类型之四则运算“加、减、乘、除”，ISPSOft 都已内建，不需额外宣告即可使用。

已内建之三角函数名称:

C 函数名称	ISPSOft 编辑名称	指令格式	功能说明
sinf	DELTA_SINF	Float DELTA_SINF (float radian) ;	Sin 运算
cosf	DELTA_COSF	Float DELTA_COSF (float radian) ;	Cos 运算
tanf	DELTA_TANF	Float DELTA_TANF (float radian) ;	Tan 运算
asinf	DELTA_ASINF	Float DELTA_ASINF (float arg) ;	Sin^{-1} 运算
acosf	DELTA_ACOSF	Float DELTA_ACOSF (float arg) ;	Cos^{-1} 运算
atanf	DELTA_ATANF	Float DELTA_ATANF (float arg) ;	Tan^{-1} 运算
sinhf	DELTA_SINHF	Float DELTA_SINHF (float arg) ;	Sinh 运算
coshf	DELTA_COSHF	Float DELTA_COSHF (float arg) ;	Cosh 运算
tanhf	DELTA_TANHF	Float DELTA_TANHF (float arg) ;	Tanh 运算

已内建之指数型函数名称

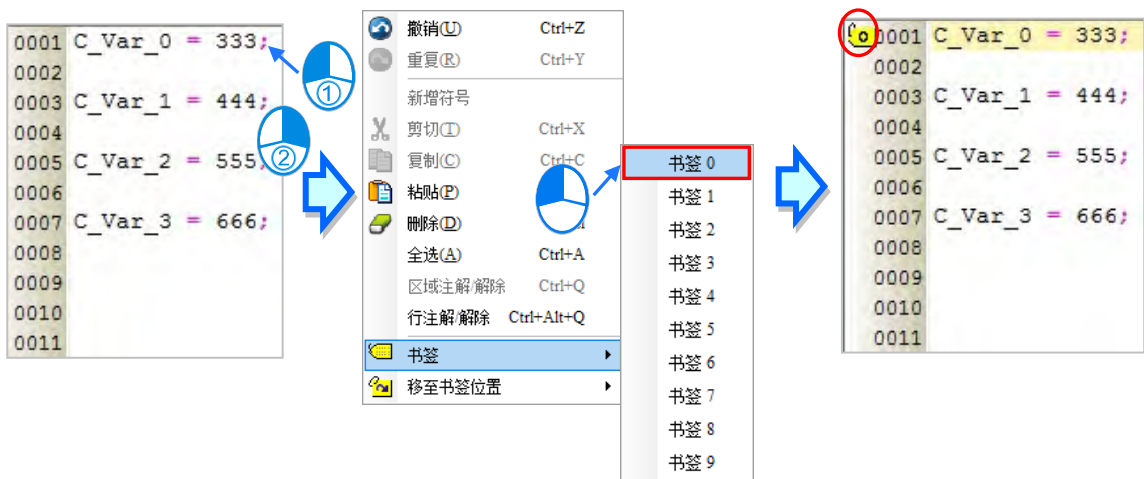
C 函数名称	ISPSOft 编辑名称	指令格式	功能说明
sqrtf	DELTA_SQRF	Float DELTA_SQRF (float value) ;	平方根运算
logf	DELTA_LOGF	Float DELTA_LOGF (float base, float value) ;	Log 运算
lnf	DELTA_LNF	Float DELTA_LNF (float arg) ;	Ln 运算
powf	DELTA_POWF	Float DELTA_POWF (float base, float power) ;	乘幂运算

16.1.5 书签与注解

书签功能为方便搜寻或移动至特定编辑位置的便利工具。

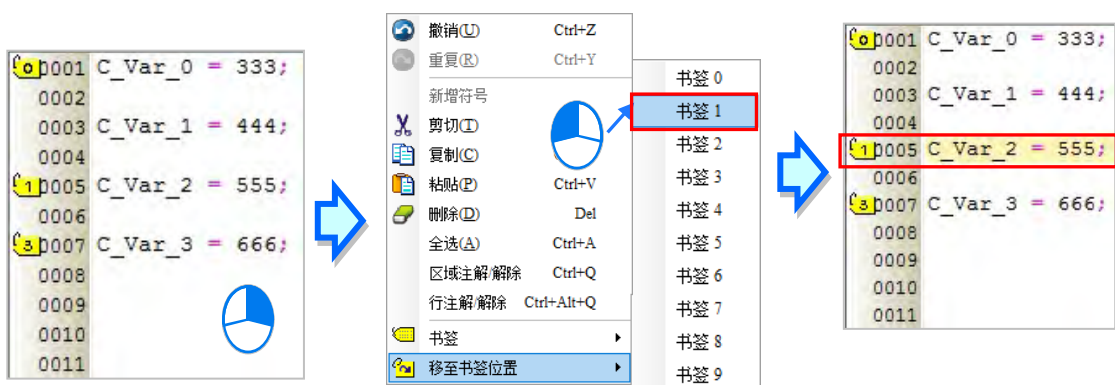
●加入/移除书签

先用鼠标单击欲加入书签的行列后按下鼠标右键，接着于快捷选项中单击「**书签**」项目，之后再于该项目中选择欲加入的书签编号即可。而要移除书签时，同样的请先用鼠标单击欲取消书签的行列后按下鼠标右键，接着于快捷选项中单击「**书签**」项目，之后再于该项目中单击与所选行列相同编号的书签即可；不过若选择的是其它编号时，所选行列的书签便会被变更为该编号。此外，若于菜单工具栏中单击 **编辑 (E) > 书签 > 移除所有书签 (R)**，则目前编辑窗口中的中的所有书签便会被全部取消。操作时若单击了一个已使用于其它行列的编号，则该编号的书签便会被搬移至目前选取的行列上。



●移至书签位置

于编辑工作区中按下鼠标右键，接着于快捷选项中单击「**移至书签位置**」项目，之后在于目录中选择移动目标的书签编号即可。



注解功能为方便用户，对选取区域或选取的行数，进行注解的工具。

● 行注解/解除

先用鼠标单击欲注解的行列后按下鼠标右键，接着于快捷选项中单击「行注解 / 解除」项目，之后在该行列的前面会自动加上 // 的注解。而要移除注解时，同样的请先用鼠标单击欲取消注解的行列后按下鼠标右键，接着于快捷选项中单击「行注解 / 解除」项目，之后在该行列的前面 // 的注解会自动移除。



区域注解/解除

先用鼠标左键选取欲注解的区域后按下鼠标右键，接着于快捷选项中单击「区域注解 / 解除」项目，之后在该选取区域的前后会自动加上 /* 与 */ 的注解。而要移除注解时，同样的请先用鼠标左键选取欲取消注解的区域后按下鼠标右键，接着于快捷选项中单击「行注解 / 解除」项目，之后在该区域的前后 /* 与 */ 的注解会自动移除。



16.1.6 范例操作说明

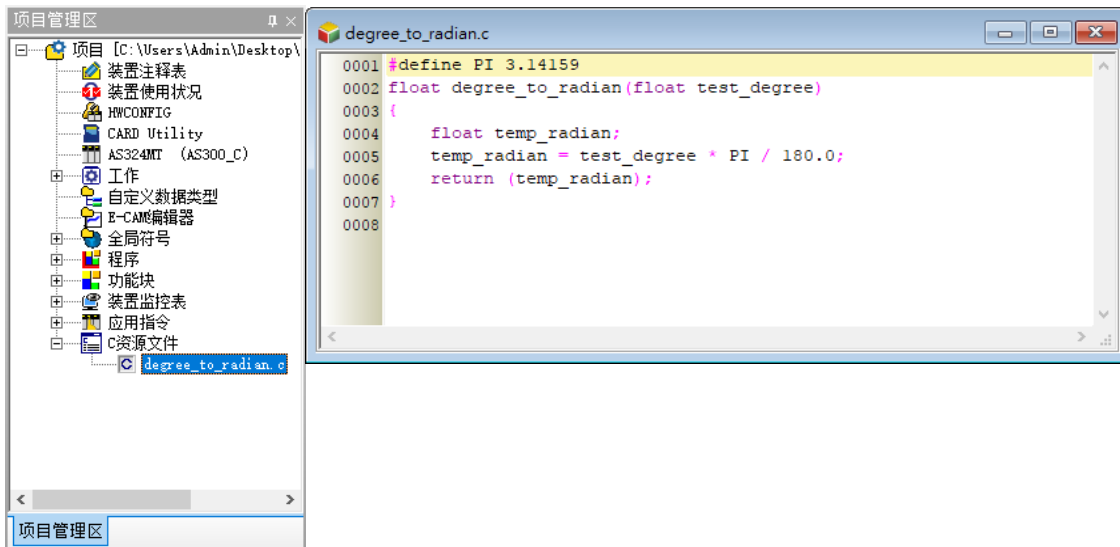
范例：每 10ms 以每次 5 度变换输出一笔 Sin 函数数值，并且用数据示波器功能，记录下 0~ 355 度的输出数值。

步骤 1：先建立 PLC 与 C 语言程序所需共享之变量符号名称参考第 16.1.2 节，如下图所示。



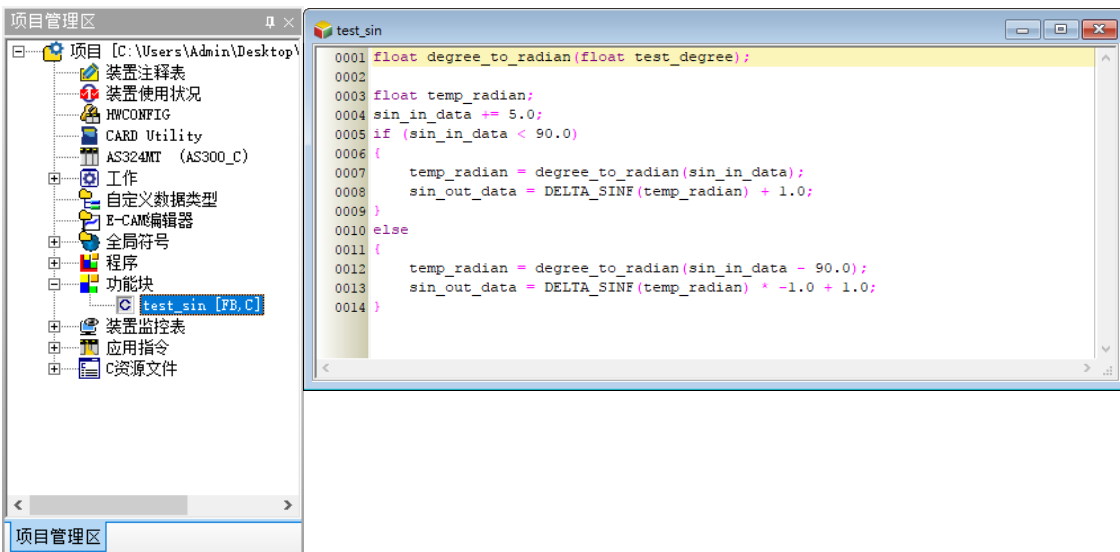
类型	符号名称	地址	数据类型	初始值	符号注释
VAR	sin_out_data	D12	REAL	N/A	
VAR	sin_in_data	D10	REAL	N/A	

步骤 2：导入一个自己已经编辑且完成测试的函数 degree_to_radian.c 参考第 16.1.2 节。此步骤用户可依状况自行删减。

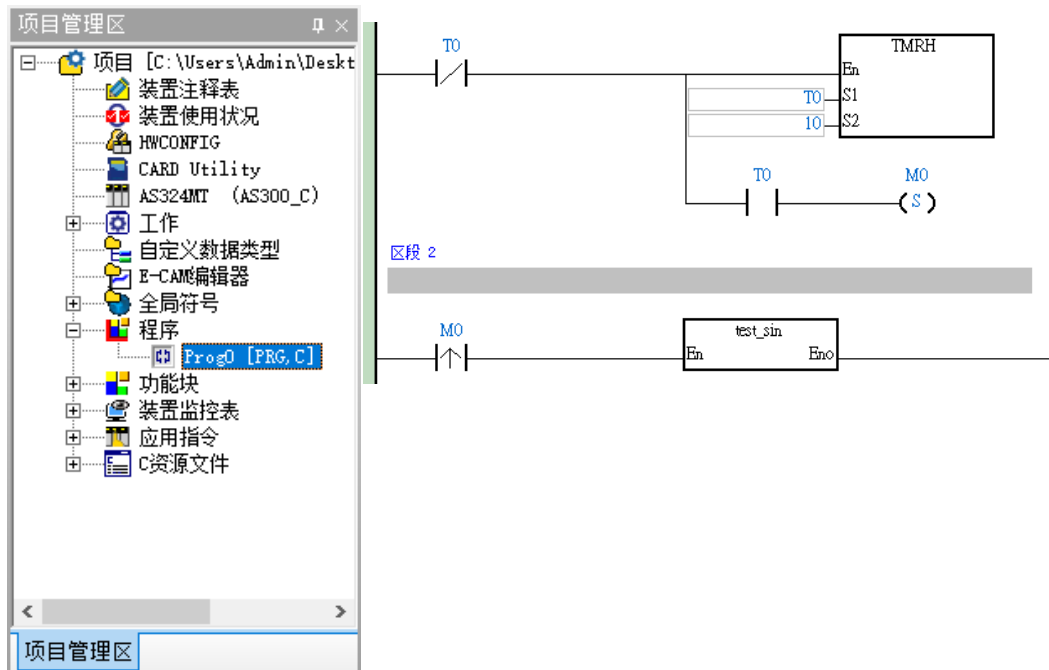


16

步骤 3：编辑一个新功能块 test_sin，并写入计算过程，如下图所示。其中 degree_to_radian 函数库。



步骤 4：建立一个程序 (POU)，并写一段每 10ms 调用一次 test_sin 的功能块 (FB)，如下图所示。



步骤 5：按下编译 (Compiler) 与下载 PLC 程序，假设都无错误发生，则过程如下图所示。



```

C:\Windows\system32\cmd.exe
compilation: ...
compilation: finished.
generate test.bin ...
generate bin file: finished
generate dis file ...
generate dis file: finished
请按任意键继续. . .

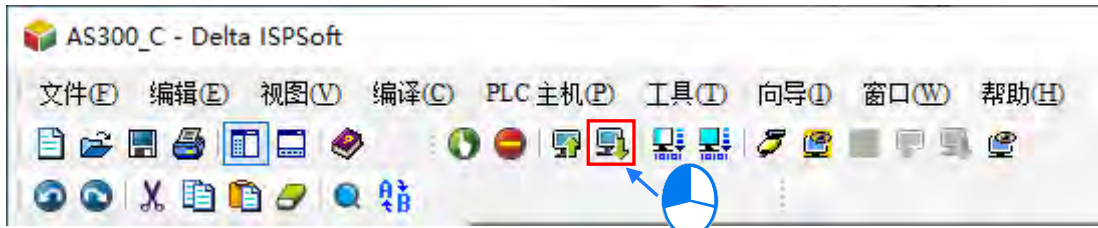
```



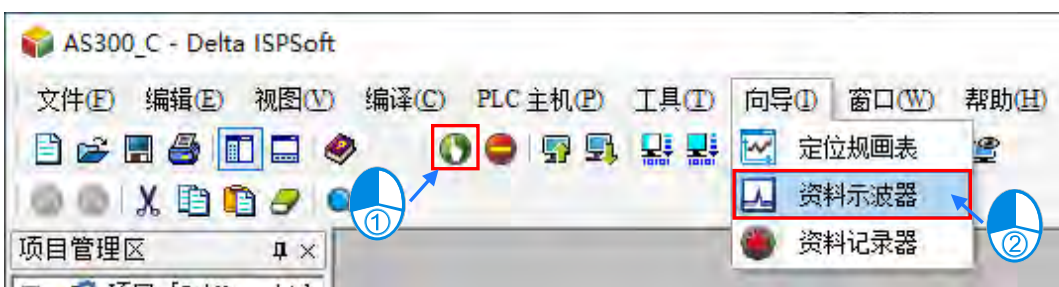
编译信息

- ✓ 0 错误
- ✓ 0 警告
- ✓ 程序 - 装置使用率：0 / 25000
- ✓ 程序 - 停电保持装置使用率：0 / 5000
- ✓ 功能块 & SFC - 装置使用率：2 / 16384
- ✓ C语言执行码使用率：352 / 65536 Bytes
- ✓ C语言变数使用率：20 / 8192 Bytes
- ✓ C语言与PLC共用变数使用率：4 / 1000 Words

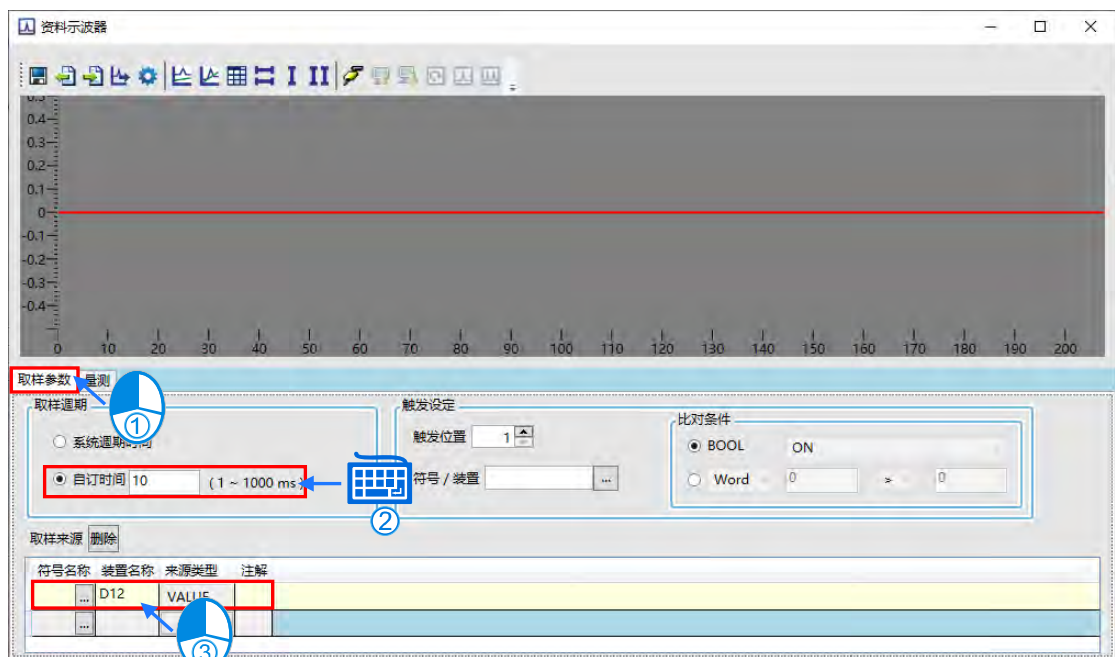
步骤 6：编译（Compiler）成功后下载 PLC 程序。



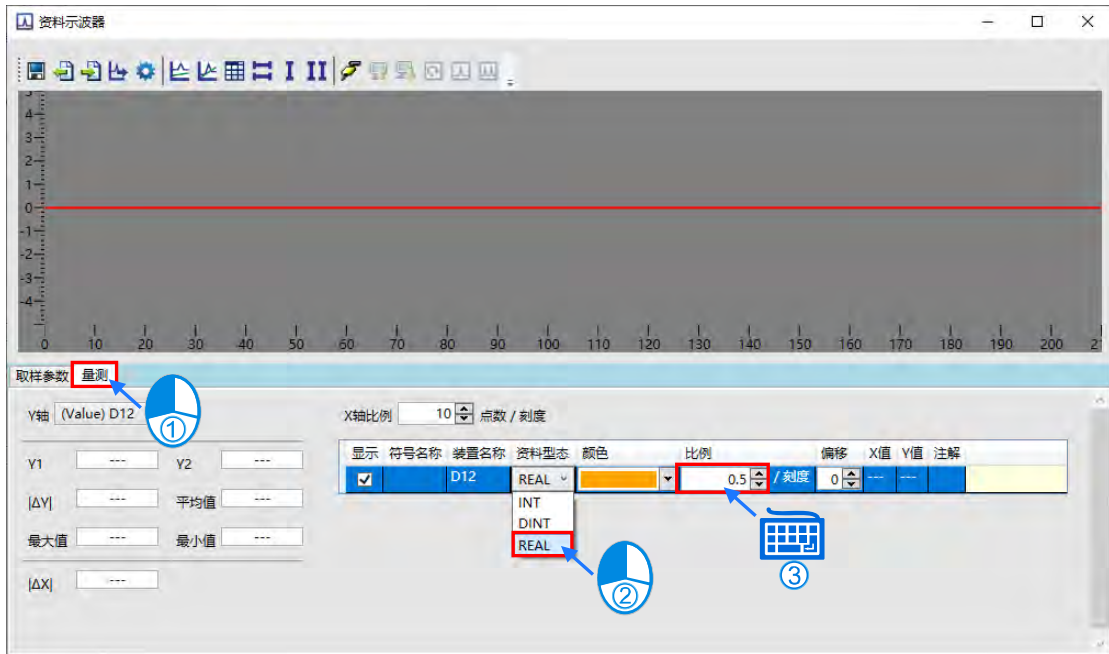
步骤 7：启动 PLC Run 之后，开启数据示波器，其过程如下图所示。



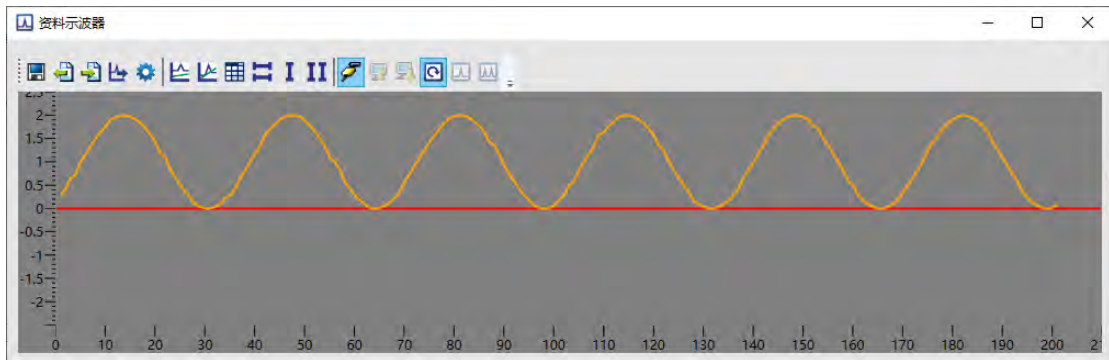
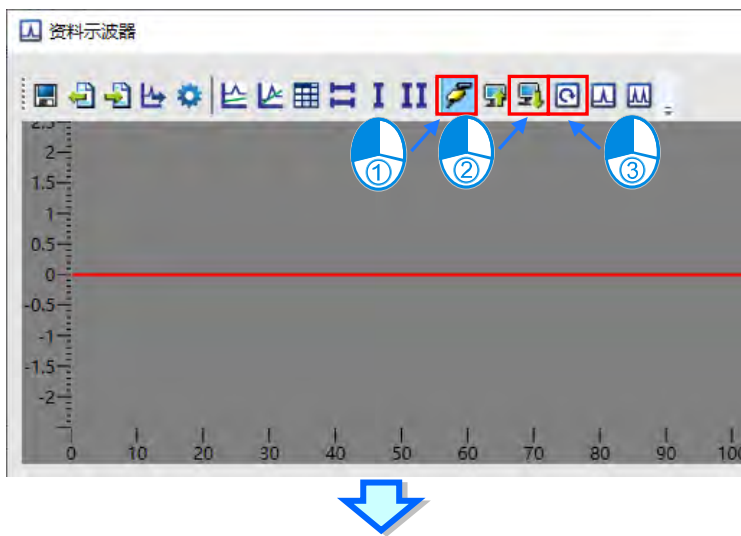
步骤 8：数据示波器设定取样参数及测量，其过程如下图所示。



16



步骤 9：数据示波器联机、下载、实时监控，其过程如下图所示。



第17章 编辑辅助工具

目录

17.1 ISPSOft 的编辑工具与辅助功能.....	17-2
17.1.1 变更 PLC 机种	17-2
17.1.2 将 DVP 机种转换为 AS 机种	17-3
17.1.3 项目的下载与上载.....	17-14
17.1.4 LD、FBD 与 CFC 编辑环境的查找替换与定位到功能.....	17-16
17.1.5 IL 与 ST 编辑环境的查找替换功能	17-19
17.1.6 SFC 编辑环境的查找替换功能	17-22
17.1.7 符号表的查找功能.....	17-24
17.1.8 项目查找与替换	17-26
17.1.9 打印功能.....	17-27
17.2 装置与寄存器管理	17-30
17.2.1 DVP 机种的装置注解表	17-30
17.2.2 AH/AS 系列机种的装置注解表	17-34
17.2.3 装置使用状况	17-41
17.2.4 寄存器编辑	17-44
17.2.5 装置状态编辑	17-51
17.2.6 DVP 机种的文件寄存器编辑	17-56
17.3 步序定位	17-63
17.3.1 使用步序跳转	17-63
17.4 程序比对	17-64
17.4.1 程序比对简介	17-64
17.4.2 与文件比对	17-65
17.4.3 与 PLC 比对	17-67

17.1 ISPSoft 的编辑工具与辅助功能

17.1.1 变更 PLC 机种

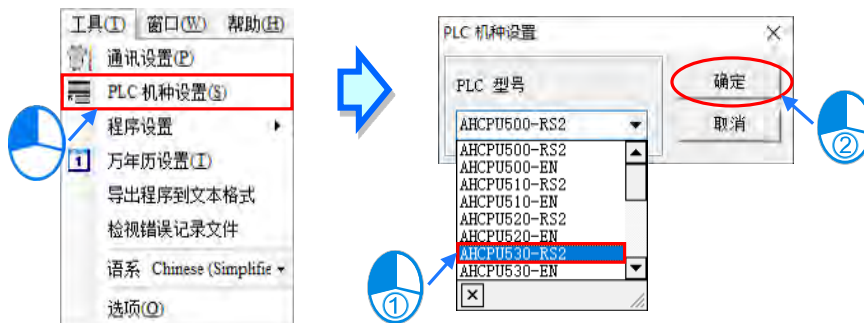
ISPSoft 允许用户变更项目机种。大部分的系列机种提供双向互转而部分机种只提供单向转换。如 DVP 机种可以转换到 AS 机种，但是 AS 机种不能转换到 DVP 机种；AH5x0 机种可以转换到 AH560 机种，但 AH560 机种不能转换到 AH5x0 机种。详细可以参考下表。

目的 来源	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
AS	V	V	V	V	V	只支持 ES3	
AH5x0		V	V	V	V		
AH5x1			V	V	V		
AH560			V	V	V		
AHxxEMC					V		
DVP	V					V(*1)	
AS5xx DVPxxMC							V

*1. DVP-ES3 机种不能转换到其他 DVP 机种，但其他 DVP 机种可以转换到 DVP-ES3 机种。

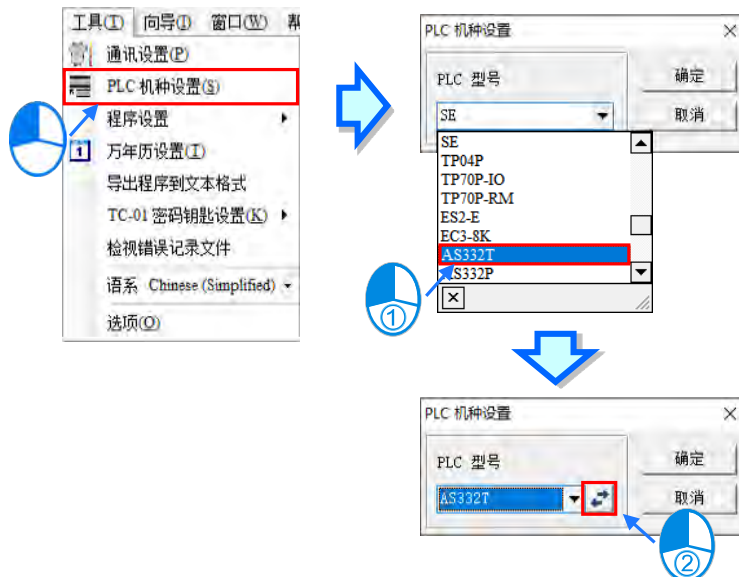
变更机种会将目前项目的内容改变，所以在变更机种之前建议将项目另外保存一份，且变更前后的机种功能、装置范围或支持的指令可能都会有所差异，因此变更后请务必再次确认原本的程序、参数设定、硬件规划与网络规划...等部分，是否可让系统正常运作。

当要变更项目机种时，请于菜单工具栏中单击 **工具 (I) > PLC 机种设定 (S)**，之后于设定窗口中选择要变更的 PLC 型号后按「确定」键即可。

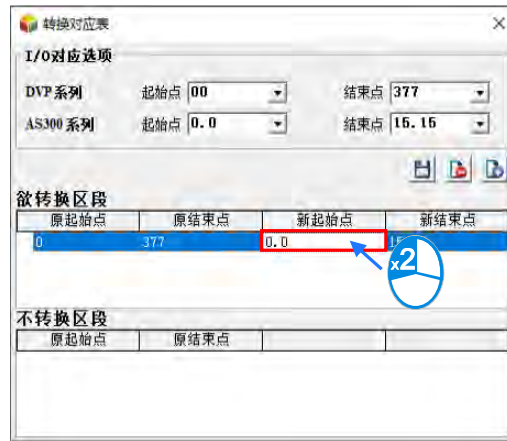


17.1.2 将 DVP 机种转换为 AS 机种

若要进行 DVP 机种转换为 AS 机种，请在菜单工具栏中单击 **工具 (T) > PLC 机种设置 (S)**，之后设定窗口中选择要变更的 AS 机种后，可单击选择字段旁边的双箭头按钮，进入「转换对应表」，做 I/O 转换的配置。



对应表中原起始点及原结束点，代表 DVP 机种之输入 X、Y 的起始及结束点；新起始点及新结束点，则代表 AS 机种之输入 X、Y 的起始及结束点，用户需注意，DVP 系列 I/O 对 AS300 系列 I/O 转换范围不一致时，会出现「警告：范围不相符」；对要设定的表格列，双击鼠标左键，此时保存/删除按钮将出现，即可用上方下拉式选单进行选择或是直接选取要变更字段直接进行修改，如下图。

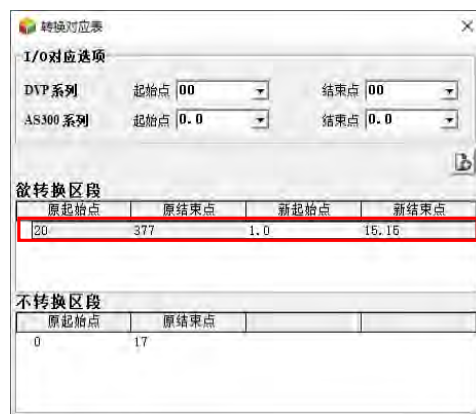


OR



当设定完要变更的点位，单击保存按钮，便可完成设定，此时会列出「欲转换区段」以及「不转换区段」。

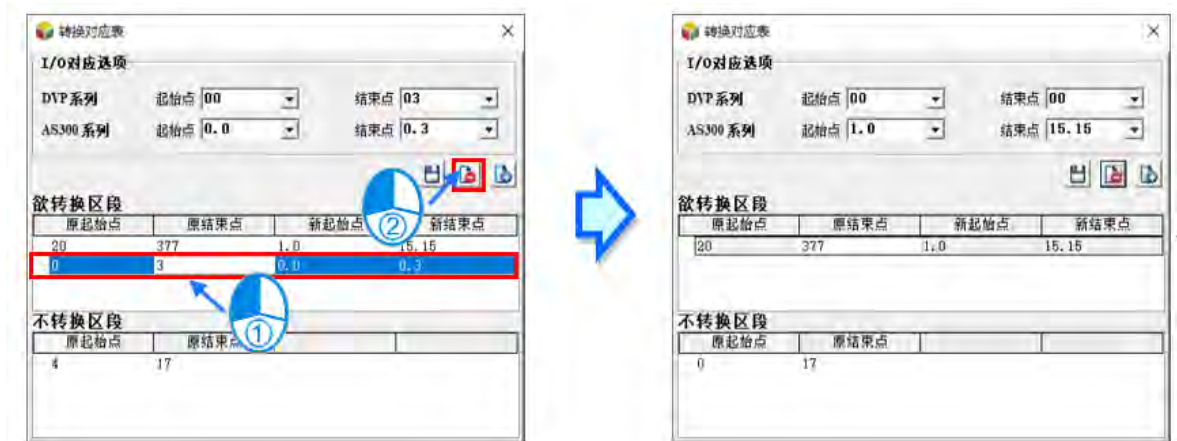
17



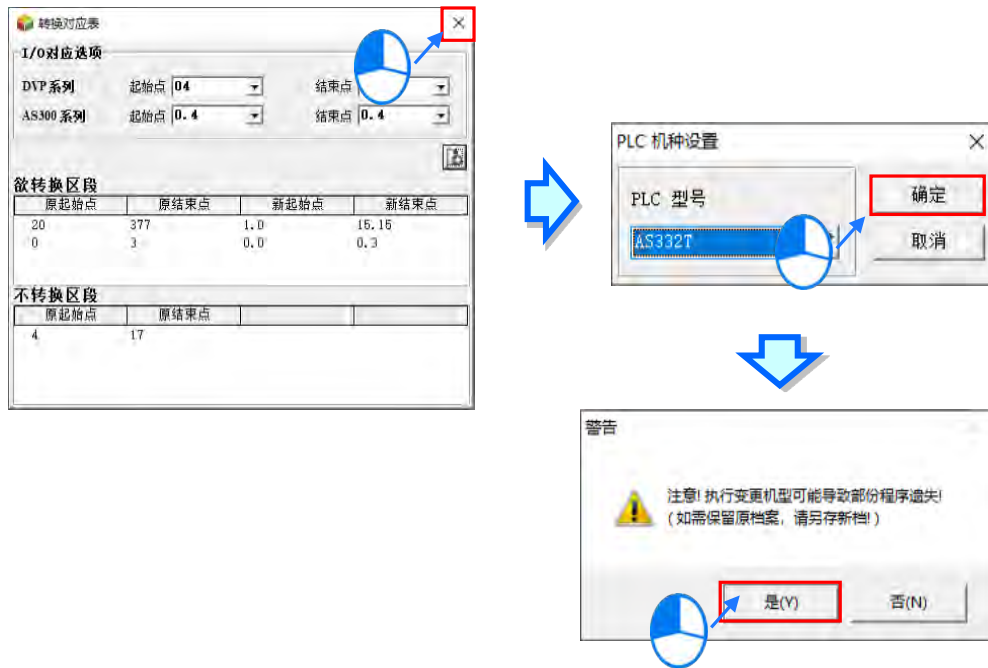
要新增一片段 I/O 转换，可对表格列单击鼠标左键，输入 I/O 字段，再单击新增按钮，即可新增片段



要删除已设定的对应表格列，可对表格列双击鼠标左键，再单击删除按钮，便可完成删除。



设定完转换对应表，即可退出转换对应表窗口，再对窗口，单击「确认」，会出现询问警告窗口，告知执行转换变更机型可能导致部份程序遗失（此时若用户欲保留原文件，可先单击「否」，取消转换后，另存完文件后再进行转换），是否要进行转换，若选择「是」，则会开始进行转换工作。



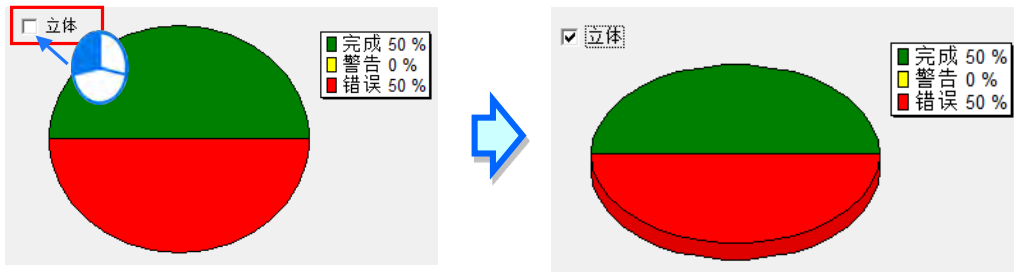
当完成转换工作后，会出现「转换报告」，如下图。



用户可在「转换报告」上方双击鼠标左键，将窗口转换为动态模式，再将窗口向右拉开并往上移，以方便观察。



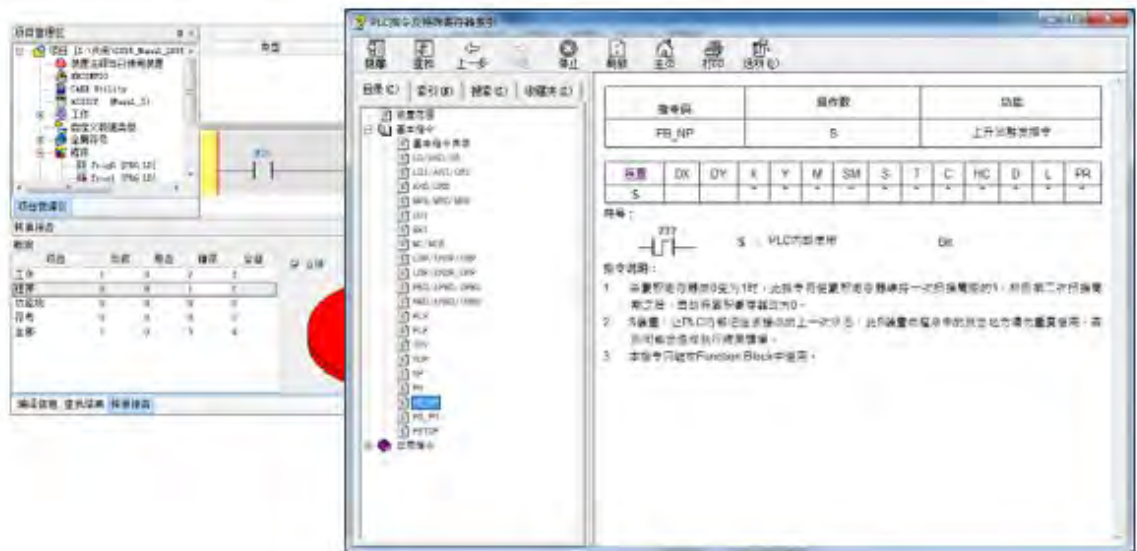
转换报告左侧「概观」，会列表出转换后的警告、错误及完成项目，内容分为工作、程序、功能块、符号及全部，左下方的「全部」，计算每一行之加总，而右上方的「全部」，计算每一列的加总；中间「圆饼图」会根据所选项目显示错误、警告及完成的比例图，绿色部分为「完成」、黄色部分为「警告」、红色部分为「错误」，若勾选圆饼图左上方的「立体」，则圆饼图会呈现立体型式。



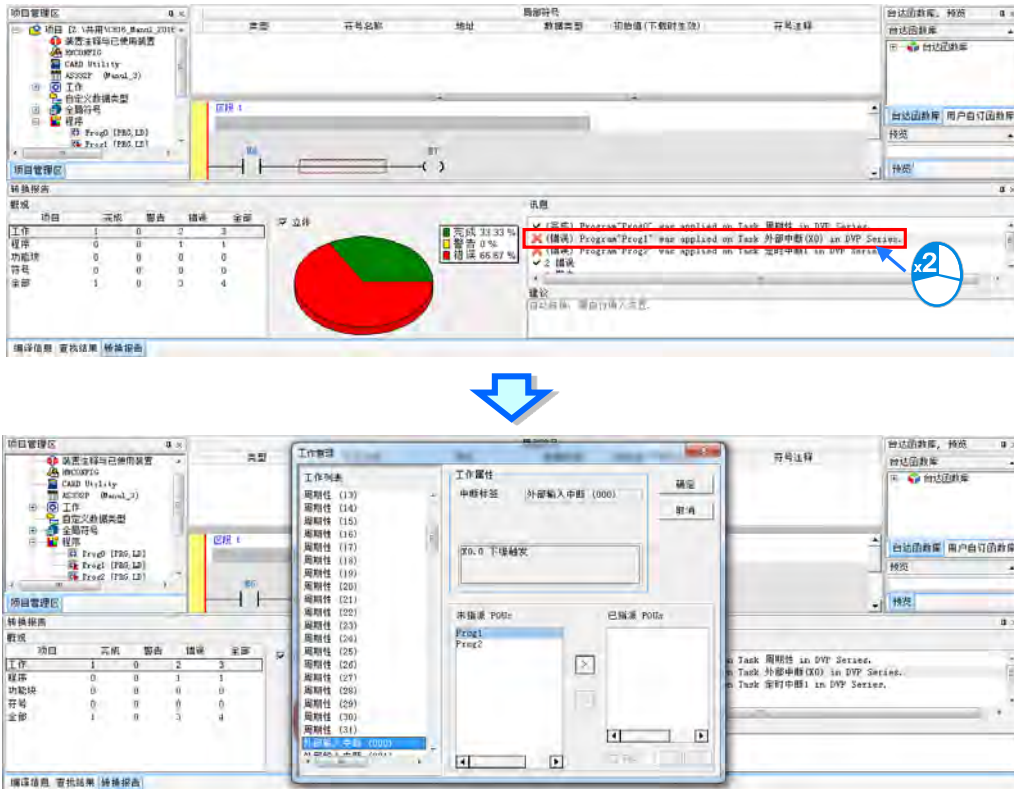
转换报告右侧「信息」，会在可完全转换的数据前端显示(完成)，若转换后可能发生调试问题的数据前端则会显示(警告)，若转换后发生无法转换的数据前端则会显示(错误)，用户也可在「概观」表格栏单击想要单独检视完成、警告或错误数据内容，则「信息」框就只会显示相对应内容，如下图。



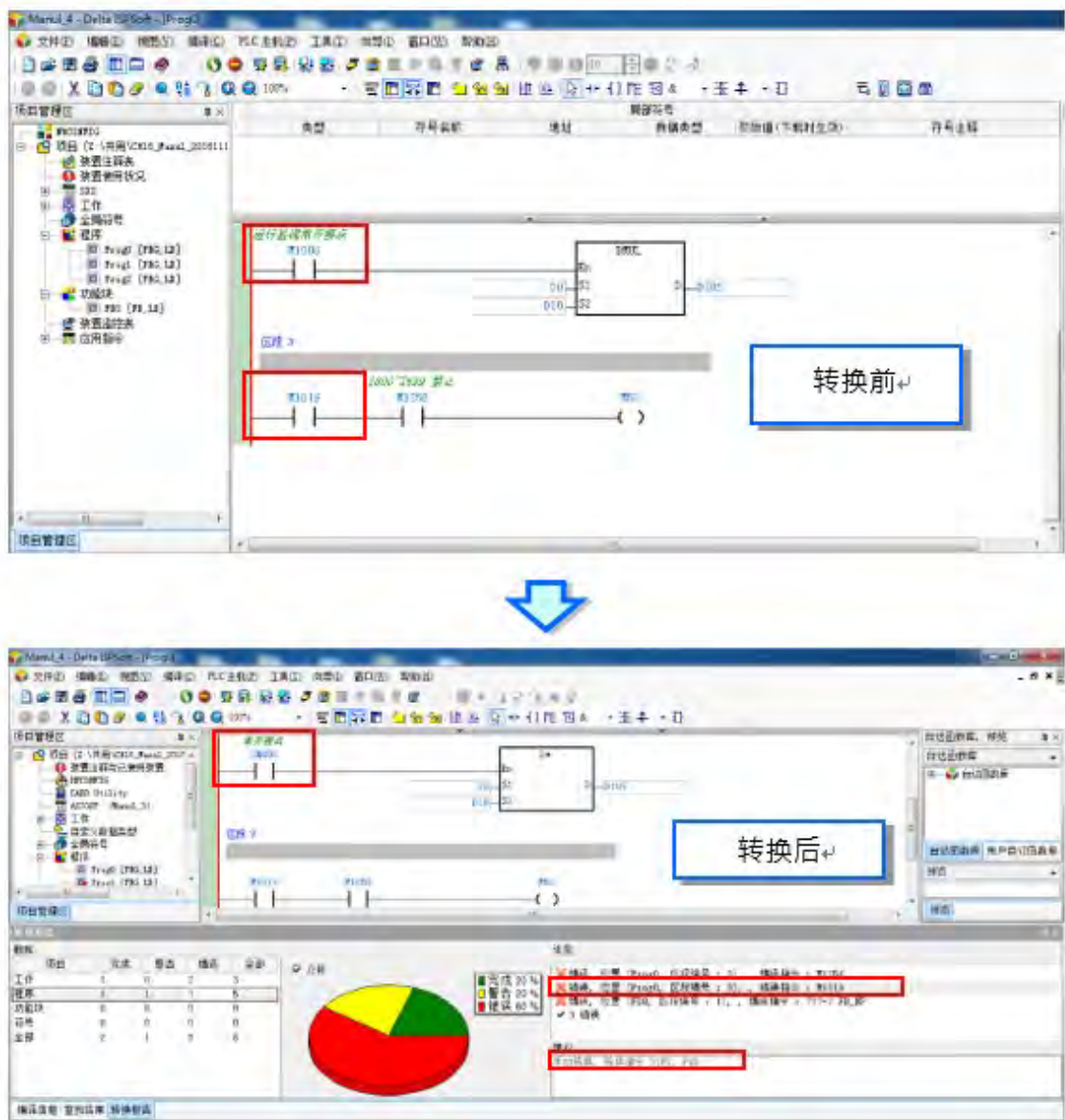
转换报告右侧下方「建议」，则会根据转换后有出现「警告」或「错误」的信息，给予调适修正的建议。针对转换后，某些指令需做重新调适修改，会有相对应索引文件显示，用户可根据索引文件提示的内容，做适当的修饰。



若转换后，在「工作管理」部份有所变动，「建议」框中会显示建议的解决方式，用户可对「信息」框中相对应的信息双击鼠标左键，此时会跳出「工作管理」窗口，以供用户调整。

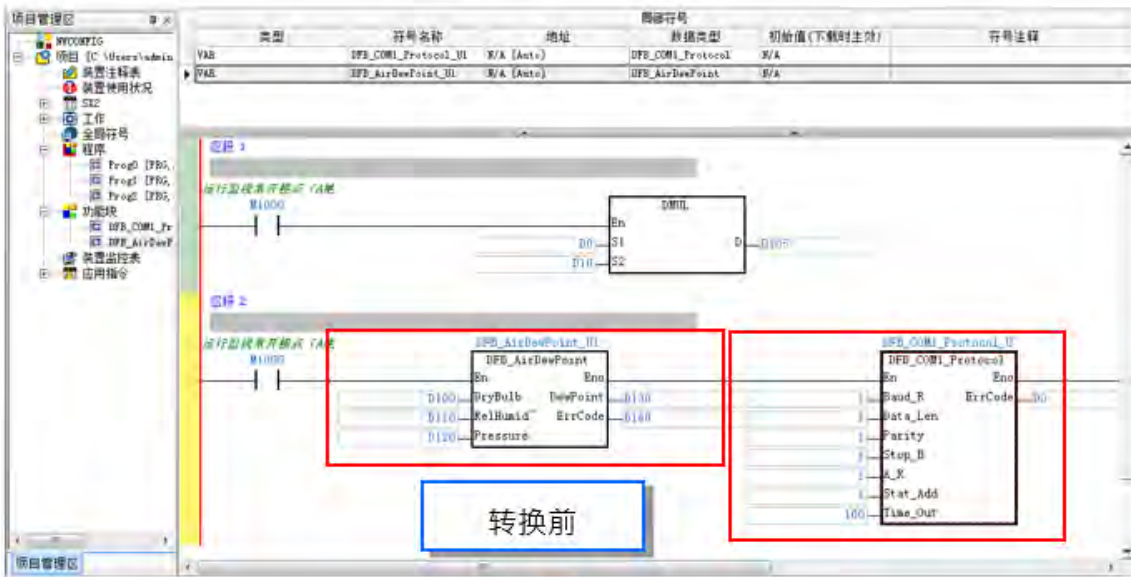


在做程序转换时，DVP 系列机种之特殊辅助继电器(特 M)或特殊数据缓存器(特 D)，若 AS300 系列机种有相对应的特殊内部电驿或特殊内部缓存器可转换，则会自动生成；若没有可转换，则「建议」框中，会显示调试的方法。



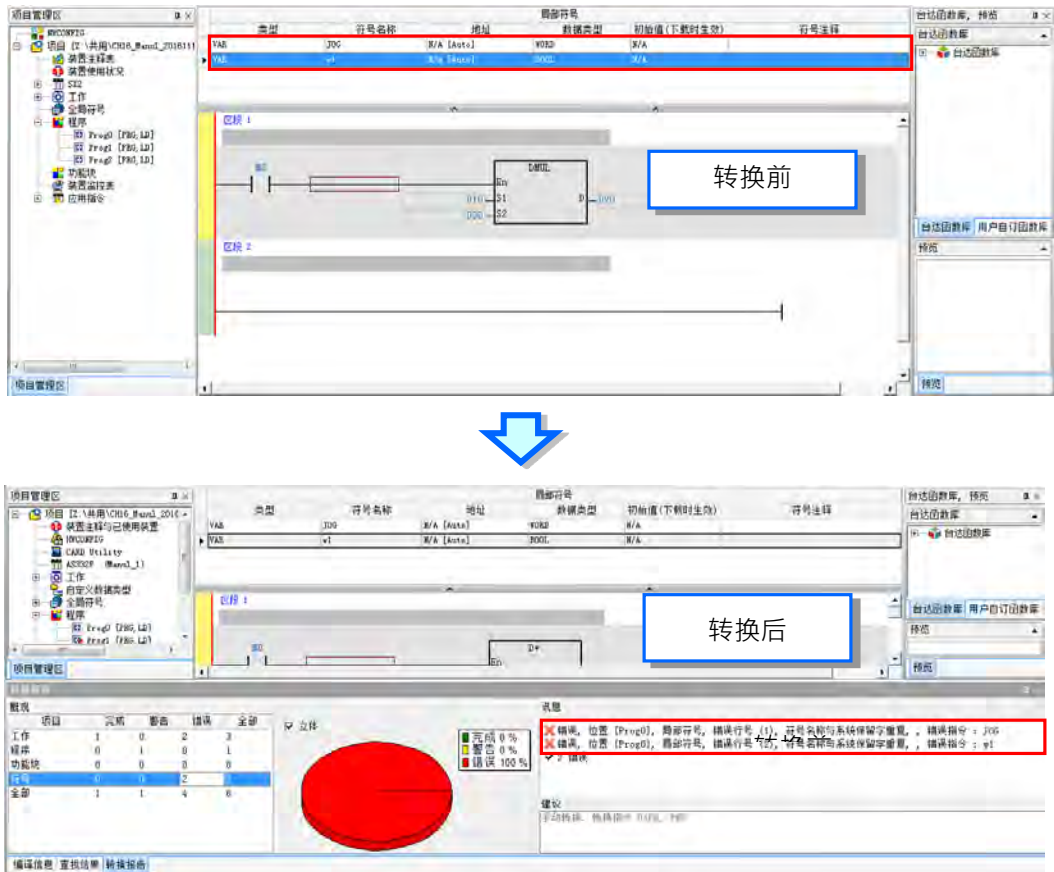
17

在做程序转换时，若 DVP 机种与 AS300 系列机种都有支持「台达功能块」，则在转换之后，会自动生成，若转换之后，AS300 系列机种没有可对应的「台达功能块」，功能块则会出现叉号的错误提示，如下图所示。

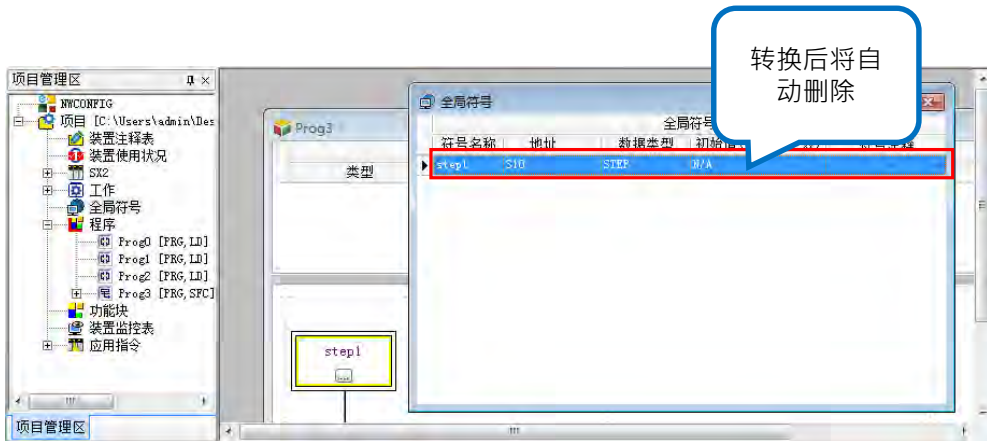


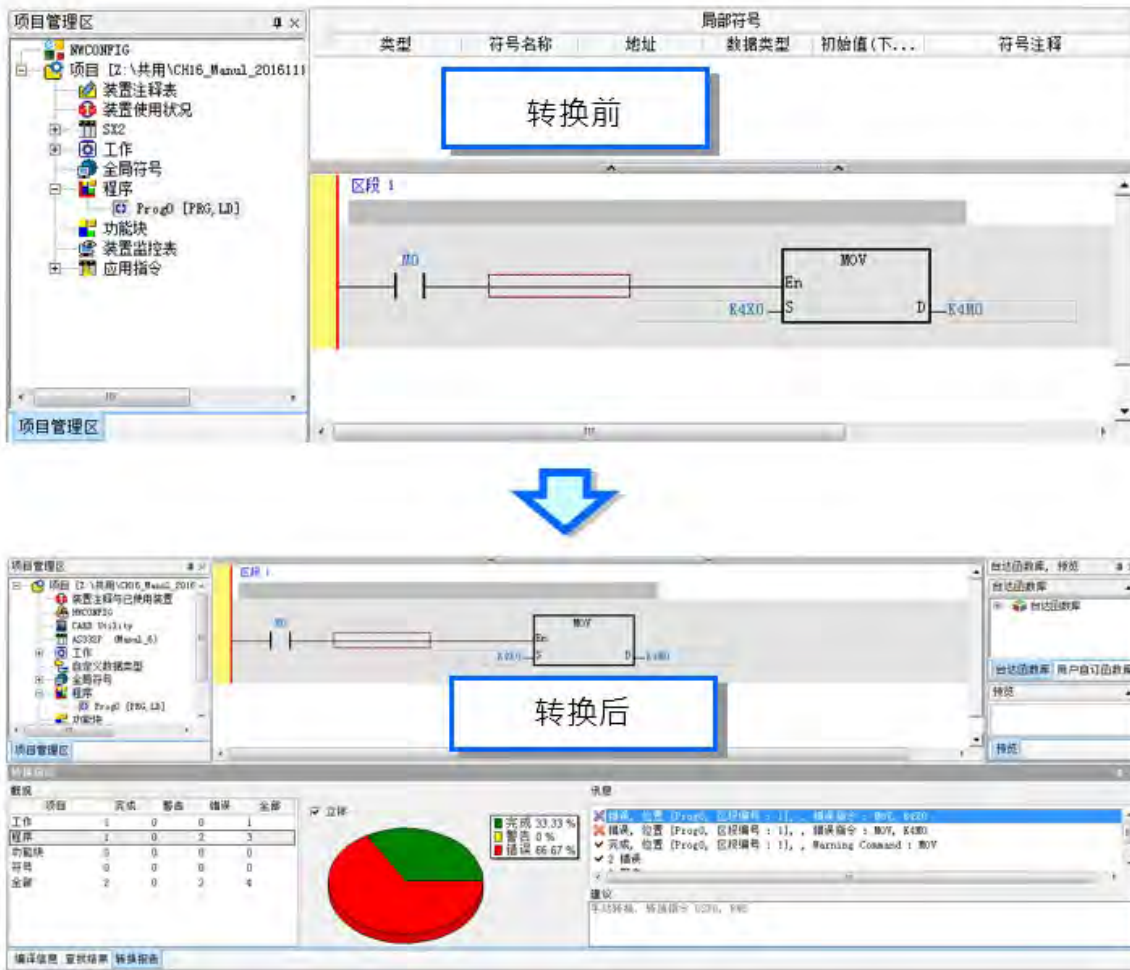
17

在做程序转换时，若 DVP 机种使用的「符号名称」与转换后的 AS300 系列机种关键词，例如:应用指令、装置名称，有所冲突时，转换后会出现错误信息。




在做程序转换时，DVP 系列机种转 AS 系列机种转换，只支持梯形图(LD)及顺序功能图(SFC)语法转换。若在 DVP 系列机种使用的变量类型是 AS 系列不支持的，转换后将会被自动删除。例如在变量表宣告一个数据类型为 STEP 的变量，在转换后此变量将会被自动删除；若在 DVP 系列机种有使用到 Kn 装置表示法，在转换为 AS 系列机种时会发生错误。






17

17.1.3 项目的下载与上载

ISPSOft 的上下载功能，除了针对项目中的程序进行操作之外，同时亦可包含其他的功能或参数。欲进行下载时，请于菜单工具栏中单击 **PLC 主机 (P)** > **PLC 程序传输** > **下载 (D)**，或直接于图标工具栏中单击  图标，接着再在出现的窗口中勾选欲下载的项目后按下「开始传输」即可。



而欲进行上传时，菜单工具栏中单击 **PLC 主机 (P)** > **PLC 程序传输** > **上传 (U)**，或直接单击图标工具栏的  示，之后再于出现的窗口中勾选欲上传的项目后按下「开始传输」即可。



上下载窗口中，支持不同功能的各系列机种会有不同的上下载项目。关于各选项的说明请参考下表。

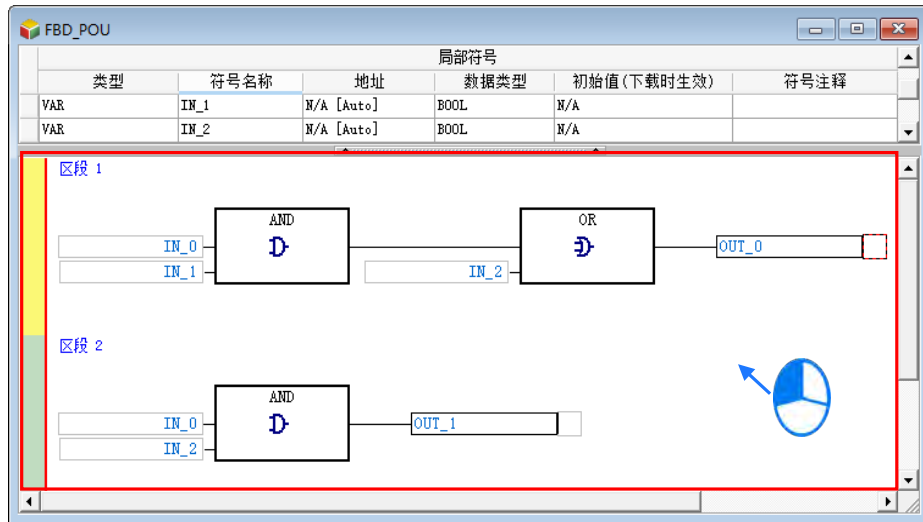
传输选项	说明	AH/AS		DVP		AS5xx DVPxxMC	
		上传	下载	上传	下载	上传	下载
执行码	程序编译后的 PLC 执行码	V	V	V	V	V	V
程序	用户建立的项目程序	V	V	V	V	V	V
专案批注	项目程序的批注	V	V	V	V	V	V
POU 批注	POU 与 FB 的批注	V	V	V	V	V	V
区段批注	POU 与 FB 程序的区段批注	V	V	V	V	V	V
符号批注	全局符号及区域符号的符号批注	V	V	V	V	V	V

传输选项	说明	AH/AS		DVP		AS5xx DVPxxMC	
		上传	下载	上传	下载	上传	下载
项目密码与 PLC 密码 同步	将项目密码同步到主机的 PLC 密码		V		V		V
工作参数	工作参数设定					V	V
停电保持范 围	停电保持范围的设定值				不支持 ES3		
停电保持符 号表	符号表内的停电保持符号					V	V
CANopen 参数	CANopen Builder 参数设定值					V	V
符号初始值	将符号初始值写入主机对应的 内存位置		V		V		
万年历	将万年历设定的日期/时间写 入 PLC		V		V		V
定位规划表	定位规划表内容	只支 持 AS	只支 持 AS	只支 持 ES3	只支持 ES3		
轴参数	轴参数内容	只支 持 AHxx EMC	只支 持 AHxx EMC			V	V
Ethernet/IP 符号表	Ethernet/IP 符号表内容		只支 持 AS		只支持 ES3		
E-CAM	E-CAM 内容	只支 持 AS 与 AHxx EMC	只支 持 AS 与 AHxx EMC	只支 持 ES3		V	V

17.1.4 LD、FBD 与 CFC 编辑环境的查找替换与定位到功能

针对不同的编辑窗口，ISPSOft 所提供的**查找**、**替换**与**定位**功能皆会有所不同，而本节则先针对 LD、FBD 与 CFC 的编辑环境之功能操作加以说明。

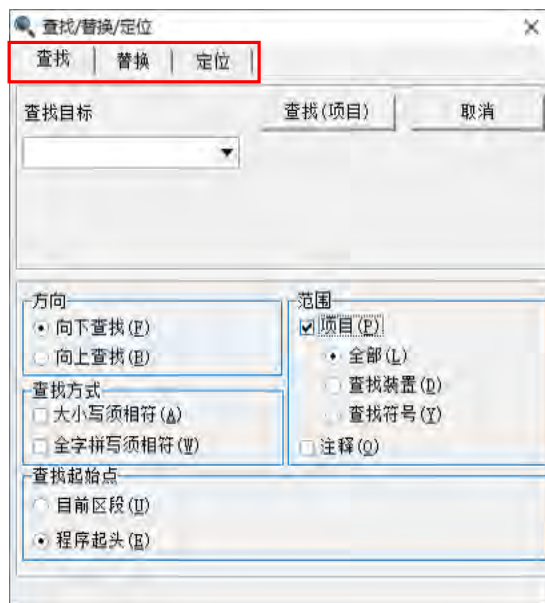
(1) 请开启工作窗口，接着用鼠标左键单击程序编辑区。



- (2) 在菜单工具栏的「编辑 (E)」项目下选取查找、替换与定位，或选取图标工具栏中对应的图标。皆可打开查找/替换/定位窗口，也可藉由切换窗口上方的卷标来更换操作的功能。



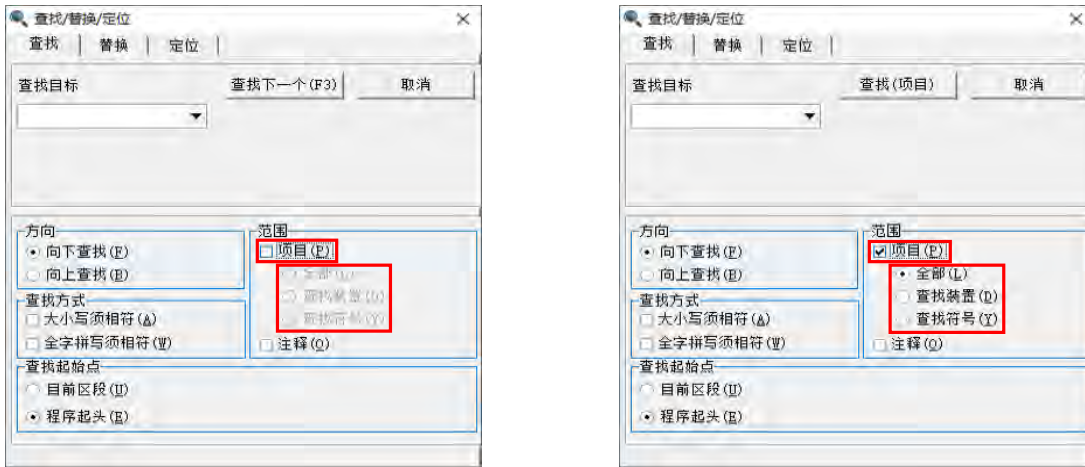
OR



17

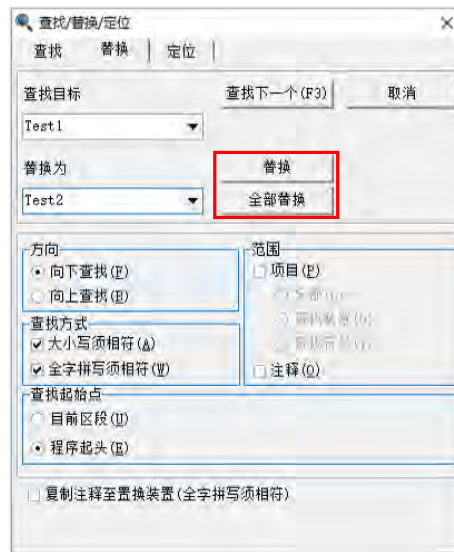
下图为查找功能的操作页面，必须设定查找目标，并在下列字段中选择查找条件，而设定完毕后按下「查找下一个」按钮或键盘的【F3】键便可逐一进行查找的工作，且即使关闭查找窗口后，按下键盘的【F3】键仍可继续查找最后一次查找的目标物。当范围不勾选项目时，查找的范围仅限于目前正在

编辑的窗口，当范围勾选项目时，可勾选**全部**或**查找装置**或**查找符号**，来进行分类查找，结果将列于信息显示区中。

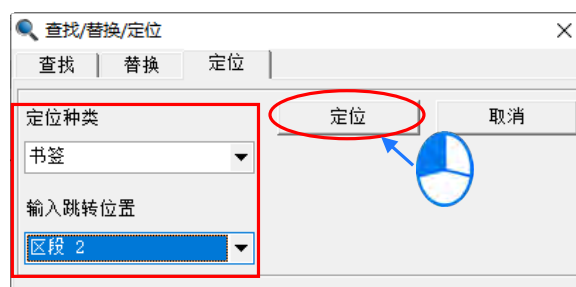


项目	说明
查找目标	直接输入查找目标，或是单击字段右侧的箭头以下拉挑选查找过的字符串。 查找装置 ：查找符合目标的所有装置。 查找符号 ：查找符合目标的所有符号。
方向	于此处选择查找的方向。
选项	大小写须相符 ：比对字符串须与查找目标的大小写完全一致才符合查找条件。 全字拼写须相符 ：比对字符串须与查找目标的内容完全一致才符合查找条件。若勾选 全字拼写须相符 时可再勾选 复制注释至置换装置 (全字拼写须相符) ，将来源装置注释复制到目标的装置注释中。
查找起始点	目前区段 ：由目前单击的编辑位置开始查找。 程序起头 ：依据查找的方向，由程序的最前面或最后面开始查找整段程序。在 CFC 语言固定为此模式。
范围	勾选查找时所包含的范围。(勾选「项目」时，「全部」、「查找装置」、「查找符号」才会使能。) 全部 ：查找符合目标的所有装置与符号。 查找装置 ：查找符合目标的所有装置。 查找符号 ：查找符合目标的所有符号。 注释 ：查找符合目标的所有注释。

下图为**替换**功能的操作页面，输入查找目标，并于「**替换为**」字段直接输入欲替换的字符串，按下「**替换**」按钮便可将目前所找到的目标物字符串变更为输入的替换字符串，而按下「**全部替换**」按钮则会一次替换所有符合查找条件的目标物。



在 LD 与 FBD 语言中的「到」页面可以执行快速移动的功能，在「定位种类」字段下拉选择要采用的定位种类，共有区段卷标与书签两种，接着再于「输入跳转位置」字段下拉选择该类别的定位目标，完成后按下「定位」按钮便可引导用户快速移动至程序中的指定位置。

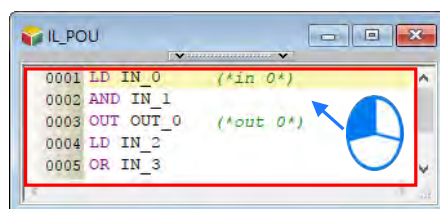



17

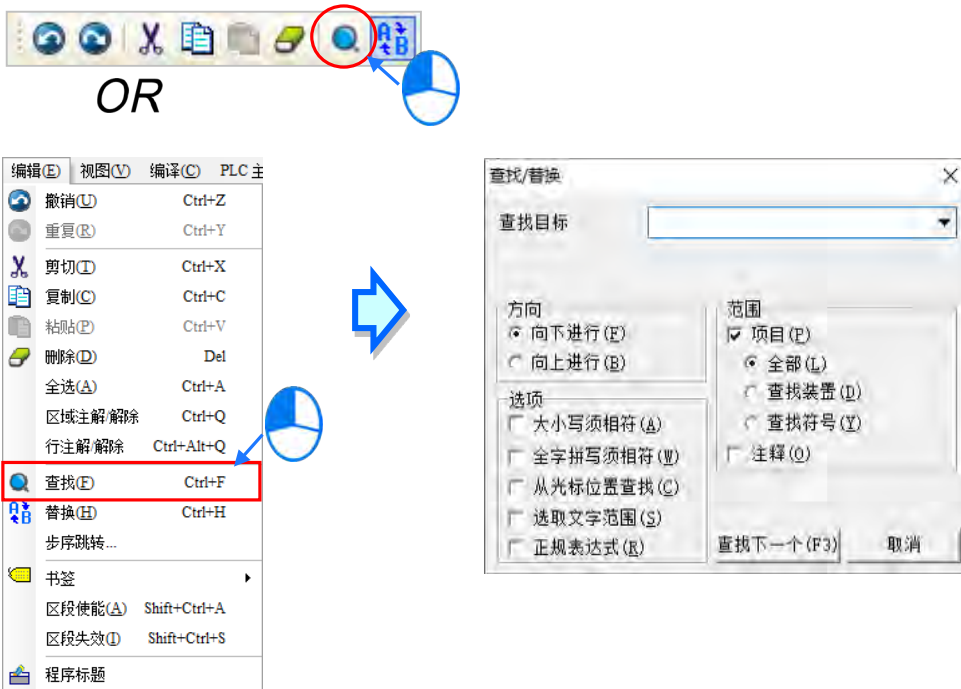
17.1.5 IL 与 ST 编辑环境的查找替换功能


在 IL 与 ST 的环境中只可使用查找与替换的功能。

(1) 请开启工作窗口，接着用鼠标左键单击程序编辑区。

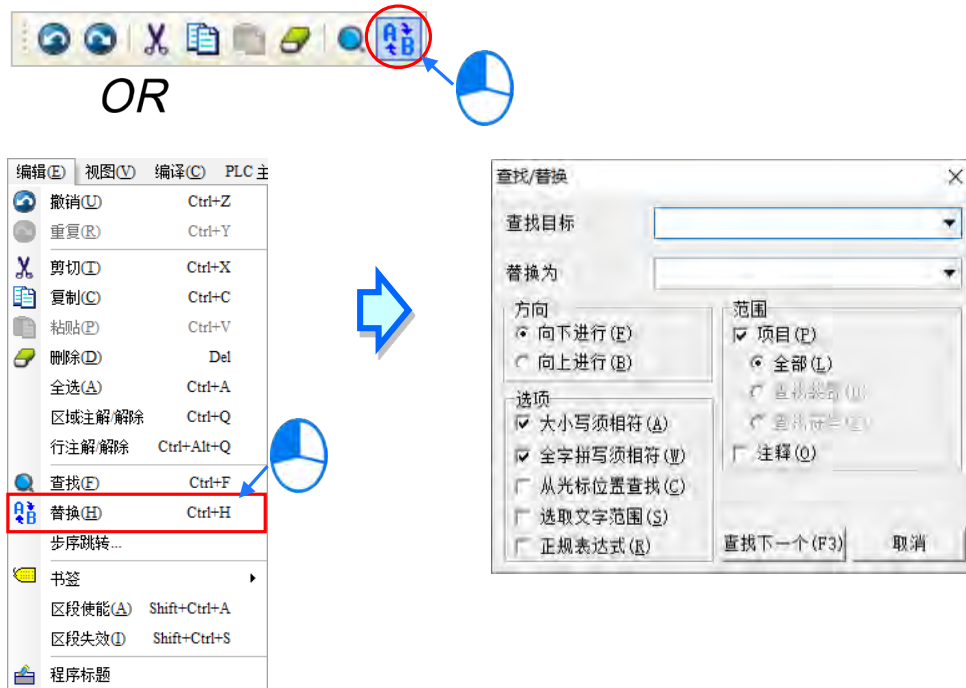


(2) 菜单工具栏单击 **编辑(E)** > **查找(F)** 或单击图标工具栏的  图标即可开启查找操作窗口。

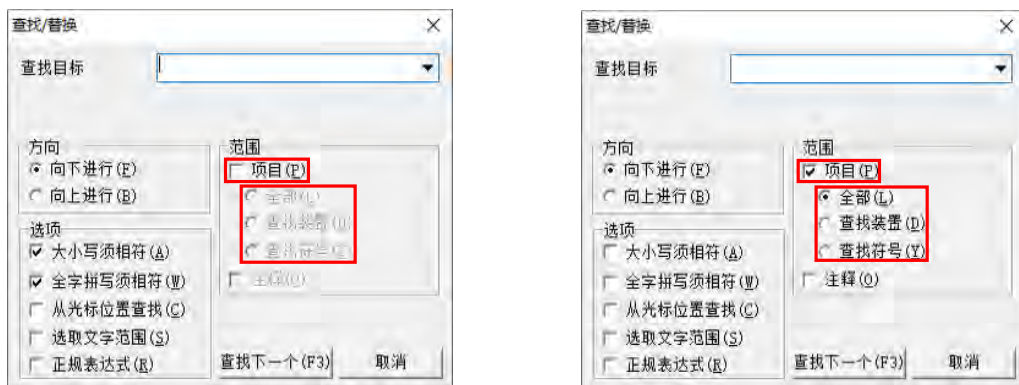


(3) 菜单工具栏单击 **编辑 (E)** > **替换 (H)** 或单击图标工具栏的  图标即可开启**替换**操作窗口。

17



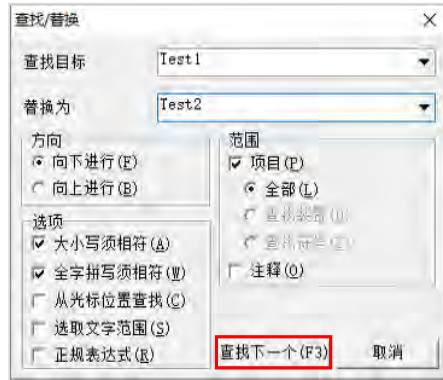
下图为**查找**功能的操作页面，必须设定查找目标，并于下列字段中选择查找条件，而设定完毕后按下「**查找下一个**」按钮或键盘的【F3】键便可逐一进行查找的工作，且即使关闭查找窗口后，按下键盘的【F3】键仍可继续查找最后一次的查找目标。当范围不勾选**项目**时，**查找**的范围仅限于目前正在编辑的窗口；当范围勾选**项目**时，可勾选**全部**或**查找装置**或**查找符号**，来进行分类查找，结果将列于信息显示区中。



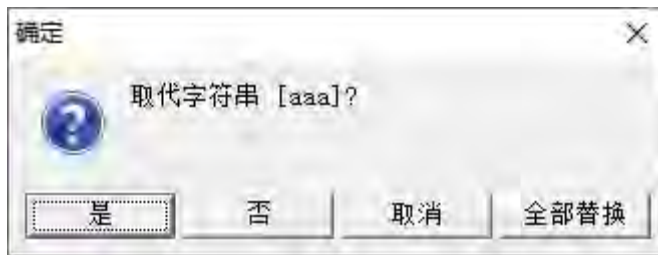
查找条件	说明
查找目标	直接输入查找目标，或是点击字段右侧的箭头以下拉挑选查找过的字符串。
方向	于此处选择查找的方向。
选项	<p>大小写须相符：比对字符串须与查找目标的大小写完全一致才符合查找条件。</p> <p>全字拼写须相符：比对字符串须与查找目标的内容完全一致才符合查找条件。若勾选全字拼写须相符时可再勾选复制注释至替换装置(全字拼写须相符)，将来源装置注释复制到目标的装置注释中。</p> <p>从光标位置查找：只由目前光标的位置开始查找（未勾选此项时，系统会查找整个程序）。</p> <p>选取文字范围：只由框选的文字范围开始查找（未勾选此项时，系统会查找整个程序）。</p> <p>正规表达式：勾选此项后，「查找目标」字段便可套用正规表示式的语法。</p>
范围	<p>查找时所包含的范围。</p> <p>（查找分页勾选「项目」时，「全部」、「查找装置」、「查找符号」才会使能。）</p> <p>全部：查找符合目标的所有字符串。</p> <p>查找装置：查找符合目标的所有装置。</p> <p>查找符号：查找符合目标的所有符号。</p> <p>注释：查找符合目标的所有注释。</p>

* 正规表示式 (Regular Expression) 为一种标准语法，例如使用「*」号代表通配符，相关细节请自行查询相关技术文件。

下图为**替换**功能的操作页面，输入查找目标，并于下列字段中选择查找条件；而执行替换操作时输入查找目标，并于「**替换为**」字段直接输入欲替换的字符串，设定完毕后按下「**查找下一个 (F3)**」按钮便可替换操作，在替换页面时勾选「**项目**」只有「**全部**」会使能。



当执行替换操作且找到符合条件的目标物时，画面将会出现询问窗口，而窗口中的按钮功能如下。



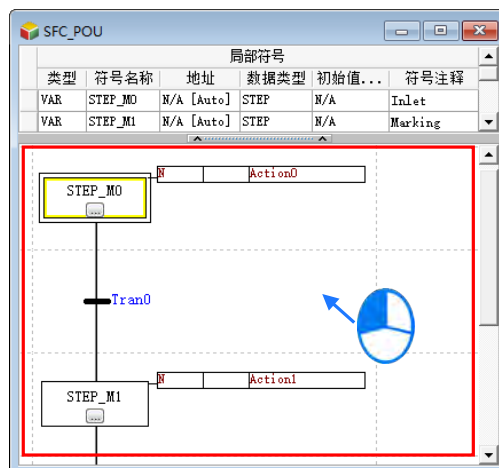
按钮	功能说明
是	将目前查找到的字符串变更为替换字符串，并继续查找下一个目标物。
否	不变更目前查找到的字符串，并继续查找下一个目标物。
取消	停止查找的操作并关闭此窗口
全部替换	将所有符合查找条件的目标字符串全部变更为替换字符串

17

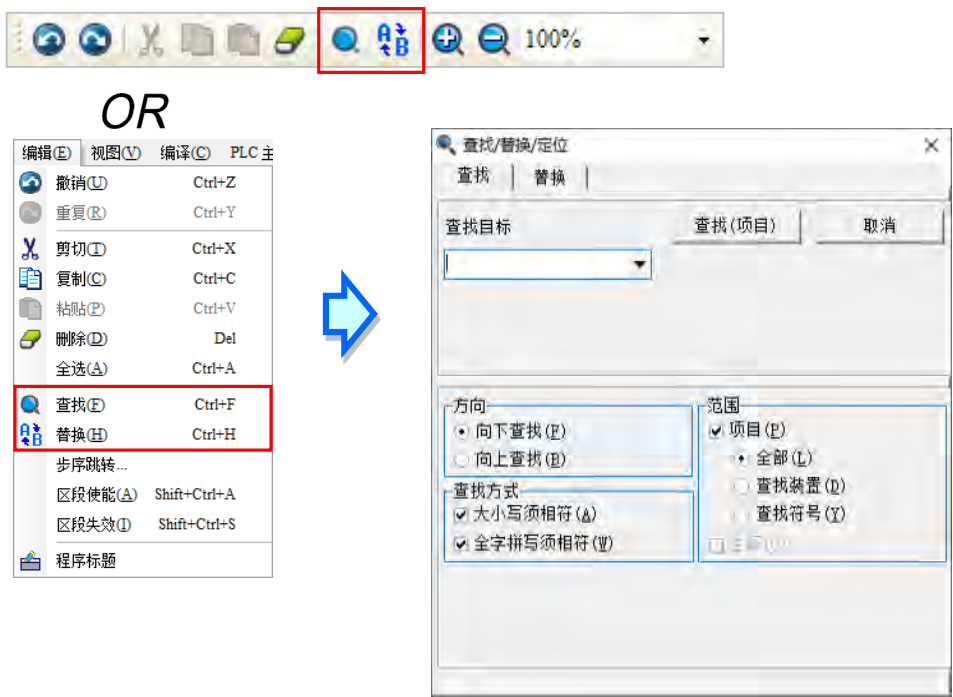
17.1.6 SFC 编辑环境的查找替换功能

在 SFC 的编辑环境中使用查找替换功能。

- (1) 请开启工作窗口，接着用鼠标左键单击程序编辑区。



(2) 在菜单工具栏的「编辑 (E)」项目下选取**查找、替换**，或选取图标工具栏中对应的图标，皆可打开**查找/替换/定位窗口**，也可藉由切换窗口上方的卷标来更换操作的功能。



下图为**替换**功能的操作页面，输入查找目标，并于「**替换为**」字段直接输入欲替换的字符串，按下「**替换**」按钮便可将目前所找到的目标物字符串变更为输入的替换字符串，而按下「**全部替换**」按钮则会一次替换所有符合查找条件的目标物；若勾选「**项目**」并点选「**替换 (项目)**」将会把项目内所有字符串进行替换。



查找条件	说明
查找目标	直接输入查找目标，或是点击字段右侧的箭头以下拉挑选查找过的字符串。
方向	于此处选择查找的方向。


查找条件	说明
选项	<p>大小写须相符：比对字符串须与查找目标的大小写完全一致才符合查找条件。</p> <p>全字拼写须相符：比对字符串须与查找目标的内容完全一致才符合查找条件。若勾选全字拼写须相符时可再勾选复制注释至置换装置(全字拼写须相符)，将来源装置注释复制到目标的装置注释中。</p>
范围	<p>查找时所包含的范围。</p> <p>(查找分页勾选「项目」时，「全部」、「查找装置」、「查找符号」才会使能。)</p> <p>全部：查找符合目标的所有字符串。</p> <p>查找装置：查找符合目标的所有装置。</p> <p>查找符号：查找符合目标的所有符号。</p> <p>注释：查找符合目标的所有注释。</p>

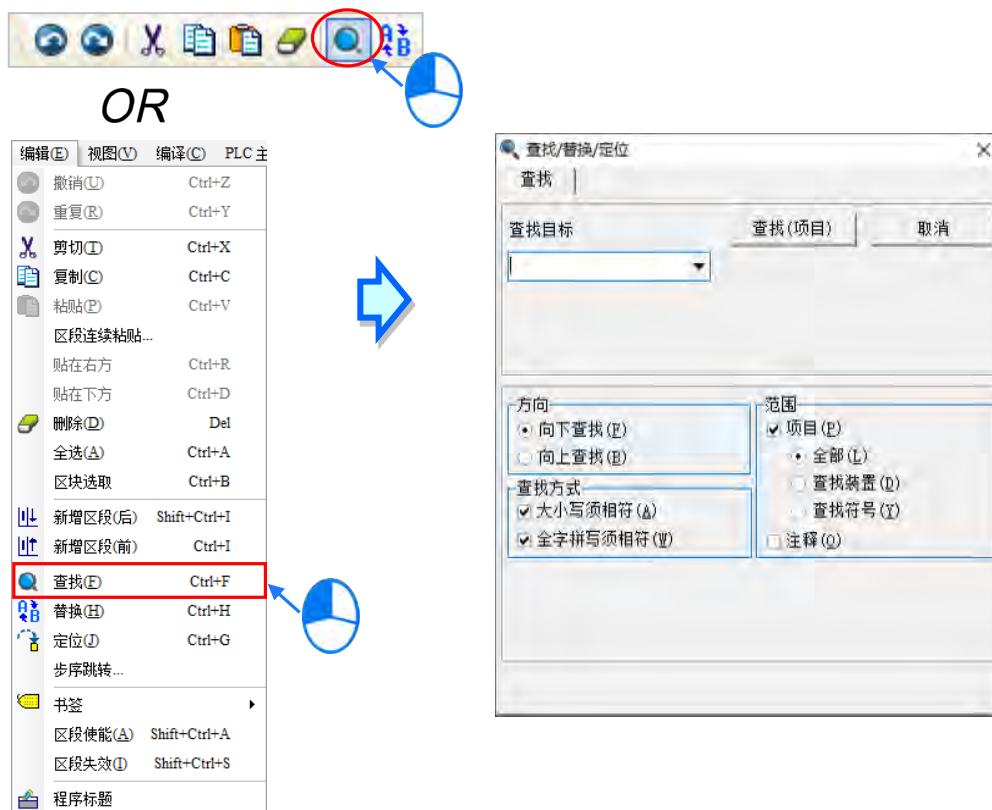
17.1.7 符号表的查找功能

在区域符号表与全局符号表只可使用**查找**的功能，只能对目前正在编辑的窗口进行操作。

(1) 开启要进行操作的编辑窗口后，在符号表上单击一下鼠标左键。



(2) 在菜单工具栏单击 **编辑 (E)** > **查找 (F)** 或单击图标工具栏中的  图标即可开启操作窗口。

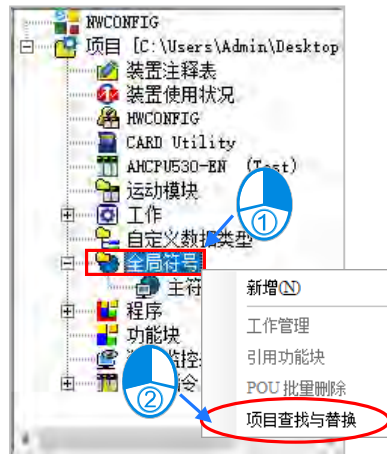


在范围选取整体项目后按下「**查找(项目)**」按钮即可快速在整体项目查找到所需要的内容，也可以不选取整体项目后按下「**查找下一个(F3)**」按钮或键盘的【F3】键便可逐一进行查找的工作，且即使关闭查找窗口后，按下键盘的【F3】键仍可继续查找最后一次查找的目标物。

查找条件	说明
查找目标	直接输入查找目标，或是点击字段右侧的箭头以下拉挑选查找过的字符串。
方向	于此处选择查找的方向。
选项	<p>大小写须相符：比对字符串须与查找目标的大小写完全一致才符合查找条件。</p> <p>全字拼写须相符：比对字符串须与查找目标的内容完全一致才符合查找条件。若勾选全字拼写须相符时可再勾选复制注释至置换装置(全字拼写须相符)，将来源装置注释复制到目标的装置注释中。</p>
范围	<p>查找时所包含的范围。</p> <p>(查找分页勾选「项目」时，「全部」、「查找装置」、「查找符号」才会使能。)</p> <p>全部：查找符合目标的所有字符串。</p> <p>查找装置：查找符合目标的所有装置。</p> <p>查找符号：查找符合目标的所有符号。</p> <p>注释：查找符合目标的所有注释。</p>

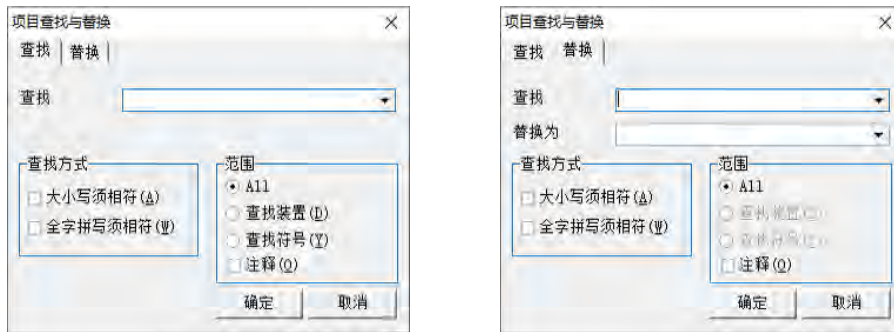
17.1.8 项目查找与替换

在项目管理区的**全局符号表**、**程序**或**功能块**项目上按下鼠标右键，接着在快捷选单中单击「**项目查找/替换**」，可打开此功能，用于查找/替换项目中的某个装置或符号。



下图为**查找与替换**功能的操作页面，可藉由窗口上方的卷标来进行切换，必须设定查找目标，并于下列字段中选择查找条件，且只有查找可以使用「**查找装置**」与「**查找符号**」选项，设定完毕后，若在「**查找**」的页面中按下「**确定**」键，接着系统便会开始进行查找的工作；而若在「**替换**」页面中按下「**确定**」键，则系统便会在完成查找后，自动将符合条件的目标物字符串全部变更为「**替换为**」字段中的字符串，且此动作将无法还原。

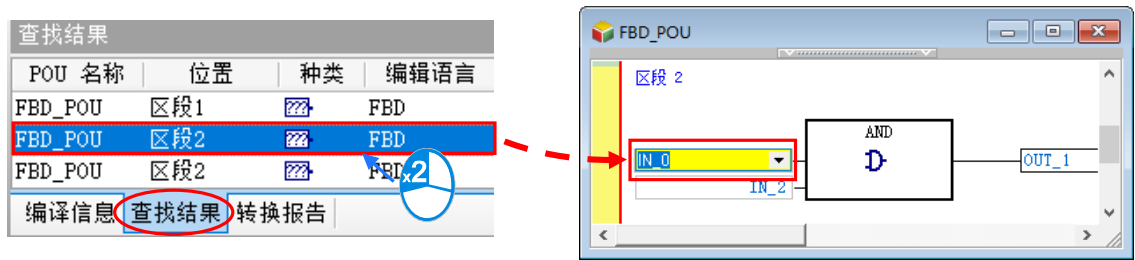
17



查找条件	说明
范围	查找装置：查找符合目标的所有装置。 查找符号：查找符合目标的所有符号。 注释：查找符合目标的所有注释。
选项	大小写须相符：比对字符串须与查找目标的大小写完全一致才符合查找条件。 全字拼写须相符：比对字符串须与查找目标的内容完全一致才符合查找条件。

无论是**查找**或**替换**的操作，所有查找的结果将会显示在 ISPSOft 窗口下方的**信息显示区**，且于某个查

找结果的对象上双击鼠标左键后，系统便会自动引导用户移动至该目标物的所在位置。

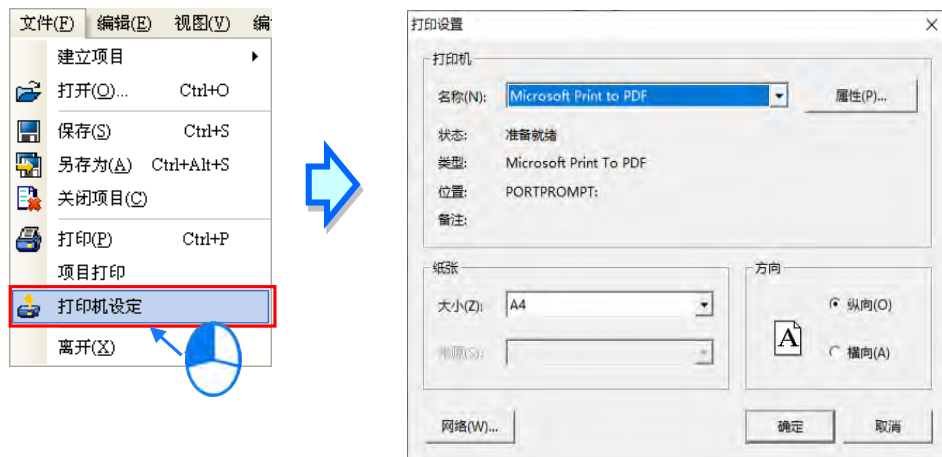


17.1.9 打印功能

ISPSOft 对于编辑工作区中的内容亦提供了打印的功能，相关说明如下。

● 打印机设定

菜单工具栏中单击 **文件 (F) > 打印机设定**，之后便可开启打印机的设定窗口，在打印文件之前可先于此选择要使用的列表机与打印格式。



● 项目打印

菜单工具栏中单击 **文件 (F) > 项目打印**，之后便可勾选要打印的项目，可选的项目包含程序、功能块与全局符号表，而完成后按下「确定」键即可依据目前的打印机设定将所选的项目打印出来。



下图为打印的结果，打印内容会包含区域符号表，而注释部分则会依据当时窗口的注释显示状态。

宣告类别	符號	位址	型態	初值	符號註解
VAR	ET_INS_1	N/A [Auto]	EQUIP_TRIG	N/A	
VAR	ET_INS_2	N/A [Auto]	EQUIP_TRIG	N/A	

區段 1									
RUNNING	<table border="1"> <tr> <td>ET_INS_1</td> <td></td> </tr> <tr> <td>EQUIP_TRIG</td> <td></td> </tr> <tr> <td>En</td> <td>Eno</td> </tr> <tr> <td>InP_SNR_1</td> <td>SNR_IN TRIG TRIG_1</td> </tr> </table>	ET_INS_1		EQUIP_TRIG		En	Eno	InP_SNR_1	SNR_IN TRIG TRIG_1
ET_INS_1									
EQUIP_TRIG									
En	Eno								
InP_SNR_1	SNR_IN TRIG TRIG_1								

17

● 窗口打印

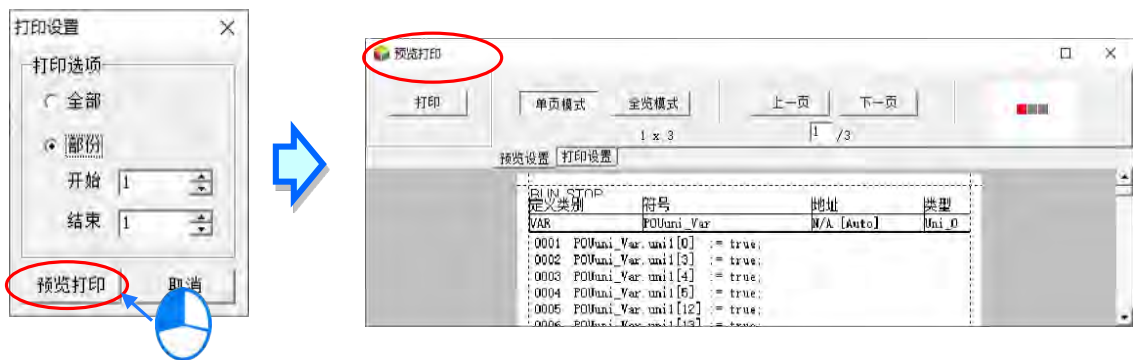
若只要打印目前正在编辑的窗口内容时，请于菜单工具栏中单击 **文件 (F) > 打印 (P)**，或直接单击图标工具栏中的 图标，之后便会开启打印区域的选择窗口，而根据编辑窗口内容的不同，选择之后所打印的结果也会不同，请参考下表。



编辑窗口	选择全部范围	选择部分范围
全局符号表	打印完整的全局符号表	全局符号表无法选择打印部分范围
LD 与 FBD 程序	打印完整的程序与区域符号表	打印完整区域符号表及选择范围内的程序 (所设定的 起始 与 结束 编号代表区段编号)
IL 与 ST 程序	打印完整的程序与区域符号表	打印完整区域符号表及选择范围内的程序 (所设定的 起始 与 结束 编号代表程序行号)
SFC 程序	打印完整的区域符号与 SFC 图面 (不包含 Actions 与 Transitions)	SFC 程序无法选择打印部分范围
CFC 程序	打印完整的程序与区域符号表	CFC 程序无法选择打印部分范围

*.Actions 与 Transitions 的内容仍必须在开启其窗口之后才可进行打印。

设定完成后按下「预览打印」即可开启打印管理工具，而关于**打印管理工具**的说明则请参考 附录 C。



17.2 装置与寄存器管理

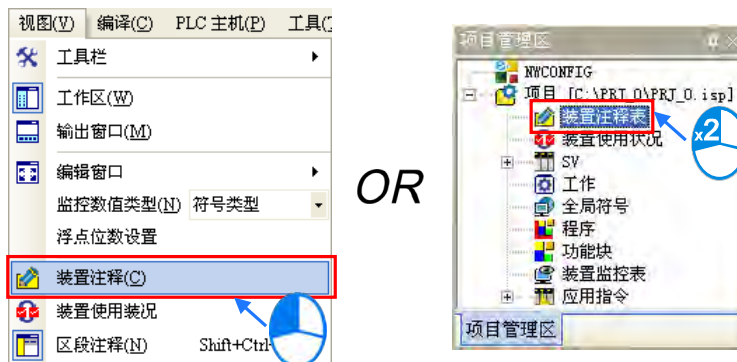
17.2.1 DVP 机种的装置注解表

进行 DVP 机种的项目开发时，于菜单工具栏单击 **文件(F)** > **导出** > **导出装置注解...**，之后便可将此项目中的装置注解导出；而要重新导入时，只要单击 **文件(F)** > **导入** > **导入装置注解...**即可。

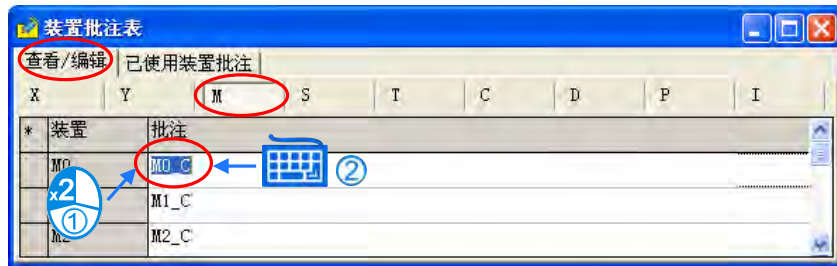


当要批次对装置注解进行管理时，请于菜单工具栏中单击 **视图(V)** > **装置注解(C)**，或直接在项目管理区的「装置注解表」项目上双击鼠标左键，之后系统便会自动开启装置注解的管理窗口。

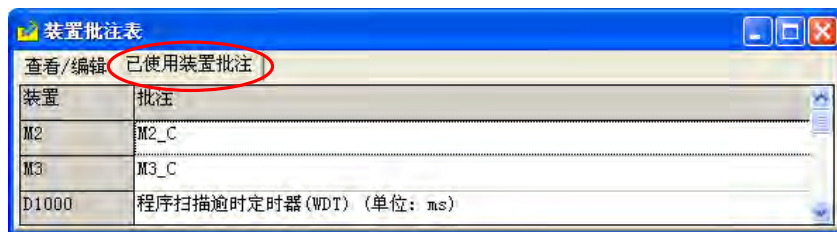
17



「查看/编辑」页面会列出所有装置的批注。先在页面上方选择装置类别后，页面中便会列出该装置范围的所有批注状况，而当装置名称的左侧出现「*」符号时，代表该装置已在程序当中被使用。当要编辑批注时，只要在要编辑的字段上双击鼠标左键，之后再以键盘输入批注内容即可。

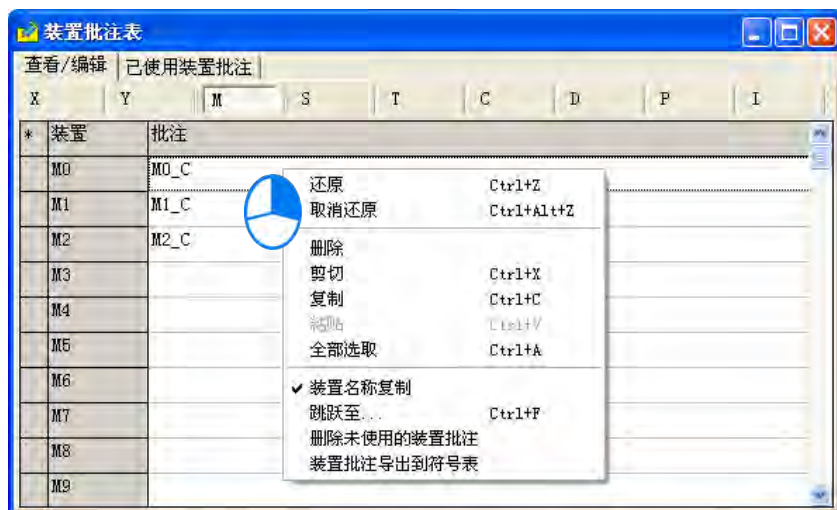


「已使用装置批注」页面则不分装置类别，全部有设定装置批注的装置与其批注都会全部列出，且于此页面中同样可对各装置的批注进行编辑，其方法则与前述「查看/编辑」页面中的编辑方式相同。



- *1. 装置批注与符号批注是彼此独立的，而此表显示的仅是装置批注的部份，变量符号的批注并不会显示在列表当中。
- *2. 特殊继电器与特殊寄存器本身已具有默认的批注，但仍可重新编辑，不过重新编辑的批注只限在这个项目中才有效。

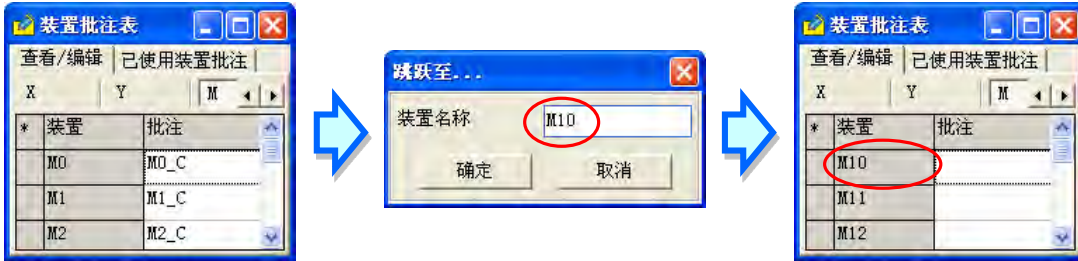
列表中按下右键后，弹出的快捷选单中除了基本的 Windows 操作之外，亦包含许多专属的功能。



● 跳转至

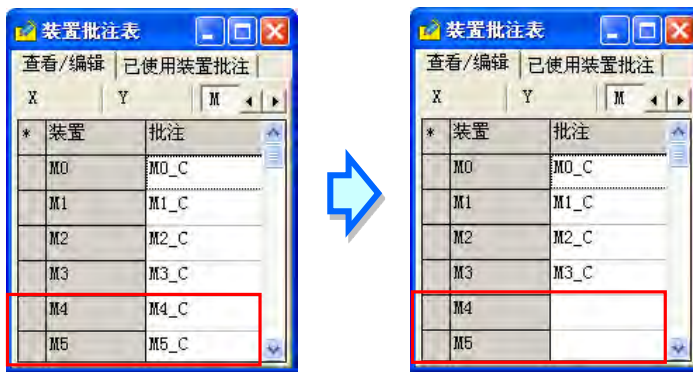
单击此功能后画面便会出现一窗口，在「装置名称」字段输入要检视的装置名称后按下「确定」键，之后列表的显示便会立即调整至可检视该装置的范围。

输入设备名称时，装置的类别必须与目前的页面一致，否则便会找不到该装置；而于「已使用装置批注」页面中使用此功能时，输入的装置名称也必须确实位于该清单之中才可进行操作。



● 删除未使用的装置批注

单击此项后，对于未使用于程序当中的装置，系统会自动清除这些装置的批注。



● 装置名称复制

当选取批注表中的项目，并执行复制或剪下的功能后，该项目便会暂存于 Windows 的剪贴簿中，并且可于其他的软件中被贴上，如 EXCEL、记事本...；而在操作前，若「装置名称复制」的项目有被勾选时，复制或剪下的内容便会自动包含装置的名称，请参考下列两个 EXCEL 表的比较。

装置批注表

X	Y	M	S	T	C	D	P	I
*	装置	批注						
	M0	M0_C						
	M1	M1_C						
	M2	M2_C						
	M3	M3_C						
	M4							

还原 Ctrl+Z
取消还原 Ctrl+Alt+Z
删除
剪切 Ctrl+X
复制 Ctrl+C
粘贴 Ctrl+V

✓ 装置名称复制

	A	B	C
1	M0	M0_C	
2	M1	M1_C	
3	M2	M2_C	
4	M3	M3_C	
5			

装置名称复制

	A	B
1	M0_C	
2	M1_C	
3	M2_C	
4	M3_C	
5		

● 装置批注导出到符号表

当已为变量符号配置装置地址时，单击此项后，装置批注便可导出至对应该装置地址的符号批注。

装置批注表

X	Y	M
*	装置	批注
	M0	M0_C
	M1	M1_C
	M2	M2_C

局部符号

类型	符号名称	地址	数据类型	初始值	符号批注
VAR	VAR_0	M0	BOOL	FALSE	VAR_OC
VAR	VAR_1	N/A [Auto]	BOOL	FALSE	VAR_IC

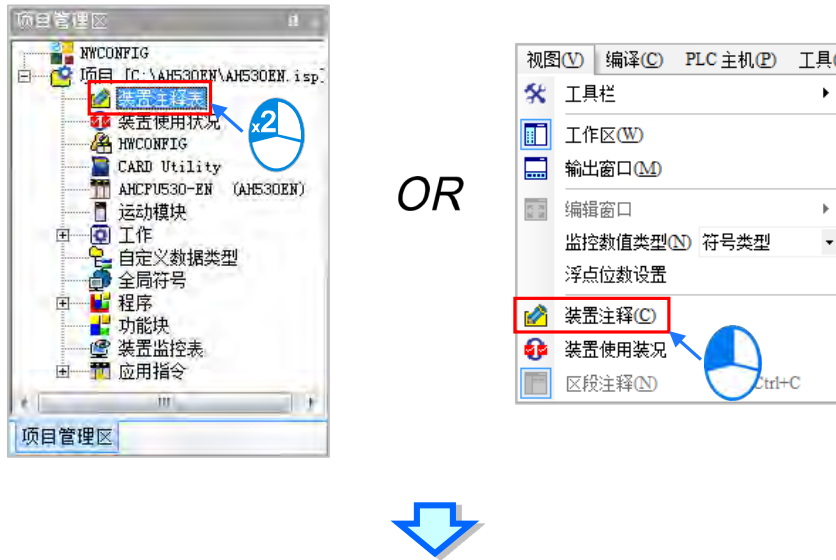
局部符号

类型	符号名称	地址	数据类型	初始值	符号批注
VAR	VAR_0	M0	BOOL	FALSE	M0_C
VAR	VAR_1	N/A [Auto]	BOOL	FALSE	VAR_IC

17.2.2 AH/AS 系列机种的装置注解表

AH/AS 系列机种本身亦提供装置管理的功能，在功能工具列中点击 **检视 (V) > 装置注解 (C)**，或在项目管理区的「**装置注解表**」项目上双击鼠标左键，之后便可开启管理窗口。

管理窗口总共包含两个页面，且由窗口上方的标签便可进行切换。但若开启此窗口后曾经变更程序内容或修改注解时，请先关闭窗口并重新编译后再开启窗口，如此系统才可取得最新的信息。



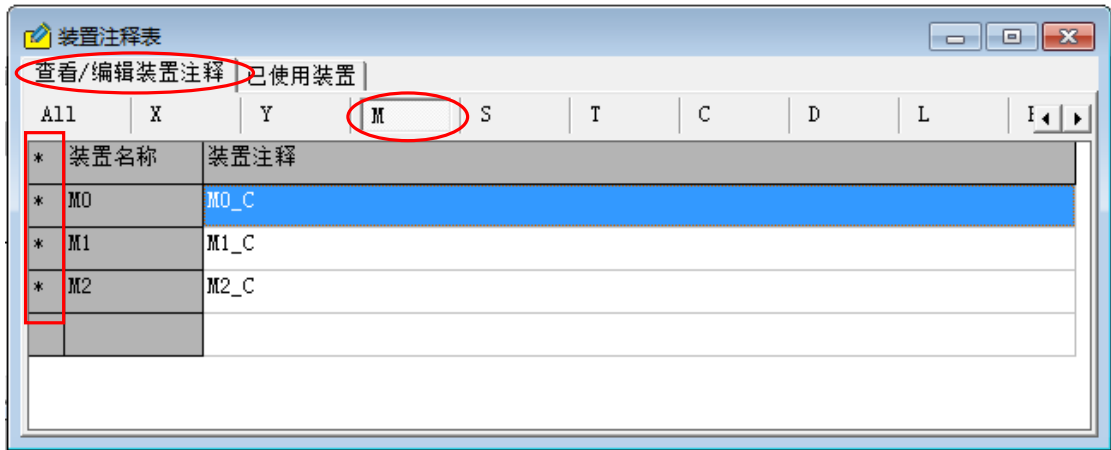
装置注解表

查看/编辑装置注解 | 已使用装置

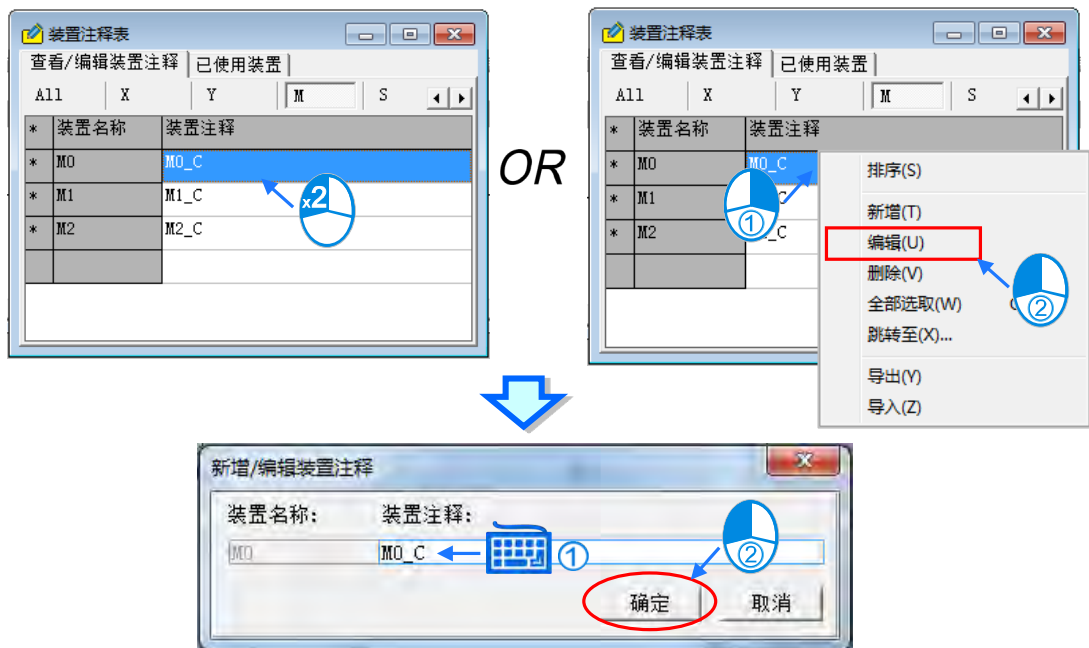
All	X	Y	M	S	T	C	D	L	I	
*	装置名称	装置注释								
*	M0	M0_C								
*	M1	M1_C								
*	M2	M2_C								
	SM0	运算错误								
	SM1	运算错误锁定								
	SM5	指令/操作数检查错误								
	SM8	逾时监视错误								
	SM9	系统错误								
	SM10	I/O总线错误								

17

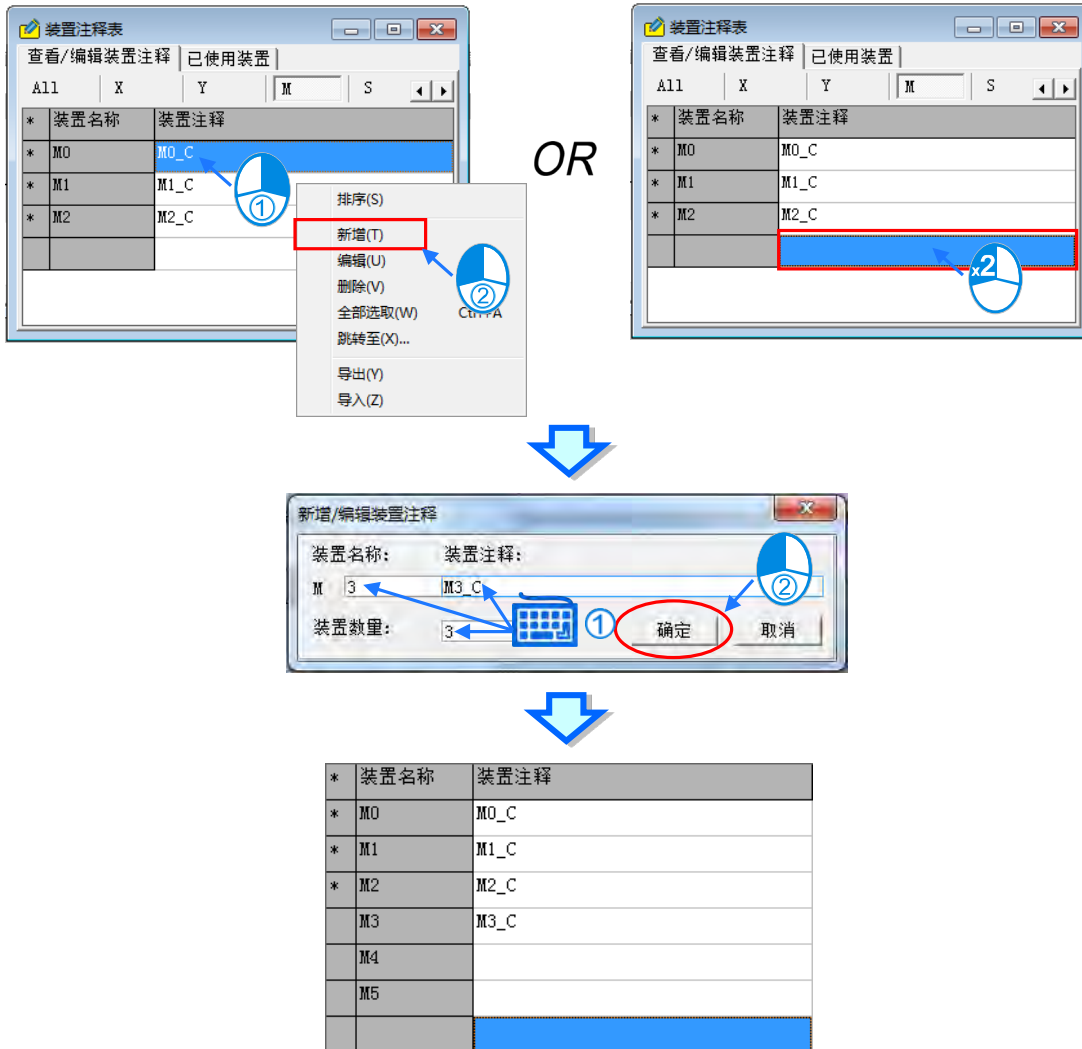
「**检视/编辑装置注解**」页面会列出所有已加上装置注解的装置清单及其注解内容，而切换此页面上方的标签则可选择要检视的装置类别，其中选择「**All**」代表不分类别，该页面会列出所有已加上装置注解的装置。此外，当装置名称的左侧出现「*」符号时，代表该装置已在程序当中被使用。



用户亦可直接于「**检视/编辑装置注解**」页面修改或新增装置注解。要修改装置注解时，请于要修改的项目上双击鼠标左键，或是按下右键后于选项点击「**编辑**」，之后便可在编辑窗口中进行修改。而其中 SR 与 SM 本身便会有系统预设的装置注解，不过用户还是可以重新编辑，但重新编辑后的注解则只限在这个项目中才有效。

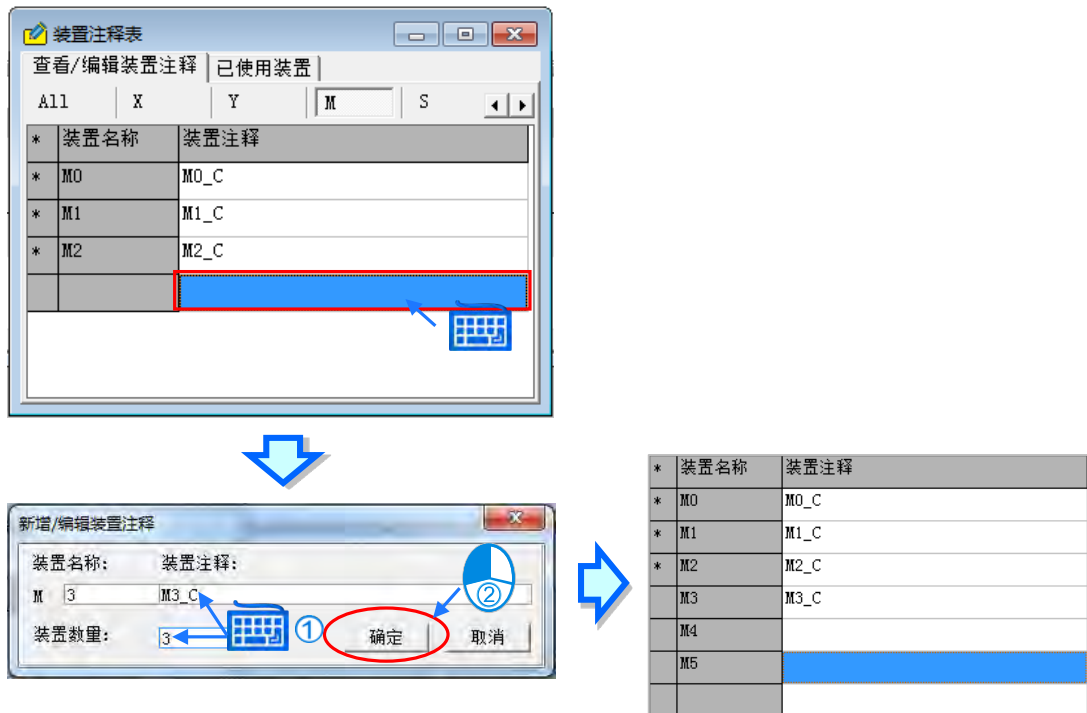


要新增装置注解时，请按下鼠标右键后于选项中点击「新增」，或直接于选项中双击鼠标左键，之后画面便会自动开启编辑窗口，接着直接于编辑窗口中输入装置位置及注解即可，窗口底部的「装置数量」，是让用户一次产生多笔装置注解栏位来用，如下图。

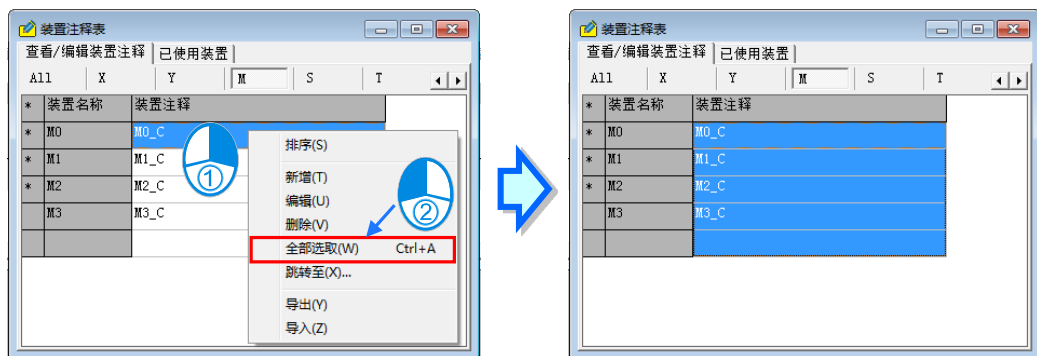


17

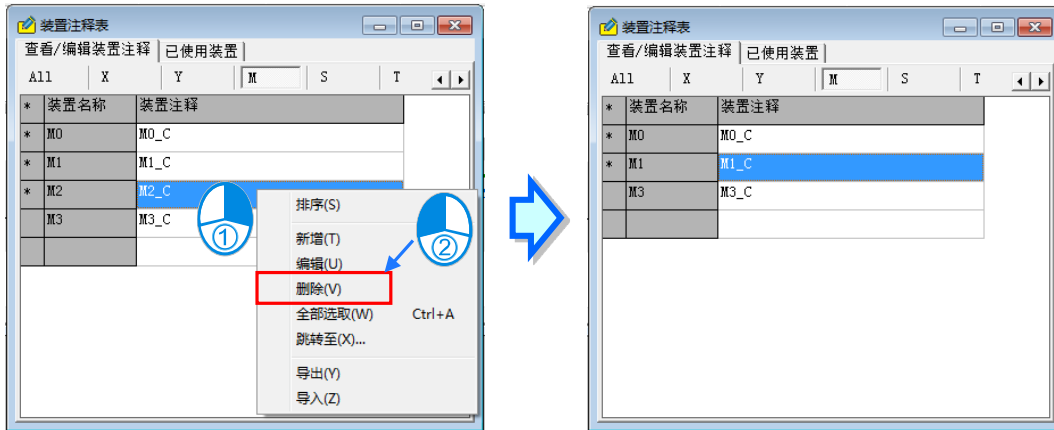
用户也可以直接在此页面以键盘输入装置位置，之后画面便会自动开启编辑窗口，接着直接于编辑窗口中输入注解以及「装置数量」即可，如下图。



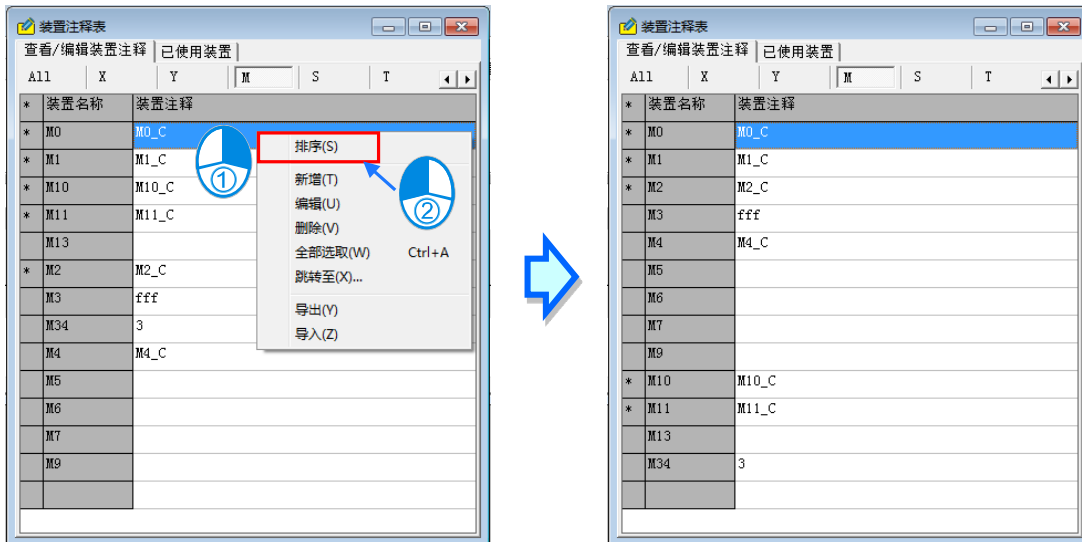
要移除装置注解时，请先点击要移除注解的项目；若要删除大量注解时，可先点击第一个项目后按下键盘的【Shift】键，接着再点击最后一个项目，之后该范围的项目便会被选取起来；而要选择整个页面的装置注解时，请直接在该页面中按下鼠标右键，之后于快捷选中单中点击「全部选取」即可。



