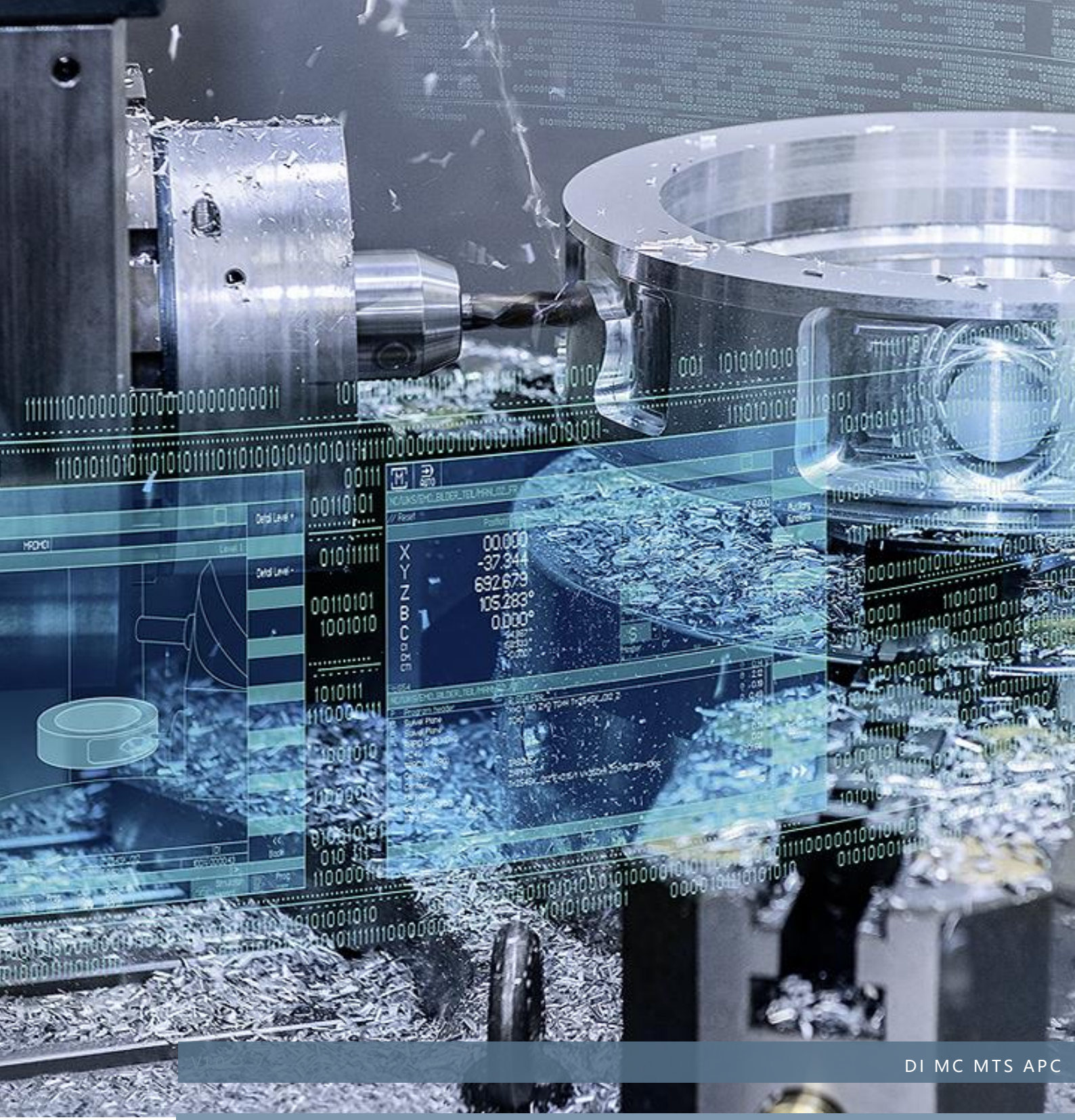


DI MC MTS APC

SINUMERIK ONE/840D SL

PLC 子程序库手册 V2.2

SINUMERIK ONE/840D SL

目录

目录

目录

目录

1 免责声明

本使用手册及样例包目录内所包含文档、PLC 程序、机床可执行程序（MPF、SPF、...）、电气图，可能与用户实际使用不同，用户可能需要先对例子程序做修改和调整，才能将其用于测试。本例程的作者和拥有者对于该例程的功能性和兼容性不负任何责任，使用该例程的风险完全由用户自行承担。由于它是免费的，所以不提供任何担保，错误纠正和热线支持，用户不必为此联系西门子技术支持与服务部门。

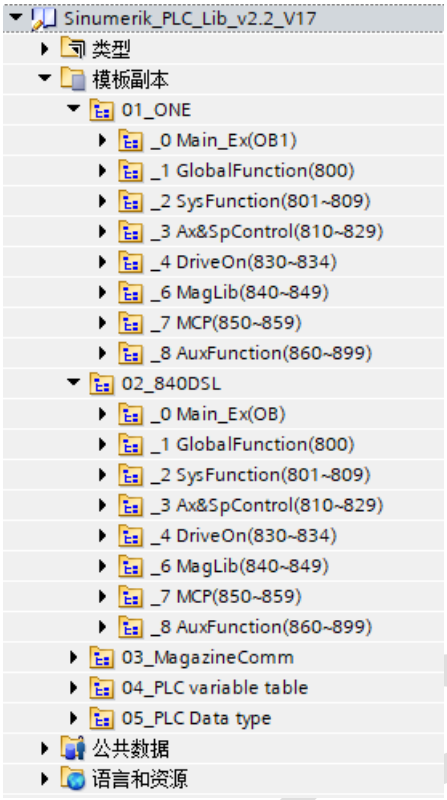
对于在使用中发生的人员、财产损失本公司不承担任何责任，由使用者自行承担风险。

以上声明内容的最终解释权归西门子（中国）有限公司所有，后续内容更新不做另行通知。

2 概述

2.1 概述

本 PLC 库程序基于 Sinumerik 840Dsl （下文简称 sl）数控系统，使用博图（TIA）软件设计的，同时适用于 Sinumeirk ONE 系统（下文简称 ONE 系统）。ONE 目录下的程序块适用于 ONE 系统，840DSL 目录下的程序块适用于 840DSL 系统。



为方便用户使用，本库文件在 GlobalDB 定义了系统常用的信号，及功能块相关的信号，使用者根据具体系统需自行调整。同时，根据 GlobalDB 信号编制示例程序，用户可根据实际情况更改，或直接与外部 IO 信号对接。

本库文件未加密保护，用户还可以根据实际需要修改程序。由于程序测试环境与用户实际使用环境可能不一致，在使用每个程序块时，需要用户根据实际情况反复测试后再使用。

2.2 命名规则

采用驼峰制，参考以下表格。

Tags	Global	Constants	PLC-Data types	Arrays
	start	MAX_SPEED	typeMotorStatus	AxesData [0..MAX] of type…
Prefix	no prefix	no prefix	prefix 'type'	no prefix
Start letter	lower case	upper case	lower case	upper case
Word Casing	camelCasing	CAPITAL_LETTERS	camelCasing	camelCasing

Basic naming Rules
- Unique, meaningful identifiers in English
- No special characters (§$%&…)
*- Maximum 24 characters

Formal Parameters	IN	OUT	INOUT	STAT	TEMP	CONSTANTS	Arrays
	enable	done	motorData	statVelocity	tempVelocity	MAX_VELOCITY	statMotors[0..MAX] of …
Prefix	no prefix	no prefix	no prefix	prefix 'stat'	prefix 'temp'	no prefix	prefix of area of declaration
Start letter	lower case	lower case	lower case	lower case	lower case	upper case	lower case
Word Casing	camelCasing	camelCasing	camelCasing	camelCasing	camelCasing	CAPITAL_LETTERS	camelCasing

DB Data block	GLOBAL	Single Instance	Multi-instance
	MotorData	InstPidHeater	instTimerMotor
Prefix	no prefix	prefix 'Inst'	prefix 'inst'
Start letter	upper case	lower case	lower case
Word Casing	camelCasing	camelCasing	camelCasing

		Prefix	Start letter	Word Casing
OB Organization block	StationMain	no prefix	upper case	camelCasing
FB Function block	ConveyorControl	no prefix	upper case	camelCasing
FC Function	Filling	no prefix	upper case	camelCasing
TO Technology object	PositioningAxis	no prefix	upper case	camelCasing
PLC tag tables		no prefix	upper case	PascalCasing
Watch tables		no prefix	upper case	PascalCasing
Traces		no prefix	upper case	PascalCasing
Measurements		no prefix	upper case	PascalCasing

*Abbreviations (only one per identifier)	
Min	minimum
Max	maximum
Act	actual, current
Next	next
Prev	previous
Avg	average
Diff	difference
Pos	position
Ris	rising edge
Fal	falling edge
Sim	simulated
Sum	sum
Old	old value

3 要求

3.1 硬件要求

- SINUMERIK 840DSL
- SINUMERIK ONE

3.2 软件版本要求

- TIA Portal STEP 7 版本>=V17.0

4 库文件使用

4.1 库文件包含功能块

PLC 模版共分为 7 个功能区，包含 13 个程序块。

功能区	编号	名称	块功能
全局功能	FC800	GBFunct840D_B	定义全局变量
	FC900	GBFunct840D_E	单个 OB1 变量运算
	DB800	GlobalDB	全局变量数据块
	DB20	DB_PLCUserData	PLC 机床数据
系统功能	FC801	Sys_MDecoder_Dyn	M 功能动态译码
	FC802	Sys_MDecoder_Ext	扩展 M 代码译码
	FB803	Sys_MDecoder_RD	读入禁止功能的 M 代码译码
	FB804	Sys_NC.PLC.ExChangeData	NC/PLC 数据交换
	FB805	Sys_Interrupt_Prog_INI	PLC 异步子程序初始化
	FB806	Sys_PasswordCancel	删除密码
	FB807	Sys_SaveAndSelectProg	加工程序存储和选择
进给轴/主轴	FC810	Ax_AllEnable	所有轴使能
	FC811	Ax_Enable_v1	进给轴控制 v1
	FC812	Ax_Enable_v2	进给轴控制 v2
	FC813	Sp_Enable	主轴使能
	FC814	Sp_ModeChange	主轴模式控制
	FB814	Ax_Enable_v3	进给轴控制 v3
	FB815	Sp_BlowAirControl	主轴油气润滑控制
	FB816	Sp_ReadSpToolType	读入主轴刀具类型
驱动上电	FC830	Dr_Estop	急停控制
	FC831	Dr_DriveOn_TIM	驱动上电 S5Timer
	FB832	Dr_DriveOn_IEC	驱动上电 IEC_Timer
刀库管理	FB840	00_TM_General	通用换刀程序示例
	FC840	TM_Load_Ack	DB71 装载应答
	FC841	TM_TCode_Ack	DB72 T 代码准备刀应答（备刀在刀库）
	FC842	TM_MCode(Arm)_Ack	DB72 M 代码应答（械手换刀）
	FC843	TM_MCode(NoArm)_Ack	DB72 M 代码应答（非机械手换刀）
	FC844	TM_init	主轴接口初始化
	FB841	TM_NCServoMag_Manual	NC 伺服刀库手动控制
	FB842	TM_NCServoMag	NC 伺服刀库控制
	FB843	TM_NCServoMag_Bero_Offset	NC 伺服刀库扩展控制
	FB844	TM_BufferMag_Ack	DB72 T/M 代码应答（备刀在机械手）
	FB845	TM_Turret	DB73 刀塔应答

	FB846	TM_HydMag	液压刀库控制（计数开关类型）
	FB847	TM_3rdServoTurret	第三方伺服刀塔控制
	FB849	TM_AckControl	T/M 代码应答管理程序
机床面板控制	FC850	OP_NCStart_Ext	标准面板增加程序启停键
	FC851	OP_MINI_BHG	Mini 手持单元
	FC852	OP_3rd_BHG	第三方手轮
	FC853	OP_OverrideEnable	倍率控制：通道、轴独立控制
	FC854	OP_OverrideEnableAll	倍率控制：通道、轴一体控制
	FC855	OP_ToggleMcsWcs_Ext	MCS<->WCS 坐标系切换
	FC856	OP_ONE_MCP_TO_483_B	ONE MCP 地址映射开始
	FC857	OP_ONE_MCP_TO_483_E	ONE MCP 地址映射结束
	FC858	OP_PowerRide	ONE MCP PowerRide 旋钮控制
	FB855	OP_PowerRideLedControl	ONE MCP PowerRide 按键控制
	FB856	OP_MCP_Clone	MCP 克隆
	FC168	OP_HT2	HT2 手轮控制
辅助功能	FC860	Aux_Chipconveyor	排屑器控制控制
	FC861	Aux_Coolant	冷却控制
	FC862	Aux_DeviceControl_1x1	单键控制系统
	FC863	Aux_DeviceControl_2x1	双键控制系统
	FC864	Aux_FlipFlop_v1	单键启停
	FC865	Aux_Hydraulic	液压控制
	FC866	Aux_Lubrication_v1	润滑控制 v1
	FC867	Aux_Lubrication_v2	润滑控制 v2
	FC868	Aux_SpLub	主轴润滑
	FC869	Aux_ThreeColorLED_v1	三色灯控制 v1
	FC870	Aux_ToggleKey_Ext	双键控制单元
	FC871	Aux_ToggleKey_INT	单键控制单元
	FC872	Aux_ThreeColorLED_v2	三色等控制 v2
	FC873	Aux_WorkLight	工作灯控制
	FC874	Aux_DeviceControl_v1	通用设备控制 v1
	FC875	Aux_DeviceControl_v2	通用设备控制 v2
	FC876	Aux_SpOriPos	主轴定向
	FC877	Aux_Lubrication_v3	润滑控制 v3

注：

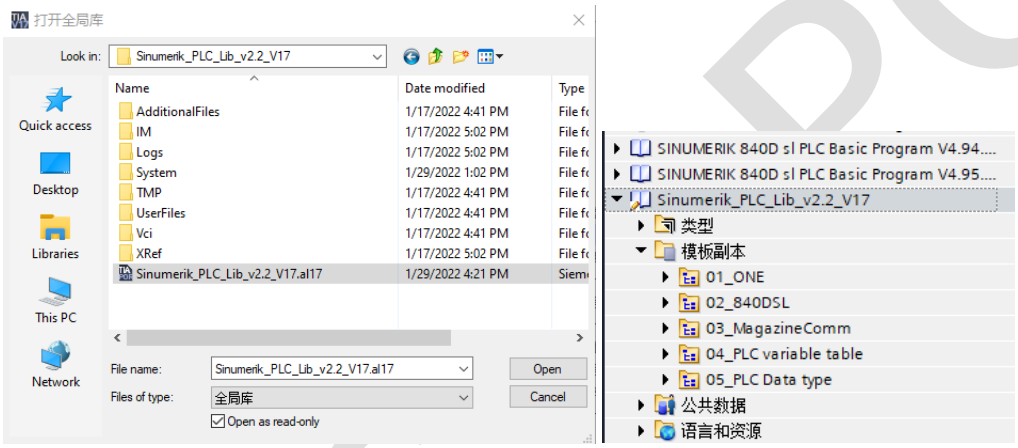
- 块名称以.sl 结尾，只适用于 sl 系统
- 块名称以.ONE 结尾，只适用于 ONE 系统
- 块名称无.sl 和.ONE 结尾，通用，适合于 sl 和 ONE 系统
- 块名称以 00 开始的程序，均为示例程序，可根据编程需要截取。

4.2 TIA 导入库文件

PLC 模版程序块范围：FC800-FC900、FB800-FB900，使用程序时，采用下面的方法导入库。

1) TIA Portal 软件中在“工程视图”工具栏选择“选项”，在其下拉菜单中选择“全局库”->”打开库”。

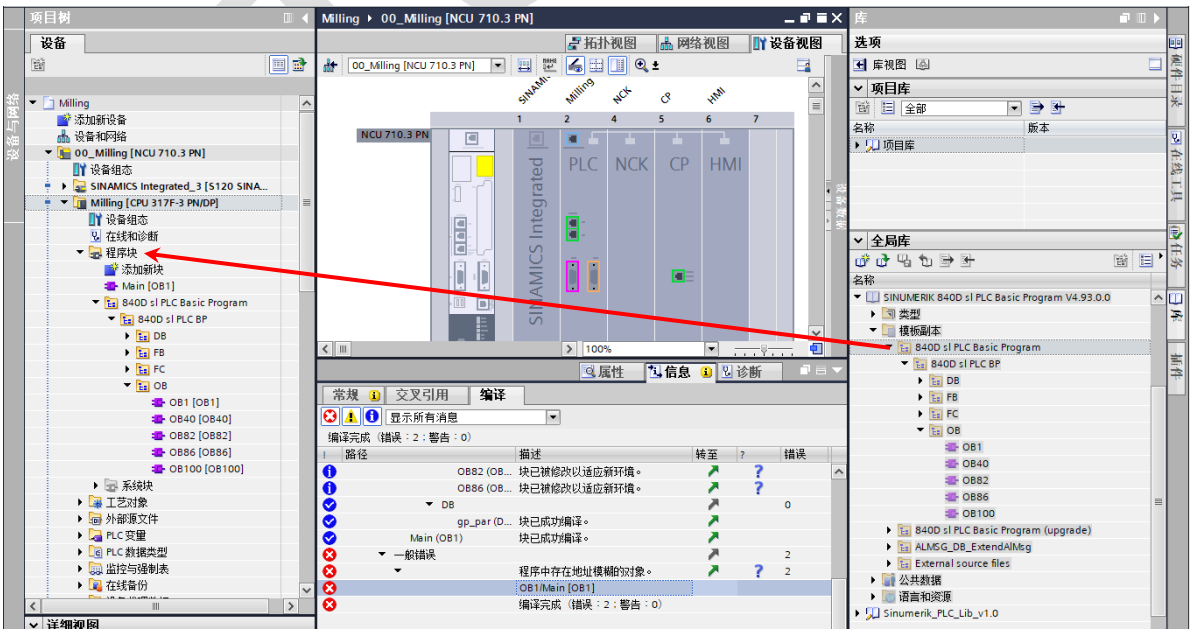
2) 选择库文件在电脑上的存储路径，直接“打开”在库选项卡就可以找到恢复的全局库，直接拖至工程项目中，直接调用就可以正常使用。



4.3 库文件加载和使用

4.3.1 装载 Sinumerik PLC basic program 程序库到项目中

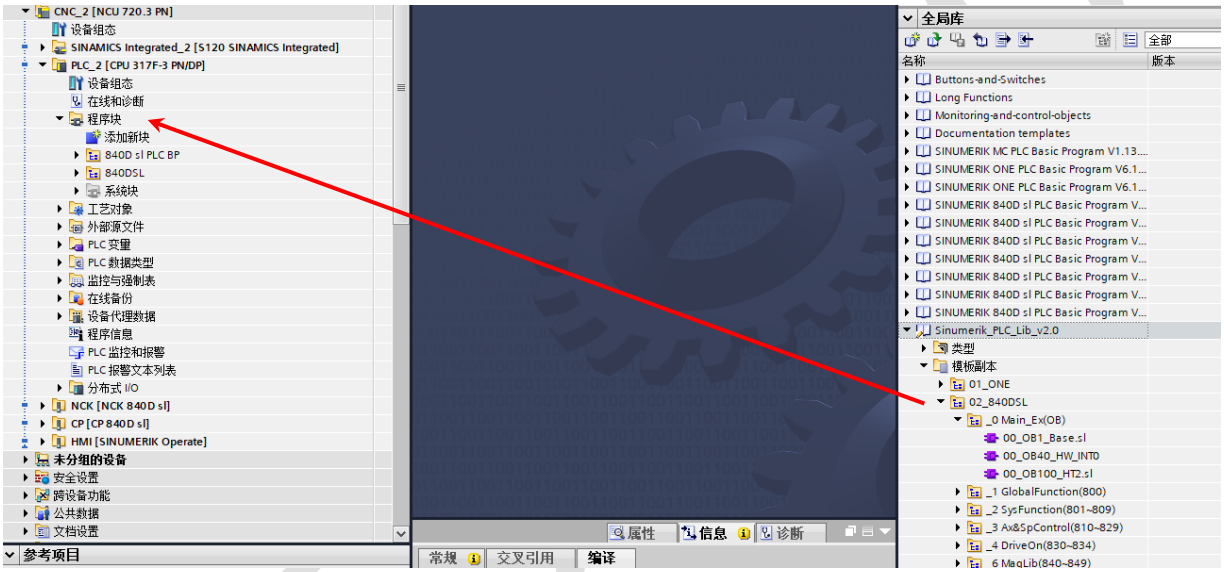
装载与硬件及软件版本相匹配的，如图，将 4.93 的 plc 系统库装载到 PLC 项目中



4.3.2 编译基本程序库

若报警，因项目中有 2 个 OB1，删除无用的 OB1，直至错误：0。

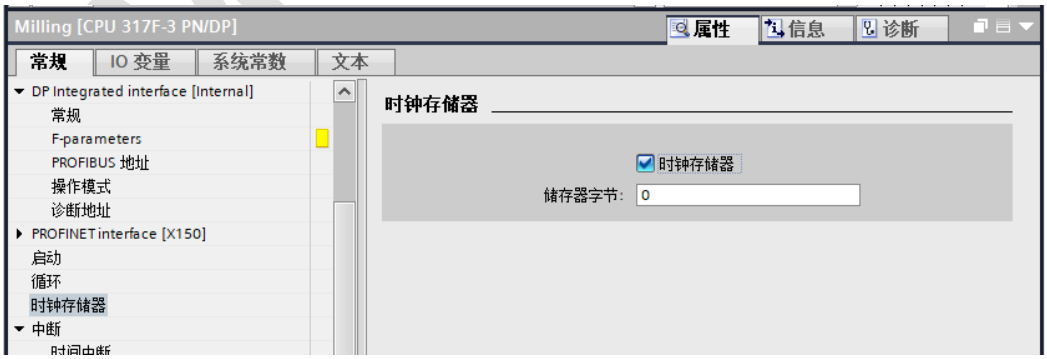
4.3.3 装载 Sinumeirk_PLC_Lib_V2.2 库文件到项目中



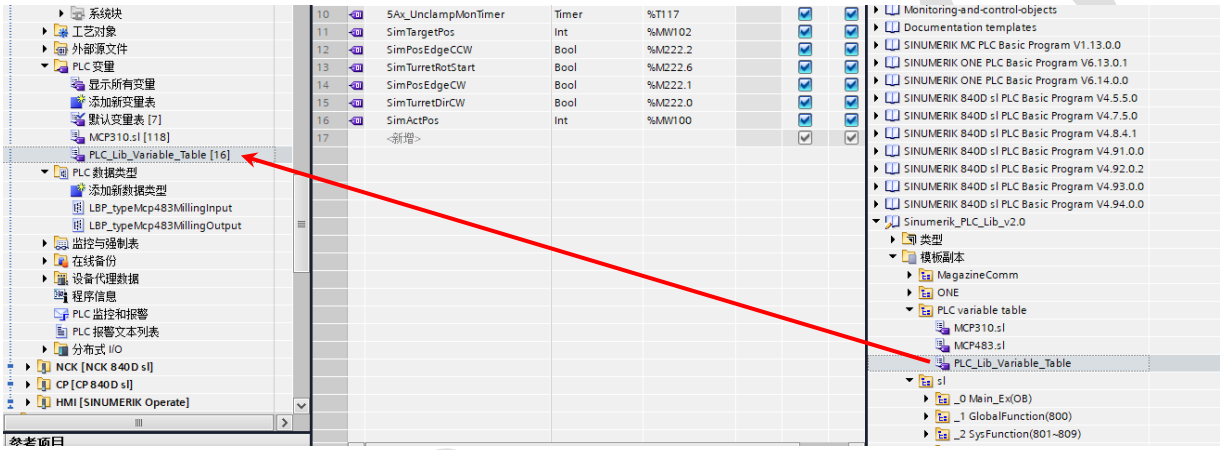
自动屏蔽 awl 源文件拷贝，同样需删除与原项目中重复的 OB1 功能块。

4.3.4 激活 PLC 时钟存储器 MB0

必须是 MB0，部分功能块以此时钟为时基计数。



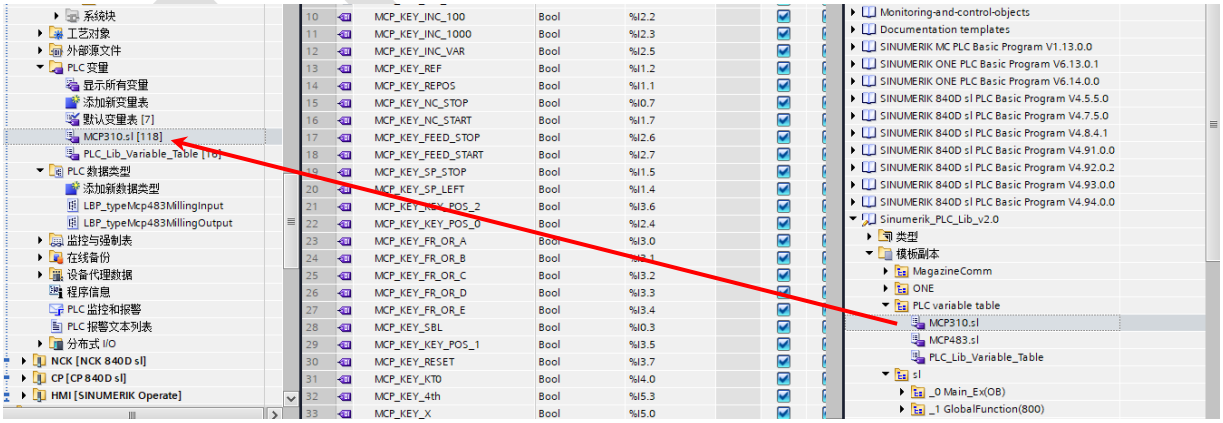
4.3.5 装载示例程序块使用的变量表



用户根据需要更新此表的设置。

4.3.6 MCP 输入输出变量表

- 对于 SINUMERIK ONE，可以直接使用对应 MCP 型号的数据类型创建 MCP 变量表
- 对于 SINUMERIK 840DSL，我们在 PLC 程序库中提供了 MCP483 和 MCP310 两种面板的 IO 符号表，输入和输出的起始地址分别为 I0.0 和 Q0.0，用户可以根据需要使用。



4.3.7 编译 Sinumeirk_PLC_Lib_V2.2 程序库

修改错误，直至正确。

5 PLC 程序库

5.1 全局功能(编号：800)