选取要删除的项目后，接着请直接按下键盘的【DEL】键，或是在按下鼠标右键后于快捷选项中点击「删除」，之后便可将所选取的装置注解全部清除。当删除项目包含 SM 或 SR 的注解时，系统的处理动作则是将被选取的 SM 或 SR 的装置注解恢复为系统预设值，而非将该项目删除。

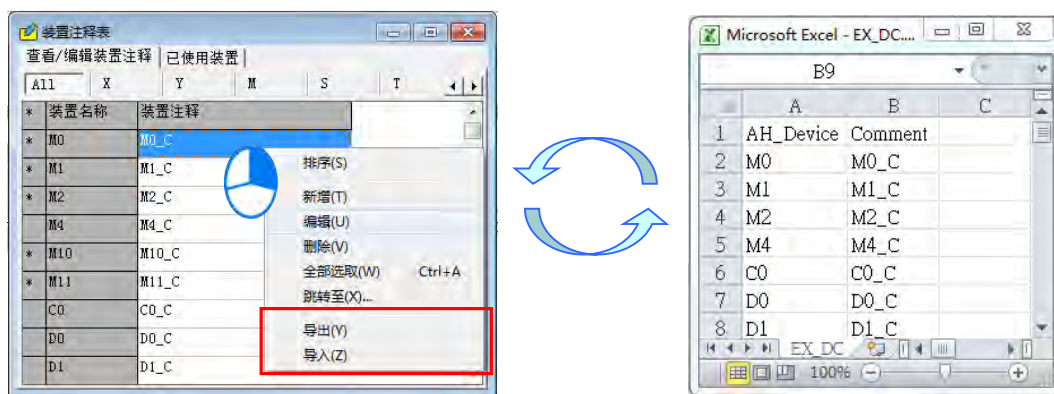


若要装置注解表中的项目重新排序时，请直接按下鼠标右键，接着于快捷选单中点击「排序」，之后所有项目便会依据装置地址而重新进行排序。



装置注解表同样可以 CSV 档的格式进行**导出与导入**的操作，用户可自行通过 EXCEL 来进行编辑与管理；而进行操作时，不分目前开启的页面，系统将会针对所有的装置注解进行操作，不过针对 SM 与 SR 的部份，导出时仅会包含用户重新定义的项目，而保持系统预设的部份将不会被导出；同样的，当导入的文件中有包含 SM 或 SR 的项目时，导入后亦只有该项目所对应的 SM 与 SR 注解会被变更，其余部分则仍会保留为系统预设值。

要进行「**导出**」的操作时，请直接按下鼠标右键，接着于快捷选项中点击「**导出**」，之后再于文件视窗设定欲导出的档名与路径即可；同样的，要进行「**导入**」的操作时，请直接按下鼠标右键，接着于快捷选项中点击「**导入**」，之后再于文件窗口选择欲导入的文件即可。



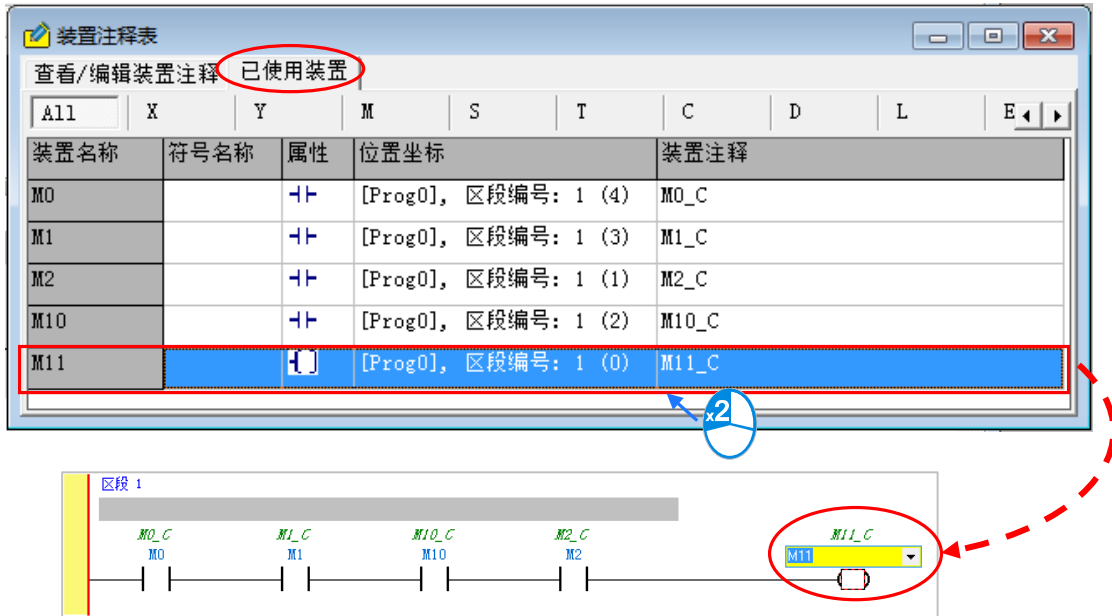
欲快速定位到某个装置时，按下鼠标右键，接着于快捷选项中点击「**跳转至**」后画面会出现一窗口，于「**装置名称**」栏位输入欲检视的装置名称后按下「**确定**」键，之后列表的显示便会立即调整至可检视该装置的范围。

输入装置名称时，装置的类别必须与目前的页面一致，否则便会找不到该装置；而于「**已使用装置**」页面中使用此功能时，输入的装置名称也必须确实位于该清单之中才可进行操作。



17

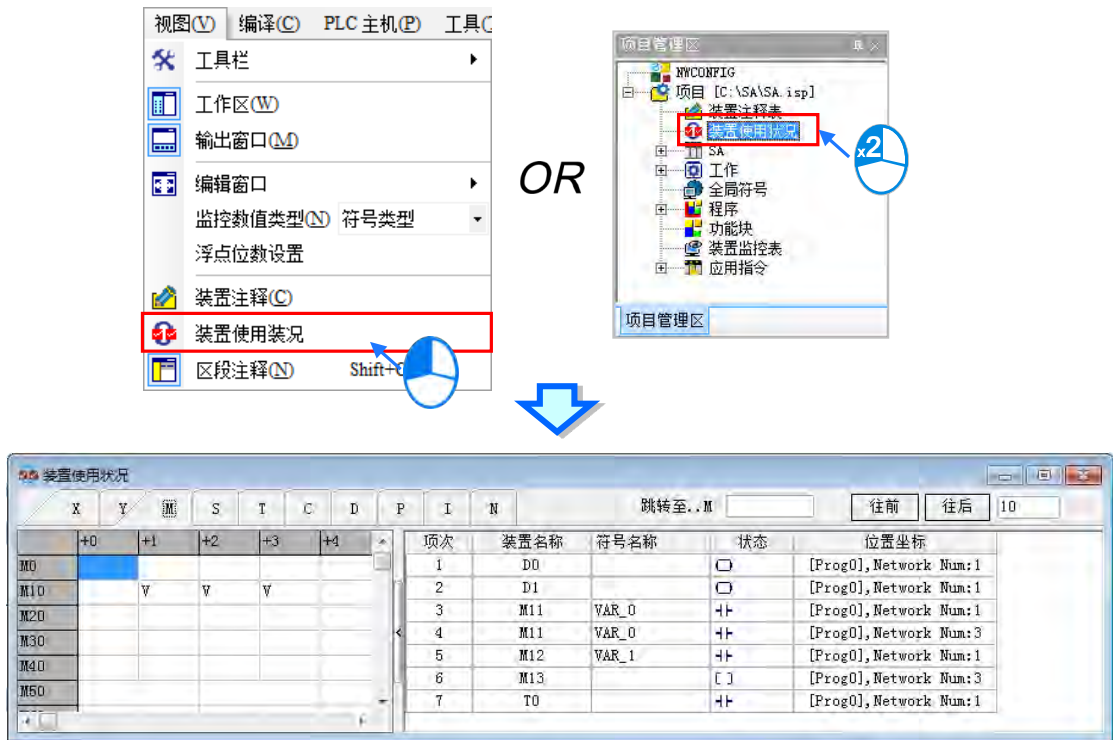
「已使用装置」页面会列出所有已在程序中被使用的装置清单与该装置的相关信息，而同样也可藉由切换此页面上方的标签来选择欲检视的装置类别。当在此页面的某个项目上双击鼠标左键时，系统便会自动引导用户移至对应该项目的所在位置。



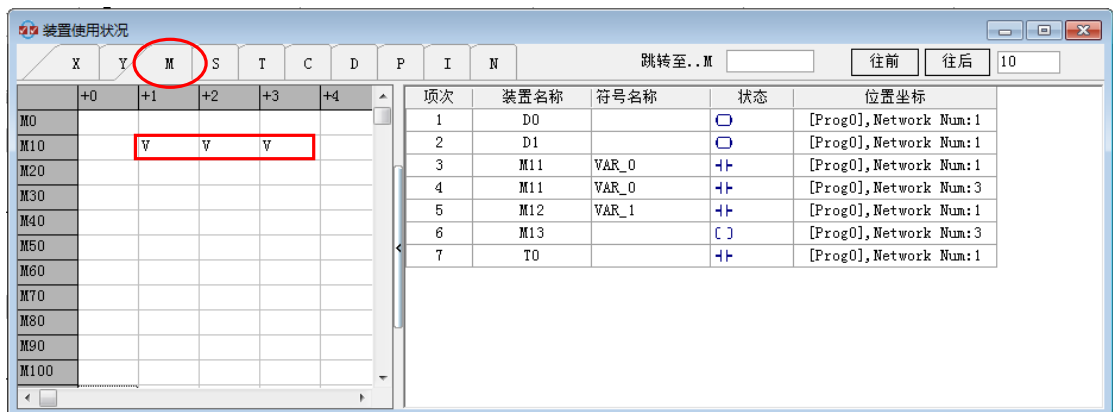
17.2.3 装置使用状况

在编辑项目时，若利用此功能便可查询项目中的装置使用情形；不过若使用 DVP 系列机种，执行此功能之前必须先完成编译的程序，此外，DVP 系列机种项目在开启此表后曾经变更程式内容时，请先关闭表格并重新编译后再开启此表，如此表格中才可取得最新的信息。

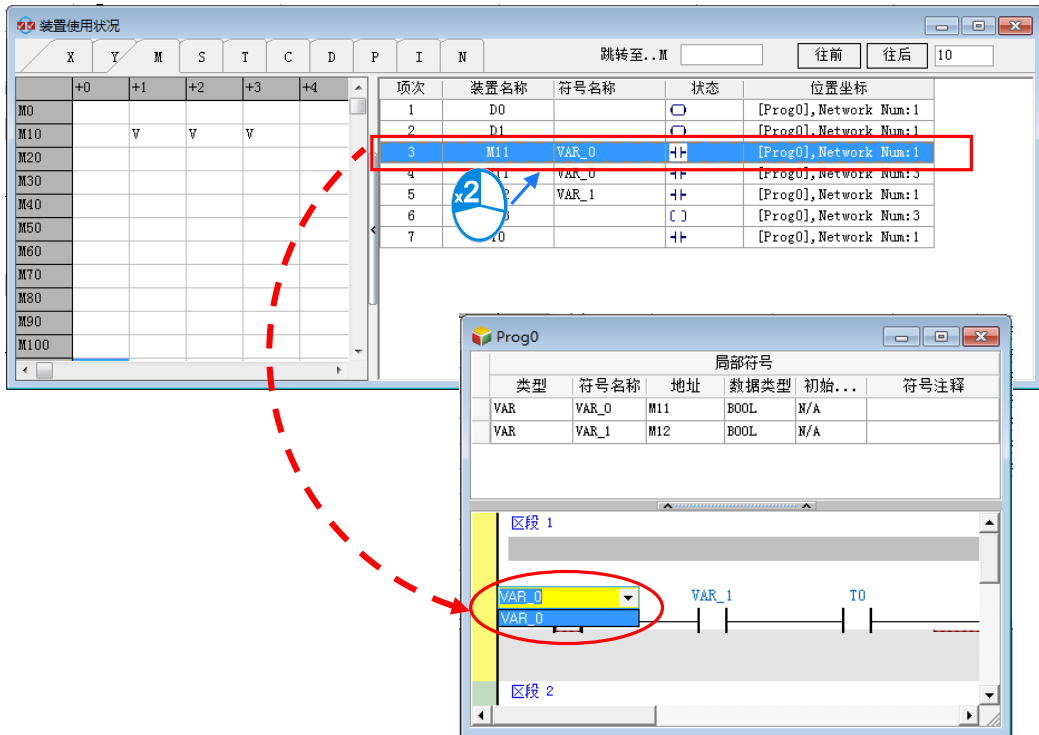
在功能工具列中点击 **检视 (V)** > **装置使用状况**，或直接在项目管理区的「装置使用状况」项目上双击鼠标左键，之后便可开启管理窗口。



管理窗口主要分为两个部份，其中左侧为装置的总表，点击窗口上方的标签便可切换欲检视的装置类别，其中有打勾的栏位便代表该装置已被用于程式当中，如下图的 M11、M12 与 M13。



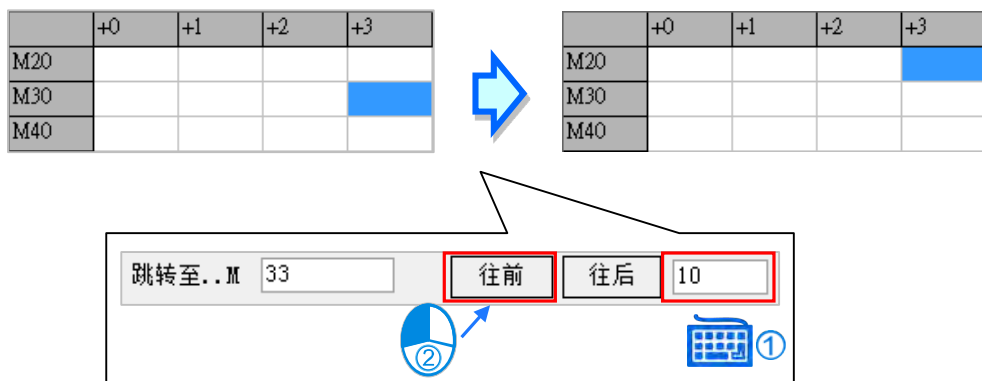
当于右侧清单的项目上双击鼠标左键时，系统便会自动引导用户移至对应项目的所在位置。

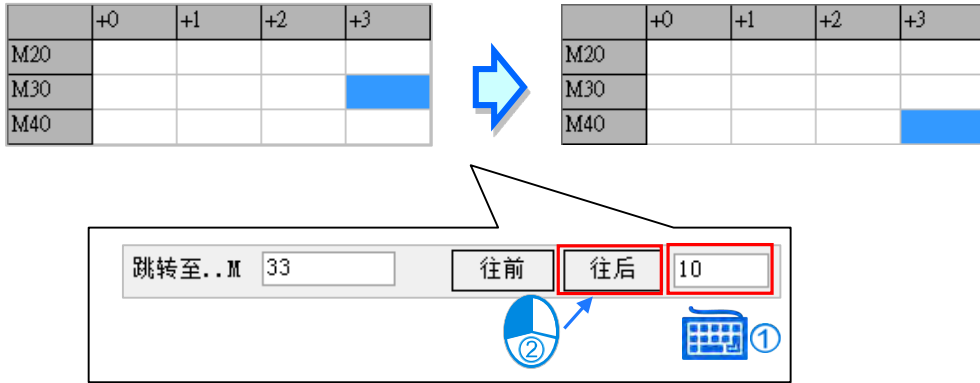


于快速移动栏位输入装置编号后按下【Enter】键，左侧表格的蓝色游标便会移动至该编号的位置。



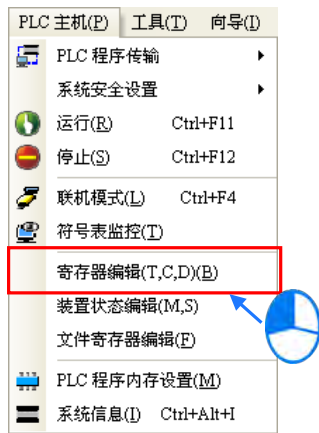
若用户欲前后移动蓝色游标至指定编号，可先填写欲移动数量再点击往前/往后按键，来做前后移动。





17.2.4 寄存器编辑

ISPSOft 提供工具用以批次编辑 T、C、D、L 装置的内容值，并可将其做为配方而进行保存与下载的操作。开启项目后，于菜单工具栏单击 **PLC 主机(P)** > **寄存器编辑**，之后画面便会开启编辑窗口。



17

开启窗口后，表格中所带入的装置数值并非由主机内部所上载回来的目前值，而是上次开启此窗口时所保存的编辑内容，但若是在此项目中第一次开启窗口时，表格的内容便会全部以 0 做为默认值。操作时，藉由切换窗口上方的卷标便可选择欲检视的装置类别。



要编辑装置内容时，请先单击欲编辑的装置字段后，直接以键盘输入数值即可。

	+0	+1	+2
D0	0	0	0
D10	0	0	0
D20	0	0	0
D30	0	0	0
D40	0	0	0

	+0	+1	+2
D0	1234	0	0
D10	0	0	0
D20	0	0	0
D30	0	0	0
D40	0	0	0

「显示模式」区域中可选择表格中的数值显示格式。

数值型态

16位

32位

显示模式

10进制

16进制

2进制

浮点数

	+0	+1	+2	+3
D0	-1	-1	0	0
D10	0	0	0	0

数值型态

16位

32位

显示模式

10进制

16进制

2进制

浮点数

	+0	+1
D0	11111111 11111111	11111111 11111111
D10	00000000 00000000	00000000 00000000

「数值类型」区域中则可选择数值的表达格式为 16 位或 32 位；当切换为 32 位时，每个装置中的数值将自动包含下一个装置地址的内容而以 32 位来呈现。

以下图为例，D0 的 16 进制值为 AAAA，D1 的 16 进制值为 BBBB，D2 的 16 进制值为 CCCC；则当切换为 32 位显示时，D0 会自动包含 D1 的值而以 32 位显示为 BBBBAAAA，D1 则包含 D2 的值而显示为 CCCCBBBB，而 D2 的显示则因为 D3 的值为 0，所以在包含 D3 的值后会显示为 CCCC。

数值型态

16位

32位

显示模式

10进制

16进制

2进制

浮点数

	+0	+1	+2	+3
D0	AAAA	BBBB	CCCC	0
D10	0	0	0	0

数值型态

16位

32位

显示模式

10进制

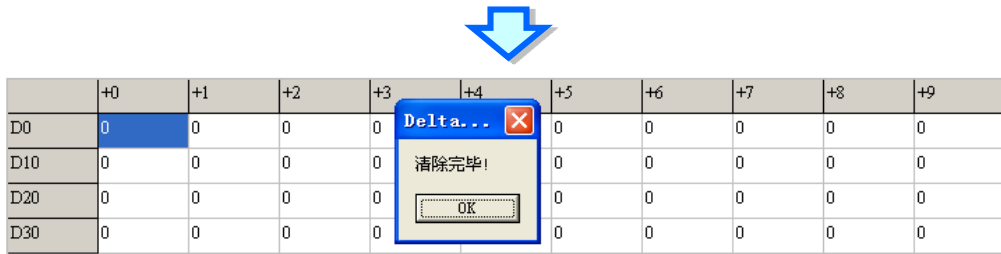
16进制

2进制

浮点数

	+0	+1	+2
D0	BBBBAAAA	CCCCBBBB	CCCC
D10	0	0	0

当按下窗口中的「全部清除」按钮时，此页面中的所有装置内容便会被清除为 0；不过此操作仅会清除目前页面中的装置数值，其他页面中的装置数值则会保持原状。



除此之外，亦可于页面中按下鼠标右键，接着于快捷选单中单击「全选」，之后再按下鼠标右键，并于选单中单击「清除为 0」便可将该页面中的所有的装置内容清除。

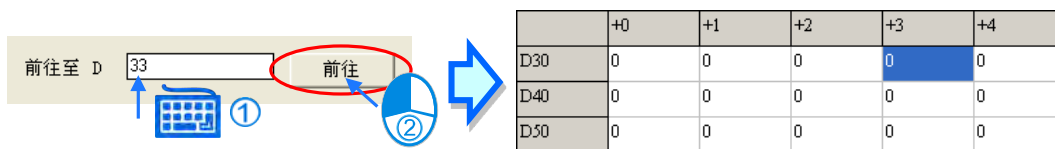
17



若只想清除部份装置时，可先使用鼠标框选欲清除的范围后，按下鼠标右键并于快捷选单中单击「清除为 0」，之后框选范围内的装置数值便会被清除。

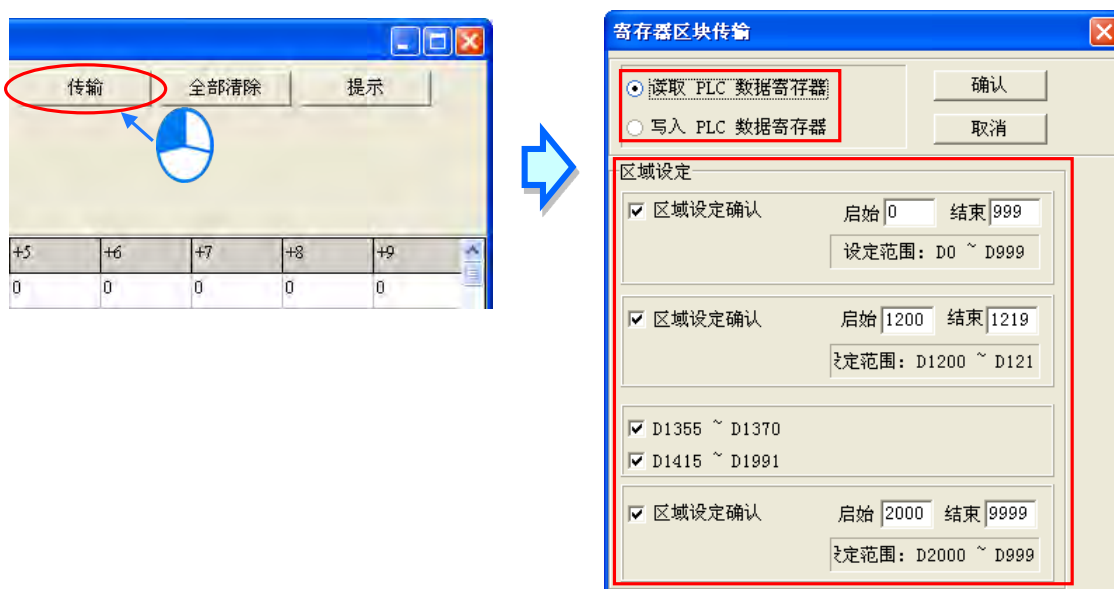


前往至字段输入设备编号后按下前往按钮，下方表格的蓝色光标便会移动至该编号的位置。



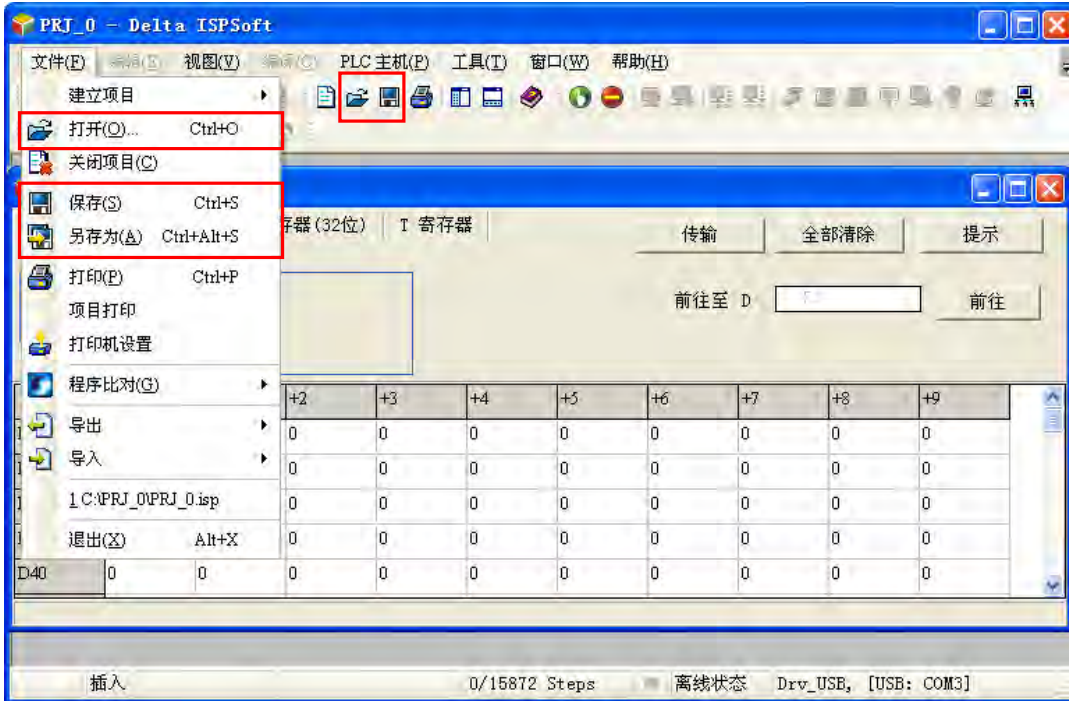
当按下窗口中的「传输」按钮后，画面中便会出现传输设定窗口，通过此窗口便可将表格中的数值下载至主机中，或是由主机中将数据读取回来，但操作前请确认目前的 ISPSOft 可与主机正常联机，详细说明请参考第 2.4 节中的相关内容。

首先在窗口中选择要执行的操作是要由 PLC 中读取数据并填入编辑区中，或是将目前编辑的值写入 PLC；而针对 D 装置的部分，用户还可于此处设定传输的范围，未在勾选范围内的数值则会保持原状；当设定完成后按下「确认」按钮即可开始传输，且传输的内容将不分装置类别，包括 T、C 以及设定范围内的 D 装置都会一并被传输。

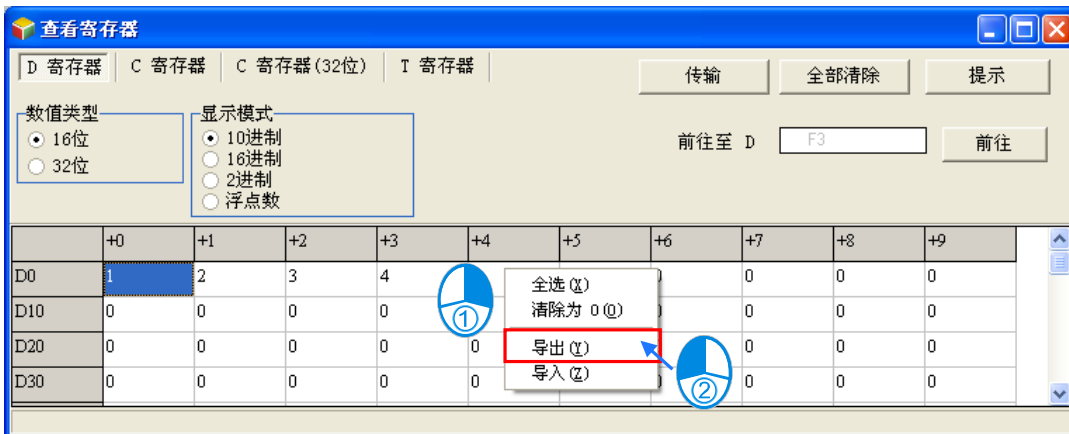


⚠ 将数值写入寄存器前，请务必确认所做的操作不会影响系统运作，或造成系统与人员的伤害。

此处所编辑的内容亦可与项目一起保存，当 ISPSOft 目前的编辑窗口为寄存器编辑工具时，于菜单工具栏中单击 **文件 (F) > 保存 (S)**，之后便可将此处的编辑内容保存为 **项目名称.dvl** 的文件，并会与 ISPSOft 的项目档 (*.isp) 存放于同一路径；当下次于该项目中开启此窗口时，系统便会自动于项目路径下开启此文件，但若项目路径下没有预定的文件时，系统则会将表格的所有装置值默认为 0。用户亦可于菜单工具栏的 **文件 (F)** 项目下单击 **另存为 (A)** 将其另外存放于自定义的路径中，而利用 **文件 (F)** 项目下的 **打开 (O)** 则可将之前保存的文件再读取回来。当然，所有的文件操作也都可以通过图标工具栏中对应的图标来进行操作。

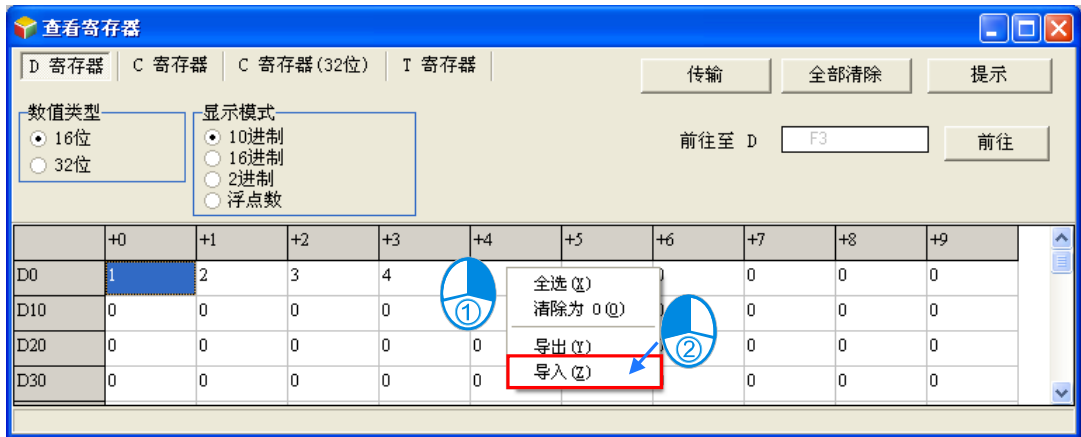


除了可将编辑的内容以文件的型式保存之外，用户亦可将其导出为标准的 CSV 档，而之后便可直接于 EXCEL 当中进行编辑；操作时请于编辑页面中单击鼠标右键，接着于快捷选单中单击「导出」项目，在出现的设定范围窗口选择导出的范围，之后再指定欲导出的路径与文件名即可，而导出的内容将仅包含当时所选取的装置类别页面。

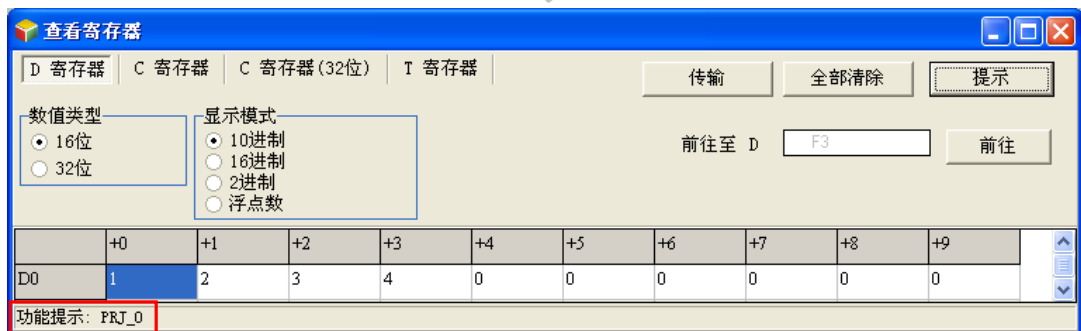
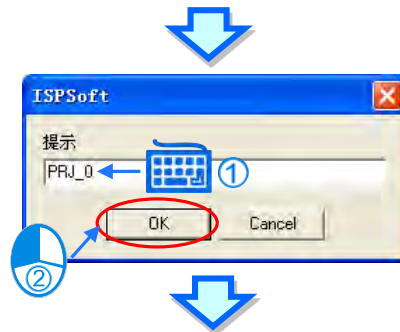
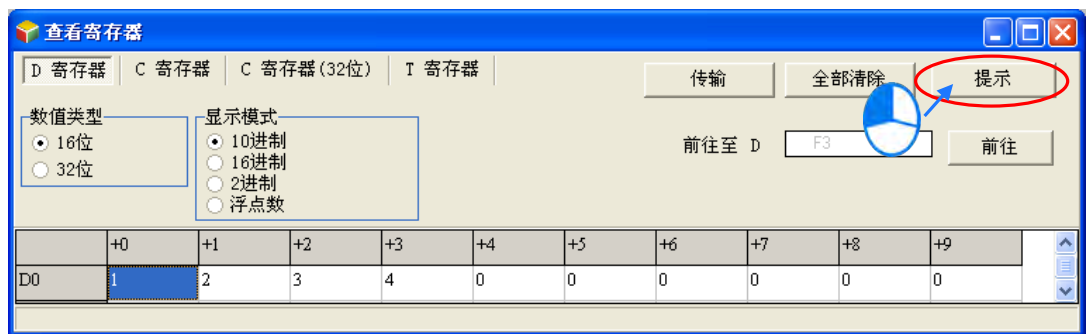


当要重新导入文件时，请于编辑页面中单击鼠标右键并于选单中单击「导入」，之后再指定欲导入的文件即可，不过导入的文件内容必须与目前页面的装置类别一致，不同装置类别的内容将无法导入。


17

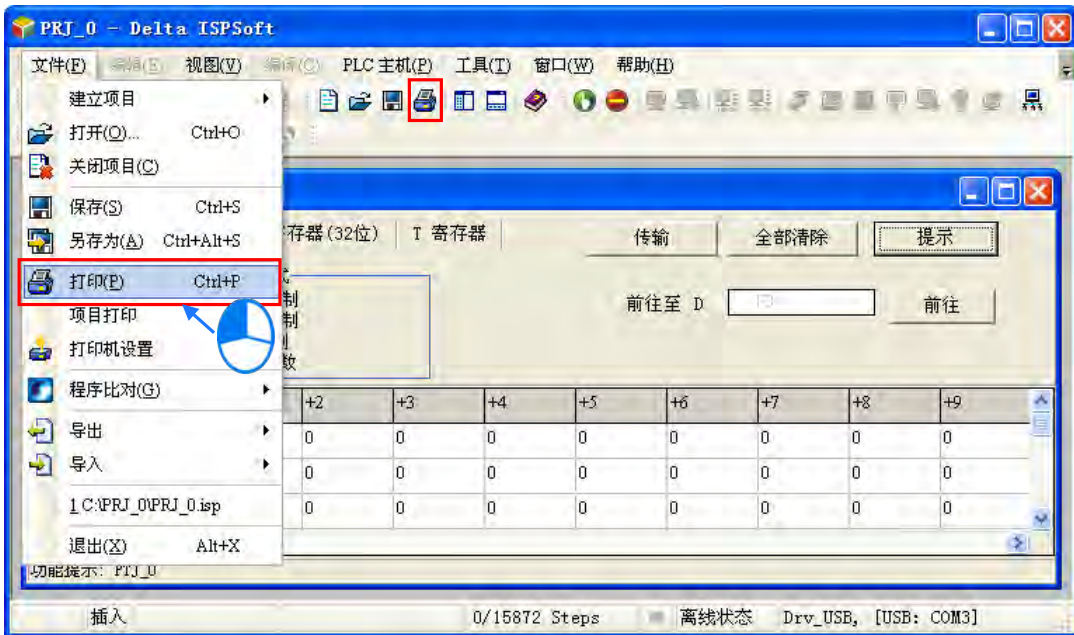


按下窗口中的「提示」按钮后，画面便会出现提示内容的输入窗口，而完成输入后，工具窗口的下方便会出现方才所设定的提示内容，且提示的内容将会随着存档时一并被保存，当之后用户开启文件时，由提示的内容便可清楚的辨别所开启的表格，藉此即可便于用户进行配方管理。



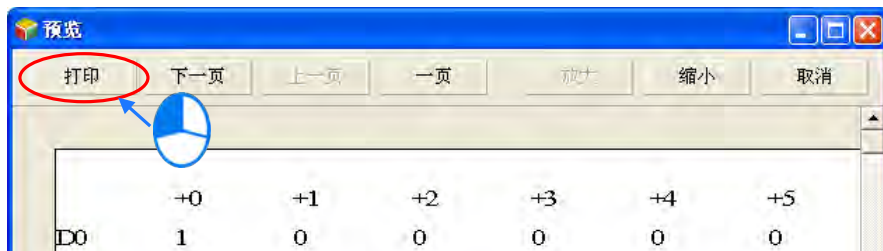
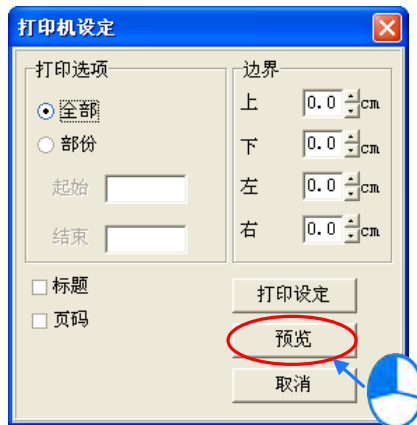
在开启某个装置类别的编辑页面时，在 ISPSOft 的菜单工具栏单击 文件 (F) > 打印 (P)，或者直

接单击图标工具栏中的  图标，之后画面便会出现打印管理窗口，而用户便可藉此将目前所选的装置类别页面打印出来。



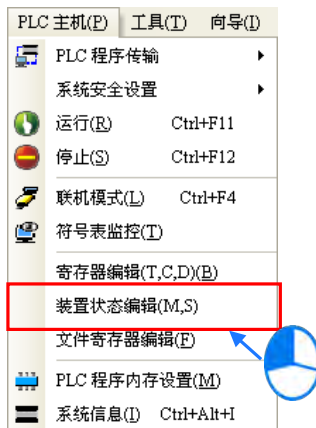
先在「打印选项」中设定欲打印的装置范围后，接着可再设定打印的边界与欲附加于页面中的项目；而按下「打印设定」则可开启打印机的设定窗口，于此可设定使用的列表机与纸张大小等参数；完成后按下「预览」便可开启预览画面，确认无误后按下预览画面的「打印」键即可将其印出。

17



17.2.5 装置状态编辑

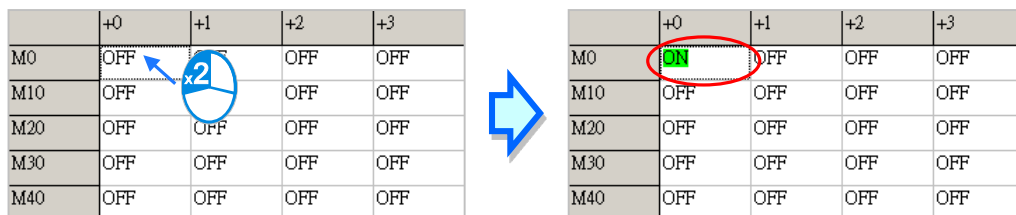
用户亦可通过 ISPSOft 所提供的工具批次编辑 M 装置与 S 装置的内容值，并可将其做为配方而进行保存与下载的操作。开启项目后，于菜单工具栏单击 **PLC 主机 (P)** > **装置状态编辑**，之后画面便会开启编辑窗口。



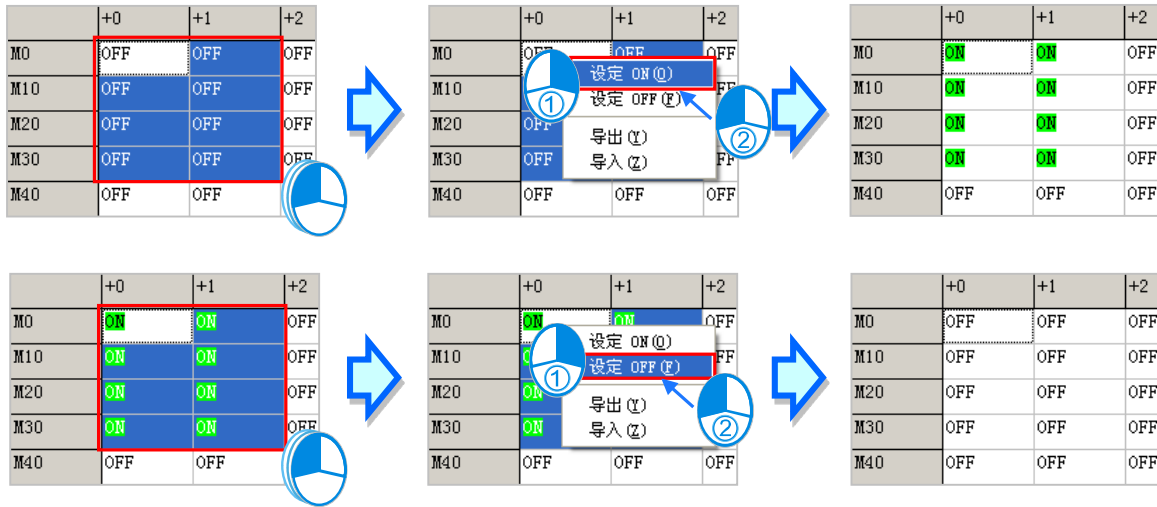
开启窗口后，表格中所带入的装置数值并非由主机内部所上载回来的目前值，而是上次开启此窗口时所保存的编辑内容，但若是在此项目中第一次开启窗口时，所有的装置便会以 OFF 做为默认值。操作时，藉由切换窗口上方的卷标便可选择欲检视的装置类别。



要切换表格中的装置状态时，请直接于要编辑的字段上双击鼠标左键即可。



当要同时设定某段范围的装置状态时，可先使用鼠标框选要设定的范围后，按下鼠标右键并于快捷菜单中单击「设定 ON」或「设定 OFF」，之后框选范围内的装置便会全部被设为所选择的状态。

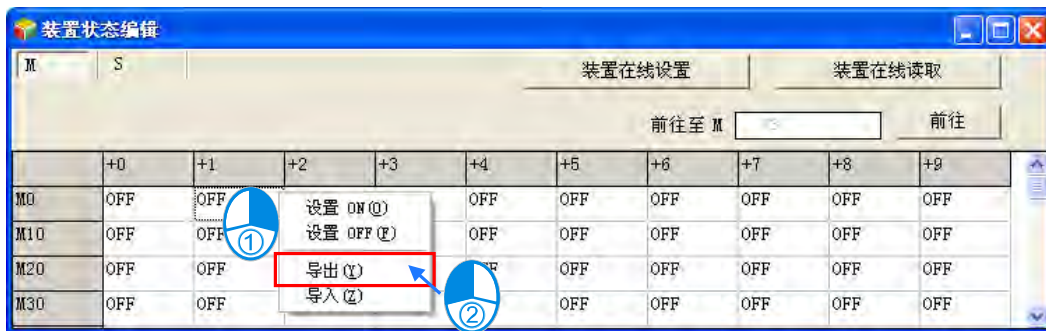


当按下窗口中的「装置在线设置」按钮后，系统便会将此处所做的编辑写入主机当中，而当按下窗口中的「装置在线读取」按钮时，系统则会将主机当中的状态读取回来，且传输时将不分装置类别，包括 M 装置与 S 装置的内容将会一并传输；不过操作前请确认目前的 ISPSOft 已与主机正常联机，详细说明请参考第 2.4 节中的相关内容。

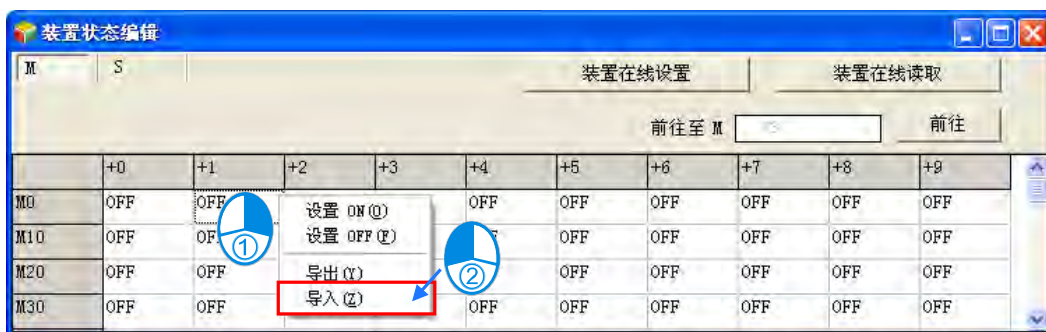


! 将数值写入主机前，请务必确认所做的操作不会影响系统运作，或造成系统与人员的伤害。

针对此处所编辑的内容，用户可将其导出为标准的 CSV 档，而之后便可直接于 EXCEL 当中进行编辑；操作时请于编辑页面中单击鼠标右键，接着于快捷选中单击「导出」项目，在出现的设定范围窗口选择导出的范围，之后再指定要导出的路径与文件名即可，而导出的内容将仅包含当时所选取的装置类别页面。

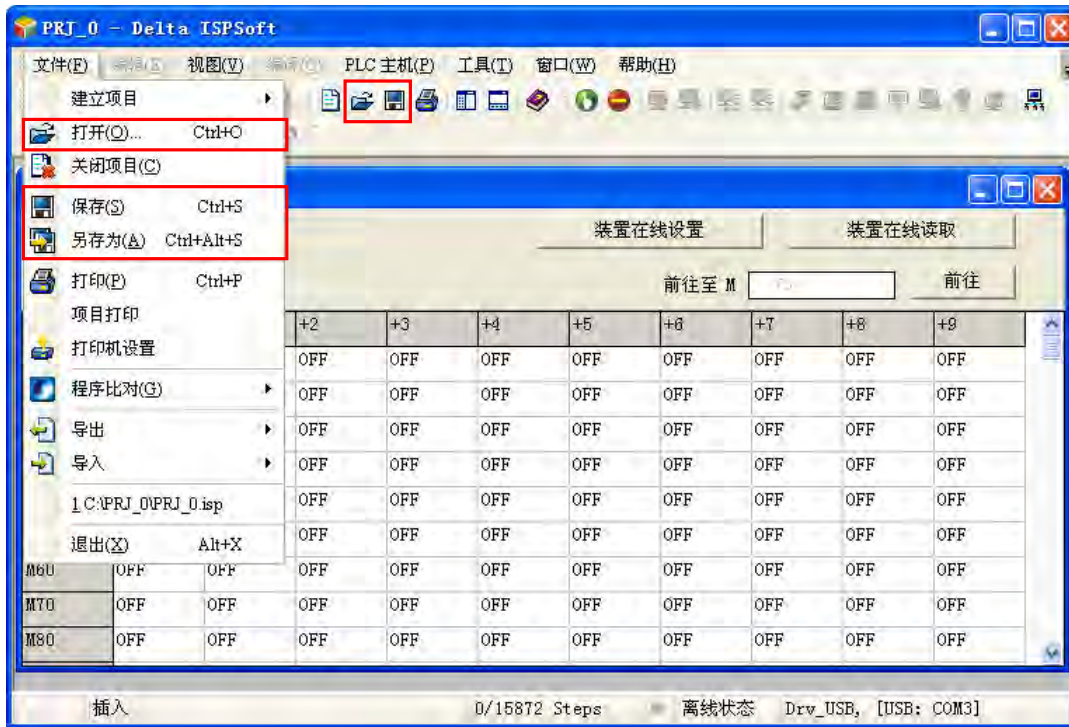



当要重新导入文件时，请于编辑页面中单击鼠标右键并于选单中单击「导入」，之后再指定欲导入的文件即可，不过导入的文件内容必须与目前页面的装置类别一致，不同装置类别的内容将无法导入。

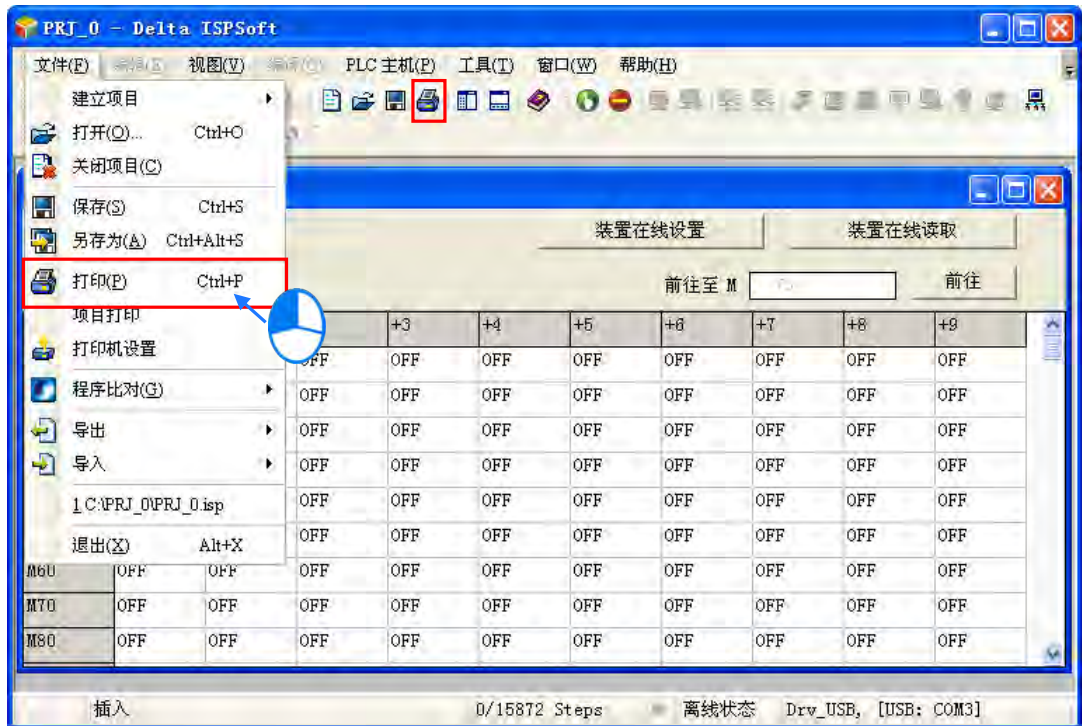


此处的编辑内容亦可与项目一起保存，当 ISPSOft 目前的编辑窗口为装置状态编辑工具时，于菜单工具栏中单击 **文件(F)** > **保存(S)**，之后便可将此处内容保存为 **项目名称.dvb** 的文件，并会与 ISPSOft 的项目档 (*.isp) 存放于同一路径；当下次于该项目中开启此窗口时，系统便会自动于项目路径下开启此文件，但若项目路径下没有预定的文件时，系统则会将所有的装置值默认为 OFF。

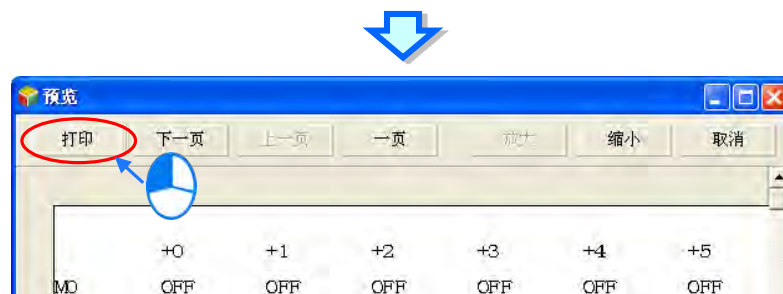
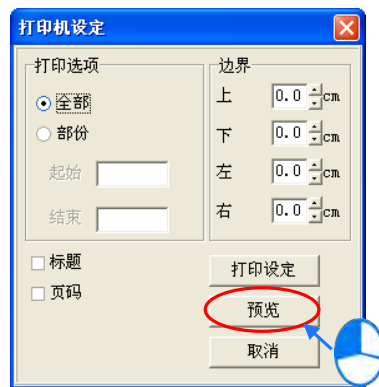
用户亦可于菜单工具栏的 **文件(F)** 项目下单击 **另存为(A)** 将其另外存放于自定义的路径中，而利用 **文件(F)** 项目下的 **打开(O)** 则可将之前保存的文件再读取回来。当然，所有的文件操作也都可以通过图标工具栏中对应的图标来进行操作。



在开启某个装置类别的编辑页面时，在 ISPSOft 的菜单工具栏单击 **文件 (F) > 打印 (P)**，或者直接单击图标工具栏中的  图标，之后画面便会出现打印管理窗口，而用户便可藉此将目前所选的装置类别页面打印出来。



先在「打印选项」中设定要打印的装置范围后，接着可再设定打印的**边界**与要附加于页面中的项目；而按下「打印设定」则可开启打印机的设定窗口，于此可设定使用的列表机与纸张大小等参数；完成后按下「预览」便可开启预览画面，确认无误后按下预览画面的「打印」键即可将其印出。



17.2.6 DVP 机种的文件寄存器编辑

针对 DVP 系列机种，用户亦可通过 ISPSOft 所提供的工具来批次编辑文件寄存器的内容值，并可将其做为配方而进行保存与下载的操作。

开启 DVP 系列机种的项目后，于菜单工具栏单击 **PLC 主机 (P)** > **文件寄存器编辑 (F)**，之后画面便会开启编辑窗口。



开启窗口后，表格中所带入的寄存器数值并非由主机内部所上载回来的目前值，而是上次开启此窗口时所保存的编辑内容，但若是在此项目中第一次开启窗口时，所有寄存器内容便会以 0 做为默认值。用户单击卷标切换不同的文件寄存器范围。



要编辑寄存器内容时，请先单击要编辑的寄存器字段后，直接以键盘输入数值即可。

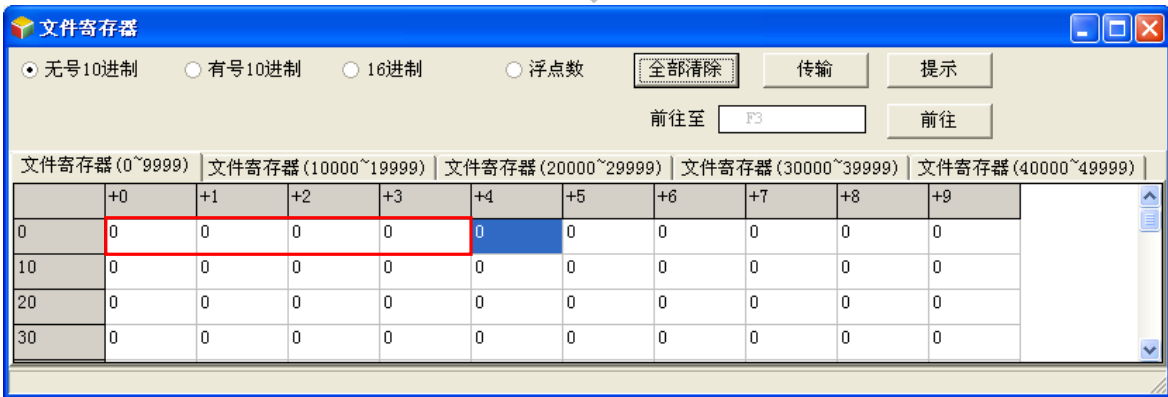
	+0	+1	+2
0	0	0	0
10	0	0	0
20	0	0	0
30	0	0	0
40	0	0	0
50	0	0	0

	+0	+1	+2
0	1234	0	0
10	0	0	0
20	0	0	0
30	0	0	0
40	0	0	0
50	0	0	0

用户可于窗口上方切换数值的显示格式。



当按下窗口中的「全部清除」按钮时，此页面中的所有寄存器内容便会被清除为 0。



除此之外，亦可于页面中按下鼠标右键，接着在快捷选单中单击「全选」，之后再次按下鼠标右键，并于选单中单击「清除为 0」便可将该页面中的所有的寄存器内容清除。

17

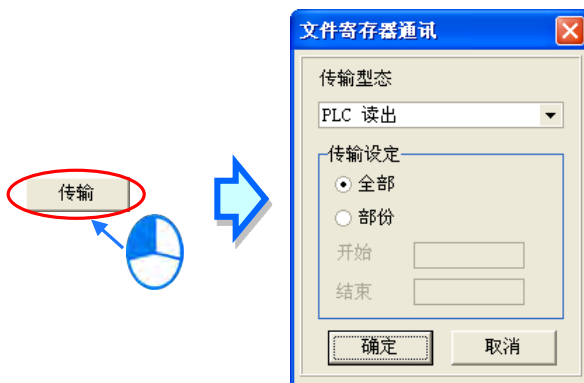


只想清除部份寄存器时，可先使用鼠标框选要清除的范围后，按下鼠标右键并于快捷选单中单击「清除为 0」，之后框选范围内的寄存器数值便会被清除。



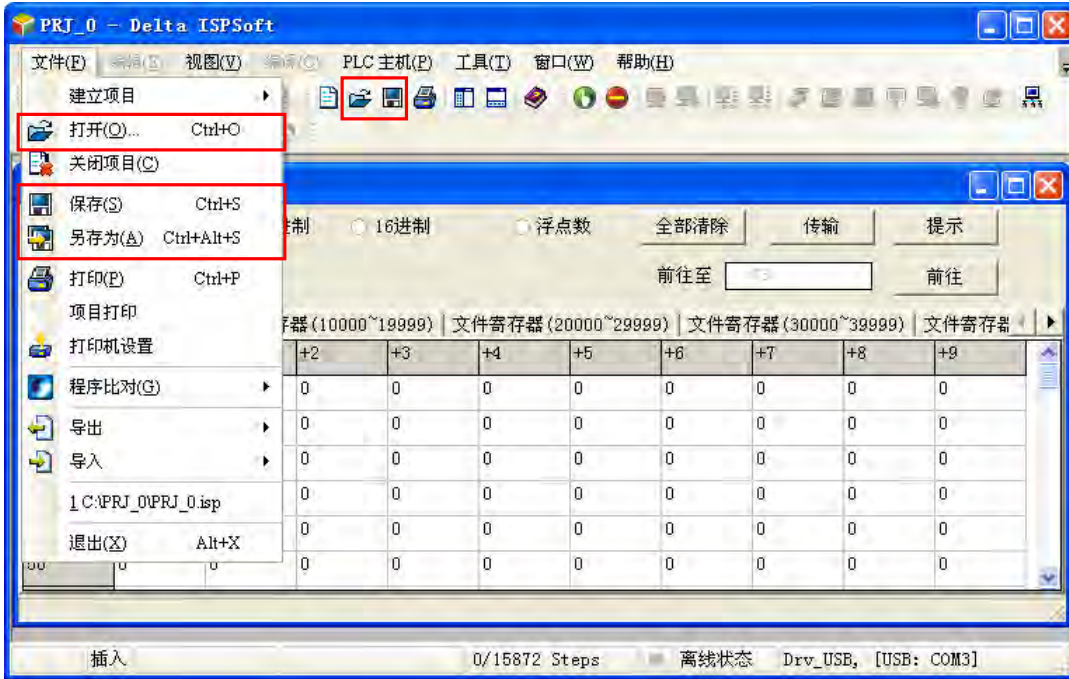
当按下窗口中的「传输」按钮后，画面中便会出现传输设定窗口，通过此窗口便可将表格中的数值下载至主机中，或是由主机中将数据读取回来，但操作前请确认目前的 ISPSOft 可与主机正常联机，详细说明请参考第 2.4 节中的相关内容。

首先于窗口中的「传输类型」字段下拉选择要执行的操作是要由 PLC 中读取数据并填入编辑区中，或是将目前编辑的值写入 PLC，之后再「传输设定」字段中设定传输的范围，完成后按下「确定」按钮便可开始传输数据。

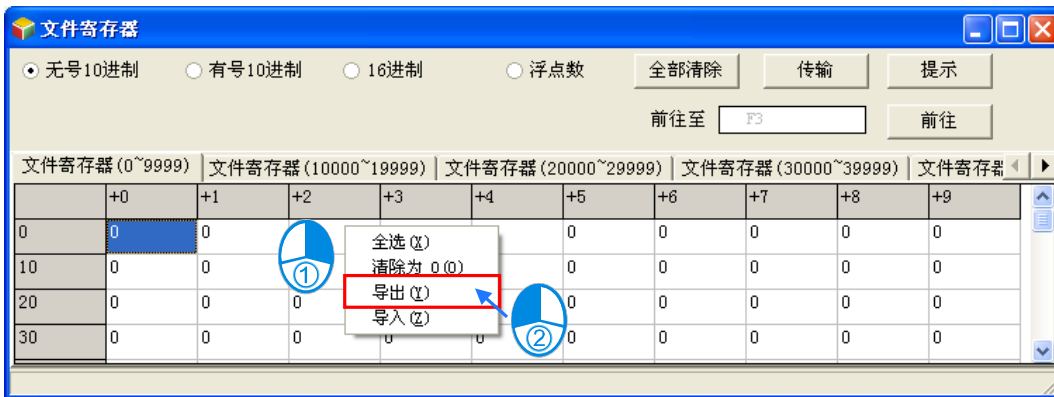


⚠ 将数值写入寄存器前，请务必确认所做的操作不会影响系统运作，或造成系统与人员的伤害。

此处所编辑的内容可与项目一起保存，当 ISPSOft 目前的编辑窗口为寄存器编辑工具时，于菜单工具栏中单击 **文件 (F) > 保存 (S)**，之后便可将此处的编辑内容保存为 **项目名称.wft** 的文件，并会与 ISPSOft 的项目档 (*.isp) 存放于同一路径；当下次于该项目中开启此窗口时，系统便会自动于项目路径下开启此文件，但若项目路径下没有预定的文件时，系统则会将所有的寄存器内容值默认为 0。用户亦可于菜单工具栏的 **文件 (F)** 项目下单击 **另存为 (A)** 将其另外存放于自定义的路径中，而利用 **文件 (F)** 项目下的 **打开 (O)** 则可将之前保存的文件再读取回来。当然，所有的文件操作也都可以通过图标工具栏中对应的图标来进行操作。

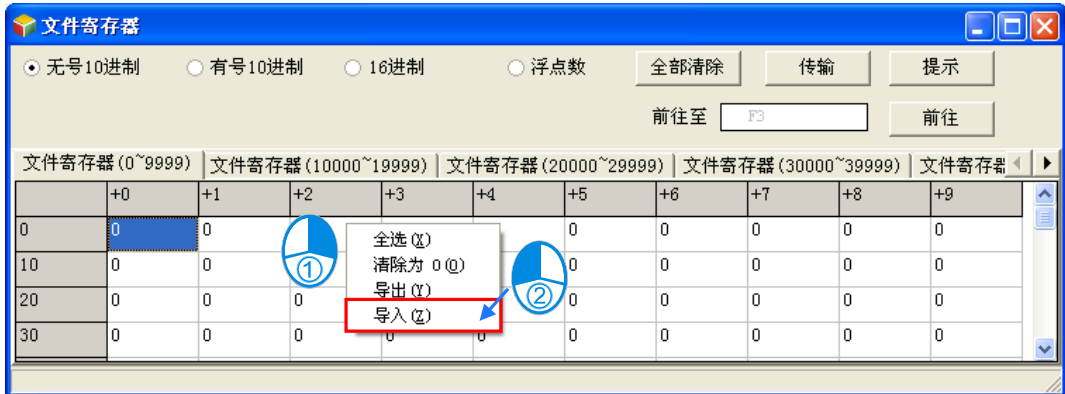


除了可将编辑的内容以文件的型式保存之外，用户亦可将其导出为标准的 CSV 档，而之后便可直接于 EXCEL 当中进行编辑；操作时请于编辑页面中单击鼠标右键，接着于快捷选单中单击「导出」项目，之后再指定要导出的路径与文件名即可。

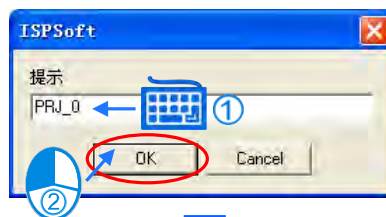



17

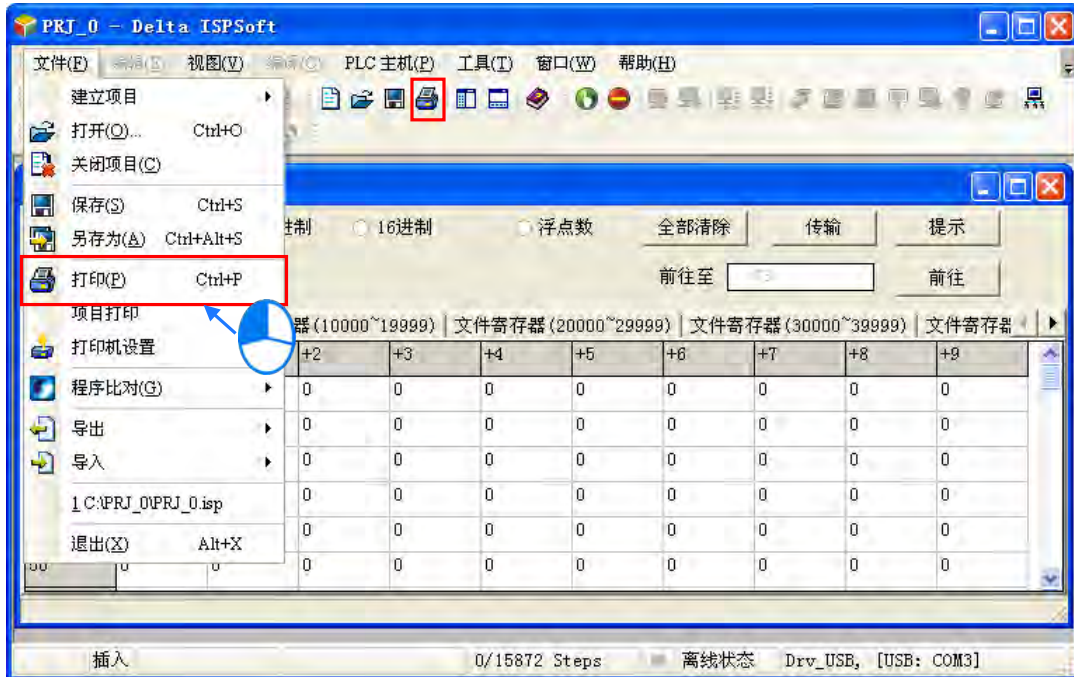
当要重新导入文件时，请于编辑页面中单击鼠标右键，接着再于快捷选单中单击「导入」，之后再指定要导入的文件即可。



按下窗口中的「提示」按钮后，画面便会出现提示内容的输入窗口，而完成输入后，工具窗口的下方会出现方才所设定的提示内容，且提示的内容将会随着存档时一并被保存，当之后用户开启文件时，由提示的内容便可清楚的辨别所开启的表格，即可便于用户进行配方管理。

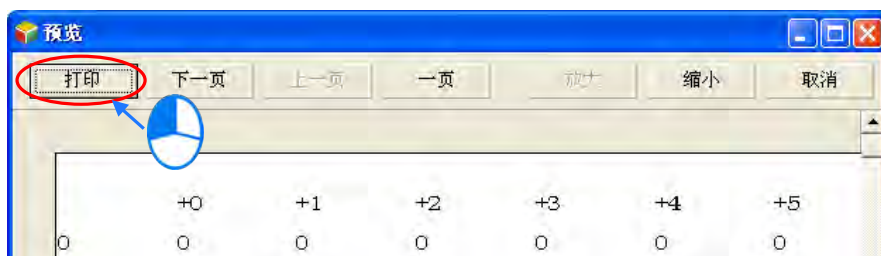
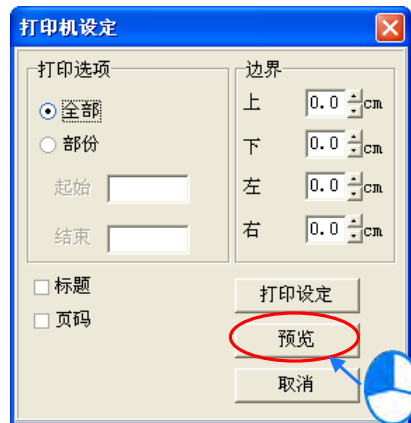


在开启寄存器编辑工具时，于 ISPSOft 的菜单工具栏单击 **文件 (F) > 打印 (P)**，或者直接单击图标工具栏中的  图标，之后画面便会出现打印管理窗口，藉此便可将目前的页面打印出来。



先于「打印选项」中设定要打印的装置范围后，接着可再设定打印的**边界**与要附加于页面中的项目；而按下「打印设定」则可开启打印机的设定窗口，于此可设定使用的列表机与纸张大小等参数；完成后按下「预览」便可开启预览画面，确认无误后按下预览画面的「打印」键即可印出。

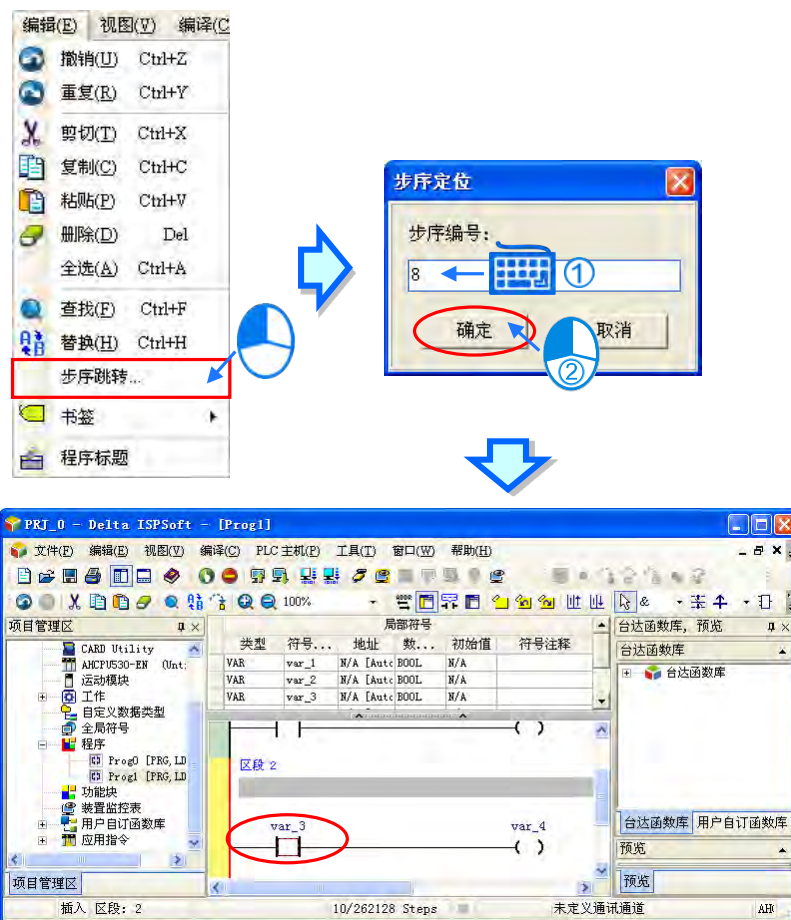
17



17.3 步序定位

17.3.1 使用步序跳转

当主机运行中发生错误时，可藉由特殊寄存器（例如 AH 系列机种的 SR1 与 SR2 的 32 位演算错误地址锁定寄存器）得知错误的步（step）数，回到项目程序中查找此步数所对应的程序，藉以了解发生错误的原因。于菜单工具栏单击 **编辑 (E) > 步序跳转...** 开启步序定位窗口，于步序编号字段输入要查询的步数后，按下确定按钮，ISPSOft 便会自动开启对应此步数的 POU 并定位到程序中。如下图。使用步序定位时，项目需先行编译。



17.4 程序比对

17.4.1 程序比对简介

ISPSOft 提供了一个程序比对工具，供用户比对计算机中的两个 ISPSOft 项目，或是 ISPSOft 项目与联机的 PLC 主机内的程序。比对的规则如下。

- 比对的目标包括机种、型号、工作、全局符号、POU、区域符号、程序。
- POU 会先比对 POU 名称，不同的 POU 名称即便内容相同也将判别为不同。
- 变量符号表即便两者有相同的内容，变量符号的排列顺序不同也将判别为不同。如下图，右侧的符号表 var_a 与 var_b 与左侧符号表的顺序不同，故前两行判别为不同。

类型	符号名称	地址	数据类型	初始值
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A

类型	符号名称	地址	数据类型	初始值
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A

- 变量符号表会比对出程序的跳行，如下图。比对的结果将只判别为左侧符号表第 3 行不存在于右侧的符号表，其他相同。

类型	符号名称	地址	数据类型	初始值
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A
VAR	var_e	N/A [Auto]	BOOL	N/A

类型	符号名称	地址	数据类型	初始值
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A
VAR	var_e	N/A [Auto]	BOOL	N/A

- LD、FBD、IL、ST 语言 POU 会比对出程序的跳行，如下图。比对的结果将只判别为左侧程序 POU 第 2 行不存在于右侧的程序 POU，其他相同。

```

0001 ld var1
0002 and var3
0003 out var2
0004 ld m0
0005 out var3

```

```

0001 ld var1
0002 out var2
0003 ld m0
0004 out var3

```

- SFC 语言 POU 的比对只要 SFC 有差异即判别为整个 POU 不同；但 SFC 中的 Action 或 Transition 不同时也会办别出 Action 或 Transition 中不同的区段或行。
- 项目需先存盘后，才可使用程序比对功能。
- 主机在下载时必须勾选**程序**项目，之后才可比对此 PLC 主机的内容。

17.4.2 与文件比对

用于比对计算机中的不同项目。请单击 **文件(F)** > **程序比对(G)** > **与项目比对(F)**，在出现的开启项目窗口中选择要比对的目标项目，按下**打开**按钮。

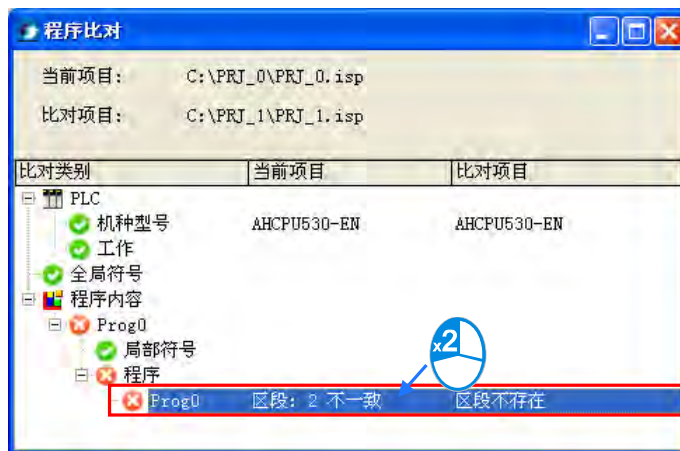


之后便会出现程序比对窗口，列出两者的异同。说明如下。



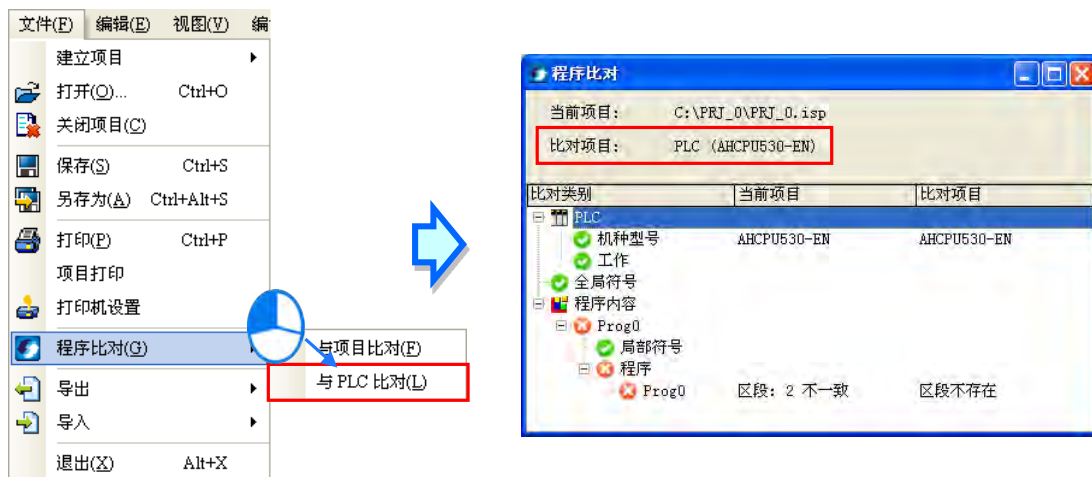
- ① 目前开启的项目位置。
- ② 比对的目标项目位置。
- ③ 比对的结果，列出机种型号、工作、全局符号、程序内容的异同。

在不一致的项目上双击鼠标左键，将自动开启此项目的内容，如下图，开启 Prog0 定位并在区段 2。

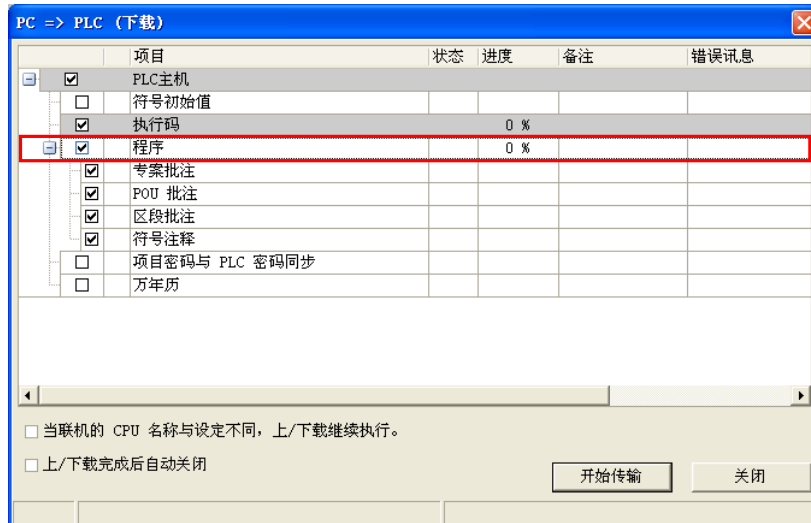


17.4.3 与 PLC 比对

用于比对主机内容与计算机中的项目。请先确认 ISPSOft 与主机已建立联机，再单击 **文件(F) > 程序比对(G) > 与 PLC 比对(L)**，ISPSOft 将读取目前联机主机的项目，开启程序比对窗口，显示比对项目内容。如下图，可以看到比对项目为目前联机的 PLC 主机机种。



比对目标的 PLC 主机在下载时必须勾选**程序**项目，若与未下载程序的主机进行比对，则会出现信息告知用户无法比对。下载功能请参考第 17.1.3 章。



MEMO

17

第18章 测试与除错工具



目录

18.1	在线监控.....	18-2
18.1.1	切换主机的运转/停止状态	18-2
18.1.2	在线监控的功能与环境介绍.....	18-2
18.1.3	在线模式下变更 X 装置的状态	18-8
18.1.4	程序监控的在线操作	18-9
18.1.5	装置监控表	18-15
18.1.6	在线编辑与在线更新	18-21
18.2	DVP 机种的除错模式.....	18-24
18.2.1	启动 DVP 的除错模式	18-24
18.2.2	设定与清除断点	18-25
18.2.3	除错模式下的程序执行	18-25
18.3	AH/AS 机种的除错模式	18-27
18.3.1	进入除错模式	18-27
18.3.2	断点操作	18-27
18.3.3	连续执行.....	18-28
18.3.4	单步执行.....	18-29
18.4	确认 PLC 状态	18-30
18.4.1	PLC 系统信息	18-30
18.4.2	PLC 系统记录	18-32
18.5	3D 坐标图	18-39
18.5.1	功能说明.....	18-39
18.5.2	建立 3D 坐标图.....	18-39
18.5.3	加入曲线.....	18-41
18.5.4	显示 3D 坐标图.....	18-43

18.1 在线监控

18.1.1 切换主机的运转/停止状态

在进行测试与除错的过程中，须要经常性地改变主机的运行状态，而通过 ISPSOft，我们可轻易的进行切换。在操作前，请确认目前 ISPSOft 已可与主机正常联机，详细说明请参考第 2.4 节。

在菜单工具栏中单击 **PLC 主机 (P)** > **运行 (R)**，或按下图标工具栏的  图标可将主机切换至 **RUN** 的状态；而在菜单工具栏中单击 **PLC 主机 (P)** > **停止 (S)**，或按下图标工具栏的  图标则可重新将主机切换至 **STOP** 状态。





通过 ISPSOft 来切换主机的状态时，并不用考虑主机本体的 **RUN/STOP** 开关位置；且由 ISPSOft 下达 **RUN/STOP** 命令后，若再次切换主机本体的 **RUN/STOP** 开关时还是可以变更主机的运行状态。


18.1.2 在线监控的功能与环境介绍

当 ISPSOft 已可与主机正常联机后，我们便可经由**在线监控**模式来对 PLC 的执行状况进行监控。关于主机与 ISPSOft 之间的联机设定方式请参考第 2.4 节的内容。




在 ISPSOft 中，**在线监控**的模式又可分为「**装置监控**」与「**程序监控**」。

监控模式	说明
 装置监控	可通过监控表来实时监控主机目前的装置状态，且在此模式下，ISPSOft 仅需更新装置状态，因此目前 ISPSOft 所开启的程序与主机内部的程序无须一致。
 程序监控	在此模式下，系统会实时将程序的运作状况显示在程序画面中，也因此系统会要求目前 ISPSOft 所开启的程序必须与主机内部的程序一致。


*.装置监控模式可单独启动，而程序监控模式则必须伴随装置监控模式一起启动。

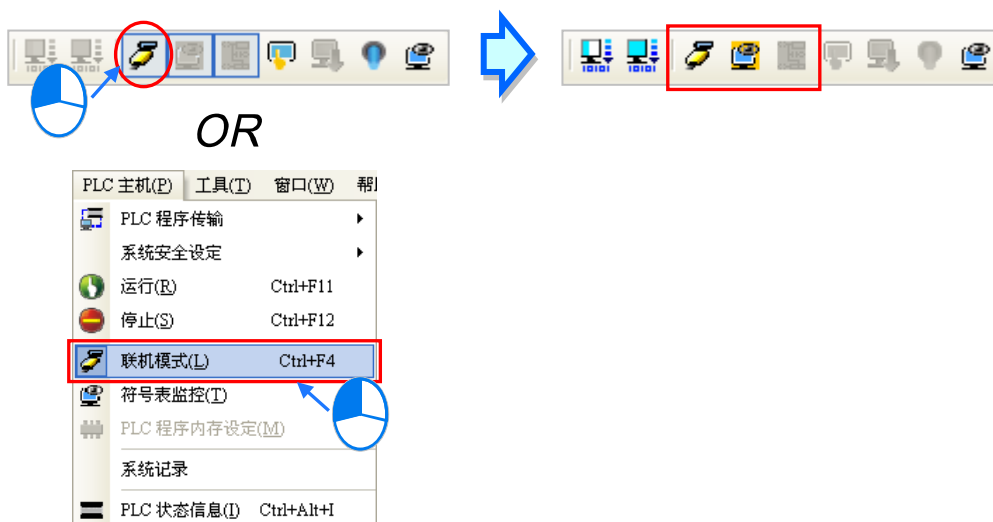
直接在菜单工具栏中单击 **PLC 主机 (P)** > **联机模式 (L)**，或按下图标工具栏的  图标便可进入**在线监控**，而此时系统将会同时启动「**装置监控**」与「**程序监控**」两种模式。



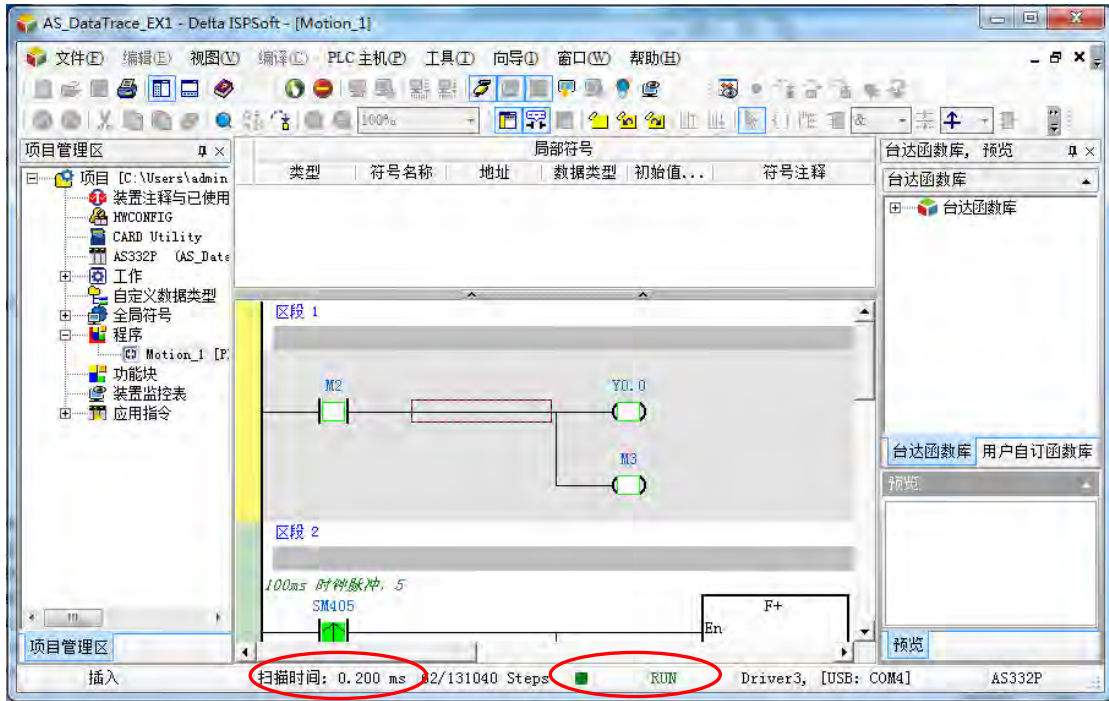
若一开始只要进行**装置监控**时，按下图标工具栏的  图标便可进入**装置监控**模式，但此时**在线监控**的图标  同样会呈现被按下的状态，代表已进入**在线监控**；而后续若要再进行**程序监控**时，只要再按下图标工具栏的  图标便可进入**程序监控**的状态。



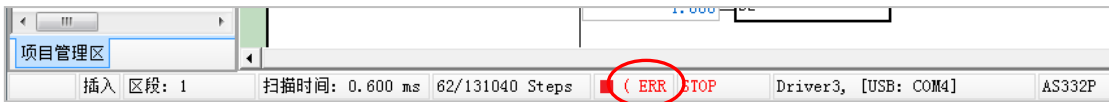
当要结束**在线监控**时，只要再次在菜单工具栏中单击 **PLC 主机 (P)** > **联机模式 (L)**，或按下图标工具栏的  图标并将它恢复为未按下的状态即可。



进入**在线模式**后，在窗口下方的状态栏便可知程序扫描时间与主机目前的状态，而位于主机状态左侧的灯号则会实时显示通讯状况，当持续进行通讯时，该灯号便会维持绿灯的状态。

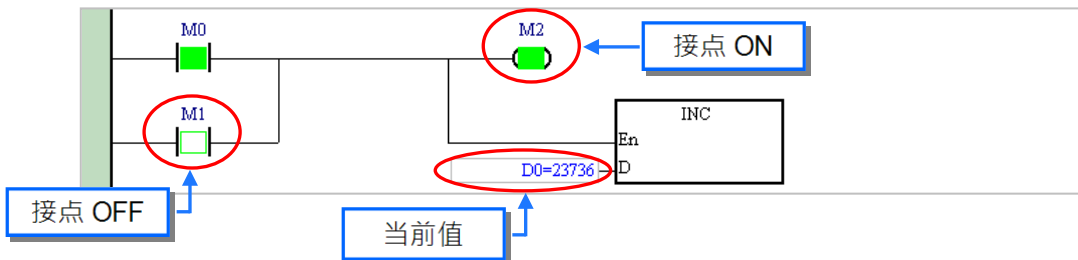


当主机处于错误状态时，部分的操作可能无法进行，例如可能无法将主机切换至**RUN**状态，此时请先参考各机种的使用手册进行错误排除。



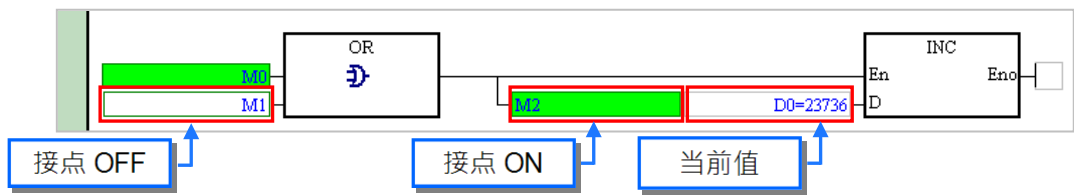
当进入「**程序监控**」的状态后，程序的画面随实时显示各个装置的状态与当前值，藉此可进行相关的除错与测试的工作；不过针对不同的编程语言，其显示的画面还是会有些许的差异。

下图为**梯形图 (LD)** 的程序监控画面。

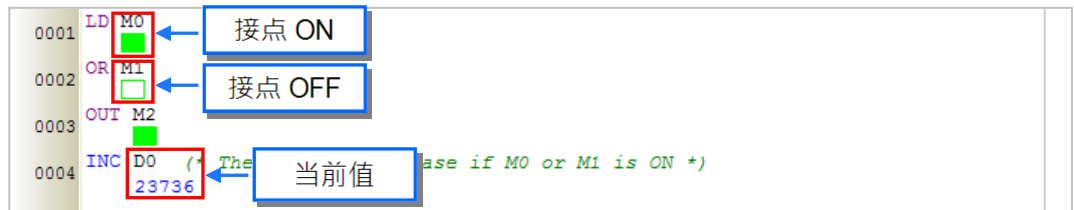


下图为**功能块图 (FBD)** 的程序监控画面。

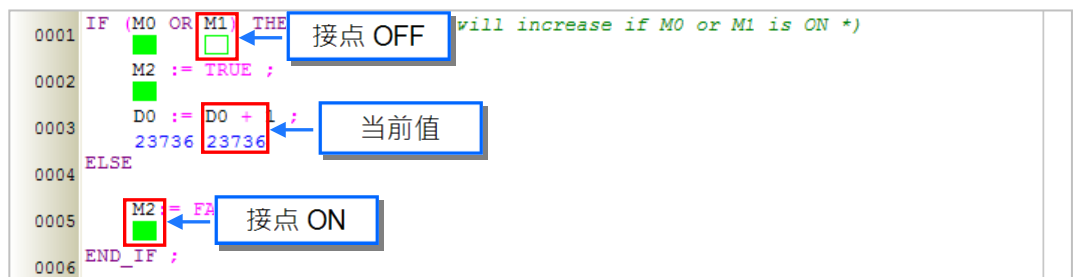
18



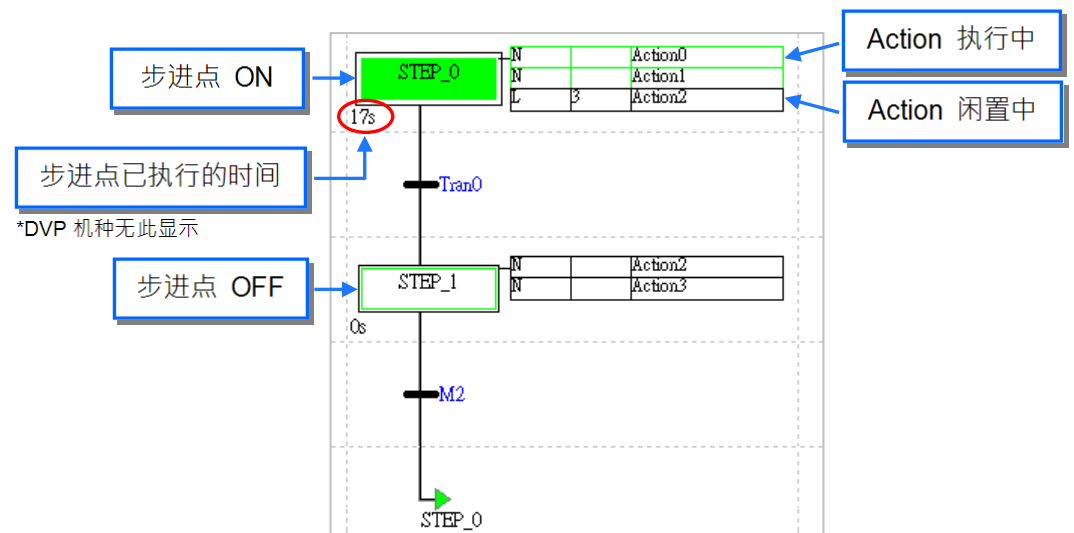
下图为指令列表 (IL) 的程序监控画面。



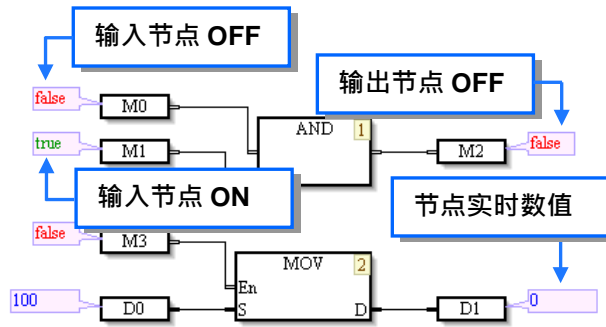
下图为结构化语言 (ST) 的程序监控画面。



下图为顺序功能图 (SFC) 的程序监控画面。除了 SFC 本身的图面可进行监控之外，该 SFC 所属的 Actions 与 Transitions 也都可进行监控。

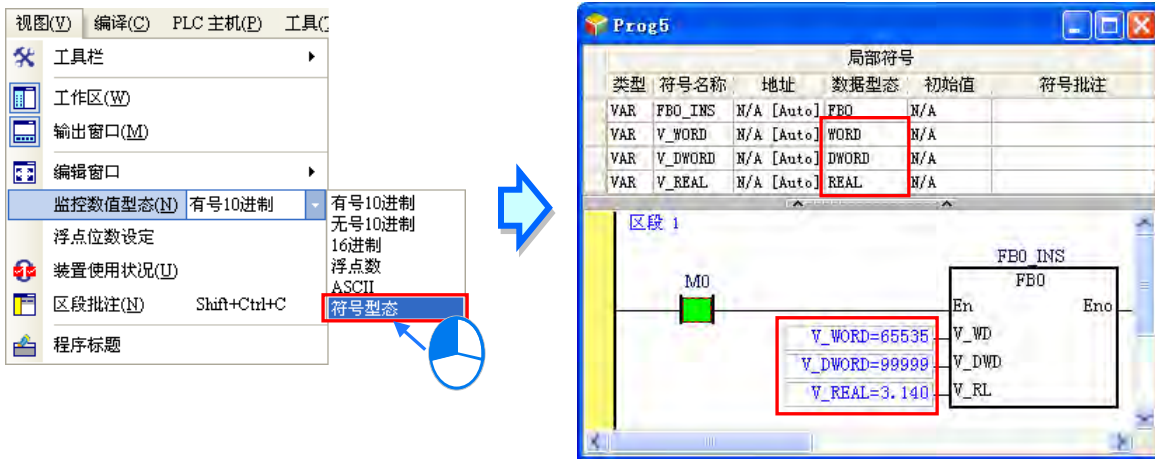


下图为连续功能图 (CFC) 的程序监控画面。



如上述各种语言的程序监控画面，除了可观察接点状态之外，同时还可监控寄存器的实时值，而为了便于视图，我们还可设定监控数值的显示格式。

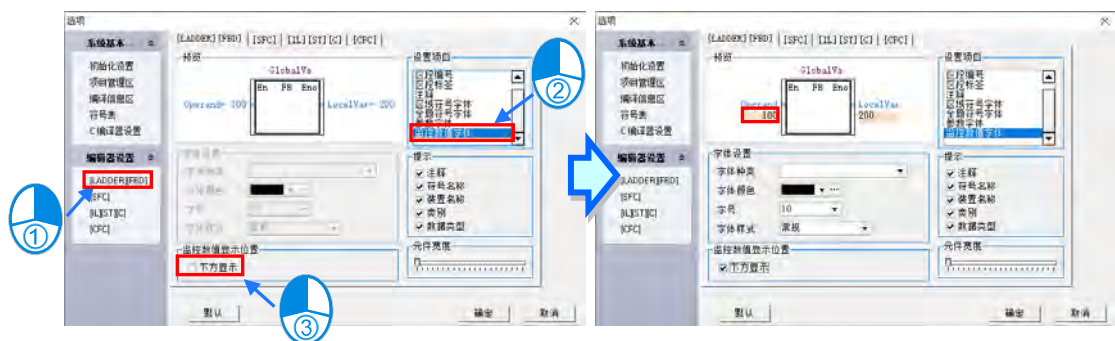
设定时，请在菜单工具栏中单击 **视图 (V) > 监控数值型态 (N)**，之后在此下拉选择数值格式，而当选择「符号型态」时，监控数值便会依据变量符号所宣告的数据类型而自动变更显示格式。



菜单工具栏中单击 **视图 (V) > 浮点位数设定**，之后可设定浮点数的显示格式。




在菜单工具栏中单击 **选项 (O) > [LADDER][FBD] > 监控数值字型**，接着在监控数值显示位置勾选下方显示，就可以将监控的数值显示于引脚下方。

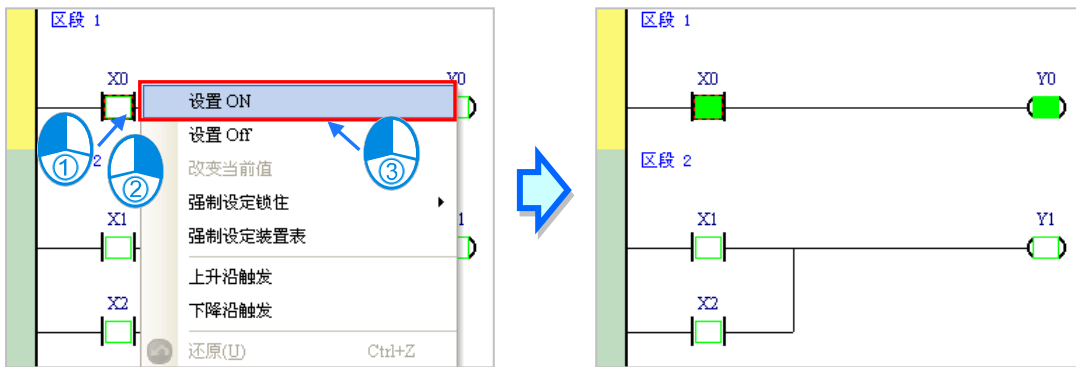


18.1.3 在线模式下变更 X 装置的状态

在「程序监控」的模式下，用户可直接在程序画面上变更接点的状态，不过根据主机的设计，部份的机种可能不允许用户在线模式下直接变更 X 装置的状态，当企图变更 X 装置的状态时，画面将会显示通讯错误的信息。相关细节请参考各机种的使用说明。



若测试上有须要对 X 装置直接进行操作时，请在图标工具栏中单击  图标，之后可开启 X 装置的设定的功能；不过当此功能启动后，外部的实际输入将无法再更新 X 接点的状态。



在离开在线模式后，设定 X 装置的功能并不会自动解除，因此在离开在线模式之前请确认是否须先将其解除，以免造成实际运作时，主机无法依据外部的输入来更新 X 接点的状态。

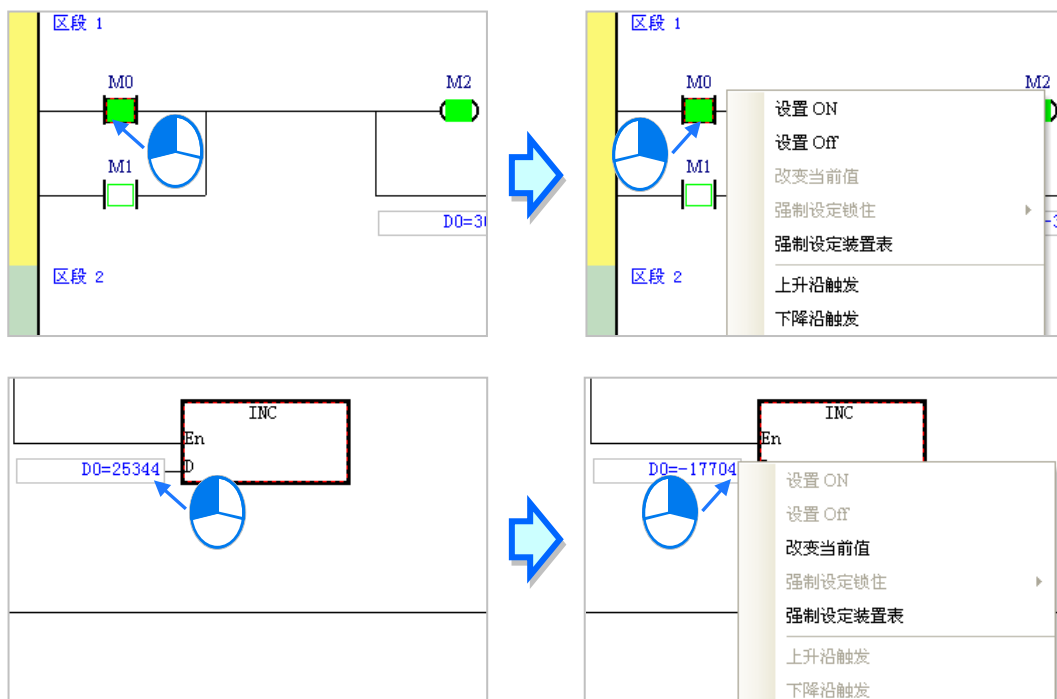


⚠ 将变更接点状态前，请务必确认所做的操作不会影响系统运作，或造成系统与人员的伤害。

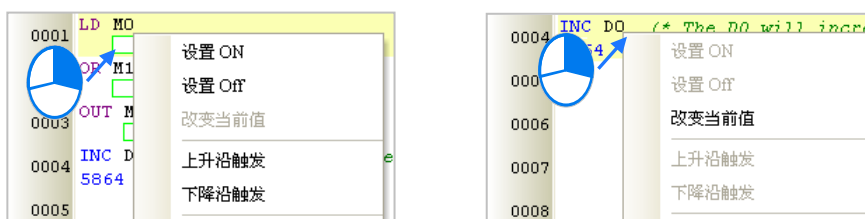
18

18.1.4 程序监控的在线操作

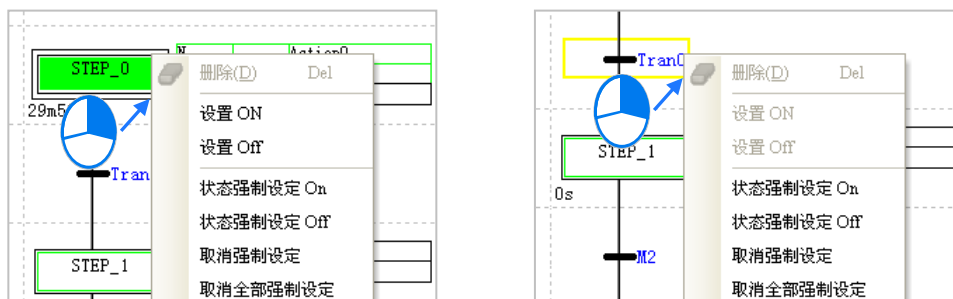
在「程序监控」的模式下，用户可直接在程序画面上变更接点的状态或寄存器的内容值，藉此可进行除错或测试的工作；而操作时，若是在 LD 或 FBD 的程序画面中，请将欲先操作的装置或指令操作数上单击鼠标左键，接着按下鼠标右键，之后便可藉由快捷选单中来进行相关的操作；不过依据编程语言的不同，在该环境下可操作的项目可能会有些许差异。



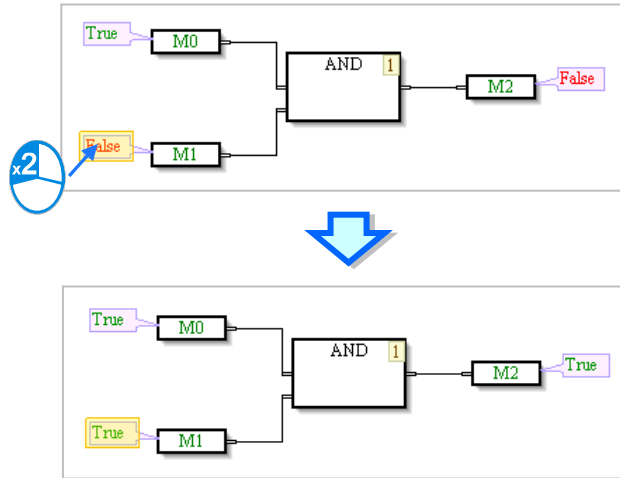
若在 IL 或 ST 的程序画面中，请直接在欲操作的操作数上单击右键，之后便可由快捷选单进行操作。



在 SFC 的图面中，请直接在步进点或转换点上单击鼠标右键，之后可由快捷选单来对其状态进行 ON/OFF 的设定操作。



在 CFC 的图面中，请直接在显示的状态上双击鼠标左键，便会在 ON/OFF 的状态中切换。



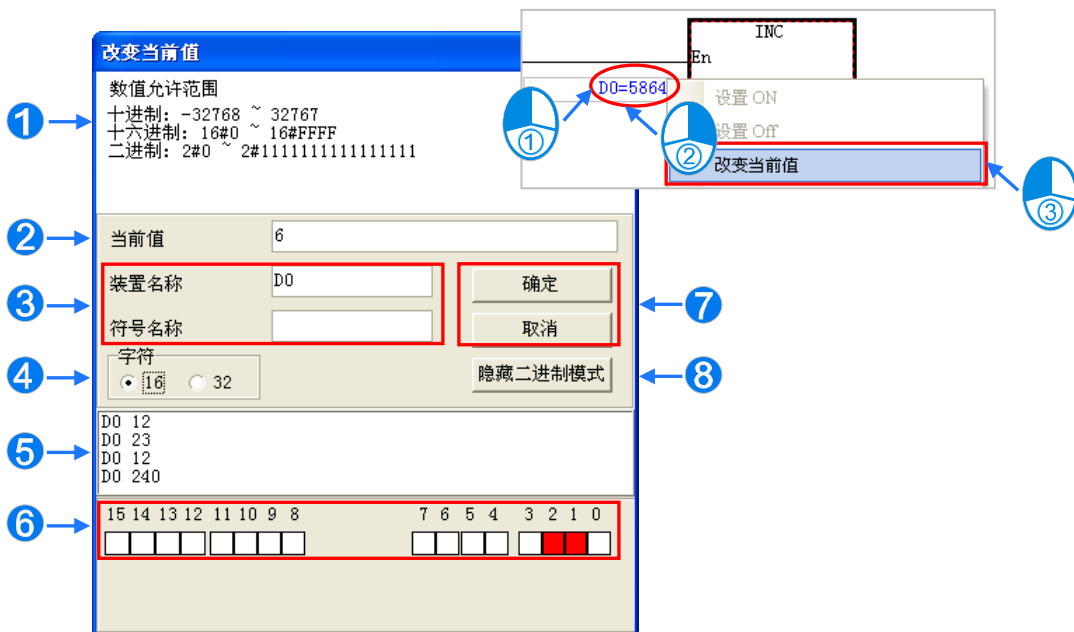
关于各操作项的说明简述，请参考下表。

操作项目	说明
设置 ON	将单击的接点状态设置为 ON。
设置 OFF	将单击的接点状态设置为 OFF。
上升沿触发	不管被单击的接点状态为何，系统皆会将该接点先设为 OFF 之后，再设为 ON。
下降沿触发	不管被单击的接点状态为何，系统皆会将该接点先设为 ON 之后，再设为 OFF。
改变当前值	执行此功能后，用户可在输入窗口中设定单击操作数的内容值。
强制设定锁住	将 X 或 Y 接点强制锁定在 ON 或 OFF 的状态。
强制设定装置表	利用表格的方式，批次对 X 或 Y 接点的强制锁定 ON/OFF 状态进行管理。

⚠ 将变更装置状态前，请务必确认所做的操作不会影响系统运作，或造成系统与人员的伤害。

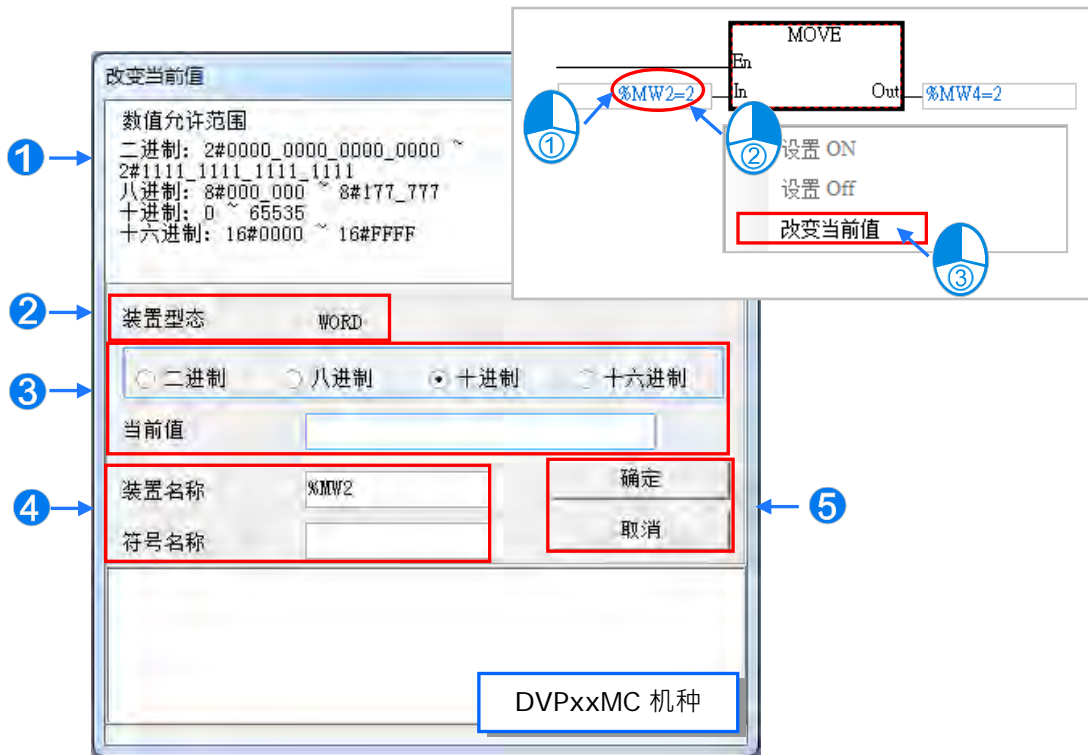
- 改变当前值

单击某一操作数后按下鼠标右键，接着在快捷选单中单击「改变当前值」，之后便可实时由下图的窗口中设定该操作数的内容值。



- ① 提示信息。
- ② 于此选择输入的数值格式为 16 位元或 32 位元或 64 位元 (不同机种，支持选择内容不同)。
- ③ 于此输入欲变更的数值。
- ④ 欲变更现在值的装置名称或变数符号名称。
- ⑤ 数值变更记录。(显示格式：装置名称 变更数值)。
- ⑥ 当开启二进制模式时，用鼠标便可直接设定各个 Bit 的 ON/OFF 状态。
- ⑦ 按下「确定」键可套用设定值，按下「取消」键则可关闭此设定窗口。
- ⑧ 单击此键可开启或隐藏二进制模式。

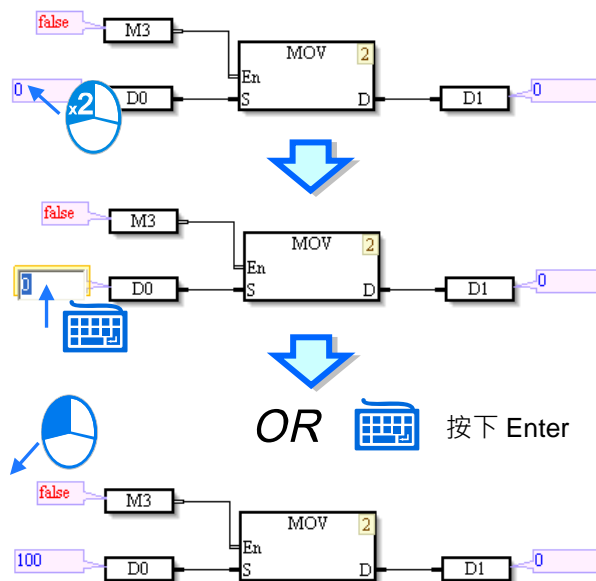
若是用户选择是 DVPxxMC 机种，则会出现以下图示。



- ① 提示信息。
- ② 此处显示目前装置的类型。
- ③ 于此输入欲变更的数值，而输入格式可为二进制、八进制、十进制或十六进制。
- ④ 欲变更现在值的装置名称或变数符号名称。
- ⑤ 按下「确定」键可套用设定值，按下「取消」键则可关闭此设定窗口。

关于 CFC 语言，改变现在值只要在显示目前数值的位置上双击鼠标左键，数值便会呈现编辑状态，输入欲改变的数值后，以鼠标点击空白处或按下键盘的 Enter 键，便完成数值的输入。

18



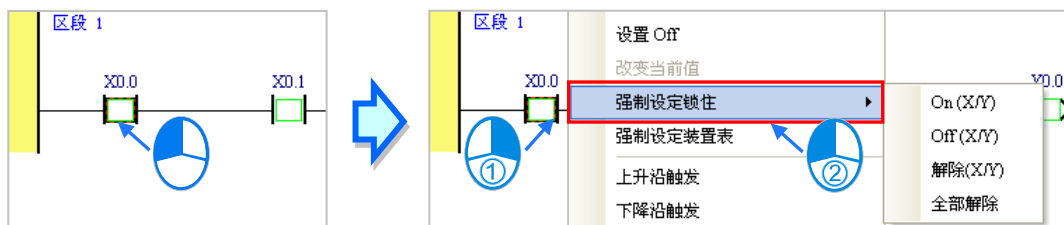
● 强制设定锁住

强制锁定 ON/OFF 的功能仅可使用在 X 或 Y 接点，其作用在于将 X 或 Y 接点锁定在某个状态上；当某个 X 接点被锁定后，实际外部输入的状况将无法改变该接点的状态；而同样的，当某个 Y 接点被锁定后，无论程序内部的运行结果为何，最后 Y 接点的实际输出仍会依据该接点被锁定的状态。

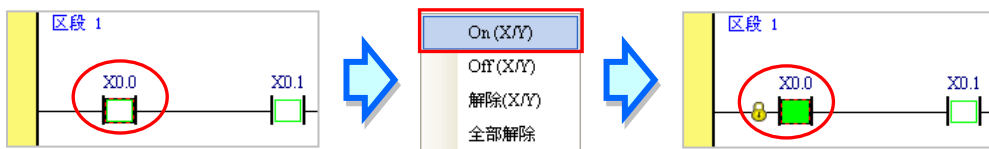
如下图，因接点 Y0.0 被强制锁定为 ON，因此即使 M0 为 OFF，Y0.0 还是会固定在 ON 的状态，且处于强制锁定 ON/OFF 状态的接点，在该接点图标旁边会出现一个大锁的符号。



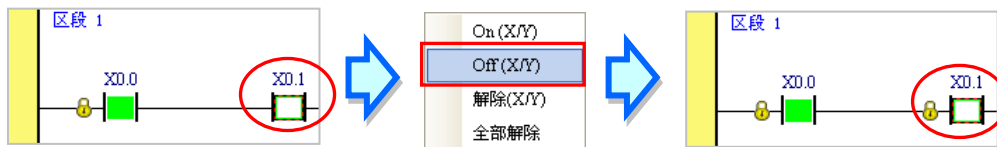
设定时，请先在欲操作的 X 或 Y 接点上单击鼠标左键，接着按下鼠标右键并在快捷选中单击「强制设定锁住」，之后便可透过其下阶的项目来进行相关的操作。



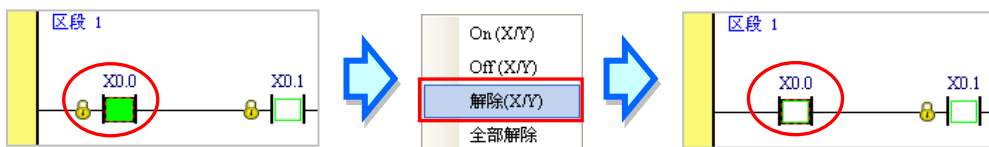
➤ **On (X/Y)**: 将单击的 X 或 Y 接点锁定在 ON 的状态。



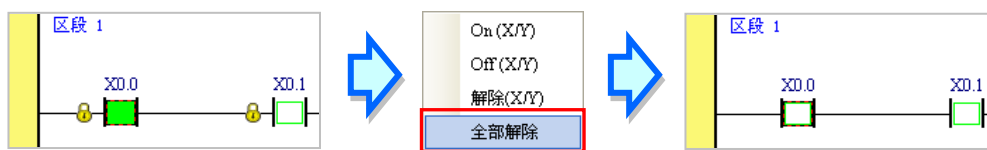
➤ **Off (X/Y)**: 将单击的 X 或 Y 接点锁定在 OFF 的状态。



➤ **解除 (X/Y)**: 解除单击接点的锁定状态。



➤ **全部解除**: 解除所有接点的锁定状态。

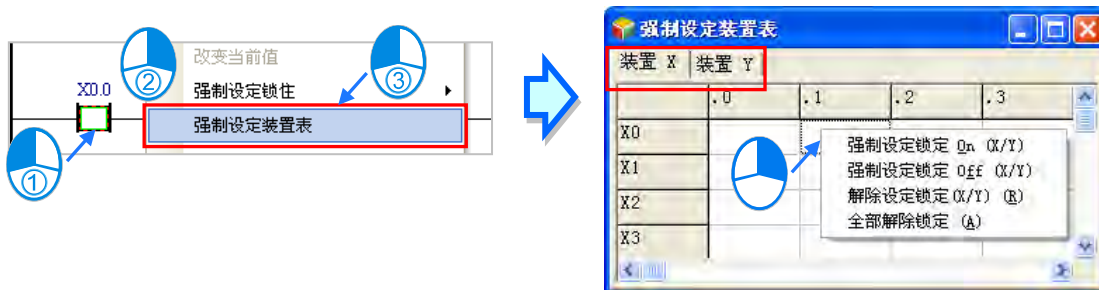


- *1.不同机种对于强制锁定 ON/OFF 的处理机制及点数限制可能不同，设定前请先参考各机种的相关使用说明。
- *2.在离开在线模式后，各接点的强制锁定状态将不会自动解除，请务必确认目前的状态是否会造成影响。

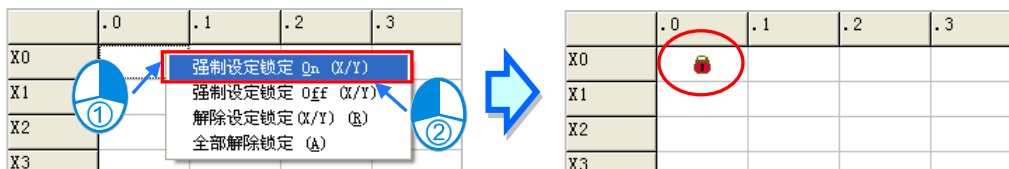
● 强制设定装置表

此功能可利用表格的方式，批次对 X 或 Y 接点的强制锁定 ON/OFF 状态进行管理。

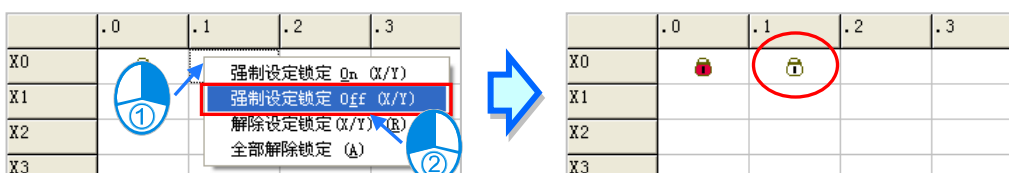
设定时，请在程序画面中按下鼠标右键，之后在快捷选单中单击「强制设定装置表」；而开启表格后，请先切换上方的卷标选择要设定的装置种类，接着在表格中单击欲设定的接点并按下鼠标右键，之后便可通过快捷选单中的项目来进行相关操作。



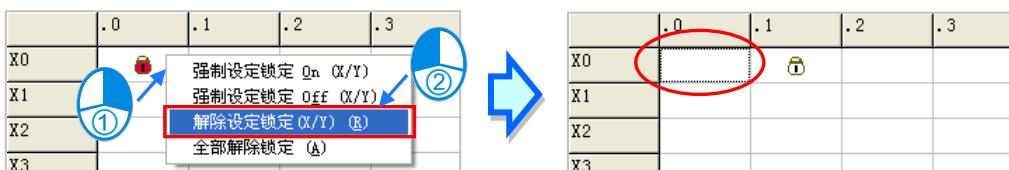
- 强制设定锁定 On (X/Y) : 将单击的接点强制设定为 ON。



- 强制设定锁定 Off (X/Y) : 将单击的接点强制设定为 OFF。



- 解除设定锁定 (X/Y) : 解除单击接点的强制状态。



- 全部解除锁定 : 不分 X 或 Y 装置，所有接点的强制状态将会全部解除。

	.0	.1	.2	.3
X0				
X1				
X2				
X3				

	.0	.1	.2	.3
X0				
X1				
X2				
X3				

*1.不同机种对于强制锁定 ON/OFF 的处理机制及点数限制可能不同，设定前请先参考各机种的相关使用说明。

*2.在离开在线模式后，各接点的强制锁定状态将不会自动解除，请务必确认目前的状态是否会造成影响。

18.1.5 装置监控表

当进入在线模式后，不论是「程序监控」或是「装置监控」模式，用户皆可通过监控表的方式来对程序中的装置或是变量符号进行监控；而在 ISPSOft 中，每个项目都允许建立一个以上的监控表，而每个监控表则至多可建立 100 个监控项目，用户可自行对所要监控的项目进行归类管理，且监控表的建立与管理皆可在脱机的状态下完成，而所建立的监控表亦将会随着项目一起储存，下次再开启项目时便无须重新建立。

开始新增一个监控表的方法有如下三种。

● 方法一

菜单工具栏中单击 **PLC 主机 (P)** > **符号表监控 (I)**。




● 方法二

在项目管理区单击「装置监控表」项目后按下鼠标右键，之后在选单中单击「新增」。

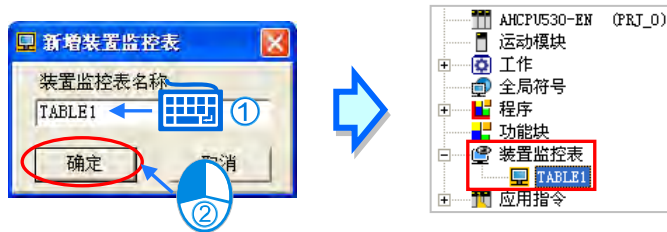


● 方法三

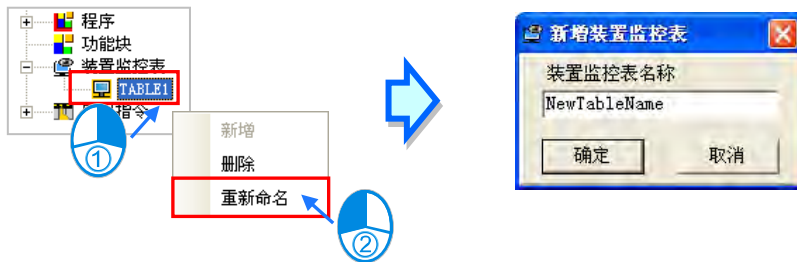
直接在图标工具栏中单击  图标。



依据上述的方式执行之后，画面便会出现一个弹出窗口，在该窗口输入监控表的名称后按下「确定」，之后在项目管理区的「装置监控表」项目下便会产生一个对象，并自动在编辑工作区开启该监控表的管理窗口，而后续便可在此窗口中进行监控项目的管理。



欲删除监控表时，请在项目管理区中单击欲删除的监控表项目后按下鼠标右键，之后在快捷选单中选择「删除」即可。当要重新命名已建立好的监控表时，请在项目管理区中单击欲重新命名的监控表项目后按下鼠标右键，接着在快捷选单中选择「重新命名」，之后便可变更该监控表的名称。

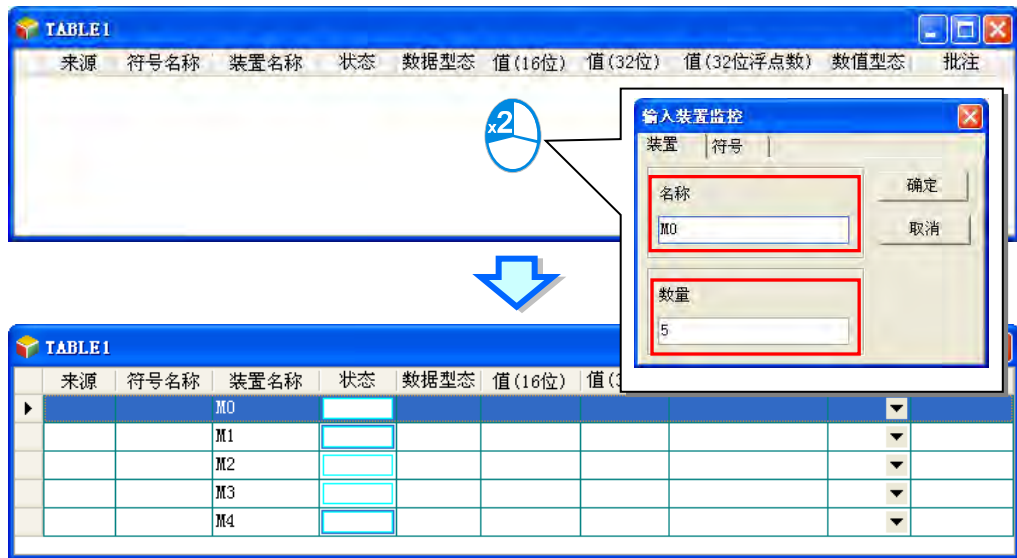


在 ISPSOft 中允许同时开启多个监控表窗口，而要开启时请移至项目管理区中，直接在欲开启的监控表项目上双击鼠标左键即可。

18



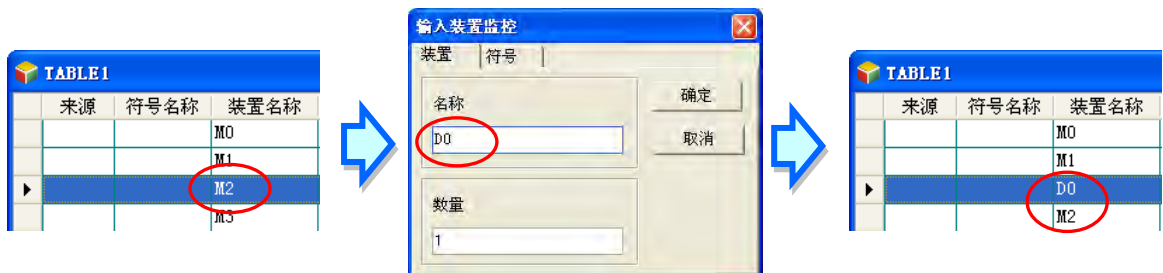
当要在监控表中加入监控项目时，请在监控表的空白处双击鼠标左键，或直接以键盘输入设备名称，之后画面便会出现设定窗口，直接在该窗口中输入启始的装置地址与欲加入的装置数量即可，不过须注意每个监控表至多只可建立 100 个监控项目。



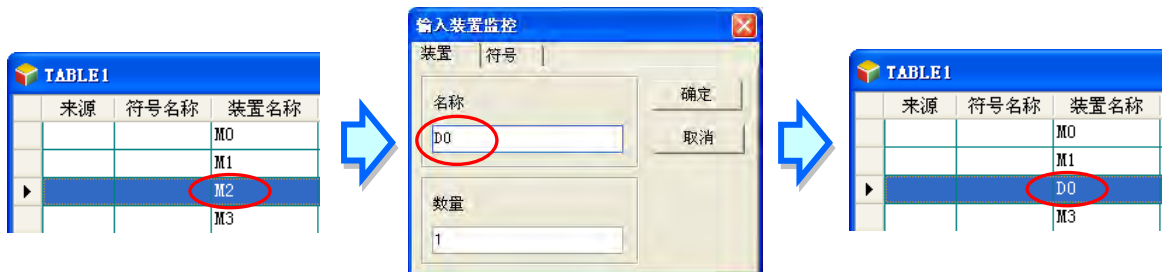
切换键盘的【Insert】键可选择插入或取代两种模式，且会显示在 ISPSoft 的状态栏中。



当处于插入模式时，新增的监控项目会插入在目前在监控表中被选取的项目上方。



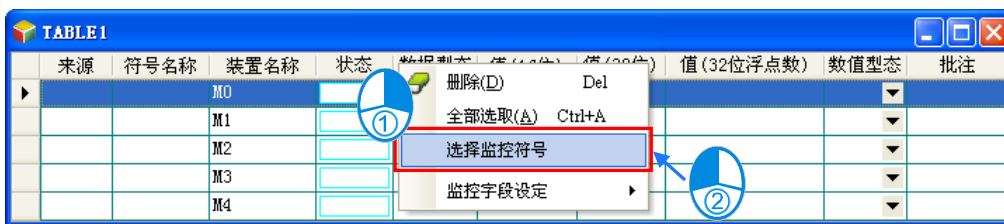
但若处于取代模式时，新增的监控项目则会取代目前被选取的项目。



在监控表中除了可加入装置之外，亦可直接在监控表中加入变量符号，

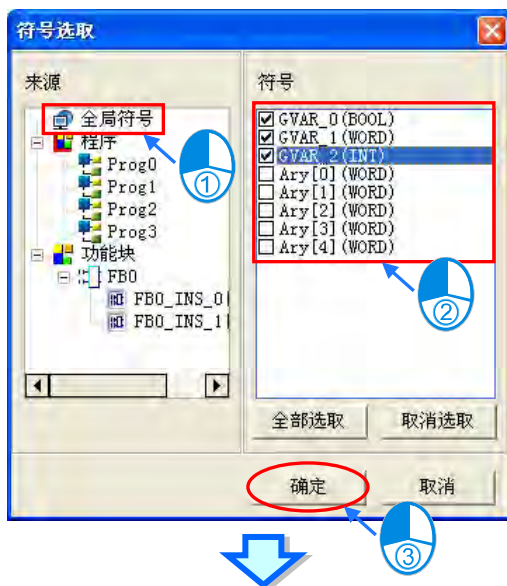


操作时只需在监控表中按下鼠标右键，之后在快捷选单中单击「选择监控符号」项目即可。



接着请在**符号选取窗口**的左侧选择符号表的来源，其来源可以是全局符号表或 POU 的区域符号表，其中若要选择功能块的区域符号时，必须先执行编译并产生功能块实例后，功能块内部的变量符号才会被配置一个实体的内存位置，而此时才可对其进行监控，且因不同的功能块实例会配置不同的内存位置，因此挑选功能块内部的区域符号时必须单击功能块实例的项目后才可进行选取。

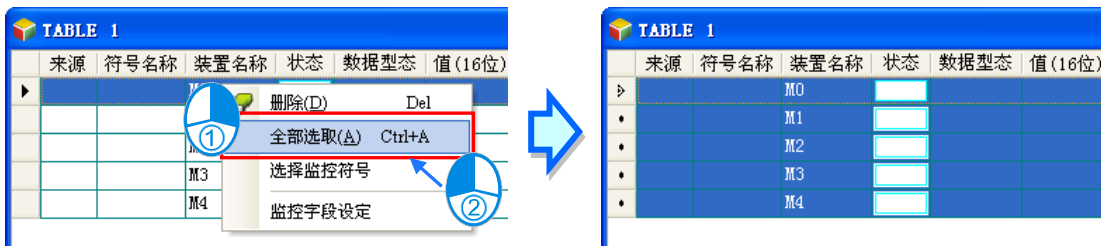
选取符号表的来源后，请再在**符号选取窗口**的右侧勾选欲加入监控的项目，其中若有数组符号时，列表中便会展开所有的数组元素；勾选完毕后按下「确定」按钮即可将所选项目加入监控表中，而之后可再继续选择其它的变量符号，当完成所有符号的加入之后，按下「取消」按钮即可关闭窗口。



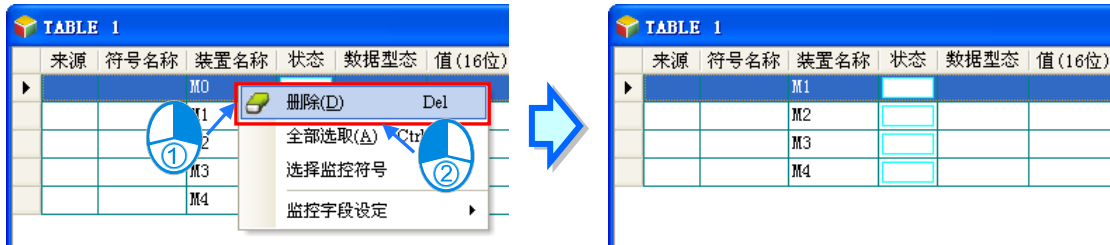
来源	符号名称	装置名称	状态	数据类型	值 (16位)	值 (32位)	值 (32位浮点数)	数值型态	批注
		M0	<input type="checkbox"/>						
		M1	<input type="checkbox"/>						
		M2	<input type="checkbox"/>						
		M3	<input type="checkbox"/>						
		M4	<input type="checkbox"/>						
Global Symbols	GVAR_0		<input type="checkbox"/>	BOOL					ERROR
Global Symbols	GVAR_1		<input type="checkbox"/>	WORD				有号数十进制	DATA
Global Symbols	GVAR_2		<input type="checkbox"/>	INT				有号数十进制	OFFSET

*. 指针类型的变量符号无法进行监控。

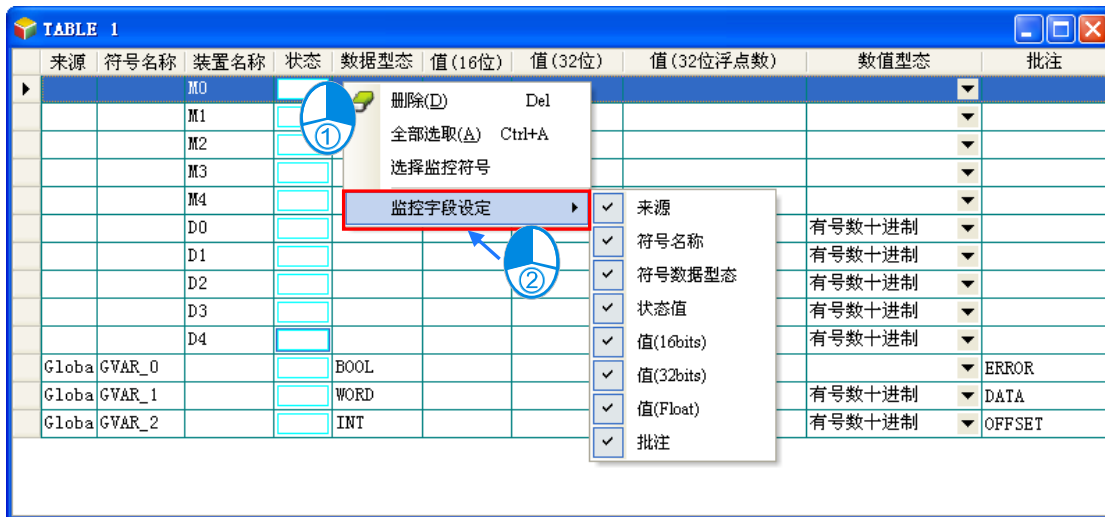
当要删除监控项目时，请先在监控表中单击欲删除的项目，而选取时若先按住键盘的【Ctrl】键则可连续选取多个项目；若要选取连续范围时，请先单击第一个项目后按住键盘的【Shift】键，之后再单击该范围的最后一个项目便可将整个范围选取起来；而若要选取全部的项目时，可直接在监控表中按下鼠标右键，接着在快捷选单中单击「全部选取 (A)」项目即可。



选取完欲删除的项目后，请在该项目上单击鼠标右键，之后在快捷选单中单击「删除 (D)」项目，亦或是直接按下键盘的【DEL】键即可。



为了便于视图，在使用时可将监控表中的部份字段设为隐藏，而设定时请直接在监控表中按下鼠标右键，接着在快捷选单中单击「监控字段设定」项目，之后可在其下阶的选项中设定欲显示的字段，取消勾选代表该字段将被隐藏。关于各字段的相关说明请参考下述。

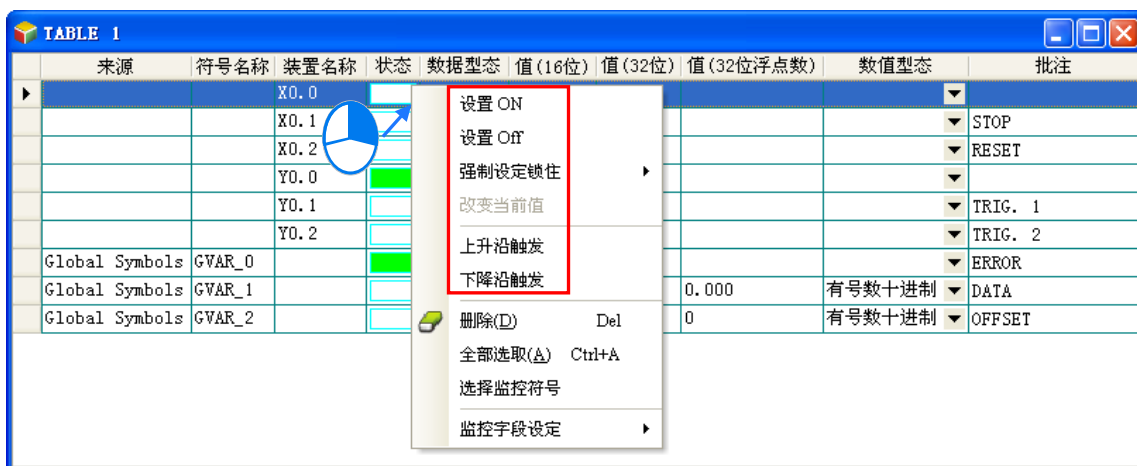


下表则为监控表中各字段的相关说明。

监控表字段	说明
来源	显示变量符号的来源。
符号名称	显示变量符号的名称。
装置名称	显示监控的装置名称。
状态	若监控的对象为位或接点时，此字段会显示其 ON/OFF 状态。
数据类型	若监控的对象为变量符号，则此字段会显示该变量符号的数据类型。
值 (8 位元)	实际连线监控时，此栏位会以 8 位元的类型来显示监控值。
值 (16 位)	实际联机监控时，此字段会以 16 位的类型来显示监控值。
值 (32 位)	实际联机监控时，此字段会以 32 位的类型来显示监控值。
值 (64 位元)	实际联机监控时，此栏位会以 64 位元的类型来显示监控值。
值 (32 位浮点数)	实际联机监控时，此字段会以 32 位的浮点数类型来显示监控值。

监控表字段	说明
值 (64 位元浮点数)	实际联机监控时，此栏位会以 64 位元的浮点数类型来显示监控值。
数值类型	在此下拉可选择联机监控时所显示的数值格式。
批注	显示监控装置的 装置批注 或监控符号的 符号批注 。

当建立好一个监控表后，接着可在在线模式下通过监控表来监控表格中的各个项目，而如同「程序监控」一般，在在线模式下，在**监控表**中单击某个项目后下鼠标右键，之后可利用快捷选单中的操作项来对监控项目的状态或内容值进行变更。关于各种操作项目的相关说明请参考第 18.1.4 节。



补充说明

监控项目一旦加入监控表之后，后续若变更监控装置的装置批注时，请先关闭监控表窗口后再重新开启以便取得更新；而若变更了变量符号的相关属性或将其由符号表中移除时，该监控项目将可能无法自动更新，且后续亦将无法在在线监控时产生作用，此时请将其删除后再重新加入监控表中。


18.1.6 在线编辑与在线更新

*.目前 DVPxxMC 机种不支持「线上编辑与线上更新」功能。

当系统处于「程序监控」的模式，且主机处于 RUN 的状态下，用户便可通过**在线编辑**与**在线更新**的功能来实时修改监控中的程序内容，而各种语言的编辑限制则请参考下表。

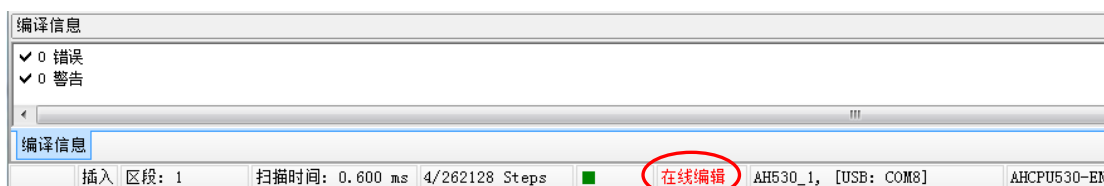
编程语言	说明
LD 语言	每次只可编辑一个程序区段，但允许新增区段，而一旦新增了一个区段后，可编辑的区域便会转移至该新区段中，亦即无法再回头编辑方才所编辑的区段。
FBD 语言	每次只可编辑一个程序区段，但允许新增区段，而一旦新增了一个区段后，可编辑的区域便会转移至该新区段中，亦即无法再回头编辑方才所编辑的区段。
IL 语言	没有特别的行数限制。
ST 语言	没有特别的行数限制。


编程语言	说明
SFC 语言	SFC 的图形无法在在线编辑时进行修改，但可以修改 Actions 与 Transitions 内部的程序，其限制则依据所使用的编程语言而与上述相同。
CFC 语言	没有特别的编辑限制。
共通限制	每次更新后只能编辑一个 POU。 在线编辑时无法修改或新增变量符号。

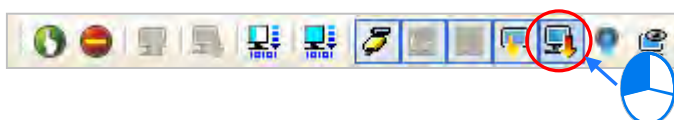
欲进行在线编辑时，请在「程序监控」的模式下，并确认主机处于 RUN 的状态后，直接在图标工具栏中按下  图标即可。




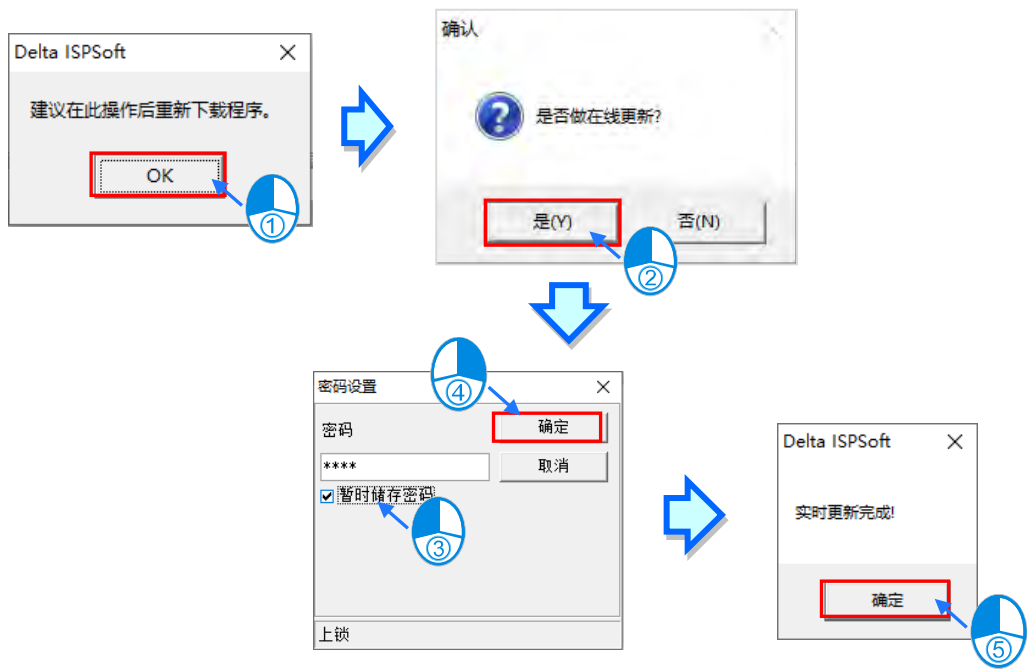
进入在线编辑之后，在 ISPSOft 的状态栏中便会显示为「在线编辑」的状态。




当完成程序的编辑并成功编译之后，按下图标工具栏的  图标便可将修改的程序下载至主机中进行更新，而若程序尚未完成编译时，系统便会主动要求用户进行编译的工作。



若是使用 DVP 机种，在单击  图标后，做下载时，出现「建议在此操作后重新下载程序」提示窗口，此时，若选择「OK」，接着单击「是 (Y)」，此时如果有设 PLC 密码就会弹出询问密码视窗，可以勾选暂时储存密码，这样直到项目重启之前使用在线更新就不会再次询问密码，接着单击「确定」，就会将目前编译后的程序执行码下载至 PLC。



若要将 PLC 原始码下载至 PLC，则需再单击 ，勾选「程序」选项，将 PLC 原始码下载进 PLC，才可同步 PLC 原始码。





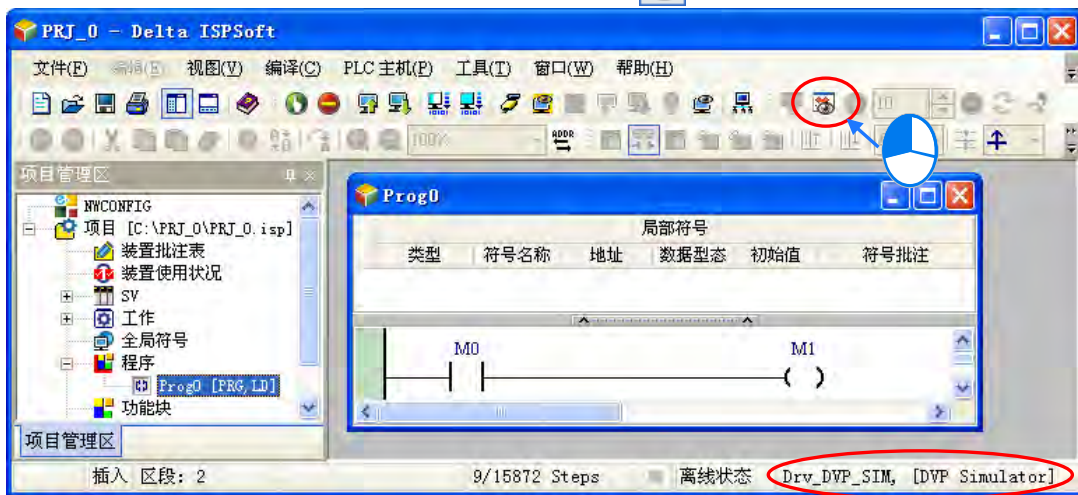
18.2 DVP 机种的除错模式

ISPSOft 针对 DVP 机种有提供**除错模式**的功能，而其内容包括了断点操作、单步执行、限定次数执行...等基本的除错工具；藉由这些工具，用户可方便的针对程序的部份进行侦错与测试。而在开始进入 DVP 的除错模式之前，请先留意下列两点的注意事项。

- (a) DVP 的除错模式仅能在仿真器的联机模式下执行。
- (b) 在线模式下无法启动**除错模式**，但仍可在进入**除错模式**之后启动**装置监控**的功能。

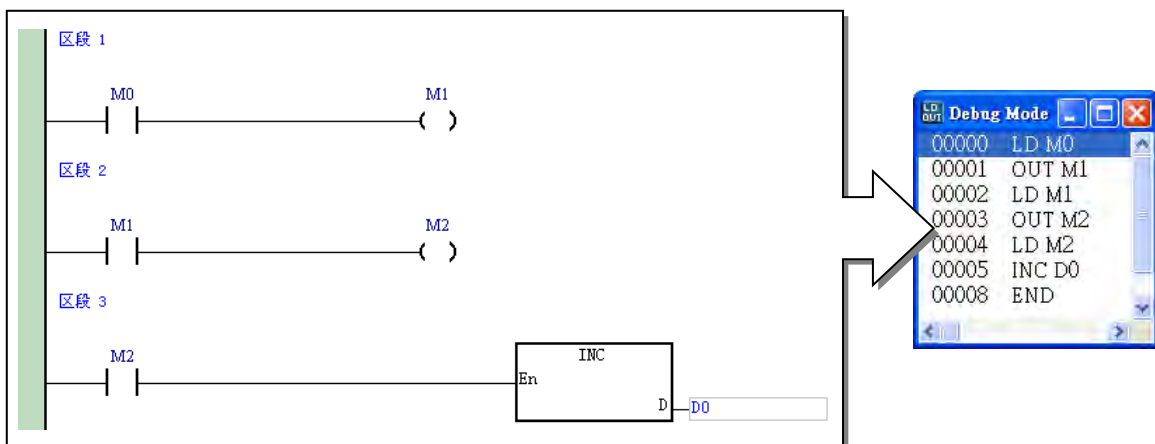
18.2.1 启动 DVP 的除错模式

首先请将系统设为 **DVP 仿真器**的联机模式，操作细节请参考第 2.4 节的相关说明，之后在图标工具栏中按下  图标即可进入除错模式，而此时画面便会会出现一个除错窗口，所有的除错工作都将该窗口中进行。后续若要离开除错模式时，只要再次按下  图标并恢复为未按下的状态即可。



DVP 的除错窗口是将程序编译后而重新以 IL 语言来所呈现，其中蓝色背景的行列为程序目前暂停的位置，而暂停的位置则是尚未执行的程序。

18

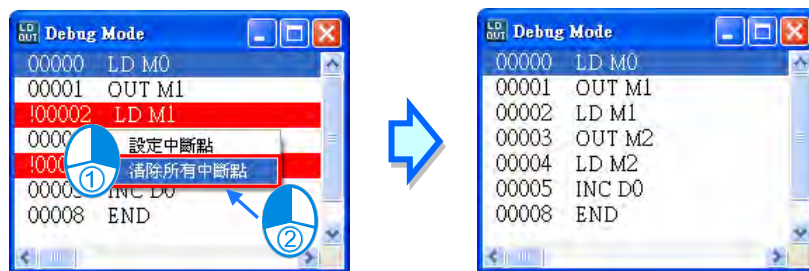


18.2.2 设定与清除断点

除错窗口中欲设定断点的行列上按下鼠标右键，并在快捷选单中单击「设定断点」项目即可加入一个断点，而在同一行号上再重复一次相同操作则可取消断点的设置。在除错窗口中，加入断点的行号前方会出现一个「！」的符号，且该行列的背景会以红色来显示。





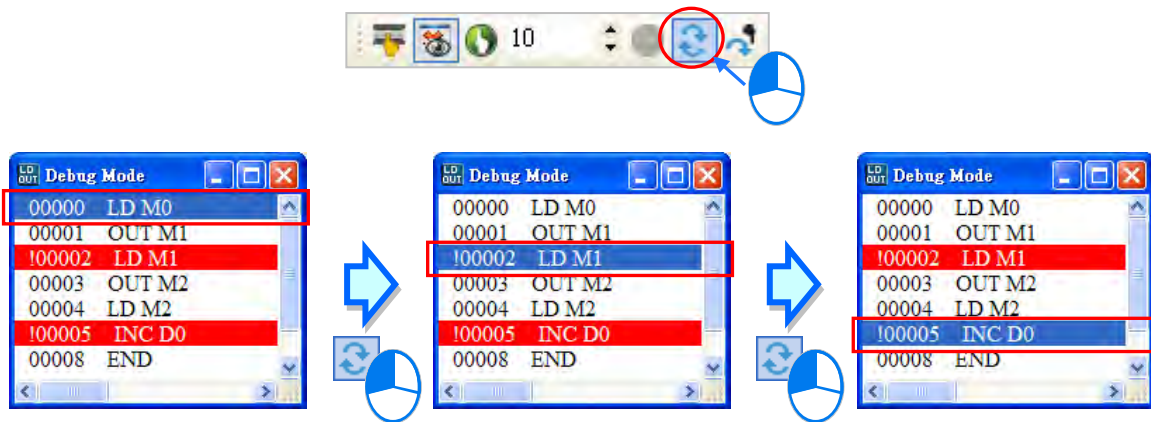
当要清除所有的断点设置时，请在除错窗口中按下鼠标右键，之后在快捷选单中单击「清除所有断点」项目即可。




18.2.3 除错模式下的程序执行

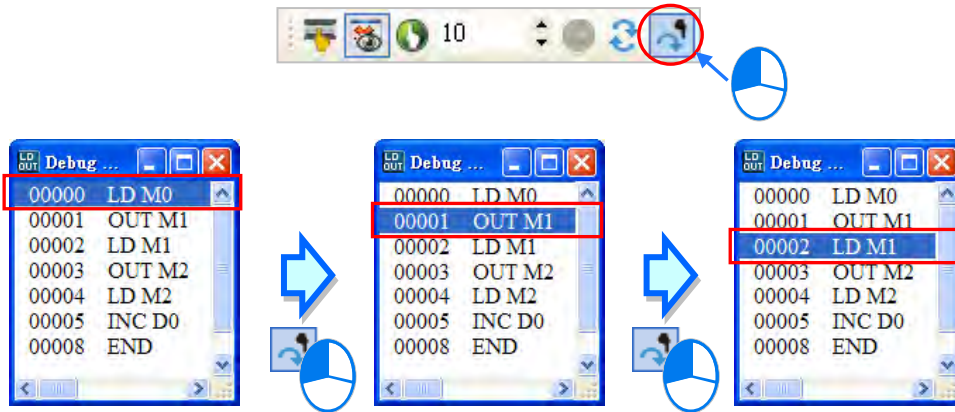
● 连续执行

按下图标工具栏的  图标便会开始连续执行到设定断点的程序为止，此时再次按下  图标便会继续执行到下一个断点，而若未设定任何断点时，程序便会不间断的来回扫描执行。




● 单步执行

在程序执行动作暂停的状况下，每按下一次图标工具栏的  图标可执行一步的程序。




● 限定次数执行

利用此功能可让程序在扫描周期的次数到达限定次数时自动停止，且过程中完全不受断点的影响。


执行前请先在图标工具栏的 字段上设定执行次数，至多可设定至 32767，完成后按下该字段左侧的  图标即可开始执行。

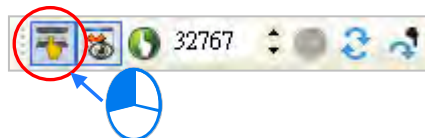


● 停止与重置

在执行过程中，只要按下同一工具栏的  图标后便会停止执行动作。



而按下图标工具栏的  图标可重置所有除错模式下的执行步序及状态值，而下次再执行时便会由程序的开头重新执行。




*.进入 DVP 的除错模式后，通过监控表仍可在除错模式的执行过程中监控各装置的变化状况。

18.3 AH/AS 机种的除错模式

18.3.1 进入除错模式

AH/AS 系列机种的除错模式可支持各种编程语言 (AH560 冗余系列机种, 在达成同步后, 不支持除错模式), 且虽然不同语言在画面的呈现上会有所差异, 但大致上的操作是相同的; 不过在 SFC 中, 除错模式的操作仅可在 Actions 当中进行, 且该 Actions 所属的步进点必须处于执行状态; 此外, 在除错模式下, 若 Transitions 的动作无法产生步进点的转移时, 用户必须自行在 SFC 的图面中切换转移点的 ON/OFF 状态来转移步进点。



先将主机切换至 RUN 并启动「程序监控」, 之后再按下图标工具栏的  图标即可进入除错模式。

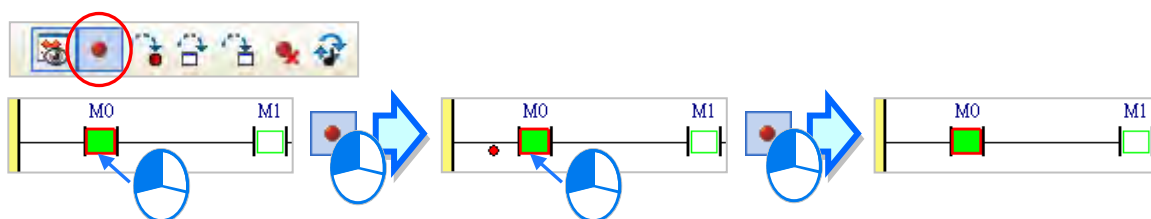



若在 ISPSOft 启动除错模式之前, 主机本身就已经在除错模式的状态中 (可由状态栏得知), 此时便无法再让目前的 ISPSOft 画面切换至除错模式; 其原因有可能是因为上次未正常退出除错模式或者是网络上的其他用户正在对该主机进行除错的工作; 若要重置主机的除错模式时, 可先将该主机的状态切换至 STOP, 接着再切换至 RUN, 然后便可依据上述的说明重新启动除错模式。




18.3.2 断点操作

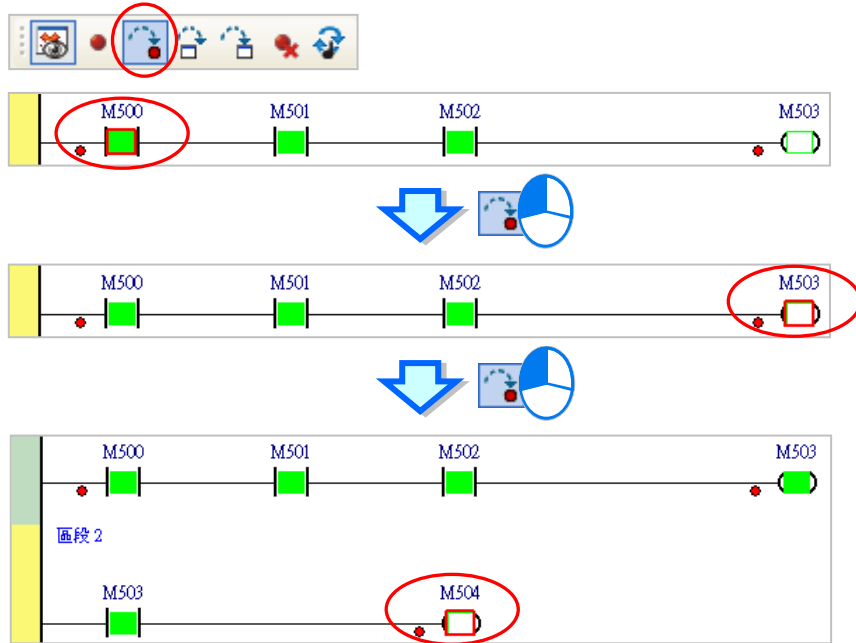
在 AH500 系列机种中, 一个项目程序 (包含所有 POU) 最多可设定至 10 个断点。欲设置断点时, 请先在程序画面中单击欲设置的位置, 之后按下图标工具栏的  图标便可加入断点; 同样的, 在已设定断点的位置上再次按下  图标则可将该断点取消。







当要清除所有的断点时, 请直接在图标工具栏中按下  图标即可。

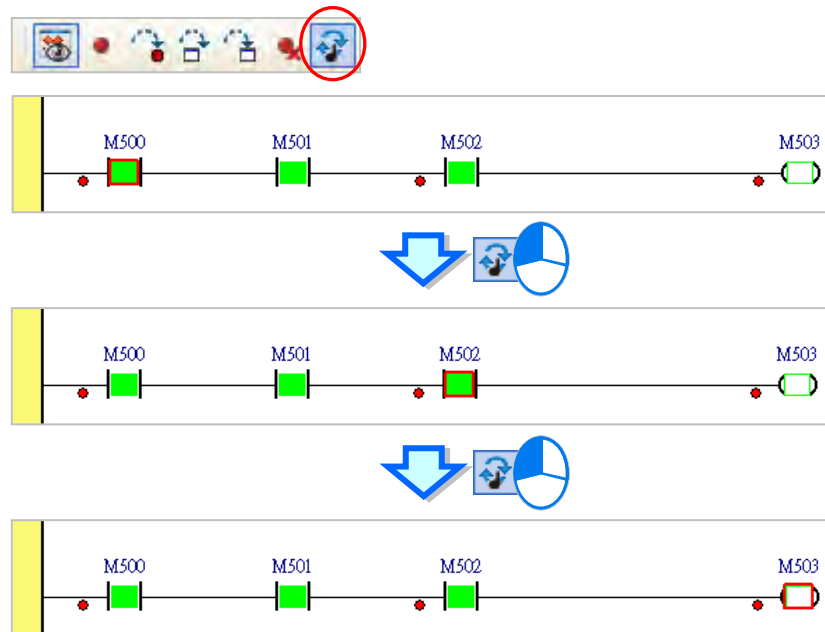


当程序中有设置断点时，每一次按下图标工具栏的  图标，系统便会自动执行程序至下一个断点后暂停，而暂停的位置则是尚未执行的程序。







18.3.3 连续执行



当按下图标工具栏的  图标后，系统便会开始连续执行到设定断点的程序暂停，此时再次按下  图标便会继续执行到下一个断点；但若未设置任何断点时，按下  图标后程序便会不间断的来回扫描执行，而在执行过程中，再把  图标按起便可停止程序的执行。

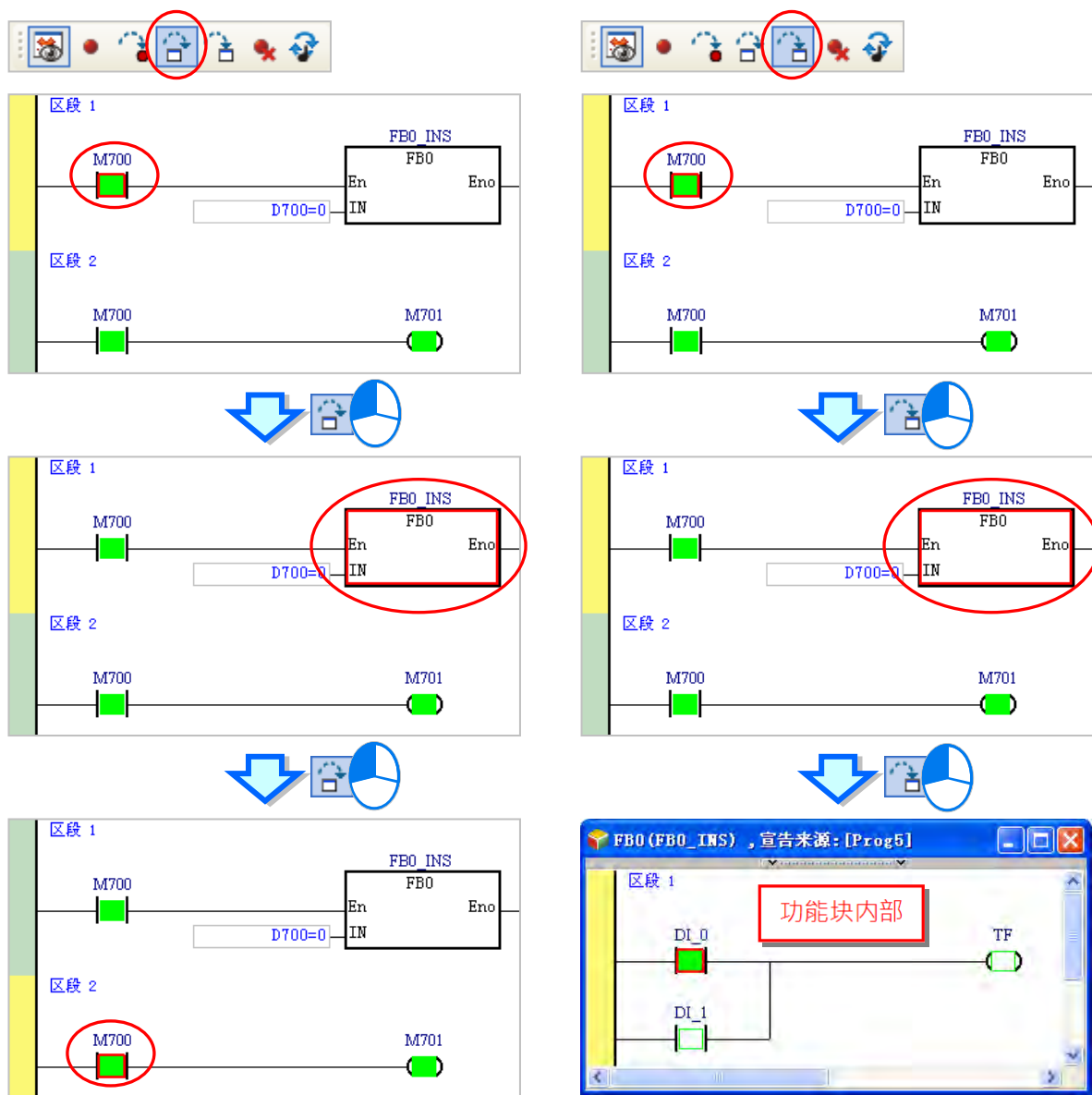


18

18.3.4 单步执行

每一次在图标工具栏中按下  图标或  图标时，系统便会在执行完目前位置的程序之后，在下一个程序之前暂停；至于两个图标的功能差别在于遇上**功能块**时，若按下  图标便会直接执行完功能块的功能，并在下一个程序前暂停；而按下  图标则会进入功能块当中，并在功能块内部继续进行单步执行。不过倘若功能块有设定 POU 密码保护或者功能块 EN 接点的状态为 OFF 时，不论按下哪一个按钮，单步执行皆不会进入功能块的内部。此外，在 ST 语言中进行单步执行时，必须配合断点的设置才可正常操作，而 SFC 的 Action 与执行顺序在 SFC POU 之后的其他 POU 亦必须设定断点才可正常操作。

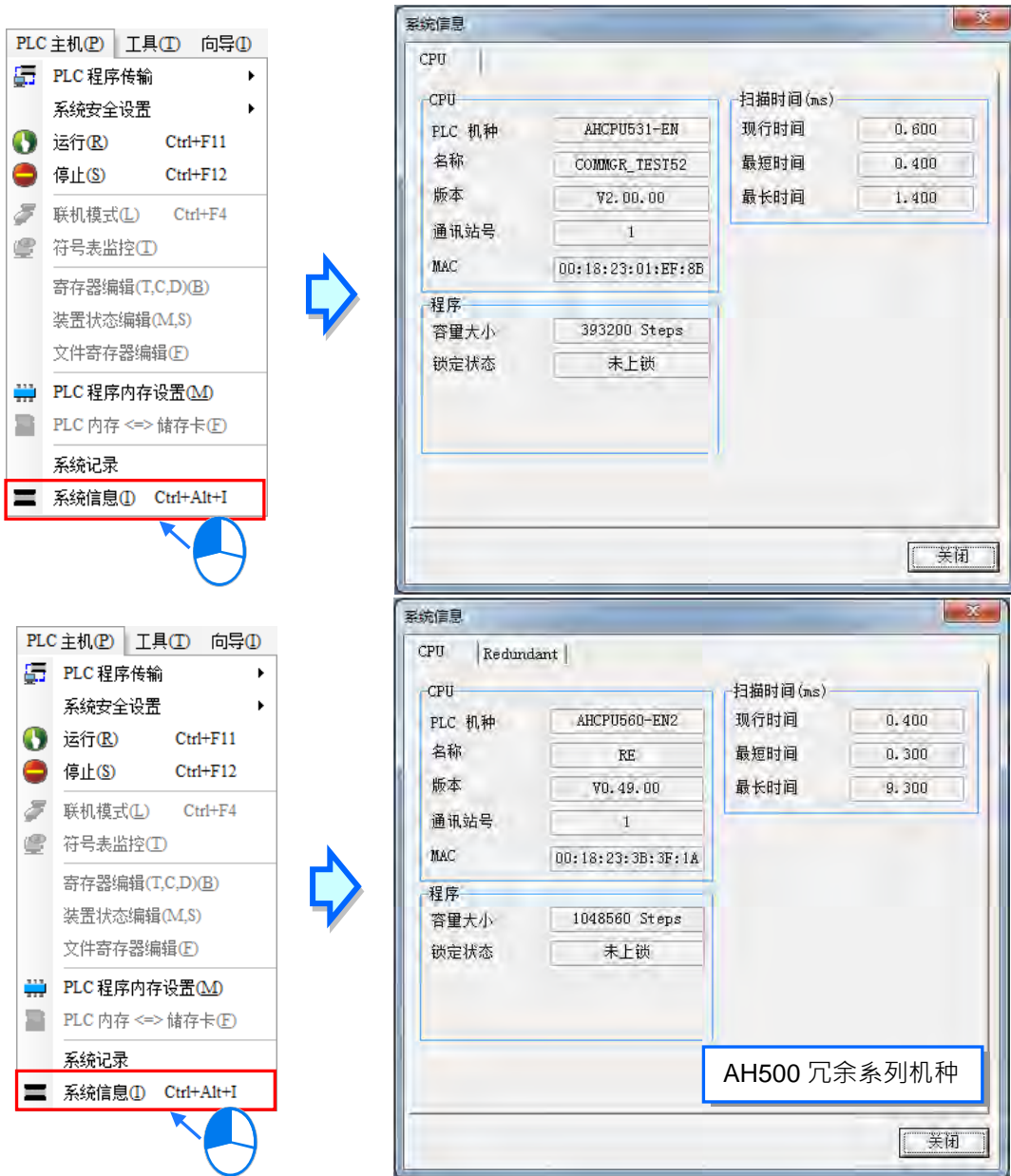
下列图标则分别以  与  图标来比较进行单步执行时的差异。



18.4 确认 PLC 状态

18.4.1 PLC 系统信息

在 ISPSOft 可与主机正常联机的情况下，在菜单工具栏单击 **PLC 主机 (P)** > **PLC 系统信息 (I)**，系统便会主动抓取主机内部的相关信息，并将其显示在画面当中；若是 AH560 冗余系列机种，则系统信息内会多一个「Redundant」分页。



Redundant 分页信息为当前冗余系统状态或执行相关设定，此页面仅在当前系统状态为「冗余系统就绪」时方可进行操作；在「单机模式」时只会显示冗余系统状态信息)。如下图。



Redundant 分页内的系统信息介绍如下：



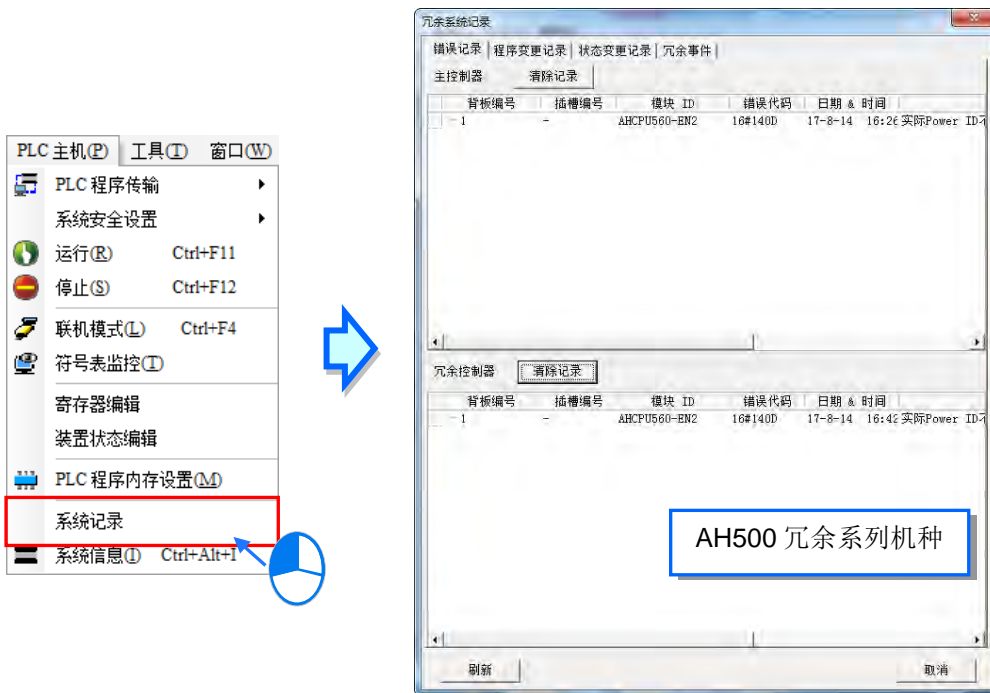
- ❶ **CPU ID** : A 与 B 为控制器的识别代码，可以检查识别代码来判断冗余系统是否发生控制器切换，也可以利用指令 API2901 「RCS」来进行设定或读取。
同步设定 : 冗余系统的资格检查与同步动作的设定，可设定的选项为「Always」及「Condition」；相关内容与检查条件，请参阅该产品相关手册。
下载设定 : 设定完 **CPU ID** 与 **同步设定** 项目后，必须单击此处下载到 PLC 主机。
- ❷ **启动同步** : 主控制器对冗余控制器发起资格检查动作，如通过资格检查则进行同步
取消同步 : 取消冗余控制器资格
控制系统切换 : 执行控制器切换，冗余控制器必须符合资格才允许控制器切换
取得控制权 : 在没有主控制器存在时，使用此功能将冗余控制器转换成主控制器

- ③ 显示当前控制器当前同步数据量与历史最大同步数据量单位为 (KBytes)，也可以利用特殊寄存器「SR18」~「SR21」进行确认。
 - ④ **当前连接控制器**：显示当前所链接的为主控制器或冗余控制器
当前系统状态：显示的状态为「冗余系统验证中」、「冗余系统就绪」、「单机模式」、「冗余系统失败」及「光纤同步电缆未连接」。
 - ⑤ **更新**：更新 Redundant 分页内的所有信息
- *.Redundant 分页内配合 AH560 冗余系列机种详细操作内容，请参阅该产品相关手册。

18.4.2 PLC 系统记录

与主机正常联机的情况下，于菜单工具栏中单击 **PLC 主机 (P) > 系统记录**，便可检视程序的变更及错误记录；若是 AH560 冗余系列机种，则可以检视主控制器及冗余控制器的记录，如果没有侦测到主控制器/冗余控制器，则主控制器/冗余控制器检视区将呈现反灰。「系统记录」窗口中总共包含数个页面，并可通过上方的标签来进行切换，所显示的数据不会实时更新，但可点击「更新」按钮来更新数据。

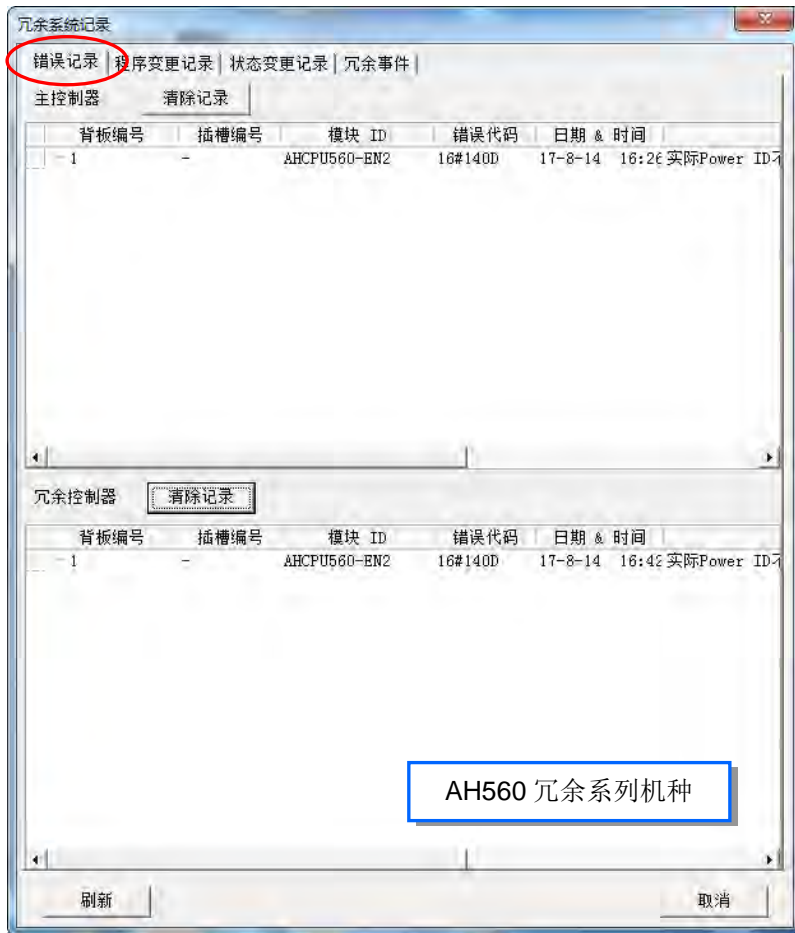
18




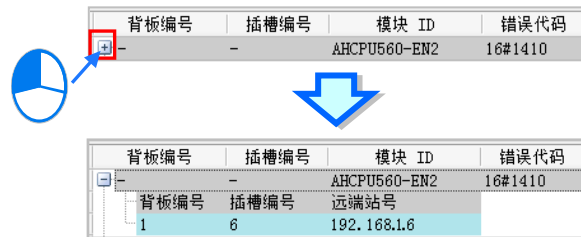
● 错误记录

于此页面可以检视主机及所有模块的错误记录，且在表格中会明确标示每笔记录的错误代码、发生时间及错误描述。另外还会包含产生该笔错误的模块 ID，以及该模块所在的背板与插槽编号。当按下「更新」按钮时，系统便会再次向主机撷取数据，并将其更新于表格中；而按下「清除记录」时，除了会将表格与主机内部的错误记录清除之外，还会一并重置主机的错误状态，但如果发生错误的原因依旧存在时，主机便仍会停留在错误状态中，并会重新产生新的错误记录。



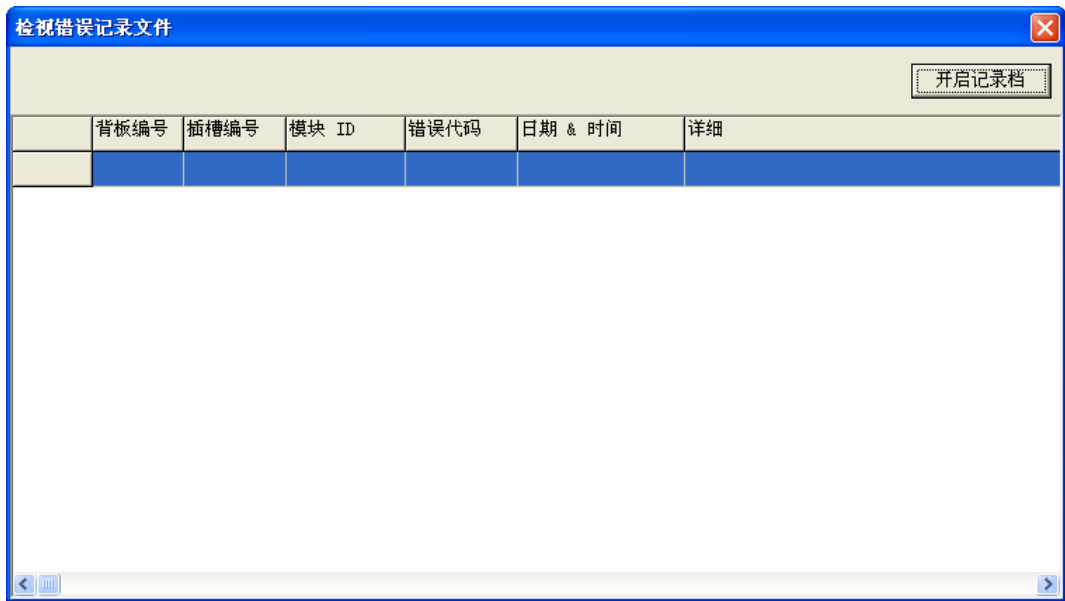


在主机有连接到远程模块时，错误纪录的检视会提供远程模块所在背板编号、插槽编号及 RTU 模块远程站号，需检视此类数据可单击  来做检视。



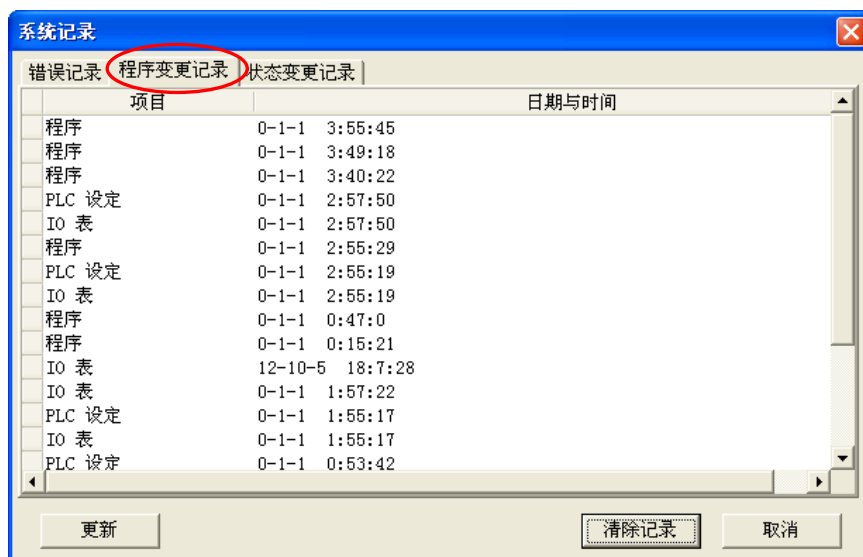
18

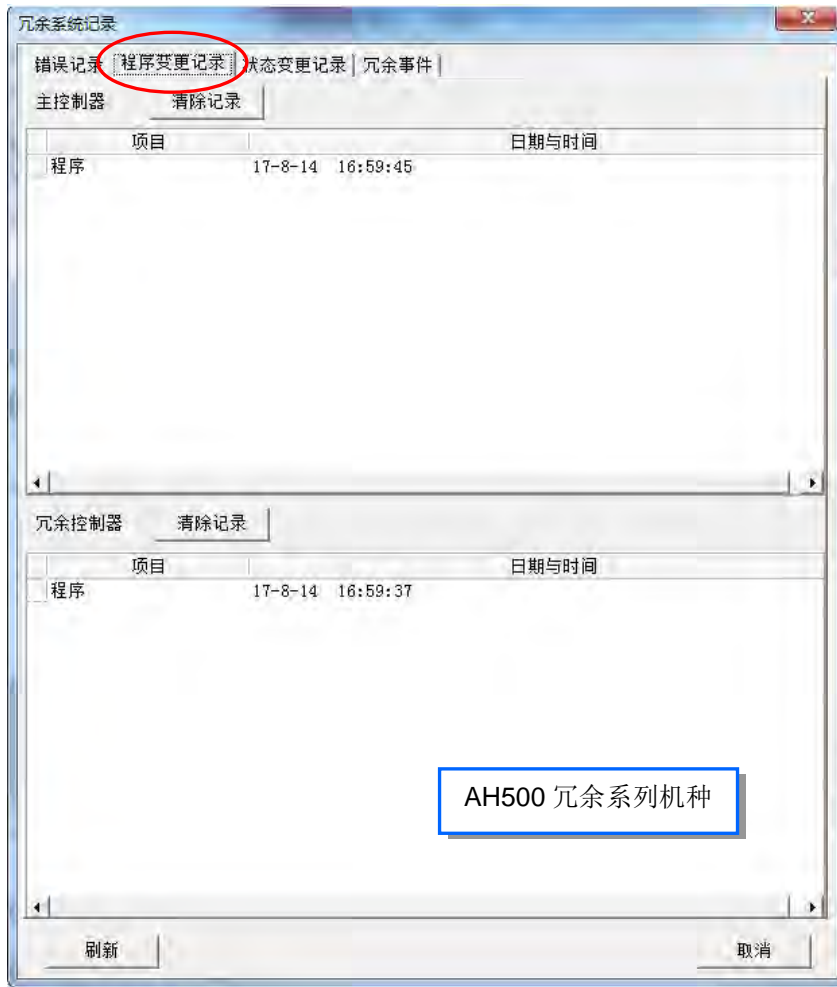
另外，错误记录可设定存放至 SD 卡后，再将记录文件搬移到计算机管理，设定方式请参考第 3 章。此时可以单击 **工具 (I)** > **检视错误记录文件**，按下「开启记录档」按钮，选择错误纪录存放的路径，便视图历史错误记录内容。



● 程序变更记录

此页面会显示对该主机进行上下载的执行记录，其中包括项目程序与规划参数的上下载；同样的，当按下「更新」按钮时，系统便会再次向主机获取数据，并将其更新在表格中；而按下「清除记录」按钮时，系统便会将表格与主机内部的程序变更记录一并清除。

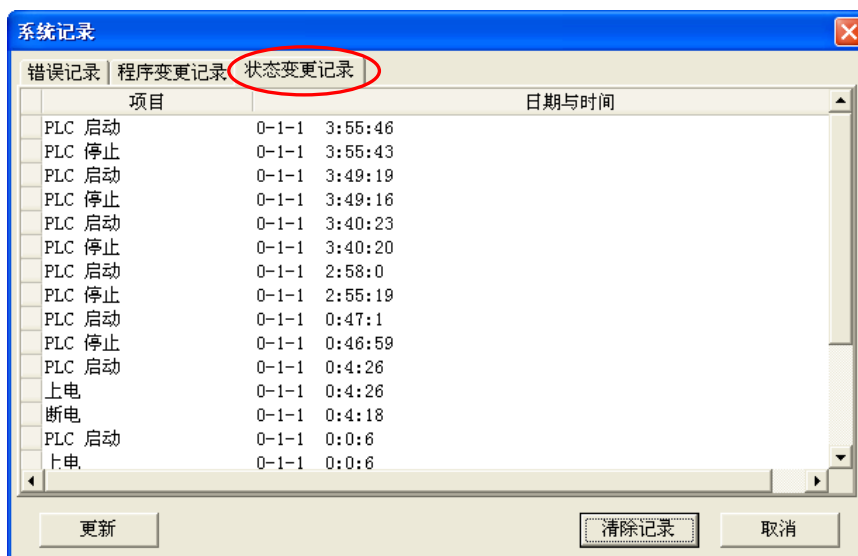


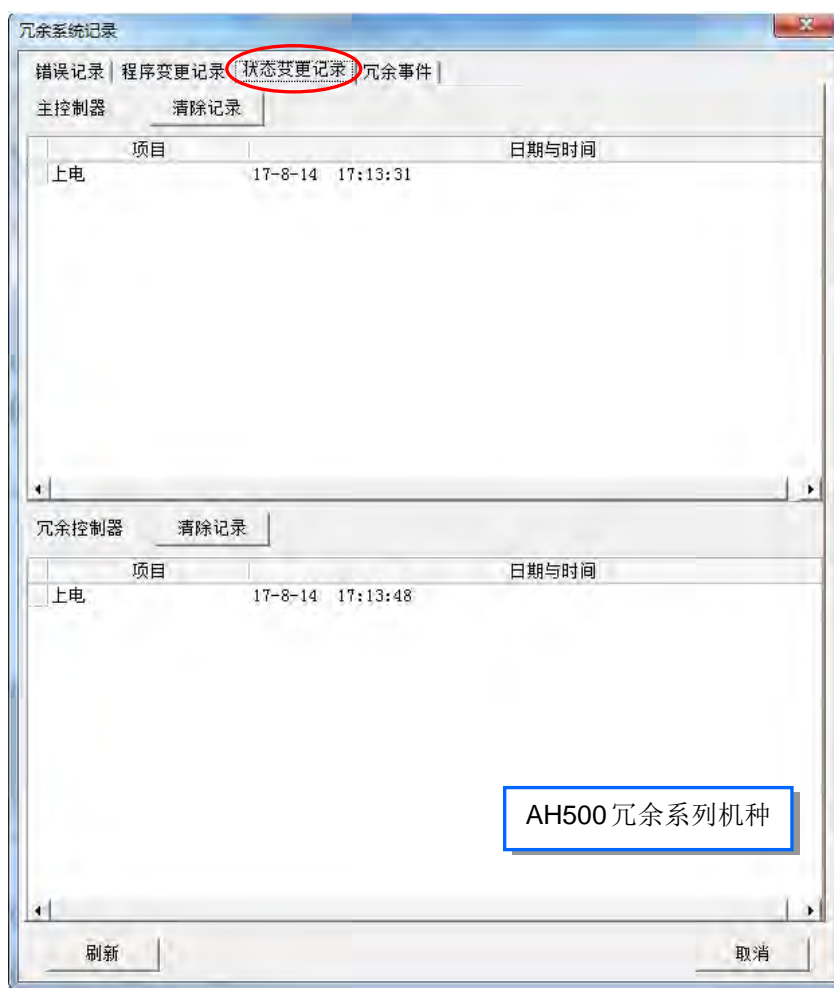


● 状态变更记录

此页面会显示主机状态的变更记录。当按下「更新」按钮时，系统便会再次向主机获取数据，并将其更新在表格中；按下「清除记录」按钮时，系统便会将表格与主机内部的状态变更记录一并清除。

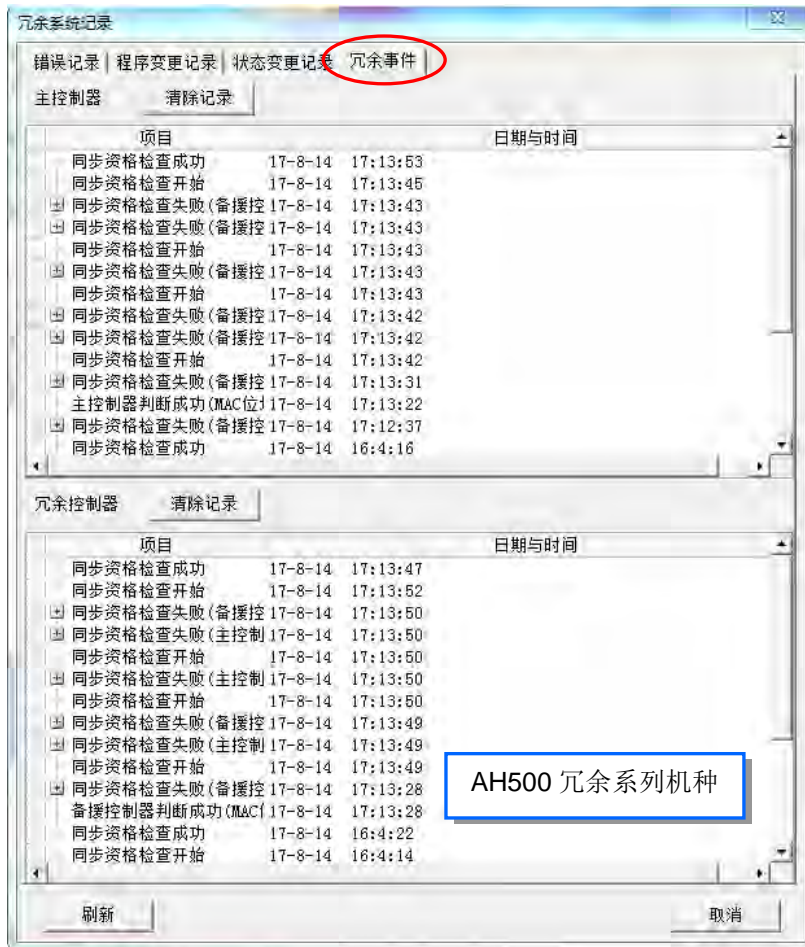
18





● 冗余事件

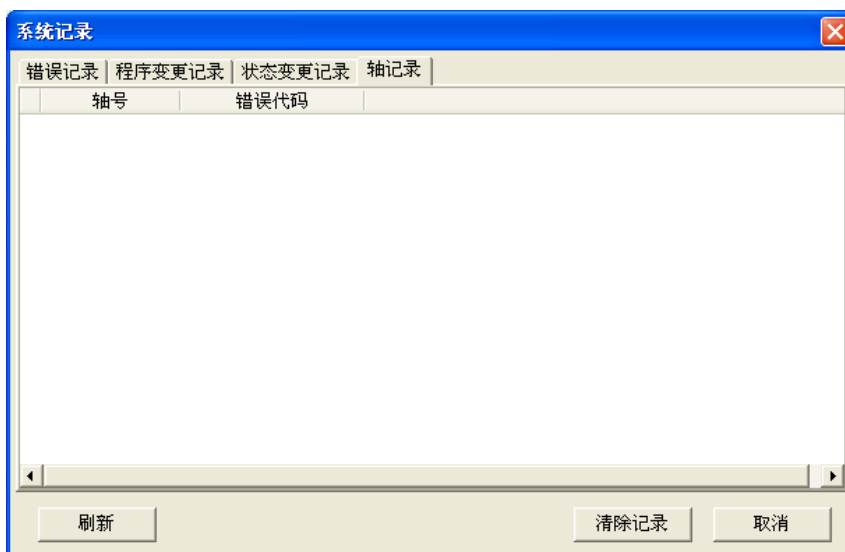
此页面会显示主控制器及冗余控制器的事件记录。当按下「更新」按钮时，系统便会再次向主控制器及冗余控制器获取数据，并将其更新在表格中；而按下主控制器「清除记录」按钮时，系统便会将主控制器表格与主控制器内部的冗余事件一并清除，若按下冗余控制器「清除记录」按钮时，系统便会将冗余控制器表格与冗余控制器内部的冗余事件变更记录一并清除。



*.冗余事件目前仅支持 AH560 冗余系列机种。

● 轴纪录

页面会显示关于轴的错误记录。(轴号栏位会显示轴号,而轴号括弧内会显示轴符号名称)。当按下「更新」按钮时,系统便会再次向主机获取数据,并将其更新在表格中;而按下「清除记录」按钮时,系统便会将表格与主机内部的轴纪录一并清除。



*.轴记录目前仅支持 AHxxEMC 机种。

18.5 3D 坐标图

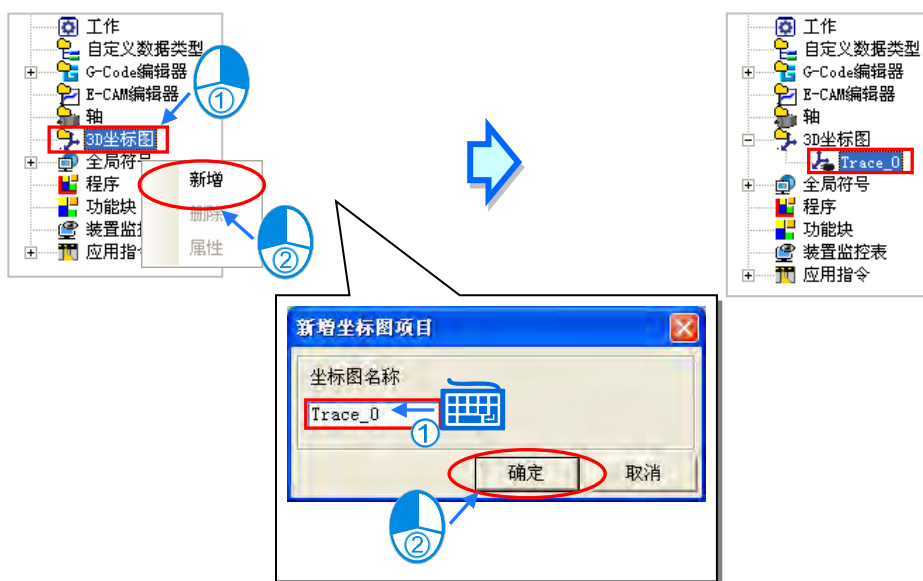
18.5.1 功能说明

监控模式中，当运动程控多轴运动时，可搭配 3D 坐标图功能画出多轴间的运动轨迹，提供用户判断运动的正确性；也可监控任何变量符号或装置以分析数值趋势。

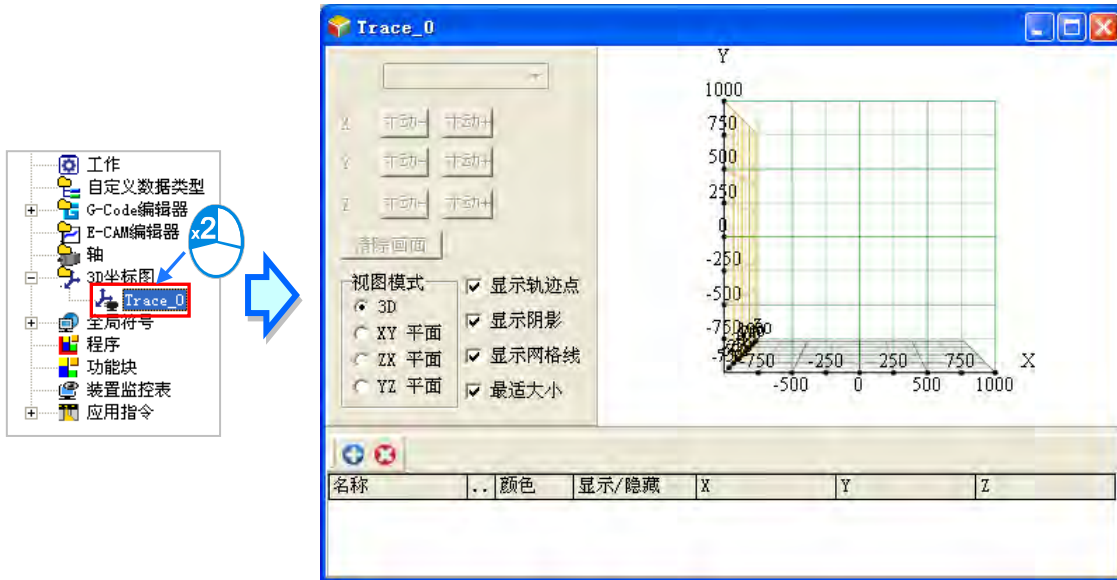
*.本功能目前仅支持 AHxxEMC 机种。

18.5.2 建立 3D 坐标图

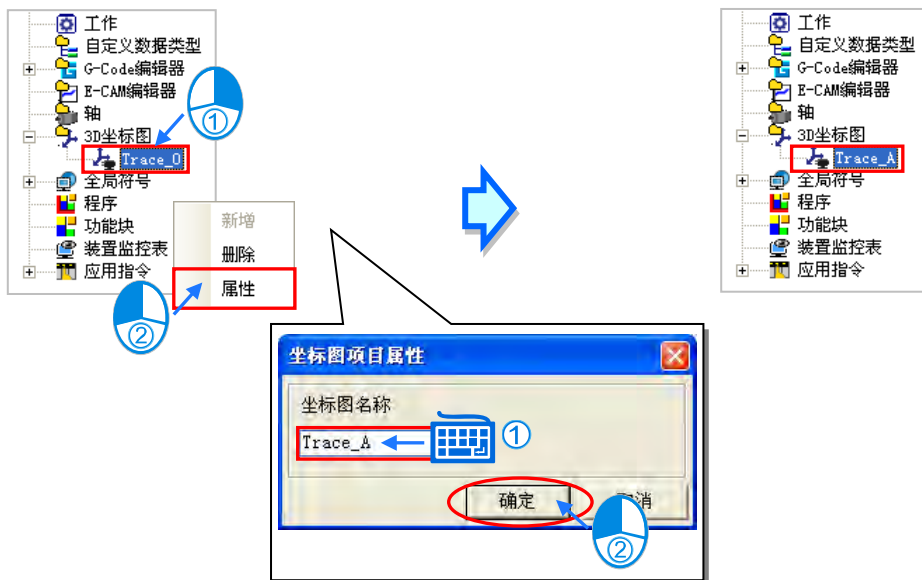
项目管理区的 3D 坐标图项目按下右键，单击「新增」，输入坐标图名称，建立 3D 坐标图项目。项目最多可加入 10 组 3D 坐标图项目。



一旦建立了一个 3D 坐标图项目后，便会自动开启此 3D 坐标图页面。关闭窗口后用户也可自行双击此 3D 坐标图项目再次开启窗口，如下图。




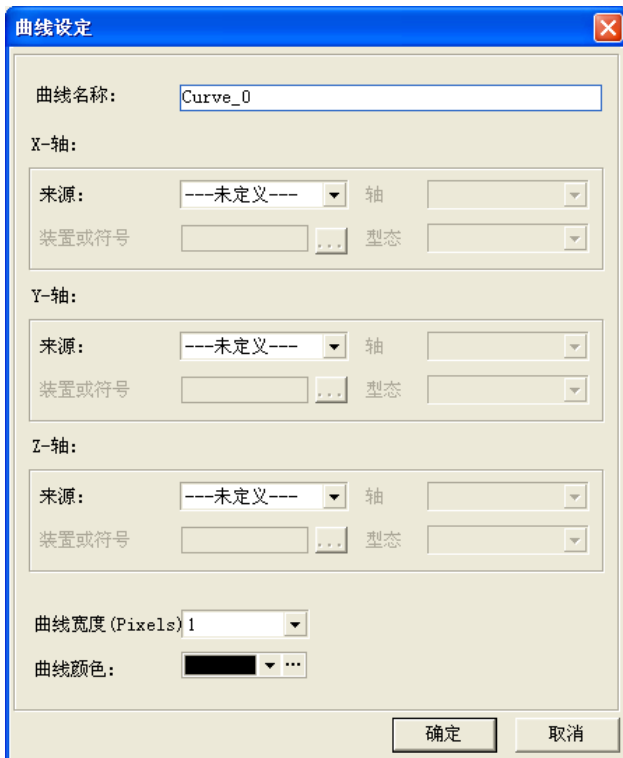
在已建立的 3D 坐标图项目上单击鼠标右键，可在快捷选单单击「删除」，删除此项目；或是单击「属性」，重新设定项目名称。



18

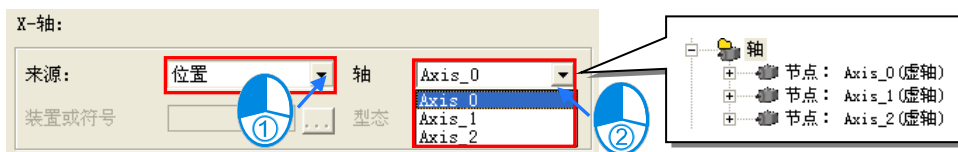
18.5.3 加入曲线

单击 3D 坐标图页面下方的  按钮新增一条曲线，出现曲线设定窗口如下图。



输入**曲线名称**，分别选择坐标轴的 X-轴、Y-轴、Z-轴的来源，不同的来源需搭配不同的设定字段，说明如下。

当来源选择为**位置**或**速度**时，代表此坐标轴数据源为任一轴项目的位置或速度，因此可在**轴**的下拉选单选择已建立的轴项目。当主机执行运动程序时，便可实时监控此轴的位置或速度。

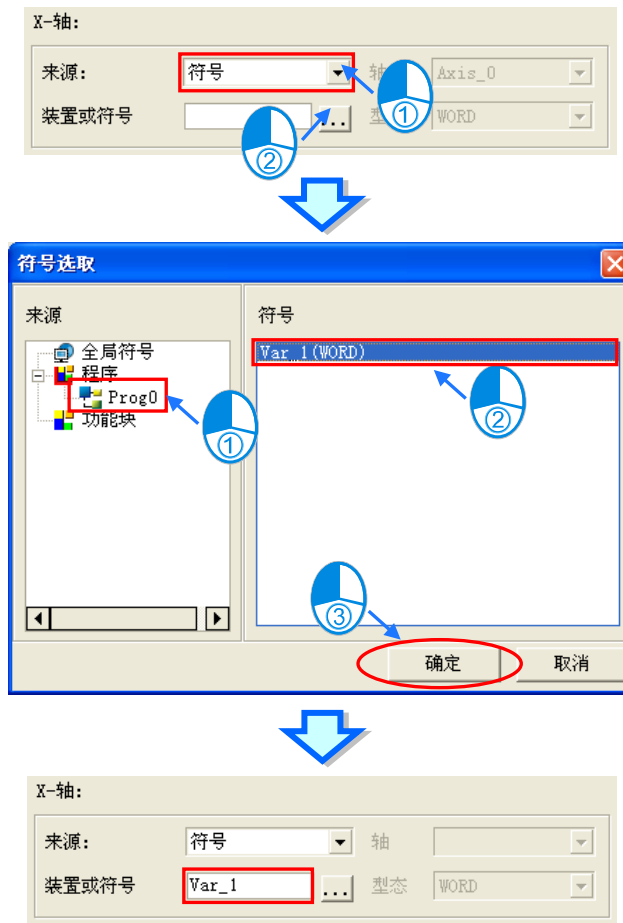


当来源选择为**装置**时，可直接在**装置或符号**字段输入设备地址，例如 DO，在**型态**字段下拉选单选择监控此装置的类型，如下图。

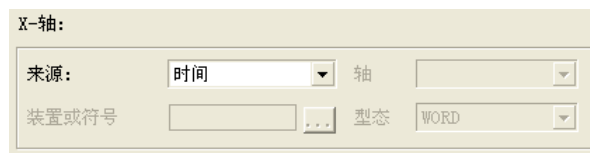


当来源选择为**符号**时，可单击  按钮开启符号选取窗口，由已建立的全局符号或区域符号选择要监控

的变量符号，并且会根据此变量符号的设定，自动带入类型。



当来源选择为时间时，此坐标轴数值将随时间每秒递增。



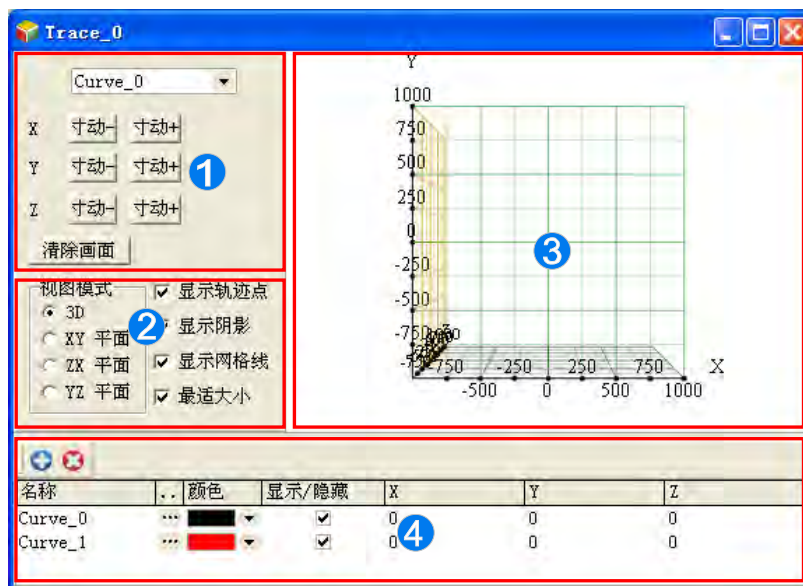
18

再在曲线设定窗口选择曲线宽度与曲线颜色，按下「确定」按钮完成曲线设定，便会将此条曲线加入 3D 坐标图下方列表。可选择某条曲线后按下 按钮删除该条曲线；按下 按钮可再次开启曲线设定窗口；由颜色下拉选单改变曲线颜色，勾选选显示/隐藏域，切换是否显示此条曲线在 3D 坐标图中。一组 3D 坐标图最多可加入 10 条曲线。

名称	颜色	显示/隐藏	X	Y	Z
Curve_0		<input checked="" type="checkbox"/>	0	0	0

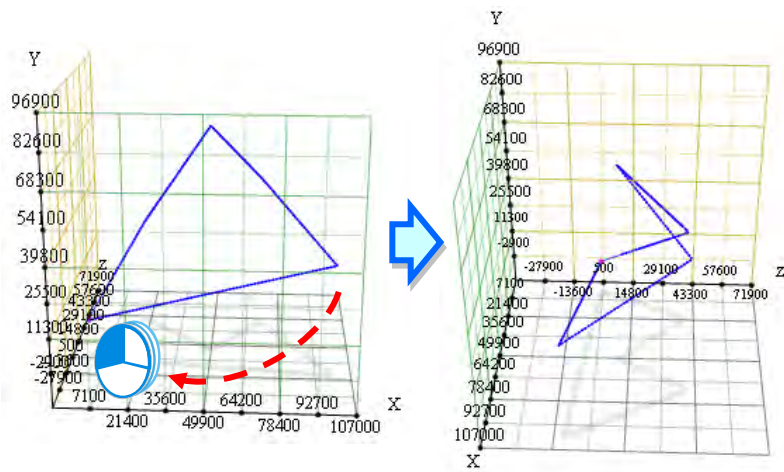
18.5.4 显示 3D 坐标图

完成 3D 坐标图的建立之后，确认运动主机与计算机已建立联机，再依照前述章节完成程序下载与执行监控功能后，便可显示 3D 坐标图。说明如下。

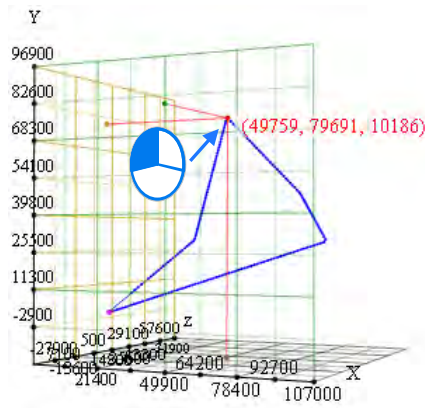


- ❶ 由下拉选单选择区域❹已建立的曲线，再分别对各个坐标轴单击寸动-或寸动+进行正负方向的寸动，或按下清除画面按钮，清除已绘制的 3D 坐标图。
- ❷ 在视图模式可切换 3D 的视角，或是由 X/Y/Z 轴所构成的平面。其他的选项可以切换是否显示轨迹点、阴影效果与坐标网格线；当勾选最适大小时，右方的显示区域将会根据目前轨迹的大小自动调整坐标轴范围。
- ❸ 3D 坐标图显示区域。
- ❹ 建立 3D 坐标图中要显示的曲线。

3D 坐标图轨迹图形可使用鼠标左键拖曳图形后，旋转图形由不同角度观察，如下图，也可以鼠标滚轮调整坐标的刻度大小。



若鼠标光标指到轨迹上时，会出现坐标信息，如下图。



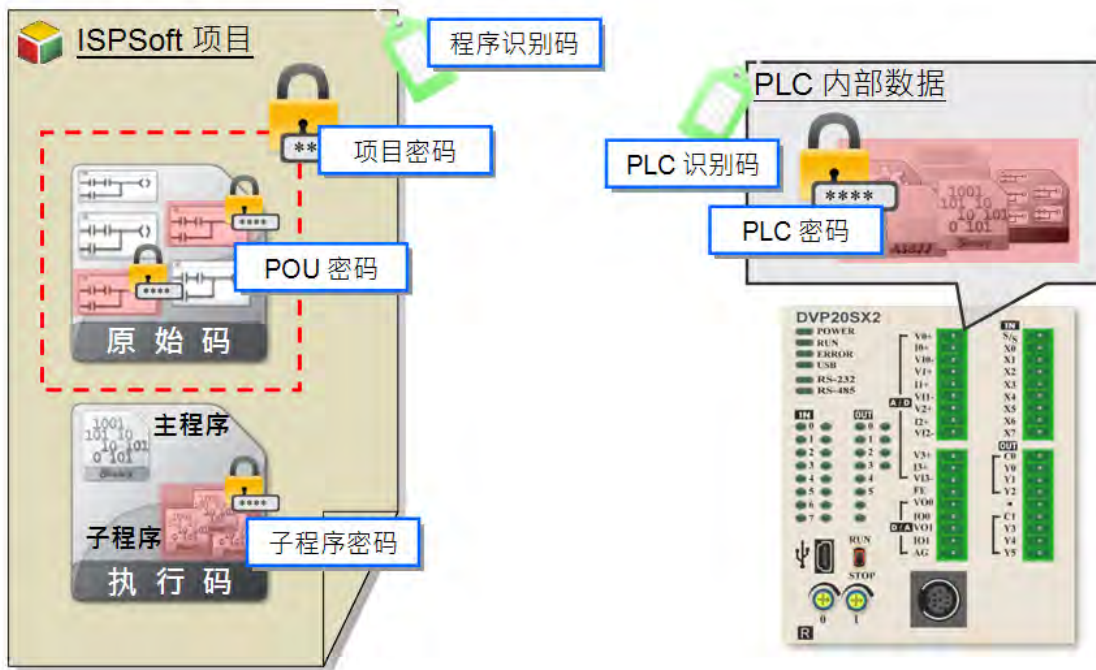
第19章 密码管理与数据保护机制

目录

19.1	ISPSOft 的密码架构	19-2
19.2	程序识别码与 PLC 识别码	19-3
19.2.1	程序识别码的设定与解除	19-3
19.2.2	PLC 识别码的设置与解除	19-4
19.3	项目密码与 PLC 密码	19-5
19.3.1	项目密码的设置与解除	19-5
19.3.2	PLC 密码的设置与解除	19-7
19.3.3	项目密码与 PLC 密码同步	19-8
19.4	POU 密码	19-9
19.5	子程序密码	19-10
19.5.1	子程序密码简介	19-10
19.5.2	子程序密码的设定与解除	19-11
19.6	其它的密码与数据保护功能	19-11
19.6.1	程序上载失效	19-11
19.6.2	只读装置区域设定	19-12
19.6.3	TC-01 密码钥匙设定	19-13

19.1 ISPSOft 的密码架构

针对使用者所开发的项目，ISPSOft 亦提供了多重的保护机制，其目的在于确保用户所开发的程序或核心技术都可以得到完善的保护。下列即为 ISPSOft 所提供的密码保护架构，其中每种密码都可选择性的设定或解除，使用者可依据实际的需求而采用不同的数据保护机制。



密码种类		说明	AS/AH AS5xx DVPxxMC	DVP
ISPSOft 项目	程序识别码	做为第一道把关的认证信息，且因进行项目下载时，设定于 ISPSOft 项目中的「程序识别码」与设定于主机中的「PLC 识别码」必须完全一	4~16 个字	4~8 个字
	项目密码	用来保护整体项目程序的密码。若某个项目启动了「项目密码」的保护功能，则当用户要开启该项目中的 POU 时，系统便会要求输入正确	4~16 个字	4 个字
	POU 密码	保护 POU 程序的密码功能，且每个 POU 皆可独立设定专属密码，用户可用来保护特定的程	4~16 个字	4~16 个字
	子程序密码	用来保护执行码中的子程序，可防止通过上传执行码的方式来解析功能块内部的程序。		4~8 个字
	设计工具密码	为了保护各项设计数据，不同机种对于支持的设计工具也提供密码保护，包括 E-CAM 密码与定位规划表上载保护密码等，将于各功能章节	4~16 个字	

密码种类		说明	AS/AH AS5xx DVPxxMC	DVP
PLC	PLC 识别码	做为第一道把关的认证信息，且因进行项目下载时，设定于 ISPSOft 项目中的「程序识别码」与设定于主机中的「PLC 识别码」必须完全一致才	4~16 个字	4~8 个字
	PLC 密码	用来保护 PLC 内部数据的密码功能，用户必须输入正确的密码之后才可对受「PLC 密码」保护	4~16 个字	4 个字

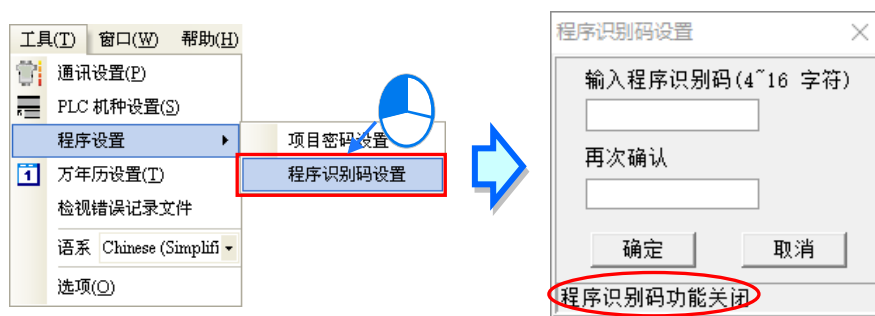
19.2 程序识别码与 PLC 识别码

识别码是做为第一道把关的认证信息，当进行下载时，设定于 ISPSOft 项目中的「程序识别码」与设定于主机中的「PLC 识别码」必须完全一致才可进行操作，因此识别码的保护机制可用来限制有设定「程序识别码」的项目只能下载至特定的主机中，而该主机必须设定相同的「PLC 识别码」。

进行项目程序的上传时，若该主机有设定「PLC 识别码」时，系统会在开始上传数据前要求用户输入正确的识别码以进行验证；而当开启一个有设定「程序识别码」的项目时，系统同样会要求输入正确的识别码后才允许开启，因此识别码可用来做为用户存取数据的权限认证。

19.2.1 程序识别码的设定与解除

功能工具栏中点选 **工具 (I)** > **程序设定** > **程序识别码设定**，之后便会出现识别码的设定窗口，而由该窗口的下方便可得知目前开启的项目是否有启动「程序识别码」的保护功能。



若项目目前的「程序识别码」功能为关闭状态时，窗口中便会有两个设定识别码的字段，而要设定识别码时，请先后于两个字段中输入相同的识别码后按下「确定」键即可。

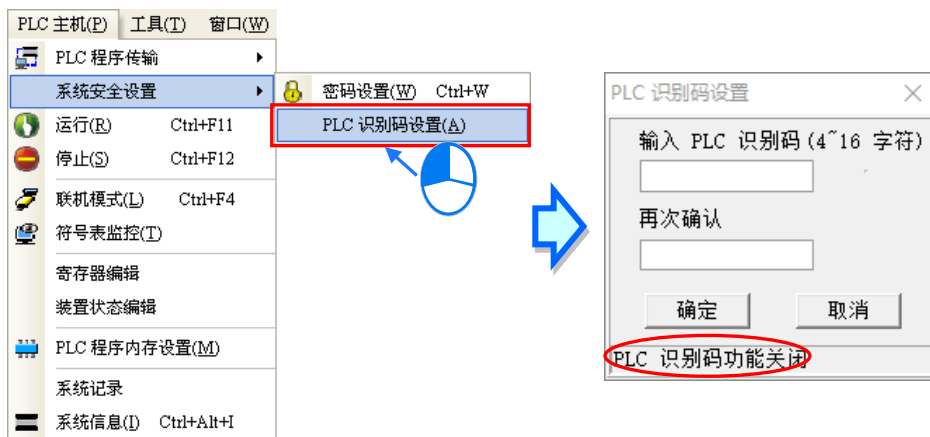
若目前「程序识别码」的功能为启动状态，则窗口中只会有一个用来解除识别码的字段，而要解除识别码时，请直接于此字段输入正确的识别码后按下「确定」键即可。

设定的识别码中仅能有英文字母、数字及空白，其中英文字母必须区分大小写；不同机种的识别码字数限制可能不同，请依照窗口字段的提示进行设定。当设定完成后，窗口的状态便会自动切换，确认无误后按下「取消」键便可关闭窗口。



19.2.2 PLC 识别码的设置与解除

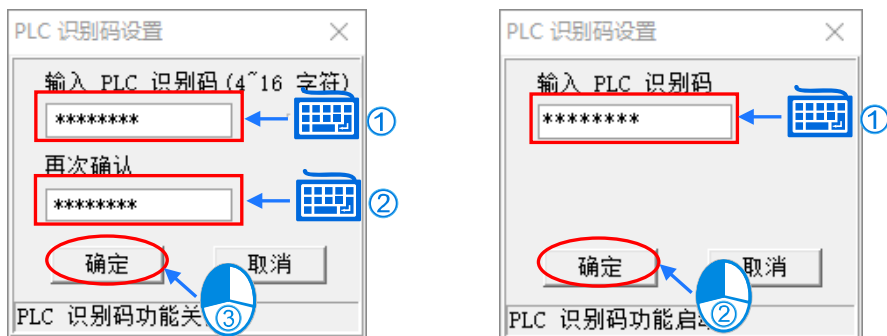
要设定「**PLC 标识码**」之前，请先确认 ISPSOft 已可与主机正常联机，点选功能工具栏中 **PLC 主机 (P)** > **系统安全设定** > **PLC 标识码设定 (A)**，之后 ISPSOft 便会主动联机至 PLC 并将主机的状态读取回来；当成功读取主机的信息之后，ISPSOft 便会自动开启识别码的设定窗口，而由该窗口的下方便可得知主机目前的「**PLC 标识码**」功能是否启动。



若主机目前的「**PLC 标识码**」功能为关闭状态时，窗口中便会有两个设定识别码的字段，而要设定识别码时，请先后于两个字段中输入相同的识别码后按下「**确定**」键即可。

若目前「**PLC 标识码**」的功能为启动状态，则窗口中只会有一个用来解除识别码的字段，而要解除识别码时，请直接于此字段输入正确的识别码后按下「**确定**」键即可。

设定的识别码中仅能有英文字母、数字及空白，其中英文字母必须区分大小写；不同机种的识别码字数限制可能不同，请依照窗口字段的提示进行设定。当设定完成后，窗口的状态便会自动切换，确认无误后按下「**取消**」键便可关闭窗口。



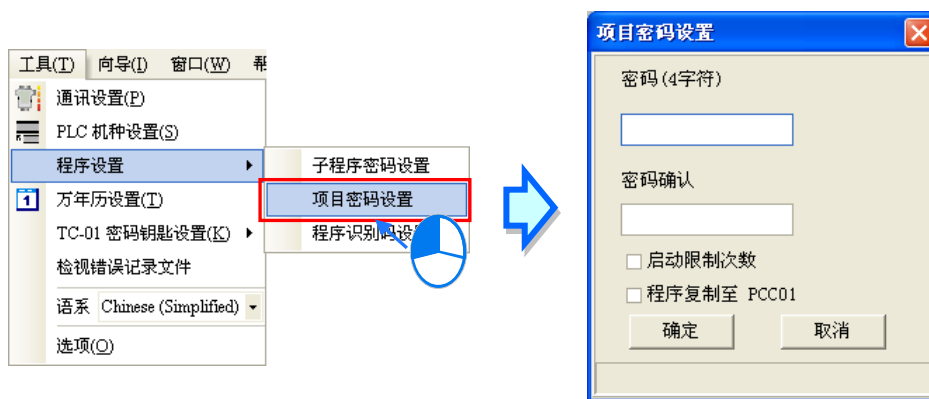
19.3 项目密码与 PLC 密码

「项目密码」的主要功能在于保护整体的项目程序，若用户开启某个设有「项目密码」的项目后，当使用者欲开启该项目中的 POU 时，系统便会要求输入正确的密码之后才允许开启，且「项目密码」可设定输入的限制次数，当用户输入密码的错误次数超过最大限制时，该项目便会自动关闭。

「PLC 密码」则是设定于主机当中而用来保护主机内部数据的一组密码，当要进行项目的上传或下载时，系统都会要求输入正确的密码之后才允许后续的操作。设定「PLC 密码」时可一并设定密码验证的限制次数，当用户输入密码的错误次数超过最大限制时，该主机便会被锁定而不再允许进行上传或下载的操作，此时唯有将 PLC 回复至出厂设定值之后，锁定的状态才可解除。

19.3.1 项目密码的设置与解除

功能工具栏中点选 **工具(I)** > **程序设置** > **项目密码设置**，之后便会出现「项目密码」的设定窗口，且由该窗口的状态便可得知目前开启的项目是否有启动「项目密码」的保护功能。当开启的是 DVP 机种的项目时，在「项目密码」的窗口中还会出现「程序复制至 PCC01」的功能项目。

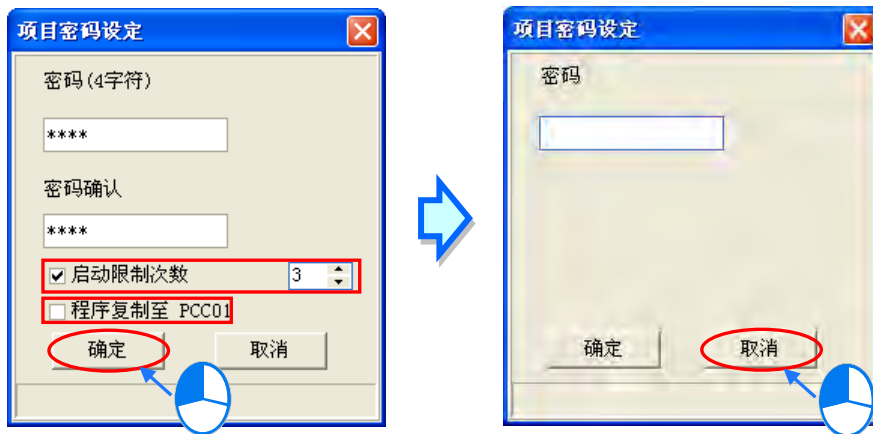


若目前的「项目密码」为关闭状态时，窗口中便会有两个设定密码的字段，而要启动「项目密码」的功能时，请先后于两个字段当中输入相同的密码，而输入的密码只可为英文字母、数字及空白，其中英文字母必须区分大小写；不同机种的密码字数限制可能不同，请依照窗口的提示进行设定。



若要设定密码输入的限制次数时，请先勾选「启动限制次数」之后，于其右侧的字段设定次数；而要启动「程序复制至 PCC01」的功能（详见下述）时，请直接勾选该项目即可。

完成设定后请按下「确定」键，若设定无误时，窗口便会自动切换为启动「项目密码」的状态，此时按下「取消」键便可关闭窗口。



当目前「项目密码」的功能为启动状态，则窗口中只会会有一个用来解除密码的字段，而要解除密码时，请直接于此字段输入正确的密码后按下「确定」键即可。同样的，若成功解除密码后，窗口便会自动切换为关闭「项目密码」的状态，此时按下「取消」键便可关闭窗口。

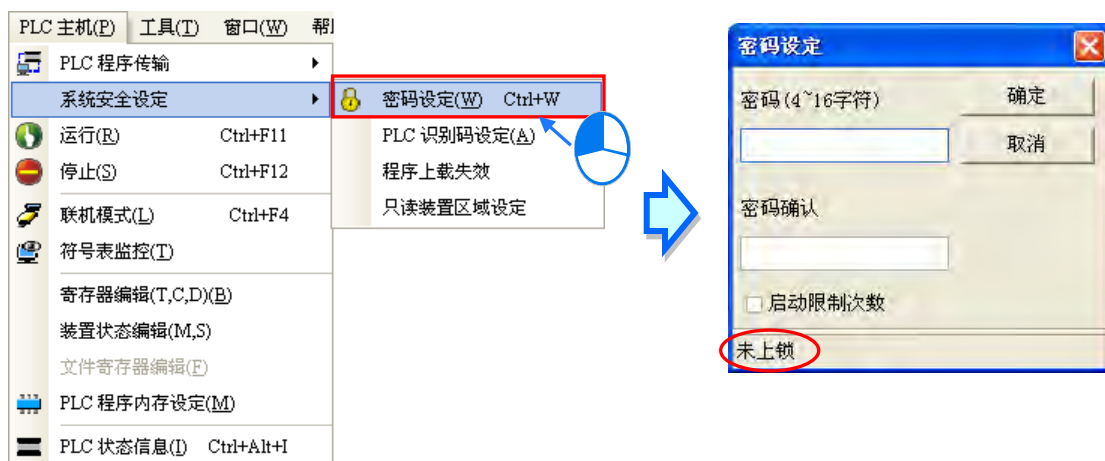


补充说明

PCC01 (产品全名为 **DVPPCC01**) 为 DVP 系列机种专用的数据备份卡，而在此节内容当中所提及的「程序复制至 PCC01」功能，其主要的功能在于下载项目至 PLC 主机后，系统将会一并将主机设置于某个状态，而在此状态下将 **PCC01** 链接至主机来进行备份时，系统便会将所下载的项目程序也一并备份至 **PCC01** 当中；不过在下载完毕后至 **PCC01** 链接至主机之前的期间，若主机有再接收到其它的通讯命令，如通过 ISPSOft 检视 PLC 信息或切换运行状态，此时所设定的「程序复制至 PCC01」的功能状态便会失效。此功能目前仅支持 DVP 系列的 ES2 机种，关于支持此功能的韧体版本与 **DVPPCC01** 的使用方式请参考相关使用手册或技术文件。

19.3.2 PLC 密码的设置与解除

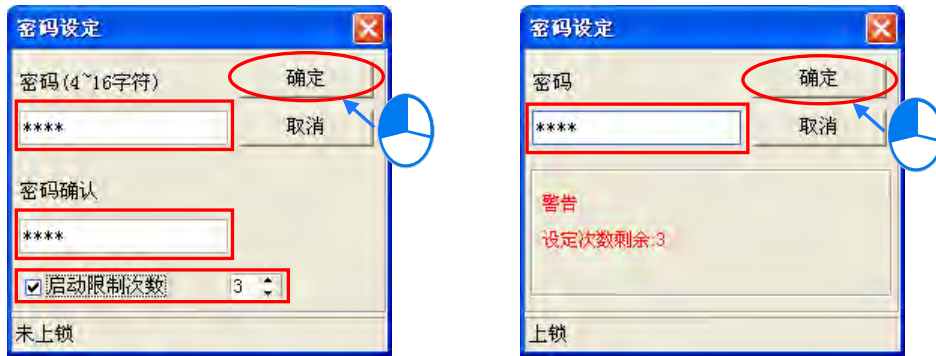
欲设定「PLC 密码」之前，请先确认 ISPSOft 已可与主机正常联机，于功能工具栏中点选 **PLC 主机 (P)** > **系统安全设定** > **密码设定 (W)**，之后 ISPSOft 便会主动联机至 PLC 并将主机的状态读取回来；当成功读取主机的信息之后，ISPSOft 便会自动开启密码的设定窗口，而由该窗口的下方便可得知主机目前的「PLC 密码」功能是否启动。



若主机目前的「PLC 密码」功能为关闭状态时，窗口中便会有两个设定密码的字段；若要设定密码时，请先后于两个字段中输入相同的密码；而若要设定密码输入的限制次数时，请先勾选「启动限制次数」之后，再于其右侧的字段设定次数，完成后按下「确定」键即可。

若目前「PLC 密码」的功能为启动状态，则窗口中只会会有一个用来解除密码的字段，而要解除密码时，请直接于此字段输入正确的密码后按下「确定」键即可。

设定的密码中仅能有英文字母、数字及空白，其中英文字母必须区分大小写；不同机种的密码字数限制可能不同，请依照窗口字段的提示进行设定。当设定完成后，窗口的状态便会自动切换，确认无误后按下「取消」键便可关闭窗口。



19.3.3 项目密码与 PLC 密码同步

ISPSOft 有提供一个便利的功能，可让用户在下载的过程中同步设定「项目密码」与「PLC 密码」。执行下载操作时，若在传输选项当中勾选「项目密码与 PLC 密码同步」的项目时，则下载完成之后，主机内部的「PLC 密码」将被改写为与「项目密码」一致。

若在下载时勾选「项目密码与 PLC 密码同步」的项目，但该项目本身未设定「项目密码」时，系统将会要求用户完成「项目密码」的设定；当使用者以开启旧档的方式开启一个项目，且该项目本身有设定「项目密码」时，下载选项中的「项目密码与 PLC 密码同步」项目便会强迫勾选。



19.4 POU 密码

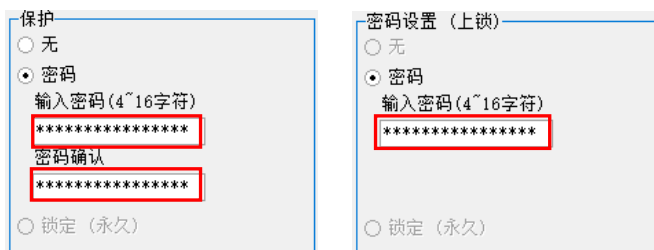
ISPSOft 允许用户各别为不同的程序或功能块 POU 设定一组专属的密码。当用户要开启有设定密码保护的 POU 时，系统便会要求输入正确的密码之后才允许开启。利用此功能便可对于包含特定程序代码或核心技术的数据多加一道保护机制。关于「POU 密码」的设定与解除方式如下。

请于项目管理区中，在欲设定密码的 POU 项目上按下鼠标右键，之后于选单中点选 属性 以开启 POU 的属性窗口，而于属性窗口的「密码设定」区域中便可知目前「POU 密码」的状态。



若目前的「POU 密码」功能为关闭状态时，区域中便会有两个设定密码的字段；当要设定密码时，请先后于两个字段中输入相同的密码，而输入的字符可为英文字母、数字或符号，其中英文字母必须区分大小写，完成后按下「确定」键即可。

若目前的「POU 密码」功能为启动状态，则窗口中只会有一个用来解除密码的字段，而要解除密码时，请直接于此字段输入正确的密码后按下「确定」键即可。

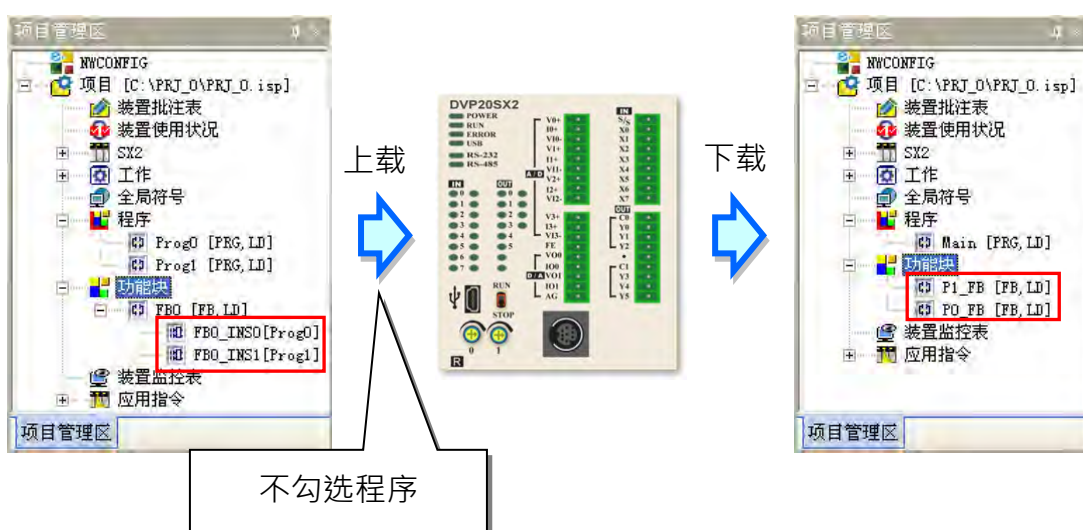


19.5 子程序密码

* 目前 DVPxxMC 机种不支持子程式密码。

19.5.1 子程序密码简介

当一个 DVP 机种的项目程序在编译完成之后，系统所产生的执行码可能会包含主程序、一般子程序与中断子程序几个部份，其中一般子程序的内容则是由原始码中的功能块 POU 编译而来；而若后续在进行下载时不勾选「程序」而仅下载「执行码」的部份，则上传完成后，执行码中的一般子程序便会再度被转换为 ISPSOft 中的功能块 POU，且因「POU 密码」只会保护原始码的 POU 程序，因此由执行码重新转换的 POU 便不受其保护，而使用者便可随意开启并对功能块内部的内容进行解析。



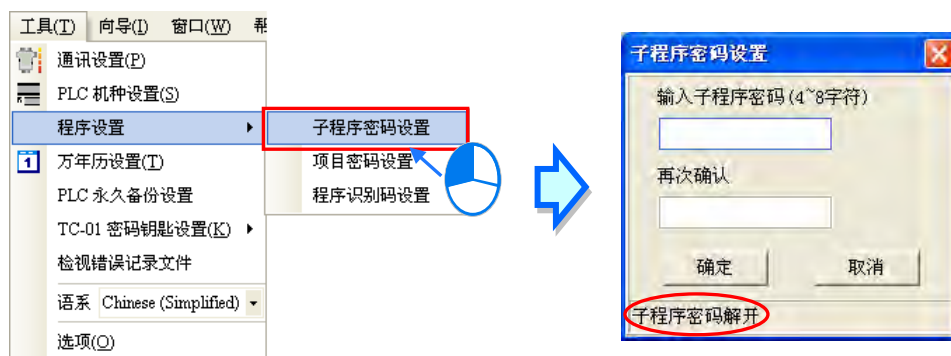
一般而言，开发者会将核心技术或较底层的程序包装成功能块，但若通过上述的方式取得执行码时，所有的技术便有可能会被解析出来，因此为了避免此种状况，使用者可在项目下载之前另外加上「子程序密码」的保护机制；若下载的程序有设定「子程序密码」且功能块 POU 有设定密码时，当执行码中的子程序被上传回来并转换为功能块之后，先前所设定「子程序密码」便会被转换为所有功能块 POU 的「POU 密码」，因此当用户要开启功能块时，系统便会要求用户输入正确的密码之后才允许开启。

19



19.5.2 子程序密码的设定与解除

功能工具栏中点选 **工具 (I)** > **程序设置** > **子程序密码设置**，之后便会开启「子程序密码」的设定窗口，而由该窗口的下方便可得知目前的「子程序密码」功能是否启动。



若目前「子程序密码」的功能为关闭状态时，窗口中便会有两个设定密码的字段；当要设定密码时，请先后于两个字段中输入相同的密码，完成后按下「**确定**」键即可。

若目前「子程序密码」的功能为启动状态，则窗口中只会有一个用来解除密码的字段，而要解除密码时，请直接于此字段输入正确的密码后按下「**确定**」键即可。

设定的密码中仅能有英文字母、数字及空白，其中英文字母必须区分大小写；当设定完成后，窗口的状态便会自动切换，确认无误后按下「**取消**」键便可关闭窗口。

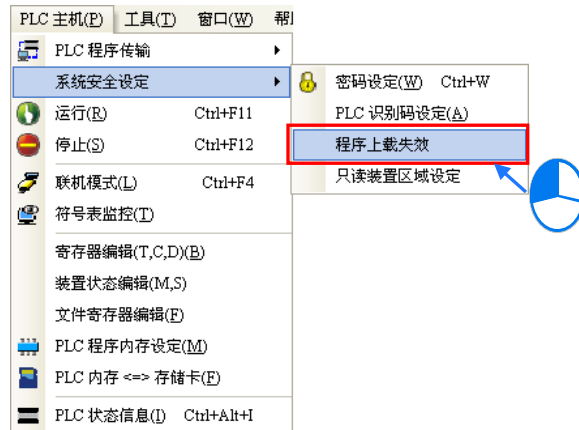


19.6 其它的密码与数据保护功能

19.6.1 程序上传失效

此功能主要在于限制 PLC 的上载操作。当启动此功能后，主机的上传操作便会被禁止，而后续唯有将主机恢复至出厂默认值才可解除。若主机有设定「**PLC 密码**」时，设定前必须先将其解除，且设定完成后，因已不再需要「**PLC 密码**」的功能，因此用户将无法再设定「**PLC 密码**」。

欲设定此功能时，请于功能工具栏中点选 **PLC 主机 (P)** > **系统安全设定** > **程序上传失效** 即可。且因设定之后，主机内部的数据将无法上传，因此请务必确认是否须将主机的数据事先进行备份。

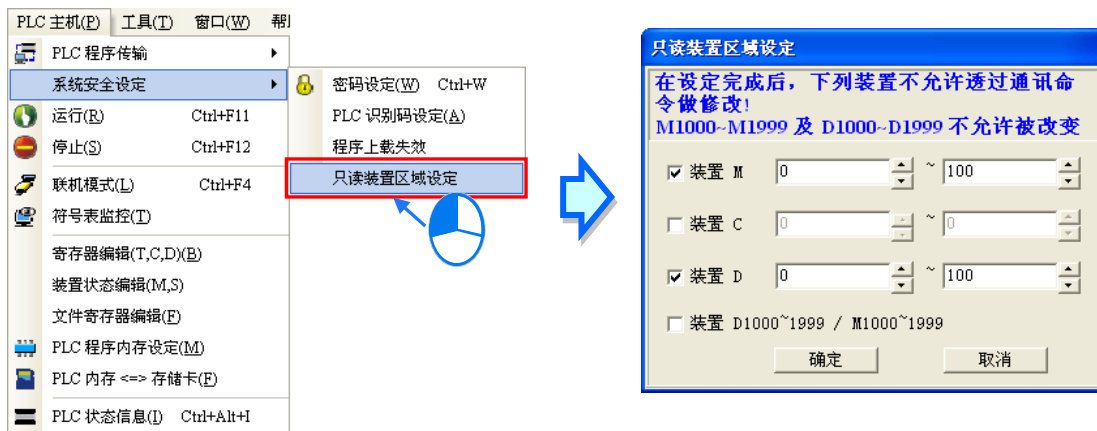


*.使用此功能前，请确认欲设定的主机机种与其初体版本是否支持，相关细节请自行参考各机种的操作手册或技术文件。

19.6.2 只读装置区域设定

此功能在于避免 PLC 内部的装置内容被随意变更。当启动此功能后，设定范围内的装置将无法再通过通讯联机的操作来变更内容值，而唯有将主机恢复至出厂默认值才可解除此状态。

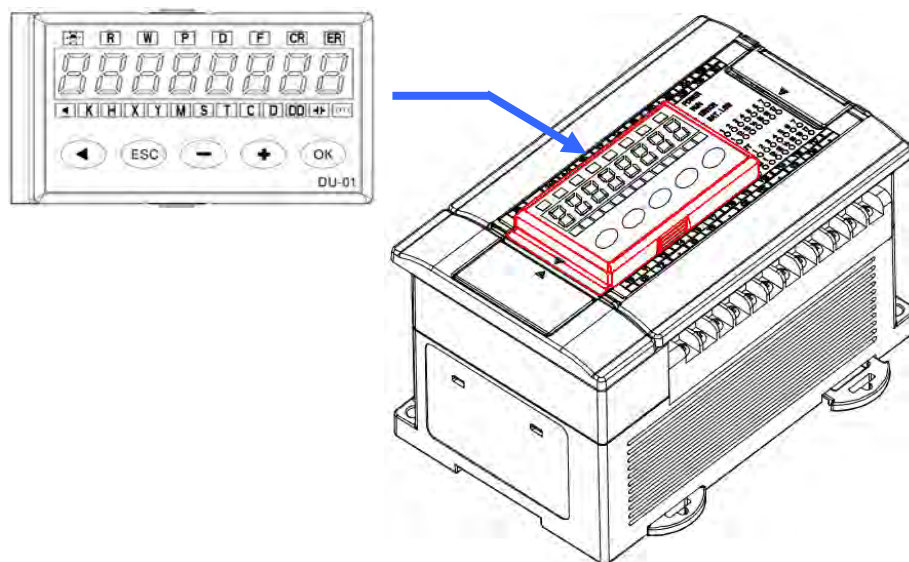
欲设定此功能时，请于功能工具栏中点选 **PLC 主机 (P)** > **系统安全设定** > **只读装置区域设定**，之后于设定窗口中勾选欲设置此功能的装置类别，并设定装置范围，完成后按下「**确定**」键即可。