5.1.1 DB800: GlobalDB（全局功能数据块）

预设功能块及常用外设信号，主要用于功能块的使用及测试，用户根据需要将此表信号与机床外部信号对接，直接驱动示例程序。

在名称为 MCP 的结构体中，目前提供的是 MCP483 的数据结构，如果要使用 MCP310，需要自行从 MCP 数据类型中将数据结构复制到 DB800 中。

信号列表

信号	类型	描述
Clock	Struct	时钟脉冲信号
MCP	Struct	MCP483 面板相关信号
HandWheel	Struct	手轮相关信号
ONE	Bool	常”1”信号
ZERO	Bool	常“0”信号
FirstScan	Bool	PLC 首个扫描周期
RealMachine	Bool	真实 / 虚拟机床
SinumerikONE	Bool	Sinumerik ONE / 840DSL 系统
EMG	Struct	急停相关信号
Mode	Struct	工作模式相关信号
ModeGroup_Ready	Bool	模式组就绪
ModeGroup_Reset	Bool	模式组复位
AllChan_Reset	Bool	所有通道复位
NC	Struct	NC（DB10）相关信号
MachineStartOK	Bool	机床启动正常
DriveOnOK	Bool	驱动正常
DB7	Struct	DB7 相关信号
SysFunct	Struct	SysFunction 功能组各功能块接口信号
DriveOn	Struct	DriveOn 功能组各功能块接口信号
AxSpControl	Struct	Ax&SpControl 功能组各功能块接口信号
AuxFunc	Struct	AuxFunction 功能组各功能块接口信号

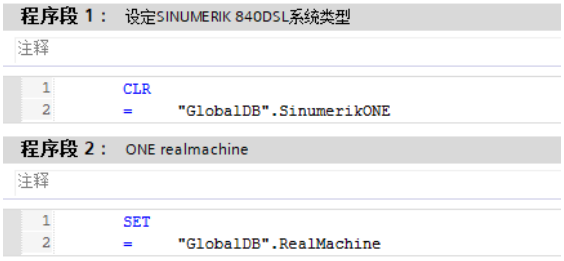
5.1.2 FC800: GBFunct840D_B（全局数据处理功能块）

程序功能

全局程序块的主要功能是统一在项目中经常使用的变量，例如时钟信号、常位信号、急停信号等等，以便项目的团队配合、调试、移植以及 HMI 开发等等。所有变量均使用 DB800 中的接口进行交互。本功能块中采用了指针方式访问如 DB7、DB11 等数据块，因此可以解决访问 Sinumerik ONE 和 840DSL 的 Toolbox 中同一数据块时因不同名称而造成的符号寻址问题。

注意事项

请务必在 FC800 的 Network 1 和 2 中正确设定当前项目的控制器类型及是否为 CMVM 虚拟调试，以免造成其他程序块的问题。



应用实例

程序	注释
CALL "GBFunct840D"	//调用 FC800

注：MCP 中的 OUT 信号，只能读取当前 MCP 的输出状态，不能修改 MCP 的输出状态。

5.1.3 FC900: GBFunct840D_E（全局数据处理功能块）

与 GBFunct840D_B 功能相呼应，在 FC10 之前调用，处理单 PLC 周期信号。

应用示例：

5.1.4 DB20：DB_PLCUserData （PLC 机床数据）

a. NC 参数设置数据的数量

14504	整数型用户数据的数量	32		po
14506	十六进制型用户数据的数量	32		po
14508	浮点型用户数据的数量	32		po

b. TIA 中根据 NC 参数设置定义同样结构的 DB20 数据块。其中 Hex 类型的数据定义为二维数组，方便在编程时直接访问其中某一个 Bit。

c. One 需要额外一步，在 OB100 的 FC1 中指定参数输出的 DB 块。

```
UserDataIntArray  :="DB_PLCUserData".MD14510    P#DB20.DBX0.0
UserDataHexArray  :="DB_PLCUserData".MD14512    P#DB20.DBX64.0
UserDataRealArray :="DB_PLCUserData".MD14514    P#DB20.DBX96.0
```

d. 使用 DB20 编程，只读，取 NC 参数设置值。

5.2 系统功能（编号：801~809）

5.2.1 FC801: Sys_Mdecoder_Dyn（M 功能动态译码）

程序功能

对通道中选择执行 M 功能（M0...M99），在 PLC 周期内解码 M 功能对应的 DB 位。

形式参数说明

信号	I/O	类型	取值范围	含义
iChan_No	I	INT	1...10	M 功能通道号
iM_Funct	I	INT	0...99	M 功能代码
oM_Signal	IO	BOOL		M 功能对应的解码接口位

应用实例

在 MDA 或 AUTO 方式下执行 M20 激活 Q401.7。

程序		注释
CALL "Sys_Mdecoder_Dyn"		//调用 FC801
iChan_No	:=1	//M 功能生效通道
iM_Funct	:=20	//M 功能：M20
oM_Signal	:=M90.0	//M20 功能触发动态解码，PLC 周期内置 1
A	M 90.0	
S	Q 401.7	//Q401.7 置位

信号周期

一个 PLC 周期

注意事项

下面系统 M 代码不支持动态译码。

- M00, M01, M02
- M03, M04, M05
- M17, M19, M30
- M40, M41, M42, M43, M44, M45
- M70, M71

5.2.2 FC802: Sys_MDecoder_Ext（扩展 M 代码译码或称静态译码）

程序功能

对通道中执行带扩展地址的 M 功能（例如：M2=20），在 PLC 周期内解码 M 功能并输出到目标位。

形式参数说明

信号	I/O	类型	含义
iChanNr	I	INT	M 功能所在通道
iExtMAddr	I	INT	M 功能扩展地址（Mx=yy 中的 x）
iMFunct	I	INT	M 功能号（Mx=yy 中的 yy）
oMStrobe	IO	BOOL	M 功能解码输出位

应用实例

在 MDA 或 AUTO 方式下执行 M2=20 并输出到 Q400.0

程序	注释
CALL “Sys_MDecoder_Ext”	//调用 FC802
iChanNr :=1	//M 功能生效通道 1
iExtMAddr :=2	//扩展 M 地址 2
iMFunct :=20	//M 功能：20
oMStrobe :=M100.0	//M2=20 功能触发动态解码，PLC 周期内置 1
A M 100.0	
S Q 400.0	//Q400.0 置位

信号周期

直至被下一个 M 代码替代。

5.2.3 FB803: Sys_MDecoder_RD（读入禁止功能的 M 代码译码）

程序功能

使用系统 DB75 功能，设置一组同组 M 代码，程序执行 M 代码，自动读入禁止（Read-Disable），PLC 需在 DB76 相应位应答，解除读入禁止，程序继续执行。

主要用于 NC 程序和 PLC 动作协作。本程序块支持 32 个 M 代码译码，支持多重背景块。

前提条件

在 OB100 中，需要设定 FB1（840DSL）或 FC1（ONE）的接口信号 ListMDecGrp 确定 M 代码译码组的数量。

注意事项

由于 840DSL 的 Toolbox 基本程序中默认不包含 DB75，因此需要将“External source files”目录下的 mdeclist.awl 源文件复制到项目中并编译生成 DB75 数据块。如果缺少 DB75 会造成 PLC 停机。

形式参数说明

信号	I/O	类型	含义
sMCodeStart	I	INT	起始 M 代码
sMCodeEnd	I	INT	结束 M 代码
iReset	I	BOOL	复位 M 代码
iAcks	I	ARRAY[0..31] of BOOL	M 代码应答
oSignals	O	ARRAY[0..31] of BOOL	M 代码输出
变量名称	**定义类型**		
iAcks	M[sMCodeStart]~M[sMCodeEnd]代码应答		
oSignals	M[sMCodeStart]~M[sMCodeEnd]代码输出		

应用实例

M 功能组 M80~M90 解码，处理 M81 代码请求及应答

程序	注释
CALL " Sys_MDecoder_RD ", "DB803"	//调用 FB803，背景数据块 DB803
sMCodeStart :=80	//起始 M 代码
sMCodeEnd :=90	//结束 M 代码
iReset :="GlobalDB".McpIn.reset	//Reset 信号
iAcks :=	//应答 M 代码请求
oSignals :=	//M 代码请求
A " DB803".oSignals[1]	//M81 代码请求
A xxxx	//互锁条件
= " DB803".iAcks[1]	//M81 代码应答

5.2.4 FB804: Sys_NC.PLC.ExChangeData（NC/PLC 数据交换）

程序功能

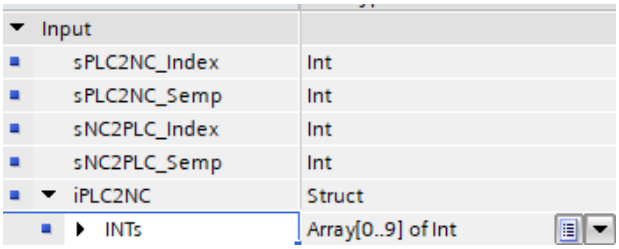
FC21 执行 NC/PLC 数据交换时是以 FC21 any 指针定义的数据类型传输的。

即若 NC/PLC 之间交换数据为 INT，则 any 指针必须定义为 word 类型，否则传输数据错误。

所以在使用本功能块时，请先定义 iPLC2NC 及 oNC2PLC 的数据结构，并指定统一的数据类型，然后修改数据区域长度。如果使用 Word 类型传输 16 位 Bool 变量，请注意高低字节的颠倒。

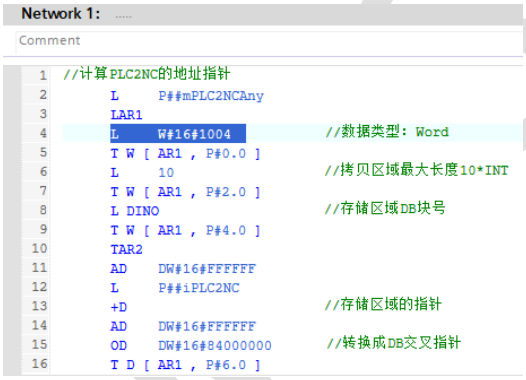
例：传输 10 个整型（INT）从 PLC 到 NC

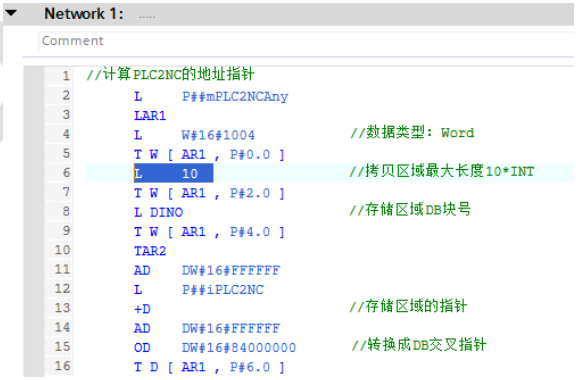
1）iPLC 定义成 0..9 的整型数组（Array[0..9] of Int）。



2）修改 PLC 程序，依据下表选择传输的数据类型，Any 指针定义成 Word 型 W#16#1004，同时修改程序中的数据类型及长度。

B#16#01[1)]	BOOL	Bits
B#16#02	BYTE	bytes, 8 bits
B#16#03	CHAR	8-bit characters
B#16#04	WORD	16-bit words
B#16#05	INT	16-bit integers
B#16#06	DWORD	32-bit words
B#16#07	DINT	32-bit integers
B#16#08	REAL	32-bit floating-point numbers





应用实例

程序	注释
CALL #"mNC.PLC.ExChangeData" sPLC2NC_Index :=0 sPLC2NC_Semp :=-1 sNC2PLC_Index :=20 sNC2PLC_Semp :=-1 iPLC2NC :="GlobalDB".SysFunct."Sys_NC.PLC.ExChangeData".iPLC2NC oNC2PLC :="GlobalDB".SysFunct."Sys_NC.PLC.ExChangeData".oNC2PLC	//PLC→NC 从$A_DBB[0]开始的 18 个字节 //NC→PLC 从$A_DBB[20]开始的 18 个字节

5.2.5 FB805: Sys_Interrupt_Prog_INI（PLC 异步子程序初始化）

程序功能

通过参数“Reset”参数初始化来定义的异步子程序。

本程序分 sl 和 ONE 版。

参数设置（举例）

子程序路径:DB100 中 path

子程序名称:DB100 中的 ProgName

中断号： Wvar1

优先级： Wvar2

也可根据需要，将上述 4 个变量作为 FB 块的输入参数，从外部调用设置即可。

形式参数说明

信号	I/O	类型	取值范围	含义
iChanNo	I	INT	1-10	异步子程序所属通道号
iInterruptNo	I	INT	1-256	异步子程序中断号
iPath	I	STRING[32]		异步子程序存储路径
iProgName	I	STRING[128]		异步子程序名称
iReset	I	BOOL		异步子程序初始化报警复位
iReq	I	BOOL		异步子程序初始化请求
oDone	O	BOOL		异步子程序初始化完成
oError	O	BOOL		异步子程序初始化错误

应用实例

程序	注释
CALL “Sys_Interrupt_Prog_INI”, “DB805”	//调用 FB805，背景数据块 DB805
iChanNo :=1	//通道号
iInterruptNo :=1	//中断号
iPath :=DB100.Path	//程序路径
iProgName :=DB100.ProgName	//程序名称
iReset :=I3.7	//报警复位
iReq :=DB10.DBX104.7	//NCU 就绪
oDone :=M300.0	//初始化完成
oError :=M300.1	//初始化错误

说明

oDone=1：初始化完成

oError=1：初始化子程序失败时，并输出 FB7 报警代码

报警后，按 Reset 重新初始化，若报警无法清除，联系西门子。

5.2.6 FB806: Sys_PasswordCancel（删除密码）

程序功能

通过 PI 服务删除 HMI 访问等级密码。

本程序分 sl 和 ONE 版。

注意事项

1) 此程序块功能是基于 FB7（背景数据块 DB16，使用时注意与使用系统的版本）实现的。
2) 如果删除口令的功能编写为重新上电自动删除密码，需要注意的是回装机床整体备份数据（即批量调试的 ARC 文件）的时候会在传完 PLC 后口令降级，可能导致后面的 NC 数据和驱动数据无法回装到系统中。所以在这种情况下，需要在回装数据前把 NCU 上的 PLC 拨码开关拨至 STOP，PLC CPU 处于停止状态后再会回装数据。

形式参数说明

信号	I/O	类型	含义
iReset	I	BOOL	PI 服务故障复位
iReq	I	BOOL	删除密码请求信号
oDone	O	BOOL	删除密码 PI 服务完成
oError	O	BOOL	删除密码 PI 服务出错

应用实例

通过 I7.5 删除 HMI 系统密码，访问等级将至 0。

程序	注释

CALL “Sys_PasswordCancel”, “DB806”
iReset :=I7.4
iReq :=I7.5
oDone :=Q5.3
oError :=Q5.4
	//调用 FB806，背景数据块 DB806
	//复位
	//触发删除密码的 PI 服务
	//完成
	//出错

5.2.7 FB807: Sys_SaveAndSelectProg（加工程序存储与选择）

程序功能

通过 PLC 实现加工程序的存储和再次调用。如果当前激活的程序后期还要再次调用，可以直接触发参数“ReadProgName”；而当触发参数“SelectProgName”时，原先保存的程序可以再次被选择激活，在 AUTO 方式下就可以直接执行程序。如果存在多个程序需要保存再次调用，可以通过多次调用本功能块实现。

本程序分 sl 和 ONE 版。

形式参数说明

信号	I/O	类型	含义
iReadProgName	I	BOOL	读取当前激活程序，存储
iSelectProgName	I	BOOL	选择存储的程序
iReset	I	BOOL	复位
oProgName	O	STRING[32]	程序名称
oWorkPieceName	O	STRING[128]	工件名称
oWorkActivePieceName	O	STRING[128]	路径
oDone	O	BOOL	完成
oError	O	BOOL	错误

应用实例

使用 I5.4 记录需要保存的程序，当需要再次调用的程序时，按下 I5.5 即可再次选择保存的程序。

程序	注释

CALL “Sys_SaveAndSelectProg” , “DB807”
iReadProgName :=I5.4
iSelectProgName :=I5.5
iReset :=I3.7
oProgName := P#DB800.DBX268.0
oWorkPieceName := P#DB800.DBX302.0
oWorkActivePieceName := P#DB800.DBX432.0
oDone :=Q5.6
oError :=Q5.7
	//调用 FB807，背景数据块 DB807
	//记录需要保存的程序名称
	//再次激活保存的程序

5.3 进给轴/主轴（编号：810~829）

5.3.1 FC810: Ax_AllEnable（所有轴使能）

程序功能

通过 DB7 判断所有有效的轴，并激活这些轴的第一编码器、控制使能、脉冲使能、倍率等轴运行的必要信号。

注意事项

该功能块一旦调用，将激活所有轴的使能，主要用于测试。

如需单独控制各轴，请勿使用此功能块。

应用实例

程序	注释
CALL " Ax_AllEnable"	//调用 FC810

5.3.2 FC811: Ax_Enable_v1（进给轴控制 v1）

程序功能

用于激活单轴的控制使能、脉冲使能、硬限位、第二软限位、参考点、测量系统切换等功能，加入了安全门信号，以保证安全。

注意事项

- 在“iMeasure2Active”配置为 TRUE 前，需要配置第二测量系统参数，否则测量系统无效。
- 只有在轴驱动就绪（DB3n.DBX93.5=1）时，轴伺服使能 DB3n.DBX2.1 才能置 1。
- 当安全门信号为低电平，无轴使能激活

形式参数说明

信号	I/O	类型	取值范围	含义
sAxNr	I	INT	1-31	机床轴号 1)
iControlEnable	I	BOOL		外部控制使能信号
iPulseEnable	I	BOOL		外部脉冲使能信号
iMeasure2Active	I	BOOL		激活直接测量系统，即第二测量系统。
iHwLimitPlus	I	BOOL		正向硬限位信号，低电平触发
iHwLimitMinus	I	BOOL		负向硬限位信号，低电平触发
iRefCam	I	BOOL		参考点信号，高电平触发
iOverrideActive	I	BOOL		倍率激活
iProtectDoorClosed	I	BOOL		安全门开关状态信号，高电平安全门闭合
iSw2ndLimitPlus	I	BOOL		激活第二正软限位 2)
iSw2ndLimitMinus	I	BOOL		激活第二负软限位

1）在通用机床数据 MD10000 中得到轴的轴号，例如 MD10000[0]=”MX1”,那么 X 轴的轴号就是 1。
2）第二软限位设置参数 MD36120&MD36130，回参考点且激活 PLC 接口信号后第二软限位生效。

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Ax_Enable_V1"	//调用 FC811
sAxNr :=1	//机床轴号
iControlEnable :="GlobalDB".AxSpControl.Ax_Enable_V1.iControlEnable	//外部控制使能
iPulseEnable :="GlobalDB".AxSpControl.Ax_Enable_V1.iPulseEnable	//外部脉冲使能
iMeasure2Active :=0	//轴第二测量系统不激活
iHwLimitPlus :="GlobalDB".AxSpControl.Ax_Enable_V1.iHwLimitPlus	//硬限位开关正，常闭点
iHwLimitMinus :="GlobalDB".AxSpControl.Ax_Enable_V1.iHwLimitMinus	//硬限位开关负，常闭点
iRefCam :="GlobalDB".AxSpControl.Ax_Enable_V1.iRefCam	//参考点挡块信号，常开点
iOverrideActive :=1	//倍率激活
iProtectDoorClosed :="GlobalDB".AxSpControl.Ax_Enable_V1.iProtectDoorClosed	//安全门状态
iSw2ndLimitPlus :="GlobalDB".AxSpControl.Ax_Enable_V1.iSw2ndLimitPlus	//第二软限位开关正，回参考点后生效
iSw2ndLimitMinus :="GlobalDB".AxSpControl.Ax_Enable_V1.iSw2ndLimitMinus	//第二软限位开关负，回参考点后生效

5.3.3 FC812: Ax_Enable_v2（进给轴控制 v2）

程序功能

用于激活单轴控制使能、脉冲使能、硬限位、参考点、测量系统切换等功能，可激活轴的跟随模式，同时集成了外部抱闸控制的信号输出。

形式参数说明

信号	I/O	类型	取值范围	含义
sAxNr	I	INT	1-31	机床轴号 1)
iControlEnable	I	BOOL		外部控制使能信号
iPulseEnable	I	BOOL		外部脉冲使能信号
iMeasure2Active	I	BOOL		激活直接测量系统，即第二测量系统。
iHwLimitPlus	I	BOOL		正向硬限位信号，低电平触发
iHwLimitMinus	I	BOOL		负向硬限位信号，低电平触发
iRefCam	I	BOOL		参考点信号，高电平触发
iOverideActive	I	BOOL		倍率激活
iFollowUpMode	I	BOOL		跟随模式
oBrakeRelease	IO	BOOL		外部抱闸释放信号

1）在通用机床数据 MD10000 中得到轴的轴号，例如 MD10000[0]=”MX1”,那么 X 轴的轴号就是 1。

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Ax_Enable_V2"	//调用 FC812
sAxNr :=3	//机床轴号
iControlEnable :="GlobalDB".AxSpControl.Ax_Enable_V2.iControlEnable	//外部控制使能
iPulseEnable :="GlobalDB".AxSpControl.Ax_Enable_V2.iPulseEnable	//外部脉冲使能
iMeasure2Active :=0	//轴第二测量系统不激活
iHwLimitPlus :="GlobalDB".AxSpControl.Ax_Enable_V2.iHwLimitPlus	//硬限位开关正，常闭点
iHwLimitMinus :="GlobalDB".AxSpControl.Ax_Enable_V2.iHwLimitMinus	//硬限位开关负，常闭点
iRefCam :="GlobalDB".AxSpControl.Ax_Enable_V2.iRefCam	//参考点挡块信号，常开点
iOverideActive :=1	//倍率激活
iFollowUpMode :=0	//跟随模式
oBrakeRelease :="GlobalDB".AxSpControl.Ax_Enable_V2.oBrakeRelease	//外部抱闸释放信号输出

5.3.4 FC813: Sp_Enable（主轴使能）

程序功能

控制主轴使能输出、测量系统选择及激活主轴倍率。

控制使能除了受外部 iSpControlEnable 信号影响，还要求系统发出旋转指令，即 DB3x.DBX64.6/64.7 为 1。

参数说明

信号	I/O	类型	含义
sSPAxNr	I	INT	主轴的机床轴号
sActMeasSys	I	BOOL	测量系统（0=无测量系统，1=电机编码器，2=外置编码器）
iSpLubPressOK	I	BOOL	主轴润滑压力正常
iSpControlEnable	I	BOOL	外部控制使能信号
iSpPulseEnable	I	BOOL	外部脉冲使能信号
oSpEnableOut	IO	BOOL	主轴使能输出，用于模拟量主轴

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Sp_Enable "	//调用 FC813
sSPAxNr :=6	//主轴的机床轴号 6
sActMeasSys :=1	//激活第一测量系统（电机编码器）
iSpLubPressOK :=1	//润滑压力正常
iSpControlEnable :="GlobalDB".AxSpControl.Sp_Enable.iSpControlEnable	//外部控制使能
iSpPulseEnable :="GlobalDB".AxSpControl.Sp_Enable.iSpPulseEnable	//外部脉冲使能
oSpEnableOut :="GlobalDB".AxSpControl.Sp_Enable.oSpEnableOut	//模拟主轴输出至变频器的使能信号

5.3.5 FC814: Sp_ModeChange（主轴模式控制）

程序功能

用于一键切换机床主轴的工作模式，当 iSpMode=1 时为定位模式，当 iSpMode=2 时主轴切换为速度控制模式。主轴工作模式的切换调用了标准块 FC18。

形式参数说明

信号	I/O	类型	含义
sSpAxNr	I	BOOL	主轴的机床轴号
iSpMode	I	BOOL	主轴模式：1- 位置模式　2-速度模式
iReset	I	INT	复位
Ref_Dword	IO	DWORD	中间变量外部存储接口

应用实例

通过 MCP 按键 T3 切换主轴模式。此处使用了功能块 FC864 ”FlipFlop”对信号进行切换，详细说明请见辅助功能章节。

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL "Aux_FlipFlop.v1" iKey :="MCP_KEY_T3" iReset :="MCP_KEY_RESET" oLamp :="MCP_LED_T3" Funct :=#SpMode Ref :=#SpModeRef	//调用 FC864 单键翻转功能块，通过 MCP 的 T3 按键对信号#SpMode 进行 0 和 1 的切换
L 1 A #SpMode JC PMD L 2 PMD: T "GlobalDB".AxSpControl.Sp_ModeChange.iSpMode	//1 位置模式 //2 速度模式

CALL " Sp_ModeChange "
sSpAxNr :=6
iSpMode :="GlobalDB".AxSpControl.Sp_ModeChange.iSpMode
iReset := "GlobalDB".MCP.In.reset
Ref_Dword :="GlobalDB".AxSpControl.Sp_ModeChange.Ref_Dword
	//调用 FC814
	//主轴机床轴号 6
	//主轴模式 1 or 2
	//复位按键
	//中间变量外部存储接口（双字）

5.3.6 FB814: Ax_Enable_v3（进给轴控制 v3）

程序功能

用于激活单轴控制使能、脉冲使能、硬限位、参考点、测量系统切换等功能。该功能块主要用于控制转台等回转类型轴，通过时间设定（程序块内）可以精确控制夹紧、放松的延时时间和监控时间，并且只有当自动方式下程序运行时才会将抱闸常开。手动方式下只有当轴被选中时，抱闸才会延时松开，当选择其他轴时，该轴会延时抱闸。同时，也可以通过 M 代码控制抱闸的夹紧与放松。

形式参数说明

信号	I/O	类型	取值范围	含义
sAxNr	I	INT	1-31	机床轴号 1)
iControlEnable	I	BOOL		外部控制使能信号
iPulseEnable	I	BOOL		外部脉冲使能信号
iMeasure2Active	I	BOOL		激活直接测量系统，即第二测量系统。
iHwLimitPlus	I	BOOL		正向硬限位信号，低电平触发
iHwLimitMinus	I	BOOL		负向硬限位信号，低电平触发
iNoHwLimit	I	BOOL		无硬限位
iHwHighActive	I	BOOL		硬限位高电平有效
iRefCam	I	BOOL		参考点
iOverrideActive	I	BOOL		倍率激活
iClamp	I	BOOL		夹紧反馈信号
iUnclamp	I	BOOL		松开反馈信号
iOFF1	I	BOOL		来自驱动上电程序块的 OFF1 输出信号
sClampMCode	I	INT		夹紧 M 代码
sUnclampMCode	I	INT		松开 M 代码
sClampMonTime	I	S5Time		夹紧监控时间
sUnClampMonTime	I	S5Time		松开监控时间
oMsgClampErr	O	BOOL		夹紧故障
oMsgUnClampErr	O	BOOL		松开故障
oBrakeRelease	IO	BOOL		抱闸输出