*.设定前请确认欲设定的主机机种与其初体版本是否支持此功能，相关细节请自行参考各机种的操作手册或技术文件。

19.6.3 TC-01 密码钥匙设定

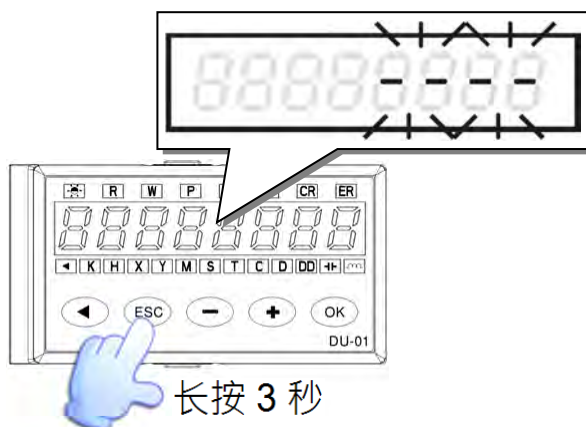
此功能主要是针对数字设定显示器 **DVPDU-01** 所提供的辅助工具。下列即为 **DVPDU-01** 的图片，其主要的应用是提供 DVP 系列主机一个简易的操作接口。



DVPDU-01 的功能可分为装置监控用的 **TS-01** 模式与数据读写用的 **TC-01** 模式；而在 **TC-01** 模式的应用中，为了可对有设定「**PLC 密码**」的主机进行数据读写，事先便可在 **DVPDU-01** 当中设定一组与「**PLC 密码**」相同的「**TC-01 密码钥匙**」，之后当进行数据读写时，系统便会在确认 **DVPDU-01** 中的「**TC-01 密码钥匙**」与主机的「**PLC 密码**」一致后才会开始进行读写；当主机未设定「**PLC 密码**」时，在数据读写的同时系统亦会自动将「**TC-01 密码钥匙**」设为该主机的「**PLC 密码**」。

下列则为设定与清除「**TC-01 密码钥匙**」的操作描述，而设定前请先确认 **DVPDU-01** 处于等待的状态，若目前的状态为错误或是处于其它的操作过程中，请先完成操作或将该状态解除。更进一步的说明请参考 **DVPDU-01** 的使用手册或相关技术文件。

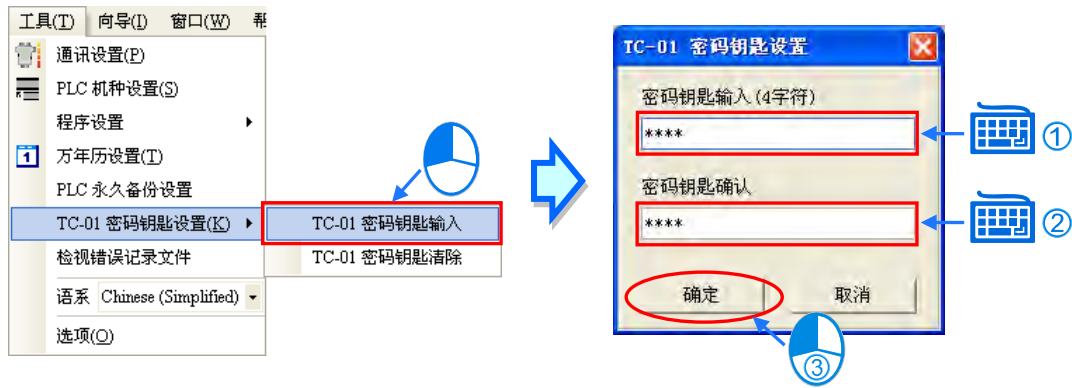
(1) 长按 **DVPDU-01** 面板的【**ESC**】键 3 秒，之后显示器的部份便会进入密码设定画面。



(2) 于 ISPSoft 中设定或清除「TC-01 密码钥匙」。

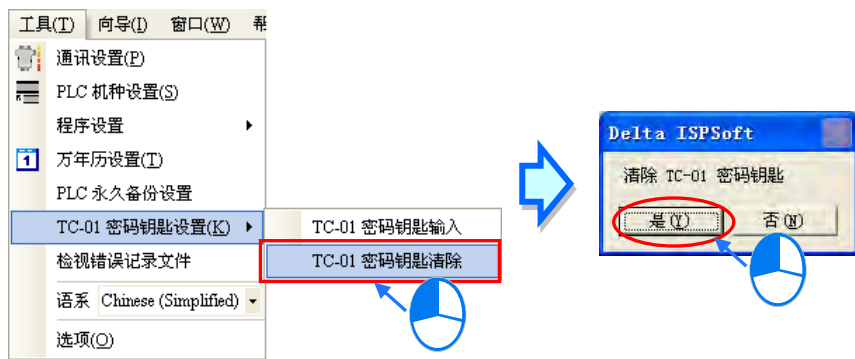
● 设定 TC-01 密码钥匙

于功能工具栏点选 **工具 (I)** > **TC-01 密码钥匙设置 (K)** > **TC-01 密码钥匙输入**，之后于设定窗口中先后输入两次相同的密码，完成后按下「**确定**」键即可。

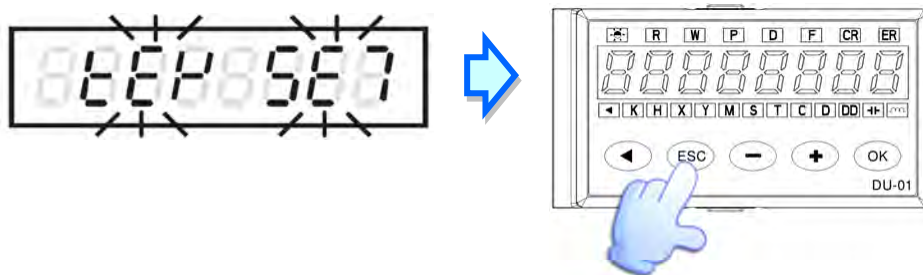


● 清除 TC-01 密码钥匙

于功能工具栏中点选 **工具 (I)** > **TC-01 密码钥匙设置 (K)** > **TC-01 密码钥匙清除** 即可。



(3) 设定完成后，显示器的画面会显示「KEY SET」，接着请再按下面板的【ESC】键离开即可。



第20章 网络规划与数据交换

目录

20.1 网络规划工具 - NWCONFIG	20-3
20.1.1 NWCONFIG 简介	20-3
20.1.2 相关基础介绍	20-4
20.1.3 NWCONFIG 的通讯设定	20-6
20.1.3.1 NWCONFIG 中的联机机制	20-6
20.1.3.2 设定通讯参数	20-8
20.1.4 工作流程	20-9
20.2 网络架构的部署	20-13
20.2.1 部署节点	20-13
20.2.2 连结网络	20-17
20.2.3 调整或删除已部署的装置及网络	20-21
20.2.4 节点与网络属性的设定	20-24
20.2.5 装置或网络的隐藏/显示	20-28
20.2.6 合法的网络架构	20-31
20.2.7 下载绕送路径表 (Routing Table)	20-34
20.2.8 绕送路径测试	20-36
20.3 PLC Link 的建构	20-38
20.3.1 打开 PLC Link 的设定窗口	20-38
20.3.2 选择主站装置 (步骤 1)	20-40
20.3.3 设定通讯参数 (步骤 2)	20-41
20.3.4 设定数据交换表 (步骤 3)	20-43
20.3.4.1 数据交换表简介	20-43
20.3.4.2 设定区块	20-45
20.3.4.3 数据交换表的机种同步性	20-47
20.3.4.4 数据交换表的管理操作	20-49
20.3.5 PLC Link 的监控功能	20-50
20.3.6 设定 PLC Link 的注意事项	20-54
20.4 Ether Link 的建构	20-55

20.4.1	Ether Link 简介	20-55
20.4.2	开启 Ether Link 的设置窗口	20-56
20.4.3	数据交换表的建立与管理	20-58
20.4.4	装置清单与图标区的操作	20-61
20.4.5	Ether Link 的运作模式	20-63
20.4.6	下载 Ether Link 设定	20-65
20.4.7	上传 Ether Link 设定	20-67
20.4.8	删除异步装置	20-69
20.4.9	启动/关闭在线监控模式	20-70
20.4.9.1	启动监控	20-70
20.4.9.2	监控状态显示	20-72
20.4.9.3	关闭监控	20-72
20.4.10	在线启动/停止 Ether Link (标志模式)	20-74
20.4.10.1	启动 Ether Link	20-74
20.4.10.2	停止 Ether Link	20-75
20.4.11	监控表与错误记录	20-77
20.5	NWCONFIG 的管理与应用	20-78
20.5.1	保存与打印	20-78
20.5.2	整体下载	20-79
20.5.2.1	下载参数说明	20-79
20.5.2.2	下载操作说明	20-79
20.5.3	ISPSOFT 中的绕送应用	20-81

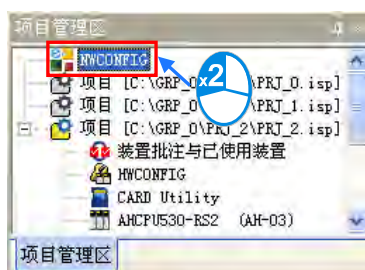
20.1 网络规划工具 - NWCONFIG

20.1.1 NWCONFIG 简介

NWCONFIG 为 ISPSOft 所提供的网络规划工具，功能在于规划整个项目的网络架构，并藉此建立常态性的数据交换机制，而其主要负责的工作如下，详细说明将于本章后续的内容逐一介绍。

- (a) 规划整个项目的网络部署，并建立数据的传送路径。
- (b) 规划 RS485 的数据交换机制 - **PLC Link**。
- (c) 规划 Ethernet 的数据交换机制 - **Ether Link**。

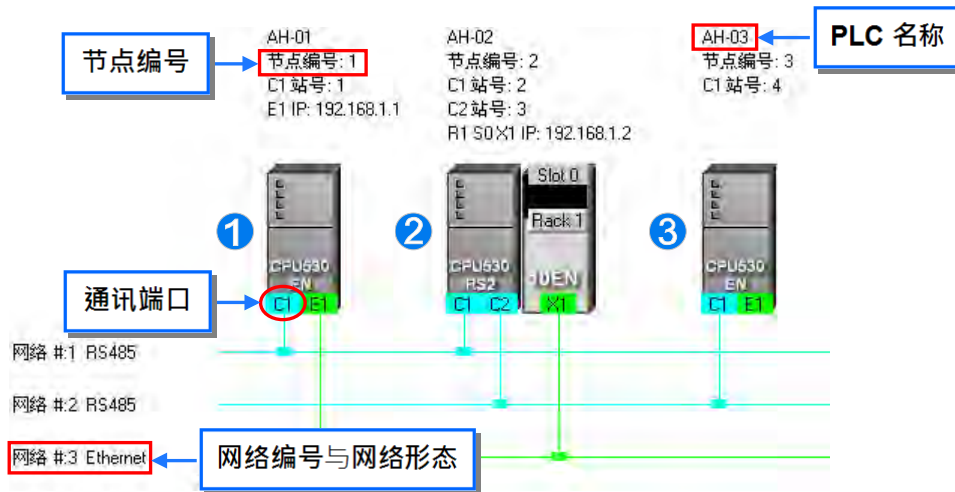
NWCONFIG 的定位在于规划整个项目的网络架构，因此其所在的位置会在整个项目目录的最上层，而要开启 **NWCONFIG** 时，请直接于项目管理区的「**NWCONFIG**」上双击鼠标左键即可。



- ① 图标工作区：为主要的工作区，用户可于此规划整个网络架构。
- ② 装置列表：此区以目录管理的方式呈现所有可使用的装置资源。
- ③ 讯息窗口：此处会显示操作过程中的相关资讯。

20.1.2 相关基础介绍

在开始进行网络规划之前，针对操作过程所需具备的一些基础认知，我们将在本节当中先加以介绍。



● 装置与网络

装置为 **NWCONFIG** 中最基本的组件，其所指的可以是 PLC 主机、扩充模块，或是用户自行定义的设备；而此处的**网络 (NETWORK)** 则是指连接各个装置的逻辑通道，且每个网络都必须指定一个唯一的网络编号，而类型则有 RS-485 及 Ethernet 两种。另外，因各装置用以连接至网络的端点即是该装置的通讯口，因此当一个装置具有两个以上的通讯口时，该装置便可同时连接至不同编号的网络。关于 NWCONFIG 中的通讯口标示将于第 20.2.2 节当中另行介绍。

● PLC 名称

如上图的「AH-01」、「AH-02」与「AH-03」。当 PLC 机种为 **AH5x0** 系列时，**PLC 名称** 则必须依据 HWCONFIG 中的设定，而主要的功能是在通讯时做为主机的识别，详细说明请参考第 3.1.4 节；但以其它非 **AH5x0** 系列的机种而言，**PLC 名称** 则如同主机的批注一般，本身并没有任何操作上的意义。

● 节点 (Node) 与节点编号

节点 (Node) 指的是在一个网络中可以独立运作的基本单位，如上图的 ① ~ ③，其中节点 ② 虽包含主机与网络模块两个装置，但因模块本身无法独立运作，因此仍须将两者合并而视为一个节点。**AH5x0** 系列机种有支持「绕送功能」，即跨主机的联机操作，例如可通过上图的节点 ① 来对节点 ③ 进行监控；而使用此功能的前提则是须先规划数据的传送路径，并对路径中的每个数据转送点都赋予一个**节点编号**；当在进行规划时，仅有 **AH5x0** 系列的机种可设定**节点编号**，且同一编号的网络上也不可存在相同**节点编号**的节点；而最后完成路径规划并下载至各个主机之后，每个节点的主机便会拥有属于自己的绕送表 (Routing Table) 以作为转送数据的依据。

● 站号

用来识别 RS485 网络上的工作站，站号依据通讯口进行配置，基本上一个通讯口代表一个工作站，且同一个网络编号上的工作站号不可重复。因此当一个节点本身包含数个串行端口 (COM Port) 时，只要有连上网络的串行端口 (COM Port) 都必须为其指定一个站号。

● IP 地址与 DHCP 模式

IP 地址是通讯口在网络中的地址，基本上一个通讯口便可代表一个网络端点，因此当一个网络中包含数个通讯口时，只要有连上网络的通讯口都必须指定一个 IP 地址，且 IP 地址不能重复。

DHCP 则为动态分配 IP 地址的机制，网络中必须存在 DHCP 分配的主机，当某个通讯口使用 DHCP 模式时，该通讯口便可自动从 DHCP 主机取得一个 IP 地址。

● 子网掩码

用来划分子网域的参数值。

● PLC Link

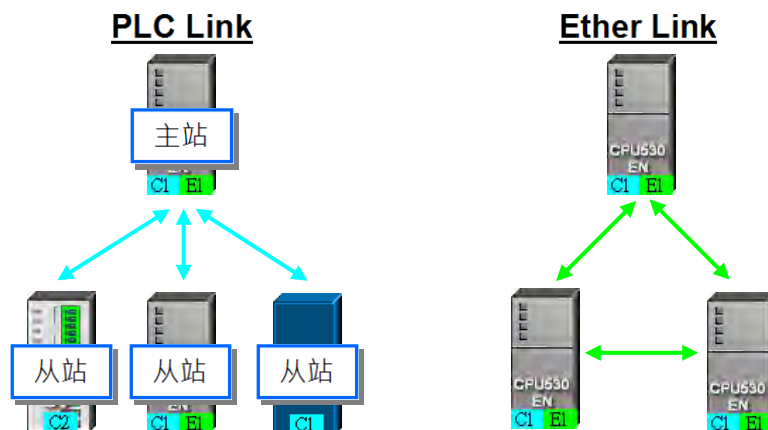
通过 RS485 的联机来进行数据交换的网络机制。当某个 RS485 的网络上存在多个节点时，我们便可于该网络中规划一个数据交换的机制；当设置参数被下载至设为主站的主机且开始运行后，通过特殊继电器与特殊寄存器的操作，系统便可进行数据交换的动作。

PLC Link 是一种主从架构，一个 RS485 的网络上只能有一个主站，而其它从站则必须被动的接收主站轮流发出的读写命令，因此从站与从站之间无法直接交换数据，而必须通过主站的转发。

● Ether Link

通过 Ethernet 的联机来进行数据交换的网络机制。当某个 Ethernet 的网络上存在多个节点时，我们便可于该网络中规划一个数据交换的机制，并指定其运作的方式；当 Ether Link 的设置参数被下载至各个主机且开始运行后，系统便会依据设定的启动模式而自动地进行数据交换的动作。

有别于 PLC Link 的主从架构，Ether Link 的操作概念则是允许所有节点对其他的节点发出索取数据的需求，但只能被动的在接收到需求之后才可将数据传送给提出需求的节点；没有强迫写入的动作，因此相对的有较高的安全性，且通过 TCP/IP 的协议，系统会自动管理报文传送的机制，相较于 PLC Link 的轮流读写则会有较高的执行效率。



*.关于 RS485 与以太网 (Ethernet) 的相关概念，请自行参考其它书籍或技术文档。

20.1.3 NWCONFIG 的通讯设定

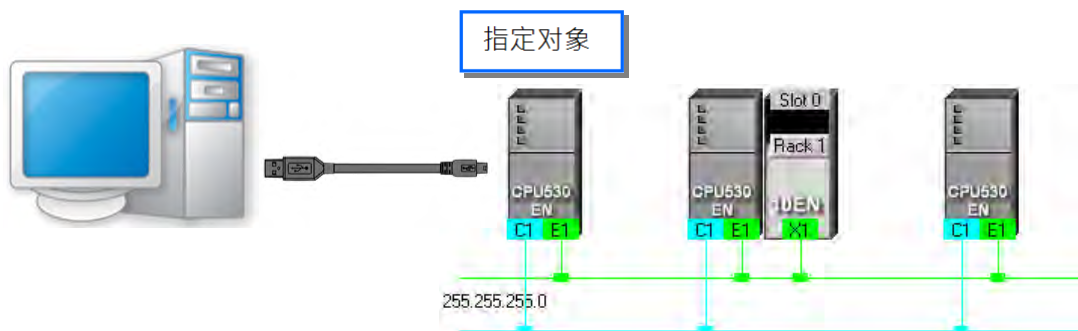
NWCONFIG 的主要工作是进行网络功能的组态与规划，而在过程中无可避免的须要对所规划的网络节点进行参数上下载或是监控的动作，且此处所规划的网络节点，其中可能包含了 ISPSOft 项目机种以外的装置，因此用户必须另外在 NWCONFIG 当中设定通讯参数，而无法直接沿用 ISPSOft 当中的设定。在开始说明如何设数之前，以下会先针对 NWCONFIG 中的通讯机制做简单的介绍，以便在进行设定时可选择适当的通讯参数。

20.1.3.1 NWCONFIG 中的联机机制

当在 NWCONFIG 中进行上下载或在线监控时，其操作的对象将可能是网络规划中的任何一个节点，且这些操作在 NWCONFIG 中也可能会提供单点与多点两种操作模式，因此进行操作前请务必依据操作的模式来考虑联机的方式，并依此选择适当的通讯参数。

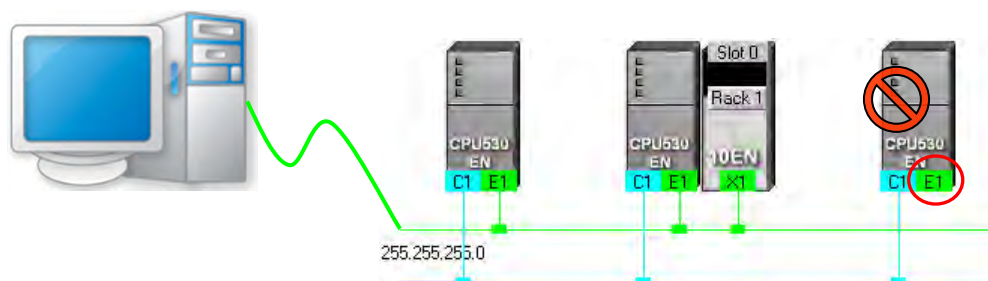
● 单点模式

仅针对指定的装置进行上下载或在线监控，因此操作前请务必确认所指定的装置与实际联机至计算机的装置是否一致，否则便可能发生误操作。

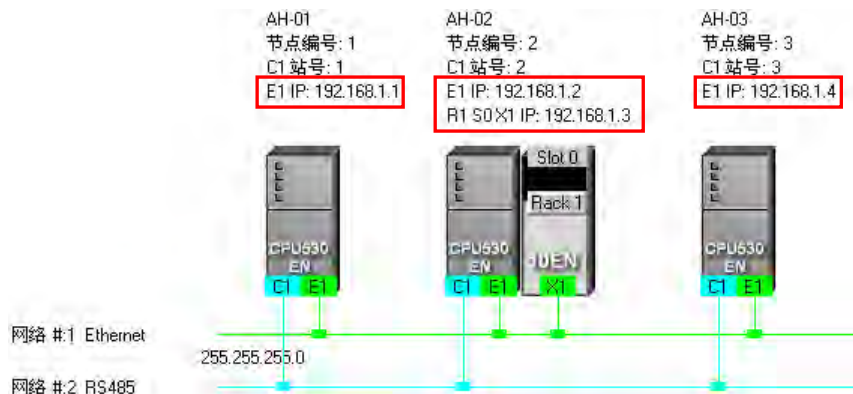


● 多点模式

同时对整个网络上的节点进行操作，但进行多点模式的前提是要进行操作的所有装置都必须连接至 Ethernet 的网络上，且每个装置都必须设定好正确的 IP 地址，而 COMMGR Driver 也必须选择为 Ethernet 的联机型式。

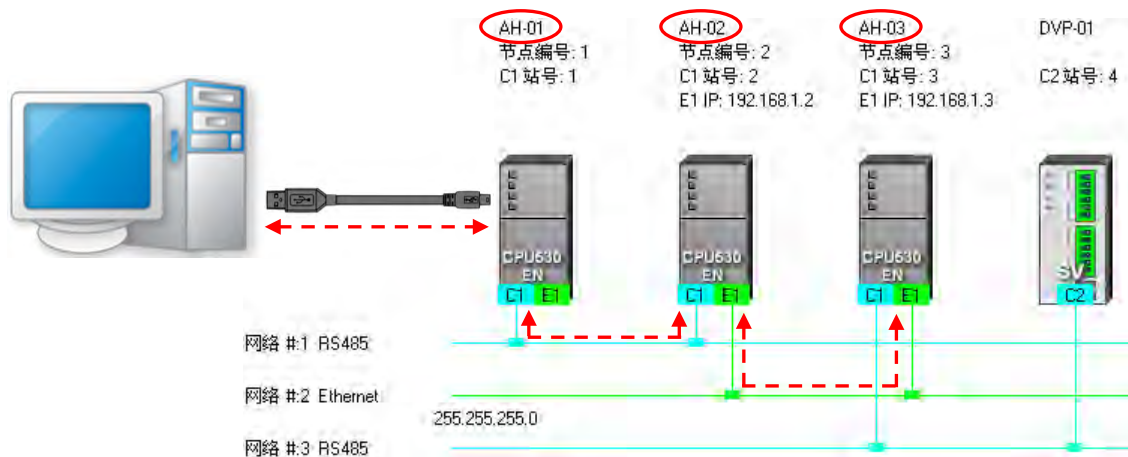


不论是单点或多点模式，若选择的 COMMGR Driver 为 Ethernet 的型式，而之后在进行上下载或监控操作时，系统便会依据用户在 NWCONFIG 中所规划的 IP 地址来进行通讯，因此进行联机时请务必确认各装置内部的实际 IP 地址与 NWCONFIG 中的设置一致，且实际架设的网络也必须与规划的配置一致或兼容，否则便可能在操作得过程中发生误动作，或收发到非预期的数据。



除了上述所介绍的方式之外，NWCONFIG 也可通过**报文绕送模式**的机制来进行联机操作。

报文绕送模式是 AH5x0 系列机种所提供的功能，主要的概念是通过转送的机制来对非直接接续的装置进行联机；如下图，当实际与计算机联机的装置为 AH-01，但操作上却须要与 AH-03 进行联机时，此时便可启动**报文绕送模式**，并指定 AH-01 为**初始站台**，而之后当计算机送出通讯命令时，该命令便可经由 AH-01 → AH-02 → AH-03 的路径送达要联机的装置 AH-03。




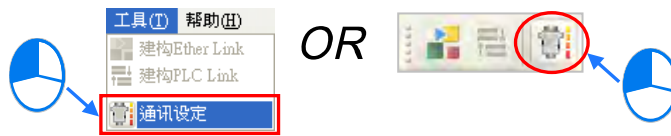
下列则为使用**报文绕送模式**的相关注意事项。

- 必须先于 NWCONFIG 完成所有网络的规划，并将产生的**绕送路径表 (Routing Table)** 下载至路径中的每个节点。详细内容将于第 20.2 节当中另行介绍。
- 只有 AH5x0 系列的机种可提供绕送的功能，而 DVP 系列机种或其它装置则不可做为转发报文中继站，不过还是可以当作绕送路径的终点站，也即要进行通讯的对象。

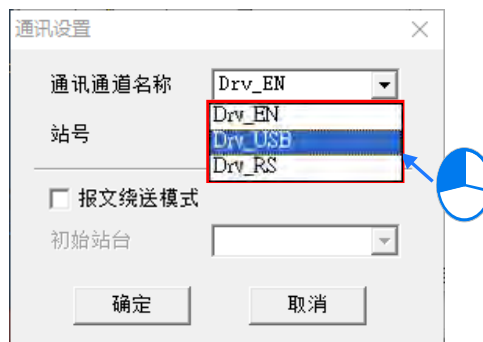
20.1.3.2 设定通讯参数

下列为在 NWCONFIG 中设定通讯参数的步骤，请按照步骤操作。

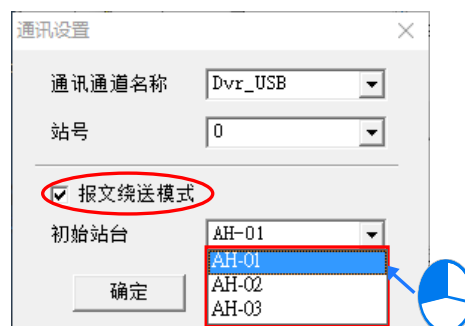
- (1) 正确启动**通讯管理员 - COMMGR**，且于 COMMGR 中事先建立好搭配联机所需的 Driver 并确认状态为「OK」。
- (2) 当使用**单点模式**进行操作时，请先确认要进行操作的对象与实际联机的装置是否一致；而若要进行**多点模式**的操作时，请确认所有参与操作的装置都必须连接至 Ethernet 的网络上，且须与 NWCONFIG 中的规划一致。
- (3) 于 NWCONFIG 的功能工具栏中点选 **工具 (I) > 通讯设定**，或点选图标工具栏的  图标，接着在画面中便会出现通讯设定的窗口；请依下述说明进行设定，完成后按下「**确定**」即可。



- (4) 于设定窗口中的「**通讯通道名称**」字段下拉选择要使用的 COMMGR Driver。



- (5) 先勾选「**报文绕送模式**」，接着于下方的「**初始站台**」处下拉选择与计算机实际连接的装置。



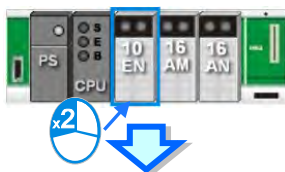
20.1.4 工作流程

因网络的规划会牵涉到一个系统的整体运作，因此它的工作流程必须是循序渐进的，而在本节的内容中，针对完成一个网络规划所需的工作流程会先做大略的介绍，至于细节操作的部份则将于后续的节日当中逐一说明；当然，此处所介绍的方式，仅是在一般情况下可有效率完成工作的使用建议，其方式不尽然可应用所有的案例，用户仍可依据实际情况或个人习惯来加以调整。

- (1) 在通过 ISPSOft 架构系统之前，请事先完成整个网络系统的规划设计，内容必须包括网络中各节点所采用的 PLC 机种或装置种类、PLC 主机是否需搭配通讯模块、每个节点之间如何接续、进行接续的通讯端口所规划的 IP 地址或 RS485 站号、RS485 的通讯参数等；除此之外，可能还须事先规划预期进行数据交换的设备与数据交换的区块，当然这个部分可能会牵涉到 PLC 程序的设计；在完成这些规划之后，接着便可在 ISPSOft 中逐一建构起整个网络的架构。
- (2) ISPSOft 中建立项目，如果在系统中有两部以上的台达 PLC 时，我们会建议直接将其建立为群组项目，详细操作方式请参考第 2.2 节的相关说明。



- (3) 针对 AH5x0 机种，请逐一开启群组中每个单机项目的 HWCONFIG 工具，并完成每个机种的硬件规划；内容包括模块配置、通讯模块参数设定以及 CPU 参数的 PLC 标签名称、通讯口设定、以太网基本设定...等；详细操作方式请参考第 3 章的相关说明。



编辑区			
概况			
- AH10EN-5A			
网络参数		参数名称	地址
网络参数		操作模式	单一 IP 地址 (Host 1 and Switching)
功能列表		Host 1(X1) 模式	静态 IP
IP 过滤参数		Host 1(X1) IP 地址	192.168.0.5
IO 对映参数		Host 1(X1) 子网掩码	255.255.255.0
		Host 1(X1) 网关器	192.168.0.1



编辑区

概况 数据交换

- AHCPU530-EN

- 系统信息

名称

名称	参数名称	设定值
I	名称	AH-02
	注释	



编辑区

概况 数据交换

- AHCPU530-EN

- 系统信息

名称

系统

停电保持装置范围

通讯端口

以太网 - 基本设置

+ 以太网 - 进阶设置

通讯端口

参数名称	设定值
传输方式	RS-232
数据长度	7
奇偶校验	偶
停止位	1
波特率	9600
传输模式	ASCII
通讯站号	1
传输错误自动询问次数	3
通讯逾时时间	3000



编辑区

概况 数据交换

- AHCPU530-EN

- 系统信息

名称

系统

停电保持装置范围

通讯端口

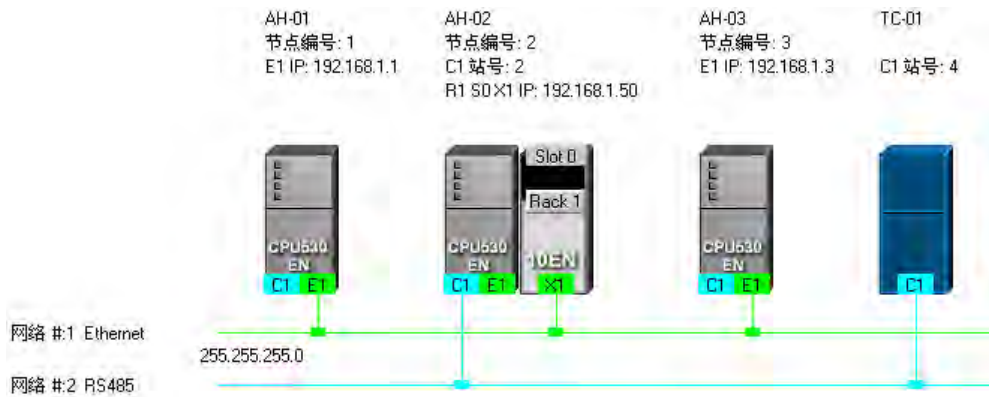
以太网 - 基本设置

+ 以太网 - 进阶设置

以太网 - 基本设置

参数名称	设定值
IP 地址模式	静态
IP 地址	192.168.1.1
子网掩码	255.255.255.0
网关地址	192.168.1.1
联机保持时间	60

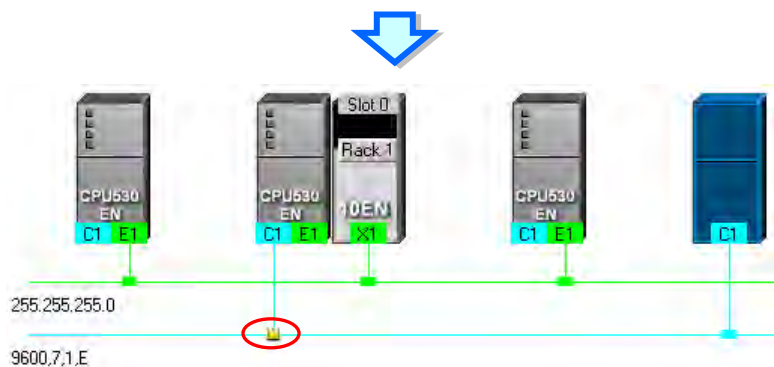
(4) 于 NWCONFIG 中完成网络部署。



(5) 建立 PLC Link 或 Ether Link 的数据交换机制，而此两者之间是独立运作的，因此在规划时并没有先后顺序的差别；规划时须注意所有的数据读写区块要避免发生地址重迭，否则在实际运作时便可能产生错乱，因此建议在规划之前能够依据程序的设计而事先定义数据交换的地址区块。下图为 PLC Link 的数据交换表，且在完成设定之后，网络规划的画面上会加上主站的标示。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置型态
1	4	读	D3000~D3000	<=	16#1000~16#1000	1	启动	MODBUS Device
		写	D3001~D3001	=>	16#1001~16#1001	1		
2	0	读	D100	<=	D4096	0	关闭	Unknown
		写	D100	=>	D4096	0		
3	0	读	D200	<=	D4096	0	关闭	Unknown
		写	D200	=>	D4096	0		
4	0	读	D300	<=	D4096	0	关闭	Unknown
		写	D300	=>	D4096	0		
5	0	读	D400	<=	D4096	0	关闭	Unknown
		写	D400	=>	D4096	0		

导出 重置 检查设定 上传 下载 下载并监控 完成



下图则为 Ether Link 的数据交换表。



- (6) 依序将项目程序、HWCONFIG 的硬件规划参数与 NWCONFIG 的网络规划参数一一下载至每个主机当中；其中若网络规划的内容包含 DVP 机种或 AH5x0 机种以外的装置，请自行参照各装置的使用说明逐一完成其通讯参数或相关功能的设定。

下图则为 NWCONFIG 的下载选项窗口，其中依据各节点的规划不同，可下载的选项也会不同。



20.2 网络架构的部署

20.2.1 部署节点

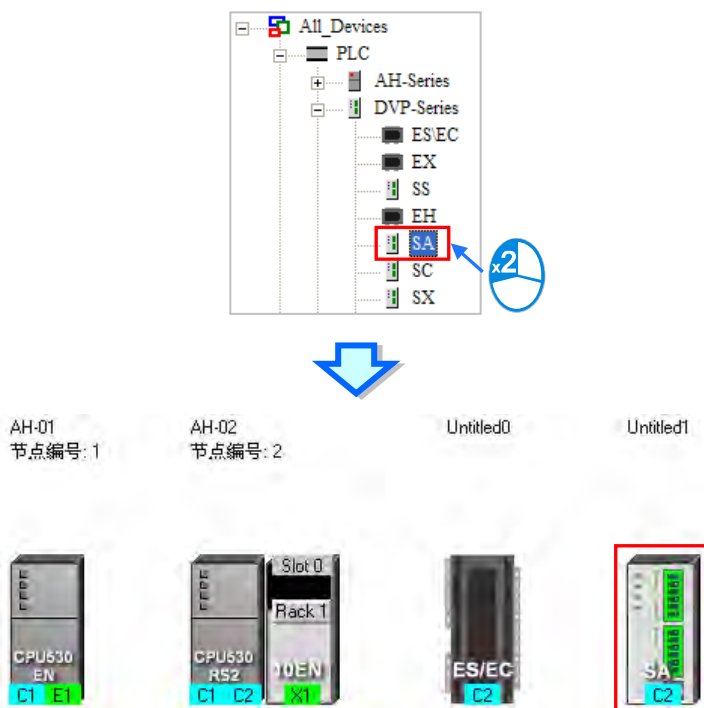
当首次开启某个项目中的 NWCONFIG 时，系统会依据项目中所建立的机种，不分 AH5x0 或 DVP 系列，所有建立的项目机种都会自动被部署于工作区中，且 AH5x0 机种还会依据 HWCONFIG 的设定而将对应的模块与节点参数一并带出来，而这些节点于后续的操作中也将不允许任意变更或删除。



除了依据项目而自动带出的节点之外，用户仍可另外加入其它的 PLC 主机或装置，而加入新装置的方式共有下列两种。

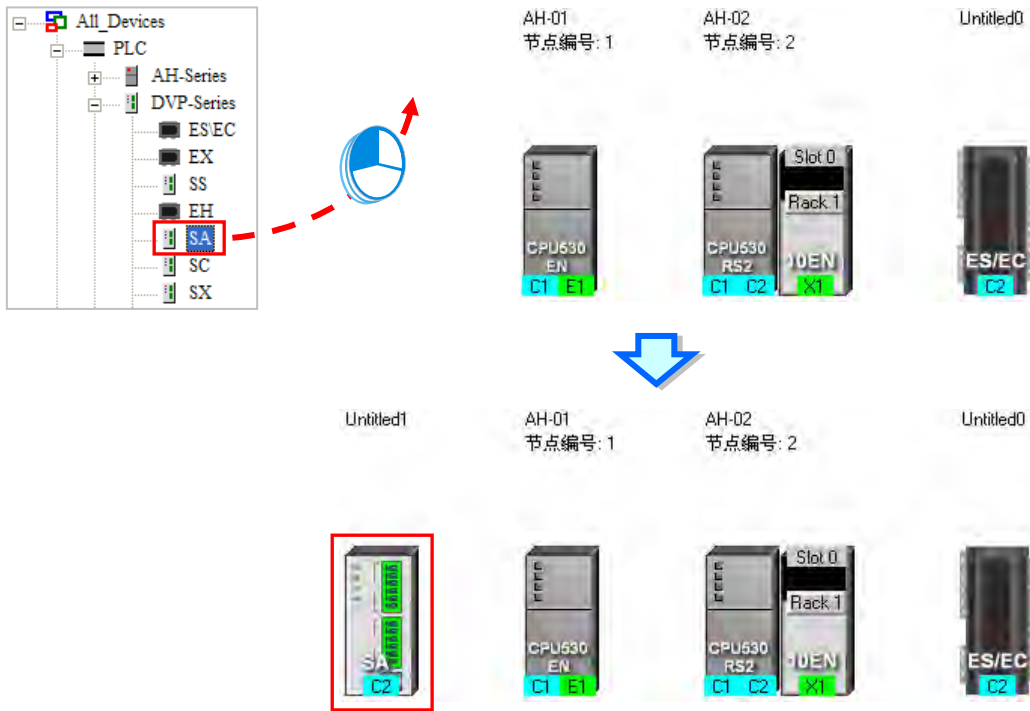
- 方法一

左侧的装置列表中要选择要新增的主机或装置型号后，该对象上双击鼠标左键，之后该主机的图标便会被排列在所有装置的右侧。

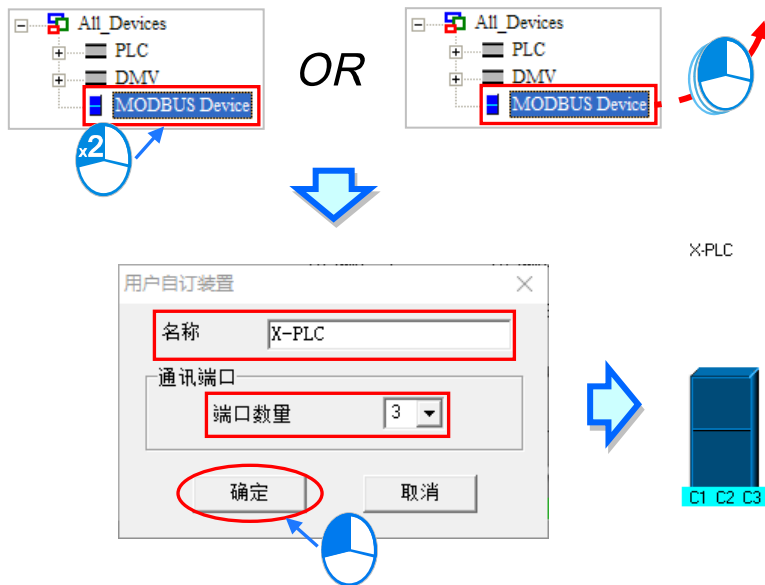


● 方法二

左侧的**装置列表**中点选要新增的主机或装置型号后，直接将其拖曳至要放置的位置即可。



在 NWCONFIG 中，用户也可由自行定义的方式加入标准 MODBUS 的装置；操作时请先于**装置列表**中点选「**MODBUS Device**」的图标，接着依据下图两种方式将其加入工作区中；弹出用户自定义装置窗口时于「**名称**」字段输入该装置的识别名称，接着于「**端口数量**」下拉，选择此装置的端口数量，至多可设定 3 个，完成后按下「**确定**」即可。

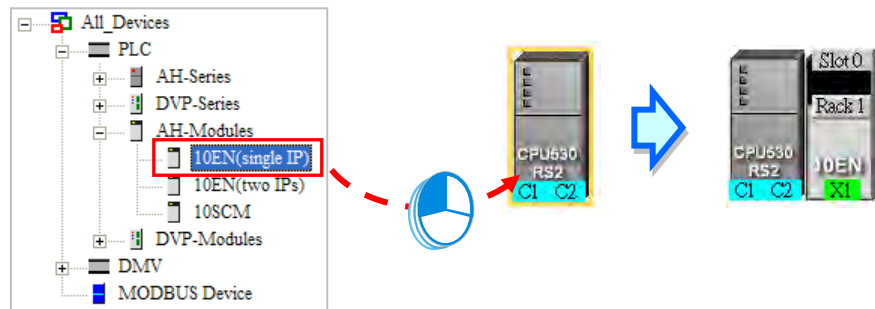



依据项目内容而自动带出来的第一台 AH5x0 系列主机，其模块配置是与 HWCONFIG 同步，所以将无法于 NWCONFIG 当中为其加上模块，若要改变其配置必须到 HWCONFIG 设定。

要加入通讯模块时也可通过下列两种方式。

● 方法一

于左侧的**装置列表**中点选模块型号后，直接将其拖曳至要附挂该模块的 PLC 主机即可。

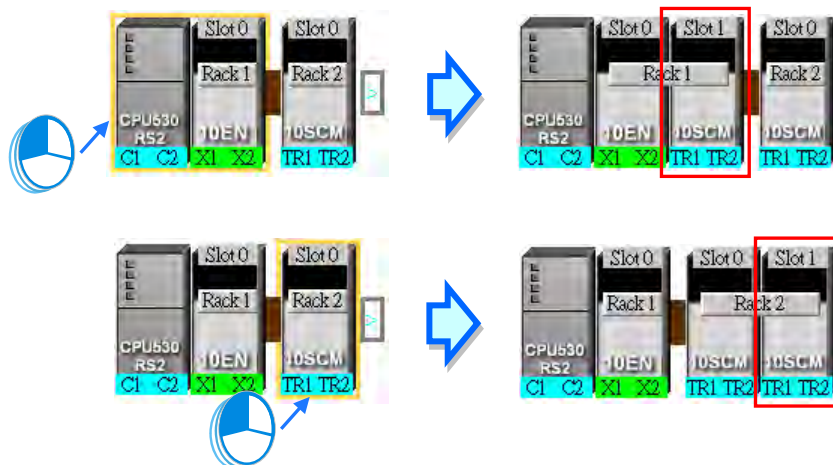



但若该主机本身不支持所选择的模块时，鼠标便会变为  的图标。

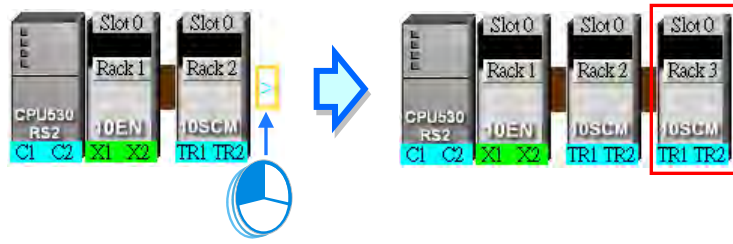


在 AH5x0 系列的机种中，10EN 模块仅可放置于主背板上，且至多可放置 8 个，而 SCM 模块则可以附挂在延伸背板上，且数量的限制会依据每个背板的插槽数量而定。

当通过拖曳的方式来加入 SCM 模块时，用户还可决定将其拖曳至哪个背板上放置。

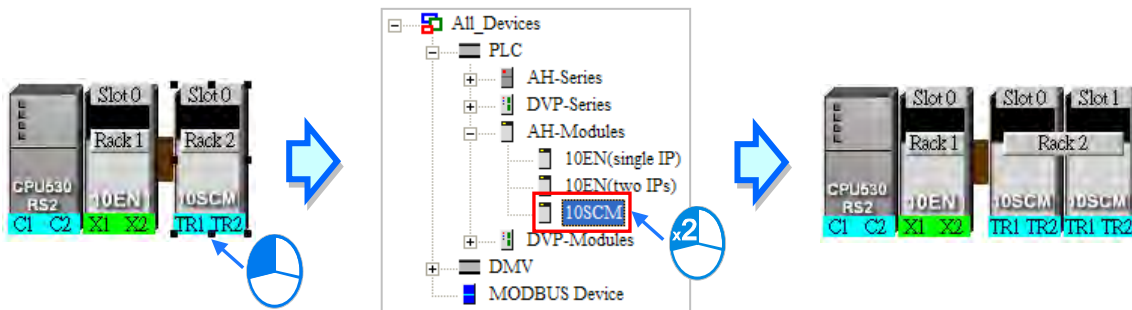


当要将 SCM 模块放置在其它的背板上时，请将其拖曳至该节点右侧的  图标处放开，此时于该节点的图标上便会自动新增一个延伸背板。



● 方法二

先于图标工作区点选要加上模块的主机或背板，之后再于左侧装置列表的模块型号上双击鼠标左键。



不论通过哪种方式，新增模块的背板与插槽编号都可能与实际的配置不同，因此加上模块之后可能仍必须重新调整节点的属性。





20.2.2 连结网络

完成节点的部署之后，接下来便是要将所有的节点依据事先的规划连上所属的网络。

于工作区中新增网络的作法共有下列几种方式，而所支持的网络类型则有 Ethernet 与 RS485 两种。

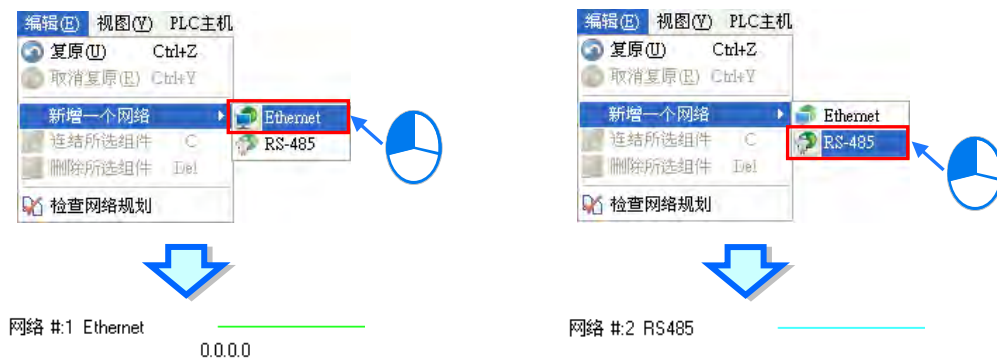
● 方法一

点击图标工具栏的  图标便可新增一 Ethernet 网络，而点击  图标则可新增一 RS485 网络。



● 方法二

功能工具栏中点选 **编辑 (E)** > **新增一个网络**，接着再于其下层选项中选择网络类型。



● 方法三

工作区的空白处点击鼠标右键，接着于快捷菜单中点选「新增一个网络」后在这个下层选择类型。



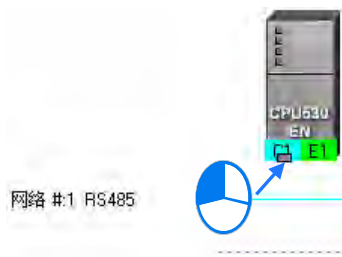
在每个节点图标下方都会显示该节点的通讯口，其中蓝色代表 RS485，而绿色则代表 Ethernet，至于通讯端口上的编号则会与该装置所定义的端口编号一致，例如 E1 便代表 Ethernet 的第一个端口，C1 代表 COM 1，C2 则代表 COM 2；不过在 SCM 的模块图标上，端口编号则会显示 TR1 与 TR2，而 10EN 的端口则会显示为 X1 或 X2，其编号的方式主要是要与实际模块上的端口标示一致；此外，当某个 Ethernet 通讯端口的 IP 地址被设为动态配置 (DHCP)，或是 SCM 模块的某个端口模式被设为 MODBUS 以外的型式时，该端口的颜色将会呈现灰色且无法连接至任何网络。



将工作区中的节点连上网络的方式有多种，说明如下。

● 使用鼠标拖曳来将装置连结至网络

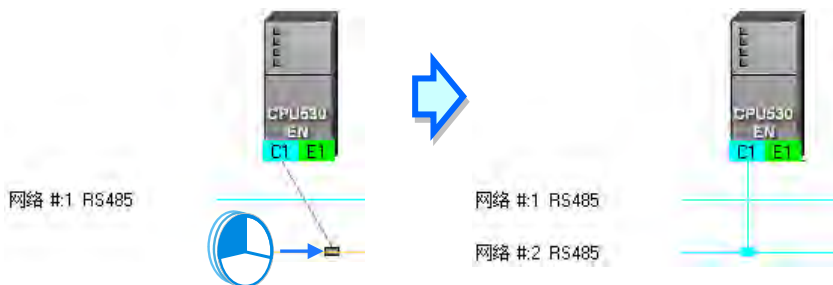
首先要在要连接的装置通讯口上按住鼠标左键不放，此时于现有网络的最下方会出现一条虚线。



将鼠标拖曳至已建立的网络上时，若该网络的类型与通讯口一致，则放开鼠标后便可建立链接。

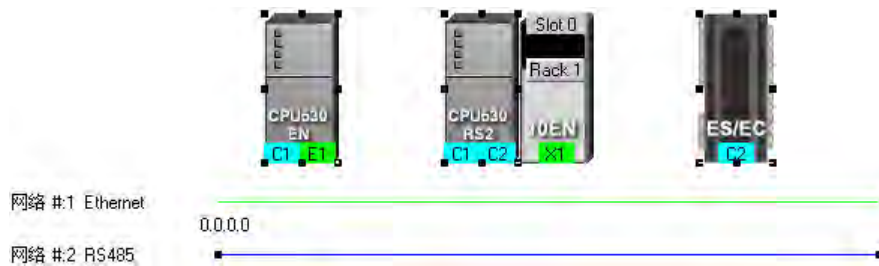



若将鼠标拖曳至最下方的虚线放开，则系统便会先新增一条相同类型的网络后，再将其接上。

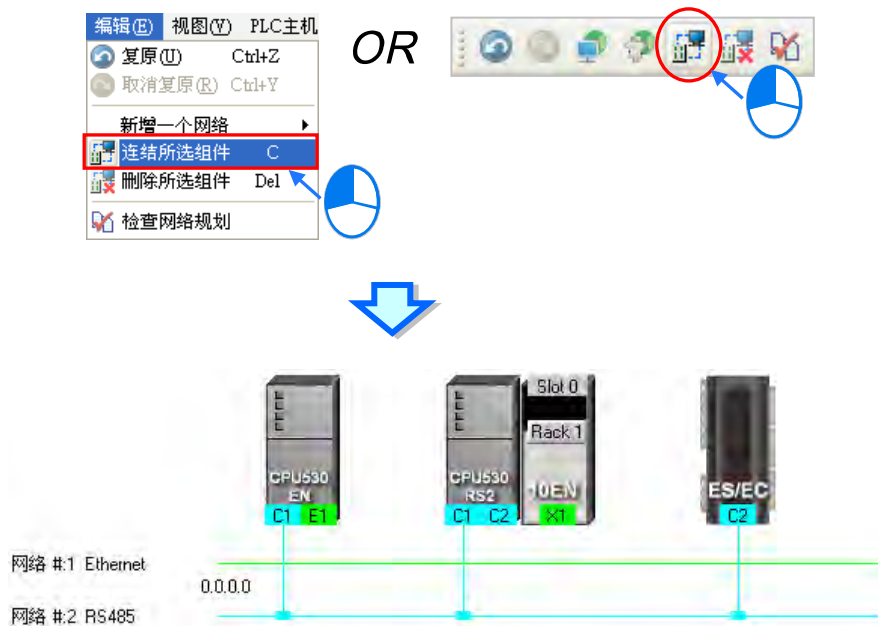


● 连接单个或多个装置至现有网络

- (1) 按住键盘的【Shift】键后，用鼠标依序点选要连接的装置与网络，而选取时须遵守下述两个原则。
- (a) 选取时，主机与模块须视为独立的个体，而所选取的装置至少都须保留一个未连接，且与选取网络相同类型的通讯口。
- (b) 同时间可选取多个装置，但只能选取一条网络。



- (2) 于功能工具栏中点选 **编辑 (E) > 连结所选组件**，或点击图标工具栏的  图标，接着系统便会自动将所选取的装置连接至选取的网络上。

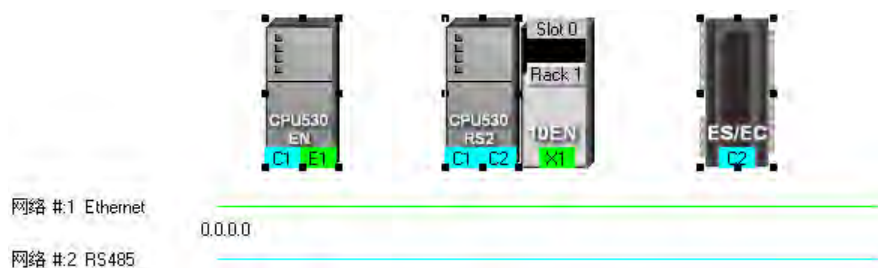



补充说明

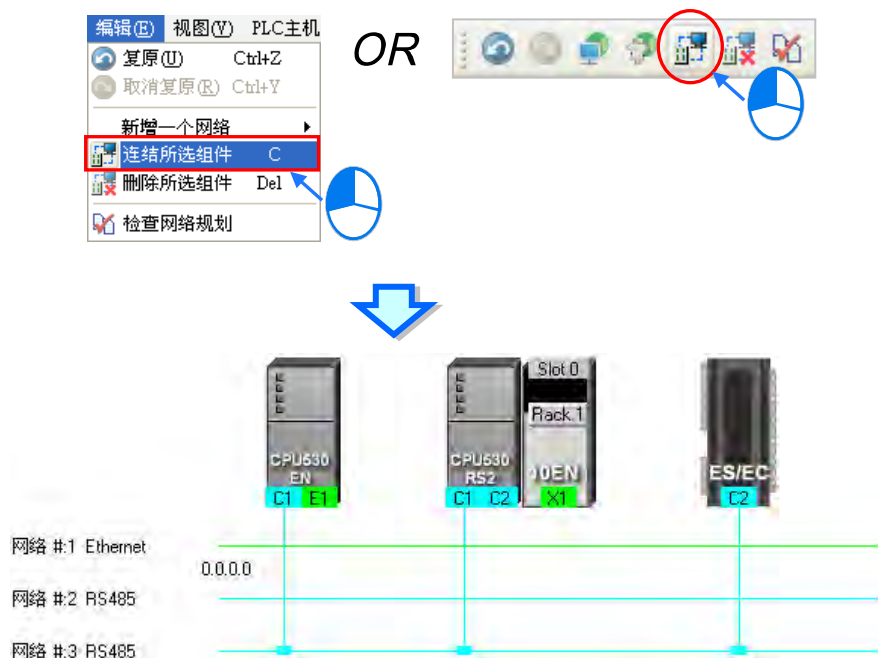
- (a) 当选取的对象不符合上述 (1) 的两个原则时，系统将不会建立链接，并且会出现警示讯息。
- (b) 当选取的装置包含一个以上符合连接条件的通讯口时，系统会优先连接编号较小的通讯口。
- (c) 可使用鼠标框选连续位置的装置，而使用键盘的【Ctrl】+【A】则可选取所有的装置与网络。
- (d) 选取的对象是以装置为单位，因此若同时选取某个节点的主机与模块，且两者都符合上述的原则时，两者的通讯口都会同时被连接至所选取的网络上。

● 自动连接单个或多个装置至一新增网络

- (1) 按住键盘的【Shift】键后，用鼠标依序点选要连接的装置，但不点选网络；选取时，主机与模块须视为独立的个体，而所选取的装置至少都须保留一个未连接且类型相同的通讯口。



- (2) 功能工具栏中点选 **编辑 (E)** > **连结所选组件**，或点击图标工具栏的  图标，接着系统便会自动新增一条网络，并将所选取的装置连接至该新增网络上。



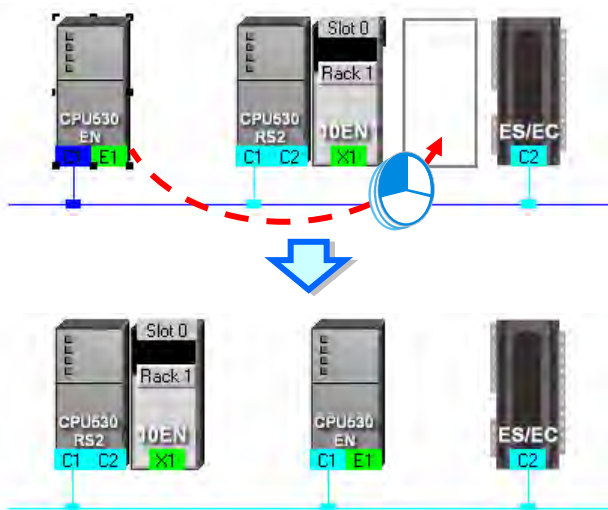
补充说明

- 当选取的装置不符合上述 (1) 的选取原则时，系统将不会建立链接，并且会出现警示讯息。
- 当选取的装置包含一个以上符合连接条件的通讯口时，系统会优先连接编号较小的通讯口。
- 可使用鼠标框选连续位置的装置；而使用键盘的【Ctrl】+【A】则可选取所有的装置与网络。
- 选取的对象是以装置为单位，因此若同时选取某个节点的主机与模块，且两者都符合上述的选取原则时，两者的通讯口都会同时被连接至网络上。
- 在进行上述操作时，系统将会自动侦测选取装置中空闲且类型一致的通讯端口，但若同时符合 RS-485 与 Ethernet 两种类型时，系统将会优先建立 Ethernet 的连接。

20.2.3 调整或删除已部署的装置及网络

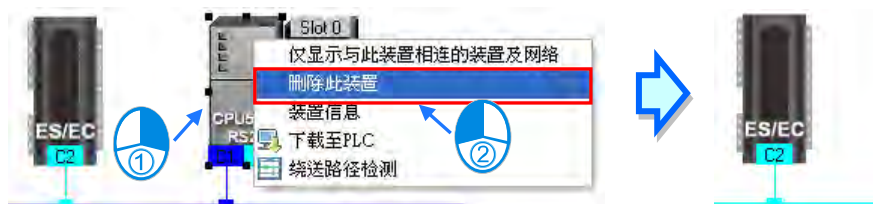
● 调整节点的排列顺序

工作区中的某个节点图标上按住鼠标左键，接着便可将其拖曳至其它的位置上摆放，如此便可改变节点图标的排列顺序；且因 NWCONFIG 中的节点图标只能放在同一高度并持续向右方扩展，因此无法将节点图标拖曳至上方或下方的位置摆放。

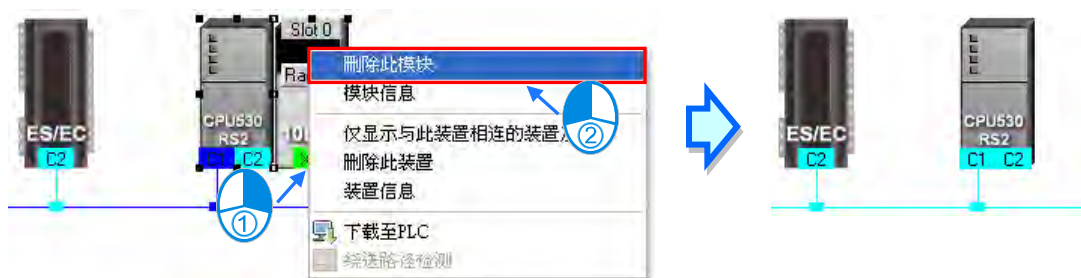


● 通过右键菜单删除单一装置

要删除整个节点时，请在主机的图标上点击鼠标右键，接着于快捷菜单中点选「删除此装置」，接着连同主机与附挂的模块便会一并被删除；但依据项目设定而自动带出来的主机或模块则无法删除。

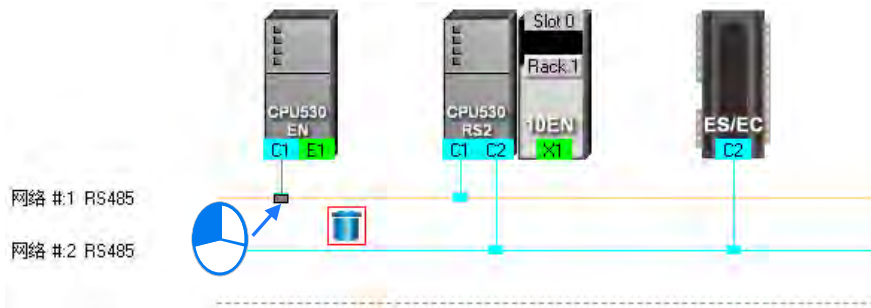


若只要删除模块时，请点选要删除的模块之后按下鼠标右键，接着于快捷菜单中点选「删除此模块」，接着该模块便会被删除。

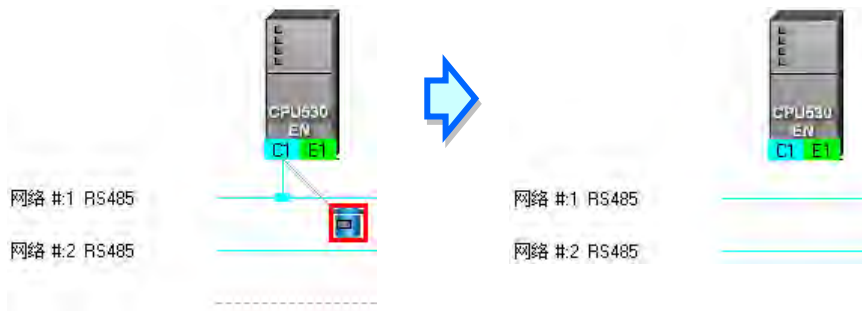


● 调整网络链接

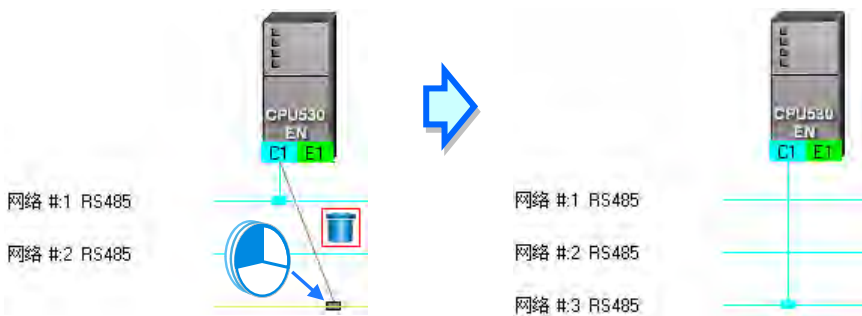
网络上要调整链接的接点上按住鼠标左键，此时于该节点的右下方会出现一个垃圾桶的图标，而在网络的最下方也会出现一条虚线。



若将鼠标拖曳至垃圾桶图标放开时，该端口的链接便会被取消。



若将鼠标拖曳至下方的虚线时，系统便会先新增一条网络后，再将通讯口连接至该网络上。

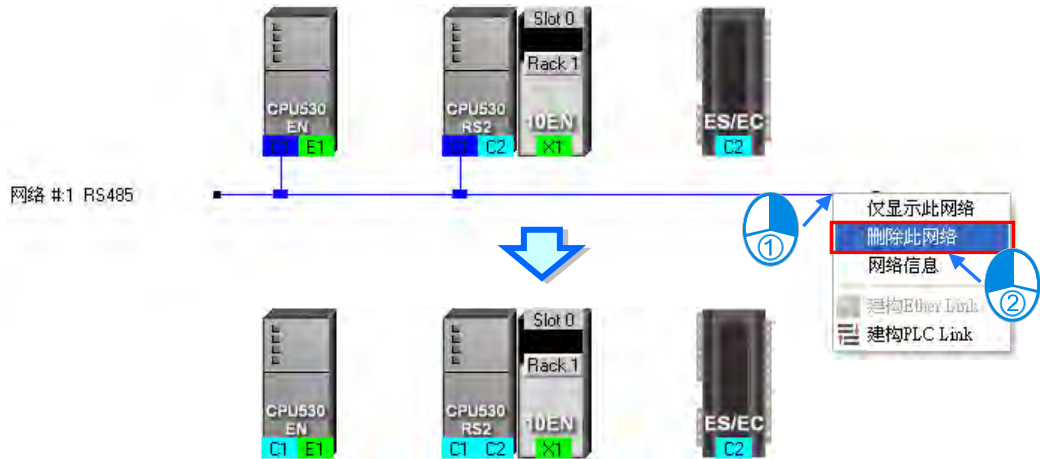


当将鼠标拖曳至另一相同类型的网络上放开后，便可将该通讯口的链接变更至该网络。



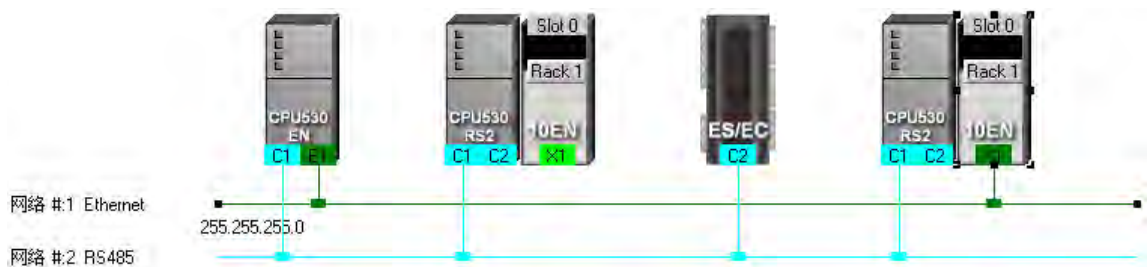
● 通过右键菜单删除单一网络


在要删除的网络上点击鼠标右键，接着于快捷菜单中点选「删除此网络」，之后该网络便会被删除，而原本连接至该网络的接线也会一并消失。

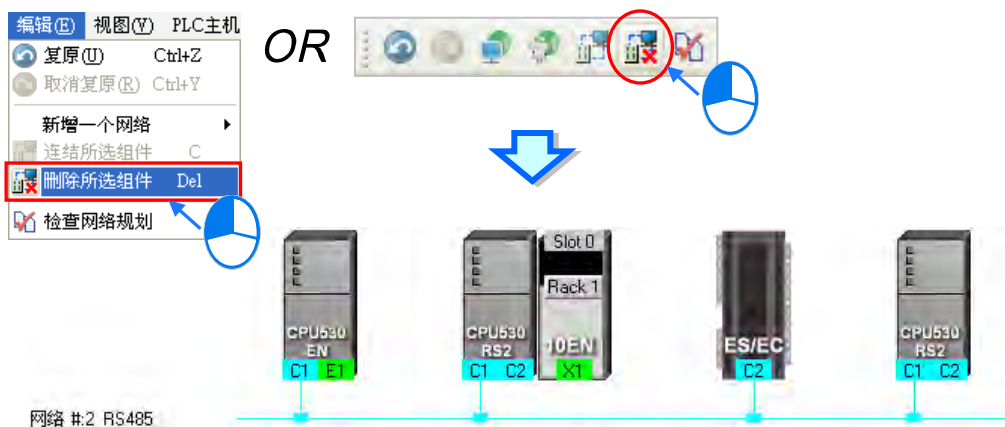


● 同时删除多个装置或网络

首先请于工作区中点选要删除的对象，而按住键盘的【Shift】键时，便可用鼠标同时点选多个组件；也可用鼠标框选连续位置的组件，或使用键盘的【Ctrl】+【A】将所有的对象同时选取起来。

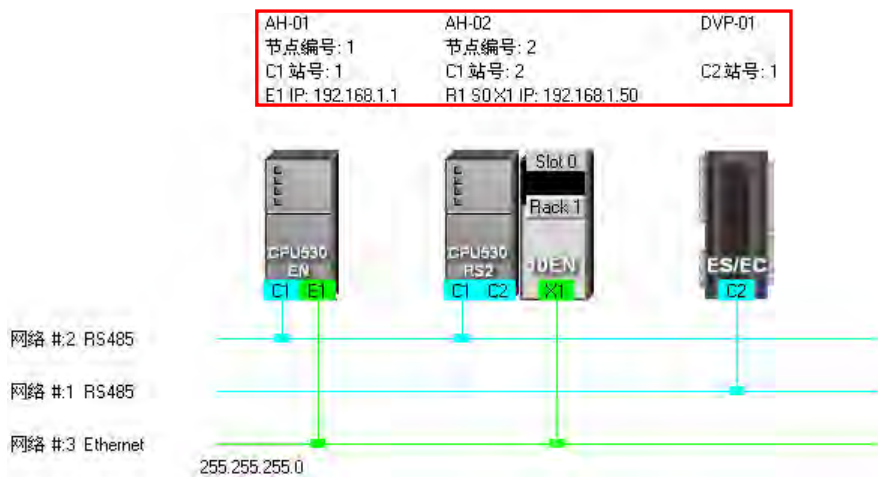


选取完要删除的组件后，于功能工具栏中依序点选 **编辑 (E)** > **删除所选组件**，或点击图标工具栏的  图标，也或是直接按下键盘的【DEL】键即可删除所选的组件，不过若是依据项目设定而自动带出来的节点则无法将其删除；此外，删除主机时，附挂于该主机的模块也会一并被删除。

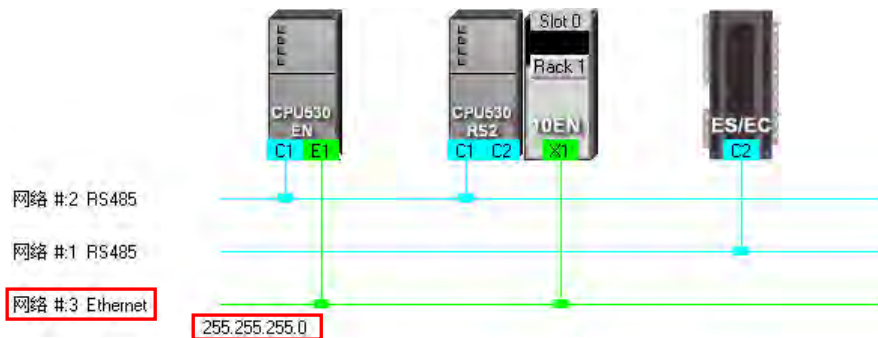


20.2.4 节点与网络属性的设定

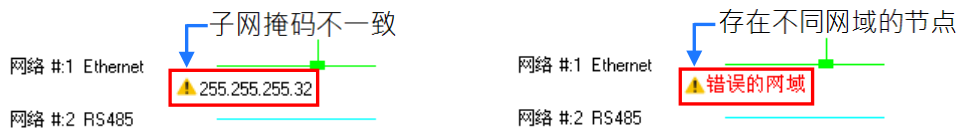
当完成网络的部署之后，于各个节点的上方会显示该节点的相关信息，而显示的内容除了 PLC 名称与节点编号之外，同时还会显示有连接至网络的通讯端口属性，若是 RS485 端口则会显示站号，Ethernet 端口则会显示 IP 地址；而其中只有连接网络的通讯端口才会有信息显示，未连接任何网络的端口将不会显示信息；属于通讯模块的端口信息还会显示背板与插槽编号以便于识别。



除了节点信息之外，于各个网络的左侧也会显示该网络的编号与类型，而在 Ethernet 的网络上还会显示该网络的子网掩码设定。



当链接至同一个网络编号的 Ethernet 端点中，若其中有某个端点所设定的网域不同，或是子网掩码设定不一致时，在该网络上便会出现警示的图标；而当子网掩码设定不一致时，该网络上的子网掩码值将会显示最严谨的那一组。



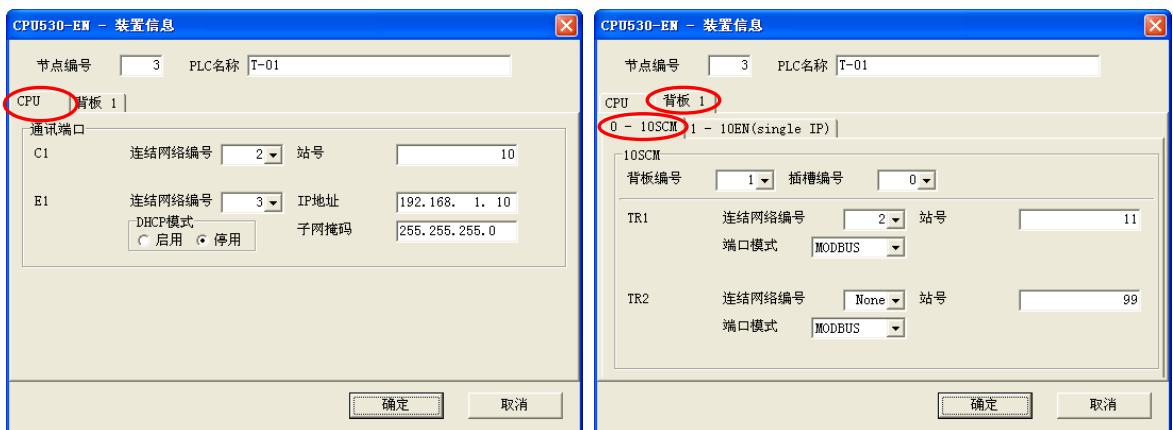
● 设定节点属性

要开启节点属性窗口时，可通过下列两种方式。

- (a) 直接于要设定属性的主机图标上双击鼠标左键，或点选主机图标后按下鼠标右键，接着于快捷菜单中点选「装置信息」。
- (b) 直接于要设定属性的模块图标上双击鼠标左键，或点选模块图标后按下鼠标右键，接着于快捷菜单中点选「模块信息」。

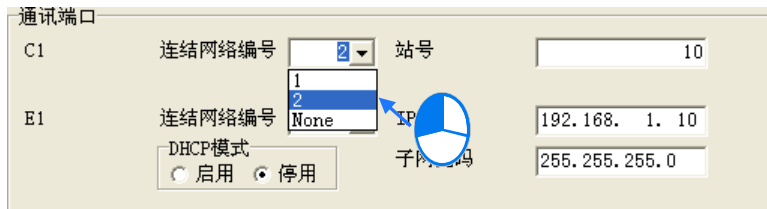


在节点属性的设定窗口中，主机与模块的属性会被放在不同的页面，而用户则可通过上方的选项卡来进行切换，且开启窗口时，系统将会依据用户所点选的装置而自动切换到对应的页面。

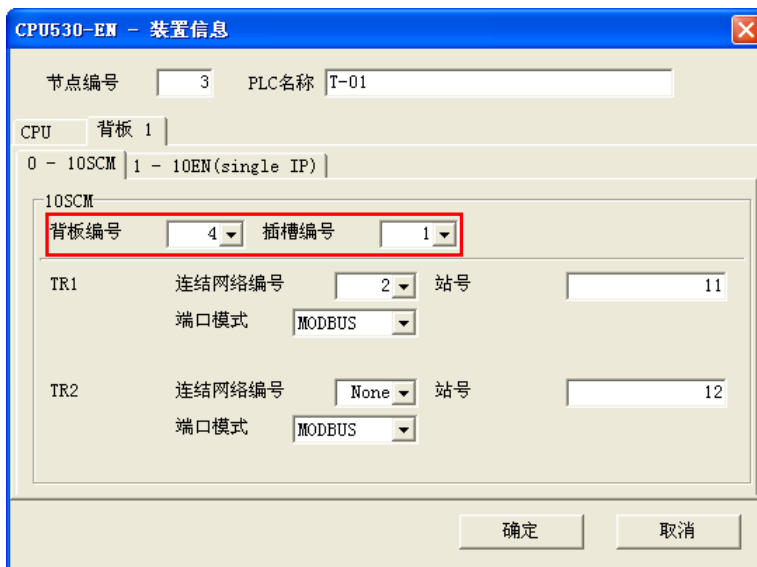


若所点选的装置是依据项目设定而自动带出来的第一台 AH5x0 系列机种，则开启的属性窗口中，其属性字段大部份是灰阶的，其属性必须通过 HWCONFIG 才可进行设定。

在设定页面中，每个通讯口都会有一个对应的「**链接网络编号**」字段，于下拉选单中便会出现可链接的网络编号选项；若选择 **None** 时，则该端口的网络联机便会被删除。



在模块的设定页面中还可重新定义模块的**背板**与**插槽编号**。

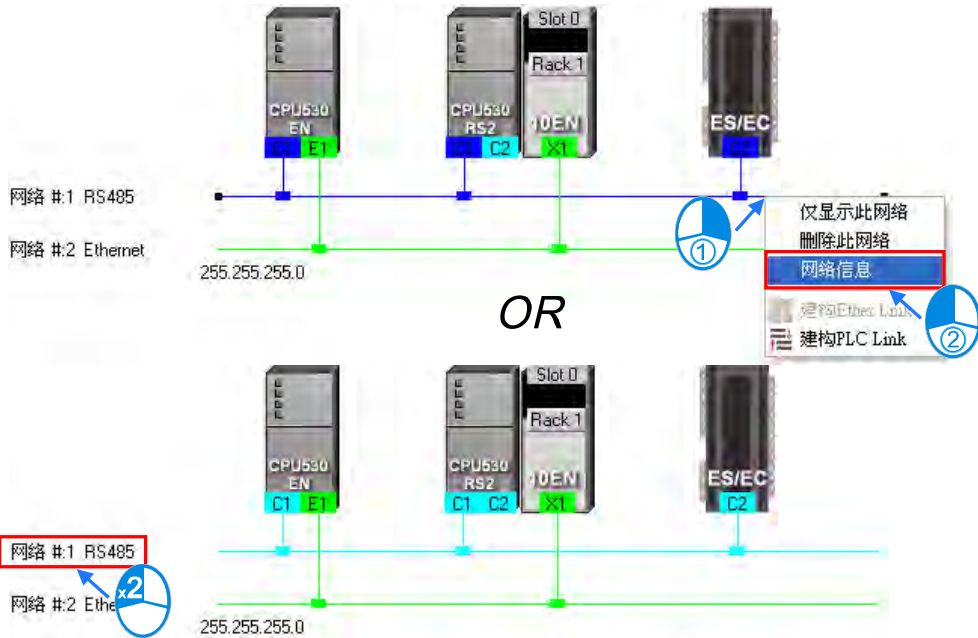


完成设定后请按下「**确定**」键，而此时该节点的状态与上方的信息便会立即更新。

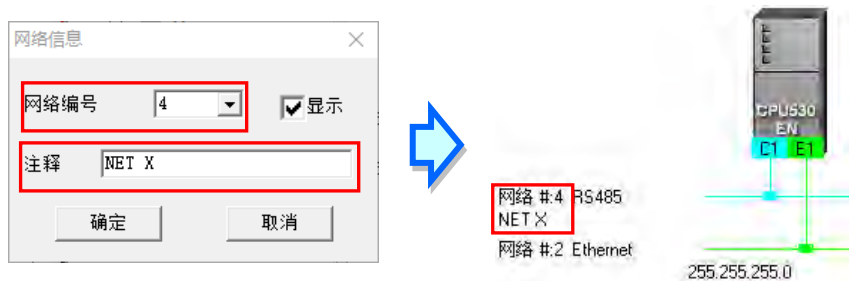


● 设定网络属性

点选要设定属性的网络后按下鼠标右键，接着于快捷菜单中选择「网络信息」，也或直接从该网络左侧的文字上双击鼠标左键，接着便会开启网络属性的设定窗口。



于网络属性的设定窗口中，我们可重新定义该网络的编号，但只可选择尚未被使用的网络编号，而除此之外，我们尚可为该网络加上注释；完成设定之后按下「确定」键便可立即应用所做的变更。



网络信息窗口中的「显示」项目取消勾选后，此网络会隐藏且有连的通讯口会以虚线来表示。



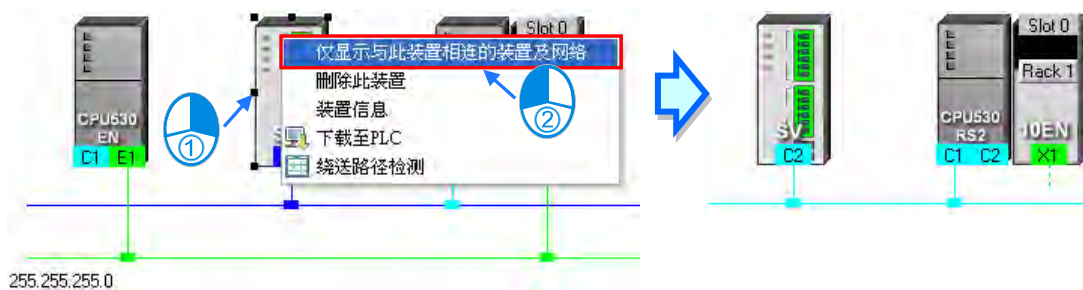
20.2.5 装置或网络的隐藏/显示

为了便于编辑与查看，图标区的装置与网络皆可由用户自行定义其显示与隐藏的状态，且由下方的状态栏中便可知目前显示的装置与总共的装置数量。当某个装置的端口有虚线时，便代表该端口有链接至某个被隐藏的网络。



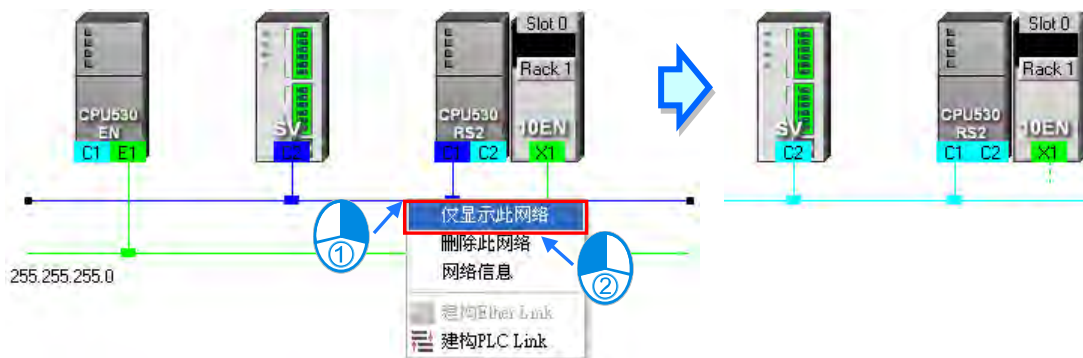
- 只显示与某一装置相关的对象

某一节点的图标上点击鼠标右键后，在快捷菜单中选取「仅显示与此装置相连的装置及网络」，接着图标区便只会显示与此节点直接相连的网络与装置。



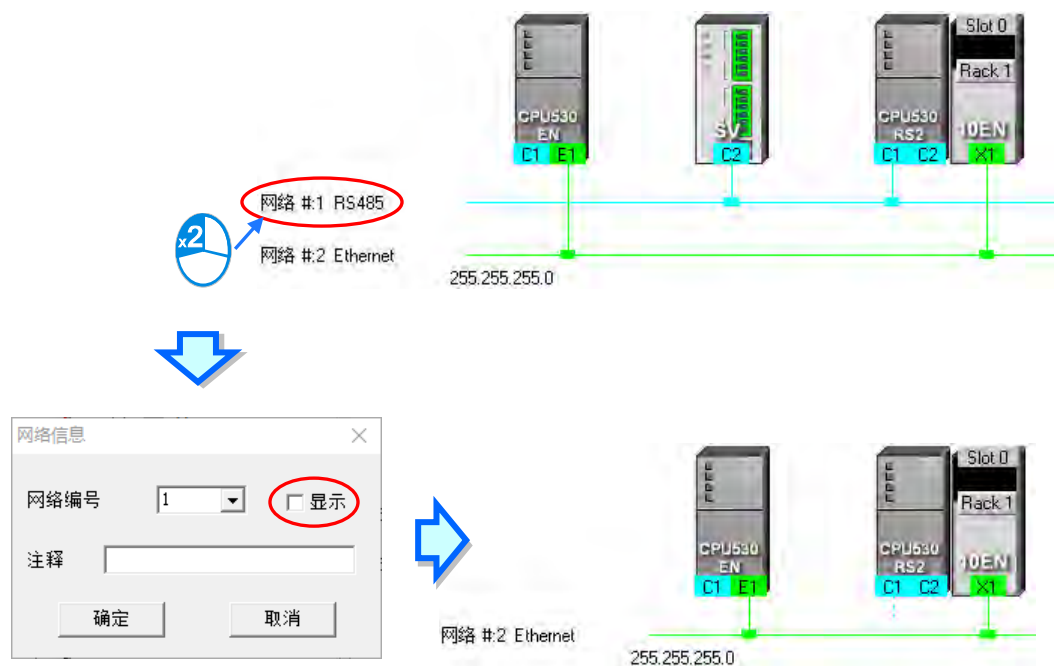
- 只显示某一网络上的装置

在某一网络上点击鼠标右键后，接着再于快捷菜单中选取「仅显示此网络」，接着图标区便只会显示与此网络直接相连的装置。



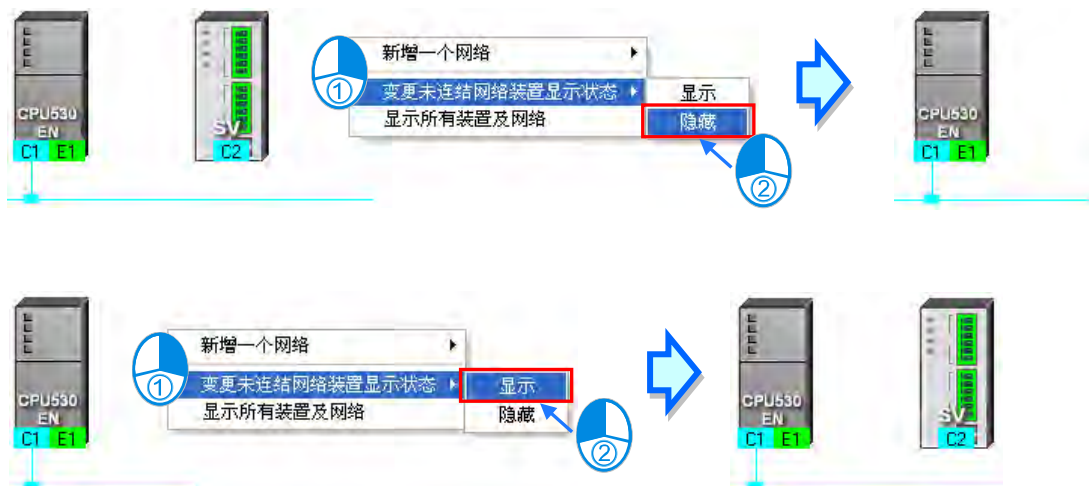
● 隐藏某一网络与连接至该网络的装置

开启网络属性窗口后，将「显示」的勾选取消，此网络会隐藏且有连的通讯口会以虚线来表示。




● 隐藏/显示未连接的装置

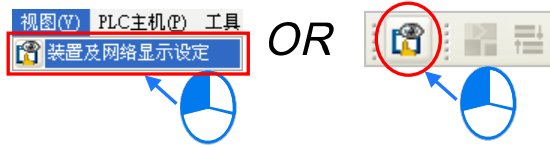
要隐藏或显示图标工作区中未连接至任何网络的装置时，请先于图标工作区的空白处点击鼠标右键，接着在快捷菜单中选取「变更未连结网络装置显示状态」，接着便可于下阶的菜单中设定装置的显示状态；不过每次操作时仅会对目前工作区中的装置进行设定，而后续加入的装置则不受其影响。



● 批次设定所有组件的显示状态

利用此功能便可以网络为单位，而同时设定所有装置的显示状态。

首先请于功能工具栏中点选 **视图(V) > 装置及网络显示设定**，也或直接点选图标工具栏的  图标便可开启设定窗口。



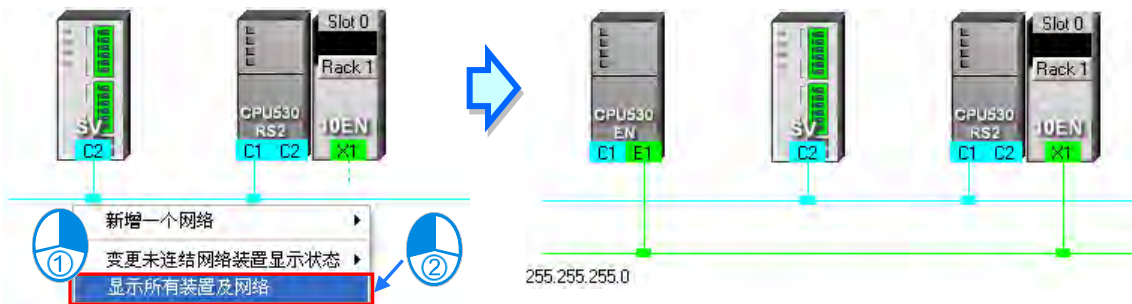
设定窗口的「网络」区域中勾选要显示的网络编号，未勾选的选项即代表设为隐藏，而通过「全选」则可同时勾选或取消所有对象的勾选状态。

在「无连结网络装置」的区域中，用户可设定目前图标区中未连接至任何网络的装置显示状态；而同样的，每次的操作只会对目前工作区中的装置进行设定，而后续加入的装置则不受其影响。




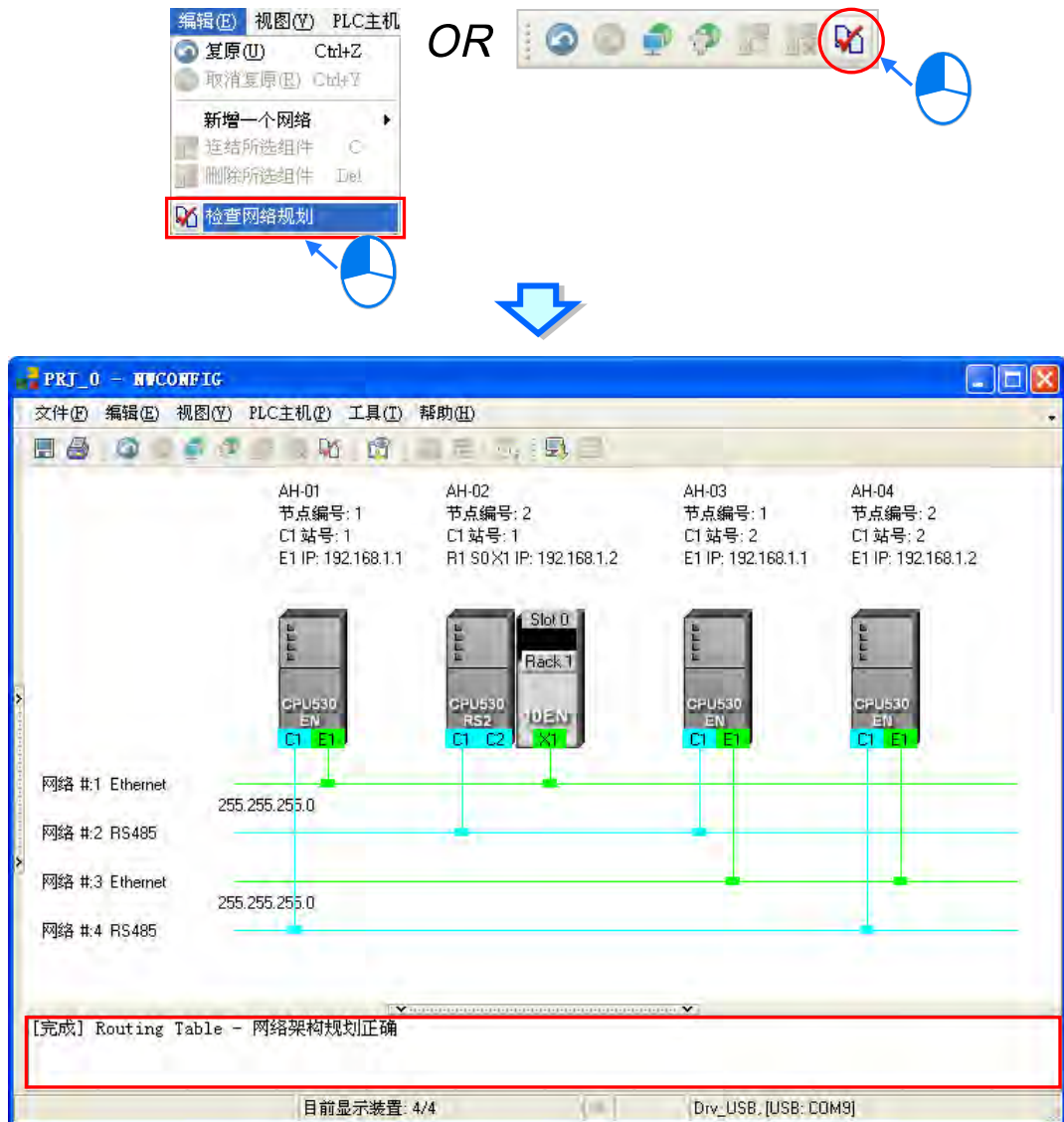
● 显示所有组件

图标工作区的空白处点击鼠标右键后，在快捷菜单中选取「显示所有装置及网络」，接着所有被隐藏的装置及网络便会重新恢复显示。



20.2.6 合法的网络架构

当完成网络架构的部署之后，于功能工具栏中依序点选 **编辑 (E)** > **检查网络规划**，也或于图标工具栏中选  图标，接着便可对目前所部署的网络进行合法性的检查，而检查的结果则会显示于下方的讯息窗口中。



在进行检查时，系统仅会针对有连接网络的端口进行确认，也可通过显示于图标工作区的节点或网络信息来确认规划是否正确，而主要的检查项目则包括**节点编号**、**RS485 站号**与**IP 地址**；若依据上图的内容来说，整体看来会觉得存在**节点编号**、**RS485 站号**与**IP 地址**重复的问题，因此我们可以每次只检视一条网络，便可清楚的确认到连接至同一个网络上的节点，其**节点编号**、**RS485 站号**与**IP 地址**有没有发生重复的现象。

● 节点编号

原则为同一个网络上的节点编号不得重复，而通过每次只查看一条网络的方式，我们便可清楚的确认到连接至同一个网络上的节点，其编号并没有发生重复的现象。



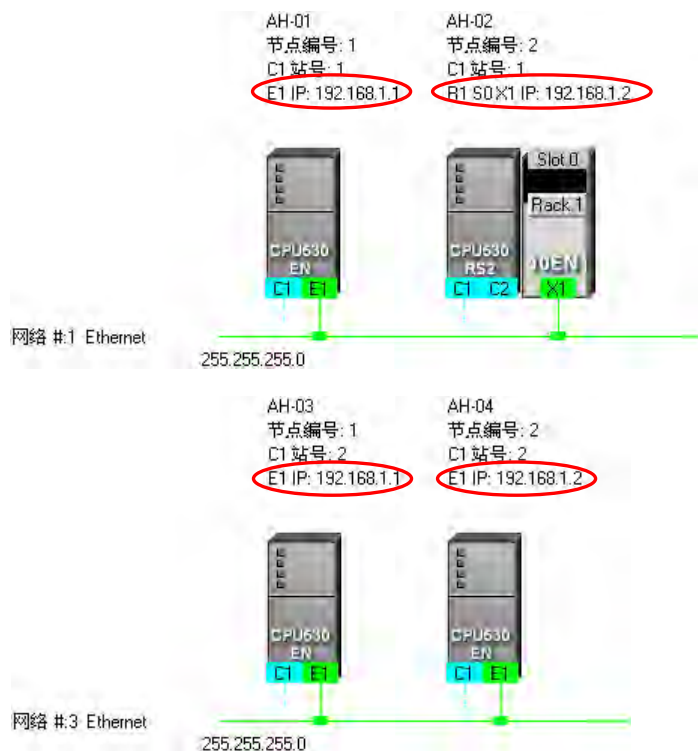
● RS485 站号

原则为同一个网络上的站号不得重复，而通过每次只查看一条 RS485 网络的方式，我们便可清楚的确认到连接至同一个网络上的 RS485 端口，其站号也没有发生重复的现象。



● IP 地址

原则为同一个网络上的 IP 地址不得重复，因此我们可逐一查看每一条 Ethernet 网络上的端口信息，且在检查之后便可确认同一个网络上的 IP 地址并没有重复。



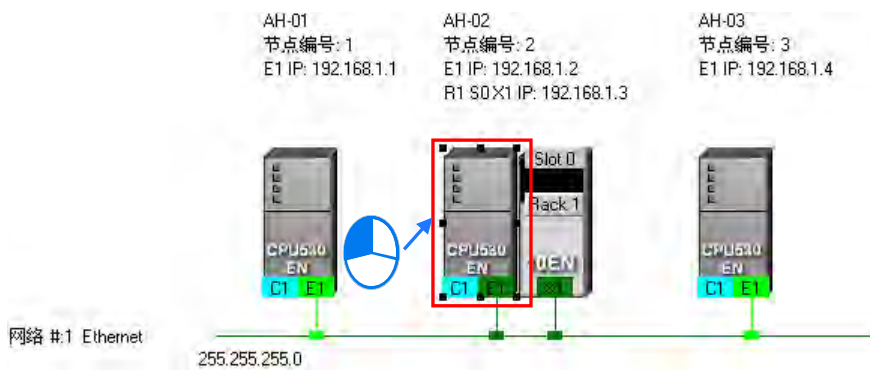
20.2.7 下载绕送路径表 (Routing Table)


当完成网络架构的部署并检查无误后，接着便可将绕送路径表 (Routing Table) 下载至各节点的主机当中，每个主机的内部只会保存与其本身相关的路径数据，所以不同节点会被下载的绕送路径表 (Routing Table) 数据也会不同，而下载时也必须逐一完成每个节点的数据下载。

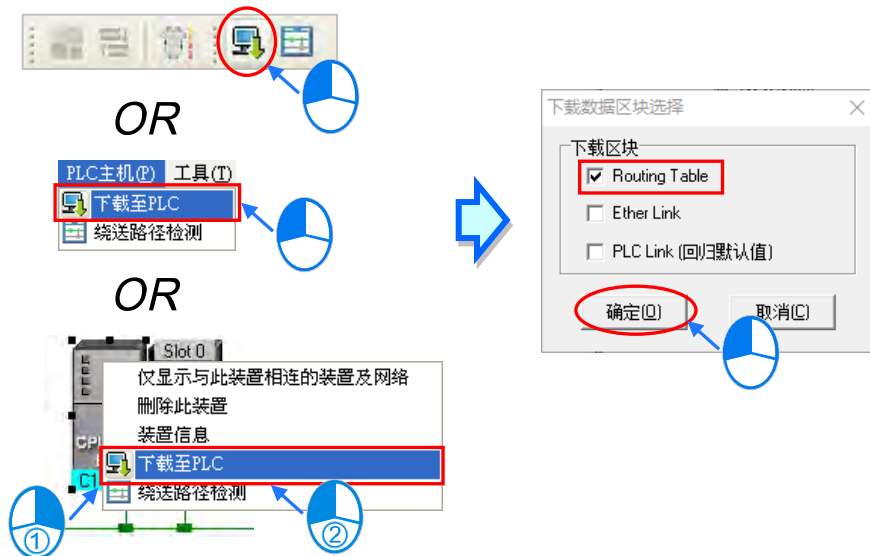
绕送路径表 (Routing Table) 的下载方式可分为单点与多点两种模式，且下载时的通讯设定不可使用报文绕送模式。

● 单点模式

首先选择要下载的点，并于图标工作区中点选该节点的主机部分，且因绕送功能只支持 AH5x0 系列的机种，因此选择其它机种或装置时将无法下载绕送路径表 (Routing Table)。

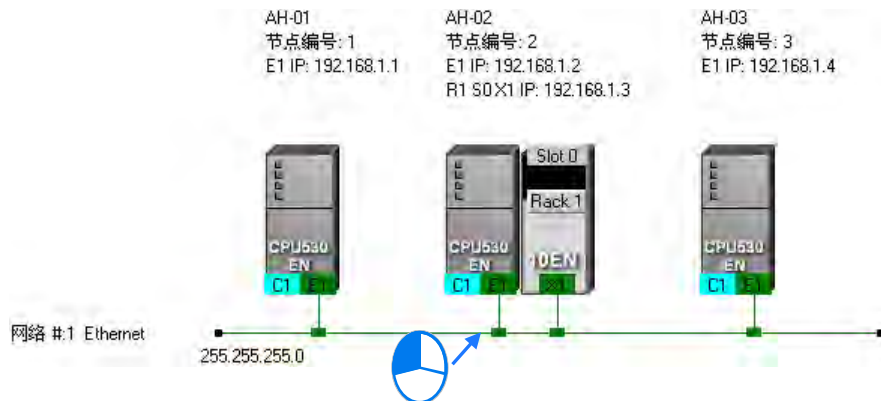



点选要下载的主机之后，请于功能工具栏中点选 **PLC 主机 (P) > 下载至 PLC**，或直接从图标工具栏中点选  图标，同时也可于主机图标上点击鼠标右键，接着再于右键快捷菜单中点选「**下载至 PLC**」即可；而接着勾选「**Routing Table**」，完成之后按下「**确定**」便可将绕送路径表 (Routing Table) 下载至指定的主机当中。

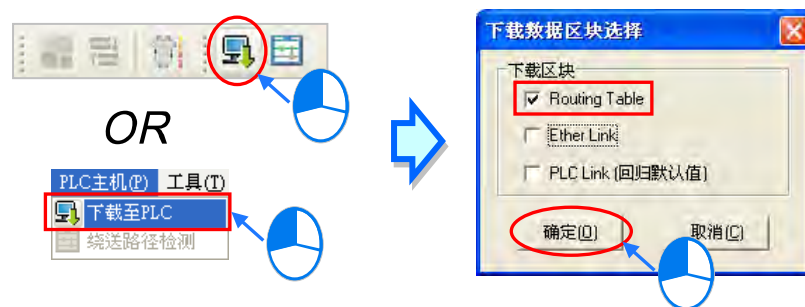


● 多点模式

通信设置中的 COMMGR Driver 必须为 Ethernet 的型式，接着于图标区中点选其中一条的 Ethernet 网络，之后便可将参数下载至与该网络连接的所有节点。



点选完网络后，请于功能工具栏中点选 **PLC 主机 (P)** > **下载至 PLC**，或直接于图标工具栏中点选  图标即可；而接着勾选「**Routing Table**」，完成之后按下「**确定**」键便可将绕送路径表 (**Routing Table**) 下载至与该网络连接的所有节点。



补充说明

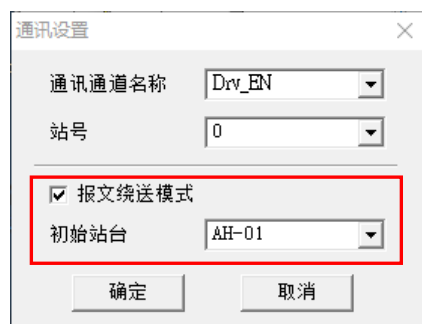
当点选的下载对象有包含 **Ether Link** 或 **PLC Link** 的参数可下载时，于下载选项的窗口中也可将该选项一并勾选，但当勾选对象为灰阶时，便代表该对象无法下载；关于 **Ether Link** 与 **PLC Link** 的相关内容将于后续的章节当中另行介绍。


因为每个主机内部只会保存与其本身相关的路径数据，因此无法将绕送路径表 (**Routing Table**) 自主机内部上传并于 NWCONFIG 的工作区中重建，因此系统并未提供绕送路径表 (**Routing Table**) 的上传功能。

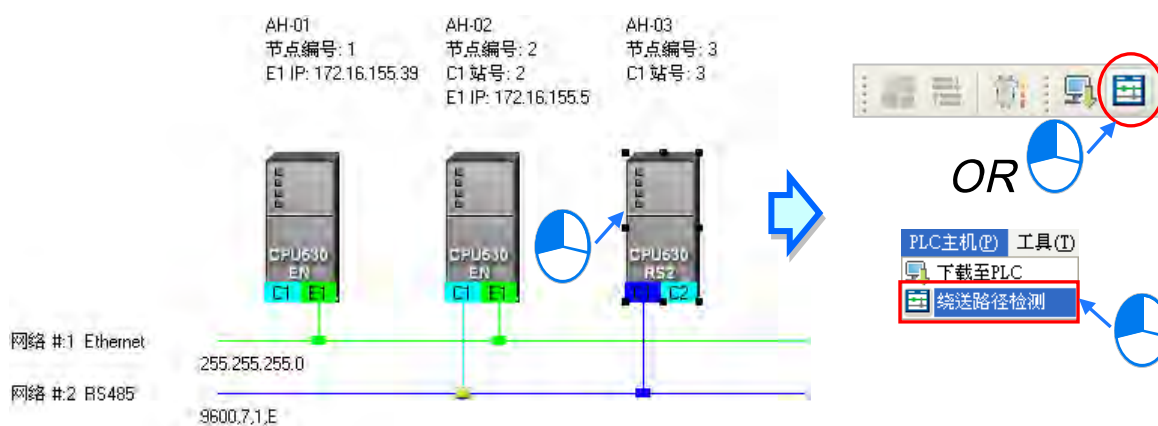
20.2.8 绕送路径测试

当完成绕送路径表 (Routing Table) 的下载之后, 接下来便可通过 NWCONFIG 所提供的功能进行绕送路径的测试。操作的步骤如下。

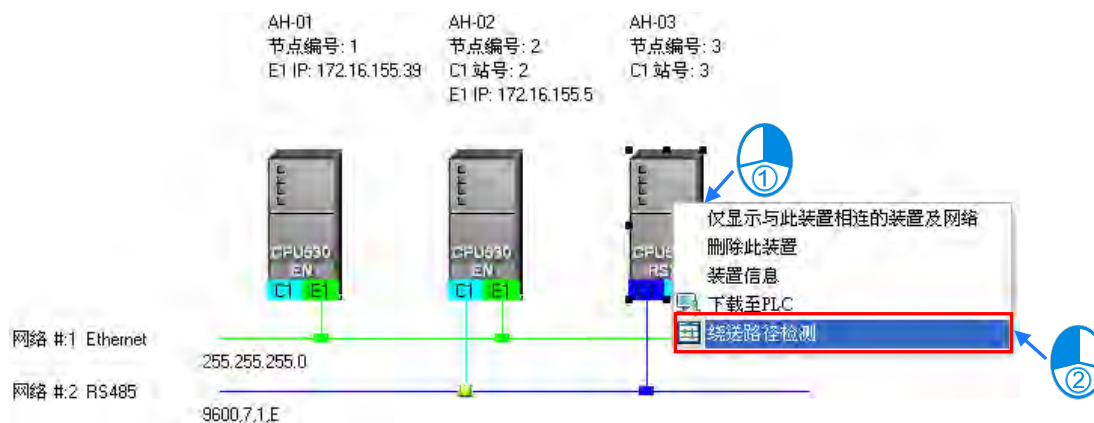
- (1) 请先确认所有节点都已依照 NWCONFIG 中的配置完成接线, 且已处于正常运作的状态, 而每个装置内部的设定也都与 NWCONFIG 中的设定一致。接着请于通讯设定窗口中启用「报文绕送模式」, 并选择「初始站台」。



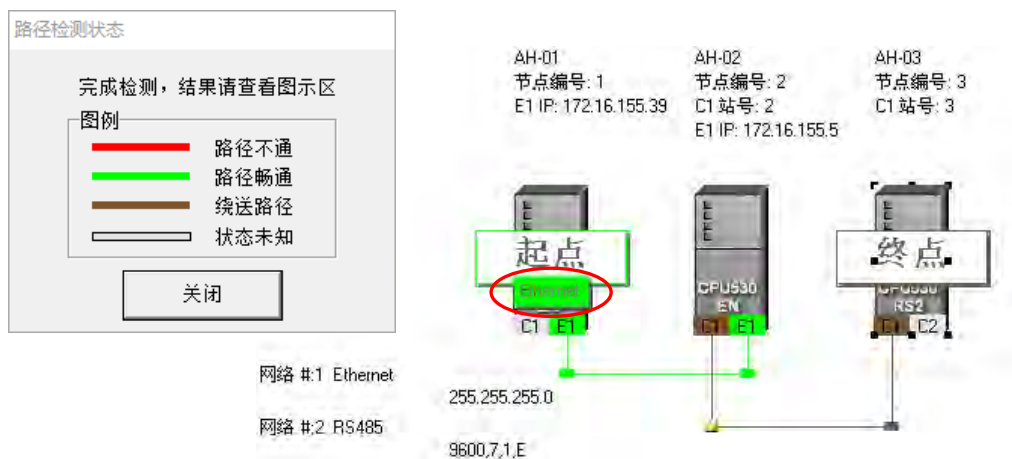
- (2) 于图标工作区中点选测试路径的终点装置之后, 直接点选图标工具栏中的  图标, 或是于功能工具栏中依序点选 **PLC 主机 (P)** > **绕送路径检测**。



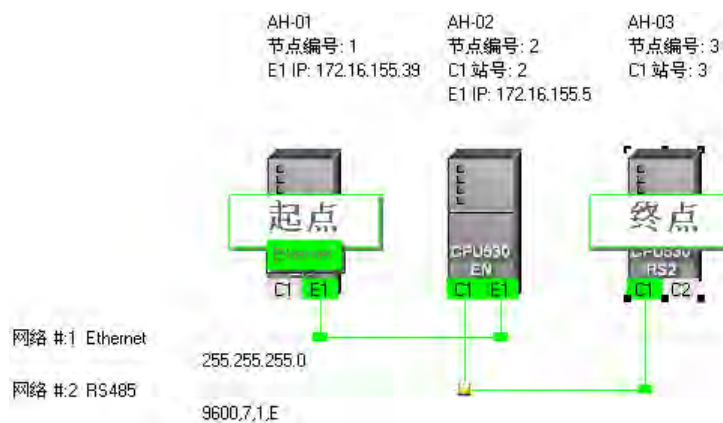
同时也可点选路径的终点装置后按下鼠标右键, 接着于快捷菜单中点选「绕送路径检测」。



- (3) 执行「**绕送路径检测**」的功能之后，于图标工作区中便会有对应的表达图标，而此时画面也会出现状态窗口。其中于指定为**初始站台**的节点图标中尚会显示计算机与**初始站台**所采用的联机方式，请参考下图中的红框。



- (4) 完成检测后，画面上便会显示最后检测的结果。



补充说明

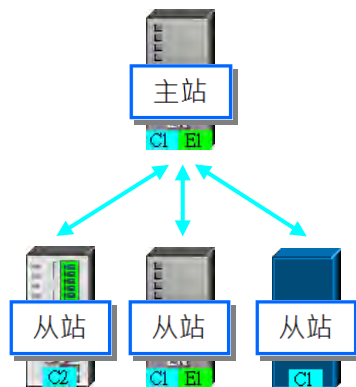
若检测的结果失败时，请再次确认下列事项。

- (a) 请确认硬件配置都与 NWCONFIG 中的配置一致、且每个节点都已正确的连上网络并处于正常运作的状态。

20.3 PLC Link 的建构

PLC Link 为通过 RS485 的联机来进行数据交换的网络机制。当某个 RS485 的网络上存在多个节点时，我们便可于该网络中规划一个数据交换的机制；而当设置参数被下载至设为主站的主机且开始运行之后，通过特殊继电器与特殊寄存器的操作，系统便可自动进行数据交换的动作。


PLC Link 是一种主从架构，一个 RS485 的网络上只能有一个主站，主站会轮流对其它从站发出读写命令，而其它从站则必须被动的在接收到主站的命令后才可做出响应，因此从站与从站之间无法直接交换数据，但若在应用上需要从站之间的数据交换时，规划时便必须通过主站来建立转发的机制。

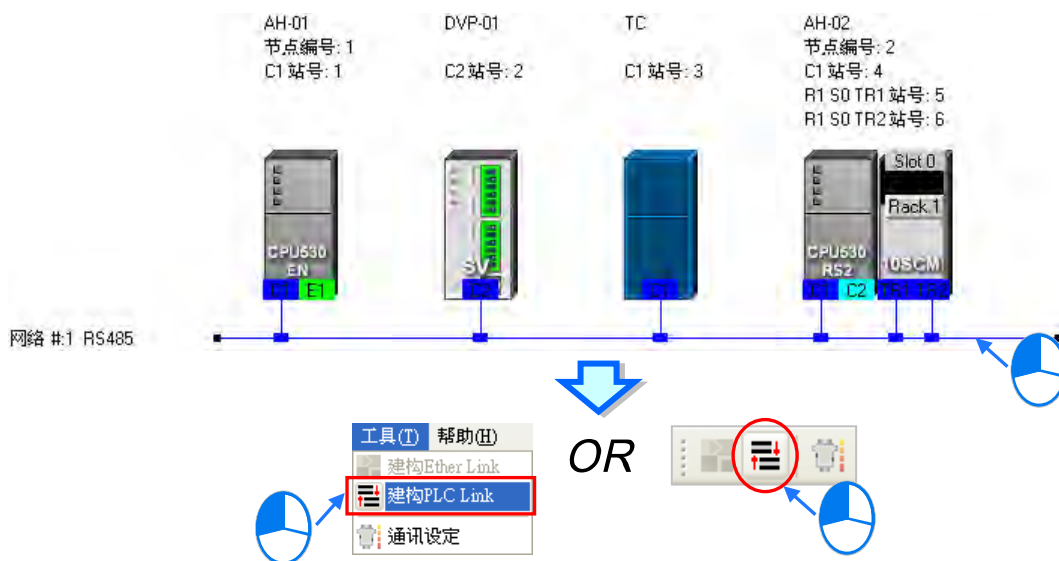


20.3.1 打开 PLC Link 的设定窗口

开始建构 PLC Link 之前请务必确认所有的网络设置正确无误后，依照下述方法进行操作。

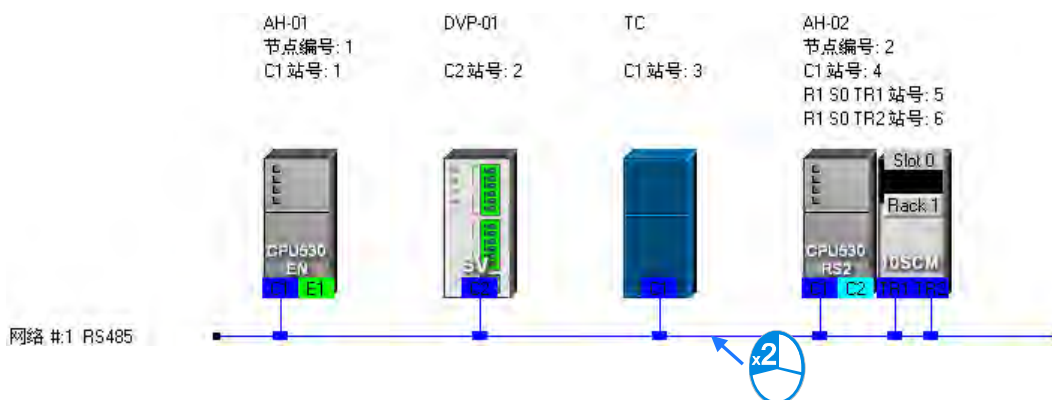
● 方法一

首先点选要建构 PLC Link 的网络。接着于功能工具栏中点选 **工具 (I) > 建构 PLC Link**，或点击图标工具栏中的  图标。



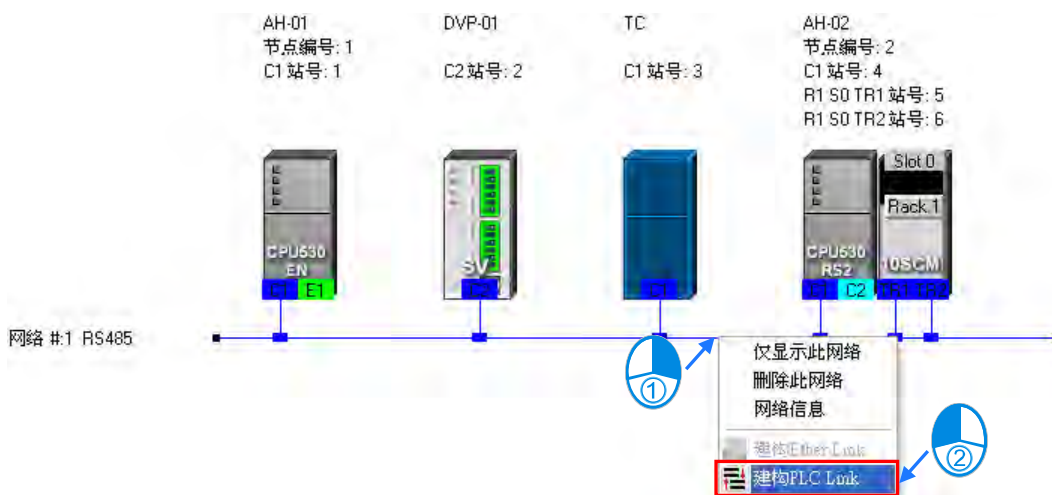
● 方法二

于要建构 PLC Link 的网络上双击鼠标左键。



● 方法三

于要建构 PLC Link 的网络上点击鼠标右键，接着于快捷菜单中点选「建构 PLC Link」。

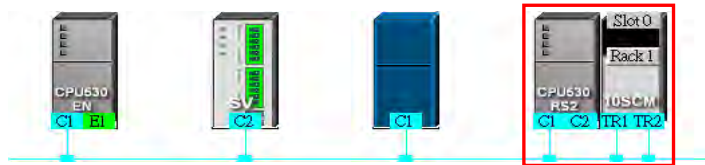


依据上述的其中一种方式操作完毕之后，画面便会开启 PLC Link 的规划工具，而 PLC Link 的规划工具则是以引导的方式逐步带领用户完成 PLC Link 的建构，其主要的步骤分别是 **选择主站装置** → **设定通讯参数** → **设定数据交换表**，而当开启规划工具之后，系统将会直接进入上次的操作画面，因此若该网络为第一次进行规划时，开启的画面便会进入第一个步骤。

PLC Link 的实际执行是通过特殊继电器与特殊寄存器的操作来进行，而此处所介绍的规划工具则只是提供较为亲切的用户界面，以协助用户将 PLC Link 的相关参数下载至对应的特殊继电器与特殊寄存器当中，因此实际执行的情况仍会依据当时 PLC 内部的特殊继电器与特殊寄存器而定，所以进行架构时，请务必在正式执行之前确认各主机在 PLC Link 这部份的相关功能与限制，以确保所规划的 PLC Link 功能可正常运作。


20.3.2 选择主站装置 (步骤 1)

指定主站时只可选择 AH5x0 系列或 DVP 系列的 PLC 机种，且因不是所有主机或模块的通讯口都可被指定为主站，因此设定之前请详阅各机种与模块的使用说明。若一个节点中有多个通讯端口被连接至同一网络，且若这些通讯口皆可成为主站时，则最后不论指定哪一个通讯口为主站，实际进行 PLC Link 操作的都将会是该节点的主机，因此执行的结果也都会相同的。

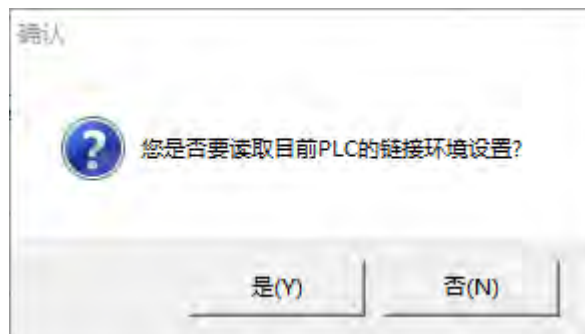


(1) 在「主站装置」的下拉菜单中指定主站，而此处的下拉菜单则仅会列出可设为主站的通讯口。



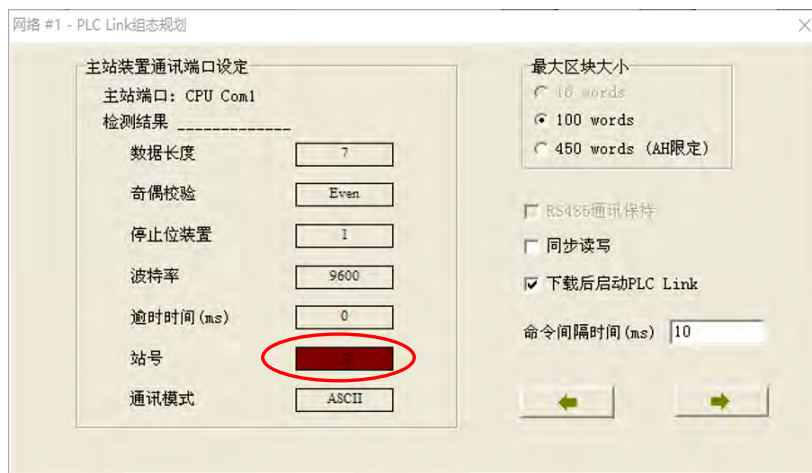
(2) 选择完成后，按下  图标便可进入下一个步骤，而此时系统会先询问用户是否要直接上传主站装置内部的 PLC Link 设定，若选择「否」，画面便会直接进入第二步骤；若选择「是」，系统便会自指定为主站的装置中，将该节点主机内部的 PLC Link 设定上传回来，而之后便会直接将上传的数据显示在第三步骤的画面中；不过在执行上传操作前，请先确认下列事项。

- (a) 请确认计算机与指定的主站装置之间已正确用通讯线缆完成接续。
- (b) 已建立正确的 COMMGR Driver，且该 Driver 已处于「OK」状态。
- (c) 已正确完成 NWCONFIG 中的通讯设定。



20.3.3 设定通讯参数 (步骤 2)

此步骤主要在设定进行 PLC Link 时的通讯参数，其中在画面左侧的区域则会显示由主站上传回来的通讯端口参数，而在同一网络上的所有 PLC Link 从站，其通讯参数皆须与主站一致才可正常运作；不过若未执行上传的动作时，此区域的各个字段便会显示「未知」，且当上传回来的站号与网络配置的站号不一致时，该字段便会呈现红色以提醒用户。



● 最大区块大小

PLC Link 的数据交换是以区块为单位，且一个 PLC Link 的规划中，依据机种的不同，至多可设定 16 组或 32 组的数据交换区块，而此处便是设定每个交换区块的最大数据长度；根据指定为主站机种的不同，其可设定的最大数据长度也会有所差异，设定时请先查阅各机种的相关说明。



相关设定	说明	DVP	AH5x0
16 Words	单笔数据交换区块最大长度 16 Words。只能使用 D 装置。	V	V

相关设定	说明	DVP	AH5x0
100 Words	单笔数据交换区块最大长度 100 Words。只能使用 D 装置。	V	V
450 Words	单笔数据交换区块最大长度 450 Words。且可以使用 M、D 或 L 装置，主站与从站的装置类别无须相同。		V

● RS485 通讯保持

DVP 系列的机种在断电重开后，所有的通讯参数将可能会被恢复为默认值，勾选「RS485 通讯保持」的选项后，主机便会在由 STOP 切换至 RUN 时，重新将保存的通讯参数载回；关于 DVP 系列机种的串行端口通讯参数，请参考各机种的操作手册。

● 同步读写

一般的状况下，主站对从站发出的写入要求与读取要求命令会分开传送，但若勾选「同步读写」的选项时，利用特定的 MODBUS 功能码（16 进制码为 17），主站便可在一次命令当中完成读与写的动作，如此便可提高数据交换的效率；不过设定时，请确认参与数据交换的所有装置都有支持同步读写的 MODBUS 功能码，否则便可能在接收到主站的命令后，会因无法识别该功能码而造成读写失败。除非选择「450 Words」的模式，否则启动同步读写时仅可选择 D 装置。

● 下载后启动 PLC Link

若有勾选此项目，当完成 PLC Link 的规划，并将其下载至主机之后，便会同时启动 PLC Link 的动作。

● 命令间隔时间

此处用于设定主站每次发出命令的时间间隔。



完成此页的设定后，按下  即可进入下一步骤；而按下  则可回至上一个步骤。

20.3.4 设定数据交换表 (步骤 3)

20.3.4.1 数据交换表简介

下列即为 PLC Link 的数据交换表，当 PLC Link 开始执行后，主站装置便会依据数据交换表的设定，轮流向各个从站进行数据读写的动作。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置类型
1	2	读	D0~D99	<=	D100~D199	100	启动	CPU530-RS2
		写	D0~D99	=>	D200~D299	100		
2	3	读	D100~D199	<=	D100~D199	100	启动	CPU530-RS2
		写	D100~D199	=>	D200~D299	100		
3	4	读	D200~D299	<=	D100~D199	100	关闭	CPU530-RS2
		写	D200~D299	=>	D200~D299	100		
4	0	读	D300	<=	16#1000	0	关闭	Unknown
		写	D300	=>	16#1000	0		
5	0	读	D400	<=	16#1000	0	关闭	Unknown
		写	D400	=>	16#1000	0		

表格字段	说明
序号 (#)	数据交换区块的序号，且依据机种的不同，至多可设定 16 组或 32 组的区块。
从站站号	数据交换区块属于的站号，同一个站号可规划为多组不同区块的从站。若此处显示为 0 则是代表未定义的从站，而非广播站号。
读/写	「读」：表示此列所显示的是主站读取从站的数据区块。 「写」：表示此列所显示的是主站写入从站的数据区块。
主站装置地址	主站于此组区块中所使用的装置地址范围。
<=>	「<=」：表示此列所显示的是主站读取从站的数据区块。 「=>」：表示此列所显示的是主站写入从站的数据区块。
从站装置地址	从站于此组区块中所使用的装置地址范围。当此组区块的从站为自行定义的「MODBUS Device」装置时，此范围会以 16 进制的地址编号来显示。
长度	数据交换区块的长度
状态	设定是否启动此区块的数据交换功能，且当此区块有任何的错误设定时，状态便会强制切为「关闭」，而当状态为「关闭」时，该笔数据的表格底色便会呈现灰色。
装置型态	显示数据交换区块的硬件装置名称。

下图便是一个数据交换表的范例，其中启动 PLC Link 功能的共有#1、#2、#4与#5共4组区块，而其数据交换的规划则请参考下表。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置类型
1	2	读	D3000~D3099	<=	D2500~D2599	100	启动	SV
		写	D3100~D3149	=>	D2600~D2649	50		
2	2	读	D3000~D3009	<=	D3000~D3009	10	启动	SV
		写	D3200~D3209	=>	D3200~D3209	10		
3	0	读	D0~D99	<=	16#0000~16#0063	100	关闭	Unknown
		写	D0~D99	=>	16#0000~16#0063	100		
4	3	读	D3400~D3424	<=	16#1000~16#1018	25	启动	MODBUS Device
		写	D3500~D3524	=>	16#1025~16#103D	25		
5	6	读	D3600~D3699	<=	D3000~D3099	100	启动	CPU530-RS2
		写	D3700~D3799	=>	D3100~D3199	100		

区块序号	说明
#1	主站会读取 2 号从站 (SV 机种) 的 D2500~D2599 共 100 Words 的数据，并将其存放于主站的 D3000~D3099，同时也会将主站本身的 D3100~D3149 共 50 Words 的数据写入从站的 D2600~D2649 当中。
#2	主站会读取 2 号从站 (SV 机种) 的 D3000~D3009 共 10 Words 的数据，并将其存放于主站的 D3000~D3009，同时也会将主站本身的 D3200~D3209 共 10 Words 的数据写入从站的 D3200~D3209 当中。
#4	主站会自 3 号从站 (MODBUS Device) 的寄存器地址 16#1000~16#1018 当中读取共 25 Words 的数据，并将其存放于主站的 D3400~D3424 当中，同时也会将主站本身的 D3500~D3524 共 25 Words 的数据写入从站的 16#1025~16#103D 寄存器当中。
#5	主站会读取 6 号从站 (CPU530-RS2 机种) 的 D3000~D3099 共 100 Words 的数据，并将其存放于主站的 D3600~D3699，同时也会将主站本身的 D3700~D3799 共 100 Words 的数据写入从站的 D3100~D3199 当中。

20.3.4.2 设定区块

在要设定数据交换区块的行列上双击鼠标左键即可开启设定窗口。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置类型
1	2	读	D3000~D3099	<=	D2500~D2599	100	启动	SV
		写	D3100~D3149	=>	D2600~D2649	50		
2	2	读	D3000~D3009	<=	D3000~D3009	10	启动	SV
		写	D3200~D3209	=>	D3200~D3209	10		
3	0	读	D0~D99	<=	16#0000~16#0063	100	关闭	Unknown
		写	D0~D99	=>	16#0000~16#0063	100		
4	3	读	D3400~D3424	<=	16#1000~16#1018	25	启动	MODBUS Device
		写	D3500~D3524	=>	16#1025~16#103D	25		
5	6	读	D3600~D3699	<=	D3000~D3099	100	启动	CPU530-RS2
		写	D3700~D3799	=>	D3100~D3199	100		

底部按钮：导入 | 导出 | 重置 | 检查设置 | 上传 | 下载 | 下载并监控 | 完成



参数设置

链接装置

站号: 机种:

链接状态

关闭

启动

读取

<p>主站参数设置</p> <p>起始位置: <input type="text" value="D 3000"/> 数据长度: <input type="text" value="100 Words"/></p>	←	<p>从站参数设置</p> <p>起始位置: <input type="text" value="D 2500"/></p>
---	---	--

写入

<p>主站参数设置</p> <p>起始位置: <input type="text" value="D 3100"/> 数据长度: <input type="text" value="50 Words"/></p>	→	<p>从站参数设置</p> <p>起始位置: <input type="text" value="D 2600"/></p>
--	---	--

确定 取消

● 链接装置

「站号」字段是指定此组数据交换区块所属的从站，不可选择主站站号但可以选择在别的数据交换区块用过的从站站号，如果选择 0 则右侧的「机种」字段会显示「Unknown」表示没有定义，否则会显示从站站号对应的硬件装置名称。

链接装置

站号: 机种:

链接状态

关闭

启动

● 链接状态

设定是否启动此组数据交换区块的数据交换功能。

链接装置		链接状态
站号	机种	<input type="radio"/> 关闭
4	CPU530-RS2	<input checked="" type="radio"/> 启动

● 「读取」

两边的装置类型必须相同，数据长度为要交换的笔数，将「从站参数设定」的装置起始位置算起的笔数，读取到「主站参数设定」的起始位置算起的笔数。

读取		
主站参数设定	从站参数设定	
起始位置	数据长度	起始位置
D	3000	100 Words
		D
		2500

● 「写入」

两边的装置类型必须相同，数据长度为要交换的笔数，将「主站参数设定」的装置起始位置算起的笔数，写入到「从站参数设定」的起始位置算起的笔数。

写入		
主站参数设定	从站参数设定	
起始位置	数据长度	起始位置
D	3100	50 Words
		D
		2600

规划区块时，同一区块的数据可被写至多个不同的位置中，但却不允许不同区块的数据存放于同一位置；以上图为例，主站 D3100~D3149 的内容可被重复写至不同的从站，但主站的 D3000~D3099 这个区块却不能再接收其它从站的数据。

20.3.4.3 数据交换表的机种同步性

在数据交换表中的「装置型态」字段，其内容会依据该站号所对应的主机型号来显示，而对应的依据则是目前 **NWCONFIG** 图标工作区中的配置；同一个主机有多个站号的时候，其「装置型态」显示的都会是该主机型号。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置型态
1	2	读	D0~D99	<=	D100~D199	100	启动	SV
		写	D0~D99	=>	D200~D299	100		
2	3	读	D100~D199	<=	16#1000~16#1063	100	启动	MODBUS Device
		写	D100~D199	=>	16#1000~16#1063	100		
3	4	读	D200~D299	<=	D100~D199	100	启动	CPU530-RS2
		写	D200~D299	=>	D200~D299	100		
4	5	读	D300~D399	<=	D100~D199	100	启动	CPU530-RS2
		写	D300~D399	=>	D200~D299	100		
5	6	读	D400~D499	<=	D100~D199	100	启动	CPU530-RS2
		写	D400~D499	=>	D200~D299	100		



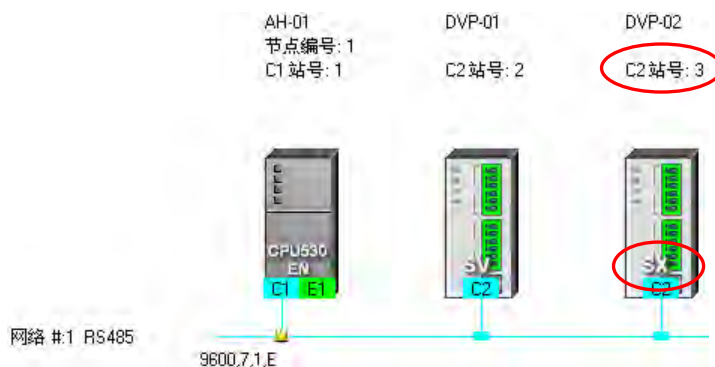
当已完成规划并将所设定的数据储存后，有变更机种的部分会以红色的字体来提醒用户，且状态则会自动切换为「关闭」。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置型态
1	2	读	D0~D99	<=	D100~D199	100	关闭	DMV1000
		写	D0~D99	=>	D200~D299	100		




当主站装置为 AH5x0 系列的机种时，执行上传后若配置与 **NWCONFIG** 目前的配置不符时，有变更机种的部分会以红色的字体来提醒用户，且状态则会自动切换为「关闭」。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置型态
2	3	读	D100~D199	<=	16#1000~16#1063	100	关闭	SX
		写	D100~D199	=>	16#1000~16#1063	100		



而当主站装置为 DVP 系列的机种时，因 DVP 系列的主机内部不会储存数据交换表中的装置类型，因此上传回来的数据将无法比对机种是否一致，所以「装置型态」的字段仍会依据图示工作区中的机种名称来显示，但如果存在不合理的设定时，其状态则会自动被切换为「关闭」。

20.3.4.4 数据交换表的管理操作

在数据交换表的下方有提供一些操作按钮，而关于各个按钮的功能则请参考下表的说明，其中若按下画面中的  便可回至上一个步骤。

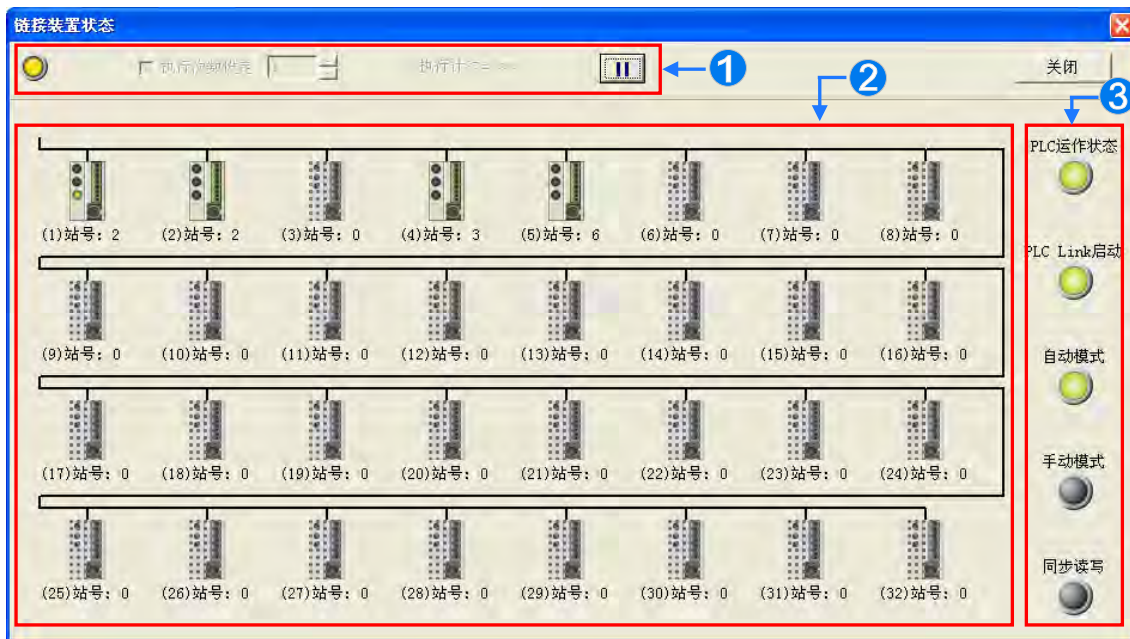
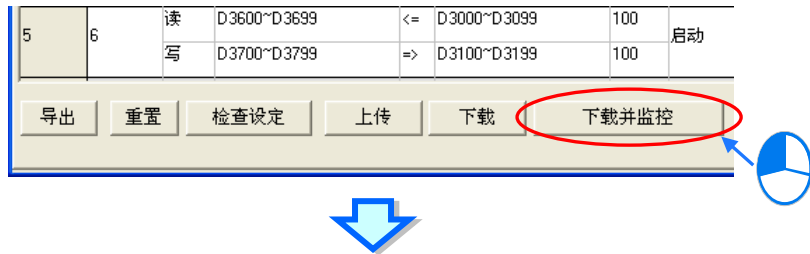


操作项	功能说明
导出	将数据交换表汇出为 CSV 档。
重置	将数据交换表的内容重置为初始状态。
检查设定	检查建立的数据交换表是否正确。
上传	上传主站的数据交换表。
下载	下载数据交换表至主站。
下载并监控	下载数据交换表至主站，并在下载完毕后自动开启监控窗口。
完成	关闭窗口前系统会主动询问是否要储存所做的变更。

在 PLC Link 执行上下载或监控操作前，请务必确认实际联机的装置是否正常。

20.3.5 PLC Link 的监控功能

进行监控前，请先确认系统与主机正常联机，接着按「下载并监控」便可开启 PLC Link 的监控窗口；主站主机必须处于「RUN」的状态下，PLC Link 的操作才可正常执行。



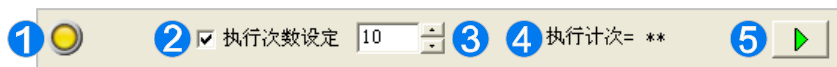
- ❶ 功能操作区：对 PLC Link 的数据交换功能进行操作。
- ❷ 图标监控区：显示主站对每个从站的读写状态，并且可于此处进行在线的数据读写。
- ❸ 状态灯号区：灯号显示目前的 PLC Link 状态。各灯号的意义请见下表。

灯号	说明
PLC 运作状态	亮起表示 PLC 主机处于 RUN。
PLC Link 启动	亮起表示 PLC Link 启动。
自动模式	亮起表示 PLC Link 的执行模式为「自动模式」。
手动模式	亮起表示 PLC Link 的执行模式为「手动模式」，而当执行次数到达「轮询次数」之后，灯号便会熄灭。
同步读写	亮起表示启动同步读写的功能。

● 功能操作区

对 PLC Link 的数据交换功能进行设定或操作，此处所做的操作都会直接改写主站主机内部所对应的

特殊寄存器与特殊继电器，所以离开 PLC Link 的监控功能之后，PLC Link 的运作仍会保持在最后操作的状态，因此在离开监控画面前请务必确认目前的 PLC Link 状态是否正确。



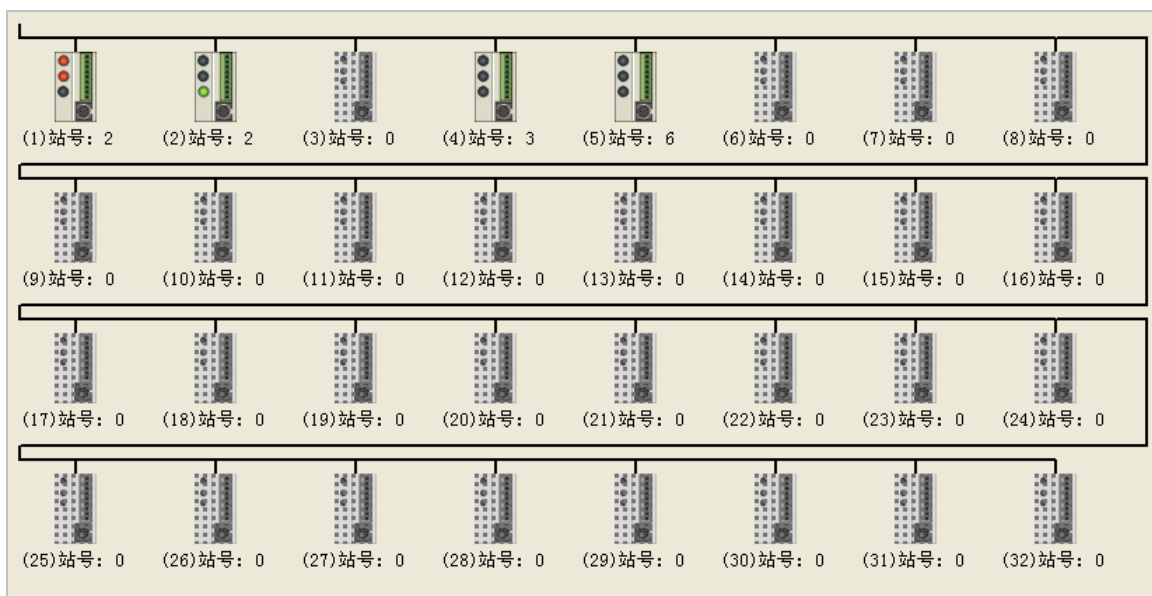
- ❶ 当灯号闪烁时就是指计算机与 PLC 主机保持联机。
- ❷ 勾选此项时，会依照❸所设定的「轮询次数」执行，最大为 65535。
- ❸ 若有勾选❷，开始执行之后，会显示目前已执行的次数；而计算的方式则是对所有从站完成一遍的数据读写。
- ❹ 启动或停止监控的运作。

当在监控的过程中发生联机中断时，窗口的右上角便会出现「恢复监控」的按钮；在排除问题之后，按下此按钮便可重新进行联机。

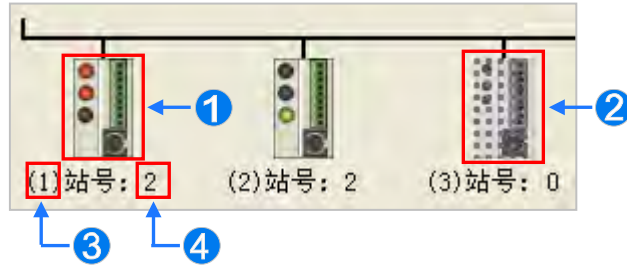


● 图标监控区

PLC Link 在进行运作时，主站主机会依据规划而轮流对各区块所指定的从站进行读写，且通过监控区中的图标，用户便可得知主站主机对每个区块进行读写时的状态。



图标监控区中的图标代表的是主站的读写情况，而非从站装置本身的状态。图标的表达说明如下。

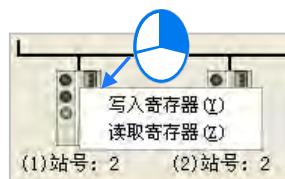


- ❶ 区块读写状态的图标。通过图标上的灯号便可得知此区块的读写状态。
- ❷ 图标代表该区块未定义 PLC Link 的功能，或功能状态为关闭。
- ❸ 号码对应的是数据交换表中的「序号」字段，代表区块的序号。
- ❹ 号码对应的是数据交换表中的「从站站号」字段，代表该区块所指定的从站站号。

AH5x0 系列与 DVP 系列机种的 PLC Link 错误标志有些许差异，因此显示会有所不同，请参考下表的说明。

主站机种	区块图标	说明
AH5x0 系列		<ul style="list-style-type: none"> ❶ 亮起表示读取从站数据发生错误时。 ❷ 亮起表示数据写入从站发生错误时。 ❸ 亮起表示正在进行此区块的数据交换。
DVP 系列		<ul style="list-style-type: none"> ❶ 亮起表示从站进行数据读写发生错误。 ❷ 亮起表示正在进行此区块的数据交换。

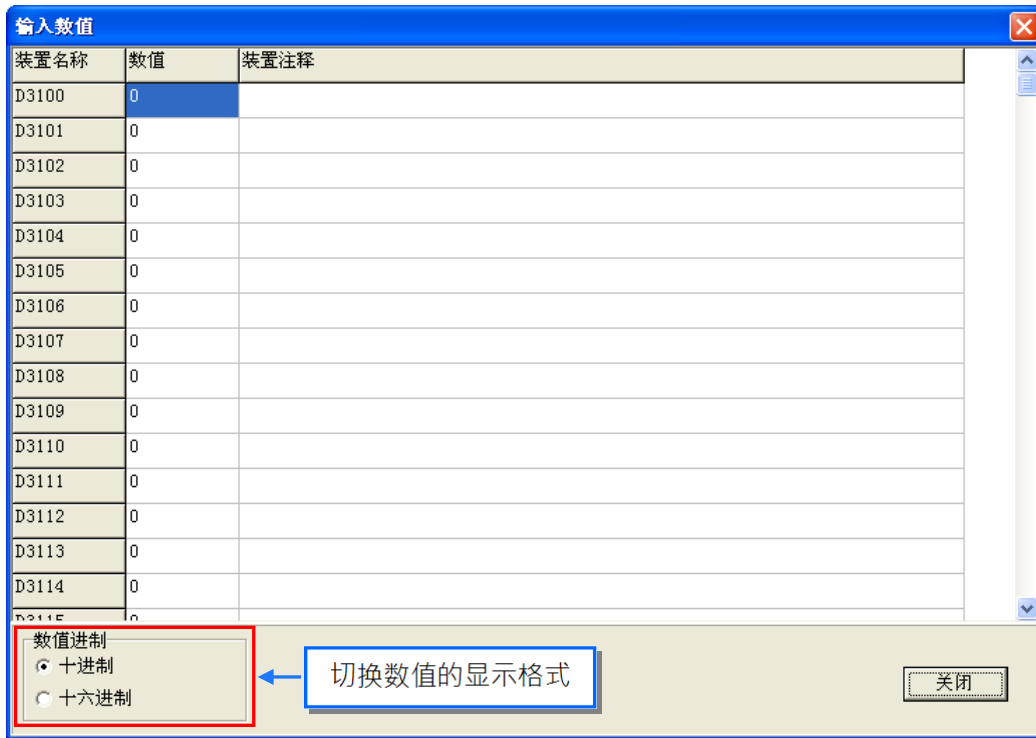
點選某一区块的图标后按下鼠标右键，接着便可于快捷菜单中选取要进行在线编辑的寄存器类别。



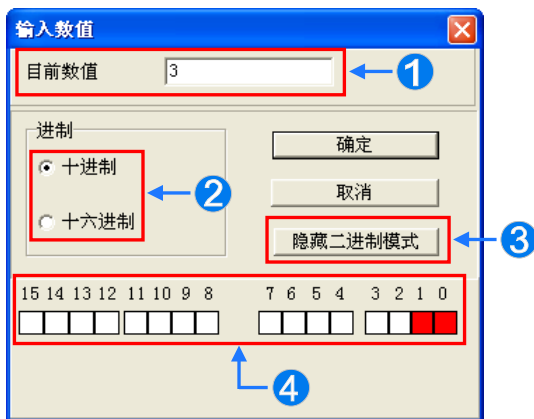
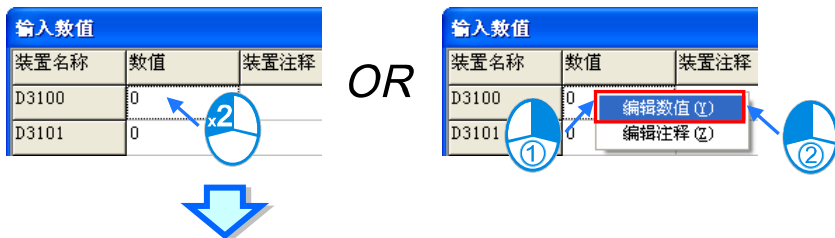
- **写入寄存器**：指主站中要写入从站数据的装置位置，如下图的 D3100~D3149。
- **读取寄存器**：指主站中要读回从站数据的装置位置，如下图的 D3000~D3099。

#	从站站号	读/写	主站装置地址	<=>	从站装置地址	长度	状态	装置型态
1	2	读	D3000~D3099	<=	D2500~D2599	100	启动	SV
		写	D3100~D3149	=>	D2600~D2649	50		

当选取完寄存器类别后，画面便会开启对应该区块的寄存器监控表。



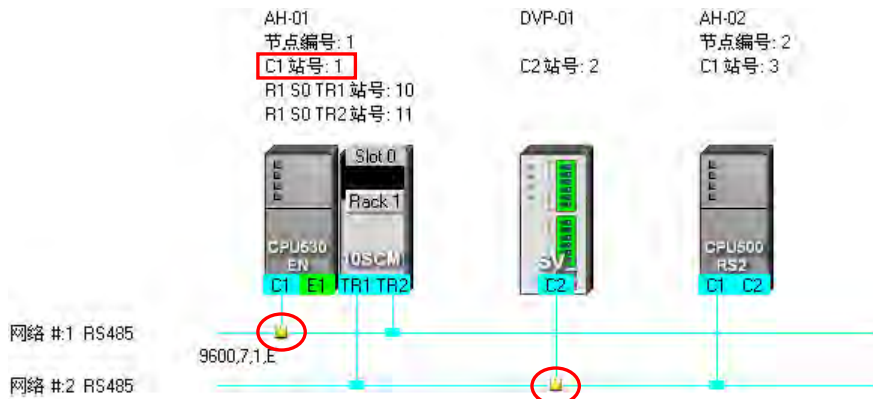
在某个装置的「数值」字段上双击鼠标左键，或是按下右键后于快捷菜单中点击「编辑数值 (Y)」，接着便可于数值输入窗口中设定该寄存器的内容值；而于「装置注释」字段上双击鼠标左键，或是于右键的快捷菜单中点选「编辑注释 (Z)」时便可为该装置加上注释，不过此处的注释为 PLC Link 专用的信息，与 ISPSOft 编程中的装置批注无关。



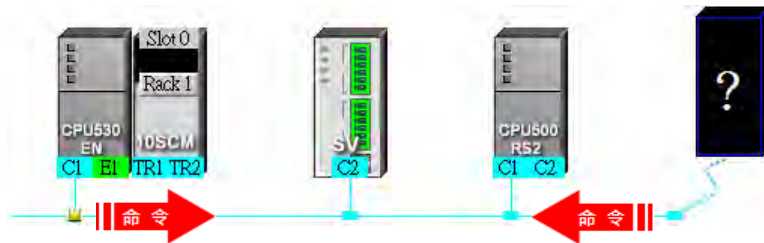
- ① 于此字段输入要设定的数值。
- ② 此字段可切换输入数值时的格式。
- ③ 开启或隐藏二进制的编辑功能。
- ④ 开启二进制的编辑功能后，于此处便可以二进制的方式设定数值。

20.3.6 设定 PLC Link 的注意事项

规划 PLC Link 时，同一个 RS485 的网络上只能有一个主站，而当设为主站的节点有多个串行端口时，该节点就不允许再成为其他 RS485 网络上的主站，只能当从站；如下图的 **AH-01**，该节点的套接口 **C1** 为**网络#1** 上的主站，而通过模块的 **TR1** 端口就不能被指定为**网络#2** 的主站，只能当从站；**AH-01** 当中的 C1 与 TR2 都被连接至**网络#1**，且因两个端口的站号是彼此独立的，因此在**网络#1** 的 PLC Link 规划中，TR2 被指定为从站的状况将是被允许的。



一个网络同时间只可以有一台主站发出命令，因此在网络规划完毕之后，请务必确认实际的接线与 NWCONFIG 中的规划一致或兼容，避免产生类似下图的错误。



当重新定义主站时，若原主站的 PLC Link 参数未重新设定，便可能形成同时有两个主站发出读写命令而产生错误；因此当要变更主站时，请务必确认原主站的 PLC Link 动作已停止。



* 除了上述状况外，任何同时间有两个或以上装置发出读写命令的状况都应该避免。

20.4 Ether Link 的建构

20.4.1 Ether Link 简介

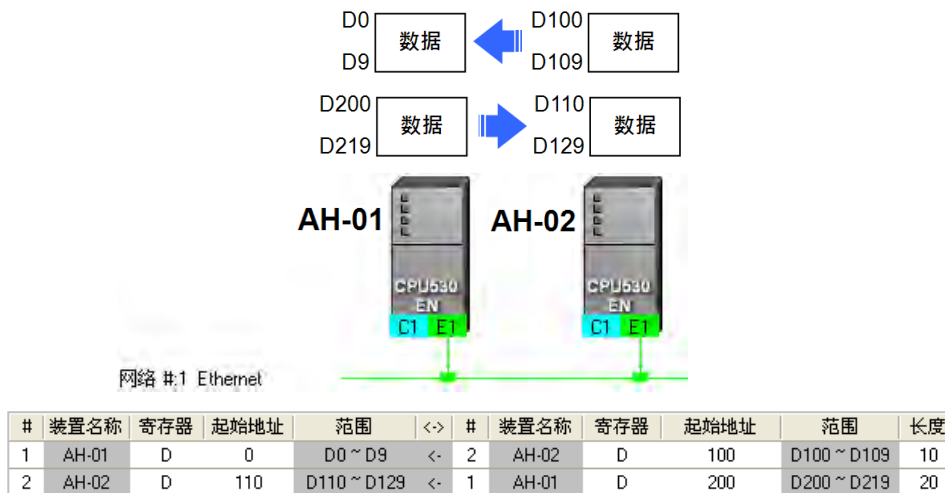
Ether Link 为通过以太网 (Ethernet) 的联机来进行数据交换的网络机制。当某个 Ethernet 网络上存在多个节点时，我们便可于该网络中规划一个数据交换的机制；之后当设定参数被下载至各个节点的主机之后，根据设定的运作方式，系统便可自动进行数据交换的动作。

于 PLC Link 架构中，数据交换的操作是着重在主站的读取与写入，而相较于 PLC Link 的主从架构，Ether Link 则是采用较安全的**数据索取机制**，其操作则是以网络上的**数据需求节点**为主。

在 Ether Link 的架构下，网络上的每个节点都可以成为**数据需求节点**，并主动的发出需求向其它节点索取数据，但却必须被动的在接收到其它节点的需求之后，才可以将数据回传给提出需求的节点，因此在这样的架构下，并不会会有节点主动或强制的将数据写入其它装置的情况。基于 TCP/IP 的通讯协议，所有的**数据需求节点**都可随时提出索取数据的需求，系统本身则会自动安排数据报文的传送机制，因此在进行数据交换时，这样的机制便可提高执行的效率。

请参考下图的范例，当要建构如下的数据交换关系时，规划上便必须针对两个**数据需求节点**，分别为其建立各自的需求对象于**数据交换表**中。

- (a) **AH-01** 向 **AH-02** 索取 D100~D109 的数据，并将取得的数据存放于 **AH-01** 的 D0~D9。
- (b) **AH-02** 向 **AH-01** 索取 D200~D219 的数据，并将取得数据存放于 **AH-02** 的 D110~D129。




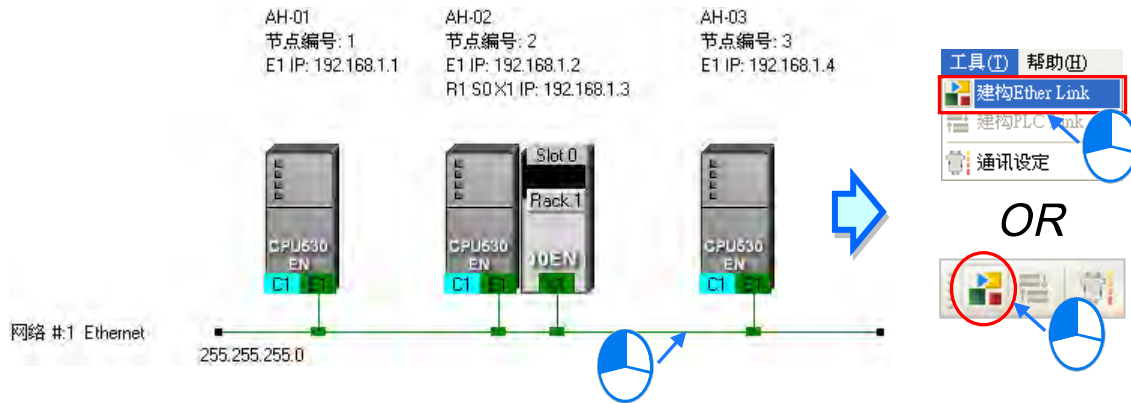
Ether Link 的设定是以网络为单位，而当某个节点被连接至不同编号的网络时，设定上便可为每个网络各自设定一组 Ether Link 的参数，因此同一个主机内部可能会保存多组的 Ether Link 的参数；且因 Ether Link 并非如同 PLC Link 一般的主从架构，因此每个节点在不同编号的网络上都可以同时成为**数据需求节点**与**数据提供节点**；例如以上图的第一笔数据来说，**AH-01** 是**数据需求节点**，而 **AH-02** 则是**数据提供节点**；而以第二笔数据来说，**AH-01** 是**数据提供节点**，而 **AH-02** 则是**数据需求节点**。

20.4.2 开启 Ether Link 的设置窗口

建构 Ether Link 之前请先确认所有的网络部署设定正确无误后，依照下述方法进行操作。

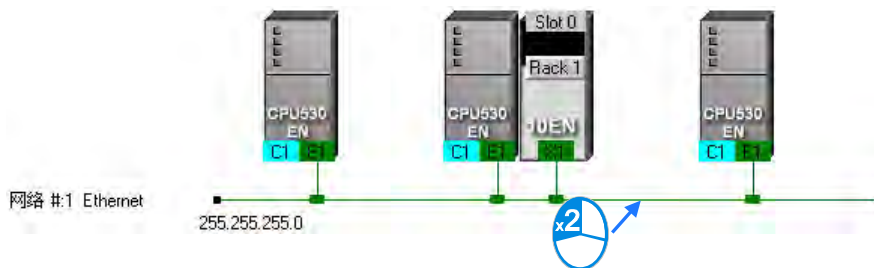
● 方法一

点选要建构 Ether Link 的网络后，于功能工具栏中点选 **工具 (I) > 建构 Ether Link**，或直接点击图标工具栏中的  图标。



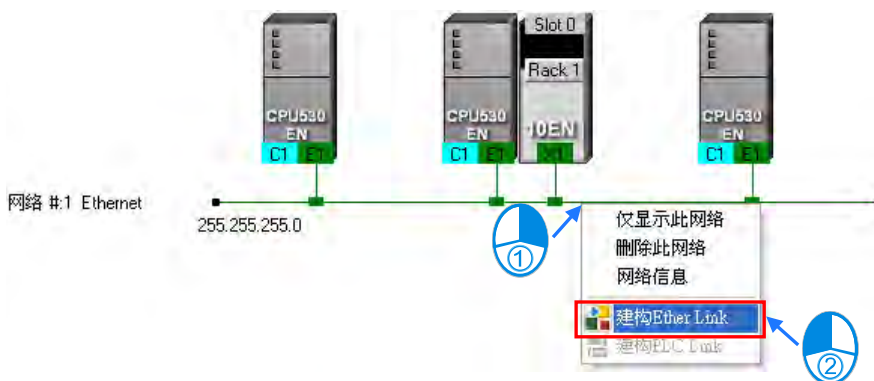
● 方法二

于要建构 Ether Link 的网络上双击鼠标左键。

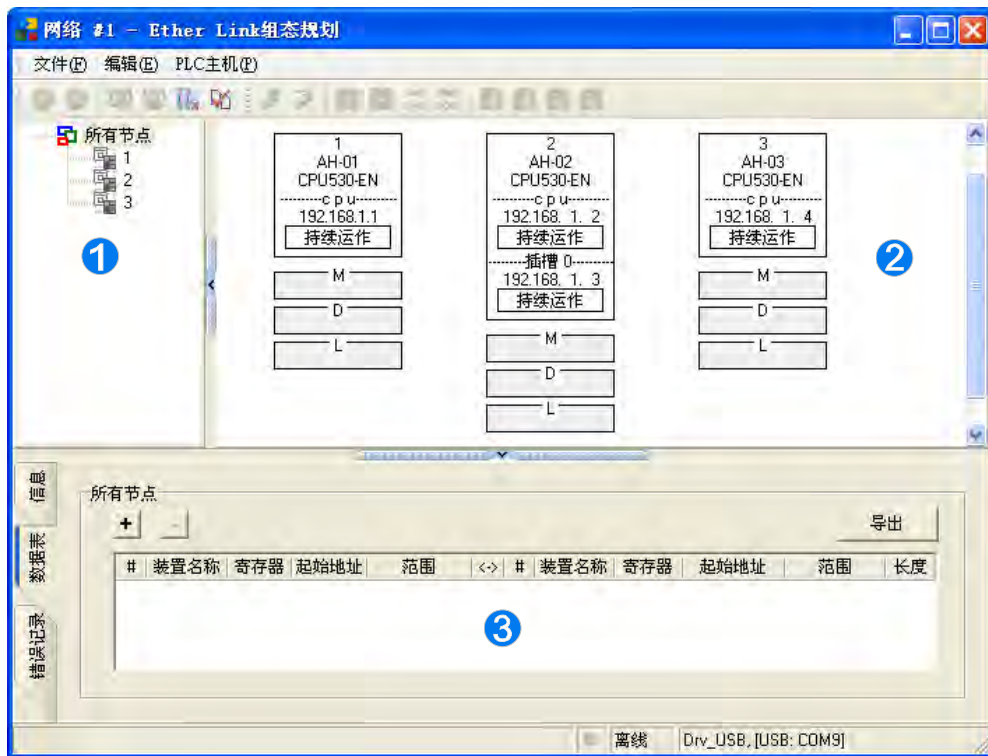


● 方法三

于要建构 Ether Link 的网络上点击鼠标右键，接着于快捷菜单中点选「建构 Ether Link」。

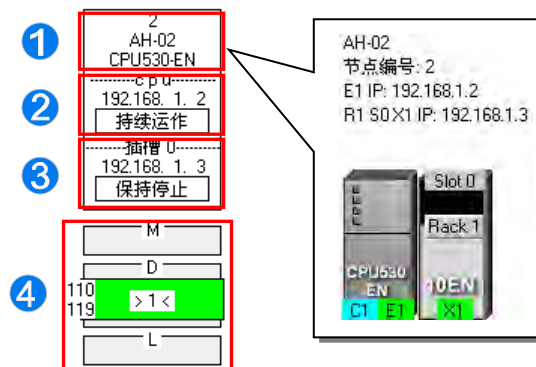


下图便是 Ether Link 的设置画面。



- ❶ 装置清单：于此选择某节点后，图标区便会显示该节点及被该节点提出需求的其它节点。
- ❷ 图标区：此区会以图标表达数据需求节点与数据提供节点的相关设定与信息。
- ❸ 信息窗口：此区包括数据交换表、操作信息以及装置的错误记录...等 3 个页面。

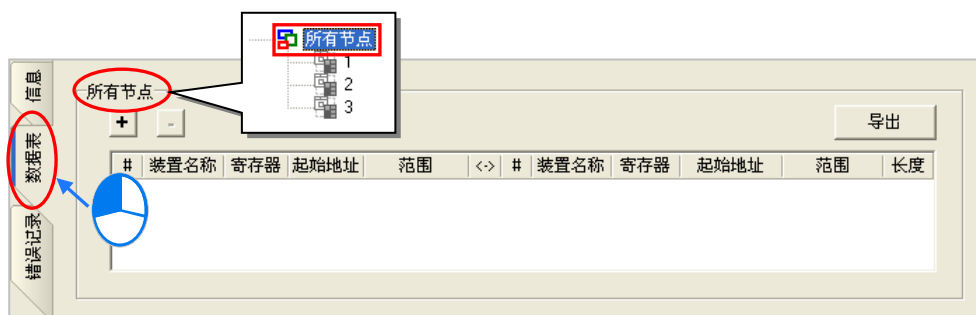
下列则为图标区中的节点图标与其标示的相关说明



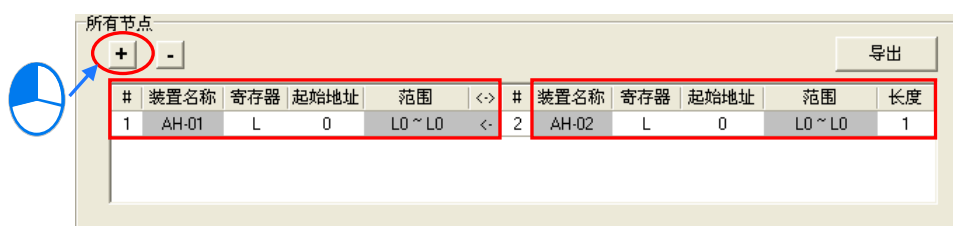
- ❶ 由上至下依序为节点编号、PLC 名称及機種型号。
- ❷ 分隔线的 **cpu** 字样，代表此处显示的是主机的 IP 地址与其 Ether Link 的启动模式。
- ❸ 分隔线的 **插槽 0** 字样，代表此处显示的是插槽 0 的模块 IP 地址与其 Ether Link 的启动模式。
- ❹ 此处显示的为数据交换区块的设定。上图 D 区块的 **>1<** 代表向节点 1 索取数据，而左侧的数字则代表取自节点 1 的数据将会存放于节点 2 的 D110~D119。至于区块的颜色则是依据节点编号随机配置；另外，M 与 L 区块无任何显示，代表该区块并无向其它装置索取数据。

20.4.3 数据交换表的建立与管理

点选「数据表」的选项卡即可切换至数据交换表的操作页面；其中在表格的左上方会显示目前装置清单中所选取的节点，而表格也只会显示以该节点为数据需求节点时的数据。

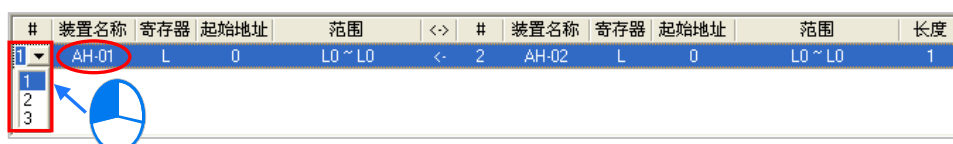


于表格上方点击 **+** 键即可新增一笔数据，且数据的内容可分为两个部份，左侧代表数据需求节点在取得数据后的存放位置；而右侧则代表要索取数据的对象（数据提供节点）及索取的数据范围。

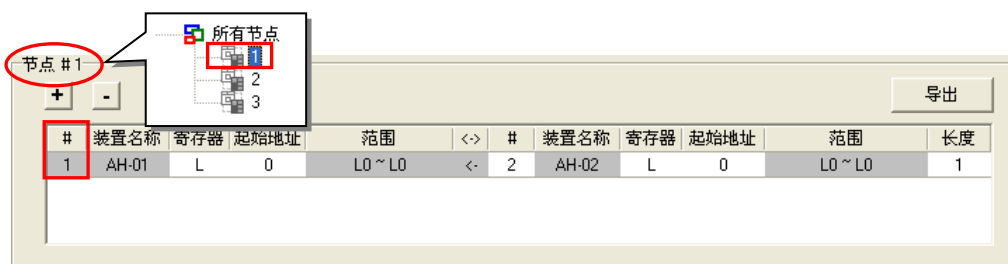


设定数据交换区块的步骤如下。

- (1) 于数据需求节点的「#」字段下拉选择节点编号，选择完毕后，「装置名称」字段便会显示对应该节点编号的 PLC 名称。



不过当目前装置清单中所选取的不是「所有节点」，而是特定的节点编号时，因表格仅会列出以该节点为数据需求节点时的数据，所以此字段将会固定为装置清单中所选取的节点编号。



- (2) 于**数据需求节点**的「寄存器」字段下拉选择装置类别，并于「起始地址」字段输入地址编号，以决定取得数据后的存放位置。

#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D500	<-	2	AH-02	L	0	L0 ~ L0	1

- (3) 于**数据提供节点**的「#」字段下拉选择节点编号，且选择时，需求节点与提供节点的编号不可相同，选择完毕后，「装置名称」字段也会显示对应该节点编号的 PLC 名称。

#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D500	<-	2	AH-02	L	0	L0 ~ L0	1

- (4) 再来则要设定要索取的数据区块。首先请于**数据提供节点**的「寄存器」字段下拉选择装置类别，而所选取的类别则无须与**数据需求节点**一致，接着再于「起始地址」字段输入地址编号即可。


#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D500	<-	2	AH-02	M	1000	M1000 ~ M1015	1

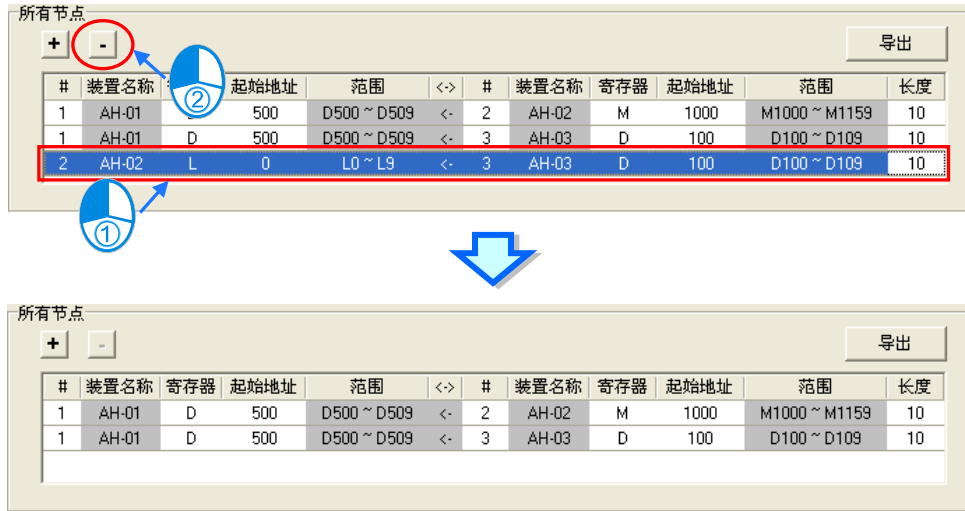
- (5) 最后再于「长度」字段设定要索取的数据长度（单位为 Word），且最大不得超过 1900；而设定完成后，于两侧的「范围」字段便会自动换算数据交换区块的范围。

#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10

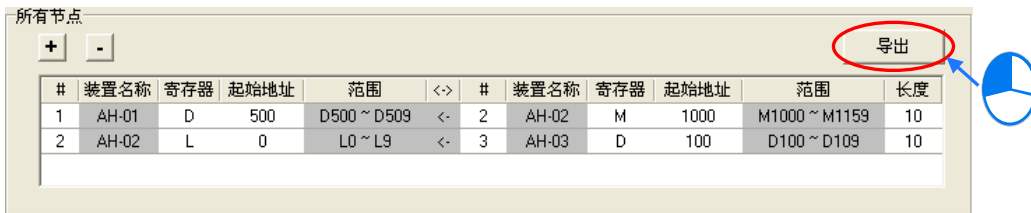
接下来便可依据上述的说明逐一建立每笔数据，但须注意所设定的数据区段中，**数据需求节点**的存放区段不得重迭，而**数据提供节点**则无特别限制；不同的需求方可要求同一区块的数据，但不可将不同的数据存放在同一区块。


#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
1	AH-01	D	500	D500 ~ D509	<-	3	AH-03	D	100	D100 ~ D109	10
2	AH-02	L	0	L0 ~ L9	<-	3	AH-03	D	100	D100 ~ D109	10

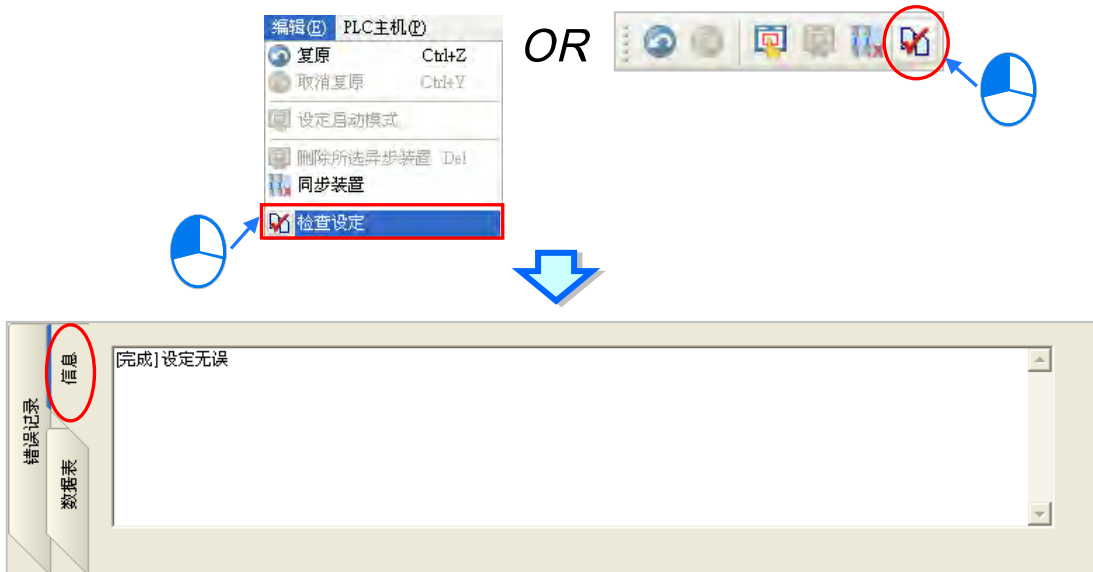
当在过程中在要删除某笔数据时，请先选取该行数据后按下  键即可。



按下表格右上角的「导出」键则可将数据交换表的内容导出为 **CSV** 文件。



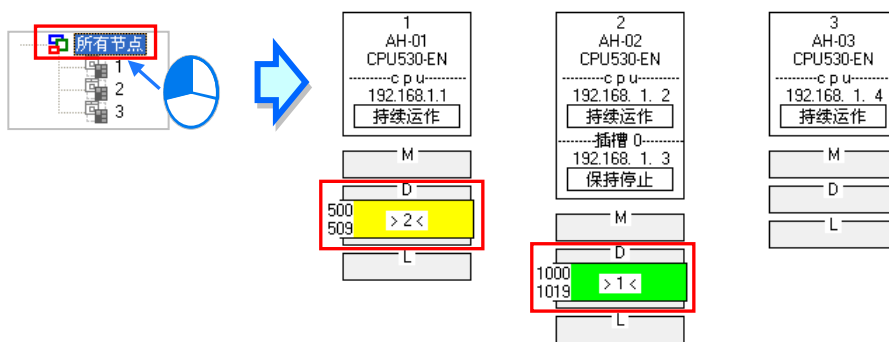
规划完数据交换表后，于功能工具栏中点选 **编辑 (E) > 检查设定**，或按下图标工具栏中的  图标即可对所设定的数据进行检查，而检查的结果将显示于下方信息窗口中的「信息」页面。



20.4.4 装置清单与图标区的操作

于**装置清单**点选「所有节点」时，**图标区**便会列出所有节点的图标，下方的**数据交换表**也会列出所有数据交换区块的设置，在各个节点图标的地址区块中，也会显示各节点用来存放索取数据的地址范围以及索取数据的对象（**数据提供节点**）。

如下图**节点 1**的图标，可以得知**节点 1**的 D500~D509 将会用来存放取得的数据，且由区块中的 >2< 可得知该数据是来自**节点 2**；同样的，我们也可由**节点 2**的图标得知其 D1000~D1019 的区块将会用来存放取自**节点 1**的数据。

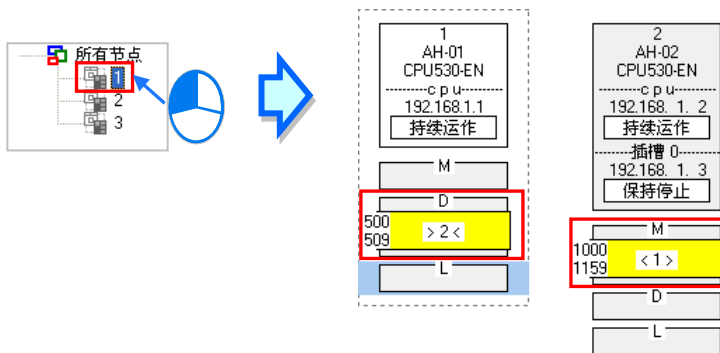


#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D509	< 2 >	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	< 1 >	1	AH-01	L	0	L0 ~ L19	20

而当于**装置清单**点选某个特定编号的节点时，**图标区**便只会列出该节点与被该节点索取数据的其它节点图标，而下方的**数据交换表**也仅会列出以该节点为**数据需求节点**时的数据。

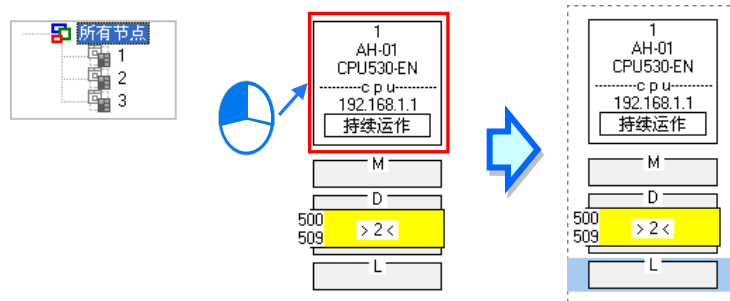
如下图，其中**节点 1**图标的外围有一个虚线的方框，代表目前**节点 1**处于选取状态，而**节点 2**的图标则以灰色底色来呈现，代表**节点 2**目前为**数据提供节点**；在这个状态中，**节点 2**的图标将无法选取，且**节点 2**图标下方的数据区块也只会显示提供数据给**节点 1**的部份。除此之外，因**节点 3**并未提供数据给**节点 1**，所以在这个状态下，**节点 3**的图标便不会显示。

由**节点 1**的图标中可得知，**节点 1**的 D500~D509 会存放取自**节点 2**的数据；而由**节点 2**的图标与图标区块中的 <1> 则可得知，**节点 2**会将本身 M1000~M1159 的数据提供给**节点 1**。

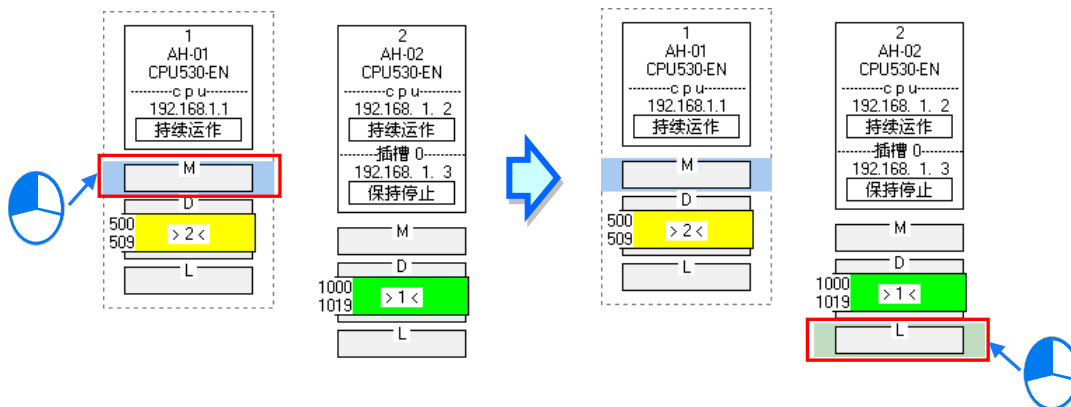


#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D509	< 2 >	2	AH-02	M	1000	M1000 ~ M1159	10

当在**装置清单**中选取某个特定的节点编号时，图标区中的该节点便会自动处于选取状态；但当**装置清单**中选取的是「**所有节点**」，且要在图标区中选取某个节点时，只要在该节点的节点信息处点击鼠标左键，接着便可将该节点选取起来；选取时须注意鼠标必须点击在节点信息的位置，若点击在下方的数据区块则无法将该节点选取起来，而被选取起来的节点图标外围便会有一个虚线的方框。



当某个节点被选取起来之后，表示要将该节点指定为**数据需求节点**，此时可先于该节点的下方点选装置类别，点选后该区块便会出现浅蓝色光标；接着再点选其它节点（**数据提供节点**）的装置类别，而点选后该区块则会出现浅绿色光标；须注意点选**数据提供节点**的装置类别时，只须直接点选该节点下方的区块，而不可将该节点整个选取起来，否则便等于重新指定**数据需求节点**。



完成之后，按下数据交换表 **+** 键便可新增一笔数据，并会直接带入**数据需求节点**与**数据提供节点**的节点编号与**寄存器类别**。

#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	<-	1	AH-01	L	0	L0 ~ L19	20



#	装置名称	寄存器	起始地址	范围	<>	#	装置名称	寄存器	起始地址	范围	长度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	<-	1	AH-01	L	0	L0 ~ L19	20
1	AH-01	M	0	M0 ~ M15	<-	2	AH-02	L	0	L0 ~ L0	1

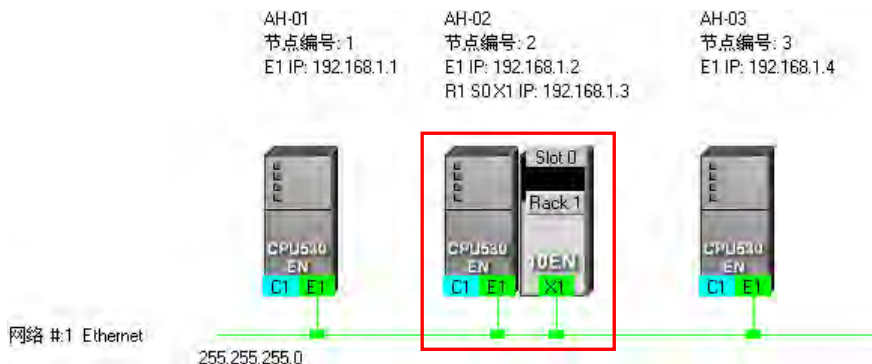
20.4.5 Ether Link 的运作模式

Ether Link 的启动模式共有三种，相关说明请参考下表。

运作模式	说明
保持停止	主机在运作的期间都不会进行数据交换的动作。
持续运作	一旦主机开始运行后，数据交换的动作便会持续进行。
标志模式	数据交换的动作将依据主机内部的特殊继电器状态来运行，而选择标志模式后，针对启动数据交换的特殊继电器，用户还可设定其初始状态。

*关于 Ether Link 的相关特殊继电器，请参考各机种的使用手册或相关技术文档。

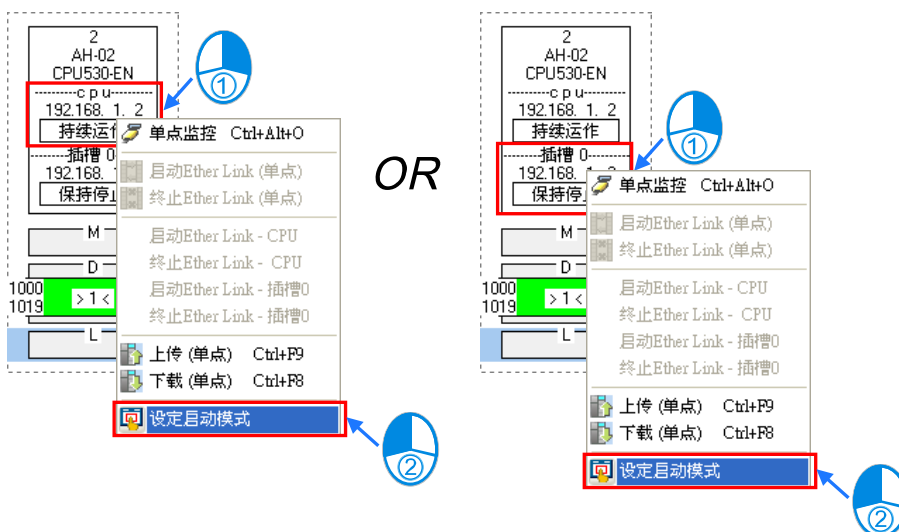
每组网络的 Ether Link 运作是以节点为单位，而当该节点有多个通讯端口被连接到同一个编号的网络上时，每个通讯端口的启动模式皆可分开设定；实际运作时，系统便会根据设定自动分配每个通讯端口的传送与接收，而当有某个通讯端口掉线时，系统也会自动将工作转移至该节点的其他通讯端口。



要设定某个节点的启动模式时，请参考下列的几种方式。

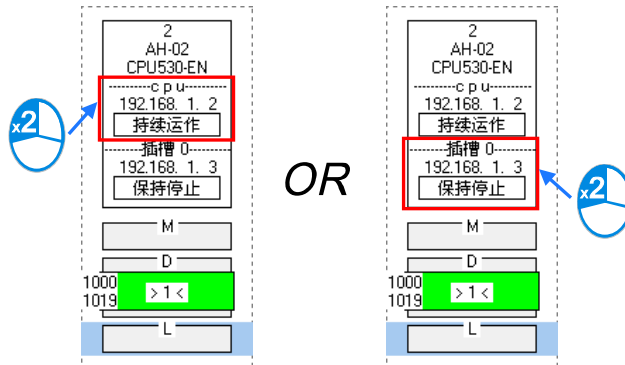
● 方法一

于 CPU 或模块信息区块上点击鼠标右键，接着于快捷菜单中选取「设定启动模式」。




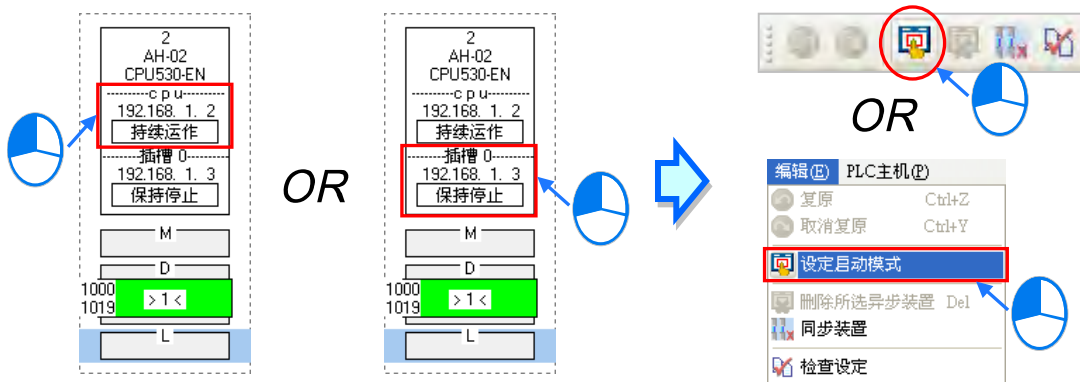
● 方法二

于要设定启动模式的 CPU 或模块信息区块上双击鼠标左键。



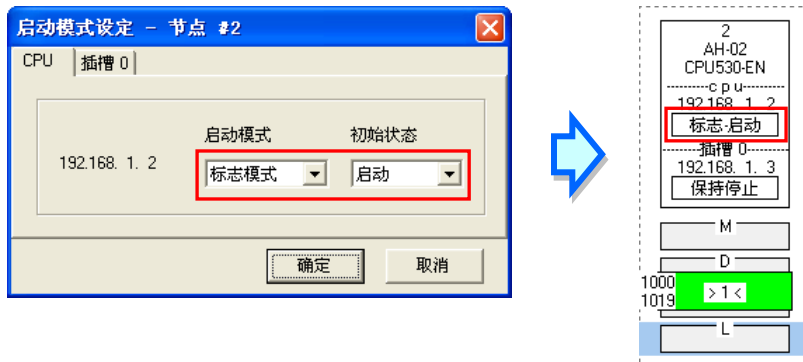
● 方法三

于要设定启动模式的 CPU 或模块信息区块上点击一下鼠标左键后，于图标工具栏中点选  图标，也或于功能工具栏中点选 编辑 (E) > 设定启动模式。



依据上述的方式操作之后，画面便会出现启动模式的设定窗口，而依据在节点图标上所点选的位置，开启的窗口也会自动切换至对应的页面，不过用户仍可通过上方的选项卡自行切换。

接着请直接在「启动模式」的字段下拉选择要设定的模式，而当选择的是标志模式时，请接着于「初始状态」的字段下拉选择对应特殊继电器的初始状态。完成后按下「确定」键即可。




20.4.6 下载 Ether Link 设定

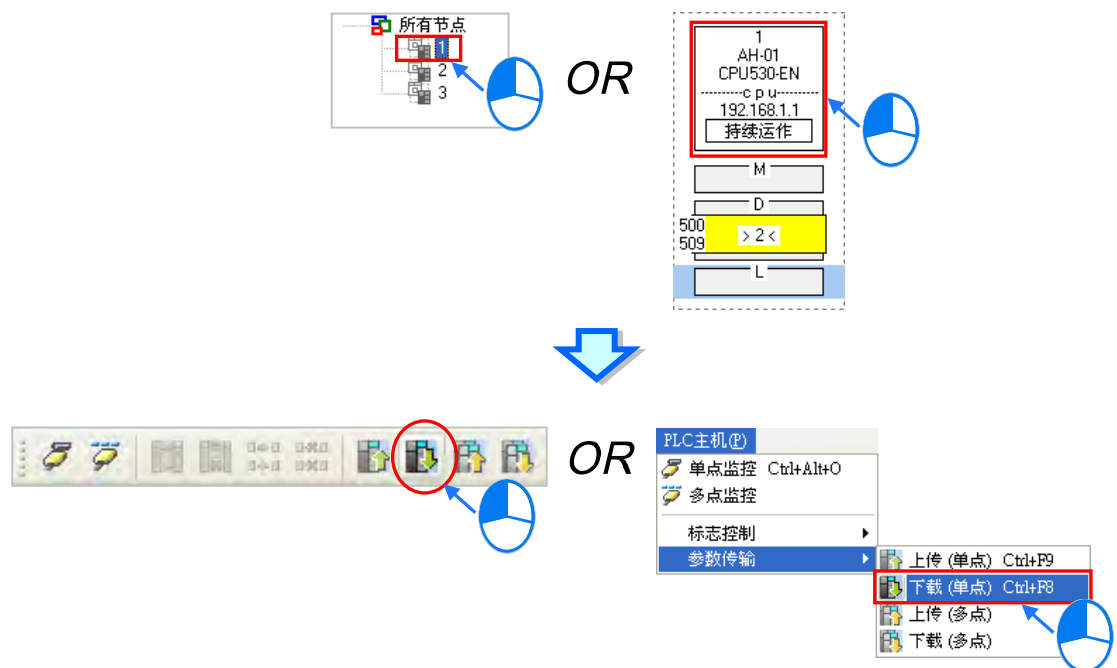
当完成所有的设定后，必须将 Ether Link 的参数下载至 PLC 主机当中，主机才可依据设定的参数进行数据交换的动作。

● 单点下载

仅下载所选主机的数据需求设定，并设定该主机与模块的启动模式；下载之前，请确认系统与主机或模块联机都正常。

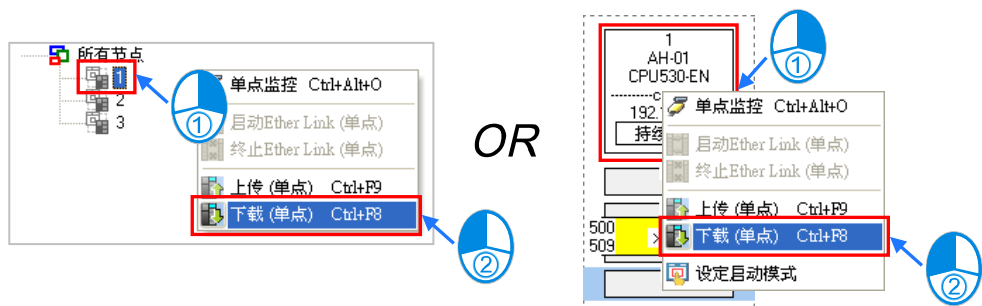
➤ 方法一

选取要下载参数的数据需求节点，之后于图标工具栏中点选  图标，或是于功能工具栏中依序点选 **PLC 主机 (P)** > **参数传输** > **下载 (单点)**。



➤ 方法二


于装置清单或图标区中，点选要下载参数的数据需求节点后按下鼠标右键，接着于快捷菜单中点选「下载 (单点)」。

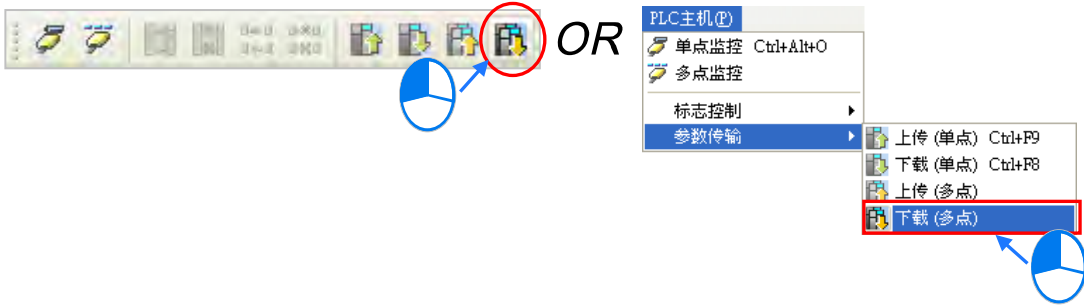


● 多点下载

下载所有节点的数据需求设定与启动模式；下载之前，请确认系统与主机或模块联机都正常。

➢ 方法一

于功能工具栏中点选 **PLC 主机 (P)** > **参数传输** > **下载 (多点)**，或于直接点击  图标。



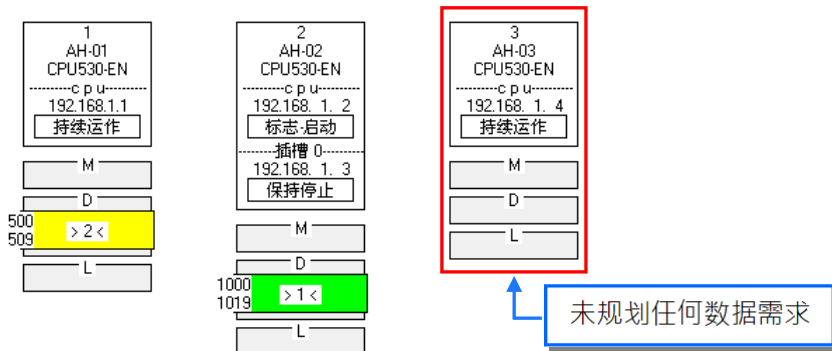
➢ 方法二

于装置清单中点选「所有节点」后按下鼠标右键，接着于快捷菜单中点选「下载 (多点)」。



补充说明

当设定的参数存在一个未规划任何数据需求的节点时，在进行多点下载后，该节点对应此组网络编号的 Ether Link 参数会被设定为无数据需求。




20.4.7 上传 Ether Link 设定

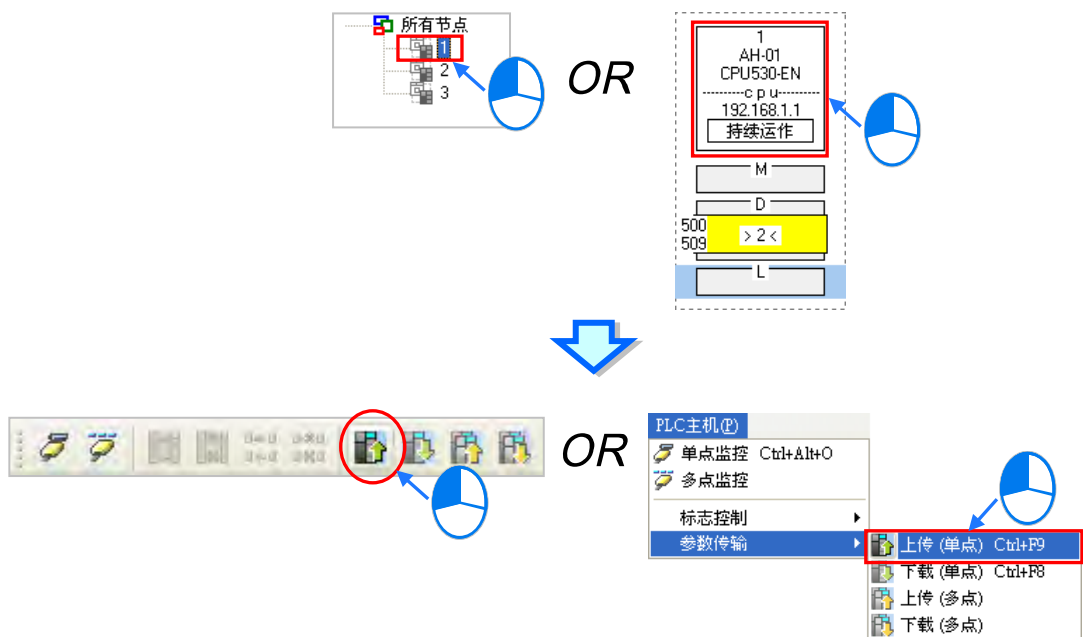
利用上传的功能便可重新取得该主机内部的 Ether Link 设定参数。

● 单点上传

只上传单个节点的 Ether Link 设置参数；上载之前，请确认系统与主机或模块联机都正常。

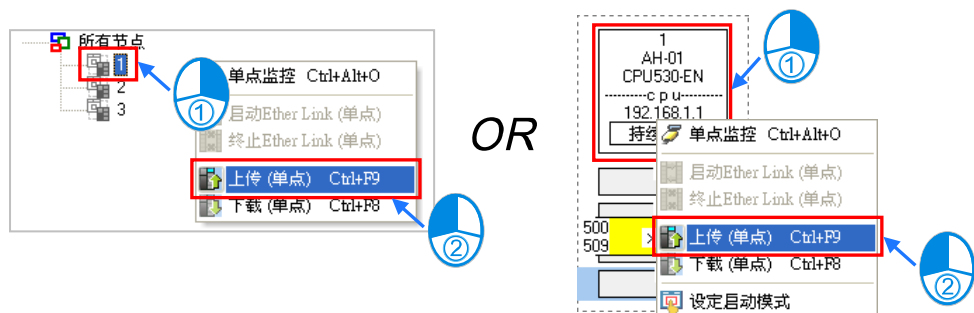
➤ 方法一

先选取要上传参数的数据需求节点，接着于图标工具栏中点选  图标，或是于功能工具栏中依序点选 **PLC 主机 (P)** > **参数传输** > **上传 (单点)**。



➤ 方法二


于装置清单或图标区中，点选要上传参数的数据需求节点后按下鼠标右键，接着于快捷菜单中点选「上传 (单点)」。

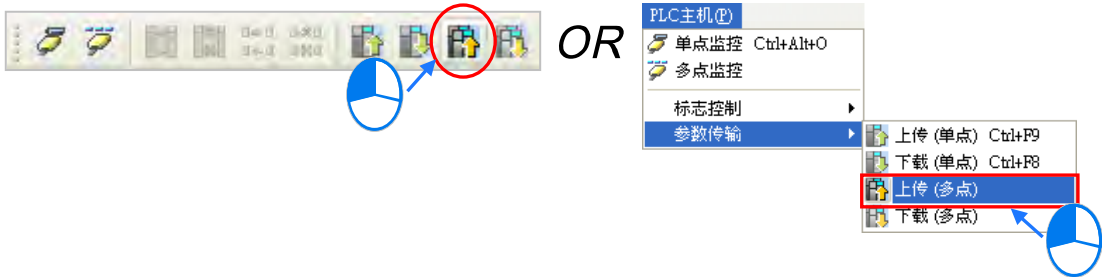


● 多点上传

上传所有节点的 Ether Link 设置参数；上载之前，请确认系统与主机或模块联机都正常。

➢ 方法一

于功能工具栏中点选 **PLC 主机 (P)** > **参数传输** > **上传 (多点)**，或于直接点选  图标。



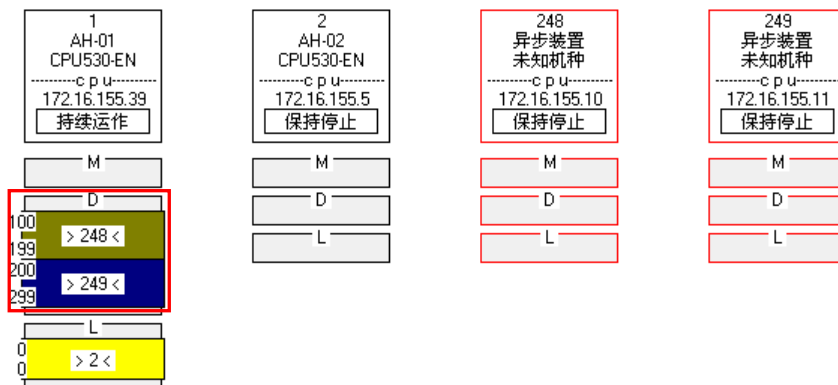
➢ 方法二

于装置清单中点选「所有节点」后按下鼠标右键，接着于快捷菜单中点选「上传 (多点)」。




补充说明

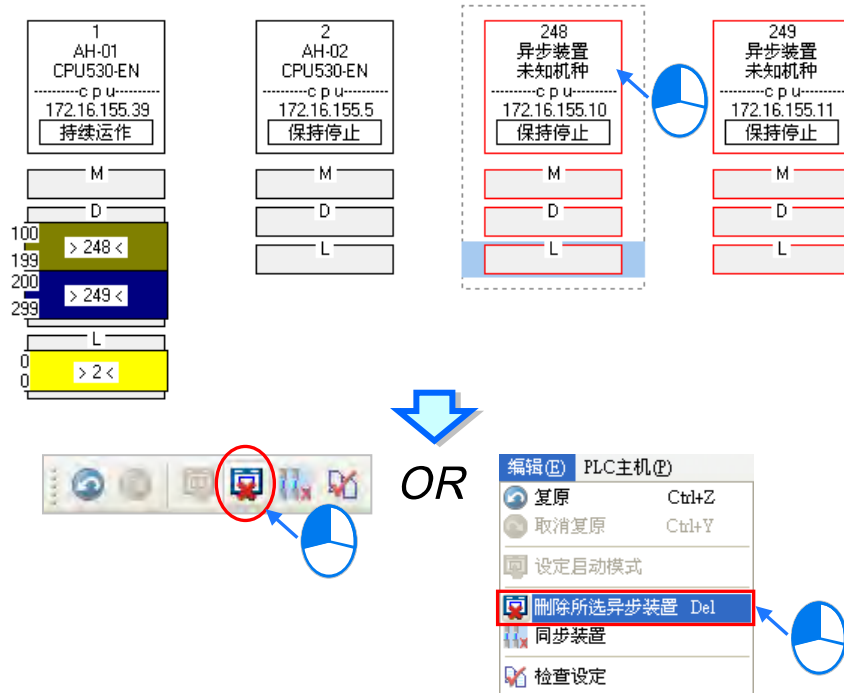
重新上传 Ether Link 的参数后，如果有未配置于 NWCONFIG 网络中的节点时，该节点会以红色框线显示「异步组件」，而当 Ether Link 的设置存在此种异步组件时，就不能进行监控或下载的动作。




20.4.8 删除异步装置

当存在异步组件时，将不能进行监控或下载的动作，此时先确认产生异步组件的原因，若是因为 NWCONFIG 中的网络配置有误，请先修正网络配置后再重新上传 Ether Link 的参数；但若是上传的参数已不适用目前的网络配置时，请将异步组件删除后再进行后续的操作。

于图示区中点选欲删除的异步组件后，于功能工具栏点选 **编辑 (E) > 删除所选异步组件**，也可以按图示工具栏中的  图示删除所选的异步组件。



要同时删除所有的异步装置时，请直接于功能工具栏中点选 **编辑 (E) > 同步装置**，或按下图标工具栏中的  图标即可。



20.4.9 启动/关闭在线监控模式

在 Ether Link 的设定窗口中，用户也可通过在线监控来操作或测试 Ether Link 的功能，而在线监控的启动/关闭方式又可分为单点与多点。

启动/关闭方式	说明	
单点	功能	每次仅启动或关闭所选节点的在线监控功能。
	条件	ISPSOft 可与所选的主机正常联机，且已完成 NWCONFIG 的通讯设定。
多点	功能	同时启动或关闭所有节点的在线监控功能。
	条件	所有节点都已接上网络并且 ISPSOft 通过 Ethernet 进行联机，而在 NWCONFIG 的通讯设定中，也须选择 Ethernet 型式的 Driver。



在进行监控前请再次确认所有的节点都已依照 NWCONFIG 中的规划完成架设并已正常运作。

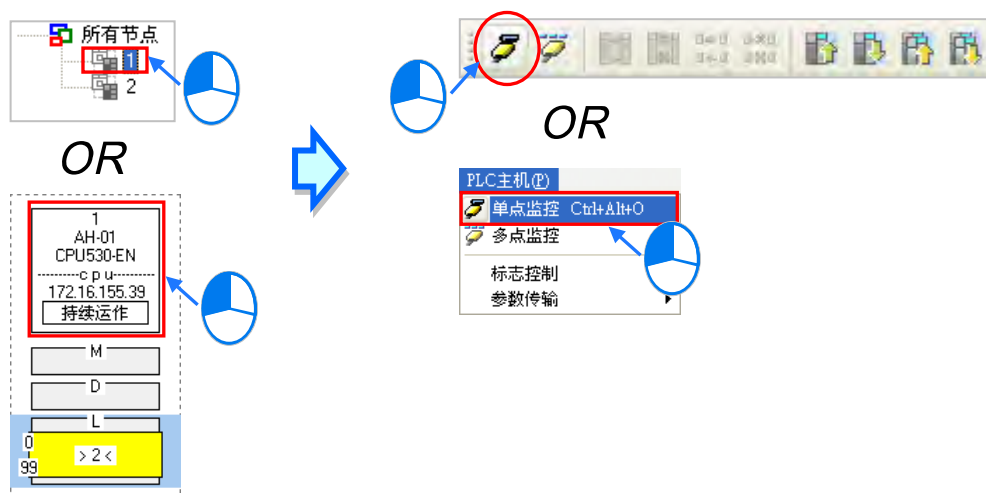
- (a) 每个节点都已依照 NWCONFIG 中的规划接上网络。
- (b) 每个节点都已通过 HWCONFIG 设定各个通讯端口的以太网络参数，并下载至各个主机与模块当中，而设定的参数则必须与 NWCONFIG 中的规划一致。
- (c) 已正确的将 Ether Link 的设定参数下载至主机当中。
- (d) 各个节点都已上电并处于正常运作的状态。

20.4.9.1 启动监控


● 启动单一节点的监控

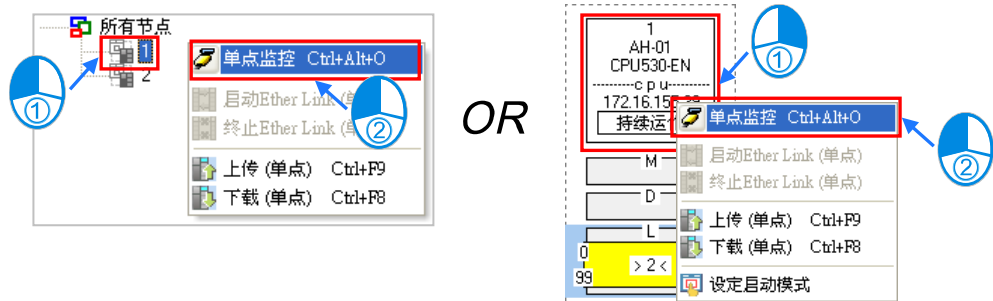
➢ 方法一

选取要进行监控的数据需求节点，接着于图标工具栏中将  图标按下，或是于功能工具栏中依序点选 **PLC 主机 (P)** > **单点监控**。而进入单点监控后， 图标也会呈现按下的状态。




方法二

于**装置清单**或图标区中，点选要进行监控的**数据需求节点**后按下鼠标右键，接着于快捷菜单中点选「**单点监控**」。而进入单点监控后， 图标也会呈现按下的状态。



启动所有节点的监控

方法一

于功能工具栏点选 **PLC 主机 (P)** > **多点监控**，或直接将图标工具栏中的  图标按下。



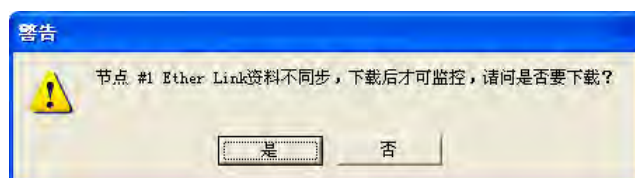
方法二

于**装置清单**中点选「**所有节点**」后按下鼠标右键，接着于快捷菜单中点选「**多点监控**」。



补充说明

进入监控画面前，系统会先比对窗口中的 Ether Link 设定与主机当中的设定是否相同，若不同，系统便会要求重新下载设定参数。

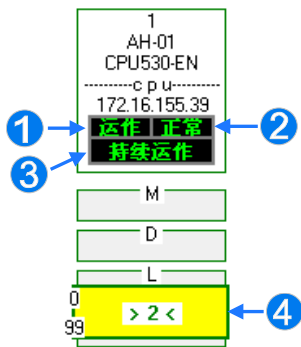


20.4.9.2 监控状态显示

当进入监控模式之后，监控中的节点在**装置清单**中的图标便会呈现彩色。



而此时在节点的图标中也会以不同的文字或图标来显示目前 Ether Link 的运作状态。




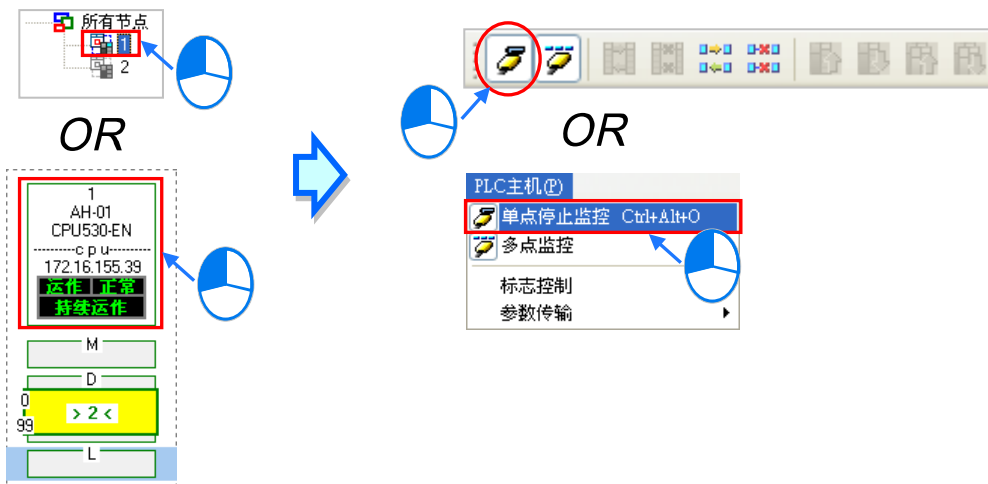
状态	说明
❶ 运作/停止	运作 : Ether Link 功能启动
	停止 : Ether Link 功能关闭
❷ 运作状态	正常 : 运作正常
	错误 : 运作异常
❸ 启动模式	保持停止 : 启动模式为保持停止
	持续运作 : 启动模式为持续运作
	标志模式 : 启动模式为标志模式
❹ 数据区块	> 2 < : 数据交换中
	◀ 2 ▶ : 数据交换动作停止

20.4.9.3 关闭监控

● 关闭单一节点的监控

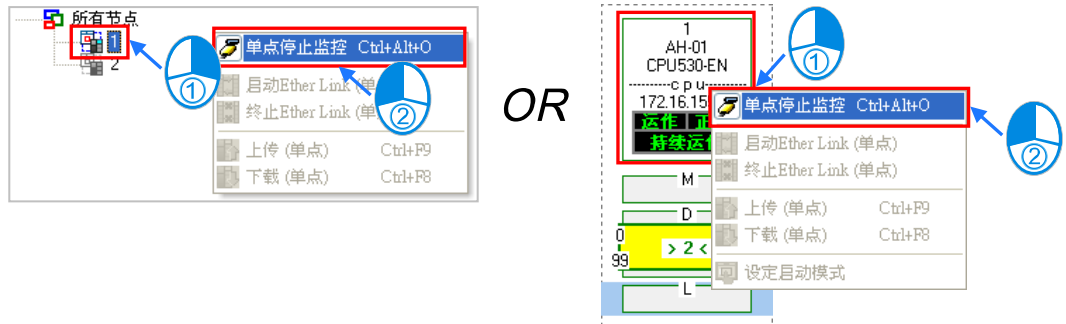
➢ 方法一



先选取要关闭监控的**数据需求节点**，接着于图标工具栏中将  图标重新按起，或是于功能工具栏中依序点选 **PLC 主机 (P)** > **单点停止监控**。

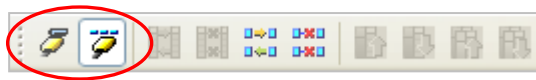


方法二

于装置清单或图标区中，点选要关闭监控的数据需求节点后按下鼠标右键，接着于快捷菜单中点选「单点停止监控」。




当所选节点的监控功能被关闭接着，点选该节点时，图标工具栏的  图标会呈现未按下的状态，但若还有其它的节点处于监控状态时，图标工具栏的  图标则仍会呈现按下的状态。



● 关闭全部节点的监控

方法一

于功能工具栏中点起 **PLC 主机 (P)** > **多点监控**，或重新将图标工具栏中的  图标按起。



方法二

于装置清单中点选「所有节点」后按下鼠标右键，接着于快捷菜单中将「多点监控」点起。




20.4.10 在线启动/停止 Ether Link (标志模式)

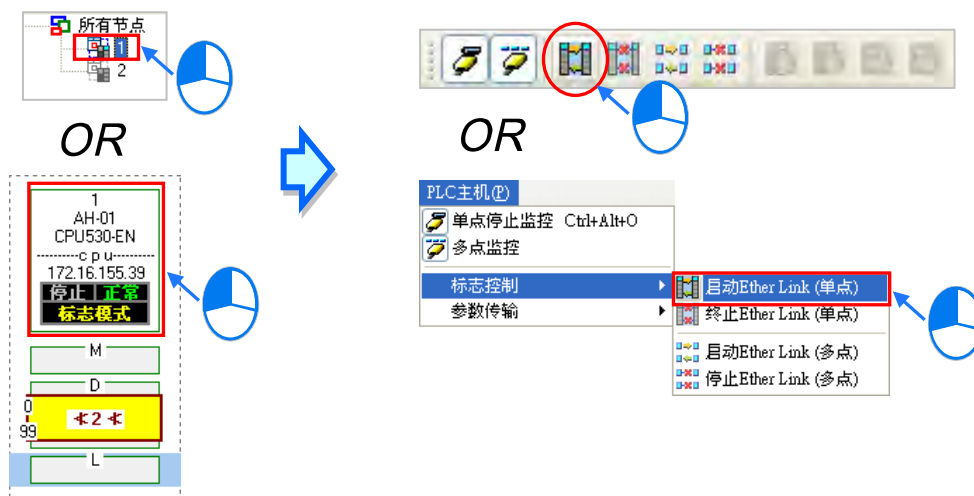
当节点的启动模式为**标志模式**时，于在线监控状态下便可直接操作该标志，并藉此启动或停止某个监控节点的 Ether Link 运作；不过此操作仅限监控中的节点。

20.4.10.1 启动 Ether Link

- 启动单一节点的 Ether Link 功能

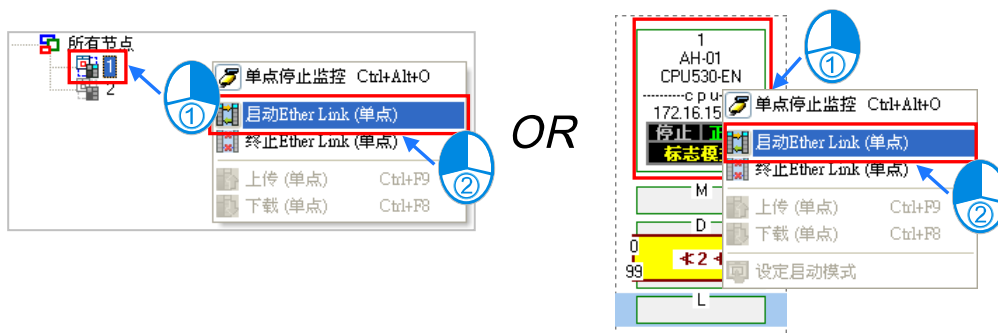
- 方法一

选取要启动 Ether Link 的节点，接着于图标工具栏中点击  图标，或是于功能工具栏中依序点选 **PLC 主机 (P)** > **标志控制** > **启动 Ether Link (单点)**。




- 方法二

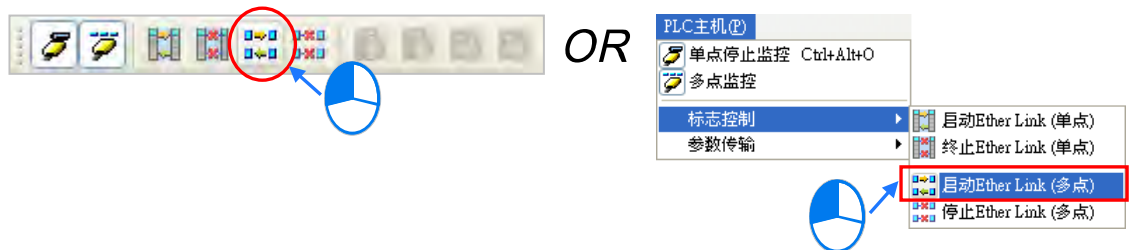
于**装置清单**或**图标区**中，点选取要启动 Ether Link 的节点后按下鼠标右键，接着于快捷菜单中点选「**启动 Ether Link (单点)**」。



● 启动所有节点的 Ether Link 功能

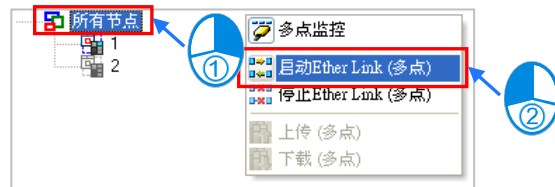
➤ 方法一

于功能工具栏中依序点选 **PLC 主机 (P)** > **标志控制** > **启动 Ether Link (多点)**，或直接在图标工具栏中点击  图标。



➤ 方法二

于装置清单中点选「所有节点」后按下右键，接着于快捷菜单中点选「启动 Ether Link (多点)」。

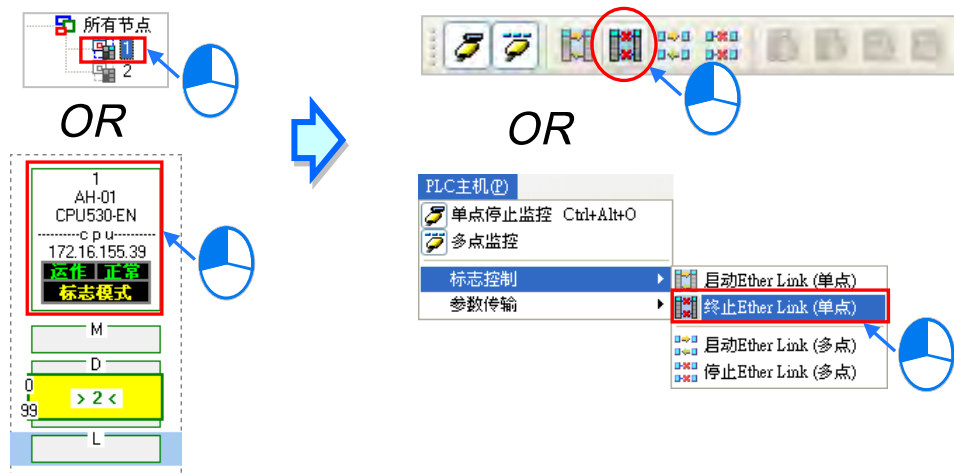


20.4.10.2 停止 Ether Link

● 停止单一节点的 Ether Link 功能

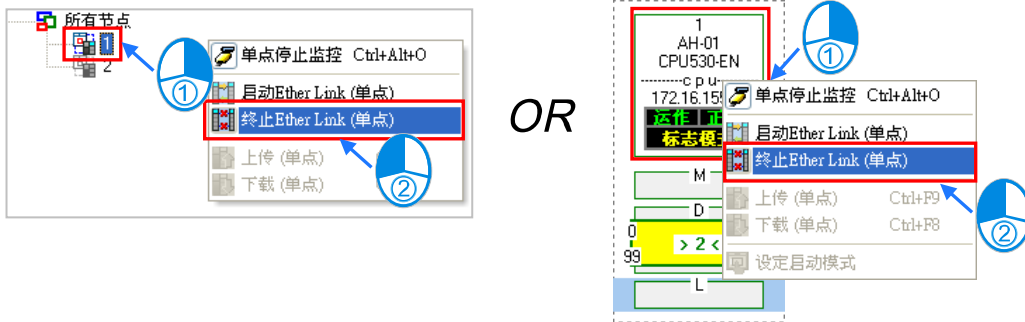
➤ 方法一

选取要停止 Ether Link 的节点，接着于图标工具栏中点击  图标，或是于功能工具栏中依序点选 **PLC 主机 (P)** > **标志控制** > **终止 Ether Link (单点)**。



➤ 方法二

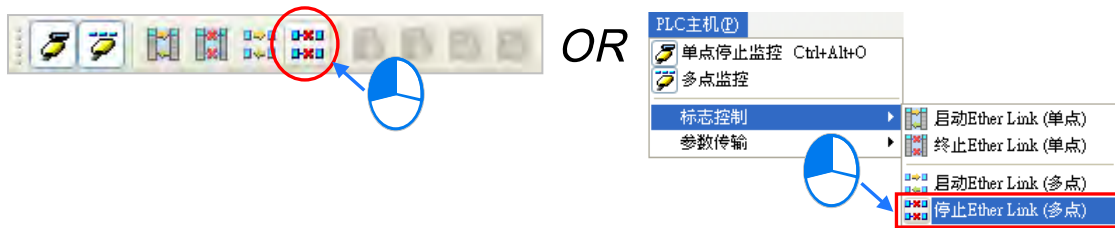
于**装置清单**或图标区中，点选取要停止 Ether Link 的节点后按下鼠标右键，接着于快捷菜单中点选「**终止 Ether Link (单点)**」。



● 停止所有节点的的 Ether Link 功能

➤ 方法一

于功能工具栏中依序点选 **PLC 主机 (P)** > **标志控制** > **停止 Ether Link (多点)**，或直接在图标工具栏中点击 图标。



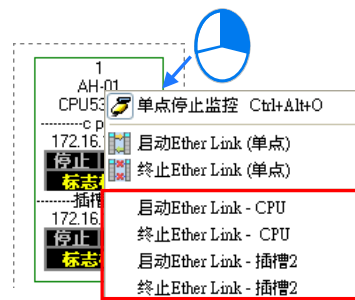
➤ 方法二

于**装置清单**中点选「**所有节点**」后按下右键，接着于快捷菜单中点选「**停止 Ether Link (多点)**」。



补充说明

当操作的节点本身包含多个通讯端口时，于该节点图标上按下右键后，接着便可通过快捷菜单来单独启动或停止某个端口的 Ether Link 功能。不过因 Ether Link 的数据交换是以节点为单位，因此即使某个端口的 Ether Link 功能被关闭了，数据交换的动作仍可能经由该节点的其它端口持续进行。



20.4.11 监控表与错误记录

于监控节点的数据区块上双击鼠标左键，接着便可开启该区块的数值监控窗口，不过此监控窗口的功能仅供数值查看，用户无法于此对表格中的数值进行变更。



- ① 于此字段可下拉选择数值的显示格式。
- ② 此处显示的是监控内容所对应的的区块信息。
- ③ 于此字段输入装置地址后按下键盘的【Enter】键，接着便可将该地址移至可视范围。


当选取某个节点时，下方讯息窗口中的「错误记录」页面便会显示该节点主机内部的错误记录；而当勾选「Ether Link 错误过滤」时，表格中便只会列出与 Ether Link 相关的错误记录；按下「更新」便可重新撷取主机内部的错误记录并将其显示于表格中，而按下「清除」键则可将此页面与主机内部的错误状态一并清除。

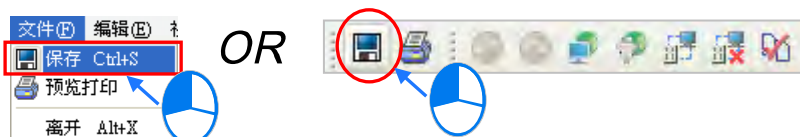



错误记录的内容会包含主机与模块所产生的错误记录，而当某笔错误记录是产生自模块时，除了从「装置型态」的字段得知该主机的型号之外，也可由于「背板编号」与「插槽编号」两个字段得知该模块的所在位置；但若是 CPU 所产生的错误，「背板编号」与「插槽编号」字段则不会有数字显示。

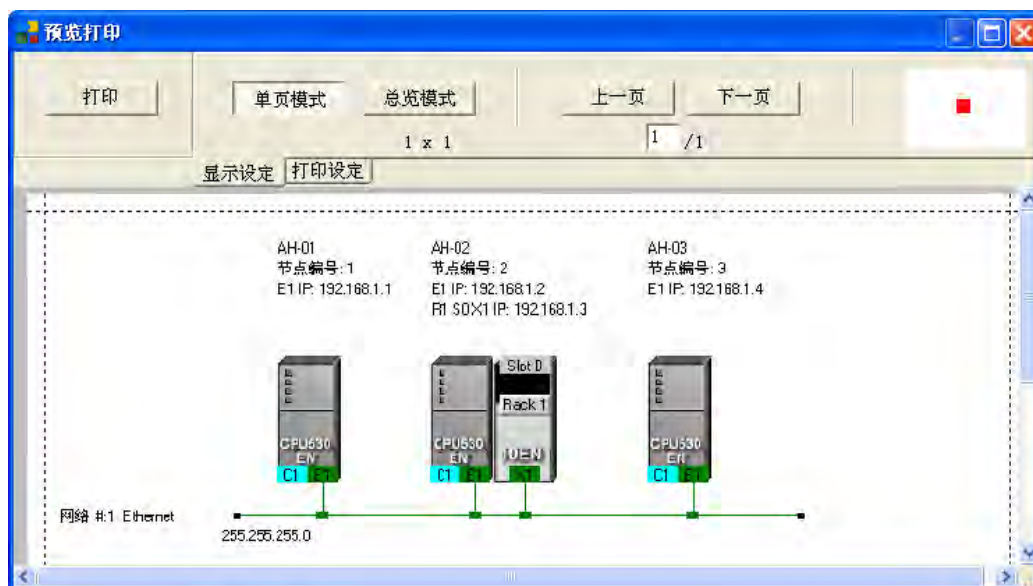
20.5 NWCONFIG 的管理与应用

20.5.1 保存与打印

要进行保存时，请于功能工具栏中点选 **文件 (E) > 保存**，或直接于图标工具栏中点选  图标，便可将 NWCONFIG 中的参数进行保存，且保存的内容将包括网络配置、PLC Link 参数与 Ether Link 参数；而保存后，于项目的路径下将产生与项目同名，但扩展名为 *.nw 与 *.nwsd 的两个文件。



当要进行打印时，请于功能工具栏中点选 **文件 (E) > 预览打印**，或直接于图标工具栏中点选  图标，接着系统便会自动开启**打印管理工具**，但打印的内容只会包含网络部署的图面。关于**打印管理工具**的说明请参考 **附录 C**。



若在应用上须要打印 **PLC Link** 或 **Ether Link** 的数据时，请于各自的设定窗口中利用「导出」的功能将**数据交换表**导出为 CSV 文件，接着再通过 EXCEL 将其打印出来即可。

20.5.2 整体下载

NWCONFIG 的图标工作区中，对网络的绕送路径表 (Routing Table)、PLC Link 的参数与 Ether Link 的参数做整体的同步下载，而下载时也必须逐一完成每个节点数据的下载。

20.5.2.1 下载参数说明

● 绕送路径表 (Routing Table)

每个主机内部只会保存与其本身相关的路径数据，因此不同节点会被下载的绕送路径表 (Routing Table) 数据也会不同。

● PLC Link 参数

仅有指定为主站的主机须要下载 PLC Link 参数，但对于其它的从站而言，下载 PLC Link 便会将其内部对应 PLC Link 的特殊继电器与特殊寄存器恢复为默认值；因此若不确定该装置是否保有先前被设为主站的 PLC Link 参数时，请一并对其下载 PLC Link 参数，以免发生同一网络有多个主站的情形。

● Ether Link 参数

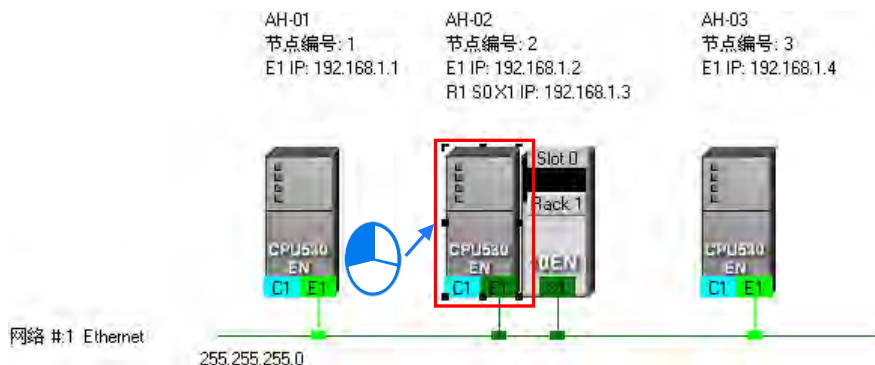
在数据交换表中有提出数据需求的节点都必须进行下载，不过在各主机的内部则仅会保存与本身相关的数据需求表；若某装置在 Ether Link 的规划中没有设定任何数据需求时，对其进行下载之后便会清除该装置内部的数据，而同时也将依据 Ether Link 中的规划设定该装置的启动模式。


20.5.2.2 下载操作说明

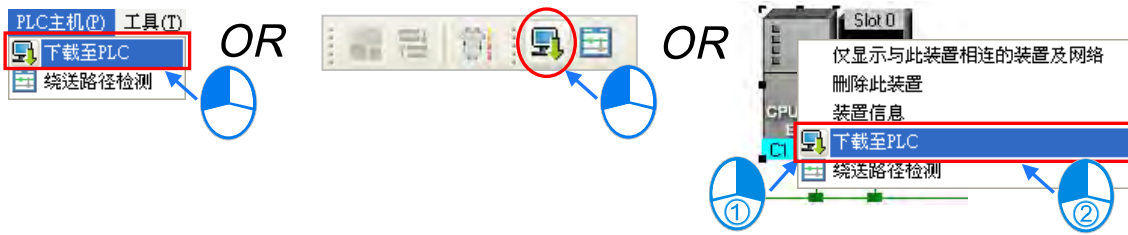
整体下载的方式可分为单点与多点两种模式，且下载时的通讯设定不可使用报文绕送模式。

● 单点模式


首先请选择要下载的节点，并于图标工作区中点选该节点的主机图标。

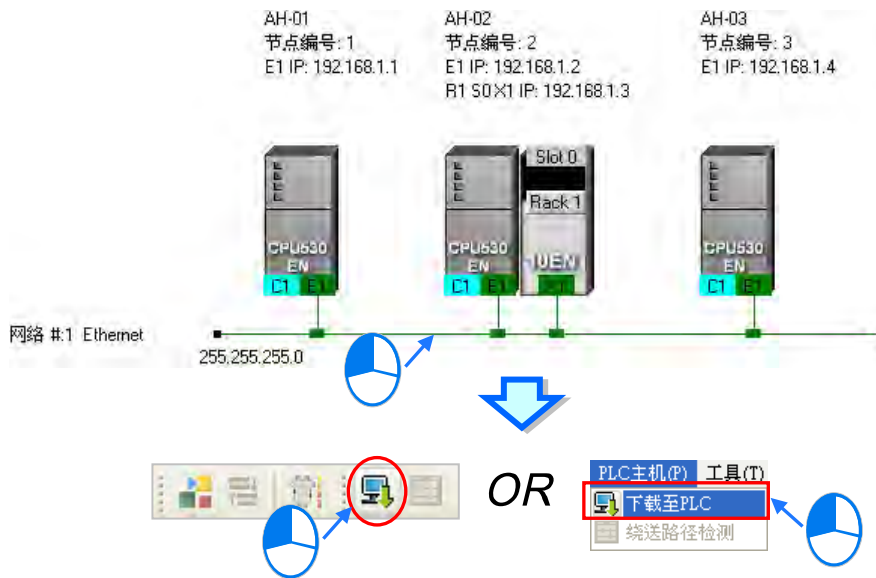


接着于功能工具栏中点选 **PLC 主机 (P)** > **下载至 PLC**，或于图标工具栏中点选  图标，同时也可于主机图标上点击鼠标右键，接着再于右键快捷菜单中点选「**下载至 PLC**」即可。

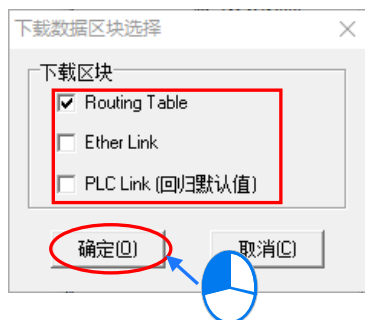


● **多点模式**

图标区中点选其中的 **Ethernet** 网络后，于功能工具栏中点选 **PLC 主机 (P)** > **下载至 PLC**，或 directly 于图标工具栏中点选  图标即可。而若实际的接线与 NWCONFIG 中的配置一致时，接着便可将参数下载至与该网络连接的所有节点。



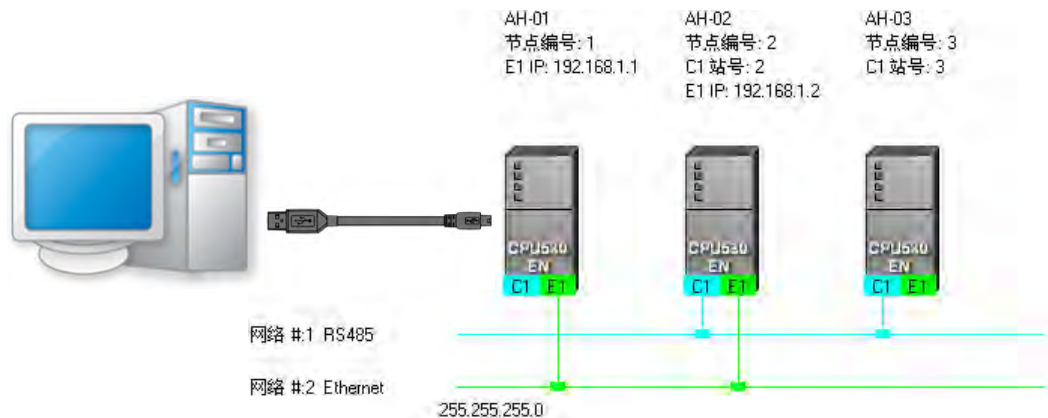
依据上述的步骤操作之后，画面会出现下载选项的设定窗口，而其中若有呈现灰阶的选项时，表示不支持此项功能；勾选完要下载的项目后按下「**确定**」键即可。



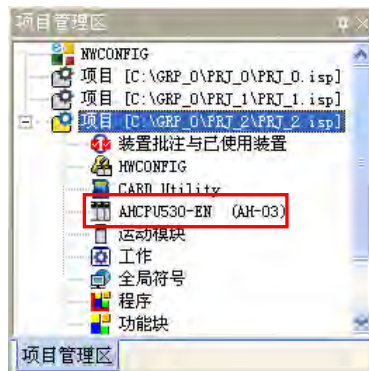
20.5.3 ISPSOft 中的绕送应用

当在 NWCONFIG 中完成网络架构的规划，且已成功下载绕送路径表 (Routing Table) 至各个节点后，于 ISPSOft 中便可通过绕送功能来对非直接连接的主机进行上下载或监控；不过当要在 ISPSOft 中使用绕送功能时，其操作的对象则必须都是 AH5x0 系列的机种。

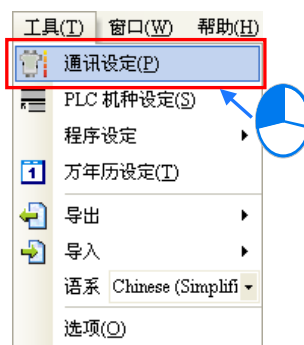
如下图，当实际连接的主机是 AH-01，但要对 AH-03 进行监控时，通过绕送功能便可达成。



(1) 因要进行监控的是 AH-03，因此请先于群组项目中启动 AH-03 所属的项目。



(2) 接着在功能工具栏中依序点选 工具 (I) > 通讯设定 (P)。



- (3) 设定窗口中勾选「**报文绕送模式**」，接着于「**初始站台**」字段下拉选择报文绕送的起点站。
- 一般而言，「**初始站台**」会是与计算机实际联机的主机，此处的范例为 AH-01，但若计算机同时连接其它主机，或是通过 Ethernet 与多个主机进行联机时，请依据 NWCONFIG 的网络规划来选择第一个接收报文的装置，也即计算机直接与其通讯的对象。此外，当选择使用**报文绕送模式**时，在「**通讯通道名称**」所选择的对象便必须是可与「**初始站台**」联机的 COMMGR Driver。
- 完成后按下「**确定**」键即可结束设定，而接下来便可直接通过 AH-01 来对 AH-03 进行监控。

通讯设置

通讯通道名称 Dvr_USB

站号 0

报文绕送模式

初始站台 AH-01

确定 取消

第21章 数据的备份与还原

目录

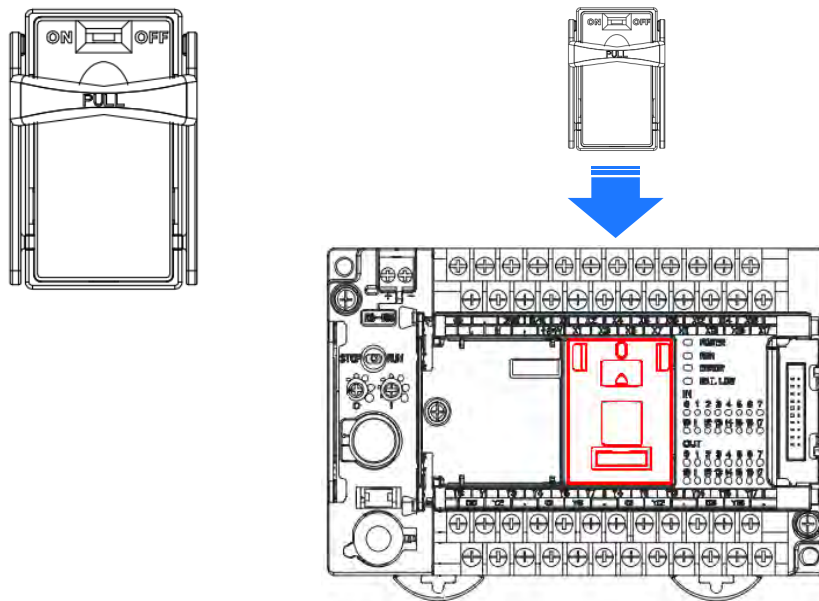
21.1 数据备份存储卡辅助工具	21-2
21.1.1 数据备份存储卡简介	21-2
21.1.2 数据备份存储卡的操作	21-3
21.2 PLC 永久备份设定.....	21-3
21.3 存储卡工具 - CARD Utility (AH/AS 机种限定)	21-4
21.3.1 软件介绍.....	21-6
21.3.2 备份操作.....	21-8
21.3.3 还原操作.....	21-15
21.3.4 命令列模式指令操作说明	21-19

21.1 数据备份存储卡辅助工具

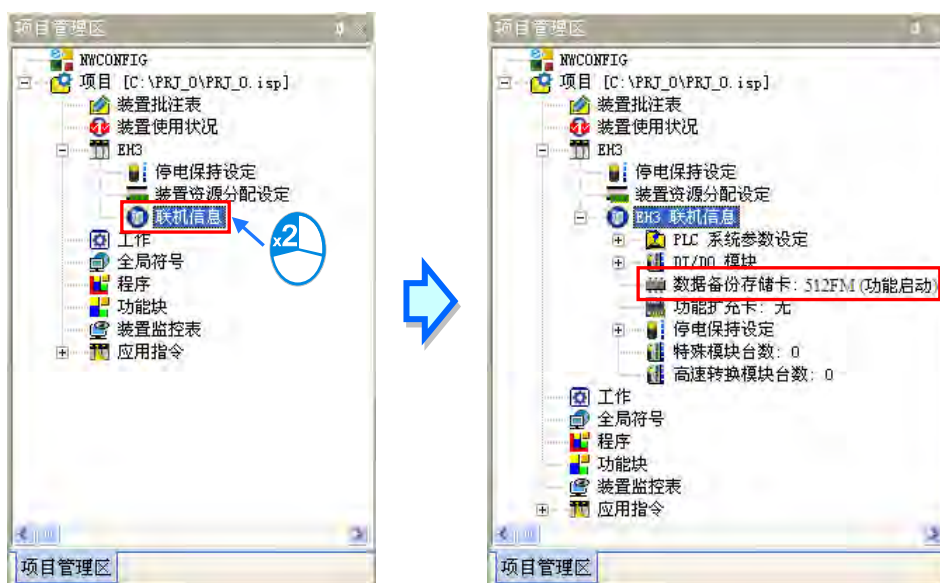
21.1.1 数据备份存储卡简介

21

DVP 系列的 EH2 与 EH3 机种可搭配其专属的数据备份存储卡，其产品型号分别是 **DVP-256FM** (限 EH2 使用，乳白色外壳)与 **DVP-512FM** (限 EH3 使用，黑色外壳)，产品外观如下图。

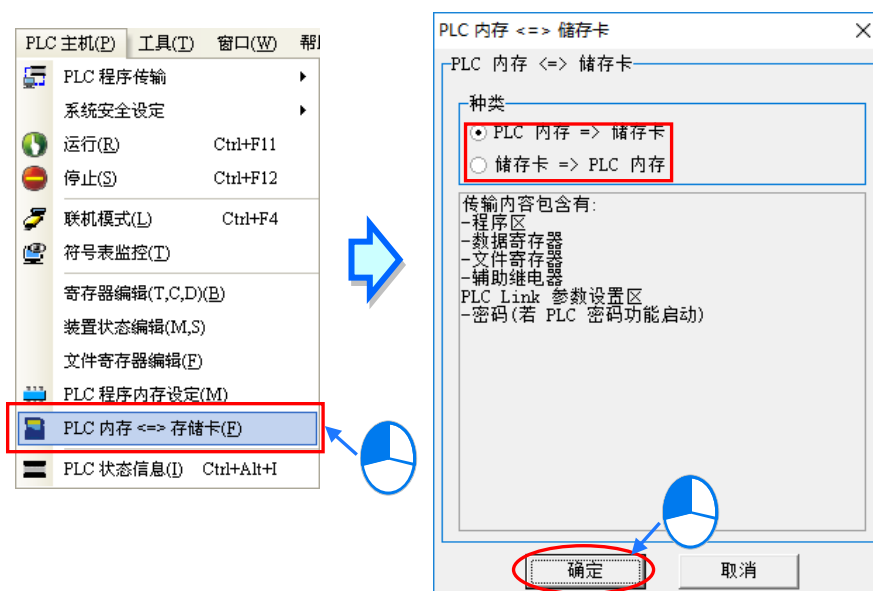


当存储卡安装完毕并将 PLC 上电后，ISPSOft 的项目管理区中展开机种目录，接着于「联机信息」上双击鼠标左键，此时若 ISPSOft 可与主机正常联机，则主机的系统信息便会被显示在「联机信息」的下层项目中，而通过其中的「数据备份存储卡」项目便可知目前存储卡的状态。



21.1.2 数据备份存储卡的操作

要进行备份存储卡的操作之前，请先确认 ISPSOft 与主机正常联机，之后于功能工具栏中点击 **PLC 主机 (P)** > **PLC 内存<=>存储卡 (E)**，接着选择数据传输的种类后按下「**确定**」键即可。