1）在通用机床数据 MD10000 中得到轴的轴号，例如 MD10000[0]=” MX1”,那么 X 轴的轴号就是 1。

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL #mFB814_Axis5 sAxNr :=5 iControlEnable :="GlobalDB".AxSpControl.Ax5_Enable_V3.iControlEnable	

iPulseEnable	:="GlobalDB".AxSpControl.Ax5_Enable_V3.iPulseEnable	
iMeasure2Active	:=0	
iHwLimitPlus	:=1	
iHwLimitMinus	:=1	
iNoHwLimit	:=1	
iHwHighActive	:=0	
iRefCam	:=0	
iOverrideActive	:=1	
iClamp	:="GlobalDB".AxSpControl.Ax5_Enable_V3.iClamp	//来自外部电机的夹紧反馈信号
iUnclamp	:="GlobalDB".AxSpControl.Ax5_Enable_V3.iUnclamp	//来自外部电机的松开反馈信号
iOFF1	:="GlobalDB".DriveOn.Dr_DriveOn_IEC.oOFF1	
sClampMCode	:=10	//夹紧 M 代码 M10
sUnclampMCode	:=11	//松开 M 代码 M11
sClampMonTime	:=s5t#200ms	//夹紧监控时间 200ms
sUnClampMonTime	:=s5t#200ms	//松开监控时间 200ms
ClampMonTimer	:="T40"	
UnClampMonTimer	:="T41"	
oMsgClampErr	:="GlobalDB".AxSpControl.Ax5_Enable_V3.oMsgClampErr	
oMsgUnClampErr	:="GlobalDB".AxSpControl.Ax5_Enable_V3.oMsgUnClampErr	
oBrakeRelease	:="GlobalDB".AxSpControl.Ax5_Enable_V3.oBrakeRelease	

5.3.7 FB815: Sp_BlowAirControl（主轴油气润滑控制）

程序功能

用于控制主轴的油气润滑时序。

系统启动并且使能激活后（iOn 信号上升沿），无论主轴是否旋转，油气润滑均会启动，如果主轴未启动，则超过设定时间后，油气润滑停止；如果主轴旋转过，则主轴停止后超过设定时间，油气润滑停止。

油气润滑的工作分为两个阶段，第一阶段打油的次数及每个阶段打油时间和间隔时间都可通过形参设定。

主轴旋转包括：

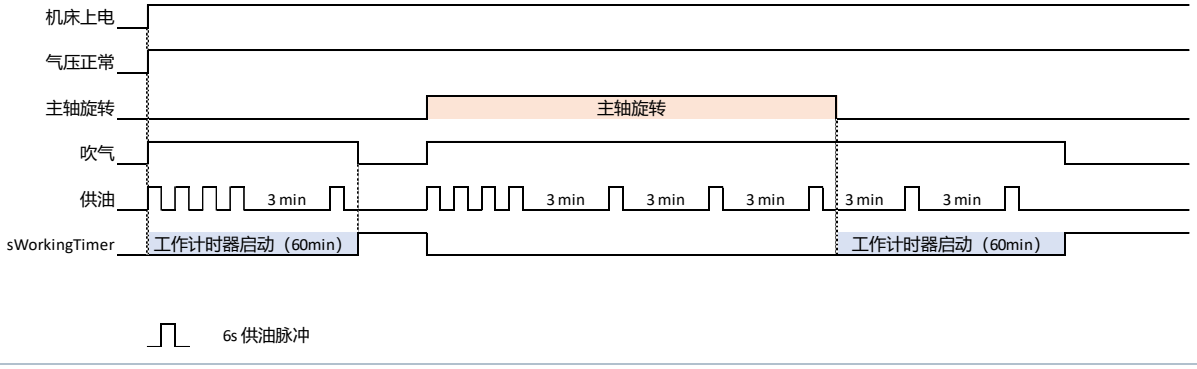
- AUTO/MDA 下通过 NC 程序启动主轴
- JOG 方式下在 TSM 界面启动主轴

不包括：JOG 方式下点动主轴

时序图

吹气间隔 60min，第一阶段打油 3 次，每次 6s，间隔 6s；第二阶段每次打油 6s，间隔 3min。

实际应用时，各工作时间可根据需要调整。



参数说明

信号	I/O	类型	含义
sSpAxNr	I	INT	主轴的机床轴号
sCycles	I	INT	第一阶段润滑次数
sOffTime	I	INT	吹气关断延时（分钟）
sPulseOnTime1st	I	INT	第一阶段单次打油时间（秒）
sPulseOffTime1st	I	INT	第一阶段单次打油间隔（秒）
sPulseOnTime2nd	I	INT	第二阶段单次打油时间（秒）
sPulseOffTime2nd	I	INT	第二阶段单次打油间隔（秒）
iOn	I	BOOL	机床上电
iSpAirPressureOK	I	BOOL	供气气压（正常为 1）
iSpLubLoadOK	I	BOOL	主轴润滑过载（正常为 1）
iSpLubLevelLow	I	BOOL	主轴润滑油液位低（正常为 0）
oSpBlowAir	O	BOOL	主轴气路输出
oSpBlowOil	O	BOOL	主轴油路输出
oSpAirPressureAlarm	O	BOOL	气压报警输出
oSPLubOverloadAlarm	O	BOOL	润滑过载输出
oSpLubLevelAlarm	O	BOOL	液位报警输出

应用举例

通过 MD14510[4]-[8]设定时间参数，第一阶段打油 3 次。

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

DB_PLCUserData 为用户自定义 PLC 数据块 DB20

程序	注释
CALL “Sp_BlowAirControl”, “DB815”	//调用 FB815，背景数据块 DB815
sSpAxNr :=6	//主轴号 6
sCycles :=3	//第一阶段润滑 3 次
sOffTime :="DB_PLCUserData".UserDataInt[4]	//MD14510[4]吹气间隔时间（分钟）
sPulseOnTime1st :="DB_PLCUserData".UserDataInt[5]	//MD14510[5]第一阶段打油时间（秒）
sPulseOffTime1st :="DB_PLCUserData".UserDataInt[6]	//MD14510[6]第一阶段打油间隔（秒）
sPulseOnTime2nd :="DB_PLCUserData".UserDataInt[7]	//MD14510[7]第二阶段打油时间（秒）
sPulseOffTime2nd :="DB_PLCUserData".UserDataInt[8]	//MD14510[8]第二阶段打油间隔（秒）
iOn :="GlobalDB".AxSpControl.Sp_BlowAirControl.iOn	//机床启动信号
iSpAirPressureOK :=1	//气压正常信号
iSpLubLoadOK :=1	//润滑过载信号
iSpLubLevelLow :=0	//润滑液位信号
oSpBlowAir :="GlobalDB".AxSpControl.Sp_BlowAirControl.oSpBlowAir	//主轴气路输出
oSpBlowOil :="GlobalDB".AxSpControl.Sp_BlowAirControl.oSpBlowOil	//主轴油路输出

oSpAirPressureAlarm :=	//气压报警输出
oSPLubOverloadAlarm :=	//润滑过载报警输出
oSpLubLevelAlarm :=	//润滑液位报警输出

5.3.8 FB816: Sp_ReadSpToolType（读取主轴刀具类型）

程序功能

用于获取缓冲区刀库 9998 各刀位（主轴、卡爪）上刀具的类型。

通过两次调用 FB2 完成：

第一次：通过变量$TC_MPP6[9998, x]获取指定 9998 缓冲区刀库中刀具的内部刀号

第二次：通过变量$TC_DP1 获取指定刀号的刀具类型

使用前提

读取刀具信息必须有刀具管理功能生效，MD18080 Bit0 及 MD20310 Bit0 须激活，详细设置请参考刀具管理相关的调试文档

参数说明

信号	I/O	类型	含义
iReq	I	BOOL	请求信号
iTOANo	I	INT	TOA 号（通道号）
iSpLocNo	I	INT	刀位号（例如：9998 的 1 为主轴、2 为卡爪 1、3 为卡爪 2）
oDone	O	BOOL	读取完成信号
oError	O	BOOL	读取出错信号
oToolType	O	REAL	读取的刀具类型

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

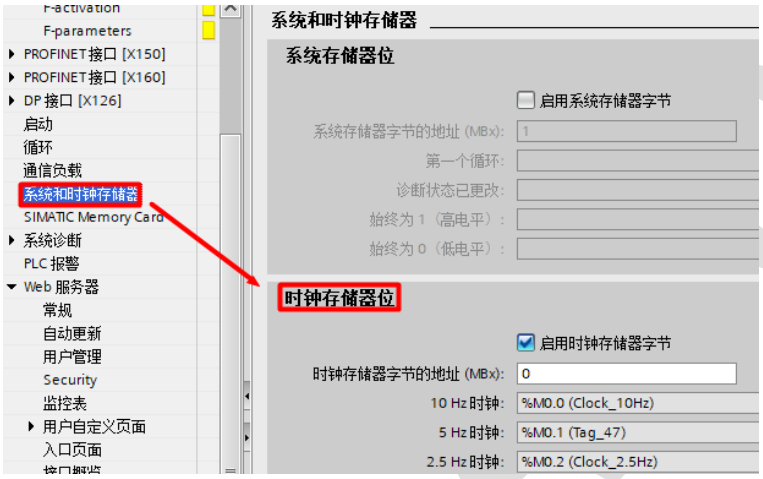
程序	注释
CALL “Sp_ReadSpToolType”, “DB816”	//调用 FB816，背景数据块 DB816
iReq :="iReadToolType"	//请求读取刀具类型
iTOANo :="GlobalDB".AxSpControl.Sp_ReadSpToolType.iTOANo	//TOA 号
iSpLocNo :="GlobalDB".AxSpControl.Sp_ReadSpToolType.iSpLocNo	//刀位号
oDone :="GlobalDB".AxSpControl.Sp_ReadSpToolType.oDone	//读取完成
oError :="GlobalDB".AxSpControl.Sp_ReadSpToolType.oError	//读取错误
oToolType :="GlobalDB".AxSpControl.Sp_ReadSpToolType.oToolType	//输出主轴刀具类型 如：平面铣刀——120.0 3D 测头 ——710.0

5.4 驱动上电（编号：830~834）

5.4.1 FC830: Dr_Estop（急停控制）

程序功能

用于急停信号的处理，包括急停的触发和应答。PLC 上电首次扫描周期或急停信号为低电平均会激活急停，仅当急停输入信号为高电平，且通过参数“Quit”进行急停应答，急停状态即可复位。在急停信号激活时，信号灯会根据“Flashing_Frequency”设置的信号变化频率闪亮。建议在 PLC CPU 中设置时钟信号，作为报警灯闪烁频率的触发信号。方法如下：



形式参数说明

信号	I/O	类型	含义
iEmgSwitch	I	BOOL	急停信号，高电平：未触发急停
iEmgAck	I	BOOL	急停信号确认/响应
iFirstScan	I	BOOL	开机标志（控制系统重启）
iFlashFrequency	I	BOOL	报警灯闪烁频率信号
oEmgLamp	IO	BOOL	急停报警灯，急停信号激活时，报警灯向界发出信号

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Dr_Estop "	//调用 FC830
iEmgSwitch :="GlobalDB".DriveOn.Dr_EStop.iEmgSwitch	//低电平触发急停报警
iEmgAck :="GlobalDB".MCP.In.reset	//报警响应信号
iFirstScan :="GlobalDB".FirstScan	//PLC 首次扫描周期信号
iFlashFrequency :="GlobalDB".Clock._2Hz	//时钟信号
oEmgLamp :="GlobalDB".DriveOn.Dr_EStop.oEmgLamp	//急停报警指示灯

5.4.2 FC831: Dr_DriveOn_TIM（驱动上电 S5Timer）

程序功能

用于控制驱动系统的 EP/OFF1/OFF3 使能信号，需要使用两个 S5Timer，由用户通过形参指定。

本程序块参考《840DSL 简明调试手册 V4.8 SP2》中“驱动系统上电时序”的相关说明。

- 上电过程：上电信号（需保持）->EP24 输出->EPReady 输入->OFF1 输出->OFF1Ready 输入->OFF3 输出
- 下电过程：上电信号取消 ->OFF3 输出取消
  ->OFF1 输出延时（OFF1_Delay_Timer）取消
  ->EP 输出延时（EP_Delay_Timer）取消

形式参数说明

信号	I/O	类型	含义
sOFF1_DelayTimer	I	TIMER	OFF1 信号延时关断定时器
sEp_DelayTimer	I	TIMER	EP 信号延时关断定时器
iOFF1_DelayTime	I	S5TIME	OFF1 信号延时关断时间
iEp_DelayTime	I	S5TIME	EP 信号延时关断时间
iDriveOnCondition	I	BOOL	驱动上电启动信号
iEpReady	I	BOOL	EP 就绪
iOFF1Ready	I	BOOL	OFF1 就绪
oEP	O	BOOL	EP 使能输出
oOFF1	O	BOOL	OFF1 使能输出
oOFF3	O	BOOL	OFF3 使能输出
Ref_Byte	IO	BYTE	中间变量外部存储接口

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Dr_DriveOn_TIM "	//调用 FC831
sOFF1_DelayTimer :="Off1_Delay_Timer"	//OFF1 延时关断定时器
sEp_DelayTimer :="EP_Delay_Timer"	//EP 延时关断定时器
iOFF1_DelayTime :=S5T#200ms	//OFF1 延时关断时间
iEp_DelayTime :=S5T#500ms	//EP 延时关断时间
iDriveOnCondition :="GlobalDB".DriveOn.Dr_DriveOn_Tim.iDriveOnCondition	//驱动上电启动
iEpReady :="GlobalDB".DriveOn.Dr_DriveOn_Tim.iEpReady	//NCU X132.10 EPReady
iOFF1Ready :="GlobalDB".DriveOn.Dr_DriveOn_Tim.iOFF1Ready	//NCU X132.9 OFF1Ready
oEP :="GlobalDB".DriveOn.Dr_DriveOn_Tim.oEP	//EP24->ALM/SLM 的 X21.3
oOFF1 :="GlobalDB".DriveOn.Dr_DriveOn_Tim.oOFF1	//OFF1>NCU X122.1
oOFF3 :="GlobalDB".DriveOn.Dr_DriveOn_Tim.oOFF3	//OFF3->NCU X122.2
Ref_Byte :="GlobalDB".DriveOn.Dr_DriveOn_Tim.Ref_Byte	//块内中间变量外部存储字节

5.4.3 FB832: Dr_DriveOn_IEC（驱动上电 IEC_Timer）

程序功能

用于控制驱动系统的 EP/OFF1/OFF3 使能信号，EP 与 OFF1、OFF1 与 OFF3 的延迟时间可以通过形参设定，程序块中所用到的定时器均为 IEC_Timer，不占用系统的 S5Timer。

- 上电过程：按键触发-> EP24 输出->EPReady 输入->EP_OFF1 延时->OFF1 输出->OFF1Ready 输入->OFF1_OFF3 延时->OFF3 输出
- 下电过程：长按按键 2s-> OFF3 输出取消->OFF1_OFF3 延时->OFF1 取消->OFF1_EP 延时->EP 取消
- 当系统急停激活时（DB10.DBX106.1），仅 OFF3 取消（制动停止）

形式参数说明

信号	I/O	类型	含义
iDriveOnKey	I	BOOL	自定义按键信号
iEPReady	I	BOOL	Infeed Ready 信号，来自 NCU 的 X132.10
iOFF1Ready	I	BOOL	Infeed Operation 信号，来自 NCU 的 X132.9
iEp_OFF1_DelayTime	I	DINT	Ep->OFF1 延迟时间，单位：ms
iOFF1_OFF3_DelayTime	I	DINT	OFF1->OFF3 延迟时间，单位：ms
oEP	Q	BOOL	EP 使能输出
oOFF1	Q	BOOL	OFF1 使能输出
oOFF3	Q	BOOL	OFF3 使能输出
oDriveOnLed	Q	BOOL	上电状态指示灯输出

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL “Dr_DriveOn_IEC”, “DB832”	//调用 FB832，背景数据块 DB832
iDriveOnKey :="GlobalDB".DriveOn.Dr_DriveOn_IEC.iDriveOnKey	//启动按键
iEPReady :="GlobalDB".DriveOn.Dr_DriveOn_IEC.iEPReady	//NCU X132.10 EPReady
iOFF1Ready :="GlobalDB".DriveOn.Dr_DriveOn_IEC.iOFF1Ready	//NCU X132.9 OFF1Ready
iEp_OFF1_DelayTime :=200	//EP OFF1 延迟 200ms
iOFF1_OFF3_DelayTime :=200	//OFF1 OFF3 延迟 200ms
oEP :="GlobalDB".DriveOn.Dr_DriveOn_IEC.oEP	//EP24->ALM/SLM 的 X21.3
oOFF1 :="GlobalDB".DriveOn.Dr_DriveOn_IEC.oOFF1	//OFF1->NCU X122.1
oOFF3 :="GlobalDB".DriveOn.Dr_DriveOn_IEC.oOFF3	//OFF3->NCU X122.2
oDriveOnLed :="GlobalDB".DriveOn.Dr_DriveOn_IEC.oDriveOnLed	//上电状态指示灯

5.5 刀库控制（编号：840~849）

此版本刀库控制程序支持双主轴和双刀库。支持的刀库类型有：链式刀库（备刀在机械手）、斗笠式刀库、凸轮机械手刀库（备刀在刀库）、矩阵式刀库、夹臂式刀库和车床刀塔。

5.5.1 刀库程序示例

程序功能

PLC 部分: FB840 (00_TM_General)，涵盖所有刀库类型的控制主程序，使用时根据不同刀库进行调整。实际调试用户需要编写的按换刀步骤的应答程序，替换 Network22~24 部分的模拟应答程序。其他说明参考 FB840 程序中的注释。

NC 部分：包括各种类型 L6、TChange 换刀子程序和刀库配置文件，用户可以按实际刀库配置选择。同时还附上刀库设置的重要 NC 参数，详细刀库配置过程请参考《840DSL 刀库管理简明调试手册》。

相关说明

PLC程序								
Network	功能分类	功能	通用程序 FB840					
			链式（NC伺服）+矩阵	链式（液压）	凸轮机械手	斗笠式	夹臂式刀塔	第三方伺服刀塔
1	基本块	初始化	√	√	√	√		
2		FC8刀库信息刷新	√	√	√	√		
3		NC/PLC数据交换	√（用于SP1）	√	√	√		
4		M代码解码				√	√	
5		主轴接口信号初始化	√（2个主轴）	√	√（1个主轴）	√（1个主轴）		
6		装载点应答	√（链式2个，矩阵1个）	√	√（2个装载点）	√（2个装载点）		
7		复位	√	√	√	√	√	√
8	换刀指令应答	多刀库T/M应答逻辑控制	√					
9		（备刀在机械手）T/M应答	√（主轴1刀库1）	√				
10			√（主轴2刀库1）					
11		（备刀在刀库）T应答	√		√	√		
12		（备刀在刀库，无机械手）M应答	√			√		
13		（备刀在刀库，有机械手）M应答			√			
14		（刀塔）T应答					√	√
15	刀库手动信号	刀库手动控制	√	√	√	√	√	√
16		刀库启动	√	√	√（1个主轴）	√（1个主轴）	√（刀塔部分）	
17	刀库控制程序	NC伺服刀库FB2读取轴位置	√					
18		NC伺服刀库控制程序	√					
19		液压刀库控制程序		√	√	√	√	
20		第三方伺服刀库控制程序						√
21		矩阵式刀库（无刀库动作）	√					
22	模拟应答信号	模拟应答：链式刀库TCI、T、M	√	√				
23			√					
24		模拟应答：凸轮机械手			√			

NC程序								
			链式（NC伺服）+矩阵	链式（液压）	凸轮机械手	斗笠式	夹臂式刀塔	第三方伺服刀塔
1		L6换刀子程序	L6					
2		手动换刀	wManualTool	wManualTool	wManualTool	wManualTool		
3		PLC换刀	wPLC_CHG	wPLC_CHG	wPLC_CHG			
4		NC换刀	wNC_CHG_Matrix		wNC_CHG_Matrix	wNC_CHG_NoArm		
5		夹臂式刀具到主轴					TOOL2SP	
6		夹臂式刀具到刀库					TOOL2MAG	
7		夹臂式刀塔换刀					TCHANGE_JB	
8		车床刀塔换刀						TCHANGE_TURN
9		刀库配置文件	MagConf_2Sp2Mag	MagConf_Arm	MagConf_Arm	MagConf_NoArm	MagConf_JB	MagConf_Turret

主要NC机床数据							
机床参数	含义	链式（NC伺服）+矩阵	链式（液压）	凸轮机械手	斗笠式	夹臂式刀塔	第三方伺服刀塔
MD10715	$MN_M_NO_FCT_CYCLE	6				-	-
MD10716	$MN_M_NO_FCT_CYCLE_NAME	L6				-	-
MD10717	$MN_T_NO_FCT_CYCLE_NAME	-				TCHANGE	
MD15710	$MN_TCA_CYCLE_NAME	-				TCHANGE	
MD18080	$MN_MM_TOOL_MANAGEMENT_MASK	BH=Bit0,1,3					20BH=Bit0,1,3,9
MD20270	$MC_CUTTING_EDGE_DEFAULT	1					-2
MD20310	$MC_TOOL_MANAGEMENT_MASK	80400BH=Bit0,1,3,14,23					
MD22550	$MC_TOOL_CHANGE_MODE	1				0	
MD22560	$MC_TOOL_CHANGE_M_CODE	206				-	-

5.5.2 PLC 程序块清单

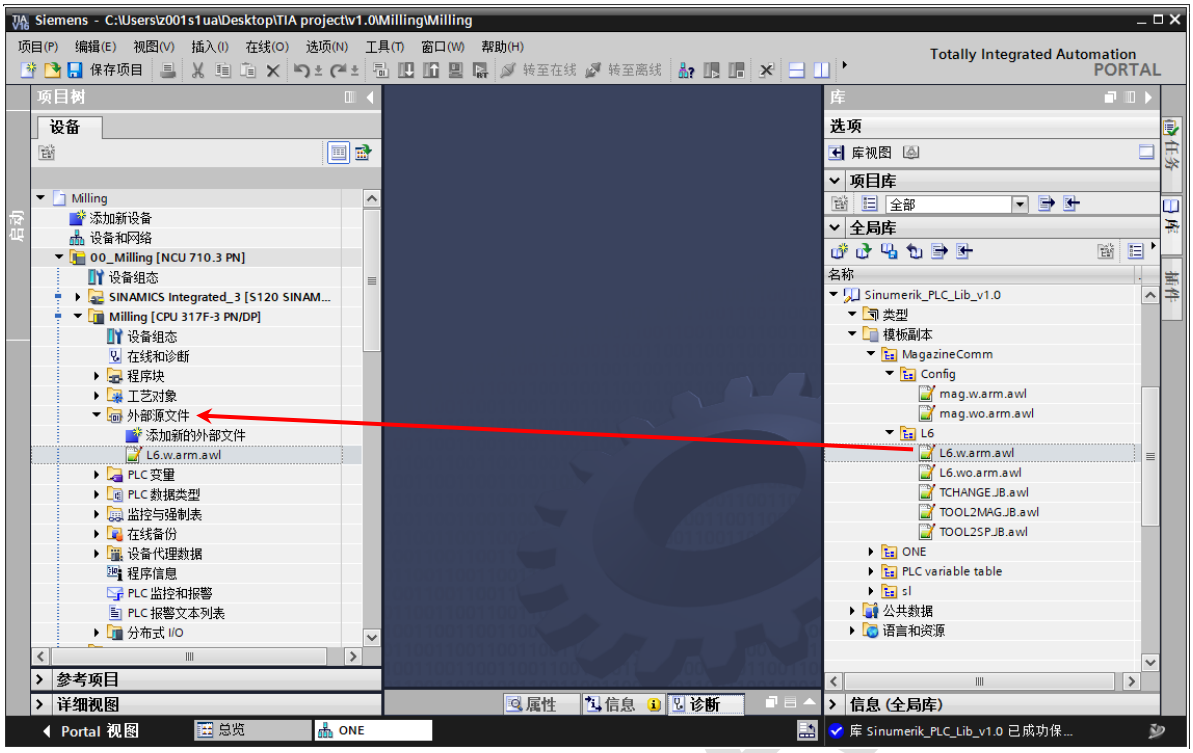
名称	功能块号	说明
00_TM_General	FB840	通用换刀程序示例
TM_Load_Ack	FC840	DB71 装载应答
TM_TCode_Ack	FC841	DB72 T 代码准备刀应答（备刀在刀库）
TM_MCode(Arm)_Ack	FC842	DB72 M 代码应答（械手换刀）
TM_MCode(NoArm)_Ack	FC843	DB72 M 代码应答（非机械手换刀）
TM_init	FC844	主轴接口初始化
TM_NCServoMag_Manual	FB841	NC 伺服刀库手动控制
TM_NCServoMag	FB842	NC 伺服刀库控制
TM_NCServoMag_Bero_Offset	FB843	NC 伺服刀库控制扩展
TM_BufferMag_Ack	FB844	DB72 T/M 代码应答（备刀在机械手）
TM_Turret_Ack	FB845	DB73 刀塔应答
TM_HydMag	FB846	液压刀库控制（计数开关类型）
TM_3rdServoTurret	FB847	第三方伺服刀塔控制
TM_AckControl	FB849	T/M 代码应答管理程序

5.5.3 NC 换刀子程序

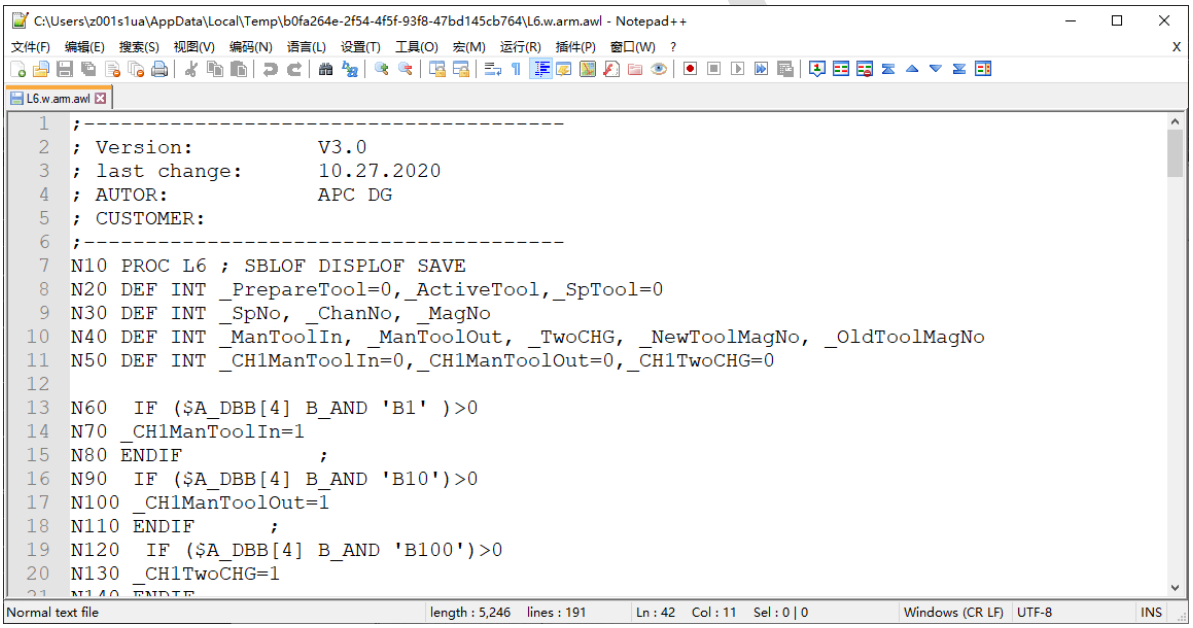
Magazine->L6 目录包含换刀子程序的 awl 源文件。

名称	说明
L6.awl	**换刀程序，使用时修改后缀为.SPF，根据刀库类型修改调用的子程序：** **PLC 换刀：待机位和凸轮机械手刀库，调用 wPLC_CHG 换刀** **NC 换刀：斗笠式刀库和矩阵式刀库，调用 wNC_CHG 换刀**
TChange_JB.awl	**夹臂式刀库换刀程序（使用时修改名称为 TChange.SPF）**
TOOL2MAG.awl	夹臂式刀具到刀库子程序
TOOL2SP.awl	夹臂式刀具到主轴子程序
TChange_Turn.awl	**车床刀塔换刀程序（使用时修改名称为 TChange.SPF）**
wManualTool.awl	手动刀具换刀子程序
wPLC_CHG.awl	PLC 换刀子程序（如：链式刀库机械手换刀）
wNC_CHG_NoArm.awl	斗笠式刀库 NC 换刀子程序（使用时修改名称为 wNC_CHG.SPF）
wNC_CHG_Matrix.awl	矩阵式刀库 NC 换刀子程序（使用时修改名称为 wNC_CHG.SPF）

使用方法：源文件拖至项目->外部源文件目录中



双击源文件，使用 TIA 配置的文本编辑器打开，另存为...