*.关于数据备份存储卡的详细说明与支持的机种、韧体版本等信息请参考相关的使用手册或技术文件。

21.2 PLC 永久备份设定

EH2 / EH2-L / EH3 / EH3-L / SV / SV2 几个机种，其机器本体即有内建 Flash 的内存空间用来备份主机的数据。相关功能及使用限制请参考各机种的手册或说明文件。

要通过 ISPSOft 进行数据的备份或还原之前，请先确认 ISPSOft 与主机正常联机，接着于功能工具栏中依序点击 **工具 (I)** > **PLC 永久备份设定**，待出现操作窗口后，请先选择数据的传输选项，接着勾选要传输的内容项目，之后再选择是否启动警告信息（此功能是设定当数据因电池电压不足遗失，PLC 自动重新还原数据时，PLC 是否要弹出警告信息），完成后按下「**确定**」键即可。



21.3 存储卡工具 - CARD Utility (AH/AS 机种限定)

开始之前请先参考 CARD Utility 备份还原支持列表、确认你的机种是否支持你要备份的项目。

*AS532EST、AS564EST、DVP50MC-04S、DVP50MC-06S 机种目前尚不支持

	AS DVP-ES3	AH5x0	AH5x1RS	AH5x1EN AH560	AHxx EMC	AS5xx DVPxxMC	DVP
源代码 (注1)	V	V	V	V	V	V	
执行码	V	V	V	V	V	V	V
项目密码 程序标识符	V	V	V	V	V	V	V
HWCONFIG 参数设定	V	V	V	V	V	V	
主机与模块 配置表、各 模块参数设 定	V	V	V	V	V	只支持 AS5xx	
装置内容 (注2)	M · D · C · HC · FR	X · Y · M · SM · SR · D · B · W · L · S · TB · T · CB · C · HCB · HC · E	X · Y · M · SM · SR · D · W · L · S · TB · T · CB · C · HCB · HC · E	X · Y · M · SM · SR · D · W · L · S · TB · T · CB · C · HCB · HC · E	X · Y · M · SM · SR · D · W · L · S · TB · T · CB · C · HCB · HC · E	(注3)	D, C, HC, T, FR
通讯数据 交换表	COM1 COM2 Ethernet FEN02 FunctionCard1 FunctionCard2 EIP I/O connection table CAN DS301 (注4)	RS485 Ethernet	COM1 COM2	COM1 Ethernet RTU IO table EIP I/O connection table	COM2 Ethernet EtherCAT	CANopen	
定位规划表	V						

	AS DVP-ES3	AH5x0	AH5x1RS	AH5x1EN AH560	AHxx EMC	AS5xx DVPxxMC	DVP
初始值表	V		V	V	V		
ECAM	V				V	V	
Gcode						V	
轴参数					V	V	
只读区域 设定及 备份文件 注释							V
C 语言 执行码	V						
数据记录器 取样设定	V						
EIP TAG	V						
OPC-UA TAG	V						
HW TAG	V						
IO Link 模块	V						
网页设定	V (注 4)						

注 1：将 ISPSOft 的项目数据导出为备份文件（DUP）时，可选择是否备份源代码。从主机备份至计算机端或内存卡时，若主机内原本就没有下载源代码则产生的备份文件也不包含源代码。不包含源代码的备份文件无法被还原成 ISPSOft 项目。

注 2：装置备份还原机制

注 3：DVPxxMC、AS5xx 机种可使用 DMD 文件备份还原停电保持变量当前值。

注 4：CAN DS301 以及网页设定只支持 CPU 与 SD 卡间的备份还原。

	备份			还原		
	ISP 项目 至计算机	主机 至计算机	主机 至内存卡	计算机 至 ISP 项目	计算机 至主机	内存卡 至主机
AS DVP-ES3	有装置文件 (dvl · dvb · wft) 存在时 会备份装置	可选择要 不要备份 装置	可选择要 不要备份 装置	备份文件中有 装置数据时会 还原装置文件 (dvl · dvb · wft)	备份文件中有 装置数据时就 会还原	备份文件中 有装置数据 时就会还原
AH5x0	不备份装置			不还原装置		

	备份			还原		
	ISP 项目 至计算机	主机 至计算机	主机 至内存卡	计算机 至 ISP 项目	计算机 至主机	内存卡 至主机
AH5x1RS						
AH5x1EN AH560						
AHxxEMC						
AS5xx DVPxxMC	不备份装置	不支持	不支持	不支持	不还原装置	不支持
DVP	可选择要不要备份装置文件(dvl, wft)的内容	可选择要不要备份装置	不支持	不支持	备份文件中有装置数据时就会还原	不支持

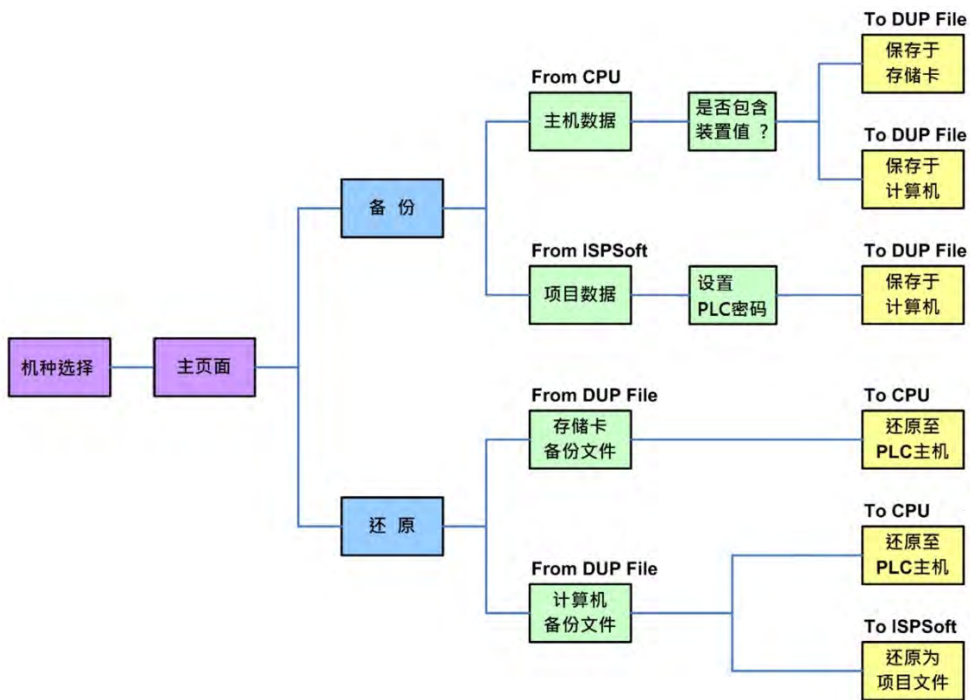
21.3.1 软件介绍

AH/AS 系列机种本身即内建 SD 卡的插槽，且通过存储卡的操作便可对主机进行数据的备份与还原；而在 ISPSOft 中，针对 AH/AS 系列机种也提供便于操作的存储卡工具 - CARD Utility，其主要的功能是协助用户在存储卡、主机及 ISPSOft 项目中的数据进行备份与还原，而其备份的数据则包含程序代码、参数设定、硬件与网络规划以及主机内部的装置内容值，而 HWCONFIG 中所储存的智能型模块设定相关数据，除了 ECAT Builder 支持备份与还原外，其他软件工具尚不支持备份与还原。关于各系列机种的 SD 卡规格与使用限制请参考各机种的操作手册或相关技术文件。

AH5x0 系列机种关于网络规划的部份，因每个主机的内部只会储存与自身相关的网络规划数据，因此备份时也只会备份与所选 PLC 或 ISPSOft 项目主机相关的部份，其内容则包括 Routing Table 与 Ether Link 的数据；而同样的，在进行还原时，即使选择将备份的数据还原回 ISPSOft 的项目文件，而还原之后，该项目文件中的 NWCONFIG 数据仍将无法重建。关于网络规划参数的细节请参考第 20 章当中的相关说明。

下列为 CARD Utility 所支持的功能简述与流程示意图。

- 将主机内部存放的数据导出为备份文件 (*.dup)，并可选择将其存放在 PLC 端的存储卡，或是计算机端的自定义路径。而在操作时，也可选择是否要将主机内部装置的目前数值一并进行备份。
- 将 ISPSOft 的项目资料导出为备份文件 (*.dup)，但只能存放于计算机端的自定义路径，AS 系列机种可以将寄存器编辑 (.dvl)、装置状态编辑 (*.dvh)、文件寄存器编辑 (*.wft) 所储存的文件视为装置值进行备份。
- 将存储卡 (PLC 端) 中的备份文件重新载回 CPU 主机当中。
- 将储存在计算机端的备份文件 (*.dup) 重新载回主机当中，或是选择重新还原为 ISPSOft 的项目文件；而 AH 系列机种当选择还原为 ISPSOft 项目时，系统将自动略过装置内容值以及网络规划的部份。AS 系列机种将计算机端的备份文件还原为 ISPSOft 项目时，寄存器编辑 (.dvl)、装置状态编辑 (*.dvh)、文件寄存器编辑 (*.wft) 的内容也会被还原。



要启动软件时，请在项目管理区的「CARD Utility」项目上双击鼠标左键，系统便会自动开启操作精灵页面，先选择要进行操作的系列机种，按下「下一步」按钮进入主页面。



21.3.2 备份操作


当要进行备份的来源或目的位置有任何一个为 CPU 主机或存储卡时，请在执行前确认 ISPSOft 与主机正常联机。

21

- (1) 主页面选择「备份 (至 DUP 文件)」选项，之后按「下一步」进行后续的操作。



- (2) 选择备份的来源项目，确定后按「下一步」以进行后续的操作。

当选择「ISP 项目」时，请接着点击下方字段的  按钮以选择要备份的 ISP 项目文件 (*.isp)，但若所选文件中的程序尚未完成编译，则于执行时系统将会显示提示信息；此时请重新编译并储存，而完成后便可再次对该文件进行备份的操作。



(3) 当备份的来源为「CPU」时，便会提示备份是否包含主机装置的内容值。



若使用 DVP 机种时且备份的来源为「ISP 项目」时，跳出备份内容设定，用户可以自行设定要备份的项目。



(4) 选择备份文件 (*.dup) 的存放位置，若选择的备份来源为 ISPSOft 项目时，此处则仅可选择存放于计算机端。

a. 选择存放于 PLC 端的存储卡时，而 AH 系列机种备份文件的文件名与路径会固定如下。

存储卡根目录\PLC CARD\AH500\SysDup\AUTOEXEC.dup。

AS 系列机种备份文件的文件名与路径则会固定如下。

➤ **AS300 系列机种**


存储卡根目录\SDCard\PLC CARD\AS300\SysDup\AUTOEXEC.dup。

➤ **AS200 系列机种**

存储卡根目录\SDCard\PLC CARD\AS200\SysDup\AUTOEXEC.dup。

➤ **DVP-ES3 机种**

存储卡根目录\SDCard\PLC CARD\ES3\SysDup\AUTOEXEC.dup。

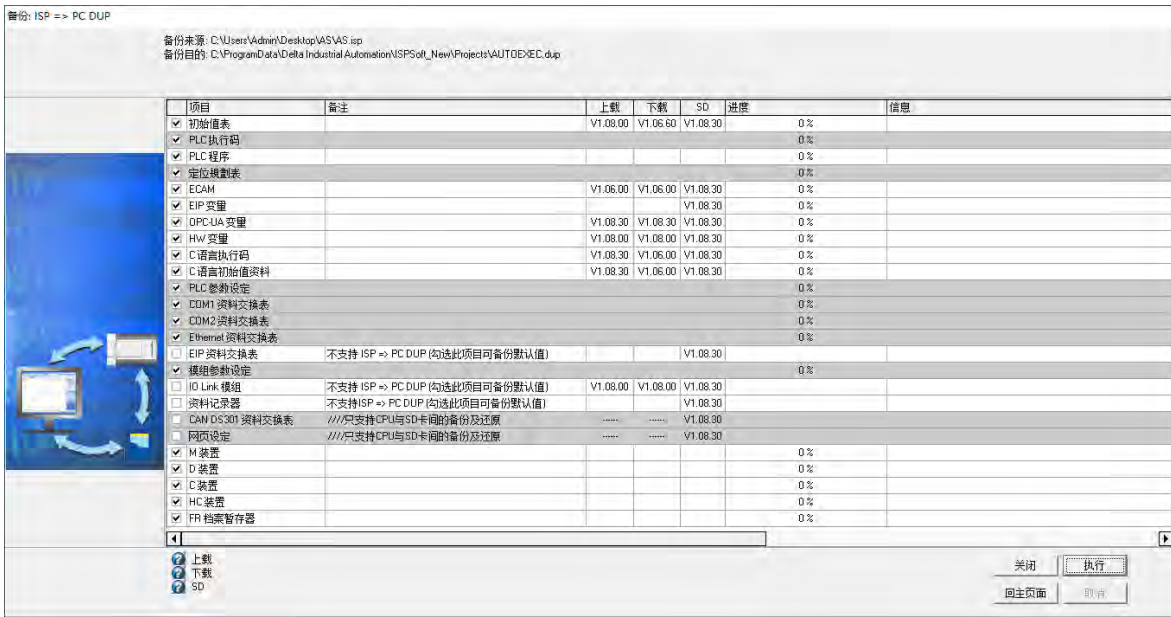
b. 选择存放于计算机端时，请点击下方字段右侧的  按钮以配置文件名与路径。



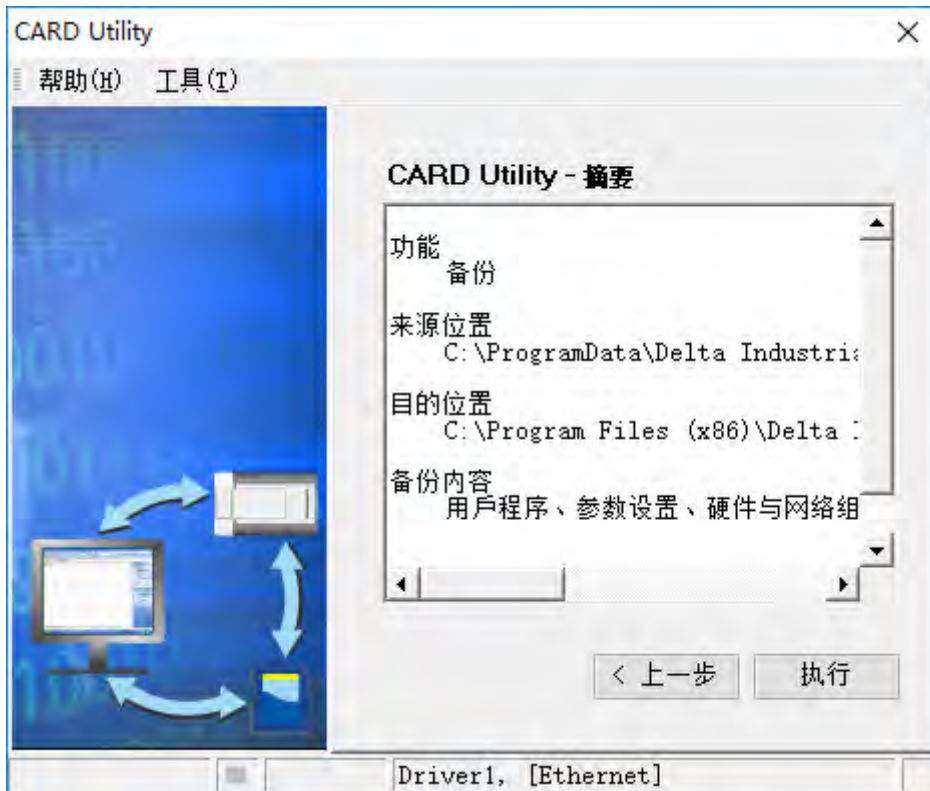
- (5) 若选择的备份来源为 ISPSOft 项目时，可另行设定 PLC 密码，选择 **PLC 密码与项目密码相同** 或是 **设定新的 PLC 密码**，若选择设定新的 PLC 密码，将会出现密码设定窗口供用户设定密码与限制次数。



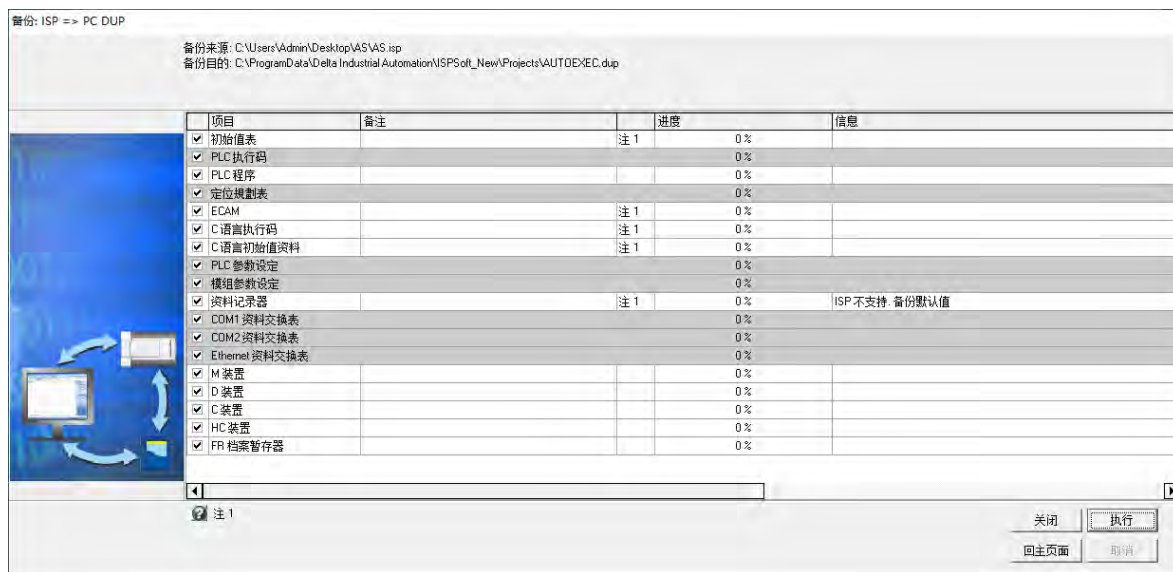
21



(6) 确认摘要的内容与要进行的操作一致后，按下「执行」即可开始进行备份的动作。



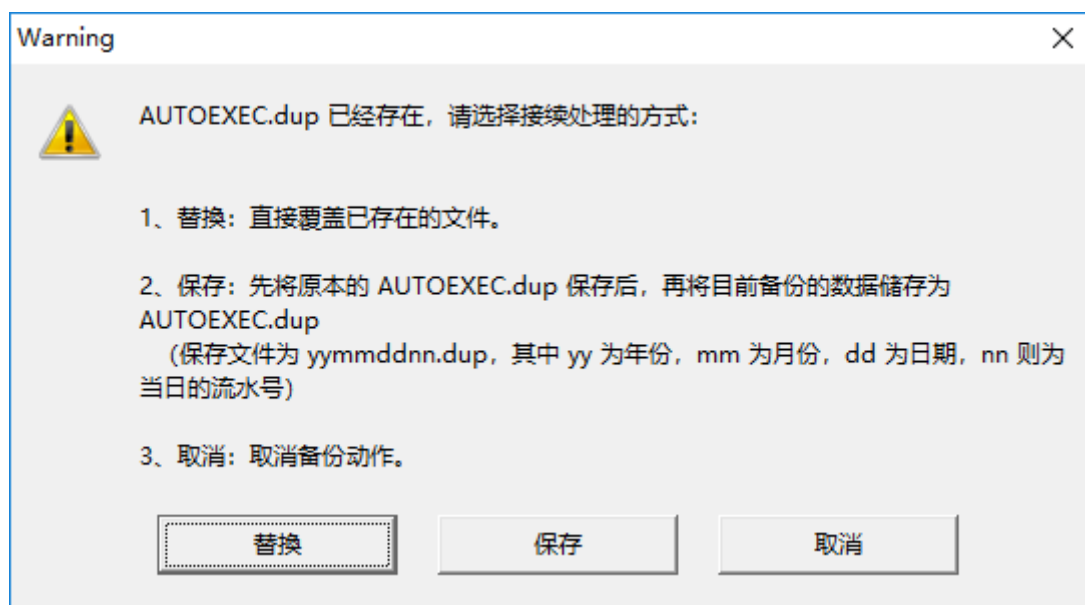
若使用 AS 机种则会弹出如下图的窗口，灰底的选项是固定不能取消，其余白底的选项用户可以依照需求来设定，设定完成后按下**执行**按键，即可以开始备份。



21

将主机的数据备份至存储卡的过程中，用户可按下「取消」键来中断软件端的动作，但主机端的系统仍可能会继续处于备份的状态，此时则可将主机断电以强制中断，但所产生的备份文件便可能不完整，因此事后请务必将存储卡中的备份文件删除。

当选择将数据备份至存储卡时，因其文件名与路径皆是固定的，因此若存储卡中已存在旧有的备份文件时，系统将会出现询问窗口，此时请依据提示的信息以选择最适合的做法。



当备份的来源有开启密码保护的功能时，其处理的机制如下。

操作项目	说明
CPU → 存储卡	CPU 中的「PLC 标识符」与「PLC 密码」会一并包含于备份文

操作项目	说明
CPU → 计算机端	系统会先要求输入主机的「 PLC 标识符 」与「 PLC 密码 」，并在确认无误后才会开始进行备份；而完成备份之后，「 PLC 标识符 」与「 PLC 密码 」的数据将会一并包含于备份文件中。
ISPSOft 项目 → 计算机端	项目中的「 程序标识符 」、「 项目密码 」与备份时输入的「 PLC 密码 」会一并包含于备份文件。

(7) 完成操作后，接着便可选择回到主页面继续操作，或是直接关闭。



21.3.3 还原操作

当要进行还原的来源或目的位置有任何一个为 CPU 主机或存储卡时，请在执行前确认 ISPSOft 与主机正常联机。

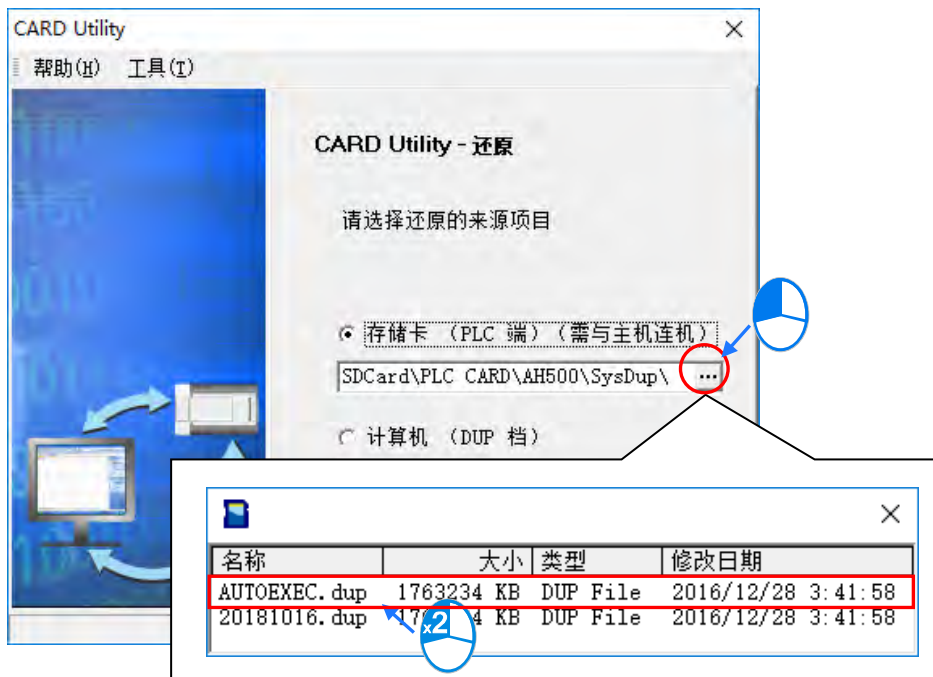
(1) 主页面选择「还原 (从 DUP 文件)」选项，之后按「下一步」以进行后续的操作。



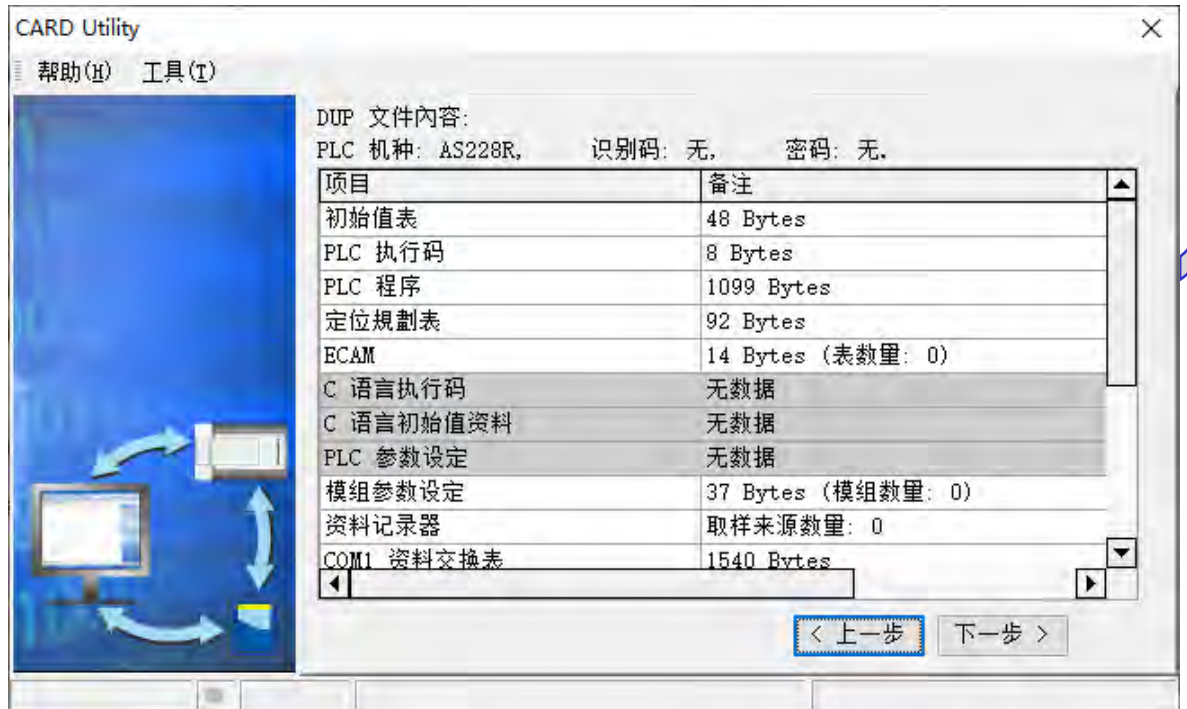
(2) 单击路径字段右侧的  按钮选择要还原的备份文件。



若指定的备份文件是存放于 PLC 端的存储卡，则选择文件时，系统会自动联机至 PLC 并将存储卡指定路径中的所有备份文件显示于窗口中，接着请直接于要还原的备份文件上双击鼠标左键即可。



若使用 AS 机种则会弹出如下图的窗口，会提示还原的内容，接着按「下一步」继续。



(3) 选择还原目的位置，完成设定后请按「下一步」。

- a. 若要将备份文件的数据重新载回主机时，请选择「CPU」项目。
- b. 若选择「ISP 项目」时，请接着点击路径字段的 [...] 按钮以设定还原后的文件名与路径，而当设定的路径为已经存在相同文件名时，执行后该文件的内容将被还原的数据所覆盖。



(4) 确认摘要的内容与要进行的操作一致后，按下「执行」即可开始进行还原的动作。



当执行的操作是将数据还原至主机时，若用户按下等待界面的「取消」而中断了系统的动作，此时所还原的数据将可能不完整，若不再重新执行还原操作，请务必将 CPU 主机恢复为出厂默认值，以免造成运作上的错乱。若备份文件的来源位置是存储卡时，即使在软件端取消了还原动作，主机系统仍可能会继续执行还原的程序，此时可将主机断电以强制中断。

当还原的来源或目的位置有包含密码的信息或开启密码保护的功能时，其处理的机制如下。

操作项目	说明
存储卡 → CPU	<ul style="list-style-type: none"> a. 备份文件与 CPU 中的标识符必须一致，否则便中断程序。 b. 当 CPU 中有设定「PLC 密码」时，备份文件与 CPU 中的密码必须一致，否则便中断程序。 c. 当 CPU 没有设定「PLC 密码」，而备份文件有包含密码时，系统会执行还原并将备份文件的密码设为 CPU 的「PLC 密码」。
计算机端 → CPU	<ul style="list-style-type: none"> a. 备份文件与 CPU 中的标识符必须一致，否则便中断程序。 b. 当 CPU 中有设定「PLC 密码」时，备份文件与 CPU 中的密码必须一致，否则便中断程序并显示提示信息。 c. 当 CPU 没有设定「PLC 密码」，而备份文件有包含密码时，系统会执行还原并将备份文件的密码设为 CPU 的「PLC 密码」。
计算机端 → ISPSOft 项目	还原时，若备份文件中有设定「PLC 密码」，则会出现提示窗口询问用户输入，正确才可进行还原。还原后，备份文件中的项目密

操作项目	说明
	码与程序标识符将被还原到 ISPSOft 项目 ; 而备份文件中的「PLC 密码」、「PLC 标识符」则无法保存至 ISPSOft 项目。

(5) 完成操作后，接着便可选择回到主页面继续操作，或是直接关闭。



21

21.3.4 命令列模式指令操作说明

CARD Utility 支持透过命令列模式的指令操作，如需要使用可以到 ISPSOft 的安装目录下找到 PLC CARD.exe 的执行文件，默认安装路径为 C:\Program Files (x86)\Delta Industrial Automation\ISPSOft 3.10\PLC_CARD，接着按照下列说明操作。

- 指令与指令之间使用空白做为区隔。
- /Lock、/UnLock 不能同时使用。/F、/V 不能同时使用。/C 与 /IP 不能同时使用。
- /DLINK_USB, /DLINK_CDC, /DLINK_ETH 不能同时使用。
- 指令及参数不区分大小写，只有密码设定有区分。
- 不需要输入参数的指令

指令	功能
/RUN 或 /R	启动 PLC
/Stop	停止 PLC
/H	隐藏所有信息窗口，但程序结束会回传信息代码
/SaveOFF	关闭回传信息存盘功能

指令	功能
/?	显示帮助信息。使用此指令时，其他指令无效。
/DLINK_USB	指定使用 HMI DirectLink USB 方式通讯。此指令不能与/C、/IP、/DLINK_ETH、/DLINK_CDC 同时使用。只有 DVP 机种支持此指令。
/DLINK_CDC	指定使用 HMI DirectLink CDC 方式通讯。此指令必需配合/C 指令指定通讯端口。此指令不能与/IP、/DLINK_ETH、/DLINK_USB 同时使用。只有 DVP 机种支持此指令。
/DLINK_ETH	指定使用 HMI DirectLink Ethernet 方式通讯。此指令必需配合/IP 指令指定通讯地址。此指令不能与/C、/DLINK_CDC、/DLINK_USB 同时使用。只有 DVP 机种支持此指令。

● 需要输入参数的指令 (Pn 代表输入的参数，参数之间以"，"隔开)

指令	参数	功能
/PLC=P1	P1=PLC 型号	指定要通讯的 PLC 型号，例如：/PLC=AS332T。 支持的型号如下： AS300N · AS320T · AS320P · AS324MT · AS332T · AS332P。 AS218TX · AS218PX · AS218RX · AS228T · AS228P · AS228R · ES3。 ES Series: ES · EX · SS · EC。 EP Series: SA · SX · SC。 EH Series: EH · EH2 · SV · EH2-L。 ES2 Series: ES2 · EX2 · SX2 · SA2 · MC。 EH3 Series: EH3 · EH3L · SV2。 SE Series: SE。 ES2-E Series: ES2-E。 SS2 Series: SS2。
/F=P1	P1=文件名	下载 DUP 文件到 PLC，例如：/F=test.dup
/C=P1	P1=通讯端口编号	设定 PC 的 COM Port 通讯端口编号，例如：/C=1
/Lock=P1, P2	P1=要设定的密码。 P2=限制次数 (限制次数设 0 表示不限制)	PLC 无密码状态，且 DUP 文件中亦无密码，当下载完成之后，可以此指令设定 PLC 密码，例如 /Lock=1234, 0。(注：若 DUP 文件中有密码则下载完成后会自动将该密码设定成 PLC 密码)
/UnLock=P1	P1=验证的密码	PLC 已经被锁定密码，若 DUP 文件中无与 PLC 相同的密码，下载前须解除 PLC 密码，例如 /UnLock=1234。(注：解除 PLC 密码后，PLC 处于无密码状态，若要再次锁定密码需使用/Lock 指令或下载包含密码的 DUP 文件)

指令	参数	功能
/P=P1, P2, P3, P4	P1=Baud Rate P2=Data Bits P3=Parity P4=Stop Bits	设定通讯协议·默认值为 9600, 7, e, 1 不更改的值不用输入·但是","一定要输入。 ex1 : 全部都要更改 /p=115200, 8, n, 2 ex2 : 只更改 Baud Rate /p=115200, ,, " ,"一定要输入
/Save=P1	P1=文件名	指定回传信息代码储存路径及文件名。 指定路径目录不存在时·会自动产生该目录·未使用此指令或指定参数为空时 (/save=) 以预设文件名 "Result.txt"存在当前路径。 Ex : /Save=C:\test\result1.txt 储存内容请参考下面所列之回传代码表。
/IP=P1	P1=IP address	有指定/IP 指令时·表示以 Ethernet 方式通讯·P1 设定通讯之 IP address。此指令不能与/C 参数同时使用。 Ex: /IP=192.168.1.5
/T=P1,P2	P1=转换来源文件 P2=转换目标文件	DVZ/DUP 文件格式转换。使用此指令时·其他指令无效。只有 DVP 机种支持此指令。
/V=P1	P1=要与 PLC 内部程序做验证的文件名	验证 PLC 程序。只有 DVP 机种支持此指令。 Ex: /V=test.dup
/STN=P1	P1=站号	设定计算机端使用之通讯的站号·例如 :/STN=5 表示要对站号为 5 的 PLC 通讯·省略本指令时使用广播方式通讯。
/SetPLCSTN=P1	P1=站号	更改 PLC 的站号·例如 :/SetPLCSTN=5 表示要将 PLC 的站号修改成 5。只有 DVP 机种支持此指令。
/B=P1,P2,P3,P4	P1=另存 DUP 文件名, P2=验证的密码, P3=验证的标识符 P4=要不要备份装置 (N : 不要· Y:要)	读出 PLC 内部的程序及装置内容·并另存成 DUP 檔。 只有 DVP 机种支持此指令。 Ex1: PLC 内部无密码也无 ID·不备份装置 /B=p1.dup,,,N Ex2 : PLC 内部密码锁定但无 ID·要备份装置 /B=p1.dup,pw99,,Y P4 为空白时·默认为要备份装置
/BP=P1,P2,P3,P4	P1=目的 DUP 文件名 P2=来源项目文件名 (dvp/isp) P3=来源寄存器文件 (dvl) P4=来源文件寄存器 (wft)	将指定的项目文件备份成 DUP 檔·P1 及 P2 一定要输入·P3 及 P4 可省略。只有 DVP 机种支持此指令。 例如 : 将 test.dvp 备份成 out.dup /BP=out.dup,test.dvp /BP=d:\dir\out.dup,d:\dir1\test.dvp 指定的目录不存在会自动建立目录·文件已存在时会被覆盖。

范例：

Ex1 :经由 COM port 1 将 test.dup 下载到 PLC AS332T 并执行 PLC 通讯协议使用默认值 9600, 7, e, 1

- 指定位于当前路径下的文档："PLC CARD.exe" /PLC=AS332T /F=test.dup /RUN /C=1
- 指定位于特定路径下的文档："d:\temp\PLC CARD.exe" /PLC=AS332T /F=d:\temp\test.dup /RUN /C=1

Ex2 : 经由 Ethernet 停止 PLC AS324MT

"PLC CARD.exe" /PLC=AS324MT /Stop /IP=192.168.1.1

Ex3: 经由 COM port 1 将 PLC 的站号修改成 2 (原本 PLC 站号为 1)

"\PLC CARD.exe" /PLC=SS2 /C=1 /STN=1 /SetPLCSTN=2

回传代码表

代码	信息
1	正常结束!
2	通讯超时!
3	PLC 的型号不符!
4	DUP 文件内容错误!
5	通讯端口已被使用或无此通讯端口!
6	无法开启文档!
7	命令错误!
8	命令 H 错误!
9	命令 P 错误!
10	站号错误!
11	通讯端口号码错误!
12	命令 R 错误!
13	波特率错误!
14	数据长度错误!
15	奇偶校验错误!
16	停止位错误!
17	命令 STOP 错误!
18	命令 V 错误!
19	密码已上锁!
20	程序比对错误!
21	PLC 中无执行码可比对!
22	保留
23	命令 C 错误!
24	命令 B 错误!
25	命令 Lock 错误!

代码	信息
26	命令 UnLock 错误!
27	密码验证失败!
28	储存备份文件失败!
29	PLC 已上锁·请先移除密码!
30	命令 IP 错误!
31	PLC 中无执行码可读取!
32	无法取得正确的 PLC ID 状态!
33	PLC ID 比对失败!
34	通讯错误!
35	下载次数已达上限!
36	不支持该型号!
37	读取 PLC 内容失败!
38	写入 PLC 失败!
39	命令失败!
40	命令 PLC 错误!
41	用户取消命令!
42	命令 STN 或 SetPLCSTN 错误!
43	命令 BP 错误!
44	EasyDownload 版本错误!

MEMO

21



第22章 G-Code 编辑器与 E-CAM 编辑器

目录

22.1 G-Code 编辑器	22-2
22.1.1 关于 G-Code	22-2
22.1.2 使用 G-Code 编辑器	22-2
22.1.3 G-Code 的指令结构	22-7
22.1.4 G-Code 功能详细介绍	22-9
22.1.4.1 G0 (快速定位)	22-9
22.1.4.2 G1 (直线插补)	22-11
22.1.4.3 G2 (顺时针圆弧插补/螺旋插补)	22-13
22.1.4.4 G3 (逆时针圆弧插补/螺旋插补)	22-16
22.1.4.5 G4 (时间延时)	22-19
22.1.4.6 G17/G18/G19 (平面选择)	22-20
22.1.4.7 G50 (准确停止)	22-21
22.1.4.8 G51 (圆弧交接)	22-22
22.1.4.9 G52 (圆滑交接)	22-23
22.1.4.10 G90 (绝对模式)	22-25
22.1.4.11 G91 (相对模式)	22-26
22.1.4.12 M 码	22-27
22.1.5 G-Code 范例-三轴同动绘出 Delta LOGO	22-29
22.2 E-CAM 编辑器	22-30
22.2.1 关于 E-CAM (电子凸轮)	22-30
22.2.2 电子凸轮意义说明	22-30
22.2.3 使用 E-CAM 编辑器	22-31

22.1 G-Code 编辑器

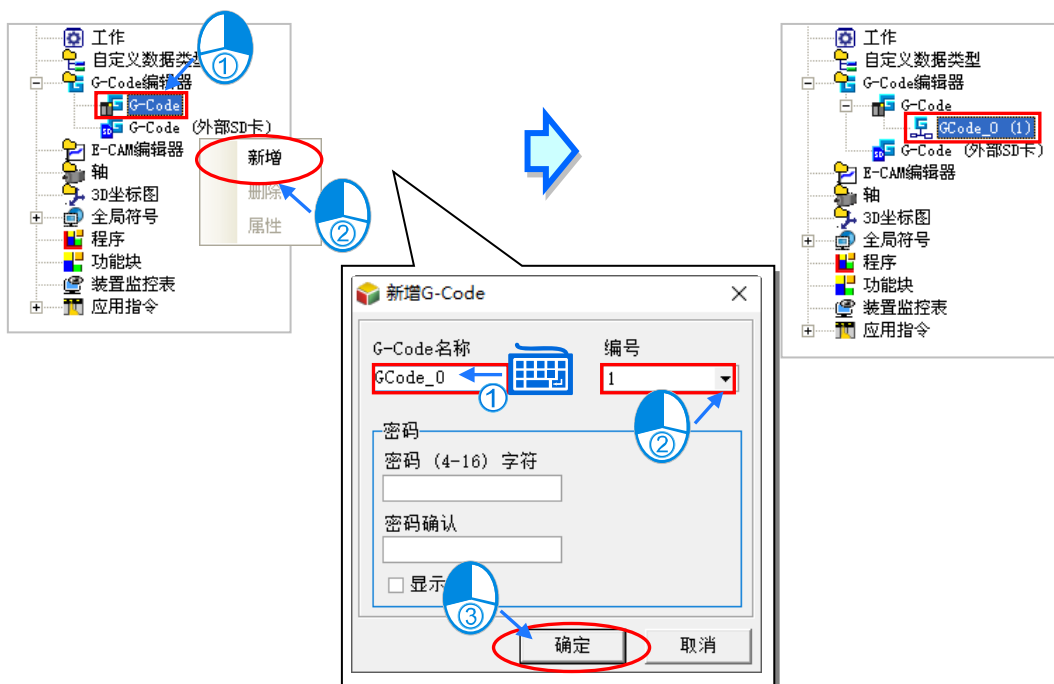
22.1.1 关于 G-Code

进行运动控制时，为了在两轴或两轴以上进行复杂的加工轨迹，用户可以通过计算机辅助制造软件（Computer Aided Manufacture）将所需加工路径生成 G-Code，再将 G-Code 输入控制器，来驱动各轴加工。由此可以节省运动程序开发的时间。G-Code 是一种广泛用于自动化设备的 CNC（Computer Numerical Control）程序语言。ISPSOft 支持市面上常用的大多数 G-Code，而不支持的 G-Code 则采略过不执行的方式处理。

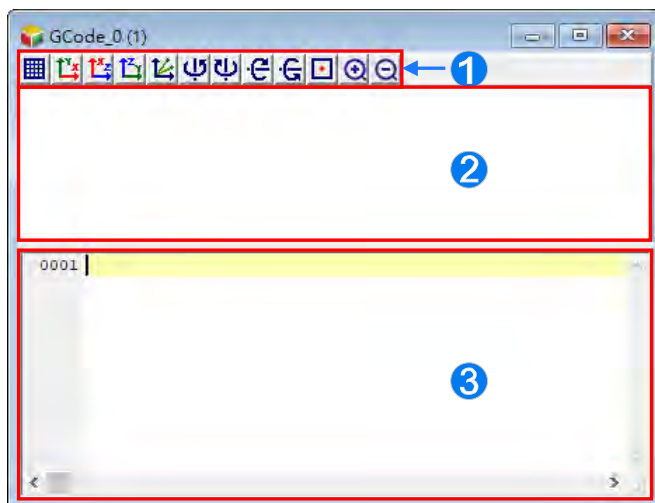
22

22.1.2 使用 G-Code 编辑器

项目管理区的 G-code 编辑器项目下会有两个子项目，下载时分别会配置到不同的位置，G-Code 项目将存放于主机；而 G-code（外部 SD 卡）则会存放至主机上安装的 SD 储存卡中（目前 ISPSOft 针对 DVPxxMC 机种不支持此功能）。在选定的项目单击 **右键** > **新增**，出现新增窗口如下图，在 G-Code 名称输入此项目名称，再选择此项目的编号，若需要可以设定密码保护 G-Code 设计数据，下次开启编辑窗口时将会询问密码。主机上最多可建立 64 组 G-code 项目，每组最大 250KB。



完成新建后将会在编辑工作区开启 G-Code 编辑窗口，功能说明如下：



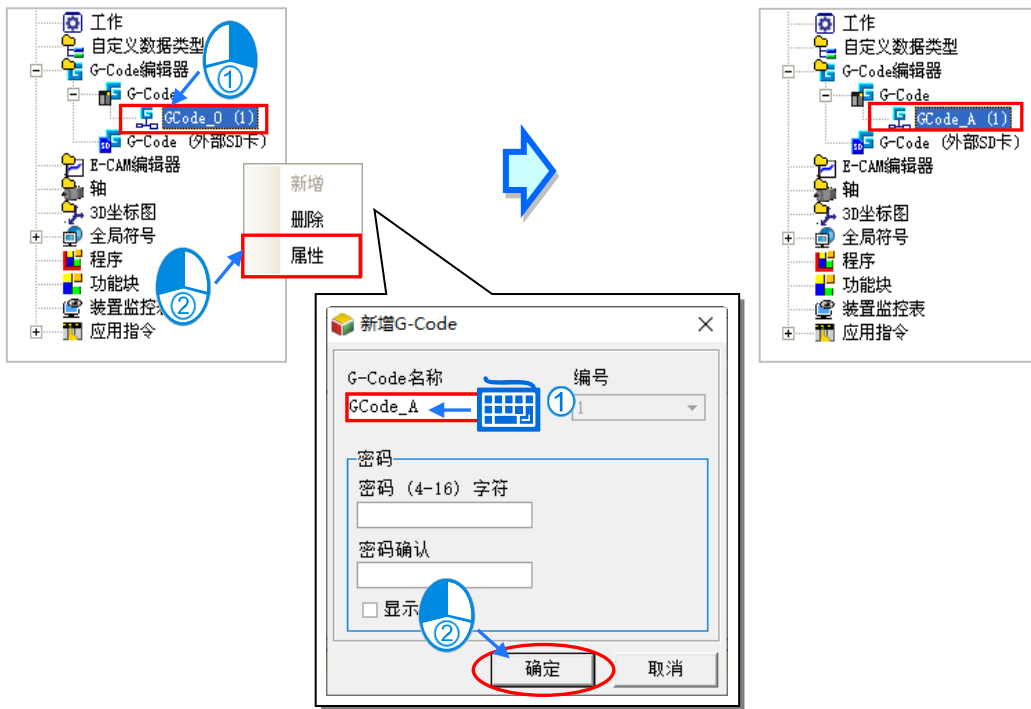
22

- ❶ 图标工具栏：提供各项功能按钮，对 G-Code 轨迹示意区画面进行调整，按钮功能说明如下。
- ❷ 轨迹示意区：依据 G-Code 程序编写区中的程序，实时显示对应的路径轨迹。
- ❸ G-Code 程序编写区：在此处输入 G-Code。

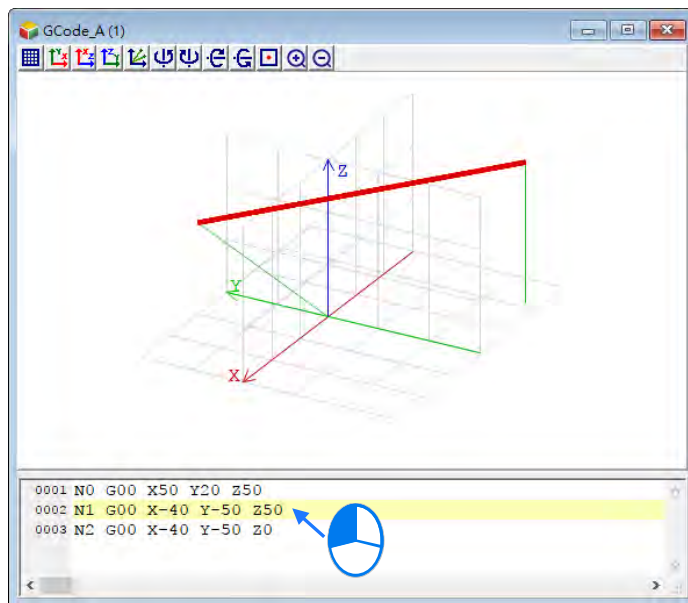
图标工具栏按钮功能说明

图标	名称	功能说明
	开启/关闭网格线	显示轨迹示意区网格线，分别开启/关闭 XY、YZ、XY 平面的网格线。
	切换到 XY 平面	切换轨迹示意区到 XY 平面
	切换到 XZ 平面	切换轨迹示意区到 XZ 平面
	切换到 YZ 平面	切换轨迹示意区到 YZ 平面
	显示 G0 轨迹	显示 G0 多轴高速定位的轨迹
	垂直逆时针旋转	对目前轨迹示意区垂直轴逆时针旋转
	垂直顺时针旋转	对目前轨迹示意区垂直轴顺时针旋转
	水平往上旋转	对目前轨迹示意区水平轴往上旋转
	水平往下旋转	对目前轨迹示意区水平轴往下旋转
	最适大小	调整轨迹示意区到最适大小
	放大	放大轨迹示意区图形
	缩小	缩小轨迹示意区图形

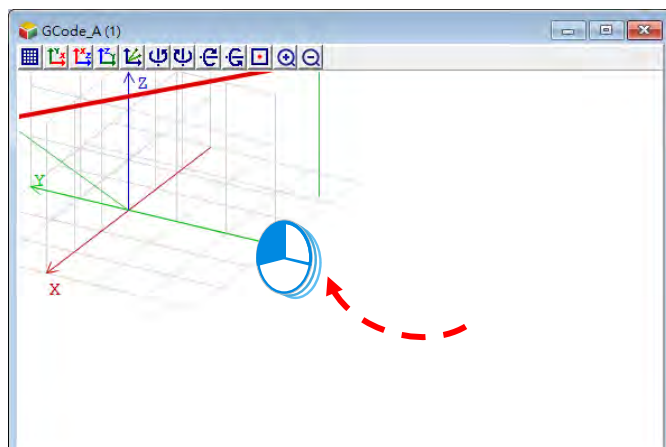
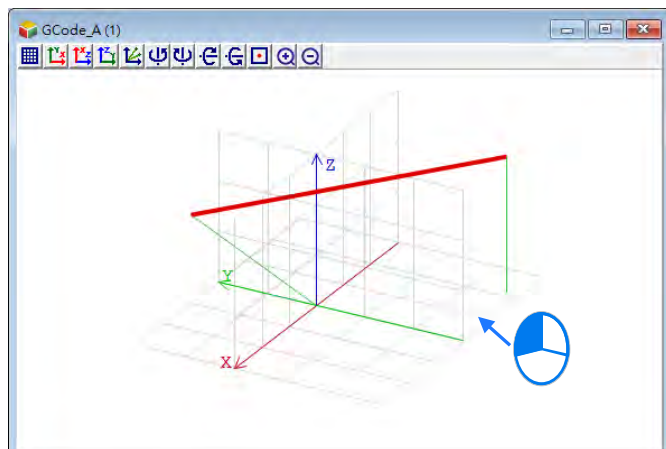
在已建立的 G-Code 项目上单击鼠标右键，可在快捷选项单击「删除」，删除此项目；或是单击「属性」，重新设定项目名称或密码，但不可变更编号。



当 G-Code 程序编写区单击某一行程序时，轨迹示意区对应的 G-Code 轨迹将以红色线段显示。



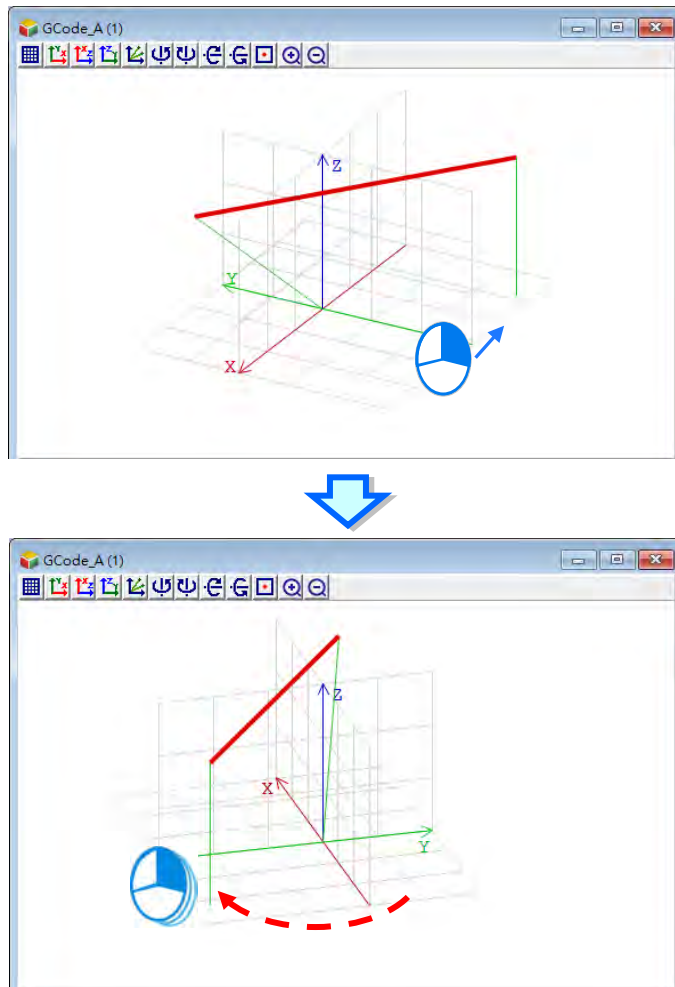
用户可以在轨迹示意区按住鼠标左键，拖曳鼠标移动画面显示的区域。



22

用户可以在轨迹示意区按住鼠标右键，拖曳鼠标旋转画面显示的区域。

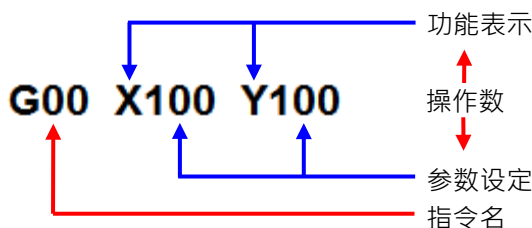
22



完成设定后便可与项目一同下载至主机，项目的上下载请参考第 17.1.3 节。

22.1.3 G-Code 的指令结构

G-Code 指令的结构可分为**指令名**及**操作数**；操作数又包括了**功能表示**与**参数设定**两部分，如下所示。指令名表示指令执行的功能；功能表示则为操作数执行此指令的目标，例如目标轴位置或速度；参数设定则为操作数参数的数值。以下 G00 指令，为以最快速度移动 X 轴与 Y 轴到指定的目标位置(100，100)。



22

所支持的 G-Code 指令的格式与用法。以下列出几点基本事项。

- 每一列 G-Code 开头都必须有一行号，例如 N1，N2，...，等等
- 在不破坏指令格式下，是否有空白皆是被允许的。例如 G01X50 Y15F20 合法且相同于 G01 X50 Y15 F20。
- G00、G01、G02、G03 指令有继承功能（即下一行省略 G 代码可比照之前的 G 代码）其它种类的 G 代码无继承功能；G01、G02、G03 速度参数 F 有延续功能。如下所示：

```

N0 G00 X50 Y15
N1 X-40 Y-50 ← G00 X-40 Y-50

N2 G01 X10 Y25 F20
N3 G03 X-40 Y-50 R100
      ↑
      G03 X-40 Y-50 R100 F20
  
```

- 运算符参数常数可输入型式：阿拉伯数字及浮点数。
例：N0 G0 X100.0 Y100.0
- 高速定位指令 (G0) 不需设定移动速度。
例：N0 G0 X100.2 Y500.0
- G00、G01 指令中，轴参数 X、Y、Z 至少存在一轴 (参数)，其它省略的轴参数值参照前面最接近的正确行中相同参数的值。
- 注释支援方式有两种：
 - 以 (和) 包覆，且不允许巢状注解，例：(这是注释)
 - 在 % 后的文字，例：%这是注释

- 编写指令方式为更符合一般 G 代码，运算元参数标记的顺序可不用按照 X、Y、Z、I、J、K、R 等顺序编写，可任意编写顺序。

例如：N0 G1 X100 Z300 Y200.45 F400 => N0 G1 X100 Y200.45 Z300 F400

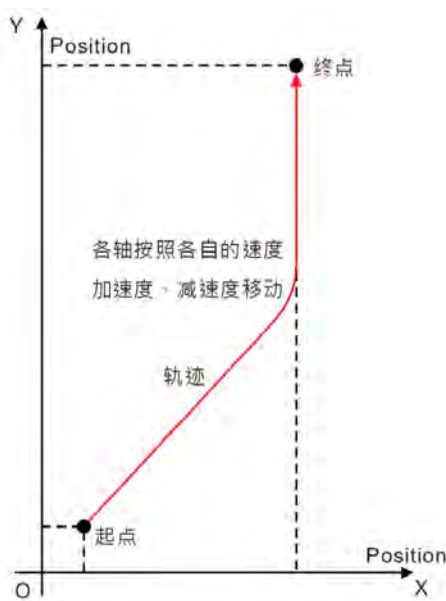
DVPxxMC 系列运动控制器支持的 G 码及输入格式如下表所示。

G 代码	功能说明	格式
G0	快速定位	N_G0 X_Y_Z_A_B_C_
G1	直线插补	N_G1 X_Y_Z_A_B_C_F_
G2	顺时针圆弧插补/螺旋插补	N_G2 X_Y_Z_I_J_F_
		N_G2 X_Y_Z_I_K_F_
		N_G2 X_Y_Z_J_K_F_
		N_G2 X_Y_Z_R_F_
G3	逆时针圆弧插补/螺旋插补	N_G3 X_Y_Z_I_J_F_
		N_G3 X_Y_Z_I_K_F_
		N_G3 X_Y_Z_J_K_F_
		N_G3 X_Y_Z_R_F_
G4	时间延时	N_G4 X_ N_G4 P_
G17	XY 平面选择	N_G17
G18	ZX 平面选择	N_G18
G19	YZ 平面选择	N_G19
G50	准确停止	N_G50
G51	圆弧交接	N_G51 D_
G52	圆滑交接	N_G52
G90	绝对模式	N_G90
G91	相对模式	N_G91
M0~M99	M 码	N_M_D_

22.1.4 G-Code 功能详细介绍

22.1.4.1 GO (快速定位)

- 功能：
各轴以指定速度从当前位置运动到终点位置。最多可控制八个轴，运动过程中各轴是相互独立的，运动轨迹如下所示。



22

- 格式：
`N_G0 X_Y_Z_A_B_C_`
- 参数说明：
N_：G 码在程序中所在行的编号。
X_：指定 X 轴终点位置，单位：单元，数据类型：REAL。
Y_：指定 Y 轴终点位置，单位：单元，数据类型：REAL。
Z_：指定 Z 轴终点位置，单位：单元，数据类型：REAL。
A_：指定 A 轴终点位置，单位：单元，数据类型：REAL。
B_：指定 B 轴终点位置，单位：单元，数据类型：REAL。
C_：指定 C 轴终点位置，单位：单元，数据类型：REAL。
- 指令说明：
 1. G0 可以控制其中一个轴或多个轴，其它轴都为默认值。
 2. 各轴运动时的速度、加速度、减速度和加速度的变化率可以通过 `DMC_SetG0Para` 指令来设定这些参数。

● 范例：

X、Y 轴的初始位置都为 10000 单元，轴参数都为默认值。

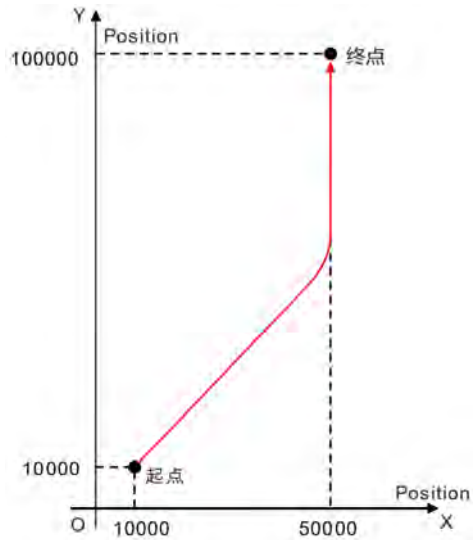
将要执行的 G 码如下：

N00 G90

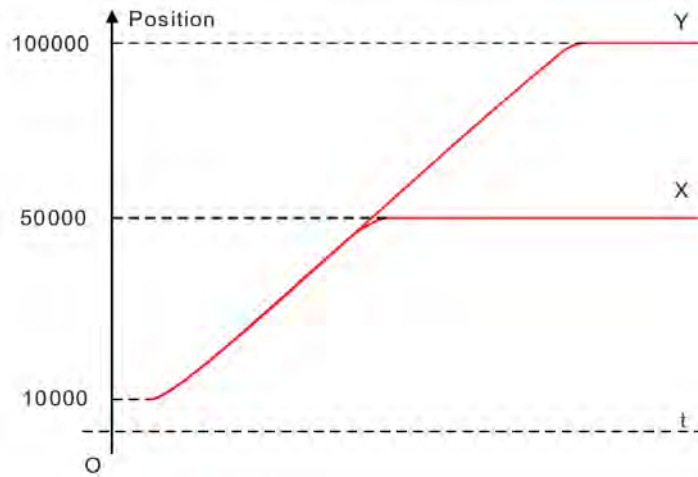
N01 G0 X50000 Y100000

G 码执行后，整个过程的 Y/X 曲线如下：

22



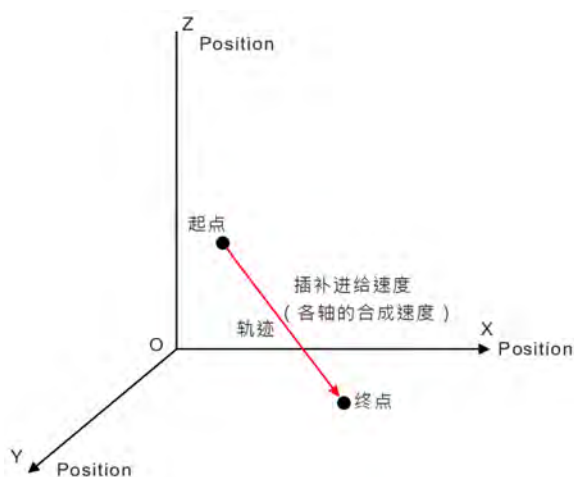
G 码执行后，整个过程的位置/时间曲线如下：



22.1.4.2 G1 (直线插补)

● 功能：

刀具以指定速度从某一点出发，直线移动到目标位置。此指令最多可控制八个轴，且各轴同起同停。如下图所示，三个轴一起控制刀具的位置。



22

● 格式：

`N_G1 X_Y_Z_A_B_C_F_`

● 参数说明：

`N_`：G 码在程序中所在行的编号。

`X_`：指定 X 轴终点位置，单位：单元，数据类型：REAL。

`Y_`：指定 Y 轴终点位置，单位：单元，数据类型：REAL。

`Z_`：指定 Z 轴终点位置，单位：单元，数据类型：REAL。

`A_`：指定 A 轴终点位置，单位：单元，数据类型：REAL。

`B_`：指定 B 轴终点位置，单位：单元，数据类型：REAL。

`C_`：指定 C 轴终点位置，单位：单元，数据类型：REAL。

`F_`：指定刀头的进给速度，单位：单元/秒，数据类型：REAL。

刀头匀速时，G 码中所有轴的合成速度与 F 值相等。计算方法如下：

若存在两个轴时，

$$F = \sqrt{V_1^2 + V_2^2}$$

若存在三个轴时，

$$F = \sqrt{V_1^2 + V_2^2 + V_3^2}$$

若存在更多轴时，按照上述方法类推。

- 指令说明：
 1. G0 可以控制其中一个轴或多个轴，其它轴都为默认值。
 2. 刀头的速度、加速度、减速度和加速度的变化率可以通过 DMC_SetG1Para 指令来设定这些参数。

- 范例：

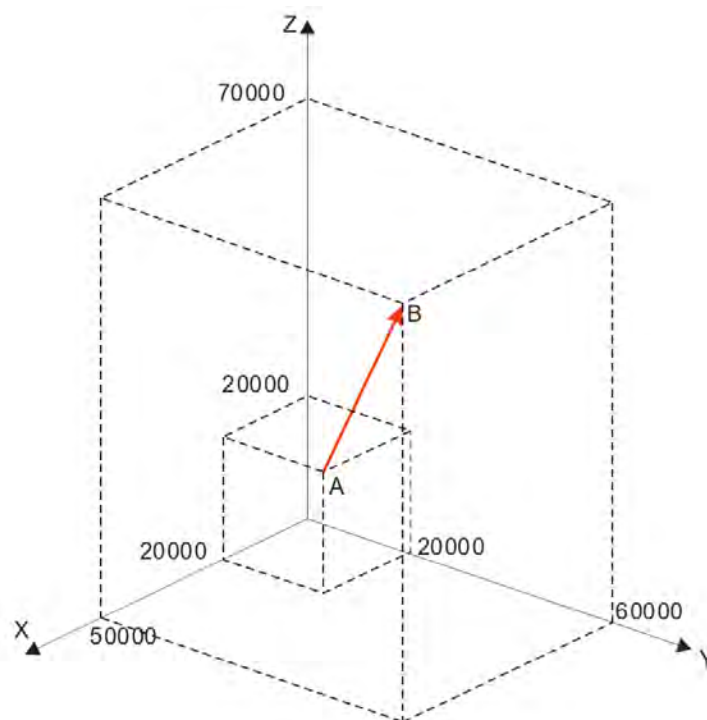
X、Y、Z 轴的初始位置都为 20000 单元，轴参数都为默认值。

将要执行的 G 码如下：

```
N00 G90
```

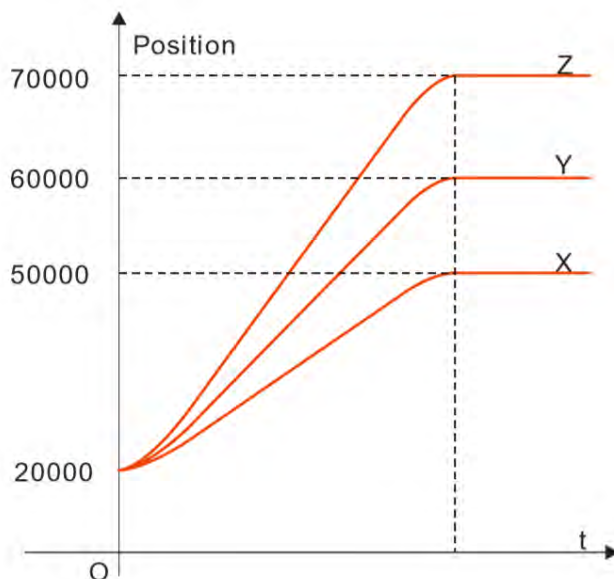
```
N01 G1 X50000 Y60000 Z70000
```

G 码执行后，整个过程的曲线如下：



22

G 码执行后，整个过程的位置/时间曲线如下：



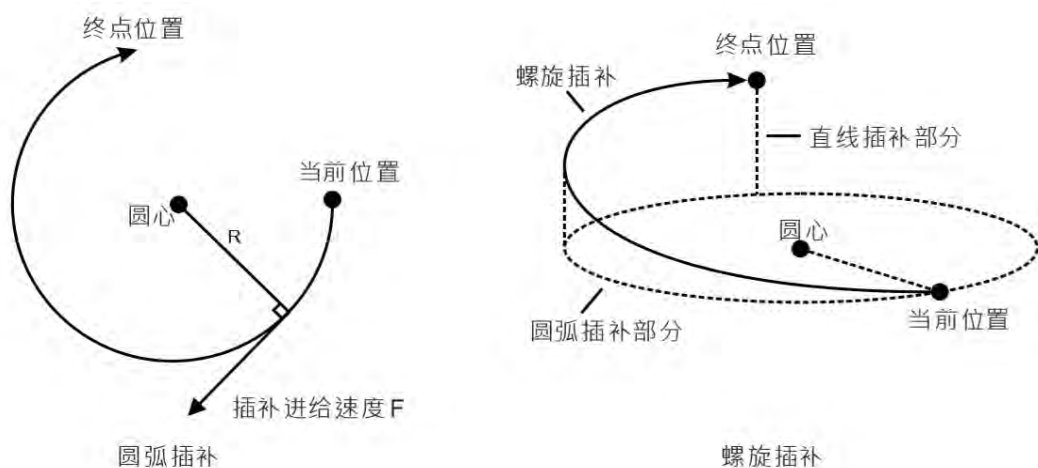
22

22.1.4.3 G2 (顺时针圆弧插补/螺旋插补)

- 功能：

圆弧插补：刀具在指定平面及由圆心或半径确定的圆弧上，以参数 F 设定的进给速度，对加工对象进行顺时针方向圆弧切削。

螺旋插补：刀具在指定的平面及圆弧上顺时针方向移动（圆弧插补）时，同时在指定平面的垂直方向上进行直线移动（直线插补），以参数 F 设定的进给速度。



- 格式：

格式 1：N_G2 X_Y_Z_I_J_F_

格式 2：N_G2 X_Y_Z_I_K_F_

格式 3 : N_G2 X_Y_Z_J_K_F_

格式 4 : N_G2 X_Y_Z_R_F_

● 参数说明 :

N_ : G 码在程序中所在行的编号。

X_ : 指定 X 轴终点位置, 单位: 单元, 数据类型: REAL。

Y_ : 指定 Y 轴终点位置, 单位: 单元, 数据类型: REAL。

Z_ : 指定 Z 轴终点位置, 单位: 单元, 数据类型: REAL。

I_J_ : 指定 XY 平面的圆心坐标位置, 单位: 单元, 数据类型: REAL。

I_K_ : 指定 XZ 平面的圆心坐标位置, 单位: 单元, 数据类型: REAL。

J_K_ : 指定 YZ 平面的圆心坐标位置, 单位: 单元, 数据类型: REAL。

R_ : 指定弧形半径, 数据类型: REAL。

F_ : 指定刀头的进给速度, 单位: 单元/秒, 数据类型: REAL。

刀头匀速时, G 码中所有轴的合成速度与 F 值相等。计算方法如下:

若存在两个轴时,

$$F = \sqrt{V_1^2 + V_2^2}$$

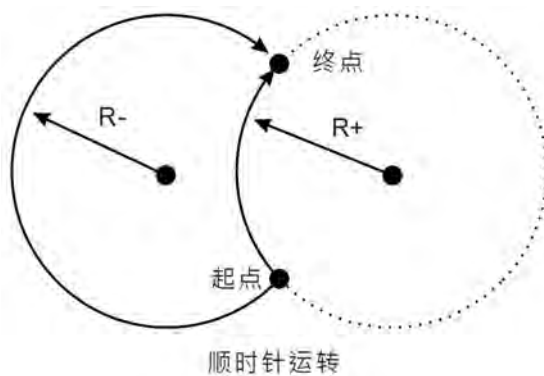
若存在三个轴时,

$$F = \sqrt{V_1^2 + V_2^2 + V_3^2}$$

若存在更多轴时, 按照上述方法类推。

● 指令说明 :

1. X、Y、Z 中的两个轴在指定平面 (平面由 G17/G18/G19 指令指定平面) 上做圆弧插补运动, 第 3 个轴垂直指定平面做直线插补运动。
2. F_可以省略。在 G-Code 程序编写区内若仅有一行指令, 省略 F 参数后, 刀头的速度、加速度、减速度由轴组参数决定, 可以通过 DMC_SetG1Para 指令来设定这些参数; 若 G-Code 程序编写区有多行指令且 G2 省略 F 参数, 则刀头的速度、加速度、减速度以刀头之前所设定的 F 参数为准, 若都没有指定 F 参数, 则以轴组参数中的速度、最大加速度、最大减速度为准。
3. 在 G90 绝对模式时, 圆弧终点是以各自方向上的 0 单元为参考的绝对坐标值。在 G91 相对模式时, 圆弧终点是相对圆弧起点的增量值。
4. 不管是绝对还是相对模式, 圆心坐标 I_J_ (I_K_/J_K_) 始终是以起点为参考的相对坐标。
5. 坐标平面与 I、J、K 的关系, 一条圆弧指令中最多出现 I、J、K 中的两个, 至于出现哪两个则由相应的平面决定, 如 XY 平面只能出现 I、J。
6. 格式 4 与格式 1、2、3 的不同在于, 它通过起点、终点和半径来确定一段圆弧。R 参数后面输入正数时 (R+) 代表选择劣弧 (小于 180 度), 输入负数时 (R-) 代表选择优弧 (大于 180 度)。下图所示实线部分为 G2 指令选择 R+和 R-时的运行轨迹, 圆弧上的箭头表示运转方向:



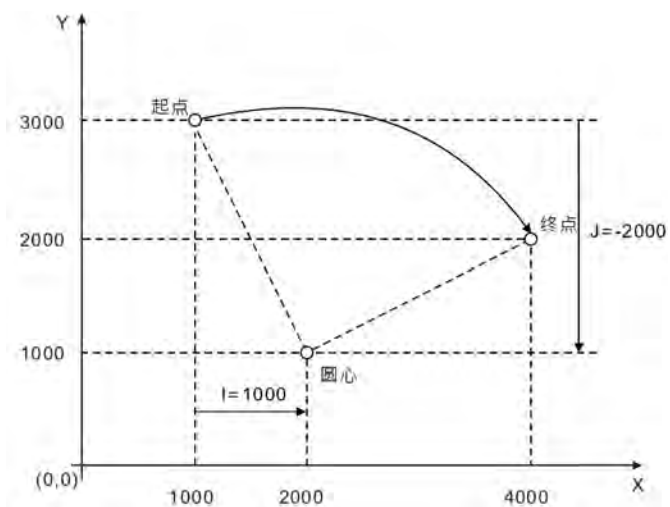
- 范例：

当前位置为 (1000 · 3000)，轴参数都为默认值，

将要执行的 G 码如下：

```
N00 G90
N01 G17
N02 G2 X4000 Y2000 I1000 J-2000 F5000
```

G 码执行后，整个过程的 Y/X 曲线如下：

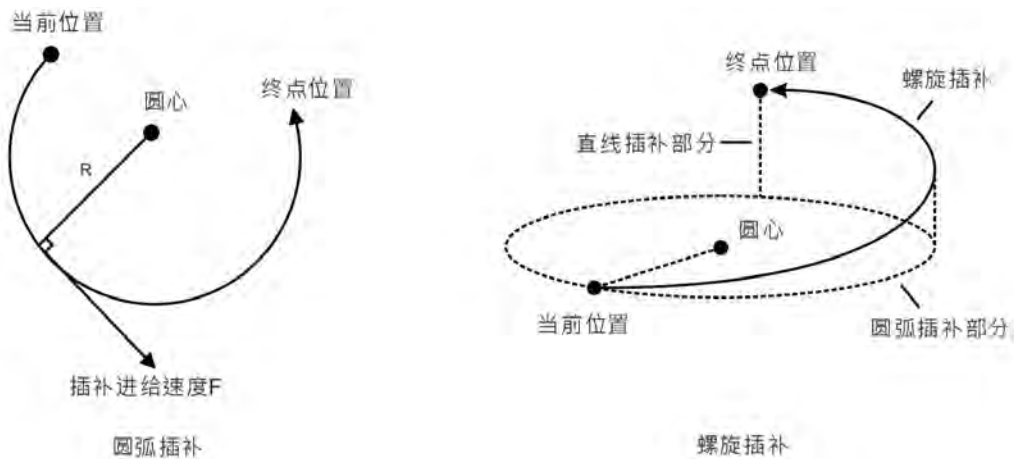


22.1.4.4 G3 (逆时针圆弧插补/螺旋插补)

- 功能：

圆弧插补：刀具在指定平面及由圆心或半径确定的圆弧上，以参数 F 设定的进给速度，对加工对象进行逆时针方向圆弧切削。

螺旋插补：刀具在指定的平面及圆弧上逆时针方向移动（圆弧插补）时，同时在指定平面的垂直方向上进行直线移动（直线插补），以参数 F 设定的进给速度。



- 格式：

格式 1：N_G2 X_Y_Z_I_J_F_

格式 2：N_G2 X_Y_Z_I_K_F_

格式 3：N_G2 X_Y_Z_J_K_F_

格式 4：N_G2 X_Y_Z_R_F_

- 参数说明：

N_：G 码在程序中所在行的编号。

X_：指定 X 轴终点位置，单位：单元，数据类型：REAL。

Y_：指定 Y 轴终点位置，单位：单元，数据类型：REAL。

Z_：指定 Z 轴终点位置，单位：单元，数据类型：REAL。

I_J_：指定 XY 平面的圆心坐标位置，单位：单元，数据类型：REAL。

I_K_：指定 XZ 平面的圆心坐标位置，单位：单元，数据类型：REAL。

J_K_：指定 YZ 平面的圆心坐标位置，单位：单元，数据类型：REAL。

R_：指定弧形半径，数据类型：REAL。

F_：指定刀头的进给速度，单位：单元/秒，数据类型：REAL。

刀头匀速时，G 码中所有轴的合成速度与 F 值相等。计算方法如下：

若存在两个轴时，

$$F = \sqrt{V_1^2 + V_2^2}$$

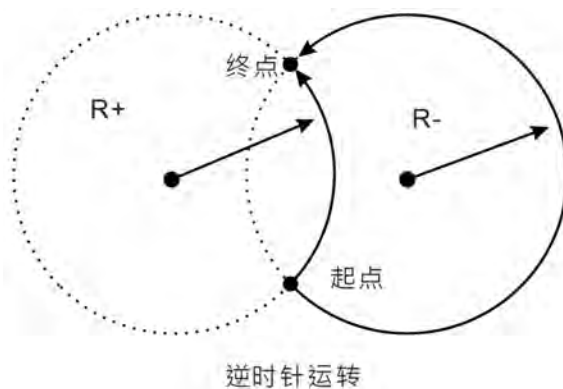
若存在三个轴时，

$$F = \sqrt{V_1^2 + V_2^2 + V_3^2}$$

若存在更多轴时，按照上述方法类推。

● 指令说明：

1. X、Y、Z 中的两个轴在指定平面（平面由 G17/G18/G19 指令指定平面）上做圆弧插补运动，第 3 个轴垂直指定平面做直线插补运动。
2. F_可以省略。在 G-Code 程序编写区内若仅有一行指令，省略 F 参数后，刀头的速度、加速度、减速度由轴组参数决定，可以通过 DMC_SetG1Para 指令来设定这些参数；若 G-Code 程序编写区有多行指令且 G3 省略 F 参数，则刀头的速度、加速度、减速度以刀头之前所设定的 F 参数为准，若都没有指定 F 参数，则以轴组参数中的速度、最大加速度、最大减速度为准。
3. 在 G90 绝对模式时，圆弧终点是以各自方向上的 0 单元为参考的绝对坐标值。在 G91 相对模式时，圆弧终点是相对圆弧起点的增量值。
4. 不管是绝对还是相对模式，圆心坐标 I_J_ (I_K_/J_K_) 始终是以起点为参考的相对坐标。
5. 坐标平面与 I、J、K 的关系，一条圆弧指令中最多出现 I、J、K 中的两个，至于出现哪两个则由相应的平面决定，如 XY 平面只能出现 I、J。
6. 格式 4 与格式 1、2、3 的不同在于，它通过起点、终点和半径来确定一段圆弧。R 参数后面输入正数时 (R+) 代表选择劣弧 (小于 180 度)，输入负数时 (R-) 代表选择优弧 (大于 180 度)。下图所示实线部分为 G2 指令选择 R+和 R-时的运行轨迹，圆弧上的箭头表示运转方向：



● 范例：

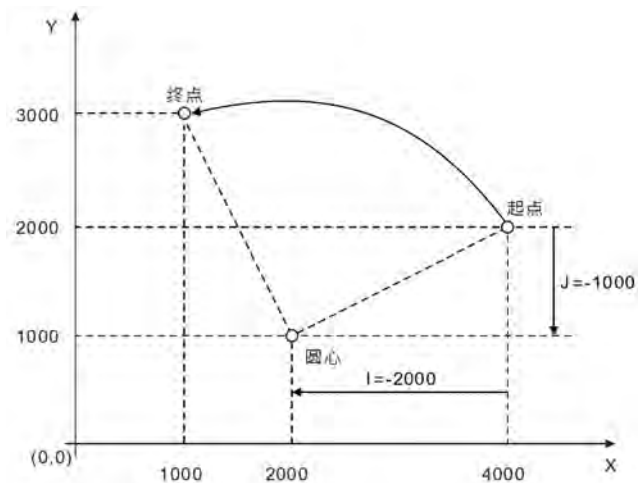
当前位置为 (4000 · 2000) · 轴参数都为默认值。

将要执行的 G 码如下：

N00 G90

N01 G17 N02 G3 X1000 Y3000 I-2000 J-1000

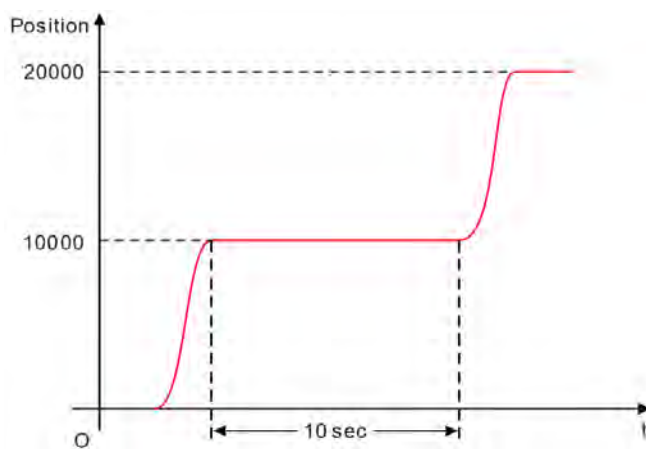
G 码执行后 · 整个过程的 Y/X 曲线如下：



22

22.1.4.5 G4 (时间延时)

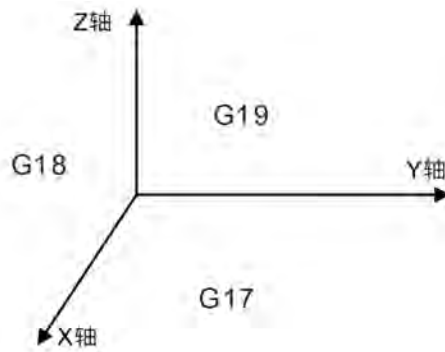
- 功能：
延时指令。
- 格式：
格式 1 : N_G4 X_
格式 2 : N_G4 P_
- 参数说明：
N_ : G 码在程序中所在行的编号。
X_ : 指定延时时间，单位：秒，数据类型：REAL。
P_ : 指定延时时间，单位：毫秒，数据类型：REAL。
- 范例：
N00 G1 X10000
N01 G4 X10 (或 N01 G4 P10000)
N02 G1 X20000
G 码执行后，整个过程的位置/时间曲线如下：



22.1.4.6 G17/G18/G19 (平面选择)

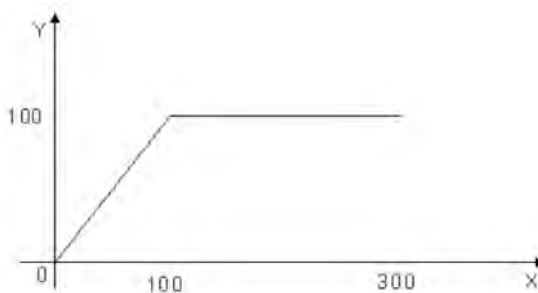
- 功能：
此三个指令用于决定圆弧/螺旋插补平面选择，对于直线插补无影响。
程序执行时，此三个工作平面可相互切换。程序中若无设定任何平面选择，系统初始状态为 XY 平面 (G17) 状态。
- 格式：
N_G17
N_G18
N_G19
- 参数说明：
N_：G 码在程序中所在行的编号。
- 平面示意图如下图所示：

22



22.1.4.7 G50 (准确停止)

- 功能：
改变交接模式为准确停止，之后的交接模式都是准确停止，执行过程中可以使用 G51/G52 进行交接模式切换。每条 G 码之间，终端机构的速度都会减为 0。
- 格式：
N_G50
- 参数说明：
N_：G 码在程序中所在行的编号。
- 范例：
N00 G50
N01 G1 X100 Y100
N02 G1 X300 Y100
G 码执行后，整个过程的 Y/X 曲线如下：



22.1.4.8 G51 (圆弧交接)

- 功能：

改变交接模式为圆弧交接，之后的交接模式都是圆弧交接，执行过程中可以使用 G50/G52 进行交接模式切换。每条 G 码之间交接，终端机构的不减速，交接曲线为圆弧。在该模式下，整体运行速度以第一条 G 码的速度为准，无法在运行中通过 F 分量变更速度。如果需要变更速度，可以通过 DMC_CartesianCoordinate 的 VelOverride 脚位变更速度超调量来控制终端机构的运行速度。

- 格式：

N_G51 D_

- 参数说明：

N_：G 码在程序中所在行的编号。

D_：圆弧的半径。

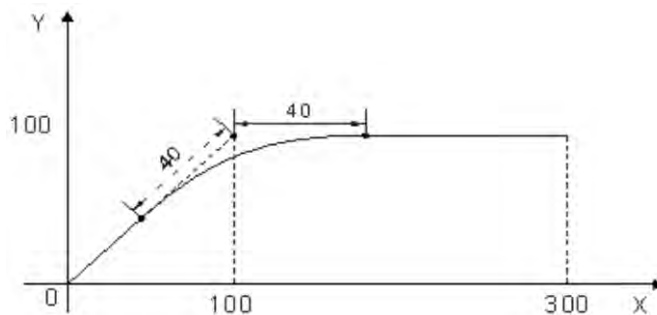
- 范例：

```
N00 G51 D40
```

```
N01 G1 X100 Y100
```

```
N02 G1 X300 Y100
```

G 码执行后，整个过程的 Y/X 曲线如下：



22.1.4.9 G52 (圆滑交接)

- 功能：

改变交接模式为圆滑交接，之后的交接模式都是圆滑交接，执行过程中可以使用 G50/G51 进行交接模式切换。每条 G 码之间交接，终端机构的不减速，适用于小线段连续插补。在该模式下，整体运行速度以第一条 G 码的速度为准，无法在运行中通过 F 分量变更速度。如果需要变更速度，可以通过 DMC_CartesianCoordinate 的 VelOverride 脚位变更速度超调量来控制终端机构的运行速度。。

- 格式：

N_G52

- 参数说明：

N_：G 码在程序中所在行的编号。

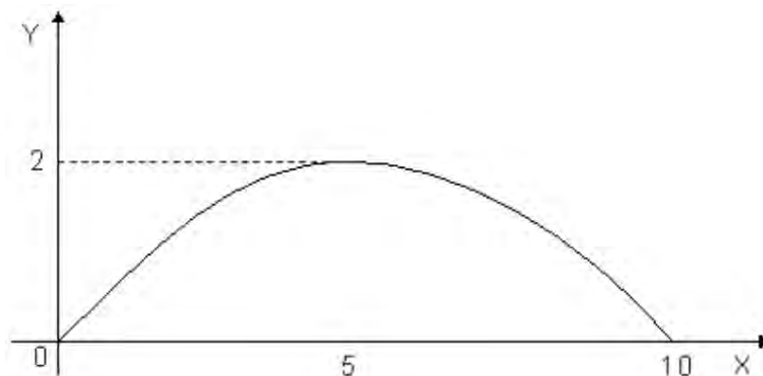
- 范例：

N00 G52	N01 G1 X0 Y0 E5 E-5 F5
N02 G1 X0.1 Y0.06282151816	N03 G1 X0.2 Y0.1255810391
N04 G1 X0.3 Y0.1882166266	N05 G1 X0.4 Y0.2506664671
N06 G1 X0.5 Y0.3128689301	N07 G1 X0.6 Y0.3747626292
N08 G1 X0.7 Y0.4362864828	N09 G1 X0.8 Y0.4973797743
N10 G1 X0.9 Y0.5579822121	N11 G1 X1 Y0.6180339887
N12 G1 X1.1 Y0.6774758405	N13 G1 X1.2 Y0.7362491054
N14 G1 X1.3 Y0.7942957813	N15 G1 X1.4 Y0.8515585831
N16 G1 X1.5 Y0.9079809995	N17 G1 X1.6 Y0.9635073482
N18 G1 X1.7 Y1.018082832	N19 G1 X1.8 Y1.07165359
N20 G1 X1.9 Y1.124166756	N21 G1 X2 Y1.175570505
N22 G1 X2.1 Y1.225814107	N23 G1 X2.2 Y1.274847979
N24 G1 X2.3 Y1.322623731	N25 G1 X2.4 Y1.369094212
N26 G1 X2.5 Y1.414213562	N27 G1 X2.6 Y1.457937255
N28 G1 X2.7 Y1.500222139	N29 G1 X2.8 Y1.541026486
N30 G1 X2.9 Y1.580310025	N31 G1 X3 Y1.618033989
N32 G1 X3.1 Y1.654161149	N33 G1 X3.2 Y1.688655851
N34 G1 X3.3 Y1.721484054	N35 G1 X3.4 Y1.75261336
N36 G1 X3.5 Y1.782013048	N37 G1 X3.6 Y1.809654105
N38 G1 X3.7 Y1.835509251	N39 G1 X3.8 Y1.859552972

22

N40 G1 X3.9 Y1.881761538	N41 G1 X4 Y1.902113033
N42 G1 X4.1 Y1.920587371	N43 G1 X4.2 Y1.937166322
N44 G1 X4.3 Y1.951833524	N45 G1 X4.4 Y1.964574501
N46 G1 X4.5 Y1.975376681	N47 G1 X4.6 Y1.984229403
N48 G1 X4.7 Y1.991123929	N49 G1 X4.8 Y1.996053457
N50 G1 X4.9 Y1.999013121	N51 G1 X5 Y2
N52 G1 X5.1 Y1.999013121	N53 G1 X5.2 Y1.996053457
N54 G1 X5.3 Y1.991123929	N55 G1 X5.4 Y1.984229403
N56 G1 X5.5 Y1.975376681	N57 G1 X5.6 Y1.964574501
N58 G1 X5.7 Y1.951833524	N59 G1 X5.8 Y1.937166322
N60 G1 X5.9 Y1.920587371	N61 G1 X6 Y1.902113033
N62 G1 X6.1 Y1.881761538	N63 G1 X6.2 Y1.859552972
N64 G1 X6.3 Y1.835509251	N65 G1 X6.4 Y1.809654105
N66 G1 X6.5 Y1.782013048	N67 G1 X6.6 Y1.75261336
N68 G1 X6.7 Y1.721484054	N69 G1 X6.8 Y1.688655851
N70 G1 X6.9 Y1.654161149	N71 G1 X7 Y1.618033989
N72 G1 X7.1 Y1.580310025	N73 G1 X7.2 Y1.541026486
N74 G1 X7.3 Y1.500222139	N75 G1 X7.4 Y1.457937255
N76 G1 X7.5 Y1.414213562	N77 G1 X7.6 Y1.369094212
N78 G1 X7.7 Y1.322623731	N79 G1 X7.8 Y1.274847979
N80 G1 X7.9 Y1.225814107	N81 G1 X8 Y1.175570505
N82 G1 X8.1 Y1.124166756	N83 G1 X8.2 Y1.07165359
N84 G1 X8.3 Y1.018082832	N85 G1 X8.4 Y0.9635073482
N86 G1 X8.5 Y0.9079809995	N87 G1 X8.6 Y0.8515585831
N88 G1 X8.7 Y0.7942957813	N89 G1 X8.8 Y0.7362491054
N90 G1 X8.9 Y0.6774758405	N91 G1 X9 Y0.6180339887
N92 G1 X9.1 Y0.5579822121	N93 G1 X9.2 Y0.4973797743
N94 G1 X9.3 Y0.4362864828	N95 G1 X9.4 Y0.3747626292
N96 G1 X9.5 Y0.3128689301	N97 G1 X9.6 Y0.2506664671
N98 G1 X9.7 Y0.1882166266	N99 G1 X9.8 Y0.1255810391
N100 G1 X9.9 Y0.06282151816	N101 G1 X10 Y0

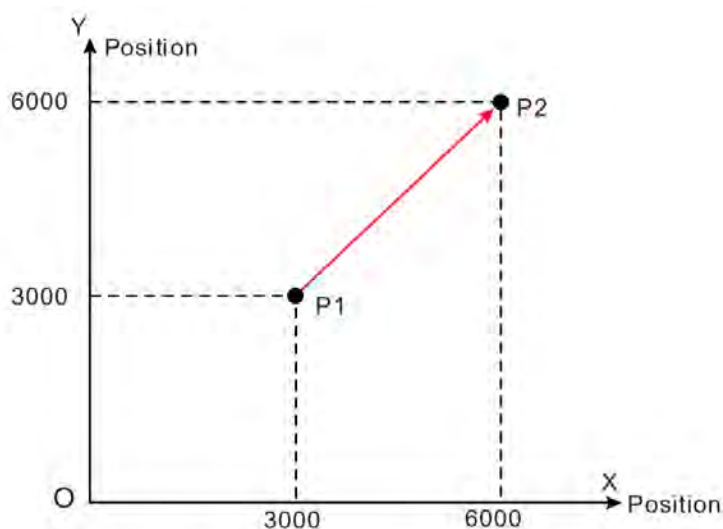
G 码执行后，整个过程的 Y/X 曲线如下：



22

22.1.4.10 G90 (绝对模式)

- 功能：
G90 执行后，其后 G 码中各轴的终点位置值均以 0 单元为基准，中途可以用 G91 指令切换为相对模式。程序默认为绝对模式。
- 格式：
N_G90
- 参数说明：
N_：G 码在程序中所在行的编号。
- 范例：
X、Y 轴的初始位置都为 3000 单元，轴参数都为默认值。
将要执行的 G 码如下：
N00 G90
N01 G0 X6000 Y6000
G 码执行后，整个过程的 Y/X 曲线如下：



22.1.4.11 G91 (相对模式)

- 功能：
G91 执行后，其后 G 码中各轴的终点位置均以从当前位置开始的增量值计算，中途可以用 G90 指令切换为相对模式。程序默认为绝对模式。

- 格式：

N_G91

- 参数说明：

N_：G 码在程序中所在行的编号。

- 范例：

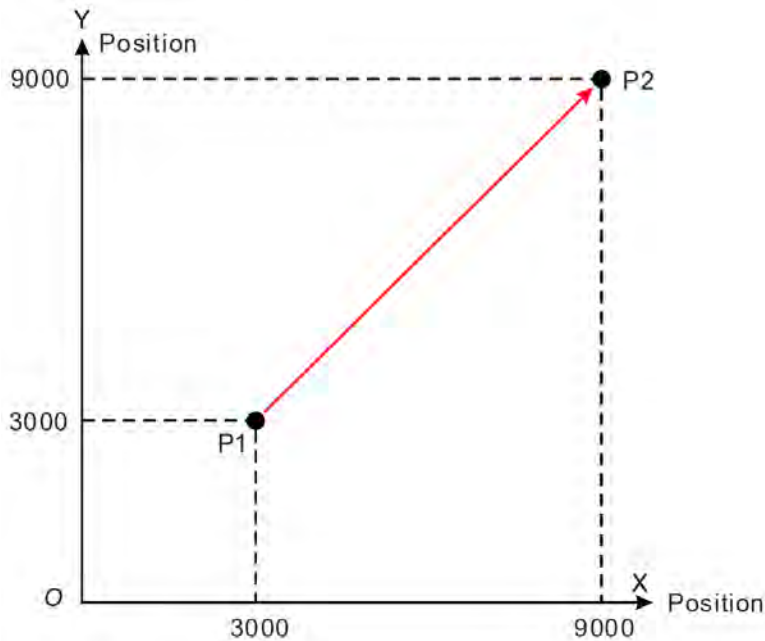
X、Y 轴的初始位置都为 3000 单元，轴参数都为默认值。

将要执行的 G 码如下：

```
N00 G91
```

```
N01 G0 X6000 Y6000
```

G 码执行后，整个过程的 Y/X 曲线如下：



22

22.1.4.12 M 码

- 功能：
与一般程序进行交互使用。
- 格式：
N_M_D_
- 指令说明：
 1. D 分量可以被省略，执行 M 码时就不会传出参数。
 2. M 码有两种使用方式：一种是和 G 码不写在同一行，一种是和 G 码写在同一行。
 3. 当 M 码和 G 码不在同一行，例如：N0 M10 D10.02；N1 G1 X100 Y100。当执行到 N0 这一行时，G 码执行会停下，与此同时，使用 DMC_ReadMFunction 指令读取 M 码状态，输出位 MFunction 置为 TRUE，并且 MFunctionValue 的值为 10.02。当使用 DMC_ResetMFunction 指令复位 M 码后，G 码才会继续向下执行 N1。
 4. 当 M 码和 G 码在同一行，并且只能放在 G 码之后。例如：N0 G1 X100 M10 D10.02。当执行到这一行时，会执行 G1，同时使用 DMC_ReadMFunction 指令读取 M 码状态，输出位 MFunction 置为 TRUE，并且 MFunctionValue 的值为 10.02。G1 执行完成后则会继续向下执行，不需要使用指令复位 M 码。

22

22

● 范例：

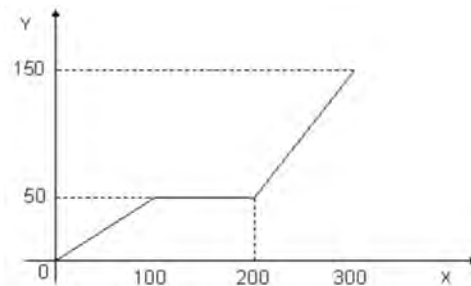
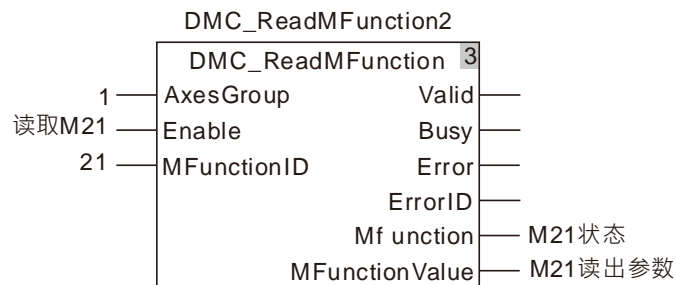
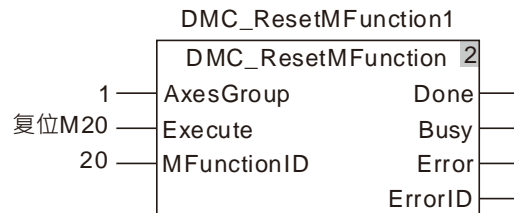
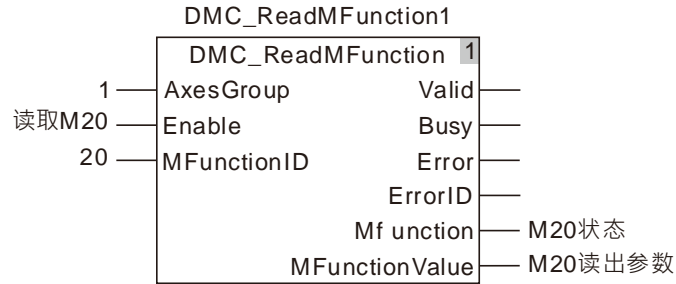
N0 G1 X100 Y50

N1 M20

N2 G1 X200 Y50 M21 D3.14

N3 G1 X300 Y150

G 码执行后，整个过程的 Y/X 曲线如下：

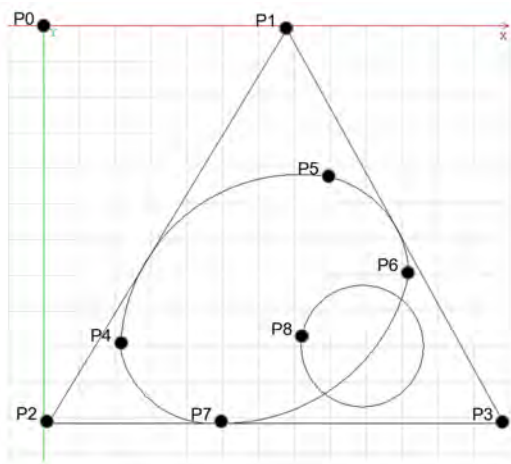


当两个变量“读取 M20”和“读取 M21”为 TRUE 时，开始执行 G 码，终端机构运行到(100 · 50)的位置停下，此时变量“M20 状态”为 TRUE，等待其它动作执行完成后，再将变量“复位 M20”为 TRUE，复位 M 代码。此时 G 码继续运行，终端机构开始运行到(200 · 50)，同时变量“M21 状态”为 TRUE，变量“M21 传出参数”

22.1.5 G-Code 范例-三轴同动绘出 Delta LOGO

使用 G-Code 与绝对寻址来编写绘制轨迹，绘出 Delta Logo。

【欲绘制图形】



【G-Code 程序代码】

```

0001 N0 G90                                     (P0)
0002 N1 G0 Z5000
0003 N2 G0 X32500 Y-500                         (P1)
0004 N3 G0 Z0
0005 N4 G1 X600 Y-53400 F10000                 (P2)
0006 N5 G1 X61500 Y-53400 F10000              (P3)
0007 N6 G1 X32500 Y-500 f10000                (P4)
0008 N7 G0 Z5000
0009 N8 G0 X23100 Y-53400                       (P5)
0010 N9 G0 Z0
0011 N10 G2 X10300 Y-43600 R12600              (P6)
0012 N11 G2 X38440 Y-20500 R23900              (P7)
0013 N12 G2 X48880 Y-33300 R13900              (P8)
0014 N13 G2 X23100 Y-53400 R25600              (P5)
0015 N14 G0 Z5000
0016 N15 G0 X34500 Y-43000                       (P9)
0017 N16 G0 z0
0018 N17 G2 X34500 Y-43000 I8150 J0 F5000      (P9)
0019 N18 G0 z5000
0020 N19 G0 x0 y0                               (P0)
0021

```

【路径说明】

程序运行步骤：

步骤一：Z 轴提笔上移后，从原点 P0 移动到达 P1。

步骤二：P1 处 Z 轴下笔，从 P1 移动到达 P2，P2 移动到达 P3，P3 移动到达 P1，第三轴提笔，完成三角形。

步骤三：从 P1 移动到达 P4，P4 处 Z 轴下笔，从 P4 移动到达 P5，P5 移动到达 P6，P6 移动到达 P7，P7 移动到达 P4，Z 轴提笔，完成椭圆形。

步骤四：从 P4 移动到达 P8，P8 处 Z 轴下笔，从 P8 直接画一个圆至 P8，Z 轴提笔，完成圆形。

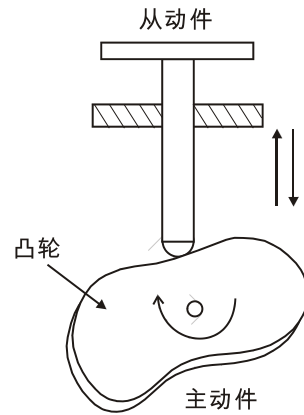
步骤五：Z 轴移动至原点 P0 后结束，DELTA LOGO 完成。

22.2 E-CAM 编辑器

22.2.1 关于 E-CAM (电子凸轮)

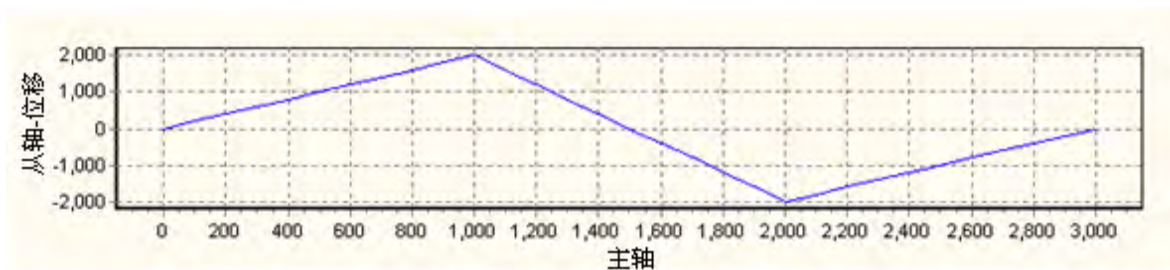
凸轮是一种不规则形状的输入件，可以经由直接接触传递运动到从动件，使从动件依设定的规律运动。机械凸轮为由凸轮、从动件以及机架等三个部份组合而成。如图，当凸轮旋转时，从动件会依照凸轮的外型，作上下的运动。当进行运动控制时，轴与轴之间常需要进行互有关连性的运动，此种运动即需要凸轮来达成。

电子凸轮 (E-CAM) 以建立电子凸轮表 (CAM Chart) 的方式，定义主动件与从动件的关系后，由控制器模拟凸轮的运动，使从轴依此关系对应主轴运动。电子凸轮的优点是只要修改电子凸轮表，而不需要变更机构，即可完成凸轮的修改，且无机构磨损的问题。



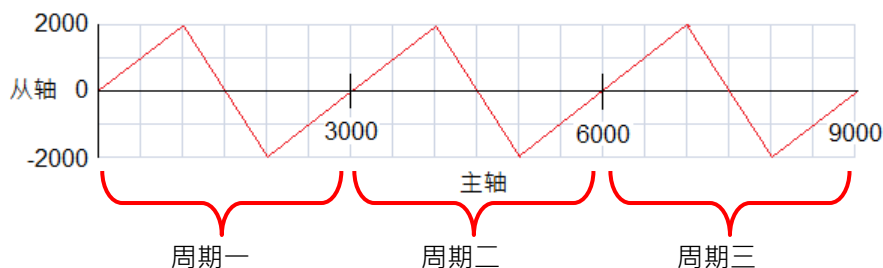
22.2.2 电子凸轮意义说明

一个完成建立的 CAM 图表如下图，CAM 图表的横轴代表主轴，纵轴代表从轴，CAM 图表上的数值皆为单位数 (例如脉冲数)，横轴的 0~3000 个单位为一个电子凸轮周期，纵轴则为相对应的从轴输出单位数。



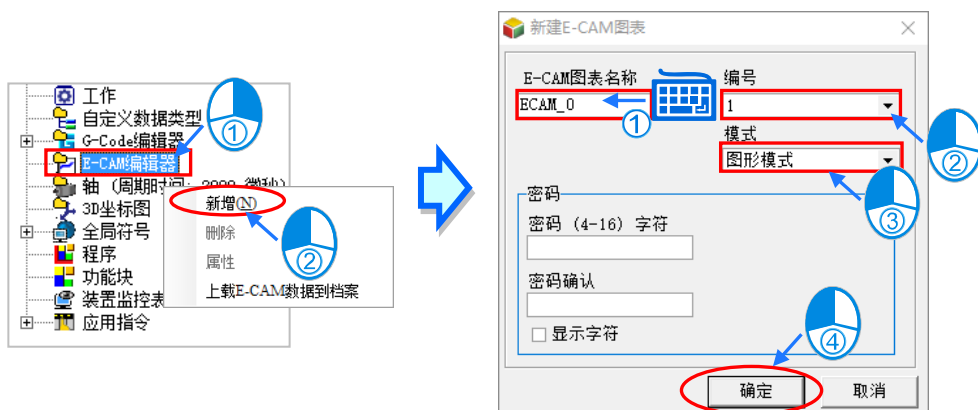
当这个 CAM 图表被作为周期性的使用时，主轴与从轴的输出关系如下图。可以知道电子凸轮将会被运动主机不断重复的使用，而当主轴完成一个电子凸轮周期时 (主轴送出 3000 个单位)，主轴的单位数会继续累加，而非回到初始值；从轴在完成一个电子凸轮周期后，则会依据电子凸轮设定的行程初始值再度重复行程的变化。

若以连接伺服马达的观点而言，主轴伺服可视为不断往同方向旋转的传动轴，从轴伺服则可视为依照电子凸轮关系不断做周期性往复运动的加工轴，这个特性可用在追剪与飞剪等应用。



22.2.3 使用 E-CAM 编辑器

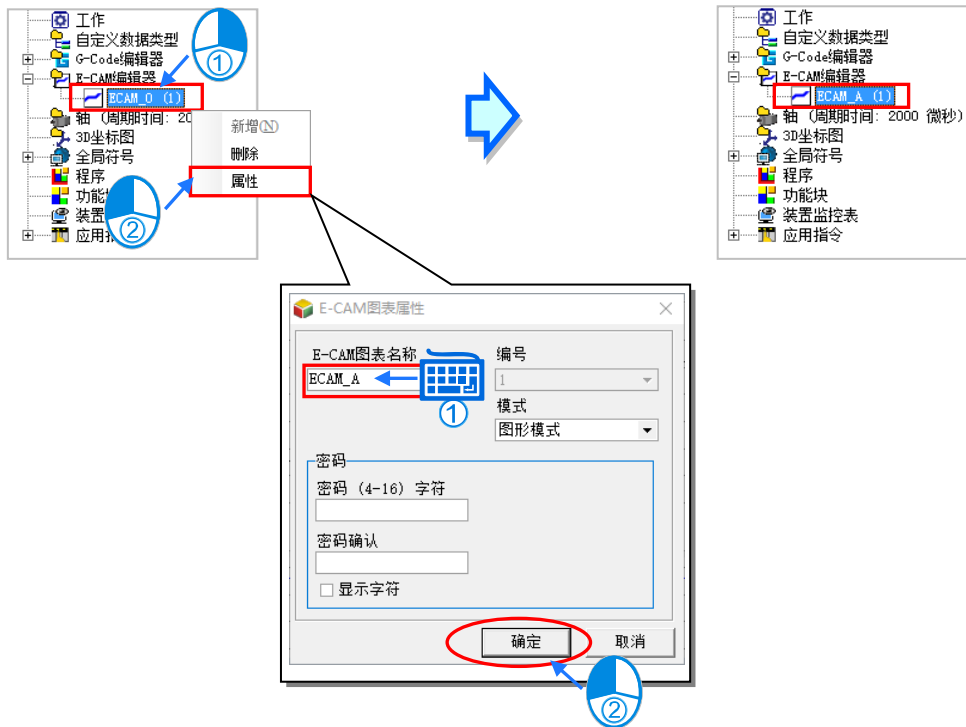
在项目管理区的 E-CAM 编辑器项目单击 **右键** > **新增**，出现新增窗口如下图，在 E-CAM 图表名称的字段输入名称，再在编号单击鼠标左键，在下拉选项选择编号，接着选择所需的模式。也可以设定密码来保护 E-CAM 设计数据，设定好密码后，若为「**图形模式**」，则下次开启编辑窗口时将会询问密码；若为「**档案模式**」，则在导入导出时会询问密码。



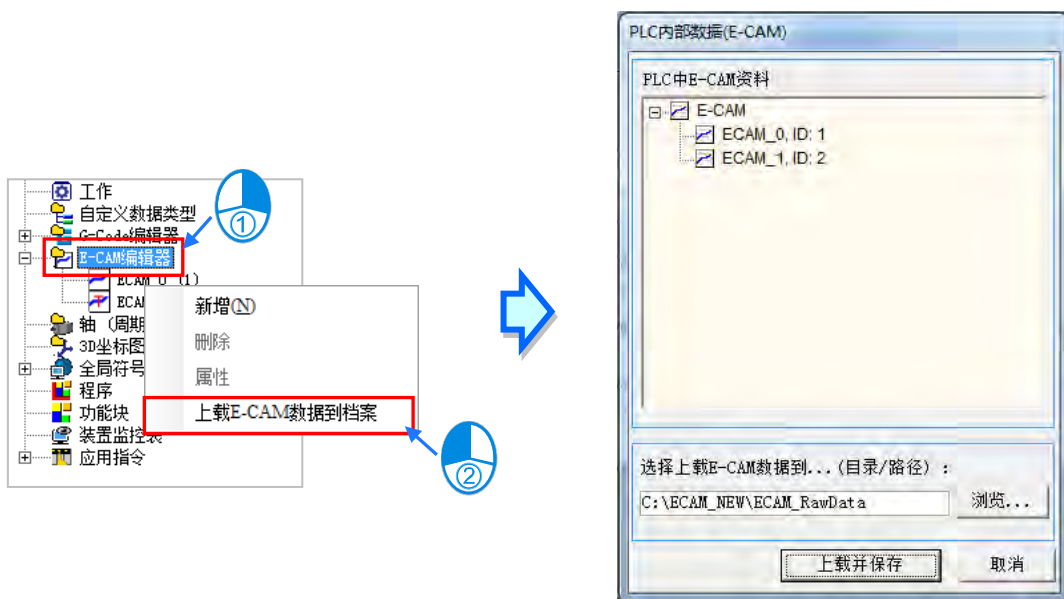
名称	功能说明	AS	AHxxEMC	AS5xx DVPxxMC
编号	E-CAM 表的编号	最大 16	最大 32	最大 32
ASDA 档案模式	ASDA 档案模式的 E-CAM 表	V		
档案模式	档案模式的 E-CAM 表		V	
图形模式	图形模式的 E-CAM 表		V	V
周期标志	当拖拉第一个与最后一个控制点，两点的从轴数据会实时同步			V
密码	E-CAM 表的密码		V	V
上载 E-CAM 到档案	从 PLC 上载主轴、从轴位置		V	

在已建立的 E-CAM 项目上单击鼠标右键，可在快捷选项单击「删除」，删除此项目；或是单击「属性」，重新设定项目名称或密码，但不可变更编号。

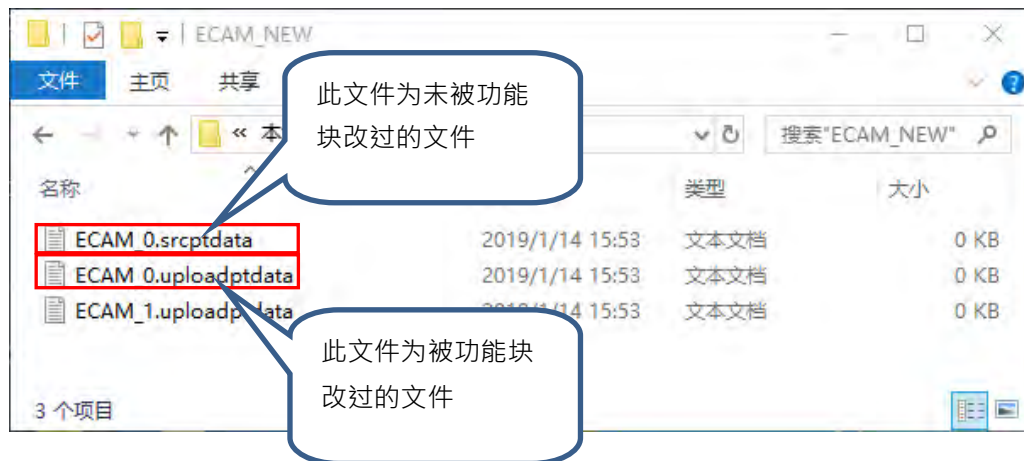
22



使用台达函数库内 AH-Motion 的功能块，例如：DFB_CamWire，在线修改过 E-CAM 图表的点位，接着可以利用「上载 E-CAM 数据到档案」的功能，将 E-CAM 表的主轴位置及从轴位置另存成档案。使用此功能，需先与 AHxxEMC 主机联机，再在项目管理区的 E-CAM 编辑器项目单击 右键 > 上载 E-CAM 数据到档案，此时，会出现 PLC 内部数据 (E-CAM) 窗口。

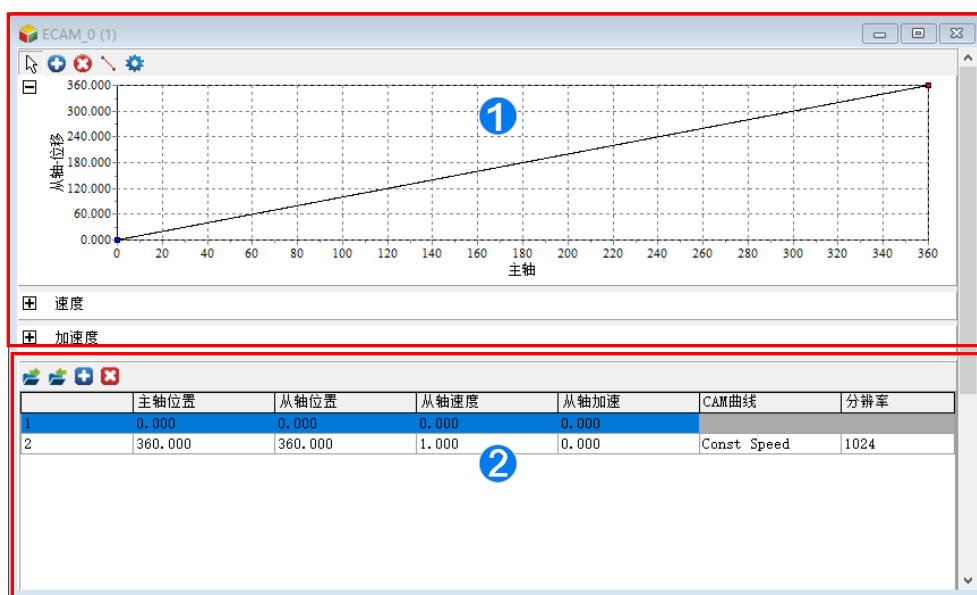


在选择完存盘路径后，单击「**上传并保存**」，这时保存路径目录底下会出现未被功能块修改过的档案 xxxx_srcptdata.txt 及被功能块修改过的档案 xxxx_uploadptdata.txt 两个档案，xxxx 为 E-CAM 的项目名称，如下图。



22

若选择「**图形模式**」，完成新建 E-CAM 图表后将会开启 E-CAM 编辑窗口，功能说明如下：






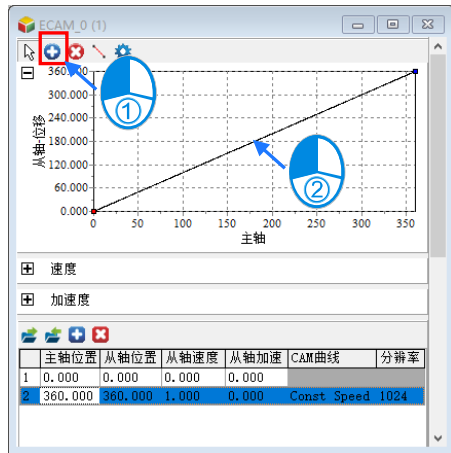
- ① 关系图区：依据数据设定区设计数据，实时显示对应的 E-CAM 位置、速度、加速度关系图。可以使用鼠标左键拖曳移动图中的关键点改变设计数据。也可使用功能按钮新增关键点或线段。
- ② 规划表区：使用功能按钮新增或删除表格，在表格中输入 E-CAM 设计数据。数据将与关系图同步。

E-CAM 编辑器的功能按钮说明如下

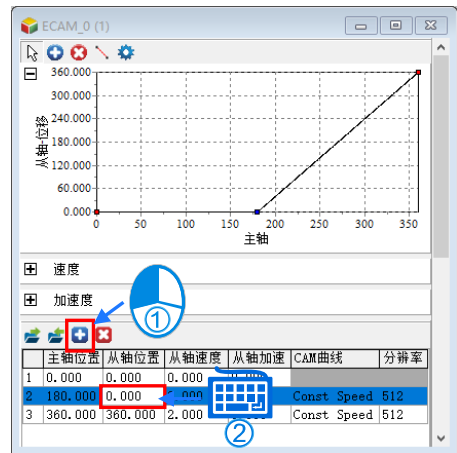
图标	名称	功能说明	AS	AHxxEMC	DVPxxMC
	选取工具	切换为选取工具	V	V	V
	新增关键点	在关系图中新增一个关键点	V	V	V
	删除关键点	在关系图中删除一个关键点	V	V	V
	新增线段	在关系图中新增包含两关键点的线段		V	V
	设定	开启设定窗口	V	V	V
	汇出	导出对应格式的 E-CAM	.csv	.dst	.dst
	汇入	导入对应格式的 E-CAM	.csv	.dst	.dst
	上载从轴数据	从档案缓存器/伺服上载数据回来	V		
	下载从轴数据	下载数据至档案缓存器/伺服	V		
	新增一行	在规划表区新增一笔关键点数据	V	V	V
	删除一行	在规划表区删除一笔关键点数据	V	V	V
	新增挺杆点	在曲线上新增挺杆点的图标 			V
	显示 CAM 表	显示 CAM 表			V
	显示挺杆点表	显示挺杆点表			V

22

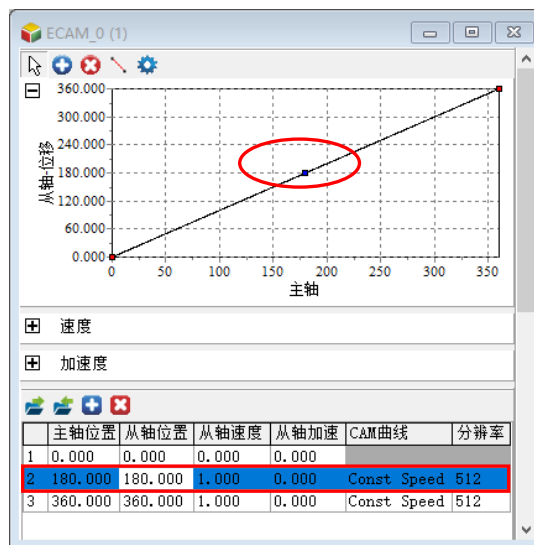
要新增点，先单击「新增关键点」 的按钮后，在关系图中要新增的位置，单击鼠标左键，便会建立一个关键点，此时下方的规划表区也会自动增加此关键点的数据，或是在规划表区单击，在于此关键点数据填入从轴位置 180 如下图。若要连续新增多个关键点，在单击 按钮后可以在关系图中多次单击鼠标左键。






OR

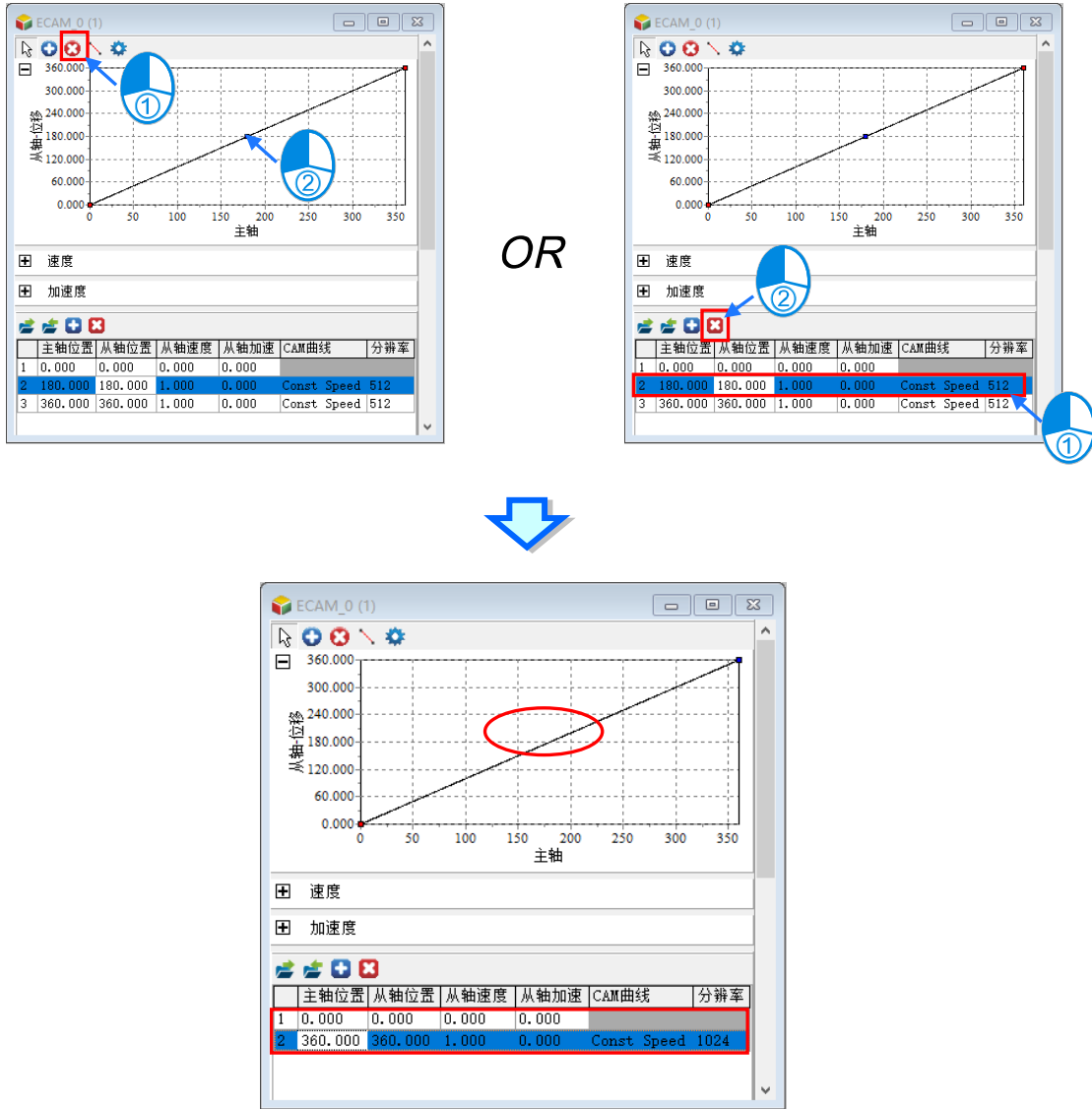


22



要删除点，在单击「删除关键点」 按钮后，在关系图中单击要删除的关键点，此时规划表区也会自动删除此关键点的数据，或在规划表区选取要删除的关键点数据，接着单击 即可删除，如下图。若要连续点删除多个关键点，单击 按钮后直接连续单击要删除的点即可。

22




在电子凸轮运用过程中，可能需要在凸轮运行到某些特定位置时触发相应的动作，这时就需要在凸轮的指定位置设定一个状态开关，当凸轮运行到该位置时，状态开关的状态发生变化，在程序中可通过状态开关的状态控制相应的动作，这个状态开关称之为凸轮挺杆。

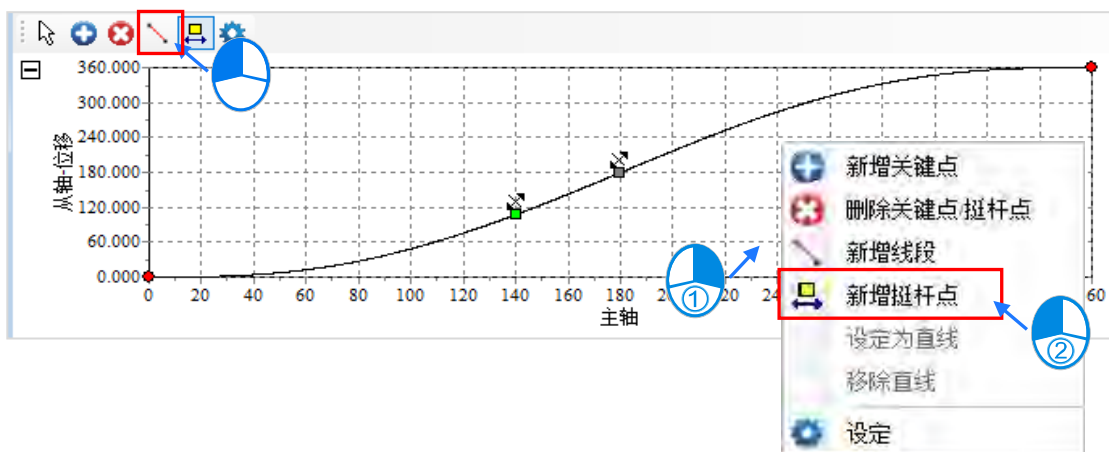
凸轮挺杆可以设定在凸轮的任何位置，且一个凸轮最多可设定 128 个凸轮挺杆。凸轮正向或反向经过时，挺杆的状态可分别进行配置（关闭、置位、复位、取反）。在控制器程序中可通过凸轮挺杆相应指令获取挺杆的状态。凸轮挺杆状态说明如下：

关闭	凸轮经过该挺杆对应的位置时，该挺杆状态不变化。
置位	凸轮经过该挺杆对应的位置时，该挺杆状态被设定为 TRUE。
复位	凸轮经过该挺杆对应的位置时，该挺杆状态被设定为 FALSE。

取反	凸轮经过该挺杆对应的位置时，该挺杆状态在 TRUE 和 FALSE 之间切换一次（如果为 TRUE 则变为 FALSE，如果为 FALSE 则变为 TRUE）。
----	--

注意：凸轮挺杆的状态在下一个凸轮周期开始时会被复位。

新增凸轮挺杆可以通过工具栏上的新增挺杆点  图标，或是在关系图中单击右键，接着在选项上选取新增挺杆点，如下图。



在规划表中每行关键点数据提供用户设定一段凸轮曲线，再由数行关键点数据组成整个凸轮曲，所有已建立的关键点数据即构成一个电子凸轮周期，用户在每笔关键点数据输入各的主轴与从轴关系，说明如下。


名称	功能说明	AS	AHxxEMC	DVPxxMC
关键点	记录 E-CAM 运动的坐标	最大 721	最大 40	最大 40
CAM 曲线 Const Speed	等速度		V	
CAM 曲线 Const Acc.	等加速度		V	
CAM 曲线 Single Hypot.	单精度三角斜边		V	
CAM 曲线 Cycloid	摆线		V	
CAM 曲线 Polynomial	多项式	V	V	V
CAM 曲线 B-Spline	B-样条		V	
分辨率	目前区段所使用的资料点數。		V	

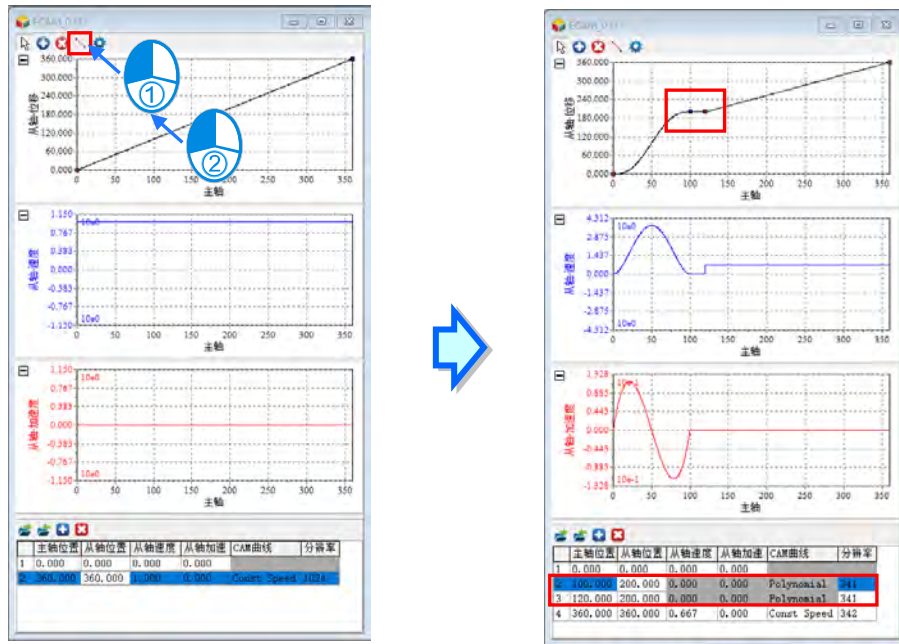
单击「设定」按钮，开启其它设定窗口，说明如下。



- ❶ 主轴位置：设定关系图中主轴显示范围。
- ❷ 从轴位置：设定关系图中从轴显示范围。
- ❸ 分辨率：设定此 E-CAM 表，曲线区间的分辨率总和。

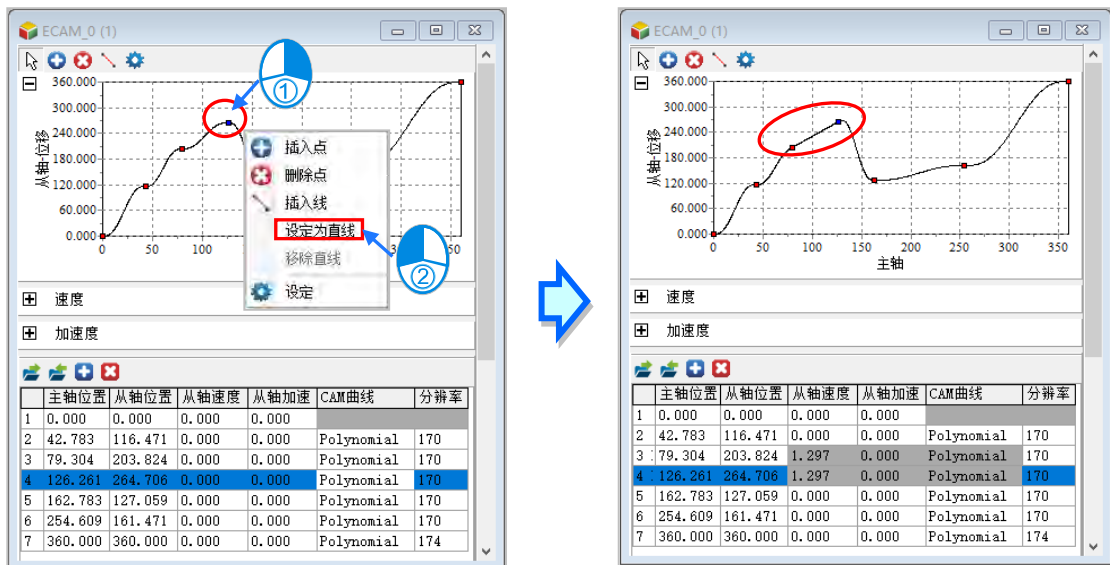
名称	功能说明	AS	AHxxEMC	DVPxxMC
主轴位置	设定关系图中主轴显示范围。	固定等间距 无法设定	V	最小值 固定为 0
从轴位置	设定关系图中从轴显示范围。	V	V	V
分辨率	设定此 E-CAM 表，曲线区间的分辨率总和。		V	

要新增直线属性的线段，在单击「新增线段」的按钮后，在关系图中要新增的位置单击鼠标左键，此时下方的规划表区也会自动增加此线段的两个关键点数据。当插入线段后，插入线段的 CAM 曲线函数会被固定在「Polynomial」的模式，左方线段编号右侧则会显示「Line Start」及「Line End」。



22

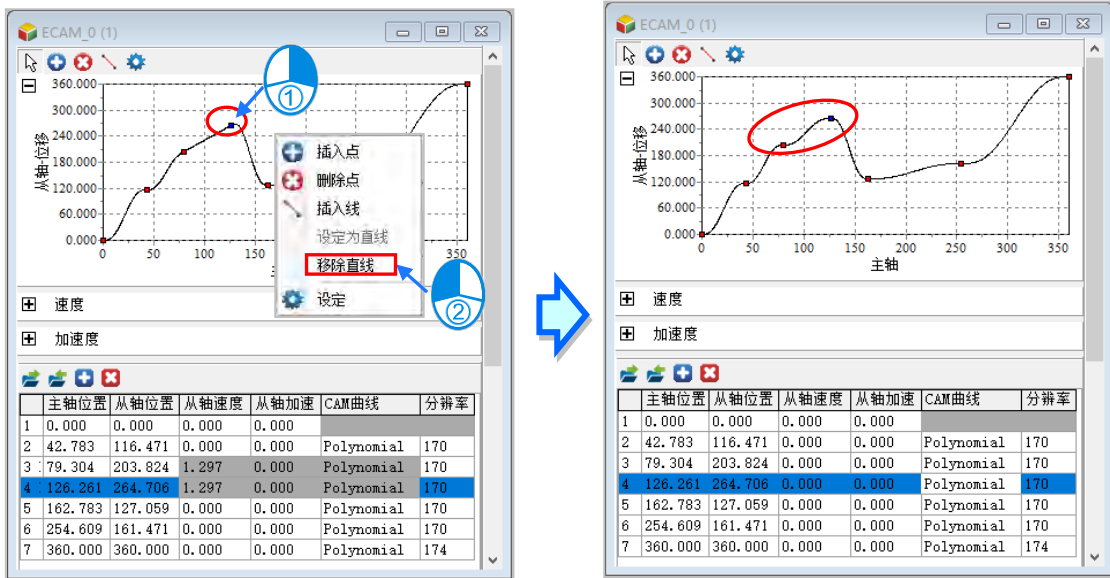
若要将 Polynomial 区间线段设定为直线，可先对关系图中的关键单击鼠标右键，接着用鼠标左键单击「设定为直线」，Polynomial 区间线段将变为直线。



使用「设定为直线」功能，需注意下列事项：

- 前一个与目前区间必须是 Polynomial 区间才可以使用此功能。
- Line Start 与 Line End 的关键点不能与其它直线区间共享直线属性。

若要移除已设定的直线属性，需用鼠标右键单击关系图中 Polynomial 区间线段的 Line Start 或 Line End 关键点，接着用鼠标左键单击「移除直线」。

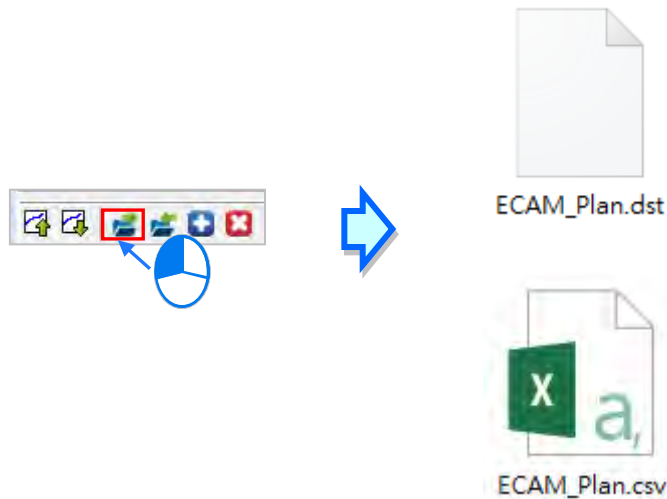


22

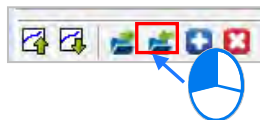
使用「移除直线」功能，需注意下列事项：

- 只有属性为 Line Start 或 Line End 关键点才可以使用此功能。

导出规划表的数据，单击规划表上方功能列的导出 图标，依照项目机型导出的文件格式会有所不同，且不同的机种的文件格式不能通用。

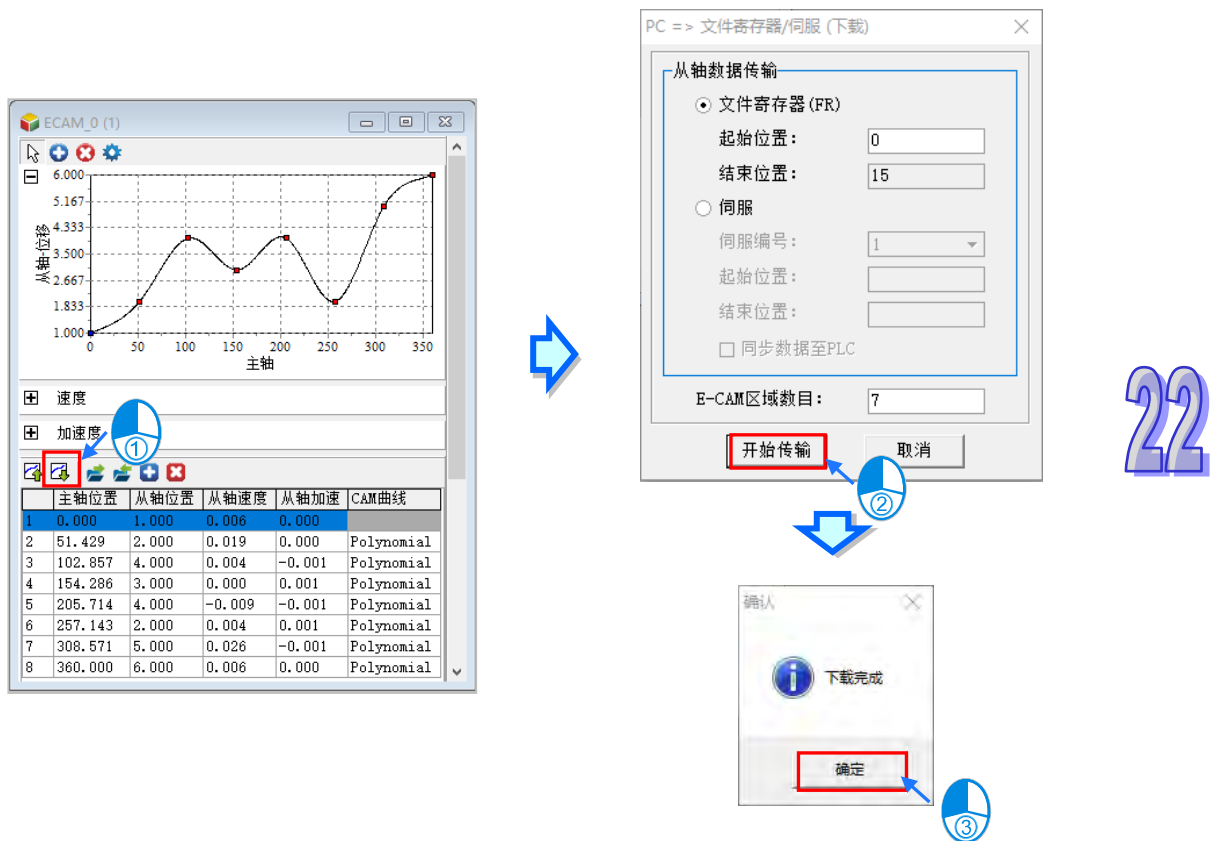


要导入规划表的档案时，直接单击导入 图标，接着选取要导入的档案就可以了。

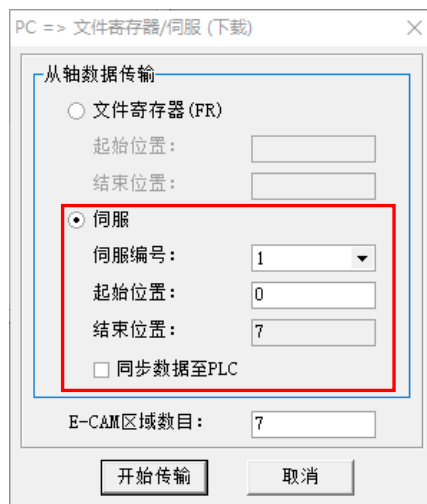


下载从轴数据必须先将 ISPSOft 与 AS 机器建立联机，下面使用一个范例来操作，首先开启一张 E-CAM 表并增加至 8 个关键点，接着单击「下载从轴数据[PC=>档案缓存器/伺服]」，之后会弹出一个窗口，选择要下载到档案缓存器 (FR) 或是伺服，E-CAM 区域数目为要存取的曲线区域，这里我们下载 8 个关键点所以等于 7 个曲线区域，当变更起始位置或 E-CAM 区域数目时，系统会自动计算结束位置，

单击「开始传输」，传输完毕后会弹出下载完成的窗口。

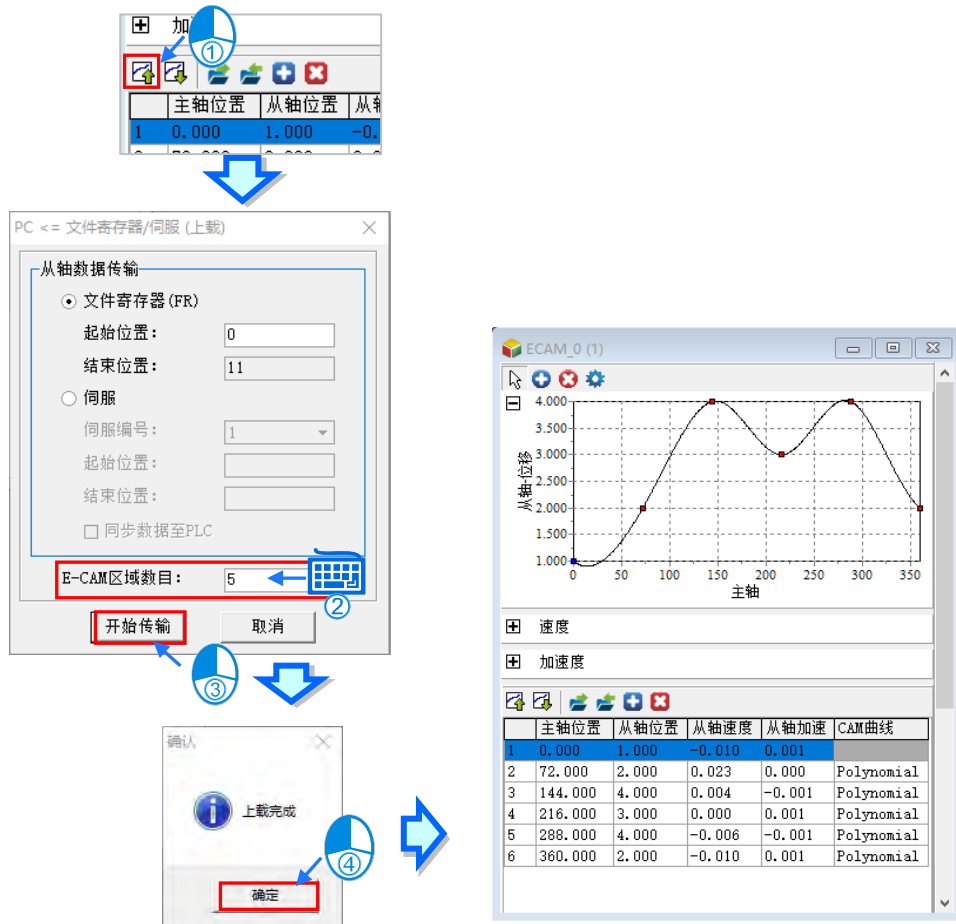


若要直接下载到伺服，则可以选择伺服，首先需要选取相对应的伺服编号，接着可以设定起始位置，若勾选「同步数据至 PLC」选项，会把当前的 E-CAM 数据同步至 PLC 中，但项目中的 E-CAM 数量与编号需与 PLC 中的一致，才能够同步数据。



接着单击「**上传从轴数据[PC<=档案缓存器/伺服]**」，之后会弹出一个窗口，此时我们只要上传 6 个关键点，所以这里我们将 E-CAM 区域数目修改为 5，并单击「**开始传输**」上传完成之后，E-CAM 表会被重新绘制为 6 个关键点，如下图。

22



若模式为「**档案模式**」，则需先将此 E-CAM 表做导出的动作，才可做编辑。首先，在要导出的 E-CAM 项目，单击鼠标右键，再选择「**汇出**」，选择路径及文件名后，单击「**保存**」，即可完成导出动作。



在导出的 CSV 档，输入 E-CAM 动作的主轴位置 (X) 及从轴位置 (Y)，输入格式为 0、正整数或正浮点数；输入位置点数，最多 2048 组 XY 位置点，最少 2 组 XY 位置点，输入完毕后，做存盘的动作，再回到 ISPSOft，做导入的动作。

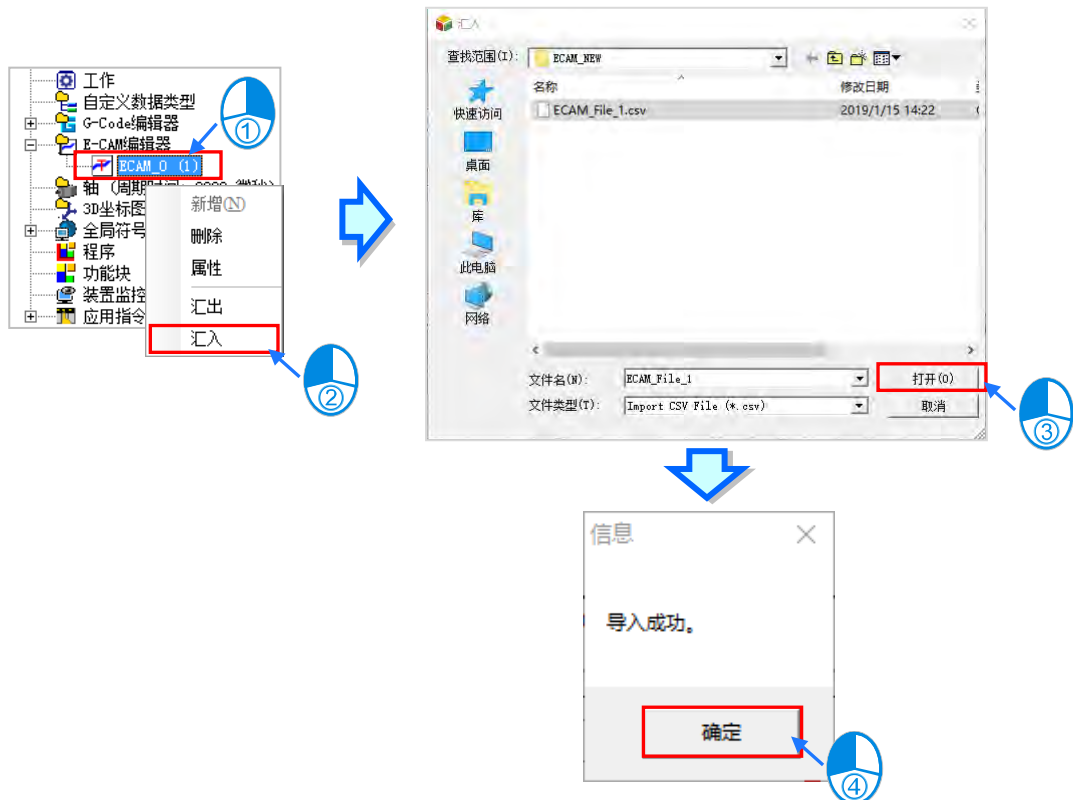
	A	B	C	D
1	:Delta	:E-Cam	Version	V1.0
2	X	Y		
3	0	0		
4	0.280561	0.804357		
5	0.561122	1.608714		
6	0.841682	2.413071		
7	1.122243	3.217427		
8	1.402804	4.021784		
9	1.683365	4.826141		
10	1.963925	5.630497		
11	2.244486	6.434855		
12	2.525047	7.239211		
13	2.805608	8.043568		

主轴位置

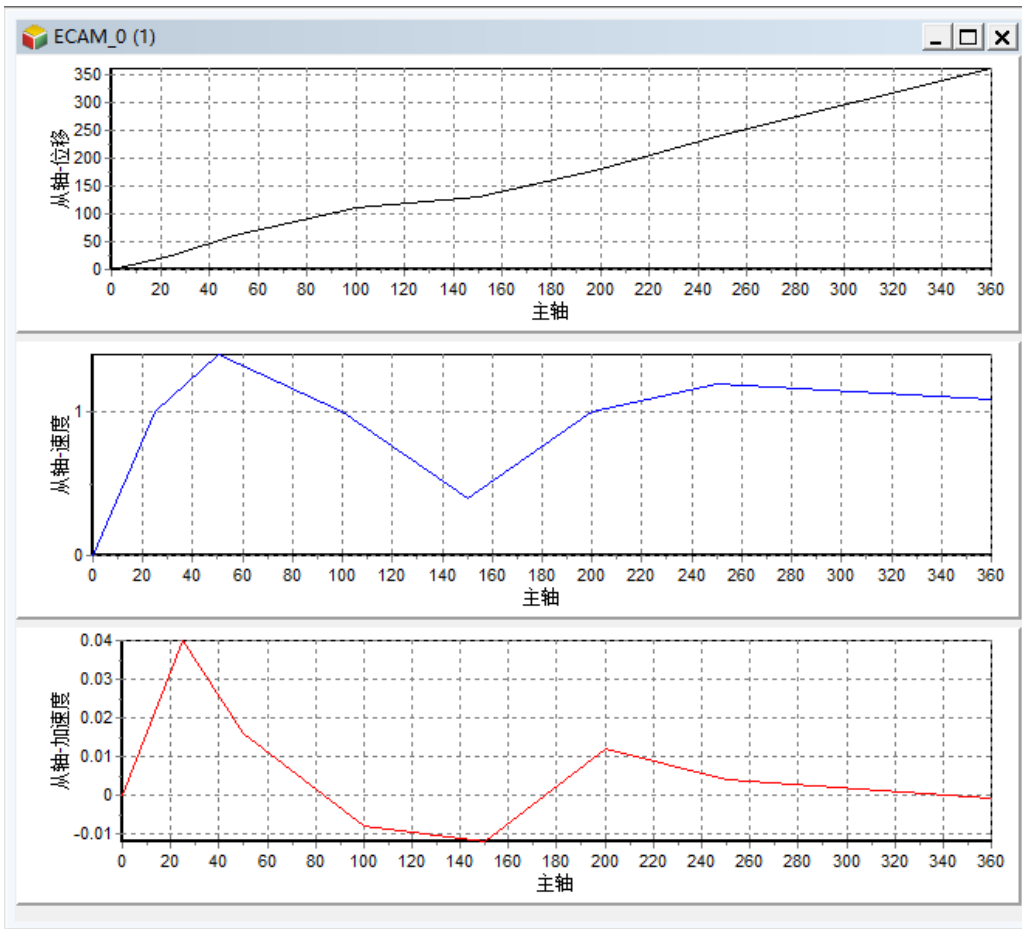
从轴位置

22

回到 ISPSOft，在要导入的 E-CAM 表项目，单击鼠标右键，选择「汇入」，并选取要开启的档案，再单击「开启旧档」，此时会跳出「汇入成功」的信息窗口，单击「确定」，即完成导入动作。

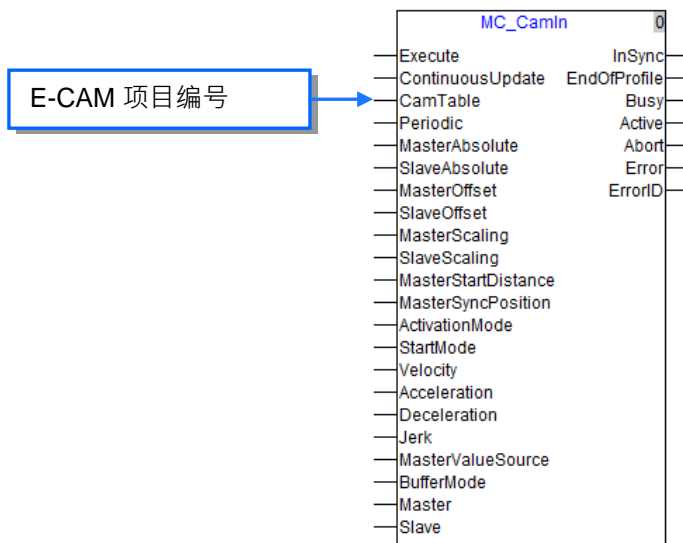


导入成功后，可以看见相对应的 E-CAM 图表，此图表仅能检视导入档案的数据图标。

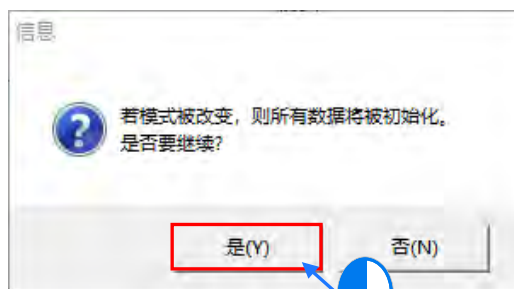


22

完成 E-CAM 编辑器的编写之后，便可在 POU 中搭配运动控制指令或功能块使用，例如以 MC_Camin 指令依 E-CAM 编辑器项目编号启动指定的 E-CAM 项目。关于 MC_Camin 的使用请参阅各机种相关指令手册。



切换 E-CAM 图表的模式时，在单击「确定」后，将出现「信息」窗口，询问「若模式被改变，则所有数据将被初始化，是否继续？」，此时选择「是」，ECAM 表数据将全被初始化。



MEMO

22

第23章 向导工具

目录

23.1 定位规划表	23-3
23.1.1 关于定位规划表	23-3
23.1.2 设定定位规划表	23-3
23.1.3 定位规划表的仿真.....	23-7
23.1.4 下载与上载定位规划表	23-9
23.1.5 轴控模式-单轴点对点模式	23-10
23.1.6 轴控模式-单轴多段输出模式	23-10
23.1.7 轴控模式-双轴直线插补模式	23-11
23.1.8 轴控模式-双轴圆弧插补模式	23-12
23.2 数据示波器	23-14
23.2.1 关于数据示波器	23-14
23.2.2 开启数据示波器	23-14
23.2.3 设定取样参数	23-15
23.2.4 取样模式.....	23-18
23.2.5 量测	23-20
23.2.6 DIA 示波器.....	23-24
23.3 数据记录器	23-34
23.3.1 关于数据记录器	23-34
23.3.2 开启数据记录器	23-34
23.3.3 设定取样参数	23-35
23.3.4 监看与录制模式	23-37
23.3.5 量测	23-40
23.4 高速计数器向导.....	23-44
23.4.1 关于高速计数器向导.....	23-44
23.4.2 使用高速计数器向导.....	23-44
23.5 NTC 向导.....	23-46
23.5.1 关于 NTC 向导.....	23-46
23.5.2 使用 NTC 向导.....	23-46
23.6 运动参数模拟	23-48

23.6.1 关于运动参数模拟.....23-48

23

23.1 定位规划表

23.1.1 关于定位规划表

ISPSoft 提供用户一种简易的表格设定，来完成单轴或两轴的运动控制，用户无须了解复杂的运动控制规范，只要完成表格的建立并下载，再搭配应用指令启动，主机便会依据设定的编号依序执行各轴输出。



23.1.2 设定定位规划表



单击工具栏的 向导 (!) > 定位规划表 启动定位规划表设定窗口。定位规划表分成三个区域说明如下图。



- ❶ 图示工具栏：提供各项操作定位规划表的按钮，功能说明如下表。
- ❷ 参数设定区：轴设定用于设定运动轴的控制方式。上载密码保护用于设定密码。定位表格用于设定轴控模式与运动路线。
- ❸ 目标仿真区：使用**相对模式（符号/地址）**或**绝对模式（符号/地址）**，可对目标位置做仿真。

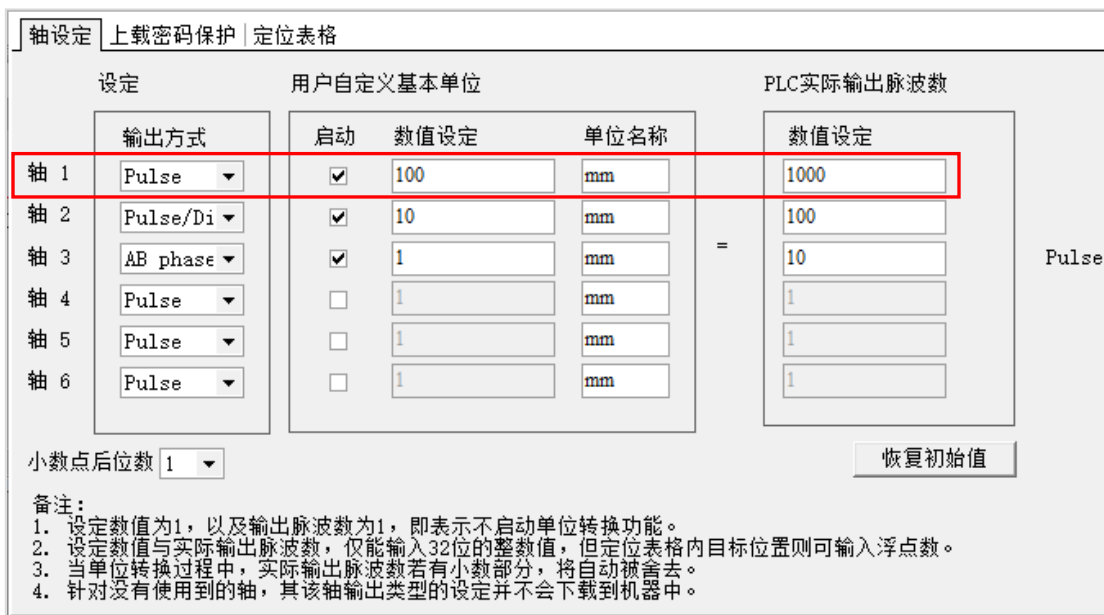
图示工具栏说明：

图示	名称	功能说明
	导入	将.pnt 导入到目前定位规划表。
	导出	将定位规划表设定导出成.pnt。

图示	名称	功能说明
	上载定位规划表	从主机读回定位规划表设定。
	下载定位规划表	将定位规划表设定写入主机。

在轴设定分页，可以设定每个轴的控制输出方式，显示的轴数会随着机种支持而数量有所不同，如果要执行双轴差补运动，则「输出方式」不可选择 Pulse，需选择 Pulse/Dir 或 AB phase。在轴输出方式后面，勾选「启动」后可以对此轴输入自定义的「数值设定」与「单位名称」，及输入 PLC 实际输出脉冲数区中的「数值设定」，如下图，则代表轴 1 在此定位规划表中的 100 个 mm 将输出 1000 个脉冲。按下「恢复初始值」按钮恢复单位设定页面的初始设定。

23



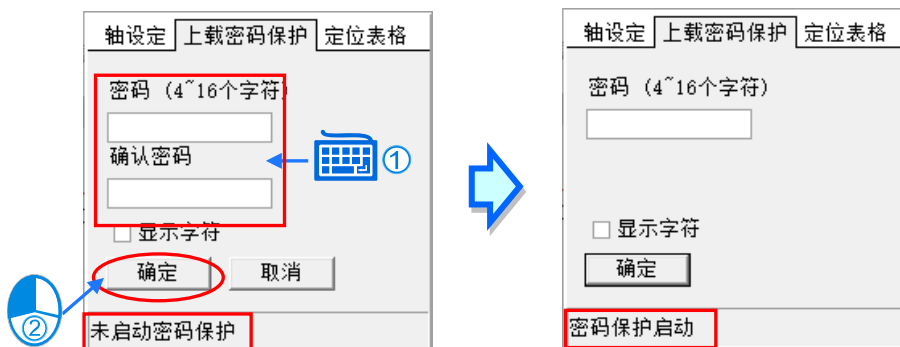
轴	输出方式	启动	用户自定义基本单位		PLC实际输出脉波数
			数值设定	单位名称	
轴 1	Pulse	<input checked="" type="checkbox"/>	100	mm	1000
轴 2	Pulse/Di	<input checked="" type="checkbox"/>	10	mm	100
轴 3	AB phase	<input checked="" type="checkbox"/>	1	mm	10
轴 4	Pulse	<input type="checkbox"/>	1	mm	1
轴 5	Pulse	<input type="checkbox"/>	1	mm	1
轴 6	Pulse	<input type="checkbox"/>	1	mm	1

小数点后位数 1

备注：
 1. 设定数值为1，以及输出脉波数为1，即表示不启动单位转换功能。
 2. 设定数值与实际输出脉波数，仅能输入32位的整数，但定位表格内目标位置则可输入浮点数。
 3. 当单位转换过程中，实际输出脉波数若有小数部分，将自动舍弃。
 4. 针对没有使用到的轴，其该轴输出类型的设定并不会下载到机器中。

在上载密码保护分页，可以设定密码，用于保护下载到主机的定位规划表在上载时必须输入密码，设定方式于密码与确认密码字段输入一致的密码后，按下「确定」成功加密的话，下方会提示密码保护启动。如果要解除密码设定时，只要再次于密码字段输入密码，按下「确定」成功解锁的话，下方会提示未启动密码保护。

⚠ 请注意「上载密码保护」仅提供上载定位规划表时需要密码，若将别的规划表下载到「上载在密码保护」的主机，定位规划表还是会被覆盖。



在定位表格分页的编辑区供新增、修改或删除用，新增完的数据会出现在列表区。

The screenshot shows the '定位表格' (Positioning Table) interface. At the top, there are tabs for '轴设定', '上载密码保护', and '定位表格'. Below the tabs are buttons for '新增', '修改', and '删除'. The main area contains a table with columns: No., 轴控模式, 轴编号, 地址模式, 目标位置, 目标速度 (Hz), 起停速度 (Hz), 加速度时间 (ms), 减速度时间 (ms), 完成后动作, 完成旗标 (...), and 下一段编号. The first row contains the values: 1, 单轴点对点, 轴 1, 相对模式, 0, 0, 0, 0, 0, 结束, and ---. A blue box labeled '编辑区' (Edit Area) highlights the first row. Below the table are navigation icons (search, home, up, down). At the bottom, there are tabs for '单轴点对点', '单轴多段输出', '双轴直线补间', and '双轴圆弧补间'. Below these tabs is another table with the same columns as the one above. A blue box labeled '列表区' (List Area) highlights the first row of this table.

要新增一笔数据，先在编辑填入各项设定字段后，按下「新增」，便会依据选择的轴控模式，加入到下方相对的轴控模式分页中。

This screenshot shows the '新增' (Add) button being clicked. A blue callout box with the number '23' and a blue arrow points to the '新增' button. Another blue callout box with the number '1' and a blue arrow points to the '完成后动作' (Action after completion) dropdown menu in the table. The table data is the same as in the previous screenshot.

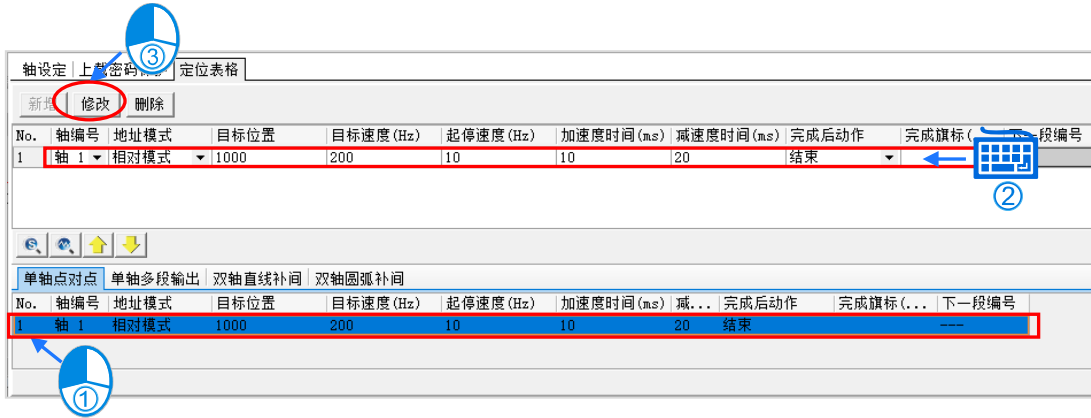


This screenshot shows the '列表区' (List Area) after the data has been added. The table now contains the following data:



No.	轴编号	地址模式	目标位置	目标速度 (Hz)	起停速度 (Hz)	加速度时间 (ms)	减速度时间 (ms)	完成后动作	完成旗标 (...)	下一段编号
1	轴 1	相对模式	1000	200	10	10	20	结束		---

The first row of the table is highlighted with a red border. Below the table are navigation icons (search, home, up, down). At the bottom, there are tabs for '单轴点对点', '单轴多段输出', '双轴直线补间', and '双轴圆弧补间'. Below these tabs is another table with the same columns as the one above. The first row of this table is also highlighted with a red border.

要修改则是在列表区选取该笔数据，之后再于编辑区编辑此笔数据内容，最后按下「修改」，便会更新到列表区中，如下图。



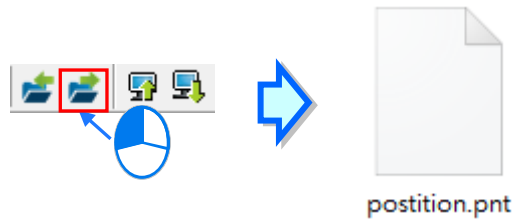
若要删除则是在列表区选取该笔数据，之后再于编辑区，按下「删除」，此笔数据便会从列表区中删除。

在列表区的每一列编号，可由上移  / 下移  来做编号增减的动作，但需注意字段「完成后动作」、「下一段编号」及「完成旗标 (I/M)」是不会随着移动，如下图。


23

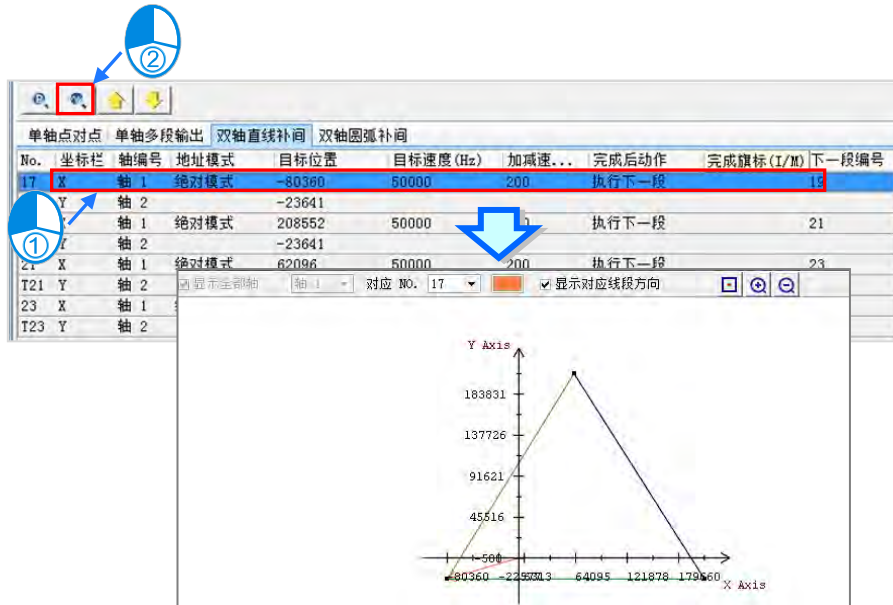


完成定位表格数据，可以单击「导出」，将文件导出成.pnt，如下图，之后就可以单击「导入」，选取.pnt文件，定位表格数据就会被导入。




23.1.3 定位规划表的仿真

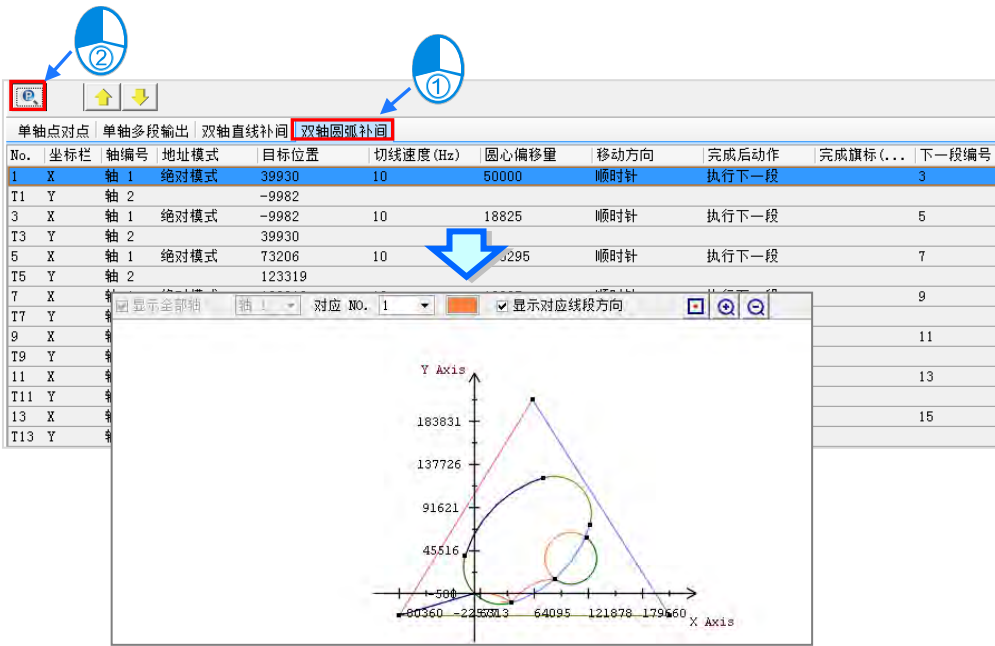
在列表区单击要仿真的任一列后，再按「指定编号绘图」，则会弹出仿真图窗口，通过图形化的方式显示规划表的路径，如下图，此时会绘制从 No.17 开始到结束的路径。



仿真图工具栏说明如下。

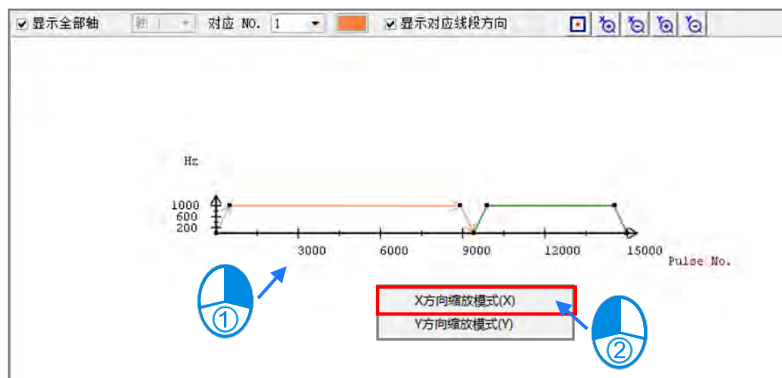
图示	功能说明
<input type="checkbox"/> 显示全部轴 轴 1 ▾	勾选显示全部轴或是只显示某一轴。
对应 NO. 1 ▾ 	选择某一编号在图形中加粗标记，也可单击右侧的色块变更颜色。
<input checked="" type="checkbox"/> 显示对应线段方向	显示箭头代表运动方向。
	调整图形为最适大小。
	依 X 方向放大图形。
	依 X 方向缩小图形。
	依 Y 方向放大图形。
	依 Y 方向缩小图形。
	放大图形。
	缩小图形。

双轴直线差补模式与双轴圆弧差补模式，可在同一仿真图中显示，混合两模式连续的轨迹，先点击列表区的「双轴圆弧补间」分页，再单击「双轴仿真模式」，如下图。



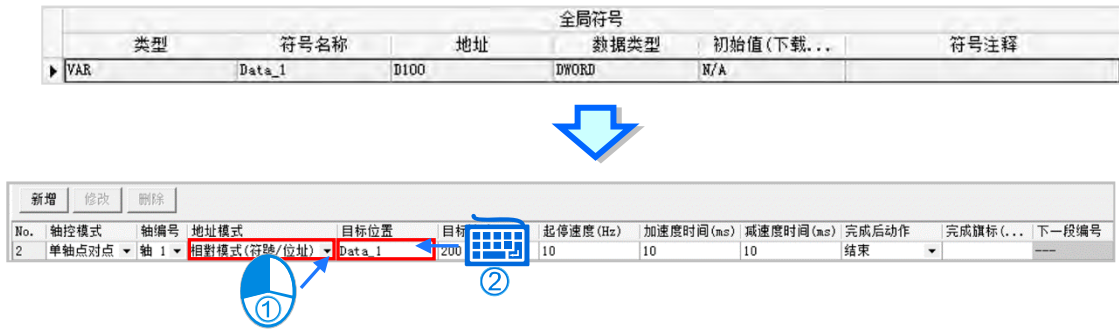
23

单轴点对点模式与单轴多段输出模式的仿真图则只能显示同一模式的轨迹，先点击列表区的「单轴点对点」分页，再单击「单轴仿真模式」，且在单轴仿真图窗口空白处，点击鼠标右键，之后会弹出选单，可以切换滚轮要控制的方向，如下图。

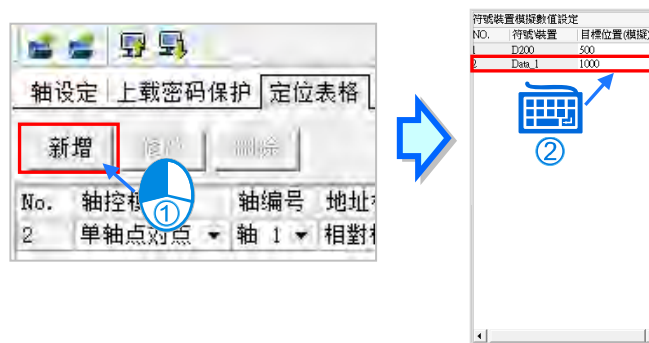


目标仿真区的功能，则需先在全局变量宣告符号，必须是 DWORD 或 DINT 以上的类型，接着改变编辑区的地址模式为「绝对模式 (符号/地址)」或「相对位置 (符号/地址)」，再于「目标位置」填入符号或装置，如下图。

⚠定位规划表的「目标位置」最小单位为 DWORD，所以若再填入 D100 则会占掉 D100 及 D101 长度，请小心使用!



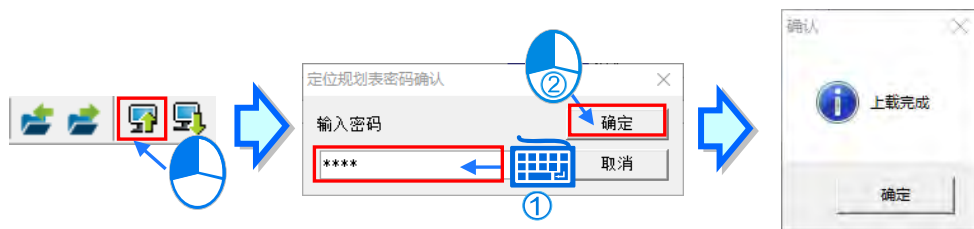
填完后，点击「新增」后在目标仿真区，将自动增加此笔数据，此时可以在此笔数据的目标位置（仿真）的项目，输入欲仿真的目标位置值，之后可以通过「单轴仿真模式」或「双轴仿真模式」，来检视仿真路径。



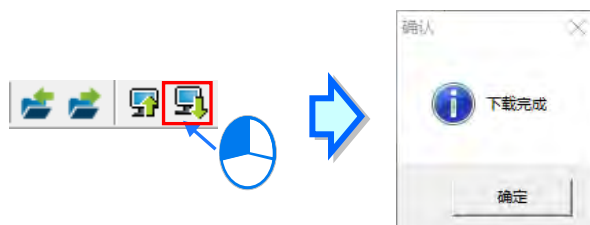
23

23.1.4 下载与上载定位规划表

使用上载与下载定位规划表功能，需先将 ISPSOFT 与主机正常联机，便可以单击「上载」将主机保存的定位规划表上载，当主机中的定位规划表有上载密码保护时，将出现密码询问窗口，输入正确的密码才可将定位规划表上载，如下图。



单击「下载」按钮，则可以将定位规划表下载至主机中。



23.1.5 轴控模式-单轴点对点模式

设定单轴点对点模式，说明如下。

No.	轴控模式	轴编号	地址模式	目标位置	目标速度(Hz)	起停速度(Hz)	加速度时间(ms)	减速度时间(ms)	完成后动作	完成旗标(...)	下一段编号
1	单轴点对点	轴 1	绝对模式	0	0	0	0	0	结束		---

- **No.**：编号由系统自动产生，最大为 200。
- **轴控模式**：选择运动的类型。此处选择为**单轴点对点**。
- **轴编号**：此笔数据控制的轴。
- **地址模式**：「**相对模式**」依据目前的坐标地址而增减。
 「**绝对模式**」依据绝对坐标做为目标地址而增减。
 「**相对模式 (符号/地址)**」依据目前所代表的坐标地址而增减，可以使用符号/地址。
 「**绝对模式 (符号/地址)**」依据绝对坐标做为目标地址而增减，可以使用符号/地址。
- **目标位置**：目标坐标。**地址模式**若选取包含 (符号/位置)，则这边可以使用符号/位置。
- **目标速度**：目标的最高速度。
- **起停速度**：开始与结束的速度。
- **加速度时间**：到达目标速度的时间。
- **减速度时间**：到达停止的时间。
- **完成后动作**：「**执行下一段**」完成此笔数据后接续到下一笔。
 「**结束**」完成此笔数据后结束动作。
- **完成旗标 (I/M)**：让此笔数据执行完毕时，将寄存器设为 ON，但只有**完成后动作**设为**结束时**才可设定寄存器范围为 M0~M8191 或 I510~I519。
- **下一段编号**：接续下一笔数据的编号，不限制指定同一个轴编号，但是须注意同一轴编号不可有两个以上表格编号或定位输出指令同时输出。若**完成后动作**设为**结束后**则自动带入 ---。

表格输入注意事项：

1. 目标速度不可输入为 0；在相对模式下，目标位置不可输入为 0。
2. I 中断编号可重复使用，但用户须自行判断此中断为哪一个列编号输出完成。
3. 启动表格输出的指令为 API2718 TPO 指令；当 PLC 正在运行中，可运用 API2719 TPWS 修改输出参数。

23.1.6 轴控模式-单轴多段输出模式

设定单轴多段输出模式，说明如下。

No.	轴控模式	轴编号	地址模式	目标位置	目标速度(Hz)	加减速时间(ms)	中断讯号(I/M)	完成后动作	完成旗标(...)	下一段编号
1	单轴多段输出	轴 1	相对模式	0	0	0		执行下一段		2

- **No.**：编号由系统自动产生，最大为 200。
- **轴控模式**：选择运动的类型。此处选择为**单轴多段输出**。
- **轴编号**：此笔数据控制的轴。

- **地址模式**：「相对模式」依据目前的坐标地址而增减。
 - 「绝对模式」依据绝对坐标做为目标地址而增减。
 - 「相对模式 (符号/地址)」依据目前所代表的坐标地址而增减，可以使用符号/地址。
 - 「绝对模式 (符号/地址)」依据绝对坐标做为目标地址而增减，可以使用符号/地址。
- **目标位置**：目标坐标。地址模式若选取包含 (符号/位置)，则这边可以使用符号/位置。
- **目标速度**：目标的最高速度。
- **加减速时间**：到达目标速度及到达停止的时间。
- **中断讯号 (I/M)**：目前表格进行尚未完成时，若中断讯号设定的寄存器为 ON，则进行下一段定位表格。可输入 I 中断编号或 M 寄存器名称。支持的中断编号请参考产品相关程序手册。
- **完成后动作**：「执行下一段」完成此笔数据后接续到下一笔。
 - 「结束」完成此笔数据后结束动作。
- **完成旗标 (I/M)**：让此笔数据执行完毕时，将寄存器设为 ON，但只有完成后动作设为结束时才可设定寄存器范围为 M0~M8191 或 I510~I519。
- **下一段编号**：接续下一笔数据的编号，不限制指定同一个轴编号，但是须注意同一轴编号不可有两个以上表格编号或定位输出指令同时输出。若完成后动作设为结束后则自动带入 ---。

23

表格输入注意事项：

1. 在相对模式下，目标位置不可输入为 0；在绝对模式下，相邻的两个列编号的目标位置不可输入相同位置。
2. 若加减速时间不为 0，加速度间隔时间固定为 1ms；若加减速时间为 0，代表不作加减速。
3. I 中断编号可重复使用，但用户须自行判断此中断为哪一个列编号输出完成。
4. 启动表格输出的指令为 API2718 TPO 指令；当 PLC 正在运行中，可运用 API2719 TPWS 修改输出参数。

23.1.7 轴控模式-双轴直线插补模式

设定双轴直线插补模式，在 XY 平面上的直线差补运动，使用此模式每次会占用两笔数据。

No.	轴控模式	坐标栏	轴编号	地址模式	目标位置	目标速度(Hz)	加减速时...	完成后动作	完成旗标(I/M)	下一段编号
1	双轴直线插补	X	轴 1	相对模式	0	0	0	结束		---
T1		Y	轴 2		0					

- **No.**：编号由系统自动产生，双轴模式会占用两笔数据，第二笔数据的编号前会加注 T，因此全部使用双轴模式时，编号最大为 100 与 T100 共 100 组。
- **轴控模式**：选择运动的类型。此处选择为双轴直线差补。
- **坐标栏**：此笔数据在仿真图中的 X 或 Y。
- **轴编号**：此笔数据控制的轴。
- **地址模式**：「相对模式」依据目前的坐标地址而增减。
 - 「绝对模式」依据绝对坐标做为目标地址而增减。
 - 「相对模式 (符号/地址)」依据目前所代表的坐标地址而增减，可以使用符号/地址。

「绝对模式 (符号/地址)」依据绝对坐标做为目标地址而增减，可以使用符号/地址。

- **目标位置**：目标坐标。**地址模式**若选取包含 (符号/位置)，则这边可以使用符号/位置。
 - **目标速度**：目标的最高速度。
 - **加减速时间**：到达目标速度及到达停止的时间。
 - **完成后动作**：「执行下一段」完成此笔数据后接续到下一笔。
- 「结束」完成此笔数据后结束动作。
- **完成旗标 (I/M)**：让此笔数据执行完毕时，将寄存器设为 ON，但只有**完成后动作**设为**结束时**才可设定寄存器范围为 M0~M8191 或 I510~I519。
 - **下一段编号**：接续下一笔数据的编号，不限制指定同一个轴编号，但是须注意同一轴编号不可有两个以上表格编号或定位输出指令同时输出。若**完成后动作**设为**结束后**则自动带入 ---。

表格输入注意事项：

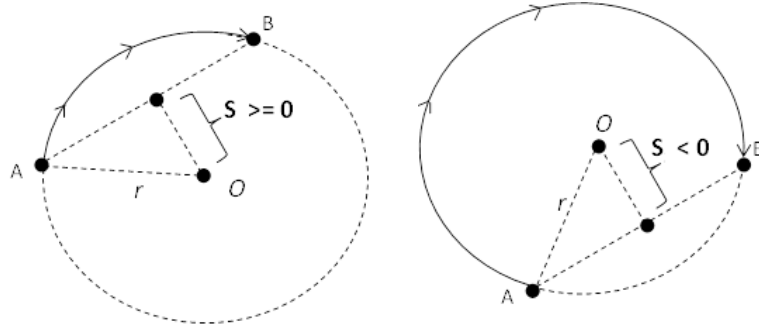
- 23
1. 当目标速度或加减速时间输入为 0 时，PLC 会自动以默认之速度；在相对模式下，目标位置不可输入为 0；在绝对模式下，相邻的两个列编号的目标位置不可输入相同位置。
 2. I 中断编号可重复使用，但用户须自行判断此中断为哪一个列编号输出完成。
 3. 启动表格输出的指令为 API2718 TPO 指令；当 PLC 正在运行中，可运用 API2720 TPWL 修改输出参数。

23.1.8 轴控模式-双轴圆弧插补模式

设定双轴圆弧插补模式，在 XY 平面上的圆弧插补运动，使用此模式每次会占用两笔数据。说明如下。

No.	轴控模式	坐标栏	轴编号	地址模式	目标位置	切线速度(Hz)	圆心偏移量	移动方向	完成后动作	完成旗标(I/M)	下一段编号
1	双轴圆弧插补	X	轴 1	相对模式	0	0	0	顺时针	结束		---
T1		Y	轴 2		0						

- **No.**：编号由系统自动产生，双轴模式会占用两笔数据，第二笔数据的编号前会加注 T，因此全部使用双轴模式时，编号最大为 100 与 T100 共 100 组。
- **轴控模式**：选择运动的类型。此处选择为**双轴圆弧插补**。
- **坐标栏**：此笔数据在仿真图中的 X 或 Y。
- **轴编号**：此笔数据控制的轴。
- **地址模式**：「相对模式」依据目前的坐标地址而增减。
 - 「绝对模式」依据绝对坐标做为目标地址而增减。
 - 「相对模式 (符号/地址)」依据目前所代表的坐标地址而增减，可以使用符号/地址。
 - 「绝对模式 (符号/地址)」依据绝对坐标做为目标地址而增减，可以使用符号/地址。
- **目标位置**：目标坐标。**地址模式**若选取包含 (符号/位置)，则这边可以使用符号/位置。
- **切线速度**：沿圆弧切线的方向运动的向量速度。
- **圆心偏移量**：圆弧目标点与起始位置的联机中心到圆心的距离。如下图的 S，由 A 点到 B 点，顺时针，S>0 代表要画小于 180° 的弧；S<0 代表要画大于 180° 的弧。



- **移动方向**：选择顺时针或逆时针方向。
- **完成后动作**：「执行下一段」完成此笔数据后接续到下一笔。
「结束」完成此笔数据后结束动作。
- **完成旗标 (I/M)**：让此笔数据执行完毕时，将寄存器设为 ON，但只有完成后动作设为结束时才可设定寄存器范围为 M0~M8191 或 I510~I519。
- **下一段编号**：接续下一笔数据的编号，不限制指定同一个轴编号，但是须注意同一轴编号不可有两个以上表格编号或定位输出指令同时输出。若完成后动作设为结束后则自动带入 ---。

23

表格输入注意事项：

1. PLC 会依切线速度，自动计算各段速度；在相对模式下，目标位置不可输入为 0；在绝对模式下，相邻的两个列编号的目标位置不可输入相同位置。
2. I 中断编号可重复使用，但用户须自行判断此中断为哪一个列编号输出完成。
3. 启动表格输出的指令为 API2718 TPO 指令；当 PLC 正在运行中，可运用 API2721 TPWC 修改输出参数。

23.2 数据示波器

23.2.1 关于数据示波器

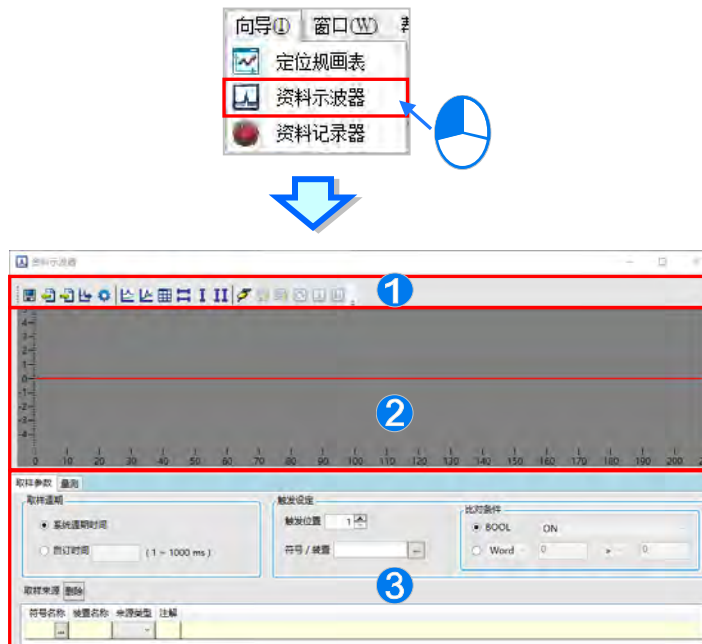
数据示波器用于在某个触发条件满足后，实时收集变量符号或装置的数值与状态，以波形图显示，方便分析数值趋势。新一代的 DIA 示波器提供全新的整合操作接口，让用户更快速方便的完成操作设定，支持机种可以参考下列表格，DIA 示波器详细用法可参考第 23.2.6 节。

机种 软件	AS	AH5x1	AHxxEMC	AS5xx DVPxxMC
示波器	V	V	V	
DIA 示波器			V (注 1)	V

23 注 1：在菜单工具栏单击 **工具 (I)** > **选项 (O)** > **初始化设定**，可以看到**数据示波器**项目，勾选**启用 SyncMode** 接着按确定即可切换，之后按照一般开启方式于**精灵 (I)** > **数据示波器**就可以打开 DIA 示波器。

23.2.2 开启数据示波器

要使用此功能时，请先编译目前项目，再单击 **向导 (I)** > **数据示波器**，开启数据示波器窗口，如下图所示。



- ① 图示工具栏：提供各项操作数据示波器的按钮，功能说明如下。
- ② 波形显示区：显示实时变量符号或装置的波形图。
- ③ 参数设定区：取样参数分页用于设定取样的来源与触发条件；量测分页用于显示量测工具所量测到的数值及调整波形的颜色与比例等。

图标工具栏按钮功能说明

图示	名称	功能说明
	保存全部	将取样设定数据与取样波形数据结果保存到指定路径中。
	导出取样设定	将取样设定内容导出到指定路径。
	导入取样设定	由指定路径将取样设定内容导入。
	导出波形数据	将取样波形数据以 .bmp 或 .csv 格式导出到指定路径。
	设定	设定数据示波器波形与环境的颜色。
	分离波形	将多条波形自动以间隔分离。
	重迭波形	将多条波形以中心线重迭。
	显示网格线	切换是否显示网格线。
	水平量测线	波形显示区将出现两条水平线，供用户量测波形。
	垂直量测线	波形显示区将出现一条垂直线，供用户量测波形。
	双垂直量测线	波形显示区将出现两条垂直线，供用户量测波形。
	联机/脱机	切换计算机与主机的联机或脱机模式。
	上载 PLC 取样设定	从主机读回数据示波器的取样设定。
	下载取样设定到 PLC	将数据示波器的取样设定写入主机。
	实时监看	计算机实时从主机读取来源的取样波形。
	单次触发	触发条件满足后，从主机读取触发前后共 200 点波形数据一次。
	连续触发	每当触发条件满足后，便从主机读取触发前后共 200 点波形数据。

23

23.2.3 设定取样参数

在取样参数分页，依照开启的项目机种不同，项目会有所不同请参考下表。**取样周期**是要设定取样的时间。**触发条件**则是要设定触发位置与条件，当触发条件满足时，主机将保存共 200 笔的取样数据，并显示到波形显示区。



取样週期

- 系统週期时间
- ECAT 週期时间倍数 (1 ~ 1000)

触发设定

触发位置: 1

符号 / 装置: [...]

触发点数: 10000

比对条件

- BOOL ON
- Word 0 > 0

名称	说明	AS	AHxxEMC	AH5x1
系统周期时间	主机扫描周期	V	V	V
ECAT 周期时间倍数	ECAT 扫描周期		最大 1000	
自定义时间	自定义的时间	最大 1000		最大 1000

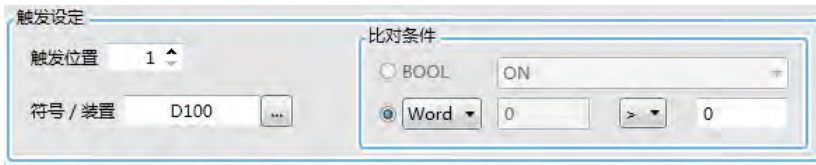
名称	说明	AS	AHxxEMC	AH5x1
触发位置	在图上要触发的位置	最大 200	最大 10000	最大 200
符号/装置	触发的装置/符号	只支持全局符号	支持全局符号、区域符号、轴符号、ECAT 符号、STRUCT 符号	只支持全局符号
触发点数	设定触发位置的最大值		最大 10000	



当符号/装置字段选择为 **BOOL** 的变量符号或装置时，比对条件会自动选择到 **BOOL**，如下图，也可在下拉选单选择 **ON**、**OFF**、**上升沿触发**、**下降沿触发**、**上升沿或下降沿触发**。当设定的符号或装置满足此设定条件时，则触发显示在数据示波器上。

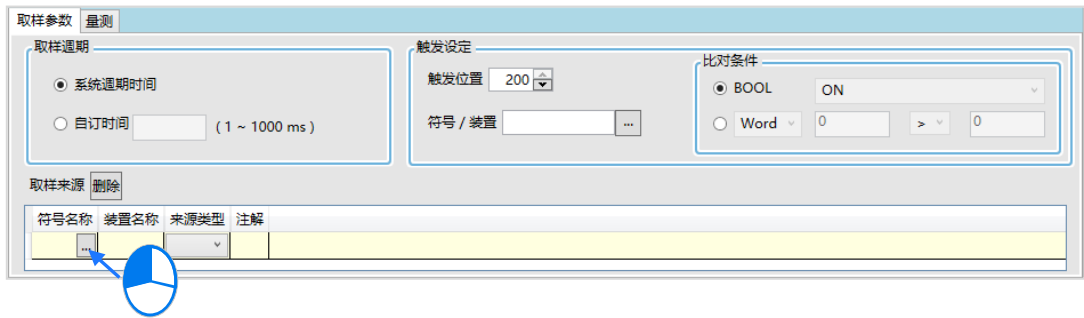


23

当符号/装置字段选择为 **Word** 或 **DWord** 或 **Real** 的变量符号或装置时，比对条件会自动选择到 **Word**，如下图，也可在下拉选单选择 **DWord** 或 **Real** 数据类型，并使用条件式 **>**、**<**、**=**、**!=**、**~**、**!~**，搭配前后的数字字段来做设定。当设定的符号或装置满足此条件时，则触发显示在数据示波器上。如下图，为当 **D100** 的数值为 **0 > 0** 时，满足触发条件；若设定为 **10 !~ 100**，则代表 **D100** 为未满 10 或超过 100，才满足触发条件。



要新增取样符号，先单击符号名称的  按钮，接着在弹出的窗口单击符  按钮，选择要加入的变量符号，或直接在 **装置名称** 字段输入设备名称，按下「**新增**」后，就会加入取样来源列表中，如下图。



OR



符号名称	装置名称	来源类型	注解
Var_1	D100	VALUE	

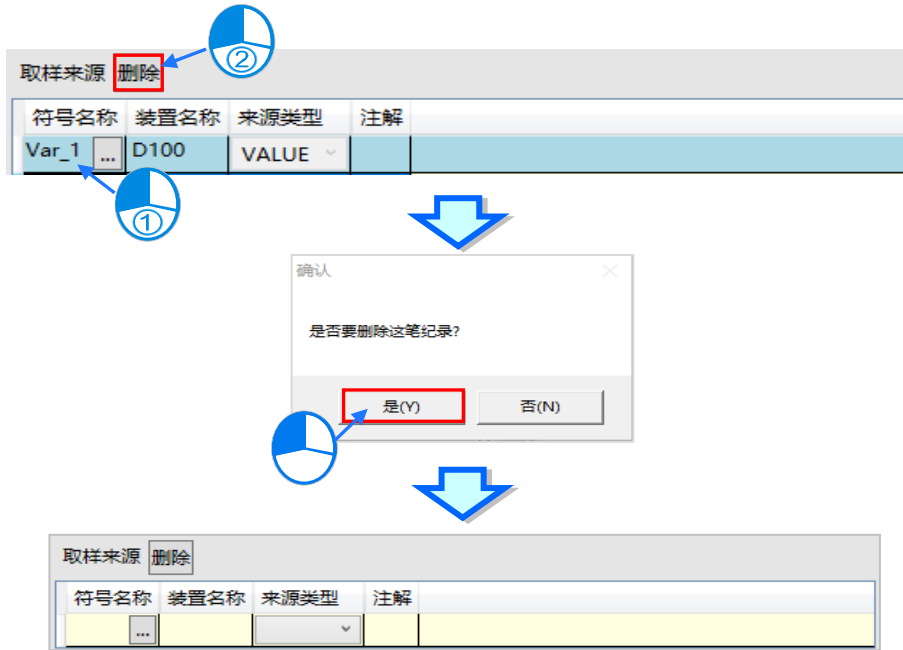
只有宣告在全局符号表的变量，并且编译过项目，才可以在这边被选取，若有手动设定装置地址或符号批注，也会一并带入。

全局符号					
类型	符号名称	地址	数据类型	初始值...	符号注释
VAR	Var_1	D100	WORD	N/A	



符号名称	装置名称	来源类型	注解
Var_1	D100	VALUE	

23

要删除已加入的取样来源时，单击要删除的项目后，按下「删除」按钮，即可把该项目删除。

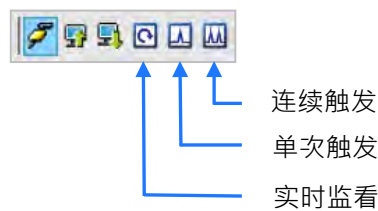


23

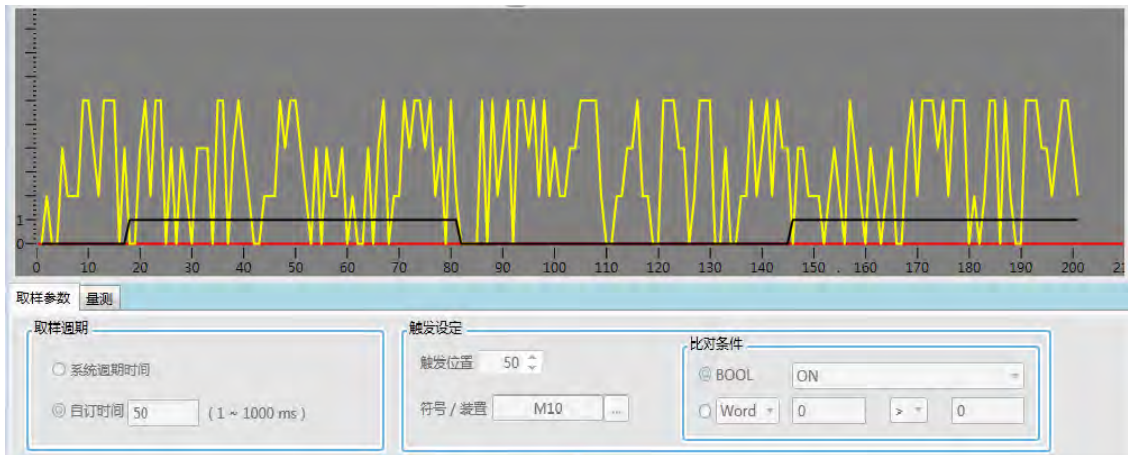
完成取样参数的设定后，可按下图示工具栏的「导出」，将取样参数设定保存成.dtr 档，之后可以在新项目使用「导入」将取样参数设定还原。

23.2.4 取样模式

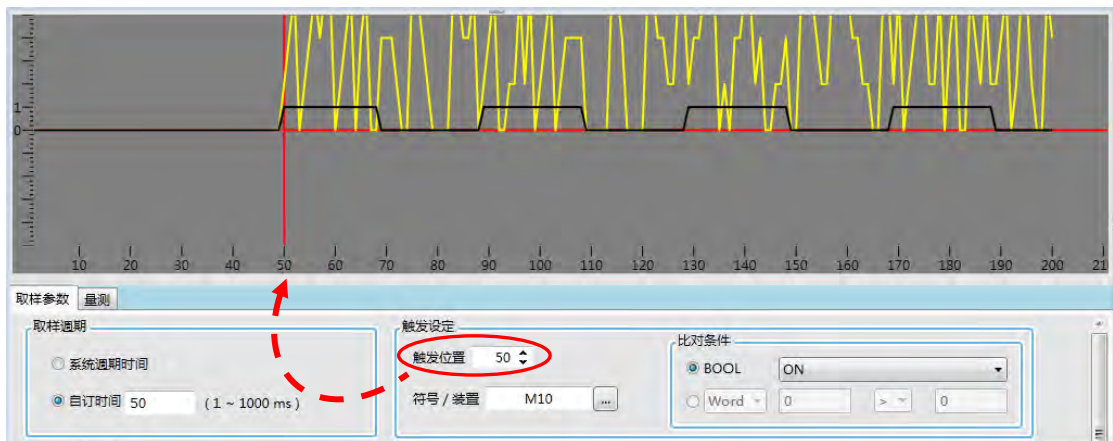
完成数据示波器的取样参数设定后，在联机模式时，先下载参数设定，便可以使用图标工具栏的按钮切换下述三种模式监看波形。



- 实时监看：依据设定的取样周期，将取样来源的变量符号或装置的数值或状态不断绘制到数据示波器上。画面会随之向左卷动，只保留最新的 200 笔数据。



- 单次触发：依据前述的触发位置设定，当触发条件满足时，将触发位置之后的取样数据，显示在波形区，只触发一次。如下图，因触发位置设为 50，故当触发条件 M10=OFF 时，会将取样来源依据取样周期取样绘制到 50~200 的范围中。波形中的红色垂直线即代表触发位置。



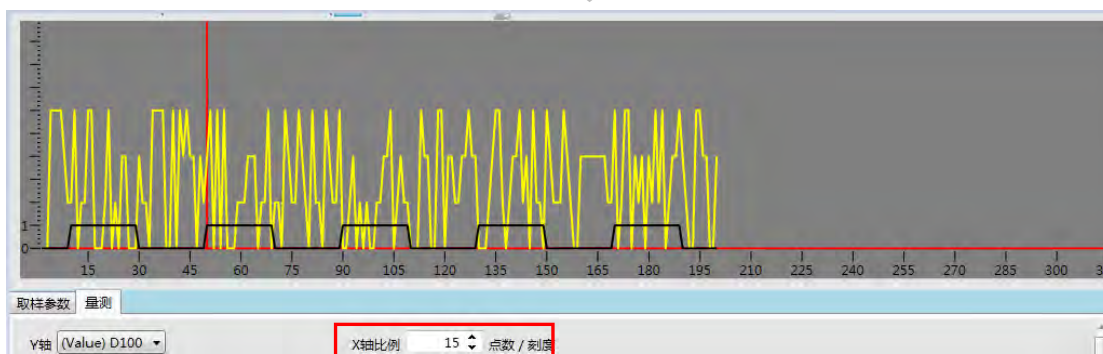
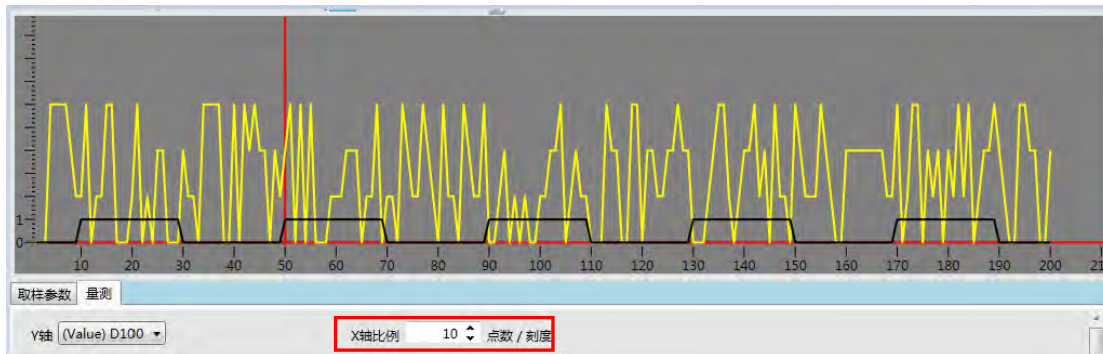
- 连续触发：依据前述的触发位置设定，数据示波器将保持不断读取的状态，而每次当触发条件满足时，将触发位置前后的取样数据，自动显示在数据示波器上，要观看图形时，须将连续触发的按钮关闭恢复为未按下的状态。如下图，因触发位置设为 50，故当触发条件 M10=ON 时，会将取样来源依据取样周期取样绘制 200 笔，包括触发位置前 49 笔与之后的 150 笔。波形中的红色垂直线即代表触发位置。



23

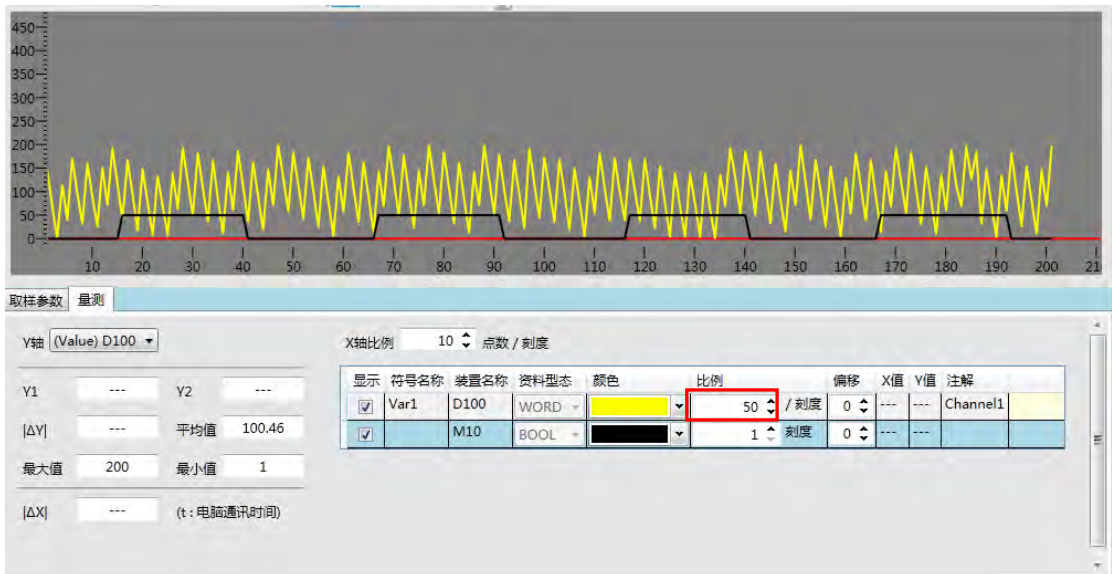
23.2.5 量测

在量测页面，可以调整数据示波器的波形外观，以达更好的呈现效果。如下图，变更 X 轴比例用以调整横坐标每刻度的点数。

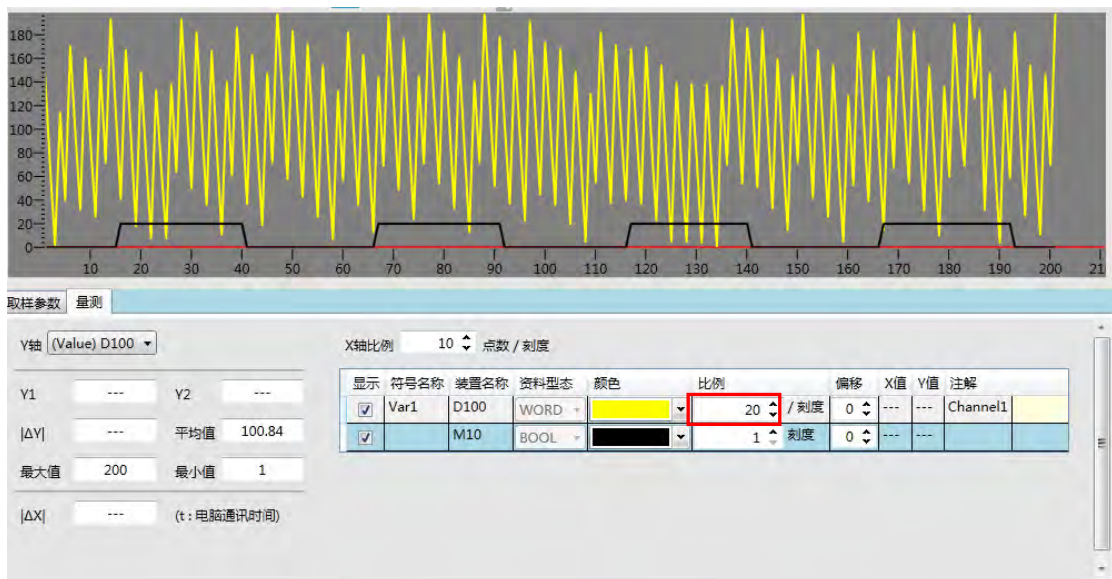


23

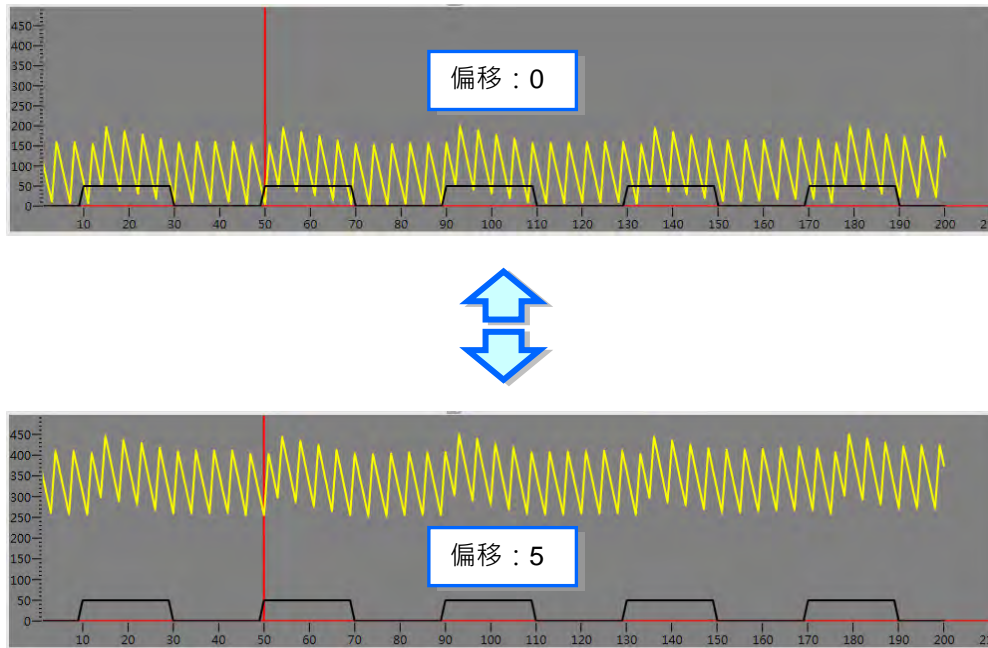
量测页面的平均值、最大值、最小值为 Y 轴所选择装置波形的信息。每个波形也可以设定显示的颜色、比例与偏移。比例为纵坐标的比例，下图为比例设为每刻度的值为 50 与 20 的比较。



23



设定**偏移**，可调整波形在画面中的位置，如下图。

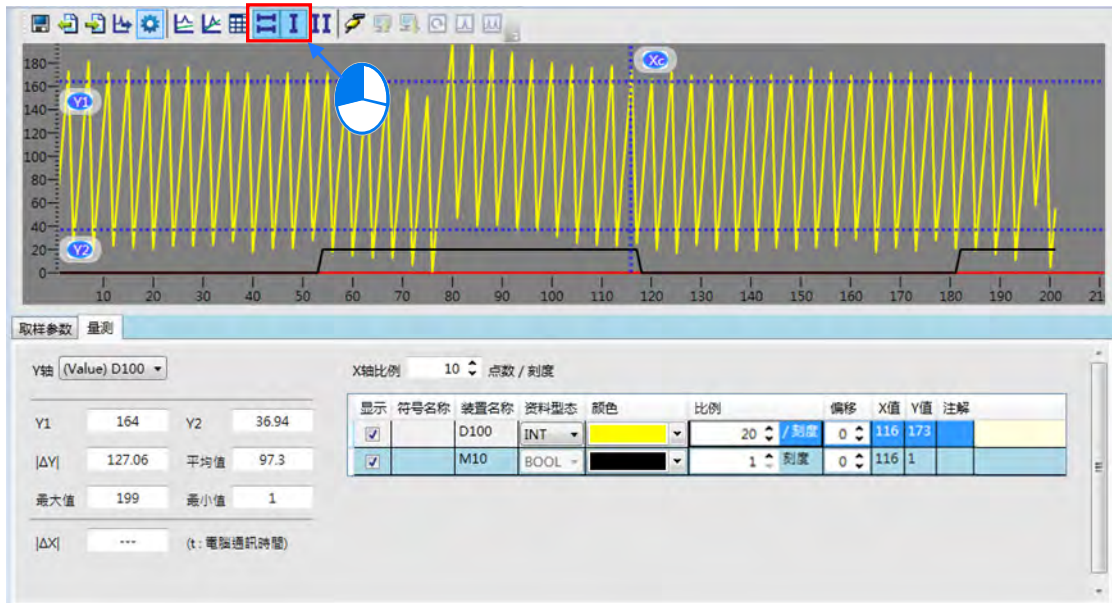


23

搭配水平量测线与垂直量测线，可以量测数据示波器的波形数值。如下图，单击 按钮开启水平量测线，单击 按钮开启垂直量测线。

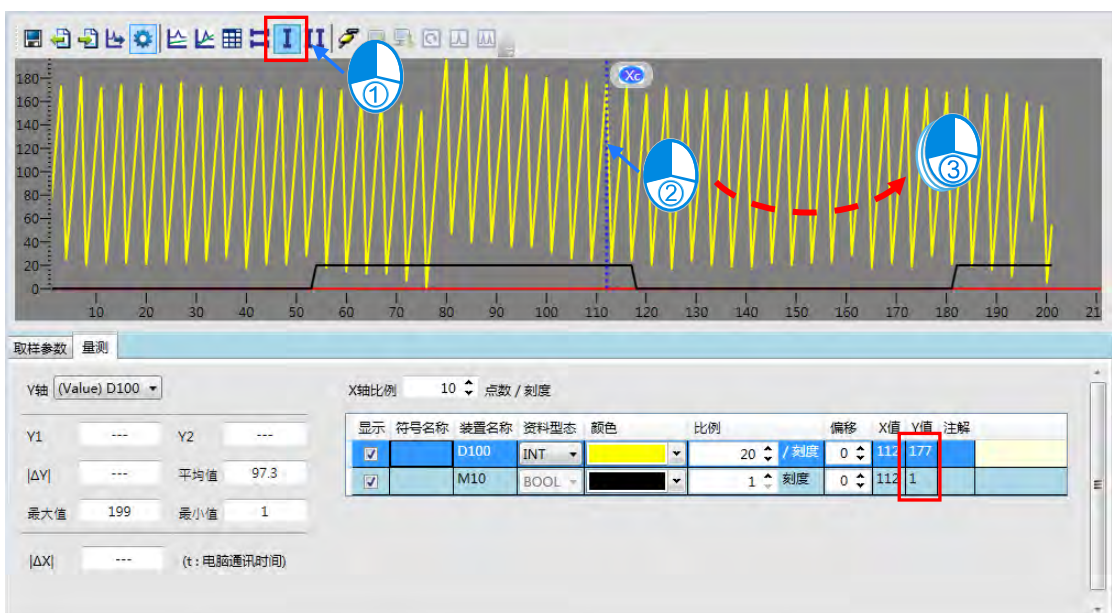
显示	符号名称	装置名称	资料型态	颜色	比例	偏移	X值	Y值	注解
<input checked="" type="checkbox"/>	D100		INT	黄色	20 / 刻度	0	116	173	
<input checked="" type="checkbox"/>	M10		BOOL	黑色	1 / 刻度	0	116	1	

搭配水平量测线，可以拖曳两条水平量测线 Y1 与 Y2，调整到要测量的位置。于参数设定区便会显示水平量测线的各项数值，**Y1** 字段代 Y1 的数值，**Y2** 字段代表 Y2 的数值，**Y1-Y2** 字段代表 Y1 与 Y2 的差值。

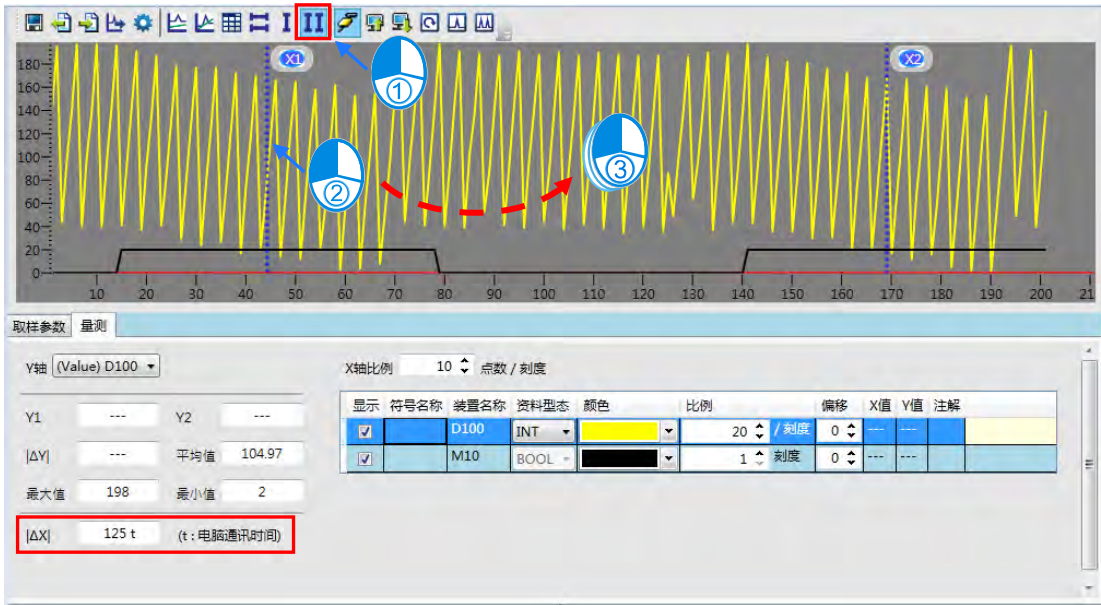


搭配垂直量测线，可以拖曳垂直量测线 Xc，则下方的 Y 值栏位，则会显示各取样来源与 Xc 交叉所对应的 Y 值。如下图。

23



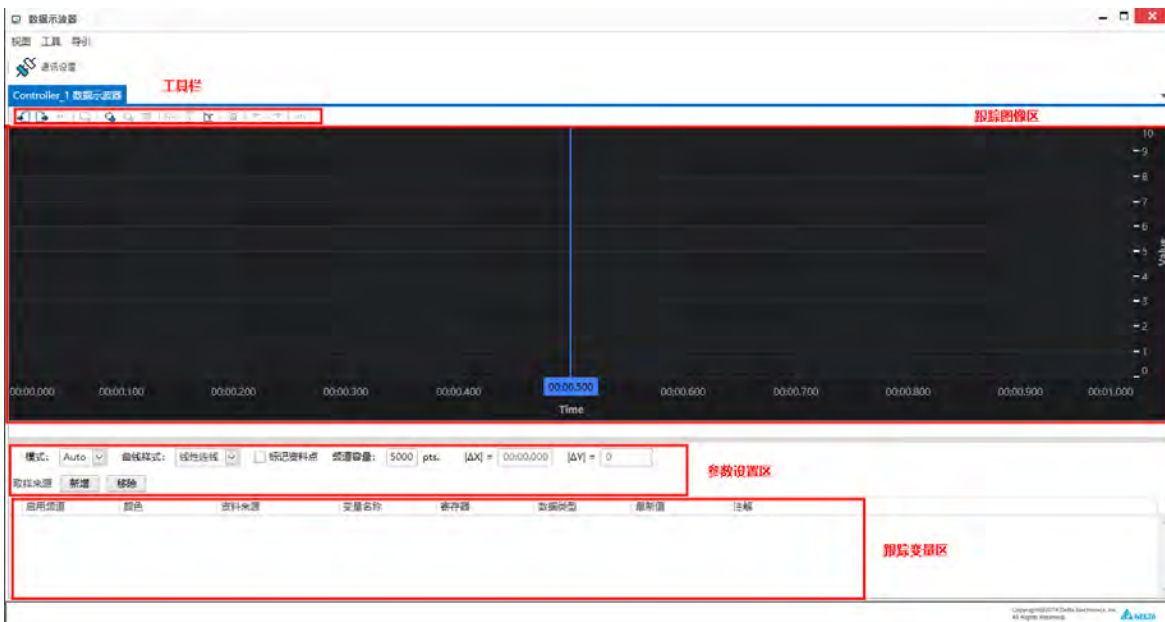
搭配双垂直量测线，可以量测数据示波器的时间值。如下图，单击 **II** 按钮开启双垂直量测线，以鼠标单击后拖曳两条垂直量测线 X1 与 X2，调整到要测量的位置，于下方的区域便会显示 **ΔX** 的时间值 (t 值为计算机与 PLC 两者一次通讯的时间)。



23

23.2.6 DIA 示波器

开启数据示波器软件后默认接口包含工具栏、跟踪图像区、参数设定区、跟踪变量区。如下图所示：



● 数据示波器工具栏

用户可以通过工具栏进行开启文件、储存文件、清除画面、开启示波器、停止示波器等操作。工具栏图及功能介绍如下：

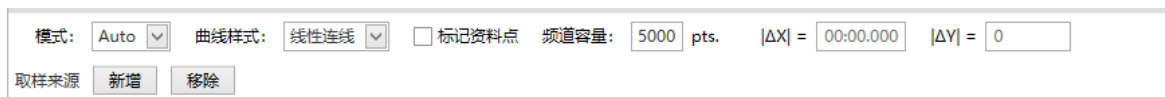
图示	功能介绍
	开启文件

图示	功能介绍
	储存文件
	导出
	画面截取
	开始示波器
	停止示波器
	清除画面
	垂直量测线
	水平量测线
	量测线置中
	最适合大小
	复原
	取消复原
	分析

23

● 数据示波器参数设定区

参数设定区内包含跟踪模式、曲线样式、标记数据点、频道容量、跟踪变量新增和移除等参数。某些参数在特定的跟踪模式下才需要设定，特定的参数将在示波器跟踪模式介绍时讲解。

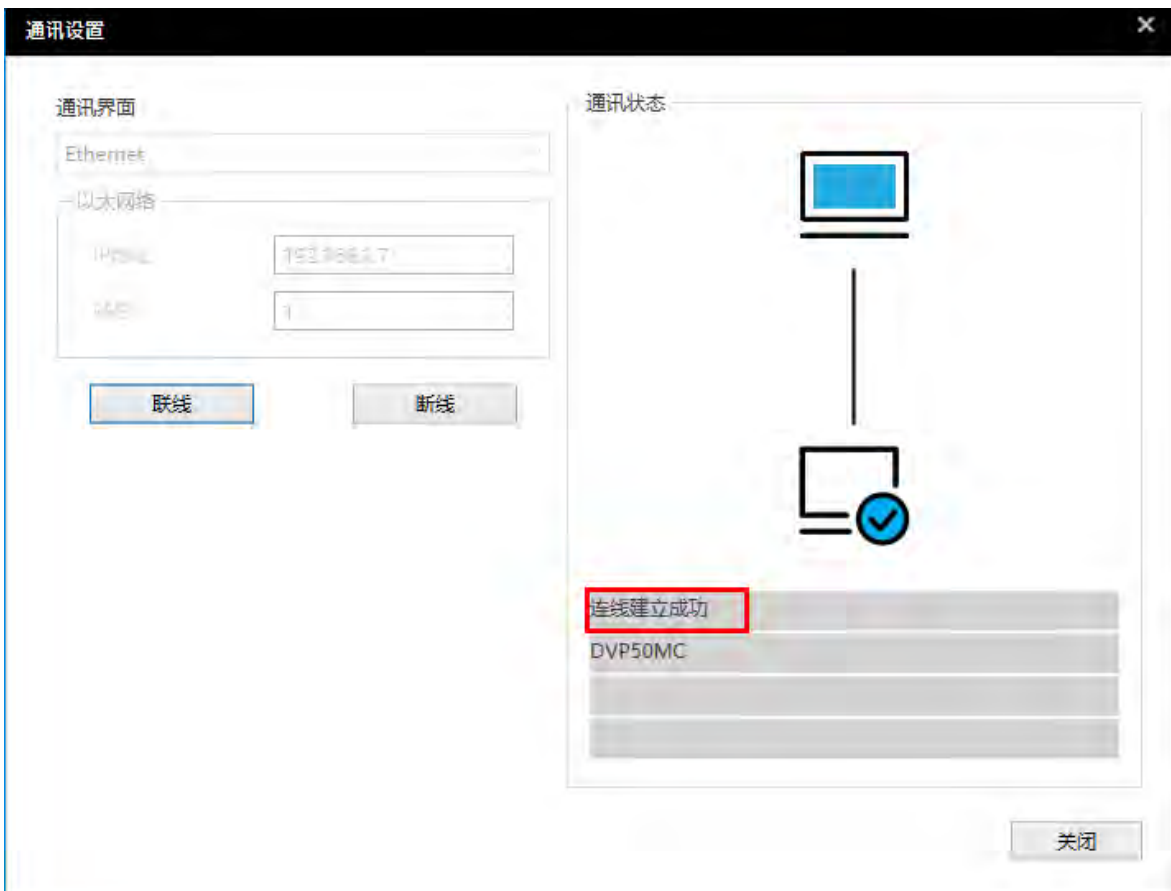


- ❶ 模式：数据示波器支持三种跟踪模式，分别为 Auto、Sync 及 Trigger 模式。
- ❷ 曲线样式：数据示波器支持两种曲线样式，分别为线性联机和数字联机。线性联机为数据点直接以直线形式连接；数字联机为数据点以梯形形式连接。
- ❸ 标记数据点：勾选标记数据点，可以在跟踪图像上显示数据点。
- ❹ 频道容量：跟踪图像上显示的数据点个数，数值越大跟踪图像的数据越多，即跟踪图像显示时长越长。Auto 模式时频道容量的最大值为 100000，Sync 模式时频道容量的最大值为 1000000。
- ❺ |ΔX|：垂直量测时差值。
- ❻ |ΔY|：水平量测时差值。

● 数据示波器操作

➤ 通信设置：

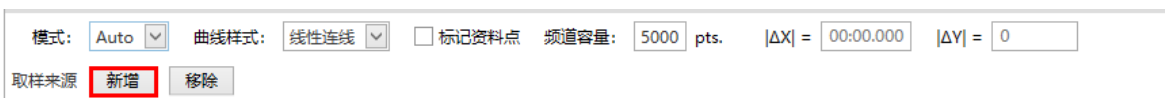
单击工具栏上方**通信设置**，会弹出通信设置接口，在进行变量跟踪时请确保通讯状态为联机建立成功，如下图所示。



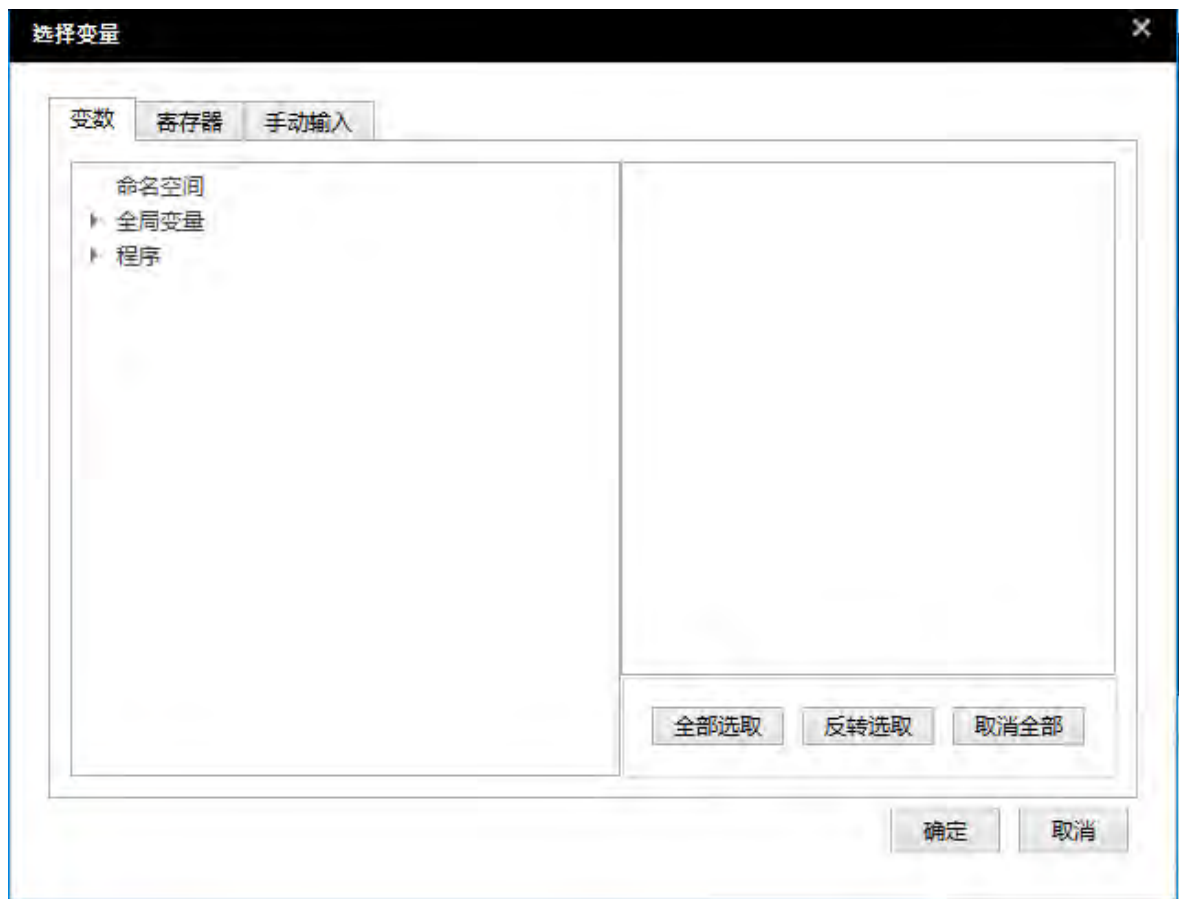
当打开数据示波器软件时，数据示波器软件会自动与控制器建立连接。若出现通讯状态为非联机建立成功状态，请检查以太网线，通信设置为 **Ethernet**，**IP 地址** 为控制器 IP，确认无误后单击 **联机** 按钮，进行手动连接。

➤ 新增跟踪变数：

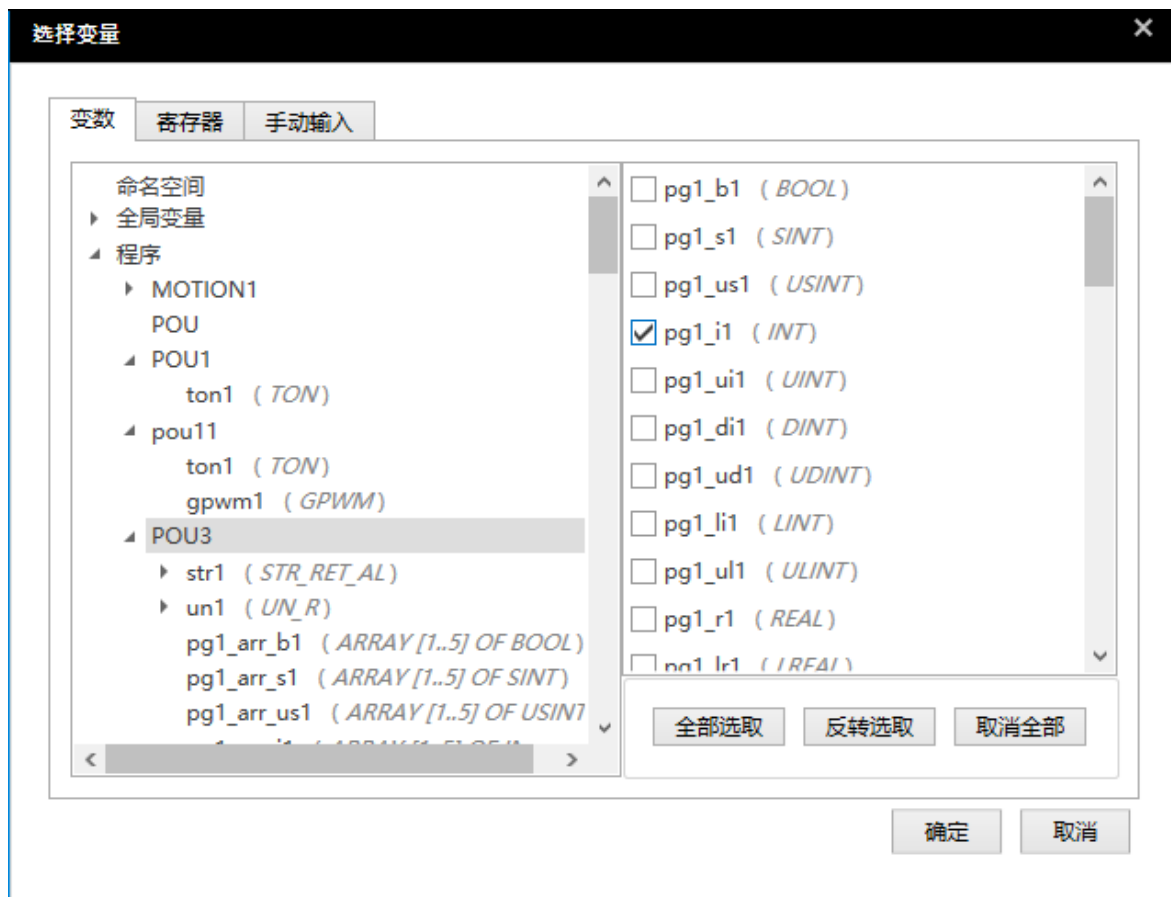
单击参数设定区取样来源 **新增** 按钮，如下图所示。



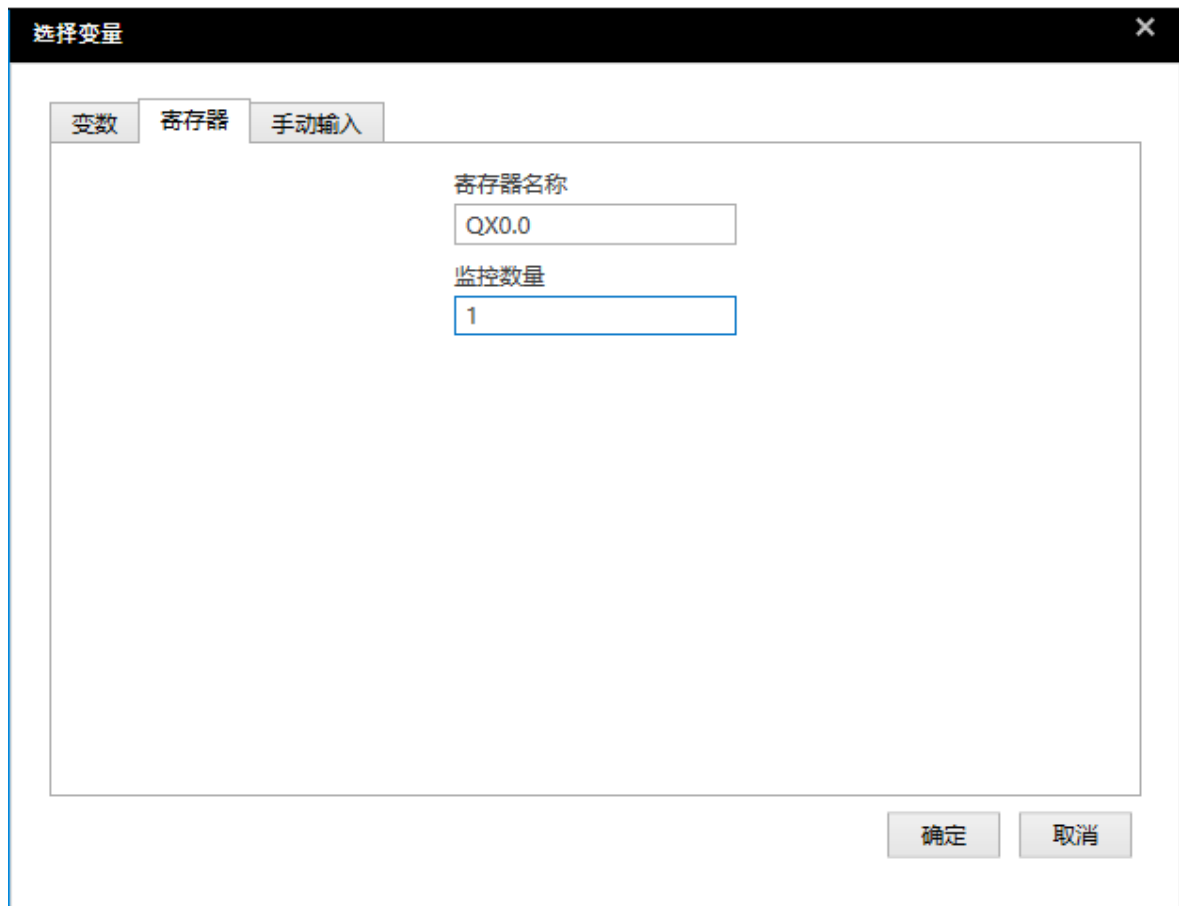
会弹出选择变量的弹窗，支持 **变量**、**寄存器**、**手动输入** 三种新增方式，如下图所示。



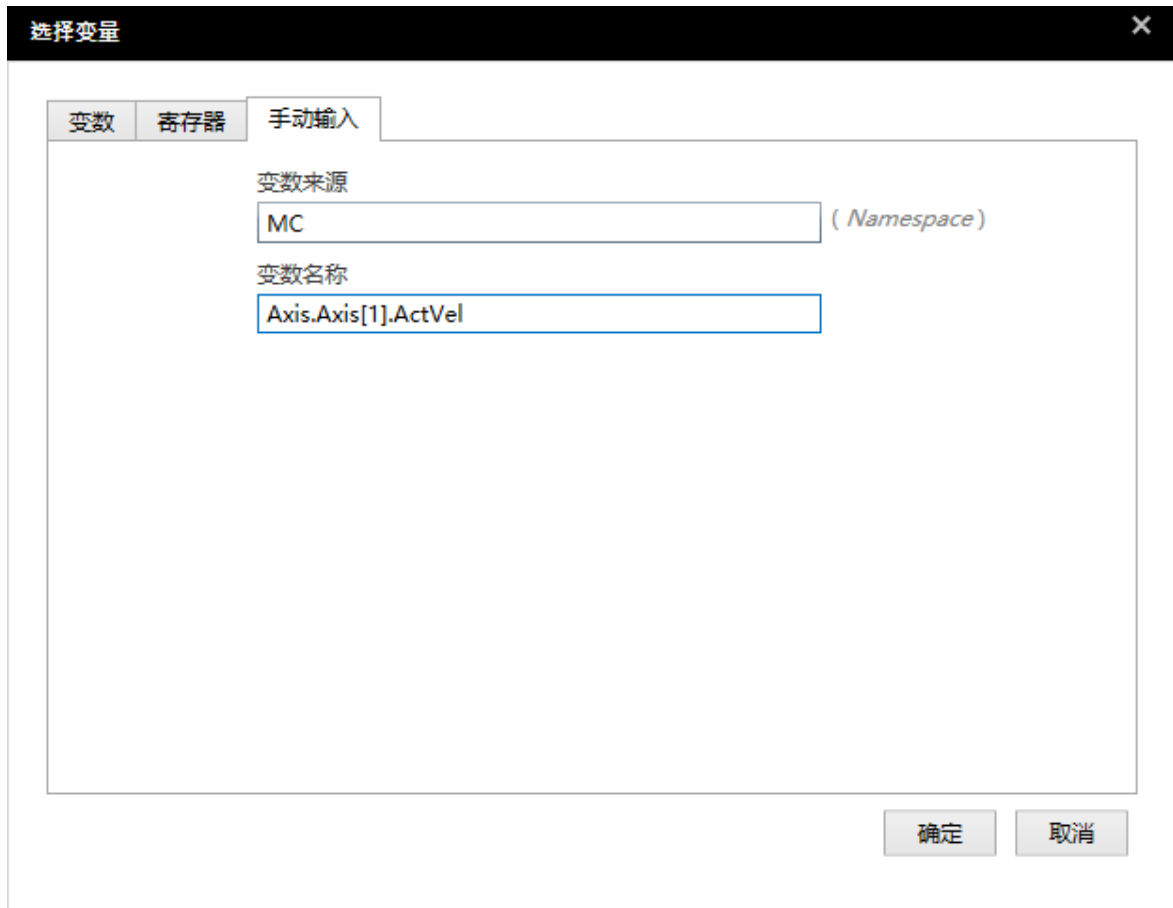
选择变量，单击**全局变量**前面的三角形展开后可以选全局变量；单击**程序**前面的三角形展开后可以选局部变量。勾选要跟踪的变量后单击**确定**按钮即可新增跟踪变量。如下图所示。




选择**寄存器**，在**寄存器名称**内填写欲跟踪的装置，在**监控数量**内填写装置数量，单击**确定**按钮即可新增跟踪变量。如下图所示。



选择**手动输入**，在**变量来源**内填写变量的来源，在**变量名称**内填写具体的跟踪变量后单击**确定**按钮可新增跟踪变量；或者选中**变量来源**单击方向下键显示所有变量来源，选择来源后单击 **Enter** 键，然后在**变量名称**中先单击方向下键可显示第一层展开提醒，然后单击方向右键可继续展开直至欲跟踪变数后，单击 **Enter** 键可新增跟踪变数。如下图所示。