请注意，由于换刀程序中使用了一些 PLC 用户数据 MD14514[x]作为各种坐标位置变量，实际调试需要根据情况自行定义并校准位置数据，以免发生碰撞！

5.5.4 刀库配置文件

Magazine->Config 目录包含刀库配置的 awl 源文件。

名称	说明
MagConf_2Sp2Mag.awl	双刀库双主轴配置程序（刀库 1：链式　刀库 2：矩阵）
MagConf_Arm.awl	凸轮机械手刀库配置程序
MagConf_NoArm.awl	斗笠式刀库配置程序
MagConf_JB.awl	夹臂式刀库配置程序
MagConf_Turret.awl	车床刀塔配置程序

使用方法同换刀子程序。

5.5.5 FC840: TM_Load_Ack（DB71 装载应答）

程序功能

刀具管理 DB71 信号应答，一个装载点支持多个刀库，每个刀库单独调用。

例：每个刀库必须配置装载点 1，所以每个调用一次装载点 1。

形式参数说明

信号	I/O	类型	取值范围	含义
sDINO	I	Int		含 FC8 背景数据块
sIFNo	I	Int		检测装载点号
sMagNo	I	Int		刀库号
sSpLocNo	I	Int		主轴刀套号
sChanNo	I	Int		通道号
sOffset	I	Int		装载点偏置
sMode	I	Bool		刀库旋转方式 0：旋转　1：不旋转
iAck	I	Bool		装载点动作完成应答
iReset	I	Bool		复位
iMagInPos	I	Bool		刀库位置到达
iMagCurrentPos	I	Int		当前刀库位置
oMagReqPos	IO	Int		请求刀库旋转位置
oMagOffset	IO	Int		刀库位置偏置
oMagStart	IO	Bool		请求刀库旋转
Ref_int	IO	Int		中间变量存储

应用实例

程序	注释

CALL "TM_Load_Ack"
sDINO :=#mDINO
sIFNo :=1
sMagNo :=1
sSpLocNo :=1
sChanNo :=1
sOffset :=0
sMode :=false
iAck :=#Load[1].iAck
iReset :=#Load[1].iReset
iMagInPos :=#Mag[1].oInpos
iMagCurrentPos :=#Mag[1].oCurrentPos
oMagReqPos :=#Load[1].oMag[1].oMagReqPos
oMagOffset :=#Load[1].oMag[1].oMagOffset
oMagStart :=#Load[1].oMag[1].oMagStartMov
Ref_int :=#Load[1].oMag[1].Ref_int
	//装载点号
	//通道号
	//装在点偏置
	//是否输出刀库旋转指令

5.5.6 FC841: TM_TCode_Ack（DB72 T 指令备刀应答）

程序功能

T 代码应答，准备预选刀具，用于刀具待机位在刀库的场合。

形式参数说明

信号	I/O	类型	取值范围	含义
T_Enable	I	Bool		1：执行 T 代码应答　　0：无效
sDINO	I	Int		含 FC8 背景数据块
sIFNo	I	Int		主轴号
sMagNo	I	Int		设置相应的刀库号
sOffset	I	Int		主轴位置偏置
iReset	I	Bool		复位
iMagInPos	I	Bool		刀库到位
oTStep	IO	Int		预留（内部）
oMagNo	IO	Int		刀库号
oMagReqPos	IO	Int		请求刀库旋转位置
oMagOffset	IO	Int		刀库位置偏置
oMagStartMov	IO	Bool		请求刀库旋转

应用实例

程序	注释

CALL "TM_TCode_Ack"
T_Enable :=#mAckControl.oRules[2]
sDINO :=#mDINO
sIFNo :=1
sMagNo :=2
sOffset :=0
iMagInPos :=#Mag[2].oInpos
oTStep :=#Sp[1].oMag[2].oStep
oMagNo :=#Sp[1].oMag[2].oMagNo
oMagReqPos :=#Sp[1].oMag[2].oMagReqPos
oMagOffset :=#Sp[1].oMag[2].oMagOffset
oMagStartMov :=#Sp[1].oMag[2].oMagStartMov
	//控制信号，来自 FB849 输出的规则，oRules[2]控制 2 号刀库 T 应答
	//主轴接口号 1
	//当前刀库号 2

5.5.7 FC842: TM_MCode(Arm)_Ack（DB72 凸轮机械手刀库 M 指令应答）

程序功能

凸轮机械手换刀应答

形式参数说明

信号	I/O	类型	取值范围	含义
M_Enable	I	Bool		1：执行 M 代码应答 0：无效
sDINO	I	Int		含 FC8 背景数据块
sIFNo	I	Int		主轴接口号（主轴 1、主轴 2...）
sGr1LocNo	I	Int		机械手 1 位置号
sGr2LocNo	I	Int		机械手 2 位置号
sMagNo	I	Int		刀库号
iM1	I	Bool		第 1 步到位应答
iM2	I	Bool		第 2 步到位应答
iM3	I	Bool		第 3 步到位应答
oMStep	IO	Int		换刀步骤 =1：抓旧刀 =2：换新刀 =3：回待机位

应用实例

程序	注释

CALL "TM_MCode(Arm)_Ack"
M_Enable := false
sDINO :=#mDINO
sIFNo :=1
sGr1LocNo :=2
sGr2LocNo :=3
sMagNo :=3
iManTool :=#Sp[1].iManTool
iM1 :=#Sp[1].iM1
iM2 :=#Sp[1].iM2
iM3 :=#Sp[1].iM3
oMStep :=#Sp[1].oMag[3].oMStep
	//控制信号，来自 FB849 输出的规则 oRules[x]，此处未使用该刀库，直接设 false
	
	//当前刀库号 3

5.5.8 FC843: TM_MCode(NoArm)_Ack（DB72 斗笠式刀库 M 指令应答）

程序功能

无机械手刀库 M 代码应答，根据主轴刀具状态，自动应答。

形式参数说明

信号	I/O	类型	取值范围	含义
M_Enable	I	Bool		1：执行 M 代码应答　　0：无效
sDINO	I	Int		含 FC8 背景数据块
sIFNo	I	Int		检测主轴号
sMagNo	I	Int		刀库号

应用实例

程序	注释

CALL "TM_MCode(NoArm)_Ack"
M_Enable :=#mAckControl.oRules[4]
sDINO :=#mDINO
sIFNo :=1
sMagNo :=2
	//控制信号，来自 FB849 输出的规则，oRules[4]控制 2 号刀库 M 应答
	
	//当前刀库号 2

5.5.9 FC844: TM_init（TM 主轴接口初始化）

程序功能

实现只与主轴接口相关的功能，如：复位、手动刀具应答，输出 DB72 接口信息

形式参数说明

信号	I/O	类型	取值范围	含义
sDINO	I	Int		含 FC8 背景数据块
sIFNo	I	Int		主轴接口号
iReset	I	Bool		复位
iManualTool	I	Bool		手动刀应答
oNewToolMagNo	O	Int		新刀刀库号
oOldToolMagNo	O	Int		旧刀刀库号
oReq	O	Bool		应答请求
oManualToolIn	O	Bool		装载手动刀具请求
oManualToolOut	O	Bool		卸载手动刀具请求

应用实例

程序	注释
CALL "TM_init" sDINO :=#mDINO sIFNo :=1 iReset :=#Sp[1].iReset iManualTool :=#Sp[1].iManTool oNewToolMagNo :=#mAckControl.InitData[1].NewToolMagNo oOldToolMagNo :=#mAckControl.InitData[1].OldToolMagNo oReq :=#mAckControl.InitData[1].Req oManualToolIn :=#mAckControl.InitData[1].ManualToolIn oManualToolOut :=#mAckControl.InitData[1].ManualToolOut	

5.5.10 FB841: TM_NCServoMag_Manual（NC 伺服刀库手动控制）

程序功能

NC 伺服轴控制的刀库，通过按键手动控制刀库正反转，输出目标到位和启动信号，与 FB842 或 FB843 NC 伺服刀库控制程序配合使用

形式参数说明

信号	I/O	类型	含义
sPlcAxisNo	I	Int	刀库轴的轴号（做 PLC 轴控制）
sMagLocNums	I	Bool	刀位数量
iKeyCW	I	Bool	刀库正转按键
iKeyCCW	I	Bool	刀库反转按键
iReset	I	Bool	复位
iMagInPos	I	Bool	刀库旋转到位信号
iMagDebug	I	Bool	刀库调试模式，激活后不按刀位运动，可用于零点微调
iMagCurrentPos	I	Int	当前刀库位置
oMagReqPos	O	Int	实际刀库轴位置
oMagStartMov	O	Bool	刀库旋转启动

应用实例

程序	注释
CALL #TM_NCServoMagMan sPlcAxisNo :=5 sMagLocNums :=10 iKeyCW :=#Mag[1].iCW iKeyCCW :=#Mag[1].iCCW iReset :=#Mag[1].iReset iMagInPos :=#Mag[1].oInpos iMagDebug :=false iMagCurrentPos :=#Mag[1].oCurrentPos oMagReqPos := oMagStartMov :=#mMagManualStart[1]	//false: 刀库按刀位点动 true:刀库以轴方式点动微调

5.5.11 FB842: TM_NCServoMag（NC 伺服刀库控制）

程序功能

NC 伺服轴控制的刀库，内部根据刀位数量计算每一个刀位的角度，通过 FC18 进行定位。

形式参数说明

信号	I/O	类型	含义
sPlcAxisNo	I	Int	刀库轴的轴号（做 PLC 轴控制）
sMagLocNums	I	Bool	刀位数量
iPosSpeed	I	Bool	定位速度
iReset	I	Bool	复位
iStartMove	I	Bool	请求刀库移动
iReqPos	I	Bool	请求刀库移动位置
iOffset	I	Int	刀库偏置
oCurrentPos	O	Bool	当前刀库位置
oInPos	O	Bool	刀库移动刀位
ActPlcAxisPos	IO	Real	实际刀库轴位置

应用实例

程序	注释
CALL #mServoMag sPlcAxisNo :=5 sMagLocNums :=10 iPosSpeed :=1000.0 iReset :=#Mag[1].iReset iStartMove :=#Mag[1].iStartMov iReqPos :=#Mag[1].iReqPos iOffset :=#Mag[1].iOffset oCurrentPos :=#Mag[1].oCurrentPos oInPos :=#Mag[1].oInpos ActPlcAxisPos :=#mAxisCurrentPos	

5.5.12 FB843: TM_NCServoMag_Bero_Offset（NC 伺服刀库控制扩展）

程序功能

NC 伺服轴控制的刀库，内部根据刀位数量计算每一个刀位的角度，通过 FC18 进行定位。

与 FB842 不同：

1) 增加了第 1 装载点根据某个轴位置偏移相应距离的功能。可通过在 iOffsetValue 管脚输入 0 值屏蔽。
2) 增加了第 1 装载点到达指定到位后以正向继续移动，直至触发外部接近开关的反馈信号，用于精确定位。可通过 iBeroExist 管脚输入 false 值屏蔽

形式参数说明

信号	I/O	类型	含义
sPlcAxisNo	I	Int	刀库轴的轴号（做 PLC 轴控制）
sMagLocNums	I	Bool	刀位数量
iPosSpeed	I	Bool	定位速度
iOffsetValue	I	Real	装载点 1 偏移量输入（外部轴坐标）
iDirection	I	Bool	偏移方向与外部轴运动方向关系 1：同向 0：反向
iBeroExist	I	Bool	是否有外部接近开关
iBeroSignal	I	Bool	外部接近开关信号
iReset	I	Bool	复位
iStartMove	I	Bool	请求刀库移动
iReqPos	I	Bool	请求刀库移动位置
iOffset	I	Int	刀库偏置
oCurrentPos	O	Bool	当前刀库位置
oInPos	O	Bool	刀库移动刀位
CalcPlcAxisPos	IO	Real	PLC 计算位置，用于计算当前刀位（输出值作参考）
ActPlcAxisPos	IO	Real	实际刀库轴位置（含偏移坐标叠加）

应用实例

程序	注释
CALL #mServoMagBeroOffset	
sPlcAxisNo :=5	
sMagLocNums :=10	
iPosSpeed :=1000.0	
iOffsetValue :=#mAxisCurrentPos[2]	//外部偏移坐标（如 W 轴实际位置）
iDirection :=true	//偏移方向与外部偏移值同向
iBeroExist :=1	//有接近开关精定位
iBeroSignal :="I33.0"	//接近开关坐标
iReset :=#Mag[1].iReset	
iStartMove :=#Mag[1].iStartMov	
iReqPos :=#Mag[1].iReqPos	
iOffset :=#Mag[1].iOffset	
oCurrentPos :=#Mag[1].oCurrentPos	
oInPos :=#Mag[1].oInpos	
CalcPlcAxisPos :=#mServoMagCalcPos	
ActPlcAxisPos :=#mAxisCurrentPos[1]	

5.5.13 FB844: TM_BufferMag_Ack（DB72 外部待机位刀库 T/M 指令应答）

程序功能

外部待机位，即备刀在机械手上的刀库 T/M 代码应答。包含 TCI 指令应答，将机械手上的刀还至刀库。

运行时，刀具管理根据换刀流程输出 oTCIStep，oTStep 和 oMStep 步骤号，每个步骤完成后，用户需要在 PLC 程序中置位对应的 iTCI，iT 和 iM 的索引位。详细步骤请参考下面的“执行流程“部分内容。

关于 iGr1GetNewTool 信号，有如下三种设定方式：

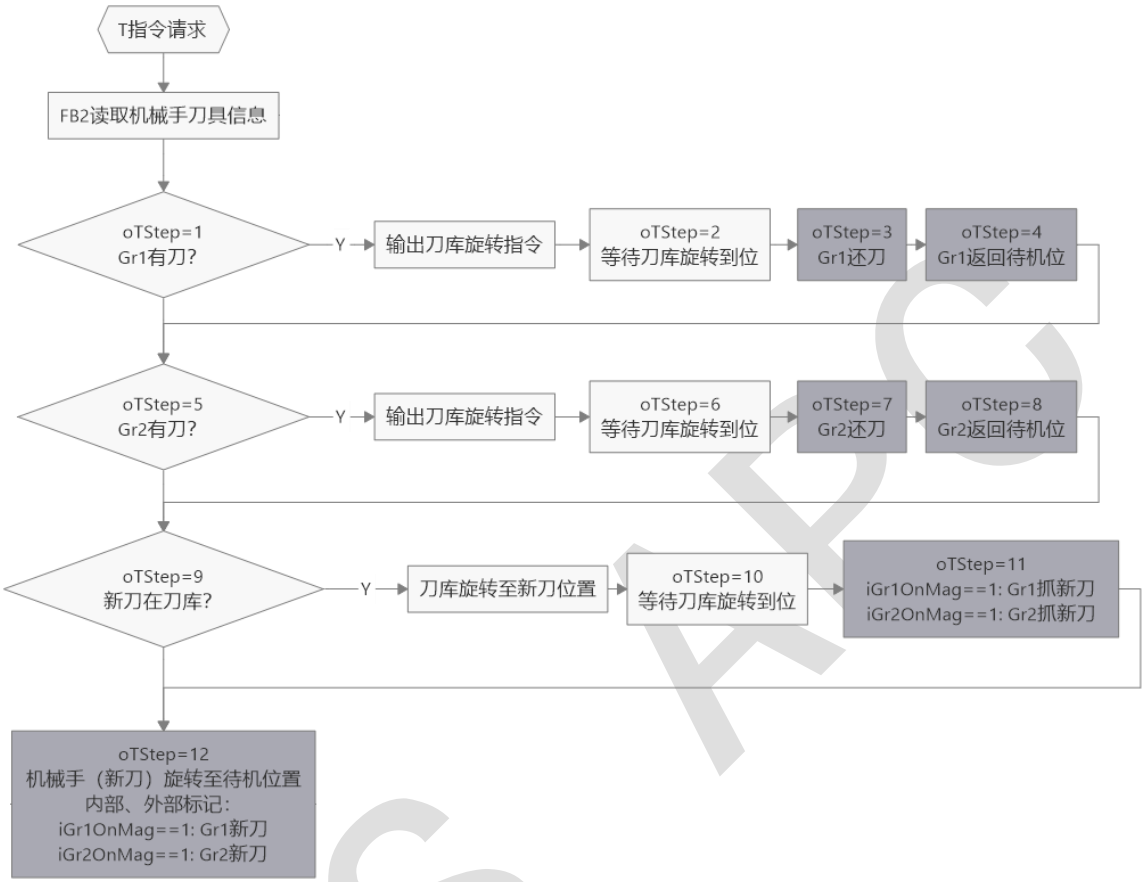
- 设置常 1：始终强制 Gr1 抓取新刀，除非新刀已经在 Gr2 上。
- 设置常 0：始终强制 Gr2 抓取新刀，除非新刀已经在 Gr1 上。
- 通过外部输入信号判断 Gr1 在刀库侧：信号为 1，Gr1 抓新刀；信号为 0，Gr2 抓取新刀。

形式参数说明

信号	I/O	类型	取值范围	含义
TCI_Enable	I			1：执行 TCI 代码应答　　0：无效
T_Enable	I			1：执行 T 代码应答　　0：无效
M_Enable	I			1：执行 M 代码应答　　0：无效
sIFNo	I	Int		检测主轴号
sTOANo	I	Int		刀具表号
sOffset	I	Int		装刀点偏置
sSp1LocNo	I	Int		主轴刀位号
sGr1LocNo	I	Int		机械手 1 刀位号
sGr2LocNo	I	Int		机械手 2 刀位号
iGr1GetNewTool	I	Bool		备刀时 1：Gr1 抓新刀　0：Gr2 抓新刀
iT	I	Array[1..16] of Bool		T 指令应答
iM	I	Array[1..16] of Bool		M 指令应答
iTCI	I	Array[1..16] of Bool		TCI 指令应答
oTCIStep	O	Int		TCI 指令应答步骤
oTStep	O	Int		T 指令应答步骤
oMStep	O	Int		M 指令应答步骤
oMagNo	O	Int		当前刀库号
oMagStartMov	O	Bool		启动刀库旋转
oMagReqPos	O	Int		刀库旋转位置
oMagOffset	O	Int		刀库偏置
oNewToolInGr1	O	Bool		换刀时：1：Gr2 抓旧刀

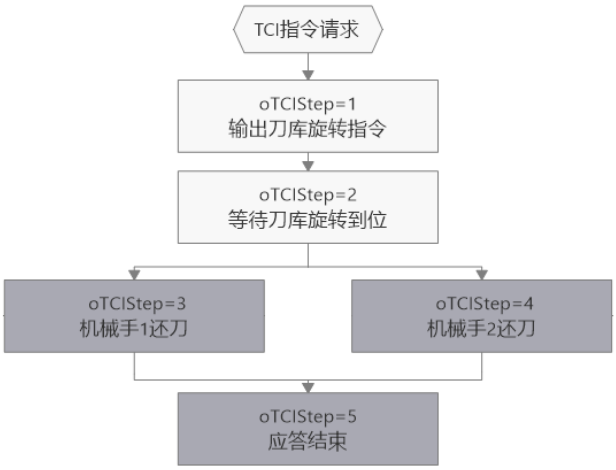
执行流程

- T 指令应答



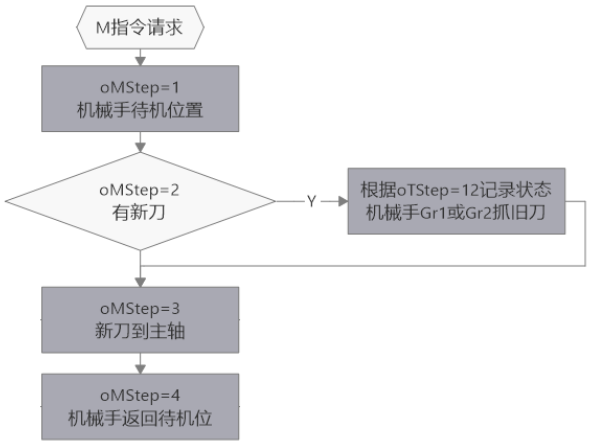
注：白色 Step 会自动应答，深色 Step 需要外部程序应答。如 oTStep=3 为深色，需要 iT[3]=1 应答，否则停留在第 3 步上。

- TCI 指令应答



注：白色 Step 会自动应答，深色 Step 需要外部程序应答。如 oTCIStep=3 为深色，需要 iTCI[3]=1 应答，否则停留在第 3 步上。

- M 指令应答



注：白色 Step 会自动应答，深色 Step 需要外部程序应答。如 oMStep=3 步为深色，需要 iM[3]=1 应答，否则停留在第 3 步上。

应用实例

程序		注释
CALL #mMagBuf_Sp1		
TCI_Enable	:=True	//TCI 控制信号始终为 True
T_Enable	:=#mAckControl.oRules[1]	//oRules[1]控制 1 号刀库 T 应答
M_Enable	:=#mAckControl.oRules[3]	//oRules[3]控制 1 号刀库 M 应答
sIFNo	:=1	
sTOANo	:=1	
sMagNo	:=1	//当前刀库号 1
sOffset	:=0	
sSpLocNo	:=1	
sGr1LocNo	:=2	
sGr2LocNo	:=3	
iMagInPos	:=#Mag[1].oInpos	
iGr1GetNewTool	:=#Sp[1].iGr1GetNewTool	
iT	:=#Sp[1].iT	//T 指令应答位 Array
iM	:=#Sp[1].iM	//M 指令应答位 Array
iTCI	:=#Sp[1].iTCI	//TCI 指令应答位 Array
oTCIStep	:=#Sp[1].oMag[1].oTCIStep	//TCI 步骤号输出
oTStep	:=#Sp[1].oMag[1].oTStep	//T 步骤号输出
oMStep	:=#Sp[1].oMag[1].oMStep	//M 步骤号输出
oMagNo	:=#Sp[1].oMag[1].oMagNo	
oMagStartMov	:=#Sp[1].oMag[1].oMagStartMov	
oMagReqPos	:=#Sp[1].oMag[1].oMagReqPos	
oMagOffset	:=#Sp[1].oMag[1].oMagOffset	
oNewToolInGr	:=#Sp[1].oMag[1].oNewToolInGr	

5.5.14 FB845: TM_Turret_Ack（DB73 刀塔应答）

程序功能

刀塔式刀库应答，支持装载、卸载、T 代码换刀等功能。该程序用于立加刀塔式刀库，车床刀塔需做特殊处理。

形式参数说明

信号	I/O	类型	取值范围	含义
sSpIFNo	I	Int		检测主轴号
sLdIFNo	I	Int		装载点号
sChanNo	I	Int		通道号
sMagNo	I	Int		刀库号
sSpLocNo	I	Int		主轴刀位号
iReset	I	Bool		复位
iAck	I	Bool		应答
iTool2Mag	I	Bool		异步刷新，主轴->刀库
iTool2Sp	I	Bool		异步刷新，刀库->主轴
iMagCurrentPos	I	Int		当前刀库位置
oMagStartMov	IO	Bool		请求刀库旋转
oMagReqPos	IO	Int		请求刀库旋转位置
oToolDone	IO	Bool		异步刷新结束

应用实例

程序	注释
CALL #mFB845	
sSpIFNo :=1	
sLdIFNo :=1	
sChanNo :=1	
sMagNo :=1	
sSpLocNo :=1	
iReset :=#Turret[1].iReset	
iAck :=#Turret[1].iAck	//装载应答
iTool2Mag :=	//刀具到刀库应答
iTool2Sp :=	//刀具到主轴应答
iMagCurrentPos :=#Mag[1].oCurrentPos	
oMagStartMov :=#Turret[1].oMagStartMov	
oMagReqPos :=#Turret[1].oMagReqPos	
oToolDone :=	

5.5.15 FB846: TM_HydMag（液压刀库控制）

程序功能

链式刀库控制程序，支持刀库手动、自动、回零、指令旋转等功能。

形式参数说明

信号	I/O	类型	取值范围	含义
sMagLocNums	I	Int		刀库刀位数量
sMagCounter	I	Int		刀库计数器号
iFirstScan	I	Bool		上电脉冲
iReset	I	Bool		复位
iRef	I	Bool		回零
iMagCount	I	Bool		刀库计数开关
iManCW	I	Bool		手动刀库正转
iManCCW	I	Bool		手动刀库反转
iStartMove	I	Bool		请求刀库移动
iReqPos	I	Int		请求刀库移动位置
iOffset	I	Int		刀库偏置
oCurrentPos	O	Int		当前刀库位置
oDiff	O	Int		移动中，距离目标刀位位置
oCW	O	Bool		刀库正转输出
oCCW	O	Bool		刀库反转输出
oInPos	O	Bool		刀库移动到位

应用实例

程序	注释

SET
R #Mag[1].iStartMov
 	
A #Turret[1].oMagStartMov
JCN P001
S #Mag[1].iStartMov
L #Turret[1].oMagReqPos
T #Mag[1].iReqPos
L 0
T #Mag[1].iOffset
P001: NOP 0
CALL #mHyMag
	//预置刀库启动 OFF
	
	//设置刀库启动条件

sMagLocNums	:=10	
sMagCounter	:="Mag1Counter"	
iFirstScan	:="GlobalDB".FirstScan	
iReset	:=#Mag[1].iReset	
iRef	:=#Mag[1].iRef	
iMagCount	:=#Mag[1].iCount	
iManCW	:=#Mag[1].iCW	
iManCCW	:=#Mag[1].iCCW	
iStartMove	:=#Mag[1].iStartMov	
iReqPos	:=#Mag[1].iReqPos	
iOffset	:=#Mag[1].iOffset	
oCurrentPos	:=#Mag[1].oCurrentPos	
oDiff	:=#Mag[1].oDiff	
oCW	:=#Mag[1].oCW	
oCCW	:=#Mag[1].oCCW	
oInPos	:=#Mag[1].oInpos	

5.5.16 FB847: TM_3rdServoTurret（第三方伺服刀塔控制）

程序功能

控制第三方伺服刀塔，提供相关接口，具体刀库动作流程在程序块内编写，本程序只提供刀库管理的应答。

形式参数说明

信号	I/O	类型	信号分类	含义
sTurretLocNums	I	Int	刀塔应答交互信号（输入）	刀库刀位数量
iReset	I	Bool		复位
iManCW	I	Bool		手动刀塔正转
iManCCW	I	Bool		手动刀塔反转
iTurretStart	I	Bool		外部输入刀塔启动信号（自动换刀）
iToolDone	I	Bool		刀塔应答完成
sTurretReqPos	I	Int		外部输入刀塔目标刀位（自动换刀）
iClamp	I	Bool	刀塔伺服交互信号（输入）	刀盘夹紧
iUnclamp	I	Bool		刀盘松开
iInPos	I	Bool		刀盘旋转到位
iServoAlarm	I	Bool		刀塔伺服报警
sTurretActPos	I	Int		刀塔实际位置
sPwrSpAxisNo	I	Int		动力头主轴轴号（0=无动力头）
iPwrSpLock	I	Bool		动力头主轴锁定
iPwrSpUnlock	I	Bool		动力头主轴释放

iPwrSpRef	I	Bool		动力头主轴零点
oMagStartMov	O	Bool	刀塔应答交互信号（输出）	刀库旋转
oAckTool2Mag	O	Bool		刀具到刀库应答（手动换刀）
oAckTool2Sp	O	Bool		刀具到主轴应答（手动换刀）
oCurrentPos	O	Int		当前刀位（用于刀塔应答）
oClamp	O	Bool	刀塔伺服交互信号（输出）	刀盘夹紧
oUnclamp	O	Bool		刀盘松开
oStart	O	Bool		启动
oMode	O	Array[0..7] of Bool		工作模式设定，位数组
oTargetPos	O	Int		目标刀位

应用实例

程序	注释
CALL #mServoTurret	
sTurretLocNums :=12	
iReset :=#Turret[1].iReset	
iManCW :=#Turret[1].iCW	
iManCCW :=#Turret[1].iCCW	
iTurretStart :=#Turret[1].oMagStartMov	
iToolDone :=#mTurretAck.oToolDone	//来自刀塔应答程序 oToolDone
sTurretReqPos :=#Turret[1].oMagReqPos	
iClamp :=	
iUnClamp :=	
iInPos :=	
iServoAlarm :=	
sTurretActPos :="SimActPos"	//模拟信号，实际不需要
sPwrSpAxisNo :=	
iPwrSpLock :=	
iPwrSpUnlock :=	
iPwrSpRef :=	
oMagStartMov :=	
oAckTool2Mag :=#Turret[1].AckTool2Mag	
oAckTool2Sp :=#Turret[1].AckTool2Sp	
oCurrentPos :=#Turret[1].oCurrentPos	
oClamp :=	
oUnclamp :=	
oStart :=	
oMode :=	//模式为位数组类型变量 Bit0..Bit7
oTargetPos :="SimTargetPos"	//模拟信号，实际不需要

5.5.17 FB849: TM_AckControl（T/M 代码应答管理）

程序功能

控制 T/M 应答功能块，例：一个主轴配置 2 个不同的刀库，就需要控制那个功能块应答。oRules 规则总共可以设定 16 条，用于控制后续 T/M 应答功能块是否有效。目前程序中使用了 4 条规则如下：

- oRules[1] / oRules[2]:刀库 1 / 2 的 T 代码应答控制
- oRules[3] / oRules[4]:刀库 1 / 2 的 M 代码应答控制

形式参数说明

信号	I/O	类型	取值范围	含义
InitData	I	结构体		信号与 TM_init 输出信号对接
oRules	O	Array[1..16] of Bool		规则

输入信号结构

信号	类型
Req	Bool
ManualToolIn	Bool
ManualToolOut	Bool
NewToolMagNo	Int
OldToolMagNo	Int

应用实例

程序	注释
CALL #mAckControl InitData :=#mAckControl.InitData oRules :=	//TM_init 输出的接口信号

内部逻辑编程基本原则：

- 备刀（新刀）的来源刀库号控制 T 代码应答
- 主轴刀具（旧刀）归还的刀库号控制 M 代码应答

5.5.18 矩阵式刀库应答

为配合刀库指令，需仿真矩阵刀库应答

如例：MAG2 为矩阵式刀库，应答装载点 1 和主轴 1 的刀库旋转指令。

程序	注释
//预置刀库启动 OFF SET R #Mag[2].iStartMov	

```
      S     #Mag[2].oInpos
    //初始化刀库当前位置
      A(
      L     #Mag[2].oCurrentPos
      L     0
      ==I
      )
      JCN   M003
      L     1
      T     #Mag[2].oCurrentPos
    M003: NOP 0

      A     #Load[1].oMag[2].oMagStartMov
      JCN   M001
      L     #Load[1].oMag[2].oMagReqPos
      T     #Mag[2].oCurrentPos
    M001: NOP 0
      A     #Sp[1].oMag[2].oMagStartMov
      JCN   M002
      L     #Sp[1].oMag[2].oMagReqPos
      T     #Mag[2].oCurrentPos
    M002: NOP 0
```

5.6 机床面板控制（编号：850~859）

5.6.1 FC850: OP_NcStart_Ext（标准面板增加程序启停键）

程序功能

此程序块不但具有标准铣床（FC19）面板功能，还可以为控制面板额外增加“NC 启动”和“NC 停止”键，并且同时把机床控制面板上的“NC 启动”和“NC 停止”灯的状态传输至增加按键开关的指示灯。

如果用户配置其它工艺类型的面板，根据需要在 FC850 中调用标准面板程序 FC24/FC25/FC26。

形式参数说明

信号	I/O	类型	取值范围	含义
BAG_No	I	BYTE	1-10	机床控制面板方式组
Chan_No	I	BYTE	1-10	机床控制面板通道号
SpindleIF_No	I	BYTE	1-31	主轴轴号
Additional_NCStart	I	BOOL		增加的程序启动键，常开
Additional_NCStop	I	BOOL		增加的程序停止键，常闭
FeedHold	IO	BOOL		通道进给保持状态
SpindleHold	IO	BOOL		通道主轴保持状态
Additional_NCStart_light	IO	BOOL		程序启动键指示灯
Additional_NCStop_light	IO	BOOL		程序停止键指示灯

应用实例

为机床配置 MCP 483C PN 面板为铣床面板，同时增加外部程序启停按键。

程序		注释
CALL “OP_NcStart_Ext”		//调用 FC850
BAG_No	:=B#16#1	//控制面板方式组
Chan_No	:=B#16#1	//控制面板所属通道
SpindleIF_No	:=B#16#4	//主轴轴号
Additional_NCStart	:= I401.0	//用户自定义程序启动键
Additional_NCStop	:= I401.1	//用户自定义程序停止键
FeedHold	:=M100.0	//进给轴保持状态输出
SpindleHold	:=M100.1	//主轴保持状态输出
Additional_NCStart_light	:=Q401.0	//用户自定义程序启动键指示灯
Additional_NCStop_light	:=Q401.1	//用户自定义程序启动键指示灯

5.6.2 FC851: OP_MINI_BHG（Mini 手持单元）

程序功能

程序块 FC851 分析 mini 手持单元的信号并将其传递到轴接口信号，对轴 DB 或通道 DB 中的接口信号进行了控制。手持单元通过手轮左侧使能按键激活，一旦激活手持就处于活动状态。通过手持单元上的轴选择开关选择相应的轴(选择轴的状态与 MCP 面板上的轴选择区轴状态同步)，然后选择功能键 F1、F2 或 F3 激活相应的增量信号 IN1、IN10 或 IN100。此时转动 MINI 手持单元上的手轮就可以移动轴，如果需要连续运行，可以通过方向键“+”和方向键“-”连续运行轴。Mini 手持单元最多激活 5 个轴，默认激活 MCP 面板上的 X/Y/Z/4/5，如果与默认值不同，可以通过修改源程序实现，例如，如果要选择激活 X/Y/Z/5/6。程序如下：

源程序 Network3

```
//手持选择轴1:Z
      AN    #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      A     #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      AN    #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_Z          #SEL_AXIS_Z

//手持选择轴2:X
      AN    #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      A     #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_X          #SEL_AXIS_X

//手持选择轴3:Y
      A     #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      A     #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_Y          #SEL_AXIS_Y

//手持选择轴4
      A     #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      AN    #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_4          #SEL_AXIS_4

//手持选择轴5
      AN    #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      AN    #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_5          #SEL_AXIS_5

//Clear no used axis
      CLR
      =     #SEL_AXIS_6          #SEL_AXIS_6
      =     #SEL_AXIS_7          #SEL_AXIS_7
      =     #SEL_AXIS_8          #SEL_AXIS_8
      =     #SEL_AXIS_9          #SEL_AXIS_9
```

修改后的程序 Network3

```
//手持选择轴1:Z
      AN    #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      A     #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      AN    #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_Z          #SEL_AXIS_Z

//手持选择轴2:X
      AN    #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      A     #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_X          #SEL_AXIS_X

//手持选择轴3:Y
      A     #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      A     #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_Y          #SEL_AXIS_Y

//手持选择轴4
      A     #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      AN    #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_5          #SEL_AXIS_5

//手持选择轴5
      AN    #Axis_bit_0          #Axis_bit_0      — axis choise bit 2
      AN    #Axis_bit_1          #Axis_bit_1      — axis choise bit 1
      A     #Axis_bit_2          #Axis_bit_2      — axis choise bit 0
      =     #SEL_AXIS_6          #SEL_AXIS_6

//Clear no used axis
      CLR
      =     #SEL_AXIS_4          #SEL_AXIS_4
      =     #SEL_AXIS_7          #SEL_AXIS_7
      =     #SEL_AXIS_8          #SEL_AXIS_8
      =     #SEL_AXIS_9          #SEL_AXIS_9
```

由于程序中一些控制信号是改写的，所以**程序块调用必须在控制面板（MCP）、ShopTurn 、 ShopMill 之前。**注意：当操作手持期间，方式组的操作方式禁止转换。

以下报警需要预先配置，并且可以通过返回值分析错误:

1) 手持单元未连接报警。
2) 手持单元连接错误报警。

激活 MINI 手持单元的基本要求

1) Enable 位为真。
2) 在控制面板进进给率置于非零状态。
3) 所有轴使能信号必须是激活的。

形式参数的说明

信号	I/O	类型	取值范围	含义
Enable	I	BOOL		手持激活
Chan_No	I	INT	1-10	手持所属通道号
MCP_No	I	INT	1-2	手持所属控制面板号
F1_KEY	I	BOOL		手持单元 F1 功能按键
F2_KEY	I	BOOL		手持单元 F2 功能按键

F3_KEY	I	BOOL		手持单元 F3 功能按键
Axis_bit_0	I	BOOL		通道中轴的编码位 0
Axis_bit_1	I	BOOL		通道中轴的编码位 1
Axis_bit_2	I	BOOL		通道中轴的编码位 2
Rapid_traverse_key	I	BOOL		手持单元快速运行按键
Plus_key	I	BOOL		手持单元方向按键“+”
Minus_key	I	BOOL		手持单元方向按键“-”
Alarm_1	IO	BOOL		手持单元未连接提示
Alarm_2	IO	BOOL		手持单元连接错误报警
Ref_DW	IO	DWORD		手持单元信号沿锁存地址

特殊说明

连接手持单元后，需要在通用机床数据 MD11350、MD11351、MD11352 中设置 mini 手持单元参数，否则手轮无效。并且可以通过 DB10.DBB68/DB10.DBB69/DB10.DBB70 查看手脉信号状态。

应用实例：配置常用 mini 手持单元

小型手持设备连接至 MCP 483C PN 的插口 X51、X52、X55 和 X60 上。

下表说明了小型手持设备的机床功能编码方式和用户键的连接。

选轴开关 X51 的编码：

2^1 X51.1 KT-IN1	2^2 X51.2 KT-IN2	2^3 X51.3 KT-IN3	开关位置	功能
0	0	0	-	未连接小型手持设备
1	1	0	0	未选择任何轴
0	1	0	Z	选择轴 Z
0	1	1	X	选择轴 X
1	1	1	Y	选择轴 Y
1	0	1	4	选择轴 4
0	0	1	5	选择轴 5

运行键/快移键 X52 的编码：

	引脚	按键	功能
KT-IN4	X52.1	+	运行键方向 +
KT-IN5	X52.2	-	运行键方向 -
KT-IN6	X52.3	∿	快移键

功能键 X55 的编码：

	引脚	按键	功能
KT-IN7	X55.1	F1	功能键
KT-IN8	X55.2	F2	
KT-IN9	X55.3	F3	

对应的 MCP 地址:

EB n + 10	KT-IN8 X55.2	KT-IN7 X55.1	KT-IN6 X52.3	KT-IN5 X52.2	KT-IN4 X52.1	KT-IN3 X51.3	KT-IN2 X51.2	KT-IN1 X51.1
EB n + 11	-	-	-	-	-	-	-	KT-IN9 X55.3
EB n + 12	-	-	-	-	-	-	-	-

应用实例

程序		注释
CALL “OP_MINI_BHG”		//调用 FC851
Enable	:=I401.7	//手持单元左侧使能键 PLC 地址
Chan_No	:=1	//手持单元所属通道号
MCP_No	:=1	//手持单元所属面板号
F1_KEY	:=I10.6	//手持单元 F1
F2_KEY	:=I10.7	//手持单元 F2
F3_KEY	:=I10.0	//手持单元 F3
Axis_bit_0	:=I10.1	//手持单元轴选编码位 0
Axis_bit_1	:=I10.2	//手持单元轴选编码位 1
Axis_bit_2	:=I10.3	//手持单元轴选编码位 2
Rapid_traverse_key	:=I10.5	//手持单元快速移动键
Plus_key	:=I11.3	//手持单元方向键+
Minus_key	:=I11.4	//手持单元方向键-
Alarm_1	:=DB2.DBX184.0	//手持单元未连接报警
Alarm_2	:= DB2.DBX180.0	//手持单元连接错误报警
Ref	:=MD900	//手持单元按键沿信号所存区

5.6.3 FC852: OP_3rd_BHG（第三方手轮）

程序功能

控制第三方接口手轮

形式参数说明

信号	I/O	类型	含义
EnableKey	I	Bool	手轮使能
Chan_No	I	Bool	通道号
MCP_No	I	Bool	MCP 号
Ax1	I	Bool	第 1 轴
Ax2	I	Bool	第 2 轴
Ax3	I	Bool	第 3 轴
Ax4	I	Bool	第 4 轴

Ax5	I	Bool	第 5 轴
x1	I	Bool	当量：1um
x10	I	Bool	当量：10um
x100	I	Bool	当量：100um
HandWheelActive	IO	Bool	手轮激活
NoAxisSelect	IO	Bool	未选轴
Ref	IO	Dword	存储

应用实例

必须在在 MCP 面板控制程序之前调用

程序	注释
CALL "OP_3rd_BHG" EnableKey :=True Chan_No :=1 MCP_No :=1 Ax1 :="GlobalDB".HandWhell.BHG.Ax1 Ax2 :="GlobalDB".HandWhell.BHG.Ax2 Ax3 :="GlobalDB".HandWhell.BHG.Ax3 Ax4 :="GlobalDB".HandWhell.BHG.Ax4 Ax5 :="GlobalDB".HandWhell.BHG.Ax5 x1 :="GlobalDB".HandWhell.BHG.x1 x10 :="GlobalDB".HandWhell.BHG.X10 x100 :="GlobalDB".HandWhell.BHG.X100 HandWheelActive :="GlobalDB".HandWhell.BHG.Active NoAxisSelect :="GlobalDB".HandWhell.BHG.NoAxisSelect Ref :="GlobalDB".HandWhell.BHG.mRef	//调用 FC852

5.6.4 FC853: OP_OverrideEnable（倍率控制：通道和轴独立控制）

程序功能

此功能块用于通道和轴的进给倍率控制，可以根据输入的通道号激活通道和通道轴（程序会自动检测指定通道内轴的进给倍率）的进给倍率。

注意：此程序块默认激活通道轴 1. 修改“ChanEnable”与“AxisEnable”参数后需要重启系统参数才能生效。

形式参数说明

信号	I/O	类型	含义
ChanEnable	I	BOOL	通道倍率使能
ChannelNo	I	INT	通道号：1...10
AxisEnable	I	BOOL	轴倍率使能
Error	O	BOOL	通道号无效

应用实例

程序	注释

CALL "OP_OverrideEnable"
ChanEnable :=True
//"GlobalDB".MCP.OverrideEnable.ChanEnable
ChannelNo :=1
//"GlobalDB".MCP.OverrideEnable.ChannelNo
AxisEnable :=TRUE
//"GlobalDB".MCP.OverrideEnable.AxisEnable
Error :="GlobalDB".MCP.OverrideEnable.Error
	//调用 FC853
	//激活通道进给倍率使能
	//通道号
	//轴进给倍率使能
	//通道号无效报错

5.6.5 FC854: OP_OverrideEnableAll（倍率控制：通道和轴一体控制）

程序功能

此功能块用于通道和轴的进给倍率控制，调用此程序块后程序会自动检测激活的通道与轴，只要把使能信号置位为1，所有激活通道和激活轴的倍率信号均被激活。

注意：此程序块默认激活通道 1 和轴 1，修改“Enable”参数后需要重启系统参数才能生效。

形式参数说明

信号	I/O	类型	含义
Enable	I	BOOL	所有通道与轴的进给倍率使能

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " OP_OverrideEnableAll" Enable :=True //"GlobalDB".MCP.OverrideEnableAll.Enable	//调用 FC854 //激活通道和轴的进给倍率

5.6.6 FC855: OP_ToggleMcsWcs_Ext（MCS<->WCS 坐标系切换）

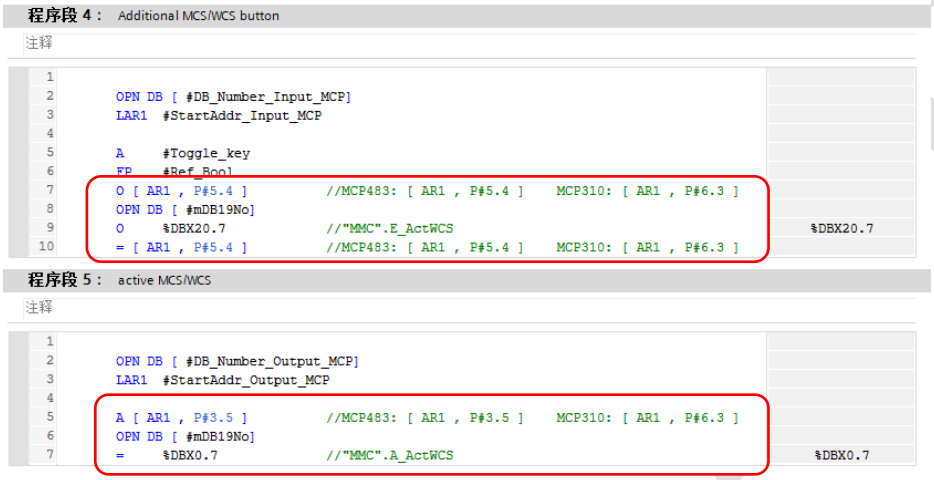
程序功能

通过用户自定义键”和“操作面板软键”切换 WCS&MCS。

用户可以通过自定义按钮切换 WCS&MCS，LED 亮表明设备处于 WCS，灭表示设备处于 MCS。并且还可以通过屏幕上的“实际 MCS”软键切换 WCS&MCS。两种切换方式可以兼容，互不影响。

注意事项

1) 程序调用位置 FC19 之前要求。
2) 本程序默认适用于 MCP483，与 FC19 配套使用。如果使用 MCP310，请参考程序内的说明修改 MCS/WCS 按键和指示灯地址



形式参数说明

信号	I/O	类型	取值范围	含义
MCP_No	I	BOOL	1-2	面板号
Toggle_key	I	BOOL		WCS/MCS 切换键：0:MCS;1:WCS
RefBit	IO	BOOL		上升沿锁存位

应用实例

实例功能：通过机床控制面板 I401.0 键切换 MCS 与 WCS。

程序		注释
CALL “OP_ToggleMcsWcs_Ext”		//调用 FC855
MCP_No	:=1	//控制面板号
Toggle_key	:=I401.0	//坐标切换键
RefBit	:=M101.0	//信号锁存位

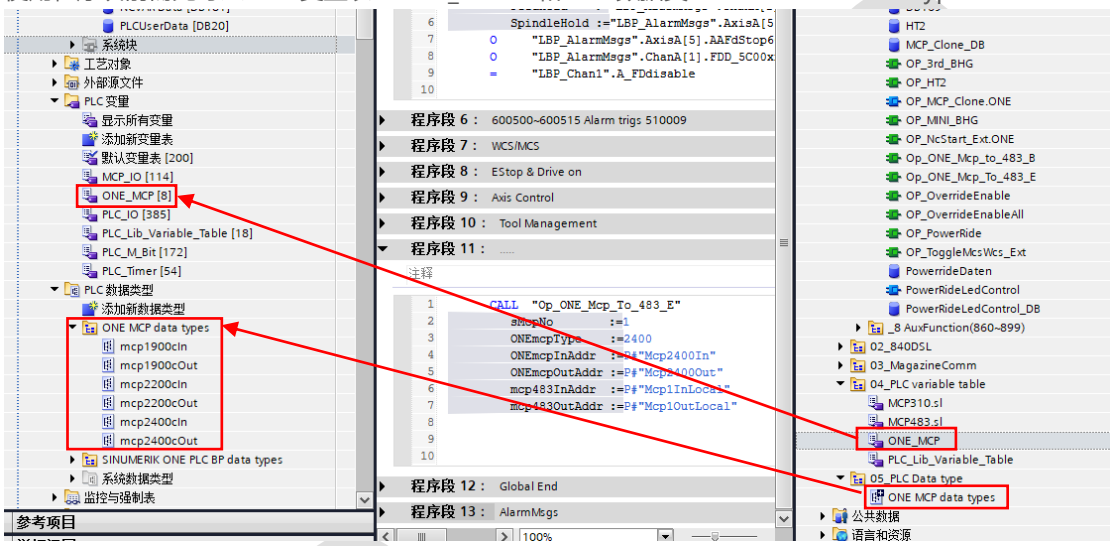
5.6.7 FC856: OP_ONE_MCP_TO_483_B（ONE MCP 地址映射开始）

程序功能

当使用新型的 SINUMERIK ONE MCP 时，其 IO 地址顺序与标准的 MCP483/MCP398 不同，可以使用本程序块将 ONE MCP 的 IO 地址映射为一个内部虚拟的 MCP483。这样在调试 MCP 时还可以继续使用原来的 MCP 符号地址及 FC19 功能块，能够大大简化 ONE MCP 的 PLC 编程工作。同时，还提供了新型 Powerride 倍率开关的相关程序。

注意事项

1) 使用程序块前需先导入 PLC 变量表“ONE_MCP”和 PLC 数据类型“ONE MCP data types”



2) 根据实际使用的 ONE MCP 类型和 IO 地址修改 PLC 变量表 ONE_MCP 的内容，例如：使用 MCP2400 时

ONE_MCP

	名称	数据类型	地址	保持	从 H...	从 H...	在 H...	监控
1	Mcp2400In	"mcp2400cIn"	%I600.0		☑	☑	☑	
2	Mcp2400Out	"mcp2400cOut"	%Q600.0		☑	☑	☑	
3	Mcp483LocalIn	"LBP_typeMcp483MillingInput"	%I500.0		☑	☑	☑	
4	Mcp483LocalOut	"LBP_typeMcp483MillingOutput"	%Q500.0		☑	☑	☑	
5	<新增>				☑	☑	☑	

其中 MCP2400 的 IO 地址 I600.0 和 Q600.0 为真实的地址，与 OB100 中的一致；MCP483Local 的 IO 地址 I500.0 和 Q500.0 应使用未被实际占用的 IO 地址区域，以免产生地址冲突。

3) FC856: OP_ONE_MCP_TO_483_B 必须与 FC857: OP_ONE_MCP_TO_483_E 成对使用
4) FC856: OP_ONE_MCP_TO_483_B 在 FC19 之前调用
5) FC857: OP_ONE_MCP_TO_483_E 在 FC19 及需要使用到 MCP483 按键地址的程序块之后调用
6) FC858: OP_PowerRide 在 FC856: OP_ONE_MCP_TO_483_B 之前调用
7) 对于 MCP2400，包含由两个用户自定义按键区 OEM1 和 OEM2，本程序中仅做了 OEM1 区域的地址映射，OEM2 区域的按键由用户自行编程处理

下图为 OB1 编程举例：

```
程序段 4： ONE MCP
注释
    CALL  "OP_PowerRide"
       ONEmcpType :=2400

    CALL  "Op_ONE_Mcp_to_483_B"
       sMcpNo              :=1
       ONEmcpType          :=2400
       ONEmcpInAddr        :=P#"Mcp2400In"
       ONEmcpOutAddr       :=P#"Mcp2400Out"
       mcp483InAddr        :=P#"Mcp1InLocal"
       mcp483OutAddr       :=P#"Mcp1OutLocal"
       feedOverRideGrayCode     :="PowerrideDaten".feedOverRideGrayCode
       spindleOverRideGrayCode :="PowerrideDaten".spindleOverRideGrayCode

程序段 5： MCP
注释
    CALL  "LBP_MCPCtrlMilling"
       BAGNo       :=BYTE#16#01
       ChanNo      :=BYTE#16#01
       SpindleIFNo :=BYTE#16#05
       FeedHold    :="LBP_AlarmMsgs".ChanA[1].FDD_5C00xx[8]
       SpindleHold :="LBP_AlarmMsgs".AxisA[5].AAFdStop60AAxx[8]
    O     "LBP_AlarmMsgs".AxisA[5].AAFdStop60AAxx[8]
    O     "LBP_AlarmMsgs".ChanA[1].FDD_5C00xx[8]
    =     "LBP_Chan1".A_FDdisable

程序段 6： 600500~600515 Alarm trigs 510009
程序段 7： WCS/MCS
程序段 8： EStop & Drive on
程序段 9： Axis Control
程序段 10： Tool Management
程序段 11： .....
注释
    CALL  "Op_ONE_Mcp_To_483_E"
       sMcpNo       :=1
       ONEmcpType   :=2400
       ONEmcpInAddr :=P#"Mcp2400In"
       ONEmcpOutAddr :=P#"Mcp2400Out"
       mcp483InAddr :=P#"Mcp1InLocal"
       mcp483OutAddr :=P#"Mcp1OutLocal"

程序段 12： Global End
程序段 13： AlarmMsgs
```