- **删除跟踪变数**：选中要删除的跟踪变量，单击参数设定区取样来源**移除**按钮。
- **开始示波器跟踪**：单击工具栏内 **开始示波器**按钮可以开始示波器跟踪。
- **停止示波器跟踪**：单击工具栏内 **停止示波器**按钮可以停止示波器跟踪。
- **图像自适应**：单击工具栏内 **最适合大小**按钮可以进行图像自适应。
- **跟踪图像放大和缩小**：长按鼠标左键框选要放大的区域可以对框选区域放大；滚动鼠标滚轮可以对鼠标光标当前位置进行放大和缩小。
- **跟踪图像平移**：长按鼠标右键同时拖动鼠标进行跟踪图像平移。
- **跟踪图像量测**：
 - 显示光标：单击工具栏内 **量测线置中**按钮，在跟踪图像居中位置显示一根垂直线显示跟踪变量的当前值。
 - 垂直量测：单击工具栏内 **垂直量测线**按钮，跟踪图像中显示出一对水平线，拖动两根水平线至量测位置，二者差值在参数设定区| ΔY |中显示。
 - 水平量测：单击工具栏内 **水平量测线**按钮，跟踪图像中显示出一对垂直线，拖动两根垂直线至量测位置，二者差值在参数设定区| ΔX |中显示。
- **存储文件**：单击工具栏内 **储存文件**按钮，储存跟踪图像

- **开启文件**：单击工具栏内  开启文件按钮，可打开之前储存的跟踪图像
- **跟踪图像中显示和隐藏数据点**：勾选和取消勾选参数设定区标记数据点

● 数据示波器跟踪模式

数据示波器支持三种跟踪模式，分别为 Auto、Sync 及 Trigger 模式。可以在参数设定区模式中
选择。

➤ Auto 模式

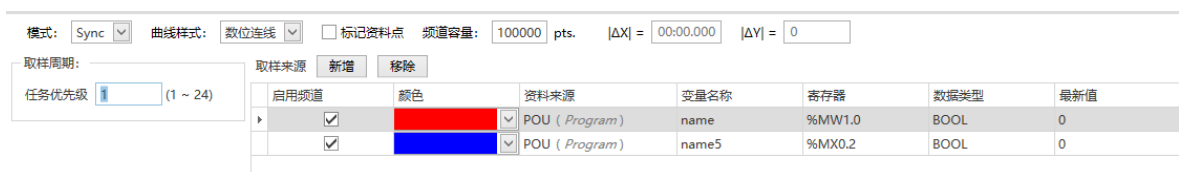
Auto 模式会依据计算机性能不确定周期地读取控制器中跟踪变量的值，跟踪图像中不能显示每一个周期的值，因此 Auto 模式仅可用于显示跟踪变量趋势时使用。Auto 模式支持同时跟踪 20 个变量。

➤ Sync 模式



Sync 模式根据数据示波器软件中选择的**任务优先级**来周期性读取控制器中跟踪变量值，因此 Sync 模式可以显示跟踪变量的每个周期的值。Sync 模式支持同时跟踪 4 个变量。

Sync 模式需设定参数如下图。**模式、曲线样式、标记数据点、频道容量、 $|\Delta X|$ 、 $|\Delta Y|$** 介绍请参考数据示波器参数设定区介绍。在此仅介绍 Sync 模式特定的参数。

任务优先级此任务优先级与 ISPSOft 软件中任务优先级一致。例如此参数设定为 1，则数据示波器的采样周期将与 ISPSOft 软件中任务优先级为 1 的任务的周期相同。



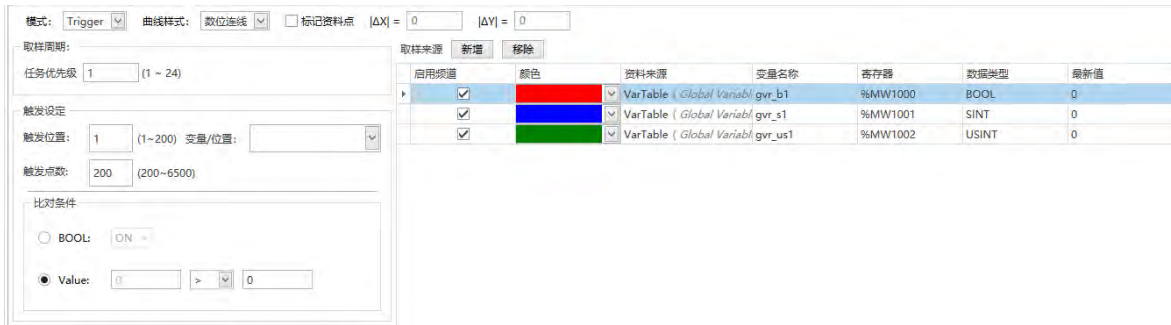
使用 Sync 模式进行跟踪步骤：

1. 确认**通信设置**内通讯状态为联机建立成功，若通讯状态不为联机建立成功，则请手动建立连接，手动建立连接方式请参考数据示波器操作通信设置。
2. 在参数设定区内设定参数：**模式**设定为 Sync，**曲线样式**设定为数字联机，**频道容量**设定为 100000，**任务优先级**设定为 ISPSOft 软件中使用的任务优先级。
3. 新增跟踪变量，新增跟踪变量方式请参考示波器操作新增跟踪变量。
4. 单击工具栏内  开始示波器按钮，开始跟踪变量。
5. 单击工具栏内  停止示波器按钮，停止跟踪变量，得到跟踪图像。

➤ Trigger 模式：

Trigger 模式需要选择一个跟踪变量作为触发变量并设定触发条件，当触发条件满足时，跟踪图像区会显示一定周期的所有跟踪变量的值，周期由**触发点数**确定。Trigger 模式支持同时跟踪 20 个变量。

Trigger 模式需设定参数如下图。**模式、曲线样式、标记数据点、频道容量、 $|\Delta X|$ 、 $|\Delta Y|$** 请参考数据示波器参数设定区介绍，**任务优先级**请参考 Sync 模式介绍。在此仅介绍 Trigger 模式特定的参数。



单击**变量/位置**右侧的下拉按钮，弹出所有已新增的跟踪变量，可以选择其中一个跟踪变量作为触发变量。

23



➤ 触发点数

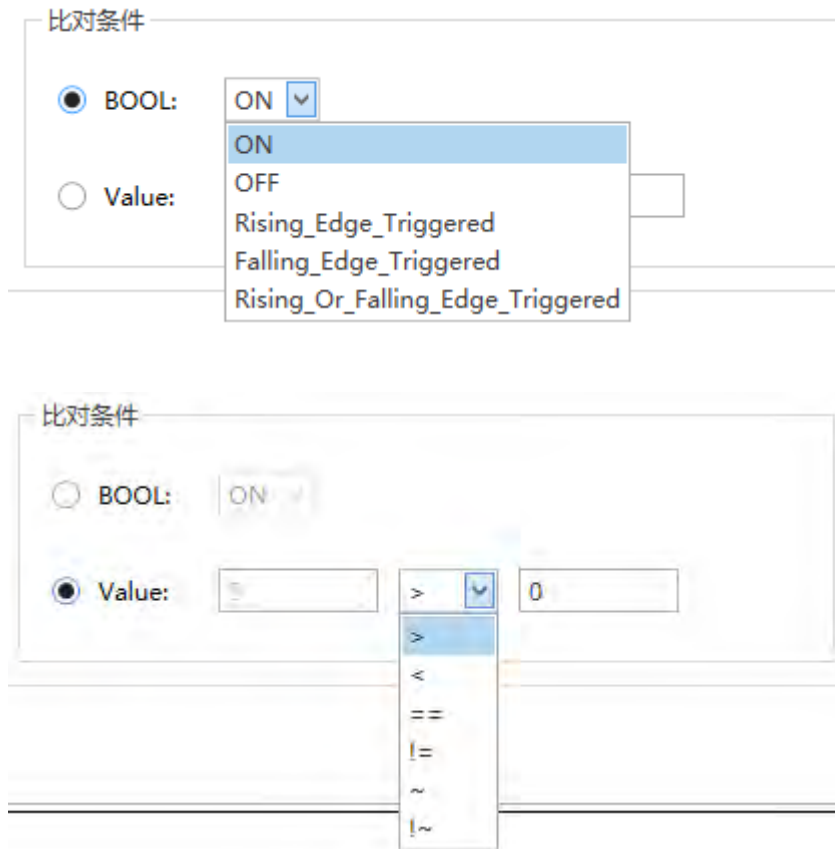
触发点数决定了当触发条件满足时，跟踪图像区会显示跟踪变量的周期数，触发点数设定越大，触发条件满足时获取到跟踪变量的周期数越多。触发点数范围为 200~6500，默认值为 200。

➤ 触发位置

触发位置决定了触发条件第一次满足时的位置。触发位置范围为 1~200，默认值为 1。若想要获取触发条件满足时的前 100 个周期跟踪变量的值，需要把**触发位置**设定为 101。

➤ 比对条件

根据触发变量的类型，比对条件有两种，分别为 **BOOL** 和 **Value**。**BOOL** 比对方式有 ON、OFF、上升沿、下降沿、上升或下降沿；**Value** 比对条件有大于、小于、等于、不等于、处于范围、范围之外。如下图所示。



23

注：

1. DVP50MC 初体 1.11.31 及以上支持数据示波器功能。
2. 若在跟踪时修改项目程序，需要重新新增所有跟踪变量。

23.3 数据记录器

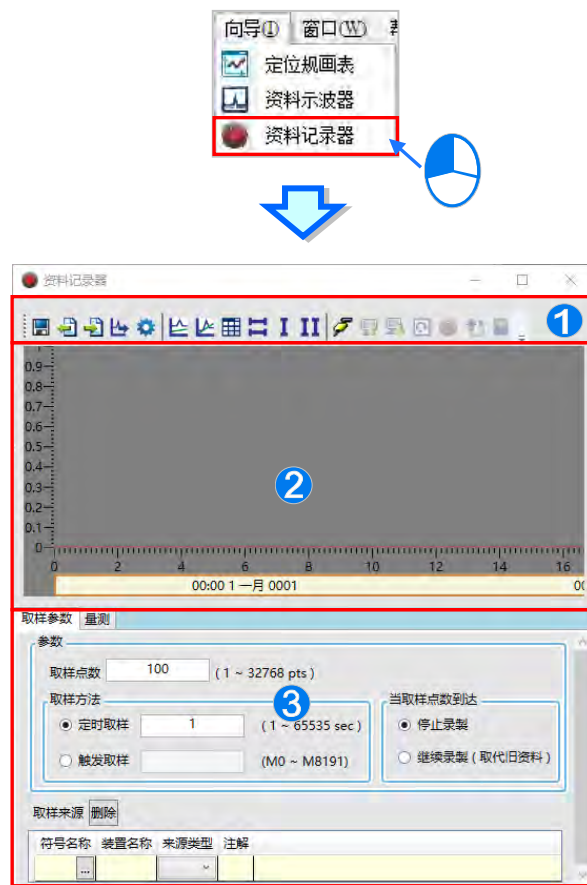
23.3.1 关于数据记录器

数据记录器用于长期记录，变量符号或装置的数值与状态，以波形图显示，并且可保存于主机内部与存储卡中，方便分析数值趋势。

23.3.2 开启数据记录器

要使用此功能时，请先编译目前项目后，再单击 **向导 (I)** > **数据记录器**，开启数据记录器窗口，如下图。

23





- ① 图示工具栏：提供各项操作数据记录器的按钮，功能说明如下表。
- ② 波形显示区：实时显示变量符号或装置的波形图。
- ③ 参数设定区：取样参数分页用于设定取样的来源与参数；量测分页会显示量测工具所量测到的数值及调整波形的颜色与比例等。

图标工具栏按钮功能说明

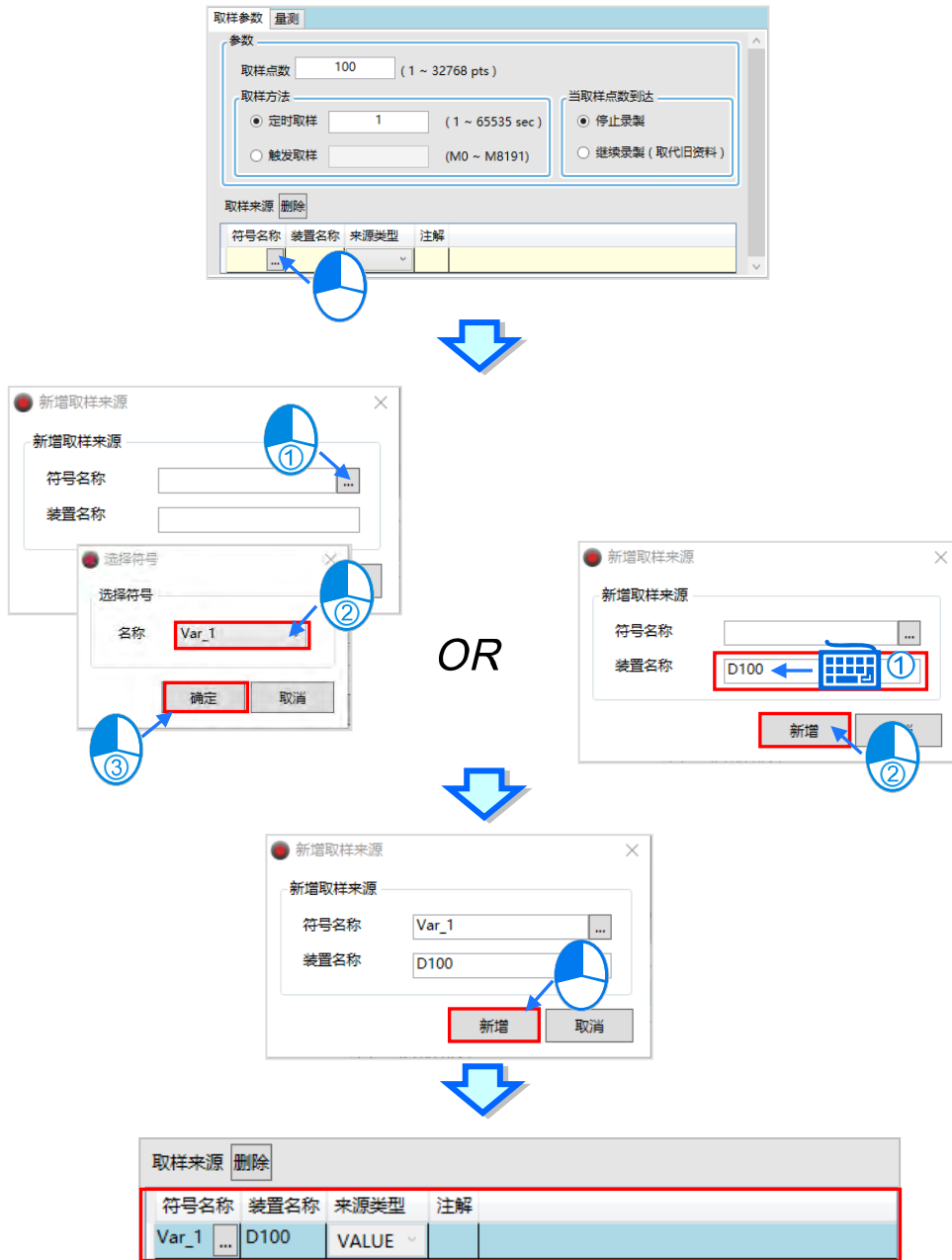
图示	名称	功能说明
	保存全部	将取样设定数据与取样波形数据结果保存到指定路径中。
	导出取样设定	将取样设定内容导出到指定路径。
	导入取样设定	由指定路径将取样设定内容导入。
	导出波形数据	将取样波形数据以 .bmp 或 .csv 格式导出到指定路径。
	设定	设定数据记录器波形与环境的颜色。
	分离波形	将多条波形自动以间隔分离。
	重迭波形	将多条波形以中心线重迭。
	显示网格线	切换是否显示网格线。
	水平量测线	波形显示区将出现两条水平线，供用户量测波形。
	垂直量测线	波形显示区将出现一条垂直线，供用户量测波形。
	双垂直量测线	波形显示区将出现两条垂直线，供用户量测波形。
	联机/脱机	切换计算机与主机的联机或脱机模式。
	上载 PLC 取样设定	从主机读回数据记录器的取样设定。
	下载取样设定到 PLC	将数据记录器的取样设定写入主机。
	实时监看	计算机实时从主机读取来源的取样波形。
	录制数据	按下后开始录制，再次按下后停止录制。
	上载 PLC 纪录	由主机读回数据记录器的纪录值
	写入存储卡	将主机的纪录数值写入到存储卡中

23

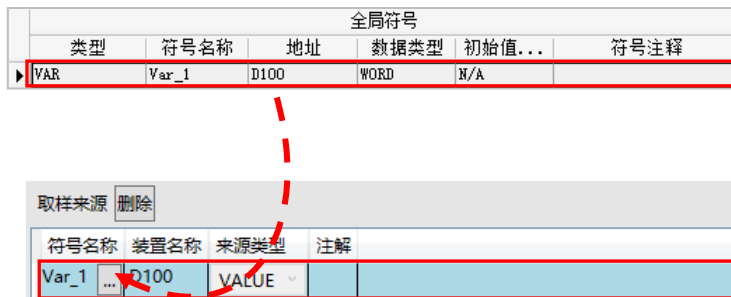
23.3.3 设定取样参数

在取样参数分页中，**取样点数**用于设定取样次数；**取样方法**可以分成**定时取样**与**触发取样**两种方式，当使用**定时取样**时，须设定取样周期，则每隔一周期取样一次；当使用**触发取样**时，须设定触发装置，当触发装置 ON 时，取样一次，并且在取样完成后，自动将触发装置 OFF，当取样点数次数到达时，可以设定**停止录制**或**继续录制（取代旧数据）**。要新增取样符号，先单击符号名称的  按钮，接着在弹出的窗口单击  按钮，选择要加入的变量符号，或直接在**装置名称**字段输入设备名称，按下「新增」后，就会加入取样来源列表中，如下图。

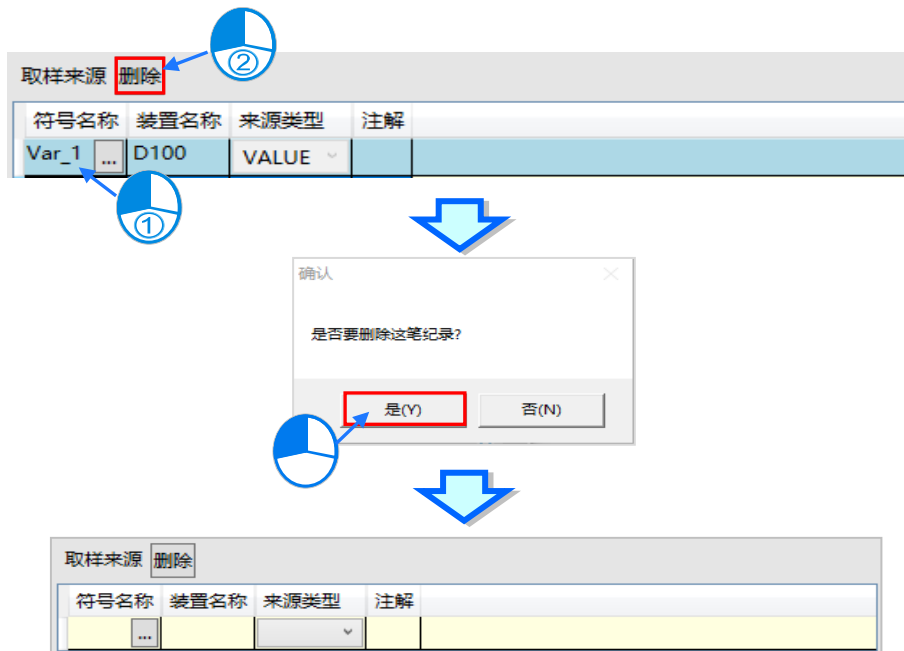
23





只有宣告在全局符号表的变量，并且编译过项目，才可以在这边被选取，若有手动设定装置地址或符号批注，也会一并带入。




要删除已加入的取样来源时，单击要删除的项目后，按下「删除」按钮，即可把该项目删除。

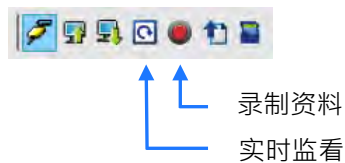


23

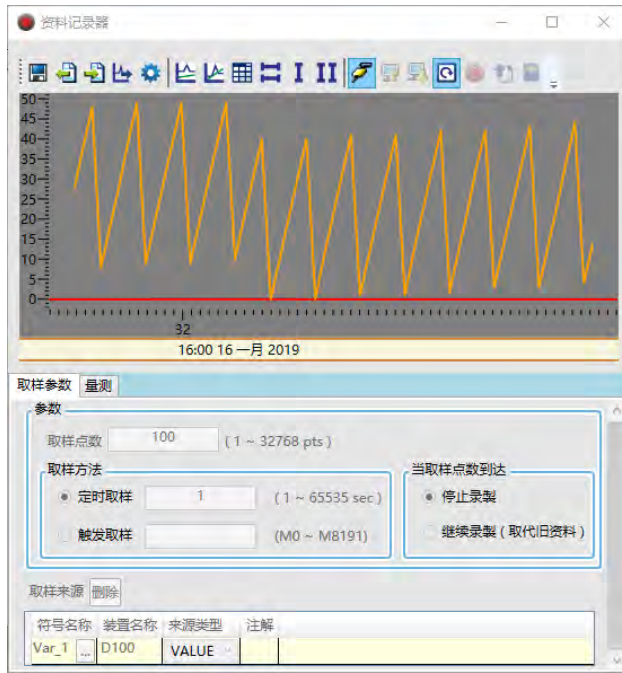
完成取样参数的设定后，可按下图示工具栏的「导出」，将取样参数设定保存成.dlg 档，之后可以在新项目使用「导入」将取样参数设定还原。

23.3.4 监看与录制模式

完成数据记录器的取样参数设定后，在联机模式时，先下载参数设定，便可以使用图标工具栏的按钮切换实时监看或录制数据。

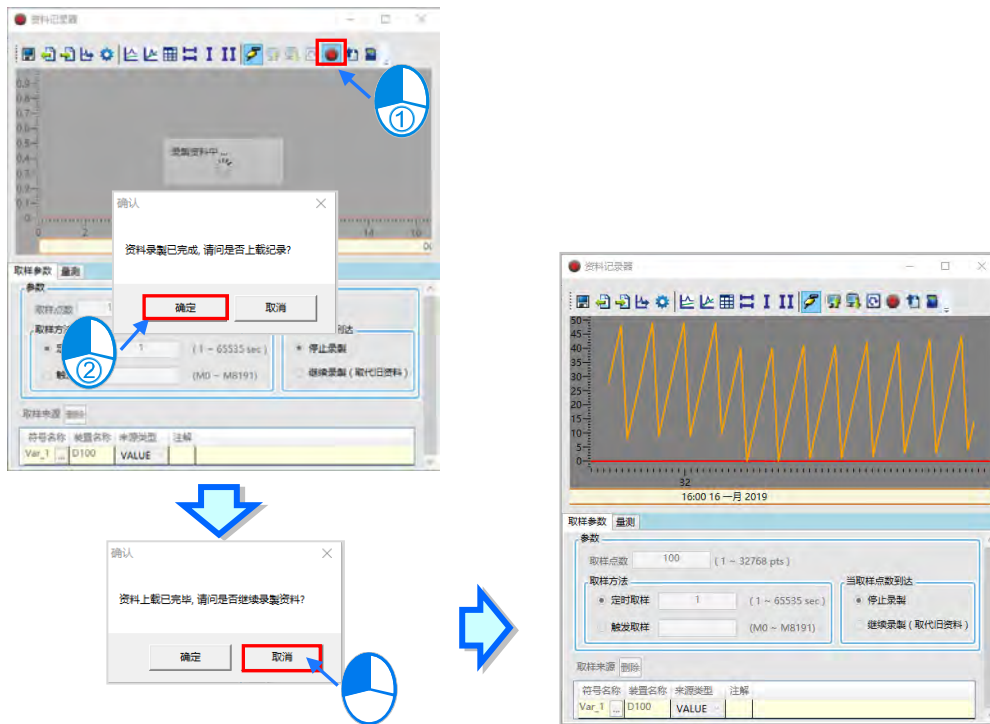


- 实时监看：依据设定的取样周期，将取样来源的变量符号或装置的数值或状态不断绘制到数据记录器上。画面会随波型自动调整到最适范围，只保留最新的取样点数个的数据。主机在此模式不会记录取样来源。



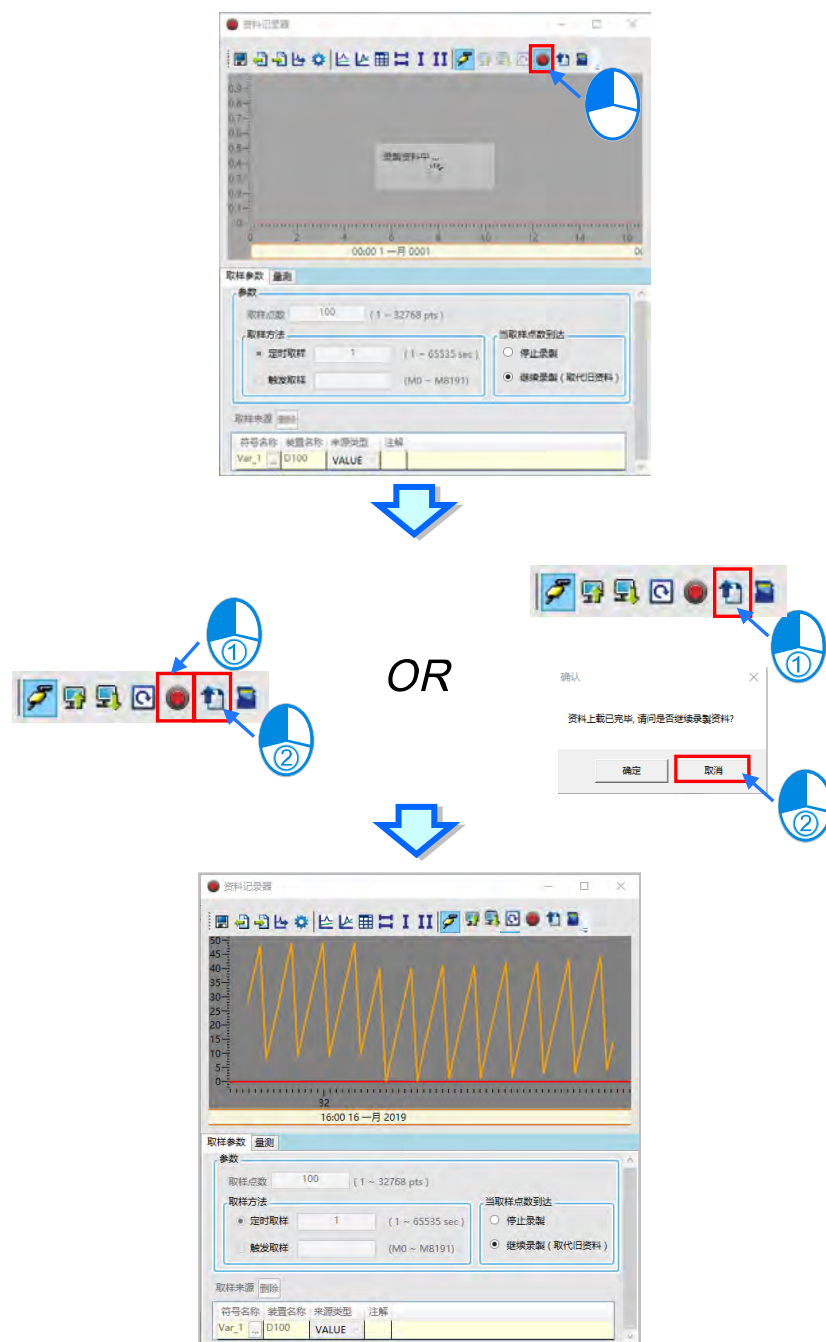
23

- 录制数据 :录制取样来源的变量符号或装置数值与状态 ,并保存在主机的内部存储器中。按下「**录制数据**」开始录制。使用取样点数到达**停止录制** ,当设定的取样点数到达后 ,便会弹出确认窗口 ,询问是否上载录制记录 ,此时单击**确定** ,接着会弹出确认窗口 ,确认是否再次录制 ,单击**取消**后便会将上载的数据显示在波形显示区中。



若使用取样点数到达**继续录制 (取代旧数据)**时 ,主机便会不断录制取样源数据 ,并保留最新的取样点数范围的数据 ,可以按下「**录制数据**」,待停止录制后再按下「**上载 PLC 记录**」,或是在不停止录

制时直接按下「**上载 PLC 记录**」接着会弹出确认窗口，确认是否再次录制，单击取消后，便会将主机录制的数据显示在波形显示区中。



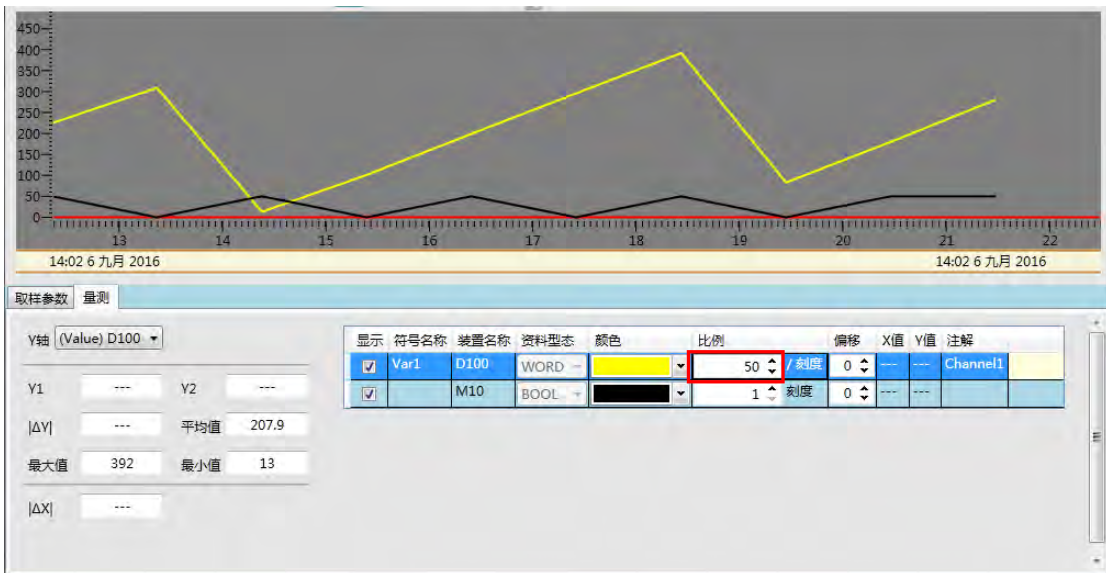
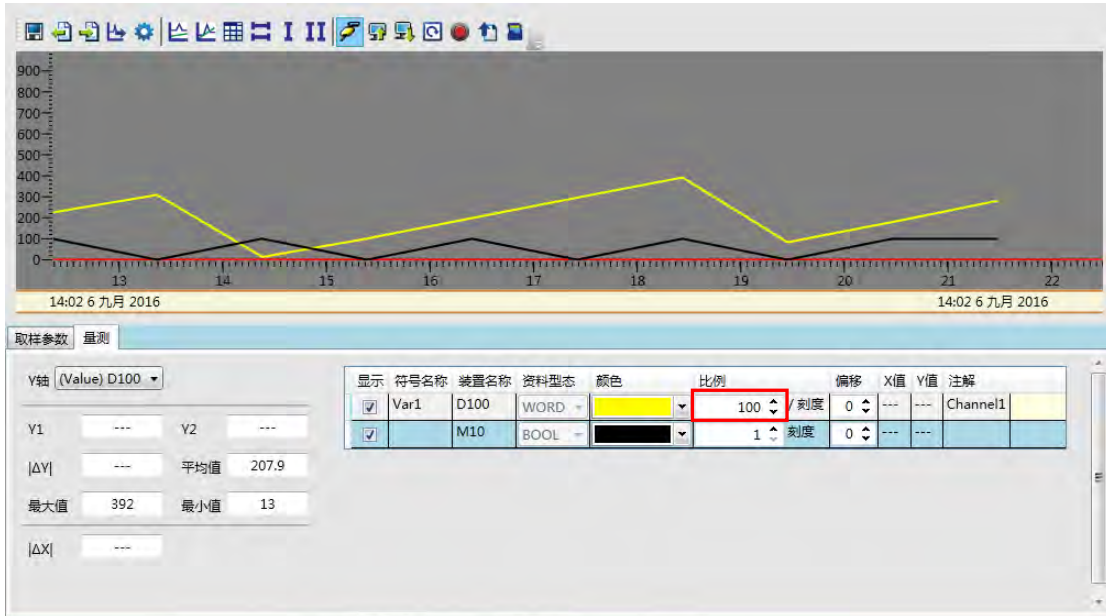
23

不论是否正在录制数据，用户可以按下「**写入存储卡**」，将主机目前的录制数据写入到存储卡根目录\PLC CARD\机种\Log\DATA_LOGGER_年月日_时分秒。

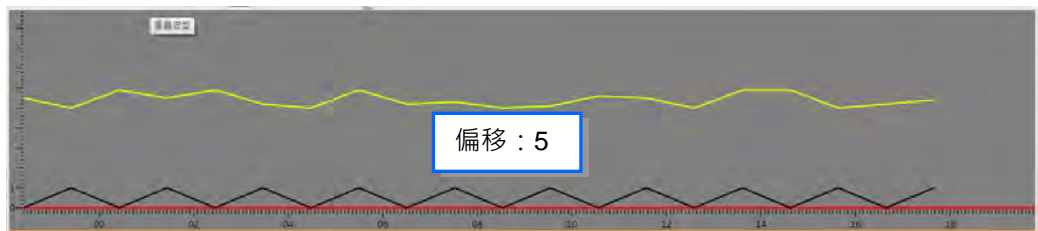
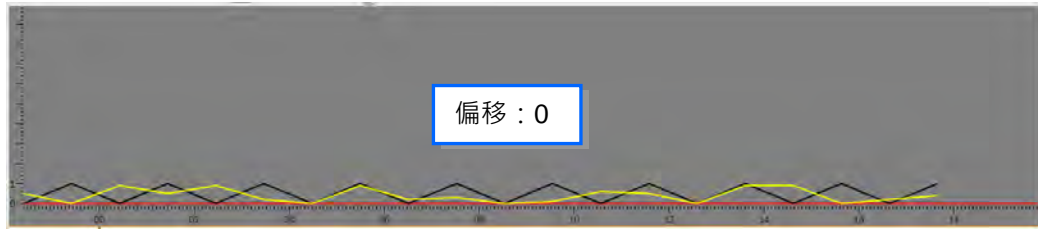
23.3.5 量测

在量测页面的平均值、最大值、最小值为 Y 轴所选择装置波形的信息。每个波形也可以设定不同的颜色、比例与偏移。比例为纵坐标的比例，下图为比例设为每刻度 100 与 50 的比较。



23



设定偏移，则可调整波形在画面中的位置，如下图。



23

切换到量测页面，搭配水平量测线与垂直量测线，可以量测数据记录器的波形数值。如下图，单击  按钮开启水平量测线，单击  按钮开启垂直量测线。

显示	符号名称	装置名称	资料型态	颜色	比例	偏移	X值	Y值	注解
<input checked="" type="checkbox"/>	Var1	D100	WORD	Yellow	200 / 刻度	0	2016/09/06 20:36:04	638	
<input checked="" type="checkbox"/>		M10	BOOL	Black	1 / 刻度	0	2016/09/06 20:36:04	0	

Y轴 (Value) D100

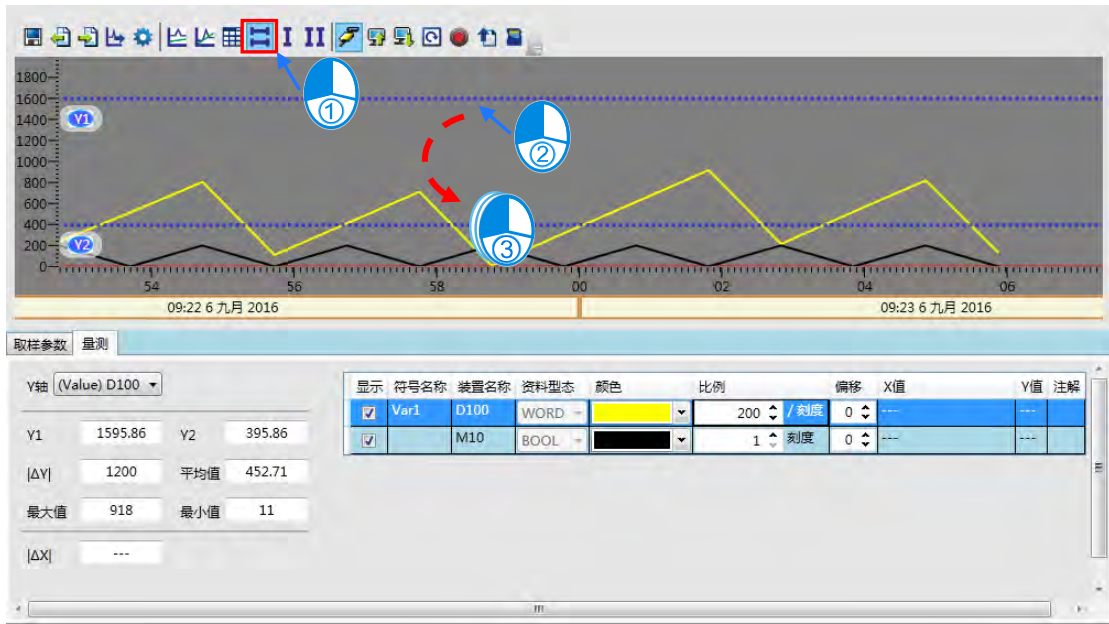
V1: 1574.89 Y2: 374.89

|ΔY|: 1200 平均值: 483.92

最大值: 909 最小值: 17

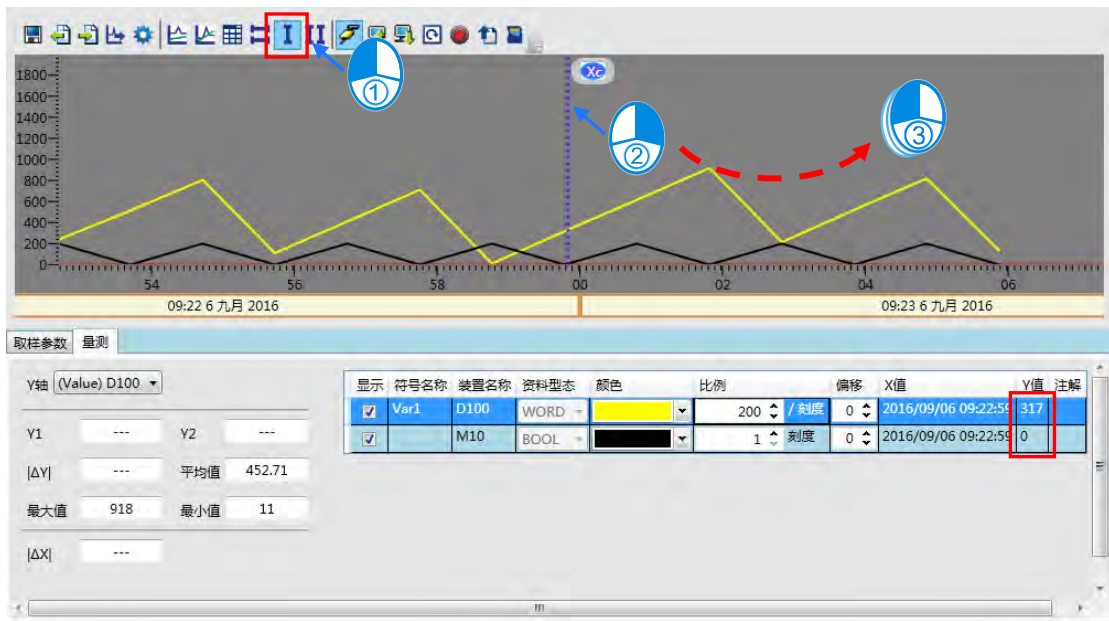
|ΔX|: ---

使用水平量测线，可以拖曳两条水平量测线 Y1 与 Y2，调整到要测量的位置。于参数设定区便会显示水平量测线的各项数值，Y1 字段代 Y1 的数值，Y2 字段代表 Y2 的数值，|ΔY| 字段代表 Y1 与 Y2 的差值。

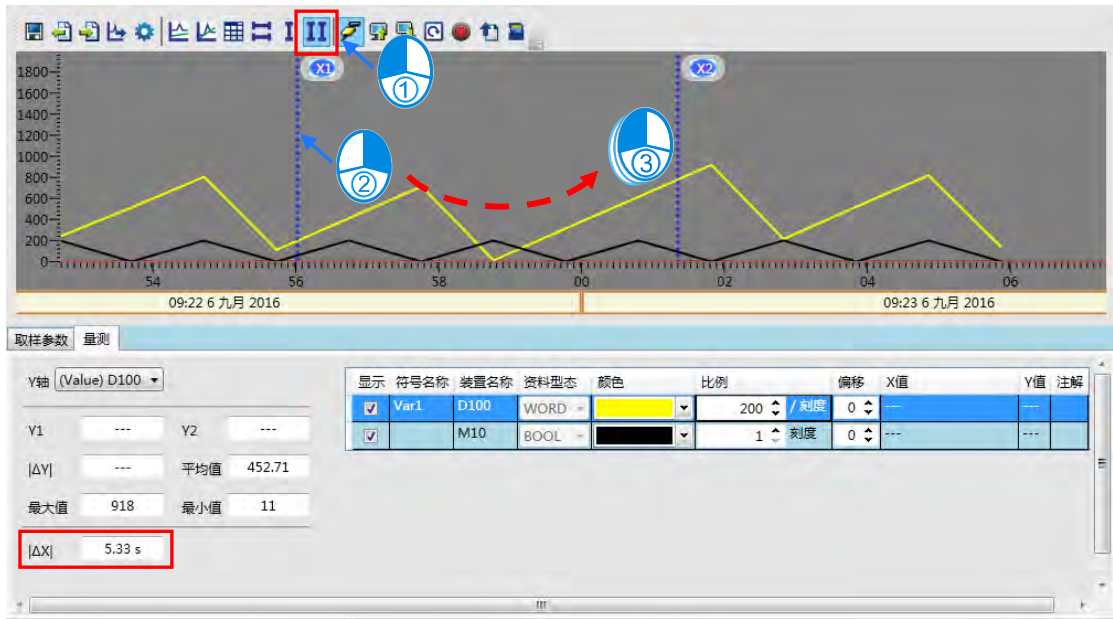


23

使用垂直量测线，可以拖曳垂直量测线 Xc，则下方的 Y 值栏位，则会显示各取样来源与 Xc 交叉所对应的 Y 值。如下图。



使用垂直量测线，可以量测数据示波器的时间值。如下图，单击 **II** 按钮开启双垂直量测线，以鼠标单击后拖曳两条垂直量测线 X1 与 X2，调整到要测量的位置，于下方的区域便会显示 ΔX 的时间值。



23

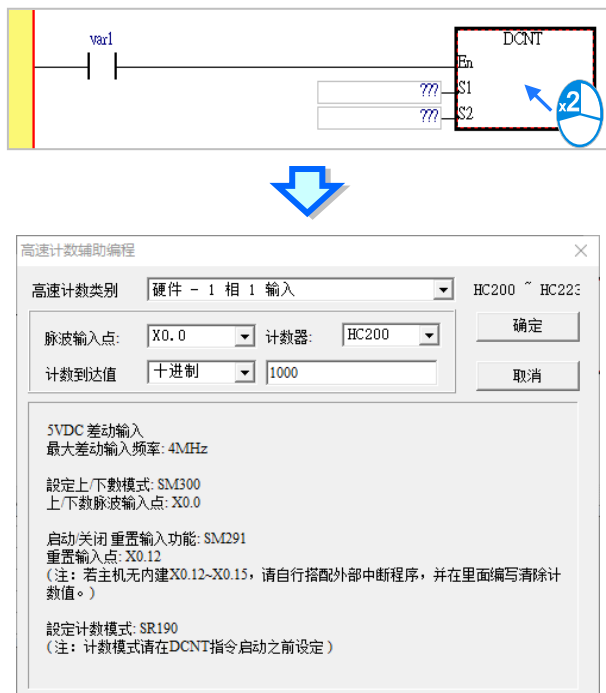
23.4 高速计数器向导

23.4.1 关于高速计数器向导

高速计数器向导提供用户依据高速计数器的类型、计数的形式、输入点编号选择对应的高速计数器装置编号，便于用户快速的设定 DCNT 指令。

23.4.2 使用高速计数器向导

于梯形图中的 DCNT 指令上，双击鼠标左键，开启高速计数器向导设定窗口。如下图。



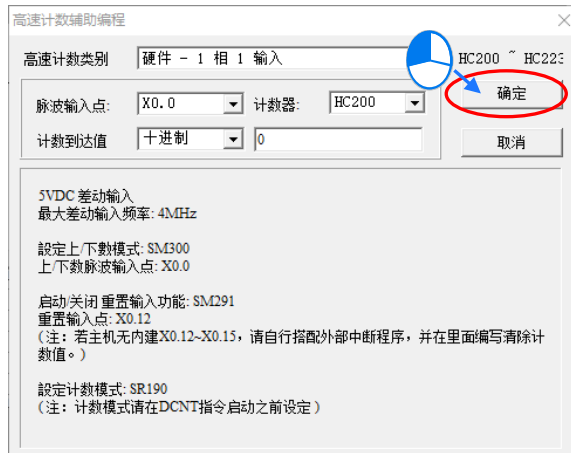
先选择高速计数器的类别，包括**硬件 1 相 1 输入**、**硬件 1 相 2 输入**...等等，接着选取脉冲输入点的位置跟计数器的装置。如下图。



再来要输入计数的数值。可以选取十进制、十六进制或使用 D 装置，接着于右方字段输入数值或 D 装置编号。



最后按下「确定」，高速计数器编号与计数器设定值将会被填入 DCNT 指令中。



23.5 NTC 向导

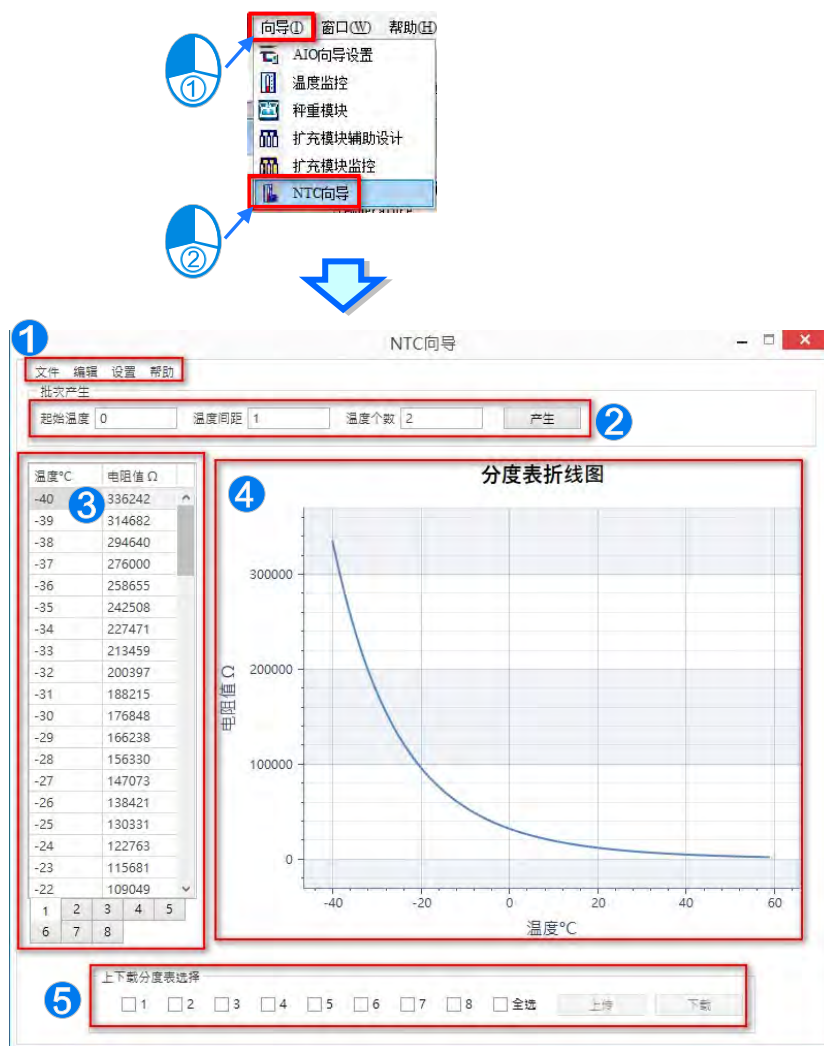
23.5.1 关于 NTC 向导

NTC 向导为 DVP08NTC-S 模块的专用向导软件，提供用户编辑自定义分度表、csv 文件导入/导出、一次上载/下载多张表、一次下载多台模块等功能，详细操作方式可参考该模块的安装手册。

23.5.2 使用 NTC 向导

要使用此功能时，请单击 向导 (!) > NTC 向导，打开 NTC 向导窗口，如下图。

23



- ❶ 选项包含 文件：文件保存、读取，分度表导入、导出功能。
- 编辑：加载默认值、清除分度表。
- 设定：通信设置，RS-485 直连模块或是通过主机连接模块。
- 说明：操作说明、软件信息说明。

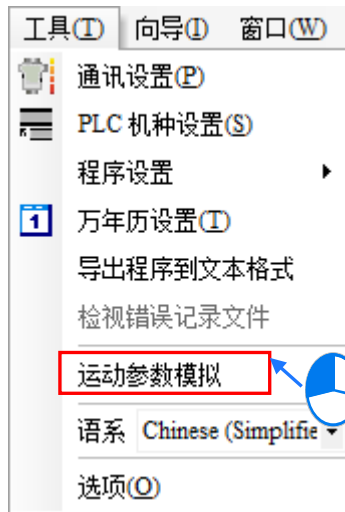
- ② 输入起始温度、温度间距、温度个数，可以产生一张空白的表格。
- ③ 分度表表格，可以填写对应的电阻值，共八张分度表可供填写，使用下方的页签切换。
- ④ 分度表折线图，依照左边的表格产生出来的折线图。
- ⑤ 上下载分度表选择，选择欲上载/下载的分度表。

23.6 运动参数模拟

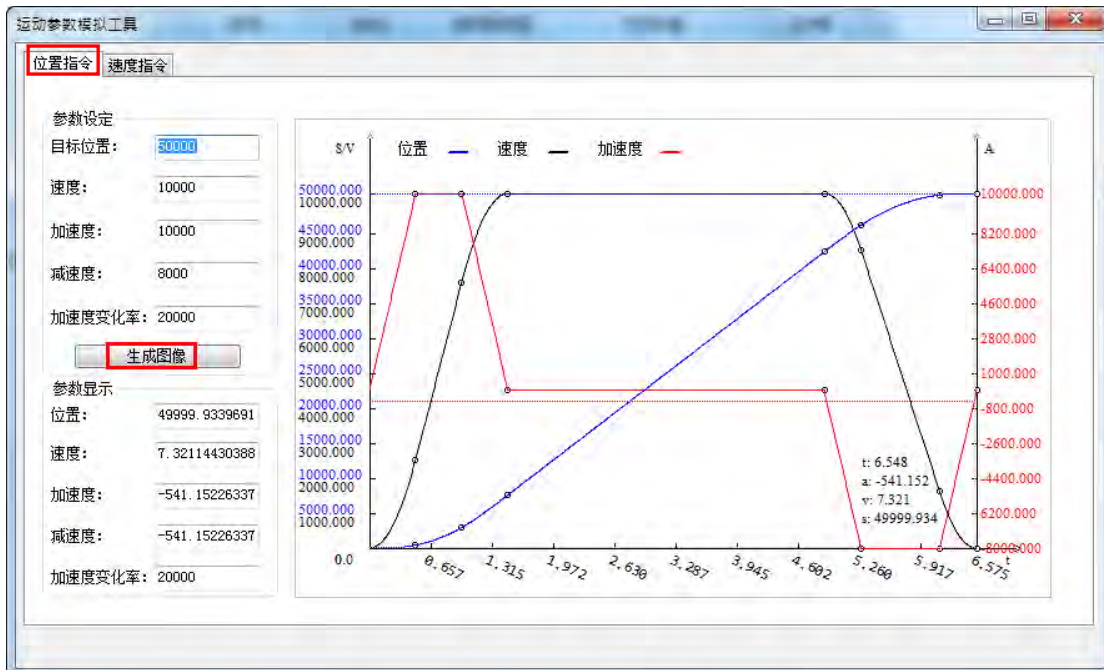
23.6.1 关于运动参数模拟

在工具栏单击工具 (**T**) > 运动参数模拟，用于输入相关参数后，显示对应的位置、速度和加速度曲线，方便用户在不知道运动指令输入参数如何填写时使用。

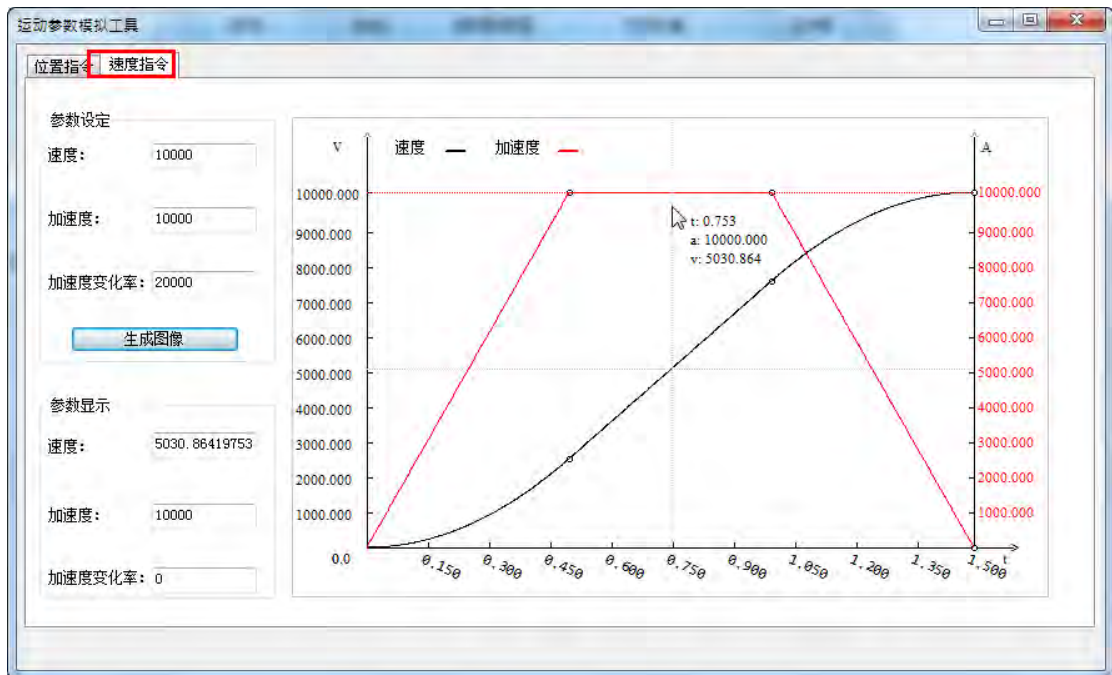
23



选择位置指令时，单击下图红色方框处生成图像按钮，右侧显示对应的位置、速度和加速度曲线。参数设定部分用于填写设定参数，参数显示用于显示右侧图像鼠标所在位置的位置、速度和加速度。



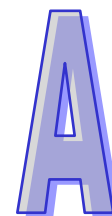
选择**速度指令**时，单击下图红色方框处**生成图像**按钮，右侧显示对应的速度和加速度曲线。参数设定部分用于填写设定参数，**参数显示**用于显示右侧图像鼠标所在的速度和加速度。



23

MEMO

23



附录A USB 联机

目录

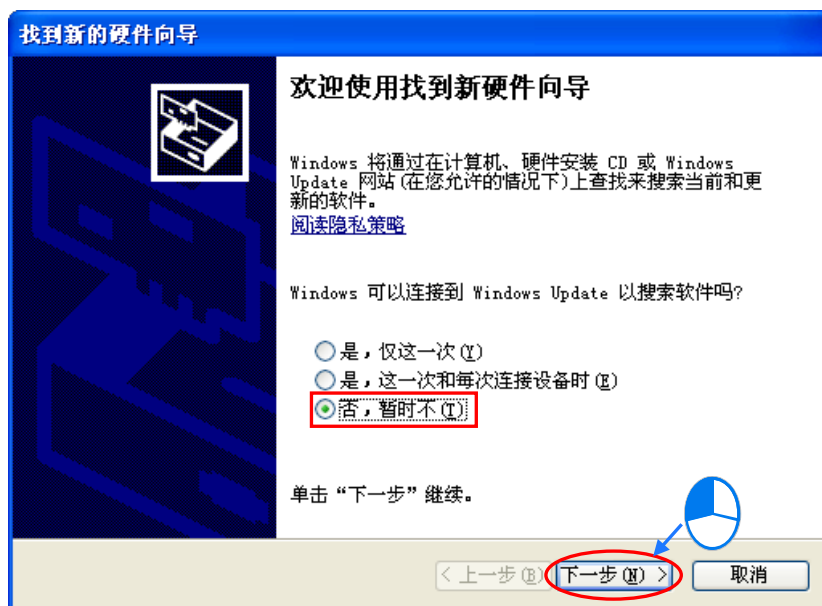
A.1	安装 PLC 的 USB 驱动程序	A-2
A.1.1	在 Windows XP SP3 环境下安装 USB 驱动	A-2
A.1.2	在 Windows7 环境下安装 USB 驱动	A-7
A.1.3	在 Windows8.1 环境下安装 USB 驱动	A-11
A.1.4	在 Windows10 环境下安装 USB 驱动	A-15
A.2	COMMGR 中建立 USB 类型的 Driver	A-18
A.3	DVP-SX2 机种的 USB 设定	A-20

A.1 安装 PLC 的 USB 驱动程序

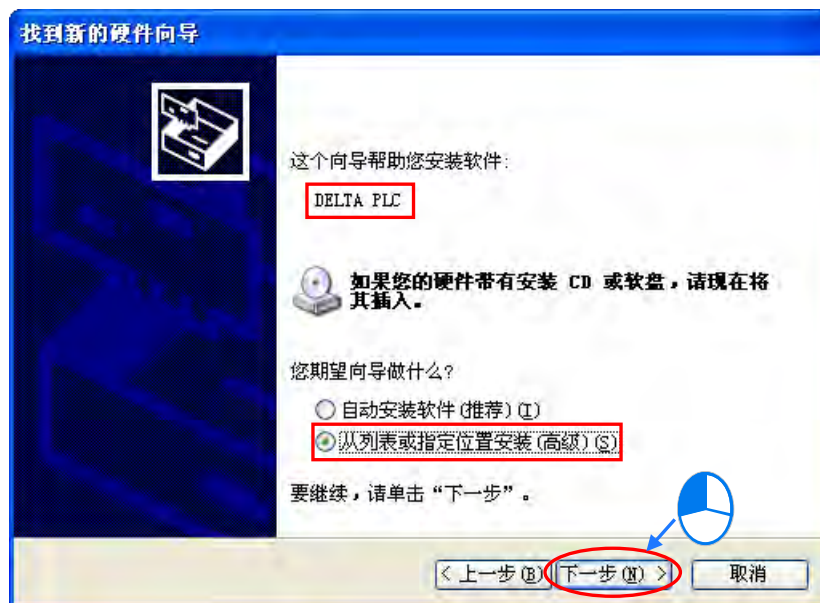
A.1.1 在 Windows XP SP3 环境下安装 USB 驱动

下列将以 Windows XP 为例，逐步示范如何安装 PLC 的 USB 驱动程序，至于其它操作系统则请自行参考该操作系统中有有关新硬件安装的相关说明。

- (1) 确认 PLC 主机已正常上电，并通过 USB 线缆将 PLC 主机连接至计算机的 USB 端口，此时画面会出现新增硬件的提示窗口，选择「不，暂时不 (N)」的选项之后单击「下一步 (N)」。



- (2) 接着窗口中将会显示侦测到的 USB 设备名称，下图为 AH500 机种的设备名称，而其它机种所显示的名称可能会有所不同。此步骤请选择「从列表或指定位置安装 (高级) (S)」。

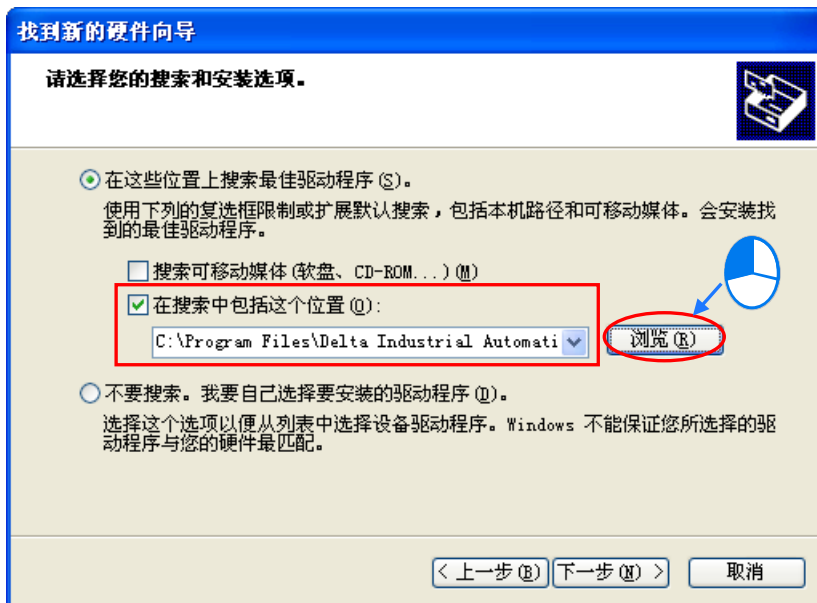


- (3) 下表为台达目前内建 USB 通讯接口的 PLC 机种，而在安装完 V2.00 版本以上的 ISPSOft 之后，各机种的驱动程序亦会被放置在下表的路径中。

机种	驱动程序路径
SX2、AH560EN2	ISPSOft 安装路径\drivers\SX2_USB_Driver\
SE	ISPSOft 安装路径\drivers\Delta_PLC_USB_Driver\
AH500 系列	ISPSOft 安装路径\drivers\Delta_PLC_USB_Driver\

*.ISPSOft 的预设安装路径为 C:\Program Files(x86)\Delta Industrial Automation\ISPSOftx.xx。

接着请根据所连接的 PLC 机种指定正确的驱动程序路径，但若通过其它方式所取得的驱动程序，则请自行指定相对的路径。完成后单击「下一步 (N)」以继续进行安装步骤。



- (4) 当于指定的路径中搜寻到正确的驱动程序之后，系统便会开始进行安装，而安装途中若出现未通过验证测试的警告窗口时，请直接按下「仍然继续 (C)」即可。



- (5) 安装结束后, 请按下「完成」键离开。



- (6) 完成安装后, 请先打开 Windows 的设备管理器, 当端口 (COM 和 LPT) 的目录中有列出方才所安装的设备名称时, 便代表驱动程序已安装成功, 而此时 Windows 系统便会自动为其配置一个 COM Port 的编号。



*.上图为 AH500 机种的设备名称，而其它机种所显示的设备名称可能会有所不同。

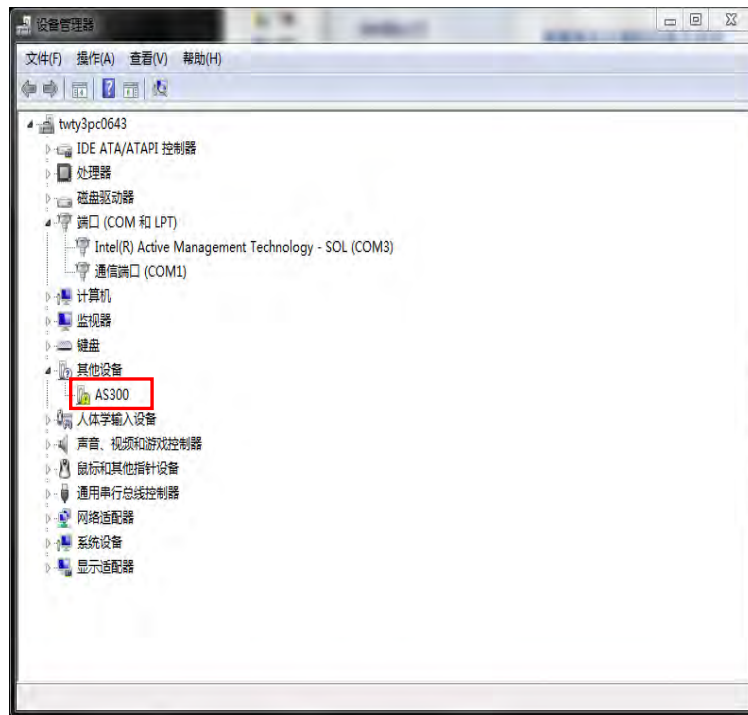
补充说明

- 当更换计算机端的 USB 端口时，系统可能会要求重新安装驱动程序，此时仅需依据前述步骤再次安装驱动程序即可。而安装完毕后，系统为其配置的 COM Port 编号亦可能会有所不同。
- 若您的 Windows XP 尚未更新至 SP3 的版本时，在安装过程中可能会出现缺少必要文件的警示讯息，此时可采取如下的两个处置方式。
 - (a) 取消安装并更新 Windows XP 至 SP3 版，之后再依据前述的步骤重新安装驱动程序。
 - (b) 自行取得所需的文件后，于出现的警示窗口中指定该文件的路径。

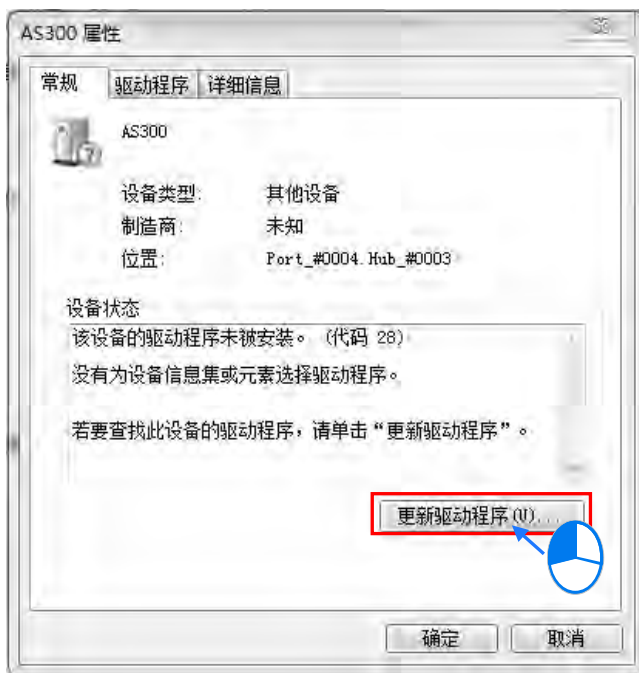
A.1.2 在 Windows7 环境下安装 USB 驱动

若用户使用 Windows 7 操作系统，接下来将举 AS300 系列逐步示范如何安装 PLC 的 USB 驱动程序。

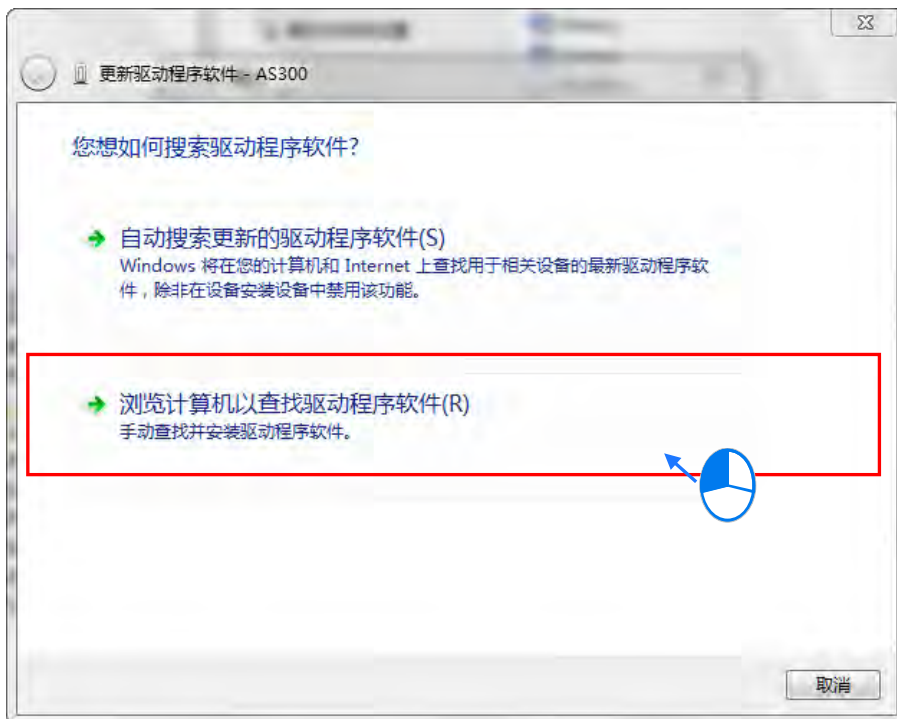
- (1) 确认 AS 主机已正常上电，并通过 USB 电缆将 AS 主机连接至个人计算机的 USB 端口。
- (2) 单击 控制台 > 设备管理器 后，双击「其他设备」选项下的「AS300」。



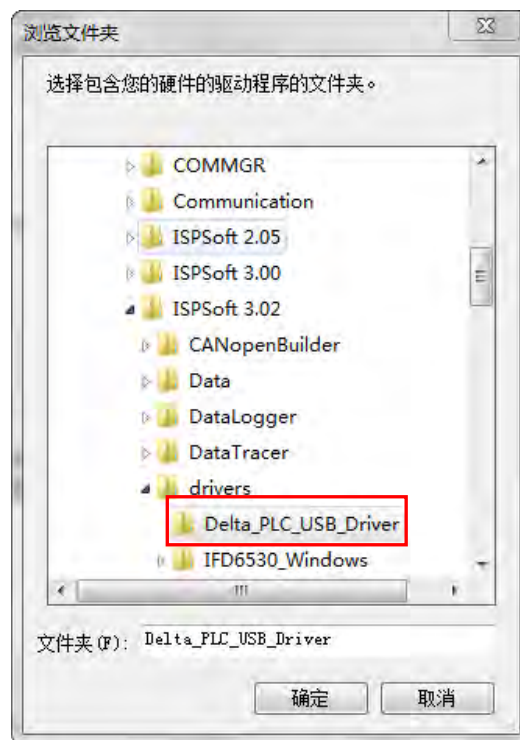
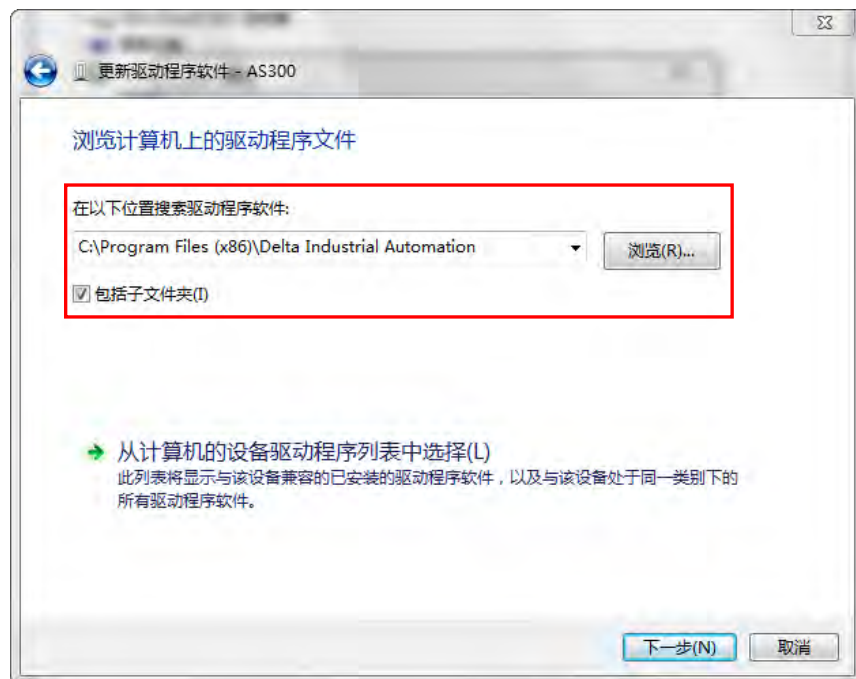
- (3) 在以下窗口中，单击「更新驱动程序」按钮，如下所示。



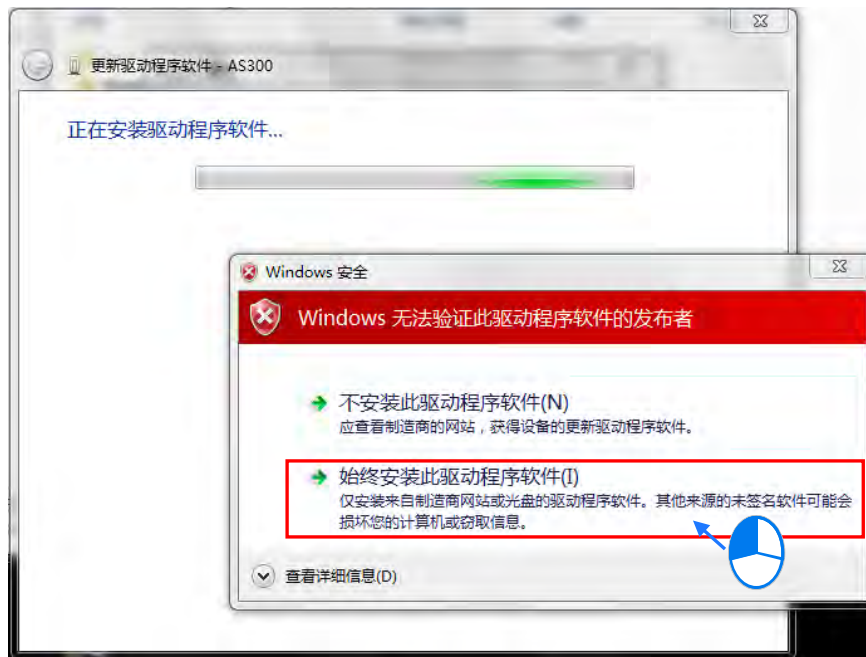
(4) 接着窗口中，请选择「浏览计算机上的驱动程序软件 (R)」。



(5) 在安装完 V3.00 版本以上的 ISPSOft 之后，AS 主机的 USB 驱动程序会被放置在 ISPSOft 安装路径\drivers\Delta_PLC_USB_Driver\，若通过其他方式取得的驱动程序，请自行指定相关路径。接着请指定正确的驱动程序路径，但若通过其他方式所取得的驱动程序，则请自行指定相对的路径。完成后单击「下一步 (N)」以继续进行安装步骤。

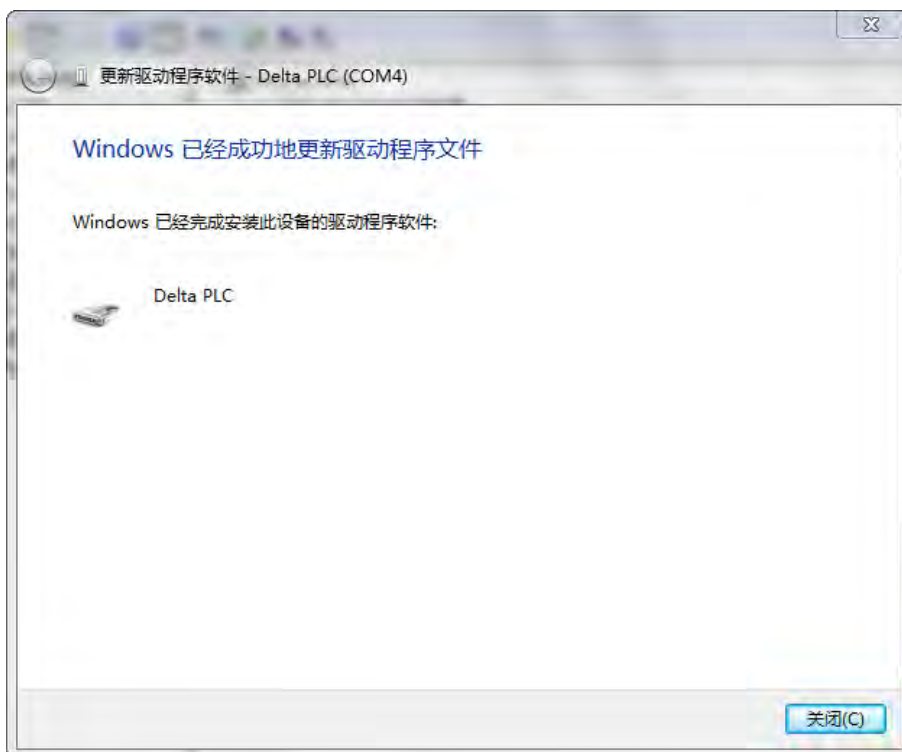


- (6) 当于指定的路径中搜寻到正确的驱动程序之后，系统便会开始进行安装，而安装途中若出现要求验证测试的警告窗口时，直接按下「仍然安装此驱动程序软件 (I)」即可。

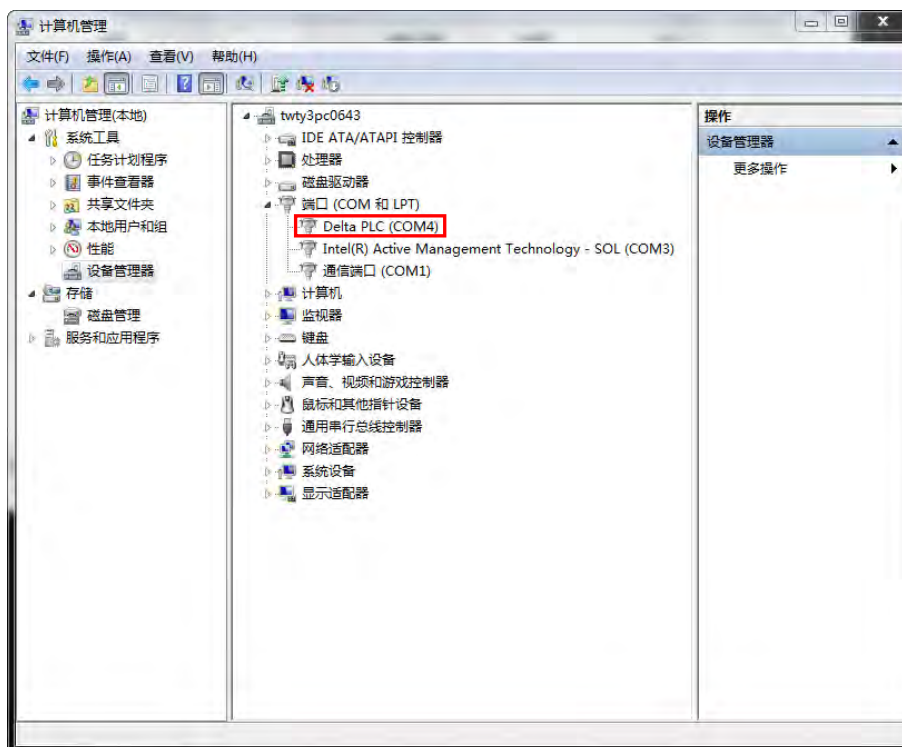


(7) 安装结束后，请按下「关闭」键离开。

A



(8) 完成安装后，在 Windows 的设备管理器中的端口 (COM 与 LPT) 项目中，有列出方才所安装的装置名称时，便代表驱动程序已安装成功，而此时 Windows 系统便会自动为其配置一个 COM Port 的编号。




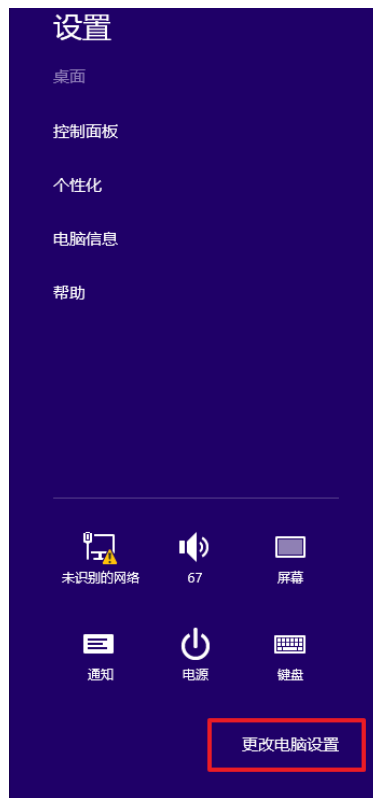
补充说明

- 当更换计算机端的 USB 端口时，系统可能会要求重新安装驱动程序，此时仅需依据前述步骤再次安装驱动程序即可。而安装完毕后，系统为其配置的 COM Port 编号亦可能会有所不同。

A.1.3 在 Windows 8.1 环境下安装 USB 驱动

若用户使用 Windows 8.1 操作系统，需先了解数字签名的内容，数字签名是一种可加入文件的电子安全性标记。它可以让您确认文件的发行者，而且可以协助确认文件自从数字签名之后并未变更。因目前 DELTA PLC USB 驱动程序无数字签名，接下来将逐步介绍如何关闭 Windows 8.1 操作系统的数字签名功能，来达成计算机端可顺利安装 DELTA PLC USB 驱动程序，此设定仅限单次使用，当计算机重新开机后，会回复原来的强制签章模式。

- (1) 按下键盘中  【WIN】+ 【I】，会显示设置接口，单击【变更电脑设置】。



- A
- (2) 在「电脑设置」画面左半部底下选择【更新和恢复】，在「更新和恢复」画面选择【恢复】，选完后选择【立即重启】。



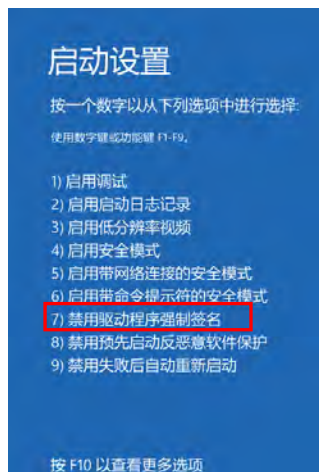
(3) 待重新启动后于选择选项画面中单击【疑难解答】；在疑难解答画面中单击【高级选项】。



(4) 高级选项画面中单击【启动设置】；在启动设置画面中单击【重启】。



(5) 系统会进入另一启动设置画面，按下数字键 7 或 F7 执行禁用驱动程序强制签名，完成后系统会重新回到 Windows 8 作业画面，此时即可安装 DELTA USB 驱动程序。



(6) 安装 DELTA USB 驱动程序可参考 Windows 7 的安装说明。

A.1.4 在 Windows10 环境下安装 USB 驱动

若用户使用 Windows 10 操作系统，需先了解数字签名的内容，数字签名是一种可加入文件的电子安全性标记。它可以让您确认文件的发行者，而且可以协助确认文件自从数字签名之后并未变更。因目前 DELTA PLC USB 驱动程序无数字签名，所以接下来将逐步介绍如何关闭 Windows 10 操作系统的数字签名功能，来达成电脑端可顺利安装 DELTA PLC USB 驱动程序，此设定仅限单次使用，当电脑重新开关机后，会回复原来的强制签章模式。

(1) 单击「开始」，再选择「设置」，然后按住「SHIFT」键并点击「重新启动」。



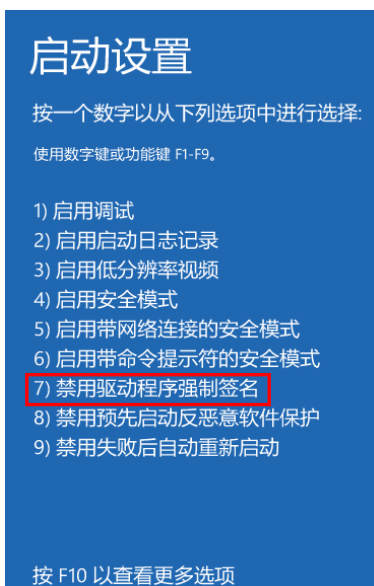
(2) 待重新启动后于选择选项画面中单击【疑难解答】；接着点选疑难解答画面中【高级选项】。



(3) 高级选项画面中单击【启动设置】，接着在启动设置画面中单击【重启】。



(4) 在启动设置画面，按下数字键 7 或 F7 执行禁用驱动程序强制签名，完成后系统会重新启动。



(5) 安装 DELTA USB 驱动程序，使用鼠标右键点击「开始」，接着点选「设备管理器」。

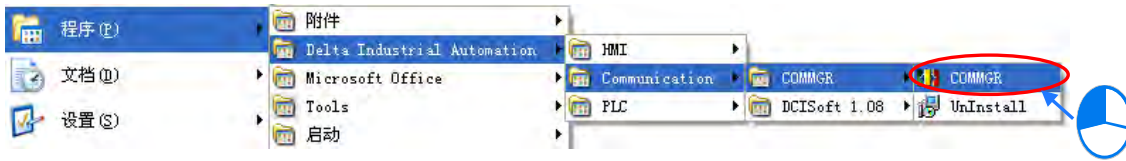


(6) 接着请参考 A1.2 在 Windows 7 环境下安装 USB 驱动程序，安装 Delta_PLC_USB_Driver。

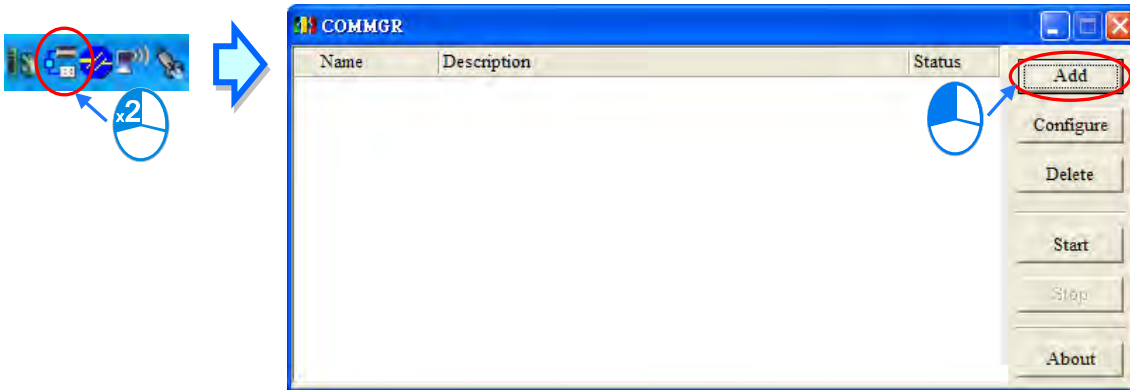
A.2 COMMGR 中建立 USB 类型的 Driver

COMMGR 中建立 USB 类型 Driver 的步骤如下。

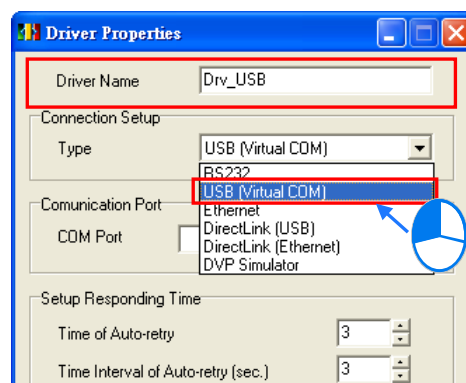
- (1) 确认 PLC 主机已正常上电，并通过 USB 线缆将 PLC 主机连接至计算机的 USB 端口。
- (2) 确定通讯管理员 COMMGR 已启动并常驻在 Windows 右下角；若否，则请至「程序」菜单中依下列路径至「Delta Industrial Automation」→「Communication」→「COMMGR」目录下单击「COMMGR」的执行快捷方式。



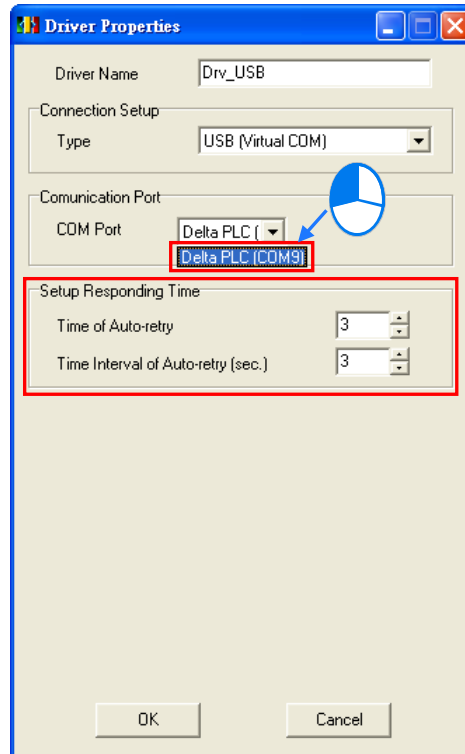
- (3) Windows 右下角的 COMMGR 图标上双击鼠标左键，之后再在 COMMGR 窗口的右侧单击「Add」键。



- (4) 首先在「Driver Name」中输入 Driver 的名称，之后在「Connection Setup」的「Type」字段中下拉选择 USB (Virtual COM)；不过当 PLC 机种为 SX2 时，联机类型则请选择 RS232，相关细节将会在下一节当中另行说明。

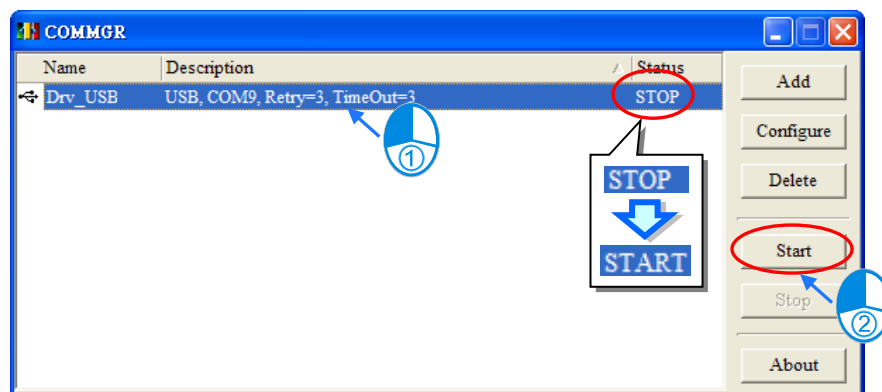


- (5) 接着请在「Communication Port」的「COM Port」字段下拉选择对应 PLC 主机的装置名称，之后在「Time of Auto-retry」设定联机错误时的重试次数，而「Time Interval of Auto-retry」参数则是设定重试的时间间隔。完成所有设定后按下「OK」键即可。



若在「COM Port」字段的下拉选单中没有对应的 PLC 选项时，请先确认 Windows 设备管理器的端口（COM 和 LPT）中，是否有对应该主机的设备项目，若否，则请确认 PLC 主机与计算机是否已通过 USB 线缆正确连接，并确认该主机的驱动程序是否已正确安装。确认无误后，请先关闭 COMMGR 的 Driver 设定窗口后再重新打开。

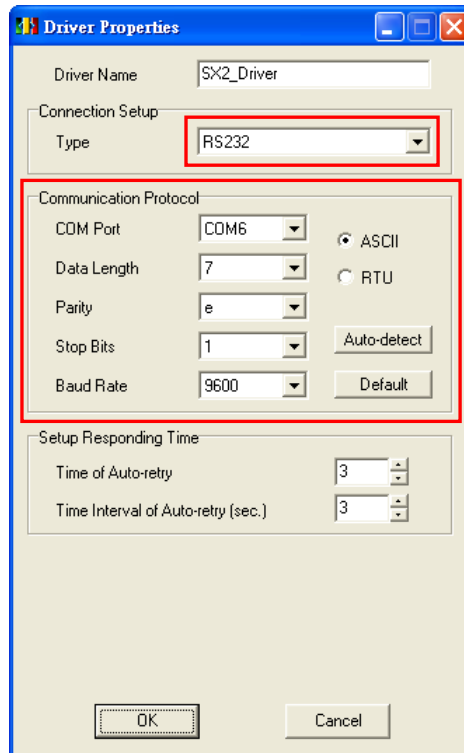
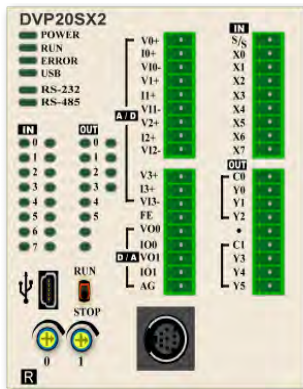
- (6) COMMGR 的管理列表中单击方才建立好的 Driver，接着于窗口的右侧单击「Start」键，待此 Driver 的 Status 字段变为「START」之后便已成功启动。



*.后续的测试以及更详细的 COMMGR 操作方式则请参考第 2.4 节的相关说明。

A.3 DVP-SX2 机种的 USB 设定

DVP-SX2 机种与其它机种的 USB 操作有些许不同，其架构是将 USB 转 RS232 的电路内建于主机中，因此虽然主机外部的接口是 USB 端口，但内部的运作仍是采用 RS232，所以在 COMMGR 中设定联机类型时也必须选择 RS232，且其设置的方式，包括「Auto-detect」功能，皆与一般的 RS232 相同。SX2 机种的 USB 端口可通过**特殊数据寄存器 D1109** 来设置其通讯参数，详细说明请参考该机种的编程手册。而于 COMMGR 中设定 Driver 时，请确认所设定的 RS232 参数是否与主机中的设置一致。





附录B 各系列机种的相关注意事项

目录

B.1	各系列机种的装置寻址.....	B-2
B.1.1	AH/AS 系列机种的装置种类	B-2
B.1.2	AH/AS 系列机种 X、Y、D、L 装置的寻址格式.....	B-3
B.1.3	AH/AS 系列机种 X/Y 接点状态的实时存取	B-4
B.1.4	DVPxxMC/AS5xx 机种的装置种类	B-5
B.1.5	DVPxxMC/AS5xx 机种装置的寻址格式.....	B-5
B.1.6	DVPxxMC/AS5xx 机种的停电保持装置.....	B-6
B.2	各系列机种的装置资源.....	B-6
B.2.1	AHCPU500-EN/AHCPU500-RS2 机种.....	B-6
B.2.2	AHCPU510-EN/AHCPU510-RS2 机种.....	B-7
B.2.3	AHCPU520-EN/AHCPU520-RS2 机种.....	B-7
B.2.4	AHCPU530-EN/AHCPU530-RS2 机种.....	B-8
B.2.5	AHCPU501-EN/AHCPU501-RS2 机种.....	B-8
B.2.6	AHCPU511-EN/AHCPU511-RS2 机种.....	B-9
B.2.7	AHCPU521-EN/AHCPU521-RS2 机种.....	B-9
B.2.8	AHCPU531-EN/AHCPU531-RS2 机种.....	B-10
B.2.9	AHCPU560-EN2 机种	B-10
B.2.10	AHxxEMC 系列机种	B-11
B.2.11	AS200/AS300 系列机种.....	B-11
B.2.12	DVPxxMC/AS5xx 系列机种	B-12
B.3	编译与上下载时间	B-12

AH/AS 系列机种的系统架构与 DVP 系列机种及 DVPxxMC/AS5xx 系列机种之间存在部份差异，因此在进行项目开发之前请务必参阅各机种的使用手册；附录中的内容则是针对在 ISPSOft 中进行 AH/AS 系列机种、DVP 系列机种及 DVPxxMC/AS5xx 系列机种的项目开发时，用户须特别注意的事项来加以补充说明。

B.1 各系列机种的装置寻址

B.1.1 AH/AS 系列机种的装置种类

下表为 AH/AS 系列机种的装置种类。

装置种类	存取属性	功能说明
X	位/字符组	输入继电器，代表数字输入 (DI) 点的状态。(*1)
Y	位/字符组	输出继电器，代表数字输出 (DO) 点的状态。(*1)
M	位	内部辅助继电器，用于储存运算过程中的位状态。
S	位	步进点继电器，用来做为 SFC 步进点的状态标志。
D	位/字符组	内部数据寄存器，用来做为运算数据的储存寄存器。(*1)
L	位/字符组	链接寄存器，主要用于数据交换的暂存之用。(*1)(*2)
T	位	定时器装置，用做位装置时则代表计时到达与否的状态标志。
	字符组	定时器装置，用做字符组装置时则代表定时器的现在值。
C	位	计数器装置，用做位装置时则代表计数到达与否的状态标志。
	字符组	计数器装置，用做字符组装置时则代表计数器的现在值。
HC	位	32 位计数器装置，用做位装置时则代表计数到达与否的状态标志。
	字符组	32 位计数器装置，用做字符组装置时则代表计数器的现在值。
E	字符组	变址寄存器，用做偏移 (Offset) 运算之用。(*3)
SM	位	特殊辅助继电器 (特 M)，做为系统特定功能的状态标志。(*4)
SR	字符组	特殊数据寄存器 (特 D)，做为系统特定功能的数据寄存器。(*4)

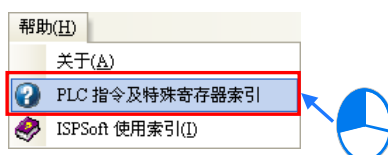
*1. 依据表达方式的不同，X、Y、D 与 L 装置可分别进行字符组或位操作，相关说明将于本节后续的内容中另行补充。

*2. 链接寄存器 (L 装置) 主要用于数据交换的暂存之用，但亦可当做一般的 D 装置来使用。AS 系列机种无 L 装置。

*3. AH/AS 机种的变址寄存器只有 E 装置，而没有 F 装置。

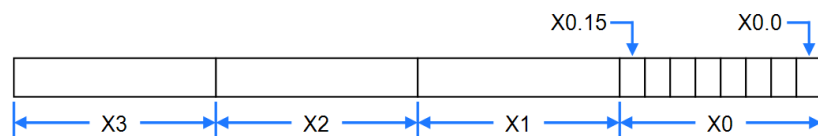
*4. 有别于 DVP 机种，AH/AS 机种将特殊辅助继电器与特殊数据寄存器独立规划为 SM 与 SR 两种类型的装置。

关于各装置的详细使用方式与限制，请参考各机种的程序手册，或在各机种项目中通过 ISPSOft 的帮助说明功能来进行查询。

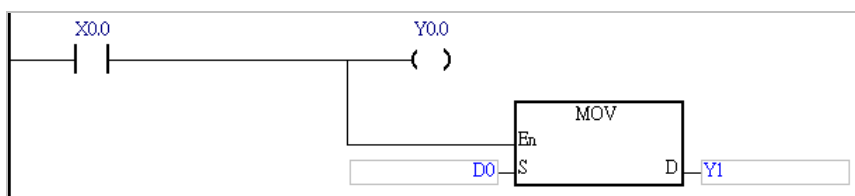


B.1.2 AH/AS 系列机种 X、Y、D、L 装置的寻址格式

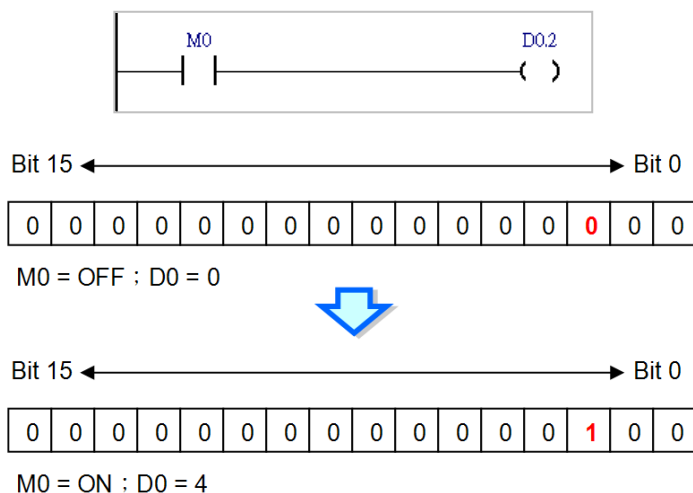
X/Y 位装置在 DVP 机种的程序是直接使用 X0、Y1...等表达方式，而 AH/AS 机种则须使用 X0.0、Y0.1 的格式来表达；AH/AS 机种的 X 与 Y 装置可直接进行字符组的操作，因此在装置表达的定义上，X0 或 Y0 都将被视为一个 WORD 的字符组装置，其中 X0.0 代表 X0 这个字符组装置的第一个位（最低位），而 X0.15 则代表最后一个位（最高位）。



如下图范例，利用把 D0 的内容搬移给 Y1 的动作，便可批次改变 Y1.0 ~ Y1.15 的位状态。



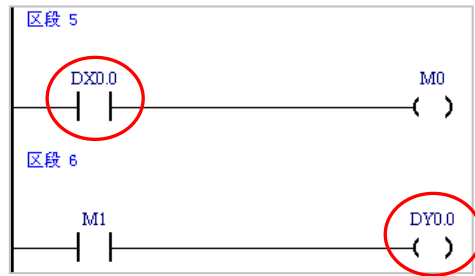
除了 X 与 Y 装置可同时使用位与字符组操作之外，在 AH/AS 系列机种中，D 与 L 等字符组装置亦可使用相同的表达方式来直接操作其中的指定位。参考下图范例，接点 M0 的状态会输出给 D0.2，因此若装置 D0 的初始值为 0，则当 M0 为 ON 时，D0 的值就会变为 4。



当 X、Y、D、L 装置在进行位或字符组操作时，基本上都是在同一个内存区块，差别仅在于存取整个字符组或是指定的位；而 T、C、HC 装置，其位与字符组装置则是完全独立的两个区块；当该装置名称用于位的指令或操作数时，其所代表的即是计数/计时到达与否的位状态标志；而若使用于字符组的指令或操作数时，该装置名称便代表计数/定时器的现在值。

B.1.3 AH/AS 系列机种 X/Y 接点状态的实时存取

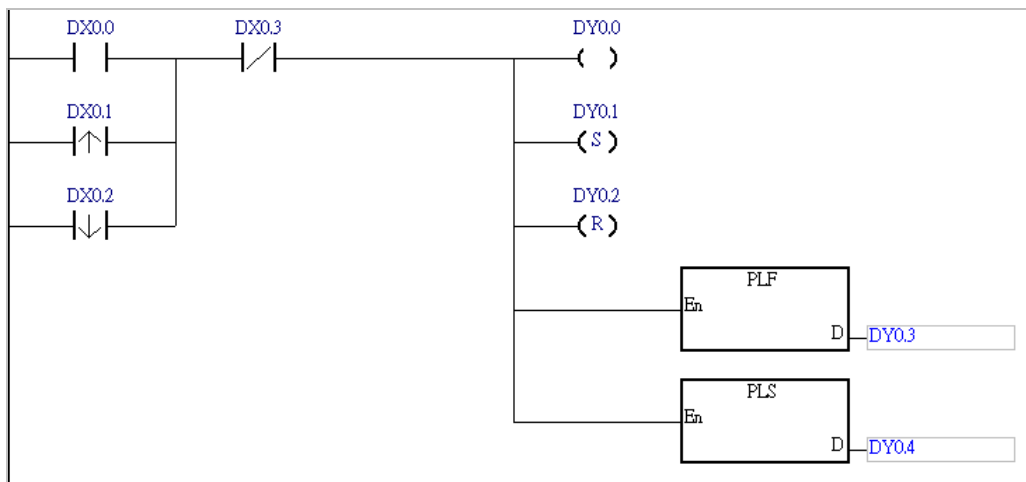
AH/AS 机种的 IO 更新时机与 DVP 机种类似，皆是在每个扫描周期的前后进行批次刷新的动作，但 AH/AS 机种尚支持特别的修饰方式，当在原本的 X 或 Y 装置前加上修饰字符「D」时，系统便会在程序执行至该指令时立即更新装置的状态；不过此用法只可修饰 X 或 Y 装置，而不可修饰变量符号。如下图，当程序执行至区段 5 时，系统便会实时读取外部输入 X0.0 端点的信号状态；同样的，当程序执行至区段 6 时，系统亦会立即将 M1 的状态经由 Y0.0 输出，而无须等待扫描周期结束。



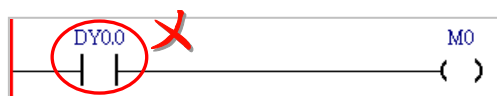
下表即为支持「D」字符修饰的指令，若在列表以外的指令中使用「D」字符修饰的 X 或 Y 装置时，系统便会将其视为错误的语法，并在编译时产生错误讯息；由下表中可得知，使用「D」字符修饰的 X 或 Y 装置只可用于位操作，且亦无法做为其他 API 应用指令的操作数。

装置种类	支持指令
X 装置	LD / LDI / LDP / LDF / OR / ORI / ORP / ORF / AND / ANI / ANDP / ANDF
Y 装置	OUT / SET / RST / PLS / PLF

下图便是一些合法的应用范例。



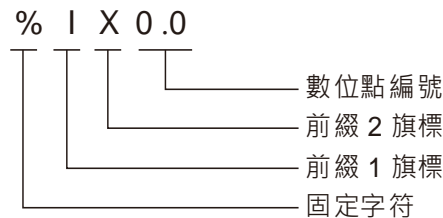
下图则为不合法的应用范例，使用「D」字符修饰的 Y 装置不可做为输入接点。



B.1.4 DVPxxMC/AS5xx 机种的装置种类

- 装置名称说明

有别于 AH/AS 机种与 DVP 机种装置，DVPxxMC/AS5xx 机种的装置，使用的是 IEC 规范，如下图。



- DVPxxMC/AS5xx 软件中使用到的相关装置清单如下：

序号	项目	内容				
1	首码 1 标志	I	Q	M	-	-
2	首码 1 名称	输入装置	输出装置	中间装置	-	-
3	首码 2 标志	X	B	W	D	L
4	首码 2 数据类型	位元装置	位元组装置	字符组装置	双字符组装置	四字符组装置
5	装置范例	%IX0.0	%IB0	%IWO	%IDO	%ILO
6		%QX0.0	%QB0	%QW0	%QD0	%QL0
7		%MX0.0	%MB0	%MW0	%MD0	%ML0



B.1.5 DVPxxMC/AS5xx 机种装置的寻址格式

- 装置对应关系如下表所示：

如下表所示，%ML0 包括 %MB0~%MB7，%MD0 包括 %MB0~%MB3，%MW0 包括 %MB0~%MB1。

装置名称	装置对应关系																																							
	第 1 个 WORD					第 2 个 WORD					第 3 个 WORD					第 4 个 WORD																								
	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15																
%MX	%MX0.0~0.7					%MX1.0~1.7					%MX2.0~2.7					%MX3.0~3.7					%MX4.0~4.7					%MX5.0~5.7					%MX6.0~6.7					%MX7.0~7.7				
%MB	%MB0					%MB1					%MB2					%MB3					%MB4					%MB5					%MB6					%MB7				
%MW	%MW0										%MW1										%MW2										%MW3									
%MD	%MD0															%MD1																								
%ML	%ML0																																							

如下表所示，%ML1 包括%MB8~%MB15，%MD2 包括%MB8~%MB11，%MW4 包括%MB8~%MB9，%MB8 包括%MX8.0~8.7。

装置名称	装置对应关系																							
	第 5 个 WORD						第 6 个 WORD						第 7 个 WORD						第 8 个 WORD					
	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15
%MX	%MX8.0 ~8.7		%MX9.0 ~9.7		%MX10.0 ~10.7		%MX11.0 ~11.7		%MX12.0 ~12.7		%MX13.0 ~13.7		%MX14.0 ~14.7		%MX15.0 ~15.7									
%MB	%MB8		%MB9		%MB10		%MB11		%MB12		%MB13		%MB14		%MB15									
%MW	%MW4						%MW5						%MW6						%MW7					
%MD	%MD2												%MD3											
%ML	%ML1																							

B.1.6 DVPxxMC/AS5xx 机种的停电保持装置

%MWO~%MW999 为停电保持装置。另外，软件中定义的变数可以选择是否为停电保持。停电保持的容量为 100K 位元组。

B.2 各系列机种的装置资源

B.2.1 AHCPU500-EN/AHCPU500-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X63	字符：X0.0~X63.15
Y	字符组：Y0~Y63	字符：Y0.0~Y63.15
D	字符组：D0~D16383	字符：D0.0~D16383.15
L	字符组：L0~L16383	字符：L0.0~L16383.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能块的「实例储存区块」数量	64 个	

B.2.2 AHCPU510-EN/AHCPU510-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X127	字符：X0.0~X127.15
Y	字符组：Y0~Y127	字符：Y0.0~Y127.15
D	字符组：D0~D32767	字符：D0.0~D32767.15
L	字符组：L0~L32767	字符：L0.0~L32767.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能块的「实例储存区块」数量	256 个	

B.2.3 AHCPU520-EN/AHCPU520-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X255	字符：X0.0~X255.15
Y	字符组：Y0~Y255	字符：Y0.0~Y255.15
D	字符组：D0~D65535	字符：D0.0~D65535.15
L	字符组：L0~L65535	字符：L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能块的「实例储存区块」数量	512 个	

B

B.2.4 AHCPU530-EN/AHCPU530-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X511	字符：X0.0~X511.15
Y	字符组：Y0~Y511	字符：Y0.0~Y511.15
D	字符组：D0~D65535	字符：D0.0~D65535.15
L	字符组：L0~L65535	字符：L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能块的「实例储存区块」数量	1024 个	

B.2.5 AHCPU501-EN/AHCPU501-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X127	字符：X0.0~X127.15
Y	字符组：Y0~Y127	字符：Y0.0~Y127.15
D	字符组：D0~D24575	字符：D0.0~D24575.15
L	字符组：L0~L24575	字符：L0.0~L24575.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	

B.2.6 AHCPU511-EN/AHCPU511-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X255	字符：X0.0~X255.15
Y	字符组：Y0~Y255	字符：Y0.0~Y255.15
D	字符组：D0~D49151	字符：D0.0~D49151.15
L	字符组：L0~L49151	字符：L0.0~L49151.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	

B.2.7 AHCPU521-EN/AHCPU521-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X511	字符：X0.0~X511.15
Y	字符组：Y0~Y511	字符：Y0.0~Y511.15
D	字符组：D0~D98303	字符：D0.0~D98303.15
L	字符组：L0~L98303	字符：L0.0~L98303.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	

B

B.2.8 AHCPU531-EN/AHCPU531-RS2 机种

装置类别	寻址范围	
X	字符组：X0~X1023	字符：X0.0~X1023.15
Y	字符组：Y0~Y1023	字符：Y0.0~Y1023.15
D	字符组：D0~D131071	字符：D0.0~D131071.15
L	字符组：L0~L131071	字符：L0.0~L131071.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	

B.2.9 AHCPU560-EN2 机种

装置类别	寻址范围	
X	字符组：X0~X4095	字符：X0.0~X4095.15
Y	字符组：Y0~Y4095	字符：Y0.0~Y4095.15
D	字符组：D0~D262143	字符：D0.0~D262143.15
L	字符组：L0~L262143	字符：L0.0~L16383.15
M	M0~M8191	
S	S0~S4095	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4095	
SM	SM0~SM4095	

B

B.2.10 AHxxEMC 系列机种

装置类别	寻址范围	
X	字符组：X0~X511	字符：X0.0~X511.15
Y	字符组：Y0~Y511	字符：Y0.0~Y511.15
D	字符组：D0~D65535	字符：D0.0~D65535.15
L	字符组：L0~L65535	字符：L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	

B.2.11 AS200/AS300 系列机种

装置类别	寻址范围	
X	字符组：X0~X63	字符：X0.0~X63.15
Y	字符组：Y0~Y63	字符：Y0.0~Y63.15
D	字符组：D0~D29999	字符：D0.0~D29999.15
M	M0~M8191	
S	S0~S2047	
C	C0~C511	
HC	HC0~HC255	
T	T0~T511	
E	E0~E14	
SR	SR0~SR2047	
SM	SM0~SM4095	

B

B.2.12 DVPxxMC/AS5xx 系列机种

装置名称	装置范围
%IX	%IX0.0~%IX127.7
%QX	%QX0.0~%QX127.7
%MX	%MX0.0~%MX131071.7
%IB	%IB0~%IB127
%QB	%QB0~%QB127
%MB	%MB0~%MB131071
%IW	%IW0~%IW63
%QW	%QW0~%QW63
%MW	%MW0~%MW65535
%ID	%ID0~%ID31
%QD	%QD0~%QD31
%MD	%MD0~%MD32767
%IL	%ILO~%IL15
%QL	%QL0~%QL15
%ML	%ML0~%ML16383

B**B.3 编译与上下载时间**

ISPSOft 的编译时间与程序的大小或变量符号的多寡，及使用的计算机等级有关；而上下载时间则依据项目大小，与联机的种类及设定的通讯速度有关。在此列出以下数据供用户参考。

计算机等级：CPU Intel(R) I7 · RAM 4GB · Windows 7 32 位。

项目内容：5000 行一个接点输出一个线圈的基本指令的阶梯形图，2 个变量符号。

编译时间：约 2 秒

项目内容：容量约 10K STEP

下载时间：通过 USB 约 4.9 秒

上载时间：通过 USB 约 4.6 秒



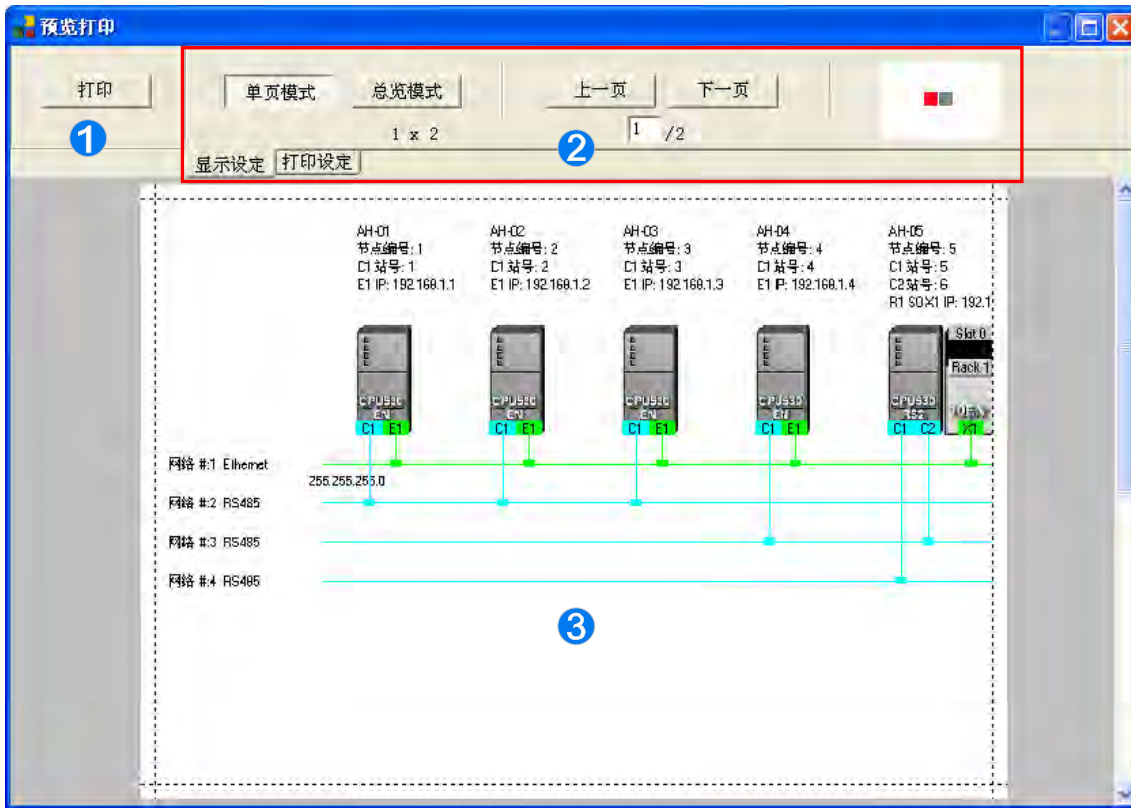
附录C 打印管理工具

目录

C.1 环境简介	C-2
C.2 操作功能说明.....	C-3

C.1 环境简介

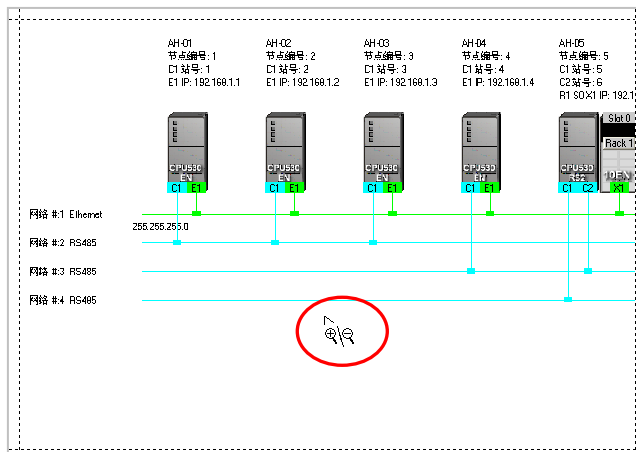
打印管理工具主要是提供一个便利的操作界面，用户可在窗口中进行打印预览与打印格式的设置。



- ❶ 打印按钮：完成设置并确认无误后，按下此键便可开始打印。
- ❷ 设定操作区：共分为「显示设定」与「打印设定」两个页面，而藉由下方的选项卡便可进行切换。
- ❸ 打印预览区：此区将显示打印结果的预览画面。

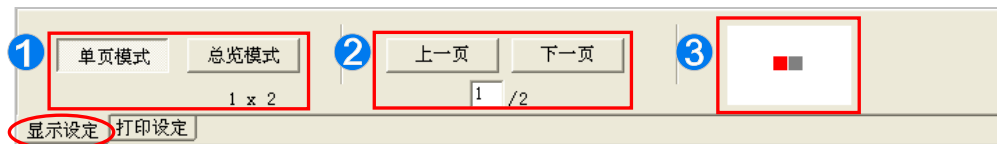


当将鼠标移至打印预览区并点击左键时，鼠标的图示便会变为放大镜的状态；此时若按下鼠标左键，预览画面便会随之放大，而按下右键时，画面便会再度缩小。



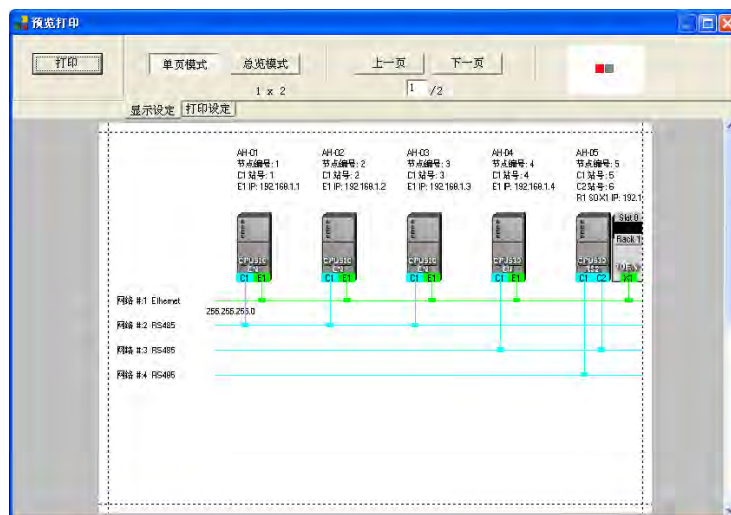
C.2 操作功能说明

下图即为设定操作区的「显示设定」页面，其主要的功能在于调整下方打印预览区的显示状态。

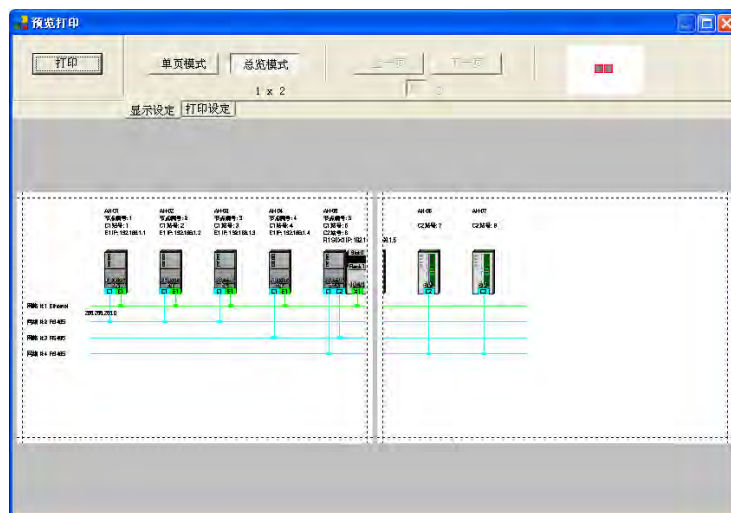


- ❶ 当打印的范围超过纸张大小时，系统将会自动将其分割为多个页面，而于此处便可切换预览画面的显示方式；其中在按钮下方的数字则代表页面切割的数量。
- ❷ 透过此处的按钮便可切换显示页面，而于其下方则会显示目前的页码与总页数。
- ❸ 此处的图标代表目前页面与整体文档的相对位置，其中红色区块即为目前页面。

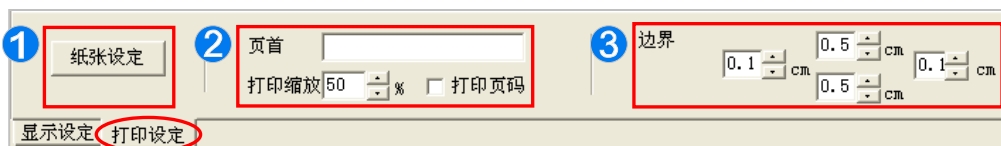
● 单页模式



● 总览模式



下图则为设定操作区的「打印设定」页面，其主要的功能在于设定打印的格式。



- ❶ 按下「纸张设定」按钮便可打开打印机的设置窗口。
- ❷ 在此可设置文件的打印比例，同时也可设定打印时是否添加页首文字与页码。
- ❸ 此处可设定打印文档的边界。



当完成所有的设置并确认无误之后，按下「打印」键便可开始打印。