ONE MCP 的 PowerRide 倍率开关功能，详见 5.6.9 章节

ONE MCP 映射到虚拟 MCP483 开始程序

FC19 MCP483 程序

ONE MCP 映射到虚拟 MCP483 结束程序，详见 5.6.8 章节

形式参数说明

信号	I/O	类型	取值范围	含义
sMcpNo	I	INT	1-2	面板号
ONEmcpType	I	INT	1900 2200 2400	面板类型
ONEmcpInAddr	I	Pointer	ONE_MCP 变量表中的实际面板输入输出地址	ONE MCP 输入地址指针
ONEmcpOutAddr	I	Pointer		ONE MCP 输出地址指针
mcp483InAddr	I	Pointer	ONE_MCP 变量表中的 MCP483Local 输入输出地址	MCP483 输入地址指针
mcp483OutAddr	I	Pointer		MCP483 输出地址指针
feedOverRideGrayCode	I	USInt	FB21 对应的输出管脚	进给倍率格雷码
spindleOverRideGrayCode	I	USint		主轴倍率格雷码

应用实例

程序	注释
CALL "Op_ONE_Mcp_to_483_B"	//调用 FC856
sMcpNo :=1	//控制面板号
ONEmcpType :=2400	//ONE MCP 2400
ONEmcpInAddr :=P#"Mcp2400In"	//ONE_MCP 变量表中真实 MCP 输入地址指针
ONEmcpOutAddr :=P#"Mcp2400Out"	//ONE_MCP 变量表中真实 MCP 输出地址指针
mcp483InAddr :=P#"Mcp483LocalIn"	//ONE_MCP 变量表中虚拟 MCP 输入地址指针
mcp483OutAddr :=P#"Mcp483LocalOut"	//ONE_MCP 变量表中虚拟 MCP 输出地址指针
feedOverRideGrayCode :="PowerrideDaten".feedOverRideGrayCode	//进给倍率格雷码
spindleOverRideGrayCode :="PowerrideDaten".spindleOverRideGrayCode	//主轴倍率格雷码

5.6.8 FC857: OP_ONE_MCP_TO_483_E（ONE MCP 地址映射结束）

程序功能

与 FC856: OP_ONE_MCP_TO_483_B 成对使用，用于将虚拟 MCP483 的输出信号转换为真实 ONE MCP 的输出

形式参数说明

信号	I/O	类型	取值范围	含义
sMcpNo	I	INT	1-2	面板号
ONEmcpType	I	INT	1900 2200 2400	面板类型
ONEmcpInAddr	I	Pointer	ONE_MCP 变量表中的实际面板输入输出地址	ONE MCP 输入地址指针
ONEmcpOutAddr	I	Pointer		ONE MCP 输出地址指针
mcp483InAddr	I	Pointer	ONE_MCP 变量表中的 MCP483Local 输入输出地址	MCP483 输入地址指针
mcp483OutAddr	I	Pointer		MCP483 输出地址指针

应用实例

程序	注释
CALL "Op_ONE_Mcp_to_483_E"	//调用 FC857
sMcpNo :=1	//控制面板号
ONEmcpType :=2400	//ONE MCP 2400
ONEmcpInAddr :=P#"Mcp2400In"	//ONE_MCP 变量表中真实 MCP 输入地址指针
ONEmcpOutAddr :=P#"Mcp2400Out"	//ONE_MCP 变量表中真实 MCP 输出地址指针
mcp483InAddr :=P#"Mcp483LocalIn"	//ONE_MCP 变量表中虚拟 MCP 输入地址指针
mcp483OutAddr :=P#"Mcp483LocalOut"	//ONE_MCP 变量表中虚拟 MCP 输出地址指针

5.6.9 FC858: OP_PowerRide（PowerRide 倍率开关）

程序功能

ONE MCP 上的 PowerRide 倍率开关相关特性程序

形式参数说明

信号	I/O	类型	取值范围	含义
ONEmcpType	I	INT	1900 2200 2400	面板类型

应用实例

程序	注释
CALL "Op_PowerRide" ONEmcpType :=2400	//调用 FC858 //ONE MCP 2400

5.6.10 FB855: OP_PowerRideLedControl（PowerRide 按键）

程序功能

本功能块用于对 Powerride 的按键进行编程，通过该按键可以出发 Cycle Start 从而启动程序。此处仅提供编程示例，具体功能由编程人员根据实际情况编写。

形式参数说明

信号	I/O	类型	含义
OverRidePushButtonLed	IO	USInt	按键指示灯数值
OverRidePushButton	IO	Bool	按键地址
OverRideToPosOne	IO	Bool	将倍率值锁定在位置 1

应用实例

程序	注释

A "Mcp2400In".PowerrideFeed.pushButton
O "Mcp2400In".KeyPadModeGroup.cycleStart
= "Mcp2400In".KeyPadModeGroup.cycleStart
 	
CALL "OP_PowerRideLedControl", "PowerRideLedControl_DB"
OverRidePushButtonLed :="PowerrideDaten".feedOverRidePushButtonLed
OverRidePushButton :="PowerrideDaten".feedOverRidePushButton
OverRideToPosOne :="PowerrideDaten".feedOverRideForcePos1
	//调用 FC857
	//控制面板号
	//ONE MCP 2400
	//ONE_MCP 变量表中真实 MCP 输入地址指针
	//ONE_MCP 变量表中真实 MCP 输出地址指针
	//ONE_MCP 变量表中虚拟 MCP 输入地址指针
	//ONE_MCP 变量表中虚拟 MCP 输出地址指针

5.6.11 FB856: OP_MCP_Clone（MCP 克隆）

程序功能

同一 NCU 配置两个 MCP，并且两个 MCP 工作在相同的方式组下，当激活其中一个 MCP 时，另一 MCP 失效，但失效的 MCP 的按键指示状态与激活的 MCP 同步（支持 JOG、MDA、AUTO 方式下切换），并且在切换 MCP 时主轴倍率与进给倍率在通道中保持，只有当激活 MCP 倍率开关位置与转换前激活的 MCP 相同时，当前激活的 MCP 倍率开关才能生效。如果每个 MCP 同时还配置一个 HT2 手持单元，并且能够实现在激活 MCP 时，同时激活与之对应的手持 HT2 单元。

前提条件

- 两个 MCP 工作在同一方式组下。
- 在程序块中需要有标准块 FC19。
- 需要在 OB100 中设置两个 MCP 的 IO 地址、通讯地址、通讯类型。
- 需要在 OB100 中设置首次启动时激活的 MCP 配置的 HT2 手持单元的地址。
- FB856 功能块仅适用于铣床版 MCP，内部程序调用了 FC19。

注意事项

- 首次下载程序后，需要重启 PLC，否则可能出现 MCP 闪烁的问题。
- FB856 调用位置与 FC19 遵循相同的原则，即：FC2 后，FC10 前。
- 如果用户需要使用自定义键，地址与在 OB100 中定义的 MCP1 地址相同，即与当前激活 MCP 一致。
- MCP 切换后仅当倍率开关与切换前激活的 MCP 相同时，激活的 MCP 倍率开关才生效。
- HT2 手持单元连接模块的硬件地址必须不同，不能重复。
- MCP 的 IO 设置地址必须不同，否则 MCP 的操作状态无法同步，且会导致与之对应的 HT2 无法正常工作。

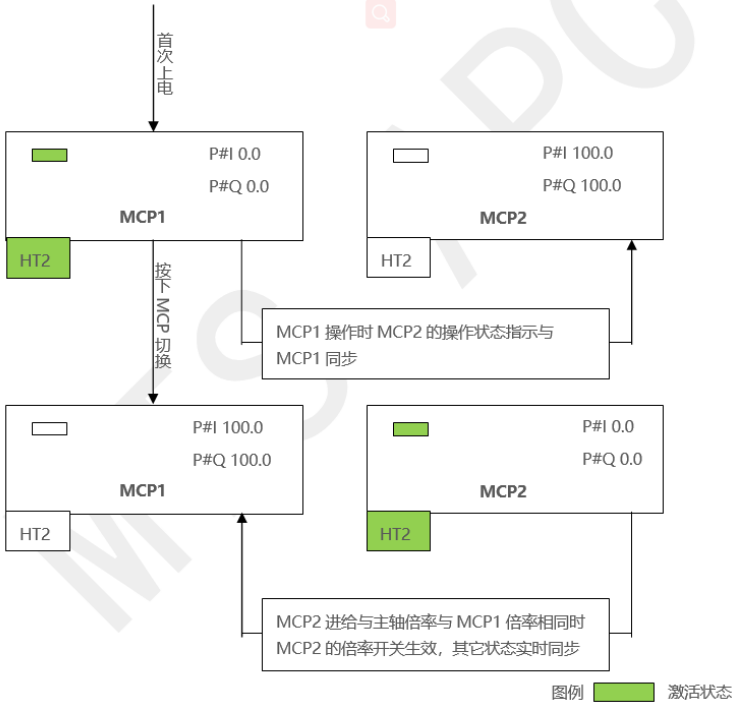
实验条件

硬　件	
MCP1	MCP 483C PN：6FC5303-0AF22-0AA1
MCP2	MCP 483C PN：6FC5303-0AF22-0AA1
HT1	6FC5303-0AA00-2AA0
HT2	6FC5303-0AA00-2AA0
连接盒 PN Basic 1	6AV6671-5AE01-0AX0
连接盒 PN Basic 2	6AV6671-5AE01-0AX0
工业以太网交换机	Scalance X005：6GK5005-0BA00-1AA3
MCP 配置方式	
IE 面板	

形式参数说明

信号	I/O	类型	取值范围	含义
BAGNo	I	BYTE	B#16#00 - B#16#0A	开机首次激活的 MCP，运行方式信号需要传输至的 BAG 的编号。
ChanNo	I	BYTE	B#16#00 - B#16#0A	通道信号需要传输至的通道的编号
SpindleIFNo	I	BYTE	0 - 31 (B#16#1F)	主轴数据需要传输至的进给轴/主轴的编号（对应机床轴的编号）
ChangeOver	I	BOOL		MCP 切换键，用户自定义
sTimer	I	TIMER		MCP 切换定时器
HT2_1	I	INT		HT2 硬件开关编码，需要转换为十进制，绑定 MCP1，同时需要填写在 OB100 中
HT2_2	I	INT		HT2 硬件开关编码，需要转换为十进制，绑定 MCP2
FeedHold	O	BOOL	0 (FALSE), 1 (TRUE)	MCP 进给停止，模态生效
SpindleHold	O	BOOL	0 (FALSE), 1 (TRUE)	MCP 主轴停止，模态生效
Msg	O	BYTE	Bit: 0 (FALSE), 1 (TRUE)	Bit0：第一面板激活 Bit1：第二面板激活 Bit2：进给倍率未同步 Bit3：主轴倍率未同步 Bit4-bit7：预留

面板切换功能示意图



图例

应用实例

程序	注释
CALL “OP_MCP_Clone”, “DB856”	//调用 FB856，背景数据块 DB856
BAGNo :=B#16#1	//BAG 编号 1
ChanNo :=B#16#1	//通道编号 1
SpindleIFNo :=B#16#4	//主轴接口编号:4，用户根据需要输入主轴号
ChangeOver :=I107.7	//面板切换按键
sTimer :=T21	//面板切换延时
HT2_1 :=14	//HT2：1#地址
HT2_2 :=15	//HT2：2#地址
FeedHold :=M100.0	//进给轴进给保持
SpindleHold :=M100.1	//主轴禁止保持
Msg :=DB2.DBB185	//指定报警 DB 块，bit0:第一面板激活；bit1:第二面板激活
SET	
S Q 5.7	//激活面板状态灯，用户自定义
R Q 105.7	//克隆面板状态灯，用户自定义

在 OB100 为形式参数赋值

程序	注释
CALL "LBP_ConfigBP"	//调用 FC1，840DSL 为 Call FB1, DB7
MCPNum :=2	//MCP 数量
MCP1In :=P#I 0.0	//MCP1 输入起始地址
MCP1Out :=P#Q 0.0	//MCP1 输出起始地址
MCP1BusAdr :=192	//MCP1 地址

… …
MCP2In :=P#I 100.0
MCP2Out :=P#Q 100.0
MCP2BusAdr :=193
… …
MCPBusType :=B#16#55
HTIf :=5
HTIn :=P#M 300.0
HTOut :=P#M 320.0
… …
HTAdr :=14
… …
	
	//MCP2 输入起始地址
	//MCP2 输出起始地址
	//MCP2 地址
	
	//MCP1&MCP2 的通讯形式
	//HT2 接口
	// HT2 的发送数据
	// HT2 的接收数据
	
	//HT2 地址

5.6.12 FC168: OP_HT2（HT2 手轮控制）

程序功能

HT2 手持单元的控制，支持匹配 MCP483 和 MCP310。按 X/Y/Z 按键直接激活对应轴的工件坐标系，此时按右上角空白功能键可以切换到该轴的机床坐标系，松开按键返回到工件坐标系；按 4thAxis 按键，屏幕上显示最多 12 个轴（MCP310 最多 9 个轴），选择后激活对应轴的机床坐标系。

使用 FC168 时，需要在 DB168 做如下修改：

- MachineAxisName: 按照实际的机床轴顺序设定轴名称
- xConf_ch1..10：为每个通道设定通道轴的序列，与 MD20070 相同，旋转轴输入负值即可
- ModeGroup：个位和十位将方式组分配给通道，百位设定 MCP 的类型
  0xx: FC19/MCP1
  1xx: FC19/MCP2
  2xx: FC24/MCP1
  4xx: FC24/MCP2

调用时需要自行生成 FB2 的背景数据块 DB169。

详细说明请参考《HT2 调试 20120214.pdf》。

形式参数说明

信号	I/O	类型	含义
BHG_on_condition	I	Bool	手轮激活条件
BHG_stop	I	Bool	手轮停止
inch	I	Bool	轴单位：英制
BHG_activ	IO	Bool	手轮激活
BAGNo	IO	Byte	当前方式组号
Menu	IO	Byte	当前 HT2 显示菜单
ChanNo	IO	Byte	当前通道号

应用实例

必须在在 MCP 面板控制程序之前调用

程序	注释
CALL " OP_HT2"	//调用 FC168，HT2 手轮控制
BHG_on_condition :=True	
BHG_stop :=False	
inch :=False	
BHG_activ :="GlobalDB".HandWhell.HT2.Active	
BAGNo :="GlobalDB".HandWhell.HT2.BAGNo	
Menu :="GlobalDB".HandWhell.HT2.Menu	
ChanNo :="GlobalDB".HandWhell.HT2.ChanNo	
CALL "MCP_IFM"	//调用 FC19，MCP483 控制
BAGNo :=1	
ChanNo :="GlobalDB".HandWhell.HT2.ChanNo	
SpindleIFNo :=4	
FeedHold :="GlobalDB".MCP.out.FeedStop	
SpindleHold :="GlobalDB".MCP.out.SpindleStop	

5.7 辅助功能（编码：860~899）

5.7.1 FC860: Aux_Chipconveyor（排屑器控制）

程序功能

用于控制排屑器的运行，提供手动操作及程序自动控制两种接口，并可以进行交叉控制。排屑器在自动运行程序结束后，可根据设定时间（分钟）延时自动停止。

手动控制：正转通过按键进行单键启停控制，反转通过另一按键点动，主要用于铁屑缠绕时的故障排除。

自动控制：可以设定启动和停止的 M 代码，支持带扩展地址的 M 代码，自动控制仅支持正转。

前提条件

排屑器电机正常信号为 1

形式参数说明

信号	I/O	类型	含义
iKeyFWD	I	BOOL	排屑器正转按键
iKeyBWD	I	BOOL	排屑器反转按键
iConveyorMotorOK	I	BOOL	排屑器电机正常信号
iLamptest	I	BOOL	灯测试按键
sChanNr	I	INT	M 代码所在通道号，99=所有通道有效
sExtMAddr	I	INT	扩展 M 地址，0=无扩展地址
sMFunctON	I	INT	排屑器启动 M 代码
sMFunctOFF	I	INT	排屑器停止 M 代码
sDelayTimeOFF	I	INT	程序结束后延迟关断时间（分钟）
oMsgMotorProt	IO	BOOL	排屑器电机保护消息
oLedFWD	IO	BOOL	正转按键指示灯
oLedBWD	IO	BOOL	反转按键指示灯
oFWD	IO	BOOL	正转信号输出
oBWD	IO	BOOL	反转信号输出
RefDW	IO	Array[0..1] of DWord	中间变量寄存双字数组

应用实例

手动方式通过 MCP 的 T5(正转)和 T4(反转)，自动方式在加工程序中通过 M56(正转)或 M57(反转)对排屑器进行控制。RefDW 中间变量寄存去需要在 DB 块中建立 Dword 数组变量：Array[0..1] of DWord

程序	注释
CALL " Aux_Chipconveyor"	//调用 FC860
iKeyFWD :="McpIn".customerKey5	//MCP 按键 T5
iKeyBWD :="McpIn".customerKey4	//MCP 按键 T4
iConveyorMotorOK := I103.0	//排屑器电机保护开关未动作

iLamptest	:="McpIn".customerKey15	//测试灯按键 MCP T15 按键
sChanNr	:=1	//M 功能所在通道 1
sExtMAddr	:=0	//无 M 扩展地址
sMFunctON	:=56	//启动排屑器 M 代码 M56
sMFunctOFF	:=57	//停止排屑器 M 代码 M57
sDelayTimeOFF	:=1	//程序结束后 1 分钟排屑器停止运行
oMsgMotorProt	:=M1200.0	//排屑器电机保护开关动作
oLedFWD	:="McpOut".customerKey5	//MCP 指示灯 T5
oLedBWD	:="McpOut".customerKey4	//MCP 指示灯 T4
oFWD	:=Q50.0	//排屑器正转输出
oBWD	:=Q50.1	//排屑器反转输出
RefDW	:="GlobalDB".ChipConveyor.RefDW	//中间变量寄存

5.7.2 FC861: Aux_Coolant（冷却控制）

程序功能

此功能块用于控制设备的冷却单元，它包括 1 个冷却泵和 2 个冷却阀，通过选项位可以激活需要的冷却阀，当冷却阀开始工作时冷却泵会自动工作。

冷却泵可以通过按键或 M 功能启动，通过输入参数可以定义需要的 M 功能，并且可以根据需要激活相应通道的 M 功能，如果所用通道 M 功能同时激活，通道号输入 99 即可。在冷却过程中，如果打开安全门，冷却阀立即关闭；当安全门关闭时，冷却阀再次打开，注意，如果没有安全门，请把安全门关闭信号置为 1，否则冷却阀不能正常工作。冷却阀的工作状态通过两个指示灯来体现(可以通过指示灯测试键直接测试冷却阀指示灯工作是否正常)，

当系统系统出现故障时，下面的报警信号将被输出。

- 冷却泵泵保护开关掉闸
- 冷却系统液位过低

前提条件

- 未触发急停信号
- 冷却单元控制电压正常
- 冷却泵保护开关处于工作状态

形式参数说明

信号	I/O	类型	含义
iKeyCool1	I	BOOL	冷却阀 1 控制开关
iKeyCool2	I	BOOL	冷却阀 2 控制开关
iValve1Enable	I	BOOL	冷却阀 1 使能
iValve2Enable	I	BOOL	冷却阀 2 使能
iCooMotorOK	I	BOOL	冷却泵保护开关，常闭
iContrVoltON	I	BOOL	控制电源激活，常闭
iCool_Level	I	BOOL	冷却单元液位

iEMG	I	BOOL	急停信号，常闭
iReset	I	BOOL	复位信号
iPowerOnReset	I	BOOL	重新上电
iDoorClosed	I	BOOL	安全门关闭
iLEDTest	I	BOOL	状态等测试，常开
sChanNo	I	INT	通道号:1...10;99(所有通道)
sExtMAddr	I	INT	M 扩展功能地址:0...99
sMCool1ON	I	INT	冷却阀 1 开 M 指令：0...32767
sMCool2ON	I	INT	冷却阀 2 开 M 指令：0...32767
sMCoolOFF	I	INT	冷却阀 1&2 关闭 M 指令：0...32767
oValveCool1	IO	BOOL	冷却阀 1
oValveCool2	IO	BOOL	冷却阀 2
oCool_pump	IO	BOOL	冷却泵
oCool1LED	IO	BOOL	冷却阀 1 状态指示灯
oCool2LED	IO	BOOL	冷却阀 2 状态指示灯
oMsgMotorProt	IO	BOOL	电机保护开关触发，自定义报警
oMsgCoolLevel	IO	BOOL	冷却液位报警，自定义报警
oDeviceActivated	IO	BOOL	程序功能激活
RefDW	IO	DWORD	内部状态字(双字)

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序		注释
CALL " Aux_Coolant"		//调用 FC861
iKeyCool1	:="GlobalDB".AUX_COOLANT.iKeyCool1	//冷却阀 1 启动按键
iKeyCool2	:="GlobalDB".AUX_COOLANT.iKeyCool2	//冷却阀 2 启动按键
iValve1Enable	:="GlobalDB".AUX_COOLANT.iValve1Enable	//冷却阀 1 使能
iValve2Enable	:="GlobalDB".AUX_COOLANT.iValve2Enable	//冷却阀 2 使能
iCooMotorOK	:="GlobalDB".AUX_COOLANT.iCooMotorOK	//冷却泵保护开关信号
iContrVoltON	:="GlobalDB".AUX_COOLANT.iContrVoltON	//冷却单元电压接通信号
iCool_Level	:="GlobalDB".AUX_COOLANT.iCool_Level	//冷却单元液位信号
iEMG	:="GlobalDB".AUX_COOLANT.iEMG	//急停信号
iReset	:="GlobalDB".AUX_COOLANT.iReset	//复位信号
iPowerOnReset	:="GlobalDB".AUX_COOLANT.iPowerOnReset	//系统重新上电
iDoorClosed	:="GlobalDB".AUX_COOLANT.iDoorClosed	//安全门关闭检测信号
iLEDTest	:="GlobalDB".AUX_COOLANT.iLEDTest	//LED 测试按键
sChanNo	:="GlobalDB".AUX_COOLANT.sChanNo	//通道号
sExtMAddr	:="GlobalDB".AUX_COOLANT.sExtMAddr	//M 扩展地址
sMCool1ON	:="GlobalDB".AUX_COOLANT.sMCool1ON	//冷却阀 1 打开 M 功能

sMCool2ON	:="GlobalDB".AUX_COOLANT.sMCool2ON	//冷却阀 2 打开 M 功能
sMCoolOFF	:="GlobalDB".AUX_COOLANT.sMCoolOFF	//冷却阀 1&2 关闭 M 功能
oValveCool1	:="GlobalDB".AUX_COOLANT.oValveCool1	//冷却阀 1
oValveCool2	:="GlobalDB".AUX_COOLANT.oValveCool2	//冷却阀 2
oCool_pump	:="GlobalDB".AUX_COOLANT.oCool_pump	//冷却泵
oCool1LED	:="GlobalDB".AUX_COOLANT.oCool1LED	//冷却阀 1 状态指示
oCool2LED	:="GlobalDB".AUX_COOLANT.oCool2LED	//冷却阀 2 状态指示
oMsgMotorProt	:="GlobalDB".AUX_COOLANT.oMsgMotorProt	//冷却泵保护开关断开
oMsgCoolLevel	:="GlobalDB".AUX_COOLANT.oMsgCoolLevel	//冷却单元液位报警输出
oDeviceActivated	:="GlobalDB".AUX_COOLANT.oDeviceActivated	//冷却单元激活状态
oFlagword	:="GlobalDB".AUX_COOLANT.oFlagword	//内部状态字

5.7.3 FC862: Aux_DeviceControl_1x1（单键控制系统）

程序功能

此程序块主要功能为通过单独按键控制系统启停，并通过返回状态形成闭环控制。

形式参数说明

信号	I/O	类型	含义
sMon(200ms)	I	INT	监控时间，当量：200Ms
sPules(5Hz)	I	BOOL	计时脉冲 5Hz
iOverLoad(c)	I	BOOL	系统保护开关
iReadyCondition	I	BOOL	系统就绪条件
iReset	I	BOOL	系统复位
iKey	I	BOOL	系统启停键
iTestKey	I	BOOL	系统状态灯测试
oDeviceReady	O	BOOL	系统就绪输出
oActuator	O	BOOL	控制系统
oStatusOnLed	O	BOOL	控制系统状态指示灯
oOverLoadMsg	O	BOOL	控制系统保护开关报警输出
oConditionMsg	O	BOOL	控制系统未就绪报警输出
RefDW	IO	Dword	内部状态双字

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Aux_DeviceControl_1x1"	//调用 FC862
sMon(200ms) :=100	//监控之间，时间当量：200ms
sPules(5Hz) :="GlobalDB".Clock._5Hz	//计时脉冲
iOverLoad(c) :="GlobalDB".AuxFunc.DeviceControl_1x1."iOverLoad(c)"//低电平有效	//系统保护信号
iReadyCondition :="GlobalDB".AuxFunc.DeviceControl_1x1.iReadyCondition	//系统就绪条件
iReset :="GlobalDB".AuxFunc.DeviceControl_1x1.iReset	//系统复位
iKey :="GlobalDB".AuxFunc.DeviceControl_1x1.iKey	//系统启停按键
iTestKey :="GlobalDB".AuxFunc.DeviceControl_1x1.iTestKey	//系统状态指示输出灯测试
oDeviceReady :="GlobalDB".AuxFunc.DeviceControl_1x1.oDeviceReady	//系统就绪返回信号
oActuator :="GlobalDB".AuxFunc.DeviceControl_1x1.oActuator	//系统输出
oStatusOnLed :="GlobalDB".AuxFunc.DeviceControl_1x1.oStatusOnLed	//系统状态指示
oOverLoadMsg :="GlobalDB".AuxFunc.DeviceControl_1x1.oOverLoadMsg	//系统保护报警输出
oConditionMsg :="GlobalDB".AuxFunc.DeviceControl_1x1.oConditionMsg	//系统未就绪报警输出
RefDW :="GlobalDB".AuxFunc.DeviceControl_1x1.RefDW	//内部状态字

5.7.4 FC863: Aux_DeviceControl_2x1（双键控制系统）

程序功能

此程序块主要功能为通过两个按键控制系统启停，并通过返回状态形成闭环控制。

形式参数说明

信号	I/O	类型	含义
sMon(200ms)	I	INT	监控时间，当量：200Ms
sPules(5Hz)	I	BOOL	即使脉冲 5Hz
iOverLoad(c)	I	BOOL	系统保护开关
iReadyCondition	I	BOOL	系统就绪条件
iReset	I	BOOL	系统复位
iKeyOn	I	BOOL	系统启动键
iKeyOff	I	BOOL	系统停止键
iTestKey	I	BOOL	系统状态灯测试
oDeviceReady	O	BOOL	系统就绪输出
oActuator	O	BOOL	控制系统
oStatusOnLed	O	BOOL	控制系统状态指示灯
oOverLoadMsg	O	BOOL	控制系统保护开关报警输出
oConditionMsg	O	BOOL	控制系统未就绪报警输出
RefDW	IO	Dword	内部状态双字

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序	注释
CALL " Aux_DeviceControl_2x1"	//调用 FC863
sMon(200ms) :=100	//监控之间，时间当量：200ms
sPules(5Hz) :="GlobalDB".Clock._5Hz	//计时脉冲
iOverLoad(c) :="GlobalDB".AuxFunc.DeviceControl_2x1."iOverLoad(c)"	//系统保护信号
iReadyCondition :="GlobalDB".AuxFunc.DeviceControl_2x1.iReadyCondition	//系统就绪条件
iReset :="GlobalDB".AuxFunc.DeviceControl_2x1.iReset	//系统复位
iKeyOn :="T1" //"GlobalDB".AuxFunc.DeviceControl_2x1.iKeyOn	//系统启动按键
iKeyOff :="T2_Key"//"GlobalDB".AuxFunc.DeviceControl_2x1.iKeyOff	//系统关闭按键
iTestKey :="GlobalDB".AuxFunc.DeviceControl_2x1.iTestKey	//系统状态指示输出灯测试
oDeviceReady :="GlobalDB".AuxFunc.DeviceControl_2x1.oDeviceReady	//系统就绪返回信号
oActuator :="GlobalDB".AuxFunc.DeviceControl_2x1.oActuator	//系统输出
oStatusOnLed :="T2_MCP"//"GlobalDB".AuxFunc.DeviceControl_2x1.oStatusOnLed	//系统状态指示 //系统保护报警输出
oOverloadMsg :="GlobalDB".AuxFunc.DeviceControl_2x1.oOverloadMsg	//系统未就绪报警输出
oConditionMsg :="GlobalDB".AuxFunc.DeviceControl_2x1.oConditionMsg	//内部状态字
RefDW :="GlobalDB".AuxFunc.DeviceControl_2x1.RefDW	

5.7.5 FC864: Aux_FlipFlop_v1（单键启停）

程序功能

实现单一按键控制启停，即按一次启动，再按一次停止

形式参数说明

信号	I/O	类型	含义
iKey	I	BOOL	输入按键
iReset	I	BOOL	复位或重置信号输入
oLamp	O	BOOL	指示灯输出
Funct	IO	BOOL	功能状态输出
RefBit	IO	BOOL	中间变量存储位

应用实例

单键控制输出

程序	注释
CALL“Aux_FlipFlop.v1”	//调用 FC864
iKey :=I7.6	//MCP 按键 T2
iReset :=I3.7	//MCP 按键 Reset
oLamp :=Q5.6	//MCP 按键 T2 指示灯
Funct :=Q300.0	//输出信号
RefBit :=M500.0	//中间变量块外存储（M 位或 DB 块的 Bool 类型变量）

5.7.6 FC865: Aux_Hydraulic（液压控制）

程序功能

此功能块用于控制液压单元，通过两个按键控制它的启停，停止按键要求接入常闭点。

液压单元启动基本前提：

- 未触发急停信号
- 液压单元控制电压正常
- 液压泵保护开关处于工作状态

设置监控时间后，按下液压启动键，液压单元工作且信号灯亮起，如有必要也可以通过信号灯测试按键测试信号灯是否能够正常工作。液压单元在建立压力时(监控时间内)，压力报警输出无效。

当系统系统出现故障时，下面的报警信号将被输出

- 液压泵保护开关掉闸
- 液压油过滤器堵塞
- 液压系统压力不足

形式参数说明

信号	I/O	类型	含义
iHydON	I	BOOL	液压启动信号
iHydOFF	I	BOOL	液压关闭信号
iHydOK	I	BOOL	液压泵保护 OK，常闭
iFilterOK	I	BOOL	液压油过滤器 OK，常闭
iPressureOK	I	BOOL	液压压力 OK，常闭
iContrVoltON	I	BOOL	控制电源激活，常闭
iEMG	I	BOOL	急停信号，常闭
iReset	I	BOOL	复位信号
iFlag_5Hz	I	BOOL	5HZ 信号
iLEDTest	I	BOOL	状态指示测试信号
iPowerOnReset	I	BOOL	重新上电信号
sMonitoringTime	I	INT	监控时间，单位：sec；范围：0...32767
oHydraulics	IO	BOOL	液压泵
oHydLED	IO	BOOL	液压工作指示灯
oMsgMotorProt	IO	BOOL	液压泵保护开关报警
oMsgFilter	IO	BOOL	液压系统过滤器报警
oMsgpressure	IO	BOOL	液压系统压力报警
oFlagword	IO	DWORD	内部状态字
oFlagword_Timer	IO	INT	计数状态字

应用实例

接口信号通过 GlobalDB（DB800）转换，用户可自行修改，详情请参考例子程序。

程序		注释
CALL " Aux_Hydraulic"		//调用 FC865
iHydON	:="GlobalDB"."AUX__Hydraulic".iHydON	//液压启动按键
iHydOFF	:="GlobalDB"."AUX__Hydraulic".iHydOFF	//液压停止按键(常闭)
iHydOK	:="GlobalDB"."AUX__Hydraulic".iHydOK	//液压保护开关信号
iFilterOK	:="GlobalDB"."AUX__Hydraulic".iFilterOK	//液压油过滤器信号
iPressureOK	:="GlobalDB"."AUX__Hydraulic".iPressureOK	//液压系统压力信号
iContrVoltON	:="GlobalDB"."AUX__Hydraulic".iContrVoltON	//液压单元电压接通信号
iEMG	:="GlobalDB"."AUX__Hydraulic".iEMG	//急停信号
iReset	:="GlobalDB"."AUX__Hydraulic".iReset	//复位信号
iFlag_5Hz	:="GlobalDB"."AUX__Hydraulic".iFlag_5Hz	//5Hz 时钟信号
iLEDTest	:="GlobalDB"."AUX__Hydraulic".iLEDTest	//液压指示灯测试键
iPowerOnReset	:="GlobalDB"."AUX__Hydraulic".iPowerOnReset	//系统重新上电
sMonitoringTime	:="GlobalDB"."AUX__Hydraulic".sMonitoringTime	//监控时间
oHydraulics	:="GlobalDB"."AUX__Hydraulic".oHydraulics	//液压泵
oHydLED	:="GlobalDB"."AUX__Hydraulic".oHydLED	//液压系统指示灯
oMsgMotorProt	:="GlobalDB"."AUX__Hydraulic".oMsgMotorProt	//液压泵保护报警输出
oMsgFilter	:="GlobalDB"."AUX__Hydraulic".oMsgFilter	//过滤器报警输出
oMsgpressure	:="GlobalDB"."AUX__Hydraulic".oMsgpressure	//液压系统压力报警输出
oFlagword	:="GlobalDB"."AUX__Hydraulic".oFlagword	//内部状态字
oFlagword_Timer	:="GlobalDB"."AUX__Hydraulic".oFlagword_Timer	//液压压力建立计时

5.7.7 FC866: Aux_Lubrication_v1（润滑控制 v1）

程序功能

此功能块用于控制设备中心润滑单元，中心润滑单元可以通过按键、系统重新启动以及自定义 M 功能启动。

中心润滑单元启动基本前提：

- 未触发急停信号
- 润滑单元控制电压正常
- 润滑单元保护开关处于工作状态

中心润滑单元工作的三个阶段：

阶段 1：中心润滑单元启动，润滑压力建立，在设定时间内，系统压力未达到要求压力，压力报警输出。

阶段 2：润滑系统建立压力正常，润滑单元工作至设置的润滑时间结束。

阶段 3：在润滑结束后，润滑间隔时间开始计时，到达计时时间，润滑单元再次启动。

除了设置启动润滑间隔时间启动润滑单元外，还可以通过外部按键或 M 指令启动它。可以根据润滑单元所在通道激活响应通道的 M 功能，如果需要激活所有通道的 M 功能，在输入参数" sChanNo"输入 99 即可。

当系统系统出现故障时，下面的报警信号将被输出

- 润滑保护开关掉闸
- 润滑系统液位过低
- 润滑系统压力不足

形式参数说明

信号	I/O	类型	含义
iLubKey	I	BOOL	润滑启动按钮
iLubLevel	I	BOOL	液位开关信号，1:液位 OK
iLub_P	I	BOOL	润滑压力信号，1:压力 OK
iLubOVRLD	I	BOOL	润滑单元保护触点，1:OK
iLampTest	I	BOOL	状态测试按钮，
iOptBit	I	BOOL	选项功能
iPowerOnReset	I	BOOL	设备上电复位信号
iReset	I	BOOL	复位信号
iEMG	I	BOOL	急停信号
iContrVoltON	I	BOOL	润滑单元控制电源 OK
iFlag_5Hz	I	BOOL	5Hz 信号
sChanNo	I	INT	通道号
sExtMAddr	I	INT	M 功能号:0...99
sMLubON	I	INT	润滑启动 M 指令：0...32767
sLubInterval	I	INT	润滑间隔时间，单位：min；范围：0...32767
sDelayOffTime	I	INT	润滑单元压力建立时间，单位：sec；范围：0...32767
sMonitoringTime	I	INT	监控时间，单位：sec；范围：0...32767
oLubrication	IO	BOOL	中心润滑单元
oLubricationLED	IO	BOOL	中心润滑单元工作指示
oOillevelLED	IO	BOOL	中心润滑单元液位指示
oErrorLED	IO	BOOL	中心润滑单元错误指示
oMsgOillevel	IO	BOOL	中心润滑单元液位报警
oMsgMotorProt	IO	BOOL	中心润滑单元保护报警
oMsgPressure	IO	BOOL	中心润滑单元压力报警
oDeviceActivated	IO	BOOL	程序激活
oFlagword	IO	DWORD	内部状态字
oFlagwordTimer	IO	INT	润滑压力建立计时器
oFlagwordInterval	IO	INT	润滑间隔计时器
oFlagwordDelayTime	IO	INT	润滑延时计时器

应用实例

程序		注释
CALL "Aux_Lubrication"		//调用 FC866
iLubKey	:="GlobalDB".AUX_LUBRICATION.iLubKey	//润滑启动按键
iLubLevel	:="GlobalDB".AUX_LUBRICATION.iLubLevel	//润滑单元液位信号
iLub_P	:="GlobalDB".AUX_LUBRICATION.iLub_P	//润滑单元压力信号

iLubOVRLD	:="GlobalDB".AUX_LUBRICATION.iLubOVRLD	//润滑单元保护开关信号
iLampTest	:="GlobalDB".AUX_LUBRICATION.iLampTest	//润滑单元指示灯测试按键
iOptBit	:="GlobalDB".AUX_LUBRICATION.iOptBit	//润滑单元选项位
iPowerOnReset	:="GlobalDB".AUX_LUBRICATION.iPowerOnReset	//系统重新上电
iReset	:="GlobalDB".AUX_LUBRICATION.iReset	//复位信号
iEMG	:="GlobalDB".AUX_LUBRICATION.iEMG	//急停信号
iContrVoltON	:="GlobalDB".AUX_LUBRICATION.iContrVoltON	//润滑单元电压正常
iFlag_5Hz	:="GlobalDB".AUX_LUBRICATION.iFlag_5Hz	//5Hz 时钟信号
sChanNo	:="GlobalDB".AUX_LUBRICATION.sChanNo	//通道号
sExtMAddr	:="GlobalDB".AUX_LUBRICATION.sExtMAddr	// M 扩展地址
sMLubON	:="GlobalDB".AUX_LUBRICATION.sMLubON	//润滑打开 M 功能
sLubInterval	:="GlobalDB".AUX_LUBRICATION.sLubInterval	//润滑间隔时间
sDelayOffTime	:="GlobalDB".AUX_LUBRICATION.sDelayOffTime	//润滑压力创建时间
sMonitoringTime	:="GlobalDB".AUX_LUBRICATION.sMonitoringTime	//润滑监控时间
oLubrication	:="GlobalDB".AUX_LUBRICATION.oLubrication	//润滑单元
oLubricationLED	:="GlobalDB".AUX_LUBRICATION.oLubricationLED	//润滑单元指示灯
oOillevelLED	:="GlobalDB".AUX_LUBRICATION.oOillevelLED	//润滑单元液位指示灯
oErrorLED	:="GlobalDB".AUX_LUBRICATION.oErrorLED	//润滑单元错误指示灯
oMsgOillevel	:="GlobalDB".AUX_LUBRICATION.oMsgOillevel	//润滑单元液位报警输出
oMsgMotorProt	:="GlobalDB".AUX_LUBRICATION.oMsgMotorProt	//润滑单元保护开关断开
oMsgPressure	:="GlobalDB".AUX_LUBRICATION.oMsgPressure	//润滑单元压力报警输出
oDeviceActivated	:="GlobalDB".AUX_LUBRICATION.oDeviceActivated	//润滑单元激活状态
oFlagword	:="GlobalDB".AUX_LUBRICATION.oFlagword	//内部状态字
oFlagwordTimer	:="GlobalDB".AUX_LUBRICATION.oFlagwordTimer	//润滑计时
oFlagwordInterval	:="GlobalDB".AUX_LUBRICATION.oFlagwordInterval	//润滑间隔计时
oFlagwordDelayTime	:="GlobalDB".AUX_LUBRICATION.oFlagwordDelayTime	//润滑延迟计时

5.7.8 FC867: Aux_Lubrication_v2（润滑控制 v2）

程序功能

M 代码和手动控制润滑，FC866 简化版。

形式参数说明

信号	I/O	类型	含义
sChannelNo	I	INT	通道号
sM_Lubrication_Start	I	INT	M 代码启动润滑
sMonitoring_Time	I	INT	润滑压力到达监控时间
sLubrication_Interval_Time	I	INT	润滑启动时间
sDelay_Off_Time	I	INT	润滑暂停时间
s5Hz	I	BOOL	计数脉冲
iByPass	I	BOOL	设备旁路

iOverLoad	I	BOOL	润滑泵过载
iOilLevel	I	BOOL	润滑单元液位信号
iEMG	I	BOOL	急停
iMsgQuit	I	BOOL	清除报警
iKeyStart	I	BOOL	手动润滑
iStatusTest	I	BOOL	润滑单元指示灯测试按键
oRelayLubrication	O	BOOL	润滑泵输出
oRunningLed	O	BOOL	润滑运行中
oOverloadMsg	IO	BOOL	润滑电机过载
oOilLevelMsg	IO	BOOL	润滑油液位低报警
RefDW	IO	Array[0..1] of DWord	内部使用

应用实例

程序	注释
CALL "Aux_LubricationControl" sChannelNo :=1 //"GlobalDB".AuxFunc.LubricationControl.sChannelNo sM_Lubrication_Start :="GlobalDB".AuxFunc.LubricationControl.sM_Lubrication_Start sMonitoring_Time、 :=10 //"GlobalDB".AuxFunc.LubricationControl.sMonitoring_Time sLubrication_Interval_Time :=100 // sDelay_Off_Time :=20 //"GlobalDB".AuxFunc.LubricationControl.sDelay_Off_Time s5Hz :="GlobalDB".Clock._5Hz//"GlobalDB".LubricationControl.s5Hz iByPass :="GlobalDB".AuxFunc.LubricationControl.iByPass iOverLoad :="GlobalDB".AuxFunc.LubricationControl.iOverLoad iOilLevel :=TRUE //"GlobalDB".AuxFunc.LubricationControl.iOilLevel iEMG :=FALSE //"GlobalDB".AuxFunc.LubricationControl.iEMG iMsgQuit :="GlobalDB".AuxFunc.LubricationControl.iMsgQuit iKeyStart :="GlobalDB".AuxFunc.LubricationControl.iKeyStart iStatusTest :="GlobalDB".AuxFunc.LubricationControl.iStatusTest oRelayLubrication :="GlobalDB".AuxFunc.LubricationControl.oRelayLubrication oRunningLed :="GlobalDB".AuxFunc.LubricationControl.oRunningLed oOverloadMsg :="GlobalDB".AuxFunc.LubricationControl.oOverloadMsg oOilLevelMsg :="GlobalDB".AuxFunc.LubricationControl.oOilLevelMsg RefDW :="GlobalDB".AuxFunc.LubricationControl.RefDW	

5.7.9 FC868: Aux_SpLub（主轴润滑）

程序功能

输出主轴润滑信号（该信号为持续信号）并检测主轴运行时润滑压力。

参数说明

信号	I/O	类型	含义
sSPAxNr	I	INT	主轴的机床轴号
iSpLubPumpPowerOK	I	BOOL	润滑泵电源正常
iSpLubPressOK	I	BOOL	润滑压力正常
iSpLubOilLevelLow	I	BOOL	润滑油液位低
iSpLubOilFilterBlock	I	BOOL	润滑油过滤器堵塞
oSpLubON	IO	BOOL	润滑输出
oMsg_SpLubPressLow	IO	BOOL	润滑压力报警
oMsg_SpLubPumpNoPower	IO	BOOL	润滑泵未启动报警
oMsg_SpLubOilFilterBlock	IO	BOOL	润滑油过滤器堵塞报警
oMsg_SpLubOilLevelLow	IO	BOOL	润滑油液位低报警

应用实例

程序	注释
//全局变量初始化	
L "gp_par".MaxAxis	
T "GlobalDB".DB7.MaxAxis	
CALL "Aux_SpLub"	//调用 FC868
sSPAxNr :=4	//主轴的机床轴号 6
iSpLubPumpPowerOK :=TRUE	//润滑泵电机供电正常
iSpLubPressOK :=TRUE	//润滑压力正常
iSpLubOilLevelLow :="GlobalDB".AuxFunc.SpLub.iSpLubOilLevelLow	//润滑油液位低
iSpLubOilFilterBlock :="GlobalDB".AuxFunc.SpLub.iSpLubOilFilterBlock	//润滑油过滤器堵塞
oSpLubON :="GlobalDB".AuxFunc.SpLub.oSpLubON	//主轴润滑启动
oMsg_SpLubPressLow :="GlobalDB".AuxFunc.SpLub.oMsg_SpLubPressLow	//润滑油压力报警
oMsg_SpLubPumpNoPower :="GlobalDB".AuxFunc.SpLub.oMsg_SpLubPumpNoPower	//润滑泵电机未启动报警
oMsg_SpLubOilFilterBlock :="GlobalDB".AuxFunc.SpLub.oMsg_SpLubOilFilterBlock	//润滑油过滤器堵塞报警
oMsg_SpLubOilLevelLow :="GlobalDB".AuxFunc.SpLub.oMsg_SpLubOilLevelLow	//润滑油液位低报警

5.7.10 FC869: Aux_ThreeColorLED_v1（三色灯控制 v1）

程序功能

控制三色灯，支持蜂鸣器输出

红色指示灯：当有报警时指示灯亮，报警条件需要用户自行在 Network 3 中添加

黄色指示灯：程序执行结束（M02/M30）时，指示灯亮

绿色指示灯：AUTO 或 MDA 方式程序运行时指示灯亮

蜂鸣器：红色报警灯或外部触发条件满足

形式参数说明

信号	I/O	类型	含义
sChanNr	I	INT	三色灯反映状态的通道
iLampTest_Key	I	BOOL	灯测试按键
iRedLedCondition	I	BOOL	红灯触发外部条件
LEDClock	I	BOOL	闪烁时钟信号
iBeepCondition	I	BOOL	蜂鸣器发声外部条件
oLED_Red	IO	BOOL	红色指示灯输出
oLED_Yellow	IO	BOOL	黄色指示灯输出
oLED_Green	IO	BOOL	绿色指示灯输出
oBeep	IO	BOOL	蜂鸣器输出
RefW	IO	Array[0..2] of Word	中间变量寄存字数组

应用实例

接口信号来自”GlobalDB” DB800 对应接口，RefWord 需要在 DB 块中建立 Word 类型数组 Array[0..2] of Word

程序	注释
CALL "Aux_ThreeColorLED"	//调用 FC869
sChanNr :="GlobalDB".AuxFunc.ThreeColorLED.sChanNr	//三色灯指示通道 1 的状态
iLampTest_Key :="GlobalDB".AuxFunc.ThreeColorLED.iLampTest_Key	//灯测试按键 MCP T15
iRedLedCondition :="GlobalDB".AuxFunc.ThreeColorLED.iRedLedCondition	//红灯外部条件
LEDClock :="GlobalDB".Clock._1Hz	//闪烁时钟
iBeepCondition :="GlobalDB".AuxFunc.ThreeColorLED.iBeepCondition	//蜂鸣器外部条件
oLED_Red :="GlobalDB".AuxFunc.ThreeColorLED.oLED_Red	//红色指示灯输出
oLED_Yellow :="GlobalDB".AuxFunc.ThreeColorLED.oLED_Yellow	//黄色指示灯输出
oLED_Green :="GlobalDB".AuxFunc.ThreeColorLED.oLED_Green	//绿色指示灯输出
oBeep :="GlobalDB".AuxFunc.ThreeColorLED.oBeep	//蜂鸣器输出
RefWord :="GlobalDB".AuxFunc.ThreeColorLED.RefWord	//中间寄存器存储字数组

5.7.11 FC870: Aux_ToggleKey_Ext（双键控制单元）

程序功能

此程序块主要功能为通过 2 个按键控制系统启停。一般至少需要调用两次此程序块。

形式参数说明

信号	I/O	类型	含义
iKey	I	BOOL	系统启停键
oLamp	IO	BOOL	系统状态输出指示
RefBit	IO	BOOL	内部状态位

应用实例

程序	注释
CALL " Aux_ToggleKey_Ext "	//第一次调用 FC870，启动设备
iKey :="GlobalDB".AuxFunc.ToggleKey_Ext.iKey	//系统启动
oLamp :="GlobalDB".AuxFunc.ToggleKey_Ext.oLamp	//系统状态输出
RefBit :="GlobalDB".AuxFunc.ToggleKey_Ext.RefBit	//内部位
//**********	
CALL " Aux_ToggleKey_Ext "	//第二次调用 FC870，停止设备
iKey :="GlobalDB".AuxFunc.ToggleKey_Ext.iKey	//系统关闭
oLamp :="GlobalDB".AuxFunc.ToggleKey_Ext.oLamp	//系统状态输出
RefBit :="GlobalDB".AuxFunc.ToggleKey_Ext.RefBit	//内部位

5.7.12 FC871: Aux_ToggleKey_INT（单键控制单元）

程序功能

此程序块主要功能为通过 1 个按键控制系统启停。

形式参数说明

信号	I/O	类型	含义
iReset	I	BOOL	系统复位
iKey	I	BOOL	系统启停键
oLamp	I	BOOL	系统状态输出指示
RefBits	IO	Array[0..1] of Bool	内部状态位

应用实例

程序	注释
CALL "ToggleKey_INT"	//调用 FC871
iReset :="GlobalDB".AuxFunc.ToggleKey_INT.iReset	//系统复位
iKey :="GlobalDB".AuxFunc.ToggleKey_INT.iKey	//系统启动
oLamp :="GlobalDB".AuxFunc.ToggleKey_INT.oLamp	//系统状态输出
RefBits :="GlobalDB".AuxFunc.ToggleKey_INT.RefBits	//内部位

5.7.13 FC872: Aux_ThreeColorLED_v2 （三色灯控制 v2）

程序功能

控制三色灯，支持蜂鸣器输出，可通过使能位选择是否激活蜂鸣器及分别设定红灯和黄灯亮时是否激活蜂鸣器。

红色指示灯：当有报警时指示灯亮，在 Network 3 中通过掩码设定哪些 PLC 用户报警需要亮红灯。

黄色指示灯：程序执行结束（M02/M30）时，指示灯亮

绿色指示灯：AUTO 或 MDA 方式程序运行时指示灯亮

蜂鸣器：红色或黄色灯亮时，根据各自有效条件设置响

形式参数说明

信号	I/O	类型	含义
sChanNr	I	INT	三色灯反映状态的通道
iLampTest_Key	I	BOOL	灯测试按键
iBeepEnable	I	BOOL	蜂鸣器使能
iBeepYellow	I	BOOL	黄灯亮时蜂鸣器有效
iBeepRed	I	BOOL	红灯亮时蜂鸣器有效
oLED_Red	IO	BOOL	红色指示灯输出
oLED_Yellow	IO	BOOL	黄色指示灯输出
oLED_Green	IO	BOOL	绿色指示灯输出
oBeep	IO	BOOL	蜂鸣器输出
oBeepActLed	IO	BOOL	蜂鸣器状态有效，可用于外部指示灯
RefDW	IO	Array[0..1] of DWord	中间变量寄存字数组

应用实例

接口信号来自”GlobalDB” DB800 对应接口，RefDW 需要在 DB 块中建立 Word 类型数组 Array[0..2] of Word

程序	注释
CALL "Aux_ThreeColorLED_v2"	
sChanNr :=1	//三色灯指示通道 1 的状态
iLampTest_Key :="GlobalDB".AuxFunc.ThreeColorLED_v2.iLampTest_Key	//灯测试按键
iBeepEnable :="GlobalDB".AuxFunc.ThreeColorLED_v2.iBeepEnable	//蜂鸣器使能
iBeepYellow :="GlobalDB".AuxFunc.ThreeColorLED_v2.iBeepYellow	
iBeepRed :="GlobalDB".AuxFunc.ThreeColorLED_v2.iBeepRed	
oLED_Red :="oLEDRed"	//红色指示灯输出
oLED_Yellow :="oLEDYellow"	//黄色指示灯输出
oLED_Green :="oLEDGreen"	//绿色指示灯输出
oBeep :="oBeep"	//蜂鸣器输出
oBeepActLed :="GlobalDB".AuxFunc.ThreeColorLED_v2.oBeepActLed	//蜂鸣器有效
RefDW :="GlobalDB".AuxFunc.ThreeColorLED_v2.RefDW	//中间寄存器存储字数组

5.7.14 FC873: Aux_WorkLight（工作灯控制）

程序功能

机床工作照明灯控制，单键开关控制，可设定是否开机点亮。

形式参数说明

信号	I/O	类型	含义
iLampOnAtStart	I	BOOL	照明灯上电时亮起
iWorkLight_Key	I	BOOL	工作灯按键
oWorkLight	O	BOOL	工作灯输出
oWorkLightLed	IO	BOOL	工作灯按键指示灯
RefB	IO	BYTE	中间寄存器字节

应用实例

程序	注释
CALL "Aux_WorkLight" iLampOnAtStart :="GlobalDB".AuxFunc.WorkLight.iLampOnAtStart iWorkLight_Key :="GlobalDB".MCP.In.customerKey9 oWorkLight :="oWorkLight" oWorkLightLed :="MCPOut".customerKey9 RefB :="GlobalDB".AuxFunc.WorkLight.RefB	

5.7.15 FC874: Aux_DeviceControl_v1（外设控制 v1）

程序功能

通用的对外设的单键和 M 代码启停控制，集成保护监控信号的输入和报警的输出。功能块具有 Enable 信号，如果不许要执行，将 iEnable 设为 False 即可。

前提条件

- 电机热保护信号 iMotorProt 为 1
- 监控信号（如：液位、压力）iMonitorSignal 为 1
- 门关闭信号 iDoorClosed 为 1

形式参数说明

信号	I/O	类型	含义
iEnable	I	BOOL	设备使能
iKey	I	BOOL	设备控制开关
iMotorProt(c)	I	BOOL	设备保护开关，常闭
iContrVoltON(c)	I	BOOL	控制电源激活，常闭
iMonitorSignal(c)	I	BOOL	其他监控信号（如：液位），常闭

iEMG(c)	I	BOOL	急停信号，常闭
iReset	I	BOOL	复位信号
iPowerOnReset	I	BOOL	重新上电
iDoorClosed	I	BOOL	安全门关闭
sChanNo	I	INT	通道号:1...10;99(所有通道)
sExtMAddr	I	INT	M 功能号:0...99
sMCoolON	I	INT	设备开 M 指令：0...32767
sMCoolOFF	I	INT	设备关 M 指令：0...32767
oDeviceOn	IO	BOOL	设备启动
oDeviceLED	IO	BOOL	设备状态指示灯
oMsgMotorProt	IO	BOOL	电机保护开关触发，自定义报警
oMsgMonitor	IO	BOOL	其他监控信号报警，自定义报警
RefDW	IO	DWORD	内部状态字(双字)

应用实例

程序	注释
CALL "Aux_DeviceControl_v1"	
iEnable :="GlobalDB".ONE	
iKey :="GlobalDB".MCP.In.customerKey15	
iMotorProt(c) :="GlobalDB".AuxFunc.Coolant_v2."iMotorProt(c)"	
iContrVoltON(c) :="GlobalDB".MachineStartOK	
iMonitorSignal(c) :="GlobalDB".AuxFunc.Coolant_v2."iMonitorSignal(c)"	
iEMG(c) :="GlobalDB".EMG.Button	
iReset :="GlobalDB".MCP.In.reset	
iPowerOnReset :="GlobalDB".FirstScan	
iDoorClosed :="GlobalDB".AuxFunc.Coolant_v2.iDoorClosed	
sChanNo :=1	//M 代码在通道 1 生效
sExtMAddr :=0	
sMCoolON :=8	//M8 控制开启
sMCoolOFF :=9	//M9 控制关闭
oDeviceOn :="oCoolant"	
oDeviceLED :="MCPOut".customerKey15	
oMsgMotorProt :="GlobalDB".AuxFunc.Coolant_v2.oMsgMotorProt	
oMsgMonitor :="GlobalDB".AuxFunc.Coolant_v2.oMsgMonitor	
RefDW :="GlobalDB".AuxFunc.Coolant_v2.RefDW	

5.7.16 FC875: Aux_DeviceControl_v2（外设控制 v2）

程序功能

通用的对外设的单键和 M 代码启停控制，相比与 FC874（外设控制 v1）,FC875 仅包含控制功能，不考虑安全保护及相关报警输出。功能块具有 Enable 信号，如果不许要执行，将 iEnable 设为 False 即可。

形式参数说明

信号	I/O	类型	含义
iEnable	I	BOOL	设备使能
iKey	I	BOOL	设备控制开关
iEMG(c)	I	BOOL	急停信号，常闭
iReset	I	BOOL	复位信号
iPowerOnReset	I	BOOL	重新上电
sChanNo	I	INT	通道号:1...10;99(所有通道)
sExtMAddr	I	INT	M 功能号:0...99
sMCoolON	I	INT	设备开 M 指令：0...32767
sMCoolOFF	I	INT	设备关 M 指令：0...32767
oDeviceOn	IO	BOOL	设备启动
oDeviceLED	IO	BOOL	设备状态指示灯
RefDW	IO	DWORD	内部状态字(双字)

应用实例

程序	注释
CALL "Aux_DeviceControl_v2"	
iEnable :="GlobalDB".ONE	
iKey :="GlobalDB".MCP.In.customerKey14	
iEMG(c) :="GlobalDB".EMG.Button	
iReset :="GlobalDB".MCP.In.reset	
iPowerOnReset :="GlobalDB".FirstScan	
sChanNo :=1	//M 代码在通道 1 生效
sExtMAddr :=0	
sMCoolON :=8	//M8 控制开启
sMCoolOFF :=9	//M9 控制关闭
oDeviceOn :="oCoolant"	
oDeviceLED :="MCPOut".customerKey14	
RefDW :="GlobalDB".AuxFunc.Coolant_v2.RefDW	

5.7.17 FC876: Aux_SpOriPos（主轴定向）

程序功能

主轴一键定向，定向后指示灯亮，主轴为位控模式；再按一次定向指示灯熄灭，主轴恢复速度模式。可通过接口设定定向角度和速度

形式参数说明

信号	I/O	类型	含义
iKey	I	BOOL	主轴定向按键
iReset	I	BOOL	复位
iEMG	I	BOOL	急停
sSpNr	I	INT	主轴轴号
sOriPos	I	REAL	主轴定向角度
sOriVel	I	REAL	主轴定向速度
oLed	IO	BOOL	主轴定向指示灯
oInPos	IO	BOOL	主轴定向到位
RefDW	IO	Array[0..1] of DWORD	中间寄存器双字数组

应用实例

程序	注释

CALL "Aux_SpOriPos"
iKey :="GlobalDB".MCP.In.axis7
iReset :="GlobalDB".MCP.In.reset
iEMG :="GlobalDB".EMG.Button
sSpNr :=4
sOriPos :=123.4
sOriVel :=500.0
oLed :="MCPOut".axis7
oInPos :="GlobalDB".AuxFunc.SpOriPos.oInPos
RefDW :="GlobalDB".AuxFunc.SpOriPos.RefDW
	//主轴机床轴号 4
	//主轴定向角度
	//主轴定向速度

5.7.18 FC877: Aux_Lubrication_v3（润滑控制 v3）

程序功能

此功能块用于控制设备中心润滑单元，中心润滑单元可以通过按键、系统重新启动的方式启动。相较与 FC866，优化了控制逻辑和接口信号，取消了 M 代码控制。

中心润滑单元工作的三个阶段：

阶段 1：中心润滑单元启动，润滑压力建立，在设定时间内，系统压力未达到要求压力，压力报警输出。

阶段 2：润滑系统建立压力正常后，润滑单元工作才开始计时，至设置的润滑时间结束。

阶段 3：在润滑结束后，润滑间隔时间开始计时，到达计时时间，润滑单元再次启动。

当系统系统出现故障时，下面的报警信号将被输出

- 润滑保护开关掉闸
- 润滑系统液位过低
- 润滑系统压力不足

前提条件

- 液位报警信号 iLubLevel 为 0
- 润滑压力信号 iLubPress 为 1
- 润滑泵过载信号 iLubOVL 为 1

形式参数说明

信号	I/O	类型	含义
iEnable	I	BOOL	使能信号
iLubKey	I	BOOL	润滑启动按钮
iLubLevel(o)	I	BOOL	液位开关信号（常开）
iLubPress(c)	I	BOOL	润滑压力信号（常闭）
iLubOVL(c)	I	BOOL	润滑单元保护触点（常闭）
iPowerOnReset	I	BOOL	设备上电信号
iReset	I	BOOL	复位信号
iEMG	I	BOOL	急停信号
iContrVoltON	I	BOOL	控制电源 OK
iFlag_1Hz	I	BOOL	1Hz 信号
sLubInterval	I	INT	润滑间隔时间，单位：min；范围：0...32767
sLubOnTime	I	INT	润滑单元启动时间，单位：sec；范围：0...32767
sMonitoringTime	I	INT	监控时间，单位：sec；范围：0...32767
oLubrication	IO	BOOL	中心润滑单元
oLubricationLED	IO	BOOL	中心润滑单元工作指示
oMsgOillevel	IO	BOOL	中心润滑单元液位报警
oMsgMotorProt	IO	BOOL	中心润滑单元保护报警
oMsgPressure	IO	BOOL	中心润滑单元压力报警
oMonitorTime	IO	INT	润滑压力建立计时器
oOffTime	IO	INT	润滑间隔计时器
oOnTime	IO	INT	润滑延时计时器
RefW	IO	WORD	内部状态字

应用实例

程序	注释
CALL "Aux_Lubrication_v2" iEnable :="GlobalDB".ONE iLubKey :="GlobalDB".MCP.In.axis9 iLubLevel(o) :="GlobalDB".AuxFunc.Lubrication_v2."iLubLevel(c)" iLubPress(c) :="GlobalDB".AuxFunc.Lubrication_v2."iLubPress(c)" iLubOVL(c) :="GlobalDB".AuxFunc.Lubrication_v2."iLubOVL(c)" iPowerOnReset :="GlobalDB".FirstScan iReset :="GlobalDB".MCP.In.reset	

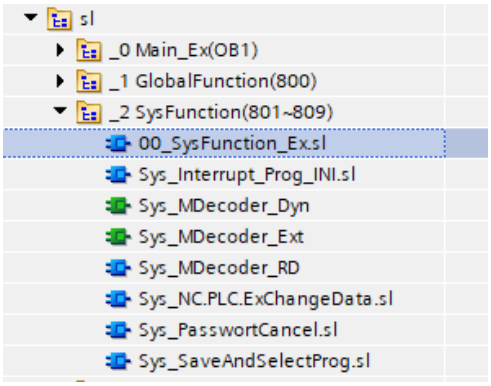
iEMG	:="GlobalDB".EMG.Button	
iContrVoltON	:="GlobalDB".MachineStartOK	
iFlag_1Hz	:="GlobalDB".Clock._1Hz	
sLubInterval	:=50	//润滑间隔 50 分钟
sLubOnTime	:=5	//润滑启动 5 秒
sMonitoringTime	:=20	//润滑压力监测时间 20 秒
oLubrication	:="oLub"	
oLubricationLED	:="MCPOut".axis9	
oMsgOillevel	:="GlobalDB".AuxFunc.Lubrication_v2.oMsgOillevel	
oMsgMotorProt	:="GlobalDB".AuxFunc.Lubrication_v2.oMsgMotorProt	
oMsgPressure	:="GlobalDB".AuxFunc.Lubrication_v2.oMsgPressure	
oMonitorTime	:="GlobalDB".AuxFunc.Lubrication_v2.oMonitorTime	
oOffTime	:="GlobalDB".AuxFunc.Lubrication_v2.oOffTime	
oOnTime	:="GlobalDB".AuxFunc.Lubrication_v2.oOnTime	
RefW	:="GlobalDB".AuxFunc.Lubrication_v2.RefW	

6 编程示例说明

库中 00_开头的功能为编程示例功能块，主要用于如何使用功能块。

每个功能组都有一个或多个示例程序块，用户可直接使用示例功能块或截取示例功能块中的部分代码用于自己的项目。

如：SysFunction 中的 00_SysFucntion_ex.sl 程序块为示例模块。



6.1 铣床（840DSL）

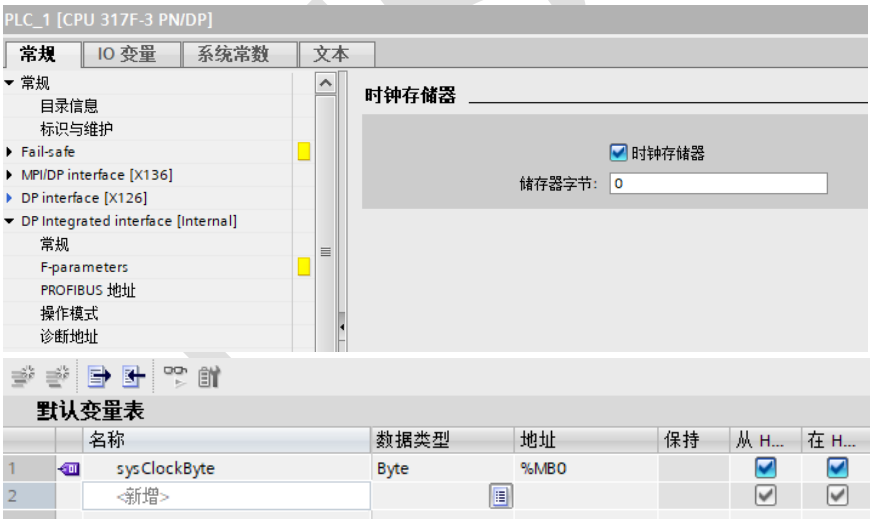
6.1.1 系统配置和组态

硬件配置

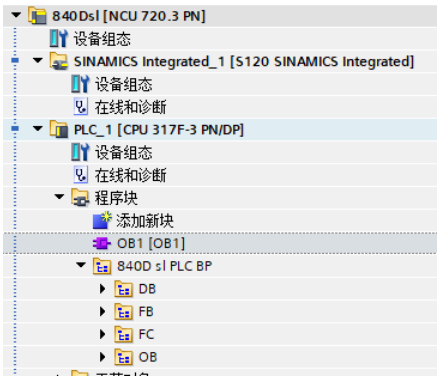
NCU720.3B+MCP483+HT2+6*轴（X，Y，Z，A，C，SP）

设置时钟存储器

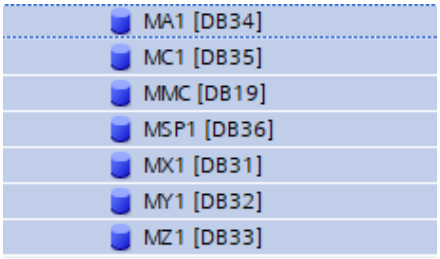
MB0 设置时钟存储器，并定义变量名称



拷贝 840Dsl 标准库文件

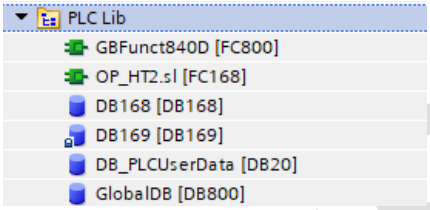


更改轴 DB 块名称，便于编程识别



6.1.2 PLC 编程

使用以下库文件

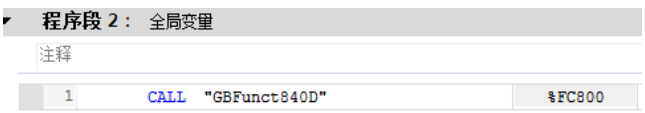


OB100

定义 MCP 和 HT2 地址

```
CALL  "RUN_UP", "gp_par"
   MCPNum          :=1
   MCP1In          :=P#I0.0
   MCP1Out         :=P#Q0.0
   MCP1StatSend    :=P#Q8.0
   MCP1StatRec     :=P#Q12.0
   MCP1BusAdr      :=192
   MCPBusType      :=B#16#55
   BHG             :=5
   BHGIn           :=P#I20.0
   BHGOut          :=P#Q20.0
   NCKomm          :=1
```

刷新全局变量



激活 MCP 和 HT2

```
CALL  "OP_HT2"
   BHG_on_condition     :=True
   BHG_stop             :=False
   inch                 :=False
   BHG_activ            :="GlobalDB".HandWheel.HT2.Active
   BAGNo                :="GlobalDB".HandWheel.HT2.BAGNo
   Menu                 :="GlobalDB".HandWheel.HT2.Menu
   ChanNo               :="GlobalDB".HandWheel.HT2.ChanNo

CALL  "MCP_IFM"
   BAGNo                :=B#16#1
   ChanNo               :="GlobalDB".HandWheel.HT2.ChanNo
   SpindleIFNo          :=b#16#6
   FeedHold             :="GlobalDB".MCP.status.FeedHold
   SpindleHold          :="GlobalDB".MCP.status.SpHold
```

报警

OB1 最后一个网络调用 FC10

```
CALL  "AL_MSG"
   ToUserIF             :=True
   Quit                 :="GlobalDB".MCP.In.reset
```

6.2 铣床（SINUMERIK ONE）

6.2.1 系统配置和组态

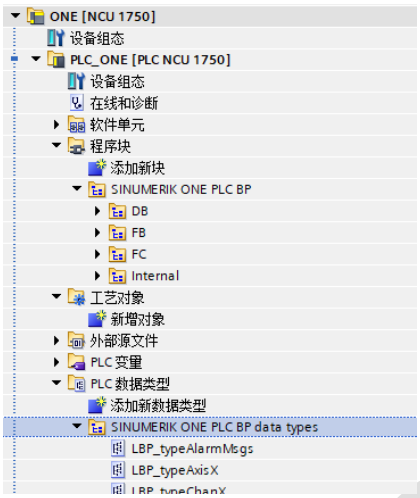
硬件配置

NCU1750+MCP483+BHG+6*轴（X，Y，Z，A，C，SP）

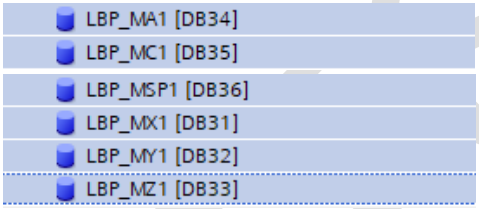
设置时钟存储器

MB0 设置时钟存储器，更改变量名称 Clock_Byte-> sysClockByte

拷贝 ONE 标准库文件和数据类型文件



更改轴 DB 块名称，便于编程识别



6.2.2 PLC 编程

OB100

定义 MCP 地址

```
CALL  "LBP_ConfigBP"
   MCPNum          :=1
   MCP1In          :=P#I0.0
   MCP1Out         :=P#Q0.0
   MCP1BusAdr      :=192
   MCPBusType      :=B#16#55
```

刷新全局变量

```
CALL  "GBFunct840D"
AN    "GlobalDB".NC.Ready
BEC
```

激活 MCP 和第三方手轮

```
CALL  "OP_3rd_BHG"
   EnableKey          :=True
   Chan_No            :=1
   MCP_No             :=1
   Ax1                :="GlobalDB".HandWheel.BHG.Ax1
   Ax2                :="GlobalDB".HandWheel.BHG.Ax2
   Ax3                :="GlobalDB".HandWheel.BHG.Ax3
   Ax4                :="GlobalDB".HandWheel.BHG.Ax4
   Ax5                :="GlobalDB".HandWheel.BHG.Ax5
   x1                 :="GlobalDB".HandWheel.BHG.x1
   x10                :="GlobalDB".HandWheel.BHG.X10
   x100               :="GlobalDB".HandWheel.BHG.X100
   HandWheelActive    :="GlobalDB".HandWheel.BHG.Active
   NoAxisSelect       :="GlobalDB".HandWheel.BHG.NoAxisSelect
   Ref                :="GlobalDB".HandWheel.BHG.mRef

CALL  "LBP_MCPCtrlMilling"
   BAGNo              :=1
   ChanNo             :=1
   SpindleIFNo        :=4
   FeedHold           :="GlobalDB".MCP.status.FeedHold
   SpindleHold        :="GlobalDB".MCP.status.SpHold
```

报警

OB1 最后一个网络调用 FC10

```
CALL  "LBP_GenerateAlarmMsgs"
   ToUserIF           :=true
   Quit               :="GlobalDB".MCP.In.reset
   NumActAlarmMsgs    :="LBP_ConfigData".GenerateAlarmMsgs.NumActAlarmMsgs
   Error              :="LBP_ConfigData".GenerateAlarmMsgs.Error
   StatusID           :="LBP_ConfigData".GenerateAlarmMsgs.StatusID
   Status             :="LBP_ConfigData".GenerateAlarmMsgs.Status
```

7 典型机床基本 PLC 编程示例

7.1 铣床

示例 00_Milling(OB1)，实现铣床基本功能

7.1.1 机床配置

直线轴 X,Y,Z + 旋转轴 SP + 主轴

7.1.2 PLC 程序功能

- 基本程序：FC2，FB25000，FB25002
- 基本功能："MCS_WCS"，FC19
- 辅助控制
  - 急停
  - 驱动系统使能：SLM；ALM
- 轴控制，如使能；限位

8 参考文献

1. FBPLC_0306_en, 03/2006
2. Template_E_01_00_12（SIOS）
3. 840D sl 子程序库手册 V1.3

9 附录

9.1 DB75 扩展 M 功能译码

程序功能

借助“基于列表的 M 解码”功能，可通过基本程序对最多 256 个带扩展地址的 M 功能进行解码。 此功能通过 FC1参数“ListMDecGrp”（需解码的 M 功能组的数量）激活。 带扩展地址的 M 功能通过解码列表指定给信号列表中的信号。 此时还会进行分组。

解码列表 (DB75)

解码列表的源文件” LBP_MFuncDecListConfig”（需要安装SINUMERIK ONE 对应版本Toolbox）随基本程序提供。激活功能前，必须将解码列表(DB75)传输至PLC并执行重启。

当一个 M 功能被包含在解码列表中时，系统会对其进行解码。 M 功能解码时，系统会针对特定组置位信号列表中的信号。 信号列表中信号置位的同时，基本程序会在对应 NC 通道中将接口信号“读取禁止”置位。 一旦用户将该通道输出的所有信号复位（即应答），系统会立即为此通道复位该接口信号。

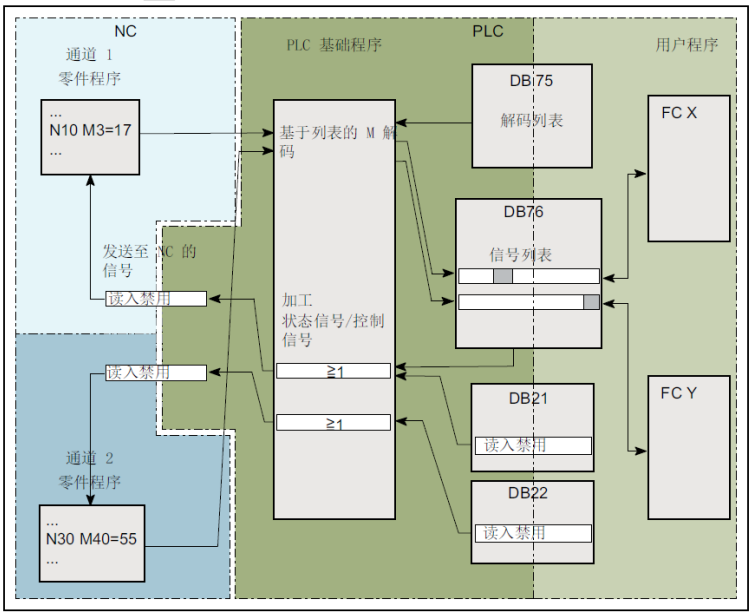
信号列表 (DB76)

功能激活时，基本程序会在数据块 DB76 中建立信号列表。 之后针对每个依据列表解码的 M 信号，信号列表（DB76）的对应组中都会置位一个信号。 同时系统还会在输出 M 功能的通道中置位“读取禁止”信号。 一旦用户将该通道输出的所有信号复位（立即应答），系统会立即为此通道复位该接口信号。

快速辅助功能

在将解码列表中包含的 M 功能作为“快速辅助功能”输出时，系统不会为对应的 NC 通道置位读取禁止。

下图显示了基于列表的 M 解码的结构：



激活

M 解码通过 FC1 参数“ListMDecGrp”激活通过此参数设定待分析或待解码的 M 功能组的数量。 设定的参数值 = 1 ... 16 时，功能激活。

解码列表（DB75）的属性和结构

解码列表（DB75）的属性：

- 解码列表只有一个，为跨通道功能。
- 解码列表最多可包含 16 个组。
- 一个组最多包含 16 个信号
- 对于每个待解码的 M 功能组，解码列表中都包含一条对应的记录
- 带扩展地址的 M 功能在解码列表中通过对应组的第一个和最后一个 M 功能指定给信号列表中的待置位信号。

– 第一个 M 功能：参数： “MFirstAdr” ≙信号或位 0

– 最后一个 M 功能：参数： “MLastAdr” ≙ 取决于与第一个 M 功能偏差的最大信号或位 15

解码列表（DB75）的结构：

解码列表中的一条记录由 3 个参数组成，其分别被指定给一个组：

组	扩展 M 地址	组中的第一个 M 地址	组中的最后一个 M 地址
1	MSigGrp[1].MExtAdr	MSigGrp[1].MFirstAdr	MSigGrp[1].MLastAdr
2	MSigGrp[2].MExtAdr	MSigGrp[2].MFirstAdr	MSigGrp[2].MLastAdr
...	...	...	...
16	MSigGrp[16].MExtAdr	MSigGrp[16].MFirstAdr	MSigGrp[16].MLastAdr

信号的类型和取值范围：

信号	类型	取值范围	含义
MExtAdr	INT	0 至 99	扩展 M 地址
MFirstAdr	DINT	0 至 99,999,999	组中的第一个 M 地址
MLastAdr	DINT	0 至 99,999,999	组中的最后一个 M 地址

信号列表（DB76）的属性

信号列表（DB76）有以下属性：

- 信号列表只有一个，为跨通道功能。
- 信号列表中每个 M 功能组最多包含 16 个信号。

示例

需要对 3 个 M 功能组进行解码：

- 组 1： M2 = 1 至 M2 = 5
- 组 2： M3 = 12 至 M3 = 23
- 组 3： M40 = 55

解码列表和信号列表的结构

组	解码列表 (DB75)			信号列表 (DB76)
	扩展 M 地址	组中的第一个 M 地址	组中的最后一个 M 地址	
1	2	1	5	DB76.DBX0.0 ... DBX0.4
2	3	12	23	DB76.DBX2.0 ... DBX3.3
3	40	55	55	DB76.DBX4.0

说明

在OB100中完成输入并将解码列表（DB75）传输至AG后，必须执行重启。基本程序会在重启过程中创建信号列表（DB76）。接下来系统例如会在通道 1 中启动 NC 程序。 该程序中包含一个扩展 M 功能（M3=17）。 M功能解码（M3 ≙ 组 2）时，系统会置位信号列表（DB76）中的对应信号（DBW1.5），并在通道 1 中置位接口信号“读取禁止”。 NC 程序的执行停止。 此外还会在通道 1 的通道 DB 中显示“M 功能扩展地址”和“M 功能编号”。一旦用户将通道 1 输出的信号列表（DB76）中的所有信号复位，从而完成应答，该通道中的“读取禁止”信号会立即复位。

程序代码（直接修改源文件,然后编译即可）

程序	注释

DATA_BLOCK DB 75
TITLE =
VERSION: 0.0
STRUCT
MSigGrp : ARRAY [1 .. 16] OF STRUCT
MExtAdr : INT;
MFirstAdr : DINT;
MLastAdr : DINT;
END_STRUCT;
BEGIN
MSigGrp[1].MExtAdr := 2
MSigGrp[1].MFirstAdr := L#1
MSigGrp[1].MLastAdr := L#5
MSigGrp[2].MExtAdr := 3
MSigGrp[2].MFirstAdr := L#12
MSigGrp[2].MLastAdr := L#23
MSigGrp[3].MExtAdr := 40
MSigGrp[3].MFirstAdr := L#55
MSigGrp[3].MLastAdr := L#55
END_DATA_BLOCK
	
	//组 1 的扩展 M 地址
	//组中的第一个 M 地址
	//组中的最后一个 M 地址
	//组 2 的扩展 M 地址
	//组中的第一个 M 地址
	//组中的最后一个 M 地址
	//组 3 的扩展 M 地址
	//组中的第一个 M 地址
	//组中的最后一个 M 地址

OB100
Call FC1
...
ListMDecGrp := 3
...
);
	//中FC1的结构插入用于设定待解码 M 功能数量的参数“ListMDecGrp”以激活功能。
	
	//对 3 个组进行 M 解码

NC	PLC

//在MDA或AUTO方式执行
M2=50
	//读入禁止，对应信号置1
	DB76.DBX0.1=1

10 作者/联系人

MTS APC

2020.03.03

11 版本信息

版本	日期	修改内容
V1.0	2020.03.03	草稿，程序处于测试状态
V2.0	2021.06.05	更新刀库管理程序，补充辅助功能程序，修复 Bug
V2.2	2022.01.29	更新刀库管理程序，补充 ONE MCP 程序，修复 Bug